Learning to Slice: Self-Supervised Interpretable Hierarchical Representation Learning with Graph Auto-Encoder Tree

Jinning Li
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
jinning4@illinois.edu

Ruipeng Han
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
ruipeng2@illinois.edu

Jingying Zeng
College of Computing, Georgia Institute of Technology Atlanta, GA, USA
joyzeng@gatech.edu

Dachun Sun
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
dsun18@illinois.edu

Chenkai Sun
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
chenkai5@illinois.edu

Hanghang Tong
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
htong@illinois.edu

ChengXiang Zhai
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
czhai@illinois.edu

Boleslaw K. Szymanski
Department of Computer Science, Rensselaer Polytechnic Institute Troy, NY, USA
szymab@rpi.edu

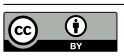
Tarek Abdelzaher
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
zaher@illinois.edu

Abstract

The perceptions and decisions of individuals on social networks are deeply rooted in their intrinsic beliefs, which makes it possible to infer social beliefs from user behavior and message interactions. While existing research models these interactions as graphs and learns their representations, interpretability remains a significant challenge. In real-world scenarios, the interpretation of beliefs is nested within subject scopes of different granularity (such as topics and locations), posing additional challenges for belief discovery. In this paper, we introduce the Interpretable Graph Auto-Encoder Tree (IGAT), a novel end-to-end framework that jointly encodes hierarchical subject scopes and corresponding beliefs as a unified, interpretable hierarchical representation. IGAT integrates the interpretable hierarchy of Model Trees with disentangled representation learning models. We propose a differentiable Slice Mechanism to dynamically optimize internal node splitting and jointly train a leaf model to learn disentangled belief subspaces. The aggregation of these subspaces yields a unified representation, offering interpretations for both subjects and beliefs. Experimental evaluations on three real-world Twitter datasets show that IGAT achieves a consistent improvement of 1.49%-5.61% in F1-score, accuracy, and purity in the belief discovery task, as well as its effectiveness in various downstream analytical applications.

CCS Concepts

• **Computing methodologies** → **Learning latent representations**; • **Information systems** → **Social networks**.



KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737023

Keywords

Interpretable Hierarchical Representation; Non-Negative Graph Auto-Encoder Tree; Belief Discovery; Social Networks

ACM Reference Format:
Jinning Li, Ruipeng Han, Jingying Zeng, Dachun Sun, Chenkai Sun, Hanghang Tong, ChengXiang Zhai, Boleslaw K. Szymanski, and Tarek Abdelzaher. 2025. Learning to Slice: Self-Supervised Interpretable Hierarchical Representation Learning with Graph Auto-Encoder Tree. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737023

KDD Availability Link:
The code and data are available at https://doi.org/10.5281/zenodo.15570095.

1 Introduction

The psychological theories [16, 32] have suggested that the perceptions, judgments, and actions of humans are deeply affected by a set of intrinsic *beliefs*, particularly in the context of social networks [5, 41, 44]. As a result, user behavior on social networks reflects the underlying beliefs of both the users and the disseminated messages. Understanding and modeling these beliefs is crucial for studying and predicting the opinions, preferences, and conflicts for individuals and communities. A common approach to modeling beliefs is through vectorized belief representations (embeddings) [9, 23, 34]. These embeddings enable social analysts and downstream applications by leveraging the coordinates, similarities, and dynamics of the learned belief embeddings.

Modeling social interaction data (such as *retweet*, *like*, and *follow*) as graphs has proven effective, with techniques like graph convolutional networks (GCNs) [12, 39] and graph auto-encoders (GAEs) [20, 33] used to encode social entities (such as *users* and *messages*) into node representations. However, these learned representations often lack **interpretability**, preventing them from

effectively reflecting the beliefs of social entities. Interpretability is essential, as it offers deeper insights into user behaviors and beliefs for researchers and policy-makers. Moreover, interpretable embeddings can enhance downstream performance by explicitly capturing meaningful features, thereby supporting applications such as public crisis prevention and targeted advertising. A recent model, InfoVGAE [23], introduces interpretability by disentangling the embedding space through non-negative and orthogonal constraints, assuming all beliefs are mutually exclusive. However, it does not capture the real-world scenarios where beliefs are subject-specific and mutually exclusive only within the same subject scope (e.g., topic, location). As the real-world example of belief separations shown in Figure 1, the beliefs such as *Help Ukraine* and *Support EDCA*, when across different subjects (topics), may entangled with each other and are difficult to be separated or interpreted without hierarchical modeling. Moreover, multiple subjects with different granularity can form hierarchical tree-like structure. Developing a **hierarchical** representation that captures interpretability at both the subject and belief levels remains a significant challenge.

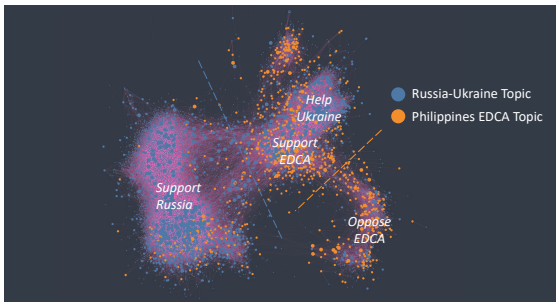


Figure 1: A real-world belief separation scenario. Blue and orange points represent posts, colored by topic. Purple connections indicate positive interactions between sources. Beliefs are separated within topic while entangled across topics.

The Model Tree [31] is known for its interpretability and has proven effective for classification and regression tasks. However, only limited research has investigated its potential in hierarchical graph representation learning, primarily because traditional Model Trees are *discrete* and *non-differentiable*. Meanwhile, other work [23, 38] has demonstrated the effectiveness of non-negativity constraints in promoting embedding disentanglement in both graph learning and contrastive learning. Inspired by these findings, we propose a novel tree-structured, *end-to-end* disentangled hierarchical graph representation learning model named **Interpretable Graph Auto-Encoder Tree** (**IGAT**). IGAT combines the interpretable hierarchy of Model Trees with the representation-disentangling ability of Non-Negative Graph Auto-Encoders (NGAE) to achieves a higher-level interpretability, as illustrated in Figure 2.

The inputs to IGAT consist of interaction data among social entities, as well as the entities' features (e.g., *text*, *images*, *geographic coordinates*). We model these interactions as a heterogeneous graph using an interaction adjacency matrix and preprocess each feature into a similarity matrix. Unlike existing model tree approaches that apply greedy algorithms to find discrete split thresholds based on criteria such as Gini Impurity, Variance, or Information Gain [6, 30], IGAT proposes a differentiable node-splitting model named the Slice Mechanism (SLM). SLM takes an interaction or similarity matrix as input and encodes it into slice vectors using NGAE and softmax, where each dimension represents the likelihood of assigning social entities to a child node's subgraph. It is trained to choose either the social interaction matrix itself (self-slice) or one of the feature similarity matrices (external slice) by minimizing the entropy of the slice vectors. This mechanism is jointly trained with the disentangled belief embedding model in the leaf nodes. While intermediate nodes capture the hierarchical structure of subjects, the leaf node model learns subject-specific disentangled belief subspaces. Aggregating these subspaces yields an interpretable hierarchical representation (IHR) that unifies subject- and belief-level interpretability for the belief discovery task (Section 3.1) and other applications.

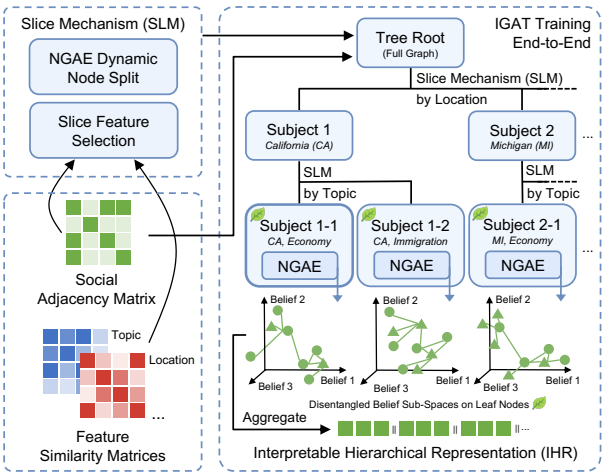


Figure 2: The Interpretable Graph Auto-Encoder Tree (IGAT) model, which dynamically constructs a model tree using the Slice Mechanism (SLM) and trains belief subspaces in leaf nodes to form an Interpretable Hierarchical Representation.

We evaluate the effectiveness of IGAT on three real-world Twitter datasets for the belief discovery task. Our results show that IGAT, while retaining interpretability, achieves an average improvement of 1.49%-5.61% in accuracy, F1-score, and purity over graph-based models, large language models (LLMs), and combined graph-LLM baselines, such as InfoVGAE [23], DIFFPOOL [43], GPT-3.5/4o [18], DeepSeek [10], and TIMME [40]. Qualitative evaluations and ablation studies further underscore IGAT's potential for automated subject and social belief analysis. While IGAT is inherently self-supervised, we additionally propose an alignment-based semi-supervised algorithm to further improve its performance on sparse data without compromising interpretability.

2 Related Work

Classic decision trees (e.g., CART [6]) greedily partition the feature space, and model trees extend this approach by fitting local regression models [31]. Recent work explores non-greedy, optimization-based decision trees for improved interpretability [4], while soft decision trees use probabilistic routing [13] and differentiable decision trees integrate gradient-based optimization [47]. Hierarchical

clustering methods like Genie [15] build a tree-structured clustering hierarchy using Gini or Bonferroni indices, while few studies integrate these tree-based models with graph learning for hierarchical, interpretable representations. Meanwhile, user-message interactions, representing beliefs and preferences, are typically captured via collaborative filtering [29, 42] or non-negative matrix factorization [2], and more recently, graph-based representation learning approaches such as VGAE [20] use a GCN-based encoder to learn latent variables. Interpretable representations often rely on non-negativity constraints [38], disentangled belief embedding [23], or large language model annotations [8, 22, 45], but they lack coarse-grained interpretability. Hierarchical representation learning methods (e.g., CommPOOL [36] and DIFFPOOL [43]) support hierarchical and coarse-grained community assignments but do not yield disentangled or interpretable embeddings. H-DRB [17] employs a rule-based system that trains rules and models iteratively for interpretability, risking suboptimal solutions, and Hier2Vec [14] explains propagations via extracted spanning trees but the yielded embeddings are not interpretable. In contrast, our end-to-end approach unifies a hierarchical model-tree structure with entity-level representation disentanglement, ensuring top-down interpretability of fine-grained beliefs and coarse-grained subjects, enabling dynamic node splitting that incorporates both external features and graph topology. While representation models intrinsically interpret embeddings through dimension-wise disentanglement, stand-alone explainer models, such as GNNInterpreter [37], GLGExplainer [28], and GDM [3], have also been explored to jointly train auxiliary models that match and interpret the original GNN's output distribution. However, these post-hoc approaches may introduce additional bias and complexity when bridging distribution gaps.

