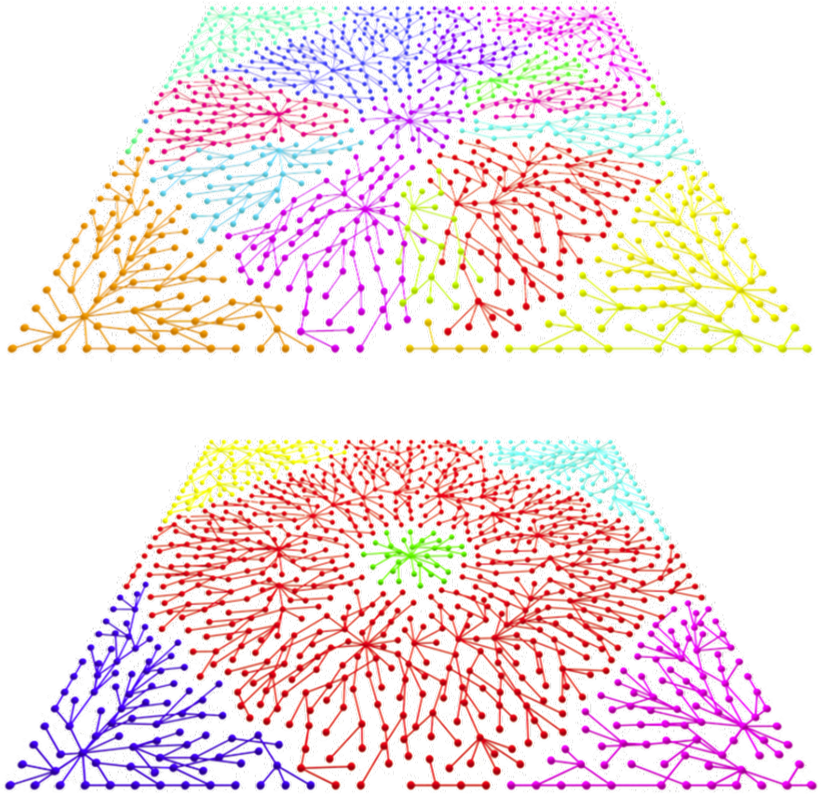
CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas
Computer Science Department
Stanford University

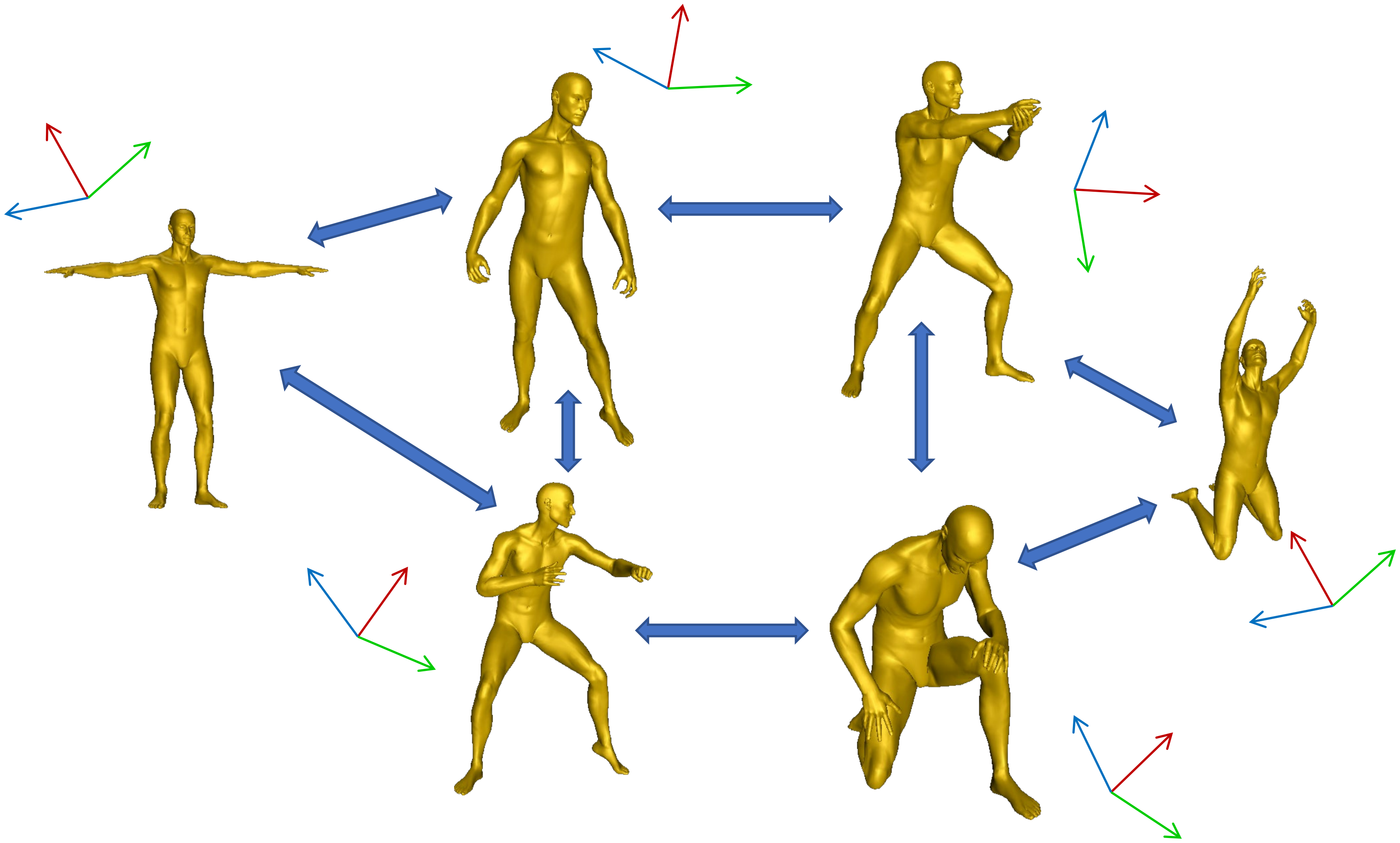
Lecture 17
26 May 2021



Leonidas Guibas Laboratory
Geometric Computing

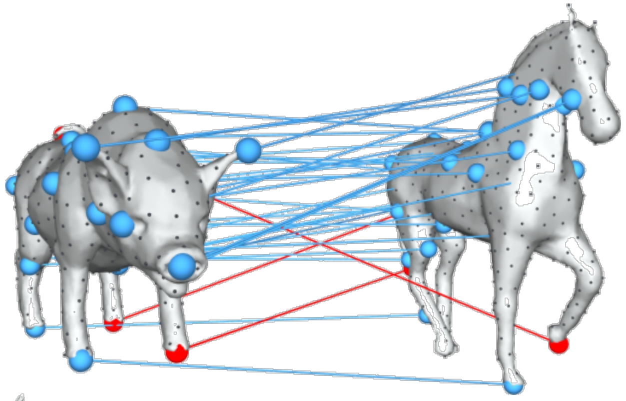
Last Time: Functional Map Networks

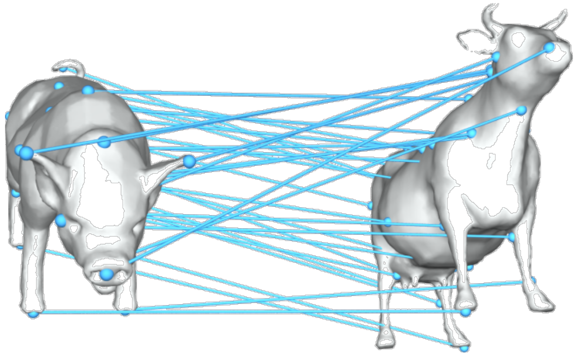
Consistency in Map Networks for Related Data



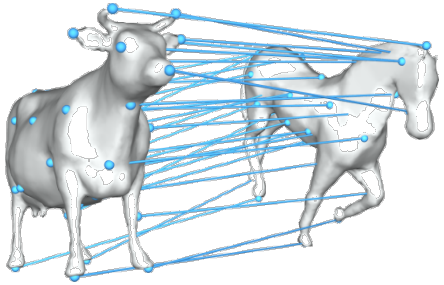
Networks of “samenesses”

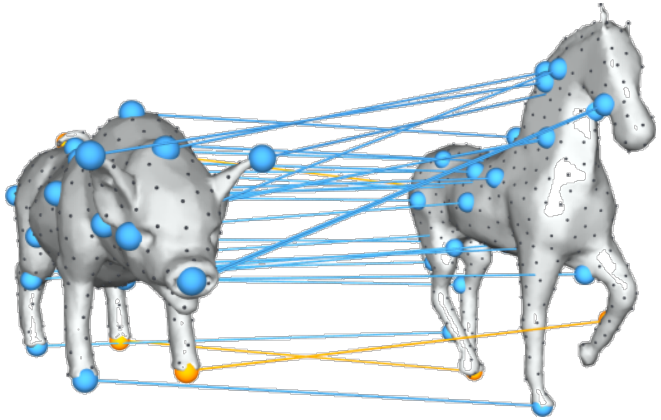
Fixing Maps





← Intermediate object →

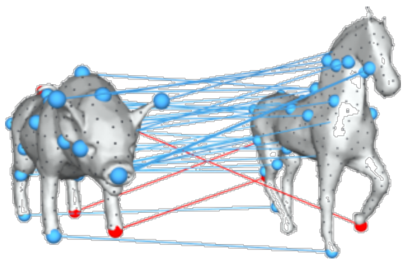




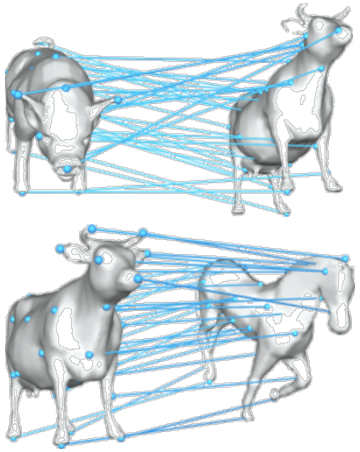
[Q. Huang, G. Zhang, L. Gao, S. Hu, A. Bustcher, and L. Guibas, 2012]

Cycle-Consistency ≡ Low-Rank

- In a functional map network, commutativity, path-invariance, or cycle-consistency are equivalent to a low rank or semidefiniteness condition on a big mapping matrix



$$X = \begin{pmatrix} I_m & X_{1,2} & \cdots & X_{1,n} \\ X_{1,2} & I_m & \cdots & \cdots \\ \vdots & \vdots & I_m & X_{(n-1),n} \\ X_{n,1} & \vdots & X_{n,(n-1)} & I_m \end{pmatrix}.$$



- Conversely, such a low-rank condition can be used to
  - regularize and clean up functional maps
  - extract shared structure

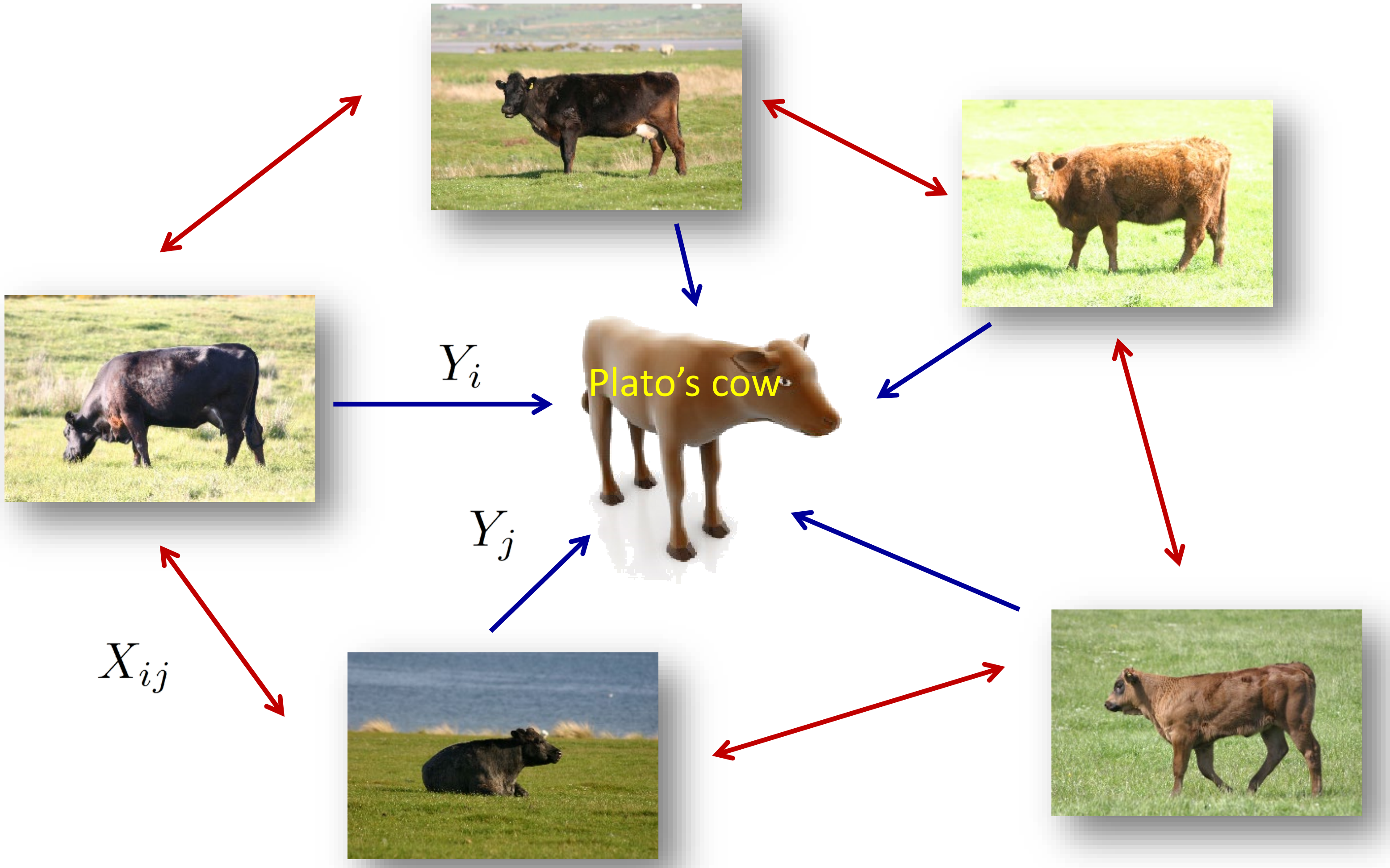
Map Synchronization by Matrix Factorization

$$X = \begin{bmatrix} I_m & X_{12} & \cdots & X_{1n} \\ X_{21} & I_m & \cdots & \vdots \\ \vdots & \vdots & \ddots & X_{n-1,n} \\ X_{n1} & \cdots & X_{n,n-1} & I_m \end{bmatrix}$$

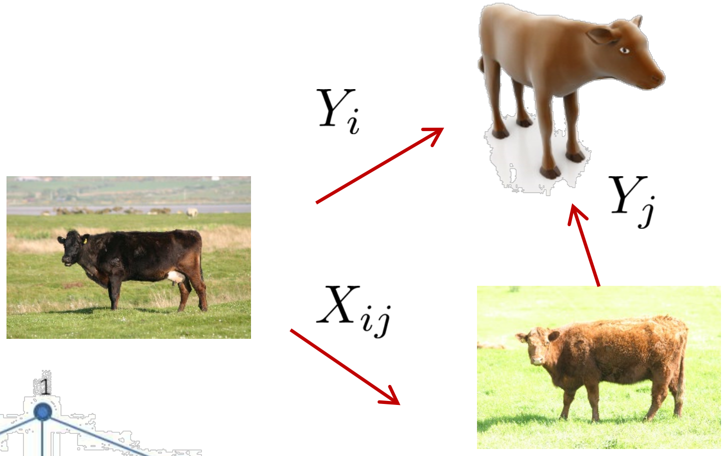
$$X_{ij} = X_{j1} X_{i1}^T$$

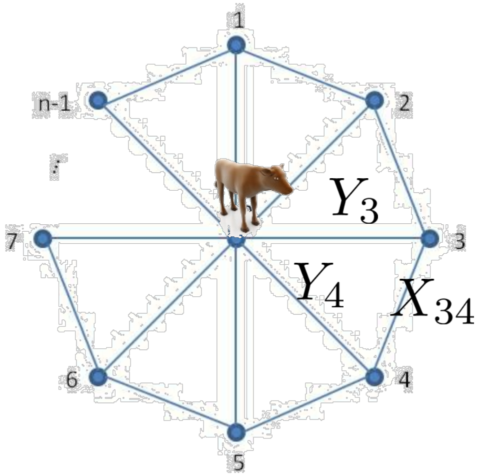
$$= \begin{bmatrix} I_m \\ \vdots \\ X_{n1} \end{bmatrix} \begin{bmatrix} I_m & \cdots & X_{n1}^T \end{bmatrix}$$

Image Co-Segmentstion



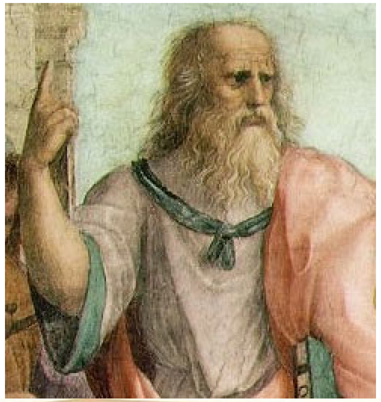
- Plato's allegory of the cave: a latent space

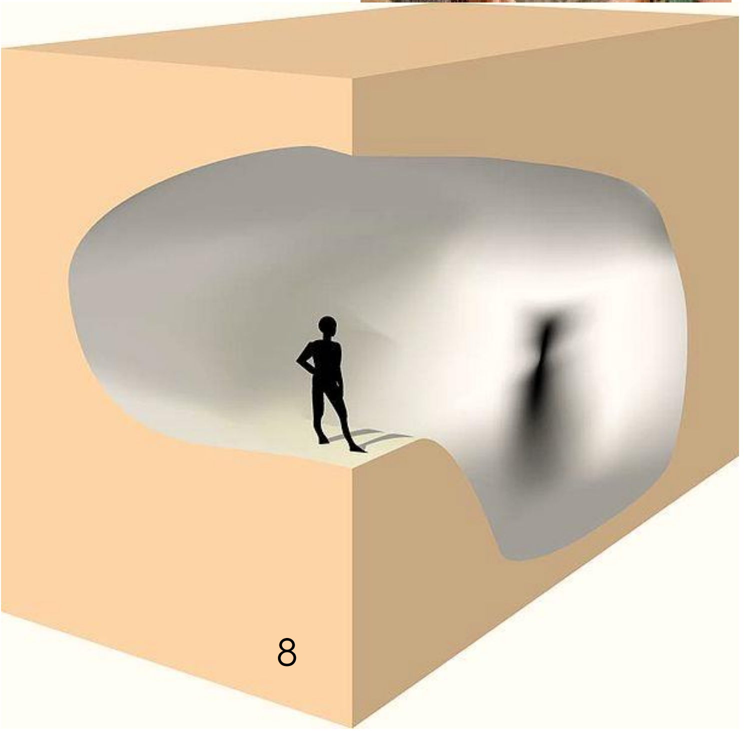




$$X_{ij} \approx Y_j^{-1} Y_i$$

X 30x30, *Y* 30x20





Generating Consistent Segmentations

- Two objectives for segmentation functions
  - consistent under functional map transportation

$$f^{\mathrm{map}} = \sum_{(i,j)\in\mathcal{G}} w_{ij} \|X_{ij}\mathbf{f}_i - \mathbf{f}_j\|_{\mathcal{F}}^2$$

consistent

We look for network fixed points!

  - and agreement with normalized cut scores:

$$f^{\mathrm{seg}} = \sum_{i=1}^{N} \mathbf{f}_i^T B_i^T L_i B_i \mathbf{f}_i$$

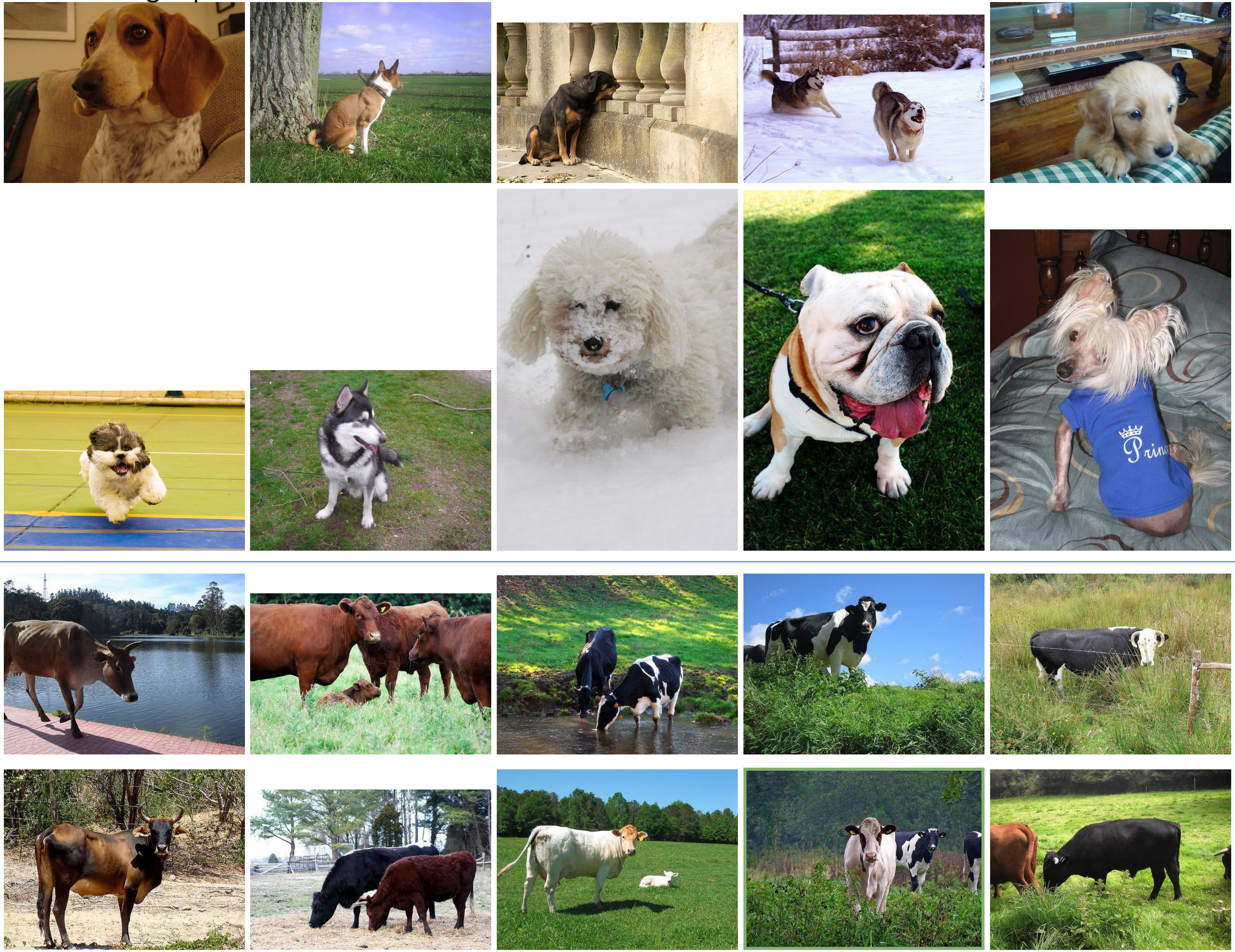
Easy to incorporate labeled images with ground truth segmentation

- Joint optimization:

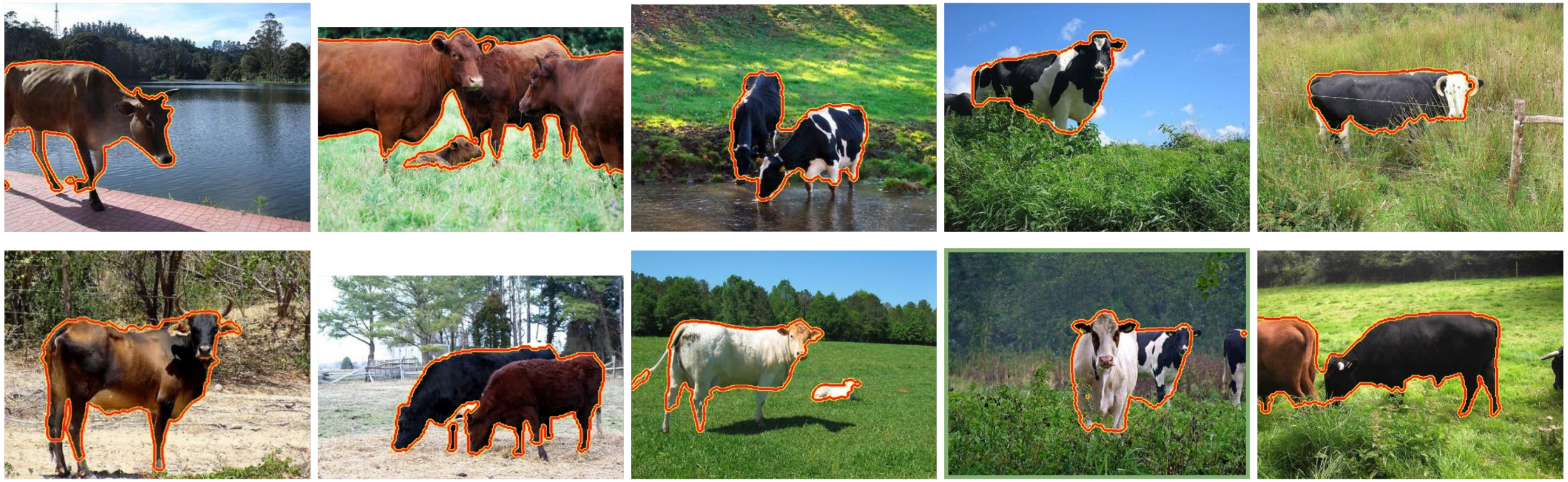
$$\min\ f^{\mathrm{seg}} + \gamma f^{\mathrm{map}} \quad s.t. \sum_{i=1}^{N} \|\mathbf{f}_i\|^2 = 1$$

Eigen-decomposition problem

PASCAL: 10 images per class are shown

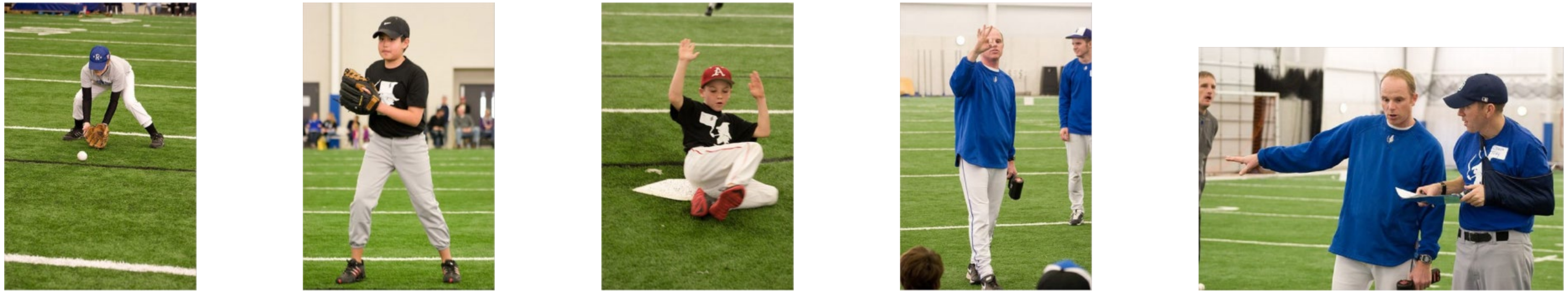


PASCAL: 10 images per class are shown



Apple + picking

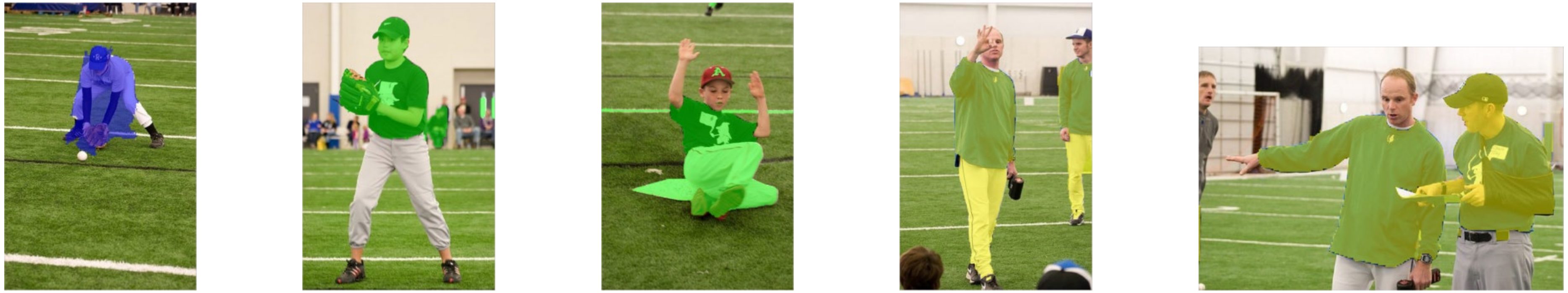
Baseball + kids



Butterfly + blossom

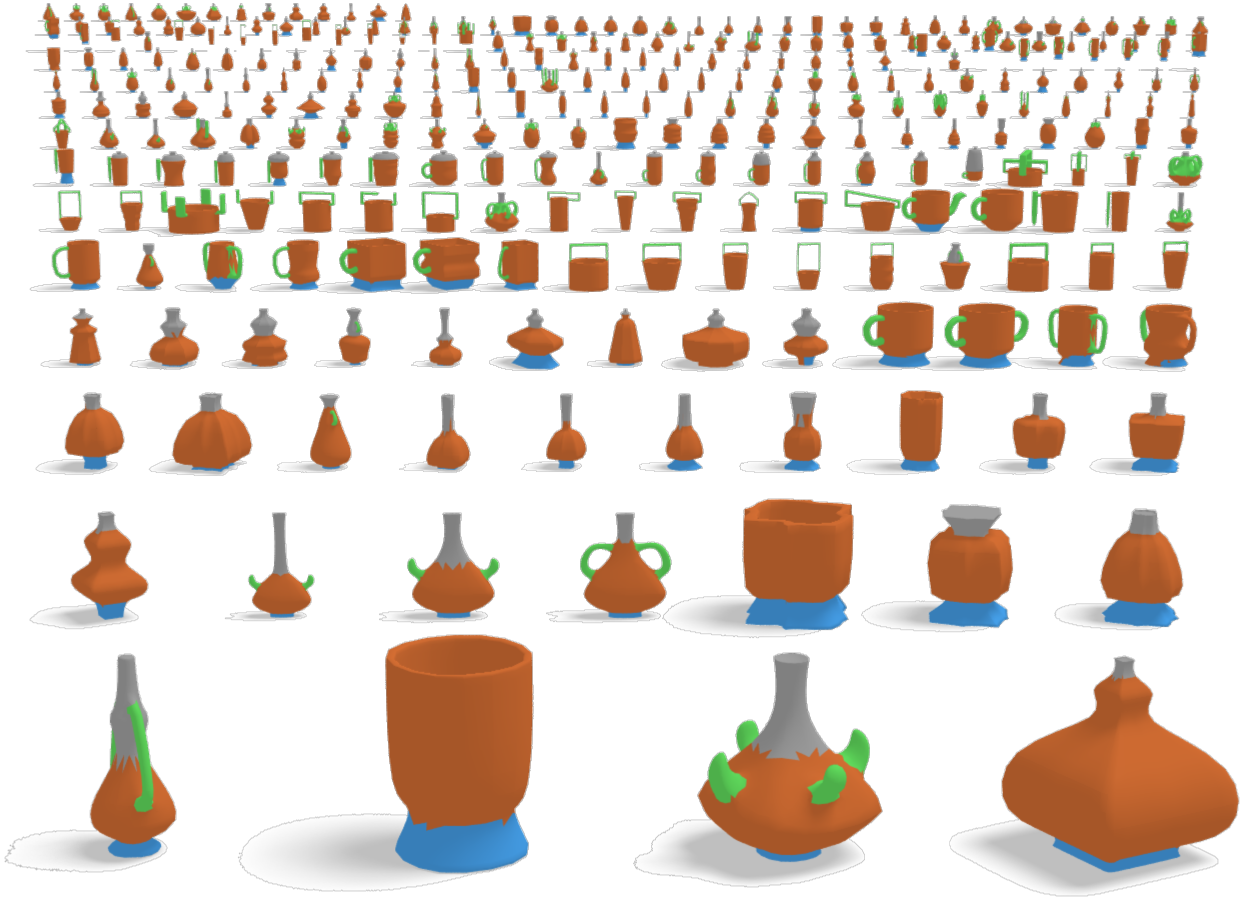
Apple + picking (red: apple bucket; magenta: girl in red; yellow: girl in blue; green: baby; cyan: pumpkin.)

Baseball + kids (green: boy in black; blue: boy in grey; yellow: coach.)

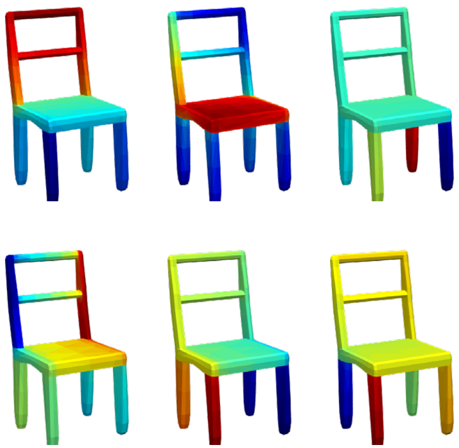


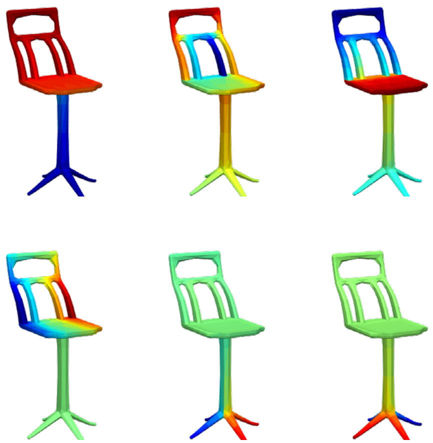
Butterfly + blossom (green: butterfly in orange; yellow: butterfly in yellow; cyan: red flower.)

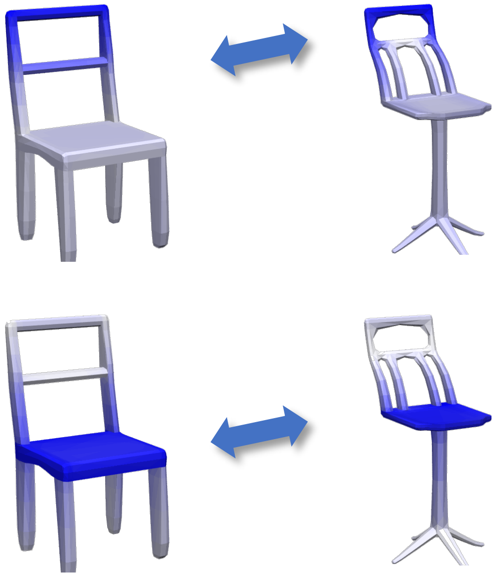
Consistent Shape Segmentation

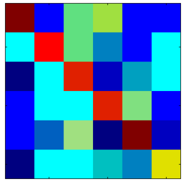


Start From Noisy Shape Descriptor Correspondences









Lift to
functional form

$$C_i X_{ij} \approx D_j$$

C_i • • • D_i

Joint Map Optimization

- Step 1: Convex low-rank recovery using robust PCA – we minimize over all *X*

convex!

trace norm
$\|X\|_\star = \sum_i \sigma_i(X)$

$$X^\star = \lambda \|X\|_\star + \min_X \sum_{(i,j)\in\mathcal{G}} \|X_{ij} C_{ij} - D_{ij}\|_{2,1}$$

$\|A\|_{2,1} = \sum_i \|\vec{a}_i\|$

Dual ADMM

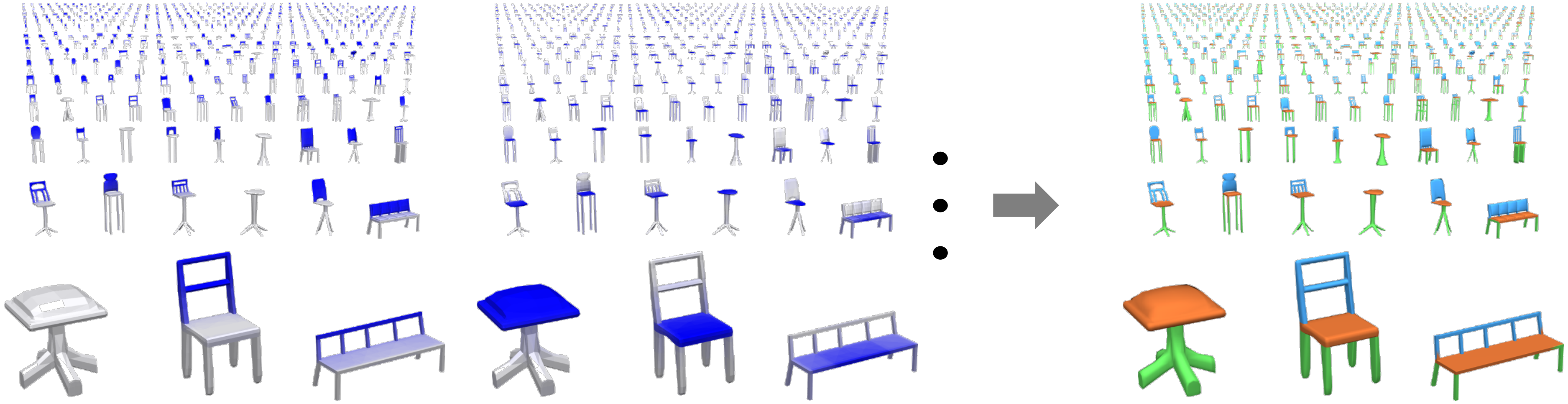
- Step 2: Perturb the above *X* to force the factorization

$$\sum_{1\le i,j\le N} \|X^\star_{ij} - Y_j^+ Y_i\|_F^2 + \mu \sum_{i=1}^{N} \sum_{1\le k<l\le L} (\mathbf{y}_{ik}^T \mathbf{y}_{il})^2$$

Non-linear least squares
Gauss-Newton descent

The Y_i give us the desired latent spaces

Consistent Shape Segmentation



Via 2nd order MRF on each shape independently

Detour: ZoomOut

ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, S. Melzi, J. Ren, A. Sharma, E. Rodolà, P. Wonka, M. O., SIGGRAPH Asia 2019

Many Open Functional Maps Questions

1. How can we build up a functional map *progressively*?

2. Given a small functional map, can we use it to transfer *high frequency functions*?

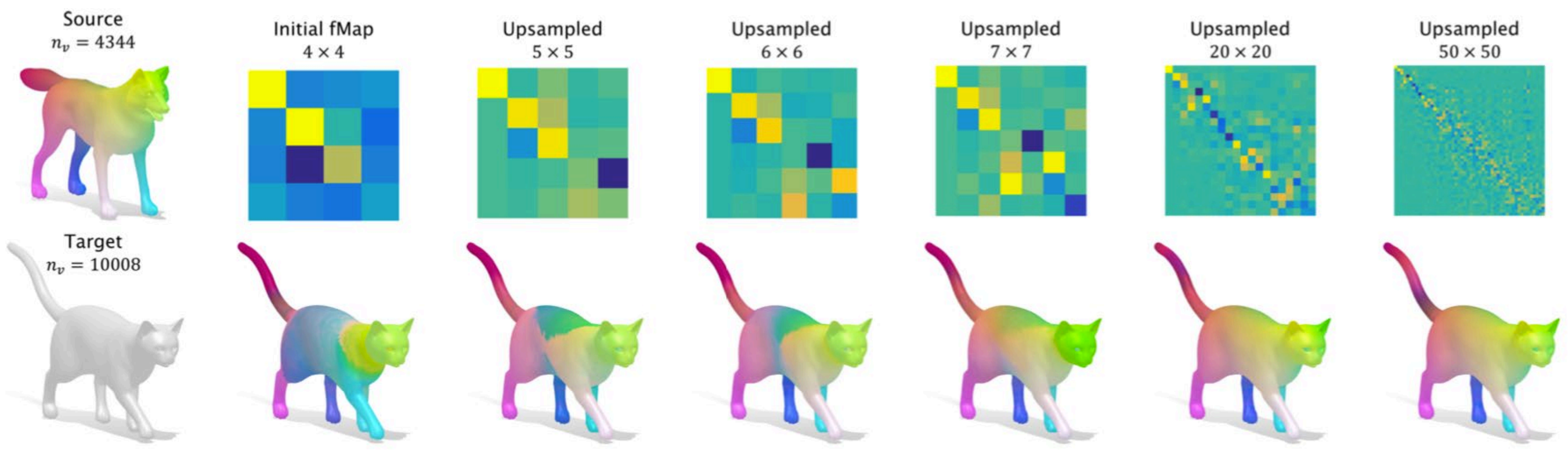
3. Simplify and speed-up *functional map refinement*?

ZoomOut

A two-lines-of-code algorithm:

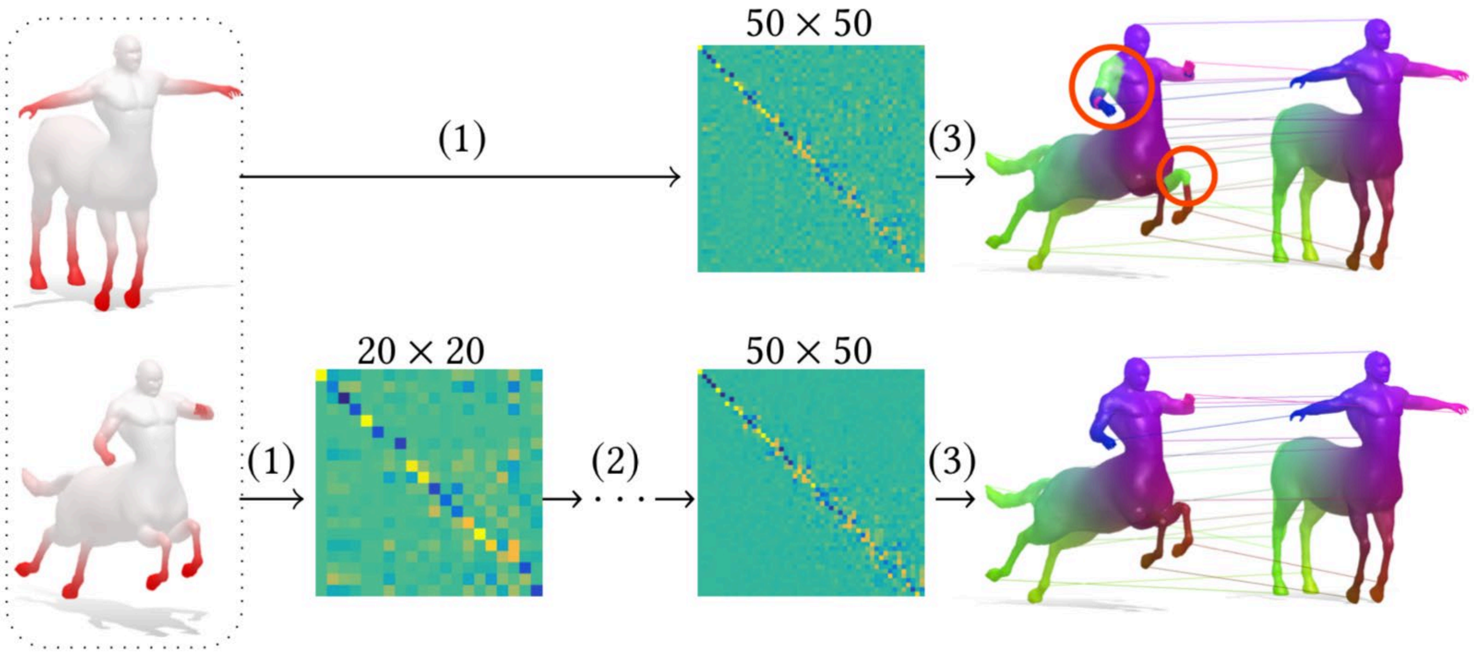
1) Given a functional map *C1* of size *k* x *k* convert it to a p2p map *T.*
2) Convert *T* to C2 of size (*k*+1) x (*k*+1)

Repeat for progressively larger *k*



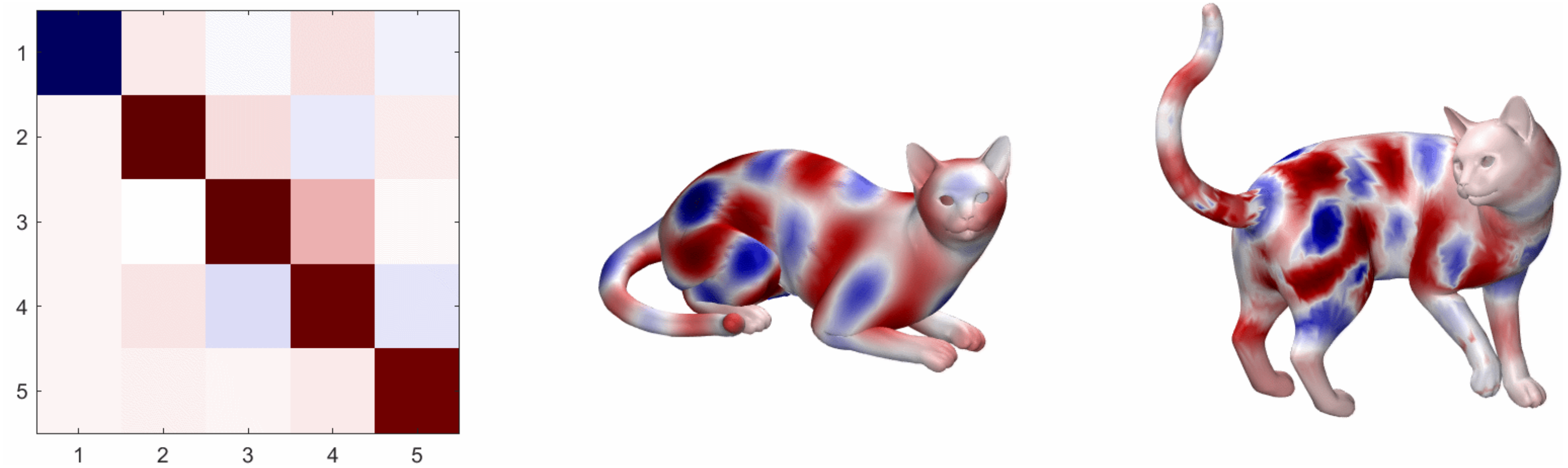
ZoomOut

Upsampling vs. computing directly:



ZoomOut – Results

From 5x5 to 50x50



ZoomOut – Results

From 20x20 to 120x120

ZoomOut – Rationale

Consider the optimization problem:

$$\min_{\mathbf{C}\in\mathcal{P}} E(\mathbf{C}), \text{ where } E(\mathbf{C}) = \sum_k \frac{1}{k} \|\mathbf{C}_k^T \mathbf{C}_k - I_k\|_F^2 .$$

$\mathbf{C}\in\mathcal{P}$: functional map arising from some pointwise map.

