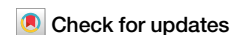


<https://doi.org/10.1038/s44260-025-00027-1>

Strong and weak random walks on signed networks

S. A. Babul^{1,2} , Y. Tian³ & R. Lambiotte^{1,2}

Random walks are essential for analyzing complex networks. On signed networks, where edges can be positive or negative, designing random walks that capture signed community structure is challenging. Communities in signed networks typically have predominantly positive internal edges and negative external edges. While prior methods focus on strong balance (two communities), this scenario is rare in empirical networks. We introduce a random walk framework tailored to weak balance, accommodating networks with more than two communities. This approach generates a similarity matrix that enables effective community detection. Comparing strong and weak walks on synthetic and empirical networks, we demonstrate that weak walks outperform strong walks in scenarios involving more than two communities or asymmetric link densities. Our findings suggest that replacing strong walks with weak walks could enhance other signed network random-walk algorithms, broadening their applicability to more realistic network structures.

Random walks play a central role in several aspects of network science¹. Random walks are often used as a linear, tractable model for diffusive processes on networks, allowing us to explore questions such as the impact of certain network properties², e.g. their community structure or their degree distribution, on the speed of the diffusion, e.g. via the mixing time³. Random walks are also a key ingredient in several methods to extract information from large-scale networks. Important areas of application include node centrality which can be captured by the density of walkers on that node in the stationary state⁴, community detection where groups of nodes can be defined as communities based on their tendency to retain flows of walkers for long times^{5,6}, or the design of kernels/embeddings capturing the similarity between nodes on a graph^{7,8}. Among the advantages of random-walk based approaches, one can cite their intuitive interpretation, in terms of flows along the edges, their connection to well-established theories, such as that of Markov chains, their linearity, opening up the toolbox of linear algebra, and their ease of use in a distributed setting, each of the walkers being independent from the others, and their capacity to probe the structure of the network at multiple scales.

In this article, our purpose is to explore how random walks can be defined on, and help us understand the structure of, signed networks. In signed networks, edges carry a value, that is positive or negative to model trustful or mistrustful, activatory or inhibitory, cooperative or antagonistic interactions, depending on the context^{9–11}. Random walks on signed networks have been studied in the past, either implicitly, by investigating the properties of the signed Laplacian e.g. for node embedding¹² or explicitly, by defining rules for how traversing a negative edge affects the random

walker¹³, in a model that we will refer to as a strong random walk for reasons that will appear clear later. As in many generalisation of “classical networks”, generalising the notion of random walks to signed networks is not unique, and the choice should be motivated either by the process that one aims to model or by the underlying structure that one would like to extract. What makes the power of random walks on unsigned networks is their deep connection to specific structural patterns, communities and bottlenecks, e.g. via the Cheeger inequality¹⁴. Communities can be seen as noisy connected components, i.e. connected components to which random edges have been added. The important question to define the *right* model of random walks on signed networks is thus: what is the prototypical structure that one aims to identify?

The large-scale structure of signed networks is usually viewed through the prism of balance theory. There exist two forms of balance. Strong balance¹⁵ refers to the situation when the network can be partitioned into two groups, such that positive edges are inside the groups, and negative edges between them. Equivalently, balanced networks are such that any cycle in the graph has an even number of negative edges. A weaker version of balance¹⁶ relies on the condition that there is no cycle with only one single negative edge. Graphs that exhibit weak balance can be partitioned into k clusters with positive edges inside, and negative edges connecting them. It is well-known that the signed Laplacian and the equivalent strong random walk allow to efficiently recover networks with polarisation between two groups, akin to strong balance¹². It is also well-known that they may fail to uncover and properly characterise networks exhibiting a structure related to weak balance, and thus to serve as a model for the inference of a structure

¹Mathematical Institute, University of Oxford, Oxford, UK. ²The Turing Institute, British Library, London, UK. ³Nordita, Stockholm University and KTH Royal Institute of Technology, Stockholm, Sweden. ✉e-mail: shazia.babul@new.ox.ac.uk; yu.tian@su.se; lambiotte@maths.ox.ac.uk

into more than two groups¹⁰. Alternative approaches have been proposed to solve this problem, for instance through linear programming as in the method proposed in¹⁷ which locates the optimal partition of the graph, or through other notions of Laplacian, e.g. SPONGE¹⁸ or the repelling Laplacian¹⁹, but they do not have a random walk interpretation and thus do not inherit the advantages described above. The main contribution of this article is the definition of a random walk process, called the weak random walk, specifically designed for structures derived from weak balance, hence particularly fitted for clustering problems involving more than two groups in signed networks. In essence, the process consists in allowing the walker to explore the graph around a seed node only until it encounters a negative edge for a second time.

This article is organised as follows. In section II, we provide an overview of the key concepts on unsigned networks and random walks, in particular related to the definition of a similarity matrix between nodes. In section III, we introduce a first type of random walk on signed networks, called the strong walk, and show its connections to the signed Laplacian. In section IV, we then discuss how to exploit such a random walk in order to define a similarity matrix between nodes of a signed network. In section V, we identify the limitations of the strong walks for networks exhibiting more than two clusters, and introduce weak walks to remedy the problem. In section VI and VII, we compare the results of simple unsupervised and semi-supervised algorithms based on the two types of walks, showing that weak walks present an advantage in terms of accuracy for certain families of networks. Finally, in section VIII, we conclude and discuss implications of our work for the design of algorithms.

Results

Notations and basics

We consider an undirected signed graph, characterised by the adjacency matrix A_{sij} , so that $A_{sij} = 1$ indicates the presence of a positive edge. $A_{sij} = -1$ indicates the presence of a negative edge. We decompose that matrix into two matrices, encoding the positive and negative edges separately, each associated to an unsigned, undirected network: $A_{sij} = A_{+ij} - A_{-ij}$. In the signed graph, the total number of connections of each node, positive or negative, is thus given by $k_i = \sum_j |A_{sij}|$, which defines the degree. The total number of nodes is denoted as N , and that of edges is denoted by M . In the following, we will also use the diagonal matrix of degrees $K_{ij} = k_i \delta_{ij}$ and the adjacency matrix of the graph where signs are not considered $A_{u,ij} = A_{+ij} + A_{-ij}$. Note that $k_i = \sum_j A_{u,ij}$. Our purpose is to model diffusive processes on signed networks. Here, we will focus on discrete-time walks, but everything can be generalised to continuous times. In the following derivations and experiments, we consider unweighted networks, but the generalisation to weighted ones is straightforward. The methods we describe can be applied to both complete and non-complete graphs, with the condition that each node must have at least one edge.

In the case when the graph does not have negative edges $A_{u,ij} = A_{+ij}$, a discrete-time random walk evolves according to

$$n_j(t+1) = \sum_i n_i(t) T_{u,ij} \quad (1)$$

where $n_i(t)$ is the density of walkers on node i at time t and $T_u = K^{-1}A_u$ is the transition matrix of the associated Markov chain. This equation describes a microscopic process where an ensemble of walkers all move synchronously at discrete times. At each step, a walker located on some node i jumps to one of its neighbours, with a probability equal to one over the number of such neighbouring nodes. The spectral properties of the transition matrix, and how they relate to the associated graph structure, are well known¹. In particular, we note that T_u shares the same spectrum as the symmetric matrix $K^{-1/2}A_uK^{-1/2}$, which implies that its eigenvalues are all real and that its left and right eigenvectors, $\mathbf{u}_\alpha$ and $\mathbf{v}_\alpha$ respectively, are related by $\mathbf{u}_\alpha = K\mathbf{v}_\alpha$. The eigenvalues λ_α are in the interval $[-1, 1]$, and the multiplicity of the dominant eigenvalue, $\lambda_1 = 1$ is equal to the number of connected components in the graph. The difference between λ_1 and the largest non-one

eigenvalue, $\lambda_1 - \lambda_2$, called the spectral gap, determines the mixing time of the process and quantifies the presence of bottlenecks hindering the flow of walkers across the system². When the graph is connected, its corresponding left eigenvector, denoted by π is given by $\pi_i = k_i/(2M)$, which characterises how walkers are distributed across the nodes in the stationary state.

