EÖTVÖS LORÁND UNIVERSITY

FACULTY OF SCIENCE

STATISTICAL PHYSICS, BIOLOGICAL PHYSICS AND PHYSICS OF QUANTUM SYSTEMS PROGRAMME
AT THE DOCTORAL SCHOOL OF PHYSICS

Hyperbolic geometry of complex networks: models of network growth and embeddings of real networks

DOCTORAL (PHD) DISSERTATION
DOI: 10.15476/ELTE.2023.008

Author:
Bianka Kovács

Supervisor:
Gergely Palla, DSc

HEAD OF THE DOCTORAL PROGRAMME:
Gábor HORVÁTH, DSc

HEAD OF THE DOCTORAL SCHOOL:
Jenő GUBICZA, DSc

2023

Acknowledgements

At this point, I would like to thank each of the people without whom the writing of this dissertation would not have been possible at all. First of all, I would like to thank Dr. Gergely Palla for supervising my work over the past years. Besides, I thank Sámuel Gáspár Balogh for the joint work.

I am also very grateful to Dániel Molnár for the "only five minutes" he devoted to provide me company during each and every phase of the work, to Robert Bolla for always trying to sustain my life functions, to Norman for always being a dog, and thus, my best coworker at home office, and to the rest of my family.

Contents

List of Figures

List of Abbreviations

AMI	**a**djusted **m**utual **i**nformation
AUPR	**a**rea **u**nder **p**recision-**r**ecall curve
AUROC	**a**rea **u**nder **r**eceiver **o**perating **c**haracteristic curve
BA	**B**arabási–**A**lbert
CCDF	**c**omplementary **c**umulative **d**istribution **f**unction
COM	**c**enter **o**f **m**ass
***d*PSO**	d-dimensional PSO
E-PSO	PSO with a time-dependent number of **e**xternal links emerging per time step
ER	**E**rdős–**R**ényi
Euc	**Euc**lidean
fPSO	PSO with a tunable multiplying factor f in the initial radial node coordinates
GR	**g**reedy **r**outing
HOPE	**H**igh-**O**rder **P**roximity preserved **E**mbedding
HOPE-R	HOPE with considering the first dimension of the embedding to be **r**edundant
HOPE-S	HOPE with **s**hifting the mean of the proximity matrix to 0
hydra	**hy**perbolic **d**istance **r**ecovery and **a**pproximation
hyp	**hyp**erbolic
ISO	**Iso**map
LE	**L**aplacian **E**igenmaps
MIC	**m**odel-**i**ndependent **c**onversion
ncMCE	**n**oncentered **m**inimum **c**urvilinear **e**mbedding
nPSO	**n**onuniform PSO
PDL	**p**roportion of **d**eleted **l**inks
PR	**p**recision-**r**ecall
PSO	**p**opularity-**s**imilarity **o**ptimization
ROC	**r**eceiver **o**perating **c**haracteristic
SBM	**s**tochastic **b**lock **m**odel
SPL	**s**hortest **p**ath **l**ength
SVD	**s**ingular **v**alue **d**ecomposition
TREXPEN	**t**ransformation of **ex**ponential shortest **p**ath lengths to **E**uclidea**n** measures
TREXPEN-R	TREXPEN with considering the first dimension of the embedding to be **r**edundant
TREXPEN-S	TREXPEN with **s**hifting the mean of the proximity matrix to 0
TREXPIC	**t**ransformation of **ex**ponential shortest **p**ath lengths to hyperbol**ic** measures
WCC	**w**eakly **c**onnected **c**omponent

Introduction

In the thriving field of network theory, the topic of hidden geometric structures underlying complex networks gains more and more attention. The corresponding main hypothesis states that characterizing the network nodes by spatial coordinates in a hidden metric space enables the expression of the network topology, i.e. the patterns of node-node interactions in terms of some geometric measures. Though the most trivial choice for hosting geometric structures is the "flat" Euclidean space, recently several indications have been revealed showing that complex networks may rather fit in curved spaces. While both positively curved (i.e., spherical) and negatively curved (i.e., hyperbolic) spaces are well known in mathematics, the topology of real-world networks is mainly assumed to be consistent with the rules of the latter. Presuming a connection between the network structure and hyperbolic geometry naturally opens the way for generating realistic synthetic graphs in hyperbolic spaces or placing observed networks (i.e. embedding them) in hyperbolic spaces in accordance with their connection structure. This dissertation deals with hyperbolic models of network growth and hyperbolic embeddings of networks that both associate small distances between the spatial position of the nodes in a hyperbolic space with small distances measured along the graph or large connection probabilities.

The main finding that led to the assumption of a relationship between complex networks and the hyperbolic space is that a tree-like structure representing some sort of hierarchical organization typically can be related to real-world graphs, while hyperbolic spaces can be interpreted as "continuous" trees. This analogy is explained in more detail in Chapter 1, where all the information used throughout this work regarding hyperbolic geometry is given, including different possibilities for displaying hyperbolic spaces in the Euclidean space and the corresponding formulas of the hyperbolic distance.

One application of hyperbolic geometry with respect to complex networks is the simulation of the formation of real-like network structures using distance-dependent connection probabilities for nodes distributed in a hyperbolic space. Chapter 2 sheds light on the capability of hyperbolic network models for resembling many typical network features through the well-known popularity-similarity optimization (PSO) model of network growth and its different variations, like the extensions I introduced in Refs. [T1] and [T3]. The realistic nature of these artificial networks manifests itself in the natural emergence of the small-world property, a scale-free degree distribution, a strong clustering and, as I showed in Ref. [T2], even a strong emergent community structure, which all are commonly observed properties of real networks.

The second branch of network geometry discussed in this dissertation is the area of hyperbolic node embeddings, where the aim is to create such an arrangement of the nodes of a real network in the hyperbolic space that reflects the observed network topology, namely where nodes that are topologically close to each other are hyperbolically close too and most of the links connect hyperbolically not too distant nodes. The low-dimensional vector representation of the nodes given by the position vectors can be utilized in the prediction of possibly missing links (following the principle that the connection probability is a decreasing function of the hyperbolic distance), can reveal a network's mesoscopic structure constituted by the communities of the nodes (by gathering together the nodes that have similar connection preferences), or can enable efficient navigation throughout the whole network relying on local neighborhood relations.

Related to the node embeddings, first, Chapter 3 considers the broadly-studied problem of embedding undirected networks, i.e. graphs having only symmetric connections. On the one hand, Sects. 3.1–3.2 describe popular dimension reduction techniques (Laplacian Eigenmaps

and Isomap) creating embeddings that minimize pairwise Euclidean distances of the topologically close nodes and, supplemented by some of my additions presented in Refs. [T1] and [T3], show how such Euclidean embeddings can be transformed into hyperbolic ones following the PSO model. On the other hand, Sect. 3.3 explains how the method named hyperbolic distance recovery and approximation (hydra) avoids the Euclidean-hyperbolic conversion step by performing dimension reduction directly in the hyperbolic space.

Then, to close, Chapter 4 tackles the novel problem of the hyperbolic embedding of directed networks (in which the relationships between the nodes are not symmetric), detailing my embedding methods from Ref. [T4]. These were basically inspired by the originally Euclidean Isomap (Sect. 3.1.2) and the inherently hyperbolic hydra (Sect. 3.3) methods of undirected networks, and another dimension reduction technique given by the High-Order Proximity preserved Embedding (HOPE, see Sect. 4.1.1) that places the nodes of directed networks into Euclidean spaces, representing small topological distances between the nodes with large inner products between their position vectors. While Sect. 4.1 provides solutions for creating hyperbolic embeddings of Euclidean origin – importantly, replacing the PSO-based Euclidean-hyperbolic conversion with a new, model-independent one –, Sect. 4.2 defines an algorithm for embedding directed networks directly in the hyperbolic space. Finally, Sect. 4.3 presents some applications of these new methods, demonstrating their competence for interpreting both directed and undirected, or even weighted links in networks.

1 Hyperbolic geometry and its connection to networks

Most complex networks exhibit some kind of hierarchical organization – if nothing else, at least a containment hierarchy of the classes of network nodes formed based on some node attributes, describing that general classes (lying on the top of the hierarchy) can be split into more specialized categories, which include even more tight classes of node traits [1, 2]. Such hierarchical relationships can be described by a tree-like structure, which is closely related to hyperbolic geometry. Namely, a tree with branching factor b can be interpreted as a discretized version of a two-dimensional hyperbolic space of curvature $K = -(\ln b)^2$ [2].

The negatively curved hyperbolic plane is usually visualized on the Euclidean plane (the curvature of which is 0) with the help of the native disk [2] that represent the hyperbolic plane as a Euclidean disk of infinite radius – see Sect. 1.1 for the detailed definition. As it is suggested by Fig. 1.1, the nodes of a tree (corresponding to node attribute categories in the above example) can be mapped to domains of the native disk or, in other words, the hyperbolic plane can be considered as a continuous tree that consists of an infinitely large number of nodes and is rooted at the origin of the native disk. Assuming that the analog of the level of the tree lying at r hops from the root is the origin-centered annulus on the native disk with inner radius r and outer radius $r + \Delta r$, and one node of the tree at this level corresponds to a sector of this annulus given by an angular range $[\theta, \theta + \Delta\theta]$, the equivalence of a hyperbolic plane and a tree becomes clear from a metric perspective. First, the area[1] of a disk centered in the origin of the native disk representing the hyperbolic plane of curvature $K = -(\ln b)^2$ increases with the radius r as $e^{\ln(b)\cdot r}$, just like its equivalent in the b-ary tree, i.e. the number of nodes[2] located at not more than r hops from the root. Second, the perimeter[3] of an origin-centered disk grows with its radius in the same way as the number of nodes[4] belonging to one level of the tree with the increase of the distance from the root, namely also as $e^{\ln(b)\cdot r}$. And finally, measuring the hyperbolic distance between two positions on the native disk reflects the same characteristics as walking along the tree: as moving sideways on the tree requires ascending on one branch to the corresponding branching point and then descending on the other branch, the geodesics in the hyperbolic geometry (along which the hyperbolic distances are measured) are bent towards the origin of the native disk.

All things considered, since complex networks may possess hidden tree-like structures and trees embed into hyperbolic spaces, hyperbolic spaces seem to naturally host complex networks. In the diverse studies of network geometry relying on this finding, several other representations of the hyperbolic space are also often applied besides the native disk [2–7], or in higher-dimensional cases, the native ball. The next sections review three commonly used models of hyperbolic geometry: Sect. 1.1 defines the native ball, Sect. 1.2 presents the Poincaré ball [8–10] and Sect. 1.3 deals with the hyperboloid model [9–12].

---

[1] In native representation of the two-dimensional hyperbolic plane of curvature $K = -\zeta^2$ (where $\zeta = \ln(b)$ in the current case), the area of the disk of radius r centered in the origin is $2\pi(\cosh(\zeta r) - 1)$, where $\cosh(\zeta r) = = [\exp(\zeta r) + \exp(-\zeta r)]/2 \sim \exp(\zeta \cdot r)$ for large values of r.

[2] The number of nodes in a tree of branching factor b at not more than r hops from the root can be calculated as $[(b+1)b^r - 2]/(b-1) \sim b^r = \exp(\ln(b^r)) = \exp(\ln(b) \cdot r)$ for large r values.

[3] In native representation of the two-dimensional hyperbolic plane of curvature $K = -\zeta^2$ (where $\zeta = \ln(b)$ in the current case), the perimeter of the disk of radius r centered in the origin is $2\pi \sinh(\zeta r)$, where $\sinh(\zeta r) = = [\exp(\zeta r) - \exp(-\zeta r)]/2 \sim \exp(\zeta \cdot r)$ for large values of r.

[4] The number of nodes in a tree of branching factor b at r hops from the root is $(b+1)b^{r-1} \sim b^r = \exp(\ln(b^r)) = = \exp(\ln(b) \cdot r)$ for large r values.

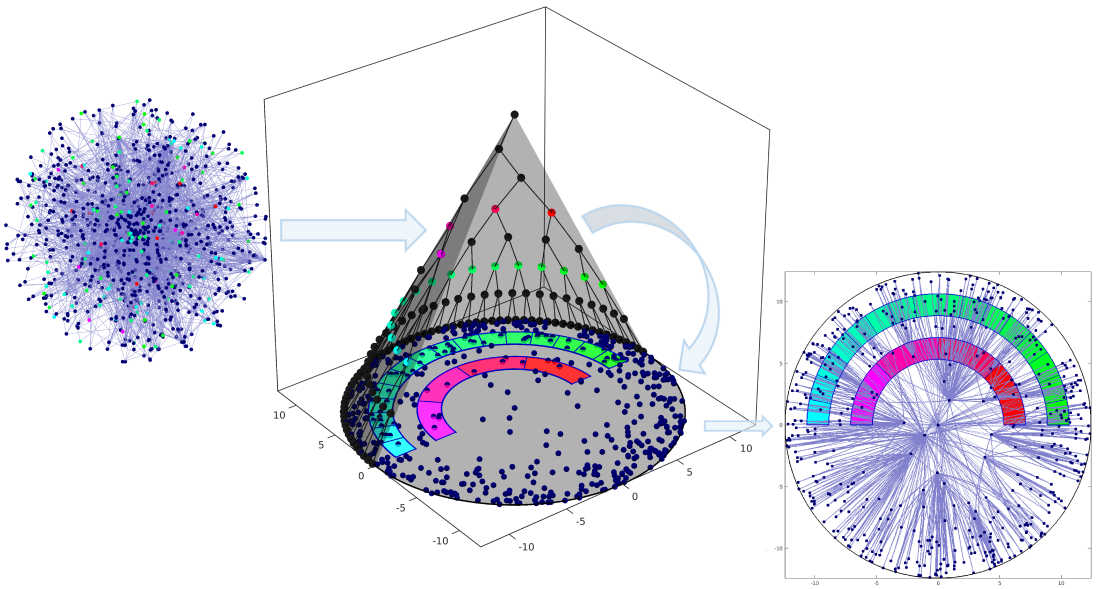

FIGURE 1.1: **The analogy between the hidden tree-like structure of complex networks and the hyperbolic plane.** Even in the absence of an obvious hierarchical organization, networks often have a tree-like structure underneath, formed e.g. by a containment hierarchy among the node attribute categories. On the left, some nodes of a network are colored according to their class of traits with different red and green hues, while in the middle the network's underlying tree is shown. Trees can be interpreted as discrete hyperbolic spaces, meaning that the nodes of the given tree can be mapped to the domains of the native disk that represents a hyperbolic plane. As the number of nodes of the tree grows exponentially downwards, the number of domains to which an annulus on the native disk is divided increases exponentially outwards; nevertheless, since the hyperbolic plane expands exponentially, each colored domain of the native disk corresponding to a single node of the tree has the same area. The illustrated connection between the hidden tree-like structure of a network and the hyperbolic plane implies that hyperbolic geometry can reveal a natural representation of complex networks, as exemplified by the layout on the right.

1.1 Native representation of the hyperbolic space

Throughout this work, when generating or embedding a network in the d-dimension-al hyperbolic space of curvature $K < 0$, the nodes will be placed simply in a ball of infinite radius in the d-dimensional Euclidean space (for which $K = 0$), which we call the native ball (or, in the two-dimensional case, the native disk). In this ball, a point's usual, Euclidean radial coordinate r (corresponding to its Euclidean distance from the ball's center) is equal to its hyperbolic radial coordinate, i.e. its hyperbolic distance from the center of the ball. Besides, the angles between hyperbolic lines crossing the ball can be interpreted in the usual way, meaning that the angular distance 'in the hyperbolic sense' is just the same as its Euclidean counterpart. However, most hyperbolic lines differ from the Euclidean lines and can be depicted as an arc intersecting the boundary of the native ball perpendicularly – the only exceptions are the diameters of the ball that can be considered both as Euclidean and hyperbolic lines.

The most important thing to keep in mind when working in a hyperbolic space instead of a Euclidean one is that the hyperbolic distances have to be measured along not Euclidean, but hyperbolic lines, and thus, the hyperbolic distance formula strongly differs from the Euclidean one. Namely, the hyperbolic distance x between two points given by the Cartesian coordinate vectors $\underline{u} = (u_1, u_2, ..., u_d)$ and $\underline{v} = (v_1, v_2, ..., v_d)$ of norms $\|\underline{u}\| = \sqrt{\sum_{q=1}^{d} u_q^2} \equiv r_u$ and $\|\underline{v}\| = = \sqrt{\sum_{q=1}^{d} v_q^2} \equiv r_v$ is usually calculated from the hyperbolic law of cosines written in the native

representation of the hyperbolic space as

$$\cosh(\zeta x) = \cosh(\zeta r_u)\cosh(\zeta r_v) - \sinh(\zeta r_u)\sinh(\zeta r_v)\cos(\theta_{u,v}), \tag{1.1.1}$$

where $\zeta = \sqrt{-K}$, and $\theta_{u,v} = \arccos\left(\frac{\underline{u}\cdot\underline{v}}{\|\underline{u}\|\,\|\underline{v}\|}\right) = \arccos\left(\frac{\sum_{q=1}^{d} u_q v_q}{r_u r_v}\right)$ is the angular distance between the given two points. Note that $r_u = 0$ yields $x = r_v$, and for $r_v = 0$ simply $x = r_u$. In the case of $\theta_{u,v} = 0$, $x = |r_u - r_v|$, while for $\theta_{u,v} = \pi$, $x = r_u + r_v$.

As it is described in Ref. [2], for sufficiently large ζr_u and ζr_v (for which $e^{-\zeta r_u} \approx 0$ and $e^{-\zeta r_v} \approx 0$) and an angular distance $\theta_{u,v}$ that is larger than $2 \cdot \sqrt{e^{-2\zeta r_u} + e^{-2\zeta r_v}}$ but small enough to allow the approximation $\sin(\theta_{u,v}/2) \approx \theta_{u,v}/2$, the hyperbolic distance x can be written as

$$x \approx r_u + r_v + \frac{2}{\zeta} \cdot \ln\left(\sin\left(\frac{\theta_{u,v}}{2}\right)\right) \approx r_u + r_v + \frac{2}{\zeta} \cdot \ln\left(\frac{\theta_{u,v}}{2}\right). \tag{1.1.2}$$

This approximating formula of the hyperbolic distance conveys some substantial messages. First, it shows the contribution of the radial coordinates and the angular distance to be easily separable (just like e.g. in the case of the Euclidean inner product $\underline{u} \cdot \underline{v} = r_u \cdot r_v \cdot \cos(\theta_{u,v})$). Besides, it implies that small hyperbolic distances are yielded by small angular distances and/or small radial coordinates. As a comparison, although small Euclidean distances usually also correspond to small angular distances, there is no such radial position in Euclidean space that is relatively close to any other point of the space. Finally, Eq. (1.1.2) indicates that the domain that is hyperbolically close to a given point is stretched rather radially, towards the center of the native ball, than sideways (i.e., angularly), toward positions of similarly large radial coordinates. Specifically, a hyperbolic ball is usually drop shaped, as exemplified in the two-dimensional case in Fig. 1.1.1, unless it is centered on the origin of the native ball, in which case the hyperbolic ball looks the same as a Euclidean one, with the difference that its volume expands exponentially, (for not too small values of its radius r_{hyp} [13]) as

$$V_d^{\text{hyp}} = \frac{e^{\zeta\cdot(d-1)\cdot r_{\text{hyp}}} - 1}{\zeta \cdot (d-1) \cdot 2^{d-1}}, \tag{1.1.3}$$

instead of the Euclidean, polynomial formula

$$V_d^{\text{Euc}}(r_{\text{Euc}}) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} \cdot r_{\text{Euc}}^d. \tag{1.1.4}$$

1.2 Poincaré ball model of the hyperbolic space

Similarly to the native representation, in the Poincaré ball model the d-dimensional hyperbolic space of curvature $K = -\zeta^2$ is represented by a d-dimensional ball; however, the radius of the Poincaré ball is not infinite but 1, meaning that the points lying at a Euclidean distance of 1 from the ball's center correspond to the hyperbolically infinitely distant positions. In the Poincaré ball, the hyperbolic distance x between two points given by the Cartesian coordinate vectors $\underline{u} = (u_1, u_2, ..., u_d)$ and $\underline{v} = (v_1, v_2, ..., v_d)$ of norms $\|\underline{u}\| = \sqrt{\sum_{q=1}^{d} u_q^2}$ and $\|\underline{v}\| = \sqrt{\sum_{q=1}^{d} v_q^2}$ can be written as

$$x(\underline{u}, \underline{v}) = \frac{1}{\zeta} \cdot \operatorname{arccosh}\left[1 + 2 \cdot \frac{\|\underline{u} - \underline{v}\|^2}{(1 - \|\underline{u}\|^2) \cdot (1 - \|\underline{v}\|^2)}\right], \tag{1.2.1}$$

where $\|...\|$ denotes the Euclidean norm.

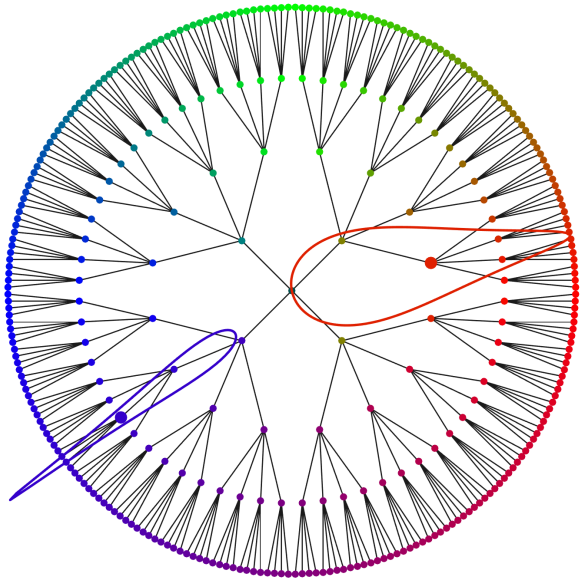

FIGURE 1.1.1: **The drop shape of hyperbolic circles.** The figure shows the native disk representing the hyperbolic plane, with a tree drawn on it that can be considered as a discretized version of the hyperbolic plane. The orange and purple curves are hyperbolic circles on the native disk, denoting those positions that lie at a given hyperbolic distance from the enlarged node in the center of the given circle. The radius of the orange circle is equal to the radial coordinate of the enlarged orange node; thus, as in the native representation the radial coordinates are equal to the hyperbolic distances measured from the center of the disk, this orange circle goes through the origin, i.e. the root of the analogous tree. The radius of the purple circle is the same as that of the orange one. Both curves show that positions that are hyperbolically close to a given point lie in an increasing angular range toward the disk center, showing that smaller radial coordinates can compensate for larger angular distances. Note that, however, due to the outward (exponential) expansion of the hyperbolic plane (indicated by the increase of the number of leaves of the analogous tree towards its outer levels), the hyperbolic area of a larger angular sector of an origin-centric ring close to the disk center is actually not bigger than the hyperbolic area of an angularly more restricted sector that lies farther away from the origin.

According to this formula, the hyperbolic distance between the point given by $\underline{u}$ and the ball center $\underline{o} = (0, 0, ..., 0)$ is

$$x(\underline{u}, \underline{o}) = \frac{1}{\zeta} \cdot \operatorname{arccosh}\left[1 + 2 \cdot \frac{\|\underline{u}\|^2}{(1 - \|\underline{u}\|^2)}\right], \tag{1.2.2}$$

meaning that a radial coordinate $r_{\text{Poincaré}}$ given in the Poincaré ball model can be converted to a radial coordinate r_{native} in the native representation (where the radial coordinate is equal to the hyperbolic distance from the origin) with the formula

$$r_{\text{native}} = \frac{1}{\zeta} \cdot \operatorname{arccosh}\left[1 + 2 \cdot \frac{r^2_{\text{Poincaré}}}{(1 - r^2_{\text{Poincaré}})}\right]. \tag{1.2.3}$$

The angular coordinates are the same in the two models; thus, the direction of the position vectors does not change during the Poincaré-native conversion, only their length (i.e., their Euclidean norm).

1.3 The hyperboloid model of the hyperbolic space

The hyperboloid model represents the d-dimensional hyperbolic space of curvature $K = -\zeta^2$ in the $d+1$-dimensional Euclidean space as the upper sheet of a two-sheet hyperboloid. For example, the three-dimensional hyperboloid depicted in Fig. 1.3.1 can be used for visualizing the hyperbolic plane. In the hyperboloid model, the hyperbolic distance between two points given by the Cartesian coordinate vectors $\underline{u} = (u_1, u_2, ..., u_d, u_{d+1})$ and $\underline{v} = (v_1, v_2, ..., v_d, v_{d+1})$ can be calculated as

$$x(\underline{u}, \underline{v}) = \frac{1}{\zeta} \cdot \text{arccosh}(\underline{u} \circ \underline{v}), \tag{1.3.1}$$

where $\underline{u} \circ \underline{v}$ is the Lorentz product

$$\underline{u} \circ \underline{v} = u_1 v_1 - (u_2 v_2 + u_3 v_3 + ... + u_{d+1} v_{d+1}) \tag{1.3.2}$$

between the two position vectors. Note that the Lorentz product of a position vector lying on the hyperboloid with itself is 1, yielding $x(\underline{u}, \underline{u}) = \frac{1}{\zeta} \cdot \text{arccosh}(1) = 0$.

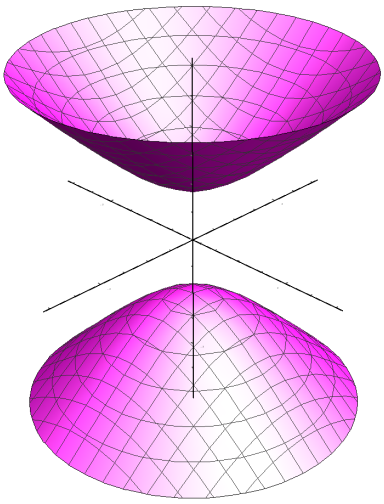

FIGURE 1.3.1: **Hyperboloid of two sheets.** The coordinate denoted as the first one is measured along the vertical axis, and is always positive in the case of the upper sheet.

In the hyperboloid, the position vector corresponding to the center of the native ball is $\underline{o} \equiv (o_1, o_2, o_3, ..., o_{d+1}) = (1, 0, 0, ..., 0)$. According to Eqs. (1.3.1) and (1.3.2), the hyperbolic distance between $\underline{o}$ and a point of the hyperboloid given by $\underline{u} \in \mathbb{R}^{d+1}$, i.e. the given point's radial coordinate (or the Euclidean norm of its position vector) in the native representation of the hyperbolic space can be written as

$$\begin{aligned} x(\underline{u}, \underline{o}) &= \frac{1}{\zeta} \cdot \text{arccosh}(\underline{u} \circ \underline{o}) = \frac{1}{\zeta} \cdot \text{arccosh}(u_1 \cdot 1 - (u_2 \cdot 0 + u_3 \cdot 0 + ... + u_{d+1} \cdot 0)) = \\ &= \frac{1}{\zeta} \cdot \text{arccosh}(u_1), \end{aligned} \tag{1.3.3}$$

meaning that the radial coordinate of a point in the native ball can be expressed with its first coordinate in the hyperboloid in itself. Meanwhile, the point's direction vector in the native ball can be expressed using solely its other hyperboloid coordinates as $\frac{(u_2, u_3, ..., u_{d+1})}{\sqrt{u_2^2 + u_3^2 + ... + u_{d+1}^2}}$.

2 Models of network growth based on hyperbolic geometry

Hyperbolic network models [2–4, 7, 14–16] generate networks by connecting nodes that are arranged in a geometric space of some negative curvature K. Several models have been proposed that differ either in the way the network nodes are distributed in the hyperbolic space (community formation can be influenced via predefined, non-uniform angular distributions) or in the mechanism of the network formation (static models only create connections after assigning a spatial position to all the network nodes, while dynamic models introduce the nodes and their links gradually). However, the connection rule is always the same in hyperbolic network models: smaller hyperbolic distances yield higher connection probability. Thus, according to the approximating formula $x_{ij} \approx r_i + r_j + \frac{2}{\sqrt{-K}} \cdot \ln\left(\frac{\Delta\theta_{ij}}{2}\right)$ [2], nodes at smaller r radial coordinates and smaller $\Delta\theta$ angular distances become more attractive for the other nodes, and the connection preference towards the nodes at the smallest hyperbolic distances can be interpreted as an optimization of a trade-off between a popularity (radial) and a similarity (angular) component. The first dynamic one of the hyperbolic network models was named after this principle as popularity-similarity optimization (PSO) model [3], although the static models [2, 7, 15] can be viewed as a manifestation of the very same theorem too.

This chapter reviews some variants and properties of the popularity-similarity optimization model with a particular emphasis on those aspects where my work also contributed to the field of dynamic hyperbolic network models. First, Sect. 2.1 describes the original PSO model [3], where the network nodes are introduced one by one on the hyperbolic plane and each node connects at its appearance to a given number of previously appeared nodes. Then, Sect. 2.2 shows how the model can be extended with the emergence of links that connect previously appeared nodes [3] and how the creation of these so-called internal links can be simulated by making the number of links formed by the new node a time-dependent function [4]. After that, I generalize the E-PSO model to also simulate the destruction of internal links besides their formation [T1], and demonstrate the capability of this generalized E-PSO model for making the average degree of the subgraphs that span between nodes whose degree has risen above a given degree threshold a nonconstant, either increasing or decreasing function of the degree threshold [T1], which property was not recognized previously. As another less-recognized property of the model, Section 2.3 demonstrates through the results of my numeric investigation [T2] that despite the uniformity of the angular distribution of the network nodes, the PSO model is able to generate networks with strong community structure. Finally, Sect. 2.4 describes how I generalized the algorithm of the original, two-dimensional PSO model to any number of dimensions d [T3], and presents my numerical study regarding the degree distribution (for which an analytical formula was derived in a joint work with Sámuel G. Balogh in Ref. [T3]), the average clustering coefficient and the community structure of dPSO networks.

2.1 The original, two-dimensional popularity-similarity optimization model

The popularity-similarity optimization (PSO) model [3] places the network nodes one by one on the native disk representation [2] of the hyperbolic plane and connects the newly arriving node to the previously appeared ones with probabilities determined by its hyperbolic distance from the older nodes. Its most basic version (i.e., the case of using a deterministic connection

rule and no popularity fading) simulates such a network growth where a new node always connects to previously appeared nodes minimizing simply the product between a candidate's birth time (that is assumed to indicate the candidate's popularity as older nodes have more chances to form connections) and (following the fundamental principle of homophily stating that nodes tend to connect with nodes of similar attributes) its distance from the new node in a one-dimensional attribute space, thus balancing between the two essential attractive features of the candidates given by their popularity and their similarity to the new node. In its more general form, the tunable parameters of the PSO model are the following:

- The curvature $K \in \mathbb{R}^-$ of the hyperbolic plane, tuned via the parameter $\zeta = \sqrt{-K}$. Changing ζ corresponds to a simple rescaling of the hyperbolic distances between all the node pairs. Usually ζ is simply set to 1 (i.e. $K = -1$).
- The number of nodes $N \in \mathbb{Z}^+$ at the end of the network growth.
- The number of links $m \in \mathbb{Z}^+$ established by each node at its appearance after the mth one, determining the average degree of the network ($\langle k \rangle \approx 2m$). The first m nodes form a complete graph.
- The popularity fading parameter $\beta \in (0, 1]$ that sets the speed of the outward drift of the nodes on the native disk during the network growth. $\beta = 1$ corresponds to no popularity fading, when the nodes do not move away from their initial position. The exponent γ of the power-law decaying tail of the degree distribution ($p(k) \sim k^{-\gamma}$) can be calculated from the popularity fading parameter as $\gamma = 1 + 1/\beta$.
- The temperature $0 \leq T$, $T \neq 1$ that controls the average clustering coefficient $\bar{c}$ of the network. Lower temperatures result in higher average clustering coefficients. Small values of T can yield strong clustering even for large number of nodes N[1], while the clustering is zero in the $N \to \infty$ limit for any temperature above 1.

To demonstrate the impact of the different model parameters, Figs. 2.1.1–2.1.3 show PSO networks on the native disk, as they were produced by the model.

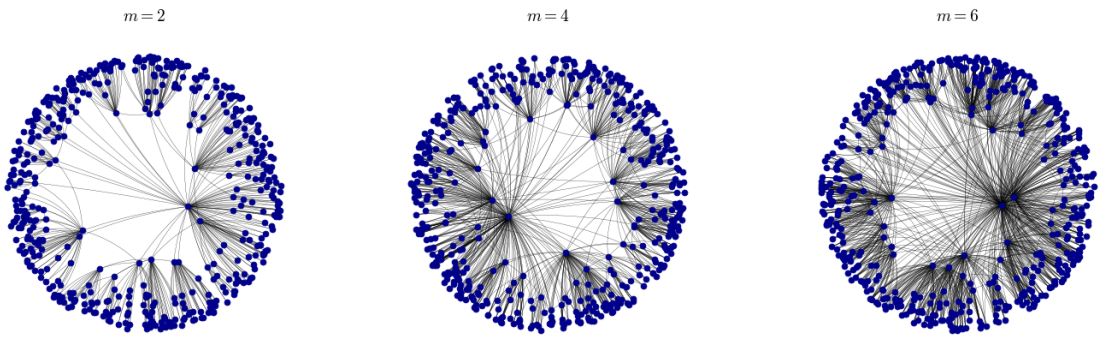

FIGURE 2.1.1: **PSO networks generated at different settings of the parameter m.** From left to right, with the increase in m, the average degree $\langle k \rangle \approx 2m$ becomes larger. Each network was constructed in the native representation of the hyperbolic plane of curvature $K = -1$, setting the total number of nodes N to 500, the popularity fading parameter β to $2/3$ and the temperature T to 0.

[1]Note that in the case of networks that grow simply according to the rule of preferential attachment [17], clustering is asymptotically zero [18].

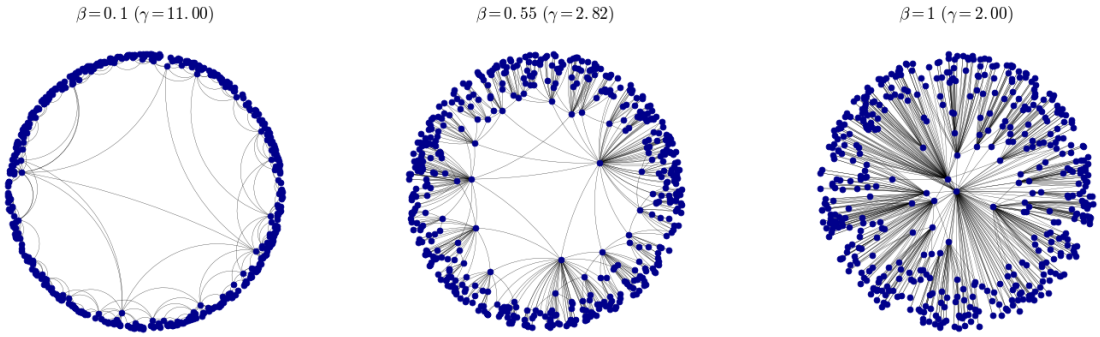

FIGURE 2.1.2: **PSO networks generated at different settings of the popularity fading parameter β.** From left to right, with the increase in β, the outward drift of the nodes during the network growth slows down and the occurring largest node degree increases, i.e. the degree decay exponent γ decreases. Each network was constructed in the native representation of the hyperbolic plane of curvature $K = -1$, setting the total number of nodes N to 500, the parameter m to 2 (yielding $\langle k \rangle \approx 4$) and the temperature T to 0.

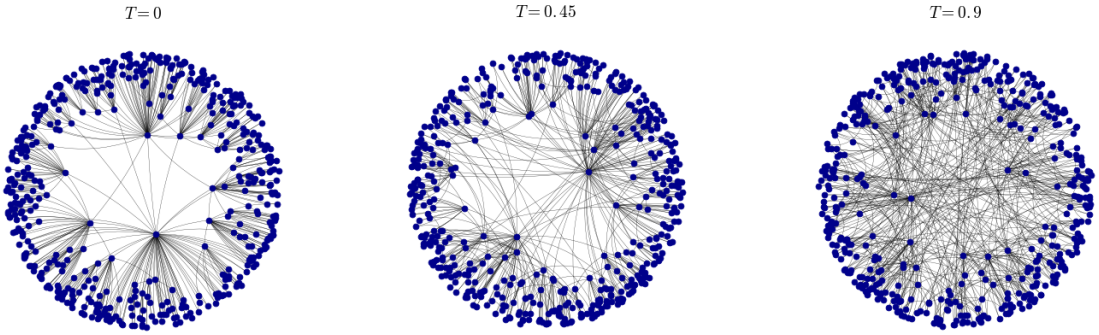

FIGURE 2.1.3: **PSO networks generated at different settings of the temperature T.** From left to right, with the increase in T, connections between farther nodes become more probable and the average clustering coefficient decreases. Each network was constructed in the native representation of the hyperbolic plane of curvature $K = -1$, setting the total number of nodes N to 500, the parameter m to 2 (yielding $\langle k \rangle \approx 4$) and the popularity fading parameter β to $2/3$.

An example of the network generation process is depicted in Fig. 2.1.4. But exactly how the networks are constructed in the PSO model? The random graph generation starts from an empty network, to which a new node j is added at each time step $j = 1, 2, ..., N$ as follows:

1. The new node j appears on the native disk with an angular coordinate θ_j sampled uniformly at random from $[0, 2\pi)$ and the radial coordinate

   (a) $r_{jj} = \frac{2}{\zeta}\ln(j)$ at $T < 1$, and

   (b) $r_{jj} = \frac{2T}{\zeta}\ln(j)$ at $1 < T$,

   which yields a radial node density that increases exponentially when moving outwards on the exponentially expanding hyperbolic plane [4].

2. To simulate popularity fading – or in other words, to mitigate the extent to which the birth time affects a node's attractivity compared to its similarity to the other nodes –, the radial coordinate of each previously (at time $i < j$) appeared node i is increased. The outward drift is carried out according to the formula $r_{ij} = \beta r_{ii} + (1 - \beta) r_{jj}$. Note that the radial order of the nodes always remains the same as their appearance order, expressing the

early-appearing nodes' higher attractivity that arises from their higher popularity, i.e. the larger number of their connections collected through their longer lifespan.

3. The new node j connects to previously appeared nodes. Only one connection can be established between each node pair.

   (a) At the first $m+1$ time steps, when the number of older nodes does not exceed m, the new node j connects to all of them.

   (b) At the further steps, m older nodes are selected to which the new node j becomes connected.

      i. At $T = 0$, the connection rule is deterministic: node j simply connects to those m nodes, for which the hyperbolic distance x calculated according to the hyperbolic law of cosines is the smallest.

      ii. At temperatures $T > 0$, any older node $i = 1, 2, ..., j-1$ connects to the new node j with probability

$$p(x_{ij}) = \frac{1}{1 + e^{\frac{\zeta}{2T}(x_{ij} - R_j)}}, \tag{2.1.1}$$

      where the cutoff distance R_j is set to

      A. $R_j = r_{jj} - \frac{2}{\zeta} \cdot \ln\left[\frac{2T}{\sin(T\pi)} \cdot \frac{1 - e^{-\frac{\zeta}{2}(1-\beta)r_{jj}}}{m(1-\beta)}\right]$ if $\beta < 1$ and $T < 1$,

      B. $R_j = r_{jj} - \frac{2}{\zeta} \cdot \ln\left[\frac{T}{\sin(T\pi)} \cdot \frac{\zeta r_{jj}}{m}\right]$ if $\beta = 1$ and $T < 1$,

      C. $R_j = r_{jj} - \frac{2T}{\zeta} \cdot \ln\left[\left(\frac{2}{\pi}\right)^{\frac{1}{T}} \cdot \frac{T}{T-1} \cdot \frac{1 - e^{-\frac{\zeta}{2T}(1-\beta)r_{jj}}}{m(1-\beta)}\right]$ if $\beta < 1$ and $1 < T$, or

      D. $R_j = r_{jj} - \frac{2T}{\zeta} \cdot \ln\left[\left(\frac{2}{\pi}\right)^{\frac{1}{T}} \cdot \frac{T}{T-1} \cdot \frac{\zeta r_{jj}}{2Tm}\right]$ if $\beta = 1$ and $1 < T$,

      ensuring that the expected number $\bar{k}_j$ of older nodes to which the new node j connects is equal to m.

