Learning Deep Models with Primitive-Based Representations

Despoina Paschalidou

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

The slides for this talk are available online at:

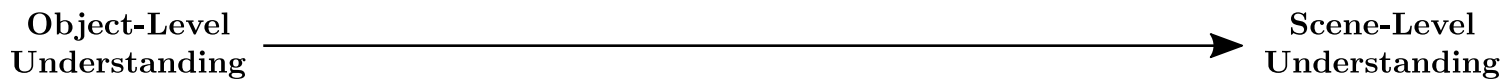
https://paschalidoud.github.io/talks/phd-defense.pdf


ΚΑΦΕ ΟΥΖΕΡΙ
Γιώργη


WALL·E

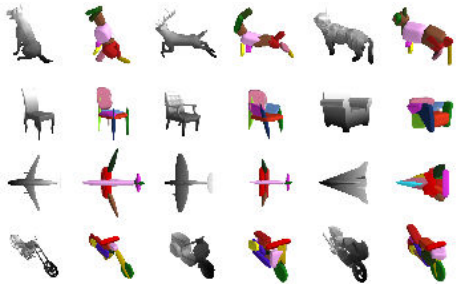
To achieve true AI we need to develop systems that can robustly **reason about the world both in object level and in scene level.**

About My Research

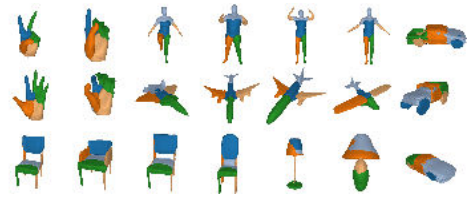


About My Research

Expressive Part-based Representations



Superquadrics Revisited:
Learning 3D Shape Parsing beyond Cuboids
CVPR 2019

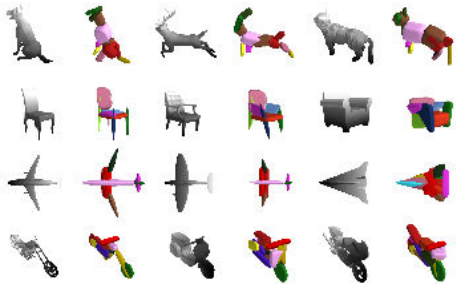


Neural Parts:
Learning Expressive 3D Shape Abstractions
with Invertible Neural Networks
CVPR 2021

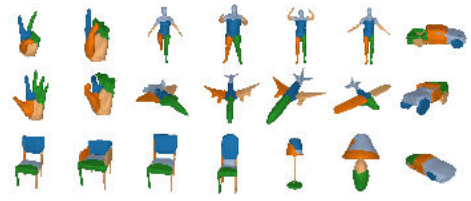
Object-Level Understanding → **Scene-Level Understanding**

About My Research

Expressive Part-based Representations

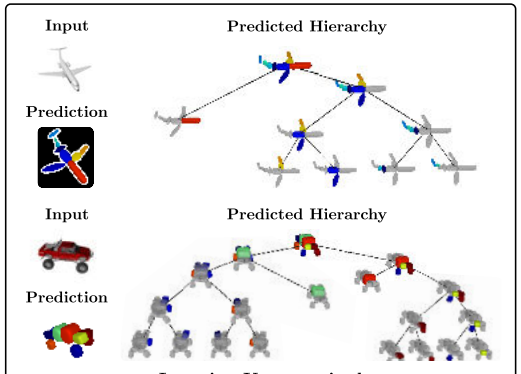


Superquadrics Revisited:
Learning 3D Shape Parsing beyond Cuboids
CVPR 2019



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

Structure-Aware Part-based Representations

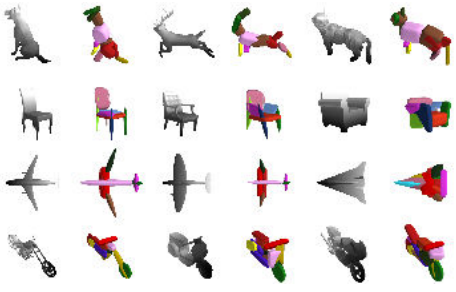


Learning Unsupervised
Hierarchical Part Decomposition
of 3D Objects from a Single RGB Image
CVPR 2020

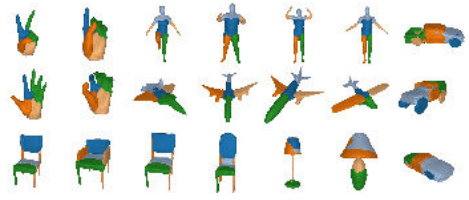
Object-Level Understanding → **Scene-Level Understanding**

About My Research

Expressive Part-based Representations

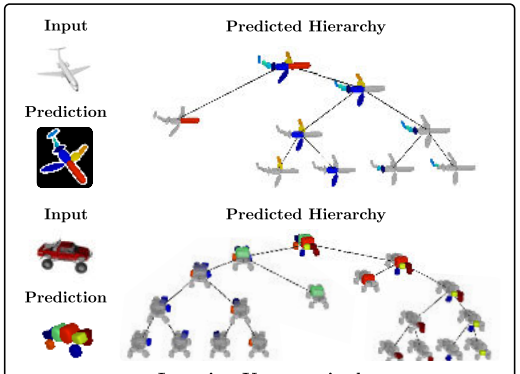


Superquadrics Revisited:
Learning 3D Shape Parsing beyond Cuboids
CVPR 2019



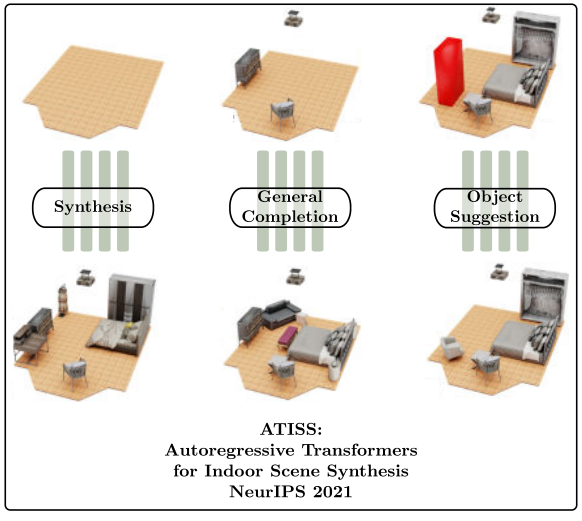
Neural Parts:
Learning Expressive 3D Shape Abstractions
with Invertible Neural Networks
CVPR 2021

Structure-Aware Part-based Representations



Learning Unsupervised
Hierarchical Part Decomposition
of 3D Objects from a Single RGB Image
CVPR 2020

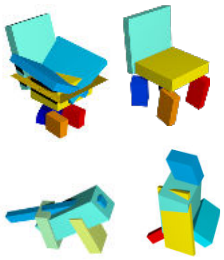
Structure-Aware Object-based Representations for Scene Synthesis



ATISS:
Autoregressive Transformers
for Indoor Scene Synthesis
NeurIPS 2021

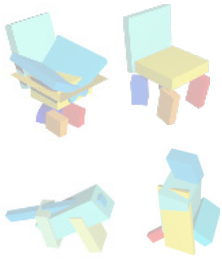
Object-Level Understanding → **Scene-Level Understanding**

Expressive Part-based Representations

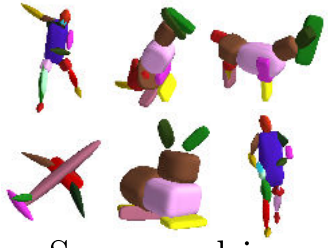


Cuboids
Tulsiani et al. 2017

Expressive Part-based Representations



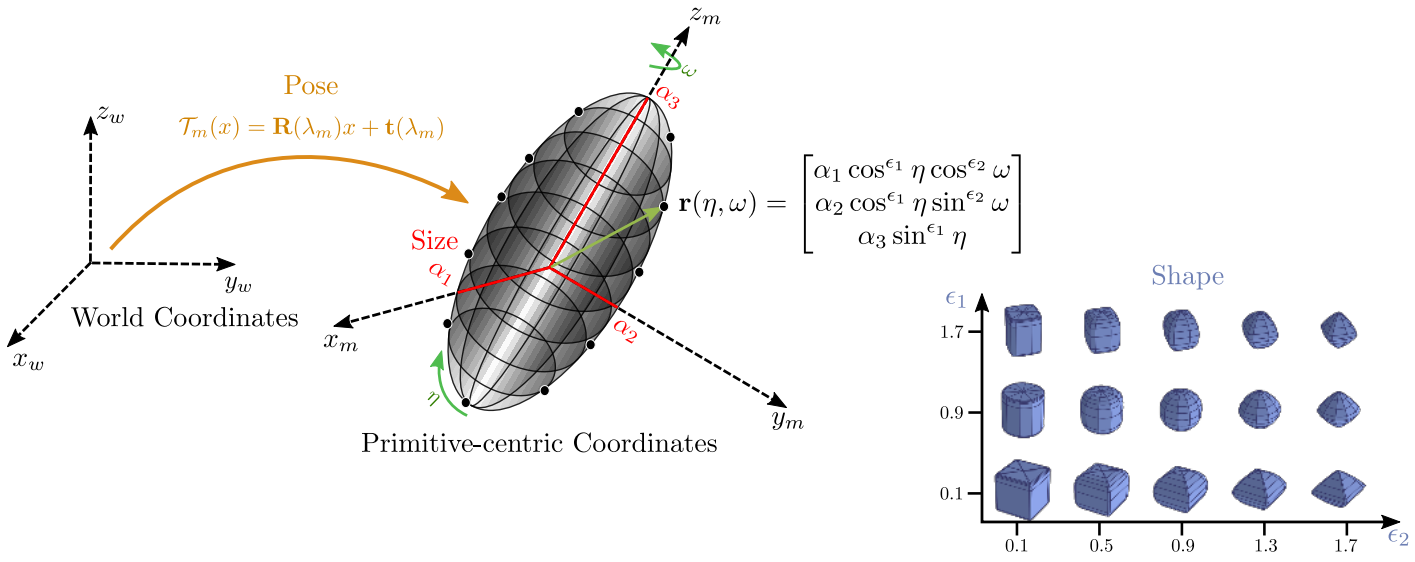
Cuboids
Tulsiani et al. 2017



Superquadrics
Paschalidou et al. 2019

Superquadric Surfaces as Primitives

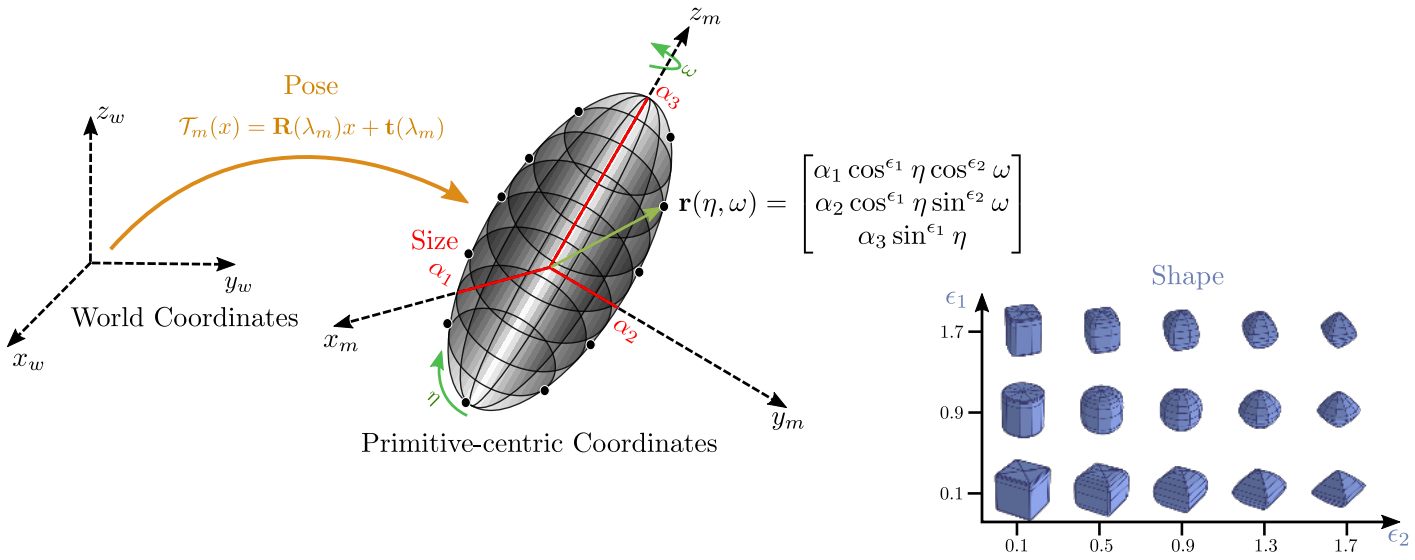
*Everything in nature takes its form from the **sphere**, the **cone** and the **cylinder**.* - Paul Cezanne.



Superquadrics **represent a diverse class of shapes in a single continuous parameter space.**

Superquadric Surfaces as Primitives

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

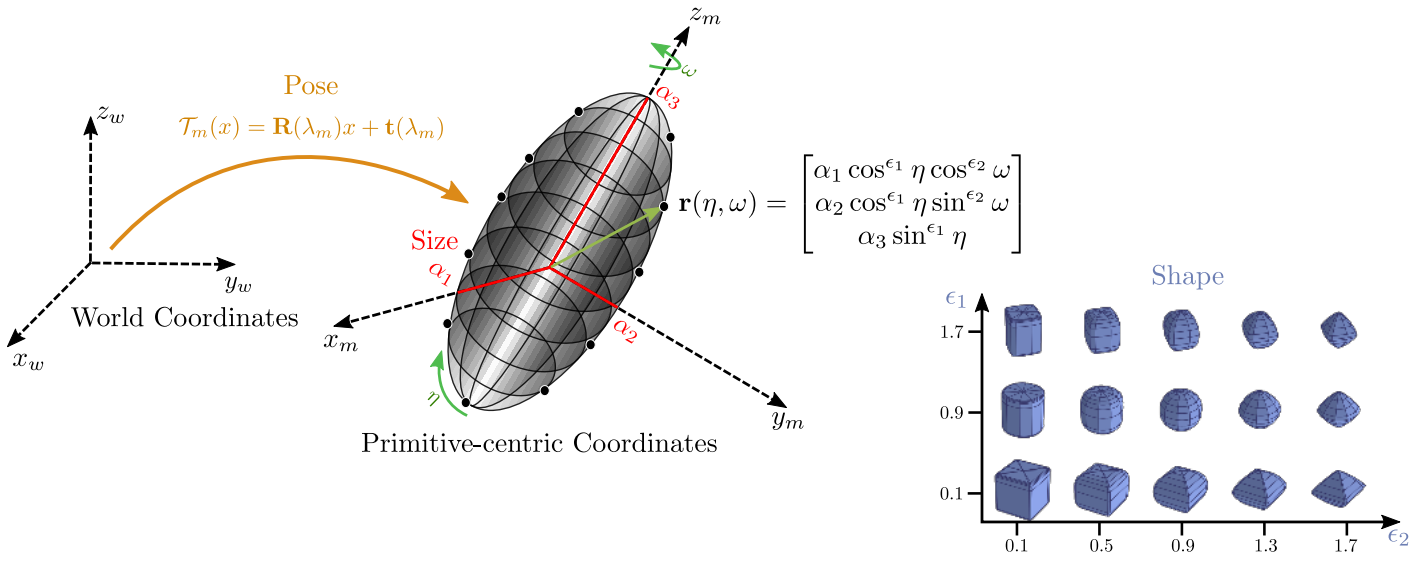


Superquadrics **represent a diverse class of shapes in a single continuous parameter space.**

- Fully described with just 11 parameters

Superquadric Surfaces as Primitives

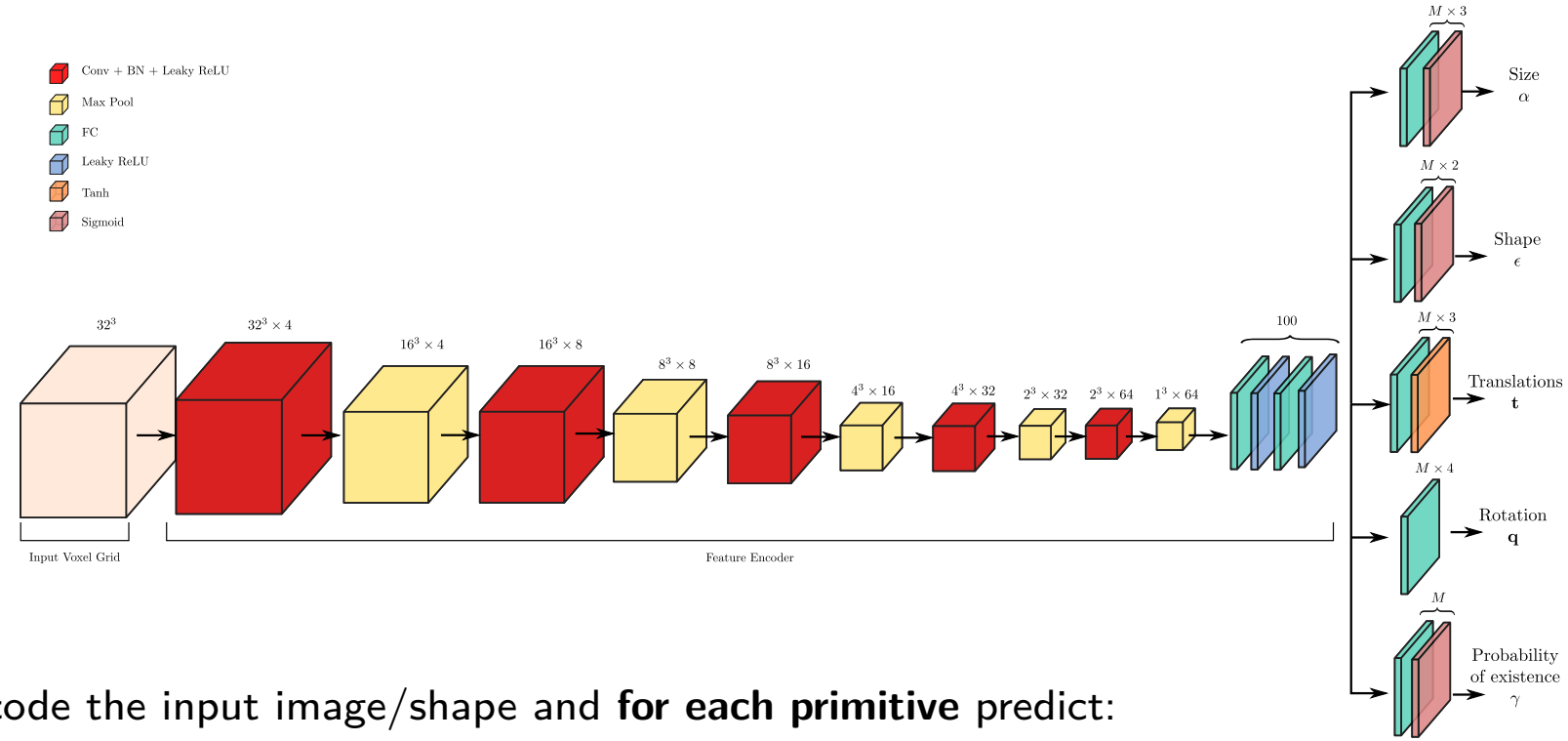
*Everything in nature takes its form from the **sphere**, the **cone** and the **cylinder**.* - Paul Cezanne.



