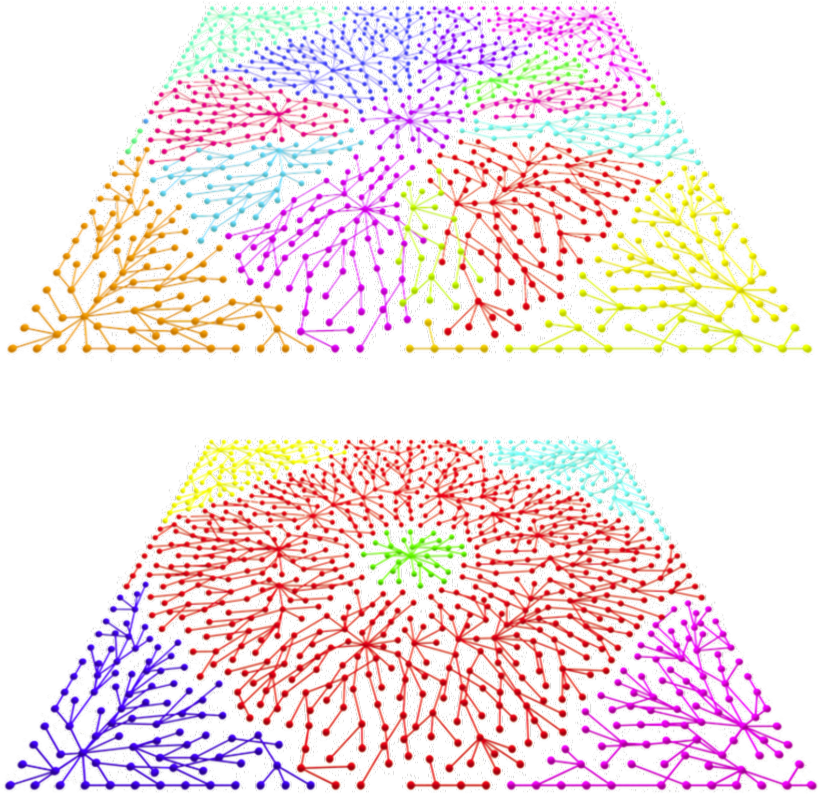
CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas
Computer Science Department
Stanford University

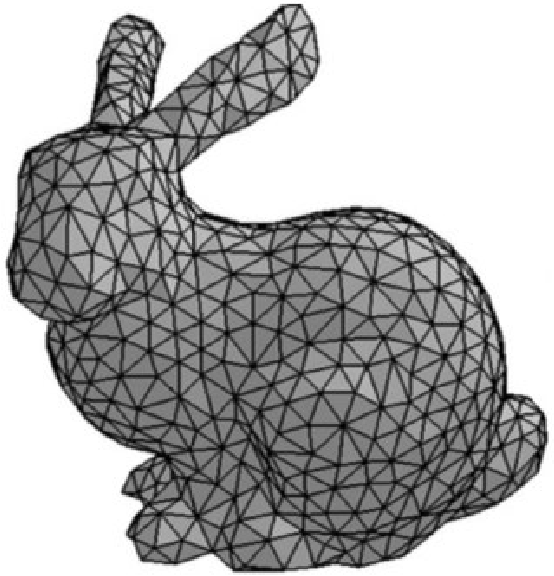


Lecture 18
10 June 2020

Leonidas Guibas Laboratory
Geometric Computing

Last Time: Deep Learning on Graphs and Meshes

CNNs on Irregular, Non-Euclidean Domains



3D shape graph

General graphs

Different Formulations of Non-Euclidean CNNs

Spectral domain

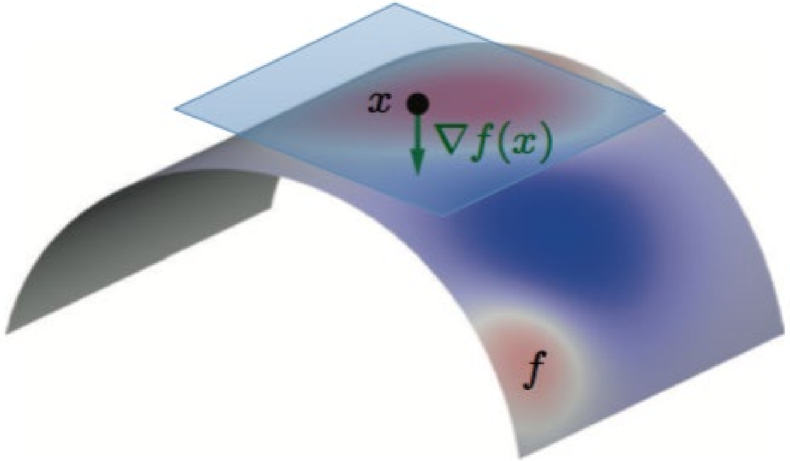
Spatial domain

Spectral Methods: Laplacian is Key

- Intrinsic gradient operator

$$\nabla f : L^2(\mathcal{X}) \to L^2(T\mathcal{X})$$

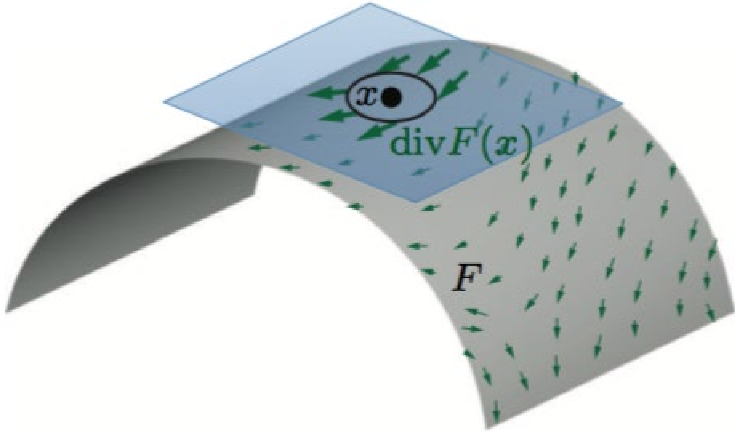
"direction of steepest change of f"



- Intrinsic divergence operator

$$\text{div} : L^2(T\mathcal{X}) \to L^2(\mathcal{X})$$

"net flow of field F at x"

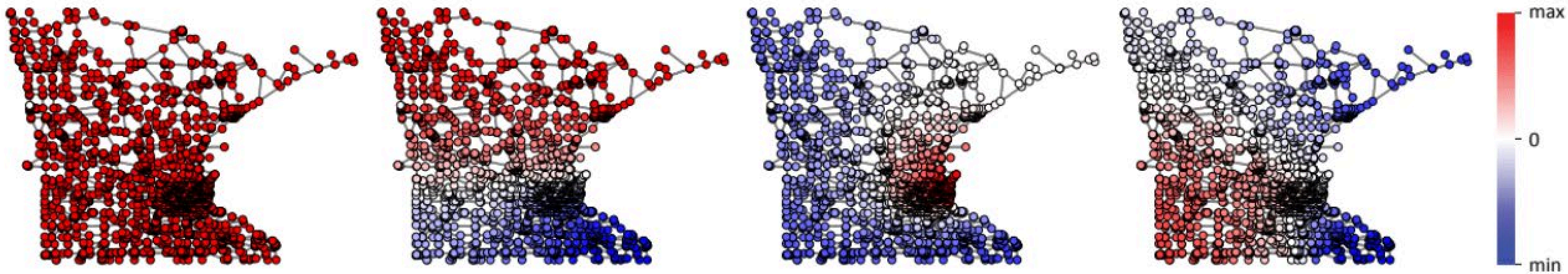


Going Non-Euclidean: Laplacian is Key

Eigendecomposition of a graph Laplacian

$$\boldsymbol{\Delta} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top$$

where $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_n)$ are orthogonal eigenvectors ($\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \mathbf{I}$) and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ the corresponding non-negative eigenvalues



ϕ_1 ϕ_2 ϕ_3 ϕ_4

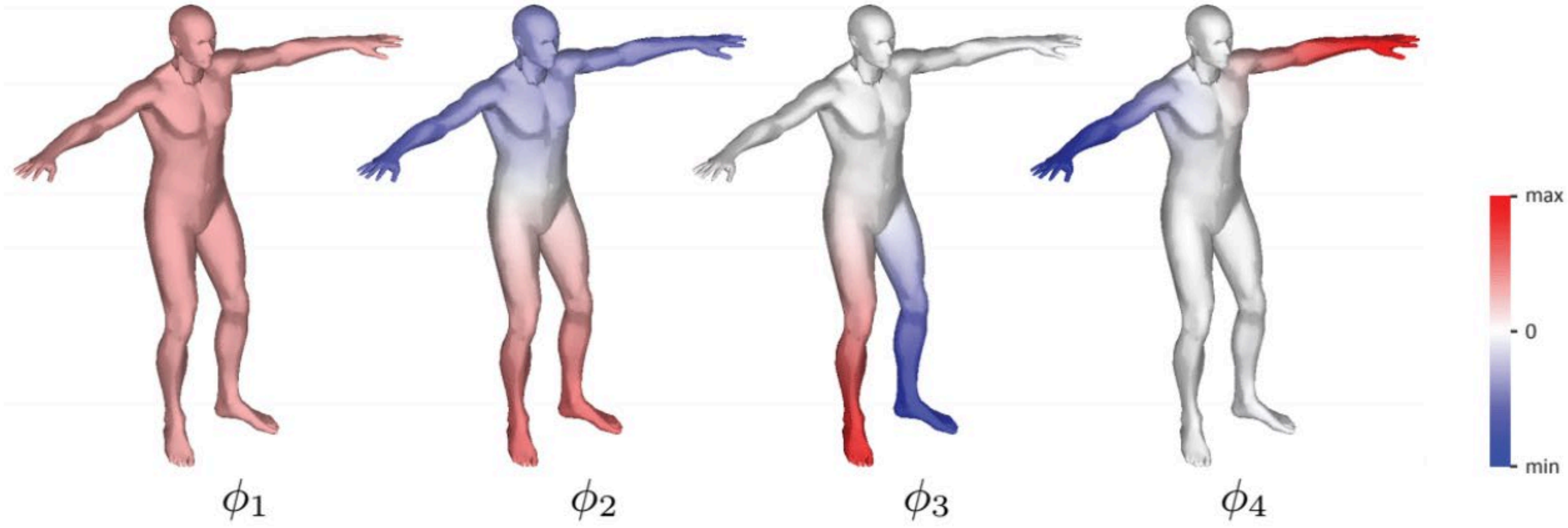
First eigenfunctions of a graph Laplacian

Going Non-Euclidean: Laplacian is Key

Eigendecomposition of a graph Laplacian

$$\boldsymbol{\Delta} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top$$

where $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_n)$ are orthogonal eigenvectors $(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \mathbf{I})$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ the corresponding non-negative eigenvalues



First eigenfunctions of a manifold Laplacian

Convolution: Euclidean space

Given two functions $f, g : [-\pi, \pi] \to \mathbb{R}$ their convolution is a function

$$(f \star g)(x) = \int_{-\pi}^{\pi} f(x')g(x - x')dx'$$

- Shift-invariance: $f(x - x_0) \star g(x) = (f \star g)(x - x_0)$
- Convolution theorem: Fourier transform diagonalizes the convolution operator $\Rightarrow$ convolution can be computed in the Fourier domain as

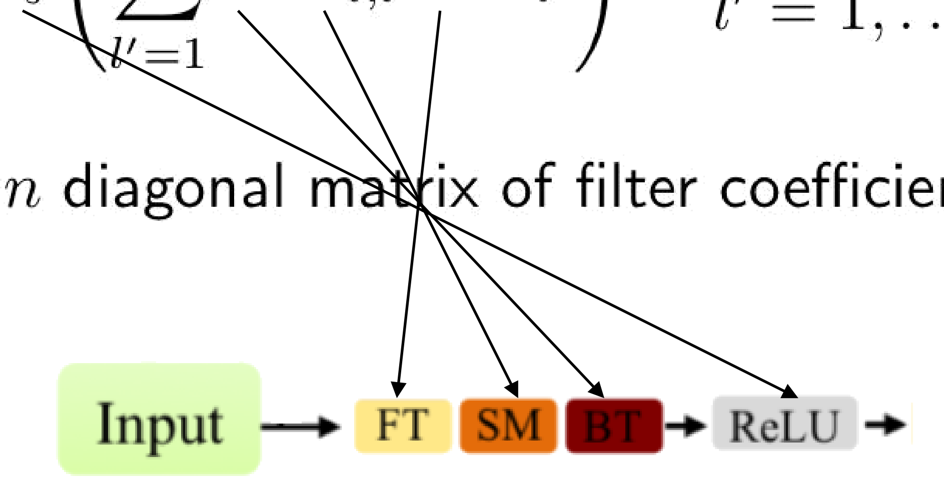
$$\widehat{(f \star g)} = \hat{f} \cdot \hat{g}$$

Spectral CNNs

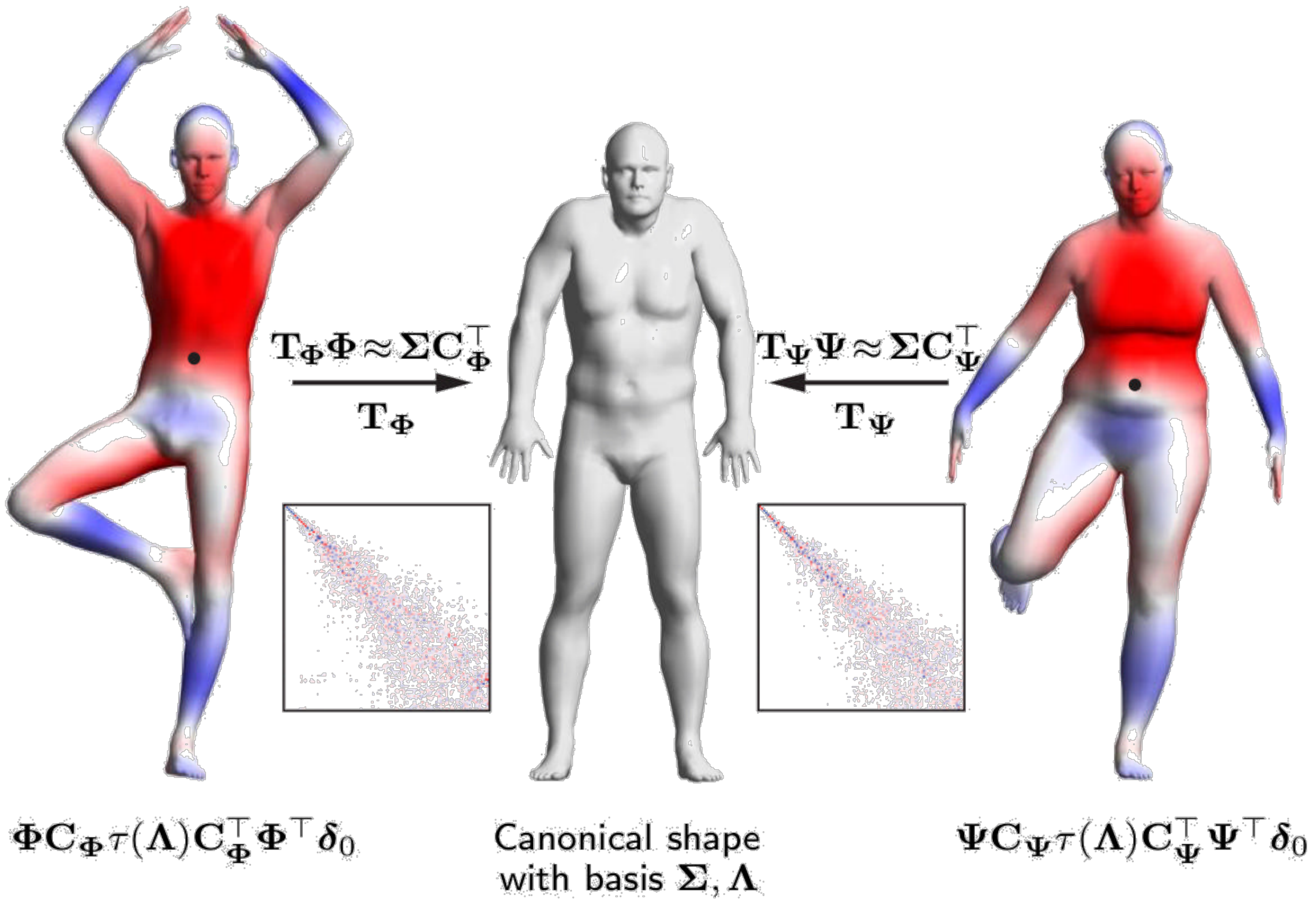
Convolutional layer expressed in the spectral domain

$$\mathbf{g}_l = \xi \left(\sum_{l'=1}^{p} \mathbf{\Phi} \hat{\mathbf{W}}_{l,l'} \mathbf{\Phi}^\top \mathbf{f}_{l'} \right) \qquad \begin{array}{l} l = 1, \ldots, q \\ l' = 1, \ldots, p \end{array}$$

where $\hat{\mathbf{W}}_{l,l} = n \times n$ diagonal matrix of filter coefficients



Spectral Basis Synchronization

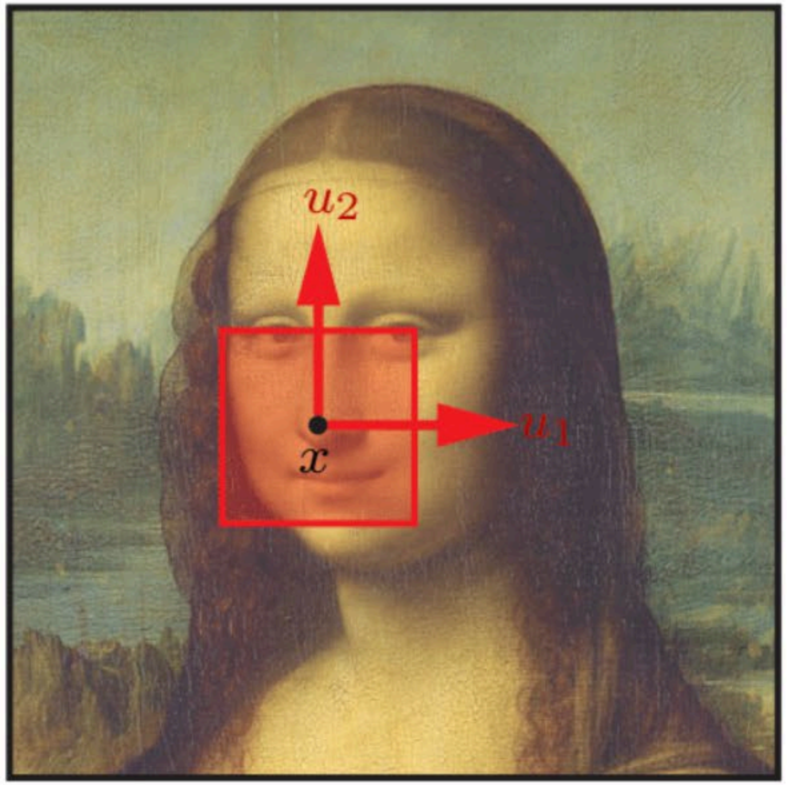


Apply spectral filter $\tau(\lambda)$ in synchronized bases $\Phi \mathbf{C}_\Phi$ and $\Psi \mathbf{C}_\Psi$

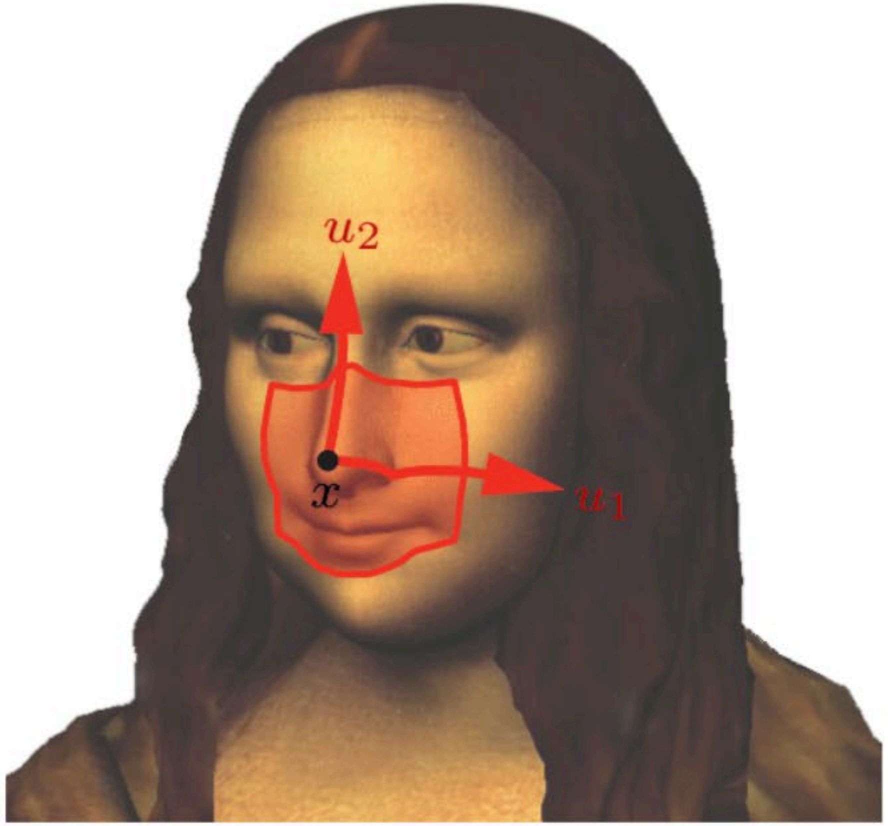
$\Rightarrow$ similar results!

Yi et al. 2017

Primal Methods: Patch Operators



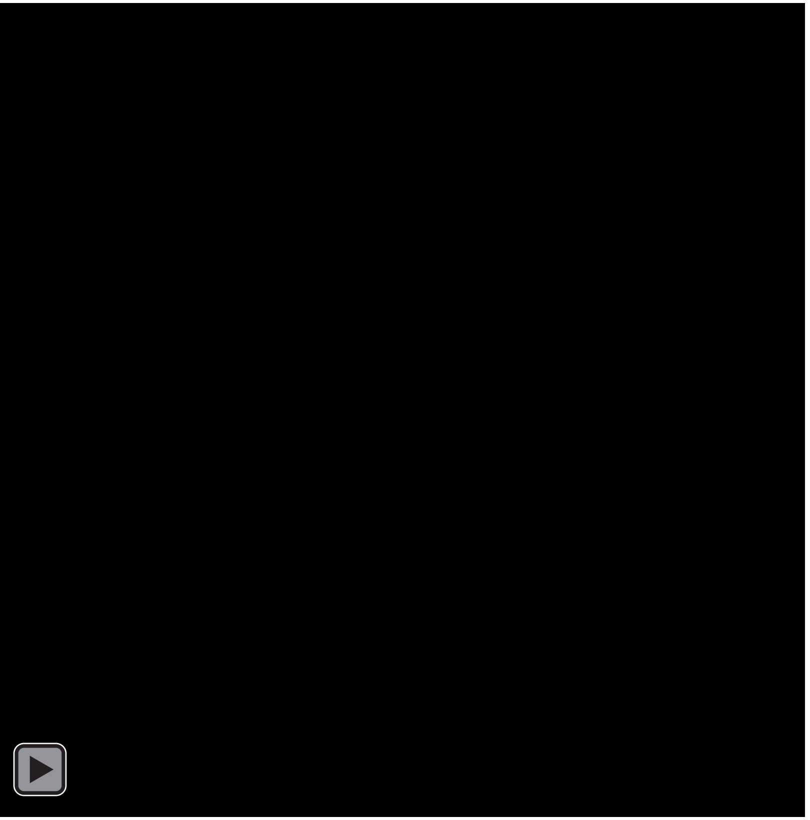
Image



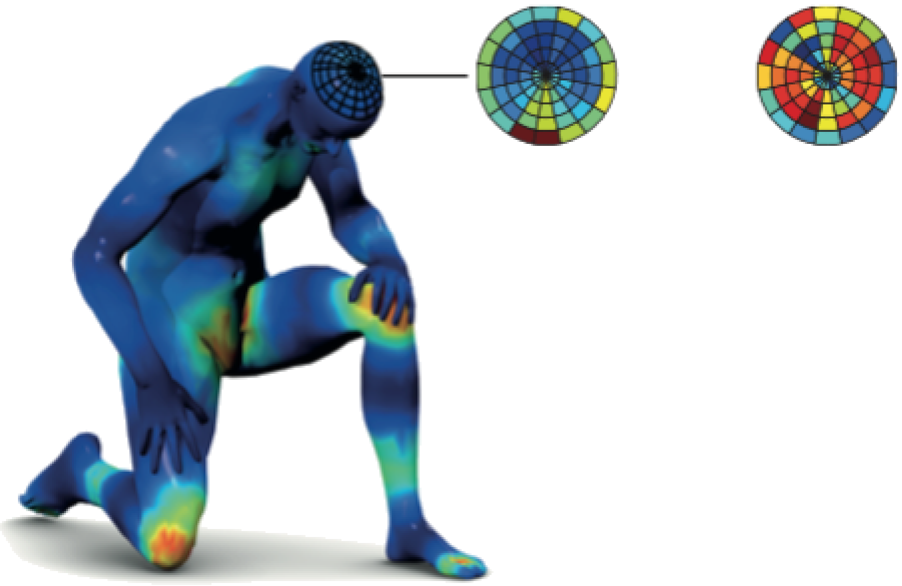
Manifold

Geodesic CNNs

- Geodesic convolution = apply filter a to patches extracted from $f \in L^2(X)$ in local geodesic polar coordinates

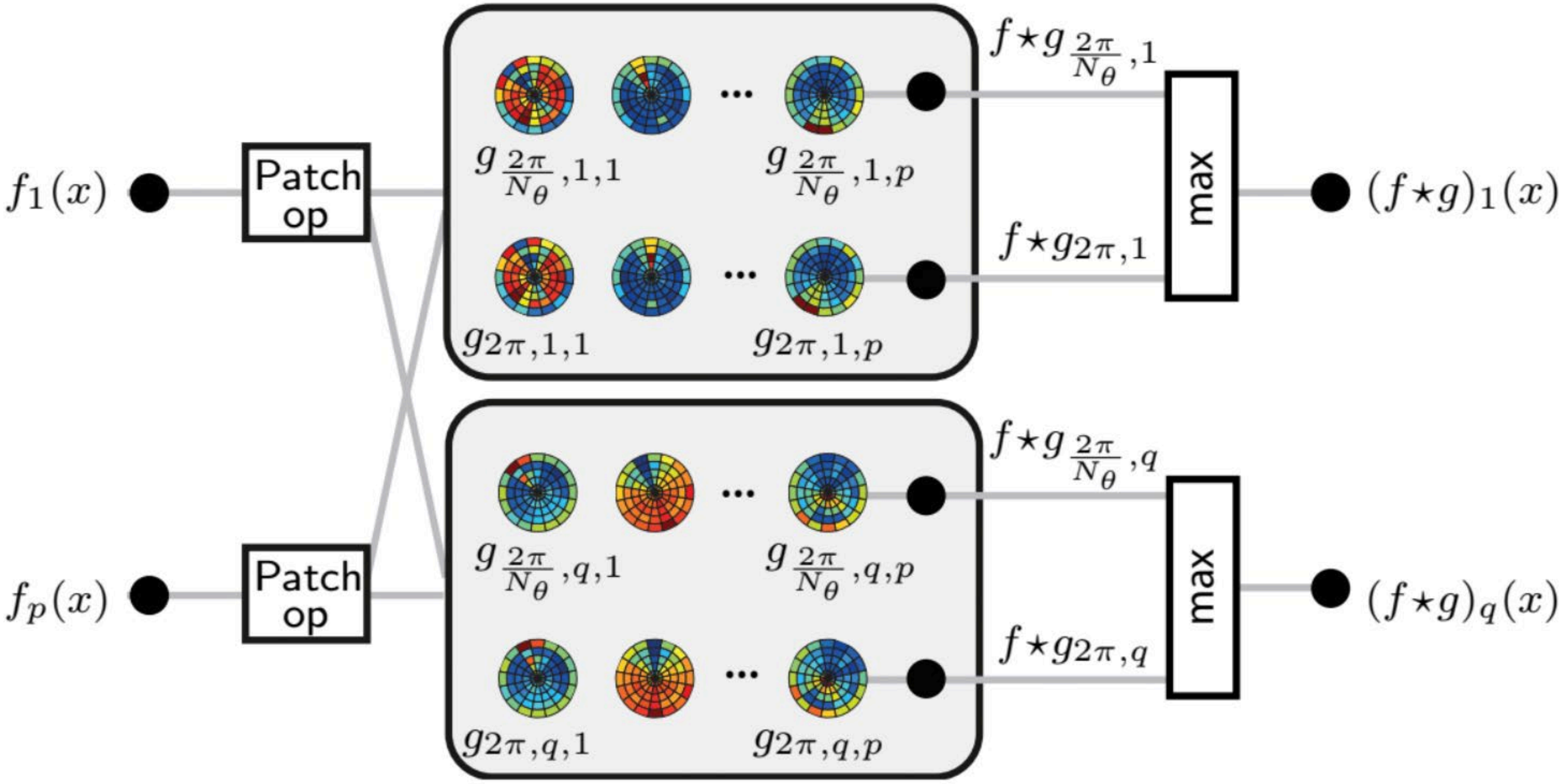


$$(f \star a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r,\theta)} \underbrace{a(\theta, r)}$$



rotation ambiguity

Handling Rotation Ambiguity via Pooling



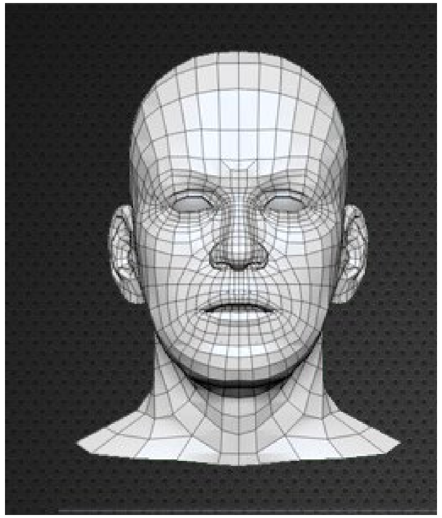
Conv. layer $$(f_l \star g)_{\Delta\theta,l}(x) = \xi\left(\sum_{\ell=1}^{p}(f_\ell \star g_{\Delta\theta,l,\ell})(x)\right)$$

Angular max pooling $$(f \star g)_l(x) = \max_{\Delta\theta}(f \star g)_{\Delta\theta,l}(x)$$

Today: Shape Differences and Variability -- Class Wrap-Up

Object Shape, Material and Appearance Differences

What Exactly is a Shape Difference?

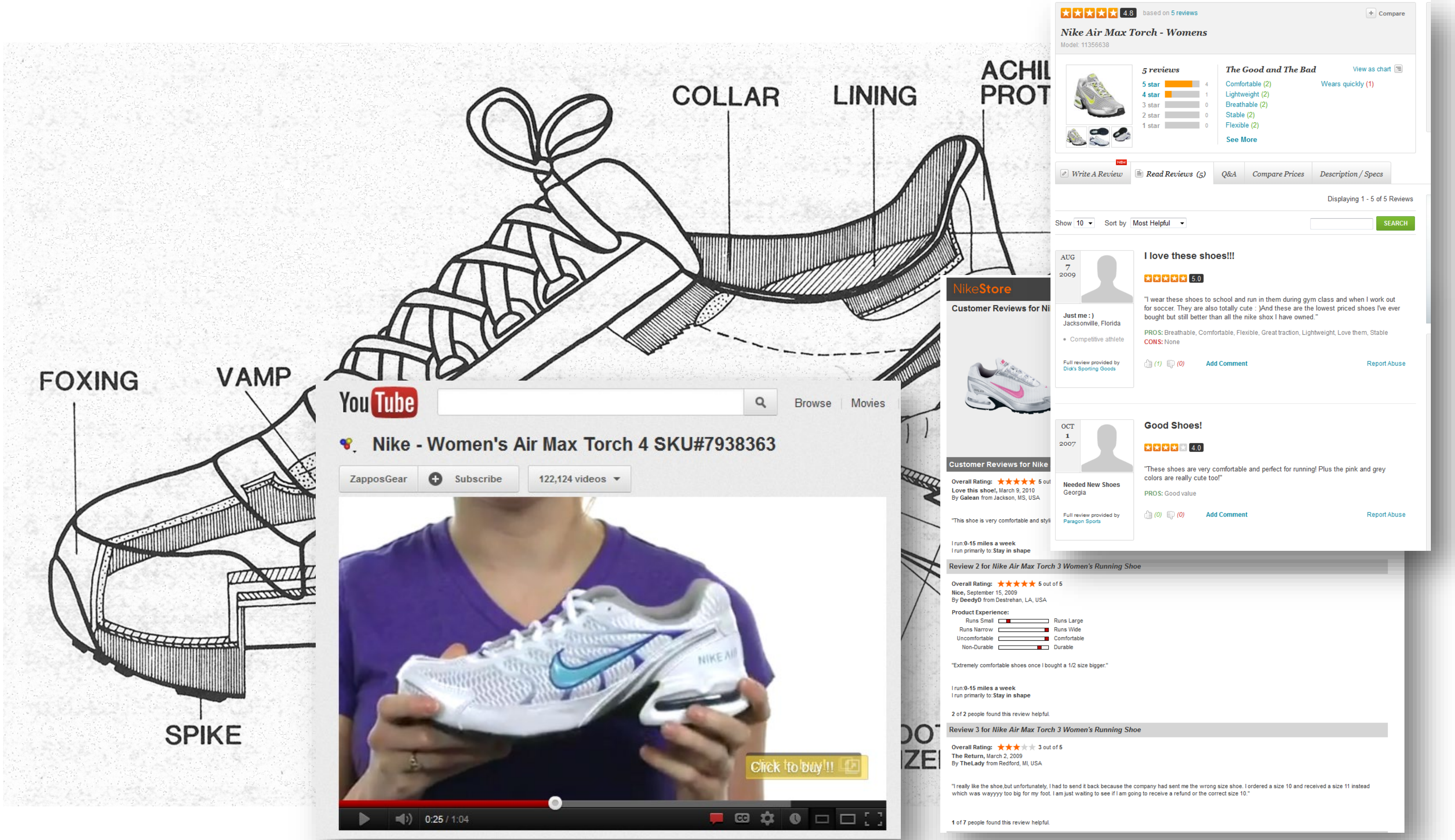


vs.

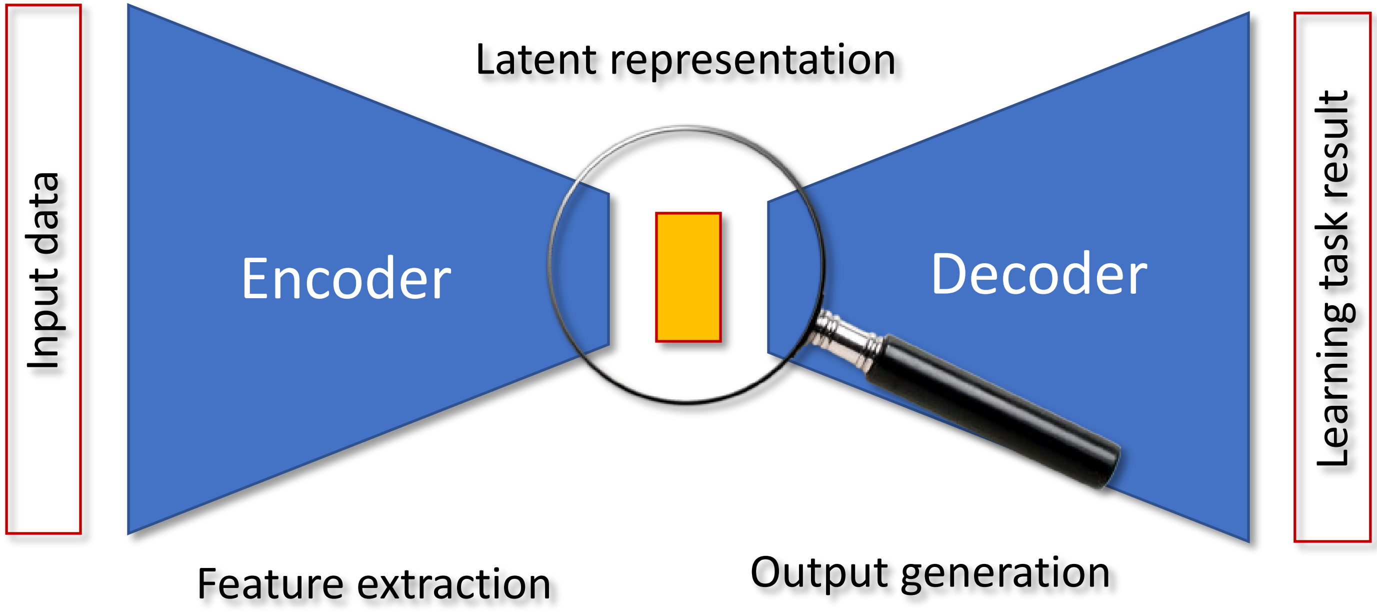
vs.

Making 3D shape differences first class citizens

Search Engines Based on Differences?

