Combating Online Misinformation by Detecting Organized Groups on Social Media

Junhao Wang, School of Computer Science

Mila/McGill University, Montreal

August, 2020

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Computer Science

Abstract

Coordinated misinformation campaign - the malicious and coordinated use of online social media for manipulation has become a pressing global problem. It aims to distort information space to confuse and distract the public, disseminate propaganda and disinformation to foster divisions, and paralyze the decision making abilities of individuals. The ultimate goals or motives of such coordinated misinformation campaigns might be hard to interpret, but their negative influence on public opinion, democracy and elections is significant.

We propose algorithmic solutions that aim to detect coordinated misinformation campaign on social media, and conduct case studies on real-world Twitter data. Specifically, we propose the Embed-Cluster-Rank framework, a three-stage algorithmic pipeline that learns low-dimensional representations for each user on a social network, clusters these representations into user clusters, and finally ranks these clusters in terms of the suspiciousness of engaging in coordinated information campaigns. We then propose three instantiations of the Embed-Cluster-Rank framework based on different embedding components - joint autoencoder, linear projection and aggregation, and tensor decomposition. We report experimental results on synthetic data, real-world Twitter data related to the 2019 Canadian Federal Election, and case studies that reveal interesting and important findings on the information landscape as well as suspicious user groups impacting the political dialog.

Abrégé

Campagne de désinformation coordonnée - l'utilisation malveillante et coordonnée des médias sociaux en ligne à des fins de manipulation est devenue un problème mondial pressant. Elle vise à déformer l'espace d'information pour semer la confusion et distraire le public, à diffuser de la propagande et de la désinformation pour favoriser les divisions, et à paralyser les capacités de prise de décision des individus. Les objectifs ou les motifs ultimes de ces campagnes de désinformation coordonnées peuvent être difficiles à interpréter, mais leur influence négative sur l'opinion publique, la démocratie et les élections est importante.

Nous proposons des solutions algorithmiques qui visent à détecter les campagnes de désinformation coordonnées sur les médias sociaux, et réalisons des études de cas sur des données Twitter réelles. Plus précisément, nous proposons le cadre Embed-Cluster-Rank, un pipeline algorithmique en trois étapes qui apprend des représentations en basse dimension pour chaque utilisateur sur un réseau social, regroupe ces représentations en grappes d'utilisateurs et classe finalement ces grappes en fonction de la suspicion de s'engager dans des campagnes d'information coordonnées. Nous proposons ensuite trois instanciations du cadre Embed-Cluster-Rank basées sur différents composants d'intégration - auto-codeur commun, projection linéaire et agrégation, et décomposition des tenseurs. Nous présentons des résultats expérimentaux sur des données synthétiques, des données Twitter réelles liées aux élections fédérales canadiennes de 2019 et des études de cas qui

révèlent des conclusions intéressantes et importantes sur le paysage de l'information ainsi que sur les groupes d'utilisateurs suspects ayant un impact sur le dialogue politique.

Acknowledgements

This work would not have been possible without support from my family. I am eternally indebted to my trusted colleague and advisor, professor Reihaneh Rabbany, who has given me ample opportunity of exploration and consistent guidance throughout my research career as a Master's student. I must also acknowledge professor Guillaume Rabusseau and professor Siamak Ravanbakhsh, who patiently explained core concepts in tensor decomposition and equivariant deep learning using their personal time. I am also thankful for my student colleagues at McGill, Mila and UBC for productive collaborations. Their camaraderie and support have made the past two years truly amazing. Lastly, I must thank the McGill Computer Science department, the Reasoning and Learning Lab, Mila, and Compute Canada for giving me consistent funding, resources, and for providing collective expertise of my fellow labmates, without any of which this work would not have been possible.

Table of Contents

Abstract i

Abrégé ii

Acknowledgements iv

1 Introduction **1**

1.1 Thesis Organization 3

2 Related Works and Background **5**

2.1 Social Networks and Social Media 5

2.2 Misinformation on Social Media 9

2.2.1 Information Operations 10

2.2.2 Detecting Misinformation Online 10

2.2.3 Organized Groups on Social Media 11

2.3 Network Anomaly Detection 12

2.4 Representation Learning for Relational Data 17

3 Methodology **19**

3.1 Problem Formulation 21

3.1.1 Simple Static Case 22

3.1.2 Complex Dynamic Case 24

3.2 Embedding Phase 26

3.2.1 Joint Autoencoder . . . 26
3.2.2 Linear Projection and Aggregation (SCG) . . . 28
3.2.3 Coupled Tensor Factorization . . . 29
3.3 Clustering and Ranking Phase . . . 31

4 Results and Discussions **34**
4.1 Joint Autoencoder . . . 34
4.1.1 Data Collection . . . 35
4.1.2 Hyper-Parameter Tuning . . . 35
4.1.3 Results . . . 37
4.2 Linear Projection and Aggregation (SCG) . . . 39
4.2.1 Validation on Synthetic Data . . . 40
4.2.2 Results on Real-World Data . . . 44
4.3 Coupled Tensor Factorization . . . 53
4.3.1 Validation on Synthetic Data . . . 53
4.3.2 Results on Real-World Data . . . 54

5 Conclusion **59**
5.1 Directions for Future Work . . . 60

Chapter 1

Introduction

The earliest form of social networks emerged shortly after the invention of the World Wide Web, such as Theglobe.com (1995), Geocities (1994) and Tripod.com (1995), where people are brought together through chatrooms and encouraged to share personal information via personal web-pages - the earliest form of user profiles. In the next two decades, waves of social network sites popped up with significant improvements on suggesting, managing and expanding friends list. In 2004, Facebook was founded and soon became the largest social network platform in the world. Social media has since then become an indispensable part of social life for most people.

The meteoric rise of social media fundamentally impacted people's everyday life. An increasing number of individuals are relying on social media to fulfill various personal and social needs. More relevant to our work, increasing amount of people around the world utilise social networking sites as an alternative news source [89]. A 2015 study estimated 63% of U.S users of Facebook or Twitter consider these networks to be their main source of news, especially political ones [7]. Such widespread usage of social media for obtaining news contributed to new forms of abusive communication - polarized online debate, political violence and abuse, misinformation - false or inaccurate information.

More specifically, coordinated misinformation campaign - the malicious and coordinated use of online social media for manipulation has become a pressing global problem. In fact, the most important geopolitical events of the 21st century (the rise of ISIS, the Russian occupation of Crimea, and the election of President Trump) all involve heavy use of social media for propaganda and misinformation purposes. The impact of such serious and malicious mass manipulation through social media incurs exponential marginal social cost when information circulates rapidly through social networks. Such manipulation of online discourse through social media is a pressing global concern [87]. Recently, the Special Counsel for the U.S. Department of Justice published their investigation into Russian "Active Measures" social media campaign, which confirmed an *organized* attempt at the state level to sow discord into the U.S. political system through social media [63]. As an example, Twitter reported possible engagement of 1.4 billion users with the suspected "trolls" from the Russian government funded Internet Research Agency (IRC) [69], and it is believed that this interference has swayed the 2016 US Presidential Election [5]. Such activities aim to distort information space to confuse and distract voters, disseminate propaganda and disinformation to foster divisions, and paralyze the decision making abilities of individuals [100]. The ultimate goals or motives of these operations might be hard to interpret, but their effect on public opinion, democracy and elections is clear [9,57]. The severity and scale of such operations motivated the social media giants like Twitter and Facebook to update their site policies [25,76]. These updated policies aim at tackling *Information Operations* - the suit of methods used to influence others through the dissemination of propaganda and disinformation [76,100], and *Coordinated Inauthentic Activities* - groups working together to mislead people about who they are and what they are doing [25].

With the popularization of social media as a news source, organized misinformation is becoming one of the most significant problems brought by technology advances in 21^{st} century. How can we monitor the information space proactively, identify such coordi-

nated activities at an early stage, to ensure a healthy democratic society? Our work focuses on fighting against these organized misinformation campaigns on social media - deliberate and strategic attempt to spread misinformation. Specifically, we present Embed-Cluster-Rank framework as a solution to this problem. It consists of modular components to study the activity in complex social media space and identify groups that are impacting the information space in an organized manner. In extreme cases, bots generated from the same script behave in an almost identical or highly correlated manner [13], and trolls or sockpuppets, who are being operated by the same person behind the scenes, exhibit lock-step behavior [48]. More generally, Embed-Cluster-Rank framework detects organized groups, members of which *amplify each others' voice and boost each others' influence*, unlike what is the norm among typical users. These are suspicious groups that need to be *further investigated by a human* as detecting trolls is not a trivial task [71] and political campaigns or activist groups might exhibit similar coordinated behaviour. Our framework makes this subsequent investigation efficient by providing a group level summarization and characterization, organized as the following.

1.1 Thesis Organization

This thesis is organized as follows:

- **Chapter 2: Related Works and Background**

  This chapter provides a high-level overview and context of our work. First, it introduces the history of the emergence of social networks, its implication for how people consume news and make sense of reality, and the looming problem of online misinformation. Next, it formalizes how data on social networks can be represented as networks and summarizes existing approaches for combating online misinformation. Finally, it zooms in on our Embed-Cluster-Rank approach to coordinated

misinformation, and introduce related literature on network anomaly detection and representation learning for relational data.

- **Chapter 3: Methodology**

  This chapter provides a formal formulation of the problem of detecting coordinated misinformation campaigns on social media, as well as details of our algorithmic contributions in the embedding phase of the Embed-Cluster-Rank pipeline. It first provides core assumptions behind the data generating process of social media data using examples of Twitter. Then it details our main contribution in the embedding phase - three embedding methods to learn low-dimensional representations from social networks. These are joint autoencoder, linear projection and aggregation, and tensor decomposition. Finally, it summarizes the clustering and ranking phase.

- **Chapter 4: Results and Discussions**

  This chapter goes into details of the results from each of our embedding method along with corresponding clustering and ranking components. It covers the experimental results on synthetic data, real-world Twitter data related to the 2019 Canadian Federal Election, and also case studies that reveal interesting and important findings on the information landscape and potential foreign interference on Canadian election.

- **Chapter 5: Conclusion**

  This chapter wraps up by summarizing the core contribution of our work - the Embed-Cluster-Rank framework, and propose interesting future research directions such as automating more parts of misinformation detection using natural language processing, as well as merging the three-stage processes into one end-to-end differentiable pipeline.

Chapter 2

Related Works and Background

In this section, we first introduce social networks, misinformation on social network, and then introduce approaches for combating coordinated misinformation on social media in the context of network anomaly detection. Finally, we cover works related to the specific embedding approaches adopted in this work.

2.1 Social Networks and Social Media

A social network is the various connections people form with each other via a social network platform, and social media is the content shared by people on the platform. In this work we use both terms inter-changeably since content and connection on social network platforms are indispensable of each other: social connections create the context for social media and content shared by people on the network evolves the network itself. Formally we represent social media as temporal heterogeneous networks that consists of entities and relations of different types, evolving through time. For example, a static slice of this network on Twitter typically contains interactions among users in terms of following and interactions between users and hashtags. To build up to the definition of such a complex data structure, we start by introducing the basics of graph.

Graph

In graph theory, a graph $G(\mathcal{V}, \mathcal{E})$ is a set of objects in which some pairs of objects are "related". The objects are called vertices or nodes - $\mathcal{V}$, and the relations are called edges - $\mathcal{E}$. In the case of social networks, nodes typically refer to users and edges refer to relations among them. Edges can be directed or undirected. For example, a relation such as being friends on Facebook is undirected since A being friend of B implies B being a friend of A and vice versa. On the other hand, a relation such as following an account on Twitter is directed since A following B does not imply the reverse to be true. The former type of graph is called undirected graph, and the latter is called directed graph. Graphs are typically represented by an adjacency matrix, a 2-D matrix $\mathbf{A} \in \{0, 1\}^{n \times m}$ where the indices of rows and columns represent left-hand side and right-hand side nodes respectively from sets of cardinality n and m, and the value on the i^{th} row and j^{th} column represent the edge going from node i to node j. The left-hand side and right-hand side nodes could be either elements from the same set or different sets. In the latter case, if the sets are disjoint we refer to the graph as a bipartite graph since the edges all go from one set to a different set of elements. In the simple scenario, edge value is either 1, indicating existence of an edge, or 0, indicating the lack thereof. In more complex scenario, it can take on real values, indicating various properties of the relation between node i and j.

Multi-relational Graph

A multi-relational graph has multiple types of edges and is typically represented by a tensor $\mathbf{T} \in \{0, 1\}^{r \times n \times m}$, with r types of edges, n left-hand side nodes and m right-hand side nodes. $\mathbf{T}_{R,:,:}$ is a slice of $\mathbf{T}$ at edge type $R \in \{1 \dots r\}$, and stores the adjacency matrix of edge type R. In more complex cases, $\mathbf{T}$ could take on real values.

As a reminder, a tensor is a multidimensional array [42]. First order tensor is a vector, second order tensor is a matrix, and tensor with three or higher orders are called

higher order tensor. Tensor $\mathbf{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ having n modes, is a geometric object that can be thought of as higher order generalizations of matrices or as multidimensional arrays. Each element of $\mathbf{T}$ can be indexed by a tuple $(i_1 \in [d_1], \ldots, i_n \in [d_n])$. $\mathbf{T}$ can also be reshaped into vectors or matrices. Vectors and matrices can also be reshaped back into tensors. The j^{th}-mode matricization of $\mathbf{T}$ is a matrix $\mathbf{X} \in \mathbb{R}^{d_j \times \prod_{k \in [1,\ldots,j-1,j+1,\ldots,n]} d_k}$. We can take j^{th} mode product between tensor $\mathbf{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ and matrix $\mathbf{X} \in \mathbb{R}^{d_j \times m}$, denoted $\mathbf{T} \times_j \mathbf{X}$. The resulting tensor of shape $\mathbb{R}^{d_1 \times \ldots d_{j-1} \times m \times d_{j+1} \times \cdots \times d_n}$. This is equivalent to taking a normal matrix multiplication between the transpose of j^{th} mode matricization of $\mathbf{T}$ and $\mathbf{X}$, and then reshape the resulting matrix back to tensor form.

Heterogeneous Graph

A heterogeneous graph not only has multiple edge types but also multiple node types and can be represented by either a tensor or a set of coupled matrix tensors. If represented by a tensor, the left-hand / right-hand side nodes include all types of left-hand / right-hand side nodes and edge types include all types across interactions among different node types, therefore leading to significantly larger and sparser tensor than coupled matrix tensors representation. On the other hand, coupled matrix tensors representation represent each interaction type as a matrix or tensor, and have a one-to-one correspondence between corresponding rows / columns to match the nodes that refer to same entities. To clarify with an example of Twitter data, we have two types of nodes (users, hashtags) and two types of relations (user-follow-user, user-use-hashtag), then the coupled matrix tensors representation of the data contain two matrices $\mathbf{A} \in \{0,1\}^{n \times n}$, $\mathbf{X} \in \{0,1\}^{n \times d}$ with n users and d hashtags, such that each row in both matrix refer to the same user, thus "coupled". On the other hand, representing the data as one large tensor would result in $\mathbf{T} \in \{0,1\}^{2 \times (n+d) \times (n+d)}$ with the two slices representing user-follow-user and user-use-hashtag relations, thus being significantly sparser. While the latter representation is fre-

quently used in knowledge graph literature due to lack of clear node type definitions, this is not the case in social network data. Social network data has clear definition of node types, such as users and hashtags. Therefore, we use coupled matrix tensors representation in this work. Extending binary values to real values is straightforward.

Coupled Matrix Tensor

A set of coupled matrix tensors are induced from multiple typed sets of nodes and typed sets of relations. The relation could be either pairwise (between two nodes) or higher-order, acting on more than two nodes. Pairwise relation that act on two sets of nodes induces a matrix. Higher-order relation induces a tensor. Relation could act on the same set, for example user-follow-user relation is a pairwise relation from the user set to the user set. It could also act on different sets, for example user-use-hashtag relation is from the user set to the hashtag set. In the higher-order case, user-use-hashtag-at-time relation acts on the set of timestamps, users and hashtags, indicating at what time user uses a hashtag. In the simple case of only pairwise relations, coupled matrix tensors are a set of adjacency matrices.

Temporal Heterogeneous Graph

A temporal heterogeneous graph represents an evolving set of typed relations acting on a set of typed nodes or entities. In the most general sense, more typed sets of relations or entities can be added, and the cardinality of each typed set of entities or relations can grow. In this work, we limit to the case where the no new types are created, typed sets of entities are fixed and the only changes through time is typed relations by being created and deleted among existing entities. Furthermore, instead of tracking real-time chain of relation addition and deletion, we take an aggregate view of the data as slices through time at different timestamp. Therefore we add a new typed entity - timestamps, and re-

formulate pairwise relations between two entity sets as higher-order relations among the two entities as well as timestamp at which the interaction was observed. This is represented using a set of coupled matrix tensors introduced above. For example, an evolving Twitter dataset that involves user-follow-user, user-use-hashtag-at-time relations can be represented by user-user adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ coupled with time-user-hashtag tensor $\mathbf{T} \in \{0,1\}^{t \times n \times d}$ on the user axis, with n users, t timestamps and d hashtags. Extending binary values to real values is straightforward.

2.2 Misinformation on Social Media

Misinformation is false or inaccurate information that is communicated whether the intention is to deceive or not. Propagation of misinformation on social media tend to elicit fear and suspicion among a population [14]. Disinformation is one of the most important types of misinformation. It has an element of deliberate deception, such as malicious hoax (fabricated falsehood disguised as truth), phishing (fraudulent attempt to obtain sensitive information) and online propaganda (persuasion to further hidden agenda). Such malicious manipulation are usually highly organized. Facebook uses the term Coordinated Inauthentic Behaviour and Twitter uses the term Information Operations to refer to such organized disinformation efforts. Furthermore, these coordinated disinformation on social media are often sponsored by state actors, and caused significant damage to democracy through erosion of social trust. A notable example was Russian government funded Internet Research Agency (IRC) that interfered and swayed the 2016 US Presidential Election. Clearly, combating online misinformation, especially organized disinformation, before it causes further widespread damage is an urgent and important problem of this century.

