



Characterizing and locating polarized communities in signed networks

Jason Niu¹ · Ahmet Erdem Saryüce¹

Received: 19 November 2024 / Revised: 1 August 2025 / Accepted: 15 August 2025 /
Published online: 10 November 2025
© The Author(s) 2025

Abstract

Extreme polarization stands as a crucial concern for fostering a healthier web ecosystem. Locating the polarized groups is pivotal in this context. These groups involve nodes forming robust agreements with each other and engaging in collective conflicts with other groups. Previous studies tackle this problem by focusing on the balanced subgraphs in which all (or small) cycles have an even number of negative edges. However, balanced subgraphs in real-world signed networks are often not inherently polarized, such as those with solely positive edges, and any method that targets balanced subgraphs results in sizable communities with dominantly positive interactions. Building on this concern, we propose to utilize cohesion to find polarized subgraphs in this work. Specifically, we identify pairs of cohesively polarized communities where each node within a community has many positive connections with the nodes in the same community and numerous negative connections with the nodes in the opposing community. We introduce a novel measure, called dichotomy, to capture both cohesion and polarization in a given pair of polarized communities. We show that optimizing dichotomy is NP-hard. As a heuristic approach, we employ balanced triangles to develop a hierarchical dense subgraph discovery algorithm, called atom decomposition, that establishes effective seedbeds for polarized communities in signed networks. To address the challenges posed by real-world signed networks, we introduce two additional algorithms to find polarized communities: photon and electron decompositions. Photon decomposition filters out the nodes that engage in unbalanced triangles and yields numerous cohesively balanced communities. Electron decomposition favors polarized triangles over positive triangles to find polarized communities with high dichotomy. Through comprehensive experiments, we demonstrate that our approaches excel in identifying cohesively polarized communities, surpassing the state-of-the-art methods across various metrics. We give interesting anecdotal findings by using our algorithms on a political network among governments in the Cold War era and a business network of company relationships/competitions. Overall, our algorithms exhibit greater effectiveness and efficiency than existing methods, rendering them practical for large-scale networks.

Keywords Balance · Cohesion · Polarized communities · Signed networks

✉ Jason Niu
jniu0881@gmail.com

Ahmet Erdem Saryüce
erdem@buffalo.edu

¹ University at Buffalo, Buffalo, NY, USA

1 Introduction

With the rise of the Internet, there is no shortage of social media platforms where people constantly form factions that conflict with each other. Extreme polarization in social media platforms impacts the health of public discourse and democracy. One important problem in this context is finding polarized groups. In recent years, detection and mitigation of polarized groups has attracted an extensive interest [1–11]. Polarized groups are often characterized as a pair of communities where nodes form strongly stable agreements in their own community and participate in collective conflicts with the nodes in the other community [12–14]. Considering the magnitude of today’s social media platforms, it is essential to devise scalable and practical algorithms to find polarized communities [12].

Signed networks are a powerful tool to model positive and negative interactions, such as friend-foe and trust-distrust relations [15, 16]. One classical measure to identify polarized groups in signed networks is the balance, which measures the stability according to the relative placement of positive and negative edges. Heider defined that a signed graph is balanced if each cycle in it contains an even number of negative edges [15]. A more practical measure is the partial balance, often defined as the ratio of the number of balanced triangles, $+++$ and $+--$, to the count of all triangles [16, 17]. Earlier works often considered (partially) balanced subgraphs as a proxy for polarized communities in signed networks [12–14, 18]. Those works define and optimize a measure, called *polarity*, which simply favors the good edges (positives within, negatives across), penalizes the bad edges (positives across, negatives within), and normalizes w.r.t. total size. Optimizing polarity results in large and (almost) balanced subgraphs. However, the balanced subgraphs in real-world signed networks are often dominated by positive interactions and hence do not offer a clear notion of conflict. The main reason for this behavior is that $+++$ triangles in real-world signed networks are significantly more abundant than $+--$ triangles, thus dominate the resulting balanced subgraphs with little to no polarization.

In this work, we propose to differentiate polarization from balance by using the notion of cohesion. We argue that any node in a polarized community pair should engage in a large number of agreements with the other nodes in the same community and also participate in many conflicts with the nodes in the other community. Accordingly, we hypothesize that cohesive (i.e., with high edge ratio) and (partially) balanced subgraphs can offer better proxies for polarized groups in real-world signed networks. To characterize the extent of polarization for a given pair of communities, we devise the *dichotomy* measure. It considers the polarity from [12], cohesion, and community size information to represent the agreement within the communities, the conflict across the communities, and the relative sizes of the communities. A subgraph with high dichotomy score contain two similar-sized cohesive communities with

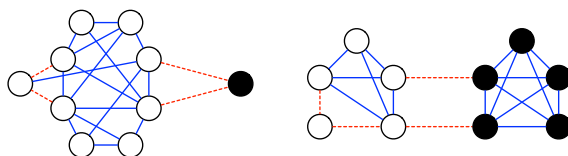


Fig. 1 In each community pair, white nodes denote the left community and black nodes are the right community. Positive edges are shown by blue straight lines and negative edges are denoted by dashed red lines. Although the right pair exhibits a better polarization, the polarity [12] and relative 3-balance scores [17] for the left pair and the right pair are the same (polarity is 3.2 and relative 3-balance is 1.0). Our dichotomy metric, on the other hand, assigns a higher score for the right pair (0.158 vs 1.422)

high polarity, where edges within a community are positive and edges across the communities are negative. Figure 1 compares dichotomy against other measures. As optimizing dichotomy is NP-hard, we study effective and practical heuristics to find polarized subgraphs with high dichotomy.

We leverage the balanced triangles to model both the cohesion and the polarization. Inspired by the truss decomposition [19], which finds cohesive regions with hierarchical relations in simple unsigned networks, we propose *atom decomposition* to find signed triangle-specific cohesive subgraphs. Atom decomposition provides good seedbeds to find balanced and polarized subgraphs. In an early empirical analysis, we show that (1) $+++$ triangles are significantly more common in the most cohesive subgraphs found by atom decomposition, and (2) not only are balanced triangles more abundant in the real-world signed networks, they are also significantly closer to each other than expected. Building on these observations, we propose two algorithms, *photon* and *electron decompositions*, to find highly-polarized subgraphs. *Photon decomposition* finds cohesively balanced subgraphs by filtering out the nodes that participate in many unbalanced triangles and applying atom decomposition with balanced triangles. The algorithm is parameterized to tune the filtering process and enable finding larger subgraphs with lower balance scores. Photon decomposition yields numerous communities with high balance and cohesion. *Electron decomposition* actively considers the nodes that are in many $+-$ triangles and few unbalanced triangles to find highly polarized pairs of communities. It is also parameterized to provide a trade-off between subgraph size and quality. Electron decomposition results in polarized communities with higher dichotomy scores than photon decomposition but less in quantity.

In an extensive experimental evaluation on real-world and synthetic networks, we show that photon and electron decomposition find higher quality communities than the state-of-the-art with respect to various measures. We give interesting anecdotal findings on a political network among governments in the Cold War era and a business network of company relationships/competitions. Last, but not least, our algorithms are more scalable than the previous methods and are practical for large-scale networks with more than 100M edges.

Our contributions can be summarized as follows ¹:

- **Separation of balance and polarization.** Previous state-of-the-art works consider balance to be equivalent to polarization which we show to be false. Polarized subgraphs are balanced but a balanced subgraph may not be polarized.
- **Dichotomy to model polarized communities.** To better define the polarization within a subgraph, we define the *dichotomy* measure, which considers the polarity metric (from [12]), cohesion, and partition sizes, to model the polarization. A subgraph with a high dichotomy score contains two highly-polarized communities with proportional sizes. We show that finding subgraphs with optimal dichotomy is NP-hard.
- **Improving cohesion through triangles.** We propose *atom decomposition* to find compact clusters of signed triangles. By utilizing this method, we show how the structure of real-world signed networks promote a triangle-based exploration for balanced and polarized communities.
- **Finding multiple balanced and polarized subgraphs.** We introduce two algorithms: *photon* and *electron decompositions*. The former finds numerous cohesively balanced subgraphs and the latter yields highly-polarized communities by focusing on polarized triangles. Our algorithms feature a filtering scheme to improve the balance and polarization before applying *atom decomposition* to find signed triangle-specific subgraphs.

¹ An earlier version of this work appeared in the companion proceedings of the ACM Web Conference 2023 [20]

- **Evaluation.** We compare our results against several baselines on real-world and synthetic networks with respect to several measures. Photon decomposition finds multiple cohesive and balanced communities. Electron decomposition yields polarized communities with the best dichotomy scores. We give interesting anecdotal findings by using our algorithms on a political network among governments in the Cold War era (Correlates of War) and a business network of company relationships/competitions (Relato Business). Our algorithms are more efficient than the alternative methods and are practical for large networks with more than 100M edges.

2 Preliminaries

We work on a simple and undirected signed graph $G = (V, E)$ where V is the set of nodes and $E = E^+ \cup E^-$ is the set of edges such that E^+ and E^- are the sets of positive and negative edges, respectively. The neighbors of a node v are denoted by $N(v)$. A triangle is a set of three nodes where each node is directly connected to the other two. We define each triangle in a signed network to be of type $+++$, $+-+$, $++-$, or $---$, where each $+$ and $-$ is the sign of a unique edge in the triangle. We utilize set notation in our pseudocode where $|$ is shorthand for ‘such that’.

In this work, we aim to find a subgraph that consists of two polarized communities where the nodes in the same community are connected with positive edges and the nodes from different communities are connected with negative edges. We denote the target subgraph, $S = (V_S, E_S)$, as the union of left and right communities, denoted by (V_L, E_L) and (V_R, E_R) , and the edges across the left and right communities, E_{LR} . Hence, $V_S = (V_L \cup V_R)$ and $E_S = (E_L \cup E_R \cup E_{LR})$. Any edge set with a sign superscript denotes the subset of edges with that sign, e.g., E_L^- is the set of negative edges in E_L . Without loss of generality, we assume that the larger of two communities (in number of nodes) is called the left community, hence $|V_L| \geq |V_R|$ by default.

