

HyperGraphDis: Leveraging Hypergraphs for Contextual and Social-Based Disinformation Detection

Nikos Salamanos¹, Pantelitsa Leonidou¹, Nikolaos Laoutaris², Michael Sirivianos¹,
 Maria Aspri³, Marius Paraschiv²

¹ Cyprus University of Technology, Limassol, Cyprus

² IMDEA Networks, Madrid, Spain

³ LSTECH ESPANA SL, Barchelona, Spain

nik.salaman@cut.ac.cy, pl.leonidou@edu.cut.ac.cy, nikolaos.laoutaris@imdea.org, michael.sirivianos@cut.ac.cy,
 aspri@lstech.io, marius.paraschiv@imdea.org

Abstract

In light of the growing impact of disinformation on social, economic, and political landscapes, accurate and efficient identification methods are increasingly critical. This paper introduces HyperGraphDis, a novel approach for detecting disinformation on Twitter that employs a hypergraph-based representation to capture (i) the intricate social structures arising from retweet cascades, (ii) relational features among users, and (iii) semantic and topical nuances. Evaluated on four Twitter datasets – focusing on the 2016 U.S. presidential election and the COVID-19 pandemic – HyperGraphDis outperforms existing methods in both accuracy and computational efficiency, underscoring its effectiveness and scalability for tackling the challenges posed by disinformation dissemination. HyperGraphDis displays exceptional performance on a COVID-19-related dataset, achieving an impressive F1 score (weighted) of approximately 89.5%. This result represents a notable improvement of around 4% compared to the other state-of-the-art methods. Additionally, significant enhancements in computation time are observed for both model training and inference. In terms of model training, completion times are accelerated by a factor ranging from 2.3 to 7.6 compared to the second-best method across the four datasets. Similarly, during inference, computation times are 1.3 to 6.8 times faster than the state-of-the-art.

Introduction

Social media platforms have evolved from mere fora for informal conversation and idea exchange into essential tools for a diverse array of entities, including non-governmental organizations, businesses, governments, and media outlets. Their ubiquity and ability to personalize content according to user behavior make them pivotal in the modern social landscape. Platforms like Twitter (currently known as X) and Facebook now serve as the principal sources of information on a worldwide scale (Westerman, Spence, and Van Der Heide 2014). Nevertheless, the dissemination of false information on these platforms can have detrimental effects on society, politics, and the economy, making disinformation a rapidly escalating cyber threat (Belova and Georgieva 2018). Contemporary techniques for identifying disinformation on social media chiefly focus on partial data, either con-

cerning user behavior or the topology of the local network. Such methods may lead to unsatisfactory results, given that the effectiveness of disinformation spread relies on a multifaceted interplay of intrinsic elements like user traits and intrinsic factors such as network features and post content.

On Twitter, users create messages (*tweets*), and others share these messages (*retweets*). A posted message initiates a *root-tweet* (the original tweet that gets retweeted), followed by a sequence of retweets forming a tree-like graph known as a *retweet cascade*. In this study, we aim to improve disinformation detection accuracy and computational efficiency on social media by introducing an integrated approach that considers both intrinsic and extrinsic variables. For this purpose, we propose HyperGraphDis, an innovative approach utilizing a hypergraph data structure to encapsulate both the content of retweet cascades and the relational information among users. The choice of a hypergraph stems from its ability to represent complex relationships more naturally than traditional graphs. Unlike standard graphs with vertices connected by edges, a hypergraph features hyperedges that can connect more than two nodes. This allows for a nuanced representation of Twitter interactions, considering both the content of tweets and retweets (referred to as retweet cascades), as well as the network of users involved in these cascades.

In the initial phase of hypergraph construction, we apply a graph partitioning algorithm to the Twitter social network, where nodes represent users and edges represent social connections. This process effectively identifies user clusters with similar interaction patterns or content sharing. The essence of graph partitioning is to optimize the divisions in such a way that intra-cluster connections are maximized while inter-cluster connections are minimized, thereby identifying natural communities within the Twitter user base. Upon identifying these user clusters, we undertake a transformation wherein each user in a cluster is substituted with a list of Twitter cascades in which they have partaken. In the hypergraph framework, each previously discovered cluster becomes a hyperedge. The Twitter cascades, which are sequences of tweets and retweets initiated by a root-tweet, are represented as the nodes within these hyperedges. This transformation inherently reshapes the problem space: it turns the complex task of disinformation classifica-

tion on Twitter from an intricate, multi-variable problem into a more straightforward node classification problem within the hypergraph.

The use of hypergraph data structures offers several advantages over traditional graphs. Unlike conventional graph edges that connect only pairs of nodes, a hyperedge links an entire cluster of users, thereby enriching the relational information in the dataset by capturing multi-way associations. This multi-faceted relational insight becomes even more potent when considering hyperedge intersections, which naturally coincide with the overlap of user clusters. Such overlaps allow for a nuanced understanding of classes of users sharing specific common features, further augmenting the dataset’s expressiveness. Moreover, hypergraphs inherently result in less dense network representations, yielding benefits in terms of reduced computational processing times and more efficient storage requirements. This makes hypergraphs an excellent choice for the effective modeling of complex relational systems. Furthermore, we enrich each Twitter cascade node with a set of features, such as the frequency of specific keywords, sentiment analysis scores, or the presence of verified accounts (with the Twitter verification badge). By doing so, we transform the issue of classifying Twitter cascades in terms of disinformation into a node classification task within the hypergraph. For the node classification task, we employ hypergraph neural networks, a modern machine learning approach optimized for hypergraph structures.

We evaluate our approach on four datasets:

(i) an extensive dataset on the 2016 U.S. presidential election; (ii) a substantial collection of tweets related to the COVID-19 pandemic, along with social engagement data (Li et al. 2020); and (iii) the Health Release and Health Story datasets, integral parts of the FakeHealth repository (Dai, Sun, and Wang 2020). Overall, the HyperGraphDis shows exceptional performance (see Table 3). More specifically, evaluating it with MM-COVID achieves an impressive F1 score of around 89.5%. It outperforms the Metagraph method (Paraschiv et al. 2022) by approximately 4%, and Cluster-GCN (Chiang et al. 2019) by 33%. Additionally, HyperGraphDis outperforms HGFND (Jeong et al. 2022) by 6.4% and other state-of-the-art ML text-based techniques by 10% (see Table 4). Furthermore, noteworthy enhancements are observed in the computation time for both model training and inference processes. Regarding model training, the completion time is notably expedited, ranging from 2.3 times to 7.6 times faster than the second-best method per dataset (in terms of F1 score). Similarly, computational time exhibits increased efficiency during inference, ranging from 1.3 times to 6.8 times faster than the second-best method across the four datasets.

The results demonstrate that our method not only yields levels of accuracy that are comparable to or exceed existing methods relying on both tweet content and user relationships, but it also does so while offering the advantages of reduced data size and faster classification run times. Moreover, the inherent design of our method makes it highly scalable, allowing it to adapt to varying sizes and complexities of data.

Data availability: Part of the US Election dataset is publicly available (Salamanos et al. 2023) under proper restrictions for compliance with Twitter’s Terms of Service (Tos) and General Data Protection Regulation (GDPR). The FakeHealth repository and the MM-COVID dataset are publicly available (Dai, Sun, and Wang 2020), (Li et al. 2020).

Related Work

Considerable research has been dedicated to the problem of identifying false information on social media platforms. Initial methods primarily focused on feature-based detection at the individual user level. The work by (Ahmed, Traore, and Saad 2017), for instance, employs a variety of feature extraction techniques on social media content to identify linguistic patterns indicative of disinformation. However, this approach fails to account for the relational data between the information sources and the users on platforms like Twitter. Similarly, (Ghosh and Shah 2018) decompose the problem into two distinct phases: feature extraction and classification, while (Zhang, Zhao, and Lecun 2015) employ character-level convolutional neural networks to merge these steps. More contemporary methods include the use of BERT language models for feature extraction as outlined by (Kaliyar, Goswami, and Narang 2021) and reviewed in depth by (Zhou and Zafarani 2020). On another front, (Castillo, Mendoza, and Poblete 2011) evaluate the trustworthiness of social media posts by examining users’ historical behavior, the tweets’ content, and additional external data. (Shu et al. 2020b) take a different tack by extracting features from individual user profiles to identify those more likely to disseminate false information. This bears resemblance to the strategies discussed in (Tacchini et al. 2017; Guo et al. 2018; Jin et al. 2017), which leverage characteristics of user types to categorize posts as false or genuine.

