Multiplex reconstruction with partial information

Daniel Kaiser,[*] Siddharth Patwardhan,[*] and Filippo Radicchi[†]
Center for Complex Networks and Systems Research, Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, Indiana 47408, USA

(Received 15 June 2022; accepted 26 January 2023; published 16 February 2023)

A multiplex is a collection of network layers, each representing a specific type of edges. This appears to be a genuine representation of many real-world systems. However, due to a variety of potential factors, such as limited budget and equipment, or physical impossibility, multiplex data can be difficult to observe directly. Often, only partial information on the layer structure of the system is available, whereas the remaining information is in the form of a single-layer network. In this work we face the problem of reconstructing the hidden multiplex structure of an aggregated network from partial information. We propose an algorithm that leverages the layerwise community structure that can be learned from partial observations to reconstruct the ground-truth topology of the unobserved part of the multiplex. The algorithm is characterized by a computational time that grows linearly with the network size. We perform a systematic study of reconstruction problems for both synthetic and real-world multiplex networks. We show that the ability of the proposed method to solve the reconstruction problem is affected by the heterogeneity of the individual layers and the similarity among the layers. On real-world networks, we observe that the accuracy of the reconstruction saturates quickly as the amount of available information increases. In genetic interaction and scientific collaboration multiplexes, for example, we find that 10% of ground-truth information yields 70% accuracy, while 30% information allows for more than 90% accuracy.

DOI: 10.1103/PhysRevE.107.024309

I. INTRODUCTION

Networks have emerged as powerful modeling frameworks for relational data over the past few decades, boasting an impressive body of supporting research and methodologies [1]. However, it has become clear recently that networks in their simplest realization are incapable of properly modeling multidimensional relational data [2–6]. The literature is indeed plenty with examples of how ignoring the multidimensionality of systems in network modeling leads to fundamental errors in the characterization of both their structural and dynamical properties (see, e.g., Refs. [2,4,7–19]).

A multiplex is probably the simplest network representation of a multirelational system [2,3,20–22]. A multiplex is a collection of single-layer networks sharing common nodes; each layer of a multiplex captures a different type or flavor of pairwise interaction among nodes. This is a convenient and meaningful representation of many real-world systems, including social [23,24] and biological systems [4,25,26].

Even if the system under study is truly a multiplex, data about its topology are often available in an aggregated single-layer format. As a matter of fact, obtaining precise information about the flavor of all edges in a real multiplex could be prohibitively expensive, too time intensive, or even physically impossible. For example, it is relatively simple to detect correlations in coarse-level changes of gene expressions. Less simple, however, is the observation of fine-grained coexpression details, being prohibitively expensive for large genetic interaction systems. Similarly, neural connectomes can be reconstructed by analyzing correlations in time series of spiking neuron activity; however, observing the details of the interactions, i.e., synaptic junction types, is too expensive for large connectomes. Despite the availability of data describing functional aggregate networks, the dynamics occurring on these functional multiplexes is distinct from the one happening on their observable aggregate counterparts [27]. In this respect, there is an apparent need for tools to infer full, true multiplex structure from an associated single-layer network. In this paper we refer to this classification problem as the multiplex reconstruction problem (MRP) (see Sec. IV for its definition).

Only a few attempts to solve the MRP exist in the literature. In Ref. [28], Bagrow and Lehmann considered the MRP in the context of temporal networks, i.e., multiplex networks where layers correspond to different snapshots of the same network at different instants of time. They obtained good prediction accuracy leveraging a sparsity-enforced lasso regression technique. The method requires full knowledge not only of the aggregated network topology, but also of the node degrees in the individual layers. Zhang *et al.* considered the MRP on two-layer multiplex networks [29]. Their solution of the MRP corresponds to the maximization of the edge clustering coefficient of the individual layers, obtained via a simulated-annealing-like algorithm. The algorithm is applicable only to small and sufficiently dense multiplex networks, for example, networks representing trades of different commodities among

[*] These authors contributed equally to this work.
[†] filiradi@indiana.edu

countries. Wu *et al.* interpreted the MRP as a statistical inference problem based on the hypothesis that network layers are instances of the configuration model [30]. They developed an expectation-maximization algorithm for solving the MRP. The algorithm can be trained on partial knowledge of the ground-truth topology of a multiplex to make predictions about the unobserved edges. The method allows one to perform prediction and classification of edges of a multiplex network, and the classification task includes the possibility of an edge to belong simultaneously to multiple layers. Wu *et al.* also provided a theoretical analysis of the MRP, relating the accuracy of reconstruction of their method to a metric of the multiplex's entropy derived under the ansatz of the configuration model. Finally, De Bacco *et al.* [31] and Tarrés-Deulofeu *et al.* [32] introduced inferential methods for the analysis of multiplex networks with potentially correlated layerwise community structure. These methods are rather general and can be used to solve various inference problems, including link prediction at the level of individual layers. Both papers consider only a fivefold cross-validation scheme, where their classifiers are trained on 80% of the ground-truth topology and the 20% of the remaining links are predicted or classified. The performance of the classifiers is excellent. Both papers study only real-world multiplex networks composed of a small number of nodes but a large number of layers, possibly because the computational time of these methods scales quadratically with the number of nodes in the system. Also, the methods may not be best suited to deal with sparse network layers, as explicitly noted by Tarrés-Deulofeu *et al.*, who stated that "the multilayer models outperform the baseline models in almost all the studied cases, except for the cases in which information is too sparse for the multilayer model to recover unobserved interactions with precision" [32].

The above five articles mark the only literature directly addressing the MRP. Some existing literature deals with the related problem of determining whether the multiplex framework is indeed required to properly model an observed single-layer network. For example, Lacasa *et al.* showed that diffusion properties of a single-layer network can be used to judge whether the network is truly represented by a single layer or if instead it is better represented by a (hidden) multiplex structure [33]. Similar goals were pursued by Santoro and Nicosia with an approach that leverages Kolmogorov's complexity [17]. Finally, Vallès-Català *et al.* developed a method to determine the reliability of links in multiplex networks [34]. The method relies on the generalization of the stochastic block model from single-layer to multilayer networks. Link reliability consists in determining whether an observed (or unobserved) edge exists or not. Vallès-Català *et al.* showed that accounting for the hidden multiplexity with their generalized stochastic block model indeed increases the ability to predict the existence and/or nonexistence of edges. However, they do not directly focus on the MRP, i.e., the classification of observed edges in types or flavors.

In this paper we introduce an algorithm able to approximate MRP solutions. The computational complexity of the algorithm grows linearly with the total number of edges in the multiplex network. Our algorithm is inspired by the degree-corrected stochastic block model in the sense that the probability of an observed edge to belong to a specific layer instead of another is a function of the degree of the nodes in the layers and the community structure of the layers [35]. The latter ingredients are learned from the partial observation of the ground-truth structure of the multiplex network. Given that the two main ingredients of the classifier are the degree sequence and the community structure of the layers, we refer to it as the degree- and community-based classifier. From the systematic study of the MRP on synthetic multiplex networks, we show that the accuracy of the classification task is strongly influenced by both the heterogeneity of the individual network layers and the correlation among the structure of the layers. From the analysis of real-world networks, we show that the accuracy of the method saturates pretty quickly as the amount of partial information available on the system increases. Roughly, having 30% partial knowledge yields 90% accuracy.

II. RESULTS

In this paper we consider the MRP on multiplex networks composed of two layers only. We work under the assumption that all edges in the multiplex network are known. Depending on the experimental setup considered, we may have access to some level of ground-truth information. We remark that in our formulation of the problem edges can belong to one layer only. The above assumptions lead to a formulation of the MRP as a binary classification task. We systematically study the MRP on both synthetic and real networks. Details on the methods, networks, and experimental setups are reported in Sec. IV.

A. Reconstruction with degree and community information

We begin our analysis under a peculiar experimental setup. Network layers are instances of the configuration model [36]. Except from the preimposed degree sequence, network layers are completely random so that no community structure characterizes the networks. We work with power-law degree distributions with a tunable degree exponent. We assume that the degree-based (D) classifier has full knowledge of the degree sequences of both layers composing the multiplex network. Except for that, the D classifier is completely uninformed about the ground-truth topology of the system; thus the test set is composed of all edges in the network. Performance is measured as the area under the receiver operating characteristic curve (ROC AUC) of the binary classification task concerning edges belonging to the test set.

We use this experimental setup to understand how difficult the MRP is on synthetic multiplex networks with variable (i) degree heterogeneity at the level of individual layers and (ii) degree-degree correlation among layers. In particular, we note that network layers are generated according to a model that is compatible with the assumption made at the basis of the D classifier; thus results from this set of experiments should represent a reliable proxy for the intrinsic difficulty of the MRP.