3 Problem Statement

3.1 Unsupervised Social Belief Discovery

Background A belief refers to the microcosmic expression of cognitive or conceptual attitudes that individuals hold or express toward a subject (e.g., topics, events) on social networks [35]. Beliefs are communicated, polarized, and are mutually exclusive within the smaller scope of a specific subject. Detecting the hierarchical scopes of subjects and their underlying beliefs remains a significant challenge. Essentially, the proposed *unsupervised belief discovery* task is a *multi-label hierarchical clustering* process aimed at grouping social media messages and users by their beliefs and associated hierarchical subjects. On social networks, in line with the homophily principle [27], users tend to interact with those who share similar beliefs, making it possible to discover these beliefs from social interactions. Since this is an unsupervised clustering process, multiple solutions may emerge. Nonetheless, this task primarily seeks the solution with the most polarized beliefs and subjects, resulting in clearer disentanglement and more effective interpretation.

Task Input We consider standard *social network data* as the input for the social belief discovery task, which includes social entities, their features (metadata), and interaction records. **(i)** The *social entities* typically include *users* and *messages*, denoted by $U = \{u_1, u_2, \dots, u_i\}$ and $M = \{m_1, m_2, \dots, m_j\}$ respectively, where $|U|$ and $|M|$ represent the number of users and messages. Depending on the platform, a message may be a tweet on Twitter, a video on YouTube, an item on Amazon, etc. We treat all exact *duplicates* (e.g., all retweets of the same tweet) as a single message m. Therefore, the total number of distinct social entities is $N = |U| + |M|$. Additionally, we use a notation $v \in U \cup M$ to represent a generic social entity of either a user or a message for brevity. **(ii)** The *features* of entities describe the social entities, such as tweet text, user profile, user demographics, images, etc. These metadata can take various formats, including discrete categories, texts, numbers, or vectors. We assume there are in total $|\mathcal{F}|$ kinds of entity features, denoted by $\mathcal{F} = \{f^1, f^2, \dots, f^k\}$. F_i^k refers to the k-th feature of the i-th entity v_i. An entity may or may not have all kinds of features, i.e. $F_i^k = \varnothing$ is also valid. **(iii)** The *interaction records* I are represented as triplets $(u, r, m) \in I$, where u is a user, m is a message, and r denotes the type of interaction (e.g., *retweet* or *post*).

Objective The goal of the social belief discovery task is to identify beliefs and their associated hierarchical subjects in an *unsupervised* setting. Unlike classification tasks with predefined labels, subjects and beliefs must be inferred from interactions I and entity features F. Consequently, solutions often involve clustering-based methods or interpretable representation learning. For each entity v, the model predicts one or more subject-belief label pairs (L_s, l_b), where $L_s = \{l_s^1, l_s^2, \dots\}$ is a cascade of subject labels, and l_b is a belief label. A hyper-parameter δ (i.e., $|L_s| \le \delta$) limits the maximum number of subject labels; for example, in IGAT, δ corresponds to the tree depth, and each L_s denotes a path from root to leaf. In a fully unsupervised scenario, L_s and l_b may be abstract (e.g., ({*Location-Cluster1*, *Topic-Cluster1*}, *Belief-Cluster2*)), and one can leverage LLMs to automate the naming of clusters.

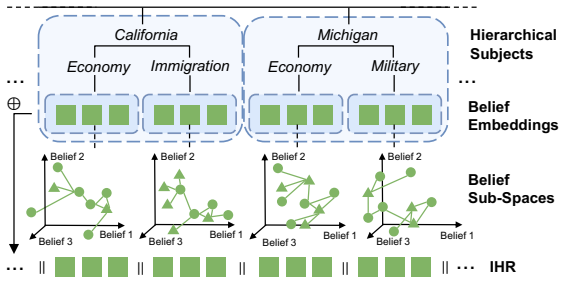


Figure 3: The structure of Interpretable Hierarchical Representation (IHR) with hierarchical subjects and belief spaces.

3.2 Interpretable Hierarchical Representation

In this paper, we tackle the unsupervised social belief discovery task through interpretable representation learning. As illustrated in Figure 3, we propose a novel design of Interpretable Hierarchical Representation (IHR) and introduce the IGAT model to generate it. Unlike traditional graph representations, IHR offers two main interpretable properties:

(1) Each subject partition (e.g. by location) of IHR dimensions forms a belief subspace, described as a cascade of *subject scopes*, representing a path from the tree's root to a leaf.
(2) Each dimension of the IHR under a subject partition represents a belief, with its value indicating the belief's intensity, as an example illustrated in Figure 6.

We introduce three hyperparameters: (i) d, the maximum branching factor of the model tree; (ii) δ, the maximum tree depth; and (iii) η, the dimensionality of each belief subspace. The actual branching factor, depth, and subspace dimensionality are learned dynamically from the data using the tree pruning algorithm in Section 4.4. Consequently, the total dimensionality d_{IHR} of the IHR satisfies $1 \le d_{IHR} \le d^{\delta}\eta$, reaching its upper bound only when no pruning is applied (i.e., a full tree). As shown in Figure 3, the IHR can be viewed as a concatenation of up to d^{δ} belief subspaces, each corresponding to a distinct subject cascade. Using the learned IHR, we derive the solution $s(v)$ for an entity v in the Social Belief Discovery task as

$$s(v) = \{(L_k, l_k) \mid IHR_v^k \ge \tau\}, \tag{1}$$

where IHR_v^k is the k-th dimension of v's IHR, τ is a threshold, L_k denotes the subject cascade for the k-th dimension, and l_k is the belief label represented by that dimension in the belief subspace.

4 Interpretable Graph Auto-Encoder Tree

In this section, we introduce the **I**nterpretable **G**raph Auto-**E**ncoder **T**ree (**IGAT**), which learns interpretable hierarchical representations (IHRs) and addresses the social belief discovery task.

As illustrated in Figure 2, the IGAT model takes social network data as input, consisting of N social entities that include $|U|$ users and $|M|$ messages. We denote users as u, messages as m, and use the generic notation v for either. In IGAT, social interactions I are represented by a heterogeneous, undirected bipartite graph with $|U|$ user nodes and $|M|$ message nodes. For simplicity, the first $|U|$ nodes correspond to users, and the subsequent $|M|$ nodes correspond to messages. We construct a weighted adjacency matrix $\boldsymbol{A}^r \in \mathbb{R}^{N\times N}$ for each edge type r. Graph edges are derived from interaction triplets (u_i, r, m_j): we set $\boldsymbol{A}^r_{i,\,|U|+j} = \boldsymbol{A}^r_{|U|+j,\,i} = 1$ if $(u_i, r, m_j) \in I$, and $\boldsymbol{A}^r_{i,\,j} = 0$ otherwise, where $i = 1, 2, \ldots, |U|$ and $j = 1, 2, \ldots, |M|$. The interaction graph is self-looped, so $\boldsymbol{A}^r_{i,i} = 1$ for all $i = 1, 2, \ldots, N$. Because our model assumes all relations r (e.g., *retweet*, *like*) indicate agreement, we simplify the multi-relation adjacency by taking the element-wise maximum across all $\boldsymbol{A}^r$: $\boldsymbol{A} = \max_r(\boldsymbol{A}^r)$. This process yields a bipartite graph in which edges only connect users and messages, $\boldsymbol{A}_{i,\,j} = 0$ whenever $1 \le i, j \le |U|$ or $i, j \ge |U| + 1$.

The entity feature $f^k \in \mathcal{F}$ of users or messages can appear in various formats in real-world scenarios, such as *discrete categories*, *float numbers*, *coordinates*, *text*, or *vectors*. These entity features serve as node-splitting criteria for forming sub-graphs (subspaces) of child nodes in the IGAT tree, where similar entities in the feature space are placed into the same child. Consequently, the pairwise feature similarity between entities is more important than their raw feature values. Therefore, we construct a similarity matrix $\boldsymbol{S}^k$ for each feature f^k, where $k = 1, 2, \ldots, |\mathcal{F}|$. The shape of $\boldsymbol{S}^k$ may vary with k, as some entities may not have the k-th feature. We assume $\boldsymbol{S}^k \in \mathbb{R}^{N_f^k \times N_f^k}$, where $N_f^k = |\{v \mid F_v^k \ne \varnothing\}|$ is the number of entities whose k-th feature is not empty. We denote by $\xi^k_{i,j}$ the pairwise similarity measure for f^k between two entities. We obtain the similarity matrix $\boldsymbol{S}^k_{i,j} \in [0, 1]$ for the k-th feature via the min-max normalization of $\xi^k_{i,j}$, i.e. $\boldsymbol{S}^k_{i,j} = (\xi^k_{i,j} - \xi^k_{\min})/(\xi^k_{\max} - \xi^k_{\min})$.

Specifically, for discrete categorical features, we set $\xi^k_{i,j} = \mathbf{1}_{F_i^k = F_j^k}$. For floating-point numbers and (geographic) coordinates, we adopt the Euclidean distance $\xi^k_{i,j} = \|F_i^k - F_j^k\|_2$ and then transform the normalized similarity by $1 - \boldsymbol{S}^k_{i,j}$. For vector embeddings, $\xi^k_{i,j} = (F_i^k)^\top F_j^k$. For multi-modal features (e.g., text or images), additional encoders $\boldsymbol{E}^k$ are used (e.g., topic encoders for text, or scene-understanding encoders for images), in which case $\xi^k_{i,j} = [\boldsymbol{E}^k(F_i^k)]^\top \boldsymbol{E}^k(F_j^k)$. By unifying different feature types into similarity matrices, it provides extra flexibility and generalization benefits for the proposed model.

Both the interaction adjacency matrix $\boldsymbol{A}$ and the similarity matrices $\boldsymbol{S}^k \in \boldsymbol{S}$ are used as inputs to train the IGAT model. The adjacency matrix $\boldsymbol{A}$, which describes user interaction behavior, serves as the kernel of the IGAT model to generate belief representations for social entities, while $\boldsymbol{S}^k$ are used as auxiliary slice features for tree node splitting. This process is analogous to data slicing and subgroup analysis in data science and statistics, where the analyst selects a target variable (e.g., disease diagnosis) and partitions the data based on auxiliary features (e.g., age, weight, smoking) to study its behavior within distinct segments. The design of IGAT and its interpretable hierarchical representation simulates this analysis process, providing additional interpretability to the system.