2.2.1 Information Operations

Information operations are activities that attempt to undermine information systems and manipulate public discourse [100]. They have existed for centuries under different names such as 'information warfares' [4], 'information dominance' [3], 'psychological operations' [54]. Information operations operate on many forms: shaping strategic narrative [61], orchestrating and sustaining online collaborative work [67,88], distorting public political sentiment [40]. The popularization of social networks such as Facebook and Twitter, lowers the barrier for implementing information operations and makes it more effective [95]. Specific strategies involved in information operations have attracted abundant research attention in recent years: controlled spread of misinformation [32], deployment of botnet [91], online trolls [23]. One fruitful line of research in this regard applies network science to understand the online information space and dynamics, and aims to detect network anomalies that engage in malicious activities [1, 77]. Our work follows this line of research and aims at creating tools to aid public understanding of information space and defend against malicious actors.

2.2.2 Detecting Misinformation Online

Many recent works [5, 62, 100] analyze the vulnerabilities of social media to information operations and coordinated inauthentic activities, and relate them to the clustering of politicized online information spaces. This phenomenon, defined as "echo-chambers", describes the gathering of like-minded individuals on online communities. As illustrated by Marwick *et al* [57], the defiance toward traditional media from part of the population leads to the emergence of alternative (possibly biased or fake) news sources. Bovet *et al.* [9] showed that Twitter trolls tend to form small, politically biased groups that propagate misleading information to normal users. Stewart *et al* [90] identified trolls as polarising elements of echo-chambers, distorting the information space. Most past works on

online misinformation detection are largely limited to classifying user-generated content or users, based on their activities [84,85,105,106].

Unlike these supervised techniques, we take a novel unsupervised approach that jointly analyzes content and user connections to detect organized groups, inspired by successful application of unsupervised techniques in anomaly and fraud detection settings, for example, to detect fake reviewers posted to artificially boost product ratings on E-commerce sites [34]. The lockstep behavior exhibited by agents who engage in information operations induces dense subgraphs within the larger graph that represents the connections among users or between users and content they engage with, hence this is related to general dense block and anomaly detection algorithms, introduced in the next section.

2.2.3 Organized Groups on Social Media

Both U.S. Department of Justice's investigation into IRA and major tech companies' updated site policy reveal that hostile mass manipulation of online discourse on social media often involves what we call Polluting Group - small set of densely connected social media accounts that aim to increase their influence through following each other and broadcasting similar set of messages. Accounts in organized groups often form subgraphs of much higher empirical edge probability than the background for both the followership network and the bipartite network between accounts and messages, which can be captured by artifacts such as hashtags on Twitter, keywords on Facebook, etc. This is similar to strategies used by fake reviewers to artificially boost product ratings on E-commerce sites [34]. Furthermore, these organized groups are much smaller in size than naturally formed network communities, and our interest is to recover exactly these **tiny clusters**, instead of the global community structure of the network. We adopt a similar notion of tiny clusters as in [64], which refer to clusters of size $O(n^{\epsilon})$ where n is the size of the graph

and $\epsilon > 0$. To build up to the definition of tiny dense subgraph, we will introduce basic definition of subgraph and its connection in the broader context of network anomaly detection literature in the next section.

2.3 Network Anomaly Detection

The original definition of an anomaly or outlier is given by Douglas M. Hawkins: "An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism." [31]. Traditional anomaly detection techniques aim to sport anomalies in unstructured set of multi-dimensional points. Network anomaly detection takes into regard the relational structure of these data points, and identify anomalies on a network. To illustrate the distinction of traditional versus network anomaly detection, we use the example of Twitter: the extent a user is suspicious of engaging in disinformation depends on her/his usage of a set of hashtags as well as how other users use the same set of hashtags, and how users relate to each other. We can not detect anomalies here by only looking at a single user without considering the inter-dependencies among users and between users and hashtags.

Broadly speaking, different network anomaly detection techniques can be applied to different types of graphs such as unsupervised / (semi-)supervised, static / dynamic, and attributed / plain graphs, and return different graph structures such as nodes, edges and subgraphs. We first focus on unsupervised detection of anomalous subgraphs on static, attributed graphs. As network anomaly detection is application specific, there is no universal criteria to evaluate the quality of detected anomalies. Because we are interested in tightly connected social media groups that broadcast similar content, thus suspicious of engaging in coordinated misinformation, we focus on empirical evaluations on real world data such as rate of user suspension, as well as on synthetic data by inspecting

subgraph density. We are also interested in qualitative evaluations such as real-world data case studies.

In the context of social networks, anomalies could mean fraudulent groups that boost each other's influence by forming highly interconnected sub-regions [68], important and influential individuals [16, 55, 79], malicious and predatory activities [12, 22]. The data used for these anomaly detection could be static or dynamic, labelled or unlabelled [77]. Works directly related to applying social network anomaly detection to detect information operations or influence campaigns have been a recent establishment [56, 60].

While anomaly detection is a well researched problem, many techniques fail to be applicable on the extremely sparse graphs with a large set of nodes which characterizes most modern social networks. ODDBALL [2] is a classical approach that defines several metrics surrounding the density, weight, rank and eigenvalues associated with anomalous subgraphs, and computes these measures to identify anomalous blocks. Another approach is presented in [70] which detects persistent patterns, called EigenSpokes, which are found in large sparse social graphs. By plotting the singular vectors of these graph against each other (called EE-plots), clear, separate lines or spokes that often align with axes (EigenSpokes) are detected. EE-plots are indicative of fundamental clustering structures within these graphs. Alternatively, matrix factorization approaches have been extremely prolific in anomaly detection literature. For example, Tong and Lin [92] adapt non-negative matrix factorization (NMF) by enforcing constraints to identify anomalies in the residual graph after typical factorization, thereby capturing anomalies in the original whole graph. In line with recent advances involving deep learning, a major contribution in anomaly detection follows from DeepFD, an architecture developed by Wang *et al* [96] based on graph embeddings of both attributed and topological graphs. Their work preserves graph structure and user behavior in order to improve adversarial robustness to fraudsters within networks of interest. With the popularization of graph neural networks,

graph convolutional autoencoder that encodes and decodes both adjacency and attribute matrix is used to rank nodes in terms of anomaly score, indicated by node reconstruction error [19].

A distinct type of network anomaly is especially relevant to detecting coordinated misinformation campaign: lockstep behavior - groups of users acting together. Such behavior induces dense blocks on data, and thus dense block detection methods have been designed for this type of anomaly and fraud detection [80–82]. Anomalous agents are usually camouflaged within the larger modular structure of the graph and thus detecting them can be a difficult task. Dense subgraph discovery is well-studied and there are different types of dense subgraphs, such as clique, quasi-clique, K-core, K-plex, Kd-clique, and K-club. Due to the combinatorial nature of graphs, detecting these structures is challenging, for instance finding exact cliques and quasi-cliques is shown to be NP-hard [51]. Classical solutions for the clique problem can be categorized as exact enumerations [72], fast heuristic enumerations [47] and bounded approximation algorithms [11], most of which have runtime at least polynomial to the size of graph. Fraudar [34] is a notable example of scalable dense subgraph detection methods that finds subgraphs with large average degree in the context of fraud/anomaly detection. Our work takes a similar notion of dense subgraph but focuses on finding subgraphs with large empirical edge probability in multiple graphs simultaneously: user connection graph and user content bipartite graph. Various extensions of dense subgraph detection include dense sub-tensor detection [81,82], online dense sub-tensor detection [83], hierarchical dense subgraph detection [104]. These methods are defined in a single mode whereas our method detects coupled blocks which enforce dense substructures in coupled matrices/graphs as discussed later in detail.

Dense block/subgraph detection for detecting anomalies on networks is also related to community detection and tiny cluster detection on graphs. It is well-known that tra-

ditional community detection techniques such as modularity optimization fail to identify clusters smaller than a scale [24]. This resolution limit depends on the size of the network and the interconnectedness of the clusters. Few works try to discover clusters of small size in graphs [52, 101]. Notably, pcv method [64] considers bipartite stochastic block models and formally defines tiny cluster to be clusters of size $O(n^{\epsilon}), \epsilon > 0$ where n is the size of the graph (right-side nodes), and finds tiny clusters with theoretical guarantees. Our work takes a similar notion of tiny cluster, but in a more general case of coupled matrices as explained later. This combining of the different sources of information is proven to be a necessity for better recovery of community structure in contextual stochastic block models [18], which applies non-rigorous cavity method from statistical physics to prove the information theoretical necessity of for better recovery of community structure in contextual stochastic block models and utilizes both the graph and node attributes for discovering them. This is motivated by prior work in [18], which applies cavity method from statistical physics to prove the information theoretical necessity of combining the different sources of information for better recovery of community structure in contextual stochastic block models. M-Zoom [81] is a classical approach to this problem, which iteratively finds and removes dense blocks to prevent duplicate block querying. Shin *et al* [82] take an offline approach to the task in D-Cube, facilitating distributed, fast detection of dense blocks with provable guarantees on the accuracy of identifying blocks.

Dense subgraph detection is closely related to graph clustering or community detection, which identifies clusters of densely connected nodes [38,102]. We want to emphasize dense block/subgraph detection for detecting anomalies on networks is related to but different from discovering intrinsic structure of the network. Community detection in networks is a task that involves finding the underlying modular structure of the graphs. In particular, the problem involves finding a grouping of nodes that attempt to ascertain an underlying clustered, segmented and relatively dense structure within a graph.

Two widely-used community detection algorithms include Louvain [8], based on modularity optimization and Infomap [75], based on information compression. Traditional approaches like modularity maximization, which measures the number of edges in identified communities in relation to the expected number of edges in an unorganized graph, suffer from small-resolution communities and do not scale well to contemporary social networks [49]. More recent research has identified the need for combining both the underlying structure of the nodes within the network as well as their inherent attributes [53], providing motivation for . Liu *et al* [53] adopt the paradigm that a network graph results from interactions among nodes, and introduce the idea of content and influence propagation via random walks, analyzing the stable structure of this dynamical system to identify communities. Jia *et al* [37] enhance node attributes by running k-nearest neighbors on the graph a priori and append this information to node representations, demonstrating that this alleviates graph sparsity issues and improves performance of community identification algorithms. Thus, our formulation in is a natural extension of this idea. Most recently, graph neural networks (GNNs) extend the convolutional neural network framework to graph structures by leveraging affine transformations of graph operators and node-wise or edge-wise activation functions. Chen *et al* [15] introduce a new family of GNNs which rely on a non-backtracking graph operator defined on the line graph of edge adjacencies, facilitating scalable inference of communities on large, sparse graphs. A separate line of work aims to detect tiny communities. Neumann proposed an elegant and simple algorithm for provably finding community of size $O(n^{\epsilon})$ in a bipartite graph generated by bipartite stochastic blockmodel, by first clustering left-side nodes based on similarity of their neighbors, and then recover right-side partition based on degree thresholding [64].

2.4 Representation Learning for Relational Data

With the popularization of representation learning on graphs, new techniques have been developed to learn graph structures and detect network anomalies. Network embedding techniques aim to map nodes or subgraphs onto Euclidean space through possibly learned functions on graphs. The most notable graph embedding techniques are unsupervised GraphSAGE [30], node2vec [28] and attri2vec [103]. We show in our experiments that these methods fail to recover tiny clusters effectively. This type of learning is inherently difficult as graphs are combinatorial structures with discretized nodes and edges. Thus, conventional learning modalities like neural networks often fail for these learning tasks as they rely on continuous representations of data. In particular, unsupervised graph representation learning is interesting as most graphs are not fully specified; connections between nodes within our data are often hidden or unknown, particularly in large scale graph structures such as social media networks. To this end, Kipf and Welling [41] develop the variational graph autoencoder that uses a graph convolutional network as an encoder which parameterizes a Gaussian latent distribution. A decoder network then reconstructs the full graph, and the authors demonstrate that such reconstruction from the latent embeddings predicts unseen or masked links in the original network with good accuracy. Moreover, their framework can be trained end to end through classical variational inference. More recently, Hamilton *et al.* [30] demonstrate greatly improved performance through their more general GraphSAGE approach. GraphSAGE is able to perform inductive learning and generate node embeddings for previously unseen graphs. Critically, even in settings where node attributes are not made explicitly available, the GraphSAGE is extendable by computing additional node features such as degree from the network topology and substituting these as the node attributes.

In the case of heterogeneous networks where there are multiple types of nodes and edges, coupled matrix tensor decomposition can be used to learn useful representations

for each node. The most popular tensor decompositions are: CP [39,45,46] and Tucker [94] decompositions which can be considered as higher-order generalizations of matrix singular value decomposition (SVD) and principle component analysis (PCA). CP decomposes a tensor as outer product of factor matrices for each mode, and Tucker decomposes a tensor as a core tensor and factor matrices for each mode. Besides CP and Tucker, more recent establishments include Tensor Train decomposition [66], coupled matrix tensor decomposition (CMTF) and multi-way clustering on graphs [6], and structured data fusion [86] which inspired our work. Particularly, which made optimizing SCG-map's objective possible with the size of our real-world data, builds upon stochastic optimization methods [43] for generalized CP decomposition [33].

Chapter 3

Methodology

We propose a unified three-stage pipeline for identifying organized groups on social media, consisting of embedding, clustering and ranking, shown in Figure 3.1. The embedding phase learns low-dimensional representations of social media users. The clustering phase groups these representations into user clusters. The ranking phase ranks the most suspicious user groups that likely engages in coordinated information campaigns to help human experts detect strategic online misinformation. Our key insight is that, by representing social media users in a low-dimensional space, traditional clustering algorithms can be exploited to discover similarly behaving groups. Furthermore, groups that not

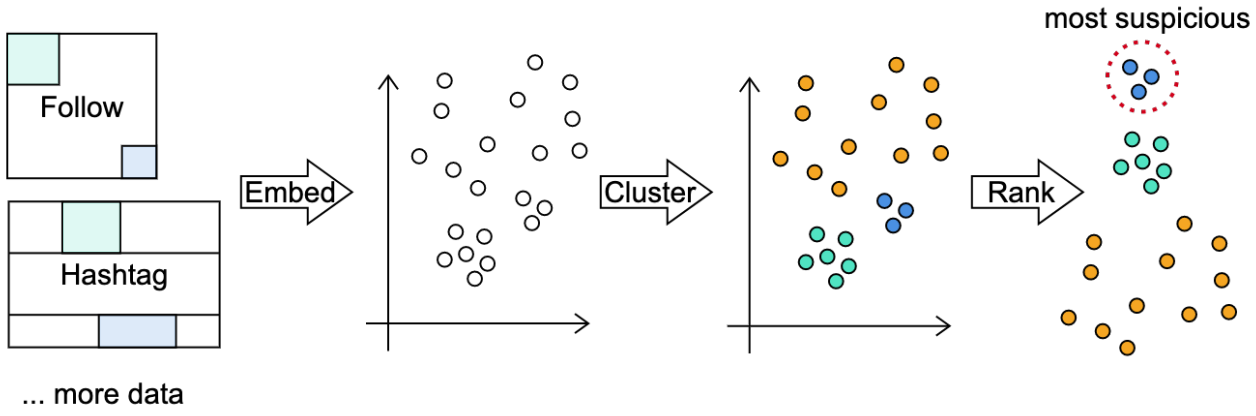

Figure 3.1: Embed-Cluster-Rank pipeline

only behave similarly, but also have much denser connections within the group relative to the background, are suspicious ones needed to be vetted by humans for whether they engage in coordinated misinformation campaign.

More specifically, we represent social media contents and connections as temporal heterogeneous networks that consists of entities and relations of different types, evolving through time. For example, typical Twitter dataset contains follower and retweet network, as well as how users engage with contents which can be abstracted as a bipartite graph between users and hashtags. Thus this heterogeneous network contains entities such as users and hashtags, and relations such as follow, retweet and hashtag-usage. Furthermore, there could be multiple snapshots of this network through time.

First, we apply dimensionality reduction techniques to obtain low-dimensional embeddings for entities to be investigated, which in most cases are social media users. The literature on dimensionality reduction is vast, and one of our core contributions is adapting, modifying and scaling up techniques from very different spectrum to solve a high-impact social problem. Next, once we have low-dimensional embeddings for users, we apply clustering algorithm to further break users into groups. In principle, any clustering algorithm can be used in this step, but we empirically identified density-based clustering to yield best detection performance. Finally, we designed simple and interpretable heuristics to rank the resulting clusters to surface up suspicious user groups for human to verify whether they are involved in misinformation campaigns.

In the following subsections, we will introduce the problem formally, and focus on explaining in depth our contributions in step one of the pipeline while covering the basics of step two and three. We propose three methods: joint autoencoder, linear projection and aggregation, and coupled tensor factorization. The first two handle simple static dataset, and the third handles both simple static and complex dynamic dataset as explained in the following section.

3.1 Problem Formulation

We define Organized Group to be *a small set of densely connected social media accounts that aim to increase their influence which are related to each other and broadcast a similar set of messages*. The first criteria regarding the small size of the network is motivated by the report of U.S. Department of Justice's investigation into Russian interference in the 2016 U.S. Presidential election: "Dozens of IRA employees were responsible for operating accounts and personas on different U.S. social media platforms; A number of IRA employees assigned to the Translator Department served as Twitter specialists; IRA specialists operated certain Twitter accounts to create individual U.S. personas" [63]. Evidence suggests that the size of the organized group is small and operationalized, and the total number of employees operating the social media accounts is limited by hiring capacity of the underlying organization. The second criteria regarding mutual influence boosting through following or other ways of relating to each other on social media is motivated by the investigation lead by the Special Counsel for the United States Department of Justice into Russian "Active Measures" Social Media Campaign. Through this campaign, Internet Research Agency, LLC (IRA), a Russian State sponsored organization was capable of reaching millions of U.S. citizens through their social media accounts on Facebook, Instagram, Tumblr, YouTube, and Twitter, by the end of the 2016 U.S. election [63]. More specifically, IRA created inauthentic social media accounts operated by a small team of employees as well as automated bots starting as early as 2014, in the names of U.S. citizens, fictitious U.S. organizations and grassroots groups, in order to garner followers and influence in online discourse and broadcast messages with hidden political agenda. Employee-operated IRA social media accounts attracted massive followers: "United Muslims of America" Facebook group had over 300,000 followers; @jenn_abrams - a Twitter account claiming to be a Virginian Trump supporter had over 70,000 followers. Bot-operated net-

work of accounts also gained considerable influence during the election (approximately 1.4 million people on Twitter) [63].

