Learning Interpretable Representations for Understanding and Generating 3D Environments

Despoina Paschalidou

Autonomous Vision Group, Max Planck Institute for Intelligent Systems
Tübingen
Computer Vision Lab, ETH Zürich

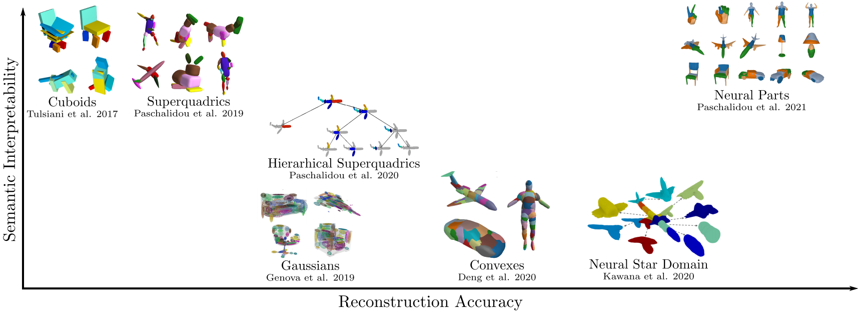
Slides are available at

https://paschalidoud.github.io/talks/learning-interpretable-representations.pdf

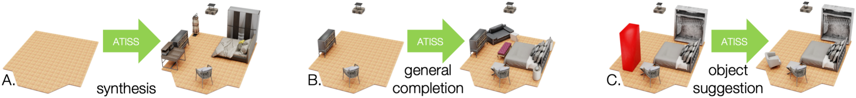

sawyer


ΚΑΦΕ ΟΥΖΕΡΙ
Τσινάρι

To achieve true AI we need to develop systems that can robustly **reason about the world both in object level and in scene level.**

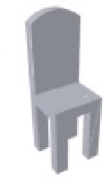


Object-level 3D understanding.

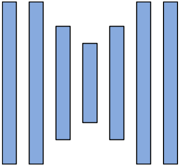


Scene-level Understanding and Generation.

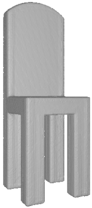
Can we learn to recover 3D geometry from a 2D image?



Input Image



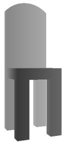
Neural Network



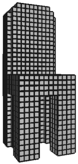
3D Reconstruction

Taxonomy of 3D Representations

Input Image

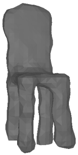


Depth

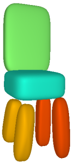


Voxel Grid

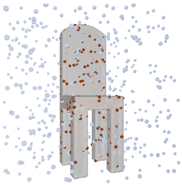
Pointcloud



Mesh

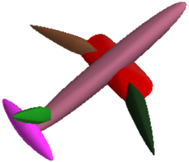


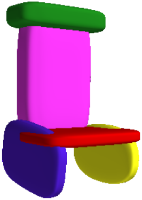
Primitives



Implicit Surface

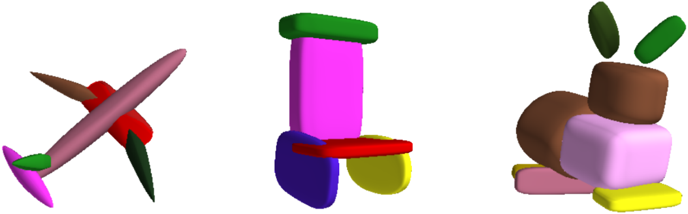
3D Geometric Primitives: Why do we care?





Primitive-based Representations:

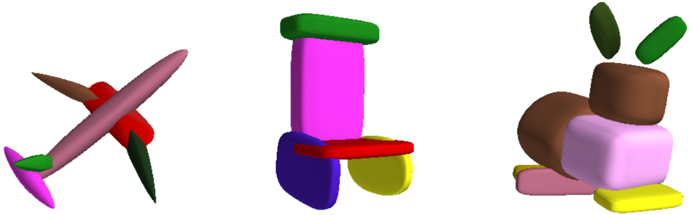
3D Geometric Primitives: Why do we care?



Primitive-based Representations:

- **Parsimonious Description**: Capture the 3D geometry using a small number of primitives.

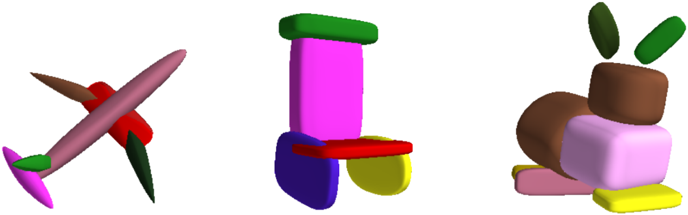
3D Geometric Primitives: Why do we care?



Primitive-based Representations:

- **Parsimonious Description**: Capture the 3D geometry using a small number of primitives.
- Convey semantic information (parts, functionality, etc.)

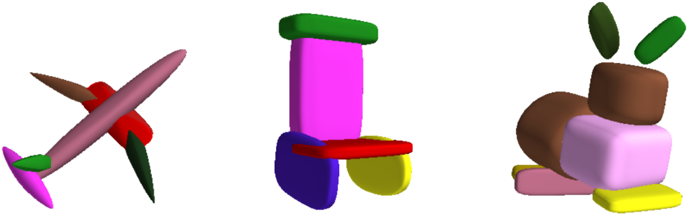
3D Geometric Primitives: Why do we care?



Primitive-based Representations:

- **Parsimonious Description**: Capture the 3D geometry using a small number of primitives.
- Convey semantic information (parts, functionality, etc.)
- **Main Challenges**:

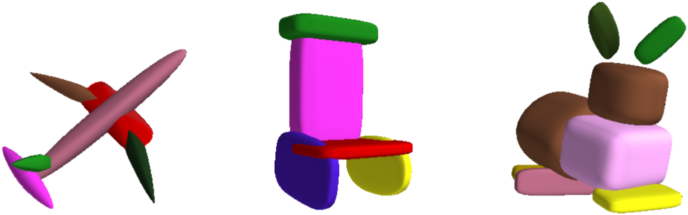
3D Geometric Primitives: Why do we care?



Primitive-based Representations:

- **Parsimonious Description**: Capture the 3D geometry using a small number of primitives.
- Convey semantic information (parts, functionality, etc.)
- **Main Challenges**:
  - Very few annotated datasets

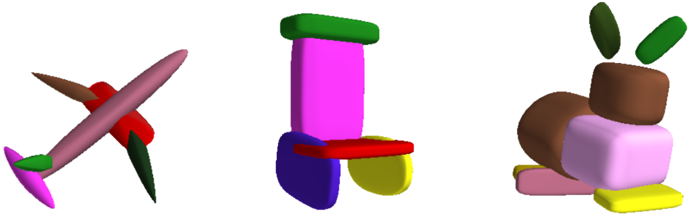
3D Geometric Primitives: Why do we care?



Primitive-based Representations:

- **Parsimonious Description**: Capture the 3D geometry using a small number of primitives.
- Convey semantic information (parts, functionality, etc.)
- **Main Challenges**:
  - Very few annotated datasets
  - Variable number of parts

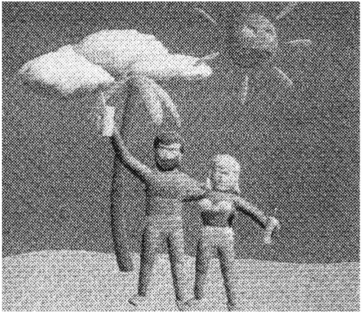
3D Geometric Primitives: Why do we care?

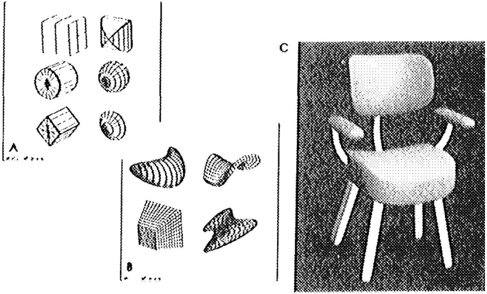


Primitive-based Representations:

- **Parsimonious Description**: Capture the 3D geometry using a small number of primitives.
- Convey semantic information (parts, functionality, etc.)
- **Main Challenges**:
  - Very few annotated datasets
  - Variable number of parts
  - What is really a semantic part?

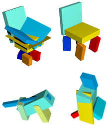
1986: Pentland's Superquadrics





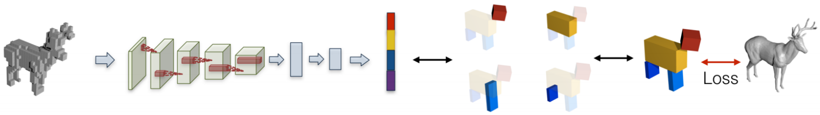
- 1 superquadric can be represented with 11 parameters
- Scene on the left **contructed with 100 primitives** required less than 1000 bytes!
- Early fitting-based approaches did not work robustly

Unsupervised Primitive-based Representations



Cuboids
Tulsiani et al. 2017

2017: 3D Reconstructions with Volumetric Primitives

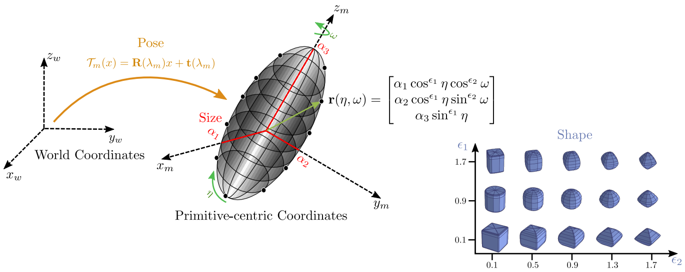


- **Unsupervised** method for learning **cuboidal primitives**
- **Variable number of primitives**
- While **cuboids are sufficient for capturing the structure** of an object they **do not lead to expressive abstractions**.
- Computational expensive reinforcement learning for learning the existence probabilities

Unsupervised Primitive-based Representations

2019: Superquadric Surfaces as Primitives

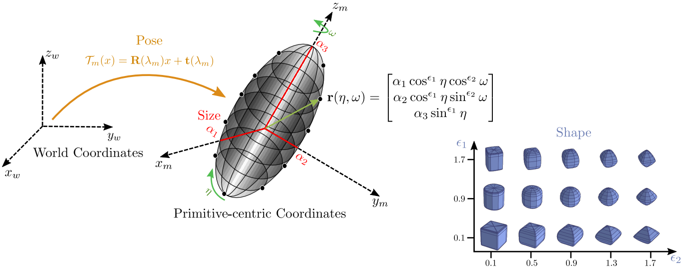
*Everything in nature takes its form from the **sphere**, the **cone** and the **cylinder**.* - Paul Cezanne.



