CSC2457 3D & Geometric Deep Learning

CNNs on Surfaces using Rotation-Equivariant Features

Ruben Wiersma, Elmar Eisemann, Klaus Hildebrandt

Date: March 9, 2021

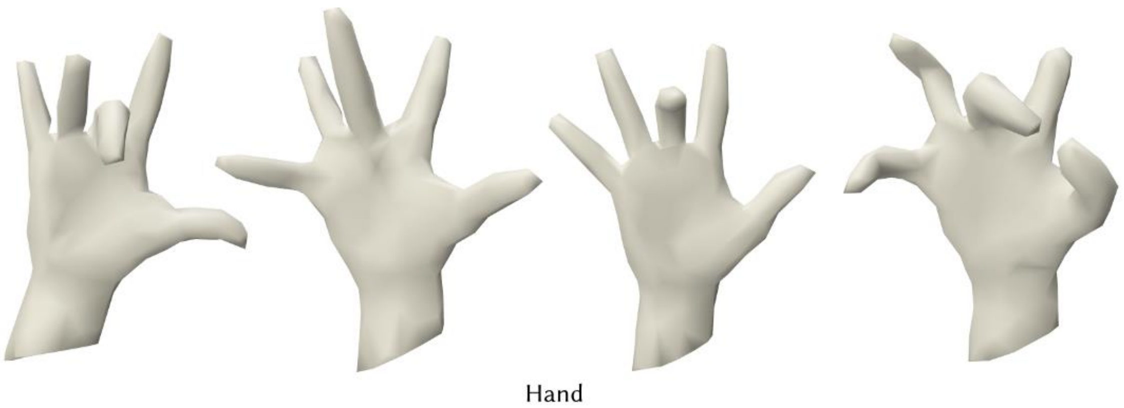
Presenter: Shichen Lu

Instructor: Animesh Garg

UNIVERSITY OF TORONTO

Motivation

- We want to be able to be able to do typical CV tasks, but for surfaces/manifolds
- This is hard due to irregularity, non-Euclidean nature, etc.
- Geometric Deep Learning: how can we extract features from these manifolds?

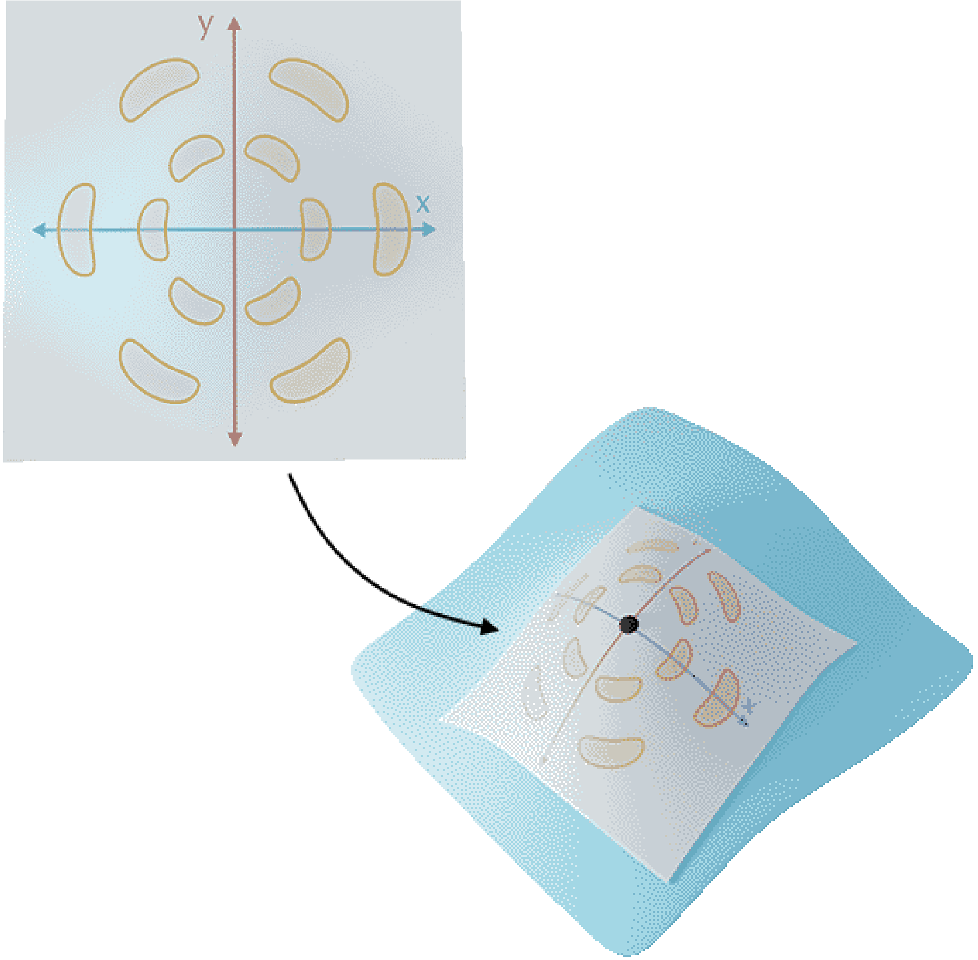


Motivation – Approach Types

- Spectral methods (e.g. Graph Convolutional Network[1])
  - Do convolution based on the graph Laplacian
  - Targeted more towards meshes/graphs than surfaces
- GNNs
  - Again target more towards meshes/graphs than surfaces
- Point Clouds (e.g. Pointnet[2])
  - Loss of expressiveness
- Symmetric Spaces (e.g. Spherical CNNs[3])
  - Specialized approaches for symmetric surfaces
  - Limited to symmetric surfaces