As we mentioned in the introduction, random walks are at the core of several algorithms^{6,20}. In particular, they are useful to capture the strength of indirect connections between nodes. In situations when the graph is sparse, so that only a small fraction of the pairs of nodes are connected (This is the case in the vast majority of real-world applications, especially when the graph is sufficiently large.), random-walk-based kernels allow to estimate the proximity of pairs of nodes that are not directly connected, by incorporating, in an unbiased (or in a supervised) way, the information of walks of any length. In case of unsigned networks, the resulting similarity between two nodes is obtained from an appropriately weighted sum of the walks between them. Typically, the existence of many short walks between two nodes ensures their proximity. Important examples include the diffusion, or heat kernels^{21–23}, the mean commute time^{24,25}, but also the random-walk with restart similarity^{26–28}. For reasons that will appear clear later, and that random walks in our model on signed networks tend to stop after a finite number of steps, the latter generalises most naturally (according to us) in the signed context.

Random walks with restart are a special type of random walk with teleportation. Take a node i , we consider a process where a walker may, at each step, either perform a jump to one of the neighbouring nodes with probability p , or teleport to the original node i with probability $(1 - p)$. It is straightforward to show that its stationary state is given by

$$\mathbf{n} = (1 - p)\mathbf{e}_i(I - pT_u)^{-1}, \quad (2)$$

where $\mathbf{e}_i$ is a row vector made of zeros except for entry i where it takes the value 1. The stationary state is, intuitively, peaked around the pinned node i , and becomes more spread, giving more importance to distant neighbours, when p increases. When p approaches 1, more and more importance is given to longer walks, and the vector converges to the stationary state π defined above. After taking a Taylor series of this expression, it is very similar to the kernel used in Deepwalk²⁹, and defines the weighted neighborhood of each node at different scales.

A similarity matrix can be obtained by repeating the process for each node and by stacking the N vectors, leading to

$$\kappa(p) = (1 - p)K(I - pT_u)^{-1}, \quad (3)$$

where we scale here each line by the node degree, to ensure that the matrix is symmetric. This quantity, or variations around it, have been used in a variety of contexts⁷, including node classification and clustering. Note that

$$\begin{aligned} \kappa_{ij}(p) &= (1 - p)k_i[(I - pT_u)^{-1}]_{ij}, \\ \kappa_{ij}(p) &= (1 - p)k_i \left[\sum_k p^k T_u^k \right]_{ij}. \end{aligned} \quad (4)$$

As each term $k_i T_{u,ij}^k$ takes the (i, j) element from $A_u K^{-1} A_u \dots K^{-1} A_u$, $\kappa(p)$ is symmetric for any value of p (The symmetry arises due to the detailed balance when teleportation to nodes is done proportional to the stationary state of the walk without restart.). Moreover, the i^{th} line (row) sums to k_i , meaning that $\kappa(p)$ can be interpreted as a weighted graph, interpolating between the original graph (with the addition of self-loops) when p approaches 0 and low rank matrix obtained from the outer product of the stationary states, $k_i k_j / (2M)$, when p approaches 1. For any value of p , $\kappa_{ij}(p)$ can be fed into standard methods for weighted networks. For instance, its Newman-Girvan modularity³⁰ can be used to uncover its multi-scale modularity, an approach reminiscent of Markov stability^{31–33}.

Strong walks

Introduction and basic properties. To generalize the classical methods defined in the previous section and, more generally, any algorithm based on random walks, one needs ways to define a random walk on a signed network. The natural way to do it is to consider the dynamics of two types of walkers, positive walkers and negative walkers¹³. In the following, we will therefore start from describing how the number of positive walkers, $n_{+,i}(t)$, and that of negative walkers, $n_{-,i}(t)$, on each node i evolve in time, so as to define different versions of random walks on signed networks. In this section, we give an interpretation of established methods for signed networks based on the signed Laplacian in the language of random walks.

The strong random walk, as we call it, is defined as follows. Positive and negative walkers perform a random walk by taking an edge, independently of its sign. If the walker takes a positive edge, it keeps its own sign; else it changes sign. The rate equation for the process is:

$$\begin{aligned} n_{+,j}(t+1) &= \sum_i (n_{+,i} T_{+,ij} + n_{-,i} T_{-,ij}), \\ n_{-,j}(t+1) &= \sum_i (n_{-,i} T_{+,ij} + n_{+,i} T_{-,ij}), \end{aligned} \quad (5)$$

where

$$\begin{aligned} T_{+,ij} &= A_{+,ij}/k_i, \\ T_{-,ij} &= A_{-,ij}/k_i, \end{aligned} \quad (6)$$

indicate the transition probabilities for a walker from i to j and whether to keep or change the sign respectively. In matrix notation:

$$\begin{aligned} T_+ &= K^{-1} A_+, \\ T_- &= K^{-1} A_-. \end{aligned} \quad (7)$$

If we consider the dynamics of the walkers independently of their sign and consider the time evolution of $n_i(t) = n_{+,i}(t) + n_{-,i}(t)$, one recovers the classical equation for random walk on networks

$$n_j(t+1) = \sum_i n_i(t) T_{u,ij}, \quad (8)$$

where $T_u = K^{-1} A_u$. If, in contrast, we consider the time evolution of the difference $\Delta_i = n_{+,i} - n_{-,i}$, that can be understood as the total “charge” on node i , we find

$$\Delta_j(t+1) = \sum_i \Delta_i(t) T_{s,ij}, \quad (9)$$

where $T_s = K^{-1} A_s$, is the signed transition matrix from which one can build the standard signed Laplacian, whose spectral properties are closely associated to the notion of strong balance and have been used for node embeddings and clustering^{10,12}.

Another interesting perspective is to consider the sets of equations (5) as one single system of equations driven by the matrix

$$T = \begin{pmatrix} T_+ & T_- \\ T_- & T_+ \end{pmatrix}. \quad (10)$$

The latter is simply the (unsigned) transition matrix of a larger graph, composed of $2N$ nodes (each node appears twice, once for positive walkers, and once for negative walkers), with adjacency matrix

$$A = \begin{pmatrix} A_+ & A_- \\ A_- & A_+ \end{pmatrix}. \quad (11)$$

The whole diffusive process can thus be understood in terms of the spectral properties of that transition matrix. The system of equations decouples

when performing the following change of variable: for each i , we define $\Delta_i = n_{+,i} - n_{-,i}$ and $n_i = n_{+,i} + n_{-,i}$, where the determinant of the transformation matrix is 2, leading to the transition matrix

$$T' = \begin{pmatrix} T_s & 0 \\ 0 & T_u \end{pmatrix}, \quad (12)$$

for these variables. The spectral properties of T are therefore determined by those of the two matrices T_s and T_u , with well-known special cases, e.g., when the graph is disconnected or (strongly) structurally balanced¹³.

Walk lengths. The strong walk can be equipped with a teleportation mechanism in order to keep walkers concentrated in the neighborhood of a target node. Given a teleportation probability $(1 - p_s)$, we consider the expected walk length before a teleportation. We know that if $p_s = 1$, the strong walk will have infinity as the expected length, thus we assume $p_s < 1$ here. For each node i , if the walk has length k , it means that teleportation does not happen at the first $(k - 1)$ time steps but the k th time step, thus the number of steps that teleportation does not occur follows a negative binomial distribution with success probability being $1 - p_s$ and the number of success being 1. Hence, the expected walk length is the sum of the expected steps that teleportation does not happen plus 1, i.e.,

$$\mathbb{E}[l|X_0 = i] = \frac{p_s}{1 - p_s} + 1 = \frac{1}{1 - p_s}.$$

Signed network similarity matrix

In the case of unsigned networks, most kernels are based on the principle that similar nodes tend to be connected by several, ideally short walks, which can be captured by the flow of walkers between them. In the case of signed graphs, one needs to distinguish between the type of walks, as some of them will make a pair of nodes more similar, while others will make that pair more dissimilar. Hence, a modelling choice has to be made to define “positive” versus “negative” walks.

