SIGMA 2024 Workshop

Berlin Mathematics Research Center MATH+

Manifold GCN: Diffusion-based Convolutional Neural Network for Manifold-valued Graphs

Martin Hanik • Gabriele Steidl • **Christoph von Tycowicz**

ZIB ZUSE INSTITUTE BERLIN

October 29, 2024

Learning as Curve Fitting

Typical supervised learning problem: for

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \quad x_i \in \mathcal{X}, \quad y_i \in \mathcal{Y},$$

approximate (or learn) the generating function $F : \mathcal{X} \to \mathcal{Y}$ in some parameterized function class

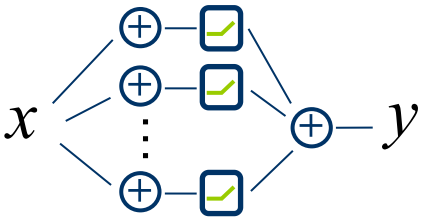
$$\mathcal{F} = \{F_\theta : \theta \in \Theta\}$$

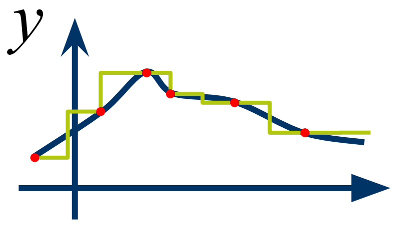
- Data space $\mathcal{X}$ usually high dimensional, e.g., for gray images:

$$\mathcal{X} = \{f : \mathbb{Z}^2 \supset R \to \{0, \cdots, 255\}\}$$

Why should we care about Data Structure in Deep Learning?

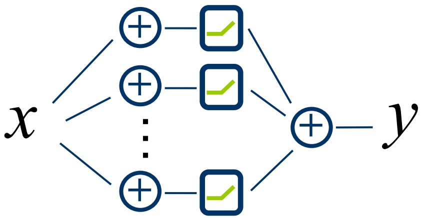
- A multilayer perceptron (MLP) can already approximate any continuous function to an arbitrary precision

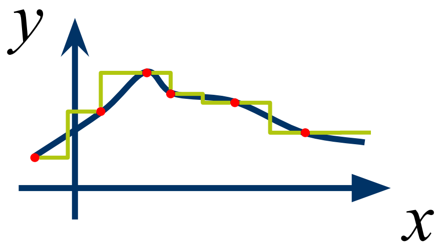




Why should we care about Data Structure in Deep Learning?

- A multilayer perceptron (MLP) can already approximate any continuous function to an arbitrary precision





- Curse of dimensionality makes this result useless in most cases

- Last decade: By utilizing the underlying structure of the problem in deep neural networks, the resulting function space $\mathcal{F}$ is manouvrable but still contains very good approximations of F