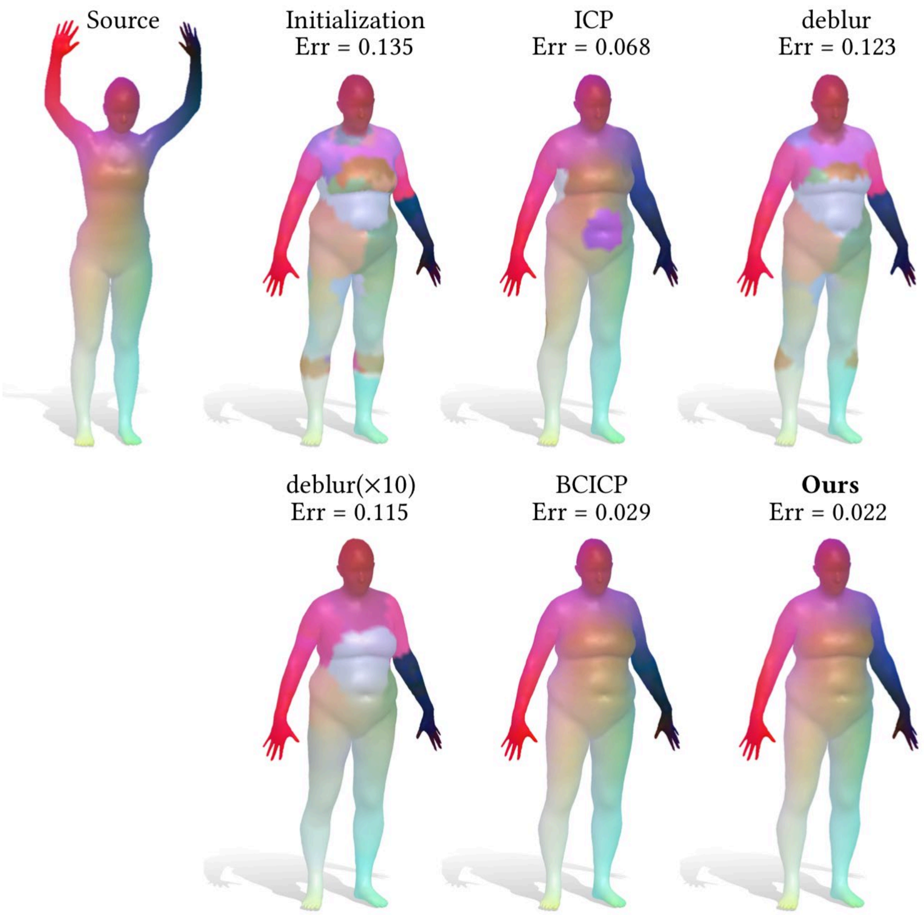
$\mathbf{C}_k$: leading principal $k \times k$ submatrix of $\mathbf{C}$.

Theorem:

$E(\mathbf{C}) = 0$ if and only if the point-to-point map is an isometry.

ZoomOut can be derived as a iterative method for solving this optimization problem.

ZoomOut – Results



Evaluated on:

- Intrinsic symmetry detection
- Complete matching
- Partial matching
- Function transfer

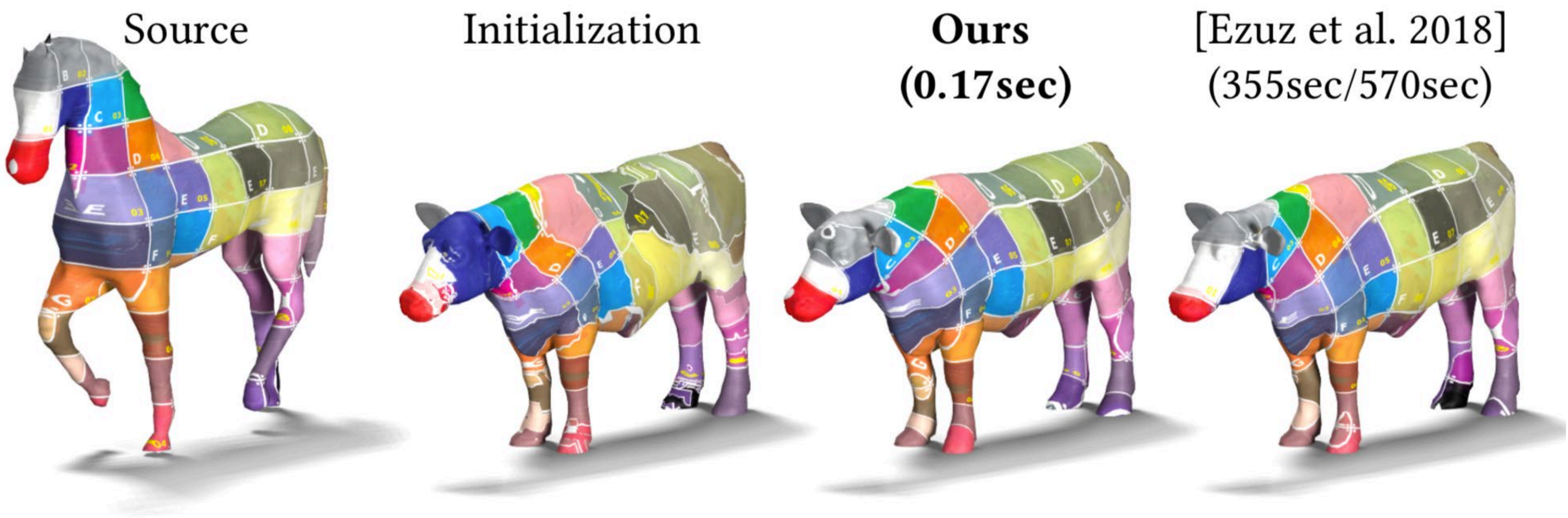
…

Compared against 14 baselines

Ours is 50-300x faster than state-of-the-art with higher accuracy

ZoomOut – Non-isometric

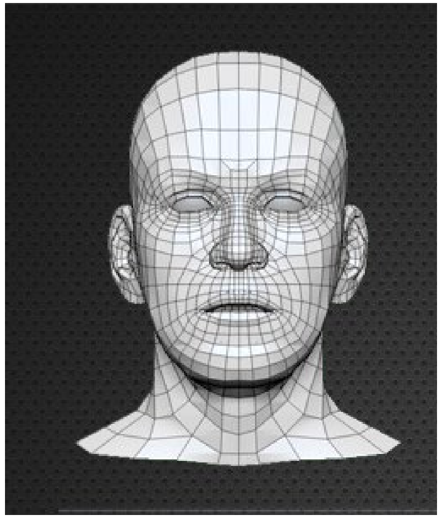
In some cases also works for non-isometric shapes



Today: Shape Differences and Variability

Object Shape, Material and Appearance Differences

What Exactly is a Shape Difference?

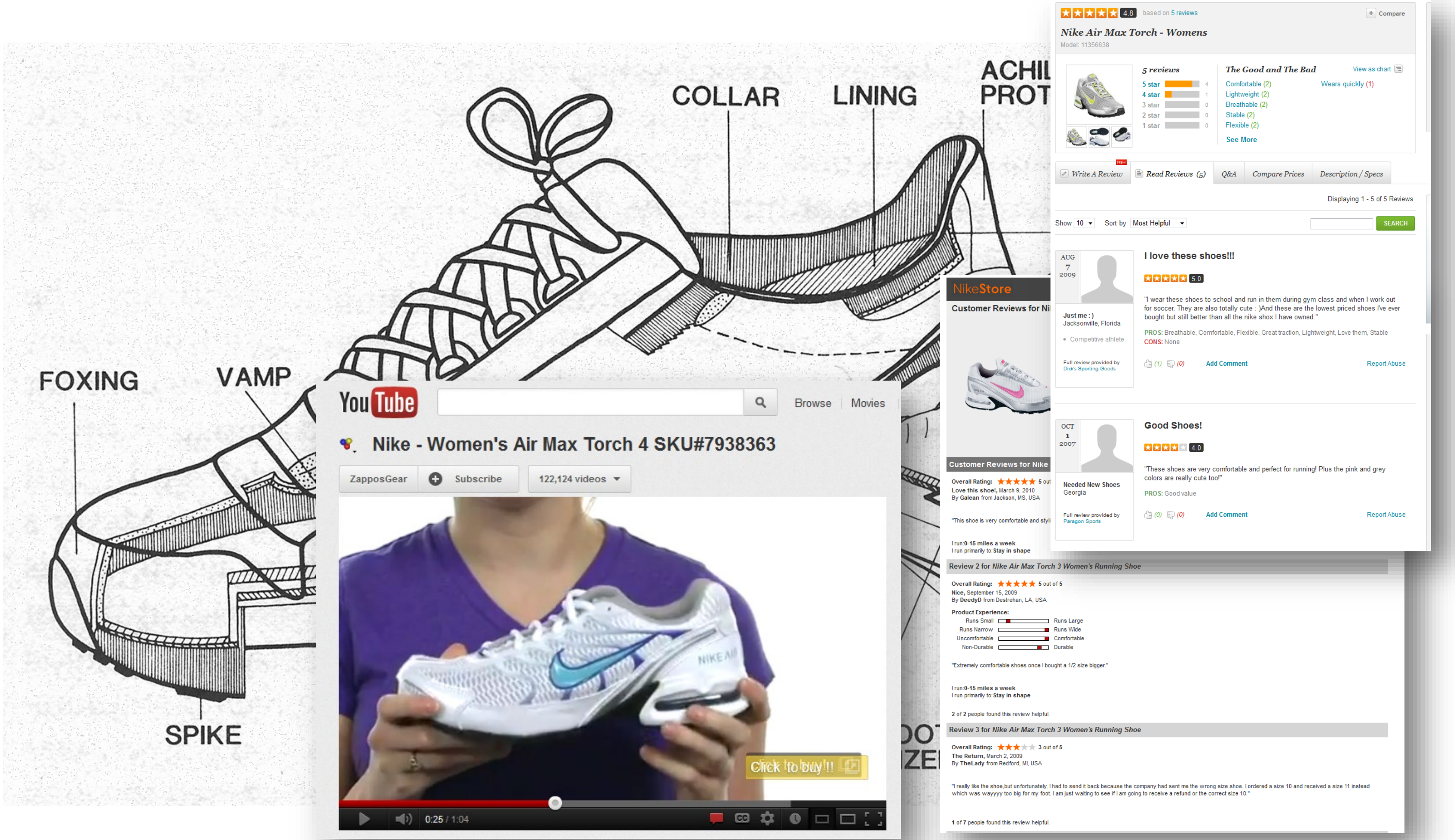


vs.

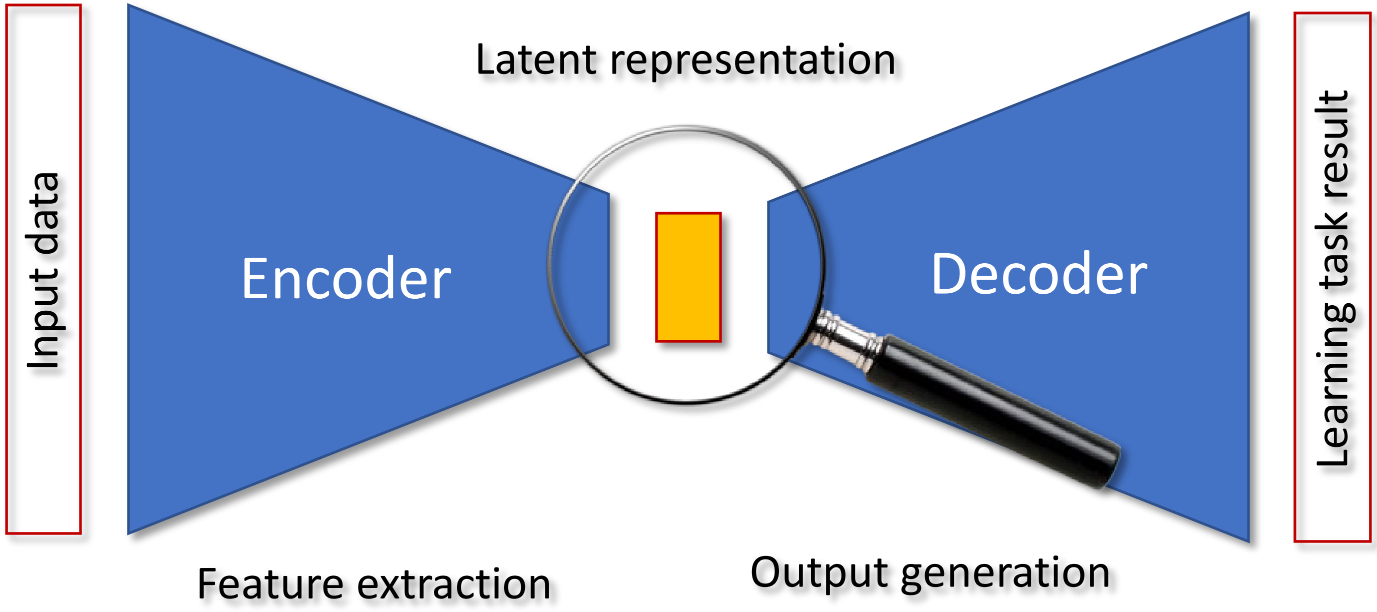
vs.

Making 3D shape differences first class citizens

Search Engines Based on Differences?

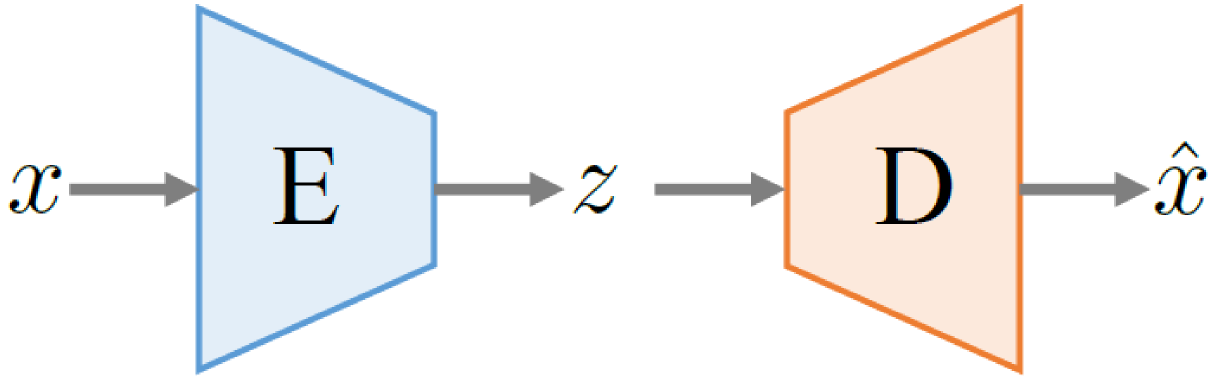


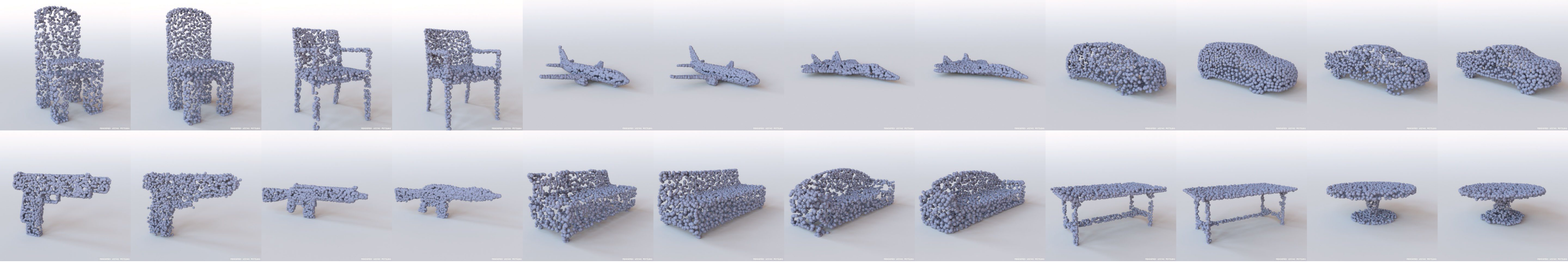
Latent Spaces in ML, Supervised or Not



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

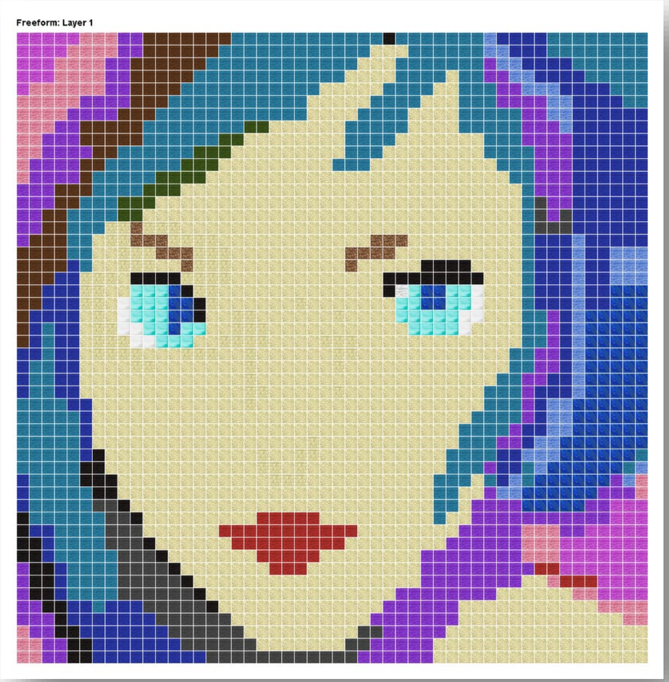
Point Cloud Auto-Encoders



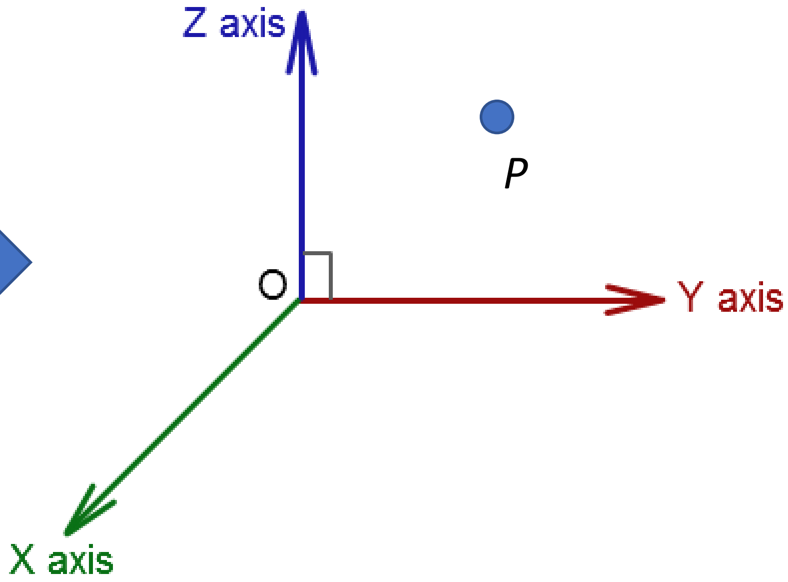


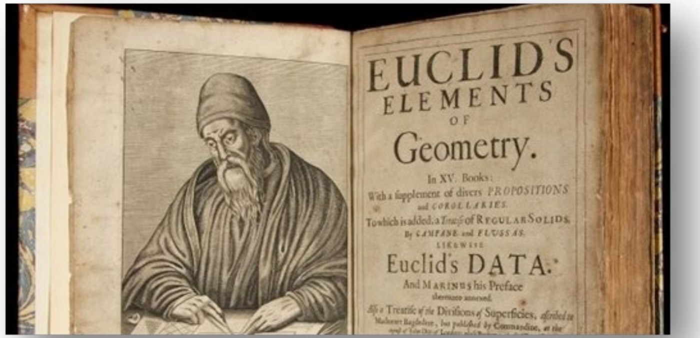
What Exactly is a Latent Space Representation?

- Input



- Latent code

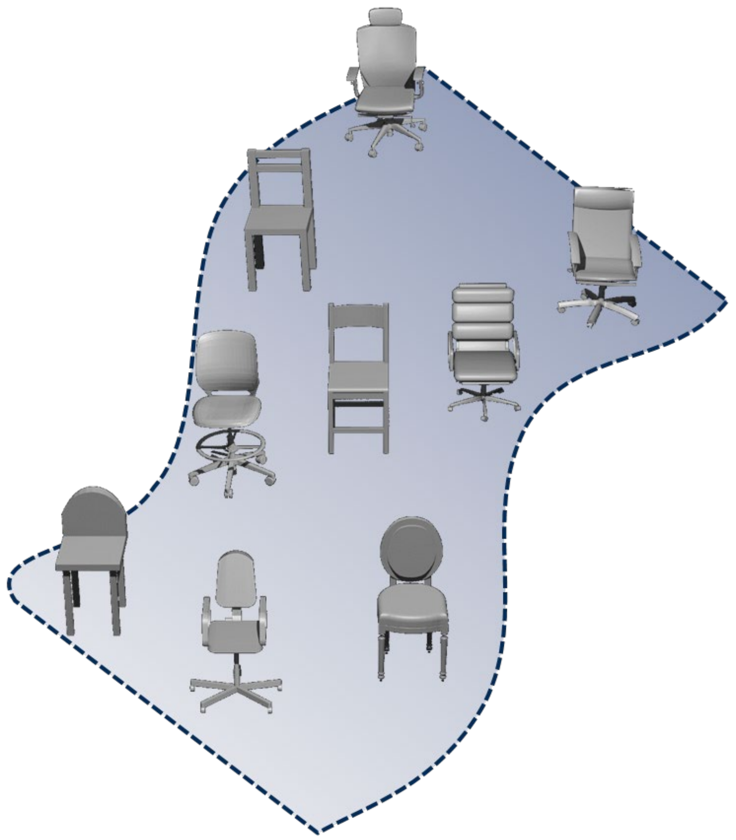




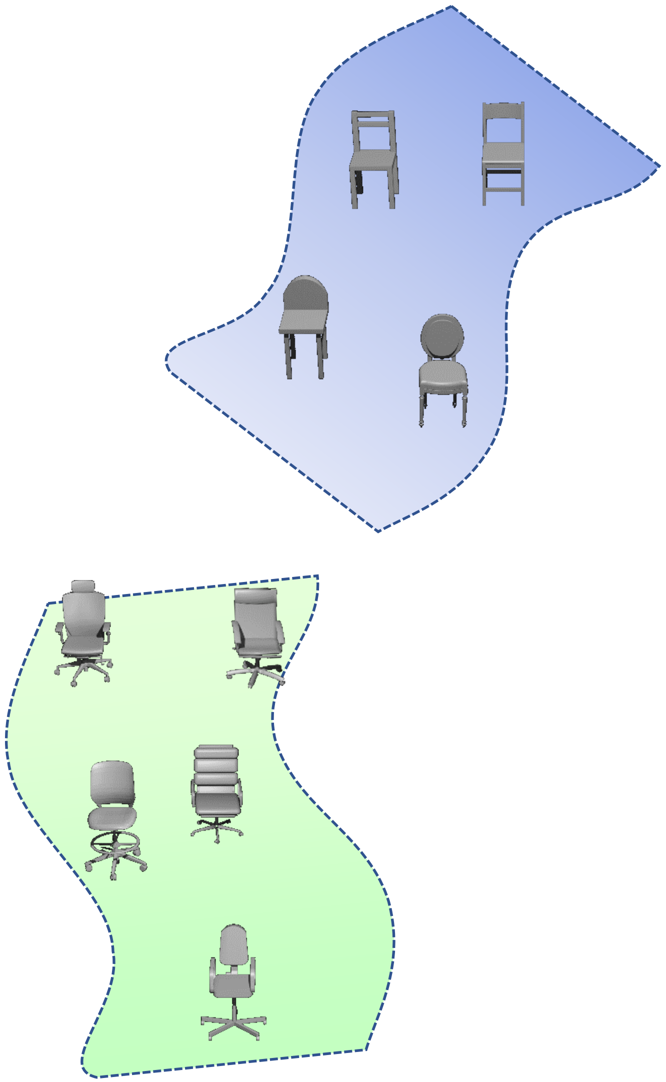
Is a Shape Difference Just a Vector in a Latent Space?

Continuous Shape Variability

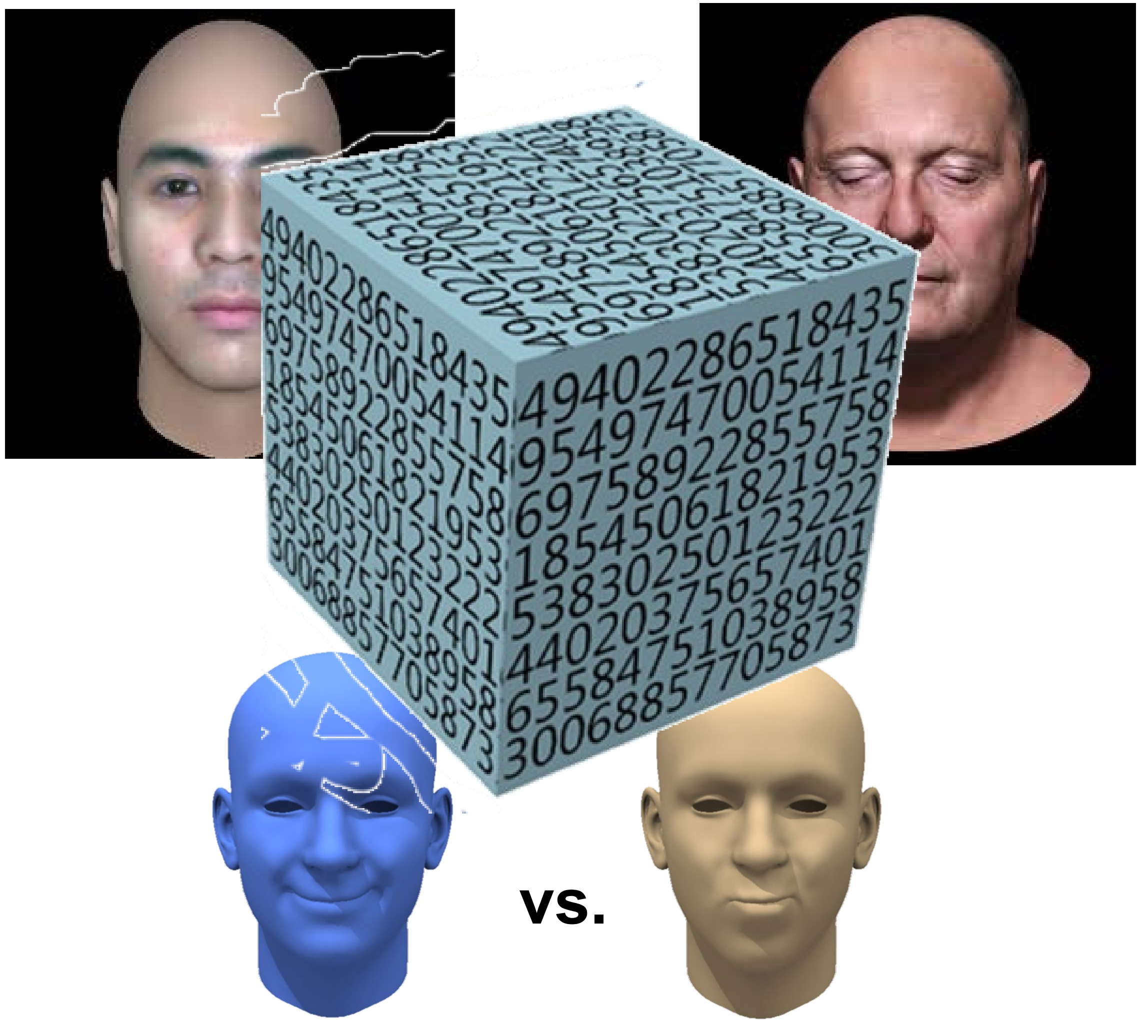
A chair manifold?



Combinatorial or Discrete Variability

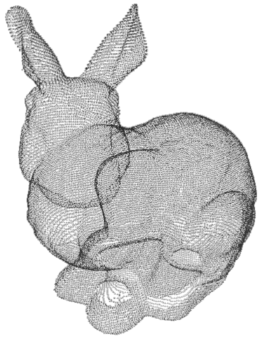


What Exactly is a Shape Difference?

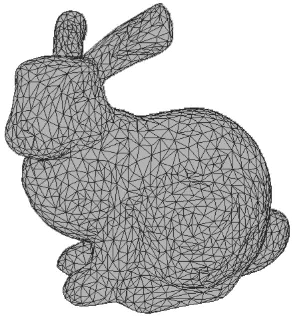


Where and how are the shapes different?

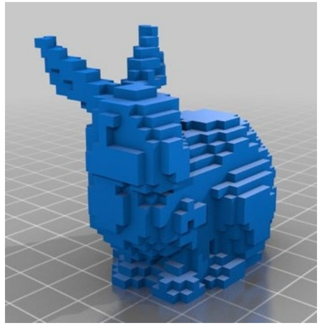
A Challenge: Multiple 3D Representations



Point Cloud



Mesh



Volumetric

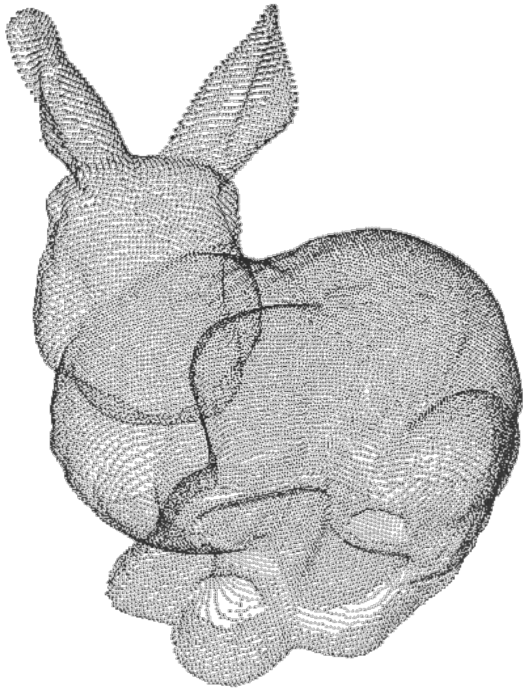
Projected View

RGB(D)

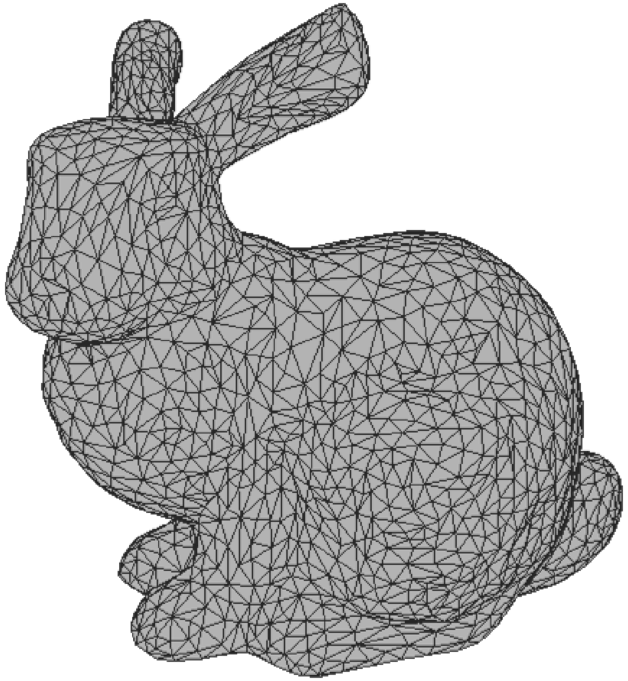
...

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

Underlying Shape Surface Discretizations



Point cloud



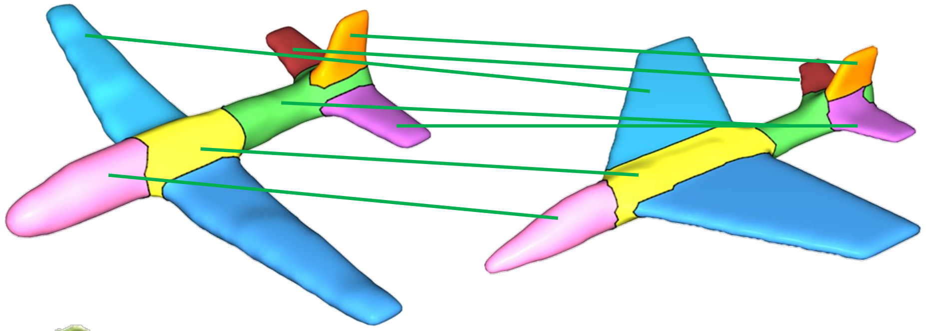
Mesh

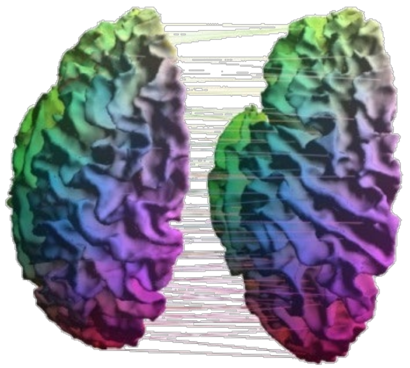
Must distinguish differences of the representations from differences of the shapes

Continuous Shape Differences Under a Map

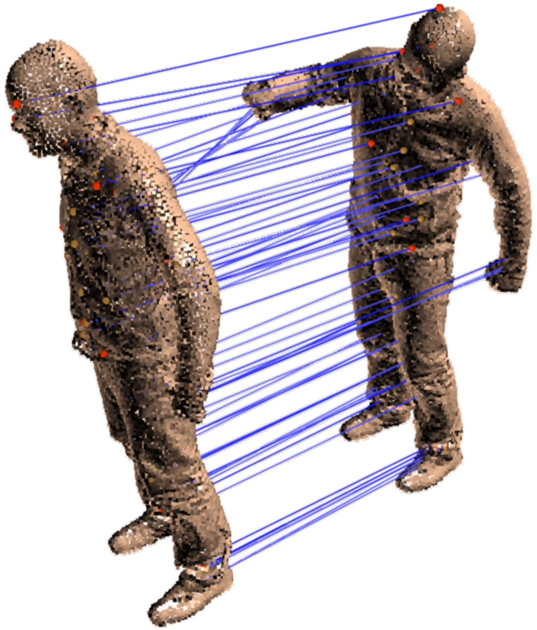
Surface Maps and Distortions

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

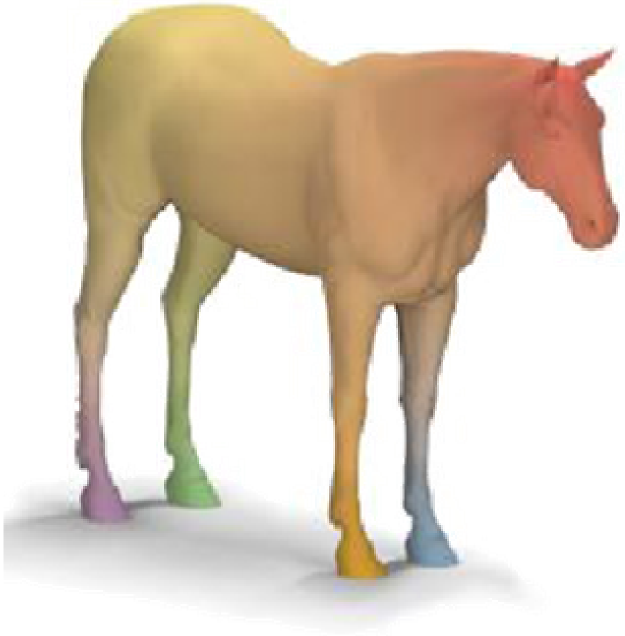




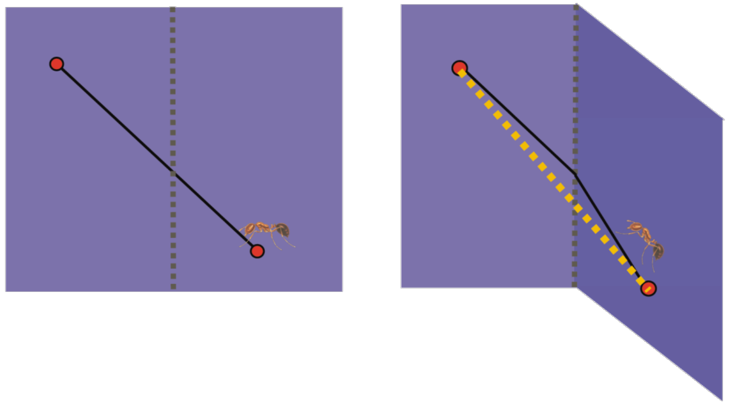
Yet the distortion estimates are forgotten as soon as the correspondences are computed

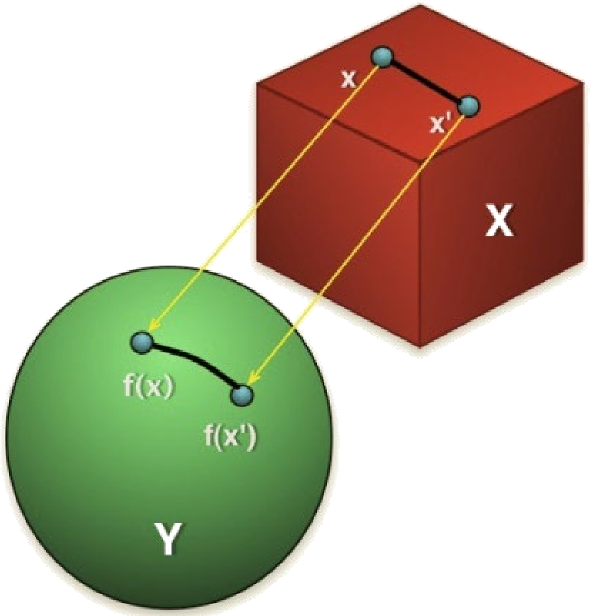


Subtlety 1: Correspondences at Multiple Scales



Subtlety 2: Intrinsic or Extrinsic Distances



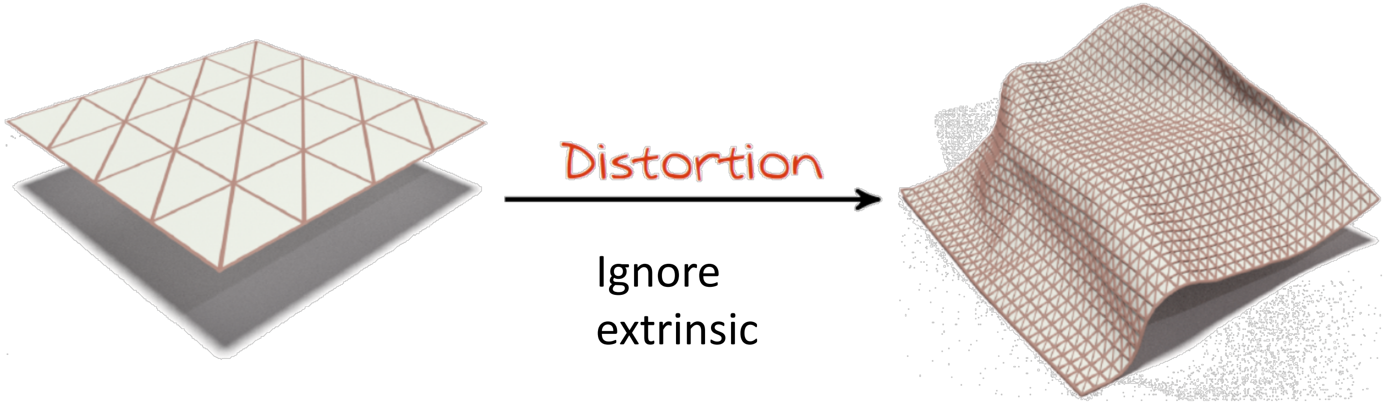


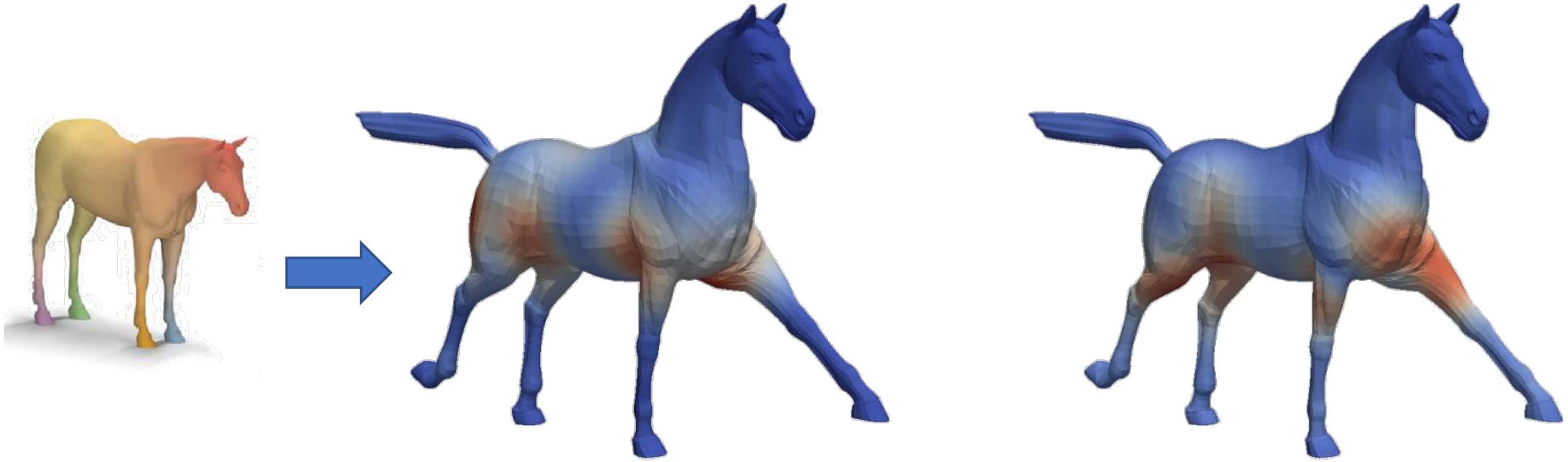
Measure on the surface or in the ambient space?

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

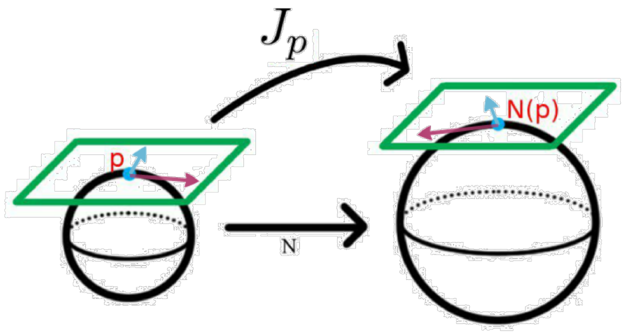


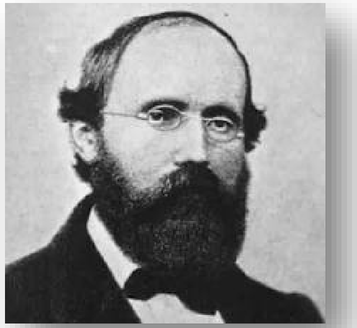


Intrinsic distortions: Area distortion Conformal distortion Length …

Classical Approach to Relating Intrinsic Metrics

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





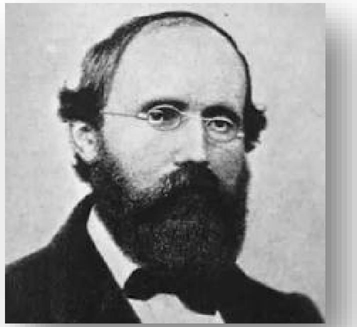
Riemann

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

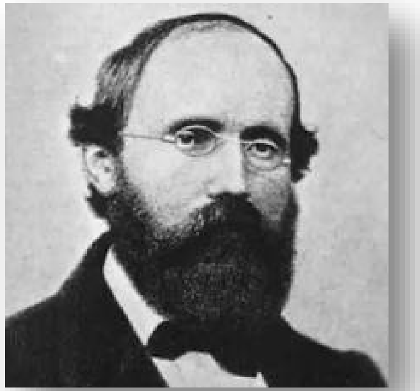
To measure distortions induced by a map, track how inner products of functions change after transporting.