Turning to methods that incorporate social context, these consider both the relationships between users sharing news and additional contextual information about the user and the news content. (Gupta, Zhao, and Han 2012) introduce an approach that evaluates credibility through a network that includes events, tweets, and users. While this conceptually aligns with our method, it overlooks the complex structural data of information cascades and their similarities. (Shu et al. 2019) suggest a co-attention mechanism between sentences and comments to explain why a news item is flagged as false by an algorithm. Concurrently, (Wu and Liu 2018) focus on identifying false information by analyzing its propagation behavior across the network.

A method closely related to ours is discussed by (Shu, Wang, and Liu 2019), where an embedding framework known as TriFN is utilized. It models the relationships between publishers and news items, as well as user interactions with those news pieces. However, it lacks the incorporation of relational data between events or cascades, which can be crucial for evaluating historical user behavior. Additionally, it does not leverage semantic features gained through topic modeling and sentiment analysis. Another noteworthy method is presented by (Shu et al. 2020a), who construct a hierarchical propagation network for false information and

engage in a comparative study between false and true news from linguistic, structural, and temporal lenses.

We mention here other recent efforts, such as (Chen et al. 2022), which employs automatic methods and machine learning models like RoBERTa-large to assess the veracity of claims in medical datasets. This work complements efforts in understanding web-based relationships, where (Hanley, Kumar, and Durumeric 2023) utilizes Common Crawl data to analyze hyperlinks between websites. In (Hanley and Durumeric 2023), the authors present a comprehensive framework to study Reddit communities, targeting misinformation, toxicity, and political polarization. Similarly, (Pratelli et al. 2023) delve into the behavior of social bots on Twitter, categorizing them based on their political leanings during the 2018 U.S. midterms. (Balci, Sirivianos, and Blackburn 2023) extend this by introducing metrics, MRD and MRS, to quantify user migrations across subreddits on the Reddit social platform. Furthermore, In (Elmas, Overdorf, and Aberer 2023), the focus is on detecting repurposed Twitter accounts using supervised learning, shedding light on their characteristics and behaviors.

In (Kozik et al. 2023), the authors conducted a meta-analysis of state-of-the-art Machine Learning techniques for text-based fake news detection, including Random Forest, CNN-based, LSTM-based, and BERT-based models. They evaluated these models on benchmark datasets, including MM-COVID. We evaluated the HyperGraphDis method using MM-COVID, consistently achieving superior performance compared to all mentioned models in terms of F1 score. The SOMPS-Net framework (Dhanasekaran et al. 2021), with the Social Interaction Graph and Publisher and News Statistics components, achieved a 72.7% accuracy and 79.6% F1 score on the Health Story dataset for fake news prediction. Evaluating HyperGraphDis on the same dataset, we observed an accuracy of 75.2% and F1 score of 72.2%. We attribute this decline in F1 score to the inherent characteristics of the cascades within the Health Story dataset, which tend to be either extremely small or predominantly structured like star-like graphs. This characteristic leads to embedding vectors that exhibit high levels of similarity, consequently affecting the overall effectiveness of our method.

The methods most closely related to ours are (Paraschiv et al. 2022) and (Jeong et al. 2022). In (Paraschiv et al. 2022), the Meta-graph method captures content and relational information. (Jeong et al. 2022) introduces a hypergraph-based neural network (HGFND) for detecting relational fake news with limited labeled data. They formulate fake news detection as a semi-supervised node classification task in a hypergraph, capturing group-level interactions among news articles. Our analysis reveals that HyperGraphDis outperforms these methods, providing superior results with enhanced computational efficiency.

In evaluating the performance metrics of our proposed method, several key factors underscore its superiority, particularly when compared to classical graph-based approaches. First, the inherently reduced connectivity of the hypergraph structure yields faster computational times and minimizes hardware resource requirements. This efficiency is not achieved at the expense of informational richness; to

the contrary, hyperedges encapsulate a wealth of multi-way relationships among nodes, providing a more nuanced understanding of underlying data patterns. Additionally, the intersections of hyperedges serve as focal points for identifying user subsets with highly detailed common features, adding an extra layer of depth to the dataset’s relational information content.

Similar Approaches

Meta-graph

In (Paraschiv et al. 2022), the authors introduce a *meta-graph* data structure to represent events in a social network. Formally, a meta-graph $\mathcal{G} = (V, E)$ consists of vertices V and edges E , where each vertex corresponds to a social network event and is characterized by a feature vector X_v .

The node feature vector X_v encapsulates user attributes, tweet content, and cascade structural information. Specifically, $X_v = (C_{\text{emb}} || X_u || T_{\text{emb}} || S \dots)$, where $||$ signifies vector concatenation. The term C_{emb} denotes the cascade embedding vector, X_u encompasses concatenated user features, T_{emb} is the text embedding vector, and S represents sentiment analysis scores if available (Perozzi, Al-Rfou, and Skiena 2014; Devlin et al. 2019).

Edges in the meta-graph are constructed based on common users or topics and are described by edge feature vectors $R_{e_{i,j}}$. An example vector is $R_{e_{i,j}} = (N_{u_{i,j}} || V_{i,j} || H_{i,j} \dots)$. Here, $N_{u_{i,j}}$ indicates the number of common users between cascades i and j , $V_{i,j}$ is a graph similarity metric, and $H_{i,j}$ is a content similarity metric between the root-tweets of the two cascades (Zager and Verghese 2008; Blondel et al. 2004; Zhao et al. 2020; Bisseling 2020; Goma and Fahmy 2013). Both node and edge feature vectors are designed for extensibility ease, allowing the incorporation of additional relational information when needed.

Despite its utility in certain scenarios, the meta-graph method presents several notable limitations that hamper its broader applicability. One primary drawback is the high density of its graph representation, which demands considerable computational resources and results in an increased hardware footprint. Additionally, the meta-graph edges are confined to capturing bipartite connections between entities, failing to represent more complex relational information. This restrictiveness also impedes the exploitation of partial feature overlaps among different user clusters. To overcome these challenges, we introduce HyperGraphDis, a novel approach that employs a hypergraph data structure to address these inherent shortcomings effectively.

HyperGraph for Fake News Detection (HGFND)

In (Jeong et al. 2022), the authors tackle the challenge of detecting fake news by framing it as a semi-supervised node classification task within the context of a hypergraph. They propose a hypergraph construction methodology that establishes connections between news pieces (nodes) based on both the news content and social context – the propagation of news on Twitter. They create a hypergraph with three hyper-edge types: **i) User**: a hyperedge for each user I.D., linking news pieces shared by a specific user; **ii) Time**: a hyperedge

