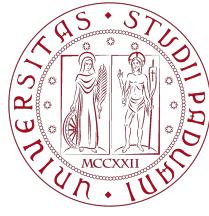


Final Report

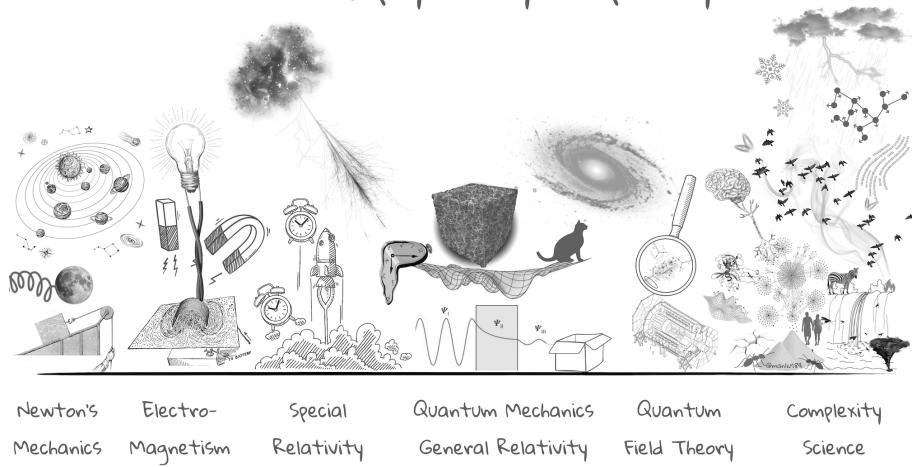
Physics of Complex Networks: Structure and Dynamics

Last update: July 13, 2025



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Areas of physics by complexity



Project # 37: Explosive percolation
Project # 45: European Railway network
Rizzi, Lorenzo

Contents

1	Explosive percolation	1
1.1	Percolation theory and phase transitions	1
1.2	Achlioptas processes on random graphs	1
1.3	Explosive percolation in SF networks	2
1.4	Supplementary material	4
1.4.1	Scaling law for Δm	4
1.4.2	Cluster size distribution	4
1.4.3	Again on SF networks	5
2	Building a railway graph using EGM Dataset	9
2.1	The EuroGlobalMap dataset	9
2.2	Building the graph	9
2.3	Supplementary material	12
3	Bibliography	15

1 | Explosive percolation

Task leader: Lorenzo Rizzi

1.1 | Percolation theory and phase transitions

Percolative processes were initially introduced on specific and regular networks (e.g. D -dimensional lattices) to model physical phenomena such as percolation of water in porous stones. However, nothing prevents us from applying the percolation's paradigm to arbitrary network topologies. Defining p as the probability that a randomly chosen edge (or node) is removed from the network, then a classical result from percolation theory is that, for certain network topologies, a phase transition (PT) occurs as p is varied. If we define S as the probability that a randomly chosen node belongs to the GCC (our *order parameter*), then, for $N \rightarrow \infty$ (*thermodynamic limit*), we have $S = 0$ for $p < p_c$ and $S = S(p) \neq 0$ for $p > p_c$.

It is a standard result [6] that classical random percolation on networks displays a *continuous* phase transitions, meaning that, at criticality, $S(p_c) = 0$ and the order parameter has no discontinuous behaviour around the critical point. In addition, there are various other indicators of an incoming phase transition widely used in statistical mechanics. We'll define n_s as the number of (finite) clusters of size s per node and χ as the average cluster size:

$$\chi = \frac{\sum_s n_s s^2}{\sum_s n_s s} \quad (1.1)$$

In second order PT, at criticality χ is expected to diverge with a well-defined critical exponent $\chi \sim |p - p_c|^{-\nu}$ and the cluster distribution n_s reduces to a power law $n_s \sim s^{-\tau}$.

However, in 2009, an inspiring article from Achlioptas et al. [1] proposed a new type of percolation process that, allegedly, leads to a discontinuous type of transition (called *explosive percolation* for its abrupt nature). However, posterior papers [4] have managed to prove that explosive percolation is actually a continuous one. In the present task, we are going to explore and simulate Achlioptas process(es) on various network topologies to recreate the explosive percolation behaviour.

1.2 | Achlioptas processes on random graphs

Classical percolation prescribes the removal (or addition) of randomly selected edges (or nodes). As mentioned in Par 1.1, this random procedure leads to a continuous phase transition. However, one can modify the rule according to which edges are removed (or added) to, hopefully, obtain new types of percolation processes.¹

¹More specifically, new universality classes or, even better, a completely different order of PT

We will proceed as follows. We start with N nodes and no edge connectivity. At each step of our simulation, we are going to add a new edge connecting two nodes in the graph (so $m(t) = t$). If those nodes are chosen at random, then we are basically performing classical (bond) percolation and building an ER network which is expected to generate a GCC when $\langle k \rangle = 1$, i.e. when the number of added edges m is $\frac{1}{2}N$. However, we can bias the choice of nodes to connect by imposing new *selection rules*. In [1], two of such rules are presented: *Product Rule* (PR) and *Sum Rule* (SR). Their mechanism is quite easy: at each iteration, select two pairs of candidate nodes $(u_1, v_1), (u_2, v_2)$. Let U_i, V_i be the size of the cluster to which u_i, v_i belongs. Then, PR (SR) rule prescribes to select and connect the pair of nodes that minimizes the quantity $U_i \cdot V_i$ ($U_i + V_i$), while discarding the other. The idea behind this selection rule is to postpone the formation of a GCC biasing the choice towards small clusters. PR and SR are examples of *unbounded rules*; at variance, *bounded rules* treat all cluster whose size is larger than K equally. When $K = 1$, we talk about the Bohman and Frieze (BF) rule: when possible, always select the edge that connects two isolated nodes.