The Art of Measurement

- A metric is defined by a functional inner product

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

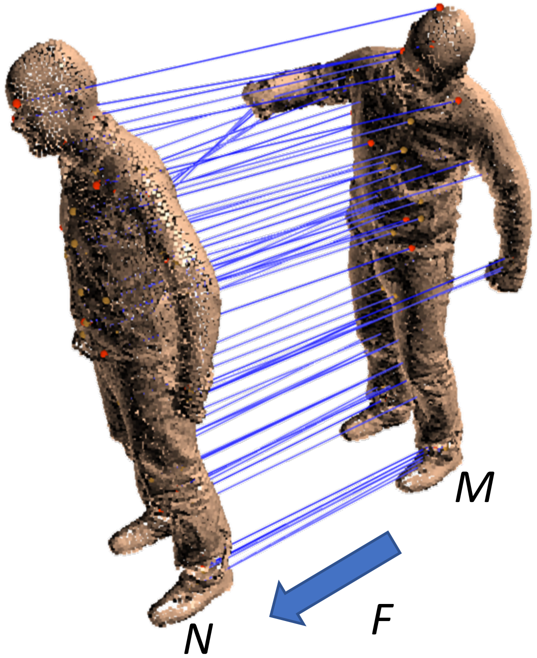
- So we can compare *M* and *N* by comparing



Riemann

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

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



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

Inner Products of Functions

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

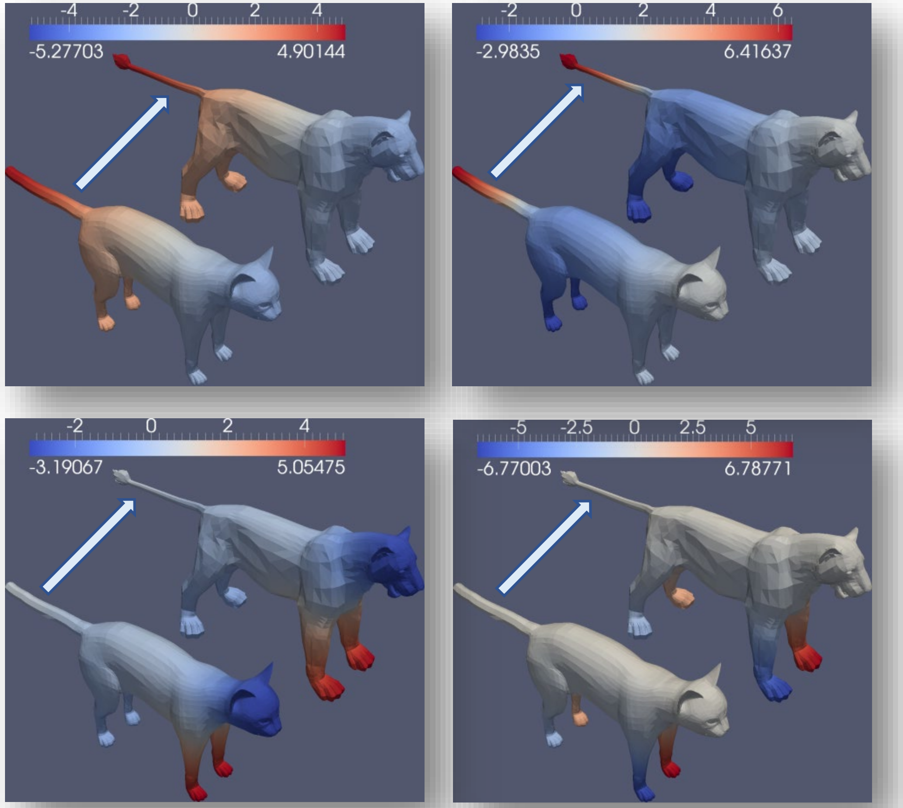
$$= \sum_{ij} f_i g_j \langle h_i(x), h_j(x) \rangle$$

$$= f^\top A g$$

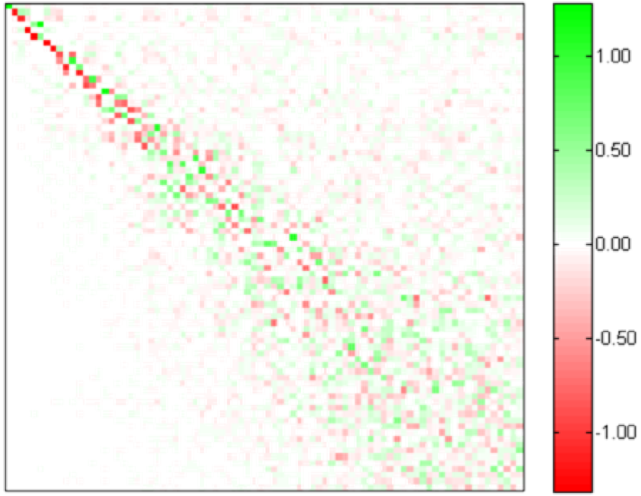
Area weights matrix

Starting from a Functional Map F

from cat to lion



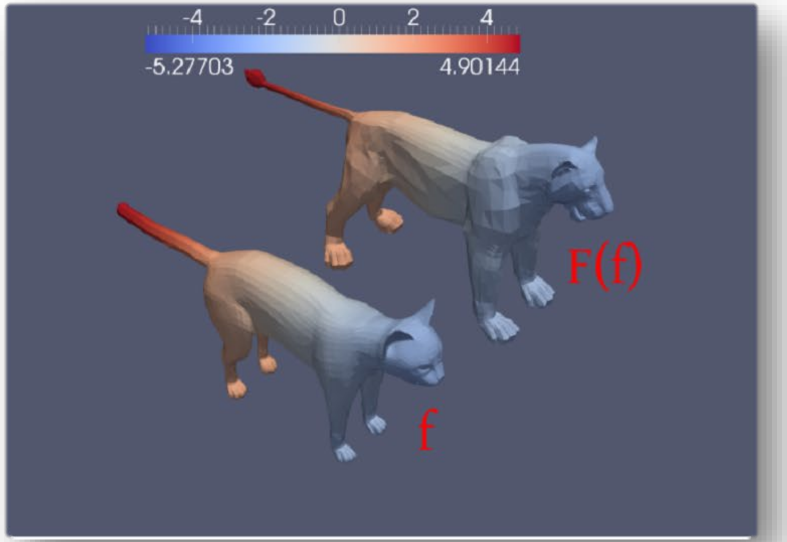
Functions on cat are transferred to lion using F

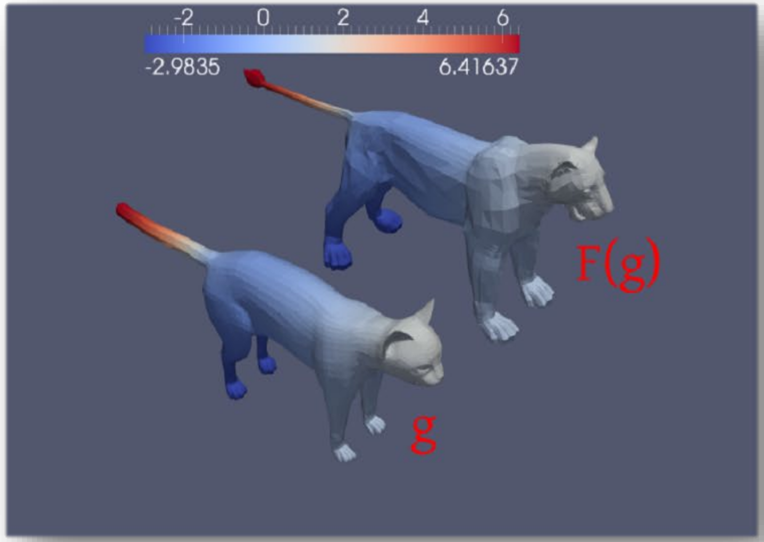


F is a linear operator (matrix)

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

Measurement Discrepancies





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

after

before

Measurement Discrepancies

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

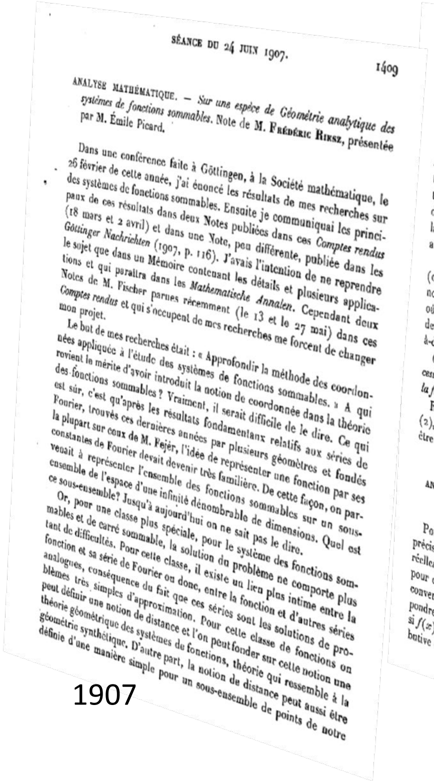
after

before

Both can be considered as inner products on the cat

The Universal Compensator

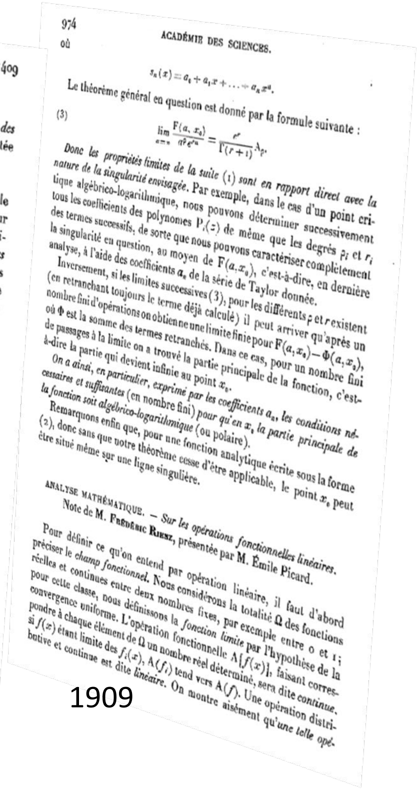
Comptes Rendus Hebdomadaires des
Séances de l'Académie des Sciences de Paris



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.

1907



974 ACADÉMIE DES SCIENCES.

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

1909

Riesz Representation Theorem

There exists a linear operator

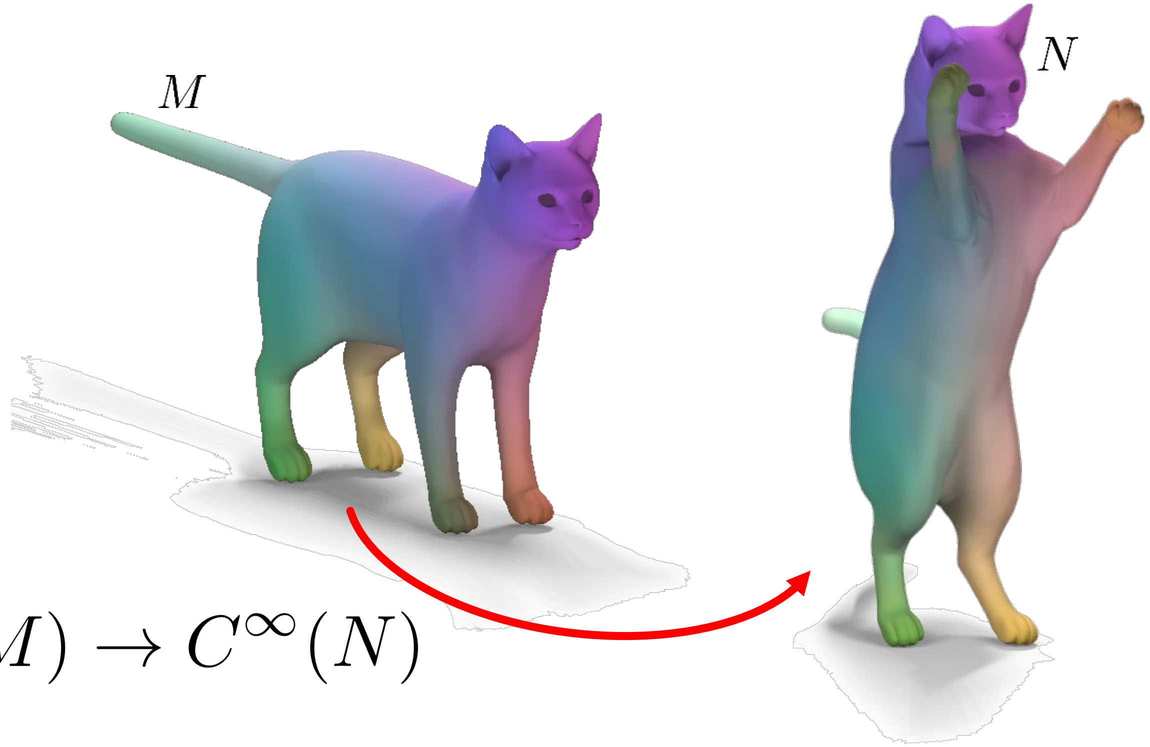
$$V : L^2(cat) \to L^2(cat)$$

such that

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

Frigyes Riesz

Riesz Representation Theorem



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

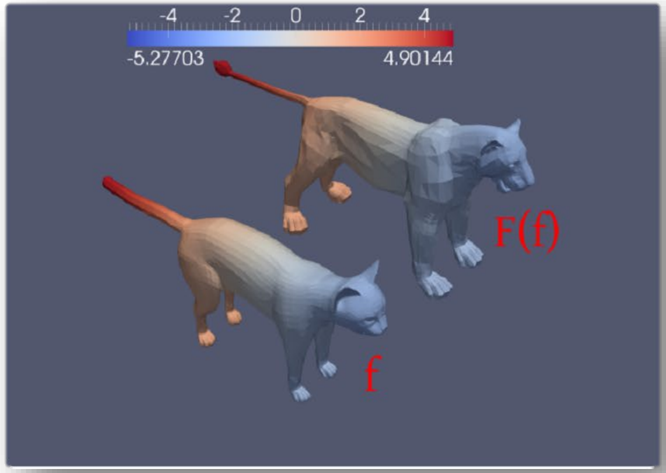
$$\exists V \text{ s.t.}$$
$$\langle F(f), F(g) \rangle^N = \langle f, V(g) \rangle^M \ \forall f, g$$

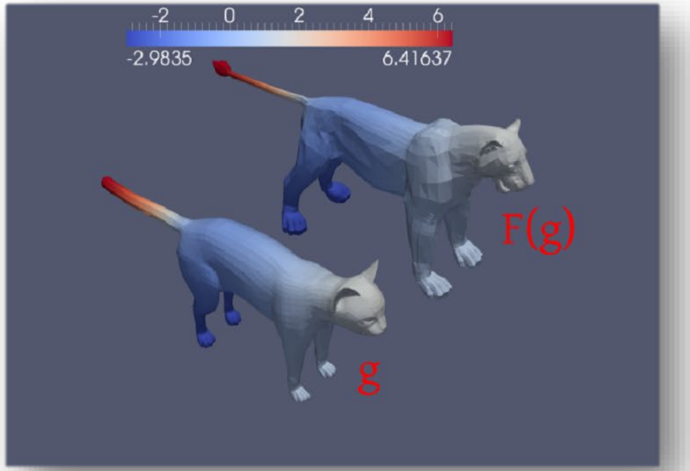
Sanity Check

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

Area-Based Shape Difference:

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





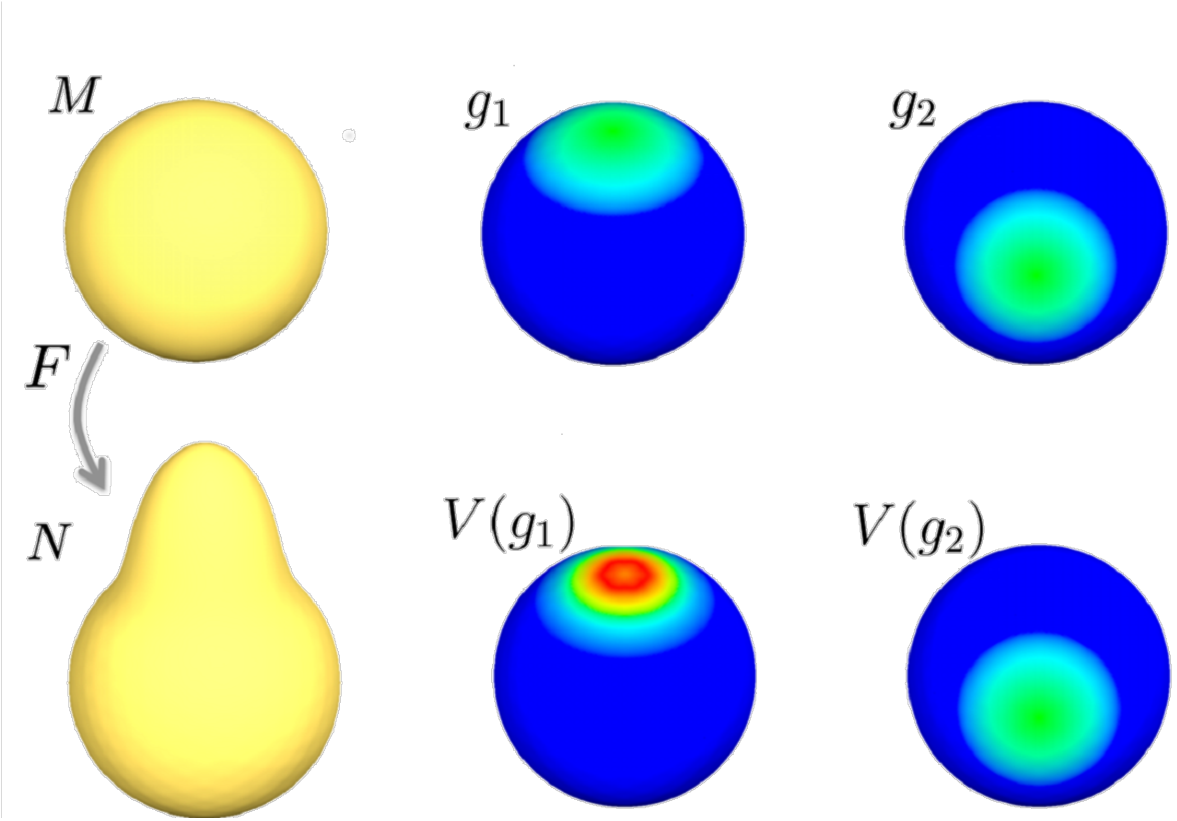
$$\int_{lion} F(f)F(g) \neq \int_{cat} fg$$

↓

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

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

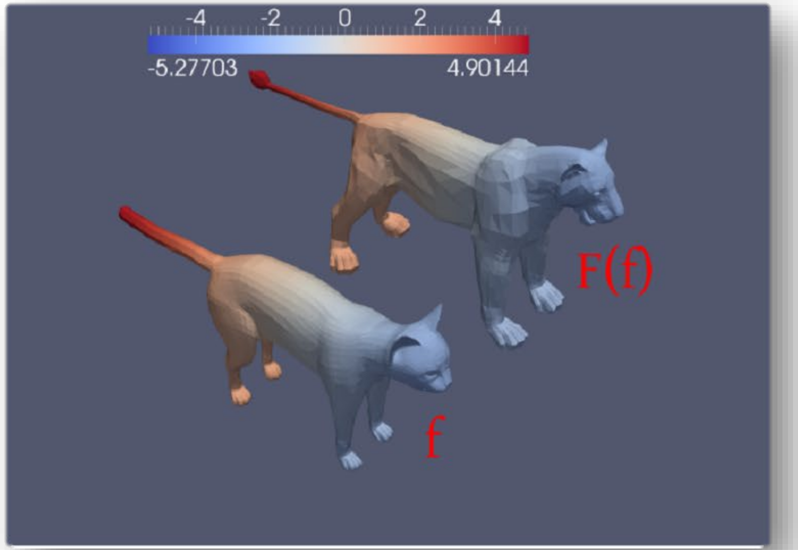
Consider a different inner-product of functions ...

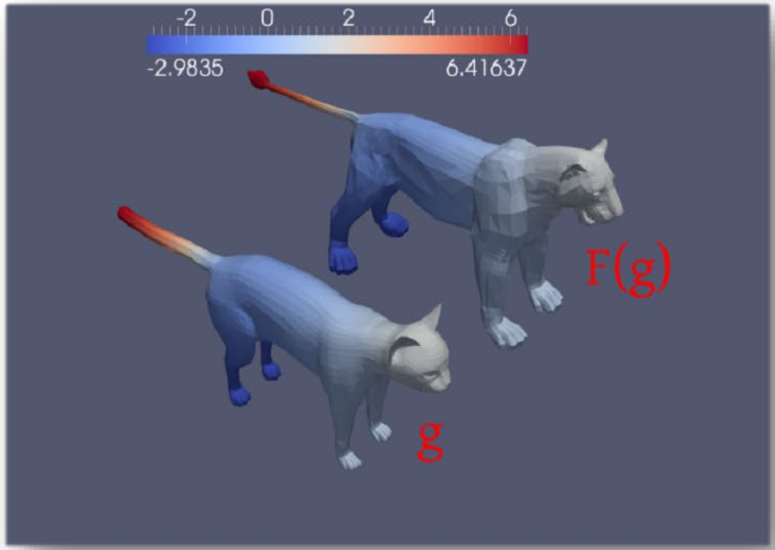
get information about conformal distortion

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

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

Conformal Shape Difference *R*



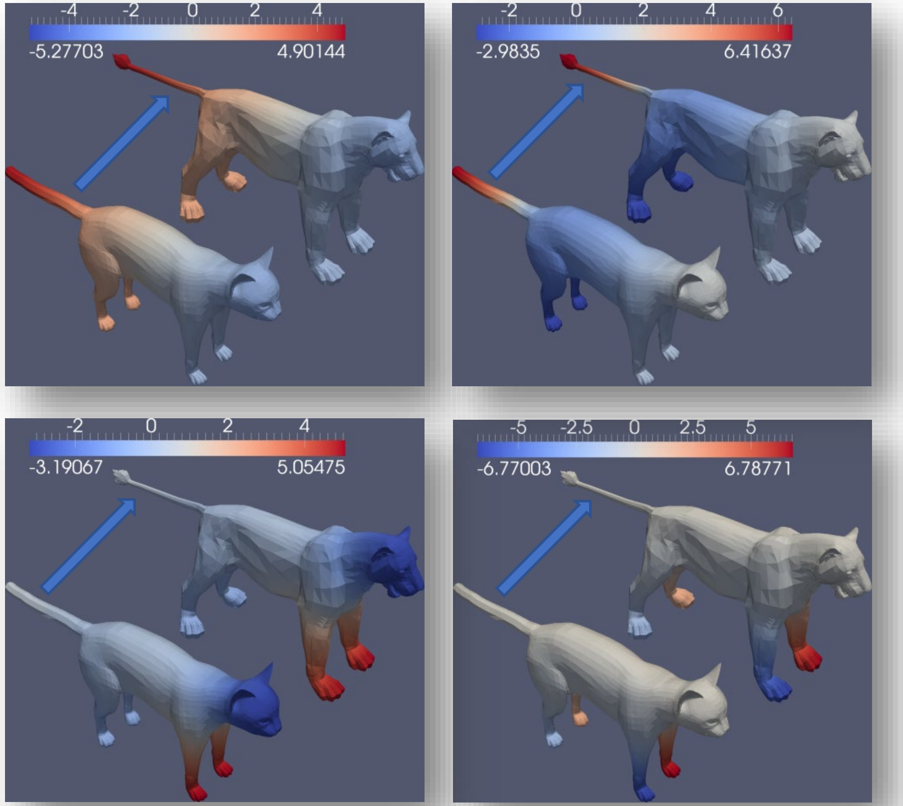


$$\int_{lion} \nabla F(f) \cdot \nabla F(g) \neq \int_{cat} \nabla f \cdot \nabla g \quad \forall f, g$$

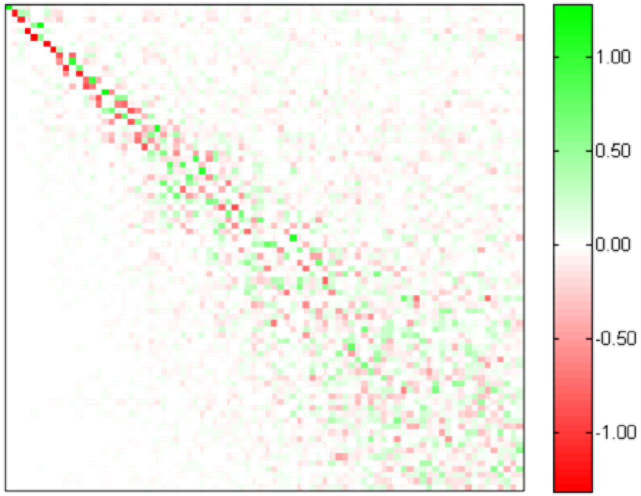
$$\int_{lion} \nabla F(f) \cdot \nabla F(g) = \int_{cat} \nabla f \cdot \nabla R(g)$$

Input: Functional Map *F*

from cat to lion



Functions on cat are transferred to lion using *F*

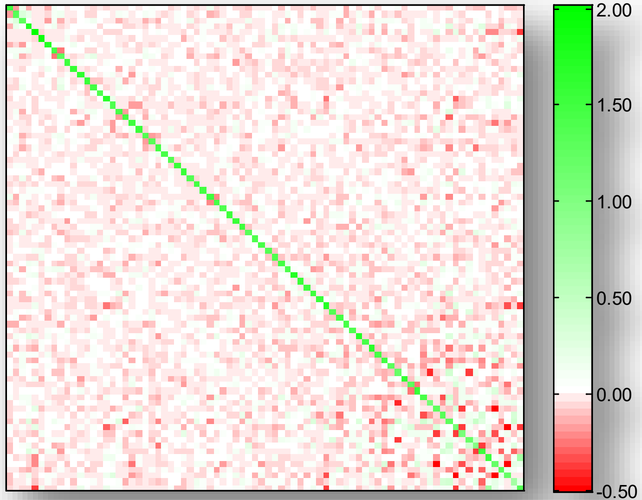


F **is a linear operator (matrix)**

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

Shape Difference Operators

V – area-based shape difference

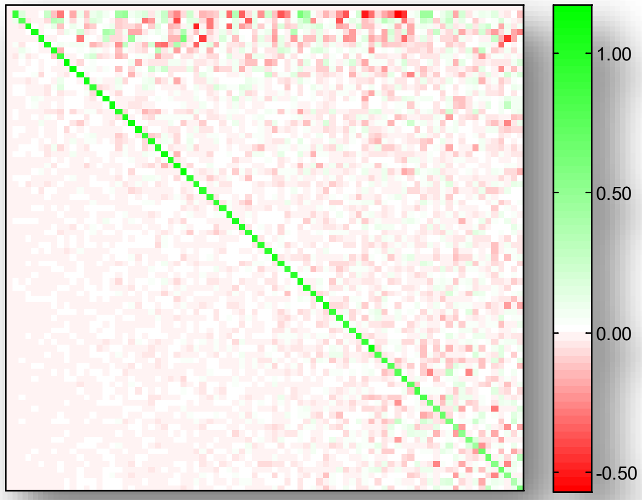


linear operator (matrix)

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

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

R – conformal shape difference

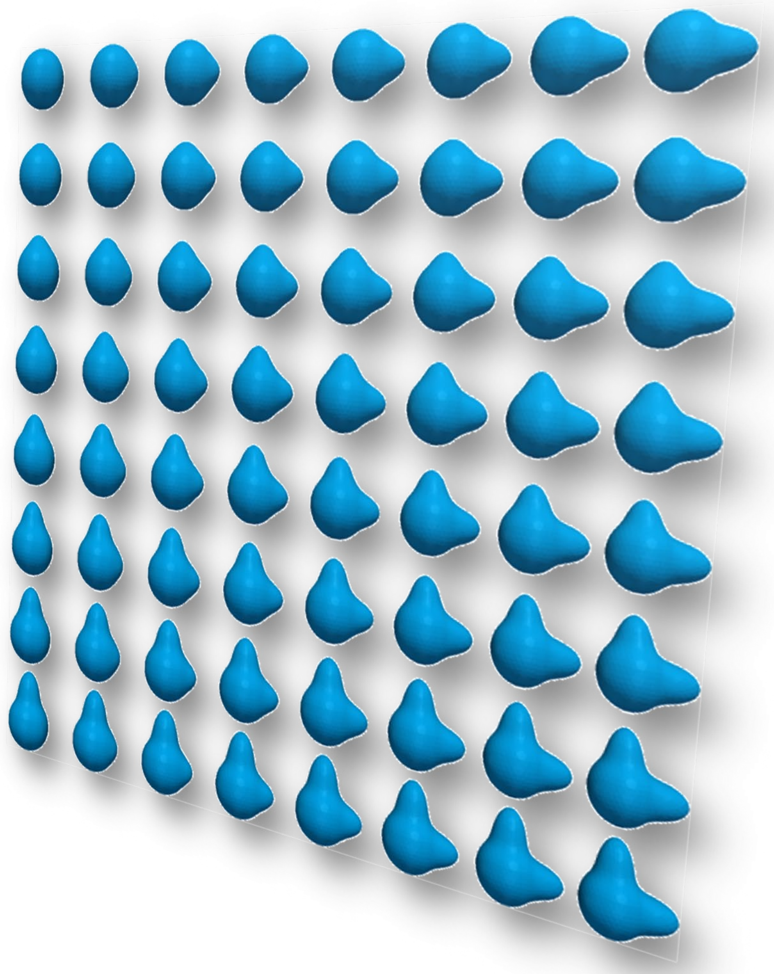


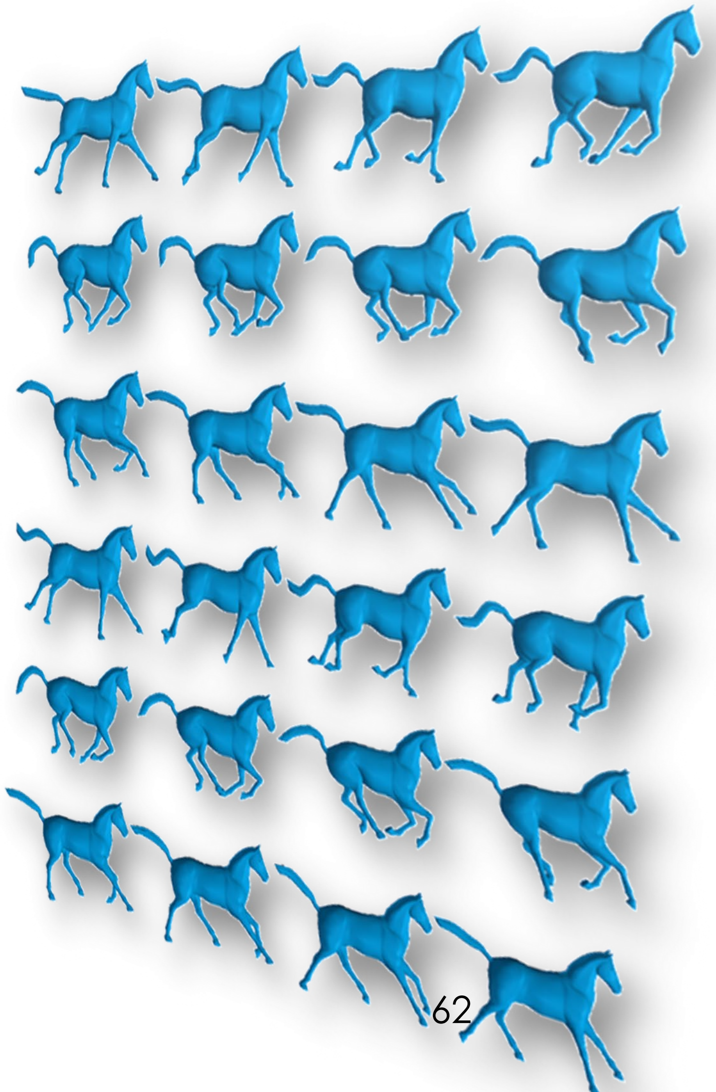
linear operator (matrix)

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

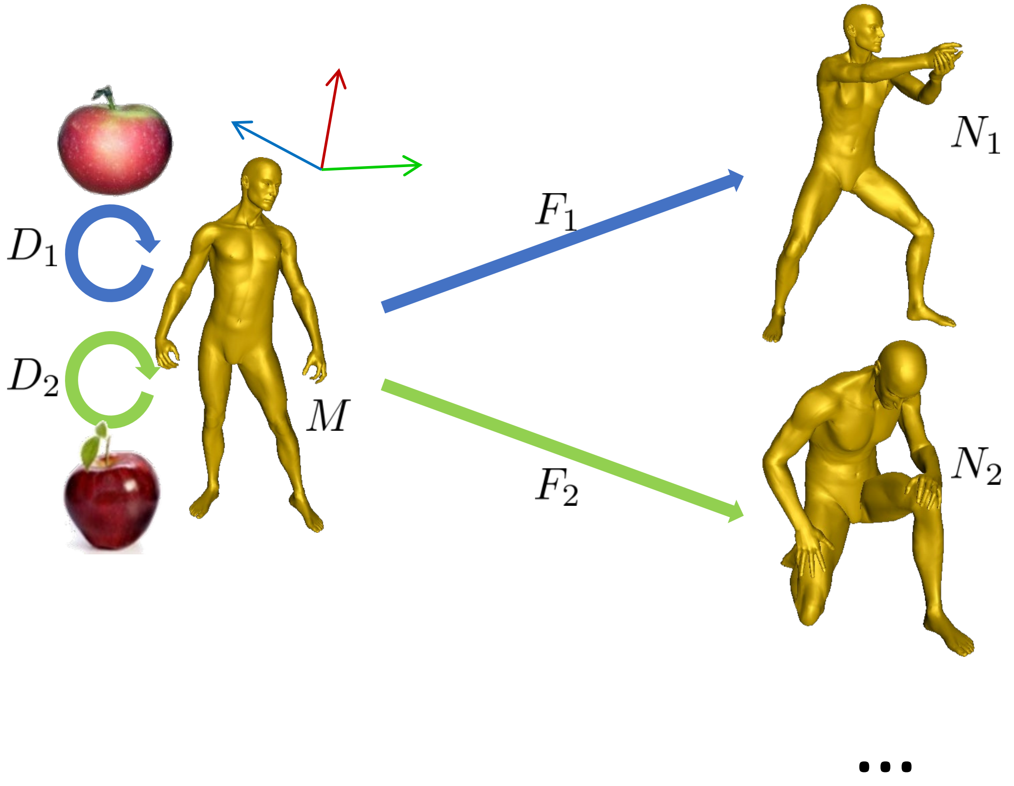
$$\int_N \nabla F(f) \nabla F(g) = \int_M \nabla f \nabla R(g)$$

