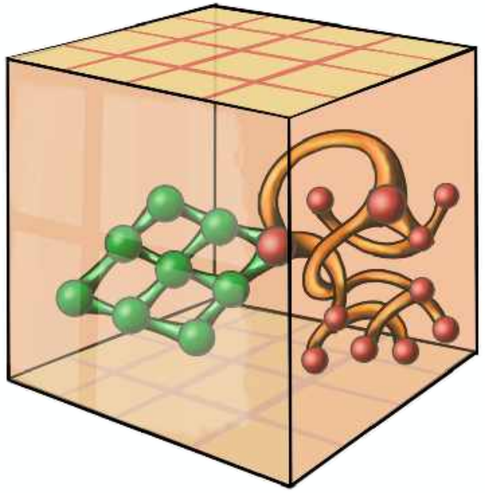
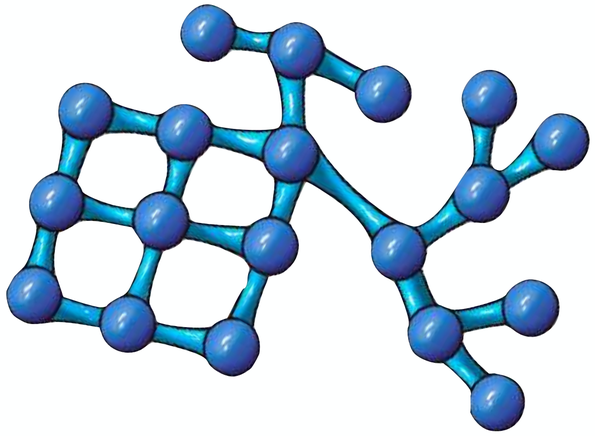
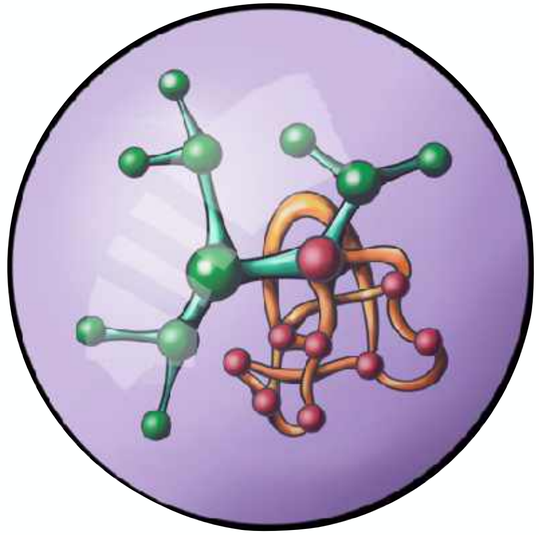
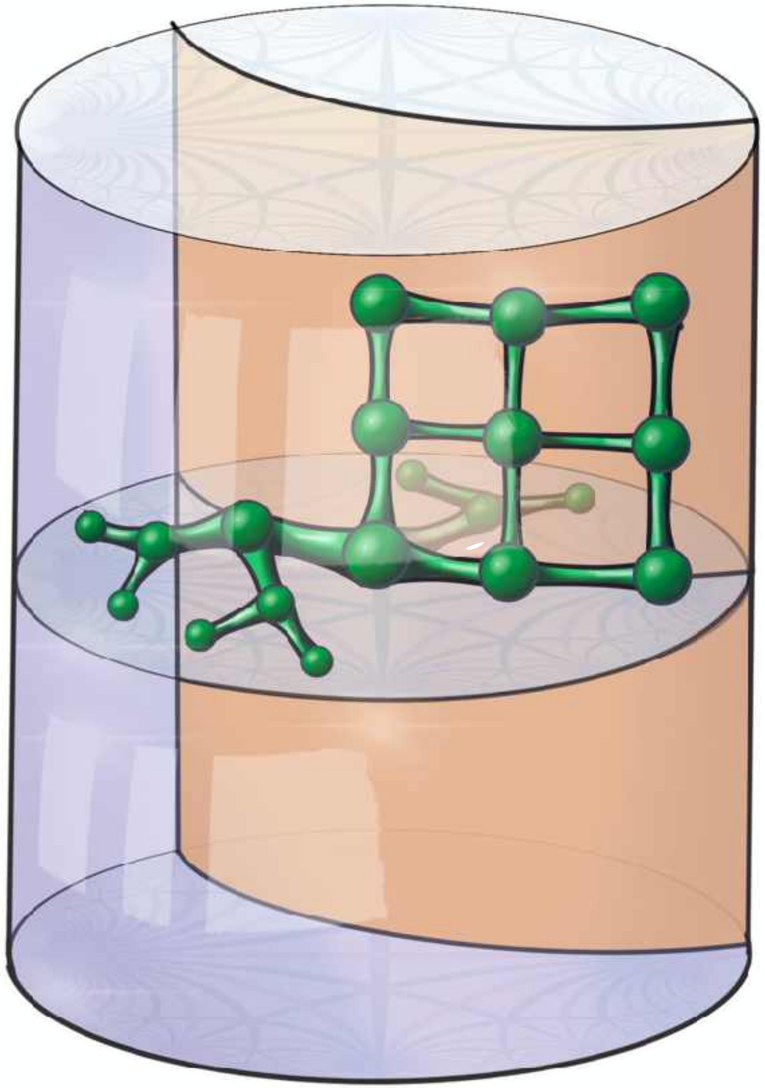


Modern Geometry &

Machine Learning







Steve Trettel, USF

Leveraging the geometry of Riemannian symmetric spaces for graph embeddings

Federico **Lopez**

Beatrice **Pozzetti**

Michael **Strube**

Steve **Trettel**

Anna **Wienhard**

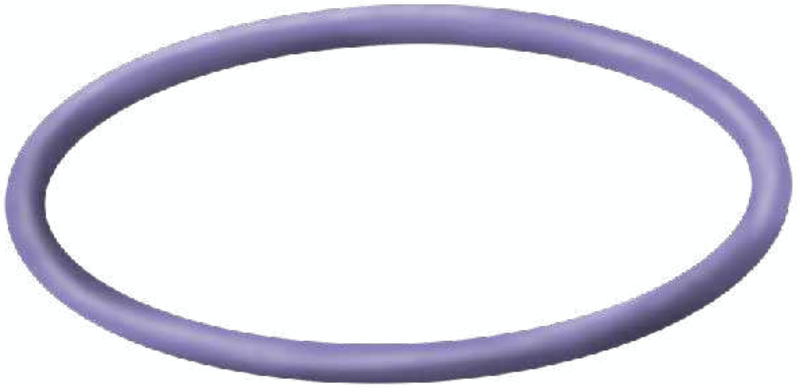
Max **Reistenberg**

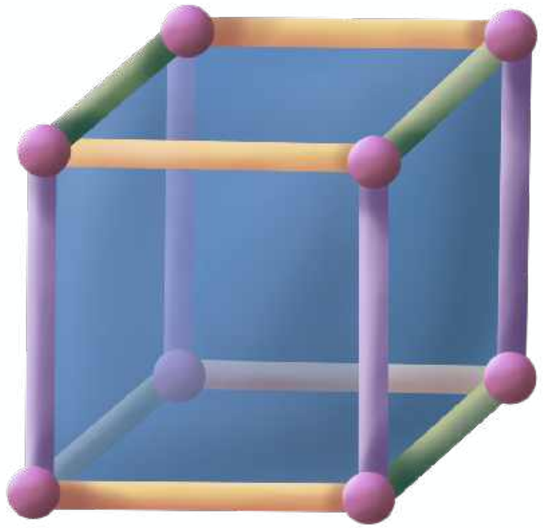
Diaaeldin **Taha**

Wei **Zhao**

The Ideas of Modern Geometry

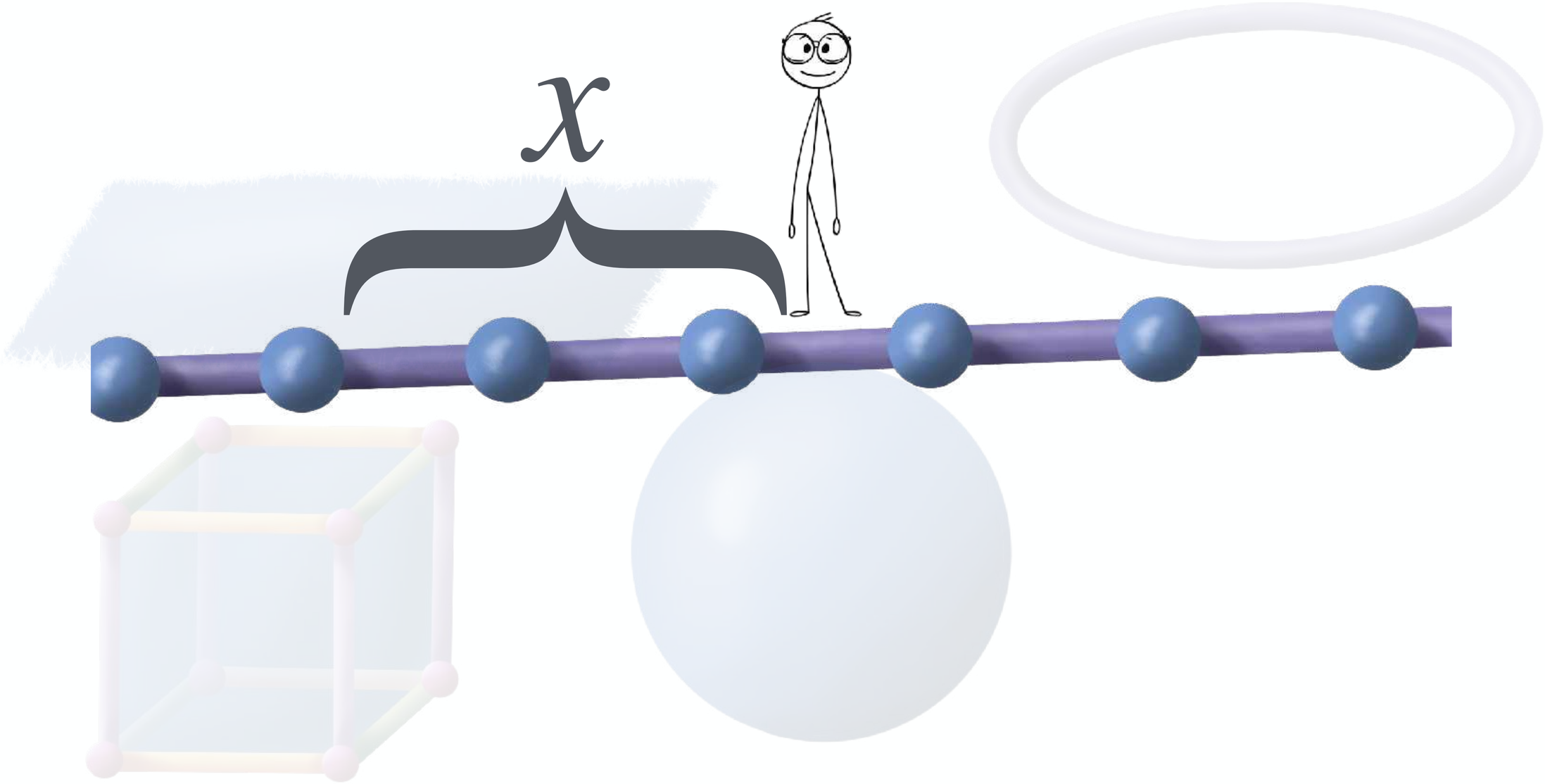
High Dimensions & Curved Spaces

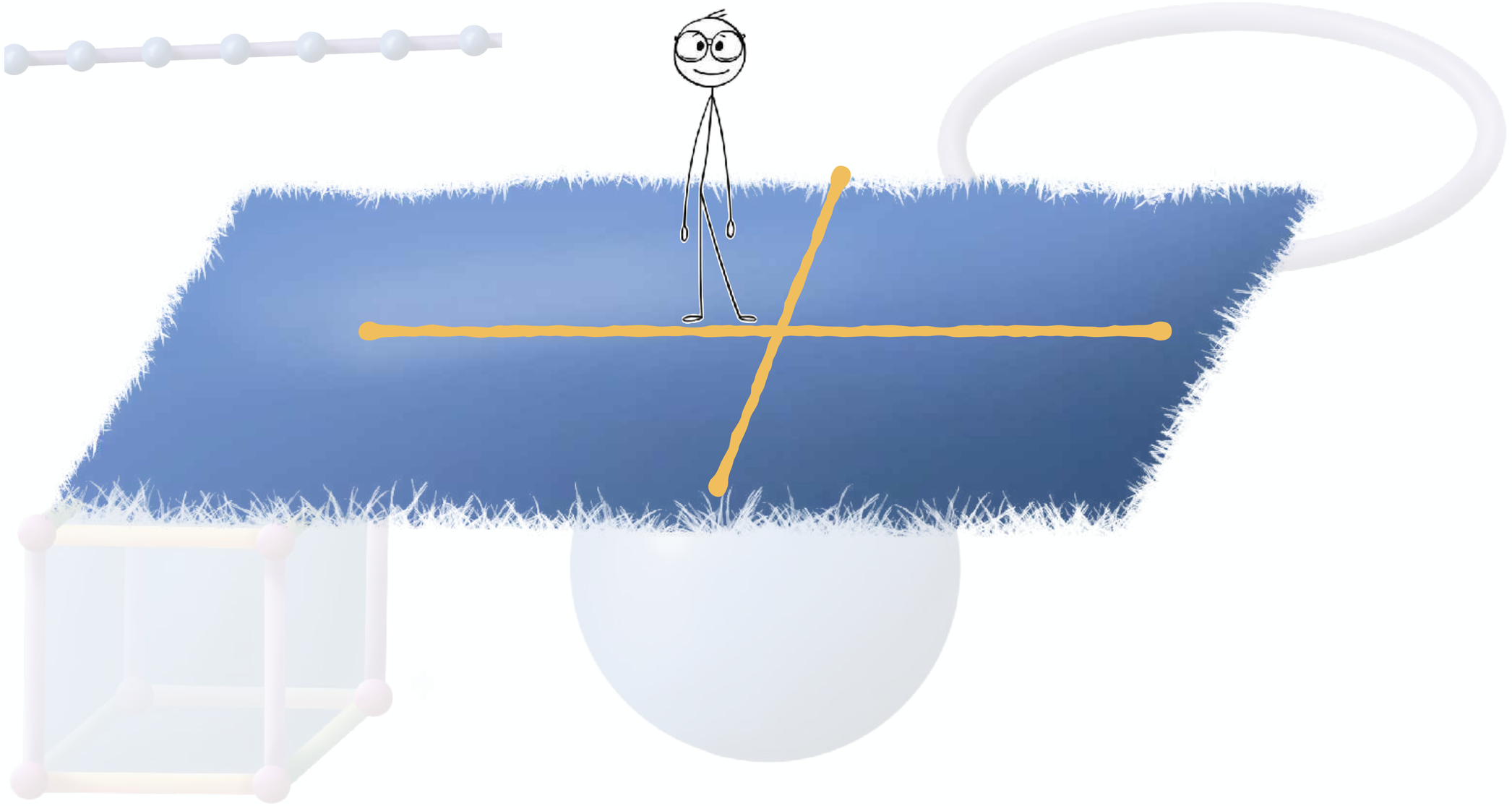


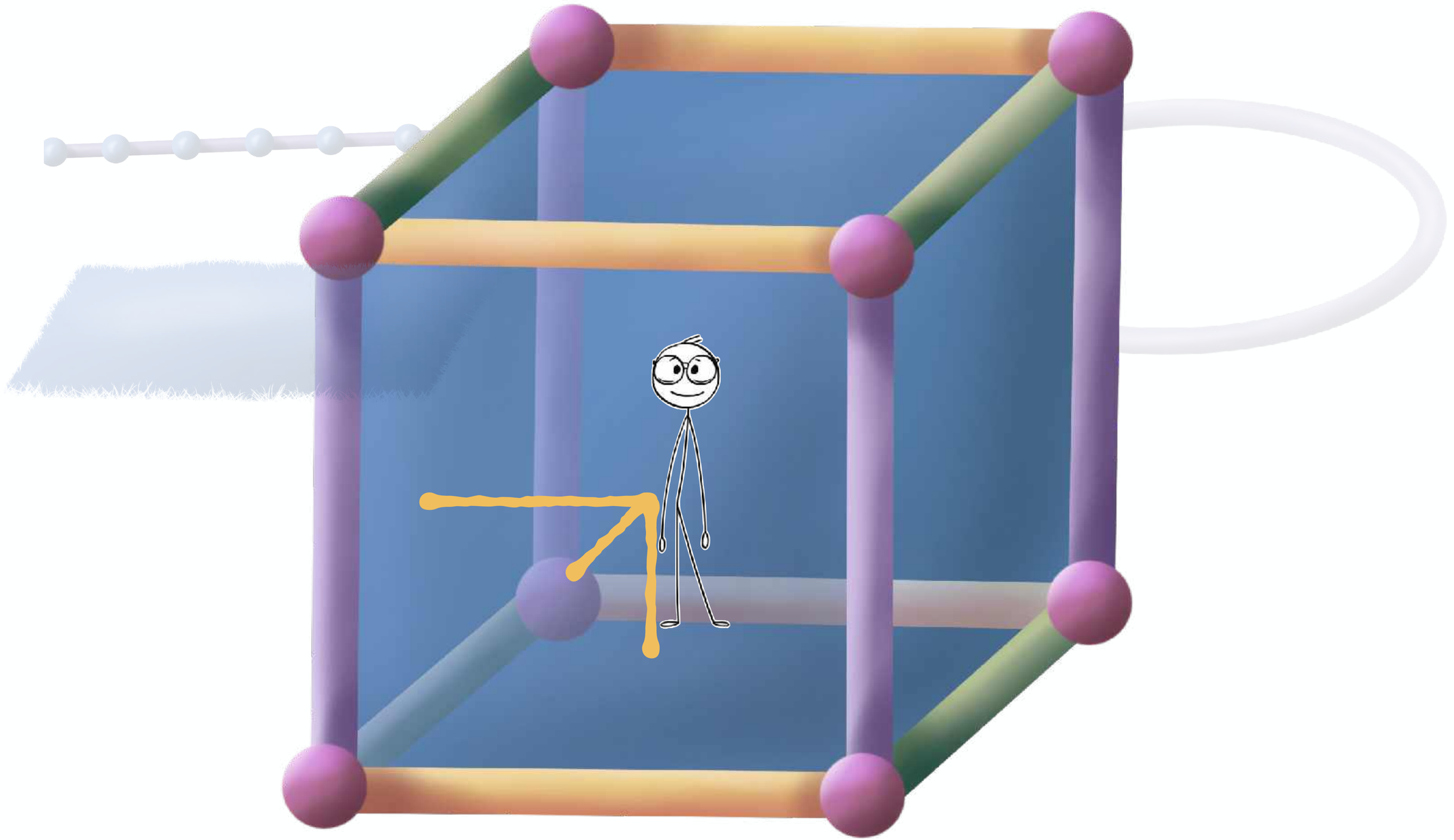


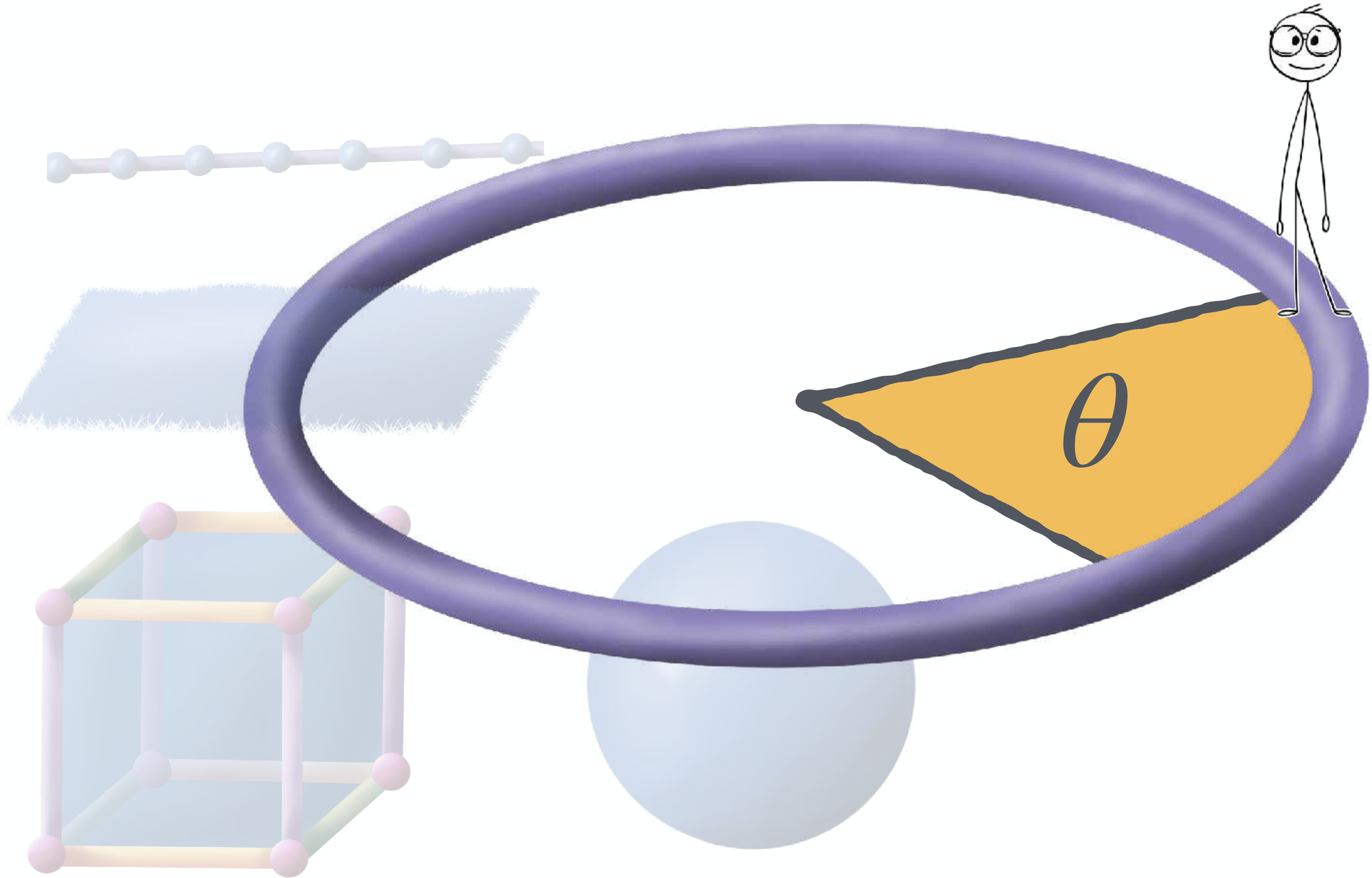
What is dimension?

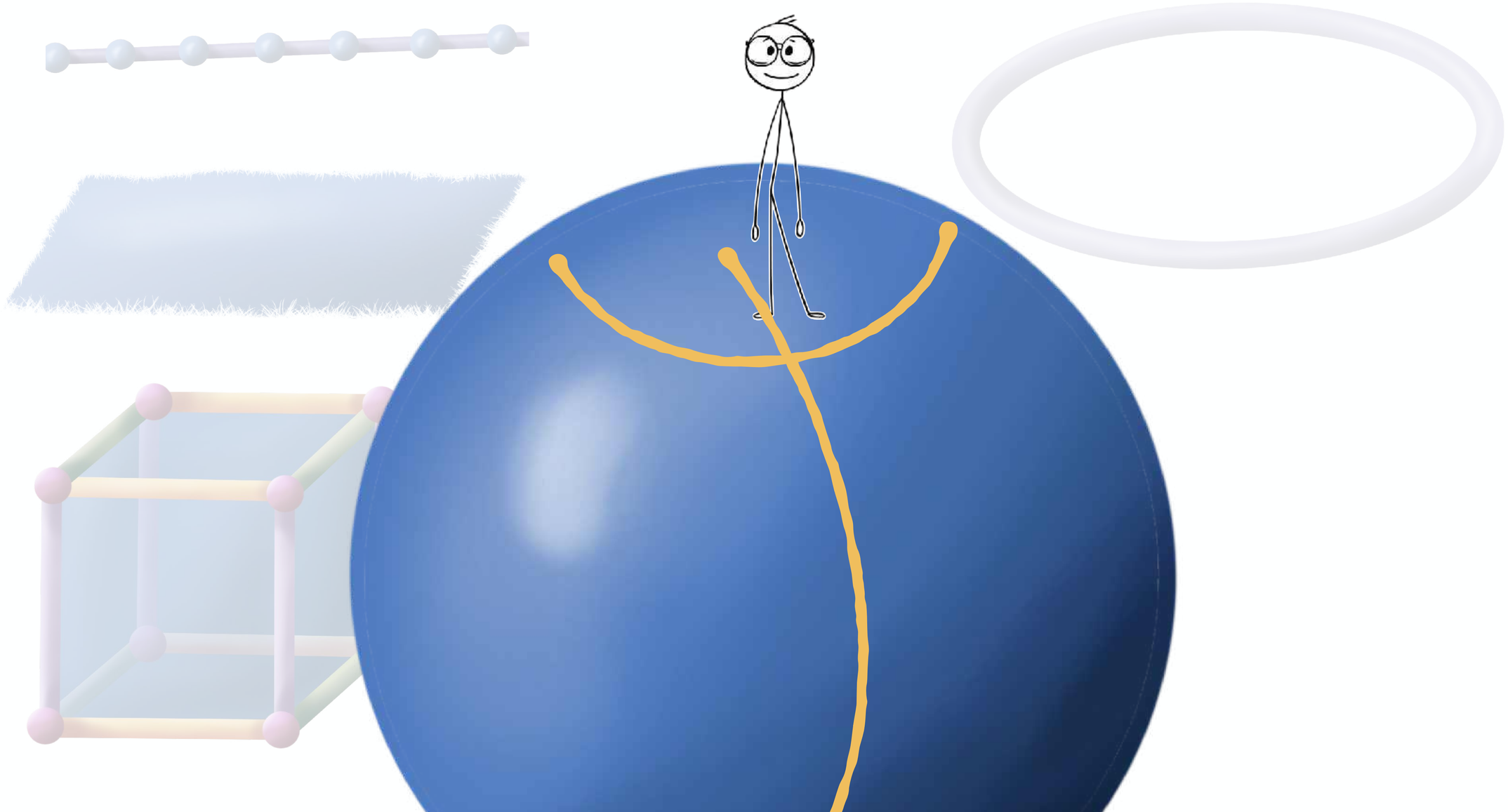
The number of pieces of information you need to completely specify a point.


x



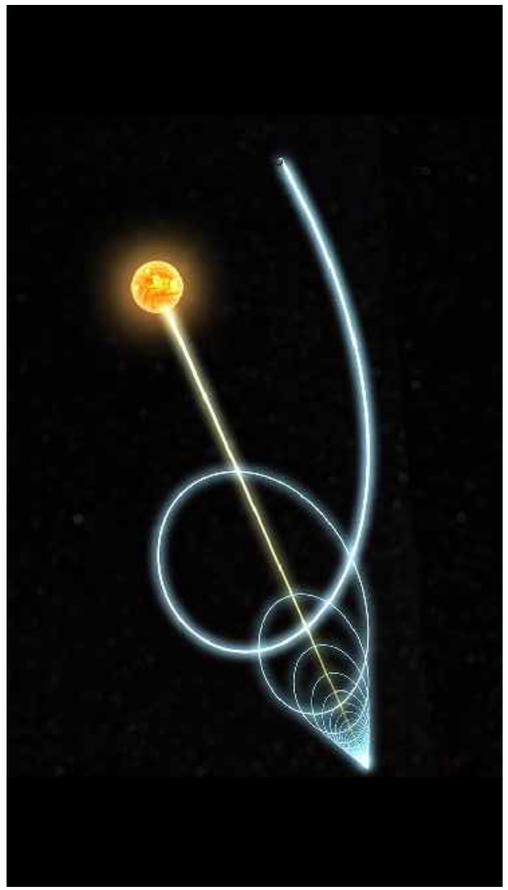


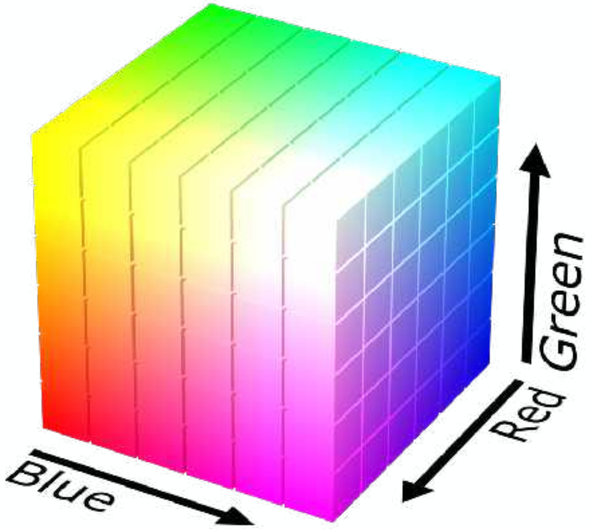

θ



This opened our eyes to a much broader world

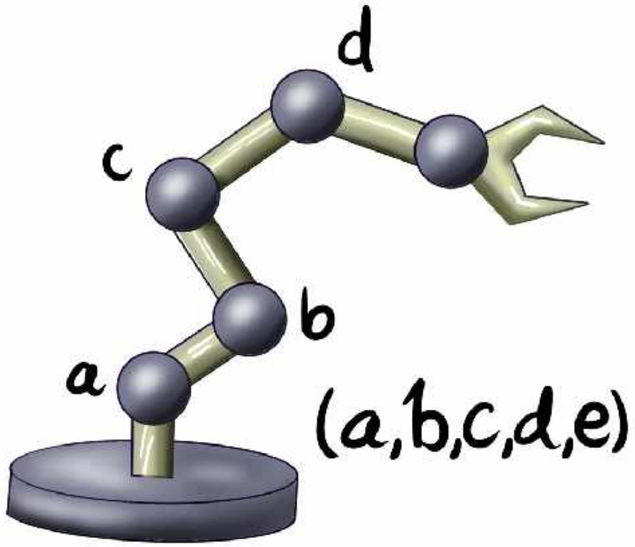
Spacetime is 4D



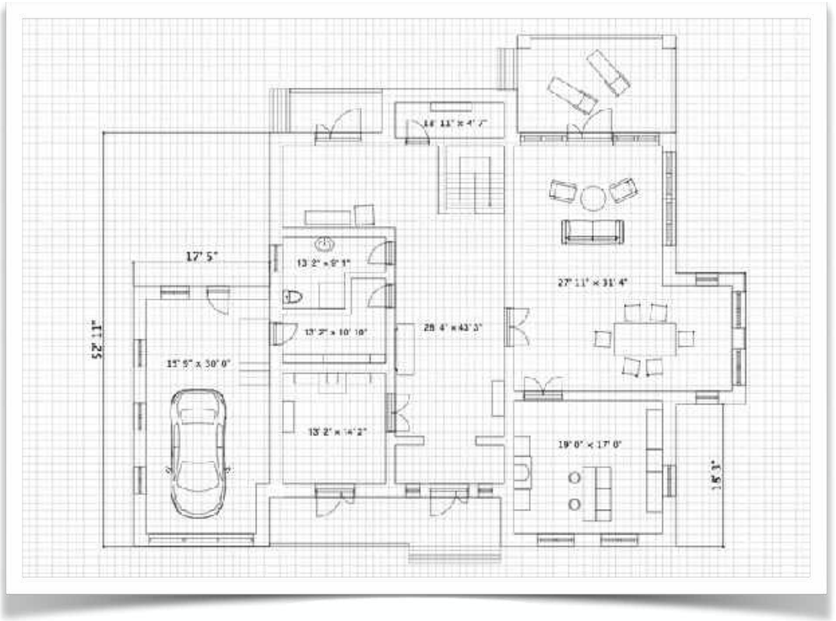


Space of colors is 3D

Configuration spaces of robots, etc

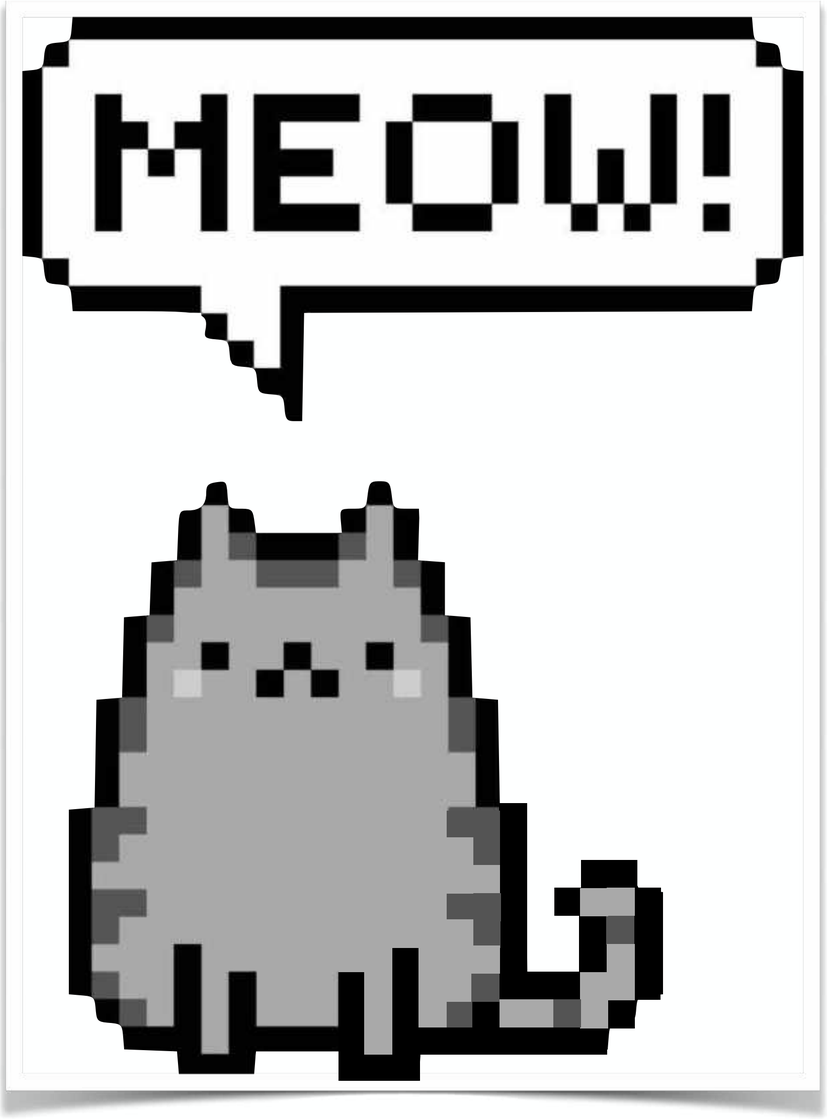


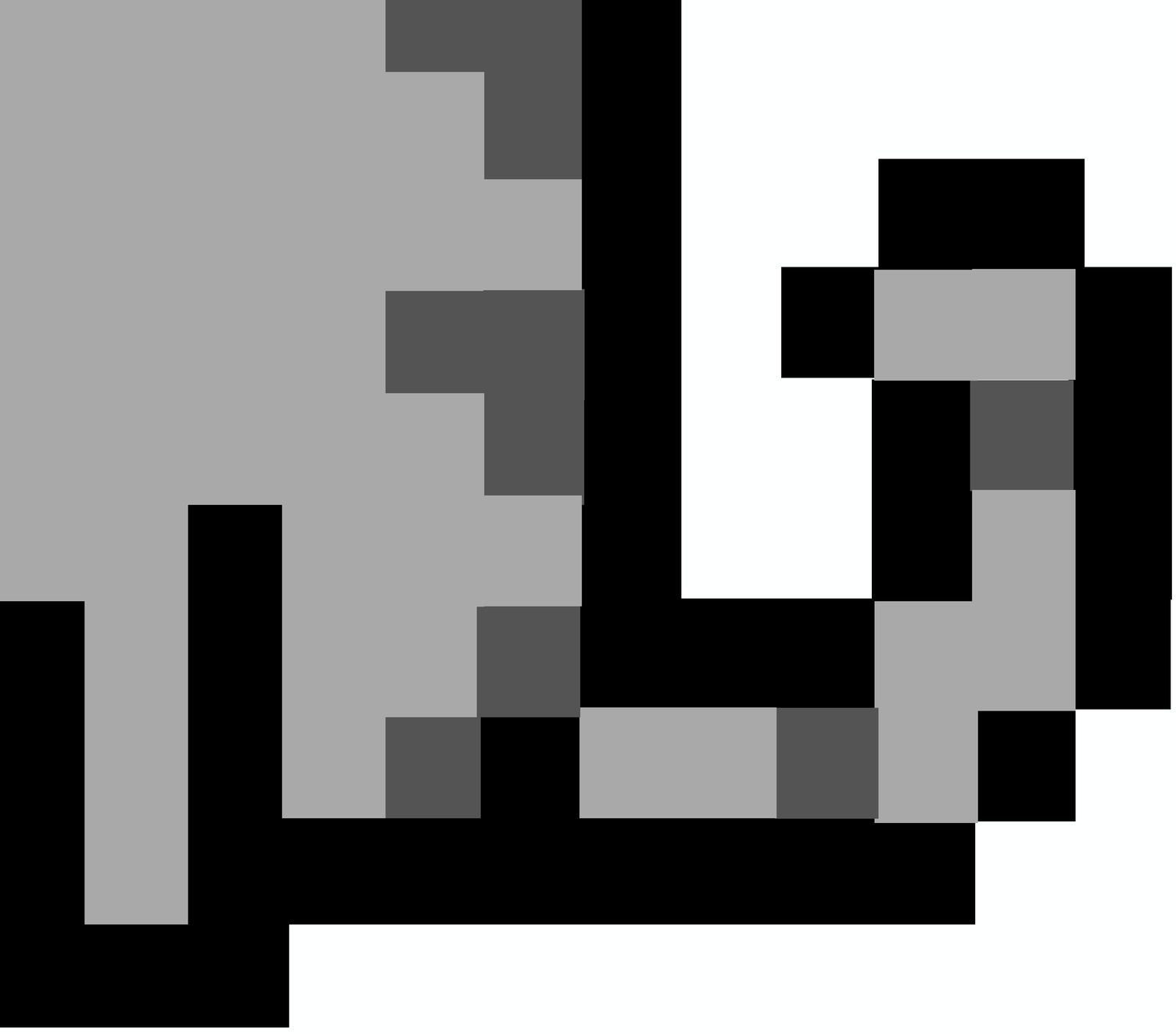
Parameter spaces of design

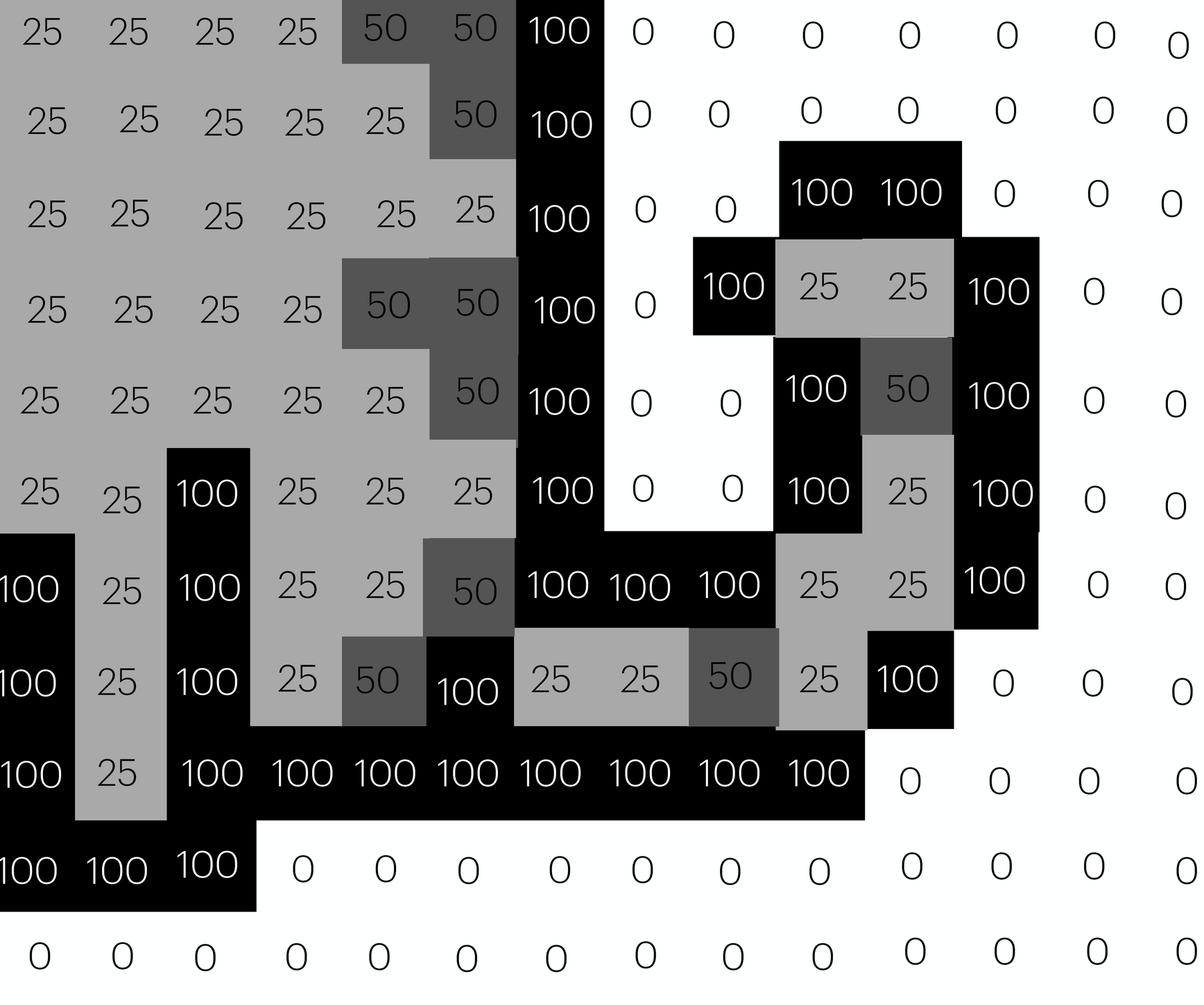


An example: the space of photographs.