For clarity, we will introduce the problem in its simple static case with a Twitter dataset that only contains follower graph and user hashtag graph, and then introduce more general formulation of complex dynamic case using a temporal Twitter dataset that further contains retweet and mention graphs with multiple snapshots through time. The task of interest is to identify organized groups from both these datasets in an interpretable and scalable fashion. Though we use Twitter datasets to introduce the problem, it easily extends to other social platforms. For notation, we use Boolean matrices and tensors to represent graphs, temporal graphs and affiliation matrices.

3.1.1 Simple Static Case

Given a simple static Twitter dataset of follower graph $\mathbf{A} \in \{0,1\}^{n \times n}$ and user hashtag bipartite graph $\mathbf{X} \in \{0,1\}^{n \times d}$ with the set of users $|\mathcal{U}| = n$ and set of hashtags $|\mathcal{H}| = d$, we consider organized groups as small sets of accounts that aim to increase their influence through following each other and broadcasting similar messages, mostly through using distinct sets of hashtags frequently. The connections among these users form subgraphs of $\mathbf{A}$ with much higher edge probability than the background. Similar subgraphs are induced by these groups on $\mathbf{X}$ with corresponding sets of hashtags. Furthermore, these organized groups are usually much smaller in size than naturally formed network communities, and our interest is to recover these tiny clusters, instead of the global community structure of the network.

We formally define organized groups on $\mathbf{A}$ and $\mathbf{X}$ by extending Latent Block Model [27], a probabilistic framework for co-clustering, to handle multiple sources of data. We first assume both $\mathbf{A}$ and $\mathbf{X}$ can be modelled by grids of Bernoulli random variables, whose distributions are specified through latent groups on the rows and columns of $\mathbf{A}$ and $\mathbf{X}$.

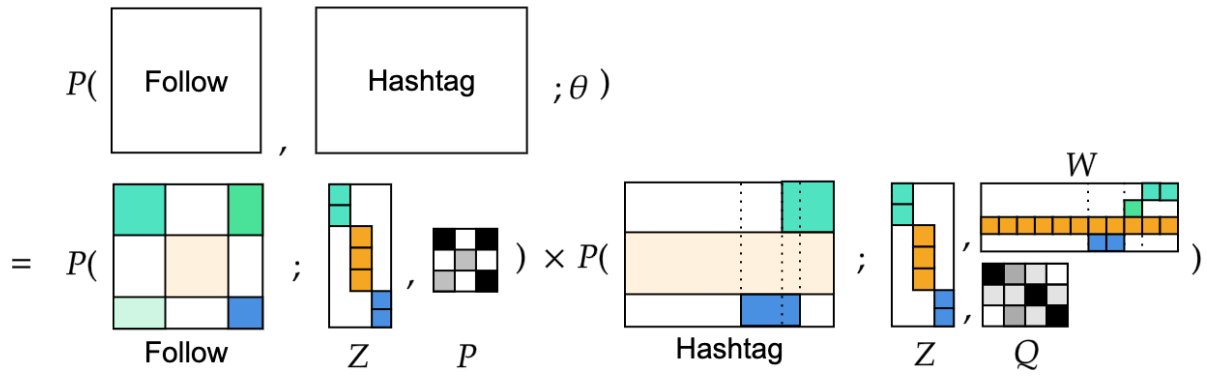

Figure 3.2: Probability factorization of simple-static case

These latent groups are specified through affiliation matrices $\mathbf{Z}$ and $\mathbf{W}$. Let user affiliation matrix $\mathbf{Z} \in \{0,1\}^{n \times g} : \sum_{k=1}^{g} \mathbf{Z}_{ik} = 1$ encode hard partition of n users into g non-overlapping latent groups, and hashtag affiliation matrix $\mathbf{W} \in \{0,1\}^{d \times m}$ encode hard partition of d hashtags into m possibly overlapping latent groups. The constraint of each row of $\mathbf{Z}$ having exactly 1 nonzero reflects the fact that we are grouping users into unique non-overlapping groups.

The mean parameters for the grids of Bernoulli random variables on $\mathbf{A}$ and $\mathbf{X}$ are stored in core matrices $\mathbf{P} \in [0,1]^{g \times g}$ and $\mathbf{Q} \in [0,1]^{g \times m}$. $\mathbf{P}_{kk'}$ represents the probability of having an edge between two nodes belonging to latent node groups k and k'. $\mathbf{Q}_{kl}$ represent the probability of having an edge between user group k and hashtag group l. Assuming $\mathbf{A}$ and $\mathbf{X}$ are conditionally independent given latent variables $\mathbf{Z}, \mathbf{W}, \mathbf{P}, \mathbf{Q}$, the probability of conditionally observing $\mathbf{A}$ and $\mathbf{X}$ can be factorized by (Figure 3.2):

$$
\begin{aligned}
P(\mathbf{A}, \mathbf{X}|\mathbf{Z}, \mathbf{W}, \mathbf{P}, \mathbf{Q}) &= P(\mathbf{A}|\mathbf{Z}, \mathbf{P}) P(\mathbf{X}|\mathbf{Z}, \mathbf{W}, \mathbf{Q}) \\
&= \prod_{ii'kk'} f(\mathbf{P}_{kk'}, \mathbf{A}_{ii'})^{\mathbf{Z}_{ik}\mathbf{Z}_{i'k'}} \prod_{ijkl} f(\mathbf{Q}_{kl}, \mathbf{X}_{ij})^{\mathbf{Z}_{ik}\mathbf{W}_{kl}} \\
\text{where } f(a,b) = a^b(1-a)^{1-b} &\text{ and } i, i' \in \mathcal{U}; j \in \mathcal{H}; k, k' \in \{1, \ldots, g\}; l \in \{1, \ldots, m\}
\end{aligned}
\tag{3.1}
$$

Given such a latent variable model we define an organized group to be a small user group k with corresponding hashtag groups $\{l\}$ that induces high edge probability on $\mathbf{A}$ and $\mathbf{X}$, essentially high $\mathbf{P}_{kk}$ and $\mathbf{Q}_{kl}$ and low $\sum_i \mathbf{Z}_{ik}$ compared to the background or naturally occurring network communities. In practice, given $\mathbf{A}$ and $\mathbf{X}$, we aim to return a list of user groups ranked by how likely it is organized.

3.1.2 Complex Dynamic Case

Consider a complex dynamic Twitter dataset of a static follower graph $\mathbf{A} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{U}|}$, a temporal user hashtag bipartite graph $\mathbf{X} \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{U}|\times|\mathcal{X}|}$, temporal retweet graph $\mathbf{R} \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{U}|\times|\mathcal{U}|}$ and temporal mention graph $\mathbf{M} \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{U}|\times|\mathcal{U}|}$, with the set of users $\mathcal{U}$, hashtags $\mathcal{H}$, and timestamps $\mathcal{T}$ for which data snapshots were taken. Thus, extending from previous section, an organized group not only densely follow each other, frequently use distinct set of hashtags, but also densely retweet or mention each other, and do so in some temporal snapshots of the data. Therefore, organized groups induce dense subgraphs on $\mathbf{A}$, and dense sub-blocks on $\mathbf{X}, \mathbf{R}, \mathbf{M}$. Similarly, these organized groups are assumed to be much smaller than naturally formed network communities.

We similarly extend Latent Block Model to handle multiple tensors. Assume $\mathbf{A}, \mathbf{X}, \mathbf{R}, \mathbf{M}$ can be modeled by Bernoulli random variable grids, whose distributions are specified through latent groups on sets of users, hashtags and timestamps.

We denote these groups $\mathcal{G}_U, \mathcal{G}_H, \mathcal{G}_T$, which are encoded through affiliation matrices $\mathbf{F}_U \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{G}_U|}$, $\mathbf{F}_H \in \{0,1\}^{|\mathcal{H}|\times|\mathcal{G}_H|}$, and $\mathbf{F}_T \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{G}_T|}$. Furthermore, we set $\sum_{i\in\mathcal{U}, j\in\mathcal{G}_U} \mathbf{F}_U = 1$ to enforce non-overlapping user groups.

The mean parameters for the grids of Bernoulli random variables on $\mathbf{A}$, $\mathbf{X}$, $\mathbf{R}$ and $\mathbf{M}$ are stored in core matrices or tensors $\mathbf{C}_A \in [0,1]^{|\mathcal{G}_U|\times|\mathcal{G}_U|}$, $\mathbf{C}_X \in [0,1]^{|\mathcal{G}_T|\times|\mathcal{G}_U|\times|\mathcal{G}_H|}$, $\mathbf{C}_R \in [0,1]^{|\mathcal{G}_T|\times|\mathcal{G}_U|\times|\mathcal{G}_U|}$ and $\mathbf{C}_M \in [0,1]^{|\mathcal{G}_T|\times|\mathcal{G}_U|\times|\mathcal{G}_U|}$. $\mathbf{C}_{Xijk}$ represents the probability of having an edge among group $i \in \mathcal{G}_T$, $j \in \mathcal{G}_U$, $k \in \mathcal{G}_H$. We similarly assume $\mathbf{A}$, $\mathbf{X}$, $\mathbf{R}$ and $\mathbf{M}$ are conditionally independent given latent variables $\mathbf{F}_U$, $\mathbf{F}_H$, $\mathbf{F}_T$, $\mathbf{C}_A$, $\mathbf{C}_X$, $\mathbf{C}_R$, $\mathbf{C}_M$, thus the probability of conditionally observing $\mathbf{A}$, $\mathbf{X}$, $\mathbf{R}$ and $\mathbf{M}$ can be factorized by:

$$\begin{aligned} &P(\mathbf{A}, \mathbf{X}, \mathbf{R}, \mathbf{M}|\mathbf{F}_U, \mathbf{F}_H, \mathbf{F}_T, \mathbf{C}_A, \mathbf{C}_X, \mathbf{C}_R, \mathbf{C}_M) \\ &= P(\mathbf{A}|\mathbf{F}_U, \mathbf{C}_A)P(\mathbf{X}|\mathbf{F}_T, \mathbf{F}_U, \mathbf{F}_H, \mathbf{C}_X)P(\mathbf{R}|\mathbf{F}_T, \mathbf{F}_U, \mathbf{C}_R)P(\mathbf{M}|\mathbf{F}_T, \mathbf{F}_U, \mathbf{C}_M) \end{aligned} \tag{3.2}$$

For clarity, we only expand $P(\mathbf{X}|\mathbf{F}_T, \mathbf{F}_U, \mathbf{F}_H, \mathbf{C}_X)$ into product of individual terms:

$$\begin{aligned} &P(\mathbf{X}|\mathbf{F}_T, \mathbf{F}_U, \mathbf{F}_H, \mathbf{C}_X) \\ &= \prod_{t\in\mathcal{T}, u\in\mathcal{U}, h\in\mathcal{H}, g_t\in\mathcal{G}_T, g_u\in\mathcal{G}_U, g_h\in\mathcal{G}_H} f(\mathbf{C}_{g_t g_u g_h}, \mathbf{X}_{tuh}) \\ &\text{where } f(a,b) = a^b(1-a)^{1-b} \end{aligned} \tag{3.3}$$

Given such a latent variable model we extend the definition of organized group from previous section to be a small user group $u \in \mathcal{G}_U$ with corresponding timestamp groups $\{t \in \mathcal{G}_T\}$ and hashtag groups $\{h \in \mathcal{G}_H\}$ that induce high edge probability on $\mathbf{A}$, $\mathbf{X}$, $\mathbf{R}$ and $\mathbf{M}$, essentially high $\mathbf{C}_{Auu}$, $\mathbf{C}_{Xtuh}$, $\mathbf{C}_{Rtuu}$, $\mathbf{C}_{Mtuu}$ and low $\sum_{i\in\mathcal{U}} \mathbf{F}_{iu}$ compared to the background or naturally occurring network communities. Similarly, in practice, given $\mathbf{A}$, $\mathbf{X}$, $\mathbf{R}$ and $\mathbf{M}$, we aim to return a list of user groups ranked by how likely it is organized.

3.2 Embedding Phase

Given a simple static or complex dynamic Twitter dataset defined in previous section, we propose the following three methods to embed users in low-dimensional Euclidean space. The first two methods: joint autoencoder, linear projection and aggregation only handle simple static case, while the third method: coupled tensor factorization is more general and can handle both cases. We use the same symbols as previous section: in simple static case, we use $\mathbf{A}$ for user follower graph and $\mathbf{X}$ for user hashtag graph; in complex dynamic case, we use $\mathbf{A}$ for static follower graph, $\mathbf{X}$ for temporal user hashtag graph, $\mathbf{R}$ for temporal user retweet graph and $\mathbf{M}$ for temporal user mention graph.

3.2.1 Joint Autoencoder

Our joint autoencoder architecture is inspired by [97], where we extend their loss functions to deal with multiple data sources. The architecture is shown in Figure 3.3 which consists of $\phi^e_{\mathbf{A}}, \phi^e_{\mathbf{X}}, \phi^e_{\mathbf{J}}, \phi^d_{\mathbf{A}}, \phi^d_{\mathbf{X}}$ where ϕ can be a function represented by a single layer of neural network or composition of multiple layers, ϕ^e is encoding function and ϕ^d is decoding function. Subscripts of $\phi : \mathbf{A}, \mathbf{X}, \mathbf{J}$ denote the information ϕ^e encodes from or ϕ^d decodes into, where $\mathbf{A}$ and $\mathbf{X}$ are input graphs, and $\mathbf{J}$ is concatenated latent representation for users using information from both graphs: $concat(\phi^e_{\mathbf{A}}(\mathbf{A}), \phi^e_{\mathbf{X}}(\mathbf{X}))$. Decoders $\phi^d_{\mathbf{A}}$ and $\phi^d_{\mathbf{X}}$ transform joint latent representation $\mathbf{J}$ to approximation of $\mathbf{A}, \mathbf{X} : \hat{\mathbf{A}}, \hat{\mathbf{X}}$.

The joint reconstruction error weighted by hyperparameters $w_{\mathbf{A}}$ and $w_{\mathbf{X}}$, with attention weights $\mathbf{W}^{\mathbf{A}}_{att}$ and $\mathbf{W}^{\mathbf{X}}_{att}$ is calculated by

$$\mathbf{H} = \phi^e_{\mathbf{J}}(\mathbf{J}) \tag{3.4}$$

$$\mathcal{L}^{\mathbf{A}}_{recon} = ||(\phi^d_{\mathbf{A}}(\mathbf{H}) - \mathbf{A}) \odot \mathbf{W}^{\mathbf{A}}_{att}||^2_F \tag{3.5}$$

$$\mathcal{L}^{\mathbf{X}}_{recon} = ||(\phi^d_{\mathbf{X}}(\mathbf{H}) - \mathbf{X}) \odot \mathbf{W}^{\mathbf{X}}_{att}||^2_F \tag{3.6}$$

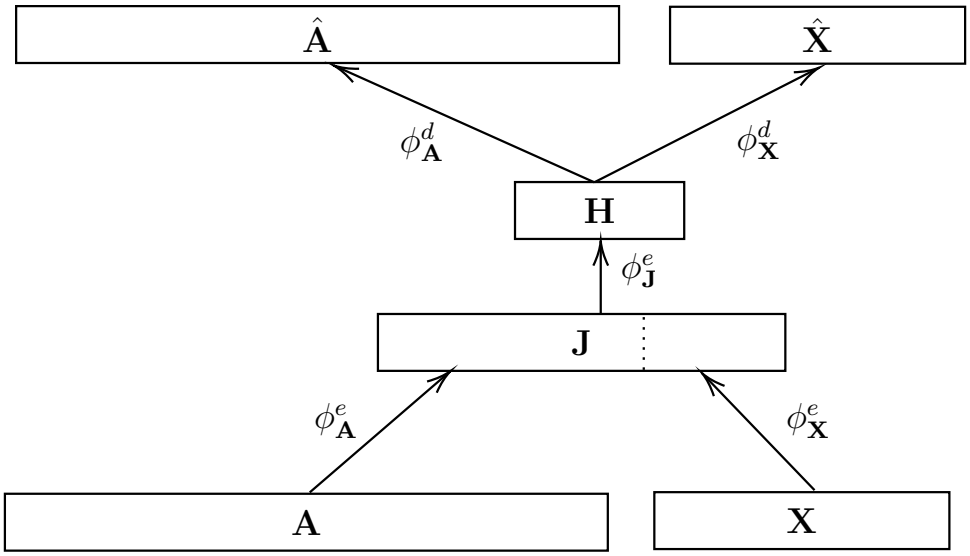

Figure 3.3: Joint Autoencoder Architecture

$$\mathcal{L}_{recon} = w_{\mathbf{A}}\mathcal{L}^{\mathbf{A}}_{recon} + w_{\mathbf{X}}\mathcal{L}^{\mathbf{X}}_{recon} \tag{3.7}$$

Besides reconstruction loss, we define similarity loss as the discrepancy between pairwise Euclidean distance of $\mathbf{H}$ and pairwise Jaccard distance of $\mathbf{A}$ and $\mathbf{X}$, weighted by the same $w_{\mathbf{A}}$ and $w_{\mathbf{X}}$. In order to compare these 2 different distance metrics, we apply a logit transformation on the pairwise Euclidean distance to compress its range to $[0, 1]$, the same as the range of pairwise Jaccard distance. Let $\mathbf{S}^{\mathbf{X}}_{Jar}$ be the pairwise Jaccard distance of rows of $\mathbf{X}$, similarly $\mathbf{S}^{\mathbf{A}}_{Jar}$ for $\mathbf{A}$, and $\mathbf{S}^{\mathbf{H}}_{Euc}$ be the pairwise Euclidean distance for latent vectors $\mathbf{H}$, and choose $\lambda \geq 0$:

$$\mathcal{L}^{\mathbf{A}}_{sim} = ||exp(-\lambda \mathbf{S}^{\mathbf{H}}_{Euc}) - \mathbf{S}^{\mathbf{A}}_{Jar}||^2_F \tag{3.8}$$

$$\mathcal{L}^{\mathbf{X}}_{sim} = ||exp(-\lambda \mathbf{S}^{\mathbf{H}}_{Euc}) - \mathbf{S}^{\mathbf{X}}_{Jar}||^2_F \tag{3.9}$$

$$\mathcal{L}_{sim} = w_{\mathbf{A}}\mathcal{L}^{\mathbf{A}}_{sim} + w_{\mathbf{X}}\mathcal{L}^{\mathbf{X}}_{sim} \tag{3.10}$$

The joint loss to minimize is weighted combination of reconstruction loss and similarity loss with weights w_{recon} and w_{sim}, plus L2 regularization loss at every layer:

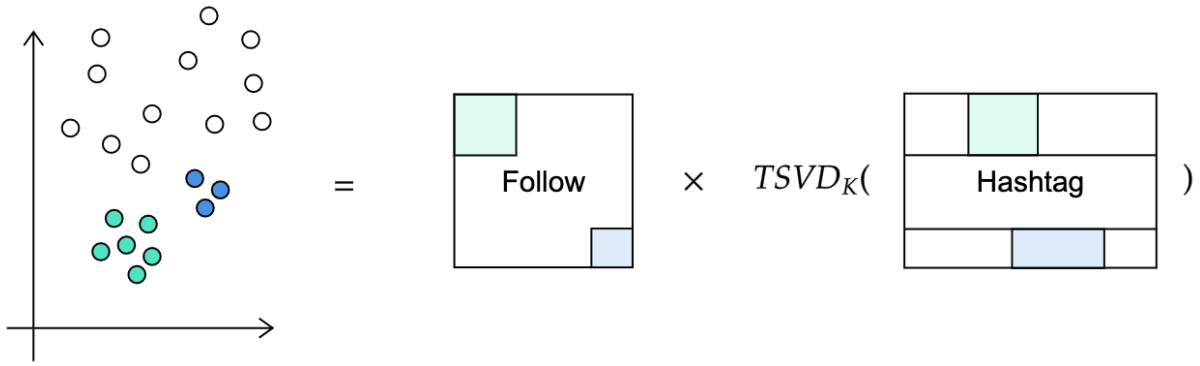

Figure 3.4: Linear projection and aggregation

$$\mathcal{L}_{joint} = w_{recon}\mathcal{L}_{recon} + w_{sim}\mathcal{L}_{sim} + \mathcal{L}_{reg} \tag{3.11}$$

In practice, we train on sampled batches instead of the entire data matrix. For each epoch, we select node $v_i \in \mathcal{V}$ uniformly at random, and sample set of nodes $\{v_j : v_j \in \{\mathcal{V} - v_i\}\}$ according to some distribution D related to the similarity between v_i and v_j. Finally, the learned embedding matrix $\mathbf{H}$ in Equation 3.4 is the user embedding.