Let us start with an intuition based on the notion of strong balance. A graph is said to be strongly balanced if it does not possess any cycle with an odd number of negative edges. Equivalently, this graph can be partitioned into two sets, say A and B , of nodes such that the positive edges are within each cluster, and the negative edges are between the two. This also implies that any walk between two nodes of the same set, say A , possesses an even number of negative edges, while any walk between two nodes of different sets possesses an odd number of negative edges.

Given this extra ingredient, we follow the steps leading us to the similarity matrix inspired by random walks with restart. For each node i , we consider a process where with probability p_s , walkers may either jump to one of its neighbouring nodes, and possibly change sign, or teleport back to the pinned node with probability $1 - p_s$. During the teleportation, the walker automatically becomes positive, independently of its sign. This choice is motivated by the fact that a node should be similar to itself. As before, the process defines for each node i a vector, but it is now a $2N$ vector, where the first N entries encode the density of positive walkers (arriving through walks with an even number of negative edges, or positive walks), while the last N entries encode the density of negative walkers (arriving through negative walks).

The walker begins as a positive walker on some node i , $\mathbf{e}_i$, a $2N$ vector. The state of the walker after k steps, $\mathbf{u}_k$ can be written in terms of the walker after $k - 1$ steps $\mathbf{u}_{k-1}$ as,

$$\mathbf{u}_k = (1 - p_s)\mathbf{e}_i + p_s \mathbf{u}_{k-1} T, \quad (13)$$

according to the dynamics defined by the transition matrix T . The stationary state of this process is defined by

$$\mathbf{n} = (1 - p_s)\mathbf{e}_i (I - p_s T)^{-1}. \quad (14)$$

The first N entries of $\mathbf{n}$ encode the flow of walkers on positive walks from i to the other nodes, hence capturing similarity between the nodes. The last N entries encode the flow of walkers on negative walks from i to the other nodes, hence encoding dissimilarity between the nodes. More precisely, stacking the first N elements of this vector defines the i -th row of a similarity matrix κ_+ . Stacking the last N elements of this vector defines the i -th row of a dissimilarity matrix κ_- . The two matrices can be seen as the positive and negative adjacency matrices of a graph where weights involve longer walks that explore larger area of the network when p_s increases. Taking the differences between the matrices defines a signed similarity matrix, which compares the relative importance of positive and negative walks between nodes. Note that in Eq. (14), we give the same importance to each node, in contrast to the Eq. (3), as there is no straightforward way to make the similarity matrix symmetric by a weighting of the nodes. Other choices could be done, for instance, by weighting the nodes by their degree. In practice, for the experiments where we need to find the entire kernel, we symmetrize the signed similarity matrix after calculating it by adding the resulting matrix to its transpose.

Weak walks

Definition and basic properties. The walk (5) appears to be appropriate to uncover the similarity of two nodes from a strong balance perspective, as walks with an even number of negative edges will increase the similarity between the nodes, while walks with an odd number of negative edges will decrease their similarity. However, this is not the case when searching similar nodes based on the notion of weak balance; see Fig. 1. A graph is said to be weakly balanced if it does not possess cycles with one single negative edge. Weakly balanced graphs are k -clusterable, meaning that they can be partitioned into k factions, where the negative edges are between the factions and the positive edges are within the factions. The condition via the appearance of one single negative edge suggests to define the dissimilarity between two nodes by the existence of walks with one single negative edge between them. Any walks involving more than one negative edge are instead considered uninformative and should not be considered when estimating the similarity between two nodes.

These arguments suggest to modify the strong walk defined by imposing that a walker can become negative only once and that it disappears when taking a negative edge a second time. This argument causes an asymmetry in the node states, as positive and negative walkers now play a different role. The dynamics is encoded by the enlarged transition matrix

$$T_w = \begin{pmatrix} T_+ & T_- \\ 0 & T_+ \end{pmatrix}, \quad (15)$$

It is easy to show that it will keep the same structure, after several iterations, and that:

$$T_w^n = \begin{pmatrix} T_+^n & U_n \\ 0 & T_+^n \end{pmatrix}, \quad (16)$$

where $U_n = \sum_{i=0}^{n-1} \binom{n-1}{i} (T_+)^i (T_-) (T_+)^{n-1-i}$. As defined here, the process leads to a leak of probability, as walkers disappear after taking a negative edge for the second time, leading to a vanishing number of walkers asymptotically (except in trivial cases).

A natural way to deal with this leakage is to adopt a random walk with restart. We assume that a negative walker teleports back to a given target node when taking a negative edge. In contrast with the case of the strong walk, an additional teleportation is not necessary for the walks to remain finite before the restart, unless in trivial situations (e.g., a connected component without a single negative edge). Moreover, the length of the walks before the restart is not controlled by a parameter, but controlled by the structure of the signed network.

Suppose we start with 1 positive walker on node x , given by the initial state vector, $\mathbf{e}_x$, where this is a vector of length $2N$. Then, the random walk with restart would be defined by the matrix:

$$T_x = \begin{pmatrix} T_+ & T_- \\ M_x & T_+ \end{pmatrix}, \quad (17)$$

where M_x is equivalently the zero matrix except column x , which is equivalent to the vector $\mathbf{v}$ where:

$$v_i = \sum_j T_{-;ij}, \quad (18)$$

such that matrix T is now stochastic and has row sum equal to 1. The negative walker will return to node x as a positive walker when it encounters another negative edge. The evolution can be determined by taking the product of the initial state vector $\mathbf{e}_x$ and the matrix T_x .

The walk can also be modified to have an additional teleportation parameter, p_w , where each walker will teleport back to the pinned node with additional probability $1 - p_w$ as a positive walker, in addition to the restart in the case of a negative walker encountering a negative edge. This results in an increase of the teleportation rate as compared to the strong walk case, thus a decrease in the expected walk length, as we will show. In this framework, we can then construct a signed similarity matrix in the very same way as for strong walks. For a random walk starting as a positive one on node x , the state of the weak walker after k steps, $\mathbf{u}_k$ can be written in terms of the walker after $k - 1$ steps $\mathbf{u}_{k-1}$ as,

$$\mathbf{u}_k = (1 - p_w)\mathbf{e}_x + p_w \mathbf{u}_{k-1} T_x, \quad (19)$$

in much the same fashion as in Equation (13). We note that we have primarily considered unweighted networks where the edges all have uniform weight of 1. However, the procedures we have outlined here can be easily generalized to weighted networks, where the weight can be interpreted as biasing the transitions of the walker. We note that although both the strong and weak walks are designed with the intuition provided by strong and weak balance, they are applicable to incompletely balanced graphs as well, which is the case of most real-life datasets. We show in Section VII the results of these walks on graphs which are unbalanced.

Walk length. We now characterise the weak walks from the perspective of expected walk length. We consider a teleportation probability $1 - p_w$. In this case, teleportation is not the only reason for the random walker to stop (or restart), where traversing negative edges for the second time can also trigger it. Hence, we cannot have a network-independent distribution, but need to consider the network structure. Particularly, we consider the enlarged transition matrix (15). We encode the expected number of visits in the matrix $Q \in \mathbb{R}^{2N \times 2N}$, where N dummy nodes are created to separate the cases when the node is visited via paths of zero or one negative edge. Then,

$$Q = \sum_{k=0}^{\infty} p_w^k T_w^k = (I - p_w T_w)^{-1},$$

where I is the identity matrix.

