Human-Centric Community Detection in Hybrid Metaverse Networks with Integrated AI Entities

Anonymous Author(s)

ABSTRACT

Community detection is a cornerstone problem in social network analysis (SNA), aimed at identifying cohesive communities with minimal external links. However, the rise of generative AI and the Metaverse introduces new complexities by creating hybrid communities of human users and AI entities. Traditional community detection approaches that overlook the interwoven presence of humans and AIs are inadequate for managing such hybrid networks, known as human-AI social networks (denoted by HASNs), especially when prioritizing human-centric communities. This paper introduces a novel community detection problem in HASNs (denoted by MetaCD), which seeks to enhance human connectivity within communities while reducing the presence of AI nodes. Effective processing of MetaCD poses challenges due to the delicate trade-off between excluding AI nodes and maintaining community structure. To address this, we propose CUSA, an innovative framework incorporating AI-aware clustering techniques that navigate this trade-off by selectively retaining AI nodes that contribute to community integrity. Furthermore, given the scarcity of real-world HASNs, we design four strategies for synthesizing these networks under various hypothetical scenarios. Empirical evaluations on real social networks, reconfigured as HASNs, demonstrate the effectiveness and practicality of our approach compared to traditional non-deep learning and graph neural network (GNN)-based methods.

KEYWORDS

community detection, human-centric, social networks, generative AI, Metaverse

ACM Reference Format:

Anonymous Author(s). 2018. Human-Centric Community Detection in Hybrid Metaverse Networks with Integrated AI Entities. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Community detection is a fundamental problem in social network analysis (SNA), focusing on identifying tightly-knit communities with minimal external connections [1][2][3]. This task is critical for analyzing social relationships and holds practical applications in marketing and personalized services, where insights into community structures enable targeted strategies and enhance user engagement [4][5].

WOODSTOCK'18, June 03–05, 2018, Woodstock, NY, USA

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

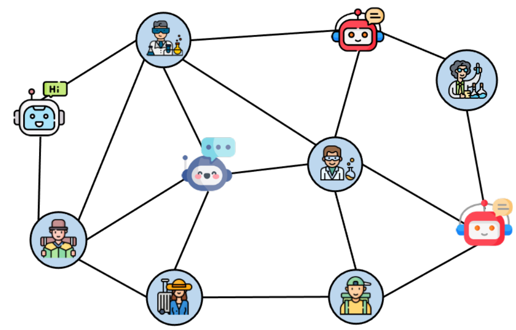
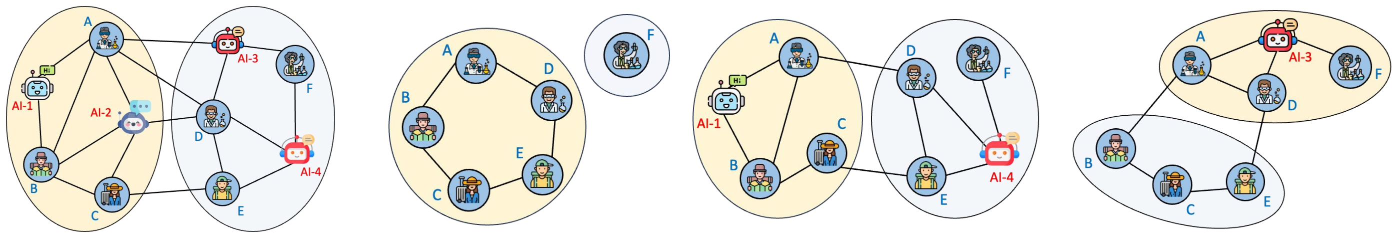


Figure 1: A schematic depiction of HASN.

The integration of generative AI and Metaverse is transforming traditional social networks, creating human-AI social networks (HASNs) (Figure 1) that blend human and AI entities within shared digital spaces. In Metaverse environments, which combine virtual reality (VR) and mixed reality (MR) components, users engage with both humans and AI-driven entities, such as avatars and virtual assistants, forming communities that span the physical and digital worlds. For example, social Metaverse platforms like Microsoft AltspaceVR [6] and Meta Horizon Workrooms [7] enable immersive, large-scale virtual gatherings, allowing users to interact with AI alongside human participants through customizable 3D environments, spatial audio, and interactive tools.

In these evolving HASNs, a fundamental shift occurs: traditional social clusters now include AI nodes, making human-centric community detection crucial. Unlike conventional networks, HASNs often aim to prioritize human interactions while accommodating AI entities that provide social support or assistance. For instance, platforms such as Nomi AI [8] create AI companions for users, enhancing social engagement and addressing issues like loneliness. Similarly, platforms like Engage [9] and Mozilla Hubs [10] host virtual events and social gatherings, integrating AI to foster rich, interactive experiences. These hybrid networks demand new community detection methods that discover human-centric communities by selectively retaining AI nodes that strengthen community cohesion and removing those that do not. This approach supports applications focused on authentic human engagement, including marketing, user recommendations, and digital companionship.

However, applying traditional community detection methods to these hybrid networks presents several challenges. For instance, (1) Applying approaches that overlook AI entities may result in communities with an excess of AIs, which are less effective for human-centric applications such as advertising and recommendation, as these efforts are not relevant for AI entities. (2) Alternatively,



(a) AI-complete clustering (b) AI-blind clustering (c) AI-anomaly clustering (d) AI-aware clustering

Figure 2: An illustrative example of community detection in an HASN.

one might consider removing all AI nodes before conducting community detection. However, this could disrupt connections between AI and human nodes, as well as among AI nodes themselves. Since community detection relies heavily on graph topology, these disruptions may lead to inaccurate community assignments for human users. (3) A third approach could involve using anomaly detection techniques to classify certain AI nodes as anomalies and remove them before clustering. However, AI node behavior often diverges from conventional "anomalies" in traditional social networks [11]. Unlike typical outliers, AI nodes may mimic human behavior, support human interactions, and integrate into communities, making this strategy likely to yield unsatisfactory results.

In this work, we introduce a novel community detection problem tailored to scenarios where humans and numerous AI entities are intertwined within social networks, with a primary focus on human closeness based on graph structure. [1] We envision an Metaverse environment where a social network comprises both human users and AI entities, forming what we term the Human-AI Social Network (HASN), as illustrated in Figure 1. Accordingly, we envisage four scenarios of HASNs existing in Metaverse [12][13]: (1) Random Interaction, (2) Introverted Humans Prioritized for Interaction, (3) Distinct AI Types, and (4) Dual-personality AIs. These proposed scenarios are grounded in the versatility of AI, allowing it to interact widely and adaptively with diverse users [14][15]. (Details are provided in the experimental setup, Section 5.1.)

