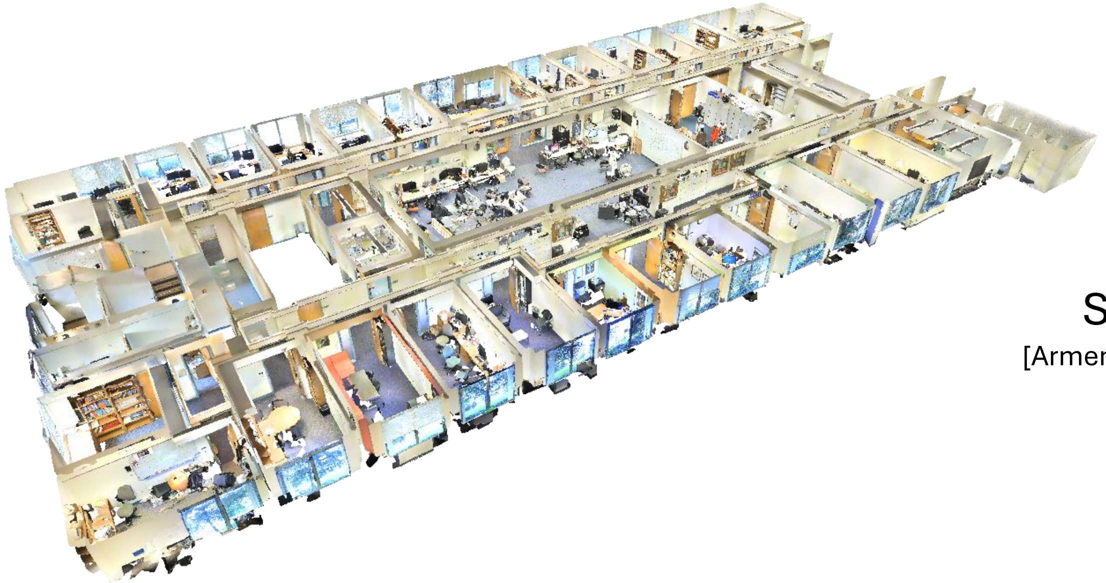
DeltaConv: Anisotropic Operators for Geometric Deep Learning on Point Clouds

Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, Klaus Hildebrandt
Computer Graphics and Visualization, TU Delft

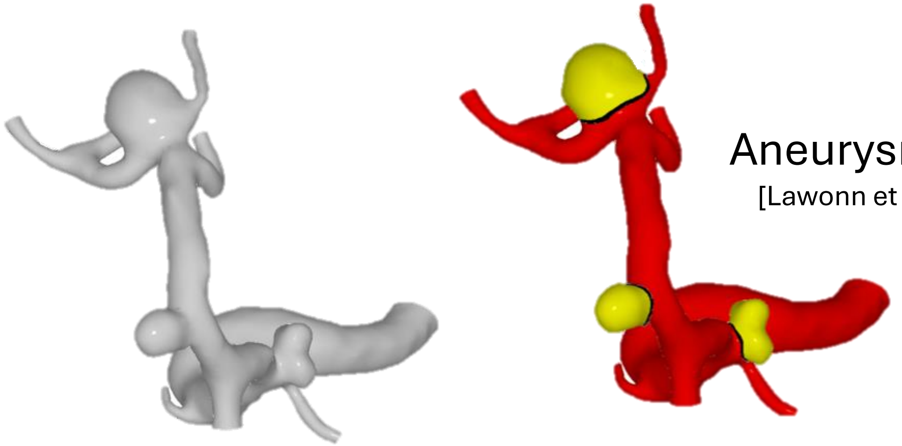
TU Delft

Lots of data on curved surfaces



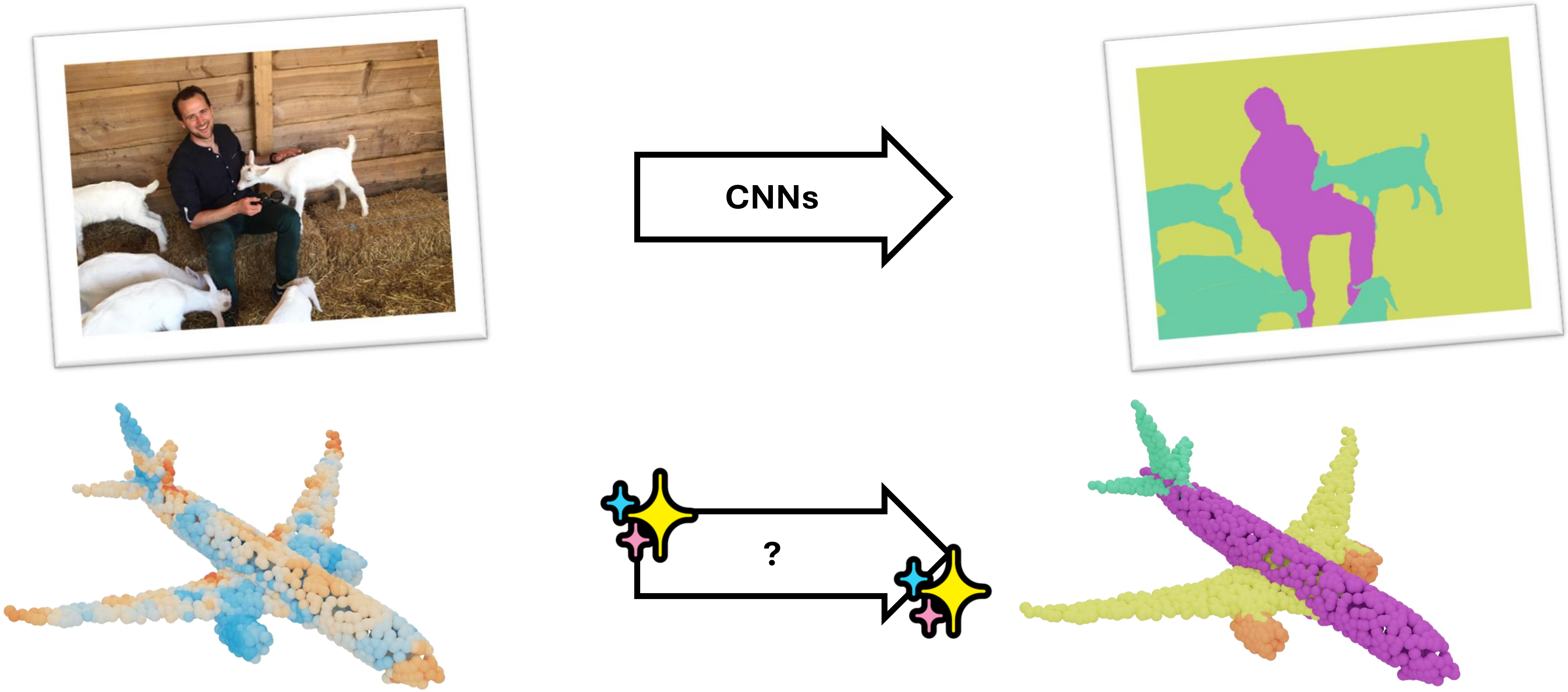
S3DIS
[Armeni et al. 2017]

Spot
Keenan Crane

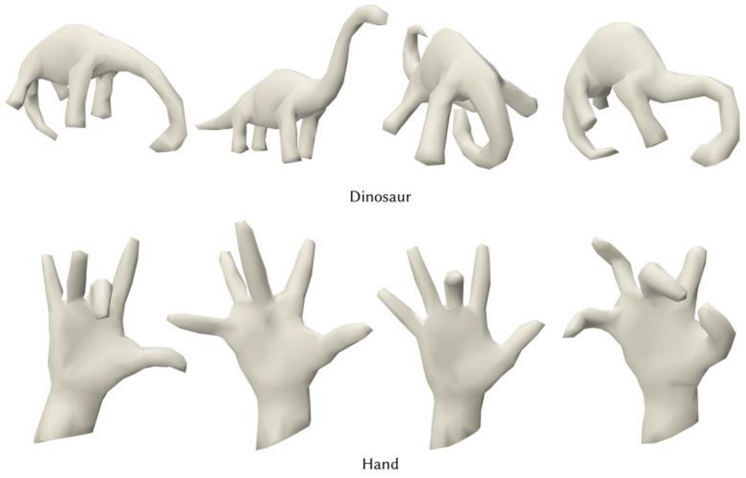


Aneurysm data
[Lawonn et al. 2019]

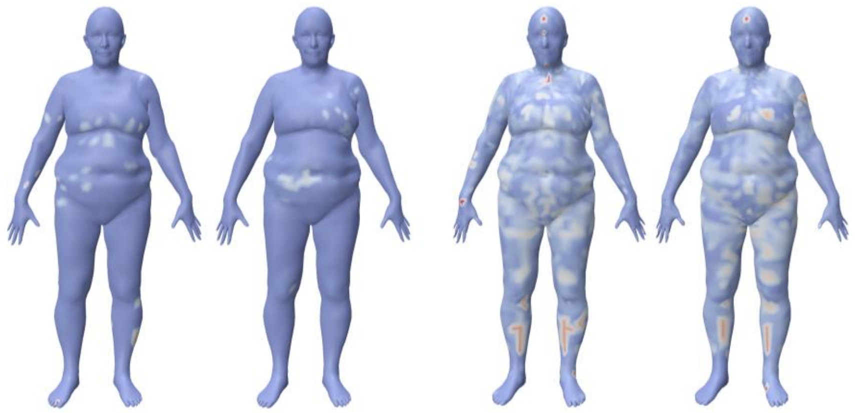
Neural networks on images



Applications for learning on surfaces



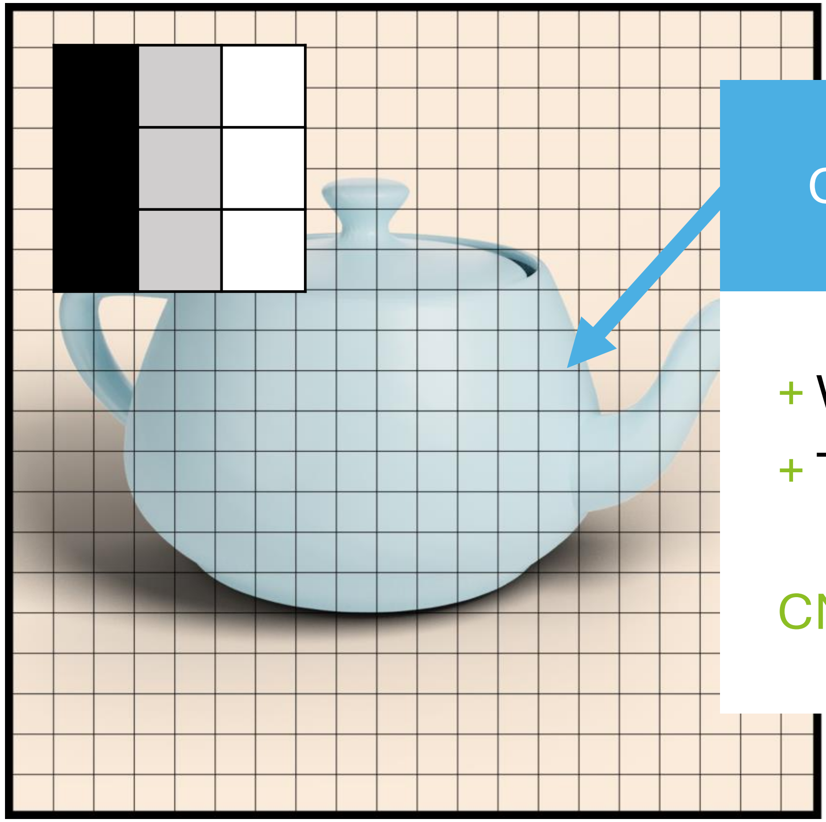
Classification
SHREC [Lian et al. 2011]



Correspondence
FAUST [Bogo et al. 2014]

Segmentation [Maron et al. 2017]