Superquadrics **represent a diverse class of shapes in a single continuous parameter space.**

- Fully described with just 11 parameters
- Their large shape vocabulary allows for **faster** and **smoother fitting** than cuboids

Architecture

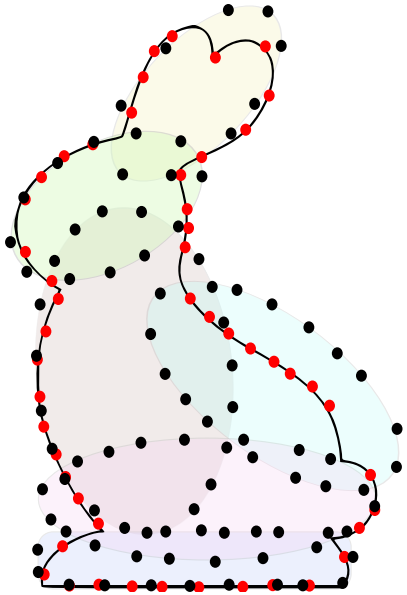


Encode the input image/shape and **for each primitive** predict:

- 11 parameters: 6 pose $(\mathbf{R}, \mathbf{t})$ + 3 scale $(\boldsymbol{\alpha})$ + 2 shape $(\boldsymbol{\epsilon})$
- Probability of existence: $\gamma \in [0, 1]$

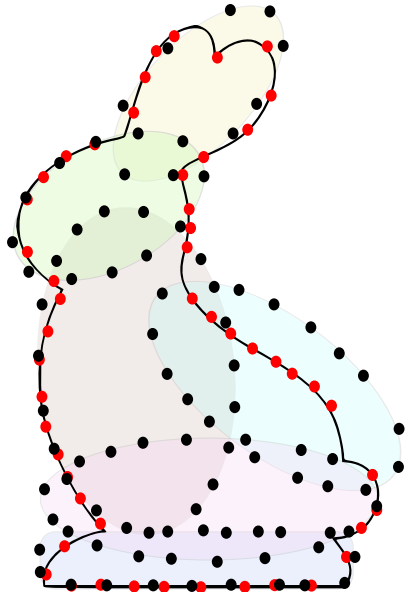
Our **supervision** comes from the **mesh** of the target object that is parametrized by a set of points **uniformly sampled on its surface**.

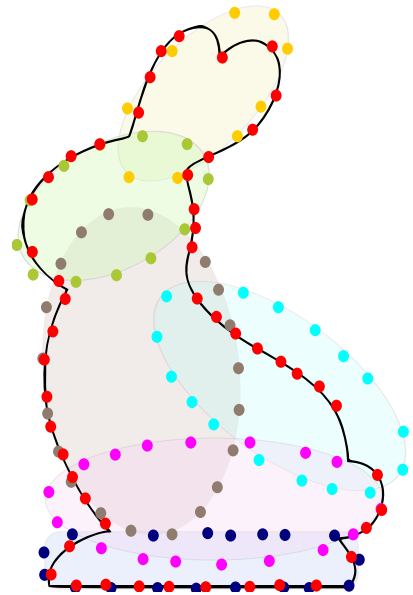
Optimization Objective



- **Pointcloud-to-Primitive Loss**: The surface of the target and the predicted shape should match.

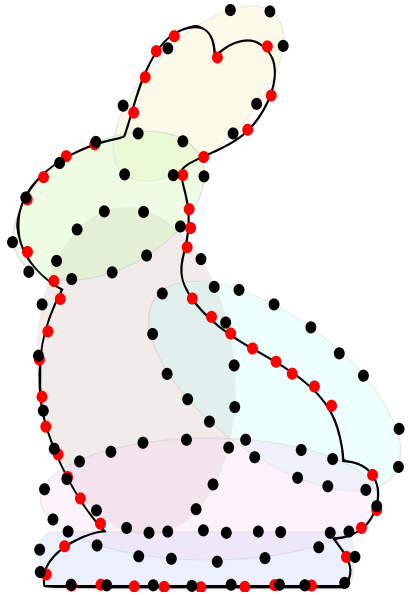
Optimization Objective

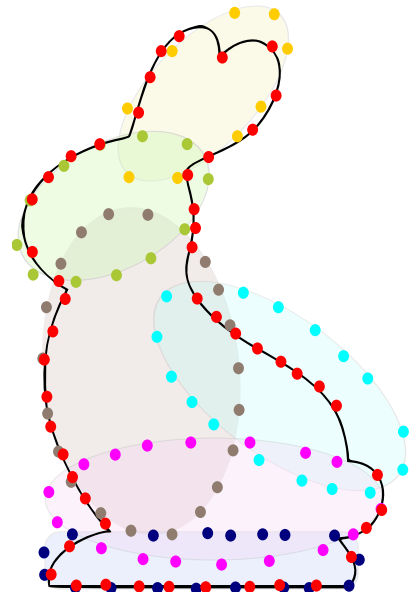




- **Pointcloud-to-Primitive Loss**: The surface of the target and the predicted shape should match.
- **Primitive-to-Pointcloud Loss**: The surface of the target and the predicted shape should match.

Optimization Objective



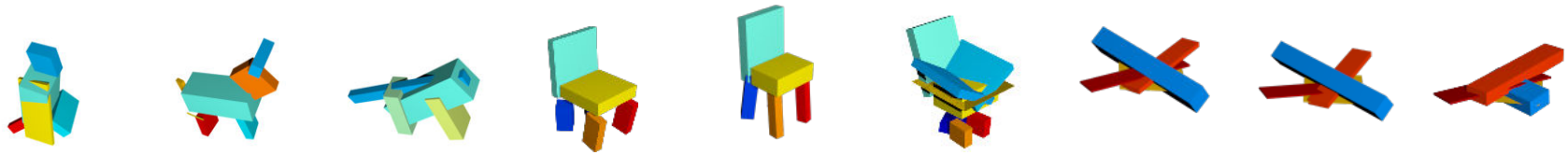


- **Pointcloud-to-Primitive Loss**: The surface of the target and the predicted shape should match.
- **Primitive-to-Pointcloud Loss**: The surface of the target and the predicted shape should match.
- **Parsimony Loss**: Encourage few components with high existense probabilities

How well does it work?

Single View 3D Reconstruction on ShapeNet

Target

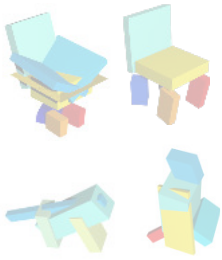


3D Cuboids

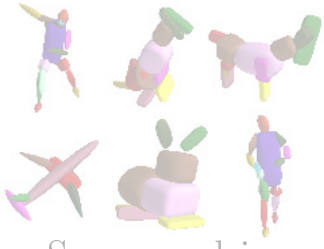
Superquadrics

	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**

Expressive Part-based Representations

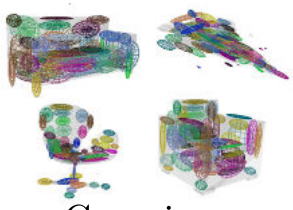


Cuboids
Tulsiani et al. 2017

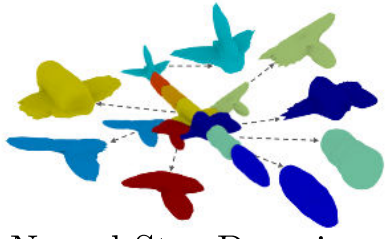


Superquadrics
Paschalidou et al. 2019

Convexes
Deng et al. 2020

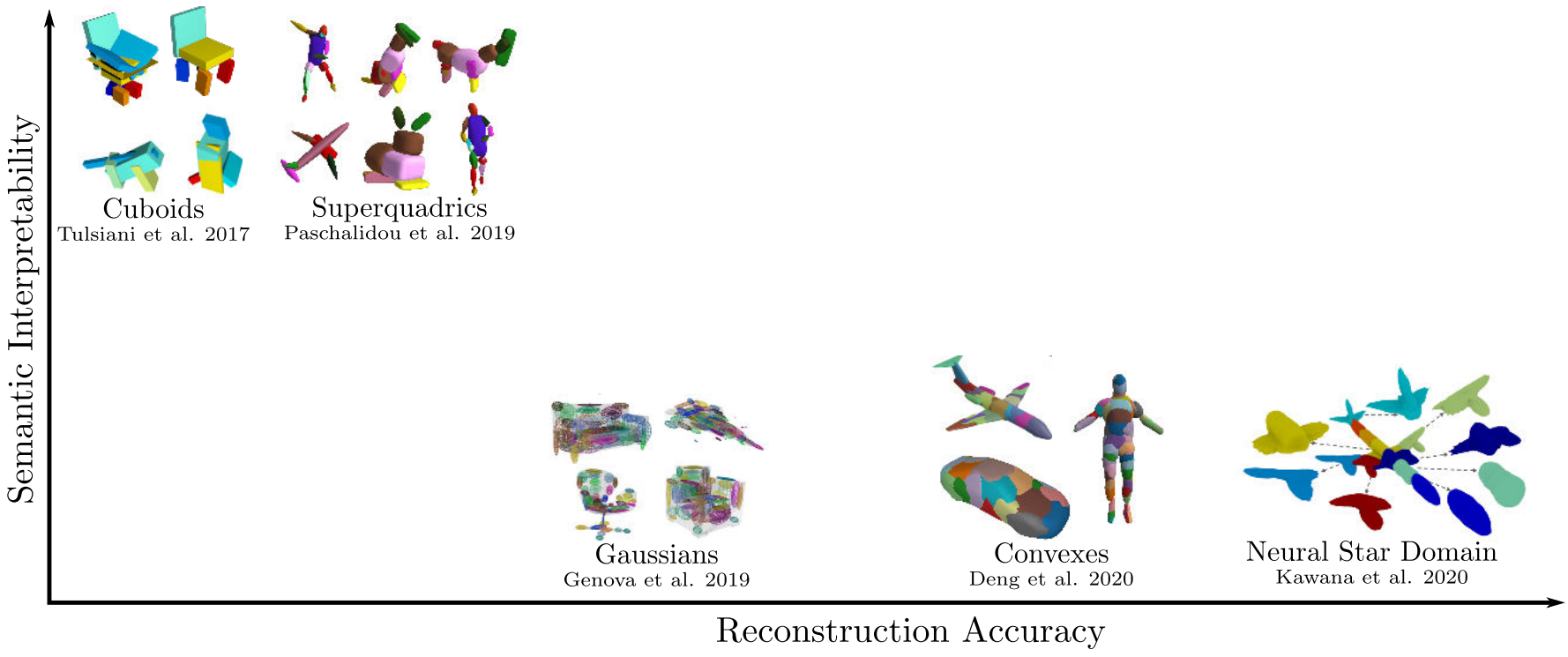


Gaussians
Genova et al. 2019



Neural Star Domain
Kawana et al. 2020

Expressive Part-based Representations



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

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

We introduce **Neural Parts** a novel 3D primitive representation that can yield accurate and semantic reconstructions using an order of magnitude less parts.

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

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

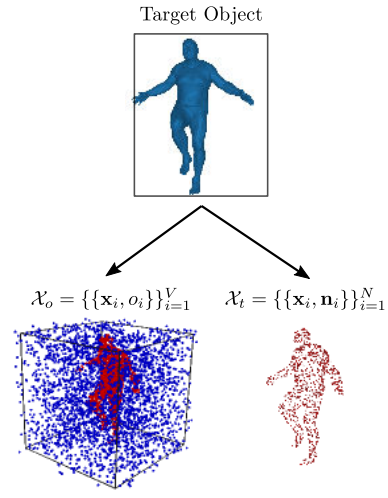
where $\mathbf{x}$ and $\mathbf{y}$ are 3D points in X and Y and $\phi_{\boldsymbol{\theta}} : Y \rightarrow X$, $\phi_{\boldsymbol{\theta}}^{-1} : X \rightarrow Y$ are continuous bijections.

System Overview

Input Image

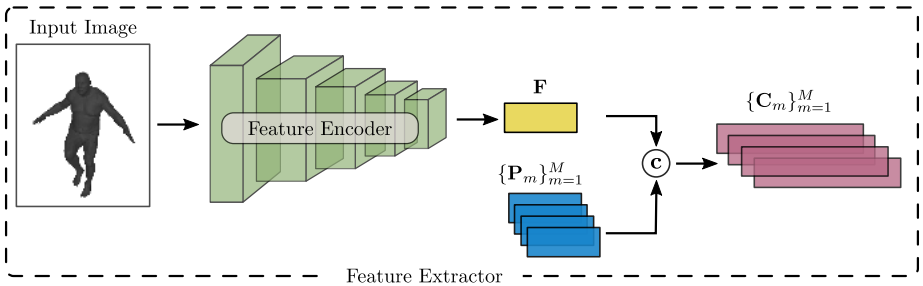
Target Object

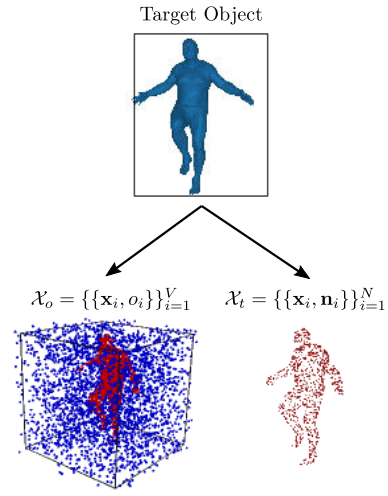
System Overview



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

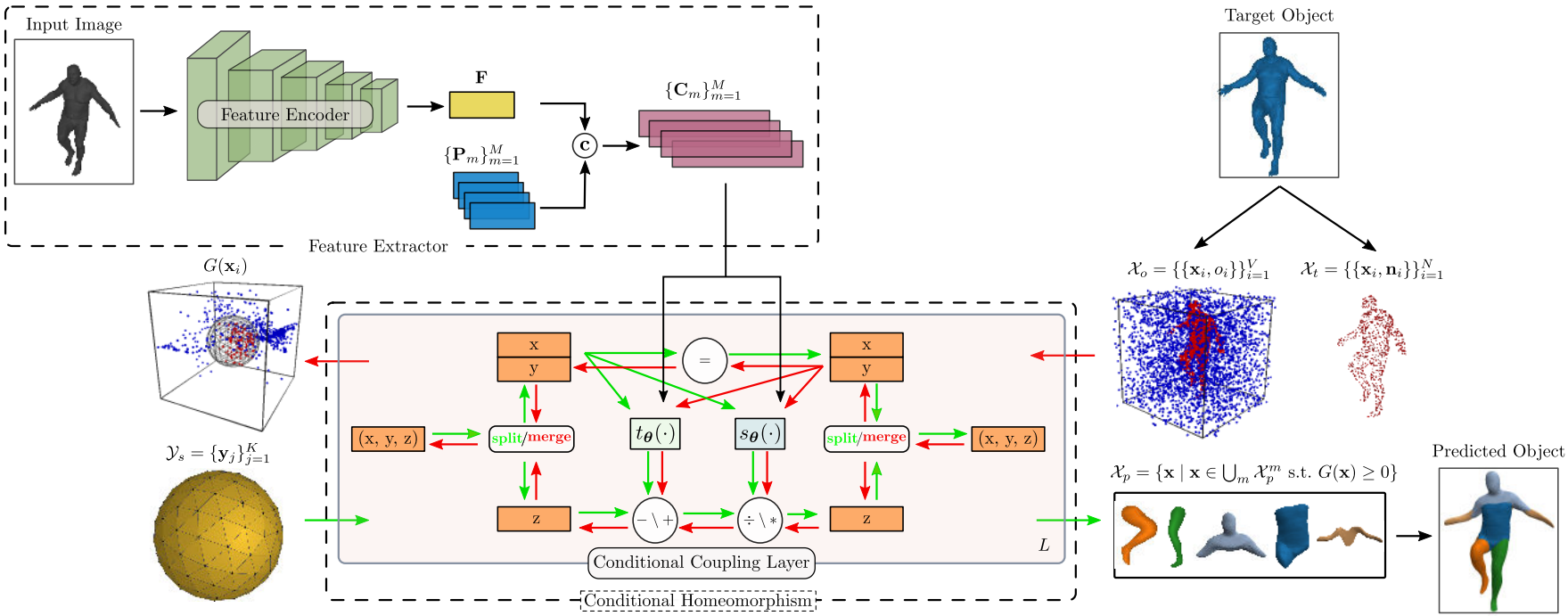
System Overview





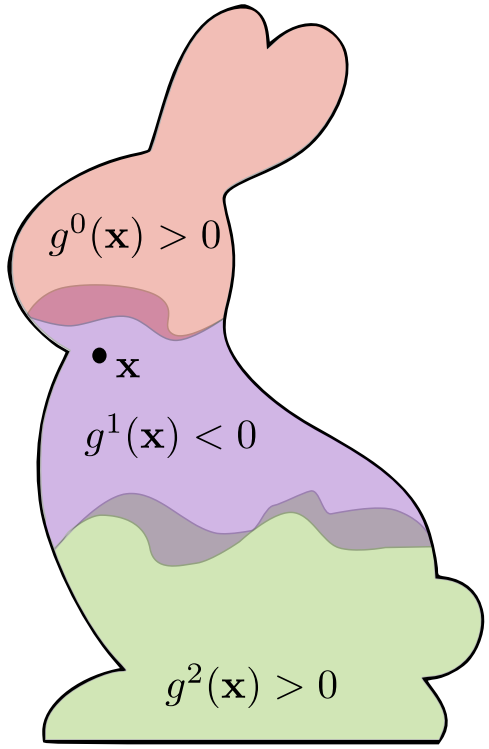
- Our **supervision** comes from a watertight mesh of the target object parametrized as **surface samples** $\mathcal{X}_t$ and a set of **occupancy pairs** $\mathcal{X}_o$.
- The **feature extractor** maps the input image into a **per-primitive shape embedding**.