As of now, we have 4 different protocols with which we can repeatedly add edges and grow a network (ER-like, PR, SR, BF). Simulations over 10 independent realizations when $N = 10^6$ are shown in Fig. 1.1a. As expected, ER growth (i.e. random rule selection) displays a continuous phase transition when $m/N = \frac{1}{2}\langle k \rangle = 0.5$. On the contrary, both PR and SR rules induce a particularly sharp transition called *explosive* because of its abrupt nature. In fact, it is so sharp that it may resemble a first order discontinuous PT. This behaviour is due to the fact that the competition mechanism (either SR or PR) tends to suppress the formation of large components, generating the necessary "powder keg" (i.e., abundant small-sized clusters, [2]) which then triggers the abrupt transition. In Supplementary Material 1.4.1, we propose the approach performed in [1] to investigate the nature of this PT. However, not all selection rule lead to an allegedly discontinuous PT: using BF procedure, a continuous transition is resumed. Along with the LCC, one can also monitor the behaviour of the average (finite) cluster size χ , defined as in Par.1.1 (Fig.1.1b). Simulations are run for a large but finite value of N , so we can't expect to see a divergence, but results are anyway convincing and show a marked peak in both ER and BF rules around their critical points. Surprisingly enough, PR and SR too display a similar divergent behaviour, which is not to be expected for first-degree PT. To characterize even further what is happening at criticality, we can study the cluster distribution n_s around m_c (discussed in greater details in Supplementary Material 1.4.2). Again, one can easily show that when $m \approx m_c$, all 4 distributions converge to a power law, suggesting again a continuous-like behaviour for PR and SR rules too.

As a matter of fact, explosive percolation was proven to be a continuous PT ([4]). The reason behind the steep ascent of S around criticality is because the critical exponent β is rather small: $S(p) \sim |p - p_c|^\beta$, $\beta \approx 0.0555$

1.3 | Explosive percolation in SF networks

We now turn our attention to a different network topology, i.e. scale-free networks whose degree distribution $P(k) \sim k^{-\gamma}$, using a configuration model. We proceed as illustrated in [7]. Let N be the initial number of initially disconnected nodes. Sampling

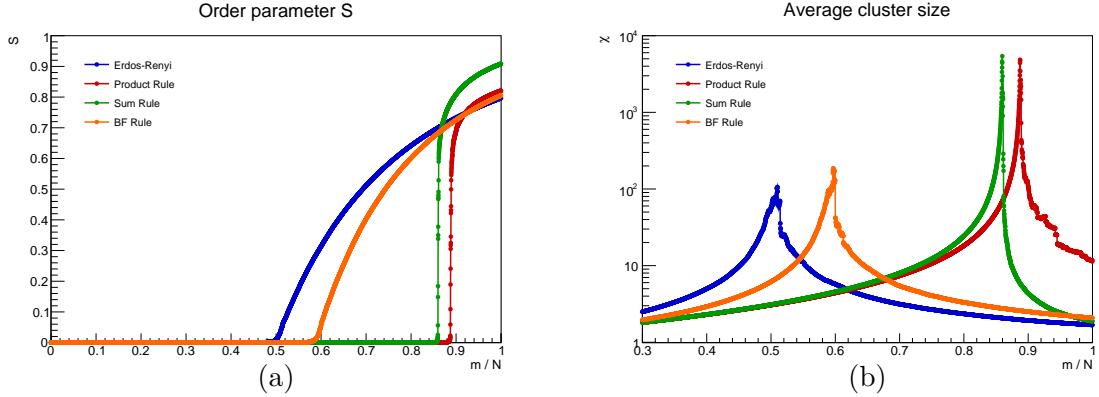


Figure 1.1: (a) LCC size vs. added edges m/N , $N = 10^6$. Both ER and BF protocol give rise to a continuous phase transition, while competitive rule like PR and SR generates a seemingly looking discontinuous PT (b) Average cluster size vs. m/N , $N = 10^6$.

from a power law, we associate to each node a fixed amount of stubs and we gradually proceed to connect them following either a PR rule or a random choice (of course, when connecting stubs at random we fall back on the classical percolation problem on scale-free networks). Results for S are shown in Fig 1.2a, 1.2b. When performing a random connection between stubs (classical percolation on SF networks), one should expect to see a critical behaviour only when $\gamma > 3$, since $\kappa \rightarrow \infty$ for $\gamma < 3$, regardless of k_{min} . In fact, when $\gamma < 3$, the order parameter smoothly increases and is 0 only when $m = 0$. When using PR rule, we observe again the abrupt change when $m \approx m_c(\gamma)$

The most interesting result is, however, that using PR for $\gamma = 2.6 < 3$ a phase transition is observed. In fact, one can show [7], [3] that, when using PR, the threshold below which the GCC is always observed (thus no PT) is $\gamma = \gamma_c \approx 2.4$. When $\gamma > \gamma_c$, a steep PT is measured. However, as already mentioned, looks can be deceiving: even if highly abrupt, those PTs are still of the second order.

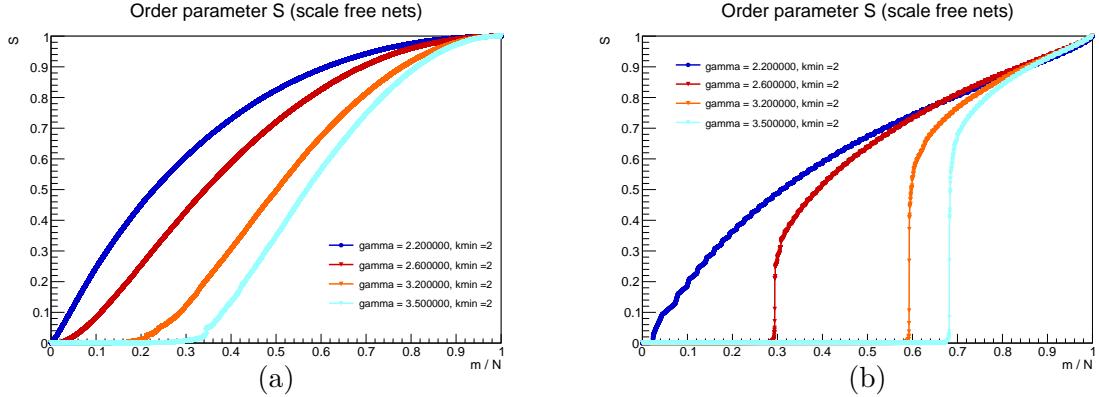


Figure 1.2: (a) LCC size vs. added edges m/N , $N = 10^6$, randomly connecting stubs until a power law is reached (classical percolation on SF networks) (b) LCC size vs. added edges m/N , $N = 10^6$, using PR rule to connect stubs. Averages on 10 independent realizations

1.4 | Supplementary material

1.4.1 Scaling law for Δm

In its seminal work [1], Achlioptas proposes an operational method to decide whether the observed PT is continuous or discontinuous. Consider again the procedure presented in Par. 1.2, where edges are gradually added $m = m(t) = t$ following different protocols (ER, PR, SR, BF) and the LCC size is monitored. We will define two edge value m_0 and m_1 at different times such that:

$$\begin{cases} m_0 = \min_{m < M} \{S > \sqrt{N}\} \\ m_1 = \min_{m < M} \{S > 0.5N\} \end{cases} \quad (1.2)$$

The difference Δ , called transition window, is defined as $\Delta = m_1 - m_0$ and provide a quantitative way to define how abrupt the PT is².

