# LEM

# Working Paper Series

## Exact maximum-likelihood method to detect patterns in real networks

Tiziano Squartini [§]
Diego Garlaschelli [o]

[§] Department of Physics, University of Siena, Italy
[˙] LEM-Scuola Superiore Sant'Anna, Pisa, Italy

2011/07                    February 2011

# Exact maximum-likelihood method to detect patterns in real networks

Tiziano Squartini

*Dipartimento di Fisica, Università di Siena, Via Roma 56, 53100 Siena (Italy)*

Diego Garlaschelli

*LEM, Sant'Anna School of Advanced Studies, P.zza Martiri della Libertà 33, 56127 Pisa (Italy) and*
*Instituut-Lorentz for Theoretical Physics, Leiden Institute of Physics,*
*University of Leiden, Niels Bohrweg 2, 2333 CA Leiden (The Netherlands)*

(Dated: February 27, 2011)

In order to detect patterns in real networks, randomized graph ensembles that preserve only part of the topology of an observed network are systematically used as fundamental null models. However, their generation is still problematic. The existing approaches are either computationally demanding and beyond analytic control, or analytically accessible but highly approximate. Here we propose a solution to this long-standing problem by introducing an exact and fast method that allows to obtain expectation values and standard deviations of any topological property analytically, for any binary, weighted, directed or undirected network. Remarkably, the time required to obtain the expectation value of any property is as short as that required to compute the same property on the single original network. Our method reveals that the null behavior of various correlation properties is different from what previously believed, and highly sensitive to the particular network considered. Moreover, our approach shows that important structural properties (such as the modularity used in community detection problems) are currently based on incorrect expressions, and provides the exact quantities that should replace them.

PACS numbers: Valid PACS appear here

## I. INTRODUCTION

Detecting relevant patterns in real networks, a fundamental problem for many research fields [1–3], relies upon the possibility to distinguish the properties explained by the presence of simple constraints from more complex and nontrivial structural features. For this reason, statistical ensembles of graphs with specified constraints, and otherwise completely random, have been introduced and systematically used as a reference to identify non-random patterns in a real network [4–23]. Such ensembles serve also as powerful models to study dynamical processes on networks displaying only a set of desired properties, and allow to highlight the dynamical effect of each property separately. The simplest and most important ensembles specify only local constraints. For unweighted networks, this amounts to specify the degree $k_i$ (number of incident edges) of each vertex ($i = 1 \ldots N$ where $N$ is the total number of vertices), and results in the so-called configuration model [4, 5, 7]. In the weighted case, the corresponding constraint is obtained by fixing the strength $s_i$ (sum of incident edge weights) of each vertex [15–17]. More in general, one could enforce different or additional properties [6, 11, 13, 14, 16, 19–23].

Unfortunately, as we discuss in detail in what follows, it turns out that even in the simplest case with local constraints, the correct generation of random ensembles corresponding to a particular real world network is problematic. Both analytical and computational approaches proposed so far have severe limitations. Motivated by this, here we propose a new maximum-entropy method that is entirely analytical and does not require the generation of randomized variants of a real network. Our method provides the exact probabilities of occurrence of random graphs with the same constraints as the real network, from which the expectation values and standard deviations (as well as all other moments) of any topological quantity of interest can be calculated mathematically. Due to its analytical character, our method is extremely faster than all available alternatives. Moreover, it can be applied to undirected, directed, binary and weighted networks in a unified way. We will illustrate the power of our approach on several real-world networks of different nature and type, by studying a range of topological properties of interest.

## II. AVAILABLE METHODS AND THEIR LIMITATIONS

We first briefly review the existing problems in the case of binary unweighted networks, which is the most frequently explored situation. A binary unweighted graph with $N$ vertices is completely specified by a $N \times N$ adjacency matrix $\mathbf{A}$ with entries $a_{ij} = 1$ if the vertices $i$ and $j$ are connected, and $a_{ij} = 0$ otherwise. Generally, one is interested in comparing the observed topological properties of a particular real-world network $\mathbf{A}^*$ against the average properties of a randomized family of networks with the same degree sequence $\vec{k}(\mathbf{A}^*) = \{k_i(\mathbf{A}^*)\}$, where $k_i(\mathbf{A}^*) = \sum_j a_{ij}^*$ is the degree (number of connections) of vertex $i$ in the network $\mathbf{A}^*$. The ensemble of binary undirected networks with specified degree sequence is known as the *configuration model* (CM) [4, 5, 7] and
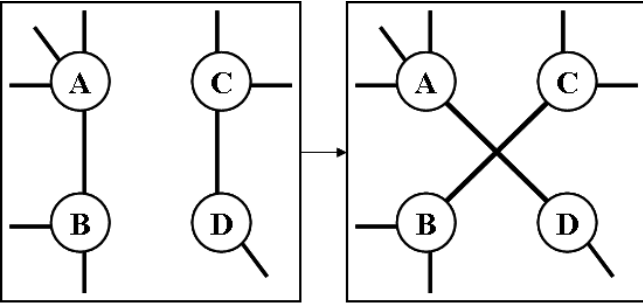
FIG. 1. An illustration of the local rewiring algorithm whose iteration allows to computationally explore the microcanonical configuration model.

is currently treated in two very different ways: computationally, by explicitly generating many random networks with the desired degree sequence and averaging the quantities of interest across the randomized networks [4, 5], or analytically, by using approximations that allow to directly estimate the average of topological properties as a function of the enforced degree sequence, without actually measuring them on any network [7, 8]. Currently, both approaches suffer from severe limitations.

A 'bottom-up' computational approach consists in assigning each vertex $i$ a number of 'edge stubs' equal to its observed degree $k_i(\mathbf{A}^*)$, and randomly matching pairs of stubs (avoiding self-loops and multiple links) until all degrees reach their desired values (*edge stub reconnection*). However, this procedure is known to get stuck in configurations where vertices requiring additional connections have no more eligible partners [4, 5]. As a consequence, one must implement a 'top-down' computational approach where the entire real network $\mathbf{A}^*$ is taken as the initial configuration, and a family of randomized variants is generated by iteratively applying a *local rewiring algorithm* (LRA) where two edges $(A, B)$ and $(C, D)$ are randomly selected and replaced by the two edges $(A, D)$ and $(C, B)$, if the latter are both not already present [4, 5] (see fig.1 for an illustration). This generates a *microcanonical* ensemble (see the Appendix for a detailed discussion) where all randomized networks have exactly the same degree sequence as the original network, and are sampled with equal probability. This method has been applied to the Internet [5], cellular networks [6] and food webs [12] to detect higher-order patterns (such as clustering and motifs) not merely due to local constraints. However, this approach is time-consuming since many (an assumed number $R \geq 4L$ where $L$ is the observed number of links) iterations of the LRA are required to obtain a single randomized variant, and the entire process must be repeated several times to produce a large number $M$ of randomized variants, on each of which any topological property $X$ of interest must be measured explicitly and averaged at the end to obtain an estimate for $\langle X \rangle$. The computational time required to obtain $\langle X \rangle$ is therefore $M(T_R R + T_X)$, where $T_R$ is the average time required to perform a single successful rewiring step and $T_X$ is that required to compute $X$ on a single network in the randomized set. Moreover, even if the sufficient statistics of the problem is the degree sequence $\vec{k}(\mathbf{A}^*)$ alone, the above approach requires the entire original network $\mathbf{A}^*$ as the starting configuration, thus making use of much more information than required in principle.

By contrast, analytical approaches seek to provide theoretical expressions to directly obtain the ensemble averages of topological properties, without generating the ensemble computationally. Two main approaches exist. One makes use of generating functions for the relevant probability distributions. In the case we are discussing here, the key quantity is the generating function $g(z) = \sum_k z^k P(k)$ of the degree distribution [7]. Unfortunately, this method assumes that the network is infinite and locally tree-like (even if in some cases this approximation turns out to perform unexpectedly well even beyond its formal range of applicability [24]), and is thus inappropriate if the size of the network is small and if the input degree distribution can only be realized by dense and/or clustered networks. In this approach, clustered or dense networks can only be generated by imposing additional constraints besides the degree sequence, such as the number of triangles attached to vertices [25], thus leading to a different ensemble which is not the one we are seeking to characterize. Moreover, theoretical predictions are obtained assuming an explicit functional form for the degree distribution $P(k)$. Therefore, they cannot be used for networks displaying a noisy, complicated or not easily parametrizable degree distribution.

A different approach looks for an analytical expression for the probability $p_{ij}$ that the vertices $i$ and $j$ are connected in the randomized ensemble [8]. Due to its probabilistic nature, this approach generates a *(grand)canonical* ensemble where even graphs violating the constraint are present and assigned different probabilities. In such a case, the constraints are realized on average, i.e. the expectation value $\langle X \rangle$ of any specified property $X$ is fixed exactly (see Appendix). While this approach is indeed very fast in providing averages of the desired properties, it has been shown [9] that it makes use of a highly approximate expression for $p_{ij}$, valid only when the original network is sparse and/or the degree distribution is not too broad. This expression is

$$p_{ij} = \frac{k_i(\mathbf{A}^*)k_j(\mathbf{A}^*)}{2L^*} \qquad (1)$$

where $L^* \equiv L(\mathbf{A}^*) = \sum_i k_i(\mathbf{A}^*)/2 = \sum_{i<j} a_{ij}^*$ is the total number of links. While the expected degree $\langle k_i \rangle = \sum_j p_{ij}$ generated by the above formula coincides with the desired degree $k_i(\mathbf{A}^*)$, the probability $p_{ij}$ may exceed 1 for pairs of highly connected nodes such that $k_i(\mathbf{A}^*)k_j(\mathbf{A}^*) > 2L(\mathbf{A}^*)$. In general, only if the degree sequence is such that

$$k_i(\mathbf{A}^*) < \sqrt{2L(\mathbf{A}^*)} = \sqrt{\sum_j k_j(\mathbf{A}^*)} \quad \forall i \qquad (2)$$

then using eq.(1) on the real network $\mathbf{A}^*$ will not lead to the above problem. While the above condition is typically obeyed by networks with narrow degree distribution such as the Erdős-Rényi random graph, it is generally violated by scale-free networks displaying a power-law degree distribution $P(k) \sim k^{-\gamma}$, and this violation becomes stronger and stronger as the density of the network increases. In particular, it is possible to show that in order to ensure eq.(2) the maximum degree $k_{max}$ in the network should not exceed the so-called *structural cut-off* $k_c \sim N^{1/2}$ [26]. This is particularly evident for dense networks where the average degree $\bar{k} = \sum_i k_i/N = 2L/N$ remains constant as $N$ increases, so that eq.(2) remains valid only if $k_{max} < \sqrt{2L} \sim N^{1/2}$. By contrast, extreme value theory shows that in networks with degree distribution $P(k) \sim k^{-\gamma}$ the maximum degree scales as $k_{max} \sim N^{1/(\gamma-1)}$, so that if $\gamma < 3$ (as observed in most real-world scale-free networks) then $k_{max} > N^{1/2}$ which exceeds $k_c$. The meaning of $p_{ij}$ being larger than 1 for some pairs of vertices in eq.(1) is that, in order to actually realize the degree sequence of the real network $\mathbf{A}^*$, one must let $i$ and $j$ be connected by more than one undirected edge. Also, since the desired equality $\langle k_i \rangle = k_i(\mathbf{A}^*)$ is only ensured if one lets the sum in $\sum_j p_{ij} = \langle k_i \rangle$ run over all vertices *including $i$ itself*, one must allow the presence of self-loops in the randomized networks. Thus, even if this is not evident at a first glance, the ensemble generated by eq.(1) does not only contain binary and loop-less undirected graphs and is thus not a proper null model for an empirical binary loop-less network $\mathbf{A}^*$ with degree sequence $\vec{k}(\mathbf{A}^*)$ violating eq.(2), as is typically the case for real-world networks with broad degree distributions. An elegant proof that the correct ensemble probability $p_{ij}$ for loop-less graphs with no multiple connections differs from eq.(1) has been proposed [9] and re-derived within the framework of maximum-entropy graph ensembles [14]. We shall exploit this result to obtain an exact method later on. We will also show that in real networks the deviation is stronger than expected, and affects sparse networks as well. An independent proof of the inadequacy of eq.(1) is that it does not generate the graph $\mathbf{A}^*$ with maximum likelihood [27]. This can be confirmed by treating $L$ as a free parameter and look for its value $L^{ML}$ that maximizes the probability to obtain $\mathbf{A}^*$. One finds that $L^{ML} \neq L(\mathbf{A}^*)$, which implies that under the maximum likelihood choice $\langle k_i \rangle \neq k_i(\mathbf{A}^*)$ and $\langle L \rangle \neq L(\mathbf{A}^*)$, violating the desired constraint on the degree sequence and the implied one on the number of links [27]. This shows that the functional form of $p_{ij}$ in eq.(1) is intrinsically problematic and does not give highest likelihood to $\mathbf{A}^*$ and to all other graphs with the same degree sequence as $\mathbf{A}^*$.

Therefore, while the available analytical methods are useful to characterise artificially generated networks with special properties, they cannot be used to correctly randomise any real-world network which is either small, clustered, or dense. Unfortunately, the above limitations are generally ignored, and eq.(1) is frequently used beyond its limits of applicability to estimate connection probabilities. Moreover, as we note later on, it is also used as a key ingredient in order to define important structural properties which implicitly rely on a comparison against the CM. Analogous problems exist in the analysis of directed and/or weighted networks. We will consider each of these cases separately in what follows.

## III. AN EXACT ANALYTICAL METHOD

The above discussion highlights that no method developed so far succeeds in obtaining randomized properties of a particular real-world network such that two requests are met simultaneously: *i)* the method is general and works for any network, even if displaying small size, high link density, and large clustering; *ii)* expected values across the ensemble can be computed analytically, without sampling the configuration space explicitly. The need to resort to the LRA as the only statistically correct method available, which however requires the artificial generation of many randomized networks, makes the general problem very complicated and all its applications time consuming.

In this paper we propose a solution to this long-standing problem. We develop an approach that combines exact expressions for the occurrence probabilities of graphs in maximum-entropy ensembles with given constraints [9, 11, 14, 21–23] with more recent results obtained applying the Maximum Likelihood principle to graph ensembles [27]. In the Appendix we describe our method in great detail. We start with a general discussion which is formally valid for any constraint, and then consider explicitly the application to real networks where a set of local constraints is enforced. We consider the cases of binary, weighted, directed and undirected networks separately. We show that in all these cases the enforcement of local constraints always leads to exact results that can be easily obtained analytically. Then we also consider an extension to non-local constraints which can still be dealt with analytically. Finally, we compare our (grand)canonical method with the corresponding microcanonical ensemble generated computationally as in the LRA. As we show, for any case of interest a choice of constraints leads to a specific set of coupled nonlinear equations to be solved. In such equations, the observed values of the enforced topological properties (e.g. the degree sequence) determine the values of an equal number of 'hidden' parameters in such a way that the real network, or any other network with the same constraints as the real one, is generated with maximum likelihood. Since only the enforced constraints enter the equations, our method only requires the knowledge of the sufficient statistics of the problem and not of the whole topology, restoring a desirable feature of randomization algorithms. Solving the maximum-likelihood equations only takes a computational time $T_E$ ranging from seconds to tens of

seconds, even on an ordinary laptop, and entirely replaces the artificial generation of many randomized variants of the original network. Once the parameters solving the equations are found, they can be directly used to obtain the expectation value $\langle X \rangle$ and standard deviation $\sigma[X]$ of any topological property $X$ of interest analytically. When useful, this also allows to obtain a $z$-score representing the number of standard deviations by which the randomized value $\langle X \rangle$ differs form the observed value $X(\mathbf{A}^*)$. The possibility to obtain the standard deviations and/or $z$-scores is very important, because it allows to assess which topological properties $X$ are consistent with their randomized value $\langle X \rangle$ within a statistical error, and which deviate significantly from the null expectation. In the former case, one can conclude that the enforced constraints completely explain the higher-order property $X$. In the latter case, the observed property cannot be traced back to the constraints, and therefore requires additional explanations or generating mechanisms besides those required in order to explain the constraints themselves. Importantly, the time required to compute the expectation value $\langle X \rangle$ of a given property $X$ exactly (formally corresponding to an average over a huge number of randomized configurations) is the same as the time $T_X$ required to compute the same property on the single original network. Therefore our method takes only a total time $T_E + T_X$ to obtain $\langle X \rangle$ exactly, which is incredibly shorter than the aforementioned time $M(T_R R + T_X)$ required by the LRA to obtain $\langle X \rangle$ approximately. Importantly, $T_E$ is independent of the complexity of the topological property $X$ to measure, which means that for complicated properties $T_E + T_X \approx T_X$. Therefore for any topological property $X$ which can be measured in a large but still reasonable time $T_X$ on the real network, the computation of its expectation value $\langle X \rangle$ will require the same time $T_X$. If the time required in order to obtain $\langle X \rangle$ is too large, it is because the time required to measure $X$ is too large as well. In other words, the property $X$ is too complicated to be computed on the real network itself. In such a case, the problem is not due to the method, but to a demanding choice of $X$ for that particular network.

## IV. RESULTS

We now show the application of our method to real networks of various type, by considering several topological properties and their randomized counterparts.

### A. Binary undirected networks

We start with the simplest case of binary undirected networks. One of the most important topological properties of a binary network is the correlation between the degrees of adjacent nodes, which has been shown to dramatically affect various structural and dynamical features [2].

