PhD thesis defense

EPFL, 2021-12-15

Leveraging topology, geometry, and symmetries for efficient Machine Learning

Michaël Defferrard

EPFL

Prof. Pierre Vandergheynst (EPFL), adviser

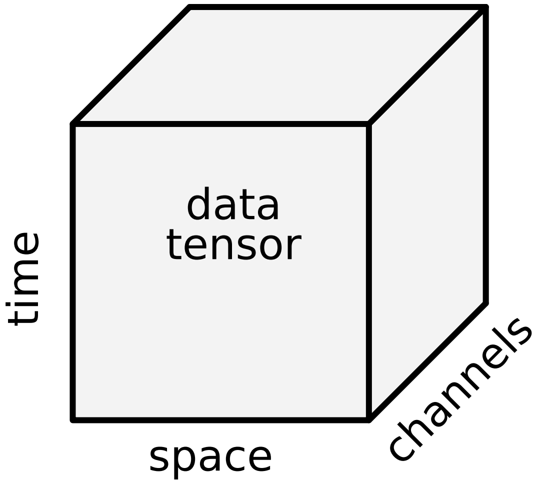
Prof. Pascal Frossard (EPFL), president

Prof. Martin Jaggi (EPFL), examiner

Prof. Max Welling (UvA, MSR), examiner

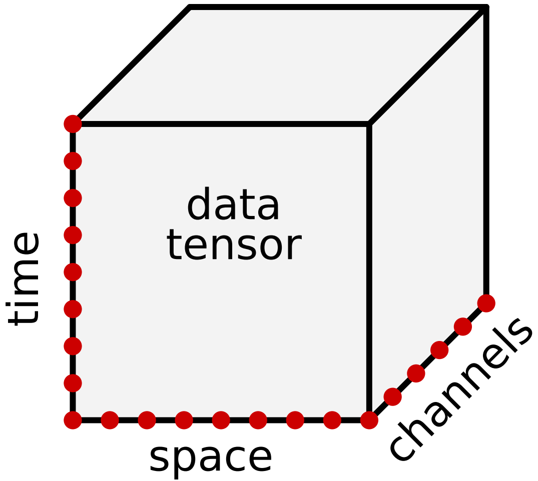
Prof. Yann LeCun (NYU, FAIR), examiner

Structured data



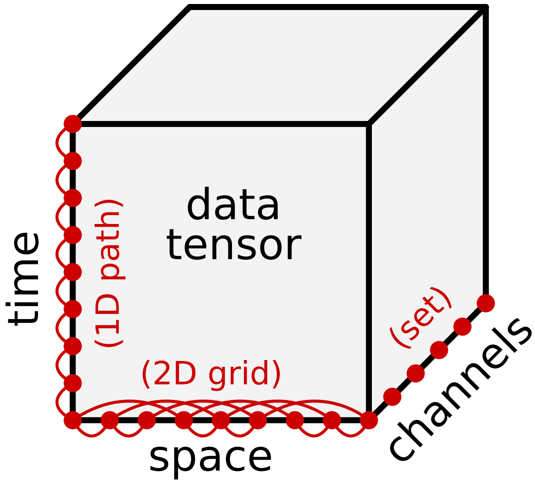
- Data is multi-dimensional.

Structured data



- Data is multi-dimensional.
- Measurements are discrete.

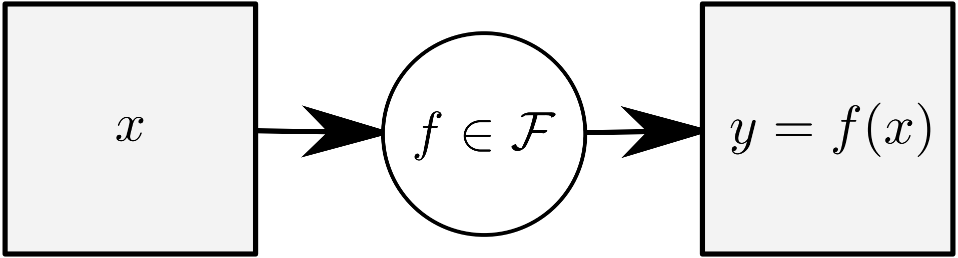
Structured data



- Data is multi-dimensional.
- Measurements are discrete.
- Dimensions are structured.

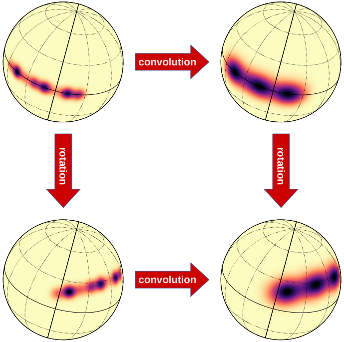
The (deep) learning revolution

From designing the solution f to designing the solution space $\mathcal{F}$.



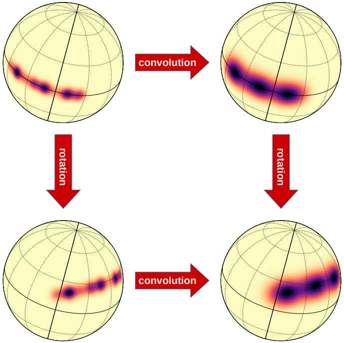
$\mathcal{F}$ is determined by the NN architecture. How to design it?

Design of solution spaces (NN architectures)

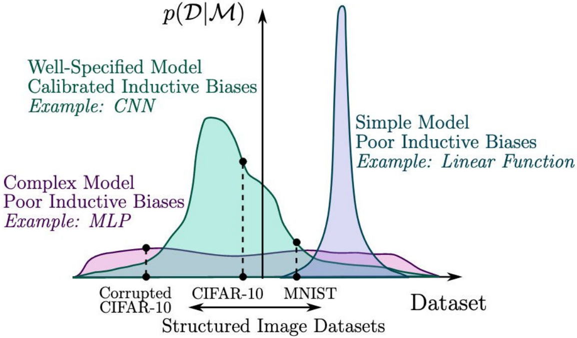


Constraints

Design of solution spaces (NN architectures)



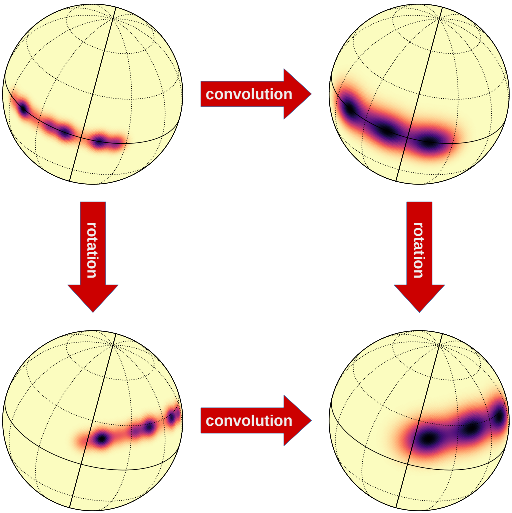
Constraints



Biases

Bias figure from Wilson and Izmailov 2020.

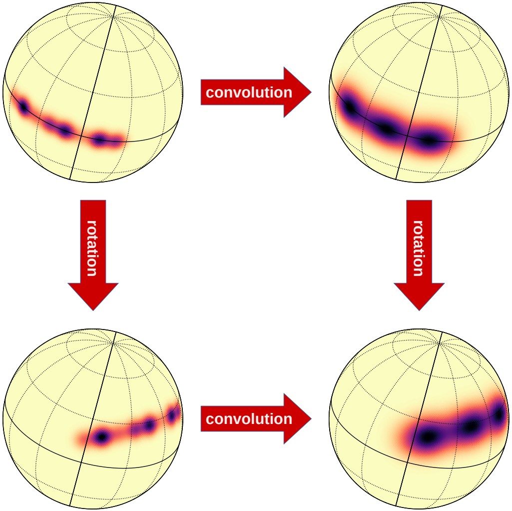
Symmetry constraints



- Equivariance for dense tasks:
  $f(P_\sigma x) = P_\sigma f(x) \quad \forall \sigma \in \mathrm{SO}(3).$
- Invariance for global tasks:
  $f(P_\sigma x) = f(x) \quad \forall \sigma \in \mathrm{SO}(3).$

Why leverage symmetries?