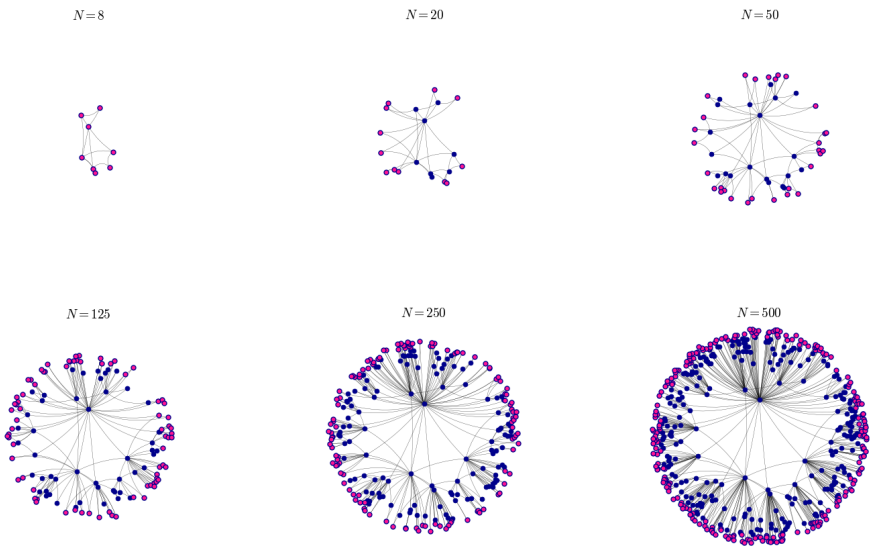

FIGURE 2.1.4: **Snapshots of a network growth in the PSO model.** Each layout depicts the nodes that appeared after the previous snapshot in pink while showing the older nodes in blue. The network was generated in the native representation of the hyperbolic plane of curvature $K = -1$, setting the final network size to 500, the parameter m to 2 (yielding $\langle k \rangle \approx 4$), the popularity fading parameter β to 2/3 (corresponding to the degree decay exponent $\gamma = 2.5$) and the temperature T to 0.

It is important to note that the trade-off between the popularity and the similarity of the nodes can not always embody the essence of the connection rule in real networks, and thus, the PSO model is not the most appropriate choice for simulating all types of networks. As an example, in Sect. III./C of the Supplementary Methods of Ref. [3], an actor network was mentioned in which two actors are connected if there is at least one film in which they have co-starred. Despite having a heterogeneous degree distribution and a relatively high average clustering coefficient, this network is claimed to be hard to interpret as a sample from the PSO model since the films of larger crews yield cliques (i.e., fully connected subgraphs) that include numerous dissimilar and not necessarily outstandingly famous (or popular) actors, whose connection can not be attributed either to their similarity or their popularity.

Yet, the PSO model provides a very valuable benchmark for the analysis of real-like networks, as it is capable of reproducing the most frequently mentioned common characteristics of real-world networks, given by the scale-free degree distribution, the small-world property and a strongly clustered structure. Figure 2.1.5 exemplifies that a heterogeneous degree distribution is indeed achievable in the PSO model and the expected power law formula $p(k) \sim k^{-\gamma}$ with the decay exponent $\gamma = 1 + 1/\beta$ can be fitted well to the degree distribution of a PSO network. Besides, the presence of the small-world property in PSO networks is illustrated by Fig. 2.1.6, showing for several parameter settings of the PSO model that the average of the shortest path lengths between all pairs of nodes does not increase faster with the number of nodes than logarithmically. Lastly, Fig. 2.1.7 demonstrates how the average clustering coefficient of PSO networks depends on the model parameters and confirms that – at the range of smaller temperatures – particularly high average clustering coefficients can arise from the model.

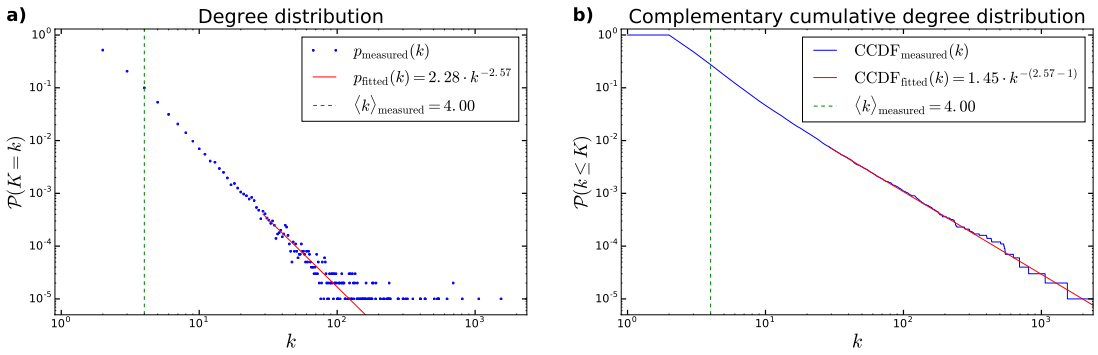

FIGURE 2.1.5: **Heterogeneous degree distribution generated by the PSO model.** Panel a) depicts the degree distribution (i.e., the fraction of nodes with degree k as a function of k) of a PSO network of $N = 10^5$ number of nodes generated on the hyperbolic plane of curvature $K = -1$ using the model parameters $m = 2$, $\beta = 2/3$ and $T = 0.3$, while panel b) shows the node degrees' complementary cumulative distribution function (CCDF, corresponding to the fraction of nodes having a degree that is not smaller than k as a function of k) of the same network. The red curves were obtained following the fitting method described in Ref. [19]. The measured data (blue) match well the curve $\mathcal{P}(K = k) = C \cdot k^{-\gamma}$ with a given C constant in panel a) and also follow closely the formula $\mathcal{P}(k \leq K) = \frac{C}{\gamma - 1} \cdot k^{-(\gamma-1)}$ in panel b), as expected. The degree decay exponent $\gamma_{\text{measured}} = 2.57$ yielded by the fitting is close to the value $\gamma_{\text{expected}} = 1 + 1/\beta = 2.50$ that was expected based on the analytical calculations of Ref. [3]. Besides, the measured average of the node degrees $\langle k \rangle_{\text{measured}}$ (denoted by the dashed green line in both panels) is practically the same as the expected value of $2m = 4$.

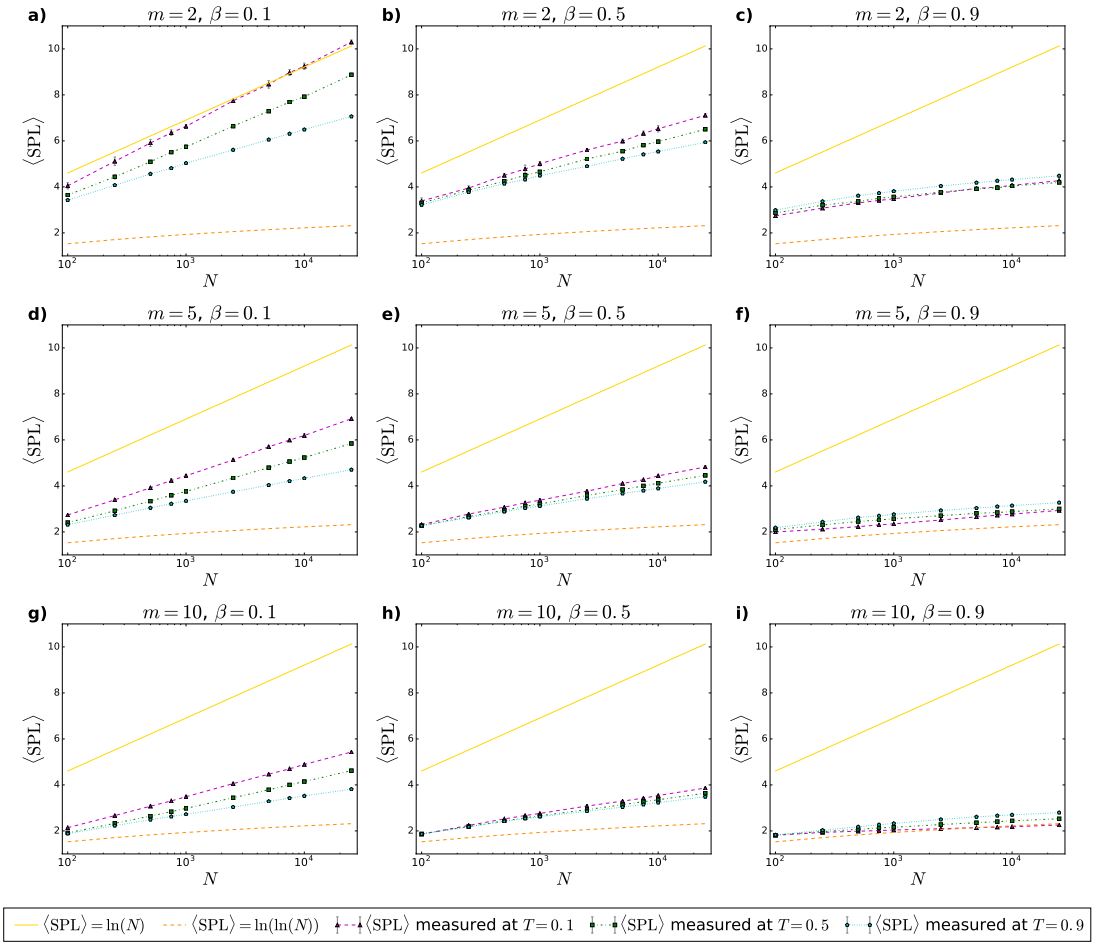

FIGURE 2.1.6: **Small-world property in the PSO model.** The curves show the average shortest path length $\langle \mathrm{SPL} \rangle$ (i.e., the smallest possible number of steps it takes to reach a node from another, averaged over all node pairs) measured in PSO networks as a function of the number of network nodes N, increasing moderately compared to the $\langle \mathrm{SPL} \rangle = \ln(N)$ curve and, in the case of large average degrees and slowly decaying degree distributions, approaching even the $\langle \mathrm{SPL} \rangle = \ln(\ln(N))$ curve, which indicates the appearance of not only small-world but even ultra small-world behavior in PSO networks. Each subplot corresponds to a certain setting of the PSO model parameters m (regulating the average degree as $\langle k \rangle \approx 2m$) and β (controlling the degree decay exponent as $\gamma = 1 + 1/\beta$), as written in the panel title. The results of different settings of the temperature parameter T are shown in different colors, as listed in the common legend at the bottom of the figure. The curvature K of the hyperbolic plane was always set to -1. 10 PSO networks were generated with each parameter setting and the displayed data points were obtained by averaging over the 10 network realizations. The (usually very small) grey error bars depict the standard deviation among the 10 networks.

2.2 Generalisation with internal links and the analogous E-PSO model

To obtain a more realistic model of network growth, one can use the generalized popularity-similarity optimization model introduced in Section VIII of the Supplementary Information of Ref. [3], where besides connecting to the newly appeared node, the previously appeared nodes are also allowed to form new connections with each other in each time step. According to the model definition, these so-called internal links are sampled according to distance-dependent probabilities, just like the external links that make the new node connected to the older ones. As derived in Ref. [4], if besides the m number of external links also L_+ number of internal links are formed between previously disconnected node pairs in each time step in the generalized PSO

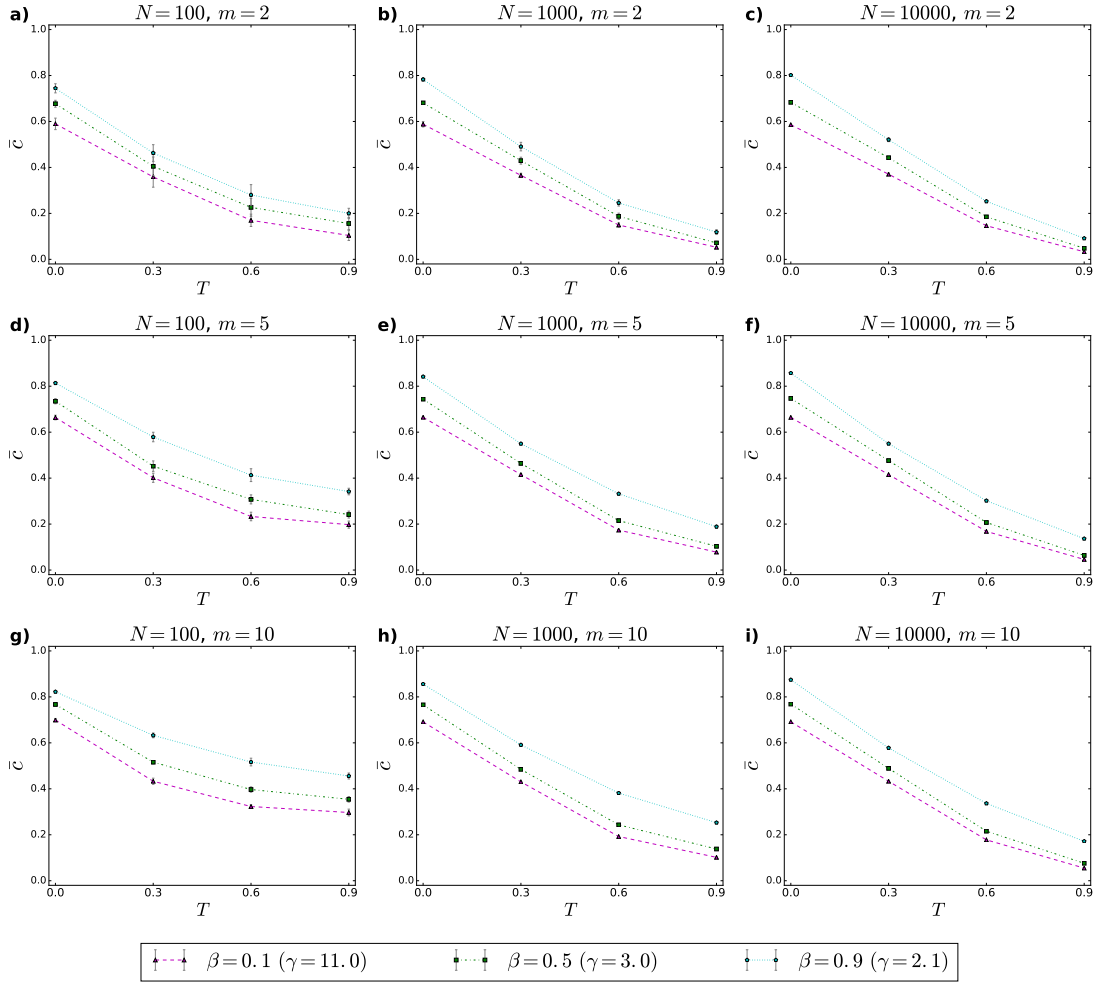

FIGURE 2.1.7: **High clustering of PSO networks.** The curves show the average clustering coefficient $\bar{c}$ (i.e., the fraction of the neighboring pairs of a node that are connected to each other, averaged over all the network nodes) measured in PSO networks of different parameter sets as a function of the temperature T, which affects $\bar{c}$ the most among the model parameters. Each subplot refers to a certain setting of the total number of network nodes N and the model parameter m that regulates the average degree as $\langle k \rangle \approx 2m$, as written in the panel title. As it is listed in the common legend at the bottom of the figure, the curves of different colors correspond to different settings of the popularity fading parameter β that primarily controls the degree decay exponent as $\gamma = 1 + 1/\beta$ but also has a moderate impact on $\bar{c}$. The curvature K of the hyperbolic plane was set to -1 in each case. 10 PSO networks were generated with each parameter setting and the displayed data points were obtained by averaging over the 10 network realizations. The (usually very small) grey error bars depict the standard deviation among the 10 networks.

model, then the expected total number of previously (at time $i < j$) appeared nodes to which the node appearing at time j connects during the network growth (until forming a network of N nodes) can be written as

$$\bar{m}_j = \begin{cases} m + L_+ \cdot \frac{1}{[1-N^{-0.5}]^2} \cdot \ln\left(\frac{N}{j}\right) \cdot \left[1 - j^{-0.5}\right] & \text{if } \beta = 0.5, \\ m + L_+ \cdot \frac{2}{\ln^2(N)} \cdot \left[\frac{N}{j} - 1\right] \cdot \ln(j) & \text{if } \beta = 1.0, \\ m + L_+ \cdot \frac{2\cdot(1-\beta)}{\left[1-N^{-(1-\beta)}\right]^2 \cdot (2\cdot\beta - 1)} \left[\left(\frac{N}{j}\right)^{2\cdot\beta-1} - 1\right] \left[1 - j^{-(1-\beta)}\right] & \text{otherwise.} \end{cases} \tag{2.2.1}$$

This is used by the so-called E-PSO model [4] that, although working only with external links, simulates a similar[2] network growth as the generalized PSO model by simply changing the number of new external links created at time j from the constant m value used in the original PSO model to the time-dependent function $\bar{m}_j$ given by Eq. (2.2.1). Creating all the links as an external one, the E-PSO model has a great advantage from the point of view of the running time of the network generation process compared to the generalized PSO model that needs the calculation of a distance-based connection probability between each pair of the previously appeared nodes in each time step. Therefore, during my work, I focused on the E-PSO model instead of the generalized PSO model.

A phenomenon that is originally not included in the E-PSO model but I added to it in Ref. [T1] is the possible disappearance of internal links during the network growth. A natural expectation is that when simulating such a network growth in which the number of added internal links is equal to the number of deleted internal links in each time step, then – since the positive and negative changes in the number of internal links cancel each other out and thus, do not affect the total number of edges in the network – one must get back the average degree $\langle k \rangle \approx 2 \cdot m$ of the original PSO model for any settings of the other model parameters. As it was shown in Ref. [4], setting the expected number of new links created at time j to $\bar{m}_j$ written in Eq. (2.2.1) yields an average degree of $\langle k \rangle \approx 2 \cdot (m + L_+)$ for the whole network of N nodes. Thus, in an E-PSO network that besides the appearance of $L_+ > 0$ number of new internal links also simulates the disappearance of $L_- > 0$ number of previously emerged internal links in each time step, the average degree obviously must follow the formula $\langle k \rangle \approx 2 \cdot (m + L_+ - L_-)$ that gives back $\langle k \rangle \approx 2 \cdot m$ for $L_+ = L_-$, meaning that the sum of the increments and the sum decrements in the number of links over all the time steps have to be equal to each other. Since this criterion must be fulfilled for any final network size N, i.e. for any number of time steps of the summation, thus the step-by-step effect of the two competing changes has to cancel each other out. Consequently, in my extended E-PSO model [T1] that is capable of simulating both the emergence and the deletion of internal links, the expected number of links created by node j at its appearance (i.e., at time j) is defined as

$$\bar{m}_j = \begin{cases} m + [L_+ - L_-] \cdot \frac{1}{[1-N^{-0.5}]^2} \cdot \ln\left(\frac{N}{j}\right) \cdot \left[1 - j^{-0.5}\right] & \text{if } \beta = 0.5, \\ m + [L_+ - L_-] \cdot \frac{2}{\ln^2(N)} \cdot \left[\frac{N}{j} - 1\right] \cdot \ln(j) & \text{if } \beta = 1.0, \\ m + [L_+ - L_-] \cdot \frac{2\cdot(1-\beta)}{\left[1-N^{-(1-\beta)}\right]^2\cdot(2\cdot\beta-1)} \left[\left(\frac{N}{j}\right)^{2\cdot\beta-1} - 1\right] \left[1 - j^{-(1-\beta)}\right] & \text{otherwise,} \end{cases} \tag{2.2.2}$$

yielding $\bar{m}_j \equiv m$ for $L_+ = L_-$, as expected. The only difference compared to Eq. (2.2.1) is that here, instead of $L_+ \geq 0$, the stepwise net number $L \equiv L_+ - L_-$ of added and removed internal links is used as a multiplying factor, which is not restricted to be nonnegative. Note that accordingly, the extended E-PSO model can not distinguish between different combinations of L_+ and L_- if $L = L_+ - L_-$ is the same, meaning that rather than simulating a network growth with specified rates of both the appearance and disappearance of internal links, this model simulates processes of given net changes in the number of internal links.

An important but previously not recognized consequence of introducing operations on internal links in the PSO model is that, using $L > 0$, it enables the emergence of a densification

[2]While the E-PSO model creates all the inward-pointing links of a node at its appearance, such links of a node in the generalized PSO model emerges partly as external links at the node's appearance and partly as internal links in following time steps, i.e. after the outward shift (or popularity fading) of the node, during which the distance (or attractivity) relations between the given node and its potential neighbors might have changed. Therefore, in practice, if one generates a network with the generalized PSO model and with the analogous E-PSO model using the same angular coordinates of the nodes, then the edge list obtained from the two models can be different even in the case of $T = 0$ that yields deterministic connections. Nevertheless, as derived in Ref. [4], in the $N \to \infty$ limit the probability that two nodes become connected until the end of the network growth is the same in the two models.

law for the subgraphs of nodes of large enough degree with the increase in the degree threshold, which is claimed to be a common feature of several real networks in Refs. [20, 21]. To show this, Fig. 2.2.1 depicts the average internal degree $\langle k_{\text{internal}} \rangle$ of the subgraphs spanning between nodes having a degree larger than a certain threshold $k_{\min}$ as a function of $k_{\min}$ for both positive and negative L values (indicated by different colors) at different β and T parameter settings. As it was described in Ref. [T1], for negative values of L (i.e., when simulating such a network growth where at each time step more internal links are deleted than created), with the increase of the degree threshold the average internal degree decreases even at relatively small values of the threshold. At $L = 0$ (i.e., in the case of the original PSO model), the average internal degree remains constant until the degree threshold does not become so large that the subgraphs become extremely small. And lastly, when L is positive (i.e., when simulating such a network growth where at each time step the number of newly created internal links is larger than the number of deleted internal links), the average internal degree becomes larger as the degree threshold begins to increase – which corresponds to the desired densification phenomenon of Refs. [20, 21].

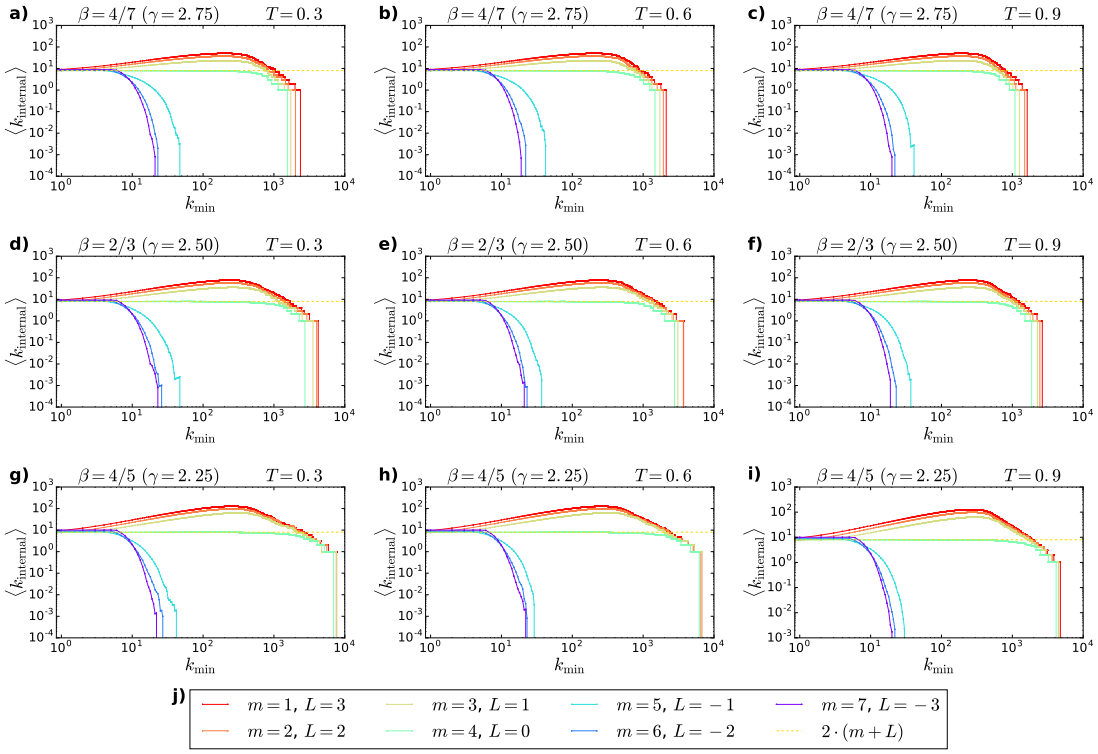

FIGURE 2.2.1: **The dependence of the average internal degree of subgraphs of nodes having a degree larger than a threshold on the value of the degree threshold in E-PSO networks of various parameter settings.** Each subplot corresponds to a certain $\beta - T$ setting written in the panel title, and each curve color corresponds to a certain setting of the parameters m (the number of links of the last node) and L (the simulated net change in the number of internal links per time step), as listed in panel j). The expected average degree $\langle k \rangle = 2 \cdot (m + L)$ was set to 8 for each network. With each parameter setting, one network of size $N = 10^5$ was generated on the hyperbolic plane of curvature $K = -1$. The figure was taken from Ref. [T1].

2.3 Automatic community formation in the PSO model

Up to this point, it has been demonstrated that the PSO model has the potential to provide a very general framework, using which several typical characteristics of real-world networks can be reproduced. However, based on their homogeneous spatial node arrangement, one could assume that PSO networks lack a very frequent network property, namely a modular structure. Therefore, different modifications of the model have been introduced [14, 16] with the aim of generating a community structure on the hyperbolic plane by making the angular distribution of the network nodes inhomogeneous. Nevertheless, as some studies [22–25] have pointed out, well-known community detection methods such as Louvain [26] usually divides the networks generated by the PSO model according to angular sectors of the hyperbolic disk, as exemplified by Fig. 2.3.1. Recognizing this phenomenon, I carried out in Ref. [T2] an extensive numerical study regarding the modular structures that can be detected in PSO networks in order to confirm that the angular inhomogeneity of the nodes is not a necessary condition for the emergence of communities, meaning that the PSO model actually has the capability to generate a strong and indeed relevant community structure. Along with the original PSO model [3] described in Sect. 2.1, Ref. [T2] also deals with the extended version [T1] of the E-PSO model [4] presented in Sect. 2.2, and investigates in details the static $\mathbb{S}^1/\mathbb{H}^2$ model [7, 20] too, showing that partitions of relatively high modularity[3] [27] can be found with different community detection approaches (namely asynchronous label propagation[4] [28], Louvain[5] [26] and Infomap[6] [29]) in a consistent way in the networks generated by any of these two-dimensional hyperbolic network models for a large variety of the model parameters, even though community formation was not an intention at the construction of these models.

As examples, Figs. 2.3.2 and 2.3.3 show the modularity Q of the partitions of PSO networks found by the asynchronous label propagation [28] and the Louvain [26] algorithms as a function of the parameters of the PSO model. For calculating the modularity $Q \in [-0.5, 1]$, as usual, a random network preserving the degree k_i of each node i was used as a non-modular reference point, and thus, the applied formula was

$$Q = \frac{1}{2E} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(A_{ij} - \frac{k_i k_j}{2E} \right) \delta_{c_i,c_j} \tag{2.3.1}$$

with E and N standing for the total number of links and nodes in the network, respectively, $A_{ij} = 1$ or 0 denoting whether nodes i and j are connected to each other or not, and $\delta_{c_i,c_j} = 1$ or 0 indicating whether nodes i and j were classed in the same community or not.

[3] Modularity measures the strength of a network partition by comparing the observed fraction of within-community links to its expected counterpart in the case of randomizing the connections between the nodes. It can be calculated in general as $Q = \frac{1}{2E} \sum_{i=1}^{N} \sum_{j=1}^{N} (A_{ij} - P_{ij}) \delta_{c_i,c_j}$, where E is the number of edges and N is the number of nodes in the network, A_{ij} is the element of the adjacency matrix corresponding to the node pair $i - j$, P_{ij} is the connection probability of the node pair $i - j$ in the null model that is considered as the non-modular reference, c_i denotes the community in which node i is classed in the given network partition and the Kronecker delta δ_{c_i,c_j} ensures that only such node pairs can contribute to the sum of which the members belong to the same community. High values of Q indicate pronounced, strong communities having large internal link density compared to the random expectation.

[4] After labeling each node with a unique community identifier, the asynchronous label propagation algorithm iterates over the nodes in random order repeatedly (using a new order for each round) and updates their label one by one: each node joins the community to which most of its neighbors currently belong.

[5] The Louvain algorithm performs a heuristic modularity maximization. Note that finding the exact maximum of modularity is a computationally hard problem.

[6] The Infomap algorithm aims at compressing the description of random walk trajectories through the categorization of the network nodes. More precisely, Infomap performs a heuristic minimization of the map equation that provides a theoretical lower bound for the code length of an average movement of an infinitely long random walk when using a map-like trajectory description based on a given community structure.

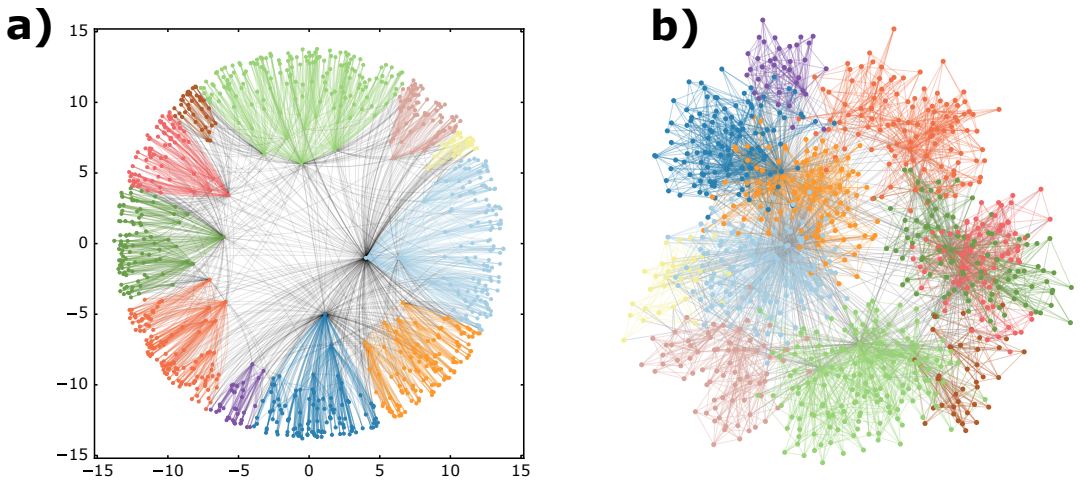

FIGURE 2.3.1: **The appearance of the modules detected by the Louvain algorithm in layouts of a PSO network.** The network of $N = 1000$ number of nodes was generated on the hyperbolic plane of curvature $K = -1$ using the model parameters $m = 5$ (resulting in the average degree $\langle k \rangle \approx 10$), $\beta = 0.7$ (corresponding to a degree decay exponent of $\gamma = 2.43$) and $T = 0.2$ (yielding an average clustering coefficient of 0.58). Panel a) depicts the network in the native representation of the hyperbolic space, using for each node its final position reached at the end of the network growth, showing that the modules found by Louvain occupy well-defined angular sectors in the hyperbolic disk. Panel b) illustrates a standard layout (namely the Prefuse force-directed layout) of the network on the Euclidean plane, demonstrating that the detected communities are clearly outlined even in such node arrangements that do not build on the hyperbolic origin of the networks. The figure was taken from Ref. [T2].

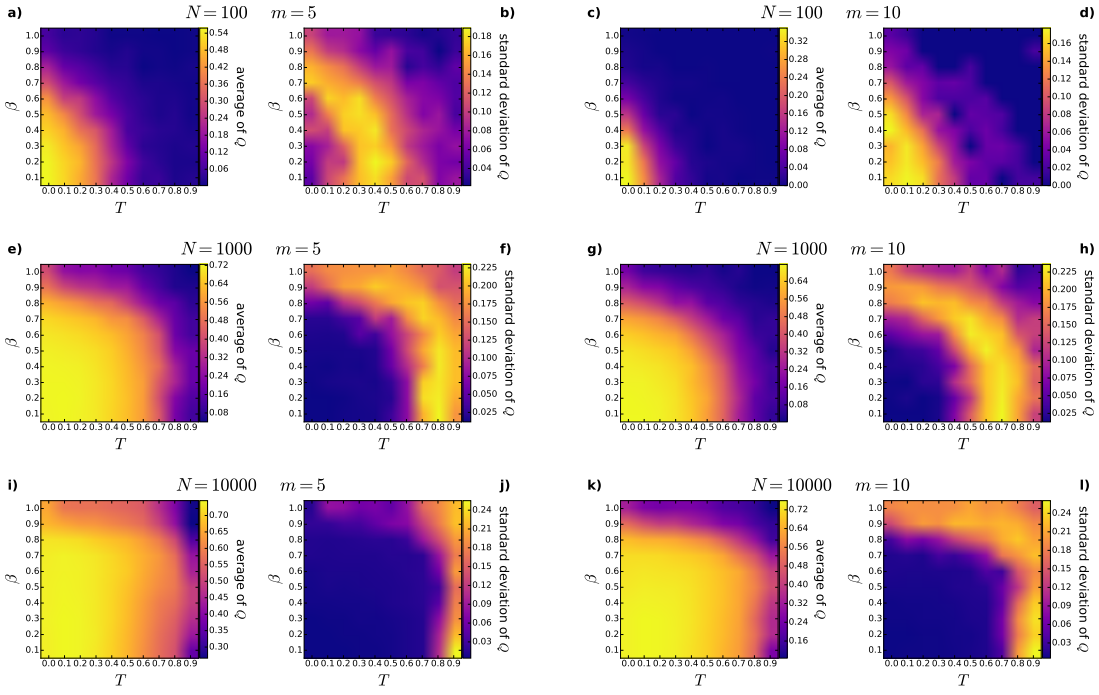

FIGURE 2.3.2: **The strength of the partitions of PSO networks detected by the asynchronous label propagation algorithm.** Each pair of subplots depicts for a given network size N and average degree $\langle k \rangle \approx 2m$ the average and the standard deviation of the measured modularity Q over 100 PSO networks as a function of the popularity fading parameter β and the temperature T. Each network was generated on the hyperbolic plane of curvature $K = -1$, using $\zeta = 1$. The figure was taken from Ref. [T2].

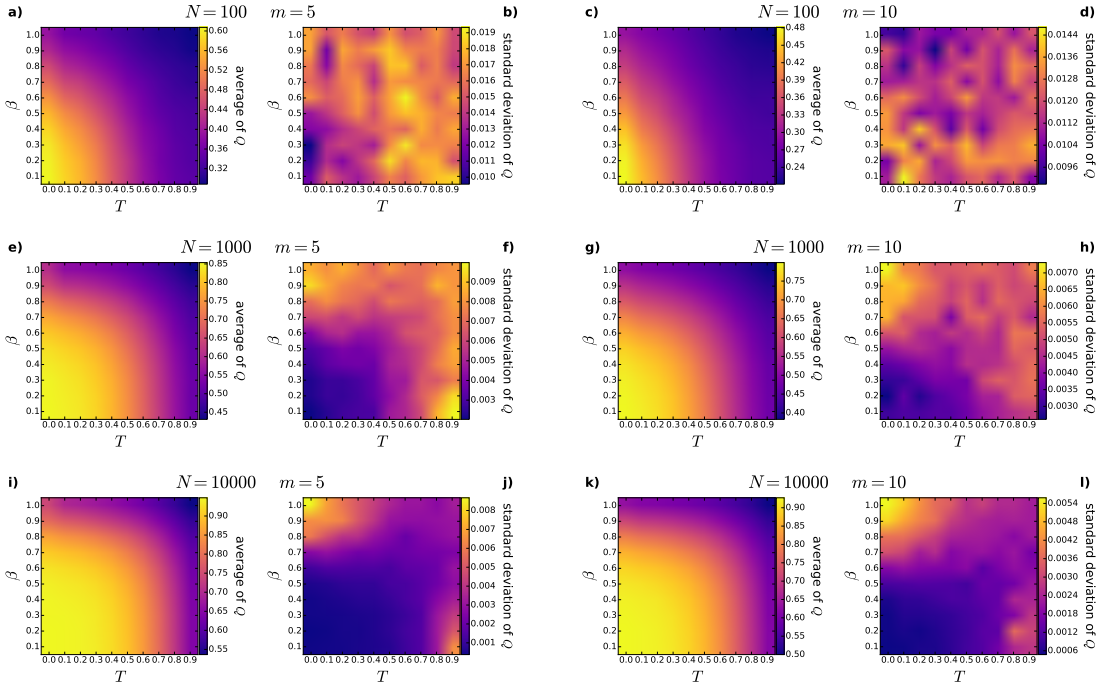

FIGURE 2.3.3: **The strength of the partitions of PSO networks detected by the Louvain algorithm.** Each pair of subplots depicts for a given network size N and average degree $\langle k \rangle \approx 2m$ the average and the standard deviation of the measured modularity Q over 100 PSO networks as a function of the popularity fading parameter β and the temperature T. Each network was generated on the hyperbolic plane of curvature $K = -1$, using $\zeta = 1$. The figure was taken from Ref. [T2].

TABLE 2.3.1: **Modularity in Erdős–Rényi (ER), Barabási–Albert (BA) and popularity-similarity optimization (PSO) graphs.** 100 networks were generated with all the models, using the same settings of the number of nodes N and the average degree $\langle k \rangle$ as in Fig. 2.3.3. Since the BA model yields a degree decay exponent $\gamma = 3.0$, for the PSO model (where $\gamma = 1 + 1/\beta$) the popularity fading parameter $\beta = 0.5$ was considered here. To maximally ensure the possibility of achieving high modularities, the communities were detected by the modularity-maximizing Louvain method in all the here-examined networks. Each cell of the table contains the average of the attained modularity over the 100 networks and the corresponding 95% confidence interval. Note that as it is shown in Fig. 2.3.3, with the increase in the temperature T, the modularity of PSO networks decreases.

	Erdős–Rényi model	Barabási–Albert model	PSO model at $\beta = 0.5$ and $T = 0$
$N = 100, \quad \langle k \rangle \approx 10$	0.2624 ± 0.0016	0.2498 ± 0.0017	0.5901 ± 0.0025
$N = 100, \quad \langle k \rangle \approx 20$	0.1631 ± 0.0010	0.1604 ± 0.0012	0.4512 ± 0.0027
$N = 1000, \quad \langle k \rangle \approx 10$	0.2798 ± 0.0006	0.2831 ± 0.0009	0.8394 ± 0.0006
$N = 1000, \quad \langle k \rangle \approx 20$	0.2008 ± 0.0005	0.1900 ± 0.0005	0.7717 ± 0.0008
$N = 10000, \langle k \rangle \approx 10$	0.2594 ± 0.0003	0.2769 ± 0.0016	0.9413 ± 0.0001
$N = 10000, \langle k \rangle \approx 20$	0.1810 ± 0.0003	0.1976 ± 0.0003	0.9140 ± 0.0002

Based on Figs. 2.3.2 and 2.3.3, it can be claimed that partitions of relatively high modularity can be found in PSO networks for a large region of the model's parameter space both with the modularity-maximizing Louvain algorithm and the asynchronous label propagation algorithm that does not build on the modularity at all but simulates the diffusion of community labels along the links. Nevertheless, when considering the high strength of the detected community structures indicated by the large modularities, it is important to bear in mind that quite high modularity values can be achieved even on Erdős–Rényi (ER) random graphs [30] and Barabási–Albert (BA) random graphs [17] under certain circumstances [31, 32]. However, as it is shown in Table. 2.3.1, in the case of these types of non-modular random networks, when

setting their size and link density equal to that of the PSO networks examined in Fig. 2.3.3, the modularity yielded even by the Louvain algorithm that aims at the maximization of the modularity lags considerably behind the values attained on PSO networks.