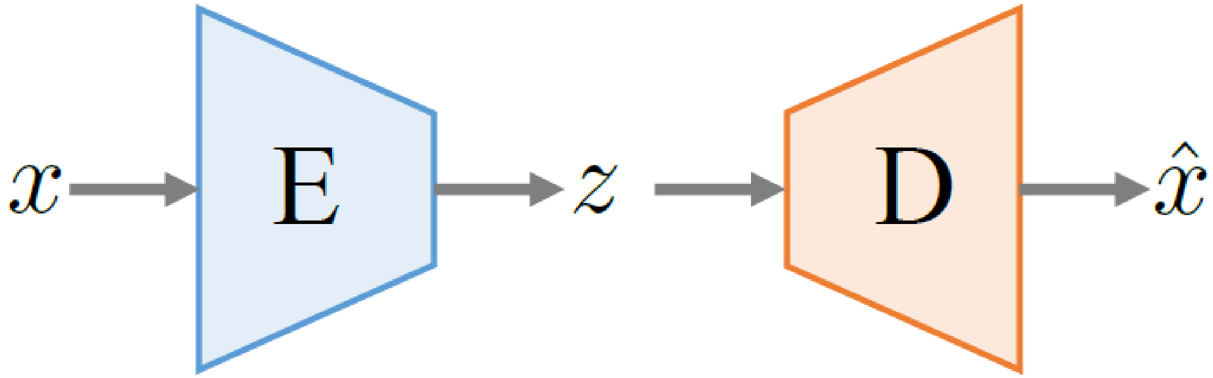


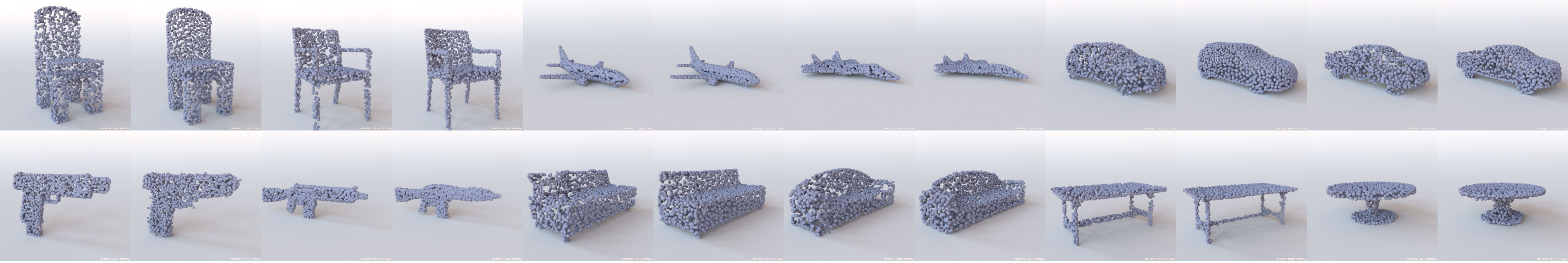
Latent Spaces in ML, Supervised or Not



A latent code acts as a low-d proxy for input data w.r.t. a learning task

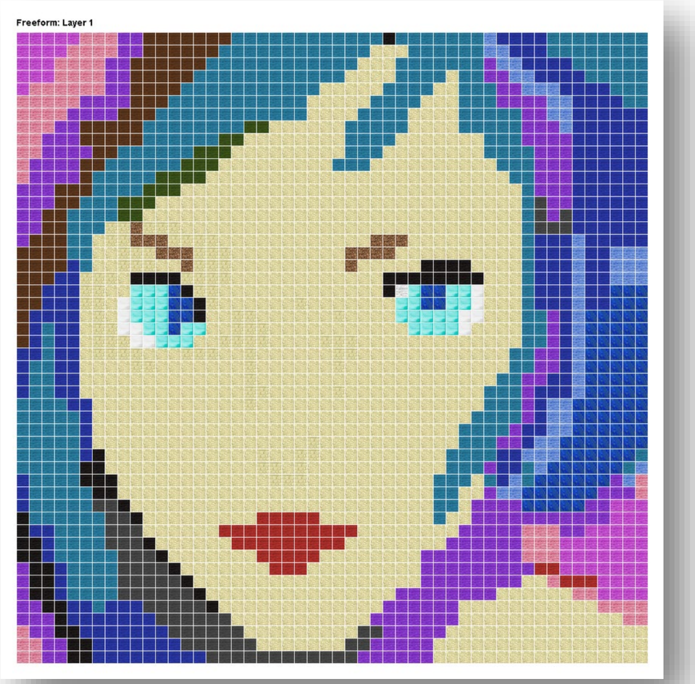
Point Cloud Auto-Encoders



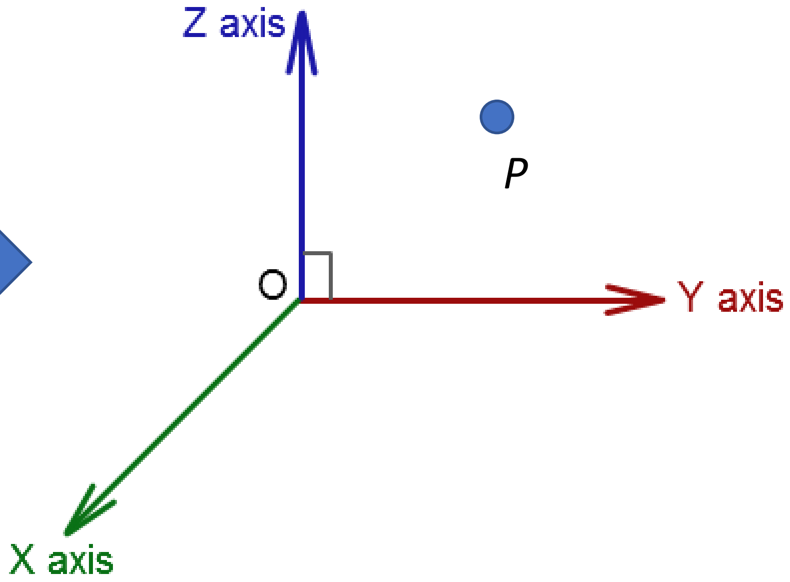


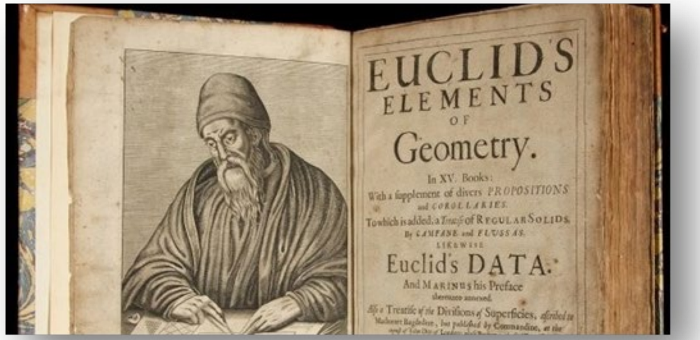
What Exactly is a Latent Space Representation?

- Input



- Latent code

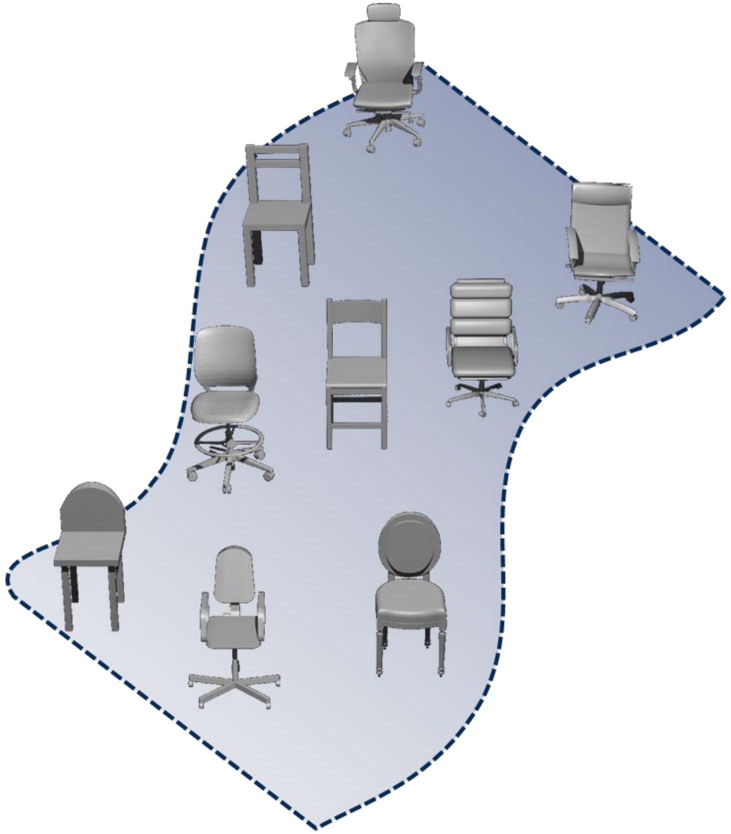




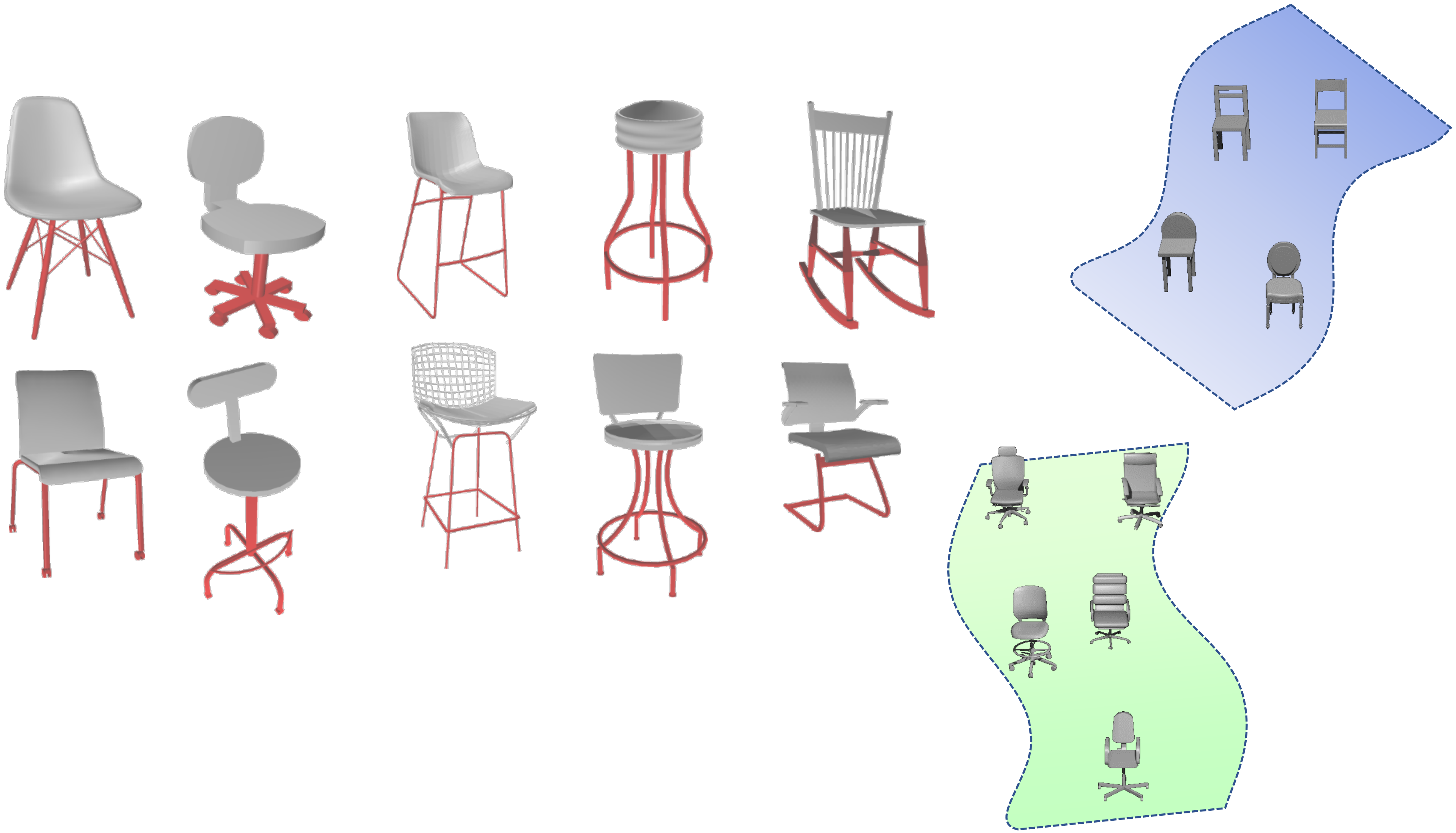
Is a Shape Difference Just a Vector in a Latent Space?

Continuous Shape Variability

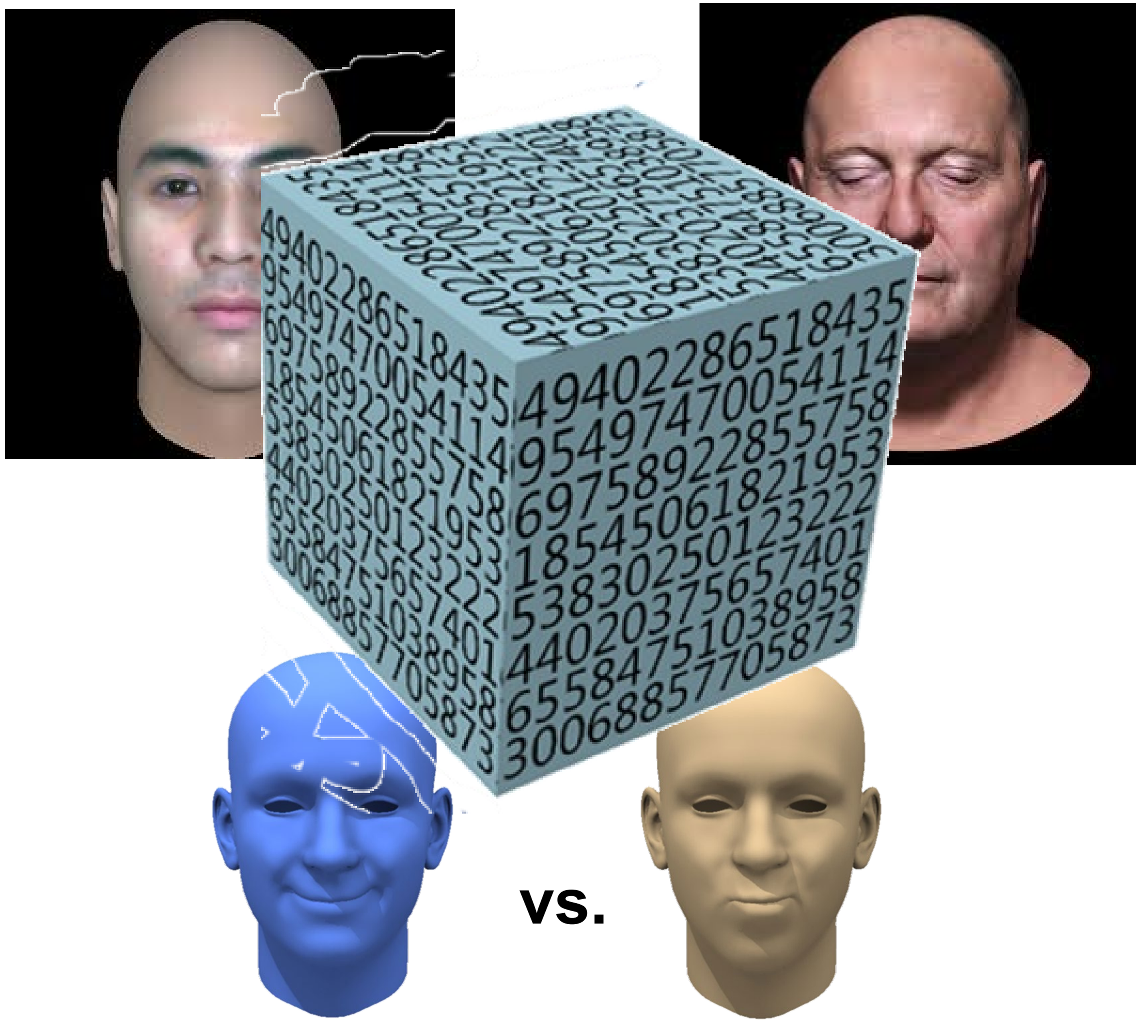
A chair manifold?



Combinatorial or Discrete Variability

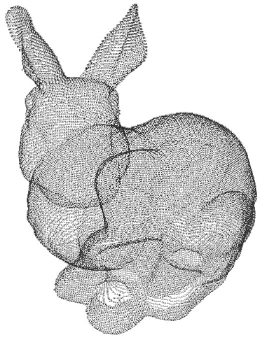


What Exactly is a Shape Difference?

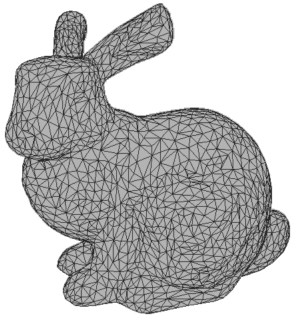


Where and how are the shapes different?

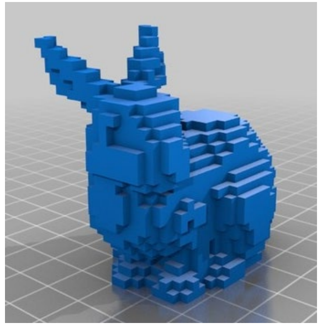
A Challenge: Multiple 3D Representations



Point Cloud



Mesh



Volumetric

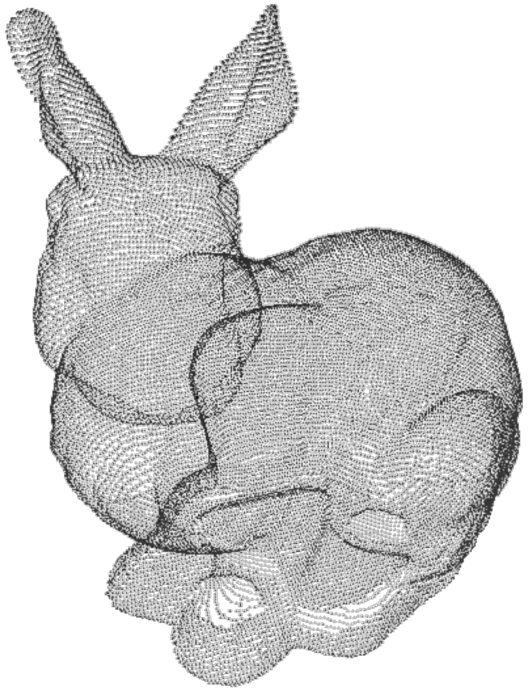
Projected View

RGB(D)

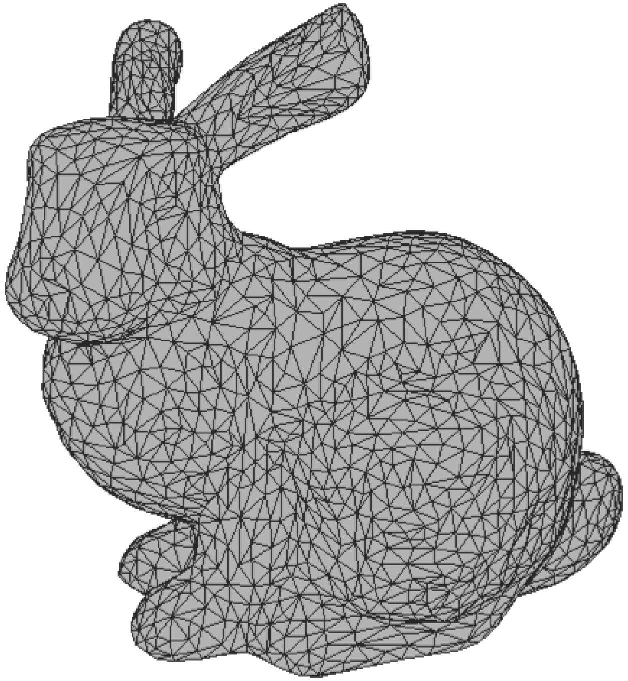
...

Irregular representations such as point clouds or meshes are a challenge for machine learning algorithms

Underlying Shape Surface Discretizations



Point cloud



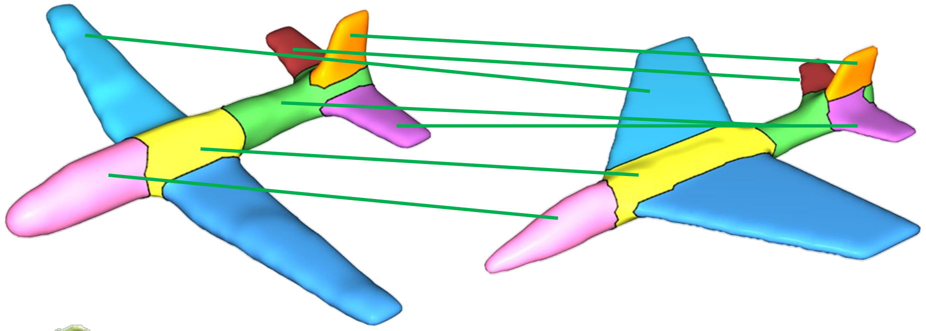
Mesh

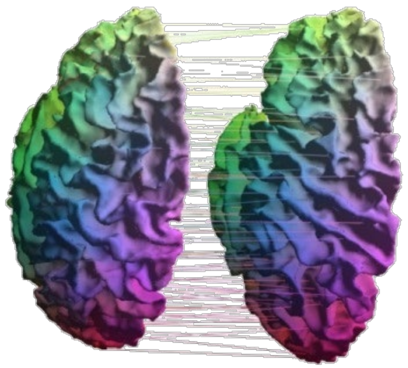
Must distinguish differences of the representations from differences of the shapes

Continuous Shape Differences Under a Map

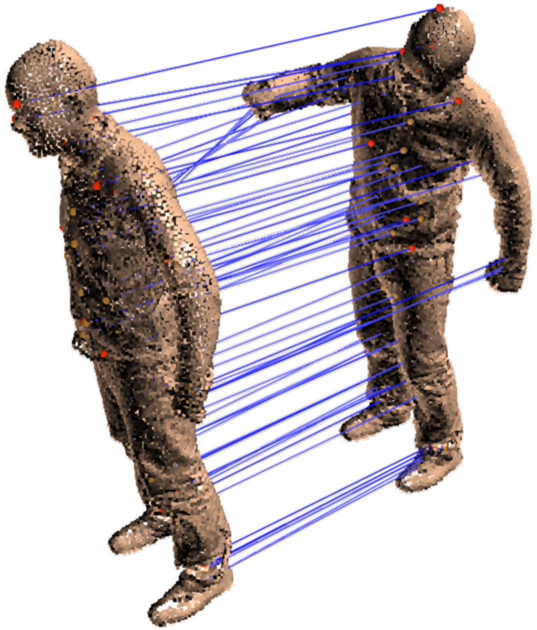
Surface Maps and Distortions

- Shape correspondences
- Often computed by minimizing some measure of distortion

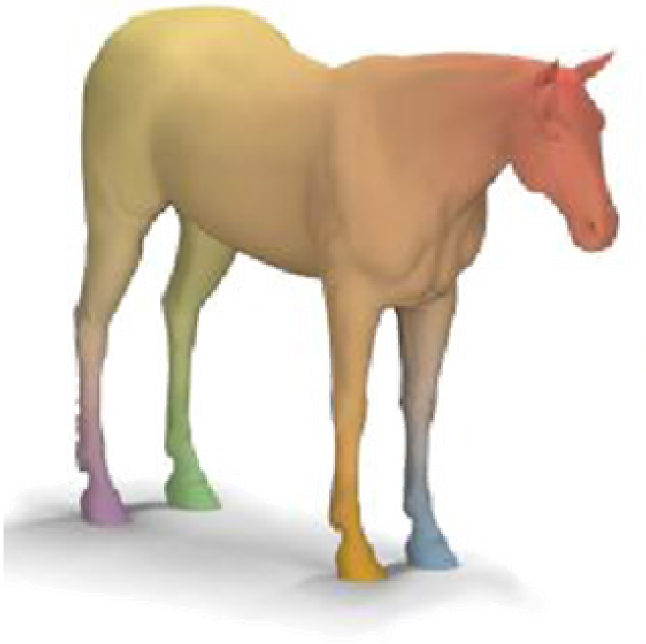




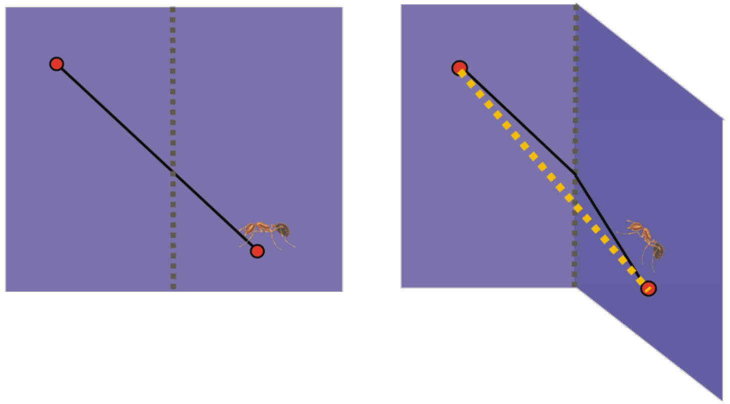
Yet the distortion estimates are forgotten as soon as the correspondences are computed

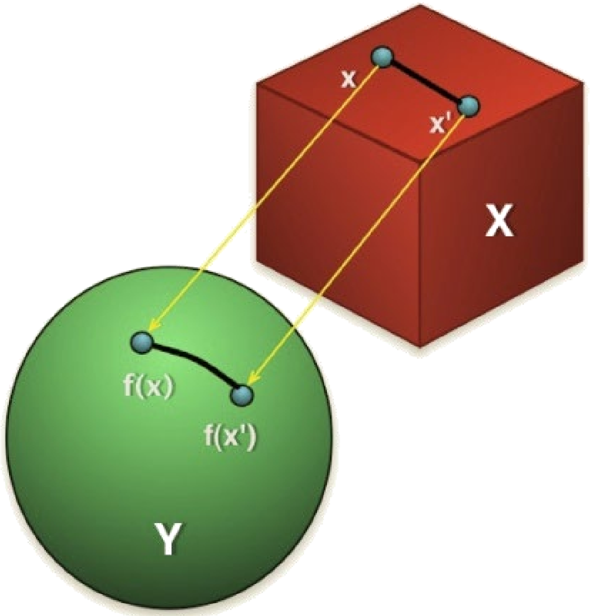


Subtlety 1: Correspondences at Multiple Scales



Subtlety 2: Intrinsic or Extrinsic Distances



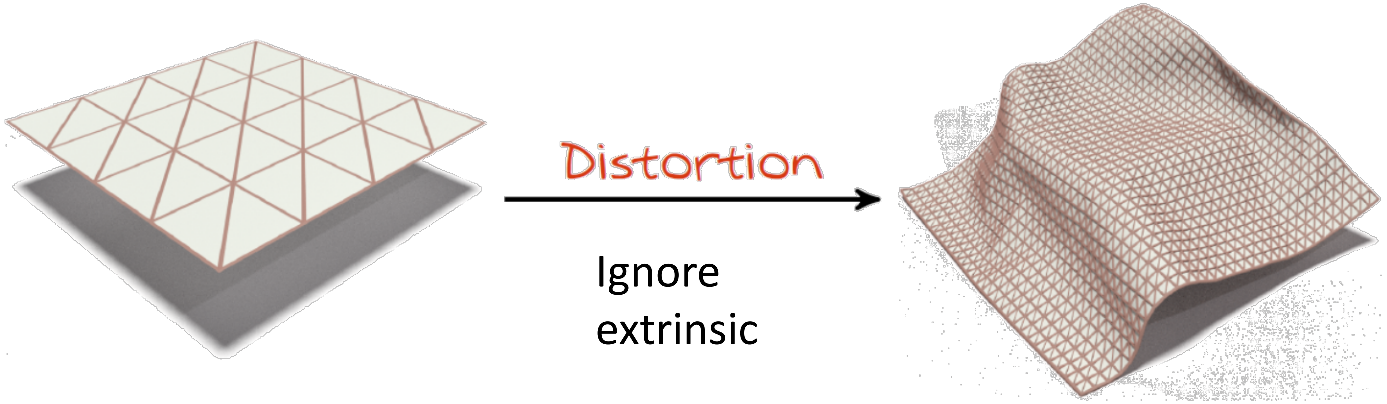


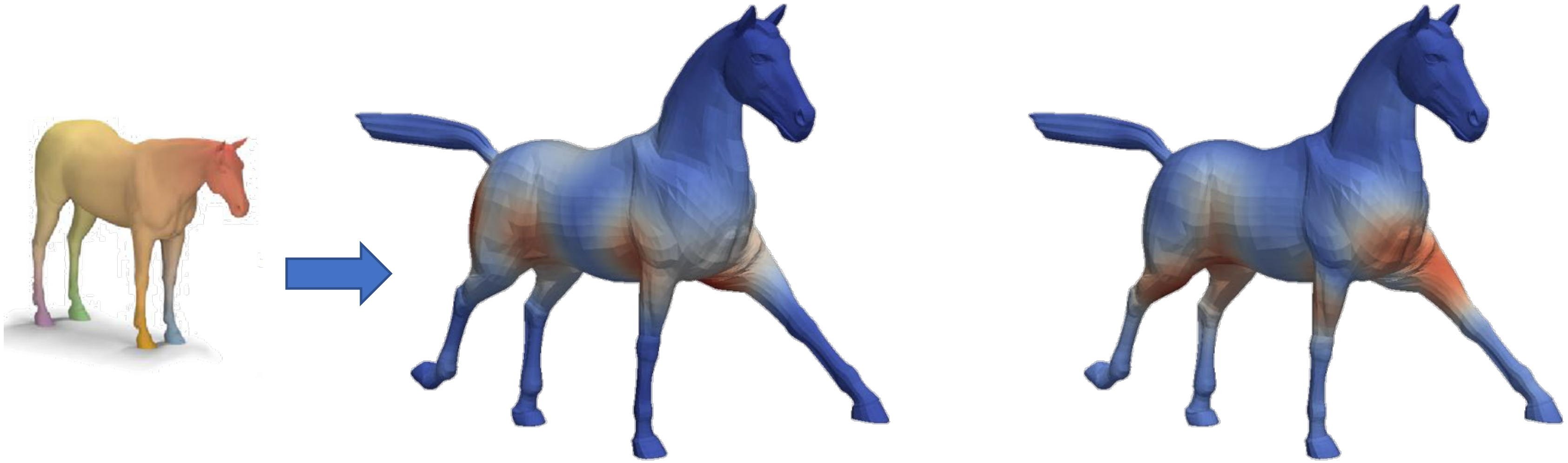
Measure on the surface or in the ambient space?

Shape Differences from Intrinsic Distortions

[R. Rustamov, M. Ovsjanikov, O. Azercot, M. Ben-Chen, F. Chazal, L. Guibas; Siggraph ’13]

Intrinsic Changes to a Metric

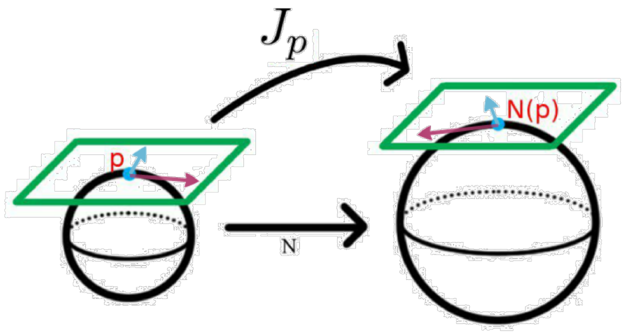


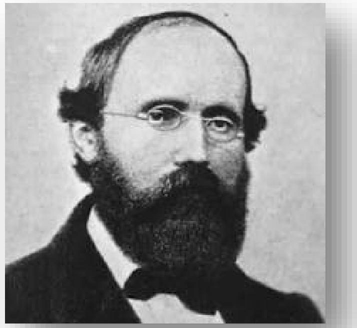


Intrinsic distortions: Area distortion Conformal distortion Length …

Classical Approach to Relating Intrinsic Metrics

To measure distortions induced by a map, we track how inner products of vectors change after transporting





Riemann

Challenges:

- point-wise information only, hard to aggregate
- noisy

A Functional View of Distortions

To measure distortions induced by a map, track how inner products of vectors change after transporting.

Riemann

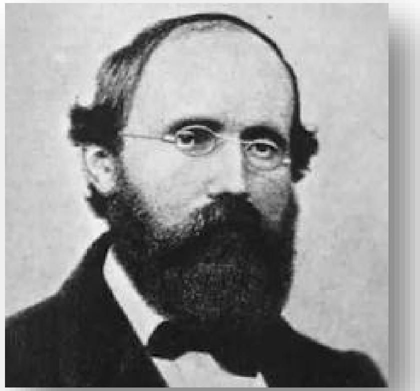
To measure distortions induced by a map, track how inner products of functions change after transporting.

The Art of Measurement

- A metric is defined by a functional inner product

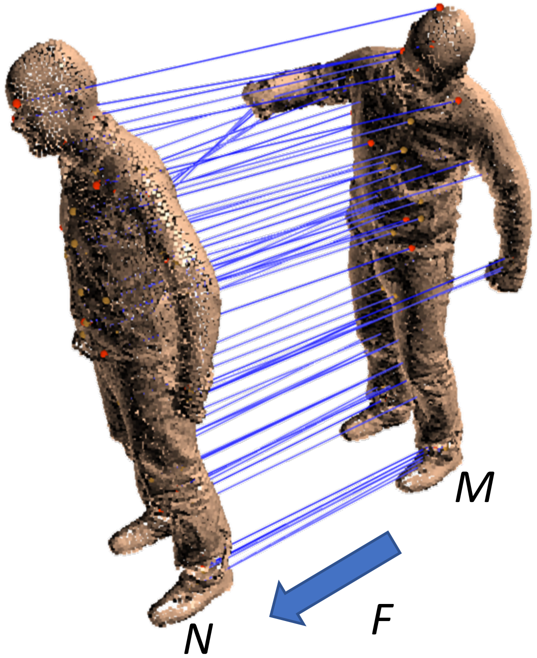
$$h^M(f,g) = \int_M f(x)g(x)d\mu(x)$$

- So we can compare *M* and *N* by comparing



Riemann

$$h^N(F(f),F(g))$$



$$h^M(f,g)$$

The functional map *F* transports these functions to *N*, where we repeat this measurement with the inner product $h^N(F(f),F(g))$

Inner Products of Functions

$$\langle f, g \rangle = \left\langle \sum_i f_i h_i(x), \sum_j g_j h_j(x) \right\rangle$$

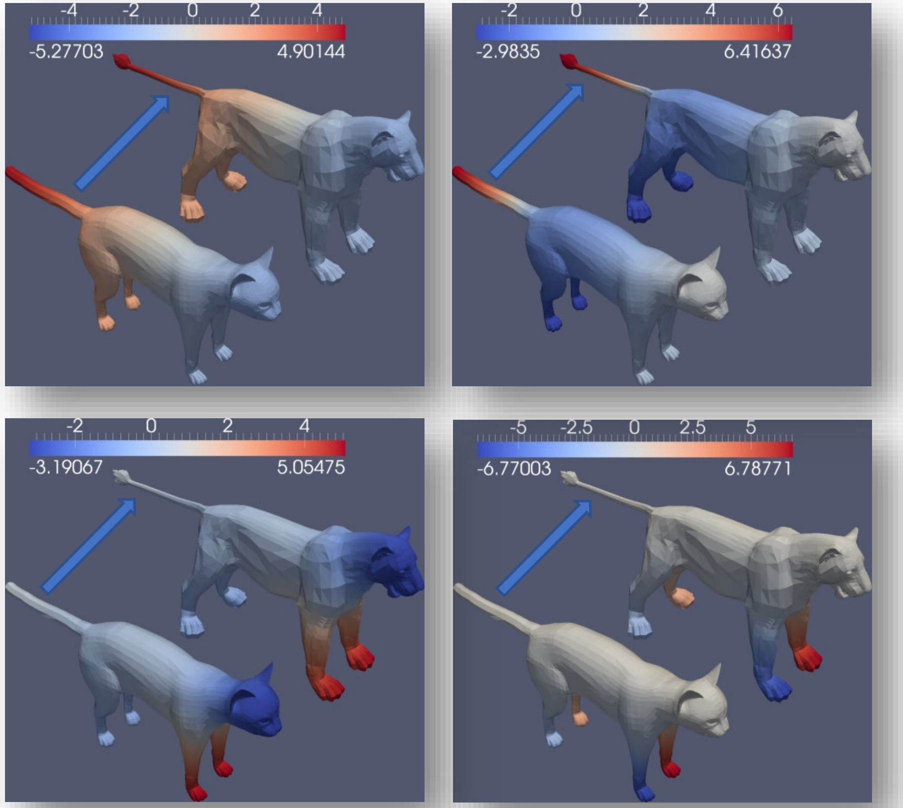
$$= \sum_{ij} f_i g_j \langle h_i(x), h_j(x) \rangle$$

$$= f^\top A g$$

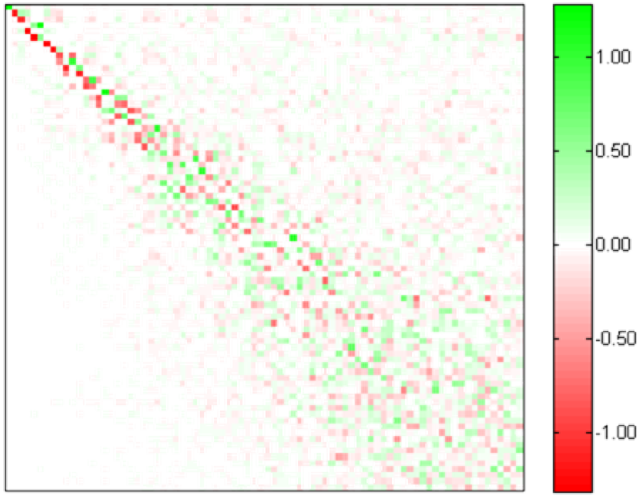
Area weights matrix

Starting from a Functional Map F

from cat to lion



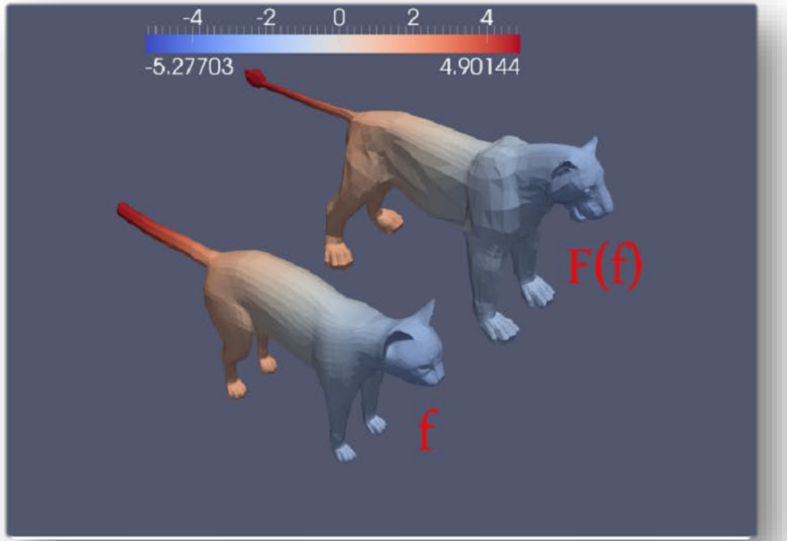
Functions on cat are transferred to lion using F

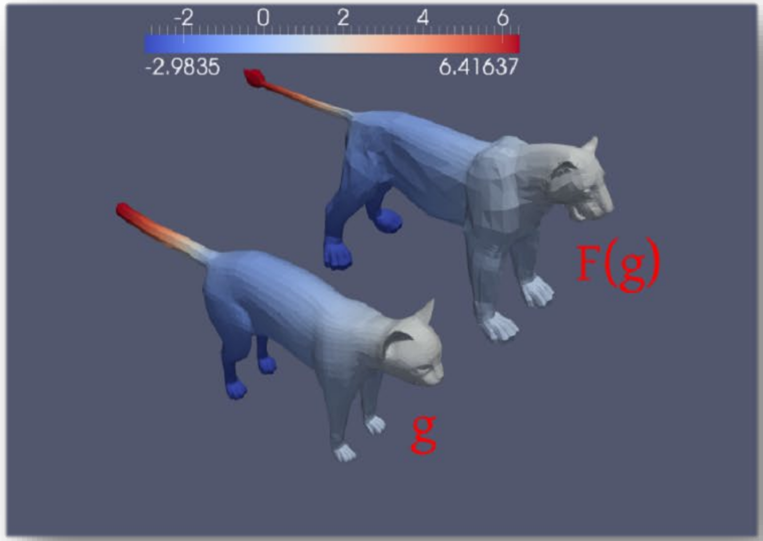


F is a linear operator (matrix)

$$F : L^2(cat) \rightarrow L^2(lion)$$

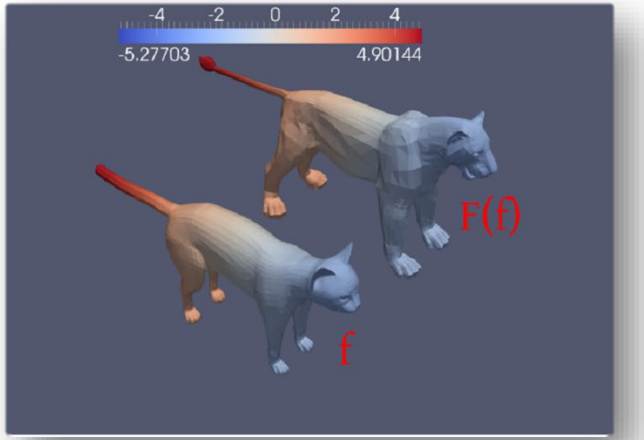
Measurement Discrepancies

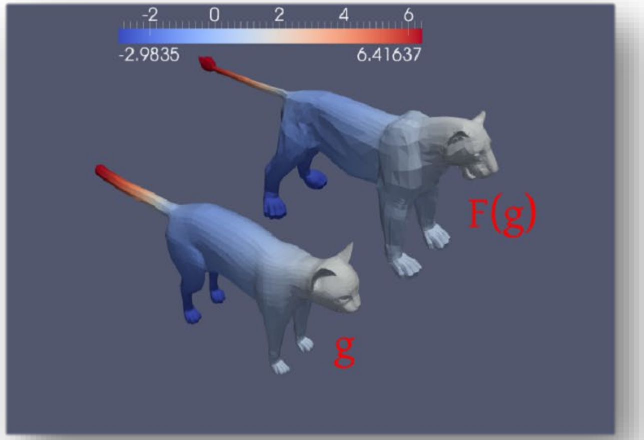




$$\int_{lion} \underbrace{F(f)F(g)\, d\mu_l}_{\text{after}} \neq \int_{cat} \underbrace{f g\, d\mu_c}_{\text{before}}$$

Measurement Discrepancies





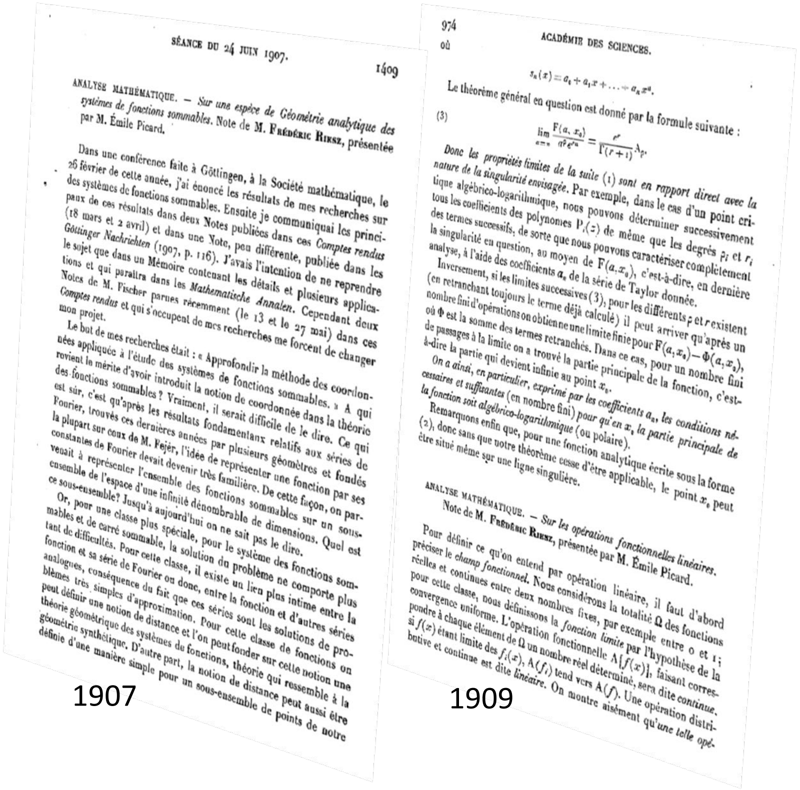
$$\int_{lion} F(f)F(g)\, d\mu_l \neq \int_{cat} fg\, d\mu_c$$

after　　before

Both can be considered as inner products on the cat

The Universal Compensator

Comptes Rendus Hebdomadaires des
Séances de l'Académie des Sciences de Paris


SÉANCE DU 24 JUIN 1907. 1409

ANALYSE MATHÉMATIQUE. — *Sur une espèce de Géométrie analytique des systèmes de fonctions sommables.* Note de M. Frédéric Riesz, présentée par M. Émile Picard.