*Superquadrics **allow complex solids and surfaces to be constructed and altered easily from a few interactive parameters.***

2019: Superquadric Surfaces as Primitives

*Everything in nature takes its form from the **sphere**, the **cone** and the **cylinder**.* - Paul Cezanne.

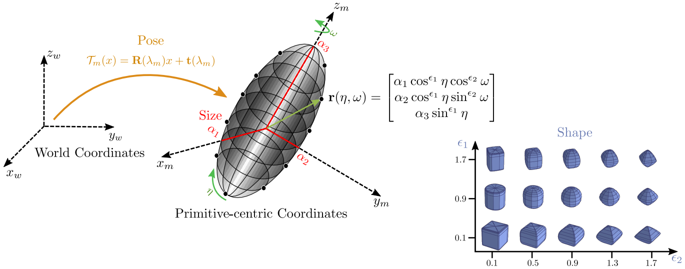


*Superquadrics **allow complex solids and surfaces to be constructed and altered easily from a few interactive parameters.***

- Fully described with just 11 parameters

2019: Superquadric Surfaces as Primitives

*Everything in nature takes its form from the **sphere**, the **cone** and the **cylinder**.* - Paul Cezanne.

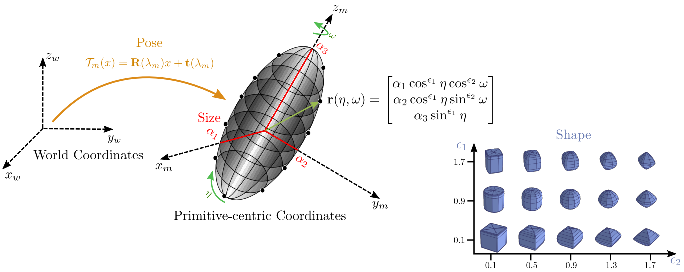


Superquadrics ***allow complex solids and surfaces to be constructed and altered easily from a few interactive parameters.***

- Fully described with just 11 parameters
- Represent a diverse class of shapes such as cylinders, spheres, cuboids, ellipsoids in a **single continuous parameter space**

2019: Superquadric Surfaces as Primitives

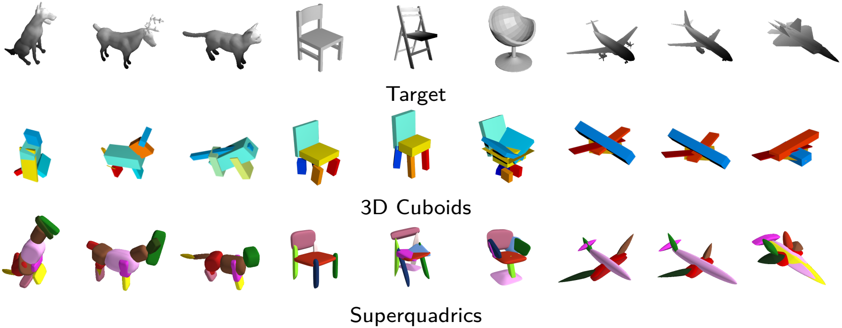
*Everything in nature takes its form from the **sphere**, the **cone** and the **cylinder**.* - Paul Cezanne.



Superquadrics ***allow complex solids and surfaces to be constructed and altered easily from a few interactive parameters.***

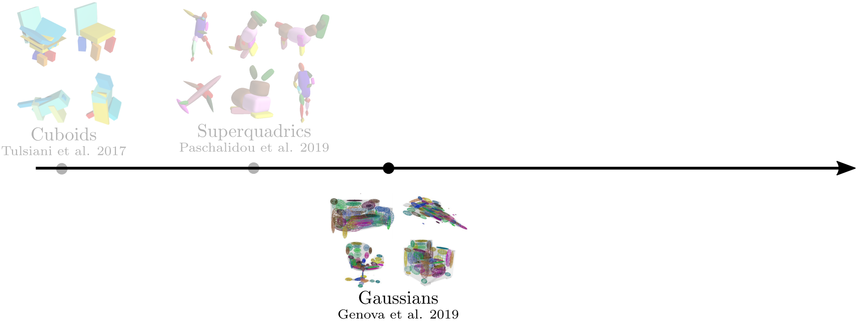
- Fully described with just 11 parameters
- Represent a diverse class of shapes such as cylinders, spheres, cuboids, ellipsoids in a **single continuous parameter space**
- Their large shape vocabulary allows for **faster** and **smoother fitting** than cuboids

2019: Superquadric Surfaces as Primitives

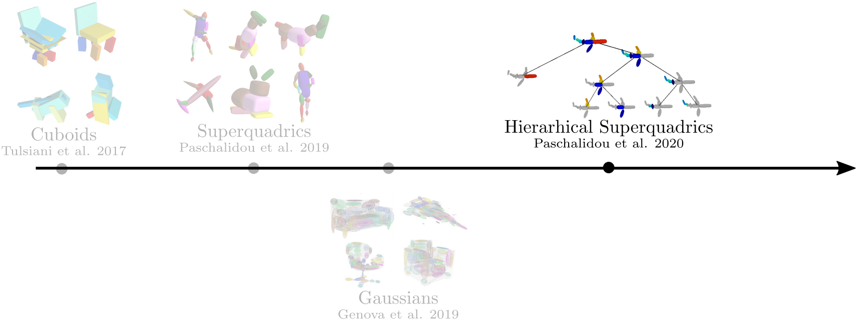


	Chamfer Distance			Volumetric IoU		
	Chairs	Aeroplanes	Animals	Chairs	Aeroplanes	Animals
3D Cuboids	0.0121	0.0153	0.0110	0.1288	0.0650	0.3339
Superquadrics	**0.0006**	**0.0003**	**0.0003**	**0.1408**	**0.1808**	**0.7506**

Unsupervised Primitive-based Representations

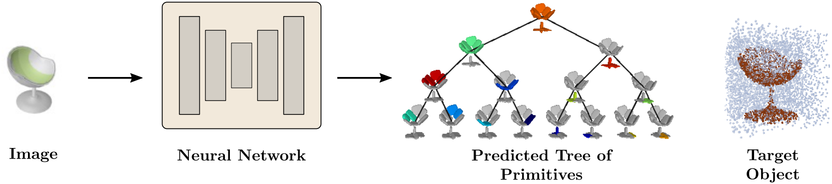


Unsupervised Primitive-based Representations

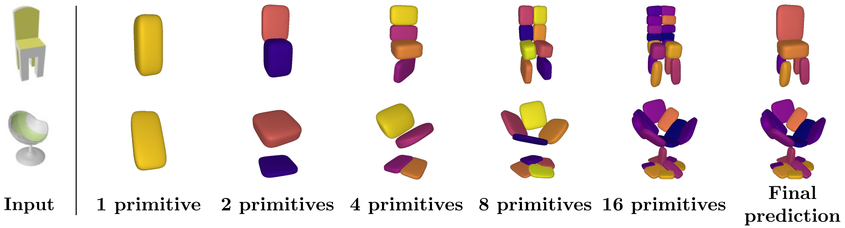


2020: Representating 3D Shapes with multiple levels of abstraction

Jointly recover **the geometry** and **the latent hierarchical layout** of an object.

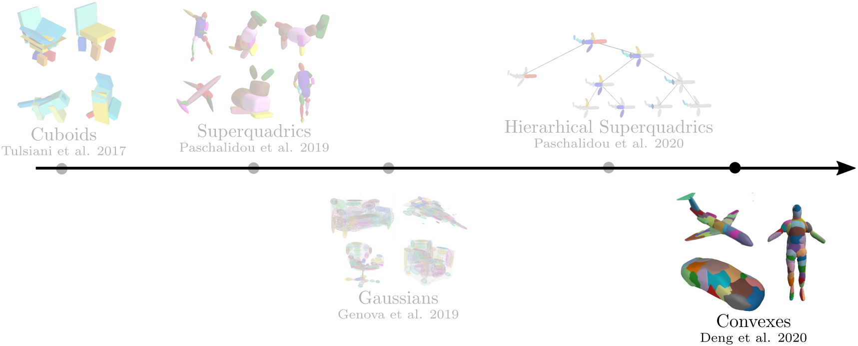


2020: Representating 3D Shapes with multiple levels of abstraction

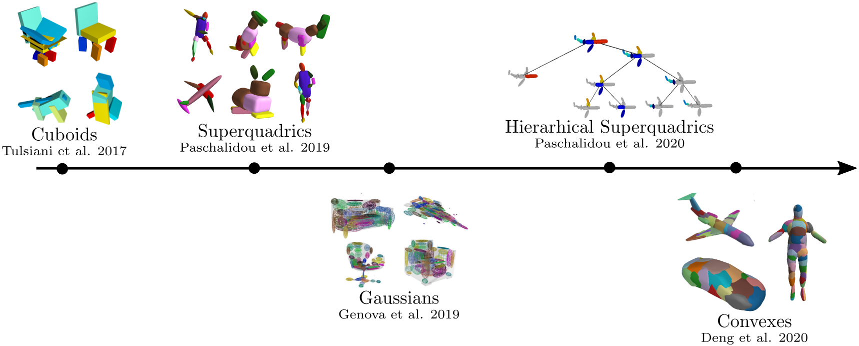


- Represent a 3D shape as a **binary tree of primitives**
- At each depth level, each node is **recursively** split into two until reaching the maximum depth
- Reconstructions from deeper depth levels are more detailed

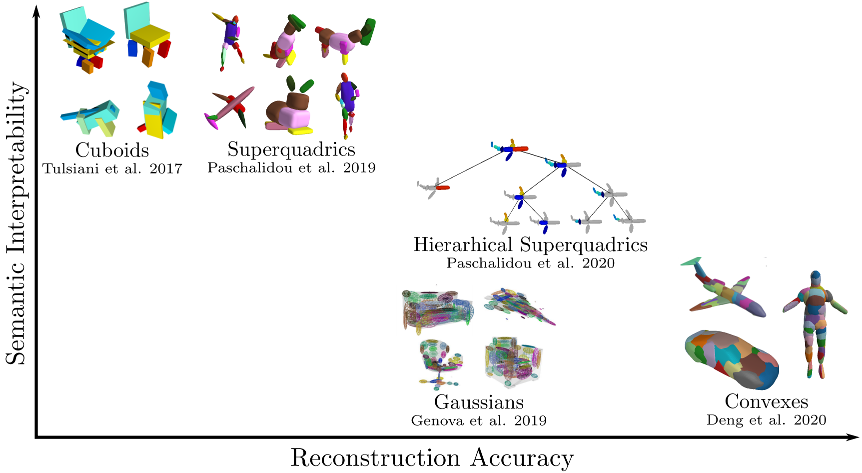
Unsupervised Primitive-based Representations



