Annals of Operations Research
https://doi.org/10.1007/s10479-023-05419-x

ORIGINAL RESEARCH

Complex networks and deep learning for copper flow across countries

Lorenzo Federico[1] · Ayoub Mounim[1] · Pierpaolo D'Urso[2] · Livia De Giovanni[1]

Accepted: 22 May 2023

Abstract

In this paper, by using a lifecycle perspective, four stages related to the extraction, refining and processing of copper were identified. The different behaviors of countries in the import/export networks at the four stages synthetically reflect their position in the global network of copper production and consumption. The trade flows of four commodities related to the extraction, refining and processing of copper of 142 nations with population above 2 millions based on the UN Comtrade website (https://comtrade.un.org/data/), in five years from 2017 to 2021, were considered. The observed trade flows in each year have been modelled as a directed multilayer network. Then the countries have been grouped according to their structural equivalence in the international copper flow by using a Multilayer Stochastic Block Model. To put further insight in the obtained community structure of the countries, a deep learning model based on adapting the node2vec to a multilayer setting has been used to embed the countries in an Euclidean plane. To identify groups of nations that play the same role across time, some distances between the parameters obtained in consecutive years were introduced. We observe that 97 countries out of 142 consistently occupy the same position in the copper supply chain throughout the five years, while the other 45 move through different roles in the copper supply chain.

Keywords Social networks · Complex networks · Supply chain · Import/export networks · Trade flows · Community detection · Multilayer Stochastic block model · Deep learning

✉ Lorenzo Federico
lfederico@luiss.it

Ayoub Mounim
amounim@luiss.it

Pierpaolo D'Urso
pierpaolo.durso@uniroma1.it

Livia De Giovanni
ldegiovanni@luiss.it

[1] Department of Political Sciences and Data Lab, Luiss University - Viale Romania, 32-00197 Rome, Italy

[2] Department of Social and Economic Sciences, Sapienza - University of Rome, P.le Aldo Moro, 5-00185 Rome, Italy

Published online: 17 July 2023

1 Introduction

Large systems can be modelled and studied as complex networks. A complex network is defined by vertices (nodes) and edges. A "vertex" represents an element that has any kind of relationship with another element in the network, and this relationship is namely the "edge". For example, in a social network, a node could be an individual and an edge the friendship/acquaintance between a couple of individuals, or in air-traffic networks, a node could be an airport and an edge a flight route between two airports. In the present case, the node is a country and the edge a trade relation between countries with respect to a commodity. The complex networks theory offers many tools for analyzing a network (Caldarelli, 2007; Newman, 2003, 2018).

Applications of complex networks have appeared in the literature to study both artificial and natural systems. Beyond applications in computer simulation, complex networks have provided understanding of a number of subjects such as: Social Networks, the World Wide Web, human interactions, food webs, the spread of diseases, genomes and protein' structures. For a review of these applications, see Albert and Barabási (2002); Newman (2003).
A fast growing research branch in complex networks attains the detection of communities (for a review, see Newman (2018)). Detection of communities (groups, clusters) in systems represented as graphs (networks) is of great importance in many disciplines. A thorough exposition of the topic, the definition of the main elements of the problem, the presentation of most methods developed, the discussion of crucial issues can be found in Fortunato (2010).

One of the most interesting lines of research is the study of whether the network accommodates a community structure (Newman, 2006; Newman & Girvan, 2004).

On a network level a community structure is any partition of the vertices into a set of communities which enables defining, for each community, internal and external edges, that is edges connecting two vertices of the given community and edges connecting a node of the given community with a node of some other community. This definition still leaves open many possibilities, and there are correspondingly many computational approaches. Many algorithms have been proposed in the literature to perform such a task (Fortunato, 2010).

The most common approaches are based on optimization. A score is assigned to each possible partition of a network into communities such that 'good' divisions get high scores, then the partition with the highest score is chosen. There are a variety of ways to assign scores. The most popular approach makes use of the quality function known as modularity, which rewards partitions that have a high number of edges between nodes in the same cluster. The modularity is based on the difference between the edges observed between communities and the edges expected in a randomized version of the network (Girvan & Newman, 2002; Newman, 2006).

Other approaches are model-based, with a variety of proposals (Matias & Robin, 2014). In the model-based approach communities are not merely a feature of the network but a primary driver of it: vertices are connected precisely because of the groups they belong to. A 'good' community structure is one for which the model generates the observed network with high probability, so that the probability can be used as a score function for finding the best partition into communities. These models have the advantage of not requiring a prior definition of what a 'good' partition is, but are able to find any kind of similarity structure among the vertices that is statistically significant.

Quantitative analysis on international trade flows with social network analysis (Wasserman & Faust, 1994) has been addressed in the literature in testing various hypotheses on the structure and dynamics of the global economy, including the core peripheral structure of the world system, the new international division of labor, and the rise of semi-peripheral countries in several newly industrializing areas (Kim & Shin, 2002; Smith & White, 1992; Snyder & Kick, 1979). In Tong and Lifset (2007), with an example based on the data of Yale STAF project on the global copper cycle, community detection in social networks is used to depict the pattern of material movement through international trade. The main characteristics of the analysis are: (i) lifecycle perspective; and (ii) the combination of physical flows with intangible social and economic relations.

For a review on the use of complex networks to model material flows in manufacturing, logistic and supply chain see Bonaccorsi et al. (2019), Funke and Becker (2020) and related references.

For a review on the use of deep learning in operation research see Kraus et al. (2020).

An extended classical input–output analysis through a multilayer network has been considered in Cornaro and Rizzini (2022). In particular, a multilayer network where sectors of each economy are connected by weighted edges representing the directed embodied energy flows has been used to detect relevant sectors and economies in the whole system.

In this paper, as in Tong and Lifset (2007), by using a lifecycle perspective, four stages related to the extraction, refining and processing of copper were identified. The different behaviors of countries in the import/export networks at the four stages synthetically reflect their position in the global network of copper production and consumption. The trade flows of four commodities related to the extraction, refining and processing of copper of 142 nations with population above 2 millions based on the UN Comtrade website (https://comtrade.un.org/data/), in five years from 2017 to 2021, were considered. The observed trade flows in each year have been modelled as a directed multilayer network. Then the countries have been grouped according to their structural equivalence in the international copper flow by using a Multilayer Stochastic Block Model. To identify clusters of nations that play the same role across time, some distances between the parameters obtained in consecutive years were introduced. To put further insight in the obtained community structure of the countries, a deep learning model has been used to embed the countries in an Euclidean plane. Different embedding models based on deep learning have been discussed in the literature, however, the implementation on the case of a directed multilayer networks is, to the best of our knowledge, a novel contribution.

The paper is organized as follows. In Sect. 2 the Multilayer Stochastic Block Model is presented. In Sect. 3 an application of the Multilayer Stochastic Model for grouping countries according to their structural equivalence in the international copper flow is presented, along with the deep learning analysis. Conclusions and future outlooks are proposed in Sect. 5.

2 Community detection in complex networks

2.1 Stochastic block model

In the Stochastic Block Model (SBM) the probability that two vertices are connected depends solely on the communities they belong to (Karrer & Newman, 2011; Nowicki & Snijders, 2001). A ‘good’ community structure is one for which the model generates the observed network with high probability, so that the probability can be used as a score function for finding the best partition into communities. Statistical inference is used to estimate the parameters of the probabilistic model.

Consider a network with n vertices, labeled in $\{1, \dots, n\}$. In the SBM model the vertices are distributed among K communities or groups so that each vertex i is associated to a random vector $\mathbf{Z}_i = (Z_{i1}, \dots, Z_{iK})$, with Z_{ik} equal to 1 if vertex i belongs to community k and 0 otherwise. Since each vertex i can belong to only one community, we then have $\sum_{k=1}^{K} Z_{ik} = 1$. The vectors $\{\mathbf{Z}_i, i = 1, \dots, n\}$ are the rows of the matrix $\mathbf{Z}$ having dimension $n \times K$. The $\mathbf{Z}_i$ are supposed to be independent identically distributed random variables with a multinomial distribution:

$$\mathbf{Z_i} \sim \mathcal{M}(1, \boldsymbol{\alpha}) \tag{1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$ and $\sum_{k=1}^{K} \alpha_{ik} = 1$.
Each edge from a vertex i to a vertex j is associated to a random variable X_{ij}, representing the weight (strength) of the edge between vertex i and vertex j. The vectors $\{\mathbf{X}_i,\ i = 1, \dots, n\}$ are the rows of the adjacency matrix $\mathbf{X}$ having dimension $n \times n$. In undirected networks $X_{ij} = X_{ji}$. Specifically, conditionally on the community of each vertex, the edges and their weights are supposed to be independent. If community k of vertex i and community l of vertex j are known, X_{ij} is distributed as $f(\cdot, \theta_{kl}) := f_{\theta_{kl}}(\cdot)$, where $f_{\theta_{kl}}$ is a probability distribution known up to a finite dimensional parameter θ_{kl} depending only on the communities of the two end vertices of the edge:

$$X_{ij}|Z_{ik} = 1, Z_{jl} = 1 \sim f(\cdot, \theta_{kl}) := f_{\theta_{kl}}(\cdot) \tag{2}$$

In the SBM the edges with extremes in the same pair of communities (k, l) are independent and identically distributed, giving rise to a *block* in the adjacency matrix $\mathbf{X}$. The parameters $\{\theta_{kl},\ k, l = 1, \dots, K\}$ are arranged in the $(K \times K)$-dimensional matrix $\boldsymbol{\Theta}$.
For computational reasons in the estimation of the parameters, distributions of the exponential family are the most used for the X_{ij} (Mariadassou et al., 2010). In the case of a binary network, $X_{ij}|Z_{ik} = 1, Z_{jl} = 1 \sim \mathcal{B}(\theta_{kl})$ where $\mathcal{B}(\theta_{kl})$ denotes the Bernoulli distribution with parameter θ_{kl} $(0 \leq \theta_{kl} \leq 1)$. In the case of a weighted network the Poisson distribution or the Gaussian distribution are the most used.
The model is completely specified by the vector $\boldsymbol{\alpha}$ together with the matrix $\boldsymbol{\Theta}$. The parameters of the model are denoted by $\boldsymbol{\gamma} = (\boldsymbol{\alpha}, \boldsymbol{\Theta})$.
The estimation of the parameter $\boldsymbol{\gamma} = (\boldsymbol{\alpha}, \boldsymbol{\Theta})$ relies on the likelihood of the observed data.