Shape Differences in Collections

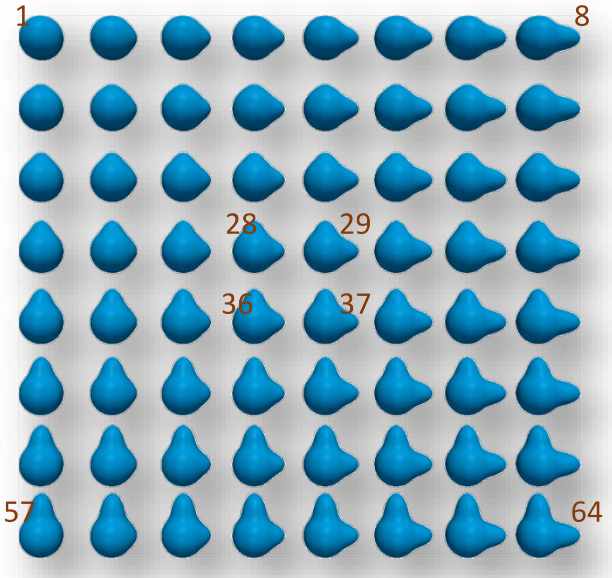


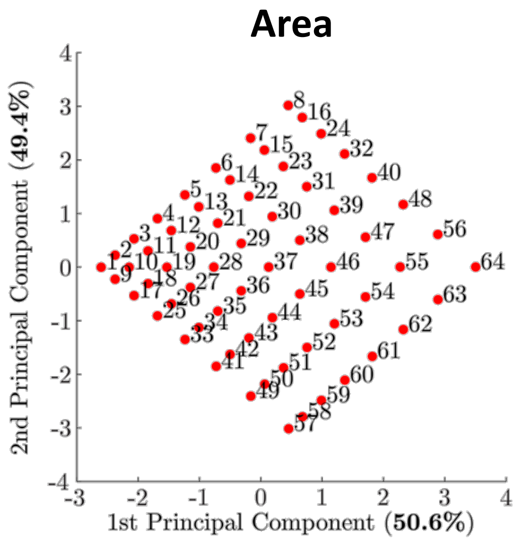


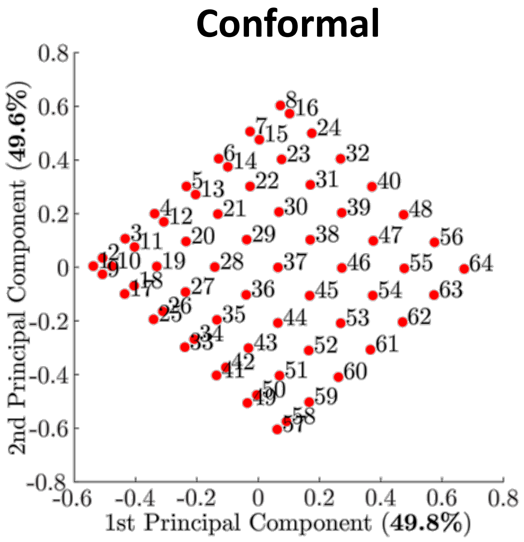
Comparing Differences

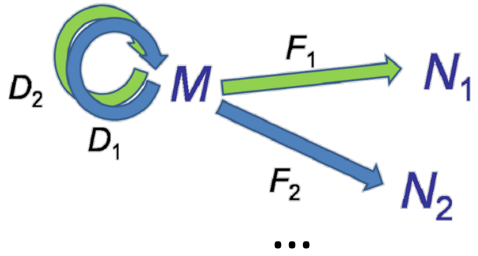


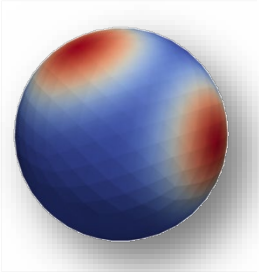
Intrinsic Shape Space

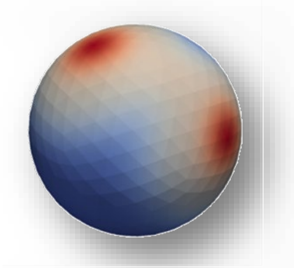






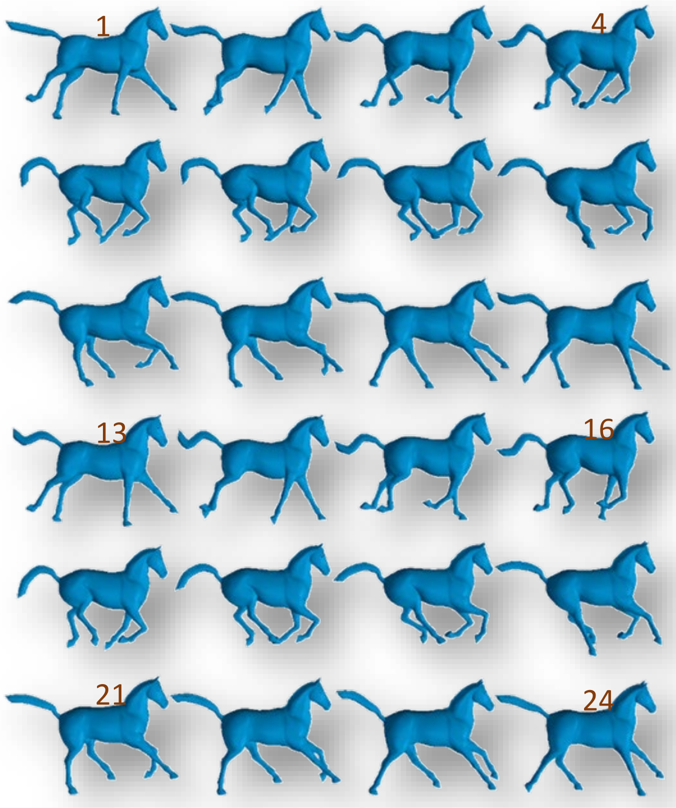


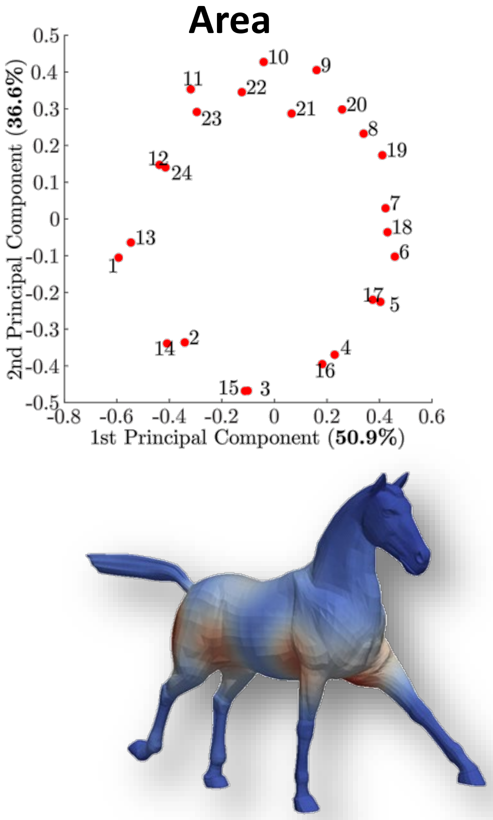


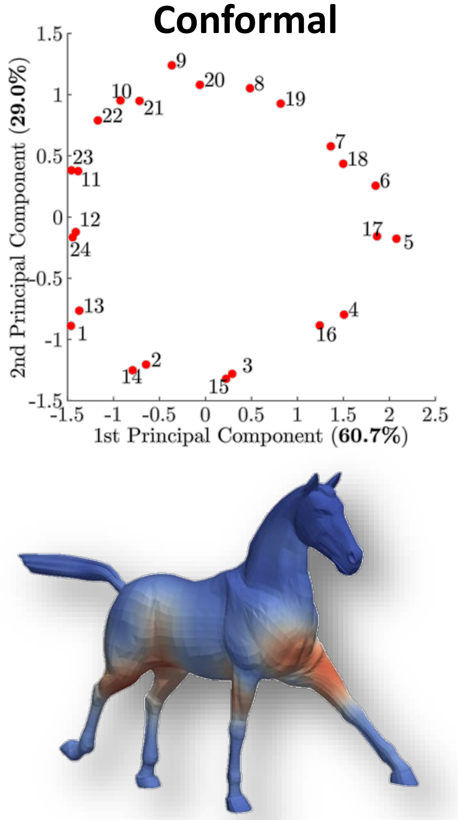


Deformation localization

Intrinsic Shape Space

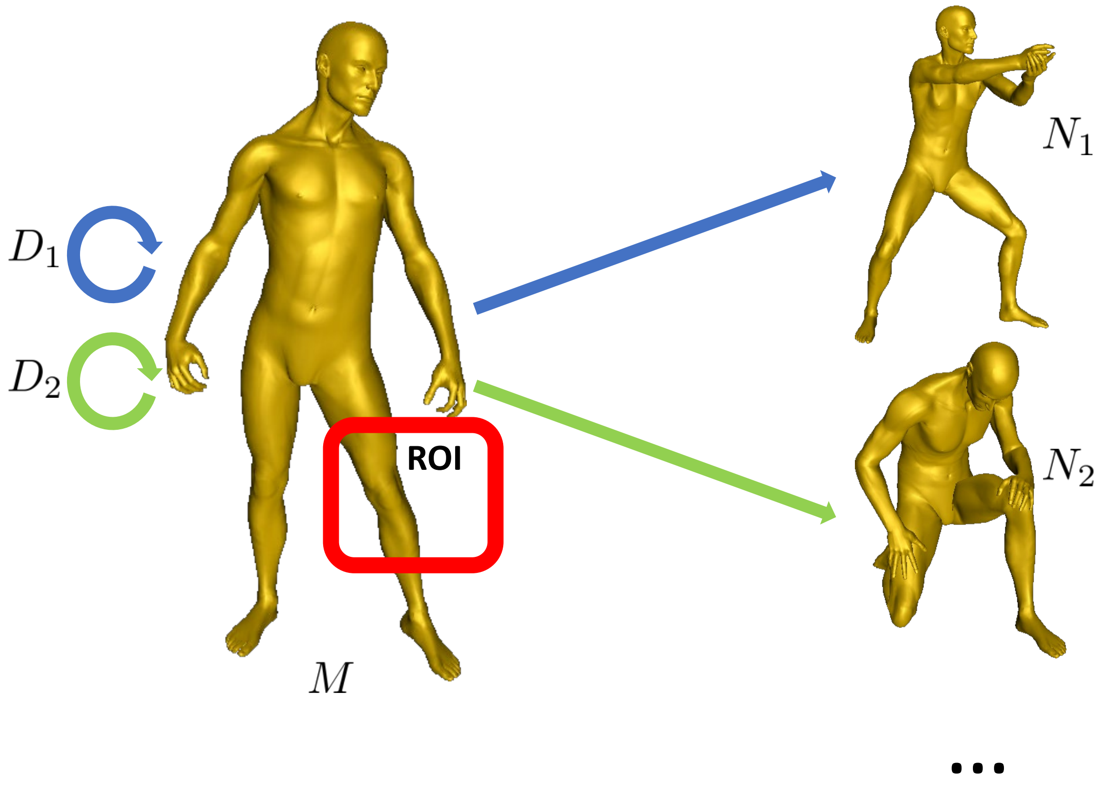






Deformation localization

Localized Comparisons

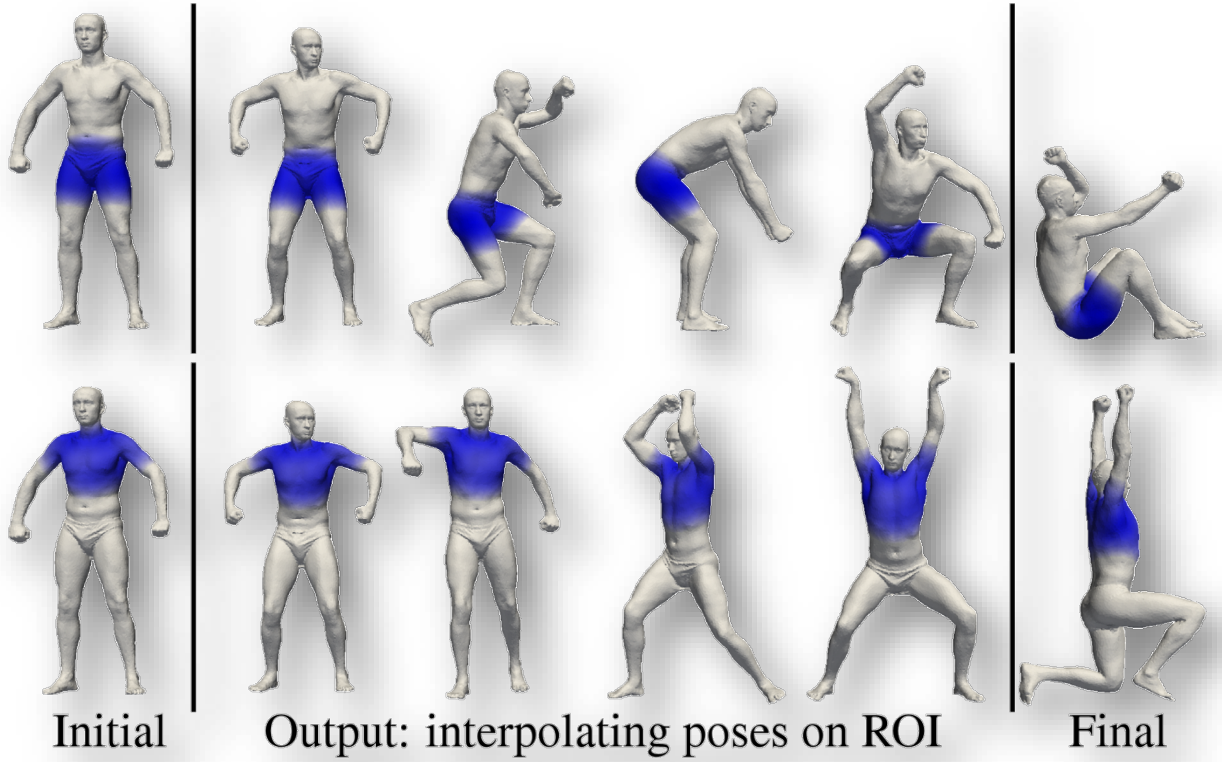


$\rho : M \to \mathbb{R}$

supported in RoI

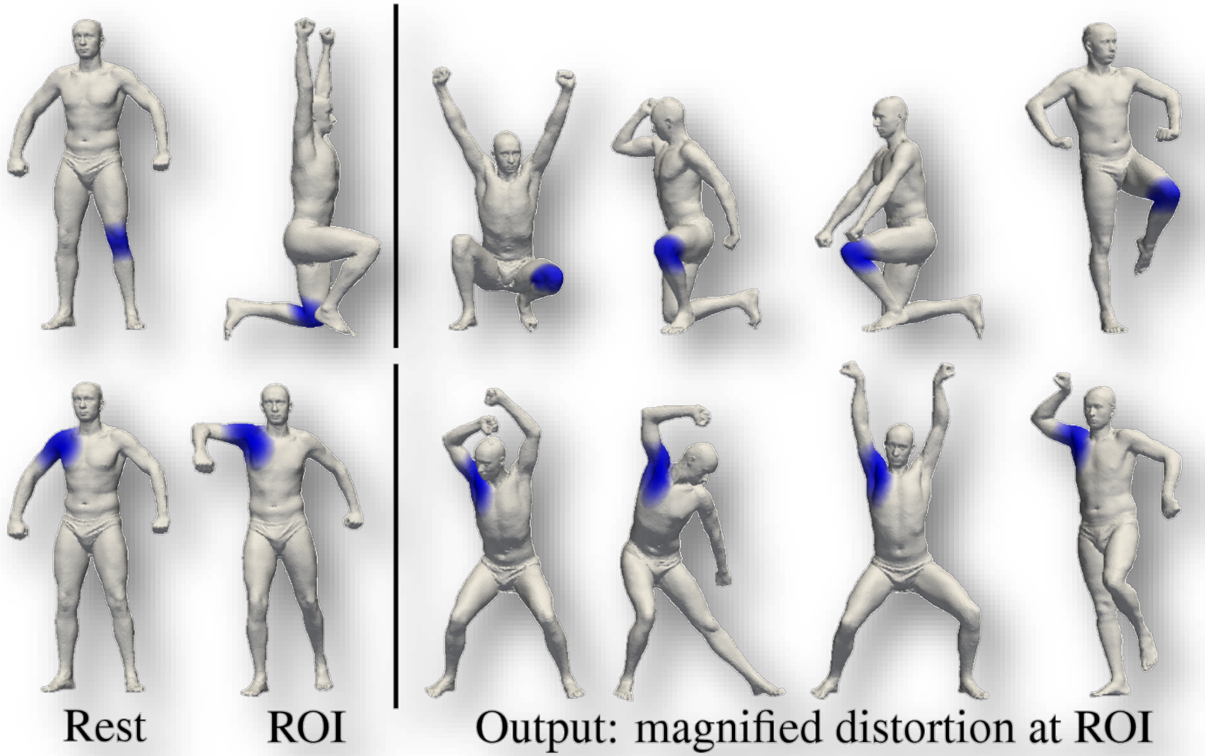
$D_1\rho$ to $D_2\rho$

Interpolation Between Poses Along RoI

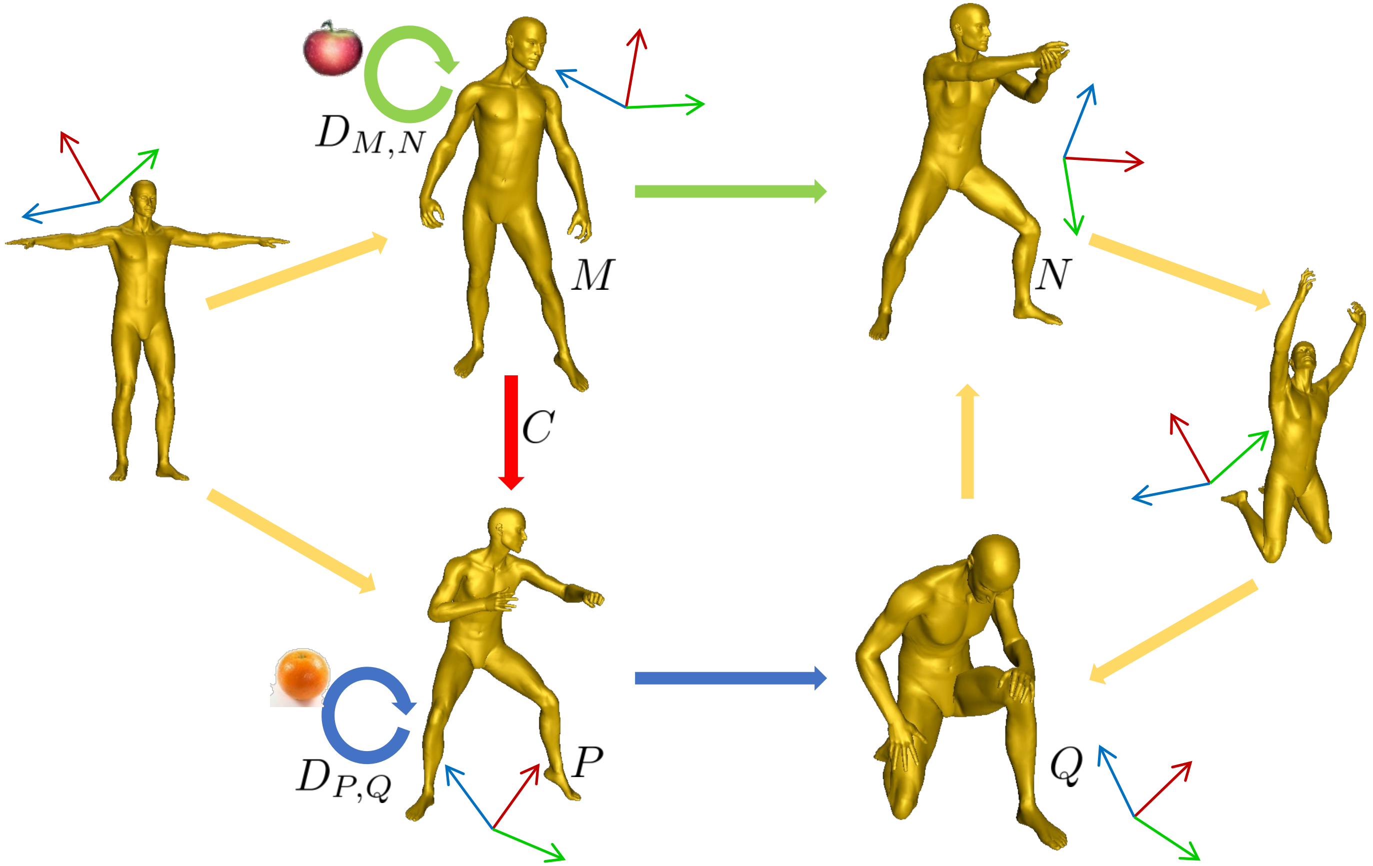


SCAPE Human Shapes

Exaggeration of Difference in RoI

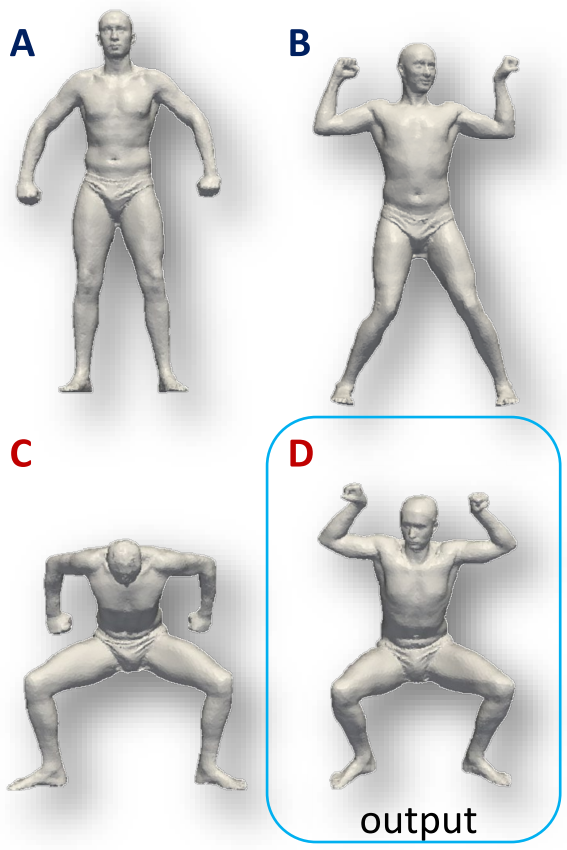


Comparing Differences II



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

Analogies: D relates to C as B relates to A



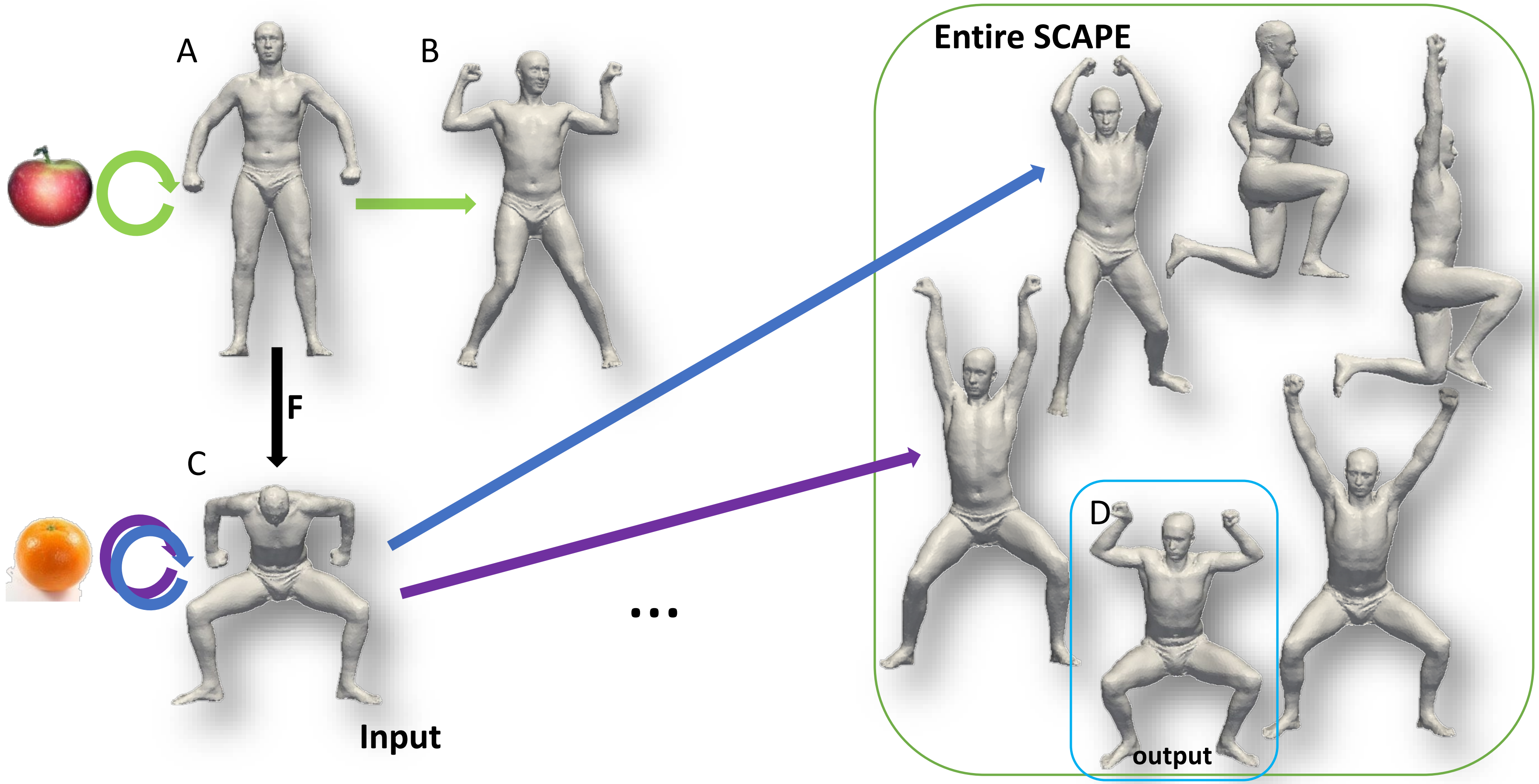
$$D = C + (B - A)$$

hands raised up

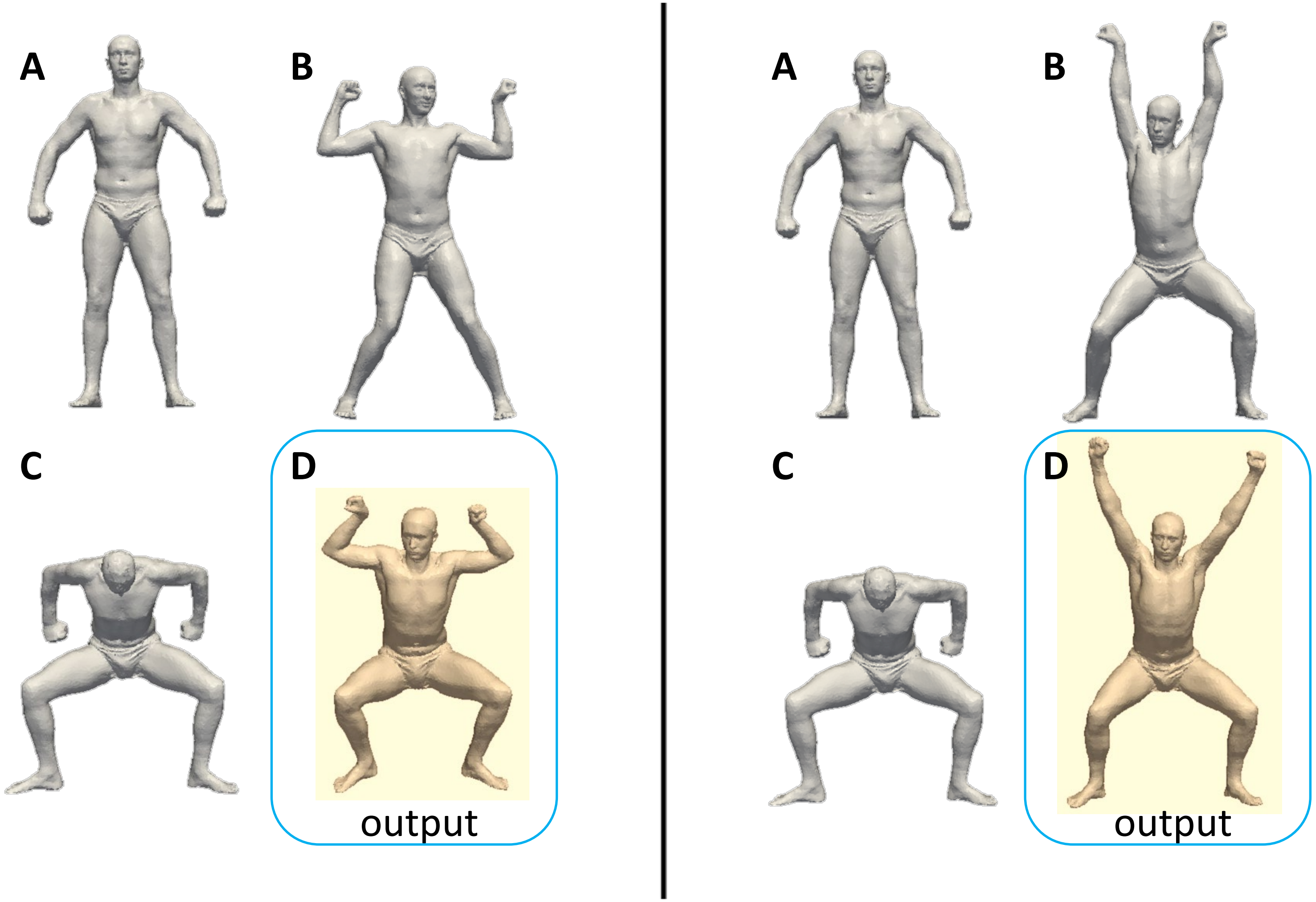
or

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

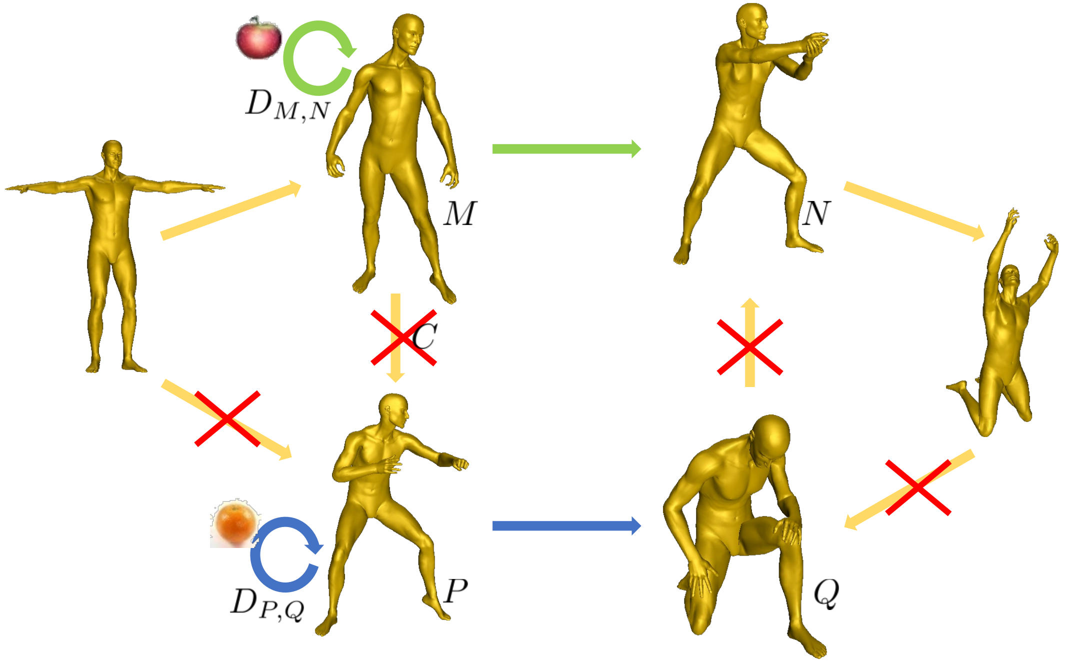
Analogies: D relates to C as B relates to A



Shape Analogies



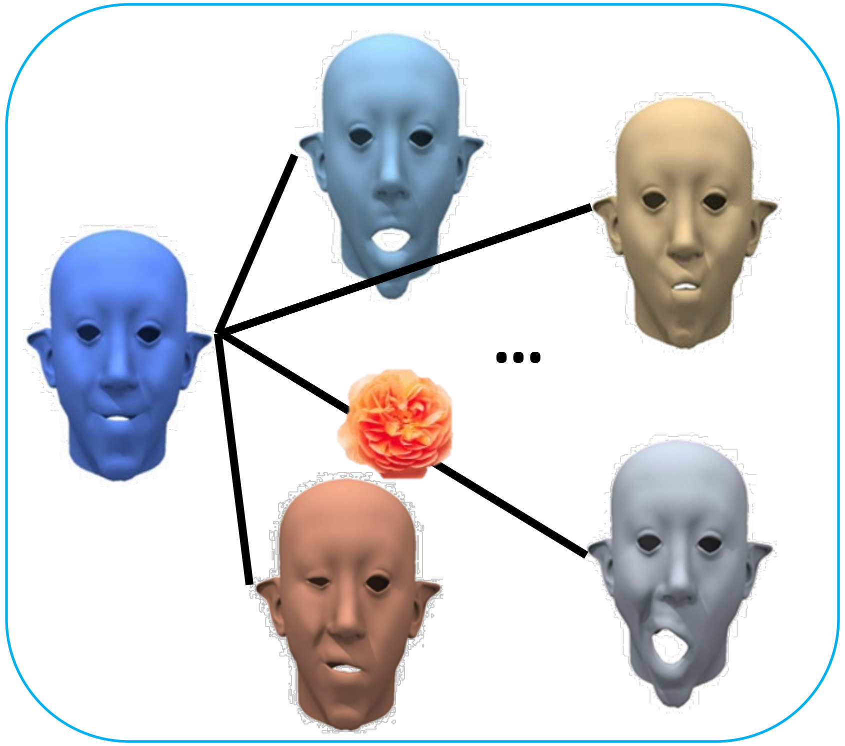
Comparing Differences III



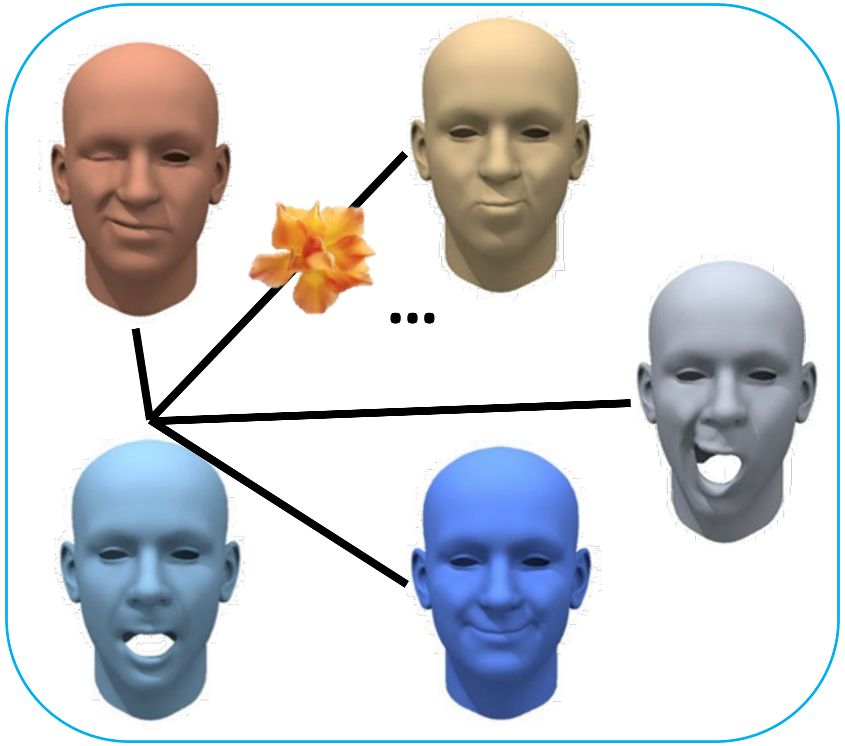
$$D_{M,N} \sim C^{-1} D_{P,Q} C$$

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

Aligning Disconnected Collections

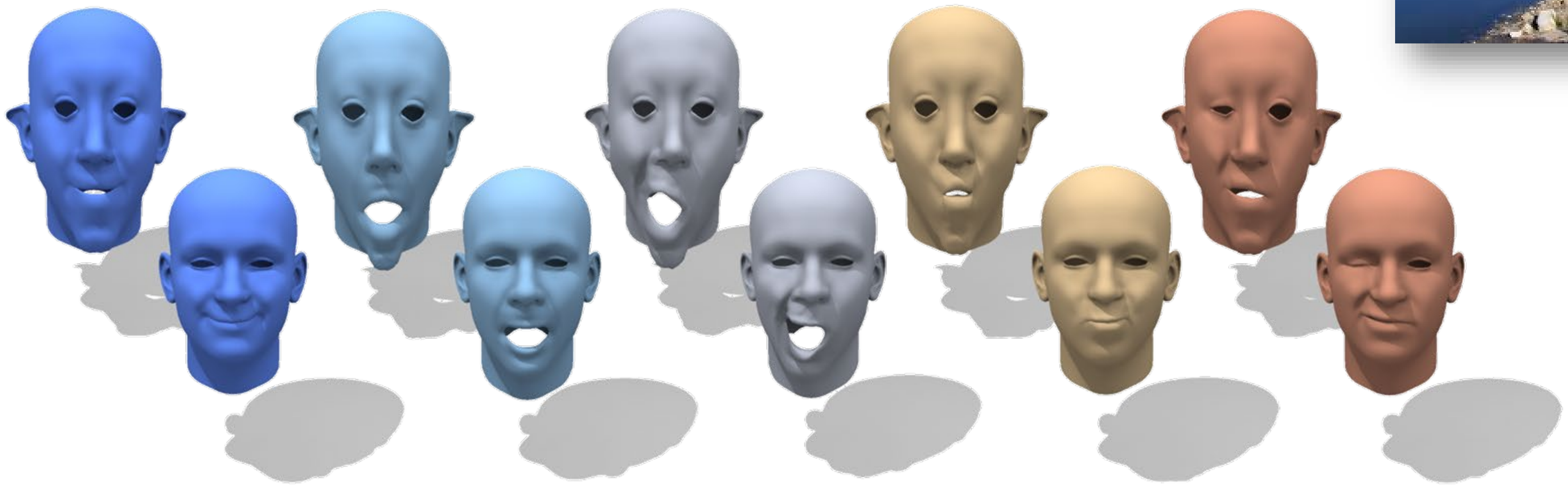


Complete graph



Complete graph

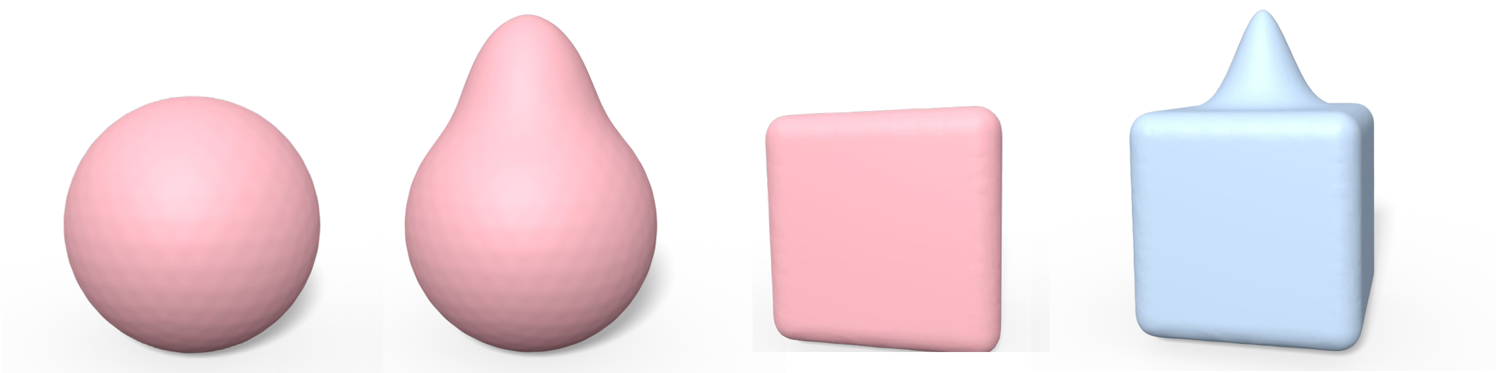
Aligning, Without "Crossing the River"



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]

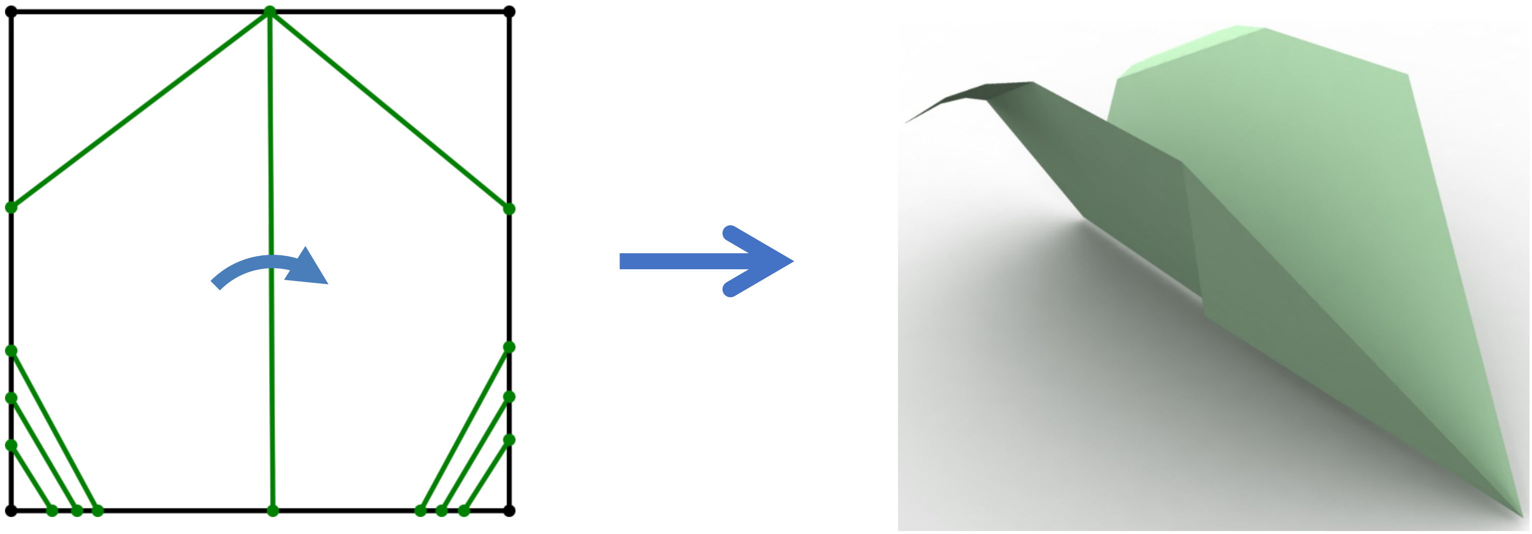
Area-based shape difference → Area weights → Triangle areas

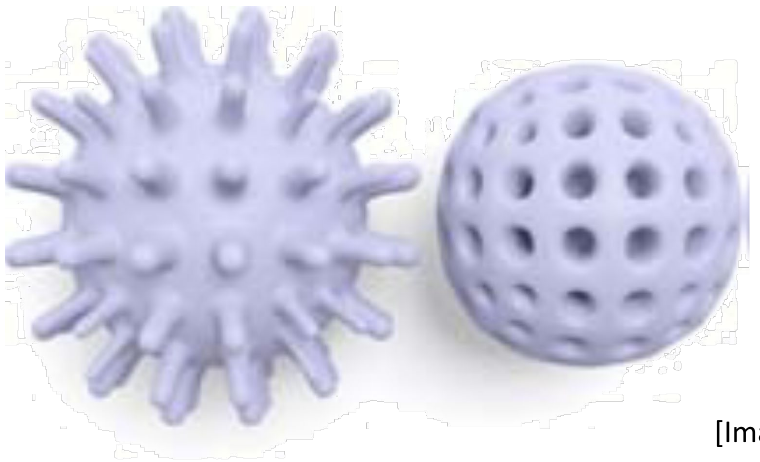
Given triangle areas and conformal inner products, can solve for squared edge lengths.

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

How to Encode Extrinsic Information?

[Image: J. Solomon]





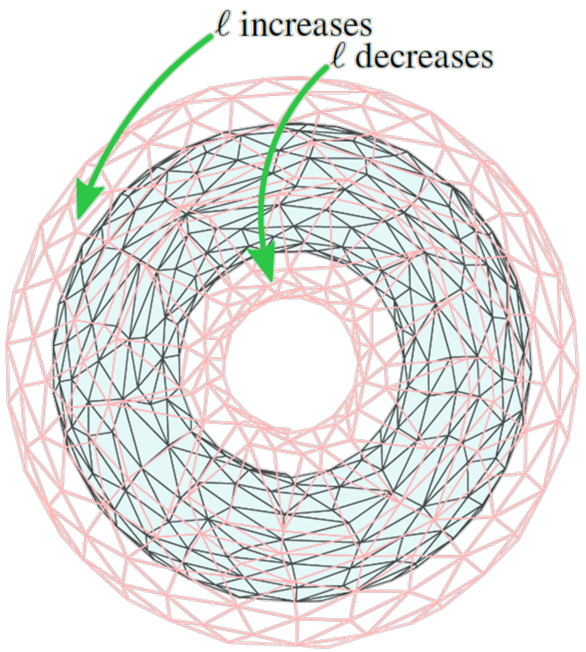
[Image: K. Crane]

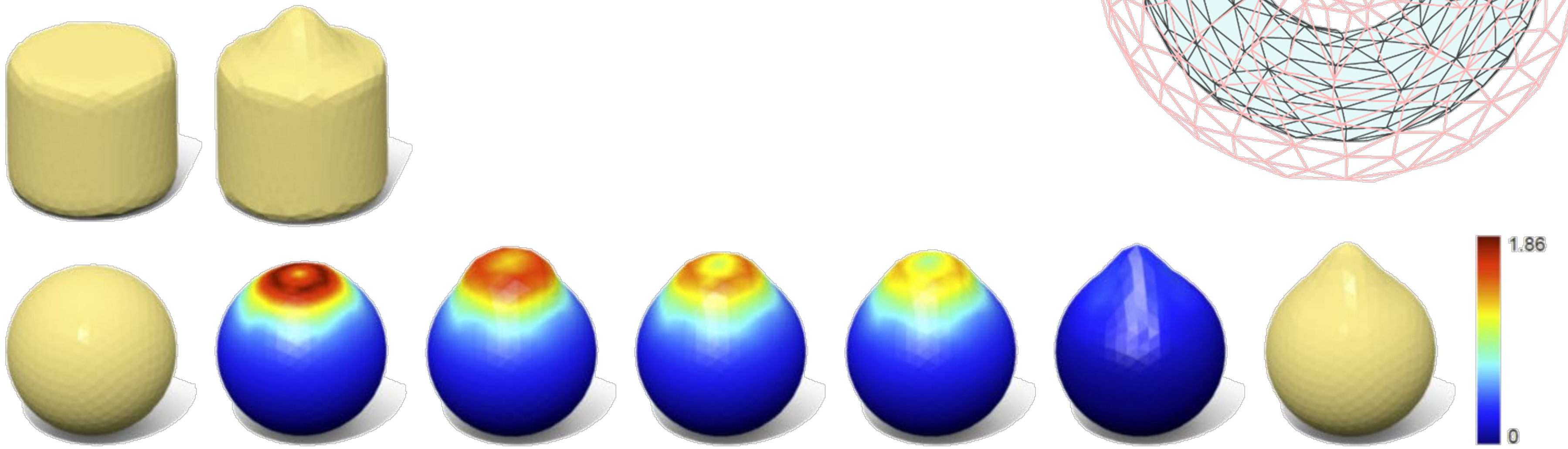
Extrinsic Shape Differences, Version 1

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



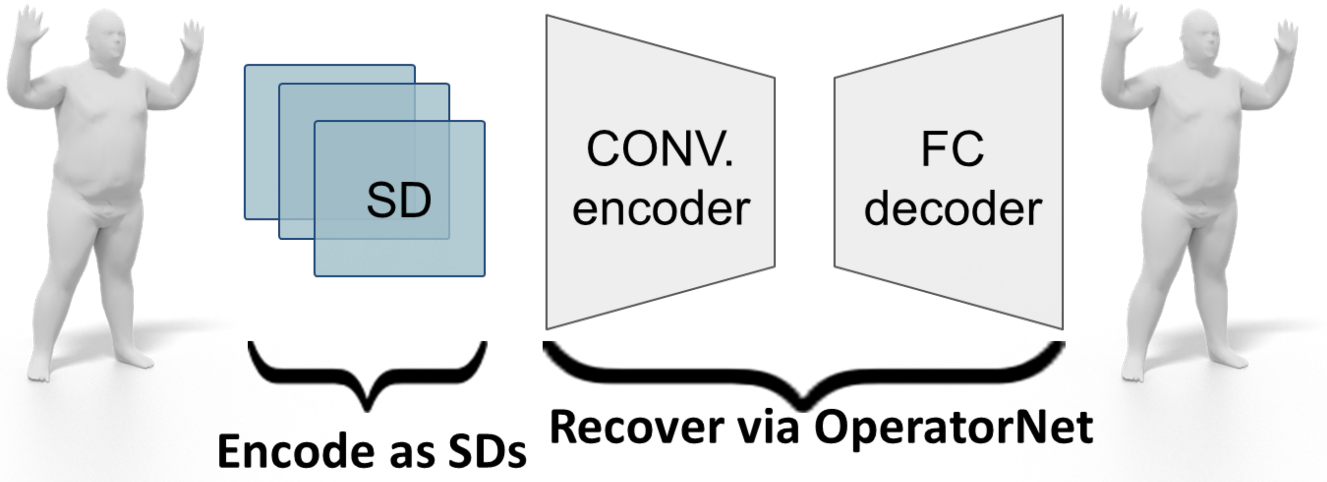


Extrinsic Shape Differences, Version 2

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

OperatorNet



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:

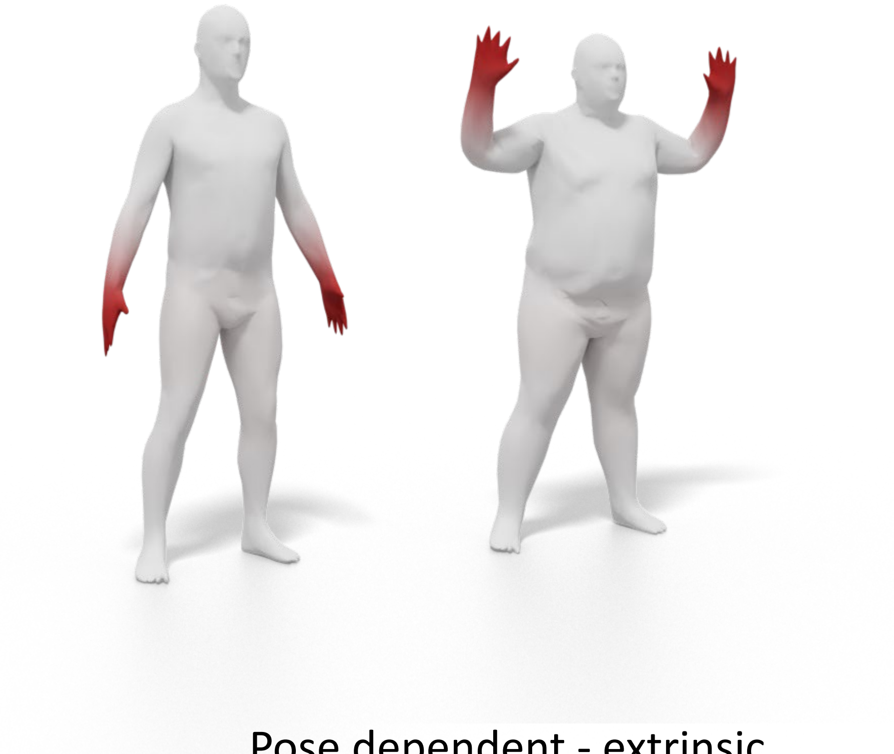
$$\begin{aligned} V &= A_{\text{source}}^{-1} F^T A_{\text{target}} F \\ R &= (L_{\text{source}} A_{\text{source}})^{-1} F^T L_{\text{target}} A_{\text{target}} F \\ E &= (M_{\text{source}} A_{\text{source}})^{-1} F^T M_{\text{target}} A_{\text{target}} F \end{aligned}$$

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.