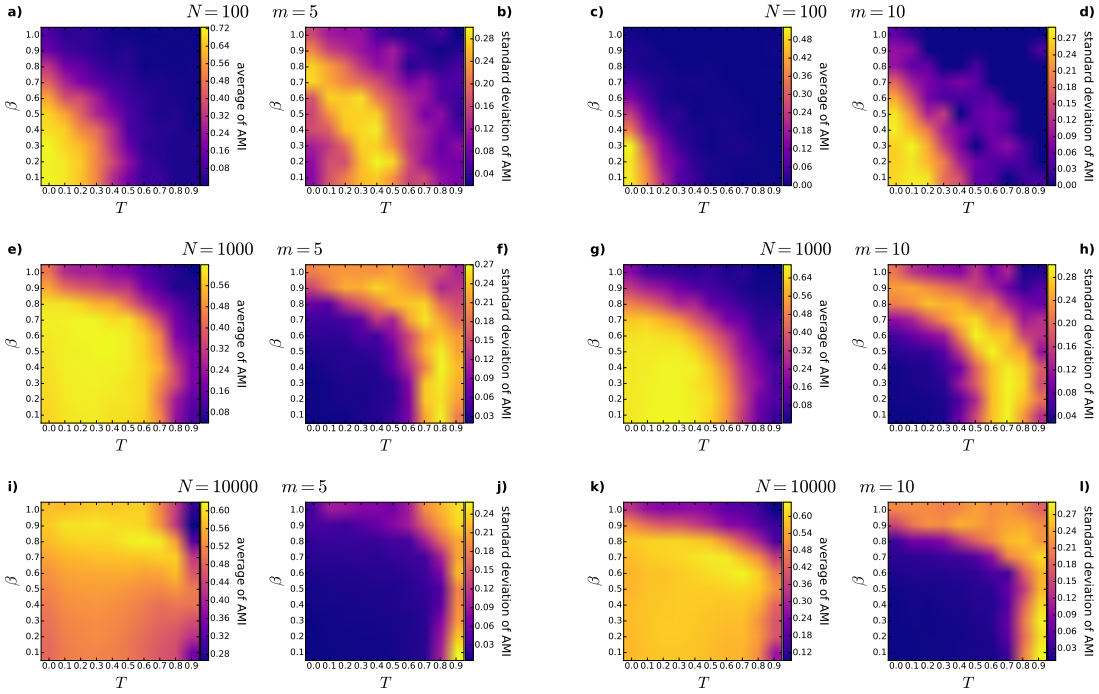

FIGURE 2.3.4: **Similarity between the partitions of PSO networks found by the asynchronous label propagation and the Louvain algorithms.** Each pair of subplots depicts for a given network size N and average degree $\langle k \rangle \approx 2m$ the average and the standard deviation of the adjusted mutual information AMI of the detected two community structures over 100 PSO networks as a function of the popularity fading parameter β and the temperature T. Each network was generated on the hyperbolic plane of curvature $K = -1$, using $\zeta = 1$. The figure was taken from Ref. [T2].

Next, Fig. 2.3.4 compares the community structures extracted from the examined PSO networks using the asynchronous label propagation and the Louvain algorithms by depicting the partitions' adjusted mutual information AMI $\in [0,1]$ [33, 34], higher values of which correspond to stronger consistency between the given network partitions. According to its definition, AMI can be calculated for two sets A and B of C_A and C_B number of communities over the same network as

$$\mathrm{AMI}(A,B) = \frac{\mathrm{MI}(A,B) - \mathbb{E}(\mathrm{MI}(A_{\mathrm{rand}}, B_{\mathrm{rand}}))}{\max\{H(A), H(B)\} - \mathbb{E}(\max\{H(A_{\mathrm{rand}}), H(B_{\mathrm{rand}})\})}, \tag{2.3.2}$$

where A_{rand} and B_{rand} indicate random partitions of the given network of N number of nodes,

$$\mathrm{MI}(A,B) = -\sum_{a=1}^{C_A}\sum_{b=1}^{C_B}\frac{N_{ab}}{N}\ln\left(\frac{N_{ab}N}{N_a N_b}\right) \tag{2.3.3}$$

is the mutual information between the examined partitions A and B, while

$$H(A) = -\sum_{a=1}^{C_A}\frac{N_a}{N}\ln\left(\frac{N_a}{N}\right) \quad \text{and} \quad H(B) = -\sum_{b=1}^{C_B}\frac{N_b}{N}\ln\left(\frac{N_b}{N}\right) \tag{2.3.4}$$

are equal to the entropies associated with the partitions A and B, with N_{ab} corresponding to the

number of nodes contained by community a of size N_a in partition A that belongs to community b of size N_b in partition B. In addition to the achieved high modularities presented in Figs. 2.3.2 and 2.3.3, Fig. 2.3.4 shows another sign for the actual relevance of the PSO networks' community structure, namely that in the range of relatively strong partitions (i.e, high modularities), the modular structures detected by the applied two methods along rather distinct principles exhibit significant consistency with each other, as indicated by AMI values that are way higher than the minimum value of 0 that could be expected in the case of comparing random partitions and often even close to the maximal AMI of 1 that indicates identical partitions.

Regarding Figs. 2.3.2–2.3.4, a naturally arising question is what are the conditions of the emergence of a community structure in a PSO network. Basically, what manifests itself through the formation of angularly separated communities even in the case of uniform angular node distribution is a fundamental characteristic of the hyperbolic geometry mentioned in Sect. 1.1 (see Fig. 1.1.1), namely that the angular range of hyperbolically close positions tightens towards the larger radial coordinates. Due to this, while minimizing the hyperbolic length of their links, new nodes in the PSO model (appearing at the periphery of the current network snapshot) tend to connect rather radially than tangentially. Thus, if the localization of the node-node connections is strong enough (meaning that the connections are strongly determined by the hyperbolic distances) and there is a large enough separation between the inner nodes, then the outer nodes can commit themselves towards the previously appeared nodes of a rather limited angular range and form the majority of their connections within a given angular sector that eventually will host a community in the network.

The localization of the links of a hyperbolic network can be enhanced by using a lower temperature T, thereby making the cutoff in the connection probability sharp as a function of the hyperbolic distance (and increasing the average clustering coefficient of the network), and by setting the number of nodes N to a large value compared to the number of links m arising at each time step, thus ensuring that the newly appearing nodes can collect all the required links from hyperbolically close nodes and they are not forced to connect to farther nodes to create all the m number of links. In the meantime, the emergence of well-separated attractive centers for the different angular regions can be facilitated by using a lower popularity fading parameter β, thereby emphasizing popularity fading, i.e. accelerating the inner nodes' outward drift during the network growth, gaining more considerable distance between them and avoiding the emergence of such hubs that are outstandingly attractive for nodes at any angular regions[7].

All things considered, according to Figs. 2.3.2–2.3.4, a particularly wide range of the PSO model parameters seems to support the natural formation of relatively strong community structures that can be detected based on fundamentally different approaches in a consistent way. Remarkably, the region of the parameter space where communities emerge from the PSO model includes the region of scale-free networks with high clustering, meaning that the PSO model is capable of generating networks that have the small-world property, a scale-free degree distribution, a high average clustering coefficient and a strongly modular structure at the same time.

[7]To generate communities in scale-free hyperbolic networks of a degree decay exponent γ close to 2, it might be more expedient to choose such a model [14–16] in which the community formation is promoted by making the angular distribution of the network nodes nonuniform. Nevertheless, as it was shown in Refs. [35] and [S1], as the number of network nodes N goes to ∞, the modularity of angularly uniform hyperbolic networks of small T temperatures approaches 1 even at $\gamma \approx 2$.

2.4 Extension of the PSO model to any number of dimensions: the dPSO model

Euclidean network embedding techniques – which represent the topological relations via Euclidean geometric measures – usually place the network nodes in higher-dimensional spaces [36–40] instead of the Euclidean plane that is often considered to be too simple to be capable of grasping the highly complex structure of real networks. Recently, it has also been revealed that even when using hyperbolic space, higher-dimensional embeddings can outperform lower-dimensional ones e.g. in link prediction and graph reconstruction tasks in author collaboration networks [8], or in the separation between different communities of a network [41]. In parallel with the demand for leveraging the potential lying behind hyperbolic spaces of higher number of dimensions, the motivation for gaining a better understanding of hyperbolic random graphs by studying their behavior even in higher-dimensional spaces has also arisen. First, a d-dimensional extension of the static random hyperbolic graph (RHG) model [2] has been introduced [13, 42]. At the same time, I began to develop a generalized version for the two-dimensional popularity-similarity optimization (PSO) model [3] described in Sect. 2.1, treating the number of dimensions of the hyperbolic space in which the networks are generated as an adjustable model parameter.

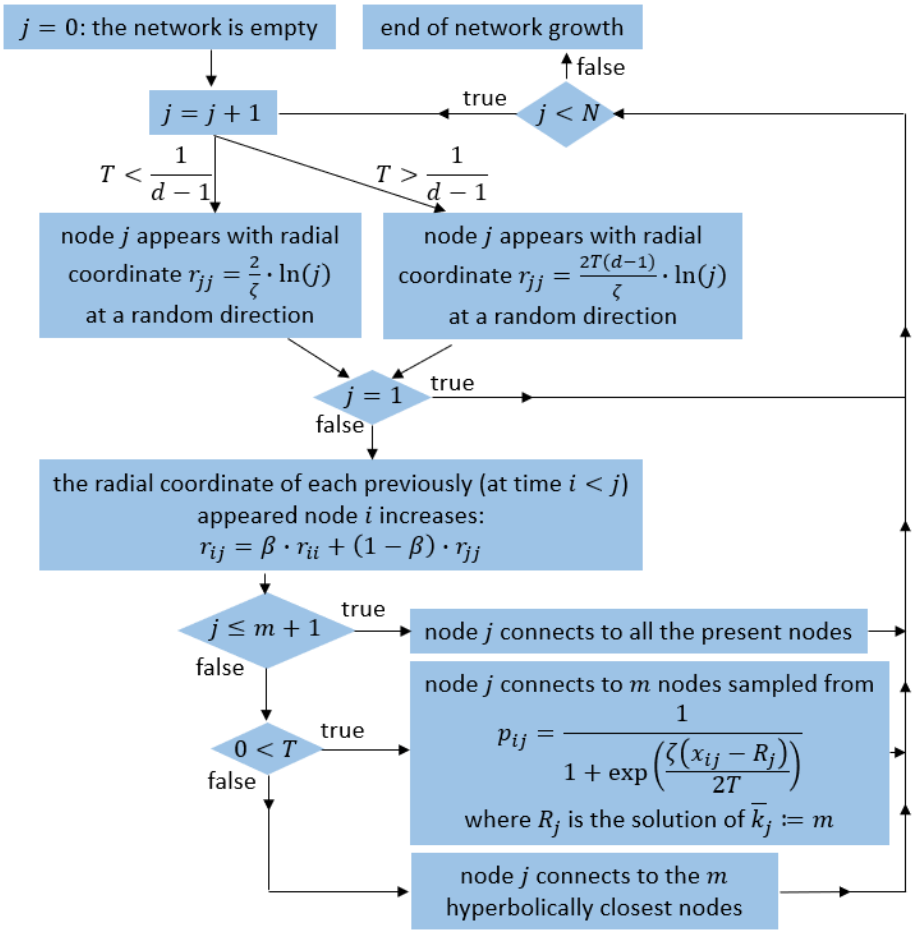

FIGURE 2.4.1: **Flowchart of the d-dimensional extension of the popularity-similarity optimization model.** Compared to the original, two-dimensional algorithm described in Sect. 2.1, here also the number of dimensions $2 \leq d \in \mathbb{Z}^+$ of the hyperbolic space is treated as a tuneable model parameter, besides the curvature $K = -\zeta^2 \in \mathbb{R}^-$ of the hyperbolic space, the final number $N \in \mathbb{Z}^+$ of nodes, the number $m \in \mathbb{Z}^+$ of new connections per time step, the popularity fading parameter $\beta \in (0, 1]$ and the temperature $0 \leq T$, $T \neq \frac{1}{d-1}$.

The algorithm of the d-dimensional PSO model, i.e. the dPSO model [T3] is presented by Fig. 2.4.1. Note that the here-indicated choice of the multiplying factor in the formula of the radial node coordinate r_{jj} in the native representation of the hyperbolic space is not the only reasonable option. As it is supported by Fig. 2.4.2 and the analytical calculations presented in the Supplementary Information of Ref. [T3], the given formulas

$$r_{jj} = \begin{cases} \frac{2}{\zeta} \cdot \ln j & \text{if } 0 \leq T < \frac{1}{d-1}, \\ \frac{2T(d-1)}{\zeta} \cdot \ln j & \text{if } \frac{1}{d-1} < T \end{cases} \tag{2.4.1}$$

yield a d-dependent degree decay exponent

$$\gamma = 1 + \frac{1}{(d-1) \cdot \beta} \tag{2.4.2}$$

both in the cold $\left(T < \frac{1}{d-1}\right)$ and the hot regime $\left(\frac{1}{d-1} < T\right)$, meaning that in this case the possible smallest degree decay exponent $\gamma_{\min} = 1 + \frac{1}{d-1}$ (obtained at $\beta = 1$) decreases, i.e. the achievable highest degree increases towards the higher-dimensional spaces. However, using e.g.

$$r_{jj} = \begin{cases} \frac{2}{\zeta(d-1)} \cdot \ln j & \text{if } 0 \leq T < \frac{1}{d-1}, \\ \frac{2T}{\zeta} \cdot \ln j & \text{if } \frac{1}{d-1} < T \end{cases} \tag{2.4.3}$$

instead, we also recover for $d = 2$ the well-known radial coordinate formulas

$$r_{jj} = \begin{cases} \frac{2}{\zeta} \cdot \ln j & \text{if } 0 \leq T < \frac{1}{d-1}, \\ \frac{2T}{\zeta} \cdot \ln j & \text{if } \frac{1}{d-1} < T \end{cases} \tag{2.4.4}$$

of the original PSO model [3]. Nevertheless, this latter choice makes the degree decay exponent γ independent of the number of dimensions $\left(\gamma = 1 + \frac{1}{\beta}\right)$, thereby excluding the possibility of decreasing γ below 2. As the two model variants given by Eqs. (2.4.1) and (2.4.3) are equivalent in the $2 \leq \gamma$ regime (see Sect. S1.4 of the Supplementary Information of Ref. [T3]), in the algorithm of the dPSO model I opted for the former option, i.e. Eq. (2.4.1), which yields a broader range of achievable degree decay exponents[8].

But why does the degree decay exponent γ become in Eq. (2.4.2) a decreasing function of the number of dimensions of the hyperbolic space at a given popularity fading parameter β? As it is explained in Ref. [T3], this behavior can be understood in the cold regime $T < 1/(d-1)$ (where, according to Eq. (2.4.1), the initial radial coordinates do not depend on d) considering that the uniform distribution of the same number of network nodes on the surface of a higher-dimensional Euclidean ball representing the hyperbolic space yields larger angular distances between the nearest neighbors since the total solid angle belonging to the d-dimensional Euclidean ball grows as the number of dimensions increases. Due to this, even at a fixed value of the popularity fading parameter β (that controls the differences between the radial coordinates of nodes of different appearance times), the increase in the number of dimensions results in an

[8] Note that increasing the number of dimensions d of the hyperbolic space is not necessarily required for achieving extremely heavy-tailed degree distributions with $\gamma < 2$ decay exponents through popularity-similarity optimization. Introducing a new variant of the PSO model named fPSO, I showed in Sect. S1.4 of the Supplementary Information of Ref. [T3] that by treating the multiplying factor in the formula of the initial radial node coordinates as a tunable model parameter – i.e., using the general formula of $r_{jj} = f \cdot \ln j$ –, the degree decay exponent γ can be decreased below 2 even when generating networks on the hyperbolic plane.

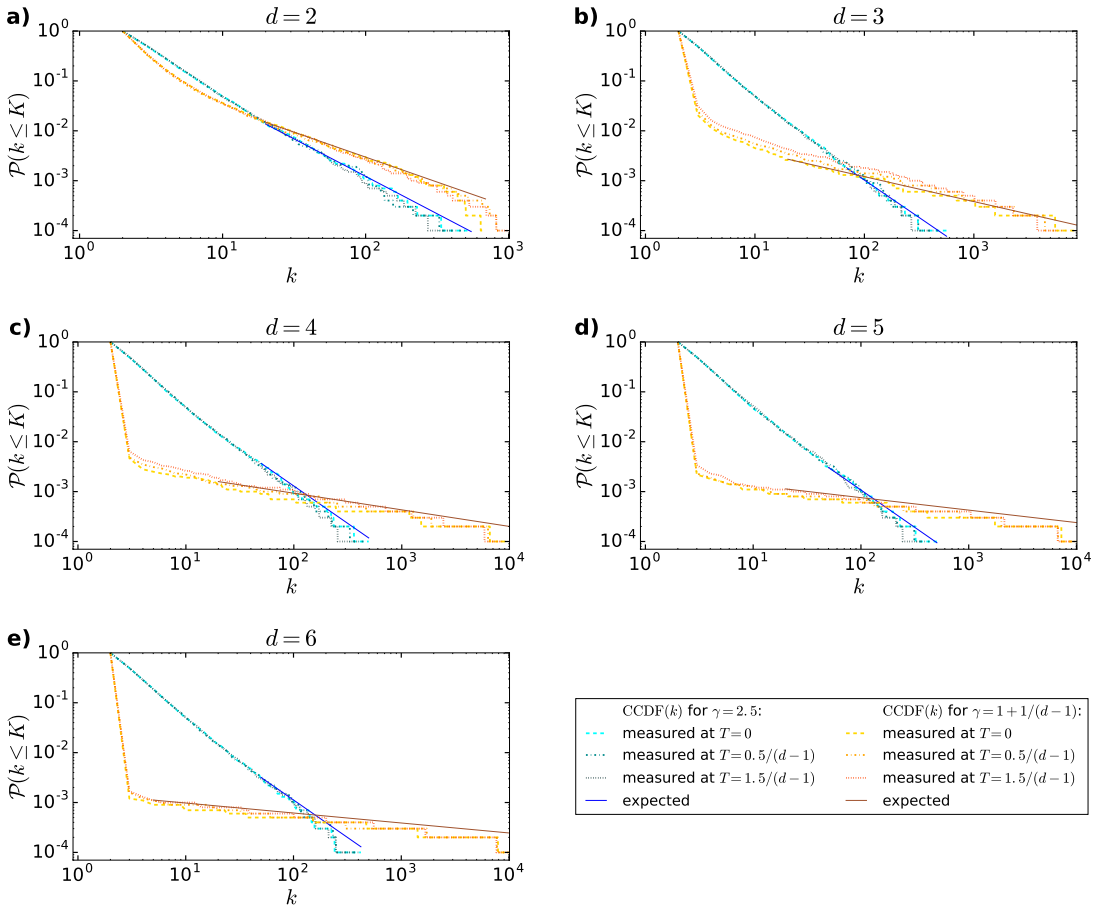

FIGURE 2.4.2: **Degree distribution of dPSO networks.** Each non-solid curve shows the complementary cumulative distribution function (CCDF) of the node degrees for a given network generated by the dPSO model, following closely for all the tested settings of the temperature T (indicated in the legend) the solid curve of $\mathcal{P}(k \leq K) \sim k^{-(\gamma-1)}$ that was expected based on the analytical study of the model. The expected degree decay exponent was set to either $\gamma = 2.5$ (blue curves), or the achievable smallest value $\gamma = 1 + 1/(d-1)$ (orange curves) that decreases as the number of dimensions d increases from panel a) to e). All the here-examined networks were built up from $N = 10000$ number of nodes in a hyperbolic space of curvature $K = -1$, creating $m = 2$ number of connections per time step. The figure was taken from Ref. [T3].

increased preference of the newly coming nodes towards the innermost nodes instead of connecting to any nodes at larger radii, even their closest angular neighbors. Therefore, the same popularity fading parameter β leads to a larger maximum degree, and thus, a smaller degree decay exponent γ for larger values of d. On the other hand, if the popularity fading parameter β is decreased (thus, the process of popularity fading is enhanced and the inner nodes are shifted more outwards) simultaneously with the growth of the number of dimensions, then the decline in the attractiveness of the angular neighbors induced by the increase in d can be compensated by the reduction in the distinguished attractivity of the inner nodes, i.e. the limitation of the angular range in which they are favored and the degree decay exponent γ can be kept at a fixed value. Note that in the hot regime given by $T > 1/(d-1)$, the probability of long-distance connections becomes so large that the above-described decrease in the attractiveness of the angular neighbors would not occur with the increase in d at fixed values of β and the initial radial coordinates. Therefore, to obtain the same d-dependent formula for the degree decay exponent γ as in the cold regime $T < 1/(d-1)$, in the hot regime the formula of the initial radial coordinates was made an increasing function of the number of dimensions d in Eq. (2.4.1), thereby

intensifying the distinguished attractivity of the inner nodes towards the higher-dimensional spaces and increasing the achieved highest node degree.

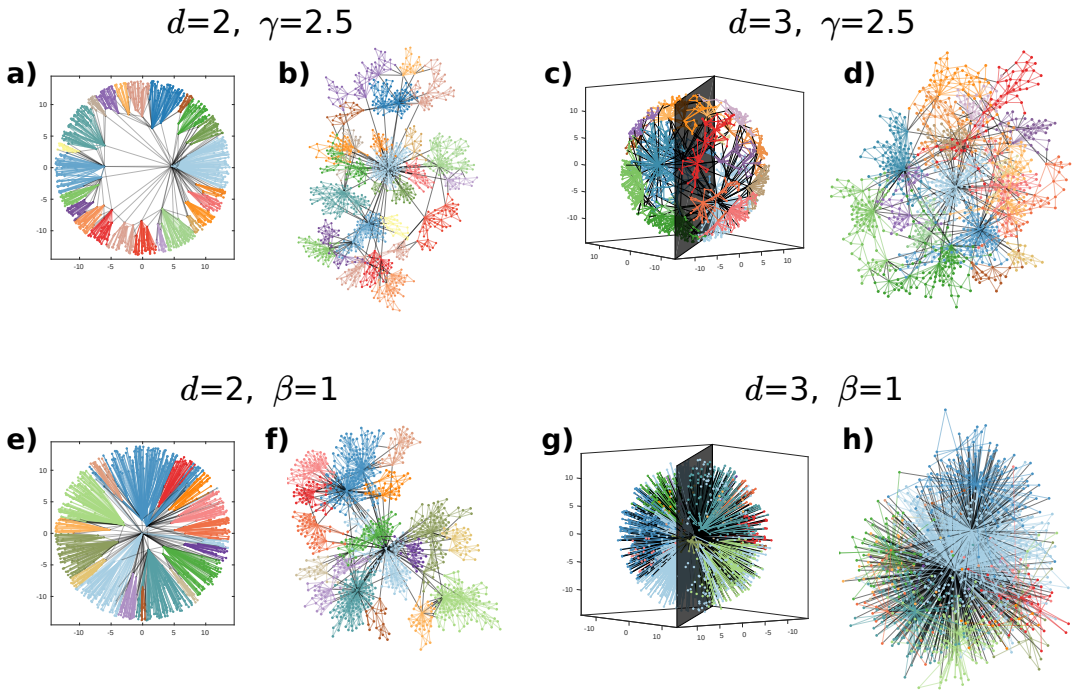

FIGURE 2.4.3: **Layouts of networks generated by the dPSO model in 2- and 3-dimensional hyperbolic spaces.** The coloring of the nodes and the links indicates communities found by the Louvain algorithm, with grey lines denoting the inter-community links. Each pair of panels (having a shared title) shows a dPSO network both as it was created in the native representation of the hyperbolic space (on the left of each panel pair) and also according to a standard layout, namely the Prefuse force-directed layout (on the right of each panel pair), demonstrating that the detected communities are clearly outlined even in such node arrangements that do not build on the hyperbolic origin of the networks. The top row of panels presents two dPSO networks having the same degree decay exponent $\gamma = 2.5$: panels a) and b) correspond to a network generated on the 2-dimensional hyperbolic plane, setting the popularity fading parameter β to $2/3$, while panels c) and d) refer to a dPSO graph created in the 3-dimensional hyperbolic space, setting the popularity fading parameter β to $1/3$. The bottom row of panels depicts two dPSO networks of the same popularity fading parameter $\beta = 1$, which yields the smallest degree decay exponent that is achievable at a given value of the number of dimensions d, namely $\gamma = 2.0$ in the 2-dimensional case shown in panels e) and f), and $\gamma = 1.5$ in the 3-dimensional case presented in panels g) and h). All the depicted networks were generated by setting the curvature K of the hyperbolic space to -1, the number of nodes N to 1000, the expected average degree $2m$ to 4 and the temperature T to 0. The average clustering coefficient $\bar{c}$ of the displayed networks and the modularity Q of their colored partitions are the following: $\bar{c} = 0.729$ and $Q = 0.882$ for panels a) and b); $\bar{c} = 0.644$ and $Q = 0.845$ for panels c) and d); $\bar{c} = 0.788$ and $Q = 0.829$ for panels e) and f); and lastly, $\bar{c} = 0.964$ and $Q = 0.354$ for panels g) and h). The figure was taken from Ref. [T3].

To get an impression about what kind of networks are formed in the dPSO model, Fig. 2.4.3 shows layouts of 2- and 3-dimensional dPSO networks. As expected from the above discussion, panels e)–h) show an increase in the size of the largest hubs towards the higher-dimensional space despite using a fixed popularity fading parameter β. In addition, the automatic emergence of angularly separated communities among angularly homogeneously distributed nodes (described in detail for the original, two-dimensional PSO model in Sect. 2.3) can be observed both in the two- and the three-dimensional cases. An important difference is, however, that when adding new dimensions to the attribute space of the nodes (i.e., when introducing new angular coordinates for each node), the community boundaries become more complex, since

e.g. while in the two-dimensional case communities of nodes held together by their common preference toward the same attractive centers are arranged along a simple circle circumference, in a three-dimensional network the communities are aligned over a sphere surface.

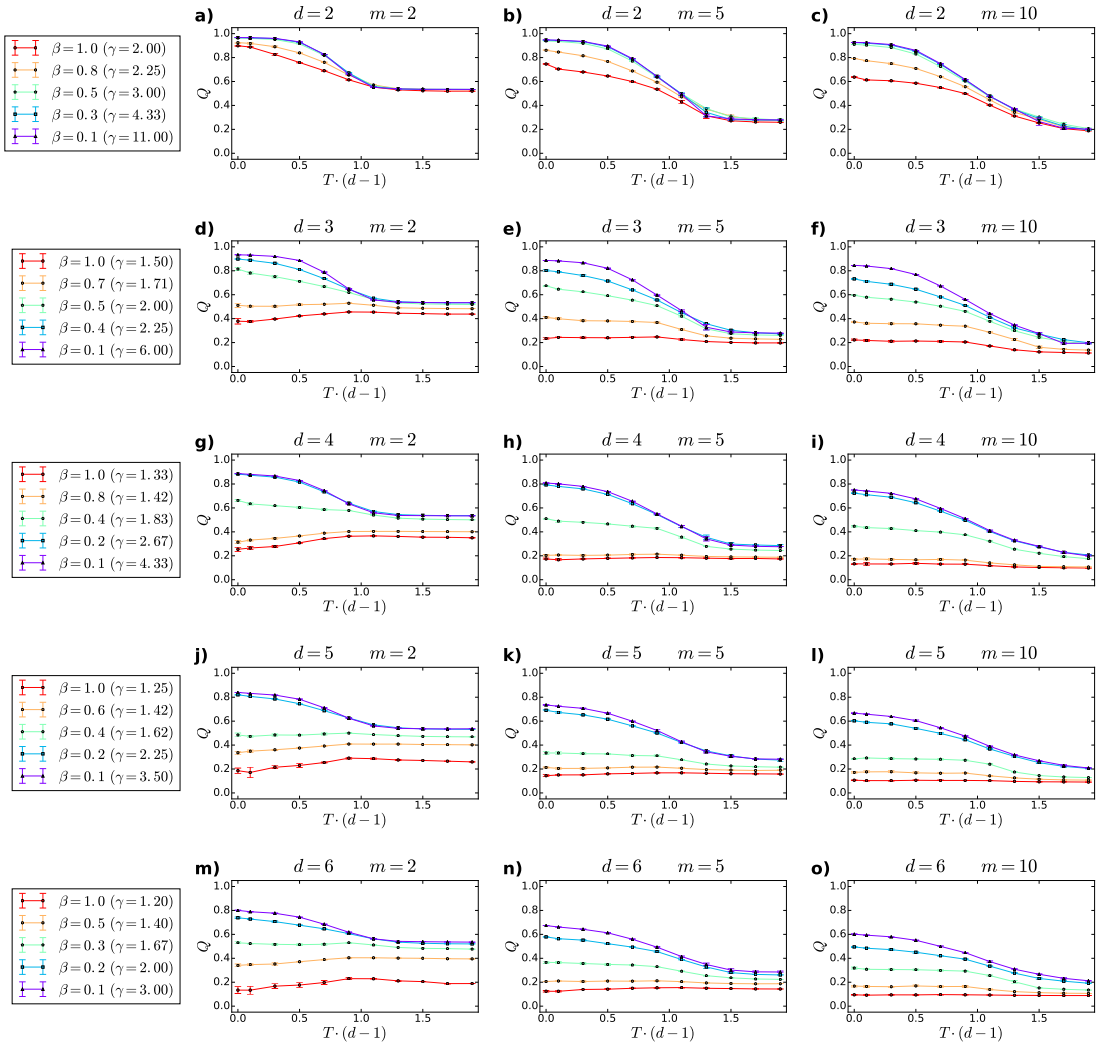

FIGURE 2.4.4: **The parameter-dependence of the modularity Q achieved in dPSO networks.** The communities were detected with the asynchronous label propagation, the Louvain and the Infomap algorithm too, and each depicted data point corresponds to the obtained highest modularity at the given model parameters. Each panel shows Q as a function of the rescaled temperature $T \cdot (d-1)$. The number of dimensions d is constant across the rows of panels, whereas the expected average degree $2m$ is constant across each panel column, as indicated by the panel titles. The different curves in a given subplot correspond to different values of the popularity fading parameter β (yielding different degree decay exponents γ), listed for each dimension in the leftmost panel of the corresponding row. All the networks consisted of $N = 10000$ number of nodes, each of them were generated in a hyperbolic space of curvature $K = -\zeta^2 = -1$, and I searched for communities once with all three community detection methods on every network. The displayed data points were obtained by averaging over 5 dPSO networks generated independently with a given set of model parameters, and the error bars indicate the standard deviations among the 5 networks. The figure was taken from Ref. [T3].

To study how the emergent community structure of PSO networks changes towards the higher-dimensional cases, I created Fig. 2.4.4 by generating dPSO networks of different dimensions according to the algorithm given in Fig. 2.4.1, searching for communities in them using the

asynchronous label propagation [28], the Infomap [29] and the Louvain [26] methods, and tracing the changes in the achieved highest modularity as a function of the model parameters. Just like in the case of Figs. 2.3.2 and 2.3.3, the modularity Q was calculated in these measurements according to Eq. (2.3.1). Based on Fig. 2.4.4, the conditions for a community structure having high modularity are the same for any value of d as those described for the two-dimensional case in Sect. 2.3. Namely, strong communities can be formed when the link formation is localized (preventing the direct connection between nodes that are far away from each other) and even the inner nodes' angular orientation is clear, enabling them to serve as the attractive centers of well-defined angular sectors and not the whole angular range of nodes. Consequently, the highest modularity values were measured at any d when the network temperature T was small, the N/m ratio was large and, at the same time, the largest occurring degree was not too high (i.e., the degree decay exponent γ was not too small).

It is important to bear in mind that the effect of the same values of the temperature T and the popularity fading parameter β strongly differs at different values of the number of dimensions d: in higher-dimensional spaces, using the same setting of T results in the emergence of connections in more random directions (weaker localization), while a given β yields smaller γ at higher d according to Eq. (2.4.2). The impact of the number of dimensions on the degree decay exponent has already been expounded above, whereas the randomizing influence of the increase in d on link formation can be explained as follows: at a larger number of dimensions – i.e., when the number of independent angular coordinates characterizing the attributes of each node is larger –, the positions indicating node attributes that are similar to that of the newly appearing node are given by a larger number of coordinate combinations, and thus, even when a small temperature value makes the new nodes connect mostly to similar nodes, the connection pattern becomes more entangled compared to a lower-dimensional case. Consequently, enhanced by the effect of the larger value of d, in a higher-dimensional space the link-randomizing influence of the increase in the temperature becomes considerable sooner, at smaller values of T. Namely, in the d-dimensional extension of the two-dimensional PSO model, the mainly applied range of $T \in [0, 1)$ corresponds to the range of $T \in [0, 1/(d-1))$, suggesting that higher values of d tighten the range of the temperature values that can be considered to be actually small, i.e. where the links are well localized.

To highlight the impact of changing the number of dimensions d on the structural properties of dPSO networks, Figs. 2.4.5 and 2.4.6 compare the modularity Q and the average clustering coefficient[9] $\bar{c}$ of dPSO networks of different d values but equal degree decay exponents (obtained with different settings of the popularity fading parameter β), using T values that lie at the same point of the temperature scale renormalized in correspondence with the given setting of the number of dimensions. Based on these figures, at small temperatures – where the highest values of Q and $\bar{c}$ can be achieved – the increase in the number of dimensions of the applied hyperbolic space at a given γ results in weaker community structures on the mesoscopic scale and weaker clustering on the local scale, while at high temperatures d and β do not influence the properties of the dPSO network individually, just through the joint adjustment of the degree decay exponent γ. As described in Ref. [T3], this phenomenon can be understood based on the above-mentioned randomizing influence of the increase in d. At not-too-high temperatures, the newly appearing nodes connect mostly to those previously appeared nodes that are relatively similar to them, where high similarity means small angular distance. However, in higher-dimensional spaces, a larger number of coordinate combinations correspond to node attributes of the same similarity from the viewpoint of the newly appearing node. Because of this, any two nodes that can be considered to be similar to a third (the new) node are less likely at higher values of d to

[9] A more detailed numerical examination of the dependence of the average clustering coefficient of dPSO networks on the different model parameters is provided by the Clustering coefficient subsection of the Results section of Ref. [T3], presenting the same tendencies at each number of dimensions d as Fig. 2.1.7 shows for the two-dimensional case, namely that $\bar{c}$ is an increasing function of m and β, and reaches its maximum at $T = 0$.

have angular positions that are relatively close to each other as well. Accordingly, on the local scale, the growth of the number of dimensions reduces the proximity, and thus, the number of shared links of the nodes to which a new node connects at its appearance, thereby decreasing the number of triangles and the average clustering coefficient in the generated network. In the meantime, on the mesoscopic scale, the modularity measured at a given degree decay exponent γ also becomes a decreasing function of d at low temperatures. Nevertheless, at high temperatures where connections between rather distant nodes are also likely to occur, it does not matter anymore whether a given degree decay exponent was obtained in a higher-dimensional space using a smaller popularity fading parameter or in a lower-dimensional space with higher popularity fading parameter, indicating that the link-randomizing effect of the high temperatures can be so strong that the similar effect of the large number of dimensions becomes negligible compared to it.

To conclude, the investigation of the PSO model's d-dimensional extension revealed that not only the hyperbolic plane but hyperbolic spaces of a slightly higher number of dimensions can be excellently applied for generating networks that simultaneously reproduce the frequently mentioned structural features of real-world networks given by the small-world property, a scale-free degree distribution, a high average clustering coefficient and a strong community structure. However, to achieve high modularity and average clustering coefficient, the number of dimensions d of the hyperbolic space has to be restricted, as the strength of the community structure and the clustering in dPSO networks of a given degree decay exponent decreases towards higher values of d. Nonetheless, it is important to note that the community structure can be strengthened in higher-dimensional spaces by distributing the nodes in an inhomogeneous way instead of sampling the directions uniformly at random. Following the idea of the two-dimensional nonuniform popularity-similarity optimization (nPSO) model [16], I introduced in Sect. S4 of the Supplementary Information of Ref. [T3] a three-dimensional nPSO model that can concentrate the network nodes in dense, possibly separated patches and I demonstrated there that by lifting the restriction of the homogeneous angular node arrangement, besides gaining some control over the number and the size of communities, the modularity of the detectable community structures can be increased.

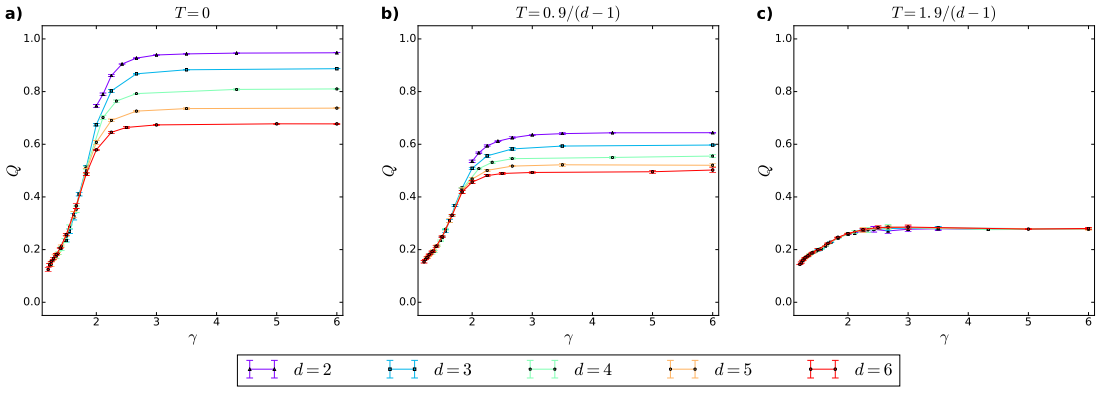

FIGURE 2.4.5: **The modularity Q achieved in d-dimensional PSO networks as a function of the degree decay exponent γ.** I searched for communities once with the asynchronous label propagation, the Louvain and also the Infomap algorithm on 5 dPSO networks of each set of model parameters and always plotted the obtained highest modularity averaged over the 5 networks, with the error bar indicating the standard deviation among the 5 networks. The panels refer to different values of the network temperature T, given in the title of the subplots. The curves of different colors correspond to different numbers of dimensions d, as listed in the legend below the panels. For a given γ and d, the popularity fading parameter was set to $\beta = \frac{1}{(d-1)\cdot(\gamma-1)}$. The curvature $K = -\zeta^2$ of the hyperbolic space was -1, the network size N was 10000 and the expected average degree $2m$ was 10 in every case. The figure was taken from Ref. [T3].