This work focuses on community detection in HASNs, especially when the primary focus is on human users (human-centric communities). Ideally, clustering should produce clusters with high human closeness and minimal AI presence, balancing the removal of AI nodes while maintaining community integrity. Let us explore some illustrative examples as follows. Figure 2 shows several approaches to cluster an HASN depending on how they consider the AI nodes: (1) *AI-complete clustering*: Figure 2 (a) shows a clustering result from a method that ignores AI nodes and performs clustering directly. However, as observed, each community contains an excessive number of AI entities, making it impractical for human-centric applications such as advertising and recommendation. (2) *AI-blind clustering*: Figure 2 (b) shows a clustering result from a method that treats all AI nodes as outliers and removes them before clustering. Human nodes D and E are initially connected to node F through AI nodes AI-3 and AI-4, respectively, suggesting that D, E, and F may belong to the same community. However, this approach can disrupt connections between AI and human nodes, as well as among AI nodes, resulting in the loss of important links and leading to unsatisfactory community results. (3) *AI-anomaly clustering*: Figure 2 (c) shows a clustering result where certain AI nodes are identified as anomalies through anomaly detection techniques and removed before clustering. However, unlike typical anomalies in traditional social networks, which often link different communities or form dense connections [11][16], AI nodes may mimic human behavior, support human interactions, and integrate into communities. Consequently, as shown, traditional anomaly detection would remove AI-2 and AI-3, resulting in inaccurate community outcomes for human users (e.g., A and E). (4) *AI-aware clustering*: Figure 2 (d) illustrates clustering results achieved by identifying AI nodes that serve as bridges between humans or connect key human nodes. For instance, AI-3 is an AI node retained during the clustering process because it effectively links humans A, D, and F. This connectivity does not occur in the aforementioned three methods (i.e., Figure 2 (a), (b), and (c)). In other words, this approach preserves AI nodes that positively impact the community while removing those that do not, thereby enhancing human closeness and fostering potential human-centric communities.

Considering the factors depicted above face the following challenges: (i) *Evaluation of AI nodes*: The behavior patterns of AI nodes may not resemble the typical "anomalies" found in traditional social networks [11]. AI may mimic human behavior, assist humans, and integrate into communities. In other words, some AI nodes are helpful for the formation of potential communities, while some are redundant. Accordingly, traditional anomaly detection methods are inadequate for evaluating AI nodes in emerging HASN graphs. (ii) *Tradeoff between AI removal and community integrity*: Removing AI nodes could lead to disconnections between AI and humans, as well as among AIs themselves. Therefore, it is crucial to identify and preserve AI nodes that enhance human closeness while removing those that do not contribute positively. For instance, some AI nodes act as bridges, enhancing human connections and facilitating the formation of potential communities. (iii) *Avoidance of local optima in AI-aware clustering*: The search space for finding the optimal combination of AI preservation/removal expands exponentially as

[1] To the best of our knowledge, this is the first study to address community detection in this hybrid human-AI scenario. While we concentrate on structural information, our approach is designed to be compatible with future integration of semantic features, which could further refine human-centric clustering by incorporating shared interests or interaction patterns.

the number of AI increases. This complexity makes this problem hard to solve and may result in solutions that settle at the local optima.

In this paper, we formulate a new problem, termed ***human-centric community detection in hybrid network (HASN) of Metaverse*** (denoted by MetaCD). Unlike previous studies focusing on community detection in purely human networks without considering AI involvement [2][3], our approach explores community detection within hybrid networks, such as HASNs (illustrated in Figure 1), composed of both human users and AI entities. Given an HASN with prior knowledge of which nodes in the network are AI nodes, the goal is to generate clusters that can *maximize human closeness with minimal AI involvement*, by balancing the removal of AI nodes and maintaining community integrity. Specifically, a desirable clustering result of an HASN should achieve two key objectives simultaneously: (1) maximizing human closeness and (2) minimizing the presence of AI nodes within each cluster.

To this end, we design a novel algorithm called *Customized AI-aware Simulated Annealing for Clustering* (denoted by CUSA), to tackle the above challenges of MetaCD. **(1)** For the first challenge in the evaluation of AI nodes, CUSA incorporates *AI Scoring* to evaluate the "*humanoid*" of an AI node in HASN. Specifically, if an AI node acts to a greater extent as a bridge between humans or connects important human nodes, its *humanoid* is higher, and vice versa. This is because it can enhance human closeness and potentially lead to the formation of new human communities or the migration of humans to different communities due to the influence of such AI nodes. **(2)** For the second challenge in the tradeoff between AI removal and community integrity, we develop the *AI-aware Louvain clustering algorithm* into CUSA. This algorithm adaptively groups nodes based on human closeness gain while also taking into account the proportion of AI presence in each community during clustering. **(3)** For the third challenge in avoidance of local optima in AI-aware clustering, CUSA infuses *AI-aware adaptive clustering (3AC) framework with a probability-based escape strategy*. This framework adaptively partitions an HASN by removing the AI node with the lowest *humanoid*. In addition to removing (or preserving) AI nodes based on their *humanoid*, we employ a predefined probability distribution to escape local optima and strive for a global optima. We evaluate the performance of CUSA on benchmark real-world social networks (i.e., Cora, CiteSeer, and PubMed) transformed into HASNs (Figure 1) using our proposed generation strategies. The contributions of this work include:

- To the best of our knowledge, MetaCD is the first attempt to study the community detection problem under the new scenarios where humans and numerous AI entities are intertwined in social networks (denoted by HASNs), especially focusing on human-centric communities. In addition, we propose four HASN scenarios, each with specifically designed generation strategies.
- To effectively address MetaCD, we develop a novel algorithm called CUSA, incorporating AI-aware clustering techniques that navigate the delicate trade-off between removing AI nodes and maintaining community structure by selectively retaining AI nodes that contribute to community integrity.
- Empirical evaluations on real social networks (reconfigured as HASNs) demonstrate the effectiveness of CUSA in forming human-centric communities compared to competitive baselines.
- Our experiments show that a carefully designed generation strategy significantly improves clustering results, enhancing human closeness and revealing potential communities.