The complete data is referred to as $(\mathbf{X}, \mathbf{Z})$, whereas $\mathbf{X}$ is referred to as the incomplete data. The decomposition $log\, \mathbb{P}(\mathbf{X}, \mathbf{Z}) = log\, \mathbb{P}(\mathbf{Z}) + log\, \mathbb{P}(\mathbf{X}|\mathbf{Z})$ holds true. From the independence of the row vectors $\mathbf{Z}_i$ of $\mathbf{Z}$ and the conditional independence of the edges X_{ij} knowing $\mathbf{Z}$ and from (1), (2) it then follows that the log-likelihood of the complete data is:

$$\begin{aligned} log\, L(\boldsymbol{\alpha}, \boldsymbol{\Theta}) &= log \prod_{i=1}^{n} \prod_{k=1}^{K} \alpha_k^{Z_{ik}} + log \prod_{i,j=1}^{n} \prod_{k,l=1}^{K} f_{\theta_{kl}}(X_{ij})^{Z_{ik}Z_{jl}} = \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} log\, \alpha_k + \sum_{i,j=1}^{n} \sum_{k,l=1}^{K} Z_{ik} Z_{jl} log\, f_{\theta_{kl}}(X_{ij}) \end{aligned} \tag{3}$$

The likelihood of the incomplete data (without knowing the labels $\mathbf{Z}$) can be obtained by summing $\mathbb{P}(\mathbf{X}, \mathbf{Z})$ over all possible realizations of the matrix $\mathbf{Z}$: $\mathbb{P}(\mathbf{X}) = \sum_{\mathbf{Z}} \mathbb{P}(\mathbf{X}, \mathbf{Z})$. This summation involves K^n terms (for each vertex i the K possible positions of the number 1 in the K-dimensional vector $\mathbf{Z}_i$), and thus may not be tractable except for small values of n. The likelihood of the incomplete data is maximized with respect to the model parameter $\boldsymbol{\gamma} = (\boldsymbol{\alpha}, \boldsymbol{\Theta})$ and the community assignments of the vertices $\mathbf{Z}$ *via* the variational extension of the expectation-maximization (EM) algorithm. The variational EM algorithm aims at optimizing a lower bound of the log-likelihood, alternating between the expectation with respect to $\mathbf{Z}$ (E-step) and the maximization with respect to $\boldsymbol{\gamma}$ (M-step). The convergence of the method towards the maximum likelihood estimates of the parameters is studied in Bickel et al. (2013) and Celisse et al. (2011).

The SBM model requires the number of communities (blocks) K as an input argument. The number of communities K can be obtained by using the Integrated Classification Likelihood (ICL) criterion tha relies on the computation of the integrated (over the model parameters $\boldsymbol{\gamma} = (\boldsymbol{\alpha}, \boldsymbol{\Theta})$) complete log-likelihood (ICL) in order to focus on inference on $\mathbf{Z}$ and K (Biernacki et al., 2000; Celisse et al., 2011; Côme & Latouche, 2015).

Degree-corrected versions have also been proposed to account for a strong degree heterogeneity inside blocks, displayed by real-world networks (Karrer & Newman, 2011). In the degree-corrected SBM a new set of parameters controlling the expected degree of each vertex is introduced.

2.2 Multilayer Stochastic block model

When several relationships of various types can occur jointly between vertices, the data are represented by multilayer (multiplex) networks. Multilayer networks refer to a collection of networks involving the same set of vertices, each network corresponding to a given type of interaction. The main objective in this paper is to group the vertices into communities sharing connection properties with other vertices of the multilayer network. We introduce the multilayer version of the SBM (Barbillon et al., 2017). Let $\mathbf{Z}$ be the $n \times K$-dimensional matrix identifying the belonging of each vertex i of n vertices to one of K communities. With the same notation in Sect. 2.1, $\mathbf{Z}_i$ are supposed to be independent identically distributed random variables over K communities with multinomial distribution with parameter $\boldsymbol{\alpha}$. Let $\{\mathbf{X}_1, \dots, \mathbf{X}_M\}$ be M directed networks (layers) relying on the same set of n vertices; it is necessary to add a subscript to each edge to take into account the layer: $\mathbf{X}_m = \{X_{ijm},\ i, j = 1, \dots, n\}$ is the square $(n \times n)$-dimensional adjacency matrix among the n vertices in layer m. The model assumes that conditionally on $\mathbf{Z}$ the random variables $\{X_{ijm},\ i, j = 1, \dots, n\}$ are independent, whereas the M random variables $\{X_{ijm},\ m = 1, \dots, M\}$ for the same edge among layers are dependent.
If community k of vertex i and community l of vertex j are known, the M-dimensional random variable $(X_{ij1}, \dots, X_{ijM})$ is distributed as $f(\cdot, \theta_{kl1}, \dots, \theta_{klM}) := f_{\theta_{kl1},\dots,\theta_{klM}}(\cdot)$, where $f_{\theta_{kl1},\dots,\theta_{klM}}$ is a probability distribution known up to a finite dimensional parameter $(\theta_{kl1}, \dots, \theta_{klM})$ depending only on the communities of the two end vertices of the edge. The parameters $\{\theta_{klm},\ k, l = 1, \dots, K, m = 1, \dots, M\}$ are arranged in the $(K \times K)$-dimensional matrices $\boldsymbol{\Theta}_1, \dots, \boldsymbol{\Theta}_M$.
In case of a binary network $(X_{ij1}, \dots, X_{ijM})|Z_{ik} = 1, Z_{jl} = 1$ could be a M-variate Bernoulli; in case of weighted networks M-variate Poisson or M-dimensional Gaussian.

Consistency results on the maximum likelihood estimators obtained by using a variational expectation-maximization procedure are proved in Barbillon et al. (2017). The number of communities is chosen by using the ICL criterion.

The optimization is performed thanks to a combination of greedy local search and a genetic algorithm.

In the Multinomial Stochastic Block Model the distribution of $(X_{ij1}, \dots, X_{ijM})$ conditionally on $\mathbf{Z}$ is multinomial (Côme et al., 2021; Côme & Jouvin, 2022):

$$(X_{ij1}, \dots, X_{ijM})|Z_{ik} = 1, Z_{jl} = 1 \sim M(L_{ij}, \theta_{kl1}, \dots, \theta_{klM}) \tag{4}$$

with $L_{ij} = \sum_{m=1}^{M} X_{ijm}$ is computed from the observed data $\{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$. This specific formulation of a multilayer stochastic blockmodel reduces significantly the number of parameters.

3 Complex networks and copper flow across countries

Our case study in this paper is the analysis of the flows of the global copper supply chain over 5 years from 2017 to 2021. In particular, we want to use complex network theory to classify countries based on the role they play in the copper supply chain. We have chosen specifically copper as a commodity whose supply chain to study as it was already analysed in detail by Tong and Lifset in Tong and Lifset (2007) with different network analysis tools so we have a benchmark to which we can compare our results. We downloaded all the trade data from the UN Comtrade website [45] from 2017 to 2021 related to the copper supply chain. In particular, we focused on 4 commodities related to the extraction, refining, and processing of copper, identified by their HS commodity codes [21] as follows:

- 260300, Copper ores and concentrates.
- 74, Copper and articles thereof.
- 7404, Copper waste and scrap.
- 841989, Machinery, plant, and laboratory equipment; for treating materials by change of temperature, other than for making hot drinks or cooking or heating food.

The need to account for multiple commodity types to represent the entire copper supply chain requires us to build a multilayer network, in which each layer corresponds to one of the 4 selected commodities. We also had to find a good way to model the data as a network in each layer, taking into account the huge differences in size and populations of the countries and dependencies in the dataset, which are reflected in trade flows that are in very different orders of magnitude. We decided to fix thresholds on the population of the countries and on the volume of the transactions above which they are included in the network. We consider the trade flows of the 142 sovereign nations with populations above 2 million. We do not consider all the dependencies and the smaller nations because they very rarely have large value transactions related to the copper supply chain and their inclusion would make the network exceedingly sparse. We also decide to model each layer as a simple network, and not to weigh the edges according to the value of the transaction, but only choose a threshold above which the transaction is considered important enough to be included in the network. This is necessary to prevent the model from over-fitting around the largest countries (China above all), whose trade volumes can be orders of magnitude larger than the average, and treat the transactions in the rest of the network as noise.

After all these considerations, we model the evolution of the global copper supply chain as a sequence of directed multilayer networks $G(t) = \{\mathbf{X}_m(t), m \in \{260300, 74, 7404, 841989\}\}$, each representing the status of the supply chain in each year $t \in \{2017, \ldots, 2021\}$, as follows:

- The vertex set $V(G(t))$ is always made up of the 142 nations we analyzed.
- For each couple of vertices (i, j) and layer m, $X_{ijm} = 1$ if during year t there was a transaction of value at least 1 million US dollars in which i exported commodity m to nation j and $X_{ijm} = 0$ otherwise.

In Fig. 1 we show a graphical representation of the adjacency matrices at time $t = 2017$.

For each multilayer directed network $G(t)$, we look for the best fit of a multinomial stochastic blockmodel using the `greed` R package with the function `MultSbm` Côme and

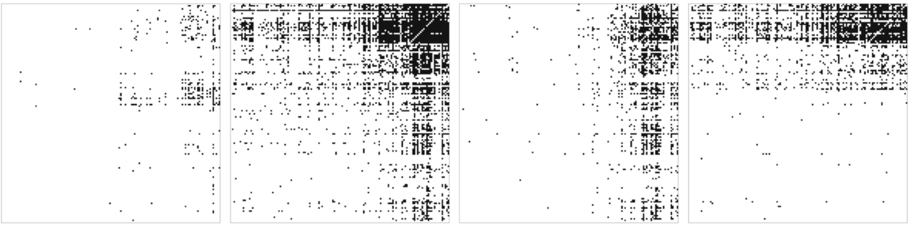


Fig. 1 Visualization of $\mathbf{X}_m(2017)$, $m \in \{260300, 74, 7404, 841989\}$. Black squares indicate that $X_{ijm=1}$

Jouvin (2022); Côme et al. (2021). As a result, for each year t we estimate the latent variables $\mathbf{Z}(t)$ and $\underline{\mathbf{\Theta}}(t) := \{\mathbf{\Theta}_1(t), \ldots, \mathbf{\Theta}_M(t)\} = \{\theta_{klm}(t), k, l = 1, \ldots, K, m = 1, \ldots, M\}$ as defined in Sect. 2.2. Our goal is not to directly compare the observed networks with those generated according to the best fit distribution of a multinomial stochastic blockmodel we found, but to use the assignment of nodes to communities represented by the latent partition $\mathbf{Z}(t)$ estimated by the function `MultSbm` to classify the nodes. We indeed observe that the real multilayer graphs $G(t), t = 2017, \ldots, 2021$ can look significantly different from those sampled from the probability distribution $P(\mathbf{X}_1, \ldots, \mathbf{X}_M | \mathbf{Z}(t), \underline{\mathbf{\Theta}}(t))$, generated using the latent variables estimated by `MultSbm`. This is the case both because the multinomial stochastic blockmodel produces multigraphs as its layers and because there might be extra randomness not explained by the model. Still, the algorithm when producing the best approximation of the observed data will group together nodes that have similar neighbours in the same layers, giving us valuable insight into the role they play in the copper supply chain.