Balance measures. A graph is balanced if its node set can be partitioned into two subsets such that each negative edge joins nodes from different subsets [21]. A popular measure for partial balance is the relative 3-balance—the ratio of the number of balanced triangles to the total number of triangles in the graph [17]. Triangles are preferred when characterizing the partial balance since triangles represent the strongest interactions [22]. However, relative 3-balance, along with other balance measures such as degree of balance and normalized frustration index [17], does not guarantee polarization as in the case of a graph with only positive edges.

Cohesive subgraphs and truss decomposition. The problem of truss decomposition is based on cohesive subgraph discovery. The cohesion of a subgraph is measured in terms of the **edge ratio**, which is the ratio of the number of edges in the subgraph to the number of node pairs. The most intuitive definition for a cohesive subgraph is a clique in which every pair of nodes is connected. However, it is often too rigid, resulting in small subgraphs with trivial significance. Thus, more relaxed forms of a cohesive subgraph have been proposed and k -truss is one such proposal that has been shown to be effective [19]:

Definition 1 A k -truss of G is a maximal connected subgraph of G where each edge participates in at least k triangles in the subgraph.

The truss number of an edge $e \in E$ (denoted by $K(e)$) is the largest k for which there is a k -truss that contains e . The edges in the k -truss are triangle-connected to each other,

which means any pair of edges e, e' in a k -truss either participates in the same triangle or connected to each other via a series of other edges $e = e_1, e_2, \dots, e_k = e'$ such that each consecutive edge pair e_i, e_{i+1} (for $1 \leq i < k$) shares a triangle [23, 24]. Truss decomposition is the process of finding the truss numbers of all the edges in a given graph through a peeling process which iteratively peels the edge with the lowest triangle count. The triangle count of an edge that is being peeled is assigned as its truss number [19]. For a given graph $G = (V, E)$, the space complexity of the truss decomposition is $O(|V| + |E|)$ and the time complexity is $O(\sum_{v \in V} |N(v)|^2)$. An edge can reside in multiple k -trusses with different k values, which results in a hierarchy where lower k -trusses contain (i.e., serve as a parent of) higher k -trusses. The terminal subgraphs which do not have a child in the truss hierarchy are called the leaf trusses. Leaves have the highest edge ratio and thus represent the strongest interactions. A truss has a depth of x from a leaf if its distance to the closest leaf is x . We use the fast hierarchy construction algorithms to obtain the actual subgraphs during the truss decomposition, for which the time complexity is the same as peeling [25].

3 Related work

Here we review prior works on finding balanced and polarized subgraphs and put them in context of our work.

Controversy in Unsigned Networks. Garimella et al. quantified controversy in unsigned networks by classifying two communities which are strongly separated [5]. They proposed a random walk-based method to measure the controversy, which is based on the probability that nodes will end in the same set after a random walk. **Random walk controversy (RWC)** is defined as follows:

$$\frac{(LL)(RR)}{(LL + RL)(LR + RR)} - \frac{(LR)(RL)}{(LR + RR)(LL + RL)} \quad (1)$$

where XY represents the number of walks which start in set X and end in set Y . We use the authors' code to calculate random walk controversy in polarized communities where negative edges are ignored. 50% of the nodes are randomly selected as starting nodes from each set and the walk terminates when another starting node is reached. Our work differs from Garimella et al.'s method by operating on signed networks and considering subgraphs with high cohesion across partitions.

Finding Balanced Communities. A related problem is to find a perfectly balanced subgraph for which the size of the node set is maximized. Figueiredo and Frota proposed a branch-and-cut approach [26] and introduced applications in risk management [27]. Ordozgoiti et al. proposed an algorithm which greedily removes the nodes from the graph until the graph is balanced, followed by adding back the nodes which do not impact balance [18]. The main difference between our work and these approaches is that we allow partial balance and also target high cohesion in the subgraphs.

For partial balance, Bonchi et al. introduced a spectral algorithm, *EIGENSIGN*, which computes the first eigenvector corresponding to the largest eigenvalue in the adjacency matrix and then discretizes its entries [12]. *EIGENSIGN* aims to find a pair of communities S that maximizes the **polarity** measure:

$$POL(S) = \frac{2 * (|E_L^+| + |E_R^+| - |E_L^-| - |E_R^-| + |E_{LR}^-| - |E_{LR}^+|)}{|V_L| + |V_R|} \quad (2)$$

where E_{LR} is the set of edges between the left and right sets. Bonchi et al. also adapted the greedy 2-approximation approach proposed by Charikar for finding the most cohesive subgraph [28], herein we refer to as *GREEDY*. It iteratively peels the node with the minimum difference between its positive and negative edge counts and the subgraph with the maximum polarity in this process is returned. In a related direction, Xiao et al. proposed an algorithm which, given two seed (disjoint) sets of nodes, outputs two disjoint subgraphs such that each contain one of the seed node sets and polarity between sets is maximized [14].

One drawback of the polarity measure is that it measures balance but not polarization.

Contrary to its name, a high value for the polarity measure does not always indicate the existence of two polarized communities. The polarity measure can be high even when one of the communities is empty—a subgraph whose edges are almost entirely positive is close to a complete agreement and there is no polarization at all. In our work, we remedy this issue by studying cohesively polarized communities. We compare our algorithms against *EIGNSIGN* and *GREEDY* in Sect. 6.

Detecting k Conflicting Groups. In this problem, the objective is to find k subsets of nodes which are positively connected within subsets and negatively connected between subsets. The problem of finding a pair of polarized communities is simply a special case of this problem for $k = 2$. Chu et al. proposed an algorithm which aims to find all groups which contain k polarized subgraphs in signed networks [29]. However, their algorithm only finds polarized communities within each local region, which often yields subgraphs of lesser quality. A better algorithm for the same problem is proposed by Tzeng et al. [13]. The authors proposed two spectral methods, *SCG-MA* and *SCG-R*, which operate on the leading eigenvector of the adjacency matrix and differ in their rounding schemes [13]. In our problem formulation, prior knowledge of k is not required, unlike [13]. Nevertheless, we compare our algorithms against *SCG-MA* and *SCG-R* in Sect. 6.

Correlation Clustering. The objective is to partition the nodes of a signed graph into a specific number of clusters such that there are mostly positive edges within clusters and mostly negative edges across clusters [30]. The 2-correlation-clustering problem is a specific case where the number of clusters is two. Bansal et al. introduced a 3-approximation algorithm which considers pairs of clusters for all $v \in V$ such that v and all its positively connected neighbors are in one cluster and all its negatively connected neighbors are in the other cluster [30]. As a follow-up, Bonchi et al. proposed returning the cluster pair which maximizes the polarity measure (Eq. 2) [12], which we refer to as *BANSAL*. Coleman et al. proposed the *PASTA-TOSS* algorithm for the 2-correlation-clustering problem which iteratively moves nodes across sets and returns the resulting distribution with the highest polarity [31]. We use *PASTA-TOSS* to partition the subgraphs returned by our algorithms into left and right communities. We compare our algorithms against *BANSAL* in Sect. 6.

Balanced Clique Enumeration. Although finding cohesive subgraphs is a fundamental graph mining problem for all kinds of networks with key applications [32–40], it is not much studied in signed networks. Existing studies only consider strict models such as cliques [41] or focus on streaming workloads [42] but have not leveraged the cohesion while finding polarized communities. Recently, Sun et al. and Chen et al. studied the problem of maximal balanced k -clique enumeration in signed networks [43, 44], which aims to find maximal cliques with no unbalanced triangles. Gao et al. introduced the maximal multipolarized clique model [45] where cliques are polarized with each other. However, the definition of

a clique is often too rigid, resulting in small subgraphs with trivial significance. Here we consider k -truss, a more relaxed model, to model cohesion.

k -truss-Based Models. There are a few recent works that attempt adapting the k -truss model for signed networks. Zhao et al. defined the signed k -truss as a subgraph where each edge takes part in at least $k - 2$ balanced triangles and there is no unbalanced triangles [46]. Their proposed solution, which we refer to as *ZHAO*, iteratively removes the edge which participates in at least one unbalanced triangle and results in the largest subgraph until the remaining graph does not contain any unbalanced triangles. Wu et al. introduced another model, signed (k, r) -truss, where each edge is in at least k balanced triangles and at most r unbalanced triangles [47]. Both Zhao et al.'s and Wu et al.'s truss-based models find a single edge-induced subgraph where only a subset of edges among the selected subset of nodes is considered as part of the subgraph [46, 47]. We compare our algorithms against *ZHAO* in Sect. 6.

4 Modeling polarized communities

In this paper, we aim to find cohesively polarized pairs of communities with non-trivial size. For cohesion, we use the *edge ratio* (see Sect. 2). We consider cohesion in our problem formulation to prevent weakly connected communities from being misrepresented as strongly polarized. This makes our algorithms more practical than the existing methods. We consider groups of nodes as a community, which are best modeled as vertex-induced subgraphs in which all the edges among the nodes are considered to be in the subgraph. We believe that this is more realistic than finding edge-induced subgraphs [46, 47] because it is often the set of entities, not specific connections, that one is interested in real-world applications.

Previous works on balanced subgraphs [12, 13, 30] consider balance as a direct link to polarization. However, balanced subgraphs do not have to be polarized as in the case of a subgraph with only positive edges. In this case, it is impossible to identify and limit the spread of polarization. Therefore, we consider a new approach based strictly on finding cohesively polarized communities of significant size.

Cohesively polarized communities. Polarized communities are two or more conflicting groups with positive connections in each group and negative connections in between the groups. *Polarized communities are balanced but the opposite is not true: a set of nodes with only positive edges is balanced but not polarized.*

To quantify the polarized communities, various measures have been proposed by earlier works, as explained in Sect. 3. The polarity measure (Eq. 2) has two disadvantages in modeling polarized communities: (1) It does not care about the size of each community separately, i.e., it is perfectly okay if one of the communities does not exist or have a trivial size; (2) It combines two different objectives, agreement inside and conflict across, hence lets one dominate the other when optimizing the measure. Aref et al. proposed two measures to remedy the second issue [48]: **cohesiveness** is defined as the fraction of positive edges to the total number of edges within sets and **divisiveness** is defined as the ratio of negative edges to the total number of edges between sets. Those two measures, however, cannot address the first issue and also do not penalize the negative edges inside each community and positive edges across the communities. A subgraph with high cohesiveness may have a lower edge ratio compared to another subgraph with similar cohesiveness. Likewise, high divisiveness is trivial if the two communities of the subgraph are connected only by a few, negative edges.