The IGAT framework is inspired by the idea of Model Trees [31], which partition input data into subgroups and fit a simple model within each leaf node, balancing interpretability with predictive accuracy. Traditional model trees rely on a greedy, locally optimal approach, support only discrete splits, and cannot be optimized end-to-end like differentiable models. Recent efforts explore non-greedy, parameterized decision trees that optimize the trade-off between explainability and accuracy [4], but limited research has focused on integrating model trees with graph representation learning for interpretable, hierarchical graph embeddings.

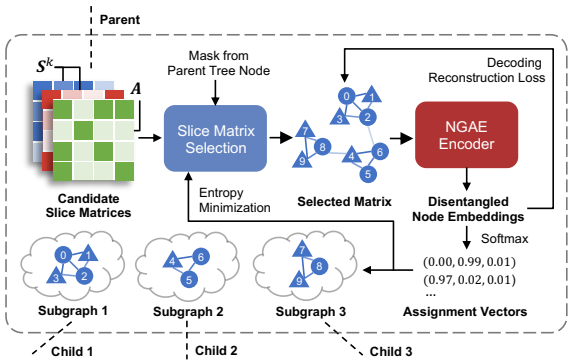


Figure 4: The structure of the Slice Mechanism (SLM), which dynamically slices the graph of parent tree node into multiple subgraphs based on candidate feature matrices.

To fill this gap, we propose a differentiable Slice Mechanism (SLM) as a core component of IGAT. SLM allows for joint optimization of node-splitting criteria with disentangled graph representation learning in the leaf nodes. The structure of SLM is shown in Figure 4. It uses a Non-Negative Graph Auto-Encoder (NGAE) to split parent-node graph data into subgraphs in a soft manner. In the following section, we start with introducing the IGAT tree node.

4.1 IGAT Tree Node

As shown in Figure 2, the IGAT model defines two types of tree nodes: internal nodes and leaf nodes. As mentioned, the initial input to IGAT consists of one interaction adjacency matrix $\boldsymbol{A}$ and k similarity matrices $\boldsymbol{S}^k \in \boldsymbol{S}$. **(i)** The *root node*, as the initial internal node, is denoted by $(\boldsymbol{A}, \boldsymbol{S}, V, P, \boldsymbol{\Phi})$. It represents the full data space containing the entire graph $\boldsymbol{A}$ of all N social entities $V = \{v_1, v_2, \ldots, v_i\}$, where $|V| = N$. All $\boldsymbol{S}^k$ are also assigned to the root node as split criteria. Following the idea of Soft Decision Trees (SDT) [19, 21], IGAT uses a non-discrete split model and defines an assignment probability $P = \{p_1, p_2, \ldots, p_i\}$ (where $|P| = N$), with p_i describing the probability of assigning entity v_i to the current node. In addition, each internal node, including the root node, integrates a trainable node-splitting model based on a Non-Negative Graph Auto-Encoder (NGAE), denoted by $\boldsymbol{\Phi}$, to split the current node with respect to $\boldsymbol{A}$ or $\boldsymbol{S}$. **(ii)** Extending the root node definition, we let $\mathcal{N}_{in}$ denote the set of all *internal tree nodes*. The i-th internal node is $(\boldsymbol{A}_i, \boldsymbol{S}_i, V_i, P_i, \boldsymbol{\Phi}_i) \in \mathcal{N}_{in}$. The first element of $\mathcal{N}_{in}$ is the root node. Here, V_i contains the $N_i = |V_i|$ entities within the scope of the i-th node. For these entities, $\boldsymbol{A}_i \in \mathbb{R}^{N_i \times N_i}$ is the local interaction matrix, and P_i is the assignment probability ($|P_i| = |V_i|$). Moreover, $\boldsymbol{S}_i^k \in \mathbb{R}^{N_{f,i}^k \times N_{f,i}^k}$ denotes the similarity matrix of feature f^k at i-th node. **(iii)** Similarly, let $\mathcal{N}_{lf}$ be the set of all *leaf nodes*, and define the l-th leaf node as $(\boldsymbol{A}_l, \boldsymbol{S}_l, V_l, P_l, \boldsymbol{\Psi}_l) \in \mathcal{N}_{lf}$. While it also uses an NGAE-based approach, the belief embedding learning model $\boldsymbol{\Psi}_l$ on each leaf node differs from the slice model in the internal nodes. As in Section 4.3, $\boldsymbol{\Psi}_l$ is designed without entropy regularization and Softmax normalization to learn disentangled belief subspaces, capturing belief separation under the subject scope of parent nodes.

4.2 Slice Mechanism

4.2.1 Node-Splitting Non-Negative Graph Auto-Encoder (NGAE). Non-negativity constraints have proven effective for inducing disentanglement in self-supervised graph representation learning. Prior work [38] shows that non-negative constraints on image embeddings improve both disentanglement and performance, and InfoVGAE [23] applies non-negative variational graph auto-encoders with auxiliary KL-Divergence control modules to disentangle social graph embeddings. However, it remains unclear how to adopt such non-negative disentanglement for dynamic node splitting in model trees, and a variational distribution is unsuitable for deterministic subgraph assignments. Hence, we introduce a Non-Negative Graph Auto-Encoder (NGAE), adapted and simplified from InfoVGAE [23], for both the internal slice mechanism and leaf-level belief subspace learning. Our modifications include: (i) removing the variational space and KL-Divergence module, using a non-variational embedding space with entropy-minimization to enhance disentanglement; and (ii) introducing Softmax-based assignment vectors to dynamically learn subgraphs (subgroups) and split internal tree nodes.

The NGAE encoder embeds the slice matrix $\boldsymbol{\Lambda} = g(\boldsymbol{A}, \boldsymbol{S})$ into disentangled graph embeddings $\boldsymbol{Z} \in \mathbb{R}^{N_\Lambda \times d}$, where d is the dimension of the latent space (i.e., the default branching factor). g is the selection function introduced in Section 4.2.2. We first apply symmetric normalization [20] to $\boldsymbol{\Lambda}$ based on node degrees, $\bar{\boldsymbol{\Lambda}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{\Lambda} \boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is a diagonal matrix of degrees, with $\boldsymbol{D}_{i,i} = \sum_{j=1}^{N_\Lambda} \boldsymbol{\Lambda}_{i,j}$ for $i = 1, 2, \ldots, N_\Lambda$. We then use an L-layer Graph Convolutional Network (GCN) as the backbone of the NGAE encoder. Let $\boldsymbol{H}^k$ denote the hidden states of the k-th layer and $\boldsymbol{H}^1$ represent the input node feature. The first $L-1$ layers of the encoder are given by

$$\boldsymbol{H}^k = \gamma\big(\bar{\boldsymbol{\Lambda}}\, \boldsymbol{H}^{k-1} \boldsymbol{W}^k\big), \quad 2 \le k \le L-1,$$

where γ is the activation function. In the final output layer, we apply the ReLU function $\mathrm{ReLU}(x) = \max(x, 0)$ to enforce non-negativity in the latent space:

$$\boldsymbol{Z} = \mathrm{ReLU}\big(\bar{\boldsymbol{\Lambda}}\, \boldsymbol{H}^{L-1} \boldsymbol{W}^L\big). \tag{2}$$

The NGAE decoder is a parameter-free inner-product function. Previous work has shown that a non-negative latent space combined with an inner-product decoder can lead to an orthogonal latent space that promotes disentangled dimensions [23]. The decoder computes the likelihood of reconstructing the slice matrix $\boldsymbol{\Lambda}$ from the disentangled representations $\boldsymbol{Z}$ as

$$p(\boldsymbol{\Lambda} \mid \boldsymbol{Z}) = \sigma\big(\boldsymbol{Z}^\top \boldsymbol{Z}\big), \tag{3}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. We use an element-wise cross-entropy loss $\mathcal{L}_{rec}$ to maximize the likelihood of reconstructing the slice matrix:

$$\begin{aligned}\mathcal{L}_{rec}(\boldsymbol{\Phi}, \boldsymbol{\Lambda}) = -\sum_{i,j=1}^{N_\Lambda} \Big[& \omega_p^{\boldsymbol{\Lambda}}\, \boldsymbol{\Lambda}_{i,j}\, \log\, p(\boldsymbol{\Lambda}_{i,j} \mid \boldsymbol{Z}_i, \boldsymbol{Z}_j) \\ & + \big(1 - \boldsymbol{\Lambda}_{i,j}\big)\, \log\Big(1 - p(\boldsymbol{\Lambda}_{i,j} \mid \boldsymbol{Z}_i, \boldsymbol{Z}_j)\Big)\Big],\end{aligned} \tag{4}$$

where $\omega_p^{\boldsymbol{\Lambda}} = \big[\sum_{i,j}(1 - \boldsymbol{\Lambda}_{i,j})\big] / \sum_{i,j} \boldsymbol{\Lambda}_{i,j}$ is the weight for positive samples, balancing the ratio of positive and negative instances across slice matrices with varying sparsity. The above formulation also works when $\boldsymbol{\Lambda}$ is a floating-point feature similarity matrix.

By optimizing the reconstruction loss $\mathcal{L}_{rec}$, all actors and messages are mapped into a disentangled embedding space $\boldsymbol{Z}$, where each dimension represents a subgroup learned from the selected slice matrix $\boldsymbol{\Lambda}$. A higher coordinate in a particular dimension of $\boldsymbol{Z}$ corresponds to a stronger assignment to that group. If no pruning is applied, the number of groups (child nodes) is fixed by the dimension d of the latent space. We then apply a Softmax function to obtain the assignment vector $\bar{\boldsymbol{Z}}_i \in \mathbb{R}^d$ for each entity v_i:

$$\bar{\boldsymbol{Z}}_i = \mathrm{Softmax}(\boldsymbol{Z}_i). \tag{5}$$

The probability of assigning entity v_i to the c-th child node is $p_i^c = \bar{\boldsymbol{Z}}_{i,c} \cdot p_i$, where $p_i \in P$ is the current assignment probability of v_i. An entity v_i is assigned to the c-th child node if $p_i^c \ge \epsilon$:

$$v_i \in V_c,\ p_i^c \in P_c \quad \text{iff} \quad p_i^c \ge \epsilon,\ c = 1, 2, \ldots, d. \tag{6}$$

where V_c and P_c denote the entity set and probability set of the c-th child node, respectively. A special case arises when $\boldsymbol{S}^k$ is selected as the slice matrix but some entities $v_j \in V$ have empty feature $f_j^k = \varnothing$, meaning they are not included in $\bar{\boldsymbol{Z}}$. For these entities, we compute an assignment vector $\widetilde{\boldsymbol{Z}}_j$ via one-hop graph aggregation:

$$\widetilde{\boldsymbol{Z}}_j = \mathrm{Softmax}\big(\boldsymbol{A}_j\, \boldsymbol{Z}\big), \quad \text{if } \boldsymbol{\Lambda} = \boldsymbol{S}^k \text{ and } f_j^k = \varnothing. \tag{7}$$

We then set $p_j^c = \widetilde{\boldsymbol{Z}}_{j,c} \cdot p_j$, and assign v_j to the c-th child node if $p_j^c \ge \epsilon$. The intuition is that when $f_j^k = \varnothing$, v_j should be included in the same subgroup as any adjacent entity v_i already belonging to

that subgroup with $A_{i,j} > 0$. In this way, every entity is assigned to at least one child node, with multi-cast assignment allowed (i.e., an entity can be assigned to multiple child nodes). The adjacency matrix A_c and the feature similarity matrices S_c of the child nodes are derived by extracting the corresponding submatrices for V_c, formally $A_c = A_{[V_c,V_c]}$ and $S_c^k = S_{[V_c,V_c]}^k$.