2 RELATED WORK

Community Detection. Community detection is the process of grouping nodes into densely connected communities with sparse interconnections, depending on the structure of the graph. Early traditional non-deep learning methods, such as the spectral clustering algorithm [17], optimize ratio and normalized-cut criteria. Louvain [18] is a well-known optimization algorithm that uses a node-moving strategy to optimize modularity. Extensions of greedy optimizations include simulated annealing [19], extremal optimization [20], and spectral optimization [21]. Recently, with the rise of various deep learning-based modeling techniques [3][22] developed for community detection, graph neural network (GNN)-based methods have emerged as the latest trend in this field. These approaches work by learning lower-dimensional vectors from the high-dimensional data of complex structural relationships [23] [24]. Furthermore, diverse information from nodes [25], edges [26], neighborhoods [27], or multigraphs [28] can be jointly recognized with special attention in the deep learning process. Additionally, deep learning explores community structures in complex real-world scenarios, such as large-scale [29], high-sparse [30], complex structural [31], and dynamic networks [32], effectively handling intricate relational data. Despite the extensive exploration of various community detection methods and their application in different real-world scenarios, these existing approaches generally operate under the common assumption that social networks are solely composed of human users. In this work, we explore a new problem of community detection formulated for scenarios where humans and numerous AI entities are intertwined within social networks. Accordingly, we propose a clustering method that prioritizes human closeness while addressing the challenges of intertwined AI entities in social networks.

Graph Anomaly Detection. Graph Anomaly Detection (GAD) identifies unusual patterns, outliers, or unexpected behaviors in graph-structured data [11][16]. GAD techniques have proven effective in various applications, including computer network intrusion detection [33], fraud detection [34], and anomaly detection in social networks [35][36]. However, this work addresses a novel issue by considering interactions between humans and AI within a new social network. Unlike traditional social networks, where "anomalies" are identified based on unusual connection patterns such as bridging different communities or forming dense links with other nodes [11], AI behavior patterns do not inherently appear abnormal. AI can mimic human behavior, assist humans, and seamlessly integrate into communities. Therefore, this distinction renders existing GAD methods unsuitable for handling this new problem.

Generative Artificial Intelligence. Generative Artificial Intelligence (GAI) is a form of AI capable of autonomously generating new content, including text, images, audio, and video. The current mainstream approach to realizing GAI involves training large language models (LLMs) [15]. Various applications of LLMs have rapidly emerged, including ChatGPT [37], Gemini [38], and Claude [39]. These LLM applications have significantly transformed our lives by adding convenience in areas such as file summarization, code generation, learning assistance, providing inspiration, and even offering life advice and psychological counseling. Building on these observations, we envision a future where social networks are seamlessly integrated with both humans and AIs. We aim to perform community detection within this hybrid network. As far as we are concerned, this concept has not been explored in previous studies, making our work a pioneering effort in this field.

3 PROBLEM FORMULATION

An HASN graph is denoted as $G(V, E)$, where $\forall v \in V$ is a set of vertices comprising the sets H (human users) and AI (AI entities), such that $|V| = |H| + |AI|$, and $\forall e \in E$ represents the set of edges between humans, AIs, and human-AI connections.

The MetaCD clustering problem aims to partition an HASN graph into K disjoint subgraphs $C_i(V_i, E_i)$, where $\bigcup_{i=1}^{K} V_i \subseteq V$ (since AI nodes and their connected edges may be removed during the clustering process) and $V_i \bigcap V_j = \emptyset$, with prior knowledge of which nodes in the network are AI nodes. The goal of MetaCD is to discover a set of clusters (subgraphs) $P = \{C_i\}_1^K = \{C_1, C_2, \ldots, C_K\}$ that can maximize human closeness with minimal AI presence. Concretely, a desirable clustering result of an HASN should achieve two key objectives simultaneously: (1) maximizing human closeness and (2) minimizing the presence of AI nodes for each cluster.

3.1 Objective Function of MetaCD

To achieve the goal of MetaCD, we employ a modularity function introduced in a seminal paper by Newman as our objective function [40]:

$$Q(P = \{C_i\}_{i=1}^{K}) = \frac{1}{2|E|} \left(\sum_{i=1}^{K} \sum_{v_p, v_q \in C_i} \left(A_{pq} - \frac{d_p d_q}{2|E|} \right) \right)$$

Modularity Q measures clustering quality in networks by comparing the density within clusters to the density between clusters. It ranges from -1 to 1, with higher scores indicating better clustering. Here, A is the adjacency matrix, A_{pq} indicates the presence of a connection between nodes p and q, and d_p is the degree of node p.

To encourage the clustering algorithm to generate cohesive communities with minimal AI presence, we modify the vanilla modularity by infusing a reward-penalty function. This function reweights the clustering quality based on the ratio of humans (and AIs) presence in each cluster C_i, defined by:

$$W(C_i) = \beta \cdot \frac{\sum_{v \in C_i} L_v}{|C_i|} - \gamma \cdot \frac{\sum_{v \in C_i} (1 - L_v)}{|C_i|}$$

where

$$L_v = \begin{cases} 1, & \text{if node } v \in H \\ 0, & \text{if node } v \in AI \end{cases}$$

This leads to a human-centric modularity HQ:

$$HQ(P) = \frac{1}{2|E|} \left(\sum_{i=1}^{K} \alpha \cdot W(C_i) \cdot \left(\sum_{v_p, v_q \in C_i} \left(A_{pq} - \frac{d_p d_q}{2|E|} \right) \right) \right)$$

Note that β is the weight for rewarding human nodes, γ is the weight for penalizing AI nodes, and α is the weight for adjusting the emphasis on human nodes in the objective function [2]. Accordingly, the purpose of MetaCD is to discover a set of clusters (subgraphs) $P = \{C_i\}_1^K$ that maximizes HQ:

$$P^* = \arg \max_{\{C_i\}_{i=1}^{k}} HQ(\{C_i\}_{i=1}^{K})$$

This objective function promotes the generation of tight-knit communities with minimal AI presence. Since certain AI entities can aid in the formation of these communities, it is crucial to identify and preserve AI nodes that can promote human closeness while removing those that can not.

4 ALGORITHM DESIGN FOR METACD

To effectively address MetaCD, we develop a novel algorithm, named *Customized AI-aware Simulated Annealing* (denoted by CUSA), which integrates several key components.