Hence, the expected walk length starting from node i is

$$\mathbb{E}[l|X_0 = i] = \sum_{j=1}^{2N} Q_{ij}.$$

Note that we only count the first N rows in Q . Therefore, the expected walk length for weak walk is

$$\mathbb{E}[l] = \sum_{i=1}^N \mathbb{E}[l|X_0 = i]P(X_0 = i).$$

We verify the theoretical results with Monte Carlo simulations on Signed Stochastic Block Models (SSBMs, see section IV A for details), where edges occur with probability 0.05 and do not flip sign from the initial configuration; see Fig. 2. We consider a uniform initial distribution, repeat the random walks starting from each node for $n_s = 100$ times and obtain the expected walk length up to either teleportation happens or it traverses the second negative edge.

Applications

In this section we present an application of the weak walk method to signed network clustering. We first show the difference between the resulting signed similarity matrices for the weak and strong walks on two small signed network examples, for which ground truth clusters exist. Next, we compare the weak walk against two other methods by performing semi-supervised clustering on a series of synthetic SSBM networks, with varying levels of noise, community numbers, and link densities. Specifically, we compare the weak walk against (1) the strong walk and (2) the Signed Random Walk with Restart (SRWR), a three-parameter signed network walk designed for semi-

supervised clustering which generates a ranking of nodes in signed networks³⁴. The SRWR model considers a random walker that changes its sign as it takes positive or negative edges according to two parameters, as well as a teleportation parameter, and exhibits high accuracy at signed link prediction. The semi-supervised clustering for each random walk is performed in much the same fashion as with PageRank³⁵; we begin with (positive) walkers on a set of labeled seed nodes. We use the similarity vector generated from the weak, strong and SRWR walks to obtain a ranking of each of the other nodes in the graph. Each unclassified node is assigned a label depending on in which walk it ranked the highest, inheriting the label of that seed node. We use the parameter-free version of the weak walk with no teleportation ($p_w = 1$). The strong walk has one parameter to adjust, the teleportation parameter (p_s), while SRWR has three parameters: two parameters which control if the walker changes sign when it takes a negative edge or not, as well as a teleportation parameter. In these experiments, we sweep through the parameter space for the strong walk and the SRWR and take the parameters with the best results. We show that the weak walk outperforms the other two methods in certain cases, despite being parameter-free. To evaluate the accuracy of the clustering assignment obtained in each case we use the Adjusted Rand Index (ARI)³⁶ to compare against the ground truth clusters. These methods can be applied both to completely balanced and incompletely balanced graphs, and Section VII provides a sense of the lower bound of the degree of balance required for these methods to work.

Computational details. In the experiments that follow, we make the choice to construct the signed similarity matrices for both the strong and weak walks by taking the state of the walk vector at $k = n$, where n is large enough that the walk is approaching a stationary state. This can be computationally written as an iterative process allowing us to solve for the similarity matrix as a whole in one computation, which is more efficient than inverting a matrix as required by Equation (14). As matrix multiplication scales well, this reduces the computational complexity of the algorithm. As well, the matrix given in Equation (19) is specific to the starting node x , requiring N matrices to be inverted. See section IV B for a description of the algorithm. Specifically, if the state of the walker (strong or weak) is given by $\mathbf{u}_k$, the similarity between node x and any other node is given by the vector s_x :

$$s_x = u_{k=n}[1 : N] - u_{k=n}[N + 1 : 2N], \quad (20)$$

since we subtract the negative walks from the positive. For the experiments on the empirical networks for which we calculate the entire similarity matrices, the kernel is made symmetric by adding it to its transpose.

Highland Tribes. There are very few empirical signed networks available that are labeled with ground truth cluster assignments. We show here two small examples with labeled data to demonstrate the distinction between the strong and weak walk kernels on these networks. The first network comes from³⁷, and represents the antagonistic or friendly relationships between 16 tribes of the Eastern Central Highlands of New Guinea,

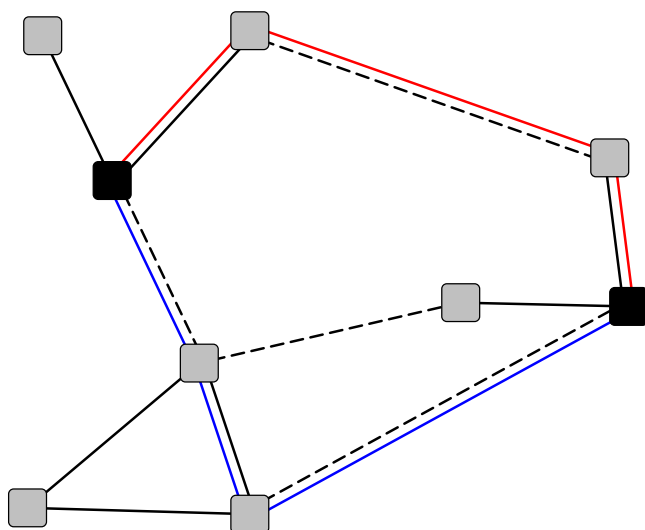


Fig. 1 | When estimating the similarity between two nodes, here colored in black, strong walks may lead to conflictual signals when the network is composed of more than 2 clusters, here illustrated via a positive walk in blue and a negative walk in red (solid: positive; dashed: negative). Weak walks only consider walks with at most one negative edge and thus properly consider the two nodes as dissimilar.

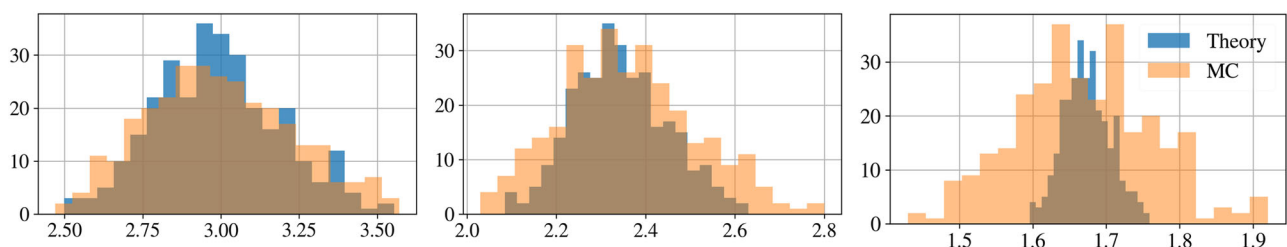


Fig. 2 | Distribution of the expected walk lengths starting from different nodes, both from the theoretical results in blue and Monte-Carlo simulations in orange on 3 community symmetric SSBM with 100 nodes in each community, $p = 0.05$ and

different levels of teleportation (left: $p_w = 1$; middle: $p_w = 0.8$; right: $p_w = 0.5$, where the strong walk with $p_s = 0.8, 0.5$ has expected walk length 5, 2, respectively).

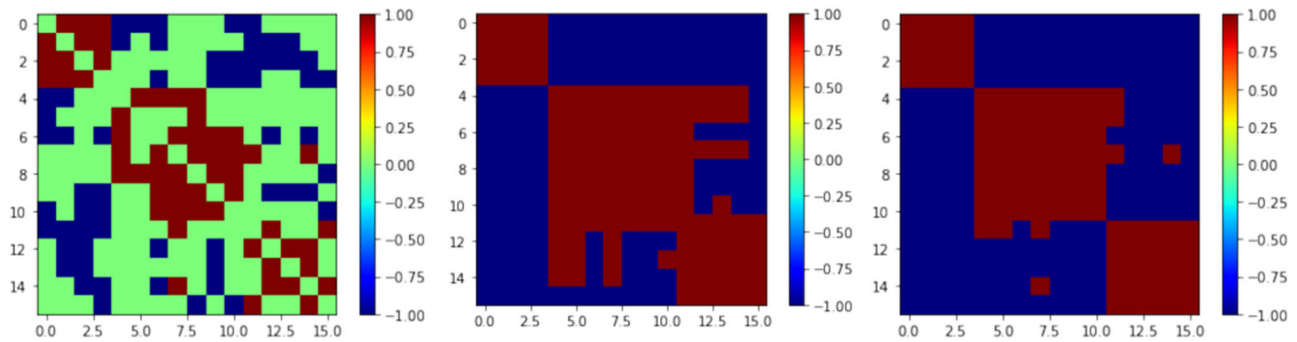


Fig. 3 | (left) Adjacency matrix for the Highland Tribes network. (centre) Strong walk kernel resulting from a 500 step walk on the Highland Tribes network with teleportation parameter $p_s = 0.7$. (right) Weak walk kernel resulting from a 500 step walk with no teleportation $p_w = 1.0$. Positive and negative values have been set to +1 or -1 respectively for clearer visualization.