These correlations can be measured by the average nearest neighbour degree (ANND), which on the real network $\mathbf{A}^*$ is defined as

$$k_i^{nn}(\mathbf{A}^*) \equiv \frac{\sum_{j \neq i} \sum_{k \neq j} a_{ij}^* a_{jk}^*}{\sum_{j \neq i} a_{ij}^*} \qquad (3)$$

While the degree is a first-order property which only depends on the number of links (topological paths of length one) entering a vertex, the ANND is a second-order property contributed by paths of length 2 (i.e. the terms $a_{ij}^* a_{jk}^*$). Similarly, a third-order (i.e. involving paths of length 3) property is the clustering coefficient $c_i$, which represents the fraction of pairs of neighbours of vertex $i$ which are mutually connected:

$$c_i(\mathbf{A}^*) \equiv \frac{\sum_{j \neq i} \sum_{k \neq i,j} a_{ij}^* a_{jk}^* a_{ki}^*}{\sum_{j \neq i} \sum_{k \neq i,j} a_{ij}^* a_{ki}^*} \qquad (4)$$

As we mentioned, it is always important to assess whether in a particular real network higher-order properties arise merely as a consequence of low-level constraints or whether they signal additional structural patterns. In particular, comparing the real network $\mathbf{A}^*$ with the CM (which provides an ensemble of random networks having, on average, the same degree sequence $\vec{k}(\mathbf{A}^*)$ as $\mathbf{A}^*$) allows to assess whether longer topological paths and the structural properties involving them are simply a random concatenation of the individual links enforced by the degree sequence, or whether they are irreducible to first-order constraints. As we discuss in detail in the Appendix, our method can solve this problem exactly by making use of an auxiliary $N$-dimensional vector $\vec{x} = \{x_1 \ldots x_N\}$ of parameters. In particular, one must look for the particular values $\vec{x}^*$ that solve the following set of $N$ coupled nonlinear equations:

$$\sum_{j \neq i} \frac{x_i^* x_j^*}{1 + x_i^* x_j^*} = k_i(\mathbf{A}^*) \qquad \forall i \qquad (5)$$

where $k_i(\mathbf{A}^*)$ is the observed degree of vertex $i$ in the real network $\mathbf{A}^*$. Once the parameter values are found, they allow to obtain analytically the expectation value $\langle X \rangle^*$ of any topological property $X$ across the desired ensemble. This simply amounts to replace the adjacency matrix entry $a_{ij}^*$ appearing in the definition of $X(\mathbf{A}^*)$ with its expectation value

$$p_{ij}^* = \langle a_{ij} \rangle^* = \frac{x_i^* x_j^*}{1 + x_i^* x_j^*} \qquad (6)$$

which represents the correct expression that should be used in place of eq.(1). Similarly, it is possible to obtain the standard deviation $\sigma^*[X]$ analytically in terms of $\vec{x}^*$ (see Appendix).

In fig.2 we show an application of our method on the network of the 500 largest US airports [28], a synaptic network [29], two protein interaction networks [30],
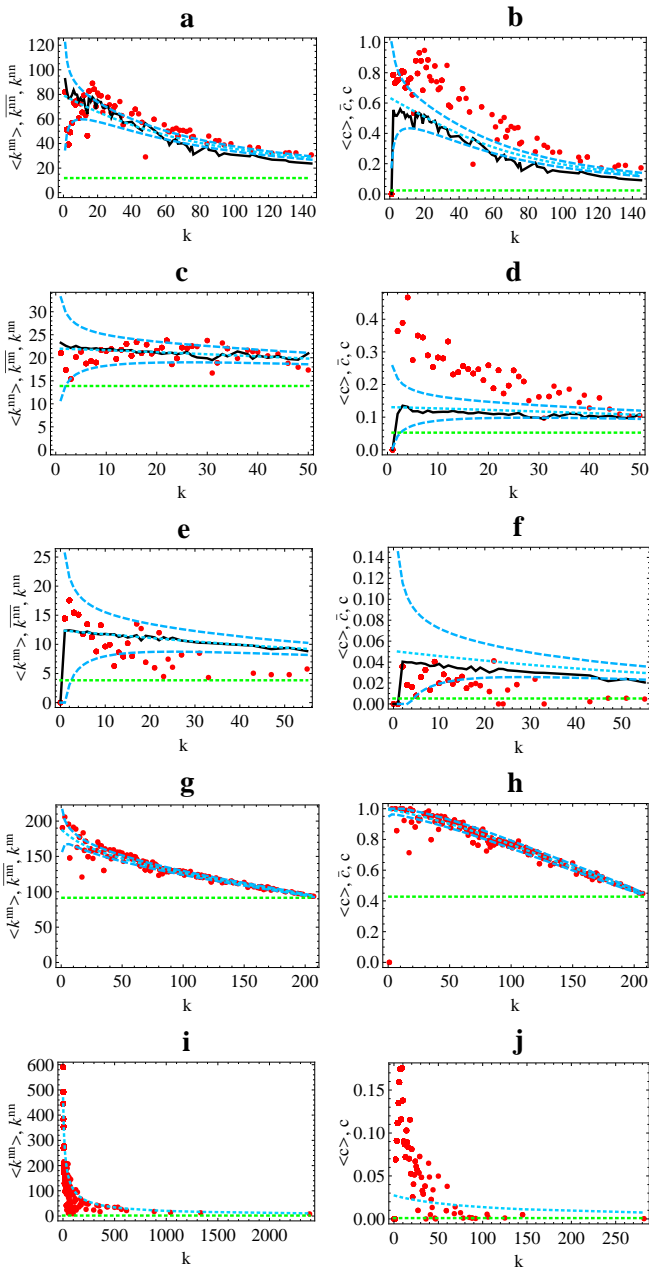
FIG. 2. Application of our method to binary undirected networks. The red points are the empirical data, the black solid curves are averages over the configuration model obtained using the local rewiring algorithm [4, 5], and the blue dashed curves are the analytical expectations (± one standard deviation) obtained using our method. The green curves are the flat expectations under the random graph model, and highlight the average level of correlation in the case when there is no dependence on the degree. The panels report $k_i^{nn}$ versus $k_i$ (left) and $c_i$ versus $k_i$ (right) for: a) and b) the network of the largest US airports ($N = 500$) [28], c) and d) the synaptic network of *Caenorhabditis elegans* ($N = 264$) [29], e) and f) the protein-protein interaction network of *Helicobacter pylori* ($N = 732$) [30], g) and h) the network of liquidity reserves exchanges between Italian banks in 1999 [31] ($N = 215$), i) the Internet at the AS level ($N = 11.174$) [32] and j) the protein-protein interaction network of *Saccharomices cerevisiae* ($N = 4.142$) [30]. The last two networks are randomized using only our method, as the local rewiring algorithm would require much more time given the large number of edges.

an interbank network [31] and the Internet at the Autonomous Systems level [32]. These are among the most studied networks of this type. We compare the correlation structure of the original networks, as measured by the dependence of $k_i^{nn}(\mathbf{A}^*)$ and $c_i(\mathbf{A}^*)$ on $k_i(\mathbf{A}^*)$, with the expected values $\langle k_i^{nn} \rangle^*$ and $\langle c_i \rangle^*$ obtained analytically using our method. We also highlight the region within one standard deviation around the average by plotting the curves $\langle k_i^{nn} \rangle^* \pm \sigma^*[k_i^{nn}]$ and $\langle c_i \rangle^* \pm \sigma^*[c_i]$. For the sake of comparison, we also report the average values obtained sampling the microcanonical ensemble with the standard local rewiring algorithm [4, 5], and the expected values over the ensemble of random graphs with the same number of links (*random graph model*, RG) As we mentioned, the microcanonical method requires the generation of many randomized variants, many rewirings per variant, and the measurement of $k_i^{nn}$ and $c_i$ on each variant separately, plus a final averaging. By contrast, our method only requires the preliminary estimation of the $\{x_i^*\}$. Then the calculation of $\langle k_i^{nn} \rangle$ and $\langle c_i \rangle$ takes exactly the same time as that of the empirical values. As can be seen, the two approaches yield very similar results (in the Appendix we provide a detailed comparison of the two methods). For the two largest networks (the protein interactions in *S. cerevisiae* and the Internet), we only report the expectations obtained using our method, as the microcanonical approach would require too much computing time.

The above results allow to interpret the effect of the degree sequence on higher-order properties. Firstly, the trends displayed by the CM are not flat as those expected in the random graph case. This confirms that residual structural correlations, simply due to the enforced constraint, are still present after the rewiring has taken place. The presence of these correlations does not require any additional explanation besides the existence of the constraints themselves. This is very different from the picture one would get by using the (wrong) expectation of eq.(1) which would yield flat trends as well, naively suggesting that correlations can never be traced back to the degree sequence alone. Secondly, while the trends observed in all the networks considered are always decreasing, they unveil different correlation patterns when compared to the randomized trends. The real interbank data are almost indistinguishable from the randomized curves, meaning that structural constraints can fully explain the observed behaviour of higher-order network properties. Instead, in the airport network the randomized curves lie below the real data (except for an opposite trend of $\langle k_i^{nn} \rangle$ for low degrees). This means that the real network is more correlated than the baseline randomized expectation, and indicates that additional mechanisms producing positive correlations must be present on top of structural effects. By contrast, in the *H. pylori*'s protein network the expected curves lie above the real data, suggesting the presence of mechanisms producing negative correlations. The same is true for the correlation structure of the Internet, confirming previous results [5],

while *S. cerevisiae*'s protein network is completely different from its randomized variants. Therefore seemingly similar trends can actually reveal very different types of structural organization. This means that measuring the topological properties alone is uninformative, and makes the comparison between real data and randomized ensembles essential. Thus the possibility to analytically and quickly characterize the latter, which was previously unavailable, is a remarkable advantage of our approach.

## B. Directed networks

We now consider binary directed networks, which are specified by an asymmetric adjacency matrix $\mathbf{A}$. The local constraints are now represented by the joint sequence of out-degrees and in-degrees $\{k_i^{out}, k_i^{in}\} = \{\sum_{j\neq i} a_{ij}, \sum_{j\neq i} a_{ij}\}$. Given a particular real network $\mathbf{A}^*$ and a measured topological property $X(\mathbf{A}^*)$, our method allows to analytically obtain the expectation value $\langle X \rangle^*$ and standard deviation $\sigma^*[X]$ across the ensemble of binary directed graphs with, on average, the same directed degree sequences $\vec{k}^{out}(\mathbf{A}^*)$ and $\vec{k}^{in}(\mathbf{A}^*)$ as $\mathbf{A}^*$ (*directed configuration model*, DCM). As shown in the Appendix, in this case our method makes use of two $N$-dimensional vectors $\vec{x}$, $\vec{y}$ of auxiliary variables, and requires that these parameters are set to the particular values $\vec{x}^*$, $\vec{y}^*$ that solve the following set of $2N$ coupled nonlinear equations:

$$\sum_{j\neq i} \frac{x_i^* y_j^*}{1 + x_i^* y_j^*} = k_i^{out}(\mathbf{A}^*) \qquad \forall i \qquad (7)$$

$$\sum_{j\neq i} \frac{x_j^* y_i^*}{1 + x_j^* y_i^*} = k_i^{in}(\mathbf{A}^*) \qquad \forall i \qquad (8)$$

The quantities $\vec{x}^*$, $\vec{y}^*$ allow to obtain $\langle X \rangle^*$ and $\sigma^*[X]$ analytically and quickly, outperforming the directed version of the LRA [46]. Note that, as in the undirected case, the method only makes use of the sufficient statistics of the problem.

We apply our method to various directed networks, by studying the second-order topological properties measured by the outward ANND and the inward ANND, which are defined as two natural generalizations of eq.(3):

$$k_i^{nn,out}(\mathbf{A}^*) \equiv \frac{\sum_{j\neq i} \sum_{k\neq j} a_{ij}^* a_{jk}^*}{\sum_{j\neq i} a_{ij}^*} \qquad (9)$$

$$k_i^{nn,in}(\mathbf{A}^*) \equiv \frac{\sum_{j\neq i} \sum_{k\neq j} a_{ji}^* a_{kj}^*}{\sum_{j\neq i} a_{ji}^*} \qquad (10)$$

In fig.3 we plot the observed values $k_i^{nn,in}(\mathbf{A}^*)$ versus $k_i^{in}(\mathbf{A}^*)$ and $k_i^{nn,out}(\mathbf{A}^*)$ versus $k_i^{out}(\mathbf{A}^*)$, as well as the expectations $\langle k_i^{nn,in} \rangle^* \pm \sigma^*[k_i^{nn,in}]$ and $\langle k_i^{nn,out} \rangle^* \pm \sigma^*[k_i^{nn,out}]$ obtained using our model (see Appendix), for three real directed networks: the neural network of *C. elegans* [29] (now in its directed version), the metabolic network of *E. coli* [33], and the *Little Rock Lake* food
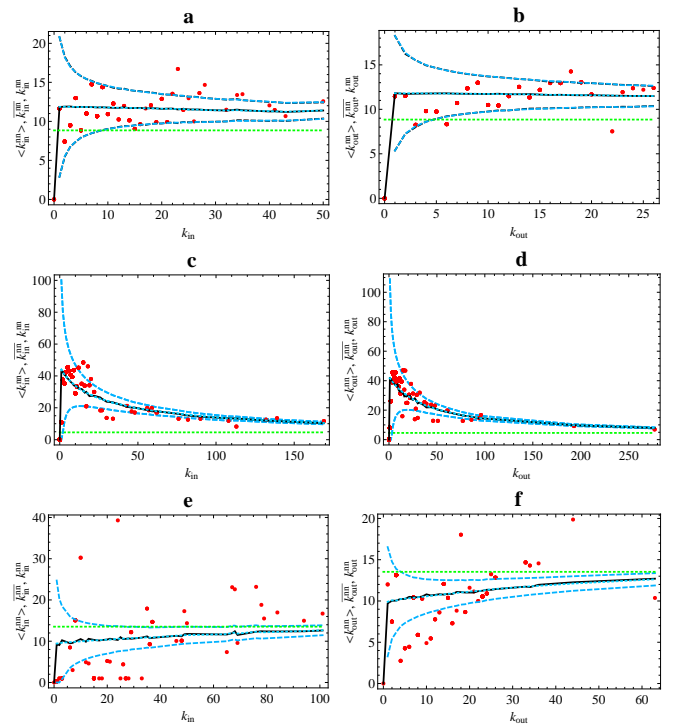


FIG. 3. Application of our method to directed networks. Red points are the empirical data, the black solid curves are expectations under the directed configuration model using the local rewiring algorithm, and the blue dashed curves are the exact expectations obtained using our method (± one standard deviation). The green curves are the flat expectations under the directed random graph model. The panels report $k_i^{nn,in}$ versus $k_i^{in}$ (left) and $k_i^{nn,out}$ versus $k_i^{out}$ (right) for: a) and b) the directed neural network of *Caenorhabditis elegans* ($N = 264$) [29], c) and d) the metabolic network of *Escherichia coli* ($N = 1078$) [33], e) and f) the *Little Rock Lake* food web ($N = 183$) [34].

web [34]. As before, we also show the microcanonical average obtained using the LRA and the expectation under the *directed random graph model* (DRG) with the same number of links. Again, we find a very good agreement between the two approaches, confirming that our method yields the correct prediction in incredibly shorter time (see Appendix for a discussion about the convergence time of the LRA to our exact results). We also confirm that while some networks (*C. elegans* and *E. coli*) are almost consistent with the null model, others (*Little Rock*) deviate significantly.

However, the most interesting point for the present analysis is that, while for the undirected networks considered above all randomized trends were decreasing, in this case we find that the three randomized trends behave in totally different ways. In the neural network, both $\langle k_i^{nn,in} \rangle^*$ and $\langle k_i^{nn,out} \rangle^*$ are approximately constant. This means that the baseline behavior for both quantities is flat and uncorrelated (as in the directed random graph, but at a different level). By contrast,

in the metabolic network the expected curves are decreasing, and thus the ensemble of randomized networks is disassortative as for the undirected graphs considered above. Finally, in the food web the constraints enforce unusual positive correlations, and the randomized ensemble is even assortative. Interestingly, while it is expected that random networks with specified degrees display a disassortative behavior [5, 9], the assortative trend is totally surprising. This is because our method extracts the hidden variables directly from the specific real world network, rather than drawing them from ad hoc distributions. The resulting values can be distributed in a very complicated fashion, invalidating the results obtained under other hypotheses. To further highlight this important point, we selected three more food webs characterized by a particularly small size (see fig.4). Small networks cannot be described by approximating the mass probability function of their topological properties (such as the degree) with a continuous probability density. Therefore in this case the difference between the expectations obtained by drawing the $\vec{x}$ and $\vec{y}$ values from analytically tractable continuous distributions and those obtained by solving eqs.(8) using the empirical degrees is particularly evident. As we show in fig.4 (where for simplicity we omit the comparison with the LRA), we confirm that the (directed) CM can display not only flat or decreasing trends, but also increasing ones. Importantly, in this case all three webs do not deviate dramatically from the null model. This means that while one would be tempted to interpret the three observed trends as signatures of different patterns (zero, negative and positive correlation), actually in all three cases the observed behavior can be roughly replicated by the same mechanism and almost entirely traced back to the degree sequence only. This unexpected result highlights once again that the measured values of any topological property are *per se* entirely uninformative, and can only be interpreted in relation to a null model.