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.

Implicit and Explicit Representation of Predicted Shape

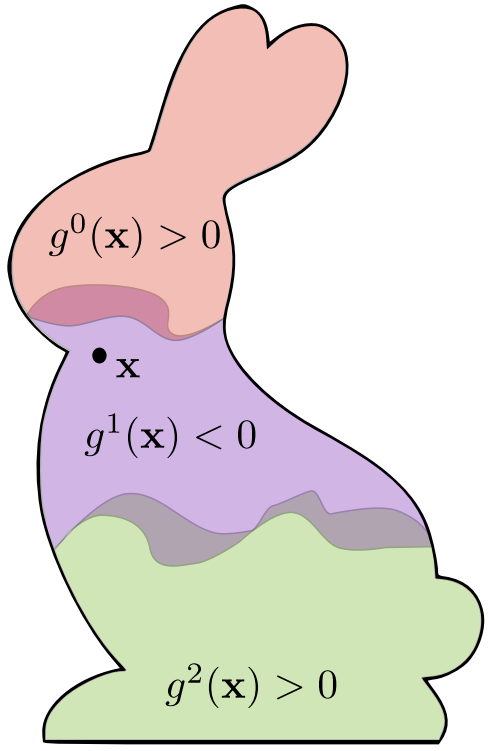


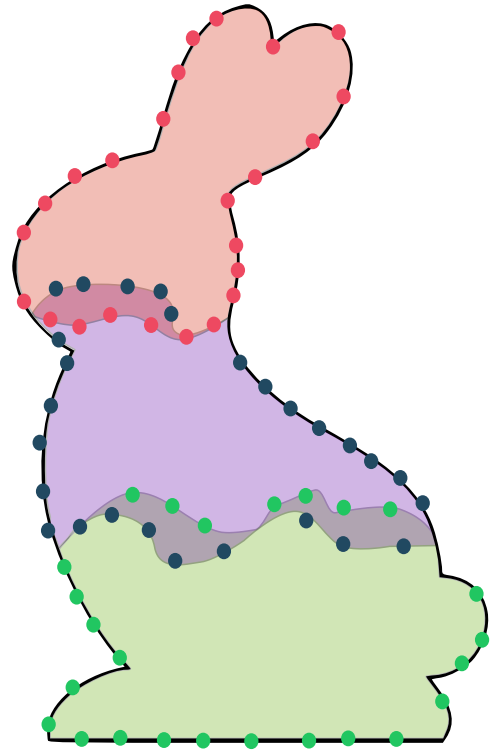
Implicit Representation:

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

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

Implicit and Explicit Representation of Predicted Shape





Implicit Representation:

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

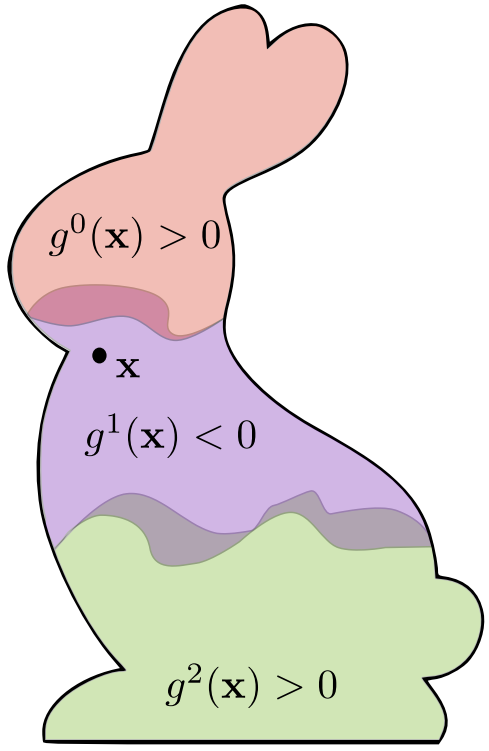
$$g^m(\mathbf{x}) = \left\| \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m) \right\|_2 - r, \ \forall \ \mathbf{x} \in \mathbb{R}^3$$

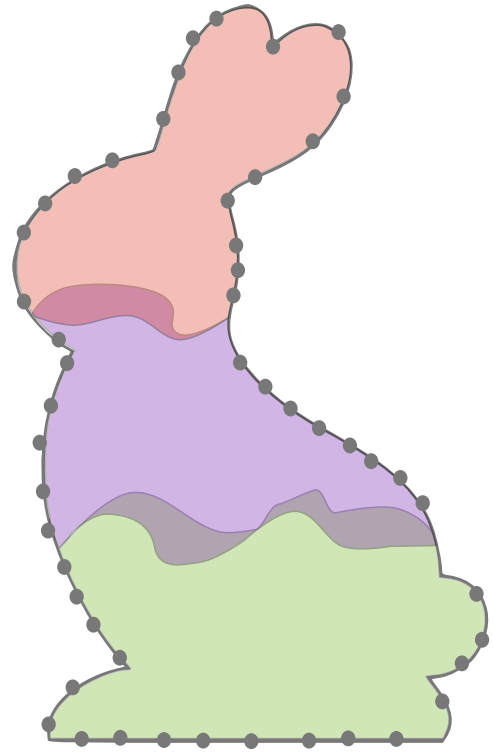
Explicit Representation:

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

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

Implicit and Explicit Representation of Predicted Shape





Implicit Representation:

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

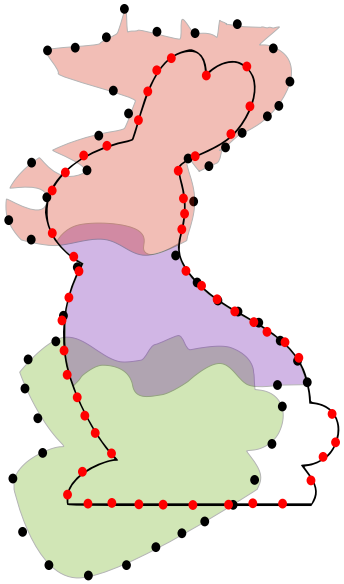
$$g^m(\mathbf{x}) = \left\| \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m) \right\|_2 - r, \ \forall \ \mathbf{x} \in \mathbb{R}^3$$

Explicit Representation:

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

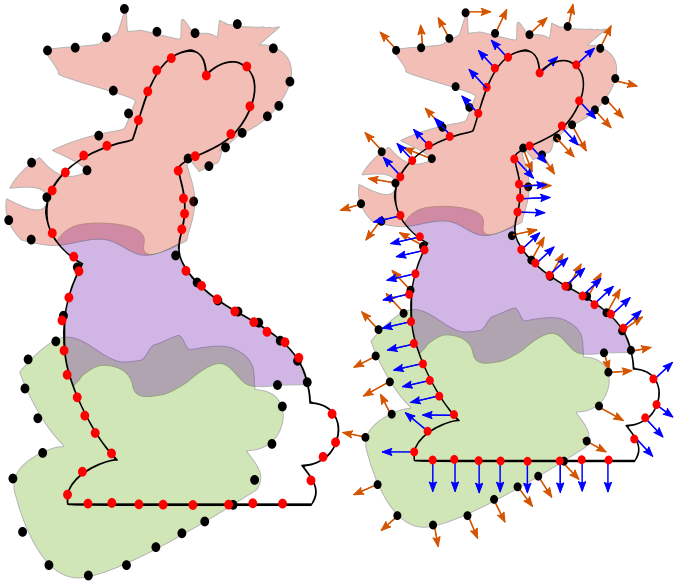
$$\mathcal{X}_p^m = \{\phi_{\boldsymbol{\theta}}(\mathbf{y}_j; \mathbf{C}_m), \ \forall \ \mathbf{y}_j \in \mathcal{Y}_s\}$$

Optimization Objective



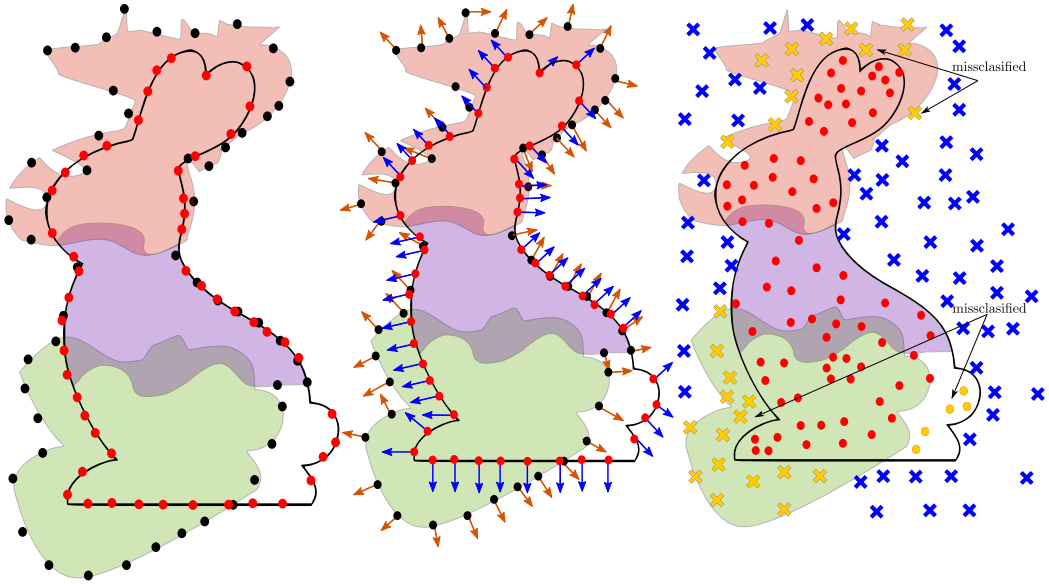
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.

Optimization Objective



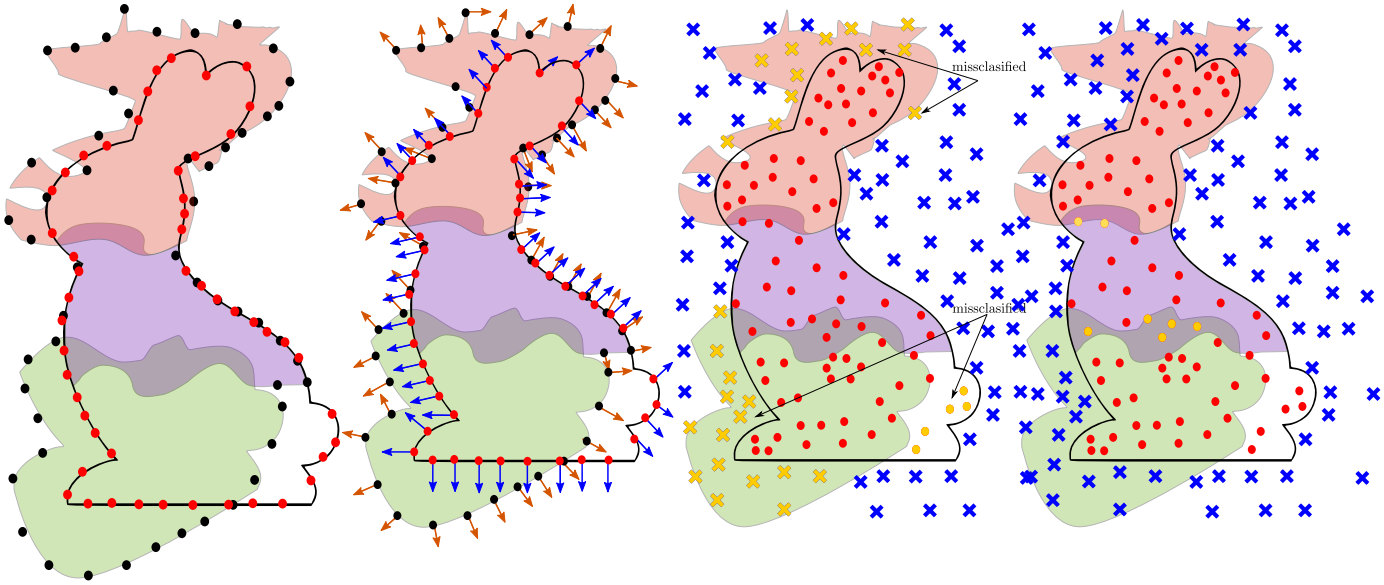
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.

Optimization Objective



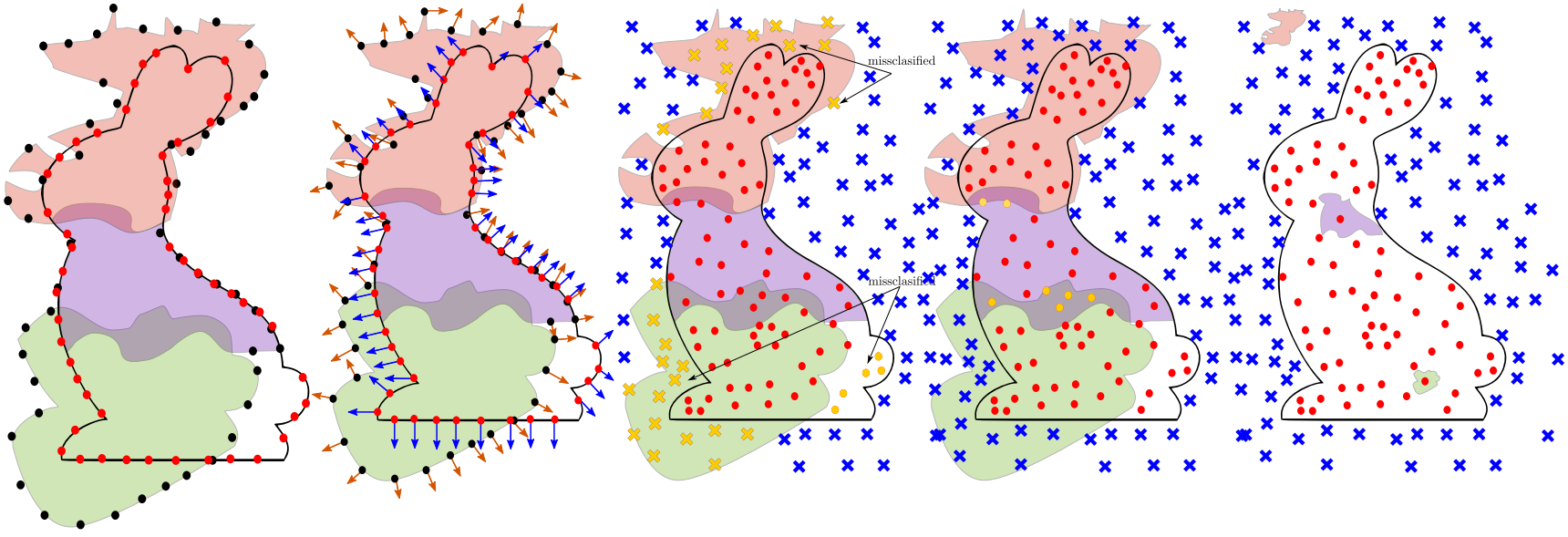
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.

Optimization Objective



- **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.