To enhance the disentanglement of the learned representation Z and the separation of subgroups specified by the assignment vector $\bar{Z}_i$, we simultaneously minimize the entropy of the assignment vector during the NGAE training. The loss function is,

$$\mathcal{L}_e = \frac{1}{N_\Lambda} \sum_{i=1}^{N_\Lambda} H(\bar{Z}_i), \tag{8}$$

where $H(\bar{Z}_i) = -\sum_{c=1}^{d} p(\bar{Z}_{i,c}) \log\big(p(\bar{Z}_{i,c})\big)$ is the Shannon entropy of $\bar{Z}_i$ computed over the latent dimension d. A lower $\mathcal{L}_e$ value indicates a clearer partition of subgroups under the slice criteria of Λ. By optimizing both the reconstruction loss $\mathcal{L}_{rec}$ and the entropy minimization loss $\mathcal{L}_e$, the Node-Splitting NGAE model introduced in this section splits the graph data into subgraphs in a differentiable, non-discrete manner, making it possible to jointly train the dynamic node-splitting model and the target model on leaf nodes.

4.2.2 Slice Matrix Selection. The Node-Splitting NGAE model (as part of the SLM mechanism) takes a slice matrix Λ chosen from either the interaction adjacency matrix A or one of the feature similarity matrices S^k. In this section, we introduce a selection function g to optimize the construction of the IGAT tree by minimizing entropy (i.e., achieving clearest interaction-based separation) after the node split. Assuming after splitting with Λ, there are d candidate child nodes $(A_c, S_c, V_c, P_c, \Phi_c),\ c = 1, \ldots, d$. Using the interaction matrix A_c of each child node , we compute the embedding Z^{A_c} with the encoder after optimizing $\mathcal{L}_{rec}(\Phi_c, A_c)$ in Equation (4). We then select the slice matrix resulting in the minimum weighted averaged entropy over all child nodes, formulated as,

$$g(A,S) = \arg \min_{\Lambda \in \{A\} \cup \{S^k\}_{k=1}^{d}} \sum_{c=1}^{d} \sum_{i=1}^{N_{\Lambda_c}} \frac{p_i^c}{N_{\Lambda_c}} H(\bar{Z}_i^{A_c}),\ g(A,S) \neq \Lambda_a, \forall a, \tag{9}$$

where $\bar{Z}_i^{A_c} = \mathrm{Softmax}(Z_i^{A_c})$ and H is the entropy function. $p_i^c \in P_c$ is the probability of assigning v_i to c-th child. Λ_a is the slice matrix of ancestors a, as child nodes cannot reuse the same slice matrix as any of ancestor nodes. In IGAT, the slice mechanism is referred to as *self-slice* if A is chosen, meaning the node split is based on topology clustering. Conversely, it is called *external-slice* when some feature matrix S^k is selected, implying a split based on external features.

4.2.3 Termination Condition. The node-splitting process terminates under three conditions:

(1) The depth of the current internal node reaches the maximum depth δ (a hyper-parameter).
(2) There is no remaining candidate slice matrix, i.e., $g(A,S) = \emptyset$. In IGAT, an internal node cannot reuse a slice matrix that any of its ancestors has already used.
(3) The entropy loss $\mathcal{L}_e$ of the current node fails to converge ($\mathcal{L}_e \geq 1 - \epsilon$, where ϵ is a small threshold), indicating none of the candidate slice matrices achieve a clear separation.

When the node-splitting procedure terminates, the node becomes a leaf and is equipped with a leaf model (belief embedding model).

4.3 Belief Embedding Model on Leaf Node

The Slice Mechanism introduced in Section 4.2 partitions the original full graph into subgraphs in a hierarchical manner. Each subgraph represents a subspace determined by the split features chosen at each internal node. Once node splitting terminates and yields a leaf node, that leaf node corresponds to a fine-grained subject scope defined by the path from the root node to this leaf. In this section, we discuss how to learn a belief embedding as a leaf node model under each fine-grained subject scope, thereby generating the final Interpretable Hierarchical Representations (IHRs).

As mentioned earlier, each leaf node is defined as a quintuple $(A_l, S_l, V_l, P_l, \Psi_l) \in \mathcal{N}_{lf}$. The goal of the belief embedding learning model Ψ_l is to capture belief clustering of entities based on their interactions, given that these interactions are already confined to a series of fine-grained subjects. Hence, we take the leaf node's interaction matrix A_l as input to the NGAE model introduced in Section 4.2.1. We then compute the disentangled belief representations Z using Equation (2) and optimize the reconstruction loss $\mathcal{L}_{rec}(\Psi, A_l)$ with Equation (4). Unlike the internal node, here we do not apply the Softmax function to compute assignment probabilities, nor do we optimize the entropy loss $\mathcal{L}_e$, because the learned belief representation should reflect the actual belief polarization on the corresponding subject, without imposing additional constraints or assumptions of strong polarization (i.e., low entropy).

The belief embedding NGAE model produces a belief embedding Z_i^l for each entity $v_i \in V_l$. We can then aggregate the belief embeddings for v_i across all leaf nodes to form an IHR:

$$\mathrm{IHR}_{v_i} = \Big\|_{l=1}^{|\mathcal{N}_{lf}|} \big[\mathbf{1}_{v_i \in V_l} \cdot Z_i^l \cdot p_i + \big(1 - \mathbf{1}_{v_i \in V_l}\big) \cdot \vec{\mathbf{0}}\big], \tag{10}$$

where $\|$ denotes concatenation operator, $\mathbf{1}$ is the indicator function, and $p_i \in P_l$ is the assignment probability for v_i. The term $\vec{\mathbf{0}}$ is a zero vector of the same dimension as Z_i^l. In practice, the IHR can be stored as a sparse matrix to save memory.

4.4 Joint Optimization and Tree Pruning

With both the Slice Mechanism for node splitting and the belief embedding model being differentiable, we can jointly train them end-to-end to obtain a more optimal solution for both the learned subject hierarchy of IGAT tree, and the disentangled belief embeddings. The following loss function is used as our joint objective:

$$\mathcal{L} = \frac{1}{|\mathcal{N}_{in}|} \sum_{i=1}^{|\mathcal{N}_{in}|} \Big[\mathcal{L}_{rec}(\Phi_i, \Lambda_i) + \mathcal{L}_{e,i}\Big] + \frac{1}{|\mathcal{N}_{lf}|} \sum_{l=1}^{|\mathcal{N}_{lf}|} \mathcal{L}_{rec}(\Psi_l, A_l). \tag{11}$$

The disentanglement design of NGAE also supports pruning dimensions of the learned representations Z, and assignment vectors $\bar{Z}$ or $\widetilde{Z}$. As mentioned, there are two hyper-parameters d and η defining the initial dimension for node-splitting NGAE (i.e., as the branching factor) and the belief embedding model in leaf nodes. Empirically, some dimensions may be redundant when the actual number of disentangled components in the dataset is smaller than d or η. Since the NGAE dimensions are orthogonal by design, the coordinates in redundant dimensions will converge to the smallest for

all entities, making it possible to detect and drop these dimensions. The pruning is applied to the weight matrix $\boldsymbol{W}^L$ in Equation (2), in the last layer of the NGAE encoder, by dropping out a k-th column,

$$\boldsymbol{W}^L := \left\{ \boldsymbol{W}^L_{[:,j]} \;\middle|\; j \neq k \right\} \text{ if } \exists k \; \forall i, \; k = \arg\min_c (\boldsymbol{Z}_{i,c}). \tag{12}$$

Through this pruning rule, the effective branching factor of the tree is dynamically adjusted during training. We summarize the initialization, joint training, and inference processes of IGAT in Algorithm 1, 2, and 3 respectively, in the Appendix A.

4.5 Alignment-Based Semi-Supervision

While the proposed IGAT model can be trained in an unsupervised fashion using the preprocessed input data $\boldsymbol{A}$ and $\boldsymbol{S}$, we introduce an optional semi-supervision objective that offers two benefits: (i) improving the effectiveness of learned representations; and (ii) assigning concrete names to the dimensions of IHR. This objective relies on *alignment* with the axes of the NGAE's orthogonal latent space and does not compromise interpretability.