### C. Reciprocity and motifs

So far, in our analysis of directed networks we have considered second-order topological properties. In principle, third-order properties can be studied by introducing directed generalizations of the clustering coefficient [36, 37]. However, there is a proliferation of possible third-order patterns due to the directionality of links. For this reason, a more complete analysis consists in counting (across the entire network) all the possible directed *motifs* [6] involving three vertices, and comparing the empirical abundances with the expected ones under the null model. As we show in a moment, our method lends itself admirably in such a case. Before presenting our results, we note however that directionality makes the possible specifications of the null model proliferate as well. In particular, besides the DCM considered above, a more refined way to randomize directed networks includes the
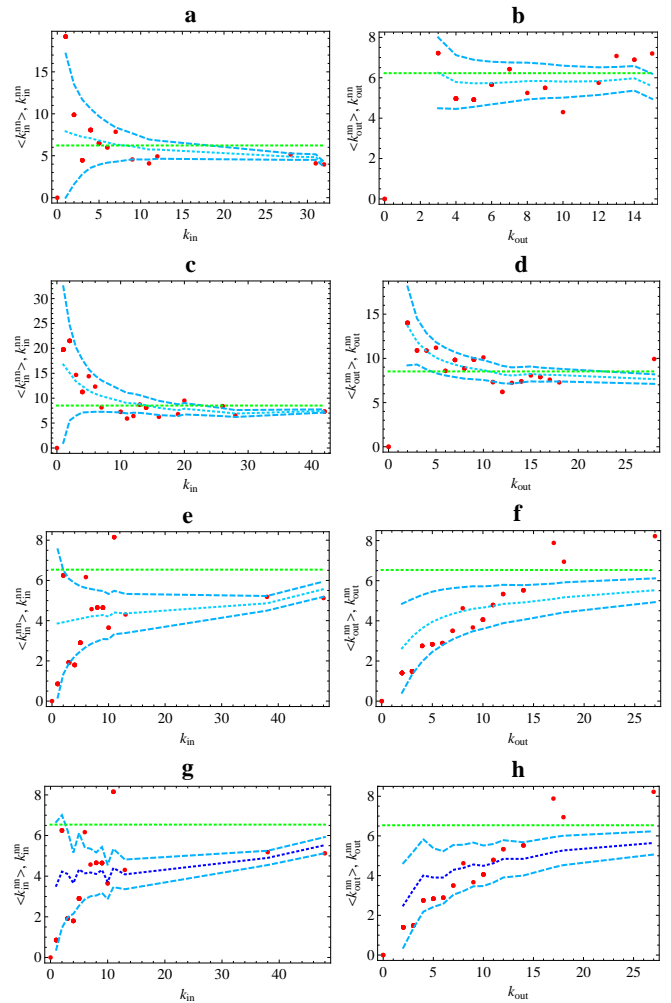


FIG. 4. Application of our method to small-size directed food webs. Red points are the empirical data and the blue dashed curves are the exact expectations ($\pm$ one standard deviation) under the directed configuration model obtained using our method. The green curves are the flat expectations under the directed random graph model. The panels report $k_i^{nn,in}$ versus $k_i^{in}$ (left) and $k_i^{nn,out}$ versus $k_i^{out}$ (right) for: a) and b) the Narragansett Bay web ($N = 35$) [35], c) and d) the Mondego Estuary web ($N = 46$) [35], e) and f) the St. Marks River web ($N = 54$) [35]. For the latter, in g) and h) we also compare the empirical data with the expectations under the reciprocal configuration model, where also the number of reciprocated links of each vertex is specified.

possibility to enforce additional constraints on the reciprocity structure [6, 11]. In other words, it is possible (and important in many applications [6, 12]) to preserve not only the total numbers $k_i^{in}$ and $k_i^{out}$ of incoming and outgoing links of each vertex, but also the number $k_i^{\leftrightarrow} \equiv \sum_j a_{ij} a_{ji}$ of reciprocated links (pairs of links in both directions) [38, 39]. This specification is equivalent to enforce, for each vertex $i$, the three quantities [11, 38] $k_i^{\rightarrow} \equiv \sum_{j \neq i} a_{ij}^{\rightarrow}$ (number of non-reciprocated outgoing

links), $k_i^\leftarrow \equiv \sum_{j \neq i} a_{ij}^\leftarrow$ (number of non-reciprocated incoming links) and $k_i^\leftrightarrow \equiv \sum_{j \neq i} a_{ij}^\leftrightarrow$ (number of reciprocated links), where $a_{ij}^\rightarrow \equiv a_{ij}(1 - a_{ji})$, $a_{ij}^\leftarrow \equiv a_{ji}(1 - a_{ij})$ and $a_{ij}^\leftrightarrow \equiv a_{ij} a_{ji}$.

Given a real directed network $\mathbf{A}^*$, we denote the null model with specified joint reciprocal degree sequences $\{k_i^\rightarrow(\mathbf{A}^*), k_i^\leftarrow(\mathbf{A}^*), k_i^\leftrightarrow(\mathbf{A}^*)\}$ as the *reciprocal configuration model* (RCM). This is an example of model with nonlocal (second-order) constraints which can still be treated analytically using our method. As we show in the Appendix, in this case one must solve the following $3N$ coupled equations:

$$\sum_{j \neq i} \frac{x_i^* y_j^*}{1 + x_i^* y_j^* + x_j^* y_i^* + z_i^* z_j^*} = k_i^\rightarrow(\mathbf{A}^*) \qquad \forall i \quad (11)$$

$$\sum_{j \neq i} \frac{x_j^* y_i^*}{1 + x_i^* y_j^* + x_j^* y_i^* + z_i^* z_j^*} = k_i^\leftarrow(\mathbf{A}^*) \qquad \forall i \quad (12)$$

$$\sum_{j \neq i} \frac{z_i^* z_j^*}{1 + x_i^* y_j^* + x_j^* y_i^* + z_i^* z_j^*} = k_i^\leftrightarrow(\mathbf{A}^*) \qquad \forall i \quad (13)$$

The expectation value of any topological property, as well as its standard deviation, can now be calculated analytically in terms of the three $N$-dimensional vectors $\vec{x}^*$, $\vec{y}^*$, $\vec{z}^*$. For instance, in fig.4g-h we repeat the analysis of the directed ANND of the St. Marks River food web, now comparing the observed trend against the RCM. In this case, we find no significant difference with respect to the DCM considered above (fig.4e-f). However, as we now show, the analysis of motifs reveals a dramatic difference between the predictions of the two null models.

If $N_m$ denotes the number of occurrences of a particular motif $m$, our method allows to calculate the expected number $\langle N_m \rangle^*$ and standard deviation $\sigma^*[N_m]$ exactly (see Appendix), and thus to obtain the $z$-score

$$z[N_m] \equiv \frac{N_m(\mathbf{A}^*) - \langle N_m \rangle^*}{\sigma^*[N_m]} \qquad (14)$$

analytically. This can be done for both the DCM and the RCM. The value of $z[N_m]$ indicates by how many standard deviations the observed and expected numbers of occurrences of motif $m$ differ. Large values of $z[N_m]$ indicate motifs that are either over- or under-represented under the particular null model considered, and that are therefore not explained by the lower-order constraints enforced. In fig.5 we show the $z$-scores for all the possible 13 non-isomorphic connected motifs with three vertices in 8 real food webs, for both null models. We also show the two lines $z = \pm 2$ to highlight the region within 2 standard deviations from the model's expectations. This analysis is similar to that of ref.[12], but is made much simpler by our method which does not require to randomize the webs through a computational algorithm preserving the (reciprocal) degree sequences. The food webs considered here are from different ecosystems (lagoons, marshes, lakes, bays, estuaries, grasses), with a prevalence of aquatic habitats. The presence of (intrinsically
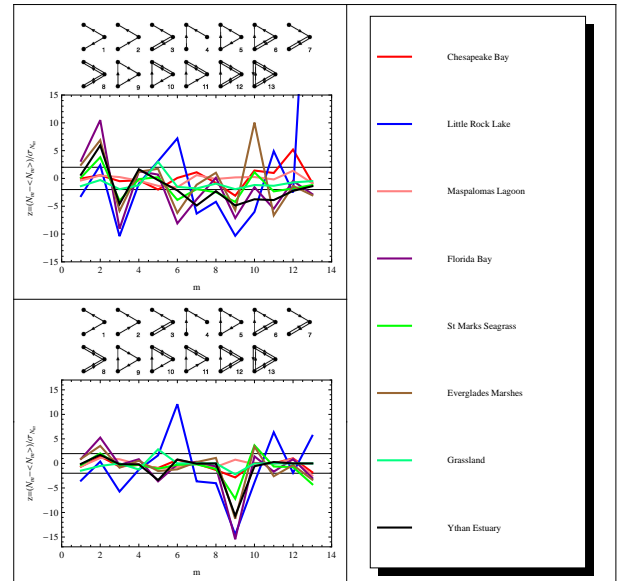


FIG. 5. Application of our method to the analysis of directed *motifs* involving three vertices in 8 real food webs. Top panel: $z$-scores obtained enforcing only the in-degree and out-degree sequences (directed configuration model). Bottom panel: $z$-scores obtained enforcing also the reciprocal degree sequence (reciprocal configuration model).

directed) predator-prey relationships implies that reciprocity is a very important quantity in food webs [12]. Thus the RCM should fluctuate less than the DCM. Indeed, this is confirmed by our analysis. The $z$-scores for the motifs $m = 2, 3, 13$ are significantly reduced from the DCM to the RCM. Also, while the motifs $m = 1, 6, 10, 11$ display large values of $z$ with opposite signs across different webs under the DCM, the signs of all statistically surprising motifs (i.e. when $|z| \gtrsim 2$) become consistent with each other under the RCM (except for $m = 13$). As a consequence, under the RCM all networks display a very similar pattern, and the most striking features of real webs become the over-representation of motifs $m = 2, 10$ (plus $m = 6, 11, 13$ for the Little Rock Lake web) and the under-representation of motifs $m = 5, 9, 13$ (plus $m = 3, 7, 8$ for Little Rock Lake). In particular, the under-representation of motif $m = 9$ (the 3-loop) is the most common pattern across all webs, and becomes stronger as the reciprocity of the web increases. Also note that in a network with no reciprocated links, the number of motifs with at least a pair of reciprocated links

is zero. Under the RCM, the expected number of these motifs remains zero. By contrast, their expected number under the DCM is always positive. Thus we confirm that the upgrade to the RCM is necessary, as its stricter constraints allow to analyze 3-vertices motifs once 2-vertices motifs (i.e. all possible dyadic patterns) are correctly accounted for. The possibility to treat the RCM analytically using our method is therefore an important step forward.

### D. Weighted networks

Remarkably, our method works equally well for weighted graphs (where the binary adjacency matrix $\mathbf{A}$ is replaced by a non-negative weight matrix $\mathbf{W}$), thanks to recent analytical results that allow to characterize maximally random weighted networks with specified properties in a way that is completely analogous to their binary counterparts [22, 23]. In a particular weighted network $\mathbf{W}^*$, the local constraints are the strength sequence $\{s_i(\mathbf{W}^*)\} = \{\sum_j w_{ij}^*\}$ (undirected case) or the joint out-strength and in-strength sequence $\{s_i^{out}(\mathbf{W}^*), s_i^{in}(\mathbf{W}^*)\} = \{\sum_j w_{ij}^*, \sum_j w_{ji}^*\}$ (directed case). We will only consider undirected weighted networks. The extension to the directed case is straightforward. The family of randomized weighted graphs with the same strength sequence as a real weighted network is sometimes denoted as the *weighted configuration model* (WCM) [15]. The available microcanonical algorithms regard each link weight as an integer multiple $w$ of a fundamental unit of weight, transform each edge of weight $w$ into $w$ edges of unit weight, and rewire the latter as in the unweighted case, now ensuring that the strength (total number of incoming edges of unit weight) of each vertex is preserved. This means replacing a list of $L^* \leq N(N-1)/2$ weighted links, summing up to a total weight $W^* = \sum_{i<j} w_{ij}^*$, with $W^* \gg N(N-1)/2$ unweighed links. As real networks have broadly distributed weights summing up to a large $W^*$, this procedure becomes very time consuming as incredibly many rewiring steps per randomized variant must be performed. As for the binary case, an alternative procedure makes use of a naive theoretical expectation [15, 40] for the expected weight of a link in the WCM, in analogy with eq.(1):

$$\langle w_{ij} \rangle = \frac{s_i(\mathbf{W}^*)s_j(\mathbf{W}^*)}{2W^*} \qquad (15)$$

However, the above expression has been shown to have as many limitations as its binary counterpart, and to be incorrect [22].

By contrast, as we show in the Appendix, our method allows to treat the WCM analytically as in the unweighted case. Given a real weighted undirected network $\mathbf{W}^*$, our method proceeds by finding the particular values $\{x_i^*\}$ solving the $N$ coupled equations

$$\sum_{j \neq i} \frac{x_i^* x_j^*}{1 - x_i^* x_j^*} = s_i(\mathbf{W}^*) \qquad \forall i \qquad (16)$$

Note the difference of sign with respect to eq.(5). As in the binary case, the knowledge of $\vec{x}^*$ allows to obtain the expectation value $\langle X \rangle^*$ and standard deviation $\sigma^*[X]$ of any weighted topological property $X$ analytically across the ensemble of weighted graphs with, on average, the same strength sequence $\vec{s}(\mathbf{W}^*)$ as the real network $\mathbf{W}^*$. Again, the time required to obtain $\langle X \rangle^*$ is as short as that required to measure the empirical value $X(\mathbf{W}^*)$, as $\langle X \rangle^*$ can be obtained by replacing $w_{ij}^*$ with the expectation value

$$\langle w_{ij} \rangle^* = \frac{x_i^* x_j^*}{1 - x_i^* x_j^*} \qquad (17)$$

into the definition of $X(\mathbf{W}^*)$. Equation (17) corrects the naive expectation (15).

In order to apply our method, we need to choose the weighted topological properties to investigate. Generalizing binary properties to weighted graphs is arbitrary, as no unique choice exist [18, 40–42]. To better highlight the generality of our approach, here we follow ref.[41] since it introduces a way to always systematically define a weighted counterpart $\tilde{X}$ for every binary property $X$. The idea is to define $\tilde{X}$ as an average of $X$ over the ensemble of binary graphs generated by a convenient connection probability $p_{ij} = f(w_{ij}) \in [0, 1]$ which is a function of the observed weights $\{w_{ij}\}$. The functional form of $p_{ij}$ can in principle be chosen depending on the empirical properties one wants to detect; however, our purpose here is using our method to compare the empirical properties with the expected ones, rather than comparing alternative definitions of the empirical properties themselves. Therefore we make the simplest choice and, given a real weighted network $\mathbf{W}^*$, we set $p_{ij} \equiv w_{ij}^*/W^*$ where $W^* \equiv \sum_{i<j} w_{ij}^* = \sum_i s_i(\mathbf{W}^*)/2$ is the total weight. This choice yields the following definition for the weighted degree [41]:

$$\tilde{k}_i(\mathbf{W}^*) = \frac{\sum_{j \neq i} w_{ij}^*}{W^*} = \frac{s_i(\mathbf{W}^*)}{W^*} \qquad (18)$$

which is simply proportional to the strength. Similarly, the weighted ANND and clustering are defined as the counterparts of eqs.(3) and (4) [41]:

$$\tilde{k}_i^{nn}(\mathbf{W}^*) \equiv \frac{\sum_{j \neq i} \sum_{k \neq j} w_{ij}^* w_{jk}^*}{W^* \sum_{j \neq i} w_{ij}^*} \qquad (19)$$

$$\tilde{c}_i(\mathbf{W}^*) \equiv \frac{\sum_{j \neq i} \sum_{k \neq i,j} w_{ij}^* w_{jk}^* w_{ki}^*}{W^* \sum_{j \neq i} \sum_{k \neq i,j} w_{ij}^* w_{ki}^*} \qquad (20)$$

In analogy with the binary case, $\tilde{k}_i^{nn}$ and $\tilde{c}_i$ can be plotted against $\tilde{k}_i$ (or equivalently $s_i$) in order to investigate the correlation structure of the weighted network.
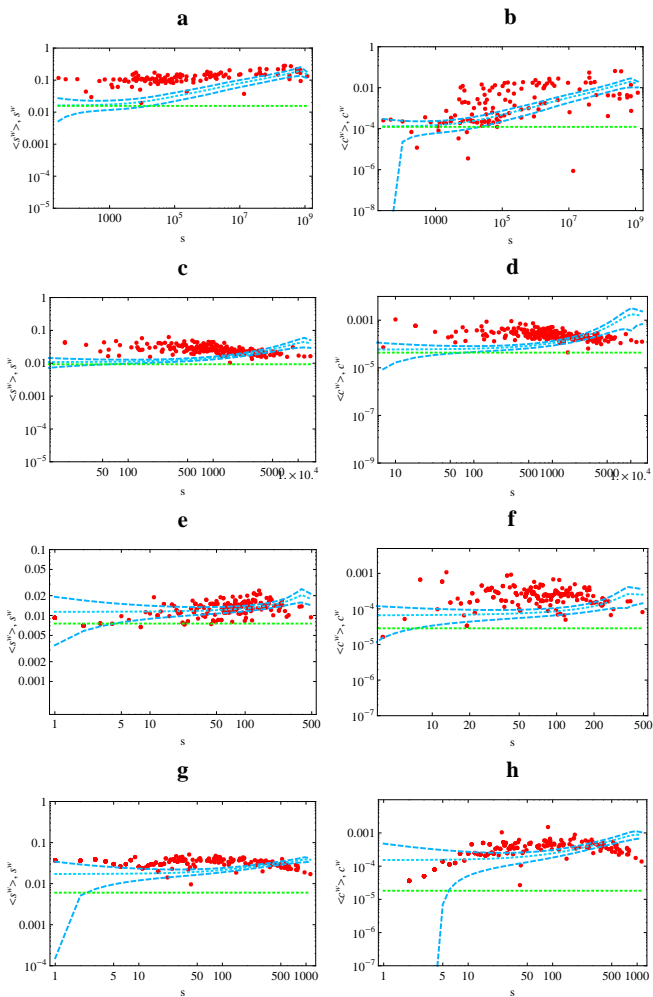
FIG. 6. Application of our method to weighted undirected networks. Red points are the empirical data and the blue dashed curves are the exact expectations obtained using our method ($\pm$ one standard deviation). Green dashed curves are the flat expectations under the *weighted random graph model*, WRG [23]. The panels report $\tilde{k}_i^{nn}$ versus $s_i$ (left) and $\tilde{c}_i$ versus $s_i$ (right) for: a) and b) the Florida Bay food web (N=128) [35], c) and d) the Italian interbank network (N=215) [31], e) and f) the *C. elegans* neural network (N=265) [29], g) and h) a snapshot of the US airport network (N=332) [28].

In fig.6 we analyze the weighted and undirected (symmetrized) versions of four networks we have already considered in the previous binary study: the the Florida Bay food web, the Italian interbank network, the C. Elegans neural network and the US airport network. We compare the empirical results with the expected trends ($\pm$ one standard deviation) under the WCM obtained using our method. For simplicity, we only show the results obtained using our method, and omit the time-consuming microcanonical comparison. Note that, since the strengths are preserved in the WCM, i.e. $\langle s_i \rangle^* = s_i(\mathbf{W}^*)$ $\forall i$, the total weight is preserved as well: $\langle W \rangle^* =$

$W^*$. We find that the empirical trends are quite scattered and variable: some are weakly increasing (Florida Bay), some are approximately constant (interbank web), others first increase and then decrease (airport network). These diverse trends must be compared with a null model which, unlike naively expected from eq.(15), is not flat and displays a not easily characterizable increasing behavior. A common feature is that, with respect to the null behavior, real weighted networks are more assortative and clustered for low values of the strength, while they are less assortative and clustered for high values for the strength. These considerations confirm that, even in the weighted case, the empirical trends are uninformative by themselves, and always require a comparison with a null model. Our method allows to treat the otherwise problematic WCM in a simple way, in straightforward analogy with the binary case.