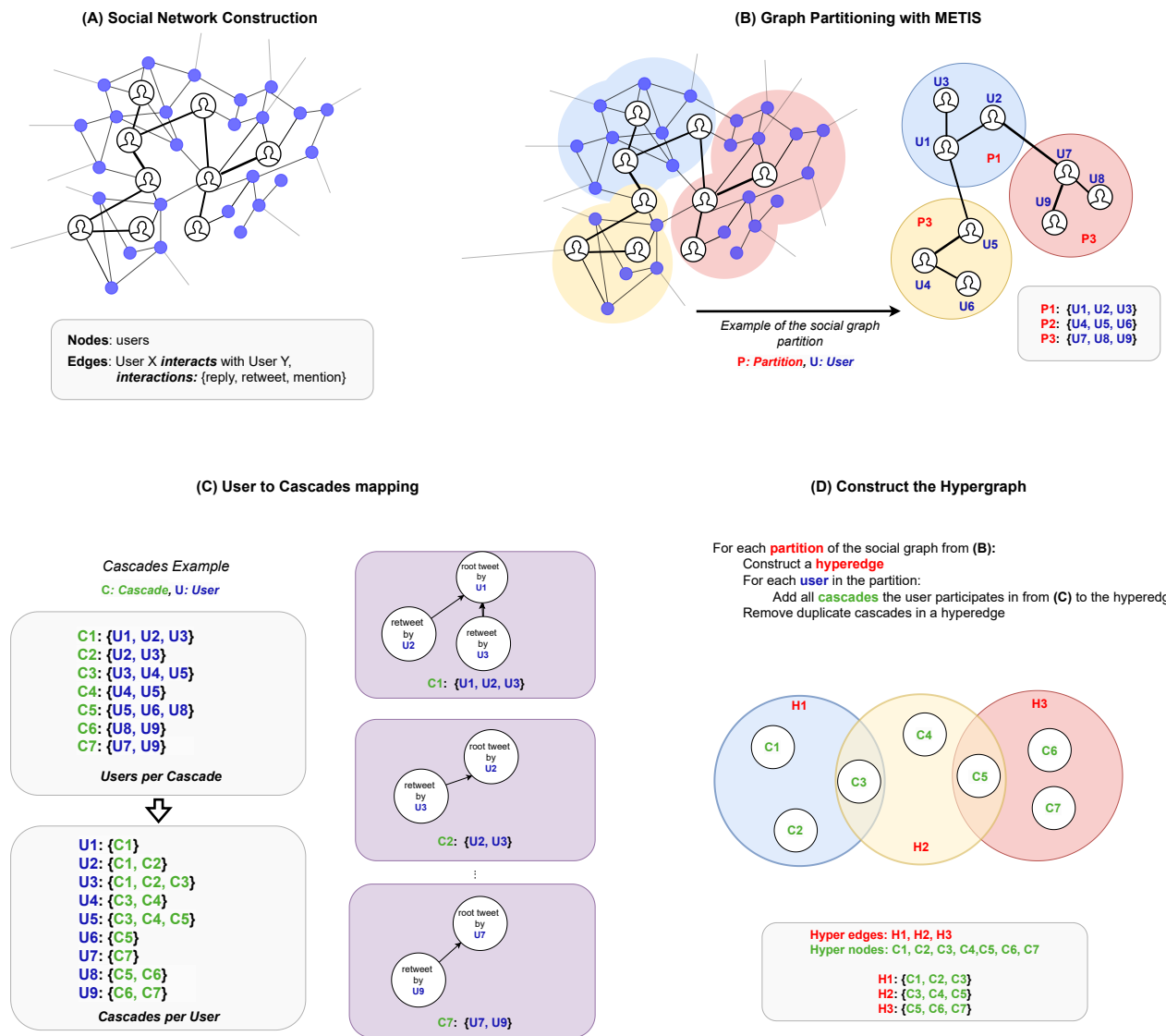


Figure 1: A toy example of the proposed hypergraph construction pipeline.

for a proximal time range, connecting news pieces shared during the same period; **iii) Entity:** a hyperedge for each entity, connecting news pieces that share the same content entity (e.g., "Covid-19").

Finally, they introduce HGFND, a hypergraph neural network model designed to handle noisy relations within the hypergraph. HGFND employs a dual-level attention mechanism, assessing the significance of news relations at both node and hyperedge levels. Experimental results on two benchmark datasets (Politifact and Gossipcop) demonstrate HGFND’s high performance in the cascade classification task with the hypergraph constructed following our approach (see Section *The HyperGraphDis Method*) and the results are shown in Table 3.

Cluster-GCN

In (Chiang et al. 2019), the authors present Cluster-Graph Convolutional Network (Cluster-GCN) that leverages graph clustering to achieve optimized Stochastic Gradient Descent (SGD)-based training in a mini-batch fashion. Cluster-GCN first partitions the graph utilizing METIS (Karypis and Kumar 1998) – a graph partitioning algorithm, to derive non-overlapping clusters of nodes. As a mini-batch of size q , it randomly selects q clusters to construct a subgraph with nodes from the chosen clusters and the links connecting them in the original graph.

For each SGD update on a mini-batch, this approach constrains GNNs to convolve exclusively within their designated subgraphs, thereby addressing the issue of neighborhood expansion. This restriction results in optimized memory consumption and reduced computational costs during training, allowing large-scale training on graph structures.

In this work, we evaluate the performance of Cluster-GCN on the meta-graph data structure as a baseline. Note that in Cluster-GCN, METIS is applied to the meta-graph to generate clusters of cascades, whereas in HyperGraphDis, METIS generates clusters of users applied to the social network.

The HyperGraphDis Method

The HyperGraphDis method addresses the detection of disinformation in social networks by exploiting network structure as well as post content and user sentiment. As we will show in the next section, the network structure has a clear impact on classification results. As part of the HyperGraphDis method, a hypergraph is constructed from the user-user social network as well as the individual tweet-retweet cascades, which will serve as the nodes of the hypergraph. The hypergraph structure is used to perform binary classification of retweet cascades into fake and non-fake classes. Two primary challenges need to be tackled to apply our methodology to the raw Twitter data. First, the actual structure of the social network, specifically the follower graph, remains largely unknown, as access to users' follower lists is not always available. Second, the data obtained through the Twitter API offer limited insights into the origins of influence within retweet cascades. The only accessible information consists of the root-tweet and the root-user associated with a particular retweet. In simpler terms, all retweeters are linked to the root user, creating a star-like cascade structure that does not always accurately represent user-to-user interactions or the source of information diffusion. The methodology steps to address the above issues are the following:

Phase-1: The Hypergraph Construction

Given a large collection of tweets, we construct a hypergraph with the retweet cascades as nodes (i) From the overall collection of tweets (tweets, replies, retweets), we construct the user-user social network – the graph of interactions between the users. This graph represents an approximation of the Twitter social network. (ii) We partition this graph (in disjoint sets of nodes/users) by applying METIS – a state-of-the-art graph partitioning algorithm. Then, in each partition (set of users), we replace the users with the retweet cascades that have participated. In simple words, clusters of cascades replace clusters of users, and each of these clusters represents a hyperedge.

Figure 1 presents a toy example of the main HyperGraphDis phases for the hypergraph construction where the classification of disinformation is taking place. The hypernodes are the cascades that are represented by feature vectors (as defined in the next Section, “Phase-2: The Hypernodes Features Vectors”). In our methodology, a hyperedge connects two or more cascades if the users of the same partition of the Twitter social graph participate in these cascades. To illustrate this, consider the social network and the corresponding graph partition in Figure 1(A) & (B). Initially, we create a user-to-cascades mapping, as shown in Figure 1(C). Subsequently, we replace each user within a given partition from Figure 1(B) with the cascades they have engaged with

from Figure 1(C). This process leads to the formation of the hypergraph (Figure 1(D)) where the hyperedges connect these sets of cascades. The construction of the hypergraph is scalable since several types of cascade connections (hyperedges) can be defined. For example, we can construct hyperedges based on tweet hashtags; i.e., two or more cascades belong to the same hyperedge if the root-tweets have the same hashtag.

In Figure 1, we illustrate the process of building the cascades hypergraph. This begins with the formation of the social network and follows through to capturing users' participation in cascades, all based on the selected Twitter dataset under consideration. A detailed description of the proposed pipeline is as follows:

(A) Social Network Construction: Initially, we create an approximation of the Twitter social graph, also known as the follow-graph, grounded in the activities of users within our dataset. Within this social network, each user corresponds to a node. We add a directed edge for any interaction between two nodes. The “interactions” are defined by the user-user actions in the Twitter platform, such as “retweet,” “reply,” or “mention”. Hence, an edge(i,j) between two nodes implies that the *node i* follows the *node j*. The output of this process is a multi-graph because of the possibility of multiple interactions (edges) between two nodes. We remove the duplicate edges and keep the earliest ones.

(B) Graph Partitioning: Having the resulting Twitter social graph, we conduct graph partitioning to identify communities or groups of users. After applying the selected graph partitioning technique, we get a list of users per partition. In this paper, we leverage the METIS graph partitioning algorithm (Karypis and Kumar 1998).

(C) User to Cascades Mapping: We define a list of users that are involved in each cascade based on the raw Twitter data. Finally, we derive a user-to-cascade mapping that denotes that a given user i participated in the set of cascades $\{x, y, z\}$.

(D) The Final Hypergraph: Using the graph partition from Step-(B) and the mapping user-to-cascade from Step-(C), we replace the users in the partition clusters with the list of cascades they participated in. This process results in sets of cascades (removing the duplicates), which will define the hyperedges of the final hypergraph.

Phase-2: The Hypernodes Features Vectors

A retweet cascade is a series of chain events upon the same root-tweet where users have retweeted the root-tweet, whereas some others have retweeted a retweet of a friend (retweet of a retweet). This tree-like structure reflects who was influenced by whom during the retweeting process. Unfortunately, the raw data by Twitter API does not provide that since it contains only the root-user and retweeter IDs. In short, the raw data represents a star-like graph where all retweeters are only connected to the root user. This form lacks sufficient structural information.

To tackle this issue, we construct a subgraph comprising the star-like graph along with past interactions involving the users who have participated in the cascade in question. Specifically, for a given retweeter i who engaged in the

retweet action at date t , we determine their pre-date t connections among their friends and ascertain which of them are among the retweeters in this cascade. Subsequently, we augment this collection of edges, if present, with the star-like configuration where each retweeter maintains a connection with the root user. Finally, we eliminate any duplicate edges and transform the resulting graph into an undirected representation. Through this methodology, the subgraph consistently retains its connectivity.