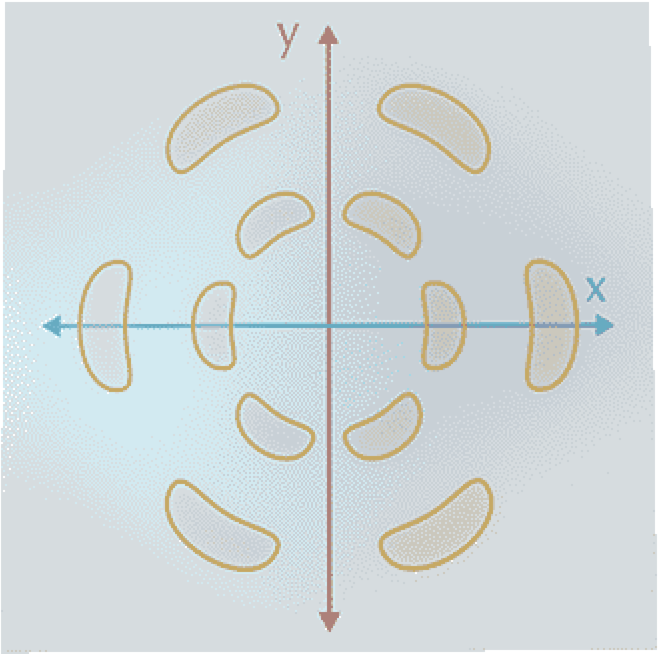
Motivation – Approach Types

- Charting based method
  - Learn a 2D Kernel
  - Define a tangent plane at the point we want to do the convolution
  - Orient the kernel onto the tangent plane
  - Map points on the tangent plane to the surface (or vice versa)
  - Do the convolution
  - Repeat for every point of interest



Motivation – Problem

- We want to apply a 2D convolution filter to a surface
- Problem: Rotation ambiguity
  - Traditional filters output different features based on the rotation of the input
  - For tangent planes on a surface, there is no predefined coordinate system
  - With which rotation should we apply the convolution filter?



Motivation – Previous Approaches

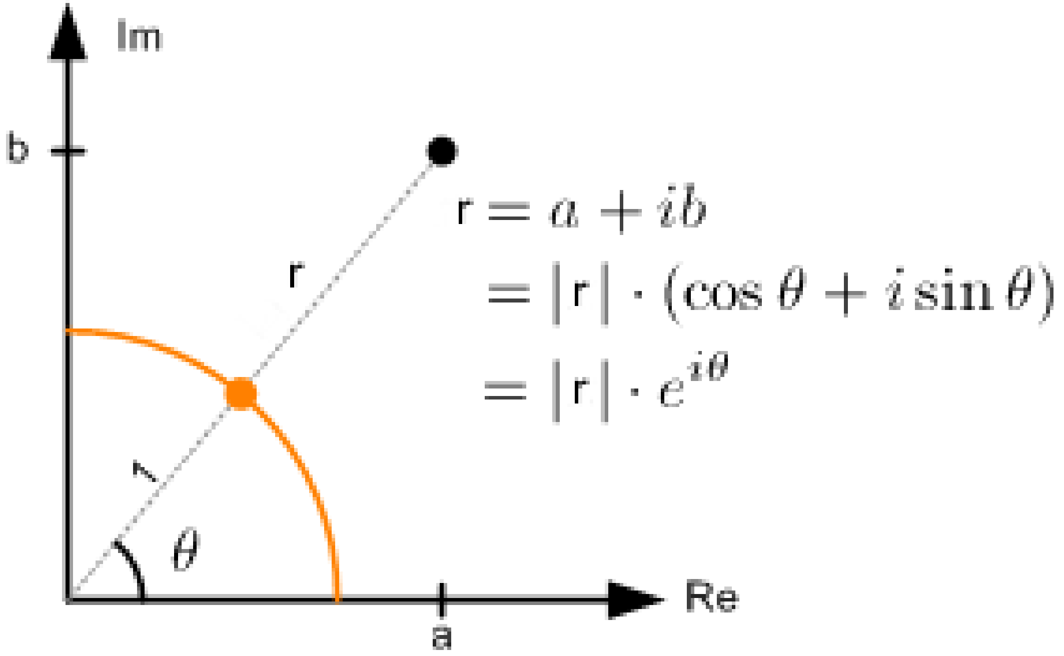
- Define a coordinate system at each point of the surface based on a metric (e.g. ACNN[4])
  - Cannot guarantee consistency of coordinate systems in local neighbourhood of a point (umbilic points)
- Sample multiple rotations and compute convolutions for all of them (e.g. GCNN[5])
  - Computationally expensive
  - Cannot sample in every direction -> discretization or interpolation