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

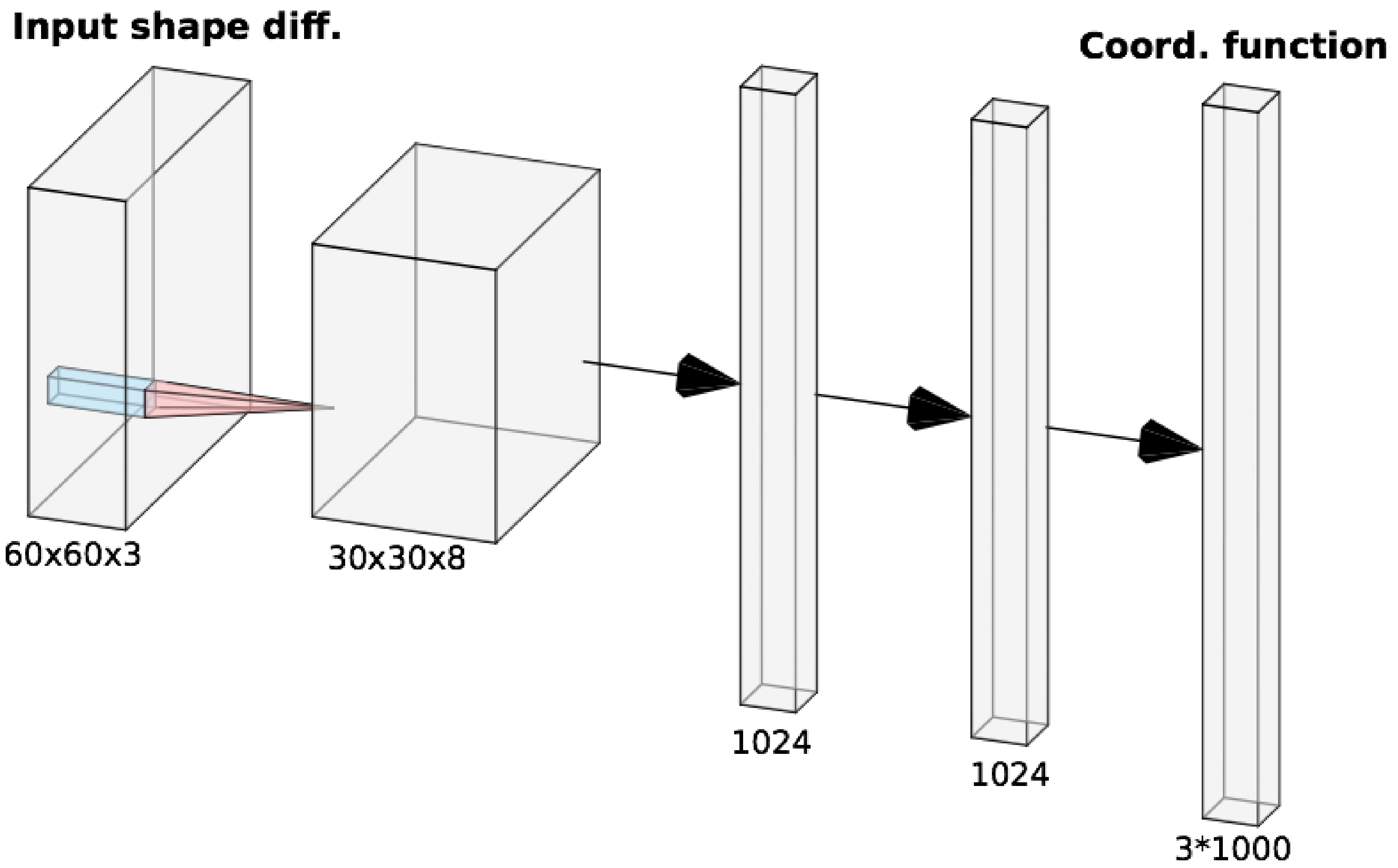
Example

Pose oblivious -- intrinsic

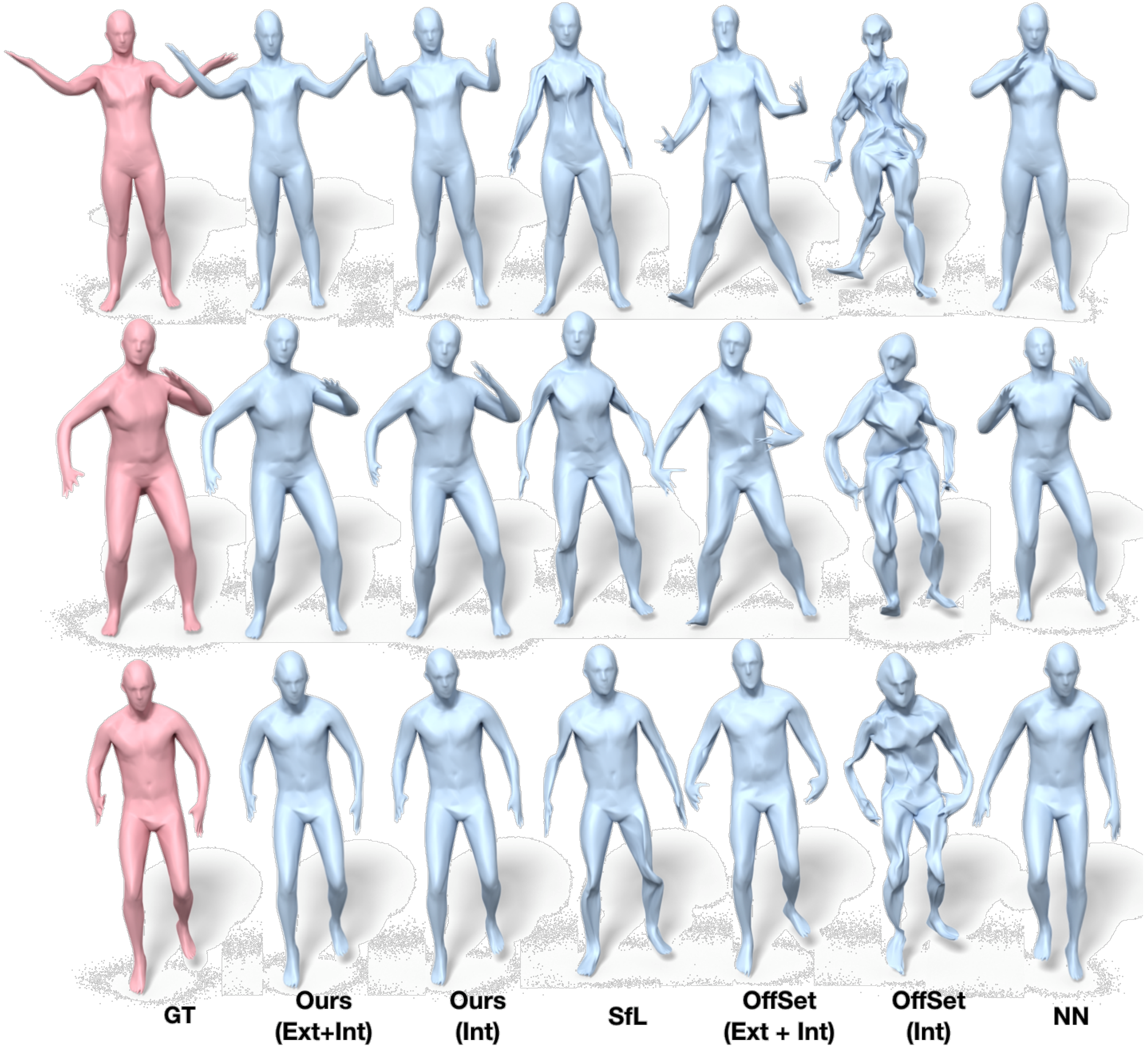


Pose dependent - extrinsic

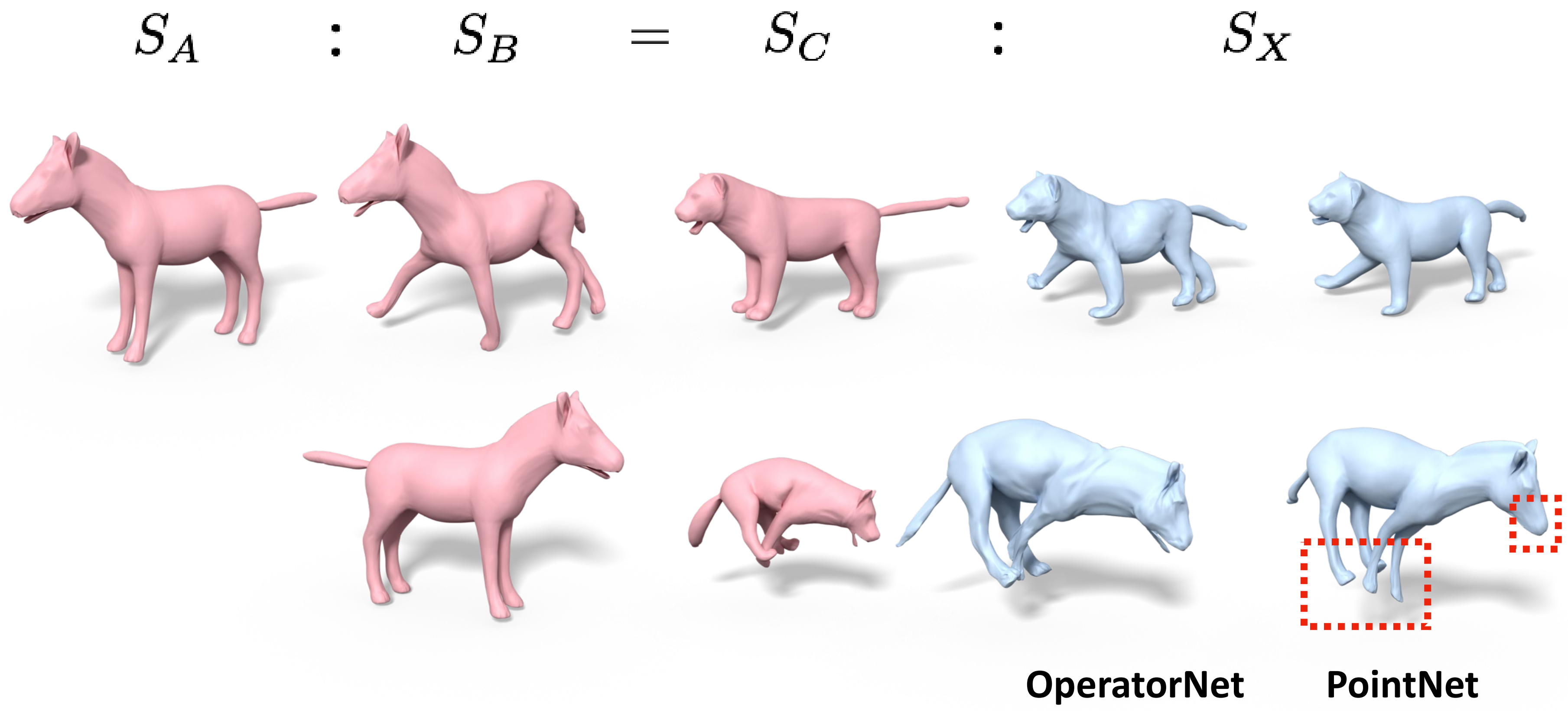
OperatorNet Reconstruction



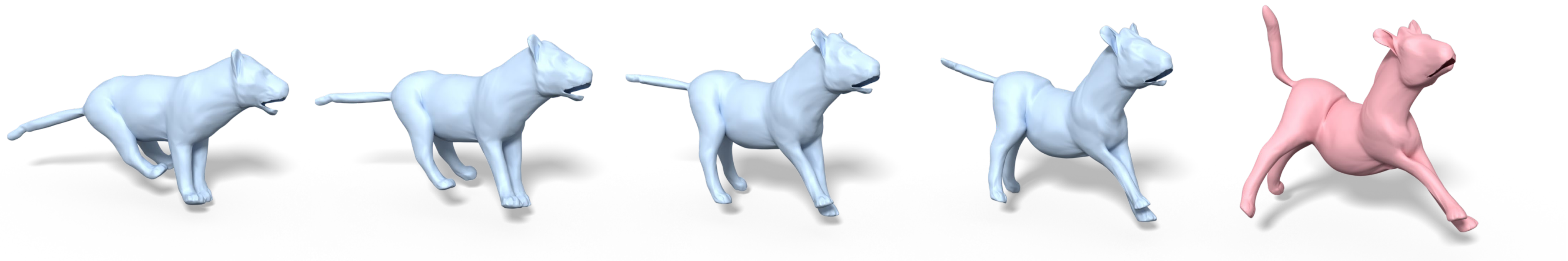
Reconstruction Comparisons



Shape Analogies

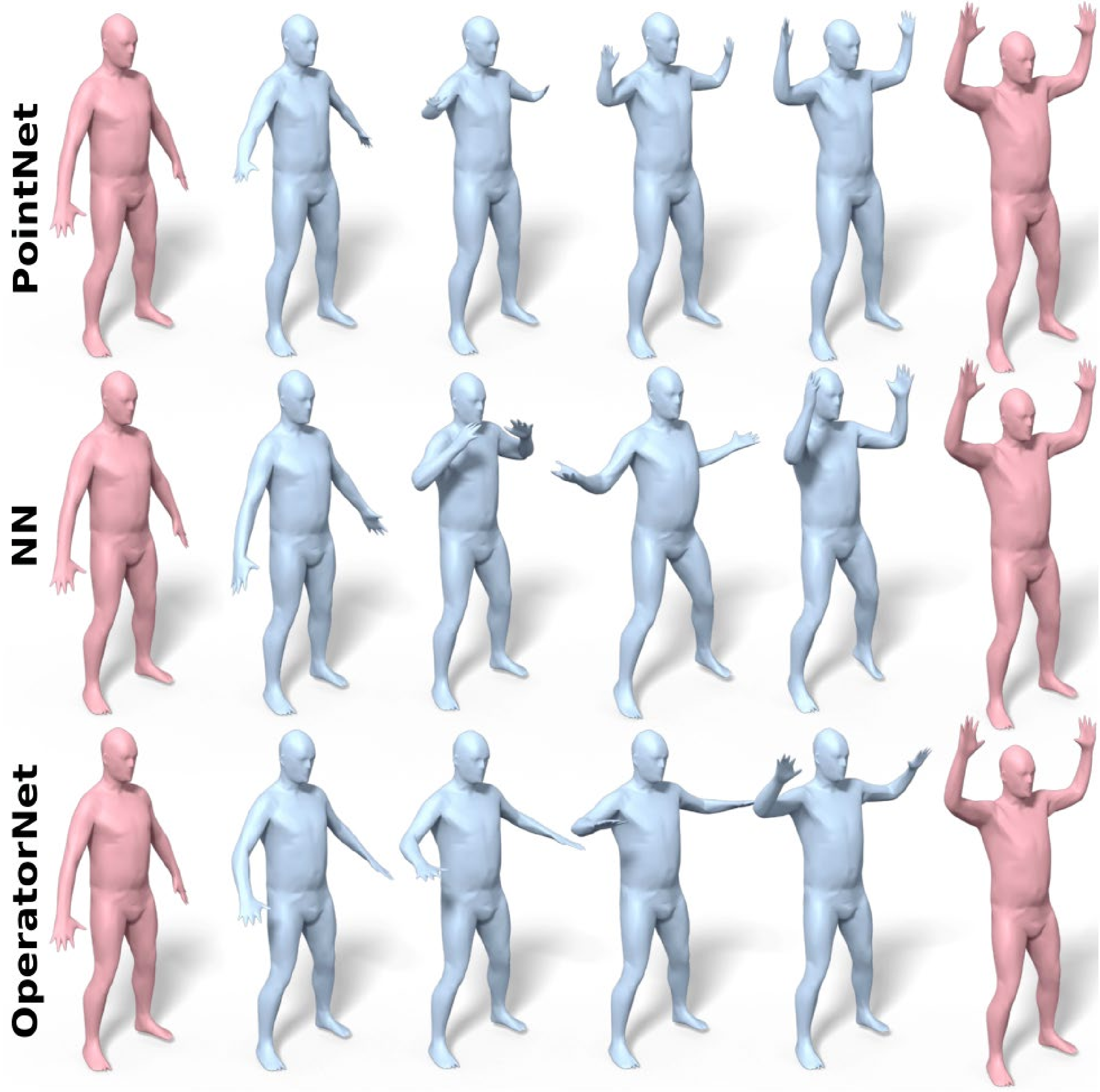


Interpolation

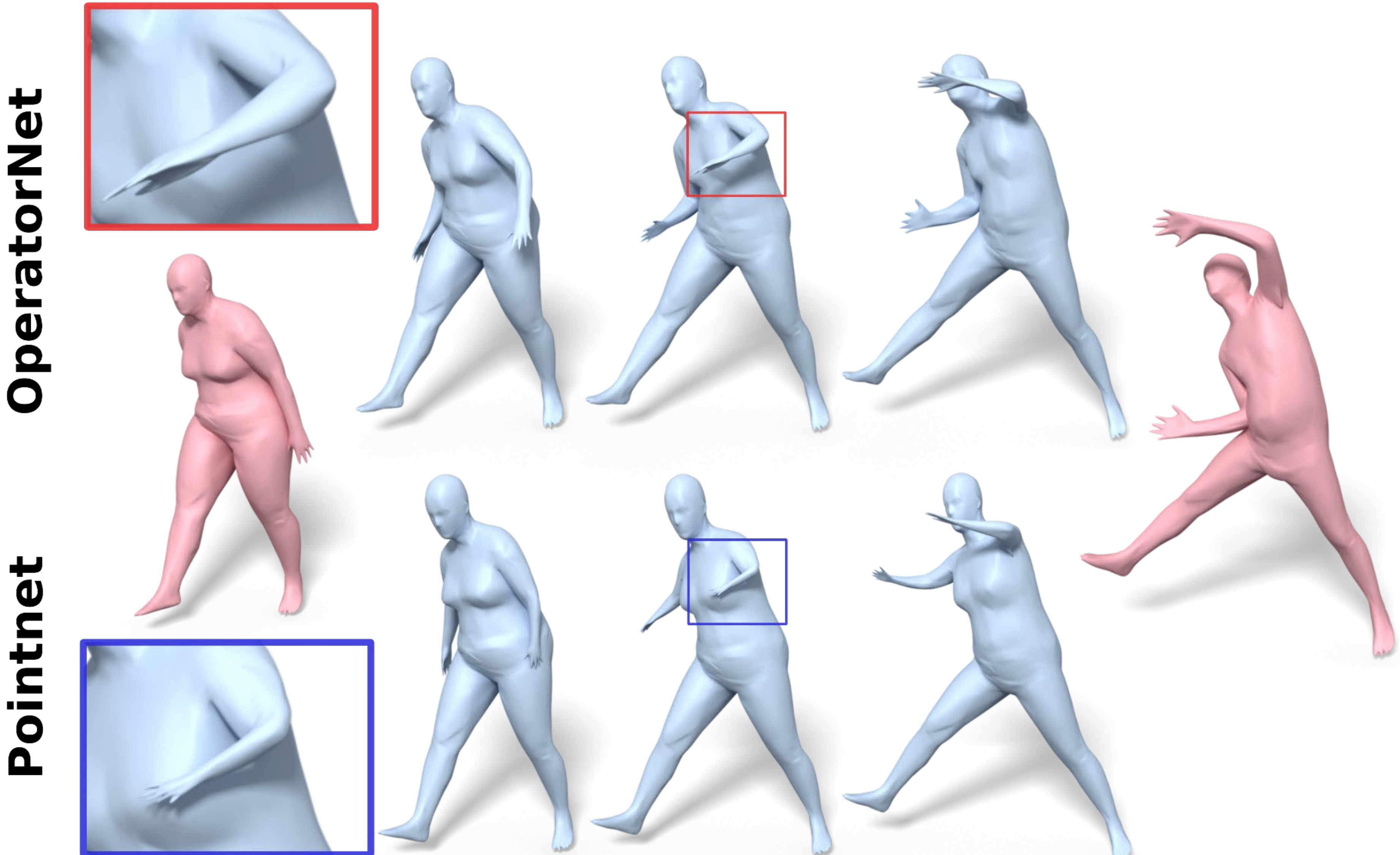


Interpolation between a dog and a horse

Shape Interpolation



Interpolation Detail



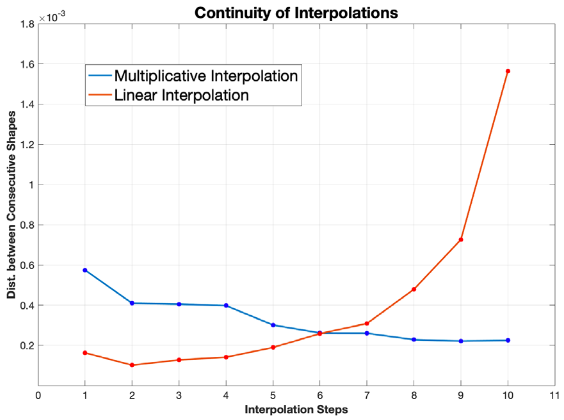
Additive or Multiplicative?

Mult. Interp.

Linear Interp.

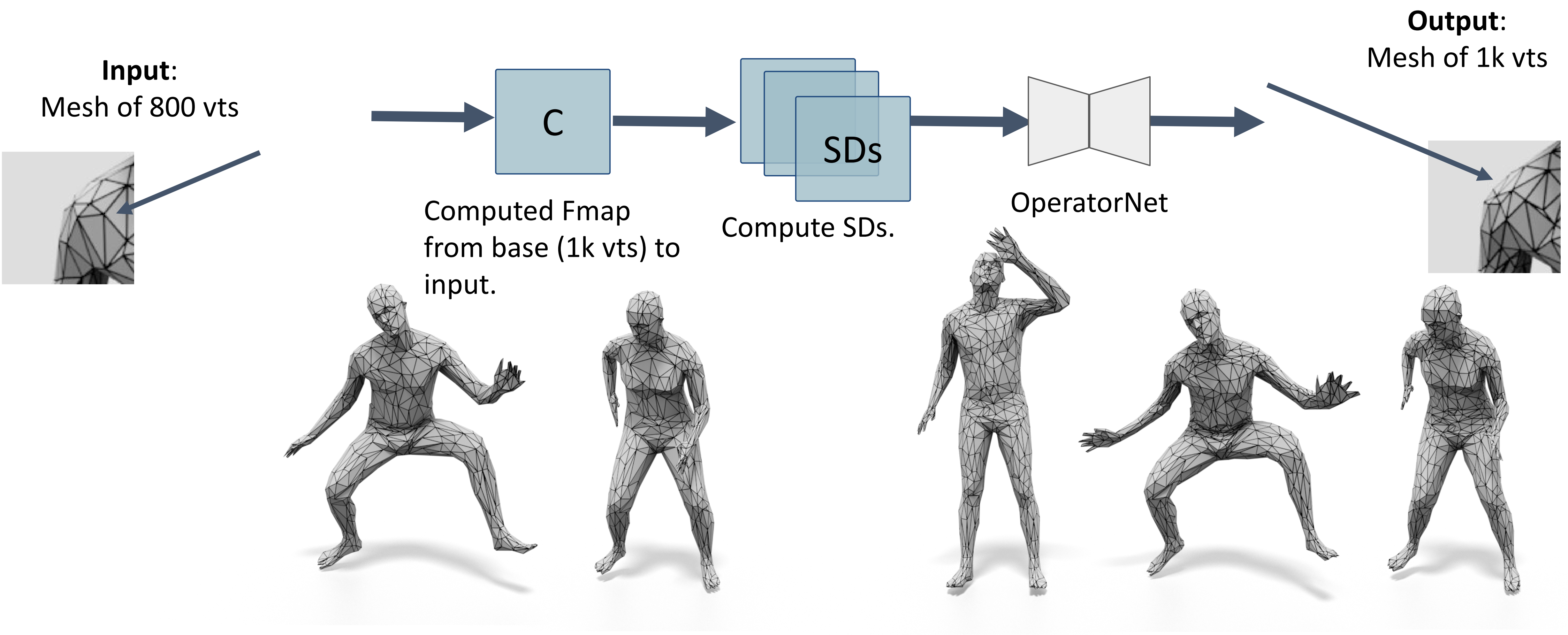
$$C \leftarrow A^t B^{1-t}$$

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



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

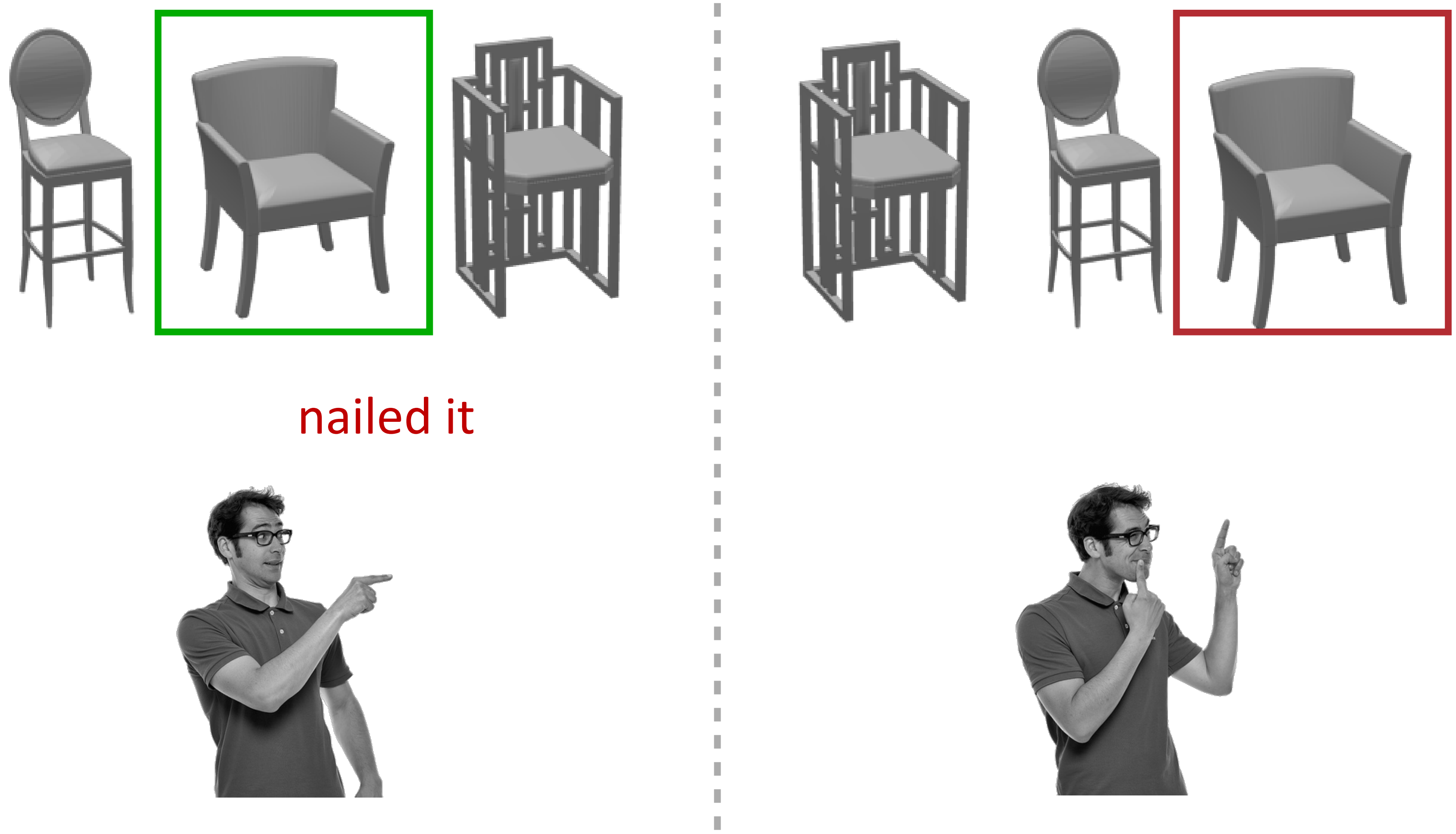
Target Object

"Gaps in the back"

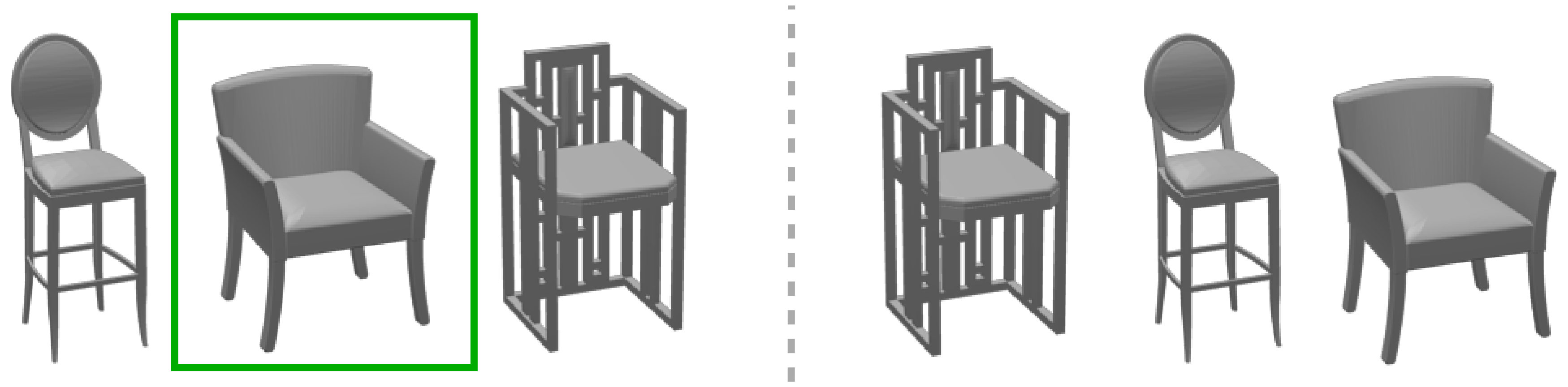
A Reference Language Game

‘looks like a sofa’

A Reference Language Game



Considerations



scrambled order
same view-point
no texture

no location
no orientation
yes geometry

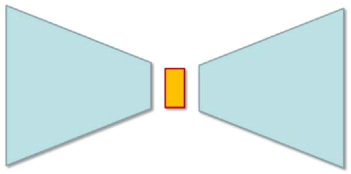
Communication Context

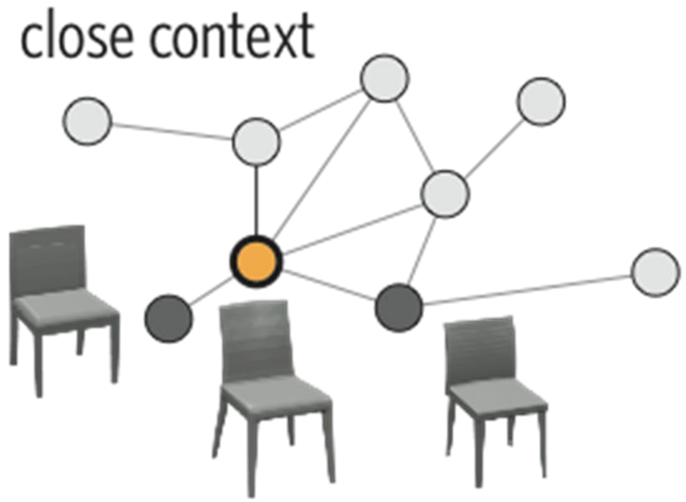
impossible

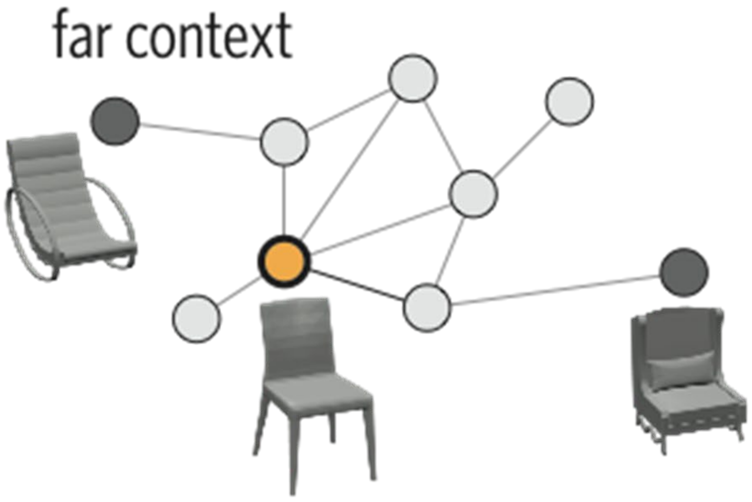
trivial

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

"it has wheels"

"has vertical lines on the back"

"rectangle back with straight legs"

Neural Listener w. Attention Model

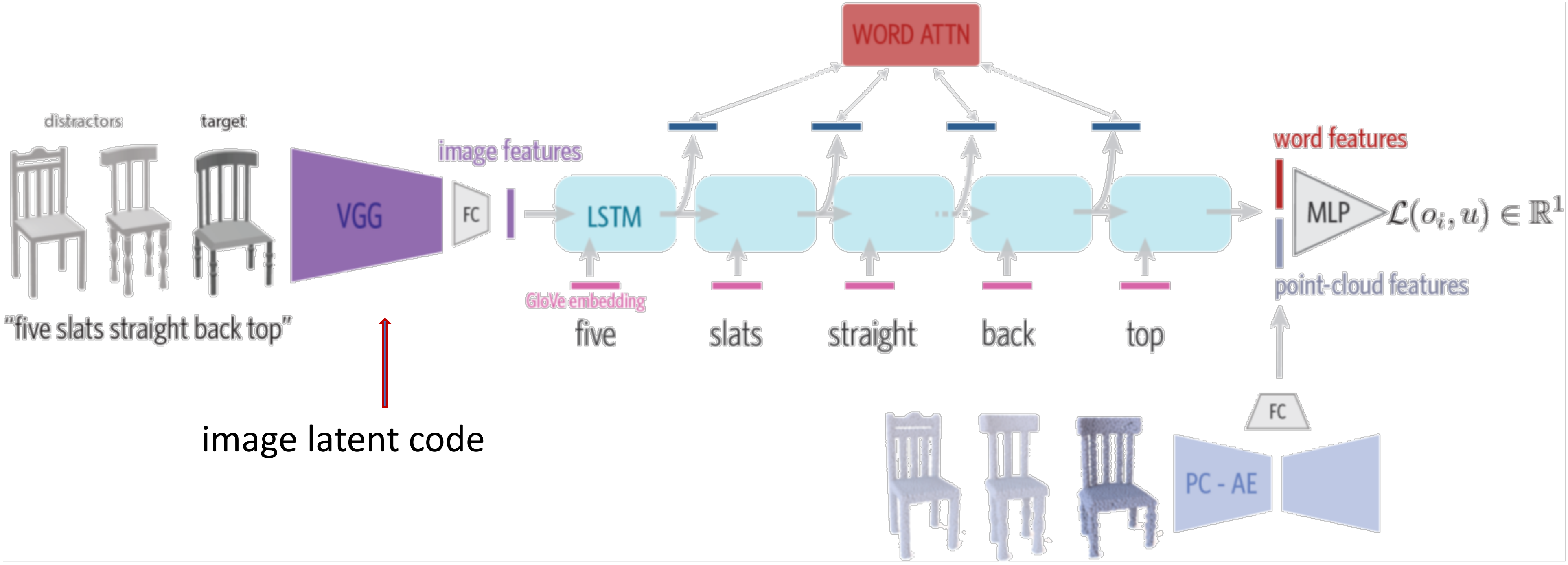
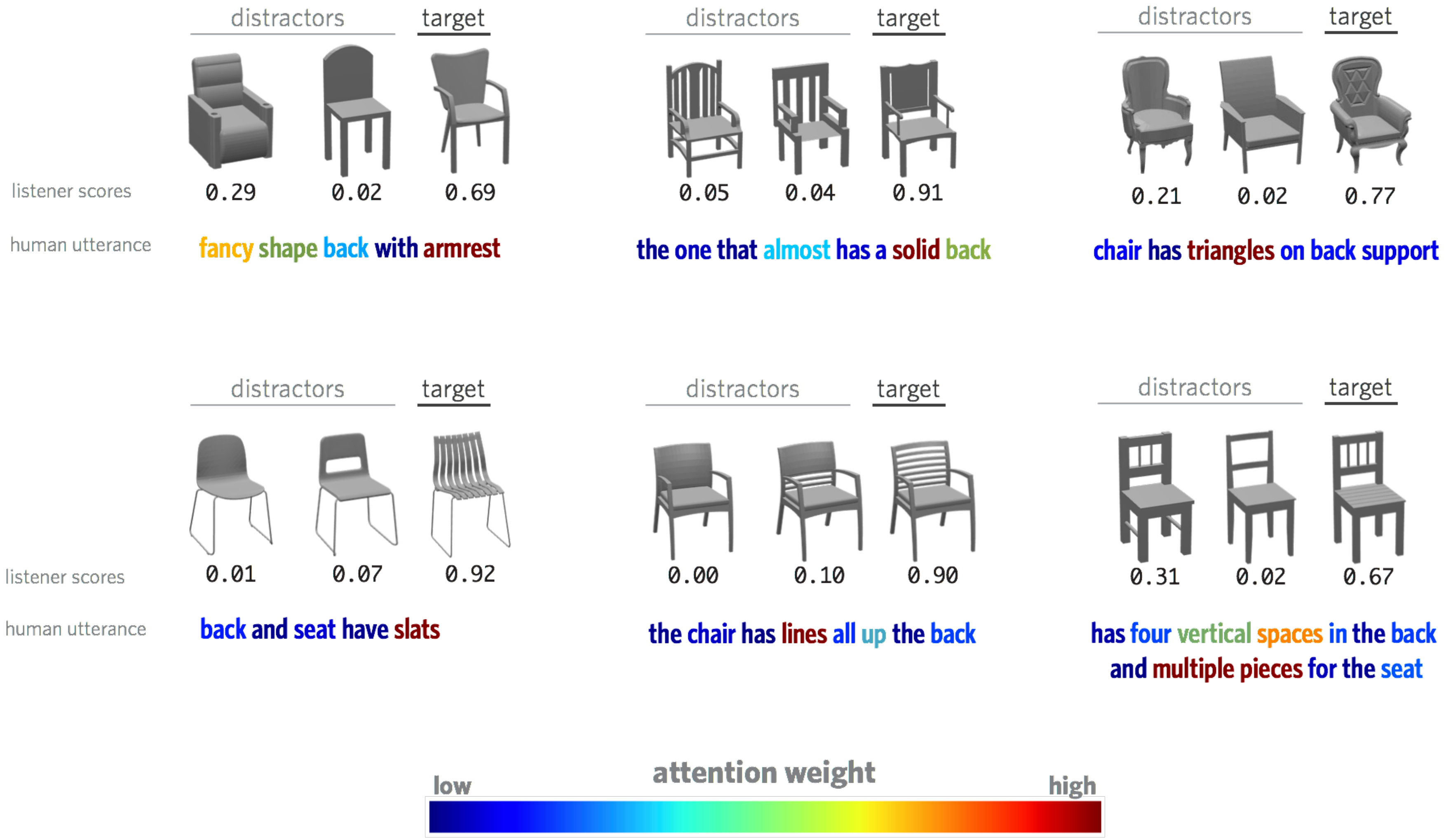


image latent code

point cloud latent code

Attentive Neural Listening

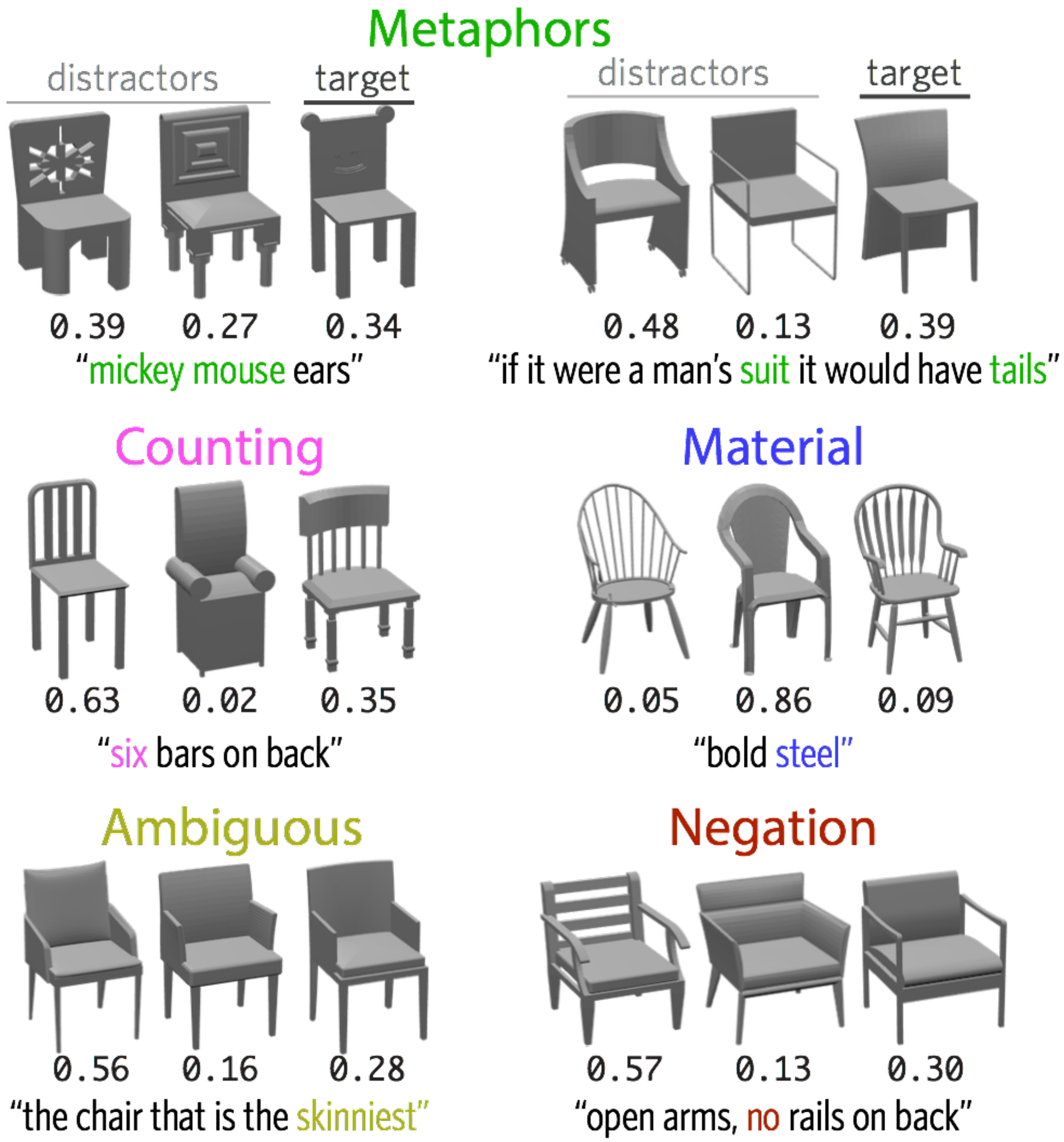


Performance

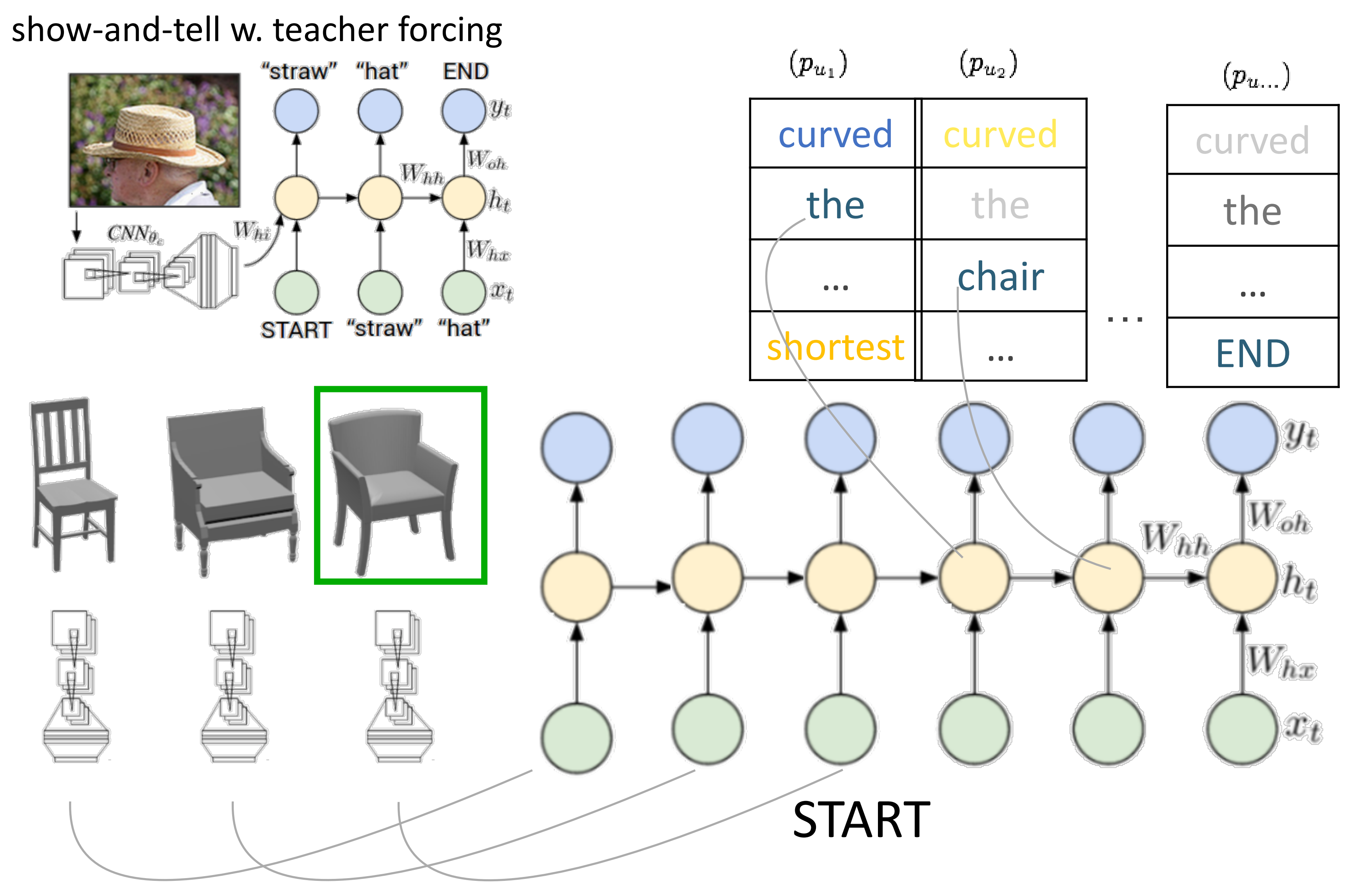
Architecture	Subpopulations			
	Overall	**Close**	**Far**	**Sup-Comp**
Together-with-Context	75.9 $\pm$ 0.5%	67.4 $\pm$ 1.0%	83.8 $\pm$ 0.6%	74.4 $\pm$ 1.3%
Separate-with-Context	79.4 $\pm$ 0.8%	**70.1** $\pm$ 1.3%	**88.1** $\pm$ 0.6%	75.2 $\pm$ 2.1%
Separate (proposed)	**79.6** $\pm$ 0.8%	69.9 $\pm$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

"tall**est**"

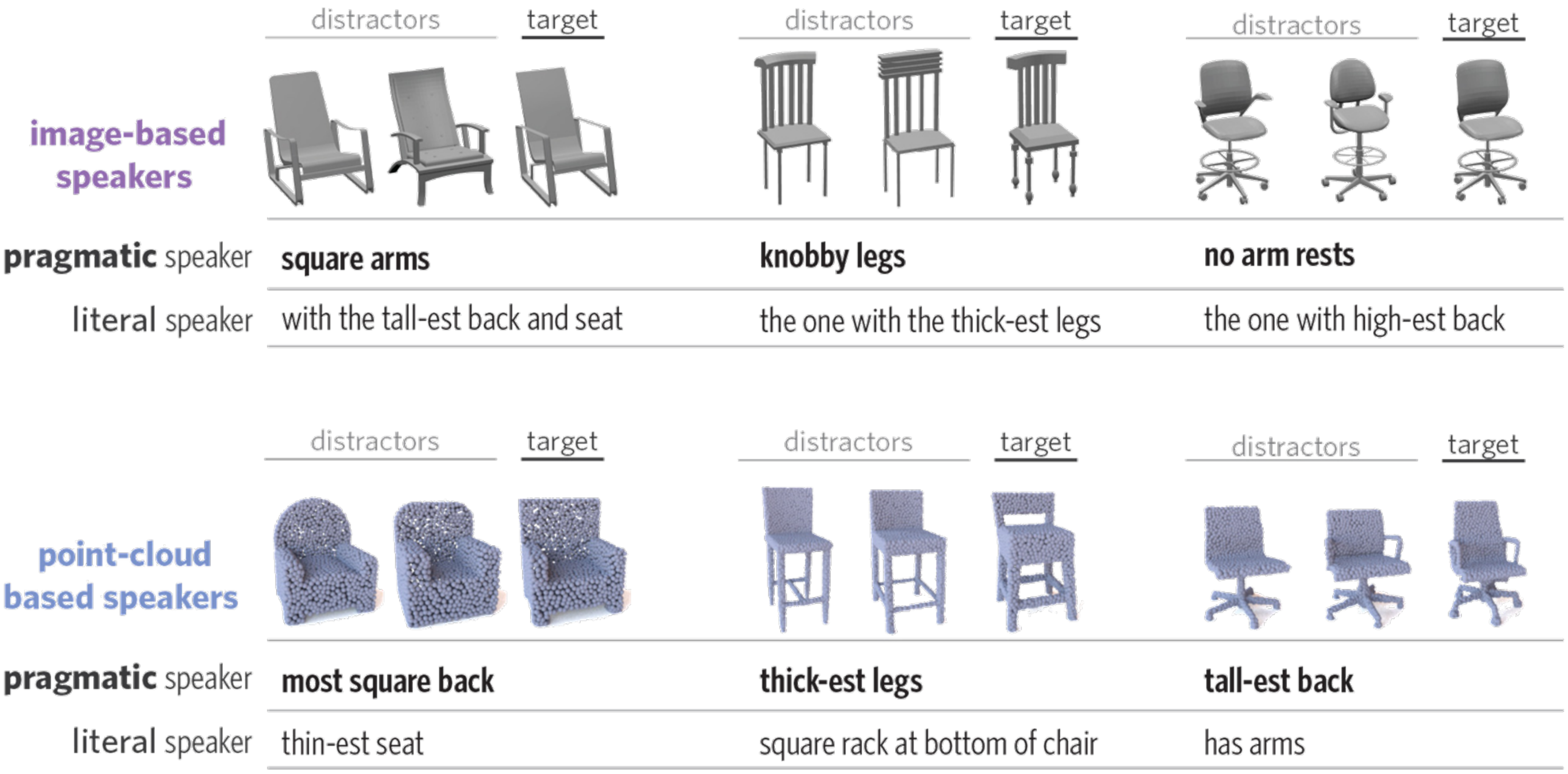
Listener Failures



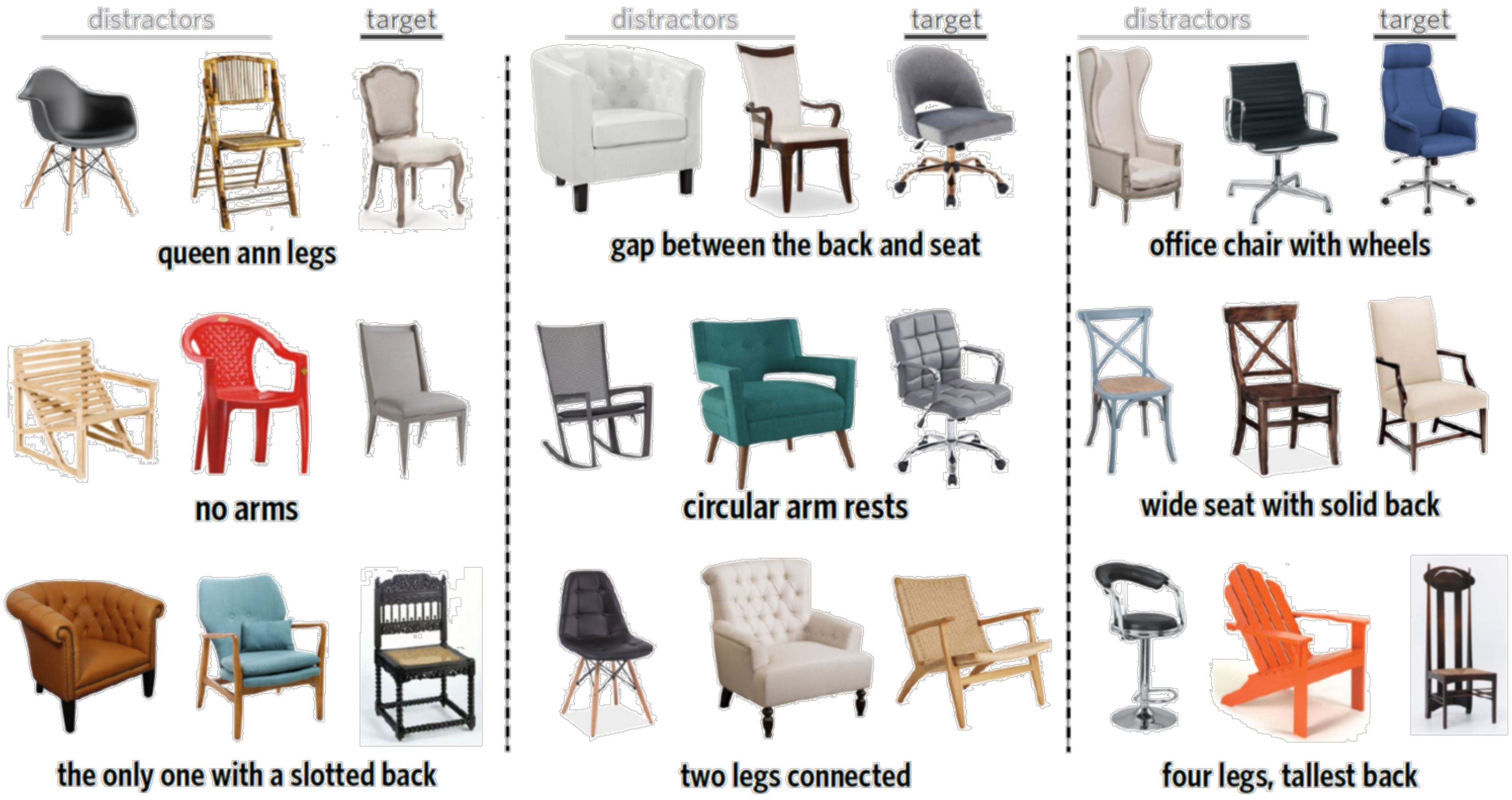
Neural Speaker



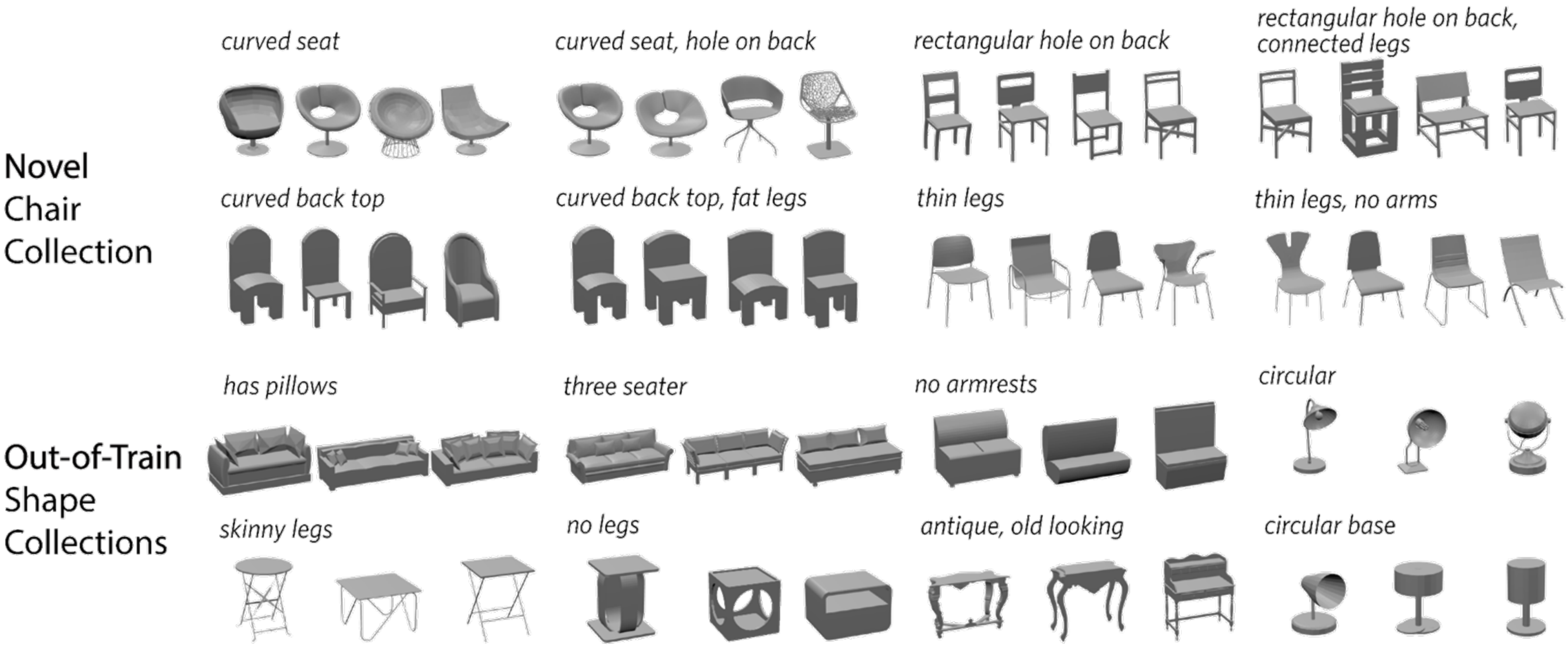
Speaker Examples



Speaker from Images



Listener Examples: Shape-Based Product Retrieval



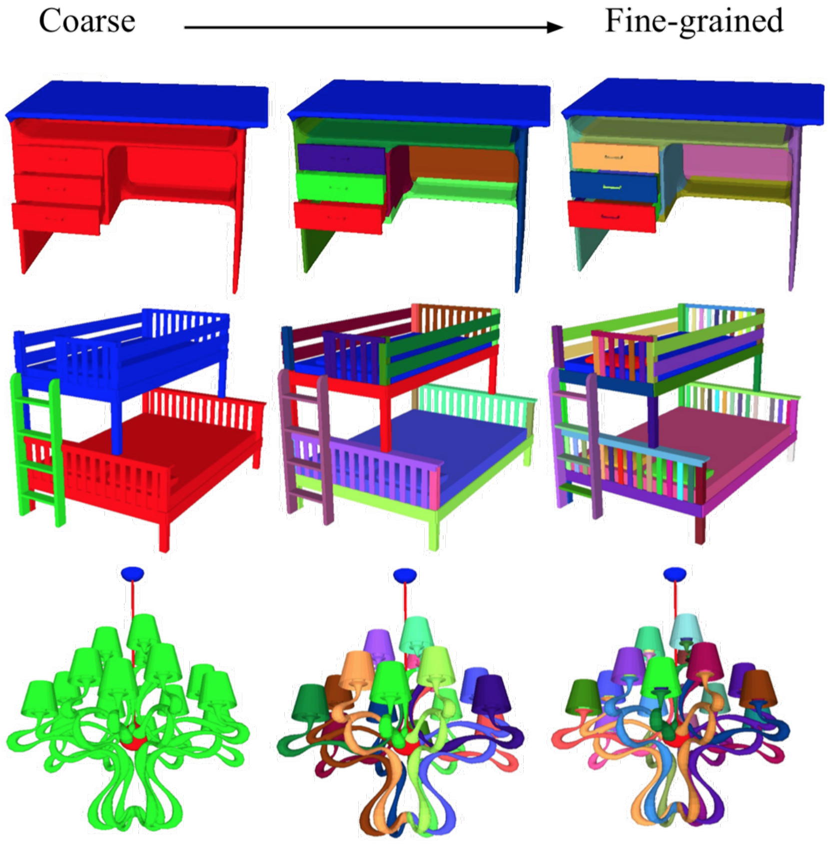
(bottom rows includes ***out-of-training*** classes)