The rationale of this approach is that each cascade (i.e., a node of the hypergraph) should have the corresponding subgraph structure as a feature that will reflect the social interactions that the retweeters had. For this purpose, we apply the DeepWalk algorithm (Perozzi, Al-Rfou, and Skiena 2014) to produce the embedding of a cascade subgraph to a low dimensional space. For the implementation, we used the *Karate Club* extension for the NetworkX (Rozemberczki, Kiss, and Sarkar 2020). The DeepWalk embedding is a $N \times 128$ matrix, where N is the number of retweeters. The default parameters are: (i) Number of random walks = 10; (ii) Length of random walks = 80 and for embeddings (iii) Dimensionality of embedding = 128.

Figure 2 provides a toy example: (i) Left-hand figure: the raw data that are provided by the Twitter API correspond to a star-like graph structure with limited structural information regarding the user-user relation between the retweeters. (ii) Right-hand figure: The augmented cascade graph. (iii) Bottom figure: the final cascade object – a set of features that represent the retweet cascade. The feature vector consists of the individual users’ features, the DeepWalk embeddings as well as topic and sentiment analysis (as the tweet-text features). This will be part of the training data of HyperGraphDis.

Phase-3: Cascade Classification

We formulate the cascade classification problem as a node classification task. The HyperGraphDis uses a Hypergraph Convolution (HypergraphConv) layer (Bai, Zhang, and Torr 2021) followed by three linear (fully connected) layers. Given a hypergraph structure from Phase 1 and 2, the nodes, their feature vectors, and the hyperedge index serve as input to the HypergraphConv layer, where information aggregation occurs within hyperedges, considering the collective influence of interconnected nodes. The output is a set of transformed node features representing the updated state of each node. The architecture employs Rectified Linear Unit (ReLU) activation functions for non-linearity after each layer. We add dropout regularization between linear layers to enhance generalization. The output of the pipeline represents the model’s predictions (i.e., cascade is fake or non-fake). While training Graph Neural Networks (GNNs), it is crucial to highlight the absence of label leakage issues. This is attributed to the fact that, during the training phase, the labels of nodes beyond the training dataset remain unknown. This ensures the integrity of the training process, preventing unintended information leakage from labels outside the designated training set.

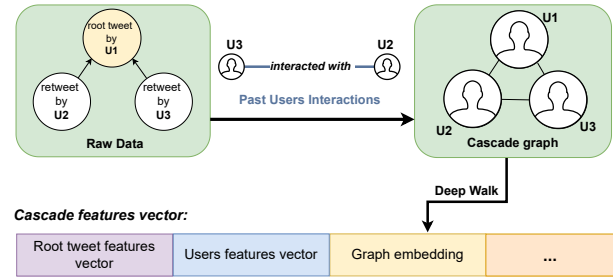


Figure 2: A toy example of a hypergraph node. Left-hand: the raw retweet data provided by Twitter API, always corresponds to a star-like graph with limited structural information. Right-hand: We enhance the raw data by appending the past interactions between the retweeters (i.e., the users) to the star-like graph. Finally, we compute the node embeddings (in this enhanced subgraph) using the DeepWalk algorithm. Bottom: The final feature vector consists of the DeepWalk embeddings together with users’ features and tweet-text features (topics, sentiments, etc.)

Dataset Descriptions

US Election

The dataset in (Salamanos et al. 2023), encompassing 152.5 million tweets from 9.9 million users during the 2016 U.S. presidential election, was harvested between September 21st and November 7th, 2016, using the Tweepy Python library. Each tweet is feature-rich, encapsulating 27 attributes related to the tweet and the corresponding Twitter account, including tweet-ID, tweet-text, user-ID, screen name, and more. Additional metadata, such as mentions and embedded URLs, are also part of the dataset. The dataset also emphasizes retweet cascades, specifically those rich in transmitted information and broad user engagement. This includes cascades where the original tweet contains at least one web or media URL and has been retweeted by a minimum of 100 users. In total, 46.4K such retweet cascades, involving 19.6 million tweets, are included for analysis. With labeled media and web URLs – embedded in the tweets – categorized as "fake," "non-fake," or "unknown," derived through a multi-step methodology and manual annotation using Fact-Checking tools, the dataset provides a well-curated set of 43.9K URLs. There are 6525 URLs labeled as "fake" and "non-fake" that are present in 8323 cascades (see Table 1).

MM-COVID

The authors in (Li et al. 2020) created MM-COVID, a multilingual and multimodal dataset to combat misinformation on COVID-19. Despite the news content, the dataset also contains Twitter’s social network and spatial-temporal information. The dataset consists of approximately 4K fake news articles and about 7K trustworthy articles related to COVID-19 across six languages. There are over 1K fake news documents in the English language subset and approximately 5K legitimate articles. We dehydrated the user engagement data only for the English language dataset subset

Dataset	Retweet Cascades		Social Network						Meta-graph	
	Fake	Non-fake	Nodes (users)	Edges	Connected Comp.	Giant Comp. % of nodes	Infomap #comm.	Louvain #comm.	Nodes (cascades)	Edges
US Election	5,890	2,433	482,171	34,691,548	1	100%	2047	14	8,319	7,763,018
Health Story	3,574	9,689	147,607	187,096	12,719	73.75%	12894	12850	13,263	100,001
MM-COVID	578	1,230	38,813	50,688	4	99.78%	47	101	1,808	87,948
Health Release	929	1,040	18,470	21,502	1,595	73.32%	1645	1659	1,969	11,581

Table 1: Cascades, social network and meta-graph statistics per dataset.

Features	US Election	MM-COVID	Fake Health
User-related			
DeepWalk Embedding	✓	✓	✓
Account Creation Date	✓	✓	✓
Verified Account	✓	✓	✓
Account Language	✓	✓	✓
Max Statuses count	✓		✓
Max Favorites count	✓		✓
Max listed count		✓	
Reaction Time	✓	✓	
Max Followers count	✓	✓	✓
Max Friends count	✓	✓	✓
Max impression count		✓	
Max reply count		✓	
Max quote count		✓	
Max like count		✓	
Text-related			
Sentiment Analysis	✓	✓	✓
Topic Detection	✓		✓

Table 2: Cascade features.

using Twarc¹ (a Python library for collecting data via Twitter API). Only a subset of the data was available to collect at the time of the dehydration of the existing dataset. In summary, we used 53,869 tweets, 10,387 replies, and 85,121 retweets by 93,152 users for the experimental evaluation. From this dataset, we kept the 1,808 cascades that have at least 10 participants (see Table 1). The tweets are in English only; the quoted tweets have been discarded.

FakeHealth

To rigorously assess the efficacy of our proposed methodology, we employ two datasets originating from the FakeHealth repository (Dai, Sun, and Wang 2020). While the authors of the repository are constrained by Twitter’s user privacy policies from publicly disseminating the full extent of user engagement and network data, they have circumvented this limitation by offering an API <https://github.com/EnyanDai/FakeHealth>. This API facilitates the acquisition of comprehensive data pertaining to users’ social engagements and networks. Using this API, we collected the following datasets: (i) **Health Release** with a size of 60,006 – 43,245 tweets, 1,418 replies, and 15,343 retweets; (ii) **Health Story** with a size of 487,195 – 357,851 tweets,

¹<https://twarc-project.readthedocs.io/en/latest/>

23,632 replies, and 105,712 retweets. It is noteworthy that the final data metrics we report exhibit a marginal reduction, approximately 1.1%, compared to the figures disseminated by the original authors. Several factors contribute to this discrepancy: (i) modifications in user settings that restrict public accessibility to their tweets, (ii) deletions of tweets by Twitter in adherence to their internal policies, and (iii) the outright deletion of Twitter user accounts.

Evaluation of Performance Benefits

In this section, we evaluate HyperGraphDis across all four previously mentioned datasets and simultaneously compare it with the Meta-graph, HGFND and Cluster-GCN approaches. First, we present the basic structures across the four datasets that are the input of the HyperGraphDis and the other three methods, and then we discuss the actual classification performance.

The code has been implemented in Python and executed in the CPU of an ASUS laptop with Intel Core i7-6700HQ CPU 2.60GHz × 8; 32GB RAM; Ubuntu 22.04.3 LTS. The HypergraphConv (Bai, Zhang, and Torr 2021), GCN-Conv (Kipf and Welling 2017) and Cluster-GCN (Chiang et al. 2019) have been implemented in PyTorch Geometric (Fey and Lenssen 2019), maintaining the following hyper-parameters constant across experiments: #epochs = 100, dropout = 0.5, learning rate = $5 * 10^{-4}$. For the HGFND we use the official implementation². We use the default parameters dropout = 0.3 and learning rate = 0.001. In all experiments, we train the models for 100 epochs. We use an 80/20 train-test split on the labeled cascade set across all datasets and methods. We repeat the experiments for five trials, where we regenerate the train-test split randomly in each trial. Finally, we use the official implementation of Infomap³ executing multi-level clustering for 10 internal trials. Louvain has been implemented by NetworkX⁴.