An image is a point in high dimensional space



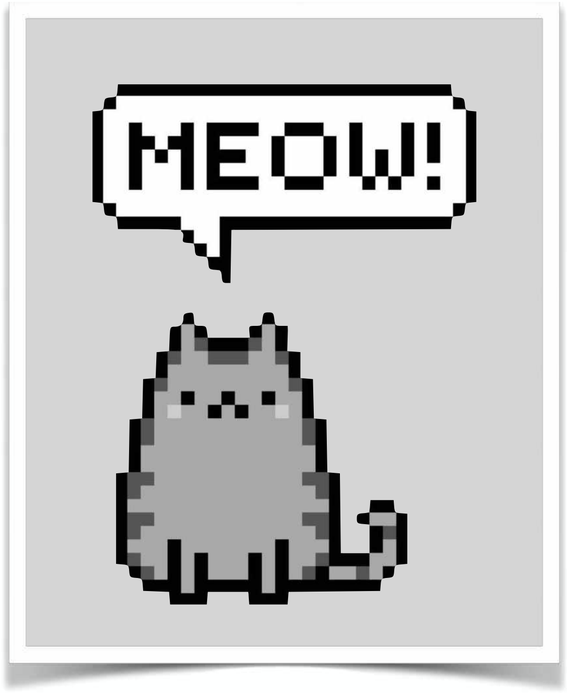



25 25 25 25 50 50 100 0 0 0 0 0 0 0
25 25 25 25 25 50 100 0 0 0 0 0 0 0
25 25 25 25 25 25 100 0 0 100 100 0 0 0
25 25 25 25 50 50 100 0 100 25 25 100 0 0
25 25 25 25 25 50 100 0 0 100 50 100 0 0
25 25 100 25 25 25 100 0 0 100 25 100 0 0
100 25 100 25 25 50 100 100 100 25 25 100 0 0
100 25 100 25 50 100 25 25 50 25 100 0 0 0
100 25 100 100 100 100 100 100 100 100 0 0 0 0
100 100 100 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

25	25	25	25	50	50	100	0	0	0	0	0	0	0
25	25	25	25	25	50	100	0	0	0	0	0	0	0
25	25	25	25	25	25	100	0	0	100	100	0	0	0
25	25	25	25	50	50	100	0	100	25	25	100	0	0
25	25	25	25	25	50	100	0	0	100	50	100	0	0
25	25	100	25	25	25	100	0	0	100	25	100	0	0
100	25	100	25	25	50	100	100	100	25	25	100	0	0
100	25	100	25	50	100	25	25	50	25	100	0	0	0
100	25	100	100	100	100	100	100	100	100	0	0	0	0
100	100	100	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Dimension of the space of images

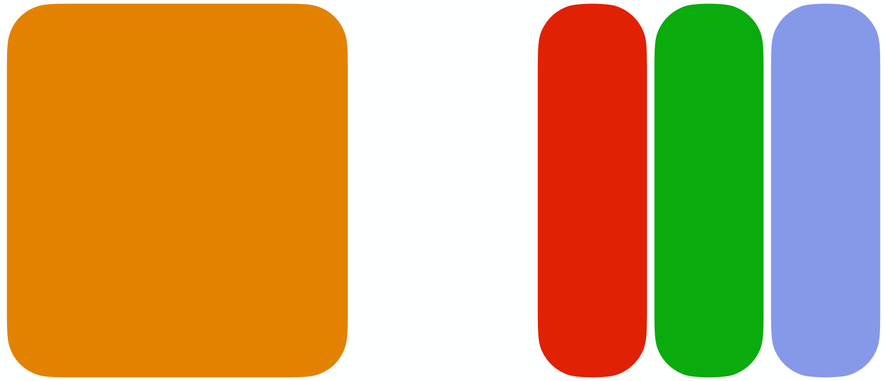
nm-dimensional



m Pixels

n Pixels

Color Pixels

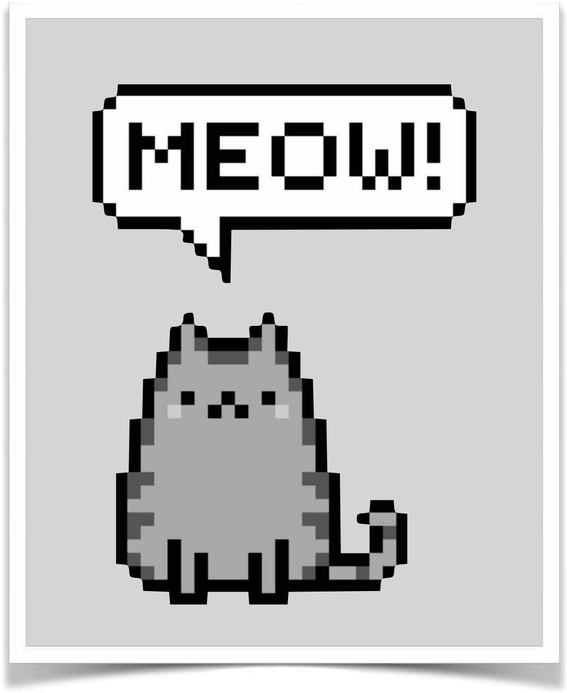


3 numbers per pixel: red, green, blue.

Space of images = 3nm-dimensional

Dimension of the space of images

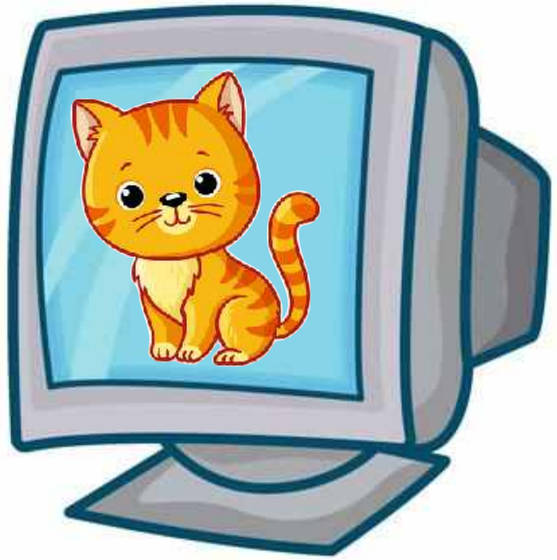
nm-dimensional



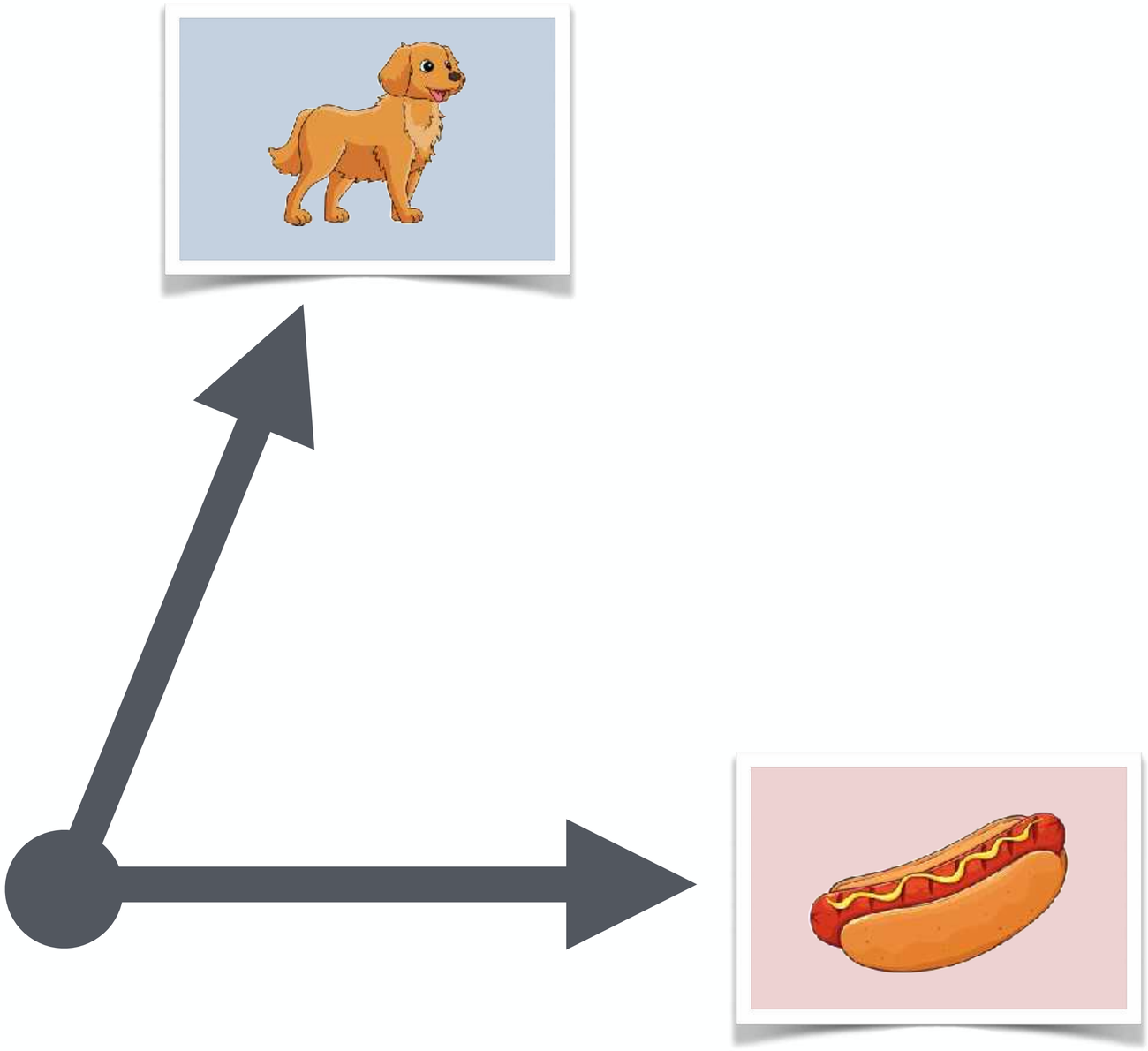
m Pixels

n Pixels

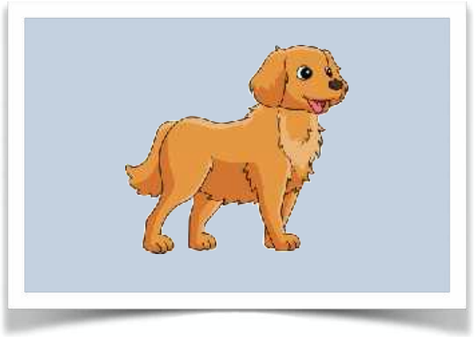
Our Example: 480p Color Images



3 x 480 x 320 = 460,800

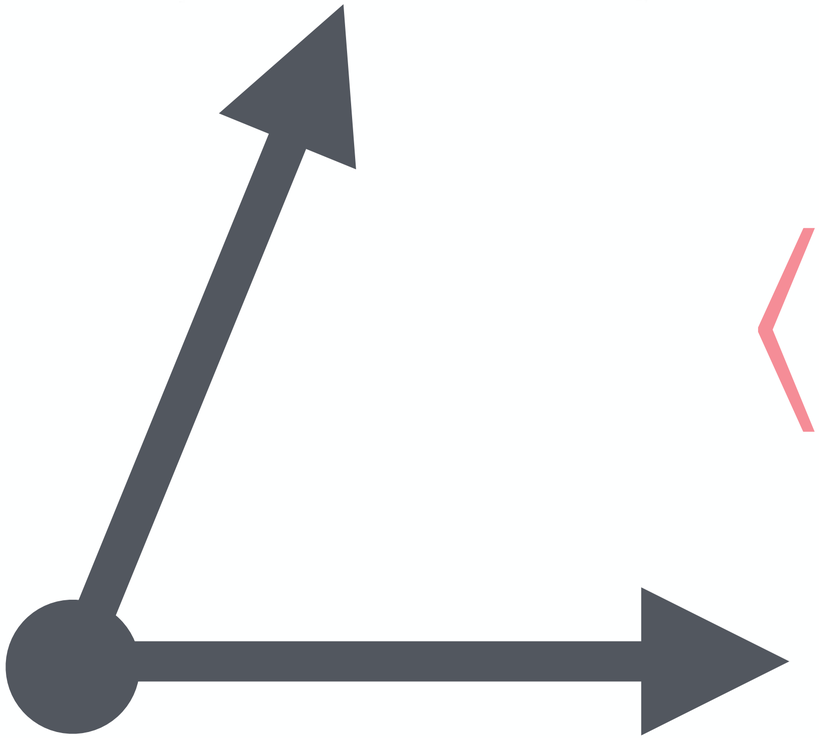


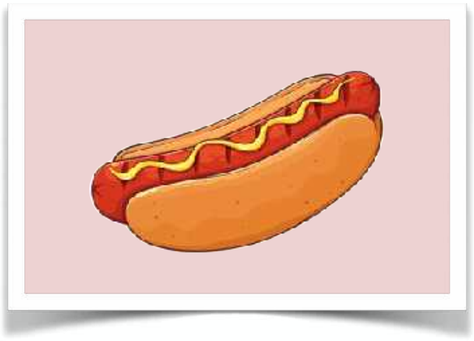
What is the *distance* between a dog and a hot dog?

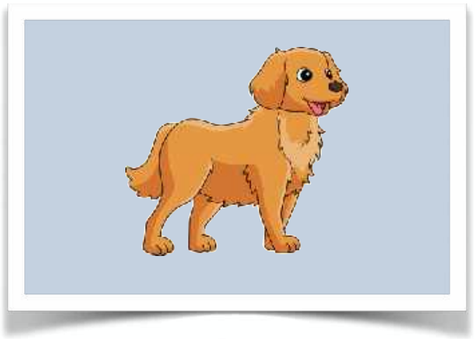


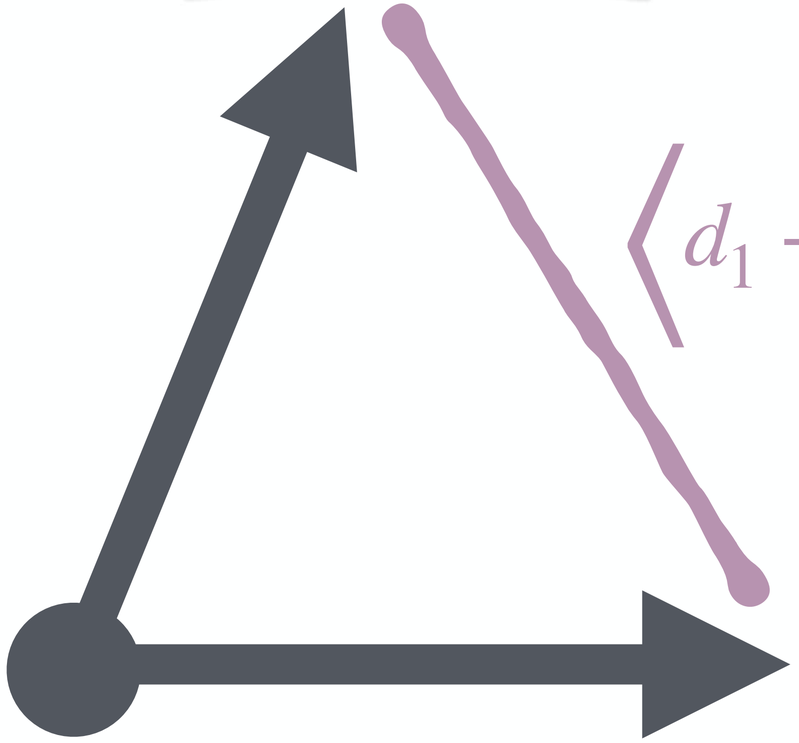
$$\langle d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8 \ldots \; d_{460,800} \rangle$$

$$\langle h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8 \ldots h_{460,800} \rangle$$

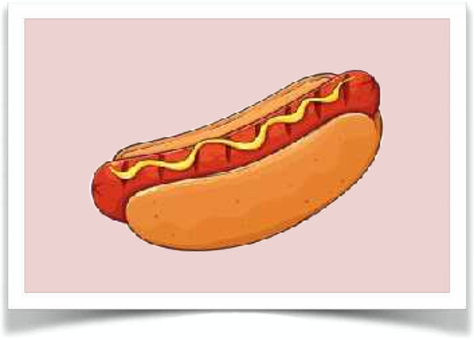


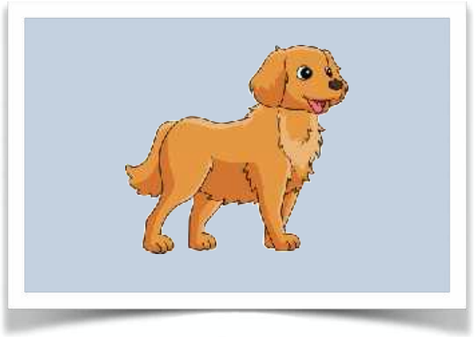




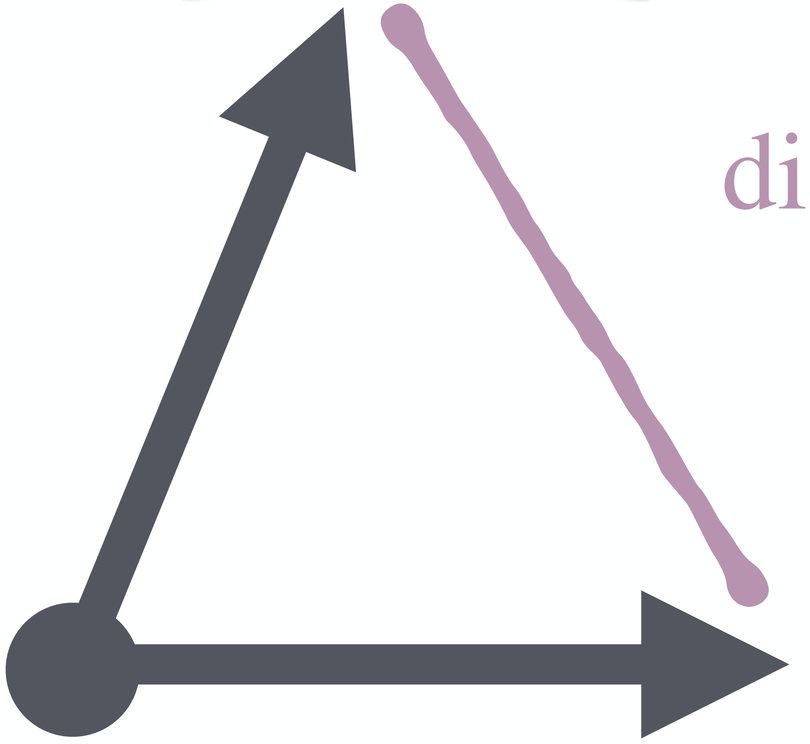


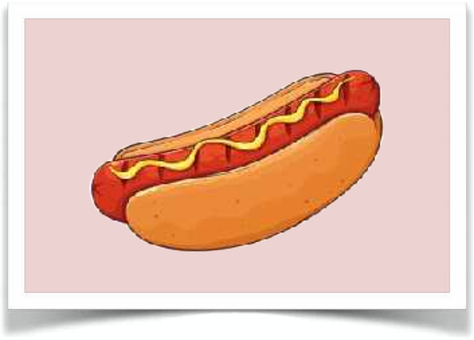
$$\langle d_1 - h_1, d_2 - h_2, \ldots \; d_{460,800} - h_{460,800} \rangle$$





$$\text{dist} = \sqrt{\sum_{i=1}^{460{,}800} (d_i - h_i^2)}$$



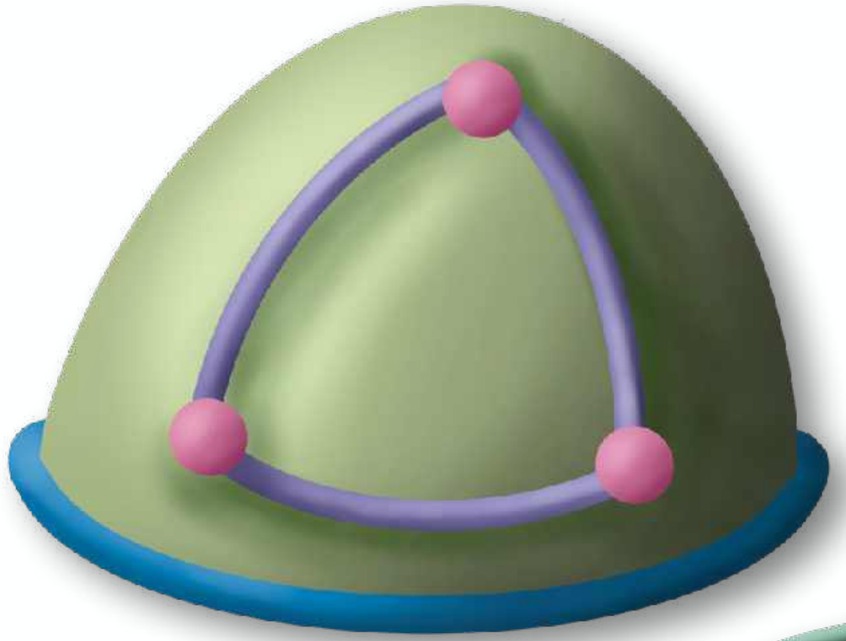


Thinking of the set of images as having *geometry* is the essential insight to many modern Machine Learning applications.

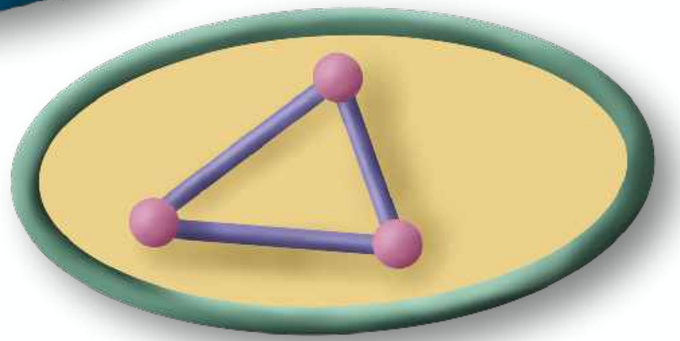
Space can be curved

Gauß

As a surveyor, Gauss had to think about ‘realistic’ geometry.

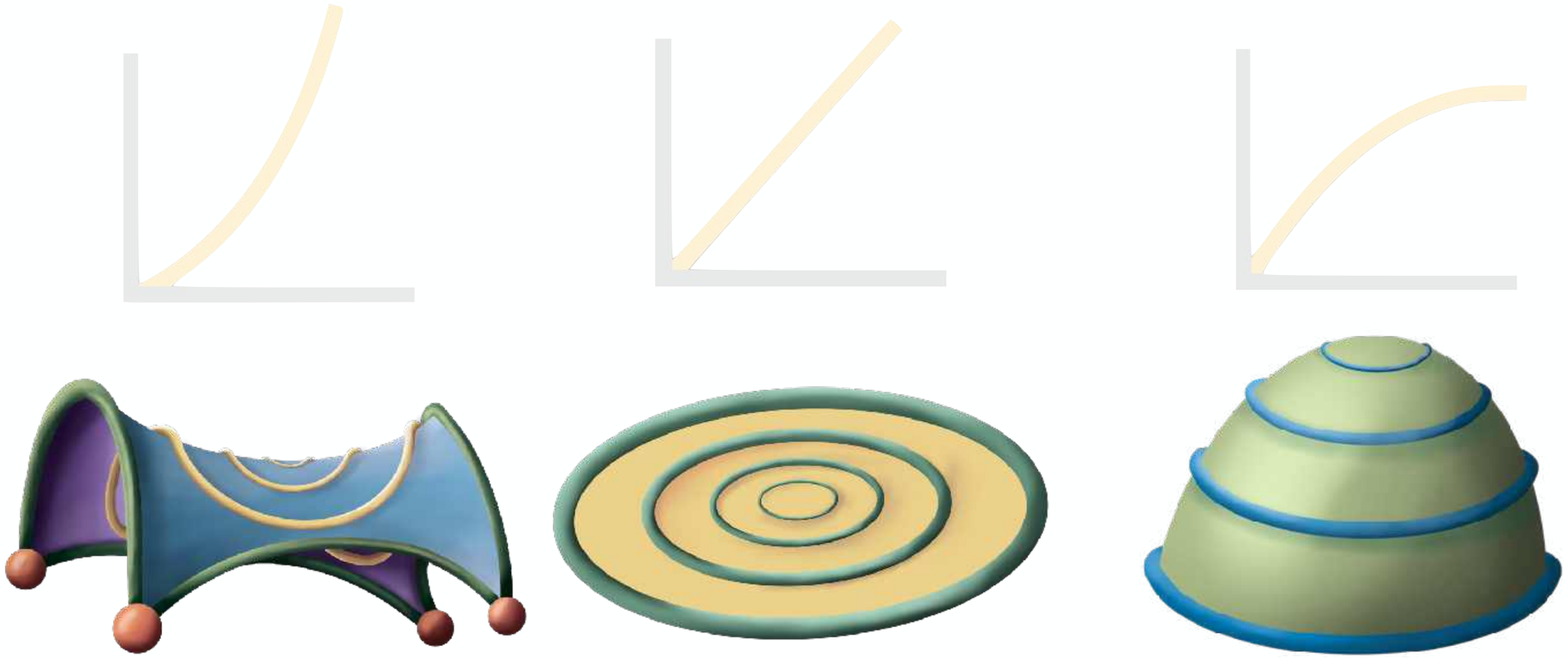


Geometry of a hilltop: triangles don’t add to 180!

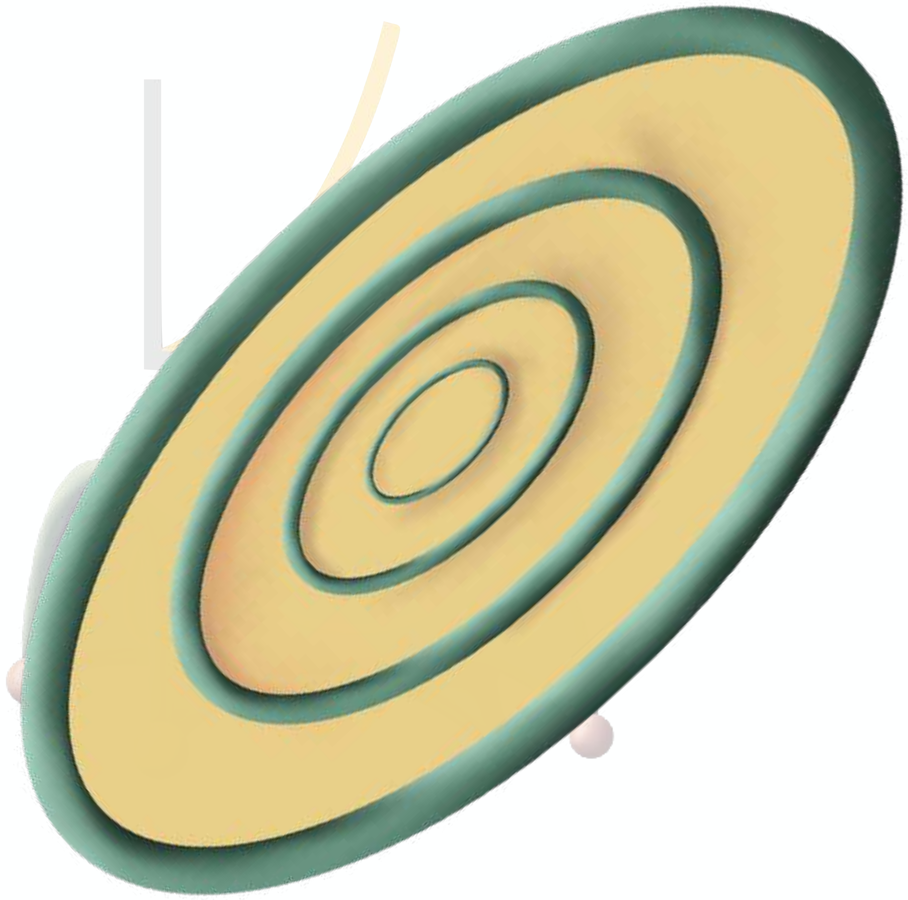


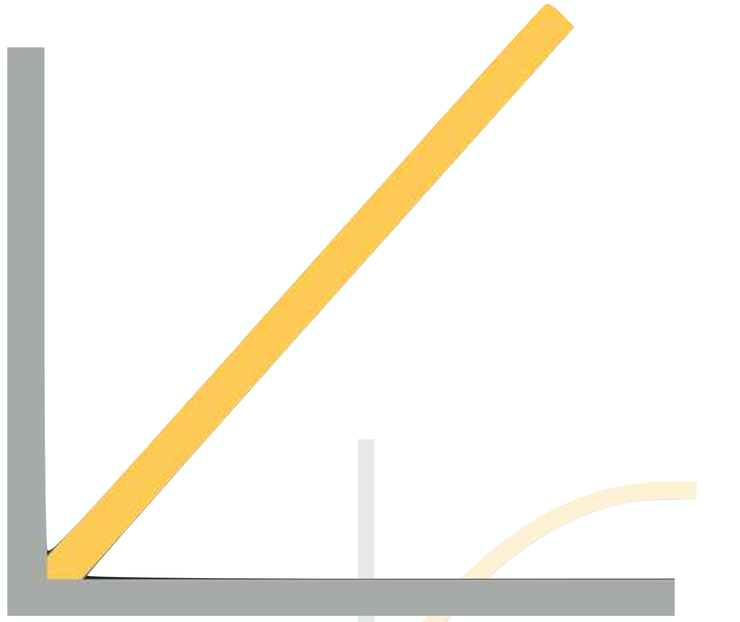
What is Curvature?

How can we quantify the deviation of a surface from the plane?



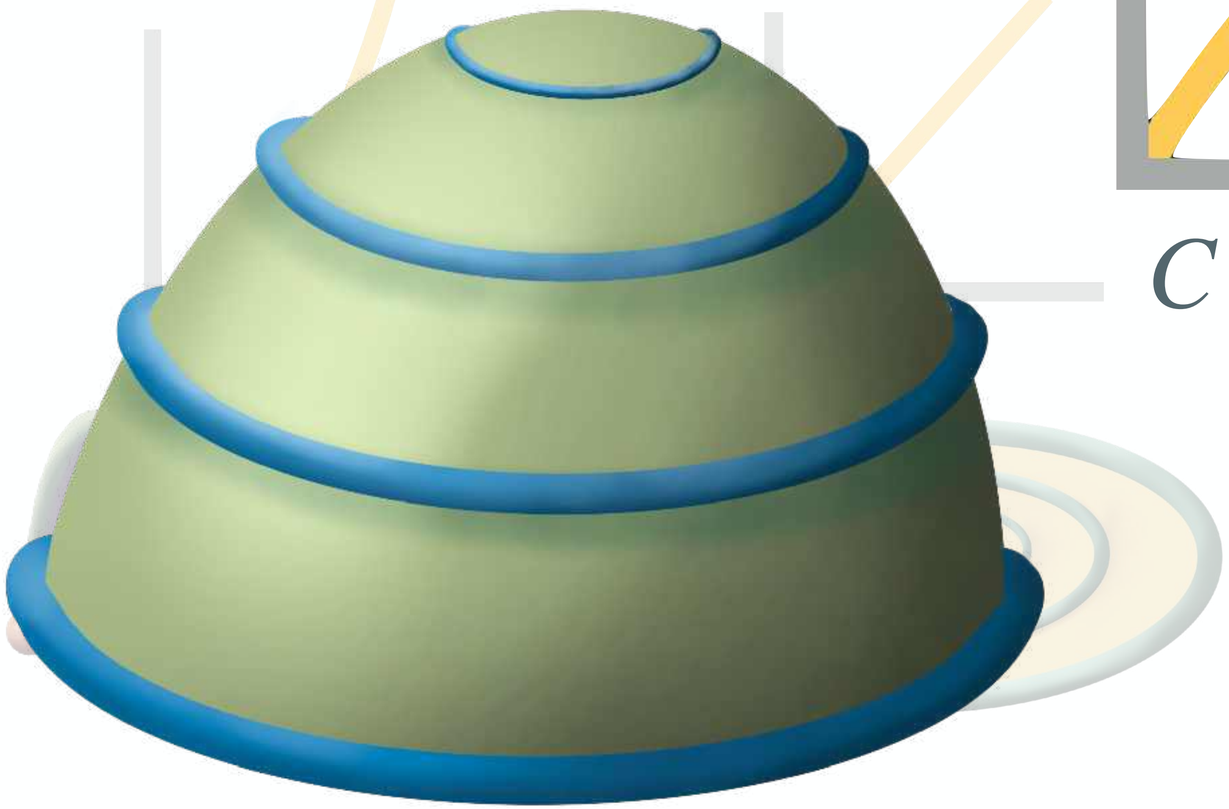
What is Curvature?

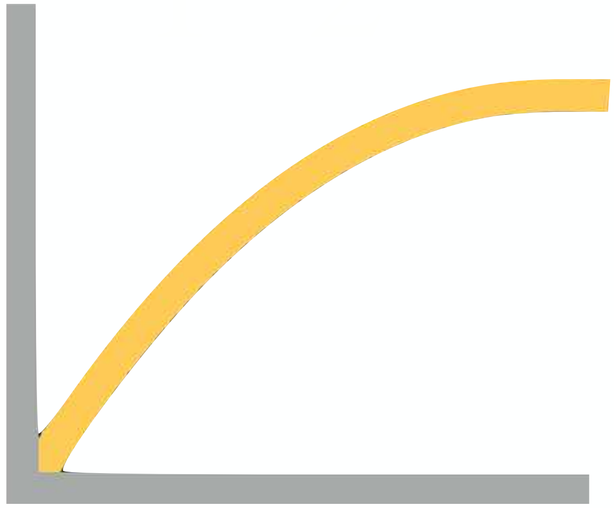




$$C = 2\pi r$$

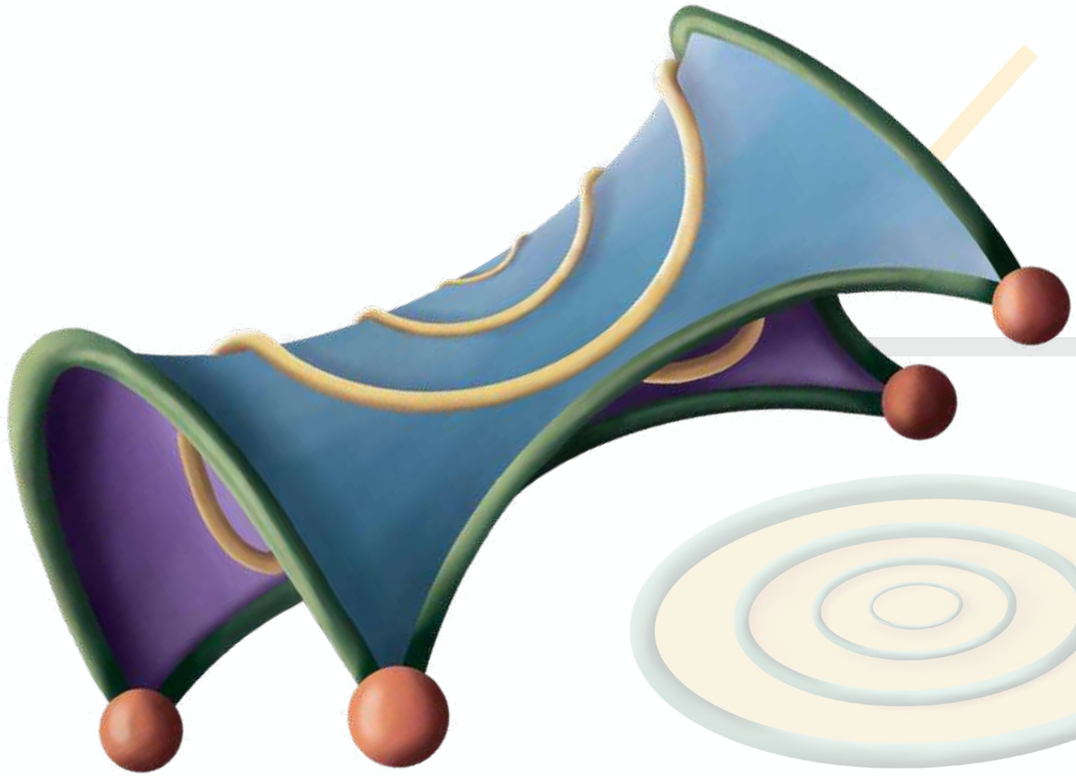
What is Curvature?

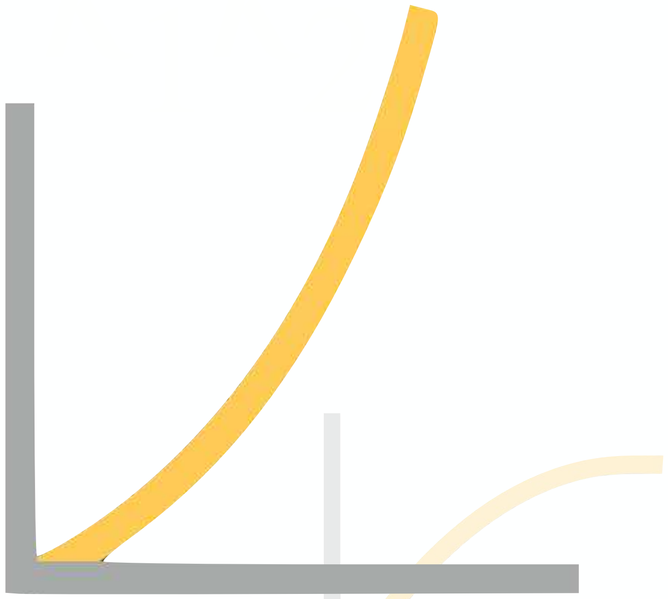




$$C = 2\pi \sin r$$

What is Curvature?



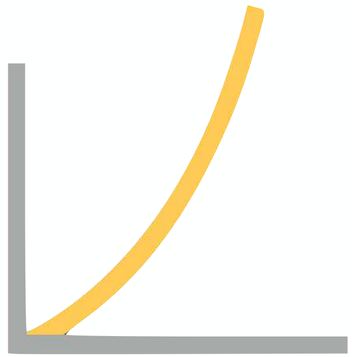


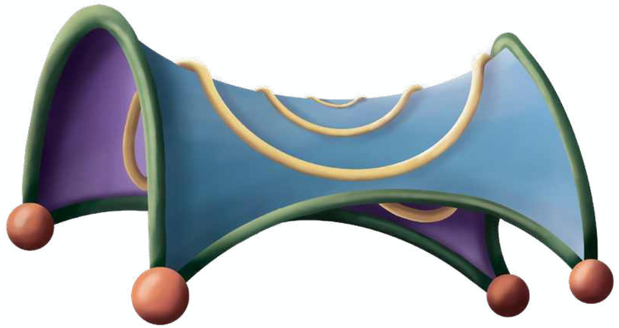
$$C = 2\pi \sinh r$$

What is Curvature?

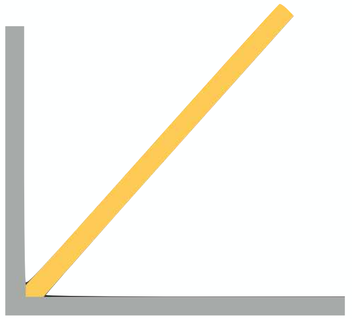
Curvature measures this difference.

