Emerging Directions in Deep Learning

Generalizing convolutions - approaches in the spatial domain

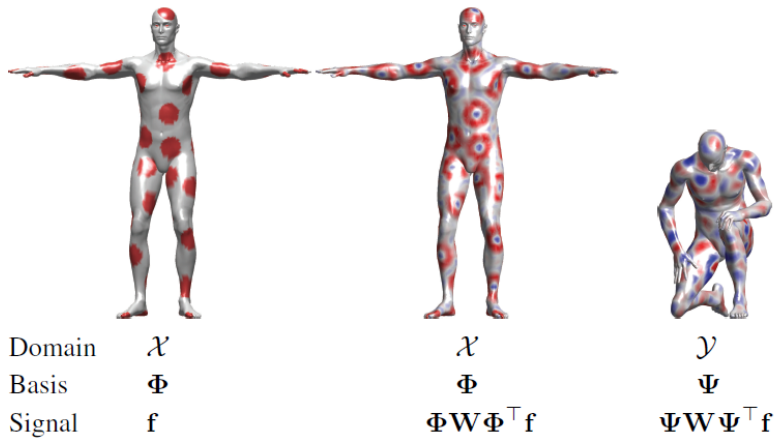
Mihai-Sorin Stupariu, 2025-2026

CNNs in the Euclidean and non-Euclidean framework

- On an Euclidean domain: the convolution can be interpreted as passing a template at each point of the domain and recording the correlation of the template with the function at that point (due to the shift-invariance).

CNNs in the Euclidean and non-Euclidean framework

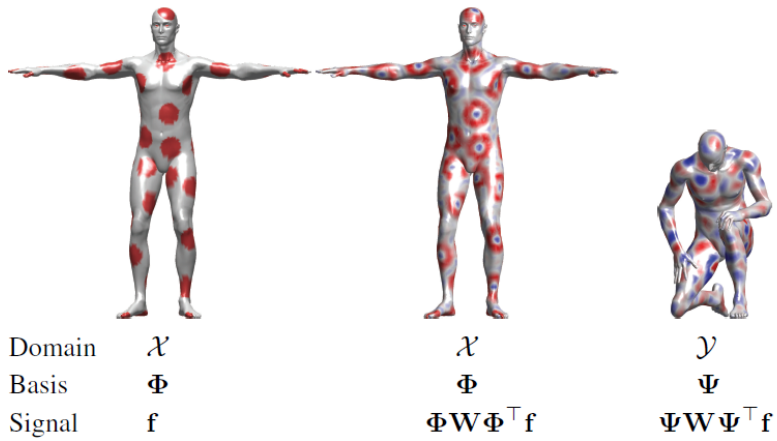
- On an Euclidean domain: the convolution can be interpreted as passing a template at each point of the domain and recording the correlation of the template with the function at that point (due to the shift-invariance).
- The spectral approach (convolution defined in the spectral domain) has a shortcoming: "locally defined" and the model cannot be adapted across different domains.



Source: Bronstein et al., 2017

CNNs in the Euclidean and non-Euclidean framework

- On an Euclidean domain: the convolution can be interpreted as passing a template at each point of the domain and recording the correlation of the template with the function at that point (due to the shift-invariance).
- The spectral approach (convolution defined in the spectral domain) has a shortcoming: "locally defined" and the model cannot be adapted across different domains.

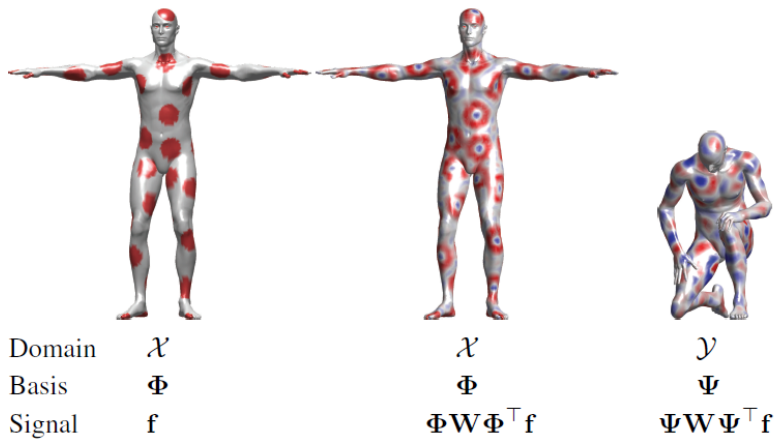


Source: Bronstein et al., 2017

- Another challenge: no immediate way to define a meaningful spatial pooling operation: (i) the number of points may vary; (ii) the spatial information is lost after pooling.

CNNs in the Euclidean and non-Euclidean framework

- On an Euclidean domain: the convolution can be interpreted as passing a template at each point of the domain and recording the correlation of the template with the function at that point (due to the shift-invariance).
- The spectral approach (convolution defined in the spectral domain) has a shortcoming: "locally defined" and the model cannot be adapted across different domains.



Source: Bronstein et al., 2017

- Another challenge: no immediate way to define a meaningful spatial pooling operation: (i) the number of points may vary; (ii) the spatial information is lost after pooling.
- Need models that: (i) work in the spatial domain; (ii) are consistent when passing from a domain to another.

Non-Euclidean CNNs on manifolds

- **Problem:** finding intrinsical correspondences between deformable 3D shapes (applications in texture mapping, animation, editing, and scene understanding)

Non-Euclidean CNNs on manifolds

- **Problem:** finding intrinsical correspondences between deformable 3D shapes (applications in texture mapping, animation, editing, and scene understanding)
- **Approach:** the 3D shapes can be modeled as manifolds

Non-Euclidean CNNs on manifolds

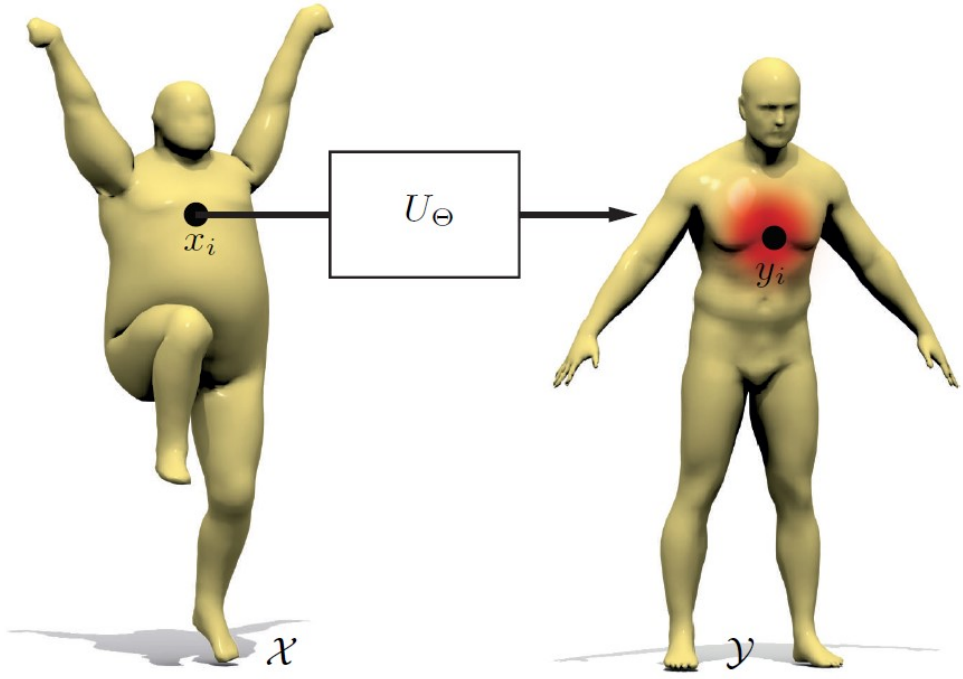
- **Problem:** finding intrinsical correspondences between deformable 3D shapes (applications in texture mapping, animation, editing, and scene understanding)
- **Approach:** the 3D shapes can be modeled as manifolds
- **ML standpoint:** the correspondence can be interpreted as a classification problem. In this problem each point on the query shape is assigned to one of the points on a reference shape (serving as a "label space")

Pointwise correspondence

- Learn the correspondence with a deep network applied to some input feature vector $\mathrm{f}(x)$ at each point x of the query shape M, producing an output $U_\Theta(\mathrm{f}(x))(y)$, which is interpreted as the conditional probability $p(y|x)$ of x being mapped to y. A loss function penalizes the deviation of the predicted correspondence from the groundtruth. **Such an approach learns pointwise correspondences.**

Pointwise correspondence

- Learn the correspondence with a deep network applied to some input feature vector $\mathrm{f}(x)$ at each point x of the query shape M, producing an output $U_\Theta(\mathrm{f}(x))(y)$, which is interpreted as the conditional probability $p(y|x)$ of x being mapped to y. A loss function penalizes the deviation of the predicted correspondence from the groundtruth. **Such an approach learns pointwise correspondences.**