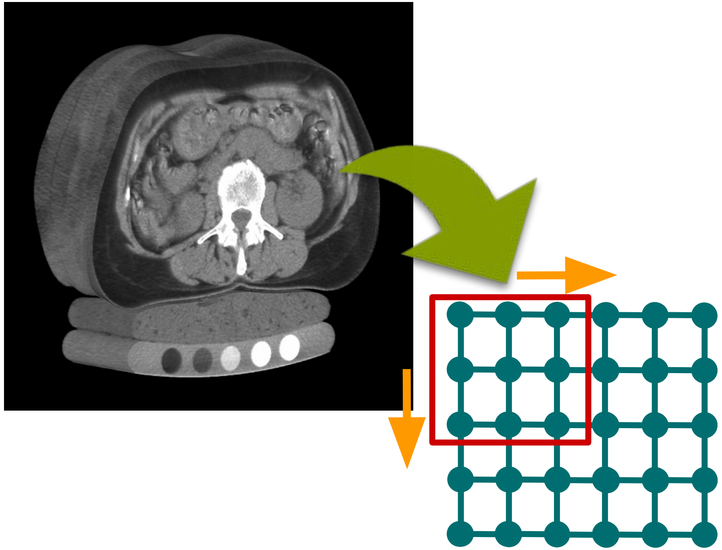
Images and Shift-invariance

One Key Observation for Images

- F is usually invariant under shifts of the domain

Idea lead to convolutional networks

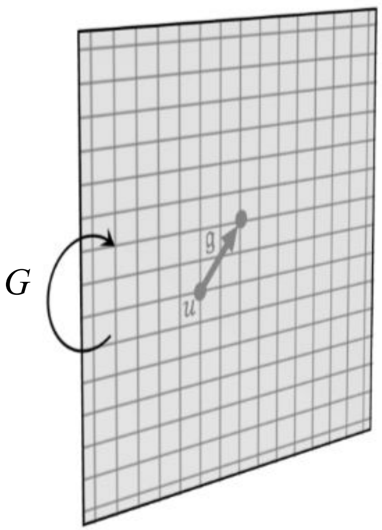
- Convolution (layer) shift-equivariant
- Final invariant (pooling) layer then makes the network output invariant to shifts
- Restricts dimension of $\mathcal{F}$ dramatically



Utilizing the structure (shift-invariance) leads to vastly superior results

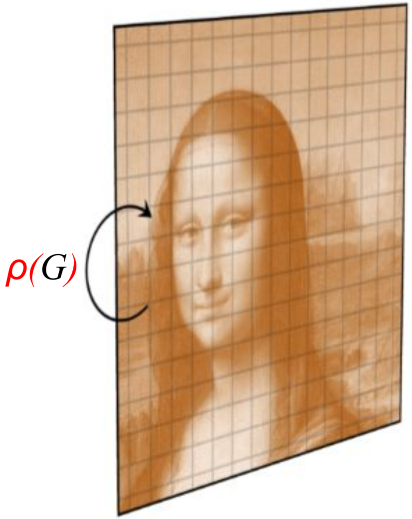
Geometric DL Blueprint

domain Ω



symmetry group G

signals $\mathscr{H}(\Omega)$



group representation $\rho(G)$

$\rho(g)f(u) = f(g^{-1}u)$

functions $\mathscr{F}(\mathscr{H}(\Omega))$

equivariance

$F(\rho(g)f) = \rho(g)F(f)$

invariance

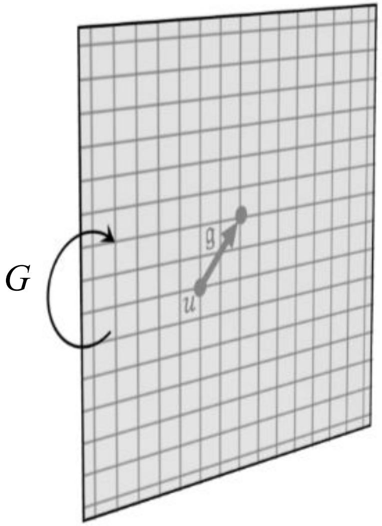
$F(\rho(g)f) = F(f)$

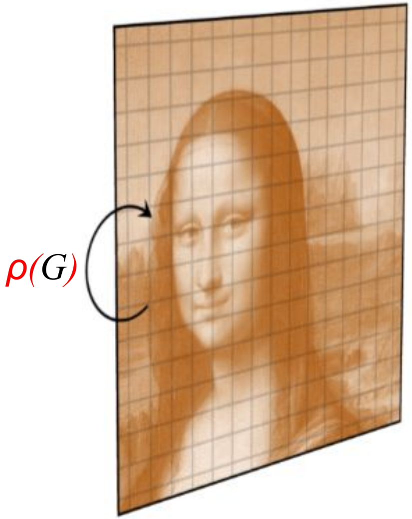
Geometric DL Blueprint (extended)

