Working Paper No.
2026-18

INET Oxford Working Paper Series
31 July 2026

Synthetic supply networks

Galvin Ng, Luca Mungo, Damien Bertrand, and François Lafond

Synthetic supply networks

Galvin Ng[1,2], Luca Mungo[3,4], Damien Bertrand[5], and François Lafond[3,6]

[1]Complexity Science Hub, Vienna, Austria
[2]Vienna University of Economics and Business
[3]Institute for New Economic Thinking, University of Oxford
[4]Macrocosm Inc, New York, USA
[5]École Polytechnique Fédérale de Lausanne, Switzerland
[6]Smith School of Enterprise and the Environment, University of Oxford

July 31, 2026

Abstract

A good representation of the population of firms and households is essential for large-scale economic models. While there exist good methods to create synthetic populations of households, creating synthetic populations of firms and, crucially, their supply chain links, is typically much harder. Here, we introduce a flexible method to create synthetic supply networks that match both the known properties of firm-level supply networks and the properties of aggregated input-output tables used in macroeconomic models. Our method is fast, and because it uses only publicly available data, it is fully reproducible and can be easily extended.

Keywords: Synthetic population; Supply networks; Loss optimization

1 Introduction

Economic shocks propagate through transactions between firms, but those transactions are rarely observable. In the few countries in which production networks data is observed, they are highly confidential, and thus hardly usable by the scientific community at large to develop high-resolution economic models. As a result, while there is growing recognition that modeling economic systems at a high level of detail can improve our ability to understand and predict their dynamics [1–6], the development of these models faces a serious bottleneck.

This problem is largely solvable by constructing synthetic populations: artificial populations that reproduce key statistical properties of the real system while preserving privacy and filling gaps in the observed data. In fact, for the representation of individual people (as workers, consumers, and members of households), there already exist many methods; the availability of detailed census data has enabled novel computational techniques for generating statistically faithful synthetic populations of households, for instance at the country level [7–10], global level [11, 12], and with applications in policy planning [13], finance [14], and in healthcare [4, 15]. By contrast, general-purpose, reproducible synthetic populations of firms connected through weighted production networks remain much less developed. Generating a synthetic production network is harder than generating a synthetic population of individuals or households: it is not enough to generate plausible firms in isolation; one must also infer who buys from whom, and along which sectoral and geographic patterns. Aside from aggregate trade flows, which have long been recorded in input-output tables, these properties were, until recently, poorly known.

The situation is, however, beginning to change. Several countries have recently developed complete maps of the business-to-business transactions taking place nationally [16]. These datasets have revealed that production networks have statistical properties that are both distinctive and universal [17]. This suggests that synthetic supply networks that respect those properties can be very helpful, as they can be close to the real-world network they are substituting for.

Developing synthetic supply networks is essential because, as with synthetic populations of individuals, a crucial issue is that of confidentiality. Statistical agencies and the machine learning community are increasingly concerned about re-identification attacks [18], creating an overall push-back on the release of useful micro-data [19]. An approach to create synthetic populations of firms would be to train a machine learning model on a confidential dataset, let it learn useful relationships in the data, and use this model to re-create synthetic populations, as suggested by [20]. However, such an approach initially requires access to confidential data. Furthermore, there is always a risk that a model may learn the data "too closely", so that releasing its weights would breach privacy to some degree. While the risk appears very low with current methods, the main risk mitigation method (differential privacy) gives guarantees that are hard to interpret [19]. Other interesting approaches are being developed, such as federated learning [21] or distributed calibration [22], which limit the need for complex agreements among data holders but still requires access to confidential data. In any case, the use of confidential data makes results harder to reproduce and extend. As a result, while learning from confidential data to create synthetic networks remains an important avenue of research, here we propose a simpler, more transparent, and immediately useful approach, that can effectively democratize the use of plausible production networks in economic models.

Our approach is entirely reproducible because it *uses only publicly available information*. This is possible thanks to two sources of information: the aforementioned statistics from the micro-data, such as the mean number of customers per firm, and the aggregate input-output tables from the OECD-ICIO dataset [23], which provide payment flows among industries (45 industries in our case). Our initial hypotheses was that combining highly distinctive statistics about the structure of the micro-network with meso-level information should help to dramatically restrict the space of plausible networks. While this is likely true, we have also found that several approaches that we have experimented with produce networks that match the targeted statistics, but are nevertheless highly implausible, as judged from visualizing the univariate and bivariate distributions. By contrast, the algorithm we introduce here is able to choose a candidate network that respects the specific statistics used as targets (in a "soft" sense), while also resulting in univariate (degrees, strengths, weights) and bivariate distributions that appear reasonable visually.

Our contribution relates to a rapidly growing literature on supply network reconstruction [24], which can be roughly divided into two blocks. First, a large body of literature is concerned with individual links. Often motivated by issues of supply chain visibility and supply chain risk management, a large part of the literature has focused on individual links, offering methods to collect data from unstructured sources, infer links using statistical inference methods, and generate synthetic datasets [20, 25–32]. Second, and thus more related to our goal here, many papers have attempted to construct supply networks with the right system-level properties. This includes a well-established literature leveraging maximum entropy methods [33–37], random network formation models [38–40], or bespoke algorithms [41].

Our contribution is very distinct from these papers because we clearly state and address a different problem: creating an entirely synthetic population that matches both micro-moments and real-world IOTs, using only publicly and freely available information, and from an algorithm that is transparent, scalable, and can be released open-source. Our synthetic populations can then easily be used to initialize the microstates of 1:1 economic models realistically (as in [40]). Importantly, the most closely related contributions [37, 41, 42] use confidential data and/or do not produce networks as close to real-world networks as our method does.

2 Results

2.1 Reconstruction pipeline and experimental setup

Our method (see *Methods* and the SI for details) takes as inputs a target number of firms, a set of empirical firm-network properties, and an industry-level input-output table. It then generates a directed, weighted firm-level production network in four main steps (Fig. 1). First, we sample separate degree sequences for the number of suppliers and customers, calibrating their marginal distributions to selected empirical properties, including their dispersion and tail behavior. Second, we couple the two degree sequences to reproduce the positive correlation between firms' numbers of suppliers and customers, and use them to generate a directed binary network. Third, we assign preliminary transaction values to the existing links using firm-level latent factors, or *fitnesses*, and node degrees. Finally, we allocate firms to industries and rescale the transaction values within each industry pair so that their aggregation reproduces the target input-output table.

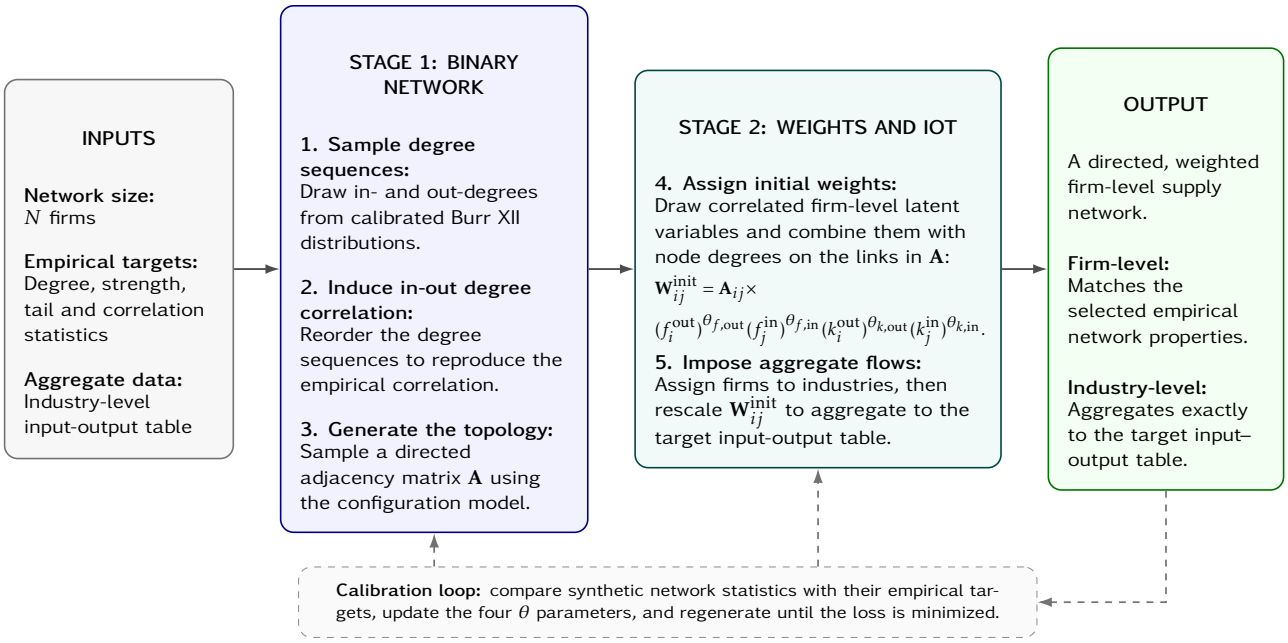


Figure 1: Overview of the synthetic firm-level supply network algorithm. The inputs are the network size, empirical network targets, and a specified industry-level input-output table. The algorithm generates a binary firm-level topology, assigns edge weights, and iteratively calibrates the θ parameters by comparing the statistics of the generated network with their empirical targets. The resulting directed, weighted firm network reproduces the selected firm-level properties and aggregates exactly to the target input–output table.

We rely on the findings in [17] to determine our reconstruction targets. The study provides a thorough review of all the published information regarding firm-level production networks, as well as an analysis of two datasets based on VAT reporting. It shows that, across different geographies and throughout time, firm-level production networks exhibit some striking statistical regularities. For instance, both in-degree (number of suppliers) and out-degree (number of customers) distributions are fat-tailed; however, the tail exponent of the in-degree distribution is ~ 2.5, while out-degree distributions typically have much fatter tails, with a tail exponent ~ 1.5. Correlations between network properties are also regular: for example, the correlation between expenses and the number of suppliers is lower than the correlation between sales and the number of customers. In the Supplementary Information, we explain how we select which statistics to target, and how we choose their numerical values.

For the main application, we generate a synthetic production network for Hungary in 2015 containing 10^5 firms. This network size is representative of the country-level production networks surveyed by [17], which contain between 10^4 and 10^6 firms. The following section assesses whether the generated network reproduces the targeted firm-level properties, whether its full distributions are empirically plausible, and whether it remains consistent with the aggregate input-output table.

2.2 Comparison of synthetic and real networks

Network-level summary statistics. Table 1 compares the properties of the synthetic network with the corresponding empirical benchmarks, distinguishing between properties that are not targeted and those that are targeted, either when constructing the binary network or when assigning weights.

The generated network closely reproduces the targeted properties of the degree, strength, and edge-weight distributions, as well as the principal relationships between binary and weighted network quantities. While some properties have an almost exact (e.g. tail exponents), others exhibit small deviations (some of the correlations or regressions coefficients). For instance, we reproduce very well the correlation between out-strengths and out-degrees and we slightly underestimate the correlation between in-strength and in-degrees; we still reproduce the fact that strength-degree correlations are smaller on the out- side than on the in-side.

Metric	Synthetic	Empirical	Target
Number of nodes	100,000	100,000	B
Number of edges	3,825,747	4,000,000	B
Share of $k^{\text{in}} = 0$	0		NT
Share of $k^{\text{out}} = 0$	26.4		NT
Mean degree	38.3	40.0	B
Max k^{in}	3,190		B
Max k^{out}	22,462		B
Mean log k^{in}	3.28	3.0	NT
Var log k^{in}	1.31	2.0	B
Mean log k^{out}	2.18	2.0	NT
Var log k^{out}	2.95	3.0	B
Mean log s^{in}	10.8	10.0	NT
Var log s^{in}	8.88	9.0	W
Mean log s^{out}	10.9	10.0	NT
Var log s^{out}	9.18	8.0	W
LWCC	95,556		NT
Avg path length	2.71	3.0	NT
Degree assortativity	-0.0688	[-0.2, -0.01]	NT
Reciprocity	0.00671	[0.03, 0.05]	NT
Avg clustering	0.0987	[0.19, 0.28]	NT
Global clustering	0.0208	low	NT

Metric	Synthetic	Empirical	Target
Hill α, k^{in}	2.53	2.5	B
Hill α, k^{out}	1.46	1.5	B
Hill α, s^{in}	1.1	1.0	W
Hill α, s^{out}	1.09	1.0	W
Hill α, influence	1.45	[1.2, 1.3]	NT
Hill α, weights	1.09	[1.1, 1.2]	NT
Corr: $k^{\text{in}} \sim k^{\text{out}}$	0.552	0.55	B
Corr: $s^{\text{in}} \sim s^{\text{out}}$	0.505	0.5	NT
Corr: $s^{\text{out}} \sim k^{\text{out}}$	0.442	0.5	W
Corr: $s^{\text{in}} \sim k^{\text{in}}$	0.592	0.75	W
OLS: $s^{\text{out}} \sim k^{\text{out}}$	0.779	0.76	W
OLS: $k^{\text{out}} \sim s^{\text{out}}$	0.251	0.33	W
OLS: $s^{\text{in}} \sim k^{\text{in}}$	1.54	1.4	W
OLS: $k^{\text{in}} \sim s^{\text{in}}$	0.227	0.4	W
TLS: $s^{\text{in}} \sim s^{\text{out}}$	0.968	1.0	W
TLS: $k^{\text{in}} \sim k^{\text{out}}$	0.496	0.7	NT
IOT RMSE (100 MUSD)	8.01e-04		W

[B] Targeted when creating adjacency matrix $\mathbf{A}_{ij}$.

[W] Targeted when creating weights W_{ij} on existing links $\mathbf{A}_{ij}$.

[NT] Not targeted.

Table 1: Network properties of a generated synthetic supply network with 100,000 firms, aggregated to the 2015 Hungarian input-output table. Values are reported to 3 significant figures, except for the number of nodes and edges, which are reported as integers.

The largest discrepancies are in some features of the binary network, such as reciprocity, clustering, and assortativity. While these are important features of real networks, they were not included in the calibration targets, and, in general, can't be reproduced with the configuration model that we used to generate the binary network. In fact, the configuration model does not include mechanisms that explicitly promote reciprocal relationships, triadic closure, or assortative matching between firms [43], but we keep it here as it is simple and very fast.

Visual evidence on distributional properties. Our experience creating networks that match the properties in Table 1 suggests that some procedures and optimization algorithms may achieve what seem good results according to these statistics, yet deliver networks that are clearly pathological when the distributions are visualized. For this reason, we show in Figure 2 the univariate and joint distributions generated by the model.

The synthetic network reproduces the asymmetric degree distributions and symmetric strengths distributions, showing patterns that closely resemble those published in Ref [17] and in the references cited therein (note, however, that for the joint distributions, Ref. [17] obfuscates bins with less than 10 observations for confidentiality reasons, so our figures are not perfectly comparable).

Consistency with IO tables. Finally, Fig. 3 compares the targeted Input-Output matrix with the one obtained by aggregating our firm-level synthetic network, showing an excellent agreement. The agreement is expected because the rescaling procedure explicitly enforces consistency with the target input-output table. However, it is not trivial that rescaling micro-values to ensure meso-scale properties keeps micro-scale properties unaffected.

