Discovering Opinion Intervals from Conflicts in Signed Graphs

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Online social media provide a platform for people to discuss current events and exchange opinions with their peers. While interactions are predominantly positive, in recent years we are increasingly witnessing conflicts based on different views and opinions. In this paper, we ask whether the conflicts in a network reveal a small and interpretable set of prevalent opinion ranges that explain the users' interactions. More precisely, we consider signed graphs, where the edge signs indicate positive and negative interactions of node pairs, and our goal is to infer opinion intervals that are consistent with the edge signs. We introduce an optimization problem that models this question, and we give strong hardness results and a polynomial-time approximation scheme by utilizing connections to interval graphs and the CORRELATION CLUSTERING problem. We further provide scalable heuristics and show that in experiments they yield more expressive solutions than CORRELATION CLUSTERING baselines. We also present a case study on a novel real-world dataset from the German parliament, showing that our algorithms can recover the political leaning of German parties based on co-voting behavior.

1 Introduction

Online social networks are essential parts of modern societies and are used by billions of people to discuss current events. Even though a majority of the interactions on such networks are positive, there has been an increase in conflicts, particularly as tensions among people with differing viewpoints grow.

As a result, gaining a deeper understanding of these conflicts has become essential. This question is often studied using *signed graphs* [25, 40, 39], where each edge has a sign that is either positive $(+)$ if two nodes interact amicably, or negative $(-)$ if the interaction is conflicting. A classic formulation used to analyze signed graphs and gain insights about the graph structure and potentially the opinions of nodes is the CORRELATION CLUSTERING problem [3]. In CORRELATION CLUSTERING, we ask to partition the nodes of a given signed graph into clusters, so as to maximize the number of edges that are consistent with the clustering (or minimize the number of inconsistent edges).

One drawback of the CORRELATION CLUSTERING formulation is that it makes hard decisions for the assignment of nodes into clusters and does not allow for a nuanced model in the presence of complex node interactions. For instance, in the landscape of European political parties, the opinions of representatives typically do not align perfectly with party lines; instead, members of ideologically-neighboring parties may agree on certain issues, while at the same time, members of the same party may disagree on other issues. Similar observations have been made for the US House of Representatives [2], however, when more parties are involved, modeling the interactions between representatives becomes increasingly complex.

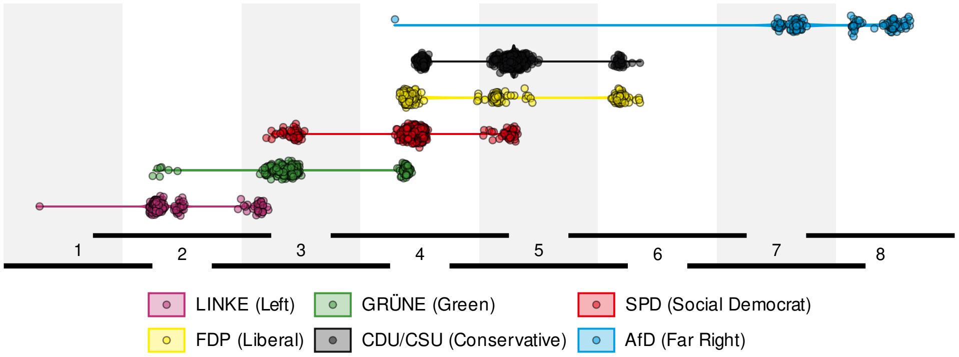


Figure 1: Visualization of our results on a signed graph based on co-voting behavior in the German parliament. We solved Opinion Interval Representation with 8 intervals, where consecutive intervals overlap (the intervals are visualized at the bottom of the figure). Each point corresponds to a German politician and assignments are visualized by mapping them to their party (y-axis) and interval (x-axis); for example, interval 3 contains politicians from SPD, GRÜNE, and LINKE. In each interval, points are shifted left or right, based on the number of positive and negative edges the corresponding politician shares with adjacent intervals.

In this paper, we introduce a novel problem to analyze signed graphs and discover structure that explains the nodes' interactions (conflicts and agreements) more accurately. Instead of assigning nodes to disjoint clusters, we seek to assign nodes to a small number of potentially-overlapping *opinion intervals*. The resulting structure can lead to meaningful insights and intuitive visualization (e.g., see Figure 1 — more discussion on the figure below). We show that our problem is more expressive than Correlation Clustering, thus resolving the drawback mentioned above. At the same time, our problem only requires the edge signs in the network as input, making it widely applicable.

Our results. First, we introduce the Opinion Interval Representation problem: Given a signed graph $G = (V, E^+ \cup E^-)$, assign an interval $I_v \subset \mathbb{R}$ to every vertex $v \in V$ such that we maximize the number of edges $\{u, v\} \in E^+$ with $I_u \cap I_v \neq \emptyset$ and $\{u, v\} \in E^-$ with $I_u \cap I_v = \emptyset$. In other words, if two nodes are connected by a positive edge, then their corresponding intervals should overlap, whereas if they are connected by a negative edge, then their intervals should be disjoint. Note that for a node v, we can think of I_v as the range of opinions that are acceptable to v and yield an amicable interaction; all opinions outside of I_v are not acceptable and yield a conflict. This problem is more expressive than the Correlation Clustering problem of Bansal et al. [3], and a related problem by Kermarrec and Thraves [29] as we explain below.

Second, we show that Opinion Interval Representation is NP-hard *even when the graph* $G^+ = (V, E^+)$ *induced by the positive edges forms a cycle*. In a sense, this is the strongest possible hardness result one could hope for since removing a single edge from the cycle $G^+ = (V, E^+)$ produces a path, for which intervals can always be assigned without any error. This implies that (unless $\mathsf{P} = \mathsf{NP}$) for Opinion Interval Representation there is no FPT algorithm that parameterizes by the number of required edge deletions and that the disagreement version of Opinion Interval Representation cannot be approximated within any multiplicative factor. It also provides novel insights into the hardness of finding forbidden induced subgraphs, which rules out several algorithm design approaches. Our reduction is based on a result of Cygan et al. [19], but making it work for cycles requires several new ideas and gets significantly more complicated. We provide an overview of the reduction in Section 2.1.

Third, we consider a constrained version of Opinion Interval Representation, where we are given a complete signed graph and parameters $\varepsilon > 0$ and $k = O(1)$. Now, we are only allowed to use k distinct intervals and each node must be assigned to one of them. This provides highly interpretable insights since the number of intervals is small. For this problem, we provide a polynomial-time approximation scheme (PTAS); specifically, we present an algorithm that computes a $(1 + \varepsilon)$-approximation in time $2^{O(k^2 \log(k/(\varepsilon\delta))/\varepsilon^3)} \cdot n$. This generalizes an algorithm by Giotis and Guruswami [24] that was developed for Correlation Clustering with a fixed number of clusters. We provide an overview of the PTAS in Section 2.2.

Fourth, from a practical point of view, we introduce heuristics that we describe in Section 3. Our heuristics are inspired by the PTAS above and include several practical improvements. In our experiments, we show that by using overlapping opinion intervals, our novel methods outperform state-of-the-art heuristics for CORRELATION CLUSTERING. On 8 real-world datasets, the results are at least 20% better and over 38% on average. This holds even when we use only 8 intervals, showing that even a small number of intervals yields expressive and interpretable representations. This result exemplifies that our problem yields more expressivity than CORRELATION CLUSTERING solutions.

Furthermore, we perform a case study on a novel dataset based on co-voting behavior in the German parliament, which we make publicly available. The output of our algorithm allows us to reconstruct the leaning of the political parties, as we demonstrate in Figure 1. Besides accurately reflecting the German political spectrum, the figure also reveals the coalition governments throughout the past decade (see Section 4). We stress that, due to the overlapping spectrum from the left to the right, finding such a structure would not be possible in existing problems like CORRELATION CLUSTERING.

We conclude the paper with several interesting questions for further research in Section 5, and present our proofs and additional experimental results in the supplementary material.

Related work. The OPINION INTERVAL REPRESENTATION problem is closely related to the SITTING ARRANGEMENT problem by Kermarrec and Thraves [29]: Given a signed graph $G = (V, E^+ \cup E^-)$, can we assign a vector $x_u \in \mathbb{R}^\ell$ to each $u \in V$ such that for all positive edges $\{u, v\} \in E^+$ and negative edges $\{u, w\} \in E^-$ the inequality $\|x_u - x_v\|_2 < \|x_u - x_w\|_2$ holds? Kermarrec and Thraves [29] presented several results for the case of $\ell = 1$, i.e., embedding G into the real line. Cygan et al. [19] improved upon this and showed that for a complete signed graph G such an assignment exists if and only if the subgraph induced by its positive edges $G^+ = (V, E^+)$ is a unit interval graph [45]. Given this characterization, we note that our problem is more expressive since we allow general (non-unit) intervals. Besides these theoretical insights, Pardo et al. [43, 44] provided heuristics for an optimization version that aims to minimize the number of violated constraints on the vectors x_u above. However, this objective is substantially different from ours and thus incomparable.

As mentioned before, CORRELATION CLUSTERING [3] is highly related to our work and is stated as follows: Given a signed graph $G = (V, E^+ \cup E^-)$, partition its vertices into disjoint clusters $C_1, \dots, C_k \subseteq V$ such that the number of positive edges within the clusters C_i and the number of negative edges between different clusters C_i and C_j, $i \neq j$, is maximized. Here, the value of k can be picked by the algorithm. CORRELATION CLUSTERING has received a lot of attention in the past two decades in social network analysis and image segmentation, spanning approximation algorithms [12, 46, 24, 18, 16, 15], more expressive formulations [6], and results in dynamic, online, and parallel settings [14, 13, 17]. There has also been continued interest in developing heuristics [42, 1, 47, 49, 35, 5, 8]. We refer to the book by Bonchi et al. [7] for more references.

Opinion formation models, such as the DeGroot model [21], the Friedkin–Johnsen model [22], or the bounded-confidence model [33, 20], are also related. These models assign a real-valued opinion to each node in a graph, which allows a more fine-grained understanding of conflicts than CORRELATION CLUSTERING. However, estimating the parameters of such models is highly challenging and requires more information than the edge signs of a signed graph [4, 37, 36]. Thus, our method is more easily applicable as it requires substantially less (and particularly less sensitive) data.

Below, we will also argue that Interval Editing problems are closely related. Specifically, in the (Unit) Interval Editing problem, the task is to transform an unsigned graph into a (unit) interval graph using a minimum number of edge deletions and insertions. This problem is known to be NP-hard already since the seminal work of Garey and Johnson [23] and it is fixed-parameter tractable (FPT) when parameterized by the number of edge insertions and deletions [10, 28, 48, 11]. We further discuss the relation of OPINION INTERVAL REPRESENTATION to these problems in Section 2.

Preliminaries. A *signed graph* $G = (V, E^+ \cup E^-)$ is given by its vertices V, positive edges E^+, and negative edges E^-, where $E^+ \cap E^- = \emptyset$. It is *complete* if $E^+ \cup E^- = \binom{V}{2}$. For $u \in V$, we write $N^+(u)$ to denote its neighbors in E^+ and $N^-(u)$ to denote its neighbors in E^-.

A graph $G = (V, E)$ is an *interval graph* if we can assign an interval $I_v = [s_v, t_v] \subset \mathbb{R}$ to all vertices $v \in V$ such that for all $u, v \in V$, it holds that $\{u, v\} \in E$ if and only if $I_u \cap I_v \neq \emptyset$. Additionally, we say that G is a *unit* interval graph if all intervals have length 1.

2 Problem Definition and Theoretical Results

In this section, we define our novel problem and state our main theoretical results.

Problem 2.1 (OPINION INTERVAL REPRESENTATION)**.** Given a signed graph $G = (V, E^+ \cup E^-)$, find a set $\mathcal{I} = \{I_v \subset \mathbb{R} \colon v \in V\}$ of non-empty, contiguous intervals that maximizes

$$\mathsf{agree}(G, \mathcal{I}) = \sum_{\{u,v\} \in E^+} \mathbb{1}(I_u \cap I_v \neq \emptyset) + \sum_{\{u,v\} \in E^-} \mathbb{1}(I_u \cap I_v = \emptyset), \tag{1}$$

where $\mathbb{1}(E)$ is indicator function, which takes value 1 if E is true and 0 otherwise.