domain Ω

signals $\mathscr{H}(\Omega)$

functions $\mathscr{F}(\mathscr{H}(\Omega))$





symmetry group G

group representation $\rho(G)$
data symmetry $r \in R$

equivariance
$F(\rho(g) f r) = \rho(g) F(f) r$

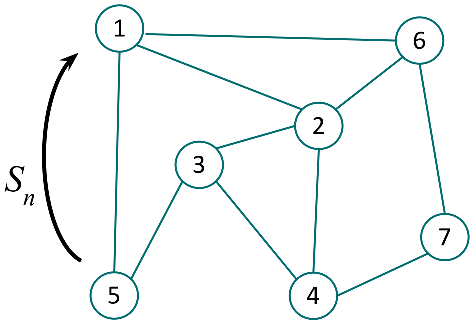
invariance
$F(\rho(g) f r) = F(f)$

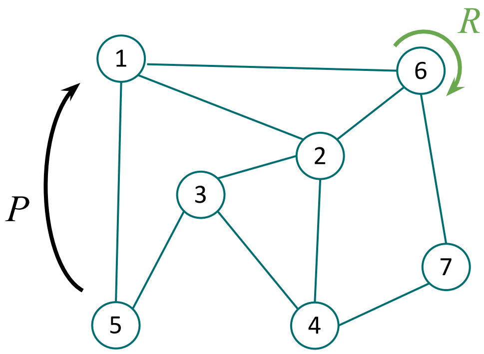
Blueprint: Graph Neural Network

graph $\mathcal{G} = (V, E, w, f)$

node features $\mathscr{H}(\mathcal{G})$

functions $\mathscr{F}(\mathscr{H}(\mathcal{G}))$





Permutation group S_n

Permutation matrix P

Rotation $R \in SO(d)$

Equivariant message passing

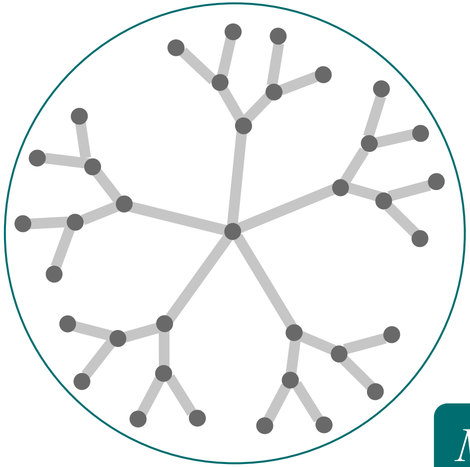
$F(PXR,\ PAP^T) = PF(X,\ A)R$

Manifold-valued Graphs

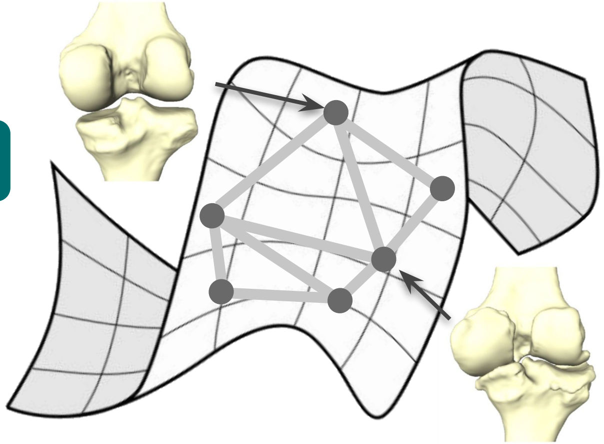
Data space $\mathcal{X}$ of weighted graphs $\mathcal{G} = (V, E, w, f)$, where

$$f : V = \{1, \ldots, n\} \rightarrow M$$

and M is the *feature space*



M… Hyperbolic space



M… Shape space

Convolution

1D case

$$(f * g)(x) = \int_{\mathbb{R}} f(x')g(x - x')dx'$$

Convolution theorem: Fourier transform diagonalizes convolution $\widehat{(f * g)} = \hat{f} \cdot \hat{g}$

$$f * g = \sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(\mathcal{M})}}_{\hat{f}_k} \underbrace{\langle g, \phi_k \rangle_{L^2(\mathcal{M})}}_{\hat{g}_k} \phi_k(x)$$