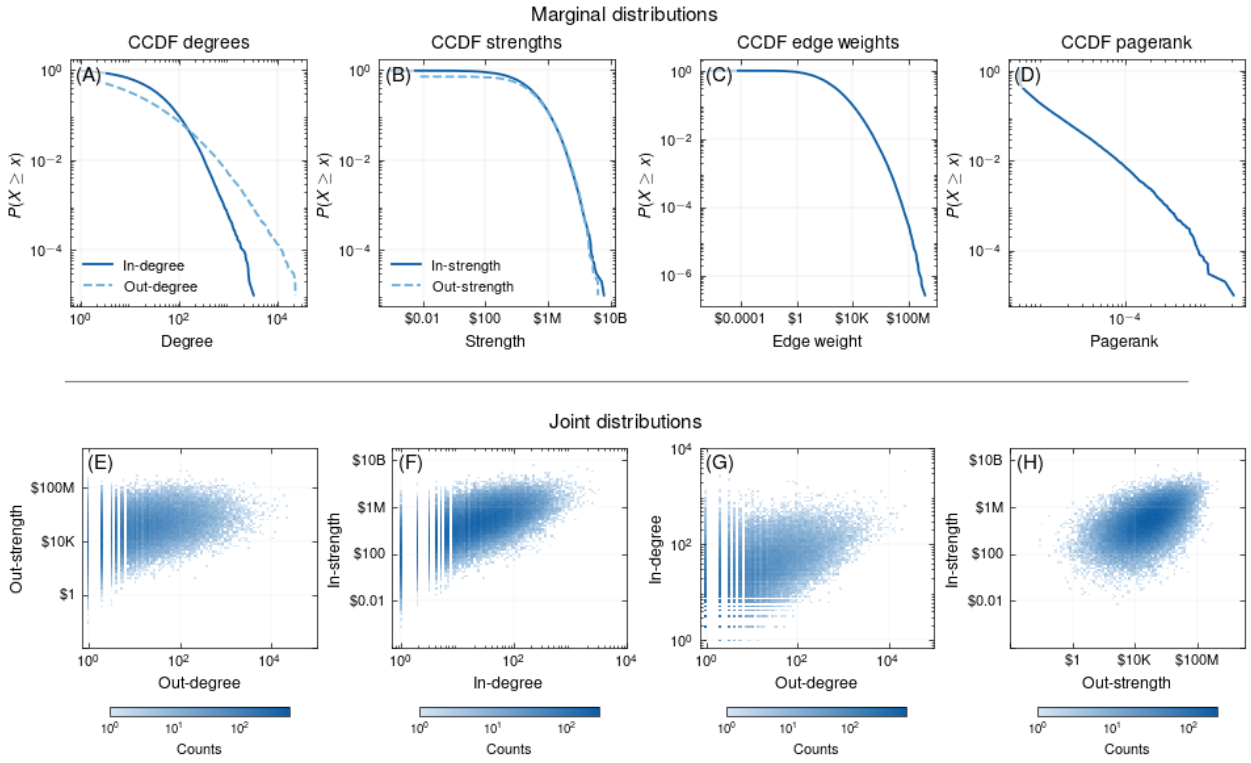


Figure 2: Distributional properties of a generated synthetic supply network of 100,000 firms, aggregated to the 2015 Hungarian input-output table. (A-D) Complementary cumulative distribution functions (CCDFs) of key node and edge-level variables. Strengths and weights are measured in USD. (E-H) Joint distributions (two-dimensional histograms with logarithmic color scaling) of key pairs of variables. All axes are displayed in logarithmic scales.

Figure 3: Comparison between the aggregated synthetic firm-level production network and the empirical 2015 Hungarian input-output table (IOT). The left panel shows the industry-level transactions obtained by aggregating the synthetic firm-level network, while the right panel shows the empirical Hungarian 2015 IOT. Colors correspond to common quantile bins computed jointly across both matrices, allowing direct comparison of transaction flows. The synthetic network reproduces the industry flows observed in the empirical IOT.

3 Discussion

Our method provides plausible synthetic networks, compatible with macro-economic aggregates, without having direct access to administrative data. This paves the way for a straightforward research agenda developing synthetic populations for all networks underlying the functioning of economic systems as measured in national accounts, pursuing an agenda advanced at a more aggregate level by [44] (see also [45] for a bottom-up reconstruction of national accounting objects in the case of consumption). This would include bipartite networks linking workers to the firms that employ them, and consumers to the firms they buy from. This can be achieved using a research pipeline similar to the one we introduced here, after researchers with access to scanner data, transaction and/or survey data (for consumer-to-firm networks, and employee-employer networks (for firm-to-worker networks) make key moments available publicly. Similarly, the study of loan-level data could provide key moments to create realistic representation of the real-financial nexus of the economy.

Our approach is deliberately simple, leaving room for increased complexity when more targetable statistics will become available. As international institutions develop distributed micro-data projects, more and more useful moments of firm networks will become available, including inter-country trade networks, geographic patterns, more detailed insights about industry-level patterns, or using the recently developed reconstructions of product-product networks [46–49] instead of the very coarse IOTs we have been using. New topological features can be incorporated as an explicit step in the construction of the adjacency matrix. For instance, while preserving the degree distribution, one can scramble the pairing of existing edges to generate the desired levels of reciprocity and clustering in the synthetic network. Features involving edge weights could potentially be incorporated by extending the loss function. It is also possible to extend our method for use when the identity of the firms and their (intermediate) sales are known, although there would likely be an incompatibility between observed firm sales and IOTs, so one would have to decide which one should be preserved.

This point raises a notable conceptual limitation of our work. As explained in some details in [17], IOTs do not record transactions in the same way as firm-level VAT data does, because a number of accounting conventions differ, for instance the treatment of wholesale and retail, transport margins, or the financial sector. We know from [17] that this problem is somewhat limited outside of specific sectors, so here we have ignored it. A simple partial solution would be to leverage the recent and important work in Ref. [50], who reintroduce wholesale and retail trade flows in IOTs, making them much closer conceptually to VAT datasets.

Our approach is straightforwardly testable. We know that some countries do have VAT data, but have not published anything about it yet. Our paper makes very clear testable predictions: one can generate synthetic networks for these countries now, and compare the synthetic networks to the actual networks only in the future. An interesting test would be to compare real and synthetic networks not only in terms of topological and statistical features, but also in terms of the behavior of specific models or stress-test algorithms such as the ESRI [51], as suggested in [24].

Our paper also provides a new problem for the machine learning community, with benchmark results to be improved upon. Our pipeline does not use generative AI methods. We have found that existing methods in AI-based graph generation did not meet our graph needs or could not be adapted in ways that were effective. We offer a clear challenge to the computer science community: develop AI-based graph generation methods that beat our established benchmark while remaining tractable for graphs at the national scale (10^6). An interesting development would be to leverage the work on synthetic populations of LLM-based personas [52–55] to create behaviorally plausible synthetic populations of firms operating in a supply chain. Our work provides a benchmark for the topological structure, and we think a useful method to initialize such approaches.

Our code is fully available and implementable in a differentiable framework, so potentially re-usable by novel approaches in that space. Our algorithm is available open source, with an easy implementation for all countries covered by the OECD ICIO tables. The user only needs to decide on a country to target and a number of firms. A configuration file allows users to modify the target values for the elements of the loss.

Methods

We generate the synthetic firm-level supply networks in two phases: first, creating a directed binary network consisting of firm-to-firm connections; second, populate edge weights conditional on the existing links targeting the correlation structure between strengths and degrees, and the aggregated economic flows from the industry-level input-output table. We provide a brief summary here. See the SI a full explanation, and the codebase for all the implementation details.

Generating the binary network

We first draw out- and then in-degree sequences from Burr XII distributions, with strictly positive parameters (c, k, λ). The Burr XII distribution $X \sim \text{BurrXII}(c, k, \lambda)$ is characterized by three parameters c, k, λ. Its cumulative distribution function (CDF) is given by $F(x) = 1 - \left[1 + (x/\lambda)^c\right]^{-k}$, where the asymptotic tail exponent is $\alpha = ck$, since $1 - F(x) \sim x^{-ck}$ as $x \to \infty$. We sample degrees from the Burr XII distribution to control the tail exponent, mean, and log-variance of the degree distributions. We calibrate the parameters numerically so that the final distribution matches user-defined targets. Since the Burr XII distribution is continuous and supported on $(0, \infty)$, the sampled observations must subsequently be discretized, truncated, and reconciled so that the total in-degree equals the total out-degree.

Because the firms' in- and out-degree sequences are drawn independently from Burr XII distributions, their correlation is initially zero. To reproduce the empirical log in- and out-degree correlation of approximately 0.55 reported by [17], we first rank firms according to their in-degrees and perturb the ordering using multiplicative lognormal noise with rank-dependent variance, $k_i^{\text{in, scrambled}} = k_i^{\text{in}} \times \varepsilon_i$, with $\varepsilon_i \sim \text{Lognormal}\left(-\frac{1}{2}(1-\tilde{r}_i)^{2\zeta}, (1-\tilde{r}_i)^{2\zeta}\right)$ where $\tilde{r}_i \in [0,1]$ denotes the normalized rank of firm i's in-degree. The out-degree sequence is then reordered to match the ranking of the scrambled in-degrees. This procedure preserves the marginal degree distributions while inducing a tunable positive correlation between in- and out-degrees. We calibrate ζ to match the empirical Pearson correlation reported by [17], obtaining $\zeta = -0.14$.

We then sample an adjacency matrix $\mathbf{A}$ from the configuration model with the in-degree and reordered out-degree sequences as inputs, and remove parallel edges and self-loops.

Generating the weighted network

We then assign weights to the edges in the adjacency matrix $\mathbf{A}$. We do this by first generating "fitnesses" (latent variables for each node, a classic idea in network science), and then using these and node degrees in a gravity-like formula to determine edge-level weights,

$$\mathbf{W}_{ij}^{\text{init}} = (f_i^{\text{out}})^{\theta_{f,\text{out}}} (f_j^{\text{in}})^{\theta_{f,\text{in}}} (k_i^{\text{out}})^{\theta_{k,\text{out}}} (k_j^{\text{in}})^{\theta_{k,\text{in}}} \mathbf{A}_{ij}, \tag{1}$$

where f are the fitnesses and $\boldsymbol{\theta} \equiv (\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}})$ denotes a set of tunable parameters. We then rescale $\mathbf{W}_{ij}^{\text{init}}$ to match the input-output table (IOT). We first describe how we determine the fitnesses and how we rescale to match the IOT, assuming the parameters $\boldsymbol{\theta}$ are known. Next, we describe our procedure to find optimal values of $\boldsymbol{\theta}$.

Assignment of weights assuming θ is known We sample in- and out-fitnesses from a joint lognormal $(\ln f^{\text{in}}, \ln f^{\text{out}}) \sim \mathcal{N}(\boldsymbol{\mu}_f, \Sigma)$. While the choice of the mean does not affect the result, we choose the covariance matrix carefully. Note that in Eq. 1, we assign each fitnes variable an exponent θ, which we optimize. Since the variance of the variable f^θ can be tuned arbitrarily by changing the value θ, we choose the variance of $\ln f$ in a way that we facilitate parameter search, by ensuring that all four θs are likely to be on similar scales. Intuitively, we expect the four θs to be on similar scales if the four variables are also on similar scales. Therefore, we assume that the variances of the log in- and log out-fitnesses are equal to the variances of the log in- and log out-degrees. Finally, for the covariance, we introduce a parameter ρ that tunes the correlations, and which we calibrate to 0.8, ensuring that the resulting log in- and log out-strength correlation is close to the empirical value of 0.5.

To match the IOT given edge-level weights $\mathbf{W}_{ij}^{\text{init}}$, we assume that the number of firms per industry is proportional to industry sales. Firm ordering follows industry ordering: the first M_1 firms are assigned to industry 1, the next M_2 firms to industry 2, and so on. While proportional allocation appears as a reasonable assumption, the method is compatible with any desired allocation strategy. With firms assigned to industries, we rescale the firm-to-firm weights to ensure that the industry-to-industry weights match a given IOT, using

$$\mathbf{W}_{ij} = \frac{\text{IOT}_{g_i g_j}}{\sum_{f \in g_i} \sum_{h \in g_j} \mathbf{W}_{fh}^{\text{init}}} \mathbf{W}_{ij}^{\text{init}}.$$

Estimation of the optimal parameters θ Finally, we calibrate the parameters $\boldsymbol{\theta} = (\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}})$ so that the generated network reproduces selected marginal and joint properties of firms' strengths and degrees. Using Optuna [56], we minimize

$$\mathcal{L} = \mathcal{L}_{\text{tail}} + \mathcal{L}_{\text{corr}} + \mathcal{L}_{\text{OLS}} + \mathcal{L}_{\text{TLS}} + \lambda_{\text{var}} \mathcal{L}_{\text{var}}.$$

For any statistic q, let $\mathcal{D}(q) = \left[\widehat{q}(\boldsymbol{\theta}) - q^\star\right]^2$ denote the squared discrepancy between its value in the synthetic network, $\widehat{q}(\boldsymbol{\theta})$, and its empirical target, $q^\star$. The tail component is $\mathcal{L}_{\text{tail}} = \mathcal{D}(\alpha^{\text{in}}) + \mathcal{D}(\alpha^{\text{out}})$, where α^{in} and α^{out} are the tail exponents of the in- and out-strength distributions, estimated using the Hill estimator. The correlation component is

$\mathcal{L}_{\text{corr}} = \mathcal{D}(\rho^{\text{in}}) + \mathcal{D}(\rho^{\text{out}})$, where ρ^{in} is the Pearson correlation between log in-strength and log in-degree, and ρ^{out} is the corresponding correlation between log out-strength and log out-degree. The OLS component is $\mathcal{L}_{\text{OLS}} = \mathcal{L}^{\text{in}}_{\text{OLS}} + \mathcal{L}^{\text{out}}_{\text{OLS}}$, with $\mathcal{L}^{\text{in}}_{\text{OLS}} = \mathcal{D}(\beta^{\text{in}}_{s\leftarrow k}) + \mathcal{D}(\beta^{\text{in}}_{k\leftarrow s})$ and $\mathcal{L}^{\text{out}}_{\text{OLS}} = \mathcal{D}(\beta^{\text{out}}_{s\leftarrow k}) + \mathcal{D}(\beta^{\text{out}}_{k\leftarrow s})$. Here, $\beta^{\text{in}}_{s\leftarrow k}$ is the OLS slope obtained by regressing log in-strength on log in-degree, whereas $\beta^{\text{in}}_{k\leftarrow s}$ is obtained from the reverse regression. The out-strength coefficients are defined analogously. The arrow points from the explanatory variable to the dependent variable. The TLS component is $\mathcal{L}_{\text{TLS}} = \mathcal{D}(\tau^{\text{in}\leftarrow\text{out}})$, where $\tau^{\text{in}\leftarrow\text{out}}$ is the total least squares slope relating log out-strength to log in-strength. Finally, the variance component is $\mathcal{L}_{\text{var}} = \mathcal{D}(v^{\text{in}}) + \mathcal{D}(v^{\text{out}})$, where v^{in} and v^{out} are the variances of log in-strength and log out-strength, respectively. All loss components have unit weight except for the variance component, which is multiplied by $\lambda_{\text{var}} = 10^{-3}$. All correlations, regression coefficients, and variances are computed using log-transformed variables.

In practice, we repeat the Optuna search procedure 200 times, with each search consisting of 200 trials. Across these repeated searches, we observe stable optimal parameter regions: the optimal value of $\theta_{k,\text{in}}$ is centered around 0, while $\theta_{k,\text{out}}$ is consistently negative. On the other hand, $\theta_{f,\text{in}}$ and $\theta_{f,\text{out}}$ are either both negative or positive. Additional details on the parameter search are provided in the Supplementary Information, where we also show that networks generated using negative $\theta_{f,\text{in}}$ and $\theta_{f,\text{out}}$ are statistically indistinguishable from the positive case. For the remainder of the paper, we use the parameters $(\theta_{k,\text{in}}\,,\ \theta_{k,\text{out}}\,,\ \theta_{f,\text{in}}\,,\ \theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$.

Randomness enters the methodology through Burr degree draws, the reordering procedure for out-degrees, sampling the adjacency matrix from the configuration model, sampling of fitnesses, and firm-to-industry assignment. Our methodology is implemented using PyTorch sparse COO tensors, so computational and memory requirements scale with the number of edges rather than the size of the full matrix. Since the generated networks have approximately constant mean degree, this corresponds to $O(N)$ rather than $O(N^2)$ scaling. We generate synthetic supply networks of sizes up to $500,000$ firms, while preserving firm-level network properties and exact aggregation with input-output tables. This scalability makes the framework suitable for large-scale agent-based and macroeconomic applications.

Code availability

The code that has been used to obtain all results in the paper is publicly available on `https://github.com/galvinngkw/synthetic_supply_networks`.

Acknowledgments

G.N. was supported by the Digital Innovation School, which is funded by the Federal Ministry of Women, Science and Research (BMFWF) and the Federal Chancellery (BKA) of Austria. F.L. would like to thank Baillie Gifford, UK Research and Innovation through Economic and Social Research Council grant PRINZ (ES/W010356/1), and the Institute for New Economic Thinking at the Oxford Martin School for funding. F.L. is an affiliated scientist with Macrocosm Inc. We are grateful to Anton Pichler, Jan Hurt, and audiences at CSH, WIFO, Centro Enrico Fermi, IMT Lucca, SMU Trade Conference 2026, APCNCS 2026, and CEF 2026 for many useful comments and discussions.

Author contributions

F.L., G.N and L.M. conceived the work and developed the methodology. G.N. and L.M. wrote the code. G.N. performed the data analysis. G.N., F.L. and L.M. analyzed and interpreted the results. All of the authors contributed to the final manuscript.

Competing interests

The authors declare no competing interests.

References

[1] Robert L Axtell and J Doyne Farmer. Agent-based modeling in economics and finance: Past, present, and future. *Journal of Economic Literature*, pages 1–101, 2022.

[2] Sebastian Poledna, Michael Gregor Miess, Cars Hommes, and Katrin Rabitsch. Economic forecasting with an agent-based model. *European Economic Review*, 151:104306, 2023.

[3] Samuel Wiese, Jagoda Kaszowska-Mojsa, Joel Dyer, Jose Moran, Marco Pangallo, Francois Lafond, John Muellbauer, Anisoara Calinescu, and J Doyne Farmer. Forecasting macroeconomic dynamics using a calibrated data-driven agent-based model. *arXiv preprint arXiv:2409.18760*, 2024.