Symmetry constraints



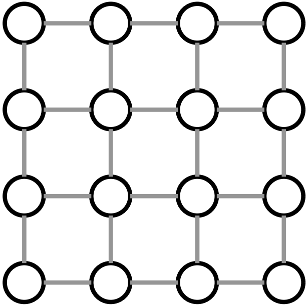
- Equivariance for dense tasks:
  $f(P_\sigma x) = P_\sigma f(x) \quad \forall \sigma \in \mathrm{SO}(3).$
- Invariance for global tasks:
  $f(P_\sigma x) = f(x) \quad \forall \sigma \in \mathrm{SO}(3).$

Why leverage symmetries?

- Data efficiency.
- Generalization guarantee.

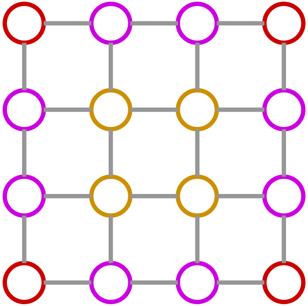
$\Rightarrow$ Principled weight sharing.

Symmetries might not be enough



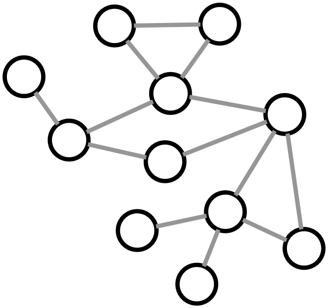
- What are the symmetries? Translations?

Symmetries might not be enough



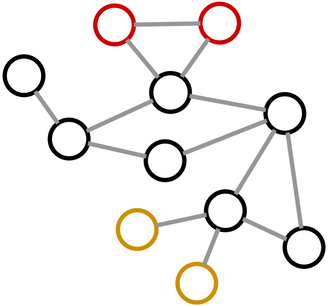
- What are the symmetries? Translations?
- Few symmetries.
- A solution: “cheat” by treating the grid as a discretization of the plane.

Symmetries might not be enough



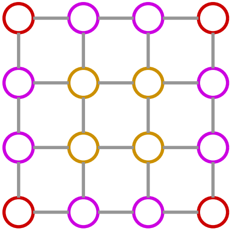
- What are the symmetries?

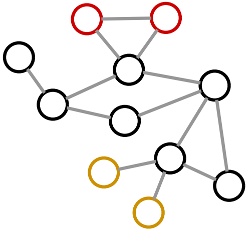
Symmetries might not be enough



- What are the symmetries?
- Asymmetric core with few symmetric motifs.
- Can’t “cheat”. No underlying continuous domain. Purely discrete.

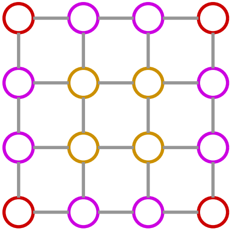
Symmetries might not be enough

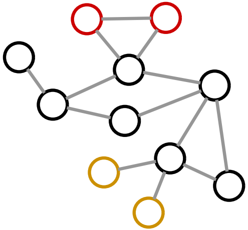




Why more weight sharing?

Symmetries might not be enough

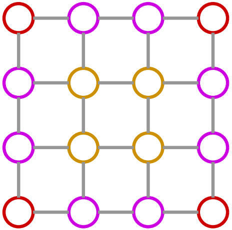


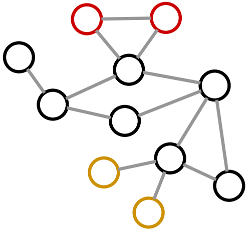


Why more weight sharing?

- Higher data efficiency.
- Stronger generalization guarantee.
- Less powerful / general / flexible.

Symmetries might not be enough





Why more weight sharing?

- Higher data efficiency.
- Stronger generalization guarantee.
- Less powerful / general / flexible.

The bias–variance tradeoff.

Research question

How to leverage the topological and geometrical structure of the data's domain to learn efficiently without the help of symmetry action?

Contributions

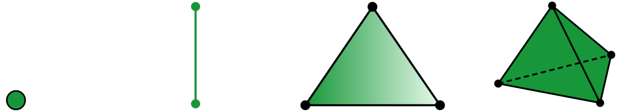
- Transitive and known symmetry groups $\Rightarrow$ group convolutions.
- Non-transitive and/or unknown symmetry groups $\Rightarrow$ generalized convolutions.

My contributions: motivation, construction, analysis, and usage
of generalized convolutions for efficient Machine Learning.

A discrete calculus

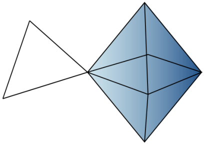
My contributions: motivation, construction, analysis, and usage
of generalized convolutions for efficient Machine Learning.

Space: simplicial complexes



0-simplex 1-simplex 2-simplex 3-simplex

d-simplices.



Simplicial complex K.

- Simplex: set of vertices.
- Simplicial complex K: set of simplices.
  Single axiom: closed under taking subsets.
- K_d: set of all d-simplices.

Data

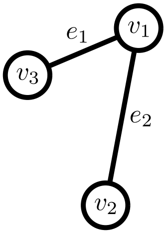
- Simplices naturally form a spatial basis.
- Vertex- ($d = 0$), edge- ($d = 1$), simplex-valued ($d \geq 2$) functions.
- Covariant d-chain $x_d \in \mathbb{R}^{|K_d|}$ and contravariant d-cochain $f_d \in \mathbb{R}^{|K_d|}$.

Duality:

$$\langle x_d, f_d \rangle = {x_d}^\top f_d$$

Topology: an incidence structure

$$K = \{\{v_1\}, \{v_2\}, \{v_3\}, \underbrace{\{v_3, v_1\}}_{e_1}, \underbrace{\{v_1, v_2\}}_{e_2}\}$$



$$B_1 = \begin{pmatrix} +1 & -1 \\ 0 & +1 \\ -1 & 0 \end{pmatrix}$$

- Ordering is arbitrary but necessary.
  $K_0 = \{\{v_1\}, \{v_2\}, \{v_3\}\}$ and $K_1 = \{e_1, e_2\}$.
- Orientation is arbitrary but necessary.
  $e_1 = \{v_3, v_1\}$ and $e_2 = \{v_1, v_2\}$.

Topology: an incidence structure

- Boundary operator ${B_d}^\mathsf{T}$:
  subdomain d-chain $x_d \to$ boundary $(d-1)$-chain ${B_d}^\mathsf{T} x_d$.

- Differential operator[1] B_d:
  data $(d-1)$-cochain $f_{d-1} \to$ finite difference d-cochain $B_d f_{d-1}$.

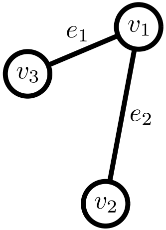
${B_d}^\mathsf{T}$ and B_d are adjoint w.r.t. dual pairing:

$$\left\langle {B_d}^\mathsf{T} x_d, f_{d-1} \right\rangle = \langle x_d, B_d f_{d-1} \rangle \qquad \int_{\partial\Omega} \omega = \int_\Omega \mathrm{d}\omega$$

[1] Also known as the coboundary or exterior derivative.

Geometry: an inner product

$$\langle f_d, h_d \rangle_{M_d} = {f_d}^\top M_d h_d$$



$$M_0 = \begin{pmatrix} \text{weight}(v_1) & 0 & 0 \\ 0 & \text{weight}(v_2) & 0 \\ 0 & 0 & \text{weight}(v_3) \end{pmatrix}$$

$$M_1 = \begin{pmatrix} \text{weight}(e_1) & 0 \\ 0 & \text{weight}(e_2) \end{pmatrix}$$

Weights can represent similarities or distances/volumes.

Codifferential operator

$$\langle B_d f_{d-1}, h_d \rangle_{M_d} = \left\langle f_{d-1}, {B_d}^{\dagger} h_d \right\rangle_{M_{d-1}}$$

Codifferential operator ${B_d}^{\dagger} = {M_{d-1}}^{-1} {B_d}^{\top} M_d$.

- ${B_d}^{\dagger}$ is adjoint to B_d w.r.t. M_d.
- Gradient B_1, divergence ${B_1}^{\dagger}$, curl B_2.