CNNs fit the data



Optimize kernel weights

+ Weight sharing (efficient)
+ Translation invariance

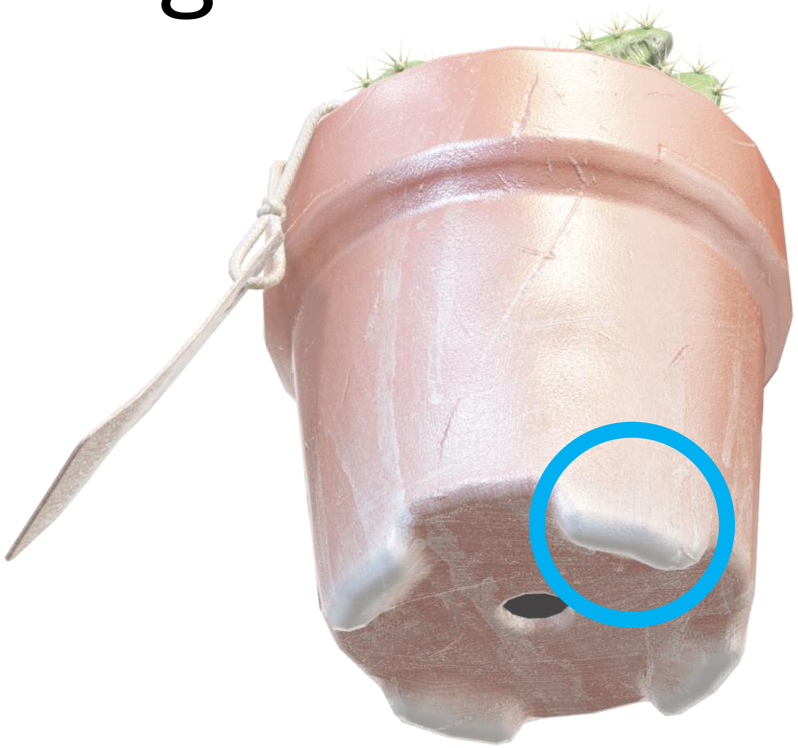
CNNs fit the data

We want the same for surfaces

Cactus courtesy of Sketchfab user vvc

We want the same for surfaces

Ridges



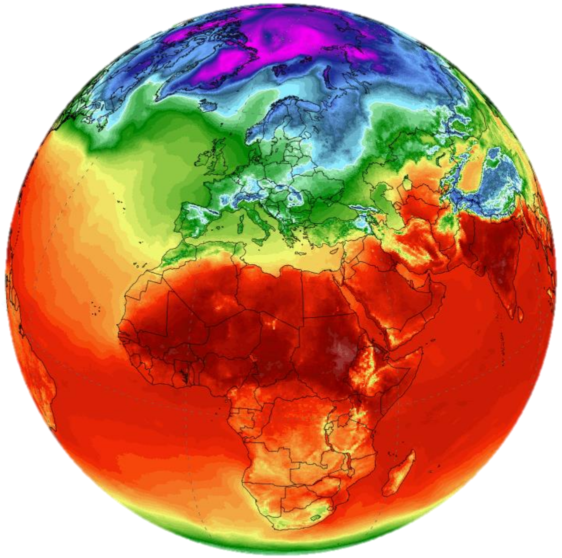
We want the same for surfaces

Corners

We want the same for surfaces

Patterns

We want the same for surfaces



Temperature

from Climate Reanalyzer (https://ClimateReanalyzer.org), Climate Change Institute, University of Maine, USA

Color

We want the same for surfaces

Convolutions
on surfaces

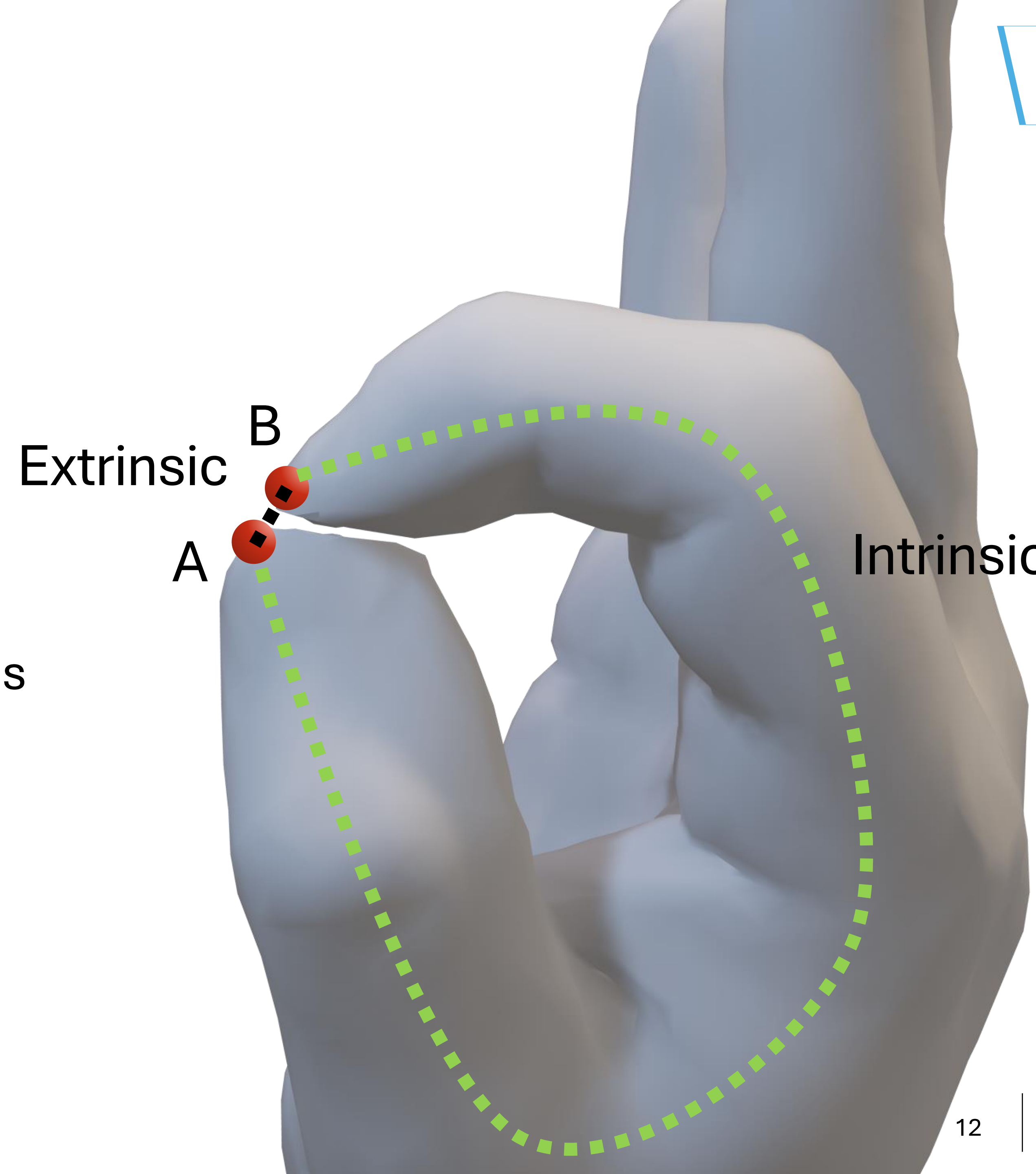
From 2D to 3D **intrinsically**

Intrinsic

+ Robust to isometric deformations

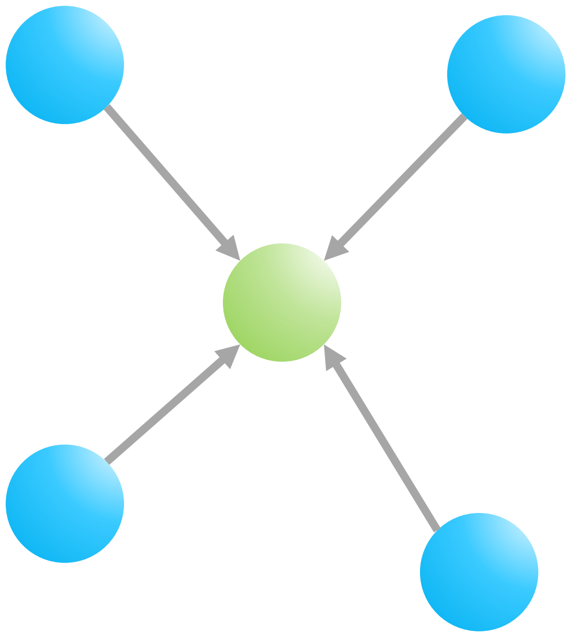
+ 2D instead of 3D

+ No/less distortion or occlusion



Learning on surfaces 101

- Convert mesh/point cloud to graph
- Vertices are nodes
- Edges to 1-ring (mesh) or neighborhood

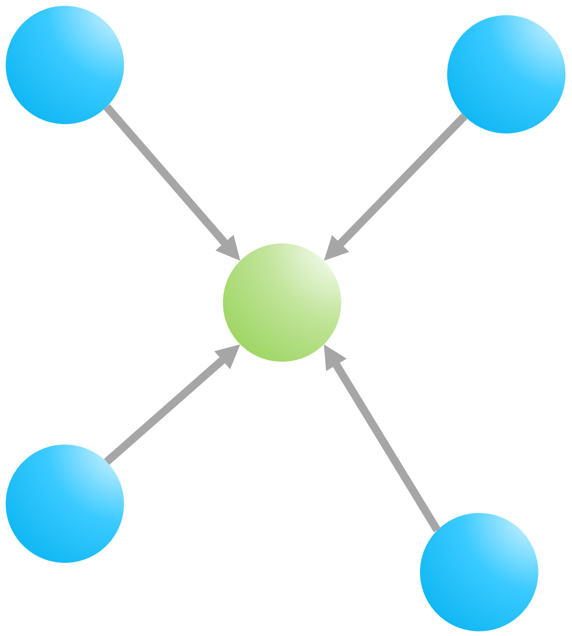


Graph-based learning

- Graph- and point based
  - GCN – Graph Laplacian

$$x_i' = \sigma(W_0 x_i + \sum_{j \in N_i} \frac{1}{c_{ij}} W_1 x_j)$$

[Kipf and Welling, 2016]

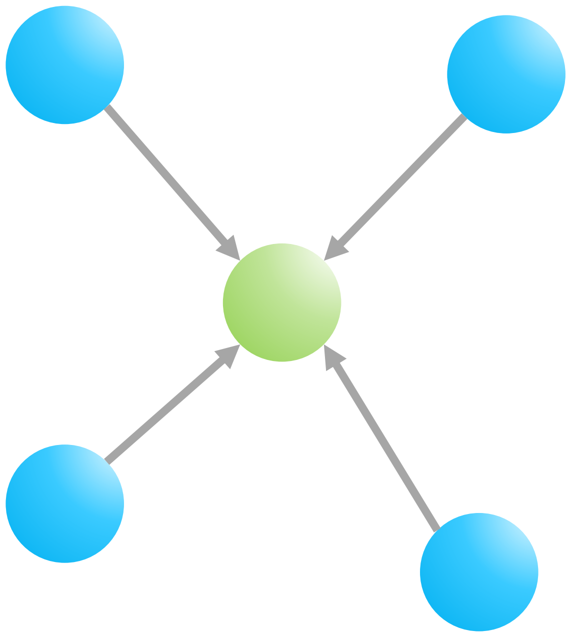


Graph-based learning

- Graph- and point based
  - GCN, PointNet++

$$x_i' = \max_{j \in N_i} h_\theta(x_j)$$

[Qi et al., 2017]

