Functional map*

Umberto Castellani, Riccardo Marin, Simone Melzi

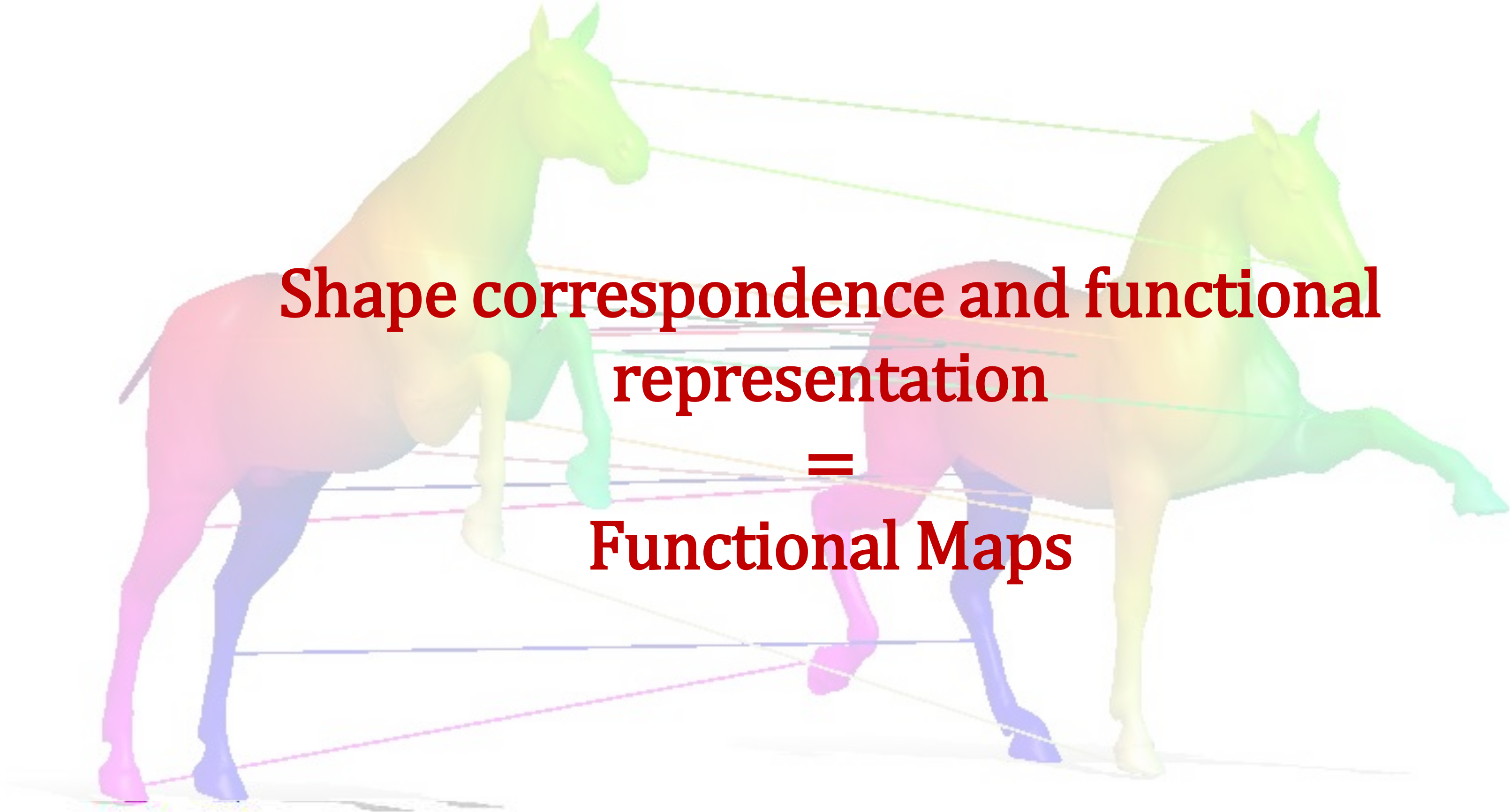
31 july 2020

PhD School on

Spectral shape analysis for 3D matching

*These slides are from the Simone Mezi's Graduate School on «**The functional representation of 3D shapes and High-Frequencies**» https://sgp2020.sites.uu.nl/graduates/

Shape correspondence



Shape correspondence and functional representation

=

Functional Maps

Outline

- 3D shape
- Functions on a 3D shape
- Functions on 2 different domains
- Functional map
- From functional map to point-to-point map

What is a shape?

Continuous setting:

2-dimensional smooth manifold (surface) embedded in $\mathbb{R}^3$

(connected,
eventually with boundary)

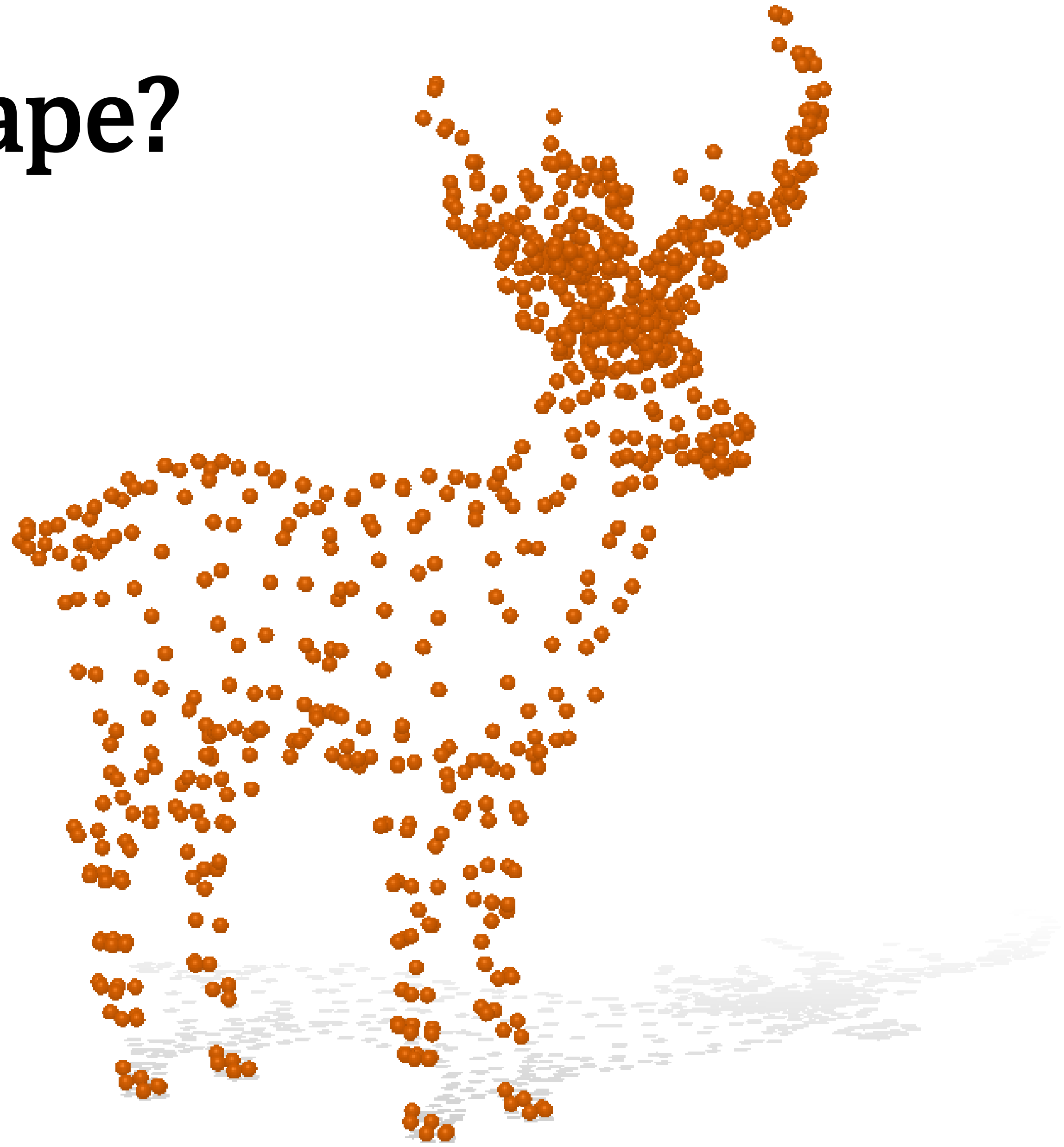
What is a discrete shape?

Discrete setting:

a set of points in $\mathbb{R}^3$

$$\mathcal{P} = \{p_1, \ldots, p_n\}$$

$$p_i \in \mathbb{R}^3$$



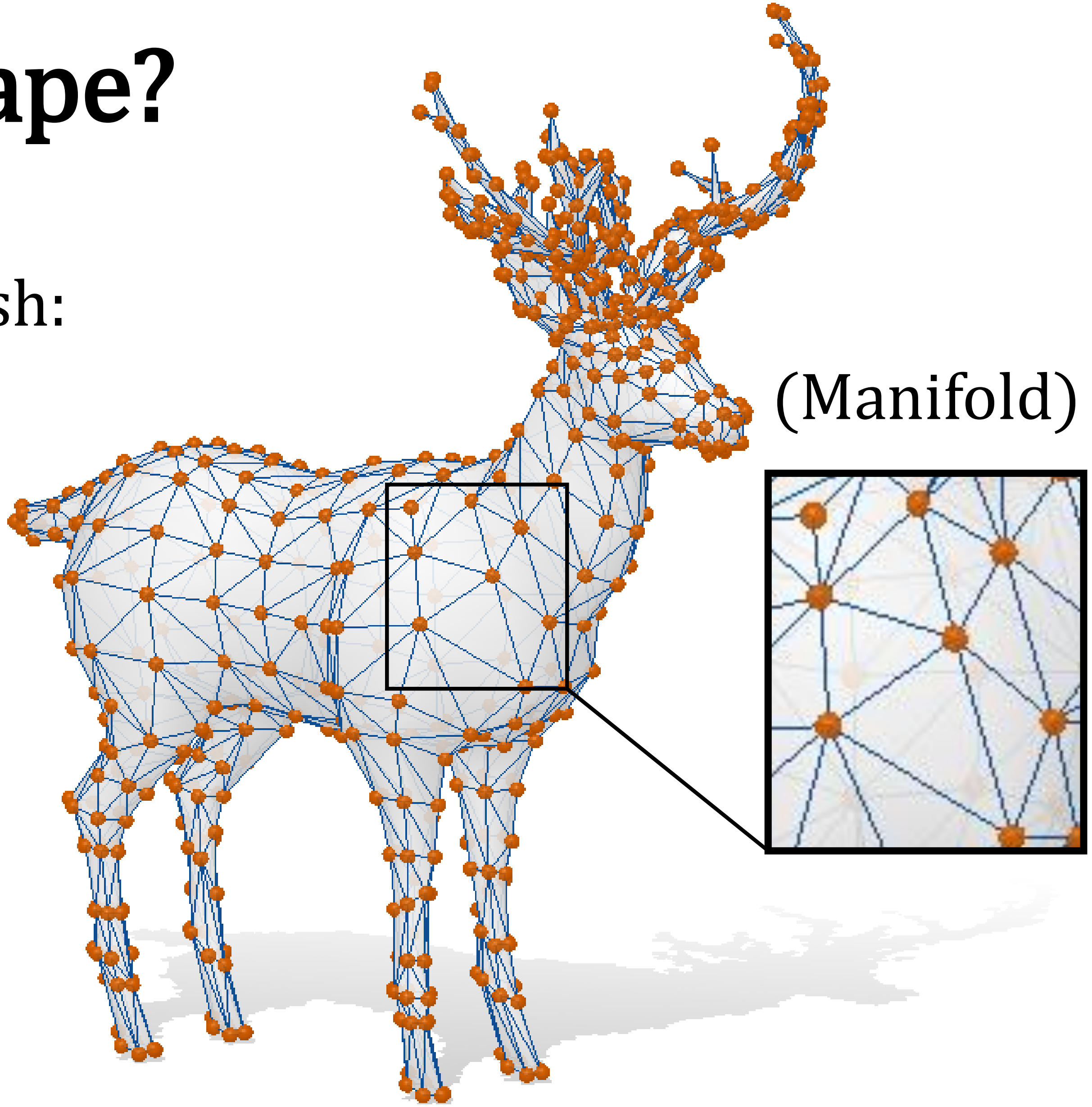
What is a discrete shape?

Represented as a triangular mesh:

vertices

edges

faces



What is a function?

Function (mathematics)

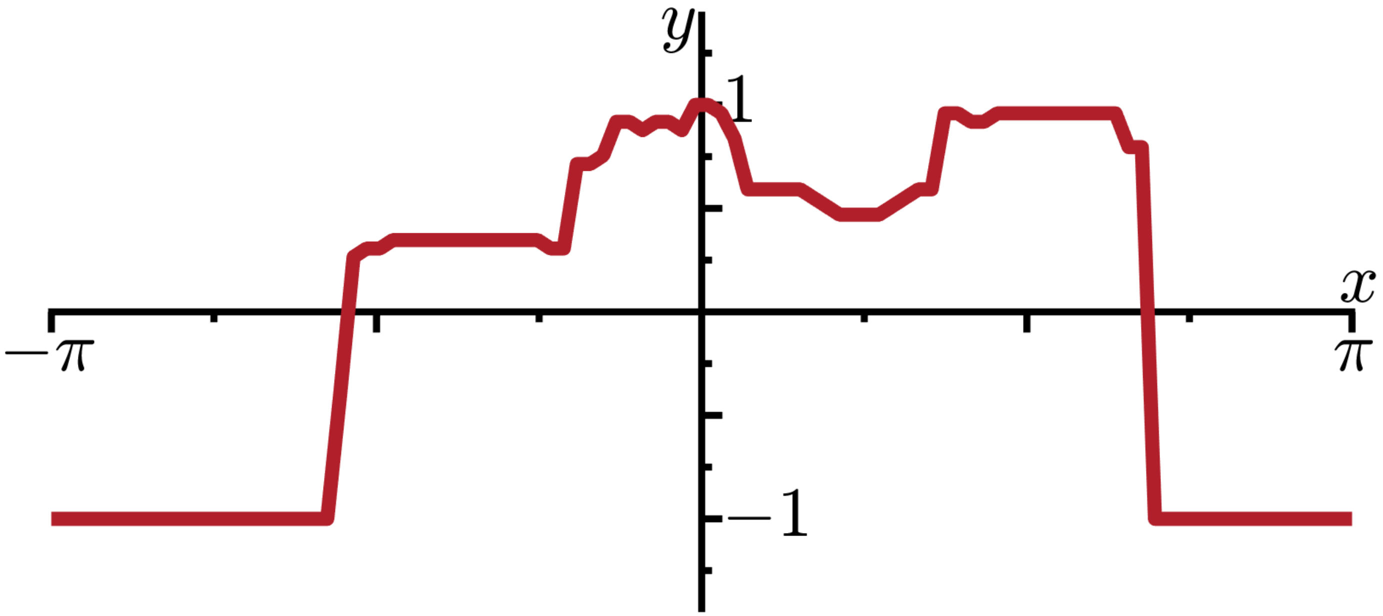
From Wikipedia, the free encyclopedia

In mathematics, a **function**[note 1] is a binary relation over two sets that associates to every element of the first set *exactly* one element of the second set. Typical examples are functions from integers to integers or from the real numbers to real numbers.

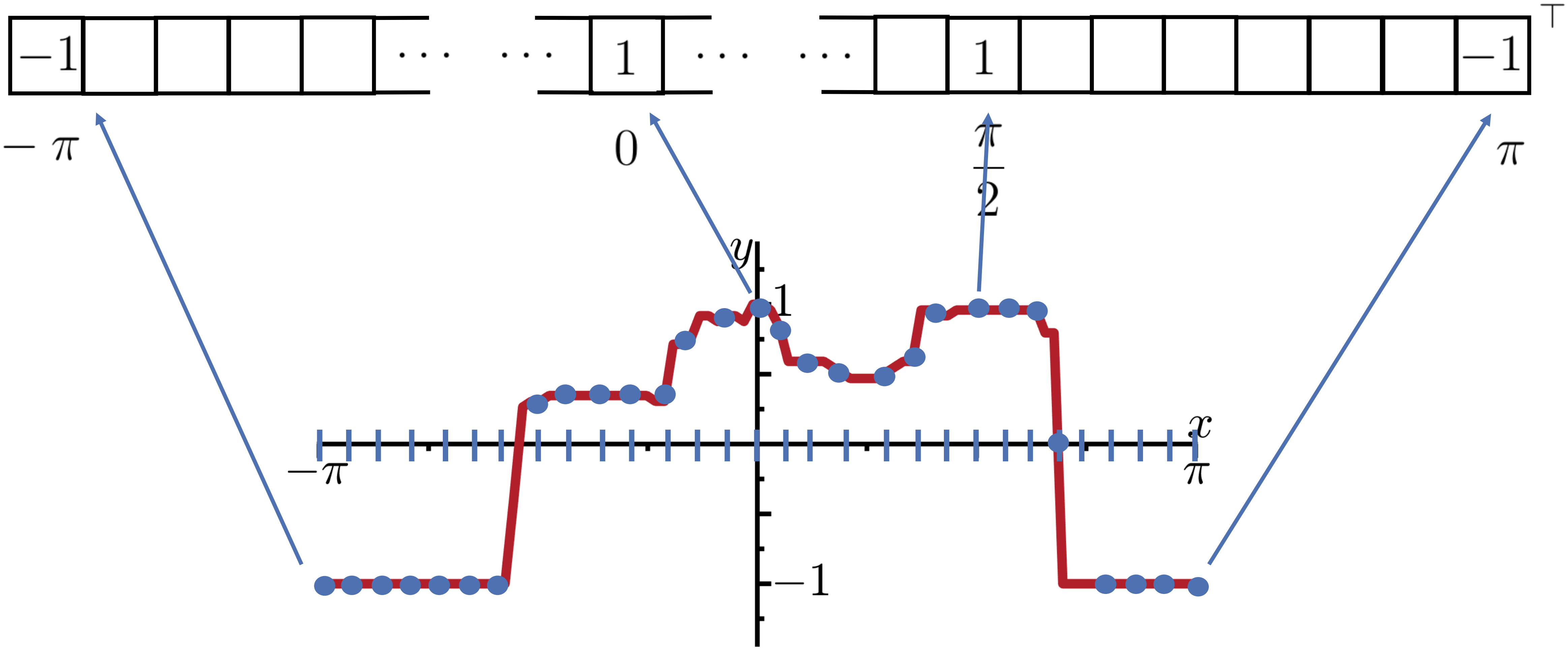
1D function (continuous)

$[-\pi, \pi] \,/\, x$	$\mathbb{R} \,/\, y$
$-\pi$	-1
$\vdots$	$\vdots$
0	1
$\vdots$	$\vdots$
$\frac{\pi}{2}$	1
$\vdots$	$\vdots$
π	-1

$$f : [-\pi, \pi] \longrightarrow \mathbb{R}$$

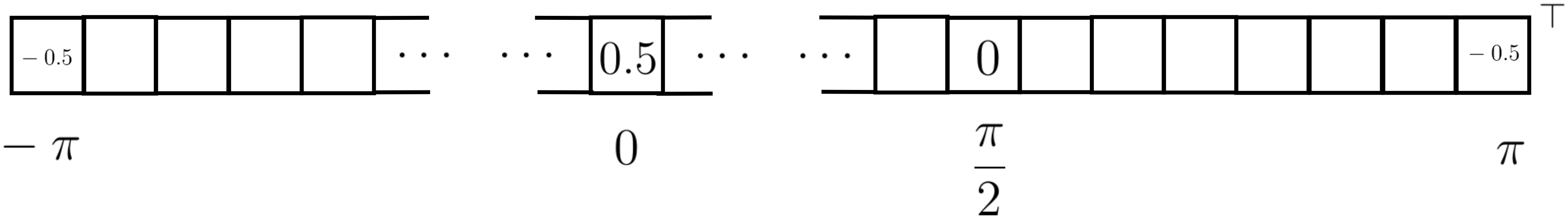


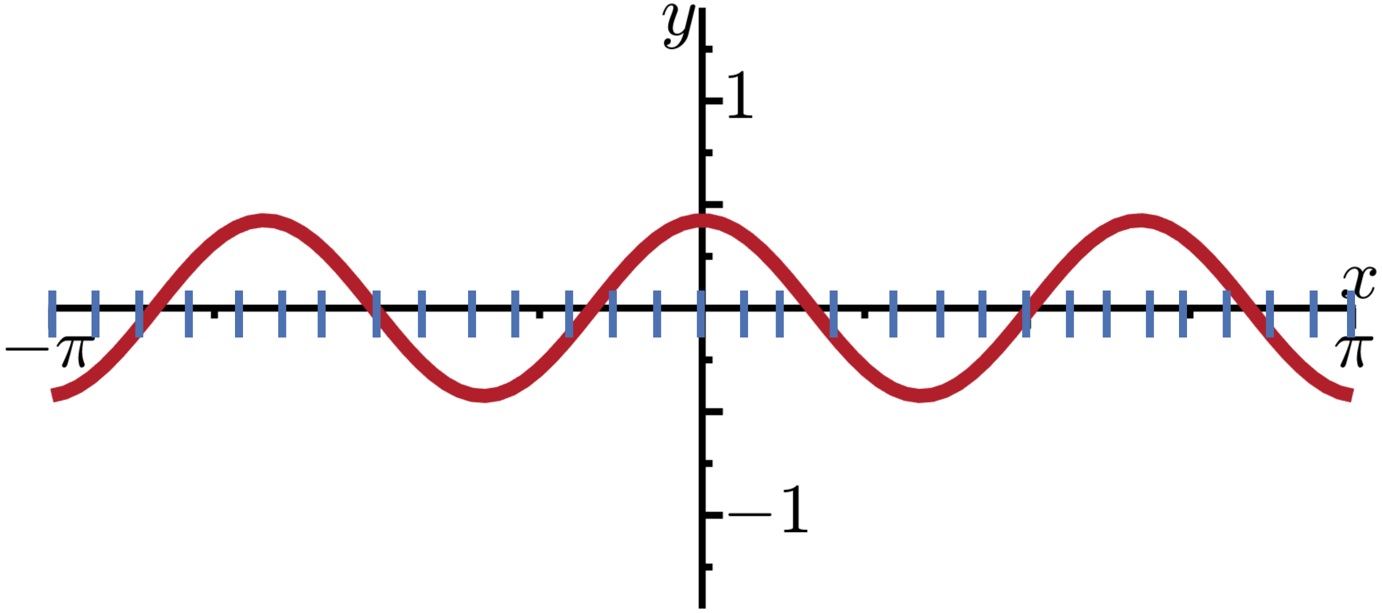
Discretization of a 1D function



Sampling n points equal distributed on the interval [-π, π]