Dirichlet energy: defines the Laplacian

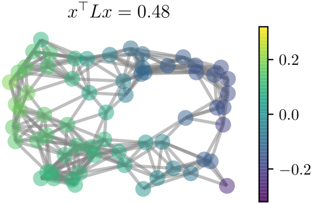
$$\left\langle {B_d}^\dagger f_d, {B_d}^\dagger h_d \right\rangle_{M_{d-1}} + \langle B_{d+1} f_d, B_{d+1} h_d \rangle_{M_{d+1}} = \langle f_d, L_d h_d \rangle_{M_d}$$

Laplacian as the second-order differential operator

$$L_d = B_d {B_d}^\dagger + {B_{d+1}}^\dagger B_{d+1}$$

Dirichlet energy: measure of variation

$$E(f_d) = \langle f_d, L_d f_d \rangle_{M_d} = \|B_d{}^{\dagger} f_d\|^2_{M_{d-1}} + \|B_{d+1} f_d\|^2_{M_{d+1}}$$



$$E(f_0) = \langle f_0, L_0 f_0 \rangle_{M_0} = \|B_1 f_0\|^2_{M_1}$$

Generalized convolutions

My contributions: motivation, construction, analysis, and usage
of generalized convolutions for efficient Machine Learning.

Graphs

- Graph G of $n = |K_0|$ vertices.
- Incidence matrix $B = B_1$.
- Unweighted vertices $M_0 = I$ and edge weights $M = M_1$.
- Laplacian $L = L_0 = B^\dagger B = B^\top M B$.

Symmetries

$$\sigma \in \mathrm{Aut}(G) \subset S_n$$

- Automorphism σ.
- Automorphism group $\mathrm{Aut}(G)$.
- $0 \leq |\mathrm{Aut}(G)| \leq |S_n|$ symmetries.

Representation (spatial basis): permutation matrix P_σ.

Equivariance

$$P_\sigma{}^\top L P_\sigma = L \qquad\qquad LP_\sigma = P_\sigma L$$

- Symmetry preserves the adjacency structure.
- The Laplacian commutes with symmetry group actions.
- The Laplacian is an intrinsic and equivariant operator.

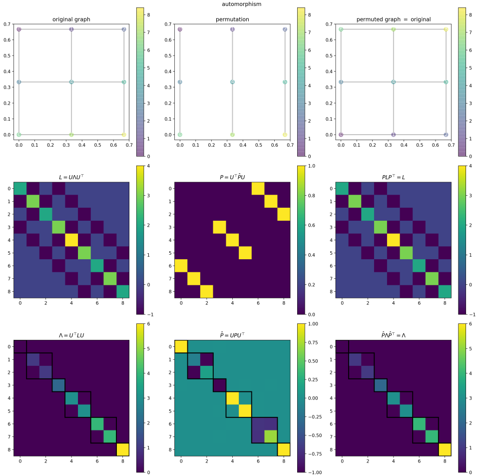
Fourier diagonalizes actions

$$L = U \Lambda U^{-1}$$

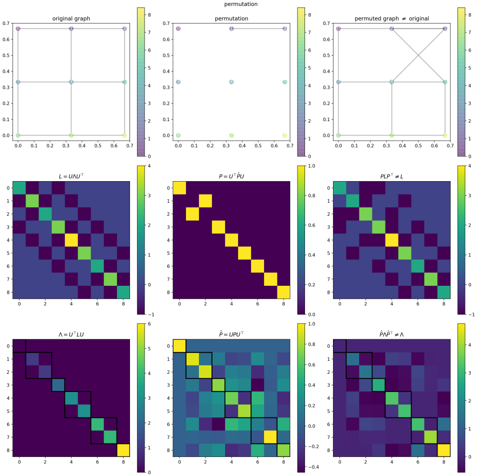
- Symmetries must act as rotations within the eigenspaces of L.
- Fourier jointly (block-)diagonalizes L and P_σ—without knowing the symmetries.

Special case of the Peter-Weyl theorem (compact groups) and Pontryagin duality (Abelian groups).

Fourier diagonalizes actions



Automorphism: $PLP^\mathsf{T} = L$.



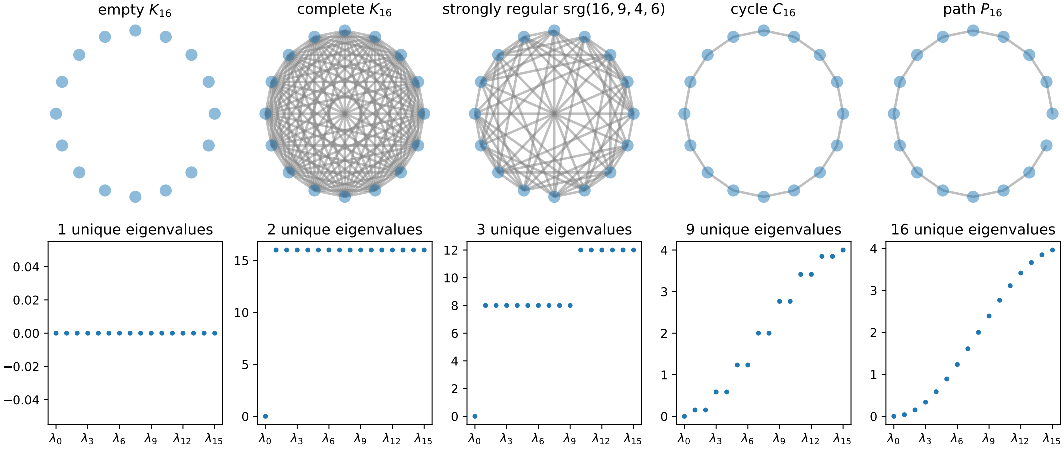
Permutation: $PLP^\mathsf{T} \neq L$.

Spectral basis

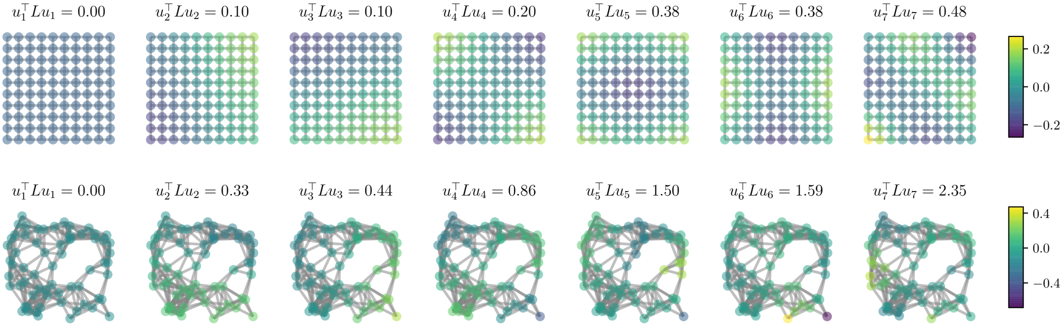
$$L = U\Lambda U^{-1}$$

- Fourier $U = [u_1, \dots, u_n]$, eigenvectors u_i.
- Squared frequencies $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$.
- Because L is positive semi-definite [spectral theorem].
- Reduces to the discrete cosine (DCT) and Fourier (DFT) transforms.

Spectral basis: eigenvalues

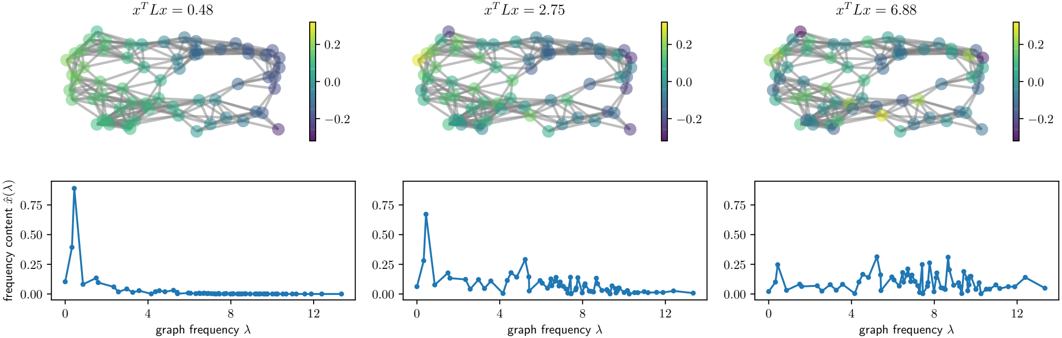


Spectral basis: eigenvectors



Spectral basis: Fourier transform

$$\hat{x} = U^{-1}x \qquad x = U\hat{x} \qquad E(x) = x^\mathsf{T} L x = \hat{x}^\mathsf{T} \Lambda \hat{x}$$



Generalized convolutions

$$\varLambda = U^{-1} L U \qquad \varPi_\sigma = U^{-1} P_\sigma U$$

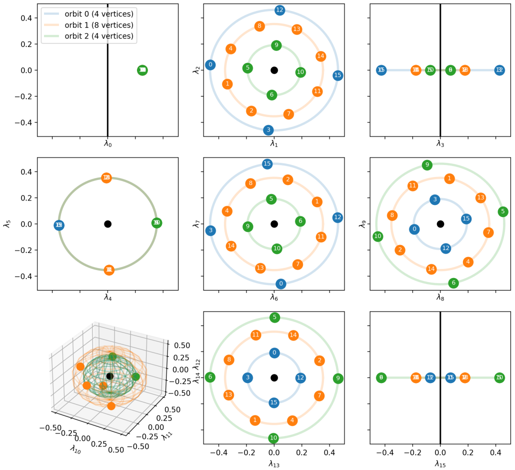
- The eigenspaces are the invariant subspaces of both operators.
- $\varLambda$ is diagonal: one value per eigenspace.
- $\varPi_\sigma$ is block-diagonal: one block per eigenspace. Each block implements a roto-reflection.