Intuitively, the problem assigns an interval I_v to each vertex v and asks that two intervals overlap if their corresponding vertices are connected with a positive edge and do not overlap if their vertices are connected with a negative edge. To connect this problem to opinions, we may consider an interval I_v for a node v as the range of opinions that are acceptable to v. The length $|I_v|$ can further be seen as a measure of v's tolerance towards the opinion spectrum to the left and to the right.

We will refer to the formulation in Problem 2.1 as the *agreement* version of OPINION INTERVAL REPRESENTATION, which asks to satisfy as many edges as possible. We will also talk about the *disagreement* version, which aims to minimize the number of edges violating the interval assignment. Their complexity is the same for exact solutions, but they differ w.r.t. approximation guarantees.

Relationship to CORRELATION CLUSTERING. Next, we observe that OPINION INTERVAL REPRESENTATION is more expressive than CORRELATION CLUSTERING: First, consider any CORRELATION CLUSTERING solution $C_1, \dots, C_k$, and consider k non-overlapping intervals $I_1, \dots, I_k$. Now we assign each vertex in C_i to the same interval I_i. Thus, if $u, v \in C_i$, then their intervals overlap, and, if $u \in C_i$ and $v \in C_j$ for $i \neq j$, then their intervals do not overlap. This implies that the optimal solution of OPINION INTERVAL REPRESENTATION will always yield an agreement at least as large as for CORRELATION CLUSTERING. Second, our interval representation is strictly more expressive and, for instance, allows us to model non-transitive node relationships and this property is illustrated in Figure 2. This is neither possible in CORRELATION CLUSTERING nor in the structural balance theory of Harary [25], in both of which no cycle with exactly one negative edge can be represented without error.

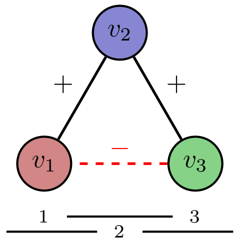


Figure 2: A triangle with one negative edge and its (exact) interval representation.

The case of complete graphs and relationship to Interval Editing. If G is a complete signed graph then it can be represented without error in OPINION INTERVAL REPRESENTATION if and only if $G^+ = (V, E^+)$ is an interval graph. That is because missing edges in G^+ correspond to negative edges in G (since G is complete). Thus, making the minimum number of edge deletions/insertions to turn G^+ into an interval graph is equivalent to flipping the minimum number of edge signs in G such that we have agreement for all edges. Hence, for complete graphs, we can rely on the rich literature on Interval Editing which asks for the minimum number of edge changes to G^+ such that it becomes an interval graph. The results of Cao [9] now imply that OPINION INTERVAL REPRESENTATION is FPT for complete graphs when only allowing a fixed number of sign changes (in one direction). However, in social networks this number will be large for real-world instances and thus these algorithms are not applicable in practice. Furthermore, our hardness results show that such FPT results are not possible in incomplete graphs when parameterized by the number of required edge deletions (see Section 2.1).

2.1 Computational hardness

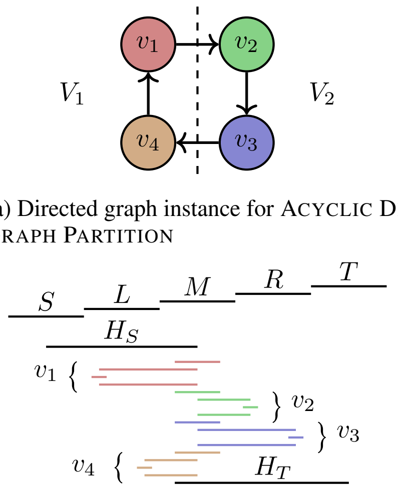
Next, we show that OPINION INTERVAL REPRESENTATION is NP-hard. We show this by using a reduction from the NP-complete problem ACYCLIC DIGRAPH PARTITION [19], where we are given a directed graph $H = (V, E)$ and have to decide whether one can partition V into two sets V_1 and V_2, such that both $H[V_1]$ and $H[V_2]$ are directed acyclic graphs.

Our hardness result is stated below. We say that an interval representation is *conflict-free* if it achieves agreement for all edges, i.e., if Equation (1) equals the number of edges in the graph.

Theorem 2.2. *There exists a polynomial-time algorithm that, given an instance $H = (V, E)$ of* ACYCLIC DIGRAPH PARTITION*, outputs an instance $G = (V', E^+ \cup E^-)$ of* OPINION INTERVAL

REPRESENTATION *with the following properties: (1) H is a YES-instance if and only if a conflict-free interval representation of G exists. (2) If H is a NO-instance, then only a single edge deletion is required to obtain a conflict-free interval representation of G. (3) $|V'| = \mathcal{O}(|V|)$, $|E^+ \cup E^-| < \binom{|V'|}{2}$, and $G^+ = (V', E^+)$ is a cycle.*

The theorem has several implications for OPINION INTERVAL REPRESENTATION in incomplete graphs: (1) The disagreement version is hard to approximate within any factor. (2) It is not FPT when parameterized by the number of required edge deletions (unless $\mathsf{P} = \mathsf{NP}$), separating it from the problem in complete graphs. (3) The result holds even when restricted to graphs $G = (V', E^+ \cup E^-)$ where $G^+ = (V', E^+)$ is a chordless cycle. This is intriguing because many algorithmic results on interval graphs rely on detecting forbidden induced subgraphs like chordless cycles of four or more vertices [9, 48, 32]. Our hardness result implies that detecting these forbidden structures is NP-hard for incomplete signed graphs.



(a) Directed graph instance for ACYCLIC DIGRAPH PARTITION

(b) Interval representation

Figure 3: Reduction from ACYCLIC DIGRAPH PARTITION

In our reduction, we construct a new graph G from an ACYCLIC DIGRAPH PARTITION instance H, and show that G can be represented conflict-free if and only if H is a YES-instance. In G, we introduce two auxiliary vertices L and R and we show that all vertices whose intervals overlap with the interval of L (R) must be in partition V_1 (V_2) in the optimal solution of ACYCLIC DIGRAPH PARTITION. Thus, the overlap structure of the intervals encodes a partition of the vertices of H. Crucially, we use negative edges to enforce a topological ordering over these partitions and the induced subgraphs $H[V_1]$ and $H[V_2]$, and we introduce further auxiliary vertices to ensure that G forms a cycle. We refer to Figure 3 for an illustration.

2.2 A PTAS for fixed k in complete graphs

From an algorithmic perspective, we provide a PTAS when G is a complete graph and when each vertex must be assigned to one of k intervals, where $k = O(1)$. Formally, we study a version of OPINION INTERVAL REPRESENTATION in which we must find k intervals $I_1, \ldots, I_k \subset \mathbb{R}$ and each vertex $v \in V$ must be assigned to one of these intervals. In practice, the small number of intervals makes the results highly interpretable. Additionally, it applies to scenarios such as analyzing political votes, where we would like to have one interval representing each party, and the number of parties is small. Our result for this restricted version of the problem is as follows.

Theorem 2.3. *Let G be a complete signed graph and let $\varepsilon > 0$, $\delta > 0$ and $k \in \mathbb{N}$ be parameters. There exists an algorithm that, with probability at least $1 - \delta$, returns a $(1+\varepsilon)$-approximate solution for* OPINION INTERVAL REPRESENTATION *when the algorithm can only use k different intervals and it runs in time $2^{O(k^2 \log(k/(\varepsilon\delta))/\varepsilon^3)} \cdot n$.*

An overview to obtain this result is as follows. Since k is fixed, we can enumerate all possible choices of k intervals with respect to their overlap structure. Now, given a fixed set of k intervals, the main observation is that this corresponds to a generalized instance of CORRELATION CLUSTERING where we are given k fixed clusters that might overlap. Specifically, when two clusters V_i and V_j overlap, we want their vertices to be connected by positive edges (rather than negative edges in the classic version of CORRELATION CLUSTERING). Then, we show that we can generalize a PTAS from Giotis and Guruswami [24] as described below.

We solve the generalized version of CORRELATION CLUSTERING by partitioning V into $m = O(1/\varepsilon)$ equally-sized subsets $V_1, \ldots, V_m$. Then, for each $i = 1, \ldots, m$, we proceed as follows. We sample a set of vertices $S_i \subseteq V \setminus V_i$ of size $\tilde{O}\left(1/\varepsilon^2\right)$. Now, we enumerate all possible assignments of S_i into $(S_{i,1}, \ldots, S_{i,k})$, where $S_{i,\ell} \subseteq S_i$ are the vertices assigned to interval I_ℓ, and for each such

Algorithm 1: Greedy Agreement Interval Assignment (`GAIA`)

```
Input: Signed graph $G = (V, E^+ \cup E^-)$, intervals $I_1, \dots, I_k$
Output: Interval assignment $(C_1, \dots, C_k)$ where $C_\ell$ are the vertices assigned to interval $I_\ell$
1 Compute an initial assignment of the vertices to the intervals;
2 for each epoch do
3     Randomly partition $V$ into $m$ sets $V_1, \dots, V_m$ of size $\frac{n}{m}$ each;
4     for $i = 1, \dots, m$ do
5         $C_\ell \leftarrow C_\ell \setminus V_i$ for all $\ell = 1, \dots, k$;      // Unassign all vertices in $V_i$
6         for $v \in V_i$ in order of maximum agreement do
7             $\ell \leftarrow \operatorname{argmax}_{\ell=1\dots k} \mathsf{agree}(v, \ell, (C_1, \dots, C_k))$;
8             $C_\ell \leftarrow C_\ell \cup \{v\}$;      // Assign $v$ to $I_\ell$
9 return $(C_1, \dots, C_k)$;
```

assignment, we greedily assign the vertices $v \in V_i$ to the interval that maximizes the agreement of v's edges to the clustering of S_i given by $(S_{i,1}, \dots, S_{i,k})$. This process gives a clustering of V_i and we show how the clusterings of the V_i can be merged to obtain a global clustering of V.

Our analysis is similar to that of [24] and shows that the sets S_i are small enough such that enumerating all assignments is not too expensive, and simultaneously large enough that for most vertices they give us a good estimate for the agreement of their edges w.r.t. a fixed clustering. This is the key to arguing that the greedy assignment will yield a good result when we consider the correct clustering of the S_i. In contrast to [24] we have to take into account the overlap of intervals when computing the estimates. As for [24], the approach does not extend to incomplete graphs or large k, since then the sets S_i become too large and enumeration would not be possible anymore.

3 Heuristic Algorithms

Next, we present our heuristic *Greedy Agreement Interval Assignment* (`GAIA`) for OPINION INTERVAL REPRESENTATION, which is given intervals $I_1, \dots, I_k$ as input to which all vertices must be assigned.

We use the following notation. For an interval I_ℓ we let $\mathsf{overlap}(\ell) = \{\ell' \colon I_\ell \cap I_{\ell'} \neq \emptyset\}$ denote the set of intervals $I_{\ell'}$ that overlap with I_ℓ. Furthermore, we will consider disjoint vertex clusters $C_1, \dots, C_k \subseteq V$ that correspond to an assignment of the vertices to the intervals, i.e., C_ℓ contains all vertices assigned to interval I_ℓ. Now, for a vertex u and $C_1, \dots, C_k$ as before, we write

$$\mathsf{agree}(u, \ell, (C_1, \dots, C_k)) = \sum_{\ell' \in \mathsf{overlap}(\ell)} \left|N^+(u) \cap C_{\ell'}\right| + \sum_{\ell' \notin \mathsf{overlap}(\ell)} \left|N^-(u) \cap C_{\ell'}\right|$$

for the number of agreeing edges when assigning vertex u to interval I_ℓ for the clustering $C_1, \dots, C_k$.