(i) To tackle the challenge in the evaluation of AI nodes, CUSA incorporates an ***AI Scoring*** mechanism to assess the *humanoid* score of each AI node in a Human-AI Social Network (HASN). CUSA achieves this by assigning distinct weights to edges based on neighboring relationships. Specifically, weights are assigned as follows: human-human edges are denoted by hh, human-AI edges by ha, and AI-AI edges by aa, with the relationship $hh > ha > aa$ (defaults set to 3, 2, and 1, respectively). CUSA then applies three representative approaches commonly used in social networks—eigenvector centrality, betweenness centrality, and clustering coefficient—to evaluate node scores. This encourages the identification and preservation of AI nodes that play significant roles in human-centric communities, aligning with MetaCD's objectives. Each approach serves a specific purpose:

(1) Eigenvector Centrality (EC) [41] considers a node important if it connects to other highly influential nodes. We apply this measure, assuming that AI nodes in the HASN are linked to prominent individuals (or vice versa). Thus, prioritizing AI nodes with high eigenvector centrality in clustering is vital for shaping cohesive community structures and enhancing human closeness.

(2) Betweenness Centrality (BC) [42] identifies a node as crucial if it serves as a bridge or intermediary within the network. This measure aligns with the objectives of the MetaCD problem, as a beneficial AI node often connects disparate human nodes, potentially reshaping community boundaries and reinforcing human closeness. Therefore, we prioritize preserving AI nodes with high betweenness centrality during clustering [3].

[2] For simplicity, we set α, β, and γ to 1 in our experiments to observe the proposed algorithm's core behavior without the added complexity of multiple parameters.

[3] To mitigate the computational cost of the shortest-path calculations in BC, we utilize an optimized version of the BC evaluation method [42].

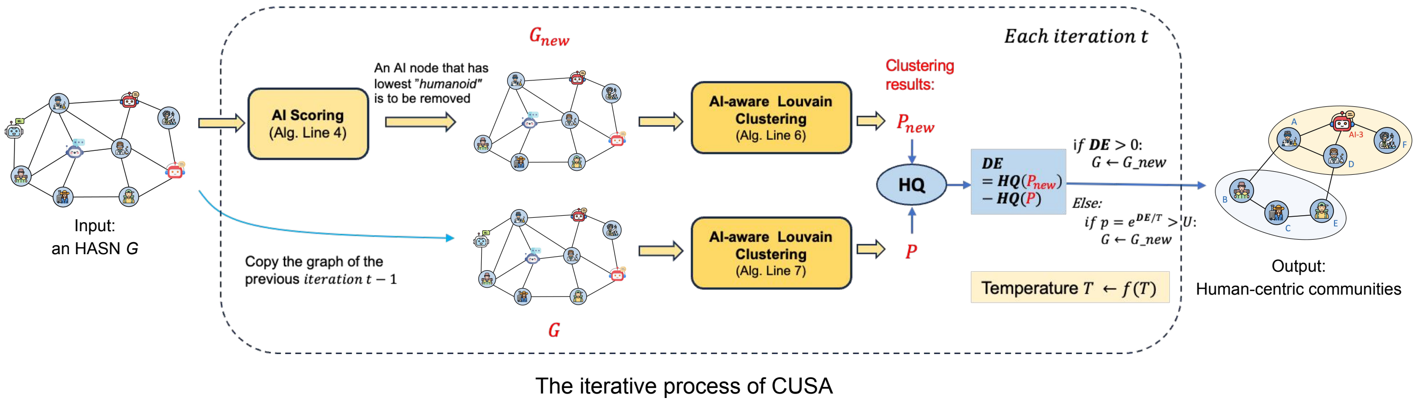


Figure 3: The framework of the proposed CUSA algorithm.

(3) Clustering Coefficient (CC) [43] regards a node as important if its neighbors are densely interconnected. However, in our approach, we prioritize retaining AI nodes with low clustering coefficient scores. This preference is based on the observation that a high clustering coefficient suggests ample interconnectivity within the subnetwork, thereby reducing the necessity for the AI node to act as a bridge among other nodes.

As shown in Table 2 in the experimental section, the combined use of EC, BC, and CC through a linear combination of their normalized scores yields the most effective results. This indicates that each measure (EC, BC, and CC) captures a distinct aspect of information regarding AI nodes, providing a more comprehensive perspective on their significance within the HASN.

(ii) For the tradeoff between AI removal and community integrity, we develop ***AI-aware Louvain Clustering Algorithm*** into CUSA. The Louvain clustering algorithm [18] optimizes modularity to detect community structures in networks. Specifically, each node is initially treated as an individual community. For each node i (or community C_i), we calculate the modularity gains ΔQ by moving node i to its neighboring communities C_j, is defined as:

$$\Delta Q(i \rightarrow C_j) = \left[\frac{\Sigma in + k_{i,in}}{2|E|} - \left(\frac{\Sigma tot + k_i}{2|E|}\right)^2\right] - \left[\frac{\Sigma in}{2|E|} - \left(\frac{\Sigma tot}{2|E|}\right)^2 - \left(\frac{k_i}{2|E|}\right)^2\right]$$

where Σin is the sum of the weights of the links inside the community C_j, Σtot is the sum of the weights of the links incident to nodes in the community C_j, k_i is the sum of the weights of the links incident to node i, and $k_{i,in}$ is the sum of the weights of the links from node i to nodes in the community C_j. At each iteration, communities are aggregated into new super-nodes, condensing into separate communities. This iterative process of evaluating node movements and merging communities continues until no further increase in modularity can be achieved.

To make the Louvain clustering algorithm more human-centric, we incorporate two reweighting terms into the modularity calculation, $W(C_j)^{before}$ and $W(C_j)^{after}$. These terms (defined in Section 3.1) are based on the human-to-AI ratio within the community C_j before and after merging node i (or community i), respectively. This approach makes the AI-aware Louvain Clustering Algorithm more human-specific by not only considering structural information in each merge step but also emphasizing the proportion of human members within each community. When node i (or community i) is merged into community C_j, these reweighting terms assess whether the merge increases the proportion of human members, leading to a human-centric modularity gain ΔHQ. This ensures that the resulting communities display a higher human proportion and cohesiveness, fostering a more human-centered clustering outcome. Mathematically, this is achieved with the following formula:

$$\Delta HQ(i \rightarrow C_j) = \theta \cdot W(C_j)^{after} \cdot \left[\frac{\Sigma in + k_{i,in}}{2|E|} - \left(\frac{\Sigma tot + k_i}{2|E|}\right)^2\right] - \theta \cdot W(C_j)^{before} \cdot \left[\frac{\Sigma in}{2|E|} - \left(\frac{\Sigma tot}{2|E|}\right)^2 - \left(\frac{k_i}{2|E|}\right)^2\right]$$

where θ is a scaling factor that can adjust the emphasis on presence of human nodes in the cluster (default set to 1). By iteratively applying this formula, the algorithm efficiently identifies a high-modularity partitioning of the network, ensuring that clusters contain a higher proportion of humans.