Generalized convolutions (spectral basis)

$$g(\Lambda) = \mathrm{diag}\big(g(\lambda_1), \ldots, g(\lambda_n)\big)$$

$$g(\Lambda)\Pi_\sigma = \Pi_\sigma g(\Lambda)$$

The action $g(\Lambda)$ of g (scaling) is orthogonal to the action Π_σ of σ (roto-reflection).



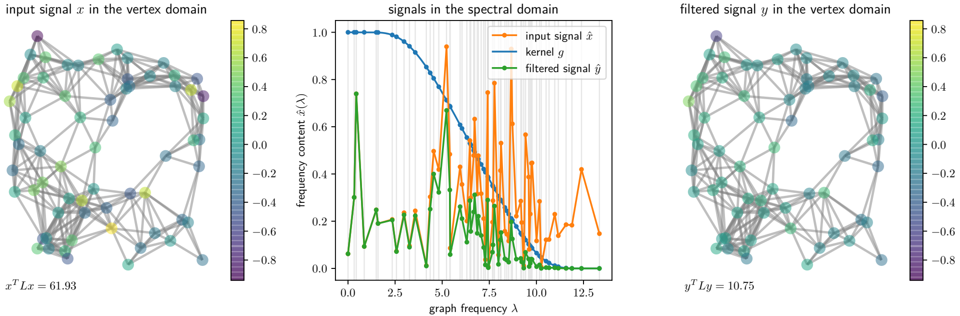
Generalized convolutions (spatial basis)

$$g(L) = U g(\Lambda) U^{-1}$$

- Multiplication operator $g(\Lambda)$ and convolution operator $g(L)$.
- $g(L)$ is an equivariant operator, the defining property of convolutions.
- Generalized because it commutes with more than symmetries.

Filtering

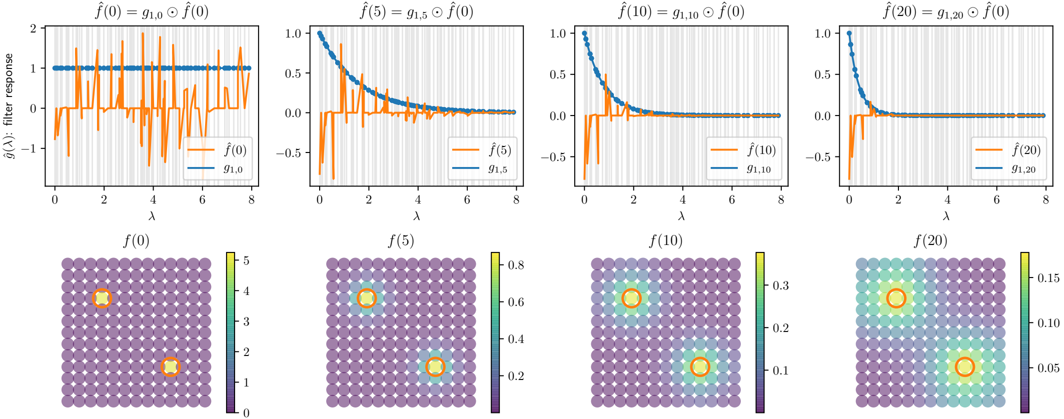
$$y = g(L)x = Ug(\Lambda)U^{-1}x$$



Left: data x in the spatial basis. Middle: data $\hat{x} = U^{-1}x$, concrete filter diag$(g(\Lambda))$, and filtered data $\hat{y} = g(\Lambda)\hat{x}$ in the spectral basis. Right: filtered data $y = U\hat{y}$ in the spatial basis.

Filtering: heat diffusion

$$-\tau L f(t) = \partial_t f(t) \quad \Rightarrow \quad f(t) = g_{\tau t}(L) f(0) \text{ with } g_{\tau t}(\lambda) = \exp(-\tau t \lambda)$$



Designing g

Design a kernel $g : \mathbb{R} \to \mathbb{R}$ such that it acts interestingly as $y = g(L)x$.

- $g(\lambda) = \exp(-\tau t \lambda)$: heat diffusion.
- $g(\lambda) = \cos\left(t \arccos\left(1 - \frac{\tau^2}{2}\lambda\right)\right)$: wave propagation.
- $g(\lambda) = \begin{cases} 1 & \text{if } \lambda_{\text{min}} < \lambda < \lambda_{\text{max}}, \\ 0 & \text{otherwise.} \end{cases}$: projection on a subspace.
- $g(\lambda) = \frac{1}{1+\tau\lambda}$: denoising with $\arg\min_y \|y - x\|_2^2 + \tau y^\intercal L y$.

Learn g if the process is unknown.

Convolution: symmetry action vs localization

Convolution with symmetry action.

$$\langle y, \delta_i \rangle = \langle T_i g, x \rangle$$

- $T_i g$ shifts g to the i^{th} vertex.
- x and g are the same objects.

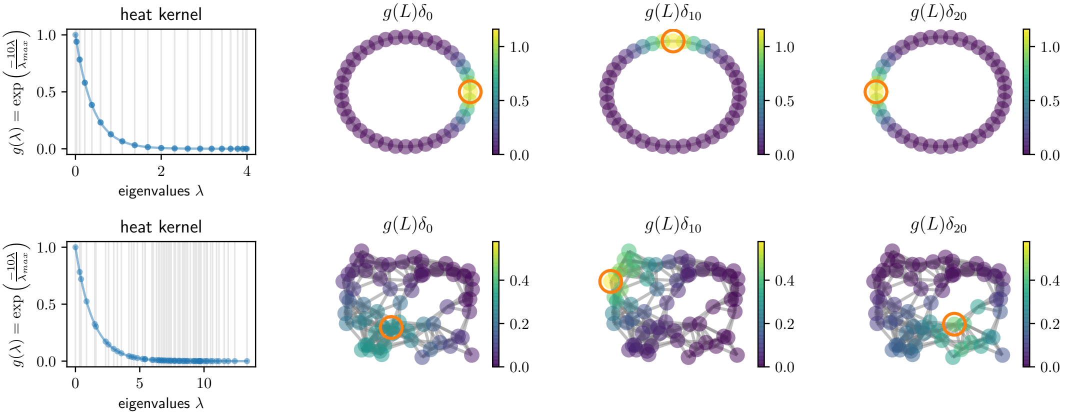
Convolution with localization.

$$\begin{aligned} \langle y, \delta_i \rangle &= \langle g(L)x, \delta_i \rangle \\ &= \langle x, g(L)\delta_i \rangle \end{aligned}$$

- $g(L)\delta_i$ localizes g at the i^{th} vertex.
- x and g are different.

Localization is a generalization of symmetry action to non-homogeneous spaces.

Convolution: symmetry action vs localization



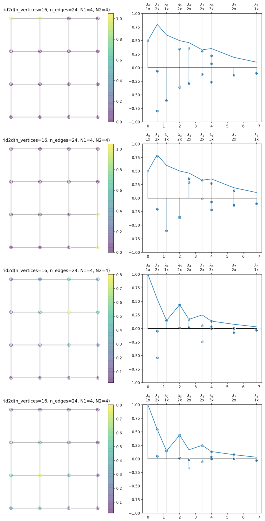
Localization reduces to symmetry action.