Unsupervised Primitive-based Representations



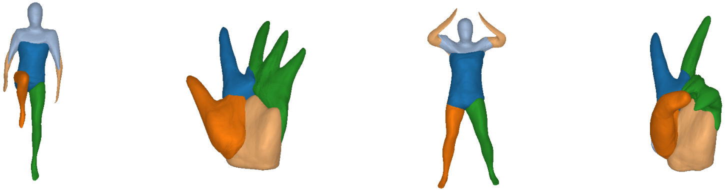
Unsupervised Primitive-based Representations



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

Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, Sanja Fidler

CVPR 2021

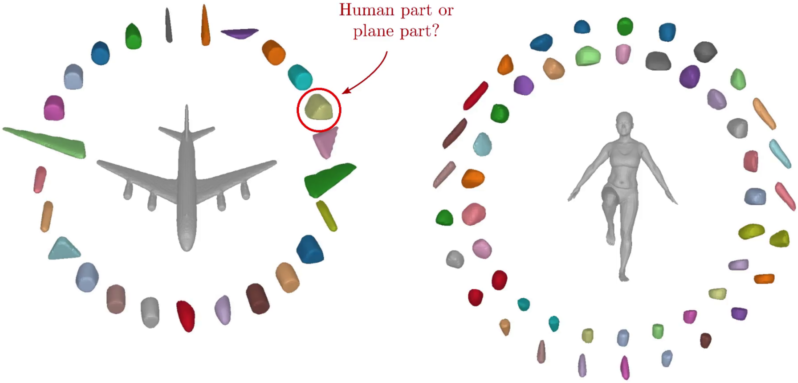


https://paschalidoud.github.io/neural_parts

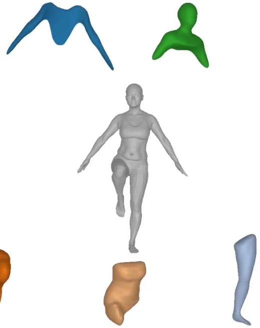
There exists a **trade-off** between the **number of primitives** and the **reconstruction quality** in primitive-based representations.

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

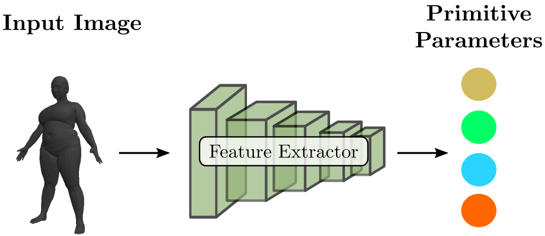
Simple parts require a large number of parts for accurate reconstructions.



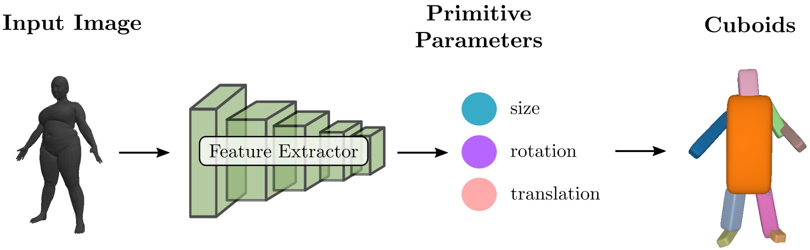
Neural Parts yield accurate and semantic reconstructions using an order of magnitude less parts.



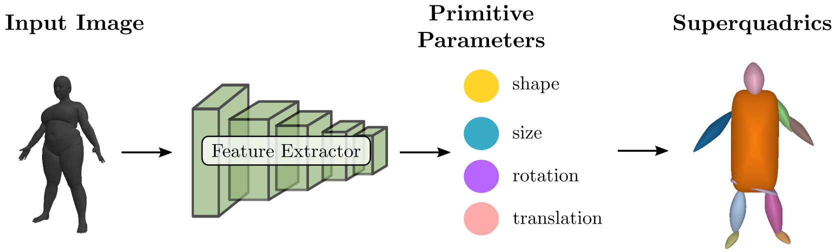
Primitive-based Learning



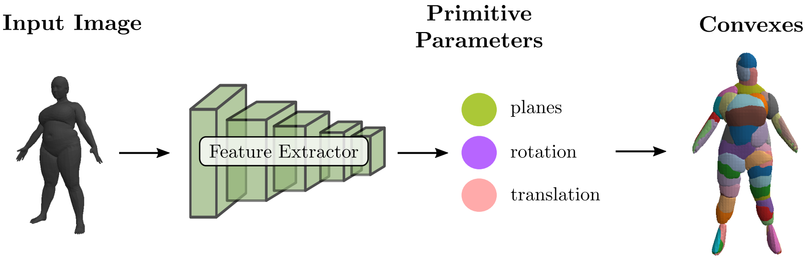
Primitive-based Learning



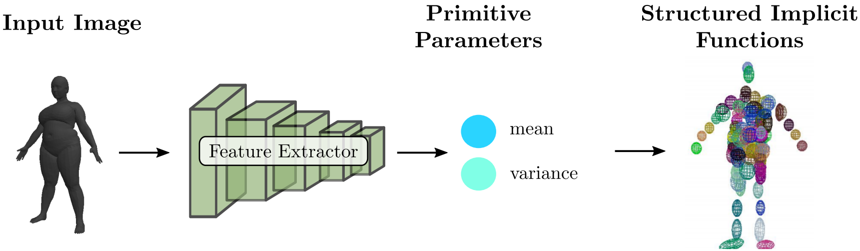
Primitive-based Learning



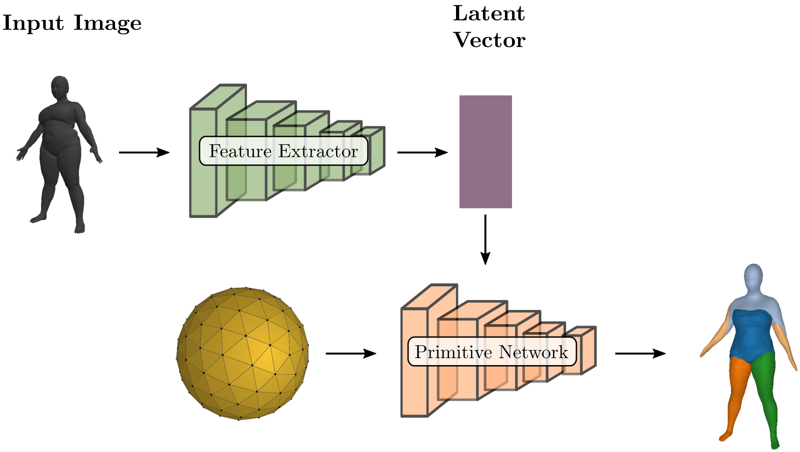
Primitive-based Learning



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

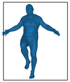
$$\mathbf{x} = \phi_{\boldsymbol{\theta}}(\mathbf{y}) \text{ and } \mathbf{y} = \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$$

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.

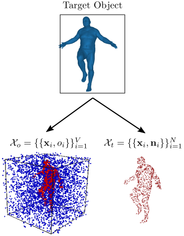
System Overview

Input Image

System Overview

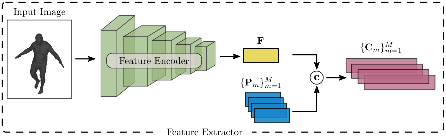


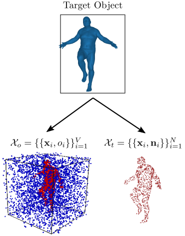
System Overview



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

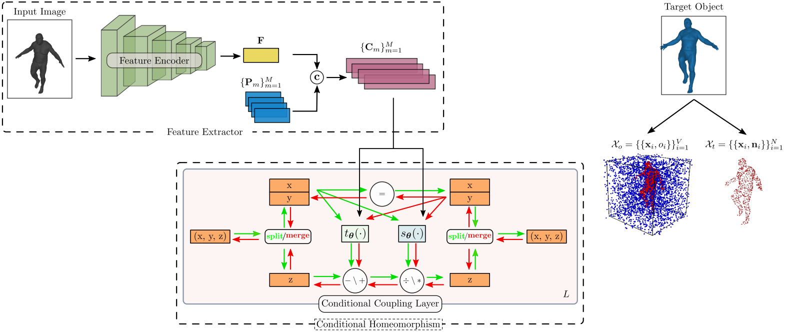
System Overview





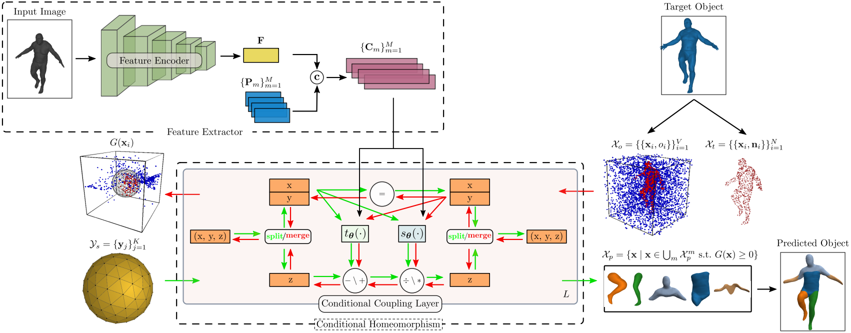
- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.