974 ACADÉMIE DES SCIENCES.

ANALYSE MATHÉMATIQUE. — *Sur les opérations fonctionnelles linéaires.* Note de M. Frédéric Riesz, présentée par M. Émile Picard.

1907 1909

Riesz Representation Theorem

There exists a linear operator

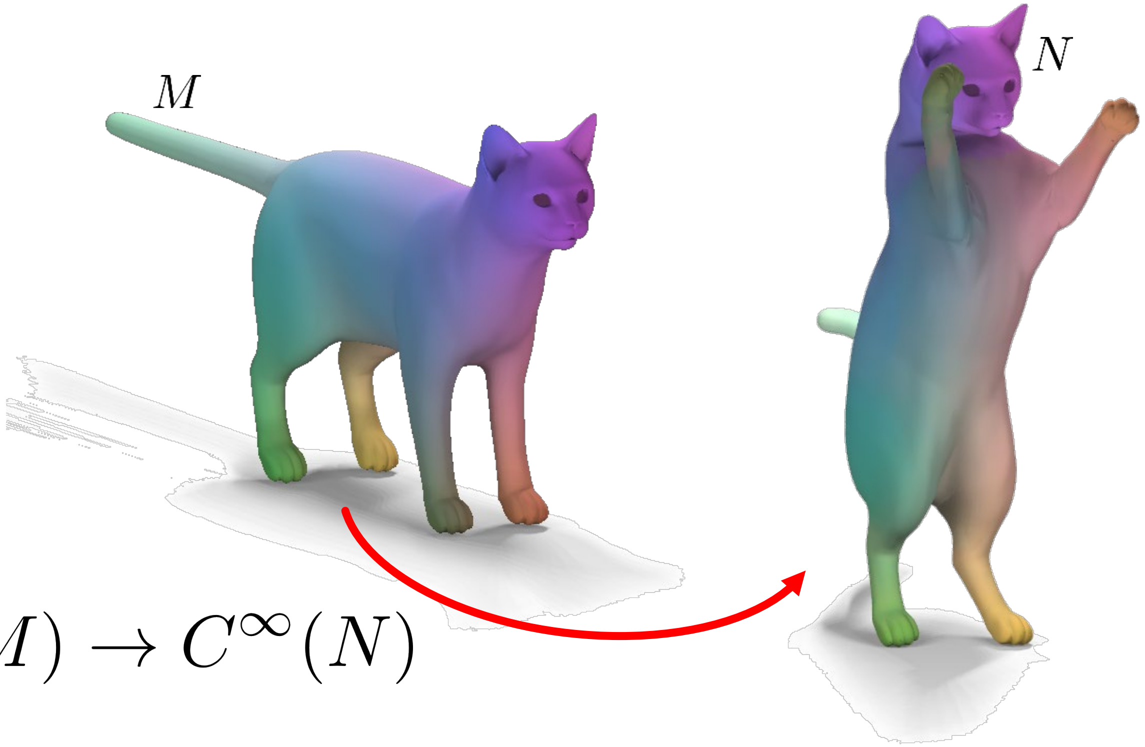
$$V : L^2(cat) \to L^2(cat)$$

such that

$$\langle f, g \rangle_{\text{after}} = \langle f, V(g) \rangle_{\text{before}}$$

Frigyes Riesz

Riesz Representation Theorem



$$F : C^\infty(M) \to C^\infty(N)$$

$$\exists V \text{ s.t.}$$
$$\langle F(f), F(g)\rangle^N = \langle f, V(g)\rangle^M \ \forall f, g$$

Sanity Check

$$
\begin{aligned}
\langle f, g \rangle &\approx f^\top A g \\
\langle F(f), F(g) \rangle^N &\approx [F(f)]^\top A_N [F(g)] \\
&= f^\top \cdot F^\top A_N F \cdot g \\
&= f^\top \cdot (A_M A_M^{-1}) F^\top A_N F \cdot g \\
&= f^\top \cdot A_M (A_M^{-1} F^\top A_N F \cdot g) \\
&\approx \langle f, (A_M^{-1} F^\top A_N F) g \rangle
\end{aligned}
$$

Area-Based Shape Difference:

$$V = A_M^{-1} F^\top A_N F$$

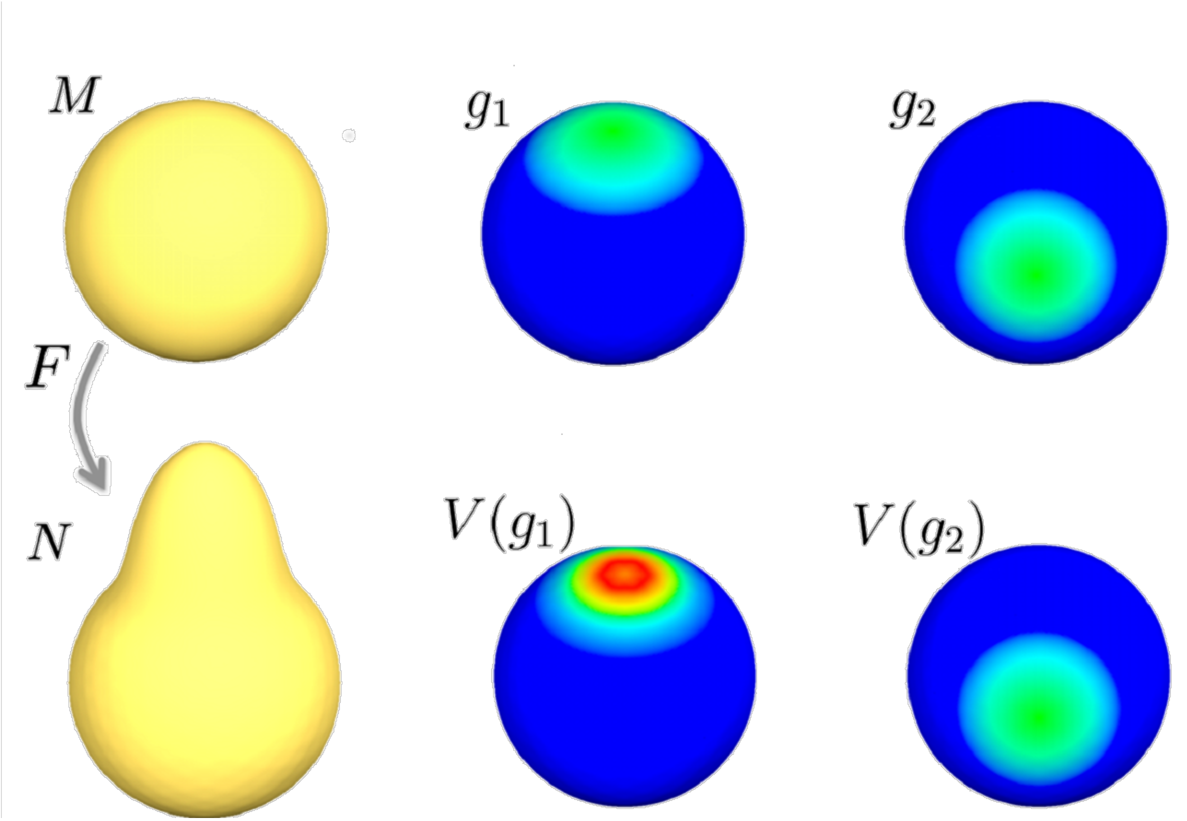
$$\int_{lion} F(f)F(g) \neq \int_{cat} fg$$

↓

$$\int_{lion} F(f)F(g) = \int_{cat} f V(g)$$

V maps functions on the cat to functions on the cat -- is a self-map of the domain

A Small Example of V



Note that *V* maps functions on *M* to functions on *M*

$$\int_N F(f)F(g) = \int_M fV(g)$$

Conformal Shape Difference R

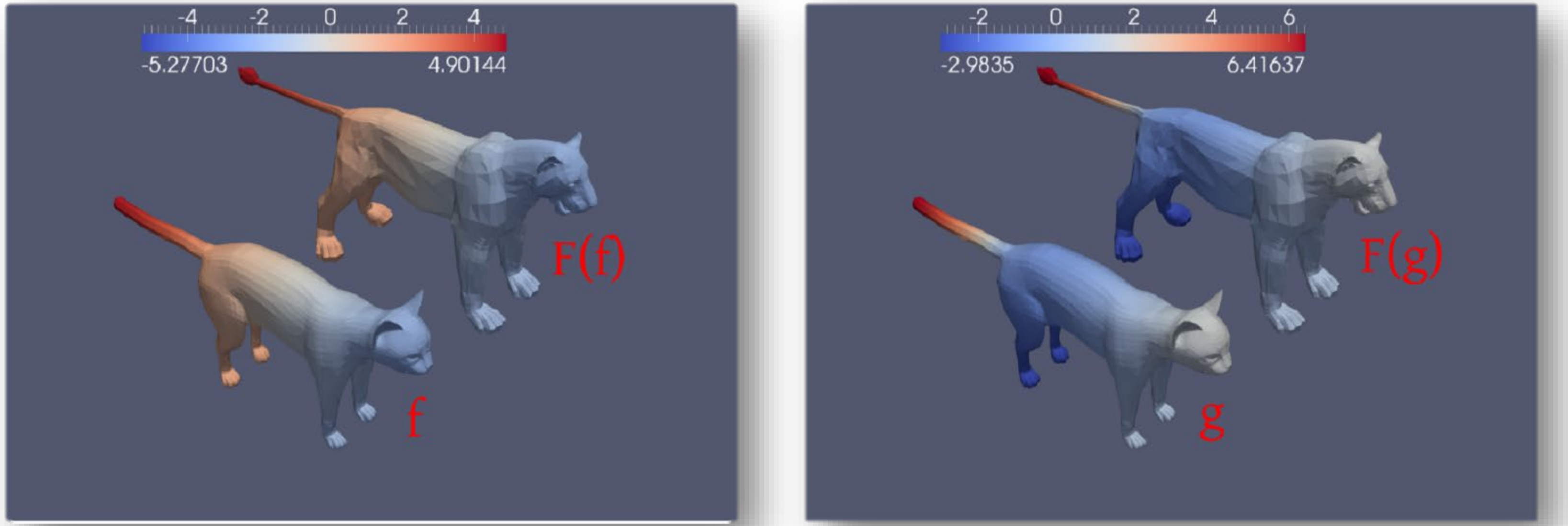
Consider a different inner-product of functions ...

get information about conformal distortion

$$\int_N \nabla F(f) \nabla F(g) = \int_M \nabla f \nabla R(g)$$

The choice of inner product should be driven by the application at hand.

Conformal Shape Difference *R*

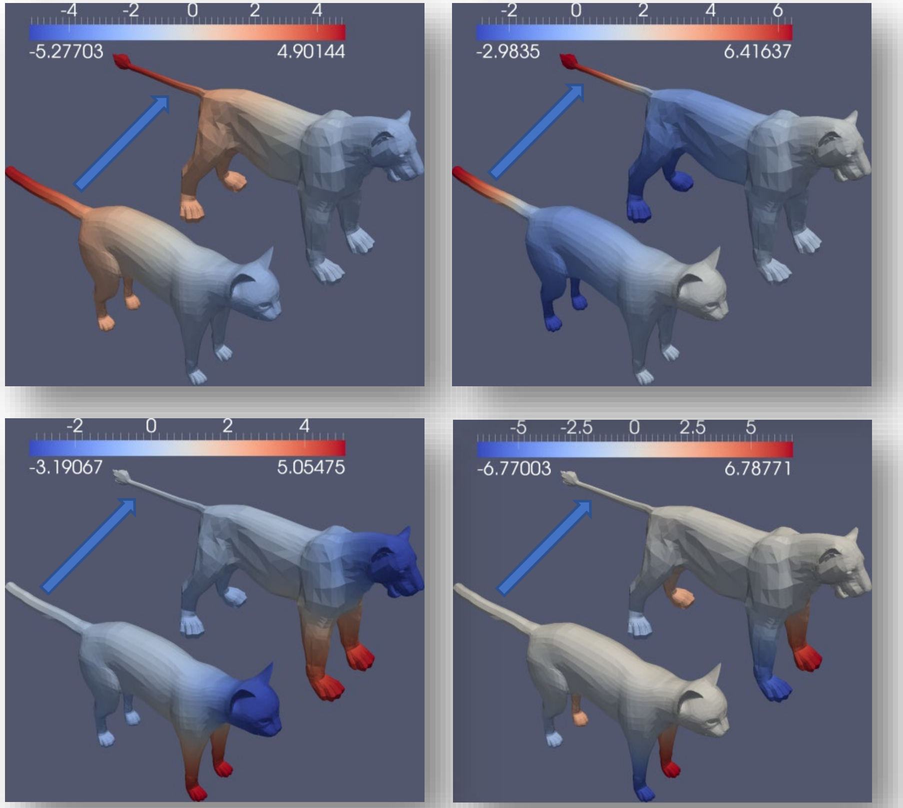


$$\int_{lion} \nabla F(f) \cdot \nabla F(g) \neq \int_{cat} \nabla f \cdot \nabla g \quad \forall f, g$$

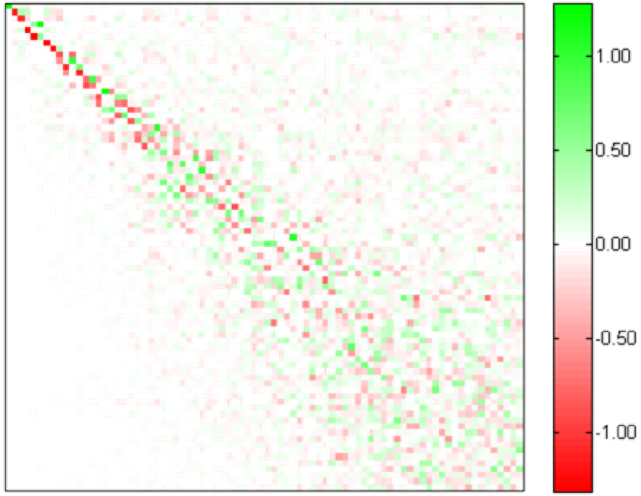
$$\int_{lion} \nabla F(f) \cdot \nabla F(g) = \int_{cat} \nabla f \cdot \nabla {\color{red}R}(g)$$

Input: Functional Map *F*

from cat to lion



Functions on cat are transferred to lion using *F*

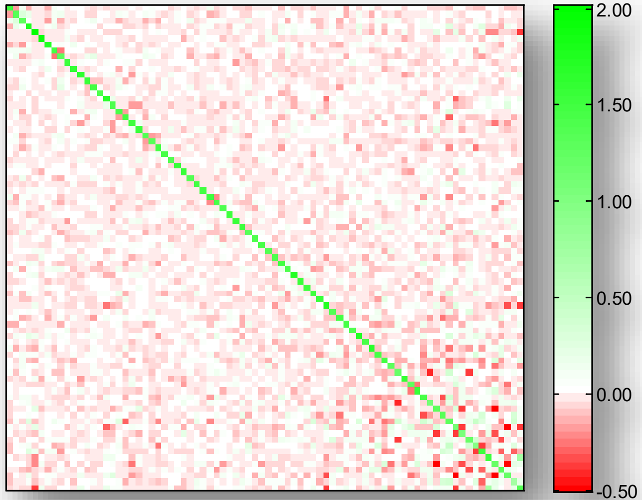


F **is a linear operator (matrix)**

$F : L^2(cat) \rightarrow L^2(lion)$

Shape Difference Operators

V – area-based shape difference

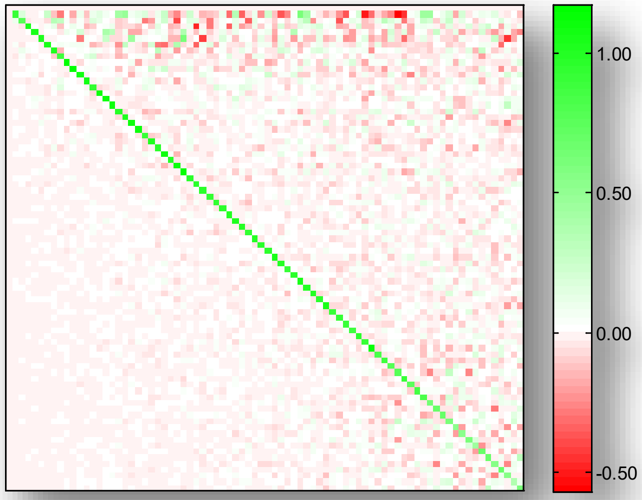


linear operator (matrix)

$$V : L^2(cat) \rightarrow L^2(cat)$$

$$\int_N F(f)F(g) = \int_M fV(g)$$

R – conformal shape difference

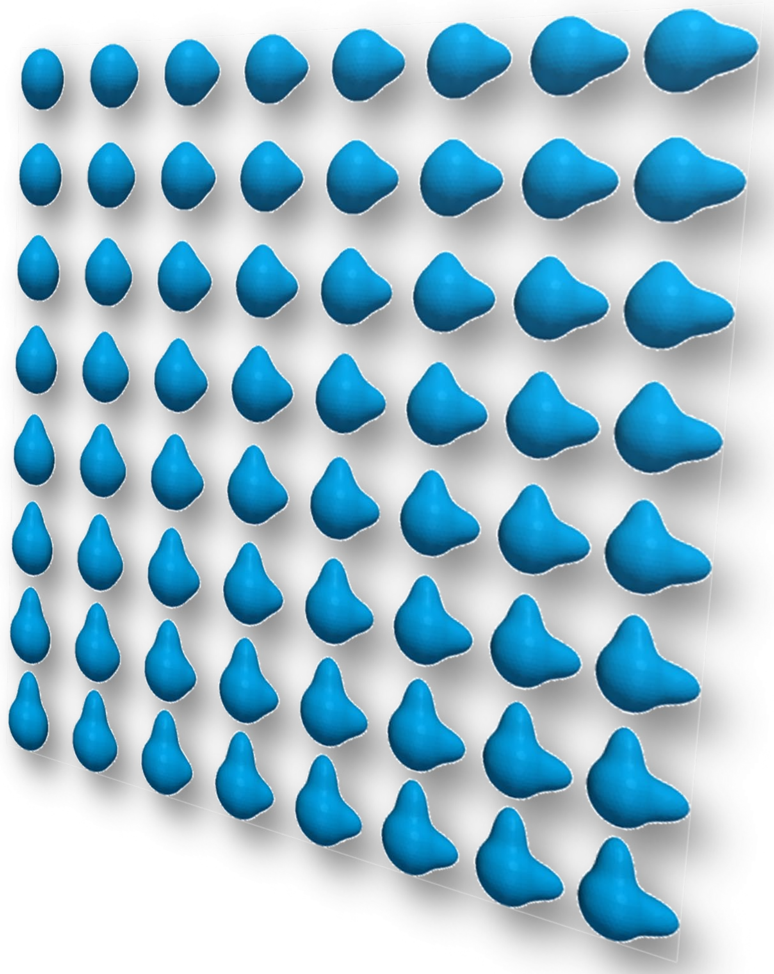


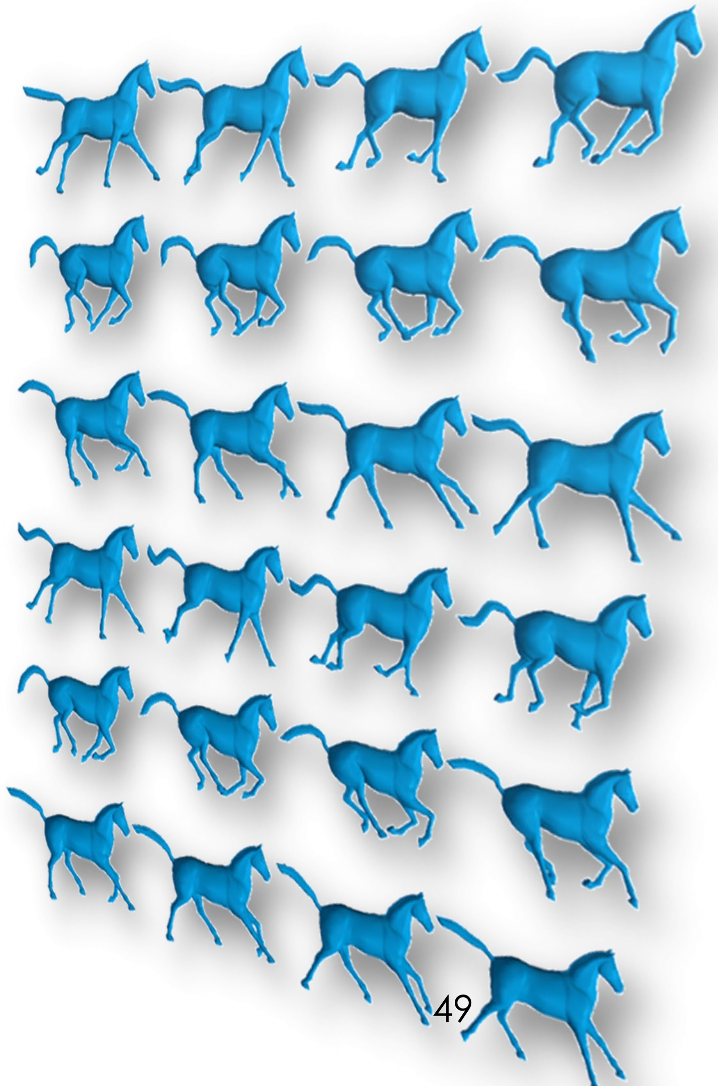
linear operator (matrix)

$$R : L^2(cat) \rightarrow L^2(cat)$$

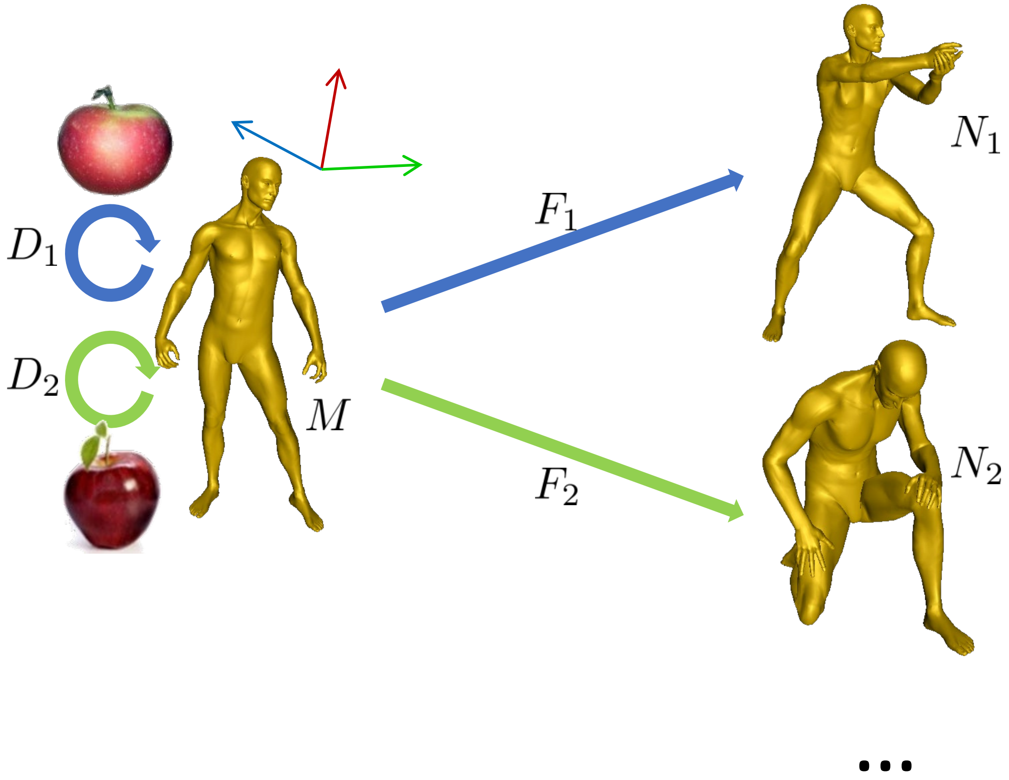
$$\int_N \nabla F(f) \nabla F(g) = \int_M \nabla f \nabla R(g)$$