Cascades, Social Networks and Meta-graphs: The statistical characteristics of the cascades, social networks, and meta-graphs constructed for the four datasets are presented in Table 1. In the US Election dataset, since the meta-graph constructed is extremely dense (15,946,910 undirected edges connect 8323 nodes/cascades), we applied the disparity filter (Serrano, Boguñá, and Vespignani 2009) for

²https://github.com/ujeong1/IEEESBigdata22_HGFND

³<https://www.mapequation.org/infomap/>

⁴<https://networkx.org>

$\alpha = 0.5$, which depicts the best classification results according to (Paraschiv et al. 2022). 49% of the edges survived the significance test. However, 4 nodes remained isolated and were discarded, resulting in a meta-graph of 8319 cascades.

Cascade Features

In all datasets, we define a threshold for the number of initial retweeters (users) that we will include in the features (to prevent out-of-memory issues). The thresholds are US Election: 250; MM-COVID: 50; FakeHealth: 60. After constructing the feature matrix, we employ PCA for dimensionality reduction. The dimensions are reduced to US Election: 90, MM-COVID: 60; FakeHealth: 60. Table 2 presents the selected parameter values and outlines the user and text-related features employed in the creation of cascade features for each dataset. Note that certain features are not available for certain datasets.

User-related Features The user-related features of a given node/cascade with N users (the retweeters plus the root-user) may include the following: The DeepWalk embedding of the correspondent subgraph – a $N \times 128$ matrix; The dates of users’ account creation; The maximum value of users’ followers count; The maximum value of users’ friends count; The maximum value of users’ statuses count; The maximum value of users’ favorites count; The maximum value per user and metric across all the tweets that she has posted: listed count; impression count; reply count; quote count; like count; A Boolean identifier whether the users’ account is verified; The language of the users’ account; users’ “reaction time”, namely, the time difference between a retweet and the root-tweet.

Text-related Features In all datasets, we have included the sentiment (*label, score*) of the root-tweet – two values, either ($2, score$) for “Positive” label or ($1, score$) for “Negative” label. We use the Transformers⁵ library for the sentiment analysis. In addition, we performed topic detection for the **US Election** dataset only, following the approach of (Paraschiv et al. 2022). For this purpose, we employ a Hierarchical Dirichlet Process (HDP) (Teh et al. 2005; Newman et al. 2009), using the Tomotopy⁶ library. Then, we include the most representative topic of the root-tweet in the cascades’ feature vector based on the three topic models (the three versions of the HDP model). For the two **FakeHealth** datasets, we set as topic (per cascade) the news’ “tags”. This information is provided by the “reviews” in the FakeHealth.

Main Results

In this section, we delve into the results obtained from implementing our HyperGraphDis method across various datasets and compare them against existing state-of-the-art models. Specifically:

(1) Evaluation of HyperGraphDis on hypergraphs constructed by METIS versus on hypergraphs produced by community detection algorithms – Infomap and Louvain (Rosvall and Bergstrom 2008; Blondel et al. 2008). Both ap-

proaches are applied to users’ social networks as we presented in Figure 1. Table 1 presents the general statistics of the social networks. We experimented with varying cluster sizes (see Table 3), ranging from 20 to 5000 (US Election and Health Story), all of which were derived using the METIS for graph partitioning. For MM-COVID and Health Release, the largest partition we conducted is 1000 due to the small size of their social networks (in terms of the number of users). Generally, the community detection methods generate clusters of users that are imbalanced in size, whereas METIS clusters are generally balanced.

(2) HyperGraphDis versus state-of-the-art: (i) The vanilla version of Meta-graph; (ii) The HGFND, when applied on the hypergraphs generated by our methodology – i.e., on the same hypergraphs that the HyperGraphDis has been tested; (iii) The Cluster-GCN, when applied on the meta-graphs – i.e., the graphs of cascades that are produced by the Meta-graph approach. METIS partitioning plays an integral role in the functionality of the Cluster-GCN. So, in this experiment, METIS is applied directly to the graphs of cascades (i.e., the meta-graphs). This experiment evaluates the hypergraph construction that we propose.

Regarding the number of GNN layers:

- HyperGraphDis: one HypergraphConv layer for US Election, MM-COVID, and Health Story; two layers for the Health Release. At the end of this Section, we present an “Ablation Analysis”, where we show that more than one HypergraphConv layer does not increase the model’s utility, in general (see Figure 3).
- Meta-graph: one GCNConv layer for the US Election, three layers for the MM-COVID, two layers for the Health Story, and one for the Health Release.
- HGFND: two layers in the Hypergraph Attention network across all datasets.
- Cluster-GCN: two GCNConv layers across all datasets. The internal METIS partitioning is 10 clusters (mini-batch of 5 clusters) for the US Election and FakeHealth datasets, and 6 clusters for MM-COVID (mini-batch of 3 clusters).

The experimental results are summarized in Table 3. The direct comparison against the previously described Meta-graph, HGFND and Cluster-GCN approaches depicts that HyperGraphDis displayed superior performance in weighted F1 score (average values on five trials). In MM-COVID the method achieves an impressive weighted test F1 score of around 89.5%, outperforming the Meta-graph by approximately 4%, HGFND by 6.4%, and Cluster-GCN by 33%. Turning to the largest dataset, US Election, HyperGraphDis yielded one of the most effective results with an F1 score of 85.73%, outperforming the Meta-graph by 2.6% and Cluster-GCN by 22.3%. In the Health Story dataset, HyperGraphDis achieved 72.23% F1 score outperforming Meta-graph, HGFND, and Cluster-GCN by 3.9%, 8.3%, and 10.4%, respectively.

The overall results suggest that HyperGraphDis has broad applicability across different domains and information types, including political events and public health crises. These performance metrics highlight not only the method’s

⁵<https://huggingface.co/docs/transformers>

⁶<https://bab2min.github.io/tomotopy/v0.12.6/en/>

Methods	Graph Partitioning method: #clusters	Accuracy (%)	F1 Score (%) (weighted)	Train Time (s)	Inference Time (s)	Accuracy (%)	F1 Score (%) (weighted)	Train Time (s)	Inference Time (s)
		US Election				Health Story			
HyperGraphDis	METIS: 20	82.86±0.55	82.76±0.54	224.45	0.681	73.04±0.54	64.62±1.5	71.41	0.249
	" 50	84.66±0.44	84.47±0.44	447.27	1.59	73.74±0.64	67.83±0.95	72.81	0.214
	" 100	84.82±0.93	84.62±0.96	703.98	2.254	74.36±0.3	67.98±0.56	74.14	0.246
	" 500	85.96±0.34	85.73±0.35	1704.78	5.212	74.97±0.52	70.58±0.82	91.46	0.269
	" 1000	85.6±0.69	85.26±0.78	2037.5	6.88	75.21±0.83	72.23±0.92	94.14	0.268
	" 2000	85.35±0.83	85.01±0.9	2472.29	7.972	73.77±0.78	70.92±0.85	115.09	0.361
	" 3000	85.9±0.75	85.51±0.76	2740.06	9.202	73.23±0.99	70.94±0.7	119.81	0.356
	" 4000	85.8±0.87	85.37±0.92	2914.75	9.856	72.91±0.69	70.95±0.84	126.15	0.37
	" 5000	85.57±0.44	85.23±0.47	3036.64	10.29	73.05±0.95	70.36±0.69	125.17	0.434
	Infomap: *	78.37±0.88	77.17±0.81	113.77	0.379	75.56±0.49	70.41±0.7	57.36	0.212
Louvain: *	77.53±0.56	74.77±0.87	41.01	0.184	75.29±0.27	70.02±0.45	58.53	0.201	
Meta-graph	– –	83.31±0.74	<u>83.09±0.78</u>	5385.48	18.002	75.27±0.46	<u>68.34±0.76</u>	220.18	0.654
HGFND	METIS: 500		out-of-memory			73.46±0.92	63.0±1.71	1429.13	1.886
	" 1000		out-of-memory			73.49±0.74	63.89±0.61	2401.68	3.248
Cluster-GCN	METIS: ‡	71.62±1.28	63.36±2.39	5851.02	15.212	73.15±0.57	61.8±0.75	236.44	0.638
			MM-COVID				Health Release		
HyperGraphDis	METIS: 20	84.48±2.21	84.52±2.3	13.91	0.062	58.02±2.94	56.85±2.62	14.24	0.068
	" 50	86.63±2.43	86.79±2.43	16.54	0.066	59.04±3.09	58.91±3.13	14.27	0.07
	" 100	86.46±1.35	86.45±1.46	18.61	0.075	55.69±1.46	55.65±1.47	15.15	0.068
	" 500	88.01±0.95	88.05±0.96	22.92	0.089	61.78±1.82	61.7±1.93	17.35	0.079
	" 1000	89.5±0.74	89.48±0.66	24.99	0.094	61.68±2.12	61.56±2.25	18.69	0.084
	Infomap: *	83.31±1.44	83.17±1.39	9.65	0.053	55.79±2.16	55.71±2.04	8.8	0.054
	Louvain: *	83.81±1.5	83.79±1.57	10.17	0.056	59.14±0.98	59.07±0.96	8.98	0.047
Meta-graph	– –	85.47±1.98	<u>85.41±2.02</u>	190.24	0.643	54.42±2.07	53.66±2.11	8.59	0.054
HGFND	METIS: 500	81.38±2.86	79.79±3.43	55.72	0.075	57.06±1.13	51.36±4.55	42.02	0.078
	" 1000	84.03±3.12	83.06±3.7	81.52	0.116	56.45±1.57	<u>53.95±3.53</u>	60.01	0.102
Cluster-GCN	METIS: ‡	68.84±1.59	56.15±2.06	175.94	0.458	50.0±3.75	41.69±3.16	24.39	0.073