To address the issues mentioned above, we define **dichotomy**, which quantifies the quality of a given pair of polarized communities S by using the polarity, cohesion, and the ratio of the community sizes, as follows:

$$POL(S) \cdot \frac{|E_S|}{\binom{|V_S|}{2}} \cdot \frac{\min(|V_L|, |V_R|)}{\max(|V_L|, |V_R|)} \quad (3)$$

$POL(S)$ represents the polarity (Eq. 2) between the communities. Dichotomy measures the polarization of a subgraph by considering all the traits of a cohesively polarized subgraph. We want to find subgraphs which maximize the polarity and cohesion while having similar left and right set sizes so one community does not overwhelm the other. These traits each correspond to a part of the dichotomy formulation.

Our main problem is defined as follows:

Problem 1 Given an undirected signed graph G , find a pair of polarized communities with optimal dichotomy.

Bonchi et al. proved that optimizing polarity (Eq. 2) is NP-hard [12]. Therefore, by extension, Problem 1 is also NP-hard. We propose heuristics (Sect. 5) to find polarized communities with high dichotomy.

5 Algorithms

We start with *atom decomposition* (Sect. 5.1), which is at the core of our algorithms, and give an early empirical evaluation to understand the structure of real-world signed networks (Sect. 5.1.1). Motivated by our observations, we then provide *photon decomposition* to find balanced and dense subgraphs (Sect. 5.2). At the end, we propose *electron decomposition* for Problem 1 to find pair(s) of cohesively polarized communities with high dichotomy score (Sect. 5.3).

5.1 Using triangles for cohesion and balance

Triangles offer a unique opportunity to capture cohesion *and* balance at the same time. The literature is rich with the methods that use triangles to model cohesive subgraphs in various kinds of networks [23, 24, 49]. Here we introduce a new subgraph definition for cohesive subgraphs with respect to a given set of triangle types in signed networks.

Definition 2 A (k, Δ) -atom of G is a maximal triangle-connected subgraph of G where each edge participates in at least k triangles of type in Δ .

Δ is the set of triangle types for which the subgraphs are to be found. If Δ has all the four signed triangle types, (k, Δ) -atom is equivalent to the k -truss in the unsigned version of the network. For simplicity, we denote balanced triangles ($\Delta = \{+++ , +- -\}$) by *bal* and unbalanced triangles ($\Delta = \{++- , ---\}$) by *unbal*. We use Δ -atom as shorthand for (k, Δ) -atom when k is not relevant. We also define the Δ -**atom number** of an edge as the largest k for which there is a non-empty (k, Δ) -atom that contains the edge. As in the case of k -truss, all the (k, Δ) -atoms in a graph form a hierarchy where subgraphs with low k values contain the subgraphs with higher k values. The largest k value for which there exists a non-empty (k, Δ) -atom is the **maximum Δ -atom number** of the graph. A (k, Δ) -atom with the

Algorithm 1: ATOM (G, Δ)

Input: $G(V, E)$: graph, Δ : set of triangle types
Output: K : Δ -atom numbers

```

1  $T(e) \leftarrow 0 \forall \text{ edge } e \in E$ 
2 foreach triangle  $t \in G$  do
3   if  $\text{type}(t) \in \Delta$  then  $T(e)++\forall \text{ edge } e \in t$ 
4 Mark every  $e \in E$  as unprocessed
5 foreach unprocessed edge  $e$  with min.  $T(e)$  do
6    $K(e) \leftarrow T(e)$ 
7   foreach triangle  $t \mid \text{type}(t) \in \Delta \wedge e \in t$  do
8     if any edge  $e' \in t$  is processed then cont. foreach edge  $e' \in t \mid e' \neq e$  do
9       if  $T(e') > T(e)$  then  $T(e')--$ 
10    if  $T(e') > T(e)$  then  $T(e')--$ 
11   Mark  $e$  as processed
12 return  $K$ 

```

maximum atom number is the **maximum Δ -atom**. A (k, Δ) -atom that does not contain any (k', Δ) -atom such that $k' > k$ is a **leaf Δ -atom** (i.e., a leaf in the hierarchy).

To find (k, Δ) -atoms in a given graph (for all k values), we introduce *atom decomposition*, *ATOM* in short, in Algorithm 1. It takes as input a signed graph G and a set of triangle types Δ and finds the Δ -atom number of all the edges. Atom decomposition is inspired by the peeling-based truss decomposition algorithm. The triangle count of each edge is initialized to the number of triangles of type in Δ that the edge participates in (lines 1-3). Then, a peeling process is performed to iteratively peel the edge with the lowest triangle count (of types in Δ) from the graph (lines 5-11). In each iteration, the Δ -atom number of the edge of interest is assigned, triangle count of the neighboring edges (with higher value) is decremented, and the edge of interest is marked as processed. At the end, Δ -atom numbers of all the edges are returned (line 12). To construct the subgraphs and hierarchy, we consider the fast hierarchy construction algorithms in [25] (details are omitted for brevity). *ATOM* finds cohesive subgraphs with maximal number of given triangle types.

Time and space complexity. Unlike truss decomposition, *ATOM* only processes the triangles of a certain type, i.e., it performs truss decomposition on a smaller graph consisting of a subset of triangles from the original graph. This translates to the two additional checks in *ATOM* to ensure that the correct types of triangles are counted (line 3) and the correct types of triangles are used in the peeling (line 7). Those checks are performed in constant time and do not take additional space, thus the time and space complexities of *ATOM* are the same as truss decomposition (see Sect. 2).

5.1.1 Early evaluation

We perform an early evaluation of *ATOM* to understand the structure of real-world signed networks in terms of signed triangles (see Table 3 for the datasets). To characterize the significance of the results in real-world networks, we use a null model for comparison, proposed by Kirkley et al. [50]. In this model, the structure of the graph is not changed (i.e., unsigned version stays the same) and only the signs of the edges are randomized while the ratio of negative edges is preserved. For each real-world network, we generate 10 randomized networks and report the average values.

Placement of signed triangles. Previous studies have shown that balanced triangles (+++, +--) are more abundant than unbalanced triangles (++-, ---) and also more frequent

Table 1 Maximum Δ -atom numbers

Real-world Networks	+++		+--		++-		---		<i>bal</i>	
	Real	Exp.	Real	Exp.	Real	Exp.	Real	Exp.	Real	Exp.
Bitcoin	9.0	7.2	8.0	2.0	3.0	3.2	2.0	1.6	10.0	7.2
Wikielections	17.0	12.1	4.0	3.0	5.0	7.1	4.0	2.6	17.0	12.1
Tw-referendum	51.0	46.3	11.0	2.1	7.0	6.0	0.0	1.2	51.0	46.3
Slashdot	34.0	19.0	7.0	4.0	3.0	12.0	3.0	3.0	34.0	19.0
Epinions	104.0	69.9	10.0	5.7	8.0	26.6	8.0	3.0	104.0	69.9
Wikipolitics	28.0	23.0	3.0	3.0	4.0	7.0	6.0	2.0	28.0	23.0
Wikiconflict	28.0	5.0	14.0	16.0	15.0	9.0	22.0	14.4	28.0	18.0

For each Δ , real denotes the real value and exp. is the average value of corresponding randomized networks. *bal* denotes {+++ , +--}

than expected in real-world signed networks [17, 51, 52]. However, there has been limited research on the relative placement of these triangles in relation to each other. We now check how close the signed triangles of same (or similar) type are placed in real-world networks and in their randomized counterparts. To quantify this, we use the maximum Δ -atom numbers (Table 1) and the average proportion of the selected triangles in leaf atoms (Table 2).

A large maximum Δ -atom number implies that the edges in the corresponding subgraph participate in a larger number of triangles of type in Δ , thus the triangles of type Δ are placed close to each other. Table 1 presents the maximum atom numbers of real networks and randomized networks (on average) for five settings of triangles. If Δ has a balanced triangle (first, second, and last settings), the maximum atom number is significantly larger in the real network than in the randomized versions. On the other hand, if $\Delta = \{+-\}$, the real maximum atom number is smaller than the expected value in all but two graphs. The $+-$ results in Table 1 are particularly striking because even in the networks with a smaller fraction of $+-$ than $++-$ triangles (Tw-referendum, Epinions), the maximum $\{+-\}$ -atom number is larger than the maximum $\{++-\}$ -atom number in real networks!

Another proxy to quantify the closeness of triangles is the proportion of selected triangle types in the resulting subgraph. If the average proportion of triangles is larger than the expected, then the leaves of the real network are typically more cohesive than expected with respect to the corresponding triangle type(s). To measure that, we compute the leaf atoms with various Δ settings in both real and randomized networks, compute the fraction of selected triangle types in each, and calculate the average value (e.g., if $\Delta = \{+-\}$, we check the fraction of $+-$ triangles in the leaf $\{+-\}$ -atoms). Note that leaf atoms have the highest edge ratio when compared to its surroundings. Table 2 presents the results for five settings. +++ and +-- each (and together) yields subgraphs with a higher fraction of selected triangles in real networks than in randomized networks. Also, ++- exhibits smaller proportions than expected for four of seven networks. We observe that not only are balanced triangles more abundant in the real networks, they are also typically closer to each other than expected, thus tend to form good seedbeds for highly balanced and/or polarized subgraphs.