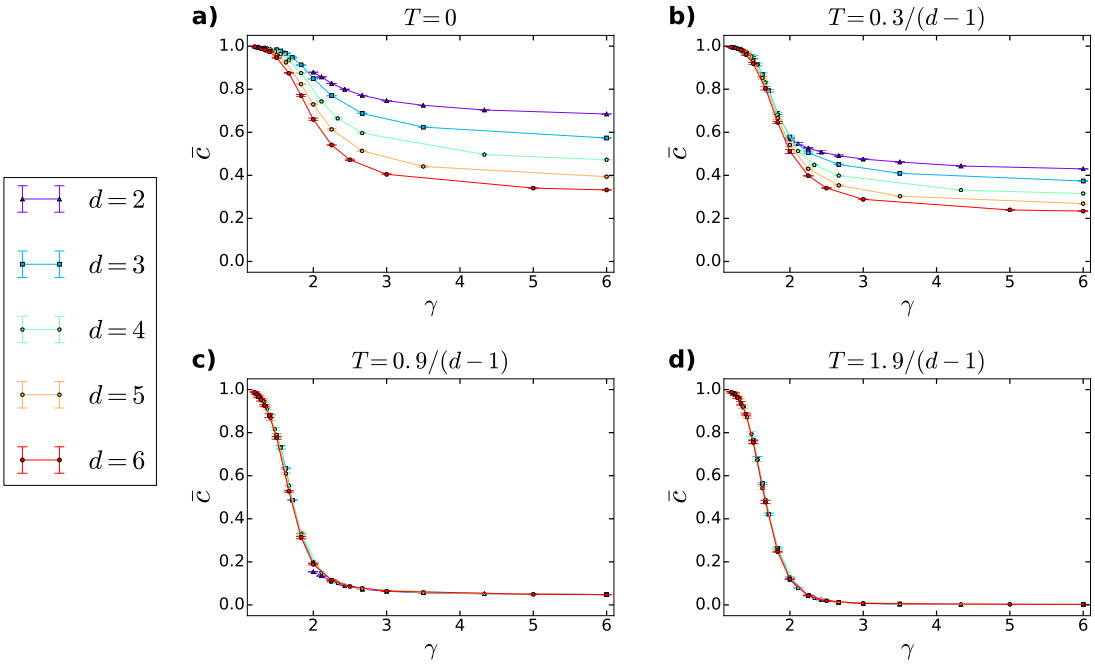

FIGURE 2.4.6: **Average clustering coefficient $\bar{c}$ measured in d-dimensional PSO networks as a function of the degree decay exponent γ.** The figure shows the average clustering coefficient averaged over 5 dPSO networks for each parameter setting, with the error bars indicating the standard deviations among the 5 networks. Each panel was created using a given value of the network temperature T specified in the panel title. The curves of different colors correspond to different values of the number of dimensions d of the hyperbolic space in which the networks were generated, as listed in the legend. The popularity fading parameter was always set to $\beta = \frac{1}{(d-1)\cdot(\gamma-1)}$. The size of the networks was $N = 10000$, the expected average degree was $2m = 10$, and each network was generated in a hyperbolic space of curvature $K = -1$. The figure was taken from Ref. [T3].

3 Embedding undirected networks in hyperbolic spaces

When embedding a network into a d-dimensional hyperbolic space, the aim is to assign a d-dimensional position vector to each network node in such a way that the topological relations of the network become articulated as relations between hyperbolic node-node distances. More precisely, nodes that are easy to reach from each other along the links (i.e., connected by short and/or numerous paths) are ideally placed hyperbolically close to each other, keeping in the meantime the distance between hard-to-reach nodes large. Since hyperbolic network models create graphs using a connection rule (smaller hyperbolic distances correspond to larger connection probabilities) that matches the embeddings' above view, it is a common practice in hyperbolic embedding algorithms to optimize the node positions in accordance with a given hyperbolic network model, trying to realize the node arrangement that would form with the highest probability when generating a network consisting of the same edges as the network to be embedded. On the other hand, the embedding task can be interpreted as a dimension reduction process, where the information contained by the adjacency matrix (which stores for each node its pairwise relationship with all the N nodes of the network) has to be mapped to a lower-dimensional representation (in which a position vector of d number of spatial coordinates is assigned for each node).

This chapter presents some well-known techniques in the hyperbolic embedding of connected (i.e., single-component), undirected networks, which contain between any two nodes a path of finite hop-length that can be traversed in both directions identically. Note that the hyperbolic embedding of directed networks is still in its infancy and is the subject of Chapter 4, while multi-component networks are usually not considered in the studies of the field of network embedding, since for arranging unconnected subgraphs in relation to each other, in the absence of any topological information about which components should be considered to be easier to reach from each other (e.g. via a later developing connection), one would need further information in addition to the edge list, such as node attributes.

Regarding the undirected networks, Sect. 3.1 describes the most popular dimension reduction techniques (namely Laplacian Eigenmaps (LE) [37] and Isomap (ISO) [36]) that have been shown [5, 6] to be able of producing such *angular* arrangement of the network nodes in the d-dimensional Euclidean space of curvature $K = 0$ that can also be used in the native representation of the d-dimensional hyperbolic space of curvature $K < 0$. Then, based on Ref. [4], Sect. 3.2 defines how the *radial* node arrangement that is the most probable in the native representation of the hyperbolic plane according to the popularity-similarity optimization (PSO) model [3] can be determined. Regarding this likelihood maximization technique, through some of my measurements that were published in Ref. [T1], Sect. 3.2.1 draws attention to a previously not considered degree of freedom in the choice of the radial coordinates, the exploitation of which can lead to an improvement in the embedding quality. Besides, Sect. 3.2.2 outlines how I assigned radial coordinates to the network nodes in the case of higher-dimensional hyperbolic embeddings in Ref. [T4] based on the dPSO model [T3] described in Sect. 2.4. Finally, Sect. 3.3 presents the hydra method [10] that embeds undirected, connected networks directly into the hyperbolic space, eliminating the dependence of the network's hyperbolic layout on the result of a preliminary Euclidean embedding method. The main innovation of hydra is that here the subject of dimension reduction is a matrix of expected Lorentz products (see Eq. (1.3.2)), which idea was also utilized in one of my directed embedding methods proposed in Ref. [T4].

3.1 Euclidean embeddings serving hyperbolic ones: angular coordinates based on dimension reduction

As it was shown in Refs. [5, 6], the angular coordinates of the network nodes yielded by Euclidean graph embedding methods[1] such as Laplacian Eigenmaps [37] (LE, detailed in Sect. 3.1.1) and Isomap [36] (ISO, detailed in Sect. 3.1.2) can also be used as the angular coordinates in a hyperbolic embedding of the network obtained in the native representation [2] of the hyperbolic space, where (as it is described in Sect. 1.1) the angular distances are equivalent with their usual, Euclidean counterpart. The rationale behind this approach is that by placing the connected nodes close to each other in the Euclidean space, these Euclidean embedding methods can inherently make the angular separation of nodes that are topologically close to each other small, just as it is expected from a hyperbolic embedding method considering the analogy between the small angular distance and the high similarity of node attributes in hyperbolic network models that works in the native representation of the hyperbolic space. Nevertheless, Euclidean embeddings that represent the topological relations of a network with Euclidean distances can not realize a hyperbolically probable radial node arrangement: while more popular nodes (that connect to more nodes) are placed in the hyperbolic space at smaller radial coordinates (that yield relatively small hyperbolic distances with a larger proportion of the other nodes), the minimization of the Euclidean distances lacks any obvious preference regarding the radial coordinates. Therefore, while mapping the small topological distances of a network to small Euclidean distances can produce such an angular node arrangement that is also suitable for a hyperbolic embedding (i.e., where small angular distances correspond to small topological distances), the radial coordinates obtained from Euclidean embedding methods like LE or ISO are usually completely disregarded and replaced by radial positions originated from a likelihood optimization regarding a given hyperbolic network model. Sect. 3.2 describes a likelihood-based calculation of hyperbolic radial coordinates, the result of which can be combined with the angular node coordinates generated e.g. by LE or ISO to obtain a hyperbolic embedding of a network, as it is exemplified in Sect. 4.3.4.

3.1.1 Laplacian Eigenmaps

Given a single-component, undirected, and possibly weighted network where a higher link weight w_{ij} indicates the higher strength or relevance of the $i-j$ connection, or in other words, a larger topological (or geometric) proximity of nodes i and j[2], Laplacian Eigenmaps (LE) assigns to every node j a position vector $\underline{u}_j$ of d number of Cartesian coordinates in the d-dimensional Euclidean space in such a way that a weighted sum of the squared distances between the connected node pairs $\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\cdot|\underline{u}_i-\underline{u}_j|^2$ is minimized while preventing the collapse of all nodes into a single point [7, 37]. Thus, LE focuses on preserving the local, neighborhood relations by putting the (strongly) connected nodes close to each other in the Euclidean space. This is achieved via the eigendecomposition of the network's Laplacian matrix[3], discarding the first eigenvector (as it belongs to an eigenvalue of 0) in the increasing order of the eigenvalues and using the next d number of eigenvectors as the coordinate vectors, each of which lists the coordinate of a given dimension for all the N number of nodes.

[1] Note that both LE and ISO were originally developed for finding lower-dimensional representations for data points of a higher-dimensional space, and their application for the embedding of networks relies on the principle that the given graph can be considered as a nearest neighbor graph constructed from a high-dimensional data set.

[2] If a distance-like link weight x_{ij} is given in a network – the higher values of which correspond to weaker connections or smaller topological proximities –, then usually [6, 7, 37] the proximity-like link weight w_{ij} applied in LE is calculated as $w_{ij}=e^{-\left(x_{ij}/\langle x_{ij}\rangle\right)^2}$ with $\langle x_{ij}\rangle$ denoting the average weight of the connections.

[3] The graph Laplacian is formed by subtracting the network's adjacency matrix from the diagonal matrix of node degrees or, in the case of weighted networks, node strengths.

3.1.2 Isomap

Using Isomap (ISO), one can assign in the d-dimensional Euclidean space a Cartesian position vector of length d to each node of a connected, undirected, and possibly weighted network, where a higher value of the link weight w_{ij} – contrary to the usual custom in network theory – corresponds to a weaker $i-j$ connection, or a larger topological (or geometric) distance between nodes i and j [6, 36]. Rather than trying to grasp specifically the individual local connections (like LE), ISO aims at preserving the global network topology directly by searching for such a node arrangement in which the pairwise Euclidean distances between the nodes are as close to the topological distances – given by the shortest path lengths (SPLs) – measured along the network as it is possible in a d-dimensional representation. To do so, ISO first converts the matrix $\boldsymbol{D}$ of expected Euclidean distances (i.e., the matrix of SPLs) to the matrix $\boldsymbol{I}$ of the corresponding expected Euclidean inner products. Setting the position of the center of mass of the network nodes to the origin, the distance-inner product conversion (i.e., the so-called centering step [6]) can be performed using the formula $\boldsymbol{I}(\boldsymbol{D}) = -\boldsymbol{H}\cdot\boldsymbol{R}(\boldsymbol{D})\cdot\boldsymbol{H}/2$, where $\boldsymbol{R}$ is the matrix of squared distances ($R_{ij} = D_{ij}^2$) and the so-called centering matrix $\boldsymbol{H}$ is defined for a network of N number of nodes using the Kronecker delta δ_{ij} as $H_{ij} = \delta_{ij} - 1/N$. Having the matrix $\boldsymbol{I}$, the task becomes to find the d-dimensional node position vectors that closely reproduce the expected inner products (and hereby also the expected Euclidean distances given by the SPLs), i.e. the rows of the coordinate matrix $\boldsymbol{X}$ that fulfills $\boldsymbol{I} = \boldsymbol{X}\cdot\boldsymbol{X}^{\mathrm{T}}$ as much as it is viable using the given number of dimensions. These position vectors are calculated using the singular value decomposition (SVD) $\boldsymbol{I} = \boldsymbol{U}\cdot\boldsymbol{\Sigma}\cdot\boldsymbol{V}^{\mathrm{T}}$. Namely, after discarding all but the first (largest) d number of singular values (thus reducing the size of the diagonal matrix $\boldsymbol{\Sigma}$ to $d\times d$) and removing all the singular vectors (the columns of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$) that belong to the discarded singular values, the matrix $\boldsymbol{X}$ of node coordinates can be written as $\boldsymbol{X} = \boldsymbol{U}\cdot\sqrt{\boldsymbol{\Sigma}}$ or, since for undirected networks $\boldsymbol{U} = \boldsymbol{V}$, as $\boldsymbol{X} = \sqrt{\boldsymbol{\Sigma}}\cdot\boldsymbol{V}^{\mathrm{T}}$, where each row of $\boldsymbol{X}$ contains the d number of Cartesian coordinates of the position vector of a given node.

3.2 Hyperbolic radial coordinates in correspondence with the PSO model

A straightforward way to embed a network in the hyperbolic plane is to find out which node arrangement would have yielded the observed edge list presuming a specific hyperbolic network model to be the generator of the given graph. Following this idea, the HyperMap method [4] places the network nodes in the hyperbolic plane by replaying the growth of the given network in the way that has the highest probability in the popularity-similarity optimization (PSO) model [3] – or, more precisely, in the E-PSO model [4] described in Sect. 2.2. According to Ref. [4], when searching for the most probable node coordinates for generating a given edge list with the assumption of a given set of E-PSO model parameters, first the maximization of the corresponding likelihood function can be performed analytically with respect to the radial coordinates (independently from the optimization of the angular positions), and then, introducing the nodes one by one in the increasing order of the obtained radial coordinates, the angular coordinate that is the most appropriate for the current network snapshot can be determined for each node numerically.

The E-PSO-based maximum likelihood estimate of the network nodes' hyperbolic radial coordinate has been used e.g. in Refs. [5, 6] to create hyperbolic embeddings from Euclidean patterns obtained with dimension reduction. These methods keep the angular node arrangement unaltered but replace the radial coordinates with those having the highest probability according to the E-PSO model [4]. Based on Ref. [4], to arrange the network nodes radially in the native representation of the hyperbolic plane of curvature $K = -\zeta^2$ (described in Sect. 1.1) achieving

the highest accordance with the E-PSO model of $L \geq 0$[4], one has to sort the N number of nodes in the decreasing order of the node degrees and assign to the jth node of this order the radial coordinate

$$r_{jN} = \beta \cdot \frac{2}{\zeta} \cdot \ln j + (1-\beta) \cdot \frac{2}{\zeta} \cdot \ln N, \tag{3.2.1}$$

where β is the estimation of the popularity fading parameter using which the network could have been generated by the PSO model. Since the degree decay exponent γ is expected to depend on the popularity fading parameter β as $\gamma = 1 + 1/\beta$ [3], the estimation of the popularity fading parameter can be calculated as $\beta = 1/(\gamma - 1)$ after determining the degree decay exponent γ by fitting a power law $\sim k^{-(\gamma-1)}$ to the complementary cumulative distribution function (CCDF$(k) = P(k \leq K)$) of the node degrees k. Such a fitting can be performed e.g. using the method described in Ref. [19], as it is exemplified by Fig. 2.1.5.

The following two subsections present my results regarding the PSO-based hyperbolic radial coordinates. Although in the usual practice the ties in the order of the node degrees are just broken randomly, Sect. 3.2.1 demonstrates that trying out several different radial orders between nodes having the same degree may be beneficial for achieving higher embedding qualities [T1]. Then, Sect. 3.2.2 describes how the above-described two-dimensional technique can be transposed to higher-dimensional hyperbolic spaces following the dPSO model [T3].

3.2.1 Unexploited freedom in the choice of the radial order when optimizing with regard to the PSO model

Following Ref. [T1], this section shows that changing the radial order between nodes of equal degree in PSO-based hyperbolic embeddings can affect the embedding quality. As an example, I embedded real networks multiple times using the HyperMap [4] and the non-centered minimum curvilinear embedding (ncMCE) [6] methods, breaking the ties in the order of node degrees always randomly. Both methods produce the radial node arrangement in accordance with the E-PSO model [4]; however, while HyperMap optimizes not only the radial coordinates but also the angular arrangement of the nodes with respect to the E-PSO model (making the angular coordinates dependent on the actual radial node order), ncMCE determines the angular coordinates independently from the radial ones, based on the singular value decomposition of a matrix of topological distances[5]. Despite the fundamental differences between HyperMap and ncMCE, permutating the radial order of the equally popular nodes seems to be beneficial for both embedding methods according to the logarithmic loss LL [4] that measures how well an embedding agrees with the E-PSO model, and also a model-independent quality measure given by the greedy routing score (GR-score) [6].

The logarithmic loss is calculated as $LL = -\ln \mathcal{L}(\boldsymbol{A} \mid \{r_{jN}, \theta_j\}, \zeta, N, m, L, \beta, T)$ from the likelihood of generating the observed adjacency matrix $\boldsymbol{A}$ with the E-PSO model parametrized by ζ, N, m, L, β and T while assuming that the E-PSO model assigned the same polar coordinates r_{jN}, θ_j to each network node $j = 1, 2, ..., N$ as the examined embedding algorithm. Given the

[4] As it is noted in Ref. [T4], at $L < 0$, the degree of the nodes in an E-PSO network can become increasing as a function of the radial coordinate when β is small enough and $|L|$ is large enough compared to m, meaning that if a negative value of L is assigned for a given network to be embedded, then the decreasing order of the node degrees does not necessarily correspond to the most probable radial node order, and therefore, caution should be exercised in these cases. Nonetheless, reducing the popularity fading parameter β not only enables the $L < 0$ values to affect the relationship between the node degrees and the order of the radial coordinates but in the meantime decreases the differences between all the radial coordinates, mitigating the effect of the choice of the radial node order.

[5] The ncMCE method is similar to Isomap presented in Sect. 3.1.2, but it omits the centering of the distance matrix before its decomposition, and instead of using the matrix of shortest path lengths measured directly along the network, ncMCE decomposes the SPL matrix measured along the minimum spanning tree of the network to be embedded.

hyperbolic distance x_{ij} measured for any node pair $i-j$ in the studied embedding and interpreting these values as the final hyperbolic distances emerged in the E-PSO model, the likelihood in question can be written as

$$\mathcal{L}(\boldsymbol{A} \mid \{r_{jN}, \theta_j\}, \zeta, N, m, L, \beta, T) = \prod_{1 \le j < i \le N} \tilde{p}(x_{ij})^{A_{ij}} \left[1 - \tilde{p}(x_{ij})\right]^{1-A_{ij}}, \tag{3.2.1.1}$$

yielding the formula

$$LL = -\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} A_{ij} \ln \tilde{p}(x_{ij}) - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (1 - A_{ij}) \ln \left[1 - \tilde{p}(x_{ij})\right] \tag{3.2.1.2}$$

for the logarithmic loss, where $\tilde{p}(x)$ is the so-called global connection probability, i.e. the probability of observing a connection in the E-PSO model between nodes lying at hyperbolic distance x from each other at the end of the network generation process, which can be calculated using as

$$\tilde{p}(x) = \frac{1}{1 + \mathrm{e}^{\frac{\zeta}{2T}(x - R_N)}} \tag{3.2.1.3}$$

using the formula of the cutoff distance R_N given in Sect. 2.1. Smaller logarithmic loss LL correspond to higher likelihood $\mathcal{L}(\boldsymbol{A} \mid \{r_{jN}, \theta_j\}, \zeta, N, m, L, \beta, T)$, and thus, to higher accordance between the given network embedding and the E-PSO model, which can be interpreted as an indicator of higher embedding quality.

The greedy routing score [6] quantifies the navigability of the embedded network relying solely on local information in each step. When performing greedy routing [43–45] on a hyperbolic embedding of a network, a walker tries to reach a destination node j walking along the links from a starting node i, knowing in each step only the current neighbors' hyperbolic distance from the destination node, and thus, stepping always along the link that currently leads the closest to the destination node. Getting stuck in a cycle makes the routing between the examined two nodes unsuccessful, and among the successful greedy paths the ones being closer in length to the corresponding shortest paths are considered to be better. The GR-score $\in [0, 1]$ (the higher the better) characterizes the fraction of successful paths among all greedy paths (the larger the better) and the length of the successful greedy paths (the smaller the better) simultaneously. It is defined for a network of N number of nodes as [6]

$$\text{GR-score} = \frac{1}{N \cdot (N-1)} \cdot \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \frac{\text{SPL}_{i \to j}}{\text{GRPL}_{i \to j}}, \tag{3.2.1.4}$$

where $\text{SPL}_{i \to j}$ stands for the hop-length of the shortest path between nodes i and j, and $\text{GRPL}_{i \to j}$ denotes the hop-length of the greedy path between the same start-destination node pair, which is considered to be infinity if the routing fails in reaching node j from node i. Note that although $\text{SPL}_{i \to j} = \text{SPL}_{j \to i}$ in undirected networks, the greedy path leading from node i to node j can differ from the greedy path leading from node j to node i even when the links are not directed.

According to Ref. [T1], when creating hyperbolic embeddings of networks multiple times with HyperMap or ncMCE, optimizing in each repetition for the E-PSO model of the same parameter settings (which are assumed to be the most congruent with the given edge list, obtained e.g. with the logarithmic loss minimizing method introduced in Ref. [T1]) but using a randomly chosen permutation of the radial order of nodes with equal degree, the distribution of the embedding quality (measured by the logarithmic loss and the greedy routing score) among the repetitions of the embedding is close to a Gaussian or consists of bell-shaped peaks. Assuming that the distribution of a quality score follows a normal distribution $\mathcal{N}(\mu, \sigma)$ (where μ denotes the mean and σ stands for the standard deviation), the expected value of the best quality score

achieved among $n_s \geq 2$ number of embedding samples can be expressed as

$$\mathbb{E}\left[\min_{1\leq i\leq n_s} LL_i\right] = \mu_{LL} - \sigma_{LL}\cdot g(n_s) \tag{3.2.1.5}$$

for the logarithmic loss and as

$$\mathbb{E}\left[\max_{1\leq i\leq n_s} GR_i\right] = \mu_{GR} + \sigma_{GR}\cdot g(n_s) \tag{3.2.1.6}$$

for the greedy routing score with the function $g(n_s)$ given by

$$g(n_s) = \sqrt{2\ln(n_s)} - \frac{\ln(\ln(n_s)) + \ln(4\pi) - 2\Gamma}{2\sqrt{2\ln(n_s)}} + \mathcal{O}\left(\frac{1}{\ln(n_s)}\right), \tag{3.2.1.7}$$

where $\Gamma = 0.5772156649...$ is the Euler–Mascheroni constant [46]. Therefore, fitting Eqs. (3.2.1.5) and (3.2.1.6) to the results of some embedding trials one can predict the best quality scores that can be achieved under larger n_s number of repetitions of the embedding. As it is shown in Fig. 3.2.1.1, this fit can be applied even when a quality distribution contains more than one bell-shaped peak since it automatically chooses the peak lying at the end of the best quality scores. Besides, Fig. 3.2.1.2 exemplifies that both Eq. (3.2.1.5) and Eq. (3.2.1.6) can be relatively well fitted to real data, not only in the case of ncMCE but also for HyperMap. Thus, it can be concluded that although the optimal radial order of equally popular nodes can not be determined analytically based on the E-PSO model, usually the different radial orderings of nodes having the same degree are not equivalent to each other from the viewpoint of the achieved embedding quality, and the possible improvement obtained by repeatedly testing random permutations of these nodes seems to be well describable by Eqs. (3.2.1.5) and (3.2.1.6).

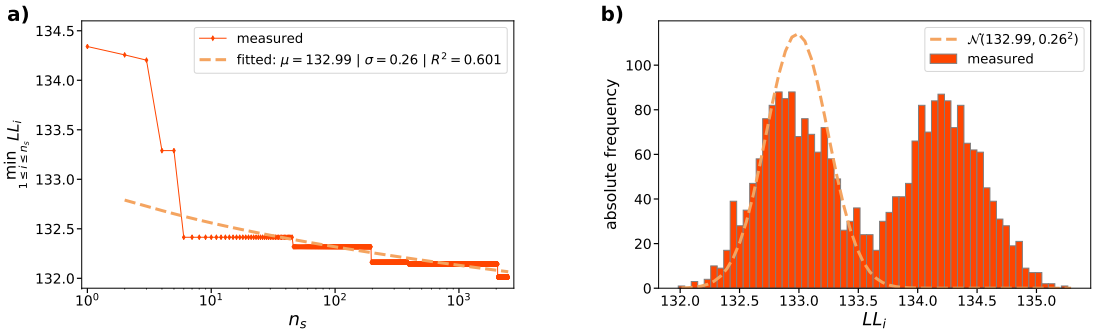

FIGURE 3.2.1.1: **The predictability of the improvement in the logarithmic loss achievable by further repetitions of hyperbolic embeddings with E-PSO-based radial node arrangement.** The 12th layer of the multiplex Pierre Auger collaboration network [47] (consisting of $N = 38$ number of nodes and $E = 102$ number of edges, describing the collaborations related to the SD-reconstruction) was embedded in the hyperbolic plane repeatedly with the ncMCE method. The logarithmic loss obtained in the ith embedding trial is denoted by LL_i. Panel a) shows the achieved best quality score (i.e., smallest logarithmic loss) as a function of the number of embedding samples n_s, together with the fitted curve following the formula of Eq. (3.2.1.5). The quality of the fit is characterized by the coefficient of determination R^2 specified in the legend. Panel b) depicts the density function of the measured quality scores together with the normal distribution having the mean μ and the standard deviation σ yielded by the fitted curve of panel a). Although the distribution of the measured quality scores among the different embedding trials is bimodal, the fit in panel a) automatically identifies the peak that is relevant for predicting the achievable best quality scores as a function of the number of tested radial node orders. The figure was taken from Ref. [T1].

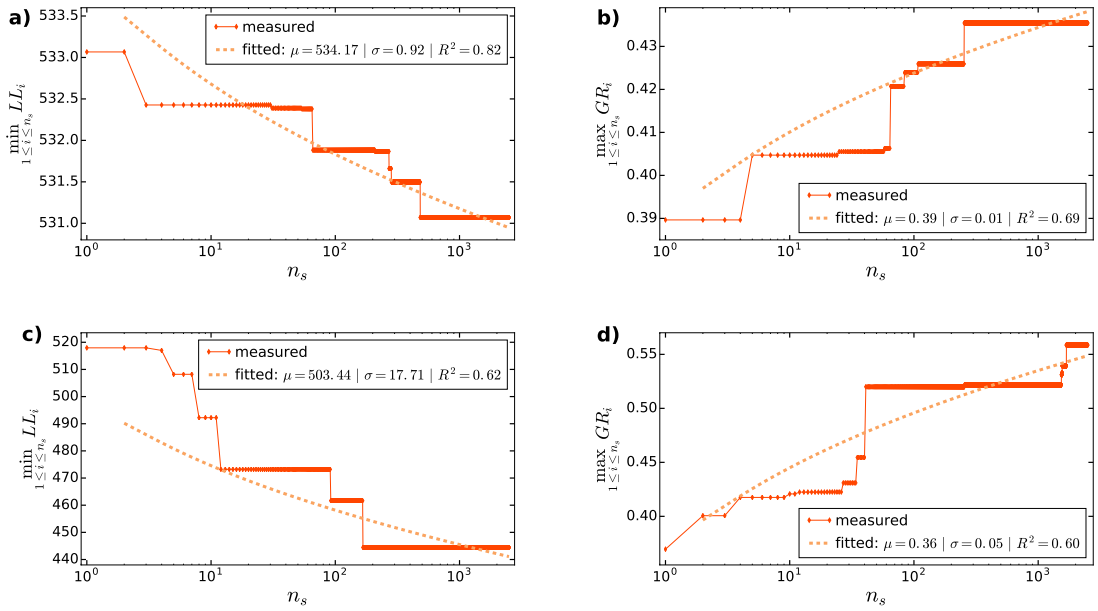

FIGURE 3.2.1.2: **The agreement between the measured and the expected improvement in the embedding performance during the repetition of hyperbolic embeddings using E-PSO-based radial node arrangement.** The figure shows the achieved best logarithmic loss (left, the lower the better) and greedy routing score (right, the higher the better) as a function of the number n_s of embedding samples in the case of a protein interaction network of $N = 161$ number of nodes and $E = 209$ number of edges from the PDZBase database [48]. Panels a) and b) refer to the ncMCE embedding algorithm, while panels c) and d) were created using the HyperMap method. The dashed curves were obtained by fitting the corresponding formula of Eqs. (3.2.1.5)–(3.2.1.6) to the measured solid curves. The legend of each panel lists the fitted coefficients, namely the mean μ and the standard deviation σ characterizing the peak on that side of the given quality score's measured distribution which corresponds to the best results, and also the coefficient of determination R^2 describing the quality of the fits. The figure was taken from Ref. [T1].

3.2.2 Hyperbolic radial coordinates in the d-dimensional space according to the dPSO model

Considering the success of the PSO-based calculation of the radial coordinates in two-dimensional hyperbolic embeddings, a naturally arising idea is that for creating d-dimensional hyperbolic embeddings, the angular coordinates obtained from a dimension reduction method could be joined with a radial node arrangement that is optimized for the dPSO model [T3]. As it is mentioned in Sect. 2.4, there is more than one option regarding the multiplying factor of the initial radial coordinates when extending the two-dimensional PSO model to any integer number of dimensions $d \geq 2$. Although of the two variations given by

$$r_{jj} = \begin{cases} \frac{2}{\zeta \cdot (d-1)} \cdot \ln j & \text{if } 0 \leq T < \frac{1}{d-1}, \\ \frac{2T}{\zeta} \cdot \ln j & \text{if } \frac{1}{d-1} < T \end{cases} \tag{3.2.2.1}$$

and

$$r_{jj} = \begin{cases} \frac{2}{\zeta} \cdot \ln j & \text{if } 0 \leq T < \frac{1}{d-1}, \\ \frac{2T(d-1)}{\zeta} \cdot \ln j & \text{if } \frac{1}{d-1} < T \end{cases} \tag{3.2.2.2}$$

only the latter can yield degree decay exponents below 2, both of these approaches may be suitable for the generation, and thus, also the hyperbolic embedding of networks with $2 \leq \gamma$. These two variants of the dPSO model provide two different ways for the radial arrangement of a network in a d-dimensional hyperbolic space of curvature $K = -\zeta^2$. Unless the average clustering coefficient of the network to be embedded is very close to 0, it can be assumed in both cases that the temperature T suiting the network is smaller than the critical value $T_c = 1/(d-1)$, and therefore, the radial coordinate formulas of the $0 \leq T < 1/(d-1)$ case can be applied. Accordingly, if the popularity fading parameter β is determined from the degree decay exponent γ as

$$\beta = \frac{1}{\gamma - 1}, \tag{3.2.2.3}$$

then the radial coordinate of the node having the jth ($j = 1, 2, ..., N$) largest degree (with ties in the order of node degrees broken arbitrarily) can be calculated in a d-dimensional embedding as

$$r_{jN} = \beta \cdot r_{jj} + (1-\beta) \cdot r_{NN} = \beta \cdot \frac{2}{\zeta \cdot (d-1)} \cdot \ln j + (1-\beta) \cdot \frac{2}{\zeta \cdot (d-1)} \cdot \ln N, \tag{3.2.2.4}$$

while if the formula

$$\beta = \frac{1}{(d-1) \cdot (\gamma - 1)} \tag{3.2.2.5}$$

is used, then the radial coordinate in question can be written as

$$r_{jN} = \beta \cdot r_{jj} + (1-\beta) \cdot r_{NN} = \beta \cdot \frac{2}{\zeta} \cdot \ln j + (1-\beta) \cdot \frac{2}{\zeta} \cdot \ln N. \tag{3.2.2.6}$$

Both of these approaches were tested in Ref. [T4] – see Sect. 4.3.4 –, joining their results with the high-dimensional angular positions obtained from the LE (Sect. 3.1.1) and the ISO (Sect. 3.1.2) methods, demonstrating that hyperbolic embeddings of good quality can be created with both techniques, assuming either the largest radial coordinate r_{NN} or the popularity fading parameter β to be dependent on the number of dimensions d. Note that at $d = 2$, the two approaches are equivalent to each other.

3.3 Dimension reduction in the hyperbolic space: the hydra method

The nodes of a single-component, undirected, and possibly weighted network where higher link weights mean weaker connections or larger dissimilarities can be placed directly in the hyperbolic space with the method named hyperbolic distance recovery and approximation (hydra) [10], which aims at finding such a node arrangement in which the pairwise hyperbolic distances give back[6] the topological distances described by the shortest path lengths measured along the network. As the expected Euclidean distances were converted to expected inner products in the ISO method, hydra converts the matrix of expected hyperbolic distances to an expected matrix of so-called Lorentz products (described in Sect. 1.3). A Lorentz product matrix, similarly to an inner product matrix in the Euclidean case, can be expressed with a

[6]The reproduction of the shortest path lengths SPL_{ij} in the form of the hyperbolic distances x_{ij} means that the loss function $\sum_{i=1}^{N}\sum_{j=1}^{N}\left(x_{ij} - \mathrm{SPL}_{ij}\right)^2$ is minimized in the embedding space of a given number of dimensions. Utilizing the monotonicity of the hyperbolic cosine function, this task can be rephrased as the minimization of the loss function $\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\cosh(\zeta \cdot x_{ij}) - \cosh(\zeta \cdot \mathrm{SPL}_{ij})\right)^2$, where $\cosh(\zeta \cdot x_{ij})$ is the Lorentz product corresponding to the hyperbolic distance x_{ij} when the curvature of the hyperbolic space can be written as $K = -\zeta^2$.

simple product of the node coordinate matrix if the nodes are placed in the hyperboloid representation of the d-dimensional hyperbolic space, namely as $\boldsymbol{X} \cdot \boldsymbol{J} \cdot \boldsymbol{X}^{\mathrm{T}}$ (see Eq. (1.3.2)), where $\boldsymbol{J}$ is the diagonal matrix of size $(d+1) \times (d+1)$ containing the values $+1, -1, -1, ..., -1$ in the diagonal. To find the coordinate matrix $\boldsymbol{X}$ of size $N \times (d+1)$ that suits this equation written up for the matrix of expected Lorentz products $\boldsymbol{L}$, hydra uses the eigendecomposition $\boldsymbol{L} = \boldsymbol{Q} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{Q}^{\mathrm{T}}$, and composes the coordinate matrix $\boldsymbol{X}$ from the eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$ and the corresponding eigenvectors (i.e., the columns of the matrix $\boldsymbol{Q}$) $\underline{q}_1, \underline{q}_2, \dots, \underline{q}_N$ as $\boldsymbol{X} = \left[\sqrt{\lambda_1}\,\underline{q}_1, \sqrt{-\lambda_{N-d+1}}\,\underline{q}_{N-d+1}, \sqrt{-\lambda_{N-d+2}}\,\underline{q}_{N-d+2}, \dots, \sqrt{-\lambda_N}\,\underline{q}_N\right]$[7]. Then, this embedding obtained in the hyperboloid representation of the d-dimensional hyperbolic space is mapped to the Poincaré ball representation (presented in Sect. 1.2), using the hyperboloid coordinates from the second to the $(d+1)$th one as the direction vector in the d-dimensional Poincaré ball, and assigning to each network node a radial coordinate that ensures the preservation of the hyperbolic distance between any node and the origin during the change between the representations. As it is explained in Sect. 4.2, inspired by the algorithm of hydra, I developed in Ref. [T4] a hyperbolic dimension reduction technique that works on directed networks too.

[7]Note that it is assumed here that the number of embedding dimensions d is not larger than the number of negative eigenvalues of the matrix of pairwise expected Lorentz products.

4 Model-independent embedding of directed networks in hyperbolic spaces

Directed links, allowing the traverse between two nodes only in one direction, can express asymmetric connections between the network nodes, such as the dominant-subordinate relations in hierarchical networks, the consumer-producer relations in food webs, or flows of different strengths along the links in different directions. Obviously, nodes that have mostly incoming links may play a very different role in the system compared to nodes whose links are mostly outgoing or nodes that possess a balanced number of in- and out-neighbors. Thus, it is reasonable to assume that link directions may carry valuable information regarding the underlying geometry of a network, and network embeddings that simply ignore link directions may suffer from severe information loss.

Nevertheless, though several hyperbolic embedding methods have been proposed in recent years, most approaches are not able to deal with directed networks. The only embedding algorithms that can take into account the link directions and use hyperbolic geometry were introduced in Ref. [49] – where two-dimensional hyperbolic embeddings were created via optimizing a likelihood according to a new, directed version of the static network generation model originally described in Ref. [2] – and in Ref. [12] – where to each network node a Gaussian distribution is assigned with a mean vector given in the hyperboloid model of the hyperbolic space (which was also used in the hydra method described in Sect. 3.3), learning the parameters of the nodes' representation using a neural network. Most recently, the area of directed hyperbolic embeddings was also contributed to by Ref. [T4], where the aim was the development of relatively simple dimension reduction techniques that can arrange the network nodes in the native representation of the hyperbolic space while capturing the asymmetry of the connections when embedding directed networks. During the development, a definite objective was to avoid the optimization for any specific network model that generates networks in the hyperbolic space, and to allow the possibility to utilize the beneficial effects of increasing the number of dimensions of the embedding space (i.e., the number of components of the position vectors describing the system), as it is completely natural in the case of Euclidean embeddings [36–40].

To enable the representation of the possibly asymmetric topological relations of a network through hyperbolic distances, the approaches introduced in Ref. [T4] can assign two position vectors to each node: the source position characterizes the behavior of the given node as a source of links, while the target position describes its function as a target of links. The more pronounced role the directionality in a network plays, the more difference emerges between the source and the target node arrangements, while for undirected networks, the obtained source positions are identical to the target positions. Although being not capable of embedding multi-component networks or isolated nodes that have no connections at all, the proposed methods can embed any weakly connected networks[1], even those in which some nodes have either no outbound or no inbound links. Naturally, nodes of zero out-degree get no source positions (as for these nodes, no connection preference is known as a source of links) and (lacking the information about what kind of nodes these tend to receive links from) nodes of zero in-degree do not appear in the embedding as targets. It is important to remark that the proposed algorithms map the topological distances to hyperbolic distances between a node's source and another

[1] Weakly connected graphs are those that form a connected graph when discarding the link directions, but do not contain a directed path from each node to every other node.

node's target position, and e.g. a small hyperbolic distance between two nodes' source representations does not mean that the two nodes are probably connected to each other directly, but that they probably establish links towards the same set of nodes.

As an example, Fig. 4.1 shows a two-dimensional hyperbolic embedding of a directed network of political weblogs [50, 51], visualized on a pair of disks, i.e. a source disk displaying each node at its source position and a target disk that contains the nodes at their target position, with the links of the network pointing from the source disk to the target disk. Note that the hyperbolic distances between the nodes must be interpreted here as all nodes would be located on a single native disk, and the separation of the source and target positions only serve visualization purposes.

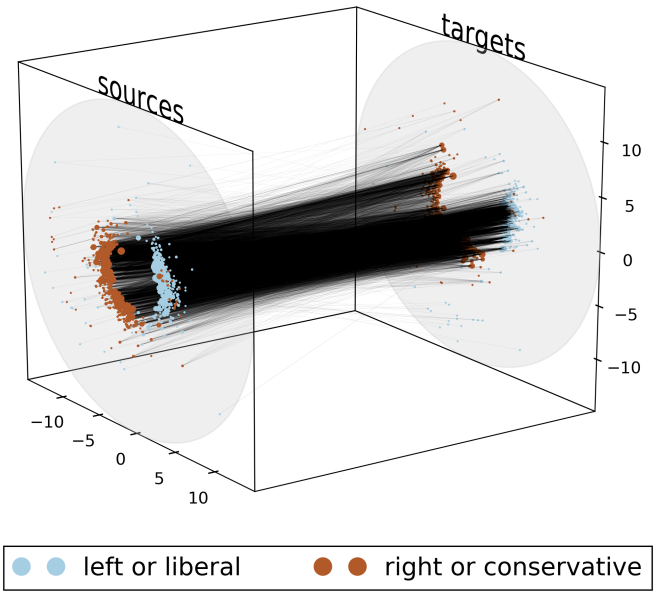

FIGURE 4.1: **Two-dimensional hyperbolic embedding of a directed network of political weblogs.** The network consists of $N = 1222$ number of U.S. political weblogs from before the 2004 presidential election, which are connected by $E = 19021$ number of hyperlinks that appear as lines extending from the source plane to the target plane. The embedding was created with my model-independent Euclidean-hyperbolic conversion MIC (Sect. 4.1.2) from the result of the Euclidean TREXPEN-R embedding method (Sect. 4.1.1) introduced in Ref. [T4]. The color of a node indicates the political leaning of the given weblog, while larger node sizes in the source and the target plane indicate larger out- and in-degrees, respectively.

Figure 4.2 presents the different embedding algorithms studied in Ref. [T4], which are the subject of the current chapter. Section 4.1 reveals how the network nodes can be placed in the native representation of the d-dimensional hyperbolic space through creating an embedding in the d-dimensional Euclidean space: Sect. 4.1.1 describes the applied Euclidean dimension reduction techniques and Sect. 4.1.2 gives the details of my model-independent Euclidean-hyperbolic conversion method named MIC. Then, Section 4.2 deals with a fundamentally different approach of hyperbolic embedding, where the dimension reduction provides an embedding directly in the hyperbolic space, eliminating the dependence on any Euclidean node arrangement. Finally, Sect. 4.3 shows some applications and experiments regarding the studied embedding algorithms. Here, Sect. 4.3.1 illustrates the methods' ability to spatially separate the nodes of different communities. Then, Sect. 4.3.2 presents the applied (model-independent) quality measures according to which the embeddings of both directed and undirected real networks are evaluated in Sect. 4.3.3 and Sect. 4.3.4, respectively. Lastly, Sect. 4.3.5 discusses how the considered embedding methods can be used on weighted networks.