Now, we describe `GAIA` and state its pseudocode in Algorithm 1. `GAIA` is based on *iterative refinement*: After computing an initial greedy assignment of all vertices, the solution is improved by reassigning vertices in multiple epochs. This reassignment procedure is carried out in batches to avoid local minima. In each epoch, the vertex set is partitioned into random batches $V_1, \dots, V_m$, and the algorithm iterates over these batches one at a time. When processing a batch V_i, all vertices in the batch are first unassigned and then reassigned using the greedy procedure described below. This can be viewed as a practical version of PTAS from Theorem 2.3, where, instead of brute-forcing solutions on out-of-batch vertices, the algorithm leverages the previously constructed greedy solution.

The core of `GAIA` is the *greedy assignment* of vertices in V_i to intervals in Line 5–8 in Algorithm 1. Here, each vertex $v \in V_i$ is assigned to the interval I_ℓ (and its corresponding cluster C_ℓ) that maximizes $\mathsf{agree}(u, \ell, (C_1, \dots, C_k))$ (breaking ties at random). Crucially, we assign the vertices with the highest agreement values first, as these vertices are easier to assign and their assignment provides more information when assigning later vertices.

We also provide a version of `GAIA` called *Variable ENergy Uphill Search* (`VENUS`), which additionally uses *simulated annealing* [31] to further increase the variability of its solutions. In `VENUS`, vertices are not necessarily assigned to the interval that maximizes $\mathsf{agree}(u, \ell, (C_1, \dots, C_k))$, but instead, each vertex is assigned to an interval selected probabilistically according to a temperature-scaled softmax distribution over agreement values. To that end, Line 7 of Algorithm 1 is replaced with $\ell \sim \text{softmax}_{\ell=1\ldots k}\frac{\mathsf{agree}(v,\ell,(C_1,\ldots,C_k))}{t}$. Here, t is a temperature parameter and controls the level of randomness during the assignment. A temperature t close to 0 corresponds to a more greedy approach, while higher temperatures lead to increasingly uniform random assignments. The annealing schedule follows exponential decay: the temperature is initialized at t_0 and multiplied by a decay factor $\alpha \in (0, 1)$ after every τ epochs. This gradually reduces randomness and encourages convergence.

4 Experiments

Next, we experimentally evaluate our algorithms. Our code is available on an anonymous GitHub repository.[1] We aim to answer the following research questions:

(RQ1) Does Opinion Interval Representation yield a substantial increase in expressiveness compared to Correlation Clustering?

(RQ2) How computationally efficient and scalable are our proposed algorithms?

(RQ3) What is the trade-off between solution quality and the number of intervals?

(RQ4) Are the solutions produced by our method interpretable?

(RQ5) Are our algorithms able to recover ground-truth interval structures?

We evaluate our algorithms on real-world datasets from SNAP [38] and KONECT [34]. We further provide a novel dataset based on voting data from the German Bundestag (parliament) between the years of 2012 and 2025 and make it available in our repository. In this dataset, each Bundestag member corresponds to a vertex in the graph, and two members are connected by a positive edge if they vote the same way in at least 75% of the sessions they both attended. Conversely, they are connected by a negative edge if their votes align in 25% of sessions or less.

In our experiments, we evaluate our base algorithm, GAIA, as well as the VENUS variant that uses simulated annealing. For VENUS, we use an initial temperature of 100 and a decay factor of $2/3$, applied every 5 epochs. Both are run with 10 batches for vertex reassignment. For the interval structure that our algorithms receive as input, unless stated otherwise, we use a chain-like structure of 8 intervals, where each interval overlaps with the next, e.g., $[0, 1], [1, 2], \ldots, [7, 8]$, and we call this interval structure an 8-Chain. This structure was chosen to find a trade-off between increased expressivity and intuitive interpretation (see also Section 4 below). Where applicable, experiments were repeated 50 times on different random seeds, and standard deviations are reported.

Numerous approaches have been proposed for solving Correlation Clustering in social networks analysis [41, 8, 27, 26] and in computer vision [30, 49, 35, 1]. To provide a representative performance overview, we selected four state-of-the-art algorithms for comparison:

- GAEC [30]: A method that incrementally merges clusters to minimize disagreement.
- GAECKLj [30]: An extension of GAEC that additionally applies local search postprocessing.
- SCMLEvo [26]: An algorithm combining multilevel local search with evolutionary techniques.
- RAMA [1]: An algorithm using polyhedral relaxation and message passing to guide cluster merging.

For each of these algorithms, we run the authors' publicly available implementations. In contrast to our algorithms, which only use 8 intervals, the baselines may use an unrestricted number of clusters.

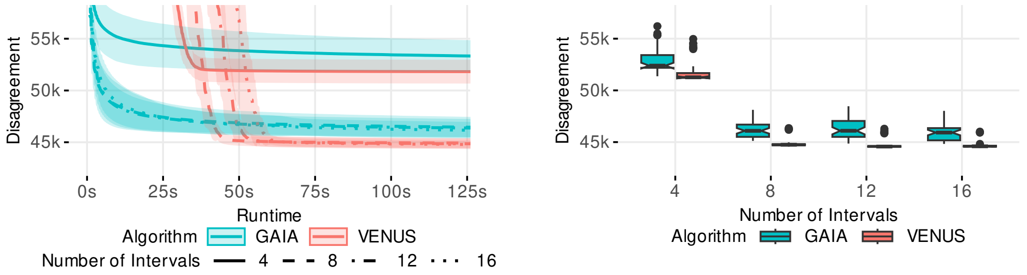
Throughout our experiments, we report the *disagreement*, i.e., the fraction of violated edges in solutions found across all real-world datasets (rather than the number of agreeing edges as in Equation (1)), as this makes the algorithms' performance easier to compare.

Expressivity analysis (RQ1). To compare the expressivity of Opinion Interval Representation with Correlation Clustering, we run our algorithms and the baselines on our real-world datasets. The results are shown in Table 1, and show that our algorithms consistently yield an improvement in disagreement of 20% to 90% over the best Correlation Clustering solution. Across all datasets, our results have 38% less disagreement on average, even though our methods only use 8 intervals, whereas the Correlation Clustering baselines use an unrestricted number. Additionally, we find that VENUS tends to outperform GAIA slightly, particularly for larger graphs.

[1] https://anonymous.4open.science/r/opinion_projection-4072

Table 1: Overview of the best solutions found by the algorithms. Reported is the fraction of edges violated (lower is better). Our algorithms use the `8-Chain` interval structure while the CORRELATION CLUSTERING algorithms use an unrestricted number of clusters.

				Our algorithms		CORRELATION CLUSTERING baselines				
Dataset	$\|V\|$	$\|E\|$	$\frac{\|E^+\|}{\|E\|}$	`GAIA`	`VENUS`	`GAEC`	`GAECKLj`	`SCMLEvo`	`RAMA`	Improvement
BitcoinOTC	5 881	21 434	0.85	**3.32**	3.55	5.58	5.57	5.57	5.64	40.39%
Chess	7 301	32 650	0.58	19.82	**19.63**	28.64	28.10	27.33	39.98	28.17%
WikiElec	7 115	100 355	0.78	**11.24**	11.26	14.13	14.13	14.13	14.45	20.45%
Bundestag	1 480	397 497	0.81	**0.25**	**0.25**	3.06	2.95	2.95	3.72	91.53%
Slashdot	82 140	498 532	0.76	9.05	**8.94**	13.75	13.66	13.52	17.17	33.88%
Epinions	131 580	708 507	0.83	4.47	**4.42**	6.83	6.68	6.67	6.86	33.73%
WikiSigned	138 587	712 337	0.88	4.94	**4.85**	6.17	6.17	6.17	6.96	21.39%
WikiConflict	116 836	2 014 053	0.38	3.44	**3.43**	5.87	5.82	5.82	6.02	41.06%



(a) Objective value over time.

(b) Final objective for different interval structures.

Figure 4: Performance of our algorithms with different configurations on the Slashdot dataset.

Computational efficiency (RQ2). Next, we assess the runtime efficiency of our algorithms by tracking the progression of the objective value over time. Representative results for Slashdot are presented in Figure 4a. `GAIA` makes the most progress in the first 15 seconds, followed by slower, incremental gains, and often stops early. `VENUS` exhibits a similar pattern, though slightly delayed, likely due to its initially high temperature which slows early convergence. However, this high initial temperature appeared to be necessary to achieve improvement over `GAIA`'s results. In most instances, both heuristics stopped after 50 epochs without improvement in the first five minutes of runtime. The algorithms' running time until convergence scales approximately linearly in the number of edges and on all datasets our methods finish in less than 30 minutes; we also elaborate on this in the appendix.

Number of intervals (RQ3). To investigate the relationship between the number of intervals and the solution quality, we ran our algorithms with 4, 8, 12, and 16 intervals, resp. As for the `8-Chain`, in each interval structure, every interval overlaps with its successor and predecessor. In Figure 4a, we illustrate the convergence behavior, and Figure 4b presents the solution quality after convergence, both on the Slashdot dataset. Our results show that using only 4 intervals leads to poor solution quality compared to the higher numbers, suggesting that such a limited structure may not adequately capture the complexity of the graph. While the solution quality improves with more intervals, 8 intervals seem sufficiently expressive for this graph, with only marginal improvements beyond that. This behavior is typical for other problem instances as well. Again, we see that `VENUS` tends to perform slightly better, and most notably, its results have much less variance compared to `GAIA`.

Interpretability (RQ4). To study the interpretability of our solutions, we perform a case study on the Bundestag dataset. We present a representative solution found by `VENUS` in Figure 1. As the dataset models co-voting behavior of politicians, we expect our interval representation to resemble the German political spectrum, and, indeed, this is the case. Our result assigns most members from the same party to the same or neighboring intervals. For each party, except the FDP, we can identify one interval consisting mainly of members of that party. We note that the slight splitting up of parties is natural due to government coalitions they formed throughout the years. Also the behavior of the FPD can be traced back to different coalition governments they were part of (they formed governments with the conservative CDU/CSU, as well as with the left/center GRÜNE and SPD). We consider the ability of our algorithms to extract such highly overlapping structure as a substantial improvement

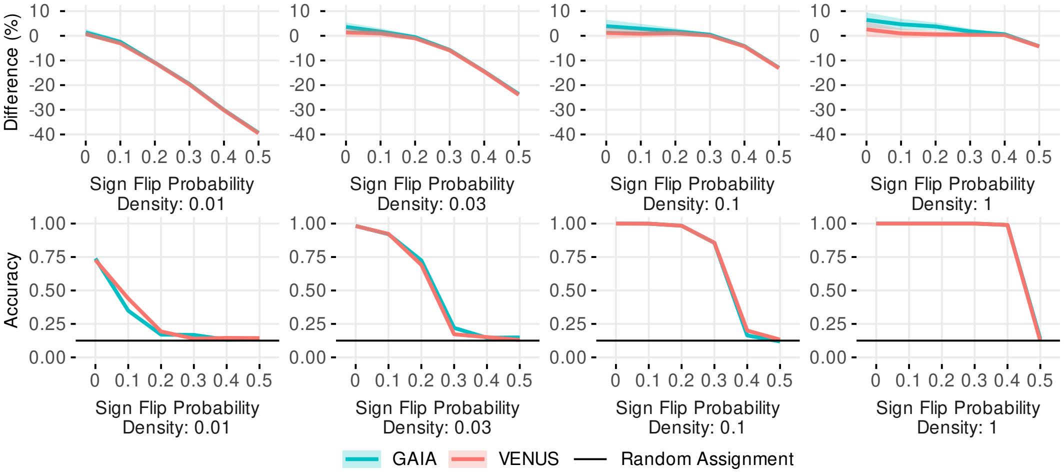


Figure 5: Results on synthetic data. For different edge density levels $|E| / \binom{|V|}{2}$, we report the normalized difference $\frac{\mathsf{agree}(G,\text{ground truth}) - \mathsf{agree}(G,\mathsf{ALG})}{|E|}$ averaged over 50 runs, and standard deviations. Negative numbers indicate an improvement over the ground truth. Further, we report the accuracies of *the solution with the lowest disagreement.* `GAIA` and `VENUS` reconstruct the ground truth under considerable levels of noise in dense graphs, and find alternative, better solutions in sparse graphs.

over Correlation Clustering, and this is also emphasized by the objective function values reported in Table 1, where our methods have 91% fewer disagreeing edges on this dataset.