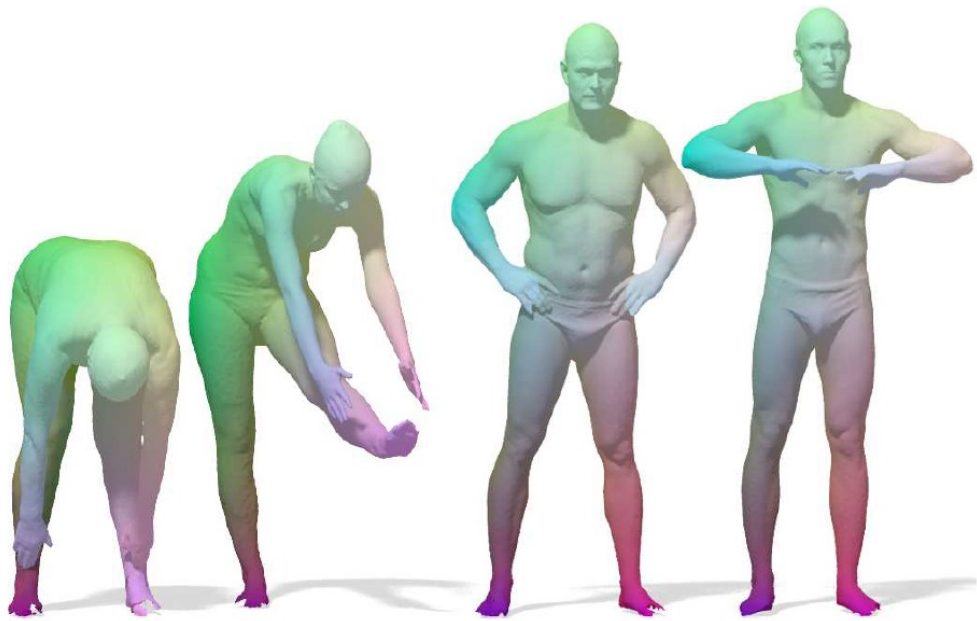
Source: Bronstein et al., 2017

Need for integrating local similarity

- Intrinsic structured prediction of shape correspondence by integrating a layer computing functional correspondence (based on the Laplacian decomposition on a Riemannian 2D-manifold) Litany et al., 2017

Need for integrating local similarity

- Intrinsic structured prediction of shape correspondence by integrating a layer computing functional correspondence (based on the Laplacian decomposition on a Riemannian 2D-manifold) Litany et al., 2017



Source: Litany et al., 2017

"Convolution-type" operation on a smooth manifold

- Try to introduce a "patch operator", but one needs to take into account that: (i) the patch operator is position-dependent; (ii) the parametrizations are local.

"Convolution-type" operation on a smooth manifold

- Try to introduce a "patch operator", but one needs to take into account that: (i) the patch operator is position-dependent; (ii) the parametrizations are local.
- Approach: consider **weigthing functions** $w_1(x,\cdot),\dots,w_J(x,\cdot)$ localized near a point x on a manifold M. Then define, for a function f the **patch operator**

$$D_j(x)f = \int_M f(t)w_j(x,t)dt, \qquad j = 1,\dots,J.$$

Consider a **"filter"** $\gamma = (\gamma_1,\dots,\gamma_j)$ and define

$$(f \star \gamma)(x) = \sum_j \gamma_j D_j(x)f.$$

Several non-Euclidean CNNs rely on suitable choices of the weights. Morover, some spectrum-free methods can be interpreted in terms of local weighting functions.

Geodesic CNN - overview

- Masci et al., 2015 Approach: introduce local geodesic system of polar coordinates to extract "patches". The patches are then passed through a cascade of filters and linear and non-linear operators.

Geodesic CNN - overview

- Masci et al., 2015 Approach: introduce local geodesic system of polar coordinates to extract "patches". The patches are then passed through a cascade of filters and linear and non-linear operators.

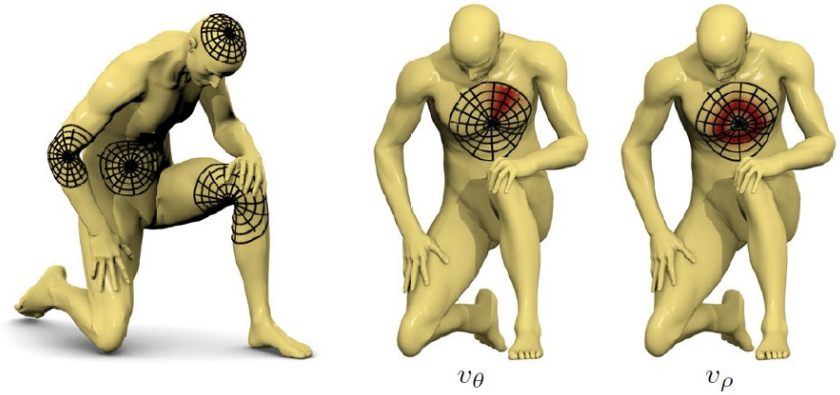


Figure 1: Construction of local geodesic polar coordinates on a manifold. Left: examples of local geodesic patches, center and right: example of angular and radial weights v_θ, v_ρ, respectively (red denotes larger weights).

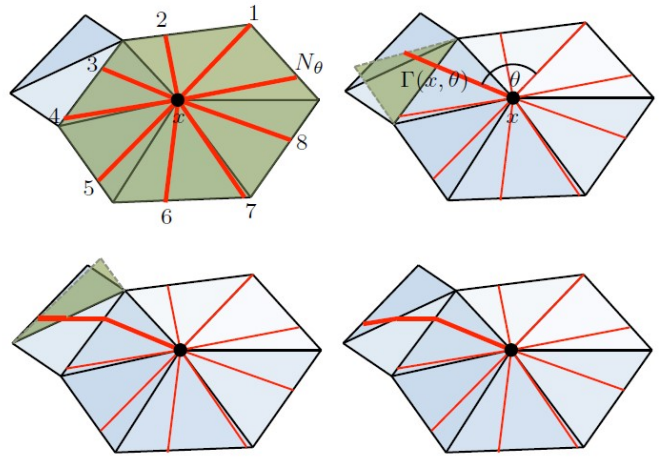


Figure 2: Construction of local geodesic polar coordinates on a triangular mesh. Shown clock-wise: division of 1-ring of vertex x_i into N_θ equi-angular bins; propagation of a ray (bold line) by unfolding the respective triangles (marked in green).

Source: Masci et al., 2015

Geodesic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider geodesic polar coordinates.

Geodesic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider geodesic polar coordinates.
  - The radial coordinate: given by the α-level sets: $\{y \in M | d_{(M,g)}(x, y) = \alpha\}$. Put $B_\alpha(x) = \{y \in M | d_{(M,g)}(x, y) \leq \alpha\}$.

Geodesic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider geodesic polar coordinates.
  - The radial coordinate: given by the α-level sets: $\{y \in M | d_{(M,g)}(x, y) = \alpha\}$. Put $B_\alpha(x) = \{y \in M | d_{(M,g)}(x, y) \leq \alpha\}$.
  - The angular coordinate: set of geodesics starting from x in direction θ.

Geodesic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider geodesic polar coordinates.
  - The radial coordinate: given by the α-level sets: $\{y \in M | d_{(M,g)}(x, y) = \alpha\}$. Put $B_\alpha(x) = \{y \in M | d_{(M,g)}(x, y) \leq \alpha\}$.
  - The angular coordinate: set of geodesics starting from x in direction θ.
  - The definition can be adapted for boundary points.

Geodesic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider geodesic polar coordinates.
  - The radial coordinate: given by the α-level sets: $\{y \in M | d_{(M,g)}(x, y) = \alpha\}$. Put $B_\alpha(x) = \{y \in M | d_{(M,g)}(x, y) \leq \alpha\}$.
  - The angular coordinate: set of geodesics starting from x in direction θ.
  - The definition can be adapted for boundary points.

  Use a chart map $\Omega(x) : B_{\rho_0}(x) \to [0, \rho_0] \times [0, 2\pi)$. Define the **patch operator**

$$(D(x)f)(\rho, \theta) = (f \circ \Omega^{-1}(x))(\rho, \theta)$$

  and the **geodesic convolution** with the filter $\gamma(\rho, \theta)$

$$(f \star \gamma)(x) = \sum_{\rho,\theta} \gamma(\rho, \theta + \Delta\theta)(D(x)f)(\rho, \theta).$$

