Neural Parts: Learning Expressive 3D Shape Abstractions with Invertible Neural Networks

Despoina Paschalidou[1,5,6] Angelos Katharopoulos[3,4] Andreas Geiger[1,2,5]
Sanja Fidler[6,7,8]

[1]Max Planck Institute for Intelligent Systems Tübingen [2]University of Tübingen
[3]Idiap Research Institute, Switzerland [4]École Polytechique Fédérale de Lausanne (EPFL)
[5]Max Planck ETH Center for Learning Systems
[6]NVIDIA [7]University of Toronto [8]Vector Institute

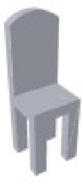
https://paschalidoud.github.io/neural_parts

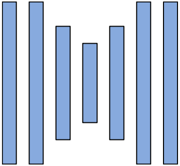
April 30, 2021

Slides are available at

https://paschalidoud.github.io/talks/neural-parts-presentation.pdf

Can we learn to infer 3D from a 2D image?





Input Image | Neural Network | 3D Reconstruction

Taxonomy of 3D Representations

Input Image

Depth

Voxel Grid

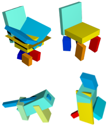
Pointcloud

Mesh

Primitives

Implicit Surface

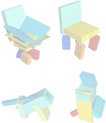
Unsupervised Primitive-based Representations



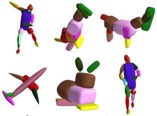
Cuboids

Tulsiani et al. 2017

Unsupervised Primitive-based Representations

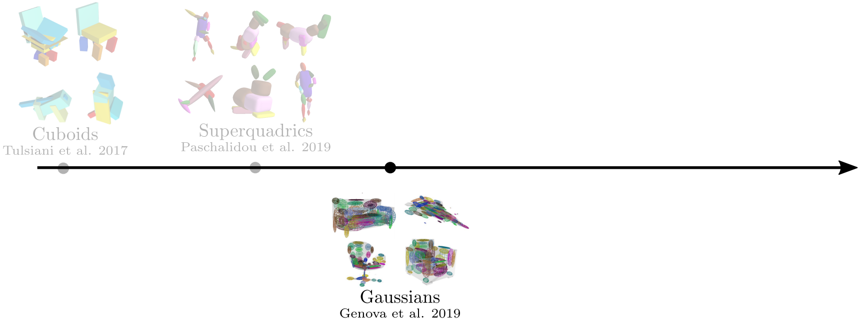


Cuboids
Tulsiani et al. 2017

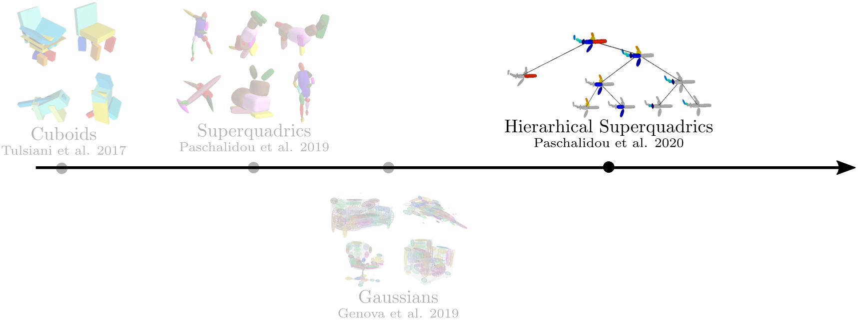


Superquadrics
Paschalidou et al. 2019

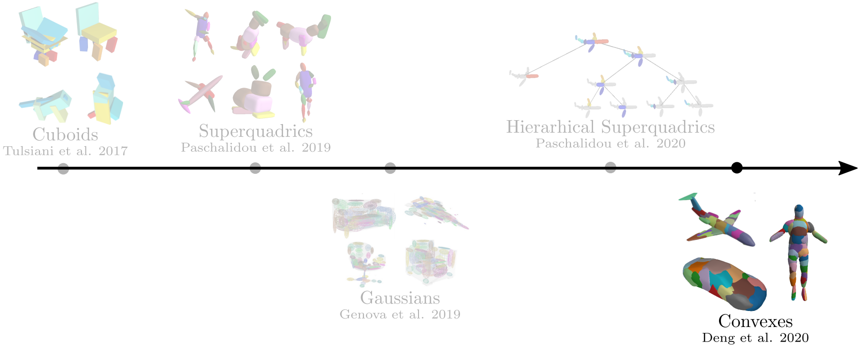
Unsupervised Primitive-based Representations



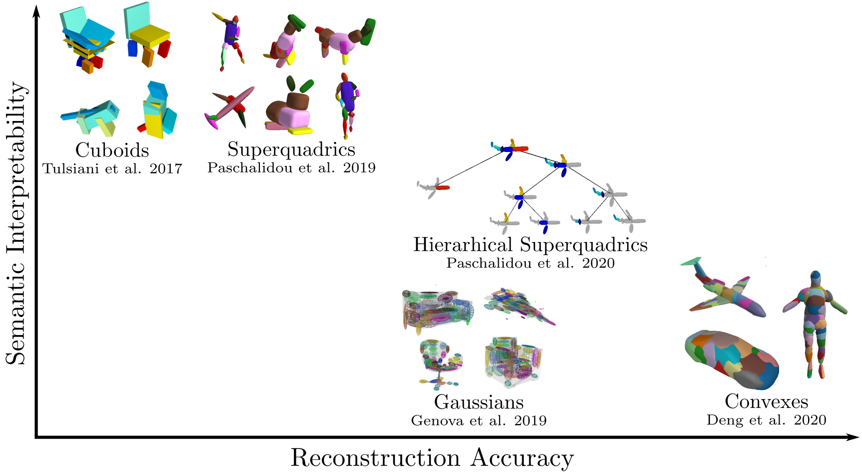
Unsupervised Primitive-based Representations



Unsupervised Primitive-based Representations



Unsupervised Primitive-based Representations

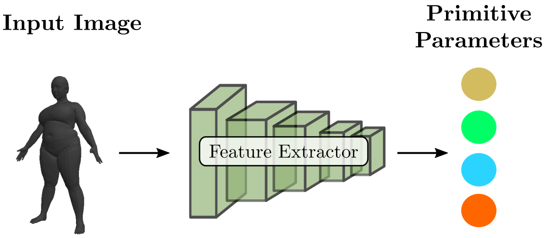


There exists a **trade-off** between the **number of primitives** and the **reconstruction quality** in primitive-based representations.