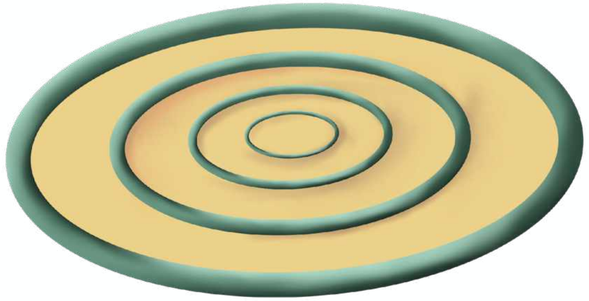
$$\kappa \propto \lim_{r \to 0} \frac{2\pi r - C(r)}{r^3}$$



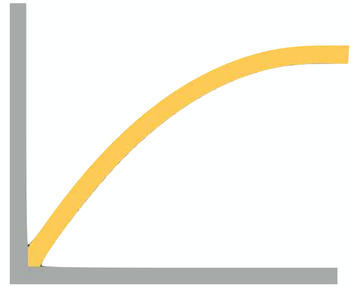


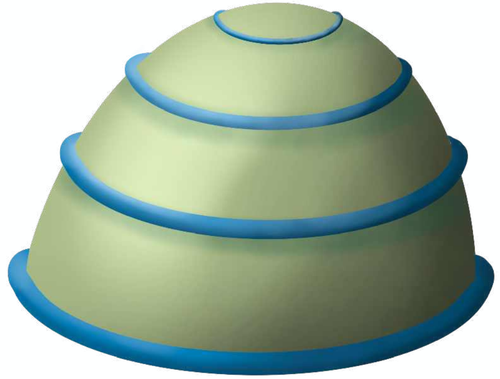
$\kappa < 0$



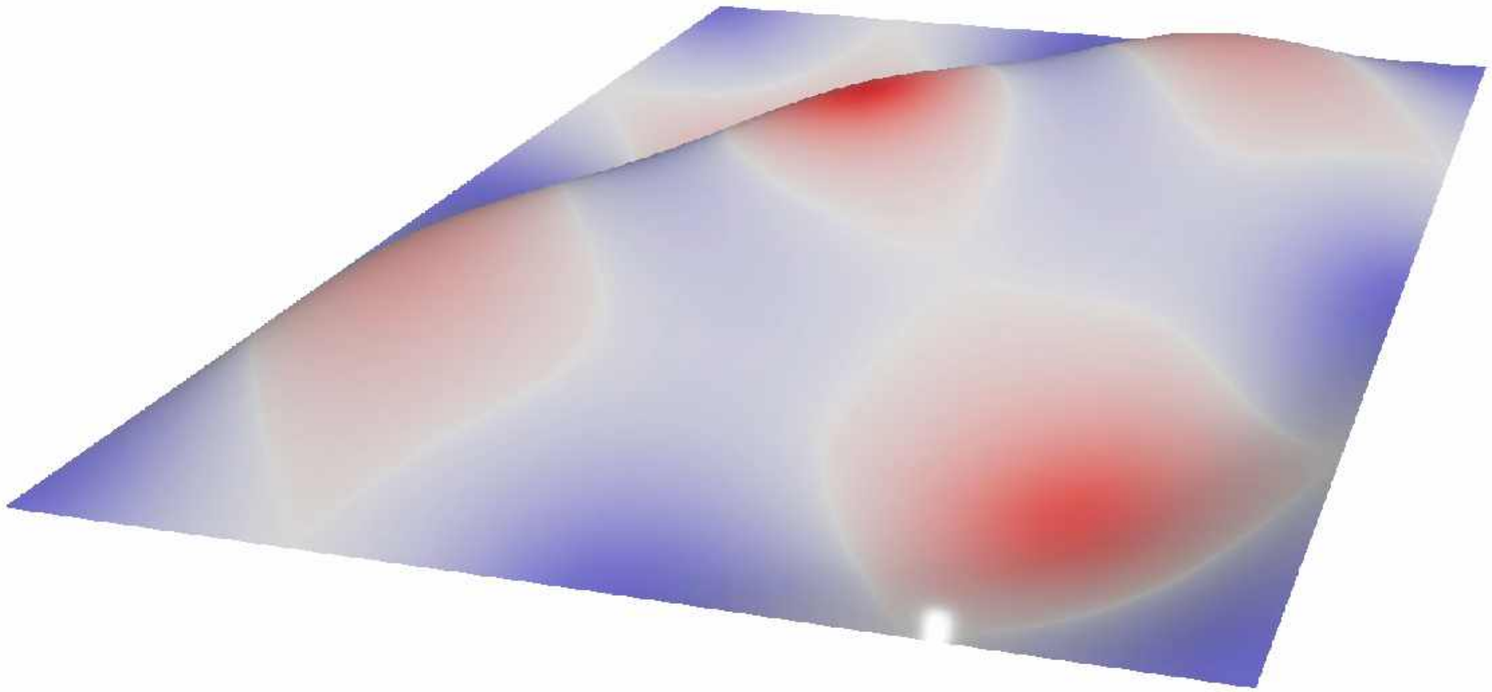


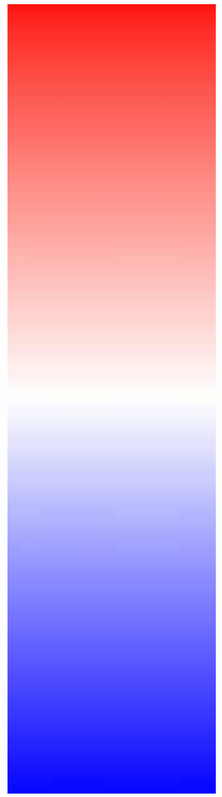
$\kappa = 0$





$\kappa > 0$

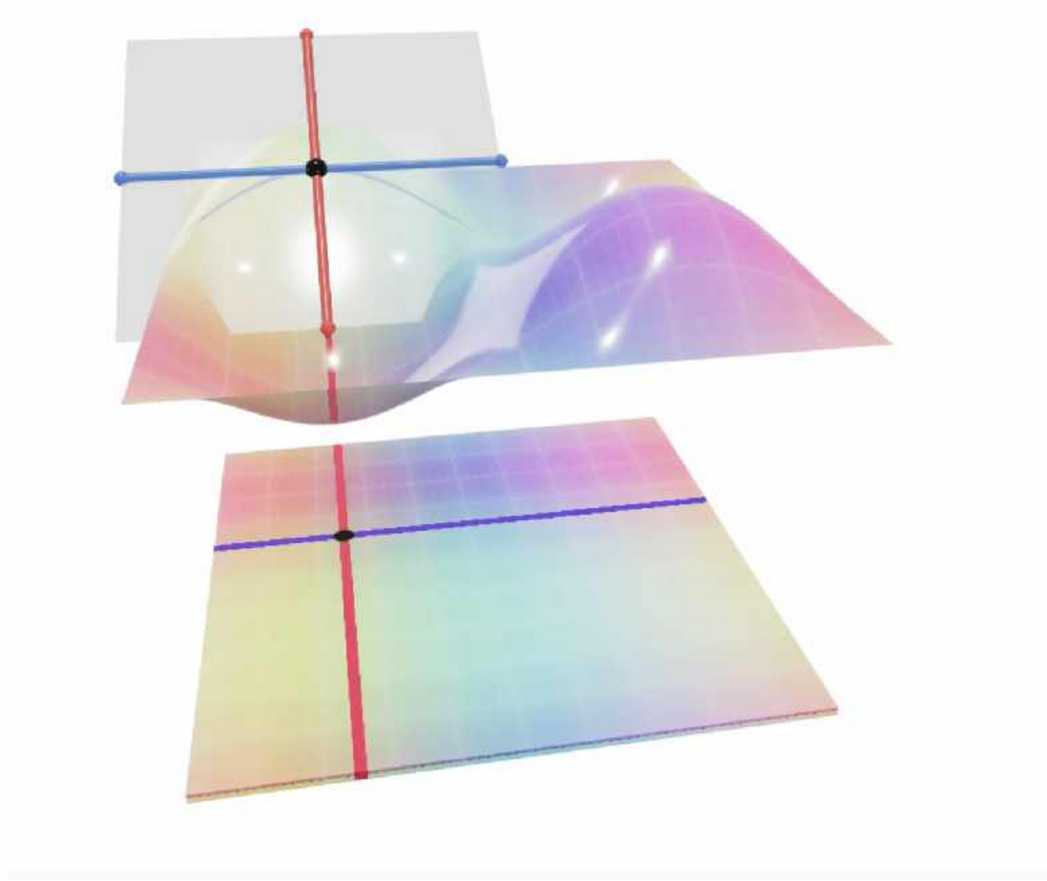




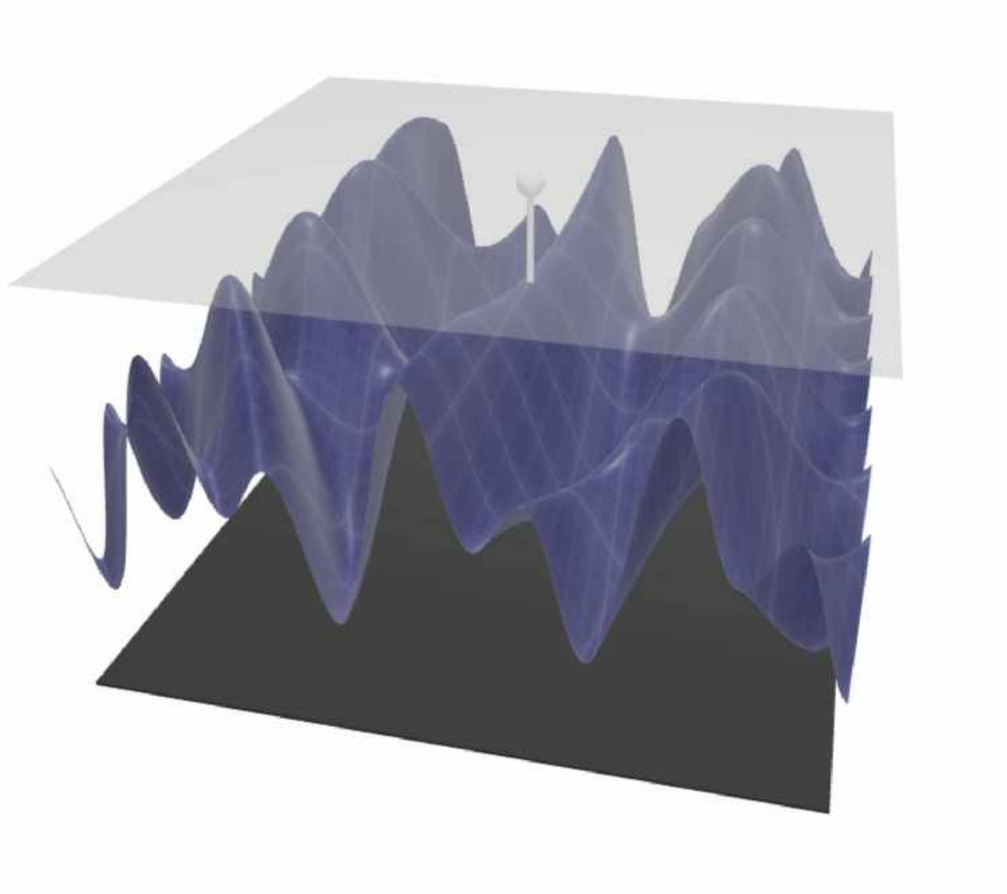
Positive Curvature

Negative Curvature

Applications to Mathematics:



Differential geometry: careful thinking about vector calculus and distances.



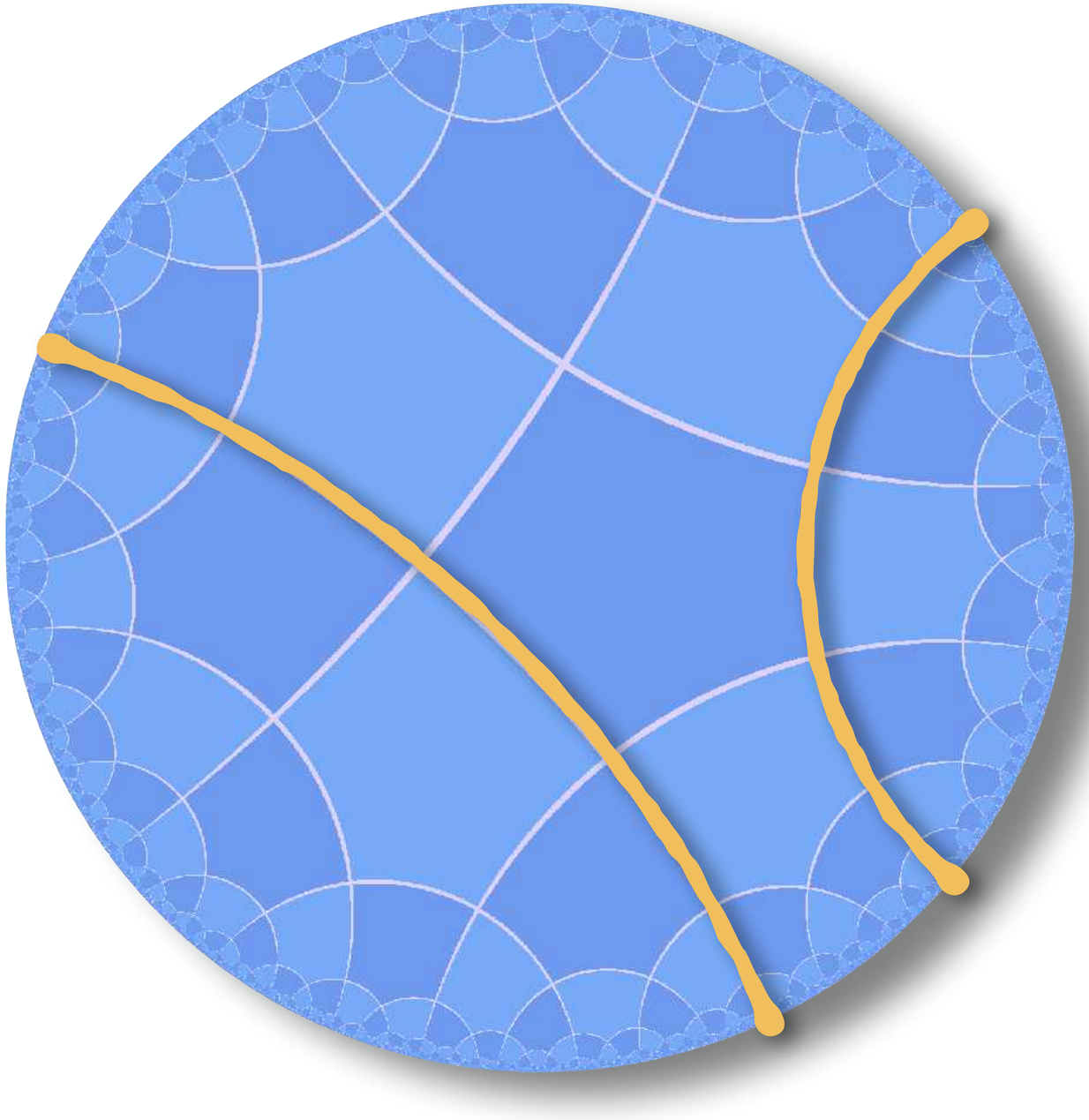
Differential topology: understanding spaces / graphs / functions by slicing.

Applications to Mathematics:

Taking curvature seriously led to the discovery of *Hyperbolic Geometry*


Bolyai Лобачевский Gauß

Proved that Euclid's 5th postulate is independent of the first 4, which was open for nearly 2,000 years.

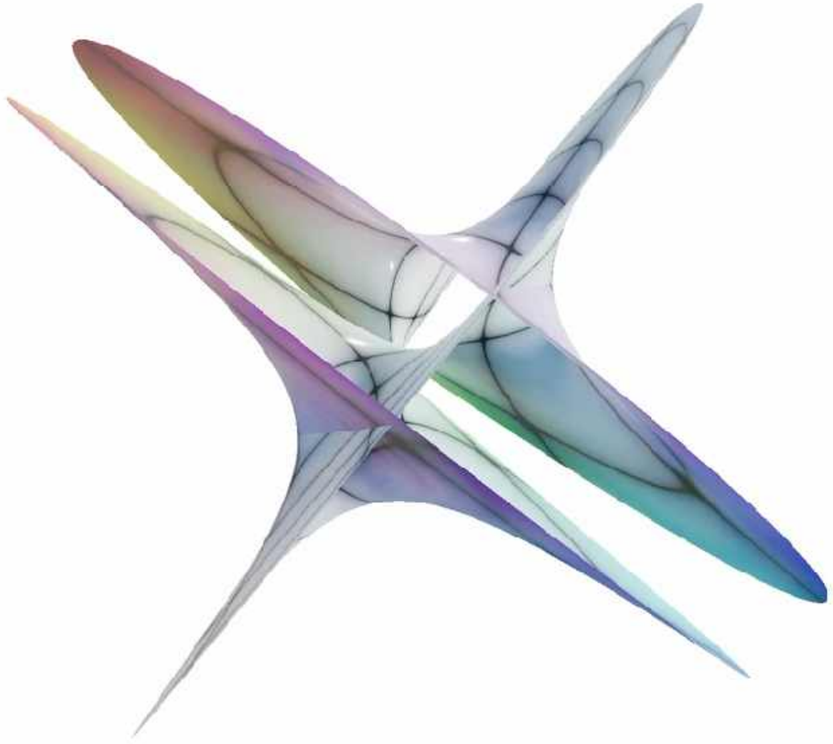


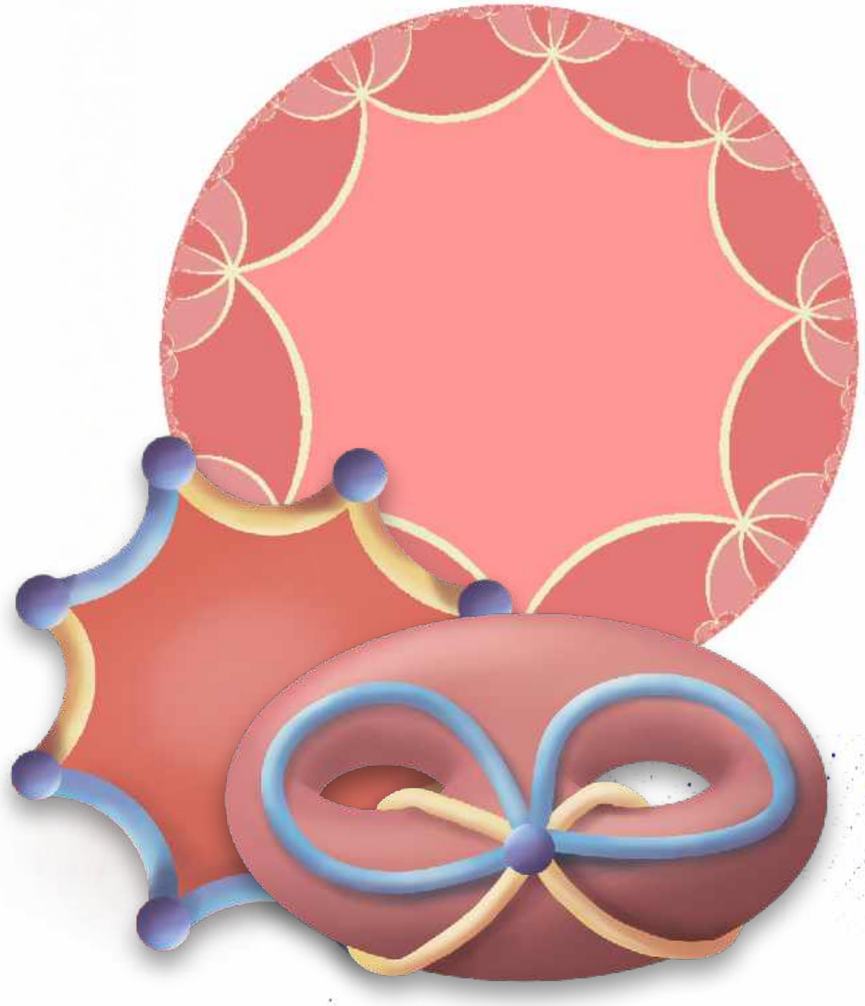
Hyperbolic geometry is everywhere!

Complex Analysis

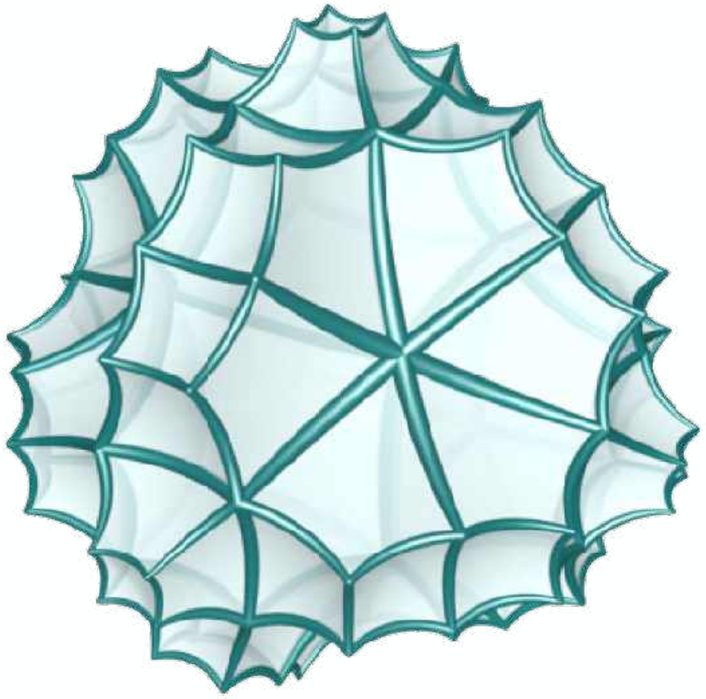
Topology

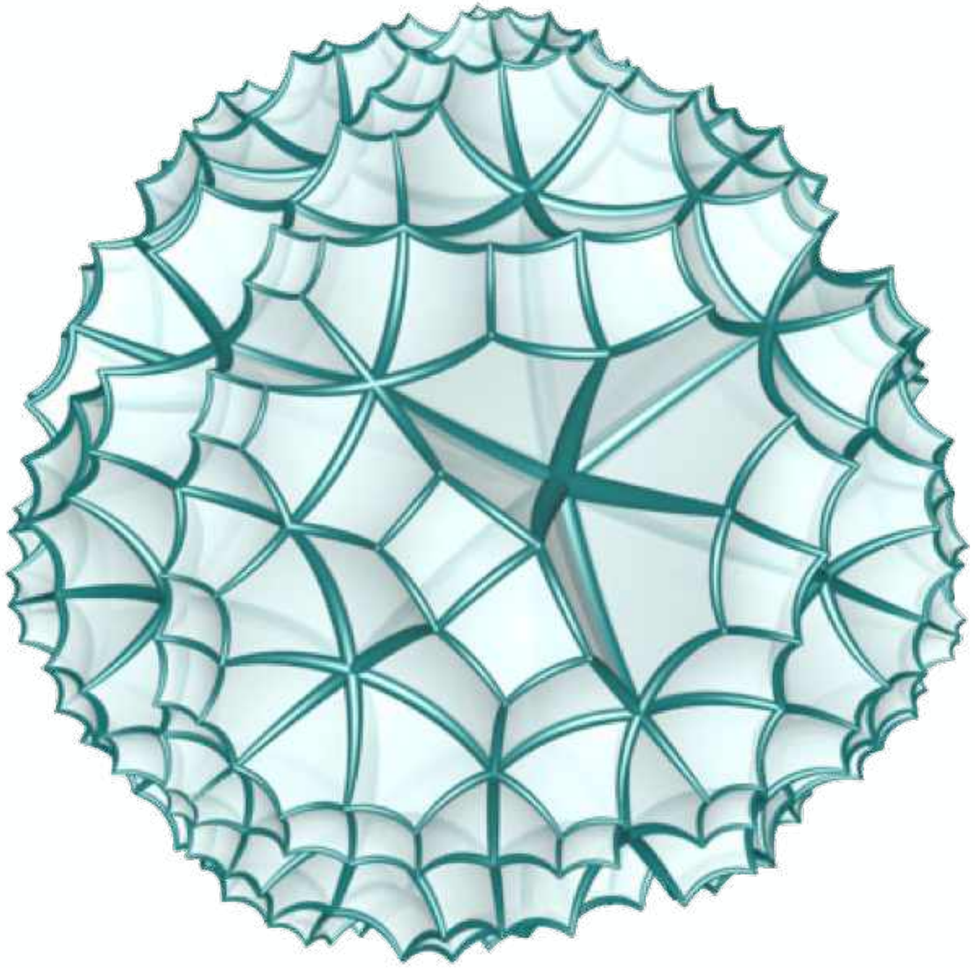
Number Theory





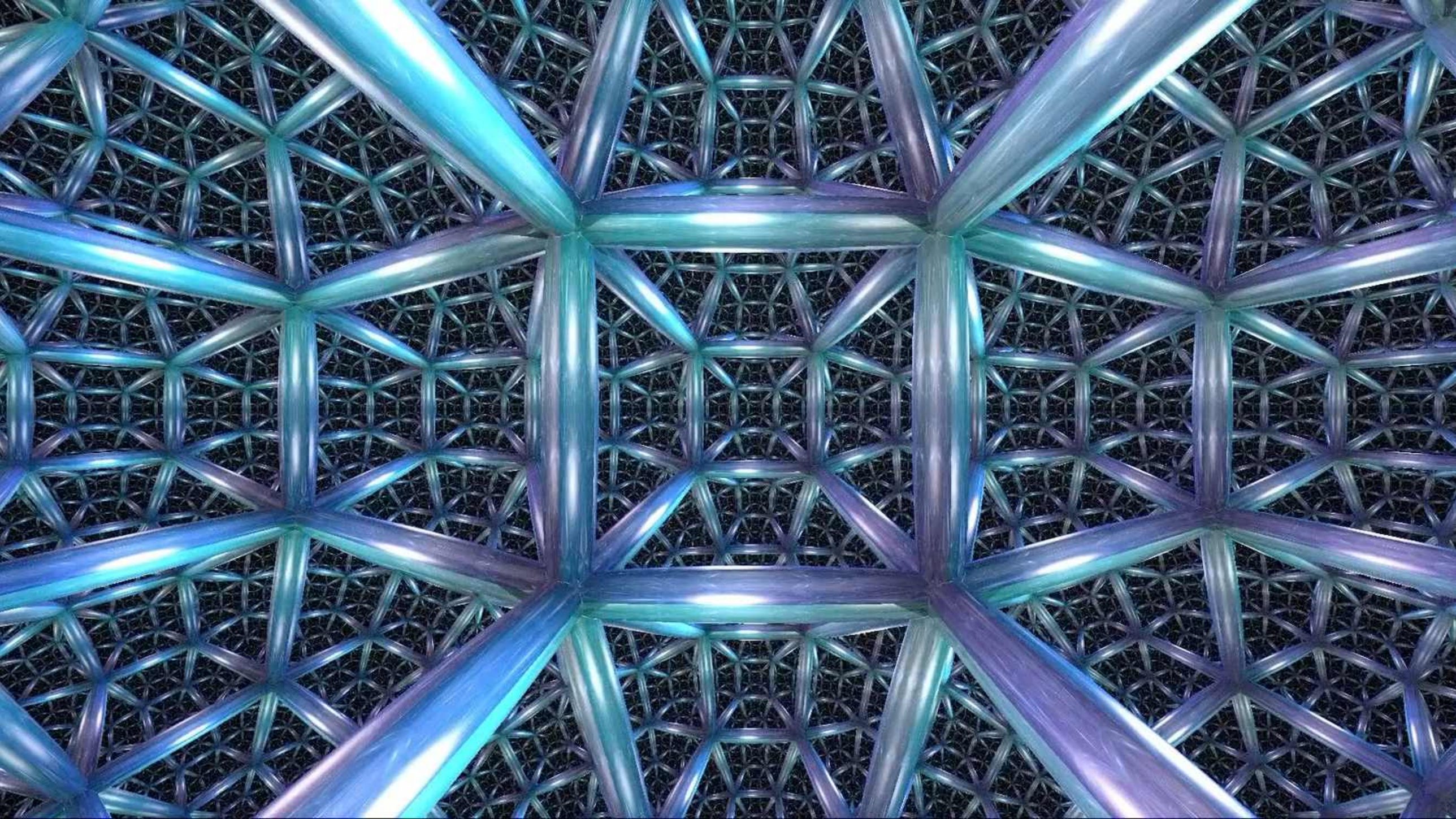
Negatively curved space is HUGE. Most of the interesting questions about 3-dimensional space turn out to be questions about hyperbolic geometry.





Exponentially many cubes stack around a central cube.

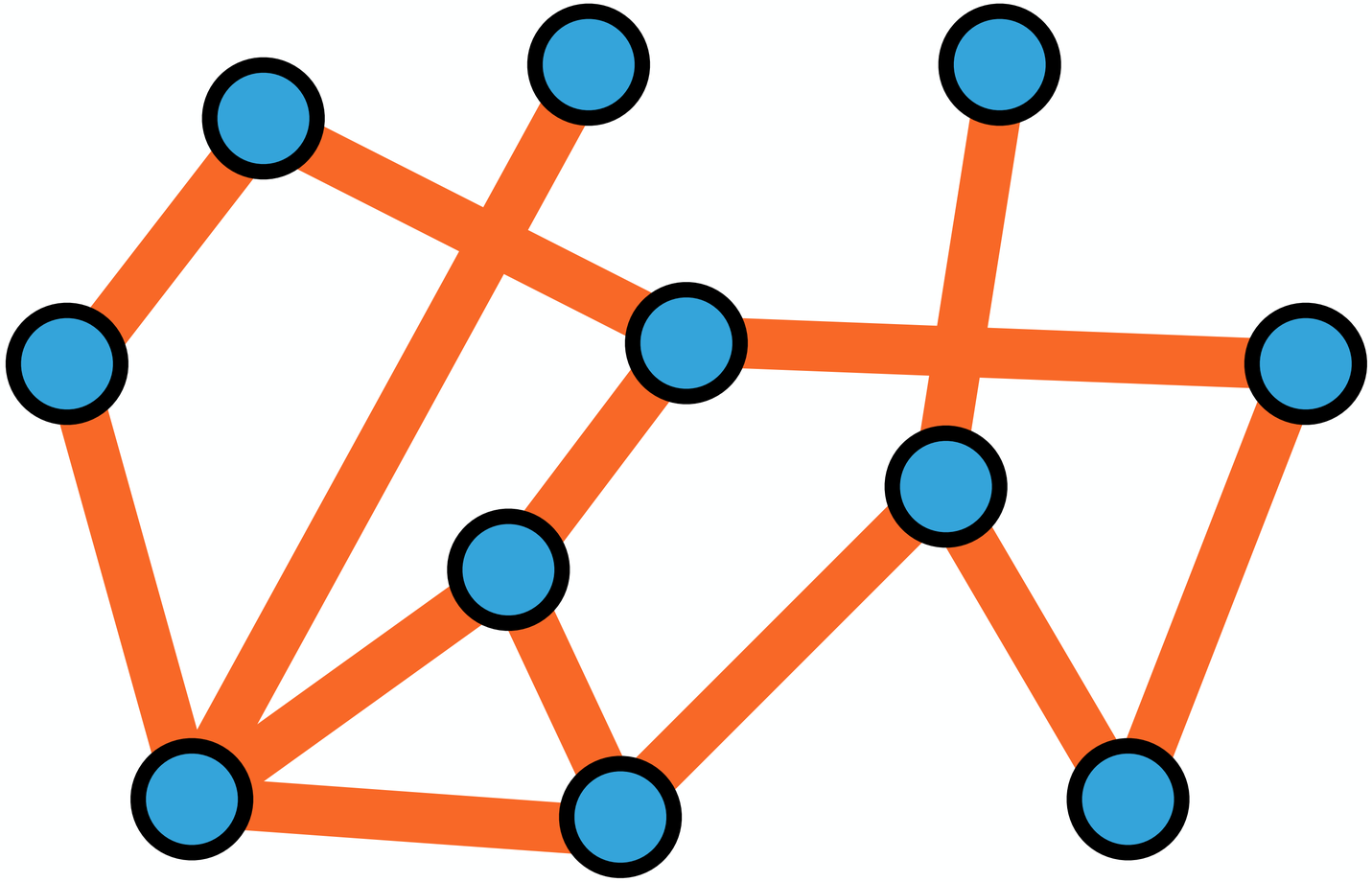
Image by Roice Nelson



Untangling Webs

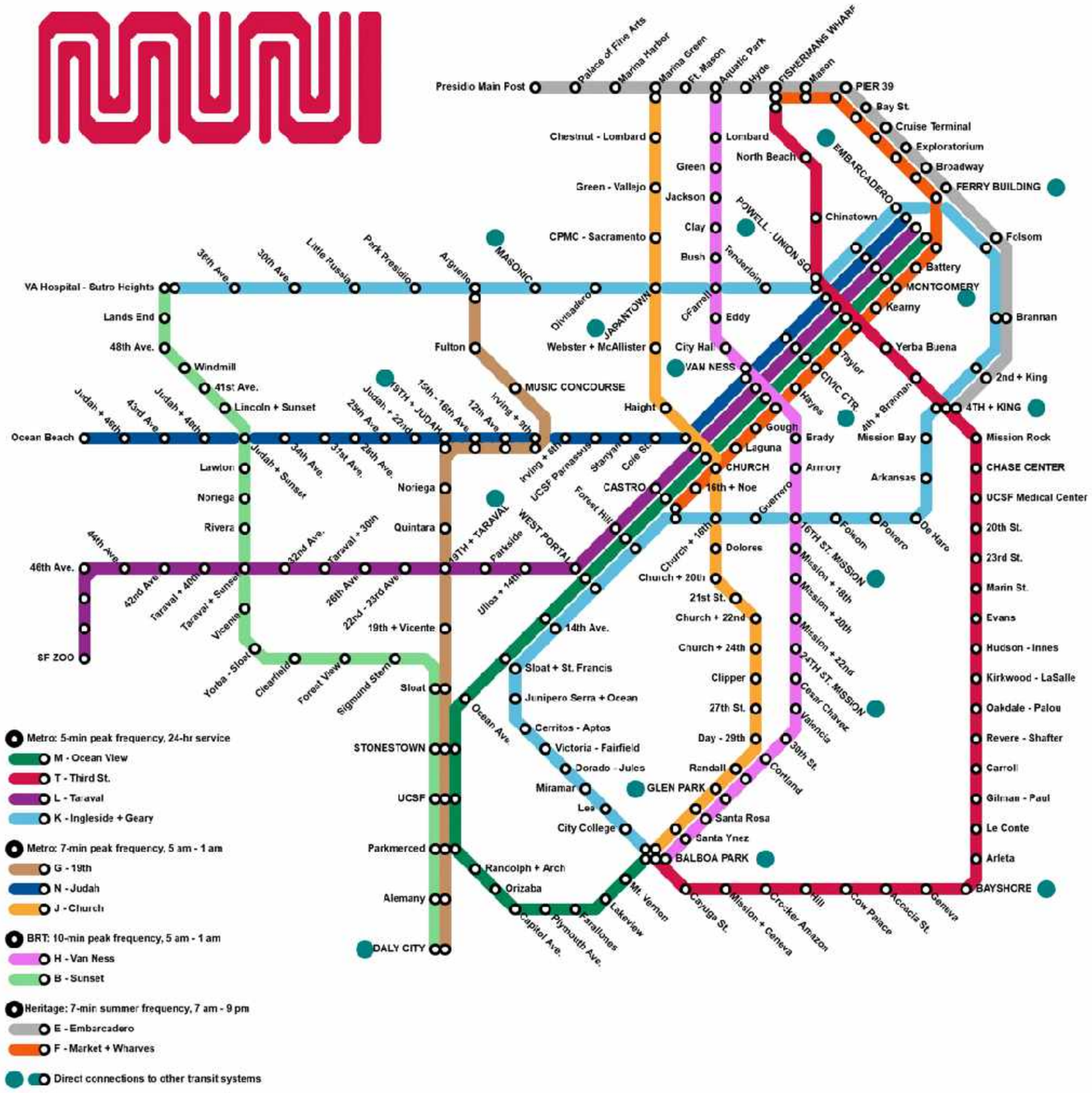
The geometry of graph embeddings.