5.2 Improving the balance

In our early evaluation in Sect. 5.1.1, we found that +++ triangles typically dominate the triangle count and balanced triangles overall are much more common. Therefore, if we

Table 2 Average proportion of the corresponding triangle type (Δ) within each leaf in the Δ -atom hierarchy. Tw-referendum returned no applicable subgraphs for $\Delta = \{---\}$

Real-world networks	+++		+--		++-		---		<i>bal</i>	
	Real	Exp.	Real	Exp.	Real	Exp.	Real	Exp.	Real	Exp.
Bitcoin	0.98	0.69	0.82	0.17	0.48	0.37	0.31	0.59	0.97	0.86
Wikielections	0.88	0.51	0.32	0.13	0.32	0.41	1.00	0.03	0.89	0.77
Tw-referendum	1.00	0.86	0.80	0.01	0.21	0.16	n/a	0.57	1.00	1.00
Slashdot	0.99	0.48	0.63	0.14	0.14	0.42	0.87	0.54	0.99	0.98
Epinions	0.99	0.59	0.50	0.08	0.26	0.36	0.49	0.01	0.98	0.96
Wikipolitics	0.95	0.71	0.18	0.15	0.22	0.31	0.83	0.18	0.96	0.94
Wikiconflict	0.71	0.06	0.58	0.44	0.62	0.28	0.84	0.26	0.73	0.68

simply use $\{+++ , +--\}$ -atoms (which we refer as *bal*-atoms hereafter) to find polarized communities, $+++$ triangles may overwhelm $+--$ triangles. Another thing is that Δ -atom is designed to maximize the number of triangles of desired type(s). However, it does not prevent the undesired type(s) of triangles from forming. As mentioned in Sect. 2, all the subgraph definitions in this work are vertex-induced, implying that all the edges among the chosen set of nodes are considered to be part of the subgraph. If one is looking for balanced subgraphs, *bal*-atoms only ensure that each edge is part of many balanced triangles but does not enforce anything about participations in unbalanced triangles. The most cohesive *bal*-atoms may contain many unbalanced triangles, e.g., 27% of the triangles in leaf *bal*-atoms of Wikiconflict are unbalanced (see the last pair of columns in Table 2). Although Δ -atom provides a simple model to find subgraphs with many triangles of interest, it is not capable to find cohesively polarized pairs of communities. Therefore, we propose to explicitly avoid unwanted triangles with a pre-processing step.

Algorithm 2: PHOTON (G, α)

Input: $G(V, E)$: graph, α : threshold in $[0, 1]$

Output: K : processed *bal*-atom numbers

// $bal = \{+++ , +--\}$, $unbal = \{+-- , ---\}$

1 $badK \leftarrow ATOM(G, unbal)$

2 $E' \leftarrow \{e \in E \mid badK(e) \geq \max(badK) * \alpha\}$

3 **foreach** endpoint node $u \in E'$ **do**

4 Remove u from G

5 $K \leftarrow ATOM(G, bal)$

6 **return** K

We introduce *photon decomposition*, *PHOTON* in short, to find highly balanced subgraphs while also preserving the density (Algorithm 2). We rank the edges based on their unbalanced triangle counts, i.e., *unbal*-atom ($\{+-- , ---\}$ -atom) number, to determine which ones need to be avoided in the resulting subgraphs. *PHOTON* takes as input a signed graph G and a threshold α , and outputs the *bal*-atom numbers of the filtered graph. *PHOTON* filters the *bad* edges according to the α threshold, which is in the interval $[0, 1]$. In pre-processing, we remove the endpoints of edges from the graph whose *unbal*-atom number is at least α of the maximum *unbal*-atom number (lines 1-4). Then, *bal*-atom decomposition is applied on the remaining graph (line 5). The top subgraph is the maximum *bal*-atom obtained at the end. If

there are multiple subgraphs with the same highest k , then we choose the subgraph with the best dichotomy. The tunable threshold provides a trade-off between balance and subgraph size. Lower threshold values typically result in smaller subgraphs with more balance, while higher threshold values usually yield larger subgraphs with lower balance.

Time and space complexity. *PHOTON* has two calls to atom decomposition (lines 1 and 5) and a filtering step (lines 2-4), thus the time and space complexities are the same as atom decomposition.

Algorithm 3: ELECTRON (G, β)

Input: $G(V, E)$: graph, β : threshold in $[-1, 1]$
Output: K : processed $\{+-\}$ -atom numbers

```

1  $T(u) \leftarrow 0 \forall \text{ node } u \in V$ 
2 foreach triangle  $t \in G$  do
3   if  $t$  is  $+-$  then  $T(u)++ \forall u \in t$  else if  $t$  is unbal. then  $T(u)-- \forall u \in t$ 
5 while  $|V| > 0$  do
6    $f \leftarrow \min_{u \in V} \frac{T(u)}{\binom{|V|-1}{2}}$  // Min. friction value
7   if  $f \geq \beta$  then break // Threshold met
8    $N \leftarrow \{u \in V \mid \frac{T(u)}{\binom{|V|-1}{2}} = f\}$  // Min. friction nodes
9   foreach node  $u \in N$  do
10    foreach triangle  $t \in G \mid u \in t$  do
11      if  $t$  is  $+-$  then
12         $T(v)-- \forall \text{ node } v \in t \mid u \neq v$ 
13      else if  $t$  is unbal. then
14         $T(v)++ \forall \text{ node } v \in t \mid u \neq v$ 
15    Remove  $u$  from  $G$ 
16  $K \leftarrow ATOM(G, \{+-\})$ 
17 return  $K$ 

```

5.3 Finding polarized communities

We propose a conflict-based algorithm which builds upon the (k, Δ) -atom model. It is well-known that real-world signed networks typically have more positive than negative edges. Therefore, previous state-of-the-art algorithms often find subgraphs with mainly positive edges. Poor conflict for these models is one of the most common reasons for low dichotomy. These algorithms find subgraphs which are balanced but may not be polarized. Ideally, a polarized subgraph is balanced and contain two distinct communities with positive edges within and negative edges in between. As shown in Sect. 5.1.1, $+++$ triangles are typically much more common than $+-$ triangles in real-world signed networks. Hence, the balanced and cohesive subgraphs found by *bal*-atom and *PHOTON* often feature one large positively connected community and one much smaller opposing community, if at all. In order to find two polarized communities of non-trivial and comparable size, $+-$ triangles must be given a higher importance than $+++$. $\{+-\}$ -atom (i.e., polarized atom) may seem like the obvious choice in this context since it maximizes the number of $+-$ triangles. However, it does not prevent the presence of other types of triangles, which would reduce the polarization between the two communities.

To prevent the presence of undesirable triangles while keeping $+-$ triangles, we propose to filter out the nodes which are least correlated to the polarization. To this end, we first rank the nodes for removal by using the difference between the count of $+-$ and unbalanced triangles $\{++-, ---\}$. This difference favors balanced triangles over unbalanced triangles. Note that we specifically ignore $+++$ triangles to put more emphasis on the presence of $+-$ triangles. In addition, we want to factor in cohesion as it is part of the dichotomy. Let $T_{ub}^{+-}(u)$ be the number of $+-$ triangles minus the number of unbalanced triangles containing u . The maximum number of $+-$ triangles a node can participate in is $\binom{|V|-1}{2}$. To measure how important a node is for the total polarization of the graph, we define the *friction* as follows (takes values in $[-1, 1]$ interval):

$$\frac{T_{ub}^{+-}(u)}{\binom{|V|-1}{2}} \quad (4)$$

We give a tunable algorithm, *electron decomposition* (*ELECTRON* in short), that leverages the friction values of the nodes to find cohesively polarized communities with high dichotomy (Algorithm 3). *ELECTRON* takes as input a signed graph G and a threshold β , and outputs the $\{+-\}$ -atom numbers of the filtered graph. *ELECTRON* has a pre-processing stage which filters the input graph according to a specified threshold β (lines 1-13). The threshold β provides a trade-off between the subgraph size and conflict—higher β values typically result in smaller subgraphs with higher degrees of conflict. After the computation of T_{ub}^{+-} values for each node (lines 1-4), the nodes with minimum friction are iteratively removed from the graph until the friction of all the nodes satisfy the threshold β , which is in the interval $[-1, 1]$ (lines 5-15). Note that once a node is removed, the friction of the neighboring nodes may change (lines 9-15). At the end, $\{+-\}$ -atom decomposition is performed on the remaining graph (line 16). After the $\{+-\}$ -atoms are computed, the two communities (left and right sets) in each subgraph is obtained by Coleman et al.'s *PASTA-TOSS* algorithm, which partitions the nodes into two communities with highest polarity [31]. The top subgraph from our algorithm is the maximum $\{+-\}$ -atom. If there are multiple subgraphs with the same highest k , then we choose the subgraph with the best dichotomy.

Time and space complexity. In addition to the atom decomposition (line 16), we perform a filtering process (lines 1-15) as a peeling computation over nodes. T_{ub}^{+-} values of the nodes are maintained in a bucket (instead of friction values, because all have the same denominator), which ensures picking the node(s) with minimum friction value in constant time. This is a variation of the (1,3)-nucleus decomposition, proposed in [24], where the triangle counts of the nodes are used for peeling. The time complexity of this filtering process is the same as truss decomposition and its space complexity is $O(|V|)$. Hence, the time and space complexity of electron decomposition are the same as atom decomposition.

6 Experimental evaluation

Here we evaluate our algorithms on real-world networks against several baselines. We use various measures in evaluation. We answer the following research questions in the marked sections:

- How does the *PHOTON* and *ELECTRON* compare against the state-of-the-art methods in finding a single best and multiple polarized communities with respect to various measures (including dichotomy)? (Sects. 6.1 and 6.2)

Table 3 Statistics of the real-world networks

Net.	$ V $	$ E $	$ E^- / E $	# triangles
CO	180	1,803	0.22	12,249
BI	5,881	21,492	0.15	33,493
RB	44,674	129,668	0.17	607,536
WE	7,115	100,693	0.22	607,279
TW	10,884	251,406	0.05	3,120,811
SL	82,140	500,481	0.23	579,565
EP	131,580	711,210	0.17	4,910,076
WP	138,587	715,883	0.12	2,978,026
WC	116,717	2,026,646	0.62	13,831,236

- How does the α and β parameters in *PHOTON* and *ELECTRON* respectively, impact the size and quality (dichotomy) of the resulting subgraphs? What values should be chosen? (Sect. 6.3)
- What does the resulting subgraphs by *PHOTON*, *ELECTRON*, and state-of-the-art look like in a political network among governments in the Cold War era (Correlates of War) and a business network of company relationships/competition (Relato Business) datasets? (Sects. 6.4 and 6.5)
- How efficient are *PHOTON* and *ELECTRON* when applied to large-scale networks, especially when compared to the state-of-the-art methods? (Sect. 6.6)

Datasets. Important statistics of the used real-world signed networks are given in Table 3. Bitcoin (BI) and Epinions (EP) are who-trusts-whom networks of the users of Bitcoin OTC and Epinions.com, respectively [53]. Tw-referendum (TW) is built from Twitter data about the 2016 Italian Referendum: an interaction is negative if two users are classified with different stances, and is positive otherwise [54]. Slashdot (SL) contains friend/foe links between the users of Slashdot [53]. Wikiconflict (WC), Wikielections (WE), and Wikipolitics (WP) contain links between users from the English Wikipedia [55]. The edges of Wikiconflict represent positive and negative edit conflicts between users. Wikielections is the network of users that voted for and against each other in admin elections. Wikipolitics contains interpreted interactions between users that have edited pages about politics. We use CoW (CO) and Rel-Business (RB) as case studies which we describe in Sects. 6.4 and 6.5, respectively. We also use two large unsigned networks for scalability evaluation: LiveJournal and Orkut. We randomly assign the edge signs and perform runtime experiments (see Table 9).