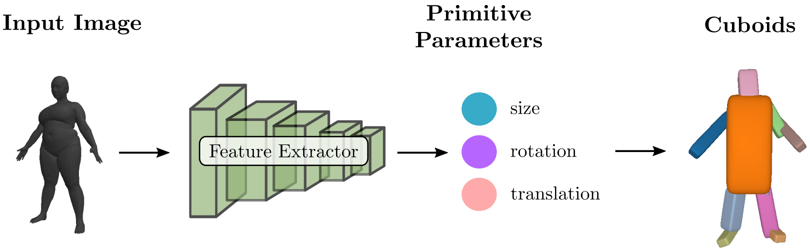
Simple parts require a large number of parts for accurate reconstructions.

Neural Parts yield accurate and semantic reconstructions using an order of magnitude less parts.

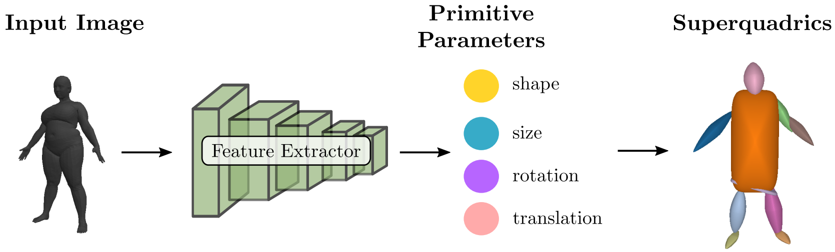
Primitive-based Learning



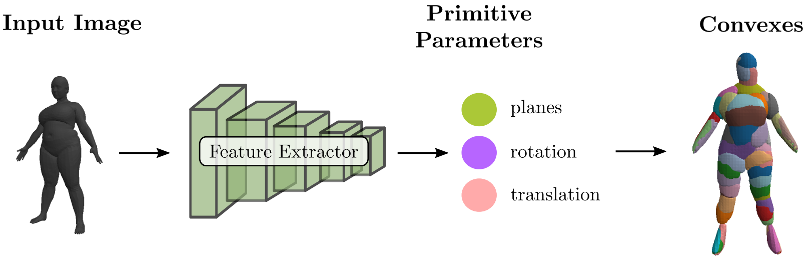
Primitive-based Learning



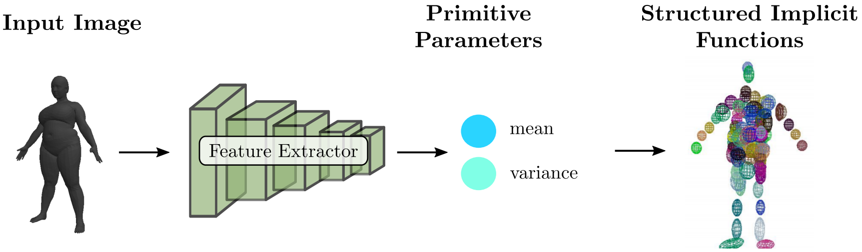
Primitive-based Learning



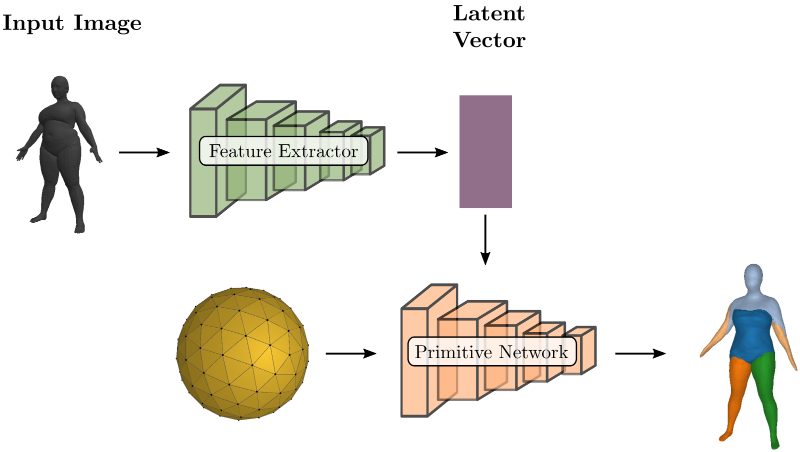
Primitive-based Learning



Primitive-based Learning



Primitive-based Learning



Homeomorphism

A **homeomorphism is a continuous map** between two topological spaces Y and X that preserves all topological properties. In our setup, a homeomorphism $\phi_{\boldsymbol{\theta}} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is

$$\mathbf{x} = \phi_{\boldsymbol{\theta}}(\mathbf{y}) \text{ and } \mathbf{y} = \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$$

where $\mathbf{x}$ and $\mathbf{y}$ are 3D points in X and Y and $\phi_{\boldsymbol{\theta}} : Y \rightarrow X$, $\phi_{\boldsymbol{\theta}}^{-1} : X \rightarrow Y$ are continuous bijections.

Source: Wikipedia

Homeomorphism

A **homeomorphism is a continuous map** between two topological spaces Y and X that preserves all topological properties. In our setup, a homeomorphism $\phi_{\boldsymbol{\theta}} : \mathbb{R}^3 \to \mathbb{R}^3$ is

$$\mathbf{x} = \phi_{\boldsymbol{\theta}}(\mathbf{y}) \text{ and } \mathbf{y} = \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$$

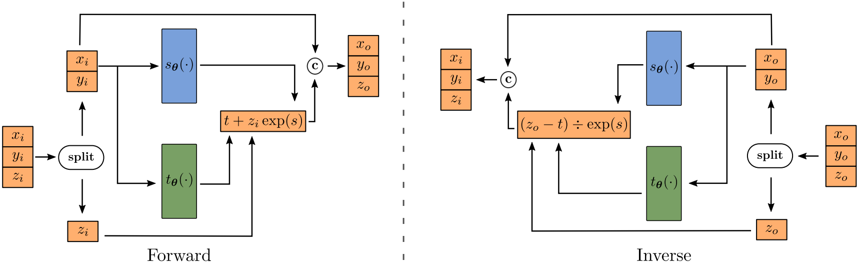
where $\mathbf{x}$ and $\mathbf{y}$ are 3D points in X and Y and $\phi_{\boldsymbol{\theta}} : Y \to X$, $\phi_{\boldsymbol{\theta}}^{-1} : X \to Y$ are continuous bijections.

Source: Wikipedia

For example, geometric object can be seen as topological space, and the **homeomorphism is a continuous stretching and bending of the object into a new shape**.

Parametrizing a Homeomorphism with an INN

A **Real NVP** models a bijective mapping by stacking a sequence of simple bijective transformation functions that **scale** ($s_{\boldsymbol{\theta}} : \mathbb{R}^2 \rightarrow \mathbb{R}$) and **translate** ($t_{\boldsymbol{\theta}} : \mathbb{R}^2 \rightarrow \mathbb{R}$) a set of points from one topological space to another.