Data and Graphs



Vertices are data

Edges are relations



The Muni Graph

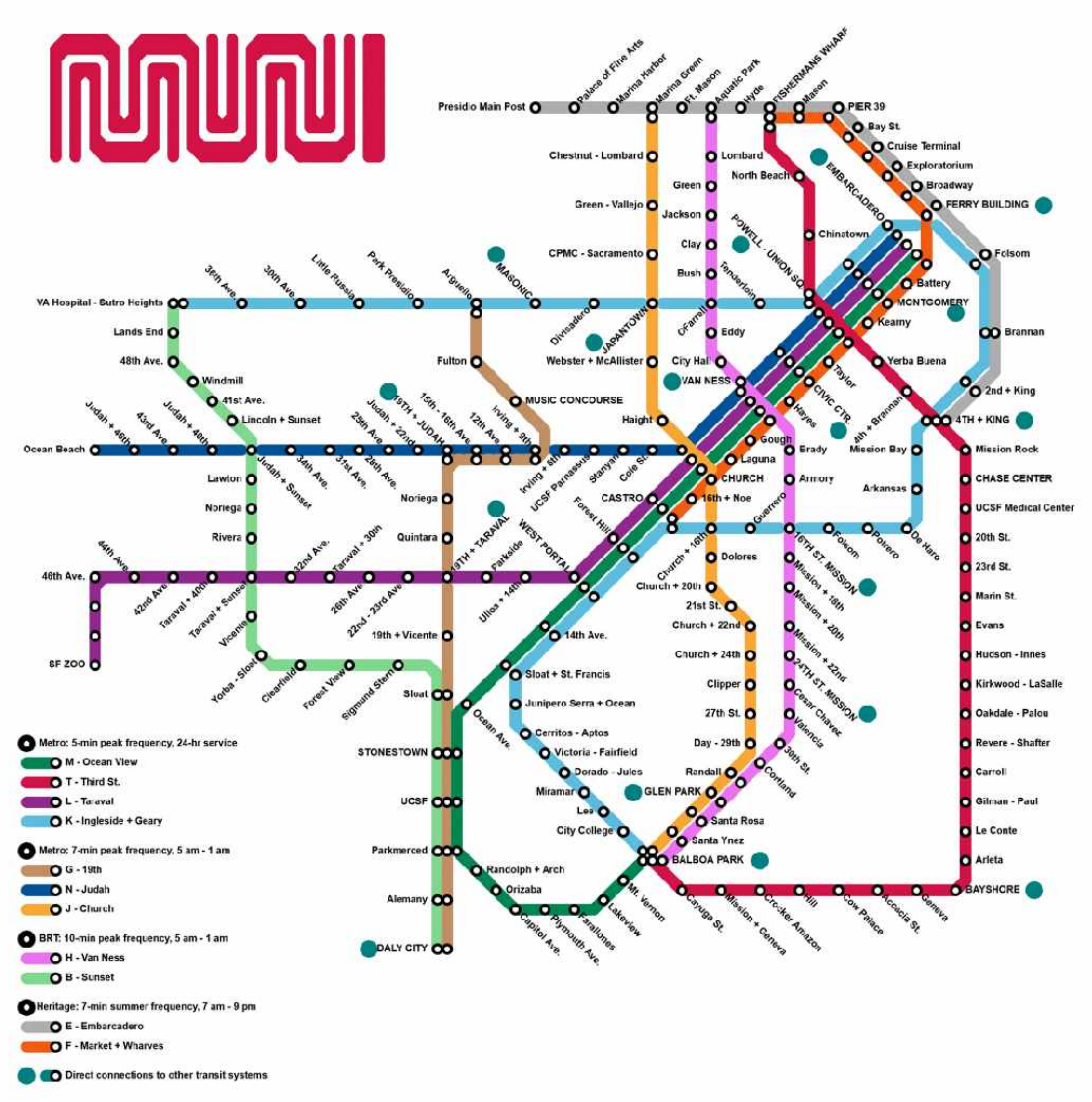
Vertices: Bus/Tram Stops

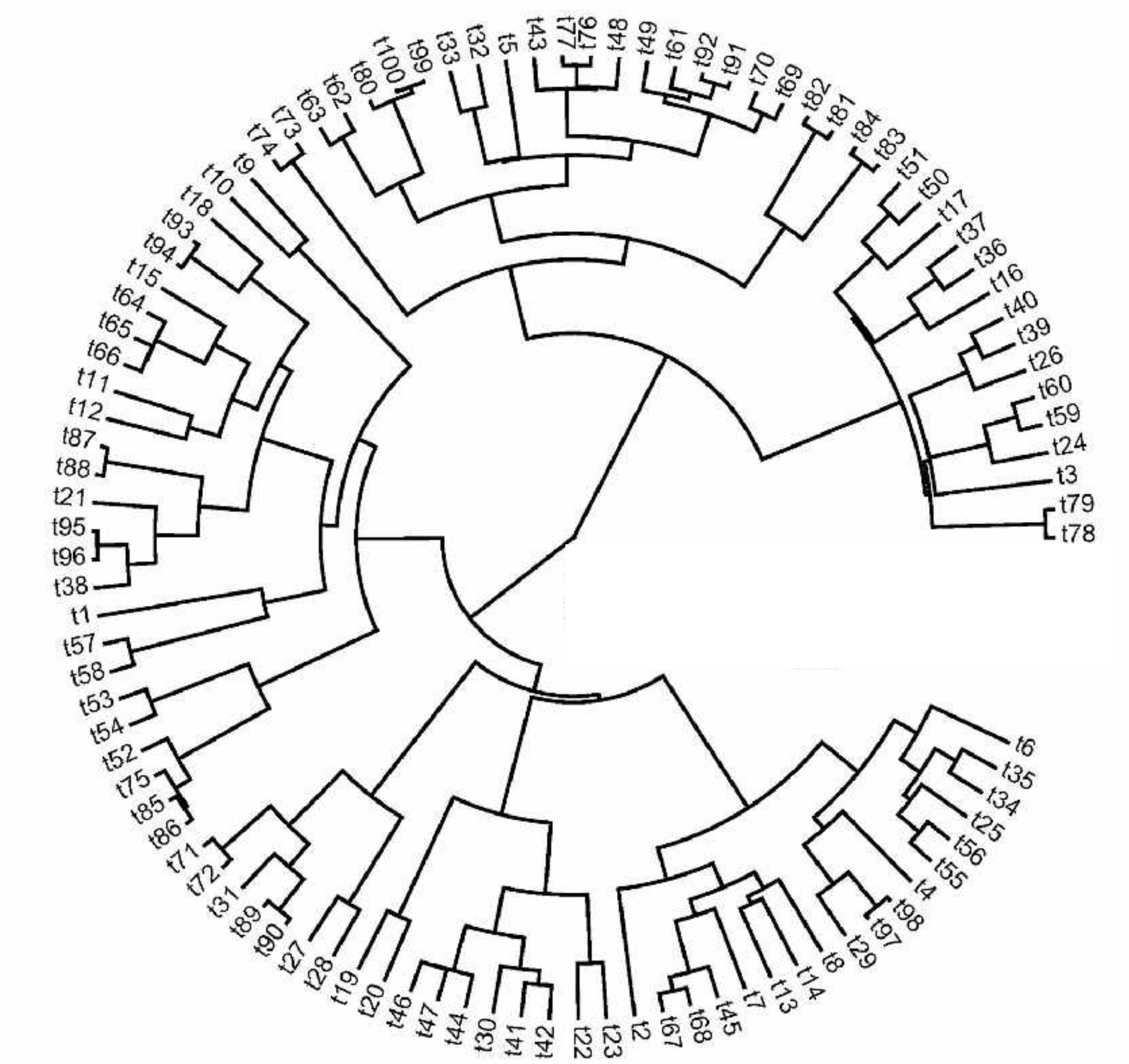
Edges: Bus/Tram Routes

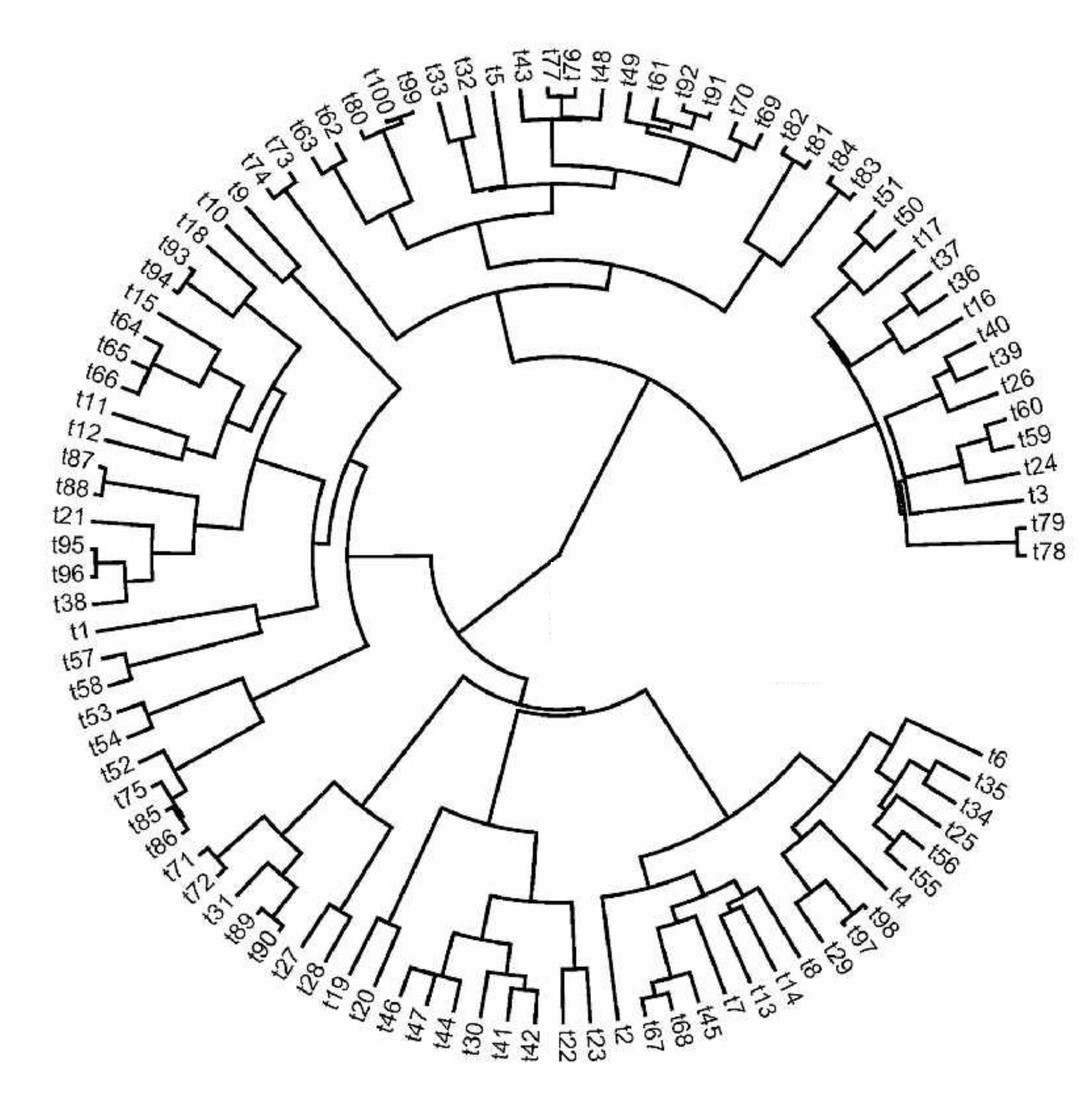
Phylogenetic Trees

Vertices: Species or Genes

Edges: Ancestry



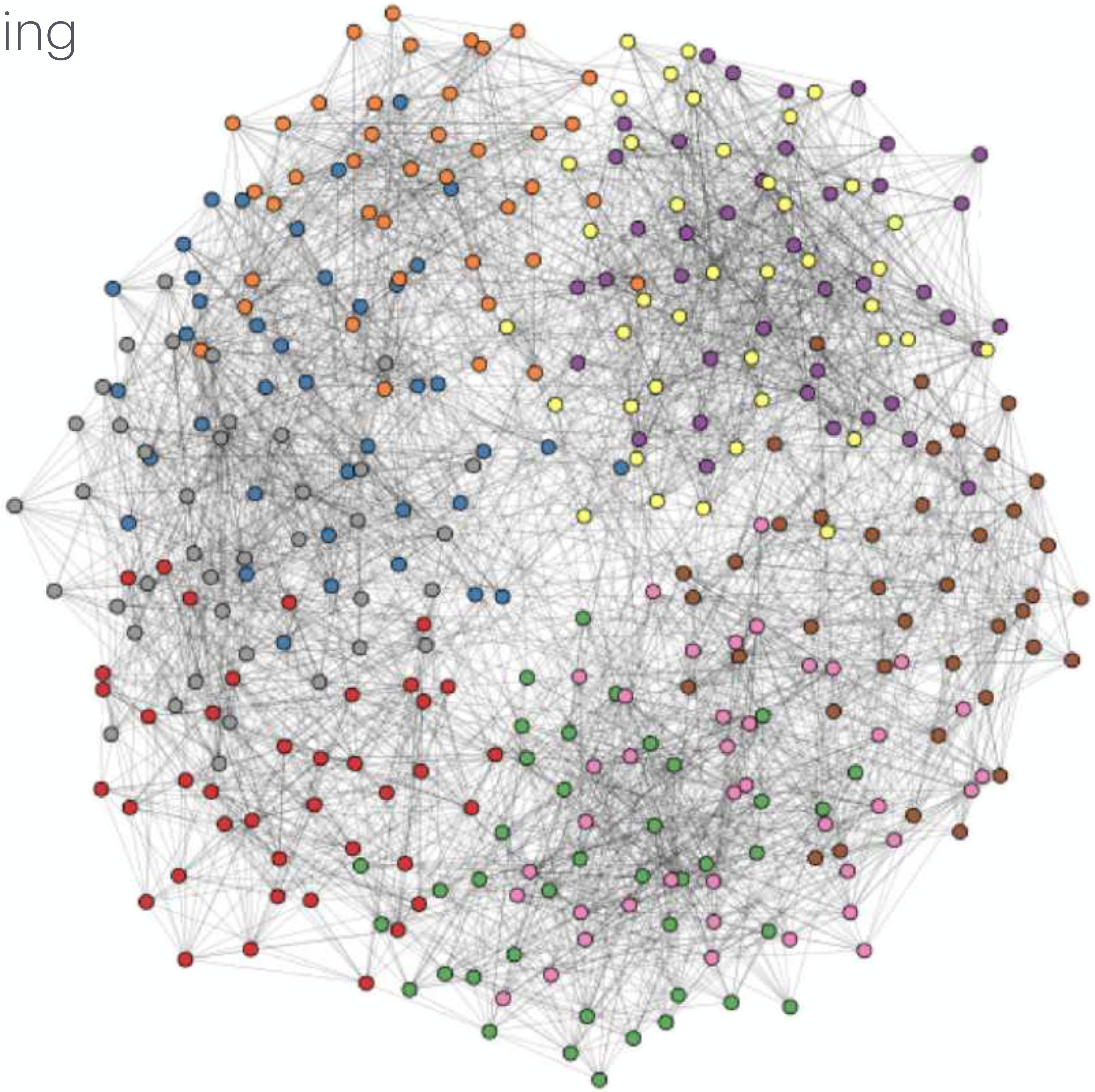


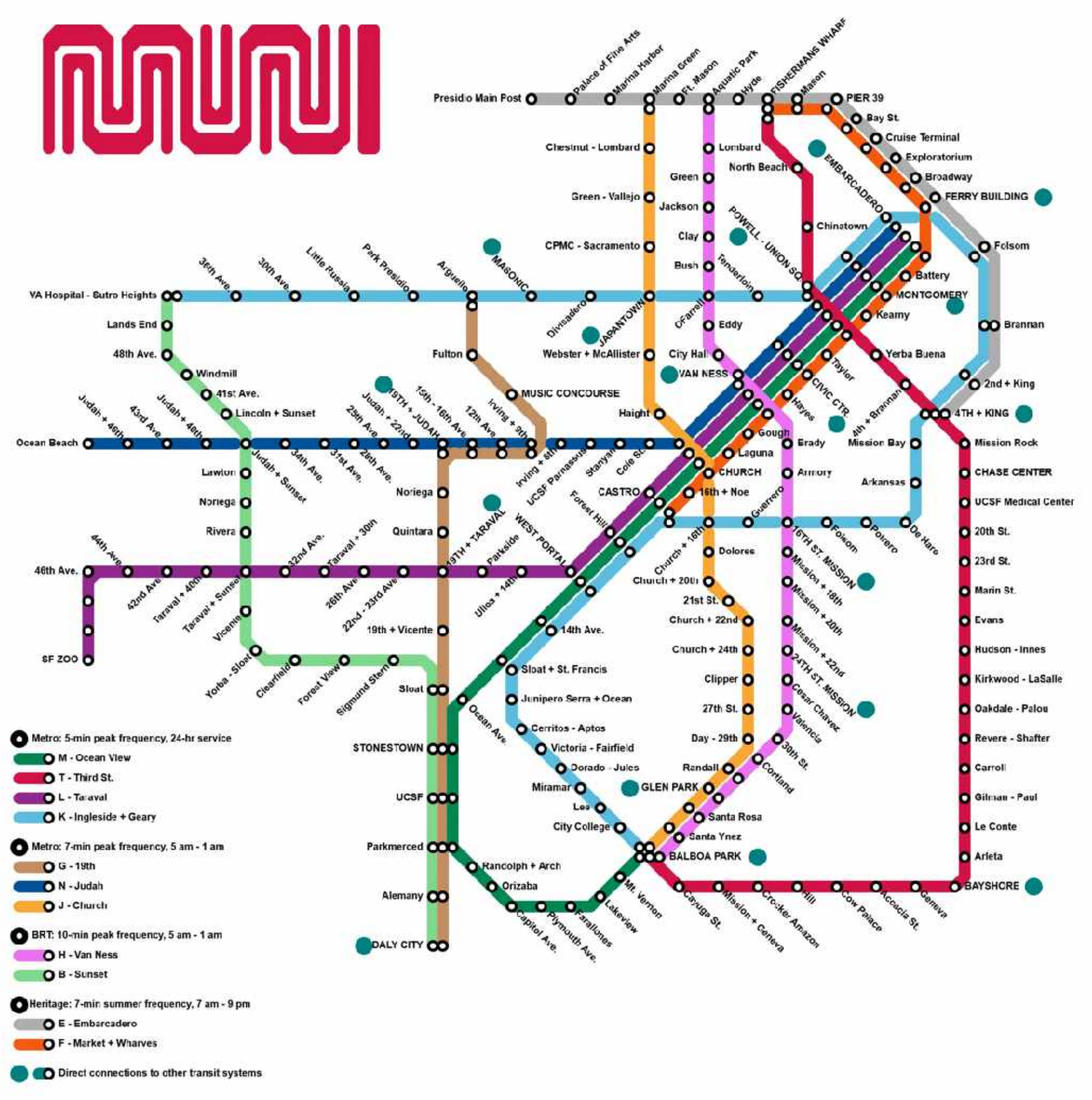


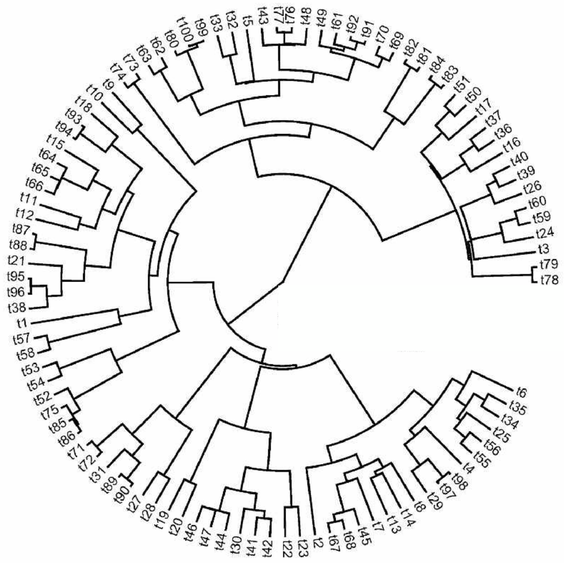
Social Networks

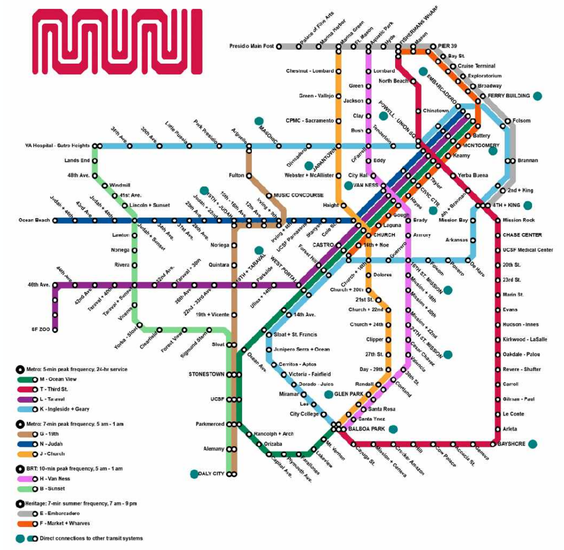
Vertices: Users

Edges: Following





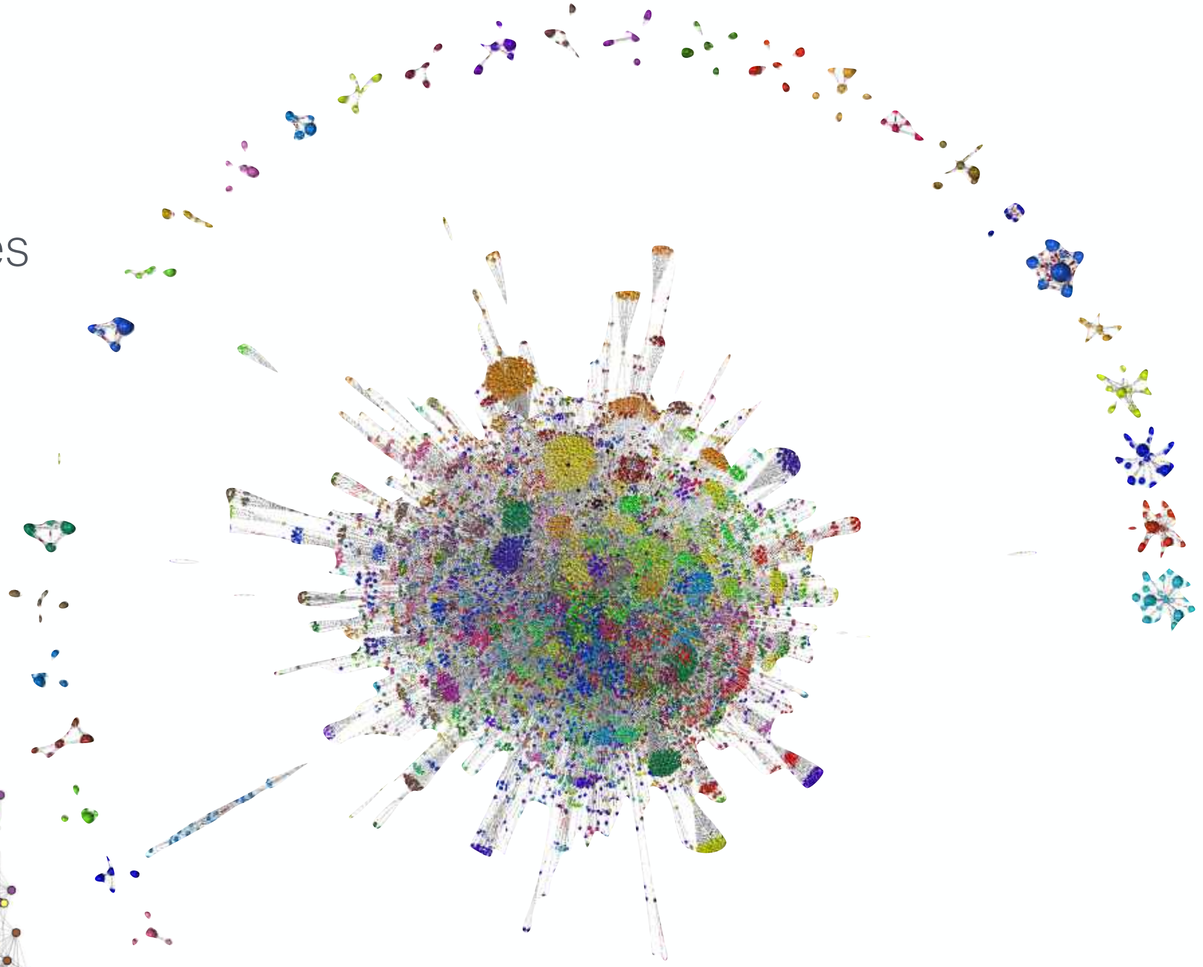


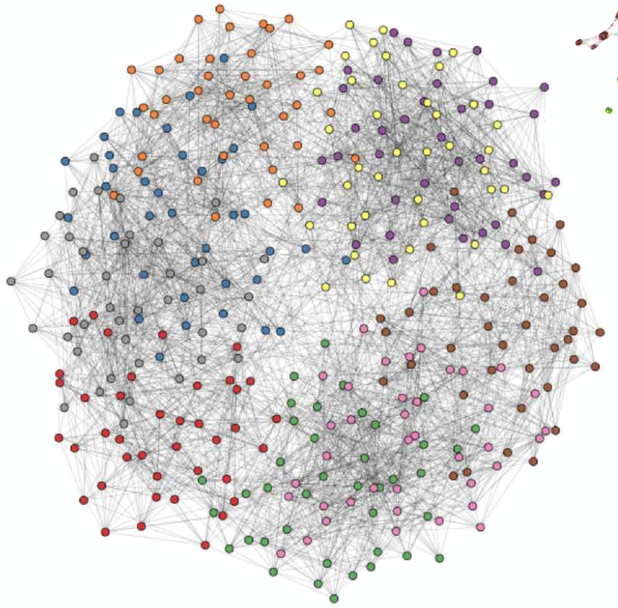


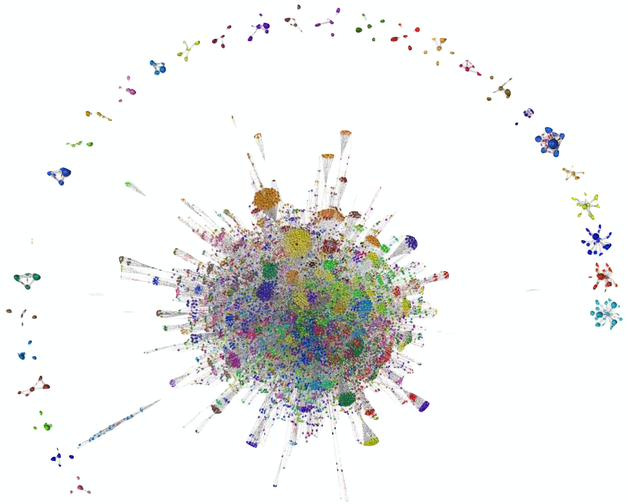
Reddit

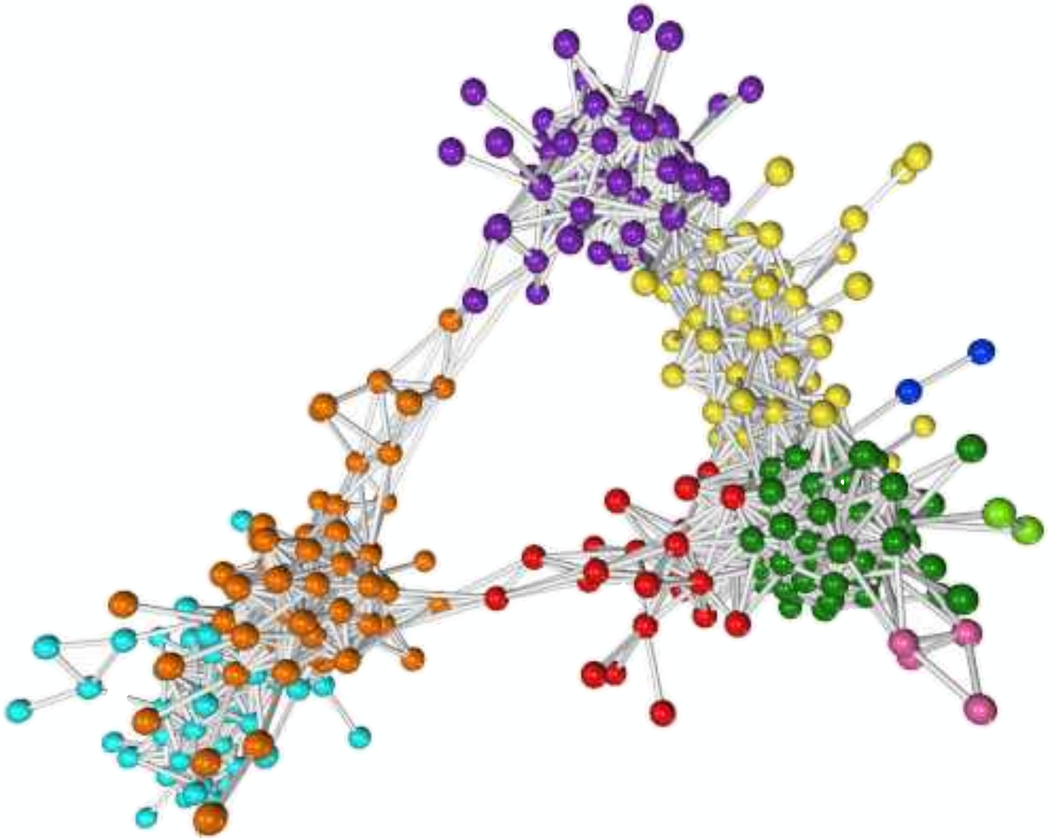
Vertices: Subreddits

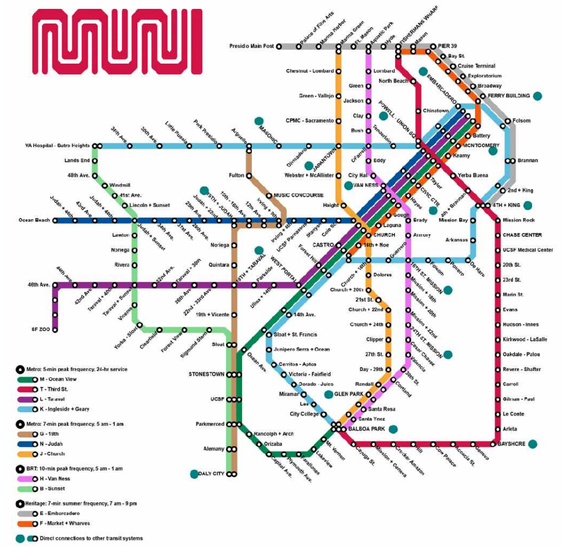
Edges: Crossreferences

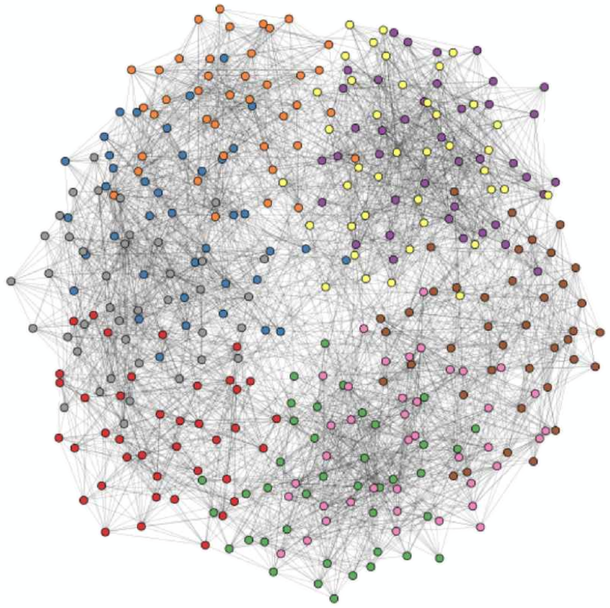








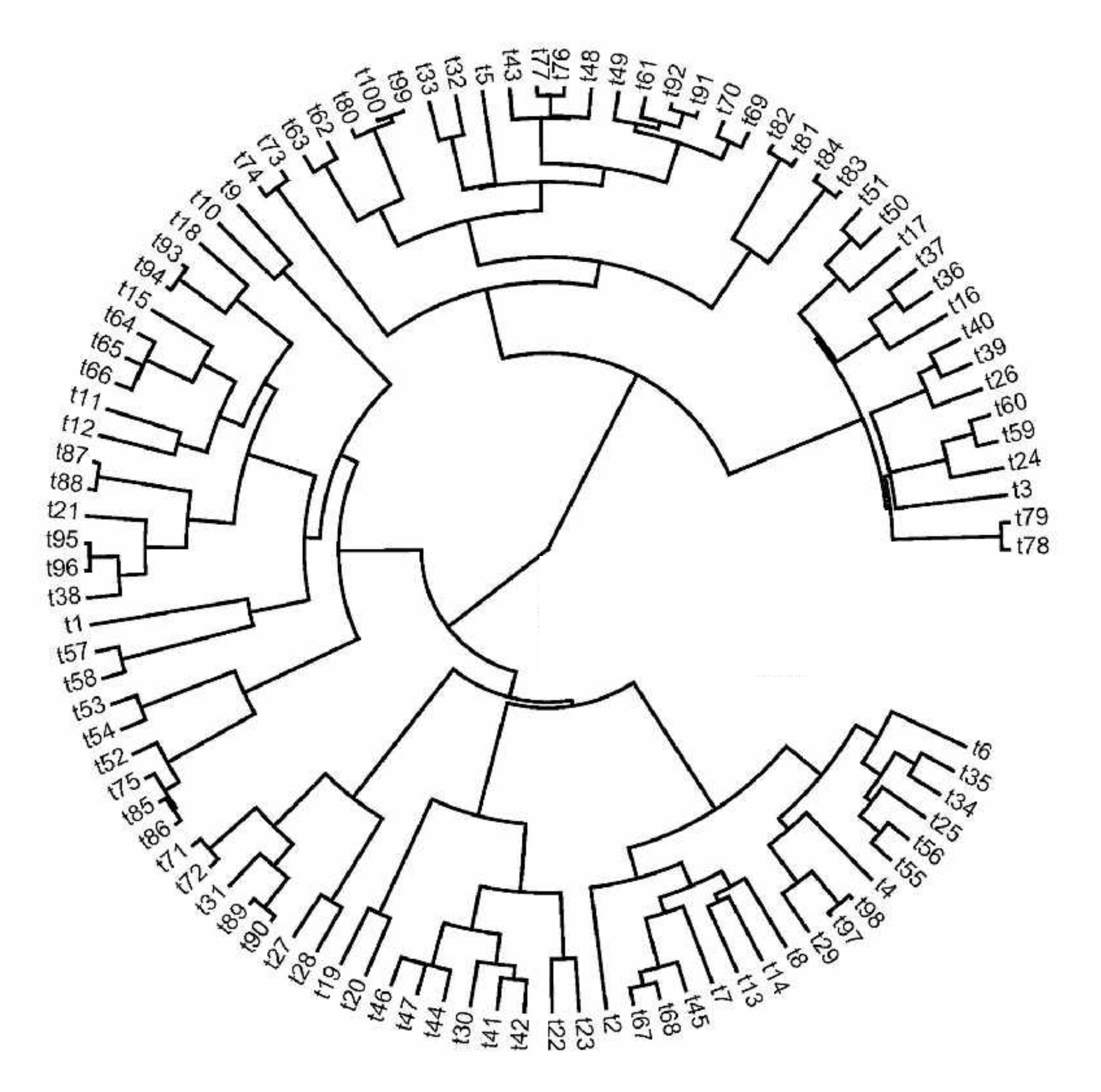


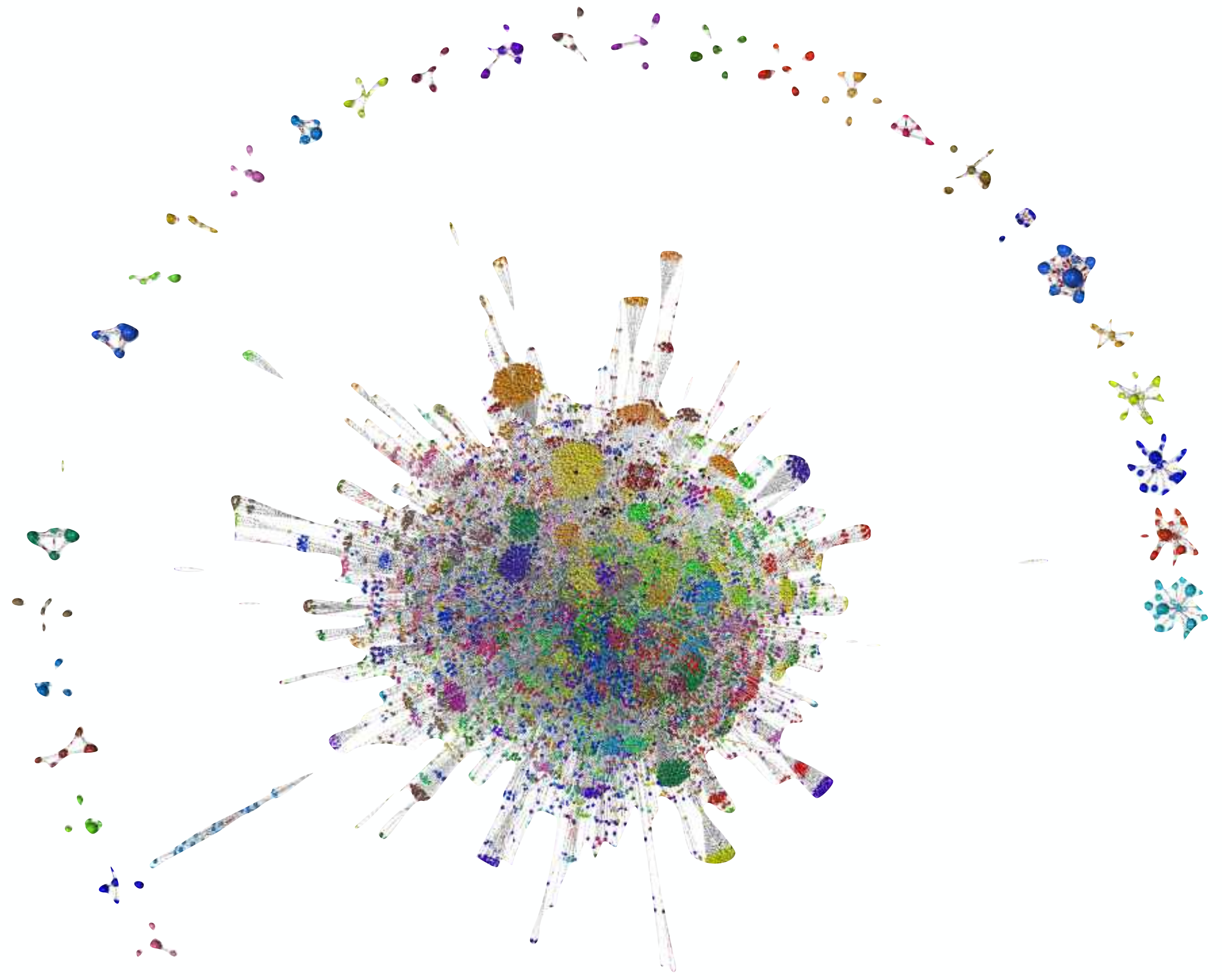


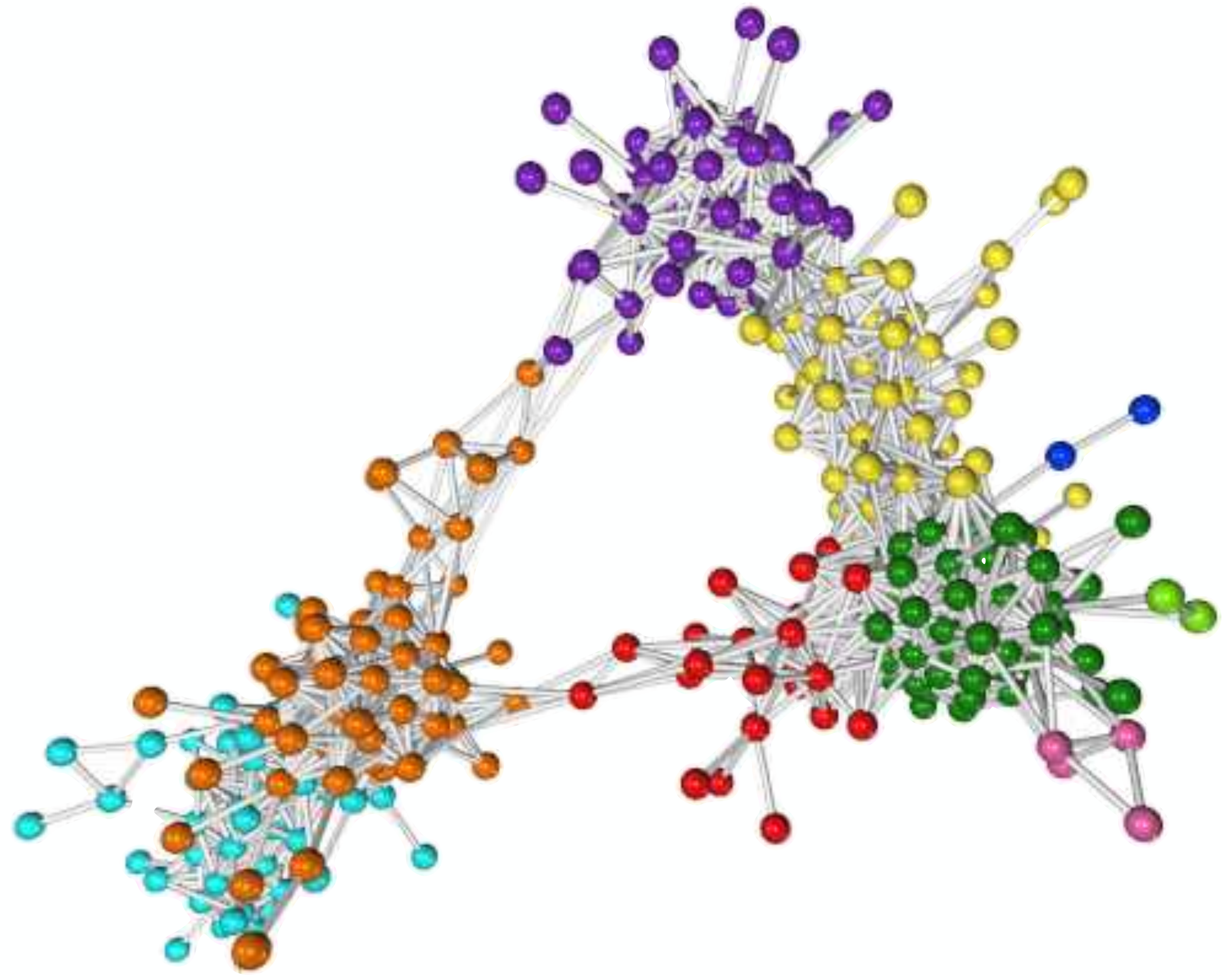
Biochemistry

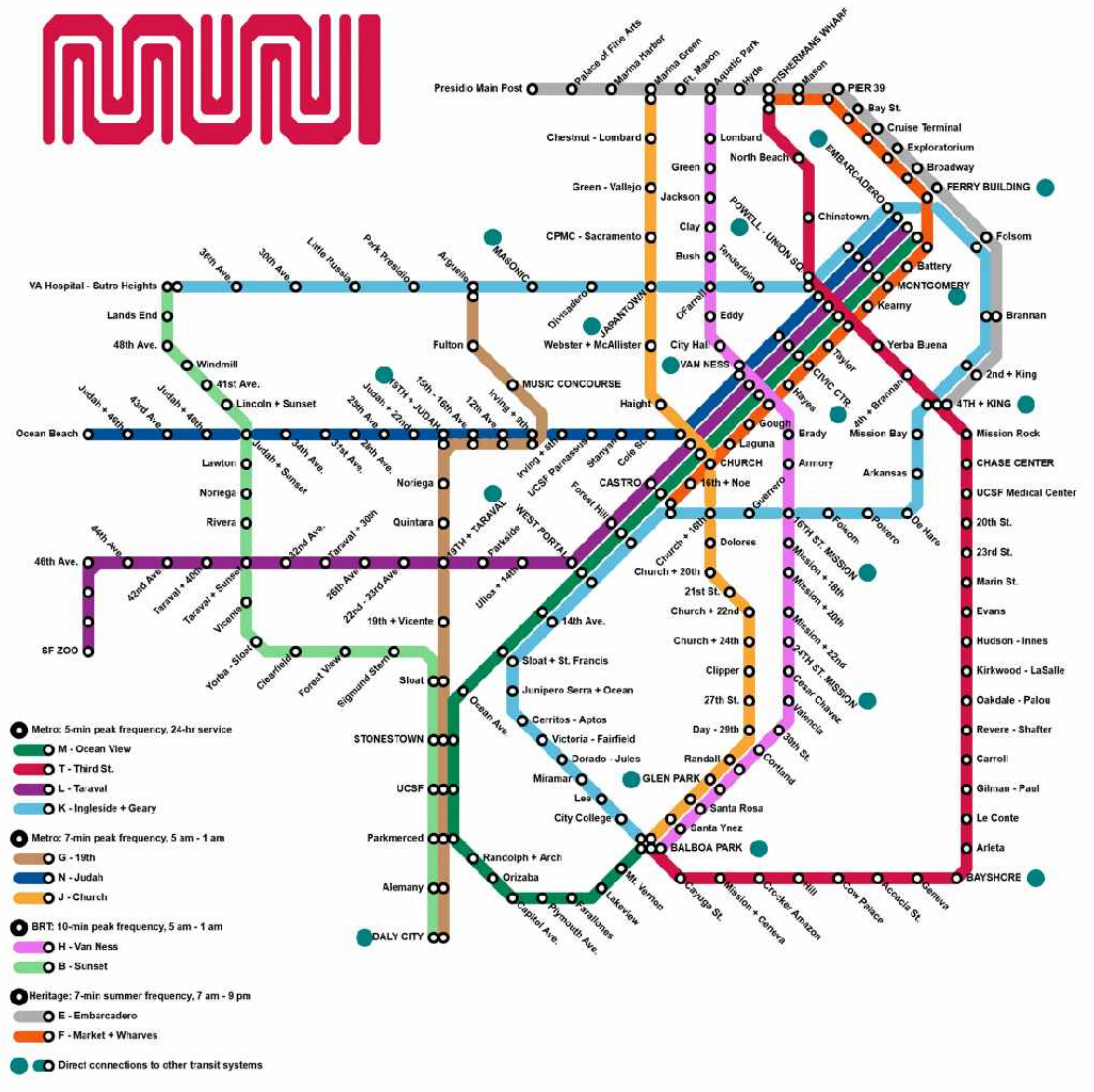
Vertices: Mouse Genes

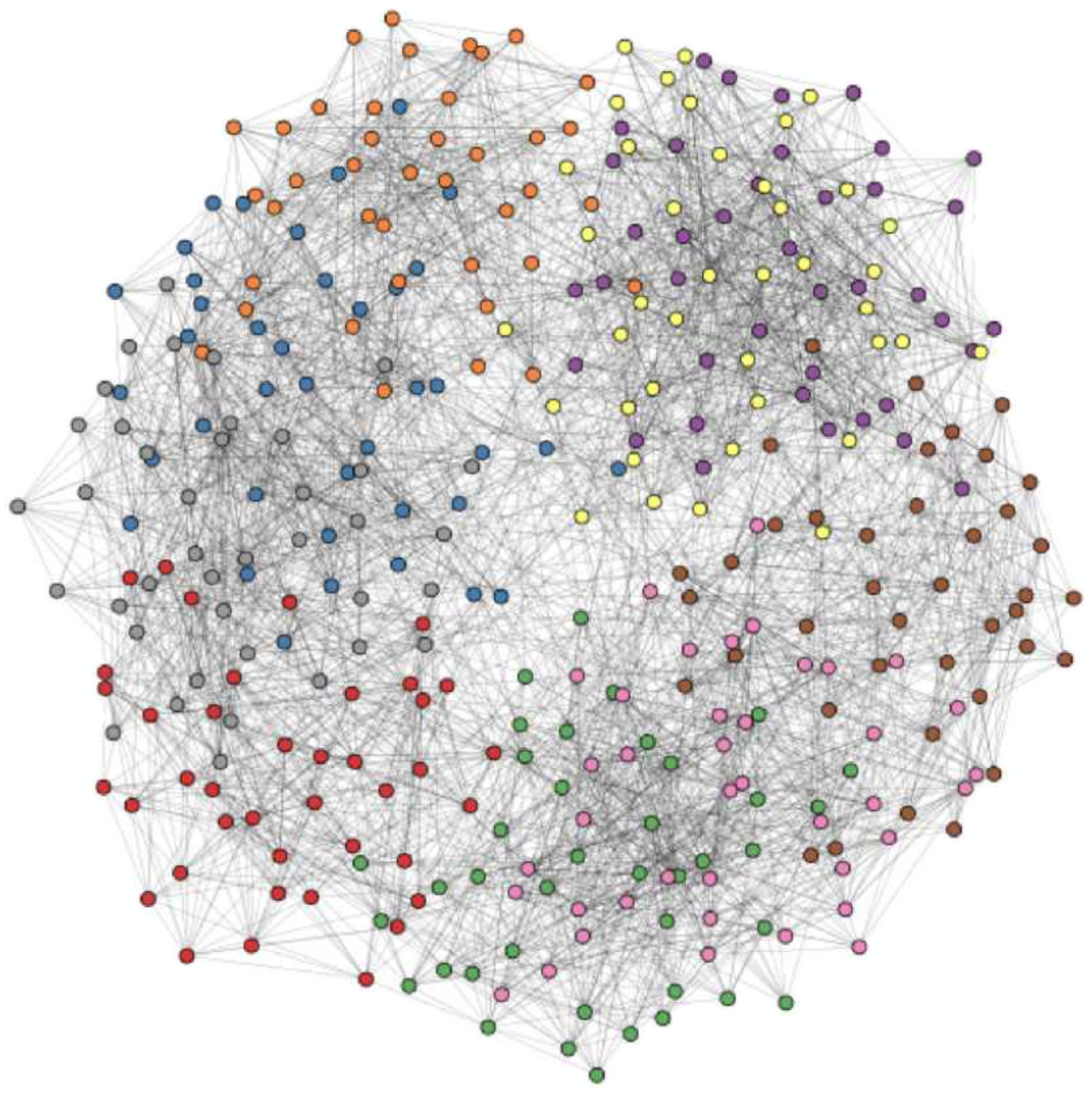
Edges: Co-expression

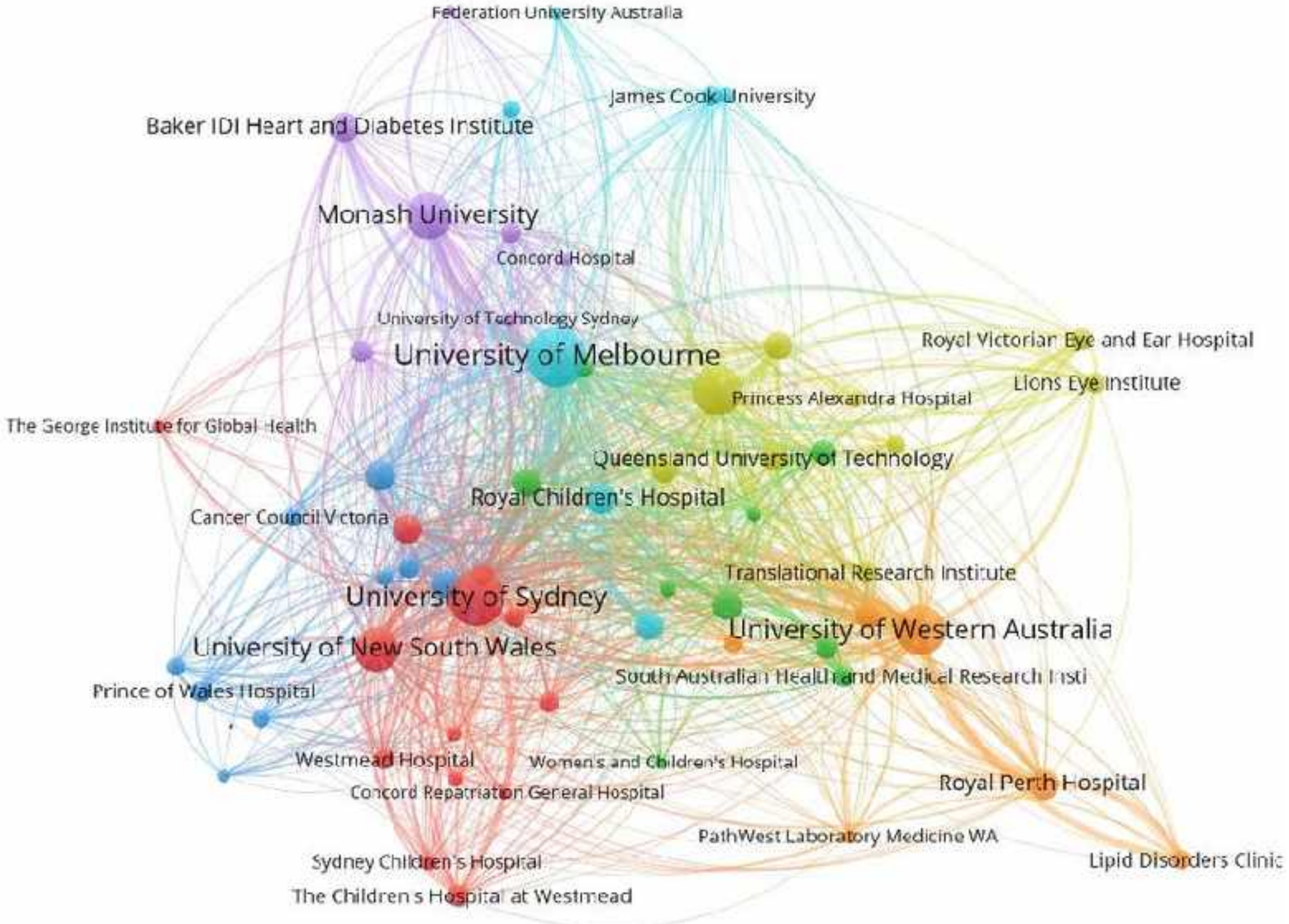


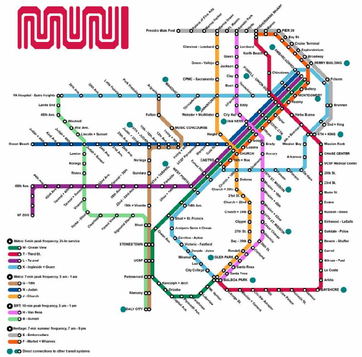






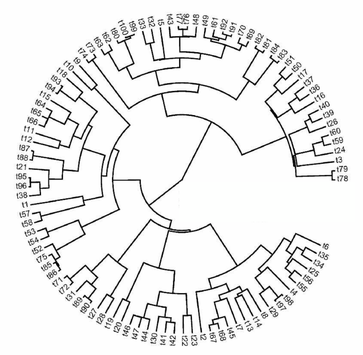



Federation University Australia
James Cook University
Baker IDI Heart and Diabetes Institute
Monash University
Concord Hospital
University of Technology Sydney
University of Melbourne
Royal Victorian Eye and Ear Hospital
Lions Eye Institute
Princess Alexandra Hospital
The George Institute for Global Health
Queensland University of Technology
Royal Children's Hospital
Cancer Council Victoria
Translational Research Institute
University of Sydney
University of Western Australia
University of New South Wales
South Australian Health and Medical Research Insti
Prince of Wales Hospital
Westmead Hospital
Women's and Children's Hospital
Concord Repatriation General Hospital
Royal Perth Hospital
PathWest Laboratory Medicine WA
Sydney Children's Hospital
Lipid Disorders Clinic
The Children's Hospital at Westmead



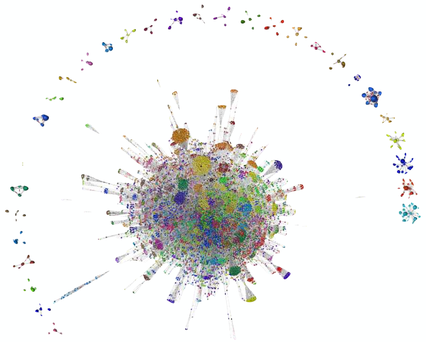
Route “Length” :

Travel Time



Genetic “Distance” :

Number of base pair changes



Community “Closeness” :

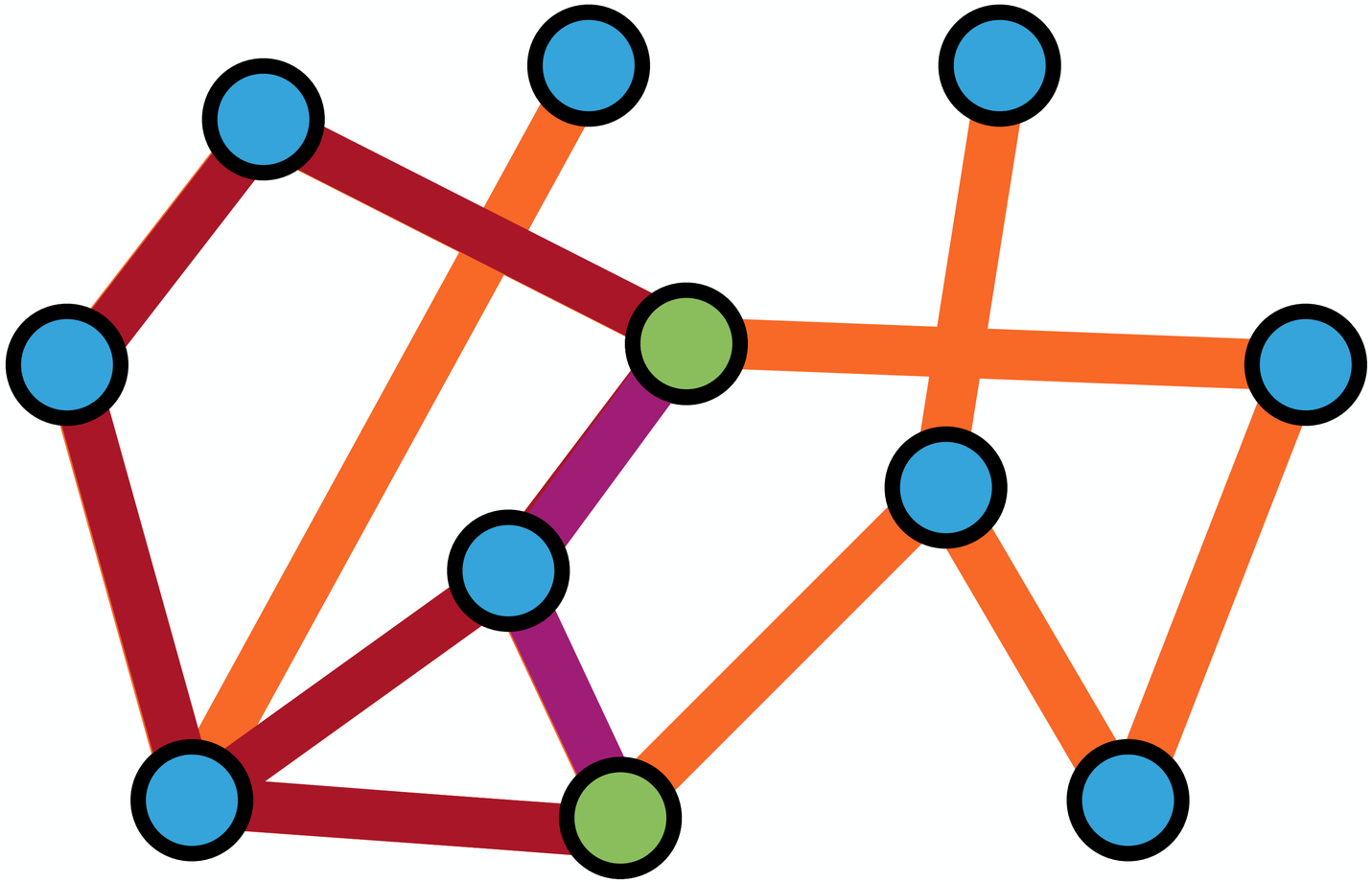
1/(Number Crossrefs)

Some connections are stronger than others.