Shape Differences in Collections

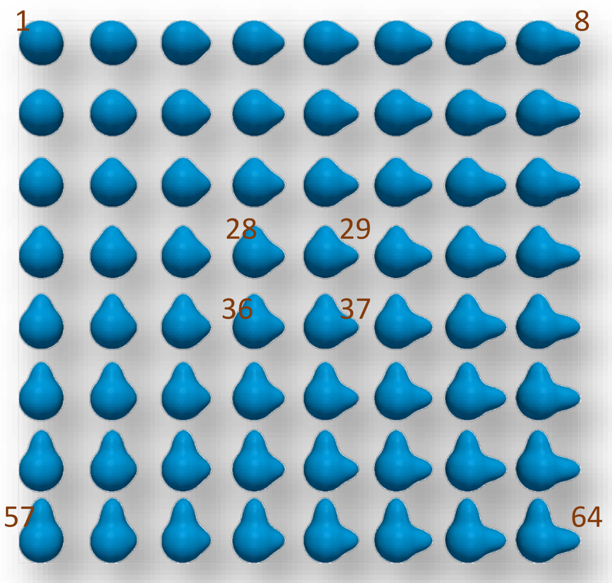


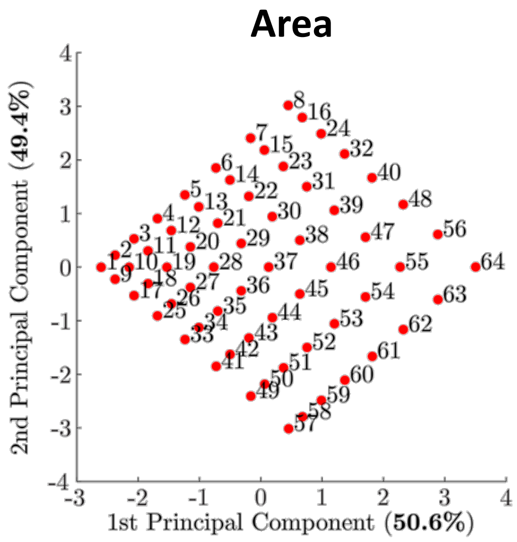


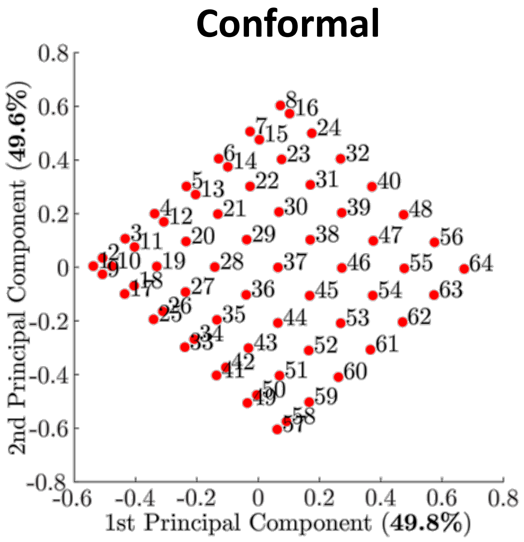
Comparing Differences

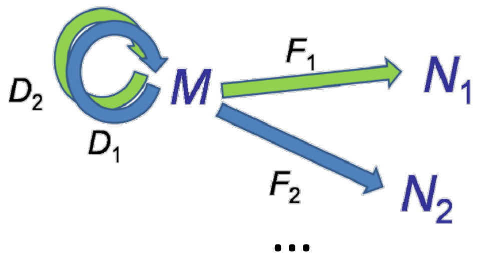


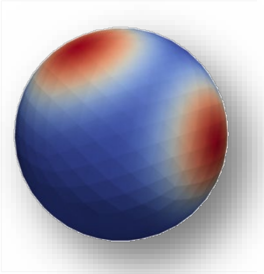
Intrinsic Shape Space

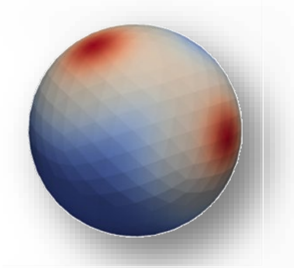






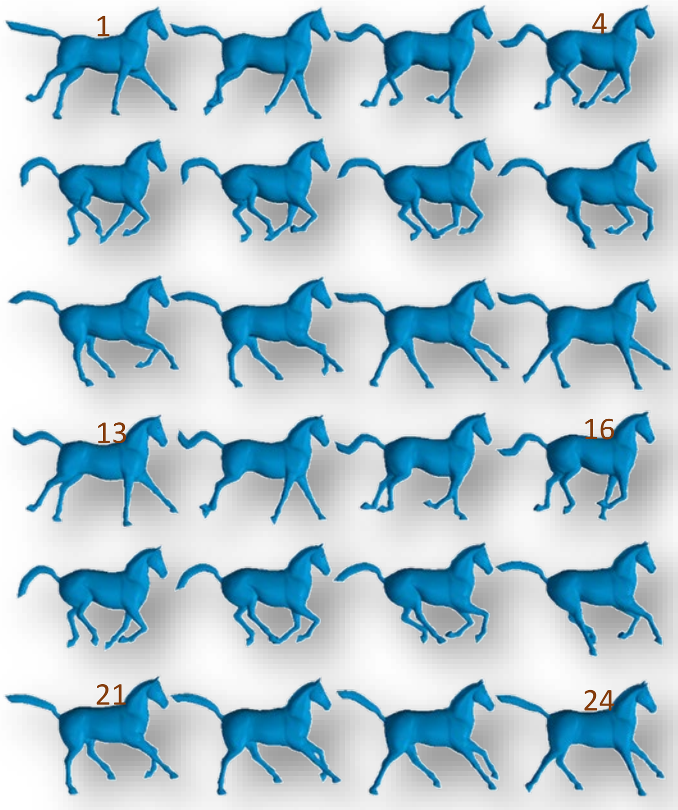


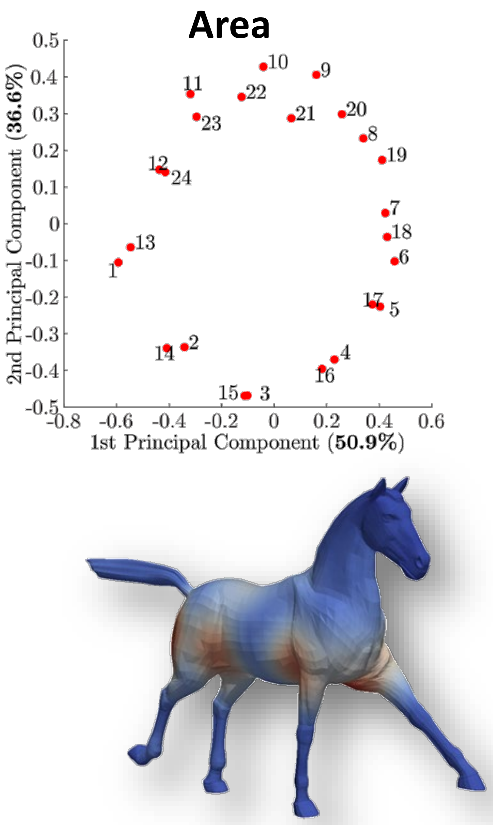


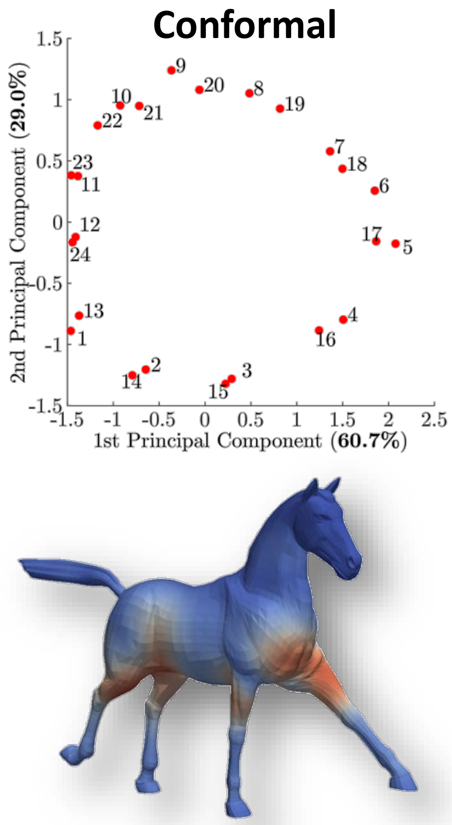


Deformation localization

Intrinsic Shape Space

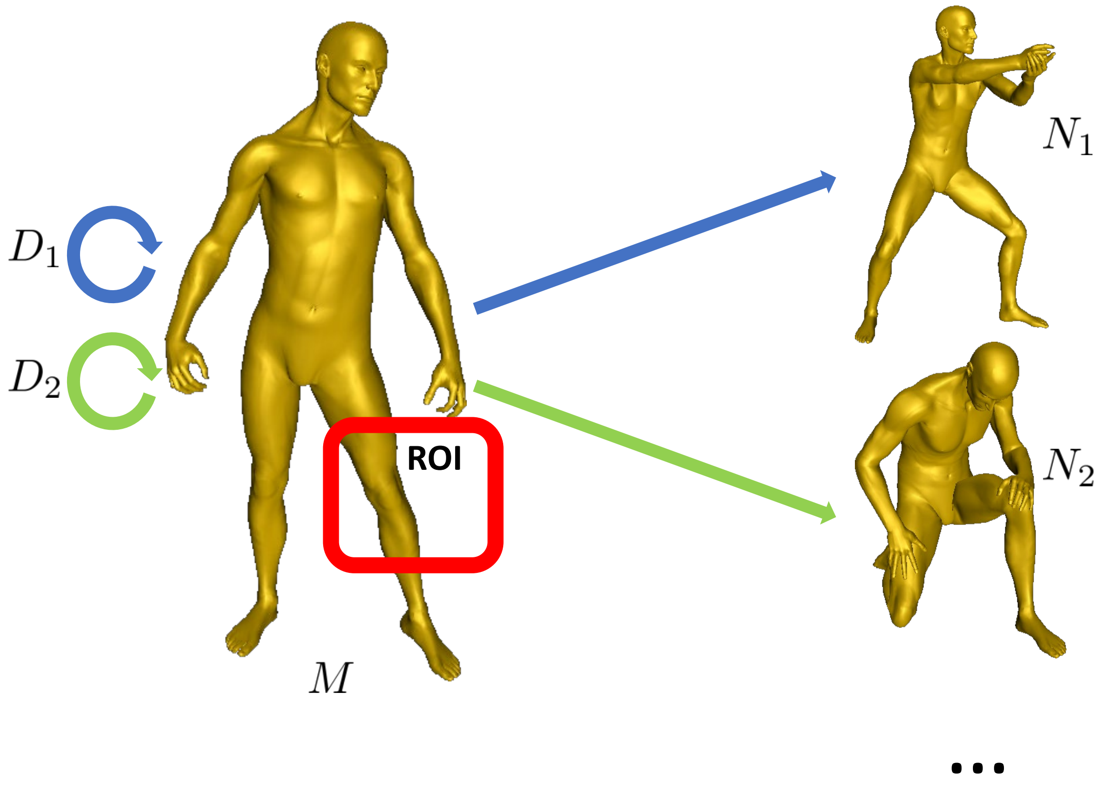






Deformation localization

Localized Comparisons

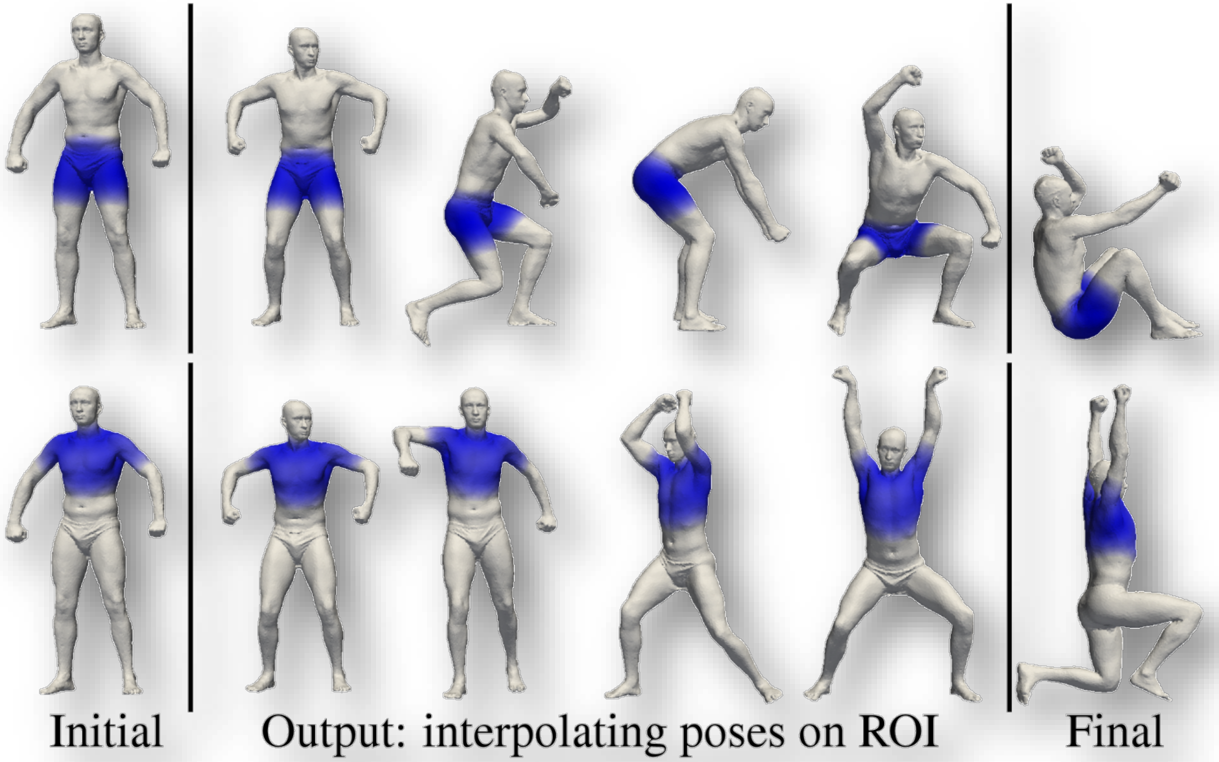


$$\rho : M \to \mathbb{R}$$

supported in RoI

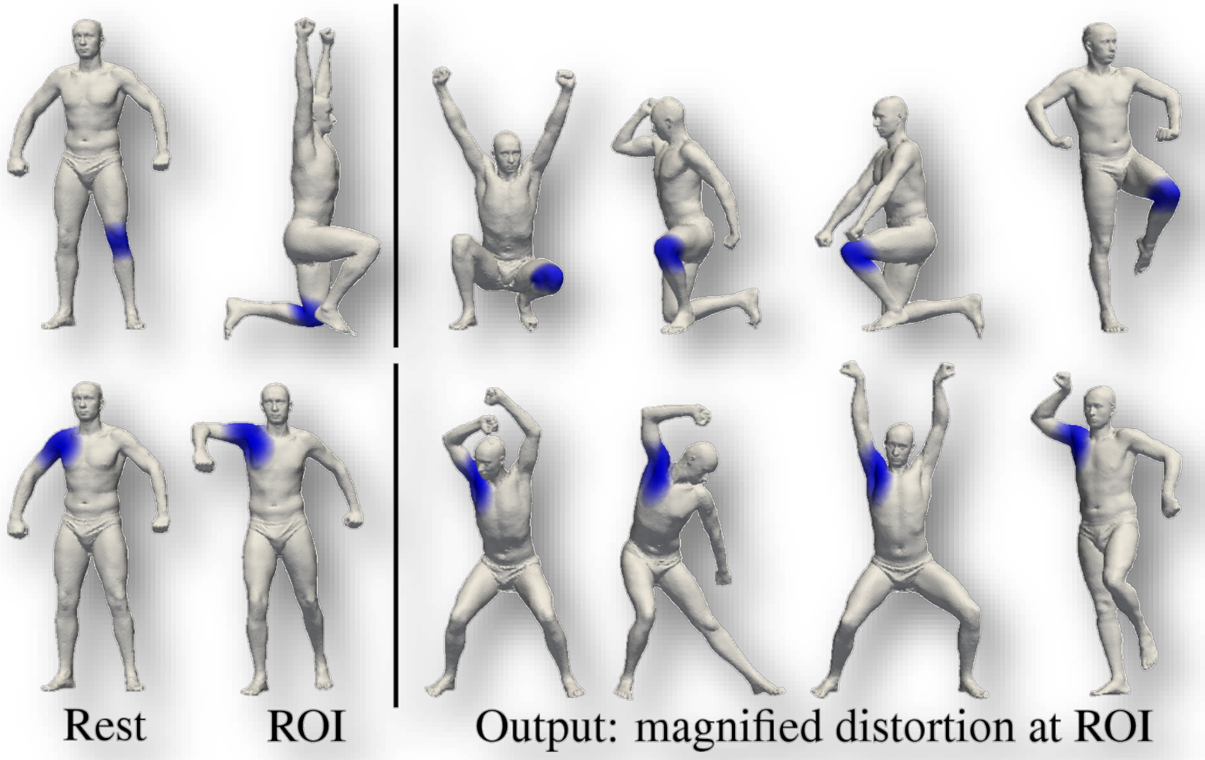
$$D_1\rho \text{ to } D_2\rho$$

Interpolation Between Poses Along RoI

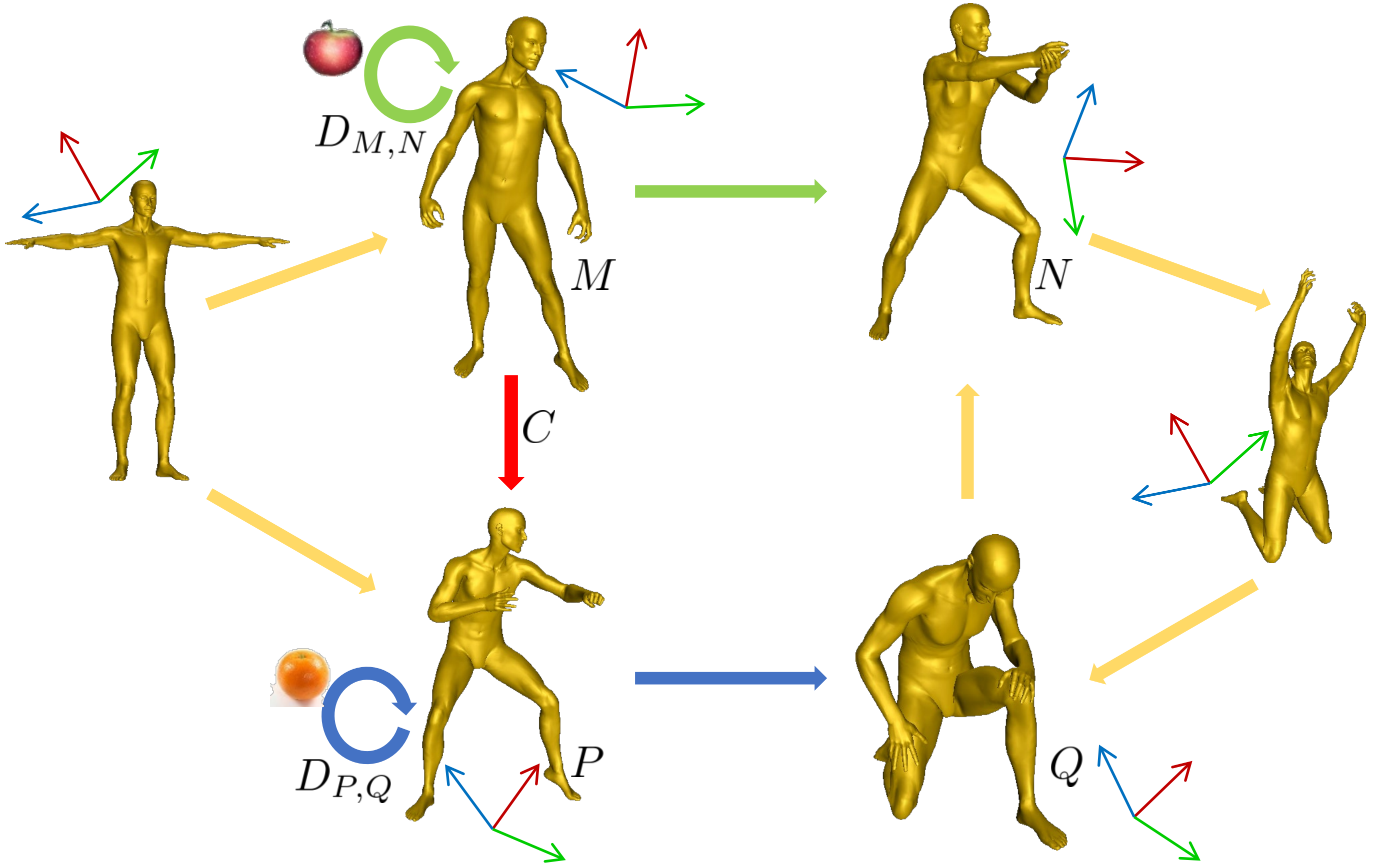


SCAPE Human Shapes

Exaggeration of Difference in RoI

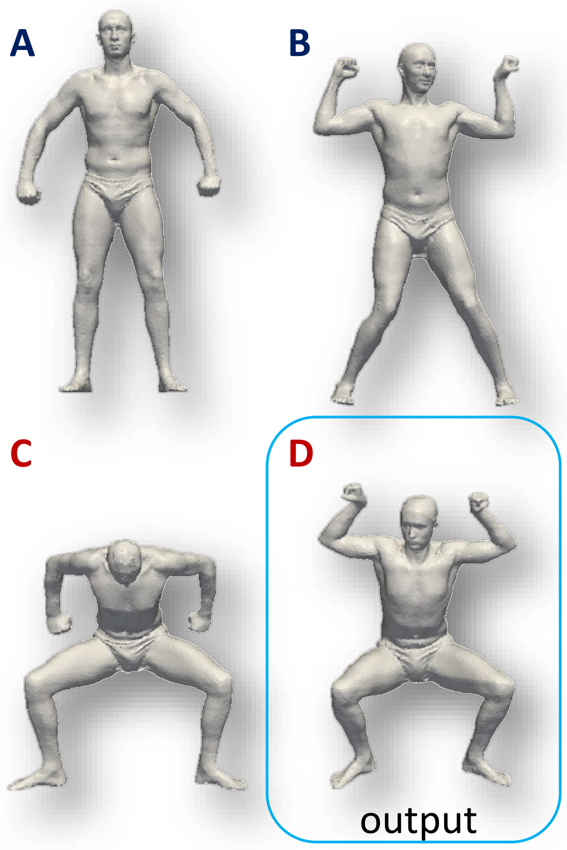


Comparing Differences II



$$D_{M,N} \sim C^{-1} D_{P,Q} C$$

Analogies: D relates to C as B relates to A



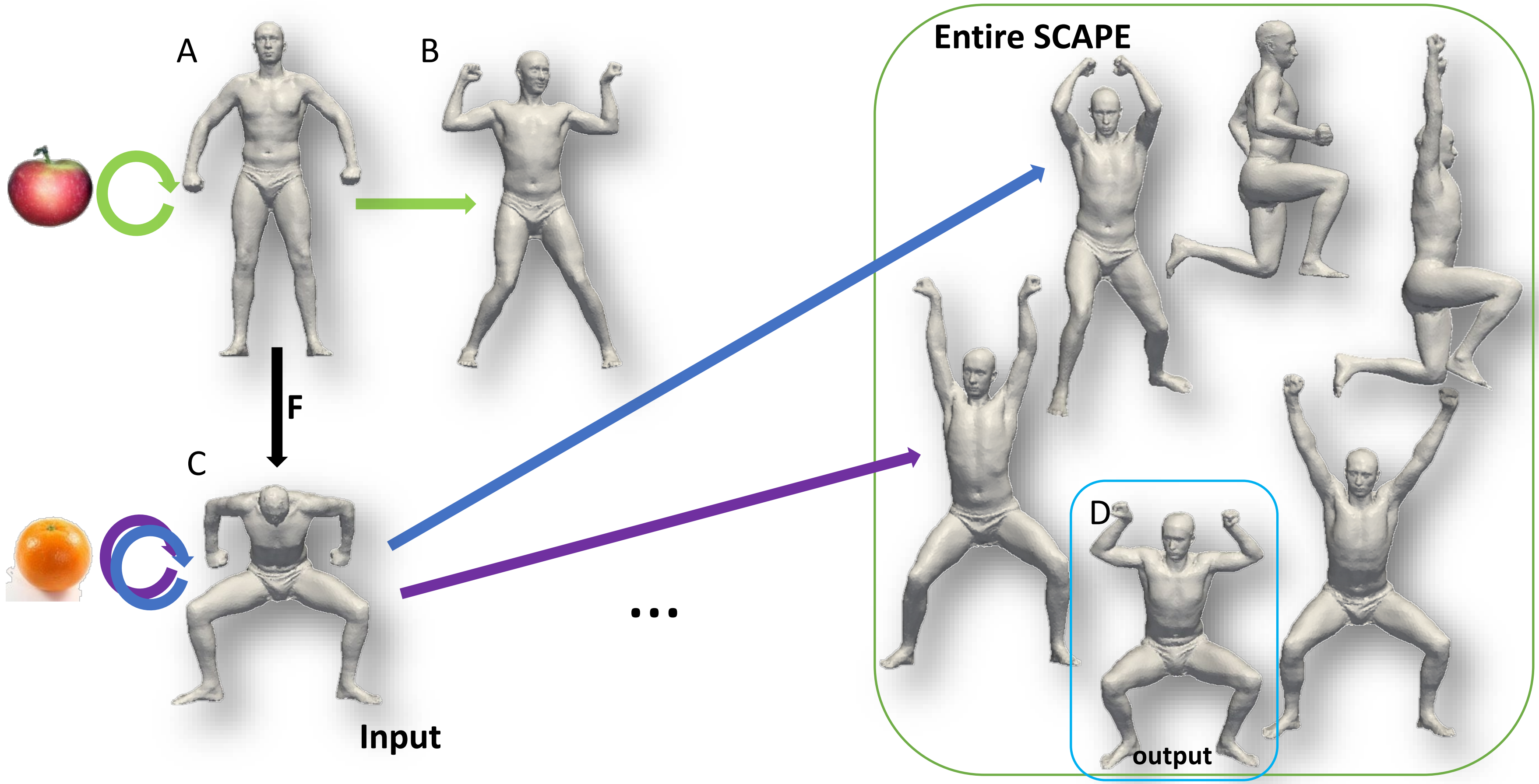
$$D = C + (B - A)$$

hands raised up

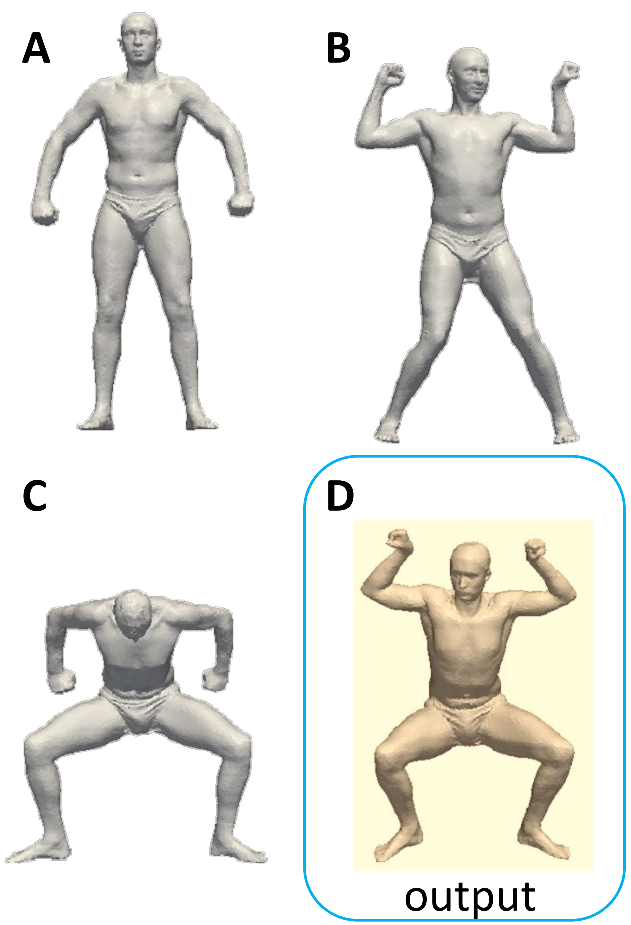
or

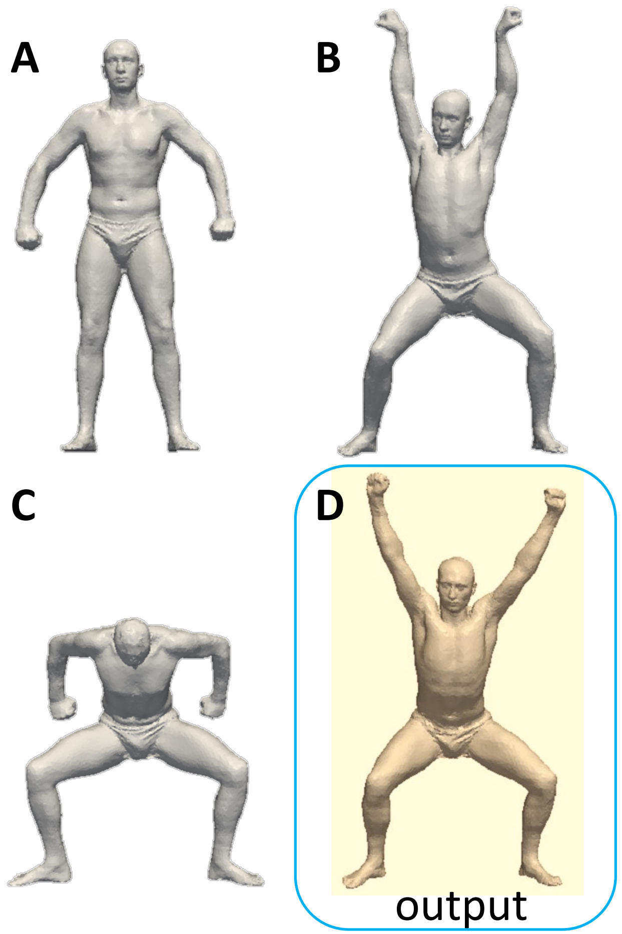
$$D = C\,B\,A^{-1}$$

Analogies: D relates to C as B relates to A

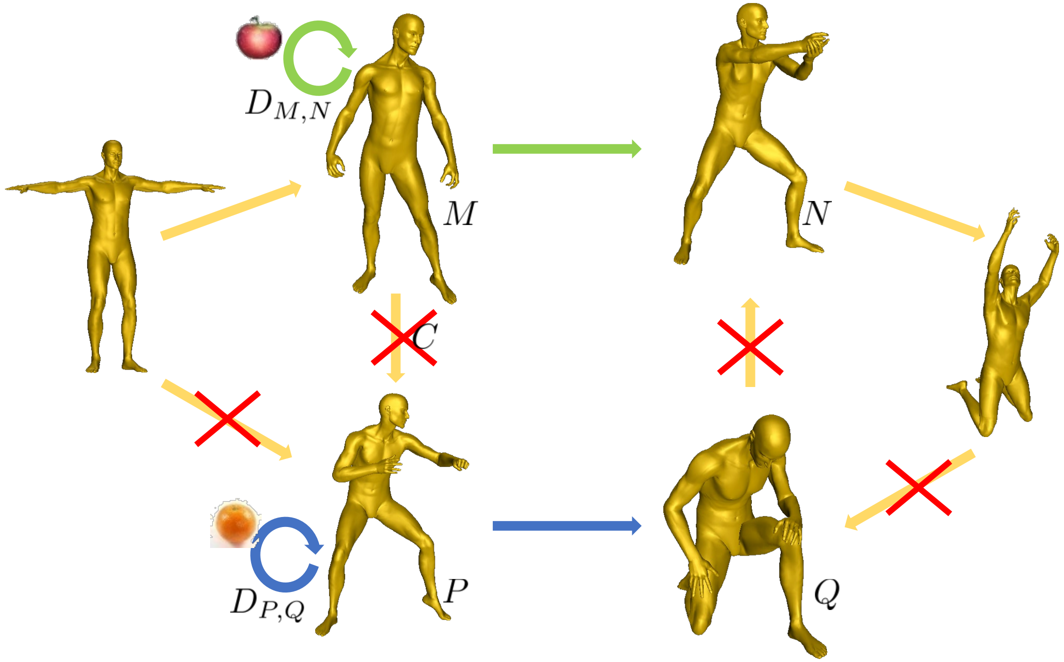


Shape Analogies





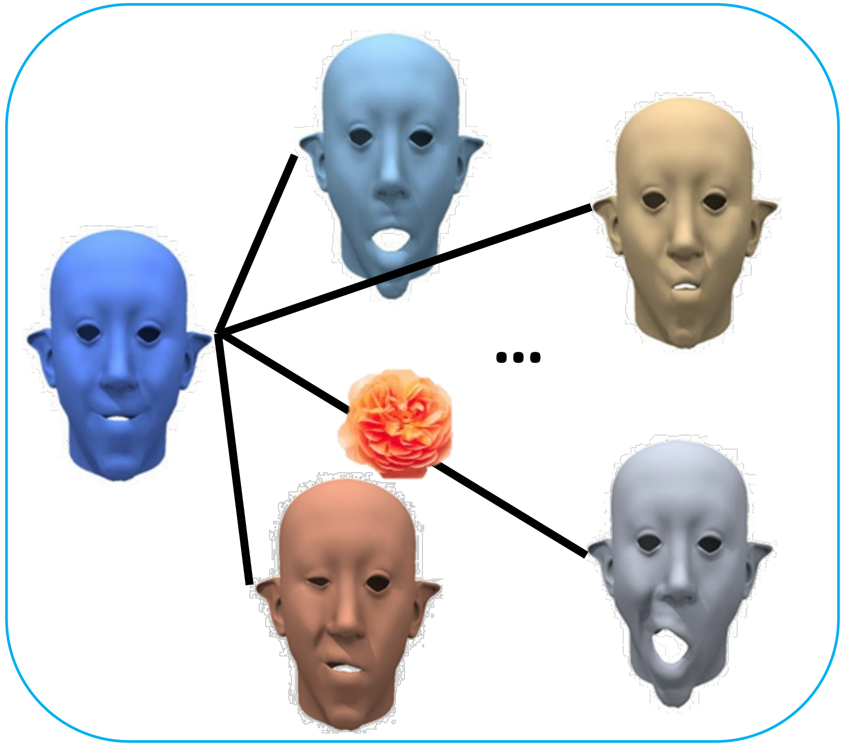
Comparing Differences III



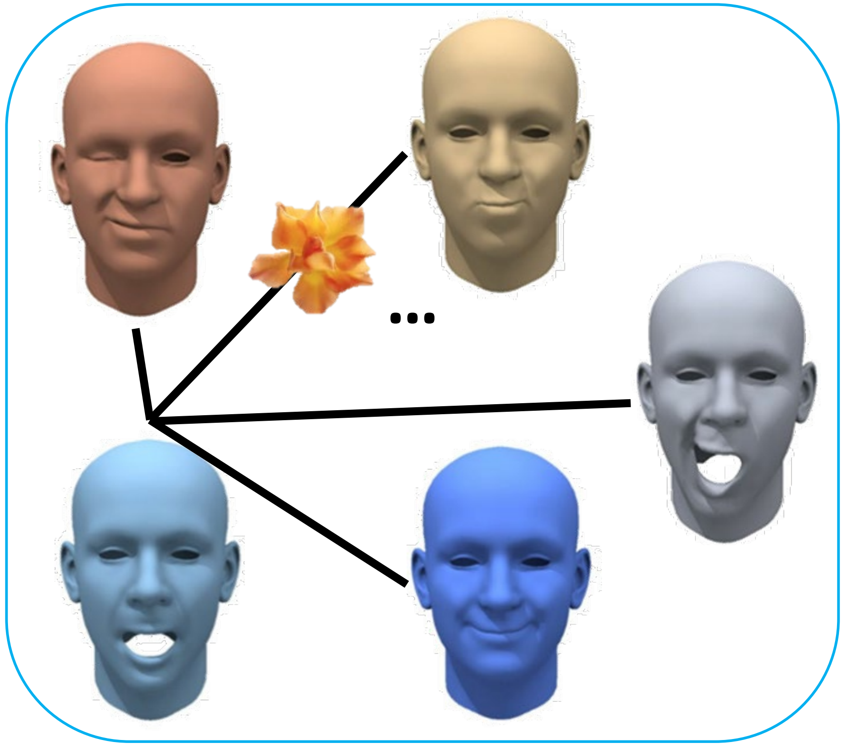
$$D_{M,N} \sim C^{-1} D_{P,Q} C$$

$$Spec(D_{M,N}) \sim Spec(D_{P,Q})$$

Aligning Disconnected Collections

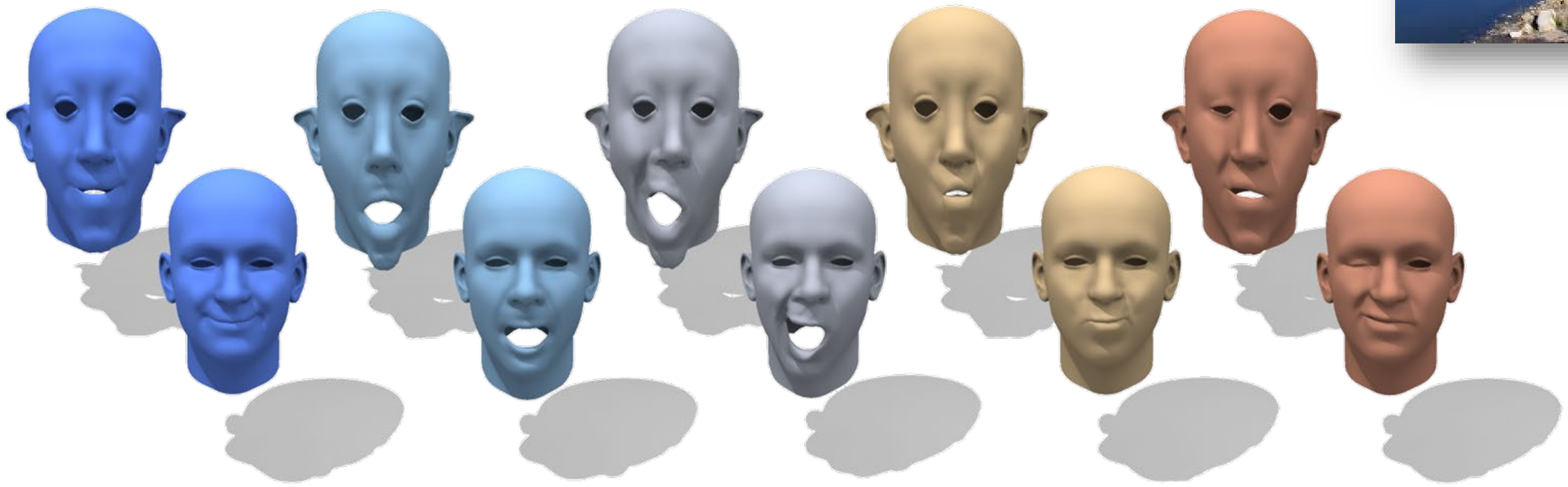


Complete graph



Complete graph

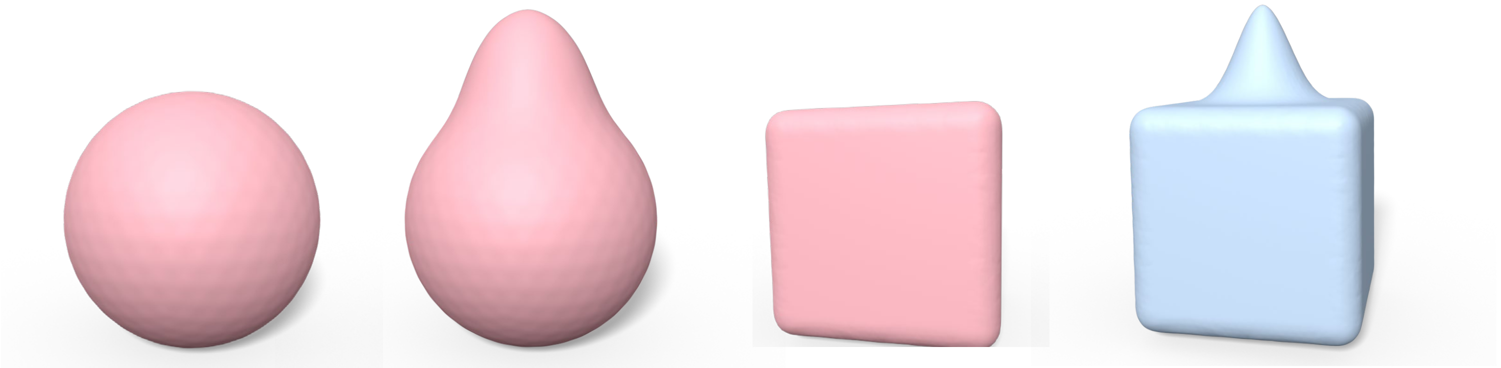
Aligning, Without "Crossing the River"



Comparing the differences is sometimes easier than comparing the originals

Shape from Differences

[E. Corman, J. Solomon, M. Ben-Chen, L.Guibas, M. Ovsjanikov; ACM ToG ‘17]
[R. Huang, P. Achlioptas, M.J. Rakotosaona, M. Ovsjanikov, L. Guibas; ICCV ‘19]



$$S_A : S_B = S_C : S_X$$

Shape Reconstruction from Differences (Full Basis)

Given the area weights, can solve for triangle areas.

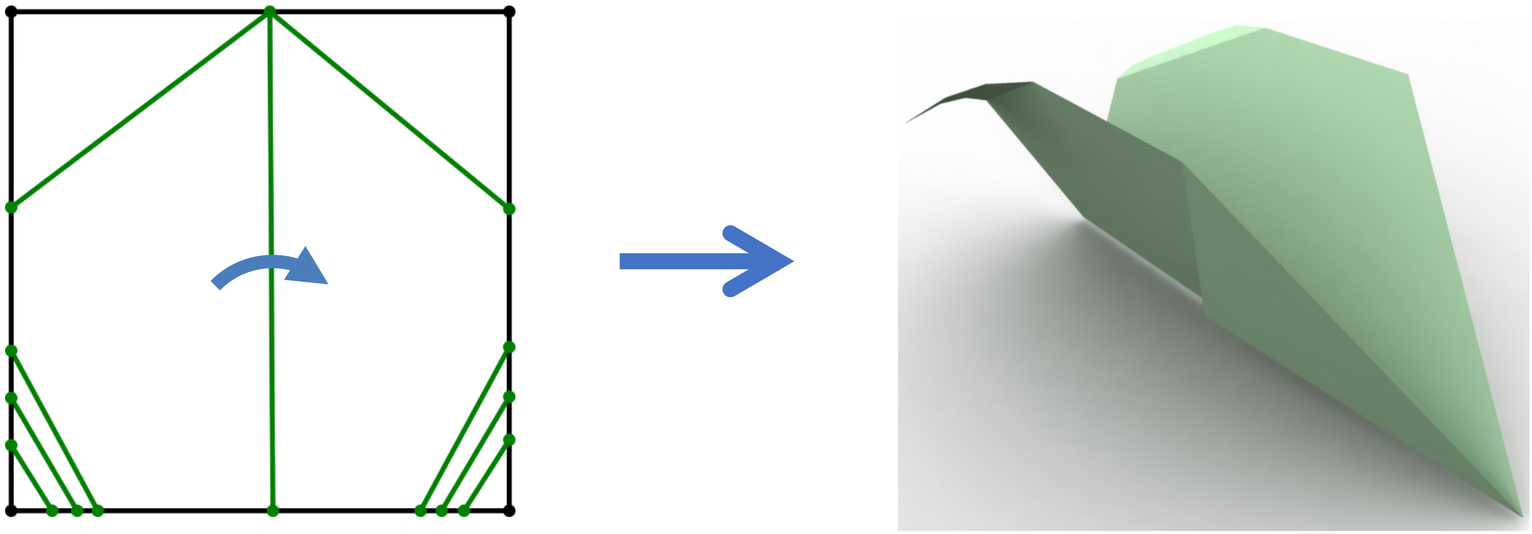
[Slide ack: J. Solomon]

Area-based shape difference → Area weights → Triangle areas

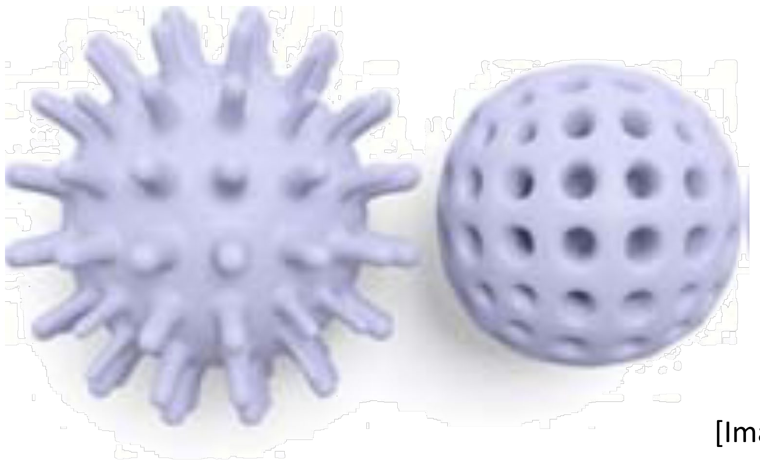
Given triangle areas and conformal inner products, can solve for squared edge lengths.

Area-based shape difference → Area weights → Triangle areas → Squared edge lengths

How to Encode Extrinsic Information?



[Image: J. Solomon]



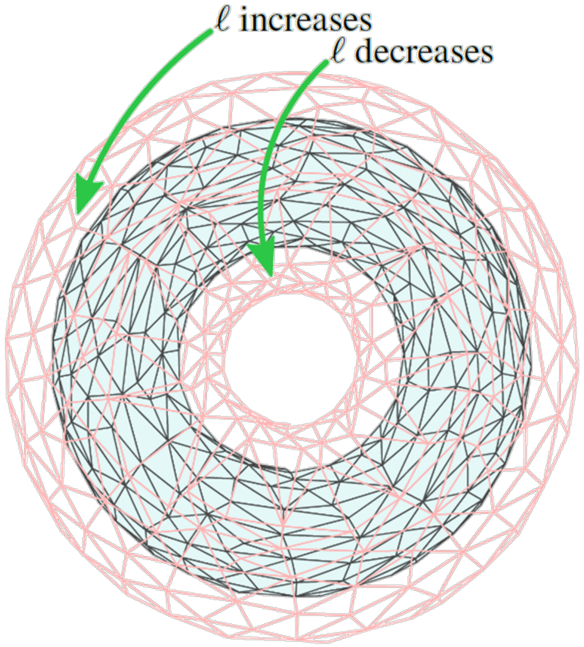
[Image: K. Crane]

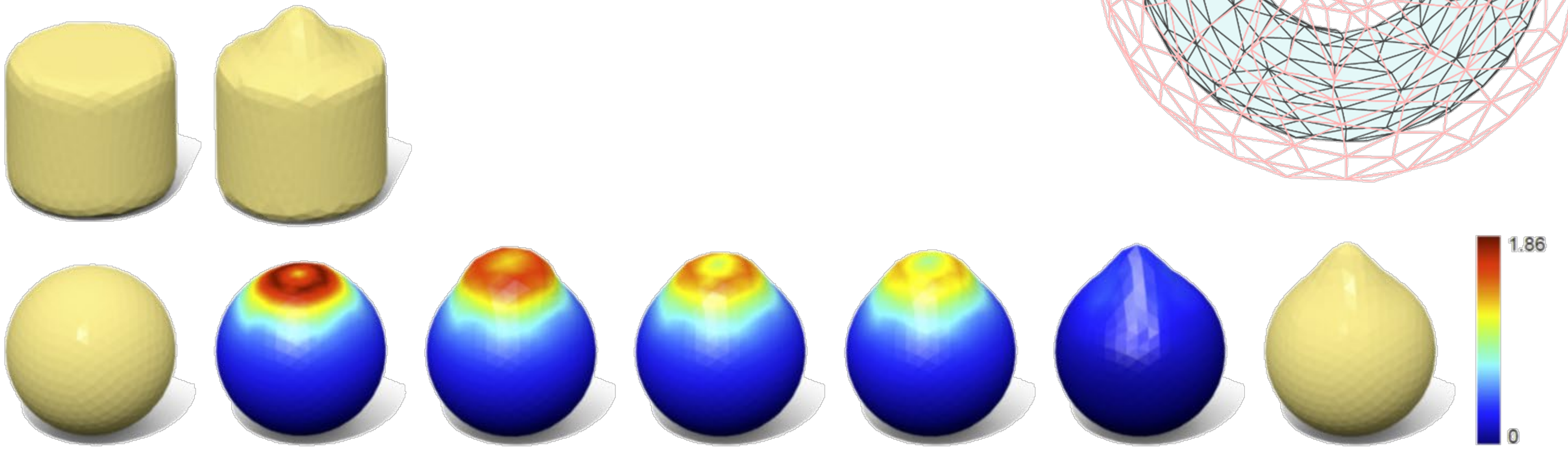
Extrinsic Shape Differences, Version 1

By adding intrinsic differences of an offset surface, we capture extrinsic distortions of the original surface!