System Overview



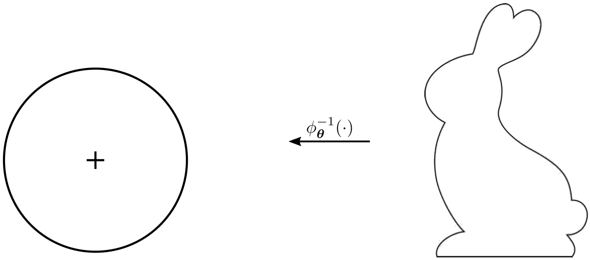
- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.
- The **conditional homeomorphism**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

System Overview



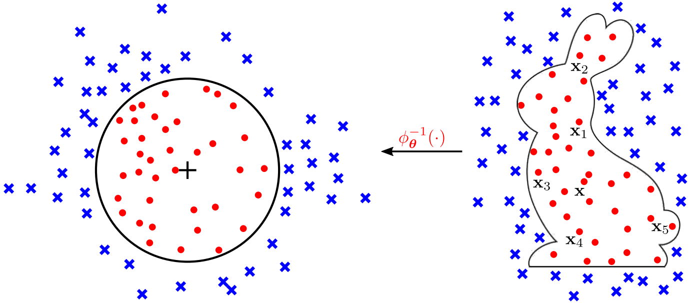
- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.
- The **conditional homeomorphism**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

Implicit Primitive Representation



where $\phi_{\boldsymbol{\theta}}^{-1}$ is the **inverse homeomorphic mapping from the primitive space to the sphere space.**

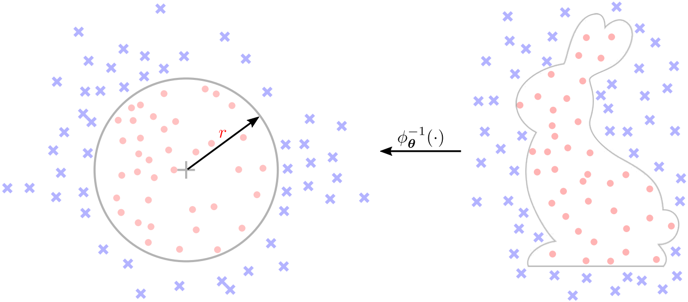
Implicit Primitive Representation



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

where $\phi_{\boldsymbol{\theta}}^{-1}$ is the **inverse homeomorphic mapping from the primitive space to the sphere space.**

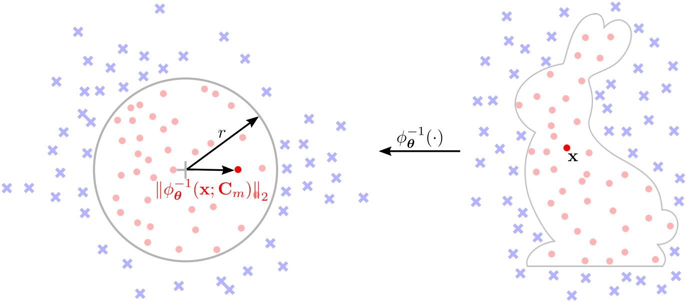
Implicit Primitive Representation



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

where $\phi_{\boldsymbol{\theta}}^{-1}$ is the **inverse homeomorphic mapping from the primitive space to the sphere space.**

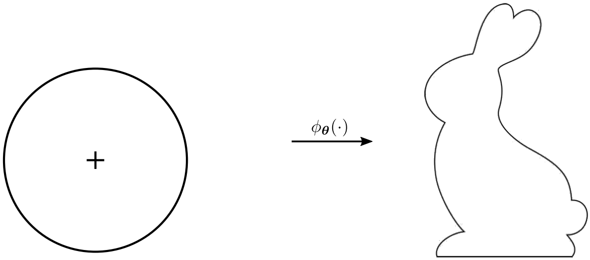
Implicit Primitive Representation



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

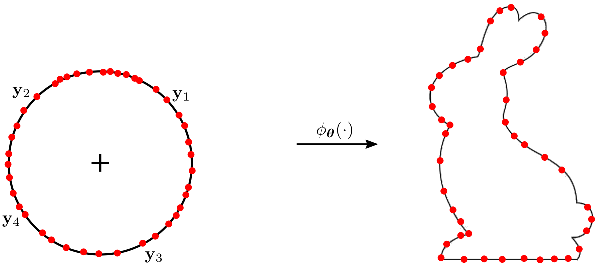
where $\phi_{\boldsymbol{\theta}}^{-1}$ is the **inverse homeomorphic mapping from the primitive space to the sphere space.**

Explicit Primitive Representation



where $\phi_{\boldsymbol{\theta}}$ is the **homeomorphic mapping from the sphere space to the primitive space.**

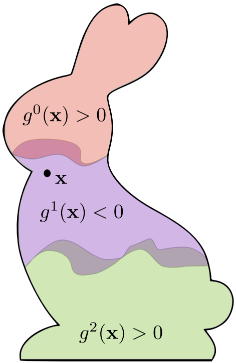
Explicit Primitive Representation



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

where $\phi_{\boldsymbol{\theta}}$ is the **homeomorphic mapping from the sphere space to the primitive space.**

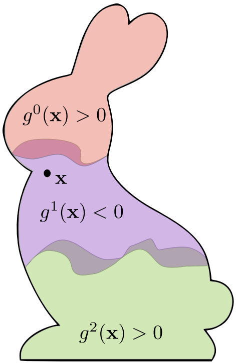
Implicit and Explicit Representation of Predicted Shape

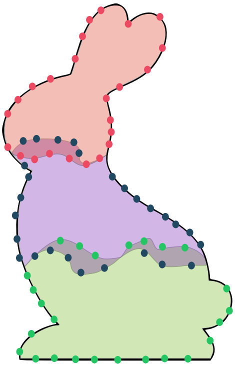


Implicit Representation:

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

Implicit and Explicit Representation of Predicted Shape





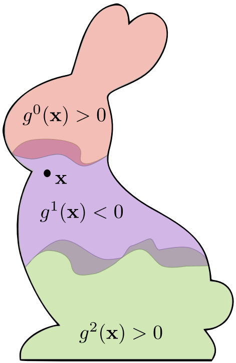
Implicit Representation:

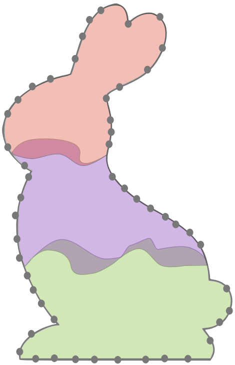
$$G(\mathbf{x}) = \min_{m \in 0 \dots M} g^m(\mathbf{x})$$

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:

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

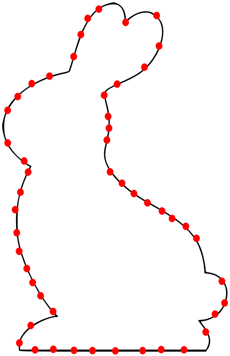
Composed of:

- $\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p)$: 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:

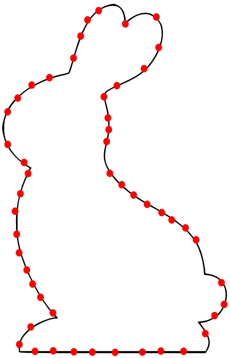
- **Target:**
  - **Surface Samples:** $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$
  - **Volumetric Samples:** $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$
- **Predicted:** $\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x})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

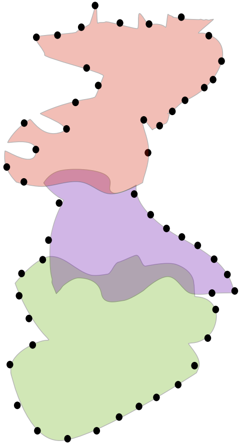
Reconstruction Loss



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

Reconstruction Loss

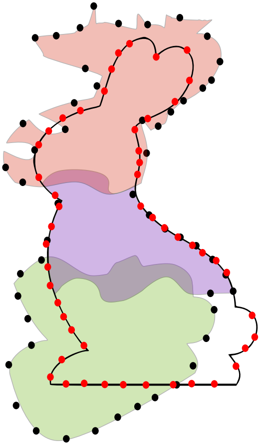




Target Surface Samples:
$\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^N$

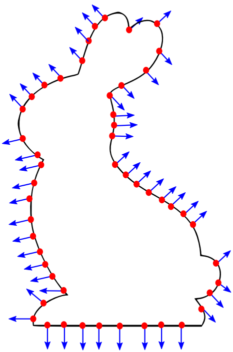
Predicted Surface Samples:
$\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$

Reconstruction Loss

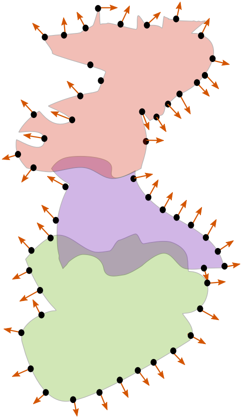


$$\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) = \frac{1}{|\mathcal{X}_t|} \sum_{\mathbf{x}_i \in \mathcal{X}_t} \min_{\mathbf{x}_j \in \mathcal{X}_p} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_j \in \mathcal{X}_p} \min_{\mathbf{x}_i \in \mathcal{X}_t} \|\mathbf{x}_i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

Normal Consistency Loss

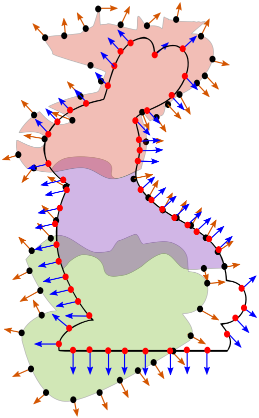


Target Surface Samples:
$\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^N$



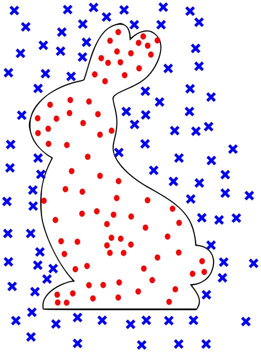
Predicted Surface Normals:
$\frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}$

Normal Consistency Loss



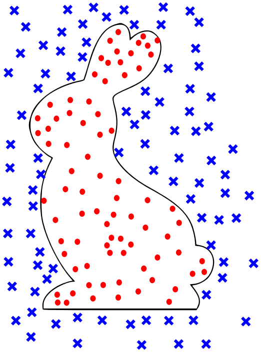
$$\mathcal{L}_{norm}(\mathcal{X}_t) = \frac{1}{|\mathcal{X}_t|} \sum_{(\mathbf{x},\mathbf{n}) \in \mathcal{X}_t} \left(1 - \left\langle \frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}, n \right\rangle\right)$$

Occupancy Loss

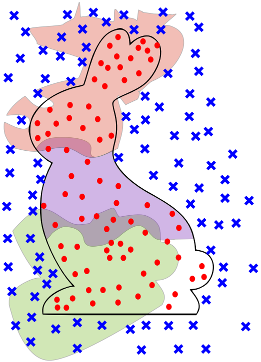


Target Volumetric Samples:
$\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Occupancy Loss