3.2.2 Linear Projection and Aggregation (SCG)

Our linear projection and aggregation method (SCG) is inspired by [64], where we extend their method to deal with additional data sources through message passing. Our method is based on two fundamental building blocks: low-rank approximation of matrix denoted by linear projection operator Π_K, where K specifies the resulting dimension after projection; and message passing on $\mathbf{A}$ denoted by operator M_A. The variants of our method differ in the order with which to apply Π_K and M_A:

$$\begin{aligned} \phi(\mathbf{A}, \mathbf{X}) &= M_A(\Pi_K(\mathbf{X})) \\ \psi(\mathbf{A}, \mathbf{X}) &= \Pi_K(M_A(\mathbf{X})) \end{aligned} \tag{3.12}$$

In this work, we chose Π_K to project input matrix on its first K left singular vectors, and chose M_A to use summation aggregator for graph message passing. Applying ϕ or ψ on input data matrices results in user embedding $\mathbf{H}$.

The motivation for our choice of projection operator stems from [64], where it provably finds tiny clusters of size $O(n^{\epsilon>0})$ given a random bipartite graph of size n in the presence of high noise, whereby previous algorithms were only capable of $\Omega(\sqrt{n})$. The core step of the algorithm is to project each row of the biajacency matrix onto its first K left singular vectors, and then apply K-Means clustering to identify clustering on the rows. This operation is conjectured to give the same theoretical guarantees as Mitra's algorithm [26], which leads to the provable recovery of tiny clusters of size $O(n^{\epsilon>0})$. Our Π_K projection operator is identical to that of [64], and shares the same guarantee of detecting tiny clusters in the hashtag graph that are otherwise hard to detect, especially given the amount of high destructive noise in real-world data.

The motivation for our aggregation operator is motivated by message passing algorithms on graphs, including graph convolution networks and label propagation

3.2.3 Coupled Tensor Factorization

Our coupled tensor decomposition method is inspired by [44], where we modified the stochastic gradient to handle coupled Tucker decomposition instead of just single-tensor CP decomposition. Our aim is to apply coupled tensor factorization to compress input data matrices and tensors into products of factor matrices and core tensors, thus revealing intrinsic structure in data and obtaining informative Euclidean embeddings for each entity set in input data.

Given $\mathbf{A}$, $\mathbf{X}$, $\mathbf{R}$, $\mathbf{M}$, we aim to learn latent variables in the form of factor matrices and core tensors $\mathbf{F}_U$, $\mathbf{F}_H$, $\mathbf{F}_T$, $\mathbf{C}_A$, $\mathbf{C}_X$, $\mathbf{C}_R$, $\mathbf{C}_M$ that reflect exactly our definition of organized groups in previous section. Note that to be able to use stochastic gradient that scales to

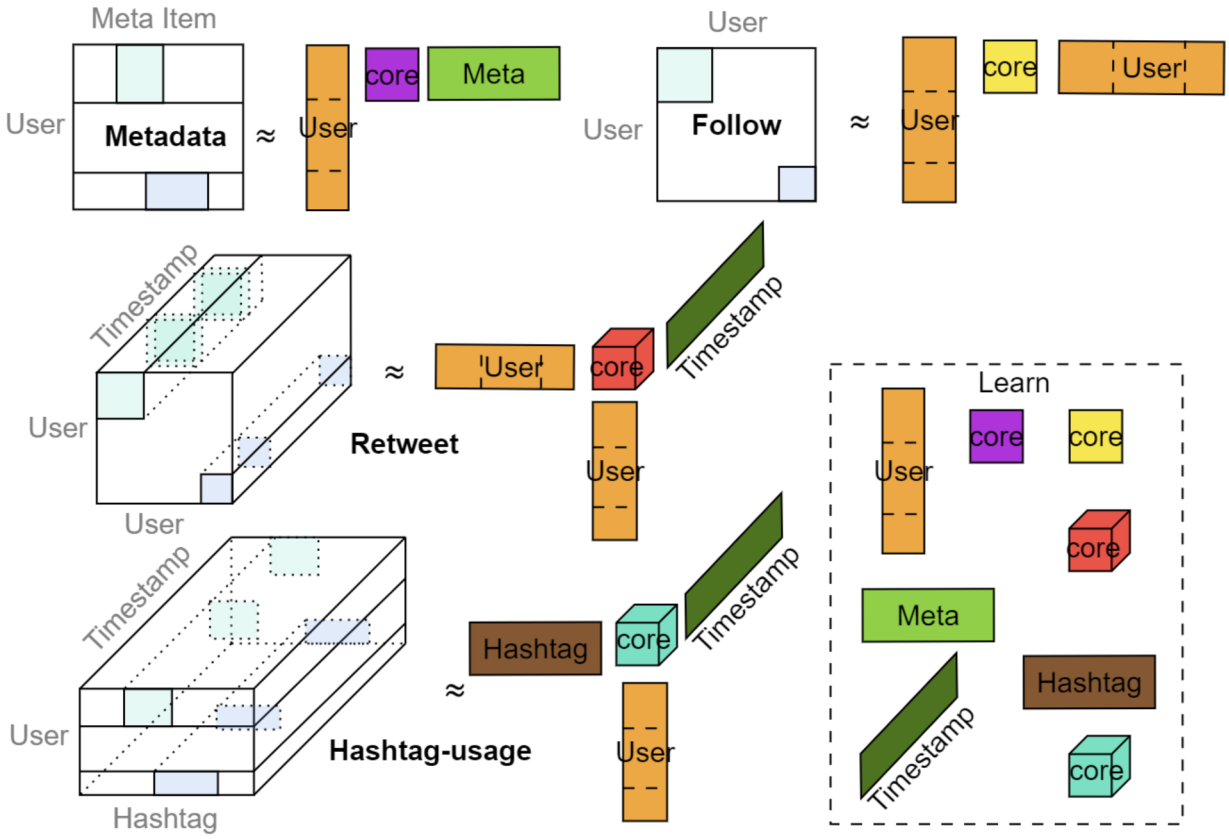

Figure 3.5: Coupled tensor factorization

large dataset, we relax all Boolean and left-stochastic constraints on learnable parameters. We use mean squared loss:

$$\begin{aligned}\mathcal{L} = ||\mathbf{A} - \mathbf{C}_U \times_1 \mathbf{F}_U \times_2 \mathbf{F}_U||^2 + ||\mathbf{X} - \mathbf{C}_X \times_1 \mathbf{F}_T \times_2 \mathbf{F}_U \times_3 \mathbf{F}_H||^2 + \\ ||\mathbf{R} - \mathbf{C}_R \times_1 \mathbf{F}_T \times_2 \mathbf{F}_U \times_3 \mathbf{F}_U||^2 + ||\mathbf{M} - \mathbf{C}_M \times_1 \mathbf{F}_T \times_2 \mathbf{F}_U \times_3 \mathbf{F}_H||^2\end{aligned} \tag{3.13}$$

We learn the latent parameters using stochastic gradient descent, and the resulting user embedding $\mathbf{H}$ is the factor matrix $\mathbf{F}_U$ shown in Figure 3.5. Note that in this decomposition scheme, user metadata and followership are static because based on our analysis of collected data, these information about users rarely change over time. However, temporal generalization of these data sources is trivial under our framework.

3.3 Clustering and Ranking Phase

Given user embedding $\mathbf{H}$ from embedding phase, we first apply clustering to discover latent user groups $\mathcal{G}_U$, and then rank them according to how likely it is organized.

Given user embedding matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$, we apply centroid-based or density-based clustering algorithms to obtain indicator vector $\mathbf{c} \in \{1 \dots K\}^n$, where K is number of clusters and $\mathbf{c}_i$ indicates the cluster index of user i.

The most popular centroid-based clustering algorithm is K-Means clustering, where K is heuristically chosen and then K random centroid vectors are initialized and iteratively updated to minimize the distances between centroids and nearby data points. This can be equivalently viewed as an optimization problem - constrained matrix decomposition:

$$\begin{aligned} &\text{Minimize}_{\mathbf{U},\mathbf{V}} ||\mathbf{H} - \mathbf{U}\mathbf{V}||_2^2 \\ &\text{subject to } \forall i \in \{1 \dots n\} : ||\mathbf{U}_{i,:}||_2 = 1, ||\mathbf{U}_{i,:}||_0 = 1 \end{aligned} \tag{3.14}$$

The data matrix is approximated by the multiplication of representation matrix $\mathbf{U} \in \{0,1\}^{n \times K}$ and archetype matrix $\mathbf{V} \in \mathbb{R}^{K \times d}$ using squared loss. $\mathbf{U}$ is constrained to be binary and have exactly one nonzero on each row i, with the index of the nonzero corresponding to its corresponding cluster index $\mathbf{c}_i$. $\mathbf{V}$ stores K centroid vectors as rows. The iterative update procedure above termed as EM (Expectation Maximization) is equivalent to alternate minimization, where at each step, either $\mathbf{U}$ or $\mathbf{V}$ is fixed and the other is updated to reach minimum loss. It is known that alternate minimization does not guarantee global optima. Therefore, in practice, multiple runs with different random initialization is used, and cluster index corresponding to the lowest loss is used. The elbow heuristic is typically used to choose appropriate K, finding a balance between minimal K that yield low enough loss. To scale K-Means to larger data, a batch update algorithm has been proposed where the same procedure is repeated on random sub-samples of the data until

convergence [78]. K-Means clustering scales linearly or sub-linearly (sampling-based) to the data size in time complexity and does not require additional storage for intermediate data structures, thus it is scalable to large social media dataset. However, it sometimes fail to capture high-density regions since it cannot differentiate region densities. It is also biased to find equal-sized clusters of spherical shapes, and also assigns a cluster index to each data point, which is not always compatible with our use case - finding tiny dense clusters in high dimensional data.

Therefore, we also use density-based clustering when the data is small enough to have reasonable runtime and memory usage. The most popular density-based clustering algorithm is DBSCAN [20], where nearby data points in dense regions are grouped incrementally into clusters and points far away from any dense regions are categorized as noise. There are three types of data points in DBSCAN: core points, edge points and noise points, where only core and edge points form clusters. Initially, all points are initialized as noise points. Two key hyperparameters control how to flip some of the noise points into core points and edge points with corresponding cluster labels: epsilon is the largest distance two points can be to be considered neighbors; min_samples is the minimum number of neighbors a point has in order to become core point. Specifically, if a point has more neighbors than min_samples, it becomes a core point and becomes a cluster. Then all its neighbors get grouped into the same cluster. Next, each of the neighbors will be checked if they can be core point. if so, the same procedure is recursively applied; otherwise, it is an edge point. Such procedure is applied for all data points until all points are visited.

Given user cluster labels $\mathbf{c} \in \{1 \dots K\}^n$, we create a suspicious score for each cluster to rank the most suspicious one on top for human experts to verify whether the cluster of users indeed engages in coordinated misinformation. Empirically we found the empirical edge probability in cluster-induced subgraph on the follower network to an effective and easy-to-calculate suspicious score. It is defined as the observed number of edges over all

possible edges among a set of users. In directed graphs such as the follower network, all possible number of edges for a cluster of n users is n^2.

Chapter 4

Results and Discussions

In this section, we go through both synthetic and real-world data experiments' results for each method described in previous section. More details for joint autoencoder and linear projection and aggregation (SCG) are available at [98, 99].

4.1 Joint Autoencoder

We first conduct synthetic experiments to choose the best set of hyperparameters for exploratory analysis on real data, as well as to demonstrate the effectiveness of joint autoencoder on binary attributed graph dense sub-block detection compared to FRAUDAR [35], a classical baseline for dense sub-block detection with only adjacency matrices, and DOMINANT [19], a Graph Convolutional Network (GCN) based approach that utilizes both adjacency and attribute matrices. We then create a joint "fingerprint" of identified clusters based on both the graph topology of cluster-induced subgraph, and attributes of nodes in the cluster, which could potentially be used to identify Information Operations in Canadian 2019 Federal Election. We also manually inspect the nodes in the three clusters with highest cluster-induced network density, and find some suspicious accounts that might have engaged in Information Operations.

4.1.1 Data Collection

The dataset comprises of 38,498 tweets from 7,298 distinct Twitter users, collected between August and October 2019. The tweets were collected using the Twitter Streaming API, using the following hashtags - #Trudeau, #TrudeauMustGo, #cdnpoli, #TrudeauResign, #LavScam, #SNCgate, #StandWithTrudeau. We also collected the list of followers for each of the 7,298 users in our dataset to construct a follower network, resulting in a total of 474,459 connections. The hashtags are further formulated as a vector of size 3,047 to represent the attributes of each node or user, denoting whether the user used a certain hashtag in the dataset. We represent the entire data as concatenated adjacency and attribute matrix as shown in Figure 4.1.

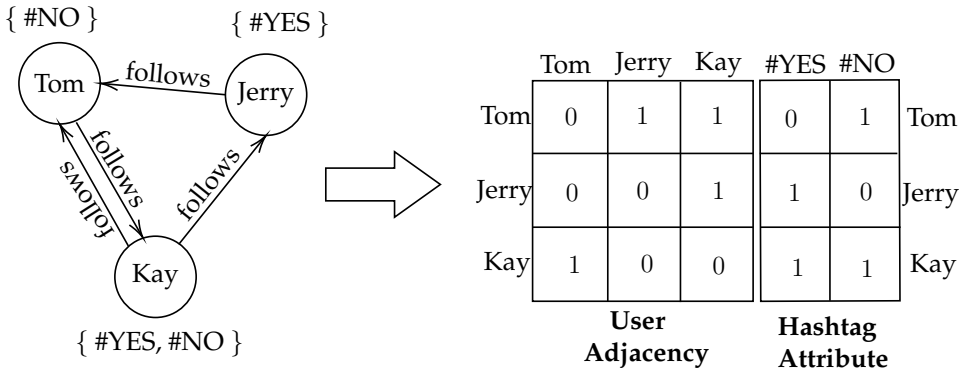

Figure 4.1: Concatenated Adjacency and Attribute Matrix

4.1.2 Hyper-Parameter Tuning

We inject artificial dense sub-block anomalies into our Twitter data in order to tune our algorithm to perform well for the unsupervised anomaly detection task. With the injected data, we conduct a random search of the hyperparameter space and identify the best hyperparameter option by F-1 score, with labels being anomaly or non-anomaly. Then we use it to identify interesting dense clusters on the real data without dense sub-block

injection. We show that our method outperforms both baselines across all injected sub-block densities in Figure 4.2.

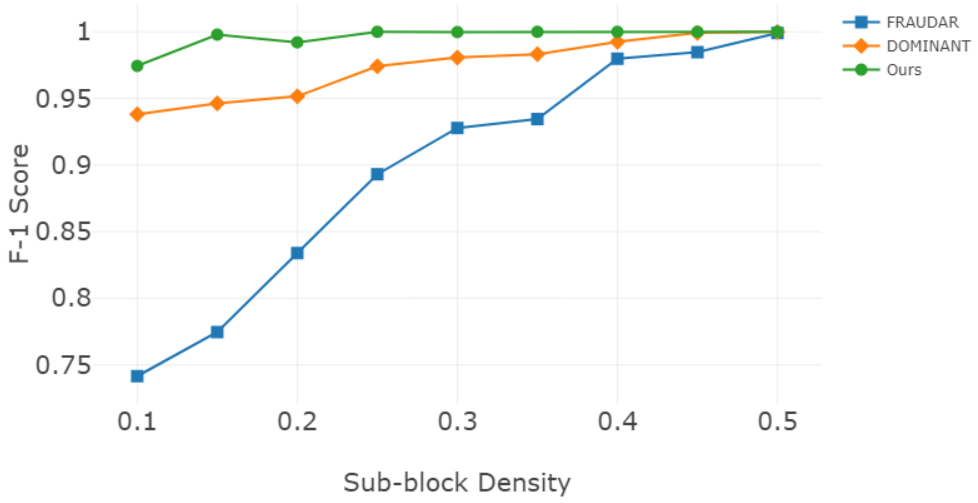

Figure 4.2: Synthetic Experiment Performance

Synthetic Data Generation

For adjacency matrix, we inject dense subgraph by injecting random dense graph with a specified density and size at sub-block indices. For attribute matrix entries, we create an empirical distribution of hashtag usage indicating how likely a random person from a sub-block would use certain hashtags, and apply add-k smoothing on this empirical distribution. Next, we sharpen the distribution by applying an exponential factor to it: $exp(\lambda \cdot)$ where λ controls for how concentrated the transformed distribution is. By sampling a certain number of hashtags from this distribution, we simulate the presence of Information Operations, where a group of highly connected users tweet a subset of hashtags. Finally we inject the bipartite graph with the specified density at these sub-block and attribute indices. For our experiment, we inject 3 dense sub-blocks of size 500, and

use the same network density for both adjacency matrix and attribute matrix, from 0.1 to 0.5 with 0.05 interval.