Although we do not consider this possibility here explicitly, for weighted networks one could also enforce additional constraints on the degree sequence. This amounts to specifying not only the strength of each vertex, but also its purely topological degree [16, 19, 20]. In this case, sampling the randomized ensemble by means of computational algorithms becomes even more difficult. By contrast, our method can still be used efficiently, as the analytical expressions characterizing the corresponding maximum-entropy ensemble have been derived recently [22]. Those results easily allow to obtain the equations implied by the ML principle, as well as the expectation values of network properties over the ensemble, in a straightforward fashion.

## V. DISCUSSION

Our method make use of the correct expressions (6) and (17) for the connection probability and expected weight respectively, in place of the incorrect naive expressions (1) and (15). While the latter depend only on the properties ($k_i$ or $s_i$) of the end-point vertices $i$ and $j$, the former depend on the entire degree or strength sequence through eqs.(5) and (16). We have shown that this has a dramatic effect on the properties of the randomized ensemble. In particular, we have found that enforcing the same set of constraints in different networks can yield very different trends for the randomized properties, whose behavior is therefore highly unpredictable *a priori*. The general expectation that randomized higher-order properties (such as $\langle k_i^{nn} \rangle$ and $\langle c_i \rangle$ in unweighted networks or $\langle \tilde{k}_i \rangle$ and $\langle \tilde{c}_i \rangle$ in weighted networks) are independent of the local ones ($k_i$ or $s_i$) turns out to be only a very infrequent possibility among the possible scenarios. Indeed, we have also found increasing and decreasing trends for the randomized quantities, and shown that the particular behavior displayed by the null model highly depends on the particular values of the constraints in the original real-world network. This makes the comparison with the particular null model even more important than

previously expected, and underlines the importance of a tractable description enabled by our analytical method.

The incorrectness of eqs.(1) and (15), as well as of their directed counterparts, has another series of undesired effects, as those expressions have been explicitly used to define important structural quantities involved in network analysis. Indeed, even when not explicitly used to randomize a network, null models unavoidably enter into the analytical expressions defining many properties of interest. For instance, many popular community detection algorithms make use of the concept of *modularity* to evaluate the quality of a partition of the network against a null case [43]. A partition into communities can be represented by the matrix $\{\delta_{ij}\}$, where $\delta_{ij} = 1$ if vertices $i$ and $j$ are assigned the same community and $\delta_{ij} = 0$ otherwise. For a binary undirected network $\mathbf{A}^*$, the modularity $Q$ of the partition $\{\delta_{ij}\}$ has been defined as

$$Q \equiv \frac{1}{2L^*} \sum_{i \neq j} \delta_{ij} \left( a_{ij}^* - p_{ij} \right) \qquad (21)$$

where $p_{ij}$ is the probability that $i$ and $j$ are connected in a suitable null model, and the most frequent choice is the CM. Similarly, for a weighted undirected network $\mathbf{W}^*$ the modularity of the partition $\{\delta_{ij}\}$ is [40]

$$Q \equiv \frac{1}{2W^*} \sum_{i \neq j} \delta_{ij} \left( w_{ij}^* - \langle w_{ij} \rangle \right) \qquad (22)$$

where $\langle w_{ij} \rangle$ is the expected weight of the link joining $i$ and $j$ in the WCM. Unfortunately, the expressions for $p_{ij}$ and $\langle w_{ij} \rangle$ are always taken to be eqs.(1) and (15) respectively. To the best of our knowledge, no rigorous assessment of the consequences of using these approximations has been provided. Therefore the problems described in the present paper affect any modularity-based community detection problem in an uncontrolled way. Our methods provides the previously unavailable exact expressions (6) and (17), whose values can be inserted into eqs.(21) and (22) to have the correct modularity. A straightforward analysis of how the correct expressions change the detected community structure of real networks is an important open point to address in the future.

## VI. CONCLUSIONS

We have presented a fast and exact method to obtain analytical results about the ensemble of randomized variants of a particular real-world network that preserve its local properties. The method works for both weighted and unweighted networks, and for both directed and undirected graphs. In any case, it requires as the input only the strength or degree sequence(s), which represent the sufficient statistics of the problem. Our approach can be extended to enforce different or additional constraints, such as the reciprocity structure in directed networks or the simultaneous specification of strengths and degrees in weighted networks. Notably, our results show that maximally random networks exhibit a diverse range of behaviors which is sensitive to the particular values of the constraints displayed by the real network, making a case-by-case comparison of the observed properties with the randomized ones necessary. This diversity of outcomes is in any case not captured by widely used but incorrect expressions for the expected properties. Unfortunately, important network properties such as the modularity completely rely on such expressions, a problem that may have therefore biased previous analyses of community structure in networks. We believe that our contribution represents a promising step towards the identification of relevant information in real networks.

## Appendix A: GENERAL MAXIMUM-LIKELIHOOD METHOD

Here we describe our maximum-likelihood method in its general formulation. Our approach combines previous analytical results (obtained by one of us [11, 22, 23] and other authors [9, 14, 21]) about the properties of maximum-entropy graph ensembles with previous results (by one of us [27]) about the maximum-likelihood estimation of free parameters in such ensembles, and adds to them a new technique to obtain analytical expressions for the expectation value and standard deviation of any topological property of interest across the ensemble. After describing the method in general terms, we derive the explicit expressions that apply in the particular cases of local constraints (for undirected, directed and weighted networks). We then consider an extension to nonlocal constraints, and finally compare our analytical method with alternative computational techniques.

### 1. Maximum-entropy probability distribution

Our method aims at characterizing analytically the properties of families of randomized variants of a particular real network. In more rigorous terms, a family of randomized network variants is a statistical ensemble of graphs where a set of structural constraints has been specified, and the rest of the topology is completely random. Let us denote by $\mathbf{G}$ a generic network in the ensemble, and by $\mathbf{G}^*$ the particular real-world network that we need to randomize. The ensemble will consist of all possible networks $\{\mathbf{G}\}$ of the same type of $\mathbf{G}^*$ (binary/weighted, directed/undirected), and will include $\mathbf{G}^*$ itself. For binary (either directed or undirected) networks, each graph $\mathbf{G}$ is completely specified by its adjacency matrix $\mathbf{A}$, i.e. $\mathbf{G} \equiv \mathbf{A}$. Similarly, for weighed (either directed or undirected) networks, each graph $\mathbf{G}$ is completely specified by its weight matrix $\mathbf{W}$, i.e. $\mathbf{G} \equiv \mathbf{W}$. We will keep our discussion completely

general and use $\mathbf{G}$ to indicate a graph of any type (directed/undirected, binary/weighted). Thus $\mathbf{G}$ can always be thought of as a matrix with entries $\{g_{ij}\}$, where $g_{ij}$ represents the (either binary or non-negative) weight of the edge $(i, j)$. Any topological property $X$ evaluates to $X(\mathbf{G})$ when measured on the particular network $\mathbf{G}$, i.e. it is an (arbitrarily complicated) function of the entries $\{g_{ij}\}$.

Each graph $\mathbf{G}$ in the ensemble has an occurrence probability $P(\mathbf{G})$ whose form is determined by the particular constrains enforced. This probability must always be such that

$$\sum_{\mathbf{G}} P(\mathbf{G}) = 1 \tag{A1}$$

where the sum runs over all graphs in the ensemble. The expectation value of a topological property $X$ across the ensemble is the mean value (ensemble average)

$$\langle X \rangle \equiv \sum_{\mathbf{G}} X(\mathbf{G}) P(\mathbf{G}) \tag{A2}$$

Let us denote the set of constraints $\{C_a\}$ by the vector $\vec{C}$, where each $C_a$ is a topological property that, unlike any other generic property $X$, we need to tune to the particular value displayed by the real network $\mathbf{G}^*$. Enforcing the constraints *exactly*, i.e. allowing only the graphs $\mathbf{G}$ such that $\vec{C}(\mathbf{G}) = \vec{C}(\mathbf{G}^*)$, results in a so-called *microcanonical* ensemble characterized by the uniform probability

$$P(\mathbf{G}) = \begin{cases} 1/\mathcal{N}[\vec{C}(\mathbf{G}^*)] & \text{if } \vec{C}(\mathbf{G}) = \vec{C}(\mathbf{G}^*) \\ 0 & \text{otherwise} \end{cases} \tag{A3}$$

where $\mathcal{N}[\vec{C}(\mathbf{G}^*)]$ denotes the number of graphs in the ensemble for which the value of each constraint $C_a$ equals the value $C_a(\mathbf{G}^*)$. Microcanonical graph ensembles are hard to deal with analytically, and they are most often sampled computationally by generating many randomized networks explicitly, using probabilistic rules that ensure that the constraints are matched exactly. Currently, such computational techniques are the only available method to randomize a real network. Unfortunately, the need to sample the ensemble explicitly and generating a large number of randomized graphs makes this approach computationally demanding, time consuming and beyond analytic control.

In order to develop a randomization method which is fast and analytically tractable, we exploit the results in ref.[14] and consider the alternative possibility to enforce the constraints *on average*, i.e. by only specifying their expectation values $\langle \vec{C} \rangle$. The resulting ensemble is a *(grand)canonical* one where each graph $\mathbf{G}$ is assigned a probability $P(\mathbf{G})$ that maximizes the Shannon-Gibbs entropy

$$S \equiv -\sum_{\mathbf{G}} P(\mathbf{G}) \ln P(\mathbf{G}) \tag{A4}$$

subject to the constraints considered. The desired maximum-entropy graph probability can be found by introducing a set of Lagrange multipliers $\vec{\theta} = \{\theta_a\}$ enforcing the constraints $\vec{C} = \{C_a\}$. The resulting conditional (on the value of $\vec{\theta}$) probability reads [14]

$$P(\mathbf{G}|\vec{\theta}) = \frac{e^{-H(\mathbf{G}, \vec{\theta})}}{Z(\vec{\theta})} \tag{A5}$$

where $H(\mathbf{G}, \vec{\theta})$ is the *graph Hamiltonian* defined as the linear combination

$$H(\mathbf{G}, \vec{\theta}) \equiv \sum_a \theta_a C_a(\mathbf{G}) = \vec{\theta} \cdot \vec{C}(\mathbf{G}) \tag{A6}$$

and the normalizing quantity $Z(\vec{\theta})$ is the *partition function*, defined as

$$Z(\vec{\theta}) \equiv \sum_{\mathbf{G}} e^{-H(\mathbf{G}, \vec{\theta})} \tag{A7}$$

The above results show that the graph probability $P(\mathbf{G}|\vec{\theta})$ always depends on the value $\vec{\theta}$, which in turn depends on the constraints considered. As a consequence, we can rewrite eq.(A2) more explicitly as a function of $\vec{\theta}$:

$$\langle X \rangle_{\vec{\theta}} \equiv \sum_{\mathbf{G}} X(\mathbf{G}) P(\mathbf{G}|\vec{\theta}) \tag{A8}$$

where $\langle \cdot \rangle_{\vec{\theta}}$ denotes that the ensemble average is evaluated at the particular parameter choice $\vec{\theta}$. The above expression clarifies that the expectation value of any topological property $X$ depends on the specific enforced constraints through $\vec{\theta}$. Different choices of the constraints imply different values of $\vec{\theta}$, $P(\mathbf{G}|\vec{\theta})$, and $\langle X \rangle_{\vec{\theta}}$.

### 2. Maximum-likelihood parameter estimation

As we mentioned, maximum-entropy graph ensembles generated by eq.(A5) have been used extensively to characterize mathematically networks with specified properties [9, 11, 14, 21, 22]. However, previous studies did not focus on the randomization of a particular real network (which is our main interest here), but rather on the effects that the specification of certain structural properties has on other aspects of network topology. As a consequence, the Lagrange multipliers $\{\theta_a\}$ have been considered as free parameters, generally drawn from carefully chosen probability densities [14, 21, 22] that allow analytical results, in terms of which the properties of the network model have been investigated. In most cases, the aim has been to explore the topological properties in the thermodynamic limit $N \to \infty$, where $N$ is the number of vertices of the network. This means that only generic statistical properties of real networks, such as a power-law degree distribution, were used to generate the

ensemble. However, this implies that the specific properties of a particular real network (such as deviations of individual vertices from the fitted degree distribution, the intrinsic finiteness of the system, etc.) are ignored and, more importantly, that there is no correspondence between the vertices of the real network and those of the model. Thus this approach allows to inspect the properties of maximum-entropy graph ensembles, but does not allow the latter to be considered as null models of *a particular real network*. As a consequence, it cannot be used to detect empirical topological patterns consisting of statistically significant deviations from a null network model.

Here we make one step forward and construct, for a given choice of the constraints, the particular maximum-entropy graph ensemble representing the family of correctly randomized counterparts of a given real network $\mathbf{G}^*$. Explicitly, we consider a grandcanonical ensemble of graphs with the same number $N$ of vertices as the real network, and for a given choice of the constraints we fit the model defined by eq.(A5) to the empirical network $\mathbf{G}^*$. To this end, we exploit previous results [27] showing that maximum-entropy graph ensembles defined by eq.(A5) are a particular class of models for which the maximum-likelihood principle provides an excellent method of parameter estimation, since they are free from problems of bias afflicting other network models. In particular, it can be easily shown [27] that the log-likelihood

$$\mathcal{L}(\vec{\theta}) \equiv \ln P(\mathbf{G}^*|\vec{\theta}) = -H(\mathbf{G}^*, \vec{\theta}) - \ln Z(\vec{\theta}) \qquad \text{(A9)}$$

to obtain the real network $\mathbf{G}^*$ is maximized by the particular parameter choice $\vec{\theta}^*$ such that the ensemble average $\langle C_a \rangle_{\vec{\theta}^*}$ of each constraint $C_a$ equals the empirical value $C_a(\mathbf{G}^*)$ measured on the real network:

$$\langle \vec{C} \rangle^* \equiv \langle \vec{C} \rangle_{\vec{\theta}^*} = \sum_{\mathbf{G}} \vec{C}(\mathbf{G}) P(\mathbf{G}|\vec{\theta}^*) = \vec{C}(\mathbf{G}^*) \qquad \text{(A10)}$$

where we have used $\langle \cdot \rangle^*$ as a shorthand notation to indicate the ensemble average $\langle \cdot \rangle_{\vec{\theta}^*}$ evaluated at the particular value $\vec{\theta}^*$. The above results means that the maximum likelihood principle indicates, for maximum-entropy graph ensembles, precisely the parameter choice that ensures that the desired constraints are met. This is not true in general: in other network models, tuning the average values of the topological properties of interest to their empirical values requires a parameter choice which in general does not maximize the likelihood to obtain the real network [27], thus introducing a bias in the analysis.

The idea to take the observed constraints $\vec{C}(\mathbf{A}^*)$ as the input and find the 'hidden' values $\vec{\theta}^*$ that generate those constraints as the most probable ones was already proposed in ref.[27] with the purpose of checking whether $\vec{\theta}^*$ correlates with some external set of empirical non-topological quantities, thus unveiling possible mechanisms shaping the network topology. Here we make progress, noting that finding the values $\vec{\theta}^*$ represents a preliminary step in order to generate the randomized ensemble we are looking for, and have complete analytic control over it. This is completely independent of whether there are external empirical quantities correlating with $\vec{\theta}^*$.

Note that in eqs.(A8) and (A10) the expectation values and the model parameters play inverted roles: while in eq.(A8) the expectation values are obtained as a function of the parameters $\vec{\theta}$ which can be varied arbitrarily, in eq.(A10) the observed constraints, which are measured on the particular real network and are therefore given as an input, are used to fix the model parameters to the values $\vec{\theta}^*$. Interestingly, this opposite line of research has been used quite extensively in traditional social network analysis (where maximum-entropy ensembles of networks are widely used under the names of $p^*$, logit or exponential random graph models [44, 45]) but has not yet been transferred to the randomization problem frequently occurring in complex networks theory. As we show below, the maximum-likelihood parameter choice is exactly what we need in order to obtain statistically correct expectations over ensembles of randomized variants of any particular real-world network. This allows to understand which properties of a real-world network can be simply traced back to the enforced constraints, and which require more complicated explanations. Another important difference with respect to the main approach followed in social network analysis is that our method allows to analyze weighted networks in exactly the same way as binary graphs, which are instead usually not studied within the $p^*$ framework. As a consequence, some of the analytical results we derive and use represent previously unavailable tools to study weighted networks (and maximum-entropy ensembles of them) through a straightforward analogy with binary networks. Finally, in all the applications we consider it is always possible to find the maximum-likelihood parameter values $\vec{\theta}^*$ exactly even for large networks, without resorting to the approximate techniques traditionally used in social network analysis. Therefore our approach extends in many directions the connection between exponential random graphs and maximum-entropy network ensembles, and strengthens considerably the existing relation between social science and network theory.

### 3. Expectation values of topological properties

Equation (A10) provides an implicit expression for the value $\vec{\theta}^*$, and solving it is equivalent to maximizing eq.(A9). The numerical value of the solution $\vec{\theta}^*$ is the key ingredient we are looking for in order to detect topological patterns in the real network $\mathbf{G}^*$ analytically, without performing any time-consuming computational randomization. Indeed, if we insert the value $\vec{\theta}^*$ into eq.(A8) we

obtain

$$\langle X \rangle^* \equiv \langle X \rangle_{\vec{\theta}^*} = \sum_{\mathbf{G}} X(\mathbf{G}) P(\mathbf{G}|\vec{\theta}^*) \qquad (A11)$$

which provides the exact expected value of any topological property $X$ across the maximum-entropy graph ensemble where the expected values $\langle \vec{C} \rangle$ of the topological properties $\vec{C}$ chosen as constraints are set equal to the empirical values $\vec{C}(\mathbf{G}^*)$ measured on the real network $\mathbf{G}^*$, as ensured by eq.(A10). For simplicity, given a real network $\mathbf{G}^*$ and a set of constraints $\vec{C}$, we shall sometimes call $\langle X \rangle^*$ the *randomized value* of the topological property $X$. The comparison of $\langle X \rangle^*$ with the empirical value $X(\mathbf{G}^*)$ allows to assess whether, in the real network $\mathbf{G}^*$, the topological property $X$ requires additional information besides that provided by the properties $\vec{C}$. If $X(\mathbf{G}^*)$ is sufficiently close to $\langle X \rangle^*$ (within a statistical error that we determine in A 4), one can conclude that the enforced constraints $\vec{C}$ fully explain the property $X$. By contrast, if $X(\mathbf{G}^*)$ is significantly different from $\langle X \rangle^*$, then the properties $\vec{C}$ do not explain the property $X$, which means that the structure of $\mathbf{G}^*$ is determined by other factors besides those determining $\vec{C}$. This allows to assess which topological properties can be traced back to (i.e. explained by) the chosen constraints in any real network, and which can not. Trivially, if $X$ is one of the properties among the enforced constraints (i.e. if $X = C_a$ for some $a$), then eq.(A10) implies $X(\mathbf{G}^*) = \langle X \rangle^*$ by construction.