As the results of Fig. 1 show, we find that the performance of our method to solve the MRP decreases as the degree exponent of the preimposed degree distribution increases. Essentially, solving the MRP is easier on multiplex networks

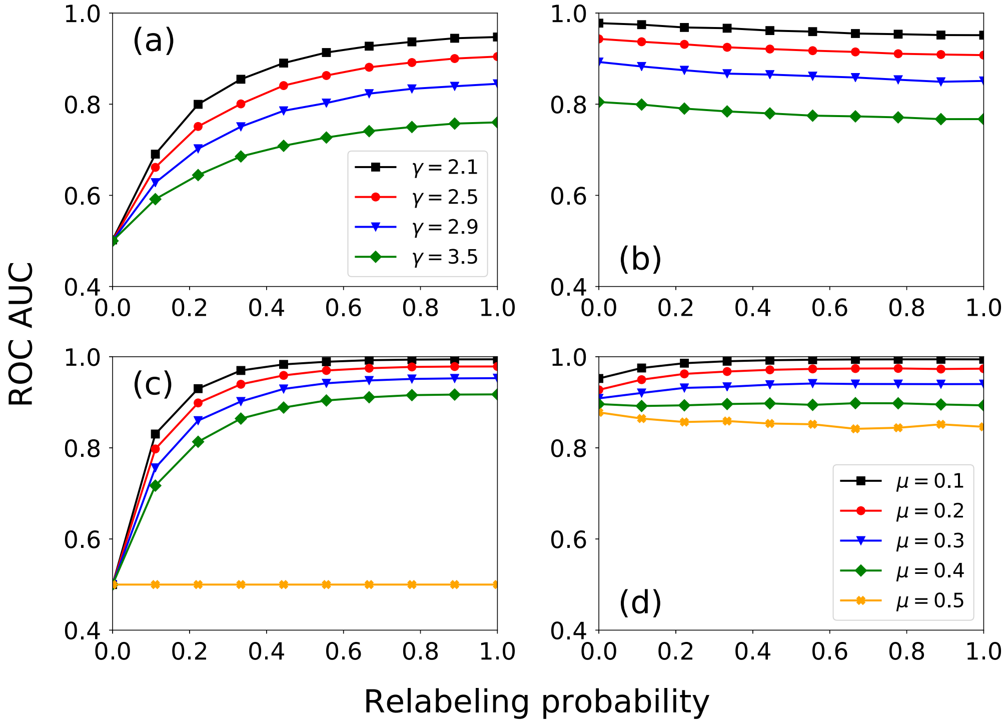

FIG. 1. Reconstruction of synthetic multiplex networks. (a) Random multiplex networks with the power-law degree distribution $P(k) \sim k^{-\gamma}$ composed of $N = 100\,000$ nodes. The minimum degree is $k_{\min} = 3$ and the maximum degree is $k_{\max} = \sqrt{N}$ if $\gamma \leqslant 3$ and $k_{\max} = N^{1/(\gamma-1)}$ if $\gamma \geqslant 3$. Different curves are obtained for different γ values. We plot the classification metric ROC AUC of the degree-based (D) classifier as a function of the probability of relabeling nodes; we relabel nodes to control for degree-degree correlation (see Sec. IV). Degrees are positively correlated when the relabeling probability is zero; they become progressively uncorrelated as the relabeling probability increases. Results are obtained over a single realization of the model. We do not observe substantial fluctuations from run to run. (b) Same as in (a) but for multiplex networks with anticorrelated degree sequences. The sequences are maximally anticorrelated for a null relabeling probability and they become uncorrelated as the relabeling probability increases. (c) Multiplex networks with preimposed community structure and power-law degree distribution $P(k) \sim k^{-\gamma}$. Graphs with $N = 10\,000$ nodes are generated according to the LFR model (see Sec. IV for details). We set degree exponent $\gamma = 2.1$, maximum degree $k_{\max} = \sqrt{N}$, and average degree $\langle k \rangle = 20$. Communities have size obeying a power-law distribution with exponent $\tau = 1.0$. The strength of the community structure is determined by the mixing parameter μ. We plot the ROC AUC of the community-based (C) classifier as a function of the probability relabeling the nodes of the multiplex. The classifier is unaware of the ground-truth degree sequences of the network layers. Different curves are obtained for different values of the mixing parameter μ controlling for the strength of the community structure of the network layers. The community structure of the two layers is correlated when no nodes are relabeled; correlation decreases as the relabeling probability increases. (d) Plot of the ROC AUC of the degree- and community-based (DC) classifier as a function of the relabeling probability for synthetic multiplex networks with preimposed community structure. The networks are the same networks as in (c). In this case, however, the classifier takes advantage also of the knowledge of the ground-truth degree sequence of the network layers.

with heterogeneous degree sequences than on multiplex networks with homogeneous degree sequences. However, it is also important that the two layers are sufficiently diverse from each other. If nodes have the same exact degrees on both layers, the performance of the D classifier is identical to that of random guessing [Fig. 1(a)]. As the degree-degree correlation of the layers decreases, performance improves. In particular, we find that the best performance is achieved for maximally anticorrelated degree sequences [Fig. 1(b)]. This is a consequence of the fact that the probability for the edge to belong to a given layer is proportional to the product of the layerwise nodes' degrees [see Eq. (2)]. When degree sequences are anticorrelated, it likely that nodes with high degree in one layer have low degree in the other layer and edges attached to such a type of nodes are relatively easy to classify correctly.

In addition, we study the performance of the community-based (C) classifier on network layers generated according to the Lancichinetti-Fortunato-Radicchi (LFR) model [37]. We use the LFR model to generate network layers with built-in community structure; we further control for the amount of degree and community-structure correlation between the layers of the multiplex. We test the performance of the classifier on these networks when informed about their ground-truth community structure, but no information on their degree sequence is used. As the results of Fig. 1(c) indicate, the performance of the C classifier increases as the strength of the community structure increases; also, the performance increases as the correlation between the community structure of the layers decreases. The results of Fig. 1(c) are due to the C classifier of Eq. (4), which expresses the probability for an edge to belong to a layer of the multiplex as proportional to

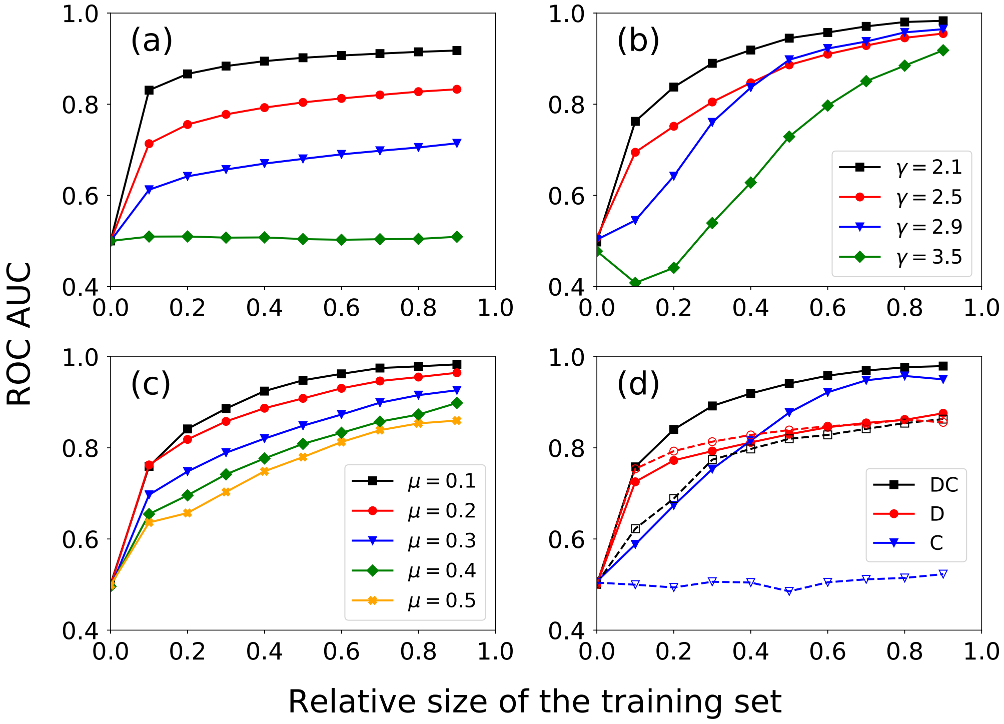

FIG. 2. Reconstruction of synthetic multiplex networks with partial edge information. (a) Plot of the performance metric ROC AUC of the degree-based (D) classifier as a function of the relative size of the training set. Multiplex networks have size $N = 100\,000$ and power-law degree distribution $P(k) \sim k^{-\gamma}$. We display results for different γ values. We impose the minimum degree $k_{\min} = 3$ for all γ values, while the maximum degree is $k_{\max} = \sqrt{N}$ for $\gamma \leqslant 3$ and $k_{\max} = N^{1/(\gamma-1)}$ for $\gamma > 3$. The degree sequences of the layers are uncorrelated. (b) Plot of the ROC AUC of the degree- and community-based (DC) classifier as a function of the relative size of the training set. Tests are performed on synthetic multiplex networks with preimposed community structure constructed according to the LFR model with $N = 10\,000$, $\langle k \rangle = 5.0$, and $k_{\max} = \sqrt{N}$ for $\gamma \leqslant 3$ and $k_{\max} = N^{1/(\gamma-1)}$ for $\gamma > 3$. The community power-law exponent is $\tau = 1.0$ and the mixing parameter is $\mu = 0.1$. The degree sequences and community structures of the layers are uncorrelated. (c) Same as in (b) but for fixed degree exponent $\gamma = 2.1$ and different μ values. (d) Comparison of the performance of the different classifiers on a synthetic graph with built-in community structure (closed symbols and full curves) and without community structure (open symbols and dashed curves). We plot the performance metric of the DC classifier, the D classifier, and the community-based (C) classifier. Graphs with community structure are constructed using the LFR multiplex model with $N = 10\,000$, $\gamma = 2.1$, $k_{\max} = \sqrt{N}$, $\langle k \rangle = 5.0$, $\tau = 1.0$, and $\mu = 0.1$. We use the configuration model with $N = 10\,000$, $\gamma = 2.1$, $k_{\min} = 3$, and $k_{\max} = \sqrt{N}$ for the graphs with no community structure. The degree sequences and community structures of the layers are uncorrelated.

the strength of the layerwise community structure. When the community structures of the layers are anticorrelated, intracommunity edges have associated a high probability to belong to the correct layer; the complementary probability is instead low due to the fact that the edge is seen as an intercommunity edge in the other layer.