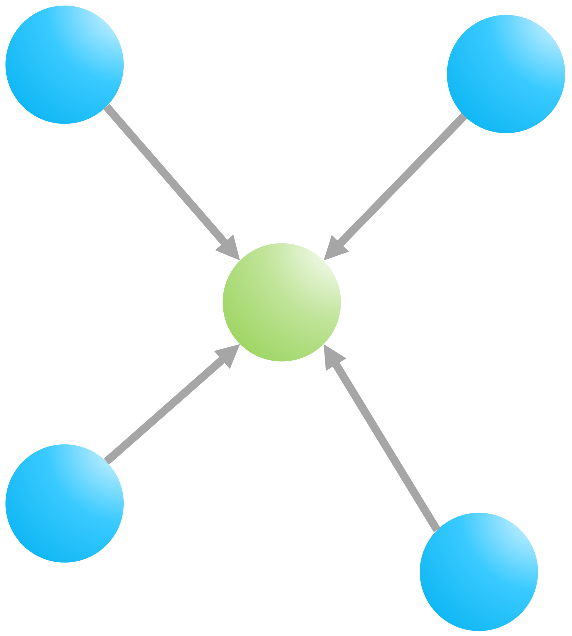


Graph-based learning

- Graph- and point based
  - GCN, PointNet++, EdgeConv

$$x_i' = \max_{j \in N_i} h_\theta(x_i, x_j - x_i)$$

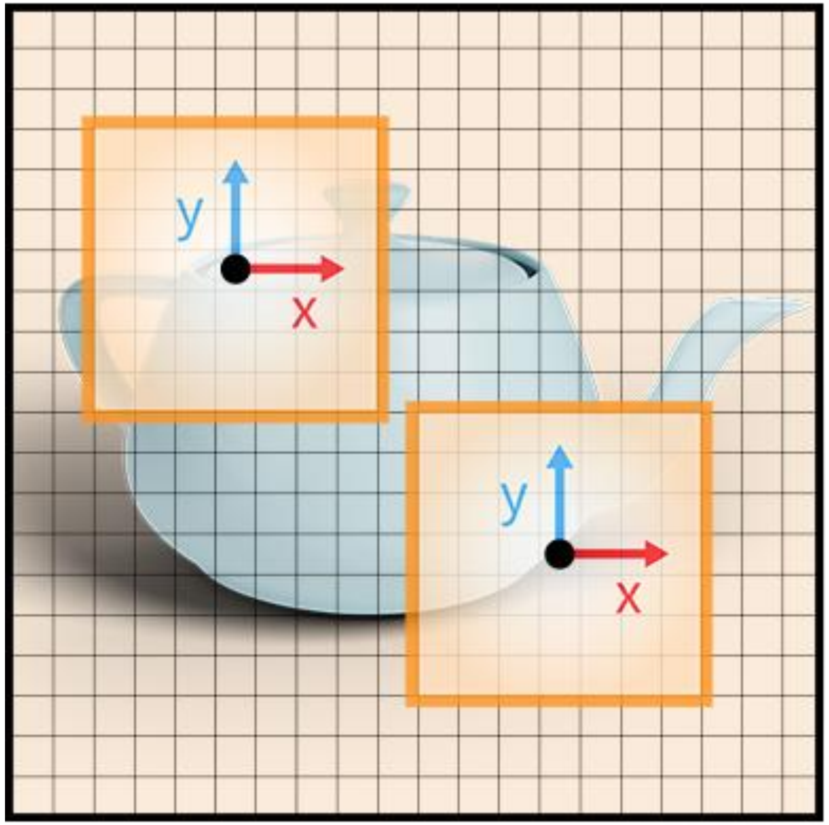
[Wang et al., 2019]



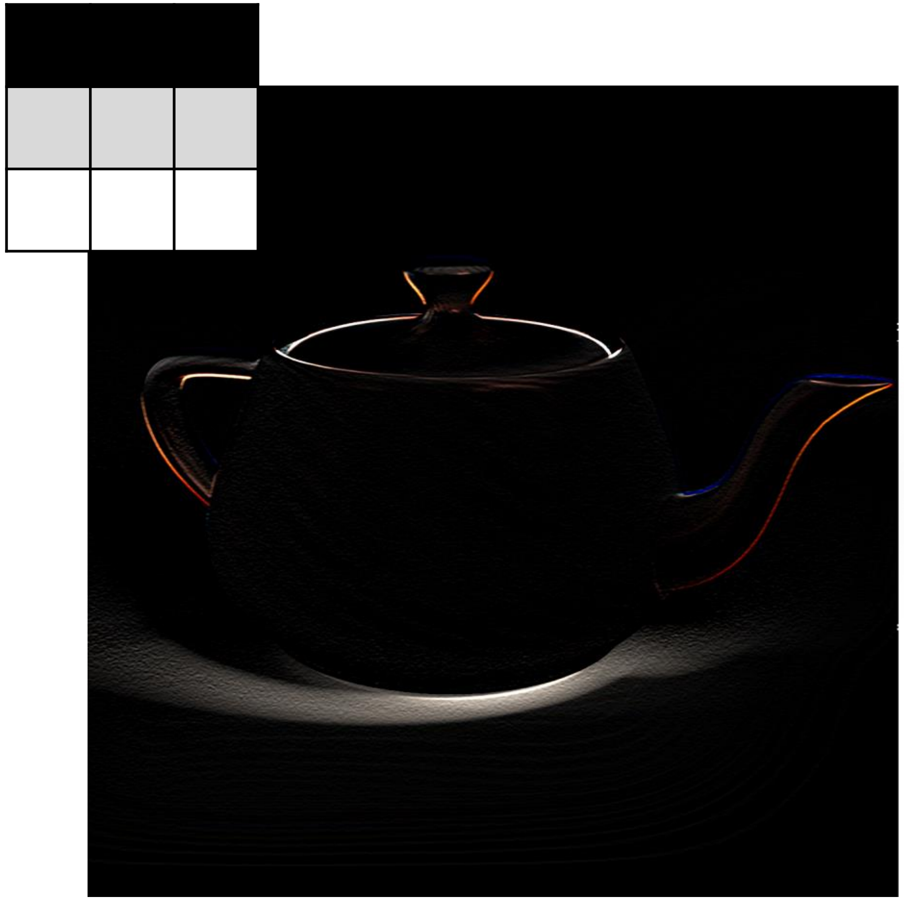
Our world is anisotropic

- Ridges, edges, corners
- Have a direction
  - a.k.a. they are anisotropic
- Anisotropic convolutions

Image CNNs can use a global coordinate system

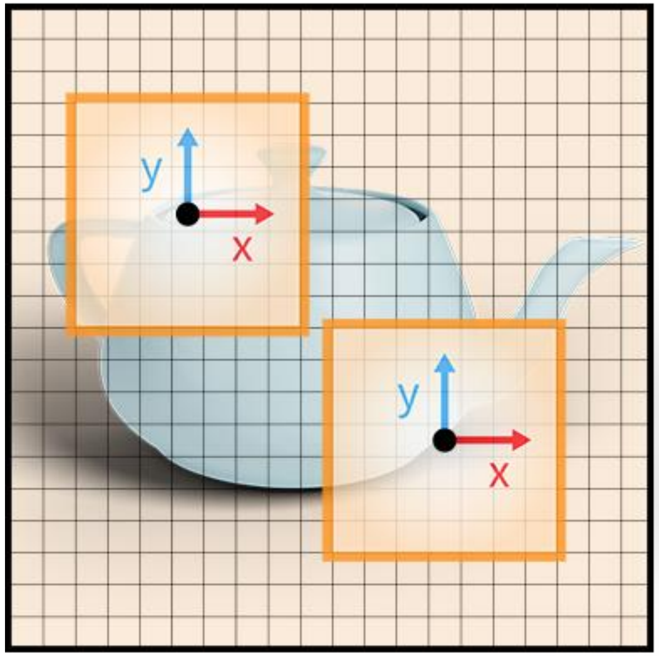


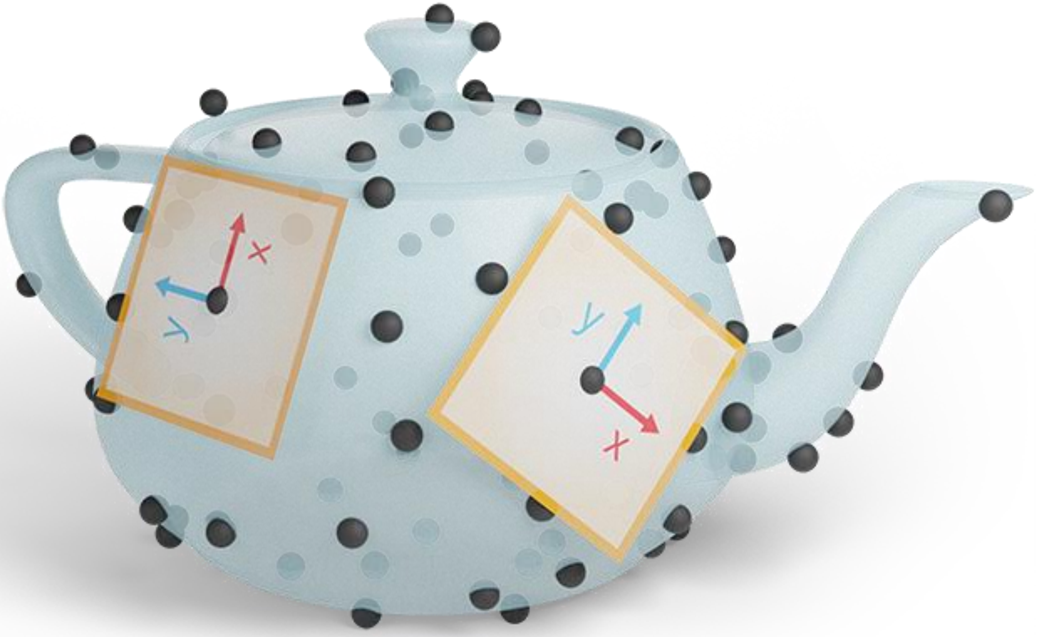
How does an image CNN use coordinates?



How does an image CNN use coordinates?

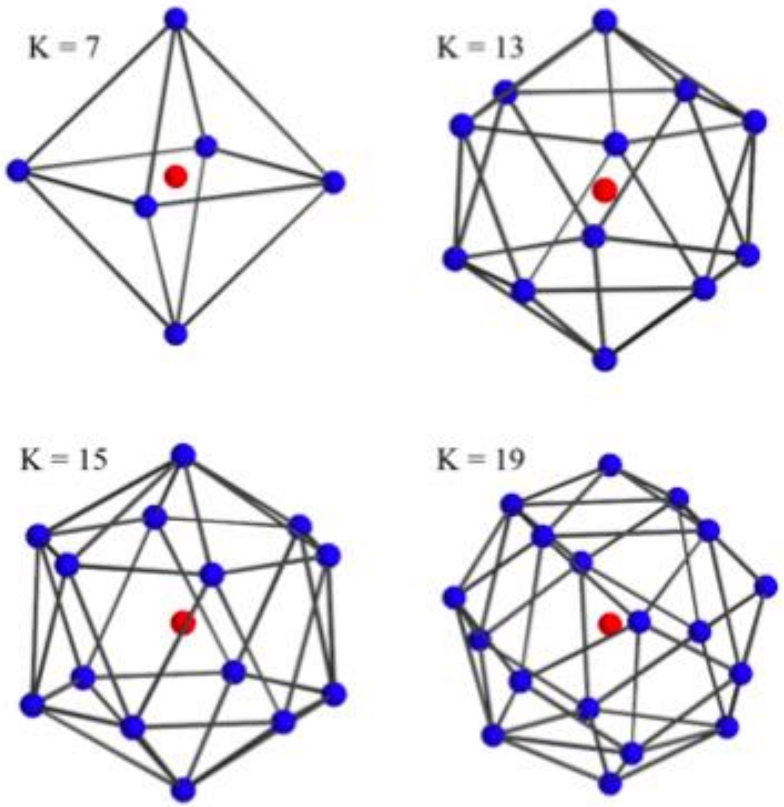
Surfaces have no global coordinate system





What did others do?