If we have a ‘weight’ on each edge, we can treat it as a length. **This gives the graph geometry!**

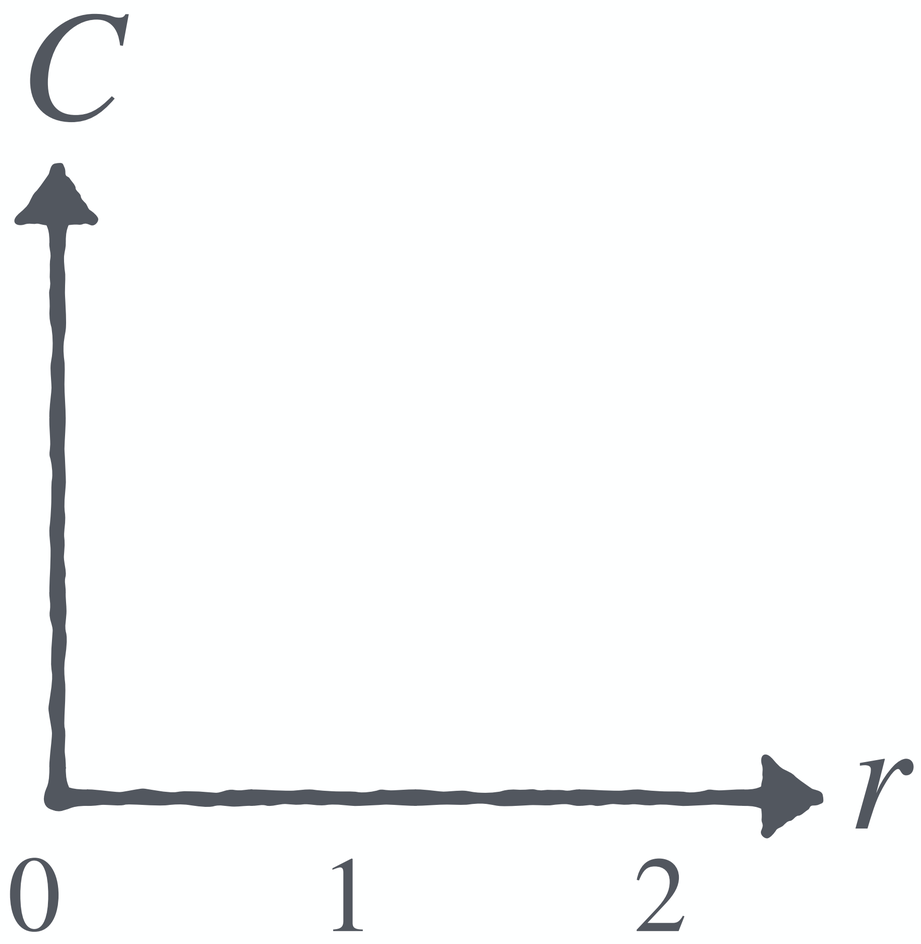
Distance gives Geometry to a Graph

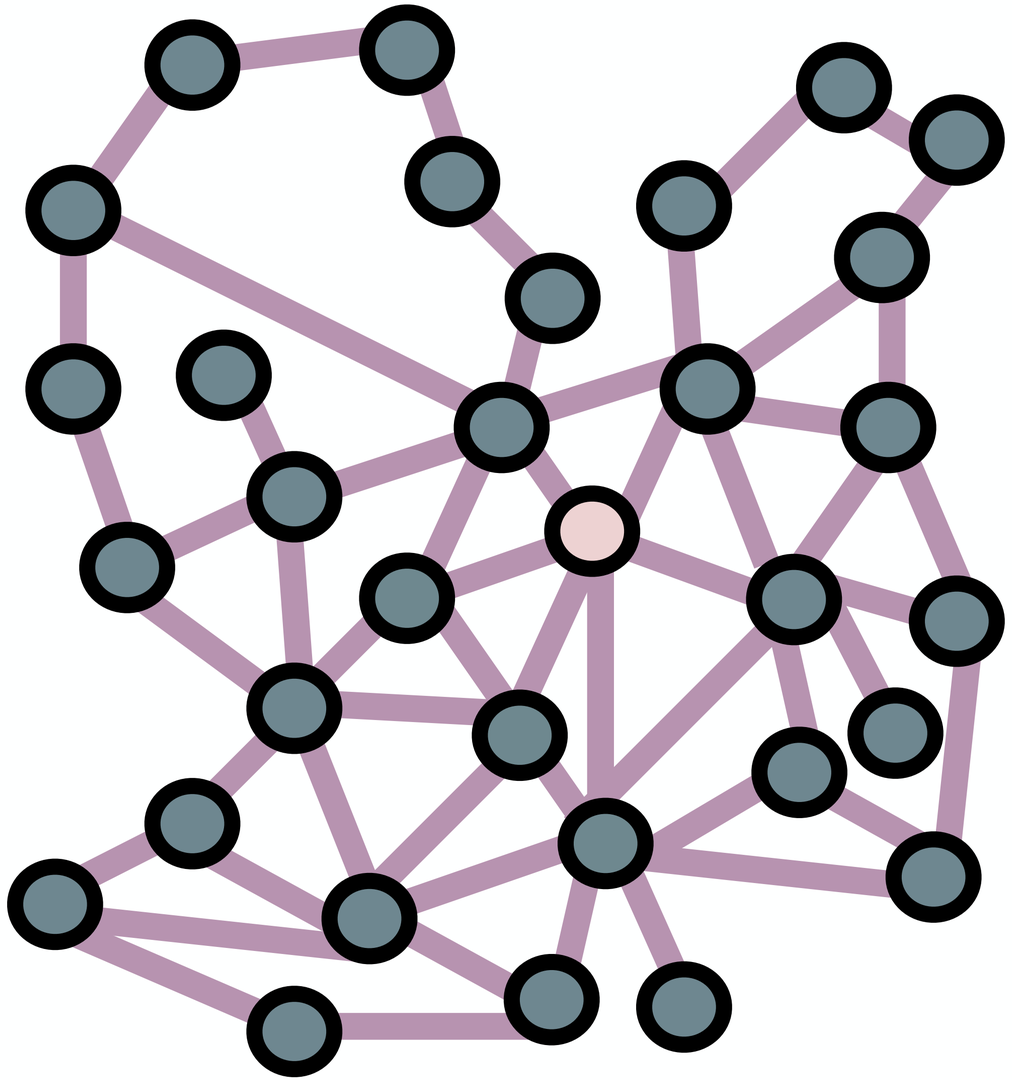
The distance between two vertices is the length of the shortest path.



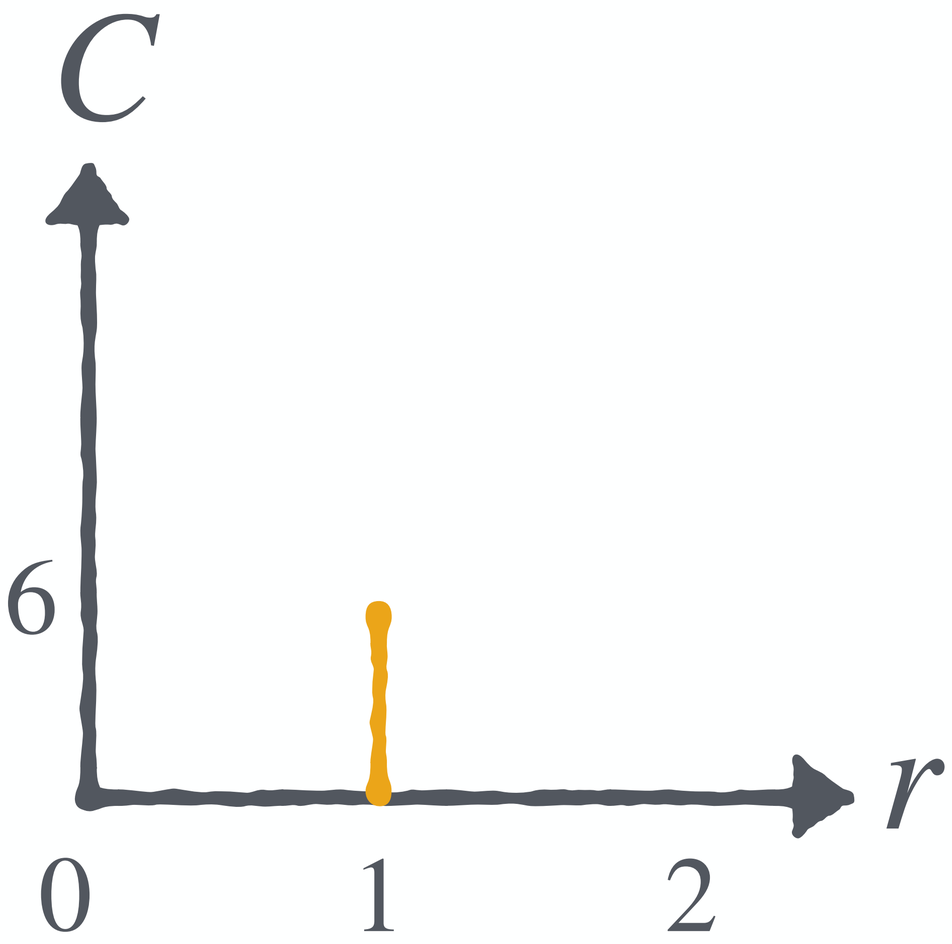
$$\ell = 4 = 2 = \text{dist}$$

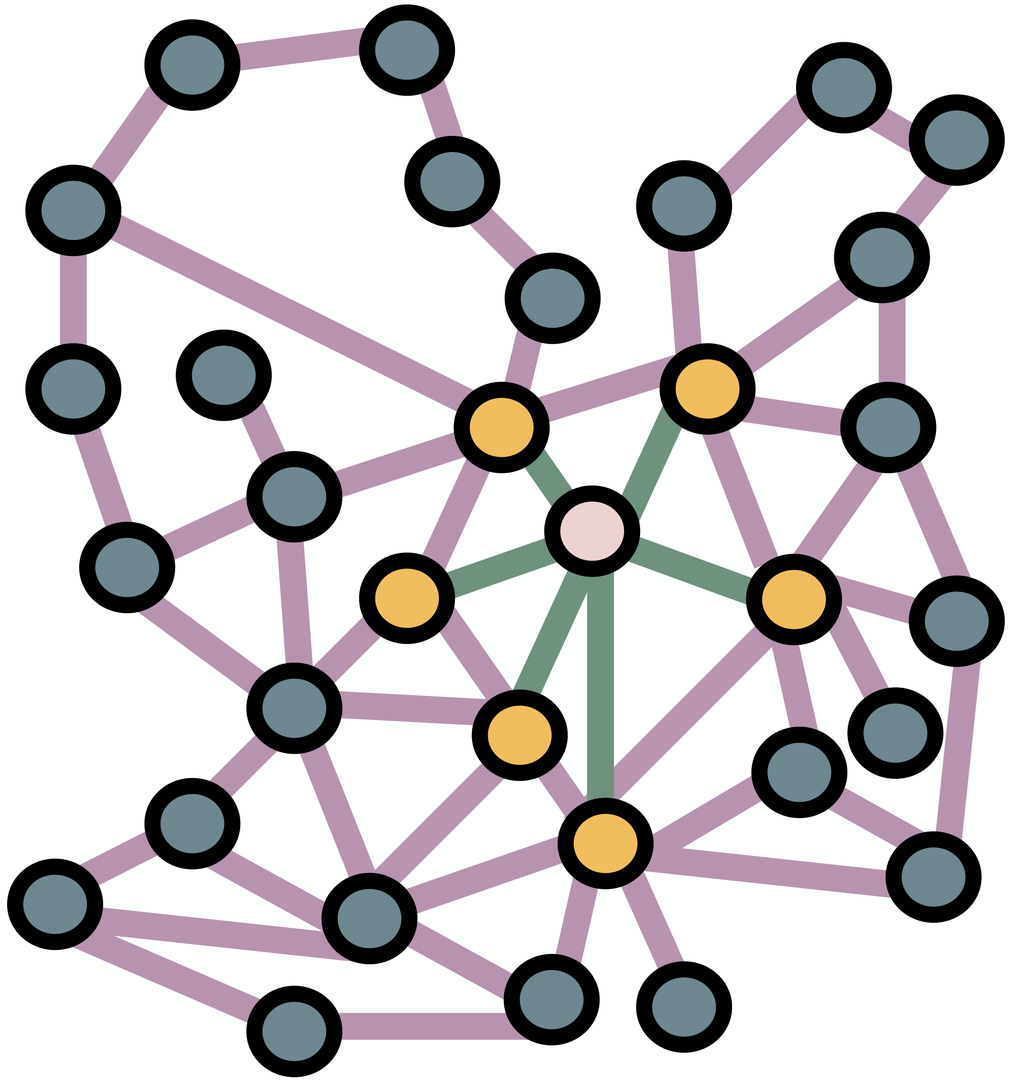
Distance gives Geometry to a Graph



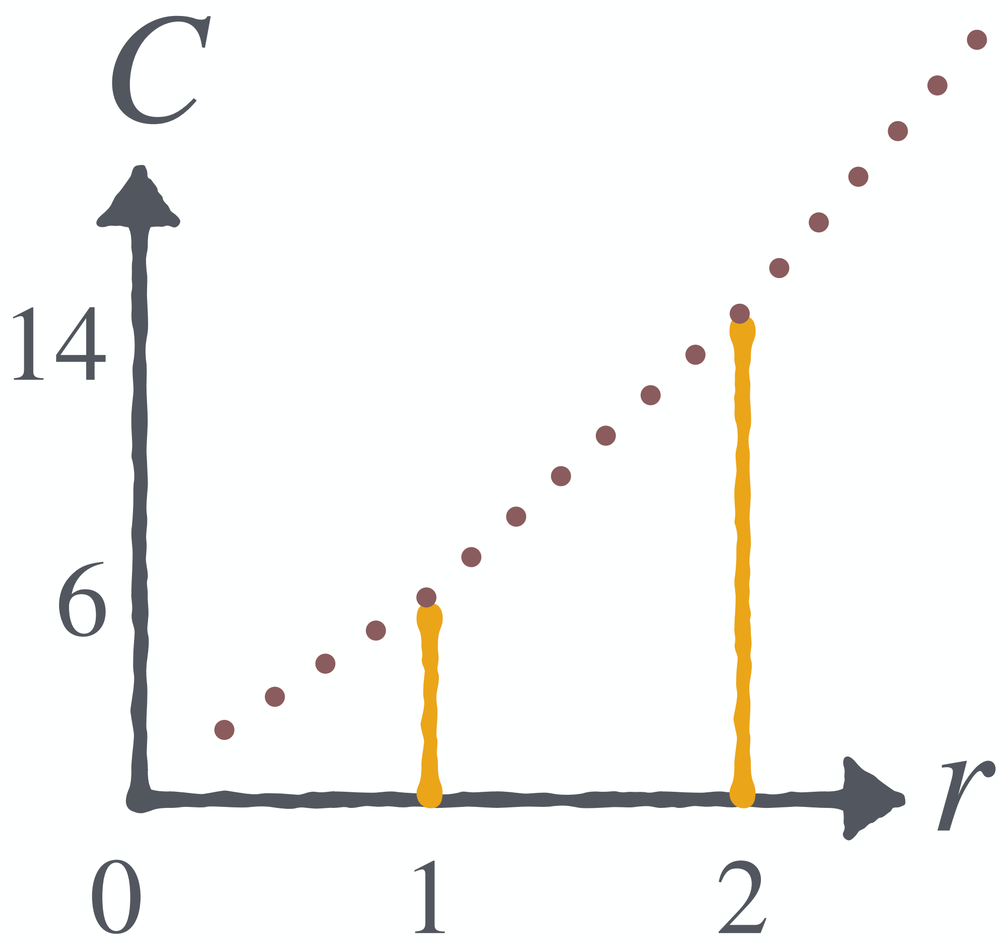


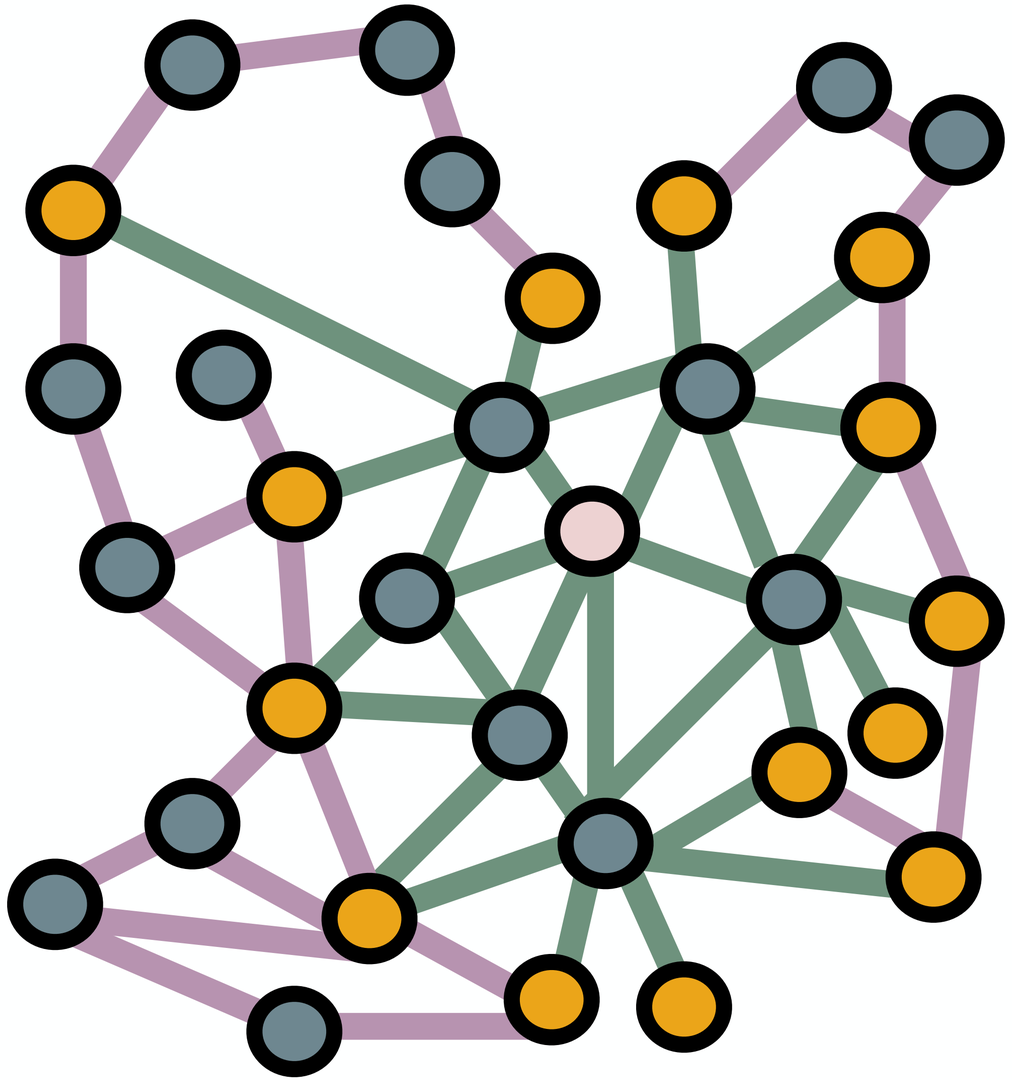
Distance gives Geometry to a Graph

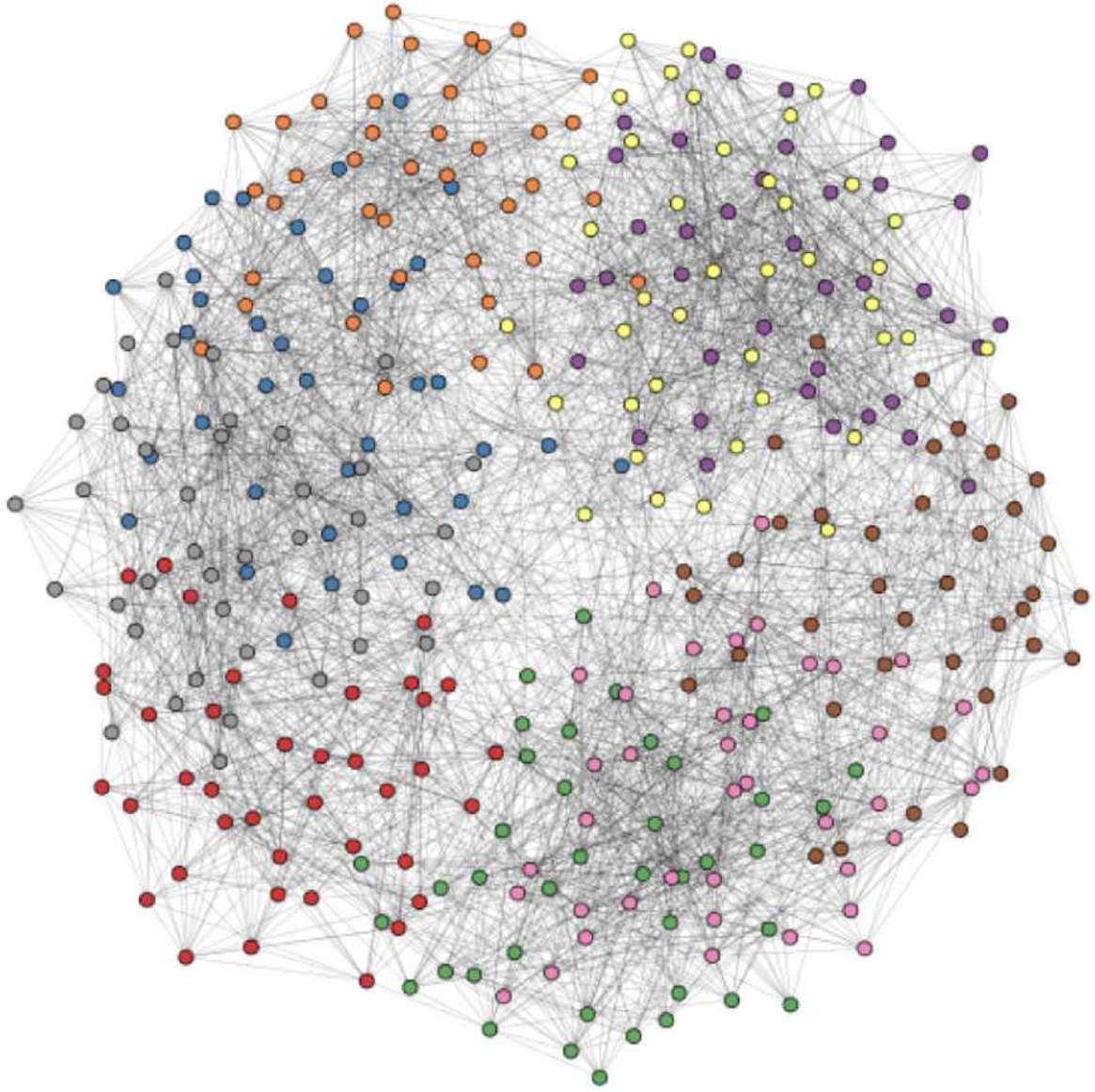




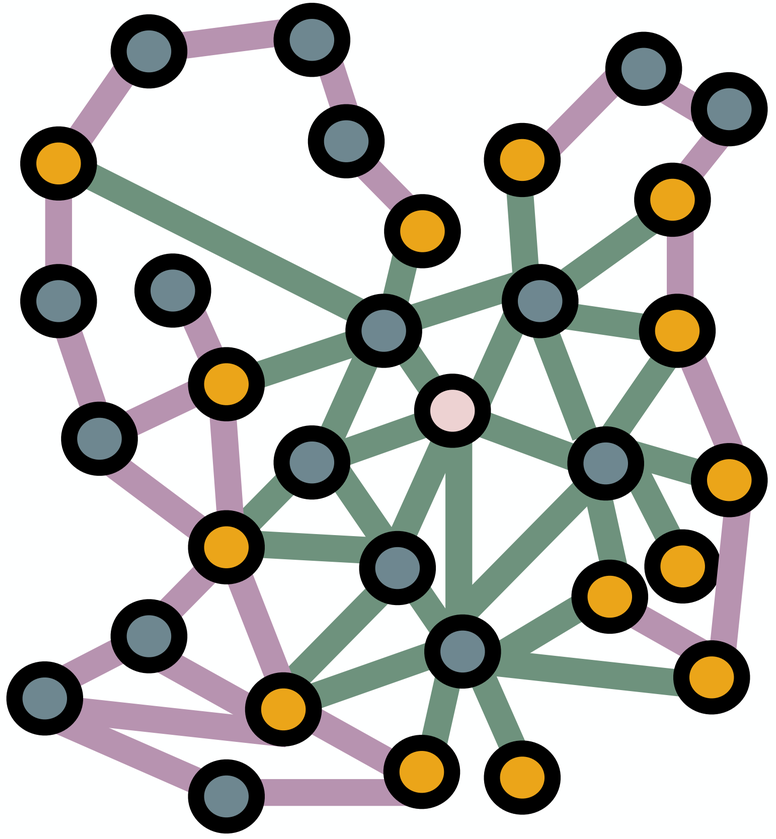
Distance gives Geometry to a Graph







Unfortunately, this can be difficult to compute

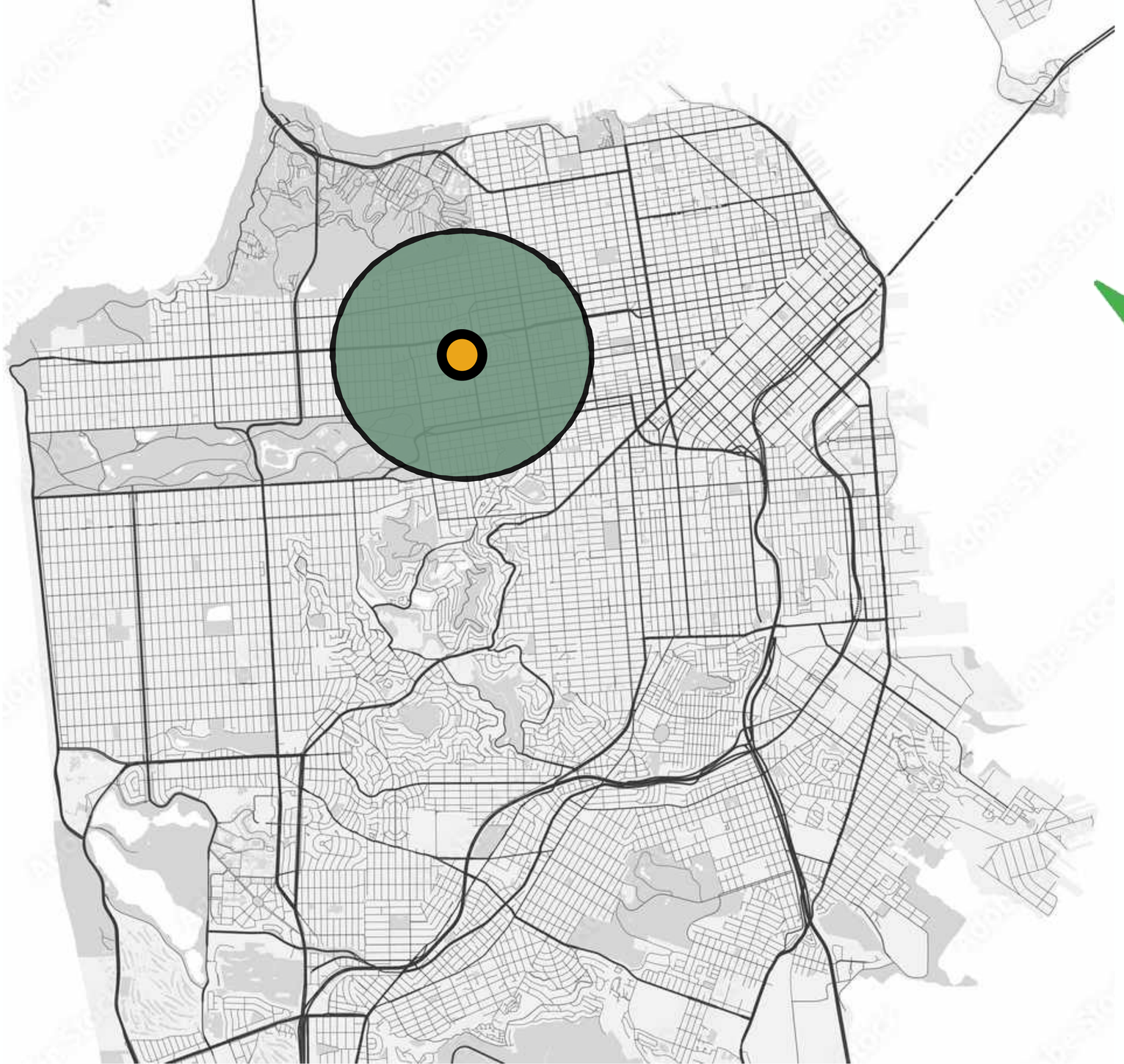


But not always! It's easy on this graph.

Finding Distances

But not always! It's easy on this graph.

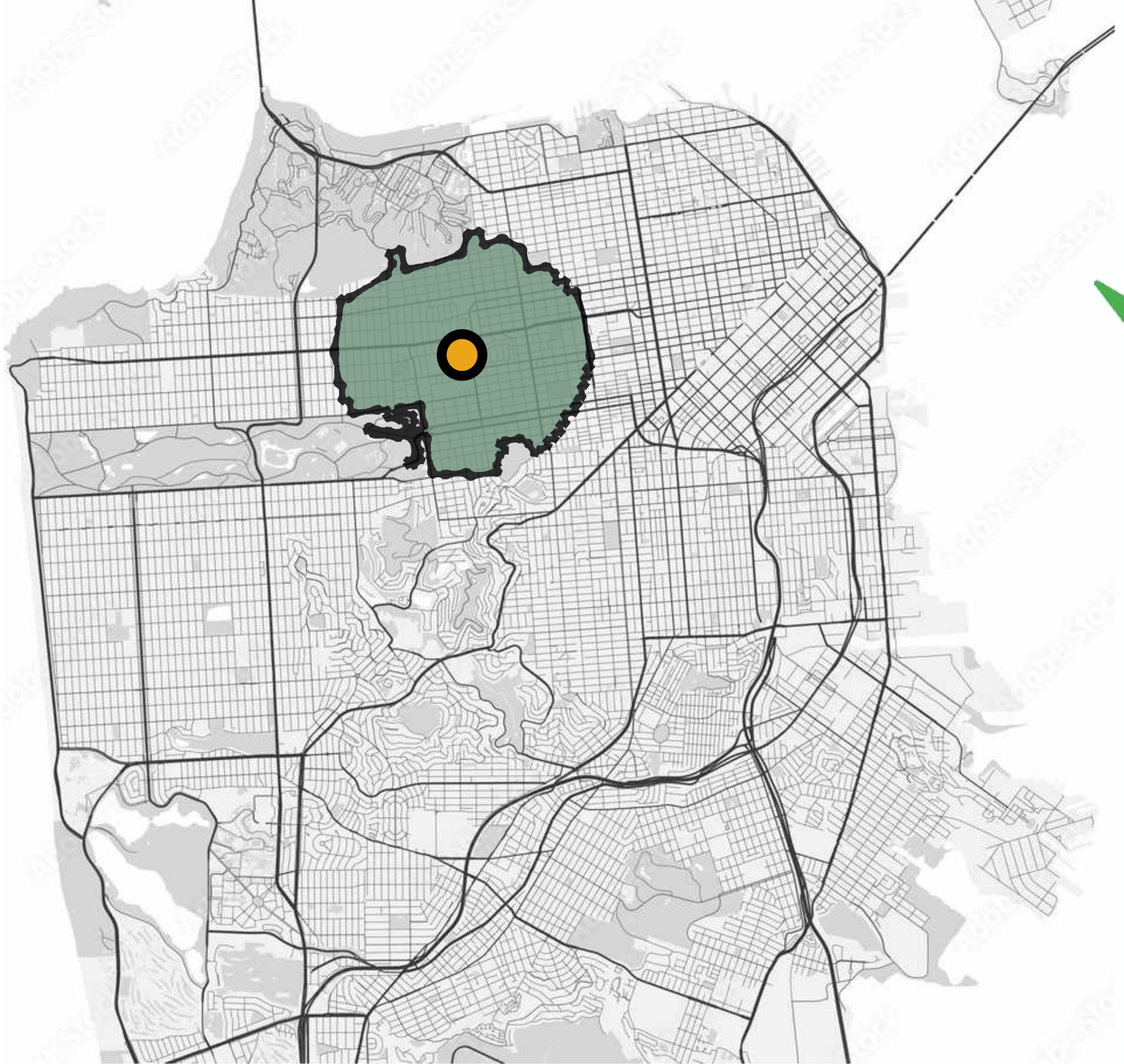
✓ Finding Distances



But not always! It's easy on this graph.

✓ Finding Distances

Finding balls



But not always! It's easy on this graph.

✓ Finding Distances

✓ Finding balls

Why its easy isn't a feature of the graph, but rather a **property of how its laid out**

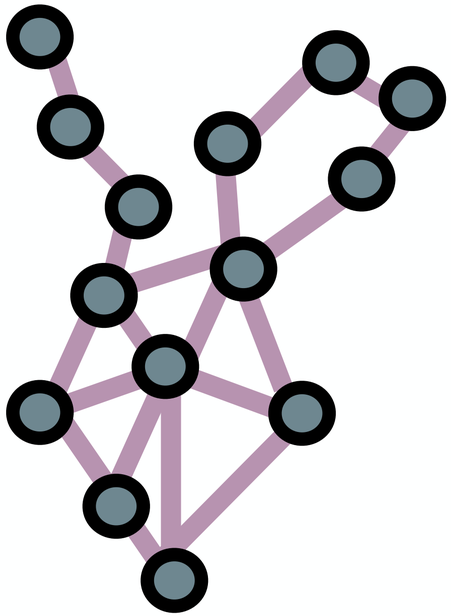
Graph distances match well with distances in the plane.

This lets us use plane geometry to estimate graph quantities.

Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning

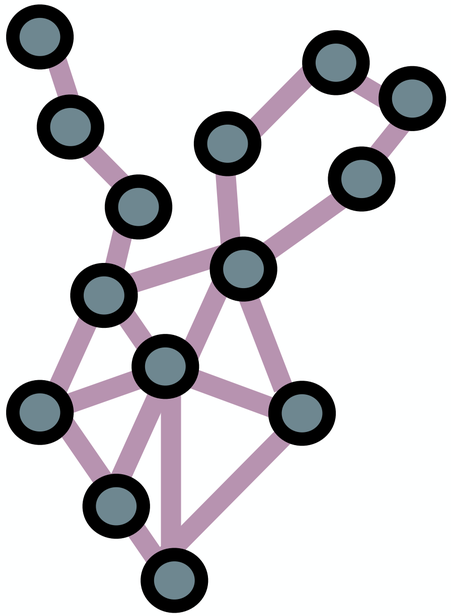
Start with a graph
G with V vertices

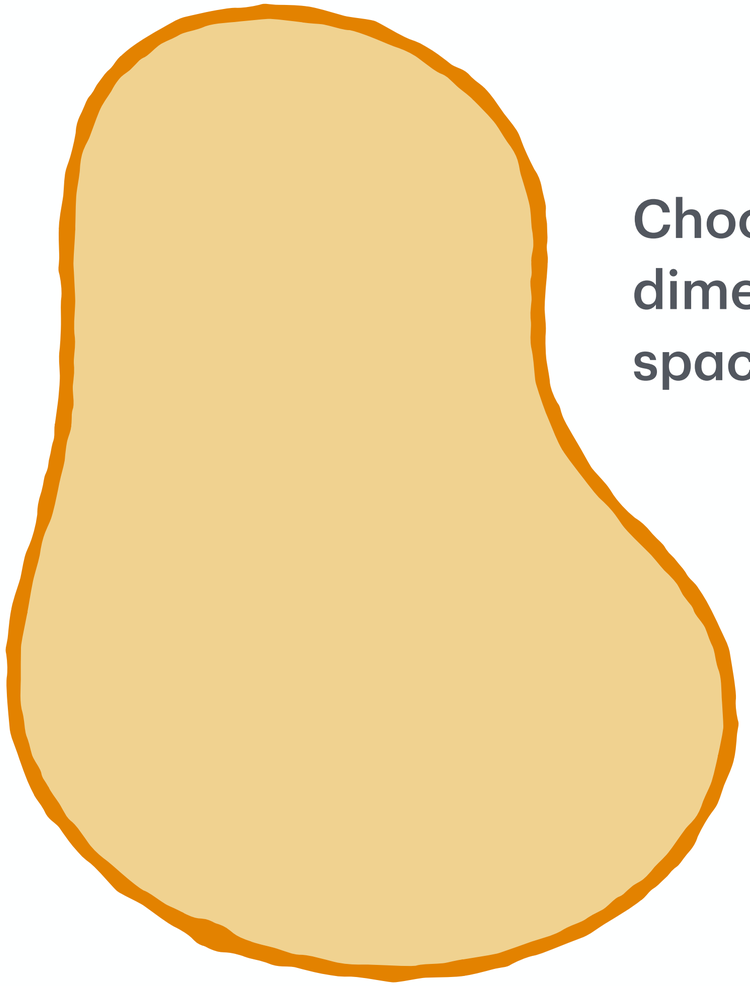


Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning

Start with a graph
G with V vertices



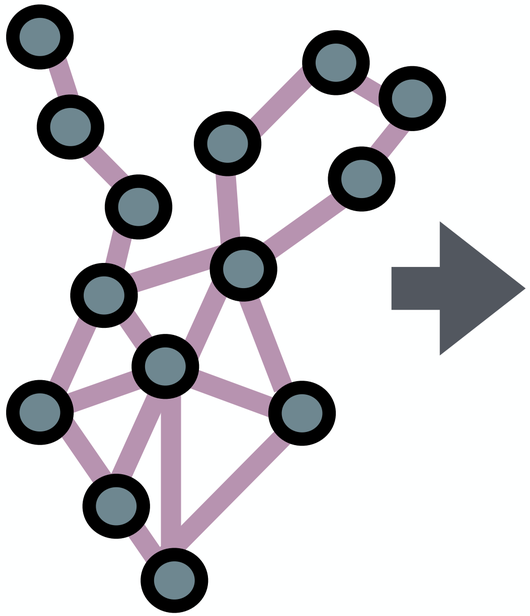


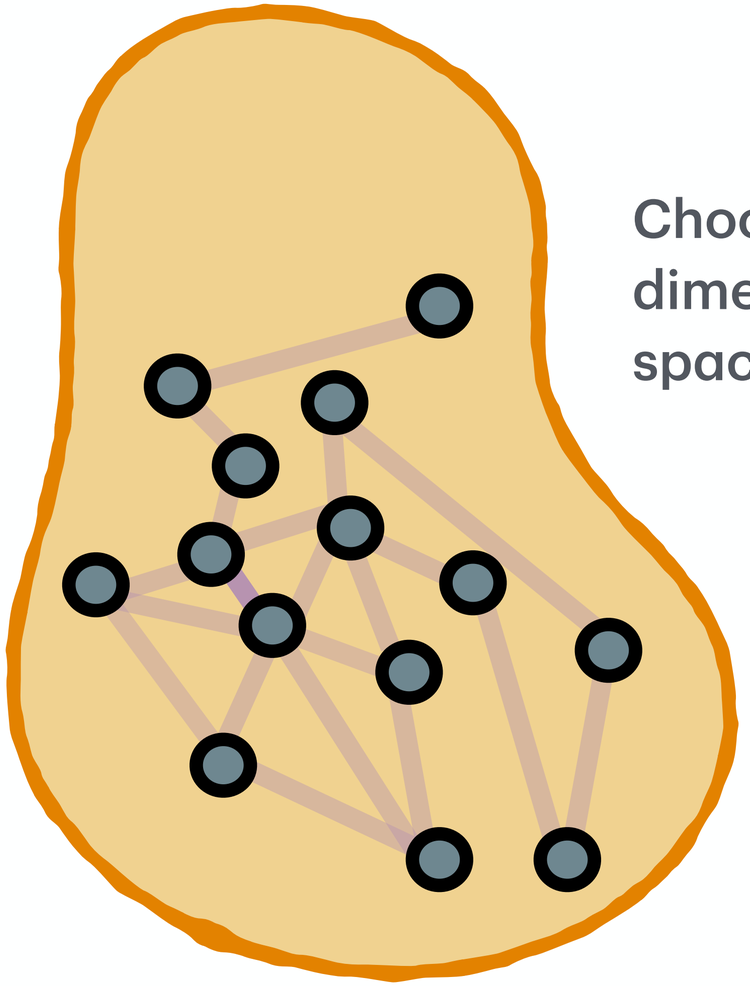
Choose a N-dimensional space X

Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning

Start with a graph G with V vertices





Choose a N-dimensional space X

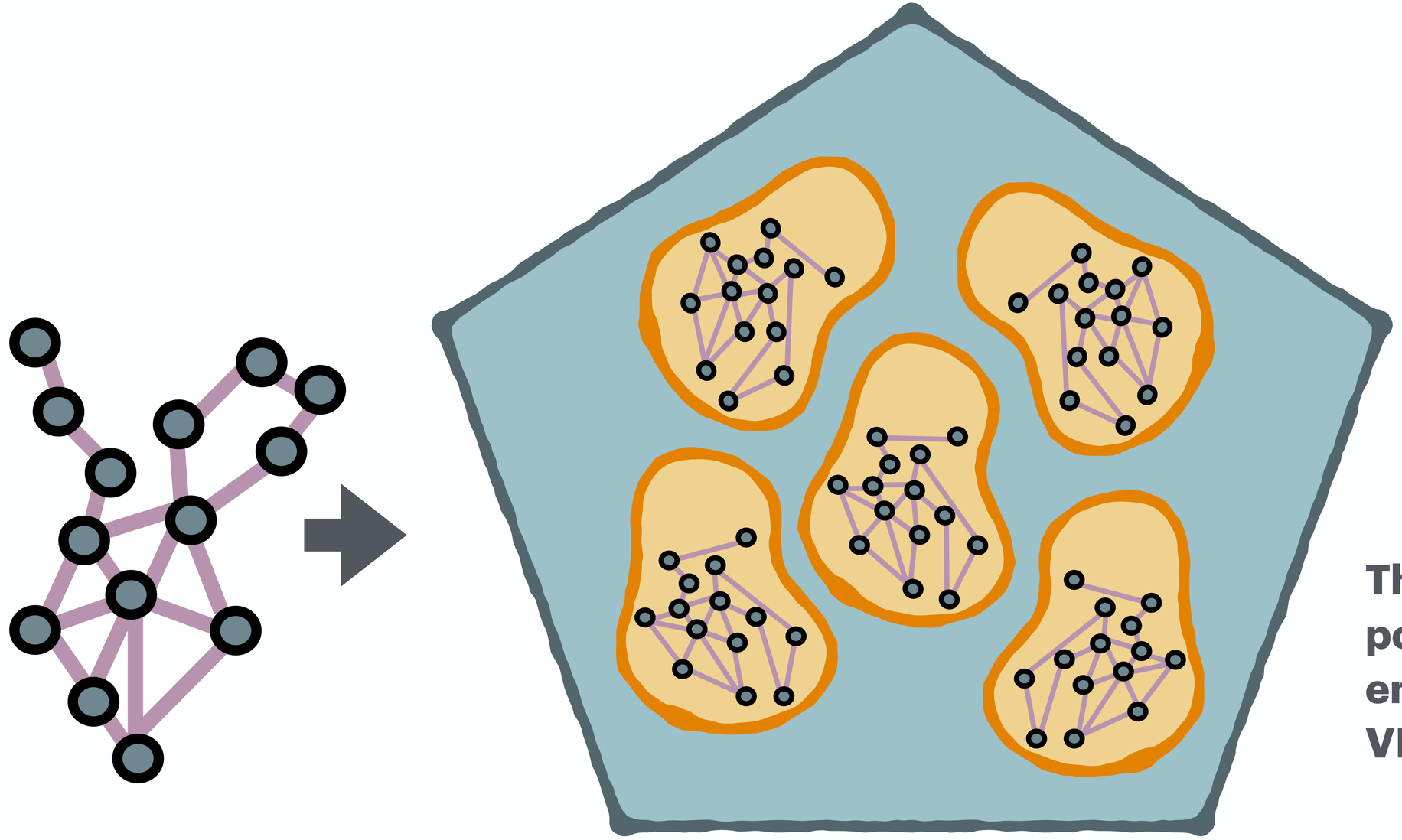
An embedding of the G in X is a choice of V points.

Each point is specified by N numbers.

An embedding of G in X is specified by VN numbers.

Embedding graphs in space

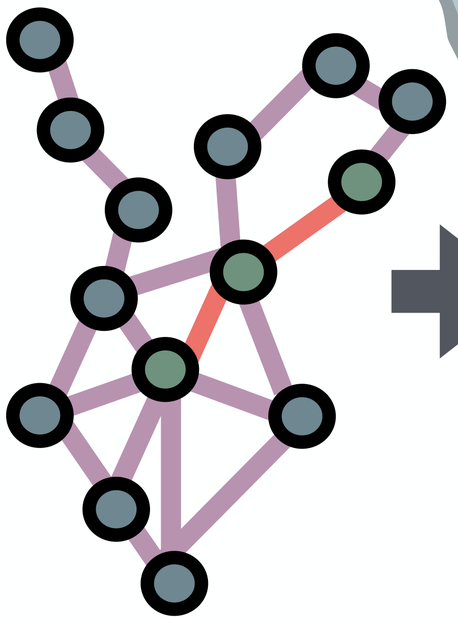
Beautiful Ideas at the Heart of Machine Learning

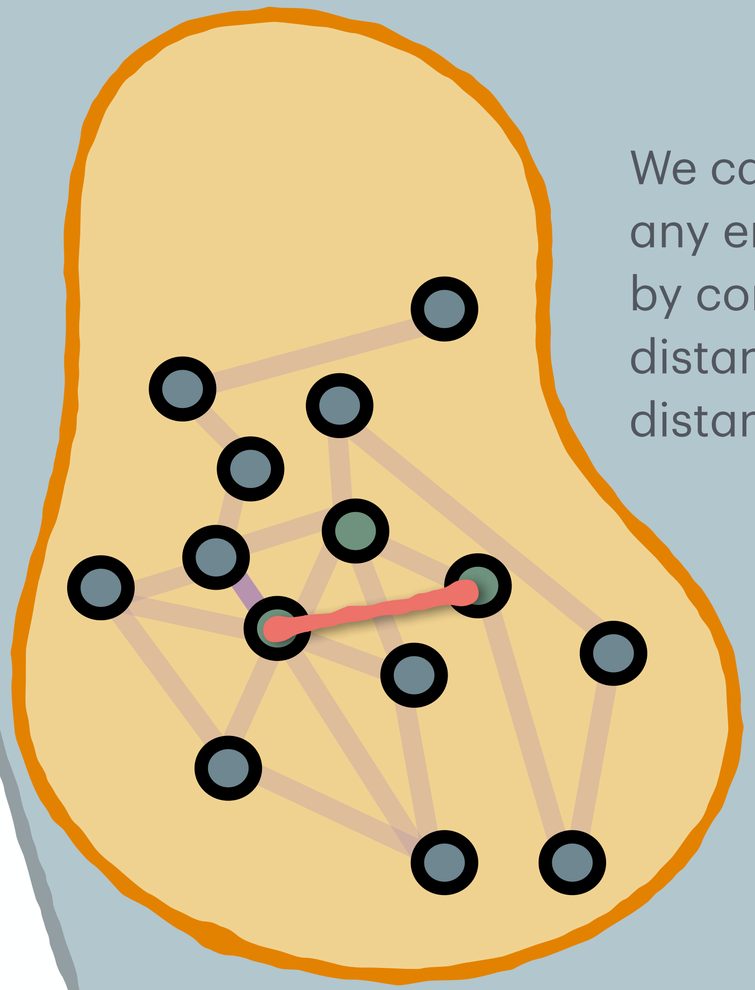


The space of all possible embeddings is VN dimensional.

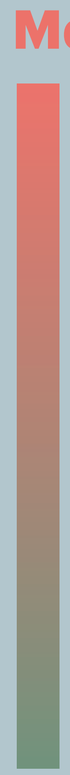
Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning



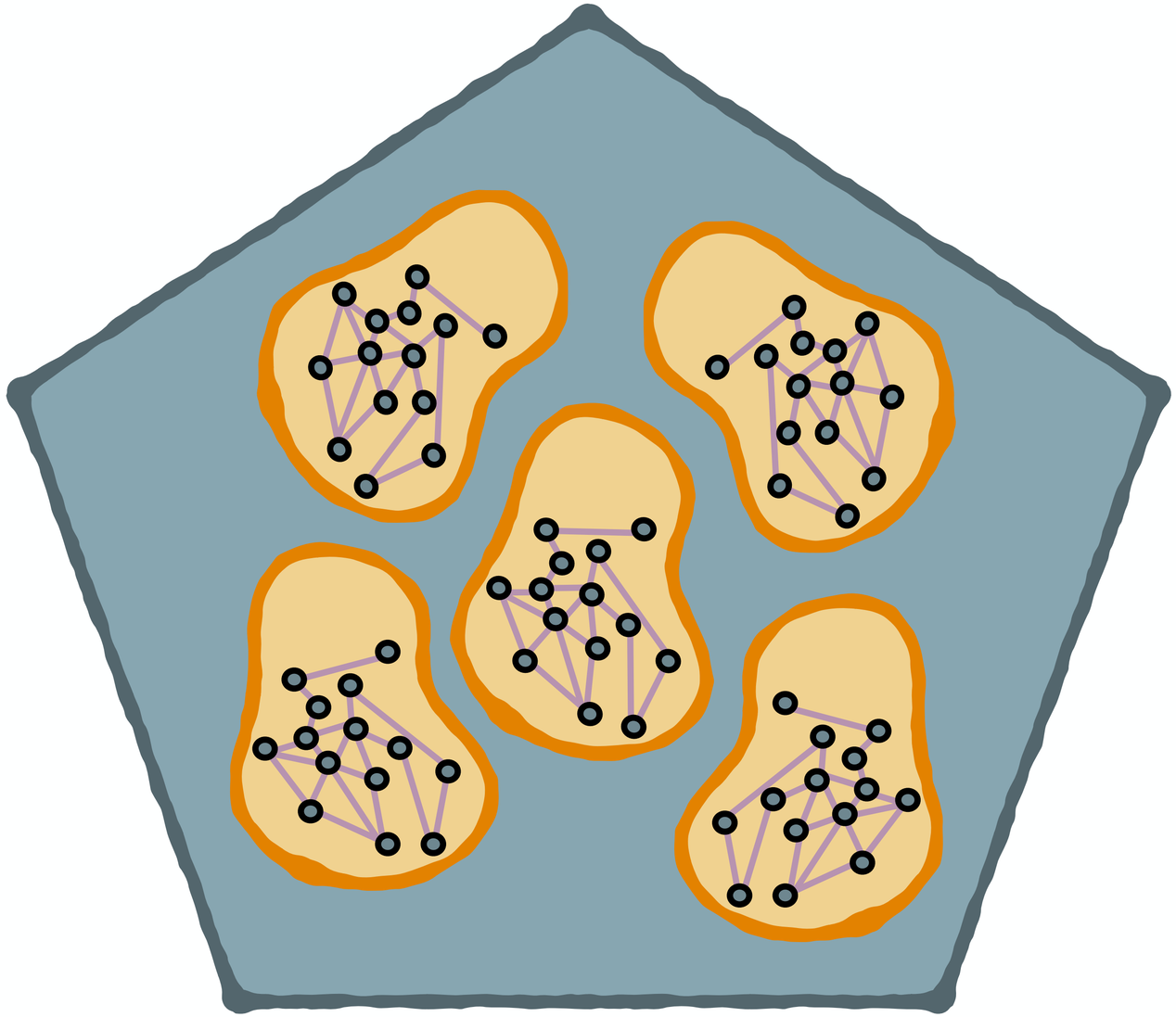


We can see how good any embedding does by comparing graph distances to geometric distances



Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning



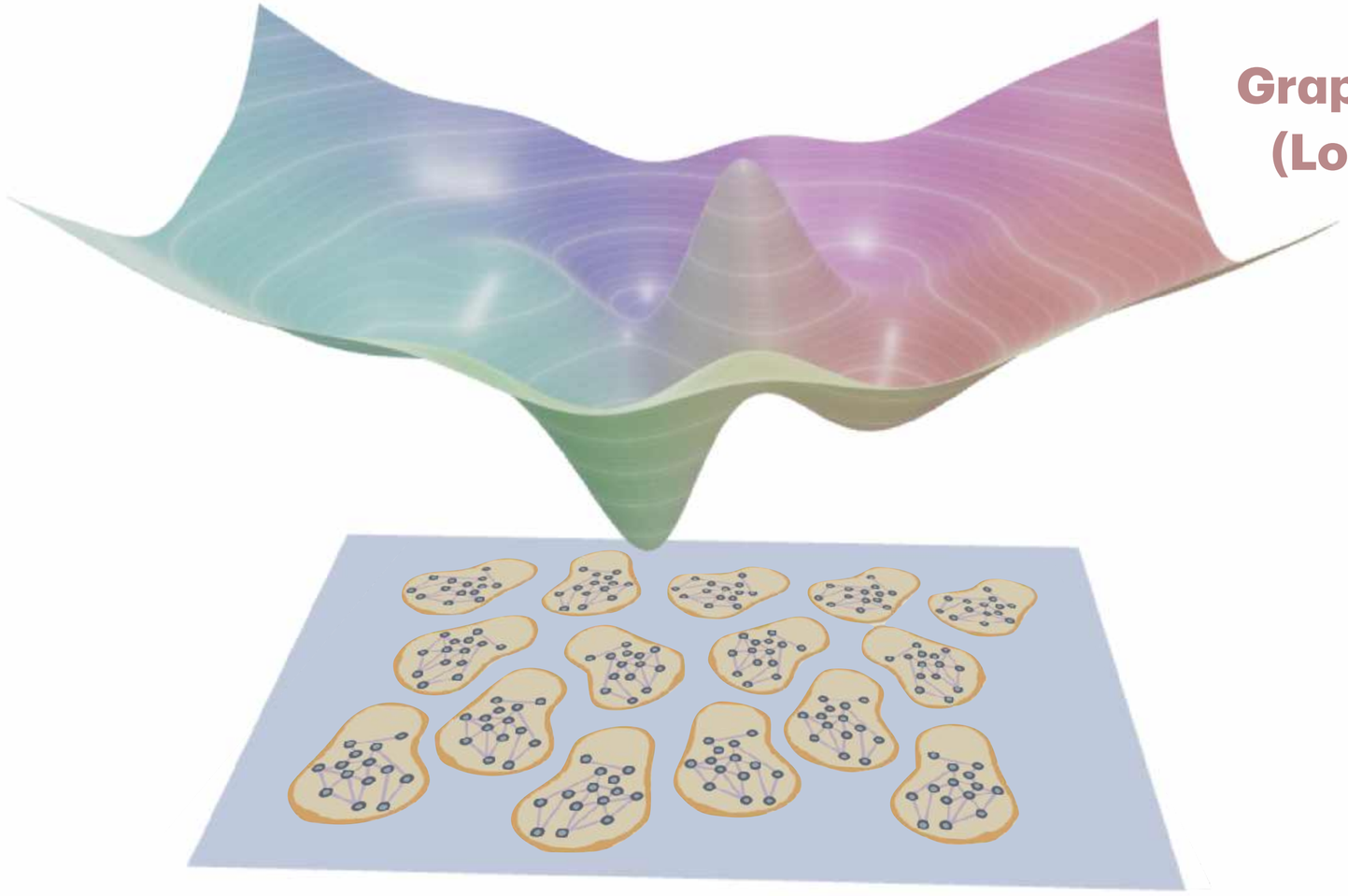
This defines a real valued function on the space of embeddings

Many Errors

Few Errors

Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning



Graph of the Error
(Loss Function)

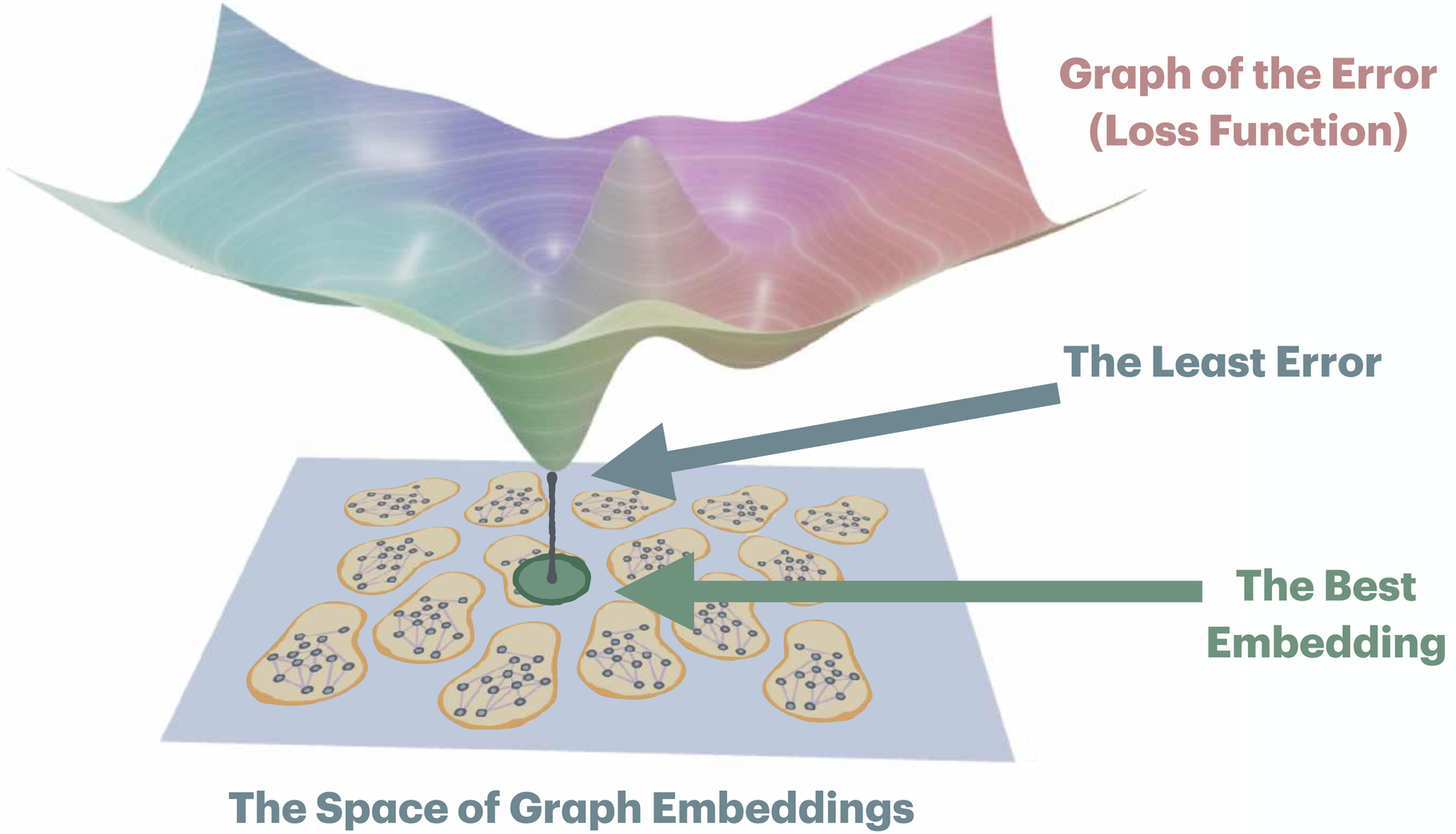
Many Errors

Few Errors

The Space of Graph Embeddings

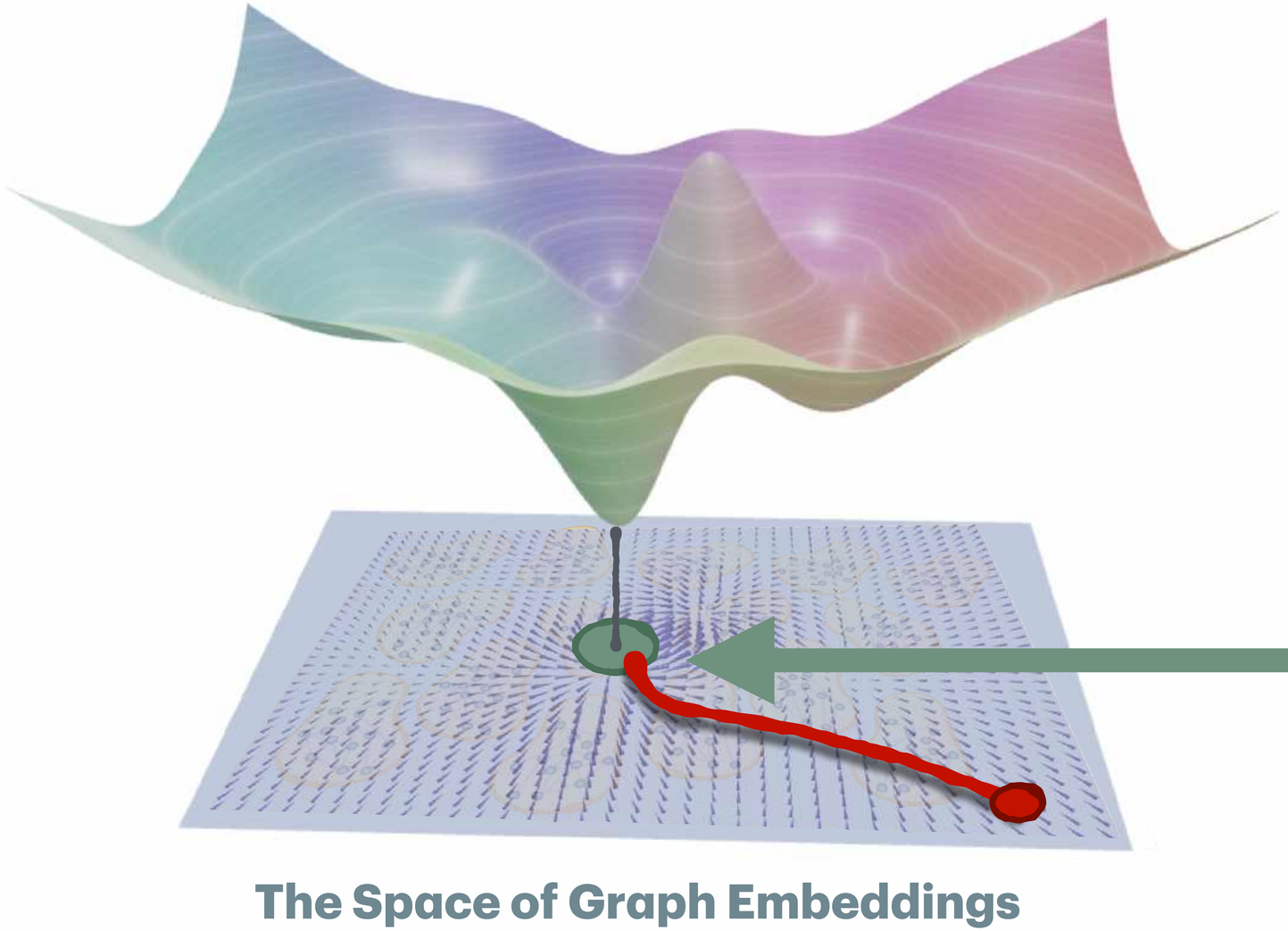
Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning



Embedding graphs in space

Beautiful Ideas at the Heart of Machine Learning



Gradient Descent

$-\nabla \text{Error}$ points in the direction of *steepest decrease*.

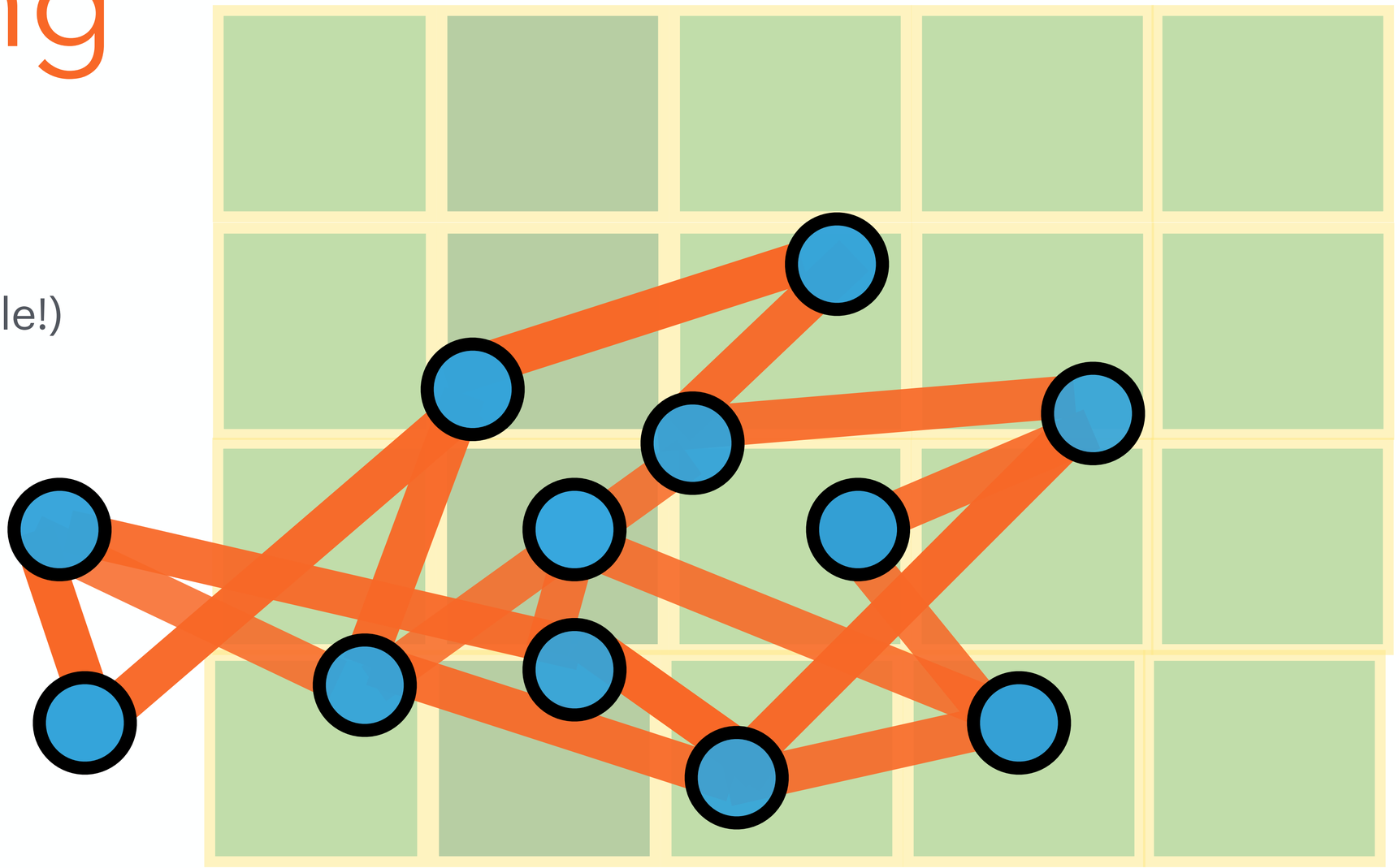
Starting at a random point, following the gradient is one tool to help find a minimum

Graph Embedding

An example:

Randomly assign a point to each vertex, giving a (terrible!) embedding.

Run Gradient Descent



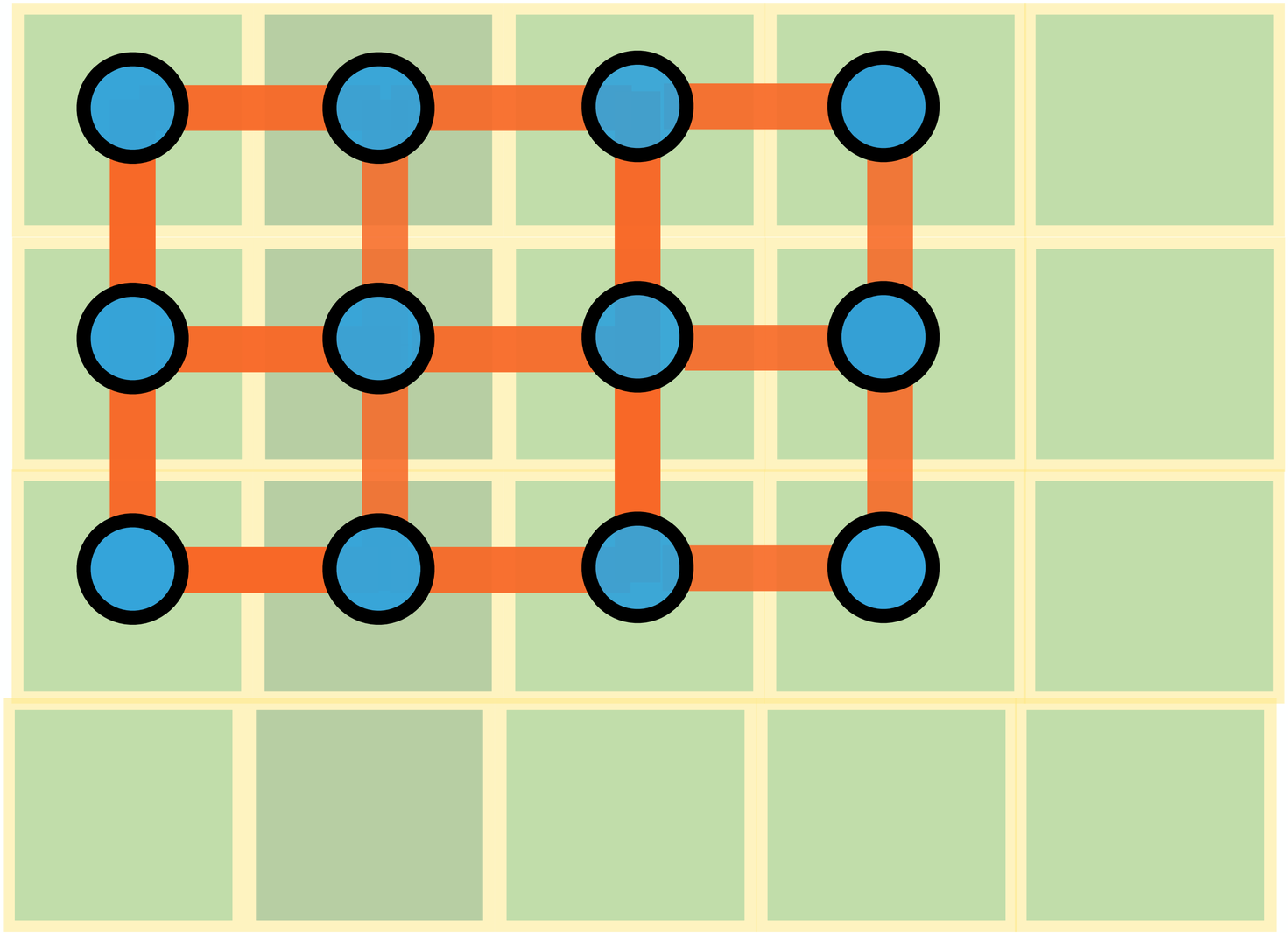
Graph Embedding

An example:

Randomly assign a point to each vertex, giving a (terrible!) embedding.

Run Gradient Descent

Produce an optimal embedding.



Graph Embedding

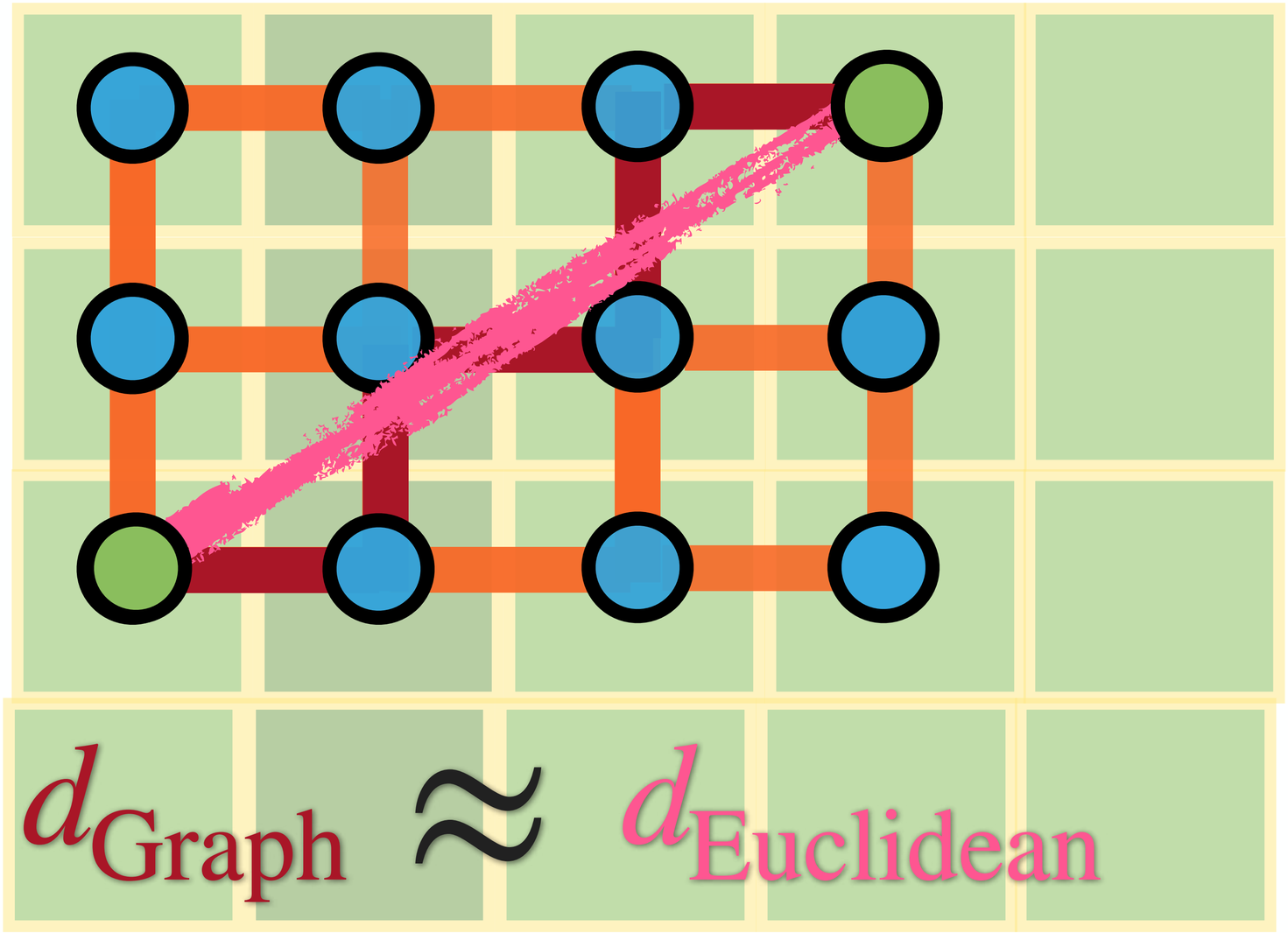
An example:

Randomly assign a point to each vertex, giving a (terrible!) embedding.

Run Gradient Descent

Produce an optimal embedding.

Using this embedding, (approximate) graph computations are easy!

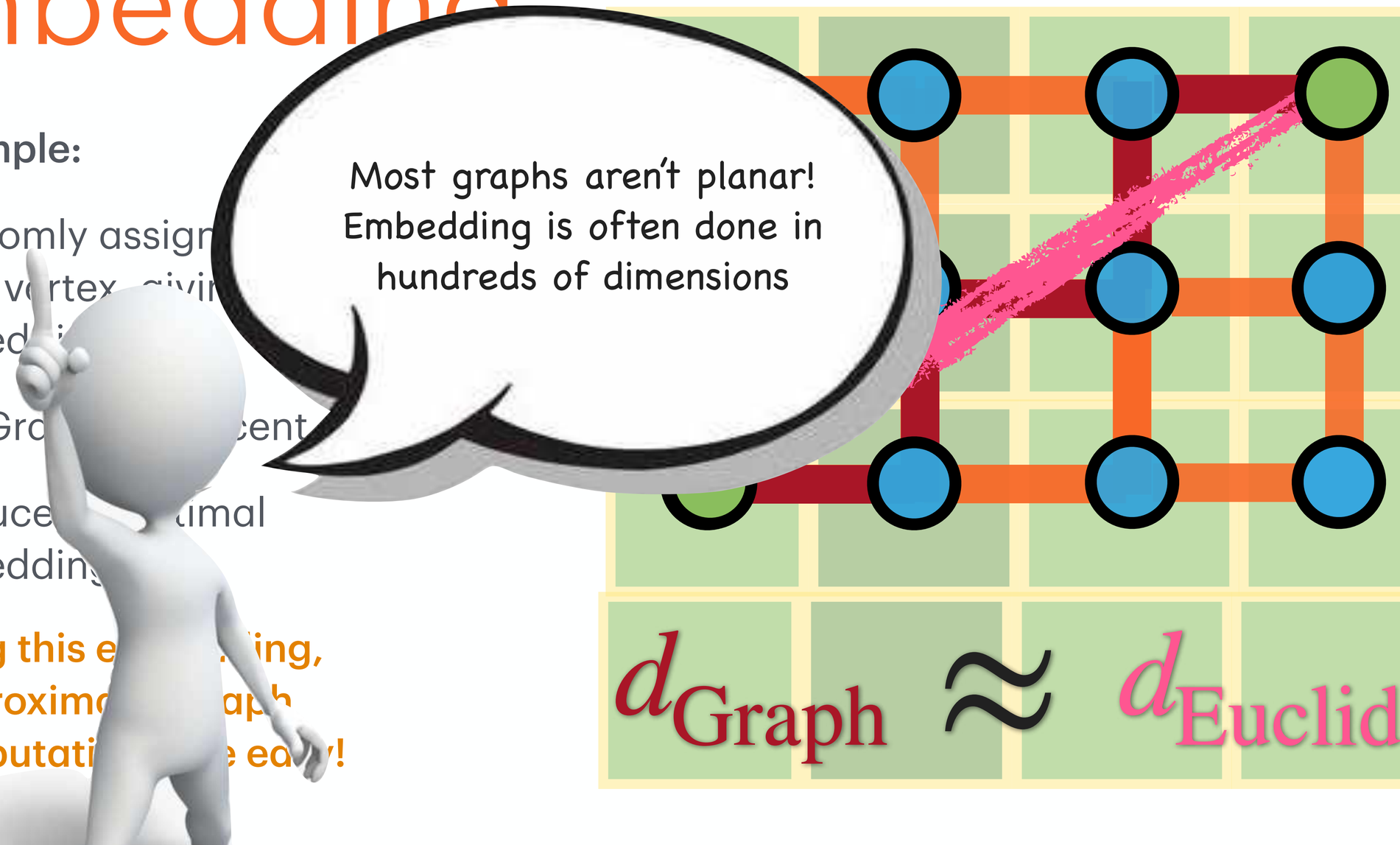


Graph Embedding

An example:

- Randomly assign each vertex, giving embedding
- Run Gradient Descent
- Produce optimal embedding

Using this embedding, (approximate) graph computations are easy!



Graph Embedding Techniques, Applications, and Performance: A Survey

Palash Goyal and Emilio

University of Southern California, Informa
4676 Admiralty Way, Suite 1001. Marina d

Abstract

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. XX, NO. XX, SEPT 2017

A Comprehensive Survey of Graph E
Problems, Techniques and Applic

Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Cha

pears in a wide diversity of real-world scena
hind the data, and thus can benefit a lot of us
However, most graph analytics methods suf
way to solve the graph analytics problem. It c
on and graph properties are maximumly pre
edding. We first introduce the formal definitio

A Tutorial on Network Embeddings

Haochen Chen[1], Bryan Perozzi[2], Rami Al-Rfou[2], and Steven Skiena[1]

[1]Stony Brook University
[2]Google Research
{haocchen, skiena}@cs.stonybrook.edu, bperozzi@acm.org, rmyeid@google.com

August 9, 2018

Abstract

Network embedding methods aim at learning low-dimensional latent representation of nodes

The Link-Prediction Problem for So

David Liben-Nowell
Department of Computer Science
Carleton College
Northfield, MN 55057 USA
dlibenno@carleton.edu

Jor
Department
Corn
Ithaca,
kleinber

Abstract

Given a snapshot of a social network, can we infer which ne
are likely to occur in the near future? We formalize this questi
and we develop approaches to link prediction based on measu
of nodes in a network. Experiments on large co-authorship n
about future interactions can be extracted from network topo
measures for detecting node proximity can outperform more

1 Introduction

As part of the recent surge of research on large, complex netw
erable amount of attention has been devoted to the computa
structures whose nodes represent people or other entities embe
edges represent interaction, collaboration, or influence between
networks include the set of all scientists in a particular disciplin
co-authored papers; the set of all employees in a large compa
on a common project; or a collection of business leaders, with
together on a corporate board of directors. The increased av
encoding such networks has stimulated extensive study of th
fication of recurring structural features. (See, for example, th
Watts [37], Grossman [11], Newman [29], and Adamic and
survey, Newman [30].)