Finally, we apply the degree- and community-based (DC) classifier to the LFR multiplex networks. We find that the combination of the two ingredients leads to a significant boost in classification performance [Fig. 1(d)]. Decreasing correlation leads to a visible improvement in classification performance only if the community structure of the layers is strong enough [Fig. 1(d)].

B. Reconstruction with partial edge information

The experimental setups considered in the preceding section are useful to understand intrinsic properties of the MRP. However, the setups are not very representative of applications in the real world. For example, it appears unrealistic to have full and exact knowledge of the layerwise community structure of the multiplex, but no information about the topology of the multiplex. From now on, we work using a standard experimental setup where we assume that the edges in the multiplex are randomly divided in a training set and a test set. The relative size of one set over the other is the main control parameter of our experiments. The classifier is trained from the knowledge of the ground-truth flavor of the edges within the training set. Also here performance is measured as the ROC AUC of the binary classification task concerning edges belonging to the test set.

We consider the MRP on synthetic multiplex networks (see Fig. 2). In this set of experiments, the degree sequences and community structures of the layers are uncorrelated. Similarly to what was reported in the preceding section, we find that the accuracy of the DC classifier grows as the heterogeneity of the degree distribution [Fig. 2(a)] and the strength of the community structure [Figs. 2(b) and 2(c)] increase. Depending on whether or not the network layers have an assortative structure, the classification is enhanced regardless of whether

TABLE I. Real-world multiplex networks. From left to right, we report the name of the data set and reference where the data set was introduced, the name of the network layers that compose the multiplex, the number of nodes within the two layers, the number of edges for each of the two layers, the number of edges shared by the two layers, the modularity of the best Louvain partition of each layer, and the normalized mutual information between the two partitions of the two layers. Here DLR denotes Docklands Light Railway.

Data set	α	β	N	$\lvert\mathcal{E}^{(\alpha)}\rvert$	$\lvert\mathcal{E}^{(\beta)}\rvert$	$\lvert\mathcal{E}^{(\alpha,\beta)}\rvert$	$Q^{(\alpha)}$	$Q^{(\beta)}$	NMI
arXiv [39]	physics.data-an	cond-mat.dis-nn	7187	11 929	4785	2556	0.89	0.96	0.77
arXiv [39]	physics.data-an	cond-mat.stat-mech	5963	13 326	1423	1159	0.86	0.98	0.75
arXiv [39]	cond-mat.dis-nn	cond-mat.stat-mech	4342	6839	2080	502	0.91	0.97	0.78
D. melanogaster [4]	direct	suppressive	7519	23 911	1798	66	0.45	0.64	0.49
D. melanogaster [4]	direct	additive	7486	23 928	1376	49	0.46	0.67	0.48
D. melanogaster [4]	suppressive	additive	1005	1395	956	469	0.66	0.73	0.59
C. elegans [40]	electric	chemical monadic	273	406	777	111	0.67	0.51	0.37
C. elegans [40]	electric	chemical polyadic	277	355	1541	162	0.70	0.44	0.38
C. elegans [40]	chemical monadic	chemical polyadic	273	258	1073	630	0.70	0.40	0.37
London [41]	underground	overground	321	301	72	11	0.82	0.78	0.67
London [41]	underground	DLR	311	312	46	0	0.83	0.70	0.69
London [41]	overground	DLR	126	83	46	0	0.77	0.69	0.70
S. pombe [4]	physical association	direct interaction	2577	1038	6622	445	0.87	0.51	0.51
S. pombe [4]	physical association	colocalization	3009	1431	30 752	52	0.84	0.23	0.44
S. pombe [4]	direct interaction	colocalization	3782	6940	30 677	127	0.52	0.23	0.40
H. sapiens [4]	physical association	direct interaction	17 415	37 361	72 004	12 284	0.53	0.41	0.32
H. sapiens [4]	physical association	colocalization	15 254	82 050	16 189	1386	0.42	0.64	0.35
H. sapiens [4]	direct interaction	colocalization	3782	6940	30 677	2238	0.52	0.23	0.40
R. norvegicus [4]	physical association	direct interaction	2520	2610	832	178	0.70	0.93	0.63
R. norvegicus [4]	physical association	colocalization	2065	2757	88	31	0.71	0.91	0.59
R. norvegicus [4]	direct interaction	colocalization	1074	991	100	19	0.92	0.91	0.78

the classifier takes advantage of its community-based component [Fig. 2(d)]. If the graph has no community structure, the DC classifier may also display lower performance than the simple D classifier. However, this seems to happen only if the size of the training set is sufficiently small; further, if the system size is increased, an apparent gap in performance is no longer visible (see Fig. 5). In all the experiments considered in Fig. 2, the accuracy of the classifier saturates quite quickly with the amount of partial information used to train the classifier. These general observations on the performance of the DC classifier to solve the MRP on synthetic networks are only mildly affected by the average degree of the network and its size. Specifically, we find that the performance of the classifier is almost unaffected by the average degree of multiplex network [see Fig. 6(a)]. Also, we find that, as the system size is increased, the performance of the reconstruction algorithm mildly increases [see Figs. 6(b) and 6(c)].

C. Time complexity of the reconstruction algorithms

The proposed DC classifier generates solutions of the MRP in a time that grows almost linearly with the total number of edges in the multiplex. The classifier benefits from the scalability of the Louvain algorithm, i.e., the method used to detect communities in the network layers [38]. We stress that the DC classifier can leverage any suitable community detection method to produce solutions of the MRP; however, the time complexity of the classifier may be dramatically increased by that of the community detection method. The fact

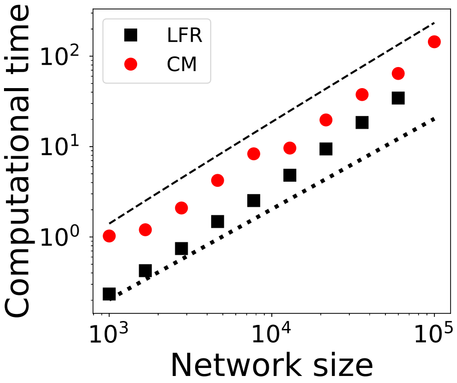

FIG. 3. Computational complexity of algorithms for multiplex reconstruction. We generate multiplex networks with variable size N. Tests are performed on synthetic multiplex networks with preimposed community structure constructed according to the LFR model with $\gamma = 2.1$, $\tau = 1.0$, $k_{\max} = \sqrt{N}$, $\langle k \rangle = 5.0$, and $\mu = 0.1$. Tests are also performed on networks without community structure that were generated using the configuration model (CM) with $\gamma = 2.1$, $k_{\min} = 3$, and $k_{\max} = \sqrt{N}$. No correlation at the level of degree sequence and/or community structure is present between the layers of the multiplex. Given a network, we provide 50% of partial information to the degree- and community-based (DC) classifier and measure the time required by the algorithm to reconstruct the multiplex. Computational time is measured in seconds. Simulations were run on an Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz. As guidelines, we display lines denoting the scalings N (dotted) and $N \log N$ (dashed).

that the DC method is able to reconstruct multiplex networks in linear time is verified in Fig. 3. There we systematically apply the DC method in the reconstruction of the topology of multiplex networks composed of synthetic network layers of variable size. By construction, these networks are sparse, so the total number of edges is proportional to the total number of nodes. In the same spirit as in the analysis of Fig. 2, we use the DC classifier to generate solutions of the MRP for a given amount of partial information. Note that even networks that do not have preimposed community structure are still analyzed according to the full pipeline of the DC classifier. The quasilinear scaling of the time of computation is apparent from our results.

D. Applications to real-world multiplex networks

Finally, we study the MRP with partial edge information on real-world multiplex networks (see Table I for the complete list of data sets analyzed). In Fig. 4 we report results for four data sets. Results for three additional data sets can be found in Fig. 7. Although some data sets include information about multiplex networks with more than two layers, we study only multiplexes formed by two layers at a time. Overall, we find a mixed behavior of the DC classifier. There are cases where the classifier is able to correctly identify the flavor of a large portion of edges even if the size of the training set is small. This fact happens for multiplex networks representing scientific collaborations [Fig. 4(a)] and genetic interactions in organisms [Figs. 4(b) and 7]. In other situations instead, for example, for a connectome multiplex network [Fig. 4(c)] and for a multiplex transportation network [Fig. 4(d)], the performance of the classifier is not exceptional.

We compare the performance of the DC classifier with a suitably modified version of the classifier by Wu *et al.* [30] (see Sec. IV for details). The classifier by Wu *et al.* relies on the configuration model and the classification of edges is obtained via the expectation-maximization algorithm. No information on the network community structure is used by the method. The classifier by Wu *et al.* excels on the London multiplex transportation network [Fig. 4(d)]. It produces a performance comparable to the one of the DC classifier in all other multiplex networks, except for those characterized by layers with sufficiently strong but uncorrelated community structure, for example, the genetic interaction multiplex of the *Drosophila melanogaster* [Fig. 4(a)]. The strength of the community structure of a network layer is measured in terms of modularity; see Table I for modularity values of the best partitions for the various multiplex networks. In the same table we also report the normalized mutual information between the best partitions of the two layers, which provides us with a quantitative proxy to judge the level of similarity between pairwise community structures.

In Fig. 8 we replicate the analysis for real networks by finding communities with Infomap [42] rather than Louvain. Although the two community detection algorithms typically find different partitions of the network layers, the performance of the DC classifier is similar in the two cases, indicating that the use of assortative structure, irrespective of its details, enhances the reconstruction of a multiplex ground-truth topology.

III. CONCLUSION

In this paper we studied the multiplex reconstruction problem (MRP), i.e., the identification of the flavor of edges in multiplex networks composed of two layers.