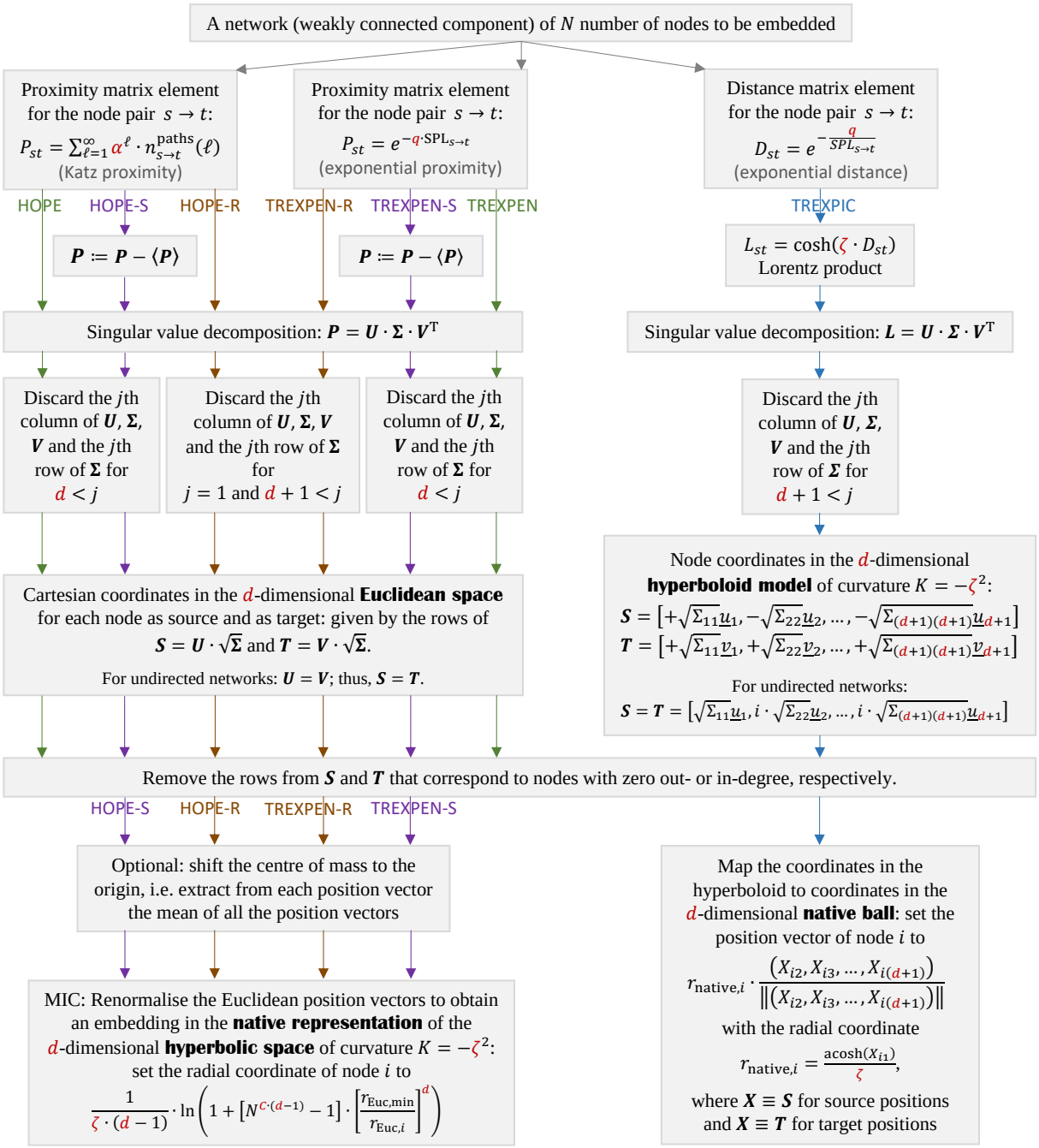

FIGURE 4.2: **Flowchart of the hyperbolic embedding algorithms proposed for directed networks.** The left side of the figure traces the algorithmic steps for creating a hyperbolic embedding with the High-Order Proximity preserved Embedding (HOPE), our TRansformation of EXponential shortest Path lengths to EuclideaN measures (TREXPEN) and their variants by converting the Euclidean node arrangement obtained from them to a hyperbolic one with our model-independent conversion (MIC). The right side of the figure shows the algorithmic steps of our method named TRansformation of EXponential shortest Path lengths to hyperbolIC measures (TREXPIC), which embeds networks directly in the hyperbolic space. The embedding parameters are written in red: the parameters α and q adjust how the elements of the reduced matrices depend on the distances measured along the graph to be embedded, d denotes the number of dimensions of the embedding space, ζ (usually set to 1) tunes the curvature of the hyperbolic space, and C (usually set to 2) controls the extent of the graph in the hyperbolic space when using MIC. The figure was taken from Ref. [T4].

4.1 Hyperbolic embedding based on a Euclidean node arrangement

As detailed in Sects. 3.1 and 3.2, it has become a common practice in the literature of hyperbolic embedding of undirected networks [5, 6] to transform Euclidean embeddings exhibiting circular node patterns into hyperbolic ones by replacing the radial coordinates obtained from dimension reduction with those that are the most probable according to the popularity-similarity optimization (PSO) model [3] of growing hyperbolic networks described in Sect. 2.1. The here-applied Euclidean embeddings – including Isomap presented Sect. 3.1.2 – are usually built on the singular value decomposition (SVD) of a matrix derived from topological distances, trying to map small distances measured along the network to small Euclidean distances [36, 52, 53]. SVD has also been used in the High-Order Proximity preserved Embedding (HOPE) [39] method that, utilizing the fact that the left and the right singular vectors of an asymmetric matrix differ from each other, assigns in directed networks two position vectors to each node: one that represents the given node as a source of links and one that characterizes its behavior as a target of links. The aim of HOPE is to reproduce the elements of a proximity matrix (where smaller topological distances are indicated with higher values) in the form of pairwise inner products between the source and target node positions. Note that compared to a distance-based Euclidean representation where there is no obvious preference regarding the radial coordinates, building on inner products makes it effortless to separate the radial and the angular node coordinates' contribution to the geometric relations.

Inspired by HOPE, I introduced the Euclidean embedding algorithm named TRansformation of EXponential shortest Path lengths to EuclideaN measures (TREXPEN) [T4], which characterizes the topological proximity between two nodes with the shortest path length between them instead of considering all the connecting paths like HOPE, as specified in Sect. 4.1.1. By representing infinite topological distances measured in the absence of any paths from a given node to an other with finite matrix elements, both methods enable the embedding of weakly connected directed networks too, besides strongly connected ones in which every node can be reached from any other node. To obtain hyperbolic-like angular patterns from HOPE and TREXPEN, following the ideas of Ref. [6], I created two modified versions in Ref. [T4] for both methods, which are also described in Sect. 4.1.1. Without assuming any specific hyperbolic model as the generator of the network to be embedded in the hyperbolic space, the created Euclidean embeddings can be transformed to hyperbolic ones using the conversion method [T4] presented in Sect. 4.1.2, which utilizes not only the angular but also the radial coordinates obtained in the Euclidean space and aims at preserving the relative attractivity of the different radial positions from the point of view of link creation.

4.1.1 Euclidean embedding methods optimizing inner products

The High-Order Proximity preserved Embedding (HOPE) [39] assigns Euclidean source and target positions to the nodes of a network based on the SVD of a matrix of topological proximities composed from the pairwise Katz indexes [54] of the network nodes. The Katz index and the corresponding element of the proximity matrix $\boldsymbol{P}$ is defined for the ordered node pair $s \to t$ as

$$P_{st} = \sum_{\ell=1}^{\infty} \alpha^{\ell} \cdot n_{s\to t}^{\text{paths}}(\ell), \tag{4.1.1.1}$$

where $n_{s\to t}^{\text{paths}}(\ell)$ stands for the total number of paths of hop-length $\ell > 0$ leading from node s to node t and $0 < \alpha < 1$ is a decay parameter controlling how fast the contribution of the paths decreases with their length ℓ (the smaller the value of α, the more important the shorter paths compared to the longer ones). Note that the Katz index P_{ss} of a node s with itself is determined solely by the number of cycles that include the given node and the paths of length 0 are not

considered in Eq. (4.1.1.1), meaning that the highest Katz indexes do not necessarily fall in the diagonal of the matrix $\boldsymbol{P}$.

In the Euclidean embedding method TRansformation of EXponential shortest Path lengths to EuclideaN measures (TREXPEN) introduced in Ref. [T4], based on the success of the SPL-based ISO method described in Sect. 3.1.2, I replaced the Katz matrix used by HOPE with an exponential proximity matrix that takes into account only the shortest path lengths (the SPLs) and not all the paths. Namely, given the shortest path length $\mathrm{SPL}_{s\to t}$ measured from node s to node t along the network, TREXPEN calculates the proximity matrix's element assigned to the node pair $s \to t$ as

$$P_{st} = e^{-q\cdot \mathrm{SPL}_{s\to t}}, \tag{4.1.1.2}$$

yielding larger values for smaller topological distances, as it is expected from a proximity-like measure. The possible largest value of the here-defined exponential proximity is 1, yielded by 0 shortest path lengths in the diagonal. The multiplying factor $q > 0$ controls the speed of the decay of the proximity with the increase in the shortest path length (the smaller the value of q, the slower the decay).

An essential property of both of the above proximities given by Eqs. (4.1.1.1) and (4.1.1.2) is that they automatically enable the embedding of not only strongly connected parts of directed networks where each node can be reached from every other node, but also weakly connected components that contain even infinitely large path lengths. This is due to the fact that the above matrix elements become a finite number, namely 0 in such cases where node t is not reachable from node s along the network (i.e., when there is no path of finite length connecting node s to node t), because paths of infinite length do not have a contribution to the sum in Eq. (4.1.1.1) (since $\alpha < 1$) and Eq. (4.1.1.2) converts $\mathrm{SPL}_{s\to t} = \infty$ to 0 since $-q < 0$.

It is important to notice that although it is possible to consider mainly just the shortest ones of the path lengths in the Katz index of Eq. (4.1.1.1) by setting the decay parameter α to extremely small values, but, at the same time, these small α values will also reduce the importance of the longer shortest paths compared to the lowest path lengths of the graph. On the contrary, the exponential proximity defined by Eq. (4.1.1.2) enables various weighting of the shortest paths of different lengths; namely, the decay from 1 is approximately linear and all the non-zero proximity values remain close to 1 for extremely small values of q (see the Taylor series expansion $e^{-q\cdot \mathrm{SPL}_{s\to t}} \approx 1 - q \cdot \mathrm{SPL}_{s\to t}$), while all the off-diagonal elements of the proximity matrix $\boldsymbol{P}$ fall very close to 0 for large q factors. All things considered, while an exponential proximity of a large q is similar to a Katz index of a small α, the decrease in q has different effects compared to the increase in α: the former increases the relative importance of the longer shortest paths, whilst the latter increases the importance of all the longer paths, not just the ones that connect two nodes via the possible least hops.

After deriving a proximity matrix $\boldsymbol{P}$ from the network topology, the HOPE method [39] and, inspired by it, the newly-introduced TREXPEN algorithm use SVD to find the node position vectors corresponding to the given proximity matrix. Given the SVD $\boldsymbol{P} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^T$ of the matrix $\boldsymbol{P}$ of topological proximities, the Euclidean coordinate matrices defined as

$$\boldsymbol{S} = \boldsymbol{U} \cdot \sqrt{\boldsymbol{\Sigma}} \tag{4.1.1.3}$$

and

$$\boldsymbol{T} = \left(\sqrt{\boldsymbol{\Sigma}} \cdot \boldsymbol{V}^T\right)^T = \boldsymbol{V} \cdot \sqrt{\boldsymbol{\Sigma}}, \tag{4.1.1.4}$$

yield $\boldsymbol{S} \cdot \boldsymbol{T}^T = \boldsymbol{U} \cdot \sqrt{\boldsymbol{\Sigma}} \cdot \sqrt{\boldsymbol{\Sigma}} \cdot \boldsymbol{V}^T = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^T = \boldsymbol{P}$. Accordingly, the elements of the $N \times N$-sized proximity matrix $\boldsymbol{P}$ are equal to the inner products between the N-dimensional Cartesian position vectors represented by the rows of the above-defined $\boldsymbol{S}$ and $\boldsymbol{T}$ matrices. For an asymmetric proximity matrix $\boldsymbol{P}$, the SVD results in $\boldsymbol{U} \neq \boldsymbol{V}$, and thus, $\boldsymbol{S} \neq \boldsymbol{T}$, meaning that in

directed networks the SVD assigns to each node two position vectors that can differ from each other. It is important to remark that although the number of rows in the coordinate matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ is always equal to the number of nodes N of the embedded graph, i.e. $\boldsymbol{S}$ and $\boldsymbol{T}$ assign a coordinate array to each one of the network nodes, some of these position vectors can be meaningless. Namely, since nodes with 0 out-degree do not function as sources of links in the network, their source positions given by the corresponding rows of the matrix $\boldsymbol{S}$ are irrelevant. Besides, the same can be said regarding the target positions assigned to nodes of 0 in-degree by the matrix $\boldsymbol{T}$. Consequently, in my investigations I considered only those nodes to be placed in the embedding space as a source of links that have non-zero out-degree and assigned a target position vector only to those nodes that have non-zero in-degree.

According to Ref. [39], if d-dimensional node position vectors are needed (meaning that the number of columns of $\boldsymbol{S}$ and $\boldsymbol{T}$ has to be d), then an optimal solution for minimizing the L2-norm of $\boldsymbol{P} - \boldsymbol{S} \cdot \boldsymbol{T}^T$ is to perform the SVD of the proximity matrix $\boldsymbol{P}$, keep only the first (largest) d number of singular values from the diagonal matrix $\boldsymbol{\Sigma}$, and calculate the $N \times d$-sized position matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ based on Eqs. (4.1.1.3) and (4.1.1.4) but using only the first d number of columns (singular vectors) in the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, i.e., use the formulas

$$\boldsymbol{S} = [\sqrt{\sigma_1} \cdot \underline{u}_1 \, , \, \sqrt{\sigma_2} \cdot \underline{u}_2 \, , \, \sqrt{\sigma_3} \cdot \underline{u}_3 \, , \, \ldots \, , \, \sqrt{\sigma_d} \cdot \underline{u}_d] \tag{4.1.1.5}$$

and

$$\boldsymbol{T} = [\sqrt{\sigma_1} \cdot \underline{v}_1 \, , \, \sqrt{\sigma_2} \cdot \underline{v}_2 \, , \, \sqrt{\sigma_3} \cdot \underline{v}_3 \, , \, \ldots \, , \, \sqrt{\sigma_d} \cdot \underline{v}_d]. \tag{4.1.1.6}$$

As it was described in Ref. [T4], when arranging the network nodes in the embedding space with the aim of reproducing the elements of a topological proximity matrix in the form of inner products (representing smaller topological distances with larger inner products), using only non-negative proximity values (i.e., non-negative expected inner products) restricts the range of angular distances between the source and target position vectors as $\pi/2 < \Delta\theta$ makes the inner product negative. In the coalescent embedding methods presented in Ref. [6], circular arrangements of the network nodes were created using two techniques: either the matrix characterizing the topological distances was centered before the dimension reduction (see the centering step in Sect. 3.1.2), or it was not centered but the first dimension of the embedding was discarded. Inspired by this, I proposed in Ref. [T4] two modifications both for HOPE and TREXPEN that enable the generation of such network layouts that are not restricted to a narrow angular range:

- HOPE-S and TREXPEN-S are those versions, in which the mean of the proximity matrix elements given by Eqs. (4.1.1.1) and (4.1.1.2) is *shifted* to 0 before performing the SVD, by subtracting the mean of all the proximity values from each element of $\boldsymbol{P}$.
- In HOPE-R and TREXPEN-R, the same proximity matrix is decomposed as in the case of HOPE and TREXPEN, but the first dimension of the embedding is considered to be *redundant* and the singular values from the second to the $d+1$th one are used in a d-dimensional embedding of a network. The rationale behind this approach is that the discarded first dimension contains information mainly about the location of the embedded network as a whole compared to the origin, while the focus of an embedding should be on the positions of the network nodes relative to each other.

Since in practice the center of mass (COM) of the circular patterns obtained from HOPE-S, TREXPEN-S, HOPE-R and TREXPEN-R does not coincide exactly with the origin of the embedding space's coordinate system, I tested in each of my measurements whether shifting the embedding's COM to the origin has an improving effect on the embedding quality. Note that in order to not distort the topology-reflecting relations between the source and the target node positions, the location of the center of mass was calculated jointly for the source and target positions.

In Ref. [6], the angular coordinates of the circular node arrangements obtained in the Euclidean space from dimension reduction were re-used in hyperbolic embeddings (see Sect. 3.1). Similarly, the Euclidean embeddings obtained from the methods named HOPE-S, TREXPEN-S, HOPE-R and TREXPEN-R can be converted to node arrangements having a typical look of hyperbolic embeddings, as it is shown in Sect. 4.1.2. Note that my Euclidean-hyperbolic conversion method presented in Sect. 4.1.2 can be applied on angularly restricted Euclidean node arrangements too, but yields in such cases hyperbolic embeddings of a limited angular range, which is uncommon and may reduce the spatial separation between the communities of the network, as it will be exemplified by the two-dimensional layouts in Figs. 4.3.1.1, 4.3.1.2 and 4.3.4.1. Therefore, during my investigations, I used the original (angularly restricted) HOPE embedding method [39] only in its well-known Euclidean form, simply to provide a reference point for measuring the quality of its new, angularly not restricted versions and their hyperbolic counterparts. Analogously, I considered both the Euclidean and the hyperbolic geometries only in the case of TREXPEN-S and TREXPEN-R and not in the case of TREXPEN, which I used solely as a Euclidean reference point.

4.1.2 From Euclidean inner product to hyperbolic distance: a model-independent Euclidean-hyperbolic conversion method

As it was presented in Sect. 3.2, the usual practice when transforming Euclidean embeddings into hyperbolic ones is to assume a specific hyperbolic network model as the generator of the network to be embedded and optimize the node positions according to that given model. To avoid the influence of a specific network model on the hyperbolic embeddings, I introduced in Ref. [T4] a model-independent conversion (MIC) for creating hyperbolic embeddings from the Euclidean methods described in Sect. 4.1.1, which map high proximities measured along the graph to high inner products in the Euclidean space. While keeping the angular node arrangement unaltered as usual, instead of simply discarding the Euclidean radial coordinates obtained from the dimension reduction, MIC derives the hyperbolic radial coordinates from the Euclidean ones, and thus, it is able to take into account the possible interdependencies appearing between the radial and the angular coordinates in the Euclidean embedding.

The embedding methods presented in Sect. 4.1.1 generate such node arrangements in the Euclidean space in which high node-node proximities measured along the graph are reflected by high inner products between the nodes' position vectors. Considering the Euclidean source position vector $\underline{s}_s$ of node s and the target position vector $\underline{t}_t$ of node t, the inner product $\underline{s}_s \cdot \underline{t}_t$ that can be treated as a proxy of the $s \to t$ connection probability can be calculated from the radial coordinates $r_s^{\text{source}} = \|\underline{s}_s\|$, $r_t^{\text{target}} = \|\underline{t}_t\|$ and the angular distance $\theta_{s\to t}$ as

$$\underline{s}_s \cdot \underline{t}_t = r_s^{\text{source}} \cdot r_t^{\text{target}} \cdot \cos(\theta_{s\to t}). \tag{4.1.2.1}$$

Consequently, in these embeddings higher connection probabilities correspond to larger radial coordinates and smaller angular distances.

MIC aims at transferring such Euclidean node arrangements into the native representation (see Sect. 1.1) of the d-dimensional hyperbolic space of curvature $K < 0$, where small hyperbolic distances are considered to be the indicators of high connection probabilities. In the native ball, the hyperbolic distance between the source position of node s (given by $\underline{s}_s$) and the target

position of node t (denoted by $\underline{t}_t$) can be approximated[2] according to Eq. (1.1.2) as

$$x_{s\to t} \approx r_s^{\text{source}} + r_t^{\text{target}} + \frac{2}{\zeta}\cdot\ln\left(\frac{\theta_{s\to t}}{2}\right) \tag{4.1.2.2}$$

with $\zeta = \sqrt{-K}$, $r_s^{\text{source}} = \|\underline{s}_s\|$, $r_t^{\text{target}} = \|\underline{t}_t\|$ and $\theta_{s\to t} = \arccos\left(\frac{\underline{s}_s\cdot\underline{t}_t}{\|\underline{s}_s\|\,\|\underline{t}_t\|}\right)$, showing that higher connection probabilities correspond to smaller radial coordinates and/or smaller angular distances in the hyperbolic space.

According to Eqs. (4.1.2.1) and (4.1.2.2), the contribution of the radial and the angular coordinates to the geometric measures in question can be well separated. Thus, as the simplest solution, the radial and the angular node arrangement can be subjected to two independent Euclidean-hyperbolic transformations. The Euclidean embeddings that optimize for inner products and the hyperbolic embeddings that optimize for hyperbolic distances treat the angular distances in the same way: smaller values yield higher connection probabilities. Therefore, MIC does not change the angular node arrangement of circular Euclidean embeddings when switching to a hyperbolic embedding, just like the earlier, PSO-based Euclidean-hyperbolic transformation described in Sect. 3.2. Nevertheless, while for obtaining larger inner products the larger radial coordinates are those that are more preferable (making the outer nodes more attractive in general), when aiming at smaller hyperbolic distances, the smaller radial coordinates are those that are more advantageous (making the inner nodes' general attractivity higher). Thus, there is an obvious need for a transformation of the radial node arrangement when turning a Euclidean embedding into a hyperbolic one.

As it was described in Ref. [T4], the basis of MIC is the straightforward assumption that the Euclidean and the corresponding hyperbolic radial arrangement of the same network must be easily reconcilable to each other when converted to the same space since both the Euclidean and the hyperbolic radial coordinates describe the same network, just like the two node arrangements obtained from these in the common space. As such a common space, i.e. as the pass-through between the polynomially expanding Euclidean and the exponentially expanding hyperbolic spaces, MIC uses the linearly expanding half-line. A radial coordinate on this half-line is calculated as

$$r_{\text{line}}(r_{\text{Euc}}) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\cdot r_{\text{Euc}}^d \tag{4.1.2.3}$$

from a radial coordinate r_{Euc} given in the d-dimensional Euclidean space and as

$$r_{\text{line}}(r_{\text{hyp}}) = \frac{e^{\zeta\cdot(d-1)\cdot r_{\text{hyp}}}-1}{\zeta\cdot(d-1)\cdot 2^{d-1}} \tag{4.1.2.4}$$

from a radial coordinate r_{hyp} given in the native representation of the d-dimensional hyperbolic space of curvature $-\zeta^2$, as using these formulas, the volume $V_d^{\text{Euc}}(r_{\text{Euc}}) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\cdot r_{\text{Euc}}^d$ and (for not too small values of the radius r_{hyp} [13]) $V_d^{\text{hyp}} = \frac{e^{\zeta\cdot(d-1)\cdot r_{\text{hyp}}}-1}{\zeta\cdot(d-1)\cdot 2^{d-1}}$ of d-dimensional Euclidean and hyperbolic balls become linear functions of the transformed radial coordinates, namely $V_d^{\text{Euc}} = r_{\text{line}}(r_{\text{Euc}})$ and $V_d^{\text{hyp}} = r_{\text{line}}(r_{\text{hyp}})$.

To determine the demanded hyperbolic embedding's mapping on the half-line from that of the Euclidean node arrangement, MIC matches for each node its relative radial attractivity (measured compared to the highest one) between the different node arrangements obtained

[2] In the measurements of the embedding quality presented later in this chapter, the hyperbolic distances were never approximated but always calculated according to the exact formula written as $x_{s\to t} = \frac{1}{\zeta}\cdot\operatorname{arccosh}\left[\cosh(\zeta r_s^{\text{source}})\cosh(\zeta r_t^{\text{target}}) - \sinh(\zeta r_s^{\text{source}})\sinh(\zeta r_t^{\text{target}})\cos(\theta_{s\to t})\right]$.

on the half-line for the same network from its representations of different geometries. In the case of an inner product-based Euclidean embedding, the most attractive radial coordinate is the largest one, $r_{\mathrm{Euc,max}}$, whereas the most attractive radial coordinate in a hyperbolic embedding is the smallest one, $r_{\mathrm{hyp,min}}$. The equivalents of these coordinates on the half-line are the positions given by $r_{\mathrm{line}}(r_{\mathrm{Euc,max}})$ and $r_{\mathrm{line}}(r_{\mathrm{hyp,min}})$. Thus, MIC derives the required hyperbolic embedding's mapping on the half-line from the formula

$$\frac{r_{\mathrm{line}}(r_{\mathrm{Euc,max}})}{r_{\mathrm{line}}(r_{\mathrm{Euc},i})} = \frac{r_{\mathrm{line}}(r_{\mathrm{hyp},i})}{r_{\mathrm{line}}(r_{\mathrm{hyp,min}})}. \tag{4.1.2.5}$$

Using Eqs. (4.1.2.3) and (4.1.2.4), this leads to the equation[3]

$$\frac{r^d_{\mathrm{Euc,max}}}{r^d_{\mathrm{Euc},i}} = \frac{e^{\zeta\cdot(d-1)\cdot r_{\mathrm{hyp},i}} - 1}{e^{\zeta\cdot(d-1)\cdot r_{\mathrm{hyp,min}}} - 1}, \tag{4.1.2.6}$$

where both sides are larger than or equal to 1.

To enable the conversion of the radial coordinates of a Euclidean embedding to the radial coordinates of a hyperbolic embedding based on the equivalence of the ratios written in Eq. (4.1.2.6), the hyperbolic radial position of one node has to be fixed beforehand. As this choice inherently determines the extent of the hyperbolic network layout, it is reasonable to treat the largest hyperbolic radial coordinate as an adjustable parameter of the Euclidean-hyperbolic conversion. In order to make the majority of the hyperbolic radial coordinates large enough to justify the approximating formula of the hyperbolic distance given by Eq. (4.1.2.2) – which served as the basis for the assumption that the contribution of the radial and the angular node positions can be similarly well separated from each other in the case of the hyperbolic distance as in the case of the inner product in the Euclidean embeddings, meaning that the radial node arrangement can be transformed independently of the angular node arrangement during the Euclidean-hyperbolic conversion –, MIC sets the largest hyperbolic value of both the source and the target radial coordinates to

$$r_{\mathrm{hyp,max}} = \frac{C}{\zeta}\cdot\ln(N) \tag{4.1.2.7}$$

with $C = 2$, setting the area of the hyperbolic disk occupied by a two-dimensional embedding of a network of N number of nodes equal to the area used by the PSO model [3] when generating a network having the same number of nodes (see Sect. 2.1). Note that according to the measurement presented in Supplementary Note 4 of Ref. [T4], increasing the parameter C of MIC does not seem to improve the embedding quality.

After choosing an $r_{\mathrm{hyp,max}}$ value, the most attractive hyperbolic radial coordinate $r_{\mathrm{hyp,min}}$ can be determined by substituting the least attractive radial positions $r_{\mathrm{Euc,min}}$ and $r_{\mathrm{hyp,max}}$ in Eq. (4.1.2.6) as

$$\frac{r^d_{\mathrm{Euc,max}}}{r^d_{\mathrm{Euc,min}}} = \frac{e^{\zeta\cdot(d-1)\cdot r_{\mathrm{hyp,max}}} - 1}{e^{\zeta\cdot(d-1)\cdot r_{\mathrm{hyp,min}}} - 1}. \tag{4.1.2.8}$$

[3] It is assumed in Eq. (4.1.2.6) that the occurring smallest radial coordinates $r_{\mathrm{Euc,min}}$ and $r_{\mathrm{hyp,min}}$ are larger than 0, providing that each node has in both the Euclidean and the hyperbolic embedding a definite angular position that can be retained when switching between the different geometries. If, however, due to numerical errors $r_{\mathrm{Euc,min}} = 0$ was obtained from a Euclidean embedding method, I first executed the Euclidean-hyperbolic conversion only for the nodes of nonzero $r_{\mathrm{Euc},i}$ values, and then I placed the nodes that fell in the origin in the Euclidean embedding to an extremely large radial coordinate (namely, 10 times larger than the largest hyperbolic radial coordinate achieved among the nodes that were not in the origin in the Euclidean case) and a random angular position in the hyperbolic space.

Using Eq. (4.1.2.7), the smallest hyperbolic radial coordinate $r_{\text{hyp,min}}$ can be expressed from Eq. (4.1.2.8) as

$$r_{\text{hyp,min}} = \frac{1}{\zeta \cdot (d-1)} \cdot \ln\left(1 + [e^{\zeta \cdot (d-1) \cdot r_{\text{hyp,max}}} - 1] \cdot \left[\frac{r_{\text{Euc,min}}}{r_{\text{Euc,max}}}\right]^d\right) = \\ = \frac{1}{\zeta \cdot (d-1)} \cdot \ln\left(1 + [N^{C \cdot (d-1)} - 1] \cdot \left[\frac{r_{\text{Euc,min}}}{r_{\text{Euc,max}}}\right]^d\right). \tag{4.1.2.9}$$

Finally, given $r_{\text{hyp,min}}$, the hyperbolic radial coordinate $r_{\text{hyp},i}$ of any node i can be calculated from the corresponding Euclidean radial coordinate $r_{\text{Euc},i}$ based on Eq. (4.1.2.6) as

$$r_{\text{hyp},i}(r_{\text{Euc},i}) = \frac{1}{\zeta \cdot (d-1)} \cdot \ln\left(1 + [e^{\zeta \cdot (d-1) \cdot r_{\text{hyp,min}}} - 1] \cdot \left[\frac{r_{\text{Euc,max}}}{r_{\text{Euc},i}}\right]^d\right) = \\ = \frac{1}{\zeta \cdot (d-1)} \cdot \ln\left(1 + [N^{C \cdot (d-1)} - 1] \cdot \left[\frac{r_{\text{Euc,min}}}{r_{\text{Euc},i}}\right]^d\right), \tag{4.1.2.10}$$

fulfilling the expectations that $r_{\text{hyp}}(r_{\text{Euc,max}}) = r_{\text{hyp,min}}$ and $r_{\text{hyp}}(r_{\text{Euc,min}}) = r_{\text{hyp,max}}$. An important remark is that although the occurring largest source and target radial coordinate in the hyperbolic embedding were set to the same value $r_{\text{hyp,max}}$, due to the possible differences emerging in the radial coordinates of the nodes' source and target representation in the Euclidean embedding, the hyperbolic source and target radial coordinates other than the largest one are not restricted to be the same. Thus, the hyperbolic node arrangements obtained using a single value of $r_{\text{hyp,max}}$ can still have the ability to capture the differences between the radial attractivity relations of the nodes as sources and as targets. Besides, note that the curvature $K = -\zeta^2$ of the hyperbolic space appears only through a multiplier of $1/\zeta$ in the radial coordinate formula given by Eq. (4.1.2.10), which becomes eliminated by the ζ multipliers in the hyperbolic law of cosines written as

$$x_{s \to t} = \frac{1}{\zeta} \text{arccosh}[\cosh(\zeta r_s^{\text{source}}) \cosh(\zeta r_t^{\text{target}}) - \sinh(\zeta r_s^{\text{source}}) \sinh(\zeta r_t^{\text{target}}) \cos(\theta_{s \to t})], \tag{4.1.2.11}$$

reducing the effect of changing the curvature of the hyperbolic space to a simple rescaling of all the hyperbolic distances. Thus, adjusting the curvature $K = -\zeta^2$ of the hyperbolic space does not have an impact on the distance-based ordering of the different node pairs (or, on the order of connection probabilities) in the case of hyperbolic embeddings created by MIC.

To illustrate the operation of MIC, Fig. 4.1.2.1 shows two-dimensional embeddings of an undirected[4] network generated by the E-PSO model [4]. According to the E-PSO model's growth process described in Sect. 2.2, having the highest radial attractivity from the point of view of the minimization of the hyperbolic distances, the early-appearing, innermost nodes collect the highest number of links on the native disk. In the displayed Euclidean embeddings that represent small topological distances as large inner products, these nodes become placed in the outermost positions, as the radial attractivity of the nodes increases outwards in this case. However, as expected, the transformation of the Euclidean layouts into hyperbolic ones transfers the largest hubs back to the innermost positions on the hyperbolic disk and places the large number of radially unattractive nodes (which are gathered around the origin on the Euclidean plane) in the outer regions, providing layouts that are more pleasant to the human eye compared to their Euclidean counterparts. Moreover, the Euclidean embeddings – and thus, also the hyperbolic

[4] As it will be explained in Sect. 4.3.4, in the case of undirected networks (or at least when the connections are symmetric in a network), HOPE, TREXPEN and their variants yield only one position vector for each network node (i.e., the produced source and target coordinate matrices are identical, $\boldsymbol{S} = \boldsymbol{T}$).

ones – seem to preserve the angular node arrangement yielded by the E-PSO model relatively well, reflecting the common preference of both geometries towards the relatively small angular distances of the connected pairs. A more quantitative evidence of the great performance of MIC is provided by Sect. S2.3 of Supplementary Note 2 of Ref. [T4], where some simple measurements demonstrate that MIC can outperform the d-dimensional extension (Sect. 3.2.2) of the widely used [5, 6, 55] PSO-based transformation of the Euclidean radial coordinates even on such hyperbolic networks that were generated by the PSO model. These investigations even cover a directed version of the E-PSO model and the dPSO-based Euclidean-hyperbolic conversion, which were introduced in Supplementary Note 2 of Ref. [T4] too.

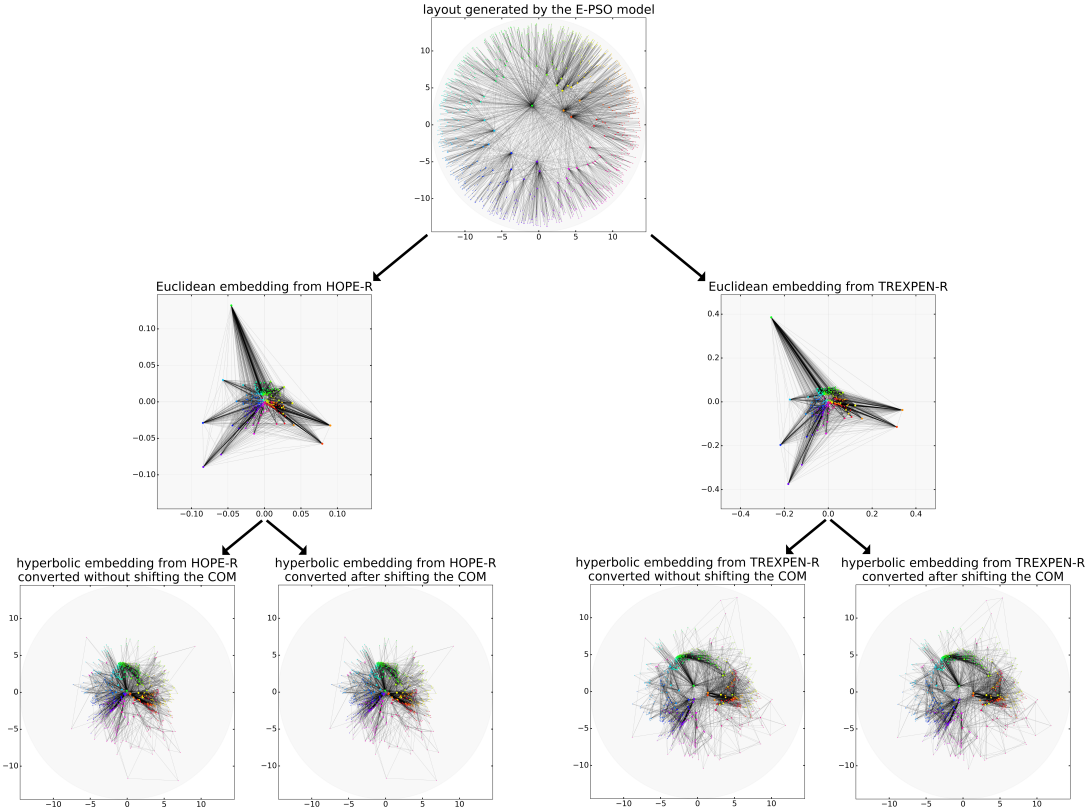

FIGURE 4.1.2.1: **The operation of the proposed model-independent Euclidean-hyperbolic embedding conversion method MIC.** The uppermost layout shows an undirected network generated in the native representation of the hyperbolic plane by the E-PSO model. Inputting the generated edge list to the Euclidean embedding methods HOPE-R and TREXPEN-R (setting the number of embedding dimensions d to 2) resulted in the layouts depicted in the middle row. The lowermost layouts were created from these Euclidean node arrangements by MIC. The result of performing the optional step of shifting the center of mass (COM) of the Euclidean node arrangement before converting it to a hyperbolic one is also shown here. The node degrees are indicated by the node sizes: nodes with larger number of connections are depicted by larger markers. The nodes are colored in each layout according to the angular coordinates originally assigned to the network nodes on the hyperbolic plane by the E-PSO model. The parameters of the E-PSO model were set to the following values for the network generation: the curvature of the hyperbolic plane was $K = -1$ (from $\zeta = 1$), the total number of nodes was $N = 1000$, the number of external links that emerged in each time step was $m = 3$, the net number of added and removed internal links per time step was $L = 2$ (yielding an average degree $\langle k \rangle \approx 2 \cdot (m + L) = 10$), the popularity fading parameter was $\beta = 0.8$ (corresponding to the decay exponent $\gamma = 1 + 1/\beta = 2.25$ of the tail $\mathcal{P}(k) \sim k^{-\gamma}$ of the degree distribution), and the temperature was $T = 0$ (resulting in an average clustering coefficient of $\bar{c} = 0.806$). The depicted HOPE-R embeddings were created using $\alpha = 5.97 \cdot 10^{-3}$, while the TREXPEN-R layouts were obtained at $q = 3.89$. The parameter C of MIC was set to 2 for all the hyperbolic embeddings. The figure was taken from Ref. [T4].

4.2 Embedding directly in the hyperbolic space

Section 4.1 detailed how a hyperbolic embedding of a directed network can be obtained through the conversion of a Euclidean node arrangement. However, the need for a Euclidean embedding that serves as a good basis for a hyperbolic one and the dependence on the ability of the Euclidean geometry to capture the network topology can be eliminated from the process of hyperbolic embedding if the network nodes are placed directly in the hyperbolic space. As it was described in Sect. 3.3, a method named hydra [10] has already provided a solution to this task in the case of undirected networks, mapping the topological distances to hyperbolic ones through the optimization of Lorentz products (see Eq. (1.3.2)) in the hyperboloid representation of the hyperbolic space (see Sect. 1.3). Inspired by this algorithm, I introduced in Ref. [T4] the method named TRansformation of EXponential shortest Path lengths to hyperbolIC measures (TREXPIC), which, contrary to hydra, is able to embed directed networks in the hyperbolic space.

Just like hydra, TREXPIC interprets the elements of a distance (or dissimilarity) matrix $\boldsymbol{D}$ derived from the network topology as the estimations of the pairwise hyperbolic distances between the network nodes and tries to find the hyperbolic node arrangement that reproduces these expected hyperbolic distances. This is done in both methods via a decomposition of the matrix of expected Lorentz products, which, according to Eq. (1.3.1), can be formed from the matrix of expected hyperbolic distances $\boldsymbol{D}$ in the hyperboloid representation of the hyperbolic space of curvature $K = -\zeta^2$ as

$$\boldsymbol{L} = \cosh(\zeta \cdot \boldsymbol{D}). \tag{4.2.1}$$

While hydra [10] simply uses the matrix of shortest path lengths as the distance matrix $\boldsymbol{D}$, TREXPIC measures the topological (or expected hyperbolic) distance of node t from node s as

$$D_{st} = e^{-\frac{q}{\mathrm{SPL}_{s\to t}}}, \tag{4.2.2}$$

where $\mathrm{SPL}_{s\to t}$ is the hop-length of the shortest path from node s to node t and $q > 0$ is a tunable parameter that regulates how fast the above-defined exponential distance measure increases towards the larger shortest path lengths[5]. These exponential distances fall in the interval $[0, 1]$, where 0 is the distance of each node from itself (occurring in the diagonal of $\boldsymbol{D}$) and the value 1 corresponds to $\mathrm{SPL}_{s\to t} = \infty$. Mapping infinite distances – indicating that there are no paths from a given node to an other – to finite matrix elements enables TREXPIC to embed also weakly connected networks besides strongly connected ones.