(iii) To prevent getting trapped in local optima during AI-aware clustering, CUSA infuses the ***AI-aware adaptive clustering (3AC) framework with a probability-based escape strategy***. This framework adaptively partitions an HASN by iteratively removing the AI node with the lowest *humanoid*, leveraging a probability-based escape strategy to enhance search flexibility. The 3AC framework is central to CUSA's process, and its steps are as follows:

(1) Evaluate *humanoid* for all AI nodes of the HASN graph and rank them.

Algorithm 1 The algorithm of CUSA

Input: An HASN $G = (V, E)$ where $|V| = |H| + |AI|$
Output: $P^* = \{C_i\}_1^K$, the K disjoint clusters, each achieving high human closeness with minimal AI presence

$T \leftarrow T_{initial}$
$G_{best} \leftarrow G$
while $(T > T_{min})$ **and** $|AI| > 0$ **do**
  $rnode \leftarrow \texttt{AIScoring}(G,\ AI)$
  $G_{new} \leftarrow \texttt{AIRemove}(G, rnode)$
  $P \leftarrow \texttt{Clustering}(G)$
  $P_{new} \leftarrow \texttt{Clustering}(G_{new})$
  $DE = HQ(P_{new}) - HQ(P)$
  if $DE > 0$ **then**
    $G \leftarrow G_{new}$
  else
    if $p = e^{DE/T} > U$ where $U \sim \text{Uniform}(0, 1)$ **then**
      $G \leftarrow G_{new}$
    end if
  end if
  if $HQ(P) \geq HQ(P_{best})$ **then**
    $G_{best} \leftarrow G$
    $P_{best} \leftarrow P$
    $T \leftarrow f(T)$
  end if
end while
return P_{best}
=0

(2) Apply the probability-based escape strategy to remove the AI node with the lowest *humanoid.*
(3) Cluster all remaining nodes to obtain partition P.
(4) Calculate the HQ for partition P.
(5) Re-evaluate *humanoid* for all remaining AI nodes and rank them again.
(6) Repeat 2-5 until the HQ of P converges to its highest value.

CUSA precisely implements the core steps of the 3AC framework, advancing through each stage to achieve the framework's objectives. Specifically, CUSA comprises three major components outlined in the 3AC framework: (1) an evaluation of AI nodes' importance (referred to as humanoid), informing the selective retention or removal of nodes based on their relevance to human-centric communities; (2) the AI-aware Louvain Clustering Algorithm, which performs clustering while accounting for the human-to-AI ratio within each community to ensure a human-centric structure; and (3) a probability-based escape strategy that dynamically adjusts the likelihood of removing nodes, helping the algorithm escape local optima and explore the solution space more effectively. Through these integrated components, CUSA iteratively adjusts the community structure by evaluating and balancing the human and AI nodes in each iteration. The pseudo-code for CUSA is presented in Algorithm 1, while the overall flowchart of the CUSA algorithm is illustrated in Figure 3 for further understanding.

Table 1: Basic statistics of benchmark datasets

Dataset	#Nodes	#Edges	Avg. #Degree
Cora [44]	2708	5429	4.01
CiteSeer [44]	3327	4732	2.84
PubMed [44]	19717	44338	4.50

5 EMPIRICAL EXPERIMENTS

5.1 Experimental Setup

Benchmark datasets. In the absence of real-world HASNs, we propose four strategies for generating such graphs based on the three widely-used datasets (i.e., Cora, CiteSeer, and PubMed). Table 1 shows some basic statistics of the three datasets for our experiments.

Initially, we generate a specific number of new AI nodes, such as $n\%$ of the total number of nodes in a real-world network dataset. Subsequently, we employ these generation strategies for inserting AI nodes into these networks, thereby constructing HASNs. Due to space limitations, we outline the four proposed generation strategies with the specific assumptions [8][12][13][15][45] in Appendix A.

After creating the HASNs using the proposed generation strategies, we anticipate the evolution of the social network over time, potentially resulting in the formation of new connections between individuals. To facilitate this evolution, we employ link prediction, a technique commonly used to identify potential connections between unconnected nodes in social networks. One classical method for link prediction is the Jaccard similarity index [46][47], which measures the similarity (or likelihood) between two nodes by calculating the ratio of the intersection to the union of their immediate neighbor nodes. Mathematically, Jaccard similarity is expressed as: $\sigma_{Jaccard}(x, y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$, where $N(x)$ indicates the number of neighbors of node x. By computing the Jaccard similarity values for all pairs of nodes, we can select the top n pairs with the highest values for connection, and iterate this process for r rounds.

Comparative Methods. We compare CUSA with four baseline methods: two traditional non-deep-learning community detection methods (Spectral and Louvain) and two GNN-based community detection methods (GCC and LSEnet).

(1) Spectral [17] is a clustering algorithm based on the normalized Laplacian matrix and the regularized adjacency matrix to minimize ratio cut.
(2) Louvain [18] detects communities by iteratively moving nodes to optimize modularity, maximizing links within communities and minimizing links between them.
(3) GCC [48] is a graph convolutional clustering method designed for efficient and scalable community detection.
(4) LSEnet [49] leverages structural embeddings for community detection, integrating node features via manifold-valued graph convolution in hyperbolic space.

Each method is equipped with three distinct approaches for handling AI nodes during clustering, namely, no removal of AIs (N), all

Table 2: The Q, HQ, AAM, and HMR results on the Cora dataset transformed into HASNs using the generation strategy (1), obtained by different AI node scoring methods used in CUSA, compared to no removal and all removal of AIs.