First, we characterized facets of the reconstruction problem. We showed that the intrinsic difficulty of the MRP depends on the level of heterogeneity of the individual layers, in the sense that multiplex networks with broad degree distributions are easier to reconstruct than multiplex networks with homogeneous degree distributions. If layers are characterized by community structure, then the stronger such a structure is, the easier the MRP is. However, within-layer structural diversity is not the only important ingredient that determines the hardness of the MRP. The type and strength of structural correlation among layers are essential too. We showed that multiplex networks that have noncorrelated or anticorrelated layers, at the level of degree sequences and/or community structure, can be reconstructed quite well. On the other hand, positively correlated layers do not allow for an easy reconstruction.

Second, we presented an algorithm for solving the MRP. The method is inspired by the degree-corrected stochastic block model, in the sense that the probability of two nodes to be connected is proportional to the product of their degrees and a factor that accounts for the community structure of the network. In our experiments, the classifier is trained on a portion of edges whose ground-truth layer is revealed. Its performance is measured in terms of the ability of classification of the remaining edges whose ground-truth layer is not revealed. We systematically studied the performance of the classifier on both synthetic and real multiplex networks as a function of the relative size of the training set of edges. We found that the performance of the classifier saturates quite quickly as the amount of information used to train the classifier increases. Roughly, 30% of known edges are sufficient to let the classifier reach 90% performance. There are, however, also multiplex networks that are not easily reconstructed by our method.

Our main results are based on community partitions identified via the modularity maximization algorithm Louvain. However, we showed that a comparable performance can be achieved using Infomap, which determines community structure solving a different optimization problem. This means that assortative structure learned from partial observation is useful for the reconstruction task irrespective of its details.

The present work can be extended in multiple directions. The generalization of the MRP to multiplex networks composed of more than two layers is one of these directions. Considering mechanisms of aggregation different from the exclusive OR that we addressed in this paper is another potentially relevant direction. Also, the two main ingredients of the reconstruction algorithm developed in this paper can be used in more sophisticated, maybe more effective ways. For example, one could think of treating the MRP as a maximum likelihood problem where the network layers of the multiplex are fitted by degree-corrected stochastic block models and edges's flavors are treated as the tunable parameters of the fit. Such an extension could also involve the use of more sophisticated models such as the multilayer

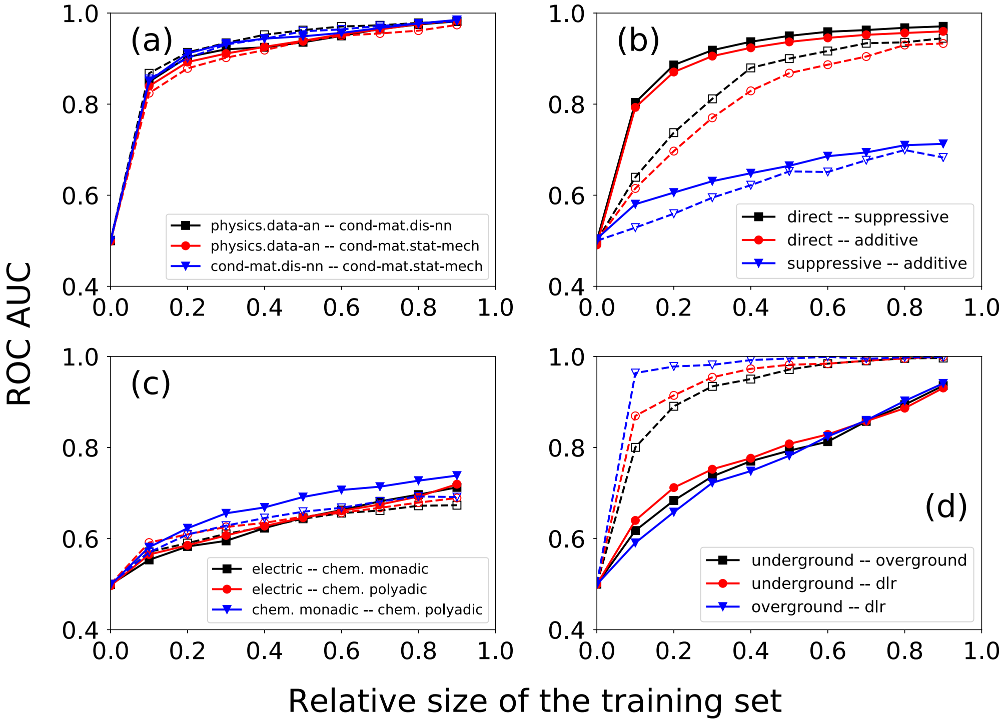

FIG. 4. Reconstruction of real-world multiplex networks. (a) Plot of the ROC AUC as a function of the relative size of the training set. We report results for the arXiv multiplex collaboration network [39]. We compare the results achieved with the degree- and community-based (DC, closed symbols and solid curves) classifier with those of the classifier of Wu *et al.* (open symbols and dashed curves) [30]. Results are averaged over ten realizations. (b) Same as in (a) but for the genetic interactions multiplex network of the *Drosophila melanogaster* [4]. Results are averaged over 100 realizations. (c) Same as in (a) but for the *Caenorhabditis elegans* multiplex connectome [40]. (d) Same as in (a) but for the London multiplex transportation network [41]. Results are averaged over 100 realizations.

degree-corrected stochastic block model [34]. Potential solutions of the MRP could be then obtained by means of likelihood maximization using standard optimization techniques such as the expectation-maximization algorithm or simulated annealing.

IV. METHODS

A. Multiplex reconstruction problem

We consider multiplex networks composed of two layers, namely, α and β. The graph representing layer α is denoted by $\mathcal{G}^{(\alpha)} = \{\mathcal{N}^{(\alpha)}, \mathcal{E}^{(\alpha)}\}$, where $\mathcal{N}^{(\alpha)}$ is the set of nodes of the layer and $\mathcal{E}^{(\alpha)}$ is the set of its edges. The same notation is used for layer β. Being a multiplex, we assume that $\mathcal{N}^{(\alpha)} = \mathcal{N}^{(\beta)}$ and we indicate the size of the network with $N = |\mathcal{N}^{(\alpha)}| = |\mathcal{N}^{(\beta)}|$. Further, we exclude the possibility that the edge (i, j) belongs simultaneously to both layers, meaning that $\mathcal{E}^{(\alpha)} \cap \mathcal{E}^{(\beta)} = \emptyset$. This choice is mainly dictated by simplicity. We stress, however, that ours is a quite reasonable assumption. All synthetic networks considered in our analysis are indeed characterized by a negligible number of edges shared by the layers. The same observation can be made also for several real-world networks studied in this paper (see Table I).

We indicate the degree sequence of the layer α with $\vec{k}^{(\alpha)} = (k_1^{(\alpha)}, k_2^{(\alpha)}, \dots, k_N^{(\alpha)})$, where $k_i^{(\alpha)}$ is the degree of node i in layer α defined as

$$k_i^{(\alpha)} = \sum_{(r,s)\in\mathcal{E}^{(\alpha)}} \delta(i, r) + \delta(i, s). \qquad (1)$$

In the above expression, $\delta(x, y)$ is the Kronecker delta function, i.e., $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ if $x \neq y$. A similar expression is used to relate the degree sequence $\vec{k}^{(\beta)}$ to the set $\mathcal{E}^{(\beta)}$.

The community memberships of the nodes in the two layers are denoted by $\vec{\sigma}^{(\alpha)}$ and $\vec{\sigma}^{(\beta)}$, respectively. Community structure is either *a priori* known based on the generative network model of the multiplex layers or inferred by the Louvain [38] or the Infomap [42] algorithm. Potentially, other community detection algorithms can be used as well.

We define the multiplex reconstruction problem as the binary classification of the individual edges in the multiplex layers, i.e., predicting whether the generic edge (i, j) belongs to either $\mathcal{E}^{(\alpha)}$ or $\mathcal{E}^{(\beta)}$. We perform the classification using partial knowledge of the ground-truth multiplex network. We consider two experimental setups.

(i) We assume to have full knowledge about some of the features of the individual nodes, i.e., degrees and/or community memberships, but no direct information about the sets $\mathcal{E}^{(\alpha)}$ and $\mathcal{E}^{(\beta)}$. We train a classifier using the available information and apply it to the classification of all edges $(i, j) \in \mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}$.

(ii) We assume to have information about a portion of the network edges, namely, the training sets $\mathcal{E}^{(\alpha)}_{\text{train}} \subseteq \mathcal{E}^{(\alpha)}$ and $\mathcal{E}^{(\beta)}_{\text{train}} \subseteq \mathcal{E}^{(\beta)}$, respectively. We train a classifier on these sets and we use it to classify all edges $(i, j) \in \mathcal{E}^{(\alpha)}_{\text{test}} \cup \mathcal{E}^{(\beta)}_{\text{test}}$, with $\mathcal{E}^{(\alpha)}_{\text{test}} = \mathcal{E}^{(\alpha)} \setminus \mathcal{E}^{(\alpha)}_{\text{train}}$ and $\mathcal{E}^{(\beta)}_{\text{test}} = \mathcal{E}^{(\beta)} \setminus \mathcal{E}^{(\beta)}_{\text{train}}$.

In both the above experimental setups, we measure the performance of the classifier using the ROC AUC, i.e., a standard metric in binary classification tasks.