Contributions

- Introduce a generalizable, circular harmonics based convolution filter for meshes that is rotation-equivariant
  - Able to solve the rotational ambiguity problem and still capture feature expressiveness
- Introduce Harmonic Surface Networks, which combines the above with pooling and nonlinearity operations for surfaces to perform classification/segmentation on meshes
- Achieves SOTA/competitive performance across multiple tasks

Background – Vector Features

- Represent each feature as a 2D vector, stored as a complex number
- Features are parameterized by the radius, r and the angle, θ



Background – Rotations

- Rotation Invariant
  - Rotating the input does not affect the output
  - The convolution filter will always output the same feature no matter the rotation of the input
- Rotation Equivariant
  - Rotating the input also affects (i.e. rotates) the output in the same way
  - Considering vector features, a rotation of the input will rotate the output vector by the same amount

Background – Harmonic Networks

- A rotation equivariant network used for CV
- Use circular harmonics to construct the convolution filter

$$W_m(r, \theta, R, \beta) = R(r)e^{\mathrm{i}(m\theta+\beta)}$$

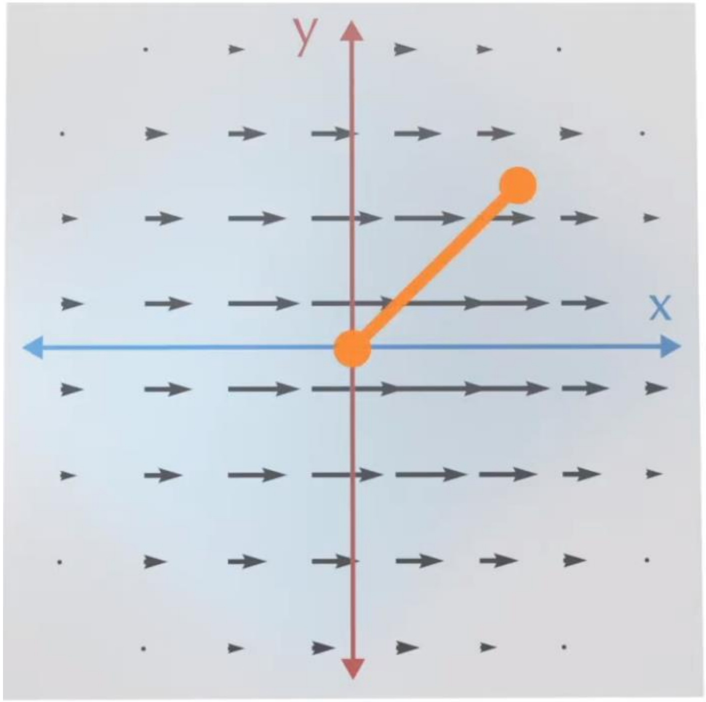
- $R(\cdot)$ is learnt radial profile, β is a learnt offset, m is rotation order
- Rotating the input to the filter is the same as rotating the output!