Full recovery is provably possible

In practice, challenging optimization problem, especially when the functional basis has been truncated

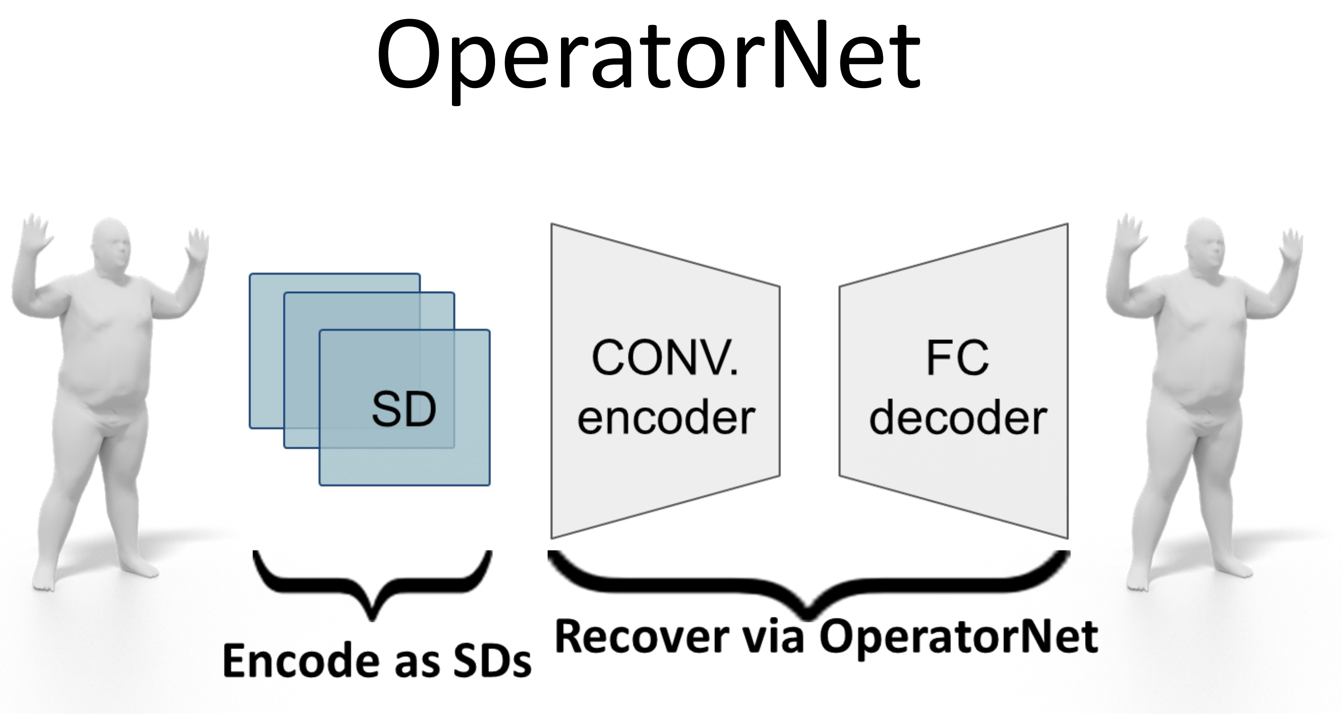




Extrinsic Shape Differences, Version 2

Decode 3D shapes via deep nets directly from shape difference operators

- **Advantages:**
  - **Compact encodings** (small matrices of size)
  - **Natural algebraic** manipulation
  - **Invariant** to rigid transformation
  - Adapted to **convolutional** neural networks
- Applications: shape interpolation, style transfer, up-sampling



Intrinsic and Extrinsic

Let us assume that the shapes are in vertex to vertex correspondence. Then we define (1) the intrinsic area-based and conformal differences as before, and (2) an extrinsic shape difference, as follows:

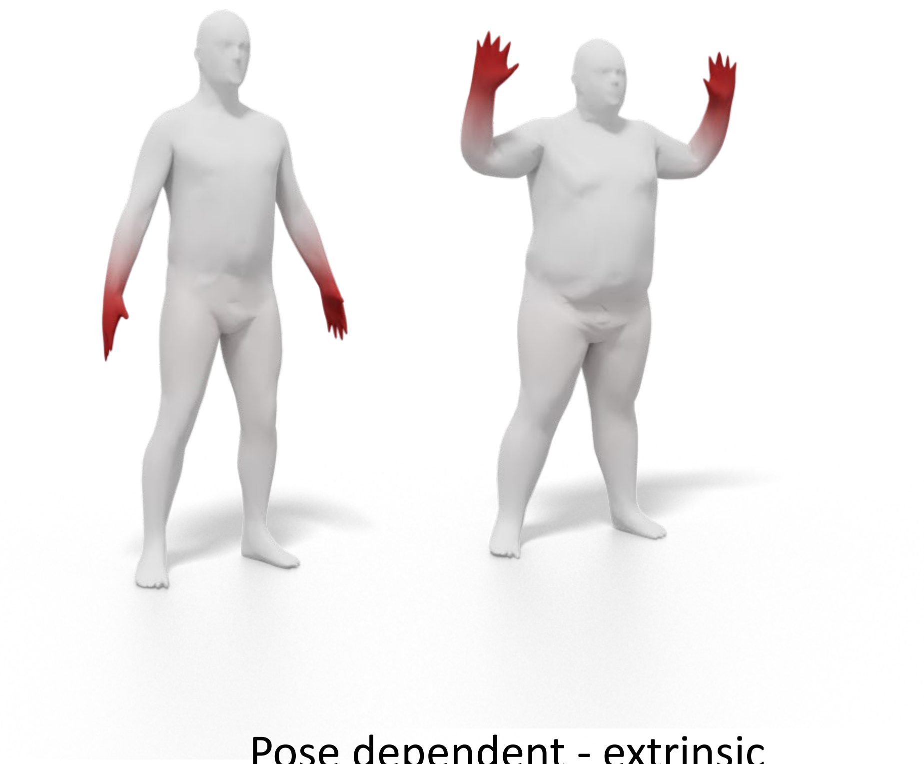
$$
\begin{aligned}
V &= A_{\text{source}}^{-1} F^T A_{\text{target}} F \\
R &= (L_{\text{source}} A_{\text{source}})^{-1} F^T L_{\text{target}} A_{\text{target}} F \\
E &= (M_{\text{source}} A_{\text{source}})^{-1} F^T M_{\text{target}} A_{\text{target}} F
\end{aligned}
$$

Here *F* is the truncated basis functional map, *A* is the area-weights mass matrix, and *L* is the standard Laplacian, and *M* is an "extrinsic" Laplacian.

$$M_{i,j} = -\|v_i - v_j\|^2 \text{ if } i \neq j, \text{ or } \textstyle\sum_{k,k\neq i} M_{i,k} \text{ if } i = j.$$

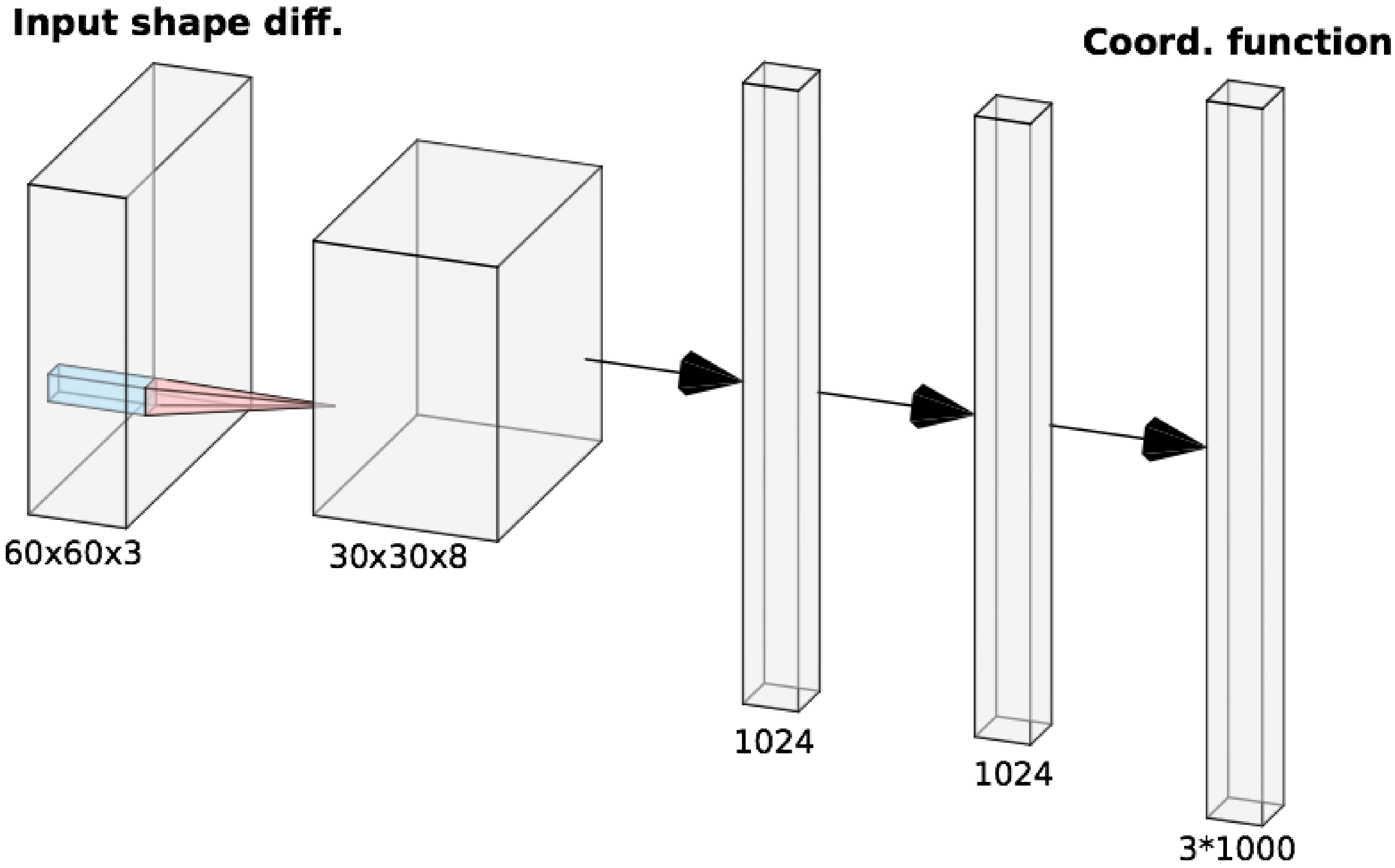
Example

Pose oblivious -- intrinsic

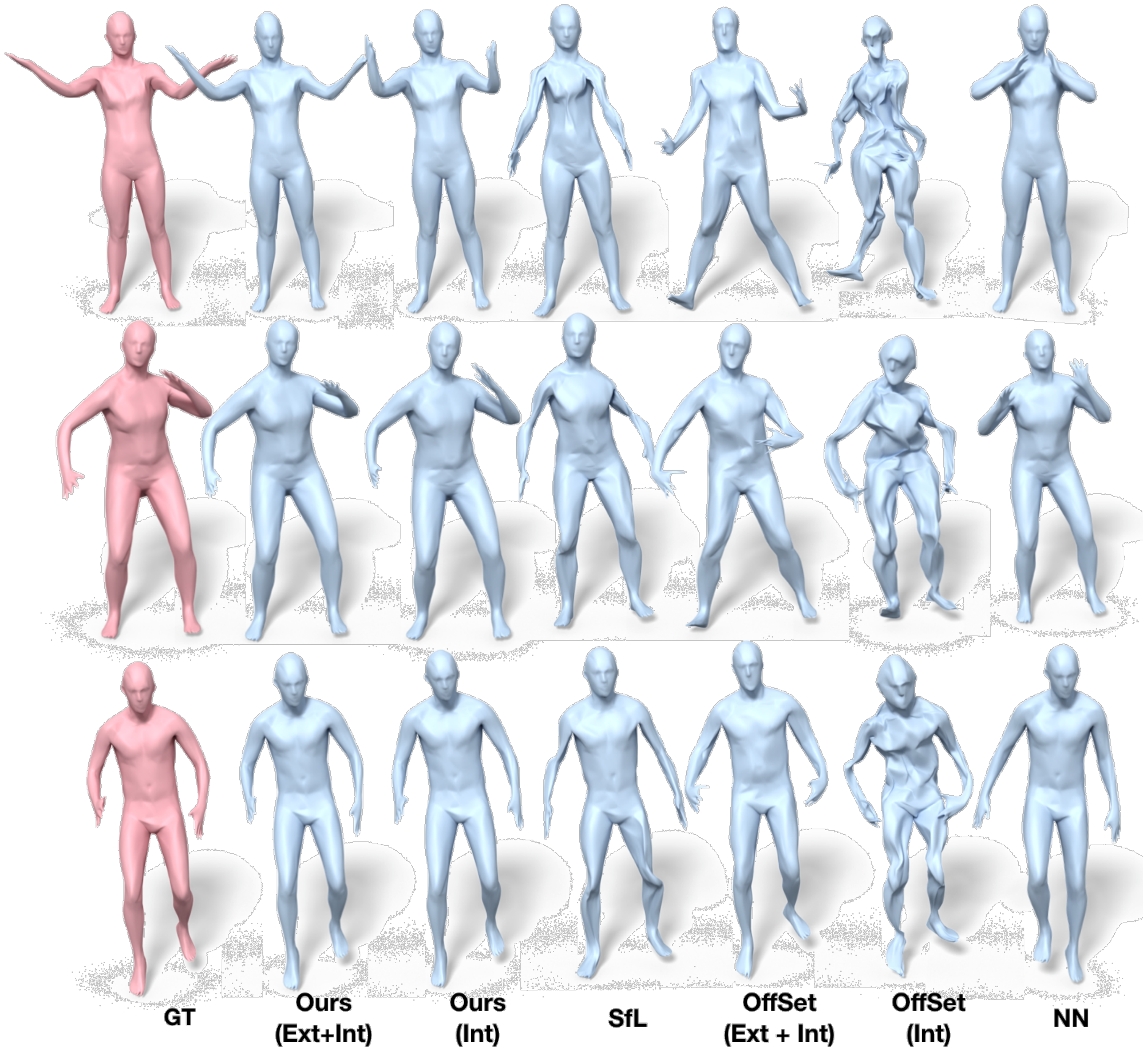


Pose dependent - extrinsic

OperatorNet Reconstruction



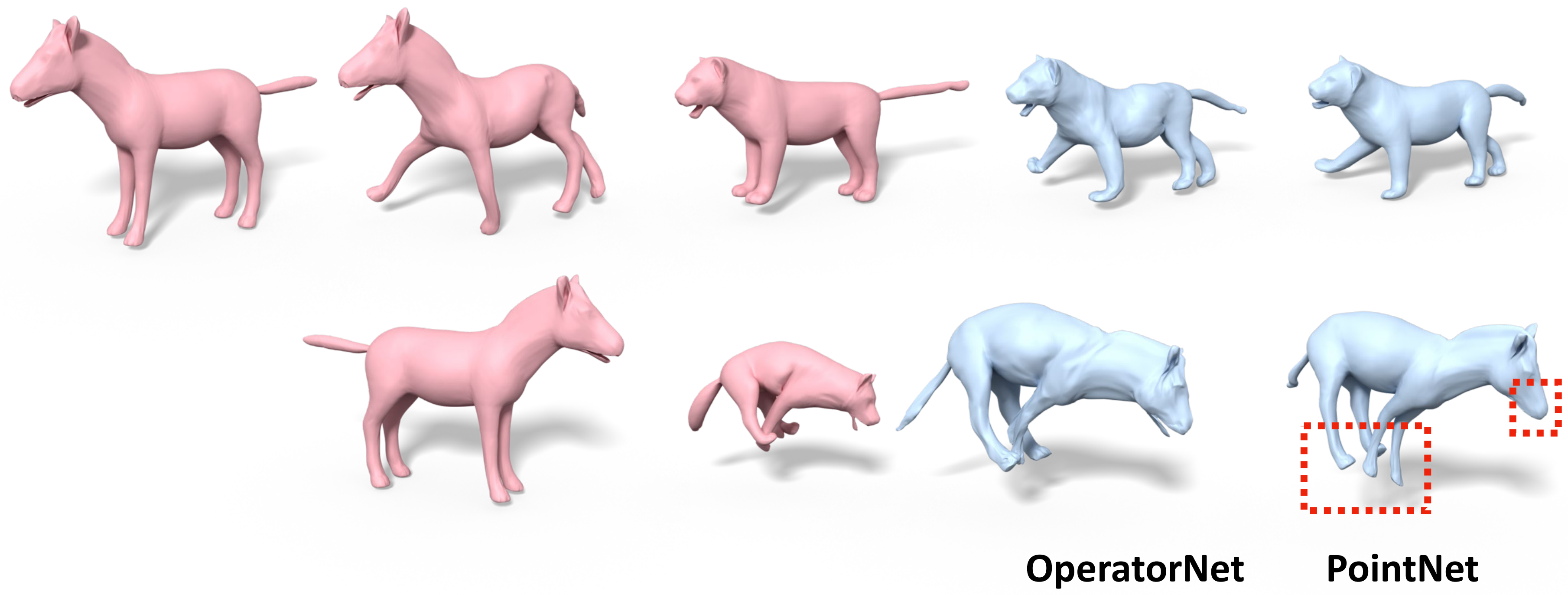
Reconstruction Comparisons



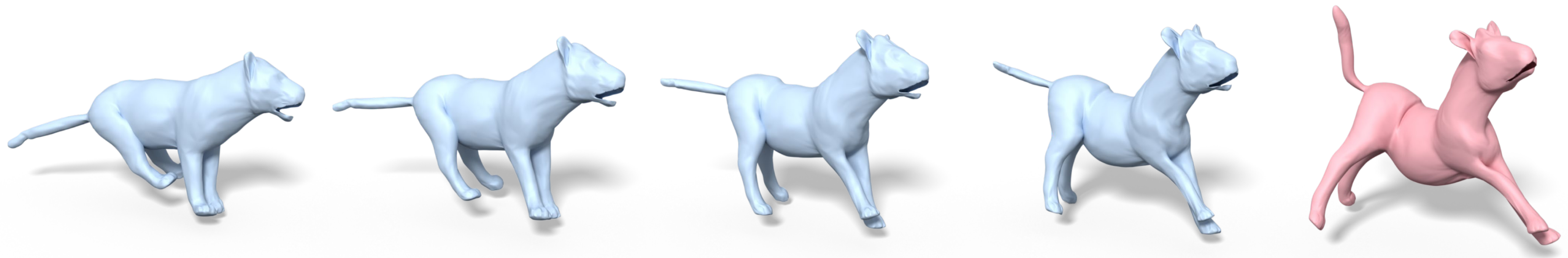
More Examples

Shape Analogies

$$S_A \quad : \quad S_B \quad = \quad S_C \quad : \quad S_X$$

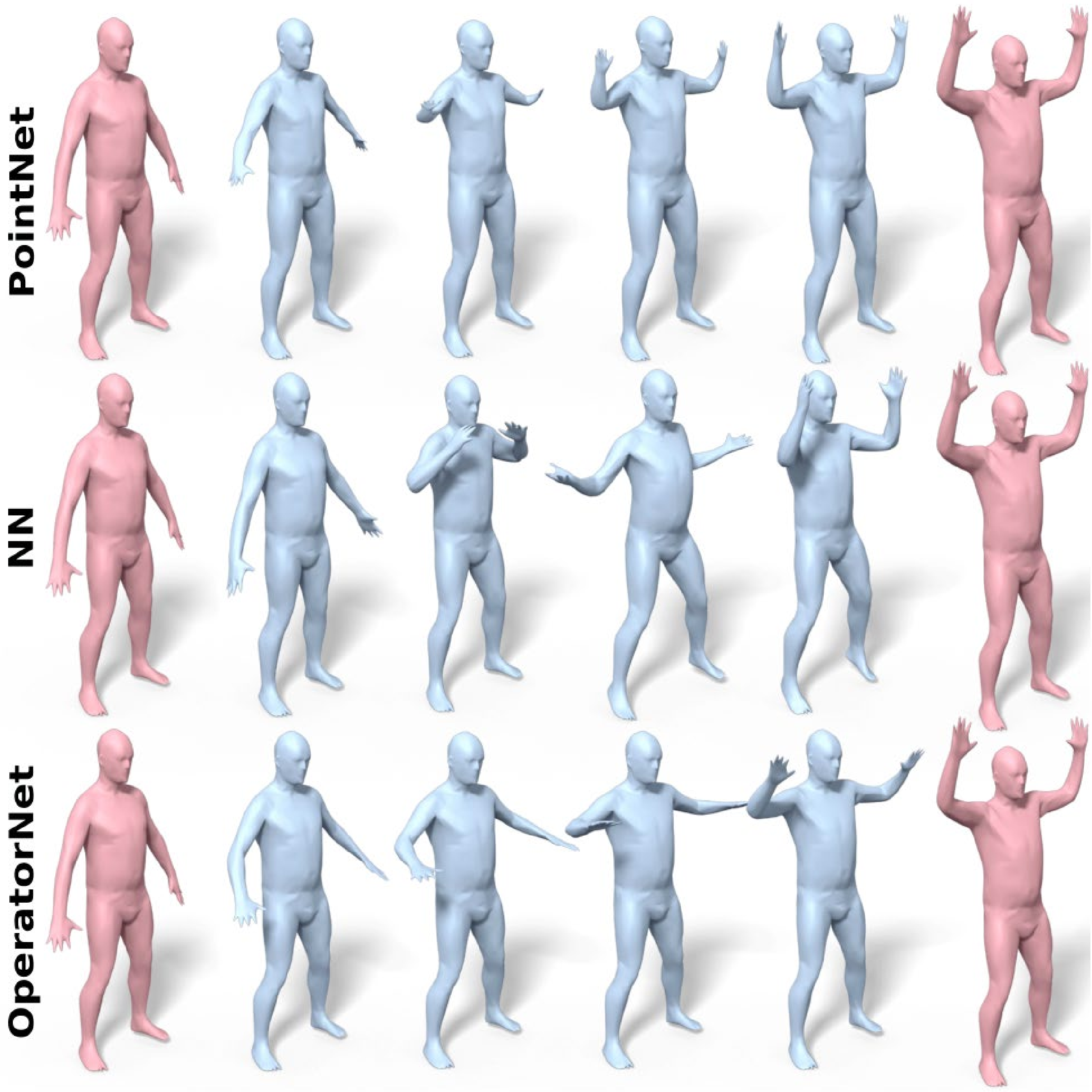


Interpolation

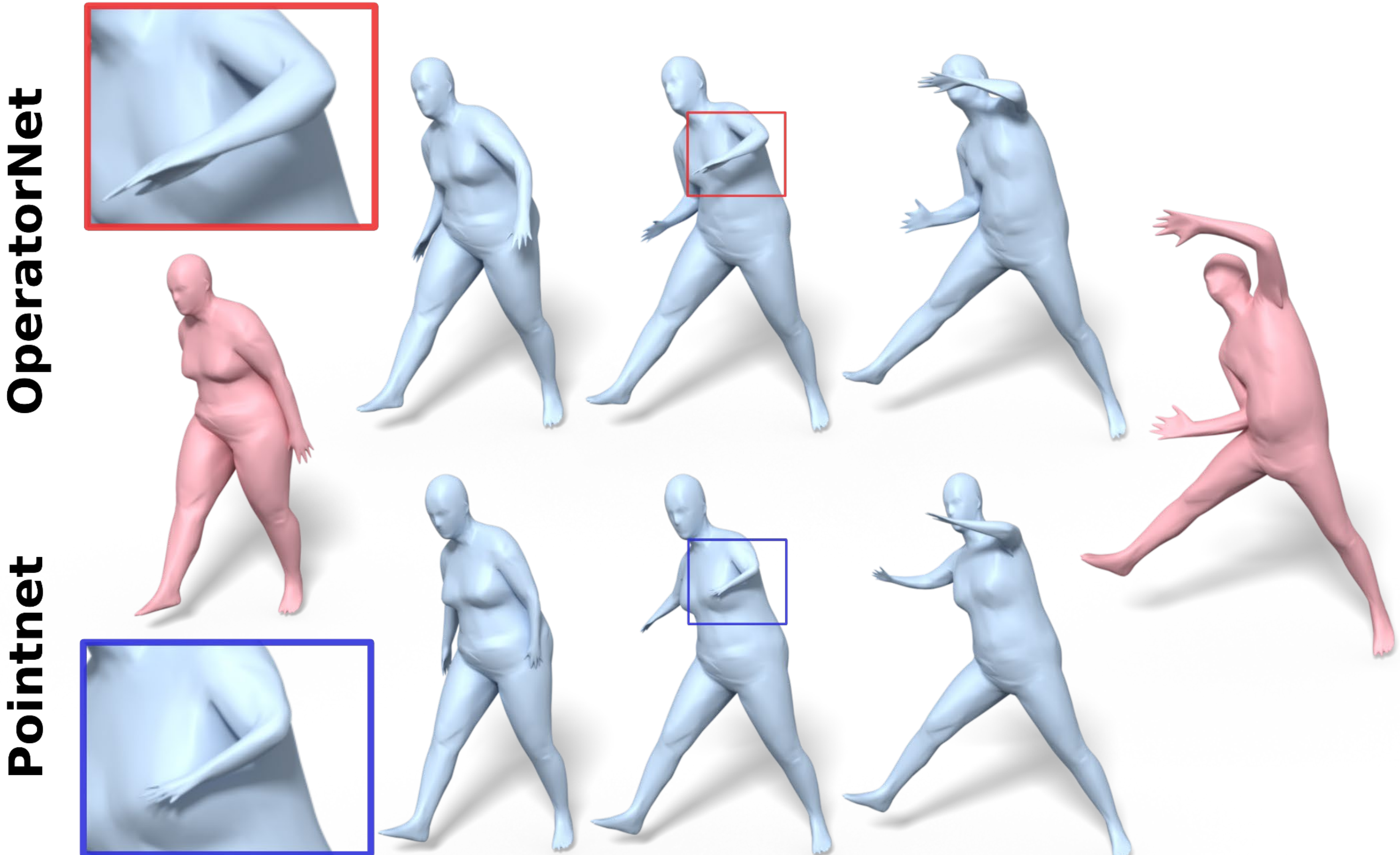


Interpolation between a dog and a horse

Shape Interpolation



Interpolation Detail



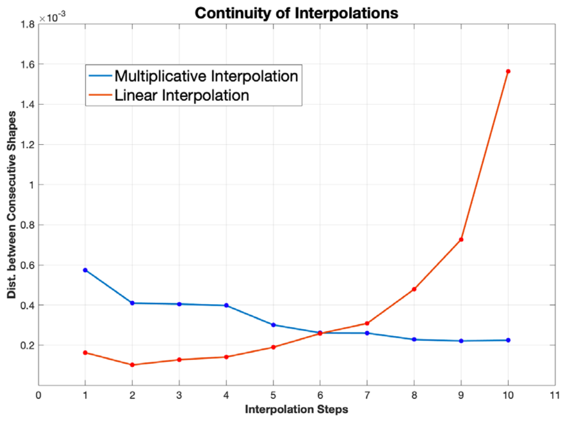
Additive or Multiplicative?

Mult. Interp.

Linear Interp.

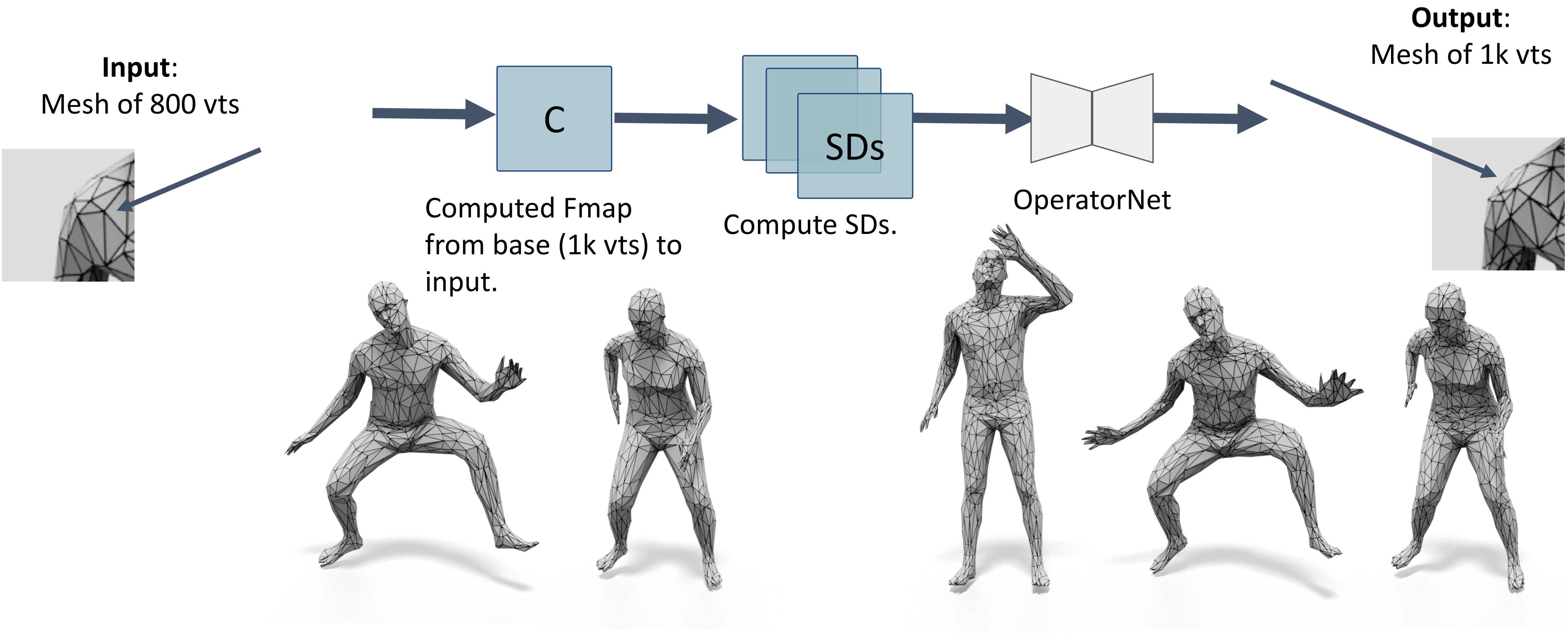
$$C \leftarrow A^t B^{1-t}$$

$$C \leftarrow (1-t)A + tB$$



Different Triangulations

Recovery of shapes in different triangulations



Shape Differences in Language

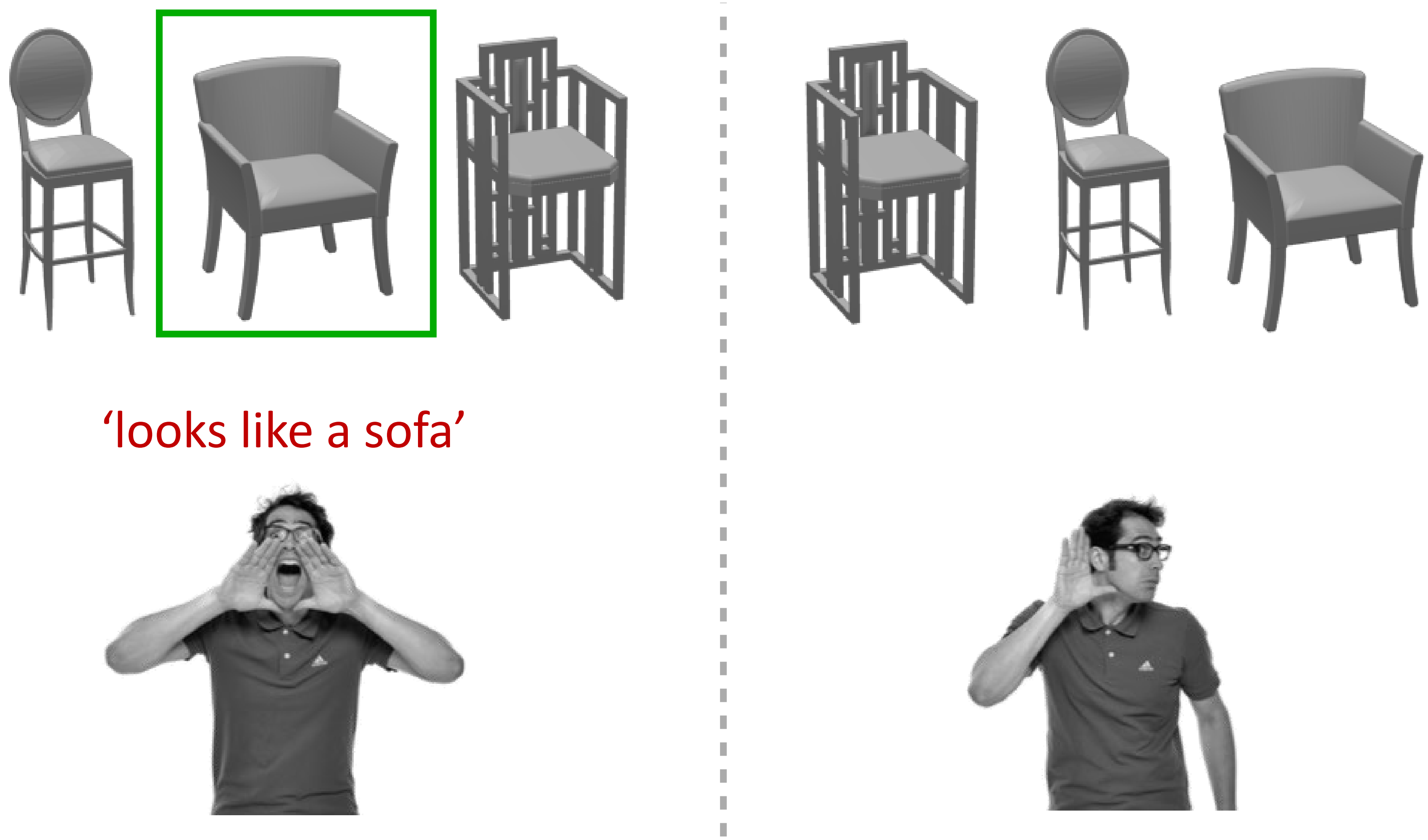
[P. Achlioptas, J. Fan, R. Hawkins, N. Goodman, L. Guibas; ICCV ‘19]

Differences in Geometry Expressed in Language

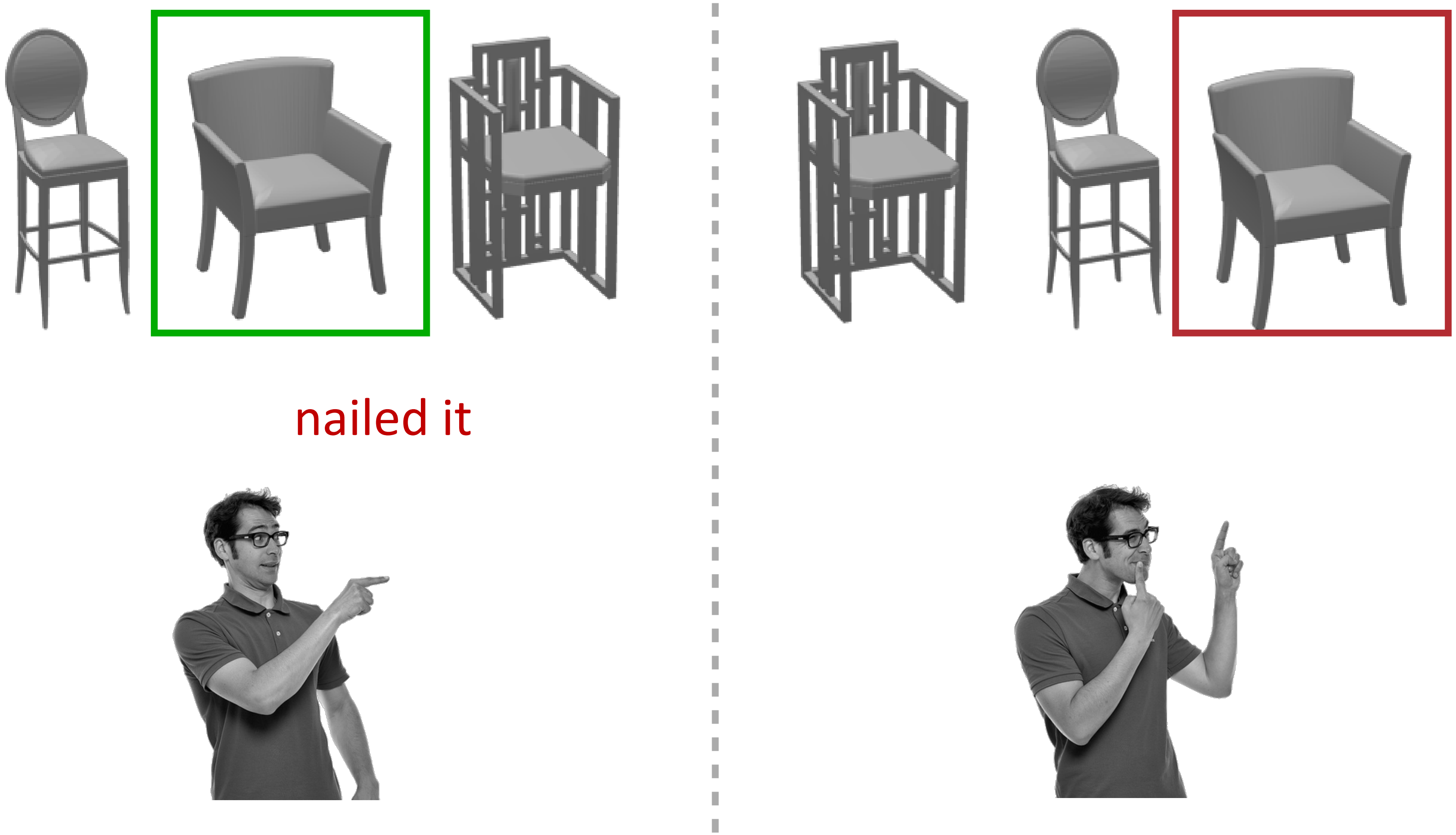
Target Object

“Gaps in the back”