Measures. To quantify the quality of the polarized communities, we consider relative 3-balance (Sect. 2), edge density/cohesion (Sect. 2), random walk controversy (RWC) (Eq. 1), polarity (Eq. 2), cohesiveness, divisiveness (Sect. 4), and dichotomy (Eq. 3) measures.

Baselines. We compare our algorithms against several state-of-the-art baselines: *BANSAL* [30], *EIGENSIGN* [12], *GREEDY* [12], *SCG-MA* and *SCG-R* ($k=2$) [13], and *ZHAO* [46] (detailed descriptions are given in Sect. 3). Since *SCG-MA* consistently finds subgraphs with lower balance compared to the other baselines, it is omitted in the results. We also use the truss decomposition [19], which we refer to as *TRUSS*, as a baseline (it is the same as $\{+++ , ++- , +-+ , ---\}$ -atom decomposition). For all baselines, we consider the vertex-

induced subgraphs that are obtained by including the endpoints of the edges outputted by the baseline.

Setup. All experiments are performed on a Linux operating system (v. Linux 3.10.0-1127) running on a machine with Intel(R) Xeon(R) Gold 6130 CPU processor at 2.10 GHz with 192 GB memory. We implemented our algorithms in C++ and compiled using gcc 6.3.0 at the -O2 level. **Code is available at** <https://tinyurl.com/polarizedDichotomy>. For each experiment, the complete set of results is available in the extended version [56].

6.1 Polarized communities

Here we compare *PHOTON* and *ELECTRON* against the state-of-the-art methods in finding the top pair of polarized communities. Table 4 shows the results obtained by the best baseline algorithm in terms of dichotomy (shown with an appended asterisk), *bal*-atom decomposition (denoted as *B-ATOM*), $\{+--\}$ -atom decomposition (shown as *P-ATOM*), *PHOTON* ($\alpha = 0.6$), and *ELECTRON* ($\beta = 0.1$) (we explain these default α and β values in Sect. 6.3). As explained in Sects. 5.2 and 5.3, the top subgraph from *PHOTON* is the maximum *bal*-atom obtained at line 5 of Algorithm 2 and the top one from *ELECTRON* is the maximum $\{+--\}$ -atom given at line 16 of Algorithm 3. If there are multiple subgraphs with the same highest k , then we choose the subgraph with the best dichotomy. For each graph and algorithm, we list the number of nodes in the left and right communities (larger one is the left w.l.o.g.), relative 3-balance, edge density, random walk controversy, polarity, cohesiveness, divisiveness, and dichotomy. Trivial subgraphs with empty right sets do not have a cohesiveness and divisiveness score.

PHOTON's top subgraph has the best relative 3-balance and edge density for 5 of 7 networks. These subgraphs often represent a closely connected positive cluster due to their small or empty right sets. However, these communities have little to no polarization, unlike the communities found by *P-ATOM* and *ELECTRON*. *P-ATOM* and *ELECTRON* consistently have the best dichotomy scores, often outperforming all other algorithms by far. Note that random walk controversy and polarity scores are not in line with the other measures in several cases, which suggests that they are not reliable measures.

6.2 Finding multiple communities

As real-world networks typically consist of multiple smaller polarized communities instead of a single large community, finding multiple high quality subgraphs is also an important problem. Our algorithms can find multiple pairs of polarized communities along with the maximal pair, which are ranked by their atom numbers. Each of these non-maximal subgraphs with high dichotomy can tell a unique story about polarization within the signed network. For *PHOTON* and *ELECTRON*, we consider all the subgraphs within a depth level of 2 from a leaf in the resulting hierarchy. For the baselines, we can find multiple sets of polarized communities by removing the resulting communities from the graph and reapplying the algorithm on the residual graph—we repeat this process to obtain at most 10 pairs of polarized communities.

Table 5 shows the average results for the multiple pairs of polarized communities obtained by the best baseline algorithm in terms of dichotomy, *bal*-atom decomposition (*B-ATOM*), $\{+--\}$ -atom decomposition (*P-ATOM*), *PHOTON* ($\alpha = 0.6$), and *ELECTRON* ($\beta = 0.1$). For each graph and algorithm, we list the number of obtained community pairs, average

Table 4 Results for the top pair of polarized communities obtained by our algorithms and baselines. $|V_L|$ and $|V_R|$ denote the number of nodes in left and right sets (larger is the left set w.l.o.g) and RWC is random walk controversy. For the baselines (*BANSAL*, *EIGENSIGN*, *GREEDY*, *SCG-R*, *ZHAO*, *TRUSS*), we show the one that gives the highest dichotomy and denote it with a trailing asterisk. *P-ATOM* denotes the $\{+ - -\}$ -atom decomposition (i.e., polarized atom decomposition) and *B-ATOM* denotes the *bal*-atom decomposition. The complete set of results is available in the extended version [56]

Network	Algorithm	$ V_L $	$ V_R $	Relative 3-balance	Density	RWC [5]	Polarity [12]	Cohesiveness [48]	Divisiveness [48]	Dichotomy
BI	<i>BANSAL</i> *	21	20	0.98	0.58	1.00	23.07	0.98	1.00	12.78
	<i>B-ATOM</i>	17	6	0.98	0.77	1.00	16.70	0.99	1.00	4.52
	<i>P-ATOM</i>	9	8	0.96	0.89	1.00	14.00	0.98	1.00	11.07
	<i>PHOTON</i>	10	3	1.00	0.95	-	11.38	1.00	1.00	3.24
	<i>ELECTRON</i>	19	17	1.00	0.66	1.00	23.00	1.00	1.00	13.52
WE	<i>BANSAL</i> *	408	7	0.79	0.12	0.83	39.49	0.88	0.89	0.08
	<i>B-ATOM</i>	65	0	0.89	0.66	-	39.08	-	-	0.00
	<i>P-ATOM</i>	155	3	0.58	0.24	-	3.00	0.53	1.00	0.01
	<i>PHOTON</i>	14	0	1.00	0.56	-	7.29	-	-	0.00
	<i>ELECTRON</i>	6	4	1.00	0.78	1.00	7.00	1.00	1.00	3.63
TW	<i>BANSAL</i> *	543	100	0.89	0.10	0.32	52.21	1.00	0.61	0.92
	<i>B-ATOM</i>	128	0	1.00	0.78	-	99.00	-	-	0.00
	<i>P-ATOM</i>	15	14	1.00	0.84	1.00	23.66	1.00	1.00	18.66
	<i>PHOTON</i>	120	0	1.00	0.79	-	93.70	-	-	0.00
	<i>ELECTRON</i>	15	14	1.00	0.84	1.00	23.66	1.00	1.00	18.66
SL	<i>SCG-R</i> *	26	6	1.00	0.23	0.64	7.12	1.00	0.92	0.38
	<i>B-ATOM</i>	77	0	0.99	0.80	-	60.36	-	-	0.00
	<i>P-ATOM</i>	49	7	0.94	0.56	1.00	7.00	0.55	1.00	0.56
	<i>PHOTON</i>	13	0	1.00	0.95	-	11.38	-	-	0.00
	<i>ELECTRON</i>	30	6	0.95	0.70	1.00	13.00	0.67	1.00	1.81
EP	<i>BANSAL</i> *	249	5	1.00	0.62	1.00	157.76	1.00	1.00	1.98
	<i>B-ATOM</i>	142	1	1.00	0.96	-	135.83	1.00	1.00	0.92

Table 4 continued

Network	Algorithm	$ V_L $	$ V_R $	Relative 3-balance	Density	RWC [5]	Polarity [12]	Cohesiveness [48]	Divisiveness [48]	Dichotomy
WP	<i>P-ATOM</i>	134	15	0.72	0.41	1.00	1.15	0.46	1.00	0.05
	<i>PHOTON</i>	142	0	1.00	0.96	–	135.03	–	–	0.00
	<i>ELECTRON</i>	28	6	0.96	0.70	1.00	5.29	0.50	1.00	0.80
	<i>BANSAL*</i>	599	11	0.94	0.08	0.58	48.45	0.98	0.41	0.07
	<i>B-ATOM</i>	86	0	0.95	0.72	–	58.72	–	–	0.00
	<i>P-ATOM</i>	10	3	1.00	0.77	–	9.23	1.00	1.00	2.13
	<i>PHOTON</i>	40	0	0.97	0.92	–	35.25	–	–	0.00
	<i>ELECTRON</i>	9	2	1.00	0.89	–	8.91	1.00	1.00	1.76
WC	<i>BANSAL*</i>	369	134	0.64	0.30	0.91	131.07	0.91	0.96	14.19
	<i>B-ATOM</i>	147	0	0.75	0.61	–	69.05	–	–	0.00
	<i>P-ATOM</i>	33	31	0.88	0.68	1.00	40.72	0.90	1.00	26.09
	<i>PHOTON</i>	29	11	0.88	0.70	1.00	25.95	0.95	1.00	6.93
	<i>ELECTRON</i>	40	36	0.98	0.62	1.00	46.08	0.98	1.00	25.68

Table 5 Results for the multiple pairs of polarized communities obtained by our algorithms and baselines. We report the averages, $|V_L|$ and $|V_R|$ denote the number of nodes in left and right sets (larger is the left set w.l.o.g.). For the baselines (*BANSAL*, *EIGENSIGN*, *GREEDY*, *SCG-R*, *ZHAO*, *TRUSS*), we show the one that gives the highest dichotomy and denote it with a trailing asterisk. *P-ATOM* denotes the $(+ - -)$ -atom decomposition (i.e., polarized atom decomposition) and *B-ATOM* denotes the *bal*-atom decomposition. The complete set of results is available in the extended version [56]