As claimed in [1], Δ should scale linearly for continuous PTs, $\Delta \sim N$. So by observing the scaling behaviour of Δ in simulations, one can characterize a bit further the PT. Results are shown in Fig. 1.3a. For both ER and BF rule, $\frac{\Delta}{N}$ seems indeed to converge to a finite value when N is large enough and the results we obtained are compatible with Achlioptas' findings. At variance, when using PR or SR rule, the curves tend to collapse at 0, suggesting a sub-linear behaviour. And in fact, plotting $\frac{\Delta}{N^\alpha}$ with $\alpha = \frac{2}{3}$ (Fig. 1.3b), results from PR and SR now stabilize to a small but different from 0 value (again, in agreement with [1]). This means that explosive percolation is characterized by a sub-linear scaling law of the critical window:

$$\Delta \sim N^{\frac{2}{3}}$$

If m_0 is interpreted as an estimate of the critical point and m_1 a point where the GCC has already emerged, than one can say that the distance between the percolating phase and the critical point gets smaller and smaller when $N \rightarrow \infty$. This implies that, at criticality, a constant fraction of the vertices is accumulated into a single giant cluster within a sub-linear number of steps [2]

However, in 2009, Ziff [8] showed that, for the regular percolation on a 2D lattice, $\Delta \sim N^\alpha$ with $\alpha \neq 1$, so this is not conclusive evidence that explosive percolation is a discontinuous PT. It is today accepted that EP under Achlioptas processes are actually continuous PTs, but with a universality class different from all the others percolative processes.

1.4.2 Cluster size distribution

Let us investigate further the behaviour of the cluster size distribution of continuous PTs. Defining n_s as the number of clusters of size s per node, then out of criticality n_s has exponentially distributed tails. When approaching the critical point, the distribution becomes a power law

$$n_s \sim s^{-\tau} \text{ for large } s$$

²The choice of the endpoints is, up to a certain extent, arbitrary. The value of m_0 is chosen because, usually, at criticality the LCC scale logarithmically with N , and thus provides an estimate of the critical point

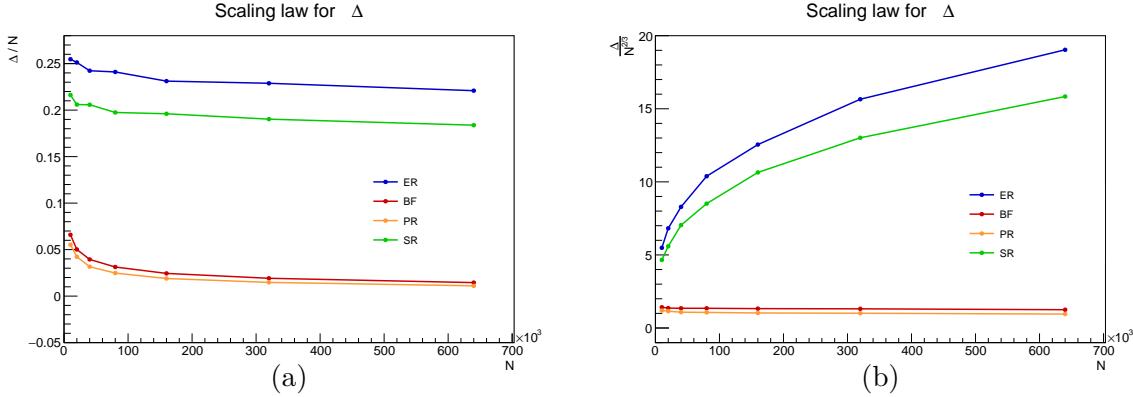


Figure 1.3: (a) LCC size vs. added edges m/N , $N = 10^6$, randomly connecting stubs until a power law is reached (classical percolation on SF networks) (b) LCC size vs. added edges m/N , $N = 10^6$, using PR rule to connect stubs. Averages on 10 independent realizations

where τ is usually called the Fisher exponent. This power law is another strong indicator of the onset of a second order PT.

Thus, we can collect the cluster distribution n_s at different value of m/N (centered around the critical point). Results carried out for each percolative process as those presented in Par. 1.2 (Erdos-Renyi, Product Rule, Sum Rule, BF rule) are shown in Fig. 1.4. And indeed, when $m \sim m_c$, the distribution gets closer and closer to a power law (a straight line in a log-log plot) in all cases, providing more evidence that explosive percolation is a continuous PT.

1.4.3 Again on SF networks

In Par. 1.3, explosive percolation was presented on scale-free networks. In this appendix, we are going to show more processed data and explain the theoretical framework behind SF networks.

Using our configuration model, we will impose a discrete degree distribution $P(k)$:

$$P(k) = \frac{1}{Z} k^{-\gamma} \quad (1.3)$$

To normalize it, we need to set at least two boundaries, k_{min}, k_{max} . We choose $k_{max} = k_{min} N^{\frac{1}{\gamma-1}}$, i.e. the natural cutoff and k_{min} is a free parameter. As $N \rightarrow \infty$, we can assume $k_{max} = \infty$ for computations purposes.