AI Scoring	Q (↑)	HQ (↑)	AAM (↑)	HMR (↑)
No removal (N)	0.7761	0.6966	4.24	42.25%
All removal (A)	0.8164	0.8164	0	0%
EC	0.8159	0.8092	25.22	20.34%
BC	0.8135	0.8051	27.32	27.96%
CC	0.8156	0.8040	**43.62**	**58.80**%
EC+BC	0.8151	0.8069	25.04	25.63%
EC+CC	0.8135	0.8036	22.58	31.26%
BC+CC	0.8113	0.8020	27.63	38.24%
EC+BC+CC	**0.8174**	**0.8190**	20.41	14.09%

removal of AIs (A), and removal of certain AIs by GAD method (D), respectively. (*cf.* Figure 2 (a)-(c))

Evaluation Metrics. For the problem of MetaCD (*cf.* Section 3.1), we employ widely used modularity Q [40] and the proposed human-centric modularity HQ as our evaluation metrics to assess the effectiveness of CUSA. The Q accesses clustering quality based on cluster closeness, while HQ does so with minimal AI involvement. Higher Q and HQ values indicate better quality of the detected communities. In addition to clustering quality, we aim to understand the impact of remaining AI on community structures after applying CUSA. We propose two metrics: Average AI-driven Migration (AAM) and Human Migration Ratio (HMR). AAM $= \frac{\#\text{human migration}}{\#\text{remaining AI}}$ measures the average number of humans migrating due to remaining AIs. HMR $= \frac{\#\text{human migration}}{\#\text{total human}}$ measures the proportion of humans moving from one community to another. Higher AAM and HMR values indicate that the preserved AIs have greater capabilities to influence human communities. Source codes are available at https://anonymous.4open.science/r/CUSA-DAFF/.

5.2 Experimental Results

5.2.1 Comparison of Different AI node Scoring Methods Used in CUSA. In the first set of experiments, we assess the performance of different AI scoring methods used in CUSA compared to no removal of AIs and all removal of AIs when clustering on the Cora dataset using the generation strategy (1). We test the three different scoring methods, eigenvector centrality (EC), betweenness centrality (BC), and clustering coefficient (CC), as well as their combination scores (denoted by "+" for the linear combination of normalized scores). The corresponding results are presented in Table 2. Inspection of Table 2 reveals three noteworthy points. First, all seven AI scoring methods (below the third line) used in CUSA outperform the standard Louvain clustering [18] with no removal of AIs and with all removal of AIs in terms of Q and HQ. Second, the EC+BC+CC stands out in performance, achieving a 5.3% and 17.6% relative gain in Q and HQ, respectively, compared to clustering with no AI removal. This suggests that EC, BC, and CC capture different aspects of information about AI nodes. Third, CC excels in terms of AAM and HMR, indicating that CC may serve as a bridge for potential communities. In the following experiments, we adopt EC+BC+CC which achieves the best Q and HQ results as our AI scoring method in CUSA (i.e., line 4 of Algorithm 1) for comparison with baselines [4].

Table 3: The HQ results on different datasets each transformed into HASNs using the generation strategy (1), obtained by CUSA in comparison to that of four clustering baselines each equipped with no removal of AIs (N), all removal of AIs (A), and removal of AIs by GAD (D).

Method		Cora	CiteSeer	PubMed
Spectral [17]	N	0.2907	0.2563	0.0245
	A	0.3721	0.2952	0.1394
	D	0.3308	0.2642	0.0553
Louvain [18]	N	0.6966	0.8047	0.6613
	A	0.8164	0.8962	0.7690
	D	0.7702	0.8438	0.7245
GCC [48]	N	0.6724	0.6353	0.6218
	A	0.7275	0.7110	0.6672
	D	0.6910	0.6762	0.6483
LSEnet [49]	N	0.6628	0.6289	0.6182
	A	0.7202	0.7089	0.6490
	D	0.6984	0.6720	0.6378
CUSA (ours)	-	**0.8190**	**0.8984**	**0.7716**

5.2.2 Comparison of CUSA with Baselines on Different Datasets. In the second set of experiments, we evaluate our CUSA against four baselines: two traditional non-deep learning-based methods Spectral [17] and Louvain [18] and two GNN-based methods GCC [48] and LSEnet [49]. These evaluations are conducted on various datasets (i.e., Cora, CiteSeer, and PubMed), each transformed into HASNs using the proposed generation strategy (1). Note again here that, we equip each baseline with three approaches for handling AI nodes during clustering: no removal of AIs (i.e., ignoring AI nodes and performing clustering directly), all removal of AIs (i.e., treating all AI nodes as outliers and removing them before clustering), and removal of certain AIs by GAD techniques (i.e., detecting certain AI nodes as anomalies using GAD and removed before clustering.) (*cf.* Figure 2. (a)-(c)). As to the GAD method, we adopt the recently proposed DCI method [50], which leverages self-supervised learning to decouple node representation learning from classification, to detect anomalous AI nodes in an HASN. Two important observations can be drawn from Table 3 which present the human-centric HQ score (modified from standard modularity Q in Section 3.1) of the resulting clusters. First, each clustering baseline achieves the highest HQ score when all AI nodes are removed from the HASN, followed by using GAD to remove AI nodes before clustering, while the performance is worst when no AI nodes are removed

[4] Due to the randomness in the HASNs synthesized using the proposed generation strategies, we carried out each experiment one hundred times and averaged the results to obtain convincing final results.

Table 4: The Q and HQ results on the Cora dataset transformed into HASNs using different generation strategies, obtained by CUSA in comparison to that of four clustering baselines each equipped with no removal of AIs (N), all removal of AIs (A), and removal of AIs by GAD (D).

Method		k% random insertion		Inverse degree insertion		Inner and outer AI mix		AI with dual personality	
		Q (↑)	HQ (↑)	Q (↑)	HQ (↑)	Q (↑)	HQ (↑)	Q (↑)	HQ (↑)
Spectral [17]	N	0.3087	0.2907	0.3878	0.3733	0.3013	0.2833	0.3206	0.3028
	A	0.3721	0.3721	0.3958	0.3958	0.3553	0.3552	0.3594	0.3594
	D	0.3378	0.3308	0.3902	0.3818	0.3148	0.3066	0.3334	0.3240
Louvain [18]	N	0.7761	0.6866	0.8123	0.7561	0.8265	0.7994	0.8094	0.7688
	A	0.8164	0.8164	0.8235	0.8235	0.8214	0.8214	0.8202	0.8202
	D	0.7990	0.7702	0.8182	0.7895	0.8238	0.8070	0.8152	0.7938
GCC [48]	N	0.6612	0.6243	0.7032	0.6795	0.7186	0.6824	0.7218	0.6920
	A	0.7275	0.7275	0.7462	0.7462	0.7523	0.7523	0.7508	0.7508
	D	0.6846	0.6731	0.7315	0.7168	0.7420	0.7325	0.7436	0.7296
LSEnet [49]	N	0.6973	0.6342	0.7284	0.6857	0.7514	0.7269	0.7213	0.7029
	A	0.7202	0.7202	0.7423	0.7423	0.7558	0.7558	0.7412	0.7412
	D	0.7023	0.6876	0.7370	0.7152	0.7520	0.7325	0.7364	0.7283
CUSA (ours)	-	**0.8174**	**0.8190**	**0.8251**	**0.8245**	**0.8310**	**0.8224**	**0.8214**	**0.8211**

across the three datasets. Second, our proposed CUSA achieves significant improvements over the baseline equipped with three approaches across three datasets since CUSA can adaptively group HASN by balancing the tradeoff between AI removal and community integrity, leading to better clustering results. The baselines, however, fail to address the issue of humans interweaving with AIs.