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

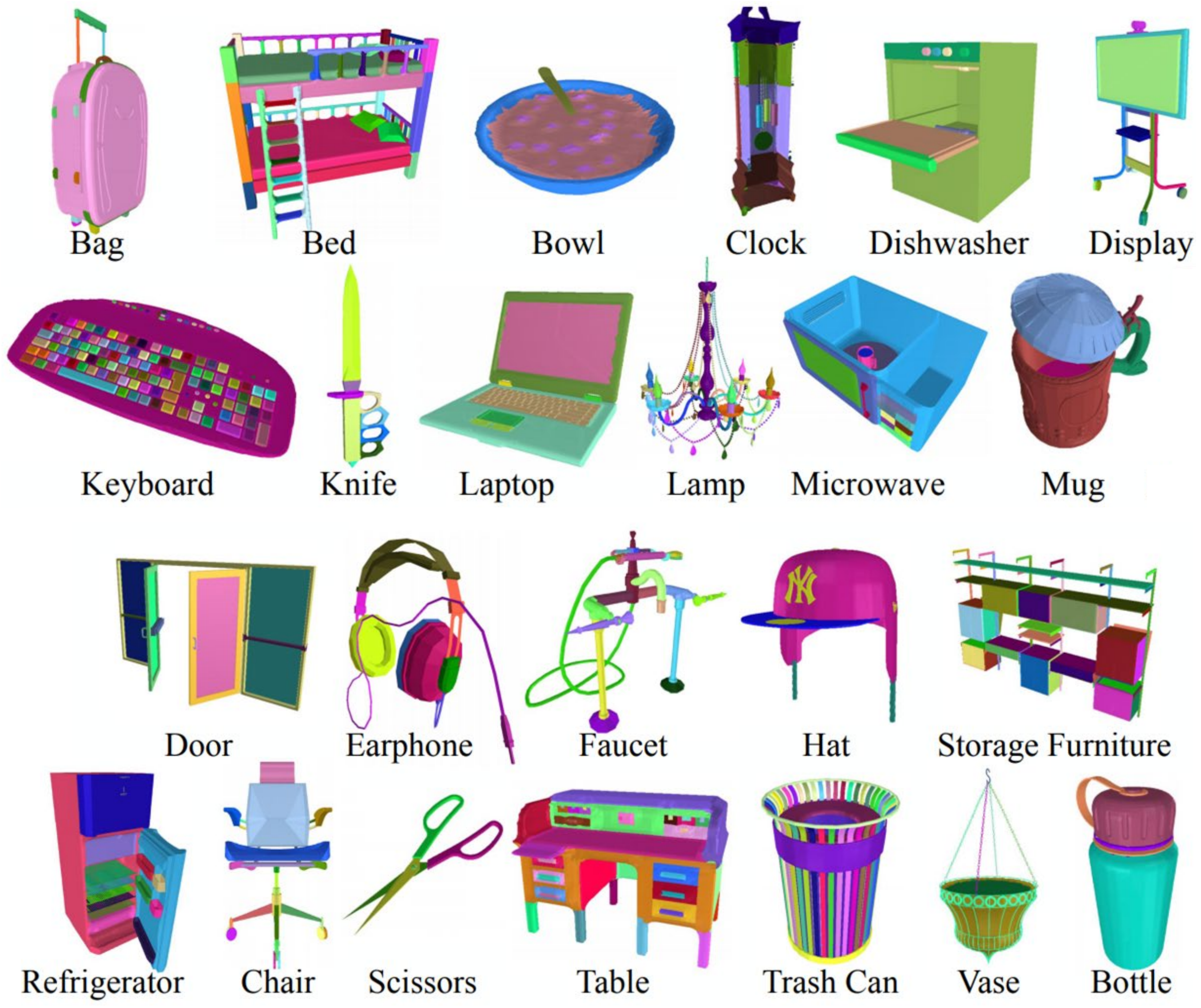
- Dataset
  - Fine-grained Parts
  - Hierarchical Segmentation
  - Instance-level Segmentation
  - Consistent Semantics
- Statistics
  - 24 Object Categories
  - 26,671 Different Shapes
  - 573,585 Different Parts
  - Avg 18 Part/shape, Max 230



PartNet Dataset

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

Based on Curated Part Hierarchies



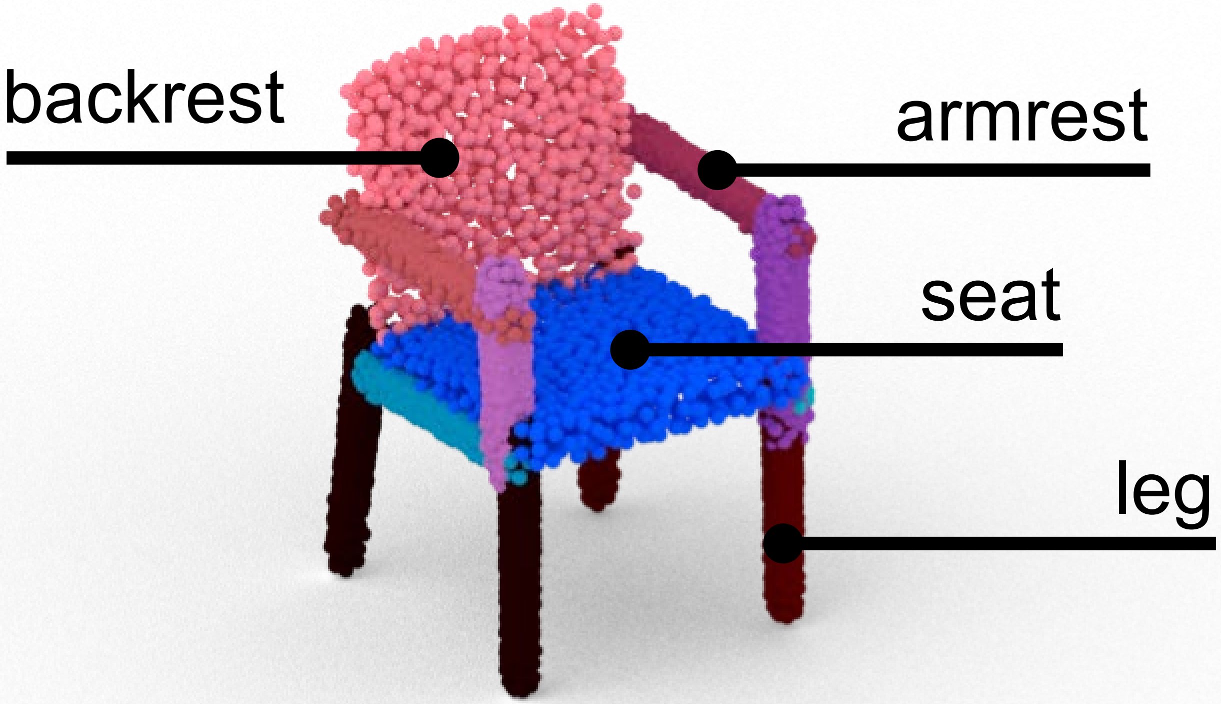
Latent Representations for Shape Structure and Structural Differences

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

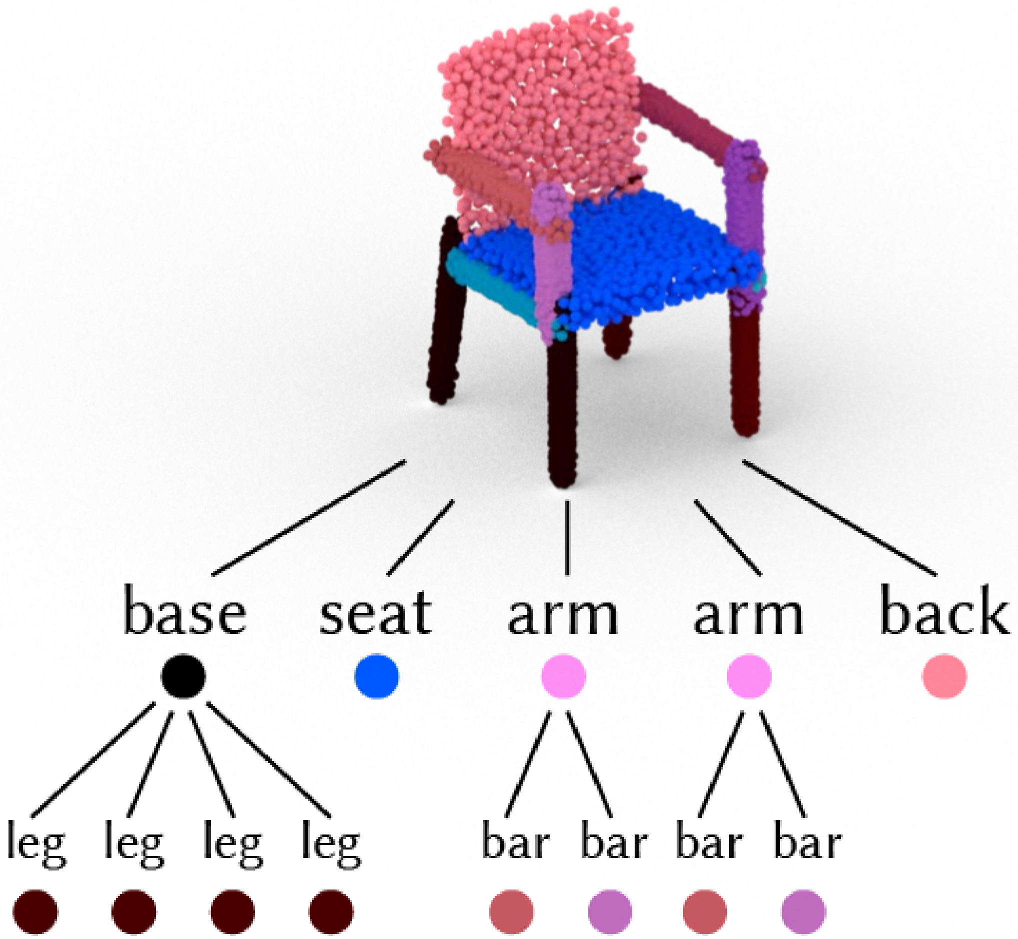
[K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, L. Guibas; ’20]

Geometry and Structure

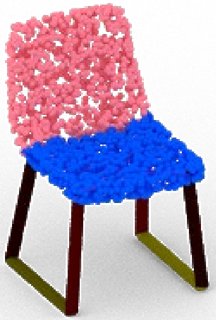
Geometry and Structure

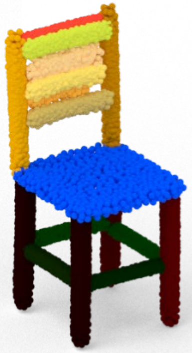
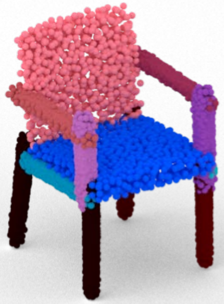


Structure: Part Hierarchy



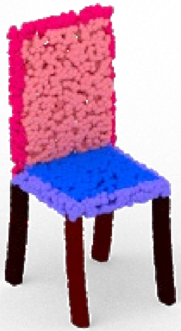
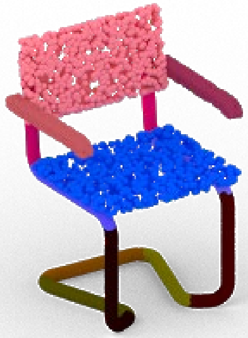
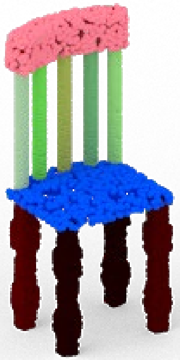
Goal: A Smooth, Explorable Shape Space

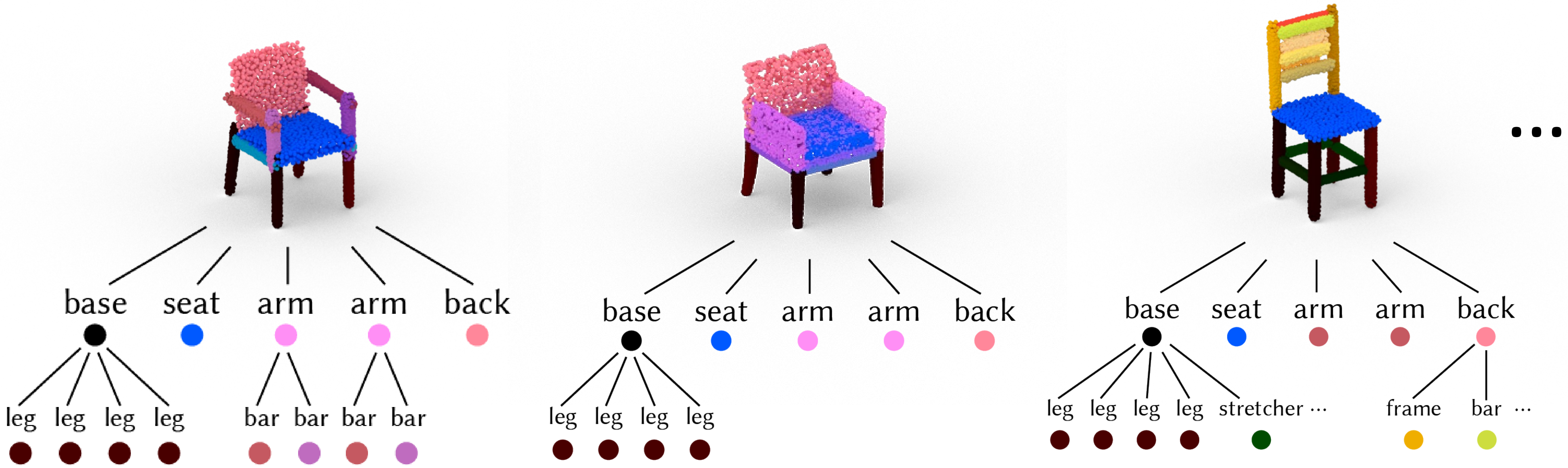



A smooth, common shape space allows for interpolation, generation, exploration, …

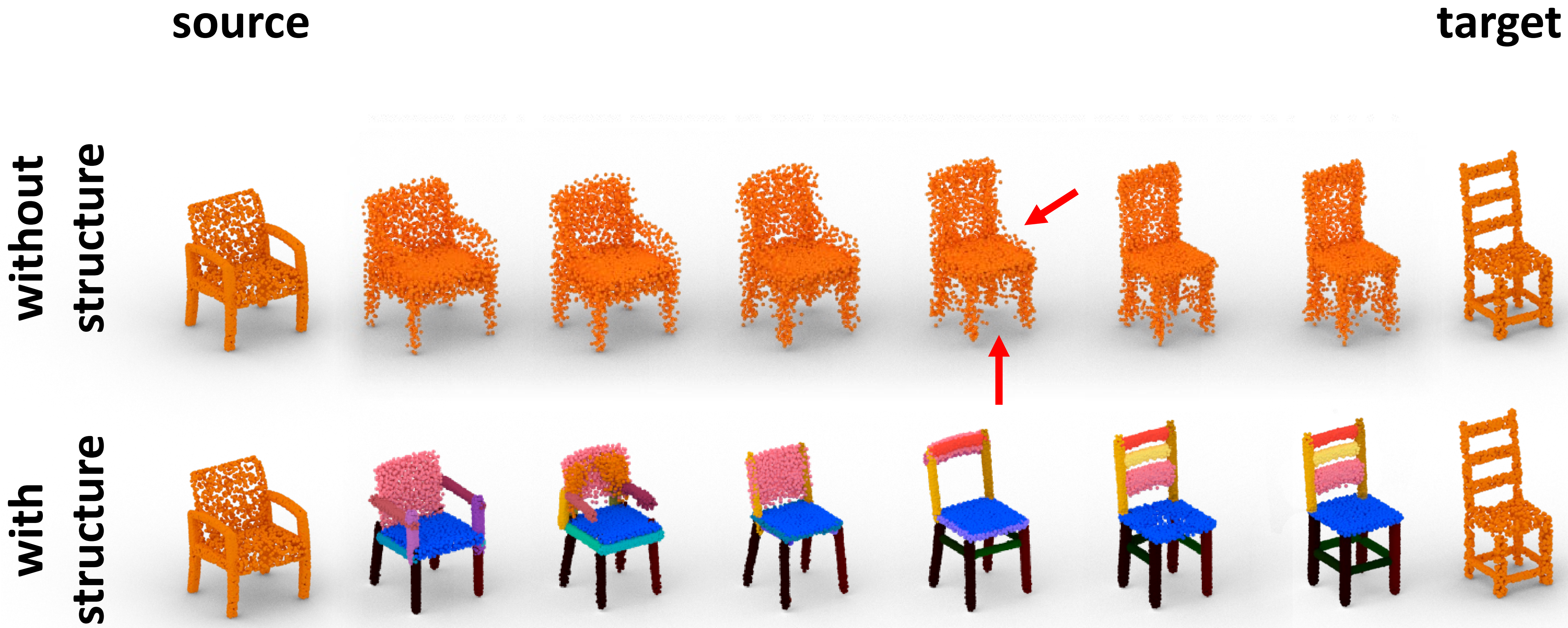
… of both **geometry** and **structure**

Structural Consistency

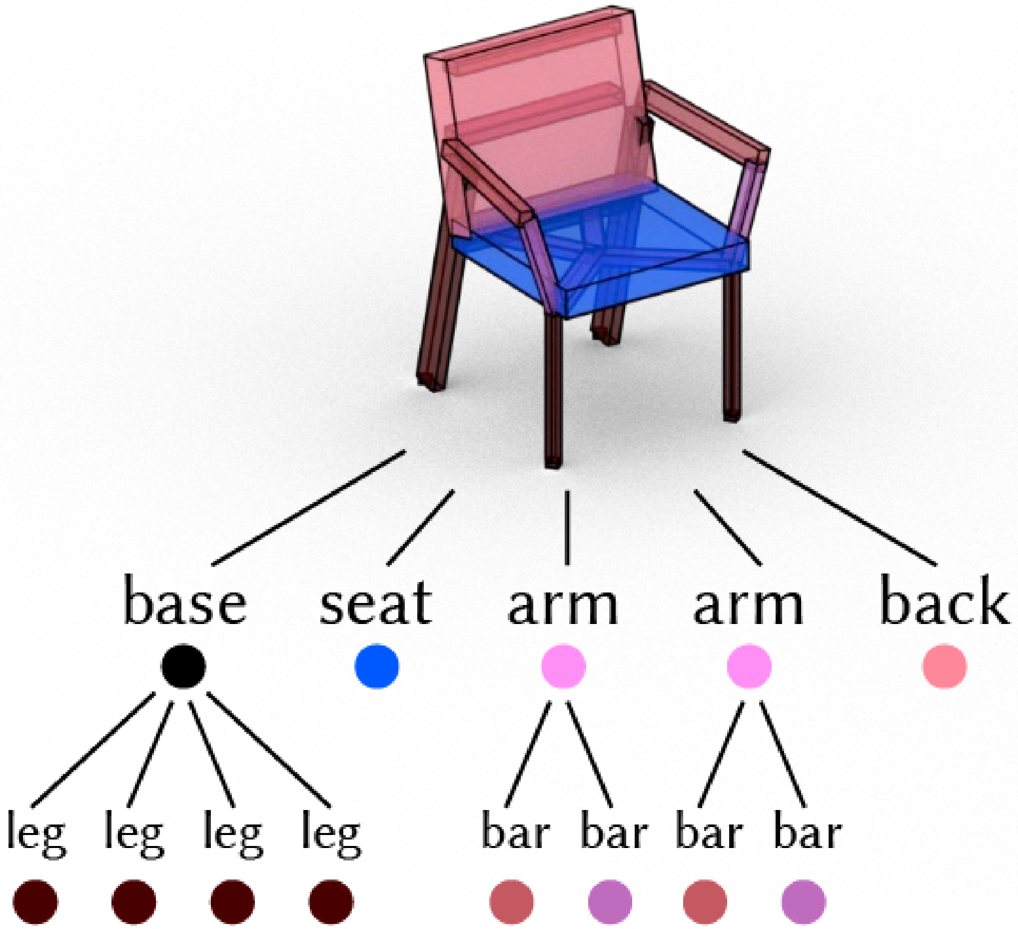


With vs. Without Structure



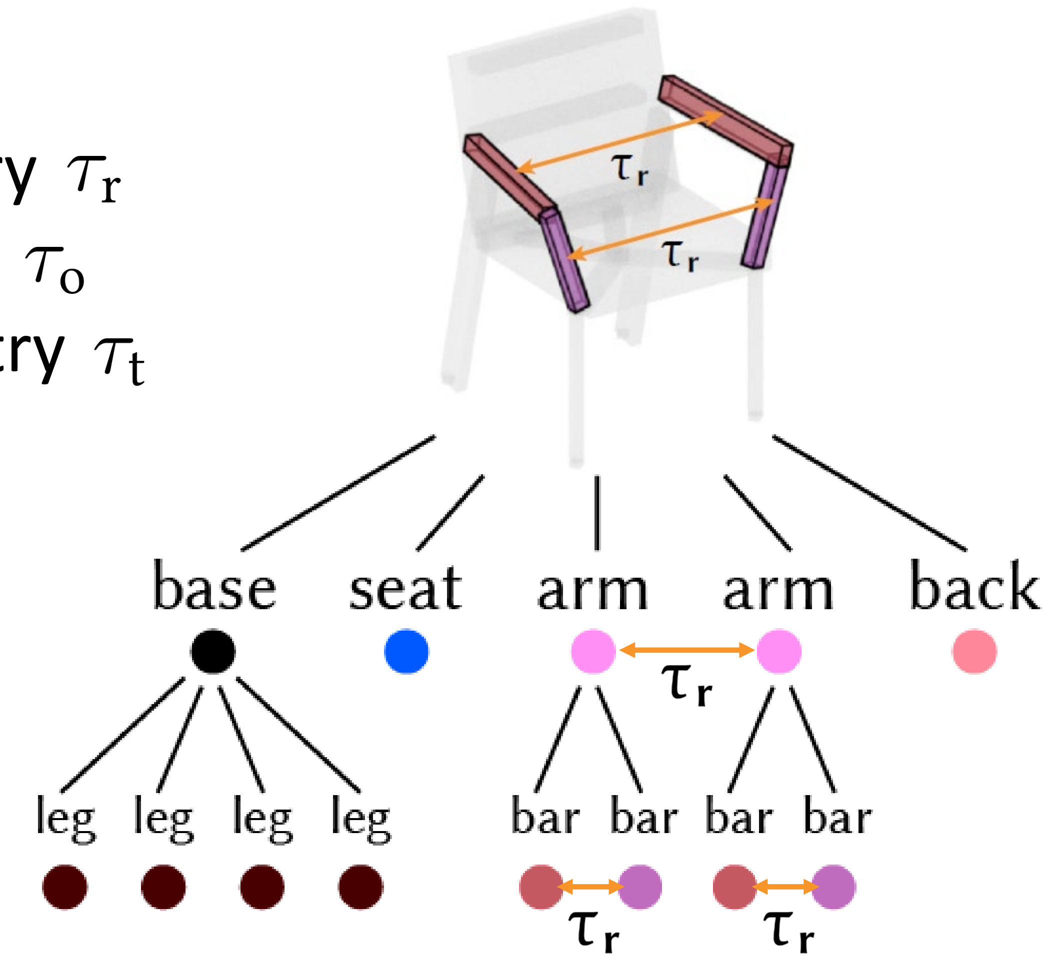
Object Representation: Part Geometry

Object Representation: Part Structure

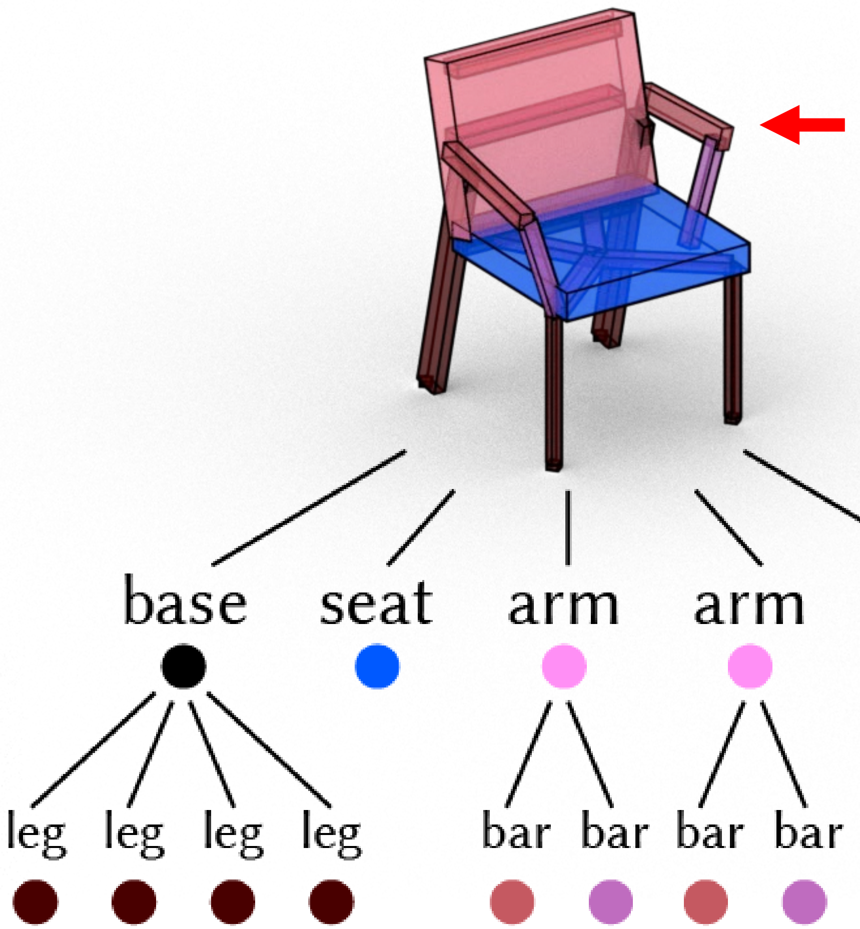


Object Representation: Sibling Relationships

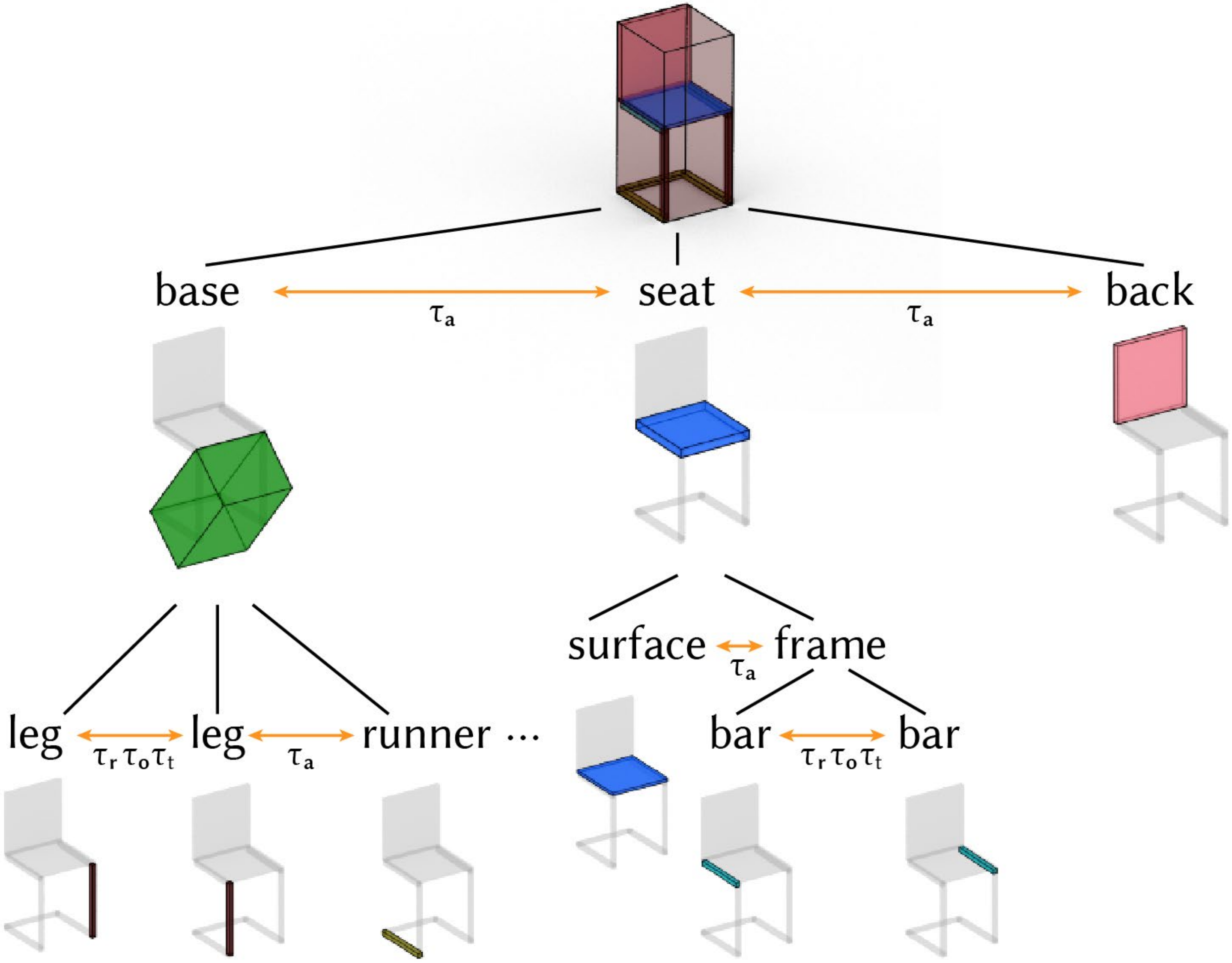
Reflectional Symmetry τ_r
Rotational Symmetry τ_o
Translational Symmetry τ_t
Adjacency τ_a