Reconstruction of ground-truth data (RQ5). Next, we evaluate our algorithms on synthetic data. We fix the `8-Chain` and generate a graph with $n = 800$ vertices as follows. We assign $\frac{n}{8}$ vertices to each interval, and we introduce edges with signs corresponding to the interval structure for $d\binom{n}{2}$ random pairs of vertices, where $d \in [0, 1]$ is the desired density of the graph. Each edge obtains a correct edge sign based on the interval structure with probability $1 - p$ and we flip the sign with probability p. In our experiments, we measure the relative change of the objective function achieved by our algorithms compared to the ground-truth assignment in percent, $\frac{\mathsf{agree}(G,\text{ground truth}) - \mathsf{agree}(G,\mathsf{ALG})}{|E|} \cdot 100$, and we also report the accuracy with which vertices are assigned to their corresponding interval.

Figure 5 shows the result of our experiments. Without sign noise, the solutions are always within 6.5% of the ground truth for `VENUS`, and within 12.5% of the ground truth for `GAIA`. We also obtain a high accuracy in reconstructing the ground-truth assignment. With increasing sign noise, the true solution becomes increasingly suboptimal to the point where both `GAIA` and `VENUS` find alternative solutions, with *better* objective values than the ground truth (this is the case when we have negative y-axis values in the plot). This increased objective value, however, comes at the cost of less accuracy in the vertex assignment. The point at which alternative solutions become viable depends heavily on the density of the graph, with denser graphs being more resilient to this phenomenon.

5 Conclusion

We introduced the Opinion Interval Representation problem and showed that it is more expressive than Correlation Clustering, both theoretically and in experiments. We gave strong hardness results for incomplete graphs, as well as a PTAS for complete graphs and fixed k. We also provided efficient heuristics, which yield significantly better objective function values than Correlation Clustering baselines, and we showed that their results are highly interpretable.

There are several interesting directions for future work. We conjecture that in the agreement version of the problem (for k not fixed), the optimal solution can always satisfy a $\frac{3}{4}$-fraction of the edges. This claim is supported by ILP-solutions that we computed on small instances, and it is tight, for instance, when taking two cliques with negative edges and connecting each pair of their vertices with a positive edge. It is also interesting to study whether a PTAS exists in this setting. Additionally, our hardness results do not allow us to rule out that for fixed k and incomplete graphs a PTAS exists.

References

[1] A. Abbas and Paul Swoboda. Rama: A rapid multicut algorithm on gpu. *CVPR*, pages 8183–8192, 2021.

[2] Samin Aref and Zachary P. Neal. Identifying hidden coalitions in the us house of representatives by optimally partitioning signed networks based on generalized balance. *Scientific Reports*, 11, 2021.

[3] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3): 89–113, 2004.

[4] Pablo Barberá. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political analysis*, 23(1):76–91, 2015.

[5] Thorsten Beier, Thorben Kröger, Jörg H. Kappes, U. Köthe, and Fred A. Hamprecht. Cut, glue, & cut: A fast, approximate solver for multicut partitioning. *CVPR*, pages 73–80, 2014.

[6] Francesco Bonchi, A. Gionis, and Antti Ukkonen. Overlapping correlation clustering. *Knowledge and Information Systems*, 35:1–32, 2011.

[7] Francesco Bonchi, David García-Soriano, and Francesco Gullo. *Correlation Clustering*. Synthesis Lectures on Data Mining and Knowledge Discovery. Springer, 2022.

[8] Michael J. Brusco and Patrick Doreian. Partitioning signed networks using relocation heuristics, tabu search, and variable neighborhood search. *Social Networks*, 56:70–80, 2019.

[9] Yixin Cao. Linear recognition of almost interval graphs. In *SODA*, pages 1096–1115, 2016.

[10] Yixin Cao. Unit interval editing is fixed-parameter tractable. *Inf. Comput.*, 253:109–126, 2017.

[11] Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(3):21:1–21:35, 2015.

[12] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *FOCS*, pages 524–533, 2003.

[13] Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. Correlation clustering in constant many parallel rounds. In *ICML*, volume 139, pages 2069–2078, 2021.

[14] Vincent Cohen-Addad, Silvio Lattanzi, Andreas Maggiori, and Nikos Parotsidis. Online and consistent correlation clustering. In *ICML*, volume 162, pages 4157–4179, 2022.

[15] Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. Correlation Clustering with Sherali-Adams. In *FOCS*, pages 651–661, 2022.

[16] Vincent Cohen-Addad, Euiwoong Lee, Shi Li, and Alantha Newman. Handling correlated rounding error via preclustering: A 1.73-approximation for correlation clustering. In *FOCS*, pages 1082–1104, 2023.

[17] Vincent Cohen-Addad, Silvio Lattanzi, Andreas Maggiori, and Nikos Parotsidis. Dynamic correlation clustering in sublinear update time. In *ICML*, 2024.

[18] Vincent Cohen-Addad, David Rasmussen Lolck, Marcin Pilipczuk, Mikkel Thorup, Shuyi Yan, and Hanwen Zhang. Combinatorial correlation clustering. In *STOC*, pages 1617–1628. ACM, 2024.

[19] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Sitting closer to friends than enemies, revisited. *Theory Comput. Syst.*, 56:394–405, 2015.

[20] Guillaume Deffuant, David Neau, Frederic Amblard, and Gérard Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98, 2000.

[21] Morris H DeGroot. Reaching a consensus. *J Am Stat Assoc*, 69(345):118–121, 1974.

[22] Noah E Friedkin and Eugene C Johnsen. Social influence and opinions. *Journal of Mathematical Sociology*, 15(3-4):193–206, 1990.

[23] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. ISBN 0-7167-1044-7.

[24] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory Comput.*, 2(13):249–266, 2006.

[25] Frank Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2 (2):143–146, 1953.

[26] Felix Hausberger, Marcelo Fonseca Faraj, and Christian Schulz. Scalable multilevel and memetic signed graph clustering. In *ALENEX*, pages 81–94, 2025.

[27] Jia-Lin Hua, Jian Yu, and Miin-Shen Yang. Fast clustering for signed graphs based on random walk gap. *Social Networks*, 60:113–128, 2020.

[28] Haim Kaplan, Ron Shamir, and Robert Endre Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28: 1906–1922, 1999.

[29] Anne-Marie Kermarrec and Christopher Thraves. Can everybody sit closer to their friends than their enemies? In *MFCS*, pages 388–399, 2011.

[30] Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjoern Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. *ICCV*, pages 1751–1759, 2015.

[31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[32] Dieter Kratsch, Ross M. McConnell, Kurt Mehlhorn, and Jeremy P. Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. *SIAM J. Comput.*, 36(2): 326–353, 2006.

[33] Ulrich Krause et al. A discrete nonlinear and non-autonomous model of consensus formation. *Communications in difference equations*, 2000:227–236, 2000.

[34] Jérôme Kunegis. Konect: the koblenz network collection. *WWW*, 2013.

[35] Jan-Hendrik Lange, Andreas Karrenbauer, and Bjoern Andres. Partial optimality and fast lower bounds for weighted correlation clustering. In *ICML*, volume 80, pages 2892–2901, 2018.

[36] Jacopo Lenti, Corrado Monti, and Gianmarco De Francisci Morales. Likelihood-based methods improve parameter estimation in opinion dynamics models. In *WSDM*, pages 350–359, 2024.

[37] Jacopo Lenti, Fabrizio Silvestri, and Gianmarco De Francisci Morales. Variational inference of parameters in opinion dynamics models. *CoRR*, abs/2403.05358, 2024.

[38] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[39] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.

[40] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Signed networks in social media. In *CHI*, pages 1361–1370, 2010.

[41] Evgeny Levinkov, Alexander Kirillov, and Bjoern Andres. A comparative study of local search algorithms for correlation clustering. In *German Conference on Pattern Recognition*, 2017.

[42] Mário Levorato, Rosa Figueiredo, Yuri Frota, and Lúcia M. A. Drummond. Evaluating balancing on social networks through the efficient solution of correlation clustering problems. *EURO Journal on Computational Optimization*, 5:467–498, 2017.

[43] Eduardo G. Pardo, Mauricio Soto, and Christopher Thraves. Embedding signed graphs in the line - heuristics to solve minsa problem. *J. Comb. Optim.*, 29:451–471, 2015.

[44] Eduardo G. Pardo, Antonio García-Sánchez, Marc Sevaux, and Abraham Duarte. Basic variable neighborhood search for the minimum sitting arrangement problem. *J. Heuristics*, 26:249–268, 2020.

[45] Fred S. Roberts. Indifference graphs. *Proof Techniques in Graph Theory*, pages 139–146, 1969.

[46] Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *SODA*, 2004.

[47] Paul Swoboda and Bjoern Andres. A message passing algorithm for the minimum cost multicut problem. *CVPR*, pages 4990–4999, 2016.

[48] Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM J. Comput.*, 38(5):2007–2020, 2009.

[49] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, U. Köthe, and Fred A. Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In *ECCV*, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claimed theoretical results are formalized in Section 2 with additional proofs in Appendices A and B. The heuristic algorithms are described in Section 3, and the claimed empirical results are supported in Section 4.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our theorems clearly state the assumptions that are necessary for them to hold. Regarding the performance of our practical algorithms, we have evaluated them to the best of our knowledge and also provide further information on their scalability in the appendix.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For each theoretical result (Theorems 2.2 and 2.3) we clearly state assumptions and provide proof sketches in Section 2. Formal proofs are provided in Appendices A and B and cross-referenced to Theorems 2.2 and 2.3, respectively.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The heuristic algorithms are described in Section 3, the dataset generation process and the parameters used to run the experiments are described in Section 4 and Appendix C. Additionally, our code is available in an anonymized repository.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code, including the experiment setup, is available as an anonymized repository, referenced in Footnote 1 in Section 4. All datasets used are either publicly available or included in the linked repository.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The parameters used to run all experiments are described in Section 4 and Appendix C. Additionally, the code to reproduce our experiments is available in an anonymized repository.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: All experimental results in Section 4 and Appendix C include averages, standard deviations, and confidence intervals where applicable. In the box plots, the notches extend $1.58 \cdot IQR/\sqrt{n}$, giving roughly a 95% confidence interval for comparing medians.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources used in the experiments are described in Appendix C.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The authors have ensured that the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We believe that our work has no broader societal impact, given that our focus is on foundational research. While we offer a new perspective on modeling opinion diversity using limited interaction data, similar insights are already frequently obtained in the real world by existing methods using richer user-level data.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all original papers where code or datasets are used in our work. In particular, for the scraped Bundestag dataset, we provide copyright- and legal notice in the anonymized repository.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The generation process for the newly introduced dataset is described in Section 4. The dataset and the code used for its generation are available in an anonymized repository.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: No crowdsourcing or experiments with human subjects were used in the paper.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No crowdsourcing or experiments with human subjects were used in the paper.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not involved in the core method development.

A Hardness Result

In this section, we prove our hardness result from Theorem 2.2. In the proof, we use the notation $I = [\ell(I), r(I)]$, where $\ell(I)$ denotes the infimum and $r(I)$ denotes the supremum of the interval.

A.1 Construction

We describe a reduction from an instance $H = (V, E)$ of ACYCLIC DIGRAPH PARTITION to an instance $G = (V', E^+ \cup E^-)$ of OPINION INTERVAL REPRESENTATION, where (V', E^+) forms a cycle.