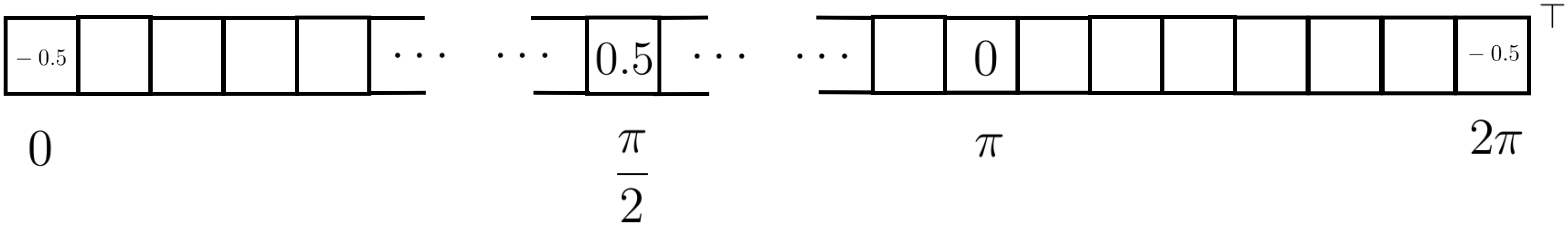
Common vectorization of 1D functions

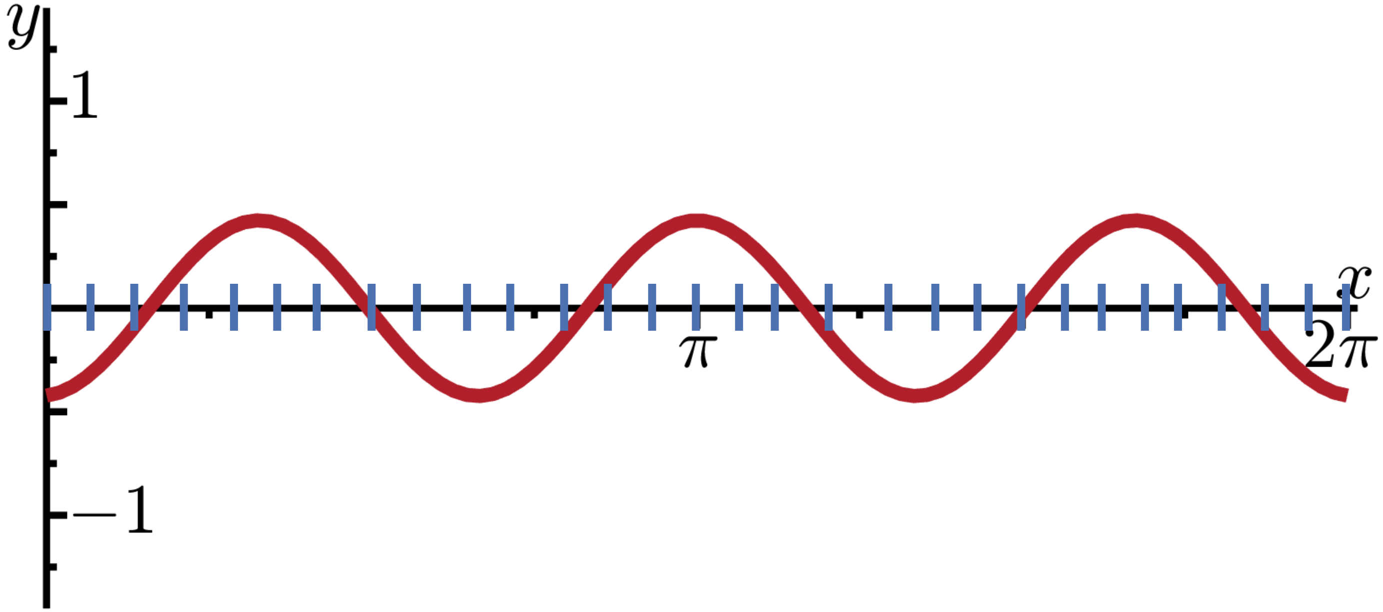




Same representation in a *n*-dimensional vector

Common template of 1D functions

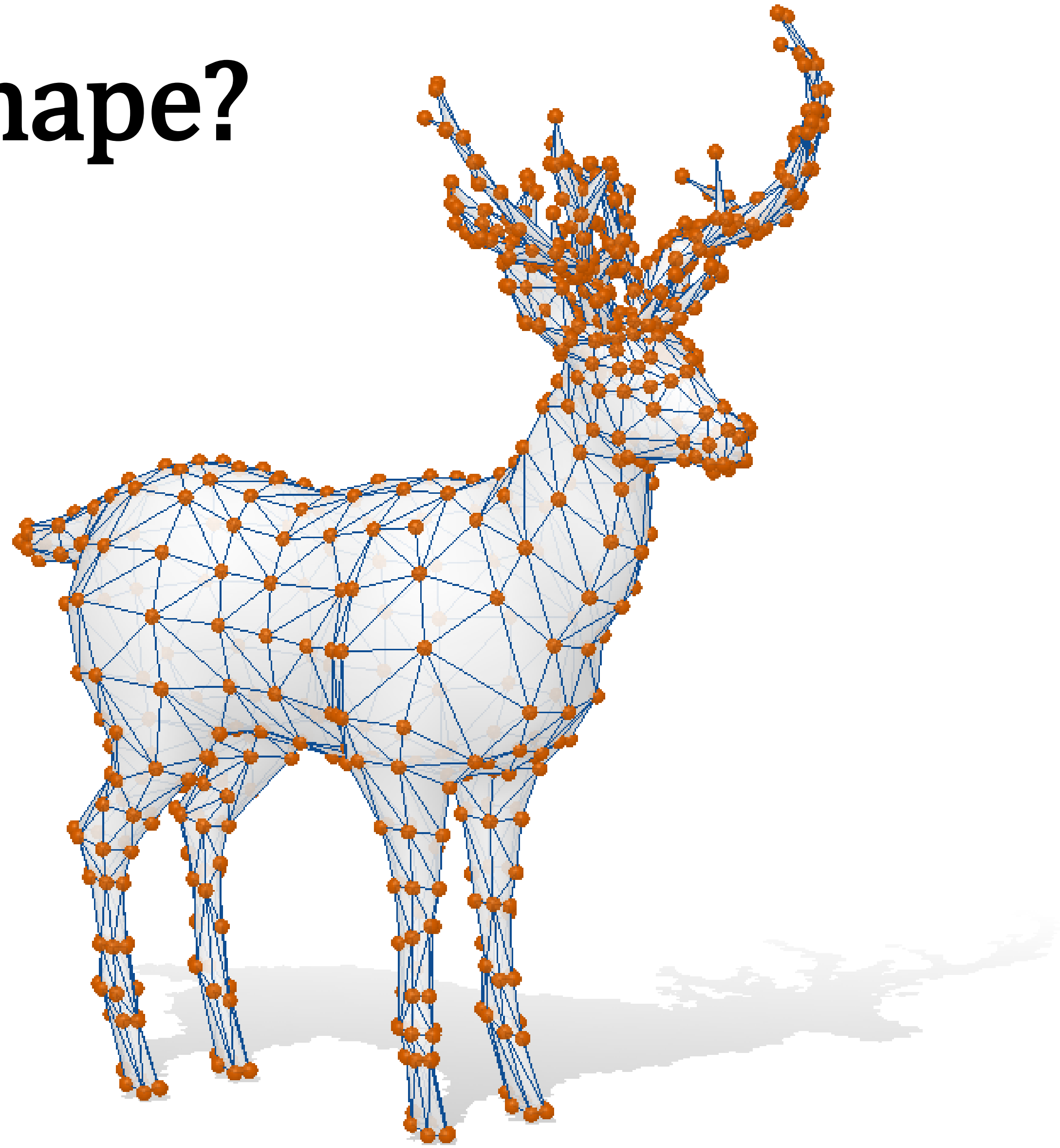




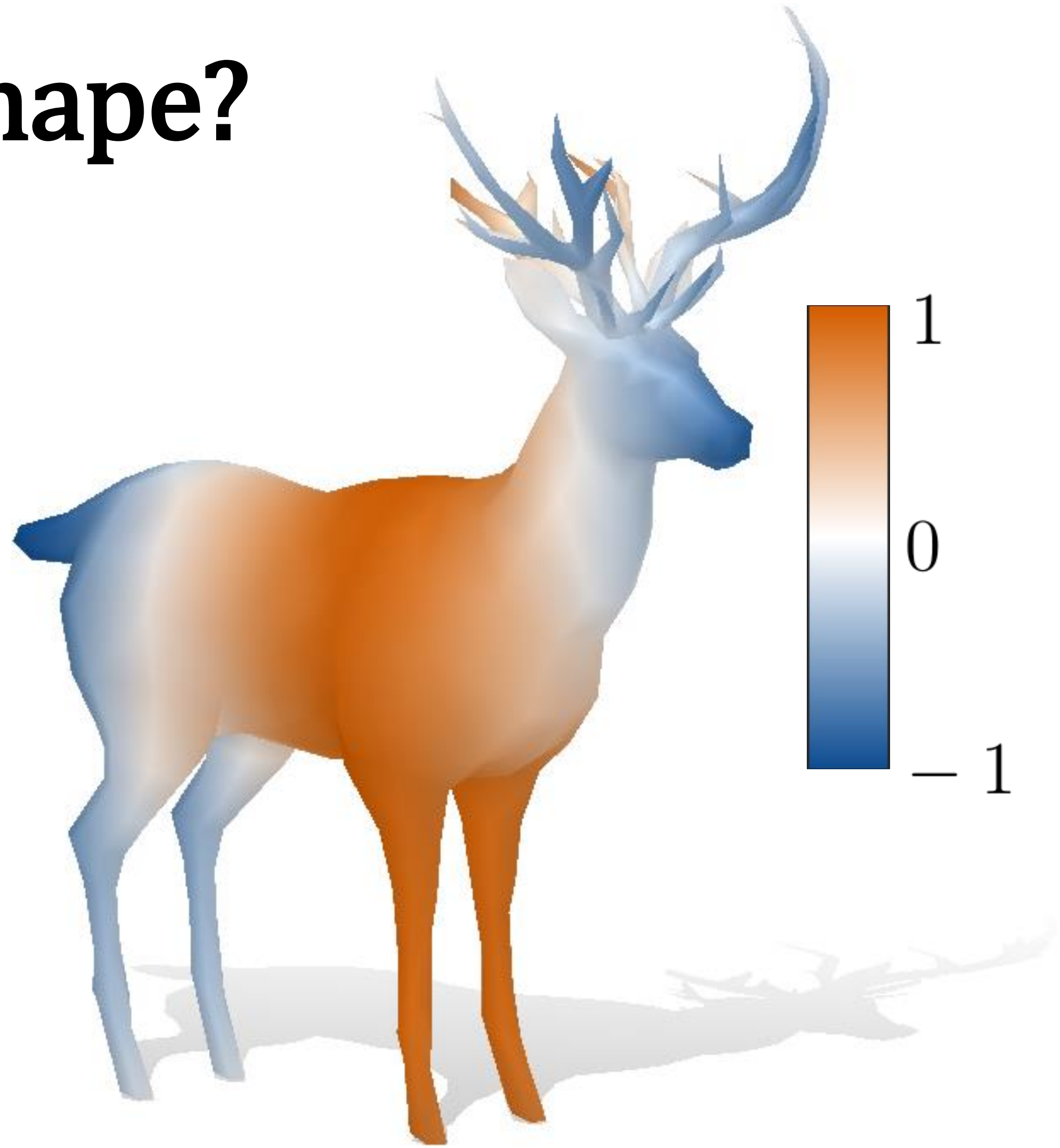
Only the affine transformation between [- π, π] and [0, 2 π] is required

What is a function on a shape?

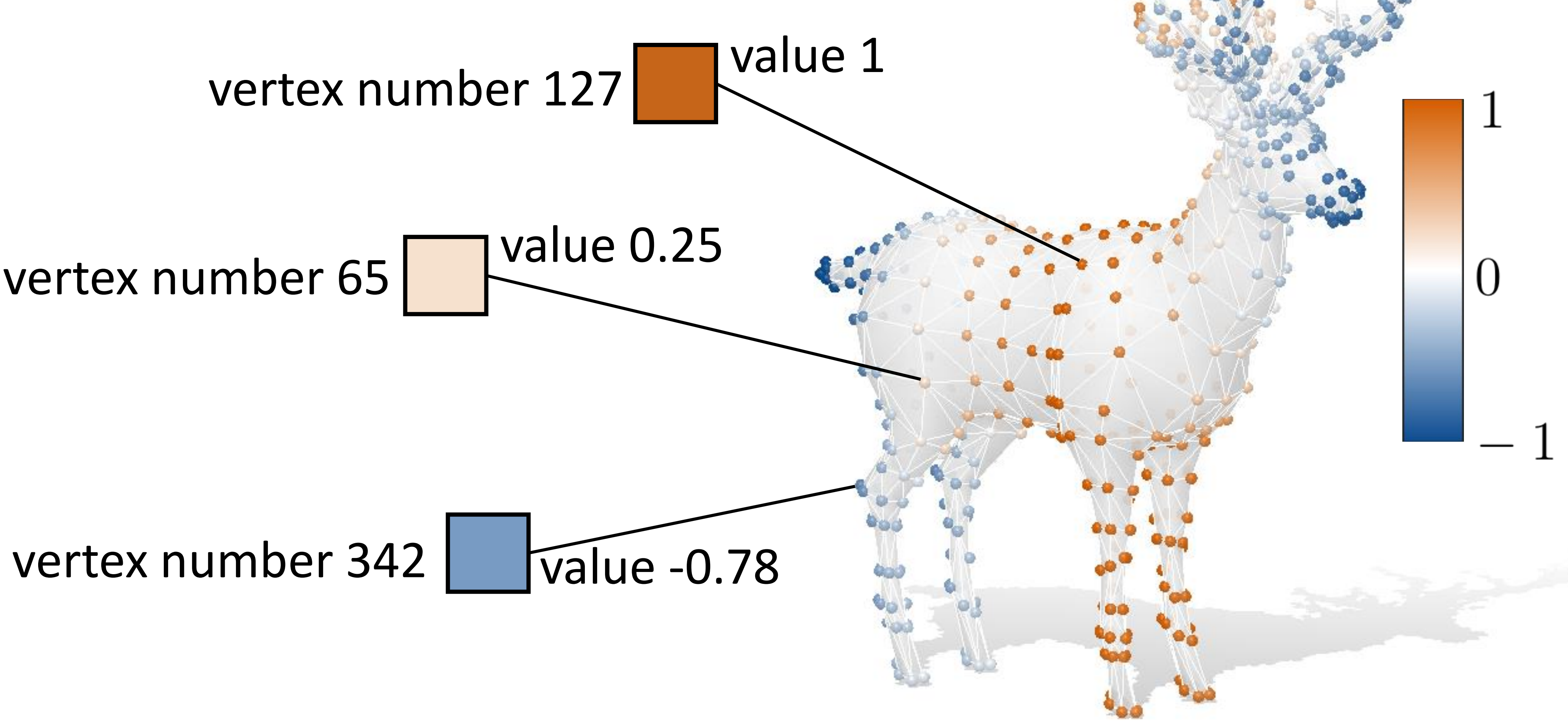
?



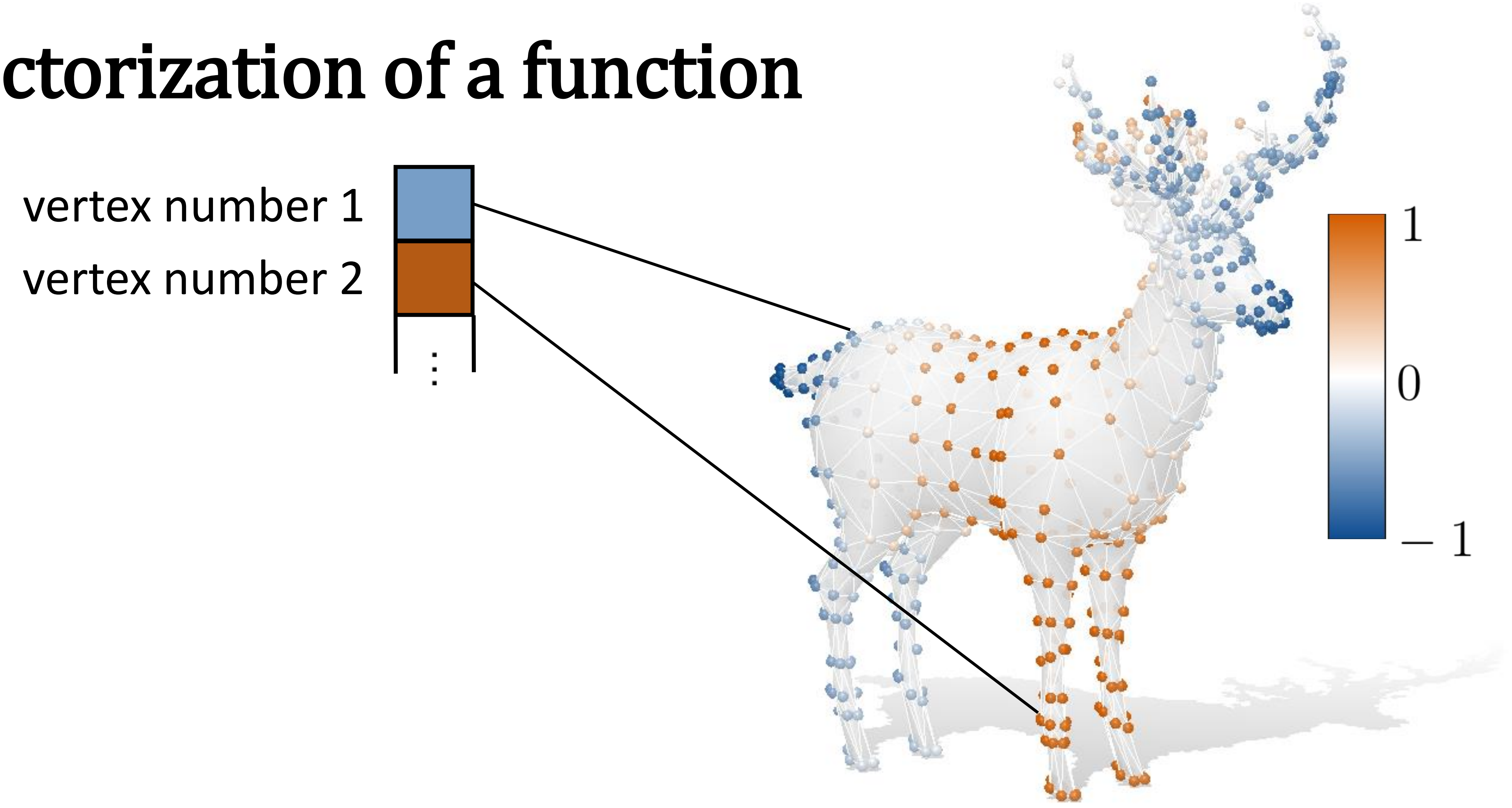
What is a function on a shape?



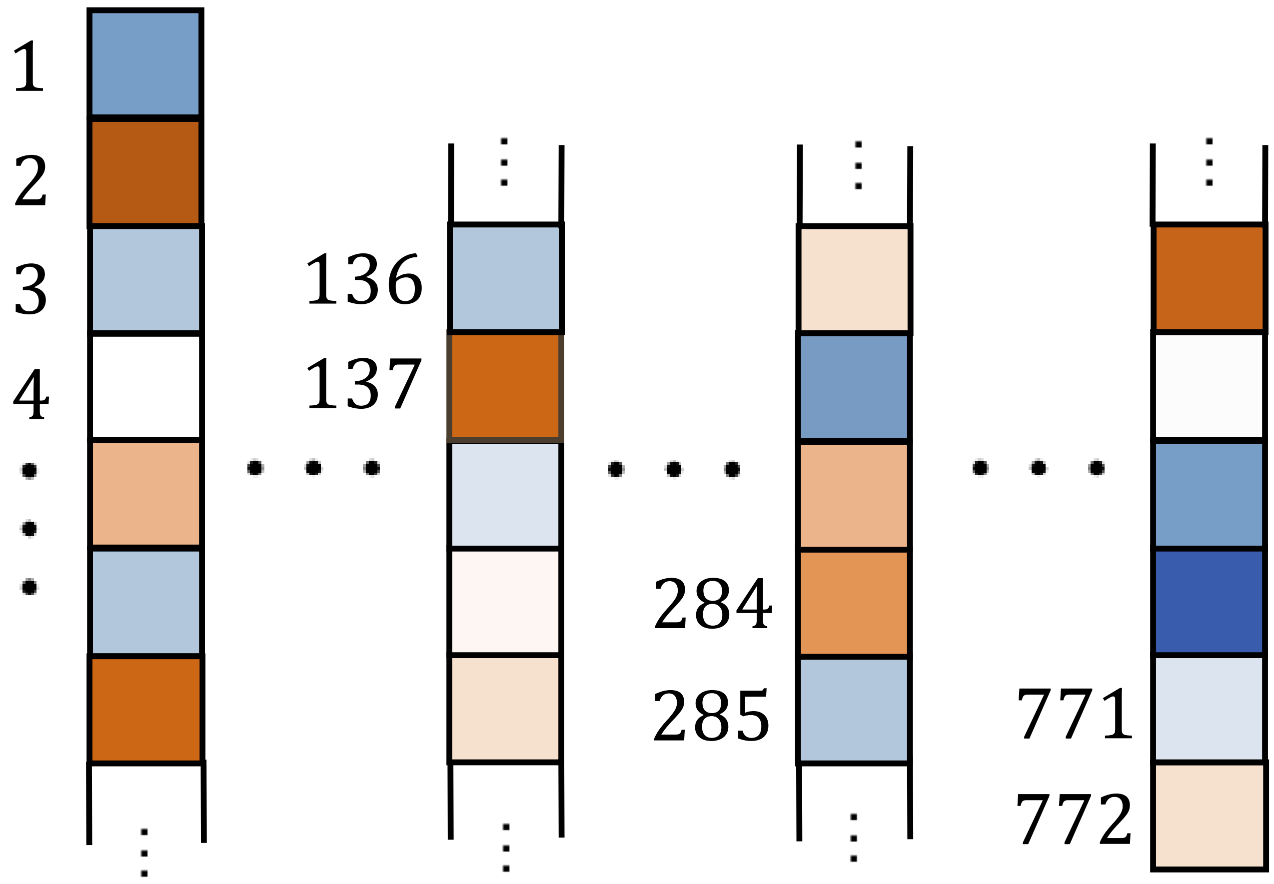
What is a function on a shape?