Network	Algorithm	# pairs	$ V_L $	$ V_R $	Relative 3-balance	Polarity [12]	Cohesiveness [48]	Divisiveness [48]	Dichotomy
BI	<i>BANSAL</i> *	10	53.8	5.1	0.91	10.76	0.93	0.77	1.79
	<i>B-ATOM</i>	7	23.3	4.1	0.95	13.25	0.97	1.00	2.03
	<i>P-ATOM</i>	9	18.7	12.4	0.94	11.71	0.80	1.00	6.14
	<i>PHOTON</i>	15	15.3	1.7	1.00	6.51	0.89	1.00	0.90
	<i>ELECTRON</i>	2	20.0	17.0	1.00	23.50	1.00	1.00	13.06
WE	<i>BANSAL</i> *	10	197.3	13.1	0.81	25.00	0.92	0.55	0.33
	<i>B-ATOM</i>	3	100.7	0.0	0.86	45.22	-	-	0.00
	<i>P-ATOM</i>	2	431.5	15.0	0.61	10.87	0.59	1.00	0.04
	<i>PHOTON</i>	9	77.6	1.8	0.97	6.94	0.95	1.00	0.01
	<i>ELECTRON</i>	2	8.0	4.0	1.00	8.07	1.00	1.00	3.10
TW	<i>BANSAL</i> *	9	294.1	15.0	0.98	54.49	1.00	0.69	0.25
	<i>B-ATOM</i>	3	137.0	0.0	1.00	102.74	-	-	0.00
	<i>P-ATOM</i>	3	34.3	25.0	1.00	36.17	1.00	1.00	17.69
	<i>PHOTON</i>	7	65.7	0.0	1.00	47.50	-	-	0.00
	<i>ELECTRON</i>	3	29.7	18.7	1.00	34.26	1.00	1.00	16.77
SL	<i>EIGENSIGN</i> *	4	191.0	80.2	0.93	38.85	0.96	0.95	0.97
	<i>B-ATOM</i>	58	17.0	0.4	0.98	9.42	0.94	1.00	0.17
	<i>P-ATOM</i>	7	42.3	5.9	0.92	5.76	0.75	1.00	0.37
	<i>PHOTON</i>	150	14.2	1.2	0.99	5.01	0.93	1.00	0.15

Table 5 continued

Network	Algorithm	# pairs	$ V_L $	$ V_R $	Relative 3-balance	Polarity [12]	Cohesiveness [48]	Divisiveness [48]	Dichotomy
EP	<i>ELECTRON</i>	2	34.0	6.0	0.95	15.22	0.70	1.00	1.83
	<i>EIGENSIGN*</i>	3	199.3	4.0	1.00	94.69	1.00	0.99	0.46
	<i>B-ATOM</i>	137	23.0	0.2	0.99	15.48	1.00	1.00	0.07
	<i>P-ATOM</i>	11	70.2	9.1	0.90	5.69	0.83	1.00	0.79
	<i>PHOTON</i>	233	18.3	0.2	1.00	12.13	0.97	1.00	0.05
WP	<i>ELECTRON</i>	3	33.0	6.7	0.95	7.73	0.54	1.00	1.08
	<i>BANSAL*</i>	10	159.1	1.5	0.97	40.85	0.99	0.61	0.04
	<i>B-ATOM</i>	26	31.8	0.1	0.95	21.26	0.98	1.00	0.02
	<i>P-ATOM</i>	10	51.4	8.1	0.87	6.54	0.94	1.00	0.44
	<i>PHOTON</i>	546	18.8	0.2	0.98	11.58	0.98	1.00	0.05
WC	<i>ELECTRON</i>	1	9.0	2.0	1.00	8.91	1.00	1.00	1.76
	<i>EIGENSIGN*</i>	7	91.6	51.7	0.54	46.65	0.86	1.00	9.91
	<i>B-ATOM</i>	70	24.8	1.1	0.34	5.02	0.51	1.00	0.06
	<i>P-ATOM</i>	154	17.6	3.1	0.34	3.21	0.51	1.00	0.56
	<i>PHOTON</i>	248	18.4	2.5	0.49	3.86	0.63	1.00	0.42
	<i>ELECTRON</i>	2	43.0	36.0	0.98	47.98	0.98	1.00	24.94

Table 6 Average results for *PHOTON* with $\alpha = 0.3, 0.6, 0.9$. R3B is the relative 3-balance

Networks	V			Density			R3B		
	0.3	0.6	0.9	0.3	0.6	0.9	0.3	0.6	0.9
BI	10.0	16.9	23.3	0.40	0.48	0.56	1.00	1.00	1.00
WE	n/a	79.3	155.3	n/a	0.32	0.42	n/a	0.97	0.94
TW	40.3	65.7	66.7	0.53	0.70	0.85	1.00	1.00	1.00
SL	14.8	15.4	17.3	0.46	0.41	0.53	1.00	0.99	0.99
EP	16.9	18.5	20.1	0.60	0.68	0.68	1.00	1.00	1.00
WP	20.7	18.9	25.0	0.46	0.67	0.66	0.99	0.98	0.95
WC	36.7	20.8	25.1	0.27	0.40	0.44	0.73	0.49	0.35

number of nodes in the left and right sets (larger set is the left w.l.o.g.), and average relative 3-balance, polarity, cohesiveness, divisiveness, and dichotomy. Trivial subgraphs with empty right sets are omitted.

PHOTON finds the most community pairs in 5 of 7 networks. This is a stark contrast to *ELECTRON*, which found at most 3 community pairs. Although *ELECTRON* often finds subgraphs of higher quality, *PHOTON* finds a plethora of communities, thanks to the detailed resulting hierarchy with many leaf subgraphs, with high relative 3-balance. Overall, *PHOTON* is effective for finding many cohesively balanced communities. *ELECTRON* has the best dichotomy score in 6 of 7 networks, often outperforming the second-best algorithm by a significant margin. Tw-referendum is the only dataset where *P-ATOM* slightly outperforms *ELECTRON*.

6.3 Threshold experiments

Here, we present the threshold experiments for *PHOTON* and *ELECTRON*. We consider all subgraphs found by our algorithms and compute their average scores.

To analyze the impact of the α threshold in *PHOTON*, we compare the results obtained for different α values. *PHOTON* finds highly balanced and dense subgraphs which are not necessarily polarized. Table 6 shows the average relative 3-balance results for *PHOTON* when α is set to 0.3, 0.6, and 0.9. In some networks, such as Wikipolitics and Wikiconflict, higher α values typically result in subgraphs with less balance. However, lower α values often result in smaller size subgraphs, many of which have less than 10 nodes—no subgraph with non-trivial size can be found in Wikielelections for $\alpha \leq 0.4$. In general, $\alpha = 0.6$ provides a good trade-off between subgraph size, density, and balance for all the graphs, hence can be chosen as the default.

For *ELECTRON*, we compare the average results for different values of β . Figure 2 shows the average number of nodes and dichotomy for all resulting subgraphs when β is 0.1, 0.2, and 0.3. *ELECTRON* avoids finding harmonious subgraphs with little to no polarization, unlike other algorithms. In general, a β value of 0.1 gives subgraphs with high dichotomy for the majority of the networks, hence suggested as default. One can choose slightly higher β values for higher degrees of conflict at the expense of smaller community sizes.

Fig. 2 Threshold experiments for *ELECTRON*. We plot the average number of nodes (left) and average dichotomy (right) for the β values of 0.1, 0.2, and 0.3

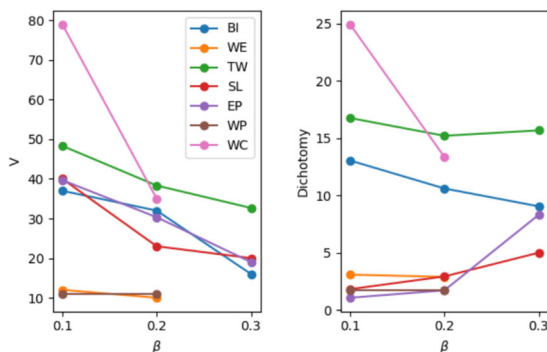


Table 7 Average results for our algorithms and baselines on CoW. NP is the number of pairs of polarized communities and the rest is defined as in Table 5. We only show the algorithms that have the best result for Pol., Coh., Div., and/or Dic. (highlighted in blue)

Algorithm	NP	$ V_L $	$ V_R $	Pol.	Coh.	Div.	Dic.
ZHAO	8	24.12	1.62	13.84	0.60	1.00	0.05
TRUSS	8	27.5	0.12	21.35	0.99	1.00	0.11
P-ATOM	3	17.0	4.67	6.64	0.76	1.00	3.50
PHOTON	6	15.17	0.83	12.86	0.99	1.00	0.47
ELECTRON	2	10.5	6.0	11.18	0.97	1.00	4.99

6.4 Case study: correlates of war

In this section, we present a case study to evaluate the algorithms on the CoW (Correlates of War) dataset [57]. Nodes in CoW are the countries, negative edges indicate a major conflict such as war, and positive edges represent alliances or peace treaties. The original data have 52 signed networks corresponding to different time periods between 1946 and 1999. Here we aggregate the data by choosing the most common sign for each edge (or the most recent, if tied). In the aggregated network, there are 180 nodes, 397 negative edges, 1,406 positive edges, and 12,249 triangles. The USA has the highest positive degree and RUS (Russia) has the highest negative degree.

Table 7 shows the average results for the obtained communities (as described in Sect. 6.2). We omit the baselines that do not perform best in any of the measures. Almost all of the baselines (except *SCG-R*) and *B-ATOM* find the same subgraph consisting of 33 countries strongly aligned with the USA—there are 554 positive and only 7 negative edges. Although *PHOTON* also finds this subgraph, the top subgraph it reports is a different alliance not found by the other algorithms. *PHOTON*'s top subgraph (the maximum *bal*-atom of the filtered graph), shown in Figure 3, has 16 African countries, such as NGA (Nigeria), LBR (Liberia), SEN (Senegal), with many currently taking part in the ECOWAS union due to their positive relationships with each other—there are 119 positive edges and only 1 negative edge.