Table 3: HyperGraphDis versus other state-of-the-art methods: The best performance (in terms of F1 score) is highlighted in bold and is achieved by HyperGraphDis across all four datasets. The second-best performance (by one of the other three methods) is underlined for each dataset. *Infomap & Louvain: Refer to Table 1 for the number of communities produced by the community detection analysis. ‡Cluster-GCN: The internal METIS partitioning consists of 10 clusters (mini-batch of 5 clusters) for the US Election and FakeHealth datasets, and 6 clusters (mini-batch of 3 clusters) for MM-COVID.

robustness in analyzing politically charged disinformation but also its aptitude for handling imbalanced data, as reflected by the high F1 score. Finally, we found that the optimum performance on these three datasets was achieved with either METIS:500 or METIS:1000 clusters. The consistency in the optimal number of clusters may also indicate an underlying structure that is particularly amenable to hypergraph-based analyses.

For the Health Release dataset, the results were notably different. The optimal performance was observed with a minimal cluster size of 500, delivering an F1 score of 61.7%. It is important to note that this dataset is inherently less dense regarding its graph representation than the other datasets under consideration. The unique nature of the Twitter cascades in the Health Release dataset, being star graphs with a relatively small number of users (18.5K), likely contributed to this divergence. Despite the lower density and smaller user participation, HyperGraphDis still out-

performed the other three methods, underscoring its flexibility and adaptability to various types of data structures.

Beyond these primary metrics, we also observed a considerable reduction in both training and inference times when employing the hypergraph approach. This is a non-trivial benefit, especially considering the computational resources that large-scale data analysis demands. Specifically, significant improvements are noted in the processing time for both model training and inference processes. The training completion time is notably accelerated, ranging from 2.3 to 7.6 times faster than the second-best method (in terms of F1 score) across the four datasets (Meta-graph in US Election, Health Story, and MM-COVID; HGFND in Health Release). Similarly, the computational time showcases enhanced efficiency during inference, with an increase of 1.3 to 6.8 times compared to the second-best method. In summary, the results confirm that HyperGraphDis offers a compelling improvement over existing methods for identifying disinforma-

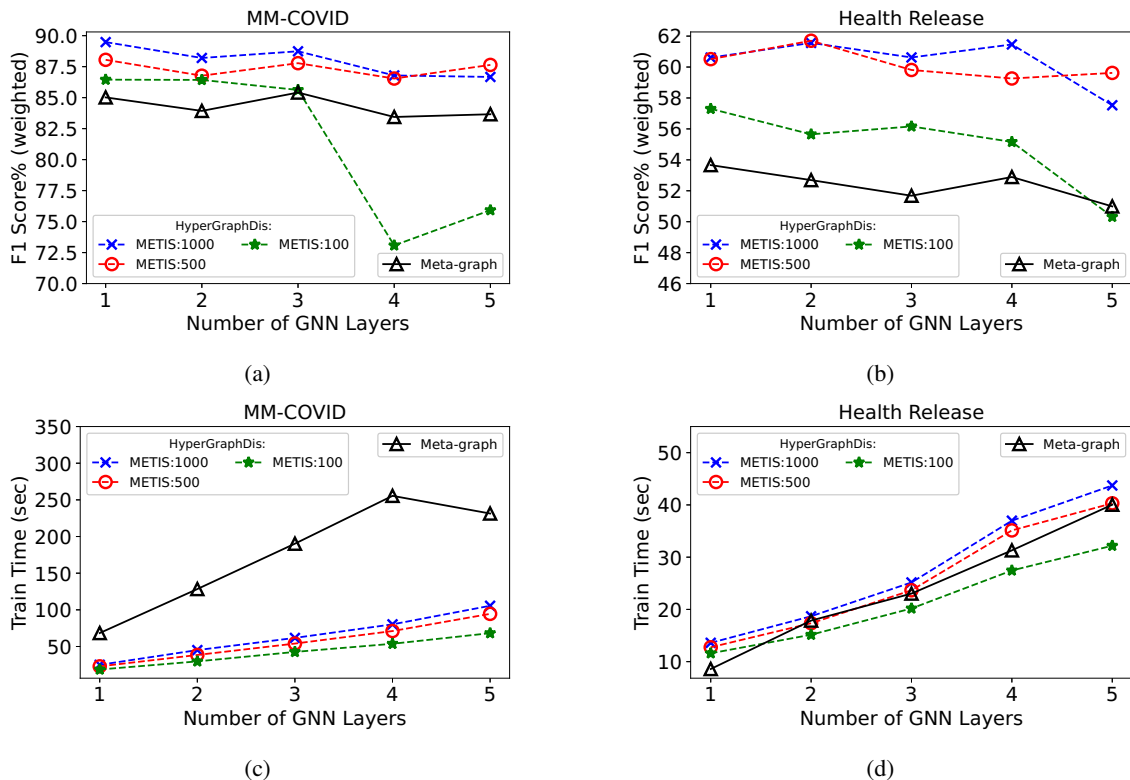


Figure 3: Ablation analysis for HyperGraphDis and Meta-graph on MM-COVID and Health Release datasets. (a) & (b): The GNN layers refer to the HypergraphConv layers in HyperGraphDis and to the GCNConv layers in Meta-graph. (c) & (d): Train process time for the models evaluated in (a) and (b). Five trials across all experiments.

tion on Twitter. Not only does it deliver more accurate and nuanced results, but it also does so more efficiently, which is an essential quality for real-time or large-scale applications.

Finally, the classification performance of HyperGraphDis on the hypergraphs that have been constructed by the Louvain or Infomap graph partitions was quite low in both accuracy and F1 score. For instance, in the US Election dataset using the Infomap partition (2074 communities), we have achieved an accuracy of 78.37% and an F1 score of 77.17%. METIS methodology involves recursively partitioning a graph into more manageable subgraphs to identify densely connected regions. The process minimizes inter-cluster connections and balances load distribution, ensuring roughly equal cluster sizes. On the other hand, although Louvain and Infomap are highly efficient in identifying communities within complex networks, the communities they produce are not well interconnected and are not balanced in size. An indicator for that is that the training time on the hypergraphs produced by the Infomap is very low for the US Election Dataset, although the number of 2047 communities is quite high (see Table 1). HyperGraphDis (Infomap) achieved a 77.17% F1 score in 113.77 seconds, whereas HyperGraphDis (METIS:500) achieved 85.73%, but the training time took 1704.78 sec.

Ablation Analysis We conducted an ablation analysis for HyperGraphDis and Meta-graph on MM-COVID and

Health Release datasets to quantify (i) the effect of the number of HypergraphConv layers in HyperGraphDis and the number of GCNConv layers in Meta-graph; (ii) the effect of the training set size on both methods. We selected the two datasets where HyperGraphDis exhibited the best and worst performance (see Table 3), and we also performed a comparative analysis against the Meta-graph (the second-best method). The ablation analysis for HyperGraphDis was performed on the hypergraphs constructed by METIS for different numbers of clusters (ranging from 100 to 1000).