5.2.3 Comparison of CUSA with Baselines on a Dataset using Different Generation Strategies. To further confirm the superiority and feasibility of CUSA, we now move on to evaluating the performance of CUSA on a dataset Cora which has been transformed into four HASNs using corresponding four proposed generation strategies: k% random insertion, inverse degree insertion, inner and outer AI mix, and AI with dual personality (*cf.* Figure 4 in Appendix A and Section 5.1). Note that each AI node-augmented Cora graph has evolved through link prediction such as Jaccard similarity [46] [47]. These evaluations were also conducted with four baselines each equipped with three approaches for handling AI nodes: no removal of AIs, all removal of AIs, and removal of certain AIs using GAD techniques for further comparison. We present the two evaluation metrics standard Q and human-centric HQ (*cf.* Section 3.1). The corresponding results are shown in Table 4, from which we can draw three important observations. First, each clustering baseline yields the best performance in terms of Q and HQ when all AI nodes are removed from the HASN, followed by using GAD to remove AI nodes before clustering, while the performance is worst when no AI nodes are removed across four HASN scenarios. Second, CUSA outperforms all baselines, leading to an average relative gain of 21.8% in Q and HQ across the baselines with all removal of AI nodes (taking inverse degree insertion as an example). In particular, the AAM and HMR of the baselines method with all removal of AI nodes are 0 and 0%, respectively, while our CUSA obtained 38.42 and 21.17%. These indeed demonstrate the efficacy of AI-aware clustering techniques for dealing with MetaCD problem since they identify and preserve AI nodes that can potentially reshape new communities and enhance human closeness. Third, as we can see, the elaborate generation strategies of the latter three result in clusters with significantly better performance compared to random insertion (k% random insertion). This also suggests that companies generating AIs can use this strategy to enhance human closeness and discover potential communities for such emerging social networks in the future.

6 CONCLUSION

To the best of our knowledge, this paper is the first to investigate human-centric community detection in hybrid networks (HASNs). By proposing four HASN scenarios, we introduce a new problem, MetaCD, which seeks to identify clusters that maximize human closeness while minimizing AI presence. To effectively address MetaCD, we develop a novel algorithm, CUSA, which incorporates AI-aware clustering techniques to balance the trade-off between AI removal and maintaining community integrity. Empirical evaluations on real social networks, transformed into HASNs, demonstrate the effectiveness of CUSA compared to top-of-the-line methods. Furthermore, we find that tailored generation strategies can enhance clustering outcomes, providing valuable insights for industries developing AIs to foster human connection and identify cohesive communities.

REFERENCES

[1] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.
[2] Di Jin, Zhizhi Yu, Pengfei Jiao, Shirui Pan, Dongxiao He, Jia Wu, S Yu Philip, and Weixiong Zhang. A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1149–1170, 2021.
[3] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, et al. A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[4] Nan Du, Bin Wu, Xin Pei, Bai Wang, and Liutong Xu. Community detection in large-scale social networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 16–25, 2007.
[5] Abhishek K Umrawal, Christopher J Quinn, and Vaneet Aggarwal. A community-aware framework for social influence maximization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4):1253–1262, 2023.

[6] A. Meijers. Global XR Conference: The Largest Global XR Community Event in the World. https://techcommunity.microsoft.com/t5/mixed-reality-blog/global-xr-conference-the-largest-global-xr-community-event-in/ba-p/3043539, 2021. Accessed: 2022-12-15.

[7] Meta. Workrooms. https://www.oculus.com/workrooms/?utm_medium=referral&utm_source=xr4work.com, 2023. Accessed: 2021-11-03.

[8] Nomi AI. I have a group chat with three AI friends, thanks to nomi ai — they're getting too smart. https://reurl.cc/MOeN3m, 2024. Accessed: 2024-07-06.

[9] ENGAGE. ENGAGE. https://engagevr.io/, 2023. Accessed: 2023-01-15.

[10] Hubs. Hubs mozilla. https://hubs.mozilla.com/?utm_medium=referral&utm_source=xr4work.com, 2023. Accessed: 2021-11-03.

[11] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12012–12038, 2021.

[12] The Conversation. Virtual influencers: meet the AI-generated figures posing as your new online friends as they try to sell you stuff. https://reurl.cc/Wx67oy, jul 2024. Accessed: 2024-07-10.

[13] TechCrunch. Meta plans to bring generative AI to metaverse games. https://reurl.cc/r9yAXZ, jul 2024. Accessed: 2024-07-10.

[14] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[15] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

[16] Prabin B Lamichhane and William Eberle. Anomaly detection in graph structured data: A survey. *arXiv preprint arXiv:2405.06172*, 2024.

[17] Arash A Amini, Aiyou Chen, Peter J Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. 2013.

[18] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[19] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[20] Stefan Boettcher and Allon G Percus. Optimization with extremal dynamics. *complexity*, 8(2):57–62, 2002.

[21] Mark EJ Newman. Spectral methods for community detection and graph partitioning. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 88(4):042822, 2013.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[23] Shima Khoshraftar and Aijun An. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1):1–55, 2024.

[24] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023.

[25] Youming Ge, Cong Huang, Yubao Liu, Sen Zhang, and Weiyang Kong. Unsupervised social network embedding via adaptive specific mappings. *Frontiers of Computer Science*, 18(3):183310, 2024.

[26] Ijeoma Amuche Chikwendu, Xiaoling Zhang, Isaac Osei Agyemang, Isaac Adjei-Mensah, Ukwuoma Chiagoziem Chima, and Chukwuebuka Joseph Ejiyi. A comprehensive survey on deep graph representation learning methods. *Journal of Artificial Intelligence Research*, 78:287–356, 2023.