Optimization Objective

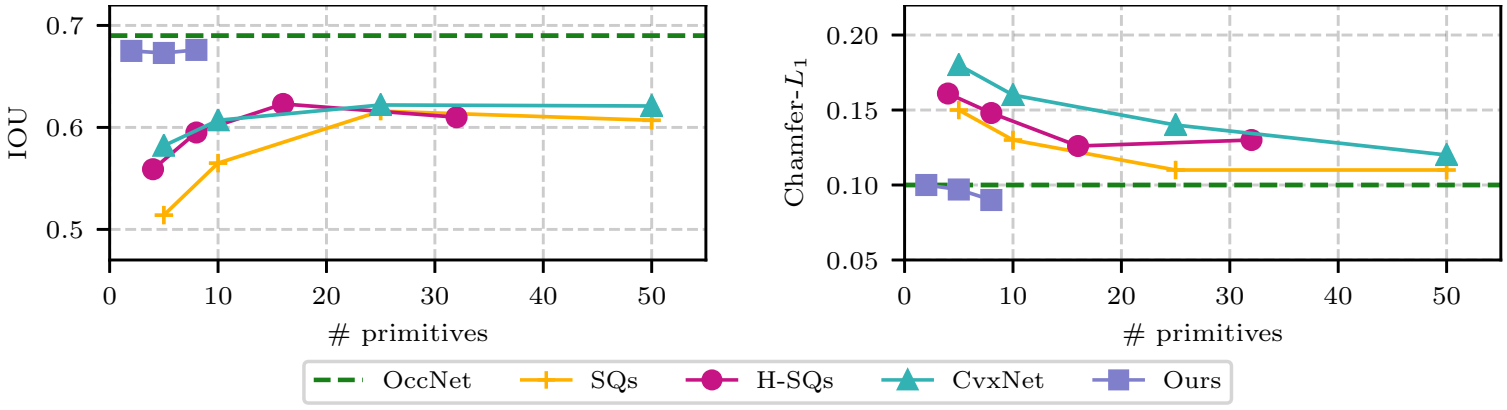


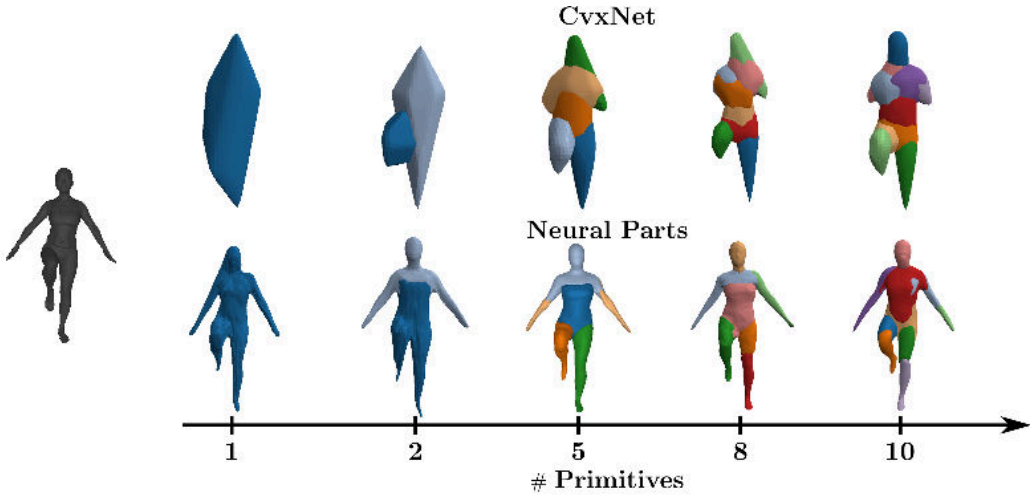
- **Reconstruction Loss**: The surface of the target and the predicted shape should match.
- **Normals Consistency Loss**: The normals of the target and the predicted shape should match.
- **Occupancy Loss**: The volume of the target and the predicted shape should match.
- **Overlapping Loss**: Prevent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



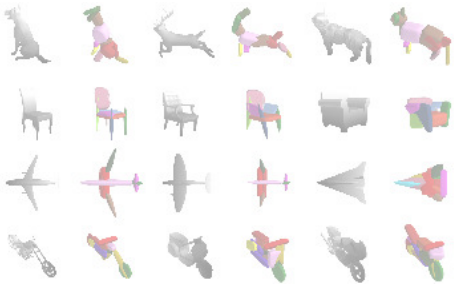


Single-view 3D Reconstruction on Animals

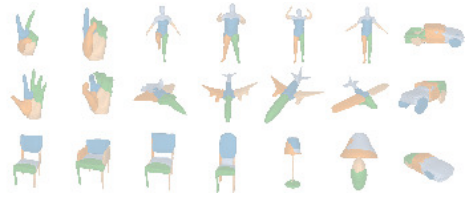
Single-view 3D Reconstruction on ShapeNet

About My Research

Expressive Part-based Representations

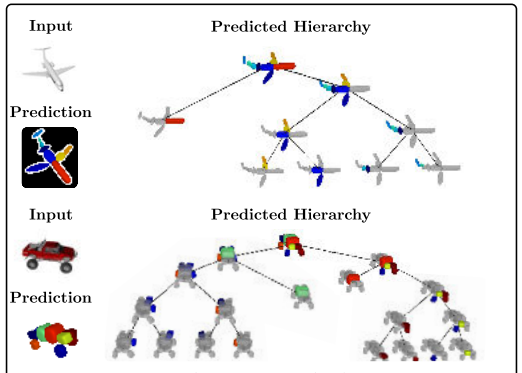


Superquadrics Revisited:
Learning 3D Shape Parsing beyond Cuboids
CVPR 2019



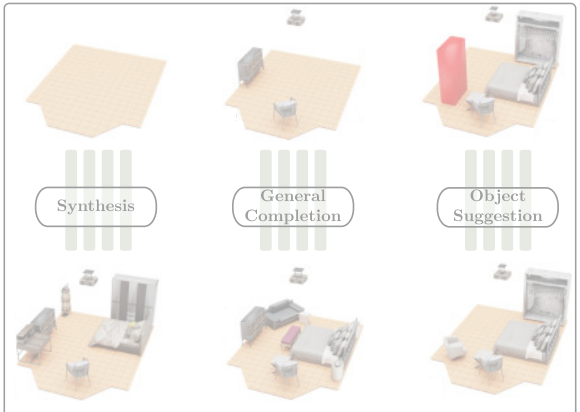
Neural Parts:
Learning Expressive 3D Shape Abstractions
with Invertible Neural Networks
CVPR 2021

Structure-Aware Part-based Representations



Learning Unsupervised
Hierarchical Part Decomposition
of 3D Objects from a Single RGB Image
CVPR 2020

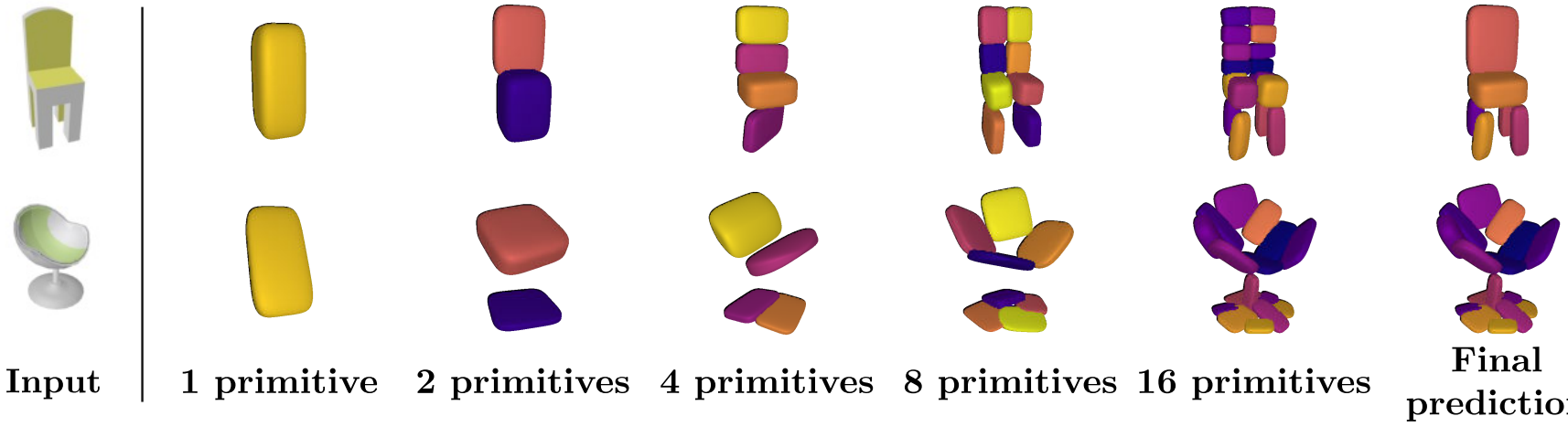
Structure-Aware Object-based Representations for Scene Synthesis



ATISS:
Autoregressive Transformers
for Indoor Scene Synthesis
NeurIPS 2021

Object-Level Understanding → **Scene-Level Understanding**

Structure-aware Representations

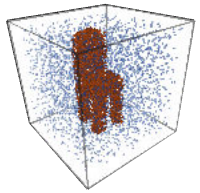


- Represent a 3D shape as a **binary tree of primitives**
- At each depth level, each node is **recursively** split into two until reaching the maximum depth
- Reconstructions from deeper depth levels are more detailed

Learning Hierarchical Part Decomposition of 3D Objects

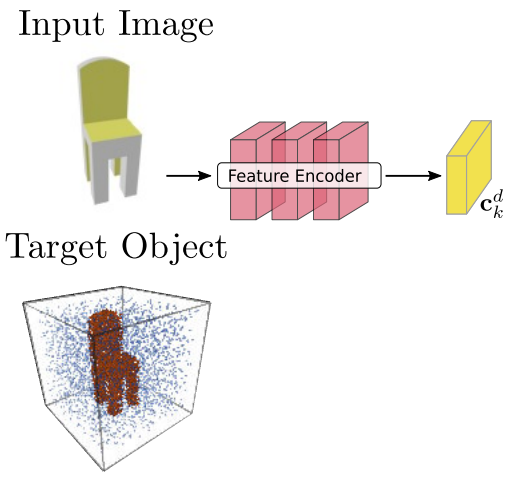
Input Image

Target Object



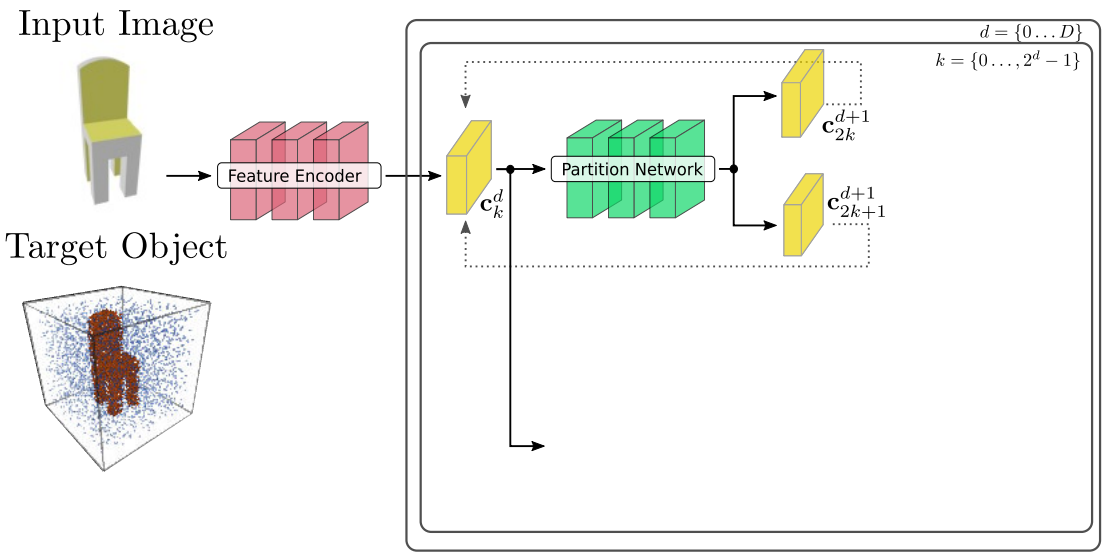
- Our **supervision** comes from a watertight mesh of the target object parametrized as a set of **occupancy pairs** $\mathcal{X}$.

Learning Hierarchical Part Decomposition of 3D Objects



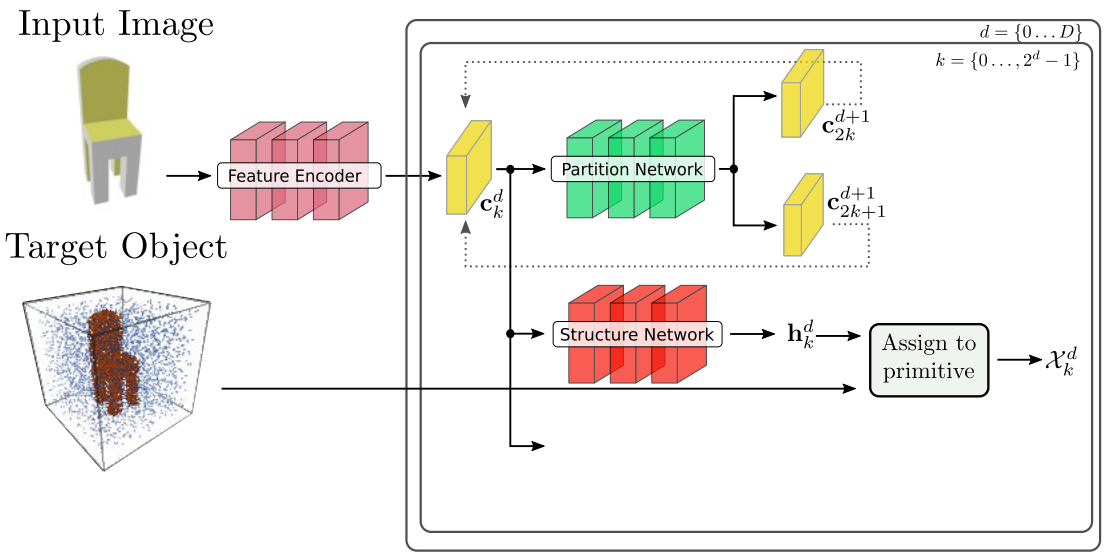
- Our **supervision** comes from a watertight mesh of the target object parametrized as a set of **occupancy pairs** $\mathcal{X}$.

Learning Hierarchical Part Decomposition of 3D Objects



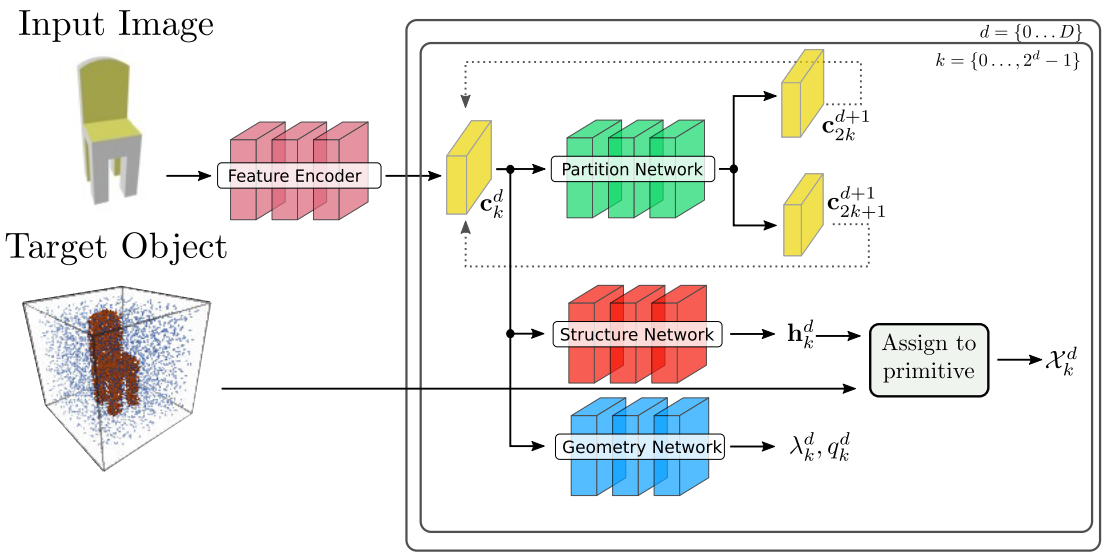
- Our **supervision** comes from a watertight mesh of the target object parametrized as a set of **occupancy pairs** $\mathcal{X}$.
- The **partition network** recursively splits the shape embedding into representations of parts.

Learning Hierarchical Part Decomposition of 3D Objects



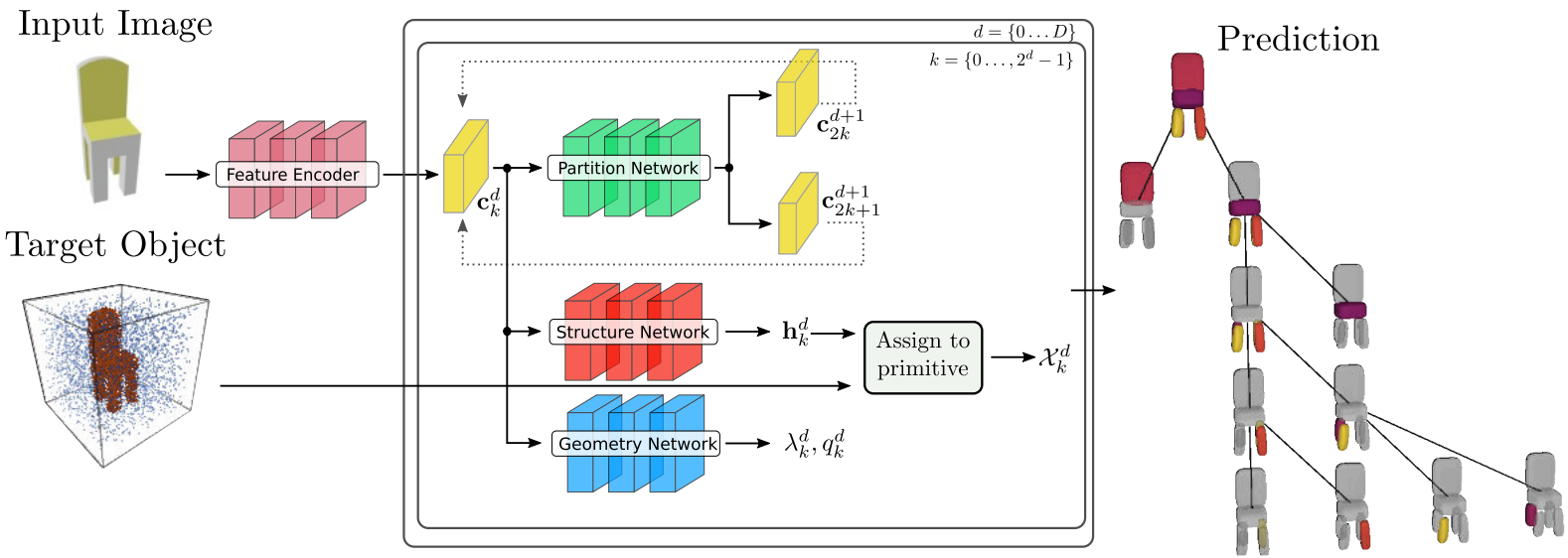
- Our **supervision** comes from a watertight mesh of the target object parametrized as a set of **occupancy pairs** $\mathcal{X}$.
- The **partition network** recursively splits the shape embedding into representations of parts.
- The **structure network** associates **each primitive with the part of the object it should represent**.