[4] Marco Pangallo, Alberto Aleta, R Maria del Rio-Chanona, Anton Pichler, David Martín-Corral, Matteo Chinazzi, François Lafond, Marco Ajelli, Esteban Moro, Yamir Moreno, et al. The unequal effects of the health–economy trade-off during the covid-19 pandemic. *Nature Human Behaviour*, 8(2):264–275, 2024.

[5] Marco Pangallo, R Maria del Rio-Chanona, et al. Data-driven economic agent-based models. In *The economy as an evolving complex system IV*, pages 1–18. Santa Fe Institute Press, 2025.

[6] Christian Diem, András Borsos, Tobias Reisch, János Kertész, and Stefan Thurner. Estimating the loss of economic predictability from aggregating firm-level production networks. *PNAS nexus*, 3(3):page064, 2024.

[7] Steven Rubinyi, Jasper Verschuur, Ran Goldblatt, Johannes Gussenbauer, Alexander Kowarik, Jenny Mannix, Brad Bottoms, and Jim Hall. High-resolution synthetic population mapping for quantifying disparities in disaster impacts: An application in the bangladesh coastal zone. *Frontiers in Environmental Science*, 10:1033579, 2022.

[8] Manon Prédhumeau and Ed Manley. A synthetic population for agent-based modelling in canada. *Scientific Data*, 10 (1):148, 2023.

[9] Imran Mahmood, Anisoara Calinescu, and Michael Wooldridge. Hierarchical population synthesis using a neural-differentiable programming approach. In *2025 Winter Simulation Conference (WSC)*, pages 139–150. IEEE, 2025.

[10] Jacopo Lenti, Lorenzo Costantini, Ariadna Fosch, Anna Monticelli, David Scala, and Marco Pangallo. Population synthesis with geographic coordinates. *arXiv preprint arXiv:2510.09669*, 2025.

[11] Marijn J Ton, Michiel W Ingels, Jens A de Bruijn, Hans de Moel, Lena Reimann, Wouter JW Botzen, and Jeroen CJH Aerts. A global dataset of 7 billion individuals with socio-economic characteristics. *Scientific Data*, 11(1):1096, 2024.

[12] Haoxiang Guan, Jiyan He, Liyang Fan, Zhenzhen Ren, Shaobin He, Xin Yu, Yuan Chen, Shuxin Zheng, Tie-Yan Liu, and Zhen Liu. Modeling earth-scale human-like societies with one billion agents. *arXiv preprint arXiv:2506.12078*, 2025.

[13] Joel Dyer, Arnau Quera-Bofarull, Nicholas Bishop, J. Doyne Farmer, Anisoara Calinescu, and Michael Wooldridge. Population synthesis as scenario generation for simulation-based planning under uncertainty. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS 2024, pages 490–498, Auckland, New Zealand, 2024. International Foundation for Autonomous Agents and Multiagent Systems. doi: 10.5555/3635637.3662899. URL https://dl.acm.org/doi/10.5555/3635637.3662899.

[14] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the first ACM international conference on AI in finance*, pages 1–8, 2020.

[15] Mauro Giuffrè and Dennis L Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *NPJ digital medicine*, 6(1):186, 2023.

[16] Anton Pichler, Christian Diem, Alexandra Brintrup, François Lafond, Glenn Magerman, Gert Buiten, Thomas Y Choi, Vasco M Carvalho, J Doyne Farmer, and Stefan Thurner. Building an alliance to map global supply networks. *Science*, 382(6668):270–272, 2023.

[17] Andrea Bacilieri, András Borsos, Pablo Astudillo-Estévez, Mads Hoefer, and François Lafond. Firm-level production networks: What do we (really) know? *Journal of Economic Dynamics and Control*, 187:105313, 2026. ISSN 0165-1889. doi: https://doi.org/10.1016/j.jedc.2026.105313. URL https://www.sciencedirect.com/science/article/pii/S016518892600059X.

[18] Luc Rocher, Julien M Hendrickx, and Yves-Alexandre De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1):1–9, 2019.

[19] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic data–what, why and how? *arXiv preprint arXiv:2205.03257*, 2022.

[20] Luca Mungo, François Lafond, Pablo Astudillo-Estévez, and J Doyne Farmer. Reconstructing production networks using machine learning. *Journal of Economic Dynamics and Control*, 148:104607, 2023.

[21] Ge Zheng, Lingxuan Kong, and Alexandra Brintrup. Federated machine learning for privacy preserving, collective supply chain risk prediction. *International Journal of Production Research*, 61(23):8115–8132, 2023.

[22] Aditi Garg and Ayush Chopra. Distributed calibration of agent-based models. In *epiDAMIK 2024: The 7th International Workshop on Epidemiology meets Data Mining and Knowledge Discovery at KDD 2024*, 2024.

[23] Norihiko Yamano, Joaquim Guilhoto, Ali Alsamawi, Colin Webb, Peter Horvát, Agnès Cimper, Carmen Zürcher, and Xue Han. Development of the oecd inter country input-output database 2021. *OECD Science, Technology and Industry Working Papers*, 2021(14), 2021.

[24] Luca Mungo, Alexandra Brintrup, Diego Garlaschelli, and François Lafond. Reconstructing supply networks. *Journal of Physics: Complexity*, 5(1):012001, 2024.

[25] Edward Elson Kosasih and Alexandra Brintrup. A machine learning approach for predicting hidden links in supply chain with graph neural networks. *International Journal of Production Research*, 60(17):5380–5393, 2022.

[26] Alexandra Brintrup, Edward Kosasih, Philipp Schaffer, Ge Zheng, Guven Demirel, and Bart L MacCarthy. Digital supply chain surveillance using artificial intelligence: definitions, opportunities and risks. *International Journal of Production Research*, 62(13):4674–4695, 2024.

[27] Serina Chang, Zhiyin Lin, Benjamin Yan, Swapnil Bembde, Qi Xiu, Chi Heem Wong, Yu Qin, Frank Kloster, Xi Luo, Raj Palleti, et al. Learning production functions for supply chains with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27878–27886, 2025.

[28] Kjartan van Driel, Leonardo Niccolò Ialongo, Pablo Andrés Astudillo, and Stefan Thurner. Generalized degrees for scalable discrete time dynamic graph generation. In *The Fourth Learning on Graphs Conference*, 2025.

[29] Seyda Köse, Christian Diem, Elma Dervic, Klaus Friesenbichler, Georg Heiler, Jan Hurt, Hernan Picatto, and Peter Klimek. Reconstructing temporal multi-relational firm networks at scale using large language models: The case of the semiconductor industry. *arXiv preprint arXiv:2605.15842*, 2026. doi: 10.48550/arXiv.2605.15842. URL https://doi.org/10.48550/arXiv.2605.15842.

[30] Tobias Reisch, Georg Heiler, Christian Diem, Peter Klimek, and Stefan Thurner. Monitoring supply networks from mobile phone data for estimating the systemic risk of an economy. *Scientific reports*, 12(1):13347, 2022.

[31] Yuya Katafuchi, Xinmeng Li, Daniel Moran, Taiki Yamada, Hidemichi Fujii, and Keiichiro Kanemoto. Construction of an enterprise-level global supply chain database. *Nature communications*, 16(1):11158, 2025.

[32] Yunbo Long, Sebastian Kroeger, Michael F Zaeh, and Alexandra Brintrup. Leveraging synthetic data to tackle machine learning challenges in supply chains: challenges, methods, applications, and research opportunities. *International Journal of Production Research*, pages 1–22, 2025.

[33] Leonardo Niccolò Ialongo, Camille de Valk, Emiliano Marchese, Fabian Jansen, Hicham Zmarrou, Tiziano Squartini, and Diego Garlaschelli. Reconstructing firm-level interactions in the dutch input–output network from production constraints. *Scientific reports*, 12(1):11847, 2022.

[34] Leonardo Niccolò Ialongo, Sylvain Bangma, Fabian Jansen, and Diego Garlaschelli. Multi-scale reconstruction of large supply networks. *arXiv preprint arXiv:2412.16122*, 2024.

[35] Ashwin Bhattathiripad and Vipin P Veetil. Reconstructing large scale production networks. *arXiv preprint arXiv:2512.02362*, 2025.

[36] Massimiliano Fessina, Giulio Cimini, Tiziano Squartini, Pablo Astudillo-Estévez, Stefan Thurner, and Diego Garlaschelli. Inferring firm-level supply chain networks with realistic systemic risk from industry sector-level data. *Scientific Reports*, 2026.

[37] Mitja Devetak and Antoine Mandel. Industry aware firm level network reconstruction. *arXiv preprint arXiv:2603.21895*, 2026.

[38] Tobias Reisch, András Borsos, and Stefan Thurner. Supply chain network rewiring dynamics at the firm-level. *PNAS nexus*, page pgag091, 2026.

[39] Andrew B Bernard and Yuan Zi. Sparse production networks. Technical report, National Bureau of Economic Research, 2022.

[40] Fanny Henriet, Stéphane Hallegatte, and Lionel Tabourier. Firm-network characteristics and economic robustness to natural disasters. *Journal of Economic Dynamics and Control*, 36(1):150–167, 2012.

[41] Jan Hurt, Katharina Ledebur, Birgit Meyer, Klaus Friesenbichler, Markus Gerschberger, Stefan Thurner, and Peter Klimek. Supply chain due diligence risk assessment for the eu: A network approach to estimate expected effectiveness of the planned eu directive. *arXiv preprint arXiv:2311.15971*, 2023.

[42] Andrew B Bernard, Emmanuel Dhyne, Glenn Magerman, Kalina Manova, and Andreas Moxnes. The origins of firm heterogeneity: A production network approach. *Journal of Political Economy*, 130(7):1765–1804, 2022.

[43] Mark Newman. *Networks*. Oxford university press, 2018.

[44] Asger Lau Andersen, Kilian Huber, Niels Johannesen, Ludwig Straub, and Emil Toft Vestergaard. Disaggregated economic accounts. *The Quarterly Journal of Economics*, 141(2):1005–1075, 2026.

[45] Gergely Buda, Stephen Hansen, Tomasa Rodrigo, Vasco M Carvalho, Alvaro Ortiz, and José V Rodríguez Mora. National accounts in a world of naturally occurring data: a proof of concept for consumption. *Cambridge Working Papers in Economics CWPE2244*, 2023.

[46] Thiemo Fetzer, Peter John Lambert, Bennet Feld, and Prashant Garg. Ai-generated production networks: Measurement and applications to global trade. Technical Report 1528, University of Warwick, Department of Economics, Coventry, UK, November 2024. URL https://wrap.warwick.ac.uk/188582/. Working paper, unpublished.

[47] Massimiliano Fessina, Andrea Tacchella, and Andrea Zaccaria. Product-level value chains from firm data: mapping trophic levels into economic growth. *arXiv preprint arXiv:2505.01133*, 2025.

[48] Lea Karbevska and César A Hidalgo. Mapping global value chains at the product level. *EPJ Data Science*, 14(1):21, 2025.

[49] Neave O'Clery, Ben Radcliffe-Brown, Thomas Spencer, and Daniel Tarling-Hunter. Deciphering the global production network from cross-border firm transactions. *arXiv preprint arXiv:2508.12315*, 2025.

[50] Celian Colon and Sebastian Poledna. Constructing supply-chain input–output tables: Mapping distribution networks from trade margins. *SSRN preprint SSRN:5796664*, 2025.

[51] Christian Diem, András Borsos, Tobias Reisch, János Kertész, and Stefan Thurner. Quantifying firm-level economic systemic risk from nation-wide supply networks. *Scientific reports*, 12(1):7719, 2022.

[52] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. doi: 10.48550/arXiv.2304.03442.

[53] Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024. doi: 10.48550/arXiv.2406.20094.

[54] Yihong Tang, Menglin Kong, Junlin He, Tong Nie, Wei Ma, and Lijun Sun. Llmsynthor: Macro-aligned micro-records synthesis with large language models. *arXiv preprint arXiv:2505.14752*, 2026.

[55] Davide Paglieri, Logan Cross, William A. Cunningham, Joel Z. Leibo, and Alexander Sasha Vezhnevets. Persona generators: Generating diverse synthetic personas for arbitrary contexts. *arXiv preprint arXiv:2602.03545*, 2026. doi: 10.48550/arXiv.2602.03545. URL https://arxiv.org/abs/2602.03545.

[56] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[57] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[58] Ronald E Miller and Peter D Blair. *Input-output analysis: foundations and extensions*. Cambridge university press, 2009.

Supplementary Information

In this SI, we (1) discuss how we choose which statistics to target, (2) explain our approach in full detail, both for the binary network creation step, and the weight attribution step, and (3) we provide robustness checks.

Our full initialization procedure is detailed in the following subsections, and is outlined in Box 4.

1. Generate a sequence of (continuous) out-degrees sampled from a Burr XII distribution. Discard and resample observations that are too large. Convert these observations to integers.
2. Generate a sequence of in-degrees sampled from another Burr XII distribution. Discard and resample observations that are too large. Convert the observations to integers. Reconcile the totals so that the sum of the in-degrees equals the sum of out-degrees.
3. Reorder the out-degrees to create a correlation with the in-degree sequence.
4. Sample an adjacency matrix $\mathbf{A}_{ij}$ from the Configuration Model based on the degree sequences $\{k^{\text{in}}\}_{i=1}^{N}$ and $\{k^{\text{out}}\}_{i=1}^{N}$ defined above.
5. Sample jointly lognormal in- and out-fitnesses $\{f^{\text{in}}\}_{i=1}^{N}$ and $\{f^{\text{out}}\}_{i=1}^{N}$ with specific mean and covariance. Multiply out-fitnesses by $\frac{\sum f^{\text{in}}}{\sum f^{\text{out}}}$ to ensure $\sum f^{\text{out}} = \sum f^{\text{in}}$.
6. Calibrate the parameters $\theta_{k,\text{in}}$, $\theta_{k,\text{out}}$, $\theta_{f,\text{in}}$, $\theta_{f,\text{out}}$ using Optuna by minimizing a loss function (Eq. 10 defined below). Denote the optimized parameters as $\theta^*_{k,\text{in}}$, $\theta^*_{k,\text{out}}$, $\theta^*_{f,\text{in}}$, $\theta^*_{f,\text{out}}$.
7. Compute an initial set of weights using
$$\mathbf{W}_{ij}^{\text{init},*} = (f_i^{\text{out}})^{\theta^*_{f,\text{out}}} (f_j^{\text{in}})^{\theta^*_{f,\text{in}}} (k_i^{\text{out}})^{\theta^*_{k,\text{out}}} (k_j^{\text{in}})^{\theta^*_{k,\text{in}}} \mathbf{A}_{ij}.$$
8. Obtain the final weights by rescaling the initial weights so that they agree with the aggregate Input-Output Tables, using
$$\mathbf{W}^*_{ij} = \frac{\text{IOT}_{g_i g_j}}{\sum_{f \in g_i} \sum_{h \in g_j} \mathbf{W}^{\text{init},*}_{fh}} \mathbf{W}^{\text{init},*}_{ij}.$$

Box 4: Procedure to generate a synthetic network. Refer to the individual subsections for details.

A Choosing target values

We choose which statistics are useful to target, and their numerical values, by a careful analysis of the results in [17], who found that a number of network statistics are very similar across countries (see especially their summary in Table 9). Ref. [17] synthesizes the literature, and analyzes two "complete" networks (based on VAT, no reporting thresholds) themselves. However, the number of networks for which statistics are reported remains low, so there is some arbitrariness in determining target values and acceptable ranges for the statistics. Similarly, Ref. [17] reports basic statistics, to allow for a synthesis of a larger literature, but does not report more sophisticated network properties. Here we are limited by this, but in principle other statistics could be added as targets in our optimization algorithm (although it would likely need to be adapted).

Table 2 provides details of how we select the target values used.