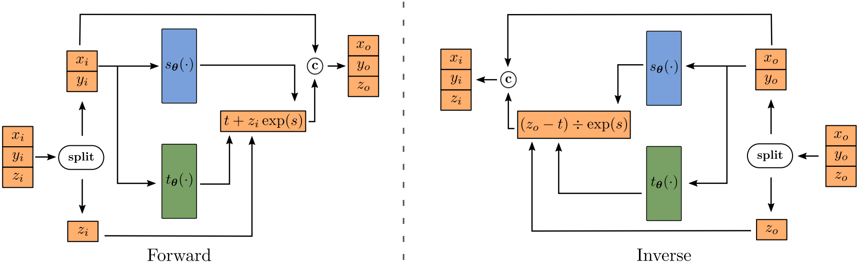
$$x_o = x_i$$

$$y_o = y_i$$

$$z_o = z_i \exp\left(s_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right) + t_{\boldsymbol{\theta}}\left(x_i, y_i\right)$$

Parametrizing a Homeomorphism with an INN

A **Real NVP** models a bijective mapping by stacking a sequence of simple bijective transformation functions that **scale** ($s_{\boldsymbol{\theta}} : \mathbb{R}^2 \rightarrow \mathbb{R}$) and **translate** ($t_{\boldsymbol{\theta}} : \mathbb{R}^2 \rightarrow \mathbb{R}$) a set of points from one topological space to another.



$$x_o = x_i$$

$$y_o = y_i$$

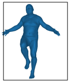
$$z_o = z_i \exp\left(s_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right) + t_{\boldsymbol{\theta}}\left(x_i, y_i\right)$$

The scale $s_{\boldsymbol{\theta}}(\cdot)$ and the translation $t_{\boldsymbol{\theta}}(\cdot)$ functions can be **implemented with arbitrarily complex networks**.

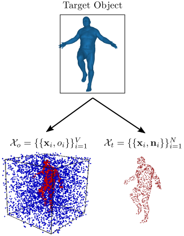
System Overview

Input Image

System Overview

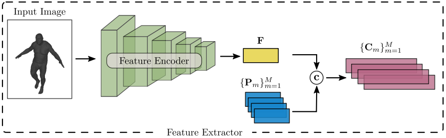


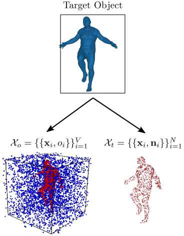
System Overview



- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.

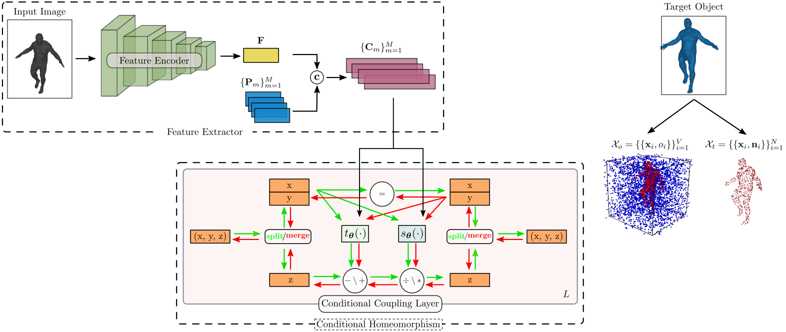
System Overview





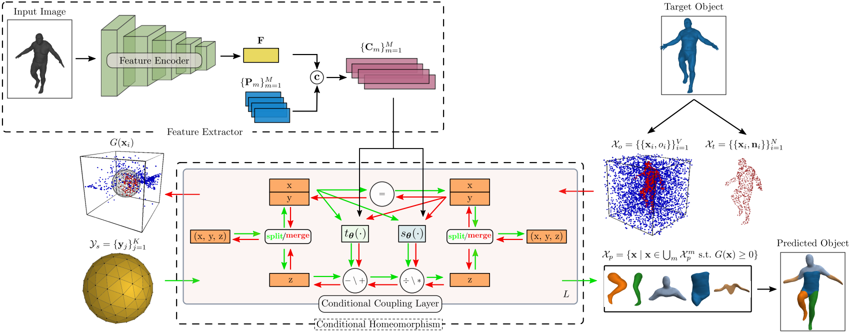
- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.

System Overview



- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.
- The **conditional homeomorphism** deforms a sphere into M primitives and vice-versa.

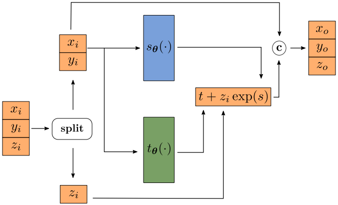
System Overview



- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.
- The **conditional homeomorphism** deforms a sphere into M primitives and vice-versa.

Conditional Homeomorphism

The **original Real NVP cannot be directly applied in our setting** as it does not consider a shape embedding.

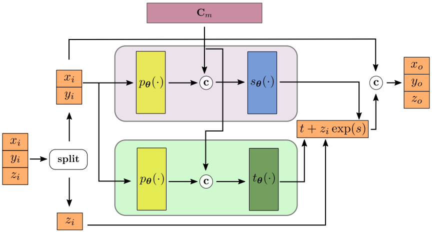


Original Coupling Layer

$$x_o = x_i$$
$$y_o = y_i$$
$$z_o = z_i \exp\left(s_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right) + t_{\boldsymbol{\theta}}\left(x_i, y_i\right)$$

Conditional Homeomorphism

We introduce a **conditional coupling layer** that implements a bijective mapping **conditioned on the per-primitive shape embedding** $\mathbf{C}_m$.



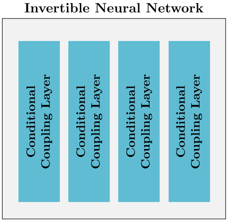
Conditional Coupling Layer

$$x_o = x_i$$

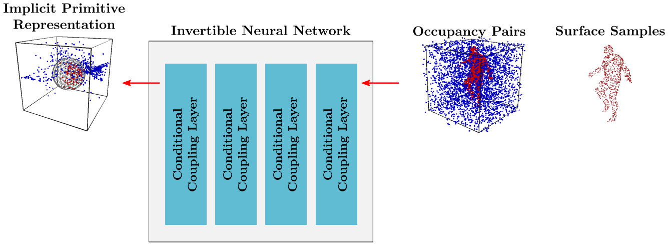
$$y_o = y_i$$

$$z_o = z_i \exp\left(s_{\boldsymbol{\theta}}\left(\left[\mathbf{C}_m; p_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right]\right)\right) + t_{\boldsymbol{\theta}}\left(\left[\mathbf{C}_m; p_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right]\right)$$

What about learning?