For uncorrelated networks, the presence of a GCC is conditioned to the Molloy-Reed coefficient κ :

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

Using Eq. 1.3, we can compute the statistical moments. Setting $k_{min} = 1$:

$$\begin{aligned} \langle k^2 \rangle &= \frac{1}{Z} \sum_{k_{min}}^{\infty} k^{2-\gamma} = \frac{\zeta(\gamma-2)}{\zeta(\gamma)} \\ \langle k \rangle &= \frac{1}{Z} \sum_{k_{min}}^{\infty} k^{1-\gamma} = \frac{\zeta(\gamma-1)}{\zeta(\gamma)} \end{aligned} \quad (1.4)$$

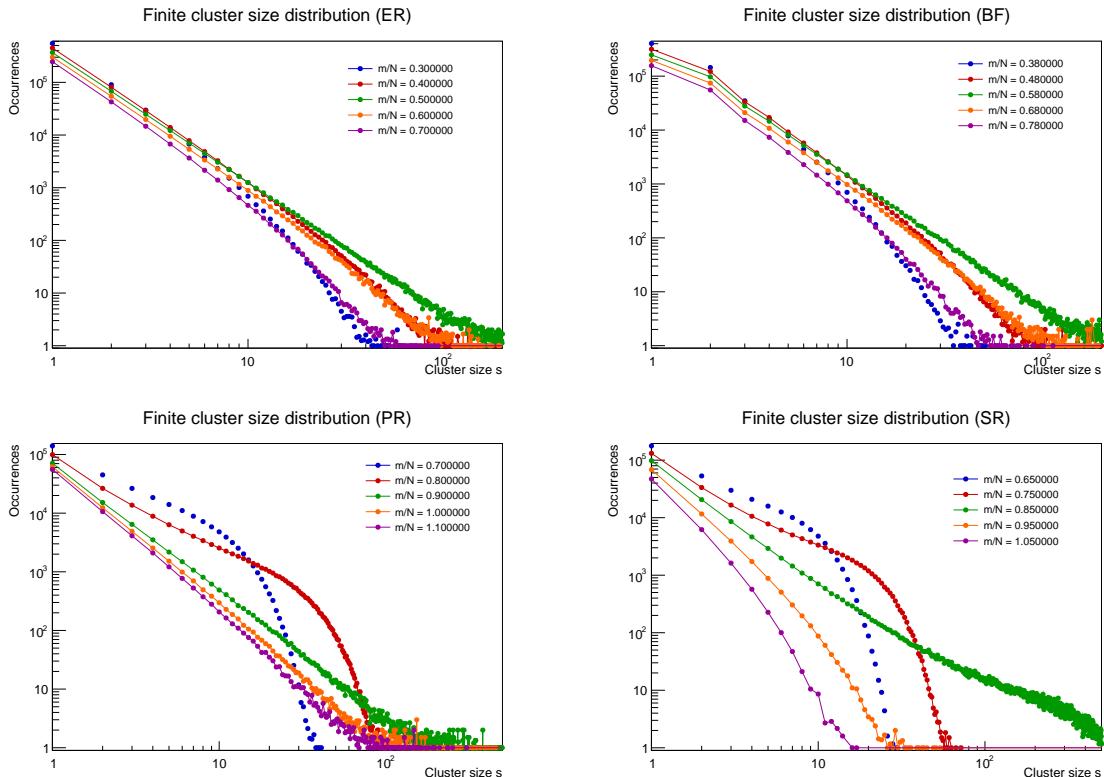


Figure 1.4: *Finite cluster size distribution at and around criticality.* As expected, when $m \approx m_c$, the distribution appears approximately as a power law (the critical value can be estimated inspecting Fig. 1.1a). Simulation run on networks with $N = 10^6$, over 10 independent realizations

whereas, for $k_{min} = 2$,

$$\begin{aligned}\langle k^2 \rangle &= \frac{1}{Z} \sum_{k_{min}}^{\infty} k^{2-\gamma} = \frac{\zeta(\gamma - 2) - 1}{\zeta(\gamma) - 1} \\ \langle k \rangle &= \frac{1}{Z} \sum_{k_{min}}^{\infty} k^{1-\gamma} = \frac{\zeta(\gamma - 1) - 1}{\zeta(\gamma) - 1}\end{aligned}\quad (1.5)$$

So that, finally,

$$\kappa_1 = \frac{\zeta(\gamma - 2)}{\zeta(\gamma)} \quad \kappa_2 = \frac{\zeta(\gamma - 2) - 1}{\zeta(\gamma) - 1} \quad (1.6)$$

The critical point for uncorrelated networks can be computed as:

$$p_c = \frac{1}{\kappa - 1} \quad (1.7)$$

If $\gamma < 3$, then $\zeta(\gamma - 2)$ diverges to infinity and $p_c = 0$, so technically there is no critical behaviour and a GCC always exists, even with low connectivity. When $\gamma > 3$, a GCC finally appears together with a PT at a fixed p_c . However, for the GCC to be there even when $p_c = 1$ (full network recovered), then $\kappa > 2$. One can easily show that, when selecting $k_{min} = 1$:

- $\gamma < 3$: $\kappa = \infty$ and $p_c = 0$. The GCC is always present and is 0 only when $p_c = 0$.
- $3 < \gamma < \gamma_c \approx 3.5$: $2 < \kappa < \infty$. The system displays two phases (percolating and non percolating) separated by a critical point
- $\gamma > \gamma_c$: $\kappa < 2$. Even at full connectivity, there's no GCC so it makes no sense to talk about a PT here

Running a classical percolation on a SF networks, we got the results presented in Fig. 1.5a. They are compatible with our theoretical understanding of the process. The dotted line represents the analytically computed value of $p_c = m_c/M$ (M being the total number of generated stubs divided by 2). Its value is not too accurate and this is probably due to the fact that k_{max} was not set to infinity but to a large finite value. Indeed, we can compute directly from the generated distribution of the stubs the value of κ and use this value in Eq. 1.7 rather than computing it analytically. If we do so, we obtain better and more realistic result

When $k_{min} = 2$, then $\kappa > 2$, $\forall \gamma > 3$. Plots regarding the behaviour of S for regular percolation when k_{min} are shown in Fig. 1.5b. Now, we see the formation of a GCC and a PT even when $\gamma = 3.8$ (which was not the case when $k_{min} = 1$). This justifies our choice of selecting $k_{min} = 2$ in Par. 1.3