[27] Shiping Wang, Jinbin Yang, Jie Yao, Yang Bai, and William Zhu. An overview of advanced deep graph node clustering. *IEEE Transactions on Computational Social Systems*, 11(1):1302–1314, 2023.

[28] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. *Advances in neural information processing systems*, 34:2148–2159, 2021.

[29] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. A survey of community search over big graphs. *The VLDB Journal*, 29:353–392, 2020.

[30] Ling Wu, Qishan Zhang, Chi-Hua Chen, Kun Guo, and Deqin Wang. Deep learning techniques for community detection in social networks. *IEEE Access*, 8:96016–96026, 2020.

[31] Yingxia Shao, Hongzheng Li, Xizhi Gu, Hongbo Yin, Yawen Li, Xupeng Miao, Wentao Zhang, Bin Cui, and Lei Chen. Distributed graph neural network training: A survey. *ACM Computing Surveys*, 56(8):1–39, 2024.

[32] Yu Pan, Feng Yao, Xin Liu, Lei Zhang, Wei Li, and Pei Wang. Identification of dynamic networks community by fusing deep learning and evolutionary clustering. 2024.

[33] Tristan Bilot, Nour El Madhoun, Khaldoun Al Agha, and Anis Zouaoui. Graph neural networks for intrusion detection: A survey. *IEEE Access*, 11:49114–49139, 2023.

[34] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14557–14565, 2023.

[35] Rose Yu, Huida Qiu, Zhen Wen, ChingYung Lin, and Yan Liu. A survey on social media anomaly detection. *ACM SIGKDD Explorations Newsletter*, 18(1):1–14, 2016.

[36] Amit Roy, Juan Shu, Jia Li, Carl Yang, Olivier Elshocht, Jeroen Smeets, and Pan Li. Gad-nr: Graph anomaly detection via neighborhood reconstruction. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 576–585, 2024.

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

[38] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[39] AnthropicAI. Introducing claude. 2023. Retrieved from https://www.anthropic.com/index/introducing-claude.

[40] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[41] Linton C Freeman et al. Centrality in social networks: Conceptual clarification. *Social network: critical concepts in sociology. Londres: Routledge*, 1:238–263, 2002.

[42] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[43] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[44] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[45] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[46] Djihad Arrar, Nadjet Kamel, and Abdelaziz Lakhfif. A comprehensive survey of link prediction methods. *The journal of supercomputing*, 80(3):3902–3942, 2024.

[47] Nur Nasuha Daud, Siti Hafizah Ab Hamid, Muntadher Saadoon, Firdaus Sahran, and Nor Badrul Anuar. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, 166:102716, 2020.

[48] Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. Efficient graph convolution for joint node representation learning and clustering. *Proceedings of the fifteenth ACM International conference on web search and data mining (WSDM)*, pages 289–297, 2022.

[49] Li Sun, Zhenhao Huang, Hao Peng, Yujie Wang, Chunyang Liu, and Philip S Yu. Lsenet: Lorentz structural entropy neural network for deep graph clustering. *Proceedings of the 41 st International Conference on Machine Learning (ICML)*, 2024.

[50] Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1239–1248, 2021.

A APPENDIX

A.1 The proposed four HASN scenarios

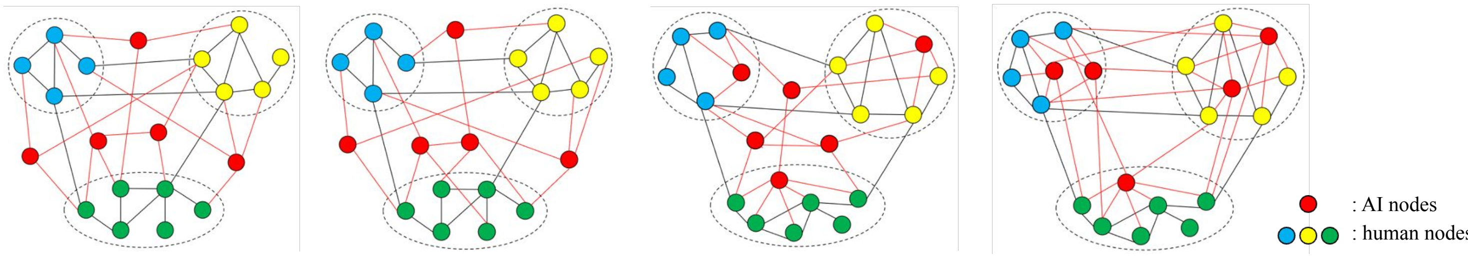
We outline the four proposed generation strategies with the specific assumptions [8][12][13][15][45], as depicted in Figure 4. These strategies reflect various interaction patterns between human and AI nodes, laying the foundation for constructing meaningful human-AI social networks (HASNs) for our experiments.

(1) ***k% random insertion***: AI interacts randomly with humans (or other AIs) with a uniform distribution of $k\%$. This setup assumes that everyone in the evolving Metaverse has equal chances to interact with AI, as depicted in Figure 4 (a).

(2) ***Inverse degree-based insertion***: The probability of an AI interacting with humans (or other AIs) is inversely proportional to the node's degree. This can be modeled using an exponential decay function, given by $prob\_connect(v_i) = a \times e^{-r \times (d_{v_i} - 1)}$, where d_{v_i} is the degree of node v_i, a is the initial probability (i.e., when $d_{v_i} = 1$), and r is the decay constant. This indicates that nodes with higher degrees are more likely to be linked by AI. This setup assumes that certain individuals tend toward introversion and prefer interactions with AI virtual entities, as depicted in Figure 4 (b).

(3) ***Introverted and extroverted AI equally distributed***: This suggests the existence of two distinct AI types: introverted and extroverted. Introverted AI engages extensively with members in their respective groups with a probability of $x\%$, while extroverted ones interact with members across various groups with a probability of $y\%$ (typically with $x > y$). This setup assumes that half of the AIs engage primarily with group members due to shared interests, while the other half possess versatility, enabling interaction across multiple groups, as depicted in Figure 4 (c).

(4) ***AI with dual personality***: This suggests that AI may demonstrate dual personality, actively engaging with members within its own community with a probability of $x\%$, while also interacting with individuals in other communities with a probability of $y\%$ (typically with $x > y$). This setup assumes that AIs adeptly interact with community members due to shared interests and are also versatile enough to engage with individuals from other communities, as depicted in Figure 4 (d).



(a) k% random insertion (b) Inverse degree-based insertion (c) Introverted and extroverted AI equally distributed (d) AI with dual personality

Figure 4: The proposed four HASN scenarios.