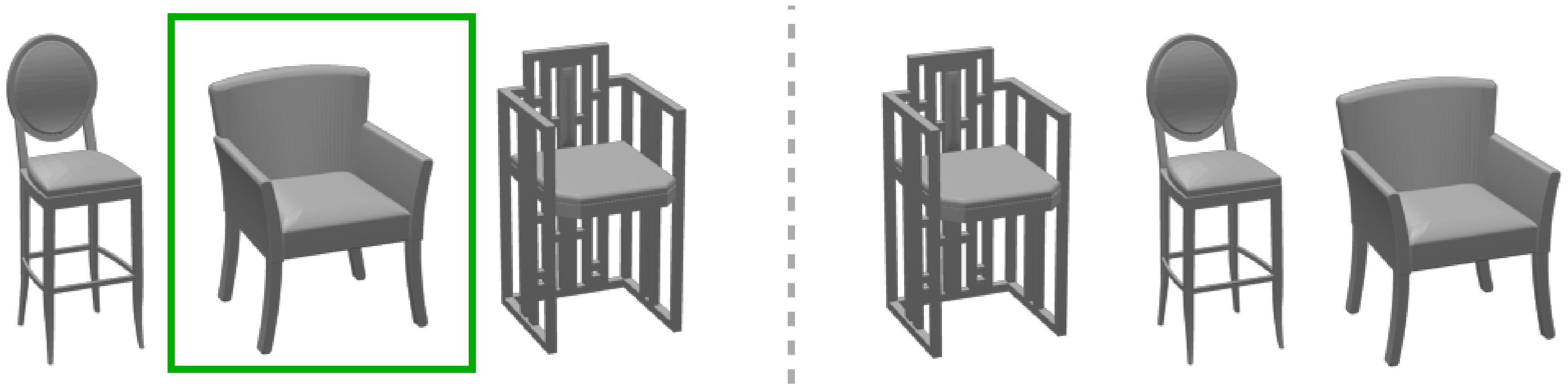
A Reference Language Game



A Reference Language Game



Considerations



scrambled order
same view-point
no texture

no location
no orientation
yes geometry

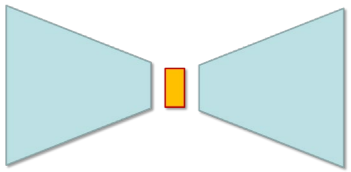
Communication Context

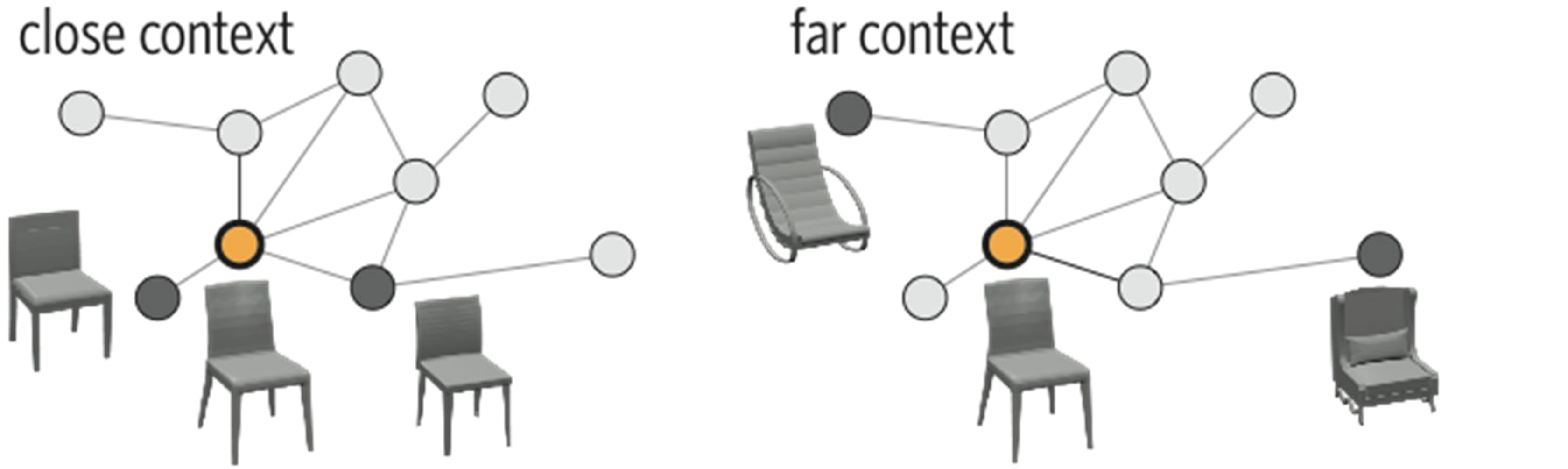
impossible

trivial

Shape Similarity

- Point cloud auto-encoder latent space
- Popular chairs via *max-degree*





“Chairs in Context” Corpus / Data Set

- 4,054 distinct contexts covering 4,511 chairs
- 78,789 utterances by 2,124 unique AMT participants

- “close” vs. “far” human accuracy: 94.2% vs. 97.2%
- 85% of utterances contain a part-related words
- “close” utterances rich in adjectives, comparatives

Utterance Examples

Target

"it has wheels"

"has vertical lines on the back"

"rectangle back with straight legs"

Neural Listener w. Attention Model

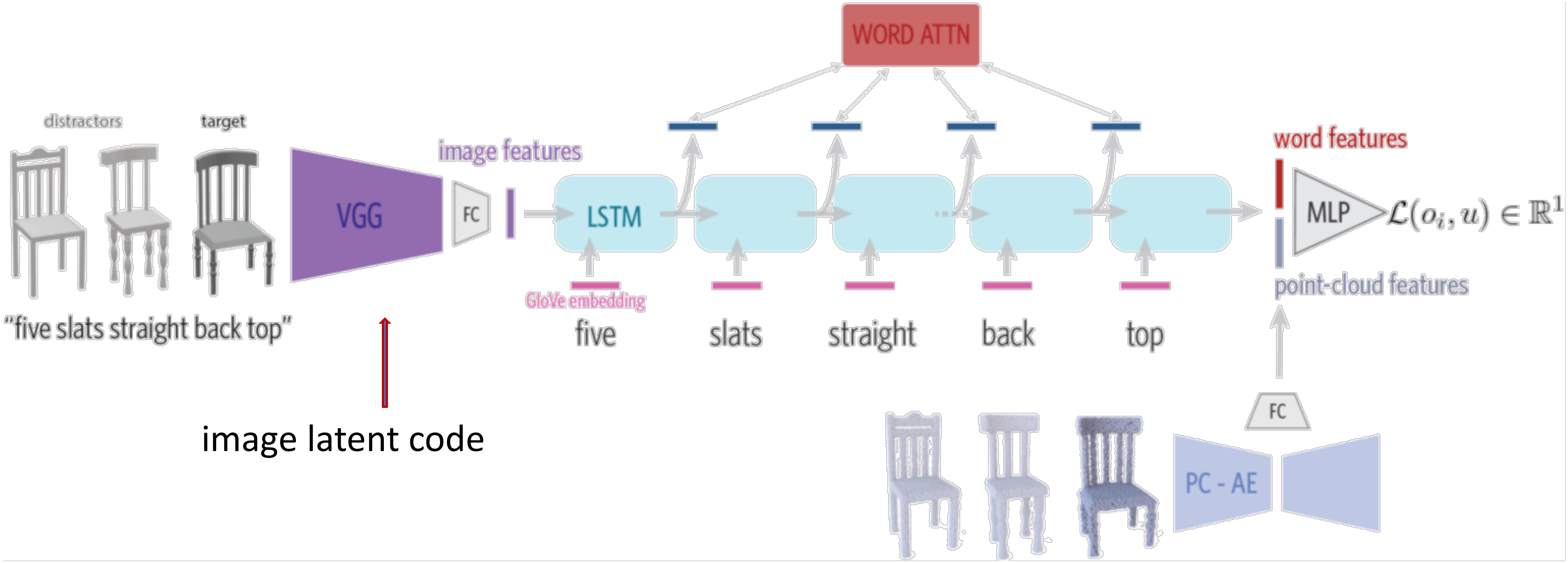
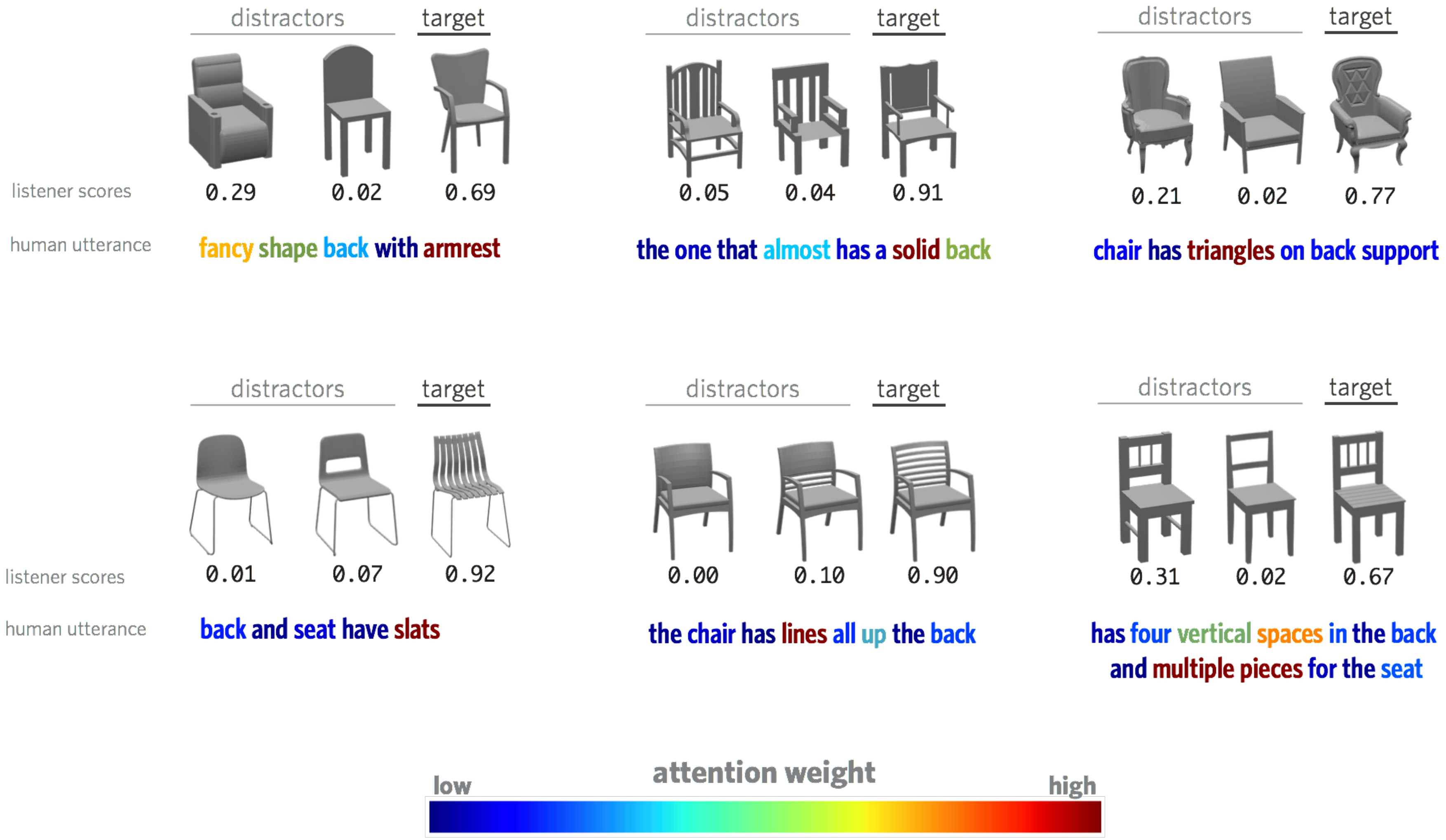


image latent code

point cloud latent code

Attentive Neural Listening

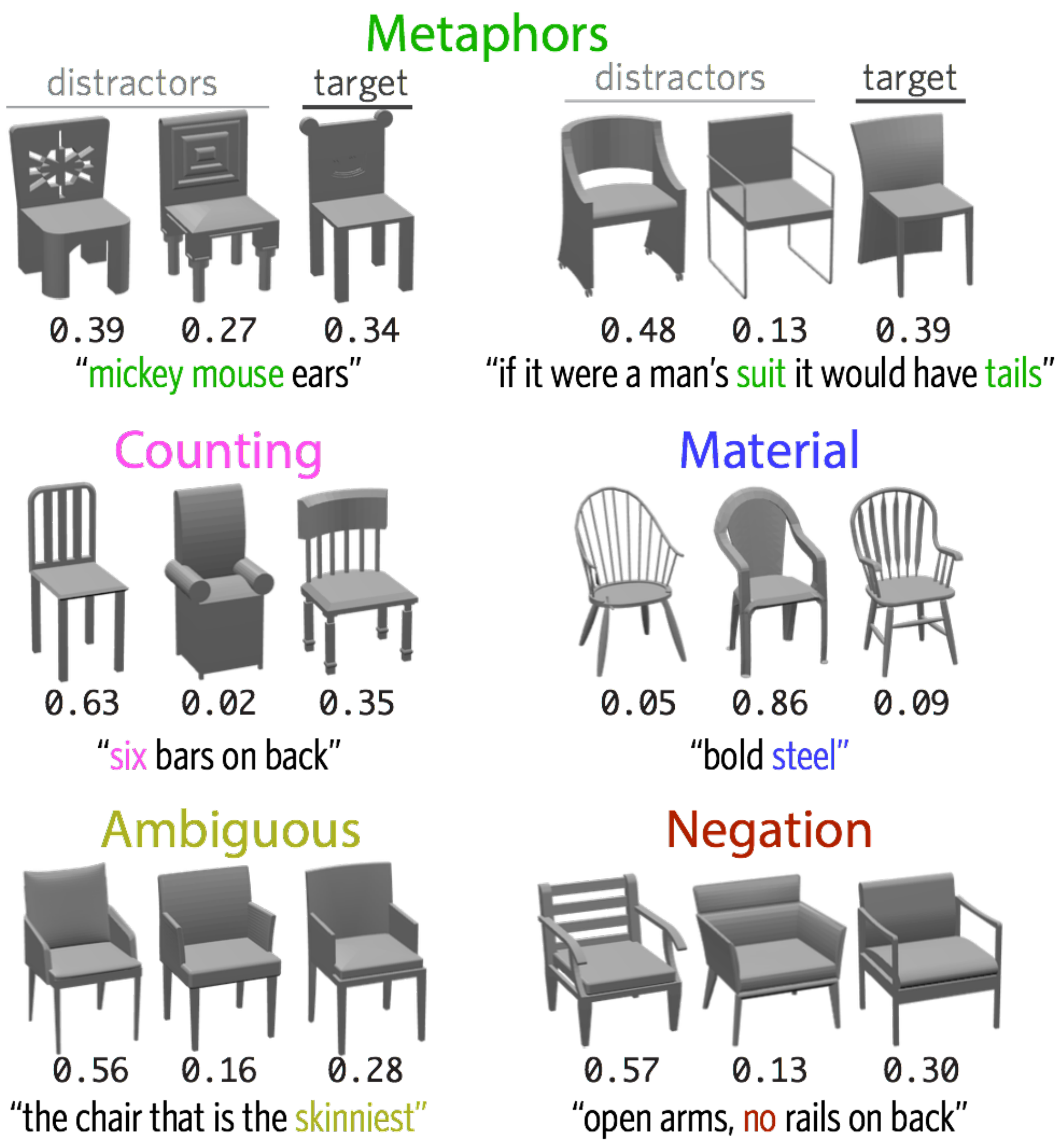


Performance

Architecture	Subpopulations			
	Overall	**Close**	**Far**	**Sup-Comp**
Together-with-Context	75.9 ± 0.5%	67.4 ± 1.0%	83.8 ± 0.6%	74.4 ± 1.3%
Separate-with-Context	79.4 ± 0.8%	**70.1** ± 1.3%	**88.1** ± 0.6%	75.2 ± 2.1%
Separate (proposed)	**79.6** ± 0.8%	69.9 ± 1.3%	**88.1** ± 0.4%	**76.0** ± 1.6%

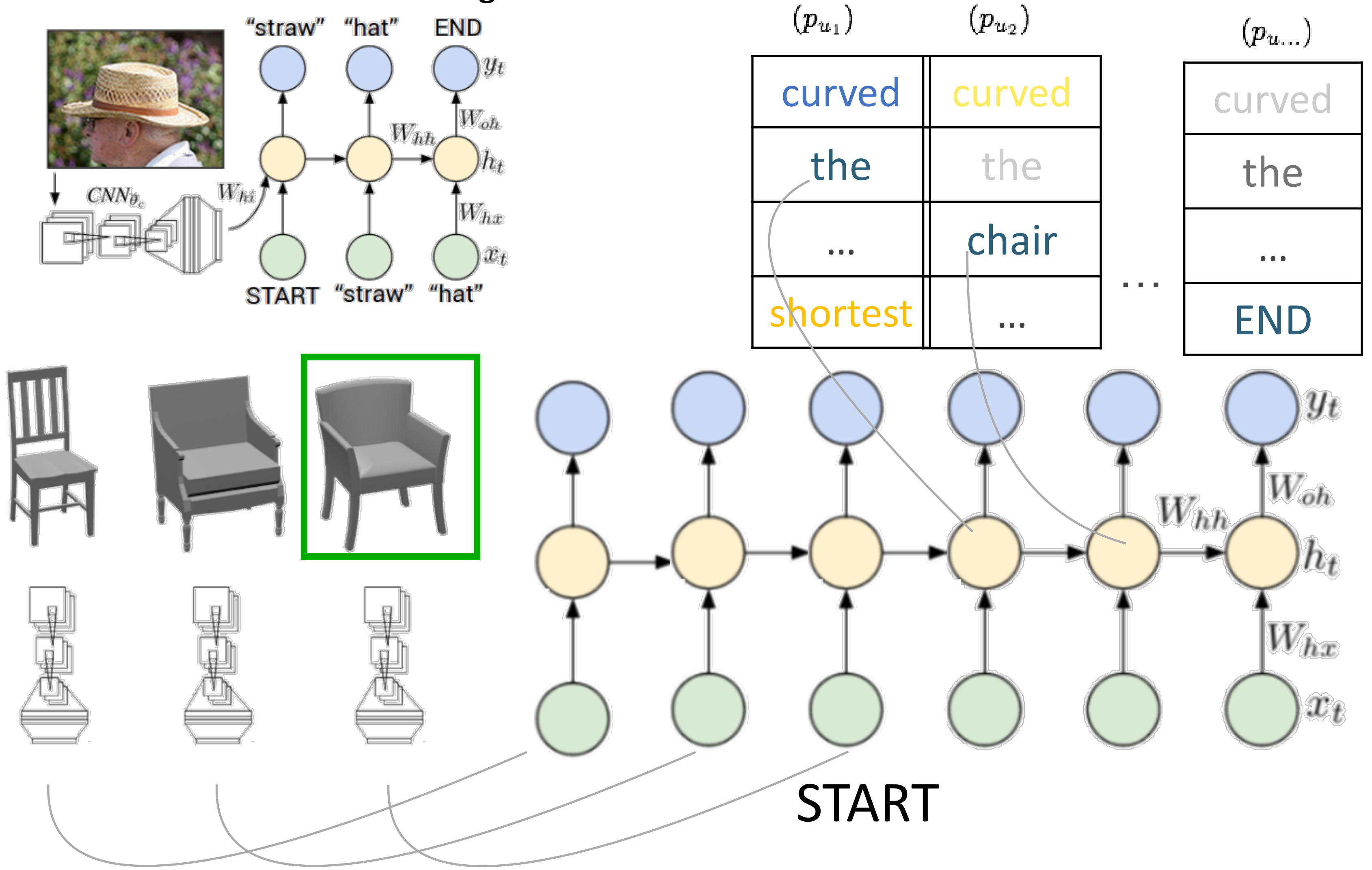
"tall**est**"

Listener Failures

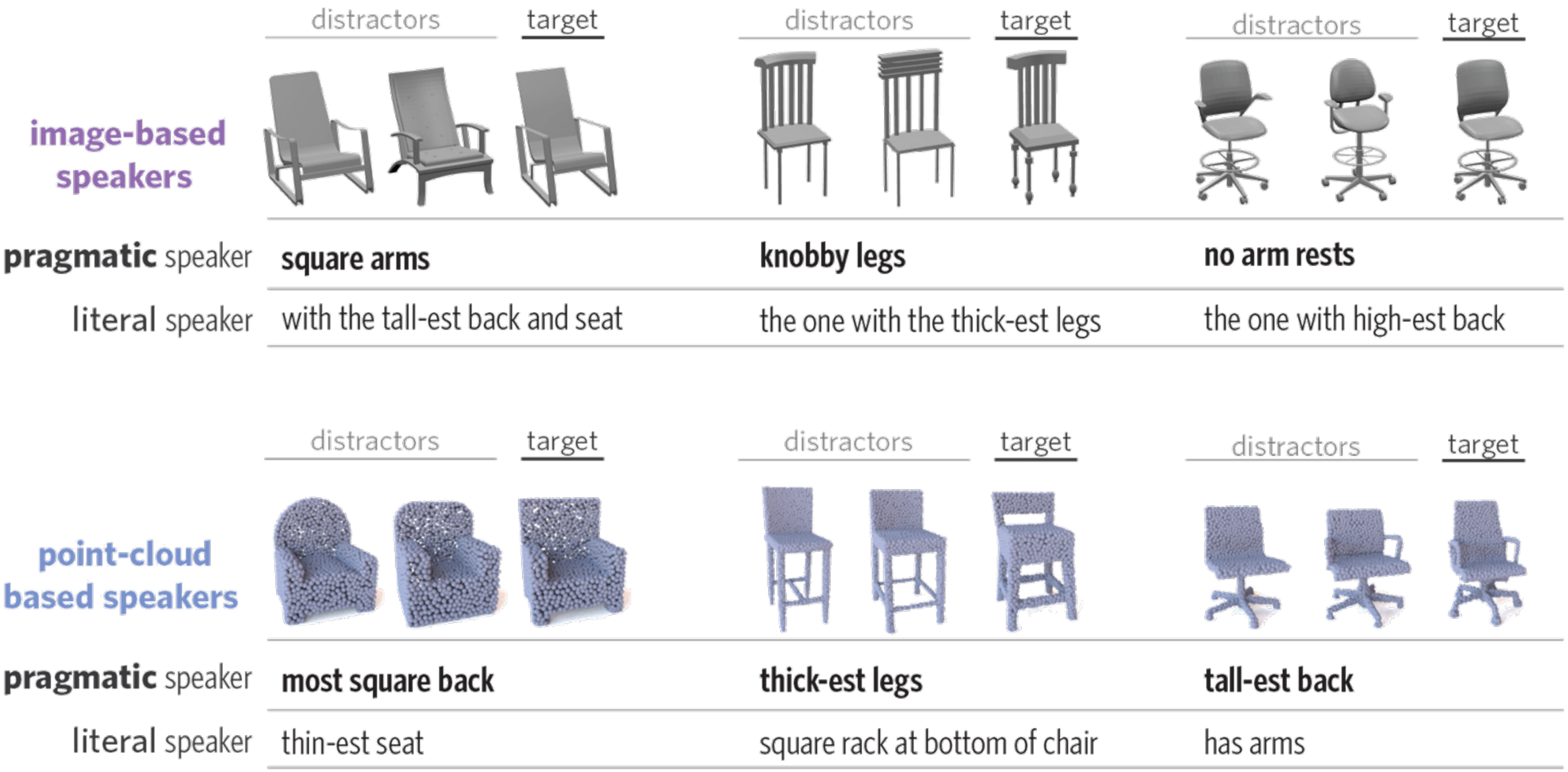


Neural Speaker

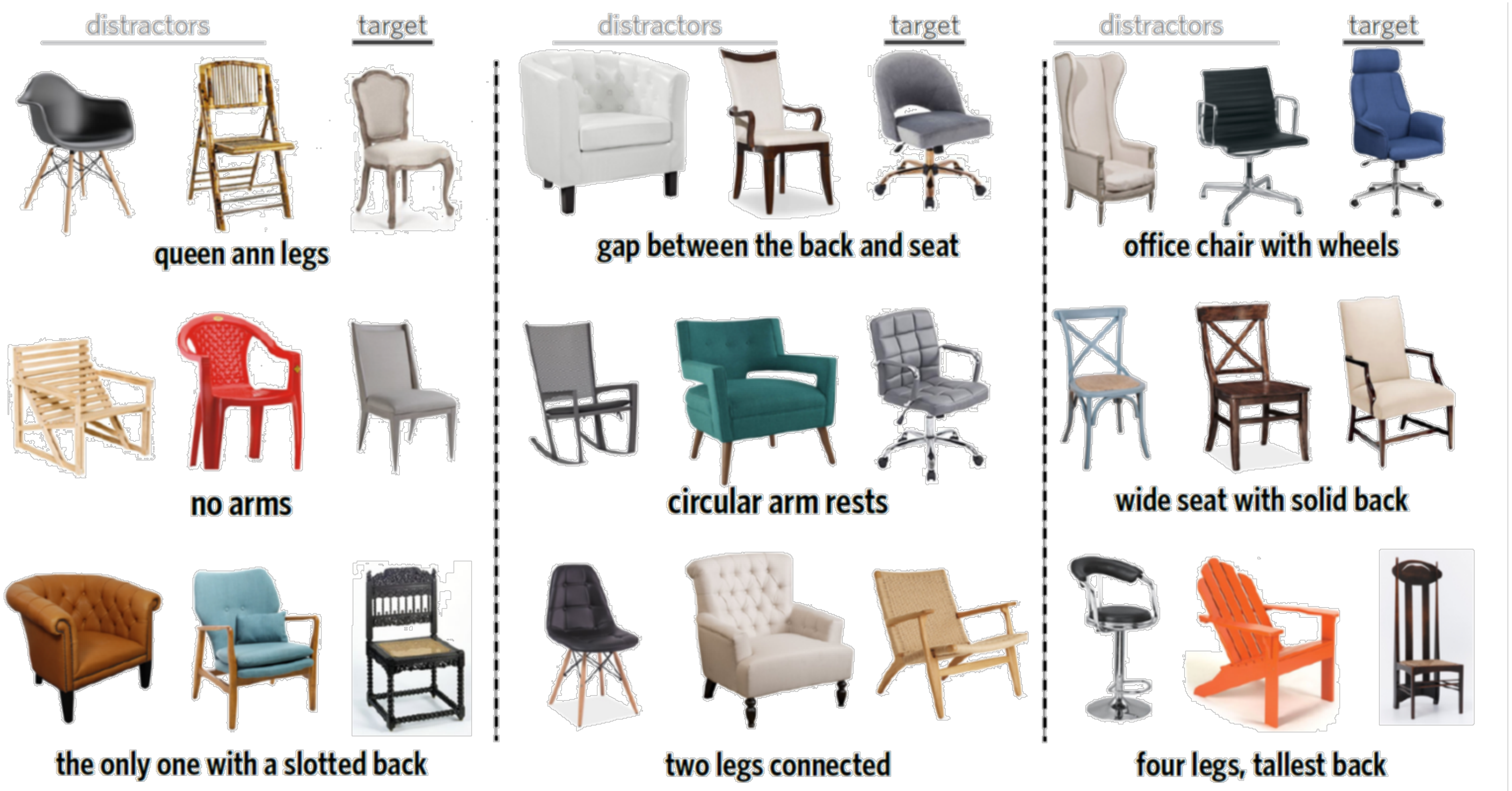
show-and-tell w. teacher forcing



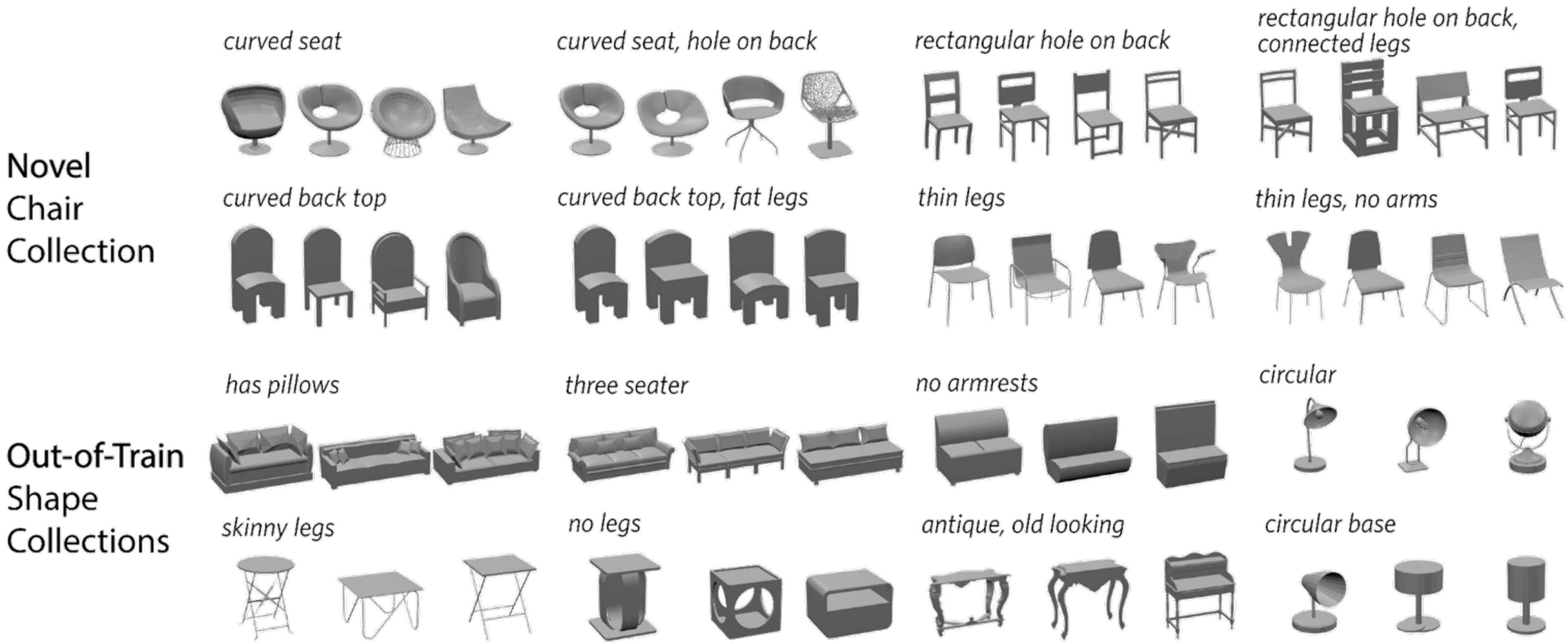
Speaker Examples



Speaker from Images



Listener Examples: Shape-Based Product Retrieval



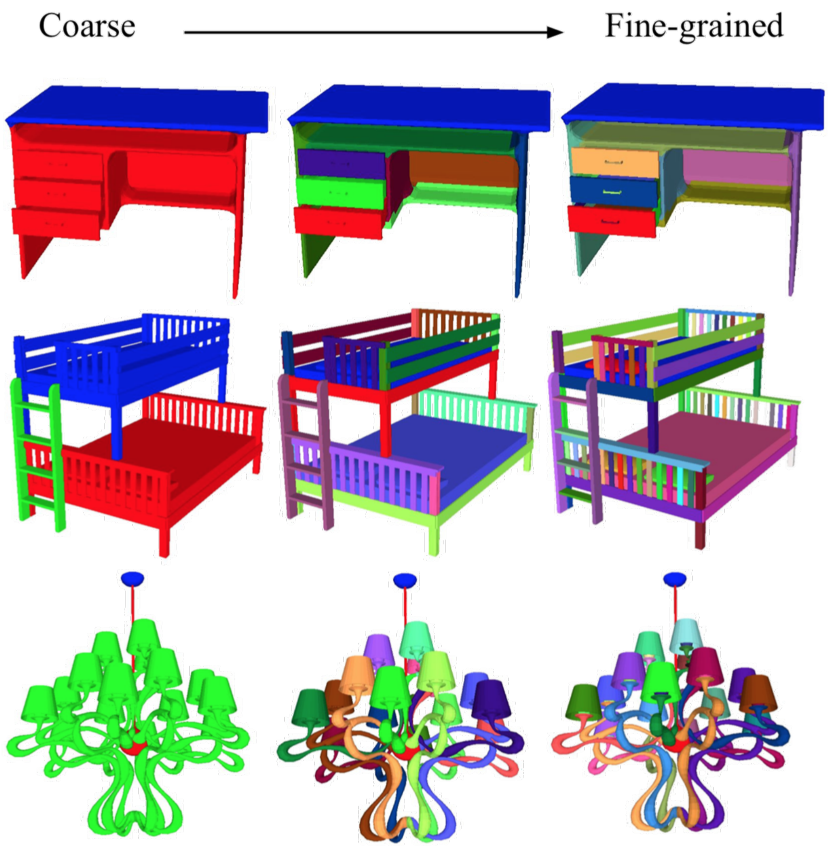
(bottom rows includes ***out-of-training*** classes)

Compositional Shape Structure: Shape Parts

[K. Mo, S. Zhu, A. Chang, L. Yi, S. Tripathi, L. Guibas, H. Su; CVPR '19]

PartNet: Fine-Grained Object Part Annotation

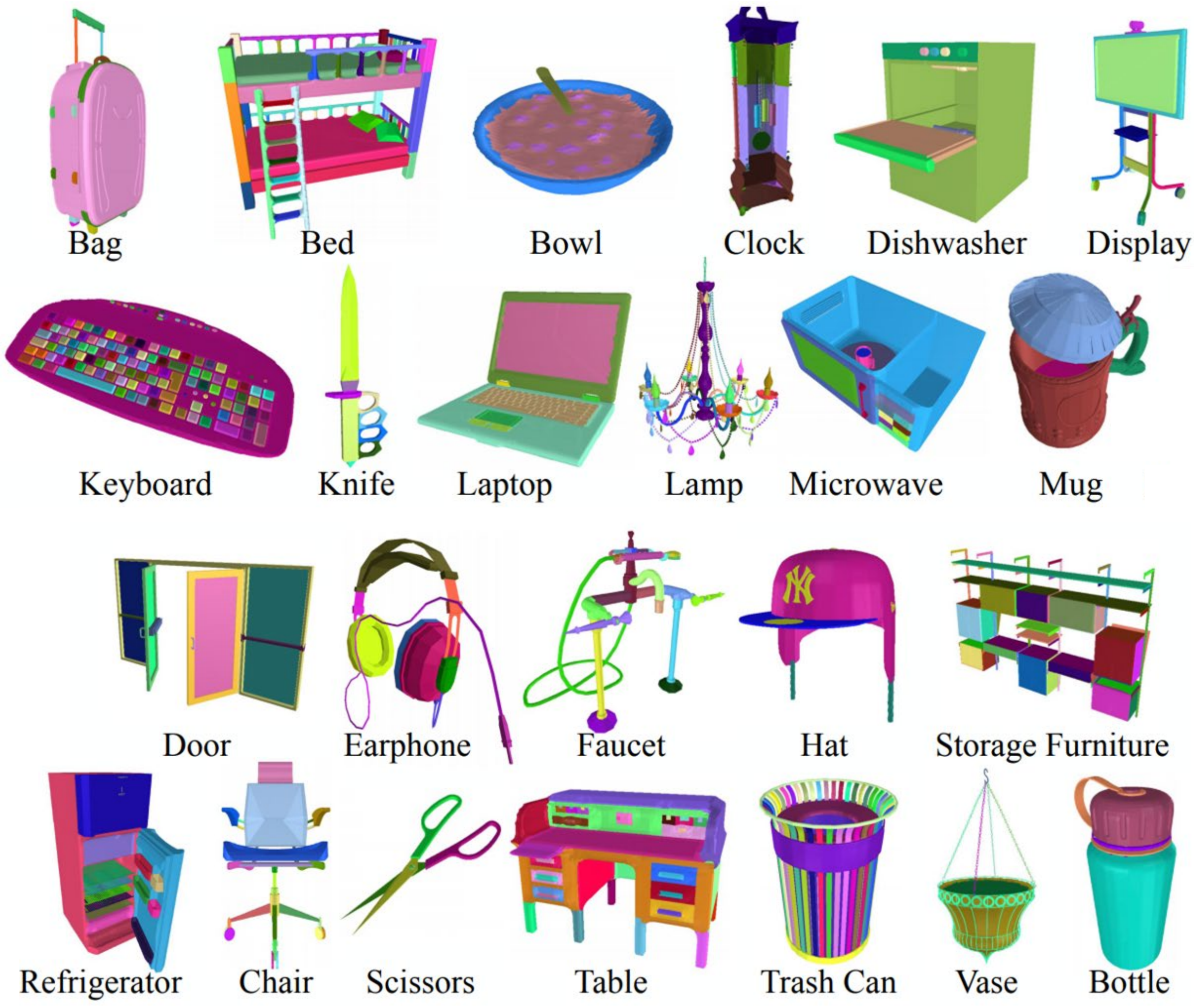
- Dataset
  - Fine-grained Parts
  - Hierarchical Segmentation
  - Instance-level Segmentation
  - Consistent Semantics
- Statistics
  - 24 Object Categories
  - 26,671 Different Shapes
  - 573,585 Different Parts
  - Avg 18 Part/shape, Max 230



PartNet Dataset

[K. Mo, S. Zhu, A. Chang, L. Yi, S. Tripathi, LG, H. Su, CVPR 2019]

Based on Curated Part Hierarchies



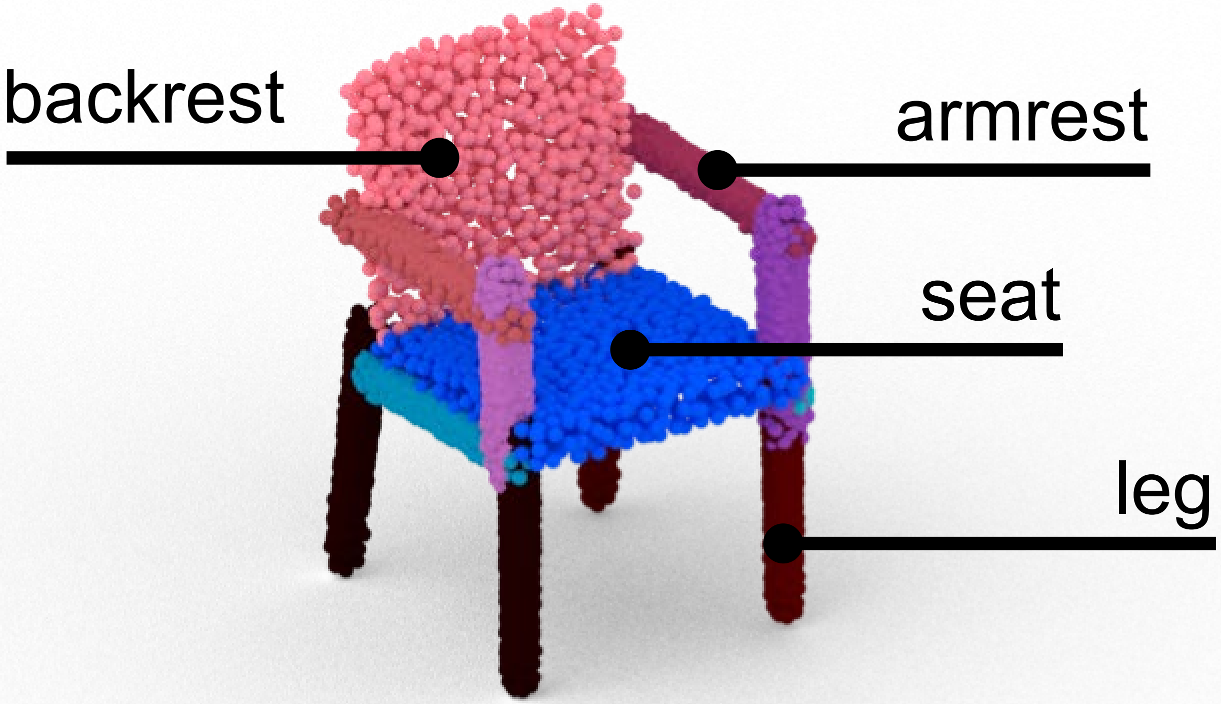
Latent Representations for Shape Structure and Structural Differences

[K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, L. Guibas; Siggraph Asia ‘19]