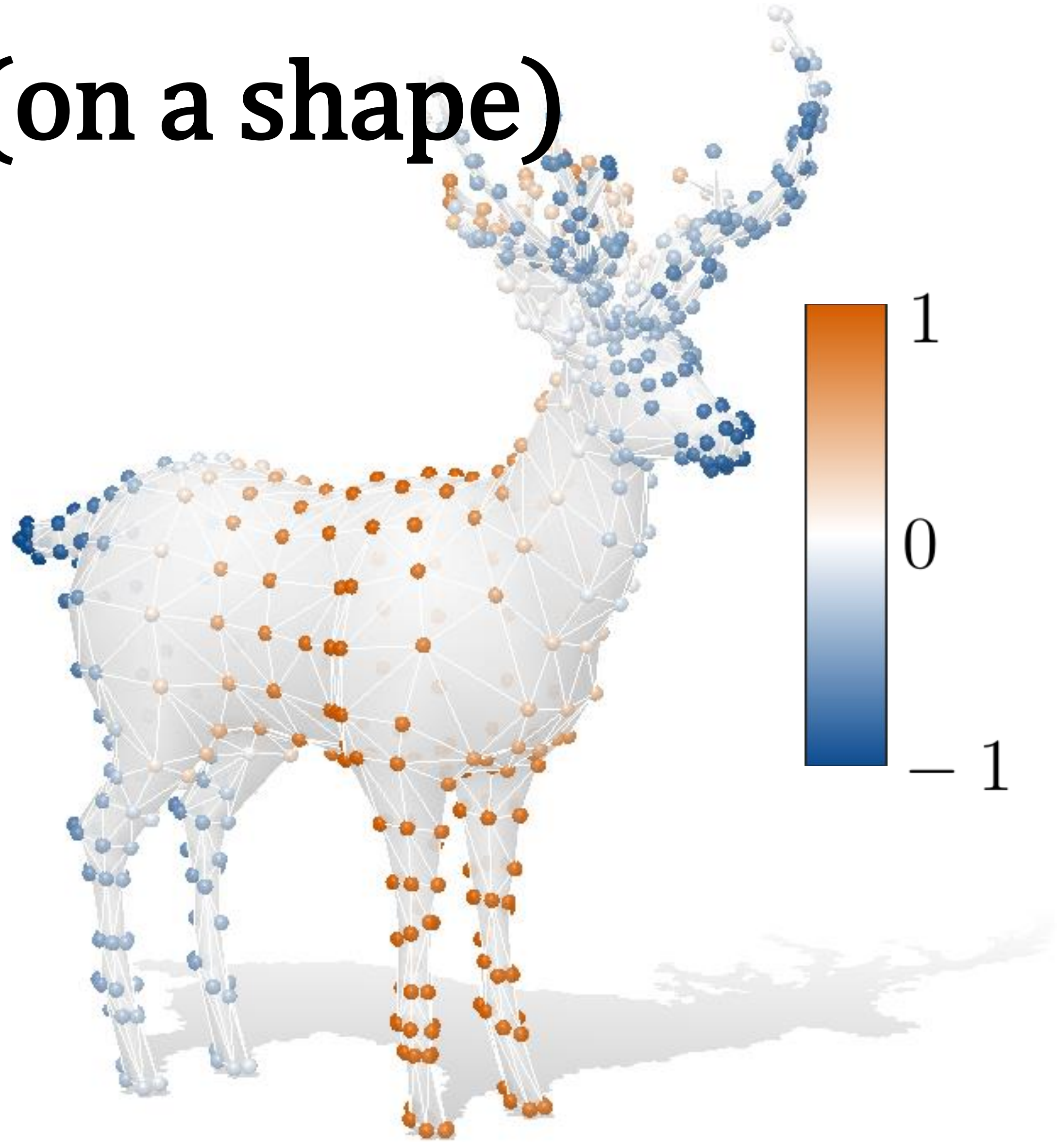


Vectorization of a function

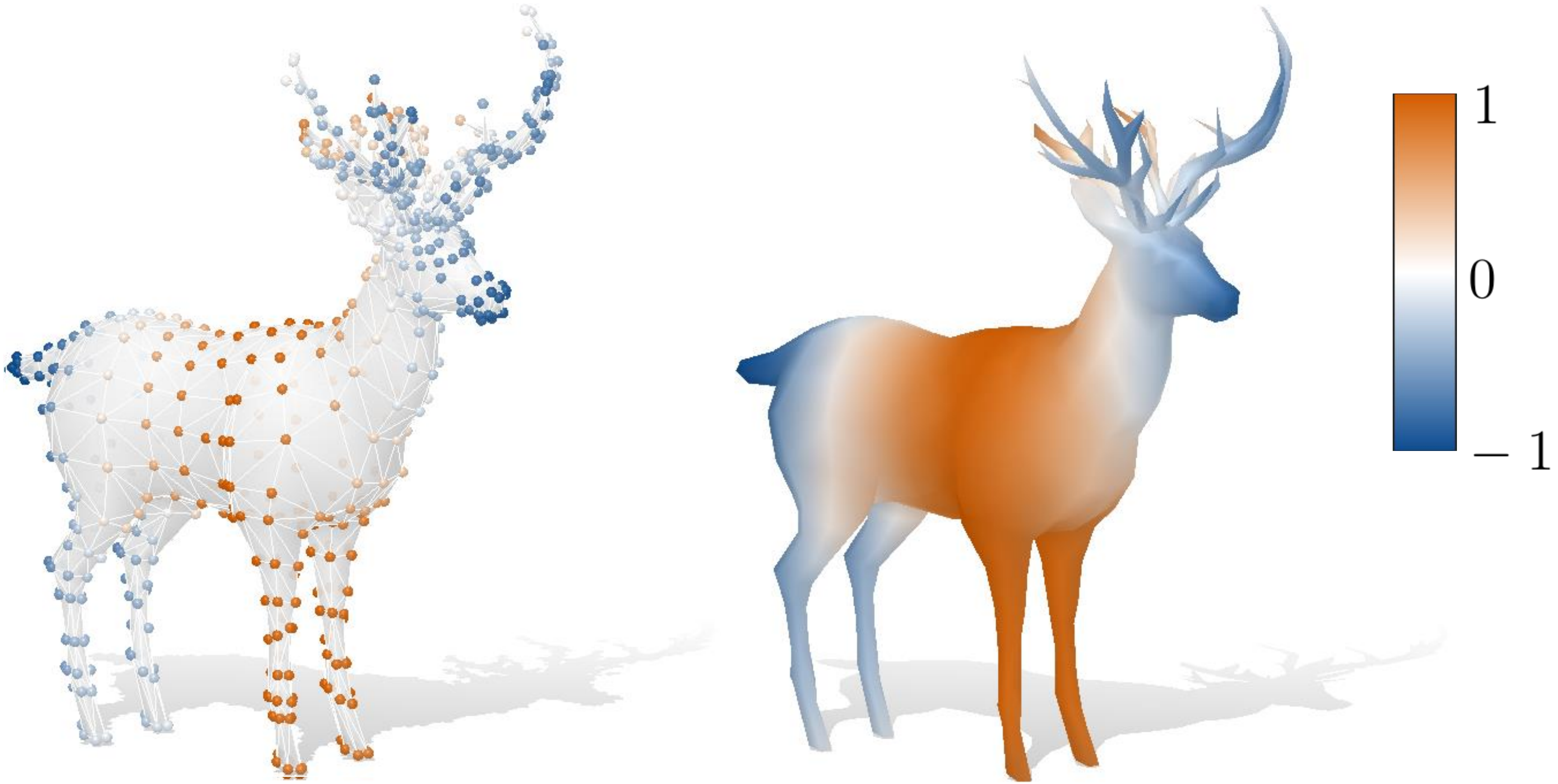


Vectorization of a signal (on a shape)