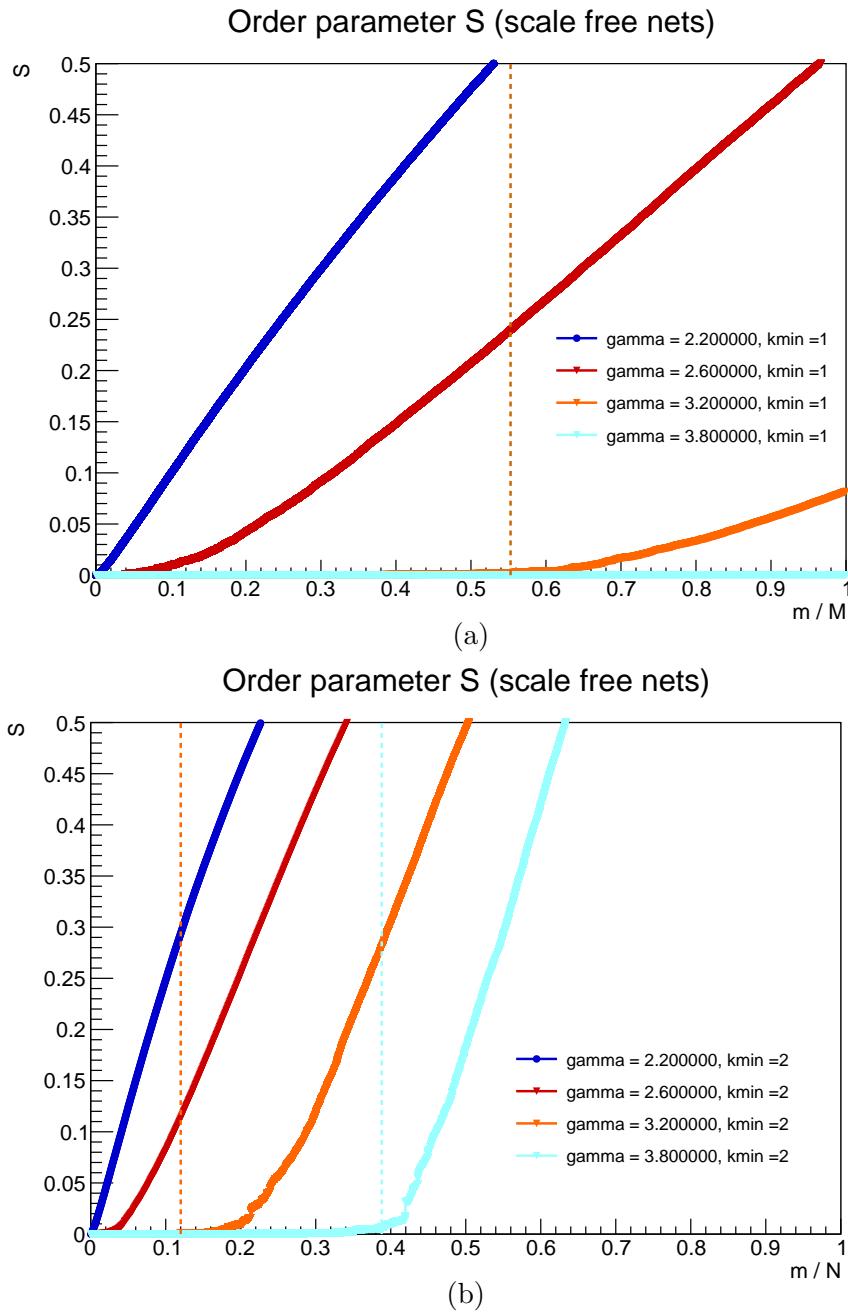


Figure 1.5: (a) Classical random percolation on SF networks when $k_{min} = 1$ and for various value of γ . Only when $\gamma = 3.2$ do we see a PT. The dotted line represent the critical point as computed in Eq. 1.7. (b) Classical random percolation on SF networks, $k_{min} = 2$. Now $\gamma = 3.8$ too displays a PT and a GCC. Dotted lines represent the critical point as computed in Eq. 1.7

2 | Building a railway graph using EGM Dataset

Task leader: *Lorenzo Rizzi*

2.1 | The EuroGlobalMap dataset

EuroGlobalMap [5] is a pan-European topographic database at global level of detail that provides extensive and precise territorial information including features such as waterways, road networks, and rail infrastructure. EGM utilizes Geographic Information System (GIS) data files (shapefiles, ...) to encode and represent spatial information for each type of *feature* (a generic entity on the surface on Earth, in GIS jargon).

In the present work, we are going to extract data on the distribution of European railways from the EGM dataset with the purpose of building a graph representation of rail connections at large scale at a national level and across the whole Europe.

2.2 | Building the graph

EGM offers a great variety of spatial information, ranging from highways to waterways. However, we are only interested in the railway infrastructure, thus we need to select the proper *feature* from the database. In particular, using R package *simple feature*, meant to digest shapefiles, we can easily retrieve two dataframes for each country. The first one contains spatial information (longitude, latitude) on all rail stations in the selected country. Together with longitude and latitude, a list of attributes associated with each station (high-speed, underground, national relevance) is also provided. Stations are considered 0-dimensional, i.e. as points p in the map. The second dataframe contains spatial information on railways lines. Each railway line is encoded as a *linestring* object, which is essentially a pair of points p_1, p_2 representing the endpoints of the straight railroad.

Stations analysis Data analysis of station positions is quite straightforward. Using R, it is easy to read the initial dataframes containing geometric points and to build new dataframes with the following columns: nodeID, nodeLabel, latitude, longitude, country_name, country_ISO3.

We run the analysis on the following countries:

AL, AT, BE, BG, CHLI, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GE, GR, HR, HU, IE, IS

However, data on railways wasn't found in the provided folder for three countries (Albania, Iceland, Cyprus) so they will be excluded

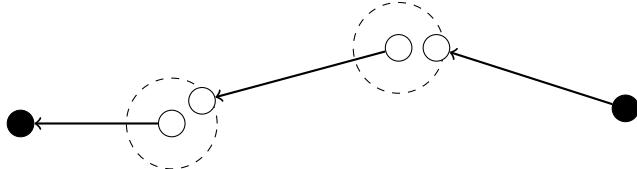


Figure 2.1: An illustration of how the proposed algorithm works. Black dots represent line endpoints that are sufficiently close to a station. Starting from a downstream node (on the left), the algorithm searches for a line in the database whose destination lies within a fixed distance (minRails) from the origin of the current downstream segment. This procedure is iterated until a node with a matched origin (i.e., associated with a known station) is found.