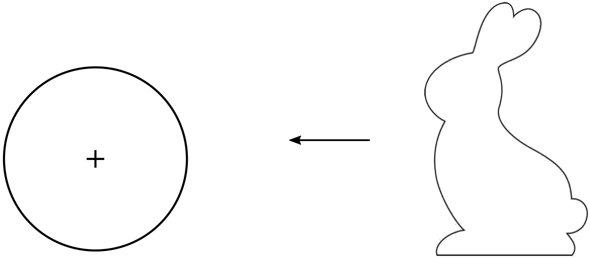


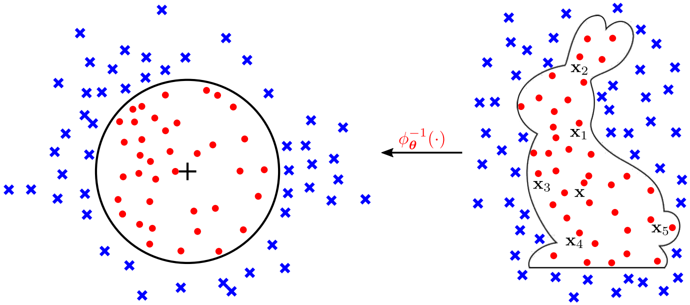
What about learning?



- **Implicit Primitive Representation**:

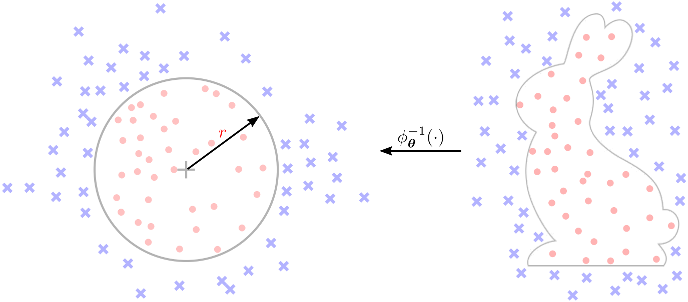
$$g^m(\mathbf{x}) = \|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m)\|_2 - r, \ \forall \ \mathbf{x} \in \mathbb{R}^3$$





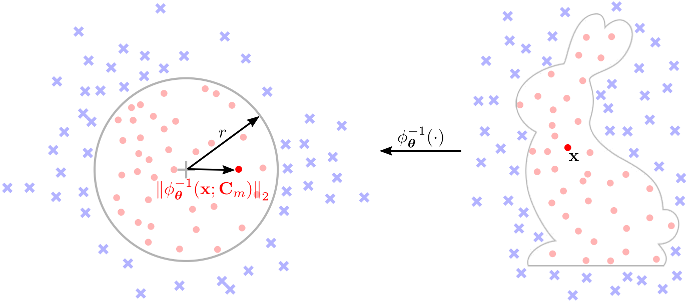
Implicit Primitive Representation:

$$g^m(\mathbf{x}) = \|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m)\|_2 - r, \ \forall \ \mathbf{x} \in \mathbb{R}^3$$



Implicit Primitive Representation:

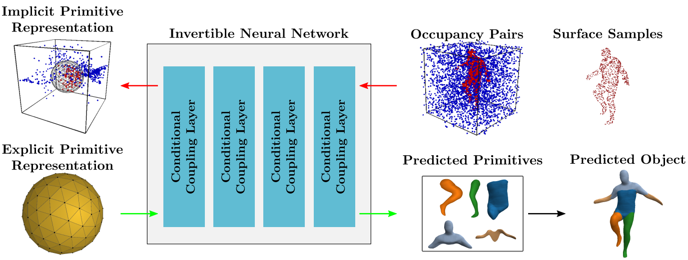
$$g^m(\mathbf{x}) = \|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m)\|_2 - r, \ \forall \ \mathbf{x} \in \mathbb{R}^3$$



Implicit Primitive Representation:

$$g^m(\mathbf{x}) = \|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x};\mathbf{C}_m)\|_2 - r,\ \forall\ \mathbf{x} \in \mathbb{R}^3$$

What about learning?

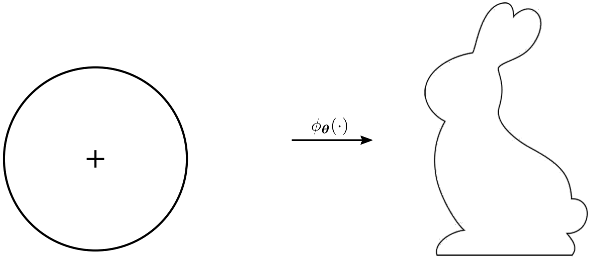


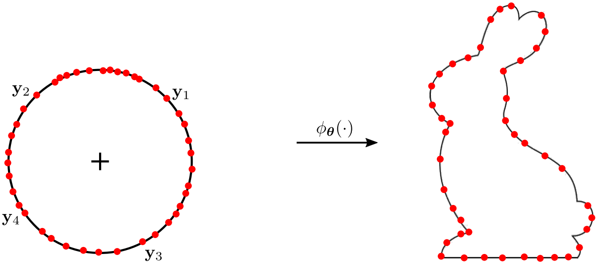
- **Implicit Primitive Representation**:

$$g^m(\mathbf{x}) = \|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m)\|_2 - r, \ \forall \ \mathbf{x} \in \mathbb{R}^3$$

- **Explicit Primitive Representation**:

$$\mathcal{X}_p^m = \{\phi_{\boldsymbol{\theta}}(\mathbf{y}_j; \mathbf{C}_m), \ \forall \ \mathbf{y}_j \in \mathcal{Y}_s\}$$

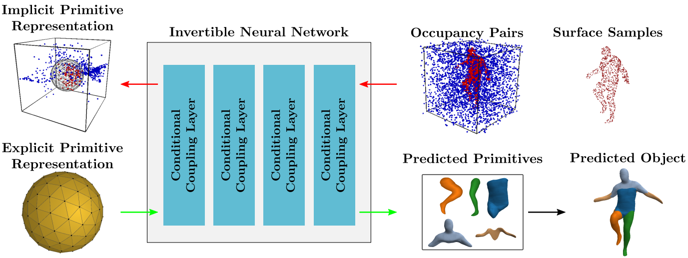

$\phi_{\boldsymbol{\theta}}(\cdot)$



Explicit Primitive Representation:

$$\mathcal{X}_p^m = \{\phi_{\boldsymbol{\theta}}(\mathbf{y}_j; \mathbf{C}_m),\ \forall\ \mathbf{y}_j \in \mathcal{Y}_s\}$$

What about learning?