The reason behind switching from the matrix of expected hyperbolic distances to the matrix of expected Lorentz products is that, based on the definition of the Lorentz product given by Eq. (1.3.2), the matrix of the pairwise Lorentz products between the nodes' position vectors in the hyperboloid representation of the d-dimensional hyperbolic space can be written as a simple product of the node coordinate matrices, namely as

$$\boldsymbol{L} = \boldsymbol{S} \cdot \boldsymbol{J} \cdot \boldsymbol{T}^{\mathrm{T}}, \tag{4.2.3}$$

where the source and the target position vectors of the nodes of a directed network are given by the rows of the matrices $\boldsymbol{S}$ and $\boldsymbol{T}$, respectively, and $\boldsymbol{J}$ is the $(d+1) \times (d+1)$-sized diagonal matrix containing the values $+1, -1, -1, ..., -1$ in the diagonal. To find a similar, product form of the matrix $\boldsymbol{L}$, hydra performs its eigendecomposition $\boldsymbol{L} = \boldsymbol{Q} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{Q}^{-1}$. In contrast, TREXPIC builts on the singular value decomposition (SVD)

$$\boldsymbol{L} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^{\mathrm{T}}, \tag{4.2.4}$$

[5]For small enough values of the multiplying factor q, $e^{-\frac{q}{\mathrm{SPL}_{s\to t}}} \approx 1 - q/\mathrm{SPL}_{s\to t}$, and the increase in q shifts the non-unit off-diagonal matrix elements towards 0.

where the non-trivial connection between the two non-diagonal matrices ($\boldsymbol{U}$ and $\boldsymbol{V}$) in the case of asymmetric $\boldsymbol{L}$ matrices gives more freedom for the emergence of differences between the source and the target position vectors of the network nodes. To reproduce the Lorentz products given in $\boldsymbol{L}$ – as much as possible in a d-dimensional embedding –, the matrix of source position vectors is composed in TREXPIC based on Eqs. (4.2.3) and (4.2.4) as

$$\boldsymbol{S} = [+\sqrt{\sigma_1}\cdot\underline{u}_1\,,\,-\sqrt{\sigma_2}\cdot\underline{u}_2\,,\,-\sqrt{\sigma_3}\cdot\underline{u}_3\,,\,\ldots\,,\,-\sqrt{\sigma_{d+1}}\cdot\underline{u}_{d+1}]\,, \tag{4.2.5}$$

and the matrix of target position vectors is identified as

$$\boldsymbol{T} = [+\sqrt{\sigma_1}\cdot\underline{v}_1\,,\,+\sqrt{\sigma_2}\cdot\underline{v}_2\,,\,+\sqrt{\sigma_3}\cdot\underline{v}_3\,,\,\ldots\,,\,+\sqrt{\sigma_{d+1}}\cdot\underline{v}_{d+1}], \tag{4.2.6}$$

where $\sigma_i = \Sigma_{jj}$ is the jth value in the descending order of the (always non-negative) singular values, while $\underline{u}_j$ and $\underline{v}_j$ denote the jth column of $\boldsymbol{U}$ and $\boldsymbol{V}$, i.e., the jth one of the left and the right singular vectors, respectively. To select the upper sheet of the two-sheet hyperboloid depicted in Fig. 1.3.1, the non-negative version of the leading singular vectors $\underline{u}_1$ and $\underline{v}_1$ must be used, the existence of which is ensured by the Perron–Frobenius theorem, stating for the non-negative matrices $\boldsymbol{L}\cdot\boldsymbol{L}^{\mathrm{T}}$ and $\boldsymbol{L}^{\mathrm{T}}\cdot\boldsymbol{L}$ that their leading eigenvector – i.e., $\underline{u}_1$ and $\underline{v}_1$, respectively – can be chosen to have only strictly positive components. Note that just like in the case of the Euclidean embedding methods described in Sect. 4.1.1, the rows of the source coordinate matrix $\boldsymbol{S}$ that correspond to nodes with 0 out-degree and the rows of the target coordinate matrix $\boldsymbol{T}$ that correspond to nodes with 0 in-degree can not be meaningful in the absence of any knowledge about what sort of connections these nodes would prefer to establish, and therefore, TREXPIC removes these rows.

After placing the network nodes in the hyperboloid model of the d-dimensional hyperbolic space, hydra eventually creates a hyperbolic embedding in the Poincaré ball representation (see Sect. 1.2) via a transformation of the position vectors that preserves the pairwise hyperbolic distances between the nodes. However, as it is described in Sect. 1.3, the node coordinates given in the hyperboloid can be also easily mapped to the native representation of the d-dimensional hyperbolic space (which is used in the Euclidean-hyperbolic transformation MIC (Sect. 4.1.2) too). Namely, given the position vectors of node i in the hyperboloid (i.e., the ith row of the matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ defined by Eqs. (4.2.5) and (4.2.6)), TREXPIC calculates the source and target position vectors in the native ball according to the formulas[6]

$$r^{\text{source}}_{\text{native},i}\cdot\frac{(S_{i2},S_{i3},\ldots,S_{i(d+1)})}{\|(S_{i2},S_{i3},\ldots,S_{i(d+1)})\|} = \frac{\operatorname{acosh}(S_{i1})}{\zeta}\cdot\frac{(S_{i2},S_{i3},\ldots,S_{i(d+1)})}{\sqrt{S_{i2}^2+S_{i3}^2+\ldots+S_{i(d+1)}^2}} \tag{4.2.7}$$

and

$$r^{\text{target}}_{\text{native},i}\cdot\frac{(T_{i2},T_{i3},\ldots,T_{i(d+1)})}{\|(T_{i2},T_{i3},\ldots,T_{i(d+1)})\|} = \frac{\operatorname{acosh}(T_{i1})}{\zeta}\cdot\frac{(T_{i2},T_{i3},\ldots,T_{i(d+1)})}{\sqrt{T_{i2}^2+T_{i3}^2+\ldots+T_{i(d+1)}^2}}. \tag{4.2.8}$$

Note that in contrast to the hyperbolic embeddings created by the Euclidean-hyperbolic conversion method MIC described in Sect. 4.1.2, in the case of TREXPIC nothing precludes that adjusting the curvature $K=-\zeta^2$ of the embedding space has an effect on the embedding performance since here already the decomposed matrix $\boldsymbol{L}$ given by Eq. (4.2.1) depends on ζ, and thus, also the coordinate matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ are influenced by ζ. Nevertheless, as it is exemplified in Supplementary Note 4 of Ref. [T4], decreasing the curvature K in TREXPIC to $K<-1$ is not necessarily advantageous from the point of view of the embedding performance; therefore,

[6] It is assumed during this conversion that the smallest first coordinates obtained in the hyperboloid (i.e., $\min_{1\le i\le N} S_{i1}$ and $\min_{1\le i\le N} T_{i1}$) are not smaller than 1. If, however, due to numerical errors this condition is not met, TREXPIC simply set the problematic native radial coordinates to $\operatorname{acosh}(1)/\zeta = 0$.

as in many previous hyperbolic embedding methods [4, 6, 7, 23, 56, 57], the default setting in TREXPIC is $K = -1$.

4.3 Evaluation of embedding performance

Following Ref. [T4], this section presents some applications of the embedding methods proposed in Sects. 4.1.1, 4.1.2 and 4.2, and provides thereby a basic comparison between the usability of the different algorithms regarding different aspects. First, some two-dimensional embeddings are shown in Sect. 4.3.1, demonstrating the differences between the angularly restricted and the circularized versions of HOPE and TREXPEN, and the automatic appearance of the communities on the layouts. Then, Sect. 4.3.2 details three tasks regarding which the embedding quality can be quantified in a model-independent manner, avoiding the comparison of the embeddings with the output of any specific, arbitrarily chosen geometric model of network formation. The embedding performance in mapping accuracy, graph reconstruction and greedy routing is evaluated for some real directed networks in Sect. 4.3.3. Then, Sect. 4.3.4 investigates the embeddings of a real undirected network, confirming that the embedding methods proposed in Ref. [T4]) are able to compete with previous, well-known dimension reduction techniques. Finally, the ability of TREXPIC, TREXPEN and its variants to grasp the additional topological information carried by link weights is investigated in Sect. 4.3.5.

As it is shown in Fig. 4.2, the studied embedding methods have a number of adjustable parameters. The parameters of the hyperbolic node arrangements, namely the curvature $K = -\zeta^2$ of the hyperbolic embedding space and the parameter C of the Euclidean-hyperbolic conversion MIC (which determines the maximum of the hyperbolic radial coordinates) were always set to their default values given by $K = -1$ or $\zeta = 1$ (also used e.g. in Refs. [4, 6, 7, 23, 56, 57] and Ref. [T1]) and $C = 2$ (reproducing the network areas yielded by the PSO model [3] in the two-dimensional case). However, regarding the other settings, I always tested several different choices in order to provide the possibility for all the methods to perform at their best:

- In the case of HOPE and its variants, 15 values of α were tested for calculating the Katz proximities, which were sampled between $\alpha_{\min} = \frac{1}{200 \cdot \rho_{\text{spectral}}(\boldsymbol{A})}$ and $\alpha_{\max} = \frac{1}{\rho_{\text{spectral}}(\boldsymbol{A})}$ equidistantly on a logarithmic scale, where $\rho_{\text{spectral}}(\boldsymbol{A})$ denotes the spectral radius of the adjacency matrix $\boldsymbol{A}$, i.e. the largest absolute value of the eigenvalues of $\boldsymbol{A}$. The upper boundary of the examined interval was chosen to enable a faster computation of the Katz matrix (see Eq. (4.3.5.2)) instead of using Eq. (4.1.1.1) that is usually not feasible in practice because of the tremendous amount of possible paths in a network.

- In TREXPEN and its variants, 15 values of q were tested for calculating the exponential proximities, which were sampled between $q_{\min} = \frac{-\ln(0.9)}{\text{SPL}_{\max}}$ and $q_{\max} = \frac{-\ln(10^{-50})}{\text{SPL}_{\max}}$ with $\text{SPL}_{\max}$ denoting the largest finite shortest path length measured along the given graph, varying the occurring smallest non-zero exponential proximity $P_{\text{min,non0}} = e^{-q \cdot \text{SPL}_{\text{max,finite}}}$ between 0.9 and 10^{-50}. To ensure that both small and large q values are sufficiently represented in the sampled set of parameters, 7 of the 15 q values were sampled from the interval $[q_{\min}, q_{\text{mid}})$ equidistantly on a logarithmic scale and 8 were sampled from $[q_{\text{mid}}, q_{\max}]$ equidistantly on a linear scale, where $q_{\text{mid}} = e^{\frac{\ln(q_{\min}) + \ln(q_{\max})}{2}}$ is the logarithmic midpoint of the examined interval $[q_{\min}, q_{\max}]$, i.e., the geometric mean of $q_{\min}$ and $q_{\max}$.

- In the case of TREXPIC, 15 settings of q were tested for calculating the exponential distances, which were sampled between $q_{\min} = \ln\left(\frac{1.0}{0.9999}\right) \cdot \text{SPL}_{\max}$ and $q_{\max} = \ln(10) \cdot \text{SPL}_{\max}$ equidistantly on a logarithmic scale, where $\text{SPL}_{\max}$ denotes the largest finite shortest path length measured along the given graph.

- The tested number of dimensions were $d = 2, 3, 4, 8, ..., 2^n \leq \frac{N}{10}$ ($n \in \mathbb{Z}^+$) for all the embedding methods, where the condition $d \leq N/10$ is intended to ensure a considerable dimension reduction.
- The angularly not restricted HOPE-S, HOPE-R, TREXPEN-S and TREXPEN-R were always tested both with and without shifting the center of mass (COM) of the network to the origin. Note that shifting all the nodes by the same vector does not affect the pairwise (Euclidean or hyperbolic) distances – and thus, it is not reasonable to shift the COM in embeddings that are evaluated based on the node-node distances like TREXPIC –, but modifies the pairwise inner products of the nodes in a Euclidean embedding, and also the hyperbolic node arrangement created by MIC can be changed by shifting the COM of the inputted Euclidean embedding.

The suitability of the applied α, q and d parameter intervals is demonstrated by Supplementary Note 4 of Ref. [T4], showing through the example of the Wikipedia network also examined in Sect. 4.3.3 that the embedding quality typically reaches a maximum within these ranges and declines at the boundaries. Besides, Supplementary Notes 4 and 7 of Ref. [T4] suggest that the Euclidean embeddings are usually hindered by the COM's shift when using the inner product as the indicator of the topological relations, while the hyperbolic embeddings yielded by MIC can benefit from balancing the inputted Euclidean node arrangement. However, for the sake of brevity, the following sections present for each embedding method only the result of the best one among the tested options in the given measurement. It is important to emphasize that I did not try to find the exact optimum of any embedding parameter, and therefore, slight variances between the quality of the different embedding algorithms must be treated with caution as these may simply be the result of the parameter settings being imperfect, and the approach that appears to be worse might prevail over the other at better parameter settings.

4.3.1 Automatic separation of communities: examples of two-dimensional layouts

When representing small topological distances between the network nodes as large Euclidean inner products or small hyperbolic distances, nodes that are close to each other along the graph tend to become gathered within small angular distances as large angular distances are not favorable neither for the maximization of Euclidean inner products nor for the minimization of hyperbolic distances. Thus, the Euclidean and hyperbolic embeddings of Sects. 4.1 and 4.2 are able to outline the groups of nodes of similar topological behavior, i.e., the communities of a network. This section demonstrates how the examined embeddings express the community structure of a network through the angular arrangement of its nodes.

Figures 4.3.1.1 and 4.3.1.2 show two-dimensional embeddings of 2 synthetic directed networks that were generated by the stochastic block model (SBM) [58–60] from 3 blocks of 100 nodes. In Fig. 4.3.1.1, the edge densities between the different blocks were defined as

$$\begin{bmatrix} 0.25 & 0.05 & 0.1 \\ 0.05 & 0.35 & 0.05 \\ 0.1 & 0.15 & 0.4 \end{bmatrix}, \tag{4.3.1.1}$$

while Fig. 4.3.1.2 refers to a network where the edge densities were given by the matrix

$$\begin{bmatrix} 0.05 & 0.25 & 0.15 \\ 0.35 & 0.05 & 0.2 \\ 0.4 & 0.2 & 0.05 \end{bmatrix}. \tag{4.3.1.2}$$

According to Eq. (4.3.1.1), in the first network most of the links emerged within the blocks, yielding an assortative block structure that consists of such groups of the nodes that are more connected to each other than to the nodes of the other communities. On the contrary, in the network defined by Eq. (4.3.1.2) the nodes are connected mostly to nodes of other blocks, creating a disassortative block structure in which the members of a group are held together by their

similar connection preference towards the other groups of the nodes. Despite the fundamental differences between the binding forces of the blocks in the above-described two block structures, HOPE-S, HOPE-R, TREXPEN-S, TREXPEN-R and TREXPIC managed to group the nodes of the 3 blocks into different angular regions in both SBM networks. Meanwhile, as expected, HOPE and TREXPEN placed both SBM networks in a restricted angular range, resulting in a less clear separation of the 3 blocks.

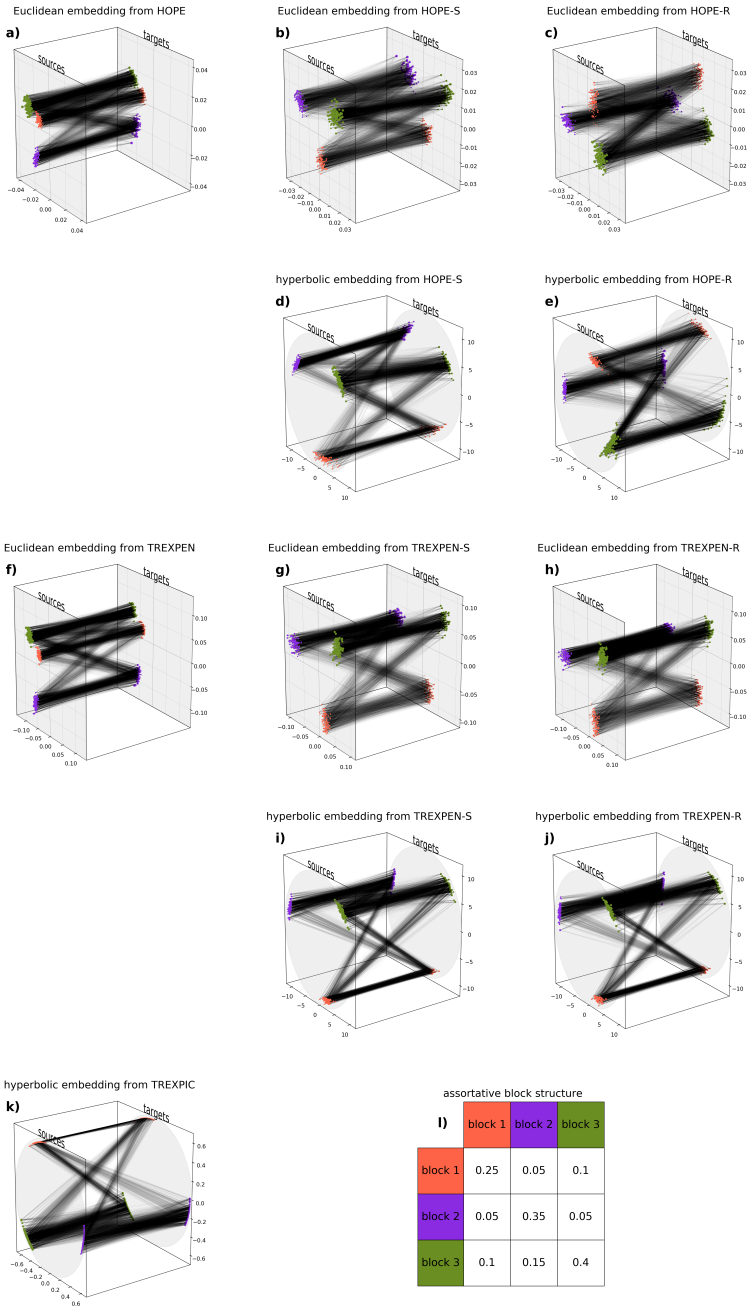

FIGURE 4.3.1.1: **Two-dimensional embeddings of a directed SBM network having an assortative block structure.** The different colors denote the different blocks in which the nodes were classified during the network generation. The node sizes are consistent with the node degrees. In the case of HOPE and its variants, the parameter α was $2.01 \cdot 10^{-3}$. The embeddings with TREXPEN and its variants were obtained at $q = 6.48$. The TREXPIC layout was created with $q = 4.55 \cdot 10^{-2}$. The parameter C of MIC was always set to 2. $\zeta = 1$ (i.e., the curvature $K = -1$) was used for all the hyperbolic embeddings. The figure was taken from Ref. [T4].

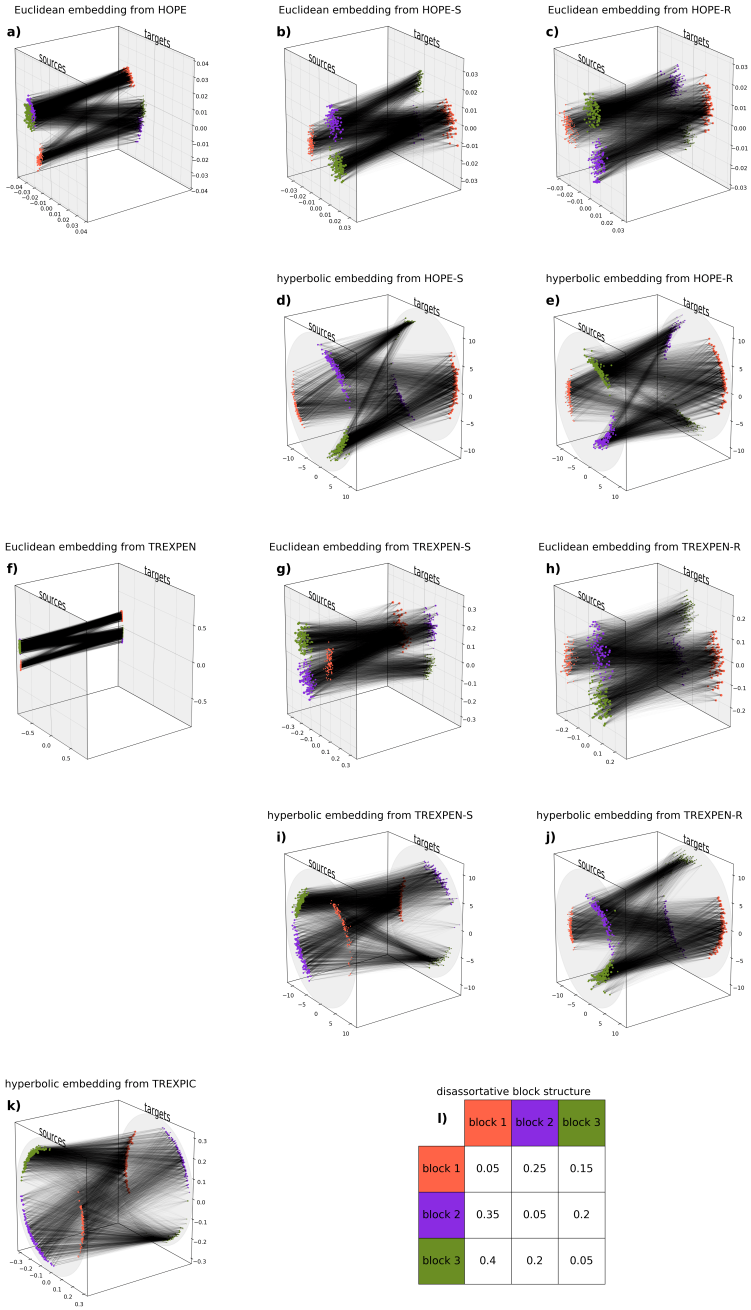

FIGURE 4.3.1.2: **Two-dimensional embeddings of a directed SBM network having a disassortative block structure.** The different colors denote the different blocks in which the nodes were classified during the network generation. The node sizes are consistent with the node degrees. In the case of HOPE and its variants, the parameter α was $1.89 \cdot 10^{-3}$. The embeddings with TREXPEN and its variants were obtained at $q = 0.26$. The TREXPIC layout was created with $q = 1.65$. The parameter C of MIC was always set to 2. $\zeta = 1$ (i.e., the curvature $K = -1$) was used for all the hyperbolic embeddings. The figure was taken from Ref. [T4].

4.3.2 Model-independent measures of embedding quality

This section defines the measures and methods that I used in Ref. [T4] for evaluating the embedding performance of the algorithms described in Sects. 4.1 and 4.2. Just like the embedding methods, all the applied quality measures are model independent, not assuming any specific model as the generator of the embedded network.

The first technique that I applied to evaluate an embedding is the calculation of the mapping accuracy $\mathrm{ACC_m} \in [-1,+1]$ that was defined in Ref. [61] as the Spearman's correlation coefficient [62] between the shortest path lengths of an undirected network and the pairwise distances between the network nodes in the embedding space. Considering that the Euclidean embedding methods described in Sect. 4.1.1 optimize for inner products, it is straightforward to evaluate them with regard to not only the Euclidean distances but also the inner products. Thus, Sects. 4.3.3 and 4.3.4 present mapping accuracies calculated for the hyperbolic distance in the case of hyperbolic node arrangements, and both the Euclidean distance and the additive inverse of the inner product in the case of the Euclidean embeddings. The expectation is that the examined embedding methods minimize the (either Euclidean or hyperbolic) distance and/or maximize the inner product between the positions of such nodes that are close to each other along the network, meaning that higher positive values of the mapping accuracy correspond to higher embedding qualities.

Naturally, in directed networks I had to take into account the directedness of the paths and compare the hop-length of the shortest path from node s to node t to the geometric distance or the inner product measured between the source position vector of node s and the target position vector of node t. It is important to note that when embedding not strongly, but only weakly connected networks, the node pairs for which the given graph does not contain any connecting paths can not be included in the calculation of the mapping accuracy, as for such node pairs the length of the shortest path is infinity. Besides, I also disregarded the pairing of each node with itself since the relation between the target and the source position of the same node does not influence the embedding quality as the reachability of a node from itself is an obvious property of any network topology and, therefore, the focus of an embedding is on the relations of a node's two representations with the other nodes and not on the adjustment of the source and the target positions of the same node compared to each other. Note that when all the proper start-destination node pairs of a network are considered, the calculation of the mapping accuracy is deterministic, and thus, there is no need for the repetition of its computation for an embedding.

Besides mapping accuracy, another model-independent technique for evaluating embedding is given by the graph reconstruction task (also considered e.g. in Refs. [39, 63]) where the aim is the differentiation between the connected and the unconnected node pairs of a network based on pairwise geometric measures calculated for the embedding of the given network[7]. Namely, the studied embeddings can be considered to be of high quality if the arrangement of the node pairs in the increasing order of the Euclidean distance, the additive inverse of the inner product or the hyperbolic distance tends to sort the links of the given network to the beginning of the node pair order (i.e. below a given threshold of the applied geometric measure) while ranking the unconnected node pairs rather at the end of the node pair order. The order between

[7] It is important to emphasize that the aim of graph reconstruction is to differentiate between such connected and unconnected node pairs that all have been inputted to the embedding algorithm. In contrast, in the so-called link prediction task the embedding method gets a pruned graph as an input and the aim is to find those unconnected node pairs in this input that were connected in the original graph. Since the difficulty level of link prediction strongly depends on the task's several tunable details (like the number of deleted links or the distribution of the deletion probability), testing graph reconstruction is more straightforward than testing link prediction, and therefore, I opted primarily for the former one. Nevertheless, as an example, for the undirected test network presented in Sect. 4.3.4, I carried out some measurements regarding both tasks, as shown by Figs. 4.3.4.3 and 4.3.4.4.

node pairs that have the same value of the given proxy of connection probability was set randomly in every case. Nevertheless, since – at proper settings of the embedding parameters – it is very rare that the same value of the given geometric measure (i.e. the same connection probability) becomes assigned to more than one node pair yielding an indefinite ordering between them, the graph reconstruction is rather deterministic. Therefore, it is not reasonable to repeat the evaluation of an embedding regarding graph reconstruction when considering all the possible source-target node pairs of a network – leaving out the pairing of each node with itself (since self-loops are disregarded by the embeddings) and those node pairs in which the out-degree of the source node or the in-degree of the target node is 0 (since to a node with 0 out- or in-degree no position is assigned by the embedding methods as source or target, respectively).

As a baseline, the graph reconstruction performance of so-called local methods can be used that, contrary to the embeddings, do not use the whole graph to estimate the connection probability of a given node pair but rank the node pairs according to such measures that depend solely on the immediate neighborhood of the two nodes in question. To be specific, higher connection probabilities are associated in Sects. 4.3.3 and 4.3.4 with higher numbers of common neighbors [64], higher node degrees (like in preferential attachment [65]) or higher values of 3 directed variations [T4] of the originally undirected resource allocation index [66] – however, for the sake of simplicity, the figures of these sections dealing with graph reconstruction always indicate only the best result obtained among the tested (altogether 5) local methods.

The graph reconstruction performance is usually characterized by 3 measures [57] that increases towards the higher embedding qualities:

- The notation Prec $\in [0,1]$ (abbreviating precision) stands for the proportion of correct guesses when treating the number of links $E_{\text{toReconst}}$ that have to be reconstructed as a known input or, in other words, the proportion of actual links among the first $E_{\text{toReconst}}$ number of node pairs in the decreasing order of the given connection probability measure. For the random predictor, Prec was calculated for each network as the ratio between the number of actual links and all the node pairs in the examined set of node pairs.

- The precision-recall (PR) curve [67] describes the proportion of the actual links among all the node pairs that become labeled as connected as a function of the proportion of the links that are successfully identified among all the links that have to be reconstructed. Moving between the different points of the PR curve corresponds to changing the threshold value of the given measure of connection probability, i.e., shifting the point in the node pair order separating the node pairs that become labeled as connected from those that become labeled as unconnected. In Sects. 4.3.3 and 4.3.4, the area [68] under the PR curve [69] AUPR $\in (0,1]$ is calculated, which gives an overall description of the graph reconstruction performances obtained at the different thresholds. In the case of the random predictor, the precision-recall curve is a horizontal line at the precision value given by the ratio between the number of actual links and all the node pairs in the examined set, yielding an AUPR equal to this constant precision value.

- The receiver operating characteristic (ROC) curve [70] describes the proportion of the links that are successfully identified among all the links that have to be reconstructed as a function of the proportion of the actually unconnected node pairs that become labeled as connected. The different points of the ROC curve are obtained using different threshold values of the given measure associated with the connection probability. In Sects. 4.3.3 and 4.3.4, the area [68] under the ROC curve AUROC $\in [0,1]$ is measured, summarizing this curve in a single number, which corresponds to the probability that a randomly chosen connected node pair gets ranked over a randomly chosen unconnected node pair in the order of the examined connection probability measure [1, 71]. For the random predictor, the ROC curve is a straight line between the points $(0,0)$ and $(1,1)$ with AUROC $= 0.5$.

In addition to mapping accuracy and graph reconstruction, I also studied the embeddings' quality from the point of view of the navigability of the embedded networks, examining the efficiency of greedy routing [43–45] on them, where the aim is to walk along the links of the network from a starting node s to a destination node t using the possible least number of steps, making each step based only on local information given by the spatial position of the current neighbors (i.e., the endpoints of the links that spring from the current node) compared to the position of the destination node in the embedding. Although the greedy routing is usually performed based on geometric (either Euclidean or hyperbolic) distances, I adopted a rather general stepping rule, and in the case of Euclidean embeddings, I tested both the minimization of the Euclidean distance and the maximization of the inner product between the target position of the neighboring nodes and the target position of the destination node. Returning to a node that has already been visited in the current walk indicates that the walk between the given pair of starting and destination nodes can not be accomplished in a greedy way. Note that – at proper settings of the embedding parameters – it is very rare that two or more neighboring nodes have the exact same geometric relation with the destination and the greedy router has to choose randomly between them. Therefore, the greedy routing is rather deterministic, and thus, with the consideration of all the possible node pairs, it is sufficient to carry out greedy routing only once for an embedding.

In the following sections, the applied measures of the performance in greedy routing are the following:

- The average hop-length of the successful greedy routes that reached the destination node and have not stopped at any other node. Smaller values mean higher embedding qualities.
- The fraction of successful greedy walks. Higher values correspond to more navigable node arrangements.
- The greedy routing score GR-score $\in [0,1]$, a higher value of which indicates a better navigability of the embedding due to a larger success rate in reaching the destination node and/or smaller hop-lengths of the successful greedy routes. For undirected networks its definition is given by Eq. (3.2.1.4) [6], while for directed networks I defined the greedy routing score in Ref. [T4] as

$$\text{GR-score} = \frac{1}{N_{\text{paths}}} \cdot \sum_{s \in S} \sum_{t \in T_s} \frac{\text{SPL}_{s \to t}}{\text{GRPL}_{s \to t}}, \tag{4.3.2.1}$$

  where $\text{SPL}_{s \to t}$ stands for the hop-length of the shortest path from node s to another node t – which is infinity if there is no path in the graph leading from s to t –, and $\text{GRPL}_{s \to t}$ denotes the greedy routing hop-length between the same pair of starting and destination nodes – which is set to infinity if the routing fails to reach node t from node s. To allow the investigation of weakly connected networks where not all the nodes are reachable from every node, I always took into account only those starting node-destination node pairs that are connected by at least one path in the graph, i.e., for which the greedy routing is at least theoretically possible. Therefore, the total number N_{paths} of the examined start-destination pairs can be smaller than $N \cdot (N-1)$, and the summations in Eq. (4.3.2.1) go over only the nodes that function as a source of links in the network, i.e. the nodes of non-zero out-degree (contained by the set S) and the destinations to which leads at least one directed path from node s (contained by the set T_s for a given starting node s, not including node s).

When evaluating embeddings of networks of moderate sizes, I considered all the proper node pairs in all the tasks. However, to keep the computational cost of the evaluation processes within reasonable limits even in the case of large networks, I maximized the number of

examined node pairs in 500000. When not using all the proper node pairs, I created 5 random samples of them and repeated the calculation of all the quality measures 5 times for each embedding. The random samples consisted of 500000 number of start-destination node pairs in the case of mapping accuracy and greedy routing. For graph reconstruction, I set the number of links $E_{\text{toReconst}}$ in each sample low enough to ensure that the total size of the sample (i.e. the sum of the number of links and the number of unconnected node pairs in the sample) remains under 500000. In this task, in order to obtain unbiased samples that well represent the total dataset, it is important to set the ratio between the number of sampled links and the total number of sampled node pairs equal to the ratio between the total number of links and the total number of proper node pairs in the network [72, 73]. Therefore, I created the examined set of node pairs for graph reconstruction by randomly sampling $E_{\text{toReconst}}$ number of connected node pairs and then the corresponding number of unconnected node pairs.

4.3.3 Embedding real directed networks

Following Ref. [T4], this section evaluates the performance of the embedding methods described in Sects. 4.1 and 4.2 from the point of view of mapping accuracy, graph reconstruction and greedy routing on the following directed real networks:

- A subnetwork of $N = 505$ number of nodes and $E = 2081$ number of edges extracted from Wikipedia's norm network of 2015 [74], where Wikipedia pages are connected to each other with directed edges that correspond to hyperlinks. I created the subgraph by omitting all nodes for which the highest value of the topic distribution does not reach 80%, i.e. I kept only those pages for which the topic was not too uncertain.

- The transcriptional regulation network [75, 76] of the yeast *Saccharomyces cerevisiae*, describing $E = 1063$ number of transcription factor-based interactions between $N = 662$ number of regulatory proteins and genes. The links point from the regulating objects toward the regulated ones. The mode of regulation was considered to be the same in each case, i.e. I did not differentiate between activators and repressors.

- A network [50, 51] of $E = 19021$ hyperlinks among $N = 1222$ number of U.S. political weblogs from before the 2004 presidential election. The blogs are characterized by their political leaning, forming 2 groups: left/liberal and right/conservative.

- A word association network [77, 78] containing $N = 4865$ number of nodes and $E = 41964$ number of links that point from the cue words toward the associated words.

The above-listed N and E values refer to the size of the largest weakly connected component (WCC) of each graph. Since nodes with no connections cannot be represented in the embedding (do not have any role in the network neither as a source nor as a target of links), only the largest weakly connected component was inputted to the embedding methods in each case. Note that throughout this section, the link weights given in some of the datasets were discarded and I assigned the weight 1 to all the edges, and the embedding of weighted networks is discussed in Sect. 4.3.5.

Regarding the embedding parameters, I used in all the measurements of this section the usual test set described in the introduction of Sect. 4.3. Because of the relatively high importance of two-dimensional cases – which are the only ones besides the three-dimensional embeddings that produce directly visualizable node arrangements –, the following figures always depict the best two-dimensional performances achieved using the different geometric measures of topological proximity, although, of course, the node arrangements obtained in higher-dimensional spaces can capture more information precisely, yielding better embedding qualities. For the two smaller graphs (i.e. the network of Wikipedia pages and the yeast transcription network),

I took into consideration all the possible node pairs in all the examined tasks, but in the case of the two larger graphs (namely the network of political blogs and the word association network), I evaluated the embedding performance only on sets of node pairs sampled according to Sect. 4.3.2 because of the high computational intensity.

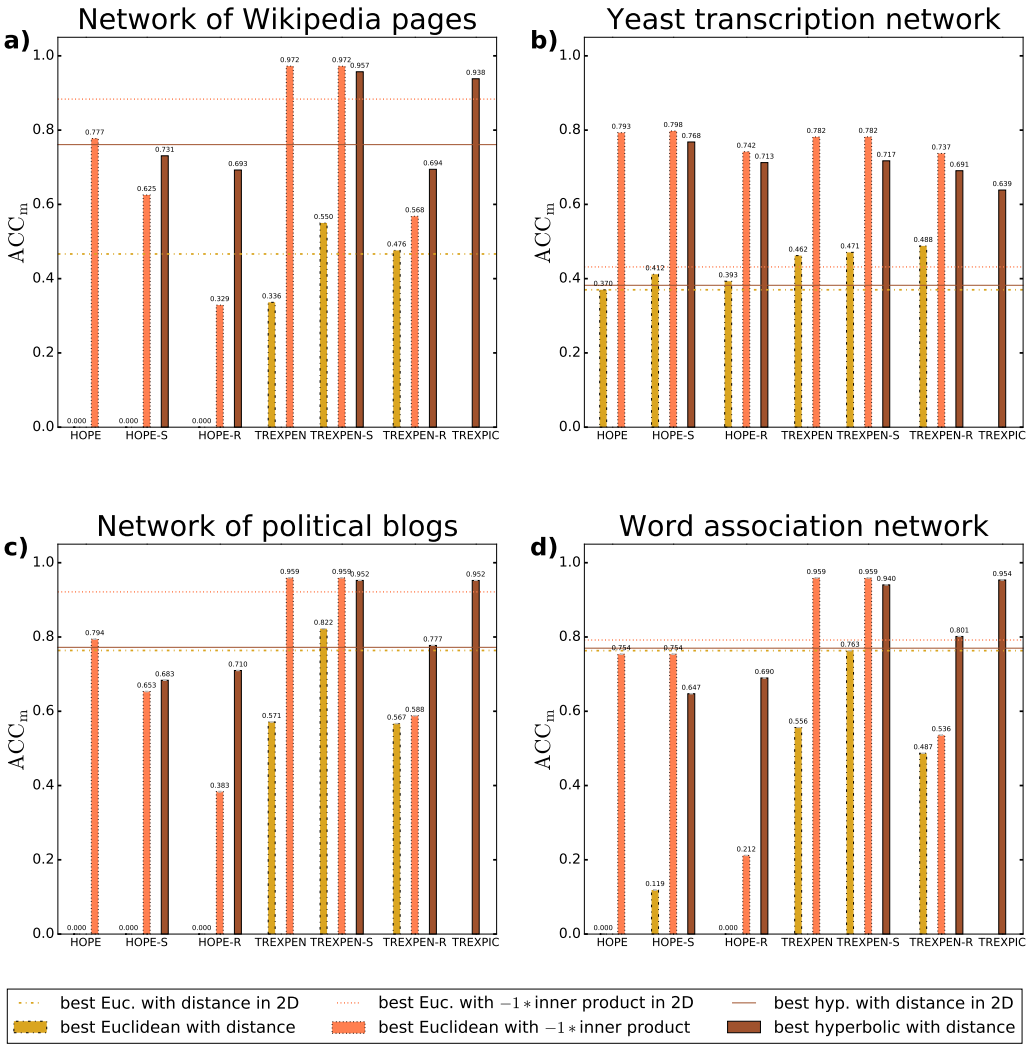

FIGURE 4.3.3.1: **Mapping accuracy on directed real networks.** Each panel refers to a real network named in the title of the panel. For the networks in panels a) and b), the mapping accuracy was measured examining each node pair connected by at least one directed path, whereas for the larger networks in panels c) and d), the mapping accuracy was measured on 5 samples of 500000 node pairs connected by at least one directed path. In the case of the larger networks, I always considered the average of the performances over the 5 samples and depicted the corresponding standard deviations with (usually very small) grey error bars. The colors indicate the used geometric measure, as listed in the common legend at the bottom of the figure. Only the best results are plotted in each panel, obtained with the parameter setting that yielded the highest values of the mapping accuracy. Note that the 0 values denote that the given methods have not achieved any positive value. The bars were created considering all the tested numbers of dimensions, whereas the horizontal lines show the best two-dimensional performances achieved among all the embedding methods. The figure was taken from Ref. [T4].