First, the set of vertices V' consists of:

1. Seven constant vertices $V'_c = \{S, L, M, R, T, H_S, H_T\}$. These vertices will be used to constrain the structure of the solution. With sets of negative edges, we will force any conflict-free representation to assign each vertex in V to a sub-interval of either L or R. The names of the vertices stand for start, left, middle, right, target, help-start, and help-target, respectively.
2. Four vertices for each vertex $v \in V$: $V'_v = \{M_v, A_v, X_v, B_v\}$, where X_v corresponds to the original vertex v in H and the other vertices are used to structure the instance.

Next, we define the set of positive edges E^+ to form a cycle over V'. For this, we use an arbitrary ordering of $V = \{v_1, \ldots, v_n\}$, and construct E^+ as the union of the following sets:

1. $E^+_c = \{\{S, L\}, \{L, M\}, \{M, R\}, \{R, T\}, \{S, H_S\}, \{H_T, T\}\}$
2. for all $v \in V$: $E^+_v = \{\{M_v, A_v\}, \{A_v, X_v\}, \{X_v, B_v\}\}$
3. $E^+_V = \{\{H_S, M_{v_1}\}\} \cup \bigcup_{i \in [1,n-1]} \{\{B_{v_i}, M_{v_{i+1}}\}\} \cup \{\{B_{v_n}, H_T\}\}$

Finally, we construct E^- as the union of the following sets:

1. $E^-_S = \{\{S, v'\} : v' \in V' \setminus \{L, H_S\}\}$, $E^-_T = \{\{T, v'\} : v' \in V' \setminus \{R, H_T\}\}$
   We connect negatively S and T to each vertex in the graph besides their positive neighbors. This forces I_S and I_T to be the outermost intervals in any conflict-free interval representation, as otherwise the interval of some negatively connected vertex intersects either of them. See Lemma A.1. To break symmetry, we assume without loss of generality that $r(I_S) < \ell(I_T)$.
2. $E^-_c = \{\{L, R\}\}$
   This edge ensures that I_L and I_R are disjoint, and in any conflict-free representation, $r(I_L) < \ell(I_R)$, due to their respective positive edges to S and T.
3. $E^-_M = \bigcup_{v \in V} \{\{M_v, L\}, \{M_v, R\}\}$
   These edges ensure that for all vertices $v \in V$ the interval I_{M_v} lies in I_M. See Lemma A.3.
4. $E^-_* = \{\{X_v, M\} : v \in V\}$
   These edges ensure that for all vertices $v \in V$, the interval I_{X_v} either lies in I_L or in I_R. See Lemma A.4.
5. $E^-_V = \{\{X_u, X_v\} : u, v \in V, u \neq v\}$
   These edges ensure that for all vertices $u, v \in V$ with $u \neq v$ the intervals I_{X_u} and I_{X_v} are disjoint.
6. $E^-_E = \bigcup_{(u,v) \in E} \{\{X_v, A_u\}, \{X_v, B_u\}\}$
   These edges enforce topological orderings of the vertices. See Lemma A.5.

This concludes the construction. It is clear that V', E^+, and E^- have the sizes claimed in the theorem.

A.2 Structural lemmas

To prove the correctness of the reduction, we will make use of a few smaller results that describe the structure of any conflict-free interval representation of G. First, notice that $\{S, T\} \in E^-$, so for any conflict-free interval representation, it must hold that $I_S \cap I_T = \emptyset$. For the rest of this analysis, assume without loss of generality $r(I_S) < \ell(I_T)$.

Lemma A.1. *For any conflict-free interval representation of G it must hold that for all $u \in V' \setminus \{S\}$: $r(I_S) < r(I_u)$ and for all $u \in V' \setminus \{T\} : \ell(I_u) < \ell(I_T)$.*

Proof. Towards a contradiction assume there exists a vertex $u \in V' \setminus \{S\}$ such that $r(I_u) \leq r(I_S)$. As (V', E^+) is a cycle, there exists a path from u to T that does not include S. The union I_p of the intervals corresponding to the vertices in this path must form an interval itself. As $r(I_u) \leq r(I_S)$, but $\ell(I_T) > r(I_S)$, $I_S \cap I_p \neq \emptyset$, and consequently there exists some vertex $x \in V' \setminus \{S\}, x \neq u$ along this path such that $I_S \cap I_x \neq \emptyset$. Note that by the construction of this path, x cannot be a positive neighbor of S, as we explicitly choose one of the two paths from u to T that does *not* include S. Then, $\{S, x\} \in E^-$, leading to a contradiction. Finally, for all $u \in V' \setminus \{T\} : \ell(I_u) < \ell(I_T)$ holds by a symmetric argument.

□

Lemma A.2. *For any conflict-free interval representation of G it must hold that (i) for all $u \in V' \setminus \{S, L, H_S\} : r(I_S) < \ell(I_u)$ and (ii) for all $u \in V' \setminus \{T, R, H_T\} : r(I_u) < \ell(I_T)$.*

Proof. Assume towards a contradiction that there exists a vertex $u \in V' \setminus \{S, L, H_S\}$ such that $\ell(I_u) \leq r(I_S)$. From Lemma A.1, we know $r(I_u) > r(I_S)$, so it follows that $I_u \cap I_S \neq \emptyset$. However, since $\{S, u\} \in E^-$ this leads to a contradiction with the assumption of a conflict-free interval representation. The proof for (ii) follows symmetrically. □

Lemma A.3. *For any conflict-free interval representation of G, it must hold that for all $v \in V$: $I_{M_v} \subset I_M$.*

Proof. By construction, the open interval $(r(I_S), \ell(I_T)) \subset I_L \cup I_M \cup I_R$. From Lemma A.2, we know that $I_{M_v} \subset I_L \cup I_M \cup I_R$. Finally, as $\{M_v, L\}, \{M_v, R\} \in E^-$, the claim holds. □

Lemma A.4. *For any conflict-free interval representation of G, it must hold that for all $v \in V$ either $I_{X_v} \subset I_L$ or $I_{X_v} \subset I_R$ but not both.*

Proof. By construction, the open interval $(r(I_S), \ell(I_T)) \subset I_L \cup I_M \cup I_R$. From Lemma A.2, we know that $I_{X_v} \subset I_L \cup I_M \cup I_R$. Furthermore, we know that $I_L \cap I_R = \emptyset$. Finally, as $\{X_v, M\} \in E^-$ the claim holds. □

Building on this statement, we can further characterize the relative location of the intervals I_{X_v} inside I_L and I_R.

Lemma A.5. *For any conflict-free interval representation of G it must hold that for all edges $(u, v) \in E$ if $I_{X_u} \subset I_R$ and $I_{X_v} \subset I_R$, then $r(I_{X_u}) < \ell(I_{X_v})$, and, symmetrically, if $I_{X_u} \subset I_L$ and $I_{X_v} \subset I_L$, then $r(I_{X_v}) < \ell(I_{X_u})$*

Proof. As we assumed that $r(I_S) < \ell(I_T)$, it follows that $\ell(I_M) < \ell(I_R) \leq r(I_M) < r(I_R)$. Towards a contradiction, assume there exists an edge $(u, v) \in E$ such that $I_{X_u} \subset I_R$ and $I_{X_v} \subset I_R$, but $\ell(I_{X_v}) \leq r(I_{X_u})$. As in any conflict-free interval representation the intervals I_{X_u} and I_{X_v} are disjoint, this implies that $r(I_{X_v}) < \ell(I_{X_u})$. Now, consider the intervals I_{M_u} and I_{A_u}. From Lemma A.3, we know that $I_{M_u} \subset I_M$, hence $r(I_{M_u}) < \ell(I_{X_v}) < r(I_{X_v}) < \ell(I_{X_u})$. By construction, I_{A_u} must overlap with I_{M_u} and I_{X_u}, hence $\ell(I_{A_u}) \leq r(I_{M_u})$ and $r(I_{A_u}) \geq \ell(I_{X_u})$. However, this implies that $I_{A_u} \cap I_{X_v} \neq \emptyset$, leading to a violation of the $\{X_v, A_u\}$ constraint introduced in E^-_E. By symmetry, this also proves the case where $I_{X_u} \subset I_L$ and $I_{X_v} \subset I_L$. □

A.3 Proof of Theorem 2.2

Equipped with Lemmas A.1 to A.5 we can now prove Theorem 2.2.

We first show that if $H = (V, E)$ is a YES-instance of ACYCLIC DIGRAPH PARTITION, then the constructed signed graph instance G has a conflict-free interval-representation in OPINION INTERVAL REPRESENTATION. Assume $H[V_1]$ and $H[V_2]$ are the two acyclic induced subgraphs of H corresponding to the partition and let $k = |V_1|$. Further, let $[v_{(1,1)}, \ldots, v_{(1,k)}]$ and $[v_{(2,1)}, \ldots, v_{(2,n-k)}]$ be topological orderings of V_1 and V_2, respectively. Now, we define intervals for V_c' as follows and depicted in Figure 3:

$I_S \coloneqq [0, 0.2]$, $I_L \coloneqq [0.2, 0.4]$, $I_M \coloneqq [0.4, 0.6]$, $I_R \coloneqq [0.6, 0.8]$, $I_T \coloneqq [0.8, 1]$, $I_{H_S} \coloneqq [0.1, 0.5]$, $I_{H_T} \coloneqq [0.5, 0.9]$.

This satisfies the constraints set by E_c^-. Then, for each $v \in V$, we assign $I_{M_v} = [0.45, 0.55]$. This satisfies all constraints imposed by E_M^-. Next, we define

$$\begin{aligned} \text{for all } i \in \{1, \dots, k\} : I_{X_{v_{(1,i)}}} &= \left[0.4 - \frac{2i+1}{16k}, 0.4 - \frac{2i}{16k}\right], \text{ and} \\ \text{for all } i \in \{1, \dots, n-k\} : I_{X_{v_{(2,i)}}} &= \left[0.6 + \frac{2i}{16(n-k)}, 0.6 + \frac{2i+1}{16(n-k)}\right]. \end{aligned}$$

This ensures that for all vertices $u \in V_1$, the interval I_{X_u} lies in $(0.2, 0.4)$, and symmetrically for all vertices $v \in V_2$, the interval I_{X_v} lies in $(0.6, 0.8)$, hence satisfying E_*^-. Further, for all vertices $u, v \in V$ with $u \neq v$ their intervals I_{X_u}, I_{X_v} are disjoint, thereby satisfying E_V^-. To conclude the construction of the interval representation, we set

$$\begin{aligned} \text{for all } v \in V_1 : I_{A_v} = I_{B_v} = [\ell(I_{X_v}), 0.5], \text{ and} \\ \text{for all } v \in V_2 : I_{A_v} = I_{B_v} = [0.5, r(I_{X_v})]. \end{aligned}$$

Now, all the constraints set in E^+, E_S^- and E_T^- are satisfied by construction. It is left to check whether the constraints set by E_E^- are satisfied. Here, I_{X_v} must not overlap I_{A_u} or I_{B_u} if there exists a directed edge $(u, v) \in E$. This is trivially satisfied if $u \in V_1$ and $v \in V_2$ or vice-versa. If both $u, v \in V_2$, then in the constructed interval representation we must have that $\ell(I_{X_v}) > r(I_{A_u}) = r(I_{B_u}) = r(I_{X_u})$. As the intervals $\{I_{X_v} : v \in V_2\}$ were constructed according to a topological ordering of V_2, this is always satisfied. The argument works symmetrically for V_1, and hence the interval representation is conflict-free.