3.1 Measuring the Matching of communities at different times

The `MultSbm` algorithm takes as input the observed tensor $\mathbf{X}(t) = \{\mathbf{X}_m(t), m = 1, \ldots, M\}$ of adjacency in the multilayer network $G(t)$ at a single time (see Fig. 1 for the adjacency in the year 2017, for the subsequent years, the plots are in Appendix A) and optimizes the values of the number of clusters K and the matrices $\mathbf{Z}(t)$ of cluster memberships and $\underline{\mathbf{\Theta}}(t)$ of estimated probabilities, as they are defined in Sect. 2.2 in order to maximize the ICL of the observed data $\mathbf{X}(t)$. They are presented in Table 5 (with names we will derive and describe later in this section) and Table 1, respectively, and the link densities (number of edges from cluster k to cluster l divided by the product of the sizes of k and l) in Fig. 2. Consequently, the `MultSbm` algorithm finds on its own coherent clusters of similar nodes for all layers at the same moment but does not provide a matching of the identities of the clusters at different times, as the parameters of the model for every multilayer network are estimated independently of the others. The `MultSbm` model outputs clusters that are just numbered from 1 to K, independently at each time, with no direct indication of how the clusters at different times relate to each other.

To understand better the evolution of the copper supply chain, the next important step is to find out whether it is possible to find some persistent properties in the data across the 5 years. The 5 multilayer networks are analysed separately and the parameters of the Multinomial Blockmodels are estimated independently of each other. What we want to understand is if it is possible to find *a posteriori* a persistent labeling of the clusters so that we can identify at each time which are the clusters of nodes which play the same role in the supply chain. What we see is that the nations are consistently classified in 5 clusters. Furthermore, we measure the Rand Index Campello (2007); Hubert and Arabie (1985) of partitions at different times

Table 1 The probability tensor $\underline{\mathbf{\Theta}}(2017)$

260300	1	2	3	4	5	74	1	2	3	4	5
1	0.052	0.008	0.027	0.011	0.002	1	0.379	0.453	0.459	0.470	0.493
2	0.017	0.000	0.000	0.087	0.000	2	0.494	0.548	0.617	0.478	0.631
3	0.158	0.000	0.037	0.167	0.023	3	0.401	0.517	0.543	0.389	0.591
4	0.260	0.120	0.286	0.647	0.000	4	0.413	0.460	0.452	0.235	0.893
5	0.044	0.000	0.086	0.353	0.014	5	0.533	0.608	0.571	0.529	0.768
7404	1	2	3	4	5	841989	1	2	3	4	5
1	0.240	0.157	0.032	0.054	0.009	1	0.329	0.382	0.481	0.465	0.496
2	0.304	0.277	0.043	0.087	0.071	2	0.185	0.176	0.340	0.348	0.298
3	0.228	0.242	0.074	0.083	0.017	3	0.213	0.242	0.346	0.361	0.369
4	0.310	0.360	0.167	0.059	0.036	4	0.017	0.060	0.095	0.059	0.071
5	0.418	0.380	0.286	0.118	0.070	5	0.005	0.012	0.057	0.000	0.148

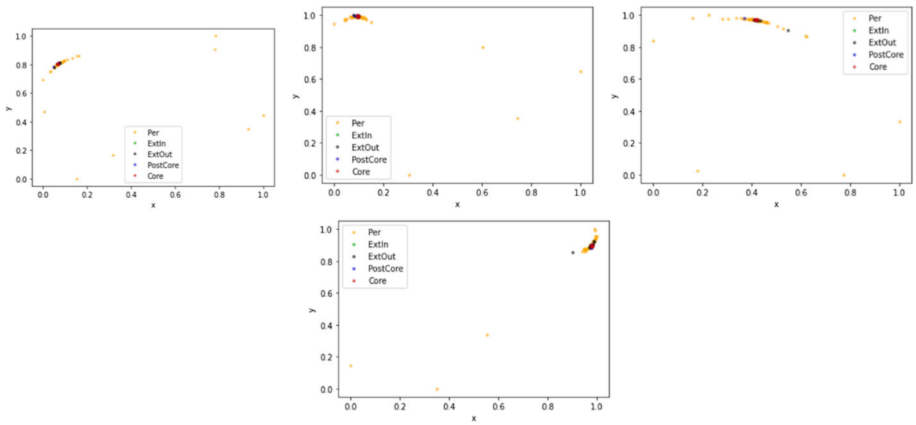


Fig. 2 The plots of the edge densities between the 5 different groups consistently identified by the `MultSbm`

and find that indeed they agree very strongly, in particular, if compared at consecutive times (Table 2). This suggests that even if we did not choose a model that actively favors consistent clustering at different times, in practice the data produce partitions of the vertices that agree substantially with each across all the 5 years considered, and in particular at consecutive times. Vertices that are in the same cluster at time t are extremely likely to also be in the same cluster at time $t + 1$ and still very likely to be in the same cluster at any time t'. Still, the Rand Index does not automatically find a way to consistently label clusters at different times so that clusters that represent similar properties are labeled the same (for example, we want to consistently label at all times the cluster of extractors of raw materials in the same way).

To define this labeling, we want to understand if the properties (as described by the model parameters $\underline{\mathbf{\Theta}}(t)$ (see e.g. Table 1 for $\underline{\mathbf{\Theta}}(2017)$, the others are in Appendix A) and the memberships as described by the matrices $\mathbf{Z}(t) = \{\mathbf{Z}_i(t), i = 1, \ldots, n\}$ (see Table 5 for the detailed membership at any time) of these clusters are somewhat inherited from one time to the next one. For example, we want to see if at any time there are identifiable clusters

Table 2 Rand Index between partitions

	2017	2018	2019	2020	2021
2017		0.86	0.85	0.86	0.85
2018			0,93	0.90	0.88
2019				0.95	0.89
2020					0.91
2021					

of countries that export raw materials (layer 260300). In particular, we recall that by the definition of the multinomial blockmodel the vectors $\boldsymbol{\theta}_{kl}(t) = \{\theta_{klm}(t), m = 1, \ldots, M\}$ are probability distributions. We want to find a consistent labeling such that if clusters k, l at time t and clusters k', l' at time $t + 1$ are labeled in the same way, then the total variation distance $d_{TV}\big(\boldsymbol{\theta}_{kl}(t), \boldsymbol{\theta}_{k'l'}(t + 1)\big)$ between the two probability distributions is small.

Formally, for each t we define the set $C(t)$ of clusters at time t, and we want to find a bijection $\sigma_{t,t+1} : C(t) \to C(t + 1)$ such that for every cluster c_{it}, we define $\sigma_{t,t+1}(c_{it})$ as the equivalent cluster to c_{it} at time $t + 1$. We will then give common labels at the sequences of clusters $(c_{it}, \sigma_{t,t+1}(c_{it}), \sigma_{t+1,t+2}(\sigma_{t,t+1}(c_{it})), \ldots)$ for all c_{it}. As at any time we have 5 clusters, the set of all the possible $\sigma_{t,t+1}$ is just the set S_5 of permutations of $\{1, 2, 3, 4, 5\}$, and consequently there are $5! = 120$ possible choices of $\sigma_{t,t+1}$. We will denote any $\sigma \in S_5$ by the sequence $(\sigma(1), \sigma(2), \sigma(3), \sigma(4), \sigma(5))$ (for example $(1, 2, 3, 4, 5)$ is the σ that matches any cluster at time t with the cluster with the same number at time $t + 1$, while $(5, 4, 3, 2, 1)$ is the σ that inverts the order of the clusters). To ground in quantitative analysis the choice of how to identify clusters of nations that play the same role in different years, we want to find a bijection $\sigma_{t,t+1}$ at any interval of time $[t, t + 1]$ that minimizes the difference between the parameters $\underline{\boldsymbol{\Theta}}(t)$ and $\mathbf{Z}(t)$ of the model at times t and $t+1$. To measure the difference between the parameters $\underline{\boldsymbol{\Theta}}(t)$, we define the following distance between the matrices of distributions $\underline{\boldsymbol{\Theta}}(t)$ and $\underline{\boldsymbol{\Theta}}(t + 1)$

$$d_1(\underline{\boldsymbol{\Theta}}(t), \underline{\boldsymbol{\Theta}}(t + 1), \sigma) = \Big(\sum_{k,l} d_{TV}\big(\boldsymbol{\theta}_{kl}(t), \boldsymbol{\theta}_{\sigma(k)\sigma(l)}(t + 1)\big)^2\Big)^{1/2}. \tag{5}$$

This is also known as the *product distance* Deza and Deza (2009) over the 25-fold product of the metric space of real-valued random variables equipped with the total variation distance. We want to minimize this distance to make sure that the probability distributions $\theta_{kl}(t)$ and $\theta_{\sigma(k)\sigma(l)}(t+1)$ of the layer to which a random edge between equivalent clusters ad consecutive times belongs are as similar as possible. In the same way, we define a distance between $\mathbf{Z}(t)$ and $\mathbf{Z}(t + 1)$ as

$$d_2(\mathbf{Z}(t), \mathbf{Z}(t + 1), \sigma) = \sum_{i,k} \frac{|Z_{ik}(t) - Z_{i\sigma(k)}(t + 1))|}{2}, \tag{6}$$

that is, $d_2(\mathbf{Z}(t), \mathbf{Z}(t + 1), \sigma)$ counts the number of nodes which are in cluster k at time t but not in cluster $\sigma(k)$ at time $t + 1$ for some k.