Figure 3 presents the average values (from 5 trials) of the weighted test F1 score. In MM-COVID, the experiments with different numbers of HypergraphConv layers (from 1 to 5) indicate that applying several layers does not increase the utility of the model (Figures 3(a) & (b)). In Health Release, a model having two HypergraphConv layers achieved a slightly higher F1 score. Regarding the Meta-graph, a 3-layer model for MM-COVID and a 1-layer for Health Release depicts the highest performance in terms of F1 score. From Figures 3(c)&(d), we observe that in MM-COVID, the training process time of the Meta-graph is heavily affected by a large number of GCNConv layers. This illustrates the limitations of the Meta-graph approach. The meta-graph structure for MM-COVID is very dense (see Table 1); consequently, the message-passing process of a model with several GCNConv layers will not be may not be efficient in

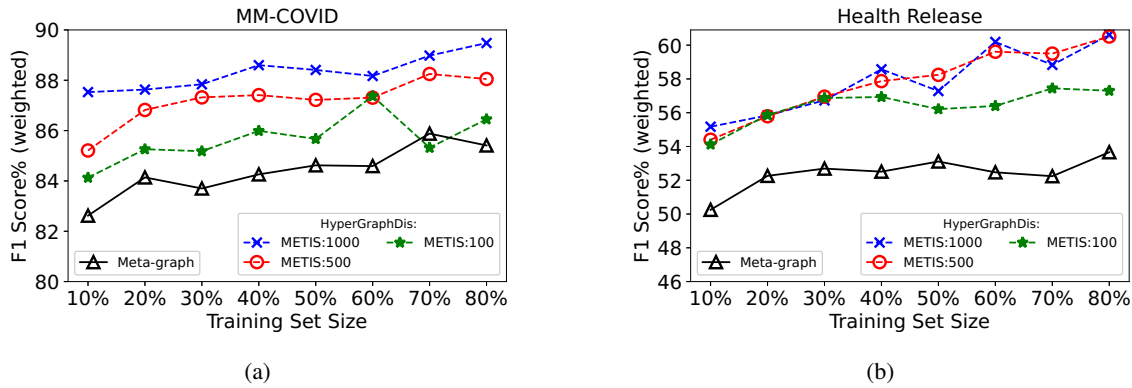


Figure 4: Further evaluation of HyperGraphDis versus Meta-graph using the optimal number of GNN layers based on the results in Figure 3(a)&(b). Namely, we use one HypergraphConv layer in MM-COVID and two layers in Health Release for the HyperGraphDis. For the Meta-graph, we use three and one GCNConv layer in MM-COVID and Health Release respectively.

terms of training process time.

We also experimented with varying training set sizes, ranging from a train-test split of 10/90 to 80/20. The difference in performance between the two methods remains approximately the same across all training sizes, indicating the superiority of HyperGraphDis (METIS:500 & 1000). Figure 4(a) shows that the HyperGraphDis (METIS:1000) in MM-COVID, a training set size of around 40% is adequate to produce a model that can achieve an F1 score close to 89%. In Health Release (Figure 4(d)), a training set size of 60% is close to optimal.

Further Discussion on HyperGraphDis Performance

In this study, we evaluated the HyperGraphDis method using the MM-COVID and Health Story datasets. Other studies that used these two datasets are the following:

MM-COVID: In (Kozik et al. 2023), a meta-analysis explored state-of-the-art Machine Learning techniques for *text-based* fake news detection. Models, including Random Forest (RF), CNN-based, LSTM-based, and BERT-based, underwent formal evaluation on benchmark datasets, such as MM-COVID. Notably, the BERT-based model achieved the highest F1 score (79.4%), while Random Forest had the lowest score (63.3%). The HyperGraphDis, with an F1 score of 89.48%, showcases a significant 10-26% improvement over conventional text-based methodologies. This highlights the importance of using social network graph data and user-related features alongside text-based features for better fake news detection.

Health Story: SOMPS-Net framework (Dhanasekaran et al. 2021), comprising two essential components: the Social Interaction Graph and the Publisher and News Statistics. Tested with the Health Story dataset, it achieved a 72.7% accuracy and 79.6% F1 score in the realm of fake news prediction. We tested HyperGraphDis using the same dataset, although we observed slightly diminished results regarding the F1 score, specifically an accuracy of 75.21% and an F1 score of 72.23%. We attribute this decline in performance to the inherent characteristics of the cascades within the Health Story dataset, which tend to be either extremely small or pre-

Method	Model	Acc(%)	F1 Score(%)	Features
MM-COVID				
Text-based ML methods*	RF	87.6	63.3	news content
	CNN	88.1	67.3	
	LSTM	90.4	75.0	
	BERT	92.5	79.4	
Health Story				
SOMPS-Net [‡]	GCN	72.7	79.6	user network, news content

Table 4: Other disinformation detection methods on MM-COVID and Health Story datasets. *(Kozik et al. 2023), [‡](Dhanasekaran et al. 2021)

dominantly structured like star-like graphs. This characteristic leads to embedding vectors with significant similarity, impacting the overall effectiveness of our method.

Limitations Although HyperGraphDis is computationally efficient (in terms of training and inference process time), making it suitable for disinformation detection in large datasets, there are still some limitations. First, a large number of tweets is needed to construct the social network – a crucial component of the HyperGraphDis pipeline (see Figure 1). The social network – the user-user interaction graph – is constructed by the interaction between the users (replies, retweets, mentions). For the produced social network to be valuable, it needs to have a rich graph structure, which heavily depends on the number of tweets collected. Secondly, in the current version of HyperGraphDis, we compute the users’ graph representation (user embeddings) on each enhanced subgraph (per cascade) by applying the DeepWalk algorithm (see Figure 2). This approach increases the overall complexity of the pipeline construction. In a future study, we aim to address this problem by incorporating "global embeddings" – i.e., the users’ graph representation will be computed once for the entire social network or in a small region around each user. Community detection or graph sampling algorithms may assist us in this direction.

Conclusion

The problem of disinformation on social media is a pressing issue with substantial social, economic, and political implications. In this paper, we introduce a novel method, HyperGraphDis, designed to combat the spread of disinformation by analyzing Twitter data through the lens of a hypergraph data structure. Hypergraphs offer several advantages over traditional graph-based methods. They provide enhanced scalability, which is crucial for efficiently analyzing vast social media datasets. We applied HyperGraphDis to diverse datasets, including health and socio-political domains to validate our approach. The method's versatility was evident, showcasing its potential across various applications. Comparative tests against state-of-the-art approaches revealed significant accuracy and F1 score improvements, crucial for imbalanced datasets common in disinformation detection. Furthermore, our method displayed improvements in computational cost and run times, reinforcing its advantages in terms of scalability and efficiency.

Ethics

This work followed the principles and guidelines on executing ethical information research and using shared data (Kennally and Dittrich 2012). The suggested methodology complies with the GDPR and ePrivacy regulations. We use Twitter datasets that have already been published. We did not use or present any identifiable user information from the datasets (e.g., Twitter user IDs). We applied text preprocessing to clean the tweets from any information that could identify specific Twitter accounts. Finally, we implemented and executed the experiments locally on our machines, so we did not upload any of the datasets to the cloud.

Acknowledgments

This work has been funded by the European Union under the projects MedDMO (Grant Agreement no. 101083756) and INCOGNITO (Grant Agreement no. 824015).

References

Ahmed, H.; Traore, I.; and Saad, S. 2017. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In Traore, I.; Woungang, I.; and Awad, A., eds., *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, 127–138. Cham: Springer International Publishing. ISBN 978-3-319-69155-8.

Bai, S.; Zhang, F.; and Torr, P. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110: 107637.

Balci, U.; Sirivianos, M.; and Blackburn, J. 2023. A Data-driven Understanding of Left-Wing Extremists on Social Media. arXiv:2307.06981.

Belova, G.; and Georgieva, G. 2018. Fake News as a Threat to National Security. *International conference KNOWLEDGE-BASED ORGANIZATION*, 24: 19–22.

Bisseling, R. 2020. *Graph matching*, 291–358. ISBN 9780198788348.

Blondel, V.; Gajardo, A.; Heymans, M.; Senellart, P.; and Van Dooren, P. 2004. A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. *SIAM Review*, 46: 647–666.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10): P10008.

Castillo, C.; Mendoza, M.; and Poblete, B. 2011. Information Credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web, WWW '11*, 675–684. New York, NY, USA: Association for Computing Machinery. ISBN 9781450306324.

Chen, C.; Wang, H.; Shapiro, M.; Xiao, Y.; Wang, F.; and Shu, K. 2022. Combating Health Misinformation in Social Media: Characterization, Detection, Intervention, and Open Issues. arXiv:2211.05289.

Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, 257–266. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.

Dai, E.; Sun, Y.; and Wang, S. 2020. Ginger Cannot Cure Cancer: Battling Fake Health News with a Comprehensive Data Repository. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1): 853–862.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*.