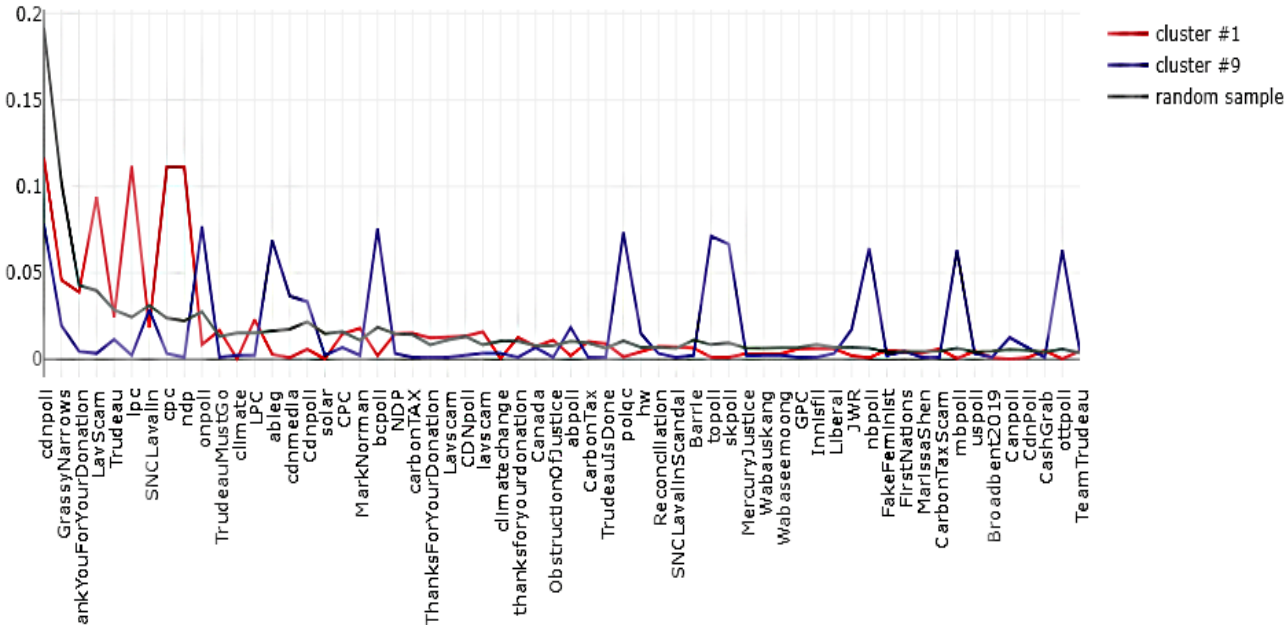

Figure 4.3: Hashtag Fingerprint: Per-cluster relative usage frequency for popular hashtags

4.1.3 Results

Using the best hyperparameter option, we create 10 clusters from the real-world data. For each cluster and its corresponding induced sub-graph of follower network, we generate hashtag fingerprint, which reflects user attribute information, as well as clustering fingerprint, which reflects key network topology information. We put our focus on 2 of the densest clusters (#9, #1), and report interesting exploratory findings.

For each cluster, we define hashtag fingerprint as the relative usage frequency of popular hashtags within cluster. Note that usage here refers to whether a hashtag is used or not in the dataset for a given user, and frequency refers to the number of users in a cluster using certain hashtag. A high relative usage frequency corresponds to highly used hash-

tag in a cluster. Hashtag fingerprints for cluster #9 and #1, and a randomly sampled set of users are shown in Figure 4.3.

By analyzing the hashtag fingerprint we note that cluster #9 exhibits interesting spikes on hashtags related to diverse locations: Alberta (#ableg), British Columbia (#bcpoli), Quebec (#polqc), Toronto (#topoli), Saskatchewan (#skpoli), New Brunswick (#nbpoli), Manitoba (#mbpoli) and Ottawa (#ottpoli). This is counter-intuitive; we normally assume people engage with each other locally, but we clearly see multi-regional cluster of users in close contact with each other.

For cluster #1, the hashtag usage centers around a recently heated political scandal related to government and corporate corruption (#LavScam), and a few prominant political parties: the Liberal Party of Canada (#lpc), the Conservative Party of Canada (#cpc), and the New Democratic Party (#ndp). This reflects the fact that an emergent cluster of users related to different political parties are talking about the recent scandal.

The hashtag fingerprint for both clusters identified through our algorithm reveals interesting insights that would otherwise be hard to obtain by going through the tweets manually. On the other hand, the hashtag fingerprint of a random sample is highly centered around the most popular hashtags and cannot yield much insight into the user group.

Sample User

We finally manually inspect the Twitter profile page of users in top 3 densest clusters (#8, #9, #1), and for each cluster, we visualized its cluster-induced subgraph and per-node HITS authority score. We use darker green gradient to denote nodes with higher HITS authority score.

Shown in Figure 4.4, we identify one Twitter account highlighted with a dotted red line that exhibits behaviors suspicious of Information Operations. The suspicious user

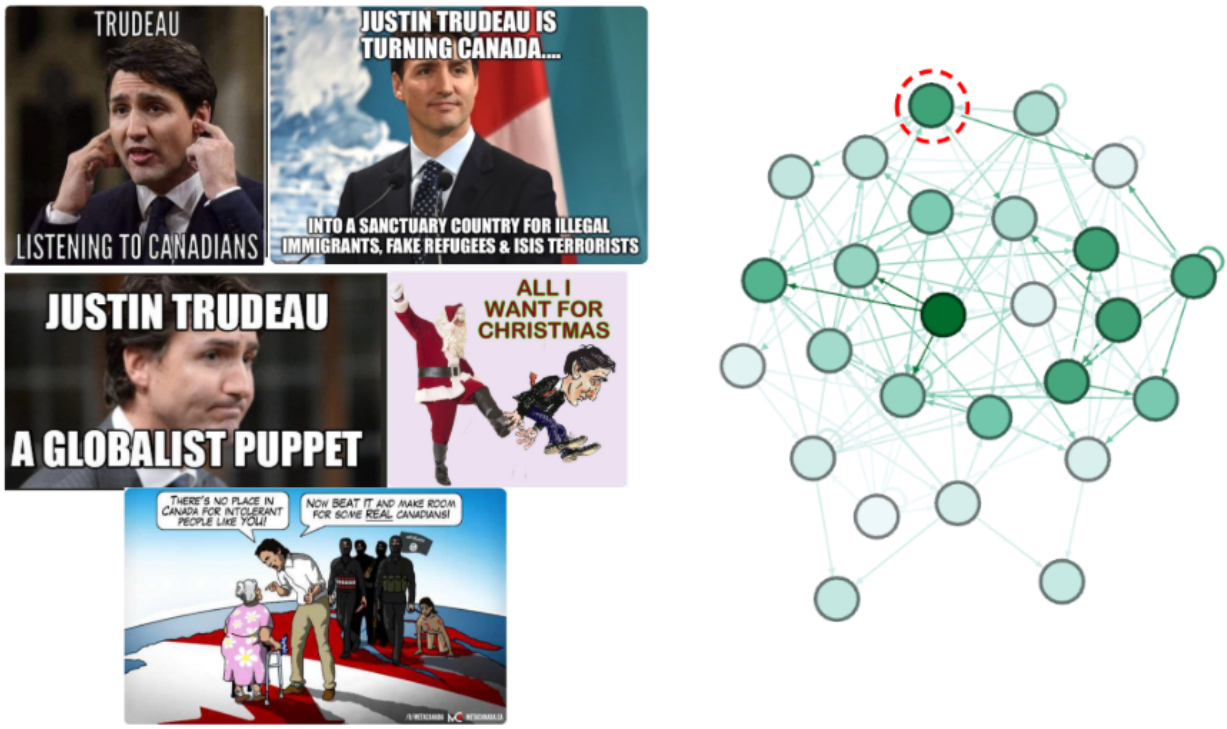

Figure 4.4: Sample Memes used by Suspicious Anomalous User in Cluster #8 and its Follower Network Location

account was created in August 2017. Since December 2017, the user started consistently creating and spreading divisive tweets and memes that demote Justin Trudeau and his administration. A sample of the political memes deployed by this user is shown in Figure 4.4. Furthermore, this user changed it user handle twice from mid-April to mid-May. Such high frequency of changing user handles might be related to malicious intent [36]. Both the identified user's posted content and behavior are suspicious of engaging in Information Operations.

4.2 Linear Projection and Aggregation (SCG)

In this section, we first verify the effectiveness and scalability of SCG (Spotting Coordinated Groups), consisting of linear projection and aggregation. This is done through a set

of synthetic experiments with builtin ground-truth, which approximate the real-world problem and enable us to provide a quantitative evaluation. Next, we discuss the observations provided by applying the method on scraped real-world Twitter data and provide several pieces of evidence on the effectiveness and interpretability of the it in unveiling the dynamics of organized groups around the 2019 Canadian federal election. We used user followership as user connection graph, and hashtag usage as user attribute graph. Due to time and budget constraints, we focus our analysis on data scraped from Twitter around this particular event. Incorporating other social media platforms and other events is part of the future works planned for this study.

4.2.1 Validation on Synthetic Data

Based on observation of our scraped dataset, real-world graphs are large, sparse and have high-dimensional node attributes. To approximate real-world data that has ground truths for organized groups, we generate synthetic attributed graphs with similar characteristics but with injected organized groups that serve as the ground truth. We compare different methods on how well they are able to recover these injected organized groups.

Parameter Settings: We generate eight organized groups with 20 nodes and 20 attributes on differently sized graphs (2,000 to 30,000 nodes/attributes) to test the effectiveness and scalability of our method. This gradually decreases the ratio of coordinated nodes from 8% to 0.5% of the original graph size, thus making the detection progressively more challenging.

Baselines: Literature on unsupervised detection of organized groups is relatively sparse, thus we carefully select unsupervised baselines from related literature: Infomap [75] and Louvain [8] from community detection; Fraudar [34] and pcv [64] from dense subgraph or tiny cluster detection; node2vec [28], attri2vec [103] and unsupervised Graph-

SAGE [30] from network embedding. pcv baseline only considers content, the Infomap, Louvain, Fraudar and node2vec only consider connections, and the other baselines incorporate both content and connections. For a subset of baselines (node2vec, graphSAGE, attri2vec), we only run them on graphs with size up to 18,000 nodes due to time and hardware constraints.

Evaluation: To evaluate partitions (how well organized groups are separated from the background and each other), we use Quality score used in [64], given k ground-truth clusters $U_{1...k}$ and s inferred clusters $\widetilde{U}_{1...s}$, and $J(\cdot,\cdot)$ as the Jaccard similarity between two sets, the Quality score is:

$$Q = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1,...,s} J(U_i, \widetilde{U}_j) \in [0,1] \tag{4.1}$$

To evaluate the ability to classify nodes as belonging to a organized group or not, we use the F1 score. We generate two instances of synthetic attributed graph for each size, and do two runs on each instance and report the mean performance across all four runs.

Performance Analysis: Figure 4.5 illustrates that principal components of our method embeddings for normal versus organized nodes are better separated compared to the other embeddings methods. This is an example embedding on synthetic graph of size 12,000. Table 4.1 reports the full results for all the baselines and settings. We can see that our method outperforms baselines significantly, especially when the organized groups only occupy a small fraction of the graph (0.5%). This indicates that the general community detection or clustering methods are not appropriate for this setting as they are designed with different assumptions, e.g. balanced clusters. The pcv baseline which is specifically designed for detecting tiny clusters fails as it is not able to incorporate the connections and only operates on one mode of the data, user contents. Furthermore, as shown in Figure 4.6, the runtime of our method is more than 10,000 times faster than

some baseline. We can show that our method scales linearly with the number of nonzero entries in $\mathbf{A}$ and $\mathbf{X}$ given some assumptions.

	$n = 2000$		$n = 6000$		$n = 10000$		$n = 14000$		$n = 18000$		$n = 22000$		$n = 26000$		$n = 30000$	
	Quality	F1	Quality	F1	Quality	F1	Quality	F1	Quality	F1	Quality	F1	Quality	F1	Quality	F1
Louvain	3.92	14.81	2.70	4.91	2.43	2.02	2.29	0.56	2.33	0.75	2.12	0.47	1.97	0.00	2.07	0.00
Infomap	11.11	0.00	11.11	0.00	11.11	0.00	11.11	0.00	11.11	0.00	11.11	0.00	11.11	0.00	11.11	0.00
node2vec	3.52	14.81	3.35	6.52	5.74	11.63	6.21	13.56	3.38	0.61	-	-	-	-	-	-
attri2vec	3.93	0.00	4.91	0.55	5.69	0.00	6.66	0.00	6.5	0.00	-	-	-	-	-	-
GraphSAGE	3.36	0.00	2.22	0.00	2.06	0.00	1.85	0.00	1.72	0.00	-	-	-	-	-	-
pcv	5.60	0.00	2.42	0.00	2.00	0.00	1.79	0.00	1.70	0.00	1.58	0.00	1.62	0.00	1.60	0.00
Fraudar	-	15.64	-	5.21	-	3.15	-	2.26	-	1.76	-	1.44	-	1.22	-	1.06
our method	**11.39**	**64.97**	**13.19**	**86.72**	**13.33**	**97.66**	**13.44**	**99.53**	**13.27**	**100.00**	11.00	**75.00**	10.72	**75.00**	**13.41**	**100.00**

Table 4.1: Our method consistently and significantly outperforms baselines in terms of Quality and F1 score on synthetic graphs.

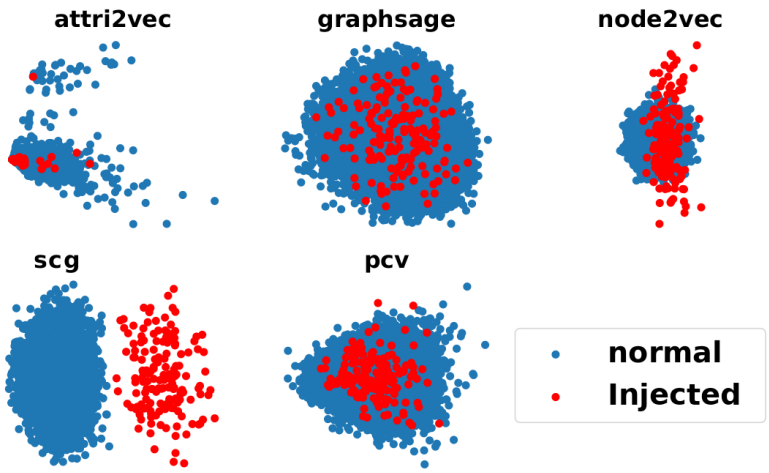

Figure 4.5: our method scg provides better separation for normal versus coordinated nodes.

4.2.2 Results on Real-World Data

Data Collection: Since April 2019, we started collecting tweets related to the 2019 Canadian federal election through the Twitter streaming API filtered by a seed hashtag set based on significant political events in Canada (list of the hashtags used and details are provided in the supplementary materials). We collected sampled tweets between April

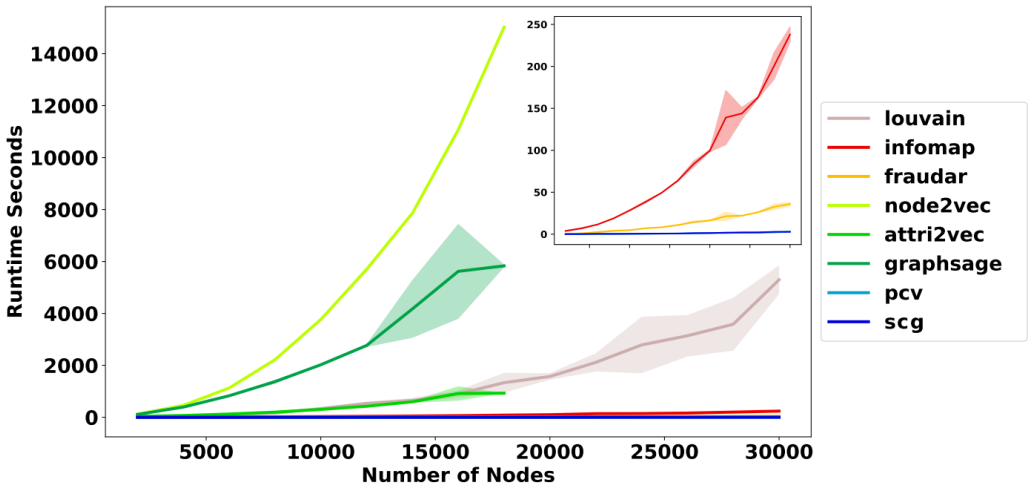

Figure 4.6: our method is significantly faster than most baselines: more than 10,000 times faster than node2vec when graph size is 18,000. The inset plot shows the same comparison focused on the scalable algorithms.

and October 2019 and developed custom scraping pipeline to scrape all followers for Twitter users who used these hashtags. For each user, we tracked *all* hashtags usage in his or her sampled tweets and created an attribute vector where each entry is the frequency of using a specific hashtag. For cross validation, we also tracked whether users been suspended between April and October, and collected their Botometer [17] score from a commonly used API. This API measures the extent to which a Twitter account exhibits similarity to the known characteristics of social bots based on user-generated meta-data, activities, and content, without structural information about his or her follower network. For more details, please refer to the supplementary materials.