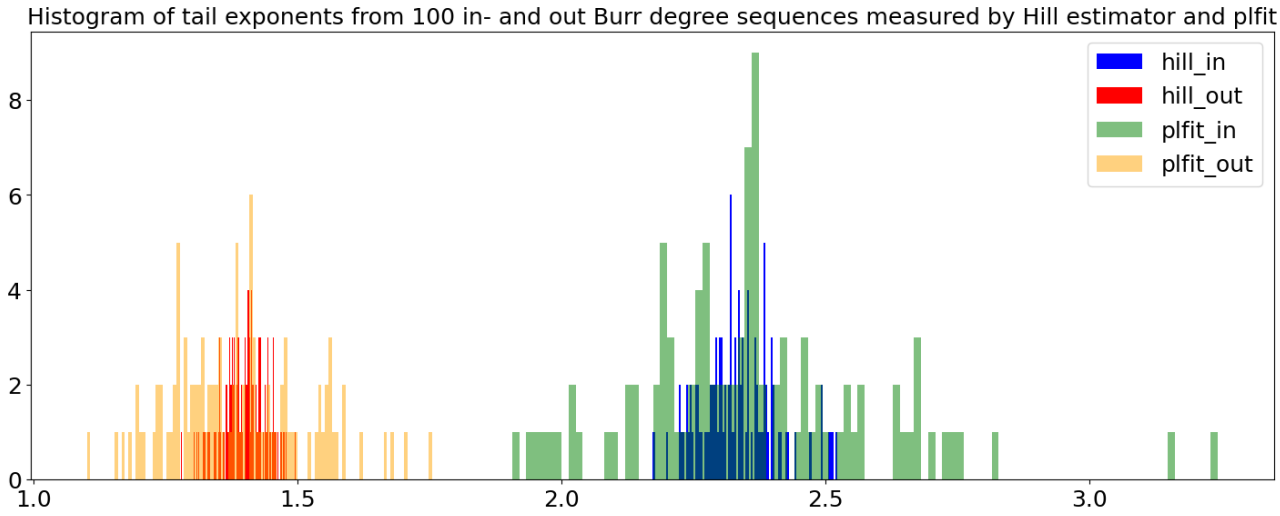
For power law tail exponents, a threshold above which the tail is measured should be determined. Bacilieri et al. [17] reports the results from various estimators that compute this threshold endogenously, showing some

variation (especially for in-degrees, as expected because the exponent is higher), and report the Newman-Clauset-Shalizi [57] Hill estimator (`plfit`) in their main text. For optimization purposes, it is easier to specify the threshold in advance. We have chosen a threshold roughly in line with the threshold found endogenously by `plfit` in [17]. Taking the example of degree distributions, we have chosen a threshold of 1% (see Table 2). Figure 5 confirms that our procedure generates networks with not only the right exponent when measured using a fixed 1% threshold, but also when measured using the `plfit` algorithm. The exponent measured using the `plfit` algorithm is substantially more volatile across simulation runs - which is why our algorithm implements a fixed threshold.

Property	Target value	Source	Property	Target value	Source
Mean degree	$0.86N^{1/3}$	S1	Tail exponent k^{in}	2.5	S5
Mean of log k^{in}	3	S2	Tail exponent k^{out}	1.5	S5
Mean of log k^{out}	2	S2	Tail exponent s^{in}	1	S4
Variance of log k^{in}	2	S2	Tail exponent s^{out}	1	S4
Variance of log k^{out}	3	S2	Tail exponent weights	1.15	S5
Maximum k^{in}	At most 5% of N	–	Tail exponent influence	$[1.2, 1.3]$	S4
Maximum k^{out}	At most 40% of N	–	Tail prob. Hill degrees	0.01	S6
Mean of log s^{in}	10	S3	Tail prob. Hill strengths	0.02	S7
Mean of log s^{out}	10	S3	Tail prob. Hill weights	0.0015	S8
Variance of log s^{in}	9	S3	Tail prob. Hill influence	0.04	S9
Variance of log s^{out}	8	S2	Corr: $k^{\text{in}} \sim k^{\text{out}}$	0.55	S10
Average path length	3	S4	Corr: $s^{\text{in}} \sim s^{\text{out}}$	0.5	S10
Degree assortativity	$[-0.2, -0.01]$	S4	Corr: $s^{\text{out}} \sim k^{\text{out}}$	0.5	S11
Reciprocity	$[0.03, 0.05]$	S4	Corr: $s^{\text{in}} \sim k^{\text{in}}$	0.75	S11
Average clustering	$[0.19, 0.28]$	S4	OLS: $s^{\text{out}} \sim k^{\text{out}}$	0.76	S11
Global clustering	low	S4	OLS: $k^{\text{out}} \sim s^{\text{out}}$	0.33	S12
			OLS: $s^{\text{in}} \sim k^{\text{in}}$	1.4	S11
			OLS: $k^{\text{in}} \sim s^{\text{in}}$	0.4	S12
			TLS: $s^{\text{in}} \sim s^{\text{out}}$	1	S4
			TLS: $k^{\text{in}} \sim k^{\text{out}}$	0.7	S5

[S1] From [17] and [16].
[S2] Rounded to nearest integer from Table B.10 of [17]; Ecuador (2015) and Hungary (2021) coincide.
[S3] Ecuador (2015) values from Table B.10 of [17], rounded to nearest integer.
[S4] From Table 9 of [17].
[S5] Midpoint of estimates reported in Table 9 of [17].
[S6] At tail probability 0.01, Hill estimates align on average with `igraph plfit`. See Figure 5.
[S7] Nearest 1% from Ecuador (2015) and Hungary (2021), Table C.17 and C.18 of [17].
[S8] Ecuador (2015), Table C.16 of [17].
[S9] Ecuador (2015), Table C.19 of [17].
[S10] Inferred from Ecuador (2015) and Hungary (2021), Table B.11 of [17].
[S11] Midpoint between Ecuador (2015) and Hungary (2021), Table B.11 of [17].
[S12] Computed from $\text{OLS}(y \sim x)\text{OLS}(x \sim y) = \text{Corr}(x, y)^2$.

Table 2: Network properties and their target values, based on empirical findings in [17]. Denote in-degree by k^{in}, out-degree by k^{out}, in-strength by s^{in}, and out-strength by s^{out}. Mean log-strengths are computed in units of 1 USD. All correlations, Ordinary Least Squares (OLS), and Total Least Squares (TLS) coefficients are computed on log-transformed variables. Ref. [17] report a mean degree of around 30-50, but suggest a 1/3 scaling exponent with size N by studying various countries for which networks are observed over time. We obtain the prefactor 0.86 to obtain a degree of about 40 for $N = 10^5$.



	mean	median	std	min	max
hill_in	2.33	2.33	0.0715	2.17	2.52
hill_out	1.4	1.4	0.0419	1.28	1.5
plfit_in	2.36	2.35	0.228	1.91	3.24
plfit_out	1.39	1.38	0.129	1.1	1.75

Figure 5: We generate 100 independent pairs of in- and out degree sequences using $(\alpha, \nu) = (2, 10)$ for the out-degrees and $(\alpha, \nu) = (3, 3)$ for the in-degrees. For each pair, tail exponents are estimated using the Hill estimator based on the top 1% of degrees and compared with estimates obtained from igraph `plfit`. The Hill and `plfit` estimators have nearly identical means, though `plfit` shows greater variability.

B Generating firm-level binary networks

B.1 Generate out-degrees sequences

We first draw out- and then in-degree sequences from Burr XII distributions, with strictly positive parameters (c, k, λ). Denote a Burr XII random variable by

$$X \sim \text{BurrXII}(c, k, \lambda), \tag{2}$$

with its Cumulative Distribution Function (CDF) given by

$$F(x) = 1 - \left[1 + \left(\frac{x}{\lambda}\right)^c\right]^{-k}. \tag{3}$$

Let U be a Uniform$[0,1]$ random variable. Then,

$$X = \lambda\left((1-U)^{-\frac{1}{k}} - 1\right)^{\frac{1}{c}}. \tag{4}$$

We want to find parameters of the Burr XII distribution that will give observations with specific tail exponent, mean and variance of the log-transformed values. From the CDF in Eq. (3), it is clear that as $x \to \infty$, $1 - F(x) \sim x^{-ck}$, so the asymptotic tail exponent is

$$\alpha = ck.$$

Then for a user-defined tail exponent $\alpha > 1$, mean $E[X] \equiv \mu$, and log-variance $\text{Var}(\ln X) \equiv \nu$ we can find the Burr XII parameters c, k, λ as follows. We solve $\nu = f(\alpha, k)$ numerically for k, where f is

$$\nu = \frac{\frac{\pi^2}{6} + \psi_1(k)}{(\alpha/k)^2}, \tag{5}$$

and $\psi_1(\cdot)$ is the trigamma function (See Corollary E.2.1 for its proof). Knowing k, the parameters c and λ are then determined using

$$c = \frac{\alpha}{k}, \quad \lambda = \frac{\mu}{kB\left(k - \frac{1}{c}, 1 + \frac{1}{c}\right)}, \tag{6}$$

where $B(\cdot,\cdot)$ is the Beta function. In principle, this gives us a straightforward mapping, helping us choose parameters of the Burr XII distribution that will give us the desired measured statistics.

However, in practice, we cannot obtain a distribution with the desired properties simply by sampling from a Burr XII distribution with these analytically determined statistics, for three reasons: (1) the tail exponent described above is valid only asymptotically, (2) the Burr XII distribution is continuous, so we have to discretize it, and (3) we have to truncate the distribution because firms can not have more partners than the number of other firms. We describe next each of these issues, and then how they impact naive Burr XII draws, and how we have dealt with this in our final algorithm.

Finite size tail exponent. While asymptotically, the tail of the Burr XII distribution behaves as a power law with slope $\alpha = ck$, at each finite percentile the slope can be quite different. In Corollary E.2.2, we calculate how the slope of the CCDF changes over the percentiles. We find that the complementary cumulative distribution function (CCDF), defined as $\bar{F}(x) = 1 - F(x)$, has a log-log slope given by

$$-\frac{d\log\bar{F}(x)}{d\log x} = \alpha\left(1 - p^{\frac{1}{k}}\right) = k\sqrt{\frac{\frac{\pi^2}{6} + \psi_1(k)}{\nu}}\left(1 - p^{\frac{1}{k}}\right), \tag{7}$$

evaluated at the point x_p satisfying $\bar{F}(x_p) = p$. The value p denotes the tail probability at which the slope is calculated. Equation (7) implies that the log-log slope depends on both tail probability[1] p and log-variance ν.

[1] See Section A for a discussion of the choice of tail probability p used in estimating the Hill exponents.

Discretization. Since the Burr XII distribution is continuous, we discretize the sampled values so that they represent integer-valued firm degrees. We do this by rounding to the nearest integer.

Truncation. Although the Burr XII distribution is supported on $(0, \infty)$, the theoretical maximum degree of a firm in a firm-level network is the number of firms. Moreover, empirical evidence in [17] shows that roughly, the largest out-degree is about an order of magnitude smaller than the number of nodes, and the largest in-degree is about an order of magnitude smaller than the largest out-degree. To capture these two properties in our synthetic binary network, we discard and resample any sampled out-degrees greater than 40% of the total number of firms, and any sampled in-degree greater than 5% of the number of firms.

Consequences for the choice of parameters. For the tail exponent, we have seen from Eq. (7) that the estimated tail exponent depends on the tail probability p and the log-variance ν. To better understand how discretization and truncation affect the final measured statistics of draws from a Burr XII distribution, we have performed an ablation study. Table 3 shows the results. Discretizing Burr XII observations lowers the realized log-variance relative to the initial value ν. As shown in the bottom-most block, for $N = 100,000$, an out-degree sequence drawn from the Burr XII distribution with $\nu = 10$ has a log-variance of 10, as expected; after discretization, however, the measured value decreases to 2.95. Truncating the discretized observations has negligible effect on both the tail exponent and log-variance, aside from reducing the maximum out-degree from 85,028 to 30,616.

To solve these issues, we calibrate α and ν by trial and error, and then compute the corresponding Burr XII parameters (c, k, λ) using Equations (5) and (6).

Figure 6 presents a grid over input tail exponent α and log-variance ν for out-degrees. To generate a truncated out-degree sequence for a network of $N = 100,000$ firms, mean $\mu \approx 40$, log-variance ≈ 3, tail exponent ≈ 1.4, and a maximum out-degree capped at $0.4N$, this meshgrid shows that values of $\alpha = 2$ and $\nu = 10$ will produce the desired properties.

Finally, to ensure robustness against stochastic variation in truncated Burr XII draws, for networks with $N = 100,000$ firms, we retain only out-degree sequences with tail exponent in the range $1.4 < \alpha < 1.5$, log-variance $\nu > 2.95$, and a maximum out-degree of at least 10% of firms. This restriction guarantees that the largest out-degree lies between 10% and 40% of the network size, consistent with empirical magnitudes.

B.2 Generate Burr in-degrees

As for the in-degrees, we sample an in-degree sequence from a Burr XII distribution, then truncate and turn the observations to integers. Specifically, we discard and resample any in-degree greater than $0.05N$. Similar to the out-degree case, Table 3 shows that the realized Hill exponent and log-variance for in-degrees are lower than their user-defined values. Accordingly, we calibrate the input parameters of the in-degree Burr XII distribution by trial and error. Figure 7 presents a grid over input tail exponent α and log-variance ν for in-degrees. To generate an in-degree sequence in a network of $N = 100,000$ firms with $\mu \approx 40$, $\alpha \approx 2.35$, $\nu \approx 2$, and a maximum in-degree capped at $0.05N$, we draw from a Burr XII distribution with parameters solved to match $\alpha = 3$ and $\nu = 3$.

B.3 Reconciling in- and out-degrees sums

In a directed graph, the sum of in- and out-degrees are equal. In our procedure above, although our draws for the in- and out-degrees are designed to have similar means, they are highly unlikely to have exactly the same sum. We reconcile the totals by modifying the initial in-degree sequence to match the sum of out-degrees. Let

$$\Delta = \sum_i k_i^{\text{out}} - \sum_j k_j^{\text{in}}.$$

We draw $|\Delta| \geq 0$ independent indices from $\{1, \ldots, N\}$ with selection probabilities

$$p_i = \frac{k_i^{\text{in}}}{\sum_j k_j^{\text{in}}}, \quad i = 1, \ldots, N.$$

	for fixed mean = 40			**for mean** $\bar{k} = 0.86N^{1/3}$ **(rounded to nearest** 10**)**	
	$N = 100,000$	$N = 10,000$	$N = 1,000$	$N = 10,000$ $\bar{k} = 20$	$N = 1,000$ $\bar{k} = 10$
Burr with discretization, truncation, and reconciliation					
mean degree	39 (0.917)	33.8 (1.48)	19.6 (1.54)	18.1 (0.933)	7.28 (0.804)
out-deg hill exponent	1.4 (0.0408)	1.67 (0.138)	5.83 (1.96)	1.51 (0.132)	2.53 (0.713)
in-deg hill exponent	2.33 (0.0695)	4.57 (0.389)	24.7 (6.41)	2.9 (0.237)	12.9 (3.78)
out-deg log-variance	2.95 (0.0151)	2.92 (0.045)	2.51 (0.105)	2.43 (0.0437)	1.87 (0.111)
in-deg log-variance	2.01 (0.0118)	1.93 (0.028)	1.29 (0.0537)	1.68 (0.0294)	1.16 (0.0588)
max out-deg	30,616 (5,939)	3,682 (267)	381 (17.4)	3,277 (512)	330 (50.8)
max in-deg	4,007 (561)	463 (20.8)	73.6 (7.19)	450 (29)	50 (5.87)
Burr with discretization and truncation					
in-deg mean	40 (0.269)	39.9 (0.8)	35.2 (1.66)	20 (0.422)	9.84 (0.581)
in-deg hill exponent	2.31 (0.0687)	2.37 (0.217)	6.45 (1.95)	2.35 (0.227)	2.95 (0.884)
in-deg log-variance	2.02 (0.00805)	2.02 (0.0261)	1.93 (0.0744)	1.73 (0.0219)	1.4 (0.0578)
max in-deg	6,728 (3,722)	2,429 (673)	380 (18)	1,434 (630)	247 (69)
Burr with discretization only					
out-deg mean	40 (2.17)	40 (4.83)	40.1 (18)	20 (2.42)	9.96 (3.43)
in-deg mean	40 (0.267)	40 (0.871)	40 (2.7)	20 (0.441)	9.97 (0.665)
out-deg hill exponent	1.39 (0.0416)	1.41 (0.132)	1.67 (0.553)	1.41 (0.131)	1.66 (0.538)
in-deg hill exponent	2.31 (0.0691)	2.35 (0.225)	2.79 (0.929)	2.35 (0.225)	2.76 (0.903)
out-deg log-variance	2.95 (0.015)	2.96 (0.0484)	2.96 (0.151)	2.45 (0.0454)	1.99 (0.135)
in-deg log-variance	2.02 (0.00808)	2.02 (0.0256)	2.02 (0.0802)	1.73 (0.022)	1.40 (0.0589)
max out-deg	85,028 (171,357)	23,628 (34,725)	6,558 (16,108)	11,994 (17,927)	1,603 (2,777)
max in-deg	6,862 (4,178)	3,123 (2,303)	1,324 (932)	1,525 (998)	332 (226)
Burr					
out-deg mean	40 (3.41)	40 (5.13)	40 (17.7)	20 (2.33)	9.97 (3.80)
in-deg mean	40 (0.277)	40 (0.855)	40 (2.74)	20 (0.43)	9.99 (0.685)
out-deg hill exponent	1.39 (0.0409)	1.4 (0.132)	1.59 (0.529)	1.41 (0.132)	1.6 (0.543)
in-deg hill exponent	2.31 (0.0681)	2.34 (0.221)	2.66 (0.911)	2.33 (0.226)	2.64 (0.91)
out-deg log-variance	10 (0.0611)	10 (0.195)	10 (0.612)	10 (0.198)	10 (0.617)
in-deg log-variance	3 (0.0182)	3 (0.0583)	3 (0.183)	3 (0.0587)	3 (0.185)
max out-deg	84,604 (312,946)	23,879 (38,485)	6,519 (15,630)	11,901 (16,318)	1,581 (3,144)
max in-deg	6,899 (4,097)	3,108 (2,099)	1,337 (987)	1,563 (1,061)	334 (238)

Table 3: Ablation study of the effects of discretization, truncation, and reconciliation on degree sequences sampled from the Burr XII distribution $X \sim \text{BurrXII}(c, k, \lambda)$, with tail exponent $\alpha = ck$, log-variance $\nu = \text{Var}(\ln X) = \frac{\frac{\pi^2}{6} + \psi_1(k)}{c^2}$, and mean $\mu = \lambda k\, B\left(k - \frac{1}{c}, 1 + \frac{1}{c}\right)$. Degree sequences are generated by specifying the mean μ, with $\alpha = 3$, $\nu = 3$ for in-degrees and $\alpha = 2$, $\nu = 10$ for out-degrees. Each block reports averages and standard deviations (in parentheses) over 5000 sampled sequences under different combinations of engineering procedures. The table tracks how mean degree, maximum degree, Hill exponents, and log-variances change before and after these procedures, and how the resulting values deviate from their targets across network sizes. Because the out-degree sequence is sampled first and the in-degree sequence is reconciled to match it, reconciliation does not affect out-degree statistics; these are omitted from the "**with discretization and truncation**" block. Two regimes are considered: a fixed target mean degree $\mu = 40$, and a size-dependent mean $\bar{k} = 0.86N^{1/3}$ (rounded to the nearest 10). Without engineering procedures, the sampled sequences exactly match the specified mean $\mu = 40$ and log-variances (3 and 10, respectively), but the estimated Hill exponents differ from the specified values ($\alpha = 3$ and 2) due to the large log-variances (see Figures 6, 7 and Eq. (7)). With discretization, truncation, and reconciliation, systematic distortions emerge, especially in smaller networks: as N decreases, Hill exponents increase and log-variances decrease. These results explain the deviations observed in Figure 12, including an upward bias in tail exponents and downward deviation in mean degree for small N.