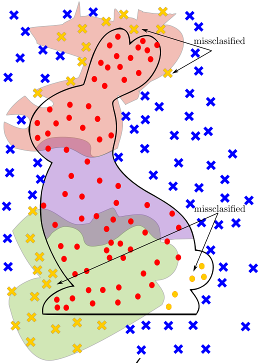


Target Volumetric Samples:
$\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$



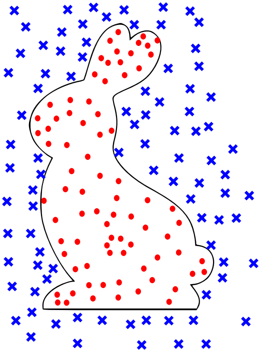
Predicted Volumetric Samples:
$G(\mathbf{x}) = \min_{m \in 0 \ldots M} g^m(\mathbf{x})$

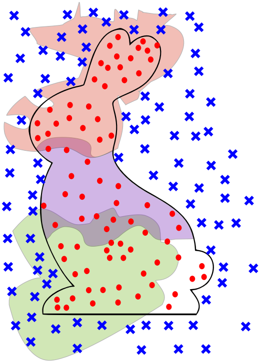
Occupancy Loss



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

Overlapping Loss

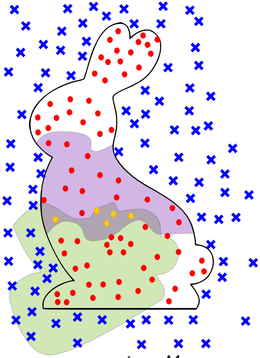




Target Volumetric Samples:
$\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

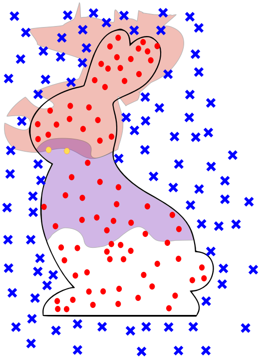
Predicted Volumetric Samples:
$G(\mathbf{x}) = \min_{m \in 0 \ldots M} g^m(\mathbf{x})$

Overlapping Loss



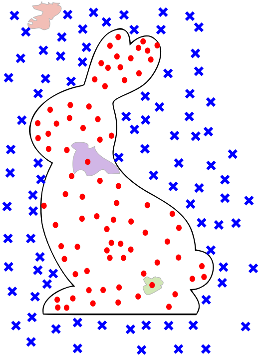
$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-g^m(\mathbf{x})}{\tau}\right) - \lambda\right)$$

Overlapping Loss

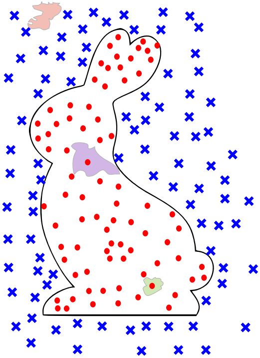


$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-g^m(\mathbf{x})}{\tau}\right) - \lambda\right)$$

Coverage Loss

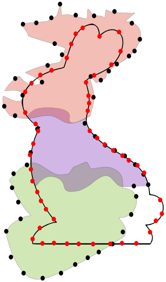


Coverage Loss



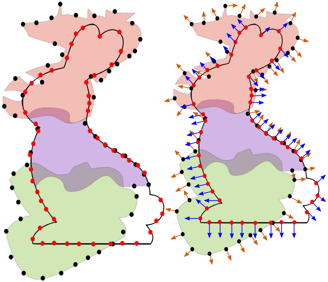
$$\mathcal{L}_{cover}(\mathcal{X}_o) = \sum_{m=1}^{M} \sum_{\mathbf{x} \in \mathcal{N}_k^m} \max\left(0, g^m(\mathbf{x})\right)$$

Loss Functions: Summary



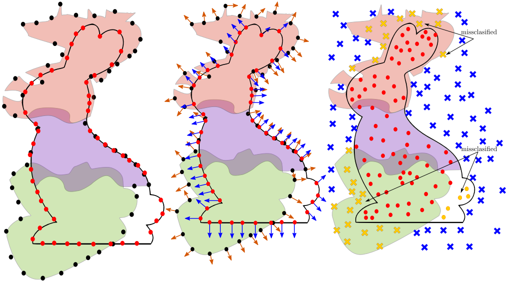
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.

Loss Functions: Summary



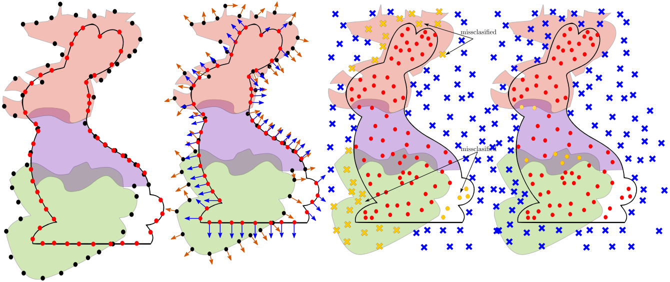
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.

Loss Functions: Summary



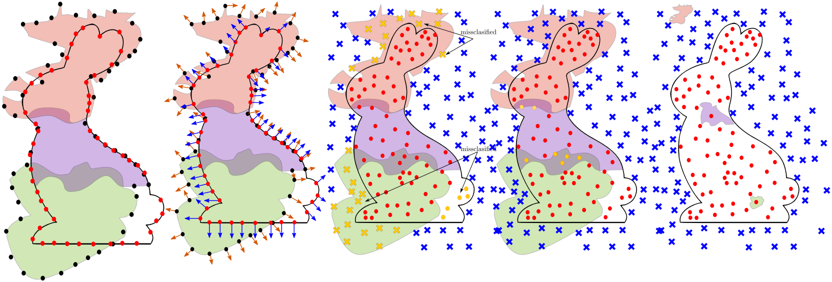
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.

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

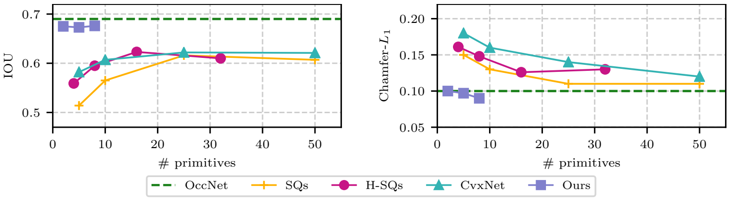


- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.
- **Overlapping Loss**: Prevent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

How well does it work?

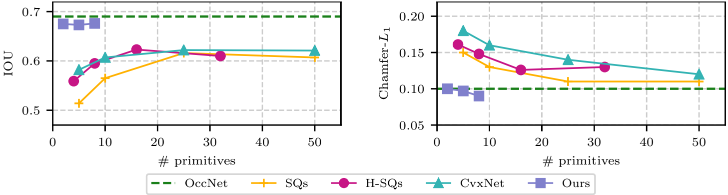
Representation Power of Primitive-based Representations

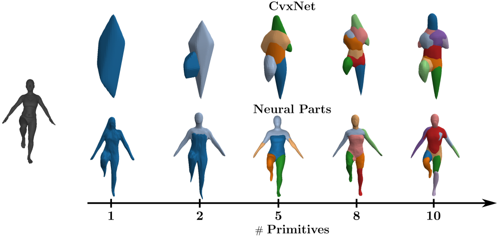
Neural Parts decouple the reconstruction quality from the number of parts.



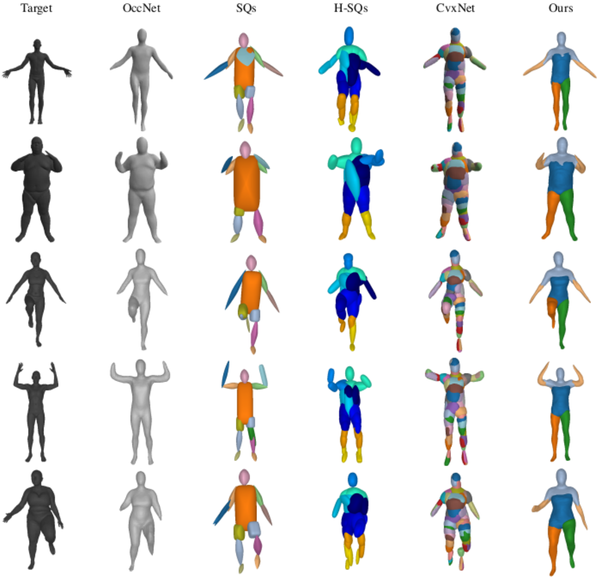
Representation Power of Primitive-based Representations

Neural Parts decouple the reconstruction quality from the number of parts.



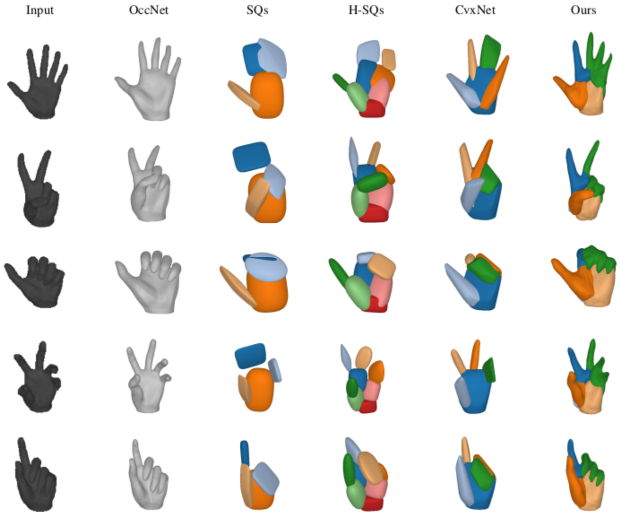


Single-view 3D Reconstruction on D-FAUST

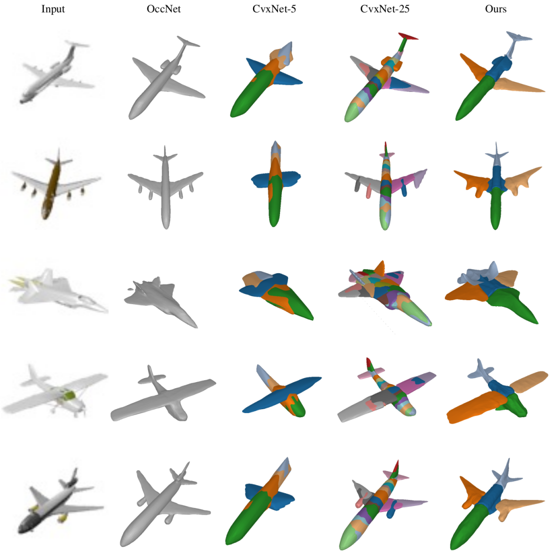


Neural Parts: Learning Expressive 3D Shape Abstractions with Invertible Neural Networks, CVPR, 2021