$$[W_m \star x^{\phi}](p) = e^{\mathrm{i}m\phi}[W_m \star x^{0}](p)$$

Background – Harmonic Networks

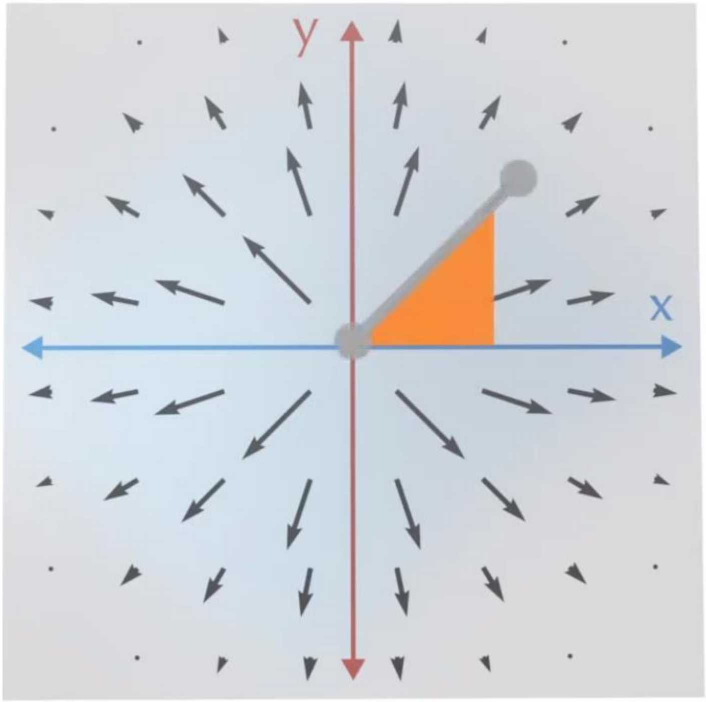
Rotation-invariant

$R(r)e^{i\beta}$



Rotation-equivariant

$R(r)e^{i(\theta+\beta)}$



Example Kernel

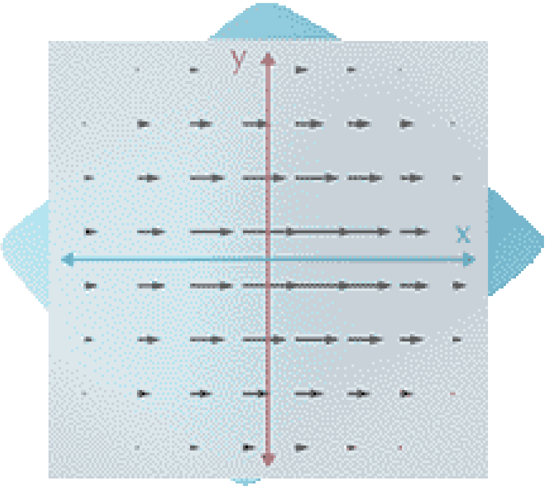
$R(r) = 1 - r$

$\beta = 0$

Background – Harmonic Networks

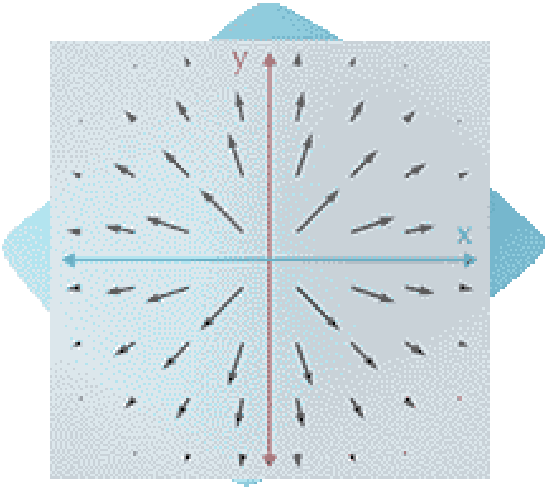
Rotation-invariant

$$R(r)e^{i\beta}$$



Rotation-equivariant

$$R(r)e^{i(\theta+\beta)}$$



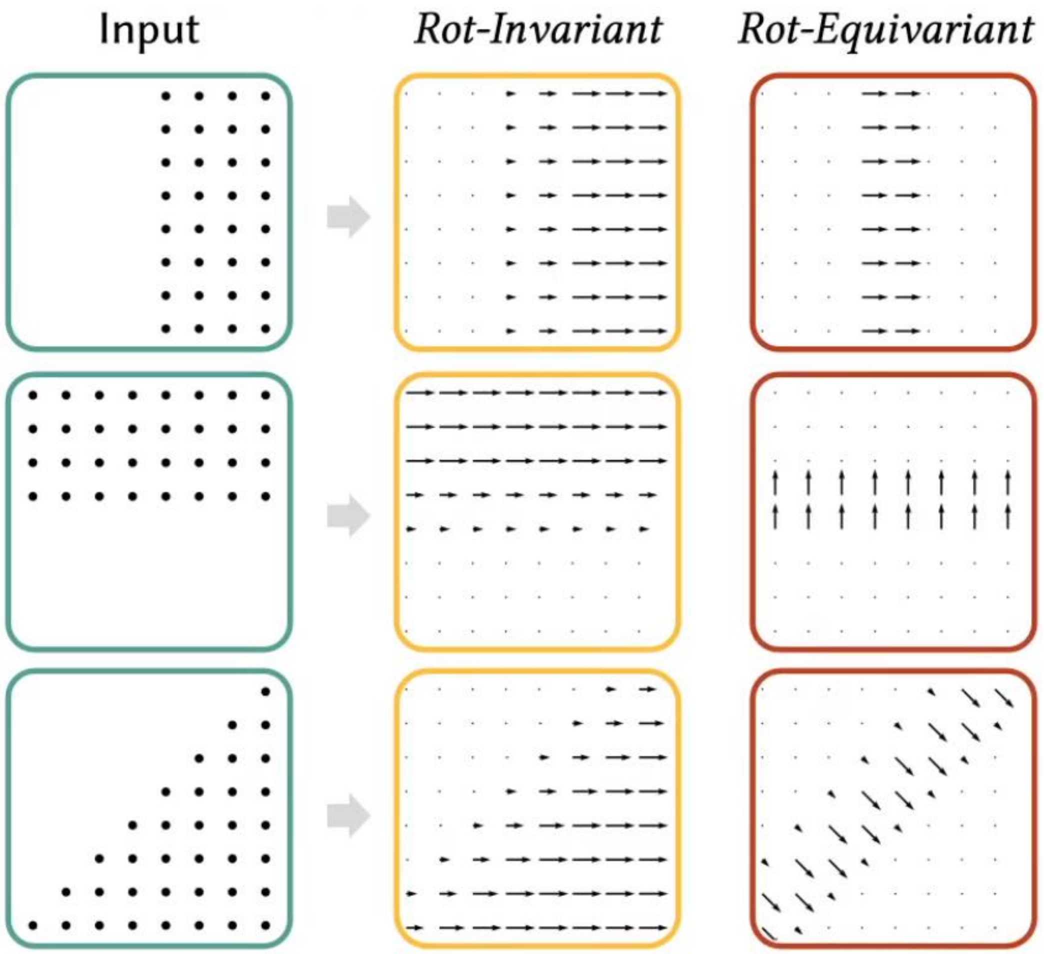