- Graph- and point based
  - GCN, PointNet++, EdgeConv
- 3D kernel (extrinsic)
  - KPConv, MinkowskiNet, SSCN

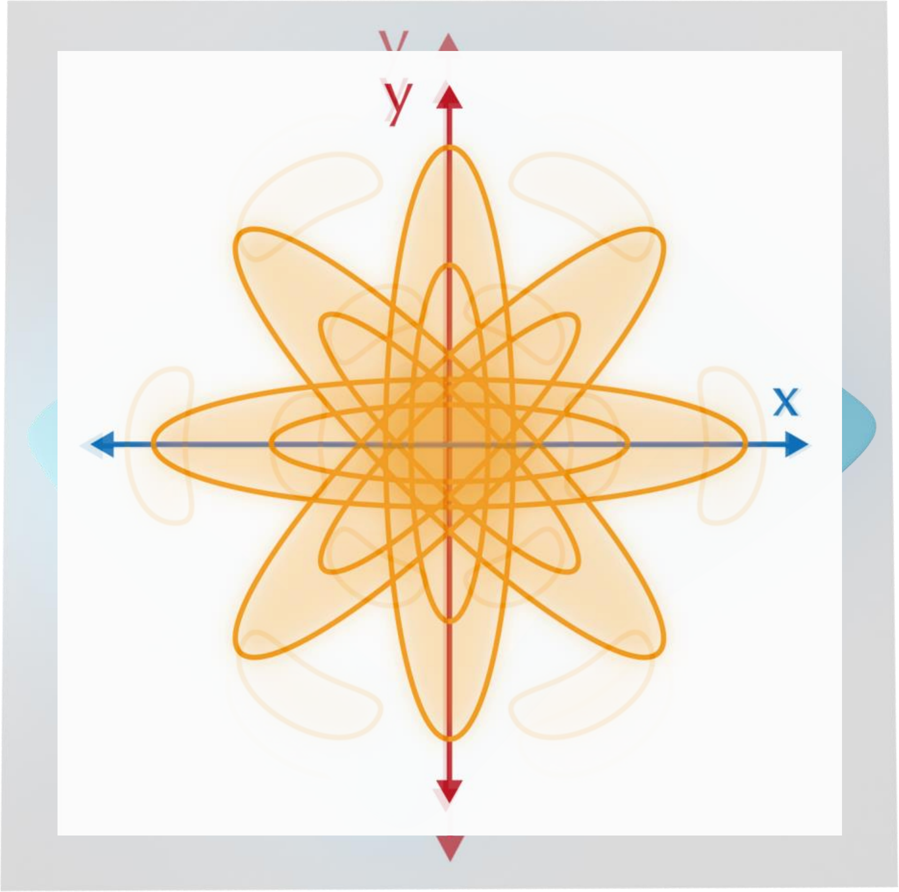


KPConv
[Thomas et al. 2019]

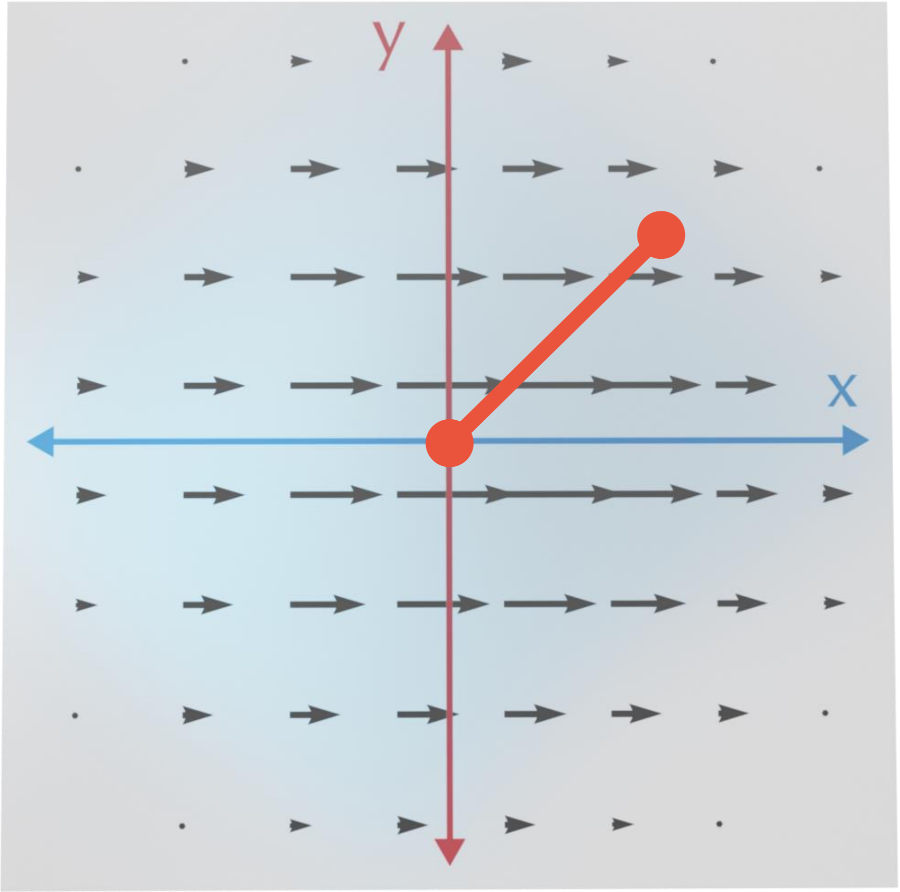
TU Delft

What did others do?

- Graph- and point based
  - GCN, PointNet++, EdgeConv
- 3D kernel (extrinsic)
  - KPConv, MinkowskiNet, SSCN
- 2D kernels *on* surfaces (intrinsic)
  - GCNN, ACNN, MoNet, MDGCNN, **HSN**

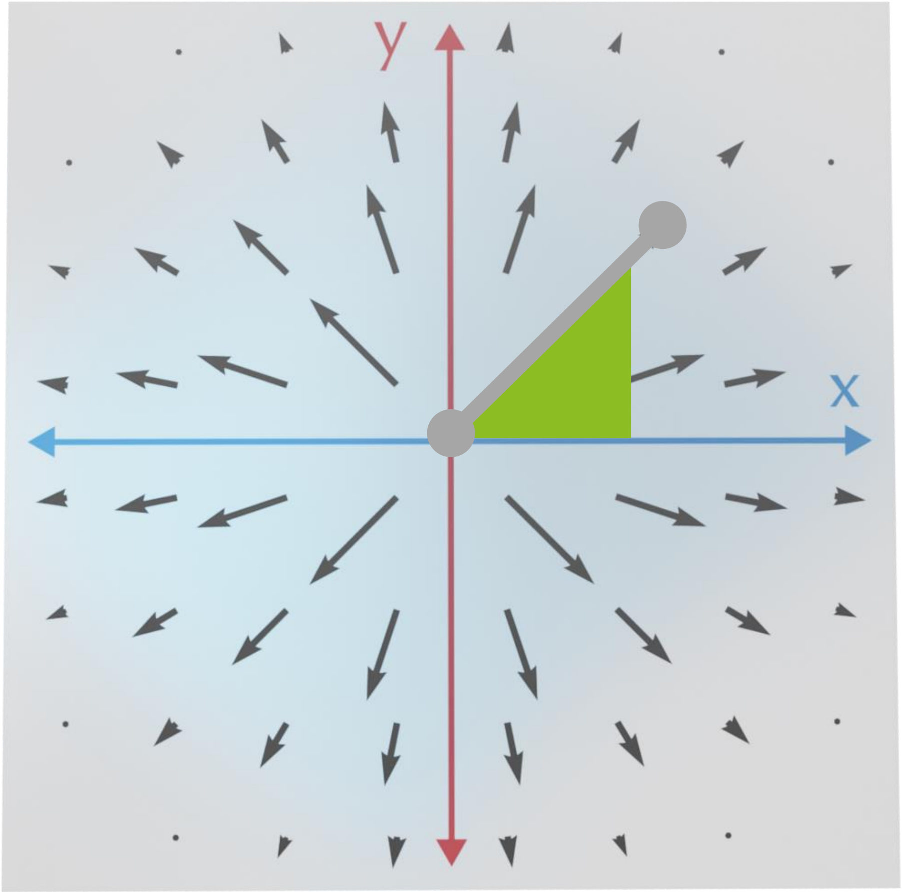


$R(r)e^{i\beta}$



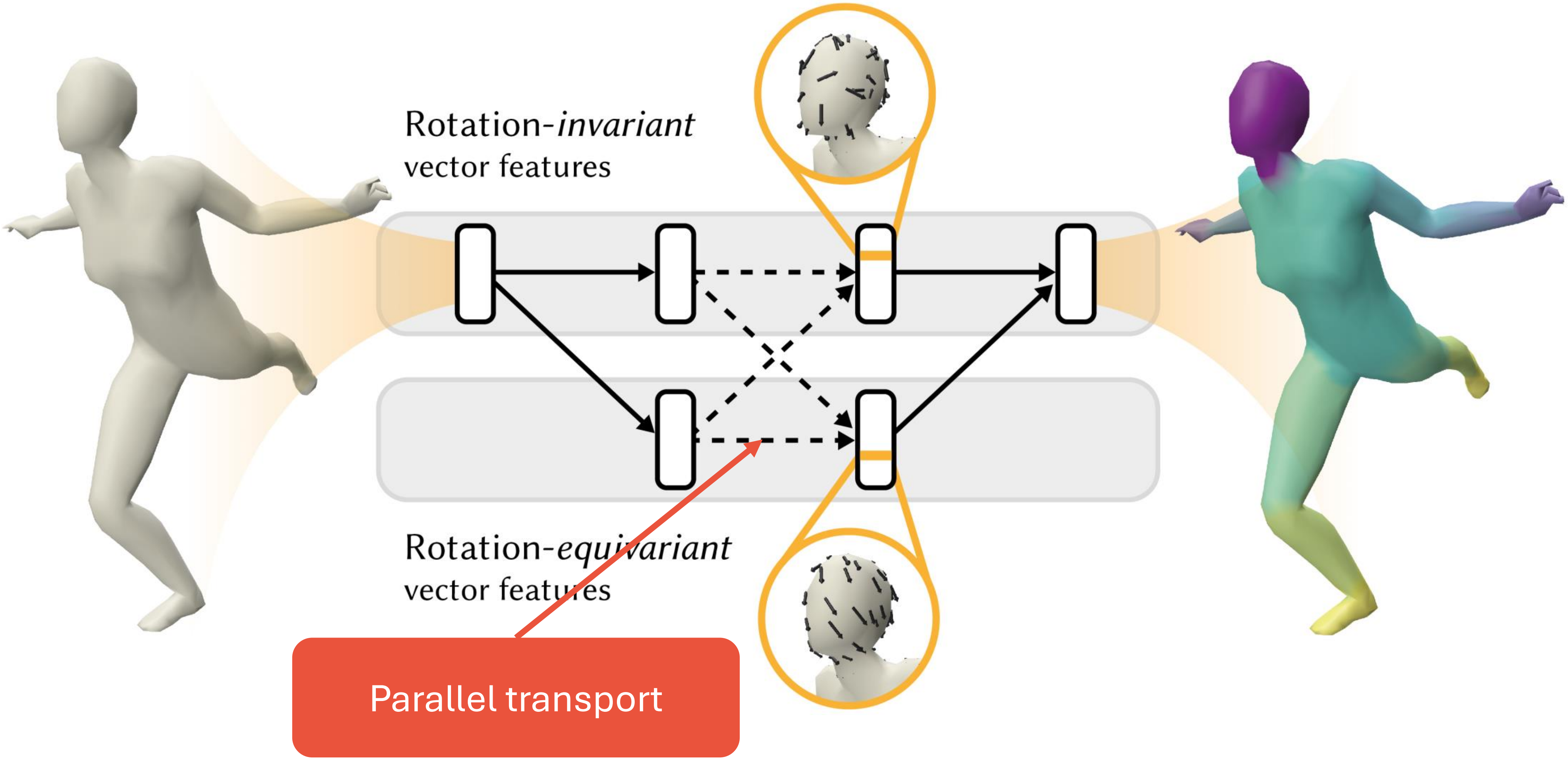
Rotation invariant

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



Rotation equivariant

Harmonic Surface Networks



Limitations

- Requires a good exponential map (can be expensive to compute, tricky)
- Circular harmonics are expensive to evaluate
- Can we simplify?