$\Delta\phi_k = \lambda_k \phi_k$

Localized spectral graph filter[1] ⇨ $\hat{g}_k = \tau(\lambda_k)$

~~Polynomial~~

$\tau(\lambda) = e^{-t\lambda}$ ⇨ heat kernel

1 [Defferrard et al. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. NeurIPS]

Graph Convolutional Filter

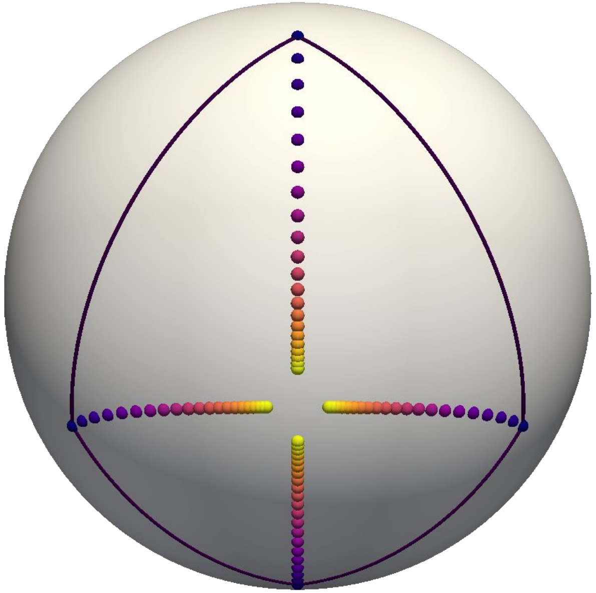
Manifold-valued Laplacian[2] $\Delta_G : \mathcal{H}(V;\mathcal{M}) \to \mathcal{H}(V;T\mathcal{M})$

Diffusion layer discretizing the diffusion equation

$$\frac{d}{dt} f(v,t) = -\Delta_G f(v,t)$$

Diffusion time as a continuous network parameter

- ranging from purely local to totally global
- no need for choosing neighborhood sizes manually