Geodesic CNN - the architecture

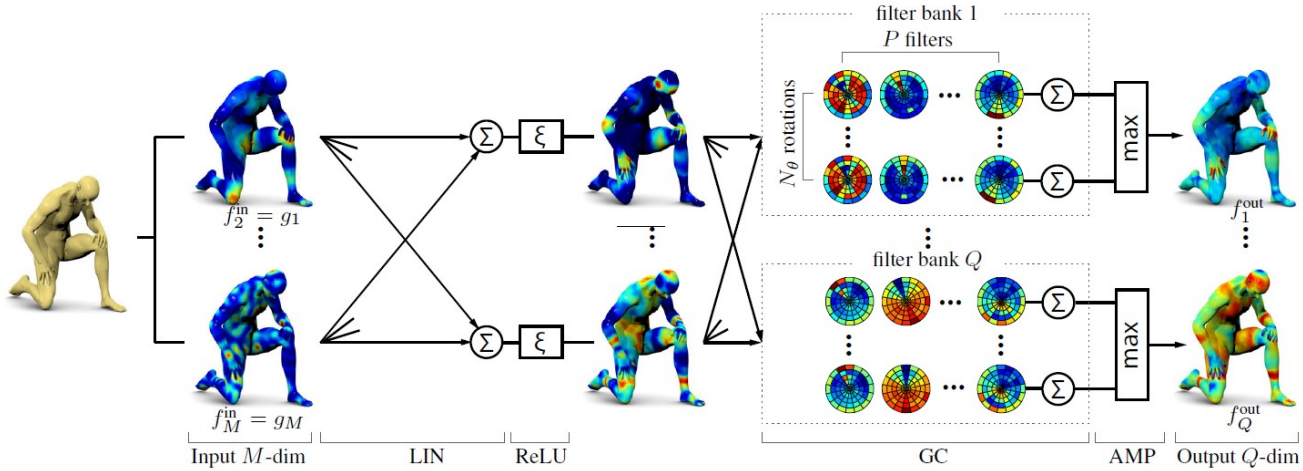


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

Geodesic CNN - the architecture

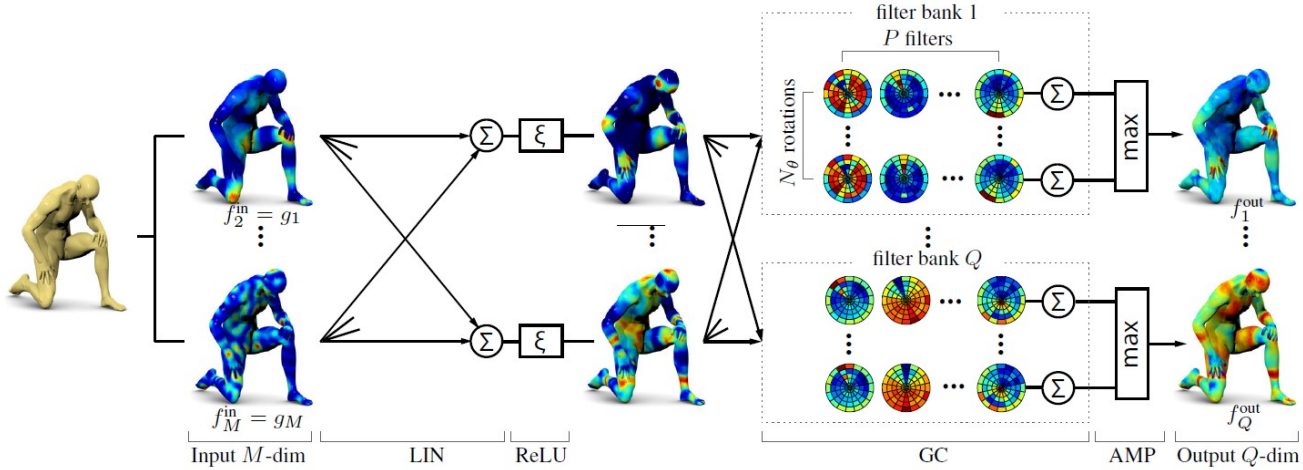


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

- Input: $\mathrm{F} = (f(x_1), \ldots, f(x_N))$ is a $P \times N$ matrix of input features (Heat Kernel Signature, geometry vectors, etc.) at all points of the mesh

Geodesic CNN - the architecture

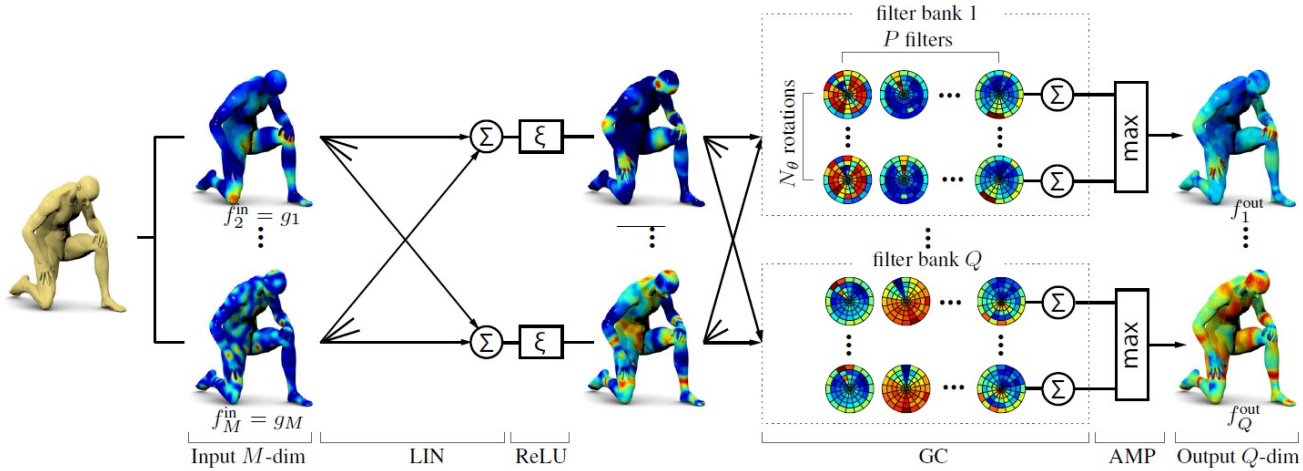


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

- Input: $\mathrm{F} = (f(x_1), \ldots, f(x_N))$ is a $P \times N$ matrix of input features (Heat Kernel Signature, geometry vectors, etc.) at all points of the mesh
- LIN (linear layer) $f_q^{\text{out}}(x) = \xi(\sum_{p=1}^{P} \lambda_{qp} f_p^{\text{in}}(x))$

Geodesic CNN - the architecture

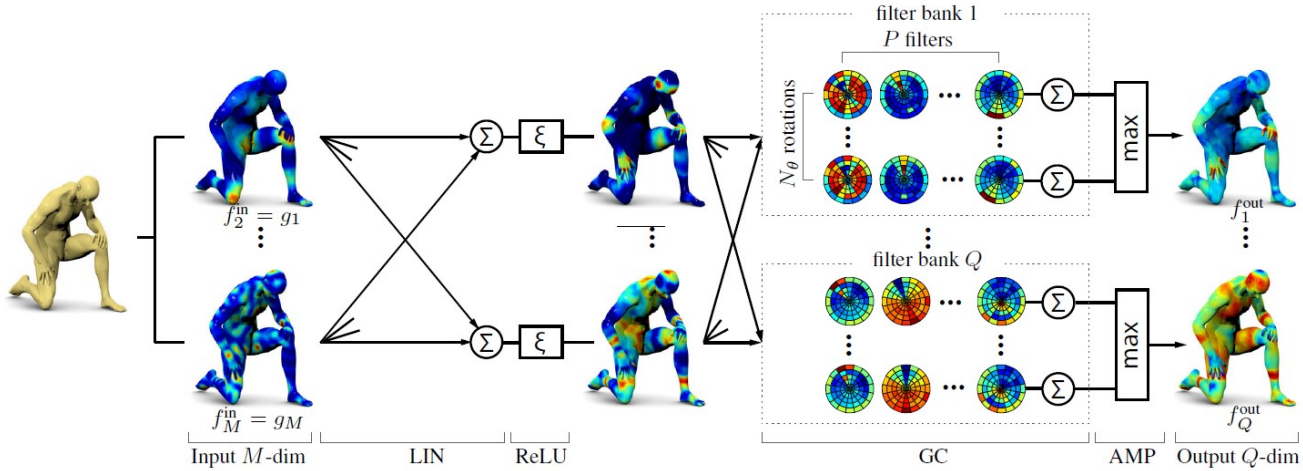


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

- Input: $\mathrm{F} = (f(x_1), \ldots, f(x_N))$ is a $P \times N$ matrix of input features (Heat Kernel Signature, geometry vectors, etc.) at all points of the mesh
- LIN (linear layer) $f_q^{\text{out}}(x) = \xi(\sum_{p=1}^{P} \lambda_{qp} f_p^{\text{in}}(x))$
- GC (geodesic convolution) $f_{\Delta\theta,q}^{\text{out}} = \sum_{p=1}^{P} (f_p \star \gamma_{\Delta\theta,qp})(x)$

Geodesic CNN - the architecture

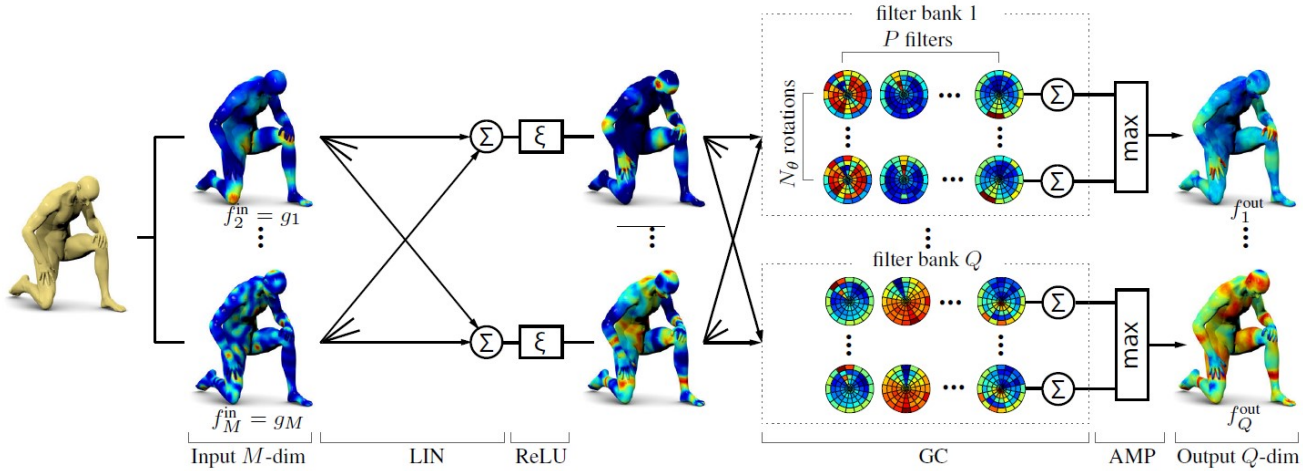


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

- Input: $\mathrm{F} = (f(x_1), \ldots, f(x_N))$ is a $P \times N$ matrix of input features (Heat Kernel Signature, geometry vectors, etc.) at all points of the mesh
- LIN (linear layer) $f_q^{\text{out}}(x) = \xi(\sum_{p=1}^{P} \lambda_{qp} f_p^{\text{in}}(x))$
- GC (geodesic convolution) $f_{\Delta\theta,q}^{\text{out}} = \sum_{p=1}^{P} (f_p \star \gamma_{\Delta\theta,qp})(x)$
- AMP (angular max-pooling): $f_p^{\text{out}}(x) = \max_{\Delta\theta} f_{\Delta\theta,p}^{\text{in}}(x)$