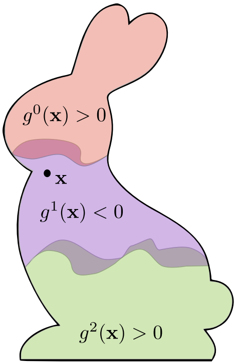
- **Implicit Representation of predicted shape:**

$$G(\mathbf{x}) = \min_{m \in 0 \dots M} g^m(\mathbf{x}),$$

- **Explicit Representation of predicted shape:**

$$\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$$

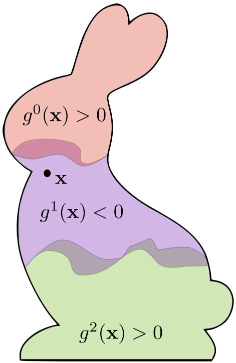
Implicit and Explicit Representation of Predicted Shape

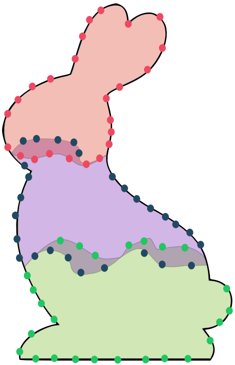


Implicit Representation:

$$G(\mathbf{x}) = \min_{m \in 0 \ldots M} g^m(\mathbf{x}),$$

Implicit and Explicit Representation of Predicted Shape





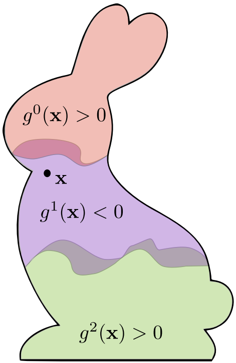
Implicit Representation:

$$G(\mathbf{x}) = \min_{m \in 0 \ldots M} g^m(\mathbf{x}),$$

Explicit Representation:

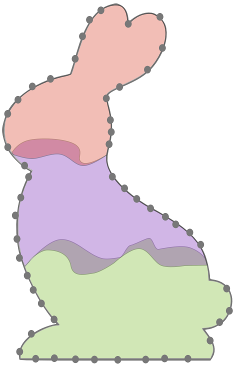
$$\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$$

Implicit and Explicit Representation of Predicted Shape



Implicit Representation:

$$G(\mathbf{x}) = \min_{m \in 0 \ldots M} g^m(\mathbf{x}),$$



Explicit Representation:

$$\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$$

Loss Functions

Overall Loss:

$$\mathcal{L} = \mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) + \mathcal{L}_{occ}(\mathcal{X}_o) + \mathcal{L}_{norm}(\mathcal{X}_t) + \mathcal{L}_{overlap}(\mathcal{X}_o) + \mathcal{L}_{cover}(\mathcal{X}_o)$$

Composed of:

- $\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p)$: Reconstruction Loss
- $\mathcal{L}_{occ}(\mathcal{X}_o)$: Occupancy Loss
- $\mathcal{L}_{norm}(\mathcal{X}_t)$: Normal Consistency Loss
- $\mathcal{L}_{overlap}(\mathcal{X}_o)$: Overlapping Loss
- $\mathcal{L}_{cover}(\mathcal{X}_o)$: Coverage Loss

Loss Functions

Overall Loss:

$\mathcal{L} = \mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) + \mathcal{L}_{occ}(\mathcal{X}_o) + \mathcal{L}_{norm}(\mathcal{X}_t) + \mathcal{L}_{overlap}(\mathcal{X}_o) + \mathcal{L}_{cover}(\mathcal{X}_o)$

Composed of:

- $\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p)$: Reconstruction Loss
- $\mathcal{L}_{occ}(\mathcal{X}_o)$: Occupancy Loss
- $\mathcal{L}_{norm}(\mathcal{X}_t)$: Normal Consistency Loss
- $\mathcal{L}_{overlap}(\mathcal{X}_o)$: Overlapping Loss
- $\mathcal{L}_{cover}(\mathcal{X}_o)$: Coverage Loss

Target and Predicted Shape:

- **Target:**
  - **Surface Samples:** $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$

Loss Functions

Overall Loss:

$$\mathcal{L} = \mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) + \mathcal{L}_{occ}(\mathcal{X}_o) + \mathcal{L}_{norm}(\mathcal{X}_t) + \mathcal{L}_{overlap}(\mathcal{X}_o) + \mathcal{L}_{cover}(\mathcal{X}_o)$$

Composed of:

- $\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p)$: Reconstruction Loss
- $\mathcal{L}_{occ}(\mathcal{X}_o)$: Occupancy Loss
- $\mathcal{L}_{norm}(\mathcal{X}_t)$: Normal Consistency Loss
- $\mathcal{L}_{overlap}(\mathcal{X}_o)$: Overlapping Loss
- $\mathcal{L}_{cover}(\mathcal{X}_o)$: Coverage Loss

Target and Predicted Shape:

- **Target:**
  - **Surface Samples:** $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$
  - **Volumetric Samples:** $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Loss Functions

Overall Loss:

$$\mathcal{L} = \mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) + \mathcal{L}_{occ}(\mathcal{X}_o) + \mathcal{L}_{norm}(\mathcal{X}_t) + \mathcal{L}_{overlap}(\mathcal{X}_o) + \mathcal{L}_{cover}(\mathcal{X}_o)$$

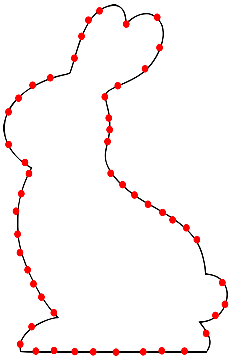
Composed of:

- $\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p)$: Reconstruction Loss
- $\mathcal{L}_{occ}(\mathcal{X}_o)$: Occupancy Loss
- $\mathcal{L}_{norm}(\mathcal{X}_t)$: Normal Consistency Loss
- $\mathcal{L}_{overlap}(\mathcal{X}_o)$: Overlapping Loss
- $\mathcal{L}_{cover}(\mathcal{X}_o)$: Coverage Loss

Target and Predicted Shape:

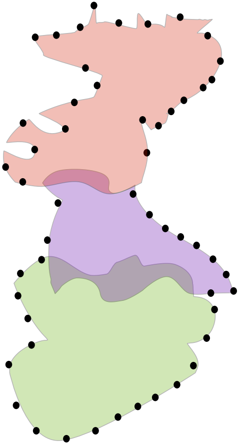
- **Target:**
  - **Surface Samples:** $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$
  - **Volumetric Samples:** $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$
- **Predicted:** $\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$

Reconstruction Loss



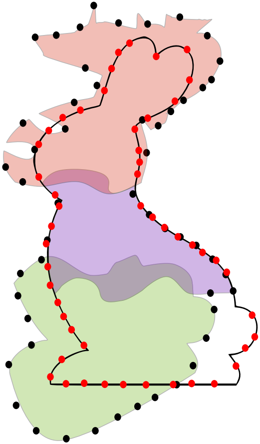
Target Surface Samples: $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^N$

Reconstruction Loss



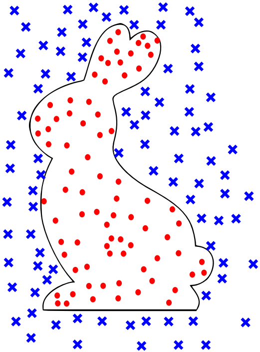
Predicted Surface Samples: $\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$