Railway lines analysis We already have the nodes of our graph, and we now need to connect these nodes. The second dataframe is composed of *linestring*, i.e. pairs of spatial coordinates. Unfortunately, however, EGM does not provide direct connectivity information between stations, but only the coordinates of the endpoints of each railroad segment. Hence, we extract the endpoints of each lines from the EGM database and, by comparing distances using a spatial threshold, associate these points with the nearest railway station. If no stations are found within the spatial threshold, a placeholder is used instead (NA in R). By executing this procedure and mapping endpoints to known stations using a fixed threshold (minStation), a significant number of lines in the EGM database remain unmatched. This occurs because a single row in the EGM database (i.e., a line) typically represents only a segment of a railroad connecting two stations, and therefore some lines may have endpoints located in rural or intermediate areas. The algorithm that was used to retrieve the full edge between nodes is the following:

- Fix a threshold minStation . For each line in the EGM database, extract its endpoints and compare them with the previously computed stations. If the distance is less than minStation , associate the endpoint with the corresponding station's `nodeID`; otherwise, assign NA.
- At the end of this procedure, some lines will have both origin and destination matched. These lines are moved to a definitive dataframe, and their two endpoints are joined as an edge in the graph.
- Identify *downstream* segments, i.e., lines whose destination is close to a station while the origin is set to NA (i.e., too far from any station). Then, iterate over all non-definitive lines and search for the optimal *upstream* link, defined as the nearest line (within a distance less than minRails) to the origin of the downstream segment. Repeat this process until a complete chain is formed, i.e., the downstream node is successfully connected to an upstream node.

An illustration of the algorithm is shown in Fig.2.1. Once the edges connecting stations are found, one can write the information in a `.txt` file and plot the results

Results Our algorithm relies on two tunable parameters: minStation and minRails , typically ranging between 1 and 30 km depending on the size of the country. Some results are shown in Fig.2.2. Overall, the outcomes are quite satisfactory when compared to actual railway maps. However, the algorithm presents some limitations. In

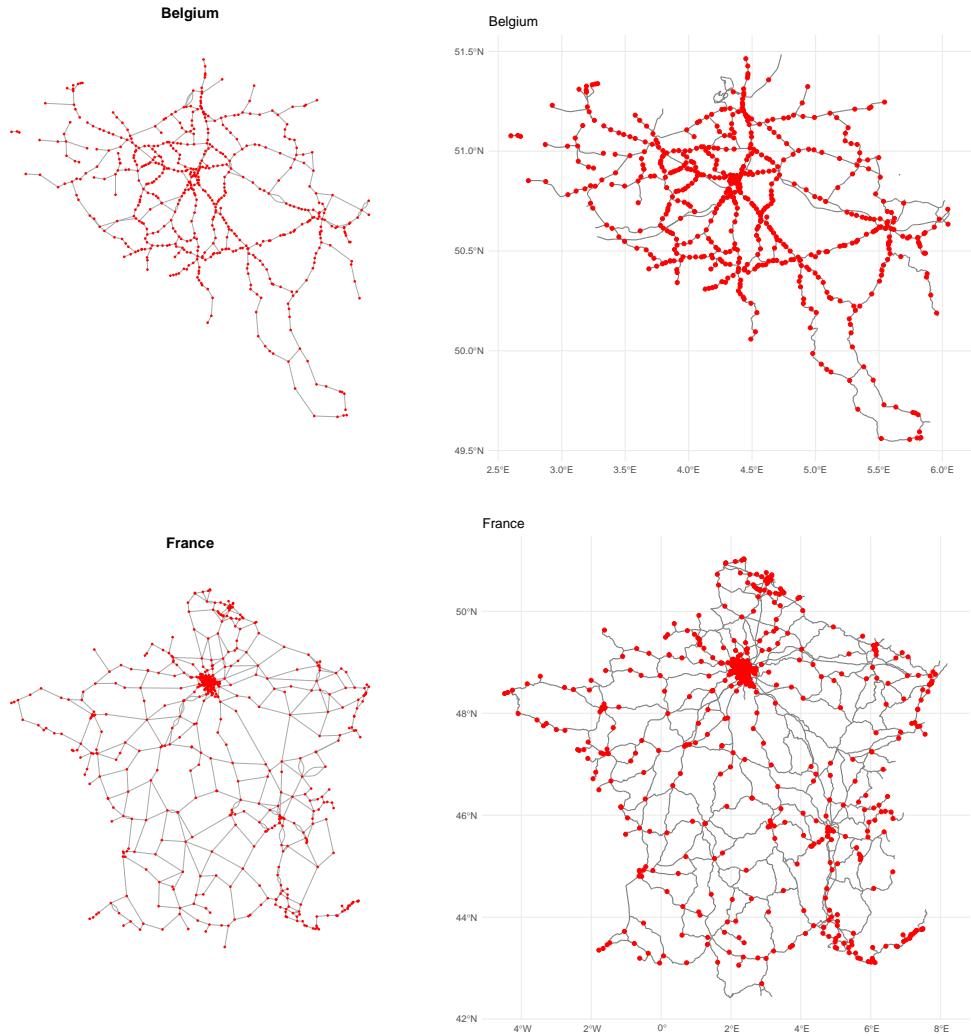


Figure 2.2: On the left, resulting graphs of Belgium’s or France’s railroad network with stations as nodes ($\text{minStation} = 20\text{km}$, $\text{minRails} = 20\text{km}$ for Belgium, $\text{minStation} = 40\text{km}$, $\text{minRails} = 40\text{km}$ for France). On the right, the corresponding spatial rendering from EGM GIS data

areas where railway lines are densely tangled (as in France), it may generate multiple redundant links or, conversely, remove valid links, resulting in disconnected segments. Another issue arises from line intersections: it may occur that railway segments intersect at locations where no station is present. In such cases, the algorithm may fail to properly follow the upstream chain, leading to station disconnections. This happens because the climbing rule illustrated in Fig. 2.1 selects only the nearest upstream link, potentially discarding other plausible connections. A more robust version of the algorithm should instead consider all upstream candidates within a fixed spatial threshold.