We find out that in our case the sequence of permutations:

$$(5, 4, 3, 2, 1), (1, 2, 3, 4, 5), (1, 2, 3, 4, 5), (1, 2, 3, 4, 5), \tag{7}$$

minimizes both $d_1(\underline{\boldsymbol{\Theta}}(t), \underline{\boldsymbol{\Theta}}(t + 1), \sigma)$ and $d_2(\mathbf{Z}(t), \mathbf{Z}(t + 1), \sigma)$ at all 4 time intervals, thus making it a natural choice for the matching of clusters at different times, without the need

Table 3 The 5 best choices of $\sigma_{t,t+1}$ at each time to minimize $d_2(\mathbf{Z}(t), \mathbf{Z}(t+1), \sigma)$

2017–2018	–	2018–2019	–	2019–2020	–	2020–2021	–
(5, 4, 3, 2, 1)	27	(1, 2, 3, 4, 5)	16	(1, 2, 3, 4, 5)	8	(1, 2, 3, 4, 5)	17
(5, 3, 4, 2, 1)	41	(1, 3, 2, 4, 5)	34	(1, 3, 2, 4, 5)	27	(1, 2, 5, 4, 3)	37
(5, 4, 2, 3, 1)	45	(1, 5, 3, 4, 2)	36	(1, 5, 3, 4, 2)	28	(1, 3, 2, 4, 5)	37
(5, 4, 1, 2, 3)	47	(1, 2, 5, 4, 3)	39	(1, 2, 5, 4, 3)	29	(1, 5, 3, 4, 2)	40
(5, 2, 3, 4, 1)	47	(1, 2, 4, 3, 5)	40	(1, 3, 5, 4, 2)	38	(1, 2, 3, 5, 4)	44

Table 4 The 5 best choices of $\sigma_{t,t+1}$ at each time to minimize $d_1(\underline{\boldsymbol{\Theta}}(t), \underline{\boldsymbol{\Theta}}(t+1), \sigma)$

2017–2018	–	2018–2019	–	2019–2020	–	2020–2021	–
(5, 4, 3, 2, 1)	0.542	(1, 2, 3, 4, 5)	0.316	(1, 2, 3, 4, 5)	0.262	(1, 2, 3, 4, 5)	0.357
(4, 5, 3, 2, 1)	0.737	(1, 2, 3, 5, 4)	0.593	(1, 2, 3, 5, 4)	0.595	(1, 2, 3, 5, 4)	0.650
(5, 3, 4, 2, 1)	0.771	(1, 2, 4, 3, 5)	0.651	(1, 2, 4, 3, 5)	0.692	(1, 2, 4, 3, 5)	0.668
(4, 3, 5, 2, 1)	0.866	(1, 2, 4, 5, 3)	0.743	(1, 2, 5, 4, 3)	0.763	(1, 2, 5, 4, 3)	0.764
(3, 4, 5, 2, 1)	0.879	(1, 2, 5, 4, 3)	0.747	(1, 2, 5, 3, 4)	0.802	(1, 2, 4, 5, 3)	0.786

to make a more complex decision based on a trade-off between the minimization of the two distances (Tables 3, 4).

We visualize the movement across the different clusters of the nodes in the alluvial plot in Fig. 3, the individual evolution of each country is presented in full detail in Table 5.

This consistency in the classification made us decide not to use more complex models for dynamic networks such as the `dynsbm` developed by Matias and Miele in Matias and Miele (2017) which would trade off some specificity in the identification of the multilayer structure of the network at any time in exchange for an automated matching between the communities found at two different instants. We finally label the consistent clusters throughout the 5 years according to the role that they play in the supply chain (Table 5):

- *Core*: Countries that import raw materials and take part in the entire processing chain. This is represented by cluster 1 in 2017 and 5 in the subsequent years.
- *PostCore*: Countries that do not participate in the initial extraction and processing but deal in processed copper. This is represented by cluster 2 in 2017 and 4 in the subsequent years.
- *ExtIn*: Countries that extract raw material and have a role in the entire processing chain. This is always represented by cluster 3.
- *ExtOut*: Countries that extract and export raw materials but do not participate in their refining and processing. This is represented by cluster 4 in 2017 and 2 in the subsequent years.
- *Per*: Countries that have a very minor role in the supply chain. This is represented by cluster 5 in 2017 and 1 in the subsequent years.

We notice that the `MultSbm` algorithm has a higher definition on the denser part of the network. Nodes with a very high degree are classified in a more specific way because their imprecise classification would cause a much larger decrease in the likelihood, while the algorithm tends to group the low-degree nodes in a single cluster that makes up roughly half of the entire vertex set (see Fig. 3 and Table 6).

Table 5 Clustering of the countries in each year

	Name	2017	2018	2019	2020	2021
1	Angola	ExtIn	Per	Per	Per	Per
2	Albania	Per	Per	Per	ExtOut	ExtOut
3	Argentina	ExtIn	ExtIn	ExtIn	ExtIn	ExtIn
4	Algeria	Per	Per	Per	Per	Per
5	Afghanistan	Per	Per	Per	Per	Per
6	Australia	ExtIn	ExtIn	ExtIn	ExtIn	ExtIn
7	Austria	Core	PostCore	PostCore	PostCore	PostCore
8	Bangladesh	Per	Per	Per	Per	Per
9	Armenia	ExtOut	ExtOut	ExtOut	ExtOut	ExtOut
10	Azerbaijan	Per	Per	Per	Per	ExtOut
11	Belgium	Core	Core	PostCore	PostCore	Core
12	Bosnia Herzegovina	Per	Per	Per	Per	Per
13	Bolivia	Per	Per	Per	Per	Per
14	Benin	Per	Per	Per	Per	Per
15	Belarus	Per	ExtIn	PostCore	PostCore	PostCore
16	Brazil	ExtIn	ExtIn	ExtIn	ExtIn	ExtIn
17	Bulgaria	Core	Core	Core	Core	Core
18	Botswana	Per	Per	Per	Per	ExtOut
19	Burkina Faso	Per	Per	Per	Per	Per
20	Burundi	Per	Per	Per	Per	Per
21	Canada	ExtIn	ExtIn	ExtIn	ExtIn	ExtIn
22	Cameroon	Per	Per	Per	Per	Per
23	Cambodia	ExtIn	Core	Core	Core	Core
24	Chad	Per	Per	Per	Per	Per
25	Central African Rep	Per	Per	Per	Per	Per
26	China	Core	Core	Core	Core	Core
27	Chile	ExtOut	ExtOut	ExtOut	ExtOut	ExtOut
28	Colombia	Per	Per	Per	Per	Per
29	Costa Rica	Per	Per	Per	Per	Per
30	Congo	Per	Per	Per	Per	Per
31	Croatia	PostCore	ExtIn	PostCore	PostCore	PostCore
32	Czechia	Core	PostCore	PostCore	PostCore	PostCore
33	Cuba	Per	Per	Per	Per	Per
34	Dominican Rep	Per	Per	Per	Per	Per
35	Côte d’Ivoire	Per	Per	Per	Per	Per
36	Ecuador	Per	Per	Per	Per	Per
37	Egypt	Per	Per	Per	Per	Per
38	El Salvador	Per	Per	Per	Per	Per
39	Ethiopia	Per	Per	Per	Per	ExtOut
40	Finland	Core	Core	Core	Core	Core

Table 5 continued

	Name	2017	2018	2019	2020	2021
41	France	Core	PostCore	PostCore	PostCore	PostCore
42	Gabon	Per	Per	Per	Per	Per
43	Georgia	ExtOut	ExtOut	ExtOut	ExtOut	ExtOut
44	Gambia	Per	Per	Per	Per	Per
45	Germany	Core	Core	Core	Core	Core
46	Greece	PostCore	PostCore	PostCore	PostCore	PostCore
47	Ghana	Per	Per	Per	Per	Per
48	Guatemala	Per	Per	Per	Per	Per
49	Guinea	Per	Per	Per	Per	Per
50	Indonesia	PostCore	ExtIn	ExtIn	ExtIn	ExtIn
51	India	Core	Core	Core	Core	Core
52	Hungary	Core	PostCore	PostCore	PostCore	PostCore
53	Haiti	Per	Per	Per	Per	Per
54	Honduras	Per	Per	Per	Per	Per
55	Italy	Core	PostCore	PostCore	PostCore	Core
56	Israel	PostCore	PostCore	PostCore	PostCore	PostCore
57	Iran	Per	ExtOut	Per	Per	Per
58	Ireland	Core	ExtIn	ExtIn	PostCore	PostCore
59	Iraq	Per	Per	Per	Per	Per
60	Japan	Core	Core	Core	Core	Core
61	Jordan	Per	Per	Per	Per	Per
62	Kenya	Per	Per	Per	Per	Per
63	Jamaica	Per	Per	Per	Per	Per
64	Kazakhstan	Per	ExtOut	ExtOut	ExtOut	ExtOut
65	Lebanon	Per	Per	Per	Per	Per
66	Lao People’s Dem. Rep	ExtOut	ExtOut	ExtOut	ExtOut	ExtOut
67	Kuwait	Per	Per	Per	Per	Per
68	Kyrgyzstan	Per	Per	Per	Per	Per
69	Lesotho	Per	Per	Per	Per	Per
70	Lithuania	PostCore	PostCore	PostCore	PostCore	PostCore
71	Libya	Per	Per	Per	Per	Per
72	Liberia	Per	Per	Per	Per	Per
73	Madagascar	Per	Per	Per	Per	Per
74	Malawi	Per	Per	Per	Per	Per
75	Malaysia	Core	Core	Core	Core	Core
76	Mexico	ExtIn	ExtIn	ExtIn	ExtIn	ExtIn

Table 5 continued

	Name	2017	2018	2019	2020	2021
77	Rep. of Moldova	ExtIn	ExtIn	PostCore	PostCore	PostCore
78	Mali	Per	Per	Per	Per	Per
79	Mauritania	Per	Per	Per	Per	Per
80	Morocco	ExtOut	Per	Per	Per	Per
81	Mozambique	Per	Per	Per	Per	Per
82	Namibia	ExtOut	ExtOut	ExtOut	ExtOut	ExtOut
83	Myanmar	Per	Per	Per	Per	Per
84	Mongolia	Per	ExtOut	ExtIn	Per	ExtOut
85	Netherlands	Core	PostCore	PostCore	PostCore	Core
86	New Zealand	PostCore	PostCore	PostCore	PostCore	PostCore
87	Nicaragua	Per	Per	Per	Per	Per
88	Nepal	PostCore	PostCore	PostCore	PostCore	PostCore
89	Niger	Per	Per	Per	Per	Per
90	Pakistan	PostCore	PostCore	PostCore	PostCore	PostCore
91	Nigeria	Per	Per	Per	Per	Per
92	Oman	ExtIn	Per	Per	Per	Per
93	Norway	PostCore	PostCore	PostCore	PostCore	PostCore
94	Panama	Per	Per	Per	Per	ExtOut
95	Papua New Guinea	ExtOut	ExtOut	Per	Per	ExtOut
96	Paraguay	Per	Per	Per	Per	Per
97	Peru	ExtOut	ExtOut	ExtOut	ExtOut	ExtOut
98	Poland	Core	Core	PostCore	Core	Core
99	Philippines	ExtOut	ExtIn	ExtIn	ExtIn	Core
100	Rep. of Korea	Core	Core	Core	Core	Core
101	Portugal	Core	ExtIn	PostCore	PostCore	Core
102	Romania	Core	Core	Core	PostCore	PostCore
103	Qatar	Per	Per	Per	Per	Per
104	Russian Federation	ExtIn	ExtIn	PostCore	PostCore	ExtIn
105	Saudi Arabia	Per	Per	Per	Per	Per
106	Serbia	PostCore	PostCore	ExtIn	ExtIn	ExtIn
107	Senegal	Per	Per	Per	Per	Per
108	Rwanda	Per	Per	Per	Per	Per
109	Sierra Leone	Per	Per	Per	Per	Per
110	Singapore	Core	Core	PostCore	PostCore	Core
111	Slovakia	PostCore	PostCore	PostCore	PostCore	PostCore
112	Slovenia	PostCore	PostCore	PostCore	PostCore	PostCore
113	South Africa	ExtIn	ExtIn	ExtIn	ExtIn	ExtIn
114	Somalia	Per	Per	Per	Per	Per