Conversely, suppose the constructed instance G admits a conflict-free interval representation in Opinion Interval Representation. We claim that this implies H is a YES-instance of Acyclic Digraph Partition. First, by Lemmas A.1 and A.2, any conflict-free interval representation places I_S and I_T at the extreme left and extreme right, respectively. Consequently, in the open interval $\big(r(I_S), \ell(I_T)\big)$, the intervals I_L, I_M, and I_R appear in that left-to-right order. Next, Lemma A.3 guarantees that every interval I_{M_v} for $v \in V$ is contained in I_M. Meanwhile, Lemma A.4 ensures that each I_{X_v} is contained entirely in either I_L or I_R. This setup naturally suggests a bipartition of the set V:

$$V_1 = \{v \in V : I_{X_v} \subset I_L\} \quad \text{and} \quad V_2 = \{v \in V : I_{X_v} \subset I_R\}.$$

We claim that $H[V_1]$ and $H[V_2]$ must each be acyclic. Indeed, in Lemma A.5 we show that for any directed edge $(u, v) \in E$ with $u, v \in V_1$, the intervals I_{X_u} and I_{X_v} in I_L must satisfy $r(I_{X_v}) < \ell(I_{X_u})$. Hence, the interval I_{X_v} must lie to the left of I_{X_u}. This implies a topological ordering of vertices in V_1, and thus prevents directed cycles in $H[V_1]$. A symmetric argument shows that $H[V_2]$ is acyclic. Thus, H admits a partition of its vertex set into two DAGs $H[V_1]$ and $H[V_2]$. Therefore, H is a YES-instance of Acyclic Digraph Partition, completing the proof of Theorem 2.2.

B A PTAS for a Fixed Number of Intervals

We prove Theorem 2.3, which generalizes a result of Giotis and Guruswami [24] for the agreement version of Correlation Clustering in complete signed graphs and also for a fixed number of clusters. Our analysis is similar to that of [24], but we have to adjust it such that we take into account the overlap of the given intervals.

Interestingly, Giotis and Guruswami [24] also presented a PTAS for the disagreement version of Correlation Clustering. However, their result does not extend to Opinion Interval Representation, since in our setting we may have overlapping intervals and this breaks their greedy assignment rule, as well as several of their technical arguments.

In the following, we consider a complete signed graph $G = (V, E^+ \cup E^-)$. We assume that we are given k intervals $I_1, \dots, I_k \subseteq \mathbb{R}$ as input. In the problem we consider, each vertex must be assigned to one of the intervals such that agreement is maximized.

The main work of this section will go into the proof of the following proposition from which the rest of our results follow.

Proposition B.1. *Let $\varepsilon > 0$ and $\delta > 0$. With probability at least $1-\delta$, Algorithm 2 computes an approximate solution with additive error at most $\varepsilon n^2/2$ and has running time $k^{O(1/\varepsilon^3 \log(k/(\varepsilon\delta)))} \cdot n$.*

The proposition allows us to obtain Corollary B.2, which shows that we can obtain a multiplicative $(1+\varepsilon)$-approximation guarantee for k fixed intervals. This then also implies the proof of Theorem 2.3.

Corollary B.2. *Let $\varepsilon > 0$ and $\delta > 0$. There exists an algorithm that computes a $(1+\varepsilon)$-approximate solution for* Opinion Interval Representation *with k given intervals in time $k^{O(1/\varepsilon^3 \log(k/(\varepsilon\delta)))}$ with probability at least $1-\delta$.*

Proof. First, assume that all pairs of intervals overlap. Then it does not matter how we assign the vertices because only the positive edges can be satisfied. In this case, any assignment will achieve the same objective function value as OPT. Second, assume that there is at least one pair of non-overlapping intervals. In that case, we know from Correlation Clustering that the objective function agreement must be $\Omega(n^2)$ (see the proof of Theorem 3.1 in [24]). In that case, we can use the result from Proposition B.1 to obtain a multiplicative $(1+\varepsilon)$-approximation by making the parameter ε in our additive approximation small enough. The running time claim follows from Proposition B.1. □

Proof of Theorem 2.3. Note that we only need to consider at most $2^{\binom{k}{2}}$ choices for picking k intervals (up to changing their coordinates): For any pair of intervals they either overlap or they do not. Thus, since there $\binom{k}{2}$ interval pairs, there are at most $2^{\binom{k}{2}} = 2^{O(k^2)}$ choices for the overlap.

Now we can just enumerate all possible overlap-patterns of k intervals and run the algorithm from Corollary B.2 for it in time $2^{O(k^2+1/\varepsilon^3 \log(k)\log(k/(\varepsilon\delta)))} \cdot n \leq 2^{O(k^2 \log(k/(\varepsilon\delta))/\varepsilon^3)} \cdot n$. Note that our result is correct if the algorithm succeeds for the k intervals picked by OPT, and thus we get the desired running time (and in particular we do not have to apply a union bound that the algorithm succeeds for all possible choices of $2^{O(k^2)}$ choices of intervals). □

For the remainder of this section, we work on the proof of Proposition B.1.

We present the pseudocode of our method with full details in Algorithm 2. On a high level, our algorithm works by partitioning V into $m = O(1/\varepsilon)$ equally-sized subsets $V_1, \dots, V_m$. Then, for each $i = 1, \dots, m$, we proceed as follows. We sample a set of vertices $S_i \subseteq V \setminus V_i$ of size $\tilde{O}\left(1/\varepsilon^2\right)$. Now, we enumerate all possible assignments of S_i into $(S_{i,1}, \dots, S_{i,k})$, where $S_{i,\ell} \subseteq S_i$ are the vertices assigned to interval I_ℓ, and for each such assignment, we greedily assign the vertices $v \in V_i$ to the interval that maximizes the agreement of v's edges to the clustering of S_i given by $(S_{i,1}, \dots, S_{i,k})$. This process gives a clustering of V_i and we build the final clustering by merging our solutions for the V_i to obtain a global clustering of V.

For a high-level description of the algorithm, see Section 2.2.

Recall from the main text that we write $\mathsf{overlap}(\ell)$ to denote the set of all intervals $I_{\ell'}$ that overlap with interval I_ℓ, i.e., $\mathsf{overlap}(\ell) = \{\ell' \colon I_\ell \cap I_{\ell'} \neq \emptyset\}$. Furthermore, a *clustering* $C_1, \dots, C_k$ of V is an assignment of the vertices V to intervals. In particular, C_i denotes all vertices which are assigned to interval I_i. Note that the C_i are mutually disjoint. For a vertex u and a clustering $C_1, \dots, C_k$ we write

$$\mathsf{agree}(u, \ell, (C_1, \dots, C_k)) = \sum_{\ell' \in \mathsf{overlap}(\ell)} \left|N^+(u) \cap C_{\ell'}\right| + \sum_{\ell' \notin \mathsf{overlap}(\ell)} \left|N^-(u) \cap C_{\ell'}\right|$$

which is the number of agreeing edges of vertex u for the clustering $C_1, \dots, C_k$ when assigning u to interval I_ℓ. Similarly, we define

$$\mathsf{agree}^+(u, \ell, (C_1, \dots, C_k)) = \sum_{\ell' \in \mathsf{overlap}(\ell)} \left|N^+(u) \cap C_{\ell'}\right|,$$

and

$$\mathsf{agree}^-(u, \ell, (C_1, \dots, C_k)) = \sum_{\ell' \notin \mathsf{overlap}(\ell)} \left|N^-(u) \cap C_{\ell'}\right|.$$

Algorithm 2: Maximizing agreement for fixed k

```
Input: A complete signed graph $G = (V, E^+ \cup E^-)$, contiguous and non-empty intervals
       $I_1, \dots, I_k \subseteq \mathbb{R}$, $\varepsilon > 0$
Result: A clustering $(\mathsf{ALG}_1, \dots, \mathsf{ALG}_k)$ maximizing the agreement
Partition $V$ into $m = \frac{4}{\varepsilon}$ sets $V_1, \dots, V_m$ of size $\frac{n}{m} = \frac{\varepsilon n}{4}$ each;
Sample $S_i \subseteq V \setminus V_i$ uniformly at random with replacement of size $s = \frac{32^2}{2\varepsilon^2} \log\left(\frac{64mk}{\varepsilon\delta}\right)$ for all
  $i = 1, \dots, m$;
Initialize some arbitrary clustering $(\mathsf{ALG}_1, \dots, \mathsf{ALG}_k)$;
for all possible clusterings of all $S_i$ into $(S_{i,1}, \dots, S_{i,k})$ do
    for $i = 1, \dots, m$ do
        Let $\mathsf{ALG}'_{i,1}, \dots, \mathsf{ALG}'_{i,k}$ be an empty clustering of $V_i$;
        for $u \in V_i$ do
            $\ell^* \leftarrow \operatorname{argmax}_{\ell=1,\dots,k} \mathsf{agree}(u, \ell, (S_{i,1}, \dots, S_{i,k}))$;
            Assign $u$ to $\mathsf{ALG}_{i,\ell^*}$;
    Set $\mathsf{ALG}'_\ell \leftarrow \bigcup_{i=1}^m \mathsf{ALG}'_{i,\ell}$ for all $\ell = 1, \dots, k$;
    if $\mathsf{agree}(G, (\mathsf{ALG}'_1, \dots, \mathsf{ALG}'_k)) > \mathsf{agree}(G, (\mathsf{ALG}_1, \dots, \mathsf{ALG}_k))$ then
        Set $(\mathsf{ALG}_1, \dots, \mathsf{ALG}_k) \leftarrow (\mathsf{ALG}'_1, \dots, \mathsf{ALG}'_k)$;
return $(\mathsf{ALG}_1, \dots, \mathsf{ALG}_k)$;
```

For the analysis we consider an optimal clustering denoted by $\mathsf{OPT} = (\mathsf{OPT}_1, \dots, \mathsf{OPT}_k)$. Here, $\mathsf{OPT}_\ell \subseteq V$ consists of all vertices that get assigned to interval I_ℓ in the optimal solution.

We set $\mathsf{OPT}_{i,\ell} = V_i \cap \mathsf{OPT}_\ell$. Note that $\mathsf{OPT}^i = (\mathsf{OPT}_{i,1}, \dots, \mathsf{OPT}_{i,k})$ is the clustering of OPT when constrained on the vertices in V_i.

Next, we construct a set of *hybrid* clusterings that use a part of our solution from ALG and a part of the solution from OPT. In particular, we set

$$H_{i,\ell} = \left(\bigcup_{j=1}^{i-1} \mathsf{ALG}_{j,\ell} \right) \cup \left(\bigcup_{j=i}^{m} \mathsf{OPT}_{j,\ell} \right).$$

Note that $H_{i,\ell}$ corresponds to a hybrid between ALG_ℓ and OPT_ℓ where all vertices in $V_1, \dots, V_{i-1}$ are clustered based on ALG_ℓ and all vertices in $V_i, \dots, V_m$ are clustered based on OPT_ℓ.

Now, additionally we set $\mathcal{H}_{i,\ell} = H_{i,\ell} \setminus V_i$, i.e., these are all vertices in $H_{i,\ell}$ which are not contained in V_i and thus they will be clustered either before or after the i'th iteration of our algorithm. We set $H_i = (H_{i,1}, \dots, H_{i,k})$ and note that H_i is a solution for all vertices (not just the vertices in V_i). In particular, note that $H_1 = \mathsf{OPT}$ and $H_{m+1} = \mathsf{ALG}$. Additionally, we set $\mathcal{H}_i = (\mathcal{H}_{i,1}, \dots, \mathcal{H}_{i,k})$ for the clustering given by H_i after removing the vertices in V_i.

For the rest of the analysis, for all $i = 1, \dots, m$, we consider the clustering $S_{i,1}, \dots, S_{i,k}$ of S_i that agrees with the hybrid clustering, i.e., we assume that $S_{i,\ell} = S_i \cap \mathcal{H}_{i,\ell}$. Note that this clustering must be considered by the algorithm since we exhaustively enumerate all possible clusterings of all S_i.

Lemma B.3. *Let $i \in \{1, \dots, m\}$. With probability at least $1 - \frac{\delta}{4m}$ over the randomness in S_i the following event happens: For at least a $(1 - \varepsilon/8)$-fraction of the vertices $u \in V_i$ it holds that for all $\ell = 1, \dots, k$,*

$$\left| \frac{|V \setminus V_i|}{s} \mathsf{agree}^+(u, \ell, (S_{i,1}, \dots, S_{i,k})) - \mathsf{agree}^+(u, \ell, (\mathcal{H}_{i,1}, \dots, \mathcal{H}_{i,k})) \right| \leq \frac{\varepsilon}{32} |V \setminus V_i| . \quad (2)$$

Proof. Consider any $u \in V_i$ and let $S_i = \{v_1, \dots, v_s\}$. For $\ell = 1, \dots, k$ and $j = 1, \dots, s$, let $X_{j,\ell}$ be the indicator random variable which is 1 if $v_j \in N^+(u)$ and $v_j \in S_{i,\ell'}$ for some cluster with $\ell' \in \mathsf{overlap}(\ell)$, and 0 otherwise.