Reconstruction Loss



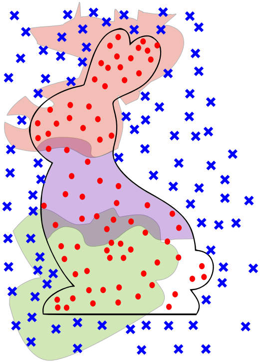
$$\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) = \frac{1}{|\mathcal{X}_t|} \sum_{\mathbf{x}_i \in \mathcal{X}_t} \min_{\mathbf{x}_j \in \mathcal{X}_p} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_j \in \mathcal{X}_p} \min_{\mathbf{x}_i \in \mathcal{X}_t} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$$

Occupancy Loss

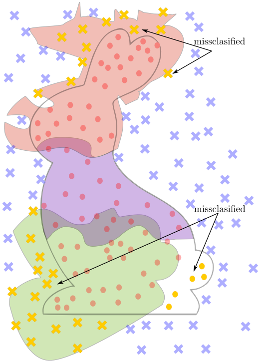


Target Volumetric Samples: $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Occupancy Loss

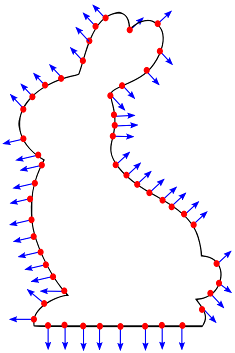


Occupancy Loss



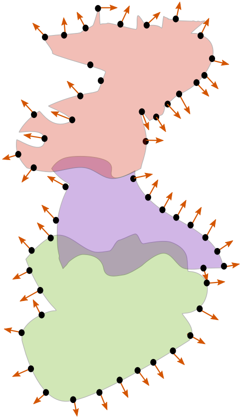
$$\mathcal{L}_{occ}(\mathcal{X}_o) = \sum_{(\mathbf{x},o)\in\mathcal{X}_o} \mathcal{L}_{ce}\left(\underbrace{\sigma\left(\frac{-G(\mathbf{x})}{\tau}\right)}_{\substack{>1 \text{ when } \mathbf{x} \\ \text{inside the predicted shape}}}, o \right)$$

Normal Consistency Loss



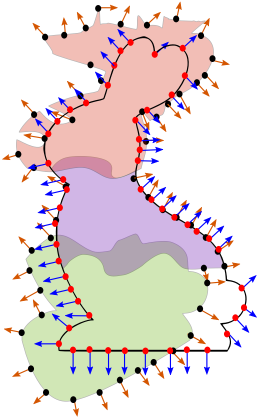
Target Surface Samples: $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$

Normal Consistency Loss



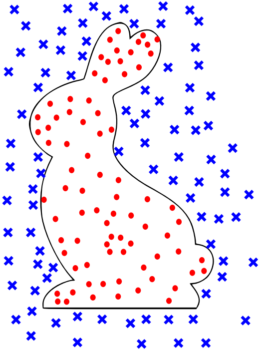
Predicted Surface Normals: $\frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}$

Normal Consistency Loss



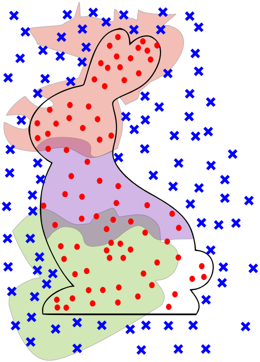
$$\mathcal{L}_{norm}(\mathcal{X}_t) = \frac{1}{|\mathcal{X}_t|} \sum_{(\mathbf{x},\mathbf{n}) \in \mathcal{X}_t} \left(1 - \left\langle \frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}, \mathbf{n} \right\rangle\right)$$

Overlapping Loss

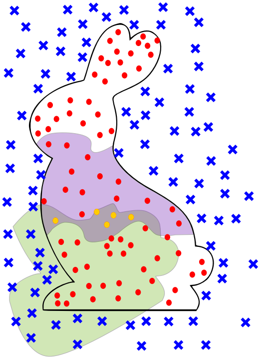


Target Volumetric Samples: $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Overlapping Loss

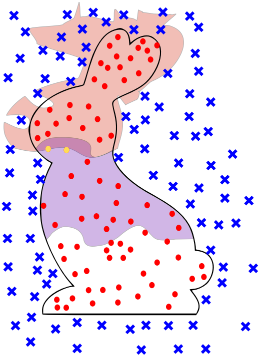


Overlapping Loss



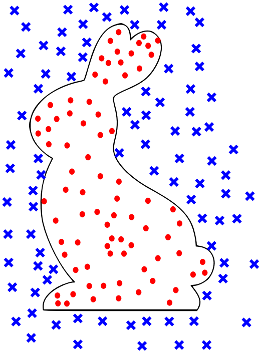
$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-\mathbf{g}^m(\mathbf{x})}{\tau}\right) - \lambda\right)$$

Overlapping Loss



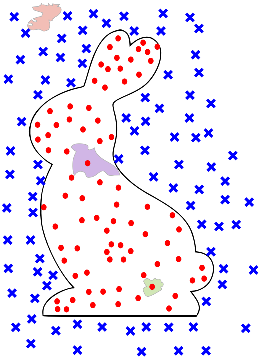
$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-\mathbf{g}^m(\mathbf{x})}{\tau}\right) - \lambda\right)$$

Coverage Loss



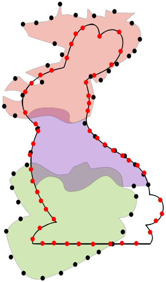
Target Volumetric Samples: $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Coverage Loss



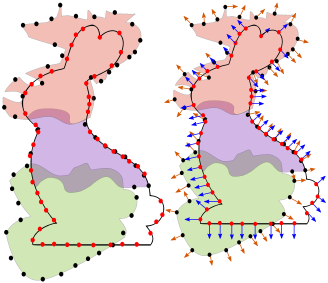
$$\mathcal{L}_{cover}(\mathcal{X}_o) = \sum_{m=1}^{M} \sum_{\mathbf{x} \in \mathcal{N}_k^m} \max\left(0, g^m(\mathbf{x})\right)$$

Loss Functions: Summary



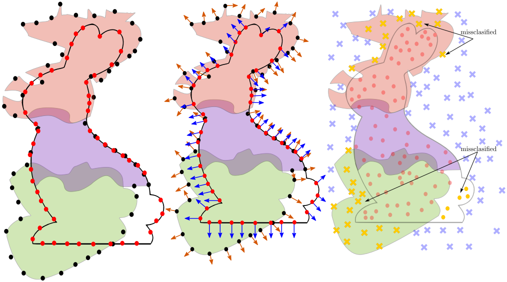
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.