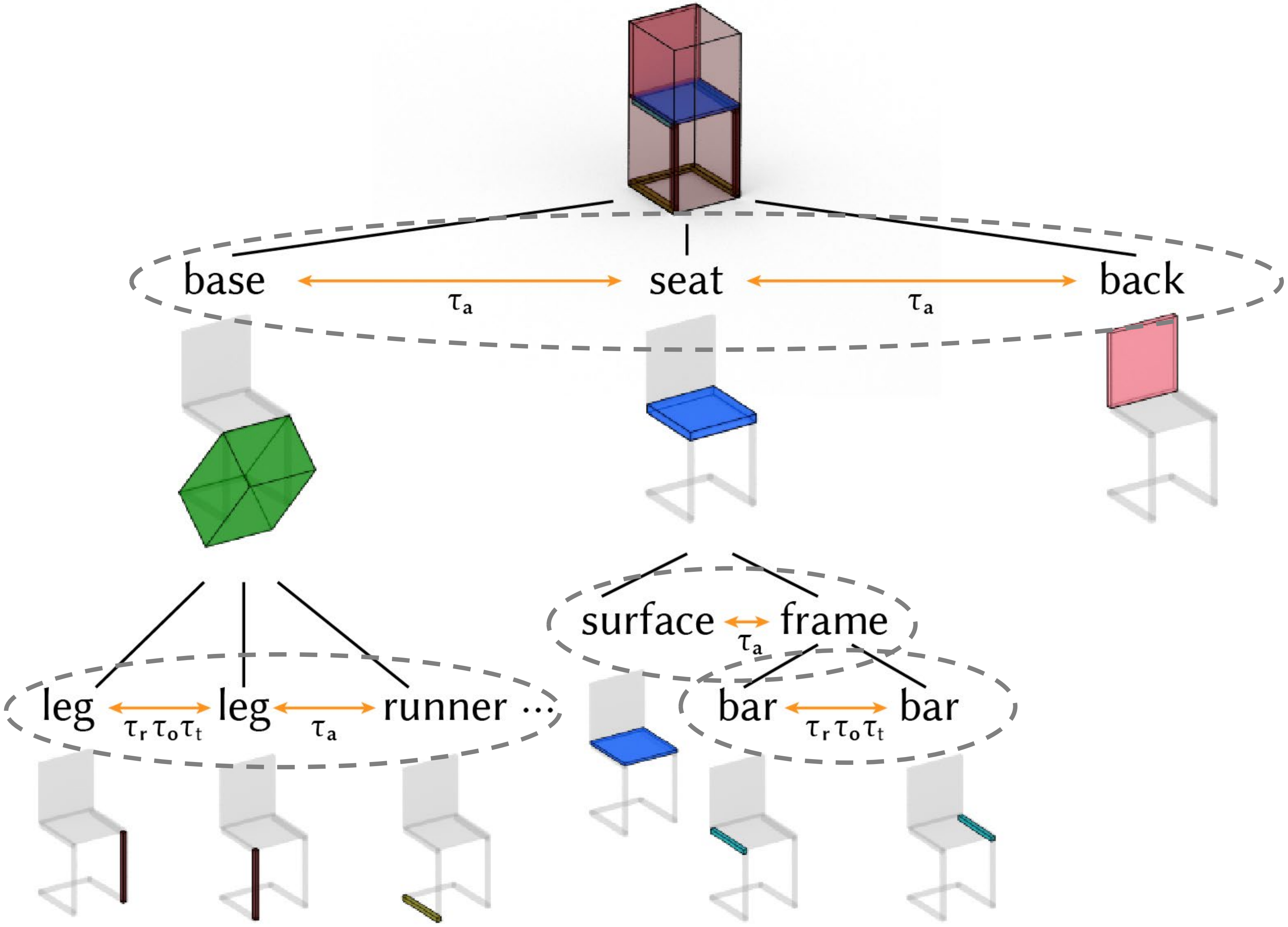
Object Representation: Part Structure



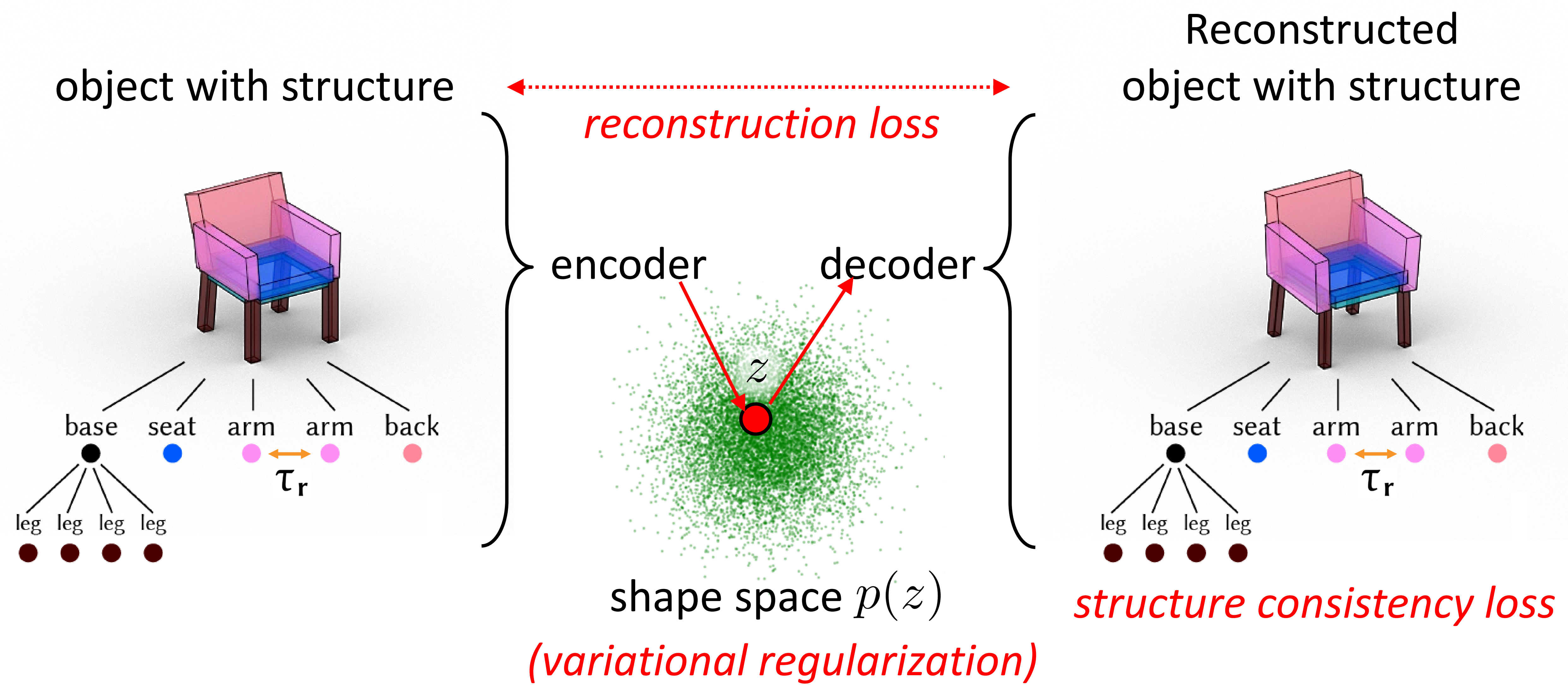
Object Representation: Example



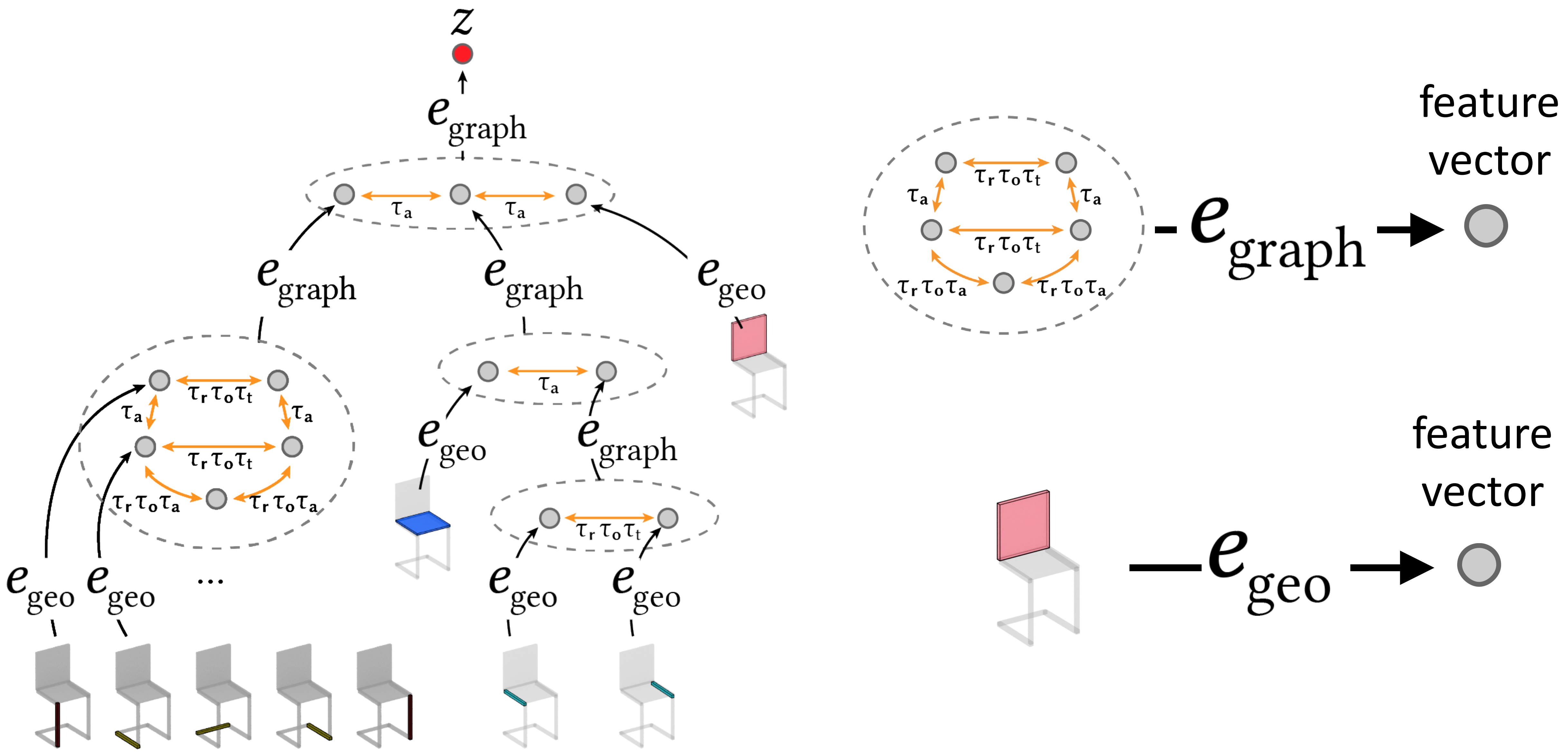
A Hierarchy of Graphs



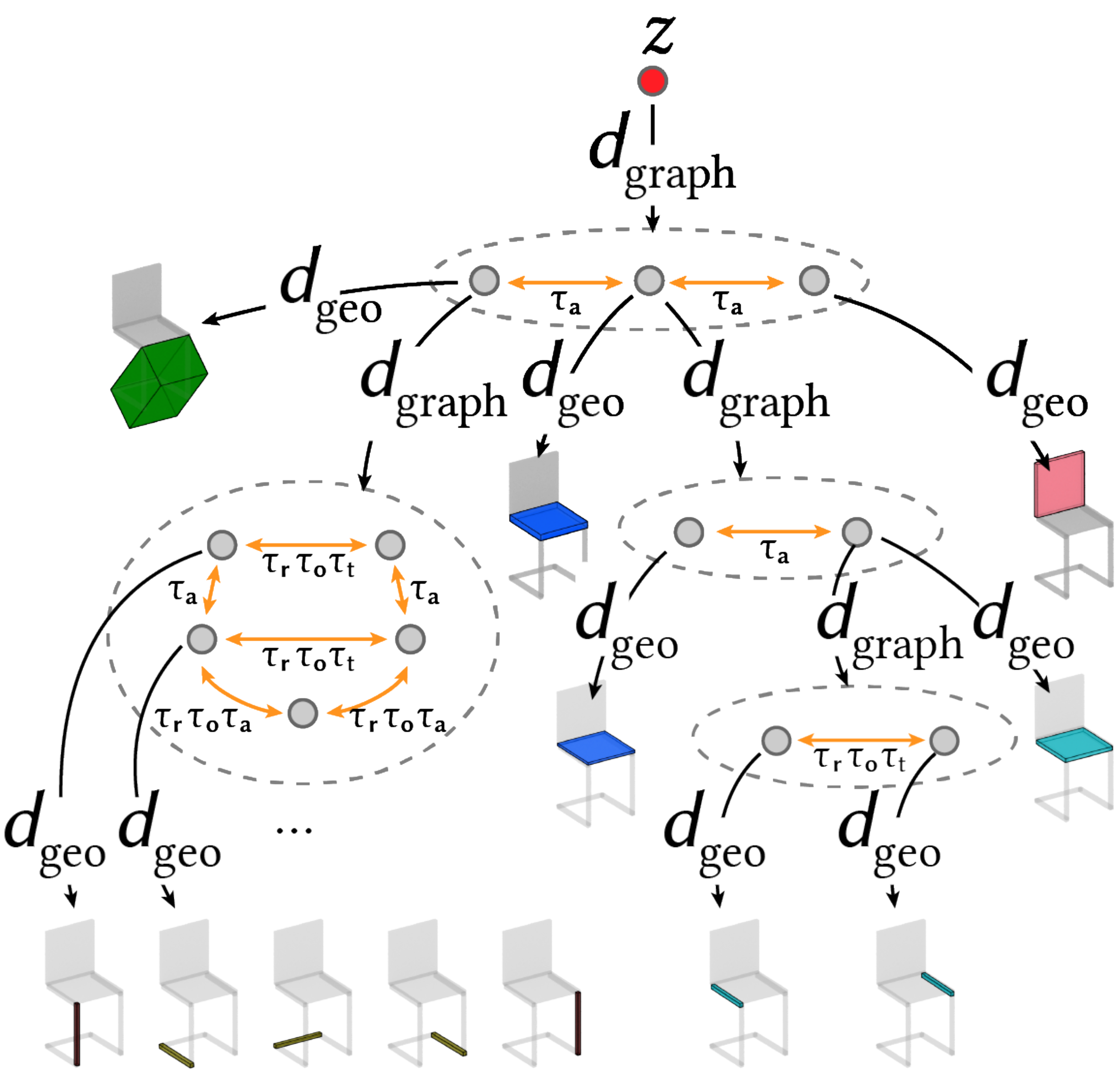
Architecture Overview: VAE Training

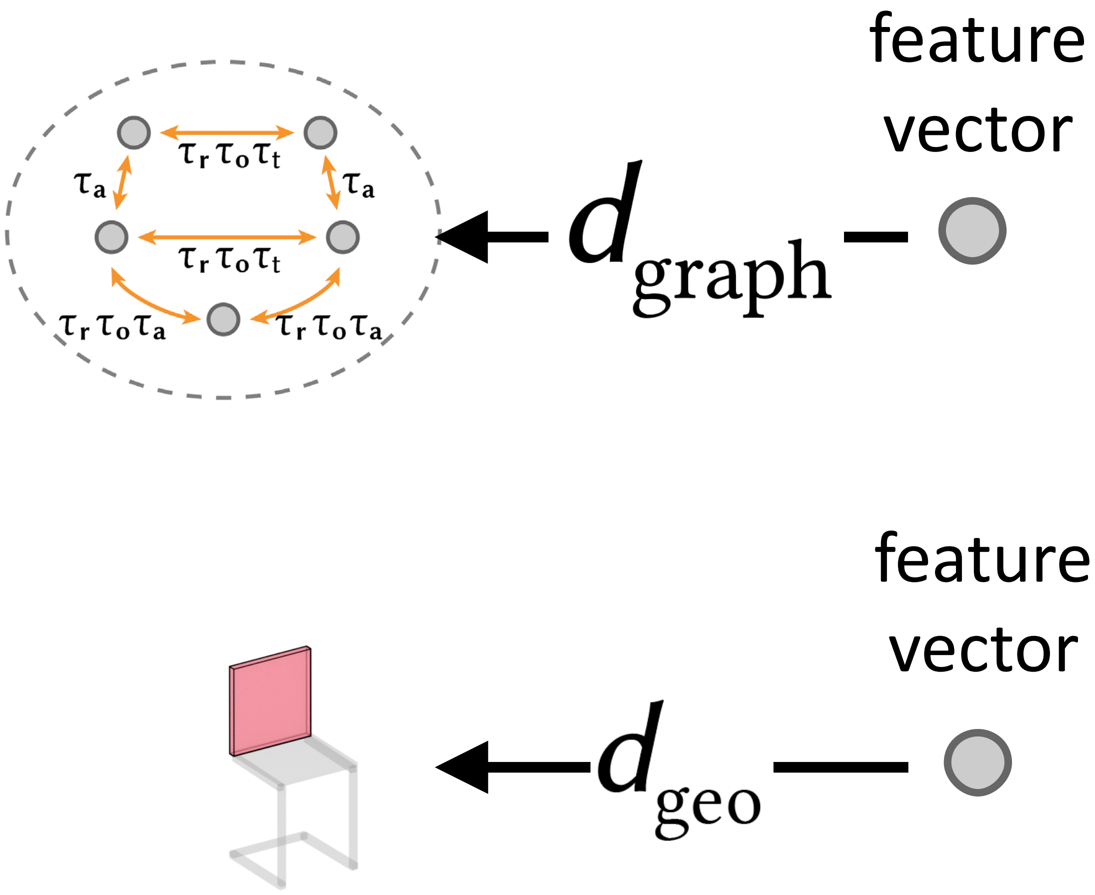


Hierarchical Graph Encoder

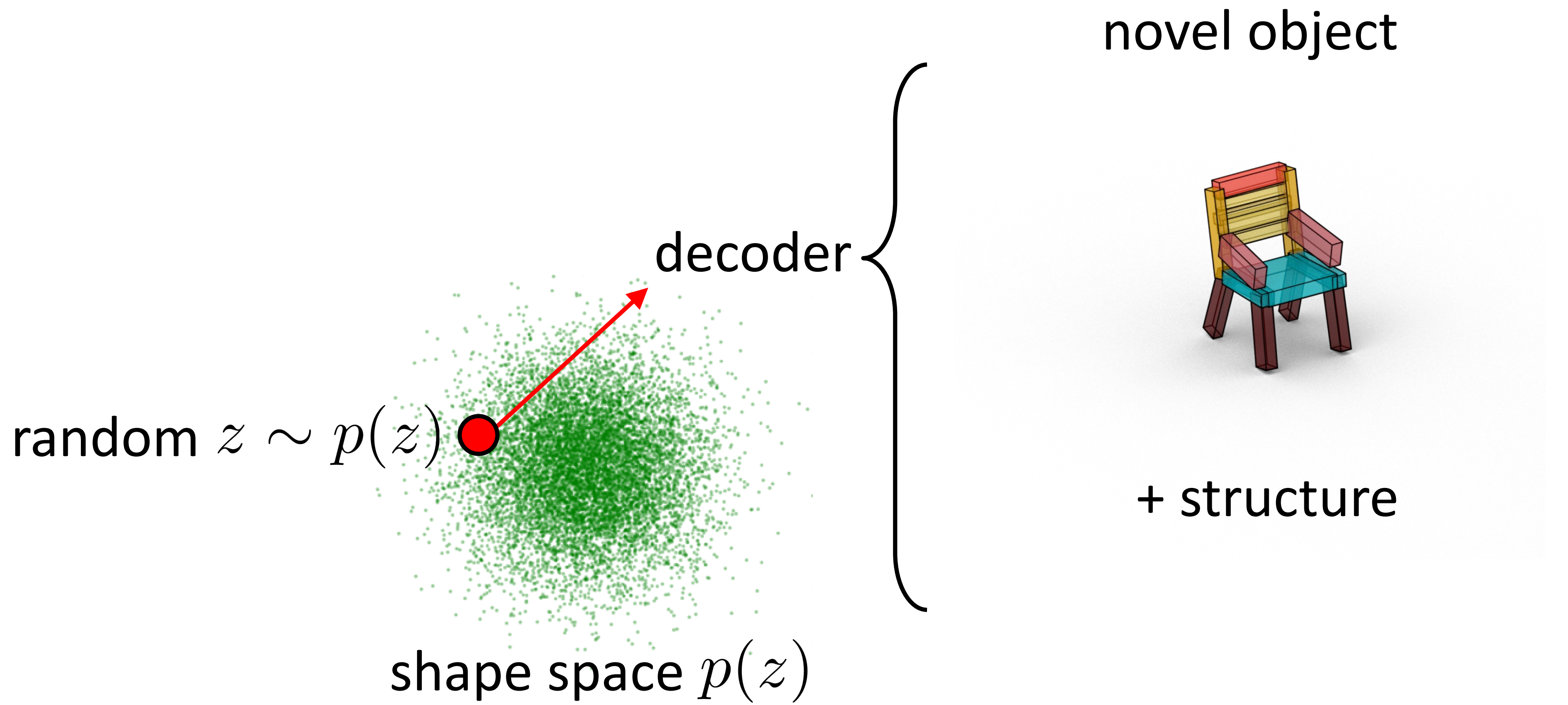


Hierarchical Graph Decoder





Application 1: Generation



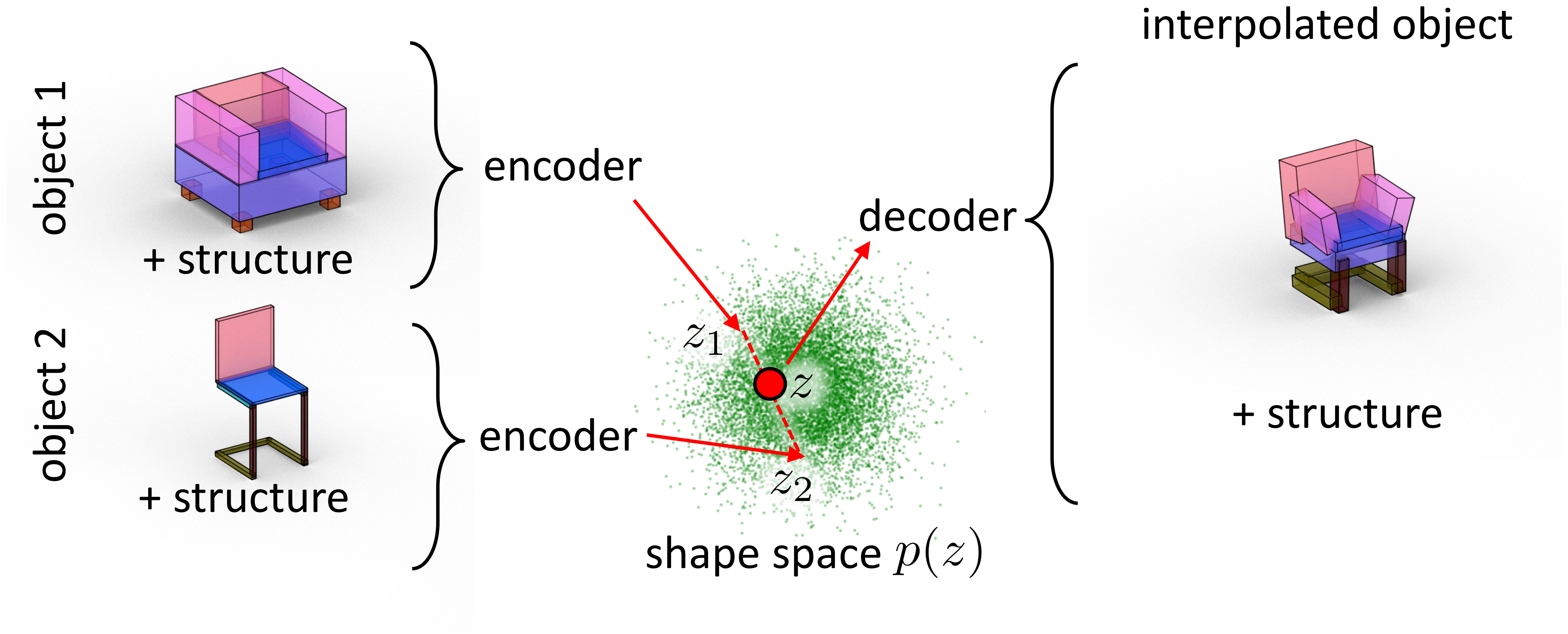
Generation

Generation

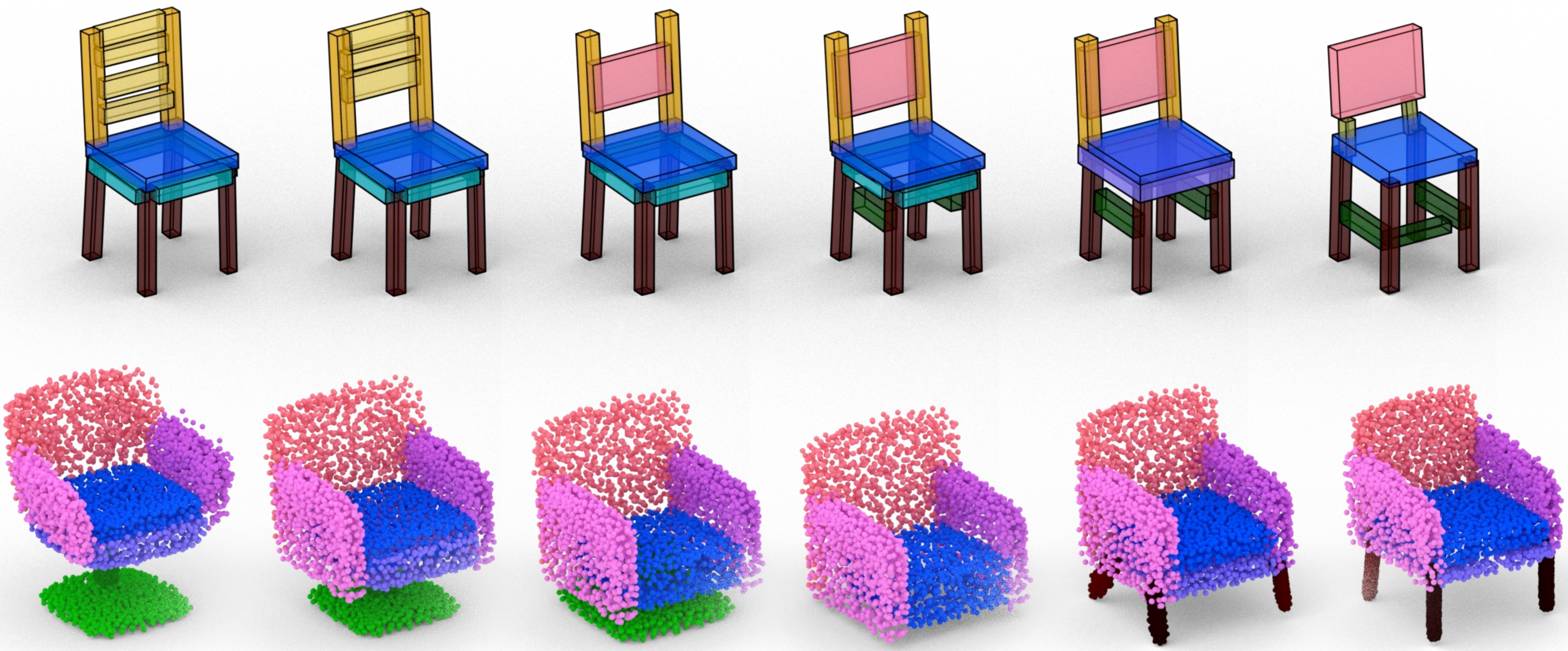
Novelty

generated closest training samples

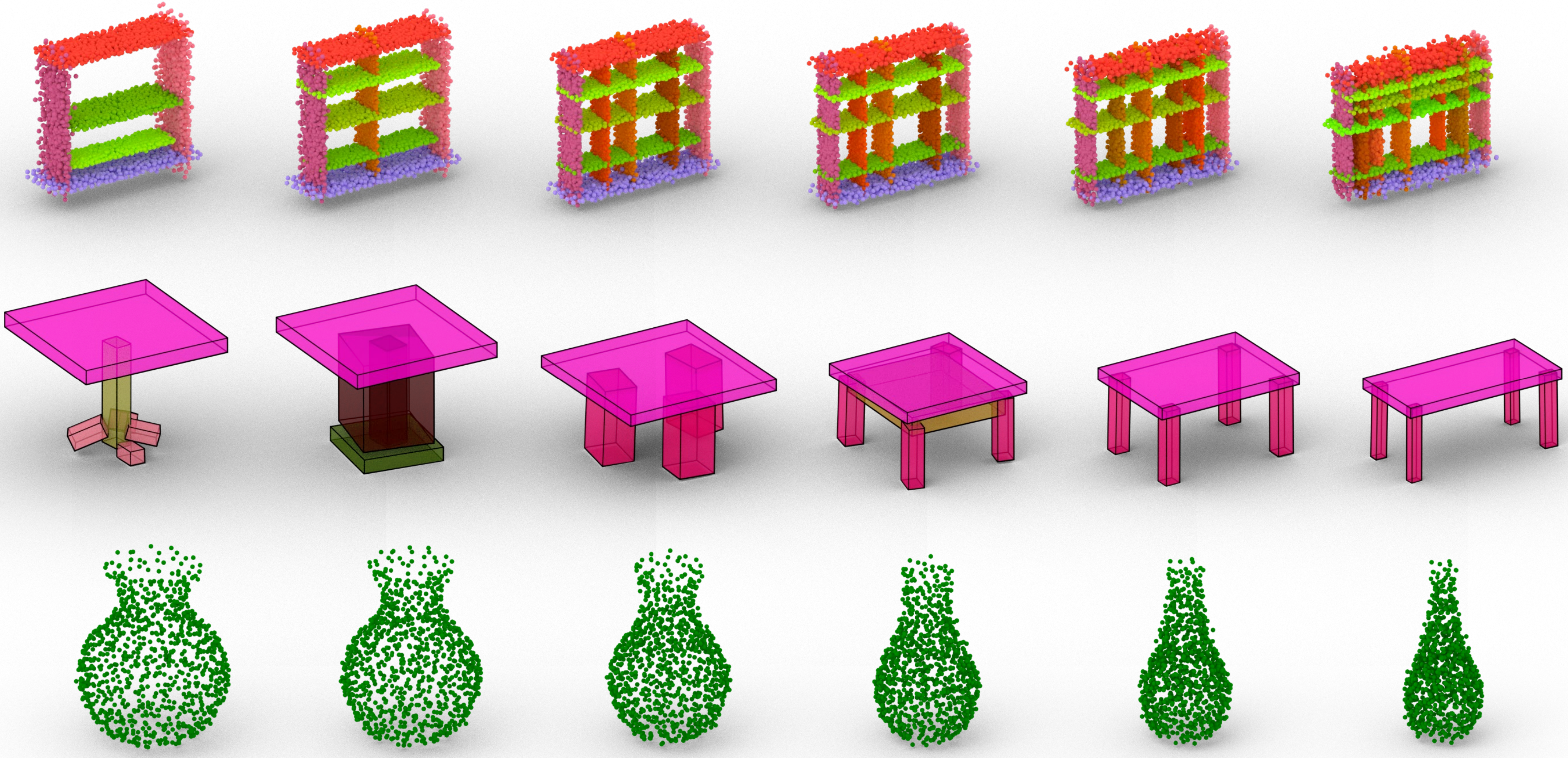
Application 2: Interpolation



Interpolation



Interpolation

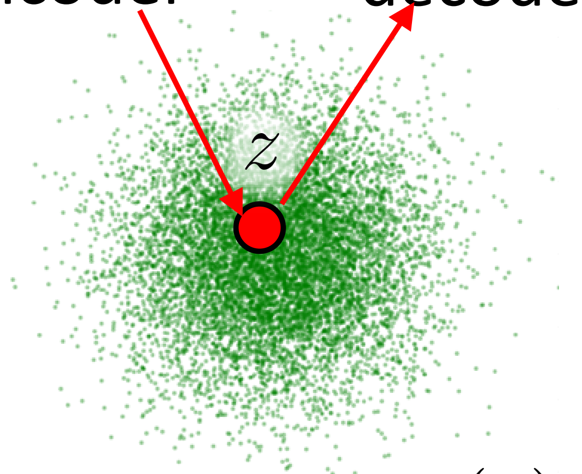


Application 3: Scan Abstraction

partial scan

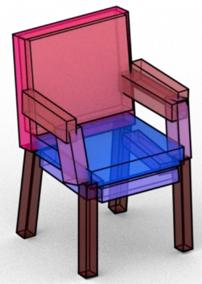
point cloud encoder

decoder



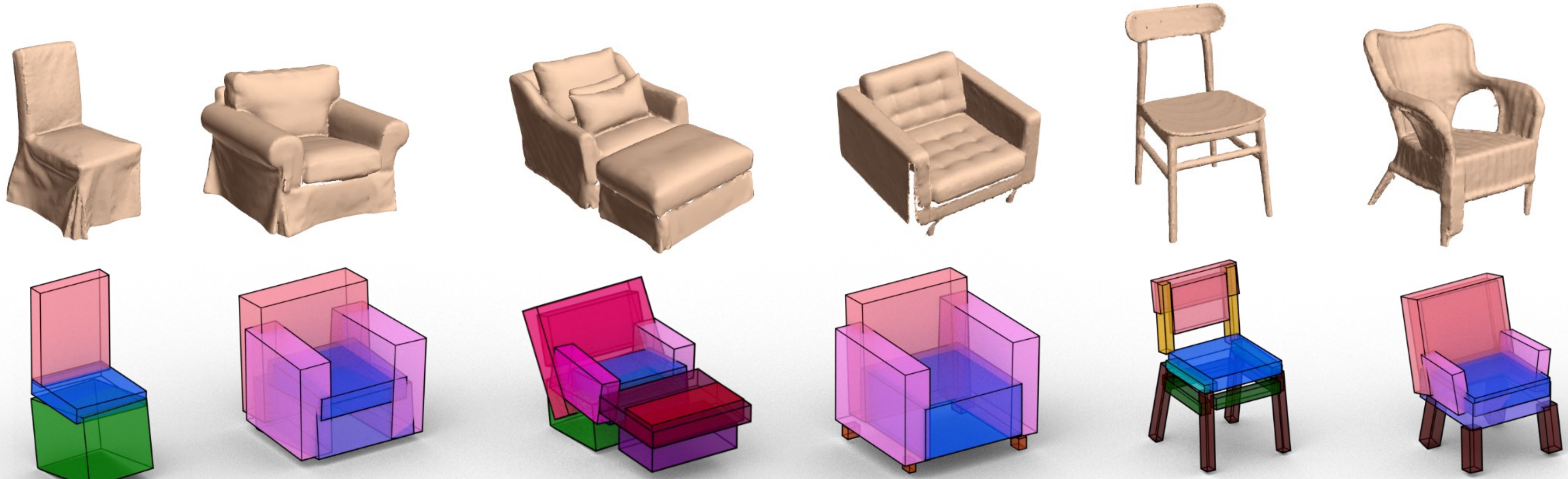
shape space $p(z)$

reconstructed object

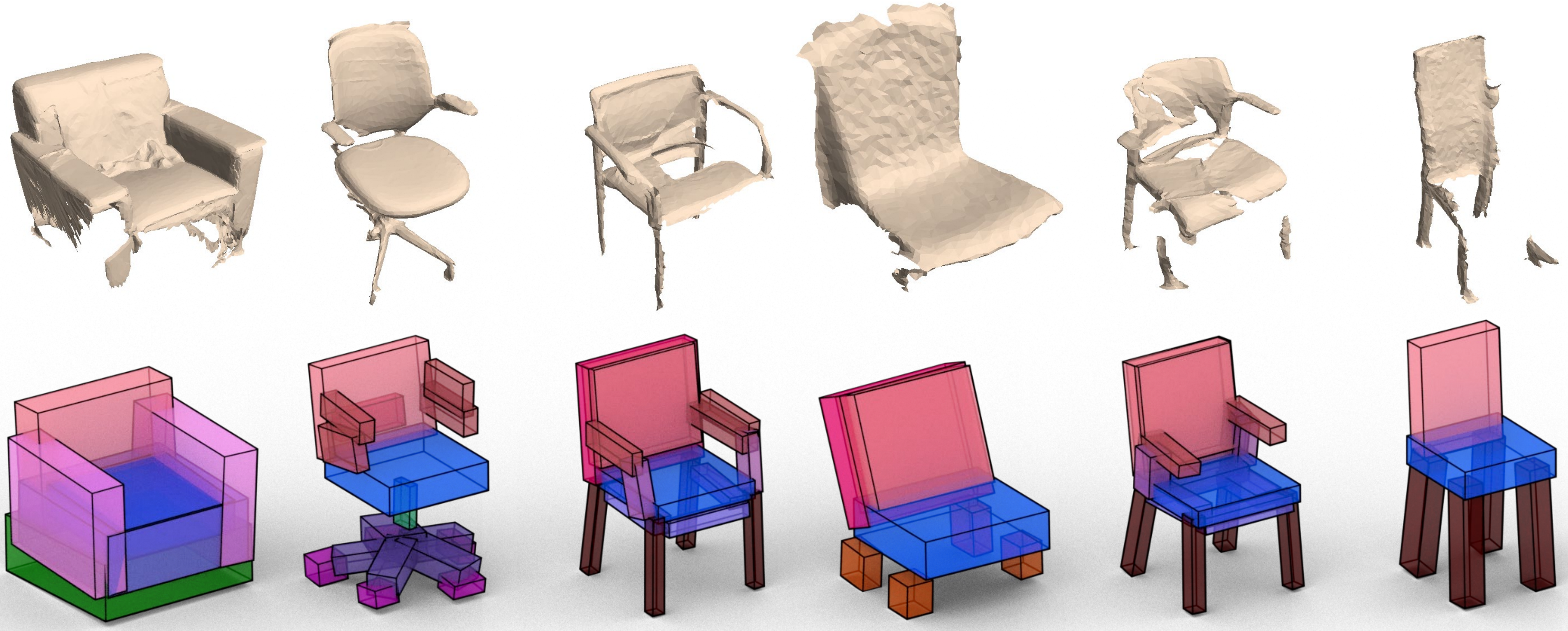


+ structure

Abstraction of Full Scans

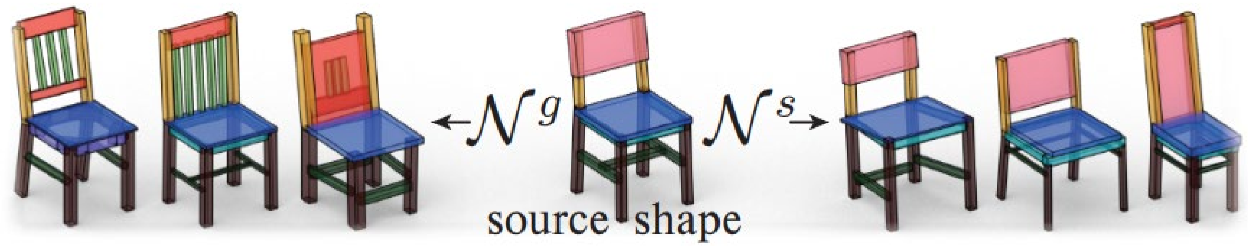


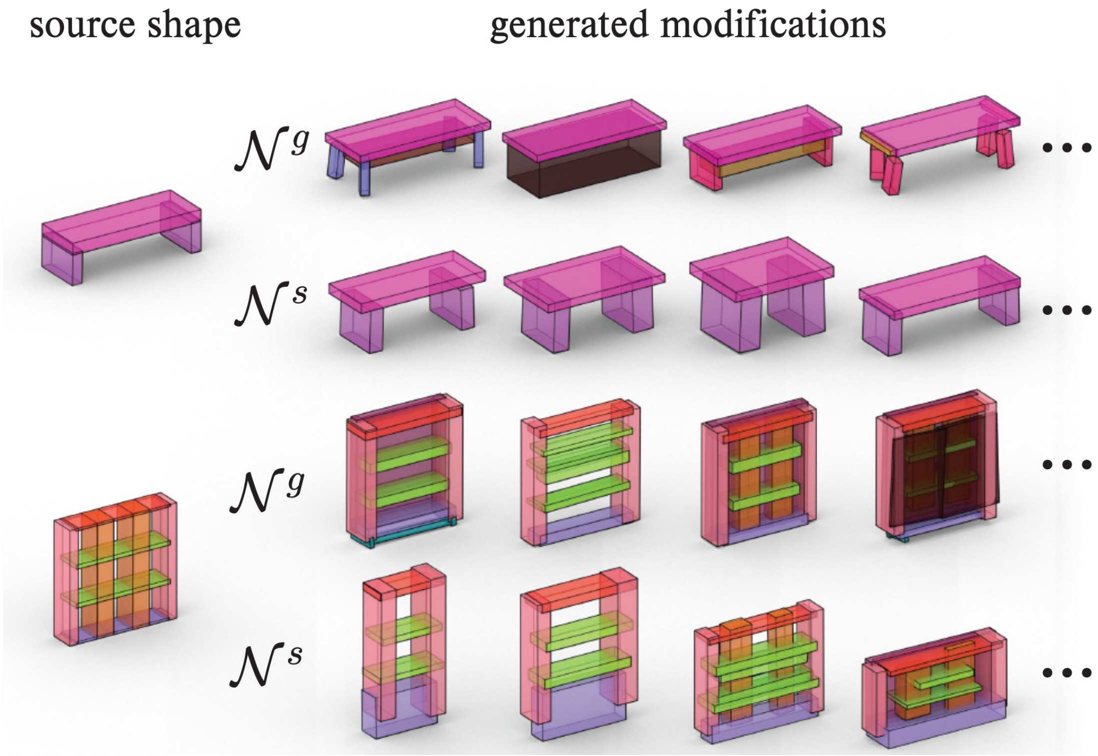
Abstraction of Partial Scans

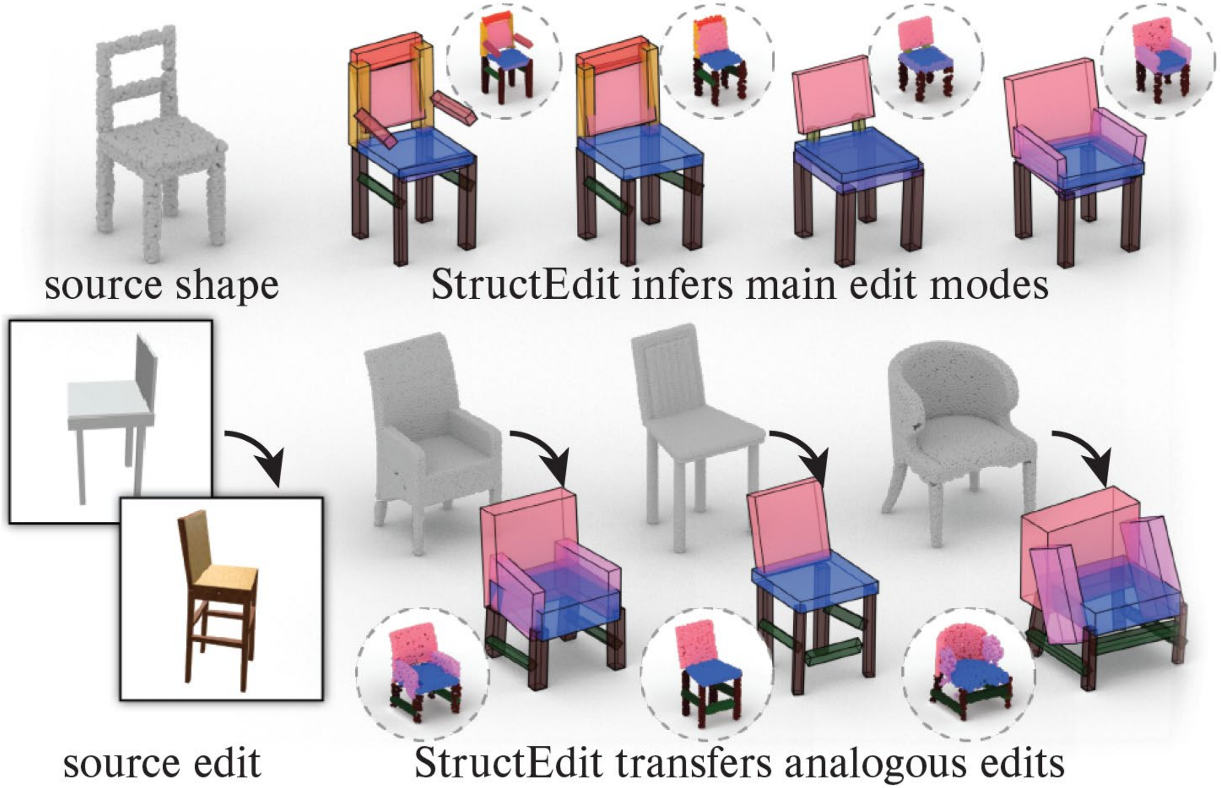


Learning Shape Variations: Geometric and Structural

Two Types of Shape Neighborhoods

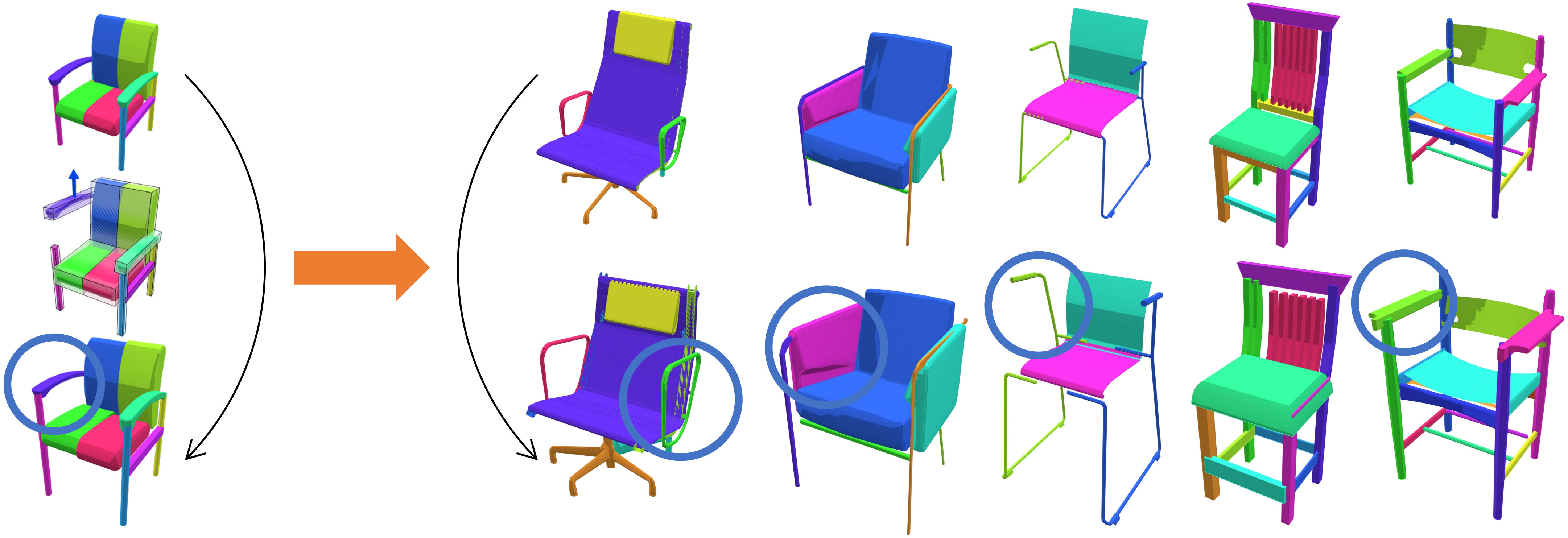






Learning and Exploiting Correlations in Deformations

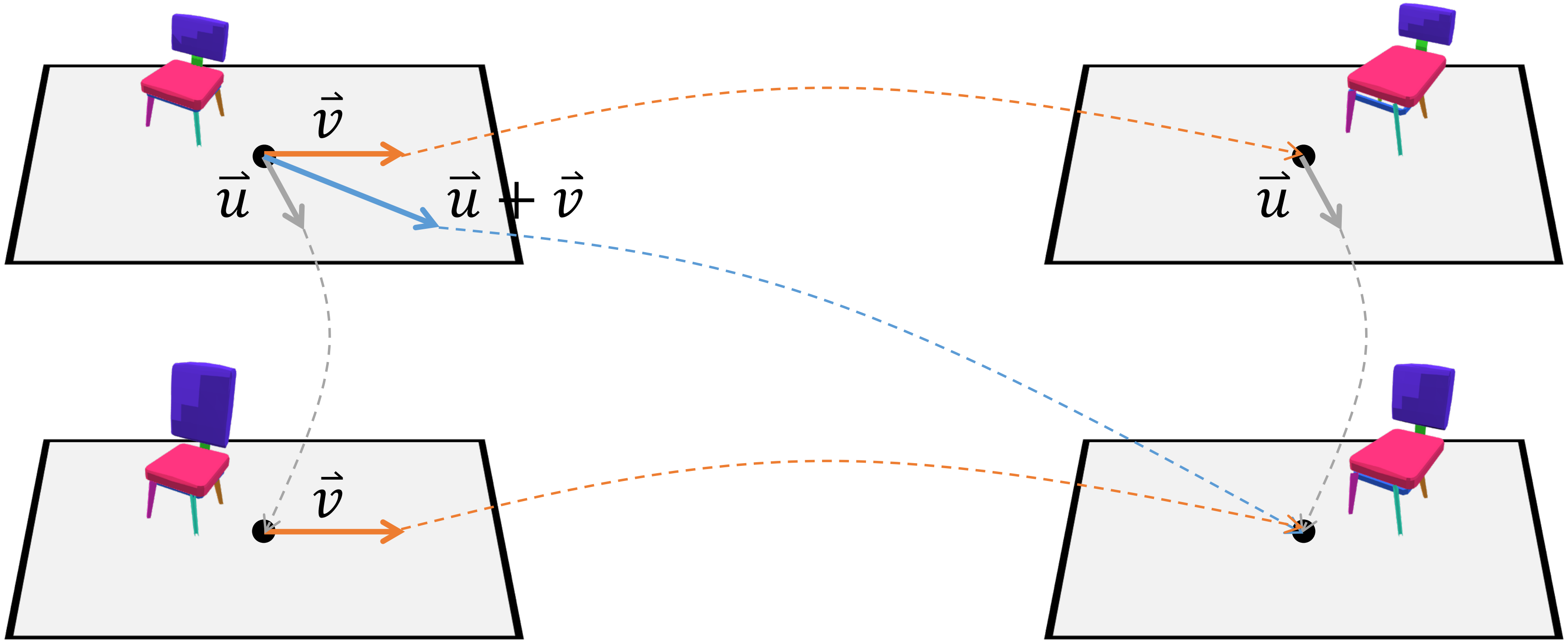
Transfer deformations across shapes without correspondences.



[M. Sung, Z. Jiang, P. Achlioptas, N. Mitra, L. Guibas; '20]

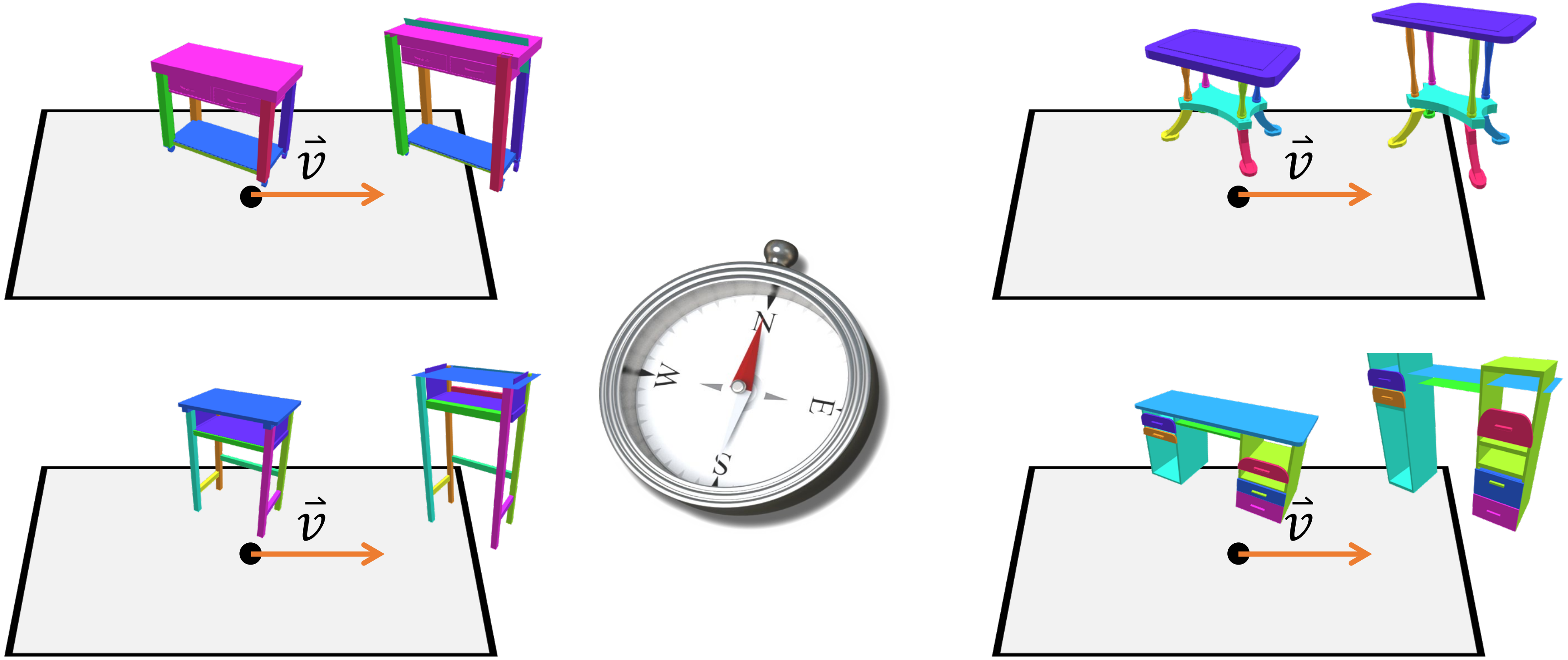
Path Invariance

We want to reach to the same destination, no matter which route we choose.



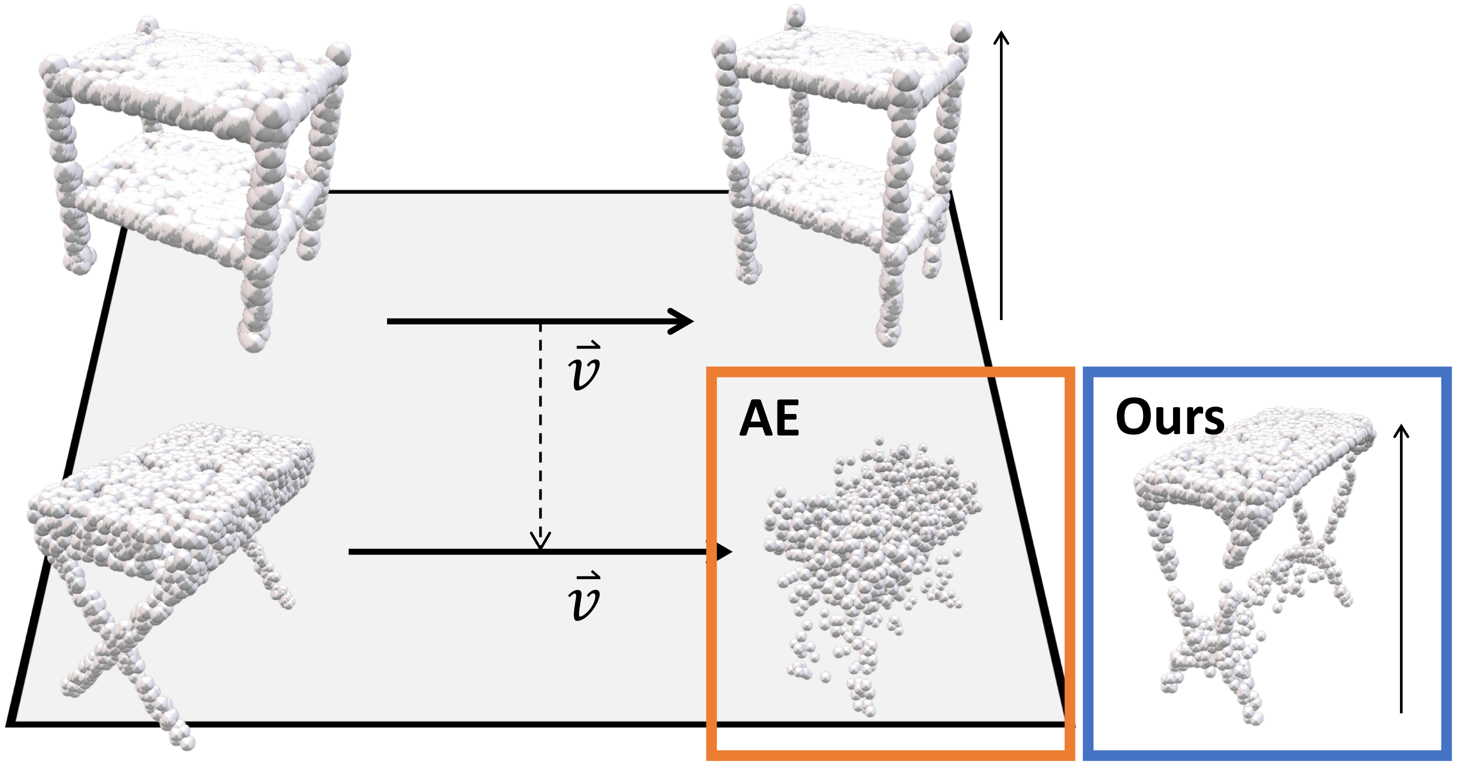
Consistency

We aim to have a latent vector meaning the same thing **everywhere**:
e.g., $\vec{v} = \langle 1, 0, \cdots, 0 \rangle$ Indicates "elongate legs".