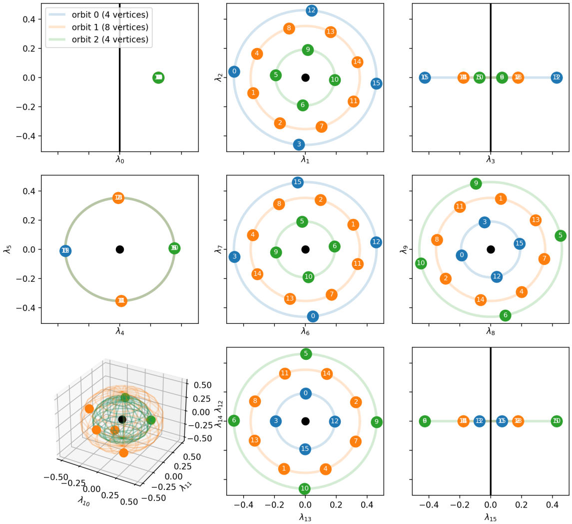
Spectral embedding

$$E_{\mathrm{g}}(f) = \langle f, \mathrm{g}(L) f \rangle = \left\| \mathrm{g}^{1/2}(\Lambda) U^{-1} f \right\|_2^2$$

- Generalization of Dirichlet energy to $\mathrm{g} \neq \mathrm{id}$.
- $\mathrm{g}^{1/2}(\Lambda) U^{-1} f$ is an embedding of f in Euclidean space that reproduces:
  - the symmetries of the space encoded in L,
  - a notion of distance set by g.

Spectral embedding





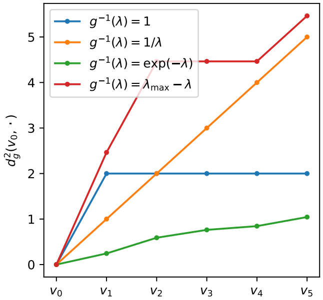
Network (vertex) embedding

$$Q = g^{1/2}(\Lambda)U^{-1}$$

- Embedding $Q = [q_1, \ldots, q_n]$, where $q_i \in \mathbb{R}^n$ represents the i^{th} vertex.
- Covariance $Q^\top Q = Ug(\Lambda)U^{-1} = g(L)$.
  PCA with principal directions u_i and variances $g(\lambda_i)$.

Distance

$$d_g^2(v_i, v_j) = \|q_i - q_j\|_2^2 = E_g(\delta_i - \delta_j)$$



Distances on a path graph.

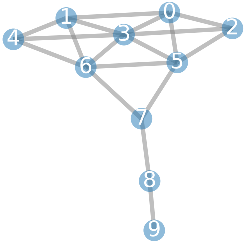
- $g^{-1}(\lambda) = 1$: Laplacian eigenmaps
  [Belkin & Niyogi ’01]
- $g^{-1}(\lambda) = 1/\lambda$: resistance/commute-time distance
  [Klein & Randić ’93] [Göbel & Jagers ’74] [Fouss et al. ’07]
- $g^{-1}(\lambda) = \exp(-2t\lambda)$: (heat) diffusion distance
  [Coifman & Lafon ’06] [Kondor & Lafferty ’02]
- $g^{-1}(\lambda) = (a - \lambda)^p, a \geq \lambda_{\max}$: p-step random-walk
  [PageRank, Brin & Page ’98]

Centrality

$$C_g^2(v_i) = \|q_i\|_2^2 = E_g(\delta_i) = \big(g(L)\big)_{ii}$$

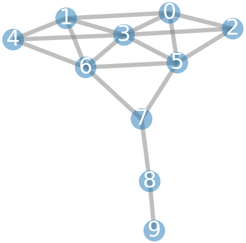
- Measures how close a vertex is to all others.
- Why? $\sum_j \|q_j - q_i\|^2 = \sum_j \|q_j\|_2^2 + n\|q_i\|_2^2 \propto \|q_i\|_2^2 = C_g^2(v_i)$.
- Closer to the origin (center of mass) implies closer to all other vertices.

Designing g: different notions of distance

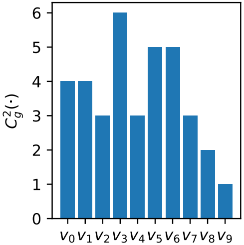


Krackhardt kite graph.

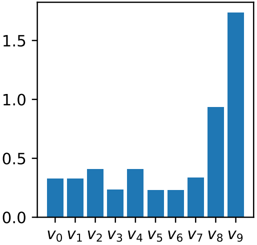
Designing g: different notions of distance



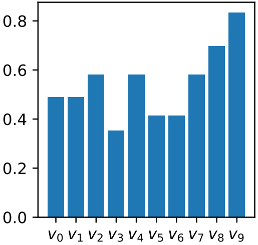
Krackhardt kite graph.



Degree centrality $g(\lambda) = \lambda$.



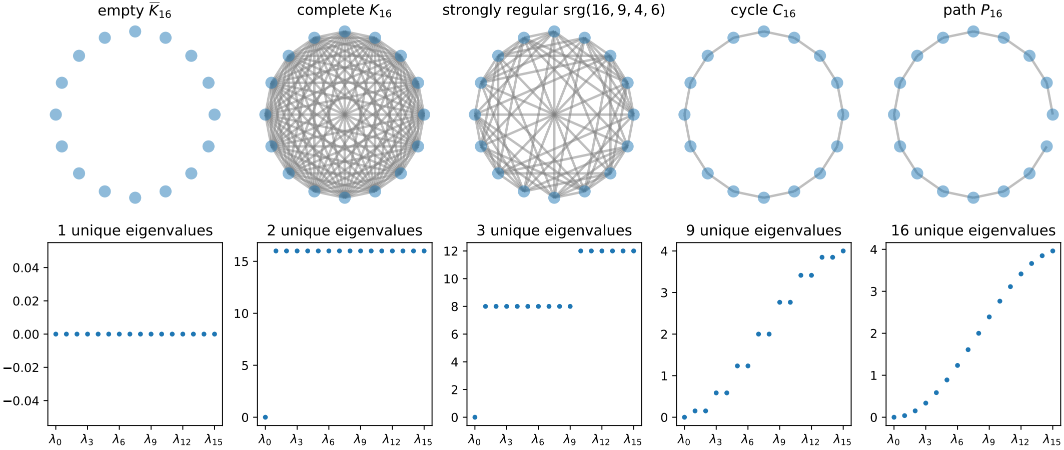
Closeness centrality $g(\lambda) = \lambda^{-1}$.



Diffusion centrality $g(\lambda) = \exp(0.2\lambda)$.

Degree centrality is contravariant, the others are covariant.
Closeness centrality with resistance instead of the typical shortest-path distance.

Learning g: degrees of freedom



Number of independent distances: from 1 to n.

Transfer across graphs

- g is an abstract convolution kernel, specified independently of any graph.
- $g(L) = g(B^\top MB) = Ug(\Lambda)U^{-1}$ is a concrete representation for a graph specified by B and M.
- $g \to g(L)$ is an homomorphism like $\sigma \to P_\sigma$ is.

Summary

1. The kernel g defines a notion of distance.
2. It is represented by the generalized convolution $g(L) = g(B^\top M B)$ on a domain specified by the topology B and geometry M.
3. $g(L)$ is mostly constrained by the domain's symmetries and complexity, constraining the functional space to learn from.
4. $g(L)$ is equivariant to unknown symmetries.
5. Filtering and embedding are one and the same.
6. Design g if you know what you want, learn it if you don't.

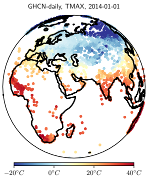
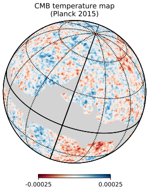
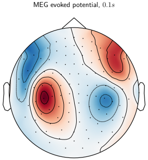
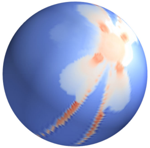
DeepSphere

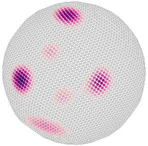
My contributions: motivation, construction, analysis, and usage
of generalized convolutions for efficient Machine Learning.