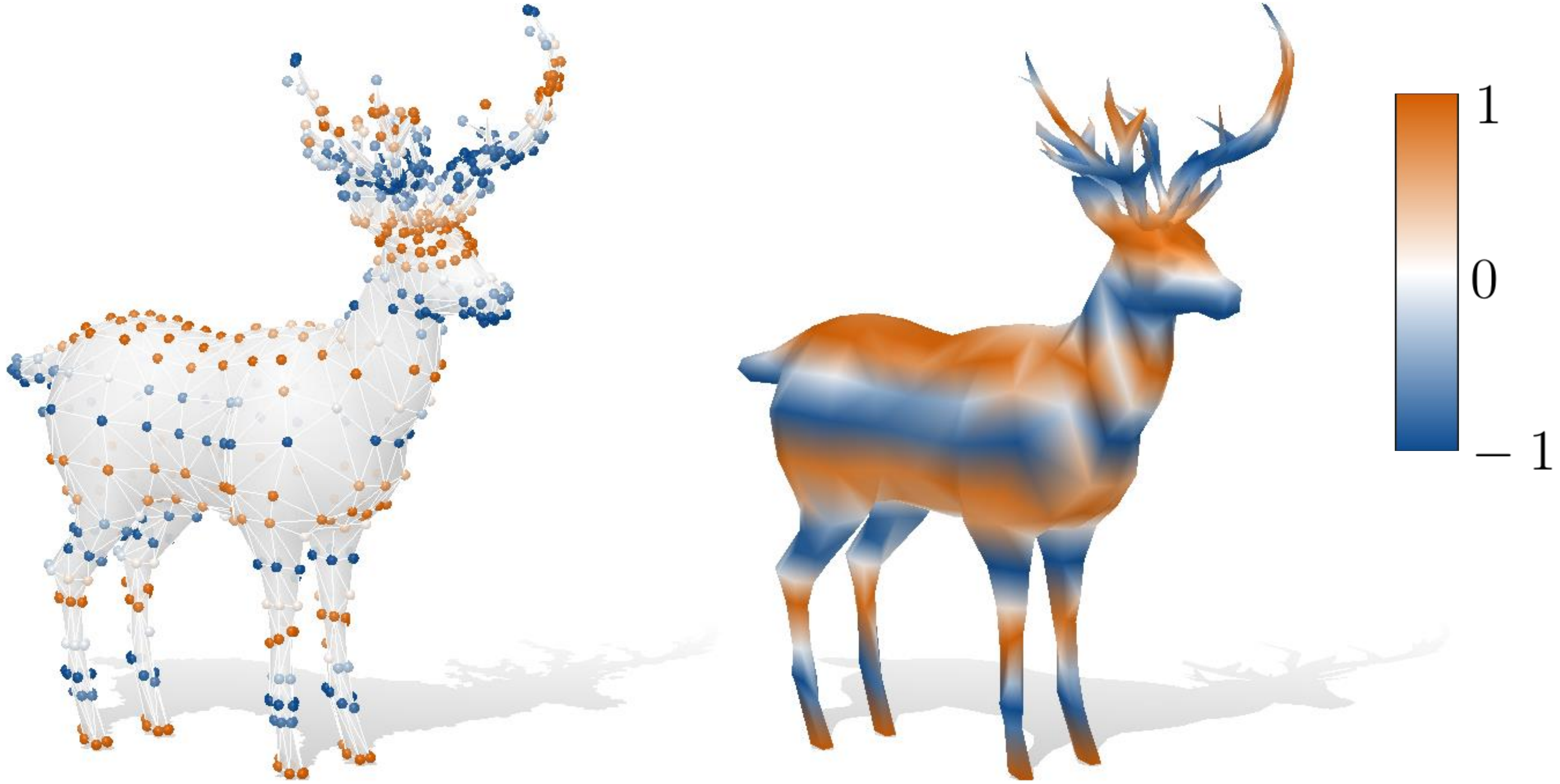




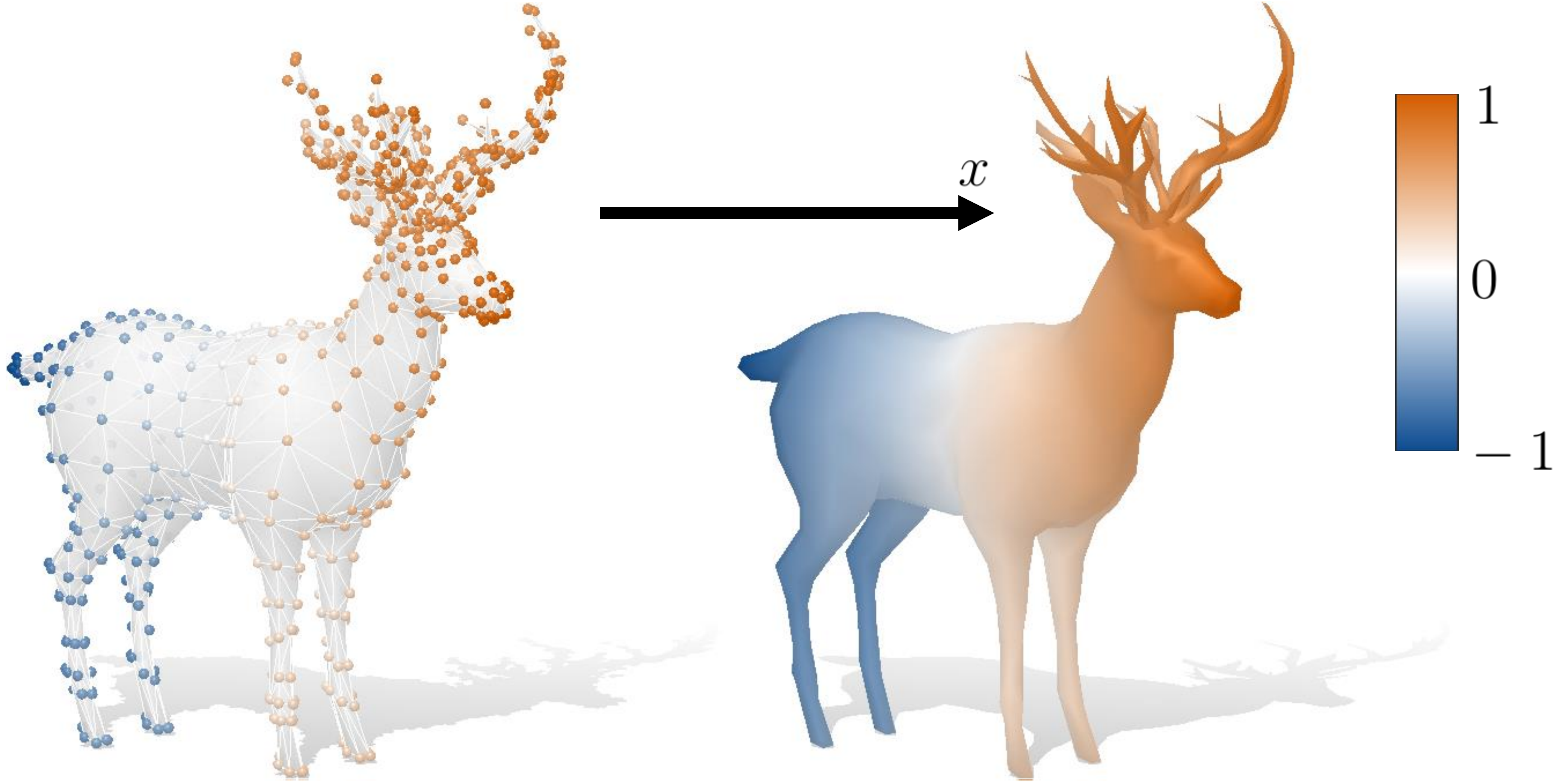
Smooth visualization



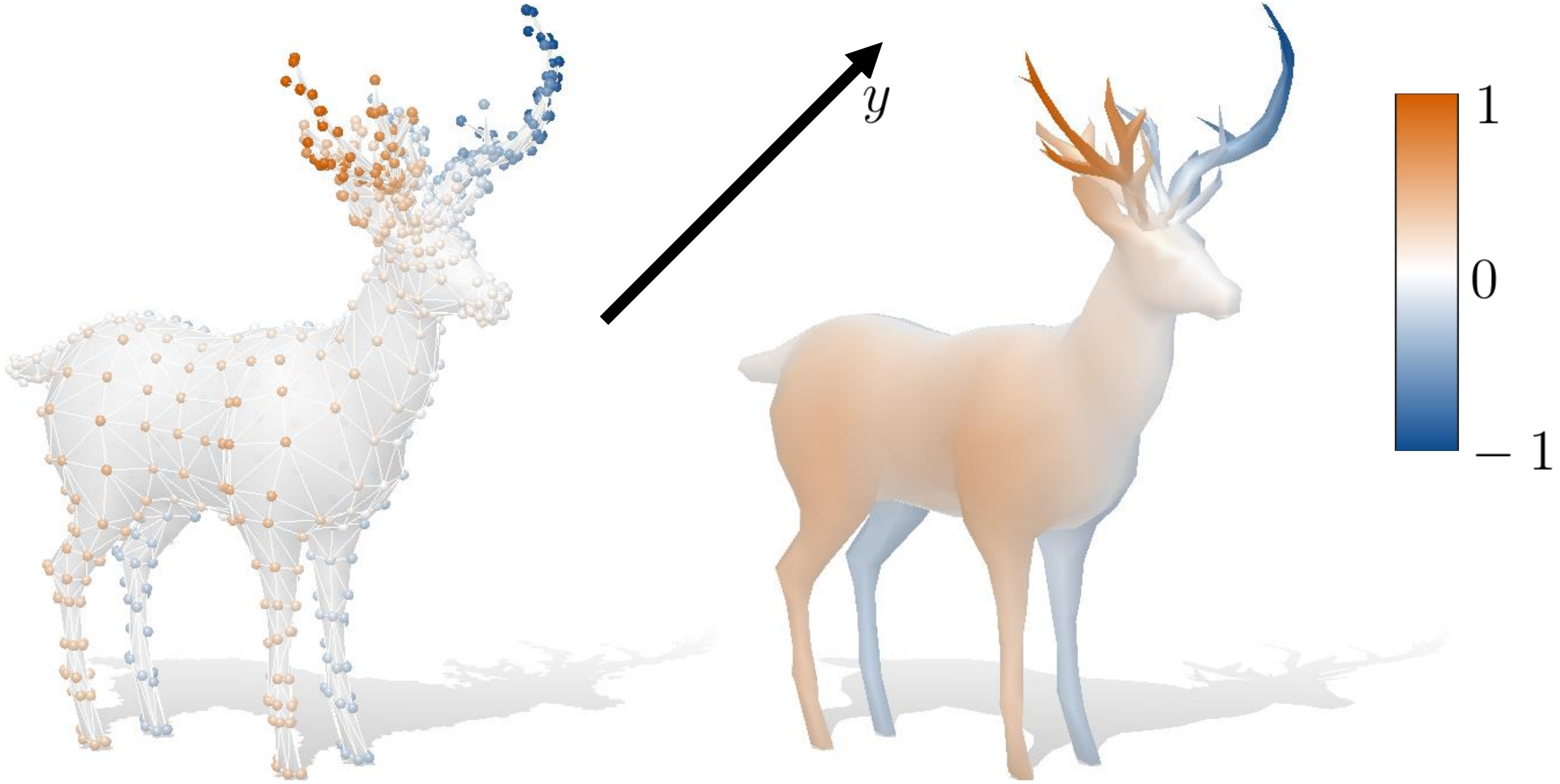
Example of function on a shape



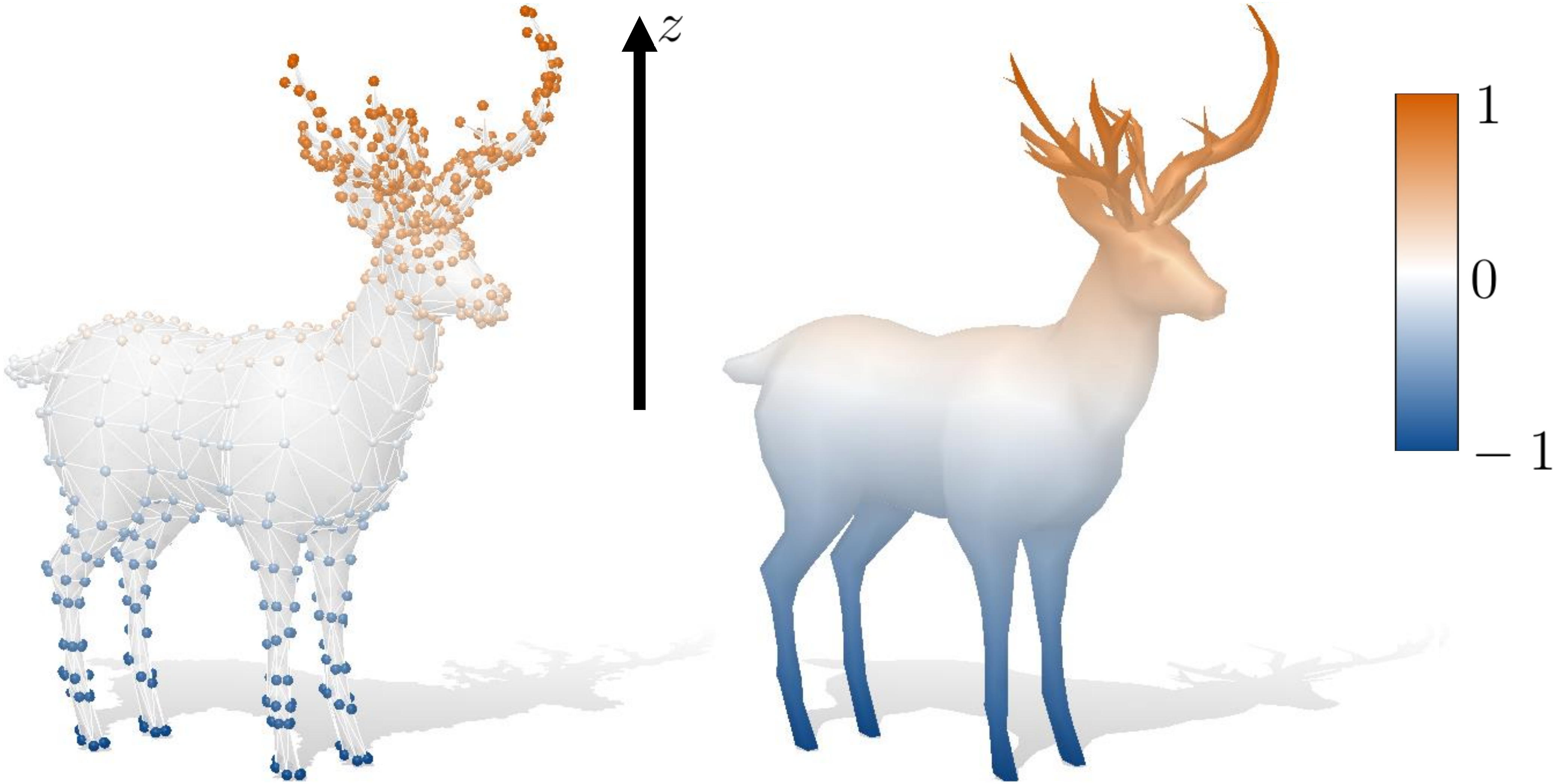
Example of function on a shape



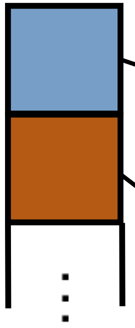
Example of function on a shape

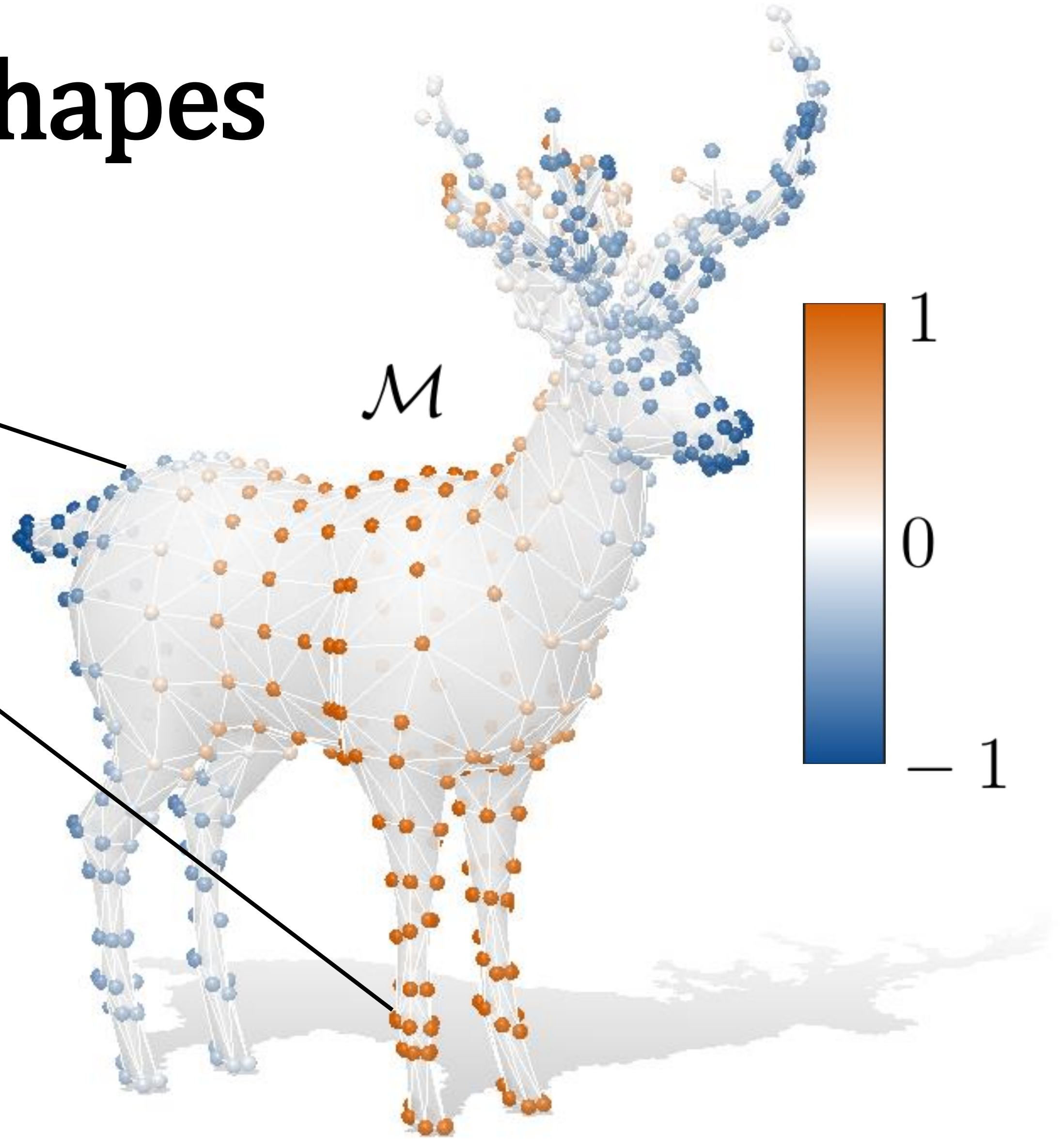


Example of function on a shape



Functional space on 3D shapes





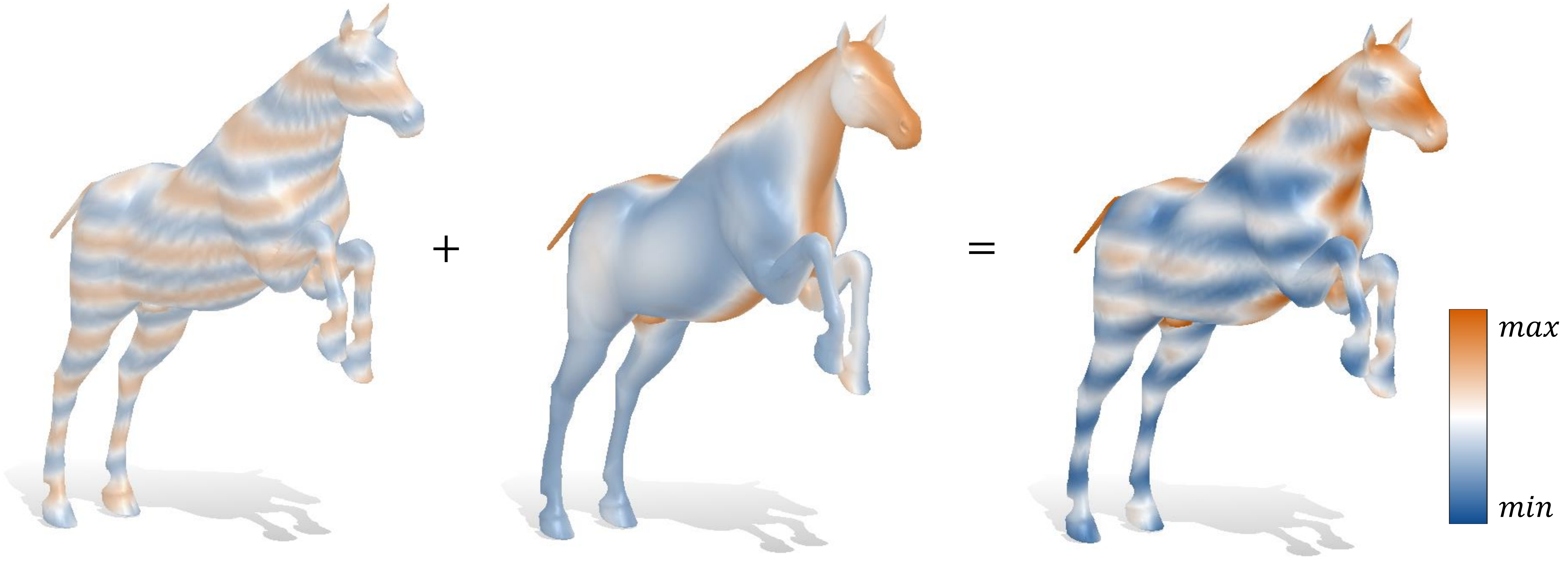
$\mathcal{M} = n$ points/vertices

f corresponds to a vector in $\mathbb{R}^n$

$F(\mathcal{M}) =$ the space of functions on $\mathcal{M}$

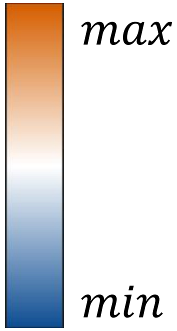
$F(\mathcal{M}) = \mathbb{R}^n$

Sum of functions

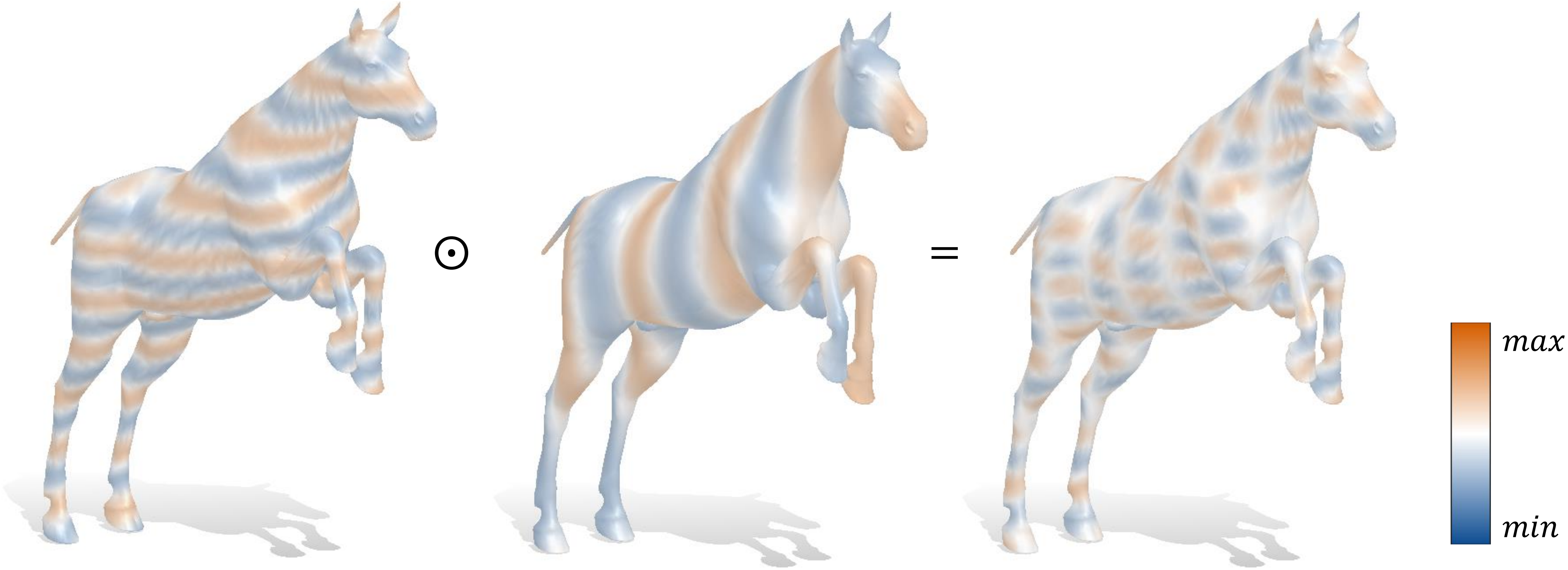


Scalar product

$3 \cdot$  $=$ 

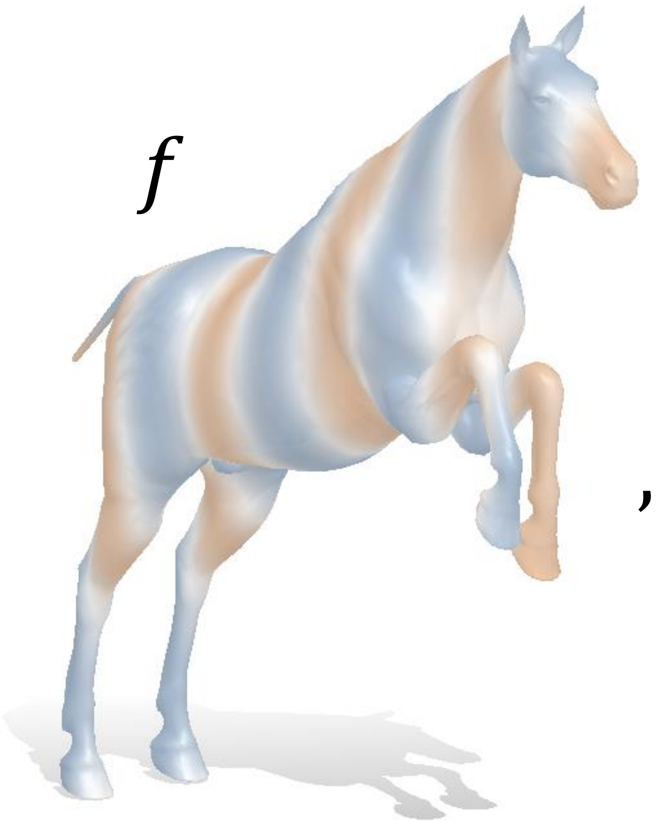


Pointwise product

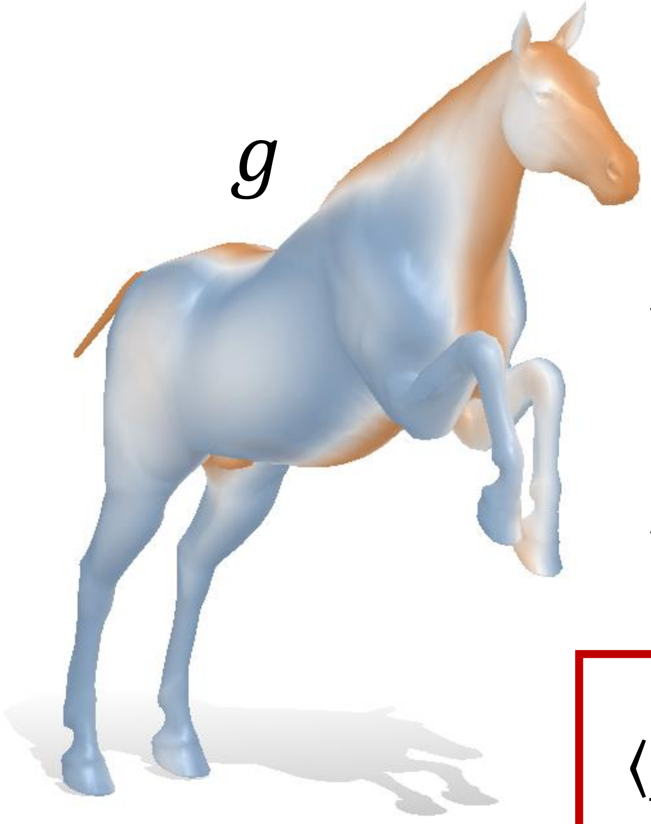


Inner product

f



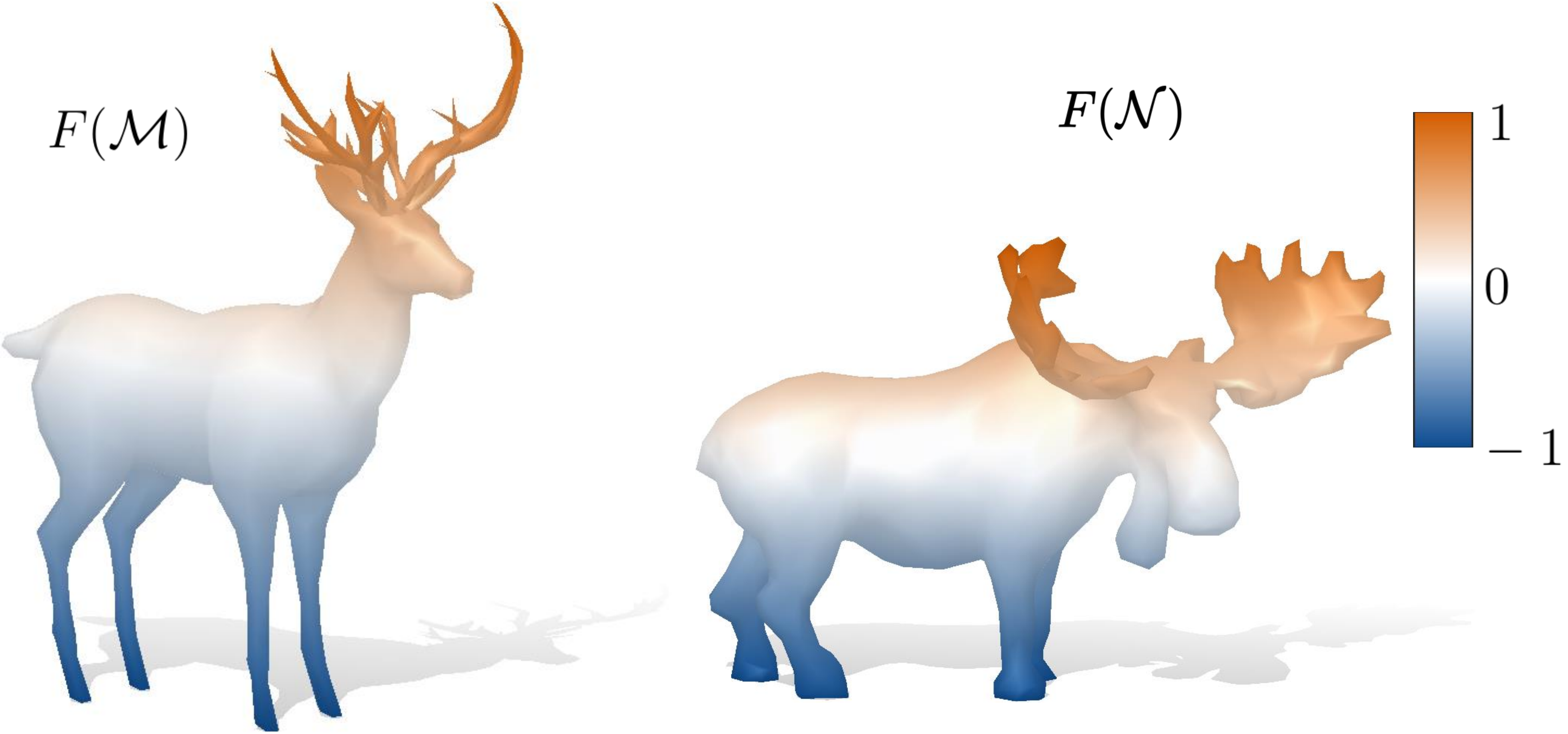
g



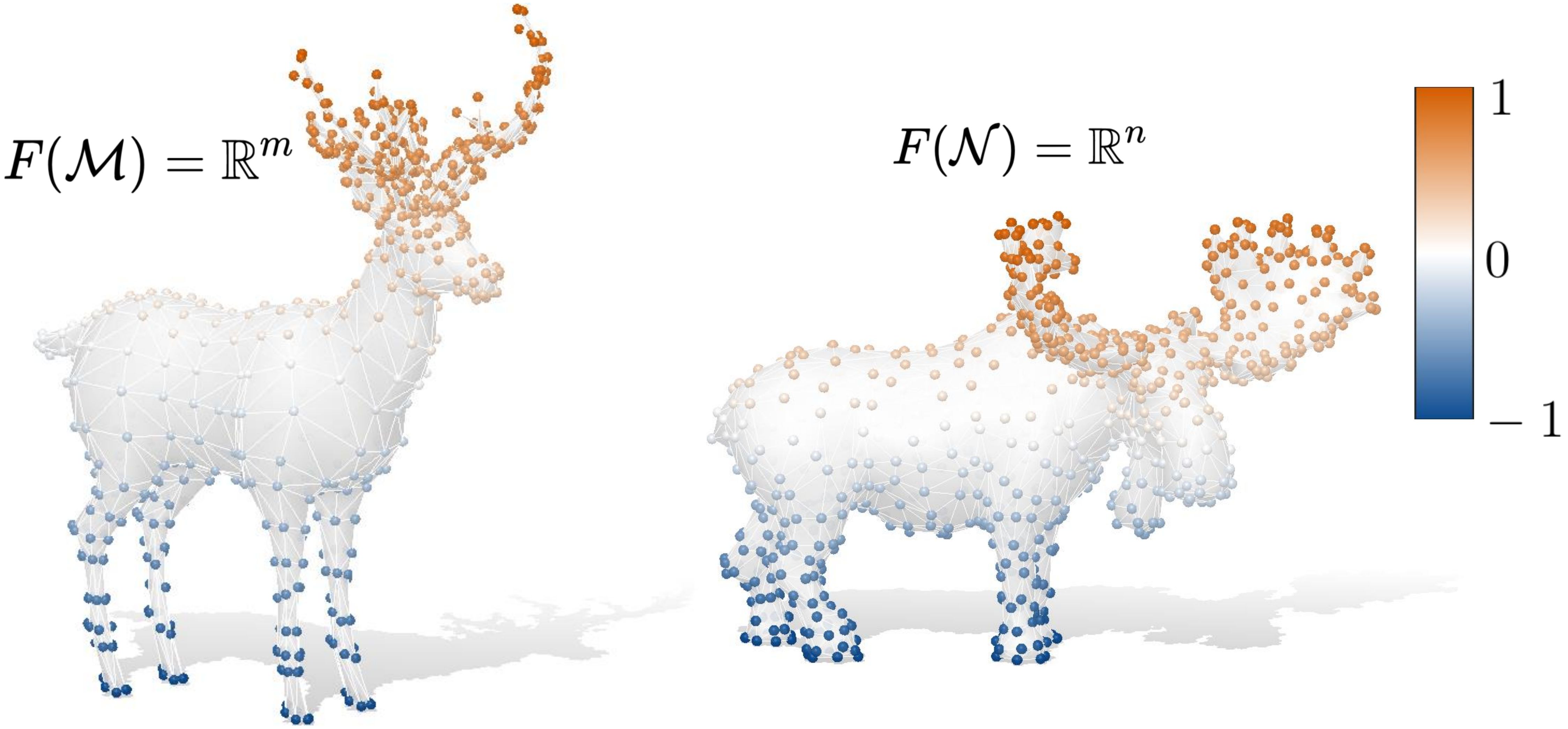
$$\langle f, g \rangle_{\mathcal{M}} = \mathbf{11.883}$$