Autoencoder Latent Space

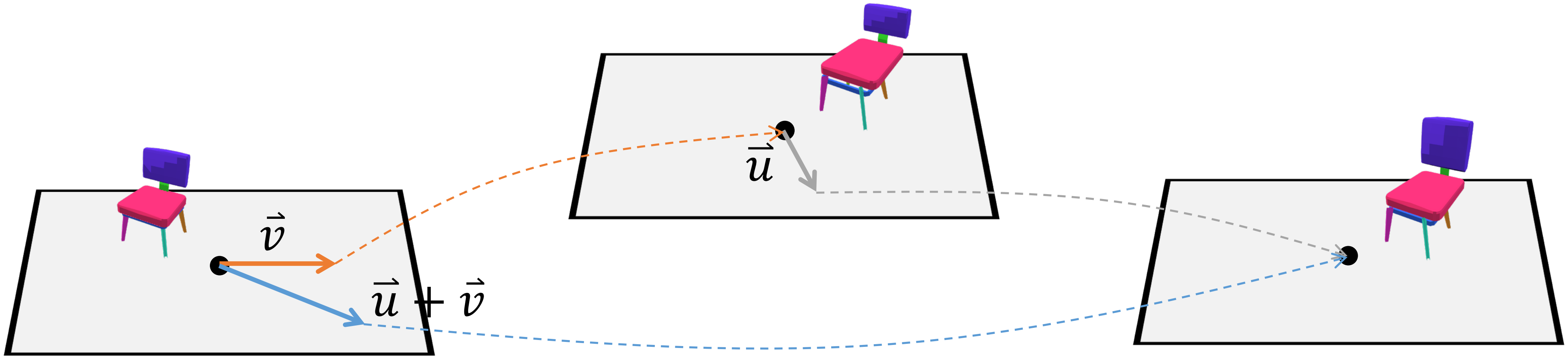
The axes of autoencoder latent spaces are not typically associated with semantically meaningful shape changes.



A Latent Space for Deformations

An affine latent action space satisfies the following property:

Additive action: $\mathrm{x} \in X,\ \vec{u}, \vec{v} \in V,\ (x \oplus \vec{u}) \oplus \vec{v} = x \oplus (\vec{u} + \vec{v})$.



Autoencoder

An action defined with an antoencoder:

$$x \oplus \vec{v} := \mathcal{D}(\mathcal{E}(x) + \vec{v}).$$

does not guarantee additivity and transitivity.

- A vector $\vec{v}$ can act differently given the shape.
- Multiple vectors can be decoded to the same deformation.

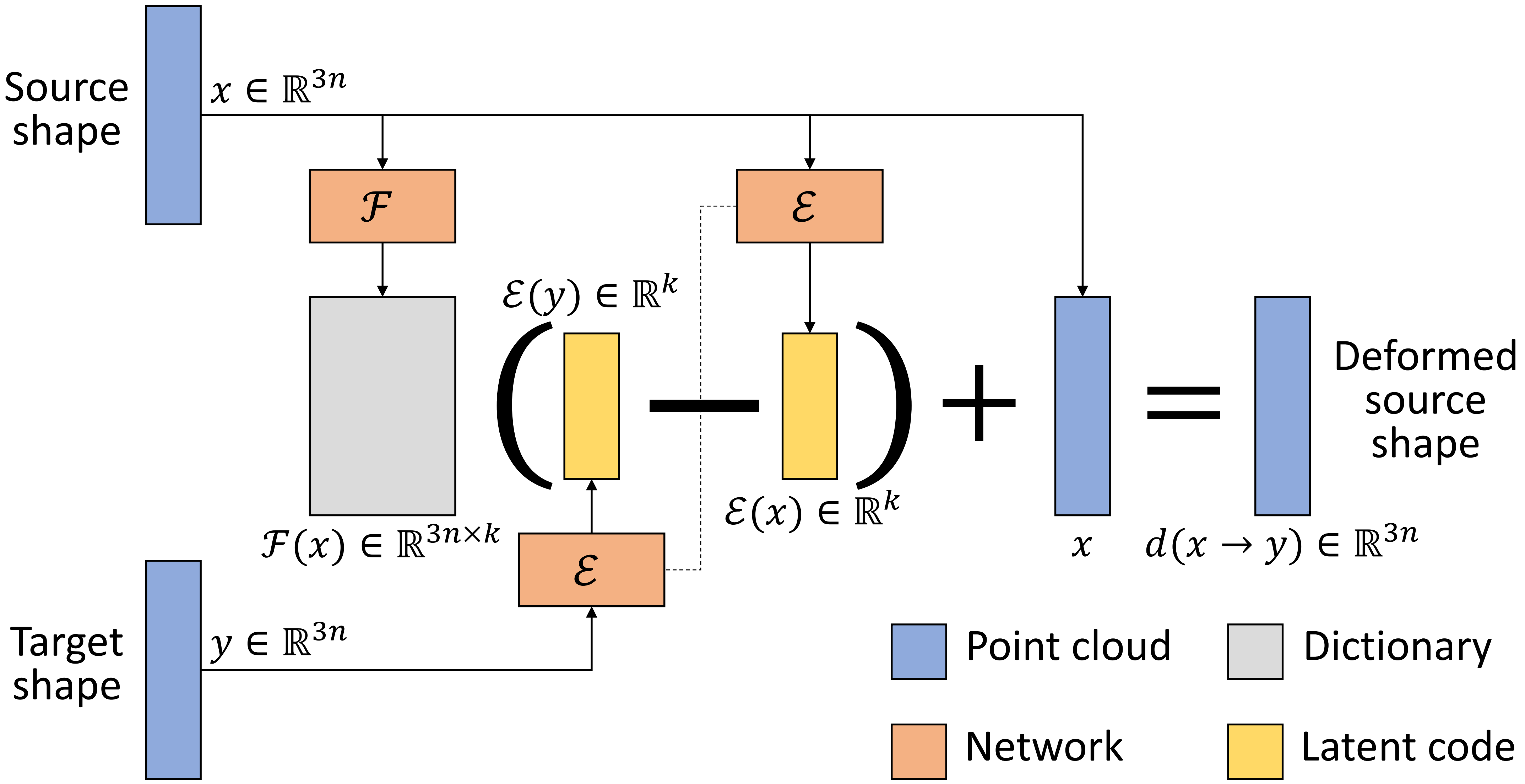
Another Solution

We predict the deformation dictionary for each shape using another dictionary prediction network $\mathcal{F} \in \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n \times k}$.

The deformation $\mathrm{d}(x \rightarrow y)$ from shape x to y is computed as:

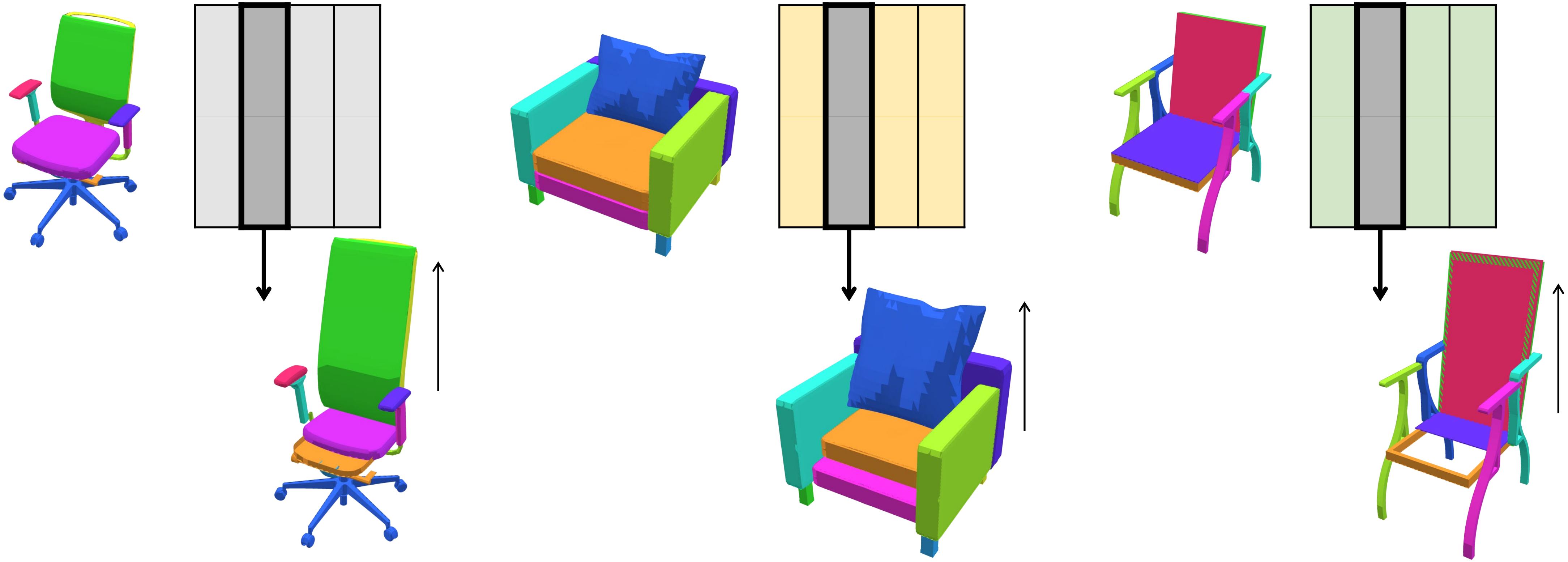
$$\mathrm{d}(x \rightarrow y) = \mathcal{F}(x)\big(E(y) - E(x)\big) + x.$$

Neural Network



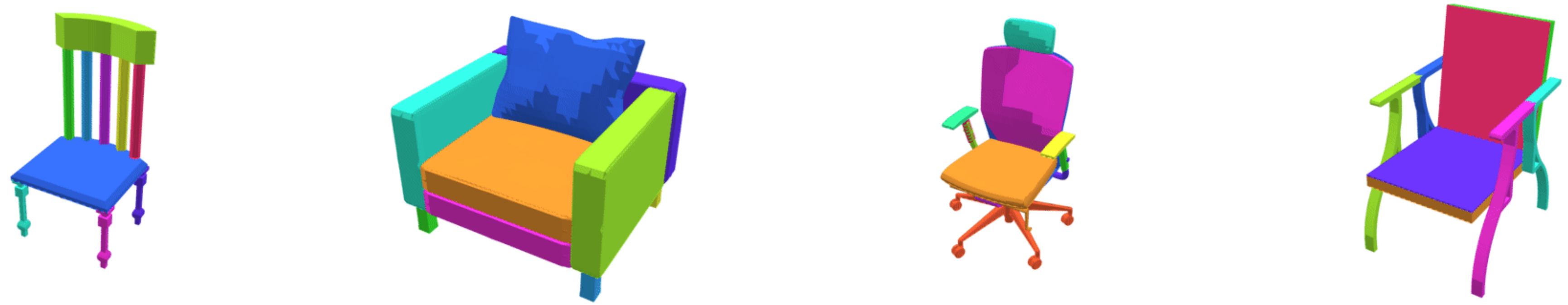
Consistency

Consistency across the deformation dictionaries **emerges** during training.



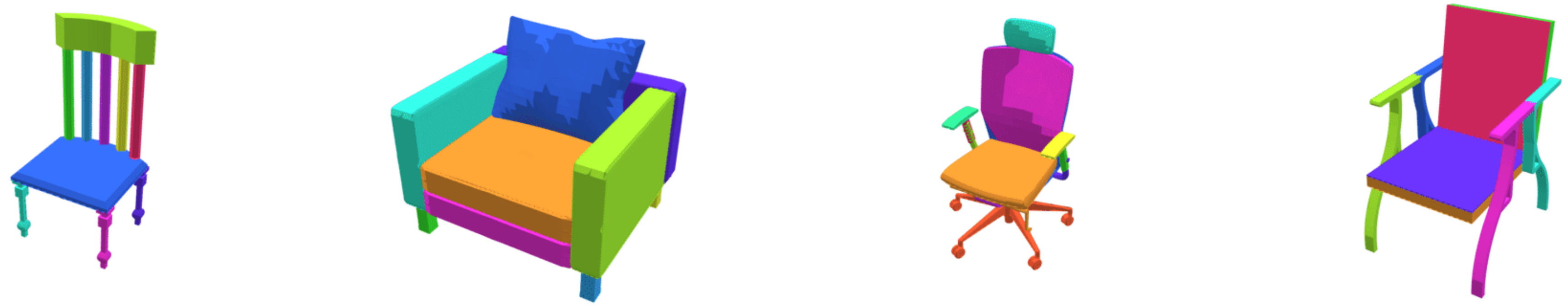
Deformation Dictionary

Translating **seat** along the up/down direction.



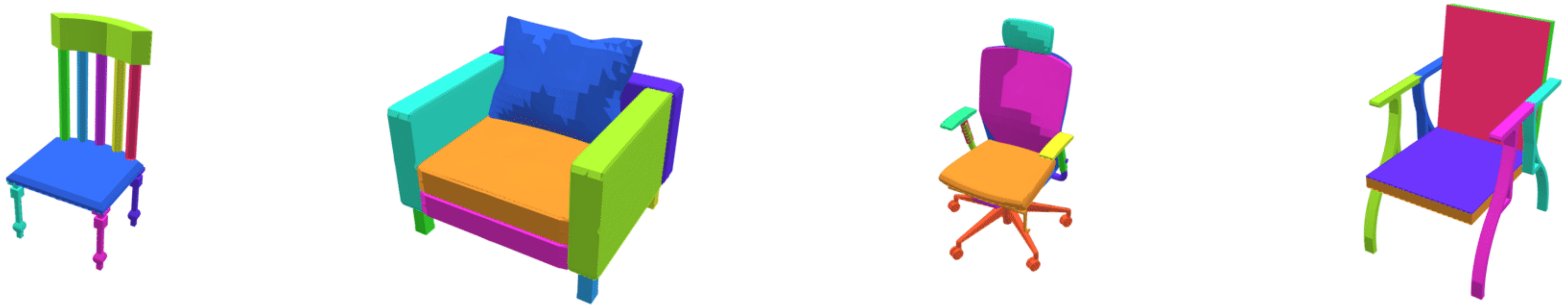
Deformation Dictionary

Translating **back** along the front/back direction.

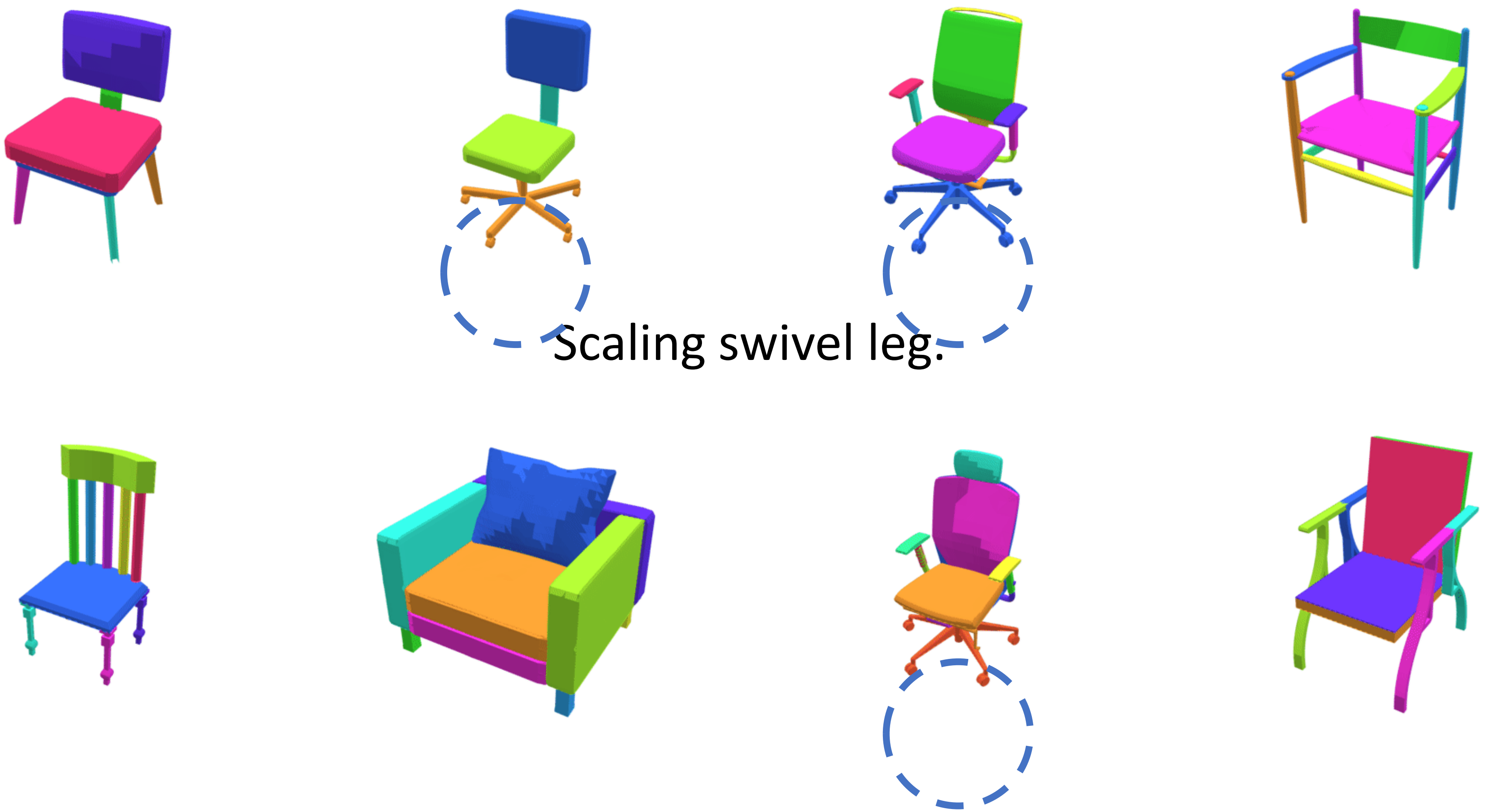


Deformation Dictionary

Scaling **back** along the up/down direction.

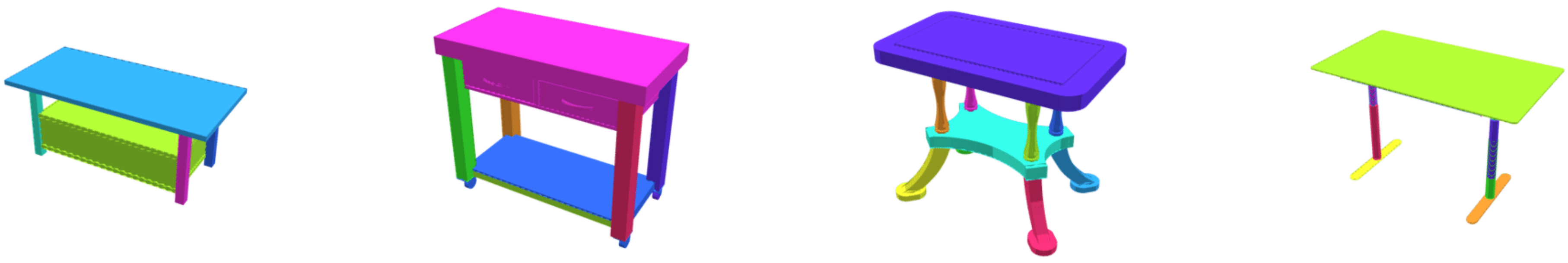


Deformation Dictionary



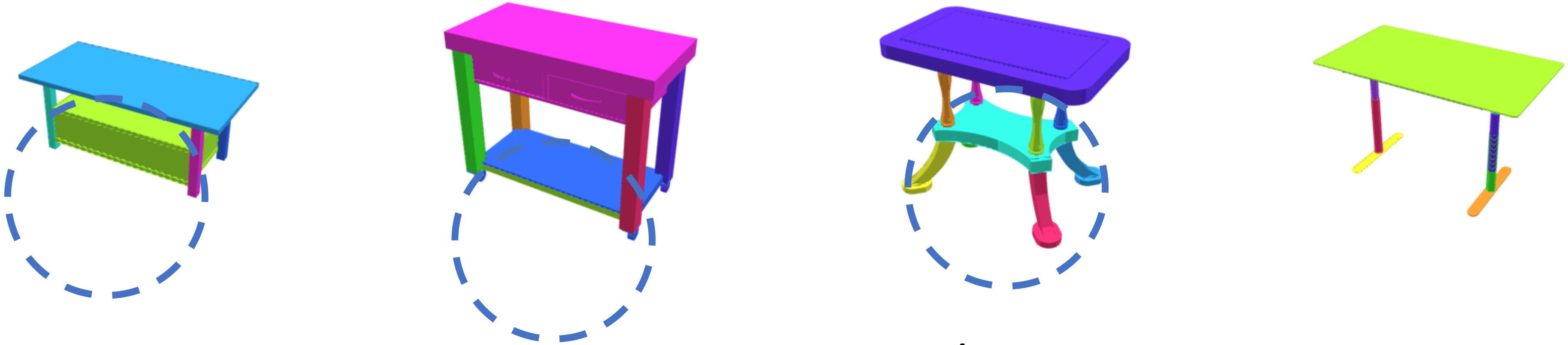
Scaling swivel leg.

Deformation Dictionary

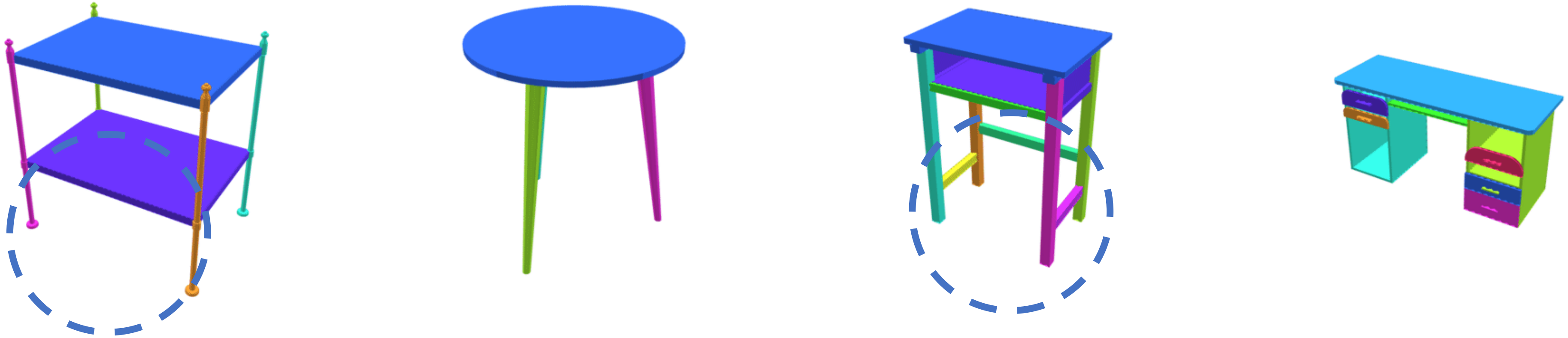


Scaling along the front/back direction.

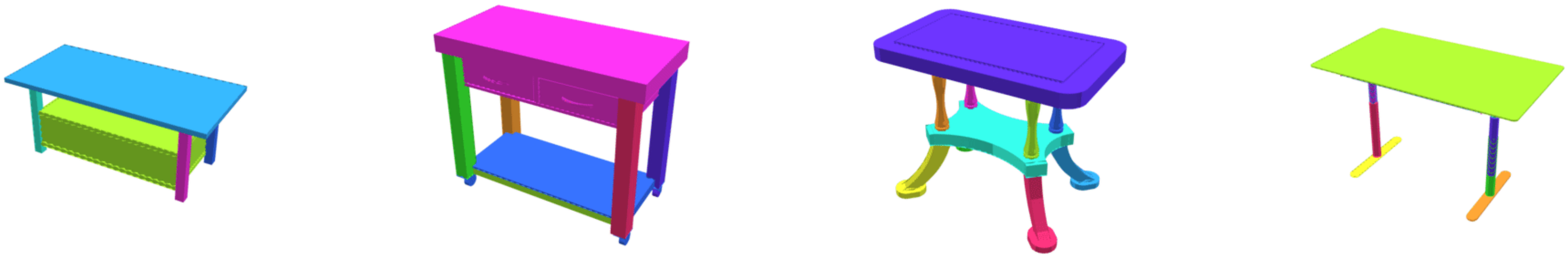
Deformation Dictionary



Translating **shelf** along the up/down direction.



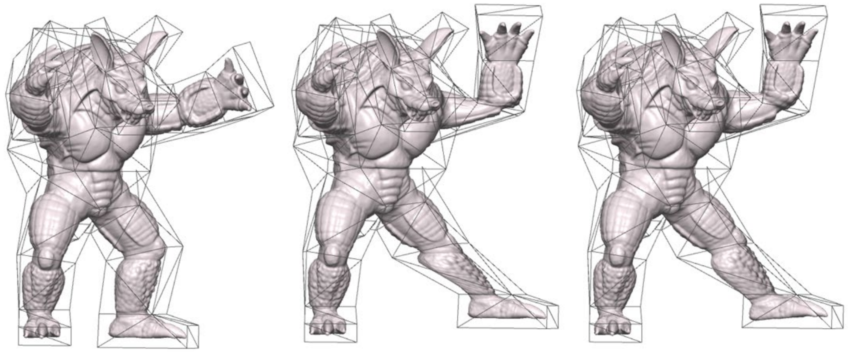
Deformation Dictionary



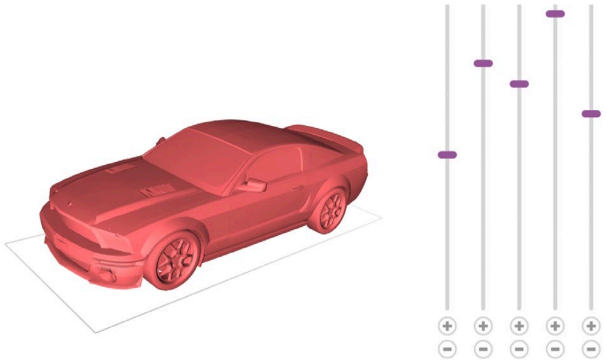
Translating **top** along the up/down direction.

Application

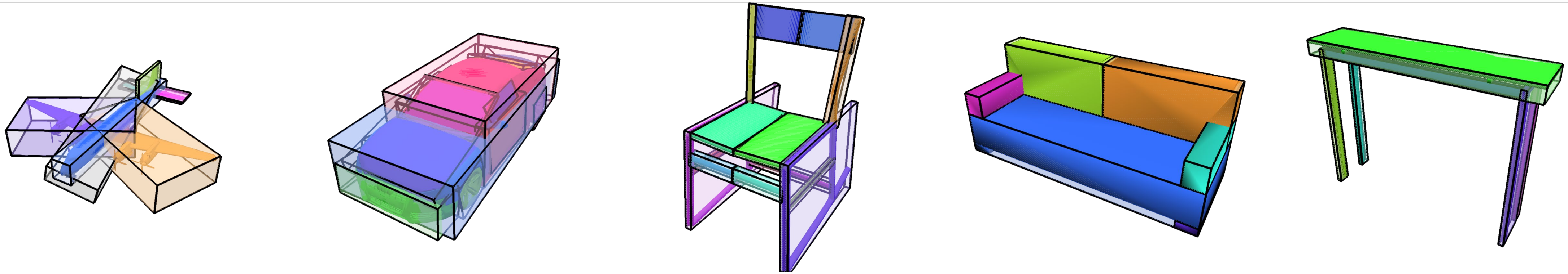
Assume that the given 3D models are equipped with **overparametrized** deformation handles.



Lipman et al., 2008.

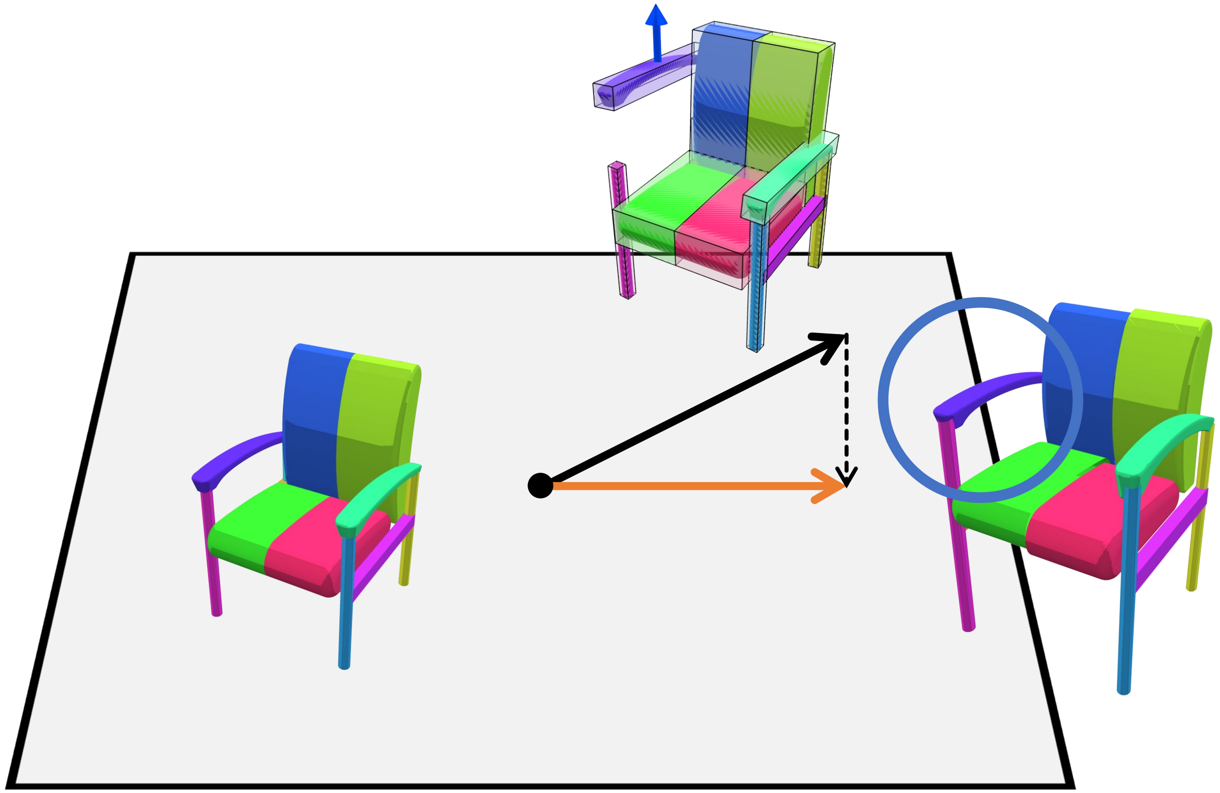


Yumer et al., 2014.



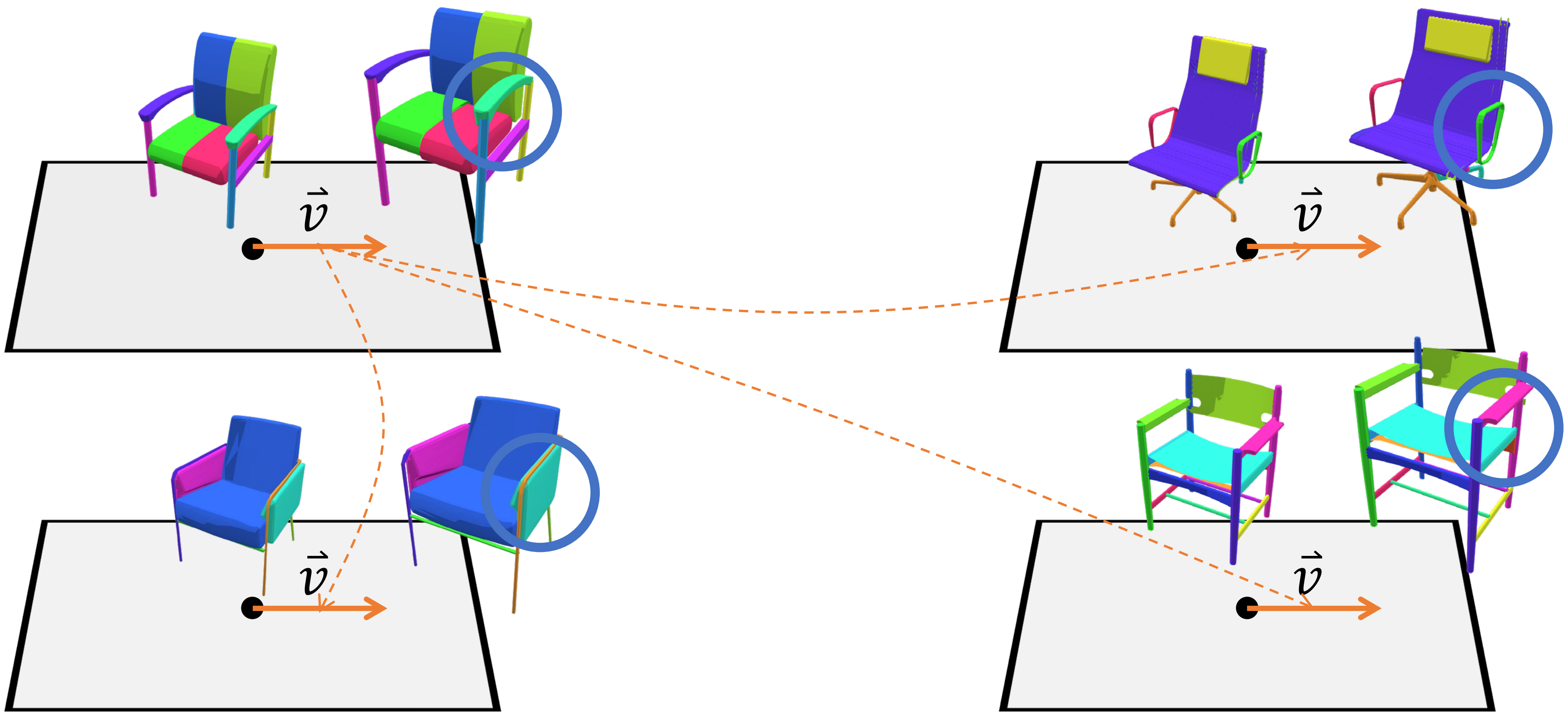
Application – Projection

A user's editing with deformation handles is projected to the latent space.

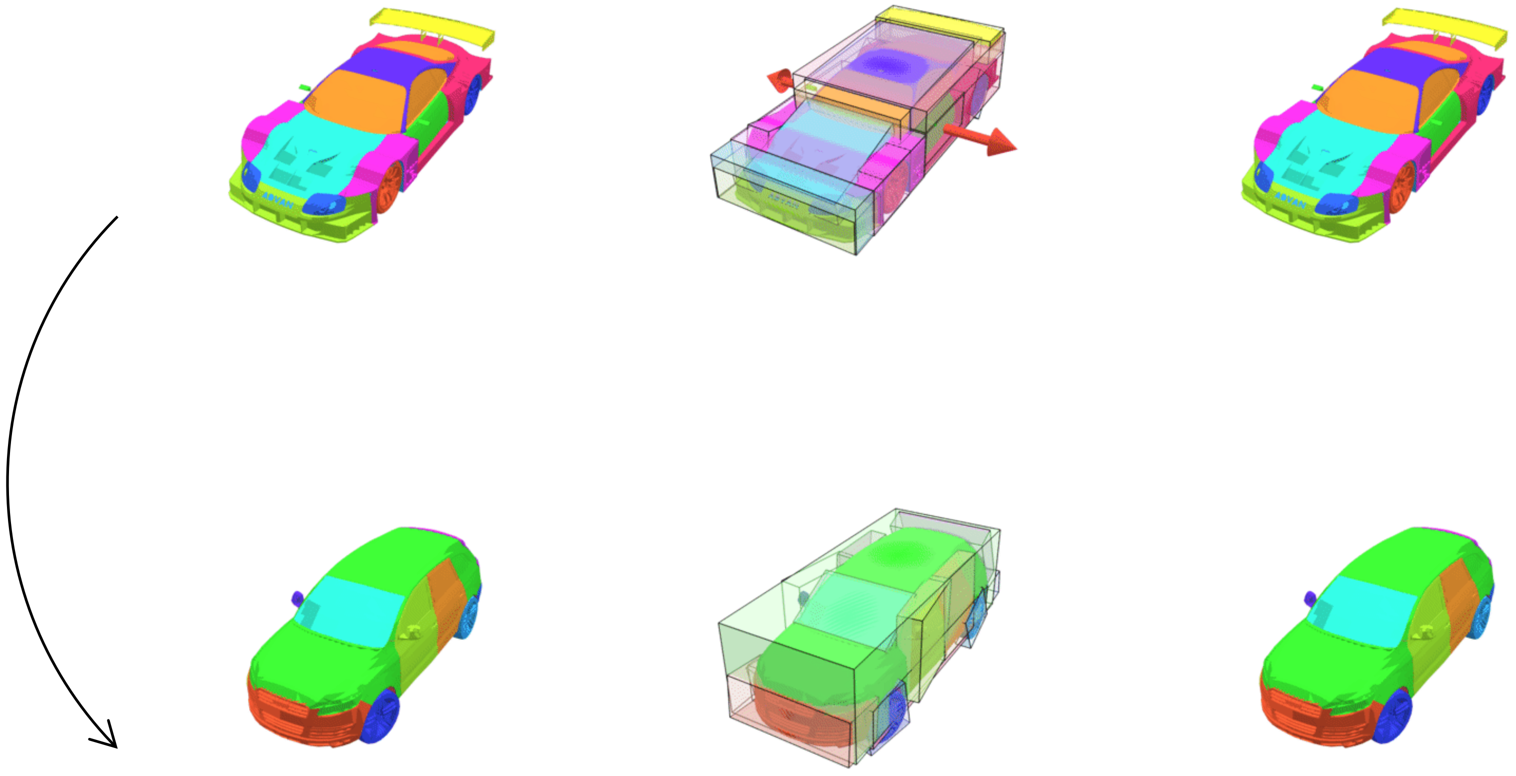


Application – Transfer

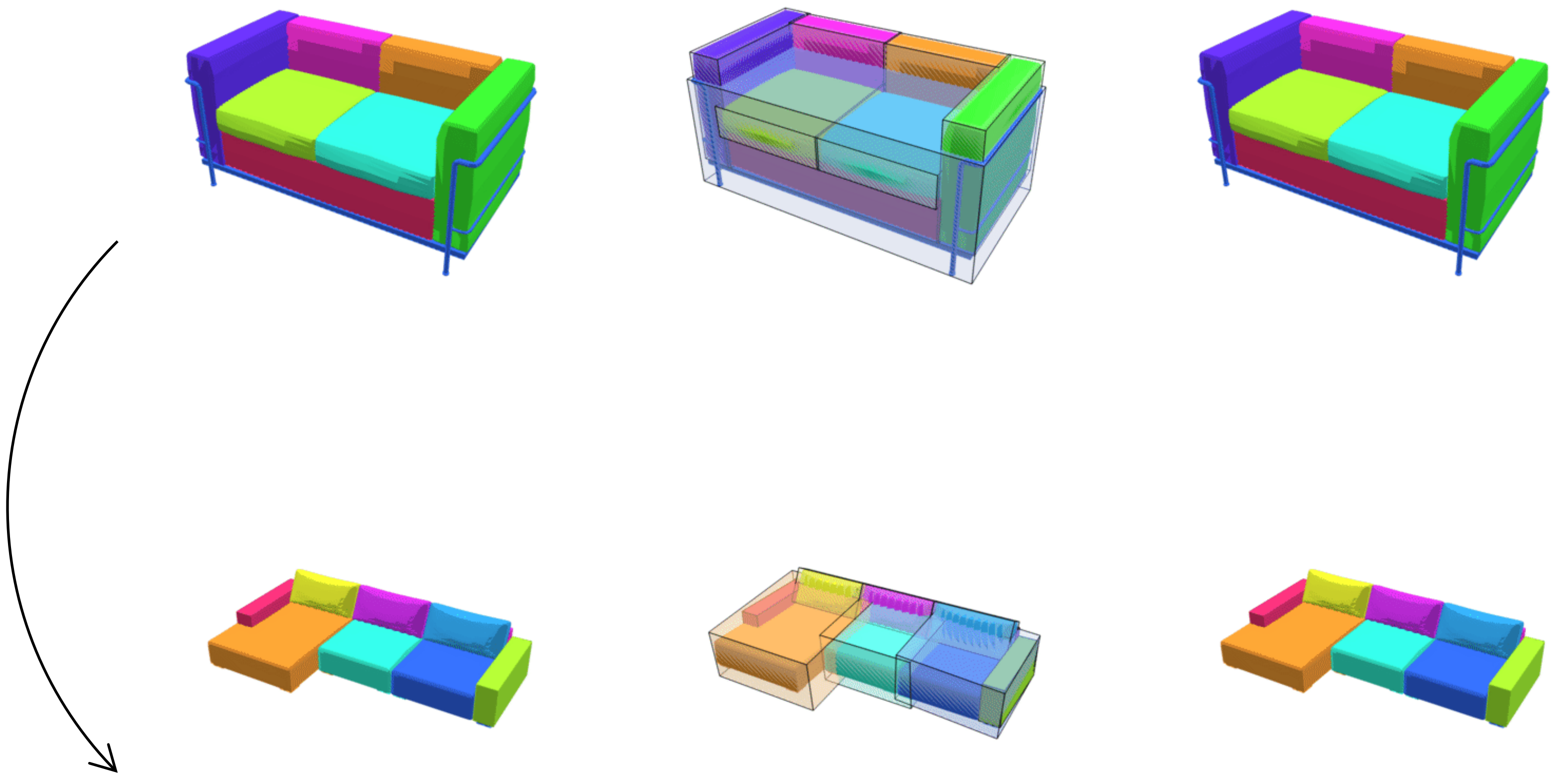
The projected deformation is also transferred to the other shapes without correspondences.



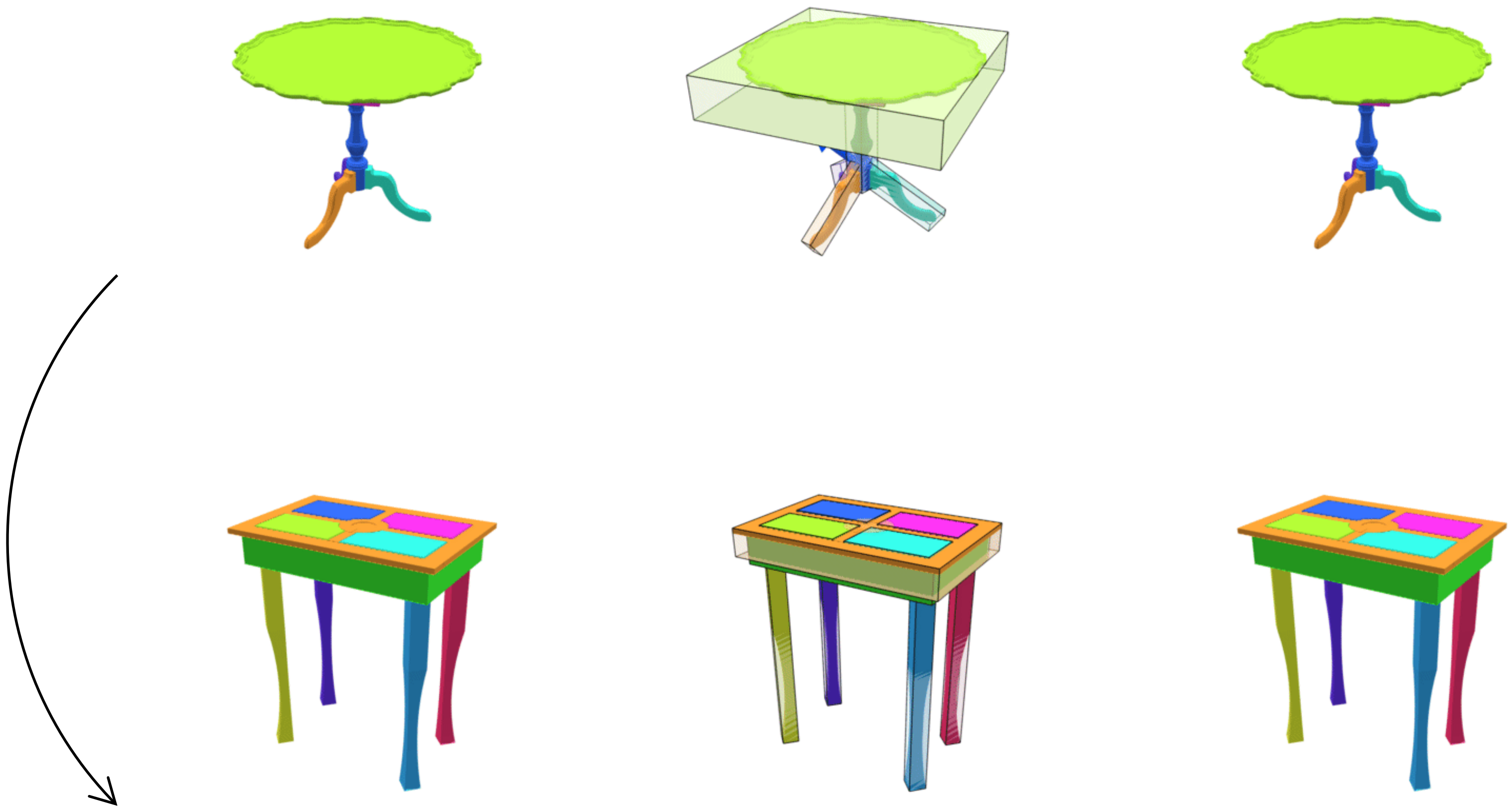
Projection & Transfer



Projection & Transfer



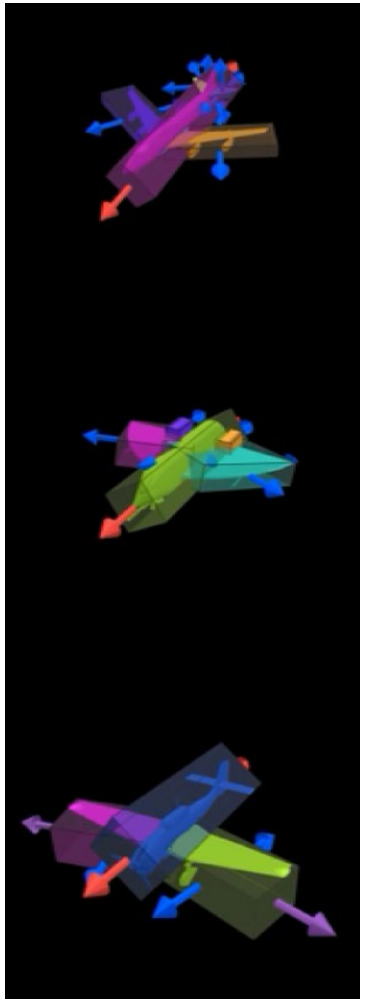
Projection & Transfer



Projection & Transfer

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.



That's All