DeltaConv

SIGGRAPH 2022

Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, Klaus Hildebrandt

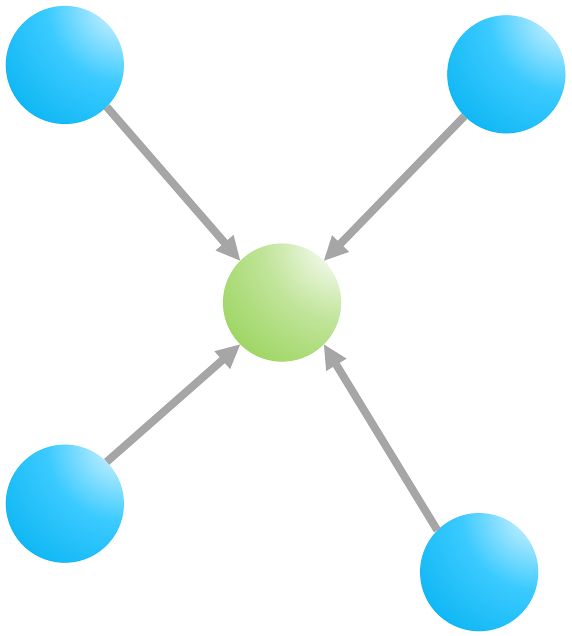
TU Delft

Laplacians in Geometric Deep Learning

- GCN – Graph Laplacian
- DiffusionNet – Laplace-Beltrami

$$x_i' = \sigma\left(W_0 x_i + \sum_{j \in N_i} \frac{1}{c_{ij}} W_1 x_j\right)$$

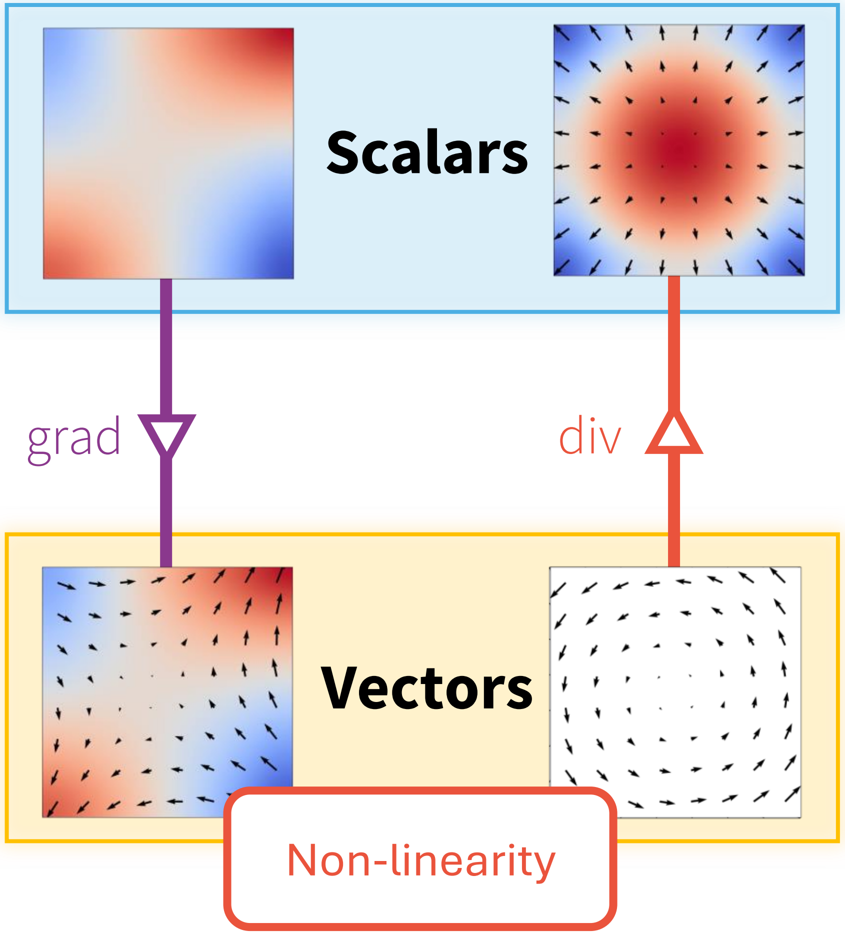
[Kipf and Welling, 2016]



Coordinate-independent

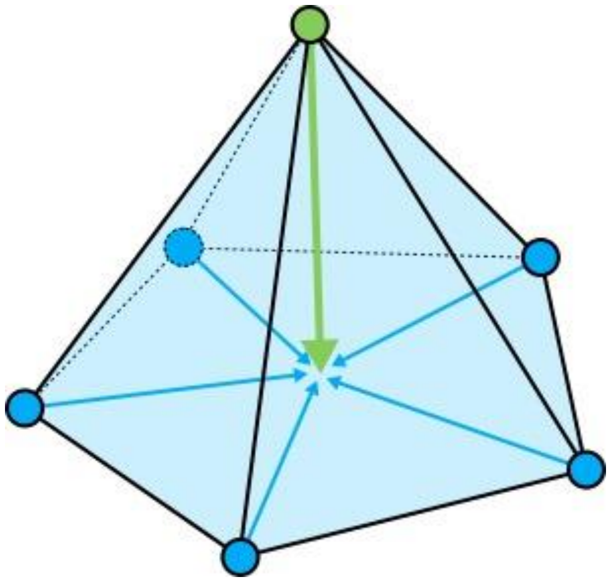
Isotropic

Making the Laplacian anisotropic



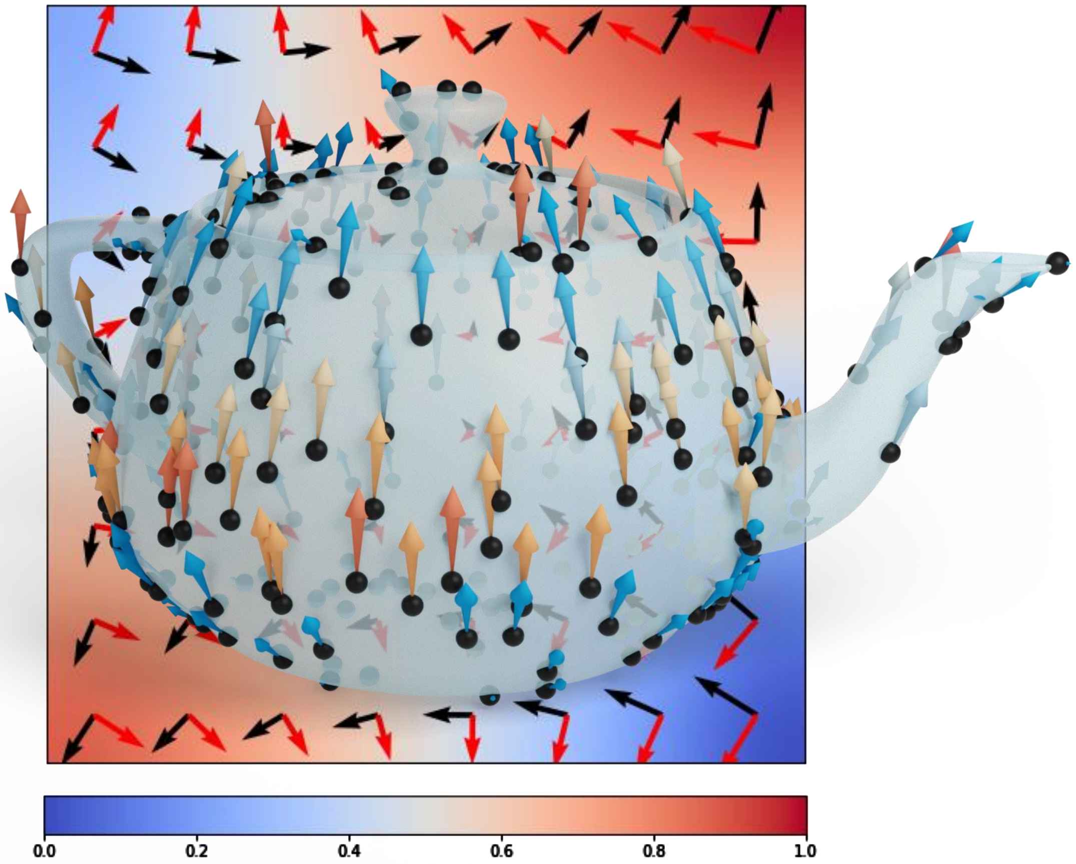
Laplacian

- Sum of second derivatives
  - Discrete setting: Difference to average of neighbors
- Used in many applications
  - Harmonic functions $\Delta y = 0$
  - Heat equation $\frac{\partial y}{\partial t} = \Delta y$
  - Spectral analysis (eigendecomposition)
- Isotropic