Problem: learning from spherical data

input $x : S^2 \times \cdots \to \mathbb{R}^d$

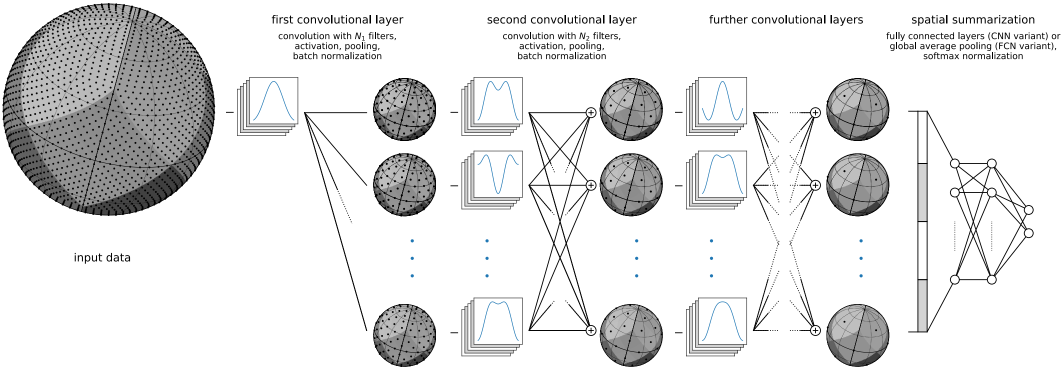
output $f(x)$

intrinsic	projection
	
	

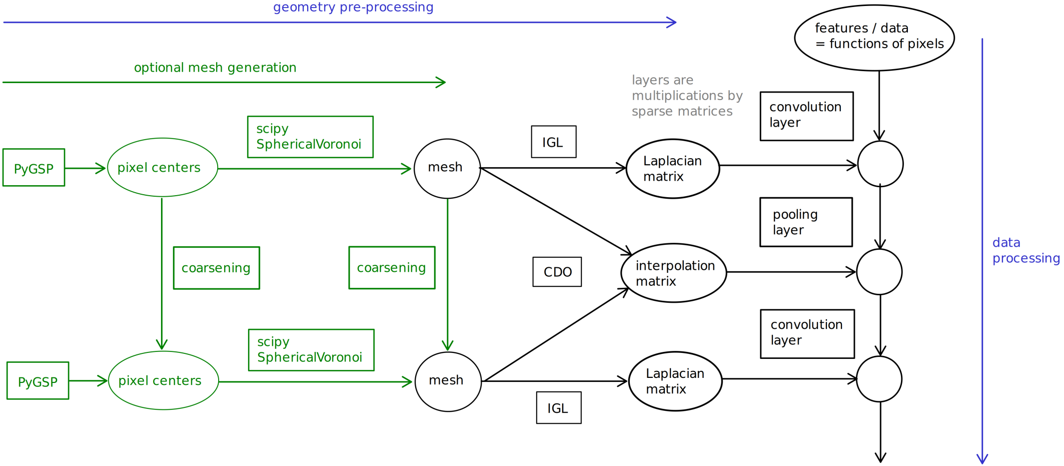
	classification	regression
global	$\begin{cases}\text{“sleepy”}\\ \text{“alert”}\end{cases}$	Ω_m, σ_8
dense		

Acoustic field from Simeoni et al. 2019. 3D shape from Esteves et al. 2018.

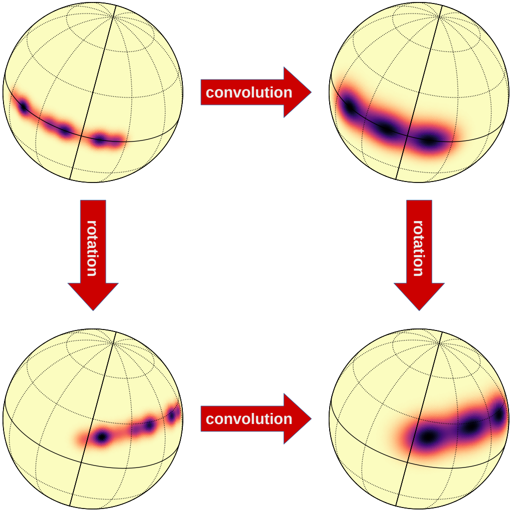
Solution: spherical neural networks



Solution: spherical neural networks



Desideratum 1: equivariant to rotations



- Equivariance for dense tasks:
  $f(P_\sigma x) = P_\sigma f(x) \quad \forall \sigma \in \mathrm{SO}(3)$.

- Invariance for global tasks:
  $f(P_\sigma x) = f(x) \quad \forall \sigma \in \mathrm{SO}(3)$.

Why exploit symmetries?

- Data efficiency.
- Generalization guarantee.

$\Rightarrow$ Principled weight sharing (convolution).

Desideratum 2: scalable

- Many inferences needed for training.
- Increasingly larger maps. ($n = 10^7$ pixels is customary in cosmology.)

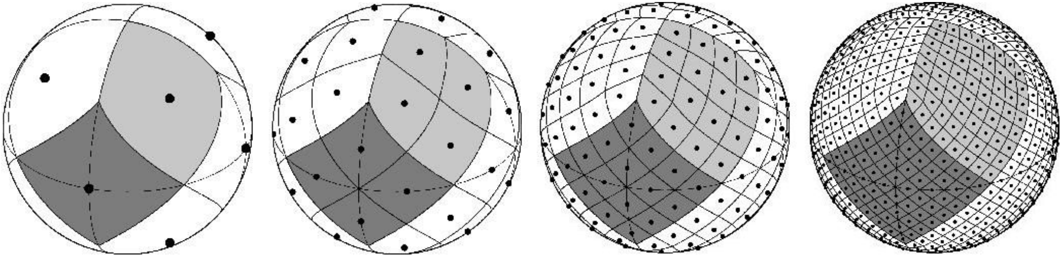
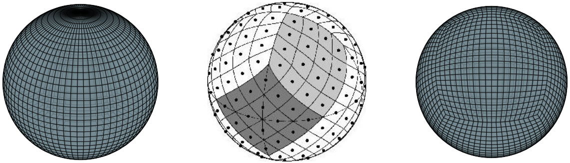
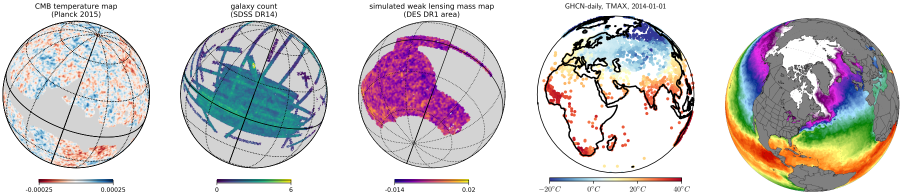


Figure from `https://healpix.sourceforge.io`.

Desideratum 3: flexible sampling, avoid interpolation



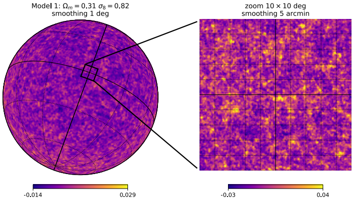
Sampling schemes: equiangular, HEALPix, cubed-sphere, icosahedral, Gauss–Legendre, etc.

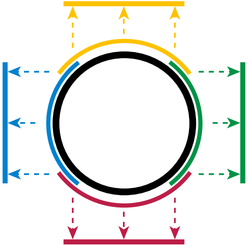


Partial and irregular sampling.

Some figures from Boomsma and Frellsen 2017 and `https://climatereanalyzer.org`.

Method 1: 2D projections





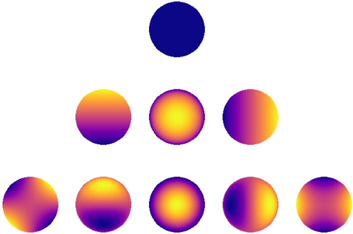
Manifold is locally Euclidean!
Project on tangent planes.

Desiderata

- ⊖ Rotation equivariance: hard to glue planes together.
- ⊕ Scalability: well developed NN architectures and implementations. Some wastes at boundaries.
- ⊖ Flexibility: only handle compact subspaces.

Charting figure from `https://en.wikipedia.org/wiki/manifold`.

Method 2: discretization of continuous domain



Spectral decomposition.

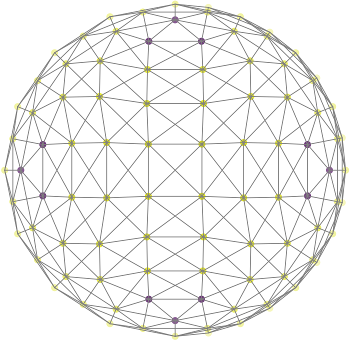
Discretize but consider the continuous symmetries.
Group convolution: multiplication in the spectrum after a spherical harmonic transform (SHT).

Desiderata

- ⊕ Rotation equivariance: well understood theory.
- ⊖ SHT is expensive. Even if faster transforms exist for some samplings.
- ⊖ Flexibility: unused pixels are mostly wasted.

Figure from `https://rodluger.github.io/starry`.