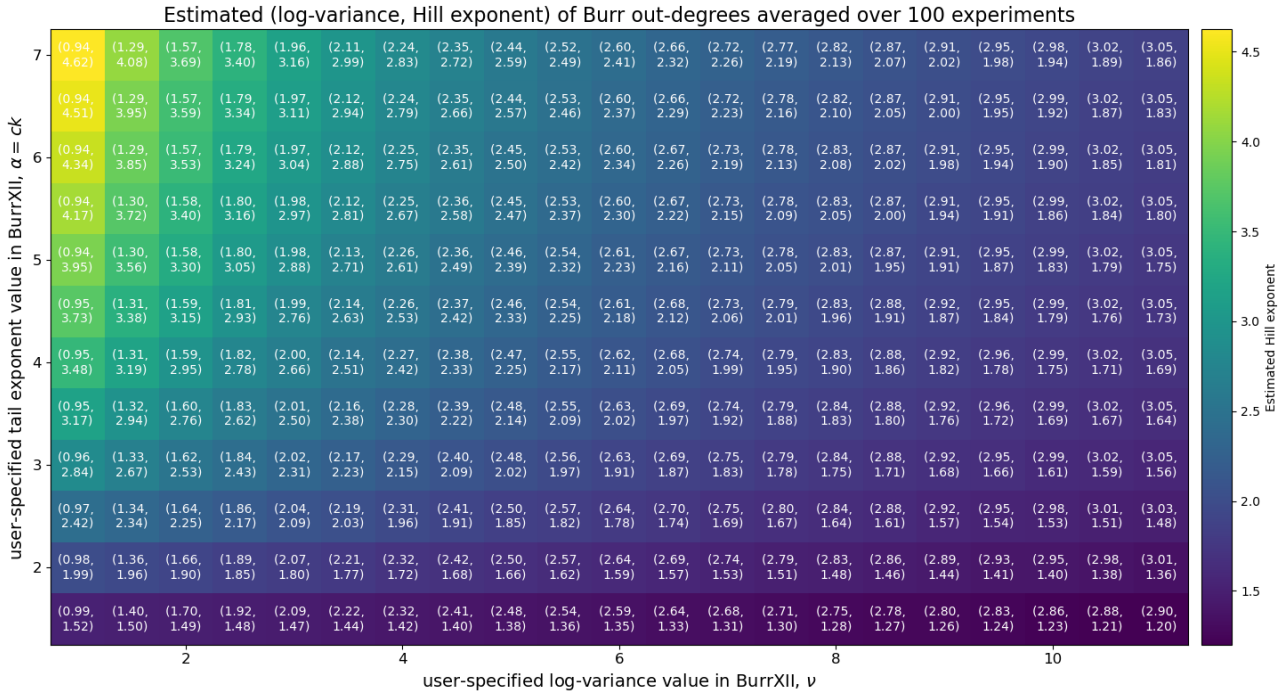


Figure 6: Meshgrid with step size 0.5 over input tail exponent α and log-variance ν for out-degrees. For each grid point, we generate 5,000 independent sequences of size 100,000. The realized Hill exponents and log-variances are displayed in white. We select $(\alpha, \nu) = (2, 10)$ as input values, as they yield a realized Hill exponent of 1.4 and log-variance of 2.95. We deliberately select an α that produces a realized Hill exponent below the empirical value of 1.5, since sampling adjacency matrices via the Configuration Model induces an upward bias, increasing the final Hill exponent in the synthetic network from 1.4 to 1.5. Since the Configuration Model does not affect the log-variance, ν is chosen to match the empirical value of 3.0 as closely as possible.

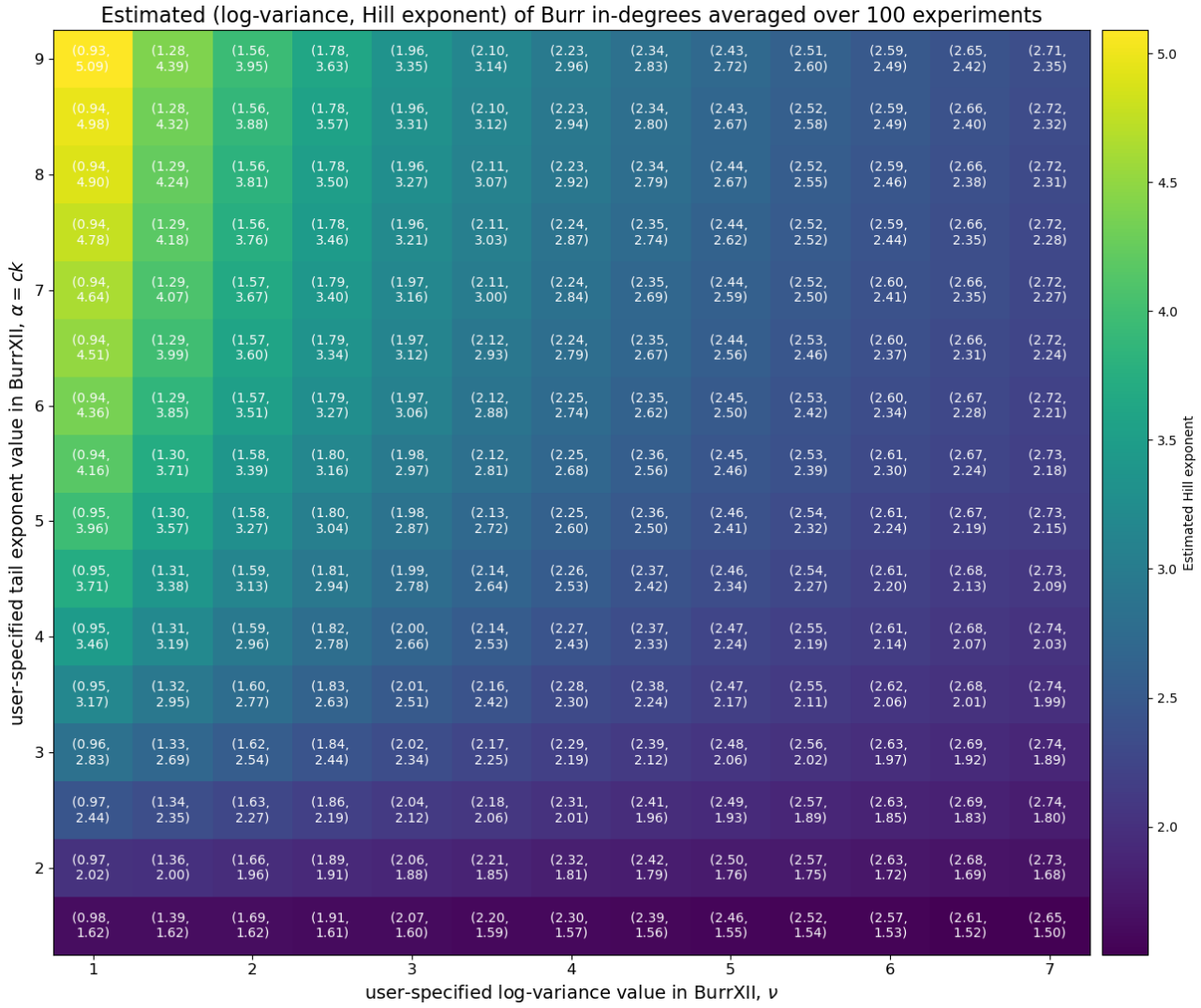


Figure 7: Meshgrid with step size 0.5 over input tail exponent α and log-variance ν for in-degrees. For each grid point, we generate 5,000 independent sequences of size 100,000. The estimated Hill exponents and log-variances are displayed in white. We select $(\alpha, \nu) = (3, 3)$ as input values, as they give an estimated Hill exponent of 2.34 and log-variance of 2.02. We deliberately select an α that yields a Hill exponent below the empirical value of 2.5, since sampling adjacency matrices via the Configuration Model induces an upward bias, bringing the final Hill exponent in the synthetic network from 2.34 to 2.5. Since the Configuration Model does not affect the log-variance, ν is chosen to match the empirical value of 2.0 as closely as possible.

This is equivalent to sampling a multinomial random vector with $|\Delta|$ trials and cell probabilities $\{p_i\}_{i=1}^N$. Let $(d_1,\dots,d_N)$ denote the resulting nonnegative counts, with $\sum_{i=1}^N d_i = |\Delta|$. If $\Delta > 0$, we update $k_i^{\text{in}} \leftarrow k_i^{\text{in}} + d_i$ for each i; if $\Delta < 0$, we update $k_i^{\text{in}} \leftarrow k_i^{\text{in}} - d_i$. Equivalently, each of the $|\Delta|$ draws increments (when $\Delta > 0$) or decrements (when $\Delta < 0$) the selected firm's in-degree by one, so that the total in-degree changes by exactly Δ. Because decrements can concentrate on firms with small degrees when $\Delta < 0$, some entries can become negative. In such cases, or if any additional sampling constraints described below are violated, we discard the sequence and repeat the procedure by drawing a new truncated Burr in-degree sequence, and apply the reconciliation again.

For networks with $N = 100,000$ firms, we only retain a reconciled in-degree sequence with tail exponent in the range $2.3 < \alpha < 2.4$, and[2] log-variance $\nu > 1.95$. Unlike the out-degree sequence, we do not impose a lower bound for the maximum in-degree.

B.4 Reordering Burr in-degrees and out-degrees

Since the firms' in- and out-degree sequences are drawn from independent Burr XII distributions, there is zero correlation between them. We could order both distributions before matching them, so there would be a perfect correlation between in- and out-degrees. In practice, however, we want to match an empirical correlation of around 0.55 [17]. To do this, we match the ordered out-degree sequence not to the perfectly ordered in-degree sequence, but to an imperfectly ordered in-degree sequence. Specifically, we will obtain this imperfect ordering as the perfect ordering of a randomly scrambled sequence. In details, the procedure works as follows.

Given the in- and out-degree sequences, we rank every firm according to its in-degree,

$$r_i = \text{rank}(k_i^{\text{in}}) \in \{0,\dots,N-1\}, \quad \tilde{r}_i = \frac{r_i}{N-1} \in [0,1].$$

Then, we multiply each in-degree k_i^{in} by a lognormal noise of unit mean and rank-dependent log-variance,

$$k_i^{\text{in, scrambled}} = k_i^{\text{in}} \times \varepsilon_i, \text{ where } \varepsilon_i \sim \text{Lognormal}\left(-\frac{1}{2}(1-\tilde{r}_i)^{2\zeta}, (1-\tilde{r}_i)^{2\zeta}\right),$$

noting that we use $\epsilon_i = 1$ for the top ranked firm ($\tilde{r}_i = 1$). Finally, we reorder the out-degree sequence to align with the *scrambled* in-degrees' ranking. Therefore, the *sorted* out-degrees are assigned to firms according to a permutation: the firm with smallest scrambled in-degree receives the smallest out-degree, the firm with the second-smallest scrambled in-degree receives the second-smallest out-degree, and so on. We calibrate the parameter ζ to ensure the Pearson correlation between the log in- and log out-degrees matches the empirical value of 0.55 reported by [17]. A reasonable value of ζ is -0.14.

B.5 Sampling the binary network

With the two degree distributions in hand, we are ready to generate a binary network. Using the `igraph` package, we sample an adjacency matrix $\mathbf{A}$ from the configuration model with the in- and reordered out-degree sequences as inputs. We remove parallel edges and self-loops.

This procedure induces an upward bias in the Hill exponents of in- and out-degrees computed from $\mathbf{A}$. To compensate for this, we deliberately choose Burr XII input parameters in Sections B.1 and B.2 that produce lower tail exponents prior to applying the configuration model (1.4 for out-degrees and 2.35 for in-degrees). After sampling, the resulting tail exponents are approximately 1.5 and 2.5, closely matching the empirical targets.

This concludes our binary network generation procedure. We now explain how we assign weights.

C Generating firm-level weighted networks

Conditional on the binary network generated, we generate weights for each existing edge. We do this by first generating "fitnesses" (latent variables for each node), and then using these and node degrees in a gravity-like

[2] We use the same sampling constraints for $N = 50,000$ firms as with 100,000 firms. For 10,000 firms, we use log-variance $\nu > 1.85$.

formula to determine edge-level weights,

$$\mathbf{W}_{ij}^{\text{init}} = (f_i^{\text{out}})^{\theta_{f,\text{out}}} (f_j^{\text{in}})^{\theta_{f,\text{in}}} (k_i^{\text{out}})^{\theta_{k,\text{out}}} (k_j^{\text{in}})^{\theta_{k,\text{in}}} \mathbf{A}_{ij}, \tag{8}$$

where f are the fitnesses and $\boldsymbol{\theta} = (\theta_{k,\text{in}}\ ,\ \theta_{k,\text{out}}\ ,\ \theta_{f,\text{in}}\ ,\ \theta_{f,\text{out}})$ denotes a set of tunable parameters. We then rescale $\mathbf{W}_{ij}^{\text{init}}$ to match the input-output table (IOT). We first describe how we determine the fitnesses, followed by how we rescale to match the IOT. Figure 8 outlines the model architecture.

C.1 Generate node-level fitnesses

Our goal is to introduce latent variables so that in Eq. 8, weights depend not only on degrees but also on an additional dimension of node heterogeneity. We have found that sampling the fitnesses from a log-normal distribution works well,

$$\begin{pmatrix} \ln f^{\text{in}} \\ \ln f^{\text{out}} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_f, \Sigma),$$

where $\boldsymbol{\mu}_f$ and Σ denote the mean vector and covariance matrix of the log fitnesses.

To determine its mean, note that because what matters is only heterogeneity, the mean of the latent variables will not impact the results substantially; we arbitrarily use $\boldsymbol{\mu}_f = (-8\ln 10, -8\ln 10)$.

For the variance, note that the fitnesses enter Eq. 8 as $(f^{\text{out}})^{\theta_{f,\text{out}}}$ and $(f^{\text{in}})^{\theta_{f,\text{in}}}$. Thus, the variance of the powered fitness terms can be adjusted by changing $\theta_{f,\text{out}}$ and $\theta_{f,\text{in}}$, which are optimized later. To facilitate parameter search, we want to ensure that all four θs are likely to be on similar scales. Intuitively, we expect the four θs to be on similar scales if the four variables are also on similar scales. Therefore, we assume that the variances of the log in- and log out-fitnesses are equal to the variances of the log in- and log out-degrees. Finally, for the covariance, we introduce a parameter ρ that tunes the correlations. Summing up, we use

$$\Sigma = \begin{pmatrix} \text{Var}(\ln k^{\text{in}}) & \rho * \sqrt{\text{Var}(\ln k^{\text{in}}) * \text{Var}(\ln k^{\text{out}})} \\ \rho * \sqrt{\text{Var}(\ln k^{\text{in}}) * \text{Var}(\ln k^{\text{out}})} & \text{Var}(\ln k^{\text{out}}) \end{pmatrix}. \tag{9}$$

We set $\rho = 0.8$. After rescaling to match the IOT, the resulting log in- and log out-strength correlation is close to the empirical value of 0.5. Finally, to ensure that total in-fitness equals total out-fitness, we scale every out-fitness by the factor $\frac{\sum f^{\text{in}}}{\sum f^{\text{out}}}$.