P-ATOM and *ELECTRON* are the only algorithms that are able to find a subgraph with a significant dichotomy. It is the top subgraph found by *ELECTRON* (the maximum $\{+ - -\}$ -atom of the filtered graph) and shown in Figure 4. One set has the countries in alliance with the USA (Western bloc) and the other set contains the countries who had a positive relationship with RUS (Eastern bloc). This is expected since the time range mapped by CoW includes the cold war period. *P-ATOM* finds a smaller version of this subgraph. None of the baseline methods can find a similar subgraph that depicts the conflict between Western and Eastern blocs.

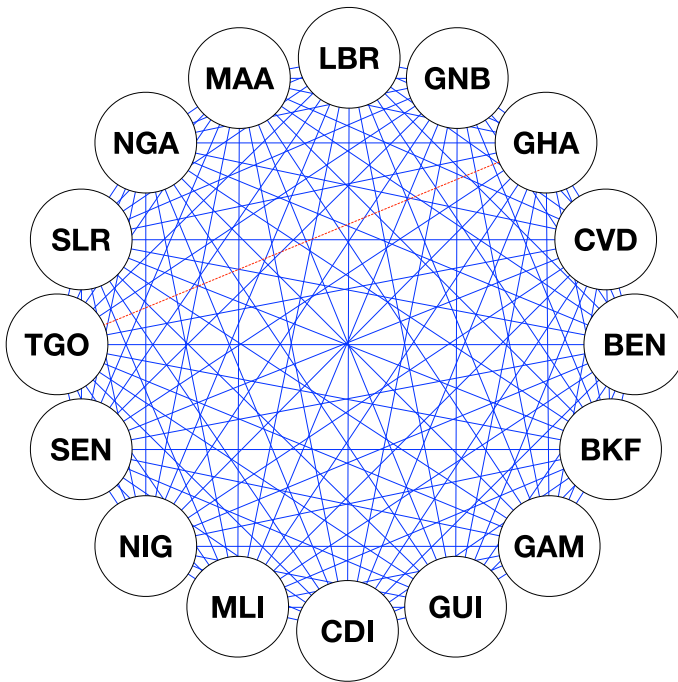
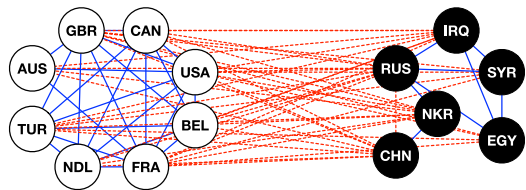


Fig. 3 The top subgraph found by *PHOTON* for CoW. Positive edges are the straight blue lines and negative ones are the dotted red lines

Fig. 4 The top subgraph found by *ELECTRON* in CoW. White and black nodes denote the left and right communities, respectively. Positive edges are denoted by straight blue lines and negative ones are shown by dashed red lines



Although a large subgraph may have high balance, it may not have high dichotomy. Subgraphs with low cohesion typically represent weaker alliances and/or conflicts where some countries may be neutral (not friendly nor hostile) with each other. On the other hand, subgraphs with high cohesion likely contain strong alliances between countries who may not be directly involved in the cold war. Subgraphs with high dichotomy consist of two strong alliances in a significant conflict with each other. Hence, capturing the polarization in the form of dichotomy is the key to measure the significance of the relationships within a subgraph.

6.5 Case study: relato business

In this section, we analyze business relationships within the Relato Business Graph Database (Rel-Business in short) [58]. Here, nodes are businesses, positive edges represent partnerships (from the partnership pages of companies), and negative edges correspond to competitors (co-bidders on AdWords). There are 44,674 nodes, 129,668 edges (22,303 of which are negative), and 607,536 triangles.

Table 8 Average results for the subgraphs obtained by our algorithms and baselines on Rel-Business. Notation is defined as in Table 7. We highlight the best results in blue

Algorithm	NP	$ V_L $	$ V_R $	R3B	Den.	Pol.	Coh.	Div.	Dic.
<i>BANSAL</i>	10	139.1	9.5	0.70	0.14	9.38	0.86	0.24	0.05
<i>EIGENSIGN</i>	2	228.5	0.0	0.55	0.10	20.57	-	-	0.00
<i>GREEDY</i>	7	180.29	9.29	0.65	0.12	12.89	0.92	0.12	0.04
<i>SCG-R</i>	0	-	-	-	-	-	-	-	-
<i>ZHAO</i>	7	175.14	0.0	0.71	0.26	21.43	-	-	0.00
<i>TRUSS</i>	38	15.95	4.97	0.05	0.59	2.77	0.09	1.00	0.32
<i>B-ATOM</i>	3	261.33	0.0	0.65	0.25	41.65	-	-	0.00
<i>P-ATOM</i>	3	317.67	1.33	0.53	0.30	14.92	0.67	1.00	0.12
<i>PHOTON</i>	7	41.14	0.14	0.93	0.47	9.36	0.96	1.00	0.00
<i>ELECTRON</i>	0	-	-	-	-	-	-	-	-

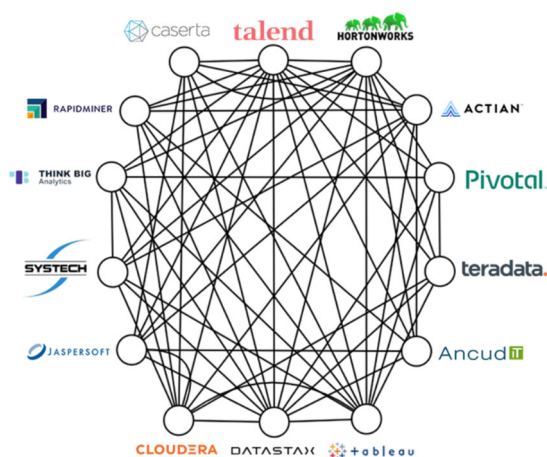
Fig. 5 One of *PHOTON*'s top subgraphs for Rel-Business.

Table 8 shows the average subgraph results found by our algorithms and baselines. Note that *SCG-R* and *ELECTRON* found no subgraphs with non-trivial size. *PHOTON* has the best average relative 3-balance with 0.93—the second best algorithm is able to give only 0.71, suggesting the superiority of *PHOTON* in finding highly balanced subgraphs. *PHOTON* also has the second best average density after traditional truss decomposition which has an average relative 3-balance of only 0.05.

In datasets where the polarization is minimal, finding strongly connected communities with high agreement is important for clustering purposes. With its relatively high cohesion and balance, *PHOTON* finds more strongly stable communities compared to the state-of-the-art baselines. For Rel-Business, *PHOTON* finds three different subgraphs with the highest *bal-atom* number of four. Figure 5 gives one of these subgraphs with no negative edges. This subgraph contains mainly data analysis companies who have strong partnerships with each other. The other two subgraphs feature positive connections within several marketing/advertising and online data security companies. Overall, *PHOTON* can distinguish between different types of companies through strongly correlated balanced communities.

Table 9 Runtime results (in sec.). Computations over 12h are shown by -

Algorithm	WC (117K 2M)	LiveJournal (4M 35M)			Orkut (3M 109M)		
		$ E^- / E $	$ E^- / E $	$ E^- / E $	$ E^- / E $	$ E^- / E $	$ E^- / E $
		0.05	0.25	0.5	0.05	0.25	0.5
BANSAL	11.5K	—	—	—	—	—	—
EIGENSIGN	344	10.2K	3.2K	607	8.3K	4.3K	5.0K
GREEDY	7.7K	—	—	—	—	—	—
SCG-R	2.8K	6.9K	2.4K	2.6K	2.7K	2.8K	3.4K
ZHAO	496	4.6K	1.9K	2.6K	20.2K	19.9K	21.7K
TRUSS	20	106	105	123	668	671	671
B-ATOM	38	352	315	316	2.0K	2.3K	1.9K
PHOTON	75	643	692	751	2.5K	3.3K	3.2K
ELECTRON	47	681	632	722	1.6K	2.0K	2.3K

6.6 Runtime performance

Finally we compare the runtimes of our algorithms and the baselines on the largest network in our dataset, Wikiconflict, and also two large unsigned networks: LiveJournal and Orkut. We randomly assign the edge signs in LiveJournal and Orkut, and used three different probabilities for negative edges, 0.05, 0.25, and 0.5, to see if the fraction of signs impact the results. We generate 10 random networks for each configuration and considered the average runtimes. We terminate the computations after 12h. *TRUSS*, *B-ATOM*, *PHOTON*, and *ELECTRON* finds multiple subgraphs through their peeling process. For the baselines, we find multiple sets of communities as explained in Sect. 6.2.

Table 9 gives the results. Our algorithms are consistently faster than all of the baselines besides *TRUSS*. *TRUSS* takes less time as it simply considers all the triangles without any specific checks for edge signs, which makes it ineffective in finding polarized communities. Both *PHOTON* and *ELECTRON* are consistently faster than the baselines, taking a few minutes for LiveJournal and less than an hour for Orkut. We do not observe any significant difference for different ratios of negative edges. Overall, our algorithms are not only effective in finding highly balanced (*PHOTON*) and highly polarized (*ELECTRON*) subgraphs, but they are also more efficient and practical than the existing methods.

7 Conclusion

Characterizing and finding polarized communities is an important problem to enable a healthier web ecosystem. Previous state-of-the-art methods simply focus on balanced subgraphs and optimize the ill-defined polarity measure which represents balance but not polarization. We define the dichotomy measure that improves upon polarity to better model the polarization. Given that maximizing dichotomy is NP-hard, we utilized balanced triangles to design a hierarchical dense subgraph discovery algorithm, named atom decomposition. This algorithm establishes effective foundations for polarized communities within signed networks. An early evaluation of atom decomposition suggested that not only are balanced triangles more abundant in the real networks, they are also typically closer to each other than expected. Motivated

by this, we introduced two additional algorithms, namely photon and electron decompositions, designed to identify polarized communities. Photon decomposition sifts through nodes engaged in unbalanced triangles, producing numerous cohesively balanced communities. Electron decomposition prioritizes polarized triangles over positive ones, identifying polarized communities with high dichotomy. Through extensive experiments, we showcase the superior performance of our approaches in identifying cohesively polarized communities, surpassing state-of-the-art methods across various metrics. Our algorithms yield compelling anecdotal findings when applied to a political network among governments in the Cold War era and a business network of company relationships/competitions. Overall, our algorithms demonstrate heightened effectiveness and efficiency compared to existing methods, making them suitable for large-scale networks.