Our method: discrete domain



Domain pixels K_0, topology B, geometry M

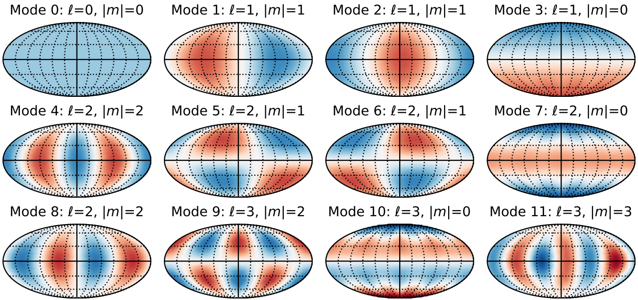
Data $x \in \mathbb{R}^n, \ n = |K_0|$

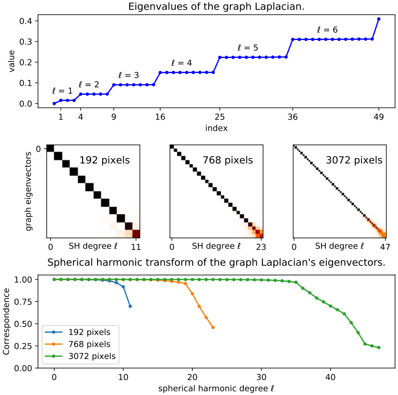
Map $g_\alpha(L)x = \sum_k \alpha_k L^k x, \ L = BMB^\intercal$

Parameters $\alpha \in \mathbb{R}^K$

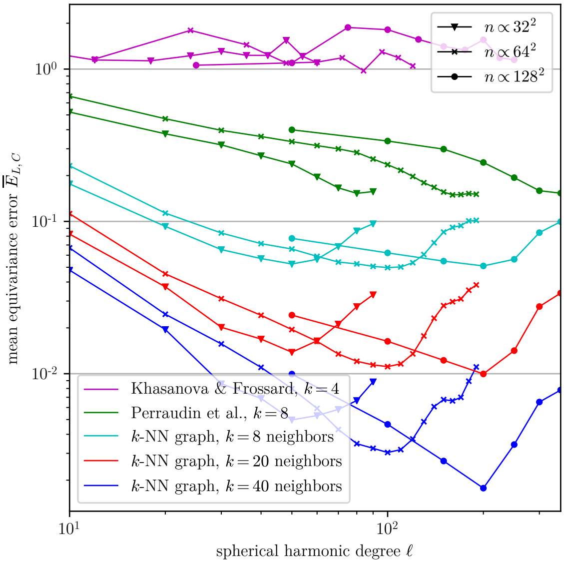
Graph Fourier basis on the sphere

- Fourier modes approximate spherical harmonics.
- The graph approximates the sphere.





Desideratum 1: equivariant to rotations



- Equivariance error:

$$\mathbb{E}_{\sigma,x}\left(\frac{\|P_\sigma L x - L P_\sigma x\|}{\|Lx\|}\right)^2$$

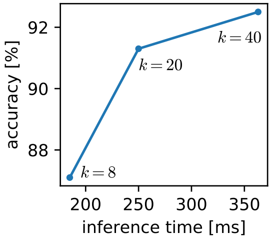
- Tradeoff between equivariance and cost (number of vertices n and edges kn) in the topology B.

- Difficulty: get the geometry M right.

Desideratum 1: it matters!

	accuracy	time
Perraudin et al. 2019, 2D CNN baseline	54.2	104 ms
Perraudin et al. 2019, CNN variant, $k = 8$	62.1	185 ms
Perraudin et al. 2019, FCN variant, $k = 8$	83.8	185 ms
$k = 8$ neighbors, optimal t	87.1	185 ms
$k = 20$ neighbors, optimal t	91.3	250 ms
$k = 40$ neighbors, optimal t	92.5	363 ms

Lower equivariance error translates to higher performance.



Tradeoff between cost and accuracy.

Desideratum 2: linear complexity

Goal: avoid the $O(n^3)$ EVD $L = U\Lambda U^{-1}$ and $O(n^2)$ matrix multiplication $U^{-1}x$ in evaluating $g(L)x = Ug(\Lambda)U^{-1}x$.

Spatial parameterization (K-hops local):

$$g_\alpha(L)x = \left(\sum_{k<K} \alpha_k L^k\right)x = \sum_{k<K} \alpha_k \bar{x}_k, \quad \bar{x}_k = L\bar{x}_{k-1}, \; \bar{x}_0 = x.$$

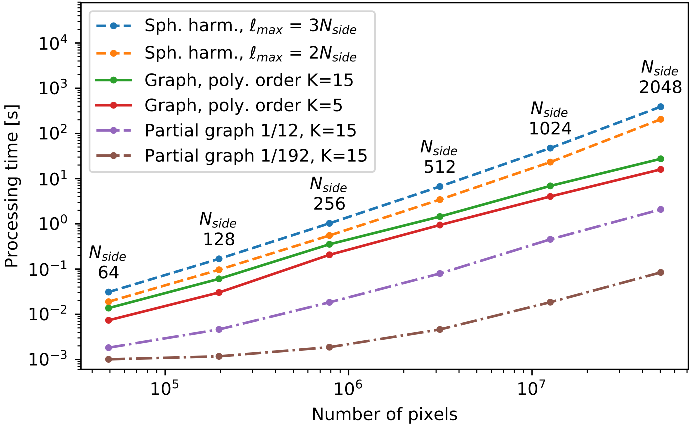
Spectral parameterization (global):

$$g_\alpha(L)x = \sum_{k\in\mathcal{K}} \alpha_k u_k {u_k}^\intercal x, \quad \alpha_k = g(\lambda_k), \quad U = [u_1, \dots, u_n].$$

Heisenberg's uncertainty principle: locality in the spatial domain implies smoothness in the spectral domain and vice-versa.

Desideratum 2: scalable

- Graph convolutions cost $O(n)$.
- Spherical convolutions cost $O(n^2)$ in general, $O(n^{3/2})$ for some samplings.

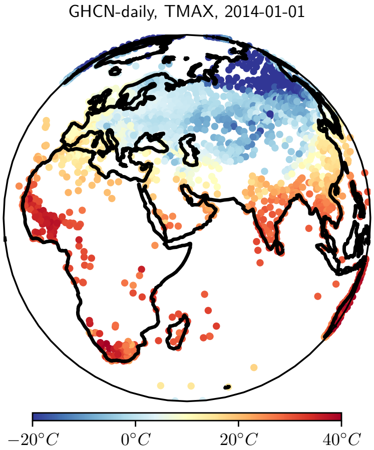


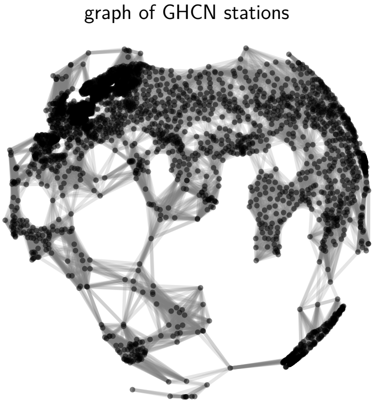
Desideratum 2: it matters!

	performance		size	speed	
	F1	mAP	params	inference	training
Cohen et al. 2018 ($b = 128$)	–	67.6	1400 k	38.0 ms	50 h
Cohen et al. 2018 (simplified, $b = 64$)	78.9	66.5	400 k	12.0 ms	32 h
Esteves et al. 2018 ($b = 64$)	79.4	68.5	500 k	9.8 ms	3 h
DeepSphere (equiangular, $b = 64$)	79.4	66.5	190 k	0.9 ms	50 m
DeepSphere (HEALPix, $N_{\text{side}} = 32$)	80.7	68.6	190 k	0.9 ms	50 m

Classification of 3D shapes (SHREC'17): anisotropy is an unnecessary price to pay.