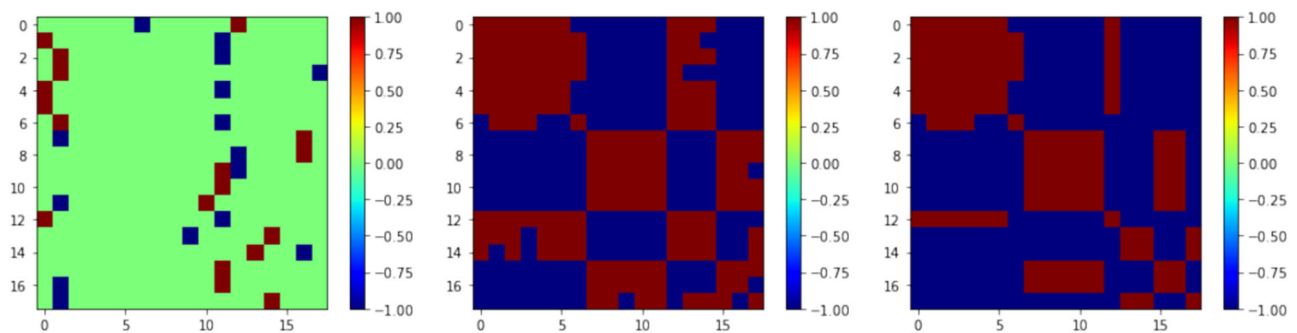


Fig. 4 | (left) Adjacency matrix for the Sampson's Monastery network. (centre) Strong walk kernel resulting from a 500 step walk on the Highland Tribes network with teleportation parameter $p_s = 0.8$. (right) Weak walk kernel resulting from a 500 step walk with no teleportation. Positive and negative values have been set to +1 or -1 respectively for clearer visualization.

represented by edge weights of -1 or $+1$ respectively. The network can be clustered into three communities, which characterise the higher order groupings of the society³⁸. Figure 3 gives the adjacency matrix for the Highland Tribes network, where the nodes have been ordered by their community groupings. This network is not perfectly balanced, but exhibits weak balance, (see³⁹ for a full characterization of the network's local and global balance). As the network has more than two communities, we expect that the weak walk will outperform the strong walk at locating the three communities. The resulting strong and weak walk kernels generated by this network with 500 steps are also given in Fig. 3. The walks themselves produce a similarity matrix which, like the adjacency matrix of the graph, can be used as input into other methods to produce community assignments. For example, adjacency matrix can be combined with other methods such as the spin-glass community detection method for signed networks based on simulated annealing introduced in⁴⁰ to obtain the community assignments in an unsupervised fashion, which can be compared against the ground truth community assignments using the ARI score. As our goal here to test whether the weak walk is able to outperform the strong walk, we take the parameter free version of the weak walk, with no teleportation. We compare against the strong walk's best performance across all possible parameter choices; as such, we choose the best teleportation parameter by inputting the kernel into the spin-glass algorithm, and comparing the community assignments against the ground truth. We take the teleportation parameter that yields the best ARI score, corresponding to $p_s = 0.7$. The strong walk kernel exhibits far more incorrect signs as compared to the weak walk: in particular it exhibits several positive connections between the second and third communities. Like the adjacency matrix, these kernels can be used as input for the spin-glass algorithm to obtain unsupervised

community assignments, allowing us to compare the resulting clustering assignments produced by the three matrices. In this case, the graph is quite small, dense and almost balanced, and so the adjacency matrix, weak walk kernel and strong walk kernel all produce an ARI of 1.0 when used as input into the spin-glass algorithm.

Sampson's monastery. In this section we present another example of a small empirical signed network with ground truth labels. The Sampson's Monastery network comes from an ethnographic study of the community structure within a New England monastery, where each relationship between the 18 novice monks has been designated as positive (friendly) or negative (antagonistic). Sampson, the ethnographer, divided the men into four groups according to the community structure he observed⁴¹. Although this example also represents a small network, the graph is more sparse than the Highland Tribes example, and has more noise in terms of edge signs that violate structural balance (see³⁹ for a full description of the network's structural balance). Figure 4 gives the network adjacency matrix, with the nodes arranged according to the four ground truth communities. Again we expect that since this network has more than two communities, the weak walk will outperform the strong walk at identifying the four communities. The resulting strong and weak walk kernels generated by this network with 500 steps are also given in Fig. 4. Again, we use the parameter-free weak walk with no teleportation, while for the strong walk, we choose the teleportation parameter of $p_s = 0.8$, which achieves the highest ARI score when combined with the spin-glass algorithm. The strong walk kernel exhibits far more incorrect signs as compared to the weak walk, generating several positive connections between the second and third

Fig. 5 | ARI from the semi-supervised clustering for the weak walk, strong walk and SRWR on 2 community SSBMs (strong balance).

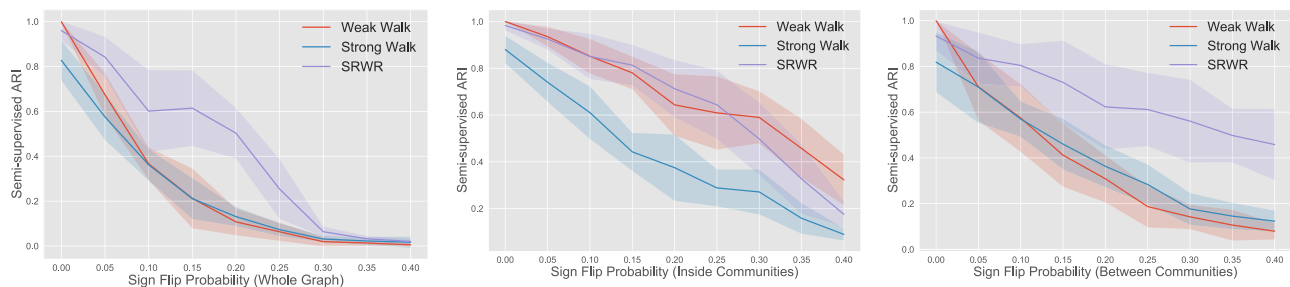
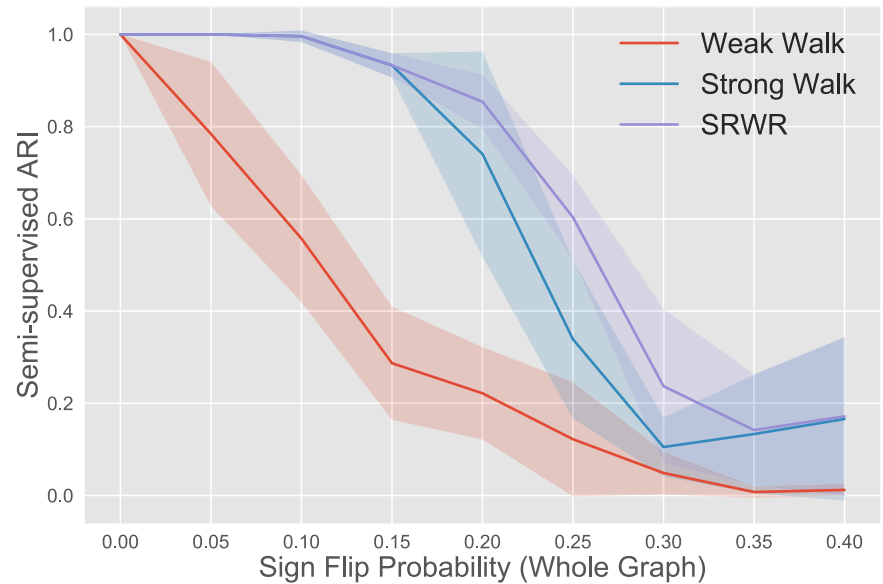


Fig. 6 | ARI from the semi-supervised clustering for the weak walk, strong walk and SRWR on 3 community SSBMs (weak balance). (left) Noise level general to whole graph. (middle) Noise is only inside communities. (right). Noise is only between communities.

communities, and the third and fourth communities. When combined with the community detection spin-glass algorithm, the parameter-free weak walk kernel obtains the best ARI (0.639) compared to the the adjacency matrix (0.442) and the strong walk kernel (0.518).