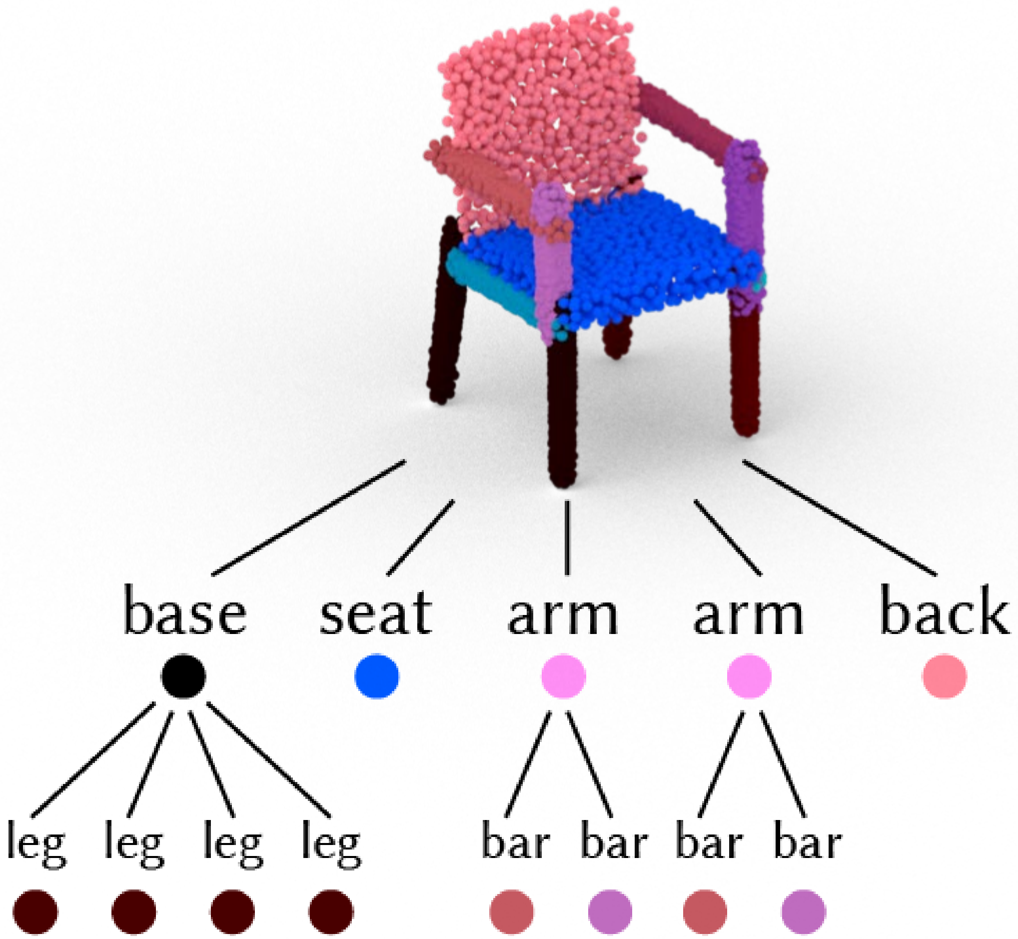
[K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, L. Guibas; ’20]

Geometry and Structure

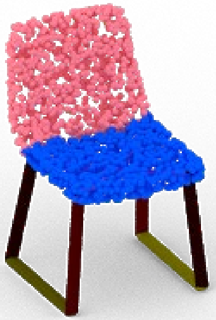
Geometry and Structure

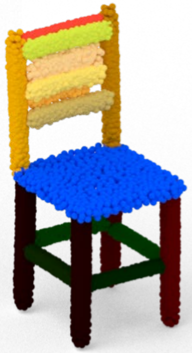


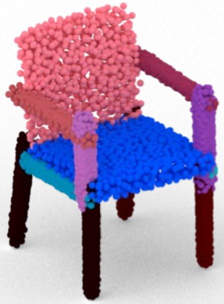
Structure: Part Hierarchy



Goal: A Smooth, Explorable Shape Space

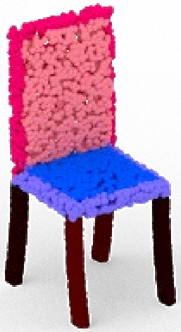
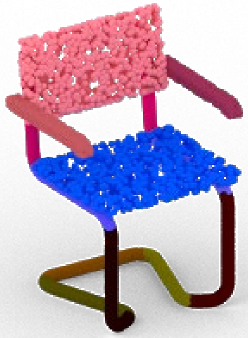
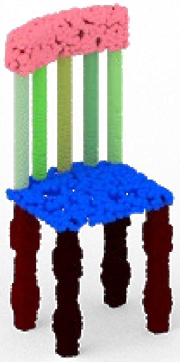





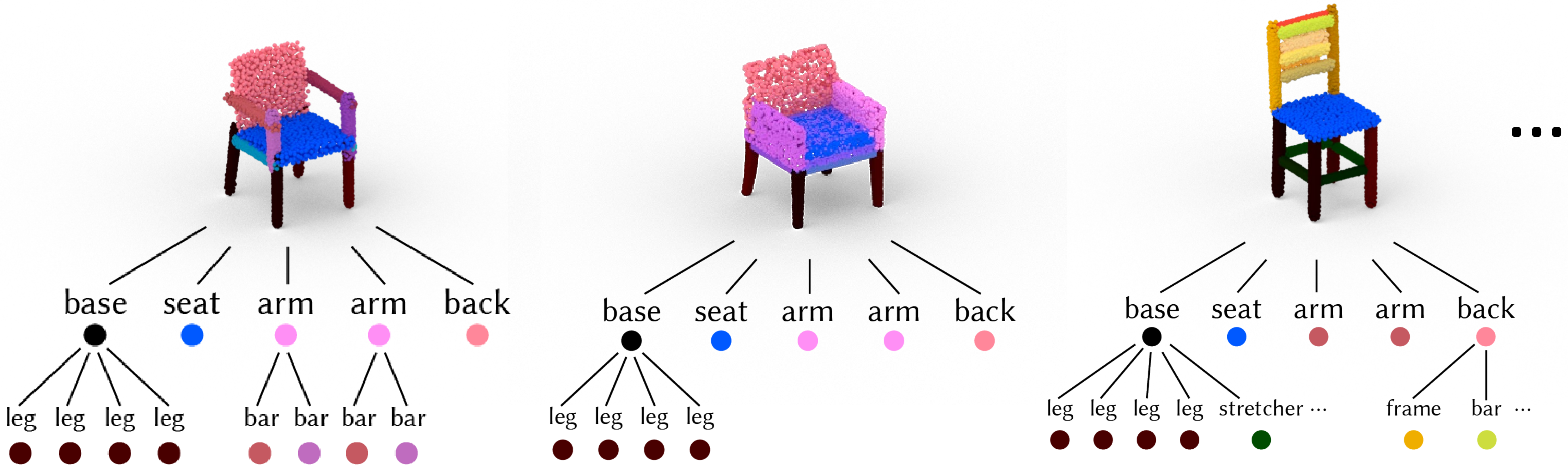


A smooth, common shape space allows for interpolation, generation, exploration, …

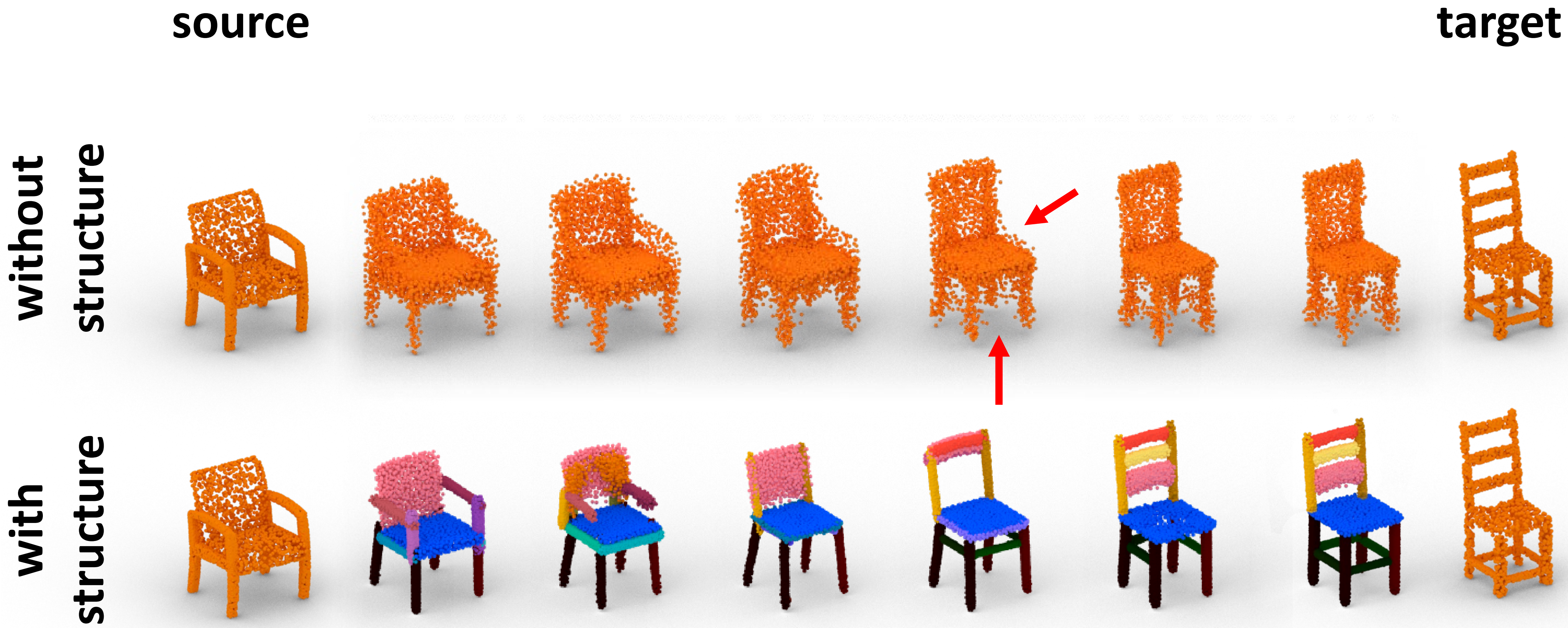
… of both **geometry** and **structure**

Structural Consistency

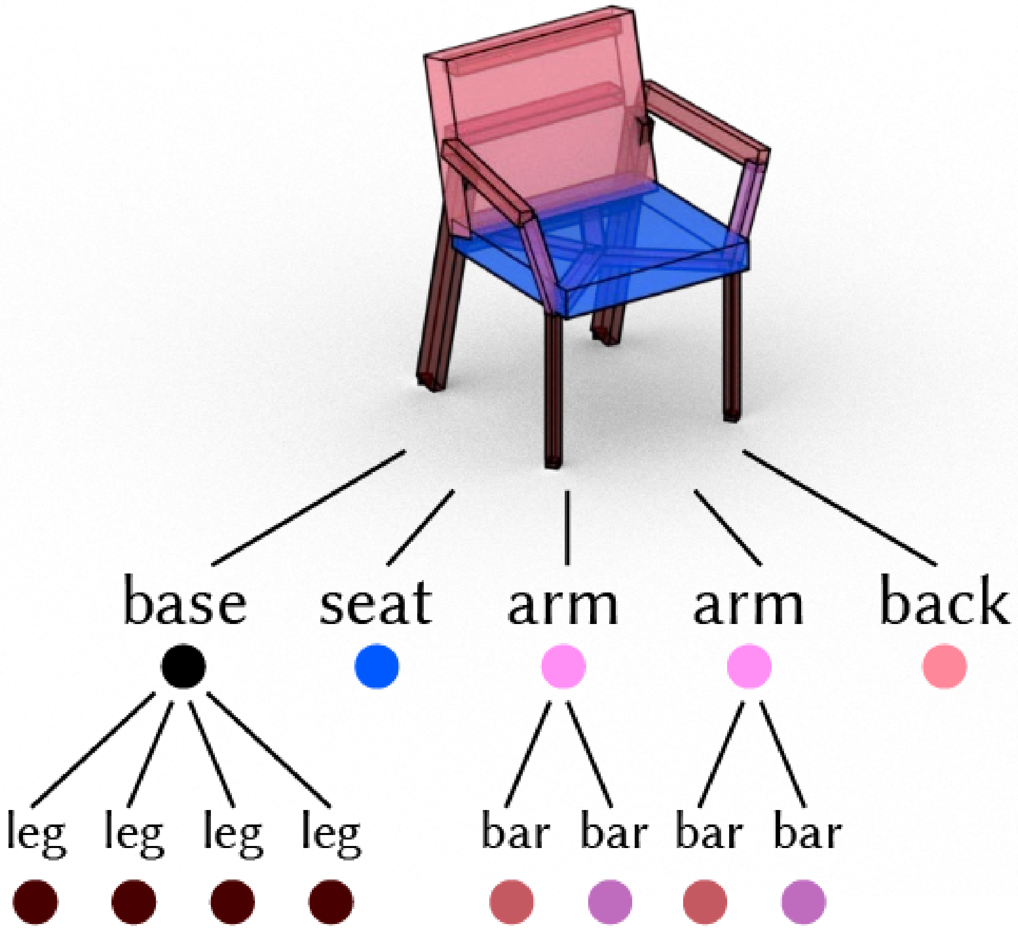


With vs. Without Structure



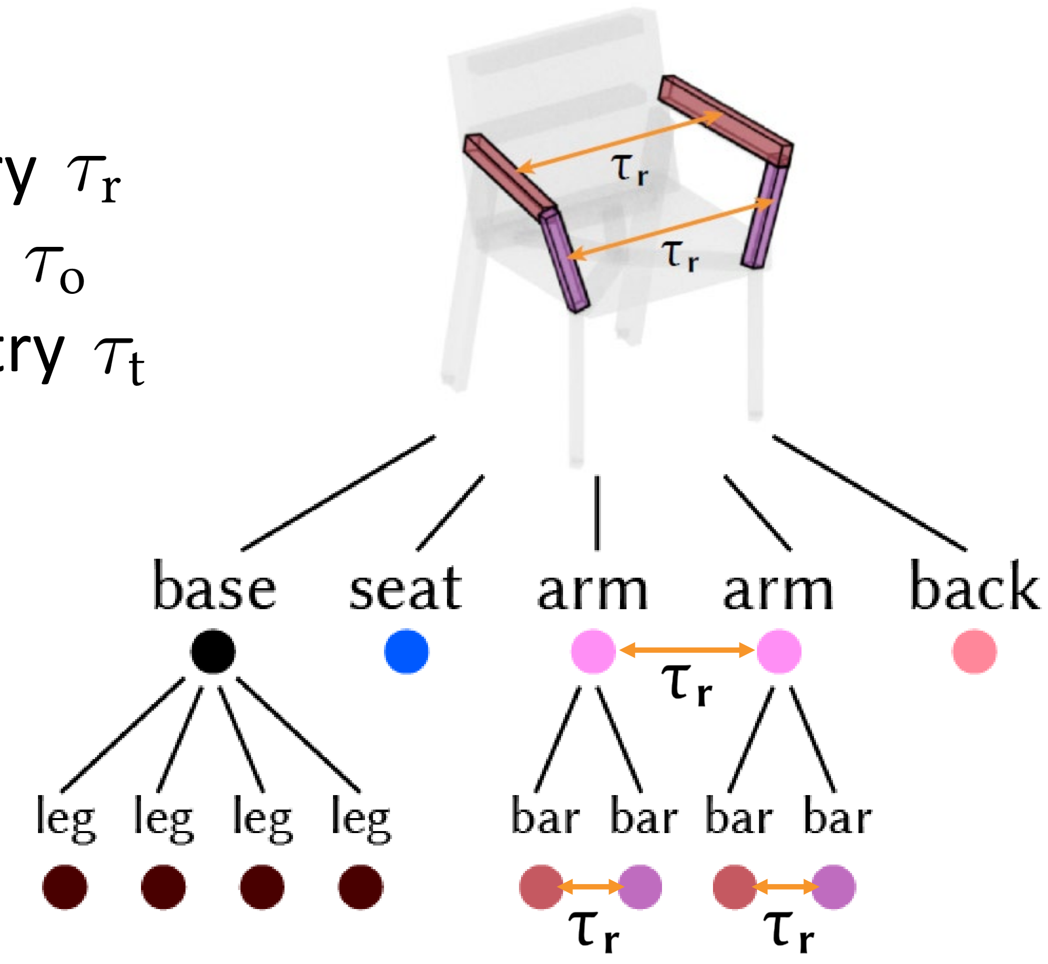
Object Representation: Part Geometry

Object Representation: Part Structure

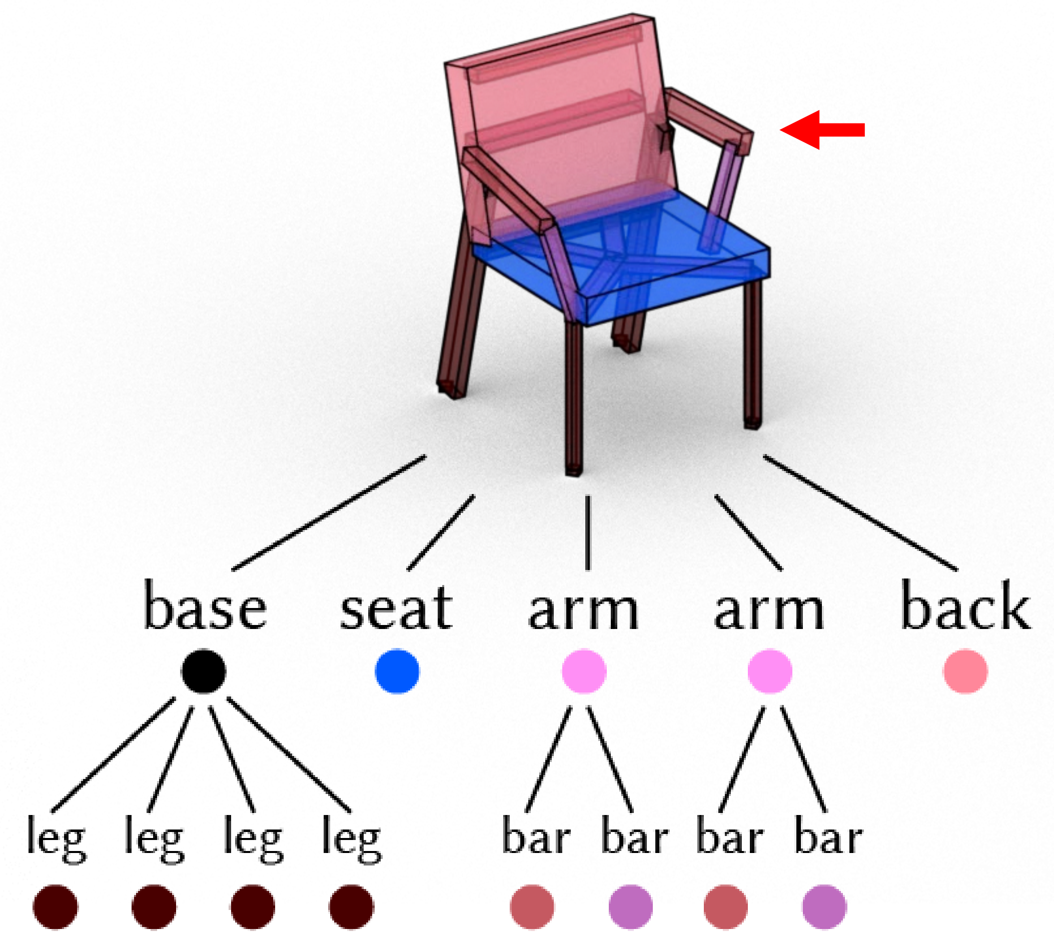


Object Representation: Sibling Relationships

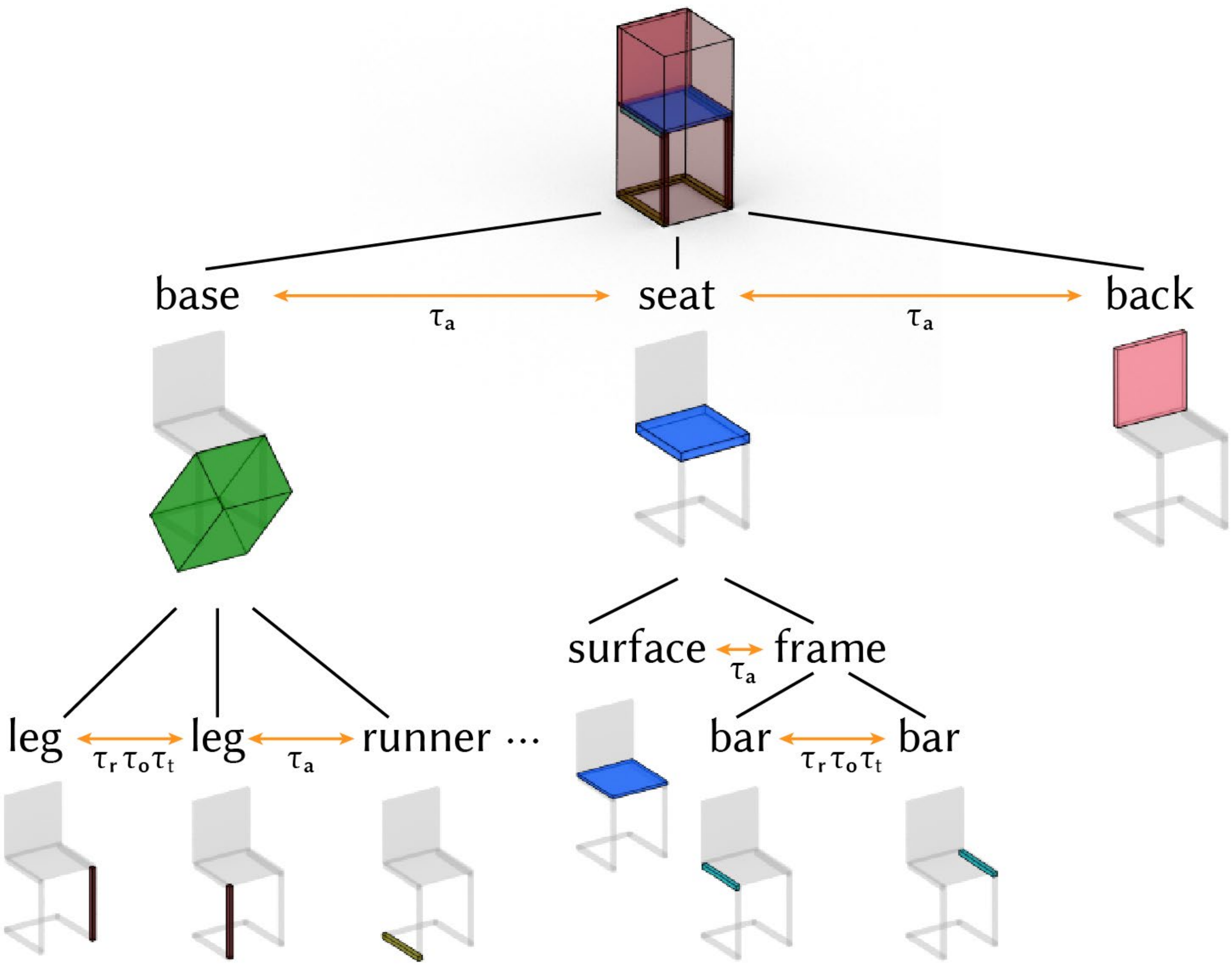
Reflectional Symmetry τ_r
Rotational Symmetry τ_o
Translational Symmetry τ_t
Adjacency τ_a