Semi-supervision begins by annotating a small subset of the entities, V_{semi}, with a series of labels, provided either by an LLM or a human annotator. Following the definition in Section 3.1, each entity $v \in V_{semi}$ is annotated with one or more label tuples $(L_s, l_b) \in Y_v$, each corresponding to a path from the root node to a leaf node. Here, $L_s = \{l_s^1, l_s^2, \dots\}$ is a sequence of subject labels (i.e., labels for internal node splits), and l_b is the label for the belief model on the leaf node. For each tuple (L_s, l_b), we retrieve the latent embedding $\boldsymbol{z}$ of v along the corresponding root-to-leaf path, which is computed via Equation (2), denoted by $\left(\{z_s^1, z_s^2, \dots\}, z_b\right)$. Defining $\mathcal{L}_{align}(l, \boldsymbol{z})$ as the alignment loss for each NGAE (whether for node-splitting or belief embedding), we write the total semi-supervision loss as:

$$\mathcal{L}_{semi} = \sum_{v \in V_{semi}} \sum_{(L_s, l_b) \in Y_v} \left[\sum_{k=1}^{|L_s|} \mathcal{L}_{align}(l_s^k, z_s^k) \; + \; \mathcal{L}_{align}(l_b, z_b) \right]. \tag{13}$$

The goal of $\mathcal{L}_{align}(l, \boldsymbol{z})$ is to maximize the likelihood that $\boldsymbol{z}$ is correctly aligned with the axis representing label l in the NGAE's orthogonal latent space. As discussed, the axes of the NGAE space are abstract, analogous to unnamed clusters in traditional clustering. We can *name* (*tag*) a disentangled axis of the NGAE to represent label l if that axis is not already named by another label. In other words, we either name a new axis or retrieve an existing named axis matched to l. Assuming that the k_l-th dimension with unit vector $\boldsymbol{e}_l$ is matched with label l, the alignment loss can be formulated as,

$$\begin{aligned} \mathcal{L}_{align}(l, \boldsymbol{z}) = -\frac{1}{d_{\boldsymbol{z}}} \sum_{i=1}^{d_{\boldsymbol{z}}} \Big[& \mathbf{1}_{i=k_l} \; \log \sigma(\boldsymbol{e}_l^\top \boldsymbol{z}) \\ & + \left(1 - \mathbf{1}_{i=k_l}\right) \; \log\Big(1 - \sigma(\boldsymbol{e}_l^\top \boldsymbol{z})\Big) \Big], \end{aligned} \tag{14}$$

where $d_{\boldsymbol{z}}$ is the dimensionality of $\boldsymbol{z}$ and σ is the sigmoid function. The semi-supervision training is then carried out by adding $\mathcal{L}_{semi}$ in Equation (13) to the total loss in Equation (11). Consequently, IGAT benefits from the flexibility of using complex models (e.g., LLMs) to annotate a small subset of entities, greatly enhancing generalization and especially the performance on sparse data, without incurring substantial extra computational overhead.

5 Experiments

We evaluate the performance of the proposed IGAT model on 3 real-world Twitter datasets including **Philippines** [22], **Russia-Ukraine War** [24], and **Election** [23]. We extend the three datasets with manual annotations for belief discovery task. We compare with 17 baselines, including *Graph Models* (VGAE [20], InfoVGAE [23], DIFFPOOL [43]), *Clustering Models* (Ginie [15], RoBERTa-KM [25], T-RoBERTa-KM [26], TwhinBERT-KM [46]), *LLMs* (DeepSeek-V3 [10], GPT-3.5 [1], GPT-4o [18], Llama 3.3 [11]), and *Graph-LLM models* (TIMME [40], SGVGAE [22]). We also evaluate model performance under the semi-supervised setting with 5% labels if applicable. The reproducibility details of datasets, annotations, baselines, and environments are introduced in Appendix B.

5.1 Social Belief Discovery Task Evaluation

We evaluate the proposed IGAT model and baselines for the social belief discovery task, which is essentially a multi-label hierarchical clustering task, as described in Section 3.1. Table 1 shows the results for accuracy, weighted F1 score, and purity score. Each experiment is repeated 10 times with different initializations. Details of the evaluation metrics are provided in Appendix B.3.

The proposed IGAT model outperforms the baselines on all evaluation metrics, except for accuracy on the Election dataset. Across the three datasets, unsupervised IGAT achieves improvements of 4.94%, 5.61%, and 1.49% over the strongest baseline. In the 5%-semi-supervised setting, IGAT also achieves the best performance on all metrics, showing improvements of 3.91%, 4.77%, and 1.78%. On the Election dataset, IGAT's performance is closer to that of the baselines, while on the Russia-Ukraine War dataset, the improvement is more significant. This disparity is explained by the sparsity of the social interaction graph, as the average entity degree in the Russia-Ukraine War dataset (22.46) is higher than in the Election dataset (2.95) and the Philippines dataset (6.92). We also observe an improvement of other graph-based models (e.g., DIFFPOOL, TIMME) as graph density increases. Meanwhile, the performance of language-based models (e.g., GPT-4o, DeepSeek) does not vary as significantly with graph density. Nevertheless, the language-based models, despite having a large number of parameters, seem to face an "upper bound" of around 70-80% in tweet understanding due to the short-form nature of tweets, which often include sarcasms and ambiguities that may not explicitly convey a user's belief. Finally, the semi-supervised version of IGAT leverages 5% LLM-generated annotations to guide its training, enhancing its robustness and performance even on sparse Election dataset.

5.2 Interpretability Analysis

In this section, we analyze the interpretability of the learned interpretable hierarchical representation (IHR). We compute the correlation between the IHR dimensions and visualize the results as a heatmap in Figure 5. The IHR dimensions reflect both the subjects captured in internal nodes and the beliefs learned in leaf nodes. The dimension labels were generated by GPT-3.5 after examining the top 20 tweets for each dimension. By studying these correlations, we can draw several useful conclusions from the dataset. For instance, the secessionists also hold the belief that the Philippines government is corrupt and opposes the socioeconomic policies of

Table 1: Evaluation results for the belief discovery task, measured using accuracy, F1 score, and purity metrics. The table presents results under both the unsupervised setting (upper section) and the 5%-semi-supervised setting (lower section). Each experiment is repeated 10 times, where applicable, to report the mean and standard deviation of the scores for robustness.

Dataset			Philippines			Russia-Ukraine War			Election		
Model Name	Param	Time	Acc. (%)	F1 Score (%)	Purity (%)	Acc. (%)	F1 Score (%)	Purity (%)	Acc. (%)	F1 Score (%)	Purity (%)
VGAE [20]	141K	29.25s	$68.20^{\pm 2.84}$	$69.03^{\pm 2.68}$	$71.56^{\pm 2.79}$	$78.83^{\pm 3.73}$	$78.71^{\pm 3.54}$	$78.54^{\pm 3.41}$	$59.83^{\pm 2.38}$	$60.16^{\pm 2.26}$	$60.74^{\pm 2.15}$
InfoVGAE [23]	154K	29.60s	$78.17^{\pm 2.24}$	$77.66^{\pm 2.47}$	$81.19^{\pm 2.43}$	$90.37^{\pm 2.41}$	$90.45^{\pm 2.36}$	$91.40^{\pm 2.34}$	$71.16^{\pm 2.01}$	$70.43^{\pm 1.90}$	$72.48^{\pm 2.14}$
DIFFPOOL [43]	242K	44.72s	$79.30^{\pm 1.63}$	$79.86^{\pm 1.83}$	$82.38^{\pm 1.92}$	$91.60^{\pm 1.81}$	$92.19^{\pm 1.73}$	$91.08^{\pm 1.77}$	$72.99^{\pm 2.15}$	$72.50^{\pm 2.10}$	$73.34^{\pm 2.17}$
Ginie [15]	0K	11.53s	$70.08^{\pm 2.45}$	$71.42^{\pm 2.36}$	$74.20^{\pm 2.40}$	$74.20^{\pm 3.31}$	$73.62^{\pm 3.10}$	$73.86^{\pm 3.24}$	$64.01^{\pm 2.78}$	$64.81^{\pm 2.63}$	$66.11^{\pm 2.73}$
RoBERTa-KM [25]	125M	30.43s	$56.64^{\pm N/A}$	$55.96^{\pm N/A}$	$56.12^{\pm N/A}$	$64.15^{\pm N/A}$	$58.35^{\pm N/A}$	$58.44^{\pm N/A}$	$52.33^{\pm N/A}$	$51.94^{\pm N/A}$	$54.90^{\pm N/A}$
T-RoBERTa-KM [26]	125M	18.37s	$62.24^{\pm N/A}$	$61.44^{\pm N/A}$	$61.42^{\pm N/A}$	$65.85^{\pm N/A}$	$58.35^{\pm N/A}$	$58.11^{\pm N/A}$	$56.67^{\pm N/A}$	$55.32^{\pm N/A}$	$62.82^{\pm N/A}$
TwhinBERT-KM [46]	279M	21.08s	$58.41^{\pm N/A}$	$57.32^{\pm N/A}$	$57.31^{\pm N/A}$	$63.66^{\pm N/A}$	$54.86^{\pm N/A}$	$54.77^{\pm N/A}$	$62.67^{\pm N/A}$	$60.77^{\pm N/A}$	$61.72^{\pm N/A}$
DeepSeek-V3 [10]	671B	911.2s	$70.09^{\pm 1.13}$	$69.61^{\pm 1.12}$	$69.60^{\pm 1.10}$	$70.07^{\pm 1.38}$	$66.17^{\pm 1.32}$	$66.05^{\pm 1.11}$	$73.10^{\pm 1.11}$	$72.81^{\pm 1.07}$	$72.78^{\pm 1.03}$
GPT-3.5 [1]	175B	144.3s	$64.99^{\pm 1.21}$	$64.17^{\pm 1.27}$	$64.11^{\pm 1.26}$	$57.22^{\pm 0.35}$	$52.63^{\pm 0.33}$	$54.09^{\pm 0.30}$	$61.90^{\pm 0.67}$	$61.79^{\pm 0.66}$	$62.09^{\pm 0.63}$
GPT-4o [18]	180B	303.8s	$73.63^{\pm 0.52}$	$72.68^{\pm 0.56}$	$72.64^{\pm 0.54}$	$79.02^{\pm 0.98}$	$75.26^{\pm 0.85}$	$73.95^{\pm 0.75}$	$\mathbf{75.83}^{\pm 0.39}$	$75.19^{\pm 0.37}$	$75.55^{\pm 0.45}$
Llama 3.3 [11]	70B	729.9s	$58.85^{\pm 3.45}$	$58.58^{\pm 3.51}$	$59.27^{\pm 3.50}$	$57.51^{\pm 1.45}$	$55.98^{\pm 1.35}$	$60.70^{\pm 1.31}$	$66.37^{\pm 2.07}$	$66.26^{\pm 2.04}$	$66.62^{\pm 1.89}$
TIMME [40]	176K	48.09s	$72.86^{\pm 1.59}$	$75.58^{\pm 1.63}$	$78.30^{\pm 1.50}$	$85.61^{\pm 2.31}$	$84.30^{\pm 2.21}$	$84.11^{\pm 2.02}$	$68.12^{\pm 1.78}$	$66.71^{\pm 1.86}$	$68.19^{\pm 1.78}$
IGAT (Ours)	346K	96.21s	$\mathbf{83.85}^{\pm 1.86}$	$\mathbf{83.20}^{\pm 2.47}$	$\mathbf{86.43}^{\pm 1.77}$	$\mathbf{96.93}^{\pm 2.50}$	$\mathbf{96.96}^{\pm 2.46}$	$\mathbf{96.39}^{\pm 1.06}$	$75.08^{\pm 2.42}$	$\mathbf{75.40}^{\pm 2.47}$	$\mathbf{79.46}^{\pm 2.53}$
Semi-RoBERTa [25]	125M	67.32s	$54.48^{\pm 5.03}$	$53.79^{\pm 4.91}$	$44.85^{\pm 5.12}$	$61.95^{\pm 3.53}$	$56.96^{\pm 2.66}$	$57.76^{\pm 2.07}$	$62.43^{\pm 3.37}$	$62.23^{\pm 3.57}$	$63.83^{\pm 3.44}$
Semi-T-RoBERTa [26]	125M	56.74s	$56.40^{\pm 3.37}$	$55.81^{\pm 3.20}$	$56.13^{\pm 3.04}$	$63.29^{\pm 1.19}$	$57.12^{\pm 1.07}$	$57.29^{\pm 1.01}$	$61.33^{\pm 3.89}$	$60.98^{\pm 3.83}$	$62.63^{\pm 3.14}$
Semi-TwhinBERT [46]	279M	97.70s	$56.28^{\pm 1.75}$	$55.30^{\pm 1.84}$	$55.36^{\pm 1.85}$	$60.34^{\pm 5.02}$	$52.31^{\pm 4.26}$	$56.19^{\pm 2.56}$	$60.47^{\pm 2.75}$	$60.25^{\pm 2.76}$	$61.85^{\pm 2.39}$
Semi-TIMME [40]	176K	62.88s	$80.35^{\pm 1.62}$	$80.19^{\pm 1.38}$	$83.34^{\pm 1.49}$	$92.74^{\pm 1.95}$	$91.58^{\pm 1.89}$	$92.92^{\pm 2.04}$	$76.06^{\pm 1.50}$	$75.91^{\pm 1.52}$	$78.38^{\pm 1.46}$
Semi-SGVGAE [22]	154K	33.25s	$81.68^{\pm 1.77}$	$81.27^{\pm 1.79}$	$83.18^{\pm 1.79}$	$93.60^{\pm 2.46}$	$92.21^{\pm 2.53}$	$92.31^{\pm 2.39}$	$75.98^{\pm 2.35}$	$76.92^{\pm 2.19}$	$78.56^{\pm 2.22}$
Semi-IGAT (Ours)	346K	105.3s	$\mathbf{84.41}^{\pm 1.35}$	$\mathbf{84.50}^{\pm 1.51}$	$\mathbf{86.85}^{\pm 0.65}$	$\mathbf{97.13}^{\pm 2.43}$	$\mathbf{97.17}^{\pm 2.47}$	$\mathbf{97.08}^{\pm 0.71}$	$\mathbf{77.63}^{\pm 1.89}$	$\mathbf{77.50}^{\pm 1.71}$	$\mathbf{79.33}^{\pm 0.69}$