Application: semi-supervised clustering

In this section, we use the weak walk kernel to perform semi-supervised clustering of several different synthetic SSBMs with community structure. This application is semi-supervised, as we use prior information about the labels of a select group of nodes, and use this information to label the rest of the graph. We compare against the kernel generated by the strong walk, and the SRWR. In these experiments we take the parameter-free version of the weak walk, with no teleportation, while we sweep through the parameter space for the strong walk and the SRWR and take the parameters which achieve the highest ARI. Consequently, these methods perform better than the weak walk in some cases, but it may be less useful in practice, since the ideal parameters are not known a priori and indeed tend to change with each iteration of the SSBM, even when the SSBM parameters are held constant.

Strong balance: two-community SSBMs. Here, we look at 2 community SSBMs with 100 nodes in each community, and 0.05 probability of link existence between each pair of nodes. We perform the semi-supervised clustering with varying levels of noise on the signs of the links, via the sign flip parameter v (see section IV A for a description of the noise parameter). For the strong and weak walks, we take the state vector at iteration $k = 100$, and set for SRWR the max iterations = 100. The other parameters for the SRWR and strong walk are chosen by parameters

sweep and change for each iteration of the graph. We take the mean and standard deviation over 10 graph realizations, and the results are given in Fig. 5. We observe that the SRWR performs the best, a feature to be balanced with the fact that it has the most parameters. The strong walk exhibits improvement over the weak walk; this is expected, as the weak walk is designed to outperform the strong walk in the case when there are more than two communities, and the strong walk has the advantage of teleportation.

Weak balance: three-community SSBMs. Here, we look at 3 community SSBMs with 100 nodes in each community, and 0.05 probability of link existence between each pair of nodes. We perform the semi-supervised clustering with varying levels of noise on the signs of the links, as in the previous example. For the strong and weak walks, we take the state vector at iteration $k = 100$, and set for SRWR the max iterations = 100. As before, the other parameters are chosen by parameters sweep and change for each iteration of the graph. We take the mean and standard deviation over 10 graph realizations, the results are given in Fig. 6 (left). We observe that the SRWR again outperforms the other two methods. The strong walk and weak walk exhibit similar performance, despite the strong walk having one extra parameter. In different empirical networks, the noise may occur along positive or negative edges, with different probabilities. Consequently, we investigate the results when the noise on the signs of the edges is restricted to either inside the communities by varying v_{inside} and setting $v_{\text{between}} = 0$, or between communities by varying v_{between} and setting $v_{\text{inside}} = 0$ (see section IV A for details). The weak walk is much more robust to noise inside communities (Fig. 6 (centre)), and indeed performs almost as well as the SRWR in this case. When the noise

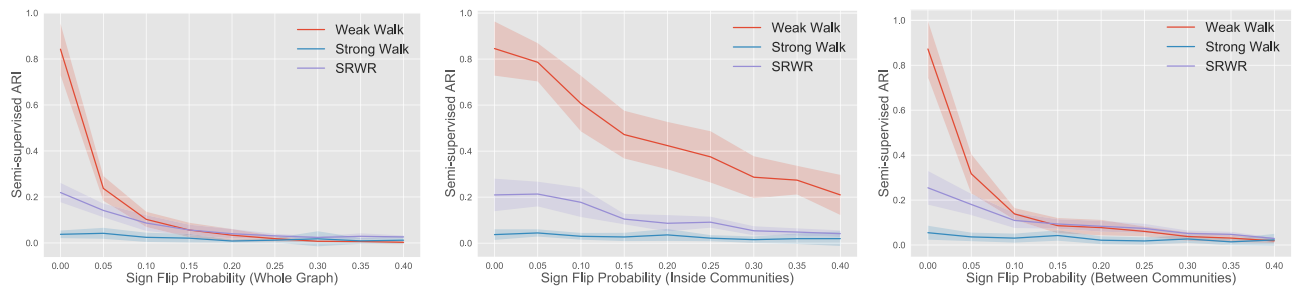


Fig. 7 | ARI from the semi-supervised clustering for the weak walk, strong walk and SRWR on 6 community SSBMs (weak balance). (left) Noise level general to whole graph. (middle) Noise is only inside communities. (right). Noise is only between communities.

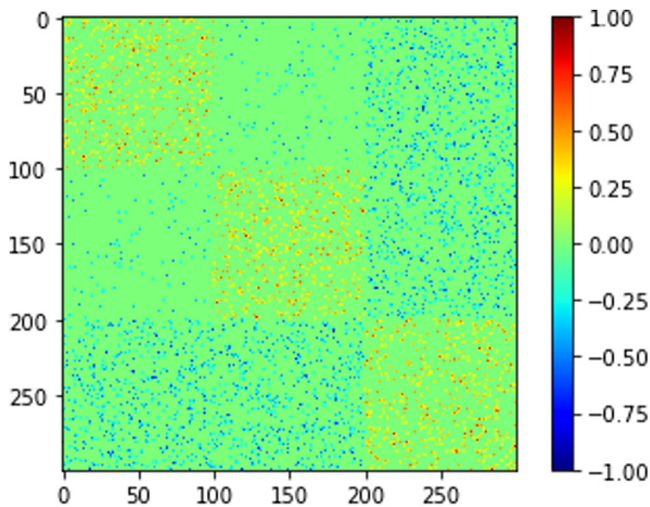


Fig. 8 | Adjacency matrix of 300 node 3-community asymmetric SSBM, with asymmetry parameter $a = 0.2$.

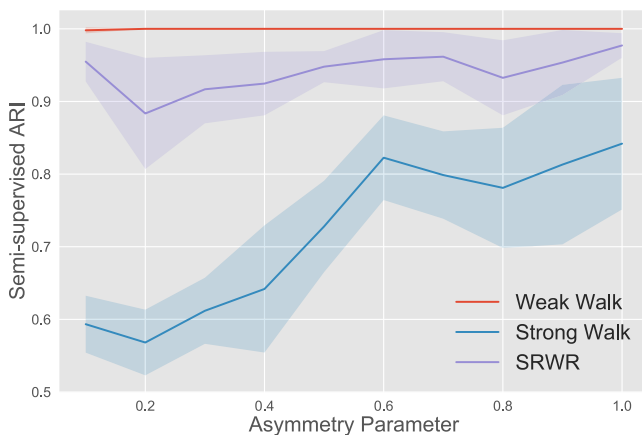


Fig. 9 | ARI from the semi-supervised clustering for the weak walk, strong walk and SRWR on 3 community SSBMs (weak balance), with varying asymmetry.

is between the communities (Fig. 6 (right)), the weak walk is less robust and the SRWR outperforms both the strong and weak walks.