Geodesic CNN - the architecture

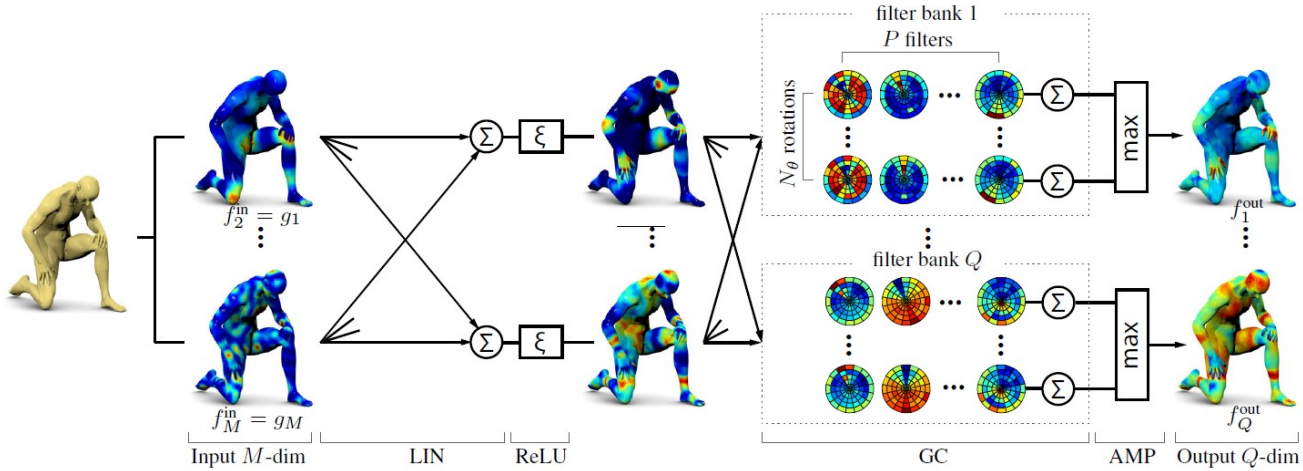


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

- Input: $\mathrm{F} = (f(x_1), \ldots, f(x_N))$ is a $P \times N$ matrix of input features (Heat Kernel Signature, geometry vectors, etc.) at all points of the mesh
- LIN (linear layer) $f_q^{\text{out}}(x) = \xi(\sum_{p=1}^{P} \lambda_{qp} f_p^{\text{in}}(x))$
- GC (geodesic convolution) $f_{\Delta\theta,q}^{\text{out}} = \sum_{p=1}^{P} (f_p \star \gamma_{\Delta\theta,qp})(x)$
- AMP (angular max-pooling): $f_p^{\text{out}}(x) = \max_{\Delta\theta} f_{\Delta\theta,p}^{\text{in}}(x)$
- Other layers: FTM (Fourier Transform Magnitude); COV (Covariance)

Geodesic CNN - the architecture

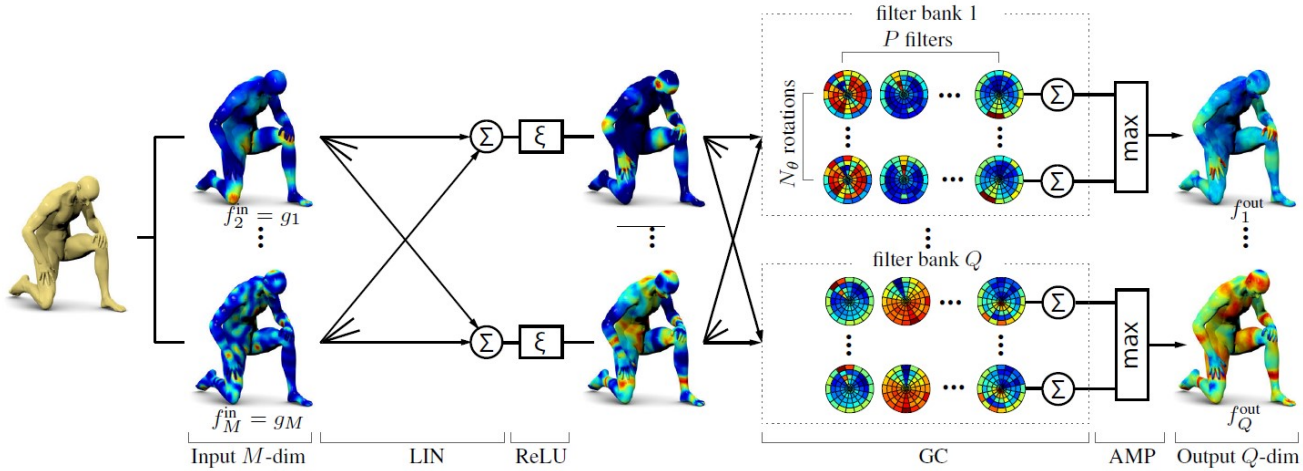


Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Source: Masci et al., 2015

- Input: $\mathrm{F} = (f(x_1), \ldots, f(x_N))$ is a $P \times N$ matrix of input features (Heat Kernel Signature, geometry vectors, etc.) at all points of the mesh
- LIN (linear layer) $f_q^{\text{out}}(x) = \xi(\sum_{p=1}^{P} \lambda_{qp} f_p^{\text{in}}(x))$
- GC (geodesic convolution) $f_{\Delta\theta,q}^{\text{out}} = \sum_{p=1}^{P} (f_p \star \gamma_{\Delta\theta,qp})(x)$
- AMP (angular max-pooling): $f_p^{\text{out}}(x) = \max_{\Delta\theta} f_{\Delta\theta,p}^{\text{in}}(x)$
- Other layers: FTM (Fourier Transform Magnitude); COV (Covariance)
- Get a non-linear expression $\psi_\Theta(\mathrm{F})$, where Θ are the learnable parameters of all layers

Geodesic CNN - applications

- **Shape correspondence.** Aim is to label each vertex of a query shape M_Q with the index of a corresponding shape on some reference shape M_R. In this case, a suitable loss function (the multinomial regression loss) is minimized on a training set of known correspondence.

Geodesic CNN - applications

- **Shape correspondence.** Aim is to label each vertex of a query shape M_Q with the index of a corresponding shape on some reference shape M_R. In this case, a suitable loss function (the multinomial regression loss) is minimized on a training set of known correspondence.
- Example of application where the performances of pre-existing methods were improved

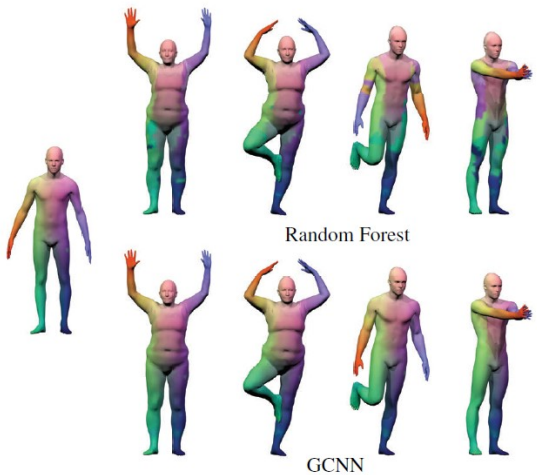


Figure 7: Example of correspondence obtained with GCNN (bottom) and random forest (top). Similar colors encode corresponding points.

Source: Masci et al., 2015

Anisotropic CNN - overview

- Boscaini et al., 2016 Approach: construct a patch operator and then define intrinsic covolution. The patches are then passed through a cascade of filters and linear and non-linear operators.

Anisotropic CNN - overview

- Boscaini et al., 2016 Approach: construct a patch operator and then define intrinsic covolution. The patches are then passed through a cascade of filters and linear and non-linear operators.
- The construction relies on the anisotropic heat kernel $h_{\alpha\theta t}$, which is in turn related to the anisotropic Laplacian

$$\Delta_{\alpha\theta} f(x) = -\mathrm{div}_M(D_{\alpha\theta}(x)\nabla_M f(x)),$$

namely it is the solution of the equation $h_t(x,t) = \Delta_{\alpha\theta} h_t(x,t)$ and it can be expressed by using a suitable basis of eigenfunctions.

Anisotropic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider local coordinates (t, θ) that behave like a polar system (t denotes the scale of the heat kernel and θ its orientation).