Data Representation and Preprocessing: We filter out users who do not have any followers or followees, and obtain a directed attributed graph G that has $n = 69,709$ nodes, $|\mathcal{E}| = 3,480,145$ edges and $d = 1,329,385$ unique hashtags as node attributes. Let J denote set of all hashtags in our data ($|J| = d$), and I denote the set of all users ($|I| = n$). We create adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$ from user followership and attribute matrix $\mathbf{X} \in \mathbb{N}^{n\times d}$ from user hashtag usage. In the following sections, we consistently use n to denote the number of users and d the number of attributes. We apply doubly-normalized TF-IDF to give more significance to uncommon hashtags, because entries of $\mathbf{X}$ are highly skewed:

$$\mathbf{X}^*_{ij} = \frac{n}{\sum_{i'\in I} \mathbf{X}^b_{i'j}} \frac{0.5 + 0.5\mathbf{X}_{ij}}{\max_{j'\in J} \mathbf{X}_{ij'}} \tag{4.2}$$

where $\mathbf{X}^b = \mathbf{X} > 0$ is a binarized attribute matrix.

Results Overview A total of 13 organized groups are detected by our method in our collected data. We visualize them in Figure 4.7, which show a clear block structure for both $\mathbf{A}$ and $\mathbf{X}$ on indices induced by these groups. This indicates the ability for our method to discover tightly connected user groups, each engaging with similar sets of hashtags.

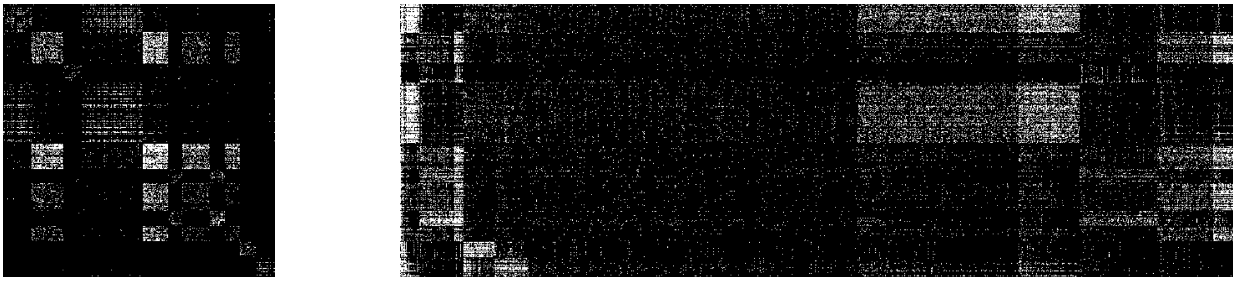

Figure 4.7: our method finds organized groups of users exhibiting block-diagonal structure in both the adjacency (left) and attribute matrix (right) on the twitter data.

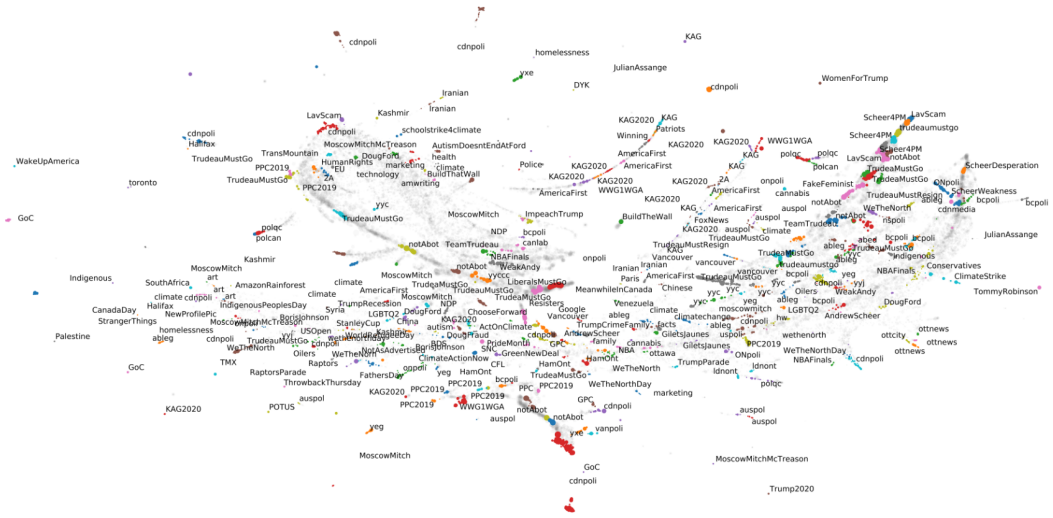

Figure 4.8: our method puts users of the same political creed (related group creeds) close together. Here nodes are the individual users, size of each node corresponds to its individual engagement in the Canadian politics. Nodes are colored the same if they belong to the same cluster.

Comparing with Baselines We compare our method with Fraudar and pcv, which are the only baseline methods that scale to our data size given our time and hardware constraints. Since no ground truth of organized groups is available for real-world data, we compare the suspension index and bot influence index of detected organized groups as a proxy; which are defined below. Given $\mathbf{s} \in \{0,1\}^n$ to where $\mathbf{s}_i = 1$ if user i has been suspended between April and October, and 0 otherwise, we define *Suspension Index* f_S of a set of user accounts I_c to measure the concentration of suspended accounts in this set

	Suspension Index	Bot Influence Index
Fraudar	1.297	1.645
SCG	**4.472**	**1.905**
pcv	1.625	1.890
Random	1	0.918

Table 4.2: our method detects users in organized groups that have the highest suspension index and bot influence index.

relative to the background, :

$$f_S(I_c) = \frac{\sum_{i \in I_c} \mathbf{s}_i / |I_c|}{\sum_{i \in I} \mathbf{s}_i / |I|} \tag{4.3}$$

Given $\mathbf{b} \in \mathbb{R}^n$ containing collected Botometer scores and $\mathbf{f} \in \mathbb{Z}^n$ containing number of followers for users in our dataset, we define *Bot Influence Index* f_B of a set of user accounts I_c to measure their average level of estimated bot influence, :

$$f_B(I_c) = \frac{\sum_{i \in I_c} \mathbf{b_i} \log(1 + \mathbf{f}_i)}{|I_c|} \tag{4.4}$$

As shown in Table 4.2, all methods perform better than uniformly sampling a set of users to be organized, but our method is the clear winner. It detected organized nodes that are over four times likely to be suspended than a random sample and has the highest bot influence index, which is directly related to our definition of organized groups - set of users that boost their influence in an inauthentic fashion. Although both metrics are not designed from ground-truth knowledge of existing organized groups, they show that our method finds interesting groups for further study, some of which we investigated in Figure 4.11. Note in the figure that such high concentration of accounts that contain suspended users, posting politically one-sided (anti-Trudeau), and potentially offensive content right before the Canadian election in 2019 is intriguing.

Discussions and Main Observations Figure 4.8 visualizes our method node embeddings for users in our dataset using UMAP [58]. The sizes of the points in the figure corre-

spond to their individual engagement in discussions around Canadian politics (Equation 4.6). Background nodes, those that reside in the largest cluster are plotted as grey with a lighter shade. Overlayed on each colored cluster of users is the group creed created by our method.

More specifically we define the significance of hashtag $j \in J$ denoted by f_S, as the mean doubly-normalized TF-IDF value across all users, :

$$f_S(j) = \frac{\sum_{i \in I} \mathbf{X}^*_{ij}}{n} \tag{4.5}$$

We set 1,000 hashtags with the highest significance be the set of *Significant Hashtags* J_S. The overlap of this set and our seed hashtag set (and their variants by changing the case of letters) gives the set of *Significant Canadian Hashtags*, which we denote by J_C. We also define *Individual Engagement* - each user's engagement with Canadian politics, denoted by f_e, as the ratio of (at-least-once) usage of hashtags in J_C by that user, :

$$\forall i \in I : f_e(i) = \frac{\sum_{j \in J_C} X^b_{ij}}{|J_C|} \tag{4.6}$$

We observe that *our method embeds groups with similar group creeds close to each other, thus forming an informative map of Twitter*: top middle occupied by American conservative groups indicated by #KAG; the center by international groups signified by #Chinese, #Iranian, #Paris; top right by pro-Scheer (#Scheer4PM) and anti-Trudeau (#TrudeauMustGo) groups; the middle right by anti-Scheer (#ScheerWeakness) groups; the middle left by climate activist groups, evidenced by #climate and #AmazonRainforest.

The adjacency matrix with block-diagonal structure induced by the detected organized groups is visualized in Figure 4.9, where we observe siloed groups as well as interacting ones, which are likely American conservative groups. Another observation is that the potential American conservative group signatured by #WWG1WGA (Where We

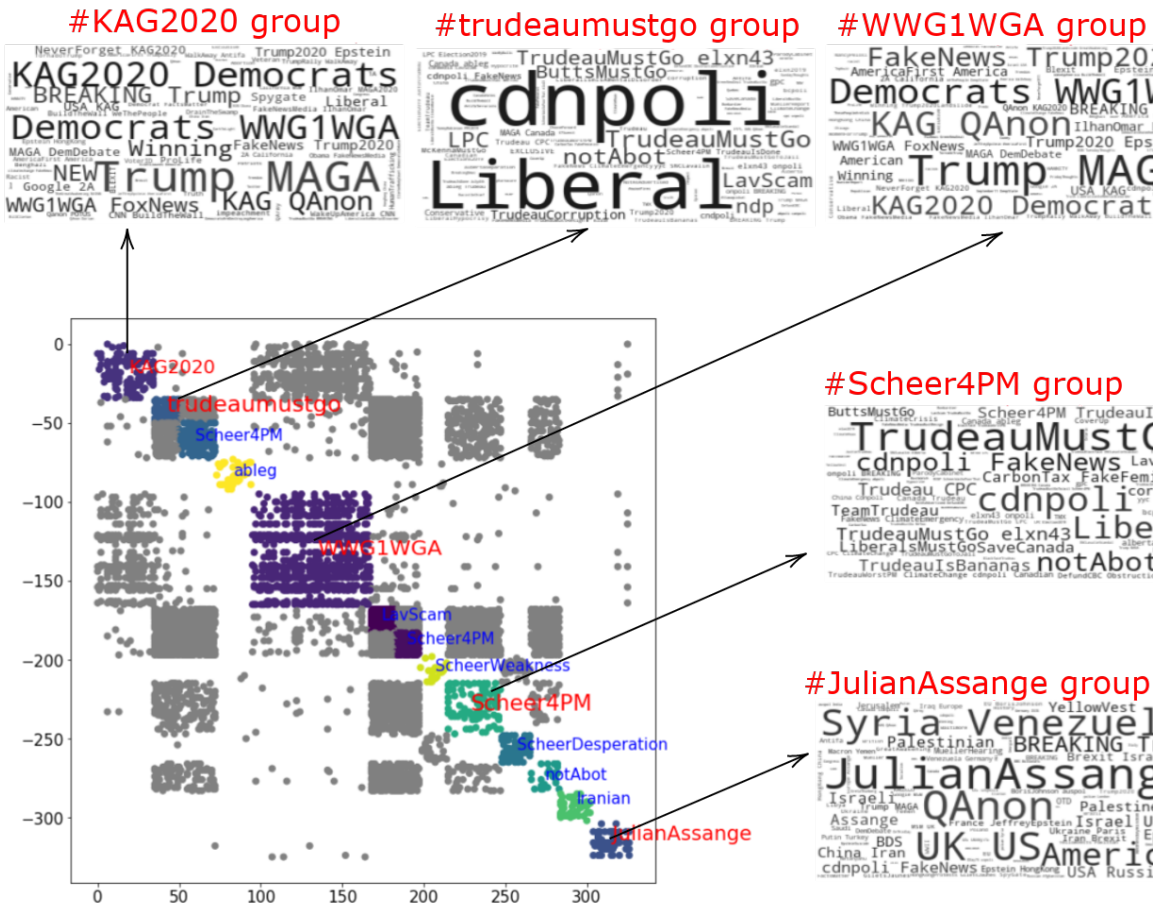

Figure 4.9: Our method detects 13 organized groups in the 2019 Canadian Federal Election including multiple #MAGA groups.

Go One, We Go All) which contains suspended users interacts with two smaller groups with the hashtag signatures of #LavScam and #Scheer4PM, which are likely Canadian anti-Trudeau and pro-Scheer groups. *This interaction could be considered a potential foreign involvement on the Canadian 2019 Election*, which is discovered independently by other researchers after our study [73,74]. Studying the impact/influence of these groups is one of our planned future studies.

Figure 4.10 illustrates the detected organized groups, plotted as red, and non-organized clusters that are not the background are plotted as colored points. The sizes of these points are proportional to their cluster engagement (Equation 4.7). We can see *that the organized*

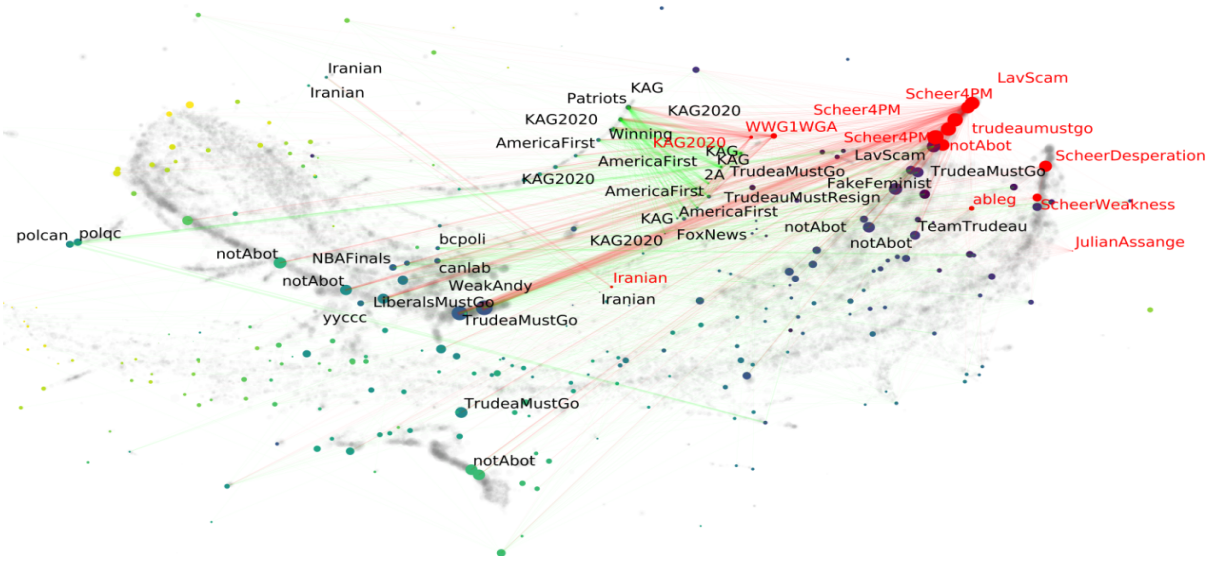

Figure 4.10: Our method summarizes Twitter dynamics of 69,709 accounts around 2019 Canadian Election, providing a bird's eye view of how detected organized groups (marked red) engage in the overall discourse.

groups are highly engaged with Canadian politics, evidenced by their node sizes, and are close to each other in the embedding space. Specifically, we define *Cluster Engagement* with Canadian politics for a set of users, I_c, as their scaled average individual engagements with Canadian politics, :

$$f_E(I_c) = log(|I_c|)\frac{\sum_{i \in I_c} f_e(i)}{|I_c|} \tag{4.7}$$

We have observed that within these organized groups, the empirical likelihood of being *suspended* between April and October is over *four times more likely* compared to a random sample. Many users in these organized groups are highly similar to those suspended accounts. We observe that *the content posted by these groups are mostly offensive*. In Figure 4.11 for example, in the large connected component in one of our detected organized group, we identified several accounts (colored black) that generated politically one-sided and potentially offensive content similar to suspended accounts (colored red): some sampled content from these accounts are appended to the figure. While our method

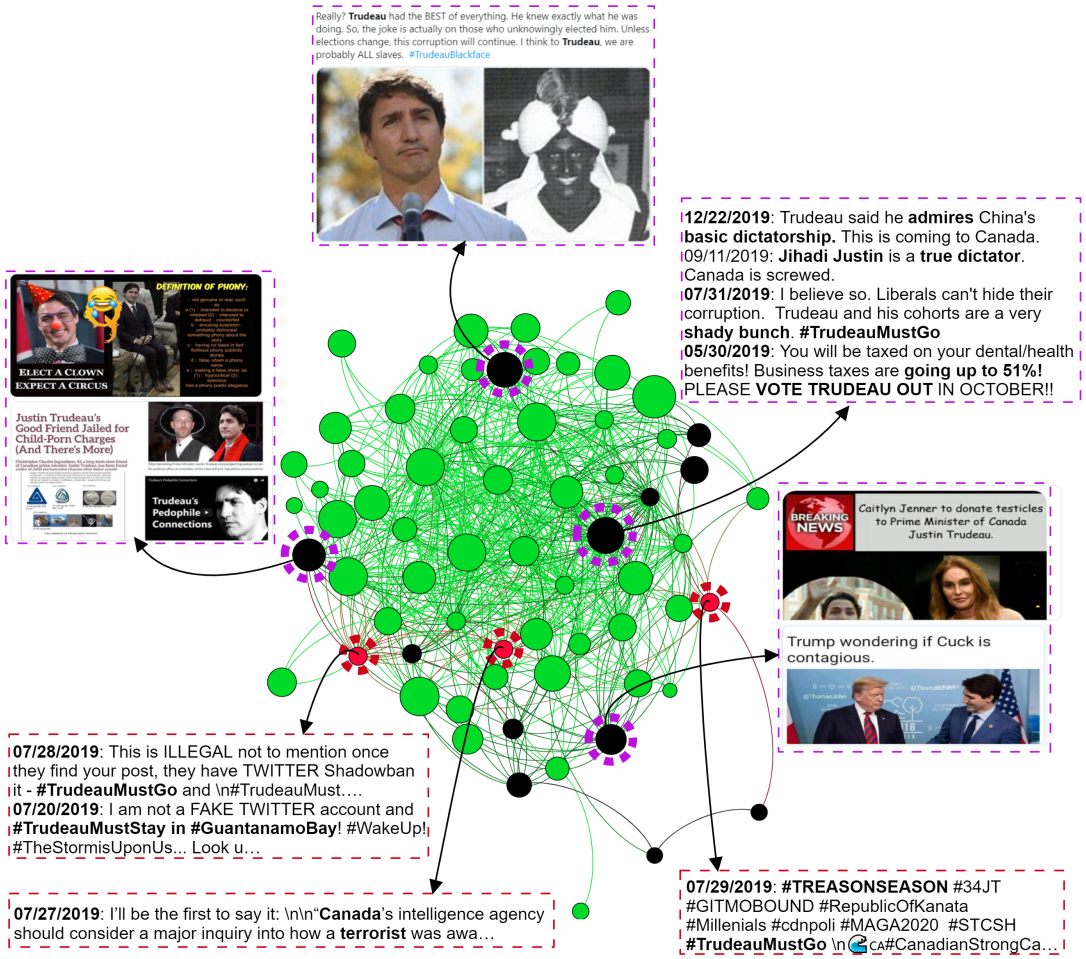

Figure 4.11: Verification with external indicators: our method detects an intriguing organized group: 3 suspended users and multiple other unsuspended users simultaneously tweet politically one-sided (anti-Trudeau) and potentially offensive content.

spot these users who are behaving similarly to the suspended users, these accounts were still active at the time of our analysis.