Desideratum 3: flexible sampling

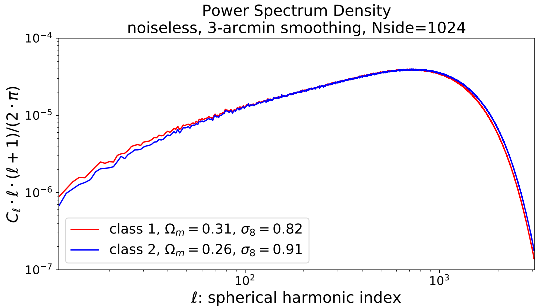




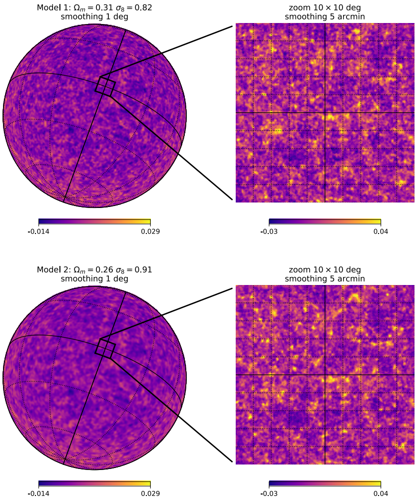
Application: discrimination of cosmological models

Classification of convergence maps created from two sets of cosmological parameters.

$$(\Omega_m, \sigma_8) = (0.31, 0.82) \text{ or } (0.26, 0.91)$$

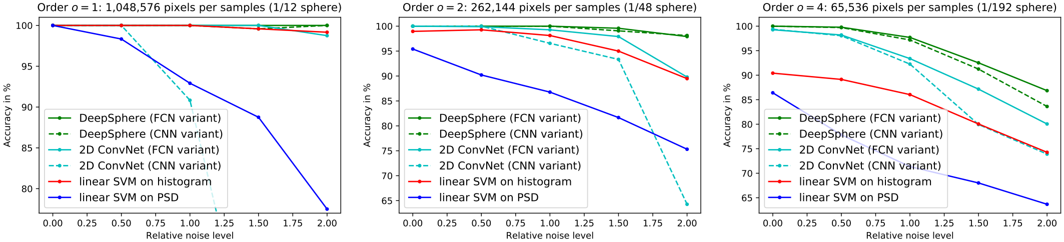


Ω_m, σ_8, smoothing chosen to get identical PS.



Maps with identical initial conditions.

Application: discrimination of cosmological models (results)

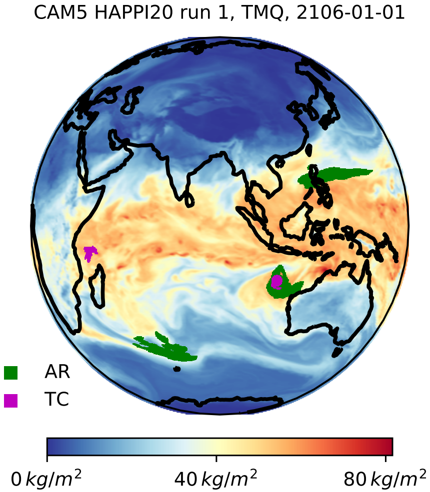


- Difficulty controlled by #pixels per sample and amount of noise.
- Better performance than SVM on PSDs and histograms.
  Those statistics destroy too much information.
- Better performance than ConvNet on 2D projections.
  Equivariance matters.

Application: climate event segmentation

Segment extreme climate events: tropical cyclones (TC) and atmospheric rivers (AR).

- >1M spherical maps
- down-sampled to 10k pixels (original 900k)
- 0.1% TC, 2.2% AR, 97.7% background
- 16 channels (e.g., temperature, wind, humidity, pressure)



Application: climate event segmentation (results)

	accuracy	mAP
Jiang et al. 2019 (rerun)	94.95	38.41
T. S. Cohen et al. 2019 (S2R)	97.5	68.6
T. S. Cohen et al. 2019 (R2R)	97.7	75.9
DeepSphere (weighted loss)	97.8 ± 0.3	77.15 ± 1.94
DeepSphere (non-weighted loss)	87.8 ± 0.5	89.16 ± 1.37

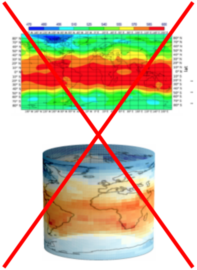
Mean accuracy (over TC, AR, BG) and mean average precision (over TC and AR).

- Anisotropy is an unnecessary price to pay.
- Check your loss!

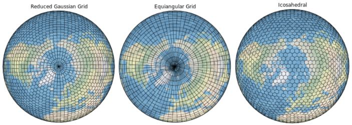
Application: weather forecasting

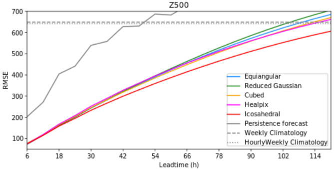
Ghiggi et al. 2022

Topology and geometry of the Earth



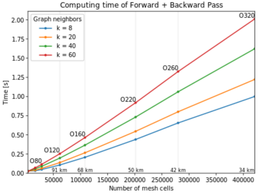
Any (unstructured) grid: no interpolation!





https://github.com/deepsphere/deepsphere-weather

Scalable



convolution $$y = \sum_k w_k L^k x$$

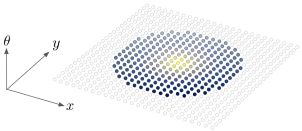
pooling $$y = Px$$

non-linearity $$y = \sigma(x)$$

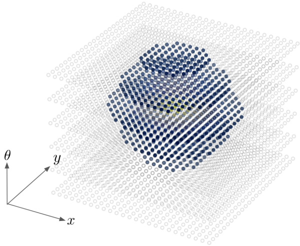
Anisotropy

ChebLieNet, Aguettaz, Bekkers, and Defferrard 2021

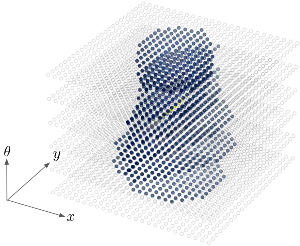
- No free lunch: lift to symmetry group, required to be transitive and known.
- Similar to group convolutions, but with control of the equivariance–cost tradeoff.



Isotropic metric on $\mathcal{M}$.



Isotropic metric on Sym($\mathcal{M}$).



Anisotropic metric on Sym($\mathcal{M}$).

Diffusion on base Riemannian manifold $\mathcal{M} = \mathbb{R}^2$ and symmetry Lie group Sym($\mathcal{M}$) $=$ SE(2).

Summary

DeepSphere: a spherical CNN that strikes

a controllable balance between desiderata.

Conclusion

My contributions: motivation, construction, analysis, and usage
of generalized convolutions for efficient Machine Learning.

Summary

Leveraging topology, geometry, and symmetries for efficient Machine Learning

Generalized convolutions

theory	emerge from the fundamentals of space: topology and geometry,
method	enable parameter sharing for non-transitive and unknown symmetry groups, to efficiently learn on arbitrary domains,
application	lead to state-of-the-art results on important real-world problems.

Impact

My contributions:

- got 5000+ citations and an h-index of 10,
- pioneered graph ML and put it on the global research agenda,
- proved useful in tackling important real-world problems.

Future: structural features and network embedding

Problem GNNs are good at leveraging graphs as a computational substrate to process data; But not to extract information from graphs.

Observation These operations are two sides of the same $g(L)$ coin. But spectral embeddings $Q = g^{1/2}(\Lambda)U^{-1} = [q_1, \dots, q_n]$ are not invariant to automorphisms. DeepWalk, LINE, PTE, and node2vec embed in a subspace for some g [Qiu et al. 2018].

Solution Centrality $C_g^2(v_i) = \|q_i\|_2^2$ and distances $\{\{d_g^2(v_i, \cdot)\}\} = \{\{\|q_i - \cdot\|_2^2\}\}$.

Future: graph isomorphism

Problem Is GI in P or NP-complete?

Observation Neither centrality $C_g^2(v_i)$ nor distances $\{\{d_g^2(v_i, \cdot)\}\}$ are complete invariants w.r.t. automorphism/isomorphism.

Michaël Defferrard @m_deff · Dec 31, 2020
2020: I got the GI disease
2021: I will find a cure or get immune

Happy New Year y'all! 🎉

The graph isomorphism disease[†]

Ronald C. Read, Derek G. Corneil

First published: Winter 1977 | https://doi.org/10.1002/jgt.3190010410

[†] Dedicated to George Pólya on his 90th Birthday.

Slides `https://doi.org/10.5281/zenodo.5780063`

Papers

Defferrard, Generalized convolutions, In preparation, 2022.

Ebli, Defferrard, Spreemann, Simplicial Neural Networks, TDA@NeurIPS, 2020.

Defferrard, Milani, Gusset, Perraudin, DeepSphere: a graph-based spherical CNN, ICLR, 2020.

Perraudin, Defferrard, Kacprzak, Sgier, DeepSphere: Efficient spherical Convolutional Neural Network with HEALPix sampling for cosmological applications, Astronomy and Computing, 2019.

Defferrard, Bresson, Vandergheynst, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, NIPS, 2016.

Code

```
https://github.com/epfl-lts2/pygsp
https://github.com/stefaniaebli/simplicial_neural_networks
https://github.com/deepsphere
https://github.com/mdeff/cnn_graph
```