Given the degrees and fitnesses, and assuming specific values for the parameter vector $\boldsymbol{\theta}$, we obtain the weighted adjacency matrix by direct calculation from Eq. 8. Before we describe how we optimize for $\boldsymbol{\theta}$, we explain the final step of the process, assigning firms to industries and rescaling weights to ensure a match with IOTs.

C.2 Rescaling a weighted network to match an input-output table

We start from a firm-level weighted network $\mathbf{W}^{\text{init}}$ consisting of N firms. We assume that the industries have a number of firms proportional to their industry sales. We first compute the preliminary industry count as

$$\tilde{M}_s = \texttt{round}\left(N \frac{\sum_{n=1}^{M} \text{IOT}_{sn}}{\sum_{m=1}^{M} \sum_{n=1}^{M} \text{IOT}_{mn}} \right),$$

where IOT_{mn} denotes the monetary flow from industry m to industry n, obtained from OECD input-output tables, and round denotes rounding to the nearest integer. Due to rounding, the sum $\sum_s \tilde{M}_s$ may be either smaller or larger than N. We therefore obtain the adjusted industry counts M_s by repeatedly adding one count to, or subtracting one count from, a uniformly randomly selected industry among those with positive current counts, until $\sum_s M_s = N$. We then assign firms sequentially according to these adjusted counts: the first M_1 firms are assigned to industry 1, the next M_2 firms to industry 2, and so on. This ensures that each firm is assigned to exactly one industry. Note that this way of assigning firms to industries is technically stochastic.

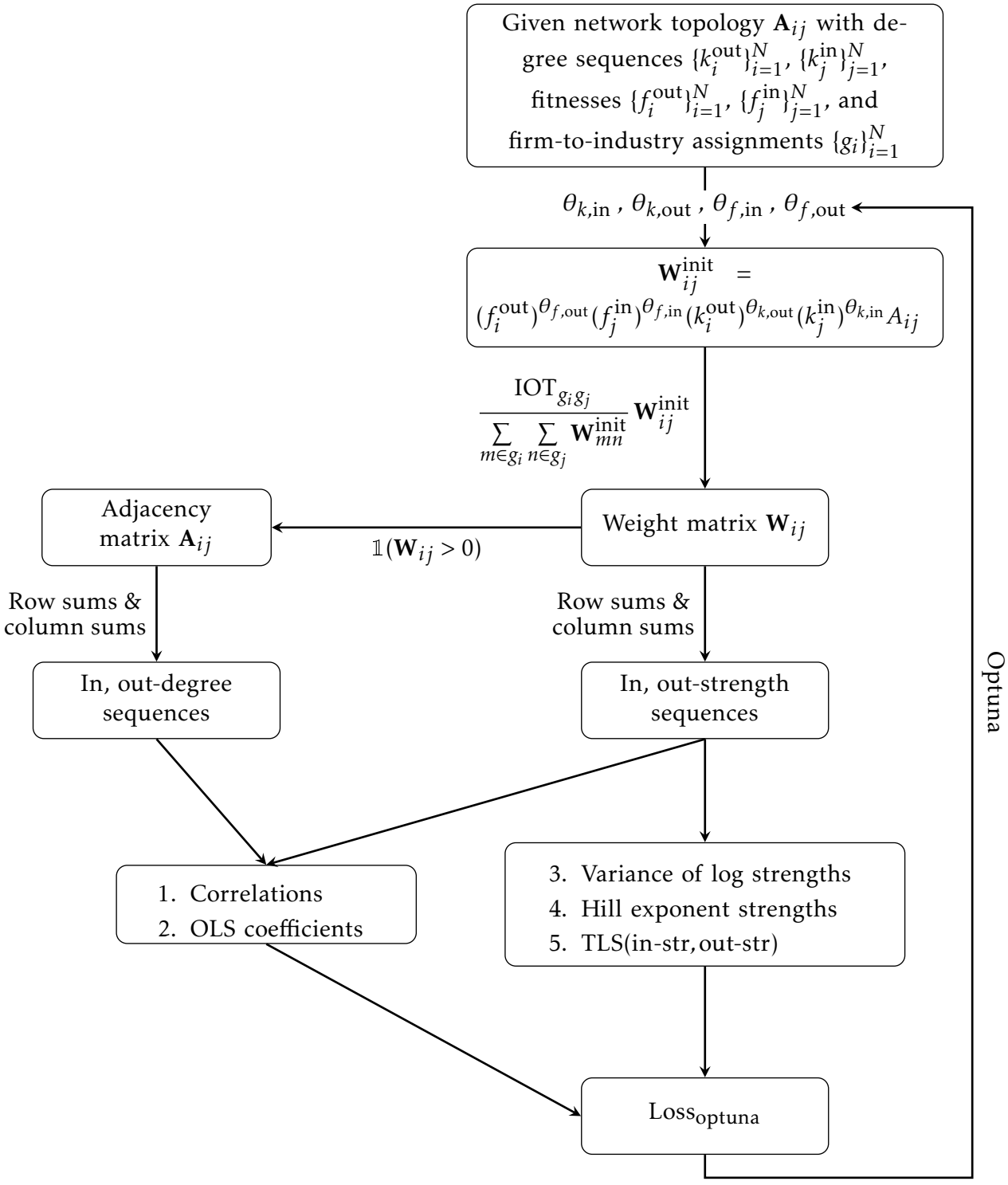


Figure 8: Architecture for the optimization of parameters $\theta_{k,\text{in}}$, $\theta_{k,\text{out}}$, $\theta_{f,\text{in}}$, $\theta_{f,\text{out}}$ in the generation of synthetic supply networks. Here $\text{TLS}(y,x)$ represents the total least squares regression coefficient of $\ln y$ against $\ln x$.

While proportional allocation appears as a reasonable assumption, our architecture is flexible enough to accommodate user-provided firm-to-industry mappings when such information is available.

With firms assigned to industries, we are ready to rescale the firm-to-firm weights to ensure that the industry-to-industry weights match a given IOT, using

$$\mathbf{W}_{ij} = \frac{\mathrm{IOT}_{g_i g_j}}{\sum_{f \in g_i} \sum_{h \in g_j} \mathbf{W}^{\mathrm{init}}_{fh}} \mathbf{W}^{\mathrm{init}}_{ij},$$

where index g_i corresponds to the industry of firm i. It is straightforward to check that the rescaled network $\mathbf{W}$ aggregates exactly into the IO table,

$$\sum_{f \in g_i, h \in g_j} \mathbf{W}_{fh} = \mathrm{IOT}_{g_i g_j}.$$

For computational purposes, the rescaling can be written compactly in matrix notation. Defining the $M \times N$ aggregator matrix $\mathbf{S}$ as

$$\mathbf{S}_{uj} = \begin{cases} 1 & \text{if firm } j \text{ belongs to industry } u \\ 0 & \text{otherwise} \end{cases},$$

the rescaled weighted network is given by

$$\mathbf{W} = \mathbf{S}^\top \left(\mathrm{IOT} \oslash \left[\mathbf{S} \mathbf{W}^{\mathrm{init}} \mathbf{S}^\top \right] \right) \mathbf{S} \odot \mathbf{W}^{\mathrm{init}},$$

where $\oslash$ denotes Hadamard (element-wise) division, $\odot$ denotes the Hadamard (element-wise) product, and $\top$ denotes the transpose operation. The industry-level aggregation of $\mathbf{W}^{\mathrm{init}}$ is given by the matrix product $\mathbf{S}\mathbf{W}^{\mathrm{init}}\mathbf{S}^\top$, see Chapter 4.9 of [58]. The Hadamard division of $\mathrm{IOT} \oslash \left[\mathbf{S}\mathbf{W}^{\mathrm{init}}\mathbf{S}^\top \right]$ yields an $M \times M$ matrix of scaling factors, where each entry represents the ratio between the corresponding value in the empirical IOT and the respective value in the aggregated network $\mathbf{S}\mathbf{W}^{\mathrm{init}}\mathbf{S}^\top$. Finally, $\mathbf{S}^\top \left(\mathrm{IOT} \oslash \left[\mathbf{S}\mathbf{W}^{\mathrm{init}}\mathbf{S}^\top \right] \right) \mathbf{S}$ disaggregates the scaling factors into an $N \times N$ matrix, which are subsequently multiplied element-wise with the original weighted matrix $\mathbf{W}^{\mathrm{init}}$: that is, we scale each element (i, j) entry of $\mathbf{W}^{\mathrm{init}}$ by the ratio between the (g_i, g_j) entry of the specified IO table and the (g_i, g_j) entry of the aggregation of $\mathbf{W}^{\mathrm{init}}$.

This concludes the description of the full weight allocation pipeline, although we have so far assumed that the parameters were given. The next section describes one of the most important steps of our method, namely how we determine the values of θs such that the properties of weighted network will match empirical real-world properties.

C.3 Calibrating parameters $\theta_{k,\mathrm{in}}, \theta_{k,\mathrm{out}}, \theta_{f,\mathrm{in}}, \theta_{f,\mathrm{out}}$

Calibrating the parameters $\boldsymbol{\theta} = (\theta_{k,\mathrm{in}}\ ,\ \theta_{k,\mathrm{out}}\ ,\ \theta_{f,\mathrm{in}}\ ,\ \theta_{f,\mathrm{out}})$ allows the synthetic supply network to capture micro-level features while remaining exactly consistent with the macro-level input-output table. We calibrate these parameters using Optuna [56], which is an open-source Python library for automated hyperparameter optimization that efficiently explores large parameter spaces by adaptively pruning unpromising candidates[3].

[3] We have explored several alternatives, including manual grid search, which quickly became too expensive but was helpful for us to understand the landscape, and PyTorch (Adam), which in our case appeared more sensitive to initial conditions and delivered lower quality results, but has been helpful for us to understand trade-offs in the multi-objective loss function.

We minimize the loss function

$$\begin{aligned}\text{Loss} = &\mathcal{D}(\text{hill(in-str)}) \\ &+ \mathcal{D}(\text{hill(out-str)}) \\ &+ \mathcal{D}(\text{Corr(out-str, out-deg)}) \\ &+ \mathcal{D}(\text{Corr(in-str, in-deg)}) \\ &+ \mathcal{D}(\text{OLS(in-str, in-deg)}) \\ &+ \mathcal{D}(\text{OLS(in-deg,in-str)}) \\ &+ \mathcal{D}(\text{OLS(out-deg,out-str)}) \\ &+ \mathcal{D}(\text{OLS(out-str, out-deg)}) \\ &+ \mathcal{D}(\text{TLS(in-str, out-str)}) \\ &+ 10^{-3} \times \mathcal{D}(\text{Var(in-str)}) \\ &+ 10^{-3} \times \mathcal{D}(\text{Var(out-str)})\end{aligned} \tag{10}$$

where, for any statistic q, $\mathcal{D}(q) = \left[\widehat{q}(\boldsymbol{\theta}) - q^{\star}\right]^2$ denotes the squared discrepancy between its value in the synthetic network, $\widehat{q}(\boldsymbol{\theta})$, and its empirical target, $q^{\star}$. Here, OLS(y, x) and TLS(y, x) denote the ordinary least squares and total least squares regressions of y against x respectively. Tail exponents are computed using the Hill estimator, Corr denotes Pearson correlation, and OLS and TLS refer to ordinary and total least squares respectively. All correlations, OLS and TLS coefficients, and variance terms are computed on log-transformed variables. All loss components are equally weighted, except for the strengths variances, which receive a weight of 10^{-3}. Figure 8 illustrates this optimization procedure.

To be clear, the θ parameters affect only the weights assigned to existing edges, while the network topology $\mathbf{A}_{ij}$ remains fixed. Thus, this optimization does not include terms targeting binary network properties.

We run Optuna for a fixed number of trials, which determines the search duration. In our implementation, we use 200 trials and search over the parameter space

$$\theta_{k,\text{in}}\ ,\ \theta_{k,\text{out}}\ ,\ \theta_{f,\text{in}}\ ,\ \theta_{f,\text{out}} \in [-2, 2]^4,$$

which we find is sufficiently broad to identify optimal parameters[4]. On a machine with an `Intel Xeon Gold` 6226R processor (12 cores, 2.90 GHz) and 125GB RAM, a single search procedure takes 7 minutes.

Figure 9 illustrates the best parameter tuples by repeating the parameter search procedure 200 times, where each search consists of 200 trials. Table 4 presents the mean and standard deviation of the best parameters.

We observe several peculiarities: the optimal value of $\theta_{k,\text{in}}$ is centered around 0, while $\theta_{k,\text{out}}$ is consistently negative, with a relatively precise estimate around the interval $[-0.8, -0.6]$. Instead, the parameters appear relatively precisely estimated, baring an identification problem: $\theta_{f,\text{in}}$ and $\theta_{f,\text{out}}$ are either both negative or both positive, with equal frequency, and similar means.

Parameter	Mean	Standard deviation
$\theta_{k,\text{in}}$	-0.0265	0.0757
$\theta_{k,\text{out}}$	-0.697	0.0644
$\theta_{f,\text{in}}$	-1.35, 1.37	0.11, 0.12
$\theta_{f,\text{out}}$	-1.27, 1.29	0.0756, 0.0623

Table 4: Mean and standard deviation of the optimal parameters obtained from Optuna. The statistics are computed from 200 independent runs, each experiment consisting of 200 trials. The parameter $\theta_{k,\text{in}}$ is centered around 0, while $\theta_{k,\text{out}}$ is consistently negative. In contrast, $\theta_{f,\text{in}}$ and $\theta_{f,\text{out}}$ are either both negative or positive (see Figure 9), so we report both cases separately.

Given these results, in the main paper, we use the values $(\theta_{k,\text{in}}\ ,\ \theta_{k,\text{out}}\ ,\ \theta_{f,\text{in}}\ ,\ \theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$.

[4]As a robustness check, we searched over different parameter spaces in Optuna, the widest being $[-4, 4]^4$, and obtained the same behavior of optimal parameters.

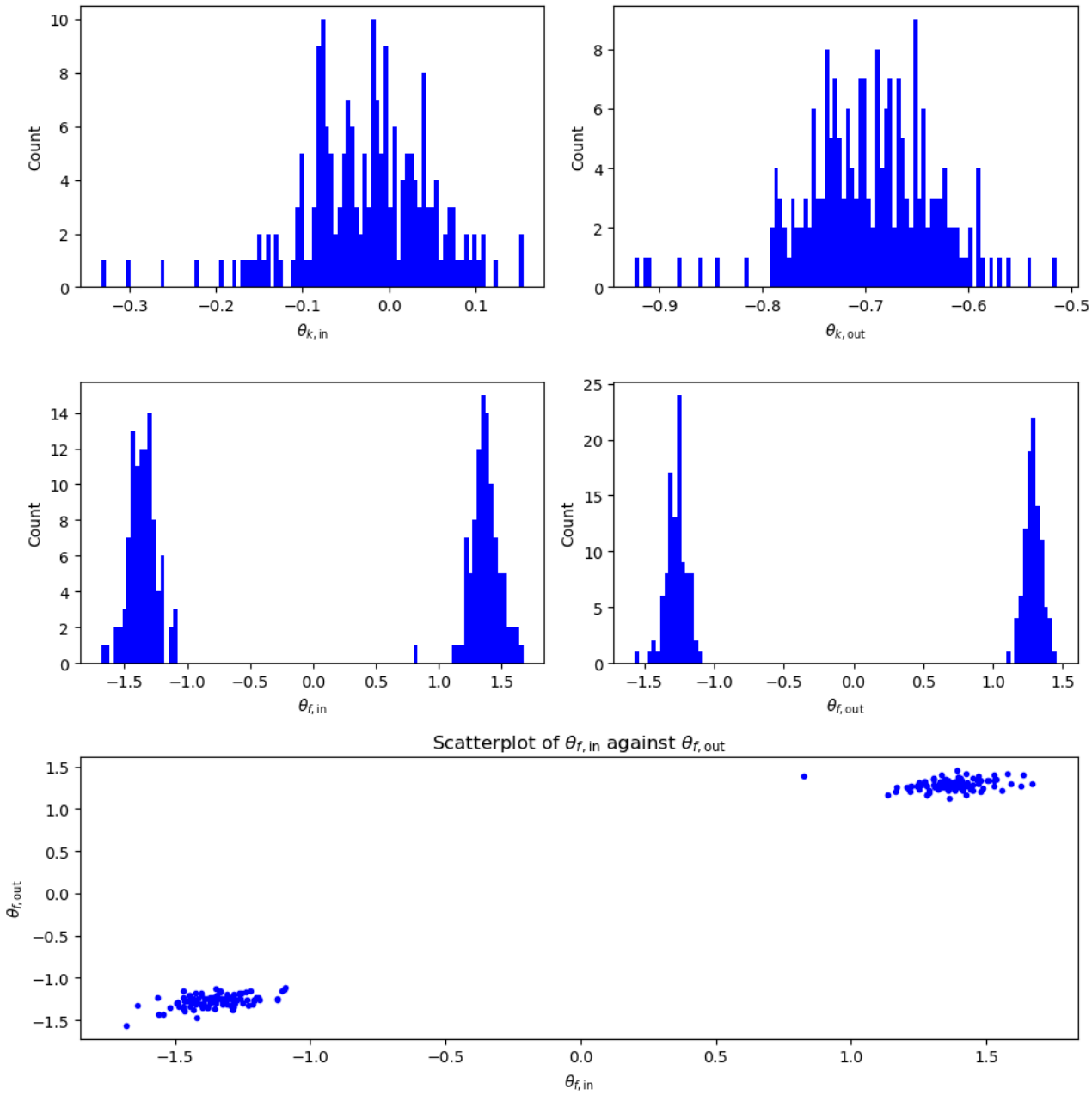


Figure 9: Distribution of optimal parameters identified using Optuna. The parameter search procedure is repeated 200 times, and the histograms report the optimal values obtained in each run. The estimates of $\theta_{k,\text{in}}$ are centered around zero, while $\theta_{k,\text{out}}$ is consistently negative. $\theta_{f,\text{in}}$ and $\theta_{f,\text{out}}$ exhibit bimodal distributions. The scatterplot in the bottom panel shows two distinct clusters, indicating that $\theta_{f,\text{in}}$ and $\theta_{f,\text{out}}$ are either both negative or both positive: positive optimal parameters occurred 101 times and negative parameters 99, i.e. both cases occur equally often. The means of these optimal parameters are reported in Table 4.