$$\langle f, g \rangle_{\mathcal{M}} = \int_{\mathcal{M}} f(x) g(x) d\mu(x)$$

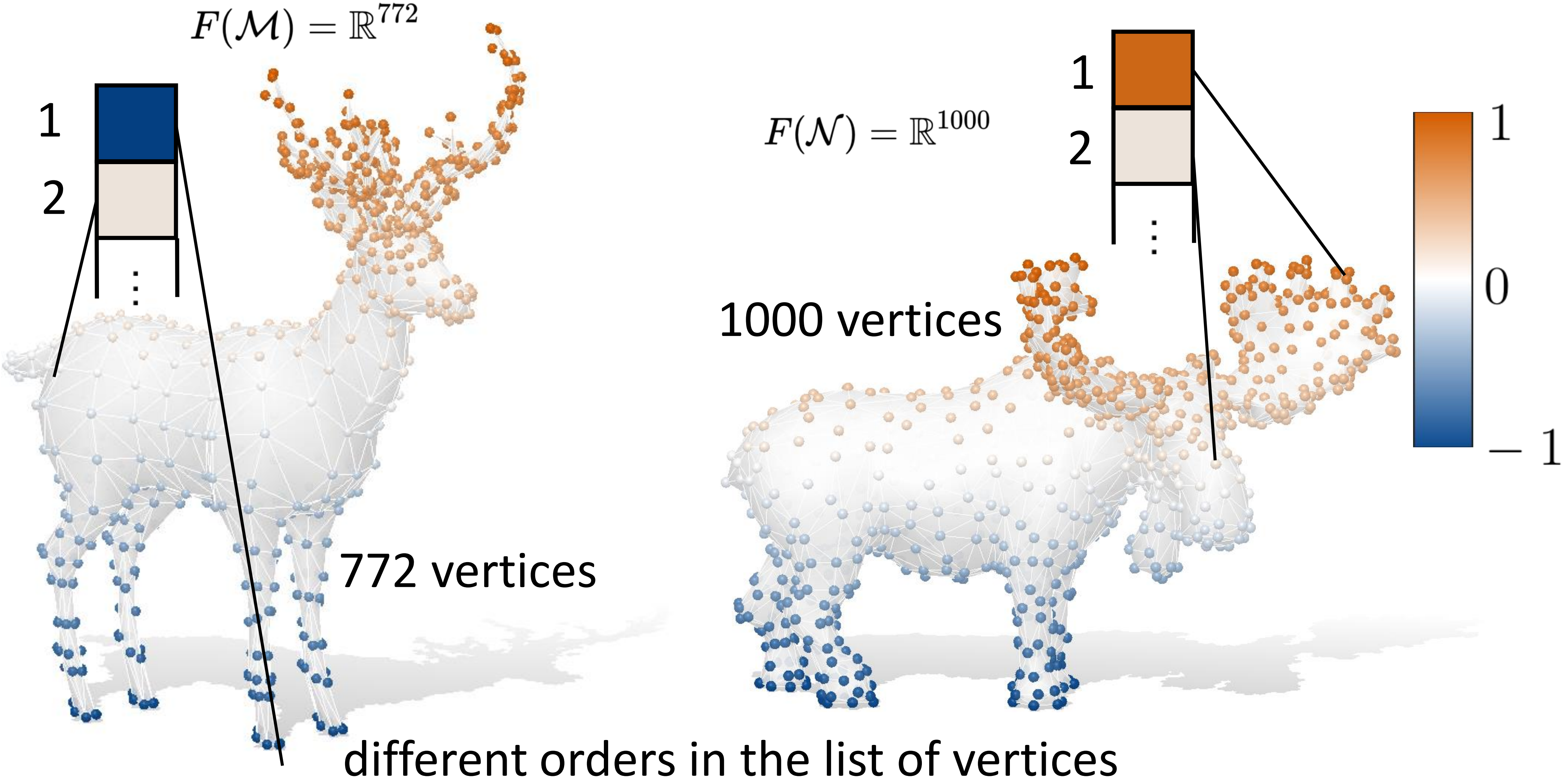
Functions on different domains



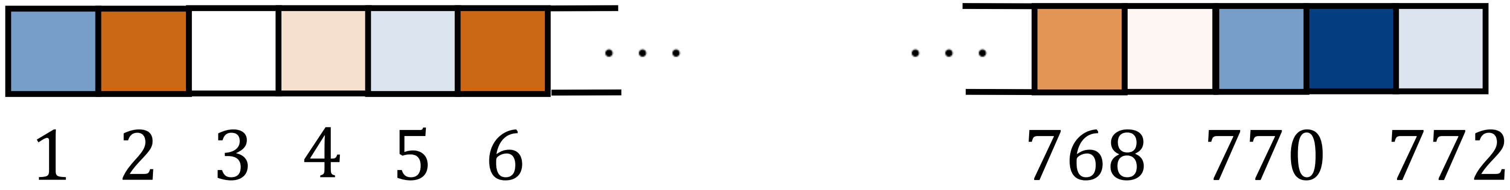
Not a common template

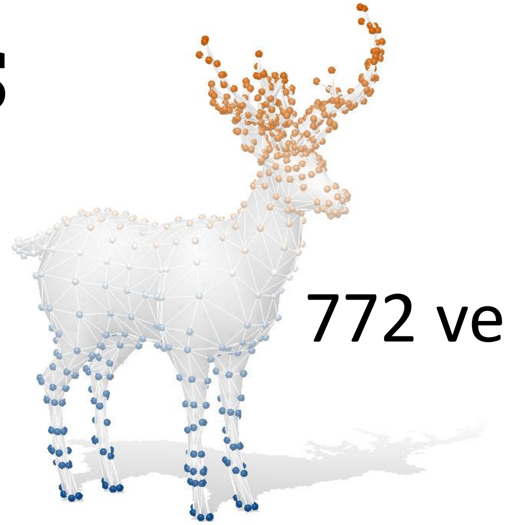


Not a common vectorization

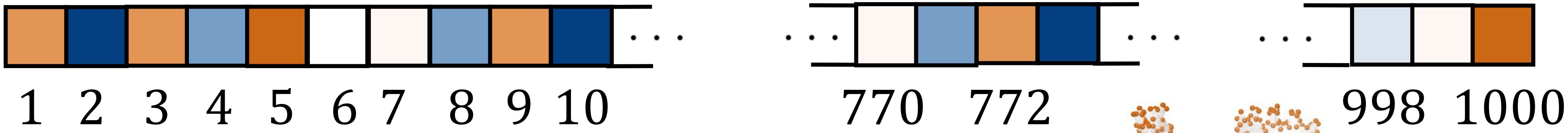


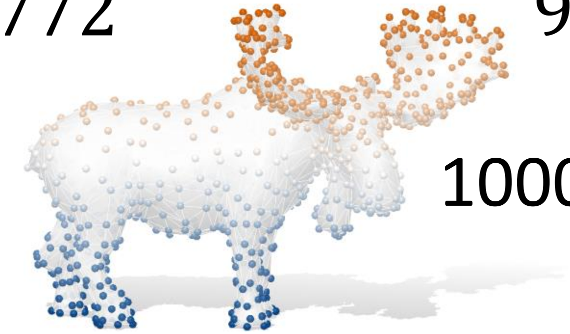
Same function different vectors





772 vertices



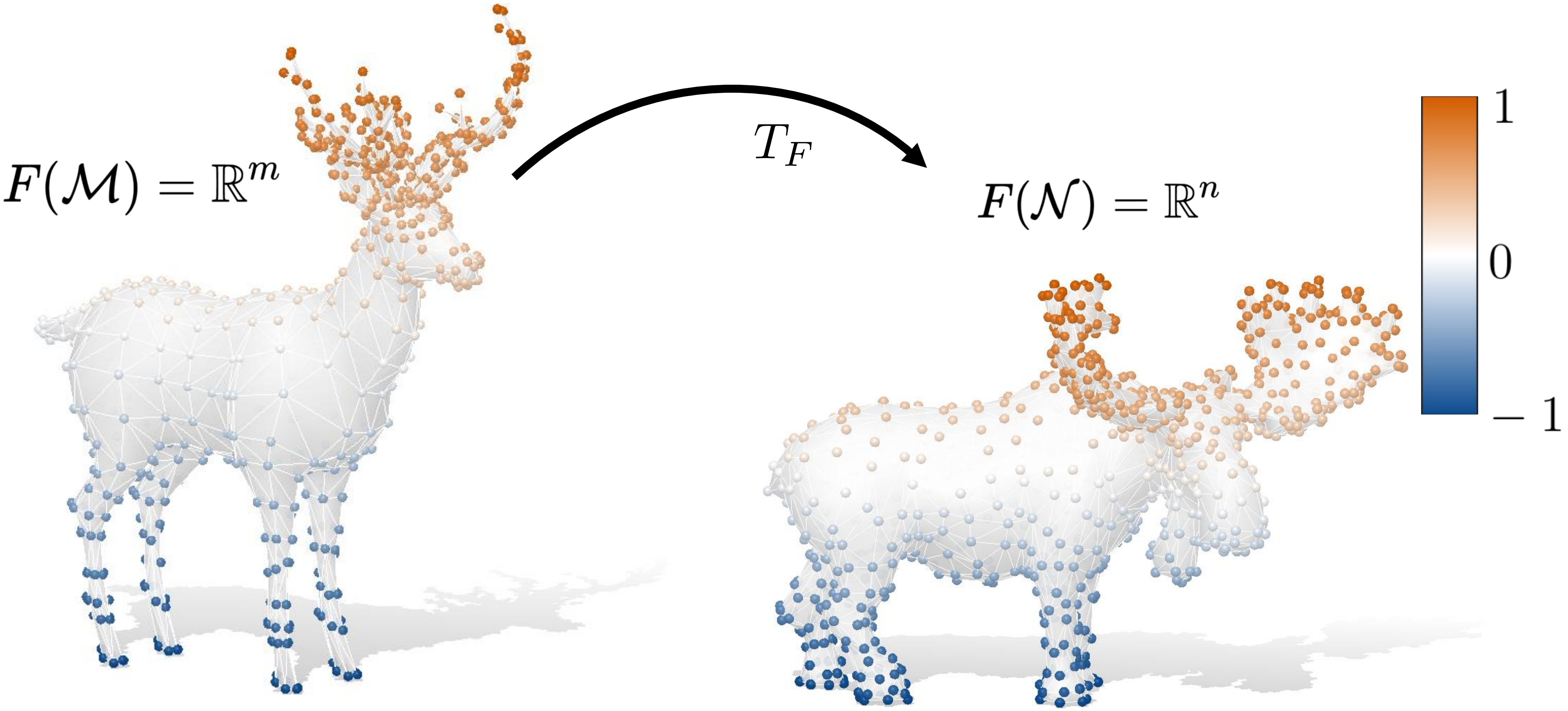


1000 vertices

These representations are not comparable!

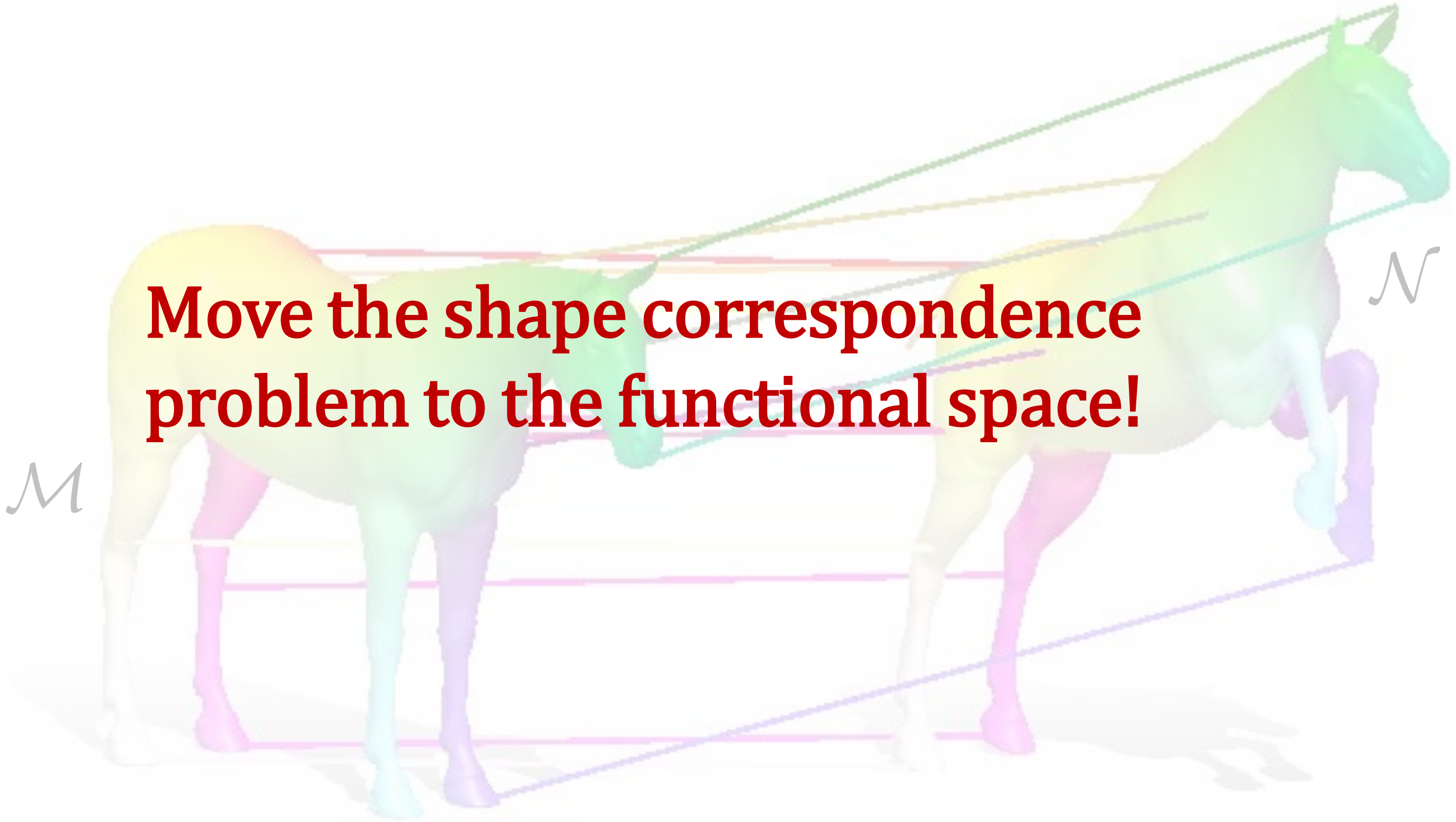
Functional map

$$T_F : F(\mathcal{M}) \longrightarrow F(\mathcal{N})$$



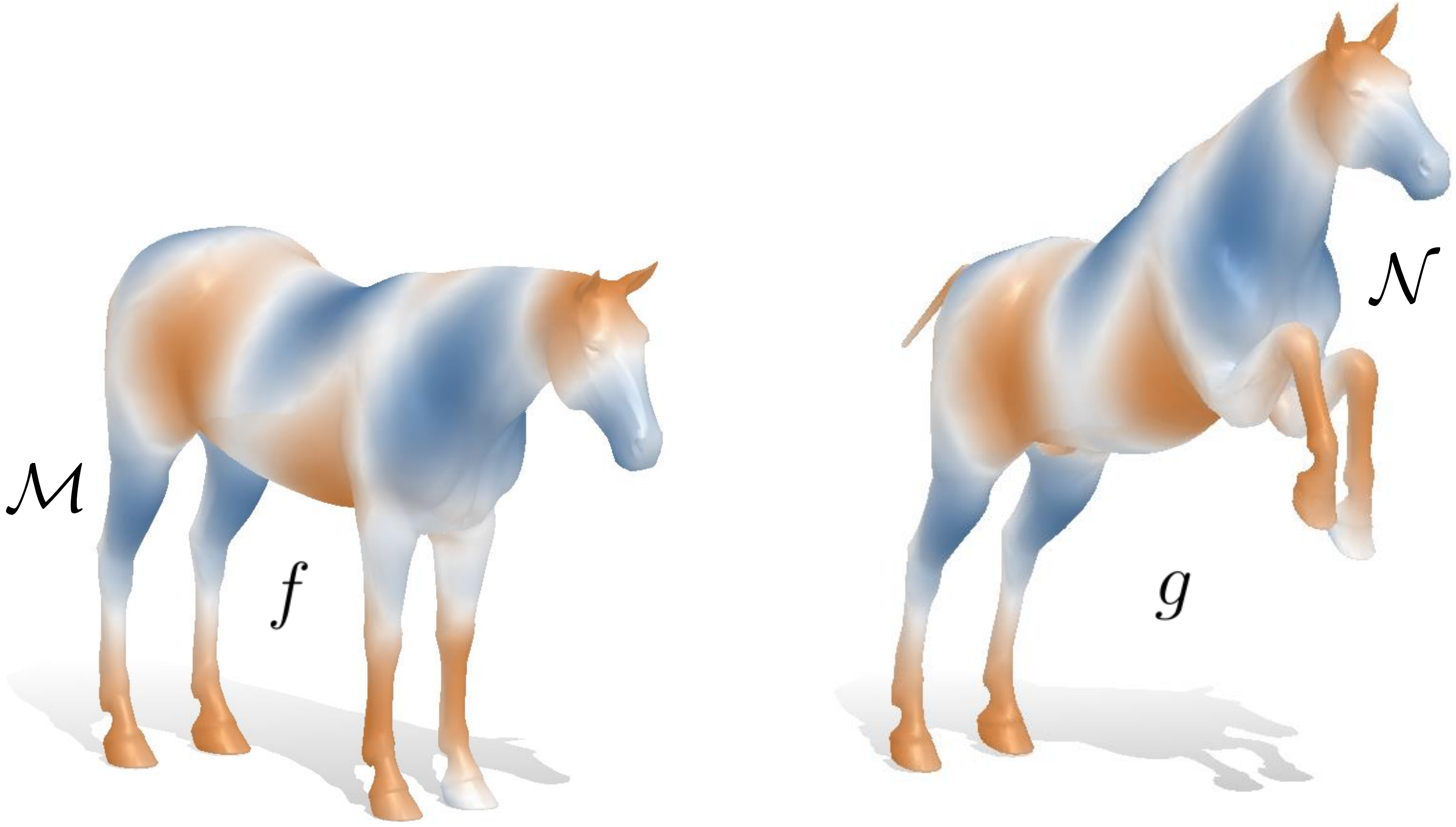
Functional shape correspondence

Shape correspondence



The problem to find a point-to-point map between $\mathcal{M}$ and $\mathcal{N}$

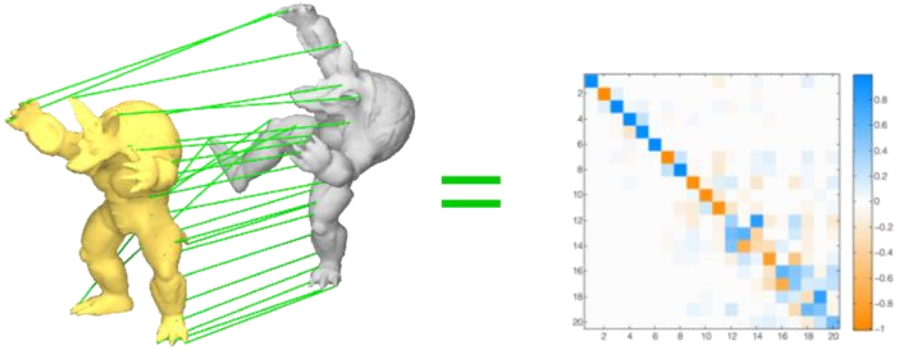
Corresponding functions



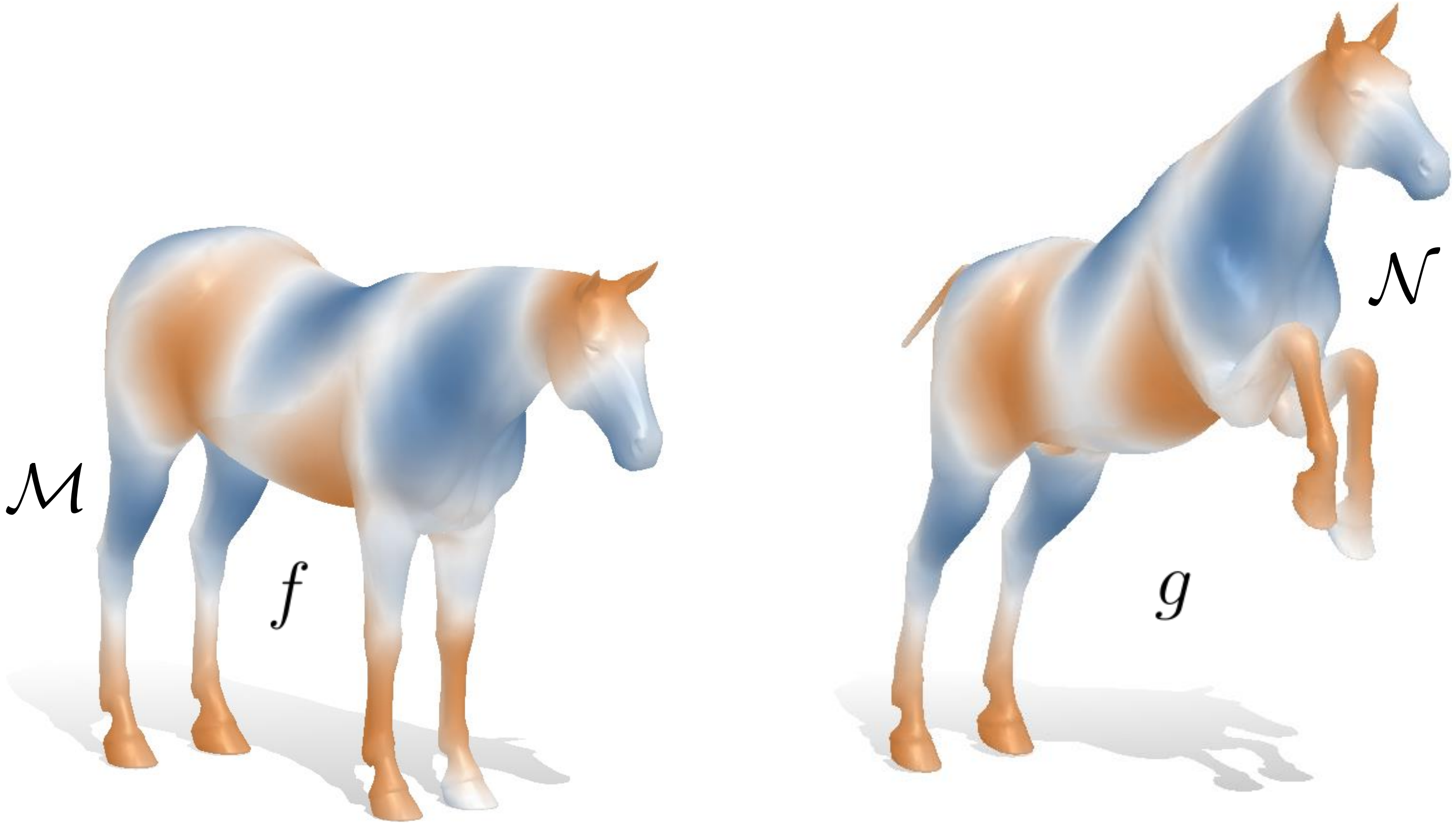
Functional Map Representation

We would like to define a representation of shape maps that is more amenable to direct optimization.

1. A compact representation for “natural” maps.
2. Inherently global and multi-scale.
3. Handles uncertainty and ambiguity gracefully.
4. Allows efficient manipulations (averaging, composition).
5. Leads to simple (linear) optimization problems.