Table 5 continued

	Name	2017	2018	2019	2020	2021
115	Spain	Core	Core	Core	Core	Core
116	Sudan	Per	Per	Per	Per	Per
117	Sri Lanka	PostCore	Per	PostCore	PostCore	PostCore
118	State of Palestine	Per	Per	Per	Per	Per
119	Switzerland	Core	Core	PostCore	Core	PostCore
120	Thailand	PostCore	PostCore	PostCore	PostCore	Core
121	Sweden	Core	Core	Core	Core	Core
122	Tajikistan	ExtIn	Per	Per	Per	Per
123	Syria	Per	Per	Per	Per	Per
124	Turkey	Core	Core	PostCore	PostCore	Core
125	Tunisia	PostCore	Per	Per	Per	Per
126	Togo	Per	Per	Per	Per	Per
127	Uganda	Per	Per	Per	Per	Per
128	Turkmenistan	Per	Per	Per	Per	Per
129	United Kingdom	Core	PostCore	PostCore	PostCore	PostCore
130	Uruguay	Per	Per	Per	Per	Per
131	United Arab Emirates	PostCore	PostCore	PostCore	PostCore	PostCore
132	Ukraine	PostCore	PostCore	PostCore	PostCore	PostCore
133	United Rep. of Tanzania	Per	Per	Per	Per	Per
134	Viet Nam	PostCore	ExtIn	ExtIn	PostCore	PostCore
135	USA	Core	ExtIn	ExtIn	ExtIn	ExtIn
136	Venezuela	Per	Per	Per	Per	Per
137	Yemen	Per	Per	Per	Per	Per
138	Uzbekistan	Per	Per	ExtOut	ExtOut	ExtOut
139	Zambia	Per	ExtOut	ExtOut	ExtOut	ExtOut
140	Zimbabwe	Per	Per	Per	Per	ExtOut
141	Eritrea	ExtOut	Per	Per	ExtOut	ExtOut
142	South Sudan	Per	Per	Per	Per	Per

This shows that the Multinomial Blockmodel is very good at distinguishing the nodes in the network both on the basis of the level of activity and the specific role they play in the supply chain. We also tried to fit the `MultSbm` to the undirected version of the network, but this resulted, as one might expect, in the loss of a lot of valuable information regarding the distinction between extractors of raw material and producers of processed goods.

We observe that 97 countries out of 142 consistently occupy the same position in the supply chain throughout the 5 years, while the other 45 changed their role in the supply chain throughout the years considered. The only country to move to a new cluster every single year is Mongolia, this might be a result of the troubled development of the Mongolian extractive industry, which was mired in the last years by court cases involving both the national authorities and the main mining companies Ahlers et al. (2020). The maximum number of different clusters occupied by a nation is 3, for Mongolia, Portugal, Philippines, Ireland, and Belarus. We also have to remark that this study is about international copper

Table 6 Contingency tables between consecutive partitions

2017–2018 (12)	1	2	3	4	5	2018–2019 (23)	1	2	3	4	5
1	0	0	3	7	16	1	76	1	0	1	0
2	2	0	3	13	0	2	2	8	1	0	0
3	3	0	8	0	1	3	0	0	11	5	0
4	2	7	1	0	0	4	0	0	1	19	0
5	71	4	1	0	0	5	0	0	0	5	12
2019–2020 (34)	1	2	3	4	5	2020–2021 (45)	1	2	3	4	5
1	76	2	0	0	0	1	70	7	0	0	0
2	0	9	0	0	0	2	0	11	0	0	0
3	1	0	10	2	0	3	0	0	9	0	1
4	0	0	0	28	2	4	0	0	1	23	7
5	0	0	0	1	11	5	0	0	0	1	12

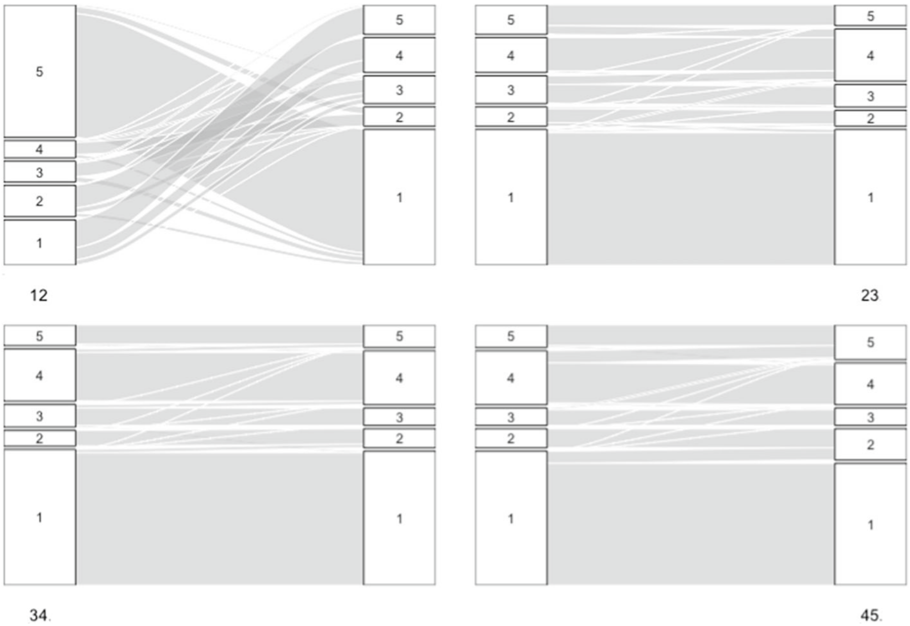


Fig. 3 Alluvial plots of the contingency tables between consecutive partitions (Table 6) (12: 2017–2018 23: 2018–2019 34:2019–2020 45:2020–2021). All individual changes are shown in Table 5

trade, not copper extraction and processing. Consequently, our model could put in peripheral clusters countries that extract, refine, and process copper mostly within their own internal market, with little international trade.

4 Deep learning: node embedding

In this section we apply a simple deep learning model to produce an embedding of the nodes of the supply-chain network studied in the previous sections into an Euclidean space. The main aim of this section is to use this technique to produce an alternative representation of the mutual relationship and interaction between the nodes of the network considered so far, in order to gain more qualitative insight and validation on the previous findings.

Different graph embedding models based on deep learning have been discussed in the literature before Grover and Leskovec (2016); Khosla et al. (2019a, b); Mikolov et al. (2013); Perozzi et al. (2014). The model we use here is a direct implementation of the well known *Skip-Gram* model Mikolov et al. (2013), originally developed in the field of Natural Language Processing (NLP), in the context of complex multilayer networks. Given a corpus text, intended as a collection of sentences, the Skip-Gram model is able to predict which words of the corpus will most likely be associated with a given input word. More precisely, define the "context" of a word as the set of n words preceding it and n words following it (the context is usually parameterized by the "window size" parameter $w = 2n + 1$). Given an input word taken from the corpus dictionary (the set of all words present in the corpus), the model learns a probability distribution over the whole corpus dictionary that models the probability that a word will be observed in the context of the input word in a sentence. The probability distribution is typically learned by some neural network architecture using the observed probability distributions on the text corpus as training data. In the training process of the model, latent vectors representation (of arbitrary dimension) of each word are learned through the hidden layers of the neural network architecture. An Euclidean space embedding can then be obtained quite simply by extracting the input's latent representations stored in the model's hidden layers.

The implementation of such model in the context of complex networks is made possible by the following correspondences:

- words $\Longleftrightarrow$ nodes of the graph,
- sentences $\Longleftrightarrow$ random walks on the graph,
- corpus $\Longleftrightarrow$ collection of random walks.

The equivalent of a corpus text is represented by a collection of sequences of adjacent nodes collected by performing random walks on the graph. The basic idea is that, just like the co-occurrence of words in sentences can be used to try to quantify the semantic similarity between words in a language, the co-occurrence of nodes in random walks is used to try to quantify structural similarities between the nodes of the network. We stress that, because the only input of the model is a collection of sequences of random walks on the network, the embedding produced by the model is only dependent on the topology of the graph, encoded in its adjacency matrix. Note that the ordered sequences of nodes that we interpret as sentences need not be necessarily coming from a random walk on the graph. In principle, any number of paths on the graph, collected in any given manner, could very well be used as a corpus to train the skip-gram model. One has to keep in mind however that, at an intuitive level, the more uniformly spread the paths are over the whole network, the more the embedded nodes' representations will be meaningful. Using random walks to collect such paths is the standard procedure in the literature simply because it is the most computationally efficient way of collecting a large number of paths while exploring the graph in a reasonably uniform manner.

4.1 Random walks

To produce an embedding of the graphs of interest we need to perform random walks on each one of the multilayered supply-chain networks $G(t) = \{\mathbf{X}_m(t), m \in \{260300, 74, 7404, 841989\}\}$ observed at the five different times $t = 2017, \ldots, 2021$. There are some subtleties in defining random walks on multilayered networks and the problem has been treated in the literature with different approaches Bonaccorsi et al. (2019); Halu et al. (2013); Iacovacci and Bianconi (2016); Iacovacci et al. (2016); Rahmede et al. (2018). Starting from the consideration that the four layers of the graph share the same set of nodes, a simple random walk on the multilayered network can be implemented as follows:

1. If M is the number of layers, fix a parameter α such that $\frac{1}{M} \leq \alpha \leq 1$.
2. Fix the length of the random walk L.
3. Choose one starting node $i \in V(G)$ and pick randomly one starting layer m.
4. Randomly choose one out-neighbor node j of the current node i in the graph $\mathbf{X}_m$ at the current layer. If i has no out-neighbours remain in i.
5. With probability α stay on the current layer, otherwise randomly choose a different layer and move to it (staying on the same node).
6. Go to step 4. and repeat until the length of the walk is equal to L.