Note that any other parameter choice $\vec{\theta} \neq \vec{\theta}^*$ would not enforce the chosen constraints and would yield an expectation value $\langle X \rangle_{\vec{\theta}}$ different from the desired one, i.e. not corresponding to the correct randomized value $\langle X \rangle^*$ for that particular network and for that particular choice of the constraints. This clarifies why previous results [9, 11, 14, 21, 22] about the properties of maximum-entropy ensembles, that were obtained using $\vec{\theta}$ as a free parameter unrelated to the empirical values $\vec{C}(\mathbf{G}^*)$ and to the real network $\mathbf{G}^*$ itself, cannot be used in order to solve the pattern detection problem considered here. Also note that $\vec{C}(\mathbf{G}^*)$ is the *sufficient statistics* of our problem, which completely determines $\theta^*$ through eq.(A10) and consequently any randomized property $\langle X \rangle^*$. The knowledge of the other topological properties of the real network $\mathbf{G}^*$ is useless. This means that two real networks $\mathbf{G}_1^*$ and $\mathbf{G}_2^*$ with exactly the same values $\vec{C}(\mathbf{G}_1^*) = \vec{C}(\mathbf{G}_2^*)$ of the constraints generate the same maximum-entropy ensemble, and give rise to the same value of $\theta^*$ and $\langle X \rangle^*$, as should be.

Clearly, the possibility to solve eq.(A10) and to obtain the randomized properties through eq.(A11) both depend on whether one manages to rewrite the formal expression for $\langle X \rangle_{\vec{\theta}}$ in eq.(A8) in a simplified form that avoids the unfeasible actual enumeration of all graphs $\{\mathbf{G}\}$ in the ensemble. In practical terms, this means that not all specifications of the constraints $\vec{C}$ allow to solve

eq.(A10) and obtain $\vec{\theta}^*$, and not all topological properties $X$ allow to be averaged exactly through eq.(A11). However, as we describe in B, the first step can always be carried out successfully whenever one considers *local* constraints as the ones of interest for us. Similarly, as we now show in general and then restate more explicitly in each particular case, the expectation value $\langle X \rangle^*$ of any higher-order topological property $X$ of interest can be rewritten, either exactly or approximately, in a way that is only as complicated as measuring $X(\mathbf{G}^*)$ on a single network, rather than on all graphs $\{\mathbf{G}\}$ in the ensemble. This represents a major advantage of our method: the computation of an expectation value across the entire ensemble of graphs is only as time-consuming as the computation of the corresponding observed value on the empirical network $\mathbf{G}^*$. Thus, if the observed value can be computed in reasonable time, the same is true for the expectation value. To see this, we write down an approximated expression for $\langle X \rangle^*$ as a Taylor expansion. Note that any property $X(\mathbf{G})$ depends in general on all the entries $\{g_{ij}\}$ of the matrix $\mathbf{G}$, which are the fundamental degrees of freedom of the problem. The ensemble average of $g_{ij}$ reads

$$\langle g_{ij} \rangle = \sum_{\mathbf{G}} g_{ij} P(\mathbf{G}|\vec{\theta}) \qquad (A12)$$

If we define the gradient matrix of any topological property $X(\mathbf{G})$ as

$$\nabla X(\mathbf{G}) \equiv \begin{pmatrix} \frac{\partial X(\mathbf{G})}{\partial g_{11}} & \cdots & \frac{\partial X(\mathbf{G})}{\partial g_{1N}} \\ \vdots & & \vdots \\ \frac{\partial X(\mathbf{G})}{\partial g_{N1}} & \cdots & \frac{\partial X(\mathbf{G})}{\partial g_{NN}} \end{pmatrix} \qquad (A13)$$

and if we denote by $\langle \mathbf{G} \rangle$ the matrix whose entries $\langle \mathbf{G} \rangle_{ij}$ are the expectation values $\langle g_{ij} \rangle$, it is possible to expand $\langle X \rangle$ around $\langle \mathbf{G} \rangle$ and write the multidimensional first-order Taylor expansion

$$X(\mathbf{G}) = X(\langle \mathbf{G} \rangle) + \sum_{i,j} (g_{ij} - \langle g_{ij} \rangle) \left( \frac{\partial X}{\partial g_{ij}} \right)_{\mathbf{G} = \langle \mathbf{G} \rangle} + \dots$$
$$= X(\langle \mathbf{G} \rangle) + (\mathbf{G} - \langle \mathbf{G} \rangle) * \nabla X(\langle \mathbf{G} \rangle) + \dots \quad (A14)$$

In the above expression, $(\cdot)_{\mathbf{G} = \langle \mathbf{G} \rangle}$ means that we are evaluating the quantity in brackets by replacing each $g_{ij}$ with $\langle g_{ij} \rangle$, and

$$\mathbf{A} * \mathbf{B} \equiv \sum_{i,j} a_{ij} b_{ij} \qquad (A15)$$

denotes the scalar product of two matrices $\mathbf{A}$ and $\mathbf{B}$, and the double sum runs over all $N(N-1)$ ordered pairs of vertices (with $i \neq j$). Note that for an undirected network, where $g_{ij} = g_{ji}$ by construction, half of the terms in the sum in eq.(A14) will be equal to zero, since one has either $\partial X / \partial g_{ji} = 0$ or $\partial X / \partial g_{ij} = 0$, depending on whether $g_{ij}$ or $g_{ji}$ appears in the formal definition of

*X*. With the above approximation, the expectation value of *X* reads

$$\langle X \rangle = X(\langle \mathbf{G} \rangle) + \dots \tag{A16}$$

since the first-order terms vanish. The above formula shows that, if one evaluates $\langle X \rangle$ by simply replacing $\mathbf{G}$ with $\langle \mathbf{G} \rangle$ into $X(\mathbf{G})$ (*linear approximation*), the difference with respect to the exact expectation value is only in the second- and higher-order terms. This is true for any value of $\vec{\theta}$, on which all expectation values depend. As already explained, our method consists in choosing the particular value $\vec{\theta}^*$ solving eq.(A10), which yields an expectation value

$$\langle X \rangle^* = X(\langle \mathbf{G} \rangle^*) + \dots \tag{A17}$$

Among all possible parameter values $\vec{\theta}$, the choice of $\vec{\theta}^*$ ensures that the deviation of the approximate value $X(\langle \mathbf{G} \rangle)$ from the exact one $X(\mathbf{G})$ in eq.(A14) is minimal, since $\langle \mathbf{G} \rangle^*$ is as close as possible to $\mathbf{G}$, if the chosen constraints $\vec{C}$ are chosen as a reference to measure the difference between $\langle \mathbf{G} \rangle$ and $\mathbf{G}$. In particular, when $X$ coincides with one of the enforced constraints $C_a$, eq.(A17) becomes an exact expression, as we mentioned. Moreover, as we show later in B, most topological properties of interest in our analysis are either multilinear functions of statistically independent matrix elements $\{g_{ij}\}$ (in which case the expectation value $\langle X \rangle^*$ is exactly $X(\langle \mathbf{G} \rangle^*)$) or ratios of multilinear functions (in which case the numerator and denominator will be separately evaluated exactly, and the approximation will only affect the ratio). This means that in most cases eq.(A17) will be an exact expression, rather than an approximate one. In general, even for uncommon or very complicated topological properties $X$ for which eq.(A17) remains an approximation, recall that we are interested in determining an interval of statistically significant values around $\langle X \rangle^*$, rather than $\langle X \rangle^*$ alone. Thus, since the difference between the exact and the approximate value of $\langle X \rangle^*$ is typically much smaller than the standard deviation of $X$ (that we obtain below), using eq.(A17) is in any case an excellent way to proceed.

The above discussion clarifies that a safe approximation to the randomized value $\langle X \rangle^*$ of any topological property of interest is given by simply replacing each $g_{ij}$ with $\langle g_{ij} \rangle^*$ in the definition of the property $X(\mathbf{G})$, in the same way as the empirical value $X(\mathbf{G}^*)$ is obtained by replacing each $g_{ij}$ with the observed entry $g_{ij}^*$ of $\mathbf{G}^*$ in the definition of $X(\mathbf{G})$. This means that the empirical value $X(\mathbf{G}^*)$ and the approximate randomized value $X(\langle \mathbf{G} \rangle^*)$ require exactly the same computational time, which makes our method faster than any other available alternative approach (and in general as fast as possible). Clearly, in order to evaluate eq.(A17) the complete knowledge of the values

$$\langle g_{ij} \rangle^* = \sum_{\mathbf{G}} g_{ij} P(\mathbf{G}|\vec{\theta}^*) \tag{A18}$$

is required. While for generic choices of $\vec{C}$ it may be impossible to obtain the formal expression for $\langle g_{ij} \rangle_{\vec{\theta}}$ and/or the particular parameter value $\vec{\theta}^*$, in B we show that local constraints always allow to obtain $\langle g_{ij} \rangle^*$ exactly. This makes the problem analytically solvable, and implies that our method becomes very simple in all the applications of interest.

### 4. Variances of topological properties

As we mentioned, another important advantage of our method is the possibility to obtain, besides the expectation value, the analytical expression for the standard deviation of any topological property of interest. This provides a statistical error allowing to detect significant deviations of any empirically observed topological quantity $X(\mathbf{G}^*)$ from its randomized value $\langle X \rangle^*$. To this end, we employ the fundamental expression relating the variance of a function of many random variables to the variances of the latter, whose most popular consequence is the general formula for the propagation of errors in experimental measurements. In our notation, the variance of a topological property $X$ across the ensemble is defined as

$$\sigma^2[X] \equiv \langle X^2 \rangle - \langle X \rangle^2 = \langle (X - \langle X \rangle)^2 \rangle \tag{A19}$$

(which depends on $\vec{\theta}$). Using the linear approximation in eq.(A14) we can write

$$\sigma^2[X] = \langle [X(\mathbf{G}) - X(\langle \mathbf{G} \rangle)]^2 \rangle \tag{A20}$$

$$= \sum_{i,j} \sum_{t,s} \sigma[g_{ij}, g_{ts}] \left( \frac{\partial X}{\partial g_{ij}} \frac{\partial X}{\partial g_{ts}} \right)_{\mathbf{G} = \langle \mathbf{G} \rangle} + \dots$$

where

$$\sigma[g_{ij}, g_{ts}] \equiv \langle (g_{ij} - \langle g_{ij} \rangle)(g_{ts} - \langle g_{ts} \rangle) \rangle$$
$$= \langle g_{ij} g_{ts} \rangle - \langle g_{ij} \rangle \langle g_{ts} \rangle \tag{A21}$$

is the covariance of $g_{ij}$ and $g_{ts}$, and

$$\langle g_{ij} g_{ts} \rangle = \sum_{\mathbf{G}} g_{ij} g_{ts} P(\mathbf{G}|\vec{\theta}) \tag{A22}$$

For the 'diagonal' terms given by $i = t$ and $j = s$, the covariance $\sigma[g_{ij}, g_{ts}]$ equals the variance

$$\sigma^2[g_{ij}] \equiv \langle g_{ij}^2 \rangle - \langle g_{ij} \rangle^2 = \sigma[g_{ij}, g_{ij}] \tag{A23}$$

(again, both $\sigma[g_{ij}, g_{ts}]$ and $\sigma^2[g_{ij}]$ depend on $\vec{\theta}$). In a different context where $X$ is a function of many experimental quantities $\{g_{ij}\}$, eq.(A21) provides the general formula for the propagation of errors (from $\{g_{ij}\}$ to $X$), if the measured value of $g_{ij}$ is used as the best estimate for $\langle g_{ij} \rangle$, and if its experimental error is used in place of $\sigma[g_{ij}]$. Here, we do not need approximate estimates for $\langle g_{ij} \rangle$ and $\sigma[g_{ij}]$, since both quantities can be completely specified: even if there is always a single observation, i.e.

the real network $\mathbf{G}^*$, the latter generates the entire ensemble of graphs which is described by the probability $P(\mathbf{G}|\vec{\theta}^*)$, as we discussed in detail in A 2.

As for the expectation value $\langle X \rangle^*$, our approach proceeds by evaluating the standard deviation $\sigma[X]^*$ at the particular parameter value $\vec{\theta}^*$ solving eq.(A10):

$$\sigma^*[X] = \sqrt{\sum_{i,j}\sum_{t,s} \sigma^*[g_{ij}, g_{ts}]\left(\frac{\partial X}{\partial g_{ij}}\frac{\partial X}{\partial g_{ts}}\right)_{\mathbf{G}=\langle \mathbf{G}\rangle^*} + \cdots} \tag{A24}$$

where

$$\sigma^*[g_{ij}, g_{ts}] = \langle g_{ij} g_{ts}\rangle^* - \langle g_{ij}\rangle^* \langle g_{ts}\rangle^*. \tag{A25}$$

Equations (A24) and (A25) show that the values

$$\langle g_{ij} g_{ts}\rangle^* = \sum_{\mathbf{G}} g_{ij} g_{ts} P(\mathbf{G}|\vec{\theta}^*) \tag{A26}$$

are the fundamental quantities, besides the averages $\langle g_{ij}\rangle^*$ given by eq.(A18), required in order to obtain the standard deviation $\sigma^*[X]$ of any topological property $X$. For generic choices of the constraints $\vec{C}$, obtaining the value of $\langle g_{ij} g_{ts}\rangle^*$ can be very complicated or even impossible, as we already discussed for $\langle g_{ij}\rangle^*$. However, as we will show, local constraints always allow to evaluate analytically all the covariances, and hence the standard deviation $\sigma^*[X]$ of any property $X$.

Equation (A24) is the key expression providing the statistical error associated with $\langle X \rangle^*$. For any topological quantity $X$, our method allows to assess whether the empirical value $X(\mathbf{G}^*)$ is consistent with the randomized value $\langle X \rangle^*$ within $z$ standard deviations (where $z$ is a conveniently chosen threshold value), i.e. whether

$$|X(\mathbf{G}^*) - \langle X \rangle^*| \le z\sigma^*[X] \tag{A27}$$

As long as the above inequality holds, it is legitimate to say that the particular property $X$ evidences no significant deviation of the real network $\mathbf{G}^*$ from a null model where the constraints $\vec{C}$ are specified. This means that the observed value $X(\mathbf{G}^*)$ requires no explanation besides those accounting for the observed values $\vec{C}(\mathbf{G}^*)$ of the constraints. By contrast, if the above inequality is violated, then one has a signature that the observed network $\mathbf{G}^*$ is not completely a result of the specification of the constraints $\vec{C}$. Additional mechanisms, besides those determining the values of the constraints, are at work. In other words, higher-order patterns which cannot be traced to low-level constraints are present, and our method is able to detect them. In practice, in order to discriminate between the two possibilities, it is useful to compute the two values

$$\langle X \rangle^* \pm z\sigma^*[X] \tag{A28}$$

which delimit the region within which an observed value $X(\mathbf{G}^*)$ would imply the acceptance of the null model from the one where an observed value $X(\mathbf{G}^*)$ would imply the rejection of the null model. As an alternative, rather than fixing a threshold value for $z$, one can directly compute the number of standard deviations by which the expected and the empirical value of $X$ differ, i.e. the *z-score*

$$z[X] \equiv \frac{X(\mathbf{G}^*) - \langle X \rangle^*}{\sigma^*[X]} \tag{A29}$$

Large positive (negative) values of $z[X]$ indicate that $X(\mathbf{G}^*)$ is substantially larger (smaller) than expected, while small values signal no significant deviation from the null model. This concludes the description of our method in its general form. In what follows, we consider the particular case of interest for the present analysis, i.e. when the constraints $\vec{C}$ are (either binary or weighted) local topological properties, or when they are nonlocal but simple enough to preserve the analytical character of the method.

## Appendix B: LOCAL CONSTRAINTS

The most important case is when the constraints $\vec{C}$ are *local* (or first-order) topological properties, i.e. properties determined by moving only one step away from a vertex, thus reaching only its first neighbours. In binary undirected networks the fundamental local property is the degree $k_i = \sum_{j \ne i} a_{ij}$, while in weighted undirected networks the corresponding quantity is the strength $s_i = \sum_{j \ne i} w_{ij}$. In directed networks, a pair of inward and outward variants of the same quantities (i.e. the in-degree $k_i^{in}$ and out-degree $k_i^{out}$, or the in-strength $s_i^{in}$ and out-strength $s_i^{out}$) characterizes the local properties of each vertex. Choosing local constraints is the natural option when one is interested in understanding the effects that the specification of low-order properties, involving only direct interactions, has on higher-order properties involving longer chains of interactions. In what follows, we therefore discuss our method in detail in the particular case of local constraints. We will consider both binary and weighted networks, and both undirected and directed links. Importantly, we will show that in all these cases the graph probability $P(\mathbf{G}|\vec{\theta})$ factorizes as

$$P(\mathbf{G}|\vec{\theta}) = \prod_{i<j} D_{ij}(g_{ij}, g_{ji}|\vec{\theta}) \tag{B1}$$

where the product runs over all unordered pairs of vertices $(i, j)$ (with $i < j$) and $D_{ij}(g, g'|\vec{\theta})$ is the dyadic probability that the pair $(g_{ij}, g_{ji})$ takes the particular value $(g, g')$, i.e. the joint probability that $g_{ij} = g$ and simultaneously $g_{ji} = g'$. Clearly,

$$D_{ij}(g, g'|\vec{\theta}) = D_{ji}(g', g|\vec{\theta}) \tag{B2}$$

Note that $D_{ij}(g, g'|\vec{\theta})$ is normalized such that

$$\sum_{g,g'} D_{ij}(g, g'|\vec{\theta}) = 1 \qquad (B3)$$

where $g$ and $g'$ run over all the allowed values for $g_{ij}$ and $g_{ji}$ ($g = 0, 1$ for binary networks, while $g = 0, 1 \cdots + \infty$ for weighted networks; the same for $g'$). The marginal probability that $g_{ij}$ takes the particular value $g$, independently of the value of $g_{ji}$, is

$$P_{ij}(g|\vec{\theta}) = \sum_{g'} D_{ij}(g, g'|\vec{\theta}) \qquad (B4)$$

and, consistently with eq.(B3), is normalized such that

$$\sum_{g} P_{ij}(g|\vec{\theta}) = 1. \qquad (B5)$$

Note that for undirected networks, where $g_{ij} = g_{ji}$ by construction, we have

$$D_{ij}(g, g'|\vec{\theta}) = \delta_{g,g'} P_{ij}(g|\vec{\theta}) \qquad (B6)$$

where $\delta_{g,g'} = 1$ if $g = g'$ and $\delta_{g,g'} = 0$ if $g \neq g'$.