Learning Hierarchical Part Decomposition of 3D Objects



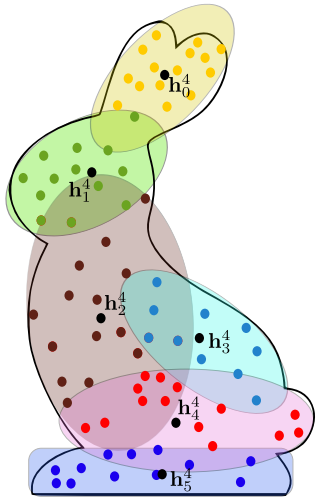
- Our **supervision** comes from a watertight mesh of the target object parametrized as a set of **occupancy pairs** $\mathcal{X}$.
- The **partition network** recursively splits the shape embedding into representations of parts.
- The **structure network** associates **each primitive with the part of the object it should represent**.
- The **geometry network** regresses the **shape, size, pose and reconstruction quality** of each primitive.

Learning Hierarchical Part Decomposition of 3D Objects



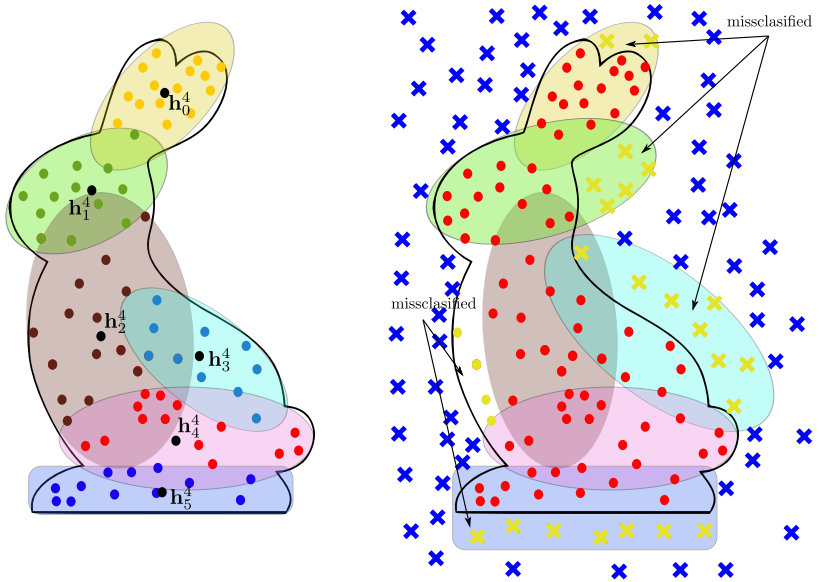
- Our **supervision** comes from a watertight mesh of the target object parametrized as a set of **occupancy pairs** $\mathcal{X}$.
- The **partition network** recursively splits the shape embedding into representations of parts.
- The **structure network** associates **each primitive with the part of the object it should represent**.
- The **geometry network** regresses the **shape, size, pose and reconstruction quality** of each primitive.

Optimization Objective



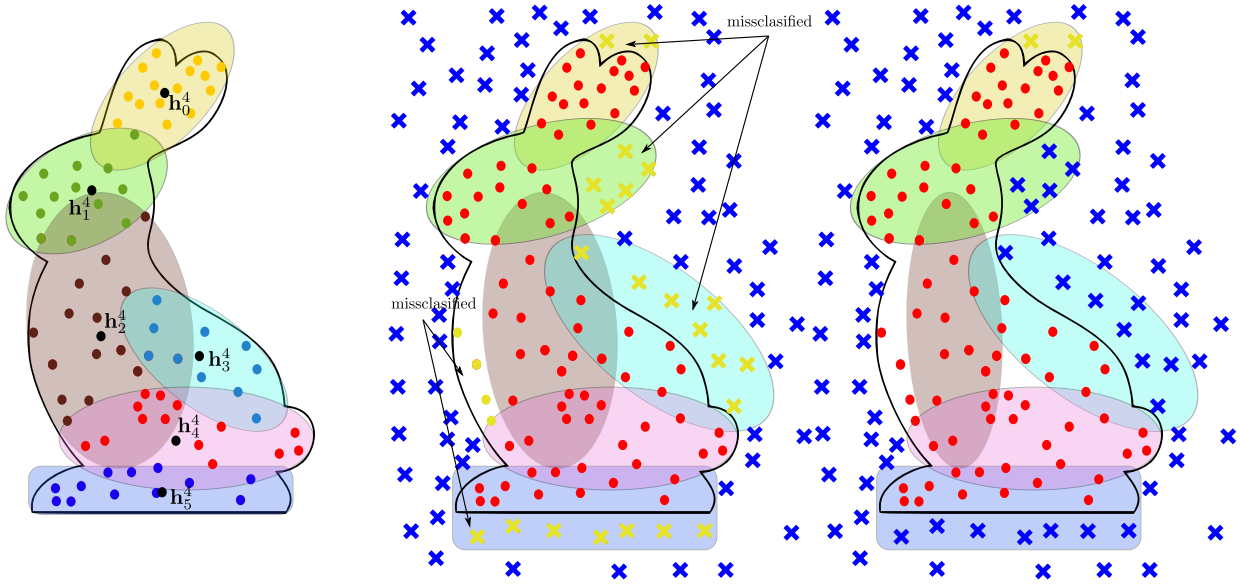
- **Structure Loss**: Distributes primitives on the target object.

Optimization Objective



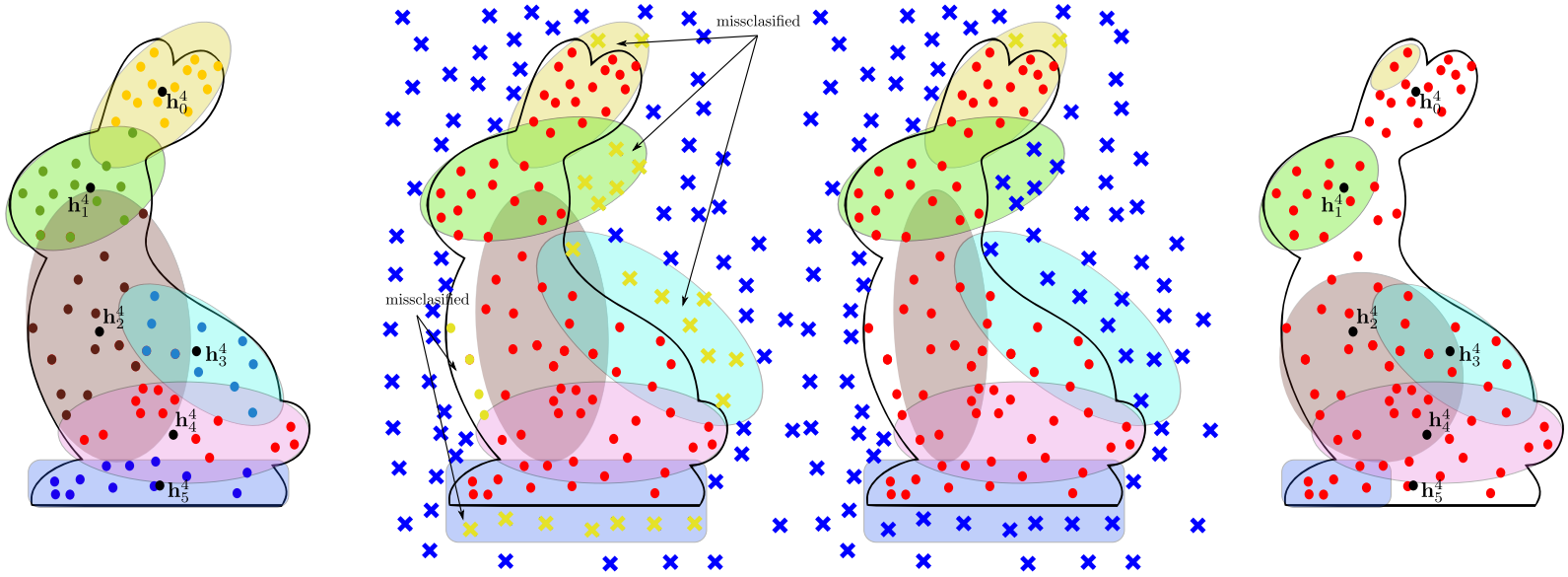
- **Structure Loss**: Distributes primitives on the target object.
- **Reconstruction Loss**: The volume of the target and the predicted shape should match.

Optimization Objective



- **Structure Loss**: Distributes primitives on the target object.
- **Reconstruction Loss**: The volume of the target and the predicted shape should match.
- **Compatibility Loss**: Maximize the reconstruction quality of each primitive.

Optimization Objective



- **Structure Loss**: Distributes primitives on the target object.
- **Reconstruction Loss**: The volume of the target and the predicted shape should match.
- **Compatibility Loss**: Maximize the reconstruction quality of each primitive.
- **Proximity Loss**: Prevent vanishing gradients

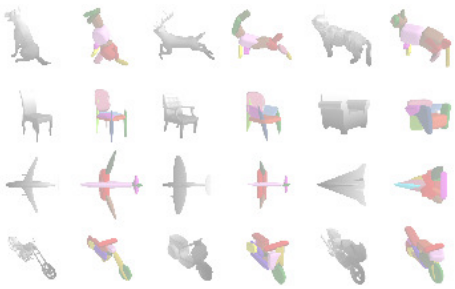
How well does it work?

Expressive Shape Abstractions

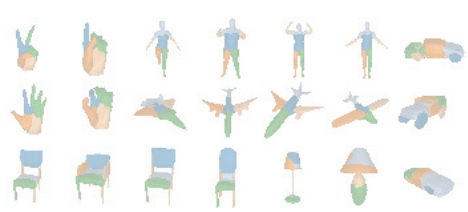
Semantic Interpretation of Learned Hierarchy

About My Research

Expressive Part-based Representations

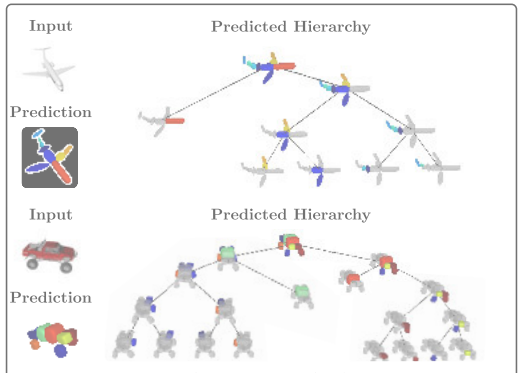


Superquadrics Revisited:
Learning 3D Shape Parsing beyond Cuboids
CVPR 2019



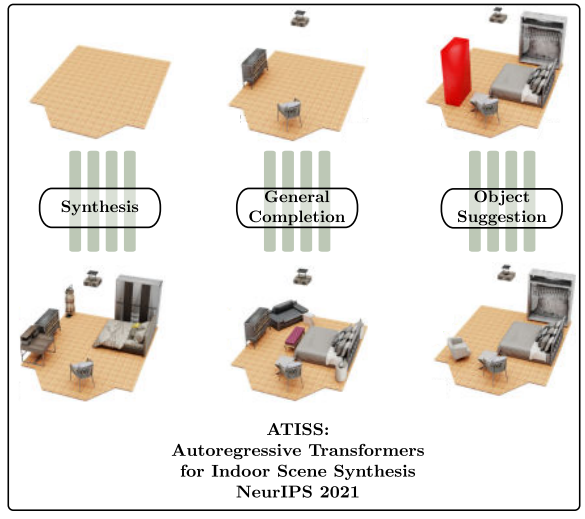
Neural Parts:
Learning Expressive 3D Shape Abstractions
with Invertible Neural Networks
CVPR 2021

Structure-Aware Part-based Representations



Learning Unsupervised
Hierarchical Part Decomposition
of 3D Objects from a Single RGB Image
CVPR 2020

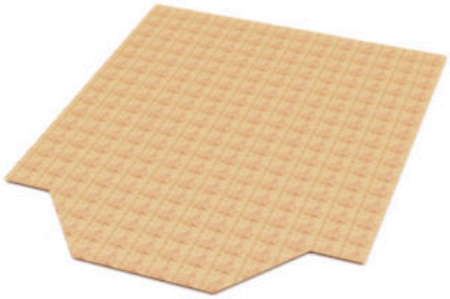
Structure-Aware Object-based Representations for Scene Synthesis

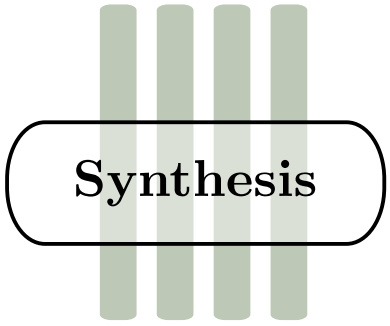


ATISS:
Autoregressive Transformers
for Indoor Scene Synthesis
NeurIPS 2021

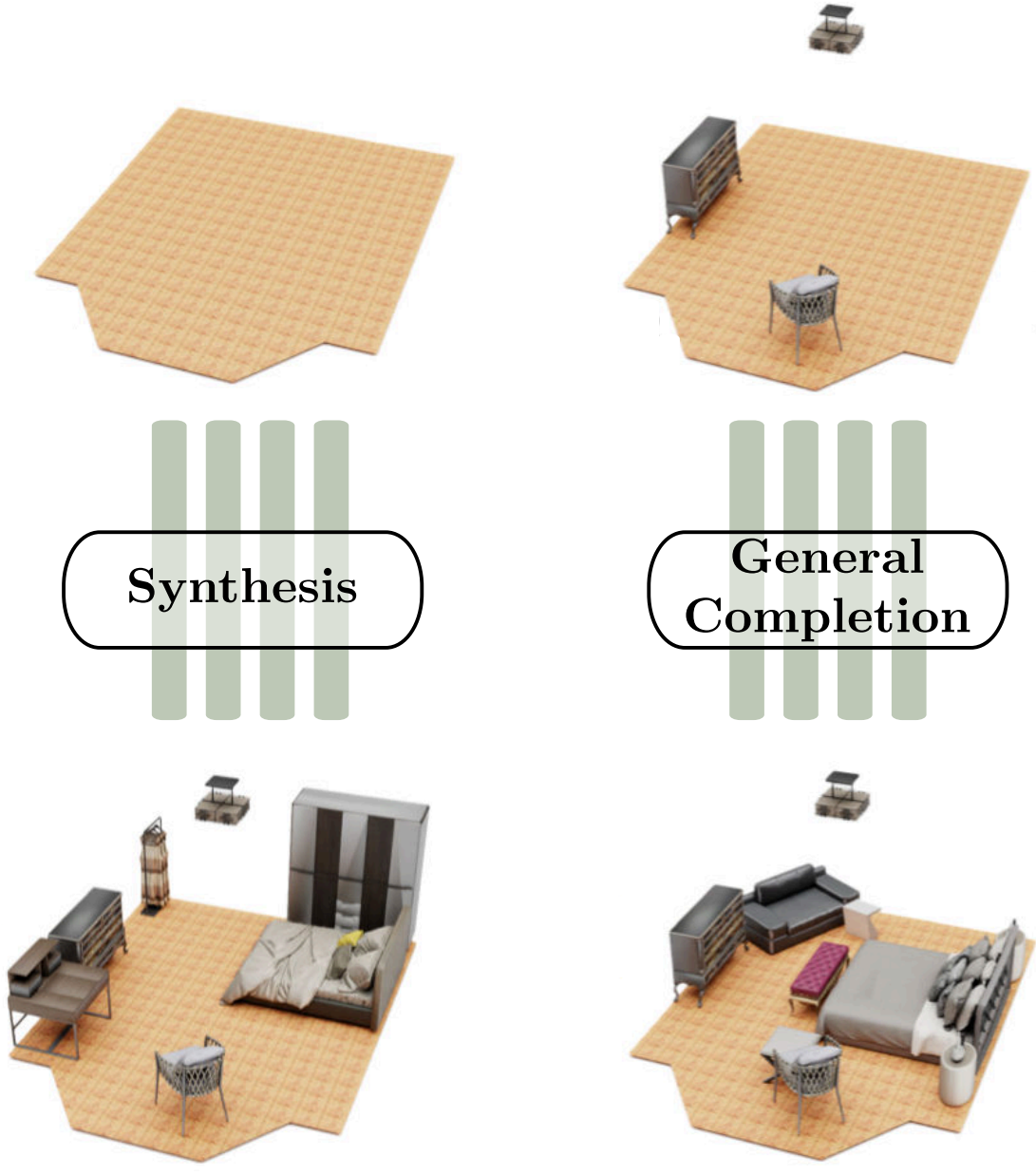
Object-Level Understanding → **Scene-Level Understanding**