B. Degree-based classifier

In the first experimental setup, we assume to have complete knowledge of the degree sequences $\vec{k}^{(\alpha)}$ and $\vec{k}^{(\beta)}$. We use a straightforward prediction model, where we pretend that the networks of layers α and β are random networks generated according to the configuration model with prescribed degree sequences $\vec{k}^{(\alpha)}$ and $\vec{k}^{(\alpha)}$, respectively [36]. Note that the network topology of each layer is assumed to be generated independently. According to our classifier, the edge $(i, j) \in \mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}$ belongs to layer α with probability

$$P[(i,j) \in \mathcal{E}^{(\alpha)} | \vec{k}^{(\alpha)}, \vec{k}^{(\beta)}] = \frac{k_i^{(\alpha)} k_j^{(\alpha)}}{k_i^{(\alpha)} k_j^{(\alpha)} + k_i^{(\beta)} k_j^{(\beta)}}. \tag{2}$$

Clearly, we have that

$$P[(i,j) \in \mathcal{E}^{(\beta)} | \vec{k}^{(\alpha)}, \vec{k}^{(\beta)}] = 1 - P[(i,j) \in \mathcal{E}^{(\alpha)} | \vec{k}^{(\alpha)}, \vec{k}^{(\beta)}]. \tag{3}$$

In the second experimental setup, we first use Eq. (1) to estimate the degree sequences $\vec{k}^{(\alpha)}_{\text{train}}$ and $\vec{k}^{(\beta)}_{\text{train}}$ from the sets $(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\beta)}_{\text{train}}$ and $(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\alpha)}_{\text{train}}$, respectively. Essentially, we pretend that a layer is formed by all edges in the aggregate except those edges that we know for sure belong to the other layer. Then we apply the classifier of Eq. (2), where $\vec{k}^{(\alpha)}_{\text{train}}$ and $\vec{k}^{(\beta)}_{\text{train}}$ replace the unknown ground-truth vectors $\vec{k}^{(\alpha)}$ and $\vec{k}^{(\beta)}$, respectively, to the edges in the test set $\mathcal{E}^{(\alpha)}_{\text{test}} \cup \mathcal{E}^{(\beta)}_{\text{test}}$.

C. Community-based classifier

In the first experimental setup, we assume to have complete knowledge of the community memberships $\vec{\sigma}^{(\alpha)}$ and $\vec{\sigma}^{(\beta)}$. We use a straightforward prediction model, where we pretend that the networks of layers α and β are random networks generated according to the homogeneous stochastic block model with prescribed community memberships $\vec{\sigma}^{(\alpha)}$ and $\vec{\sigma}^{(\alpha)}$, respectively. Note that the network topology of each layer is assumed to be generated independently. According to our classifier, the edge $(i, j) \in \mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}$ belongs to layer α with probability

$$P[(i,j) \in \mathcal{E}^{(\alpha)} | \vec{\sigma}^{(\alpha)}, \vec{\sigma}^{(\beta)}] = \frac{C(\sigma_i^{(\alpha)}, \sigma_j^{(\alpha)})}{C(\sigma_i^{(\alpha)}, \sigma_j^{(\alpha)}) + C(\sigma_i^{(\beta)}, \sigma_j^{(\beta)})}. \tag{4}$$

Clearly, we have that

$$P[(i,j) \in \mathcal{E}^{(\beta)} | \vec{\sigma}^{(\alpha)}, \vec{\sigma}^{(\beta)}] = 1 - P[(i,j) \in \mathcal{E}^{(\alpha)} | \vec{\sigma}^{(\alpha)}, \vec{\sigma}^{(\beta)}]. \tag{5}$$

In Eq. (4), $C(\sigma_i, \sigma_j)$ represents the propensity that two nodes with community memberships σ_i and σ_j are connected. For simplicity, we assume that such a propensity can be written as

$$C(\sigma_i, \sigma_j) = \nu[1 - \delta(\sigma_i, \sigma_j)] + (1 - \nu)\delta(\sigma_i, \sigma_j), \tag{6}$$

with $0 \leqslant \nu \leqslant 1$. Note that the value of the parameter ν is known to the classifier. The assumption of Eq. (6) is similar to the one underlying the Lancichinetti-Fortunato-Radicchi model [37], with the caveat $\nu \simeq \mu$, where μ is the mixing parameter of the LFR model.

In the second experimental setup, we apply a community detection algorithm to the network formed by the edges in the set $(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\beta)}_{\text{train}}$ to infer the community structure $\vec{\sigma}^{(\alpha)}_{\text{train}}$ of layer α. A similar procedure is used to infer the vector $\vec{\sigma}^{(\beta)}_{\text{train}}$. Finally, we use the classifier of Eq. (4), where $\vec{\sigma}^{(\alpha)}_{\text{train}}$ and $\vec{\sigma}^{(\beta)}_{\text{train}}$ are used in place of $\vec{\sigma}^{(\alpha)}$ and $\vec{\sigma}^{(\beta)}$, respectively. The parameter ν of Eq. (6) is estimated as

$$\nu_{\text{train}} = 1 - \frac{R^{(\alpha)}_{\text{train}} + R^{(\beta)}_{\text{train}}}{\left|(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\beta)}_{\text{train}}\right| + \left|(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\alpha)}_{\text{train}}\right|}, \tag{7}$$

where

$$R^{(\alpha)}_{\text{train}} = \sum_{(i,j) \in (\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\beta)}_{\text{train}}} \delta\big((\vec{\sigma}^{(\alpha)}_{\text{train}})_i, (\vec{\sigma}^{(\alpha)}_{\text{train}})_j\big).$$

A similar definition is used for $R^{(\beta)}_{\text{train}}$.

It is important to note that the goal of the classifier is not to infer meaningful community structure in the layers of the multiplex. Rather, we are solely concerned with leveraging assortative group structure in the reconstruction process. Therefore, the above choice of using Louvain (or Infomap) is a pragmatic one; Louvain (or Infomap) is a fast algorithm to discover assortative communities. If the communities found are spurious, then this fact will be accounted for in the parameter learned in Eq. (7), which in turn will generate scores in Eq. (6) that are similar for intra- and intercommunity edges.

D. Degree- and community-based classifier

In the first experimental setup, we assume to have complete knowledge of the degree sequences $\vec{k}^{(\alpha)}$ and $\vec{k}^{(\beta)}$ of the network layers and of the community memberships $\vec{\sigma}^{(\alpha)}$ and $\vec{\sigma}^{(\beta)}$. We define a score that is inspired by the edge connection probability in the degree-corrected stochastic block model [35]. Also in this case, we assume that the network topology of each layer is generated independently of the other. According to this degree- and community-based classifier, the edge $(i, j) \in \mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}$ belongs to layer α with probability

$$P[(i,j) \in \mathcal{E}^{(\alpha)} | \vec{k}^{(\alpha)}, \vec{k}^{(\beta)}, \vec{\sigma}^{(\alpha)}, \vec{\sigma}^{(\beta)}] = Q k_i^{(\alpha)} k_j^{(\alpha)} C(\sigma_i^{(\alpha)}, \sigma_j^{(\alpha)}). \tag{8}$$

In the above equation, Q is a normalization constant so that

$$\begin{aligned} &P[(i,j) \in \mathcal{E}^{(\beta)} | \vec{k}^{(\alpha)}, \vec{k}^{(\beta)}, \vec{\sigma}^{(\alpha)}, \vec{\sigma}^{(\beta)}] \\ &\quad = 1 - P[(i,j) \in \mathcal{E}^{(\alpha)} | \vec{k}^{(\alpha)}, \vec{k}^{(\beta)}, \vec{\sigma}^{(\alpha)}, \vec{\sigma}^{(\beta)}]. \end{aligned} \tag{9}$$

In Eq. (8), $C(\sigma_i, \sigma_j)$ represents the propensity that two nodes with community memberships σ_i and σ_j are connected. For simplicity, we assume that such a propensity can be written as in Eq. (6) and that the value of the parameter ν is known.

In the second experimental setup, we pretend that layer α is formed by all edges in the set $(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\beta)}_{\text{train}}$ and layer β is composed of all edges in the set $(\mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}) \setminus \mathcal{E}^{(\alpha)}_{\text{train}}$. We then use Eq. (1) to estimate the degree sequences $\vec{k}^{(\alpha)}_{\text{train}}$ and $\vec{k}^{(\beta)}_{\text{train}}$ and a community detection algorithm to infer the layer community structures $\vec{\sigma}^{(\alpha)}_{\text{train}}$ and $\vec{\sigma}^{(\beta)}_{\text{train}}$. Finally, we use the classifier of Eq. (8), where $\vec{k}^{(\alpha)}_{\text{train}}$ and $\vec{k}^{(\beta)}_{\text{train}}$ replace the unknown ground-truth vectors $\vec{k}^{(\alpha)}$ and $\vec{k}^{(\beta)}$, respectively, and $\vec{\sigma}^{(\alpha)}_{\text{train}}$ and $\vec{\sigma}^{(\beta)}_{\text{train}}$ are used in place of $\vec{\sigma}^{(\alpha)}$ and $\vec{\sigma}^{(\beta)}$, respectively. The parameter ν_{train} of Eq. (6) is estimated with Eq. (7).

E. Modified classifier by Wu *et al.*

The original algorithm by Wu *et al.* is conceived to solve a more complicated problem than the one considered here [30]. Specifically, it aims at predicting the existence of an edge, and if the edge exists, it aims at classifying the flavor of the edge. In the classification part of the problem, an edge can belong to one layer, the other, or both of them.