The factorization of $P(\mathbf{G}|\vec{\theta})$ according to eq.(B1) implies that eq.(A12) can be rewritten as

$$\langle g_{ij} \rangle = \sum_{g} g P_{ij}(g|\vec{\theta}) \qquad (B7)$$

which can always be obtained analytically. Using the latter, eq.(A10) can be simply rewritten exactly as

$$\vec{C}(\langle \mathbf{G} \rangle^*) = \vec{C}(\mathbf{G}^*) \qquad (B8)$$

which allows the maximum-likelihood parameter values $\vec{\theta}^*$ appearing in $\langle \mathbf{G} \rangle^*$ to be easily calculated numerically. Alternatively (e.g. depending on the software used) one can calculate $\vec{\theta}^*$ by directly maximizing the log-likelihood defined in eq.(A9), which in this case takes the simpler form

$$\mathcal{L}(\vec{\theta}) \equiv \ln P(\mathbf{G}^*|\vec{\theta}) = \sum_{i<j} \ln D_{ij}(g_{ij}^*, g_{ji}^*|\vec{\theta}). \qquad (B9)$$

In both cases, even for very large networks this preliminary parameter estimation ranges from seconds to tens of seconds using standard software packages on an ordinary laptop [47]. This implies that eq.(B7) can always be evaluated exactly at the particular parameter choice $\vec{\theta}^*$, providing the correct value $\langle g_{ij} \rangle^*$ in terms of which the ensemble average $\langle X \rangle^*$ of any topological property $X$ can be obtained analytically through eq.(A17). Thus, as we discussed, the time required to obtain $\langle X \rangle^*$ (which formally is an average over all possible graphs in the ensemble) is just the same as that required in order to measure $X(\mathbf{G}^*)$ on the real network $\mathbf{G}^*$. This makes our method incredibly faster than other randomization procedures that require the actual computational generation of many randomized variants (necessarily sampling only a part of the ensemble) of the real network, on each of which $X$ must be computed explicitly before performing a final average approximating $\langle X \rangle^*$.

The standard deviation $\sigma^*[X]$ of any property $X$ can be evaluated very easily as well. Equation eq.(B1) implies that if $(i, j)$ and $(t, s)$ are two distinct pairs of vertices then

$$\langle g_{ij} g_{ts} \rangle = \langle g_{ij} \rangle \langle g_{ts} \rangle \qquad (B10)$$
$$\sigma[g_{ij}, g_{ts}] = 0 \qquad (B11)$$

By contrast, if $i = t$ and $j = s$ then

$$\langle g_{ij} g_{ij} \rangle = \langle g_{ij}^2 \rangle = \sum_{g} g^2 P_{ij}(g|\vec{\theta}) \qquad (B12)$$
$$\sigma[g_{ij}, g_{ij}] = \langle g_{ij}^2 \rangle - \langle g_{ij} \rangle^2 = \sigma^2[g_{ij}] \qquad (B13)$$

Finally, if $i = s$ and $j = t$ we have

$$\langle g_{ij} g_{ji} \rangle = \sum_{g,g'} g g' D_{ij}(g, g'|\vec{\theta}) \qquad (B14)$$
$$\sigma[g_{ij}, g_{ji}] = \langle g_{ij} g_{ji} \rangle - \langle g_{ij} \rangle \langle g_{ji} \rangle \qquad (B15)$$

Again, all the above quantities can be obtained analytically and evaluated exactly at the particular value $\vec{\theta}^*$ solving eq.(B8). As a consequence, if eqs.(B11), (B13) and (B15) are inserted into eq.(A21), we find that the expression for the variance $\sigma^2[X]$ of any topological property $X$ reduces from eq.(A24) to the simpler formula

$$(\sigma^*[X])^2 = \sum_{i,j} \left[ \left( \sigma^*[g_{ij}] \frac{\partial X}{\partial g_{ij}} \right)^2_{\mathbf{G}=\langle \mathbf{G} \rangle^*} \right. \qquad (B16)$$
$$\left. + \sigma^*[g_{ij}, g_{ji}] \left( \frac{\partial X}{\partial g_{ij}} \frac{\partial X}{\partial g_{ji}} \right)_{\mathbf{G}=\langle \mathbf{G} \rangle^*} \right] + \dots$$

involving only a single sum over pairs of vertices. In the above expression, we have kept our convention to let the sum run always over all possible ordered pairs of vertices, thus considering the pairs $(i, j)$ and $(j, i)$ as distinct terms in the summation. For ensembles of directed networks, $g_{ij}$ and $g_{ji}$ are different random variables which may or may not be dependent on each other (depending on the enforced constraints, as we show in detail below). Equation (B16) takes care of both possibilities by including the covariance $\sigma^*[g_{ij}, g_{ji}]$. For ensembles of undirected networks, $g_{ij}$ and $g_{ji}$ are actually the same random variable and are thus perfectly correlated, which means $\sqrt{\sigma^*[g_{ij}, g_{ji}]} = \sqrt{\sigma^*[g_{ij}, g_{ij}]} = \sigma^*[g_{ij}]$. Again, eq. (B16) takes care of this by compensating the summation over a doubled number of terms with the presence of the covariances which exactly restore the correct expression. In such a way, one does not have to care whether the network is undirected when using eq.(B16), which therefore applies without modifications to all the cases we will consider below. Different cases only differ by the specific expression of $\sigma^*[g_{ij}, g_{ji}]$. This is very convenient when implementing the formula computationally. Another desirable consequence of formally treating $g_{ij}$ and $g_{ji}$ as

different variables even in undirected networks is that in eq.(B16) the derivative $\partial X/\partial g_{ts}$ of any function $X(\mathbf{G})$ of (a subset of) the entries $\{g_{ij}\}$ can always be computed by repeatedly applying the elementary rule

$$\frac{\partial g_{ij}}{\partial g_{ts}} = \delta_{it}\delta_{js} \qquad (\text{B17})$$

(where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$) for both directed and undirected graphs.

Summarizing the results discussed so far, we showed that for local constraints our method allows $\langle g_{ij}\rangle^*$, $\langle g_{ij}^2\rangle^*$ and $\langle g_{ij}g_{ji}\rangle^*$ to be computed exactly, and to use them in order to obtain the expected randomized value $\langle X\rangle^*$ and standard deviation $\sigma^*[X]$ of any topological property $X$ through eqs.(A17) and (B16) respectively. Unlike alternative computational methods, our approach is completely analytical and allows to evaluate the randomized value $\langle X\rangle^*$ in just the same time as that required to measure $X$ on the original real network $\mathbf{G}^*$, plus a negligible preliminary time required to find the parameter values $\vec{\theta}^*$ numerically through eq.(B8). The simple steps through which our method proceeds in the case of local constraints can be summarized as follows:

1. choose the desired representation for the real network $\mathbf{G}^*$ (directed/undirected, binary/weighted) and the corresponding grandcanonical ensemble of graphs $\{\mathbf{G}\}$;

2. specify the local constraints $\vec{C}(\mathbf{G})$ and use them to write the Hamiltonian $H(\mathbf{G},\vec{\theta}) = \vec{\theta}\cdot\vec{C}(\mathbf{G})$ and the probability $P(\mathbf{G}|\vec{\theta}) = e^{-H(\mathbf{G},\vec{\theta})}/Z(\vec{\theta})$ according to eqs.(A5)-(A7);

3. rewrite the graph probability analytically in the factorized form $P(\mathbf{G}|\vec{\theta}) = \prod_{i<j} D_{ij}(g_{ij}, g_{ji}|\vec{\theta})$ according to eq.(B1);

4. use $D_{ij}(g, g'|\vec{\theta})$ to determine the basic quantities $\langle g_{ij}\rangle$, $\langle g_{ij}^2\rangle$ and $\langle g_{ij}g_{ji}\rangle$ according to eqs.(B7), (B13) and (B15) respectively;

5. numerically determine the maximum-likelihood parameters $\vec{\theta}^*$ by solving eq.(B8) or alternatively maximizing eq.(B9);

6. use $\vec{\theta}^*$ to compute the ensemble average $\langle X\rangle^*$ and standard deviation $\sigma^*[X]$ of any desired topological property $X$, according to eqs.(A17) and (B16);

7. assess whether the empirical value $X(\mathbf{G}^*)$ is consistent with the randomized one $\langle X\rangle^*$ using either the interval in eq.(A28) or the $z$-score in eq.(A29).

For completeness, in the above list we have included all the logical steps involving also the initial derivation of the required analytical expressions. However, since those

expressions have already been derived in the literature for all the constraints we will consider in what follows, in practice our method reduces to a straightforward application of the last three steps. For clarity, in what follows we illustrate the method explicitly for a range of useful specific cases, i.e. for various choices of the constraints $\vec{C}$ and of the topological properties $X$. We will also highlight in more detail the advantages with respect to alternative methods.

### 1. Undirected configuration model

For unweighted undirected networks, each graph $\mathbf{G}$ in the ensemble is uniquely specified by its binary symmetric adjacency matrix $\mathbf{A}$ with entries $a_{ij} = a_{ji} = 1$ if vertices $i$ and $j$ are connected, and $a_{ij} = a_{ji} = 0$ otherwise. Generally, one considers loop-less graphs with $a_{ii} = 0$ unless otherwise specified. This fixes the first step of our method according to the list shown above. Thus we can replace $\mathbf{G} \to \mathbf{A}$ and $g_{ij} \to a_{ij}$ in our general notation used so far.

Given a real binary undirected network $\mathbf{A}^*$ with entries $\{a_{ij}^*\}$ and degree sequence $\vec{k}(\mathbf{A}^*)$, our method allows to compare the properties of $\mathbf{A}^*$ with those displayed by a randomized ensemble of binary undirected graphs having, on average, the same degree sequence as $\mathbf{A}^*$. As we mentioned in sec. II, the available methods have severe limitations. In particular, as noted in refs.[9] and [27], the incorrectness of eq.(1) is a consequence of the fact that it is not a proper maximum-entropy probability over the ensemble of binary graphs, i.e. it cannot be traced back to a Hamiltonian model as the ones described in A 1. By contrast, our method provides the correct solution. The appropriate choice is to include the constraint $\vec{C} = \vec{k}$ into eq.(A6) and obtain the corresponding correct probability [14]. This is precisely what the steps 2-4 of our method prescribe. For the sake of completeness, we briefly sketch the main results. If $\vec{C}(\mathbf{A}) = \vec{k}(\mathbf{A})$, the Hamiltonian reads

$$H(\mathbf{A},\vec{\theta}) = \sum_i \theta_i k_i(\mathbf{A}) = \sum_{i<j}(\theta_i + \theta_j)a_{ij} \qquad (\text{B18})$$

The partition function can be calculated exactly [14] as

$$Z(\vec{\theta}) = \sum_{\mathbf{A}} e^{-H(\mathbf{A},\vec{\theta})} = \prod_{i<j}(1 + e^{-\theta_i - \theta_j}) \qquad (\text{B19})$$

Therefore the graph probability can be written in the factorized form (B1) as follows

$$P(\mathbf{A}|\vec{\theta}) = \prod_{i<j} D_{ij}(a_{ij}, a_{ji}|\vec{\theta}) = \prod_{i<j} P_{ij}(a_{ij}|\vec{\theta}) \qquad (\text{B20})$$

where

$$P_{ij}(a_{ij}|\vec{\theta}) = p_{ij}^{a_{ij}}(1 - p_{ij})^{(1-a_{ij})} \qquad (\text{B21})$$

is the *mass probability function* of a Bernoulli-distributed binary random variable $a_{ij}$, with success probability

$$p_{ij} = \frac{e^{-\theta_i - \theta_j}}{1 + e^{-\theta_i - \theta_j}} \qquad \text{(B22)}$$

representing the probability that a link between $i$ and $j$ is present. Introducing the new variable $x_i \equiv e^{-\theta_i}$, not to be confused with the symbol $X$ used so far, and changing notation from $\vec{\theta}$ to $\vec{x}$, the expectation value of $a_{ij}$ is simply given by

$$\langle a_{ij} \rangle_{\vec{x}} = p_{ij} = \frac{x_i x_j}{1 + x_i x_j} \qquad \text{(B23)}$$

Also, since $a_{ij}^2 = a_{ij}$, the second moment is

$$\langle a_{ij}^2 \rangle_{\vec{x}} = \langle a_{ij} \rangle_{\vec{x}} \qquad \text{(B24)}$$

Finally, if $(i, j)$ and $(t, s)$ are two distinct pairs of vertices, then $a_{ij}$ and $a_{ts}$ are independent random variables and

$$\langle a_{ij} a_{ts} \rangle_{\vec{x}} = \langle a_{ij} \rangle_{\vec{x}} \langle a_{ts} \rangle_{\vec{x}} \qquad \text{(B25)}$$

This completes the fourth step in our method.

The fifth step consists in finding the particular parameter values $\vec{x}^*$ that maximize eq.(B9), that in this case reads

$$\mathcal{L}(\vec{x}) \equiv \ln P(\mathbf{A}^* | \vec{x}) = \sum_i k_i(\mathbf{A}^*) \ln x_i - \sum_{i<j} \ln(1 + x_i x_j) \qquad \text{(B26)}$$

Equivalently [27], the parameters $\vec{x}^*$ can be found solving the following $N$ coupled equations enforcing the desired constraints as in eq.(B8):

$$\sum_{j \neq i} \frac{x_i^* x_j^*}{1 + x_i^* x_j^*} = k_i(\mathbf{A}^*) \qquad \forall i \qquad \text{(B27)}$$

Importantly, since $x_i \equiv e^{-\theta_i}$ and $\theta_i$ is a real number, the solution we are looking for is the one where $x_i^* > 0$ $\forall i$. This solution is unique. Even for large networks, the above parameter estimation ranges from seconds to tens of seconds even on an ordinary laptop.

Once the parameters $\vec{x}^*$ are found, we can proceed to the sixth step and exploit eq.(A17) to obtain the expectation values of the properties $X$ of interest:

$$\langle X \rangle^* = \sum_{\mathbf{A}} X(\mathbf{A}) P(\mathbf{A} | \vec{x}^*) = X(\langle \mathbf{A} \rangle^*) + \dots \qquad \text{(B28)}$$

In particular, the expectation value of the ANND defined in eq.(3) is

$$\langle k_i^{nn} \rangle^* = \frac{\sum_{j \neq i} \sum_{k \neq j} \langle a_{ij} \rangle^* \langle a_{jk} \rangle^*}{\sum_{j \neq i} \langle a_{ij} \rangle^*} \qquad \text{(B29)}$$

and the expectation value of the clustering coefficient defined in eq.(4) is

$$\langle c_i \rangle^* = \frac{\sum_{j \neq i} \sum_{k \neq i,j} \langle a_{ij} \rangle^* \langle a_{jk} \rangle^* \langle a_{ki} \rangle^*}{\sum_{j \neq i} \sum_{k \neq i,j} \langle a_{ij} \rangle^* \langle a_{ki} \rangle^*} \qquad \text{(B30)}$$

where $\langle a_{ij} \rangle^* = x_i^* x_j^* / (1 + x_i^* x_j^*)$. Similarly, the standard deviation $\sigma^*[X]$ can be evaluated using eq.(B16), which here reads

$$\sigma^*[X] = \sqrt{\sum_{i,j} \left( \sigma^*[a_{ij}] \frac{\partial X}{\partial a_{ij}} \right)^2_{\mathbf{A} = \langle \mathbf{A} \rangle^*} + \dots} \qquad \text{(B31)}$$

where $\sigma^*[a_{ij}] = \sqrt{\langle a_{ij} \rangle^* (1 - \langle a_{ij} \rangle^*)} = \sqrt{x_i^* x_j^*}/(1 + x_i^* x_j^*)$. It is straightforward to obtain $\sigma^*[X]$ in terms of $\vec{x}^*$ alone, by using the derivation rule (B17):

$$\frac{\partial a_{ij}}{\partial a_{ts}} = \delta_{it} \delta_{js} \qquad \text{(B32)}$$

This can also be implemented symbolically in adequate softwares.