Scene Synthesis with Versalite User Input

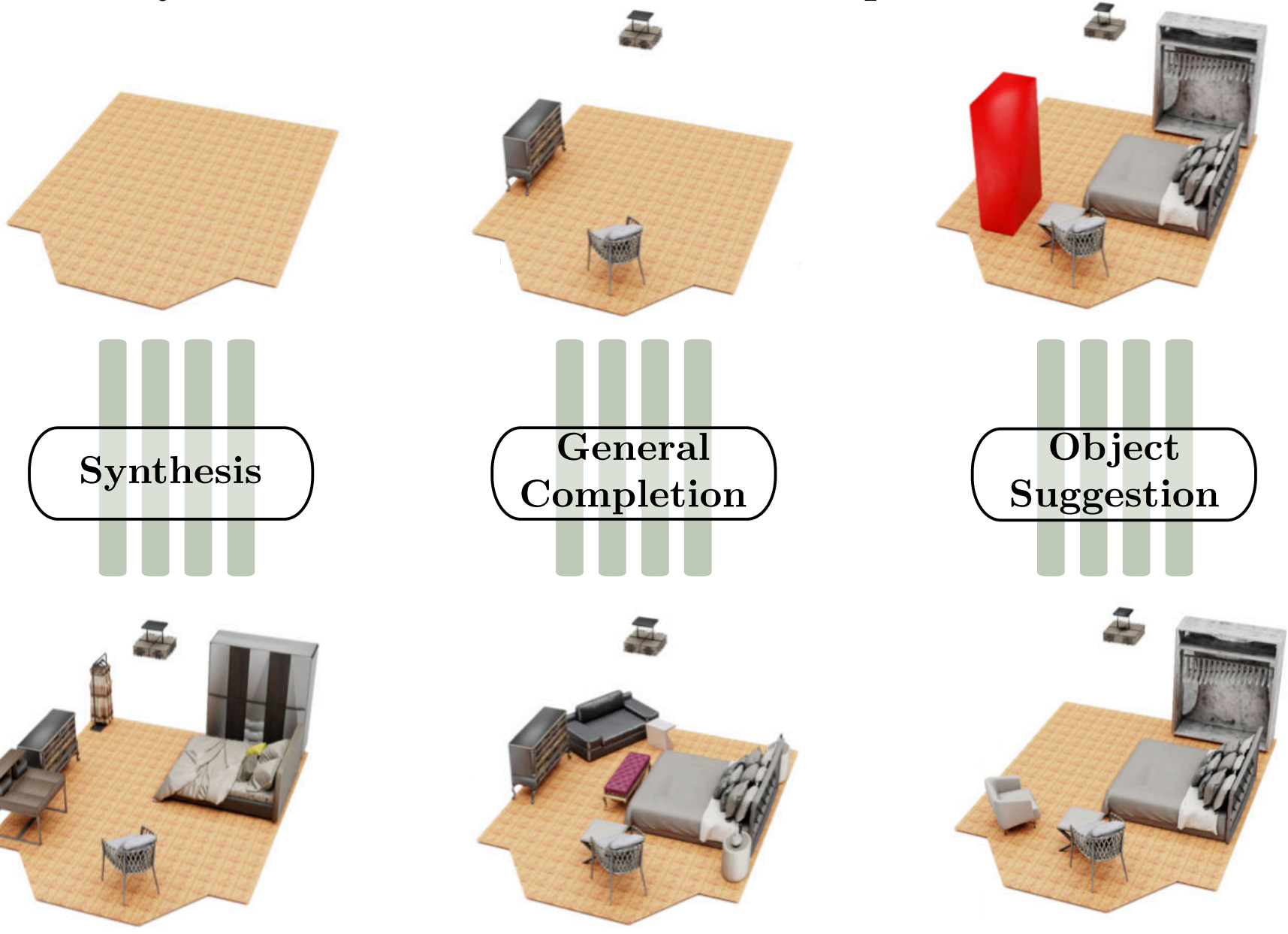




Scene Synthesis with Versalite User Input



Scene Synthesis with Versalite User Input



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

FastSynth, Ritchie et al. CVPR 2019

SceneFormer, Wang et al. ARXIV 2020

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

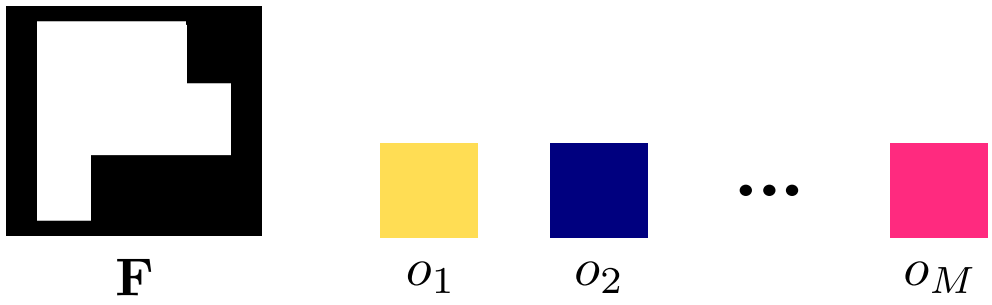
FastSynth, Ritchie et al. CVPR 2019

SceneFormer, Wang et al. ARXIV 2020

We pose scene synthesis as an **unordered set generation problem**.

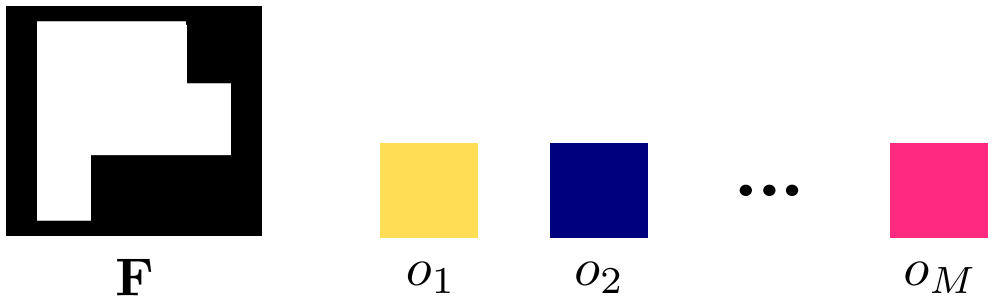
Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape $\mathbf{F}$**.



Scene Parametrization

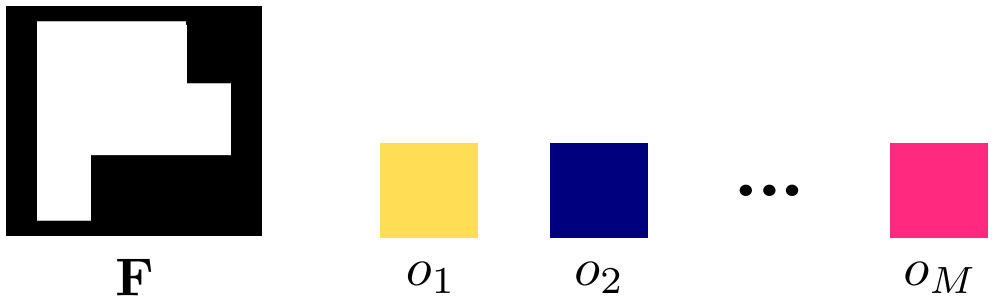
A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



Each object $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j\}$ is modelled with four random variables that describe their **category, size, orientation and location**.

Scene Parametrization

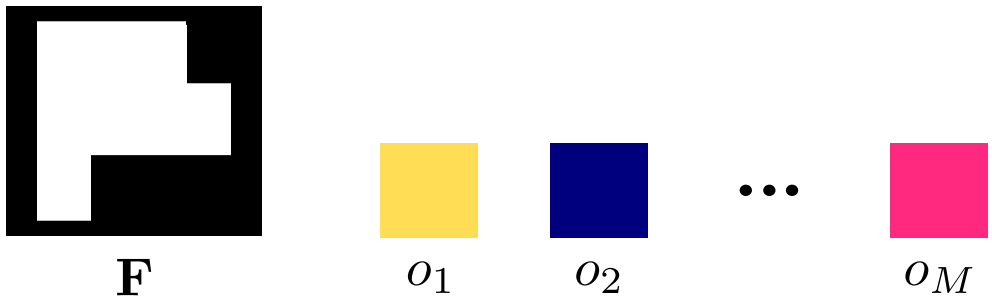
A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



Each object $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j\}$ is modelled with four random variables that describe their **category, size, orientation and location**.

Scene Parametrization

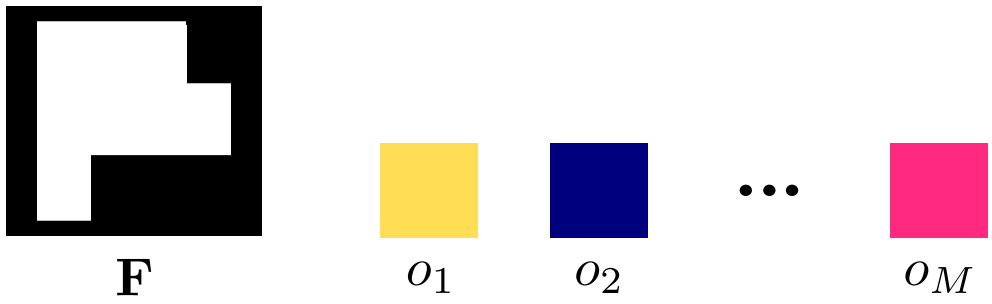
A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



Each object $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j\}$ is modelled with four random variables that describe their **category, size, orientation and location**.

Scene Parametrization

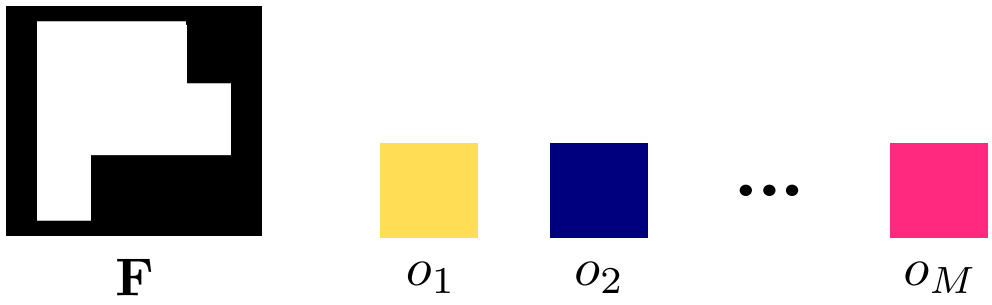
A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape $\mathbf{F}$**.



Each object $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j\}$ is modelled with four random variables that describe their **category, size, orientation and location**.

Scene Parametrization

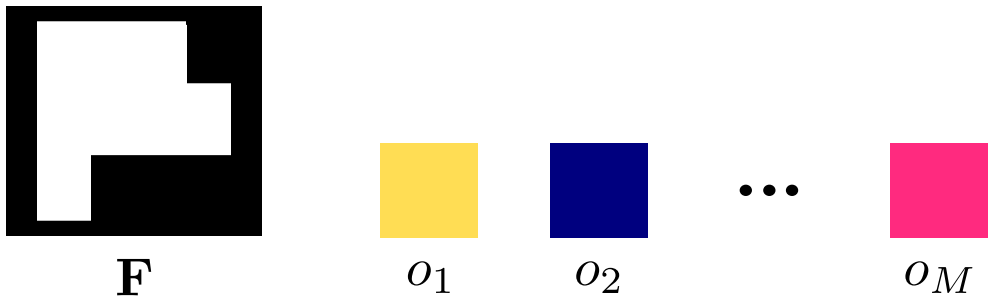
A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



Each object $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j\}$ is modelled with four random variables that describe their **category, size, orientation and location**.

Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.

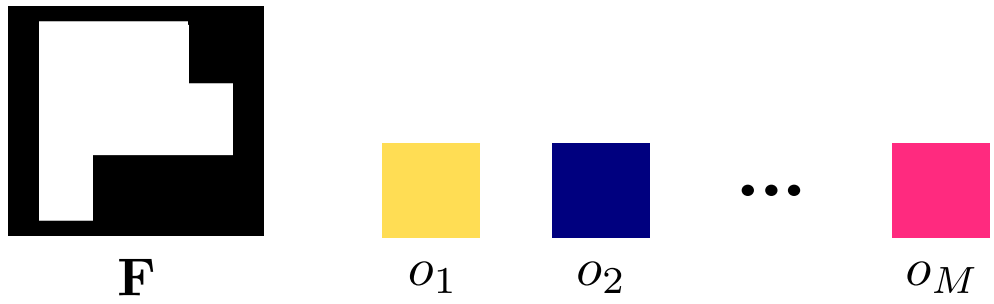


Each object $o_j = \{\mathbf{c}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j\}$ is modelled with four random variables that describe their **category, size, orientation and location**.

$$\underbrace{p_\theta(o_j \mid o_{<j}, \mathbf{F})}_{\text{Probability of generating j-th object}} = p_\theta(\mathbf{c}_j | o_{<j}, \mathbf{F}) p_\theta(\mathbf{t}_j | \mathbf{c}_j, o_{<j}, \mathbf{F}) p_\theta(\mathbf{r}_j | \mathbf{c}_j, \mathbf{t}_j, o_{<j}, \mathbf{F}) p_\theta(\mathbf{s}_j | \mathbf{c}_j, \mathbf{t}_j, \mathbf{r}_j, o_{<j}, \mathbf{F})$$

Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



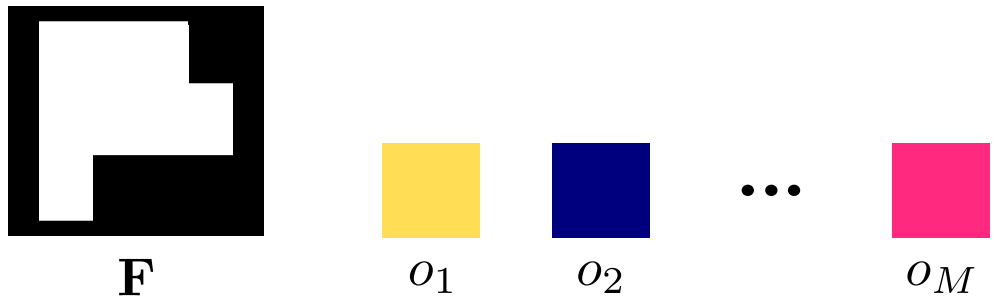
The **likelihood** of generating a scene **with any order** is:

$$\underbrace{p_\theta(\mathcal{O}|\mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with any order}}} = \sum_{\hat{\mathcal{O}} \in \pi(\mathcal{O})} \underbrace{\prod_{j \in \hat{\mathcal{O}}} p_\theta(o_j \mid o_{<j}, \mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with order } \hat{\mathcal{O}}}}$$

where $\pi(\mathcal{O})$ is a permutation function that computes the set of permutations of all objects $\mathcal{O}$ in the scene.

Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



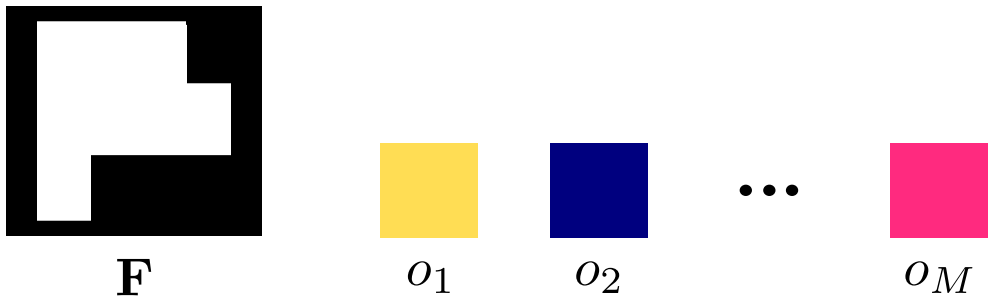
The **likelihood** of generating a scene **with any order** is:

$$\underbrace{p_\theta(\mathcal{O}|\mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with any order}}} = \sum_{\hat{\mathcal{O}} \in \pi(\mathcal{O})} \underbrace{\prod_{j \in \hat{\mathcal{O}}} p_\theta(o_j \mid o_{<j}, \mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with order } \hat{\mathcal{O}}}}$$

where $\pi(\mathcal{O})$ is a permutation function that computes the set of permutations of all objects $\mathcal{O}$ in the scene.

Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.

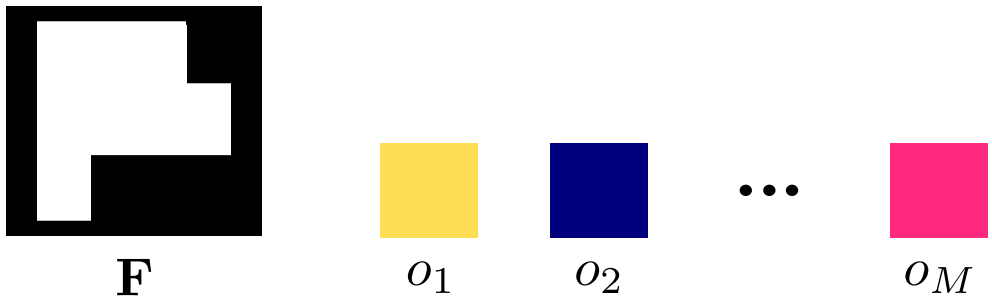


The **likelihood** of generating a scene **with any order** is:

$$\underbrace{p_\theta(\mathcal{O}|\mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with any order}}} = \sum_{\hat{\mathcal{O}} \in \pi(\mathcal{O})} \underbrace{\prod_{j \in \hat{\mathcal{O}}} p_\theta(o_j \mid o_{<j}, \mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with order } \hat{\mathcal{O}}}}$$

Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.



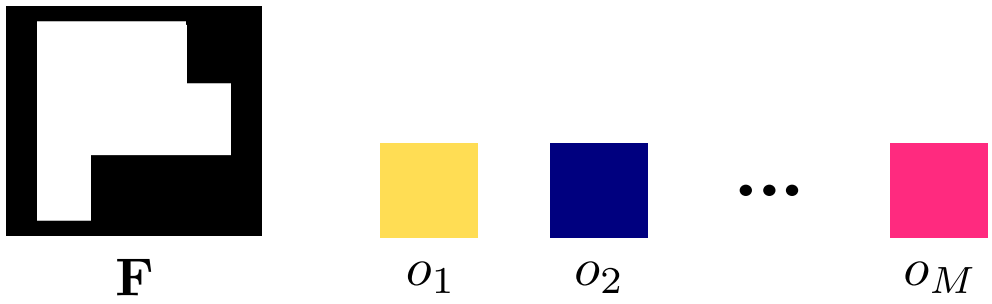
The **likelihood** of generating a scene **with all orders** is:

$$\underbrace{\hat{p}_\theta(\mathcal{O}|\mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with all orders}}} = \prod_{\hat{\mathcal{O}} \in \pi(\mathcal{O})} \underbrace{\prod_{j \in \hat{\mathcal{O}}} p_\theta(o_j \mid o_{<j}, \mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with order } \hat{\mathcal{O}}}}$$

ATISS is trained to **maximize the log-likelihood of all possible permutations of object arrangements** in a collection of scenes.

Scene Parametrization

A scene comprises an **unordered set of M objects** $\mathcal{O} = \{o_j\}_{j=1}^{M}$ and its **floor shape** $\mathbf{F}$.