Example Kernel

$$R(r) = 1 - r$$

$$\beta = 0$$

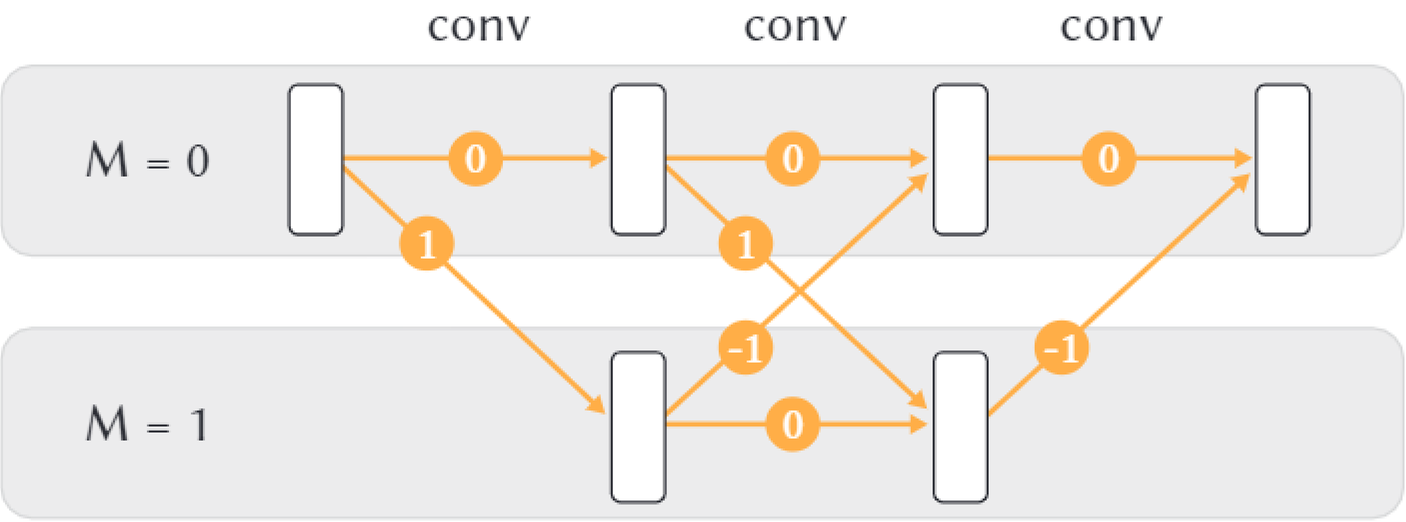
Background – Harmonic Networks

- Example Kernel
  - $R(r) = 1 - r$
  - $\beta = 0$
  - Dots on the input represent "high magnitude" feature points
- Rot-Invariant smooths input
- Rot-Equivariant finds edges



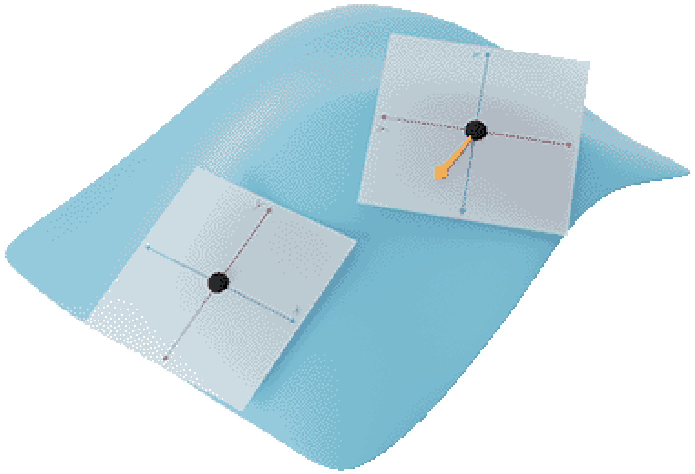
Background – Harmonic Networks

- Use different network streams for different rotation orders
- Convolution operation allows transfer between streams by changing the m parameter