Anisotropic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider local coordinates (t, θ) that behave like a polar system (t denotes the scale of the heat kernel and θ its orientation).
- The heat kernel $h_{\alpha\theta t}$ is interpreted as **local weighting function**. For an anisotropy level $\alpha > 1$ one constructs

$$(D(x)f)(\theta, t) = \frac{\int_M h_{\alpha\theta t}(x, \eta) f(\eta) d\eta}{\int_M h_{\alpha\theta t}(x, \eta) d\eta}.$$

Anisotropic CNN - the method

- Given: (M, g) Riemannian manifold. For a given point $x \in M$ consider local coordinates (t, θ) that behave like a polar system (t denotes the scale of the heat kernel and θ its orientation).
- The heat kernel $h_{\alpha\theta t}$ is interpreted as **local weighting function**. For an anisotropy level $\alpha > 1$ one constructs

$$(D(x)f)(\theta, t) = \frac{\int_M h_{\alpha\theta t}(x, \eta) f(\eta) d\eta}{\int_M h_{\alpha\theta t}(x, \eta) d\eta}.$$

- The **intrinsic convolution** is defined as

$$(f \star \gamma)(x) = \int \gamma(\theta, t)(D(x)f)(\theta, t) dt d\theta.$$

Anisotropic CNN - the architecture

- The architecture depended on the task. Examples:
  FC64+IC64+IC128+IC256+FC1024+FC512+Softmax (full mesh correspondence)
  IC32+FC1024+DO(0.5)+FC2048+DO(0.5)+Softmax (partial correspondence)

Anisotropic CNN - the architecture

- The architecture depended on the task. Examples:
  FC64+IC64+IC128+IC256+FC1024+FC512+Softmax (full mesh correspondence)
  IC32+FC1024+DO(0.5)+FC2048+DO(0.5)+Softmax (partial correspondence)
- FC (FCQ) - fully connected (see LIN - linear in the case of the Geodesic CNNs)

Anisotropic CNN - the architecture

- The architecture depended on the task. Examples:
  FC64+IC64+IC128+IC256+FC1024+FC512+Softmax (full mesh correspondence)
  IC32+FC1024+DO(0.5)+FC2048+DO(0.5)+Softmax (partial correspondence)
- FC (FCQ) - fully connected (see LIN - linear in the case of the Geodesic CNNs)
- IC (ICQ) - intrinsic convolution

$$f_q^{\text{out}}(x) = \sum_{p=1}^{P} (f_p^{\text{in}} \star \gamma_{qp})(x), \quad q = 1, \dots, Q,$$

where $\gamma_{qp}(\theta, t)$ are the learnable coefficients of the p-th filter in the q-th filter bank.

Anisotropic CNN - the architecture

- The architecture depended on the task. Examples:
  FC64+IC64+IC128+IC256+FC1024+FC512+Softmax (full mesh correspondence)
  IC32+FC1024+DO(0.5)+FC2048+DO(0.5)+Softmax (partial correspondence)
- FC (FCQ) - fully connected (see LIN - linear in the case of the Geodesic CNNs)
- IC (ICQ) - intrinsic convolution

$$f_q^{\text{out}}(x) = \sum_{p=1}^{P} (f_p^{\text{in}} \star \gamma_{qp})(x), \quad q = 1, \dots, Q,$$

  where $\gamma_{qp}(\theta, t)$ are the learnable coefficients of the p-th filter in the q-th filter bank.
- Softmax

Anisotropic CNN - the architecture

- The architecture depended on the task. Examples:
  FC64+IC64+IC128+IC256+FC1024+FC512+Softmax (full mesh correspondence)
  IC32+FC1024+DO(0.5)+FC2048+DO(0.5)+Softmax (partial correspondence)
- FC (FCQ) - fully connected (see LIN - linear in the case of the Geodesic CNNs)
- IC (ICQ) - intrinsic convolution

$$f_q^{\text{out}}(x) = \sum_{p=1}^{P} (f_p^{\text{in}} \star \gamma_{qp})(x), \quad q = 1, \ldots, Q,$$

  where $\gamma_{qp}(\theta, t)$ are the learnable coefficients of the p-th filter in the q-th filter bank.
- Softmax
- DO(ν) - dropout

Anisotropic CNN - applications

- Example of application where the performances of pre-existing methods were improved



Figure 5: Pointwise geodesic error (in % of geodesic diameter) of different correspondence methods (top to bottom: Blended intrinsic maps, GCNN, ACNN) on the FAUST dataset. For visualization clarity, the error values are saturated at 10% of the geodesic diameter. Hot colors correspond to large errors. Note the different behavior of different approaches: BIM produces large distortions with very few accurate matches; GCNN produces many near-perfect matches but also many matches with large distortion; ACNN produces very few matches with large distortion and many near-perfect matches.

Source: Boscaini et al., 2016

Mixture Model Network (MoNet) - overview

- Monti et al., 2017 MoNet: a generic spatial-domain framework allowing to design convolutional deep architectures on non-Euclidean domains such as graphs and manifolds.

Mixture Model Network (MoNet) - overview

- Monti et al., 2017 MoNet: a generic spatial-domain framework allowing to design convolutional deep architectures on non-Euclidean domains such as graphs and manifolds.
- The approach follows the general philosophy of spatial-domain methods: convolution-like operations are thought as template matching with local intrinsic patches on manifolds/graphs.

Mixture Model Network (MoNet) - overview

- Monti et al., 2017 MoNet: a generic spatial-domain framework allowing to design convolutional deep architectures on non-Euclidean domains such as graphs and manifolds.
- The approach follows the general philosophy of spatial-domain methods: convolution-like operations are thought as template matching with local intrinsic patches on manifolds/graphs.
- **Idea:** use a parametric construction of the local coordinate system.

Mixture Model Network (MoNet) - the method

- Take x a point on a manifold or a vertex of a graph. Consider points $y \in \mathcal{N}(x)$ in the neighborhood of x.

Mixture Model Network (MoNet) - the method

- Take x a point on a manifold or a vertex of a graph. Consider points $y \in \mathcal{N}(x)$ in the neighborhood of x.
- For each such y one associates a d-dimensional vector of pseudo-coordinates $\mathrm{u}(x, y)$. In these coordinates, one defines a weighting function (kernel)

$$\mathrm{w}_\Theta = (w_1(\mathrm{u}), \ldots, w_J(\mathrm{u})),$$

parametrized by some learnable parameters Θ.

Mixture Model Network (MoNet) - the method

- Take x a point on a manifold or a vertex of a graph. Consider points $y \in \mathcal{N}(x)$ in the neighborhood of x.
- For each such y one associates a d-dimensional vector of pseudo-coordinates $\mathrm{u}(x, y)$. In these coordinates, one defines a weighting function (kernel)

$$\mathrm{w}_\Theta = (w_1(\mathrm{u}), \ldots, w_J(\mathrm{u})),$$

parametrized by some learnable parameters Θ.

- **The patch operator** can be written in the general form

$$D_j(x)f = \sum_{y \in \mathcal{N}(x)} w_j(\mathrm{u}(x, y))f(y), \quad j = 1, \ldots, J.$$

- **The convolution-like operation**

$$(f \star \gamma)(x) = \sum_{j=1}^{J} \gamma_j D_j(x)f.$$

Mixture Model Network (MoNet) - the method

- Take x a point on a manifold or a vertex of a graph. Consider points $y \in \mathcal{N}(x)$ in the neighborhood of x.
- For each such y one associates a d-dimensional vector of pseudo-coordinates $\mathrm{u}(x, y)$. In these coordinates, one defines a weighting function (kernel)

$$\mathrm{w}_\Theta = (w_1(\mathrm{u}), \ldots, w_J(\mathrm{u})),$$

parametrized by some learnable parameters Θ.

- **The patch operator** can be written in the general form

$$D_j(x)f = \sum_{y \in \mathcal{N}(x)} w_j(\mathrm{u}(x, y))f(y), \quad j = 1, \ldots, J.$$

- **The convolution-like operation**

$$(f \star \gamma)(x) = \sum_{j=1}^{J} \gamma_j D_j(x)f.$$

- **The key ingredients** are (i) the pseudo-coordinates u and (ii) the weight functions $\mathrm{w}(\mathrm{u})$.

Mixture Model Network (MoNet) - a general framework

Table 1. Several CNN-type geometric deep learning methods on graphs and manifolds can be obtained as a particular setting of the proposed framework with an appropriate choice of the pseudo-coordinates and weight functions in the definition of the patch operator. x denotes the reference point (center of the patch) and y a point within the patch. $\mathbf{x}$ denotes the Euclidean coordinates on a regular grid. $\bar{\alpha}, \bar{\sigma}_\rho, \bar{\sigma}_\theta$ and $\bar{\mathbf{u}}_j, \bar{\theta}_j, j = 1, \dots, J$ denote fixed parameters of the weight functions.