Loss Functions: Summary



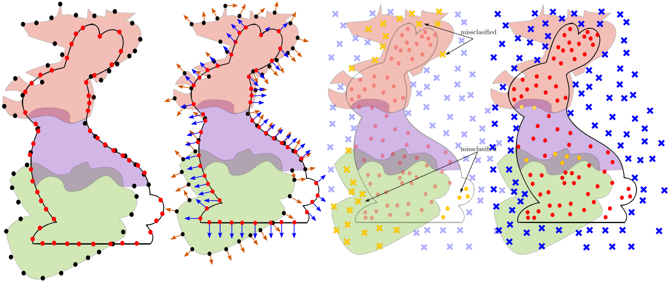
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.

Loss Functions: Summary



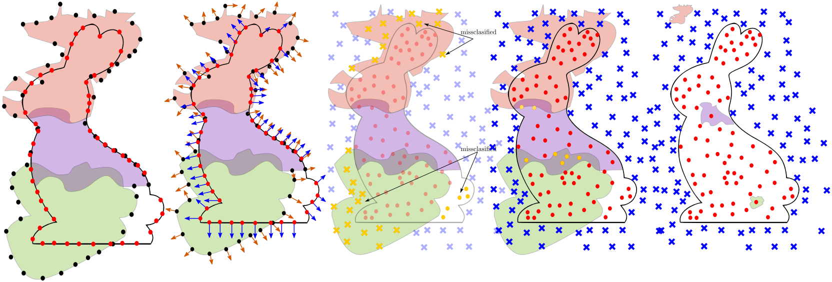
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.

Loss Functions: Summary



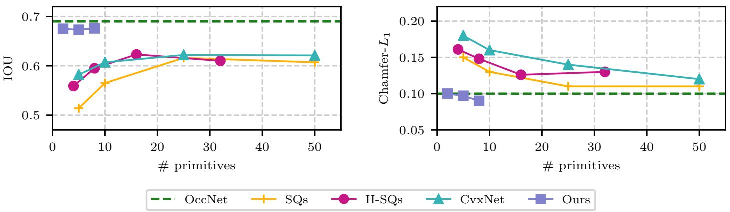
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.
- **Overlapping Loss**: Prevent overlapping primitives.

Loss Functions: Summary

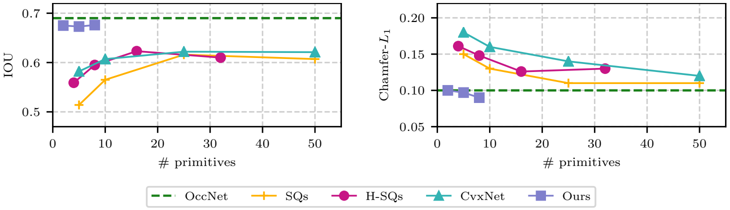


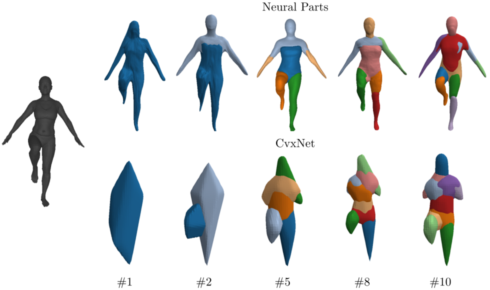
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.
- **Overlapping Loss**: Prevent overlapping primitives.
- **Coverage Loss**: Prevent degenerate primitive arrangements.

Representation Power of Primitive-based Representations



Representation Power of Primitive-based Representations





Single-view 3D Reconstruction on D-FAUST

Single-view 3D Reconstruction on FreiHAND

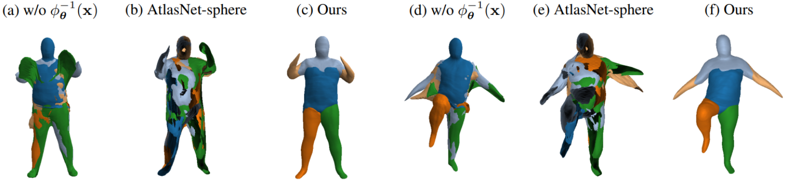
Single-view 3D Reconstruction on ShapeNet

Single-view 3D Reconstruction on various animals

Semantic Consistency

Do we really need an INN?

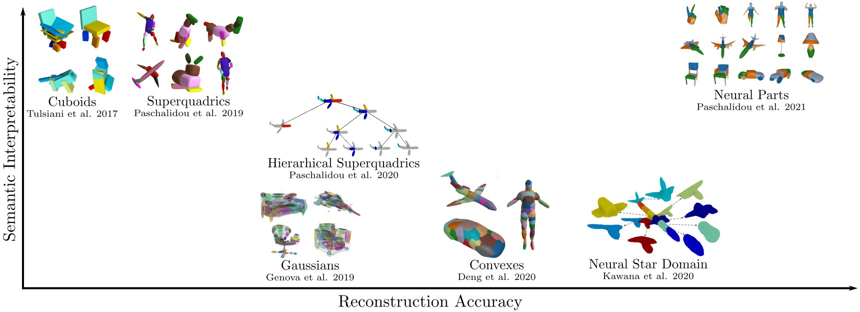
Do we really need an INN?



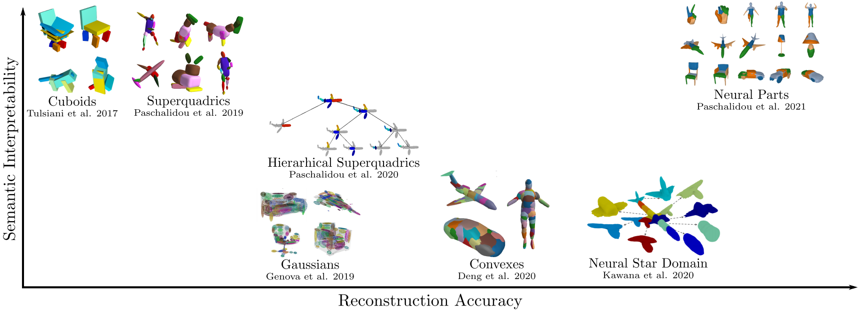
	w/o $\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$	AtlasNet - sphere	Ours
IoU	0.639	*	0.673
Chamfer-L_1	0.119	0.087	0.097

What comes next?