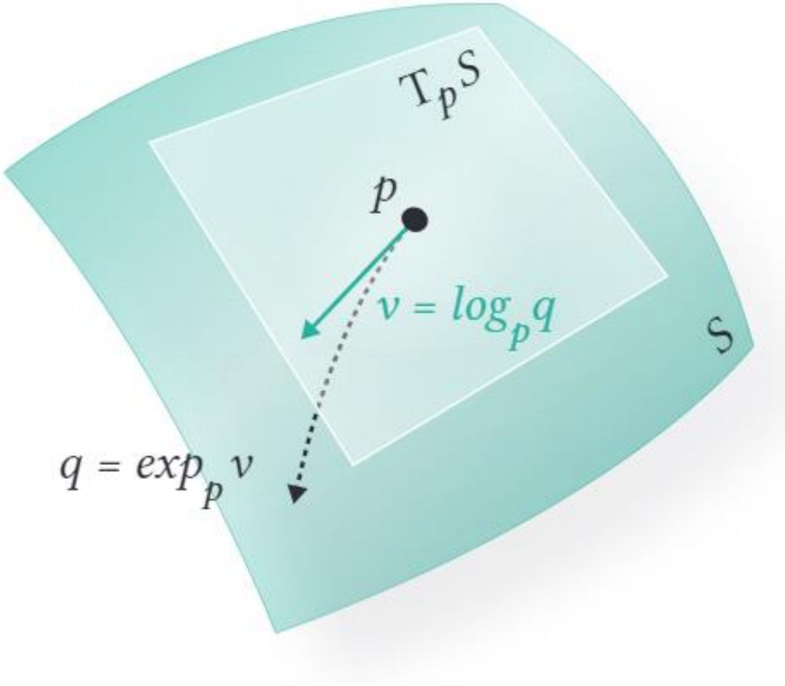
Background – Parallel Transport

- Manifolds are non-Euclidean spaces → we cant directly compare vectors from different points
- What we can do is "transport" a vector from one point to another and then compare them at the same point
- Vectors are transported by "moving" them along a curve while keeping the vector locally equivalent



Background – Exponential Map

- For each point on a manifold, we can define a tangent plane
- The Exponential Map maps from a point on the tangent plane to a corresponding point on the manifold
- We can use this to apply 2D kernels to a surface by mapping the surface point to the 2D tangent plane



Problem Setting

- Input:
  - Triangle mesh of an object
- Output:
  - Shape classification: determine class of an input mesh
  - Shape segmentation: correctly label each point on the mesh
  - Shape correspondence: find matching points between two meshes of similar shape

Method – Convolution Kernel

$$\mathrm{x}_i^{(l+1)} = \sum_{j\in\mathcal{N}_i} w_j \left(R\left(r_{ij}\right) e^{i(-\theta+\beta)} P_{j\to i}\left(\mathrm{x}_j^{(l)}\right)\right)$$

New feature at point i

- Example convolution operation from rotation invariant stream to rotation equivariant stream
- Parallel Transport + Circular Harmonics eliminates rotation ambiguity

Method – Nonlinearities and Pooling

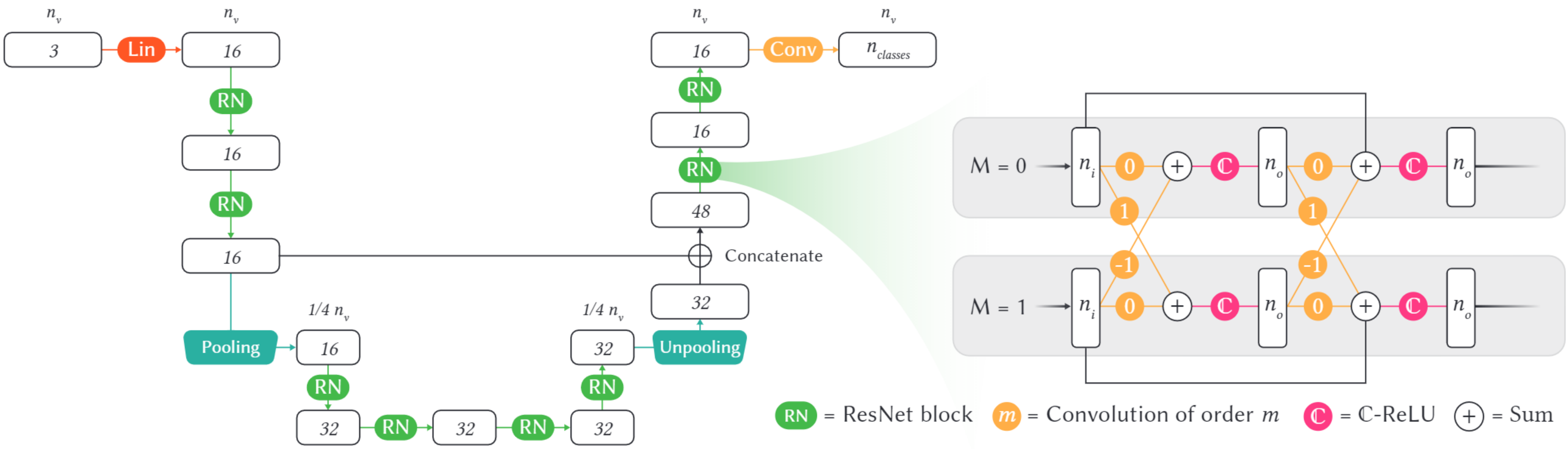
- Features are vector valued and stored as complex numbers
- Apply ReLU to the radius component of the feature + a bias

$$\mathbb{C}\text{−ReLU}_b\left(Xe^{i\theta}\right) = \text{ReLU}(X + b)e^{i\theta}$$