We modify the algorithm of Wu *et al.* to solve the classification problem considered in this paper. We use this classifier only in the second experimental setup, where partial information on the multiplex edges is provided with the training sets $\mathcal{E}^{(\alpha)}_{\text{train}}$ and $\mathcal{E}^{(\beta)}_{\text{train}}$. As in its original formulation, also here the classifier takes advantage of the expectation-maximization algorithm. First, we define the known degree of node i in layer α as

$$z_i^{(\alpha)} = \sum_{(r,s)\in\mathcal{E}^{(\alpha)}_{\text{train}}} \delta(i,r) + \delta(i,s). \tag{10}$$

A similar equation is used to define $z_i^{(\beta)}$. Then we initialize $P[(i,j)\in\mathcal{E}^{(\alpha)}_{\text{test}}|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}] = P[(i,j)\in\mathcal{E}^{(\beta)}_{\text{test}}|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}] = \frac{1}{2}$ for all edges in the test set. We then iterate the equations

$$\kappa_i^{(\alpha)} = z_i^{(\alpha)} + \sum_{(r,s)\in\mathcal{E}^{(\alpha)}_{\text{test}}} [\delta(i,r)+\delta(i,s)]P\big[(r,s)\in\mathcal{E}^{(\alpha)}_{\text{test}}\big|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}\big] \tag{11}$$

and

$$P\big[(i,j)\in\mathcal{E}^{(\alpha)}_{\text{test}}\big|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}\big] = \frac{\kappa_i^{(\alpha)}\kappa_j^{(\alpha)}}{\kappa_i^{(\alpha)}\kappa_j^{(\alpha)} + \kappa_i^{(\beta)}\kappa_j^{(\beta)}}. \tag{12}$$

We use an expression similar to Eq. (11) for $\kappa_i^{(\beta)}$. Note that $P[(i,j)\in\mathcal{E}^{(\alpha)}_{\text{test}}|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}] = 1 - P[(i,j)\in\mathcal{E}^{(\beta)}_{\text{test}}|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}]$. At the end of each iteration, we check for convergence by comparing the predicted values of the probabilities $P[(i,j)\in\mathcal{E}^{(\alpha)}_{\text{test}}|\vec{\kappa}^{(\alpha)},\vec{\kappa}^{(\beta)}]$ with those predicted in the previous iteration of the algorithm. In our tests, the condition for convergence is satisfied if the difference between consecutive values of the estimated probabilities is smaller than $\epsilon = 10^{-3}$ for all edges in the test set.

F. Multiplex networks

1. Synthetic multiplex networks

We begin by generating a degree sequence from the power-law distribution $P(k) \sim k^{-\gamma}$ for $k \in [3, k_{\max}]$ and $P(k) = 0$ otherwise. As prescribed in Ref. [43], we set $k_{\max} = \sqrt{N}$ if $2 < \gamma \leqslant 3$ and $k_{\max} = N^{1/(\gamma-1)}$ for $\gamma > 3$. We consider various values of the degree exponent γ.

To generate positively correlated degree sequences, we sort the obtained degree sequence and use it for layer α, i.e., $k_1^{(\alpha)} \leqslant k_2^{(\alpha)} \leqslant \cdots \leqslant k_N^{(\alpha)}$. The degree sequence of layer β is a copy of the one of layer α, i.e., $k_i^{(\beta)} = k_i^{(\alpha)}$ for all $i = 1, \ldots, N$. This condition creates perfectly correlated degree sequences. To decrease correlation between the degree sequences of the layers, we swap the labels of random pairs of nodes in layer β. The level of correlation is dependent on the fraction of pairs that undergo swapping. If enough pairs of node labels are swapped, the two degree sequences are completely uncorrelated. The resulting degree sequences are then independently used to generate the network layers of the multiplex using the configuration model [36].

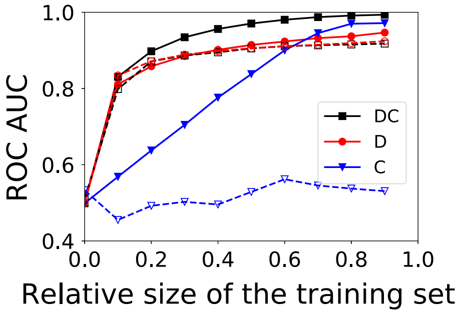

FIG. 5. Reconstruction of synthetic multiplex networks with partial edge information. The figure is the analog of Fig. 4(d) but for networks of larger size. We compare the performance of the different classifiers on synthetic graph with built-in community structure (closed symbols and solid curves) and without community structure (open symbols and dashed curves). We plot the performance metric of the degree- and community-based (DC) classifier, the degree-based (D) classifier, and the community-based (C) classifier. Graphs with community structure are constructed using the LFR multiplex model with $N = 100\,000$, $\gamma = 2.1$, $k_{\max} = \sqrt{N}$, $\langle k\rangle = 6$, $\tau = 1.0$, and $\mu = 0.1$. We use the configuration model with $N = 100\,000$, $\gamma = 2.1$, $k_{\min} = 3$, and $k_{\max} = \sqrt{N}$ for the graphs with no community structure. The degree sequences and community structures of the layers are uncorrelated.

To obtain anticorrelated degree sequences, we start from a sorted degree sequence for layer α. For layer β, we use $k_i^{(\beta)} = k_{N-i}^{(\alpha)}$ for all $i = 1, \ldots, N$. This fact ensures that the two degree sequences are maximally anticorrelated. Also, here we swap the labels of a certain fraction of pairs of nodes in layer β to decrease the level of correlation between the two degree sequences. The resulting degree sequences are then independently used to generate the network layers of the multiplex using the configuration model [36].

To study the effect of correlation between the layerwise community structures on the multiplex reconstruction problem, we generate network layers using the LFR model [37]. We begin by generating an instance of the LFR model with given set of parameters. We fix the value of the community size power-law exponent $\tau = 1.0$. We consider various values of the average degree $\langle k\rangle$. Also, we vary the degree exponent γ. We do not impose any constraint on the size and number of communities. We consider various values of the mixing parameter μ. The same network instance is used as the topology for both network layers α and β. To reduce the edge overlap between the layers to a negligible value without altering the correlation of the layers' community structure, we swap at random the labels of all pairs of nodes within the same community in layer β. To reduce the correlation among the structure of the network layers, we swap the labels for a random fraction of pairs of nodes in layer β.

2. Real multiplex networks

We analyze several real-world multiplex networks. Even if some data set consists of more than two layers, our tests are performed considering two layers at a time. For a given combination of the layers, edges shared by both layers are deleted and no information on their existence is considered in the MRP. For completeness, we report the number of edges shared by the layers of a multiplex network, namely, $|\mathcal{E}^{(\alpha,\beta)}|$ in Table I. These generally account for a very small number of edges compared to the total number of edges that are not shared by the layers. Specifically, the ratio $Z_{\text{shared}} = |\mathcal{E}^{(\alpha,\beta)}|/(|\mathcal{E}^{(\alpha)}| + |\mathcal{E}^{(\beta)}|)$ is always smaller than 0.150 except for two cases: (i) the layers "suppressive" and "additive" for the *Drosophila melanogaster* multiplex for which $Z_{\text{shared}} = 0.166$ and (ii) the layers "chemical monadic" and "chemical polyadic" of the *Caenorhabditis elegans* multiplex for which $Z_{\text{shared}} = 0.321$.

Then the set of nodes in the corresponding multiplex is given by the union of the sets of nodes of the two individual layers. Details on the data sets analyzed in the paper are reported in Table I. In the table we report also values of the modularity value for the best partition detected by the Louvain algorithm on each layer. We further measure the normalized mutual information between the layerwise partitions to assess their similarity.

All results of this paper are reproducible using code publicly available [44].

ACKNOWLEDGMENTS

S.P. and F.R. acknowledge support from the Army Research Office (Grant No. W911NF-21-1-0194); F.R. acknowledges support from the Air Force Office of Scientific Research (Grant No. FA9550-21-1-0446). The funders had no role in study design, data collection and analysis, decision to publish, or any opinions, findings, and conclusions or recommendations expressed in the paper.

APPENDIX

In Fig. 5 we repeat the analysis of Fig. 2(d) for networks with size $N = 10^5$. For the LFR model, we change the value of the average degree from $\langle k \rangle = 5$ to $\langle k \rangle = 6$ to bypass a convergence issue of the algorithm used to generate network instances. All other parameters are identical to those of Fig. 2(d).

In Fig. 6 we show that the performance of the proposed classifier improves as the size of the network increases, whereas it remains almost identical as the average degree of the network is varied.

The analysis of three additional real-world multiplex networks is summarized in Fig. 7.

In Fig. 8 we compare the performance of the DC algorithm based on either Louvain or Infomap. In all networks considered, we do not appreciate an apparent difference in performance due to the specific community detection algorithm leveraged.

Finally, in Fig. 9 we study the correlation between the performance of the DC classifier in the reconstruction of a given multiplex and the fraction of overlapping edges among the layers of the multiplex. We consider synthetic [Figs. 9(a) and 9(b)] and real-world [Figs. 9(c) and 9(d)] networks. In the case of real networks, we exclude on purpose the London transportation networks from the analysis because of their peculiarity of being graphs embedded in physical space. As the plot shows, there is a clear relationship between the value of the ROC AUC and the relative fraction of overlapping edges among the layers of the multiplex for synthetic systems. The value of the ROC AUC is inversely proportional to the fraction of edges shared between layers and disregarded by the classifier. This finding is also confirmed in real networks, although the dependence between edge overlap and reconstruction performance appears less apparent than in synthetic multiplexes.