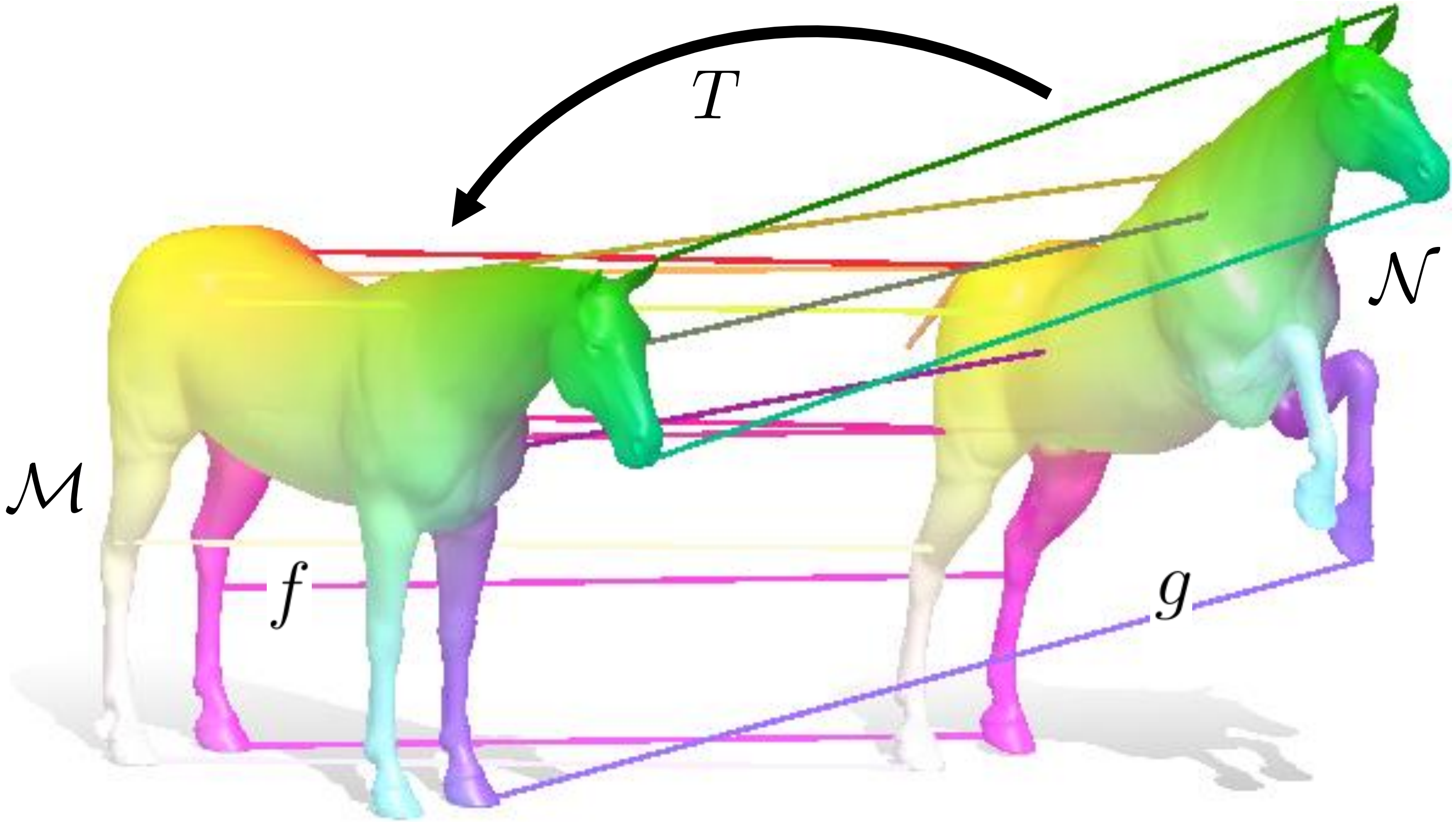


Corresponding functions



corresponding = arise from a point-to-point map

A point-to-point map $T : \mathcal{N} \to \mathcal{M}$

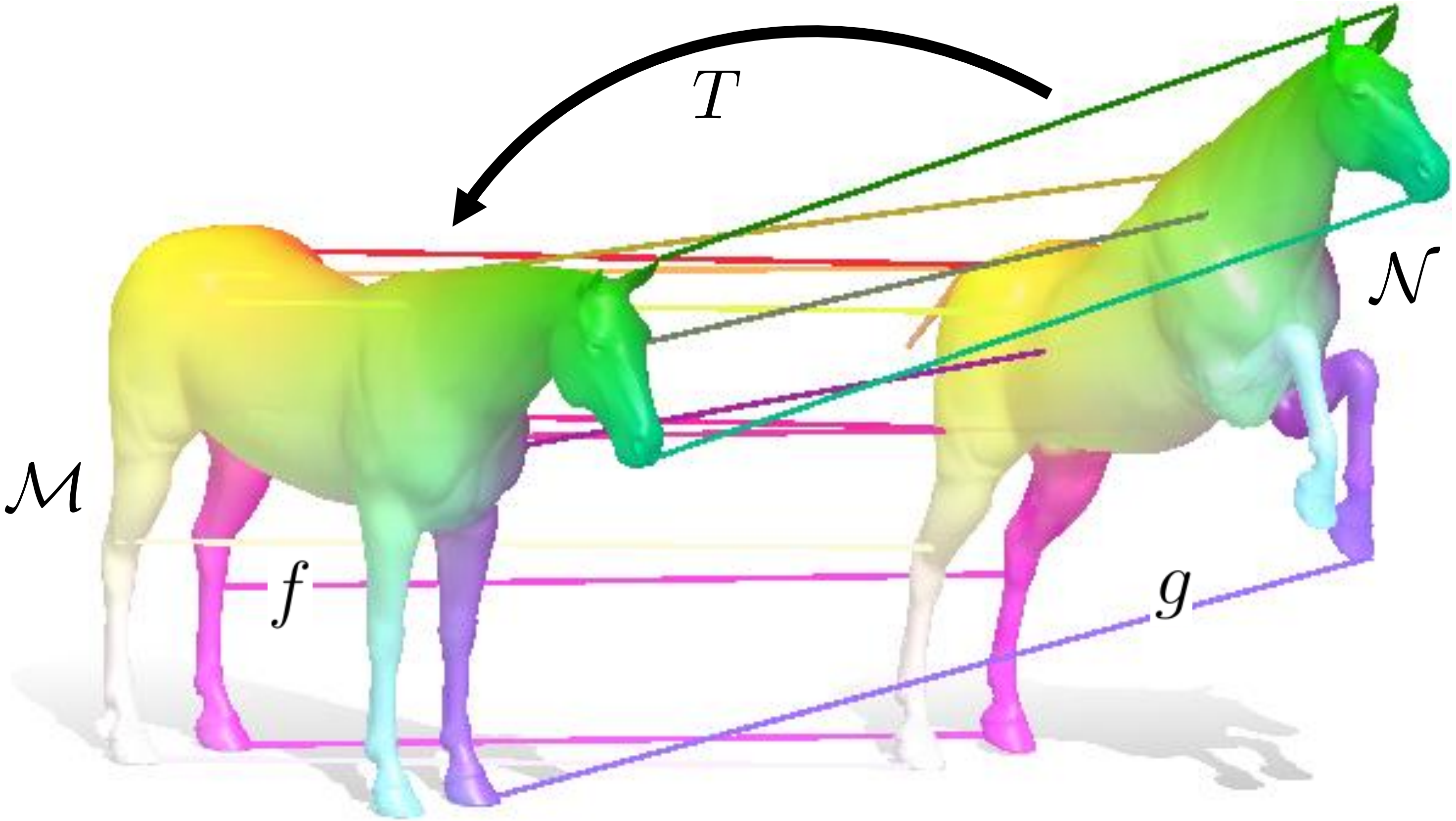


T is a point-to-point map

$$g(p) = f(T(p)) \ \forall p \in \mathcal{N}$$

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov, Ben-Chen, Solomon, Butscher, Guibas,* SIGGRAPH 2012

A point-to-point map $T : \mathcal{N} \to \mathcal{M}$

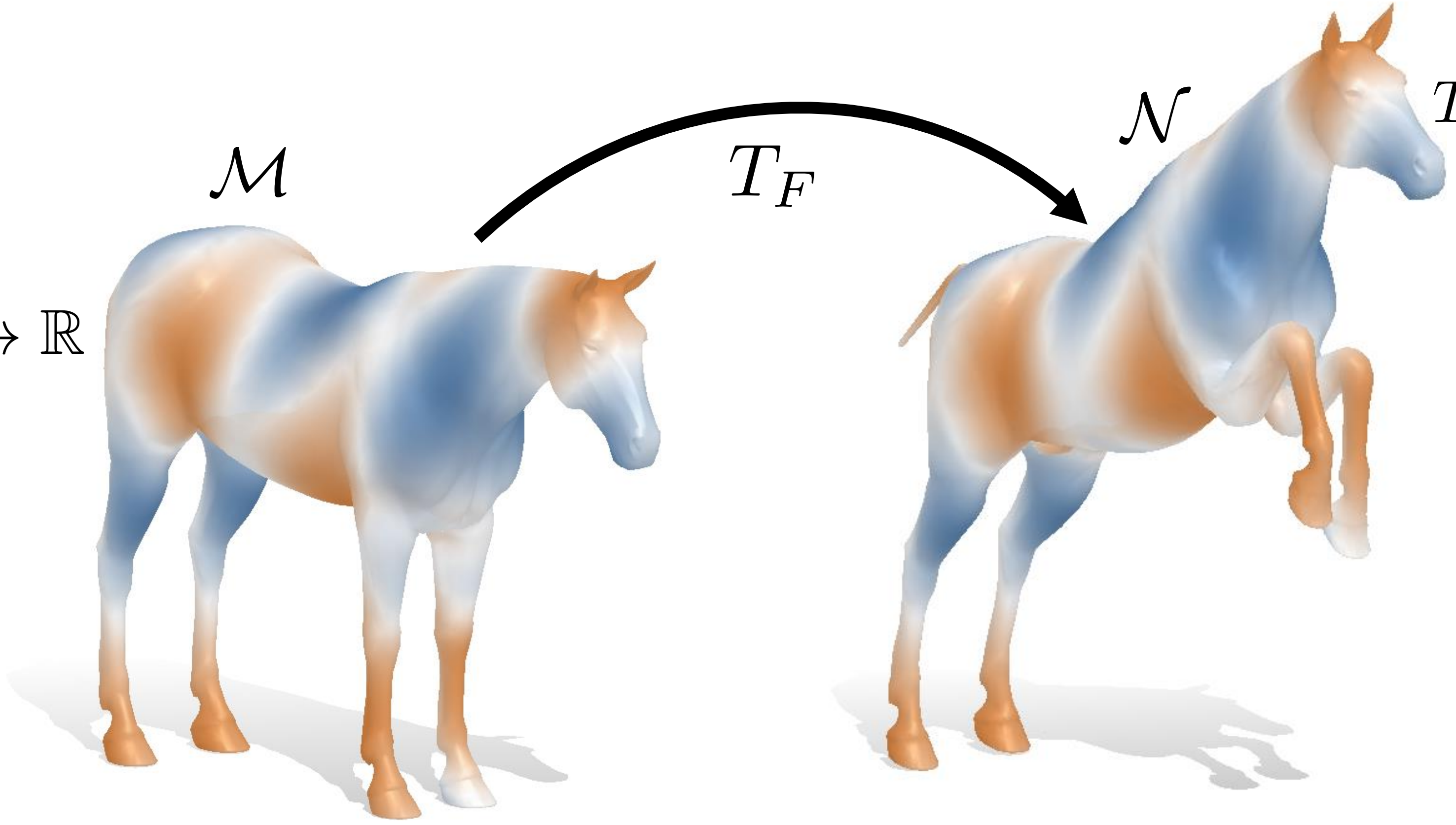


write it as a binary matrix $\Pi_{\mathcal{N}\mathcal{M}}(i,j) = 1 \Longleftrightarrow T(i) = j$

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov, Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Induces a functional map $T_F : F(\mathcal{M}) \longrightarrow F(\mathcal{N})$

$f : \mathcal{M} \to \mathbb{R}$



$T_F(f) = g : \mathcal{N} \to \mathbb{R}$

$g(y) = f(T(y))$

$$\begin{tikzcd} \mathcal{N} \arrow[r, "T"] \arrow[dr, "g"'] & \mathcal{M} \arrow[d, "f"] \\ & \mathbb{R} \end{tikzcd}$$

The transfer is defined as: $g = \Pi_{\mathcal{N}\mathcal{M}} f$

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov, Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

A functional map $T_F(f)$ is linear

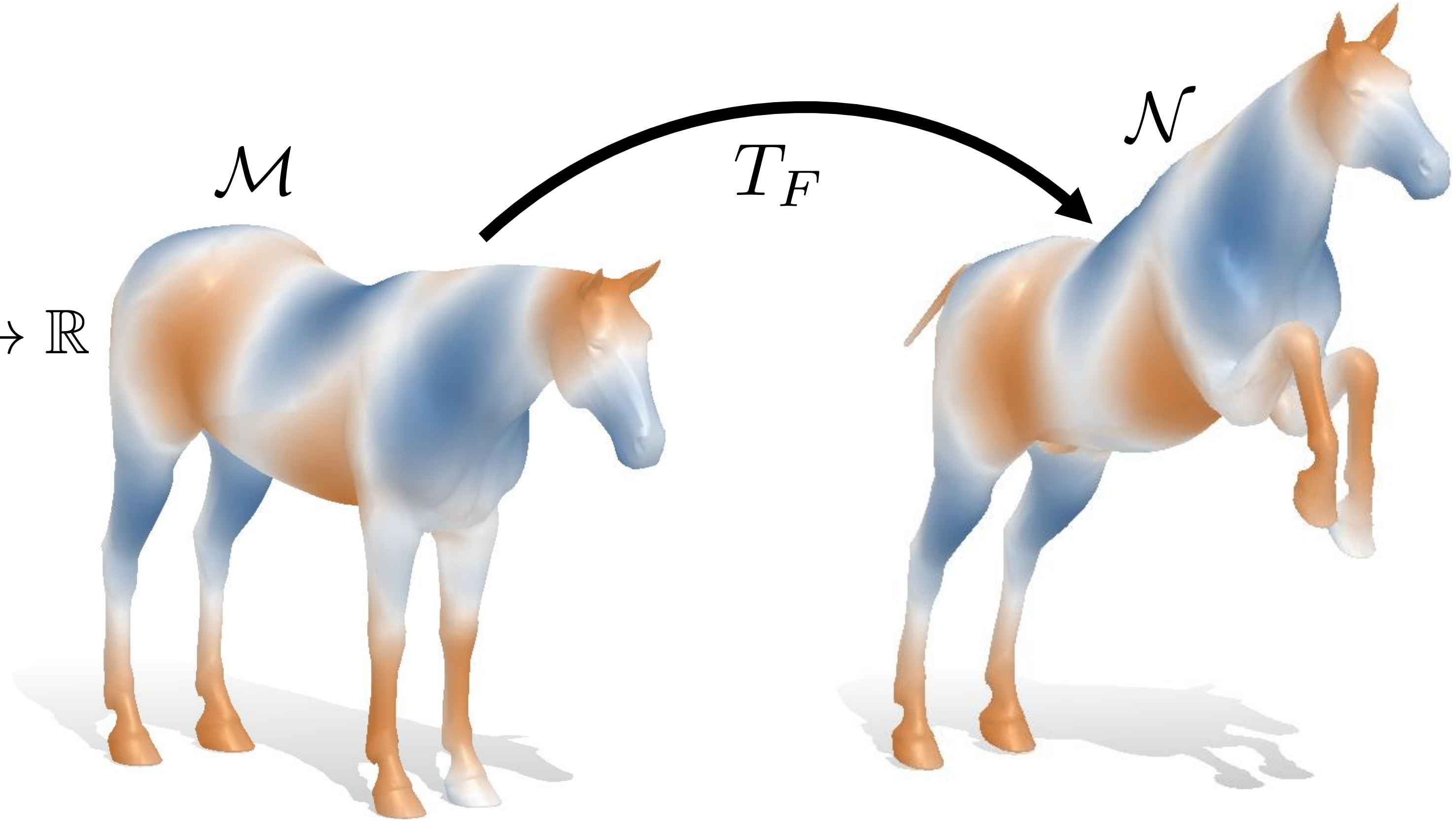
$$T_F : F(\mathcal{M}) \longrightarrow F(\mathcal{N})$$

$$T_F(\alpha_1 f_1 + \alpha_1 f_1) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$$

$$T_F(f) = g : \mathcal{N} \to \mathbb{R}$$

$$g(y) = f(T(y))$$

$$f : \mathcal{M} \to \mathbb{R}$$

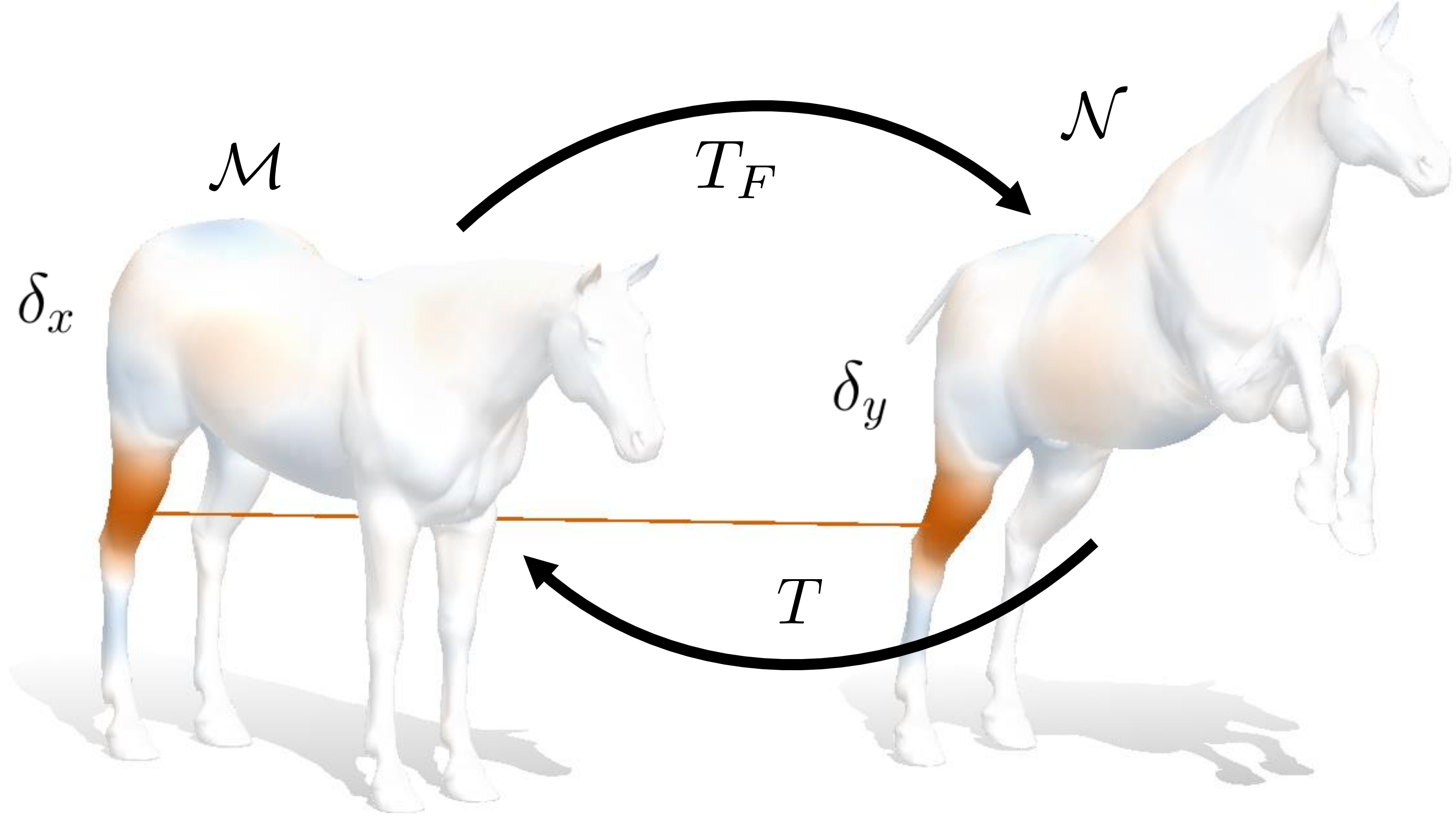


$$\begin{tikzcd} \mathcal{N} \arrow[r, "T"] \arrow[dr, "g"'] & \mathcal{M} \arrow[d, "f"] \\ & \mathbb{R} \end{tikzcd}$$

A functional map $T_F(f)$ is complete

$$T_F : F(\mathcal{M}) \longrightarrow F(\mathcal{N})$$

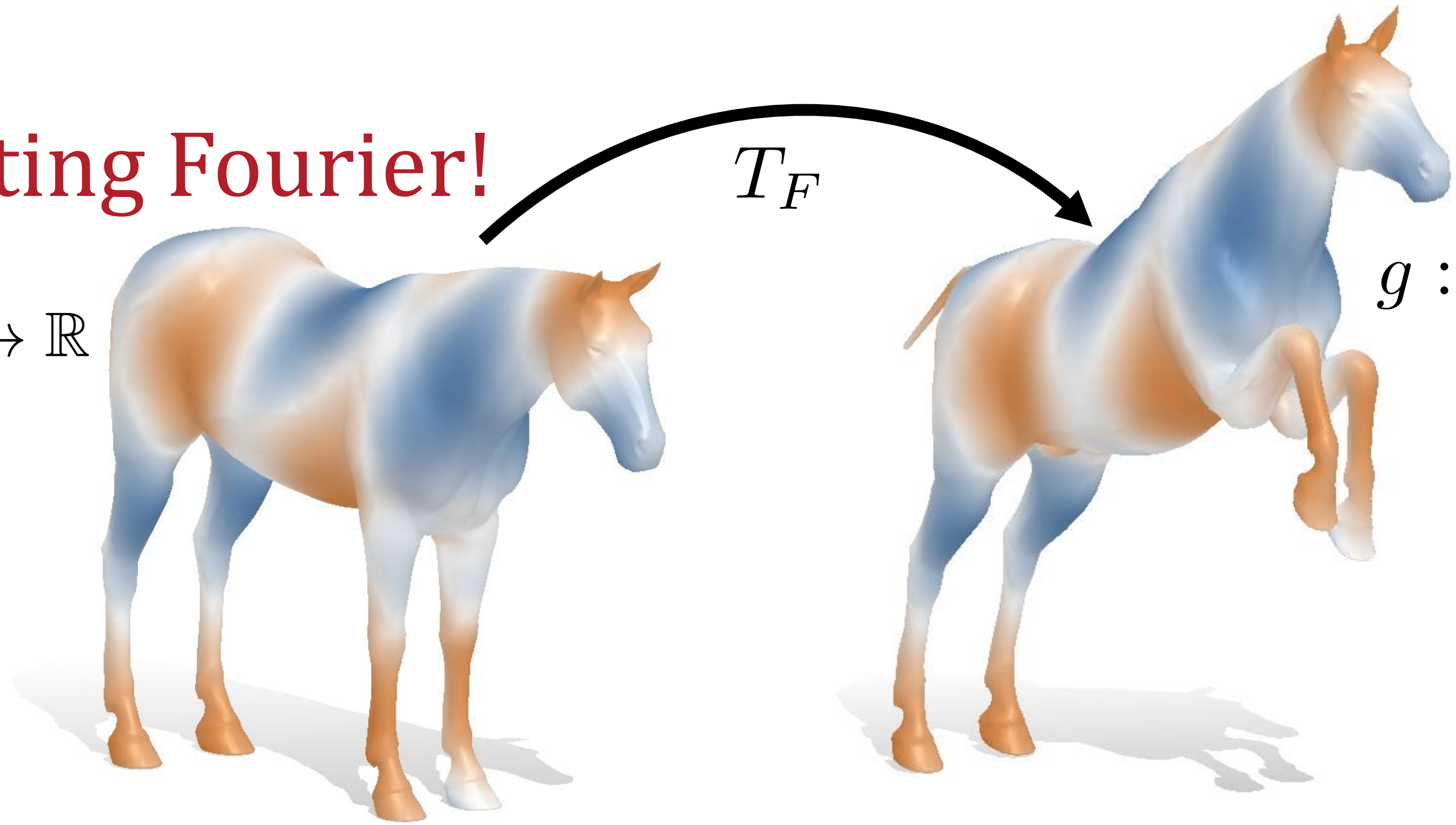
It is possible to recover *T* from indicator functions $T_F(\delta_x) = \delta_y$



$$\delta_y = \delta_x \circ T(y)$$

Observation

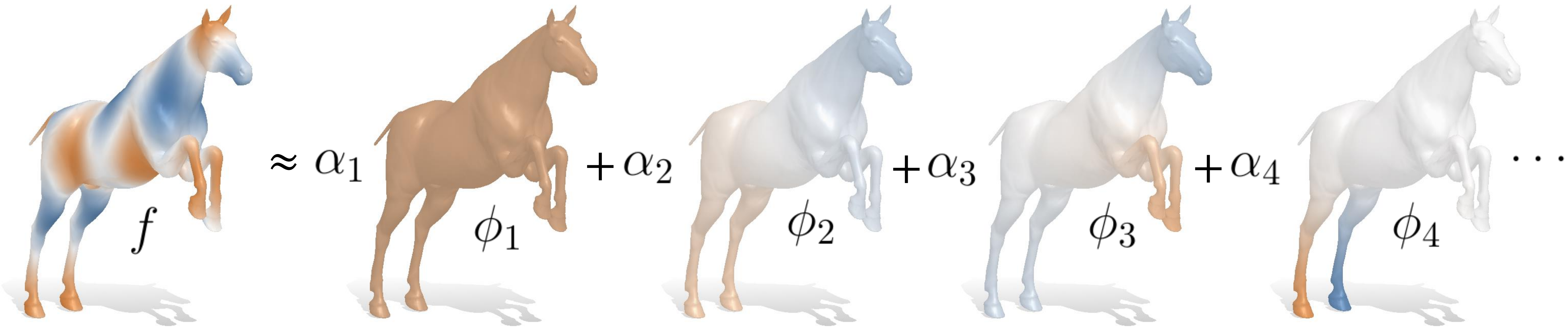
Exploiting Fourier!