Note that $\sum_{j=1}^s X_{j,\ell} = \mathsf{agree}^+(u, \ell, (S_{i,1}, \dots, S_{i,k}))$ and that the $X_{j,\ell}$ are i.i.d. random variables with expectation $\mathbb{E}[X_{j,\ell}] = \mathbf{Pr}(X_{j,\ell} = 1) = \frac{\mathsf{agree}^+(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k}))}{|V \setminus V_i|}$. Now an additive Chernoff bound gives that

$$\begin{aligned}&\mathbf{Pr}\left(\left|\frac{\mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k}))}{s} - \frac{\mathsf{agree}^+(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k}))}{|V\setminus V_i|}\right| > \frac{\varepsilon}{32}\right)\\ &< 2\exp\left(-2\left(\frac{\varepsilon}{32}\right)^2 s\right) < \frac{\varepsilon\delta}{32mk}.\end{aligned}$$

Note that this gives us the inequality from the lemma after multiplying with $|V \setminus V_i|$ on both sides.

Now let Y be the random variable denoting the number of vertices in V_i which do not satisfy the inequality above. Observe that $\mathbb{E}[Y] < \frac{\varepsilon\delta}{32mk}|V_i|$. By Markov's inequality, we get that the inequality holds for all but $\frac{\varepsilon}{8}|V_i|$ vertices with probability at least $1 - \frac{\delta}{4mk}$.

Now the lemma follows by applying a union bound. □

We note that the lemma also holds for $\mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k}))$ with the same proof.

Lemma B.4. *For $i = 0,\dots,m$ it holds that $\mathsf{agree}(G,H_{i+1}) \geq \mathsf{agree}(G,\mathsf{OPT}) - i \cdot \frac{1}{8}\varepsilon^2 n^2$.*

Proof. Consider some iteration i of the algorithm. Note that in this iteration only the vertices in V_i are assigned to clusters and thus H_i and H_{i+1} only differ by the vertices contained in V_i. Therefore, our proof will proceed by considering a vertex $u \in V_i$ that gets assigned to interval I_ℓ by the algorithm but to interval $I_{\ell'}$ in the solution H_i.

First, observe that since the algorithm assigned u to interval I_ℓ we must have that

$$\begin{aligned}&\mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k})) + \mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k}))\\ &\geq \mathsf{agree}^+(u,\ell',(S_{i,1},\dots,S_{i,k})) + \mathsf{agree}^-(u,\ell',(S_{i,1},\dots,S_{i,k}))\end{aligned}$$

which implies that

$$\begin{aligned}&\mathsf{agree}^+(u,\ell',(S_{i,1},\dots,S_{i,k})) + \mathsf{agree}^-(u,\ell',(S_{i,1},\dots,S_{i,k}))\\ &\quad - \mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k})) - \mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k})) \leq 0.\end{aligned}$$

Now set $\alpha_i = \frac{|V\setminus V_i|}{s}$ and assume that u is a vertex satisfying Equation (2). Observe that the number of agreements we might lose by this misplacement is at most

$$\begin{aligned}&\mathsf{agree}(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) - \mathsf{agree}(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k}))\\
&= \mathsf{agree}^+(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) + \mathsf{agree}^-(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k}))\\
&\quad - \mathsf{agree}^+(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) - \mathsf{agree}^-(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k}))\\
&= \mathsf{agree}^+(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) + \alpha_i\mathsf{agree}^+(u,\ell',(S_{i,1},\dots,S_{i,k})) - \alpha_i\mathsf{agree}^+(u,\ell',(S_{i,1},\dots,S_{i,k}))\\
&\quad + \mathsf{agree}^-(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) + \alpha_i\mathsf{agree}^-(u,\ell',(S_{i,1},\dots,S_{i,k})) - \alpha_i\mathsf{agree}^-(u,\ell',(S_{i,1},\dots,S_{i,k}))\\
&\quad - \mathsf{agree}^+(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) + \alpha_i\mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k})) - \alpha_i\mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k}))\\
&\quad - \mathsf{agree}^-(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) + \alpha_i\mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k})) - \alpha_i\mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k}))\\
&\leq \left|\mathsf{agree}^+(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) - \alpha_i\mathsf{agree}^+(u,\ell',(S_{i,1},\dots,S_{i,k}))\right|\\
&\quad + \left|\mathsf{agree}^-(u,\ell',(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) - \alpha_i\mathsf{agree}^-(u,\ell',(S_{i,1},\dots,S_{i,k}))\right|\\
&\quad + \left|\mathsf{agree}^+(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) - \alpha_i\mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k}))\right|\\
&\quad + \left|\mathsf{agree}^-(u,\ell,(\mathcal{H}_{i,1},\dots,\mathcal{H}_{i,k})) - \alpha_i\mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k}))\right|\\
&\quad + \alpha_i\mathsf{agree}^+(u,\ell',(S_{i,1},\dots,S_{i,k})) + \alpha_i\mathsf{agree}^-(u,\ell',(S_{i,1},\dots,S_{i,k}))\\
&\qquad - \alpha_i\mathsf{agree}^+(u,\ell,(S_{i,1},\dots,S_{i,k})) - \alpha_i\mathsf{agree}^-(u,\ell,(S_{i,1},\dots,S_{i,k}))\\
&\leq 4\cdot\frac{\varepsilon}{32}|V\setminus V_i| + 0\\
&\leq \frac{\varepsilon}{8}n.\end{aligned}$$

Thus we get that for all vertices in V_i satisfying Equation (2) we get that their total difference is at most $\frac{\varepsilon}{8}n\,|V_i| = \frac{\varepsilon^2 n^2}{32}$.

Table 2: Summary of our datasets.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert E^+\rvert$	$\lvert E^-\rvert$
BitcoinOTC	5 881	21 434	18 281	3 153
Chess	7 301	32 650	19 046	13 604
WikiElec	7 115	100 355	78 440	21 915
Bundestag	1 480	397 497	320 956	76 541
Slashdot	82 140	498 532	380 933	117 599
Epinions	131 580	708 507	589 888	118 619
WikiSigned	138 587	712 337	628 000	84 337
WikiConflict	116 836	2 014 053	762 999	1 251 054

Furthermore, as there are at most $\frac{\varepsilon}{8}\lvert V_i\rvert$ vertices that do not satisfy Equation (2), they can contribute at most $\frac{\varepsilon}{8}\lvert V_i\rvert n = \frac{\varepsilon^2 n^2}{32}$ edges that are in disagreement. The number of disagreements within V_i is at most $\lvert V_i\rvert^2 = \frac{\varepsilon^2 n^2}{16}$.

In total, we get that we have introduced $\frac{\varepsilon^2 n^2}{8}$ new disagreements due to our approximations in the i'th iteration. The lemma now follows by induction. □

Proof of Proposition B.1. The approximation ratio follows from the previous lemma. The running time follows from the fact that there are $k^s = k^{O(1/\varepsilon^2 \log(k/(\varepsilon\delta)))}$ choices to assign the s vertices in each S_i to the k intervals and we have to consider the combinations of these assignments for $m = O(1/\varepsilon)$ sets. Thus, the outer loop iterates over $k^{ms} = k^{O(1/\varepsilon^3 \log(k/(\varepsilon\delta)))}$ assignments. Furthermore, each loop iteration can be implemented in time $O(ns)$. □

C Additional Experiment Details and Results

C.1 Experiment setup

Hardware. All our algorithms are implemented in the Rust programming language. The experiments were run on a system with two AMD EPYC 9124 CPUs, 500 GB of RAM, and an NVIDIA RTX 4000 Ada Generation GPU.

Dataset preprocessing. As a preprocessing step, we convert all networks into simple undirected graphs by removing loops and multiple edges. When removing multiple edges between two vertices, we replace them with a single undirected edge whose weight is the sum of the original edges. Finally, we apply a thresholding function where positive edges are assigned weight 1, and negative edges are assigned weight -1. An overview of all our datasets is shown in Table 2.

C.2 Parameters and implementation details for VENUS and GAIA

The pseudocode of GAIA is shown in Algorithm 1. The algorithm uses a greedy approach and employs randomness when partially destroying and reconstructing solutions to improve the overall objective.

Batch size. After one initial assignment performed over the full vertex set, V is partitioned into sets of size $\frac{n}{m}$. Sequentially, each of these m subsets is unassigned and reassigned greedily. In the presented results, we generally use $m = 10$ for the sake of consistency. Depending on the specific dataset, higher or lower values might be advantageous. Lower numbers result in more aggressive reassignments and higher numbers in smaller, more incremental changes. For example, for BitcoinOTC, we observed that solutions with $m = 100$ on average achieve 1% better objective than with $m = 10$.

Breaking ties and randomization. An essential aspect of the implementation is randomised tie breaking for both the order of vertex assignment *and* the selection of the assigned cluster. Without the extensive use of randomization, the solution quality suffers greatly, and GAIA can get stuck in bad local optima early. The same can be said for the ordering of the reassignment: if vertices are reassigned in the same order in every iteration, the performance of the algorithms is substantially reduced. In our implementation, vertices are greedily assigned during the first epoch, and in each subsequent epoch, they are randomly partitioned into batches.

Vertex priority. In each batch of vertices, we select the assignment order based on the maximal potential agreement of the vertex. In our preliminary experiments performed during the algorithm development, this strategy was more effective than using a randomized order or using different characteristics like the vertex degree for ordering. However, the performance does not differ greatly, so other methods of selecting vertex ordering might be considered in the future.

Timeout and early stopping. In the presented results, both GAIA and VENUS terminate either after 30 minutes by timeout, or until the best found solution could not be improved for 50 consecutive epochs. The condition for early stopping is intentionally chosen in a very conservative manner, as often the algorithms will find very marginal improvements to their solutions late in the optimization process. To reduce runtime, using, e.g., a time limit of 3 minutes instead of 30 massively reduces solving time while only causing a significant decrease in objective value for the Epinions and WikiConflict dataset, with an average relative difference of 1% and 10% respectively.

Simulated annealing for VENUS. With the implementation of GAIA, even with the added randomness, a high variance in the solution quality can be observed. This, paired with the fast convergence time, motivated a second approach with simulated annealing: VENUS. For VENUS, every part of the algorithm remains unchanged, besides the assignment process in Line 7 of Algorithm 1. We replace the deterministic argmax with a temperature-scaled softmax over the agreement, where we define softmax over some function f as

$$\operatorname{softmax}_{\ell=1\ldots k} f(\ell) = \frac{\exp(f(\ell))}{\sum_{\ell'=1\ldots k}\exp(f(\ell'))}. \tag{3}$$

For our setting, we use the agreement scaled by a temperature parameter t to control the level of randomness, with higher levels of t yielding close to uniformly random distributions, and small values of t resulting in increasingly greedy solutions. In all our experiments, we use an initial temperature of $t_0 = 100$, and use an exponential decay schedule where after every $\tau = 5$ epochs we let $t \leftarrow t \cdot \alpha$, where $\alpha = \frac{2}{3}$.

Interval structure. In all experimental results presented in the main text, we use 8-Chains as the interval structure for both GAIA and VENUS. As seen in Figure 4b, using larger and thus more expressive chain-like structures did not substantially improve the objective value. Hence, we report the results on an 8-Chain for consistency and to strike a balance between interpretability and expressivity.