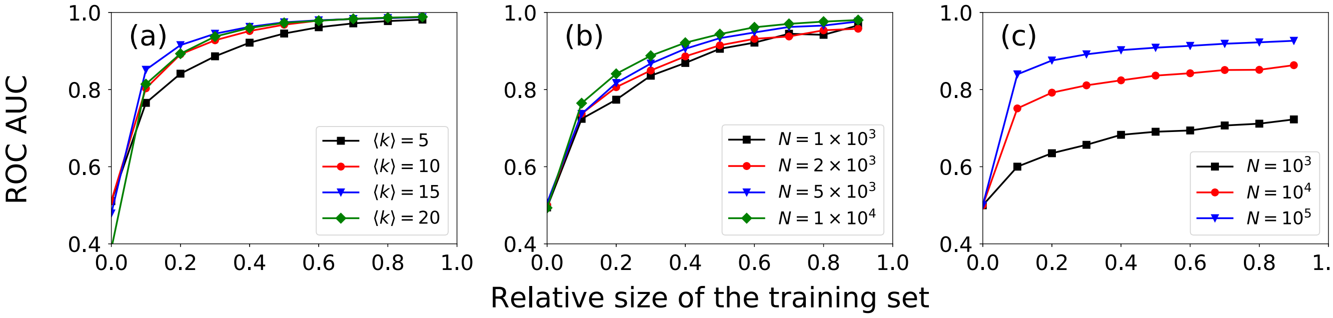

FIG. 6. Reconstruction of synthetic multiplex networks. (a) Plot of the ROC AUC of the degree- and community-based (DC) classifier as a function of the relative size of the training set. We consider multiplex networks with preimposed community structure with size $N = 10\,000$ and power-law degree distribution $P(k) \sim k^{-\gamma}$, with $\gamma = 2.1$. The strength of the community structure is determined by the mixing parameter $\mu = 0.1$. Communities have size obeying a power-law distribution with exponent $\tau = 1.0$. Different curves represent different values of average degree $\langle k \rangle$. The maximum degree is $k_{\max} = \sqrt{N}$. (b) Same as in (a) but for networks with average degree $\langle k \rangle = 5$. Different curves are obtained for different N values. (c) Plot of the ROC AUC of the degree-based (D) classifier as a function of the relative size of the training set. Networks are generated according to the configuration model with $\gamma = 2.1$, $k_{\min} = 3$, and $k_{\max} = \sqrt{N}$. Different curves are obtained for different N values.

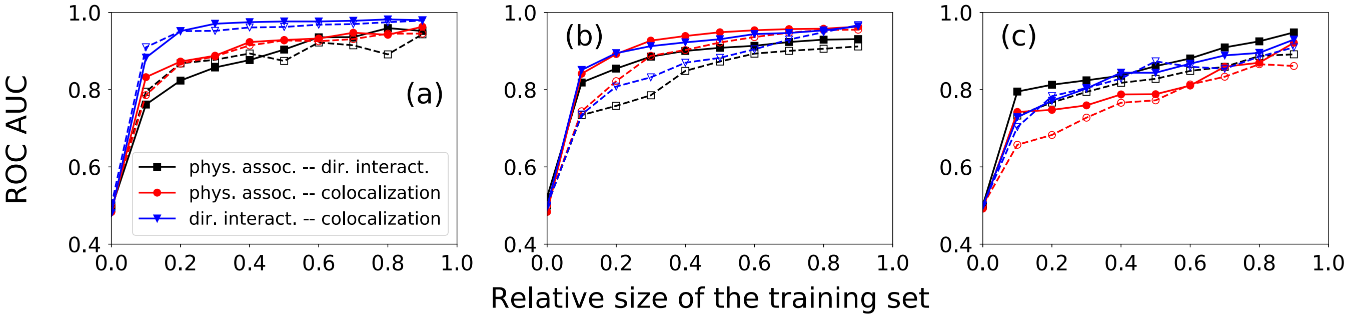

FIG. 7. Reconstruction of real-world multiplex networks. (a) Plot of the ROC AUC as a function of the relative size of the training set. We report results for the *Saccharomyces pombe* genetic multiplex network [4]. Results are averaged over ten realizations. We compare the results achieved with the degree- and community-based classifier (closed symbols and solid curves) with those of the classifier of Wu *et al.* (open symbols and dashed curves) [30]. Results are averaged over ten realizations. (b) Same as in (a) but for the genetic interactions multiplex network of the *Homo sapiens* [4]. Results are obtained over one realization. (c) Same as in (a) but for the genetic interactions multiplex network of the *Rattus norvegicus* [4]. Results are averaged over ten realizations.

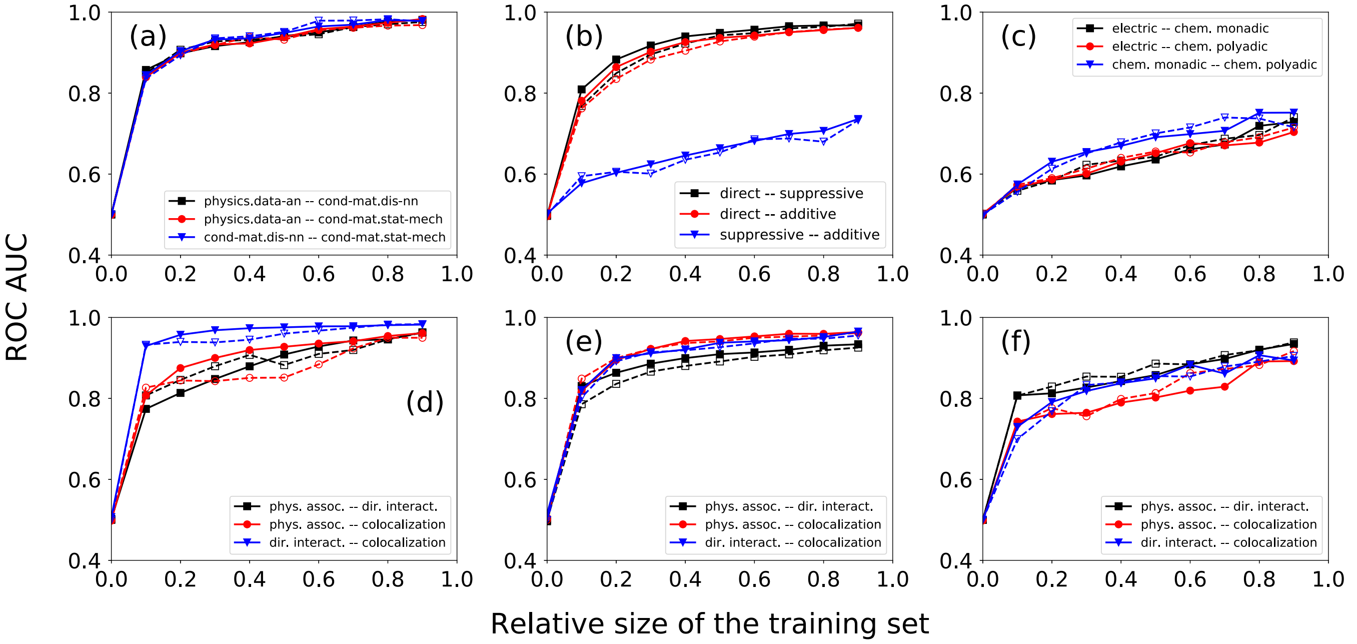

FIG. 8. Reconstruction of real-world multiplex networks. (a) Plot of the ROC AUC as a function of the relative size of the training set. We report results for the arXiv multiplex collaboration network [39]. We compare the results achieved with the degree- and community-based (DC) classifier relying on two different community detection algorithms: Louvain (closed symbols and solid curves) [38] and Infomap (open symbols and dashed curves) [42]. Results are averaged over ten realizations. (b) Same as in (a) but for the genetic interactions multiplex network of the *Drosophila melanogaster* [4]. Results are averaged over ten realizations. (c) Same as in (a) but for the *Caenorhabditis elegans* multiplex connectome [40]. (d) Same as in (a) but for the *Saccharomyces pombe* genetic multiplex network [4]. Results are averaged over ten realizations. (e) Same as in (a) but for the genetic interactions multiplex network of the *Homo sapiens* [4]. Results are obtained over one realization. (f) Same as in (a) but for the genetic interactions multiplex network of the *Rattus norvegicus* [4]. Results are averaged over ten realizations.