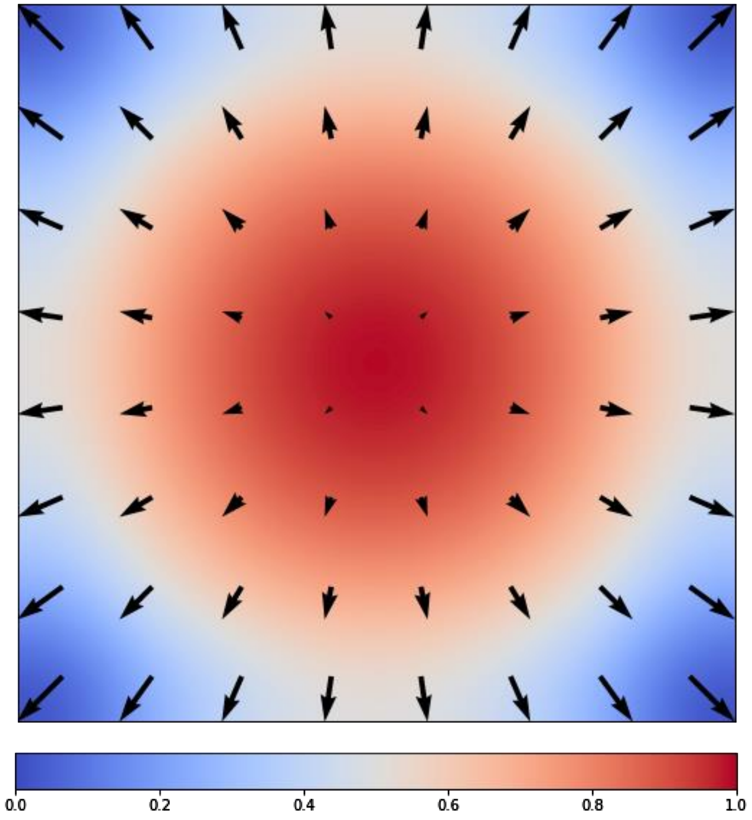
Gradient, co-gradient

- Largest rate of change + direction
- Co-gradient
- On surfaces: tangential

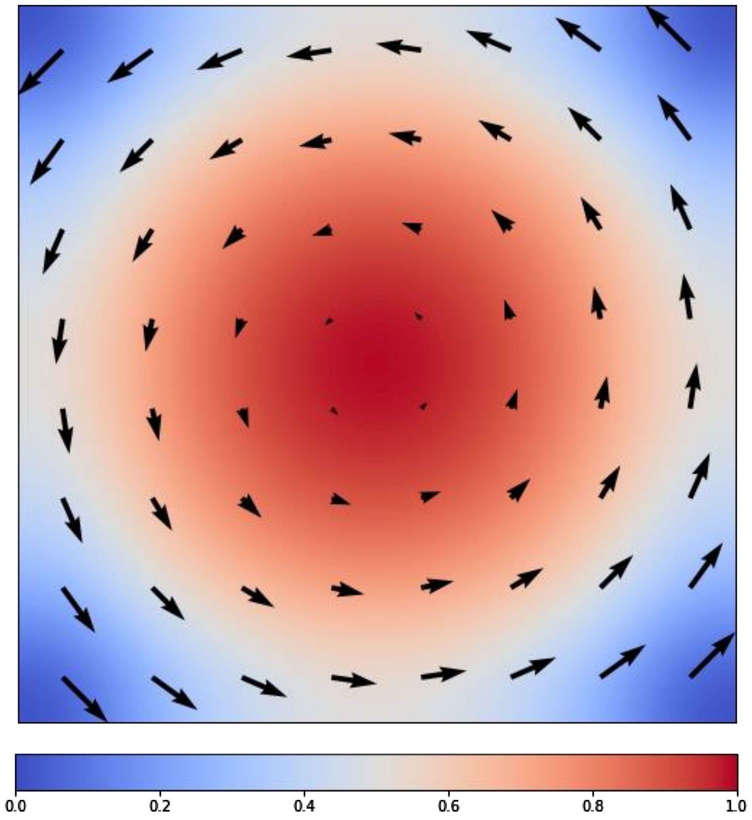


Divergence, curl

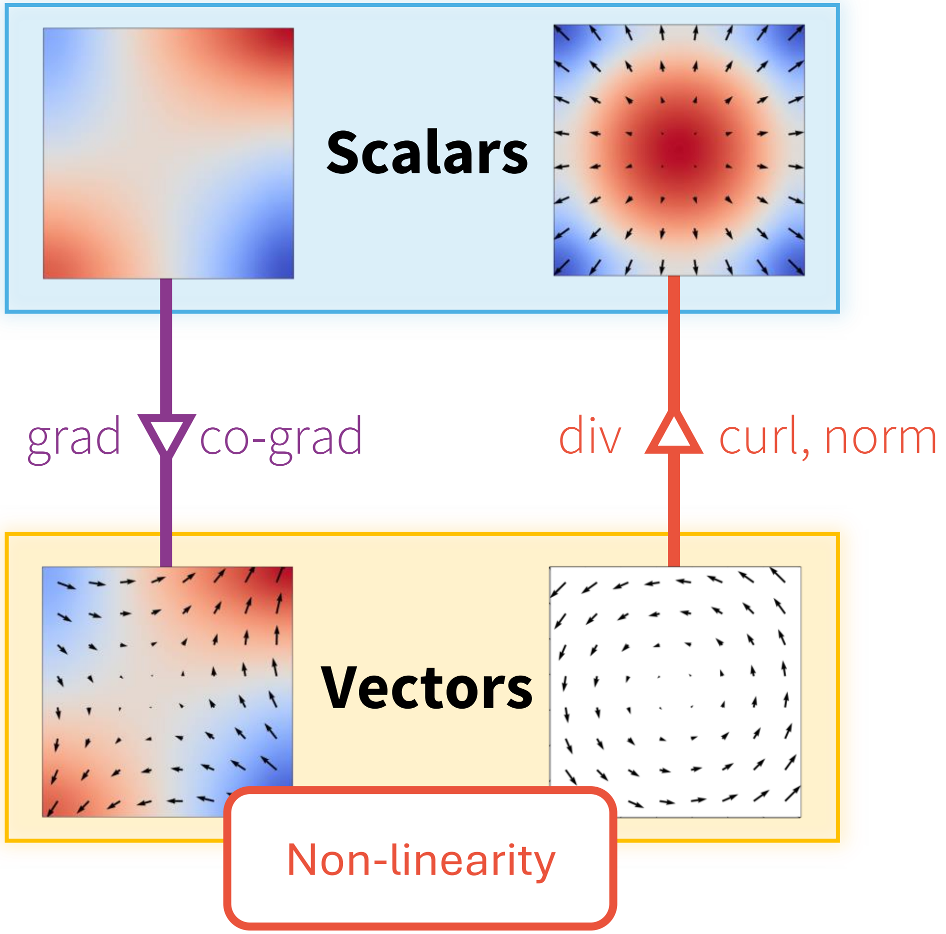
div Sinks and sources



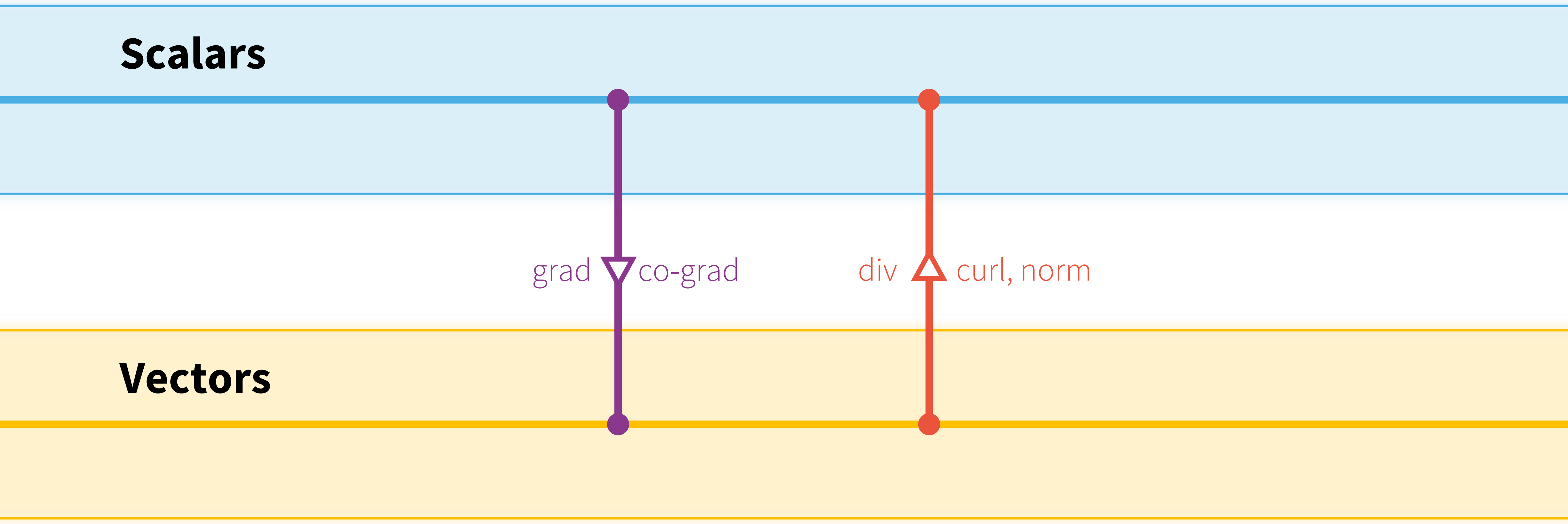
curl Vortices



Making the Laplacian anisotropic

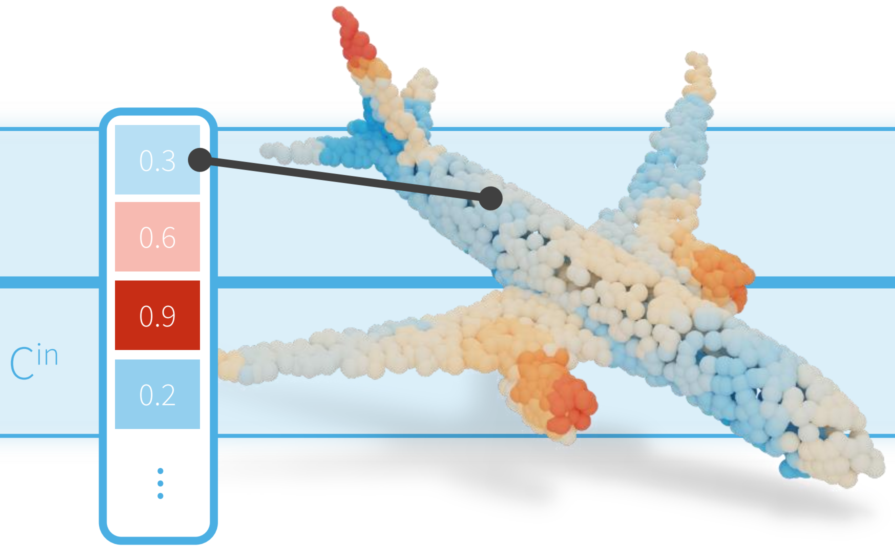


Scalar and vector streams



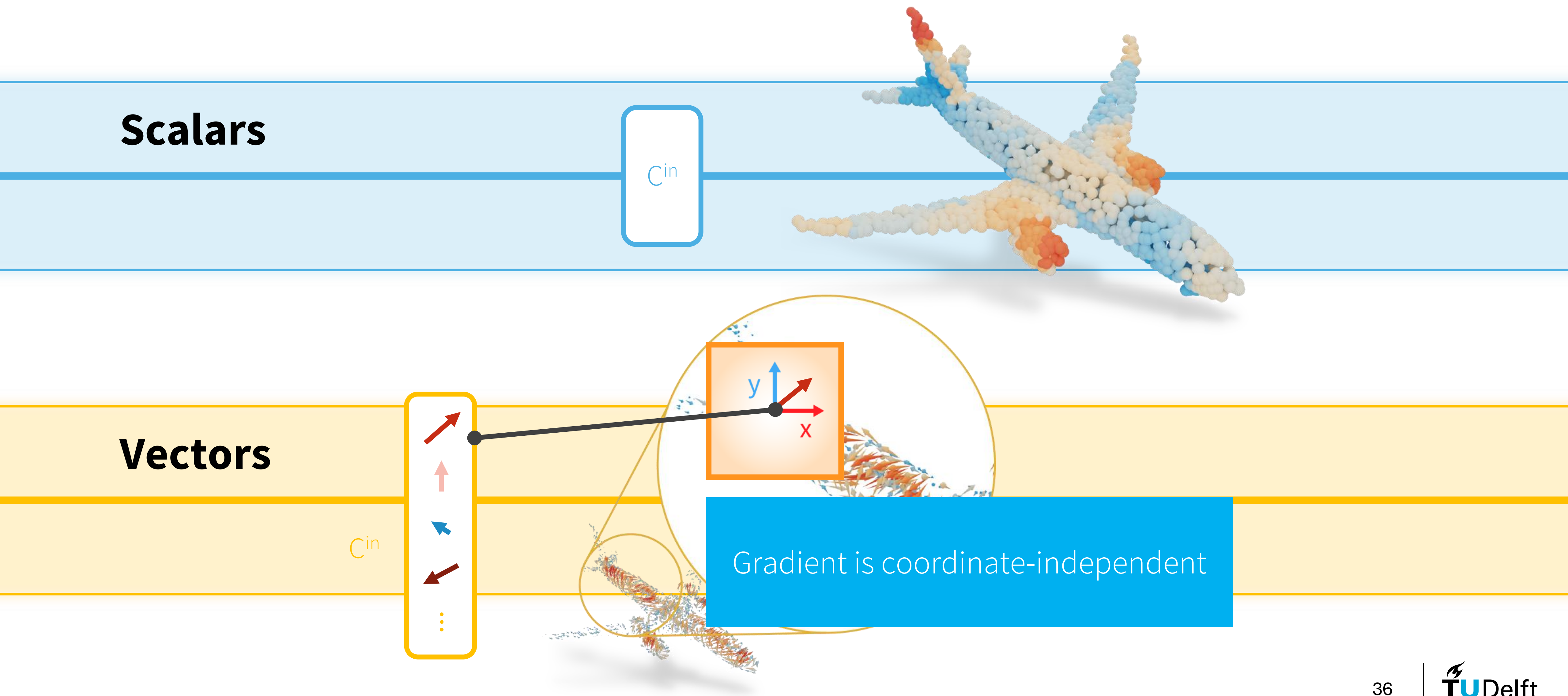
Building a feed-forward neural network

Scalars



Vectors

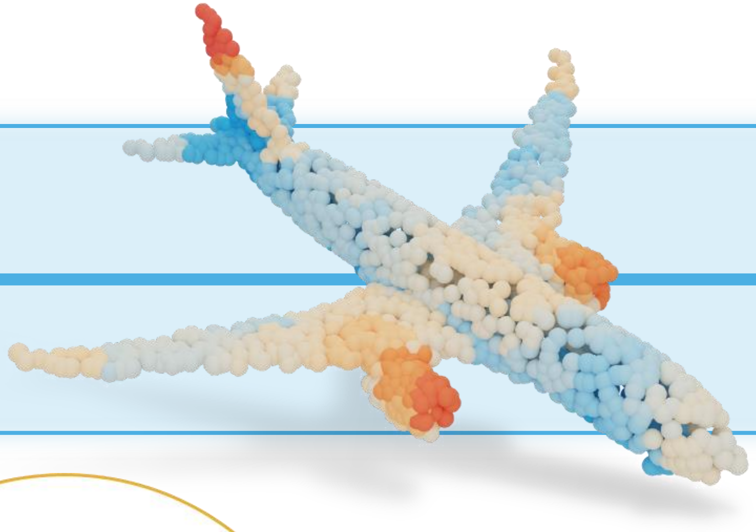
Building a feed-forward neural network



Building a feed-forward neural network

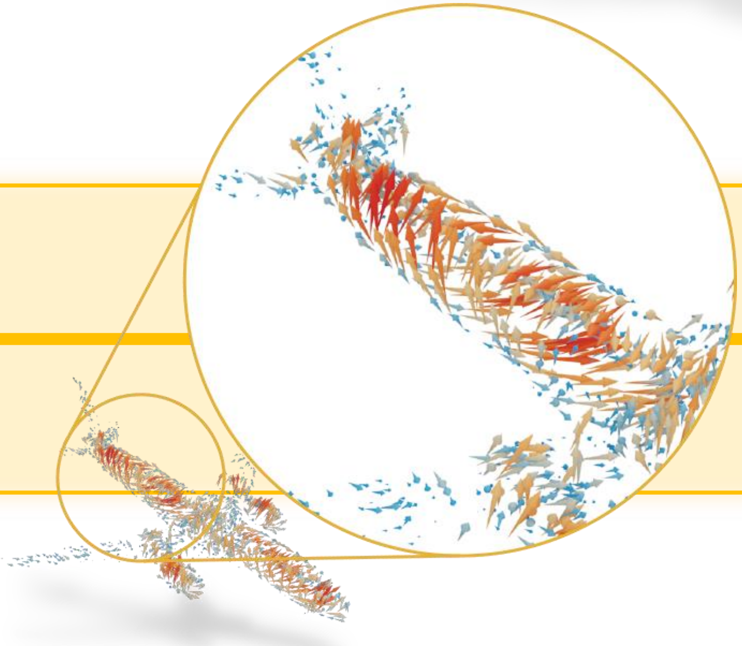
Scalars

C^{in}

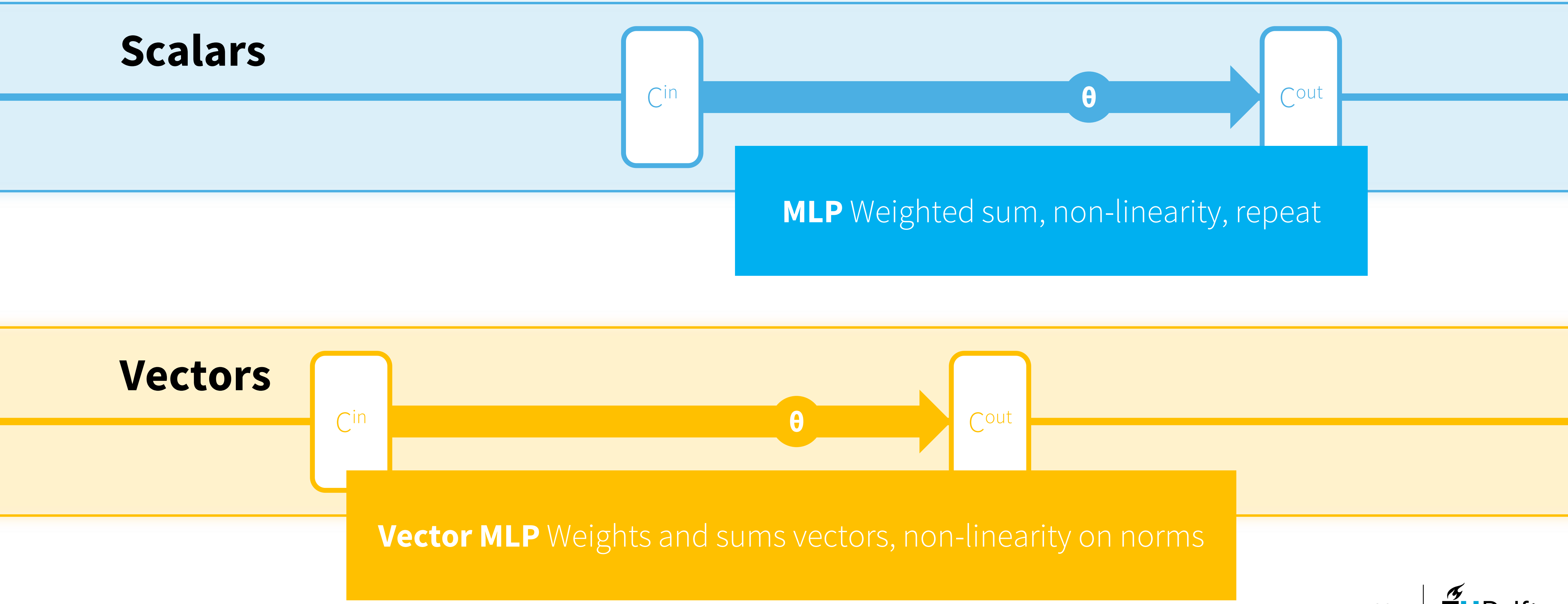


Vectors

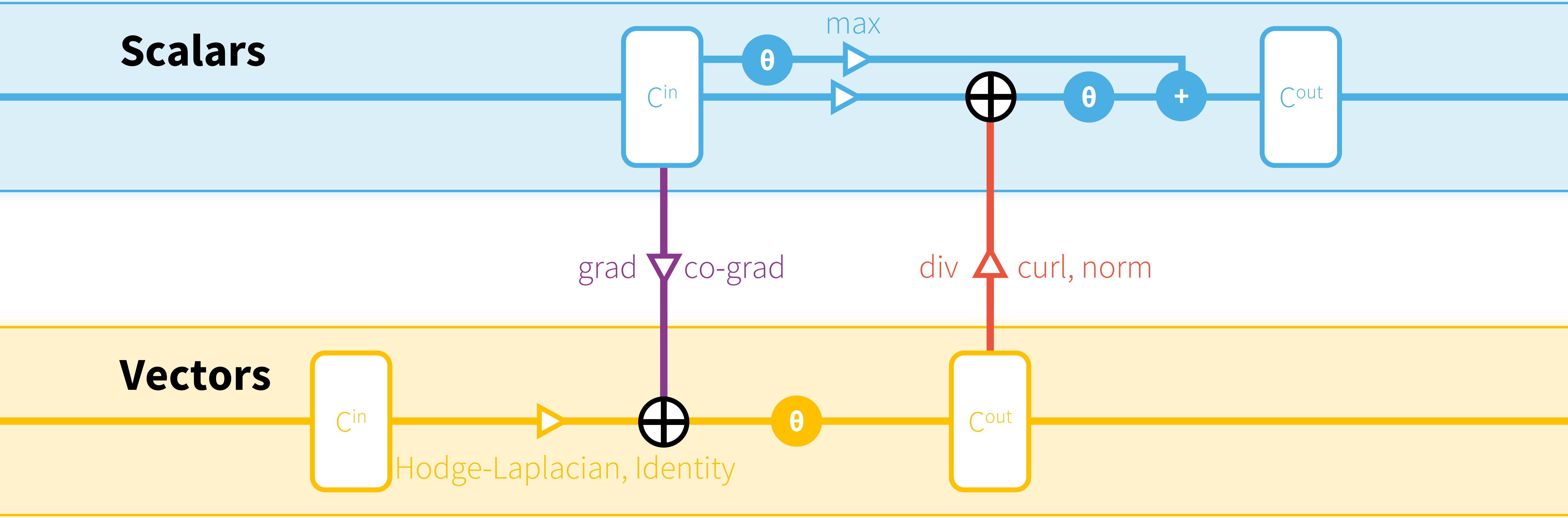
C^{in}



Building a feed-forward neural network



DeltaConv combines and composes operators