### 2. Directed configuration model

Binary directed networks have an asymmetric adjacency matrix $\mathbf{A}$ with entries $a_{ij} = 1$ if a directed link from $i$ to $j$ is there, and $a_{ij} = 0$ otherwise. Given a real binary directed network $\mathbf{A}^*$ with out-degree sequence $\vec{k}^{out}(\mathbf{A}^*)$ and in-degree sequence $\vec{k}^{in}(\mathbf{A}^*)$, our method provides analytical expressions for the expectation values and standard deviations of topological properties across the maximum-entropy ensemble of binary directed graphs with out-degree sequence $\vec{k}^{out}(\mathbf{A}^*)$ and in-degree sequence $\vec{k}^{in}(\mathbf{A}^*)$. The Hamiltonian is now

$$H(\mathbf{A}, \vec{\alpha}, \vec{\beta}) = \sum_i [\alpha_i k_i^{out}(\mathbf{A}) + \beta_i k_i^{in}(\mathbf{A})]$$
$$= \sum_{i \neq j} (\alpha_i + \beta_j) a_{ij} \qquad \text{(B33)}$$

The partition function can be calculated exactly [14] as

$$Z(\vec{\alpha}, \vec{\beta}) = \sum_{\mathbf{A}} e^{-H(\mathbf{A}, \vec{\alpha}, \vec{\beta})} = \prod_{i \neq j} (1 + e^{-\alpha_i - \beta_j}) \qquad \text{(B34)}$$

The graph probability is now

$$P(\mathbf{A} | \vec{\alpha}, \vec{\beta}) = \prod_{i < j} D_{ij}(a_{ij}, a_{ji} | \vec{\alpha}, \vec{\beta}) = \prod_{i \neq j} P_{ij}(a_{ij} | \vec{\alpha}, \vec{\beta}) \qquad \text{(B35)}$$

where

$$P_{ij}(a_{ij} | \vec{\alpha}, \vec{\beta}) = p_{ij}^{a_{ij}} (1 - p_{ij})^{(1 - a_{ij})} \qquad \text{(B36)}$$

and

$$p_{ij} = \frac{e^{-\alpha_i - \beta_j}}{1 + e^{-\alpha_i - \beta_j}} \qquad \text{(B37)}$$

Setting $x_i \equiv e^{-\alpha_i}$ and $y_i \equiv e^{-\beta_i}$, the expectation value of $a_{ij}$ is

$$\langle a_{ij} \rangle_{\vec{x}, \vec{y}} = p_{ij} = \frac{x_i y_j}{1 + x_i y_j} \qquad \text{(B38)}$$

The second moment is

$$\langle a_{ij}^2 \rangle_{\vec{x},\vec{y}} = \langle a_{ij} \rangle_{\vec{x},\vec{y}} \tag{B39}$$

Finally, if $(i,j)$ and $(t,s)$ are two distinct pairs of vertices, now including the case $(t,s) = (j,i)$, then

$$\langle a_{ij} a_{ts} \rangle_{\vec{x},\vec{y}} = \langle a_{ij} \rangle_{\vec{x},\vec{y}} \langle a_{ts} \rangle_{\vec{x},\vec{y}} \tag{B40}$$

The log-likelihood (B9) to maximize is

$$\mathcal{L}(\vec{x},\vec{y}) = \sum_i \left[ k_i^{out}(\mathbf{A}^*) \ln x_i + k_i^{in}(\mathbf{A}^*) \ln y_i \right] \\ - \sum_{i \neq j} \ln(1 + x_i y_j) \tag{B41}$$

and the values $\vec{x}^*$, $\vec{y}^*$ that realize the maximum can alternatively be found by solving the $2N$ coupled equations

$$\sum_{j \neq i} \frac{x_i^* y_j^*}{1 + x_i^* y_j^*} = k_i^{out}(\mathbf{A}^*) \qquad \forall i \tag{B42}$$

$$\sum_{j \neq i} \frac{x_j^* y_i^*}{1 + x_j^* y_i^*} = k_i^{in}(\mathbf{A}^*) \qquad \forall i \tag{B43}$$

corresponding to eq.(B8). Again, we are looking for the solution where $x_i^* > 0$ and $y_i^* > 0$ $\forall i$. Expectation values can still be obtained using eq.(B28). In particular, the directed ANNDs defined in eqs.(9) and (10) have expectation values

$$\langle k_i^{nn,out} \rangle^* = \frac{\sum_{j \neq i} \sum_{k \neq j} \langle a_{ij} \rangle^* \langle a_{jk} \rangle^*}{\sum_{j \neq i} \langle a_{ij} \rangle^*} \tag{B44}$$

$$\langle k_i^{nn,in} \rangle^* = \frac{\sum_{j \neq i} \sum_{k \neq j} \langle a_{ji} \rangle^* \langle a_{kj} \rangle^*}{\sum_{j \neq i} \langle a_{ji} \rangle^*} \tag{B45}$$

where $\langle a_{ij} \rangle^* = x_i^* y_j^* / (1 + x_i^* y_j^*)$. Similarly, the standard deviation $\sigma^*[X]$ can still be evaluated through eqs.(B31) and (B32), now using $\sigma^*[a_{ij}] = \sqrt{x_i^* y_j^*}/(1 + x_i^* y_j^*)$.

### 3. Weighted configuration model

When weighted undirected networks are considered, each graph $\mathbf{G}$ in the ensemble is specified by its non-negative symmetric matrix $\mathbf{W}$ whose (without loss of generality) integer entry $w_{ij}$ represents the weight of the link between vertices $i$ and $j$ (including $w_{ij} = 0$ if no link is there). Thus we can replace $\mathbf{G} \to \mathbf{W}$ and $g_{ij} \to w_{ij}$ in the general notation. As we mentioned in the main text, in the weighted configuration model a real weighted undirected network $\mathbf{W}^*$ with entries $\{w_{ij}^*\}$ is compared with a maximum-entropy ensemble of weighted undirected graphs having the same strength sequence $\vec{s}(\mathbf{W}^*)$. In our method, by setting $\vec{C} = \vec{s}$ into eq.(A6) we obtain the Hamiltonian

$$H(\mathbf{W}, \vec{\theta}) = \sum_i \theta_i s_i(\mathbf{W}) = \sum_{i<j} (\theta_i + \theta_j) w_{ij} \tag{B46}$$

The partition function is [22]

$$Z(\vec{\theta}) = \sum_{\mathbf{W}} e^{-H(\mathbf{W},\vec{\theta})} = \prod_{i<j} \frac{1}{1 - e^{-\theta_i - \theta_j}} \tag{B47}$$

and is only defined if $\theta_i > 0$ $\forall i$. The graph probability is [22]

$$P(\mathbf{W}|\vec{\theta}) = \prod_{i<j} D_{ij}(w_{ij}, w_{ji}|\vec{\theta}) = \prod_{i<j} P_{ij}(w_{ij}|\vec{\theta}) \tag{B48}$$

where

$$P_{ij}(w_{ij}|\vec{\theta}) = p_{ij}^{w_{ij}} (1 - p_{ij}) \tag{B49}$$

is the mass probability function of a geometrically-distributed [23] integer random variable $w_{ij}$, with success probability

$$p_{ij} = e^{-\theta_i - \theta_j} \tag{B50}$$

representing the probability that a link between $i$ and $j$ is present. Introducing $x_i \equiv e^{-\theta_i} \in [0,1)$, the expectation value of $w_{ij}$ is

$$\langle w_{ij} \rangle_{\vec{x}} = \frac{p_{ij}}{1 - p_{ij}} = \frac{x_i x_j}{1 - x_i x_j} \tag{B51}$$

Now in general $w_{ij}^2 \neq w_{ij}$, and the second moment is

$$\langle w_{ij}^2 \rangle_{\vec{x}} = \frac{p_{ij}(1 + p_{ij})}{(1 - p_{ij})^2} = \frac{(x_i x_j)(1 + x_i x_j)}{(1 - x_i x_j)^2} \tag{B52}$$

Finally, if $(i,j)$ and $(t,s)$ are two distinct pairs of vertices, then

$$\langle w_{ij} w_{ts} \rangle_{\vec{x}} = \langle w_{ij} \rangle_{\vec{x}} \langle w_{ts} \rangle_{\vec{x}} \tag{B53}$$

The log-likelihood (B9) reads

$$\mathcal{L}(\vec{x}) \equiv \ln P(\mathbf{W}^*|\vec{x}) = \sum_i s_i(\mathbf{A}^*) \ln x_i + \sum_{i<j} \ln(1 - x_i x_j) \tag{B54}$$

and the parameters $\vec{x}^*$ maximizing it solve the following $N$ coupled equations

$$\sum_{j \neq i} \frac{x_i^* x_j^*}{1 - x_i^* x_j^*} = s_i(\mathbf{A}^*) \qquad \forall i \tag{B55}$$

enforcing the desired constraints as in eq.(B8). Now the solution must be looked for in the region $0 \leq x_i < 1$ $\forall i$.

Through the parameters $\vec{x}^*$ we obtain the expectation values of the properties $X$ of interest:

$$\langle X \rangle^* = \sum_{\mathbf{W}} X(\mathbf{W}) P(\mathbf{W}|\vec{x}^*) = X(\langle \mathbf{W} \rangle^*) + \dots \tag{B56}$$

For instance, the expectation value of the weighted ANND defined in eq.(19) is

$$\langle \tilde{k}_i^{nn} \rangle^* = \frac{\sum_{j \neq i} \sum_{k \neq j} \langle w_{ij} \rangle^* \langle w_{jk} \rangle^*}{W^* \sum_{j \neq i} \langle w_{ij} \rangle^*} \tag{B57}$$

where we have used $\langle W \rangle^* = W*$ (see main text). Similarly, the weighted clustering coefficient defined in eq.(20) has expectation value

$$\langle \tilde{c}_i \rangle^* = \frac{\sum_{j \neq i} \sum_{k \neq i,j} \langle w_{ij} \rangle^* \langle w_{jk} \rangle^* \langle w_{ki} \rangle^*}{W^* \sum_{j \neq i} \sum_{k \neq i,j} \langle w_{ij} \rangle^* \langle w_{ki} \rangle^*} \quad \text{(B58)}$$

where $\langle w_{ij} \rangle^* = x_i^* x_j^* / (1 - x_i^* x_j^*)$. Similarly, according to eq.(B16) the standard deviation $\sigma^*[X]$ is

$$\sigma^*[X] = \sqrt{\sum_{i,j} \left( \sigma^*[w_{ij}] \frac{\partial X}{\partial w_{ij}} \right)^2_{\mathbf{W} = \langle \mathbf{W} \rangle^*} + \dots} \quad \text{(B59)}$$

where $\sigma^*[w_{ij}] = \sqrt{x_i^* x_j^*} / (1 - x_i^* x_j^*)$. The rule (B17) here reads

$$\frac{\partial w_{ij}}{\partial w_{ts}} = \delta_{it} \delta_{js} \quad \text{(B60)}$$

and allows to obtain $\sigma^*[X]$ in terms of $\vec{x}^*$ alone.

## Appendix C: NONLOCAL CONSTRAINTS

Our model can also be applied to more complicated cases where the constraints are no longer local. However, a necessary condition for our method to work with non-local constraints is that eq.(A8) can still be expressed exactly in a form which does not require the enumeration of all possible graphs (in other words, the partition function can be calculated analytically). In such a case, eq.(A10) can still be used to calculate the parameters $\vec{\theta}^*$ exactly as in the local case, and at the same time those parameters can be used to obtain the analytical expressions for the expected value and standard deviation of the topological properties of interest. Therefore, only a limited number of nonlocal constraints lend themselves to an analytical treatment. However, since the philosophy of randomization algorithms is always to enforce the simplest constraints in order to detect higher-order patterns, it turns out that the mathematically tractable constraints are also the ones of major interest. We now provide an explicit example of a choice of nonlocal constraints that is often used in empirical studies, and at the same time preserves the analytical character of our method and yields exact results.

### 1. Reciprocal configuration model

As discussed in the main text, a more constrained null model for a binary directed network $\mathbf{A}^*$ is one where the three reciprocal degree sequences $\vec{k}^{\rightarrow}(\mathbf{A}^*)$, $\vec{k}^{\leftarrow}(\mathbf{A}^*)$ and $\vec{k}^{\leftrightarrow}(\mathbf{A}^*)$ are specified, where

$$k_i^{\rightarrow}(\mathbf{A}^*) \equiv \sum_{j \neq i} a_{ij}^* (1 - a_{ji}^*) \quad \text{(C1)}$$

$$k_i^{\leftarrow}(\mathbf{A}^*) \equiv \sum_{j \neq i} a_{ji}^* (1 - a_{ij}^*) \quad \text{(C2)}$$

$$k_i^{\leftrightarrow}(\mathbf{A}^*) \equiv \sum_{j \neq i} a_{ij}^* a_{ji}^* \quad \text{(C3)}$$

The Hamiltonian for this model is

$$H(\mathbf{A}, \vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \sum_i [\alpha_i k_i^{\rightarrow}(\mathbf{A}) + \beta_i k_i^{\leftarrow}(\mathbf{A}) + \gamma_i k_i^{\leftrightarrow}(\mathbf{A})]$$

The nonlocality is manifest in the fact that, unlike the previous examples, now the (second-order) constraints involve products of two adjacency matrix entries. Despite this complication, the partition function can still be calculated exactly [11] as

$$Z(\vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \prod_{i<j} (1 + e^{-\alpha_i - \beta_j} + e^{-\alpha_j - \beta_i} + e^{-\gamma_i - \gamma_j}) \quad \text{(C4)}$$

The graph probability can still be expressed in the form (B1), i.e.

$$P(\mathbf{A} | \vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \prod_{i<j} D_{ij}(a_{ij}, a_{ji} | \vec{\alpha}, \vec{\beta}, \vec{\gamma}) \quad \text{(C5)}$$

In the above expression,

$$D_{ij}(a_{ij}, a_{ji} | \vec{\alpha}, \vec{\beta}, \vec{\gamma}) = (p_{ij}^{\rightarrow})^{a_{ij}^{\rightarrow}} (p_{ij}^{\leftarrow})^{a_{ij}^{\leftarrow}} (p_{ij}^{\leftrightarrow})^{a_{ij}^{\leftrightarrow}} (p_{ij}^{\nleftrightarrow})^{a_{ij}^{\nleftrightarrow}}$$

is the dyadic probability defined in terms of

$$a_{ij}^{\rightarrow} \equiv a_{ij}(1 - a_{ji}) \quad \text{(C6)}$$
$$a_{ij}^{\leftarrow} \equiv a_{ji}(1 - a_{ij}) \quad \text{(C7)}$$
$$a_{ij}^{\leftrightarrow} \equiv a_{ij} a_{ji} \quad \text{(C8)}$$
$$a_{ij}^{\nleftrightarrow} \equiv (1 - a_{ij})(1 - a_{ji}) \quad \text{(C9)}$$

and

$$p_{ij}^{\rightarrow} \equiv \langle a_{ij}^{\rightarrow} \rangle_{\vec{x}, \vec{y}, \vec{z}} = \frac{x_i y_j}{1 + x_i y_j + x_j y_i + z_i z_j} \quad \text{(C10)}$$

$$p_{ij}^{\leftarrow} \equiv \langle a_{ij}^{\leftarrow} \rangle_{\vec{x}, \vec{y}, \vec{z}} = \frac{x_j y_i}{1 + x_i y_j + x_j y_i + z_i z_j} \quad \text{(C11)}$$

$$p_{ij}^{\leftrightarrow} \equiv \langle a_{ij}^{\leftrightarrow} \rangle_{\vec{x}, \vec{y}, \vec{z}} = \frac{z_i z_j}{1 + x_i y_j + x_j y_i + z_i z_j} \quad \text{(C12)}$$

$$p_{ij}^{\nleftrightarrow} \equiv \langle a_{ij}^{\nleftrightarrow} \rangle_{\vec{x}, \vec{y}, \vec{z}} = \frac{1}{1 + x_i y_j + x_j y_i + z_i z_j} \quad \text{(C13)}$$

where we have set $x_i \equiv e^{-\alpha_i}$, $y_i \equiv e^{-\beta_i}$ and $z_i \equiv e^{-\gamma_i}$ [22]. The above expressions represent the dyadic expectation values.

A little algebra leads to the log-likelihood

$$\mathcal{L}(\vec{x}, \vec{y}, \vec{z}) = -\sum_{i<j} \ln(1 + x_i y_j + x_j y_i + z_i z_j) + \sum_i [k_i^{\rightarrow}(\mathbf{A}^*) \ln x_i + k_i^{\leftarrow}(\mathbf{A}^*) \ln y_i + k_i^{\leftrightarrow}(\mathbf{A}^*) \ln z_i]$$

and the values $\vec{x}^*$, $\vec{y}^*$, $\vec{z}^*$ that realize the maximum can alternatively [27] found by solving the $3N$ coupled equations

$$\sum_{j \neq i} \frac{x_i^* y_j^*}{1 + x_i^* y_j^* + x_j^* y_i^* + z_i^* z_j^*} = k_i^{\rightarrow}(\mathbf{A}^*) \qquad \forall i$$

$$\sum_{j \neq i} \frac{x_j^* y_i^*}{1 + x_i^* y_j^* + x_j^* y_i^* + z_i^* z_j^*} = k_i^{\leftarrow}(\mathbf{A}^*) \qquad \forall i$$

$$\sum_{j \neq i} \frac{z_i^* z_j^*}{1 + x_i^* y_j^* + x_j^* y_i^* + z_i^* z_j^*} = k_i^{\leftrightarrow}(\mathbf{A}^*) \qquad \forall i$$

corresponding to an example when eq.(A10) can be written explicitly even if the constraints are nonlocal. We are looking for the solution where $x_i^* > 0$, $y_i^* > 0$ and $z_i^* > 0$ $\forall i$.