Looking at the the group creed (signature hashtag) for each group on the Twitter maps in Figure 4.8 and 4.10 discovered by our method, we get a concise characterization of the results and explains the complex structure through which these groups are engaged in Canadian politics. Furthermore, group creeds for organized groups highly overlap with clusters that exhibit the highest ratio of suspended users, including #Iranian, #KAG2020, #notAbot, #TrudeauMustGo, and #Scheer4PM. This makes our method a useful tool for

spotting suspicious messages on social platforms that could have been manipulated by organized groups. We also verify that two of *these hashtags discovered by our method (#notAbot, #TrudeauMustGo) are later confirmed to be linked to misinformation campaigns* [59, 65]. These two hashtags have so far been the primarily used hashtags against the 2019 Canadian election, and both have been detected before mainstream media coverage. This makes our method a powerful tool to assist in detecting trending misinformation campaigns before they make a significant mark.

Our method quantifies the strength of the connection between all pairs of clusters, and thus enables the study of their potential influence. In Figure 4.10, the link between two clusters is plotted with line width proportional to their interaction; those that are connected to the detected 13 organized groups are colored red, and other links are plotted as green. We observe from Figure 4.10 that two sets of clusters have observable interactions (manifested as lines among points) among them. They are respectively represented by two sets of group creeds: (1) #KAG (Keep America Great), $AmericaFirst, #WWG1WGA (Where We Go One, We Go All) and their variants which are related to American conservative politics; and (2) #Scheer4PM, #TrudeauMustGo, #LavScam and their variants which are related to Canadian election politics. Future studies will focus on the expanding this group-level study of detected organized clusters.

A less concerning but still interesting observation is that our method identifies one out of four groups signatured by #Iranian, where two out of the four groups exhibit the highest suspension index. However, no significant connections are going outside of these three groups to other parts of the graph. Inspection of the users' tweets in these clusters reveals that the accounts in these groups are primarily concerned with immigration issues and are mostly created in February 2019, right before the passing of Bill 21, a Bill in Quebec that sets out a framework for values test for skilled workers, which impacts immi-

gration. The observed strong connection within a set of groups but weak or no connection to other parts of a graph could be a sign of a failed amplifying strategy.

4.3 Coupled Tensor Factorization

In this section, we evaluate the performance of our method. The main objective for synthetic experiments is to understand how well our method can recover ground truth partitions on the primary typed entity set (users) when varying the contrast with background, i.e. p and q. As the data size is small in synthetic case, we use nonlinear least squares detailed in [86] instead of gradient-based approach for optimization. We use popular cluster metrics: normalized mutual information for evaluation. We are interested in how our method performs against baselines that do not utilize all available information jointly. We then move on to real-world Twitter dataset related to 2019 Canadian federal election, and then quantify and qualitatively explore the political creed, engagement and influence of identified organized groups.

Algorithms. For synthetic experiments, our method was compared with the `pcv` algorithm by Neumann [64], which uses only metadata matrix; `SCG` algorithm by Wang, et al. [98], which uses both followership matrix and metadata matrix; heuristic which unfolds tensor on user axis and then apply `pcv`, which only uses retweet tensor. We use the same rank $R = 5$ of decomposition to compare all algorithms.

4.3.1 Validation on Synthetic Data

We generate synthetic coupled multidimensional array block model as defined in Equation **??**, and simulates collected Twitter data, thus having the same typed entity sets and couplings, with the primary type as user. To focus on parameters of interest, we fix the cardinality of each typed entity set (700 users, 200 hashtags, 200 metadata items, 200

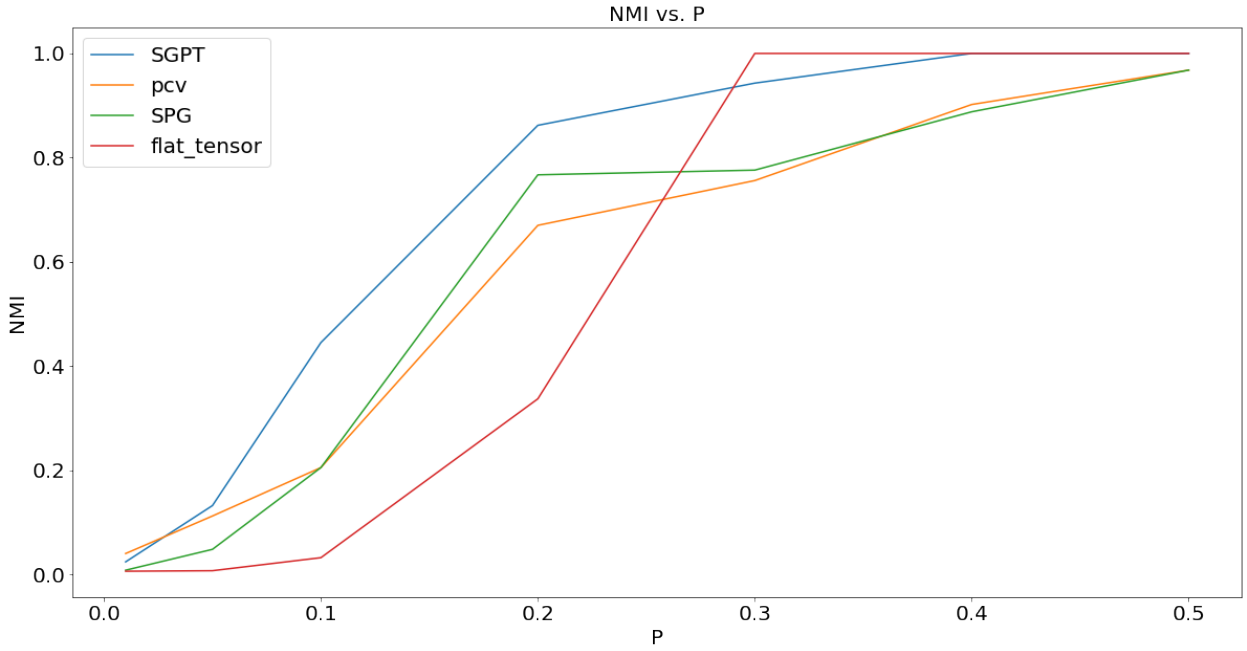

Figure 4.12: Our method compares favorably against baselines across different p, and is able to recover ground-truth partition in the presence of dominant noise.

timestamps), background entry probability $q = 0.005$, rank of decomposition $R = 5$, sizes of clusters for primary typed set (20, 20, 30, 40, 30 and the rest) and vary cluster probability p. Whenever needed, the algorithm is provided with k, p, q.

We analyze the sensitivity of the algorithms in regards to background noise. We sweep p = .01, .05, .1, .2, .3, .4, .5 and fix q = 0.005. The result is presented in Figure 4.12. our method performs competitively across different choices of p, thus robust to background noise. This is likely due to the fact that it utilizes all information sources jointly.

4.3.2 Results on Real-World Data

As the data size is prohibitive, we use stochastic gradient descent for optimizing our method objective. We then apply popular density-based clustering algorithm HDBSCAN [10] to the user embedding to obtain clusters. Next we apply our method to rank organized clusters and visualize top few clusters using our method.

Data Our Twitter data is from 2019/06/01 to 2019/09/25 discretized by day, tracking activities of users engaged with Canadian 2019 election politics - who tweeted at least once one of hashtags in Table 4.3 in this time frame. We scrape and construct the followership network of these users where each nonzero entry signifies that the user at column index follows the user at row index. We also create retweet and hashtag tensor that encode temporal usage of hashtags and retweet. Finally we extract hashtags that are used inside user self-description and users' self-revealed location as metadata items, where a nonzero means user's self description contains a certain hashtag or location. In the end, $|\mathcal{U}| = 69709, |\mathcal{M}| = 27930, |\mathcal{T}| = 117, |\mathcal{H}| = 1152$.

#cdnpoli	#canpoli
#cpc	#SenCA
#cdnleft	#pttory
#ptbloc	#gpc
#crtc	#goc
#BlackFaceTrudeau	#TrudeauMustResign
#BlackFace	#BrownFace
#ScheerLies	#elexn43
#NotasAdvertised	#TrudeauTheHyprocrite
#ptlib	#lpc
#ndp	#lavscam
#ptndp	#ptgreen
#cdnsen	#cpac
#CdesCom	#TrudeauBlackFace
#BrownFaceTrudeau	#TrudeauWorstPM
#Scheer	#Andysresume
#elxn43	#elxn19

Table 4.3: Hashtags used for crawling the data which are related to Canadian politics and the 2019 federal election.

Method We sequentially apply our method to obtain Figure 4.15, which provides an overview of the information landscape regarding 2019 Canadian Federal election. For our method, we select embedding dimension to be 5. Instead of random initialization, we initialized some of the factor matrices as SVD of flattened data matrices (user factor from

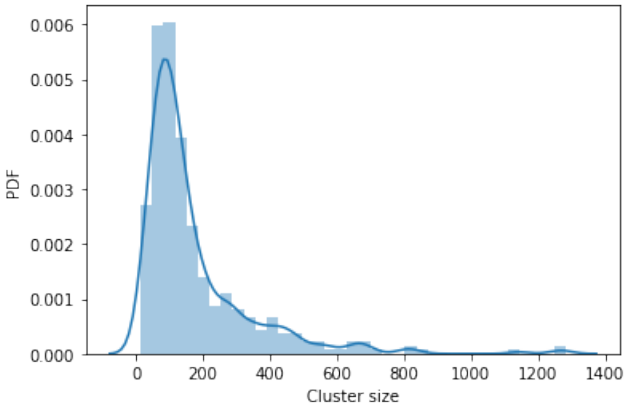

Figure 4.13: Cluster size distribution induced by MiniBatch KMeans clustering on our method user embedding

follower matrix, metadata item factor from metadata matrix, hashtag factor from flattened hashtag tensor). This has shown to speed up the training significantly. The positive and negative sample sizes for both training and evaluation are set to 1,000. We train our method until adjusted loss calculated from the fixed evaluation sample of positive and negative entries drops close to 0. This takes less than 1 hour on a normal laptop, therefore a significant improvement compared to methods in [86] since it cannot even run with non-stochastic optimizer due to prohibitive use of memory. We use the optimized factor matrix for user as user embedding, and apply MiniBatch KMeans clustering [78] to obtain the partition of users. We use the classical elbow method to obtain the number of clusters to use - 400 clusters. Figure 4.13 shows the cluster size distribution of obtained partition, where many tiny clusters are identified on the scale logarithmic to the total size of users. Next we apply our method to complete Figure 4.15. Note that in the visualization, we avoid showing generic hashtag by filtering out any word with ”poli” in it. Less heuristic approach can be used by modifying the ranking stage of our method, which we leave for future work.

Analysis We first explore results summarized in Figure 4.15 by comparing with weak labels: suspension records, whether user has been suspended in the past 6 months; and

Botometer scores [17], an API that applies natural language processing to gauge the likelihood of a Twitter user exhibiting bot behavior. In the top 13 organized clusters which comprise 641 users, 13 have record of being suspended at least once in the past 6 months, which is 4 times more likely than a random sample of users (calculated from total of 429 suspended in the pool of 69,709 users). In these groups, the average Botometer score of 0.286 is over 80% higher than the average of a random sample of users (population average is 0.156). Next, from Figure 4.15, we observe that the majority of top 13 organized groups are signified by the hashtag #TrudeauMustGo, and more intriguingly 2 groups signified by the hastag #MAGA are nested within multiple #TrudeauMustGo groups. Looking into the metadata of these #TrudeauMustGo groups, we found that the majority of them contain users that explicitly use #MAGA hashtag in their self description. The word cloud for hashtags in dense blocks identified using our method for top 13 organized groups and all groups are shown in Figure 4.14: #TrudeauMustGo and #blackface especially stand out against the background. More detailed group-level analysis is left for future work that pinpoints connection between #MAGA and #TrudeauMustGo groups and their influence over the network.

(a) Top 13 organized groups **(b)** All groups

Figure 4.14: Word cloud for top 13 organized groups vs. all groups

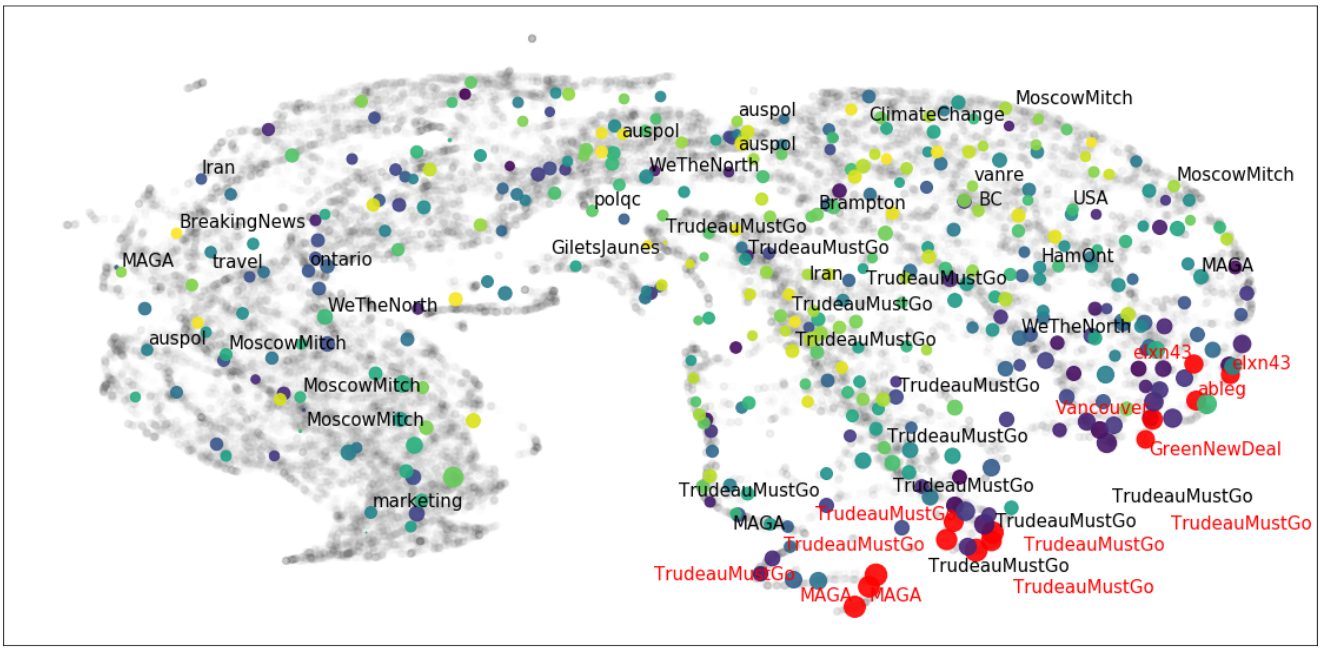

Figure 4.15: our method embeds users onto Euclidean space, marks top 13 organized clusters as red, and assigns descriptive hashtag to top 13 organized as well as randomly sampled clusters

Chapter 5

Conclusion

We have proposed the Embed-Cluster-Rank framework to help detect organized information campaign on social media. We have presented three instantiations of this framework: joint autoencoder, linear projection and aggregation, and tensor decomposition. Our empirical results on synthetic data verified our methods' ability to pick out dense subgraphs from large heterogeneous networks, which are likely to engage in coordinated activities. Our application of these methods on real-world Twitter data collected during the 2019 Canadian Federal Election revealed interesting and important results on the information landscape of Twitter regarding the election politics, as well as potential interference of foreign groups on the election.

Our framework is modular, interpretable, scalable and general. Each component of our three-stage approach can be upgraded with improving state-of-the-art, for example, the embedding phase could be extended to use deep graph neural network, the clustering phase could incorporate end-to-end deep clustering, and the ranking phase could use human-in-the-loop feedback. Because the framework learns low-dimensional representation for social media users as well as other entities in the data, tools such as interactive visualization can be developed to help experts understand what the model learns and whether there is bias in the model's results. All our methods scale linearly or sub-linearly

to the data size in runtime and memory usage, and are able to handle terabyte-scale social media data. Lastly, our framework handles data beyond Twitter, and can in principle be applied to any social media data where both the content and connections need to be taken into regard to understand the information landscape.

5.1 Directions for Future Work

The Embed-Cluster-Rank framework presented in this work facilitates detection of suspicious groups of users likely engaging in coordinated information campaigns on social media. However it is indispensable that human experts verify the user groups returned by our methods and check whether they indeed engage in spreading misinformation in a strategic manner. In some cases, it could be that the user groups is simply an interest group sharing content among themselves. Therefore, it is a promising direction to design applied natural language processing tools that can help human experts verify the truthfulness of information highly shared within or propagated from a user group. This could entail re-purposing open-domain question answering systems such as ORQA to achieve open-domain fact verification [50]. It is also an interesting direction to apply neural multi-hop reasoning to supply human experts with explicit reasons why certain facts are not true on social media [29].

On the other hand, our three-stage pipeline can be improved. For example, the embedding and clustering phase could be merged to enable end-to-end clustering using techniques such as deep graph neural network [93]. Furthermore, the truly end-to-end training would involve merging all three stages - embedding, clustering and ranking into a differential pipeline, where feedback from humans experts can directly affect all components of the pipeline. An example of such attempt is a deep reinforcement learning system developed at Facebook AI Research - Reinforced Integrity Optimizer (RIO) that catches hate speech [21].