First, Fig. 4.3.3.1 presents the mapping accuracy on the above-described four networks. As expected, TREXPEN, its variants and TREXPIC yield higher correlations between the shortest path lengths and the geometric measures compared to HOPE and its variants in most of

the cases since HOPE considers all the paths between the nodes to a certain extent, not only the shortest ones. The best overall results were produced by Euclidean embeddings, but the hyperbolic methods do not fall behind much and, in the meantime, typically prevail over the Euclidean node arrangements when considering the distances between the nodes instead of the inner products.

Second, Fig. 4.3.3.2 shows the embedding quality with respect to the graph reconstruction task of the examined four networks. The usage of Katz proximities (in HOPE and its variants) and the exponential proximities (in TREXPEN and its variants) or exponential distances (in TREXPIC) both seem to be expedient in this task. While generally the inner product in the Euclidean embeddings seems to be the best proxy for the connection probability, in the network of political blogs, with regard to the area under the PR curve (Fig. 4.3.3.2h) the best method in the two-dimensional case is a hyperbolic one. Furthermore, when focusing on the distance-based representations of the network topology, the hyperbolic embeddings clearly outperform the Euclidean ones that often even struggle to surpass the performance of the local methods.

And finally, Fig. 4.3.3.3 depicts the achieved greedy routing scores with the corresponding success rates and average hop-lengths for the examined starting node-destination node pairs in the four real networks in question. For all of these networks, the best GR-scores are achieved in the hyperbolic space; however, the distance-based routing performed in the Euclidean space is usually also effective. The inner product generally does not seem to be well usable for navigating on networks in the Euclidean space. Besides, in this task HOPE and its variants clearly fall behind the methods that we introduced here building on exponential proximities or distances instead of Katz proximities.

An important conclusion is that in the above-presented measurements regarding the mapping accuracy, the graph reconstruction performance and the greedy navigability, the hyperbolic distance was the only geometric measure based on which relatively good quality scores have been achieved in all the different tasks. Among the examined three indicators of topological proximity, the Euclidean distance performed the worst in mapping accuracy and especially in graph reconstruction, where it was often outperformed even by the simple local methods, while the results obtained using the Euclidean inner product lagged behind both that of the Euclidean and the hyperbolic distances in greedy routing. Similar results obtained for four additional directed real networks are shown in Supplementary Note 5 of Ref. [T4], reinforcing the competitiveness of the application of hyperbolic geometry in embeddings.

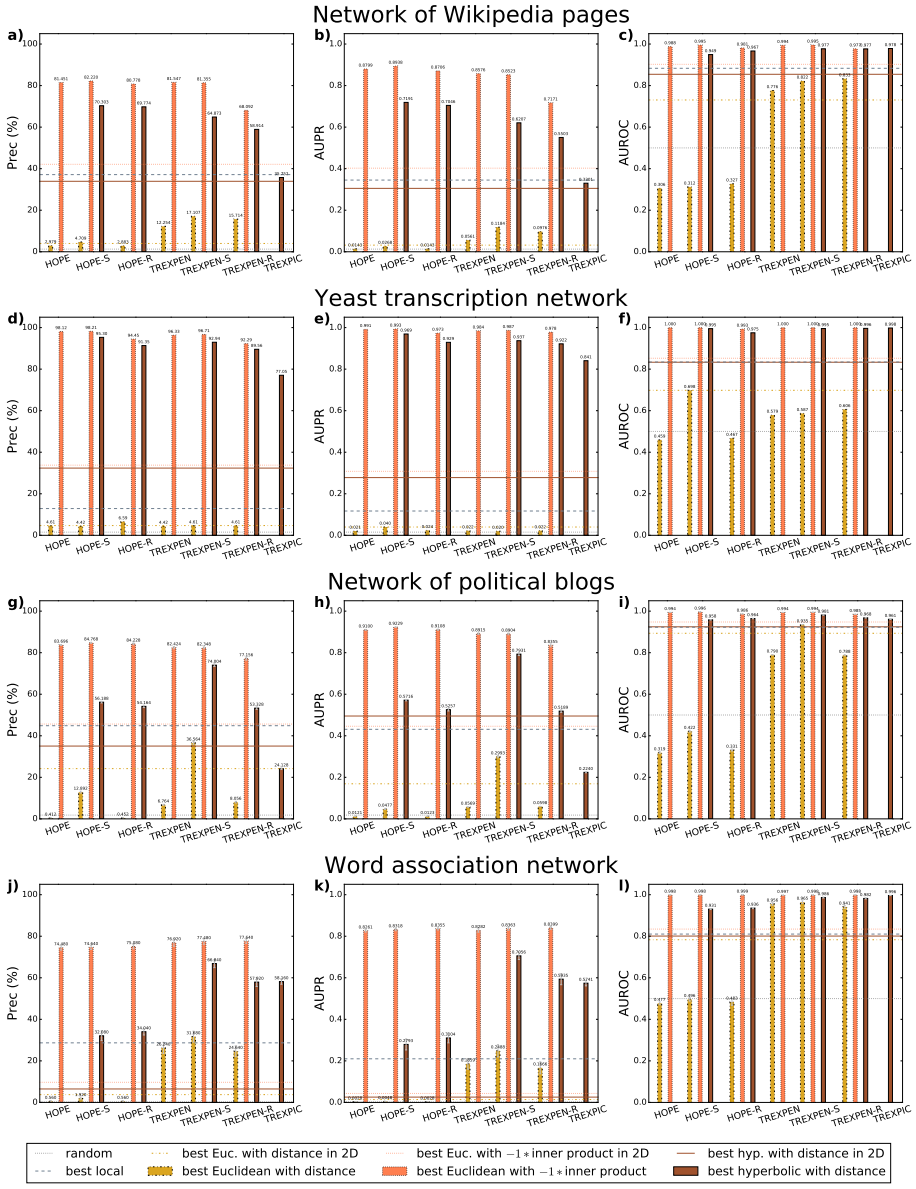

FIGURE 4.3.3.2: **Graph reconstruction performance on directed real networks.** For the networks in panels a)–f), the task was to reconstruct all the links ($E_{\text{toReconst}} = E$), whereas for the network of political blogs in panels g)–j) and for the word association network in panels j)–l), due to the large network size, the task was to reconstruct 5 samples of $E_{\text{toReconst}} = 5000$ and $E_{\text{toReconst}} = 500$ number of links, respectively. In the case of the larger networks, I always considered the average of the quality scores over the 5 samples and depicted the corresponding standard deviations with (usually very small) grey error bars. Each row of panels refers to a real network indicated in the row title, while the different columns show the different quality measures that have been studied, given by the precision obtained when reconstructing the first $E_{\text{toReconst}}$ most probable links (1$^{\text{st}}$ column), the area under the precision-recall (PR) curve (2$^{\text{nd}}$ column), and the area under the ROC curve (3$^{\text{rd}}$ column). The colors indicate the applied geometric measure, as listed in the common legend at the bottom of the figure. Using the bars, only the best results are plotted regarding all the performance measures, considering all the tested number of dimensions. The horizontal lines in color show the best two-dimensional performances achieved among all the embedding methods, whereas the grey horizontal lines correspond to the baselines provided by the random predictor and the best local method. The figure was taken from Ref. [T4].

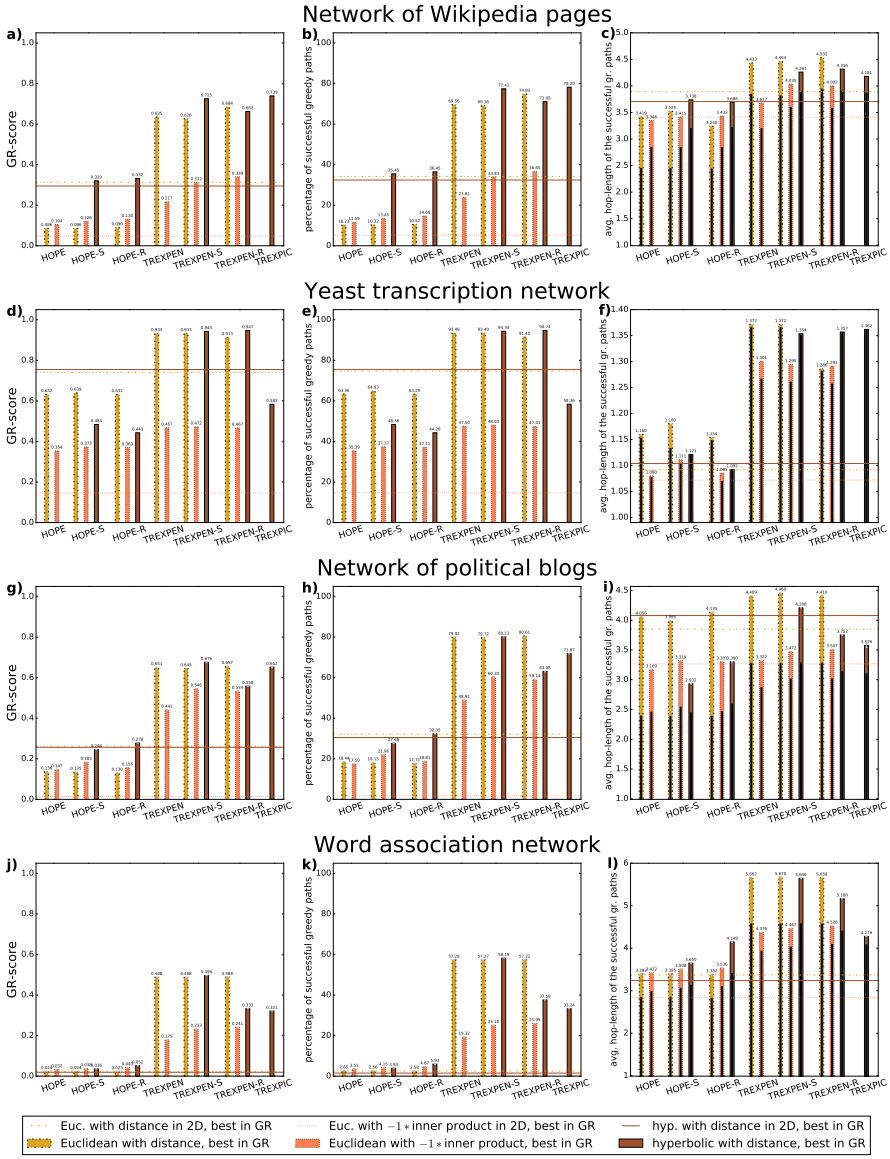

FIGURE 4.3.3.3: **Greedy routing performance on directed real networks.** For the networks in panels a)–f), the task was to perform greedy routing between each node pair connected by at least one directed path, whereas for the larger networks in panels g)–l), the task was to perform greedy routing in 5 samples of 500000 node pairs connected by at least one directed path. In the case of the larger networks, I always considered the average of the quality scores over the 5 samples and depicted the corresponding standard deviations with (usually very small) grey error bars. The colors indicate the used geometric measure as listed in the common legend at the bottom of the figure. The panels show for each method only the result of the parameter setting that turned out to be the best according to the GR-score. The bars were created considering all the tested number of dimensions, whereas the horizontal lines show the best two-dimensional average performances achieved among all the embedding methods. Each row of panels refers to a real network named in the row title, and the different columns correspond to different quality measures: the 1st column shows the greedy routing score (the higher the better), the 2nd column corresponds to the success rate of greedy routing (the higher the better), and the 3rd column depicts the average hop-length of the successful greedy paths (the smaller the better), where the grey bars indicate the average of the hop-length of the shortest paths connecting the node pairs for which the greedy routing was successful. The figure was taken from Ref. [T4].

4.3.4 Operation on undirected networks

As it is described in Sect. 4.1.1, the matrix $\boldsymbol{P}$ of topological proximities is decomposed in HOPE, TREXPEN and their variants as $\boldsymbol{P} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^{\mathrm{T}}$ to form a matrix $\boldsymbol{S} = \boldsymbol{U} \cdot \sqrt{\boldsymbol{\Sigma}}$ containing position vectors that describe the nodes' behavior as sources of links and a target coordinate matrix $\boldsymbol{T} = \boldsymbol{V} \cdot \sqrt{\boldsymbol{\Sigma}}$. For undirected networks, the proximity matrix $\boldsymbol{P}$ is symmetric, yielding $\boldsymbol{U} = \boldsymbol{V}$ in the singular value decomposition (SVD), and thus, $\boldsymbol{S} = \boldsymbol{T}$, meaning that, as expected, in undirected networks (or in directed networks having completely symmetric connections) each node is characterized by a single position vector.

When embedding an undirected network with TREXPIC, from a symmetric topological distance matrix $\boldsymbol{D}$ a symmetric matrix of expected Lorentz products is composed, yielding $\boldsymbol{U} = \boldsymbol{V}$ in the SVD $\boldsymbol{L} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^{\mathrm{T}}$. To obtain only one position vector for each node, i.e. to set $\boldsymbol{S} = \boldsymbol{T}$, I split equally between the source and the target coordinate matrices the -1 multiplying factors introduced in Eq. (4.2.5) in order to fulfill Eq. (4.2.3), meaning that I redefined $\boldsymbol{S}$ and $\boldsymbol{T}$ as

$$\begin{aligned} \boldsymbol{S} &= [+\sqrt{\sigma_1} \cdot \underline{u}_1 \,,\, \mathrm{i} \cdot \sqrt{\sigma_2} \cdot \underline{u}_2 \,,\, \mathrm{i} \cdot \sqrt{\sigma_3} \cdot \underline{u}_3 \,,\, \ldots \,,\, \mathrm{i} \cdot \sqrt{\sigma_{d+1}} \cdot \underline{u}_{d+1}] = \\ &= [+\sqrt{\sigma_1} \cdot \underline{v}_1 \,,\, \mathrm{i} \cdot \sqrt{\sigma_2} \cdot \underline{v}_2 \,,\, \mathrm{i} \cdot \sqrt{\sigma_3} \cdot \underline{v}_3 \,,\, \ldots \,,\, \mathrm{i} \cdot \sqrt{\sigma_{d+1}} \cdot \underline{v}_{d+1}] = \boldsymbol{T}, \end{aligned} \tag{4.3.4.1}$$

where $\mathrm{i} = \sqrt{-1}$ stands for the imaginary unit. Since all the coordinates from the second to the $d+1$th one are purely imaginary in the hyperboloid, the direction described by them is the same as if they all would be real numbers, and thus, the imaginary multiplying factors in Eq. (4.3.4.1) do not raise any issues.

As an example, Fig. 4.3.4.1 depicts some two-dimensional embeddings of the undirected American College Football network [79, 80] connecting $N = 115$ number of Division IA colleges via $E = 613$ number of games played during regular season Fall 2000, where each node has an attribute denoting to which of the 12 conferences it belongs. Based on these layouts, all the studied embedding algorithms automatically provide some sort of spatial separation between the groups of the nodes given by the conferences of the football teams. It is important to note that although the radial coordinates generated by TREXPIC are very close to each other, even such small differences have a substantial impact on the relations between the pairwise hyperbolic distances in the system and can yield a completely reasonable distance-based ordering of the node pairs since in the hyperbolic distance formula of Eq. (1.1.1), the radial coordinates are inputted into rapidly changing functions like sinh and cosh. Thus, as it is demonstrated by Table 4.3.4.1, TREXPIC's node arrangement depicted in Fig. 4.3.4.1k) is of a similar, or even better quality from the point of view of mapping accuracy, graph reconstruction and greedy routing than those hyperbolic layouts in Fig. 4.3.4.1 that were generated from Euclidean embeddings created by HOPE-S, HOPE-R, TREXPEN-S and TREXPEN-R with my Euclidean-hyperbolic conversion method MIC.

TABLE 4.3.4.1: **The performance in mapping accuracy, graph reconstruction and greedy routing of the two-dimensional hyperbolic embeddings that are presented for the football network in Fig. 4.3.4.1.** Despite its visually less pleasing radial arrangement, the embedding created by TREXPIC achieved the best scores in mapping accuracy and greedy routing. The best results are written in bold for each measure. The table was taken from Ref. [T4].

	HOPE-S	HOPE-R	TREXPEN-S	TREXPEN-R	TREXPIC
$\mathrm{ACC_m}$	0.350	0.347	0.357	0.352	**0.566**
Prec (%) in graph reconstruction	**50.08**	49.92	49.76	49.59	40.95
AUPR in graph reconstruction	**0.473**	0.441	0.453	0.439	0.376
AUROC in graph reconstruction	0.815	0.809	0.816	0.812	**0.868**
GR-score	0.566	0.555	0.561	0.557	**0.623**
percentage of successful greedy paths	67.00	65.63	66.17	66.01	**73.38**
average hop-length of the successful greedy paths	3.132	**3.114**	3.119	3.128	3.127

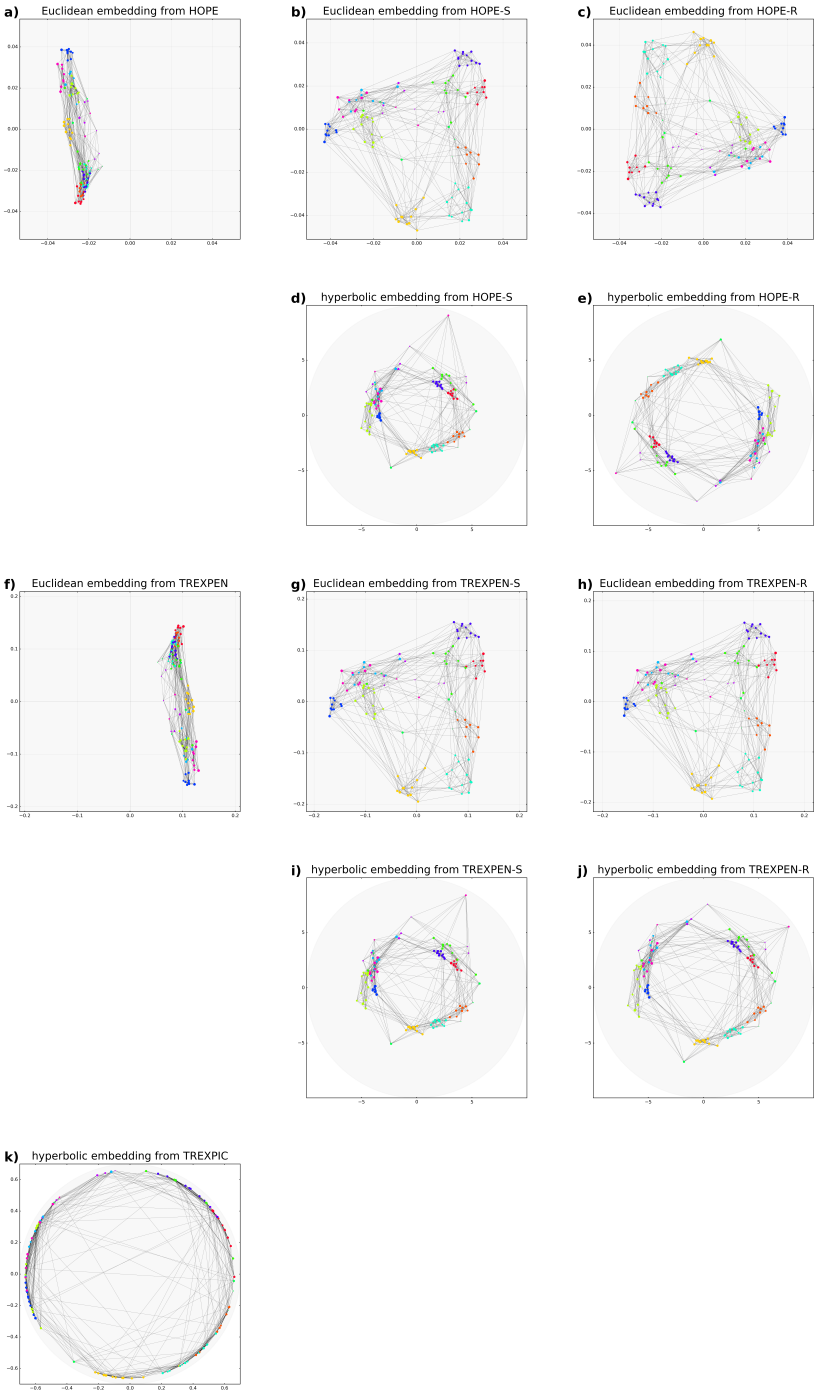

FIGURE 4.3.4.1: **Two-dimensional embeddings of the undirected American College Football network.** The color of each node indicates the conference to which it belongs. The node sizes are consistent with the node degrees. In the case of HOPE and its variants, α was set to $6.56 \cdot 10^{-3}$. The embeddings with TREXPEN and its variants were obtained at $q = 4.86$. The TREXPIC layout was created with $q = 6.07 \cdot 10^{-2}$. The parameter C of MIC was always set to 2. $\zeta = 1$ (i.e., the curvature $K = -1$) was used for all the hyperbolic embeddings. The figure was taken from Ref. [T4].

Following Supplementary Note 7 of Ref. [T4], Figs. 4.3.4.2–4.3.4.5 demonstrate that HOPE, TREXPEN, their variants and TREXPIC can provide good performance even when used on an undirected network (namely the American College Football network) and are able to cope with well-known undirected embedding techniques given by

- hyperbolic distance recovery and approximation (hydra, described in Sect. 3.3), which is similar to TREXPIC, but uses the plain shortest path lengths themselves instead of exponential distances and eigendecomposition instead of SVD,
- Laplacian Eigenmaps (LE, detailed in Sect. 3.1.1), which uses the eigendecomposition of the network's Laplacian matrix to place its nodes in the d-dimensional Euclidean space,
- Isomap (ISO, presented in Sect. 3.1.2), which, similarly to the Euclidean HOPE-S, HOPE-R, TREXPEN-S and TREXPEN-R methods, creates angularly not restricted patterns in the Euclidean space using SVD, but instead of searching for such a node arrangement that reproduces a matrix of pairwise topological proximities, builds on a matrix of expected pairwise Euclidean distances that is composed of the shortest path lengths measured along the network,
- and the hyperbolic versions of LE and ISO, in which, while keeping the angular coordinates unaltered, the Euclidean radial node arrangement obtained from the dimension reduction is replaced with the radial coordinates that have the highest probability according to the dPSO model of hyperbolic network growth [T3], as explained in Sect. 3.2.2.

To obtain well-comparable hyperbolic embeddings, I converted the results yielded by the hydra method in the Poincaré ball representation of the hyperbolic space of curvature $K = -1$ (Sect. 1.2) to the equivalent native representation (Sect. 1.1), where all the other examined methods place the nodes. For the radial conversion of the Euclidean embeddings provided by LE and ISO, I tested both dPSO-based approaches that are described in Sect. 3.2.2: making either the largest hyperbolic radial coordinate r_{NN} (Eqs. (3.2.2.3)–(3.2.2.4)) or the popularity fading parameter β (Eqs. (3.2.2.5)–(3.2.2.6)) dependent on the number of dimensions. Besides, since the radial order between the nodes having the same degree is arbitrary in the dPSO-based Euclidean-hyperbolic conversion (see Sect. 3.2.1), I re-ran this conversion 15 times in each case and always chose the best result for each quality measure. Other procedures, namely the Euclidean embeddings, hydra, TREXPIC, and also my model-independent Euclidean-hyperbolic conversion MIC were run only once with each parameter setting, as they are fully deterministic. Regarding TREXPIC and the variants of HOPE and TREXPEN, I tested the usual settings described in the introduction of Sect. 4.3 and plotted only the best results for each task. To give room for the performance-improving effect of the increase in the number of dimensions d, ensuring simultaneously a significant dimension reduction compared to the number of nodes N, I used $d = 2, 3, 4, ..., 2^n \leq \frac{N}{10}$ ($n \in \mathbb{Z}^+$) in both the earlier and the new embedding methods.

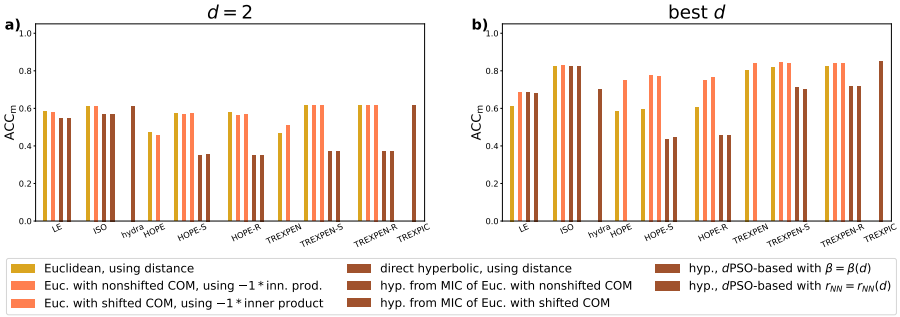

FIGURE 4.3.4.2: **Mapping accuracy on the undirected American College Football network.** The mapping accuracy was measured considering all the (unordered) node pairs of the network. The colors indicate what geometric measure was used, while the different patterns denote different characteristics regarding the way the node arrangements were created in the given geometry, as listed in the common legend at the bottom of the figure. The panels show for each method only the result of the parameter setting or trial that turned out to be the best, i.e. which yielded the highest mapping accuracy. For creating panel a), the network was embedded only in the Euclidean or hyperbolic plane, while the bars in panels b) were obtained considering all the tested number of dimensions d. The figure was taken from Ref. [T4].

First, Fig. 4.3.4.2 shows the mapping accuracies achieved by the different methods for the studied undirected real network. Then, Fig. 4.3.4.3 deals with the graph reconstruction task examining how well the embeddings can learn to differentiate between the inputted connected and unconnected node pairs (in-sample prediction), and Fig. 4.3.4.4 shows the results of some link prediction tasks investigating how well the embeddings can learn to differentiate between the missing links and the actually unconnected node pairs without having any specific input of the correct labels (out-of-sample prediction). Lastly, Fig. 4.3.4.5 presents the performances in greedy routing. According to these figures, TREXPIC and the variants of TREXPEN performed in all the tasks similarly to or even better than the previous undirected methods, while HOPE and its variants seem to be less suitable for the greedy routing task than the other algorithms.

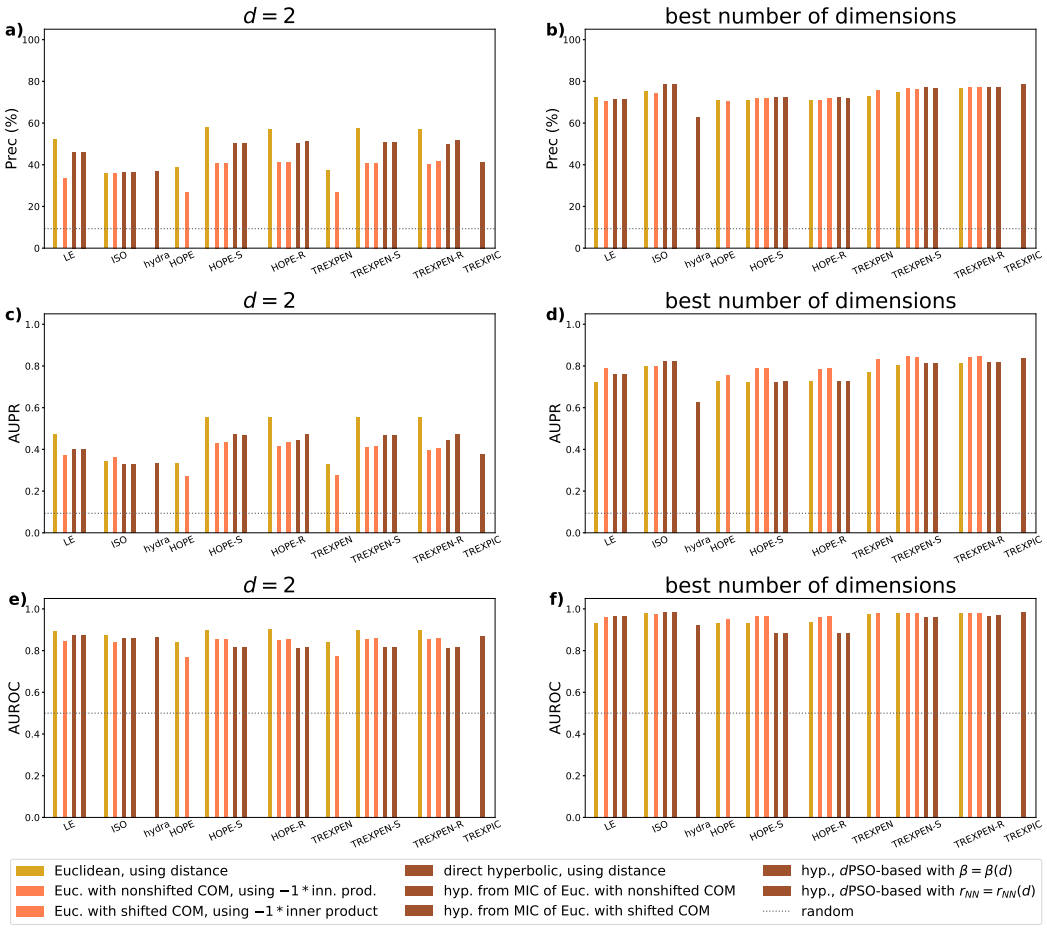

FIGURE 4.3.4.3: **Graph reconstruction performance on the undirected American College Football network.** The task was to reconstruct each one of the E number of links in the network by ranking all the (unordered) node pairs of the network using the node positions obtained when embedding the network knowing all of its links. The colors indicate what geometric measure was used, while the different patterns denote different characteristics regarding the way the node arrangements were created in the given geometry, as listed in the common legend at the bottom of the figure. Each row of panels refers to a given measure of embedding quality: panels a) and b) to the precision obtained when reconstructing the first E most probable links, panels c) and d) to the area under the precision-recall (PR) curve, whilst panels e) and f) to the area under the receiver operating characteristic (ROC) curve. The panels show for each method only the result of the parameter setting or trial that turned out to be the best regarding the given performance measure. For creating panels a), c) and e), the network was embedded only in the Euclidean or hyperbolic plane, while the bars in panels b), d) and f) were obtained considering all the tested number of dimensions d. The grey horizontal lines show the performance of the random predictor. The figure was taken from Ref. [T4].

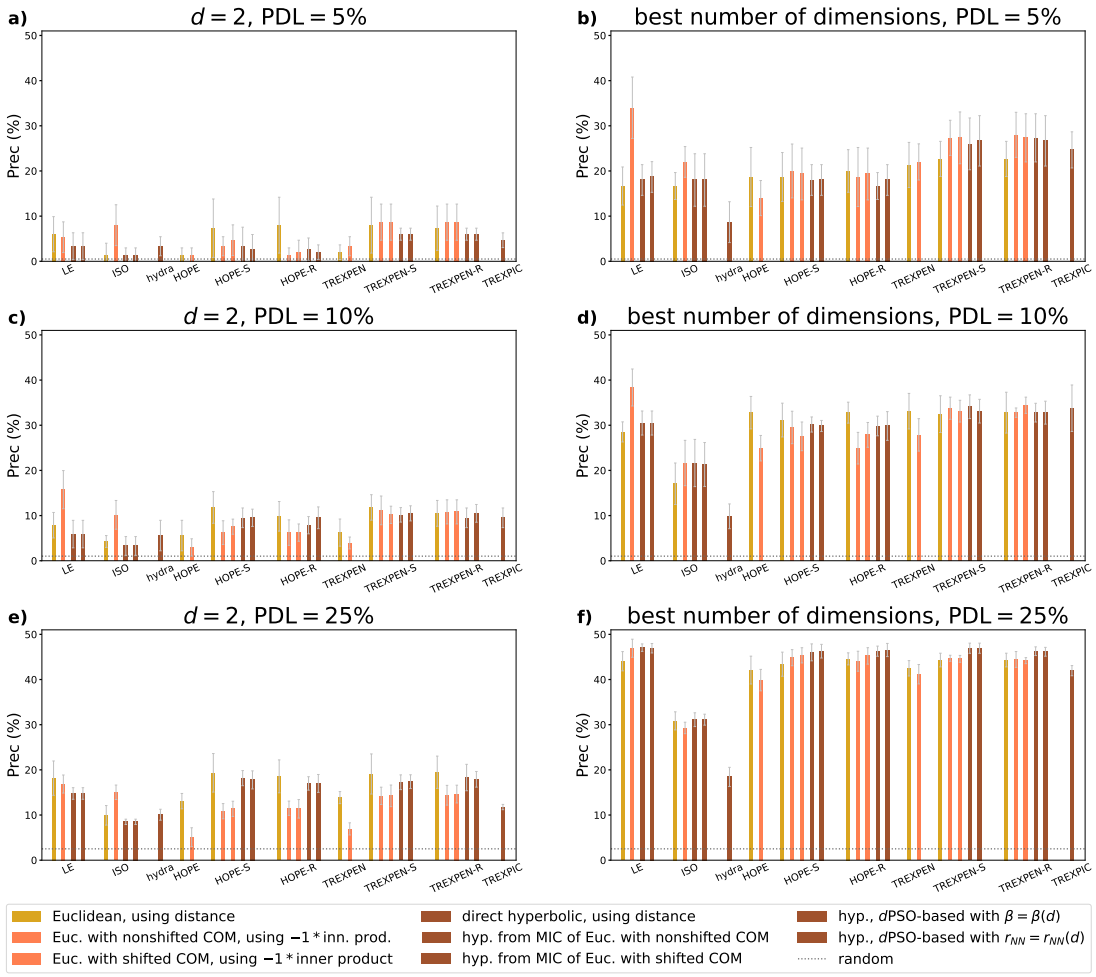

FIGURE 4.3.4.4: **Link prediction performance on the undirected American College Football network.** In this task, I randomly removed a given number of links from the original network, embedded the largest connected component of the obtained pruned graph and, by ranking all the unconnected (and unordered) node pairs that were embedded according to a given geometric measure associated with connection probability, tried to reconstruct all the E_{missing} number of the deleted links that connected in the original graph such nodes that both were embedded eventually. Each row of panels refers to a given proportion of deleted links (PDL) among all the connections of the original network. The link removal and prediction were repeated 5 times with each PDL. I considered in each case only the result of the parameter setting or trial that turned out to be the best regarding the precision measured when reconstructing the first E_{missing} most probable links (i.e., the proportion of the actually deleted links among the first E_{missing} unconnected node pairs in the order assigned by the given connection probability measure). The plotted values were obtained by averaging the results of the 5 link prediction tasks and the error bars show the standard deviations among the quality scores achieved on the 5 different sets of missing links. The colors indicate what geometric measure was used, while the different patterns denote different characteristics regarding the way the node arrangements were created in the given geometry, as listed in the common legend at the bottom of the figure. For creating panels a), c) and e), the network was embedded only in the Euclidean or hyperbolic plane, while the bars in panels b), d) and f) were obtained considering all the tested number of dimensions d. The grey horizontal lines show the average performance of the random predictor.

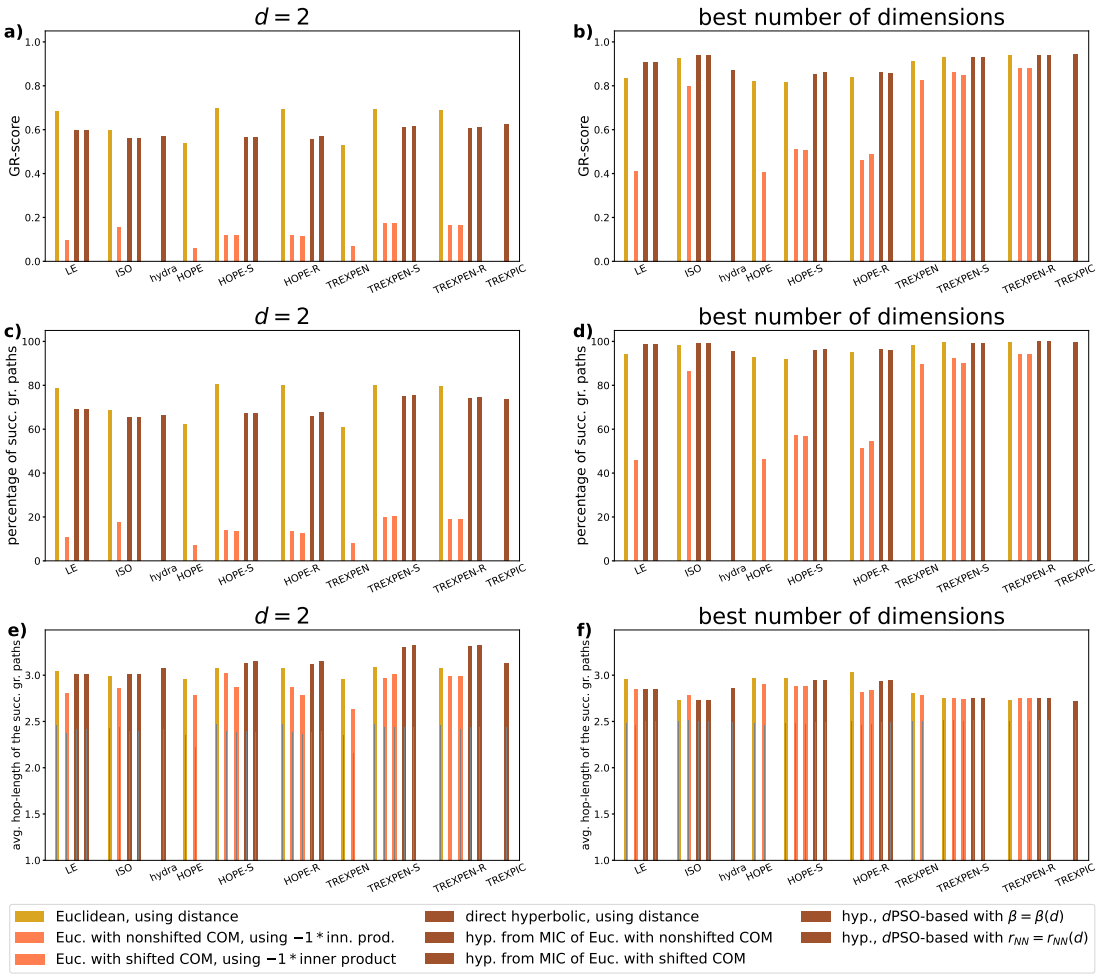

FIGURE 4.3.4.5: **Greedy routing performance on the undirected American College Football network.** The task was to perform greedy routing between each one of the (unordered) node pairs of the network. The colors indicate what geometric measure was used, while the different patterns denote different characteristics regarding the way the node arrangements were created in the given geometry, as listed in the common legend at the bottom of the figure. Each row of panels refers to a given measure of embedding quality: panels a) and b) to the greedy routing score (the higher the better), panels c) and d) to the success rate of greedy routing (the higher the better), whilst panels e) and f) to the average hop-length of the successful greedy paths (the smaller the better), depicting with grey bars also the average of the hop-length of the shortest paths connecting those node pairs for which the greedy routing was successful. The panels show for each method only the result of the parameter setting or trial that turned out to be the best regarding the GR-score. For creating panels a), c) and e), the network was embedded only in the Euclidean or hyperbolic plane, while the bars in panels b), d) and f) were obtained considering all the tested number of dimensions d. The figure was taken from Ref. [T4].

4.3.5 Considering real link weights

Many real network data include additional information regarding the node-node connections by means of link weights. Lacking such information, it is also possible to assign weights to the links (based on e.g. the number of links and common neighbors of the connected nodes [6]) to emphasize the topological properties that are already present in the unweighted network. In both cases, taking into account the weight of the links may be beneficial when embedding a network.