Dhanasekaran, P.; Srinivasan, H.; Sree, S. S.; Devi, I. S. G.; Sankar, S.; and Vijayaraghavan, V. 2021. SOMPS-Net: Attention Based Social Graph Framework for Early Detection of Fake Health News. In Xu, Y.; Wang, R.; Lord, A.; Boo, Y. L.; Nayak, R.; Zhao, Y.; and Williams, G., eds., *Data Mining*, 165–179. Singapore: Springer Singapore. ISBN 978-981-16-8531-6.

Elmas, T.; Overdorf, R.; and Aberer, K. 2023. Misleading Repurposing on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 17: 209–220.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

FORCE11. 2020. The FAIR Data principles. <https://force11.org/info/the-fair-data-principles/>.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Ghosh, S.; and Shah, C. 2018. Towards automatic fake news classification. *Proceedings of the Association for Information Science and Technology*, 55: 805–807.

Gomaa, W. H.; and Fahmy, A. A. 2013. Article: A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, 68(13): 13–18. Full text available.

Guo, H.; Cao, J.; Zhang, Y.; Guo, J.; and Li, J. 2018. Rumor Detection with Hierarchical Social Attention Network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, 943–951. New York, NY, USA: Association for Computing Machinery. ISBN 9781450360142.

Gupta, M.; Zhao, P.; and Han, J. 2012. Evaluating Event Credibility on Twitter. *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*, 153–164.

Hanley, H. W. A.; and Durumeric, Z. 2023. Sub-Standards and Mal-Practices: Misinformation's Role in Insular, Polarized, and Toxic Interactions. arXiv:2301.11486.

- Hanley, H. W. A.; Kumar, D.; and Durumeric, Z. 2023. A Golden Age: Conspiracy Theories' Relationship with Misinformation Outlets, News Media, and the Wider Internet. arXiv:2301.10880.
- Jeong, U.; Ding, K.; Cheng, L.; Guo, R.; Shu, K.; and Liu, H. 2022. Nothing Stands Alone: Relational Fake News Detection with Hypergraph Neural Networks. In *2022 IEEE International Conference on Big Data (Big Data)*, 596–605. Los Alamitos, CA, USA: IEEE Computer Society.
- Jin, Z.; Cao, J.; Guo, H.; Zhang, Y.; and Luo, J. 2017. Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs. 795–816.
- Kaliyar, R.; Goswami, A.; and Narang, P. 2021. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Application*.
- Karypis, G.; and Kumar, V. 1998. A fast and high quality multi-level scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1): 359–392.
- Kenneally, E.; and Dittrich, D. 2012. The Menlo report: Ethical principles guiding information and communication technology research. Available at SSRN 2445102.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR '17*.
- Kozik, R.; Pawlicka, A.; Pawlicki, M.; Choraś, M.; Mazurczyk, W.; and Cabaj, K. 2023. A Meta-Analysis of State-of-the-Art Automated Fake News Detection Methods. *IEEE Transactions on Computational Social Systems*, 1–11.
- Li, Y.; Jiang, B.; Shu, K.; and Liu, H. 2020. Toward A Multilingual and Multimodal Data Repository for COVID-19 Disinformation. In *2020 IEEE International Conference on Big Data (Big Data)*, 4325–4330.
- Newman, D.; Asuncion, A.; Smyth, P.; and Welling, M. 2009. Distributed Algorithms for Topic Models. *Journal of Machine Learning Research*, 10(62): 1801–1828.
- Paraschiv, M.; Salamanos, N.; Iordanou, C.; Laoutaris, N.; and Sirivianos, M. 2022. A Unified Graph-Based Approach to Disinformation Detection Using Contextual and Semantic Relations. *Proceedings of the International AAAI Conference on Web and Social Media*, 16: 747–758.
- Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining, KDD '14*, 701–710. ACM.
- Pratelli, M.; Petrocchi, M.; Saracco, F.; and De Nicola, R. 2023. Swinging in the States: Does disinformation on Twitter mirror the US presidential election system? 1395–1403.
- Rosvall, M.; and Bergstrom, C. T. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4): 1118–1123.
- Rozemberczki, B.; Kiss, O.; and Sarkar, R. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, 3125–3132. ACM.
- Salamanos, N.; Jensen, M. J.; Iordanou, C.; and Sirivianos, M. 2023. Did State-Sponsored Trolls Shape the 2016 US Presidential Election Discourse? Quantifying Influence on Twitter. In Arief, B.; Monreale, A.; Sirivianos, M.; and Li, S., eds., *Security and Privacy in Social Networks and Big Data*, 58–76. Singapore: Springer Nature Singapore. ISBN 978-981-99-5177-2.
- Serrano, M. Á.; Boguñá, M.; and Vespignani, A. 2009. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16): 6483–6488.
- Shu, K.; Cui, L.; Wang, S.; Lee, D.; and Liu, H. 2019. DEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, 395–405. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.
- Shu, K.; Mahudeswaran, D.; Wang, S.; and Liu, H. 2020a. Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1): 626–637.
- Shu, K.; Wang, S.; and Liu, H. 2019. Beyond News Contents: The Role of Social Context for Fake News Detection. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19*, 312–320. New York, NY, USA: Association for Computing Machinery. ISBN 9781450359405.
- Shu, K.; Zhou, X.; Wang, S.; Zafarani, R.; and Liu, H. 2020b. The Role of User Profiles for Fake News Detection. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '19*, 436–439. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368681.
- Tacchini, E.; Ballarin, G.; Vedova, M. L. D.; Moret, S.; and de Alfaro, L. 2017. Some Like it Hoax: Automated Fake News Detection in Social Networks. arXiv:1704.07506.
- Teh, Y. W.; Jordan, M. I.; Beal, M. J.; and M., B. D. 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. NIPS'05, 1385–1392. Advances in Neural Information Processing Systems.
- Westerman, D.; Spence, P. R.; and Van Der Heide, B. 2014. Social Media as Information Source: Recency of Updates and Credibility of Information. *J. Comp.-Med. Commun.*, 19(2): 171–183.
- Wu, L.; and Liu, H. 2018. Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18*, 637–645. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355810.
- Zager, L.; and Verghese, G. 2008. Graph similarity scoring and matching. *Applied Mathematics Letters*, 21: 86–94.
- Zhang, X.; Zhao, J.; and Lecun, Y. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Zhao, C.; Yin, Z.; Wang, H.; Zhao, X.; and Niu, D. 2020. A Novel Method for Graph Matching. 177–181.
- Zhou, X.; and Zafarani, R. 2020. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys*, 53.

AAAI ICWSM Paper Checklist

1. For most authors...
 - (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? **Yes, as this work has the purpose of combating disinformation in online media, which contributes to a reliable information ecosystem.**
 - (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? **Yes.**

- (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? **Yes.**
 - (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? **Yes, and we discuss the limitations of dataset collection of social media in a broader sense.**
 - (e) Did you describe the limitations of your work? **Yes.**
 - (f) Did you discuss any potential negative societal impacts of your work? **No, as we are solely focused on combating dis-information, utilizing trustworthy sources for data labelling, including in a real-world scenario.**
 - (g) Did you discuss any potential misuse of your work? **No.**
 - (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? **Yes.**
 - (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? **Yes.**
2. Additionally, if your study involves hypotheses testing...
- (a) Did you clearly state the assumptions underlying all theoretical results? **Yes.**
 - (b) Have you provided justifications for all theoretical results? **Yes.**
 - (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? **Yes, and we compare against previously published work.**
 - (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? **Yes, please see the technical description section for constructing the Hypergraph.**
 - (e) Did you address potential biases or limitations in your theoretical framework? **Yes, please refer to the Results section.**
 - (f) Have you related your theoretical results to the existing literature in social science? **Yes, please see the Related Works section.**
 - (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? **No.**
3. Additionally, if you are including theoretical proofs...
- (a) Did you state the full set of assumptions of all theoretical results? **N/A**
 - (b) Did you include complete proofs of all theoretical results? **N/A**
4. Additionally, if you ran machine learning experiments...
- (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **Yes, please consult the main text.**
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **Yes.**
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **No.**
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **Yes.**
 - (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? **Yes.**
 - (f) Do you discuss what is “the cost” of misclassification and fault (in)tolerance? **No.**
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
- (a) If your work uses existing assets, did you cite the creators? **Yes, all sources used, both in terms of data and previous methods are fully cited.**
 - (b) Did you mention the license of the assets? **Yes.**
 - (c) Did you include any new assets in the supplemental material or as a URL? **No.**
 - (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? **N/A**
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **N/A**
 - (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? **N/A**
 - (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? **N/A**
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
- (a) Did you include the full text of instructions given to participants and screenshots? **N/A**
 - (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? **N/A**
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **N/A**
 - (d) Did you discuss how data is stored, shared, and deidentified? **N/A**