- Pooling works the same way, but with parallel transported features

$$x_i^{(l+1)} = \frac{1}{|C_i|}\sum_{j\in C_i} P_{j\to i}\left(x_j^{(l)}\right)$$

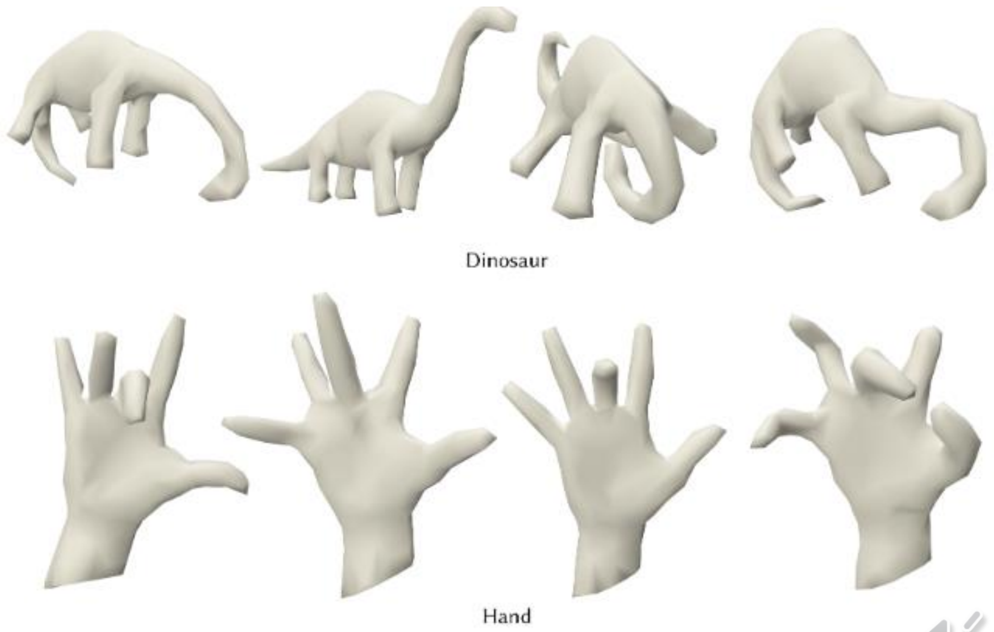
Method – Network Architecture



Results – Shape Classification

- For this task, they use only the first half of the network and only train for ¼ of the time vs other methods
- Dataset low amount of training samples
  - May favor methods that use less parameters
- Dataset has low quality meshes
  - Unfavourable for methods that rely on principal curvature

Method	Accuracy
HSN (ours)	**96.1%**
MeshCNN	91.0%
GWCNN	90.3%
GI	88.6%
MDGCNN	82.2%
GCNN	73.9%
SG	62.6%
ACNN	60.8%
SN	52.7%