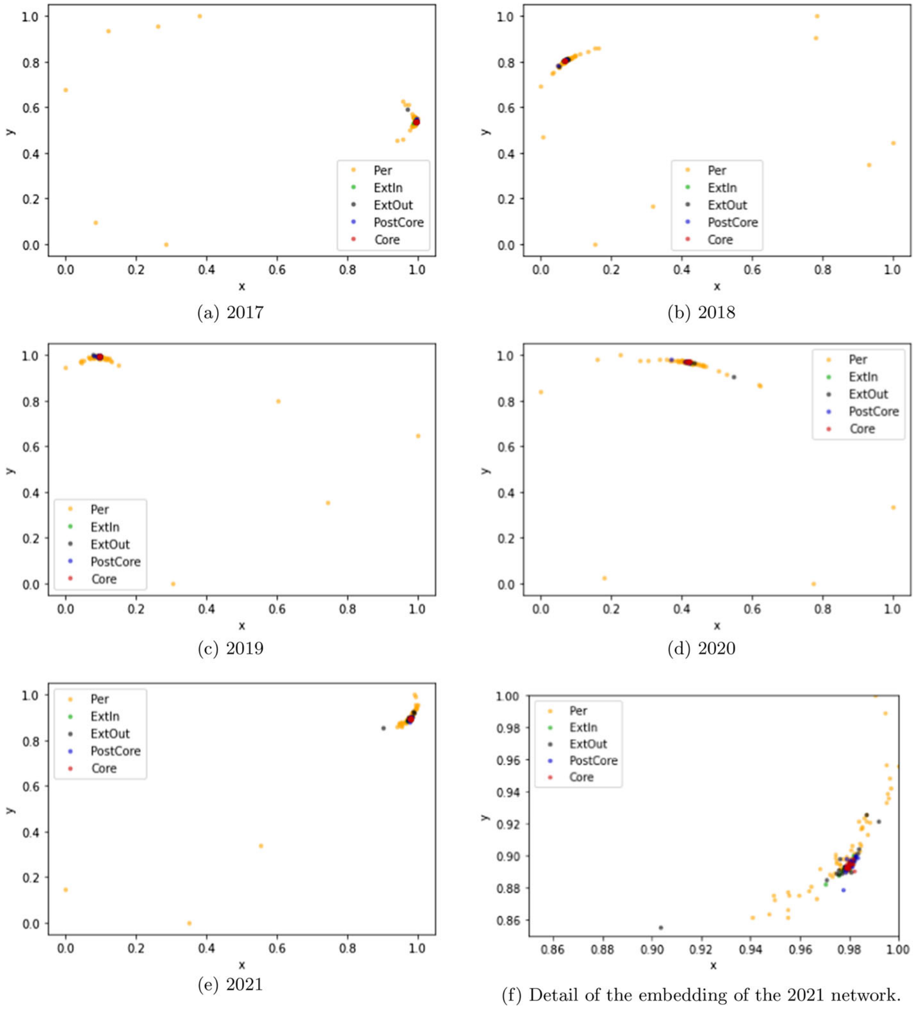
The intuition behind this random walker is that all layers are connected only through identical couples of nodes, and the strength of the connections is parameterized by the inertia parameter α, which is the probability that the walker will stay on the current layer at any given step. Notice that when $\alpha = 1$, the layers are actually disconnected and the walker will never change its current layer.

Following these rules, a random walk corpus is collected on each of the five multilayered networks. The random walks are collected by starting a single walk from each node of the network taken in some arbitrary order, the nodes' order is then randomly reshuffled and walks are started again from each node. This procedure is repeated γ times. The hyperparameter γ fixes the number of walks per node and consequently the size of the corpus used for training the model, which at the end of the collection procedure will be composed of $n \cdot \gamma$ walks, where n is the number of nodes in the network.

4.2 Embedding

Once the corpus of random walks is appropriately collected, it is then fed as the input into the Skip-Gram model so that a d-dimensional embedding of the graph's nodes can be produced as an output. Note that the data we feed into Skip-Gram contain information only about the sequence of nodes in $V(G)$ visited by the walker and not about on which layer the walker was moving at each time. In this paper we use the implementation of the model made available by the `gensim` Python library Řehůřek and Sojka (2010).

In order to discuss the relation between the learned embedding and the non-trivial structures found by the SBM, we embed the nodes in a 2-dimensional Euclidean space and we color the nodes according to the cluster membership of the nodes. As one could expect, the lower the dimensions of the embedding space are, the more information is lost in the embedding procedure, similarity to how information is lost when dimensional reduction algorithms such as principal component analysis are applied to datasets characterized by a high number of quantitative attributes Geiger and Kubin (2012). In the case of this work, the choice $d = 2$ is made for practical reasons, so that the vector representations of the nodes can be graphically represented as points in the plane. Also, as we will discuss later, the picture we



(a) 2017

(b) 2018

(c) 2019

(d) 2020

(e) 2021

(f) Detail of the embedding of the 2021 network.

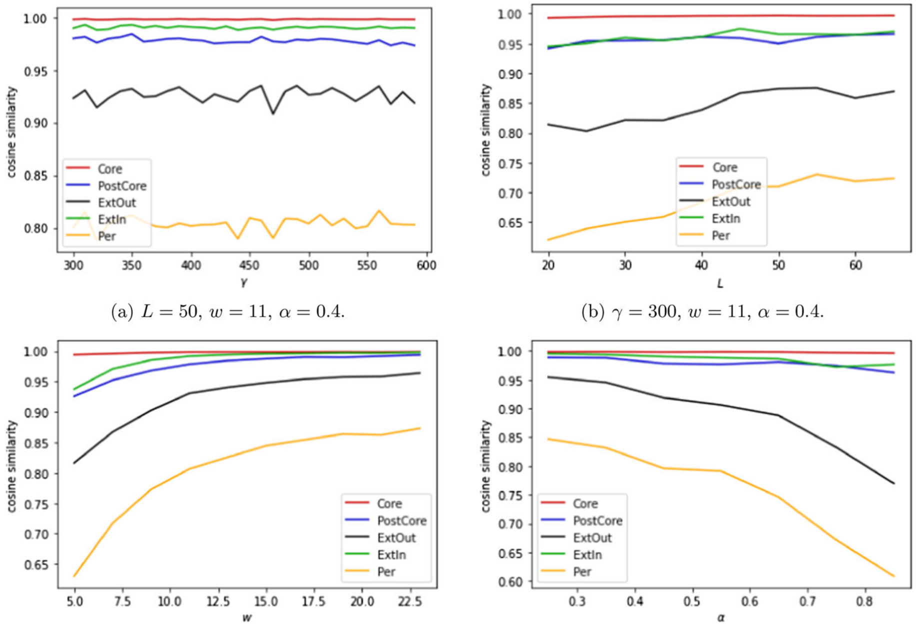
Fig. 4 Nodes embedding on the plane form the network observed at different times. Points are colored according to the clusters identified by the SBM analysis. The hyperparameters values used in the plots are: $\gamma = 300$, $L = 50$, $w = 13$, $\alpha = 0.55$

obtain via the 2-dimensional embedding is detailed enough for the purpose of this section. The embedding model we used is characterized by four hyperparameters

- γ : number of random walks per node,
- L : random walks length,
- w : window size used by the Skip-Gram model,
- α : random walker inertia.

A reasonable choice of parameters is made *a posteriori* by studying how the parameters affect the intra-clusters similarity score. In Fig. 4 we show the obtained nodes embedding for the multilayered networks observed at the five considered time steps.

The most relevant observation is that while the information on the non trivial clustering based on the “roles” of the nodes found by the SBM is lost, as one could have expected, the



(a) $L = 50,\ w = 11,\ \alpha = 0.4.$

(b) $\gamma = 300,\ w = 11,\ \alpha = 0.4.$

(c) $\gamma = 300,\ L = 50,\ \alpha = 0.4.$

(d) $\gamma = 300,\ L = 50,\ w = 11.$

Fig. 5 Average Intra-cluster similarity scores as functions of the embedding model's hyperparameters. All statistics in the figure are computed on the $\mathbf{X}_m$ (2021) network

information about the centrality of the nodes in the network is well represented by the planar embedding. This is particularly evident when looking at the embedding of the "core" cluster, by noticing that the nodes of the core are all embedded tightly together in a high density region of the plane, with all other nodes surrounding them. Another observation is that the nodes from the "Per" cluster that are isolated nodes in one or more of the network's layers are the same nodes that in the embeddings of Fig. 4a–e are significantly further apart from all other nodes, even from other "Per" nodes. The fact that the properties of the nodes of each cluster observed in the embedding are consistent for all 5 multilayer networks is further confirmation of the validity of the identification of clusters at subsequent times studied in Sect. 3.1.

To study how the hyperparameters affect the quality of the embedding, we measure the expected value of some similarity measure of points that belong to the same cluster and observe its behaviour when we change the hyperparameters. The similarity measure we use is cosine similarity, which is considered to be the measure of choice when comparing distances in the embedded space to distances in semantic space in the context of NLP. In Fig. 5 we plot the mean intra-cluster similarity measure for each cluster as a function of one of the hyperparameters.

Overall, the intra-cluster similarity is quite stable with respect to the hyperparameters range taken into consideration. This is especially true when considering the "core" cluster. This can be seen as indication that this kind of approach is very robust in identifying the most central nodes of the graph. The only hyperparameter that really a sizeable effect on the similarity scores is the window size parameter w. The behavior observed in Fig. 5c is a direct consequence of the meaning of the hyperparameter w. If one keeps increasing the window size, the probability of observing co-occurrences of any given set of connected nodes increases, and the nodes will naturally be interpreted by the model as more similar to each other. Interesting conclusions can also be drawn regarding the importance of the inertia hyperparameter α. Recalling the fact as $\alpha \to 1$ the random walkers tend to not jump between the network's layers, Fig. 5d shows that taking into account the multiple layers of the network is particularly relevant only for the "Per" and "ExtOut" clusters.