However, the application of HOPE and its variants on weighted networks is rather cumbersome. In unweighted networks, to enable a faster computation, the matrix of Katz proximities is usually calculated not with the defining formula

$$P_{st} = \sum_{\ell=1}^{\infty} \alpha^{\ell} \cdot n_{s\to t}^{\text{paths}}(\ell) \tag{4.3.5.1}$$

– already written in Eq. (4.1.1.1) – but as [81]

$$\boldsymbol{P} = \sum_{\ell=1}^{\infty} \alpha^{\ell} \cdot \boldsymbol{A}^{\ell} = (\mathbb{1} - \alpha \cdot \boldsymbol{A})^{-1} - \mathbb{1}, \tag{4.3.5.2}$$

where $\boldsymbol{A}$ is the adjacency matrix[8] of size $N \times N$, $\mathbb{1}$ is the identity matrix of size $N \times N$, and the second step uses the series expansion $(\mathbb{1} - \alpha \cdot \boldsymbol{A})^{-1} = \mathbb{1} + \alpha \cdot \boldsymbol{A} + \alpha^2 \cdot \boldsymbol{A}^2 + \mathcal{O}(\boldsymbol{A}^3)$, assuming that the decay parameter α is lower than the reciprocal of the spectral radius of the adjacency matrix $\boldsymbol{A}$, i.e. $\alpha < 1/\rho_{\text{spectral}}(\boldsymbol{A})$. In weighted graphs, where a path length ℓ is defined instead of simply the hop-length (yielding $\ell \in \mathbb{N}$) as the sum of the weights of the involved links (corresponding to $\ell \in \mathbb{R}_{\geq 0}$), Eq. (4.3.5.1) takes the form of

$$P_{st} = \sum_{0<\ell} \alpha^{\ell} \cdot n_{s\to t}^{\text{paths}}(\ell). \tag{4.3.5.3}$$

The calculation of the sum in Eq. (4.3.5.3) is highly compute-intensive due to the necessity of the exploration of all the possible paths between all the node pairs. Nonetheless, in weighted networks the matrix reformulation of the Katz index written in Eq. (4.3.5.2) is not valid anymore since $n_{s\to t}^{\text{paths}}(\ell) = \boldsymbol{A}^{\ell}$ is fulfilled only by unweighted graphs, i.e. when all the link weights are 1. Therefore, although HOPE and its variants, in theory, could embed weighted graphs too, due to computational restrictions, their application is feasible only for unweighted networks.

On the contrary, as the results below demonstrate, TREXPIC and the variants of TREXPEN are capable of utilizing the link weights of an inputted network without any complication. To use these embedding methods on networks having link weights that indicate topological distances (and not proximities), one simply has to calculate the exponential proximity of Eq. (4.1.1.2) or the exponential distance given by Eq. (4.2.2) using the weighted version of the shortest path lengths.

Figure 4.3.5.1 presents an example experiment that I carried out in Supplementary Note 8 of Ref. [T4] with the neural network of C. elegans [82–84]. To measure how precisely the inputted link weights (interpreted as distances) are reflected by given embeddings, I computed the Spearman's correlation coefficient [62] ACC_{w} of the weights with geometric measures that are considered to be an increasing function of the node-node topological distances, namely the Euclidean distance and the additive inverse of the inner product in Euclidean embeddings, and the hyperbolic distance in hyperbolic node arrangements. The successful mappings are characterized by high positive values of ACC_{w}.

In the case of Fig. 4.3.5.1a), the link weights were inputted in their original form to my embedding methods proposed in Ref. [T4]. Besides, Fig. 4.3.5.1b) depicts the case when the embedding algorithms were run using the reciprocal of the original weight values. Based on the results, the embedding methods were mostly able to optimize the node arrangement according to the link weights being inputted since the ACC_{w} values obtained with weighted embeddings exceed the corresponding baseline provided by the highest correlation that was achieved for the given link weights and geometric measure when ignoring the link weights during the embedding. Thus, it can be concluded that it might be beneficial to try out TREXPIC and the variants

[8] For unweighted networks, $A_{ij} = 0$ if there is no link pointing from node i to j and otherwise $A_{ij} = 1$; $A_{ii} = 0 \ \forall i$.

of TREXPEN also with the consideration of the link weights when such information is available or, just like it was suggested for undirected networks in Ref. [6], one may even introduce artificial link weights based on the network topology itself to facilitate the embedding process.

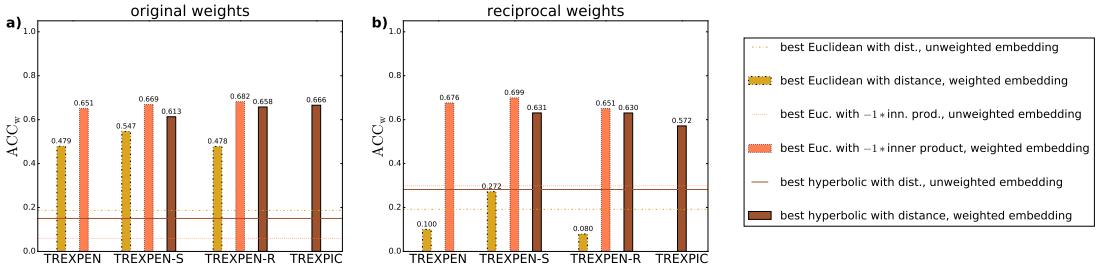

FIGURE 4.3.5.1: **Correlation between link weights and geometric measures in Euclidean and hyperbolic embeddings of a directed, weighted network.** The subject of the experiment was a neural network of C. elegans, which was embedded with my methods proposed in Ref. [T4]. In panel a), the original link weights given in the data set were treated as distances, while in panel b) the network was re-weighted with the reciprocal of the original link weights, interpreting the original link weights as proximities. The colors indicate with which geometric measure the Spearman's correlation coefficient of the non-zero link weights was calculated, as listed in the legend. The figure shows only the results of those parameter settings that turned out to be the best among my usual test set, i.e. which yielded the highest values of the correlation. The horizontal lines show the highest correlations achieved among all the algorithms when the unweighted version of the network was embedded, whereas the bars were created by also inputting the link weights to the embedding methods. The figure was taken from Ref. [T4].

Conclusion and outlook

Most networks describe such systems, the organization of which can be represented by a tree-like structure at least on the large-scale [1], and trees can be interpreted as a discrete version of a negatively curved, i.e. hyperbolic space [2]. Hence the idea of an underlying hyperbolic geometry of complex networks straightforwardly arises [2], as described in Chapter 1.

By tracing back the connection rules of the network nodes to an origin in hyperbolic geometry, many common features of real-world complex networks appear to be explainable. Namely, as Sect. 2.1 also highlighted through the example of the popularity-similarity optimization (PSO) model [3], network models that generate mostly well-localized connections between nodes distributed in a simple manner on the hyperbolic plane are known to be capable of producing such graphs that possess realistic features given by the small-world property, a scale-free degree distribution and a high average clustering coefficient.

Still, as it was pointed out in Ref. [7], the PSO model of hyperbolic network growth by construction does not reproduce a feature that is claimed by Refs. [20, 21] to be frequent in real networks, namely the densification of the subgraphs spanning between nodes of large enough degree towards the larger degree thresholds. Since degree thresholding is basically the same as selecting all nodes that are older than a given age, a PSO network's subgraph given by the nodes of a large enough degree corresponds to a snapshot of the network from its growth process. Besides, as in the PSO model each node establishes the same m number of links at its appearance, the expected average degree is approximately $2m$ for a snapshot of any time point. Thus, the average internal degree of the subgraphs spanning between the nodes of degree larger than a given threshold does not change with the degree threshold in the case of PSO networks [7] until the degree threshold does not become so large that the corresponding subgraphs become extremely small. However, as it was shown in Sect. 2.2, in the E-PSO model [4] that also simulates the emergence of $L > 0$ number of internal links in each time step besides the formation of m number of external connections, the number of links established by the new nodes at their appearance is a decreasing function of the appearance time, and thus, the average internal degree of the examined subgraphs shows an increasing tendency as the degree threshold begins to increase [T1]. In addition, with the simulation of link deletion during the network growth [T1], the average internal degree can be made a decreasing function of the degree threshold even at relatively small values of the threshold, meaning that in a rather simple extension of the PSO model, the characteristics of the curve in question become well adjustable.

Furthermore, although there have already been indications that – despite lacking any explicit community formation mechanism – angularly uniform hyperbolic network models are no stranger to the emergence of modular structures [22–25], this property was not broadly recognized for a long time. To dispel any doubts, Sect. 2.3 demonstrated the ability of the PSO model to generate networks of strongly modular structure through a detailed numerical examination [T2], confirming that generating networks on the hyperbolic plane provides the opportunity for simultaneously achieving not only the small-world property, a heterogeneous degree distribution and a strong clustering, but an actually relevant community structure too, which is also a well-known common feature of real networks.

A currently active further area of research is the generation of directed links in hyperbolic spaces [49, 85], which was also contributed by Ref. [T4], where I extended the two-dimensional E-PSO model of Sect. 2.2 to directed networks in order to provide a synthetic benchmark for testing my directed embedding methods described in Chapter 4 – however, the description of this model variant and the corresponding measurements are left out from this dissertation due to length constraints. In addition, the emerging field of hypergraphs [86, 87] offers another

direction toward which hyperbolic network models could be extended, by connecting hyperbolically close nodes not only in pairs but in groups to simulate higher-order interactions too.

Besides laying the foundation for generating networks of realistic features using a hyperbolic plane, a connection between network topology and hyperbolic geometry also suggests that a hyperbolic embedding of a graph (i.e., an arrangement of its nodes in a hyperbolic space) serves as some sort of natural representation of the network structure, where the spatial distances between the nodes reflect their topological relationships. With the development of hyperbolic embedding methods that can place the network nodes not only on the hyperbolic plane but in higher-dimensional spaces [6, 8, 10] – characterizing each node in more detail with a larger number of coordinates –, the demand for generating higher-dimensional hyperbolic test networks has naturally arisen. This was followed by the d-dimensional generalization of the PSO model of network growth [3] and a static network model working on a hyperbolic plane [2] in Ref. [T3] and Refs. [13, 42], respectively. Using the dPSO model, Sect. 2.4 revealed that changing the number of dimensions of the underlying hyperbolic space affects the properties of the generated networks: higher-dimensional hyperbolic spaces seem to limit the achievable maximal strength of the clustering and the emergent community structure of angularly uniform dPSO networks having a given degree decay exponent. Concerning network embedding, Sect. 4.3.4 demonstrated the applicability of dPSO-based radial coordinates when transforming d-dimensional Euclidean node arrangements of real networks into hyperbolic ones [T4], also utilizing the freedom in the radial ordering of equal-degree nodes recognized in Ref. [T1] and detailed in Sect. 3.2.1.

Nevertheless, as an alternative, one can eliminate the optimization for any specific hyperbolic network model at all [T4], both in the case of creating a hyperbolic embedding through the conversion of a Euclidean one that aims at retaining expected inner products between the node position vectors given by a matrix of node-node topological proximities (Sect. 4.1) and in the case of embedding a network directly in the hyperbolic space, trying to reproduce a matrix of pairwise topological distances that are interpreted as estimations of the proper hyperbolic distances in the embedding (Sect. 4.2). To enable capturing the topological relations of a network more precisely, in these new methods transferred from Ref. [T4] the matrix of expected geometric measures is always derived as a relatively fast-changing, exponential function of the shortest path lengths. Besides being model independent, the proposed algorithms of TREXPEN (TRansformation of EXponential shortest Path lengths to EuclideaN measures), its variants and TREXPIC (TRansformation of EXponential shortest Path lengths to hyperbolIC measures) provide a solution for embedding not only undirected but also directed networks into hyperbolic spaces of any number of dimensions. According to Sect. 4.3, the hyperbolic embedding algorithms presented in Sects. 4.1–4.2 are able to simultaneously perform relatively well in mapping accuracy, graph reconstruction and greedy routing. This is even more valuable in the light of that the node arrangements obtained in Euclidean spaces along similar principles (Sect. 4.1.1) did not excel in all three tasks when evaluated based on a single geometric measure: using Euclidean distances resulted in poorer outcomes regarding mapping accuracy and graph reconstruction, while inner products yielded inferior navigability in the examined Euclidean embeddings.

A further fundamental task that can be addressed via node embeddings is the detection of communities of network nodes. Since hyperbolic embeddings are expected to gather together the nodes of similar connection preferences (Sect. 4.3.1), e.g. groups of nodes that are densely connected to each other and sparsely to other nodes can be anticipated to form well-localized clusters in a hyperbolic space. The exploration of such dense clusters can be targeted using standard data clustering methods such as k-means clustering [88], DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [89] or BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [90]. A currently still open question is [91], however, whether hyperbolic embeddings or even embeddings in general can compete with commonly used community finding methods like Louvain [26], asynchronous label propagation [28] or Infomap [29].

Summary of the new scientific results in English

Hidden geometric structures underlying complex networks gain increasing attention in recent studies of network theory. The assumption of an analogy between the hyperbolic geometry and the connection rules of real-world networks yielded two natural applications: the generation of real-like artificial graphs using distance-dependent connection probabilities between nodes arranged in a hyperbolic space, and the hyperbolic embedding of real networks, where the aim is to find such a hyperbolic arrangement of the network nodes that well reflect the connection structure of the given network through the hyperbolic distances between the node position vectors. My PhD dissertation, discussing my publications given by Refs. [T1]–[T4], contributes to the fields of both hyperbolic network generation models and hyperbolic node embeddings.

It is well-known that hyperbolic network models, including the popularity-similarity optimization (PSO) model of network growth, are able to reproduce realistic network features given by the small-world property, a scale-free degree distribution and a high average clustering coefficient. In addition, as I have shown in Ref. [T1], by allowing in the PSO model the formation of new links between previously appeared nodes too and not only between the newly appearing node and the older ones, and by also simulating the disappearance of connections, the dependence of the average internal degree of the subgraphs spanning between the nodes of large enough degree on the degree threshold can be made well adjustable. Thus, a densification of such subgraphs towards the larger degree thresholds becomes achievable, which is also claimed to be a frequent property of real networks. Moreover, following some rudimentary observations from the literature, I confirmed in Ref. [T2] via a detailed numerical study that strong community structures – which are often present in real networks too – can emerge from the PSO model in a large region of the parameter space, even though this was not an intention at the construction of the model.

After the application of three- or higher-dimensional hyperbolic spaces has become more and more popular in the hyperbolic embedding methods, I explored in Ref. [T3] the implications of increasing the number of dimensions of the hyperbolic space above 2 in the PSO model. The examination of the obtained dPSO model revealed that the increase in the number of dimensions d limits the achievable maximal strength of the clustering and the emergent community structure of angularly uniform dPSO networks having a given degree decay exponent. Besides, I used the dPSO model in Ref. [T4] for transforming d-dimensional Euclidean node embeddings into hyperbolic ones, also utilizing the freedom in the radial ordering of equal-degree nodes described in Ref. [T1].

Finally, in Ref. [T4] I presented the hyperbolic embedding algorithms that I developed for embedding not only undirected graphs but also directed networks in hyperbolic spaces of any number of dimensions, avoiding the assumption of any specific hyperbolic network model as the generator of the network to be embedded and using novel, exponential measures of the topological proximity and distance. According to numerous measurements carried out on real networks, both the methods that create hyperbolic embeddings from Euclidean node arrangements with my new, model-independent conversion and the algorithm that embeds networks directly in the hyperbolic space perform relatively well in mapping accuracy, graph reconstruction and also greedy routing at the same time.

Az új tudományos eredmények magyar nyelvű összefoglalása

A közelmúlt hálózatelméleti irodalmában egyre nagyobb figyelem irányul a komplex hálózatok mögött húzódó rejtett geometriai struktúrákra. A hiperbolikus geometria és a valós hálózatok kötési szabályai közti analógia feltételezése nyomán két természetes alkalmazási terület bontakozott ki: egyrészt realisztikus tulajdonságokkal rendelkező mesterséges gráfok generálása hiperbolikus térben elrendezett pontok távolságfüggő valószínűségek szerint történő összekötésével, másrészt valós hálózatok hiperbolikus beágyazása, amelynek célja a hálózati pontok elrendezése a hiperbolikus térben oly módon, hogy a pontok pozícióvektorai között mért hiperbolikus távolságok a beágyazni kívánt valós hálózat kötési viszonyait jól tükrözzék. A PhD disszertációm, amely a [T1]–[T4] publikációimat dolgozza fel, mind a hiperbolikus hálózatgeneráló modellek, mind pedig a hiperbolikus pontbeágyazások területének fejlődéséhez hozzájárul.

Jól ismert tény, hogy a hiperbolikus hálózatmodellek, köztük a hálózatnövekedés népszerűség-hasonlóság optimalizációs (popularity-similarity optimization, PSO) modellje képesek realisztikus hálózati tulajdonságok, úgymint a kisvilág-tulajdonság, a skálafüggetlen fokszámeloszlás és a magas átlagos klaszterezettségi együttható reprodukálására. Ezenfelül, ahogy megmutattam a [T1] publikációmban, ha megengedjük a PSO modellben az új élek keletkezését korábban megjelent pontok közt is és nem csak az újonnan megjelenő pont és a régebb óta jelen lévők között, illetve szimuláljuk a kapcsolatok eltűnését is, akkor az adott értéknél nagyobb fokszámú pontok által kifeszített részgráfon belüli átlagfokszámnak a részgráfot kijelölő legkisebb fokszám értékétől való függése jól szabályozhatóvá válik. Ily módon elérhetővé válik a szóban forgó részgráfoknak a nagyobb fokszámküszöbök felé történő sűrűsödése, amit a valós hálózatok egy további gyakori jellemzőjeként tartanak számon. Továbbá a szakirodalomban fellelhető néhány kezdetleges megfigyelést követve a [T2] publikációmban egy részletes numerikus vizsgálattal igazoltam, hogy a valós hálózatoknak egy szintén gyakori jellemzőjét képező erős csoportstruktúrák létre tudnak jönni a PSO modellben a paramétertér igen nagy részében még annak ellenére is, hogy a modell megtervezésekor ez nem volt cél.

Miután a hiperbolikus beágyazó módszerekben egyre népszerűbbé vált a három- vagy magasabb dimenziós hiperbolikus terek használata, a [T3] publikációmban felderítettem a hiperbolikus tér dimenziószámának 2 fölé történő növelésének következményeit a PSO modellben. Az így kapott dPSO modell vizsgálata feltárta, hogy a fokszámeloszlás lecsengését leíró adott hatványkitevő mellett a d dimenziószám növelése korlátozza az egyenletes szögeloszlású dPSO hálózatok klaszterezettségének és emergens csoportstruktúrájának az elérhető legnagyobb erősségét. Emellett a [T4] publikációmban felhasználtam a dPSO modellt d-dimenziós Euklideszi pontbeágyazások hiperbolikussá történő transzformálására, amely során az azonos fokszámú pontok radiális sorbarendezésének a [T1] publikációmban leírt szabadságát is kihasználtam.

Végezetül a [T4] publikációmban bemutattam a hiperbolikus beágyazó algoritmusokat, amelyeket arra fejlesztettem ki, hogy ne csak irányítatlan gráfokat, hanem irányított hálózatokat is be lehessen velük ágyazni tetszőleges dimenziószámú térbe, elkerülve a feltételezését bármilyen konkrét hiperbolikus hálózatmodellnek a beágyazandó hálózat forrásaként, valamint a topológiai közelségnek és távolságnak új, exponenciális mérőszámait használva. Valós hálózatokon elvégzett számos mérés alapján a leképezési pontosság, a gráfrekonstrukció és a mohó navigáció szempontjából egyaránt viszonylag jól teljesítenek mind azok a módszerek, amelyek Euklideszi pontelrendezésekből készítenek hiperbolikus beágyazásokat a saját, modellfüggetlen konverziómmal, mind pedig az az algoritmus, ami közvetlenül a hiperbolikus térbe ágyaz be hálózatokat.

Publications during the doctoral training

Publications used in the dissertation

[T1] B. Kovács and G. Palla. Optimisation of the coalescent hyperbolic embedding of complex networks. *Scientific Reports* **11**, 8350, https://doi.org/10.1038/s41598-021-87333-5 (2021)

[T2] B. Kovács and G. Palla. The inherent community structure of hyperbolic networks. *Scientific Reports* **11**, 16050, https://doi.org/10.1038/s41598-021-93921-2 (2021)

[T3] B. Kovács, S. G. Balogh and G. Palla. Generalised popularity-similarity optimisation model for growing hyperbolic networks beyond two dimensions. *Scientific Reports* **12**, 968, https://doi.org/10.1038/s41598-021-04379-1 (2022)

[T4] B. Kovács and G. Palla. Model-independent embedding of directed networks into Euclidean and hyperbolic spaces. *Communications Physics* **6**, 28, https://doi.org/10.1038/s42005-023-01143-x (2023)

Other publications

[S1] S. G. Balogh, B. Kovács and G. Palla. Maximally modular structure of growing hyperbolic networks. Preprint at arXiv:2206.08773 [physics.soc-ph], accepted for publication in Communications Physics on March 16, 2023. https://doi.org/10.48550/arXiv.2206.08773

References

1. Clauset, A., Moore, C. & Newman, M. E. J. Hierarchical structure and the prediction of missing links in networks. *Nature* **453,** 98–101. ISSN: 1476-4687. https://doi.org/10.1038/nature06830 (2008).
2. Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A. & Boguñá, M. Hyperbolic geometry of complex networks. *Phys. Rev. E* **82,** 036106. https://doi.org/10.1103/PhysRevE.82.036106 (2010).
3. Papadopoulos, F., Kitsak, M., Serrano, M. Á., Boguñá, M. & Krioukov, D. Popularity versus similarity in growing networks. *Nature* **489,** 537–540. https://doi.org/10.1038/nature11459 (2012).
4. Papadopoulos, F., Psomas, C. & Krioukov, D. Network Mapping by Replaying Hyperbolic Growth. *IEEE/ACM Transactions on Networking* **23,** 198–211. ISSN: 1063-6692. https://doi.org/10.1109/TNET.2013.2294052 (2015).
5. Alanis-Lobato, G., Mier, P. & Andrade-Navarro, M. Efficient embedding of complex networks to hyperbolic space via their Laplacian. *Sci. Rep.* **6,** 301082. https://doi.org/10.1038/srep30108 (2016).
6. Muscoloni, A., Thomas, J. M., Ciucci, S., Bianconi, G. & Cannistraci, C. V. Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nature Communications* **8,** 1615. ISSN: 2041-1723. https://doi.org/10.1038/s41467-017-01825-5 (2017).
7. García-Pérez, G., Allard, A., Serrano, M. Á. & Boguñá, M. Mercator: uncovering faithful hyperbolic embeddings of complex networks. *New J. Phys.* **21,** 123033. https://doi.org/10.1088/1367-2630/ab57d2 (2019).
8. Nickel, M. & Kiela, D. *Poincaré Embeddings for Learning Hierarchical Representations* in *Advances in Neural Information Processing Systems* (eds Guyon, I. *et al.*) **30** (Curran Associates, Inc., 2017). https://proceedings.neurips.cc/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf.
9. Chami, I., Ying, R., Ré, C. & Leskovec, J. Hyperbolic Graph Convolutional Neural Networks. en. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)* **32,** 4869–4880. https://doi.org/10.48550/arXiv.1910.12933 (Dec. 2019).
10. Keller-Ressel, M. & Nargang, S. Hydra: a method for strain-minimizing hyperbolic embedding of network- and distance-based data. *Journal of Complex Networks* **8.** cnaa002. ISSN: 2051-1329. https://doi.org/10.1093/comnet/cnaa002 (Feb. 2020).
11. Nickel, M. & Kiela, D. *Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry* in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (eds Dy, J. G. & Krause, A.) **80** (PMLR, 2018), 3776–3785. http://proceedings.mlr.press/v80/nickel18a.html.
12. McDonald, D. & He, S. Hyperbolic Embedding of Attributed and Directed Networks. *IEEE Transactions on Knowledge and Data Engineering*, 1–12. https://doi.org/10.1109/TKDE.2022.3188426 (2022).
13. Yang, W. & Rideout, D. High Dimensional Hyperbolic Geometry of Complex Networks. *Mathematics* **8.** ISSN: 2227-7390. https://doi.org/10.3390/math8111861 (2020).

14. Zuev, K., Boguñá, M., Bianconi, G. & Krioukov, D. Emergence of Soft Communities from Geometric Preferential Attachment. *Sci. Rep.* **5,** 9421. https://doi.org/10.1038/srep09421 (2015).

15. García-Pérez, G., Serrano, M. & Boguñá, M. Soft Communities in Similarity Space. *Journal of Statistical Physics*. https://doi.org/10.1007/s10955-018-2084-z (July 2017).

16. Muscoloni, A. & Cannistraci, C. V. A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities. *New J. Phys.* **20,** 052002. https://doi.org/10.1088/1367-2630/aac06f (2018).

17. Barabási, A.-L. & Albert, R. Emergence of Scaling in Random Networks. *Science* **286,** 509–512. eprint: https://www.science.org/doi/pdf/10.1126/science.286.5439.509. https://doi.org/10.1126/science.286.5439.509 (1999).

18. Bollobás, B. & Riordan, O. *Mathematical results on scale-free random graphs* in *Handbook of Graphs and Networks* (Wiley-VCH, 2003).

19. Clauset, A., Shalizi, C. R. & Newman, M. E. J. Power-Law Distributions in Empirical Data. *SIAM Review* **51,** 661–703. https://doi.org/10.1137/070710111 (Nov. 2009).

20. Serrano, M. A., Krioukov, D. & Boguñá, M. Self-Similarity of Complex Networks and Hidden Metric Spaces. *Phys. Rev. Lett.* **100,** 078701. https://doi.org/10.1103/PhysRevLett.100.078701 (2008).

21. Serrano, M. A., Krioukov, D. & Boguñá, M. Percolation in Self-Similar Networks. *Phys. Rev. Lett.* **106,** 048701. https://doi.org/10.1103/PhysRevLett.106.048701 (2011).

22. Wang, Z., Li, Q., Jin, F., Xiong, W. & Wu, Y. Hyperbolic mapping of complex networks based on community information. *Physica A: Statistical Mechanics and its Applications* **455,** 104 –119. ISSN: 0378-4371. https://doi.org/10.1016/j.physa.2016.02.015 (2016).

23. Wang, Z., Wu, Y., Li, Q., Jin, F. & Xiong, W. Link prediction based on hyperbolic mapping with community structure for complex networks. *Physica A: Statistical Mechanics and its Applications* **450,** 609 –623. ISSN: 0378-4371. https://doi.org/10.1016/j.physa.2016.01.010 (2016).

24. Wang, Z., Sun, L., Cai, M. & Xie, P. Fast hyperbolic mapping based on the hierarchical community structure in complex networks. *Journal of Statistical Mechanics: Theory and Experiment* **2019,** 123401. https://doi.org/10.1088/1742-5468/ab3bc8 (2019).

25. Faqeeh, A., Osat, S. & Radicchi, F. Characterizing the Analogy Between Hyperbolic Embedding and Community Structure of Complex Networks. *Phys. Rev. Lett.* **121,** 098301. https://doi.org/10.1103/PhysRevLett.121.098301 (2018).

26. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008,** P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008 (2008).

27. Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69,** 026113. https://doi.org/10.1103/PhysRevE.69.026113 (2004).

28. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76,** 036106. https://doi.org/10.1103/PhysRevE.76.036106 (2007).

29. Rosvall, M. & Bergstrom, C. T. Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems. *PLOS ONE* **6,** 1–10. https://doi.org/10.1371/journal.pone.0018209 (Apr. 2011).

30. Erdős, P. & Rényi, A. *On the Evolution of Random Graphs* in *Publication of the Mathematical Institute of the Hungarian Academy of Sciences* **5** (1960), 17–61.

31. Guimerà, R., Sales-Pardo, M. & Amaral, L. A. N. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E* **70,** 025101(R). https://doi.org/10.1103/PhysRevE.70.025101 (2004).

32. Good, B. H., Montjoye, Y.-A. & Clauset, A. Performance of modularity maximization in practical contexts. *Phys. Rev. E* **81,** 046106. https://doi.org/10.1103/PhysRevE.81.046106 (2010).

33. Vinh, N. X., Epps, J. & Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research* **11,** 2837–2854. http://jmlr.org/papers/v11/vinh10a.html (2010).

34. McCarthy, A. D. & Matula, D. W. *Normalized mutual information exaggerates community detection performance* in *SIAM Workshop on Network Science 2018* (2018), 78–79. http://cs.jhu.edu/~arya/mccarthy+matula.ns18.pdf.

35. Chellig, J., Fountoulakis, N. & Skerman, F. The modularity of random graphs on the hyperbolic plane. *Journal of Complex Networks* **10.** cnab051. ISSN: 2051-1329. eprint: https://academic.oup.com/comnet/article-pdf/10/1/cnab051/41987187/cnab051.pdf. https://doi.org/10.1093/comnet/cnab051 (Dec. 2021).

36. Tenenbaum, J. B., de Silva, V. & Langford, J. C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **290,** 2319–2323. eprint: https://www.science.org/doi/pdf/10.1126/science.290.5500.2319. https://doi.org/10.1126/science.290.5500.2319 (2000).

37. Belkin, M. & Niyogi, P. *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering* in *Advances in Neural Information Processing Systems* (eds Dietterich, T., Becker, S. & Ghahramani, Z.) **14** (MIT Press, 2001). https://proceedings.neurips.cc/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf.

38. Tang, J. *et al. LINE: Large-Scale Information Network Embedding* in *Proceedings of the 24th International Conference on World Wide Web* (International World Wide Web Conferences Steering Committee, Florence, Italy, 2015), 1067–1077. ISBN: 9781450334693. https://doi.org/10.1145/2736277.2741093.

39. Ou, M., Cui, P., Pei, J., Zhang, Z. & Zhu, W. *Asymmetric Transitivity Preserving Graph Embedding* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, San Francisco, California, USA, 2016), 1105–1114. ISBN: 9781450342322. https://doi.org/10.1145/2939672.2939751.

40. Grover, A. & Leskovec, J. Node2vec: Scalable feature learning for networks. en. *KDD* **2016,** 855–864. https://doi.org/10.1145/2939672.2939754 (Aug. 2016).

41. Muscoloni, A. & Cannistraci, C. V. *Angular separability of data clusters or network communities in geometrical space and its relevance to hyperbolic embedding* Preprint at arXiv:1907.00025 [cs.LG] (2019). https://doi.org/10.48550/arXiv.1907.00025.

42. Kitsak, M., Aldecoa, R., Zuev, K. & Krioukov, D. *Random hyperbolic graphs in $d+1$ dimensions* Preprint at arXiv:2010.12303 [physics.soc-ph]. 2020. https://doi.org/10.48550/arXiv.2010.12303.

43. Kleinberg, J. Navigation in a small world. *Nature* **406,** 845. https://doi.org/10.1038/35022643 (2000).

44. Boguñá, M., Krioukov, D. & Claffy, K. Navigability of complex networks. *Nature Phys.* **5,** 74–80. https://doi.org/10.1038/nphys1130 (2009).

45. Muscoloni, A. & Cannistraci, C. V. Navigability evaluation of complex networks by greedy routing efficiency. *Proceedings of the National Academy of Sciences* **116,** 1468–1469. ISSN: 0027-8424. eprint: https://www.pnas.org/content/116/5/1468.full.pdf. https://doi.org/10.1073/pnas.1817880116 (2019).

46. Cramer, H. in, 390 (Princeton University Press, 1999). ISBN: 0691005478,9780691005478.

47. *The Pierre Auger collaboration network was downloaded from the CoMuNe Lab website: https://comunelab.fbk.eu/data.php.*

48. *The network of PDZ-domain-mediated protein–protein binding interactions was downloaded from https://konect.cc/networks/maayan-pdzbase/.*

49. Shen, D., Wu, Z., Di, Z. & Fan, Y. An Asymmetric Popularity-Similarity Optimization Method for Embedding Directed Networks into Hyperbolic Space. *Complexity* **2020,** 8372928. https://doi.org/10.1155/2020/8372928 (2020).

50. *The network between political blogs that I used for testing my embedding methods was downloaded from http://konect.cc/networks/dimacs10-polblogs/.*

51. Adamic, L. A. & Glance, N. *The Political Blogosphere and the 2004 U.S. Election: Divided They Blog* in *Proceedings of the 3rd International Workshop on Link Discovery* (Association for Computing Machinery, Chicago, Illinois, 2005), 36–43. ISBN: 1595932151. https://doi.org/10.1145/1134271.1134277.

52. Cannistraci, C. V., Ravasi, T., Montevecchi, F. M., Ideker, T. & Alessio, M. Nonlinear dimension reduction and clustering by Minimum Curvilinearity unfold neuropathic pain and tissue embryological classes. *Bioinformatics* **26,** i531–i539. ISSN: 1367-4803. eprint: https://academic.oup.com/bioinformatics/article-pdf/26/18/i531/16894826/btq376.pdf. https://doi.org/10.1093/bioinformatics/btq376 (Sept. 2010).

53. Cannistraci, C. V., Alanis-Lobato, G. & Ravasi, T. Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics* **29,** i199–i209. https://doi.org/10.1093/bioinformatics/btt208 (June 2013).

54. Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **18,** 39–43. ISSN: 1860-0980. https://doi.org/10.1007/BF02289026 (1953).

55. Alanis-Lobato, G., Mier, P. & Andrade-Navarro, M. A. Manifold learning and maximum likelihood estimation for hyperbolic network embedding. *Appl. Netw. Sci.* **1,** 10 (2016).

56. Papadopoulos, F., Aldecoa, R. & Krioukov, D. Network geometry inference using common neighbors. *Phys. Rev. E* **92,** 022807. https://doi.org/10.1103/PhysRevE.92.022807 (2015).

57. Kitsak, M., Voitalov, I. & Krioukov, D. Link prediction with hyperbolic geometry. *Phys. Rev. Research* **2,** 043113. https://doi.org/10.1103/PhysRevResearch.2.043113 (2020).

58. Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: First steps. *Social Networks* **5,** 109–137. ISSN: 0378-8733. https://doi.org/10.1016/0378-8733(83)90021-7 (1983).

59. Wang, Y. J. & Wong, G. Y. Stochastic Blockmodels for Directed Graphs. *Journal of the American Statistical Association* **82.** Full publication date: Mar., 1987, 8–19. ISSN: 01621459. https://doi.org/10.2307/2289119 (1987).

60. *For generating graphs with the stochastic block model, I used the Python function 'stochastic_block_model' available in the 'NetworkX' package at https://networkx.org/documentation/stable/reference/generated/networkx.generators.community.stochastic_block_model.html.*

61. Zhang, Y.-J., Yang, K.-C. & Radicchi, F. Systematic comparison of graph embedding methods in practical tasks. *Phys. Rev. E* **104,** 044315. https://doi.org/10.1103/PhysRevE.104.044315 (2021).

62. *The Spearman's correlation coefficients were calculated with the Python function 'spearmanr' available in the 'scipy.stats' package at https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html.*

63. Goyal, P. & Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* **151,** 78–94. ISSN: 0950-7051. https://doi.org/10.1016/j.knosys.2018.03.022 (2018).

64. Liben-Nowell, D. & Kleinberg, J. *The Link Prediction Problem for Social Networks* in *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (Association for Computing Machinery, New Orleans, LA, USA, 2003), 556–559. ISBN: 1581137230. https://doi.org/10.1145/956863.956972.

65. Huang, Z., Li, X. & Chen, H. *Link Prediction Approach to Collaborative Filtering* in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (Association for Computing Machinery, Denver, CO, USA, 2005), 141–142. ISBN: 1581138768. https://doi.org/10.1145/1065385.1065415.

66. Zhou, T., Lü, L. & Zhang, Y.-C. Predicting missing links via local information. *The European Physical Journal B* **71,** 623–630. ISSN: 1434-6036. https://doi.org/10.1140/epjb/e2009-00335-8 (2009).

67. *I computed the precision-recall pairs for different probability thresholds with the Python function 'precision_recall_curve' available in the 'sklearn.metrics' package at https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html.*

68. *I calculated the area under the PR and the ROC curves with the Python function 'auc' available in the 'sklearn.metrics' package at https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html.*

69. Boyd, K., Eng, K. H. & Page, C. D. *Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals* in *Machine Learning and Knowledge Discovery in Databases* (eds Blockeel, H., Kersting, K., Nijssen, S. & Železný, F.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), 451–466. ISBN: 978-3-642-40994-3. https://doi.org/10.1007/978-3-642-40994-3_29.

70. *I computed the receiver operating characteristic curve with the Python function 'roc_curve' available in the 'sklearn.metrics' package at https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html.*

71. Hanley, J. A. & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. en. *Radiology* **143,** 29–36. https://doi.org/10.1148/radiology.143.1.7063747 (Apr. 1982).

72. Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. *Knowledge and Information Systems* **45,** 751–782. ISSN: 0219-3116. https://doi.org/10.1007/s10115-014-0789-0 (2015).

73. Sinha, A., Cazabet, R. & Vaudaine, R. *Systematic Biases in Link Prediction: Comparing Heuristic and Graph Embedding Based Methods* in *Complex Networks and Their Applications VII* (eds Aiello, L. M. *et al.*) (Springer International Publishing, Cham, 2019), 81–93. ISBN: 978-3-030-05411-3. https://doi.org/10.1007/978-3-030-05411-3_7.

74. Heaberlin, B. & DeDeo, S. The Evolution of Wikipedia's Norm Network. *Future Internet* **8.** ISSN: 1999-5903. https://doi.org/10.3390/fi8020014 (2016).

75. *The yeast transcription network that I used for testing my embedding methods was downloaded from https://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks.*

76. Costanzo, M. C. *et al.* YPD, PombePD and WormPD: model organism volumes of the BioKnowledge library, an integrated resource for protein information. en. *Nucleic Acids Res.* **29,** 75–79. https://doi.org/10.1093/nar/29.1.75 (Jan. 2001).

77. *The word association network that I used for testing my embedding methods was downloaded from http://w3.usf.edu/FreeAssociation/.*

78. Nelson, D. L., McEvoy, C. L. & Schreiber, T. A. The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers* **36,** 402–407. https://doi.org/10.3758/BF03195588 (Aug. 2004).

79. *The American College Football network was downloaded from http://konect.cc/networks/dimacs10-football/.*

80. Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99,** 7821–7826. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.122653799. https://doi.org/10.1073/pnas.122653799 (2002).

81. Coşkun, M., Baggag, A. & Koyutürk, M. Fast computation of Katz index for efficient processing of link prediction queries. *Data Mining and Knowledge Discovery* **35,** 1342–1368. ISSN: 1573-756X. https://doi.org/10.1007/s10618-021-00754-8 (2021).

82. *The neural network that I used for testing my embedding methods was downloaded from https://networkdata.ics.uci.edu/data/celegansneural/.*

83. White, J. G., Southgate, E, Thomson, J. N. & Brenner, S. The structure of the nervous system of the nematode Caenorhabditis elegans. en. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **314,** 1–340. https://doi.org/10.1098/rstb.1986.0056 (Nov. 1986).

84. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393,** 440–442. ISSN: 1476-4687. https://doi.org/10.1038/30918 (1998).

85. Kasyanov, I., van der Hoorn, P., Krioukov, D. & Tamm, M. *Nearest-neighbour directed random hyperbolic graphs* Preprint at arXiv:2303.01002 [physics.soc-ph] (2023). https://doi.org/10.48550/arXiv.2303.01002.

86. Battiston, F. *et al.* Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports* **874,** 1–92. ISSN: 0370-1573. https://doi.org/10.1016/j.physrep.2020.05.004 (2020).

87. Bick, C., Gross, E., Harrington, H. A. & Schaub, M. T. *What are higher-order networks?* Preprint at arXiv:2104.11329 [cs.SI] (2022). https://doi.org/10.48550/arXiv.2104.11329.

88. MacQueen, J. B. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* **1,** 281–297 (1967).

89. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise* in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (AAAI Press, Portland, Oregon, 1996), 226–231.

90. Zhang, T., Ramakrishnan, R. & Livny, M. *BIRCH: An Efficient Data Clustering Method for Very Large Databases* in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (Association for Computing Machinery, Montreal, Quebec, Canada, 1996), 103–114. ISBN: 0897917944. https://doi.org/10.1145/233269.233324.

91. Tandon, A. *et al.* Community detection in networks using graph embeddings. *Phys. Rev. E* **103,** 022316. https://doi.org/10.1103/PhysRevE.103.022316 (2021).