Properties of DeltaConv

Anisotropy?

- Anisotropic diffusion [Perona-Malik, 1987]

$$\frac{\partial y}{\partial t} = \nabla \cdot (c(|\nabla y|)\nabla y)$$

Anisotropy?

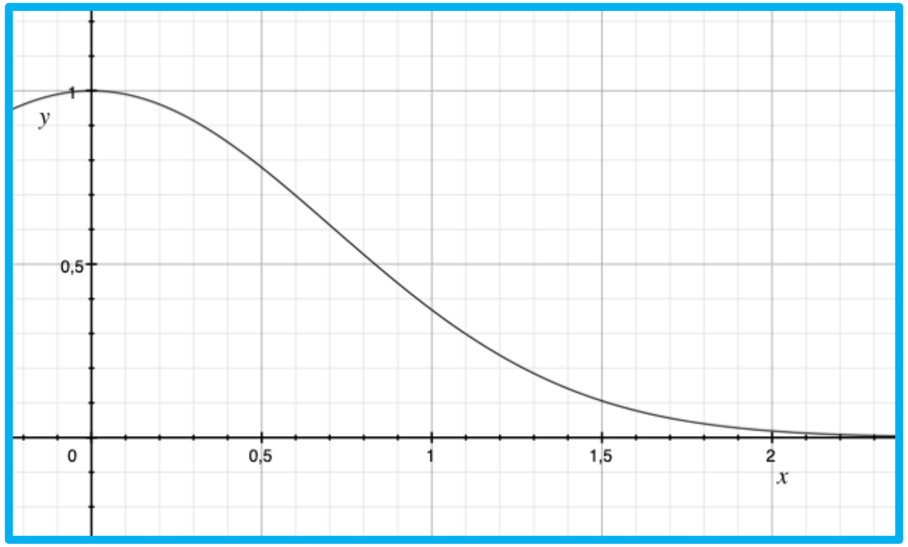
- Anisotropic diffusion [Perona-Malik, 1987]

$$\frac{\partial y}{\partial t} = \nabla \cdot (c(|\nabla y|)\nabla y)$$

Anisotropy?

- Anisotropic diffusion [Perona-Malik, 1987]

$$\frac{\partial y}{\partial t} = \nabla \cdot (c(|\nabla y|)\nabla y)$$



Anisotropy?

- Anisotropic diffusion [Perona-Malik, 1987]

$$\frac{\partial y}{\partial t} = \nabla \cdot (c(|\nabla y|)\nabla y)$$

Anisotropy?

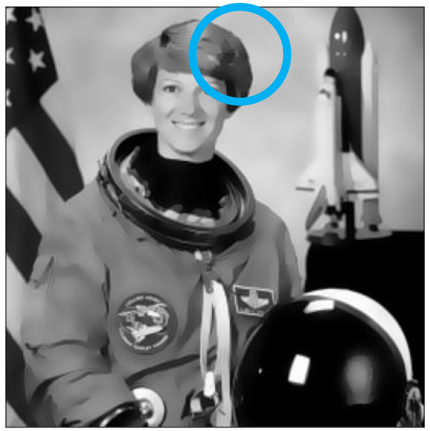
- Anisotropic diffusion [Perona-Malik, 1987]

$$\frac{\partial y}{\partial t} = \nabla \cdot (c(|\nabla y|)\nabla y)$$

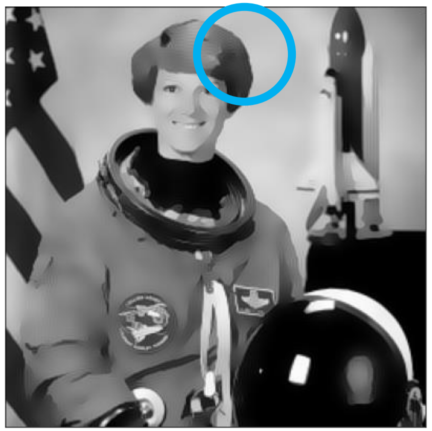
- Solve by explicit integration over time

Anisotropy?