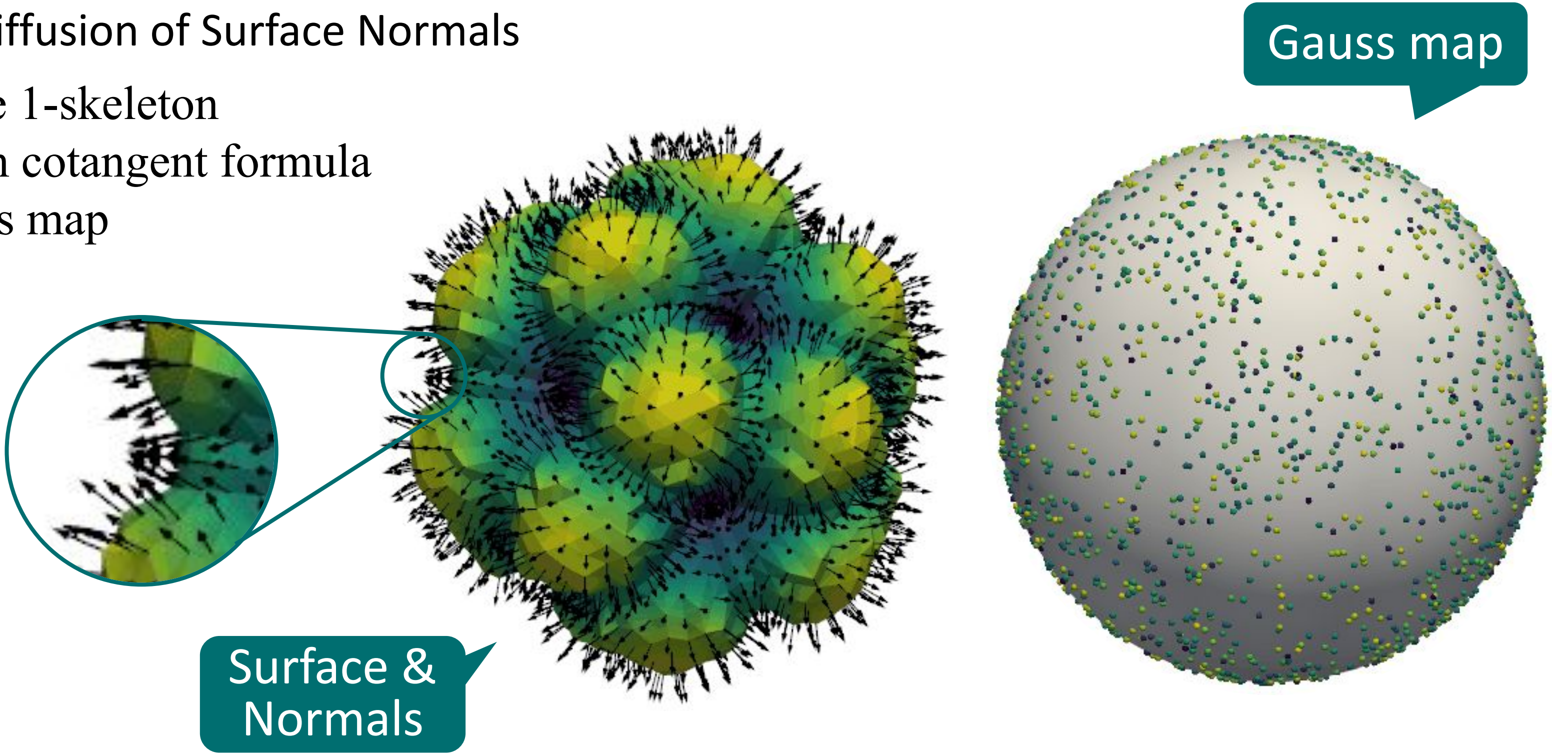
⇨ Invariant under the symmetries of the feature manifold **and** node permutations.

2 [Bergmann & Tenbrinck (2018). A graph framework for manifold-valued data. SIAM J Imaging Sci, 11(1), 325-360.]