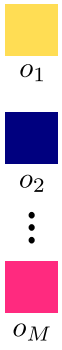


The **log-likelihood** of generating a scene **with all orders** is:

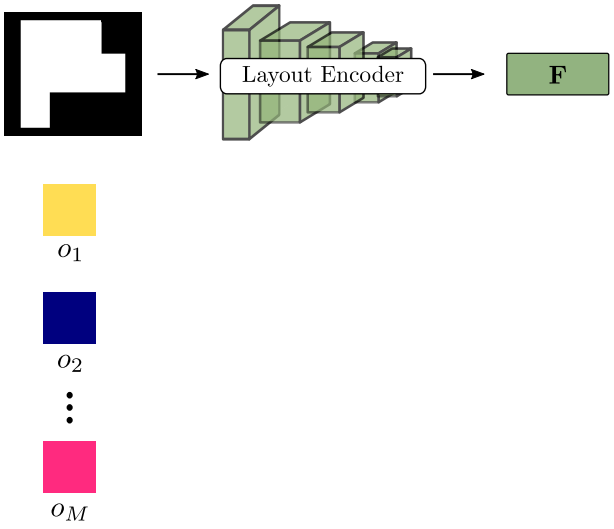
$$\underbrace{\log \hat{p}_\theta(\mathcal{O}|\mathbf{F})}_{\substack{\text{Log-likelihood of generating } \mathcal{O} \\ \text{with all orders}}} = \sum_{\hat{\mathcal{O}} \in \pi(\mathcal{O})} \underbrace{\sum_{j \in \hat{\mathcal{O}}} \log p_\theta(o_j \mid o_{<j}, \mathbf{F})}_{\substack{\text{Probability of generating } \mathcal{O} \\ \text{with order } \hat{\mathcal{O}}}}$$

ATISS is trained to **maximize the log-likelihood of all possible permutations of object arrangements** in a collection of scenes.

Scene Generation

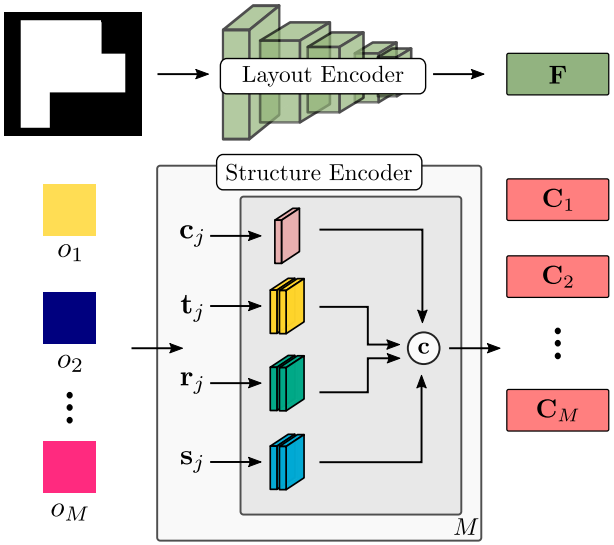


Scene Generation



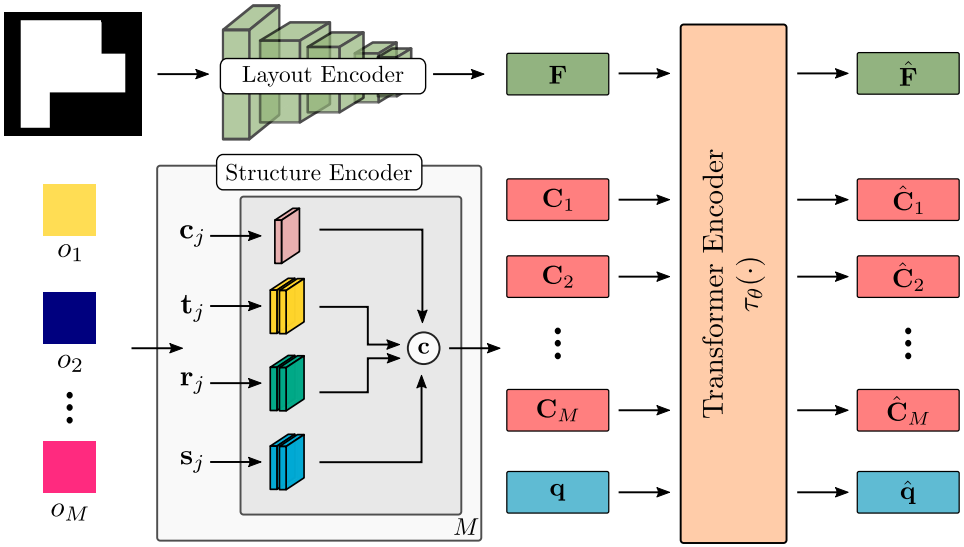
- **Layout encoder**: Computes a global feature representation for the floor.

Scene Generation



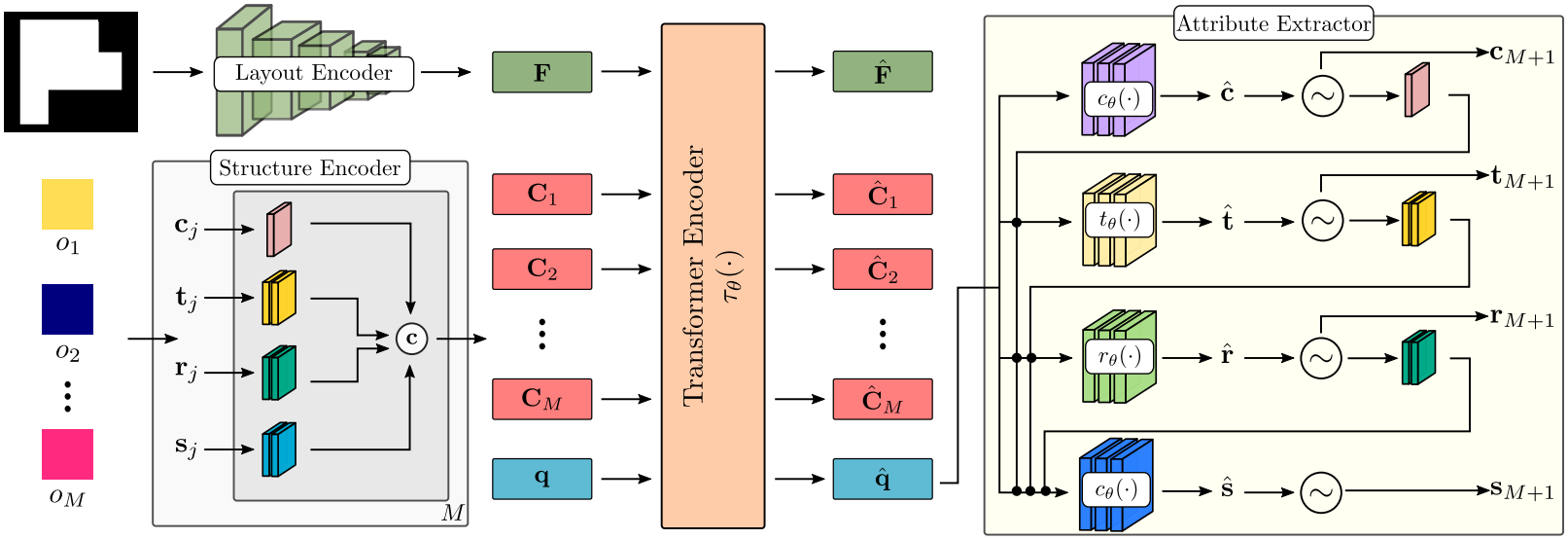
- **Layout encoder**: Computes a global feature representation for the floor.
- **Structure encoder**: Maps the j-th object to a per-object context embedding $\mathbf{C}_j$.

Scene Generation



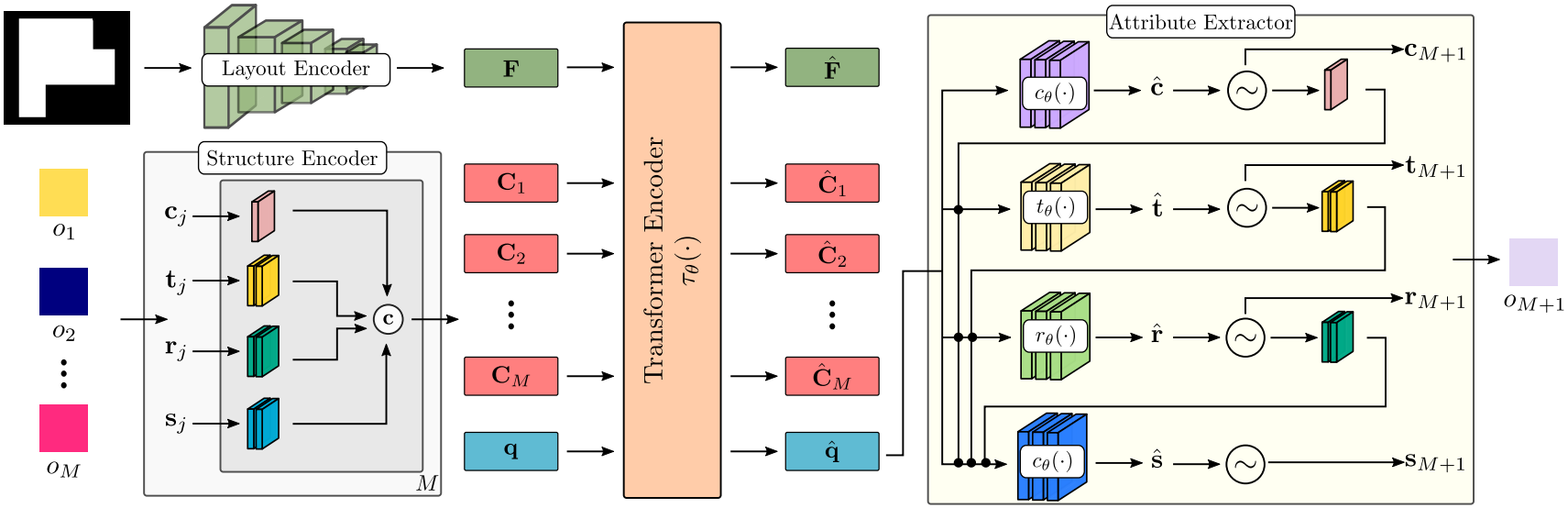
- **Layout encoder**: Computes a global feature representation for the floor.
- **Structure encoder**: Maps the j-th object to a per-object context embedding $\mathbf{C}_j$.
- **Transformer encoder**: Takes $\mathbf{F}, \{\mathbf{C}_j\}_{j=1}^{M}, \mathbf{q}$ and predicts the features $\hat{\mathbf{q}}$ of the next object to be added in the scene.

Scene Generation



- **Layout encoder**: Computes a global feature representation for the floor.
- **Structure encoder**: Maps the j-th object to a per-object context embedding $\mathbf{C}_j$.
- **Transformer encoder**: Takes $\mathbf{F}, \{\mathbf{C}_j\}_{j=1}^{M}, \mathbf{q}$ and predicts the features $\hat{\mathbf{q}}$ of the next object to be added in the scene.
- **Attribute extractor**: Predicts the object attributes of the next object.

Scene Generation

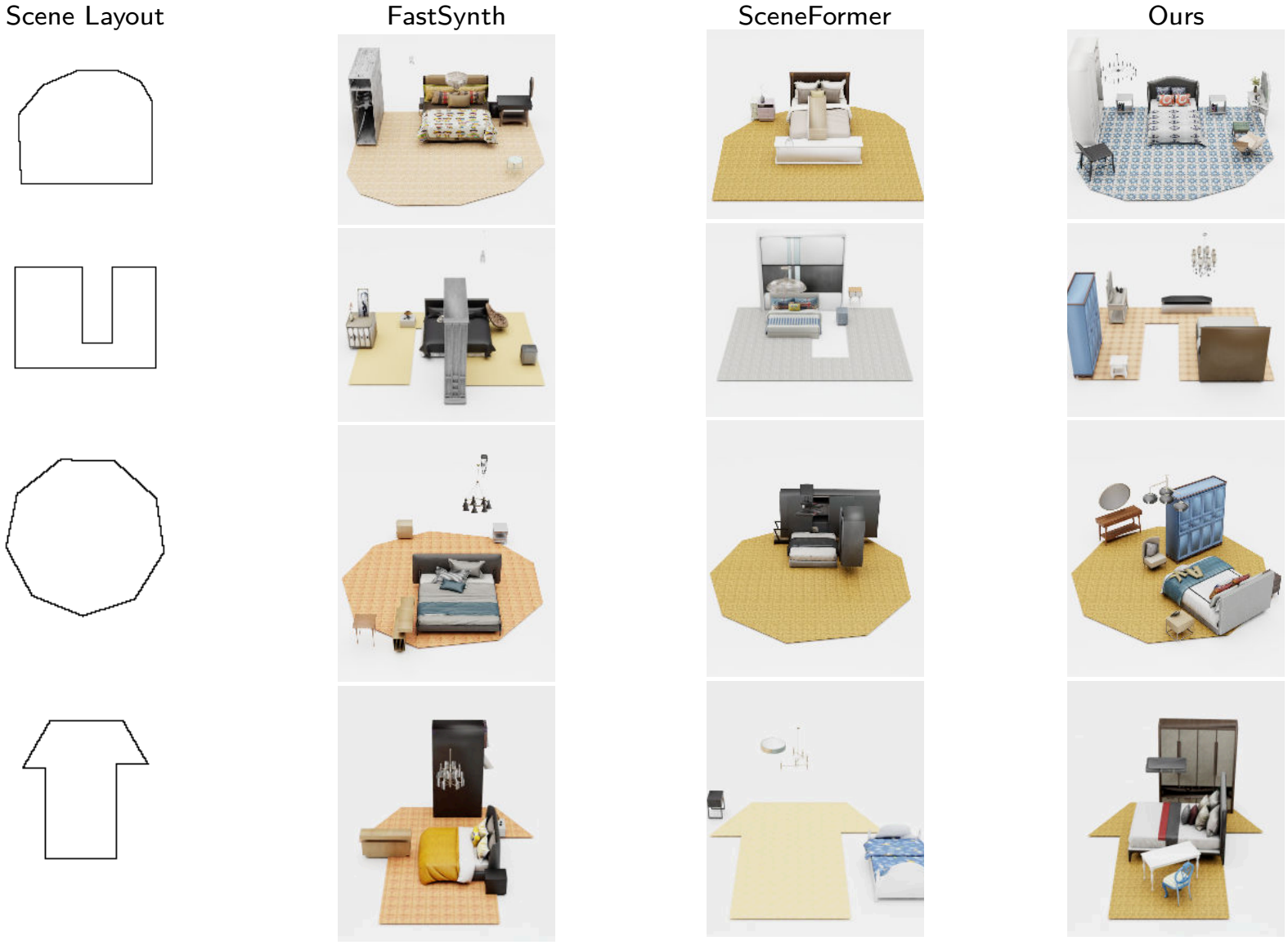


- **Layout encoder**: Computes a global feature representation for the floor.
- **Structure encoder**: Maps the j-th object to a per-object context embedding $\mathbf{C}_j$.
- **Transformer encoder**: Takes $\mathbf{F}, \{\mathbf{C}_j\}_{j=1}^{M}, \mathbf{q}$ and predicts the features $\hat{\mathbf{q}}$ of the next object to be added in the scene.
- **Attribute extractor**: Predicts the object attributes of the next object.

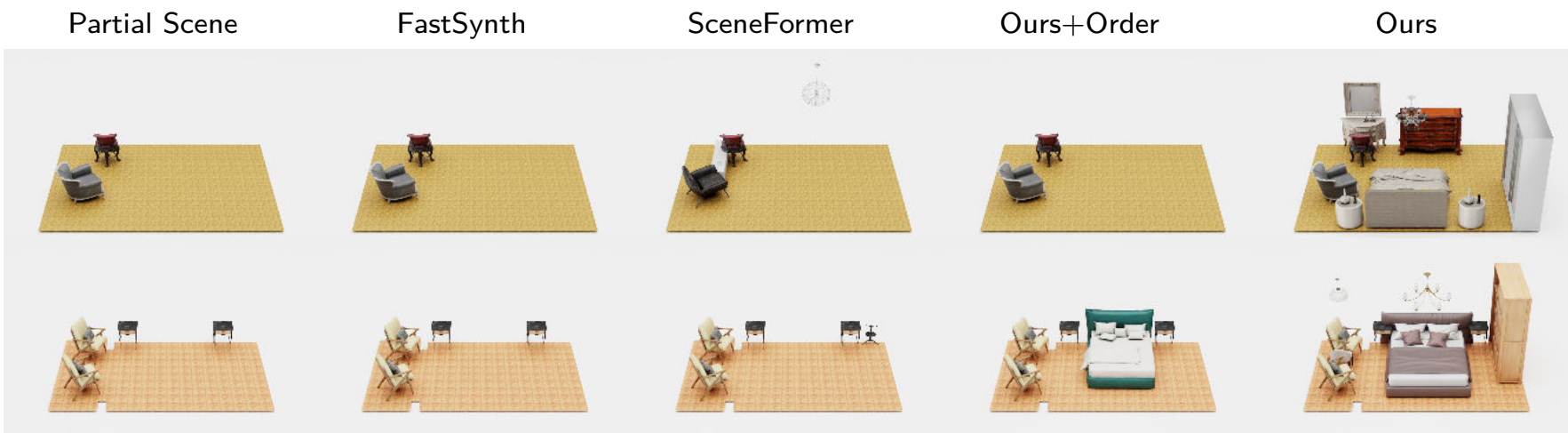
How well does it work?