Social networks are highly dynamic objects; they grow an
the addition of new edges, signifying the appearance of new i
structure. Identifying the mechanisms by which they evolve is

JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

Deep Learning on Graph

Ziwei Zhang, Peng Cui and Wenwu Zhu,

Abstract—Deep learning has been shown to be successful in a number of domains, ra

Network Representation Learning: A Survey

Daokun Zhang, Jie Yin, Xingquan Zhu *Senior Member, IEEE*, Chengqi Zhang *Senior Member, IEEE*

Abstract—With the widespread use of information technologies, information networks are becoming increasingly popular to capture
rious disciplines, such as social networks, citation networks, telecommunication networks, and
ese networks sheds light on different aspects of social life such as the structure of societies,
unication patterns. In reality, however, the large scale of information networks often makes network
xpensive or intractable. Network representation learning has been recently proposed as a new learning
lices into a low-dimensional vector space, by preserving network topology structure, vertex content,
facilitates the original network to be easily handled in the new vector space for further analysis. In this
sive review of the current literature on network representation learning in the data mining and machine
axonomies to categorize and summarize the state-of-the-art network representation learning
erlying learning mechanisms, the network information intended to preserve, as well as the algorithmic
summarize evaluation protocols used for validating network representation learning including
evaluation methods, and open source algorithms. We also perform empirical studies to compare the
lgorithms on common datasets, and analyze their computational complexity. Finally, we suggest
facilitate future study.

orks, graph mining, network representation learning, network embedding.

coming ubiquitous across a
d applications in forms of
rks, telecommunication net-
etc. The scale of these net-
millions or even billions of
ion networks plays a crucial
plications across many dis-
networks, classifying users
is useful for many impor-
, targeted advertising and
ication networks, detecting
better understand the rumor
l networks, inferring inter-
acilitate new treatments for
analysis of these networks

most existing algorithms involve calculating the spectral decomposition of a matrix [2] with at least quadratic time complexity with respect to the number of vertices. This computational overhead makes algorithms hard to scale to large-scale networks with millions of vertices.

Recently, network representation learning (NRL) has aroused a lot of research interest. NRL aims to learn latent, low-dimensional representations of network vertices, while preserving network topology structure, vertex content, and other side information. After new vertex representations are learned, network analytic tasks can be easily and efficiently carried out by applying conventional vector-based machine learning algorithms to the new representation space. This obviates the necessity for deriving complex algorithms that are applied directly on the original network.

Representation Learning on Graphs: Methods and Applications

William L. Hamilton
wleif@stanford.edu

Rex Ying
rexying@stanford.edu

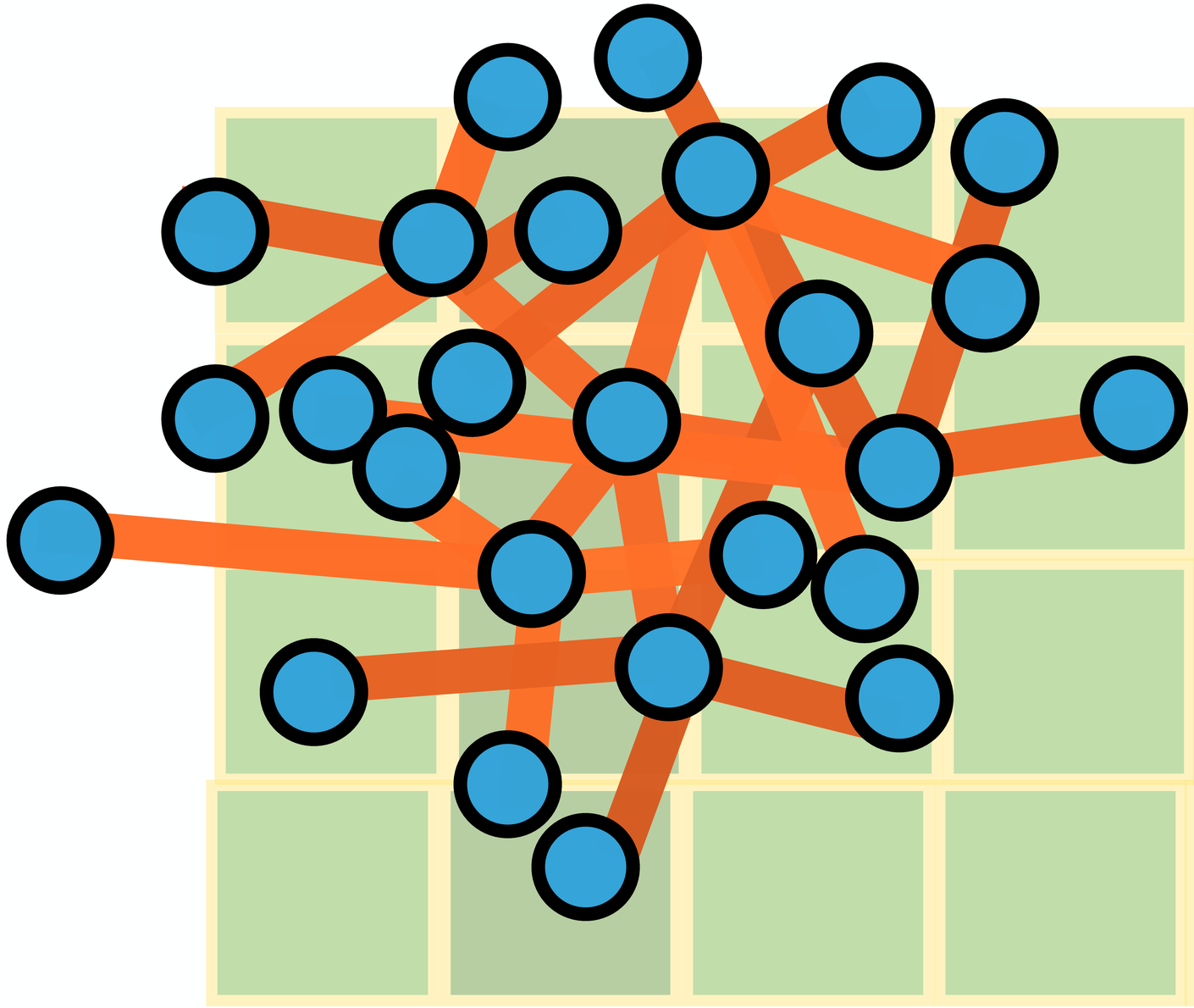
Jure Leskovec
jure@cs.stanford.edu

Department of Computer Science
Stanford University
Stanford, CA, 94305

Abstract

Machine learning on graphs is an important and ubiquitous task with applications ranging from drug design to friendship recommendation in social networks. The primary challenge in this domain is finding a way to represent, or encode, graph structure so that it can be easily exploited by machine learning models. Traditionally, machine learning approaches relied on user-defined heuristics to extract features encoding structural information about a graph (e.g., degree statistics or kernel functions). However, recent years have seen a surge in approaches that automatically learn to encode graph structure into low-dimensional embeddings, using techniques based on deep learning and nonlinear dimensionality reduction. Here we provide a conceptual review of key advancements in this area of representation

Problem: this can’t work all the time!

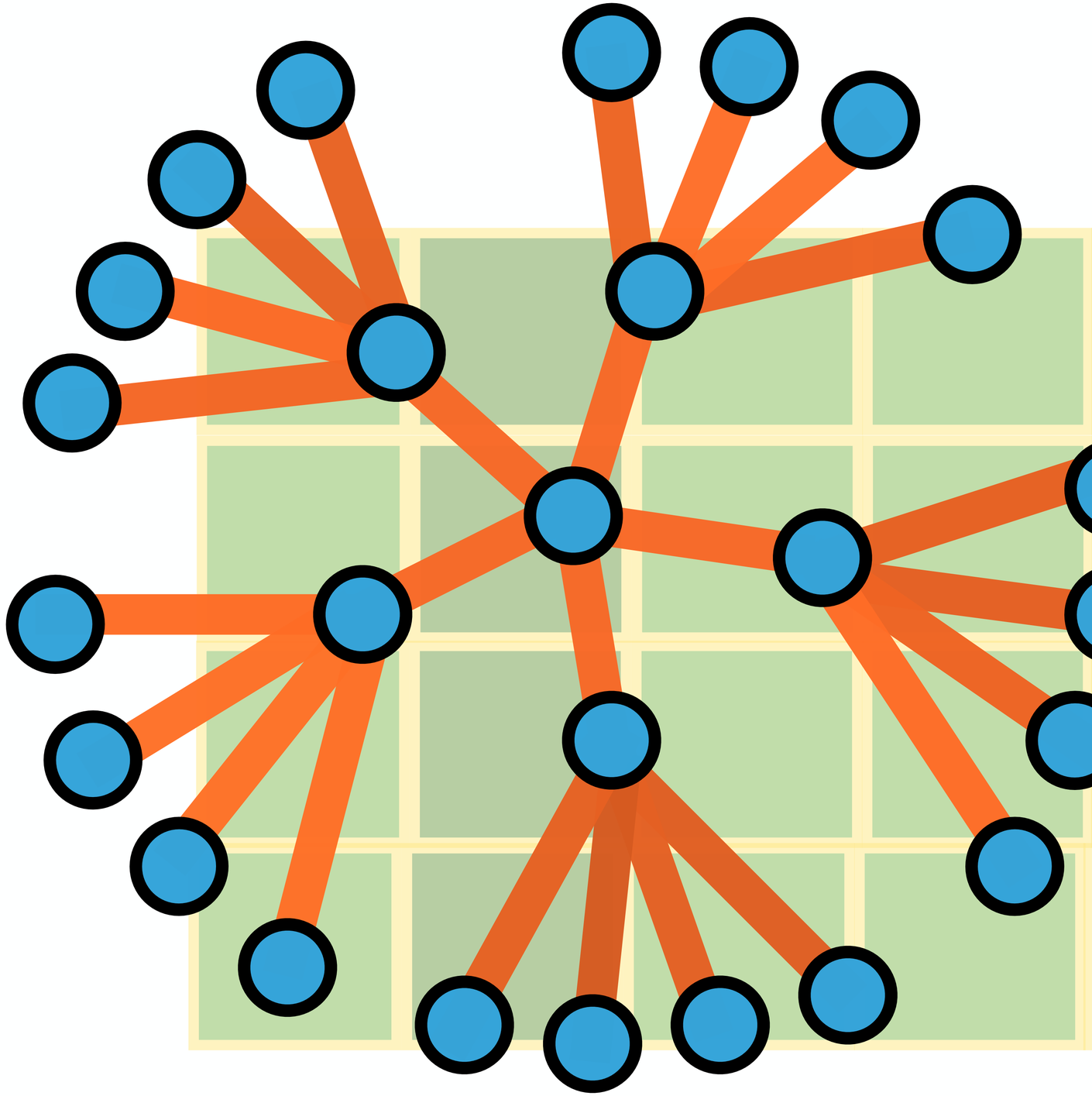


Problem: this can't work all the time!

Volume of balls in n-dimensional space grows like r^n

Balls in a the tree here grow exponentially: like 5^r

Exponentials grow faster than polynomials: big enough trees can't fit in *any* Euclidean space.



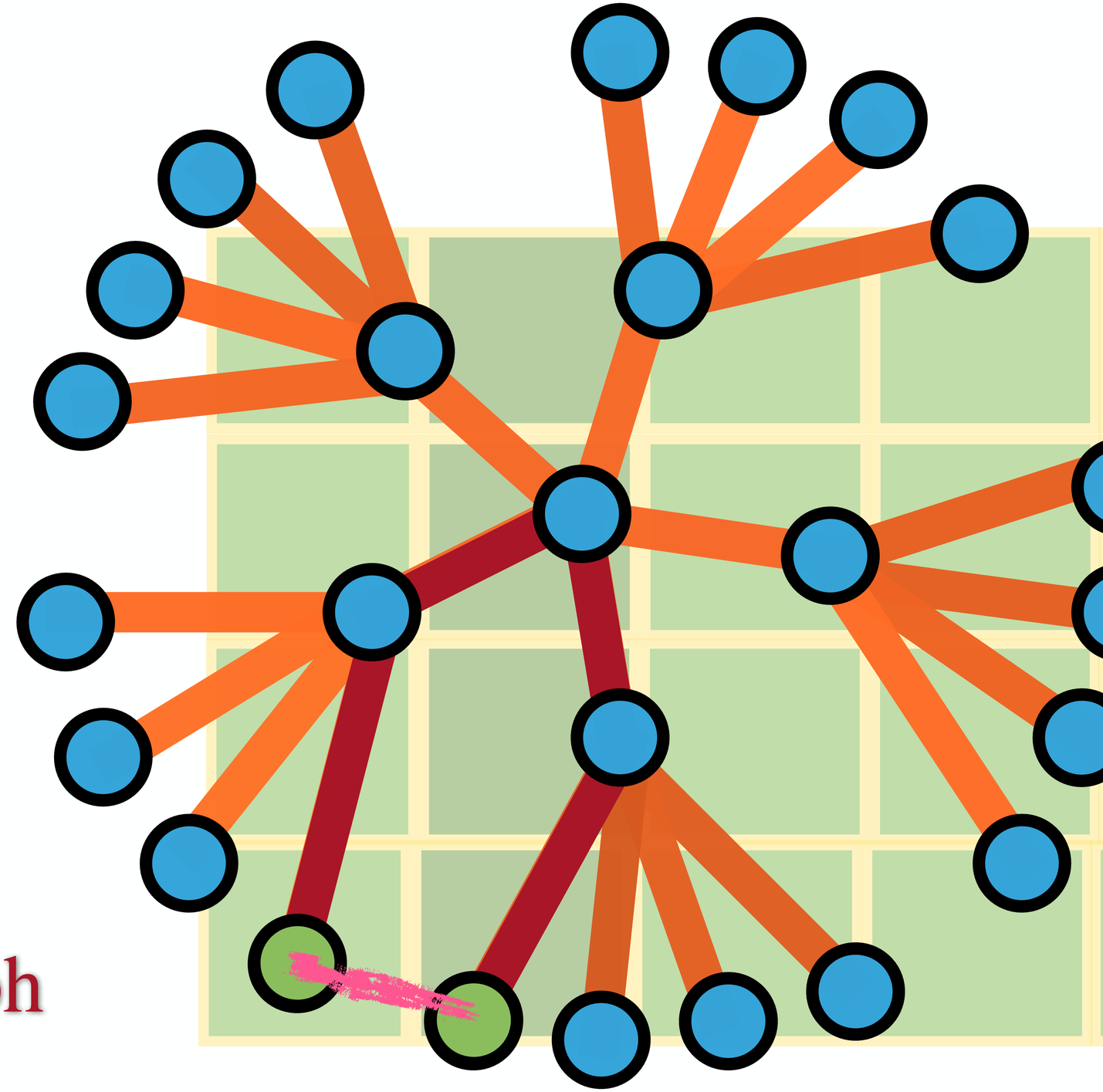
Problem: this can't work all the time!

Volume of balls in n-dimensional space grows like r^n

Balls in a the tree here grow exponentially: like 5^r

Exponentials grow faster than polynomials: big enough trees can't fit in *any* Euclidean space.

$$d_{\text{Euclidean}} << d_{\text{Graph}}$$



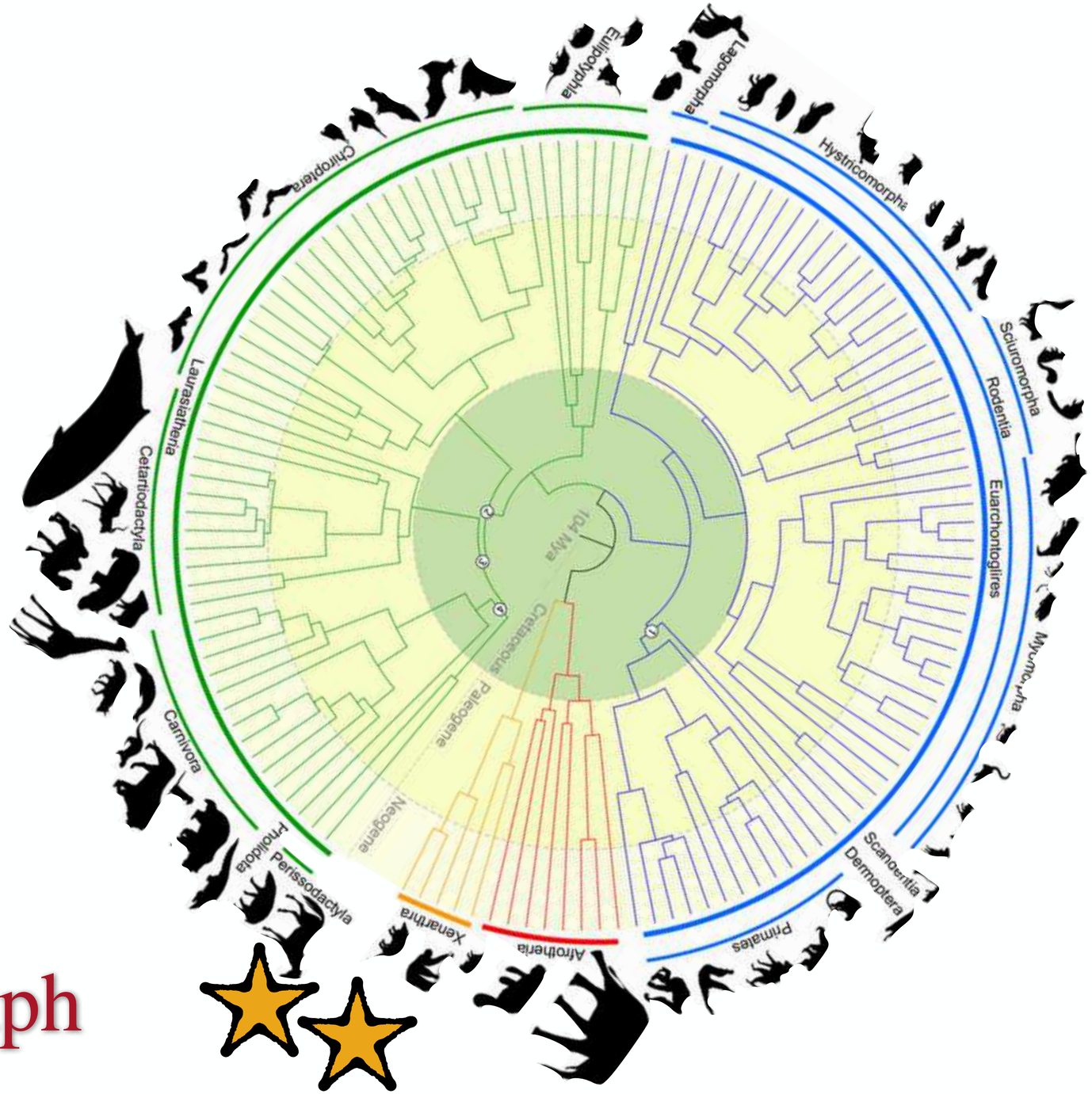
Problem: this can't work all the time!

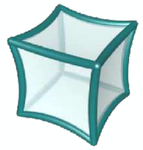
Volume of balls in n-dimensional space grows like r^n

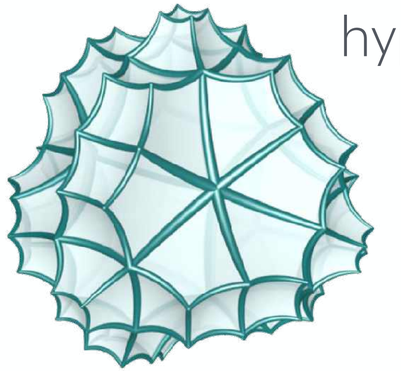
Balls in a the tree here grow exponentially: like 5^r

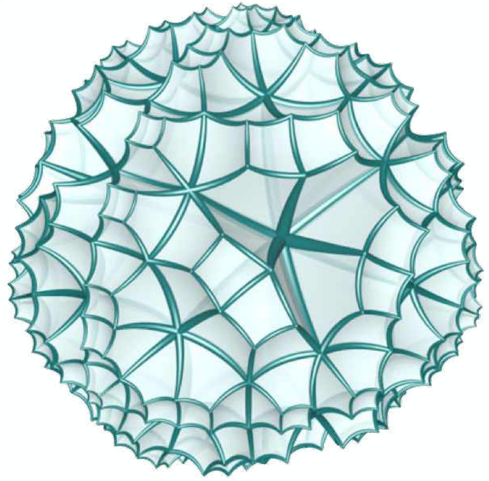
Exponentials grow faster than polynomials: big enough trees can't fit in *any* Euclidean space.

$d_{\text{Euclidean}} << d_{\text{Graph}}$





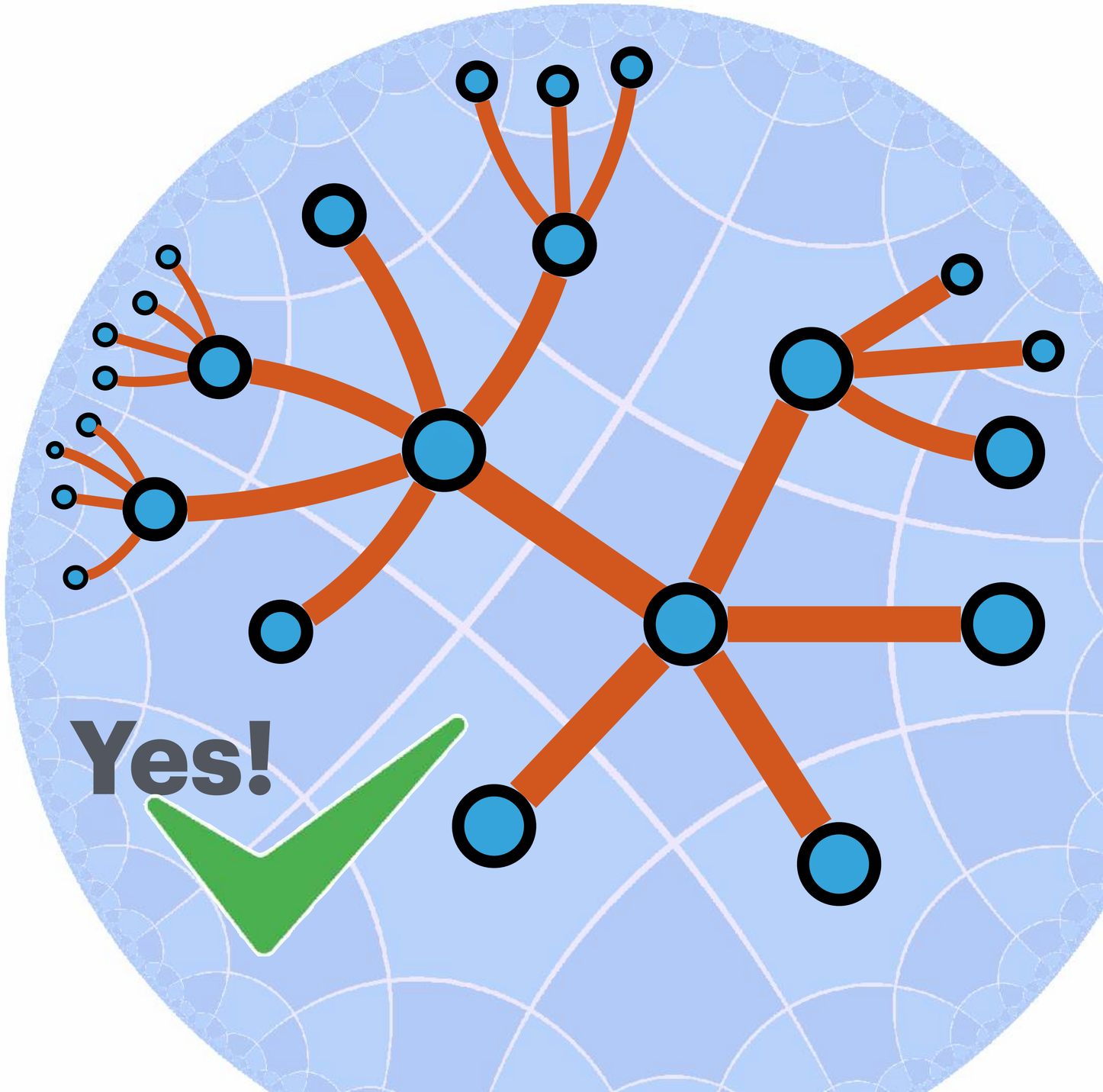




Idea:

The size of balls in negatively curved space grows exponentially!

Can we embed a tree in hyperbolic space?



Hyperbolic Geometry of Complex Networks

Dmitri Krioukov,[1] Fragkiskos Papadopoulos,[2] Maksim Kitsak,[1] Amin Vahdat,[3] and Mari

Neural E

Benjamin Paul Chaml

Hy

Learni
dimens
networ
proble
a nov
graphs
for us
model
geomet
as parti
geodes
entailm
closed
hyperb
the em
show s
over sl
represe

Hyperbolic Graph Convolutional Neural Networks

Ines Chami*† Rex Ying* † Christopher Ré† Jure Leskovec†

Low-Dimensional Knowledge Graph Embeddings via Hyperbolic Rotations

Ines Chami, Adva Wolf, Frederic Sala & Christopher Ré
Department of Computer Science
Stanford University
{chami,advaw,fredsala}@stanford.edu, chrismre@cs.stanford.edu

Abstract

Knowledge graphs (KGs) capture rich relationships between a large number of entities. Embeddings of these structures must preserve these relationships with high fidelity. Recently, hyperbolic embedding methods have achieved state-of-the-art quality in graph representation learning tasks; when embedding certain graphs, they can produce parsimonious embeddings that have higher fidelity while using much fewer dimensions than their Euclidean counterparts. Mirroring work in Euclidean space, we are the first to leverage trainable hyperbolic rotations, a key notion in providing sufficiently rich representations for the complex logical patterns found in KGs. Coupled with trainable curvature, this approach yields improved embeddings in fewer dimensions. We evaluate our method, ROTATIONH, on the WN18RR link prediction task; in the low-dimensional setting, we improve on previous Euclidean-based efforts by 2.2% in mean reciprocal rank (MRR), and in the high-dimensional setting, we achieve a new state-of-the-art MRR of 49.2%, improving on existing methods by 1.1%.

1 Introduction

ldings and Back:
lustering

ratis†† Christopher Ré†

nsionality Reduction via Horospherical Projections

Albert Gu*1 Dat Nguyen*1 Christopher Ré1

neighbor search (Krauthgamer & Lee, 2006; Wu & Charikar, 2020), hierarchical clustering (Monath et al., 2019; Chami et al., 2020a), or dimensionality reduction which is the focus of this work.

Euclidean Principal Component Analysis (PCA) is a fundamental dimensionality reduction technique which seeks directions that best explain the original data. PCA is an important primitive in data analysis and has many important uses such as (i) dimensionality reduction (e.g. for memory efficiency), (ii) data whitening and pre-processing for downstream tasks, and (iii) data visualization.

Here, we seek a generalization of PCA to hyperbolic geometry. Given a core notion of **directions**, PCA involves the following ingredients:

1. A nested sequence of affine **subspaces** (flags) spanned by a set of directions.
2. A **projection** method which maps points to these sub-

Problem: if you try to run this on real world datasets, it does a bad job

Where we got involved!

Me

Anna Wienhard

Michel Strube

Idea: it does bad when the intrinsic geometry of a graph does not look like hyperbolic space....

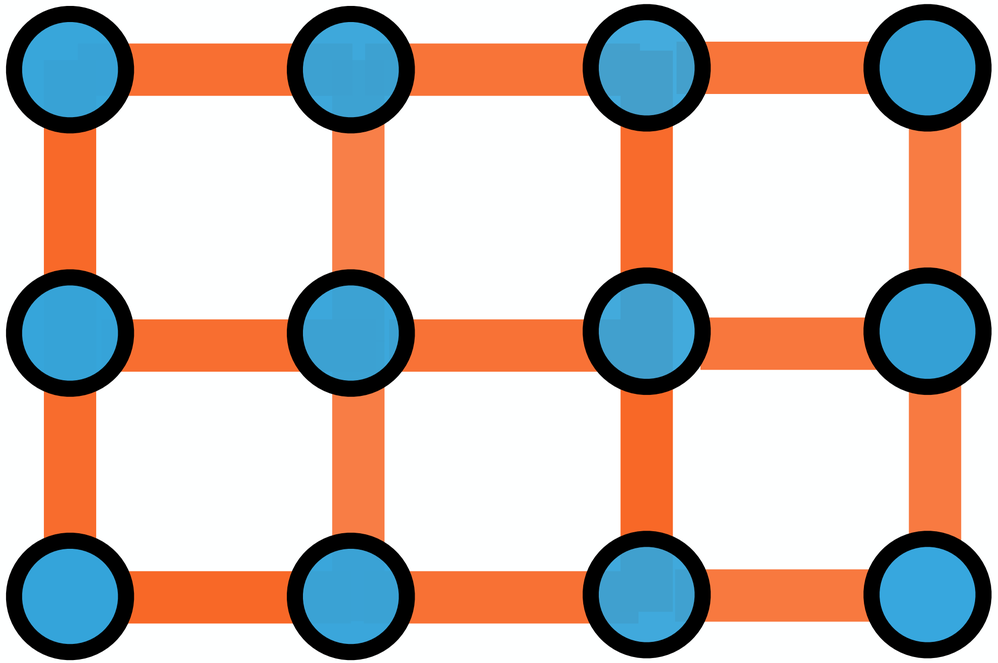
Problem: if you try to run this on real world datasets, it does a bad job

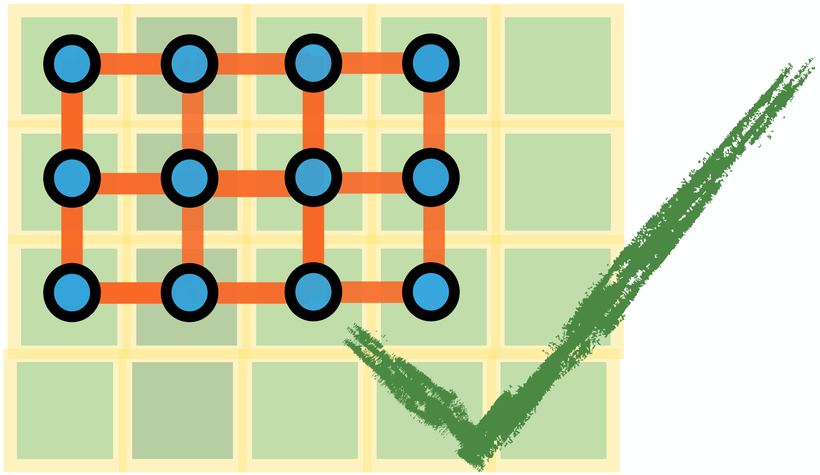
This grid-shaped graph has geometry like the Euclidean plane.

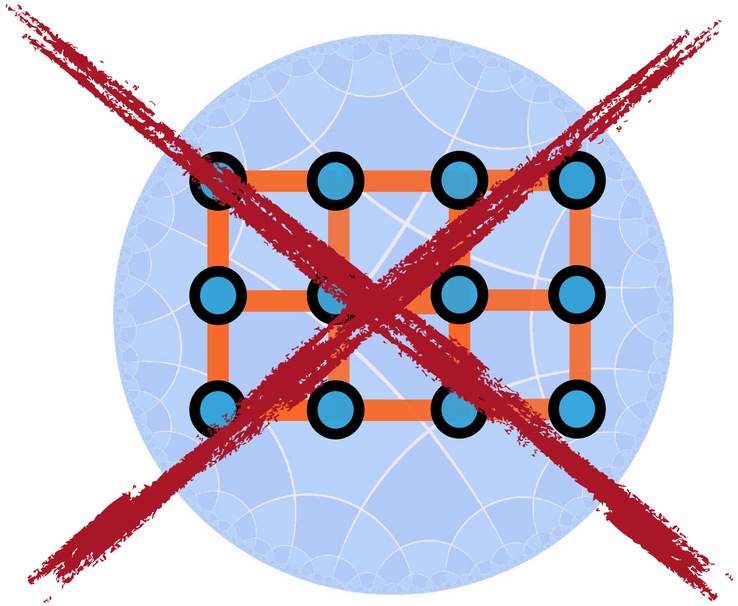
Euclidean space is very different than hyperbolic space.

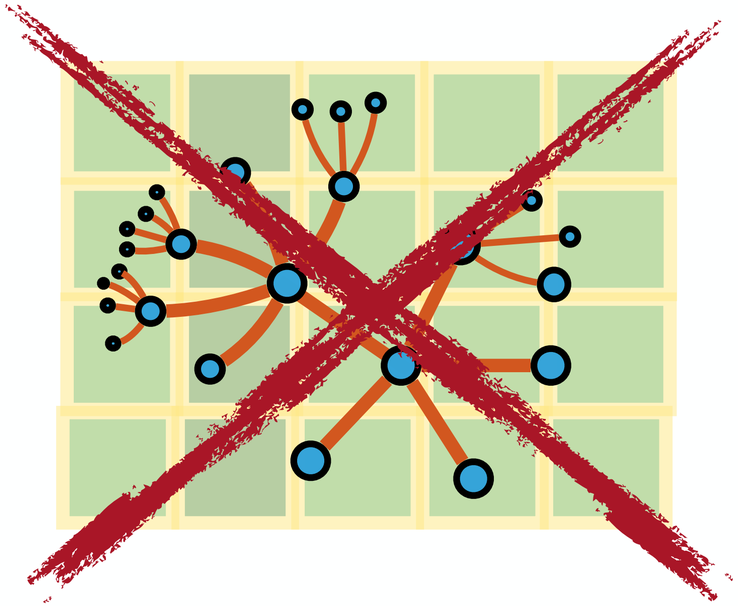
> Balls in the graph grow like r^2. Balls in hyperbolic space grow like e^r

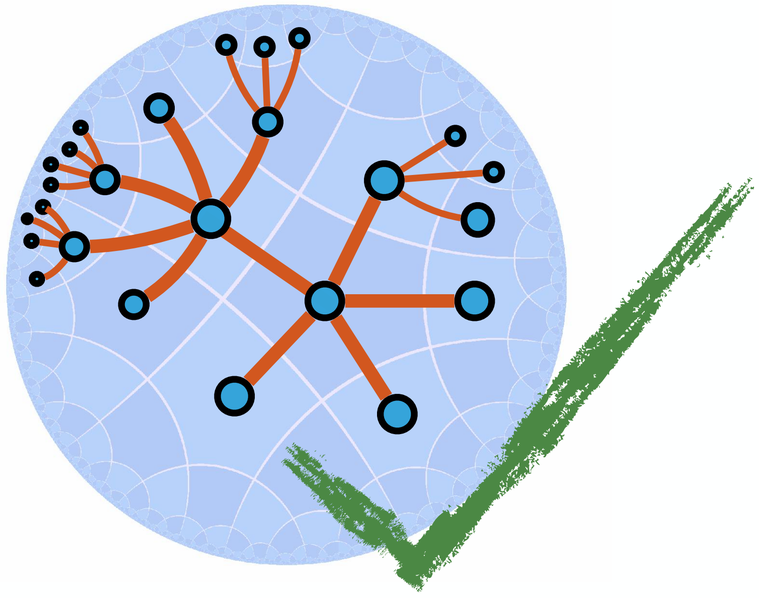
So, this grid won't fit well inside of hyperbolic space.











This is not good. To figure out where to embed our graph we need to know the geometry of our graph.

But we want something for arbitrary, real world data!

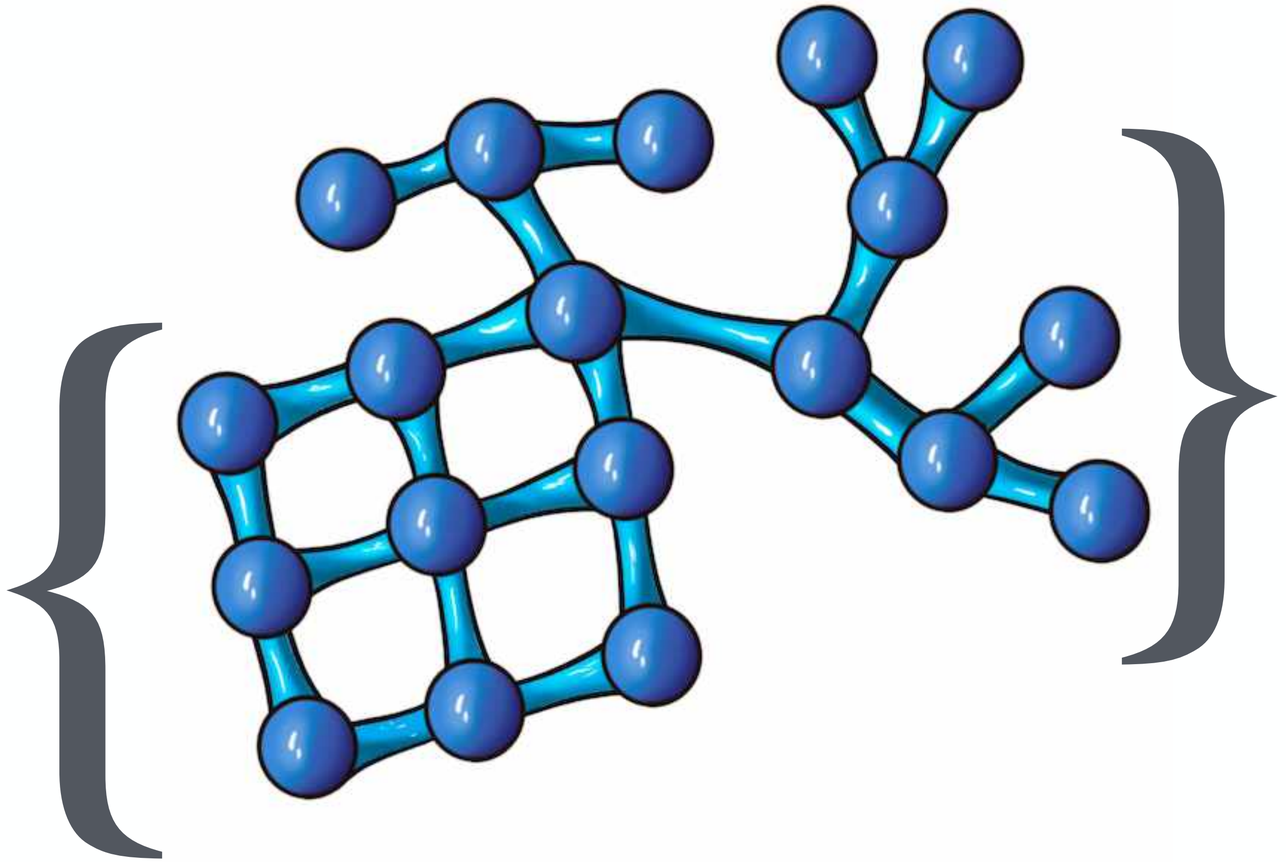
But it gets worse...

Vertices: legal cases in England and Wales.

Edges: Citations

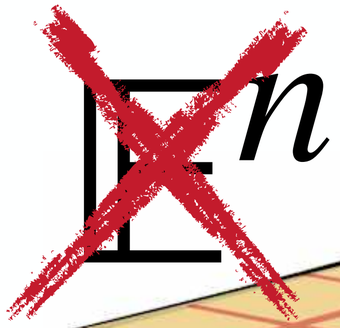
Daniel Hoadely

A small piece of the legal casework graph

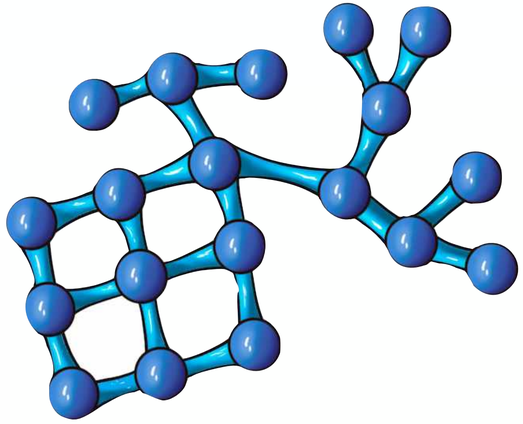


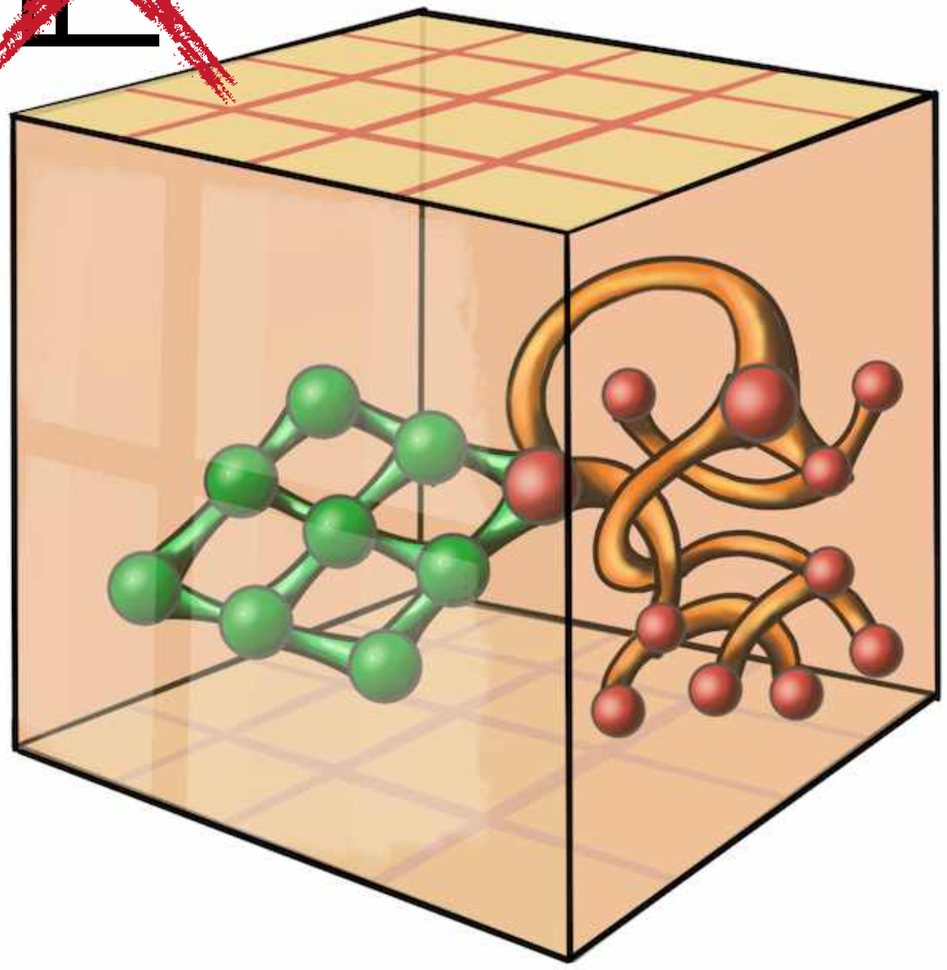
Grid-like components

Tree-like components



This graph is trouble for **BOTH** spaces





Can we build a “**universal embedder**”: a space that let’s us embed real world graphs accurately, even if we don’t know their geometry?