President Marcos. Meanwhile, those who support EDCA also exhibit beliefs of anti-China migration, pro-U.S. military presence, and pro-Marcos government which facilitated the EDCA.

In addition to examining correlations among different subjects and beliefs, we also explore the fine-grained belief embeddings under each subject. Figure 6 illustrates the embeddings of users and tweets attributed to the EDCA subject, where entities opposing EDCA lie along the x-axis while those supporting EDCA lie along the y-axis. The figure demonstrates a clear separation between entities with different beliefs, which is also reflected by the tweets assigned to each belief. The belief embeddings under other detected subjects can be found at Appendix C.

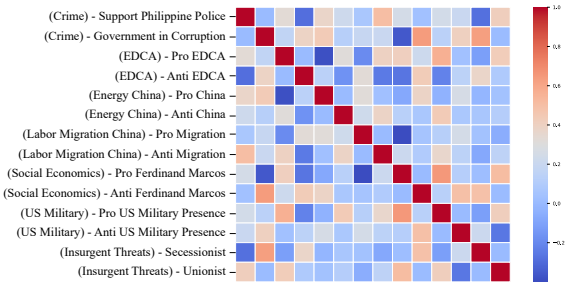


Figure 5: Correlation heatmap of IHR dimensions on the Philippines dataset. Red indicates positive, blue negative correlations. Bracketed terms denote detected subjects.

5.3 Effects of Semi-Supervision Scale

Section 4.5 presents an alignment-based method to enhance model performance and robustness, especially in sparse interaction graphs.

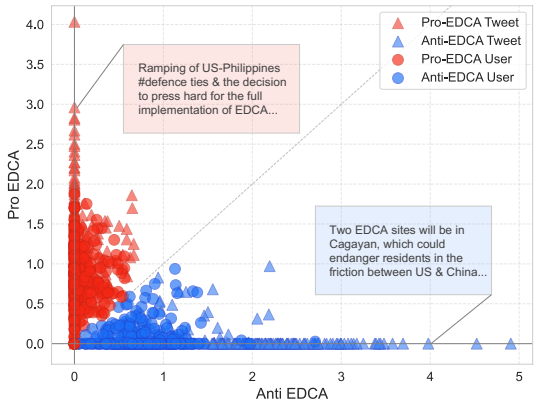


Figure 6: Belief embedding for EDCA subject. The x-axis and y-axis represent Anti-EDCA and Pro-EDCA. Triangles denote tweets, circles denote users, showing a clear belief separation.

We investigate the effect of semi-supervision utilization using LLM annotations. In Figure 7, we vary the percentage of semi-supervision and observe the resulting evaluation metrics. The result shows consistent improvements on the Election dataset as semi-supervision scale increases. This is attributed to the sparsity of Election dataset, where the graph alone provides insufficient information. In contrast, the Russia-Ukraine dataset benefits less from higher percentages of language-based supervision since its dense interaction graph already offers reliable information. Thus, we empirically choose the optimal graph-text trade-off based on graph sparsity. In our experiments, using 5% semi-supervision yields sufficient gains on sparse datasets without degrading performance on dense ones.

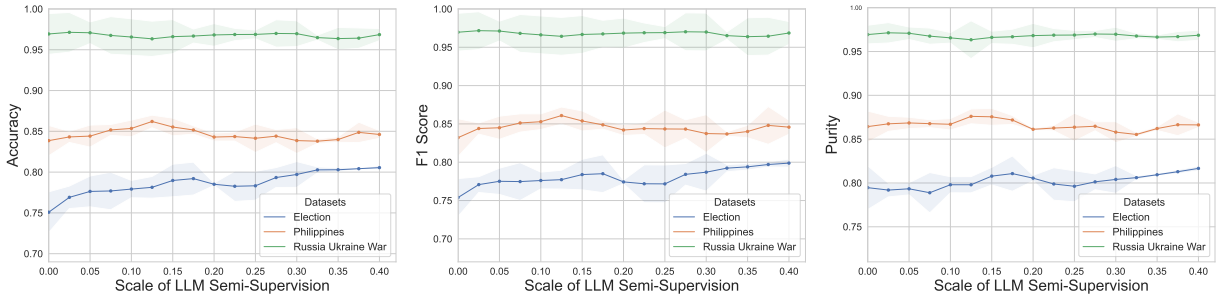


Figure 7: The performance of IGAT model under varying scales of LLM semi-supervision on accuracy, F1 score, and purity.

5.4 Ablation Study and Efficiency Analysis

We evaluate IGAT's performance after removing or replacing its sub-modules. We conduct four experiments on each dataset: (i) replacing the slice mechanism (SLM) with a clustering-based greedy splitting model, (ii) removing the entropy regularizer in Equation (8), (iii) avoiding joint training of node splitting and the leaf node model, and (iv) training without pruning the tree. The results in Table 2 show that: (i) removing SLM reduces IGAT to a simpler disentangled graph embedding model, causing a 6.98% drop, (ii) removing the entropy regularizer weakens internal node separation, leading to a 2.51% drop, (iii) avoiding joint training limits the global optimal solution for splitting and embedding, lowering performance by 4.64%, and (iv) training without pruning introduces redundant dimensions of learned IHR, resulting in a 2.35% drop. We also observe that sparser datasets (e.g., Election) show larger performance losses after removing sub-modules, underscoring the importance of each sub-module for IGAT's robustness.

Table 2: Ablation study results after removing or replacing sub-modules of the proposed IGAT model.

Dataset	Model Name	Accuracy	F1 Score	Purity
Philippines.	Replace Slice Mechanism	75.93%	75.94%	79.02%
	Without Entropy Reg.	81.53%	81.13%	83.68%
	Without Joint Training	78.46%	77.71%	81.42%
	Without Pruning	81.07%	79.34%	83.56%
	IGAT	**83.85%**	**83.20%**	**86.43%**
Ukraine	Replace Slice Mechanism	89.75%	89.86%	89.56%
	Without Entropy Reg.	92.94%	94.46%	92.82%
	Without Joint Training	90.82%	91.01%	91.21%
	Without Pruning	94.66%	95.95%	94.95%
	IGAT	**96.93%**	**96.96%**	**96.36%**
Election	Replace Slice Mechanism	69.24%	69.01%	68.46%
	Without Entropy Reg.	73.86%	74.35%	73.85%
	Without Joint Training	72.65%	72.72%	72.37%
	Without Pruning	73.07%	73.18%	73.86%
	IGAT	**75.08%**	**75.40%**	**79.46%**

We empirically evaluate the efficiency of the proposed IGAT model and baselines in Table 1, which reports parameter utilization and running times measured in the environment described in Appendix B, except for closed-source models like GPT (API-based). Graph-based models (VGAE, InfoVGAE, IGAT) use relatively few parameters (141K–346K), while LLMs require far more (125M–671B). IGAT has parameter utilization and running times comparable to graph baselines but much lower than LLM baselines, achieving a consistent improvement over all the baselines.

6 Conclusion

In this paper, we presented the Interpretable Graph Auto-Encoder Tree (IGAT), an end-to-end model that unifies hierarchical subject scopes and corresponding beliefs into a single interpretable representation. IGAT leverages a differentiable Slice Mechanism for dynamic node splitting and learns disentangled belief subspaces at each leaf, which, when combined, form a coherent hierarchy of subjects and beliefs. Experimental results on three real-world Twitter datasets confirm IGAT's effectiveness, showing a 1.49%–5.61% improvement in belief discovery (F1-score, accuracy, purity), as well as promising performance in downstream analytical tasks.

Acknowledgments

Research reported in this paper was sponsored in part by the DARPA award HR001121C0165, the DARPA award HR00112290105, and the DoD Basic Research Office award HQ00342110002. It was also partially supported by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Md Tanvir Al Amin, Charu Aggarwal, Shuochao Yao, Tarek Abdelzaher, and Lance Kaplan. 2017. Unveiling polarization in social networks: A matrix factorization approach. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[3] Steve Azzolin, Antonio Longa, Pietro Barbiero, Pietro Liò, and Andrea Passerini. 2022. Global explainability of gnns via logic combination of learned concepts. *arXiv preprint arXiv:2210.07147* (2022).