Note that these results are likely to be of independent interest, as other researchers could use our parametrization of the latent variables, Eq. 8, and these values of θs to assign weights to binary networks obtained using different methods, and likely achieve a good match to desirable weighted network properties.

To investigate more carefully whether these values (and our entire pipeline) work well in other context, the next section shows the robustness of the result to networks of different sizes, and calibrated on different IOTs.

D Robustness checks

We assess the robustness of our methodology in the following ways:

1. **Robustness against stochastic variation in the pipeline**: Generate synthetic networks using the optimal parameters identified by Optuna, where each network uses a different realization of degree sequences, fitnesses, adjacency matrices $\mathbf{A}_{ij}$, and firm-to-industry assignments.

2. **Robustness against small changes in θ parameter values**: Repeat the above by using the optimal parameters rounded to one decimal place.

3. **Robustness against choice of country.** Using the rounded optimal parameters, generate synthetic networks for different country input-output tables.

4. **Robustness against choice of number of firms.** Using the rounded optimal parameters, generate synthetic networks of varying network sizes.

D.1 Robustness of generated networks

Table 5 reports the network properties for the first two robustness checks, based on 100 networks of size $100,000$ per parameter setting using the 2015 Hungarian input-output table. The column "Positive" corresponds to networks generated with positive fitness parameters,

$$(\theta_{k,\text{in}}\,,\,\theta_{k,\text{out}}\,,\,\theta_{f,\text{in}}\,,\,\theta_{f,\text{out}}) = (-0.0265, -0.697, 1.37, 1.29),$$

while "Negative" corresponds to networks generated using negative fitness parameters,

$$(\theta_{k,\text{in}}\,,\,\theta_{k,\text{out}}\,,\,\theta_{f,\text{in}}\,,\,\theta_{f,\text{out}}) = (-0.0265, -0.697, -1.35, -1.27).$$

The column "Rounded negative" uses the rounded negative parameters,

$$(\theta_{k,\text{in}}\,,\,\theta_{k,\text{out}}\,,\,\theta_{f,\text{in}}\,,\,\theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3).$$

Figures 10 shows the corresponding distributional properties for the "positive" case (very similar results hold for the other two).

Across runs, network properties have low variation, with standard deviations typically one or even two orders of magnitude smaller than their means. For example, under the column "Negative", the in-degree Hill exponent has a mean of 2.54 and a standard deviation of 0.03, close to the target value of 2.5.

Moreover, consistent with empirical evidence from Value-Added Tax (VAT) data [17], the asymmetry between the in-degree and out-degree complementary cumulative distribution functions (CCDFs) is clearly evident, with a maximum out-degree approximately an order of magnitude larger than the maximum in-degree.

In contrast, the in-strength and out-strength CCDFs are indistinguishable, again consistent with [17].

The panel of 2D histograms further confirm the stability of joint distributions across experiments.

Overall, the results demonstrate that, for a fixed set of parameters $(\theta_{k,\text{in}}\,,\,\theta_{k,\text{out}}\,,\,\theta_{f,\text{in}}\,,\,\theta_{f,\text{out}})$, the resulting distributional and network properties remain stable across different realizations of degree sequences, fitnesses, draws from the Configuration Model, and firm-to-industry assignments. This robustness implies that the calibrated parameters can be reused to generate synthetic networks without running Optuna. For ease of use, we provide default values of $(\theta_{k,\text{in}}\,,\,\theta_{k,\text{out}}\,,\,\theta_{f,\text{in}}\,,\,\theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$ in the codebase.

Property	Positive	Negative	Rounded negative	Empirical
$\theta_{k,\text{in}}$	-0.0265	-0.0265	0	
$\theta_{k,\text{out}}$	-0.697	-0.697	-0.7	
$\theta_{f,\text{in}}$	1.37	-1.35	-1.3	
$\theta_{f,\text{out}}$	1.29	-1.27	-1.3	
Number of nodes	100,000 (0)	100,000 (0)	100,000 (0)	100,000
Number of edges	3,855,343 (23,605)	3,857,177 (24,644)	3,862,201 (26,739)	4,000,000
Share of supplier-only firms	0 (0)	0 (0)	0 (0)	
Share of customer-only firms	26.3 (0.139)	26.3 (0.16)	26.3 (0.146)	
Mean degree	38.6 (0.236)	38.6 (0.246)	38.6 (0.267)	40
Max k^{in}	2,787 (360)	2,823 (294)	2,847 (328)	
Max k^{out}	21,256 (1,837)	20,786 (1,994)	21,076 (2,160)	
Mean log k^{in}	3.28 (0.00911)	3.28 (0.0088)	3.28 (0.00795)	3
Var log k^{in}	1.31 (0.0082)	1.31 (0.00859)	1.31 (0.00826)	2
Mean log k^{out}	2.19 (0.00621)	2.19 (0.00664)	2.19 (0.00654)	2
Var log k^{out}	2.96 (0.0111)	2.96 (0.0125)	2.97 (0.0124)	3
Mean log s^{in}	10.8 (0.0349)	10.8 (0.0424)	10.9 (0.0366)	10
Var log s^{in}	9.14 (0.102)	8.96 (0.0973)	8.9 (0.11)	9
Mean log s^{out}	10.8 (0.0399)	10.9 (0.0392)	10.9 (0.0405)	10
Var log s^{out}	9.51 (0.0901)	9.25 (0.0749)	9.25 (0.0939)	8
LWCC	95,517 (71)	95,525 (61)	95,526 (75)	
Avg path length	2.74 (0.0234)	2.75 (0.0186)	2.75 (0.0214)	3
Degree assortativity	−0.0679(0.00379)	−0.0668(0.00363)	−0.0668(0.00444)	$[-0.2, -0.01] < 0$
Reciprocity	0.00647 (0.000608)	0.00655 (0.000555)	0.00656 (0.000554)	[0.03, 0.05]
Avg clustering	0.0946 (0.00603)	0.0929 (0.00563)	0.0935 (0.0068)	[0.19, 0.28]
Global clustering	0.0217 (0.000777)	0.0218 (0.000707)	0.0218 (0.000868)	low
Hill α, k^{in}	2.55 (0.0321)	2.54 (0.0328)	2.54 (0.0306)	2.5
Hill α, k^{out}	1.46 (0.0144)	1.47 (0.0177)	1.47 (0.0162)	1.5
Hill α, s^{in}	1.06 (0.0215)	1.07 (0.0223)	1.08 (0.0193)	1
Hill α, s^{out}	1.07 (0.0201)	1.08 (0.0229)	1.07 (0.0199)	1
Hill α, influence	1.45 (0.0223)	1.46 (0.0224)	1.44 (0.0219)	[1.2, 1.3]
Hill α, weights	1.06 (0.0149)	1.07 (0.016)	1.07 (0.0154)	[1.1, 1.2]
Corr: $k^{\text{in}} \sim k^{\text{out}}$	0.551 (0.00296)	0.551 (0.00294)	0.551 (0.00281)	0.55
Corr: $s^{\text{in}} \sim s^{\text{out}}$	0.508 (0.00485)	0.506 (0.0044)	0.502 (0.0051)	0.5
Corr: $s^{\text{out}} \sim k^{\text{out}}$	0.453 (0.00381)	0.455 (0.00311)	0.44 (0.00443)	0.5
Corr: $s^{\text{in}} \sim k^{\text{in}}$	0.565 (0.00352)	0.567 (0.00332)	0.584 (0.0028)	0.75
OLS: $s^{\text{out}} \sim k^{\text{out}}$	0.812 (0.00947)	0.803 (0.00746)	0.777 (0.0101)	0.76
OLS: $k^{\text{out}} \sim s^{\text{out}}$	0.253 (0.00171)	0.257 (0.00164)	0.249 (0.00218)	0.33
OLS: $s^{\text{in}} \sim k^{\text{in}}$	1.49 (0.0112)	1.48 (0.011)	1.52 (0.0122)	1.4
OLS: $k^{\text{in}} \sim s^{\text{in}}$	0.214 (0.00207)	0.217 (0.00165)	0.224 (0.00164)	0.4
TLS: $s^{\text{in}} \sim s^{\text{out}}$	0.962 (0.00945)	0.969 (0.00995)	0.962 (0.0127)	1
TLS: $k^{\text{in}} \sim k^{\text{out}}$	0.494 (0.0036)	0.494 (0.00373)	0.494 (0.00339)	0.7
IOT RMSE (100 MUSD)	0.00134 (0.000947)	0.00137 (0.000818)	0.00138 (0.00085)	

Table 5: Properties of the networks generated under different parameter settings ($\theta_{k,\text{in}}$, $\theta_{k,\text{out}}$, $\theta_{f,\text{in}}$, $\theta_{f,\text{out}}$), based on 100 networks of 100,000 firms that match the 2015 Hungarian input-output table. Entries report means, with standard deviations in parentheses. "Positive" and "Negative" denote parameter settings with positive and negative fitness respectively, while "Rounded negative" uses rounded values of the negative setting. In-degree is the number of suppliers, out-degree is the number of customers. Supplier-only firms sell to at least one other firm but buy from none ($k^{\text{in}} = 0$), whereas customer-only firms buy from at least one other firm but sell to none ($k^{\text{out}} = 0$). LWCC stands for Largest Weakly Connected Component. Mean of both log in- and out-strengths are computed in units of 1 USD. OLS and TLS stand for Ordinary and Total Least Squares, and α stands for a tail exponent computed using a Hill estimator. All values are reported to three significant figures, except for the number of nodes, edges, and LWCC, which are reported as integers. All three parameter settings closely match the empirical targets with low variability. The R^2 coefficient is obtained from $R^2 = \text{Corr}(y, x)^2$ for OLS: $y \sim x$. The RMSE of the aggregated matrix is negligible.

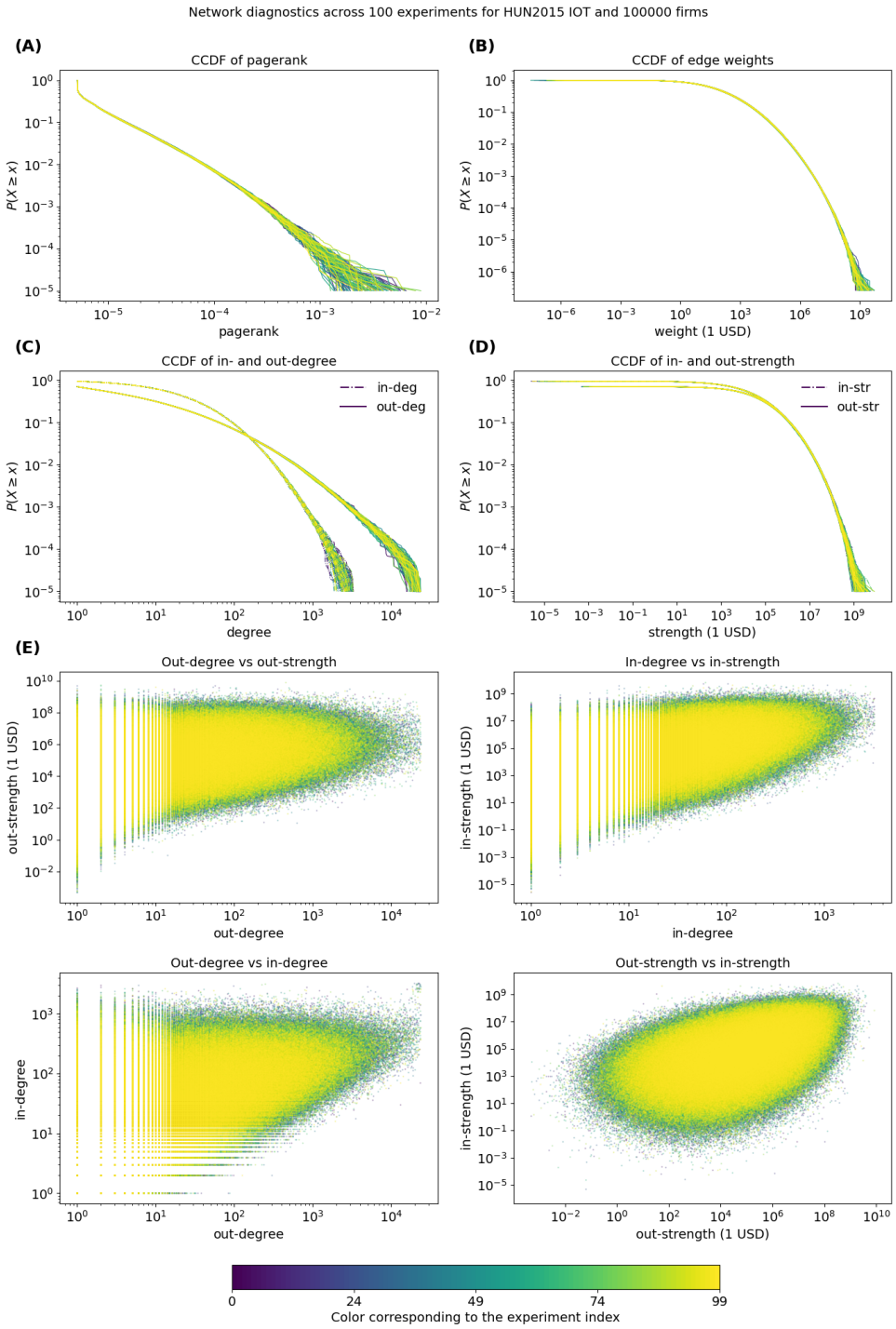


Figure 10: Distributional properties of 100 networks generated using $(\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}}) = (-0.0265, -0.697, 1.37, 1.29)$, with $\theta_{f,\text{in}}$, $\theta_{f,\text{out}}$ both positive. Randomness is introduced from different degree sequences, fitnesses, draw of adjacency matrix from the configuration model, and firm-to-industry assignment. The Figure shows that the networks are very similar across experiments. (A) Complementary cumulative distribution function (CCDF) of PageRank. (B) CCDF of edge weights. (C) CCDF of in- and out-degrees. (D) CCDF of in- and out-strengths. (E) 2D histograms of the four joint distributions consisting of out-strength against out-degree, in-strength against in-degree, in-degree against out-degree, and out-strength against in-strength.

D.2 Robustness across different countries' IOT

We examine the stability of networks generated for different country input-output tables, namely Belgium, Hungary, and Brunei Darussalam for the year 2015. For each country, we generate 40 networks of 100,000 firms using the rounded negative parameters $(\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$.

We choose Belgium and Hungary because their firm-level VAT dataset has been well-studied [17]. We chose Brunei Darussalam to provide maximum contrast: its IOT exhibits the largest deviation from Hungary among the available OECD datasets[5].

As illustrated in Figure 11, we find that the network properties are similar between countries. This demonstrates that our methodology generalizes across countries while preserving key empirical firm-level characteristics. By construction, all synthetic networks aggregate closely to their target input-output tables.

D.3 Robustness across different network sizes

Figure 12 examines how network properties vary with network size for 10,000, 50,000, 100,000, and 500,000 firms. For each network size, we generate 40 networks using the rounded negative parameters $(\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$, where all networks match the 2015 Hungarian input-output table. We fix the mean degree parameter at 40, that is, we set $\mu = 40$ when sampling Burr degrees (see Eq. (6)).