Express both f and $T_F(f)$ in terms of *basis functions:*

$$f = \sum_i a_i \phi_i^{\mathcal{M}} \qquad g = T_F(f) = \sum_i b_i \phi_i^{\mathcal{N}}$$

Fourier representation on 3D shapes



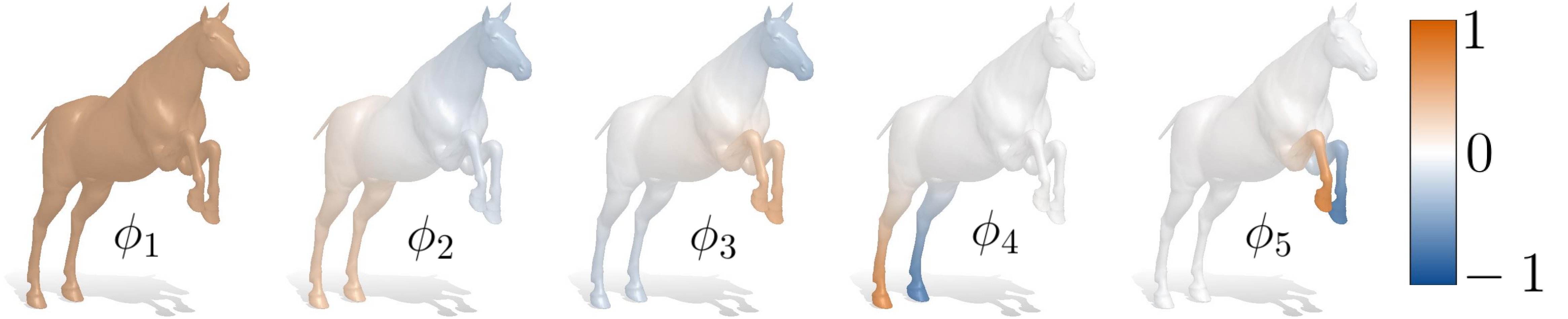
Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, SMI, 2006

Fourier Basis on 3D shapes

The eigenfunctions of the Laplace Beltrami Operator (LBO)

$$\boldsymbol{\Delta}_{\mathcal{M}}\phi_l = \lambda_l \phi_l \qquad \langle \phi_l, \phi_k \rangle_{\mathcal{M}} = \delta_l^k \qquad \lambda_l = \int_{\mathcal{M}} \|\nabla \phi_l\|^2 d\mu(x)$$

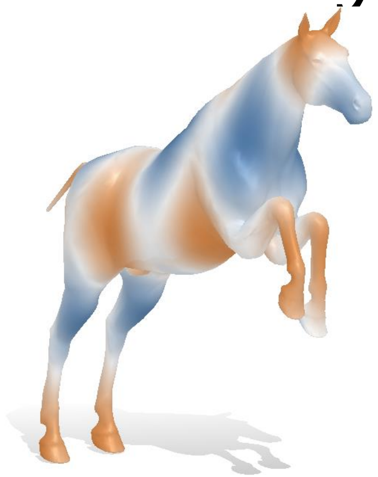
$\lambda_1 = 0$ $\quad \lambda_2 = 6.23$ $\quad \lambda_3 = 11.36$ $\quad \lambda_4 = 12.85$ $\quad \lambda_5 = 16.46$



Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, SMI, 2006

Synthesis and analysis

Given a signal: f



The analysis: $$\alpha_l = \langle f, \phi_l \rangle_{\mathcal{M}} = \int_{\mathcal{M}} f(x)\phi_l(x)d\mu(x)$$

The synthesis: $$f = \sum_{l=1}^{n} \alpha_l \phi_l = \sum_{l=1}^{n} \langle f, \phi_l \rangle_{\mathcal{M}} \phi_l \approx \sum_{l=1}^{k<n} \alpha_l \phi_l$$

Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, SMI, 2006

Notation

$$\langle f, g \rangle_{\mathcal{M}} = f^\top \mathbf{\Omega}_{\mathcal{M}} g$$

$$\mathbf{\Phi}_{\mathcal{M}} = [\phi_1,\ \phi_2,\ \ldots\ \phi_{k-1},\ \phi_k]$$

$$\mathbf{\Phi}^\dagger_{\mathcal{M}}\ s.t.\ \mathbf{\Phi}^\dagger_{\mathcal{M}} \mathbf{\Phi}_{\mathcal{M}} = I$$

$$\mathbf{\Phi}_{\mathcal{M}}\ s.t.\ \mathbf{\Phi}^\top_{\mathcal{M}} \mathbf{\Omega}_{\mathcal{M}} \mathbf{\Phi}_{\mathcal{M}} = I$$

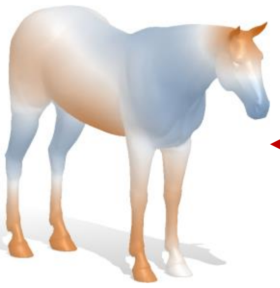
$$\mathbf{\Phi}^\dagger_{\mathcal{M}} = \mathbf{\Phi}^\top_{\mathcal{M}} \mathbf{\Omega}_{\mathcal{M}}$$

Given a signal: f

The analysis: $\boldsymbol{\alpha} = \mathbf{\Phi}^\dagger_{\mathcal{M}} f$

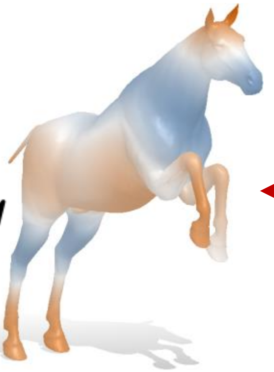
The synthesis: $f = \mathbf{\Phi}_{\mathcal{M}} \boldsymbol{\alpha} = \mathbf{\Phi}_{\mathcal{M}} \mathbf{\Phi}^\dagger_{\mathcal{M}} f$

Question



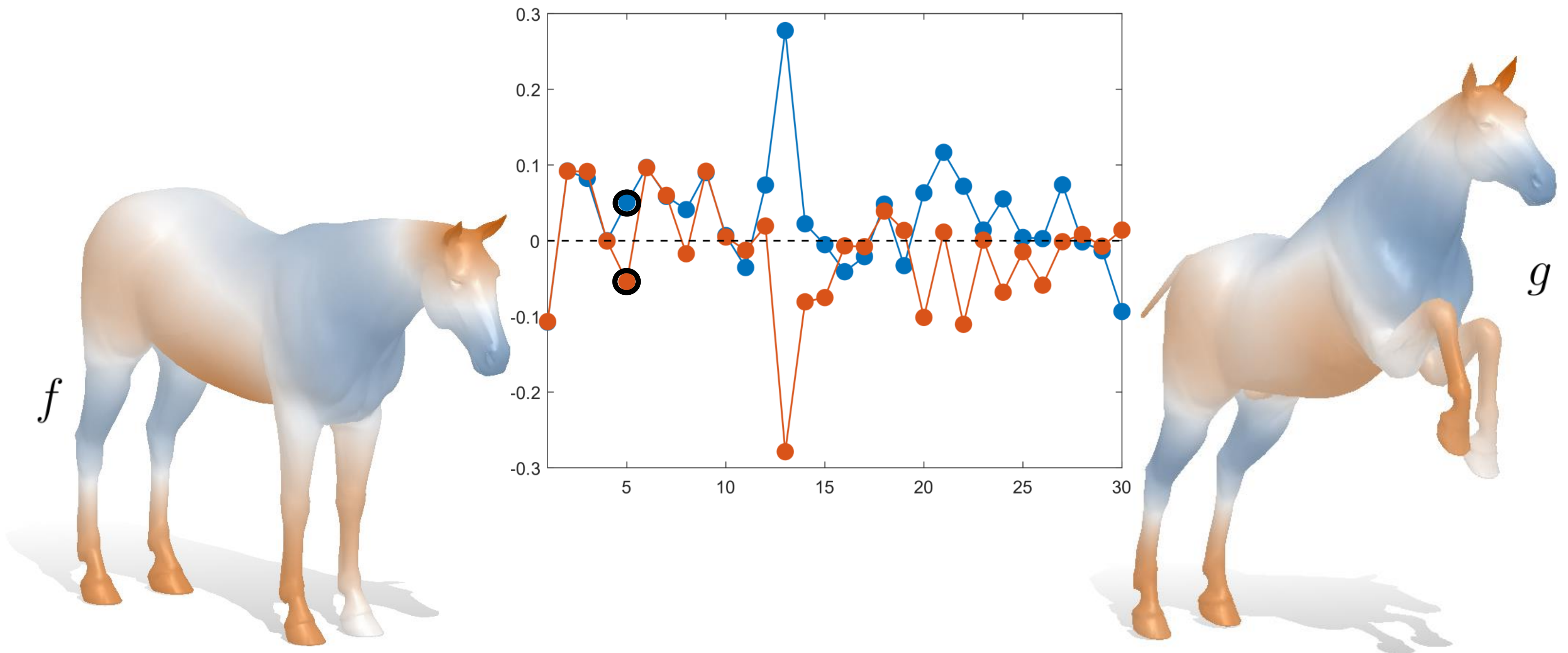
$$f \xleftrightarrow{\text{Fourier}} [\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$$

What is the relation between the two set of coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$?

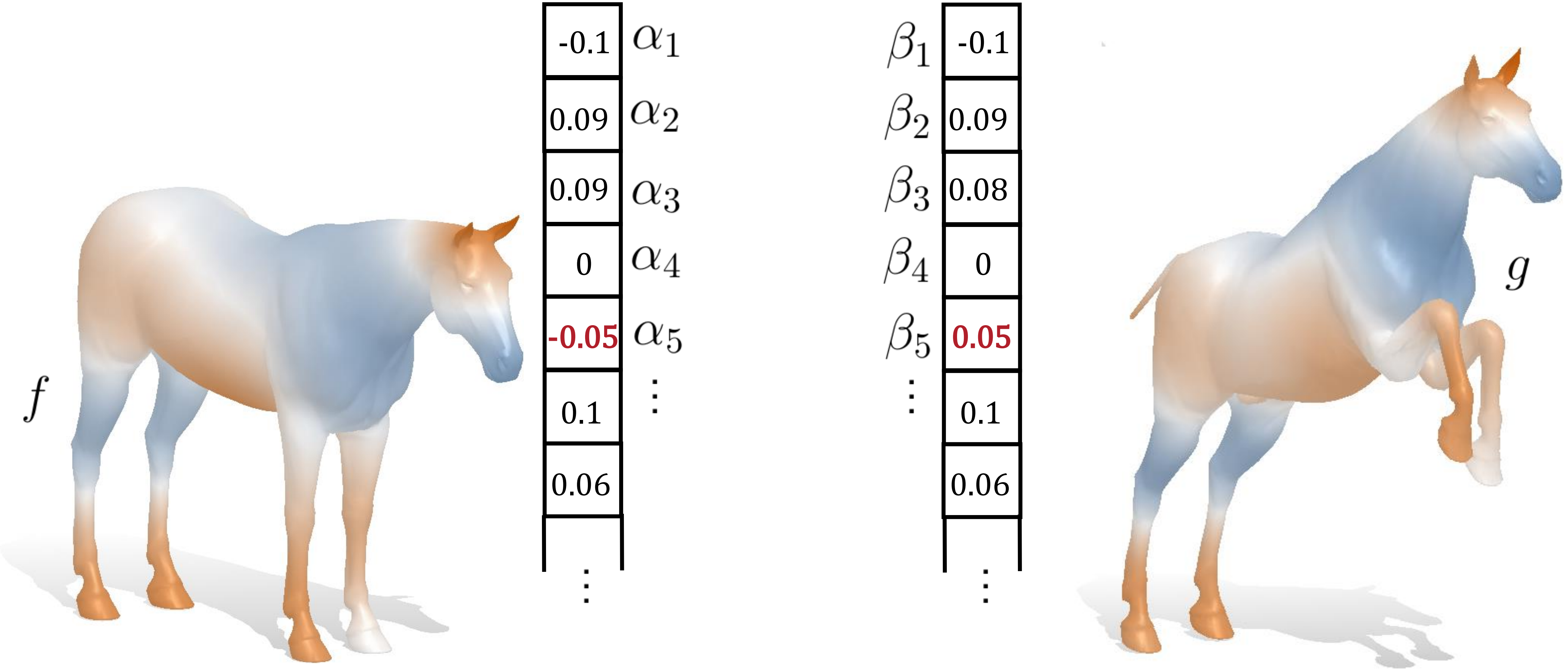


$$g \xleftrightarrow{\text{Fourier}} [\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$$

Functions on 2 different domains



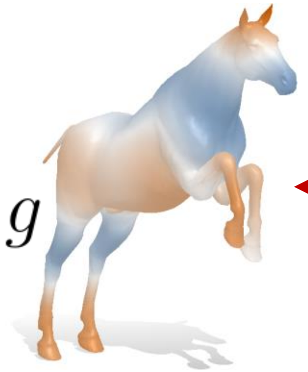
Functions on 2 different domains



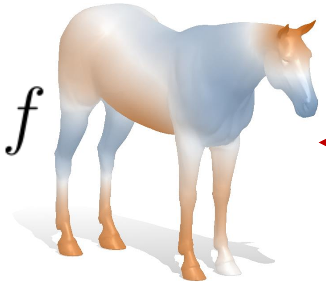
Observation

f  $\xleftrightarrow[\text{Analysis}]{\text{Fourier}}$ $[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$

Since T_F is linear, there is a linear transformation from $\{a_i\}$ to $\{b_j\}$

g  $\xleftrightarrow[\text{Analysis}]{\text{Fourier}}$ $[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$

Observation

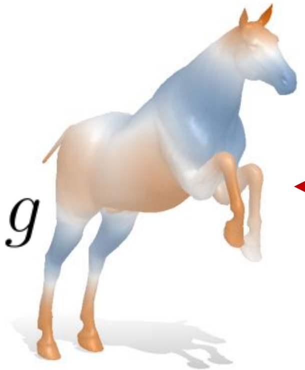
f 

Fourier Analysis

$$[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$$

$$T_F(f) = T_F\left(\sum_i a_i \phi_i^M\right) = \sum_i a_i T_F\left(\phi_i^M\right) \qquad T_F\left(\phi_i^M\right) = \sum_j c_{ij}\phi_j^N$$

$$T_F(f) = \sum_i a_i \sum_j c_{ij}\phi_j^N = \sum_j \sum_i a_i c_{ij}\phi_j^N$$

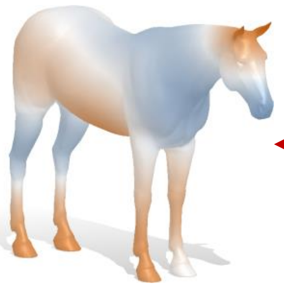
g 

Fourier Analysis

$$[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$$

Observation

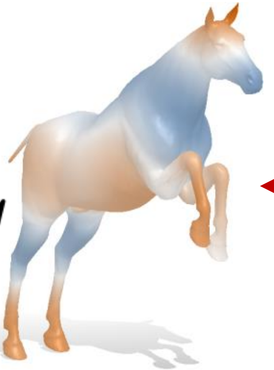
f



$\xleftrightarrow[\text{Analysis}]{\text{Fourier}}$ $[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$

$$T_F \approx \Phi_{\mathcal{N}} \mathbf{C} \Phi_{\mathcal{M}}^{\dagger} \quad \Longleftrightarrow \quad \begin{aligned} g &= T_F(f) \\ &= \Phi_{\mathcal{N}} \mathbf{C} \Phi_{\mathcal{M}}^{\dagger} f \\ &= \Phi_{\mathcal{N}} \mathbf{C} \mathbf{a} \\ &= \Phi_{\mathcal{N}} \mathbf{b} \end{aligned}$$

g



$\xleftrightarrow[\text{Analysis}]{\text{Fourier}}$ $[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$

Functional map: matrix C that translates coefficients from $\Phi_{\mathcal{M}}$ to $\Phi_{\mathcal{N}}$

Observation

f

Fourier Analysis $\longleftrightarrow$

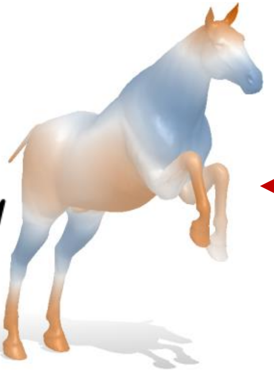
$$[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$$

Functional Map

Estimate $\boldsymbol{C}$ such that

$$\boldsymbol{C}\boldsymbol{a} = \boldsymbol{b}$$

g



Fourier Analysis $\longleftrightarrow$

$$[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$$

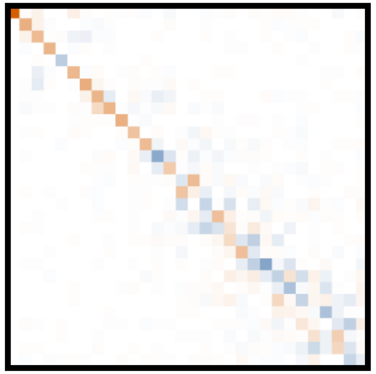
Functional map: matrix C that translates coefficients from $\Phi_{\mathcal{M}}$ to $\Phi_{\mathcal{N}}$

Observation

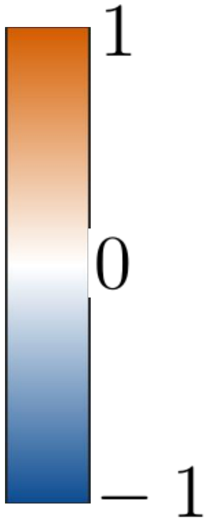
f

Fourier Analysis

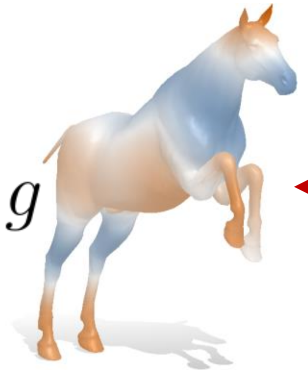
$$[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$$



$$\boldsymbol{C}\boldsymbol{a} = \boldsymbol{b}$$



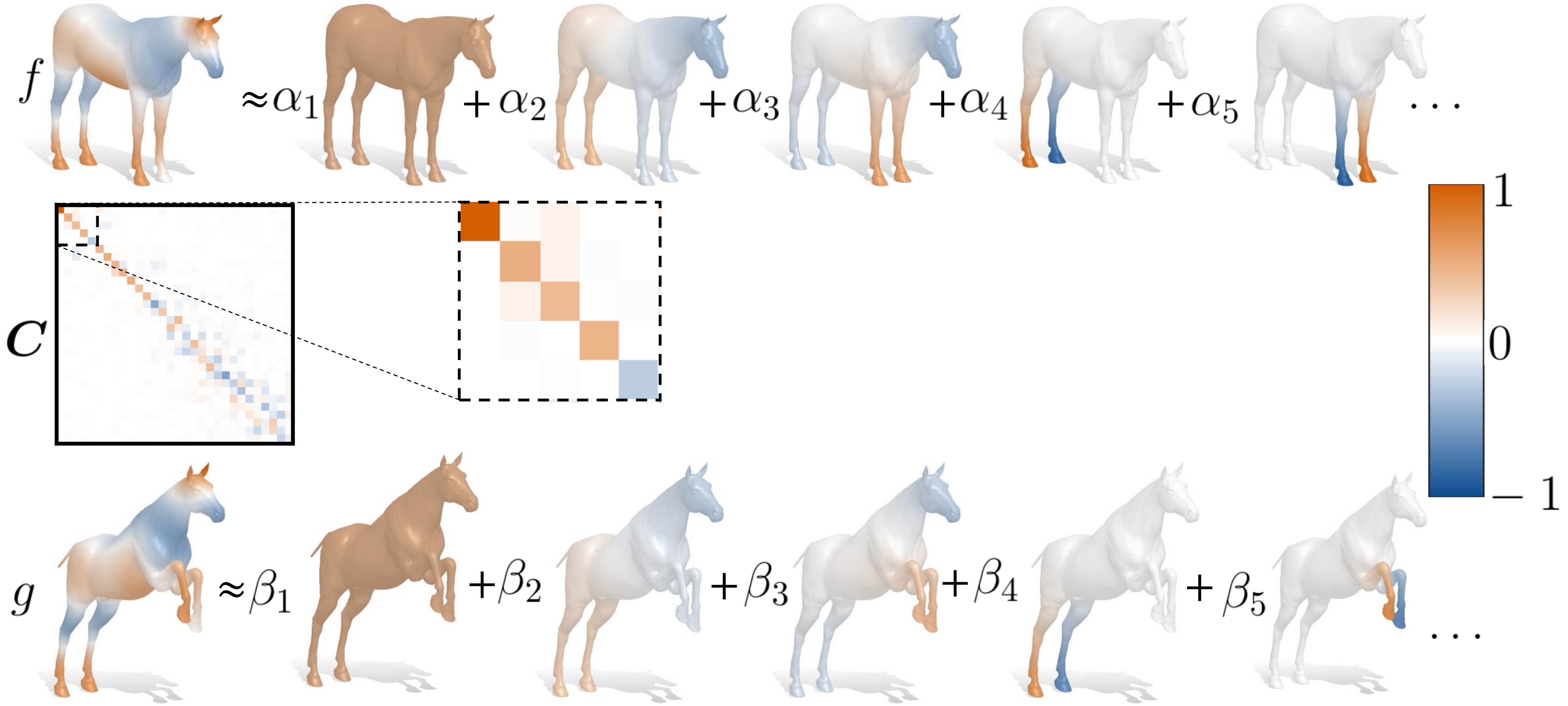
g



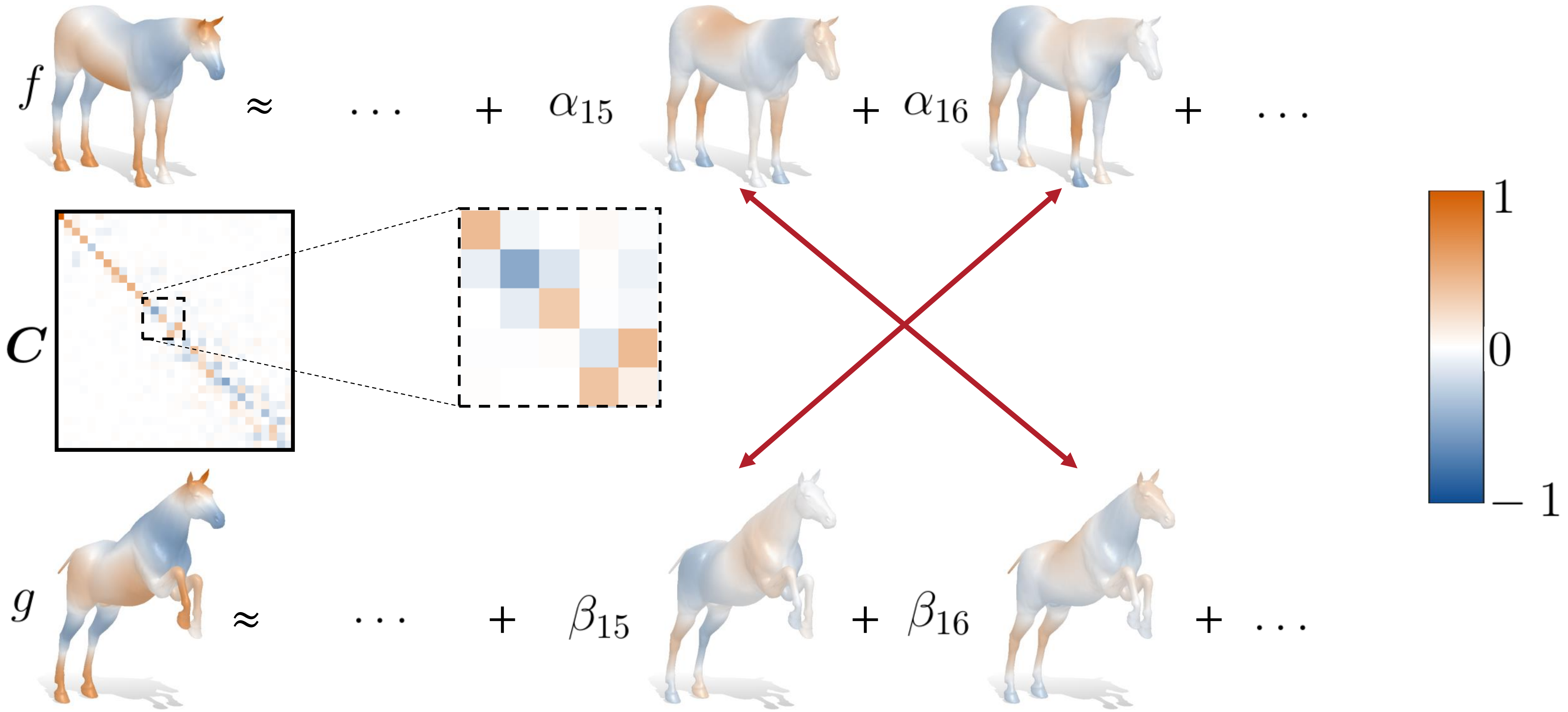
Fourier Analysis

$$[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$$

Observation



Observation



Functional map

Definition:

For a fixed choice of basis functions $\{\phi^{\mathcal{M}}\}, \{\phi^{\mathcal{N}}\}$, and a linear transformation T_F between functions, a functional map is a matrix C, s.t. for any $f = \sum_i a_i \phi_i^{\mathcal{M}}$ if $T_F(f) = \sum_i b_i \phi_i^{\mathcal{N}}$, then:

$$\mathbf{b} = C\mathbf{a}$$

C_{ij} : coefficient of $T_F(\phi_j^{\mathcal{M}})$ in the basis of $\phi_i^{\mathcal{N}}$

In an orthonormal basis: $C_{i,j} = \langle T_F(\phi_j^{\mathcal{M}}), \phi_i^{\mathcal{N}} \rangle$

Functional map

Definition:

For a fixed choice of basis functions $\{\phi^{\mathcal{M}}\}, \{\phi^{\mathcal{N}}\}$, and a linear transformation T_F between functions, a functional map is a matrix C, s.t. for any $f = \sum_i a_i \phi_i^{\mathcal{M}}$ if $T_F(f) = \sum_i b_i \phi_i^{\mathcal{N}}$, then:

$$\mathbf{b} = C\mathbf{a}$$

How to estimate C ?