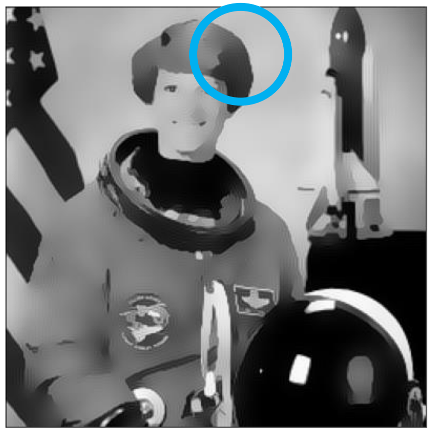
1 step



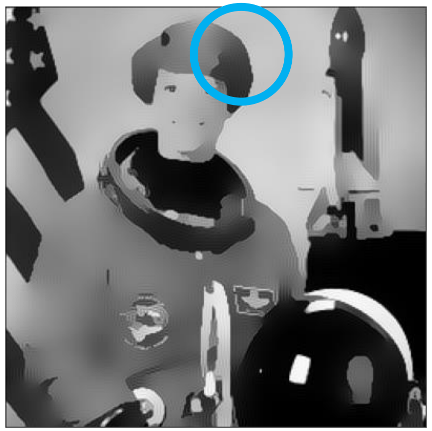
5 steps



10 steps



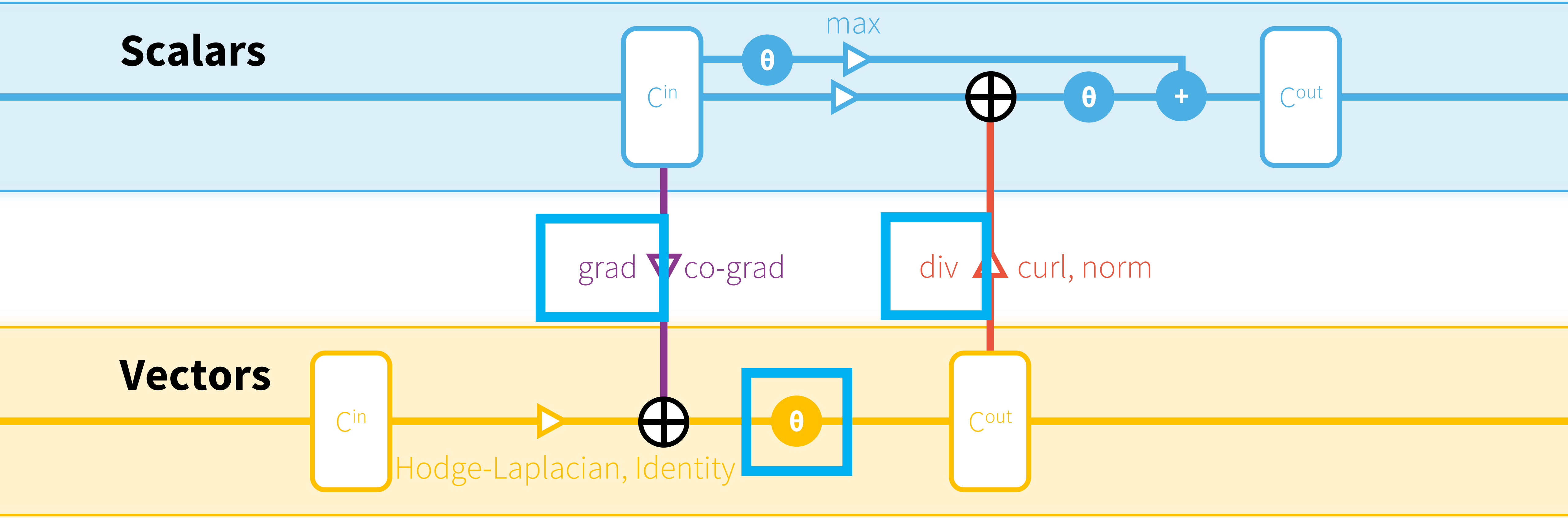
20 steps



40 steps

Astronaut image courtesy NASA

DeltaConv combines and composes operators



Ablations Perona-Malik

Input

Target

20 timesteps

DeltaConv can control anisotropic diffusion time

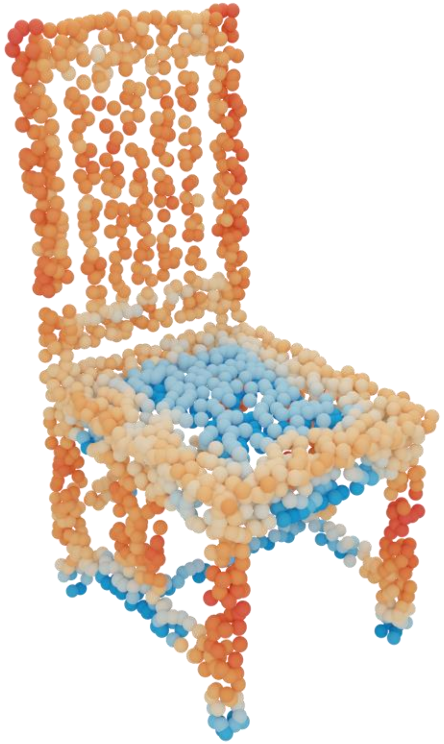
Target 5 steps

DeltaConv benefits from geometric operators

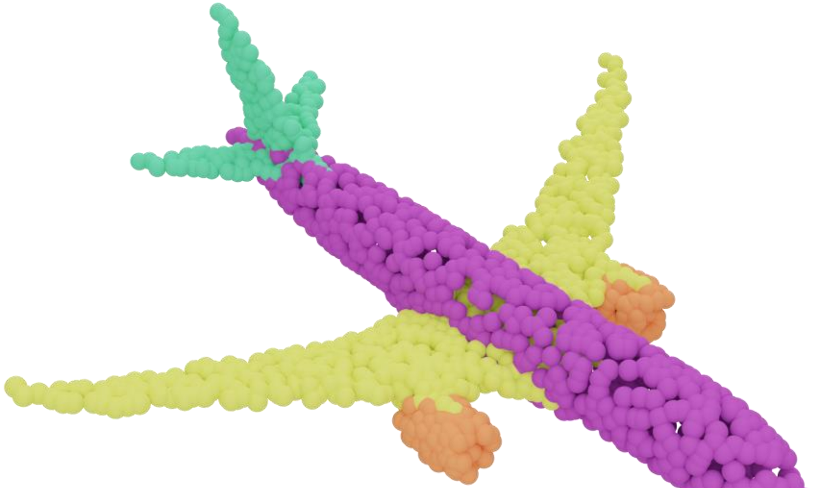
- All operators are **intrinsic**
- All operators are **coordinate-independent**
- All operators are **generalizable**
  - Hyperbolic space, higher dimensions
- All operators are available for **different discretizations**
  - In paper: images, point clouds

Experiments

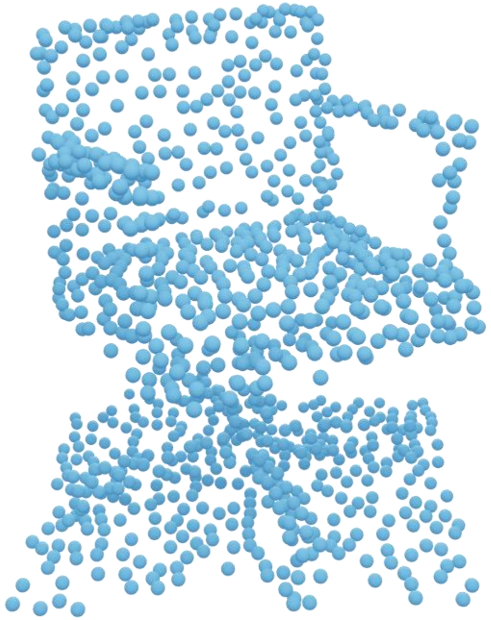
Comparisons – Point Clouds



Classification
ModelNet40

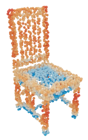


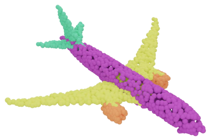
Segmentation
ShapeNet

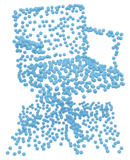


Classification
ScanObjectNN

Comparisons



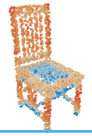


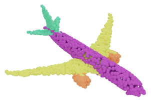


	Classification ModelNet40		Segmentation ShapeNet	Classification ScanObjectNN
Method	**Mean Class Acc.**	**Overall Acc.**	**Mean inst. mIoU**	**Accuracy**
PointNet++	-	90.7	85.1	77.9
DGCNN	90.2	92.9	85.2	78.1
KPConv rigid	-	92.9	86.4	-
PointTransformer	90.6	93.7	86.6	-
GBNet	91.0	**93.8**	-	80.5
CurveNet	-	**93.8**	86.8	-
DeltaNet (ours)	**91.2**	**93.8**	86.6	**84.7**
Delta-U-ResNet(ours)	**-**	**-**	**86.9**	

More comparisons and citations in paper

Ablations vector stream

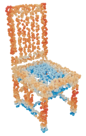




			Classification ModelNet40		Segmentation ShapeNet
Scalar convolution	**Vector stream**	**Increased params**	**Mean Class Acc.**	**Overall Accuracy**	**Mean inst. mIoU**
Laplace-Beltrami	-	-	86.1	90.4	82.5
GCN	-	-	87.3	90.4	81.1
Max aggregation	-	-	89.2	92.2	85.7

Ablations speed

- Compare with EdgeConv (no dynamic graph)
- Points instead of edge-based features
  - K times more MLP computations (e.g., k=20)

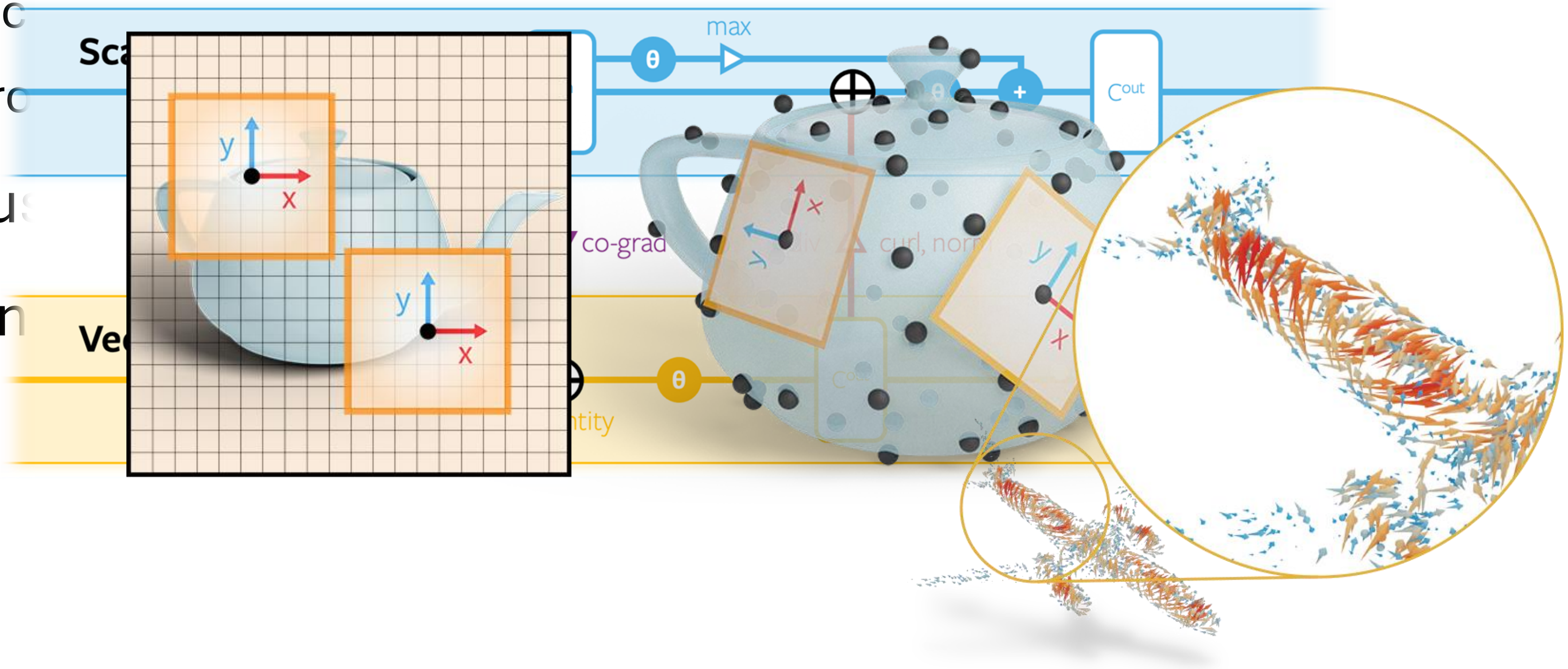


Classification
ModelNet40

Convolution	Data Transforms	Training	Backward	Inference
EdgeConv	k-nn	196ms	147ms	186ms
DeltaConv (lapl.)	k-nn + ops	80ms	5ms	80ms
DeltaConv	k-nn + ops	130ms	60ms	125ms

Conclusion

- Intrinsic anisotropic convolutions are challenging on surfaces
- DeltaConv combines and composes geometric operators
  - Intrinsic
  - Anisotro
- Easy to u
- Builds on



Try DeltaConv yourself

github.com/**rubenwiersma/deltaconv**

```
$ pip install deltaconv
```

graphics.tudelft.nl

Thank you!