Weak balance: six-community SSBMs. Here, we perform the same experiments but with a 6 community SSBMs with 50 nodes in each community, and 0.05 probability of link existence between each pair of nodes. As the number of communities increase, the problem of clustering signed networks becomes more difficult. We again perform the semi-supervised clustering with varying levels of noise on the signs of the links. For the strong and weak walks, we take the state vector at iteration

$k = 300$, and set for SRWR the max iterations = 300. The other parameters are chosen by parameters sweep and change for each iteration of the graph. We take the mean and standard deviation over 10 iterations, the results are given in Fig. 7 (left). In this case, we observe that the weak walk performs much better than the strong walk and the SRWR, although the performance decreases quickly as the noise level increases. As in the previous example with 3 communities, we investigate the results when the noise on the signs of the edges is restricted to either inside the communities as in Fig. 7 (centre) or between the communities as in Fig. 7 (right). The weak walk is indeed much more robust to noise inside communities, compared to when the noise is between the communities, outperforming both the strong walk and the SRWR by a larger margin. The distinction between the methods is even more clear than in the previous case of 3 communities. The weak walk, despite having no parameters, outperforms the other methods when the weak balance structure is more clear.

Asymmetry. In most of the previous literature on clustering signed networks, the density of links between each of the communities is held constant. In practice, this may not be the case; different pairs of communities may be connected by different link densities. This would produce, what we call here, an asymmetric SSBM. We define an asymmetry parameter a which controls the link density between communities by defining the link probability matrix as $[[0.05, 0.05a, 0.05], [0.05a, 0.05, 0.05], [0.05, 0.05, 0.05]]$ for a 3 community asymmetric SSBM. One realization of the adjacency matrix for a 3-community asymmetric SSBM with 100 nodes in each community and $a = 0.2$ is given in Fig. 8.

We perform the semi-supervised clustering on 300 node 3-community SSBMs with varying levels of asymmetry. For the strong and weak walks, we take the state vector at iteration $k = 100$, and set for SRWR the max iterations = 100. The results are given in Fig. 9. In every cases, the weak walk that we have introduced here outperforms the strong walk and SRWR. This is because walks between the first two communities can happen in two ways, either by taking a negative edge between the two, or by taking two negative edges by passing through the third community. This second walk cannot happen in the case of the weak walk, which cannot take two negative edges. When the asymmetry is high enough, when a is close to 0, the latter case is more likely, and introduces noise into the strong walk, which does not occur in the case of the weak walk.

Discussion

Random walks play a central role in the study of networks, and are at the core of several algorithms, e.g. for community detection and network embedding. Random walks allow us to explore the indirect patterns of connectivity between nodes, and their spectral properties are directly related to important network patterns, such as the cut size. The specific rules of a random walk process are a non-trivial choice, however, and it is now recognised that tuning the parameters of the model, e.g. to favour network exploration in depth versus in width, may lead to significant improvements, as was demonstrated by node2vec⁸ for instance. More generally, the choice of a random walk model should be driven by the type of structure that one

aims to isolate and/or by the nature of the diffusive processes taking place on an empirical data¹².

This paper contributes at the problem of designing a random walk model, which is “optimal” within the family of random walks over signed networks. We argue that the random walk process associated to the standard matrices for the mining of signed networks, e.g. the signed Laplacian, is inherently (and implicitly) biased to the search of a strong balance structure. In the case when the system exhibits multiple clusters, associated to weak balance, we argue that another random walk model, called the weak walk, is more compatible with the network structure.

After introducing the model and exploring some of its properties, we have considered a classical random-walk based measure of node similarity, in three versions based on the strong walk, the weak walk and the so-called SRWR walk respectively. Through a series of experiments on artificial and real-life data, including unsupervised and semi-supervised clustering, we have shown that the results of weak walks outperform those of strong walks when the graph has more than two communities, or exhibits asymmetry in the density of links. We have also shown that it is competitive against the three-parameter SRWR walk in many cases as well, despite being parameter-free.

A wide range of algorithms for network mining take, as an input, trajectories generated by a random walk model. For the design of a similarity matrix, we have made a specific, yet classical, choice for how to combine the multiple walks that may connect a pair of nodes, but other choices could have been made. For example, the similarity matrix could be defined by summing over the walks at each step, by taking the state vector after k steps, taking the stationary state of the walk as in PageRank³⁵, or by combining the results from walks of different lengths in a non-linear fashion to give greater weight to longer or shorter walks as in the similarity kernel introduced in⁷. Future research perspectives include feeding weak walk trajectories instead of strong walks ones into other algorithms and exploring applications to other typical signed network problems, such as unsupervised clustering¹⁰ and link prediction⁴³. As a specific example, the strong walk has been used to learn stances for users in signed social networks⁴⁴. Combining weak walks with these state-of-the-art methods could yield, at a minimal developing cost, an improvement in performance and flexibility for empirical networks, especially when the assumption of strong balance does not hold. Another challenge for future research is the identification of the “right” number of factions when confronted with real-world data, in order to guide the user to methods building on strong or weak walks in practice.

Methods

Signed stochastic block model

We perform most of our experiments on a signed stochastic block model (SSBM), which is a random graph model with a cluster structure that either exhibits strong or weak balance. The SSBM has N nodes, separated into k clusters. The edge probabilities between each pair of nodes are independent, and are defined by a $k \times k$ probability matrix P , where P_{ij} gives the probability that an edge exists between a node in cluster i and another node in cluster j . In the initial configuration, edges connecting nodes in the same cluster are positive, while edges connecting nodes in different clusters are negative. As in¹⁸, unlike in the unsigned case, the edge existence probabilities do not need to be different for inter- and intra-community edges, as the structure of the network is determined by the signs. We introduce a noise parameter called sign flip probability $\nu \in (0, 1)$, which gives the probability that the sign of each edge is flipped from the correct sign (positive inside communities and negative between communities). In section VI, we also introduce parameters $\nu_{\text{inside}} \in (0, 1)$ and $\nu_{\text{between}} \in (0, 1)$ which separately control the level of noise on the signs of the edges inside and between communities, respectively.

Strong/weak walk experiments: iterative design

We construct the signed similarity matrix by taking the final state vector at $k = n$, where n is large and the state is approaching the stationary state. We compute the state vector at each step with an iterative process. The way the

algorithm defines the similarity matrix is as follows. Beginning with a positive walker on node x :

$$\mathbf{u}_0 = \mathbf{e}_x. \quad (21)$$

For the strong walk, at each step the new vector $\mathbf{u}_k$ is computed from $\mathbf{u}_{k-1}$ as:

$$\mathbf{u}_k = (1 - p_s)\mathbf{e}_x + p_s\mathbf{u}_{k-1}T, \quad (22)$$

where p_s is the strong walk teleportation parameter. For the weak walk, the process is defined as in Equation (19), although it is computationally easier to use the non-Markovian matrix T_w to compute $\mathbf{u}_k$ from $\mathbf{u}_{k-1}$ as:

$$\mathbf{u}_k = (1 - (p_w\mathbf{u}_kT_w) \cdot \mathbf{1})\mathbf{e}_x + p_w\mathbf{u}_kT_w \quad (23)$$

where p_w is the weak walk teleportation parameter, $\mathbf{1}$ is the all-one vector, thus $(p_w\mathbf{u}_kT_w) \cdot \mathbf{1}$ sums over the probability that has already been allocated to the corresponding neighbours and the first term re-sets the lost probability to the original node x as a positive walker. The advantage of using an iterative process over the matrix given in Equation (17) is that this can be done for several initial states at the same time, while the matrix in Equation (17) is specific to the state where the walker begins, i.e. node x . However, the processes are equivalent since the term M_x in Equation (17) returns all probability leakage back to node x as a positive walker, exactly as the first term in Equation (23). Finally, for both the strong and weak walks we define the similarity between node x and any other node is given by the vector $\mathbf{s}_x$:

$$\mathbf{s}_x = \mathbf{u}_{k=n}[1 : N] - \mathbf{u}_{k=n}[N + 1 : 2N], \quad (24)$$

since we subtract the negative walks from the positive.