C.3 Impact of simulated annealing on solution quality

Next, we study how much the simulated annealing used in VENUS impacts its results compared to GAIA. Our detailed results on the real-world datasets are shown in Table 3. For almost all datasets, the average objective function of solutions found by VENUS is better than those found by GAIA. This demonstrates how simulated annealing can avoid local minima and continue the optimization process toward better solutions. Additionally, VENUS generally exhibits smaller standard deviations, indicating more consistency in the approach. A clear outlier to this is the Epinions dataset, where the trend reverses and the purely greedy approach performs better; we tried to understand the reason for this behavior but could not find a clear indicator.

C.4 Benchmarking against CORRELATION CLUSTERING baselines

In addition to using GAIA and VENUS to find interval assignments with partially overlapping structures, we also evaluate their performance on structures where intervals are non-overlapping (pairwise disjoint). As described in Section 2, this setup corresponds to the CORRELATION CLUSTERING problem with a fixed number of clusters. Results for a structure with 8 disjoint intervals are shown in Tables 4 and 5. Across all datasets, GAIA and VENUS achieve objective values within 0.5 percentage points of the best CORRELATION CLUSTERING baselines, despite being constrained to only 8 clusters. Moreover, the solutions produced by GAIA and VENUS show low variance, with a standard deviation of under 100 violations on all but one dataset. This indicates that the algorithms consistently found strong solutions, particularly with less variance than when using overlapping interval structures (see Table 3).

Table 3: Comparison of GAIA and VENUS for solving Opinion Interval Representation. We report the minimum disagreement (in absolute number of edges), as well as averages and standard deviations over 50 runs. The solutions were computed for a fixed 8-Chain interval structure and 10 reassignment batches.

	Gaia		Venus	
Dataset	Best	Avg±Std	Best	Avg±Std
BitcoinOTC	711	767±26	760	783±11
Chess	6 472	6 573±52	6 410	6 486±36
WikiElec	11 275	11 459±143	11 297	11 420±75
Bundestag	1 001	9 054±4 357	1 001	2 078±3 244
Slashdot	45 120	46 218±860	44 563	44 884±473
Epinions	31 687	32 789±1 300	31 286	33 258±1 306
WikiSigned	35 204	36 270±509	34 564	34 971±165
WikiConflict	69 344	69 847±193	69 014	69 264±141

Table 4: Comparison of our algorithms and Correlation Clustering baselines for solving Correlation Clustering. Reported is the percentage of edges violated in the solution (lower is better). Our algorithms use a fixed structure of 8 distinct clusters, while the Correlation Clustering algorithms are not restricted in the number of clusters.

	Our algorithms		Correlation Clustering baselines			
Dataset	GAIA	VENUS	GAEC	GAECKLj	SCMLEvo	RAMA
BitcoinOTC	**5.57**	**5.57**	5.58	**5.57**	**5.57**	5.64
Chess	27.67	27.75	28.64	28.10	**27.33**	39.98
WikiElec	**14.13**	**14.13**	**14.13**	**14.13**	**14.13**	14.45
Bundestag	**2.95**	**2.95**	3.06	**2.95**	**2.95**	3.72
Slashdot	13.70	13.59	13.75	13.66	**13.52**	17.17
Epinions	6.69	6.71	6.83	6.68	**6.67**	6.86
WikiSigned	6.30	6.21	**6.17**	**6.17**	**6.17**	6.96
WikiConflict	5.85	5.85	5.87	**5.82**	**5.82**	6.02

We believe that this finding is highly interesting, since it shows that our methods can find Correlation Clustering solutions that are on par with state-of-the-art algorithms, while also being able to solve our more general Opinion Interval Representation problem.

Additionally, it is notable that we find competitive solutions even though we only use 8 clusters, whereas the baselines might use an unrestricted number. However, this finding is not completely new and echoes findings by Brusco and Doreian [8], who observed similar objective values using few clusters on the WikiElec and Slashdot datasets.

C.5 Scalability and runtime analysis

The time until convergence for GAIA and VENUS is shown in Table 6 and Figure 6. For both algorithms, the running time scales roughly linearly with the size of the graph and the memory usage is at most 260 megabytes for the largest datasets. Despite the artificial slowdown of convergence during early epochs, VENUS is remarkably not notably slower to converge than GAIA. Further, we note that both the memory and runtime complexity of our algorithms depend linearly on the number of intervals, so the running time can vary with the size of the interval structure.

C.6 Case study

The Bundestag *dataset.* Our novel Bundestag dataset was constructed by scraping all the roll-call voting data from the German parliament between October 18th, 2012, and March 18th, 2025.[2] As official works, this voting data is not subject to copyright. From the voting data, we then generated

[2] This data is publicly available at https://www.bundestag.de/parlament/plenum/abstimmung/liste

Table 5: Comparison of the results of `GAIA` and `VENUS` when solving Correlation Clustering. We report the minimum disagreement (in absolute number of edges), as well as averages and standard deviations over 50 runs. The solutions were computed for 8 distinct clusters (i.e., 8 non-overlapping intervals) and 10 reassignment batches.

	Gaia		Venus	
Dataset	Best	Avg±Std	Best	Avg±Std
BitcoinOTC	1 194	1 201±3	1 194	1 200±4
Chess	9 035	9 128±55	9 061	9 132±72
WikiElec	14 181	14 185±2	14 182	14 185±2
Bundestag	11 736	11 756±27	11 725	11 736±9
Slashdot	68 276	68 351±30	67 759	68 334±88
Epinions	47 412	47 499±36	47 570	48 750±434
WikiSigned	44 881	44 992±148	44 217	44 274±23
WikiConflict	117 814	117 882±23	117 885	117 932±26

Table 6: Time until convergence averaged over 50 runs on different instances. We report the runtime in seconds until 50 full epochs without improvement (lower is better). Our algorithms use a fixed `8-Chain` interval structure and 10 batches for vertex reassignment.

	GAIA			VENUS		
Dataset	Best	Avg±Std	Worst	Best	Avg±Std	Worst
BitcoinOTC	1.2	2.5±0.9	5.7	1.8	3.2±0.8	6.2
Chess	3.7	8.4±2.4	15.2	3.8	8.5±2.1	13.4
WikiElec	6.1	16.2±5.5	31.1	7.1	15.9±6.2	41.5
Bundestag	5.5	8.8±1.9	16.2	10.9	15.2±2.4	22.8
Slashdot	107.5	251.9±114.8	781.2	98.9	173.8±45.3	348.5
Epinions	182.3	369.1±129.4	954.8	196.0	390.8±143.0	897.4
WikiSigned	169.6	353.5±94.3	602.9	214.6	357.5±90.6	585.3
WikiConflict	436.2	874.0±192.8	1 470.0	372.9	695.1±183.7	1 220.0

a signed graph by representing each member with a vertex and assigning a positive edge if two members vote the same way in at least 75% of sessions they both attended, and a negative edge if their votes aligned in 25% of sessions or less. This left two politicians without any edges, which were thus excluded from the signed graph. Notably, the voting data includes several politicians who changed parties during their parliamentary careers. In such cases, we treat the politicians as members of the party they were first affiliated with for visualization purposes. This could also explain slight within-party differences, e.g., the LINKE party split between 2023 and 2024.

Generation of Figure 1. With the preprocessed Bundestag instance, we let `VENUS` compute a solution and for each politician we visualize their party membership via coloring the corresponding point and the positioning it on the y-axis. The interval assignment produced by `VENUS` determines the x-position. We then slightly adjust the position inside the bins to indicate the affinity of each politician as follows: For the vertex $v \in C_\ell$ corresponding to the politician, we define its *affinity* to other clusters $C_{\ell'}$ as

$$\mathsf{affinity}(v, C_{\ell'}) = \left|N^+(v) \cap C_{\ell'}\right| - \left|N^-(v) \cap C_{\ell'}\right| . \tag{4}$$

Then we calculate the x-perturbation of v, denoted by v_x, as

$$v_x = \tanh \sum_{\ell' \neq \ell} \frac{\mathsf{affinity}(v, C_{\ell'})}{(\ell' - \ell)(|N^+(v)| + |N^-(v)|)} . \tag{5}$$

In words, the position of a vertex is shifted to the left or to the right according to the assignment of its neighbors, depending on their distance. Neighbors assigned to closer intervals affect the position more, and neighbors assigned further away affect the position less. The direction of this influence depends on the edge sign.

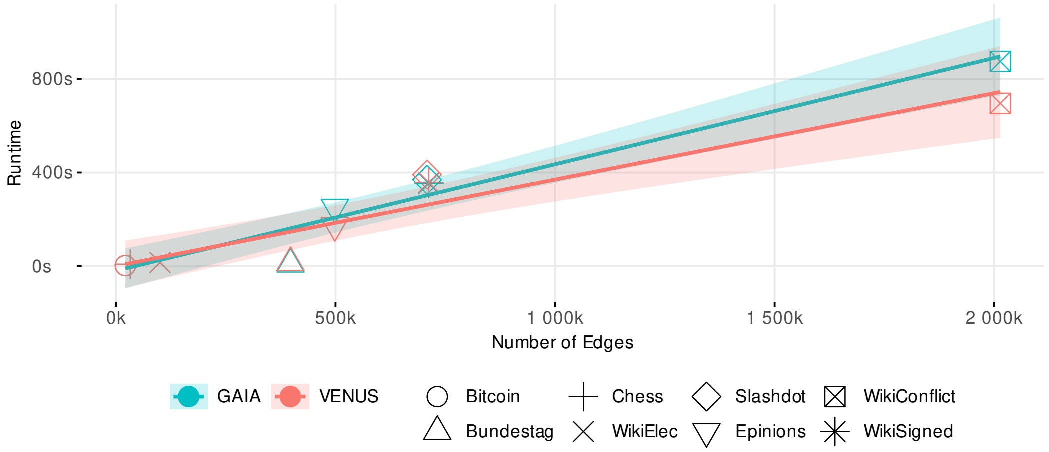


Figure 6: Time until convergence per instance.

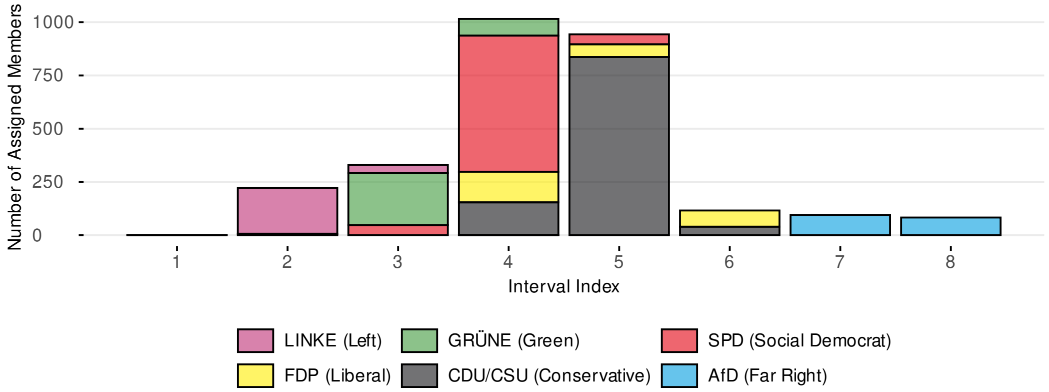


Figure 7: An alternative visualization of VENUS's results on the Bundestag dataset. Here, we present party affiliations of Bundestag members assigned to each interval.

This function simply serves the purpose of visualization, to show connectivity across intervals. It is not a direct output of our algorithm and does not have a fixed interpretation. Finally, we slightly adjust the position of the vertices in the plot with small amounts of Gaussian noise to avoid over-plotting. The figure also includes a violin plot of each political party.

Results. The solution found by VENUS is highly interpretable and can be used to generate insight about the co-voting behavior in the German parliament. This interpretation is discussed in Section 4, with an alternative visualisation in Figure 7, which shows an interval-centric view instead of the party-centric view in Figure 1. Figure 7 shows that each interval is dominated by members of one political party. Further, as discussed previously, the figure accurately reflects the political spectrum in Germany, with the exception of the FDP, which appears "split" between intervals 4–6. This can be justified by the different coalition structures in the last few legislative periods, as we discussed in the main text.