Primitive Arena

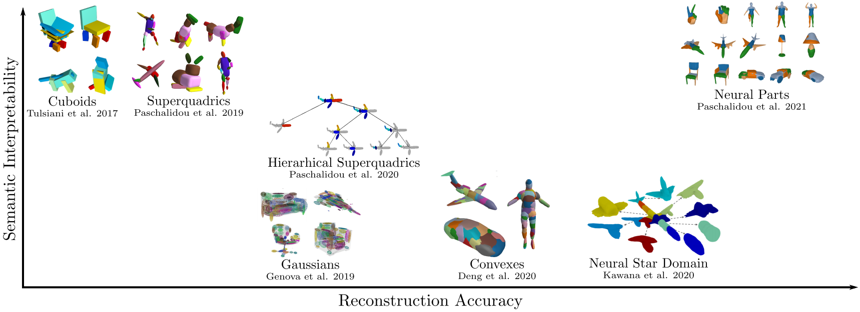


Primitive Arena



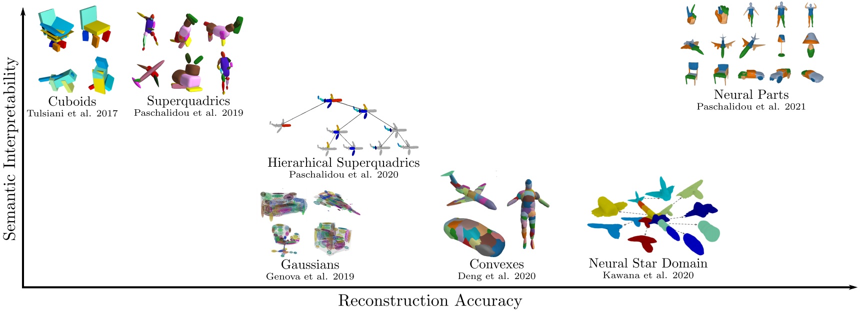
- What makes a **good primitive-representation**?

Primitive Arena



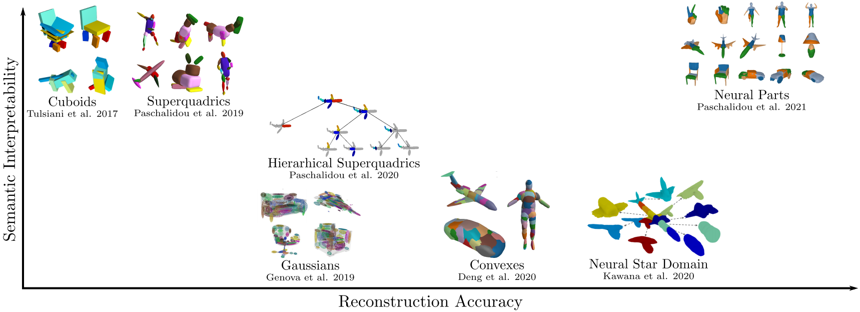
- What makes a **good primitive-representation**?
- We learn primitives by **optimizing the geometry**? Can't we do better?

Primitive Arena



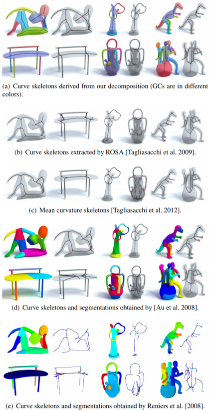
- What makes a **good primitive-representation**?
- We learn primitives by **optimizing the geometry**? Can't we do better?
- **Do we really learn semantic parts**?

Primitive Arena



- What makes a **good primitive-representation**?
- We learn primitives by **optimizing the geometry**? Can't we do better?
- **Do we really learn semantic parts**?
- Why do we need primitive-based representations?

Learning semantic parts without part-level supervision



(a) Curve skeletons derived from our decomposition (GCs are in different colors).

(b) Curve skeletons extracted by ROSA [Tagliasacchi et al. 2009].

(c) Mean curvature skeletons [Tagliasacchi et al. 2012].

(d) Curve skeletons and segmentations obtained by [Au et al. 2008].

(e) Curve skeletons and segmentations obtained by Reniers et al. [2008].

Image Source: Generalized Cylinder Decomposition, 2015

Learning parts through skeletonization

Learning semantic parts without part-level supervision

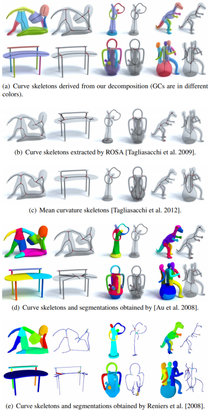


Image Source: Generalized Cylinder Decomposition, 2015

Learning parts through skeletonization

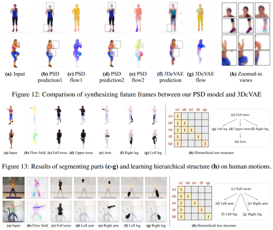


Image Source: Unsupervised Discovery of Parts, Structure and Dynamics, 2019

Learning parts from other cues (e.g. motion)

Learning semantic parts without part-level supervision

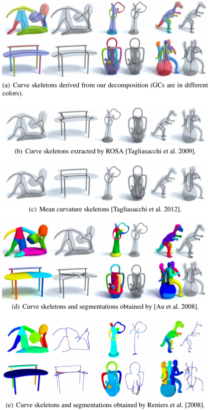


Image Source: Generalized Cylinder Decomposition, 2015

Learning parts through skeletonization

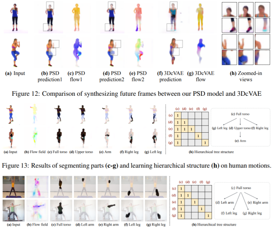


Image Source: Unsupervised Discovery of Parts, Structure and Dynamics, 2019

Learning parts from other cues (e.g. motion)

Image Source: Functionality Representations and Applications for Shape Analysis, 2018

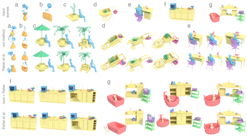


Image Source: Relationship Templates for Creating Scene Variations, 2016

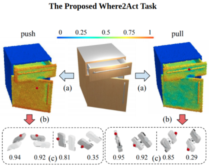


Image Source: Where2Act: From Pixels to Actions for Articulated 3D Objects, 2021

Learning functional parts

Neural Rooms: Learning to synthesize virtual rooms

Goal of this work:

- Develop an easy to use tool for synthesizing rooms
- Synthesize or complete rooms
- Re-arrange rooms
- Suggest alternative attributes

Generative model of parts for content creation

Image Source: Google Chimera

Generative model of parts for content creation

Image Source: AttribIt: Content Creation with Semantic Attributes, 2013

Reconstruction by combining parts

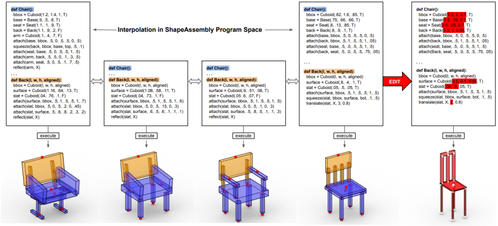


Image Source: ShapeAssembly: Learning to Generate Programs for 3D Shape Structure Synthesis, 2020

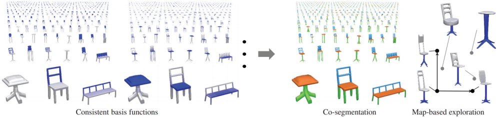


Image Source:Functional Map Networks for Analyzing and Exploring Large Shape Collections, 2013

Thank you for your attention!