5 Conclusion

In this paper, a composite model to analyze the grouping of elements of a multilayer complex network across time has been proposed. The trade flows related to the copper supply chain - import/export of four commodities related to the extraction, refining and processing of copper - of 142 nations with populations above 2 million based on the UN Comtrade website (https://comtrade.un.org/data/), in five years from 2017 to 2021, were considered. The observed trade flow each year has been modeled as a directed multilayer network. The countries have been grouped according to their structural equivalence in the international copper flow by using a Multilayer Stochastic Block Model. To put further insight into the obtained community structure of the countries, a deep learning model has been used to embed the countries in a Euclidean plane. The Multilayer Stochastic Block Model allows the identification in each considered year of the countries that play similar roles in the trade flow of the four commodities. Still, even sampling independently a different multilayer SBM for each year, without the imposition of any temporal structure, we observed 5 clusters with similar features being found consistently in all 5 years. The consistent clusters throughout the 5 years were labeled according to the role that they play in the supply chain. We observe that 97 countries out of 142 consistently occupy the same position in the supply chain throughout the five years, while the other 45 move through different roles in the supply chain. Some distances between the parameters obtained in consecutive years were also introduced. A simple deep learning model has been used to produce an embedding of the nodes of the supply- networks into a Euclidean space. The most relevant observation is that while the information on the non-trivial clustering based on the "roles" of the nodes found by the multilayer SBM is lost, as one could have expected, the information about the centrality of the nodes in the network is well represented by the planar embedding. This is particularly evident when looking at the embedding of the "core" cluster, by noticing that the nodes of the core are all embedded tightly together in a high-density region of the plane, with all other nodes surrounding.

Future work will consider a way to design a clustering algorithm that takes into account both the multilayer structure of the network with the dynamic nature of the process and can handle networks with heavy-tailed weights, which are common if we want to take into consideration also the volume of trade flows. Also, of great interest would be further investigation of the relation between the embedding generated by the multilayer node2vec and the centrality of nodes in the network. We expect the property of having central nodes being clumped up together and peripheral nodes being at greater distances to be general properties in most network models, even beyond the unweighted, directed, multilayer networks considered in the current paper. This is because in most networks the random walks starting from very central nodes (for a reasonable definition of centrality) tend to *mix* very fast, that is, in very few steps the distribution of the position of the walker ceases to depend strongly from the starting point. Consequently, all random walks starting from very central nodes are very similar after very first steps and the nodes are embedded very close to each other by the Skip-Gram algorithm.

Funding Open access funding provided by Luiss University within the CRUI-CARE Agreement. The research of Livia De Giovanni, Lorenzo Federico and Ayoub Mounim is supported by the Horizon 2020 Framework Programme through the grant Media Futures: Data-driven innovation hub for the media value chain (Grant Number 951962), and the project CEF-TC-2020-2 European Digital Media Observatory/Italian Digital Media Observatory, European Commission (Agreement Number: INEA/CEF/ICT/A2020/2394428).

Declarations

Conflict of interest Lorenzo Federico declares that he has no conflict of interest, Ayoub Mounim declares that he has no conflict of interest, Pierpaolo D'Urso declares that he has no conflict of interest, Livia De Giovanni declares that she has no conflict of interest.

Ethical approval This article does not contain any study with human participants or animals performed by any of the authors.

A Other tables and figures

In this appendix, we show all the other tables and figures that we did not fit into the main body of the paper not to make the reading too heavy, but that are of interest to those who want to examine in more detail the data. The contingency tables between consecutive partitions (also represented in the alluvial plots in Fig. 3) are presented in Table 6, the tensors of probabilities $\underline{\boldsymbol{\Theta}}(t)$ for the years 2018-2021 are presented in Tables 7-10 and the adjacency matrices $\mathbf{X}_m(t)$ of the multilayer networks for years 2018-2021 are shown in Figs. 6, 7, 8, 9.

Table 7 The probability tensor $\underline{\boldsymbol{\Theta}}(2018)$

260300	1	2	3	4	5	74	1	2	3	4	5
1	0.000	0.348	0.015	0.008	0.058	1	0.735	0.522	0.575	0.556	0.505
2	0.042	0.350	0.250	0.055	0.327	2	0.812	0.400	0.515	0.699	0.440
3	0.014	0.136	0.072	0.007	0.182	3	0.560	0.409	0.441	0.503	0.365
4	0.007	0.013	0.000	0.004	0.019	4	0.549	0.480	0.497	0.448	0.411
5	0.004	0.025	0.020	0.014	0.078	5	0.520	0.458	0.470	0.430	0.378
7404	1	2	3	4	5	841989	1	2	3	4	5
1	0.122	0.087	0.396	0.402	0.432	1	0.143	0.043	0.015	0.034	0.005
2	0.062	0.050	0.191	0.233	0.220	2	0.083	0.200	0.044	0.014	0.012
3	0.032	0.061	0.144	0.233	0.240	3	0.394	0.394	0.343	0.257	0.212
4	0.020	0.013	0.108	0.271	0.311	4	0.424	0.493	0.395	0.277	0.259
5	0.016	0.008	0.070	0.207	0.227	5	0.459	0.508	0.439	0.349	0.316

Table 8 The probability tensor $\underline{\boldsymbol{\Theta}}(2019)$

260300	1	2	3	4	5	74	1	2	3	4	5
1	0.000	0.314	0.025	0.012	0.072	1	0.724	0.371	0.506	0.532	0.485
2	0.000	0.417	0.321	0.075	0.315	2	0.875	0.333	0.491	0.654	0.450
3	0.013	0.200	0.086	0.019	0.208	3	0.521	0.436	0.457	0.471	0.351
4	0.003	0.045	0.025	0.002	0.054	4	0.552	0.446	0.481	0.443	0.386
5	0.000	0.013	0.029	0.011	0.090	5	0.556	0.481	0.463	0.454	0.352
7404	1	2	3	4	5	841989	1	2	3	4	5
1	0.104	0.200	0.411	0.445	0.437	1	0.172	0.114	0.057	0.010	0.005
2	0.000	0.083	0.170	0.263	0.234	2	0.125	0.167	0.019	0.008	0.000
3	0.037	0.055	0.161	0.267	0.251	3	0.429	0.309	0.296	0.243	0.190
4	0.021	0.009	0.093	0.240	0.282	4	0.423	0.500	0.400	0.316	0.279
5	0.015	0.013	0.074	0.190	0.274	5	0.429	0.494	0.434	0.345	0.284

Table 9 The probability tensor $\underline{\boldsymbol{\Theta}}(2020)$

260300	1	2	3	4	5	74	1	2	3	4	5
1	0.000	0.310	0.030	0.004	0.075	1	0.801	0.448	0.489	0.527	0.488
2	0.000	0.550	0.275	0.058	0.341	2	0.917	0.350	0.451	0.645	0.437
3	0.014	0.167	0.094	0.012	0.239	3	0.530	0.405	0.432	0.456	0.337
4	0.002	0.029	0.012	0.001	0.050	4	0.576	0.429	0.463	0.459	0.402
5	0.000	0.075	0.022	0.004	0.069	5	0.554	0.441	0.473	0.450	0.367
7404	1	2	3	4	5	841989	1	2	3	4	5
1	0.118	0.172	0.437	0.445	0.432	1	0.081	0.069	0.044	0.024	0.005
2	0.028	0.050	0.176	0.290	0.222	2	0.056	0.050	0.098	0.007	0.000
3	0.041	0.095	0.151	0.273	0.261	3	0.416	0.333	0.324	0.258	0.163
4	0.015	0.010	0.092	0.236	0.284	4	0.408	0.533	0.433	0.304	0.264
5	0.014	0.000	0.089	0.191	0.259	5	0.432	0.484	0.415	0.355	0.306

Table 10 The probability tensor $\underline{\boldsymbol{\Theta}}(2021)$

260300	1	2	3	4	5	74	1	2	3	4	5
1	0.000	0.156	0.013	0.000	0.041	1	0.745	0.489	0.490	0.538	0.508
2	0.029	0.387	0.189	0.082	0.286	2	0.857	0.452	0.473	0.689	0.464
3	0.014	0.133	0.064	0.009	0.185	3	0.579	0.387	0.454	0.488	0.350
4	0.008	0.000	0.015	0.000	0.022	4	0.566	0.387	0.462	0.497	0.421
5	0.001	0.022	0.020	0.005	0.070	5	0.547	0.478	0.458	0.468	0.370
7404	1	2	3	4	5	841989	1	2	3	4	5
1	0.095	0.311	0.445	0.416	0.447	1	0.161	0.044	0.052	0.046	0.005
2	0.057	0.032	0.257	0.230	0.250	2	0.057	0.129	0.081	0.000	0.000
3	0.019	0.067	0.163	0.223	0.251	3	0.388	0.413	0.319	0.279	0.213
4	0.032	0.027	0.137	0.199	0.308	4	0.394	0.587	0.385	0.304	0.249
5	0.008	0.000	0.092	0.159	0.254	5	0.443	0.500	0.430	0.368	0.307

Table 11 The number of edges between communities in each layer in the year 2017

260300	1	2	3	4	5	74	1	2	3	4	5
1	82	6	11	2	2	1	603	350	185	87	485
2	10	0	0	2	0	2	286	103	58	11	89
3	52	0	3	6	4	3	132	62	44	14	104
4	63	6	12	11	0	4	100	23	19	4	25
5	36	0	6	6	2	5	437	104	40	9	109
7404	1	2	3	4	5	841989	1	2	3	4	5
1	382	121	13	10	9	1	523	295	194	86	488
2	176	52	4	2	10	2	107	33	32	8	42
3	75	29	6	3	3	3	70	29	28	13	65
4	75	18	7	1	1	4	4	3	4	1	2
5	343	65	20	2	10	5	4	2	4	0	21

Table 12 The number of edges between communities in each layer in the year 2018

260300	1	2	3	4	5	74	1	2	3	4	5
1	0	8	2	2	37	1	108	12	77	148	324
2	2	7	17	4	55	2	39	8	35	51	74
3	4	9	17	2	79	3	155	27	104	151	158
4	3	1	0	2	12	4	223	36	175	251	255
5	3	3	9	9	49	5	348	55	207	272	238
7404	1	2	3	4	5	841989	1	2	3	4	5
1	18	2	53	107	277	1	21	1	2	9	3
2	3	1	13	17	37	2	4	4	3	1	2
3	9	4	34	70	104	3	109	26	81	77	92
4	8	1	38	152	193	4	172	37	139	155	161
5	11	1	31	131	143	5	307	61	193	221	199

Table 13 The number of edges between communities in each layer in the year 2019

260300	1	2	3	4	5	74	1	2	3	4	5
1	0	11	4	6	42	1	97	13	80	258	282
2	0	10	17	10	35	2	35	8	26	87	50
3	3	11	16	7	58	3	125	24	85	176	98
4	2	5	11	2	34	4	338	50	208	552	245
5	0	1	7	7	28	5	263	38	112	279	109
7404	1	2	3	4	5	841989	1	2	3	4	5
1	14	7	65	216	254	1	23	4	9	5	3
2	0	2	9	35	26	2	5	4	1	1	0
3	9	3	30	100	70	3	103	17	55	91	53
4	13	1	40	299	179	4	259	56	173	394	177
5	7	1	18	117	85	5	203	39	105	212	88