We believe that our algorithms will be beneficial in real-world applications that are engaged with signed networks. For example, polarized subgraphs in online discussion platforms can point to the set of users that are heavily interested in a topic with strong opinions. Those users can be moderated with a better care to control the discourse and avoid extreme polarization. For future work, it would be interesting to adapt this problem for weighted signed networks where the weight of an edge represents the magnitude of agreement or conflict between two nodes. Another avenue for further inquiry is to improve the scalability of our algorithms for massive networks with billions of edges.

Acknowledgements This research was supported by NSF award OAC-2107089 and used resources from the Center for Computational Research at the University at Buffalo [59].

Author Contributions All authors wrote the main manuscript text, A.E.S. prepared figure 1, and J.N. prepared figures 2–5. All authors reviewed the manuscript.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Baldassarri D, Bearman P (2007) Dynamics of political polarization. *Am Sociol Rev* 72(5):784–811
2. Brundidge J (2010) Encountering difference in the contemporary public sphere: the contribution of the internet to the heterogeneity of political discussion networks. *J Commun* 60(4):680–700
3. Esteban JM, Ray D (1994) On the measurement of polarization. *Econometrica: Journal of the Econometric Society*, pp 819–851
4. Garimella K, De Francisci Morales G, Gionis A, et al (2017) Reducing controversy by connecting opposing views. In: *Proceedings of the tenth ACM international conference on web search and data mining*, pp 81–90
5. Garimella K, Morales GDF, Gionis A et al (2018) Quantifying controversy on social media. *ACM Trans Soc Comput* 1(1):1–27

6. Kumar S, Hamilton WL, Leskovec J, et al (2018) Community interaction and conflict on the web. In: Proceedings of the 2018 world wide web conference, pp 933–943
7. Mejova Y, Zhang AX, Diakopoulos N, et al (2014) Controversy and sentiment in online news. arXiv preprint [arXiv:1409.8152](https://arxiv.org/abs/1409.8152)
8. Liao QV, Fu WT (2014a) Can you hear me now? Mitigating the echo chamber effect by source position indicators. In: Proceedings of the 17th ACM conference on computer supported cooperative work & social computing. association for computing machinery, New York, NY, USA, CSCW '14, pp 184–196, <https://doi.org/10.1145/2531602.2531711>
9. Liao QV, Fu WT (2014b) Expert voices in echo chambers: effects of source expertise indicators on exposure to diverse opinions. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 2745–2754
10. Morales AJ, Borondo J, Losada JC et al (2015) Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos Interdiscip J Nonlinear Sci* 25(3):033114
11. Vydiswaran VV, Zhai C, Roth D et al (2015) Overcoming bias to learn about controversial topics. *J Am Soc Inf Sci* 66(8):1655–1672
12. Bonchi F, Galimberti E, Gionis A, et al (2019) Discovering polarized communities in signed networks. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 961–970
13. Tzeng RC, Ordozgoiti B, Gionis A (2020) Discovering conflicting groups in signed networks. *Adv Neural Inf Process Syst* 33:10974–85
14. Xiao H, Ordozgoiti B, Gionis A (2020) Searching for polarization in signed graphs: a local spectral approach. *Proc Web Conf* 2020:362–372
15. Heider F (1946) Attitudes and cognitive organization. *J Psychol* 21(1):107–112
16. Cartwright D, Harary F (1956) Structural balance: a generalization of heider's theory. *Psychol Rev* 63(5):277
17. Aref S, Wilson MC (2018) Measuring partial balance in signed networks. *J Compl Netw* 6(4):566–595
18. Ordozgoiti B, Matakos A, Gionis A (2020) Finding large balanced subgraphs in signed networks. *Proc Web Conf* 2020:1378–1388
19. Cohen J (2008) Trusses: cohesive subgraphs for social network analysis. National Security Agency Technical Report 16(3.1)
20. Niu J, Sariyüce AE (2023) On cohesively polarized communities in signed networks. *Companion Proc ACM Web Conf* 2023:1339–1347
21. Harary F (1953) On the notion of balance of a signed graph. *Mich Math J* 2(2):143–146
22. Bonacich P, Lu P (2012) Introduction to mathematical sociology. Princeton University Press
23. Huang X, Cheng H, Qin L, et al (2014) Querying k-truss community in large and dynamic graphs. In: SIGMOD, pp 1311–1322
24. Sariyüce AE, Seshadhri C, Pinar A, et al (2015) Finding the hierarchy of dense subgraphs using nucleus decompositions. In: WWW, pp 927–937
25. Sariyüce AE, Pinar A (2016) Fast hierarchy construction for dense subgraphs. *Proc VLDB Endowment* 10(3)
26. Figueiredo R, Frota Y (2013) An improved branch-and-cut code for the maximum balanced subgraph of a signed graph. arXiv preprint [arXiv:1312.4345](https://arxiv.org/abs/1312.4345)
27. Figueiredo R, Frota Y (2014) The maximum balanced subgraph of a signed graph: applications and solution approaches. *Eur J Oper Res* 236(2):473–487
28. Charikar M (2000) Greedy approximation algorithms for finding dense components in a graph. In: International workshop on approximation algorithms for combinatorial optimization, Springer, pp 84–95
29. Chu L, Wang Z, Pei J, et al (2016) Finding gangs in war from signed networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1505–1514
30. Bansal N, Blum A, Chawla S (2004) Correlation clustering. *Mach Learn* 56(1):89–113
31. Coleman T, Saunderson J, Wirth A (2008) A local-search 2-approximation for 2-correlation-clustering. In: European symposium on algorithms, Springer, pp 308–319
32. Alvarez-Hamelin JI, Barrat A, Vespignani A (2006) Large scale networks fingerprinting and visualization using the k-core decomposition. In: NIPS, pp 41–50
33. Angel A, Koudas N, Sarkas N, et al (2012) Dense subgraph maintenance under streaming edge weight updates for real-time story identification. arXiv preprint [arXiv:1203.0060](https://arxiv.org/abs/1203.0060)
34. Fratkan E, Naughton BT, Brutlag DL et al (2006) Motifcut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics* 22(14):e150–e157
35. Gibson D, Kumar R, Tomkins A (2005) Discovering large dense subgraphs in massive graphs. In: Proceedings of the 31st international conference on Very large data bases, Citeseer, pp 721–732

36. Lee VE, Ruan N, Jin R, et al (2010) A survey of algorithms for dense subgraph discovery. In: Managing and mining graph data, vol 40
37. Tsourakakis C, Bonchi F, Gionis A, et al (2013) Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. KDD '13
38. Gionis A, Tsourakakis CE (2015) Dense subgraph discovery: Tutorial. In: KDD, pp 2313–2314
39. Fang Y, Yu K, Cheng R et al (2019) Efficient algorithms for densest subgraph discovery. PVLDB 12(11):1719–1732
40. Sariyuce AE (2021) Motif-driven dense subgraph discovery in directed and labeled networks. Proc Web Conf 2021:379–390
41. Li R, Dai Q, Qin L et al (2019) Signed clique search in signed networks: concepts and algorithms. IEEE Trans Knowl Data Eng 33(2):710–27
42. Cadena J, Vullikanti AK, Aggarwal CC (2016) On dense subgraphs in signed network streams. In: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, pp 51–60
43. Sun R, Zhu Q, Chen C, et al (2020) Discovering cliques in signed networks based on balance theory. In: International conference on database systems for advanced applications, pp 666–674
44. Chen Z, Yuan L, Lin X et al (2020) Efficient maximal balanced clique enumeration in signed networks. Proc Web Conf 2020:339–349
45. Gao J, Hao F, Min G, et al (2021) Maximal multipolarized cliques search in signed networks. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 2227–2231
46. Zhao J, Sun R, Zhu Q, et al (2020) Community identification in signed networks: a k-truss based model. In: Proc of the 29th ACM international conf. on information & knowledge management, pp 2321–2324
47. Wu Y, Sun R, Chen C, et al (2020) Maximum signed (k, r)-truss identification in signed networks. In: Proc of the 29th ACM international conf. on information & knowledge management, pp 3337–3340
48. Aref S, Dinh L, Rezapour R et al (2020) Multilevel structural evaluation of signed directed social networks based on balance theory. Sci Rep 10(1):1–12
49. Huang X, Lu W, Lakshmanan LV (2016) Truss decomposition of probabilistic graphs: semantics and algorithms. In: SIGMOD, pp 77–90
50. Kirkley A, Cantwell GT, Newman M (2019) Balance in signed networks. Phys Rev E 99(1):012320
51. Guha R, Kumar R, Raghavan P, et al (2004) Propagation of trust and distrust. In: Proceedings of the 13th international conference on World Wide Web, pp 403–412
52. Leskovec J, Huttenlocher D, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1361–1370
53. Leskovec J, Krevl A (2014) SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>
54. Lai M, Patti V, Ruffo G, et al (2018) Stance evolution and twitter interactions in an italian political debate. In: International conference on applications of natural language to information systems, Springer, pp 15–27
55. Kunegis J (2013) Konect: The koblenz network collection. WWW Companion, pp 1343–1350
56. Niu J, Sariyuce AE (2024) Extended version. <https://tinyurl.com/polarizedExtended>
57. Doreian P, Mrvar A (2019) Structural balance and signed international relations. Journal of Social Structure 16(1)
58. Journey R (2017) Relato business graph database. <https://data.world/datasynndrome/relato-business-graph-database>
59. Center for Computational Research (2021), University at Buffalo. <http://hdl.handle.net/10477/79221>



Jason Niu earned his Ph.D. in Computer Science from the University at Buffalo, where his work focused on large-scale graph analytics and cohesive community detection under the guidance of Dr. A. Erdem Sariyüce. Jason has held research and engineering roles at Pacific Northwest National Laboratory and the Johns Hopkins University Applied Physics Laboratory.



Ahmet Erdem Sariyüce received the BS degree in computer engineering from Middle East Technical University and the PhD degree in computer science and engineering from Ohio State University. He is currently an Associate Professor at the University at Buffalo. His research interests include graph mining, learning, and management.