Note that in the experiments we perform, we sample over all possible choices for p_s , the strong walk teleportation parameter and select the best one for each iteration of the graph. However, we keep p_w fixed at 1, such that the weak walk teleportation rate is determined entirely by the structure of the graph, to keep the weak walk parameter free.

Data availability

Most of the experiments are run on synthetic signed networks, which can be generated from the code provided. The data for the Sampson’s monastery network is available here: (<http://vlado.fmf.uni-lj.si/pub/networks/data/esna/sampson.htm>). The data for the Highland Tribes network is available here: (<http://konect.cc/networks/ucidata-gama/>).

Code availability

The code for this study is available at: (<https://github.com/saynbul/weak-walk>).

Received: 12 June 2024; Accepted: 30 December 2024;

Published online: 03 February 2025

References

- Masuda, N., Porter, M. & Lambiotte, R. Revealing the structure of complex networks with random walks. *Phys. Rep.* **716–717**, 1 (2017).
- Lambiotte, R. & Schaub, M. T. *Modularity and dynamics on complex networks*, Cambridge University Press, (2021).
- Levin, D. A. & Peres, Y. *Markov chains and mixing times*, vol. 107, American Mathematical Soc. (2017).
- Gleich, D. F. PageRank beyond the web. *SIAM Rev.* **57**, 321 (2015).
- Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl Acad. Sci.* **105**, 1118 (2008).
- Lambiotte, R., Delvenne, J.-C. & Barahona, M. Random walks, Markov processes, and diffusion on networks. *IEEE Trans. Netw. Sci. Eng.* **1**, 76 (2014).

7. Fouss, F., Saerens, M. & Shimbo, M. *Algorithms and models for network data and link analysis*, Cambridge University Press, (2016).
8. Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864 (2016).
9. Altafini, C. Dynamics of networks with multilayer interactions. *PLoS ONE* **7**, e38135 (2012).
10. Traag, V., Doreian, P. & Mrvar, A. Partitioning of networks with modularity and blockmodeling. *Advances in Network Clustering and Blockmodeling*, pp. 225, (2019).
11. Szell, M., Lambiotte, R. & Thurner, S. Multirelational networks: structure and dynamics. *Proc. Natl Acad. Sci.* **107**, 13636 (2010).
12. Kunegis, J. et al. Spectral graph clustering and embedding. in *Proceedings of the 2010 SIAM International Conference on Data Mining*, SIAM, pp. 559–570 (2010).
13. Tian, Y. & Lambiotte, R. Spreading processes on networks with nontrivial topologies. *SIAM J. Appl. Dyn. Syst.* **23**, 50 (2024).
14. Cheeger, J. A lower bound for the smallest eigenvalue of the Laplacian. *Probl. Anal.* **625**, 110 (1970).
15. Harary, F. The notion of centrality in social networks. *Mich. Math. J.* **2**, 143 (1953).
16. Davis, J. A. Clustering and the structure of social networks. *Hum. Relat.* **20**, 181 (1967).
17. Aref, S. & Neal, Z. P. Identifying communities in social networks. *Sci. Rep.* **11**, 19939 (2021).
18. Cucuringu, M., Davies, P., Glielmo, A. & Tyagi, H. Sponge model for network analysis. In *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 1088–1098 (2019).
19. Babul, S. & Lambiotte, R. Sheep dynamics in networks. *Commun. Phys.* **7**, 8 (2024).
20. Perozzi, B., Al-Rfou, R. & Skiena, S. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710 (2014).
21. Kondor, R. I. & Lafferty, J. Diffusion on graphs and its applications. in *International Conference on Machine Learning*, pp. 315–322 (2002).
22. Lawler, G. *Random Walk and the Heat Equation*, American Mathematical Society, (2010).
23. Smola, A. & Kondor, R. Heat kernel-based learning. in *Learning Theory and Kernel Machines*, B. Schölkopf and M. Warmuth, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 144–158 (2003).
24. Fouss, F., Pirotte, A., Renders, J. & Saerens, M. Commute times and spectral properties of the graph Laplacian. *IEEE Trans. Knowl. Data Eng.* **19**, 355 (2007).
25. Saerens, M., Fouss, F., Yen, L. & Dupont, P. Commute times and their use in machine learning. in *Machine Learning: ECML 2004*, Boulicaut, J., Esposito, F., Giannotti, F. and Pedreschi, D. Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 371–383 (2004).
26. Pan, J., Yang, H., Faloutsos, C. & Duygulu, P. Random walk with restart: a family of models for link prediction in large-scale graphs. in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, pp. 653–658 (2004).
27. Tong, H., Faloutsos, C. & Koren, Y. Random walk with restart: an efficient algorithm for link prediction in large-scale graphs. in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, pp. 747–756 (2007).
28. Tong, H., Faloutsos, C. & Pan, J.-Y. Random walk with restart and its applications to link prediction. *Knowl. Inf. Syst.* **14**, 327 (2008).
29. Qiu, J. et al. Network embedding as matrix factorization. in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467 (2018).
30. Girvan, M. & Newman, M. Community structure in social and biological networks. *Proc. Natl Acad. Sci.* **99**, 7821 (2002).
31. Delvenne, J., Yaliraki, S. & Barahona, M. Markov processes and their applications to network science. *Proc. Natl Acad. Sci.* **107**, 12755 (2010).
32. Lambiotte, R., Delvenne, J. & Barahona, M. Random walks and their applications to network science. *IEEE Trans. Netw. Sci. Eng.* **1**, 76 (2014).
33. Schaub, M., Delvenne, J., Yaliraki, S. & Barahona, M. The role of random walks in network science. *PLoS ONE* **7**, 1 (2012).
34. Jung, J., Jin, W., Sael, L. & Kang, U. Scalable and efficient algorithms for link prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 973–978, <https://doi.org/10.1109/ICDM.2016.0122> (2016).
35. Langville, A. N. & Meyer, C. D. *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, (2006).
36. Gates, A. J. & Ahn, Y.-Y. The impact of machine learning on link prediction. *J. Mach. Learn. Res.* **18**, 1 (2017).
37. Read, K. E. Cultures in conflict. *Southwest. J. Anthropol.* **1**, (1954).
38. Hage, P. & Harary, F. Structural models of social networks. *No Title* (1983).
39. Aref, S., Dinh, L., Rezapour, R. & Diesner, J. Multilevel analysis of networks. *Sci. Rep.* **10**, 15228 (2020).
40. Traag, V. A. & Bruggeman, J. Community detection in networks. *Phys. Rev. E* **80**, 036115 (2009).
41. Sampson, S. F. *A Novitiate in a Period of Change: an Experimental and Case Study of Social Relationships*. Cornell University, (1968).
42. Lambiotte, R., Rosvall, M. & Scholtes, I. Networks: structure and dynamics. *Nat. Phys.* **15**, 313 (2019).
43. Liu, S. Y., Xiao, J. & Xu, X. K. Link prediction in complex networks. *IEEE Trans. Netw. Sci. Eng.* **7**, 1724 (2019).
44. Pougué-Biyong, J., Gupta, A., Haghighi, A. & El-Kishky, A. Learning representations for large-scale networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 177–185 (2023).

Acknowledgements

We would like to thank Zoe Schwerkolt for her participation in the early stages of the project, and John Pougué-Biyong for his insightful comments. R.L. acknowledges support from the EPSRC grants EP/V013068/1, EP/V03474X/1 and EP/Y028872/1. S.A.B. was supported by EPSRC grant EP/W523781/1. This work was supported by The Alan Turing Institute's Enrichment Scheme. Y.T. is funded by the Wallenberg Initiative on Networks and Quantum Information (WINQ).

Author contributions

All authors conceived the idea, performed the research, and wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S. A. Babul, Y. Tian or R. Lambiotte.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

© The Author(s) 2025