Method	Pseudo-coordinates	$\mathbf{u}(x,y)$	Weight function $w_j(\mathbf{u}), j = 1, \dots, J$
CNN [23]	Local Euclidean	$\mathbf{x}(x,y) = \mathbf{x}(y) - \mathbf{x}(x)$	$\delta(\mathbf{u} - \bar{\mathbf{u}}_j)$
GCNN [26]	Local polar geodesic	$\rho(x,y), \theta(x,y)$	$\exp(-\frac{1}{2}(\mathbf{u} - \bar{\mathbf{u}}_j)^\top \begin{pmatrix} \bar{\sigma}_\rho^2 & \\ & \bar{\sigma}_\theta^2 \end{pmatrix}^{-1} (\mathbf{u} - \bar{\mathbf{u}}_j))$
ACNN [7]	Local polar geodesic	$\rho(x,y), \theta(x,y)$	$\exp(-\frac{1}{2}\mathbf{u}^\top \mathbf{R}_{\bar{\theta}_j} \begin{pmatrix} \bar{\alpha} & \\ & 1 \end{pmatrix} \mathbf{R}_{\bar{\theta}_j}^\top \mathbf{u})$
GCN [21]	Vertex degree	$\deg(x), \deg(y)$	$\left(1 - \lvert 1 - \frac{1}{\sqrt{u_1}} \rvert\right)\left(1 - \lvert 1 - \frac{1}{\sqrt{u_2}} \rvert\right)$
DCNN [3]	Transition probability in r hops	$p^0(x,y), \dots, p^{r-1}(x,y)$	$\mathrm{id}(u_j)$

Source: Monti et al., 2017

Mixture Model Network (MoNet) - the method

- Compared to previous approaches, instead of using handcrafted weight functions, one considers **parametric kernels with learnable parameters**.

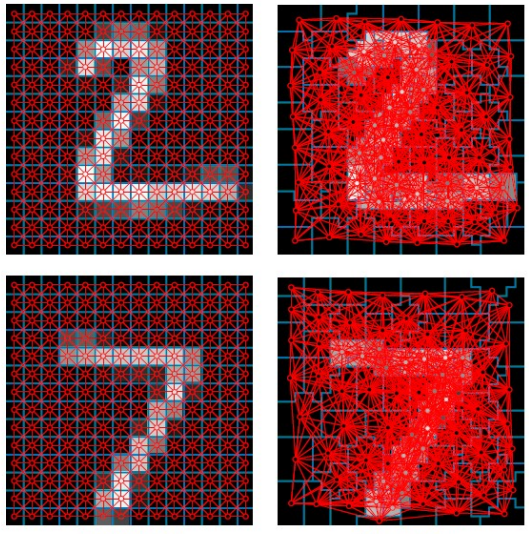
Mixture Model Network (MoNet) - the method

- Compared to previous approaches, instead of using handcrafted weight functions, one considers **parametric kernels with learnable parameters**.
- Possible choice:

$$w_j(\mathrm{u}) = \exp(-\frac{1}{2}(\mathrm{u} - \mu_j)^T \Sigma_j^{-1}(\mathrm{u} - \mu_j)),$$

where Σ_j and μ_j are learnable covariance matrix and mean vector of a Gaussian kernel, respectively. Thus, the whole approach can be interpreted as a Gaussian Mixture Model. The covariance matrices are assumed to have a diagonal form, resulting a total of $2Jd$ parameters for the patch operator.

Mixture Model Network (MoNet) - applications (I)



Regular grid Superpixels

Figure 2. Representation of images as graphs. Left: regular grid (the graph is fixed for all images). Right: graph of superpixel adjacency (different for each image). Vertices are shown as red circles, edges as red lines.

Table 2. Classification accuracy of classical Euclidean CNN (LeNet5), spectral CNN (ChebNet) and the proposed approach (MoNet) on different versions of the MNIST dataset. The setting of all the input images sharing the same graph is marked with *.

Dataset	LeNet5 [23]	ChebNet [13]	**MoNet**
*Full grid	99.33%	99.14%	99.19%
$*\frac{1}{4}$ grid	98.59%	97.70%	98.16%
300 Superpixels	-	88.05%	**97.30%**
150 Superpixels	-	80.94%	**96.75%**
75 Superpixels	-	75.62%	**91.11%**

Source: Monti et al., 2017

Mixture Model Network (MoNet) - applications (II)

- Vertex classification on generic graphs.

Mixture Model Network (MoNet) - applications (II)

- Vertex classification on generic graphs.
- The citation graphs Cora and PubMed. In each dataset, a vertex represents a scientific publication and an undirected unweighted edge represents a citation. For each vertex, a feature vector representing the content of the paper is given. The task is to classify each vertex into one of the groundtruth classes.

Mixture Model Network (MoNet) - applications (II)

- Vertex classification on generic graphs.
- The citation graphs Cora and PubMed. In each dataset, a vertex represents a scientific publication and an undirected unweighted edge represents a citation. For each vertex, a feature vector representing the content of the paper is given. The task is to classify each vertex into one of the groundtruth classes.

Table 4. Vertex classification accuracy on the Cora and PubMed datasets following the splitting suggested in [44]. Learning methods (DCNN, GCNN and MoNet) were trained and tested fifty times for showing their average behavior with different initializations.

Method	Cora	PubMed
ManiReg [4]	59.5%	70.7%
SemiEmb [42]	59.0%	71.1%
LP [45]	68.0%	63.0%
DeepWalk [28]	67.2%	65.3%
Planetoid [44]	75.7%	77.2%
DCNN [3]	$76.80 \pm 0.60\%$	$73.00 \pm 0.52\%$
GCN [21]	$81.59 \pm 0.42\%$	$78.72 \pm 0.25\%$
MoNet	$\mathbf{81.69} \pm 0.48\%$	$\mathbf{78.81} \pm 0.44\%$

Source: Monti et al., 2017

Mixture Model Network (MoNet) - applications (III)

- Learning dense intrinsic correspondence between collections of 3D shapes represented as discrete manifolds

Mixture Model Network (MoNet) - applications (III)

- Learning dense intrinsic correspondence between collections of 3D shapes represented as discrete manifolds



Figure 6. Examples of correspondence on the FAUST humans dataset obtained by the proposed MoNet method. Shown is the texture transferred from the leftmost reference shape to different subjects in different poses by means of our correspondence.

Source: Monti et al., 2017

ManifoldNet - the approach

- Chakraborty et al., 2018 and Chakraborty et al., 2019 ManifoldNet

ManifoldNet - the approach

- Chakraborty et al., 2018 and Chakraborty et al., 2019 ManifoldNet
- Main assumption: a convolution is a weighted mean. One introduces manifold-valued data convolutions, defined by using the weighted Fréchet Mean (wFM), where the weights makeup the convolution mask, to be learned.

ManifoldNet - the method

- Let (M, g) be a Riemannian manifold. Let $x_1, x_2, \ldots, x_N \in M$ be manifold-valued samples on M. Let $w_1, w_2, \ldots, w_N$ be weights that satisfy the convexity constraint ($\forall i \quad w_i > 0, \sum_i w_i = 1$). **The weighted Fréchet mean**

$$wFM((x_i)_i, (w_i)_i) = \operatorname{argmin}_{x \in M} \sum_{i=1}^{N} w_i d^2(x_i, x).$$

It is important to have an efficient approach for computing the weighted Fréchet mean.

ManifoldNet - the method

- Let (M, g) be a Riemannian manifold. Let $x_1, x_2, \ldots, x_N \in M$ be manifold-valued samples on M. Let $w_1, w_2, \ldots, w_N$ be weights that satisfy the convexity constraint ($\forall i \quad w_i > 0, \sum_i w_i = 1$). **The weighted Fréchet mean**

$$wFM((x_i)_i, (w_i)_i) = \operatorname{argmin}_{x \in M} \sum_{i=1}^{N} w_i d^2(x_i, x).$$

  It is important to have an efficient approach for computing the weighted Fréchet mean.
- In the same way that convolution is equivariant with respect to translations, the weighted Fréchet mean is equivariant with respect to the isometries of M.

ManifoldNet - the architecture

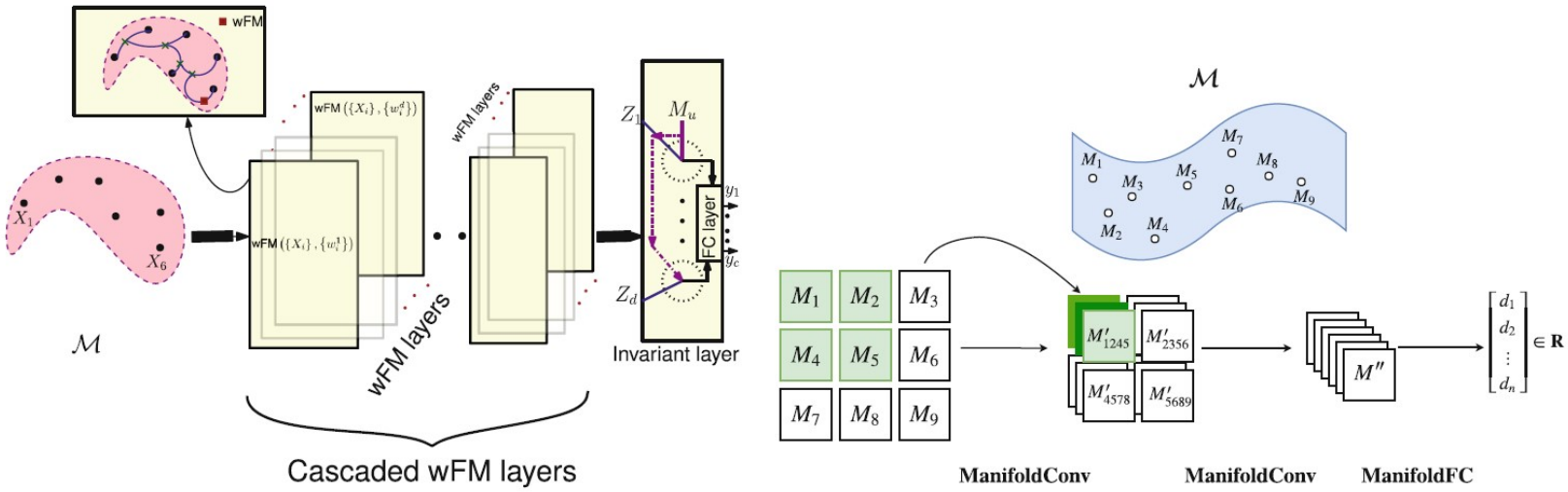


Fig. 1. *Left-Right* (a) Schematic diagram of ManifoldNet (b) 2×2 ManifoldNet conv. example