Me

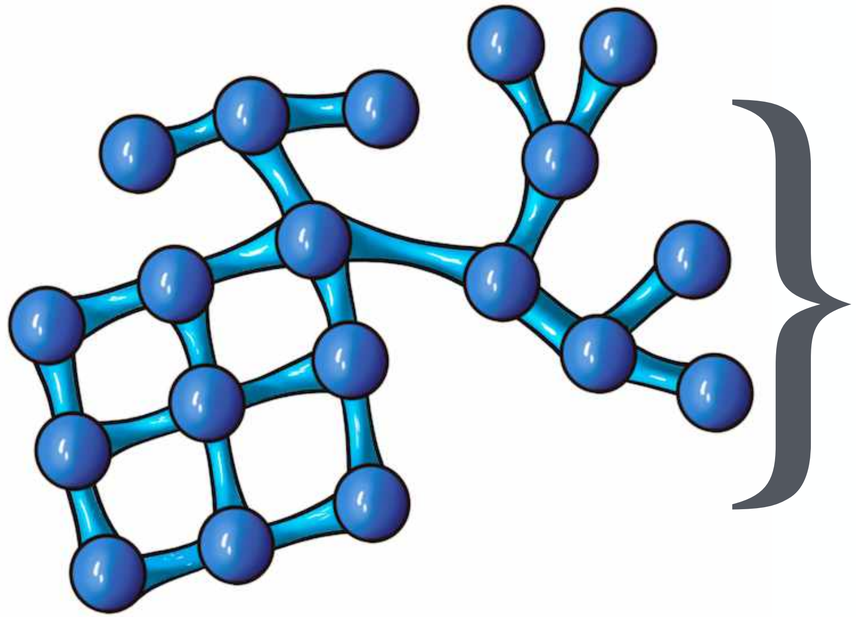
Anna Wienhard

Michel Strube

Fede Lopez

Bea Pozzetti

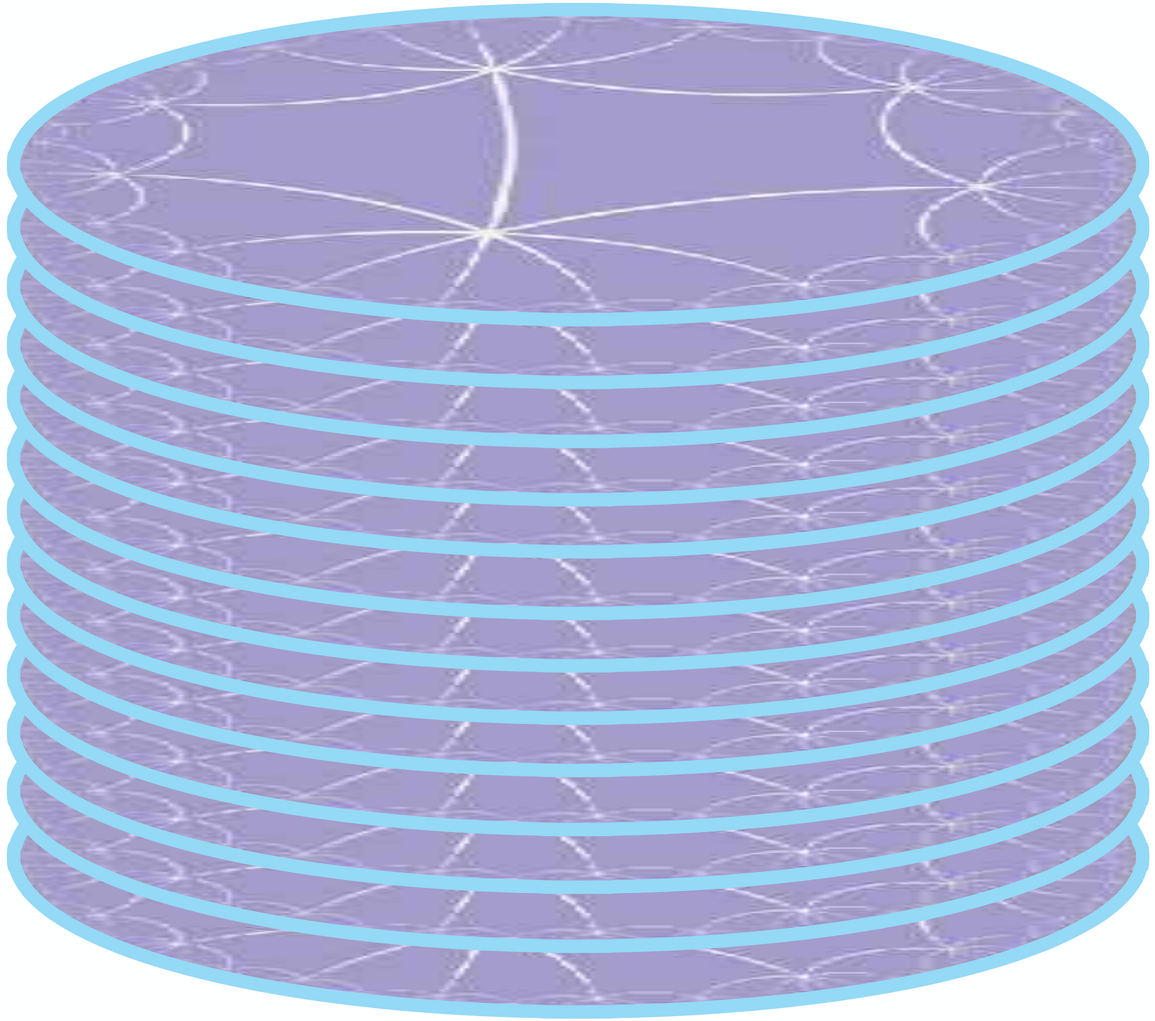
Need flat spots



Need hyperbolic spots

$$\mathbb{H}^2 \times \mathbb{R}$$

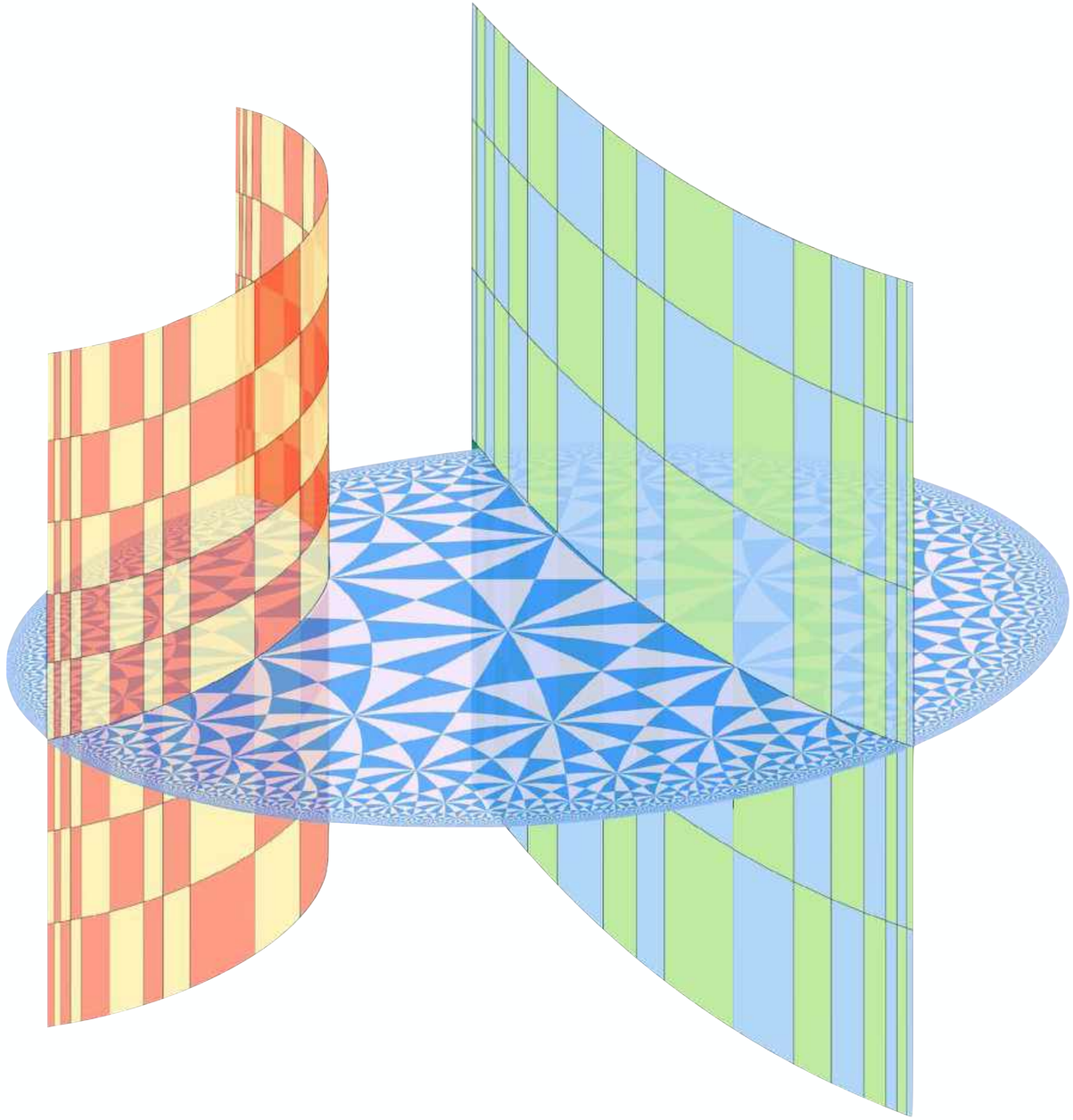
The geometry of
a stack of hyperbolic planes.

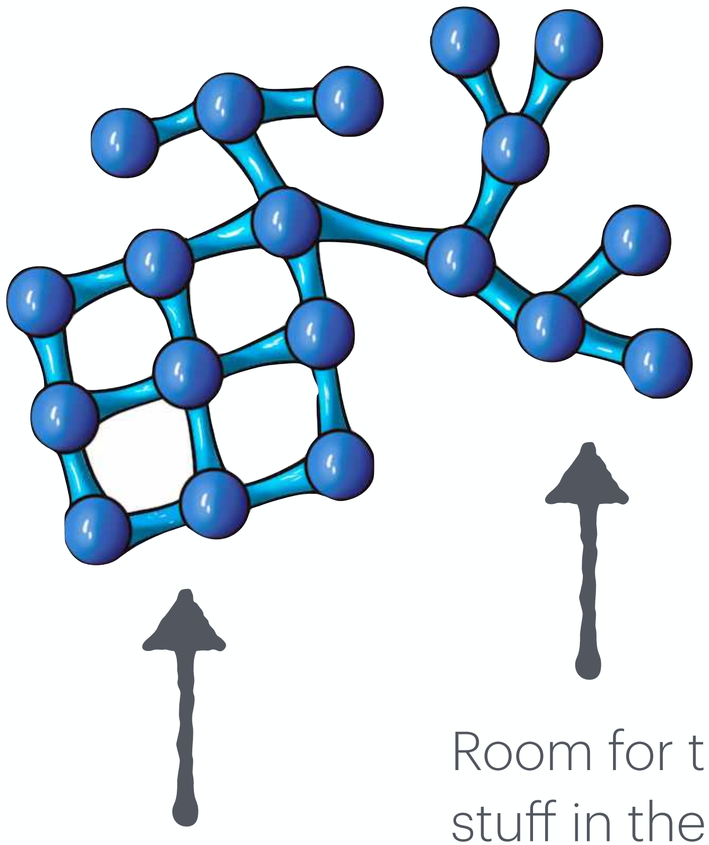


$$\mathbb{H}^2 \times \mathbb{R}$$

Horizontal slices =
Negative Curvature

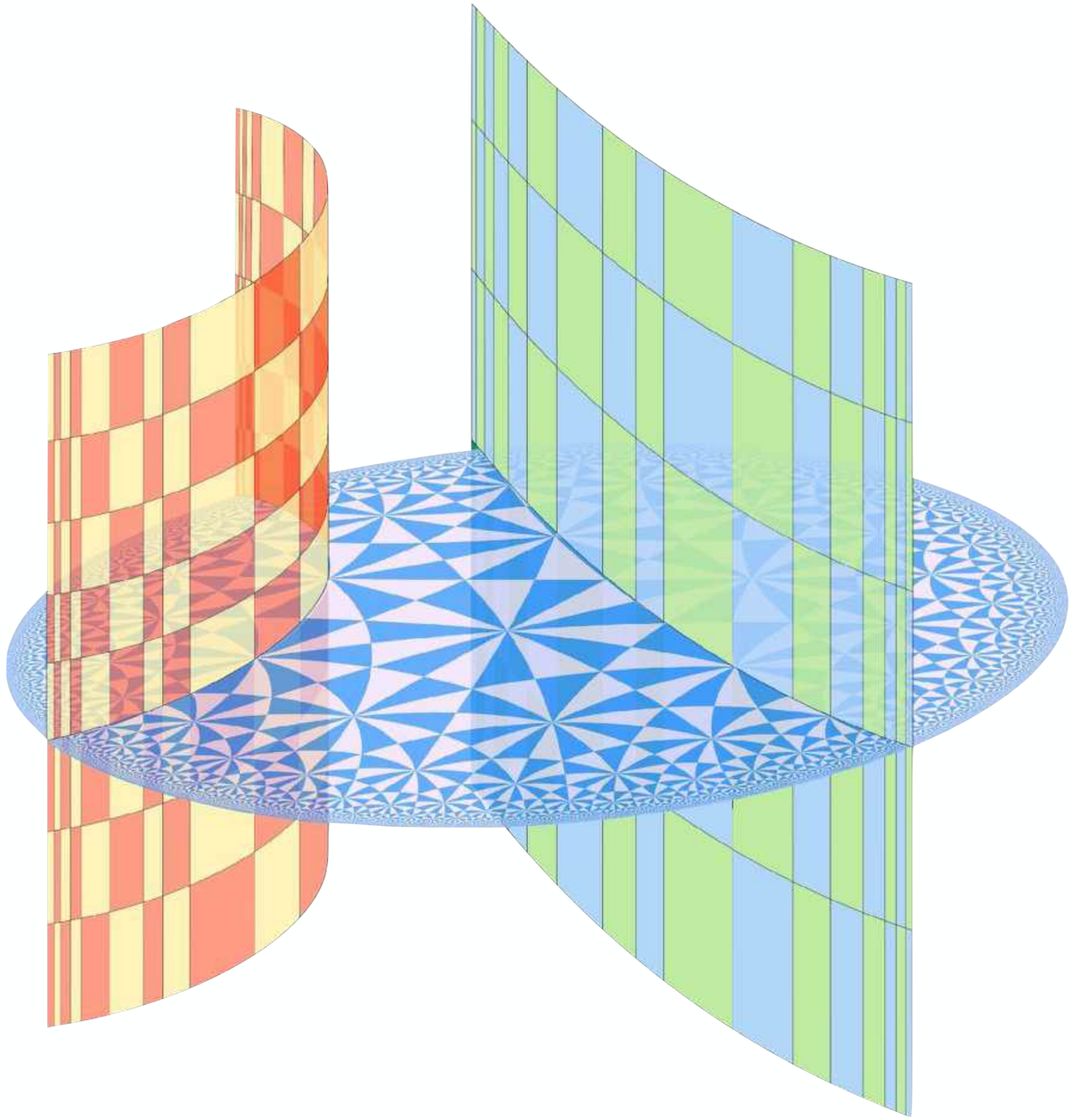
Vertical slices =
Zero Curvature

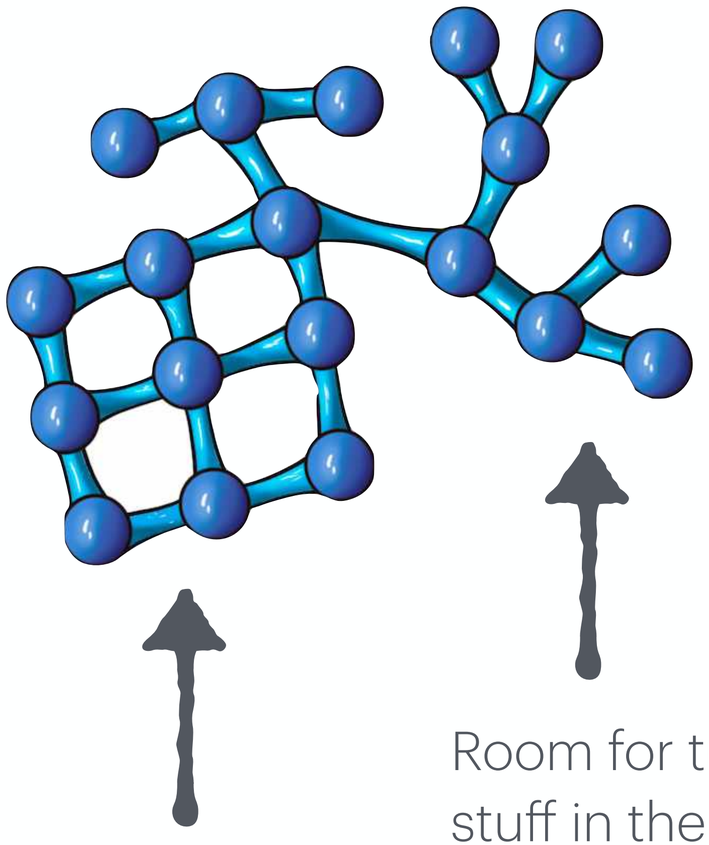




Room for grid-like stuff in the flat parts,

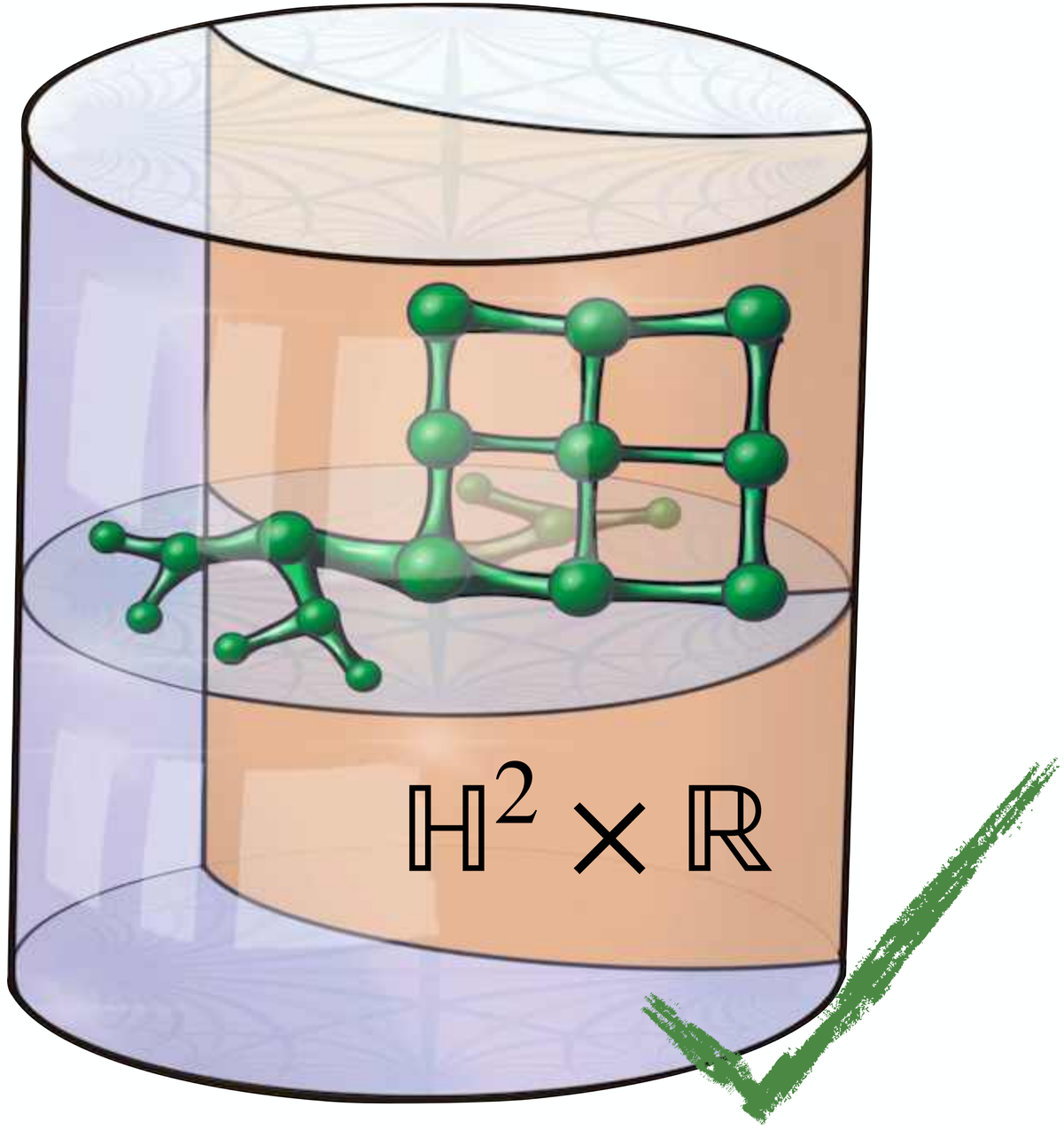
Room for tree-like stuff in the negative curvature

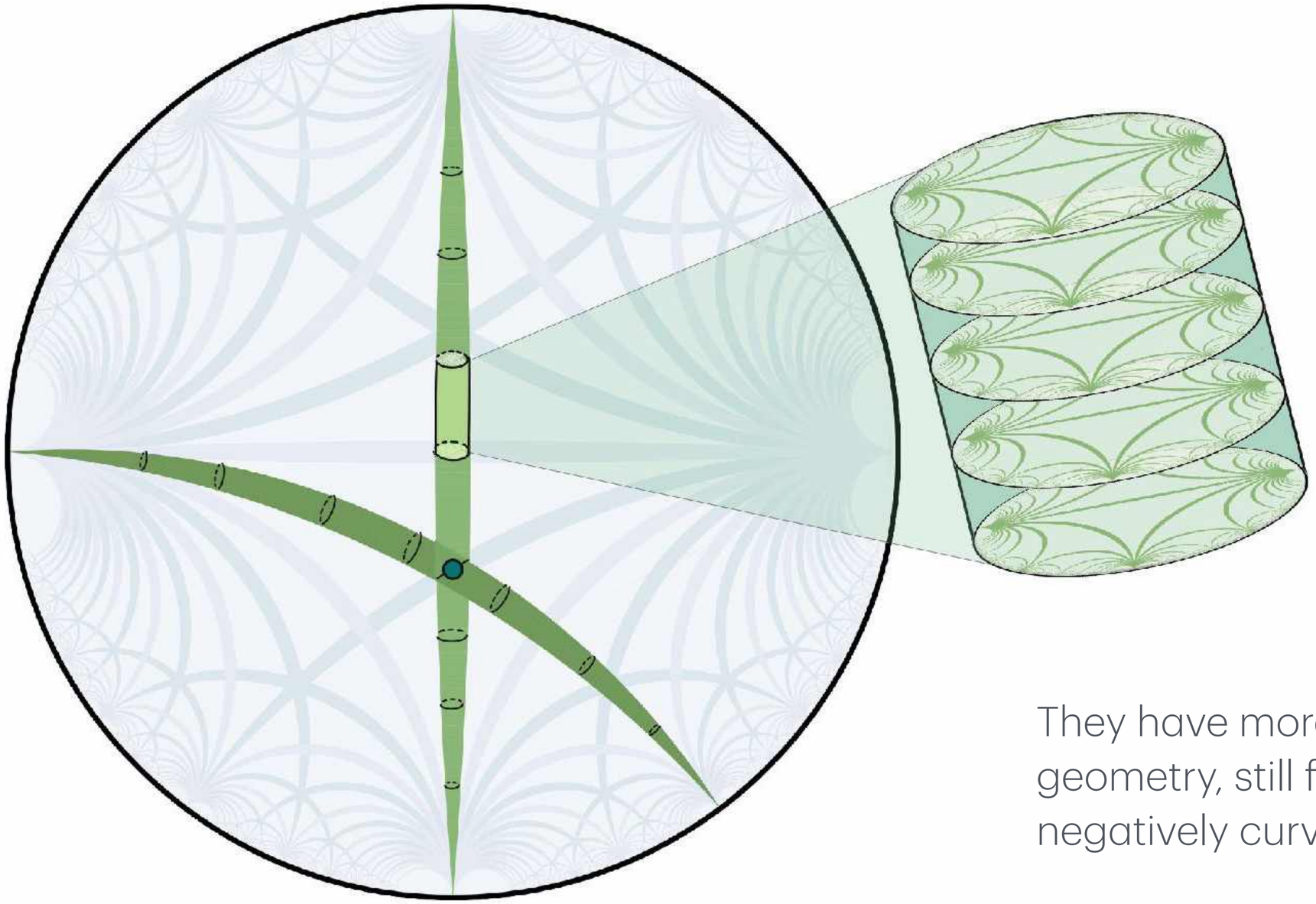




Room for grid-like stuff in the flat parts,

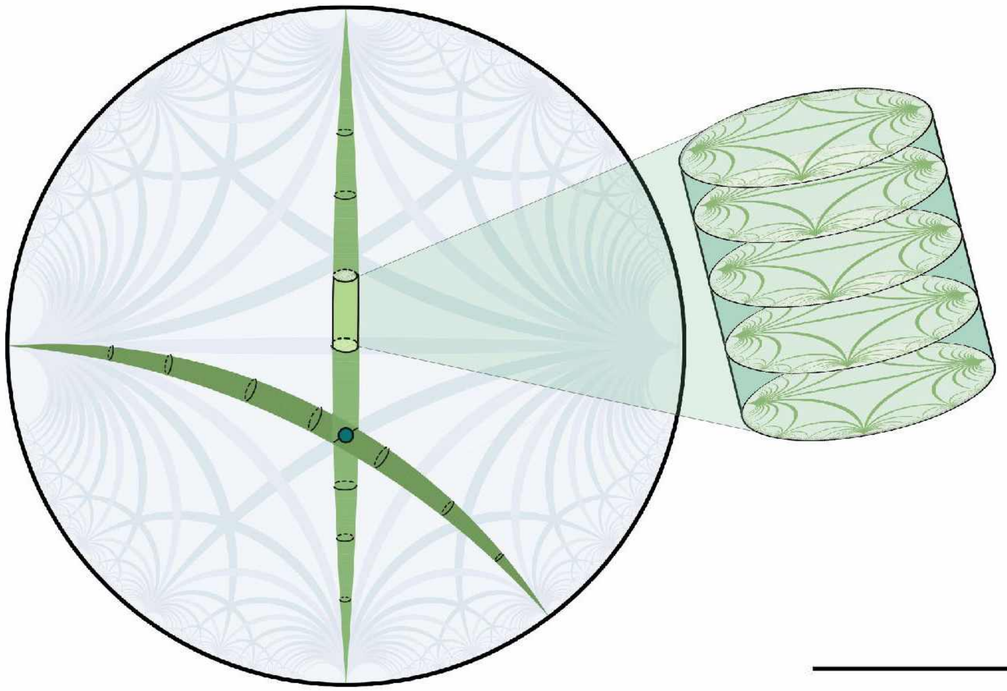
Room for tree-like stuff in the negative curvature





Higher dimensional versions of this are called *Symmetric Spaces:*

They have more complicated geometry, still full of both flat and negatively curved regions.



Higher dimensional versions of this are called *Symmetric Spaces:*

All symmetric spaces were classified in the 1920's by Elie Cartan.

Symmetric Spaces for Graph Embeddings

Type	Non-compact	Compact	$\mathrm{rk}_{\mathbb{R}}$	dim
AI	$\mathrm{SL}(n,\mathbb{R})/\mathrm{SO}(n,\mathbb{R})$	$\mathrm{SU}(n)/\mathrm{SO}(n)$	$n-1$	$\frac{(n-1)(n+2)}{2}$
A	$\mathrm{SL}(n,\mathbb{C})/\mathrm{SU}(2)$	$(\mathrm{SU}(n)\times\mathrm{SU}(n))/\mathrm{SU}(n)$	$n-1$	$(n+1)(n-1)$
BDI	$\mathrm{SO}(p,q)/\mathrm{SO}(p)\times\mathrm{SO}(q)$	$\mathrm{SO}(p+q)/\mathrm{SO}(p)\times\mathrm{SO}(q)$	$\min\{p,q\}$	pq
AIII	$\mathrm{SU}(p,q)/\mathrm{SU}(p)\times\mathrm{SU}(q)$	$\mathrm{SU}(p+q)/\mathrm{SU}(p)\times\mathrm{SU}(q)$	$\min\{p,q\}$	$2pq$
CI	$\mathrm{Sp}(2n,\mathbb{R})/\mathrm{U}(n)$	$\mathrm{Sp}(2n)/\mathrm{U}(n)$	n	$2n(n+1)$
DIII	$\mathrm{SO}^*(2n)/\mathrm{U}(n)$	$\mathrm{SO}(2n)/\mathrm{U}(n)$	$\lfloor\frac{n}{2}\rfloor$	$n(n-1)$
CII	$\mathrm{Sp}(p,q)/\mathrm{Sp}(p)\times\mathrm{Sp}(q)$	$\mathrm{Sp}(p+q)/\mathrm{Sp}(p)\times\mathrm{Sp}(q)$	$\min\{p,q\}$	$4pq$
AII	$\mathrm{SL}(n,\mathbb{H})/\mathrm{Sp}(n)$	$\mathrm{SU}(2n)/\mathrm{Sp}(n)$	$n-1$	$(n-1)(2n+1)$
D	$\mathrm{SO}(2n,\mathbb{C})/\mathrm{SO}(2n)$	$(\mathrm{SO}(2n)\times\mathrm{SO}(2n))/\mathrm{SO}(2n)$	n	$n(2n-1)$
B	$\mathrm{SO}(2n+1,\mathbb{C})/\mathrm{SO}(2n+1)$	$(\mathrm{SO}(2n+1)\times\mathrm{SO}(2n+1))/\mathrm{SO}(2n+1)$	n	$n(2n+1)$
C	$\mathrm{Sp}(n,\mathbb{C})/\mathrm{Sp}(n)$	$(\mathrm{Sp}(n)\times\mathrm{Sp}(n))/\mathrm{Sp}(n)$	n	

Table 6. The classical symmetric spaces. Row CI represents the Siegel spaces and their compact duals.

Proposal: In the machine learning architecture for graph embedding problems, replace *Linear Algebra* with the *Geometry of Symmetric Space*

Vector-valued Distance and Gyrocalculus on the Space of Symmetric Positive Definite Matrices

Federico López*
HITS

Steve Trettel
Stanford University

Michael Strube
HITS

We propose the use o
geometric informati
(SPD), and develop
erations in this curv
versatility in the tas
question answerin
in Euclidean and h
ize embeddings,
positive samples

1 Introduction

Symmetric Positive De
such as pedestrian dete
and image set classific
others. They have be
[78]), while respectin
convenient trade-off
has applied approxi
space [22, 83], or b

These methods fac
other known con
several distances
Stein metric [71]
respective geom

Symmetric Spaces for Graph Embeddings: A Finsler-Riemannian Approach

Federico López [1] Beatrice Pozzetti [2] Steve Trettel [3] Michael Strube [1] Anna Wienhard [2]

Abstract

Learning faithful graph representations as sets of vertex embeddings has become a fundamental intermediary step in a wide range of machine learning applications. We propose the systematic use of symmetric spaces in representation learning, a class encompassing many of the previously used embedding targets. This enables us to introduce a new method, the use of Finsler metrics integrated in a Riemannian optimization scheme, that better adapts to dissimilar structures in the graph. We develop a tool to analyze the embeddings and infer structural properties of the data sets. For implementation, we choose Siegel spaces, a versatile family of symmetric spaces. Our approach outperforms competitive baselines for graph reconstruction tasks on various synthetic and real-world datasets. We further demonstrate its applicability on two downstream tasks, recommender systems

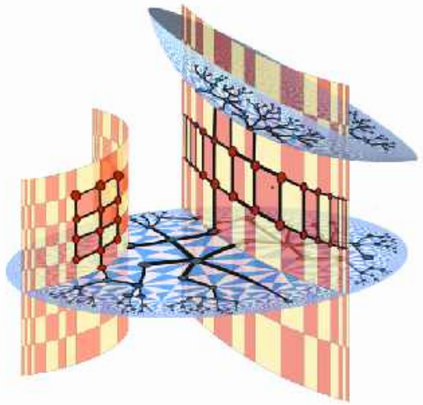


Figure 1. Symmetric spaces have a rich structure of totally geodesic subspaces, including flat subspaces (orange) and hyper-
bolic planes (blue). This compound, yet computationally tractable

Modeling Graphs Beyond Hyperbolic: Graph Neural Networks in Symmetric Positive Definite Matrices

Lopez[1], *J. Maxwell Riestenberg[2], Michael Strube[1],
ldin Taha[2], and Steve Trettel[3]
itute for Theoretical Studies, Germany
me.lastname}@h-its.org
erg University, Germany
ha}@mathi.uni-heidelberg.de
of San Francisco, USA
el@usfca.edu

n that alignment between the struc-
of an embedding space is crucial
of the data. The uniform geom-
allows for representing graphs
eatures, such as grids and hi-
r, real-world graph data is
and topological features,
mbedding spaces. In this
ce of symmetric positive
ral networks that can
levelop an innovative
8] to implement the
ks in SPD. Exper-
networks in SPD
and hyperbolic
plex graphs fo
and d

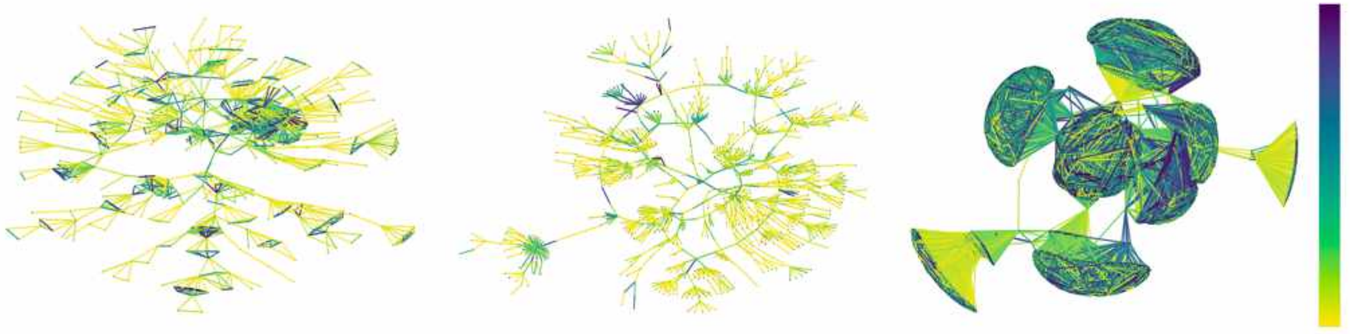


Figure 5. Edge coloring of $\mathcal{S}_2^{F1}$ for BIO-DISEASOME (left) and CSPHD (center) and FACEBOOK (right). Edge colors indicate the angle of the vector-valued distance for each edge, on a linear scale from 0 (yellow) to $\pi/4$ (blue).

Table 3: Results for Question Answering.

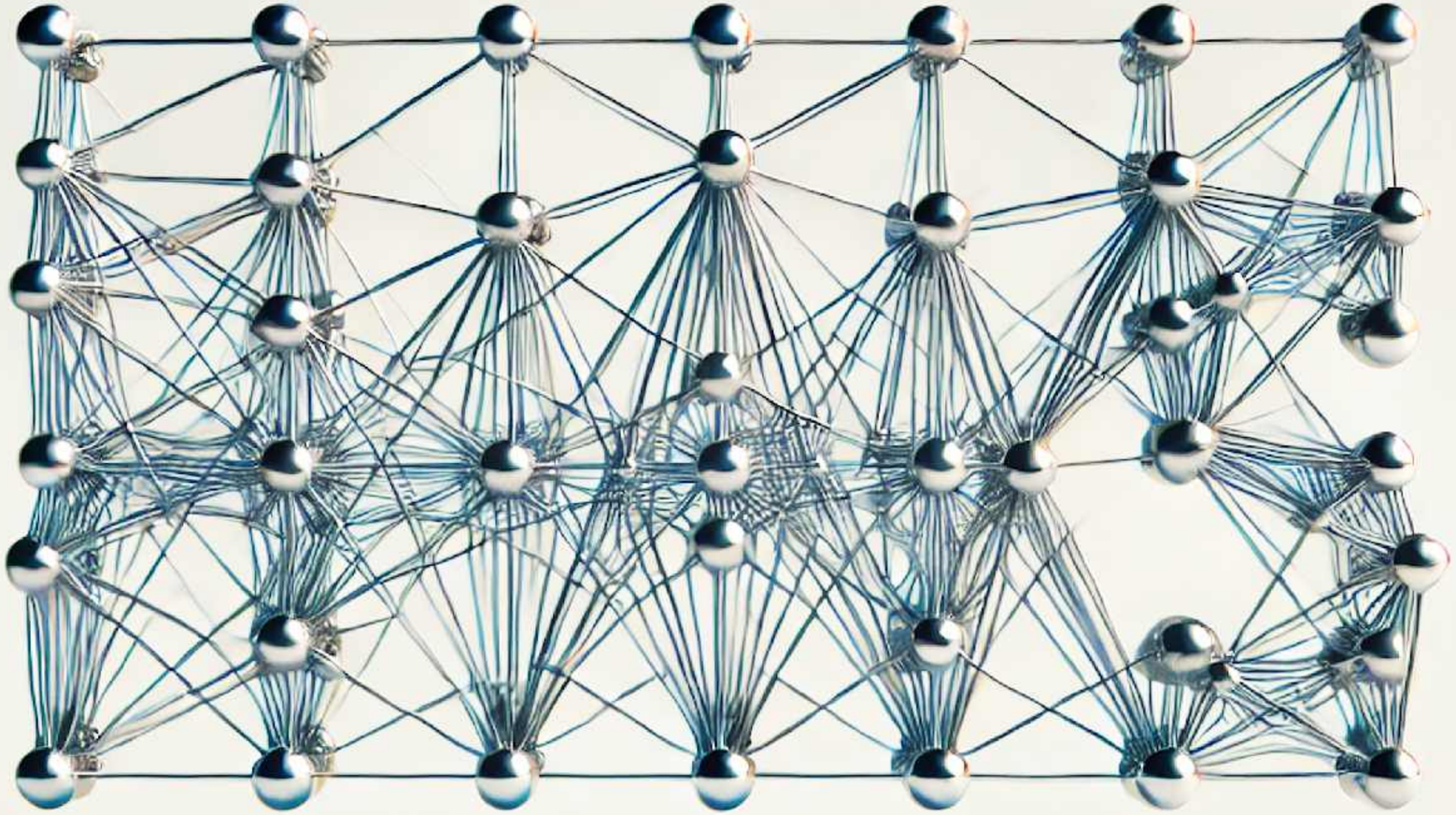
Model	TRECQA MRR	TRECQA H@1	WIKIQA MRR	WIKIQA H@1
Euclidean	55.9±2.0	41.0±2.0	43.4±0.3	22.4±1.1
Hyperbolic	58.0±1.3	39.3±2.0	44.0±0.4	22.8±0.6
$\text{SPD}^{R}_{\text{Sca}}$	55.4±0.1	37.1±0.1	**45.5±0.5**	24.4±1.1
$\text{SPD}^{F_1}_{\text{Sca}}$	57.1±0.7	38.6±0.2	44.8±0.5	24.0±0.6
$\text{SPD}^{R}_{\text{Rot}}$	58.7±1.5	41.4±2.9	44.6±0.6	23.6±0.6
$\text{SPD}^{F_1}_{\text{Rot}}$	58.1±0.5	**43.6±1.0**	43.7±0.4	23.8±0.8
$\text{SPD}^{R}_{\text{Ref}}$	57.3±0.3	40.7±1.1	43.9±0.7	23.4±2.0
$\text{SPD}^{F_1}_{\text{Ref}}$	**59.6±0.5**	42.1±1.0	44.7±1.2	**25.0±2.5**

Table 2: Results for Knowledge graph-based recommender systems.

Model	SOFTWARE MRR	SOFTWARE H@10	LUXURY MRR	LUXURY H@10	PANTRY MRR	PANTRY H@10	MINDREADER MRR	MINDREADER H@10
TRANSE	28.5±0.1	47.2±0.5	35.6±0.1	52.3±0.1	16.6±0.0	35.3±0.1	19.1±0.4	37.6±0.1
ROTC	28.5±0.3	45.4±1.4	33.0±0.1	49.8±0.2	14.5±0.0	31.3±0.2	25.3±0.3	50.3±0.6
MURE	29.4±0.4	47.1±0.4	35.6±0.7	54.0±0.3	19.4±0.1	39.5±0.2	25.2±0.3	49.9±0.6
MURP	29.6±0.3	47.9±0.3	**37.5±0.1**	**55.2±0.3**	19.4±0.1	39.8±0.2	25.3±0.3	49.3±0.2
$\text{SPD}^{R}_{\text{Sca}}$	29.4±0.4	48.1±0.8	**37.5±0.2**	55.1±0.2	19.5±0.0	39.6±0.3	25.4±0.1	49.8±0.3
$\text{SPD}^{F_1}_{\text{Sca}}$	28.8±0.1	46.9±0.5	37.3±0.3	54.1±0.9	19.0±0.1	38.8±0.2	**25.7±0.5**	49.5±0.1
$\text{SPD}^{R}_{\text{Rot}}$	**30.3±0.2**	48.6±0.9	37.2±0.1	54.8±0.4	**20.0±0.1**	**40.3±0.1**	25.3±0.0	**50.5±0.3**
$\text{SPD}^{F_1}_{\text{Rot}}$	30.1±0.1	**49.1±0.3**	36.9±0.1	54.5±0.6	19.2±0.0	39.3±0.1	**25.7±0.0**	49.5±0.2
$\text{SPD}^{R}_{\text{Ref}}$	29.6±0.2	48.0±0.5	37.3±0.2	55.0±0.2	19.3±0.0	39.7±0.3	25.3±0.0	49.1±0.1
$\text{SPD}^{F_1}_{\text{Ref}}$	29.3±0.1	47.5±0.6	36.8±0.0	54.8±0.1	18.6±0.2	38.3±0.3	24.8±0.2	47.9±1.8

Table 2. Comparison of different classifiers on Citeseer (top) and Cora (bottom). We bold the best accuracy in each row.

	$\mathbb{R}^6$	SPD_3		
	LIN-XE	LIN-XE	SVM-MM	NC-MM
GIN	48.2 ± 6.3	**68.0 ± 1.3**	67.3 ± 1.2	67.0 ± 0.8
SGC	62.6 ± 3.4	69.4 ± 1.0	**69.7 ± 0.8**	67.9 ± 1.5
Cheb	63.2 ± 2.1	54.6 ± 10.4	61.4 ± 4.4	**64.0 ± 2.3**
GAT	55.0 ± 5.2	67.3 ± 1.7	**69.2 ± 0.7**	68.1 ± 1.1
GCN	64.7 ± 2.3	**69.9 ± 0.8**	69.2 ± 0.8	68.2 ± 1.0
GIN	77.1 ± 1.0	**79.9 ± 0.6**	79.5 ± 0.6	78.8 ± 1.0
SGC	75.7 ± 3.6	81.5 ± 0.9	**81.8 ± 0.3**	81.1 ± 0.6
Cheb	71.9 ± 2.8	75.5 ± 3.9	77.9 ± 2.4	**79.2 ± 1.3**
GAT	67.9 ± 4.2	79.4 ± 0.9	**81.2 ± 1.4**	**81.2 ± 1.1**
GCN	78.1 ± 1.7	79.7 ± 0.9	**80.7 ± 0.5**	80.2 ± 1.3



THANKS