Diffusion of Manifold-valued Graphs

Example: Diffusion of Surface Normals

- V, E the 1-skeleton
- w from cotangent formula
- f Gauss map

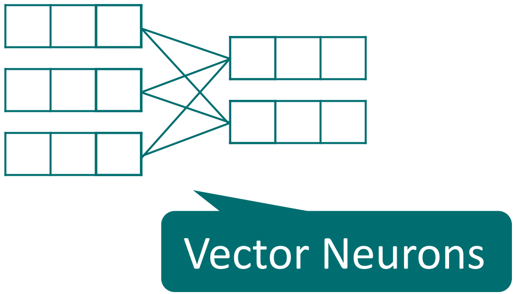


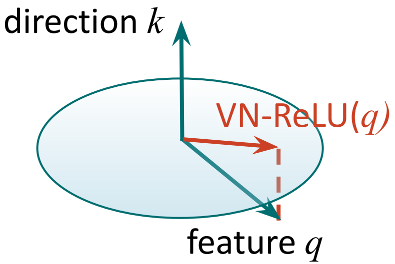
Tangent Multilayer Perceptron

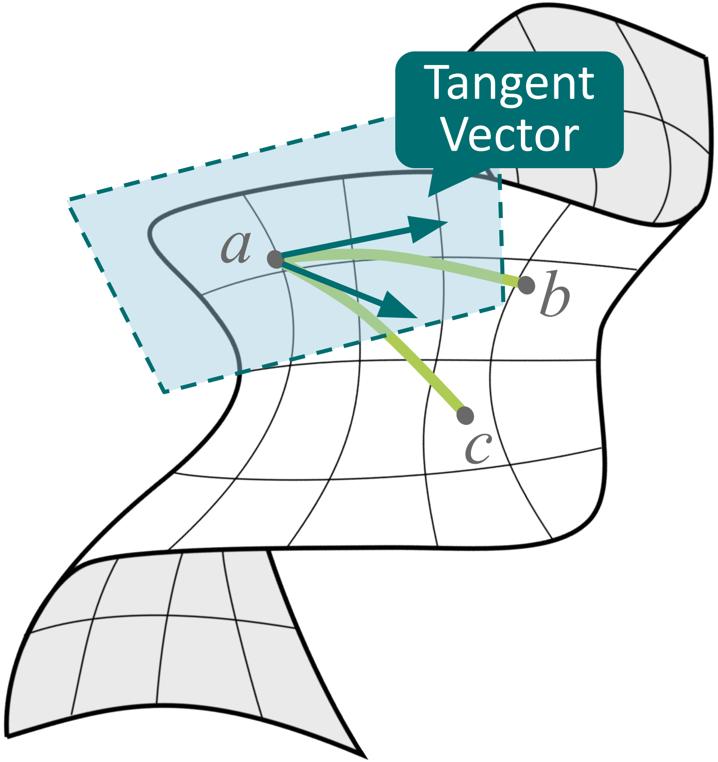
Inverse Riemannian exponential

- Map node features into tangent space
- Isometries of M ⇨ orthogonal change

Allows for equivariant neural units for vector spaces

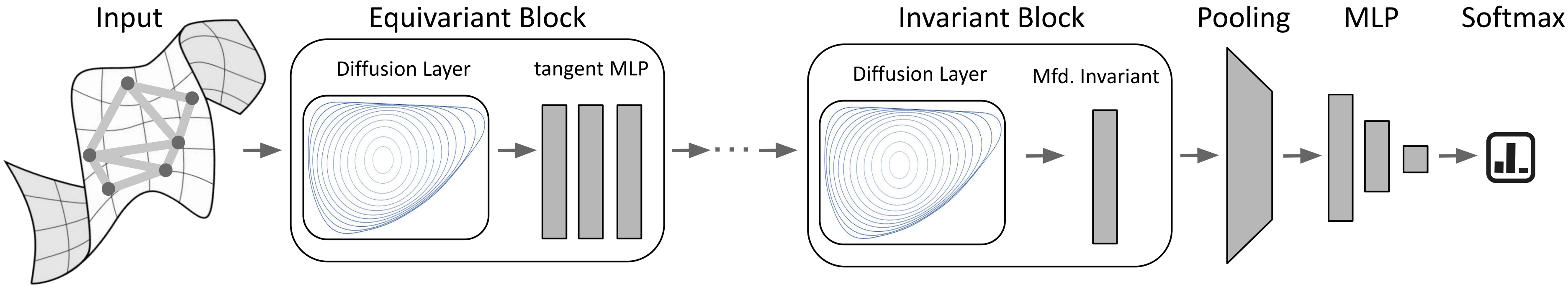






Equivariant Graph Convolutional Network

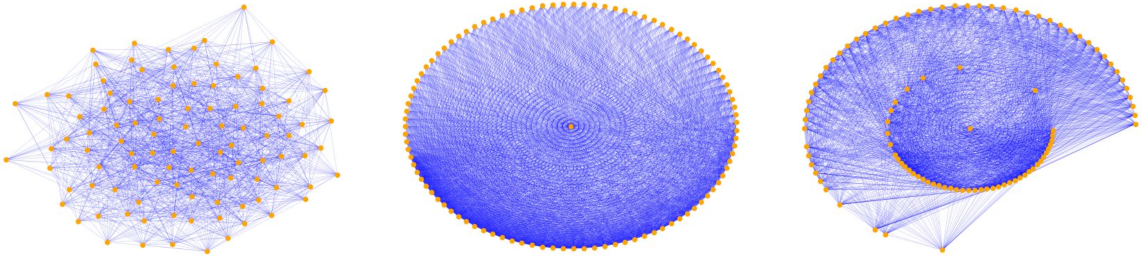
Graph-level classification architecture



Diffusion + node-wise MLP ⇨ expressive function space (incl. radially sym. convolutions[3])

3 [Sharp, N. et al. (2022). Diffusionnet: Discretization agnostic learning on surfaces. ACM Trans. Graph., 41(3), 1-16.]