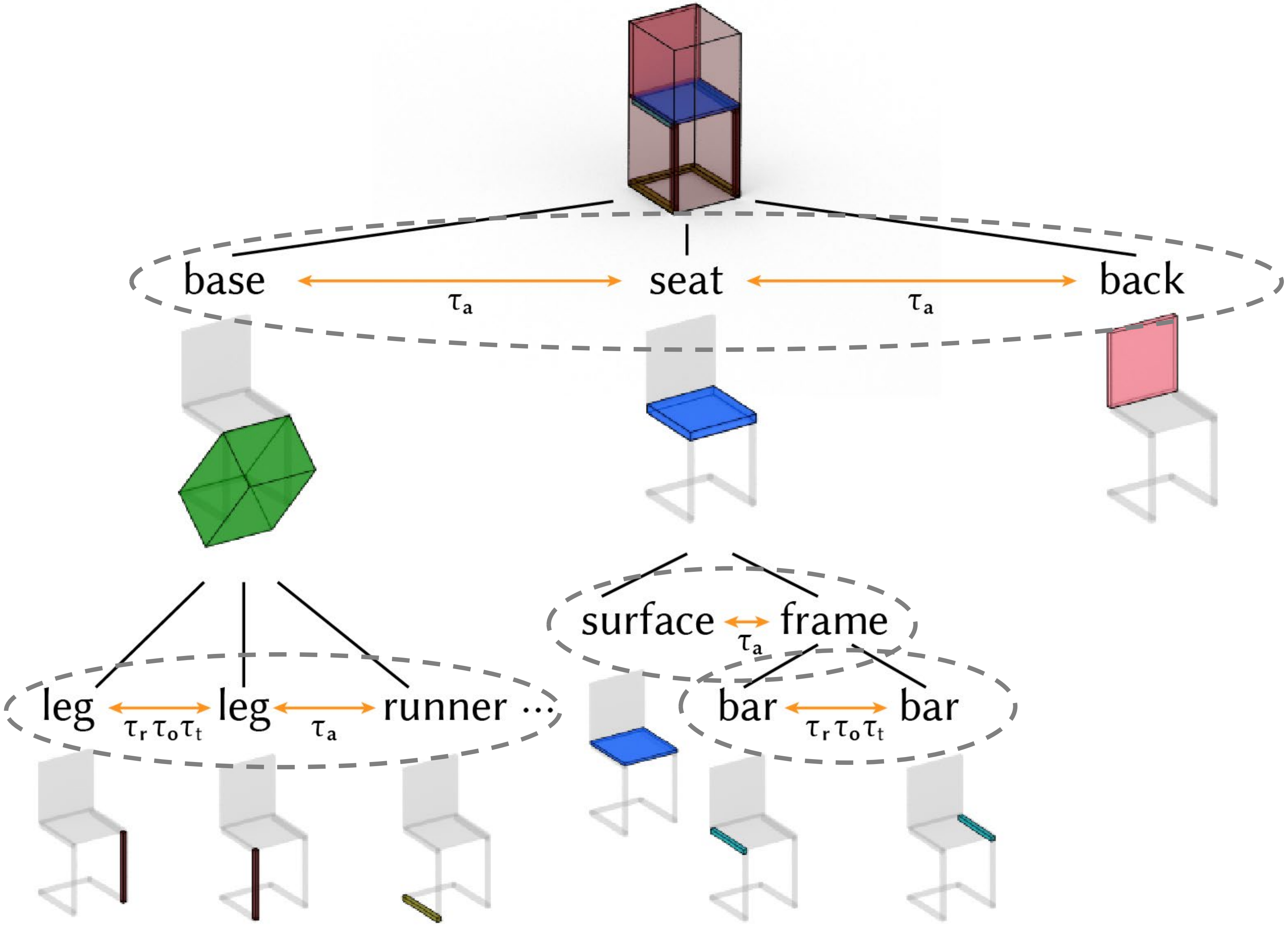
Object Representation: Part Structure



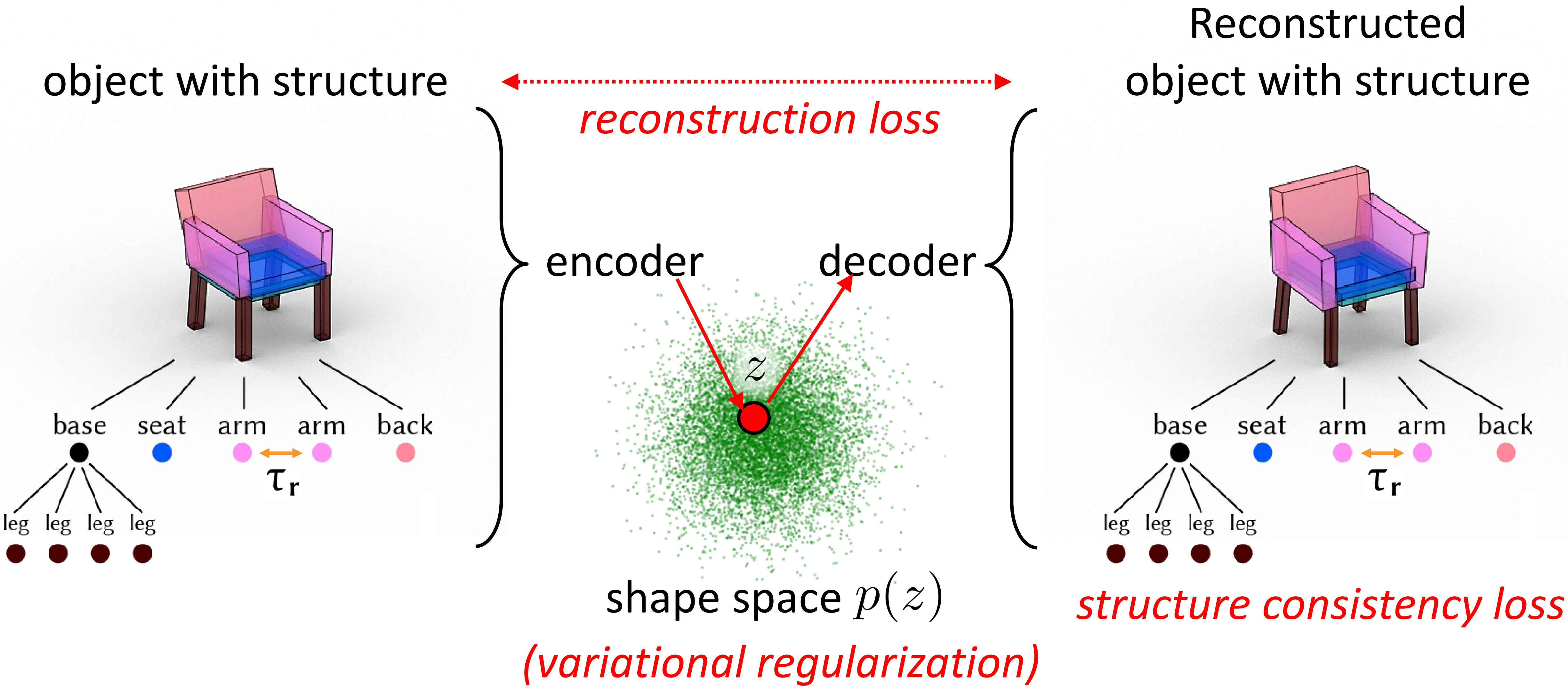
Object Representation: Example



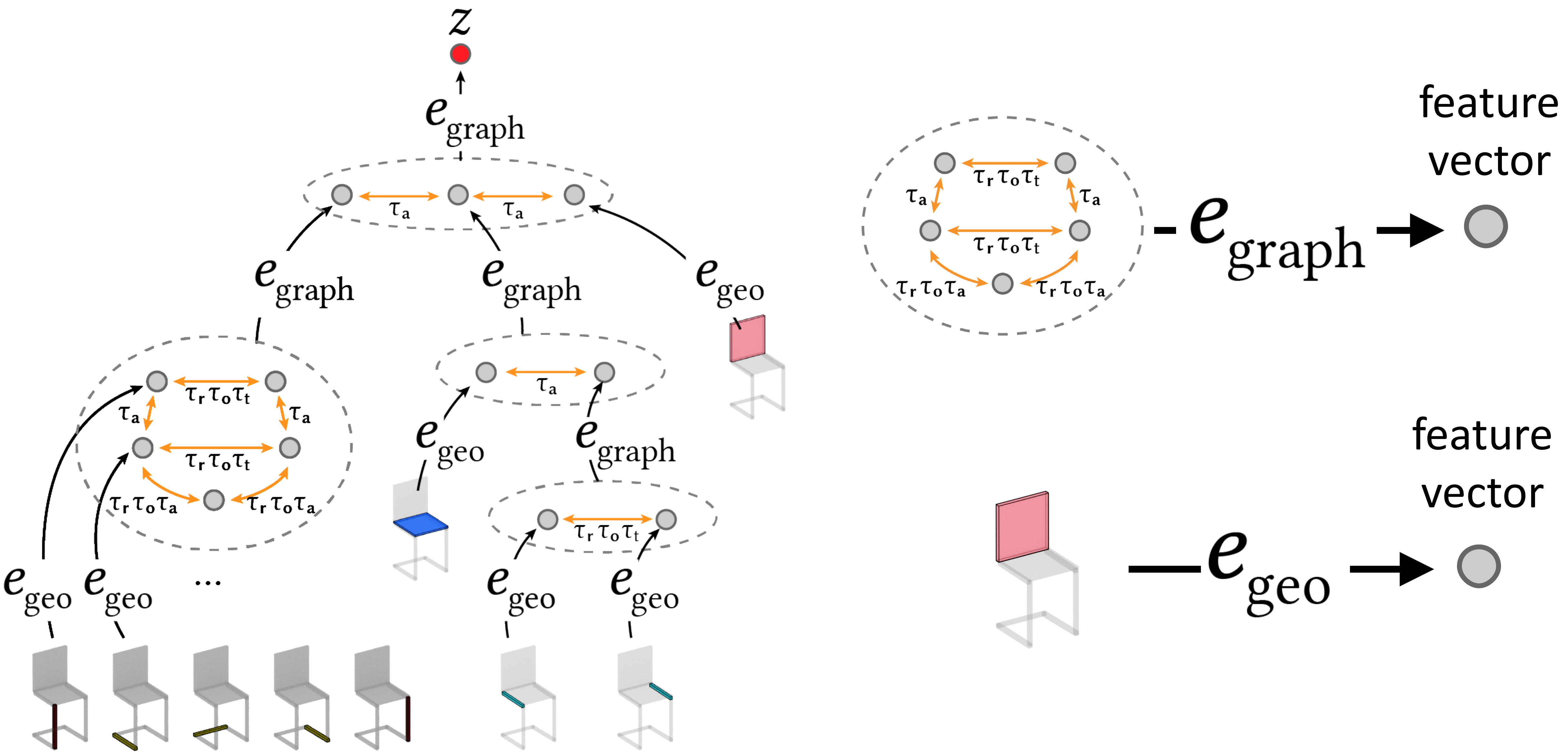
A Hierarchy of Graphs



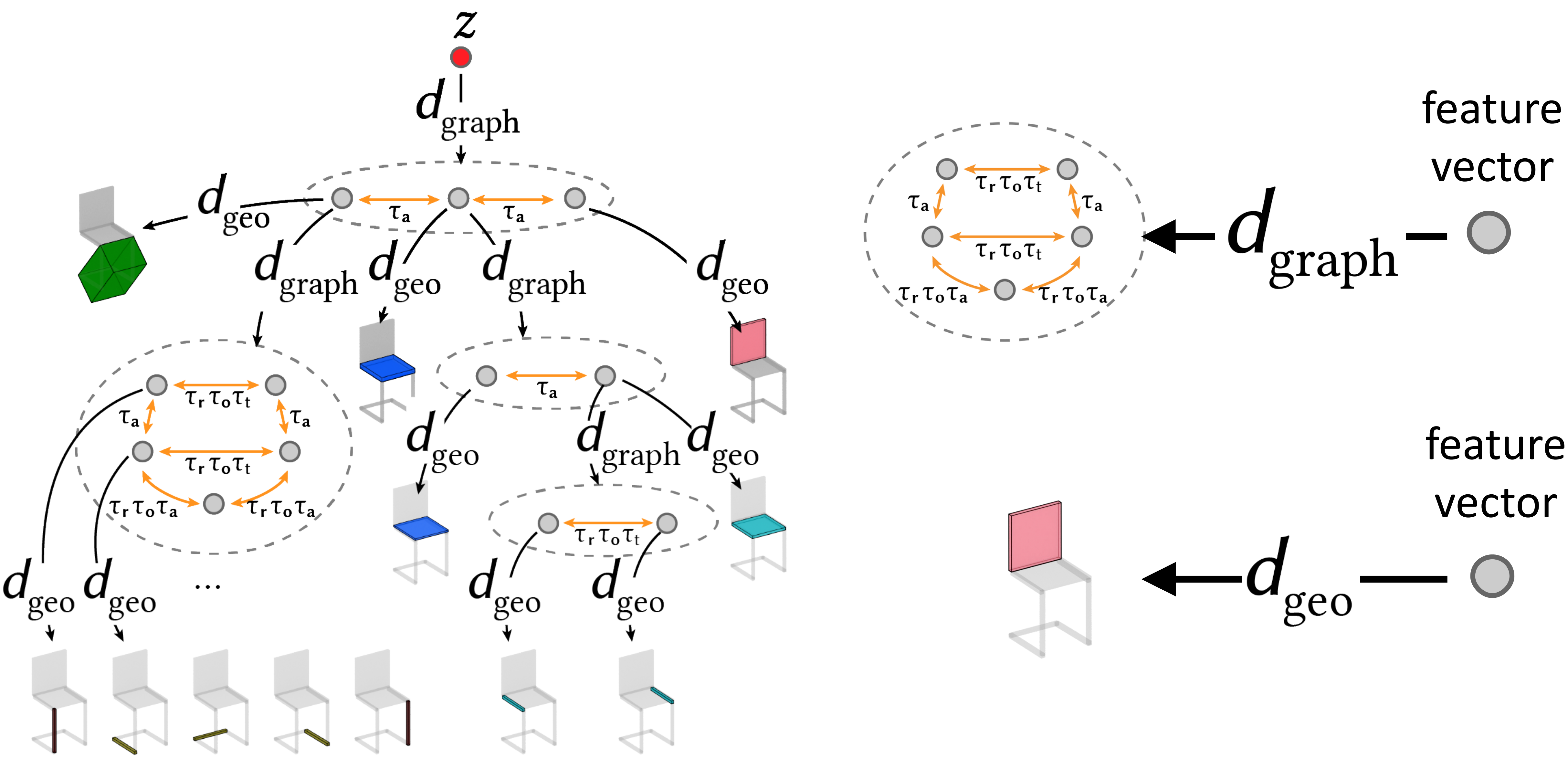
Architecture Overview: VAE Training



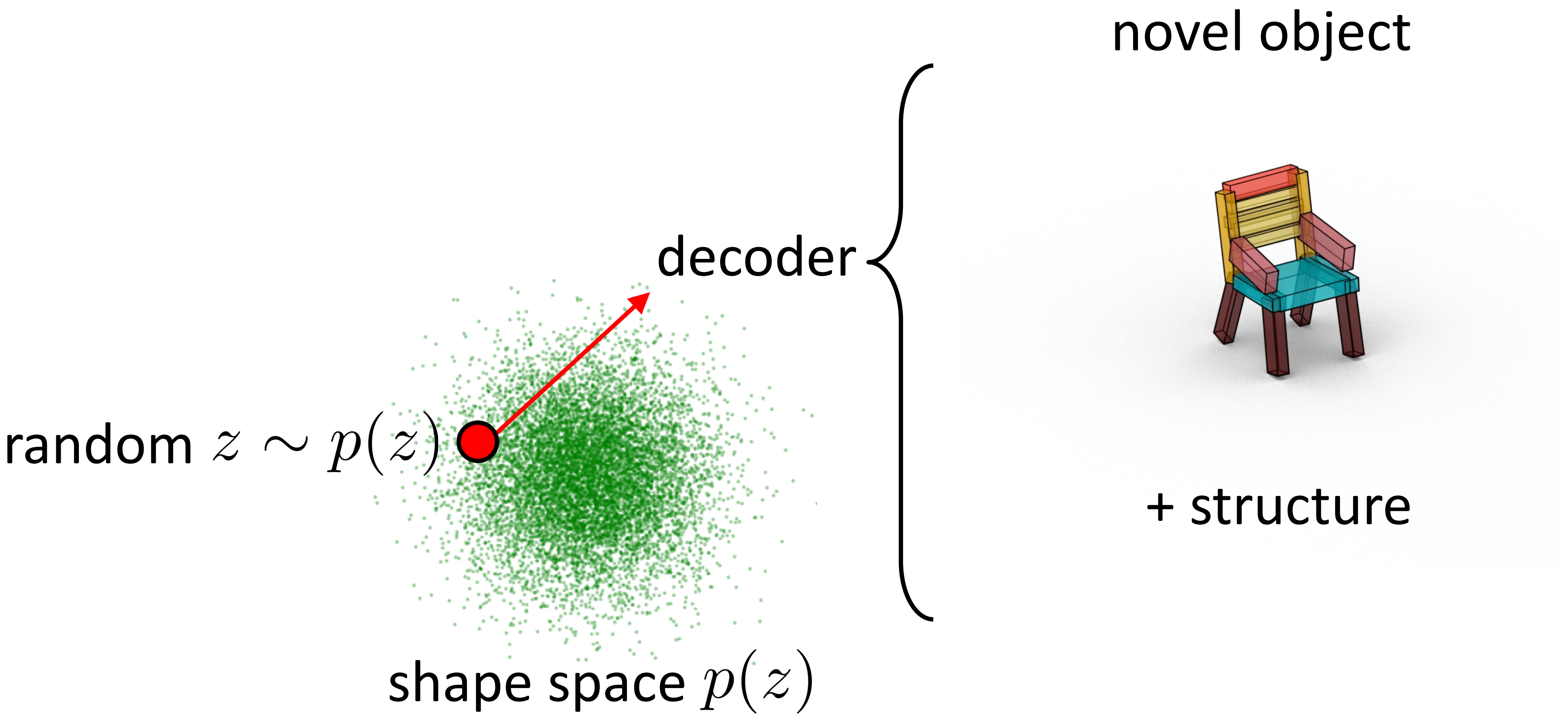
Hierarchical Graph Encoder



Hierarchical Graph Decoder

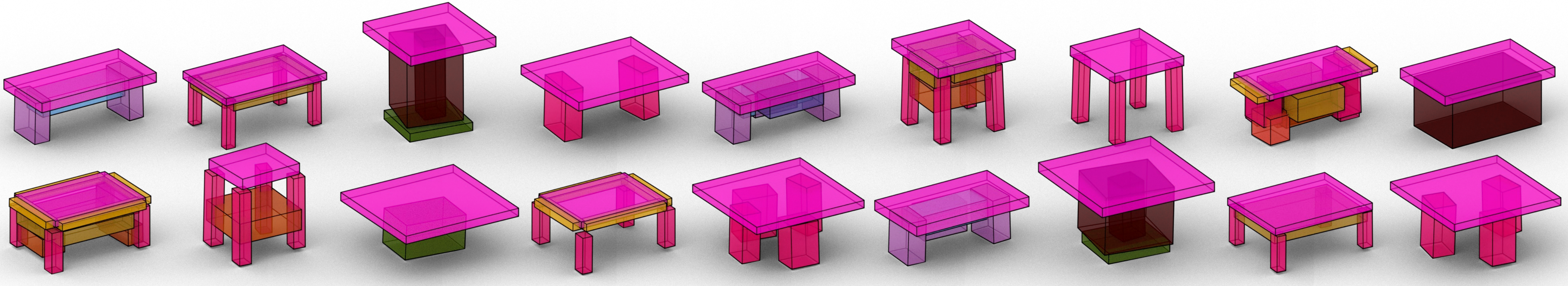


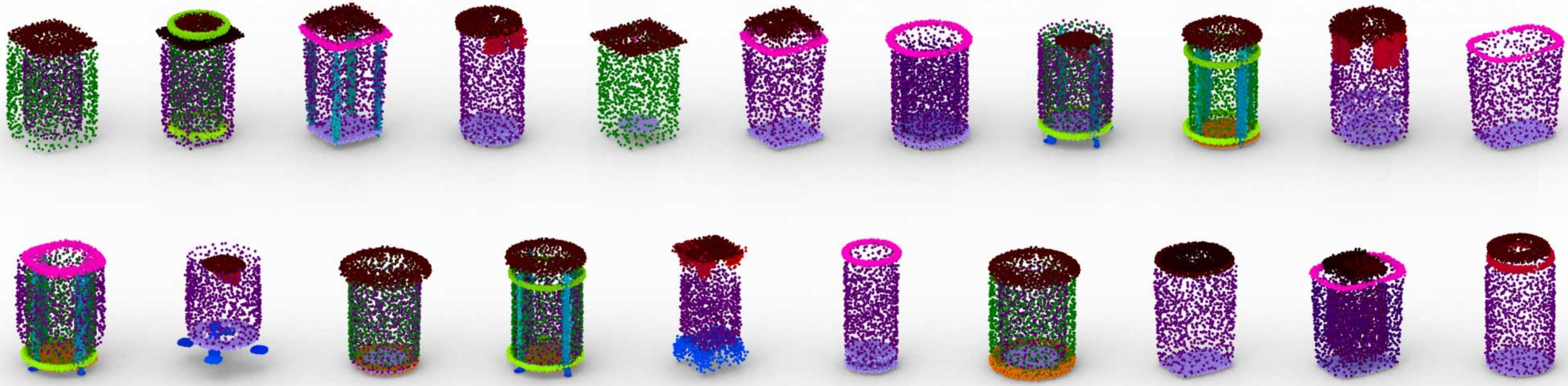
Application 1: Generation



Generation

Generation

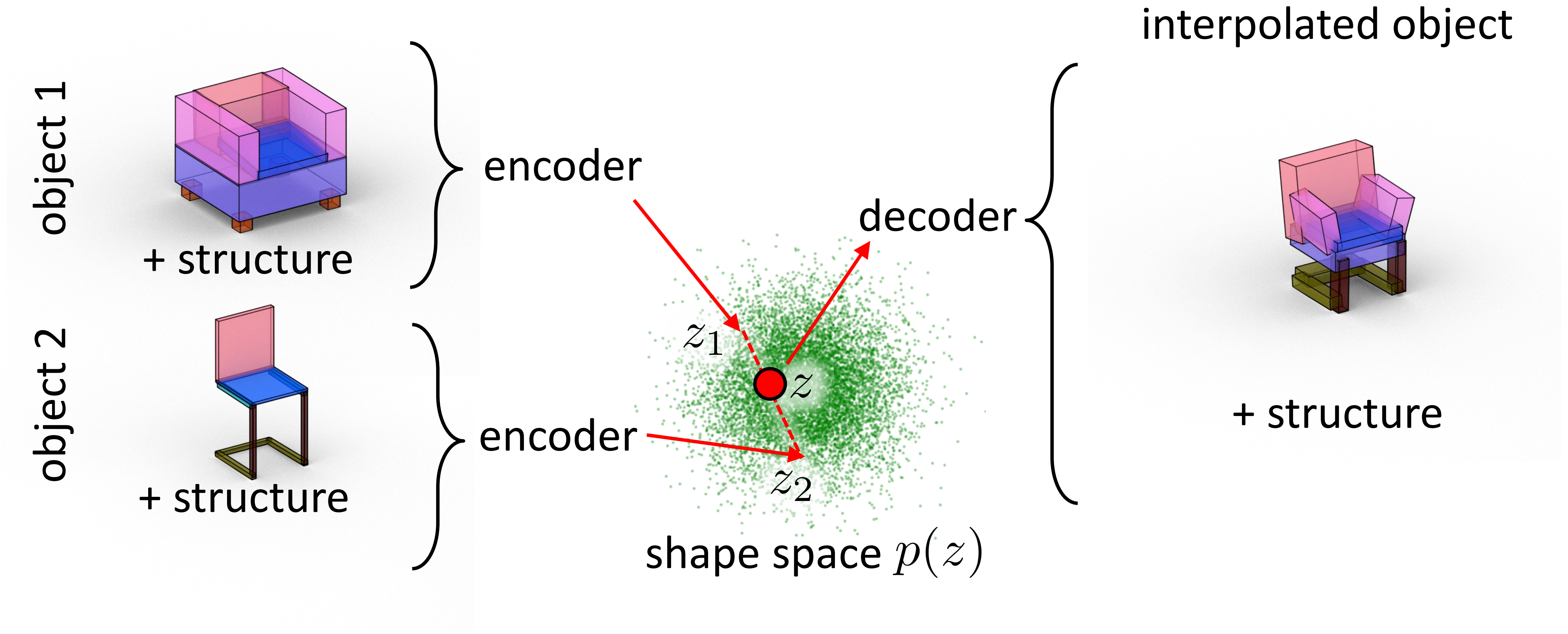




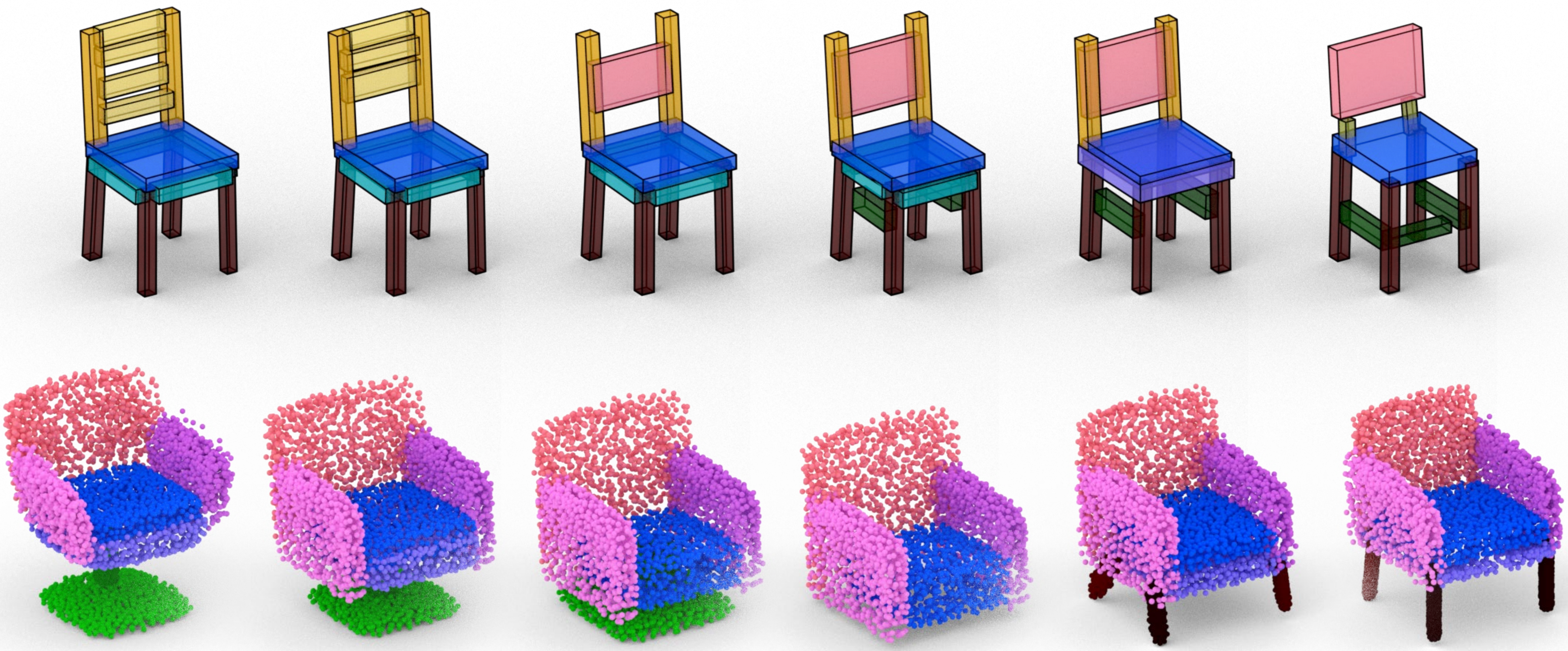
Novelty

generated closest training samples

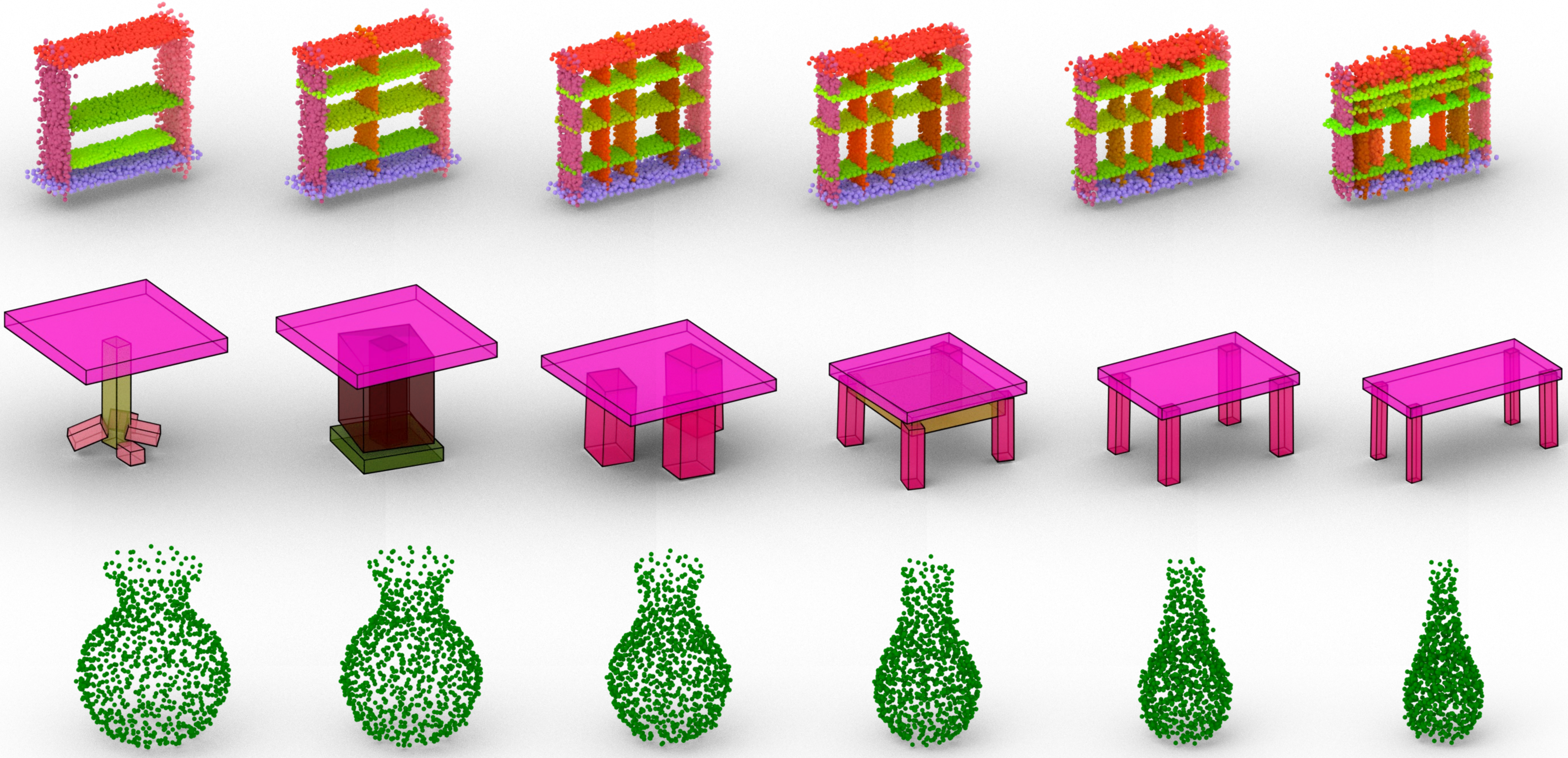
Application 2: Interpolation



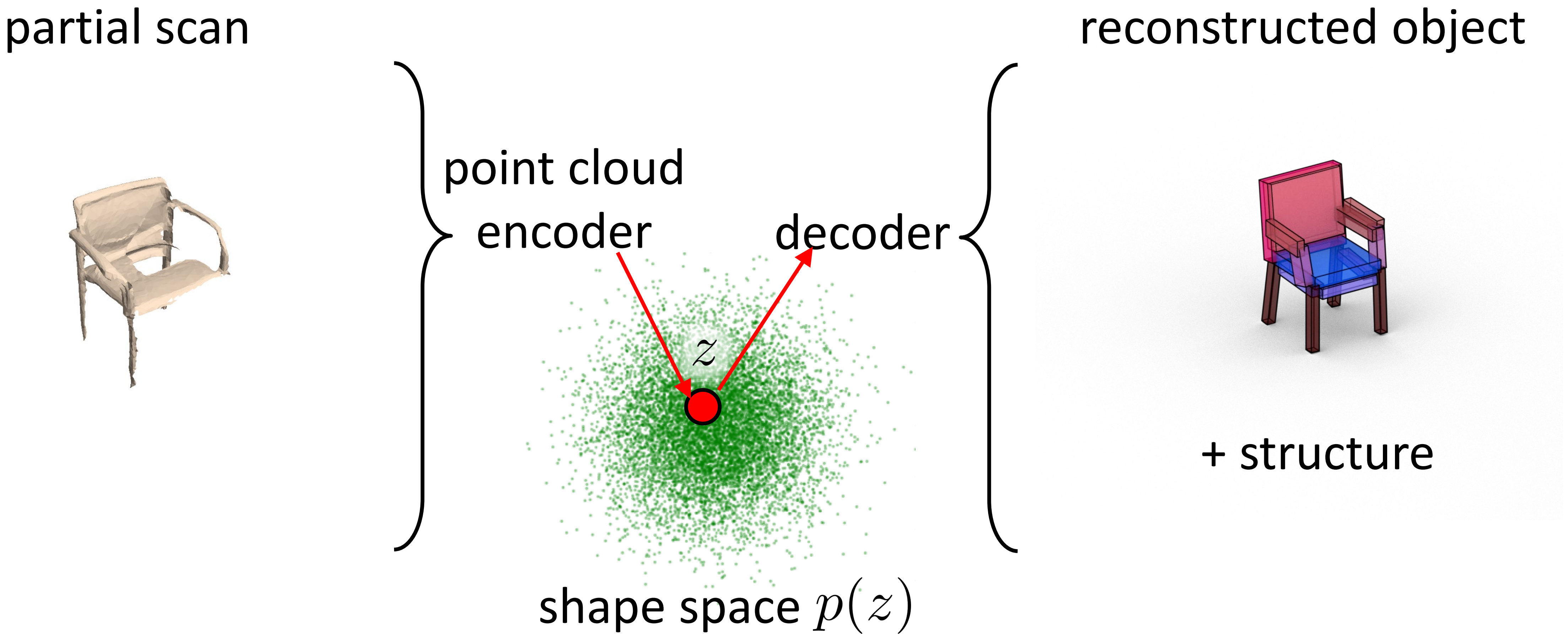
Interpolation



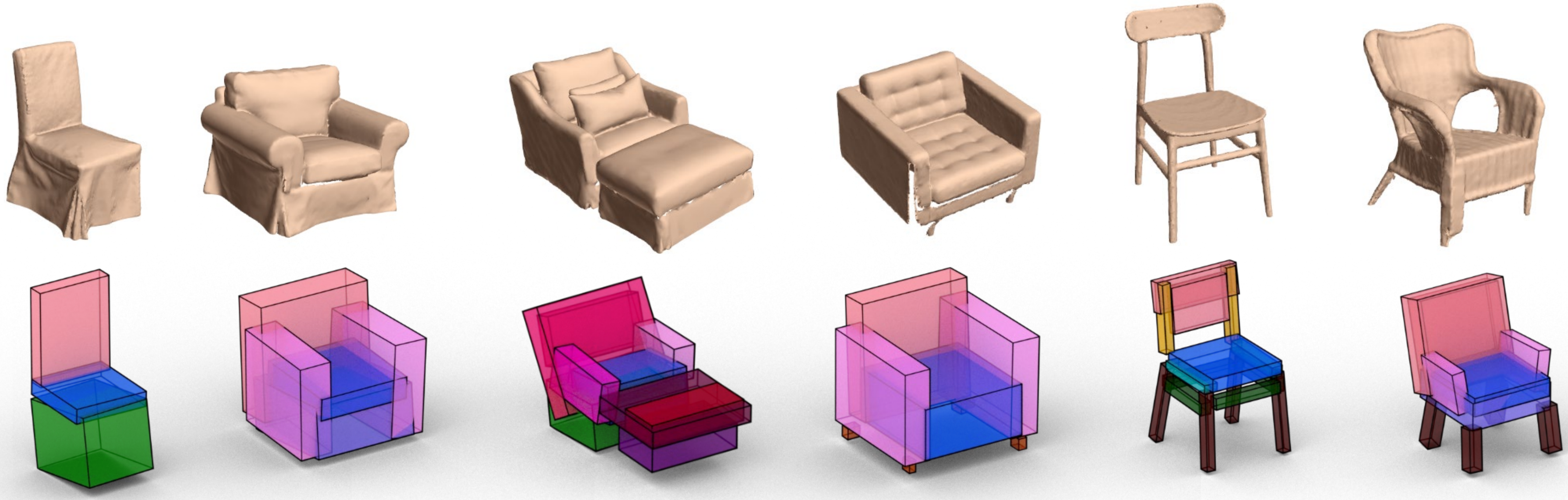
Interpolation



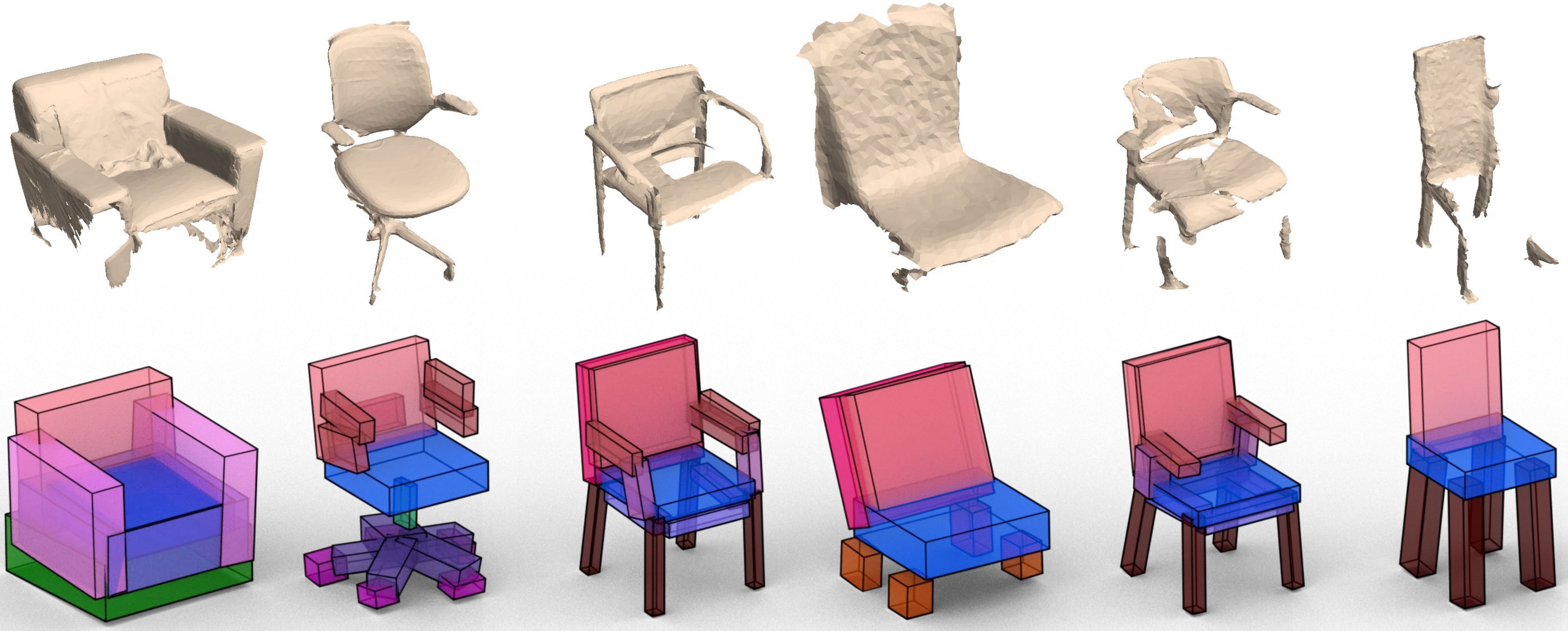
Application 3: Scan Abstraction



Abstraction of Full Scans

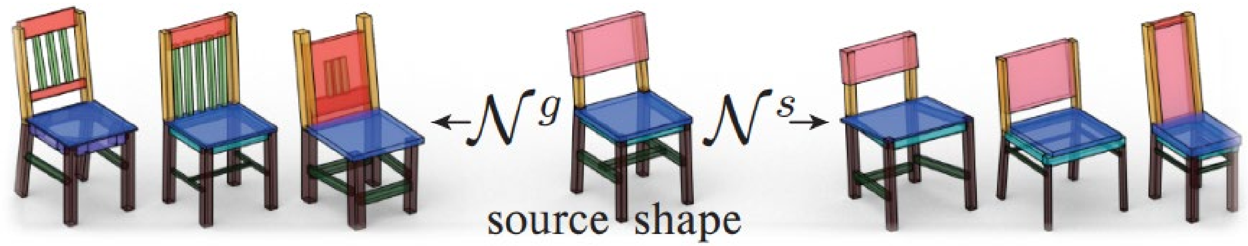


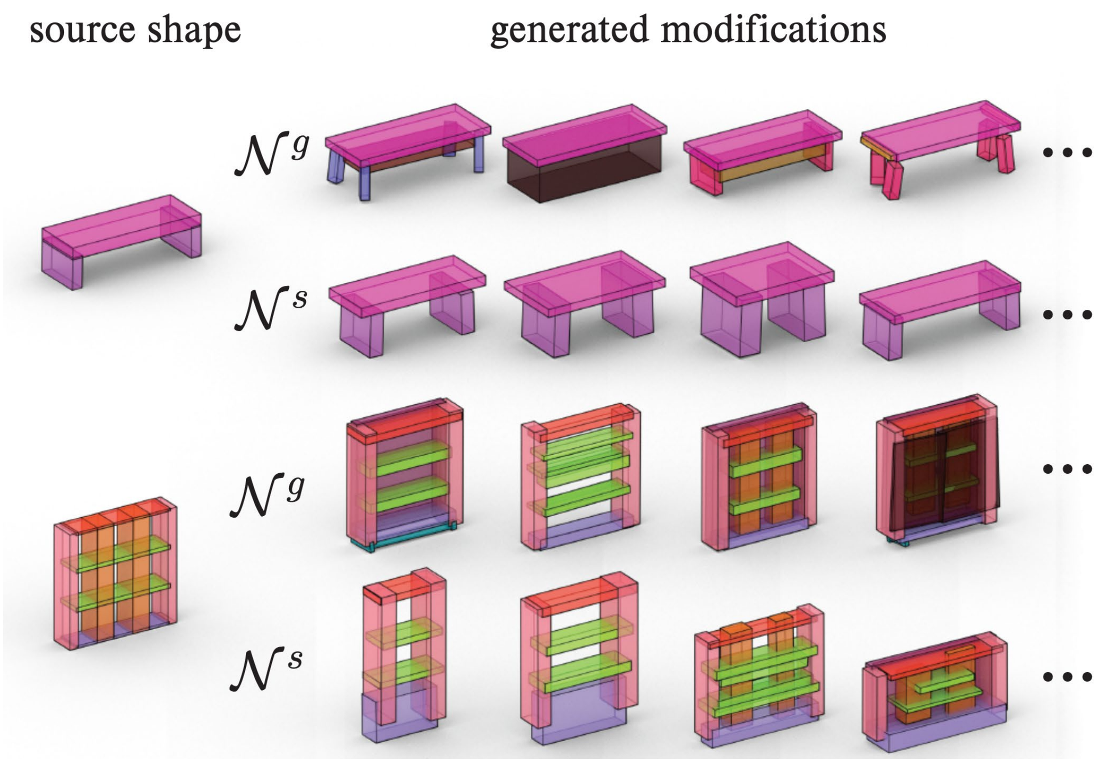
Abstraction of Partial Scans

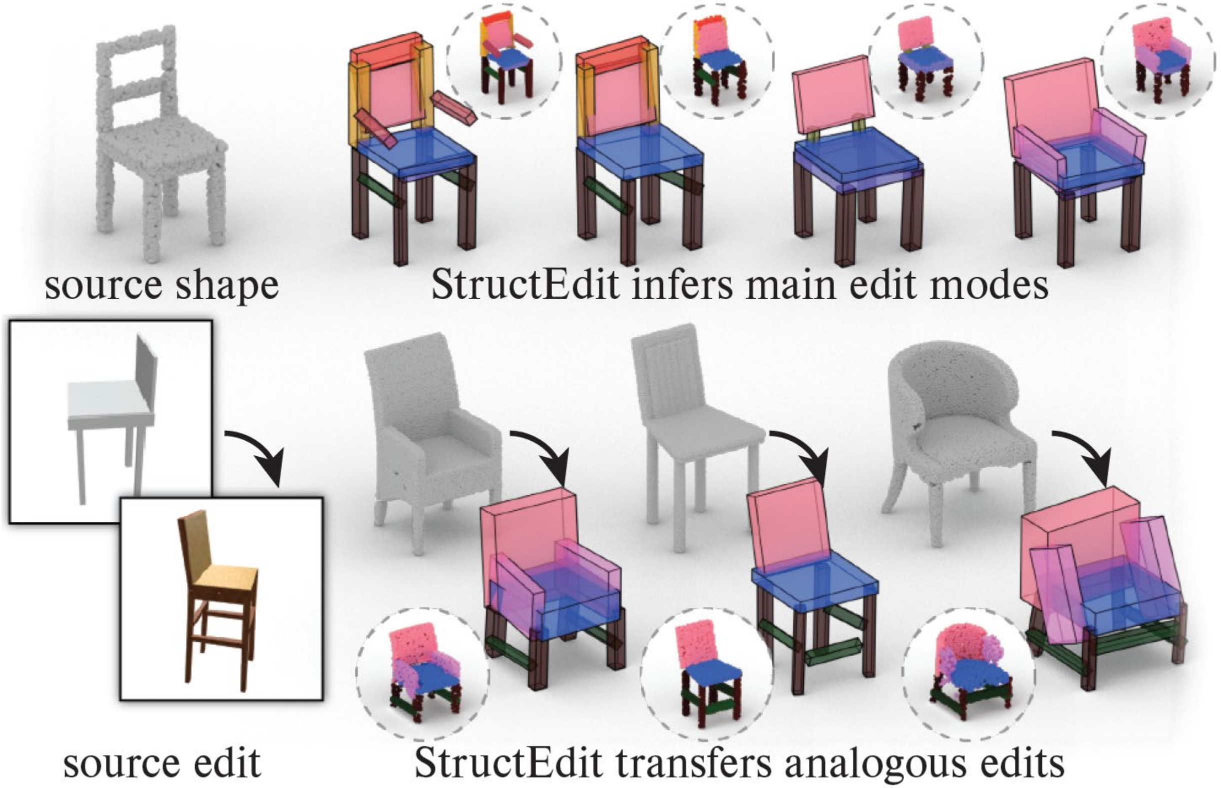


Learning Shape Variations: Geometric and Structural

Two Types of Shape Neighborhoods

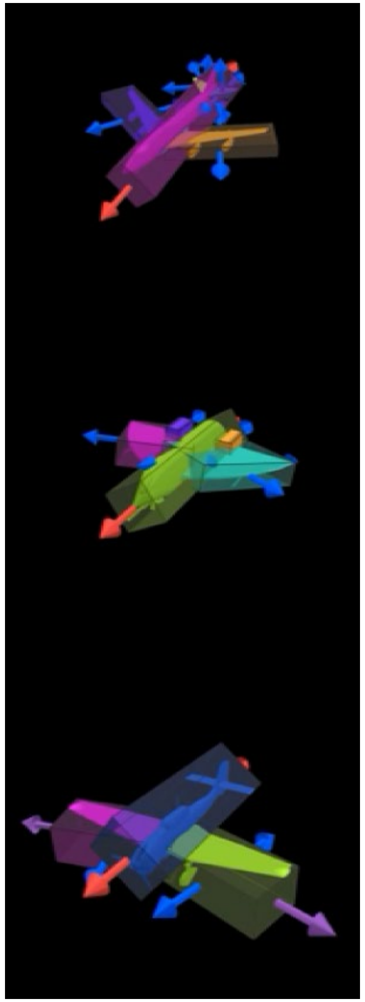






Conclusion: What s a Shape Difference?

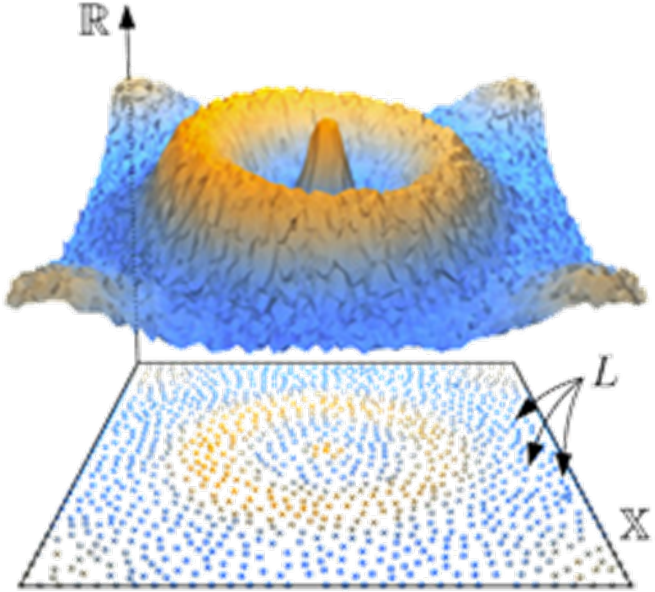
- The notion of shape differences under a map can be given a formal meaning useful in shape collection analysis.
- Differences can be used for shape interpolation, analogies (differences of differences), or reconstruction.
- Geometric shape differences can be mapped to/from language, using deep neural networks.
- Generating proper variations of a shape is an important tool for understanding its semantics
- Shape parts and compositional structure is an essential aspect of both shape generation as well as of understanding the function of shapes.



The Course

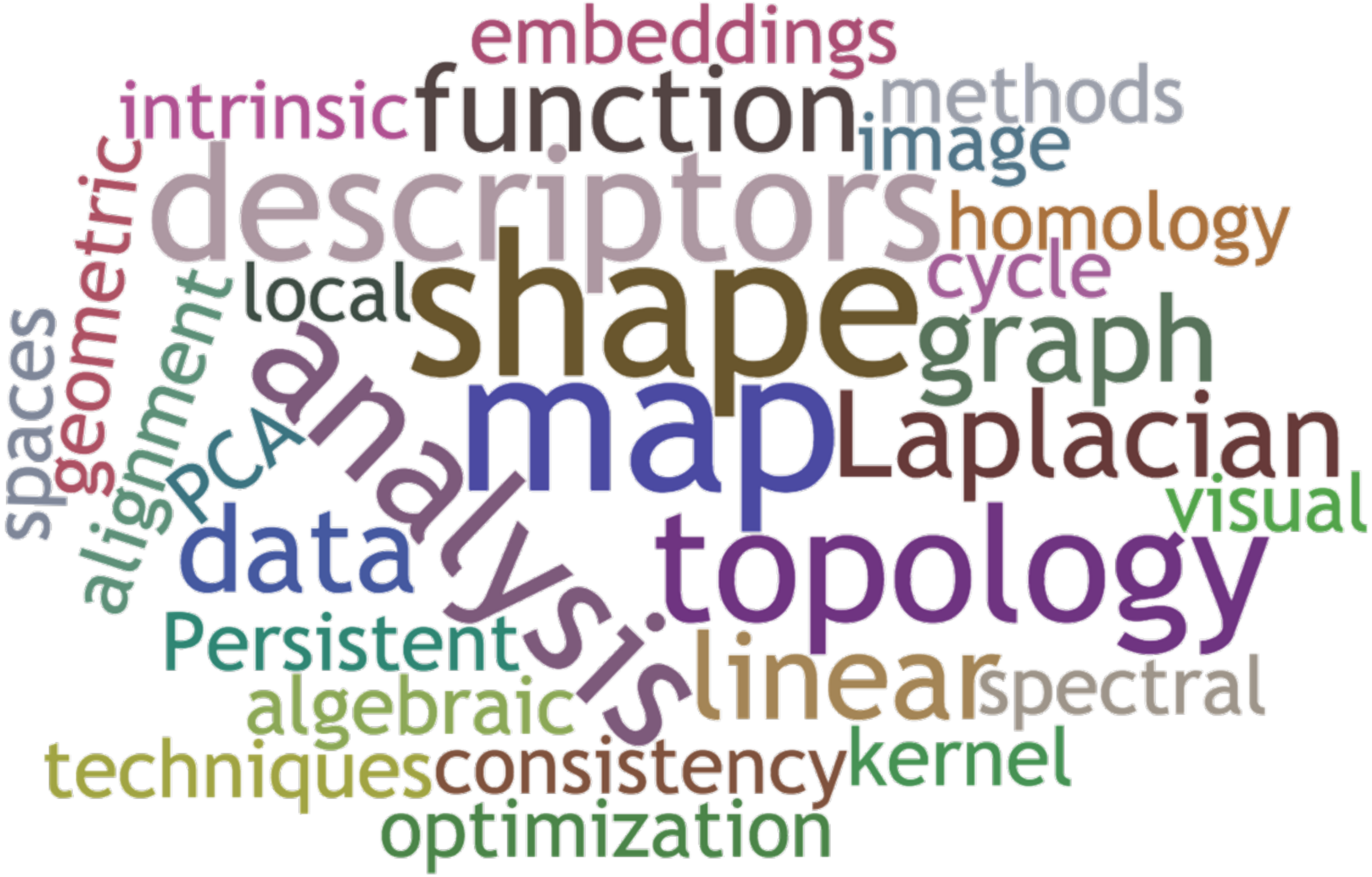
Data Analysis

- (Geometric and Topological Data) (Analysis)
- (Geometric and Topological) (Data Analysis)



Topics Covered

- Visual data sets
- PCA
- CCA
- Spectral graph methods
- MDS; NLDR
- Intro to computational topology
- Homology and persistent homology
- Mapper
- 3D geometry representations
- Shape descriptors (shape context, spin images, HKS, WKS)
- Rigid alignments (ICP, RANSAC, geometric hashing)
- Non-rigid alignments (isometric, conformal)
- Shape correspondences
- Volumetric and Multiview CNNs
- Deep learning on point clouds
- Graph and mesh CNNs
- Functional Maps
- Shape differences
- Map networks and cycle consistency


embeddings
intrinsic
function
methods
image
descriptors
homology
cycle
geometric
local
shape
graph
spaces
alignment
analysis
PCA
map
Laplacian
visual
data
topology
Persistent
linear
spectral
algebraic
techniques
consistency
kernel
optimization

Topics Not Covered

- Factor analysis
- Independent components analysis
- Nearest neighbor search
- Locality sensitive hashing
- Clustering
- Topic modeling (LDA, etc)
- Tensor decompositions
- Dictionary learning
- Sparse recovery / compressive sensing
- Mixture models
- Non-negative matrix factorization
- Matrix completion
- Zig-zag persistence
- Reeb graphs
- Random graphs and network models

Please send e-mail about topics you'd like to see covered in future CS233 offerings

The End

If you would like to pursue projects or research related to the topics of this class, please get in touch:

guibas@cs.stanford.edu