[4] Maria-Florina Balcan and Dravyansh Sharma. 2024. Learning accurate and interpretable decision trees. *arXiv preprint arXiv:2405.15911* (2024).

[5] Alessandro Bessi. 2016. Personality traits and echo chambers on facebook. *Computers in Human Behavior* 65 (2016), 319–324.

[6] Leo Breiman. 2017. *Classification and regression trees*. Routledge.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[8] Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, et al. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2502.13260* (2025).

[9] Kareem Darwish, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. 2020. Unsupervised user stance detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 141–152.
[10] DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs.CL] https://arxiv.org/abs/2412.19437
[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
[12] Ahmed El-Kishky, Thomas Markovich, Serim Park, Chetan Verma, Baekjin Kim, Ramy Eskander, Yury Malkov, Frank Portman, Sofía Samaniego, Ying Xiao, et al. 2022. Twhin: Embedding the twitter heterogeneous information network for personalized recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2842–2850.
[13] Nicholas Frosst and Geoffrey Hinton. 2017. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* (2017).
[14] Shun Fu, Guoyin Wang, and Ji Xu. 2021. hier2vec: interpretable multi-granular representation learning for hierarchy in social networks. *International Journal of Machine Learning and Cybernetics* 12, 9 (2021), 2543–2557.
[15] Marek Gagolewski, Maciej Bartoszuk, and Anna Cena. 2016. Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. *Information Sciences* 363 (2016), 8–23.
[16] Jesse Graham, Jonathan Haidt, Sena Koleva, Matt Motyl, Ravi Iyer, Sean P Wojcik, and Peter H Ditto. 2013. Moral foundations theory: The pragmatic validity of moral pluralism. In *Advances in experimental social psychology*. Vol. 47. Elsevier, 55–130.
[17] Xiaowei Gu and Plamen P Angelov. 2020. Highly interpretable hierarchical deep rule-based classifier. *Applied Soft Computing* 92 (2020), 106310.
[18] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).
[19] Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural computation* 6, 2 (1994), 181–214.
[20] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
[21] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. 2015. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision*. 1467–1475.
[22] Jinning Li, Ruipeng Han, Chenkai Sun, Dachun Sun, Ruijie Wang, Jingying Zeng, Yuchen Yan, Hanghang Tong, and Tarek Abdelzaher. 2024. Large Language Model-Guided Disentangled Belief Representation Learning on Polarized Social Graphs. In *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.
[23] Jinning Li, Huajie Shao, Dachun Sun, Ruijie Wang, Yuchen Yan, Jinyang Li, Shengzhong Liu, Hanghang Tong, and Tarek Abdelzaher. 2022. Unsupervised Belief Representation Learning with Information-Theoretic Variational Graph Auto-Encoders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1728–1738.
[24] Xinyi Liu, Ruijie Wang, Dachun Sun, Jinning Li, Christina Youn, You Lyu, Jianyuan Zhan, Dayou Wu, Xinhe Xu, Mingjun Liu, et al. 2023. Influence pathway discovery on social media. In *2023 IEEE 9th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 105–109.
[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[26] Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. TimeLMs: Diachronic Language Models from Twitter. arXiv:2202.03829 [cs.CL]
[27] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
[28] Yi Nian, Yurui Chang, Wei Jin, and Lu Lin. 2024. Globally interpretable graph learning via distribution matching. In *Proceedings of the ACM Web Conference 2024*. 992–1002.
[29] Manh Cuong Pham, Yiwei Cao, Ralf Klamma, and Matthias Jarke. 2011. A clustering approach for collaborative filtering recommendation using social network analysis. *J. Univers. Comput. Sci.* 17, 4 (2011), 583–604.
[30] J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning* 1, 1 (1986), 81–106.
[31] John R Quinlan et al. 1992. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, Vol. 92. World Scientific, 343–348.
[32] David E Rumelhart. 2017. Schemata: The building blocks of cognition. In *Theoretical issues in reading comprehension*. Routledge, 33–58.
[33] Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. 2020. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *Proceedings of the 13th international conference on web search and data mining*. 510–518.
[34] Cui Shang, Runtong Zhang, and Xiaomin Zhu. 2023. The influence of social embedding on belief system and its application in online public opinion guidance. *Physica A: Statistical Mechanics and its Applications* 623 (2023), 128875.
[35] Massimo Stella. 2022. Cognitive network science for understanding online social cognitions: A brief review. *Topics in Cognitive Science* 14, 1 (2022), 143–162.
[36] Haoteng Tang, Guixiang Ma, Lifang He, Heng Huang, and Liang Zhan. 2021. Commpool: An interpretable graph pooling framework for hierarchical graph representation learning. *Neural Networks* 143 (2021), 669–677.
[37] Xiaoqi Wang and Han-Wei Shen. 2022. Gnninterpreter: A probabilistic generative model-level explanation for graph neural networks. *arXiv preprint arXiv:2209.07924* (2022).
[38] Yifei Wang, Qi Zhang, Yaoyu Guo, and Yisen Wang. 2024. Non-negative Contrastive Learning. *arXiv preprint arXiv:2403.12459* (2024).
[39] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
[40] Zhiping Xiao, Weiping Song, Haoyan Xu, Zhicheng Ren, and Yizhou Sun. 2020. TIMME: Twitter ideology-detection via multi-task multi-relational embedding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2258–2268.
[41] Chao Xu, Jinyang Li, Dachun Sun, Jinning Li, Tarek Abdelzaher, Jesse Graham, Michael Macy, Christian Lebiere, and Boleslaw Szymanski. 2023. The Paradox of Information Access: On Modeling Polarization in the Age of Information. *IEEE Transactions on Control of Network Systems* (2023).
[42] Xiwang Yang, Yang Guo, Yong Liu, and Harald Steck. 2014. A survey of collaborative filtering based social recommender systems. *Computer communications* 41 (2014), 1–10.
[43] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems* 31 (2018).
[44] Jingying Zeng, Richard Huang, Waleed Malik, Langxuan Yin, Bojan Babic, Danny Shacham, Xiao Yan, Jaewon Yang, and Qi He. 2024. Large language models for social networks: Applications, challenges, and solutions. *arXiv preprint arXiv:2401.02575* (2024).
[45] Jingying Zeng, Hui Liu, Zhenwei Dai, Xianfeng Tang, Chen Luo, Samarth Varshney, Zhen Li, and Qi He. 2025. Cite before you speak: Enhancing context-response grounding in e-commerce conversational llm-agents. *arXiv preprint arXiv:2503.04830* (2025).
[46] Xinyang Zhang, Yury Malkov, Omar Florez, Serim Park, Brian McWilliams, Jiawei Han, and Ahmed El-Kishky. 2022. TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations at Twitter. *arXiv preprint arXiv:2209.07562* (2022).
[47] Arman Zharmagambetov, Suryabhan Singh Hada, Magzhan Gabidolla, and Miguel A Carreira-Perpinán. 2021. Non-greedy algorithms for decision tree optimization: An experimental comparison. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[48] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. SGLang: Efficient Execution of Structured Language Model Programs. arXiv:2312.07104 [cs.AI] https://arxiv.org/abs/2312.07104

A IGAT Algorithms

This appendix presents the pseudocode for the IGAT Tree algorithms. Algorithm 1 details the initialization of IGAT, Algorithm 2 describes the joint optimization process, and Algorithm 3 outlines the model inference procedure.

B Experimental Settings

B.1 Datasets

We evaluate our model and baselines on three real-world Twitter datasets-**Philippines** [22], **Election** [23], and **Russia-Ukraine War** [24]. each containing tweets, associated users, and retweets that serve as user interaction records.

B.1.1 Philippines Dataset. The Philippines Dataset [22] includes 41,017 tweets with 9,140 users. This dataset contains tweets collected from *01/01/2023* to *06/28/2023*, covering discussions about the expansion of the Enhanced Defense Cooperation Agreement (EDCA) in 2023, which is a military agreement between the Philippines and the United States. Some users belief EDCA can facilitate national security and military modernization while others oppose it

Algorithm 1 IGAT Tree Initialization

procedure InitializeTree($\boldsymbol{A}$, $\boldsymbol{S}$, V, P)
  $\mathcal{N}_{in} \leftarrow \emptyset$, $\mathcal{N}_{lf} \leftarrow \{(\boldsymbol{A}, \boldsymbol{S}, V, P, \boldsymbol{\Phi})\}$ ▷ Initialize Node Sets
  $Flag \leftarrow$ **False** ▷ Termination Flag
  while $Flag$ = **False do**
    $Flag \leftarrow$ **True**
    for each $Node = (\boldsymbol{A}_i, \boldsymbol{S}_i, V_i, P_i, \boldsymbol{\Phi}_i) \in \mathcal{N}_{lf}$ **do**
      Optimize $\boldsymbol{\Phi}_i$ using Equation (4) and (8)
      Compute $\bar{Z}$ using Equation (5)
      Select the best $\boldsymbol{\Lambda} = g(\boldsymbol{A}_i, \boldsymbol{S}_i)$ using Equation (9)
      if Termin. condition in Section 4.2.3 is not met **then**
        $Flag \leftarrow$ **False**
        $\mathcal{N}_{in} \leftarrow \mathcal{N}_{in} \cup \{Node\}$
        $\mathcal{N}_{lf} \leftarrow \mathcal{N}_{lf} \setminus \{Node\}$
        Compute $\widetilde{Z}$ using Equation (7)
        for $c \leftarrow 1$ to d **do** ▷ Generate Child Nodes
          Create $(\boldsymbol{A}_c, \boldsymbol{S}_c, V_c, P_c, \boldsymbol{\Phi}_c)$ using Equation (6)
          $\mathcal{N}_{lf} \leftarrow \mathcal{N}_{lf} \cup \{(\boldsymbol{A}_c, \boldsymbol{S}_c, V_c, P_c, \boldsymbol{\Phi}_c)\}$
        end for
      end if
    end for
  end while
  return $\mathcal{N}_{in}$, $\mathcal{N}_{lf}$
end procedure