Classification of Random Graphs



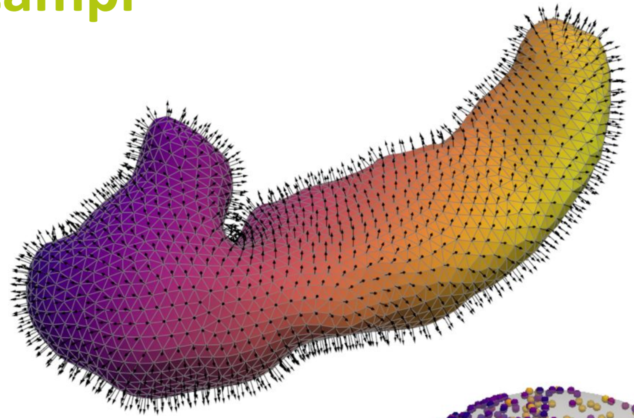
- Benchmark: classify randomly created graphs according to generating algorithm
- Embedding in hyperbolic space is empirically superior to Euclidean space
- Compared to *Hyperbolic Graph Neural Network (HGNN)*[4]

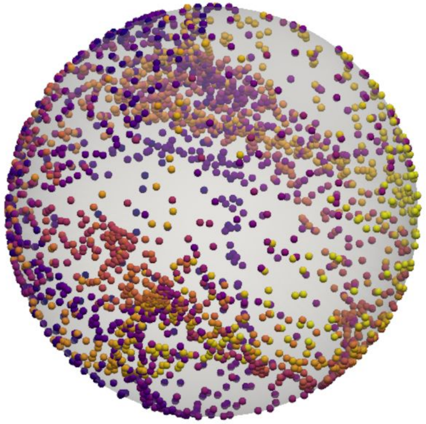
Method	**Mean F1 Score**					**# parameters**
# Graphs	90	180	360	1080	2880	
HGNN (dim=6)	0.518 ± 0.133	0.567 ± 0.085	0.605 ± 0.060	0.654 ± 0.040	0.681 ± 0.042	7053
HGNN (dim=100)	0.418 ± 0.103	0.413 ± 0.105	0.404 ± 0.097	0.547 ± 0.154	0.766 ± 0.079	161903
Ours (degree)	0.594 ± 0.114	0.638 ± 0.084	0.645 ± 0.062	0.679 ± 0.037	0.756 ± 0.046	10429
Ours (one-hot)	$\mathbf{0.633} \pm 0.129$	$\mathbf{0.687} \pm 0.093$	$\mathbf{0.744} \pm 0.059$	$\mathbf{0.787} \pm 0.034$	$\mathbf{0.820} \pm 0.024$	**429**

4 [Liu et al. (2019). Hyperbolic graph neural networks. NeurIPS, 32]

Alzheimer's Classification from Hippocampi

- Shape of hippocampus correlates with progression of Alzheimer's
- Used volume and normals of 120 triangular meshes as shape representation
- Compared against Mesh CNN[5], which is a network that learns from whole meshes
- Mimicked our network closely with Euclidean-only counterparts (GCN)





Method	Mean Accuracy	# parameters
GCN	0.752 ± 0.085	2616
Mesh CNN	0.592 ± 0.073	1320558
Ours (Sphere)	$\mathbf{0.765} \pm 0.075$	**2554**

5 [Hanocka et al. (2019). MeshCNN: a network with an edge. ACM Trans. Graph., 38(4), 1-12.]

Summary

Learning functions in high dimensions

- Cursed estimation problem in general
- Geometric DL gives constructive approach to exploit regularities

Equivariant Graph Neural Network

- Diffusion-based filter for spatial communication
- Tangent MLPs for pointwise nonlinearity

Implications for "Small Data"

- Equi-/Invariance provides essential constraints