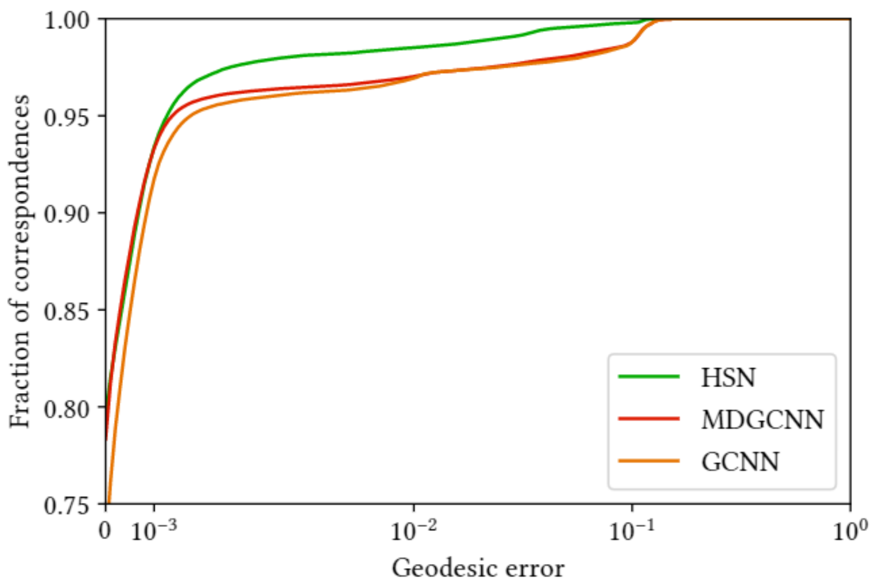
Results – Shape Segmentation

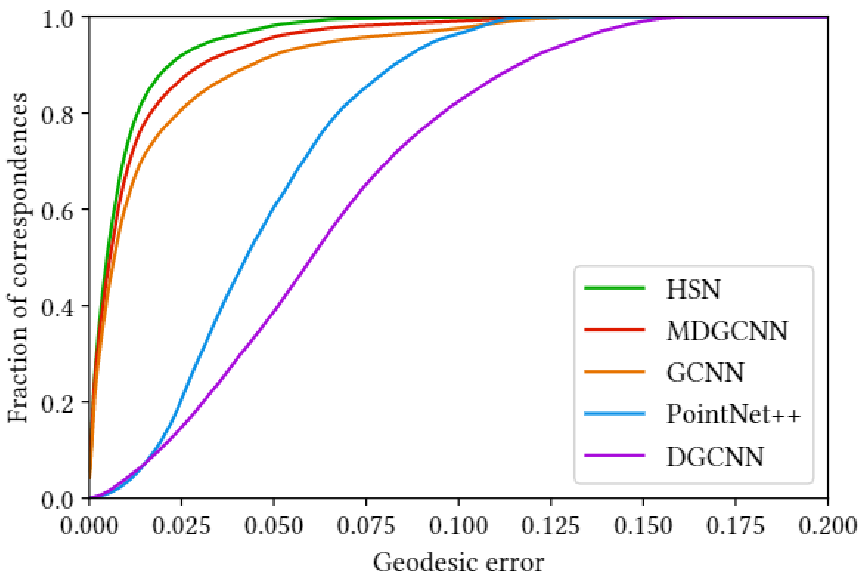
- Only sample 1024 points from each mesh to reduce computation time
  - Potentially could achieve higher accuracy with more samples
- Visualized results for one feature stream in 2nd last layer
  - Feature is both high-activation and rotationally aligned for certain body parts

Method	# Features	Accuracy
HSN (ours)	3	91.14%
MeshCNN	5	**92.30%**
SNGC	3	91.02%
PointNet++	3	90.77%
MDGCNN	64	89.47%
Toric Cover	26	88.00%
DynGraphCNN	64	86.40%
GCNN	64	86.40%
ACNN	3	83.66%

Results – Correspondence

- Left: meshes with same connectivity between shapes
- Right: meshes with irregular connectivity between shapes
- HSN seems to be more robust to connectivity differences in meshs





Results – Discussion

- MeshCNN sometimes better, but deals explicitly with meshes. The proposed approach is more general and can in theory deal with surfaces and point clouds
- Significantly better than MDGCNN and GCNN, which follow a similar charting method, while using less compute
  - Parameter usage is 75% of MDGCNN and 30% of GCNN
  - Uses ~2-4x less memory than MDGCNN

Results – Ablation

- Toy Dataset of MNIST Mapped to a Sphere
- "PC Aligned" uses principal curvature to assign basis vectors for tangent planes instead of using Parallel Transport
- Streams=0 uses only the rotation invariant kernel
- Overall, the rotation equivariant kernel + parallel transport greatly increases accuracy

Shape Classification

Method	Streams ($M = \ldots$)	Accuracy
HSN	0, 1	**96.1%**
HSN	0	86.1%
HSN (pc aligned)	0, 1	49.7%

Shape Segmentation

Method	Streams ($M = \ldots$)	Accuracy
HSN	0, 1	**91.14%**
HSN	0	88.74%
HSN (parameters $\times 4$)	0	87.25%
HSN (pc aligned)	0, 1	86.22%

Limitations

- Requires Vector Heat Method for several calculations
  - This performs poorly with poor mesh quality, or too few elements
- Results are not the best
  - Outperformed by MeshCNN in segmentation
  - Another paper has shown HSN performing just average in correspondence tasks [6]
- Computational Processing
  - Requires many pre-computed operations that can struggle for complex tasks
  - Needs to down-sample #vertices for training time

Conclusion

- We have seen Harmonic Surface Networks for surface classification, segmentation, and correspondence
- Uses a rotation-independent approach to solve the rotation ambiguity problem
- Achieves SOTA/competitive performance

References

[1] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with GraphConvolutional Networks. InICLR.

[2] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017a. PointNet:Deep Learning on Point Sets for 3D Classification and Segmentation. InCVPR. IEEE,77–85

[3] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. 2018. Spherical CNNs. InICLR

[4] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. 2016.Learning shape correspondence with anisotropic convolutional neural networks. InNeurIPS. 3189–3197

[5] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. 2015.Geodesic convolutional neural networks on riemannian manifolds. InICCV. 37–45.

[6] Sharp, N., Attaiki, S., Crane, K., & Ovsjanikov, M. (2020). Diffusion is All You Need for Learning on Surfaces. *ArXiv, abs/2012.00888*.