Single-view 3D Reconstruction on FreiHAND

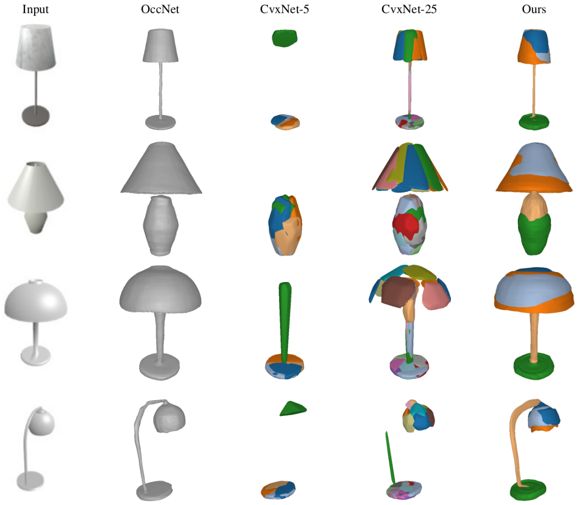


Single-view 3D Reconstruction on ShapeNet



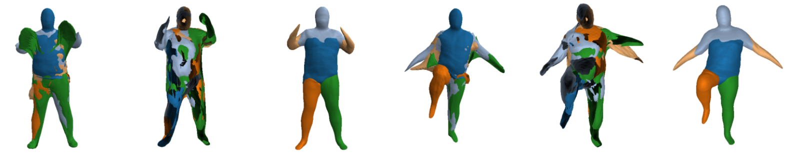
Single-view 3D Reconstruction on ShapeNet

Single-view 3D Reconstruction on ShapeNet



Do we really need an INN?

(a) w/o $\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$ (b) AtlasNet-sphere (c) Ours (d) w/o $\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$ (e) AtlasNet-sphere (f) Ours



	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

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.
- We demonstrate that **implementing homeomorphims with an INN is better than an MLP**.
- Neural Parts have well defined explicit and implicit formulations.

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.
- We demonstrate that **implementing homeomorphims with an INN is better than an MLP**.
- Neural Parts have well defined explicit and implicit formulations.
- Neural Parts **do not impose any constraint on the shape of the predicted primitive**.

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.
- We demonstrate that **implementing homeomorphims with an INN is better than an MLP**.
- Neural Parts have well defined explicit and implicit formulations.
- Neural Parts **do not impose any constraint on the shape of the predicted primitive**.
- Neural Parts allow to **decouple the reconstruction quality from the number of parts**, thus they yield both geometrically accurate and semantically meaningful shape abstractions.

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.
- We demonstrate that **implementing homeomorphims with an INN is better than an MLP**.
- Neural Parts have well defined explicit and implicit formulations.
- Neural Parts **do not impose any constraint on the shape of the predicted primitive**.
- Neural Parts allow to **decouple the reconstruction quality from the number of parts**, thus they yield both geometrically accurate and semantically meaningful shape abstractions.
- Limitations:

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.
- We demonstrate that **implementing homeomorphims with an INN is better than an MLP**.
- Neural Parts have well defined explicit and implicit formulations.
- Neural Parts **do not impose any constraint on the shape of the predicted primitive**.
- Neural Parts allow to **decouple the reconstruction quality from the number of parts**, thus they yield both geometrically accurate and semantically meaningful shape abstractions.
- Limitations:
  - High computational requirements due to the INN for the case of multiple primitives (e.g. for scenes).

Summary

- We propose Neural Parts, **a novel 3D primitive representation** as the homeomorphic mapping between a sphere and the target shape.
- We demonstrate that **implementing homeomorphims with an INN is better than an MLP**.
- Neural Parts have well defined explicit and implicit formulations.
- Neural Parts **do not impose any constraint on the shape of the predicted primitive**.
- Neural Parts allow to **decouple the reconstruction quality from the number of parts**, thus they yield both geometrically accurate and semantically meaningful shape abstractions.
- Limitations:
  - High computational requirements due to the INN for the case of multiple primitives (e.g. for scenes).
  - Similar to all primitive-based representations, the reconstructed parts are **spatially consistent without necessarily being semantic**.

ATISS: Autoregressive Transformers for Indoor Scene Synthesis

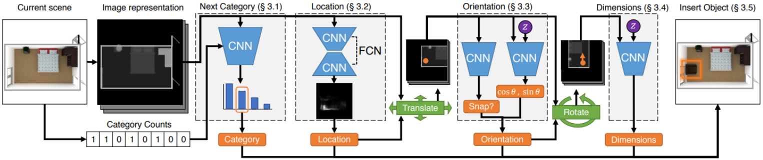
Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis,
Andreas Geiger, Sanja Fidler

Under Review

https://paschalidoud.github.io/atiss

Existing scene synthesis methods
impose unnatural constraints on the scene generation process
since they represent **scenes as ordered sequences of objects**.

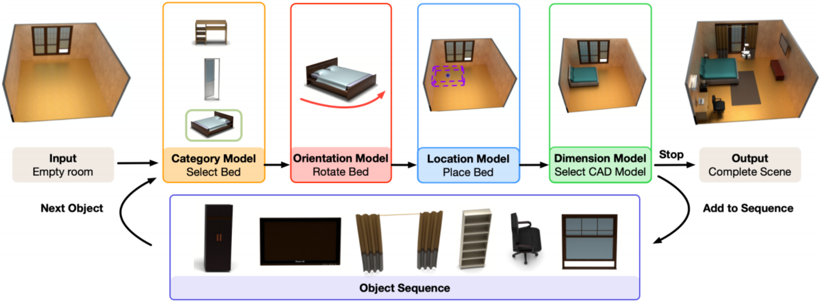
2019: Scenes as Ordered Sequences of Objects



- **Autoregressive, CNN-based generative model** of scenes as **ordered sequences of objects.**
- Supervision in the form of **2D labelled bounding boxes** as well as **auxiliary supervision** such as depth maps and object segmentation masks.
- **Operates on top-down image-based representation of a scene**, thus requires rendering after adding an object which makes it **very slow**.
- Limited applications due to the ordered sequence formulation.

Ritchie and Wang: Fast and Flexible Indoor Scene Synthesis via Deep Convolutional Generative Models. CVPR, 2019

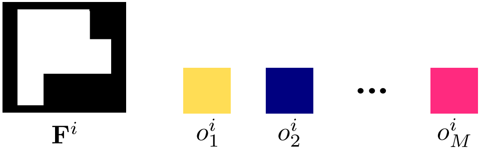
2020: Scenes as Ordered Sequences of Objects



- A series of transformers that **autoregressively** adds objects in a scene.
- Scenes are parametrized as **ordered sequences of objects**.
- Supervision in the form of **2D labelled bounding boxes**.
- Limited applications due to the ordered sequence formulation.

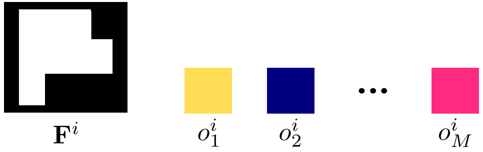
ATISS: Scene Parametrization

Each scene $\mathcal{X}_i = \left(\mathcal{O}_i, \mathbf{F}^i\right)$ comprises the **unordered set of M objects** in the scene $\mathcal{O}_i = \{o_j^i\}_{j=1}^M$ and its **floor shape $\mathbf{F}^i$**.



ATISS: Scene Parametrization

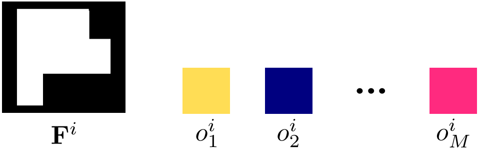
Each scene $\mathcal{X}_i = \left(\mathcal{O}_i, \mathbf{F}^i\right)$ comprises the **unordered set of M objects** in the scene $\mathcal{O}_i = \{o_j^i\}_{j=1}^M$ and its **floor shape $\mathbf{F}^i$**.



- The floor shape is modelled as the **top-down orthographic projection** of the scene's floor.

ATISS: Scene Parametrization

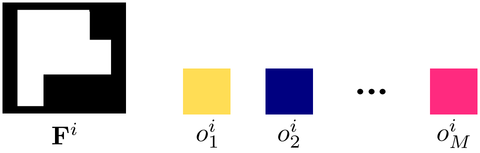
Each scene $\mathcal{X}_i = \left(\mathcal{O}_i, \mathbf{F}^i\right)$ comprises the **unordered set of** M **objects** in the scene $\mathcal{O}_i = \{o_j^i\}_{j=1}^M$ and its **floor shape** $\mathbf{F}^i$.



- The floor shape is modelled as the **top-down orthographic projection** of the scene's floor.
- Each 3D object is modelled with four random variables that describe their **category, size, orientation and location**, $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{t}_j, \mathbf{r}_j\}$.

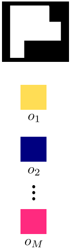
ATISS: Scene Parametrization

Each scene $\mathcal{X}_i = \left(\mathcal{O}_i, \mathbf{F}^i\right)$ comprises the **unordered set of M objects** in the scene $\mathcal{O}_i = \{o_j^i\}_{j=1}^{M}$ and its **floor shape $\mathbf{F}^i$**.



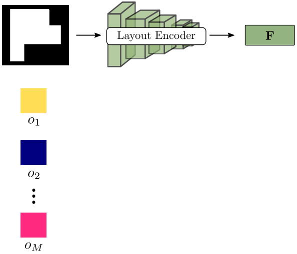
- The floor shape is modelled as the **top-down orthographic projection** of the scene's floor.
- Each 3D object is modelled with four random variables that describe their **category, size, orientation and location**, $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{t}_j, \mathbf{r}_j\}$.
- The object category $\mathbf{c}_j$ is modelled using a **categorical variable over the total number of object categories** in and the size $\mathbf{s}_j$, location $\mathbf{t}_j$ and orientation $\mathbf{r}_j$ are modelled with a **mixture of logistics distributions**.

ATISS: Scene Generation



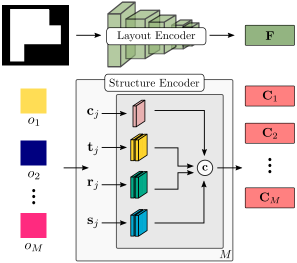
Starting from a scene parameterized as its **unordered set of** M **objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.