Source: Chakraborty et al., 2019

ManifoldNet - the architecture

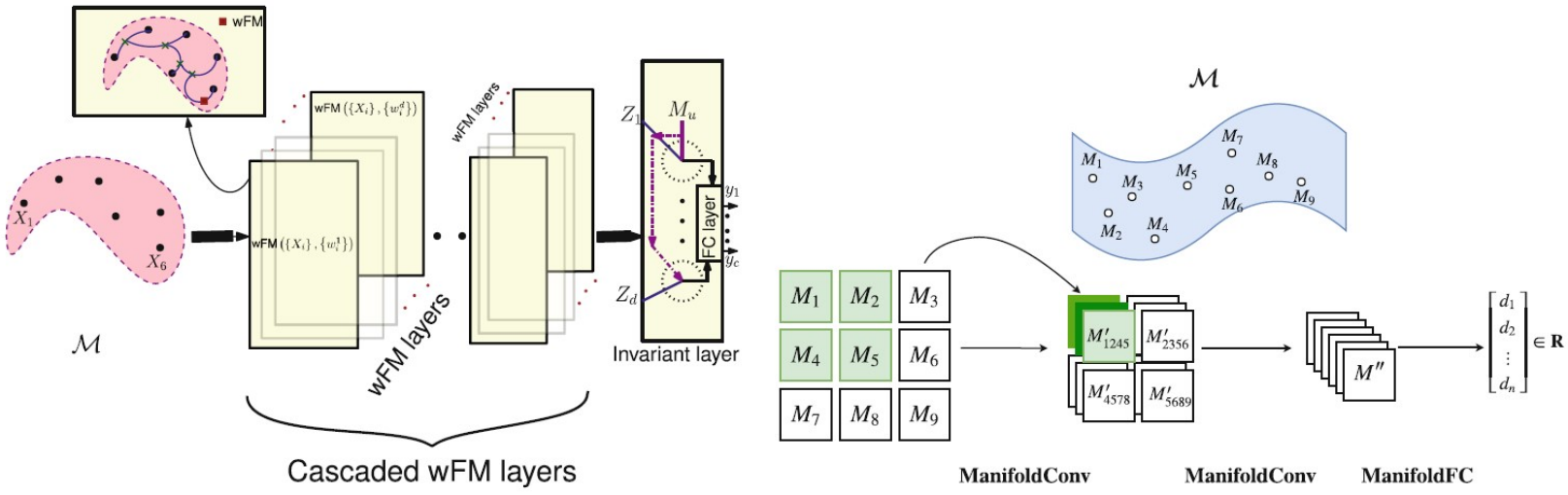


Fig. 1. *Left-Right* (a) Schematic diagram of ManifoldNet (b) 2×2 ManifoldNet conv. example

Source: Chakraborty et al., 2019

- **Comment:** The function wFM is a contraction mapping (in the sense of Mallat, 2016) for any non-trivial choice of weights. Therefore, no activation function was included in the architecture when working on manifolds with non-constant curvature.

ManifoldNet - the architecture

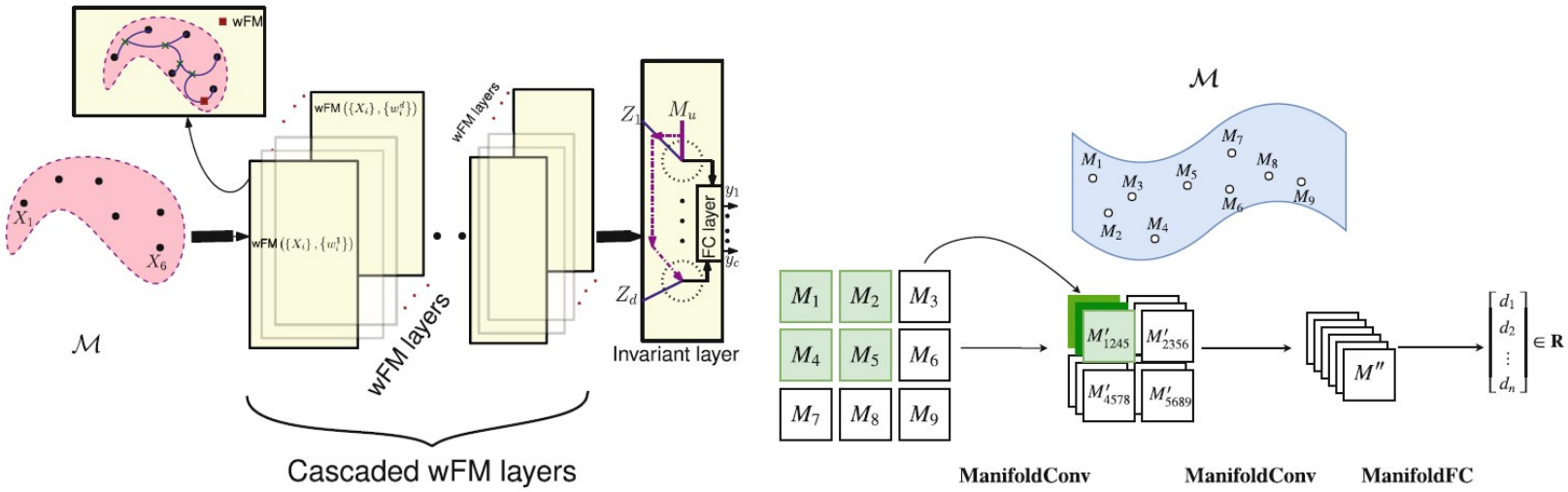


Fig. 1. *Left-Right* (a) Schematic diagram of ManifoldNet (b) 2×2 ManifoldNet conv. example

Source: Chakraborty et al., 2019

- **Comment:** The function wFM is a contraction mapping (in the sense of Mallat, 2016) for any non-trivial choice of weights. Therefore, no activation function was included in the architecture when working on manifolds with non-constant curvature.
- Experiments: computer vision problems and medical imaging tasks

Manifold-Valued Convolution (MVC) - the approach

- Bouza et al., 2020 MVCNet

Manifold-Valued Convolution (MVC) - the approach

- Bouza et al., 2020 MVCNet
- Idea: Use the local structure of the Riemannian manifold. In particular, make use of the exponential and logarithm map and use the vector space structure of the tangent space.

Manifold-Valued Convolution (MVC) - the method

- Let (M, g) be a Riemannian manifold and let $f : \mathbb{Z}^d \to M$ and $w : \mathbb{Z}^d \to \mathbb{R}$ be two functions (the signal and the filter); usually $d = 2$ or $d = 3$. **The convolution** $MVC(f, w) = f \star w : \mathbb{Z}^n \to M$ is defined as

$$(f \star w)(\alpha) = \exp_x \left(\sum_{\beta \in \mathbb{Z}^d} w(\beta - \alpha) \log_x f(\beta) \right), \quad x = FM_{\beta \in \mathbb{Z}^d}(f(\beta)).$$

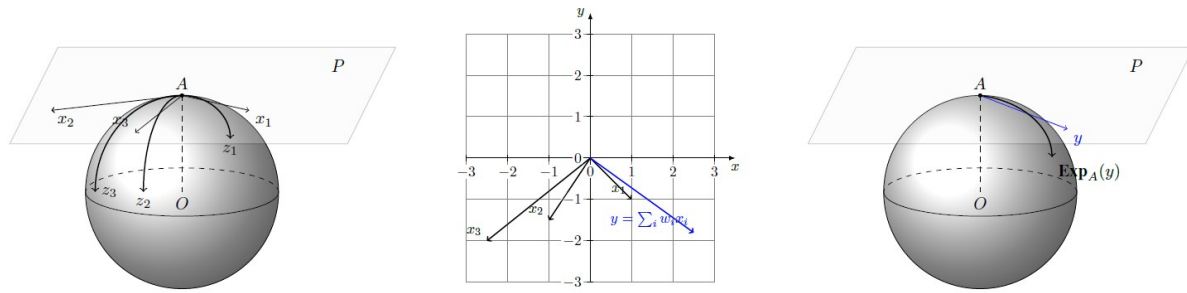
  This operation fulfills the equivariance requirements with respect both the domain and ranges of f and γ.

Manifold-Valued Convolution (MVC) - the method

- Let (M, g) be a Riemannian manifold and let $f : \mathbb{Z}^d \to M$ and $w : \mathbb{Z}^d \to \mathbb{R}$ be two functions (the signal and the filter); usually $d = 2$ or $d = 3$. **The convolution** $MVC(f, w) = f \star w : \mathbb{Z}^n \to M$ is defined as

$$(f \star w)(\alpha) = \exp_x \left(\sum_{\beta \in \mathbb{Z}^d} w(\beta - \alpha) \log_x f(\beta) \right), \quad x = FM_{\beta \in \mathbb{Z}^d}(f(\beta)).$$

This operation fulfills the equivariance requirements with respect both the domain and ranges of f and γ.