Algorithm 2 IGAT Tree Optimization

procedure OptimizeTree($\mathcal{N}_{in}$, $\mathcal{N}_{lf}$)
  for each $Node = (\boldsymbol{A}_l, \boldsymbol{S}_l, V_l, P_l, \boldsymbol{\Phi}_l) \in \mathcal{N}_{lf}$ **do**
    $\boldsymbol{\Phi}_l \leftarrow \boldsymbol{\Psi}_l$ ▷ Initialize Belief Embedding Model
  end for
  while Not Converged **do**
    for each $Node = (\boldsymbol{A}_i, \boldsymbol{S}_i, V_i, P_i, \boldsymbol{\Phi}_i) \in \mathcal{N}_{in}$ **do**
      Prune $\boldsymbol{W}^L$ in $\boldsymbol{\Phi}_i$ using Equation (12)
      Compute $\mathcal{L}_{rec}(\boldsymbol{\Phi}_i, \boldsymbol{\Lambda}_i)$ and $\mathcal{L}_{e,i}$ ▷ Forward Pass
    end for
    for each $Node = (\boldsymbol{A}_l, \boldsymbol{S}_l, V_l, P_l, \boldsymbol{\Psi}_l) \in \mathcal{N}_{lf}$ **do**
      Prune $\boldsymbol{W}^L$ in $\boldsymbol{\Psi}_l$ using Equation (12)
      Compute $\mathcal{L}_{rec}(\boldsymbol{\Psi}_l, \boldsymbol{\Lambda}_l)$ ▷ Forward Pass
    end for
    Compute total loss $\mathcal{L}$ using Equation (11)
    for each $Node = (\boldsymbol{A}_i, \boldsymbol{S}_i, V_i, P_i, \boldsymbol{\Phi}_i) \in \mathcal{N}_{in}$ **do**
      Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Phi}_i}$ ▷ Back Propagation
      Update $\boldsymbol{\Phi}_i \leftarrow \boldsymbol{\Phi}_i - \lambda \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Phi}_i}$
      Compute $\bar{Z}$ and $\widetilde{Z}$ ▷ Update Node Split
      Update $(\boldsymbol{A}_c, \boldsymbol{S}_c, V_c, P_c)$ for all children of $Node$
    end for
    for each $Node = (\boldsymbol{A}_l, \boldsymbol{S}_l, V_l, P_l, \boldsymbol{\Psi}_l) \in \mathcal{N}_{lf}$ **do**
      Compute $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Psi}_l}$ ▷ Back Propagation
      Update $\boldsymbol{\Psi}_l \leftarrow \boldsymbol{\Psi}_l - \lambda \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Psi}_l}$
    end for
  end while
  return $\mathcal{N}_{in}$, $\mathcal{N}_{lf}$
end procedure

due to concerns over sovereignty. The average connectivity of users and tweets in the Philippines dataset is 6.92, indicating intermediate graph sparsity.

B.1.2 Election Dataset. The Election dataset [23] includes 7,031 tweets and 502 users. It comprises tweets collected in *2020*, focusing on discussions about the election in the United States between conservative and liberal parties. This dataset is relatively small and sparse, with an average connectivity of 2.95, making it useful for testing the robustness of belief learning models.

B.1.3 Russia-Ukraine War Dataset. The Russia-Ukraine War dataset contains 135,135 tweets and 6,505 users collected between *05/01/2022* and *08/08/2022*, focusing on the escalation of the Russia-Ukraine war in February 2022. The non-English tweets have been translated into English using the Google Translate API. This dataset is relatively large and dense, with 22.46 average entity connectivity.

Algorithm 3 IGAT Tree Inference

procedure Inference($\mathcal{N}_{in}$, $\mathcal{N}_{lf}$)
  $Emb \leftarrow \emptyset$, $Ent \leftarrow \emptyset$ ▷ Initialize embedding and entity sets
  $Prob \leftarrow \emptyset$ ▷ Initialize assignment probability set
  for each $Node = (\boldsymbol{A}_l, \boldsymbol{S}_l, V_l, P_l, \boldsymbol{\Psi}_l) \in \mathcal{N}_{lf}$ **do**
    Compute belief embedding $\boldsymbol{Z}_l$ using Equation (2)
    $Emb \leftarrow Emb \cup \{\boldsymbol{Z}_l\}$, $Ent \leftarrow Ent \cup \{V_l\}$
    $Prob \leftarrow Prob \cup \{P_l\}$
  end for
  Compute IHR using $\boldsymbol{Z}_l \in Emb$, $V_l \in Ent$, $P_l \in Prob$, and with Equation (10)
  return IHR
end procedure

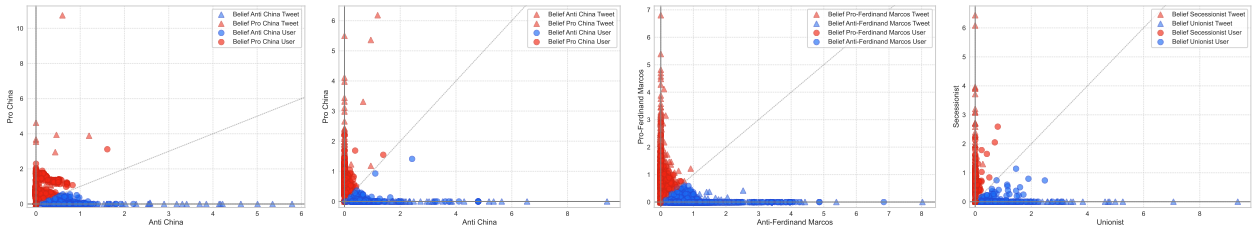
B.2 Environments

All experiments are conducted on a GPU machine with 4 Nvidia A5000 GPUs, a 64-core CPU, and 512 GiB of memory. The implementation of IGAT is based on Python 3.12.8 and PyTorch 2.6.0. Where applicable, experiments are repeated 10 times to report average results and standard deviations.

B.3 Evaluation Metrics

For each dataset, we invited candidates to manually annotate 20% of the tweets for the most representative subjects and beliefs, using majority vote as the final label. We apply accuracy, weighted F1 score, and purity score for evaluating the belief discovery task.

As described in Section 3.1, social belief discovery is a multi-label hierarchical clustering process, where each user or message may belong to one or more hierarchical clusters. We adopt a stricter evaluation policy that requires the predicted hierarchy (including all ancestor labels) to fully match the ground truth, while the order of ancestor labels does not affect correctness. This enables a fair comparison with non-hierarchical models (e.g., LLM decoders, LLM encoders with KMeans, VGAEs) by flattening the hierarchical ground truth. We focus on message-level predictions, as user-level ground-truth annotations are also aggregated from messages.



(a) Subject: Energy China **(b) Subject: Labor Migration** **(c) Subject: Social Economics** **(d) Subject: Insurgent Threats**

Figure 8: Additional visualizations of belief embeddings learned at the leaf nodes of the IGAT model. Different beliefs exhibit clear separation within each detected subject. Each belief is mapped to an individual axis, while the dotted diagonal line represents the overall trend of belief orientation.

B.4 Hyper-Parameters

In the numerical evaluation in Section 5, we set the tree depth δ to 2. In the qualitative analysis, we use a larger maximum depth of 4, incorporating additional entity metadata and features such as *age*, *language*, and *location*. We set the dimension of belief embeddings as $\eta = 2$. To compute the topic-feature similarity matrix, we use the pre-trained political topic encoder ConfliBERT[1]. The learning rate for IGAT is 0.1, with a weight decay of 1×10^{-4}. We train the Slice Mechanism for 300 warm-up iterations before jointly training it with the belief embedding model on the leaf nodes; the joint training then continues for an additional 600 iterations.

B.5 Baselines

- **VGAE**: As VGAE [20] does not directly support end-to-end hierarchical structure learning, we first apply spectral clustering on the feature similarity matrix to slice data and then apply VGAE to each subset of different subjects.
- **InfoVGAE**: Similar to VGAE, we apply Spectral Clustering on the feature similarity matrix and then use InfoVGAE [23] for each subset and subgraph.
- **DIFFPOOL**: We adapt DIFFPOOL [43] by removing the original graph-level classification objective but retaining the link prediction objective. We then use the assignment vectors to slice the data and assign entities to subjects.
- **Ginie**: Ginie is a non-parametric algorithm for hierarchical clustering of entities. We use the *genieclust* library[2] for implementation.
- **RoBERTa-KM**: We encode messages with a fine-tuned RoBERTa model (*Semi-RoBERTa*) and cluster the resulting embeddings into K groups via K-Means. Each cluster's label is assigned by majority vote on a labeled subset for classification evaluation.
- **T-RoBERTa-KM**: We follow the same procedure with a fine-tuned Twitter-RoBERTa model, clustering embeddings into K groups and labeling each cluster by majority vote.
- **TwhinBERT-KM**: Similarly, we replace the encoder with a fine-tuned TwhinBERT model, apply KMeans to the embeddings, and use majority voting for the final labeling.
- **GPT-3.5**: We use the OpenAI GPT-3.5 [7] API in a few-shot setup. The instructions and examples for all k topic-belief categories are included in the system prompt, while the messages to be classified are provided in the user prompt. The model then selects the best category among the k options.
- **GPT-4o**: We apply the same few-shot procedure to OpenAI's GPT-4o [18] API with the identical prompt.
- **DeepSeek-V3**: We conduct the experiment with the official API of DeepSeek V3 model [10] using an identical prompt and few-shot procedure as above.
- **Llama 3.3**: We host the Llama 3.3 model locally using the `sglang` [48] library and employ the same few-shot prompt structure.
- **TIMIE**: This is the unsupervised version of TIMME [40] without the semi-supervised entity classification objective.
- **Semi-RoBERTa**: We fine-tune the RoBERTa [25] model for multi-class belief classification, using the same proportion of GPT-labeled data as our IGAT model.
- **Semi-T-RoBERTa**: We fine-tune the Twitter-RoBERTa [25] model for multi-class belief classification, again using the same proportion of GPT-labeled data.
- **Semi-TwhinBERT**: Similarly, we fine-tune the TwhinBERT model [46] for multi-class belief classification with the same proportion of GPT-labeled data.
- **Semi-TIMMIE**: The semi-supervised version of TIMME [40], trained jointly on the entity classification task and graph link prediction. We use text embeddings from ConfliBERT as node features.
- **Semi-SGVGAE**: SGVGAE [22] utilizes LLM-generated annotations to semi-supervise disentangled graph representation learning via axis-guidance and connectivity enhancement.

C Case Study of Belief Embeddings

In Figure 6, we provide additional visualizations of the belief embeddings learned at the leaf nodes of the IGAT model. Different beliefs show a clear separation within each detected subject. In the figure, each belief is mapped to its own axis, while the dotted diagonal line indicates the overall trend of belief orientation. From this visualization, we can draw several insights. For example, belief polarization in the *Insurgent Threats* subject is the most significant, and beliefs under the *Energy China* subject lean notably towards the horizontal axis.

[1] https://huggingface.co/eventdata-utd/conflibert-insight-crime-multilabel
[2] https://github.com/gagolews/genieclust