As network size increases, degree disassortativity (in absolute value), reciprocity, average clustering, and global clustering all decline. This is expected because adjacency matrices are sampled from the Configuration Model, under which the global clustering coefficient scales inversely with network size [43].

Two systematic patterns emerge in smaller networks: upward-biased degree Hill exponents and greater deviation of the mean degree from its target. Both are explained by the ablation study in Table 3. For a fixed mean of 40, the block "Burr with discretization, truncation, and reconciliation" shows that, as network size decreases from 100,000 to 1,000, the Hill exponent increases from 1.4 to 5.83 for out-degrees and from 2.33 to 24.7 for in-degrees. This reflects the use of a fixed tail probability $p = 0.01$ when estimating the Hill exponents of degree distributions, which leads to fewer tail observations in smaller networks and hence an upward bias. For the same range of network sizes and fixed mean degree parameter $\mu = 40$, the mean degree falls from 39 to 19.6, indicating increasing deviation.

In contrast, regression-based properties remain stable across network sizes, indicating that the optimized parameters $(\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}})$ generalize well across scales.

All synthetic networks aggregate closely to the 2015 Hungarian IOT by construction, although aggregation accuracy improves as the network size increases.

Taken together, these results highlight the importance of choosing an appropriate network size to reproduce the key empirical firm-level characteristics, and motivate representing economic models at a $1:1$ scale.

[5] Measured by the column-normalized ℓ_1 distance $\sum_{m=1}^{45}\sum_{n=1}^{45}\left|\frac{\text{IOT}_{mn}}{\sum_{m=1}^{45}\text{IOT}_{mn}} - \frac{\text{IOT}_{mn}^{\text{Hungary}}}{\sum_{m=1}^{45}\text{IOT}_{mn}^{\text{Hungary}}}\right|$.

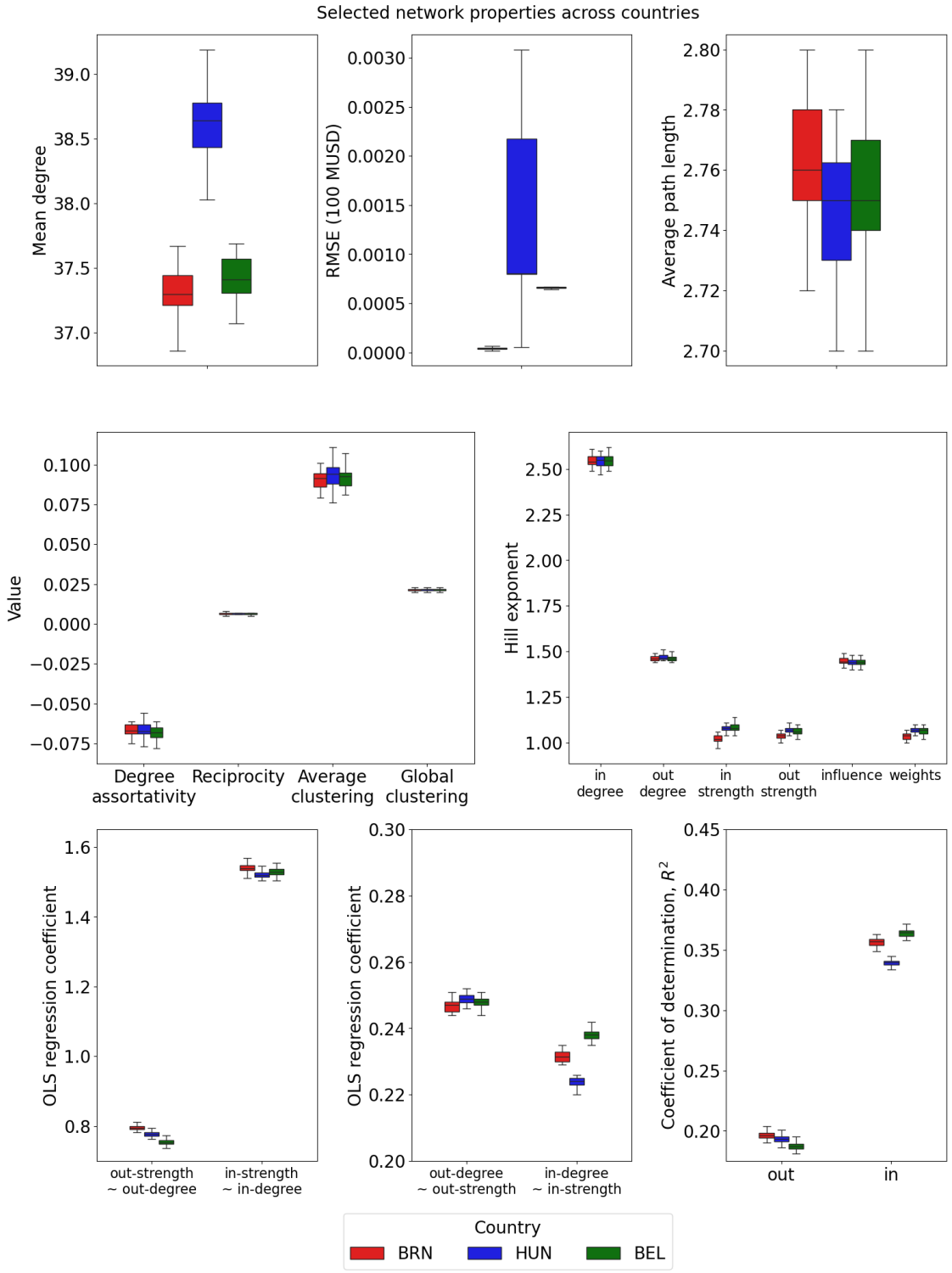


Figure 11: Network properties of generated networks that match the 2015 input-output tables of Brunei Darussalam, Hungary, and Belgium. For each country, 40 networks with 100,000 firms are generated using $(\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$, and the resulting properties are shown using boxplots. Network properties exhibit minimal variation across countries. Cross-country differences are visible but remain small relative to the relevant targets and scales: the mean degree stays close to the target value of 40, and the aggregated matrix has low RMSE relative to the scale of the input-output table.

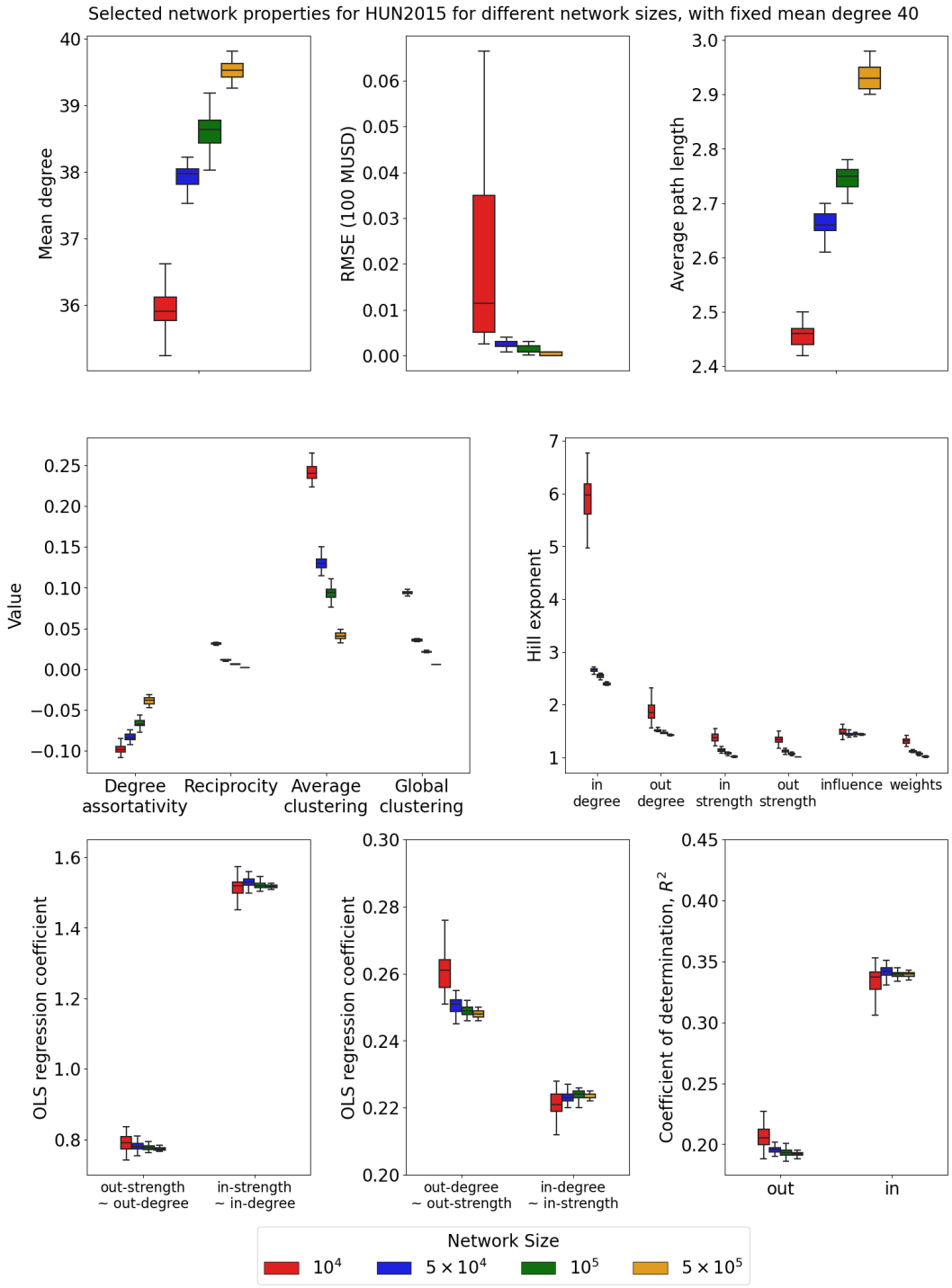


Figure 12: Network properties across different network sizes. Synthetic networks match the 2015 Hungarian input-output table. For each network size, we fix the mean degree parameter μ at 40 and generate 40 independent networks using the rounded negative parameters $(\theta_{k,\text{in}}, \theta_{k,\text{out}}, \theta_{f,\text{in}}, \theta_{f,\text{out}}) = (0, -0.7, -1.3, -1.3)$. Properties are summarized using boxplots. Systematic variation arises from discretization, truncation, and reconciliation of Burr XII draws. Smaller networks have upward-biased degree Hill exponents and greater deviation from the target mean degree. As network size increases, average path length rises, while degree disassortativity (in absolute value), reciprocity, clustering, and Hill exponents of degrees, strengths, influence, and weights decline. OLS coefficients remain stable across network sizes.

E Proofs

Lemma E.1. *Let U be a Uniform$[0,1]$ random variable. For $k > 0$, we have $1-(1-U)^{\frac{1}{k}} \overset{d}{=} Beta(1,k)$.*

Proof. Recall that the density function of a Beta$(1,k)$ random variable with support $x \in [0,1]$ is given by $k(1-x)^{k-1}$. Then we have,

$$\begin{aligned}\mathbb{P}\left(1-(1-U)^{\frac{1}{k}} \le u\right) &= \mathbb{P}\left(U \le 1-(1-u)^k\right)\\ &= 1-(1-u)^k.\end{aligned}$$

Differentiating with respect to u, the density function of $1-(1-U)^{\frac{1}{k}}$ is $k(1-u)^{k-1}$. ∎

Proposition E.2. *Let X be a BurrXII(c,k,λ) random variable with strictly positive parameters (c,k,λ), V be a Beta$(1,k)$ random variable, and U be a Uniform$[0,1]$ random variable. Then $\left(\frac{X}{\lambda}\right)^c \overset{d}{=} \frac{V}{1-V}$.*

Proof. By Eq. (4) and Lemma E.1, we have

$$\frac{V}{1-V} = \frac{1-(1-U)^{\frac{1}{k}}}{1-\left[1-(1-U)^{\frac{1}{k}}\right]} = \frac{1-(1-U)^{\frac{1}{k}}}{(1-U)^{\frac{1}{k}}} = (1-U)^{-\frac{1}{k}} - 1 = \left(\frac{X}{\lambda}\right)^c.$$

∎

Corollary E.2.1. *Denote a Burr XII random variable with strictly positive parameters (c,k,λ) by $X \sim BurrXII(c,k,\lambda)$ and suppose that $ck > 1$, then its expectation is*

$$\mathbb{E}[X] = \lambda k B\left(k-\frac{1}{c}, 1+\frac{1}{c}\right),$$

for $B(\cdot,\cdot)$ is the Beta function. Furthermore, the log-variance is given by

$$Var(\ln X) = \frac{\frac{\pi^2}{6} + \psi_1(k)}{c^2},$$

where $\psi_1(\cdot)$ is trigamma function.

Proof. For $V \sim \text{Beta}(1,k)$, we have

$$\left(\frac{X}{\lambda}\right)^c \overset{d}{=} \frac{V}{1-V} \iff X \overset{d}{=} \lambda\left(\frac{V}{1-V}\right)^{\frac{1}{c}},$$

then taking expectations, we have

$$\begin{aligned}\mathbb{E}[X] &= \lambda \mathbb{E}\left(\frac{V}{1-V}\right)^{\frac{1}{c}}\\ &= \lambda \int_0^1 \left(\frac{v}{1-v}\right)^{\frac{1}{c}} k(1-v)^{k-1}\, dv\\ &= \lambda k \int_0^1 v^{\frac{1}{c}} (1-v)^{k-1-\frac{1}{c}}\, dv\\ &= \lambda k B\left(k-\frac{1}{c}, 1+\frac{1}{c}\right).\end{aligned}$$

For the log-variance, taking log on both sides of $\left(\frac{X}{\lambda}\right)^c \overset{d}{=} \frac{V}{1-V}$, we have

$$\begin{aligned}
\ln X &= \ln\lambda + \frac{1}{c}\left(\ln V - \ln(1-V)\right) \\
\implies \operatorname{Var}(\ln X) &= \frac{\operatorname{Var}\left(\ln V - \ln(1-V)\right)}{c^2} \\
&= \frac{\operatorname{Var}\ln V + \operatorname{Var}\ln(1-V) - 2\operatorname{Cov}\left(\ln V, \ln(1-V)\right)}{c^2} \\
&= \frac{\frac{\pi^2}{6} + \psi_1(k)}{c^2},
\end{aligned}$$

where we used the properties $\psi_1(1) = \frac{\pi^2}{6}$, $\operatorname{Var}\ln V = \psi_1(1) - \psi_1(1+k)$, $\operatorname{Var}\ln(1-V) = \psi_1(k) - \psi_1(1+k)$, and $\operatorname{Cov}\left(\ln V, \ln(1-V)\right) = -\psi_1(1+k)$ for the final equality. ■

Corollary E.2.2. *Denote a Burr XII random variable with strictly positive parameters (c,k,λ) by $X \sim BurrXII(c,k,\lambda)$, whose Cumulative Distribution Function (CDF) is given by $F(x) = 1 - \left[1 + \left(\frac{x}{\lambda}\right)^c\right]^{-k}$. Define $\alpha = ck$ and assume $\alpha > 1$. Denote its Complementary Cumulative Distribution Function (CCDF) by $\bar{F}(x) = 1 - F(x) = \mathbb{P}(X \geq x)$. Define the point x_p such that $\bar{F}(x_p) = p$. Then the* log-log *slope of its Complementary Cumulative Distribution Function (CCDF) at point x_p is given by*

$$\alpha\left(1 - p^{\frac{1}{k}}\right).$$

Proof. Denote the log-log slope of the CCDF by

$$\beta(x) := -\frac{d\log\bar{F}(x)}{d\log x}.$$

Since $\log\bar{F}(x) = -k\log\left(1 + \left(\frac{x}{\lambda}\right)^c\right)$, we have

$$\frac{d\log\bar{F}(x)}{d\log x} = -\alpha\,\frac{\left(\frac{x}{\lambda}\right)^c}{1 + \left(\frac{x}{\lambda}\right)^c},$$

so

$$\beta(x) = \alpha\frac{\left(\frac{x}{\lambda}\right)^c}{1 + \left(\frac{x}{\lambda}\right)^c}. \tag{11}$$

Observe that Eq. (11) has asymptotic slope α as $x \to \infty$. Because $\bar{F}(x_p) = p$, we have

$$\left(\frac{x_p}{\lambda}\right)^c = p^{-1/k} - 1.$$

Substituting into Eq. (11) at $x = x_p$:

$$\beta(x_p) = \alpha\left(1 - p^{\frac{1}{k}}\right).$$

Lastly, from Corollary E.2.1, this equation is also equivalent to

$$\beta(x_p) = k\sqrt{\frac{\frac{\pi^2}{6} + \psi_1(k)}{\nu}}\left(1 - p^{\frac{1}{k}}\right),$$

which shows that the log-log slope depends on the log-variance ν. ■