In conclusion, our algorithmic contribution and practical real-world findings advance the technical toolkit we have to fight against one of the most pressing problem of 21^{st} century - online misinformation. Our proposal of a simple three-stage Embed-Cluster-Rank framework can provide a foundation for further development of tools against online misinformation. We believe that with the rapid advance of Artificial Intelligence technologies, such effort will keep the bad actors away and bring back trust, transparency and understanding to online communities.

Bibliography

[1] ADEWOLE, K. S., ANUAR, N. B., KAMSIN, A., VARATHAN, K. D., AND RAZAK, S. A. Malicious accounts: dark of the social networks. *Journal of Network and Computer Applications 79* (2017), 41–67.

[2] AKOGLU, L., MCGLOHON, M., AND FALOUTSOS, C. Oddball: Spotting anomalies in weighted graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2010), Springer, pp. 410–421.

[3] ALBERTS, D., AND HAYNES, R. The realm of information dominance: Beyond information war. In *First International Symposium on Command and Control Research and Technology* (1995).

[4] AYRES, R., BULLOCK, P., OKELLO, F., HARDING, B., PERDIGAO, A., AYRES, M. R., BULLOCK, M. P., ERHILI, B., HARDING, M. B., AND PERDIGAO, M. A. Information warfare: Planning the campaign.

[5] BADAWY, A., FERRARA, E., AND LERMAN, K. Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2018), IEEE, pp. 258–265.

[6] BANERJEE, A., BASU, S., AND MERUGU, S. Multi-way clustering on relation graphs. In *Proceedings of the 2007 SIAM international conference on data mining* (2007), SIAM, pp. 145–156.

[7] BARTHEL, M., SHEARER, E., GOTTFRIED, J., AND MITCHELL, A. The evolving role of news on twitter and facebook. *Pew Research Center 14* (2015), 1–18.

[8] BLONDEL, V. D., GUILLAUME, J.-L., LAMBIOTTE, R., AND LEFEBVRE, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment 2008*, 10 (2008), P10008.

[9] BOVET, A., AND MAKSE, H. A. Influence of fake news in twitter during the 2016 us presidential election. *Nature communications 10*, 1 (2019), 7.

[10] CAMPELLO, R. J., MOULAVI, D., AND SANDER, J. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining* (2013), Springer, pp. 160–172.

[11] CHARIKAR, M. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization* (2000), Springer, pp. 84–95.

[12] CHAU, D. H., PANDIT, S., AND FALOUTSOS, C. Detecting fraudulent personalities in networks of online auctioneers. In *European Conference on Principles of Data Mining and Knowledge Discovery* (2006), Springer, pp. 103–114.

[13] CHAVOSHI, N., HAMOONI, H., AND MUEEN, A. Identifying correlated bots in twitter. In *International Conference on Social Informatics* (2016), Springer, pp. 14–21.

[14] CHEN, X., SIN, S.-C. J., THENG, Y.-L., AND LEE, C. S. Why students share misinformation on social media: Motivation, gender, and study-level differences. *The journal of academic librarianship 41*, 5 (2015), 583–592.

[15] CHEN, Z., LI, X., AND BRUNA, J. Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415* (2017).

[16] CHENG, A., AND DICKINSON, P. Using scan-statistical correlations for network change analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2013), Springer, pp. 1–13.

[17] DAVIS, C. A., VAROL, O., FERRARA, E., FLAMMINI, A., AND MENCZER, F. Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web* (2016), pp. 273–274.

[18] DESHPANDE, Y., SEN, S., MONTANARI, A., AND MOSSEL, E. Contextual stochastic block models. In *Advances in Neural Information Processing Systems* (2018), pp. 8581–8593.

[19] DING, K., LI, J., BHANUSHALI, R., AND LIU, H. Deep anomaly detection on attributed networks.

[20] ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X., ET AL. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (1996), vol. 96, pp. 226–231.

[21] FAIR. *Training AI to detect hate speech in the real world*, 2020 (accessed December 15, 2020).

[22] FIRE, M., KATZ, G., AND ELOVICI, Y. Strangers intrusion detection-detecting spammers and fake profiles in social networks based on topology anomalies. *Human Journal 1*, 1 (2012), 26–39.

[23] FLORES-SAVIAGA, C. I., KEEGAN, B. C., AND SAVAGE, S. Mobilizing the trump train: Understanding collective action in a political trolling community. In *Twelfth International AAAI Conference on Web and Social Media* (2018).

[24] FORTUNATO, S., AND BARTHELEMY, M. Resolution limit in community detection. *Proceedings of the national academy of sciences 104*, 1 (2007), 36–41.

[25] GLEICHER, N. *How We Respond to Inauthentic Behavior on Our Platforms: Policy Update*, 2019 (accessed January 29, 2020).

[26] GLEICHER, N. *A simple algorithm for clustering mixtures of discrete distributions*, accessed February 29, 2020.

[27] GOVAERT, G., AND NADIF, M. Clustering with block mixture models. *Pattern Recognition 36*, 2 (2003), 463–473.

[28] GROVER, A., AND LESKOVEC, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (2016), ACM, pp. 855–864.

[29] HAMILTON, W., BAJAJ, P., ZITNIK, M., JURAFSKY, D., AND LESKOVEC, J. Embedding logical queries on knowledge graphs. *Advances in Neural Information Processing Systems 31* (2018), 2026–2037.

[30] HAMILTON, W., YING, Z., AND LESKOVEC, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems* (2017), pp. 1024–1034.

[31] HAWKINS, D. M. *Identification of outliers*, vol. 11. Springer, 1980.

[32] HOFSTETTER, C. R., BARKER, D., SMITH, J. T., ZARI, G. M., AND INGRASSIA, T. A. Information, misinformation, and political talk radio. *Political Research Quarterly 52*, 2 (1999), 353–369.

[33] HONG, D., KOLDA, T. G., AND DUERSCH, J. A. Generalized canonical polyadic tensor decomposition. *arXiv preprint arXiv:1808.07452* (2018).

[34] HOOI, B., SONG, H. A., BEUTEL, A., SHAH, N., SHIN, K., AND FALOUTSOS, C. FRAUDAR: bounding graph fraud in the face of camouflage. In *KDD* (2016), ACM, pp. 895–904.

[35] HOOI, B., SONG, H. A., BEUTEL, A., SHAH, N., SHIN, K., AND FALOUTSOS, C. Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), ACM, pp. 895–904.

[36] JAIN, P., AND KUMARAGURU, P. On the dynamics of username changing behavior on twitter. In *Proceedings of the 3rd IKDD Conference on Data Science, 2016* (New York, NY, USA, 2016), CODS '16, ACM, pp. 6:1–6:6.

[37] JIA, C., LI, Y., CARSON, M. B., WANG, X., AND YU, J. Node attribute-enhanced community detection in complex networks. *Scientific reports 7*, 1 (2017), 2626.

[38] KELLEY, S., GOLDBERG, M., MAGDON-ISMAIL, M., MERTSALOV, K., AND WALLACE, A. Defining and discovering communities in social networks. In *Handbook of Optimization in Complex Networks*. Springer, 2012, pp. 139–168.

[39] KIERS, H. A. A three–step algorithm for candecomp/parafac analysis of large data sets with multicollinearity. *Journal of Chemometrics: A Journal of the Chemometrics Society 12*, 3 (1998), 155–171.

[40] KIM, Y. M., HSU, J., NEIMAN, D., KOU, C., BANKSTON, L., KIM, S. Y., HEINRICH, R., BARAGWANATH, R., AND RASKUTTI, G. The stealth media? groups and targets behind divisive issue campaigns on facebook. *Political Communication 35*, 4 (2018), 515–541.

[41] KIPF, T. N., AND WELLING, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[42] KOLDA, T. G., AND BADER, B. W. Tensor decompositions and applications. *SIAM review 51*, 3 (2009), 455–500.

[43] KOLDA, T. G., AND HONG, D. Stochastic gradients for large-scale tensor decomposition. *arXiv preprint arXiv:1906.01687* (2019).

[44] KOLDA, T. G., AND HONG, D. Stochastic gradients for large-scale tensor decomposition. *SIAM Journal on Mathematics of Data Science 2*, 4 (2020), 1066–1095.

[45] KROONENBERG, P. M. *Three-mode principal component analysis: Theory and applications*, vol. 2. DSWO press, 1983.

[46] KRUSKAL, J. B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications 18*, 2 (1977), 95–138.

[47] KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. Trawling the web for emerging cyber-communities. *Computer networks 31*, 11-16 (1999), 1481–1493.

[48] KUMAR, S., CHENG, J., LESKOVEC, J., AND SUBRAHMANIAN, V. An army of me: Sockpuppets in online discussion communities. In *Proceedings of the 26th International Conference on World Wide Web* (2017), International World Wide Web Conferences Steering Committee, pp. 857–866.

[49] LANCICHINETTI, A., AND FORTUNATO, S. Limits of modularity maximization in community detection. *Physical review E 84*, 6 (2011), 066122.

[50] LEE, K., CHANG, M.-W., AND TOUTANOVA, K. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).

[51] LEE, V. E., RUAN, N., JIN, R., AND AGGARWAL, C. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*. Springer, 2010.

[52] LIM, S. H., CHEN, Y., AND XU, H. A convex optimization framework for bi-clustering. In *International Conference on Machine Learning* (2015), pp. 1679–1688.

[53] LIU, L., XU, L., WANGY, Z., AND CHEN, E. Community detection based on structure and content: A content propagation perspective. In *2015 IEEE International Conference on Data Mining* (2015), IEEE, pp. 271–280.

[54] LORD, C. *Political Warfare and Psychological Operations: Rethinking the US Approach*. DIANE Publishing, 1989.

[55] MALM, A., AND BICHLER, G. Networks of collaborating criminals: Assessing the structural vulnerability of drug markets. *Journal of Research in Crime and Delinquency 48*, 2 (2011), 271–297.

[56] MARCELLINO, W., SMITH, M. L., PAUL, C., AND SKRABALA, L. Monitoring social media. *Lessons for Future Department of Defense Social Media Analysis in Support of Information Operations, Rand, Santa Monica* (2017).

[57] MARWICK, A., AND LEWIS, R. Media manipulation and disinformation online. *New York: Data & Society Research Institute* (2017).

[58] MCINNES, L., HEALY, J., AND MELVILLE, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[59] MCINTOSH, E. A fake justin trudeau sex scandal went viral. canada's election-integrity law can't stop it. *News, Politics, Canada's National Observer*.

[60] MESNARDS, N. G. D., AND ZAMAN, T. Detecting influence campaigns in social networks using the ising model. *arXiv preprint arXiv:1805.10244* (2018).

[61] MISKIMMON, A., O'LOUGHLIN, B., AND ROSELLE, L. *Strategic narratives: Communication power and the new world order*. Routledge, 2014.

[62] MITCHELL, A., GOTTFRIED, J., KILEY, J., AND MATSA, K. E. Political polarization and media habits. *Pew Research Center* (Oct 2014).

[63] MUELLER, R. S., AND CAT, M. W. A. *Report on the investigation into Russian interference in the 2016 presidential election*, vol. 1. US Department of Justice Washington, DC, 2019.

[64] NEUMANN, S. Bipartite stochastic block models with tiny clusters. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 3867–3877.

[65] ORR, C. A new wave of disinformation emerges with anti-trudeau hashtag. *Election Integrity Reporting Project, Canada's National Observer*.

[66] OSELEDETS, I. V. Tensor-train decomposition. *SIAM Journal on Scientific Computing 33*, 5 (2011), 2295–2317.

[67] PALEN, L., AND LIU, S. B. Citizen communications in crisis: anticipating a future of ict-supported public participation. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM, pp. 727–736.

[68] PANDIT, S., CHAU, D. H., WANG, S., FALOUTSOS, C., AND FALOUTSOS, C. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 201–210.

[69] POLICY, T. P. Update on twitter's review of the 2016 us election. *Retrieved April 15* (2018), 2018.

[70] PRAKASH, B. A., SRIDHARAN, A., SESHADRI, M., MACHIRAJU, S., AND FALOUTSOS, C. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD (2)* (2010), vol. 6119 of *Lecture Notes in Computer Science*, Springer, pp. 435–448.

[71] RACHEL SANDLER, B. I. *Twitter CEO Jack Dorsey reportedly shared at least 17 tweets from a Russian troll*, 2018 (accessed June 8, 2020).

[72] REGNERI, M. Finding all cliques of an undirected graph. In *Seminar current trends in IE WS jun* (2007).

[73] RHEAULT, L., AND MUSULAN, A. Investigating the role of social bots during the 2019 canadian election. *Available at SSRN 3547763* (2020).

[74] ROBERTO ROCHA, C. N. *Researchers found evidence of Twitter troll activity in the last week of the federal election*, 2019 (accessed June 8, 2020).

[75] ROSVALL, M., AND BERGSTROM, C. T. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences 105*, 4 (2008), 1118–1123.

[76] ROTH, Y. *Information operations on Twitter: principles, process, and disclosure*, 2019 (accessed January 29, 2020).

[77] SAVAGE, D., ZHANG, X., YU, X., CHOU, P., AND WANG, Q. Anomaly detection in online social networks. *Social Networks 39* (2014), 62–70.

[78] SCULLEY, D. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web* (2010), pp. 1177–1178.

[79] SHETTY, J., AND ADIBI, J. Discovering important nodes through graph entropy the case of enron email database. In *Proceedings of the 3rd international workshop on Link discovery* (2005), ACM, pp. 74–81.

[80] SHIN, K., ELIASSI-RAD, T., AND FALOUTSOS, C. Corescope: graph mining using k-core analysis—patterns, anomalies and algorithms. In *2016 IEEE 16th International Conference on Data Mining (ICDM)* (2016), IEEE, pp. 469–478.

[81] SHIN, K., HOOI, B., AND FALOUTSOS, C. M-zoom: Fast dense-block detection in tensors with quality guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2016), Springer, pp. 264–280.

[82] SHIN, K., HOOI, B., KIM, J., AND FALOUTSOS, C. D-cube: Dense-block detection in terabyte-scale tensors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (2017), ACM, pp. 681–689.

[83] SHIN, K., HOOI, B., KIM, J., AND FALOUTSOS, C. Densealert: Incremental dense-subtensor detection in tensor streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), pp. 1057–1066.

[84] SHU, K., CUI, L., WANG, S., LEE, D., AND LIU, H. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 395–405.

[85] SHU, K., SLIVA, A., WANG, S., TANG, J., AND LIU, H. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter 19*, 1 (2017), 22–36.

[86] SORBER, L., VAN BAREL, M., AND DE LATHAUWER, L. Structured data fusion. *IEEE Journal of Selected Topics in Signal Processing 9*, 4 (2015), 586–600.

[87] STARBIRD, K. Disinformation's spread: bots, trolls and all of us. *Nature 571*, 7766 (2019), 449.

[88] STARBIRD, K., AND PALEN, L. Voluntweeters: Self-organizing by digital volunteers in times of crisis. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2011), ACM, pp. 1071–1080.

[89] STATISTA. *Usage of social media as a news source worldwide 2020*, 2020 (accessed September 2, 2020).

[90] STEWART, L. G., ARIF, A., AND STARBIRD, K. Examining trolls and polarization with a retweet network.

[91] STONE-GROSS, B., COVA, M., CAVALLARO, L., GILBERT, B., SZYDLOWSKI, M., KEMMERER, R., KRUEGEL, C., AND VIGNA, G. Your botnet is my botnet: analysis of a botnet takeover. In *Proceedings of the 16th ACM conference on Computer and communications security* (2009), ACM, pp. 635–647.

[92] TONG, H., AND LIN, C.-Y. Non-negative residual matrix factorization with application to graph anomaly detection. In *Proceedings of the 2011 SIAM International Conference on Data Mining* (2011), SIAM, pp. 143–153.

[93] TSITSULIN, A., PALOWITCH, J., PEROZZI, B., AND MÜLLER, E. Graph clustering with graph neural networks. *arXiv preprint arXiv:2006.16904* (2020).

[94] TUCKER, L. R. Implications of factor analysis of three-way matrices for measurement of change. *Problems in measuring change 15* (1963), 122–137.

[95] WALTZMAN, R. The weaponization of information. the need for cognitive security. testimony presented before the senate armed services committee, subcommittee on cybersecurity on april 27, 2017.[ ]. *: https://www. rand. org/pubs/testimonies/CT473. html* (2017).

[96] WANG, H., ZHOU, C., WU, J., DANG, W., ZHU, X., AND WANG, J. Deep structure learning for fraud detection. In *2018 IEEE International Conference on Data Mining (ICDM)* (2018), IEEE, pp. 567–576.

[97] WANG, H., ZHOU, C., WU, J., DANG, W., ZHU, X., AND WANG, J. Deep structure learning for fraud detection. In *2018 IEEE International Conference on Data Mining (ICDM)* (Nov 2018), pp. 567–576.

[98] WANG, J., LEVY, S., WANG, R., KULSHRESTHA, A., AND RABBANY, R. Sgp: Spotting groups polluting the online political discourse. *arXiv preprint arXiv:1910.07130* (2019).

[99] WANG, J., WANG, R., KULSHRESTHA, A., AND RABBANY, R. Anomaly detection with joint representation learning of content and connection. *arXiv preprint arXiv:1906.12328* (2019).

[100] WILSON, T., ZHOU, K., AND STARBIRD, K. Assembling strategic narratives: Information operations as collaborative work within an online community. *Proceedings of the ACM on Human-Computer Interaction 2*, CSCW (2018), 183.

[101] XU, J., WU, R., ZHU, K., HAJEK, B., SRIKANT, R., AND YING, L. Jointly clustering rows and columns of binary matrices: Algorithms and trade-offs. In *The 2014 ACM international conference on Measurement and modeling of computer systems* (2014), pp. 29–41.

[102] YANG, T., CHI, Y., ZHU, S., GONG, Y., AND JIN, R. Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning 82*, 2 (2011), 157–189.

[103] ZHANG, D., YIN, J., ZHU, X., AND ZHANG, C. Attributed network embedding via subspace discovery. *Data Mining and Knowledge Discovery 33*, 6 (2019).

[104] ZHANG, S., ZHOU, D., YILDIRIM, M. Y., ALCORN, S., HE, J., DAVULCU, H., AND TONG, H. Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining* (2017), SIAM, pp. 570–578.

[105] ZHOU, X., AND ZAFARANI, R. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315* (2018).

[106] ZHOU, X., ZAFARANI, R., SHU, K., AND LIU, H. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), pp. 836–837.