Table 14 The number of edges between communities in each layer in the year 2020

260300	1	2	3	4	5	74	1	2	3	4	5
1	0	9	4	2	43	1	109	13	66	285	281
2	0	11	14	8	43	2	33	7	23	89	55
3	3	7	13	4	66	3	116	17	60	152	93
4	1	3	4	1	35	4	315	45	156	541	279
5	0	7	5	3	26	5	277	41	106	321	139
7404	1	2	3	4	5	841989	1	2	3	4	5
1	16	5	59	241	249	1	11	2	6	13	3
2	1	1	9	40	28	2	2	1	5	1	0
3	9	4	21	91	72	3	91	14	45	86	45
4	8	1	31	278	197	4	223	56	146	359	183
5	7	0	20	136	98	5	216	45	93	253	116

Table 15 The number of edges between communities in each layer in the year 2021

260300	1	2	3	4	5	74	1	2	3	4	5
1	0	7	2	0	32	1	102	22	76	106	400
2	1	12	14	5	63	2	30	14	35	42	102
3	3	10	9	2	73	3	121	29	64	105	138
4	2	0	4	0	16	4	142	29	121	273	311
5	1	4	7	4	63	5	394	87	164	347	334
7404	1	2	3	4	5	841989	1	2	3	4	5
1	13	14	69	82	352	1	22	2	8	9	4
2	2	1	19	14	55	2	2	4	6	0	0
3	4	5	23	48	99	3	81	31	45	60	84
4	8	2	36	109	227	4	99	44	101	167	184
5	6	0	33	118	229	5	319	91	154	273	277

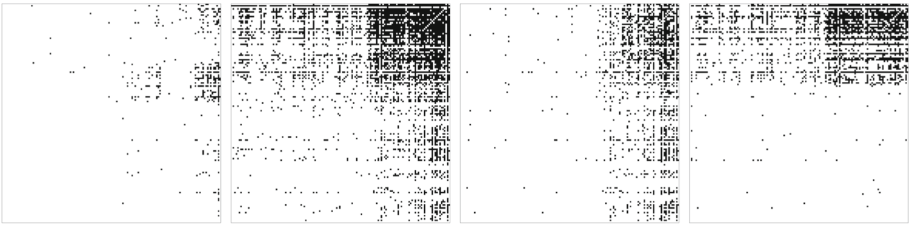


Fig. 6 Visualization of $\mathbf{X}_m(2018)$, $m \in \{260300, 74, 7404, 841989\}$

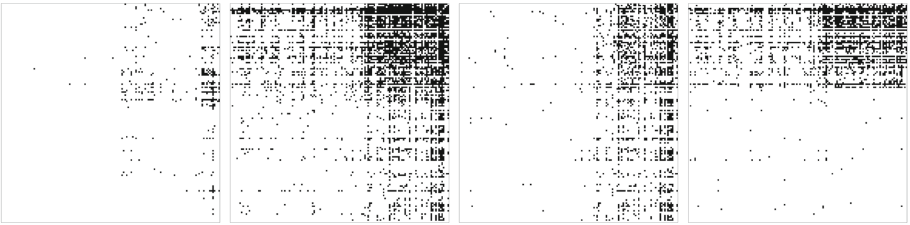


Fig. 7 Visualization of $\mathbf{X}_m(2019)$, $m \in \{260300, 74, 7404, 841989\}$

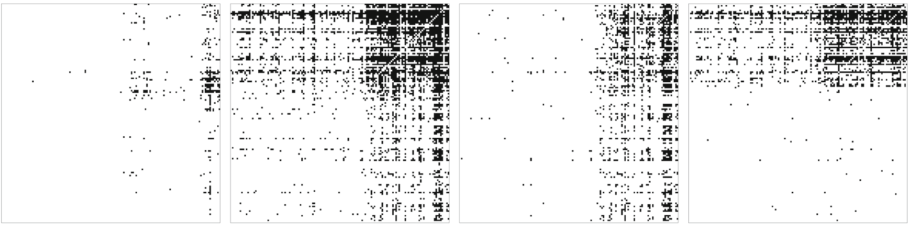


Fig. 8 Visualization of $\mathbf{X}_m(2020)$, $m \in \{260300, 74, 7404, 841989\}$

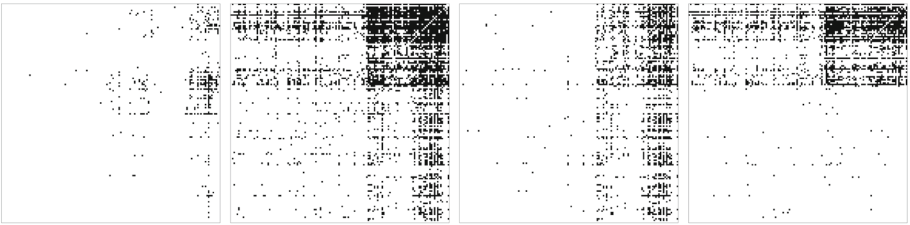


Fig. 9 Visualization of $\mathbf{X}_m(2021)$, $m \in \{260300, 74, 7404, 841989\}$

References

Ahlers, R., Kiezebrink, V., & Dugersuren, S. (2020). Undermining mongolia.

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*, 47–97.

Barbillon, P., Donnet, S., Lazega, E., & Bar-Hen, A. (2017). Stochastic block models for multiplex networks: An application to a multilevel network of researchers. *Journal of the Royal Statistical Society Series A, 180*(1), 295–314.

Bickel, P., Choi, D., Chang, X., & Zhang, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics, 41*(4), 1922–1943.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(7), 719–725.

Bonaccorsi, G., Riccaboni, M., Fagiolo, G., & Santoni, G. (2019). Country centrality in the international multiplex network. *Applied Network Science, 4*, 12.

Caldarelli, G. (2007). *Scale-free networks: Complex webs in nature and technology*. Oxford University Press.

Campello, R. (2007). A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters, 28*(7), 833–841.

Celisse, A., Daudin, J.-J., & Pierre, L. (2011). Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics, 6*, 1847–1899.

Côme, E.., & Greed, N. J. (2022). An r package for model-based clustering by greedy maximization of the integrated classification likelihood.

Côme, E., Jouvin, N., Latouche, P., & Bouveyron, C. (2021). Hierarchical clustering with discrete latent variable models and the integrated classification likelihood. *Advances in Data Analysis and Classification, 15*, 957–986.

Côme, E., & Latouche, P. (2015). Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Statistical Modelling, 15*(6), 564–589.

Cornaro, A., & Rizzini, G. (2022). Environmentally extended input–output analysis in complex networks: A multilayer approach. *Annals of Operations Research*.

Deza, M. M., & Deza, E. (2009). Encyclopedia of distances. In *Encyclopedia of distances* (pp. 1–583, Springer).

Fortunato, S. (2010). Community detection in graphs. *Physics Reports, 486*(3), 75–174.

Funke, T., & Becker, T. (2020). Complex networks of material flow in manufacturing and logistics: Modeling, analysis, and prediction using stochastic block models. *Journal of Manufacturing Systems, 56*, 296–311.

Geiger, B. C., & Kubin, G. (2012). Relative information loss in the PCA. *arXiv e-prints*, page arXiv:1204.0429.

Girvan, M., & Newman, M. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America, 99*, 7821–7826.

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD ’16 (pp. 855–864), New York, NY, USA. Association for Computing Machinery.

Halu, A., Mondragón, R. J., Panzarasa, P., & Bianconi, G. (2013). Multiplex pagerank. *PloS ONE, 8*(10), e78293.

Hs commodity codes. https://www.tariffnumber.com/. Accessed 12 February 2022.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*(1), 193–218.

Iacovacci, J., & Bianconi, G. (2016). Extracting information from multiplex networks. *Chaos An Interdisciplinary Journal of Nonlinear Science, 26*(6), 065306.

Iacovacci, J., Rahmede, C., Arenas, A., & Bianconi, G. (2016). Functional multiplex pagerank. *EPL (Europhysics Letters), 116*(2), 28004.

Karrer, B., & Newman, M. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E, 83*, 016107.

Khosla, M., Leonhardt, J., Nejdl, W., & Anand, A. (2019a). Node representation learning for directed graphs. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 395–411). Springer.

Khosla, M., Setty, V., & Anand, A. (2019b). A comparative study for unsupervised network representation learning. *IEEE Transactions on Knowledge and Data Engineering, 33*(5), 1807–1818.

Kim, S., & Shin, E.-H. (2002). A longitudinal analysis of globalization and regionalization in international trade: A social network approach. *Social Forces, 81*(2), 445–471.

Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research, 281*(3), 628–641.

Mariadassou, M., Robin, S., & Vacher, C. (2010). Uncovering latent structure in valued graphs: A variational approach. *The Annals of Applied Statistics, 4*, 715–742.

Matias, C., & Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79*(4), 1119–1141.

Matias, C., & Robin, S. (2014). Modeling heterogeneity in random graphs through latent space models: A selective review. *ESAIM Proceedings and Surveys, 47*, 541.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint* arXiv:1301.3781.

Newman, M. (2003). The structure and function of complex networks. *SIAM Review, 45*, 167–256.

Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E, 74*, 036104.

Newman, M. E. J. (2018). *Networks: An introduction*. Oxford: Oxford University Press.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E, 69*, 026113.

Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association, 96*, 1077–1087.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014) Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD ’14 (pp. 701–710), New York, NY, USA. Association for Computing Machinery.

Rahmede, C., Iacovacci, J., Arenas, A., & Bianconi, G. (2018). Centralities of nodes and influences of layers in large multiplex networks. *Journal of Complex Networks, 6*(5), 733–752.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large Corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50) Valletta, Malta. ELRA.

Smith, D. A., & White, D. R. (1992). Structure and dynamics of the global economy: Network analysis of international trade 1965–1980. *Social Forces, 70*, 857–893.

Snyder, D., & Kick, E. L. (1979). Structural position in the world system and economic growth, 1955–1970: A multiple-network analysis of transnational interactions. *American Journal of Sociology, 84*(5), 1096–1126.

Tong, X., & Lifset, R. (2007). International copper flow network: A blockmodel analysis. *Ecological Economics, 61*, 345–354.

Un comtrade databases. https://comtrade.un.org/data. Accessed 02 February 2022.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge: Cambridge University Press.

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