(a) Log map all of the points onto the tangent space, i.e. $x_i = \mathbf{Log}_A(z_i)$.

(b) Perform a weighted sum in the tangent space $T_A\mathcal{M}$ to get $y = \sum_i w_i x_i$

(c) Project the resulting vector down using the Riemannian exponential map, i.e. the output is $\mathbf{Exp}_A(y)$.

Figure 1. Tangent combination operation.

Source: Bouza et al., 2020

Manifold-Valued Convolution (MVC) - the architecture and applications

- Architecture:
  MVC + tReLU $\rightarrow$ MVC + tReLU $\rightarrow \cdots \rightarrow$ MVFC
  The number and the size of the layers depended on the experiment settings.

Manifold-Valued Convolution (MVC) - the architecture and applications

- Architecture:
  MVC + tReLU $\rightarrow$ MVC + tReLU $\rightarrow \cdots \rightarrow$ MVFC
  The number and the size of the layers depended on the experiment settings.
- Applications: (i) video classification, (ii) anatomical structure to function regression, (iii) Parkinson's disease classification

Manifold-Valued Convolution (MVC) - the architecture and applications

- Architecture:
  MVC + tReLU $\rightarrow$ MVC + tReLU $\rightarrow \cdots \rightarrow$ MVFC
  The number and the size of the layers depended on the experiment settings.
- Applications: (i) video classification, (ii) anatomical structure to function regression, (iii) Parkinson's disease classification
- For the latter, the manifold of SPD matrices was used and the approach outperformed other methods

Model	Non-linearity	# params.	time (s) / sample	Accuracy	
				Training Accuracy	Test Accuracy
MVC-net	TReLU	$\sim$ **14K**	~ 0.13	$\mathbf{0.991 \pm 0.01}$	$\mathbf{0.973 \pm 0.07}$
DTI-ManifoldNet	None	$\sim 30K$	~ 0.3	$\mathbf{0.973 \pm 0.02}$	$\mathbf{0.948 \pm 0.03}$
ResNet-34	ReLU	$\sim 30M$	$\sim$ **0.008**	$\mathbf{0.984 \pm 0.04}$	0.713 ± 0.02
CapsuleNet	ReLU	$\sim 30M$	~ 0.009	0.63 ± 0.02	0.62 ± 0.04

Figure 2. Comparison results on Diffusion MRI classification.

Source: Bouza et al., 2020

Other approaches - combined spectral and spatial representations

- **Approach:** Take advantadge of the frequency-spatial relationship (e.g. Shuman et al., 2016). Examples: (i) localized spectral CNNs, using the Windowed Fourier Transform – Boscaini et al., 2015; (ii) wavelet approach – Cheng et al., 2016. Illustration of the results obtained with (i)

Other approaches - combined spectral and spatial representations

- **Approach:** Take advantadge of the frequency-spatial relationship (e.g. Shuman et al., 2016). Examples: (i) localized spectral CNNs, using the Windowed Fourier Transform – Boscaini et al., 2015; (ii) wavelet approach – Cheng et al., 2016. Illustration of the results obtained with (i)



Figure 4: Distance map in the descriptor space. A point on the reference shape (leftmost) is compared to all other points on the same and on other shapes. Shown left-to-right: reference shape from FAUST dataset, different pose of the same shape, different subject in the same dataset, two shapes from SCAPE dataset, Gaussian noise, heavy subsampling, voxelization noise, topological noise (glued fingers and missing parts). Small distances in the descriptor space correspond to cold colors.

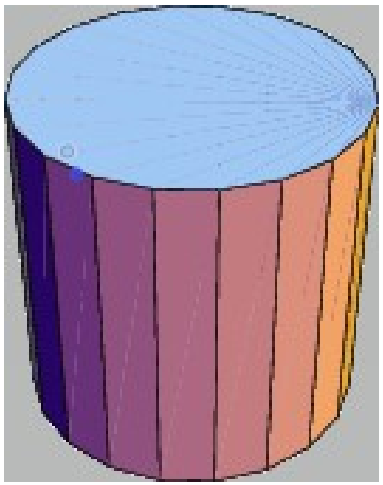
Source: Boscaini et al., 2015

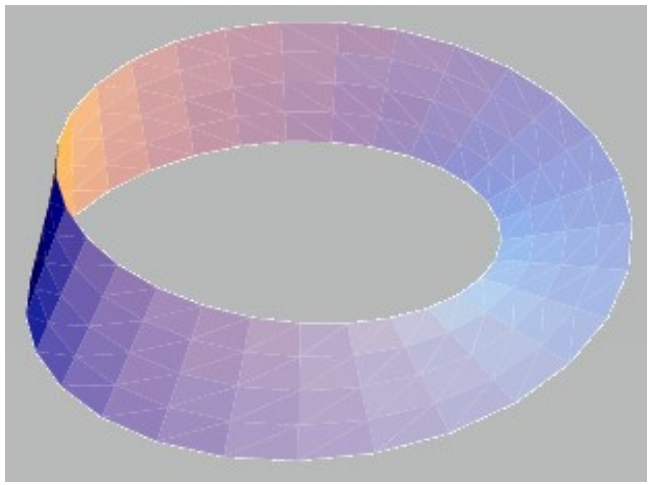
Other (recent) approaches - make use of other geometric structures related to a manifold

- Make use of **fiber bundles, connections and parallel transport** Sommer and Bronstein, 2020

Other (recent) approaches - make use of other geometric structures related to a manifold

- Make use of **fiber bundles, connections and parallel transport** Sommer and Bronstein, 2020
- What is a bundle?





Source: Wolfram MathWorld

Applications of the bundle approach

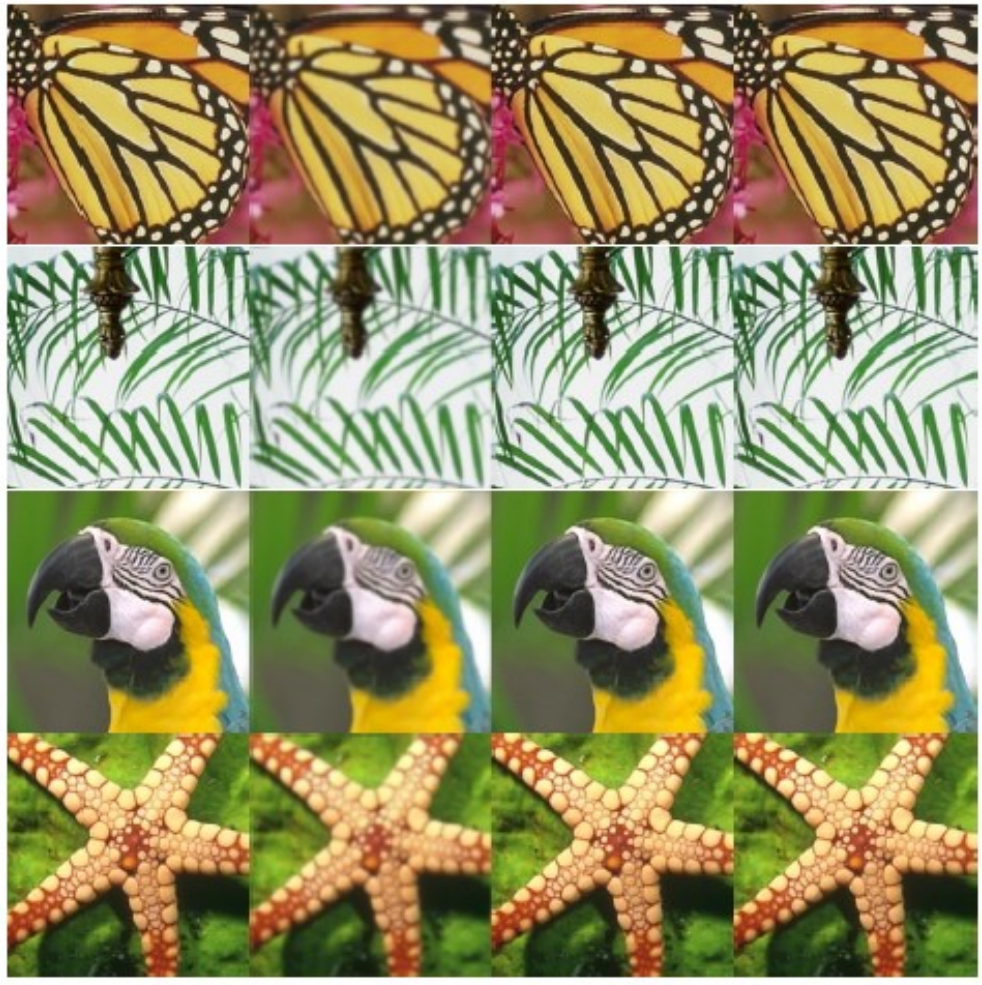


FIG. 3.6. *Deblurring: Comparison of DIP and DIP-VBTV Optimal (best results among the 5 runs in terms of PSNR). From left to right: Clean ground truth image - Corrupted image - Result of DIP - Result of DIP-VBTV Optimal.*

Source: Batard et al., 2021