If $\Pi_{\mathcal{NM}}$ is known then $\boldsymbol{C} = \boldsymbol{\Phi}_{\mathcal{N}}^{\dagger} \boldsymbol{\Pi}_{\mathcal{NM}} \boldsymbol{\Phi}_{\mathcal{M}}$ $\Longleftrightarrow$

$$g = \Pi_{\mathcal{NM}} f$$

$$\Phi_{\mathcal{N}} \mathbf{b} = \Pi_{\mathcal{NM}} \Phi_{\mathcal{M}} \mathbf{a}$$

$$\mathbf{b} = \Phi_{\mathcal{N}}^{\dagger} \Pi_{\mathcal{NM}} \Phi_{\mathcal{M}} \mathbf{a}$$

Functional map

Definition:

For a fixed choice of basis functions $\{\phi^{\mathcal{M}}\}, \{\phi^{\mathcal{N}}\}$, and a linear transformation T_F between functions, a functional map is a matrix C, s.t. for any $f = \sum_i a_i \phi_i^{\mathcal{M}}$ if $T_F(f) = \sum_i b_i \phi_i^{\mathcal{N}}$, then:

$$\mathbf{b} = C\mathbf{a}$$

How to estimate C ?

Unfortunately $\Pi_{\mathcal{N}\mathcal{M}}$ is not known..., other strategies can be employed

Matching via Function Preservation

Suppose we don't know *C*. However, we expect a pair of functions $f : \mathcal{M} \to \mathbb{R}$ and $g : \mathcal{N} \to \mathbb{R}$ to correspond. Then, *C* must be s.t.

$$C\mathbf{a} \approx \mathbf{b}$$

where $f = \sum_i a_i \phi_i^{\mathcal{M}}, \quad g = \sum_i b_i \phi_i^{\mathcal{N}}.$

Given enough $\{\mathbf{a}, \mathbf{b}\}$ pairs, we can recover ***C*** through *a linear least squares system*.

Map contraints

Suppose we don't know *C*. However, we expect a pair of functions $f : \mathcal{M} \to \mathbb{R}$ and $g : \mathcal{N} \to \mathbb{R}$ to correspond. Then, *C* must be s.t.

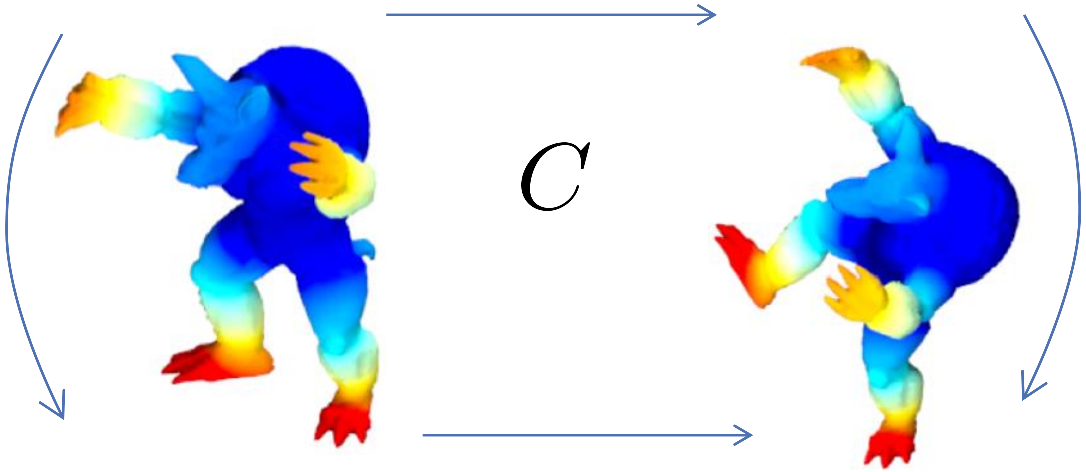
$$C\mathbf{a} \approx \mathbf{b}$$

Function preservation constraint is general and includes:

- Attribute (e.g., color) preservation.
- Descriptor preservation (e.g. Gauss curvature).
- Landmark correspondences (e.g. distance to the point).
- Part correspondences (e.g. indicator function).

Commutativity Constraints

Commutativity with other operators:



$S_{\mathcal{M}}$ $\quad C \quad$ $S_{\mathcal{N}}$

$$CS_{\mathcal{M}} = S_{\mathcal{N}}C$$

Note that the energy: $\|CS_{\mathcal{M}} - S_{\mathcal{N}}C\|_F^2$ is *quadratic* in *C*.

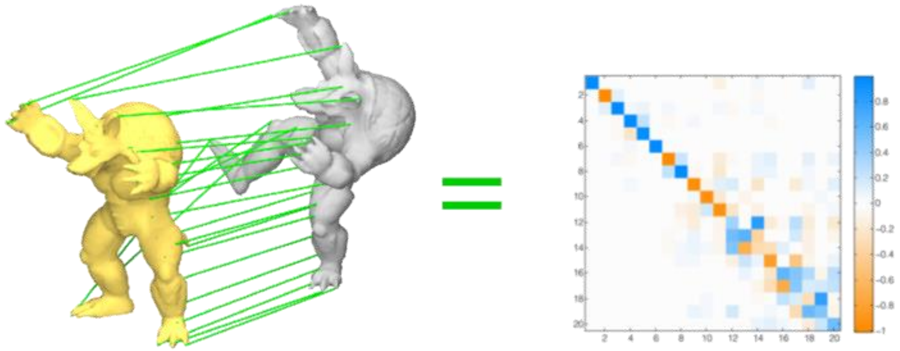
Regularization

Lemma 1:

The mapping is *isometric*, if and only if the functional map matrix commutes with the Laplacian:

$$C\Delta_{\mathcal{M}} = \Delta_{\mathcal{N}}C$$

Implies that isometries result in **diagonal** functional maps.



Regularization

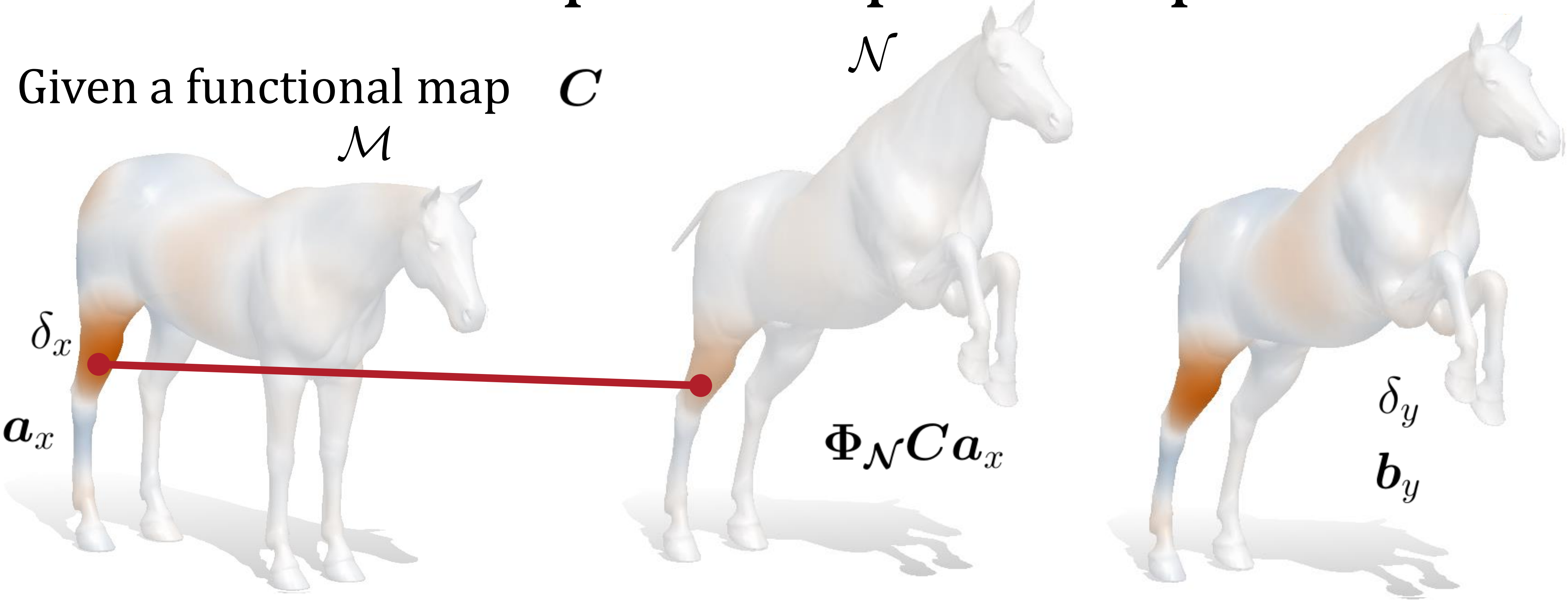
Lemma 2:

The mapping is *locally volume preserving*, if and only if the functional map matrix is *orthonormal.*

$$\mathbf{C}^\top \mathbf{C} = I$$

Conversion to a point-to-point map

Given a functional map $\boldsymbol{C}$



$$T(x) = \underset{y}{argmin} \|\boldsymbol{b}_y - \boldsymbol{C}\boldsymbol{a}_x\|_2$$

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov, Ben-Chen, Solomon, Butscher, Guibas,* SIGGRAPH 2012

Fmaps Pipeline

1. Compute the first k (~30-100) eigenfunctions of the LBO.
   Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

2. Compute probe functions (e.g., landmarks or descriptors) on $\mathcal{M}, \mathcal{N}$
   Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of $\boldsymbol{A}$ and $\boldsymbol{B}$

3. Solve $\underset{\mathbf{C}}{argmin} ||\mathbf{CA} - \mathbf{B}||_F^2 + \mathcal{R}(\mathbf{C})$

4. Convert the functional map to a point to point map T.

Functional map refinement

Option: use post-processing to improve the functional map estimation.

Iterate:

1. Compute the point-to-point map *T*.

$\mathbf{a}_x$ and $\mathbf{b}_{T(\mathbf{x})}$ from indicator function

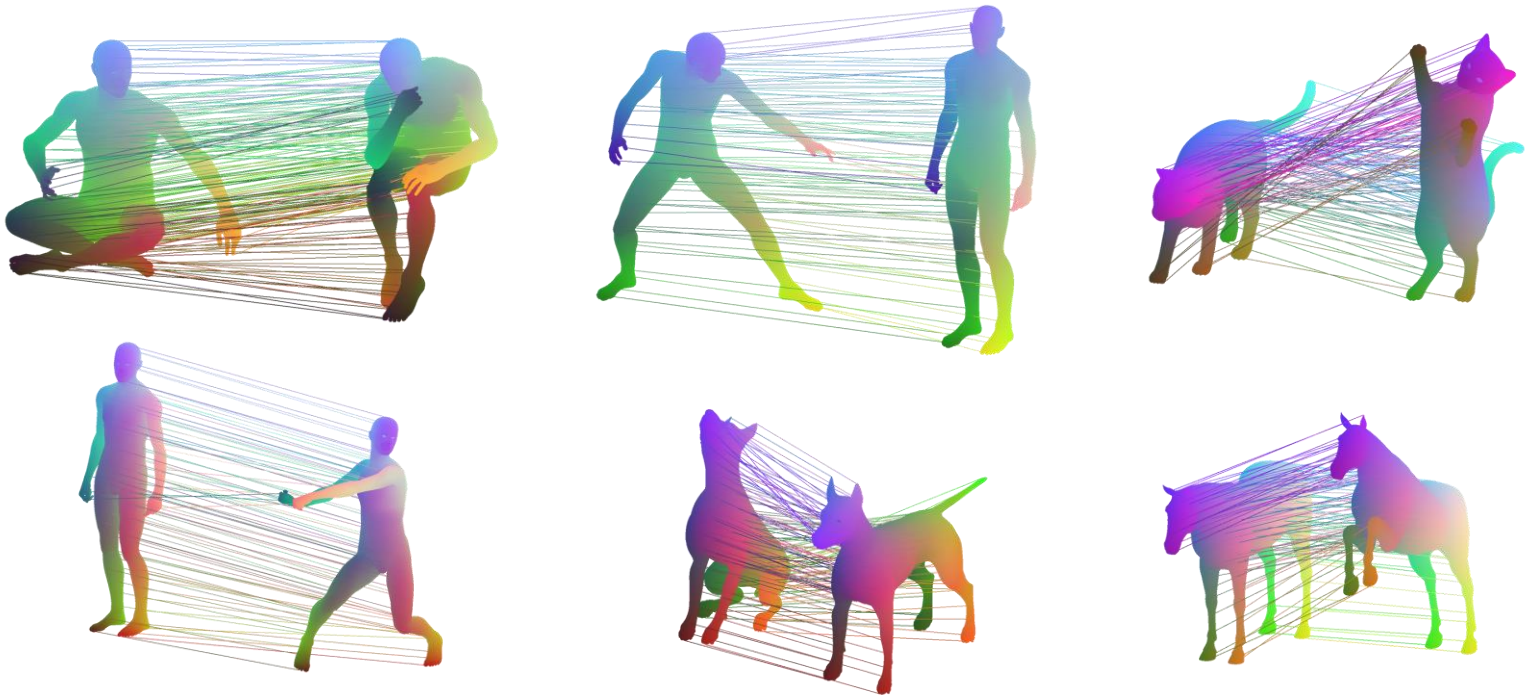
2. Solve for the functional map:

$$\arg\min_{C^\top C = I} \sum_{x \in \mathcal{M}} ||C\mathbf{a}_x - \mathbf{b}_{T(\mathbf{x})}||_2^2$$

Exactly the same objective as ICP, but in higher dimension.
Can use the same method!

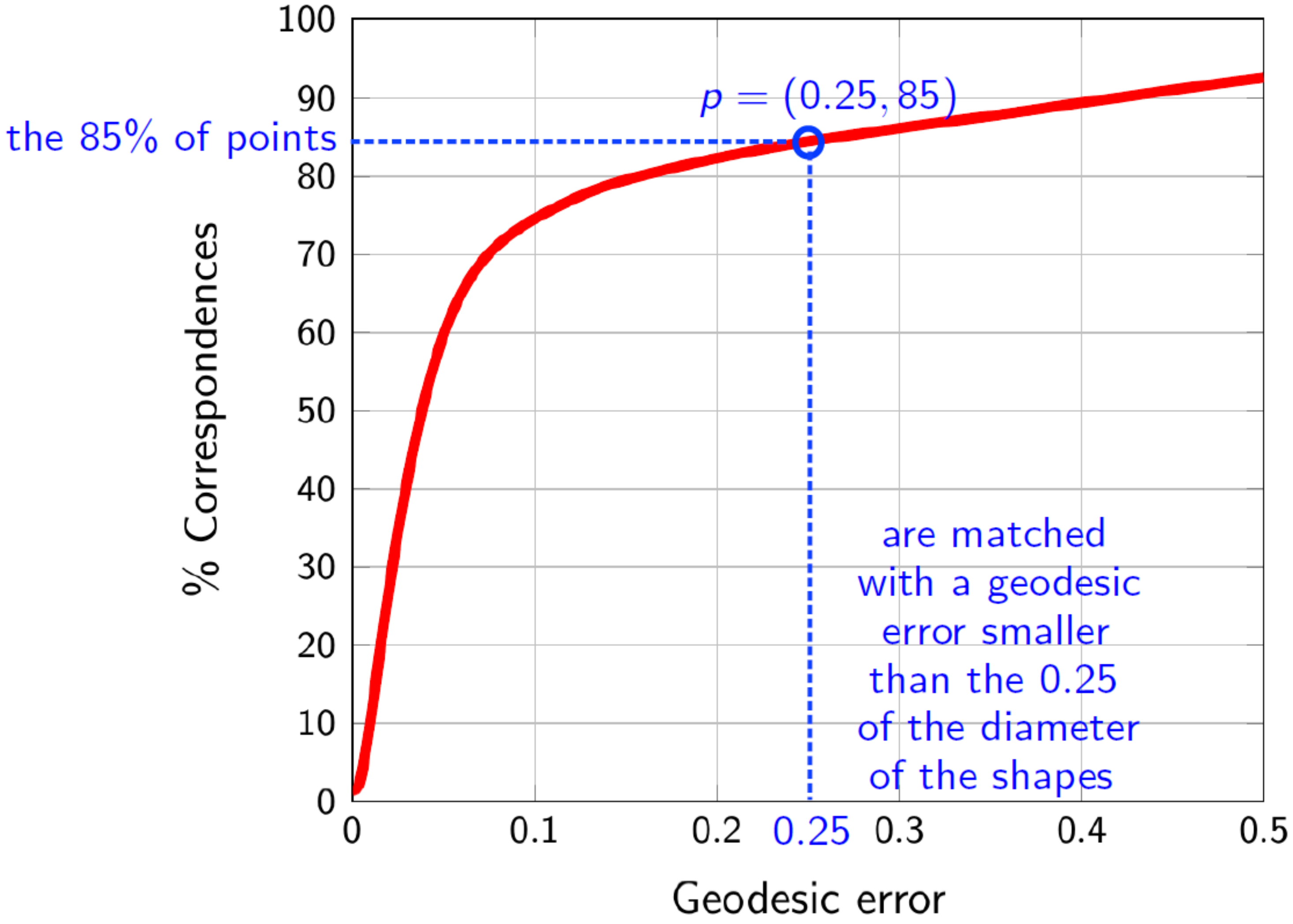
Slide credit M. *Ovsjanikov*

Fmaps qualitative results



Functional maps: a flexible representation of maps between shapes, *Ovsjanikov, Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

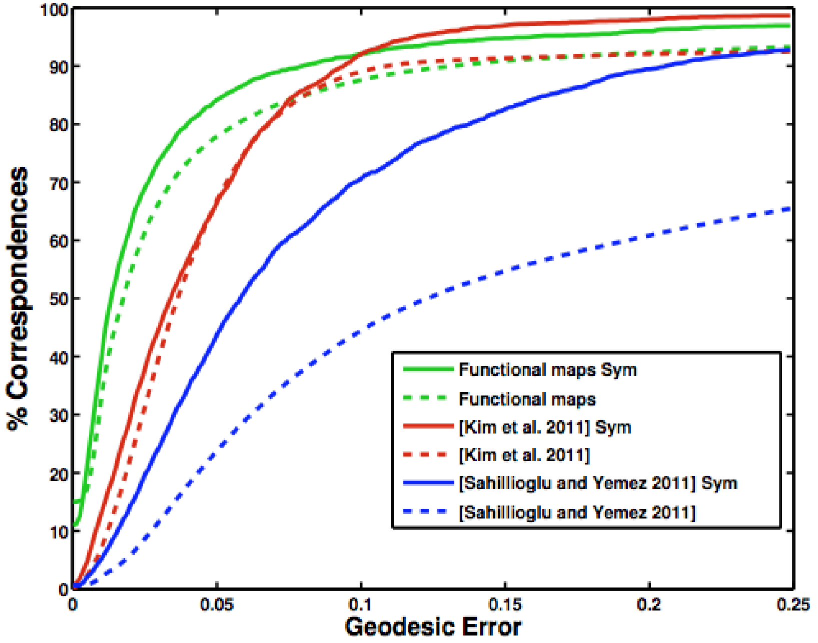
Standard evaluation



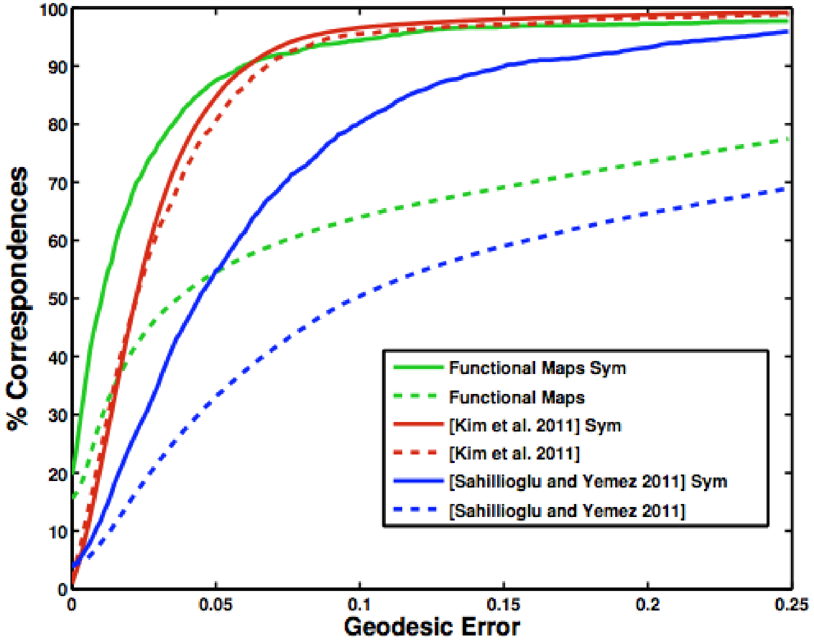
Blended intrinsic maps, *Kim, Lipman, Funkhouser*, TOG 2011

Slide credit M. *Ovsjanikov*

Fmaps quantitative results



SCAPE



TOSCA

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov, Ben-Chen, Solomon, Butscher, Guibas*, SIGGRAPH 2012

Conclusions

- Functional map has opened new perspective to deal with shape matching,
  - Transfer between shapes can be estimated without an explicit point to point mapping,
- Working on the functional space is more convenient (effective and efficient),
  - Nice constraints can be incorporated by simple algebraic constraints,
- The point-to-point matching can be estimated from the functional map using an ICP-like approach for non rigid shapes,

Open issues

- We estimate the mapping with a small k (e.g., k=100), this means that we are not dealing well with high frequency,
- The method to recover the point to point mapping from functional map can be improved,
- There are many details to clarify:
  - How to choose the best probe functions/descriptors?
  - Which regularization constraints w.r.t. the desired properties of the mapping?