ATISS: Scene Generation



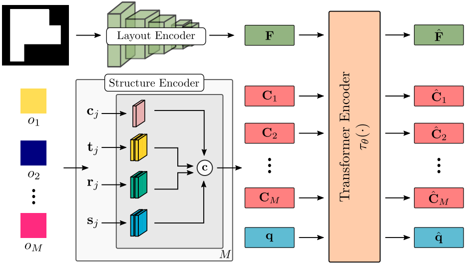
Pass the floor shape to the **layout encoder** and extract a feature representation for the floor.

ATISS: Scene Generation



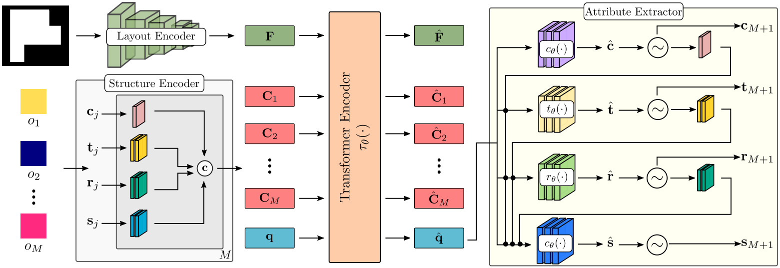
Map each object in the scene o_j to a per-object context embedding $\mathbf{C}_j$.

ATISS: Scene Generation



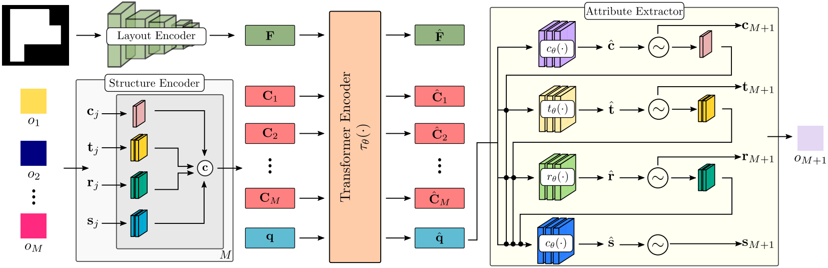
$\mathbf{F}, \mathbf{C} = \{\mathbf{C}_j\}_{j=1}^{M}$ and a **query embedding** $\mathbf{q}$ are passed to a transformer encoder that **predicts the features of the next object to be added in the scene.**

ATISS: Scene Generation



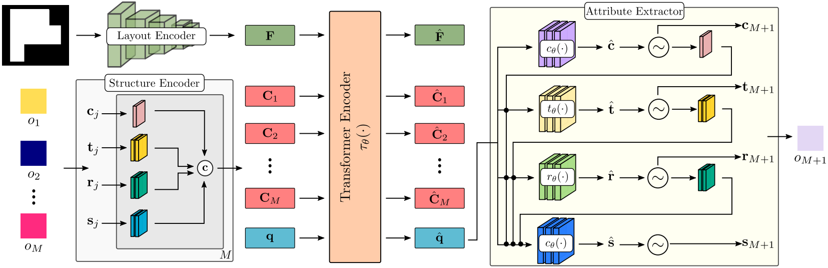
Using the predicted features $\hat{\mathbf{q}}$ the **attribute extractor** autoregressively predicts the object attributes of the next object to be added in the scene.

ATISS: Scene Generation



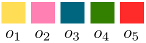
Once a new object is generated, it is appended to the objects already in the scene to be used in the next step of the generation process, **until the end symbol is generated**.

ATISS: Scene Generation

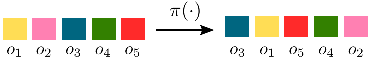


- We train ATISS to **maximize the log-likelihood of all possible permutations of object arrangements** in a collection of scenes.
- This enforces that adding an object in the scene is **equiprobable regardless of the order of the previously added objects.**

ATISS: Training Overview

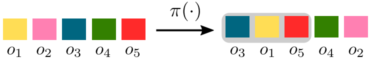


ATISS: Training Overview



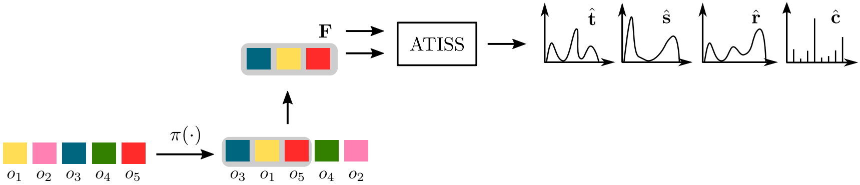
- We apply a random permutation $\pi(\cdot)$ on the M objects of a scene.

ATISS: Training Overview



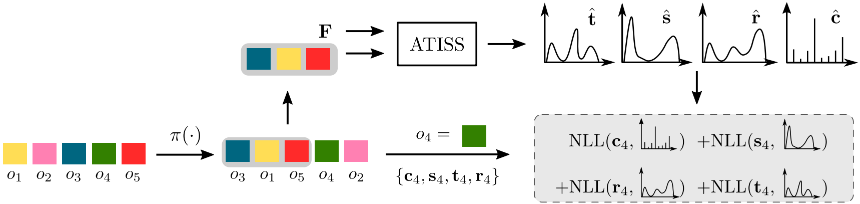
- We apply a random permutation $\pi(\cdot)$ on the M objects of a scene.
- We randomly select the first T objects to compute the context embedding $\mathbf{C}$.

ATISS: Training Overview



- We apply a random permutation $\pi(\cdot)$ on the M objects of a scene.
- We randomly select the first T objects to compute the context embedding $\mathbf{C}$.
- Conditioned on the $\mathbf{C}$ and $\mathbf{F}$, ATISS **predicts the attribute distributions of the next object** to be added in the scene.

ATISS: Training Overview



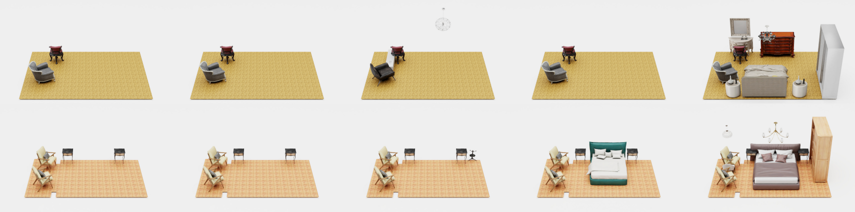
- We apply a random permutation $\pi(\cdot)$ on the M objects of a scene.
- We randomly select the first T objects to compute the context embedding $\mathbf{C}$.
- Conditioned on the $\mathbf{C}$ and $\mathbf{F}$, ATISS **predicts the attribute distributions of the next object** to be added in the scene.
- ATISS is trained to maximize the log likelihood of the $T + 1$ object from the permuted set of objects.

How well does it work?

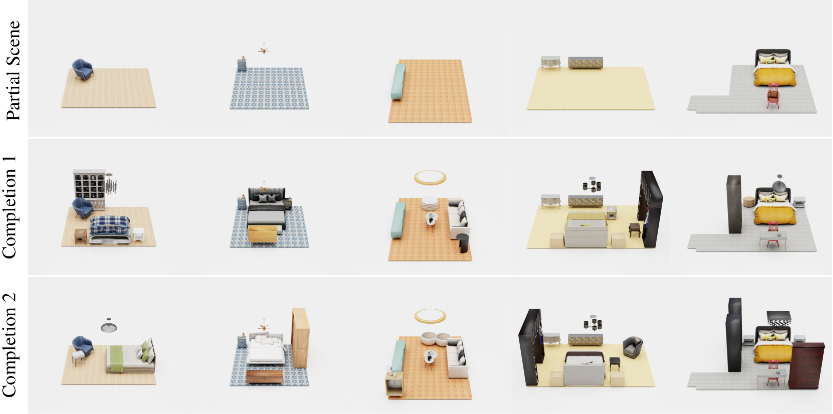
Scene Completion

We compare scene completions using our model, SceneFormer and FastSynth.

Partial Scene	FastSynth	SceneFormer	Ours+Order	Ours

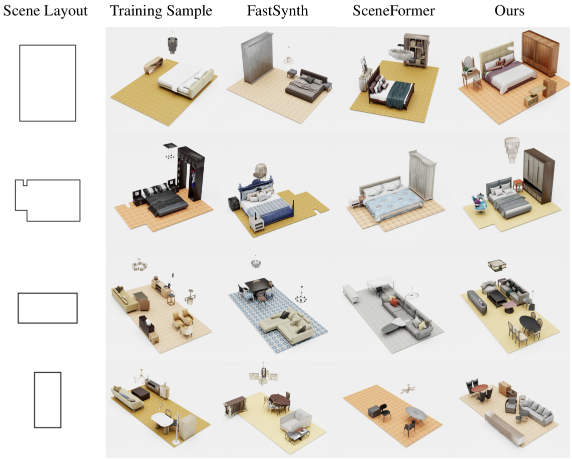


Scene Completion



Scene Synthesis

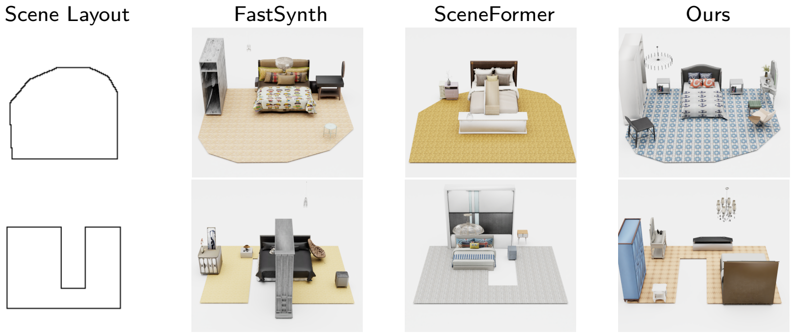
We compare the generated scenes conditioned on various floor shapes and room types using ATISS, SceneFormer and FastSynth.



Scene Synthesis

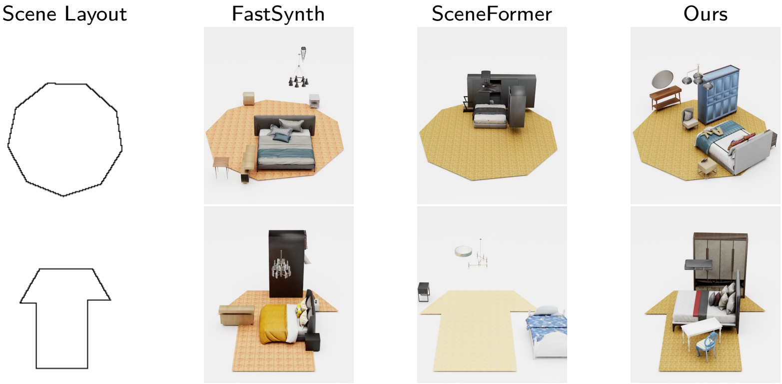
Generalization Beyond Training Data

ATISS generates plausible object arrangements conditioned on manually designed floor plans.