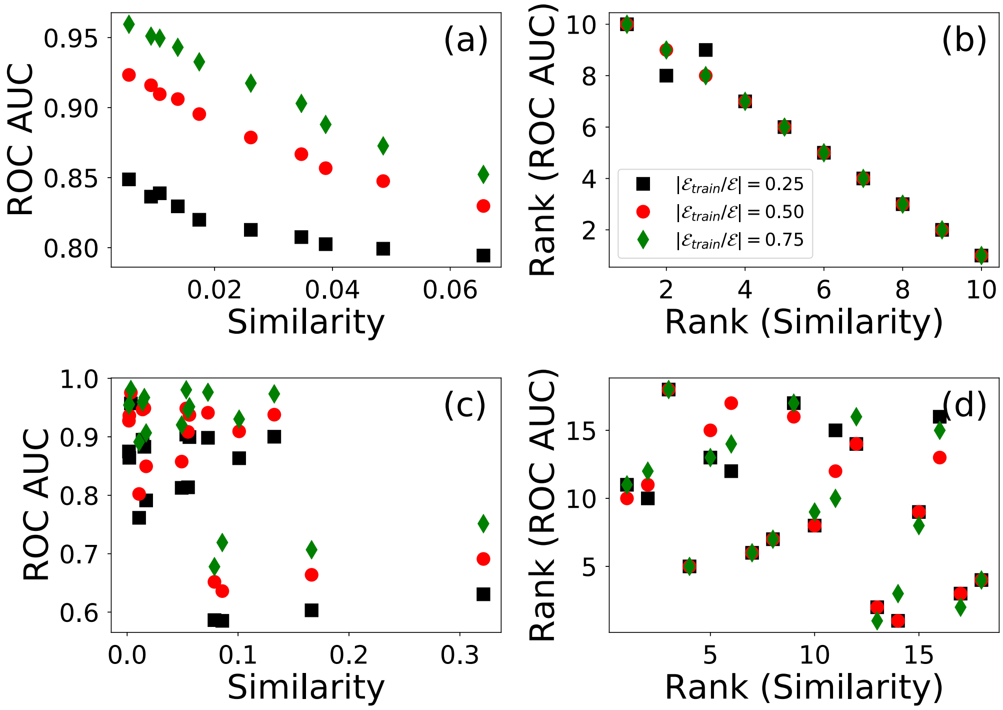

FIG. 9. Sensitivity analysis in the reconstruction of multiplex networks with edge overlap. (a) We consider LFR multiplex networks with $N = 10\,000$ nodes, average degree $\langle k \rangle = 10$, degree exponent $\gamma = 2.1$, maximum degree $k_{\max} = \sqrt{N} = 100$, community distribution exponent $\tau = 1$, mixing parameter $\mu = 0.1$, and minimum size of the communities $c_{\min} = 10$. The two layers are initially identical; then nodes of one of the layers are relabeled with probability q to control for the amount of shared edges. For $q = 0$, the two layers share all edges; for $q = 1$, the fraction of edges shared become minimal. We consider various values of the relabeling probability q. We report the values of the ROC AUC obtained by using our DC classifier as a function of the Jaccard similarity index of the set of edges of the two layers, i.e., $J = |\mathcal{E}^{(\alpha,\beta)}|/(|\mathcal{E}^{(\alpha)}| + |\mathcal{E}^{(\beta)}| + |\mathcal{E}^{(\alpha,\beta)}|)$. The different curves correspond to different choices of the relative size of the training set $|\mathcal{E}_{\text{train}}|/|\mathcal{E}|$, namely, (i) $|\mathcal{E}_{\text{train}}|/|\mathcal{E}| = 0.25$, (ii) $|\mathcal{E}_{\text{train}}|/|\mathcal{E}| = 0.50$, and (iii) $|\mathcal{E}_{\text{train}}|/|\mathcal{E}| = 0.75$, where $\mathcal{E} = \mathcal{E}^{(\alpha)} \cup \mathcal{E}^{(\beta)}$. We perform linear regression of the data points and obtain the following values of the Pearson correlation coefficient (p value): (i) $\rho = -0.93(0.00)$, (ii) $\rho = -0.99(0.00)$, and (iii) $\rho = -0.99(0.00)$. (b) Same as in (a), but we plot the rank position of the data according to the two metrics. We obtain the following values of the Spearman correlation coefficient (p value): (i) $\sigma = -0.99(0.00)$, (ii) $\sigma = -1.00(0.00)$, and (iii) $\sigma = -1.00(0.00)$. (c) Same as in (a) but for the real-world multiplex networks of Fig. 8. We measure (i) $\rho = -0.54(0.02)$, (ii) $\rho = -0.55(0.02)$, and (iii) $\rho = -0.56(0.02)$. (c) Same as in (b) but for the real-world multiplex networks of Fig. 8. We measure (i) $\sigma = -0.34(0.17)$, (ii) $\sigma = -0.44(0.07)$, and (iii) $\sigma = -0.42(0.08)$.

[1] M. E. J. Newman, Network structure from rich but noisy data, Nat. Phys. **14**, 542 (2018).

[2] G. Bianconi, *Multilayer Networks* (Oxford University Press, Oxford, 2018), Vol. 1.

[3] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, The structure and dynamics of multilayer networks, Phys. Rep. **544**, 1 (2014).

[4] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, Structural reducibility of multilayer networks, Nat. Commun. **6**, 6864 (2015).

[5] E. Vasilyeva, A. Kozlov, K. Alfaro-Bittner, D. Musatov, A. M. Raigorodskii, M. Perc, and S. Boccaletti, Multilayer representation of collaboration networks with higher-order interactions, Sci. Rep. **11**, 5666 (2021).

[6] J. Gao, A. Bashan, L. Shekhtman, and S. Havlin, *Introduction to Networks of Networks* (IOP, Bristol, 2022), pp. 2053–2563.

[7] F. Radicchi and A. Arenas, Abrupt transition in the structural formation of interconnected networks, Nat. Phys. **9**, 717 (2013).

[8] S. Gomez, A. Diaz-Guilera, J. Gomez-Gardenes, C. J. Perez-Vicente, Y. Moreno, and A. Arenas, Diffusion Dynamics on Multiplex Networks, Phys. Rev. Lett. **110**, 028701 (2013).

[9] F. Battiston, V. Nicosia, M. Chavez, and V. Latora, Multilayer motif analysis of brain networks, Chaos **27**, 047404 (2017).

[10] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, Catastrophic cascade of failures in interdependent networks, Nature (London) **464**, 1025 (2010).

[11] A. Cardillo, J. Gómez-Gardeñes, M. Zanin, M. Romance, D. Papo, F. del Pozo, and S. Boccaletti, Emergence of network features from multiplexity, Sci. Rep. **3**, 1344 (2013).

[12] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, and A. Arenas, Ranking in interconnected multilayer networks reveals versatile nodes, Nat. Commun. **6**, 6868 (2015).

[13] M. De Domenico, C. Granell, M. A. Porter, and A. Arenas, The physics of spreading processes in multilayer networks, Nat. Phys. **12**, 901 (2016).

[14] M. Diakonova, V. Nicosia, V. Latora, and M. S. Miguel, Irreducibility of multilayer network dynamics: The case of the voter model, New J. Phys. **18**, 023010 (2016).

[15] J. P. Gleeson, K. P. O'Sullivan, R. A. Baños, and Y. Moreno, Effects of Network Structure, Competition and Memory Time on Social Spreading Phenomena, Phys. Rev. X **6**, 021019 (2016).

[16] G. Zeng, D. Li, S. Guo, L. Gao, Z. Gao, H. E. Stanley, and S. Havlin, Switch between critical percolation modes in city traffic dynamics, Proc. Natl. Acad. Sci. USA **116**, 23 (2019).

[17] A. Santoro and V. Nicosia, Algorithmic Complexity of Multiplex Networks, Phys. Rev. X **10**, 021069 (2020).

[18] S. Osat, F. Radicchi, and F. Papadopoulos, *k*-core structure of real multiplex networks, Phys. Rev. Res. **2**, 023176 (2022).

[19] S. Osat, A. Faqeeh, and F. Radicchi, Optimal percolation on multiplex networks, Nat. Commun. **8**, 1540 (2017).

[20] K.-M. Lee, B. Min, and K.-I. Goh, Towards real-world complexity: An introduction to multiplex networks, Eur. Phys. J. B **88**, 48 (2015).

[21] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, and E. Estrada, Multilayer networks, J. Complex Netw. **2**, 203 (2014).

[22] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, Mathematical Formulation of Multilayer Networks, Phys. Rev. X **3**, 041022 (2013).

[23] M. Szell, R. Lambiotte, and S. Thurner, Multirelational organization of large-scale social networks in an online world, Proc. Natl. Acad. Sci. USA **107**, 13636 (2010).

[24] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, Community structure in time-dependent, multiscale, and multiplex networks, Science **328**, 876 (2010).

[25] E. Bullmore and O. Sporns, Complex brain networks: Graph theoretical analysis of structural and functional systems, Nat. Rev. Neurosci. **10**, 186 (2009).

[26] S. Lim, F. Radicchi, M. P. van den Heuvel, and O. Sporns, Discordant attributes of structural and functional brain connectivity in a two-layer multiplex network, Sci. Rep. **9**, 1 (2019).

[27] M. Zanin, Can we neglect the multi-layer structure of functional networks? Physica A **430**, 184 (2015).

[28] J. P. Bagrow and S. Lehmann, Recovering lost and absent information in temporal networks, arXiv:2107.10835.

[29] A. Zhang, A. Zeng, Y. Fan, and Z. Di, Detangling the multilayer structure from an aggregated network, New J. Phys. **23**, 073046 (2021).

[30] M. Wu, J. Chen, S. He, Y. Sun, S. Havlin, and J. Gao, Discrimination reveals reconstructability of multiplex networks from partial observations, Commun. Phys. **5**, 163 (2022).

[31] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, Community detection, link prediction, and layer interdependence in multilayer networks, Phys. Rev. E **95**, 042317 (2017).

[32] M. Tarrés-Deulofeu, A. Godoy-Lorite, R. Guimer, and M. Sales-Pardo, Tensorial and bipartite block models for link prediction in layered networks and temporal networks, Phys. Rev. E **99**, 032307 (2019).

[33] L. Lacasa, I. P. Mariño, J. Miguez, V. Nicosia, É. Roldán, A. Lisica, S. W. Grill, and J. Gómez-Gardeñes, Multiplex Decomposition of Non-Markovian Dynamics and the Hidden Layer Reconstruction Problem, Phys. Rev. X **8**, 031038 (2018).

[34] T. Vallès-Català, F. A. Massucci, R. Guimerà, and M. Sales-Pardo, Multilayer Stochastic Block Models Reveal the Multilayer Structure of Complex Networks, Phys. Rev. X **6**, 011036 (2016).

[35] B. Karrer and M. E. Newman, Stochastic blockmodels and community structure in networks, Phys. Rev. E **83**, 016107 (2011).

[36] M. Molloy and B. Reed, A critical point for random graphs with a given degree sequence, Random Struct. Alg. **6**, 161 (1995).

[37] A. Lancichinetti, S. Fortunato, and F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E **78**, 046110 (2008).

[38] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. (2008) P10008.

[39] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems, Phys. Rev. X **5**, 011027 (2015).

[40] M. De Domenico, M. A. Porter, and A. Arenas, MuxViz: A tool for multilayer analysis and visualization of networks, J. Complex Netw. **3**, 159 (2015).

[41] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas, Navigability of interconnected networks under random failures, Proc. Natl. Acad. Sci. USA **111**, 8351 (2014).

[42] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).

[43] M. Catanzaro, M. Boguná, and R. Pastor-Satorras, Generation of uncorrelated random scale-free networks, Phys. Rev. E **71**, 027103 (2005).

[44] https://github.com/kaiser-dan/parametric-dyadic-multiplex-reconstruction.