The expectation values of topological properties involving products of dyadic terms can be obtained exactly without resorting to the linear approximation in eq.(A17). For instance, the number of occurrences of a particular motif $m$, where $m$ labels one of the possible 13 non-isomorphic connected motifs with three vertices, is

$$N_m \equiv \sum_{i \neq j \neq k} a_{ij}^{m,1} a_{jk}^{m,2} a_{ki}^{m,3} \qquad \text{(C14)}$$

where $a_{ij}^{m,l}$ is one of the four possible dyadic relations $a_{ij}^{\rightarrow}$, $a_{ij}^{\leftarrow}$, $a_{ij}^{\leftrightarrow}$, $a_{ij}^{\nleftrightarrow}$, and $\{a_{ij}^{m,1}, a_{jk}^{m,2}, a_{ki}^{m,3}\}$ indicates the specific triplet of dyadic relations defining motif $m$. The exact expectation value of $N_m$ is

$$\langle N_m \rangle^* \equiv \sum_{i \neq j \neq k} \langle a_{ij}^{m,1} \rangle^* \langle a_{jk}^{m,2} \rangle^* \langle a_{ki}^{m,3} \rangle^* \qquad \text{(C15)}$$

where $\langle a_{ij}^{m,1} \rangle^*$ is given by evaluating eqs.(C10)-(C13) at the particular values $\vec{x}^*$, $\vec{y}^*$, $\vec{z}^*$. The standard deviation of $N_m$, and in general of a topological property $X$, can still be obtained using eq.(B16), i.e.

$$(\sigma^*[X])^2 = \sum_{i,j} \left[ \left( \sigma^*[a_{ij}] \frac{\partial X}{\partial a_{ij}} \right)^2_{\mathbf{A} = \langle \mathbf{A} \rangle^*} \right.$$
$$\left. + \ \sigma^*[a_{ij}, a_{ji}] \left( \frac{\partial X}{\partial a_{ij}} \frac{\partial X}{\partial a_{ji}} \right)_{\mathbf{A} = \langle \mathbf{A} \rangle^*} \right] + \dots \qquad \text{(C16)}$$

where now

$$(\sigma^*[a_{ij}])^2 = \langle a_{ij} \rangle^*(1 - \langle a_{ij} \rangle^*)$$
$$= \langle a_{ij}^{\leftrightarrow} + a_{ij}^{\rightarrow} \rangle^*(1 - \langle a_{ij}^{\leftrightarrow} + a_{ij}^{\rightarrow} \rangle^*)$$

and

$$\sigma^*[a_{ij}, a_{ji}] = \langle a_{ij} a_{ji} \rangle^* - \langle a_{ij} \rangle^* \langle a_{ji} \rangle^*$$
$$= \langle a_{ij}^{\leftrightarrow} \rangle^* - \langle a_{ij}^{\leftrightarrow} + a_{ij}^{\rightarrow} \rangle^* \langle a_{ji}^{\leftrightarrow} + a_{ji}^{\rightarrow} \rangle^*$$

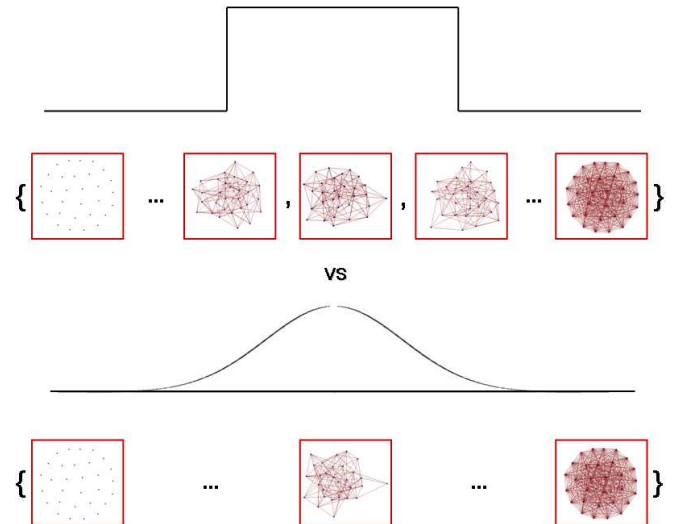which are both known exactly in terms of eqs.(C10)-(C13).



FIG. 7. Difference between the LRA-based microcanonical approach and our likelihood-based grandcanonical approach. Top: the microcanonical approach assigns non-zero probability only to the subset $\mathcal{D}(\vec{C})$ of graphs that realize the enforced constraints $\vec{C}$ (in the example shown, a given value of the number of links $L$) exactly. Bottom: by contrast, our grandcanonical approach assigns non-zero probability to all graphs, but this probability reaches its maximum value for the graphs belonging to $\mathcal{D}(\vec{C})$. In so doing, it is more robust to potential errors in the original network data (such as missing links).

## Appendix D: COMPARISON WITH COMPUTATIONAL MICROCANONICAL ALGORITHMS

The LRA-based microcanonical approach [4, 5] and our likelihood-based grandcanonical approach are in general not equivalent for finite networks. Let $\mathcal{D}(\vec{C})$ be the set of all graphs $\mathbf{G}$ that realize the enforced constraints $\vec{C} = \{C_\alpha\}$ exactly. Both approaches assign equal probabilities to all graphs that realize the constraints, i.e. $P(\mathbf{G}_1) = P(\mathbf{G}_2)$ if $\mathbf{G}_1 \in \mathcal{D}(\vec{C})$ and $\mathbf{G}_2 \in \mathcal{D}(\vec{C})$. Also, in both approaches these graphs are the most likely to occur, i.e. $P(\mathbf{G}_1) > P(\mathbf{G}_2)$ for any $\mathbf{G}_1 \in \mathcal{D}(\vec{C})$ and $\mathbf{G}_2 \notin \mathcal{D}(\vec{C})$. However the two approaches are different, the microcanonical one being very severe in assigning zero probability to any graph where the degrees are not matched exactly, i.e. $P(\mathbf{G}) = 0$ if $\mathbf{G} \notin \mathcal{D}(\vec{C})$. By contrast, in the grandcanonical approach all possible graphs can occur, even if with very different probabilities, in such a way that the ensemble average of the desired constraints over all graphs coincides with the observed values (see fig.7 for an illustration of this difference).

The above key and elegant property places grandcanonical ensembles at the basis of information theory. Notably, they are more robust to errors in the original data such as missing or overrepresented links. In presence of even a small percentage of such errors, the 'true' graph

(the unobserved one affected by errors) will never appear in the microcanonical ensemble, while it will appear with nonzero probability in the grandcanonical ensemble. As desirable, for small deviations from the observed graph the true graph will have a slightly decreased probability with respect to the one assigned by our method to the observed graph, while for larger errors the probabilities will differ by a larger amount. Therefore, while for infinite systems the microcanonical and grandcanonical ensembles become equivalent since fluctuations about the average values become negligible, in finite systems the use of grandcanonical ensembles is preferable. What is of interest for us here is the impact of the two methods on the topological properties induced on the randomized networks. To this end, we now show explicitly the relation between the two approaches when applied to particular networks. We shall only consider unweighted networks for simplicity.

In the unweighted (either directed or undirected) case, our method directly provides 'from the beginning' the explicit values of the probabilities $p_{ij}^G$ that a link from $i$ to $j$ is there. The superscript $G$ stands for 'grandcanonical', and the probability is evaluated at the parameter values that maximize the likelihood, as described above. By contrast, the microcanonical approach samples the configuration space iteratively, and the microcanonical probability $p_{ij}^M$ that a link from $i$ to $j$ is there can only be evaluated as the frequency of occurrence of the link over many randomizations. As the number of randomized networks increases, this frequency will converge to $p_{ij}^M$. However this asymptotic value will also depend on the number $R$ of elementary rewiring steps used to obtain a single randomized network. To see this, consider the trivial case $R = 0$. As no rewiring takes place, all the 'randomized' networks will in fact coincide with the original network. If the adjacency matrix of the latter has elements $\{a_{ij}\}$, this means that $p_{ij}^M = a_{ij}$. If $R$ is nonzero but still very small, $p_{ij}^M$ will not change substantially. Only if $R$ is large enough then $p_{ij}^M$ will approach $p_{ij}^G$. This is shown explicitly in fig.8, where we plot $p_{ij}^G$ as a function of $p_{ij}^M$ for all directed pairs of vertices $(i, j)$ by taking the *Little Rock Lake* food web as the starting network. As $R$ increases from $R = 0$ to $R = 10000$, the double-peaked shape (corresponding to $p_{ij}^M = a_{ij}$ independently of $p_{ij}^G$) evolves towards the identity $p_{ij}^M = p_{ij}^G$. Similar evolution patterns are observed for all the networks we analyzed. This clearly shows that in our method we obtain 'from the beginning' the values $p_{ij}^G$ to which the microcanonical $p_{ij}^M$ will converge only after several iterations. Notably, the number $R$ of rewiring steps required for $p_{ij}^M$ to converge to $p_{ij}^G$ acceptably is not known *a priori* (an indicative number being $R \geq 4L$) and without the knowledge of $p_{ij}^G$ itself. This problematic aspect of the microcanonical approach highlights another advantage of the grandcanonical one.

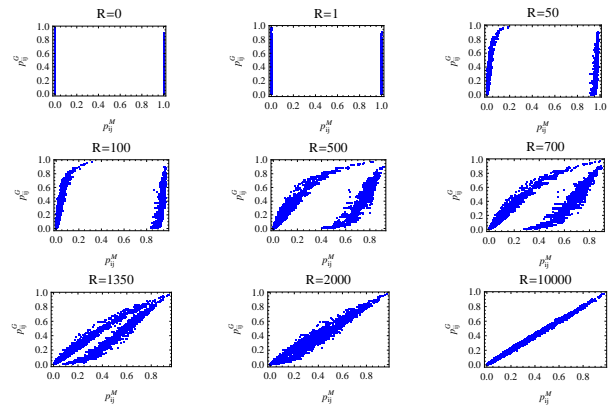As we already mentioned, the two approaches are in general not equivalent for finite networks. We can now



FIG. 8. Convergence of the microcanonical connection probability $p_{ij}^M$ (measured using the local rewiring algorithm) to the grandcanonical probability $p_{ij}^G$ (obtained using our maximum-entropy method) as the number $R$ of local rewiring moves per network increases.

state this more rigorously, and indicate at least two ways in which they may differ. First of all, $p_{ij}^M$ represent marginal probabilities, where the information about the correlations between the presence of a link between different pairs of vertices has been lost. While in the grandcanonical approach these correlations are absent, and different pairs of vertices are always statistically independent, in the microcanonical approach some weak correlations will be preserved even after many rewiring steps. These correlations arise from the microcanonical constraint of matching the degree sequence (or other constraints) exactly. Thus, while our grandcanonical method enables to compute the expected topological properties exactly, in the microcanonical approach this is not possible. Secondly, the final 'convergence' of $p_{ij}^M$ to $p_{ij}^G$ for $R \to \infty$ will in general not hold exactly. This means that the asymptotic plot of $p_{ij}^G$ versus $p_{ij}^M$ will not be a strict identity, but a narrow scatter of points close to the identity. In other words, increasing $R$ beyond a certain value will not make the quantities converge further. For some networks (such as the *Little Rock Lake* food web shown above) one may attain a better convergence than for others.

It is interesting to understand whether the degree of convergence between the two approaches depends on some property of the network. To this end, we first define two measures of discrepancy between $\{p_{ij}^G\}$ and $\{p_{ij}^M\}$, and then study how they behave on well-controlled, artificially generated networks. As measures of discrepancy, we consider the $l^2$ distance

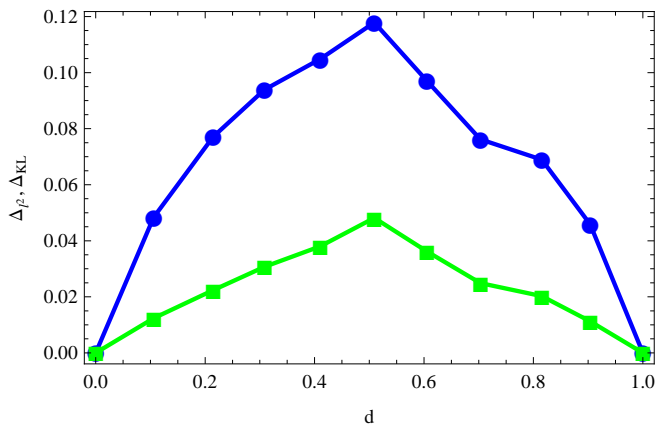$$\Delta_{l^2} \equiv \sqrt{\frac{\sum_{i \neq j} |p_{ij}^G - p_{ij}^M|^2}{N(N-1)}} \tag{D1}$$

FIG. 9. Kullback-Leibler ($\Delta_{KL}$, green squares) and $l^2$ ($\Delta_{l^2}$, blue circles) distance between microcanonical ($p_{ij}^M$) and grandcanonical ($p_{ij}^G$) marginal connection probabilities, plotted versus link density $d$.

and the Kullback-Leibler information distance

$$\Delta_{KL} \equiv \frac{\sum_{i \neq j} p_{ij}^M (\log_2 p_{ij}^M - \log_2 p_{ij}^G)}{N(N-1)} + \quad \text{(D2)}$$

$$\frac{\sum_{i \neq j}(1 - p_{ij}^M)[\log_2(1 - p_{ij}^M) - \log_2(1 - p_{ij}^G)]}{N(N-1)}$$

Note that we have normalized the above distances in such a way that both lie in the range $[0, 1]$. It is instructive

to use these distances to compare the two methods on a family of artificially generated networks. As a starting point, we considered $N = 100$ vertices, assigned each vertex a real value $x_i$ drawn randomly in the interval $[0, 1]$, and established an edge between each pair of vertices $i$ and $j$ with probability $p_{ij} = zx_ix_j/(1 + zx_ix_j)$. This choice generates maximally random networks with degree distribution controlled by $\{x_i\}$ as in eq.(B23), but has an additional parameter $z$ that tunes the overall link density $d \equiv 2L/N(N-1)$, representing the fraction of realized links. With $\{x_i\}$ kept constant, we considered various choices of $z$ and, for each of them, adopted both the microcanonical randomization and our grandcanonical method. In fig.9 we show the resulting difference between the marginal probabilities $\{p_{ij}^G\}$ and $\{p_{ij}^M\}$, as a function of link density. One can see that the two methods yield very similar results for both small and large link density, whereas for intermediate density values they display a greater difference. Even in this case, however, the distances between them are $\Delta_{l^2} \approx 0.05$ and $\Delta_{KL} \approx 0.12$, both small considering their possible range of variation.

[1] R. Albert and A.–L. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002).

[2] M.E.J. Newman, *SIAM Rev.* **45**, 167 (2003).

[3] Caldarelli, G. *Scale–free Networks. Complex Webs in Nature and Technology* (Oxford University Press, Oxford 2007).

[4] S. Maslov and K. Sneppen, *Science* **296**, 910 (2002).

[5] Maslov, S., Sneppen, K. & Zaliznyak, A. profile of the Internet. *Physica A* **333**, 529–540 (2004).

[6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon *Science* **298**, 824-827 (2002).

[7] M.E.J. Newman, S.H. Strogatz & D.J. Watts *Phys. Rev. E* **64**, 026118 (2001).

[8] Chung, F. & Lu, L. *Ann. of Combin.* **6**, 125 (2002).

[9] Park, J. & Newman, M.E.J. *Phys. Rev. E* **68**, 026112 (2003).

[10] Catanzaro, M., Boguna, M. & Pastor–Satorras, R. *Phys. Rev. E* **71**, 027103 (2005).

[11] D. Garlaschelli and M.I. Loffredo, *Phys. Rev. E* **73**, 015101(R) (2006).

[12] D.B. Stouffer, J. Camacho, W. Jiang & L.A.N. Amaral, *Proc. R. Soc. B* **274**, 1931-1940 (2007).

[13] R. Guimerá, M. Sales-Pardo L. A. N. Amaral, *Nat. Phys.* **3**, 63 (2007).

[14] Park, J. & Newman, M.E.J. *Phys. Rev. E* **70**, 066117 (2004).

[15] Serrano, M.A., & Boguna, M. *AIP Conf. Proc.* **776**, 101 (2005).

[16] Serrano, M.A., Boguna, M. & Pastor–Satorras, R. *Phys. Rev. E* **74**, 055101(R) (2006).

[17] M.-Á. Serrano, networks. *Phys. Rev. E* **78**, 026101 (2008).

[18] Barrat, A., Barthelemy, M., Pastor–Satorras, R. & Vespignani, A. *PNAS* **101**, 3747-3752 (2004).

[19] T. Opsahl, V. Colizza, P. Panzarasa J.J. Ramasco, *Phys. Rev. Lett.* **101**, 168702 (2008).

[20] Bhattacharya, K., Mukherjee, G., Saramaki, J., Kaski, K. & Manna, S. S. modelling. *J. Stat. Mech.* P02002 (2008).

[21] Bianconi, G. *Phys. Rev. E* **79**, 036114 (2009).

[22] D. Garlaschelli & M.I. Loffredo *Phys. Rev. Lett.* **102**, 038701 (2009).

[23] D. Garlaschelli, New J. of Phys. **11**, 073005 (2009).

[24] S. Melnik, A. Hackett, M. A. Porter, P. J. Mucha, J. P. Gleeson. *http://arxiv.org/abs/1001.1439*.

[25] M.E.J. Newman, *PRL* **103**, 058701 (2009).

[26] M. Boguná, R. Pastor-Satorras, A. Vespignani, *Eur. Phys. J. B* **38**, 205-209 (2004).

[27] D. Garlaschelli & M.I. Loffredo, *Phys. Rev. E* **78**, 015101(R) (2008).

[28] V. Colizza, R. Pastor-Satorras & A. Vespignani, *Nat. Phys.* **3**, 276 - 282 (2007).

[29] K. Oshio, Y. Iwasaki, S. Morita, Y. Osana, S. Gomi, E. Akiyama, K. Omata, K. Oka and K. Kawamura, *Tech. Rep. of CCeP, Keio Future* **3**, (Keio University, 2003).

[30] `http://dip.doe-mbi.ucla.edu/dip/Main.cgi`

[31] G. De Masi, G. Iori & G. Caldarelli, *Phys. Rev. E* **74**, 066112 (2006).

[32] V. Colizza, A. Flammini, M.A. Serrano & A. Vespignani, *Nat. Phys.* **2**, 110-115 (2006).

[33] H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai and A.-L. Barabási, *Nature* **407**, 651 (2000).

[34] N.D. Martinez, *Ecological Monographs* **61**, 367-392 (1991).

[35] `http://vlado.fmf.uni-lj.si/pub/networks/data/bio/foodweb/foodweb.htm`

[36] G. Fagiolo, *Phys. Rev. E* **76**, 026107 (2007).

[37] S. E. Ahnert, T. M. A. Fink, *Phys. Rev. E* **78**, 036112 (2008).

[38] D. Garlaschelli and M. I. Loffredo, *Phys. Rev. Lett.* **93**, 268701 (2004).

[39] V. Zlatic & H. Stefancic, *Phys. Rev. E* **80**, 016117 (2009).

[40] M.E.J. Newman, *Phys. Rev. E* **70**, 056131 (2004).

[41] S. E. Ahnert, D. Garlaschelli, T. M. Fink & G. Caldarelli, *Phys. Rev. E* **73**, 015101(R) (2006).

[42] J. Saramaki, M. Kivela, J.-P. Onnela, K. Kaski and J. Kertész, networks. *Phys. Rev. E* **75**, 027105 (2007).

[43] S. Fortunato, *Physics Reports* **486**(3-5), 75-174 (2010).

[44] P. Holland, S. Leinhardt, in *Sociological Methodology*, D. Heise, Ed. (Jossey-Bass, San Francisco, 1975), pp. 1-45.

[45] S. Wasserman, K. Faust, *Social Network Analysis* (Cambridge Univ. Press, New York, 1994).

[46] In the directed version of the local rewiring algorithm, two directed edges $(i, j)$ and $(k, l)$ are randomly selected and replaced with the directed edges $(i, l)$ and $(k, j)$, if the latter are not already present.

[47] We adopted the log-likelihood maximization using Wolfram Mathematica.