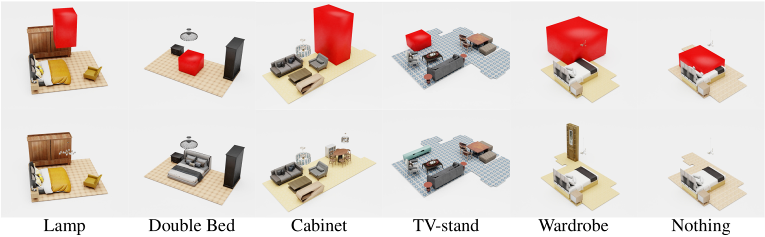
Generalization Beyond Training Data

ATISS generates plausible object arrangements conditioned on manually designed floor plans.



Objects Suggestion

ATISS can suggest objects given user-specified location constraints.



Failure Cases Correction

ATISS identifies problematic object arrangments and repositions them.

Generation Time

	Bedroom	Living	Dining	Library
FastSynth	13193.77	30578.54	26596.08	10813.87
SceneFormer	849.37	731.84	901.17	369.74
Ours	**102.38**	**201.59**	**201.84**	**88.24**

- At least $100\times$ faster than the CNN-based FastSynth for all room types.
- At least $4\times$ faster than the Transformer-based SceneFormer for all room types.

Summary

- We propose ATISS **a novel autoregressive model for unordered set generation**.

Summary

- We propose ATISS **a novel autoregressive model for unordered set generation**.
- We demonstrate that our unordered set formulation **opens up multiple interactive applications**.

Summary

- We propose ATISS **a novel autoregressive model for unordered set generation**.
- We demonstrate that our unordered set formulation **opens up multiple interactive applications**.
- ATISS has fewer parameters, **is simpler to implement and train and runs up to 8x faster** than existing methods.

Summary

- We propose ATISS **a novel autoregressive model for unordered set generation**.
- We demonstrate that our unordered set formulation **opens up multiple interactive applications**.
- ATISS has fewer parameters, **is simpler to implement and train and runs up to 8x faster** than existing methods.
- Limitations:

Summary

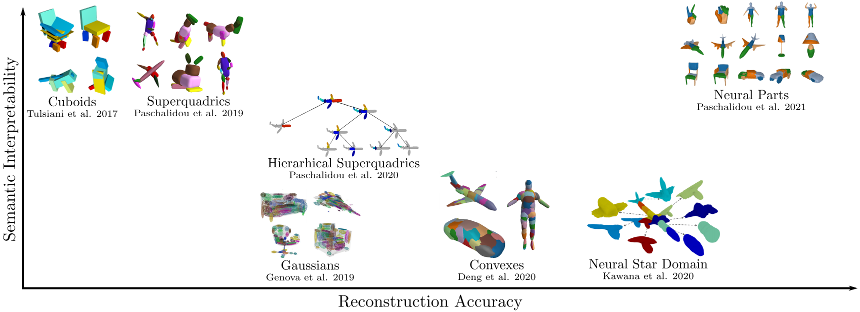
- We propose ATISS **a novel autoregressive model for unordered set generation**.
- We demonstrate that our unordered set formulation **opens up multiple interactive applications**.
- ATISS has fewer parameters, **is simpler to implement and train and runs up to 8x faster** than existing methods.
- Limitations:
  - The autoregressive generation of attributes need to follow a specific ordering.

Summary

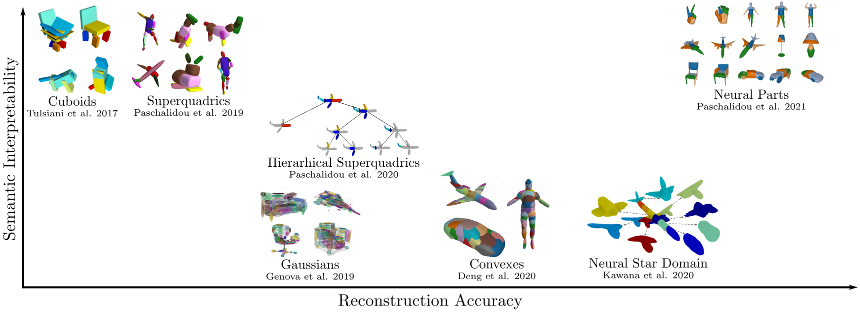
- We propose ATISS **a novel autoregressive model for unordered set generation**.
- We demonstrate that our unordered set formulation **opens up multiple interactive applications**.
- ATISS has fewer parameters, **is simpler to implement and train and runs up to 8x faster** than existing methods.
- Limitations:
  - The autoregressive generation of attributes need to follow a specific ordering.
  - Separate object retrieval module.

What comes next?

Primitive Arena

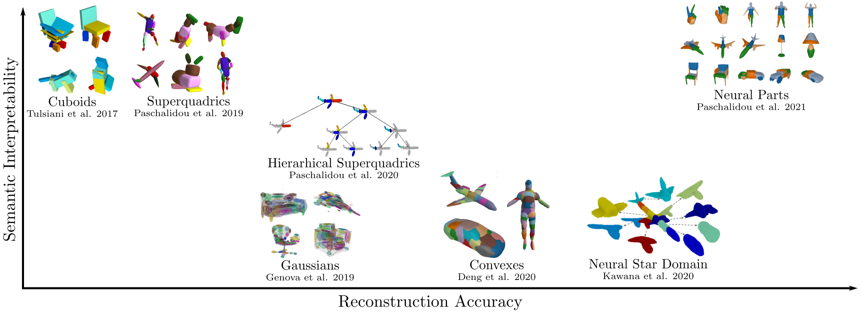


Primitive Arena



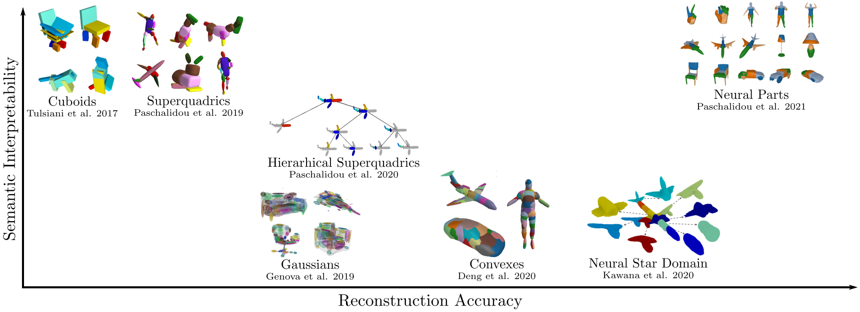
- What makes a **good primitive-representation**?

Primitive Arena



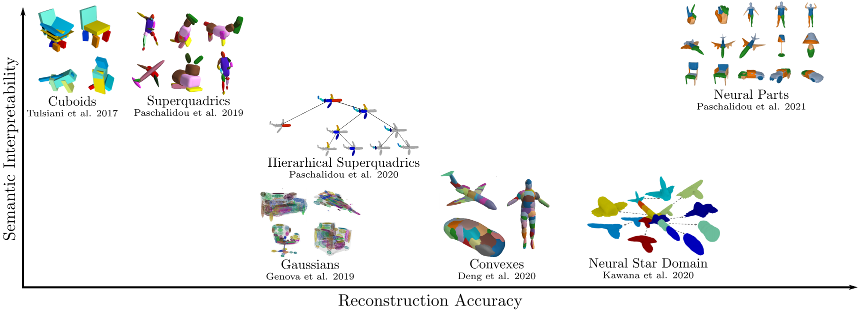
- What makes a **good primitive-representation**?
- We learn primitives by **optimizing the geometry**? Can't we do better?

Primitive Arena



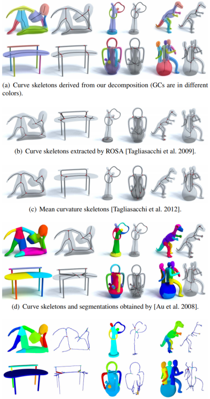
- What makes a **good primitive-representation**?
- We learn primitives by **optimizing the geometry**? Can't we do better?
- **Do we really learn semantic parts**?

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

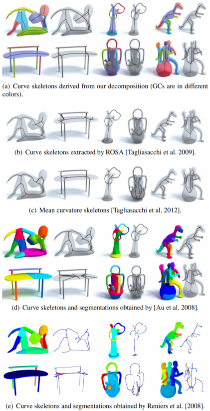


Image Source: Generalized Cylinder Decomposition, 2015

Learning parts through skeletonization

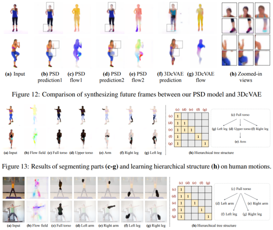


Image Source: Unsupervised Discovery of Parts, Structure and Dynamics, 2019

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

Learning semantic parts without part-level supervision

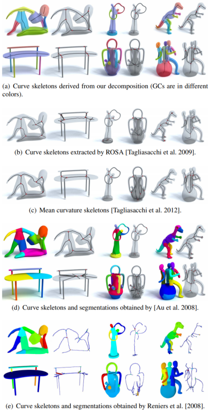


Image Source: Generalized Cylinder Decomposition, 2015

Learning parts through skeletonization

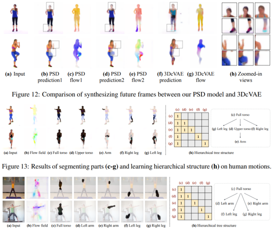


Image Source: Unsupervised Discovery of Parts, Structure and Dynamics, 2019

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

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

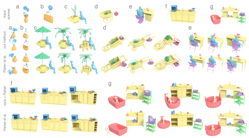


Image Source: Relationship Templates for Creating Scene Variations, 2016

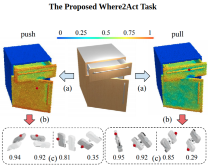


Image Source: Where2Act: From Pixels to Actions for Articulated 3D Objects, 2021

Learning functional parts

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

Thank you for your attention!