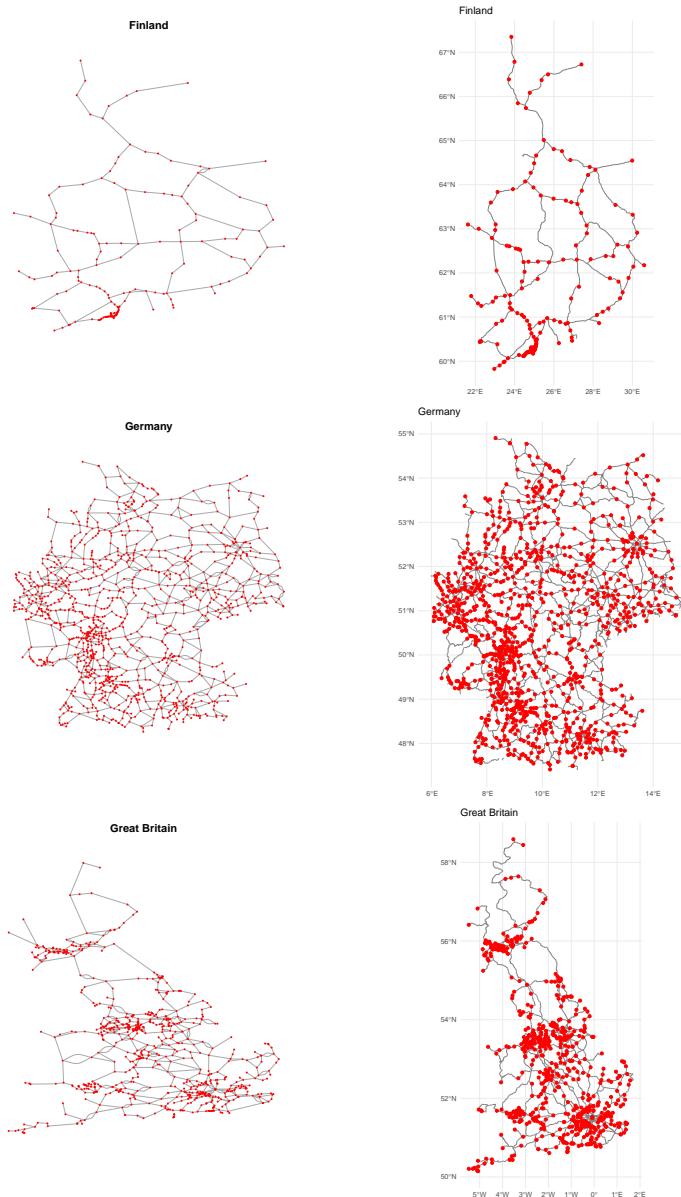


Figure 2.3: *Graphs obtained and GIS drawing for Finland, Germany, Great Britain*

2.3 | Supplementary material

We report here additional figures similar to Fig.2.2 for other european countries (Fig. 2.3). A visualization of the full European railway graph is shown in Fig.2.4.

Given the constructed graph, we compute the degree distribution for each country, as well as for the entire European network (see Fig.2.5 and Fig.2.6). These distributions appear fairly homogeneous, with a well-defined average degree. At the European level, we find:

$$\langle k \rangle \approx 2.65(6)$$

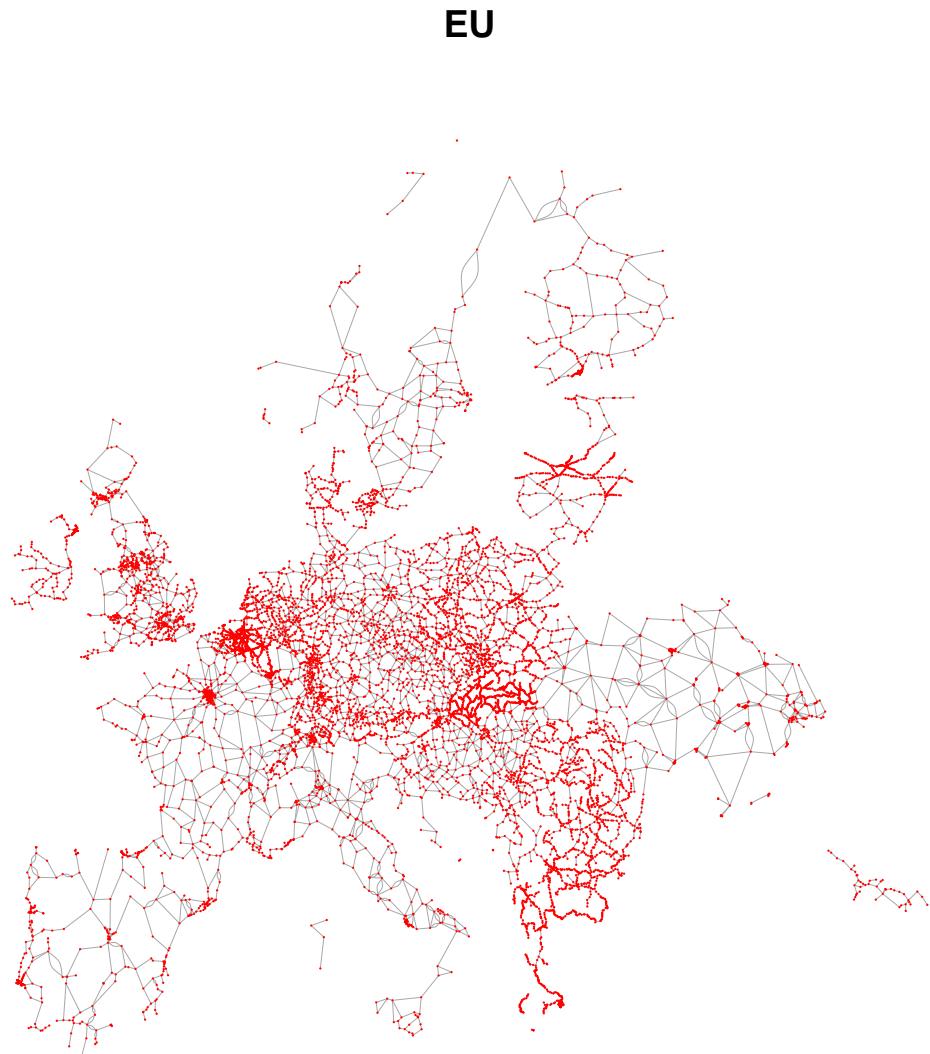


Figure 2.4: *Obtained graph representing network connectivity in Europe.*

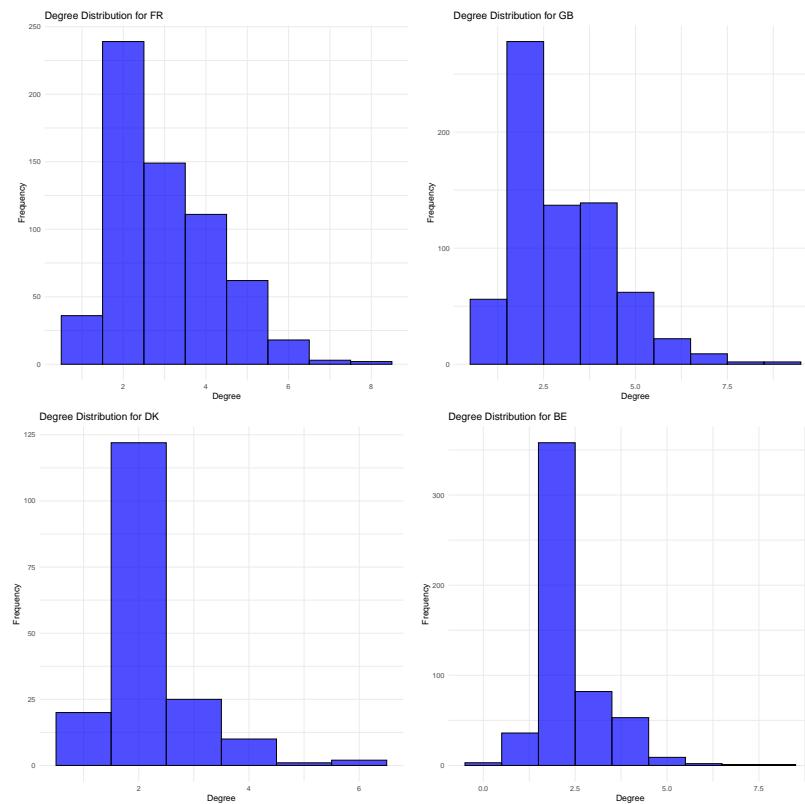


Figure 2.5: *Degree distributions for France, Great Britain, Denmark and Belgium*

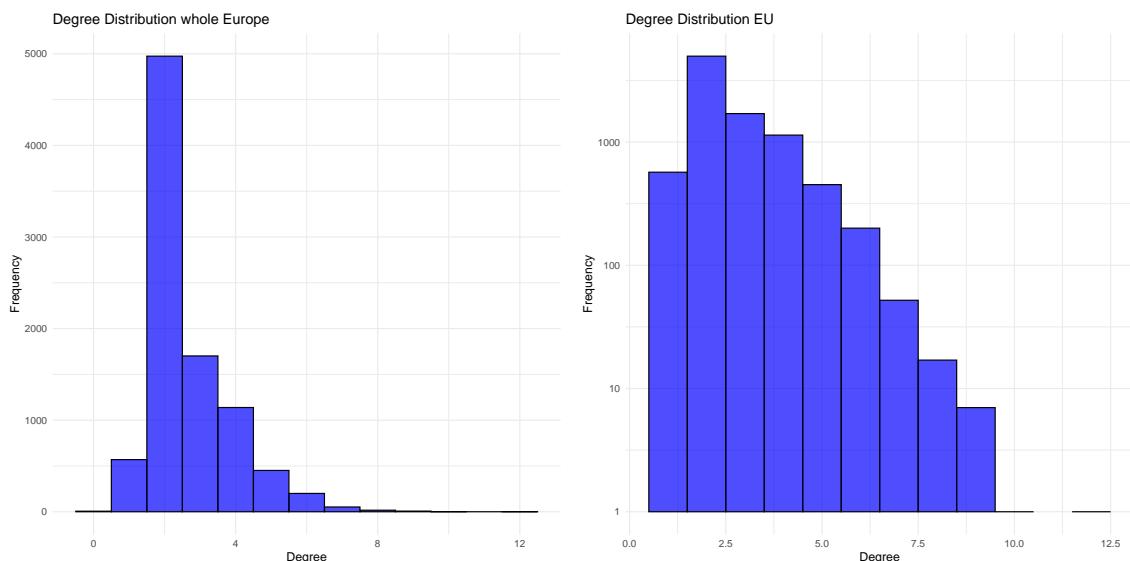


Figure 2.6: *Degree distribution across Europe. In the right panel, the y-axis is on a logarithmic scale. A visual inspection reveals that the tails decay exponentially rather than following a fat-tailed distribution, suggesting a relatively homogeneous degree distribution*

3 | Bibliography

- [1] Dimitris Achlioptas, Raissa M. D’Souza, and Joel Spencer. Explosive percolation in random networks. *Science*, 323(5920):1453–1455, 2009. doi: 10.1126/science.1167782. URL <https://www.science.org/doi/abs/10.1126/science.1167782>.
- [2] S. Boccaletti, J.A. Almendral, S. Guan, I. Leyva, Z. Liu, I. Sendiña-Nadal, Z. Wang, and Y. Zou. Explosive transitions in complex networks’ structure and dynamics: Percolation and synchronization. *Physics Reports*, 660:1–94, 2016. ISSN 0370-1573. doi: <https://doi.org/10.1016/j.physrep.2016.10.004>. URL <https://www.sciencedirect.com/science/article/pii/S0370157316303180>. Explosive transitions in complex networks’ structure and dynamics: Percolation and synchronization.
- [3] Y. S. Cho, J. S. Kim, J. Park, B. Kahng, and D. Kim. Percolation transitions in scale-free networks under the achlioptas process. *Physical Review Letters*, 103 (13), September 2009. ISSN 1079-7114. doi: 10.1103/physrevlett.103.135702. URL <http://dx.doi.org/10.1103/PhysRevLett.103.135702>.
- [4] R. A. da Costa, S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Explosive percolation transition is actually continuous. *Phys. Rev. Lett.*, 105:255701, Dec 2010. doi: 10.1103/PhysRevLett.105.255701. URL <https://link.aps.org/doi/10.1103/PhysRevLett.105.255701>.
- [5] EuroGeographics. Euroglobalmap. <https://www.mapsforeurope.org/datasets/euro-global-map>, 2021. Accessed: 2025-07-12.
- [6] Ming Li, Run-Ran Liu, Linyuan Lü, Mao-Bin Hu, Shuqi Xu, and Yi-Cheng Zhang. Percolation on complex networks: Theory and application. *Physics Reports*, 907: 1–68, April 2021. ISSN 0370-1573. doi: 10.1016/j.physrep.2020.12.003. URL <http://dx.doi.org/10.1016/j.physrep.2020.12.003>.
- [7] Filippo Radicchi and Santo Fortunato. Explosive percolation: A numerical analysis. *Phys. Rev. E*, 81:036110, Mar 2010. doi: 10.1103/PhysRevE.81.036110. URL <https://link.aps.org/doi/10.1103/PhysRevE.81.036110>.
- [8] Robert M. Ziff. Explosive growth in biased dynamic percolation on two-dimensional regular lattice networks. *Phys. Rev. Lett.*, 103:045701, Jul 2009. doi: 10.1103/PhysRevLett.103.045701. URL <https://link.aps.org/doi/10.1103/PhysRevLett.103.045701>.