Scene Synthesis

Generalization Beyond Training Data



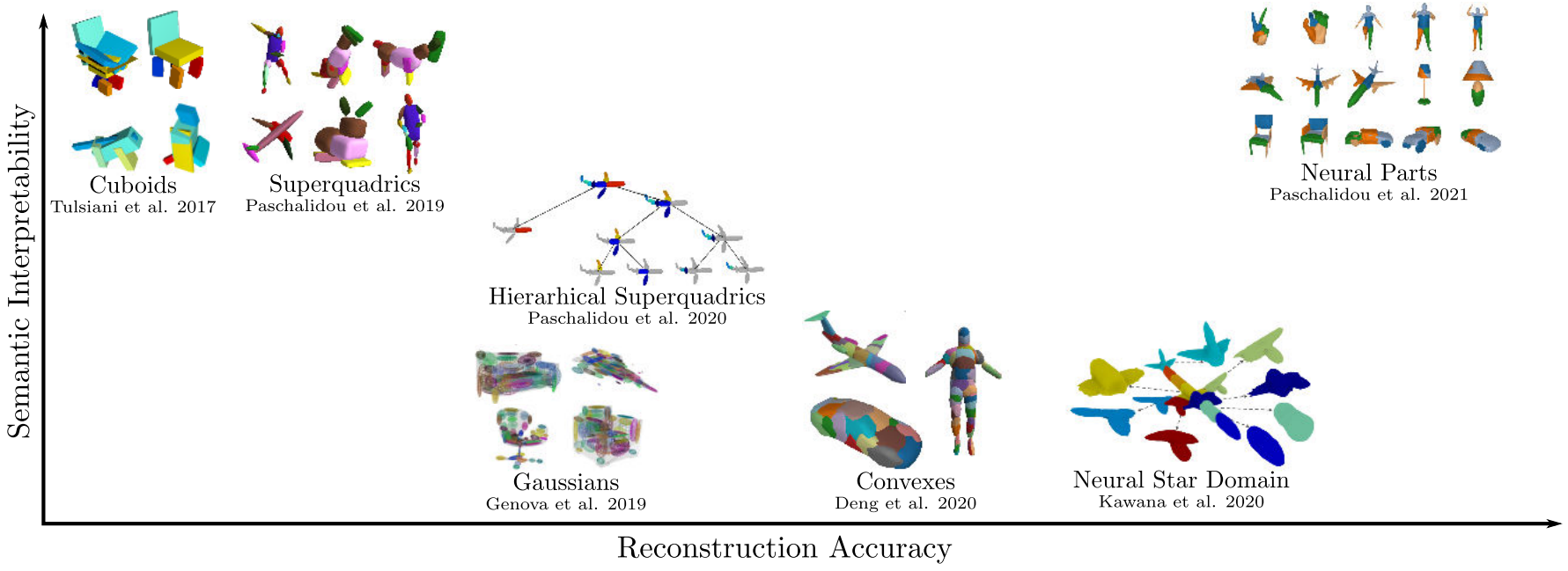
Scene Completion



Objects Suggestion

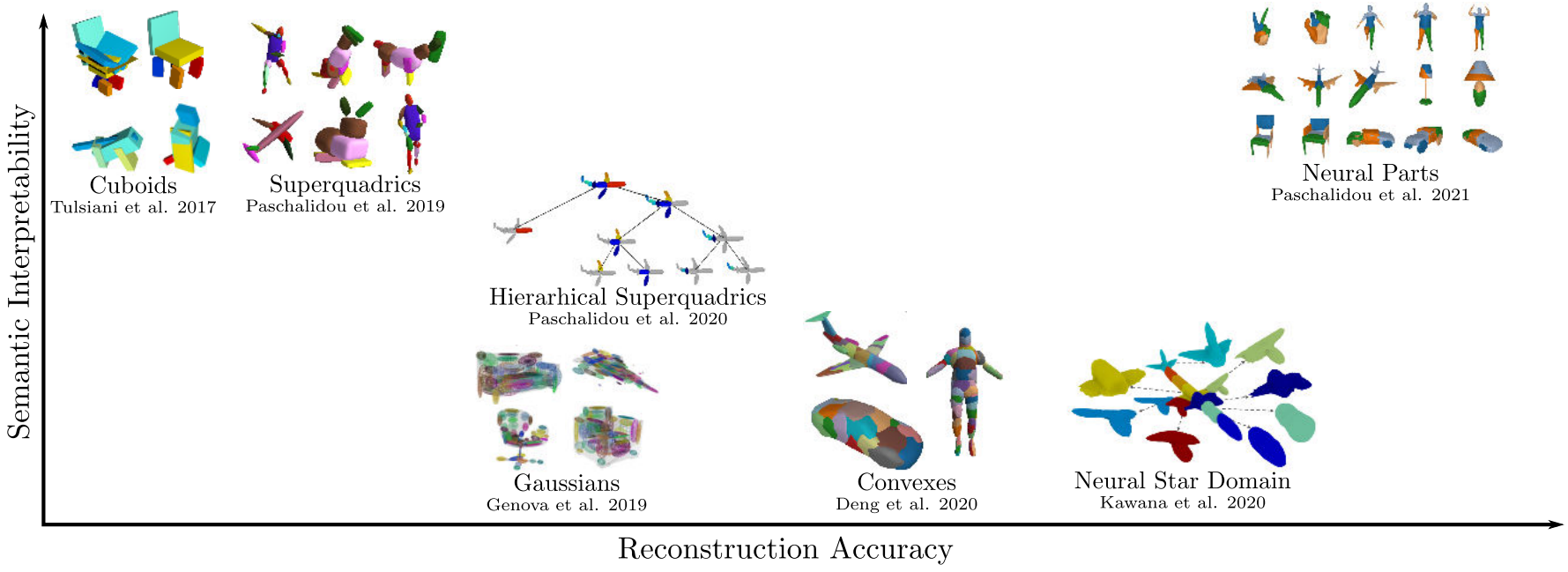
Failure Cases Correction

Conclusions - Object-Level Understanding



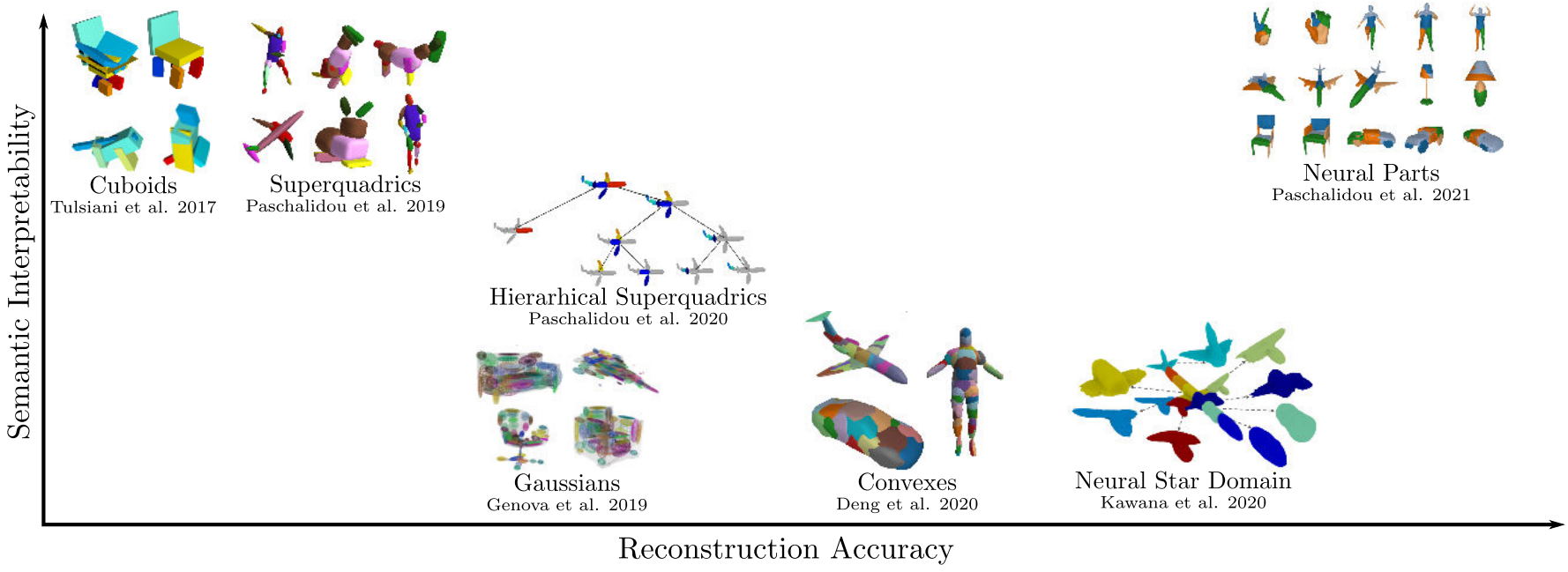
- Introduce primitive representations that are not limited to a specific family of shapes

Conclusions - Object-Level Understanding



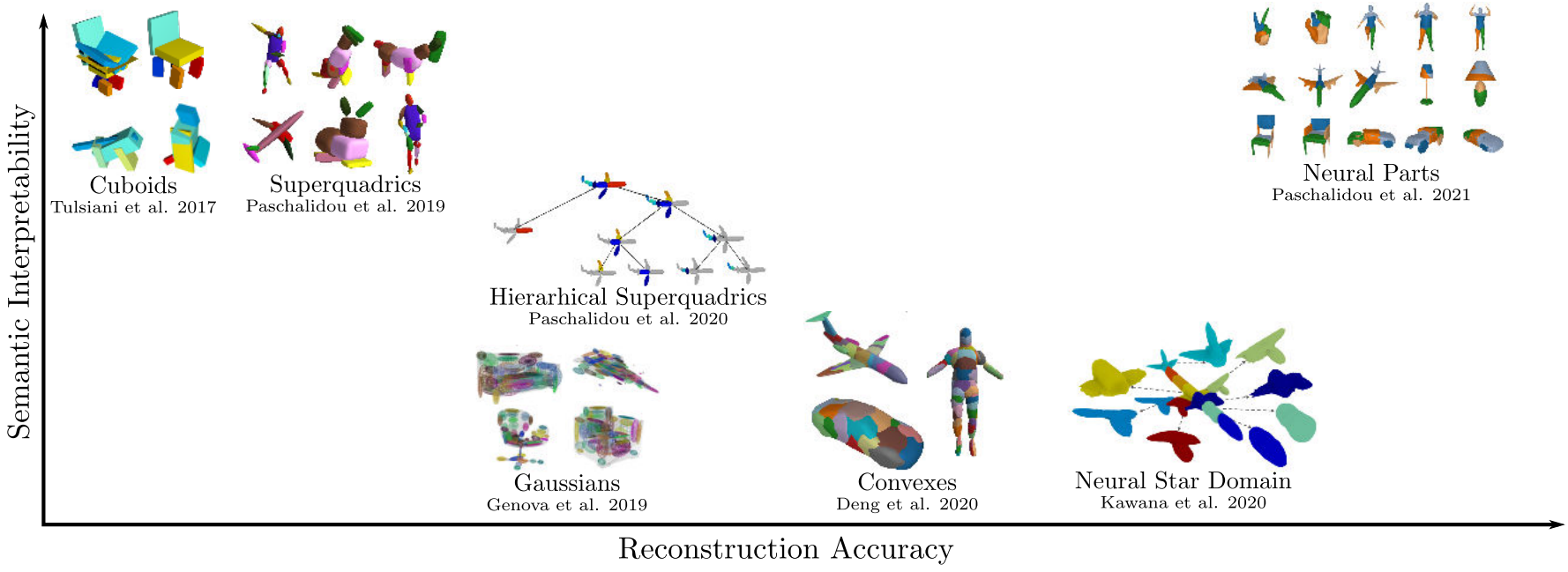
- Introduce primitive representations that are not limited to a specific family of shapes
- No need to compromise abstraction to improve reconstruction

Conclusions - Object-Level Understanding



- Introduce primitive representations that are not limited to a specific family of shapes
- No need to compromise abstraction to improve reconstruction
- A representation that jointly reasons about parts and part relationships

Conclusions - Object-Level Understanding

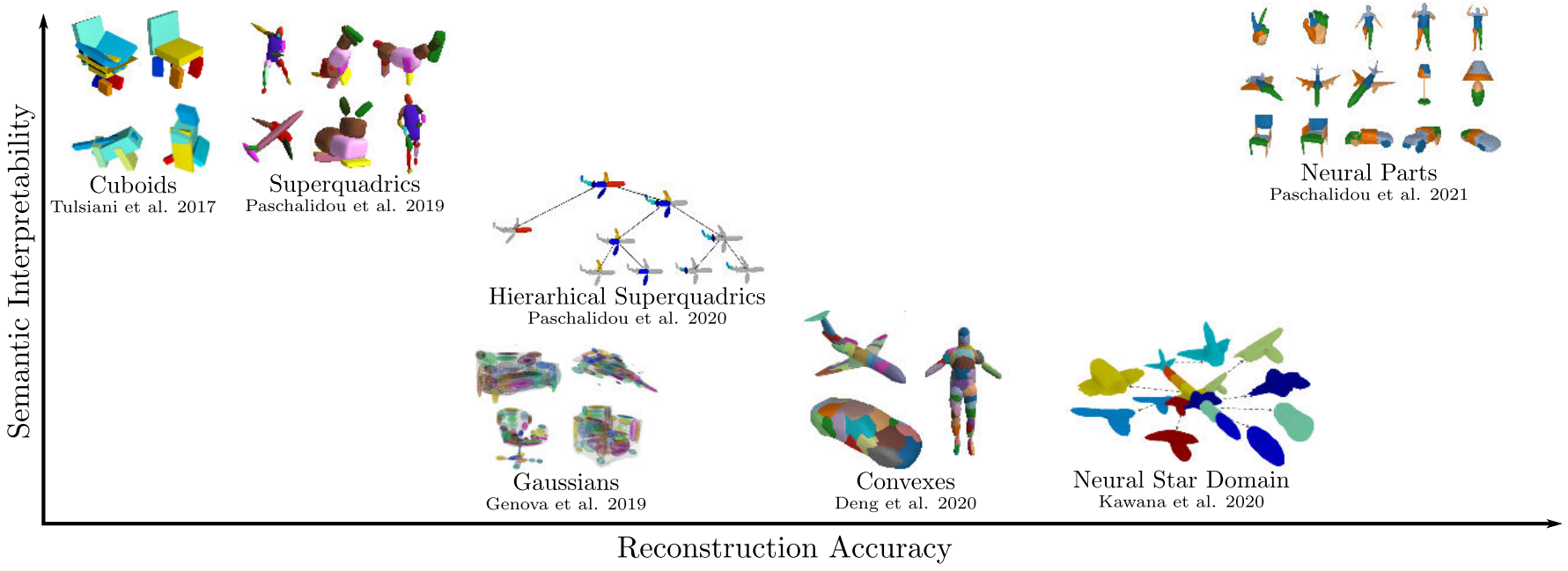


- Introduce primitive representations that are not limited to a specific family of shapes
- No need to compromise abstraction to improve reconstruction
- A representation that jointly reasons about parts and part relationships

Open Questions:

- **Do we really learn semantic parts**?

Conclusions - Object-Level Understanding

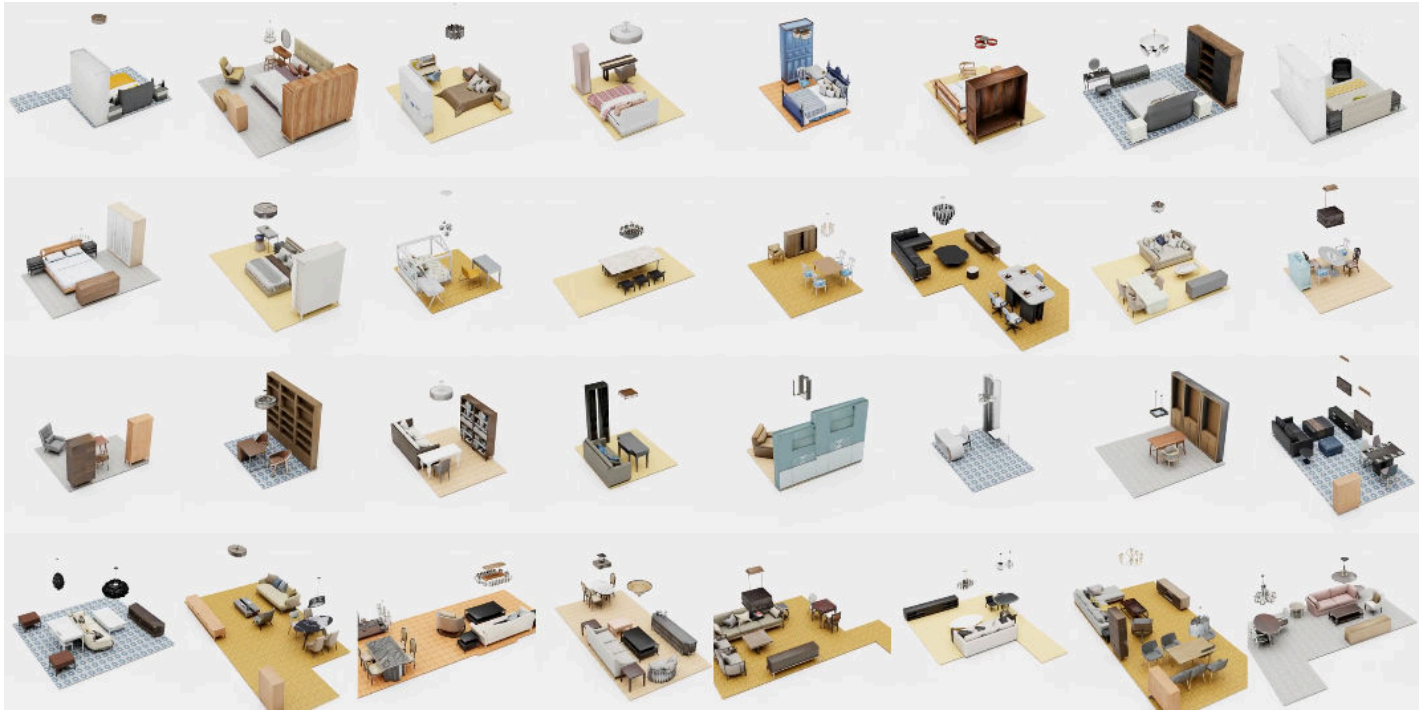


- Introduce primitive representations that are not limited to a specific family of shapes
- No need to compromise abstraction to improve reconstruction
- A representation that jointly reasons about parts and part relationships

Open Questions:

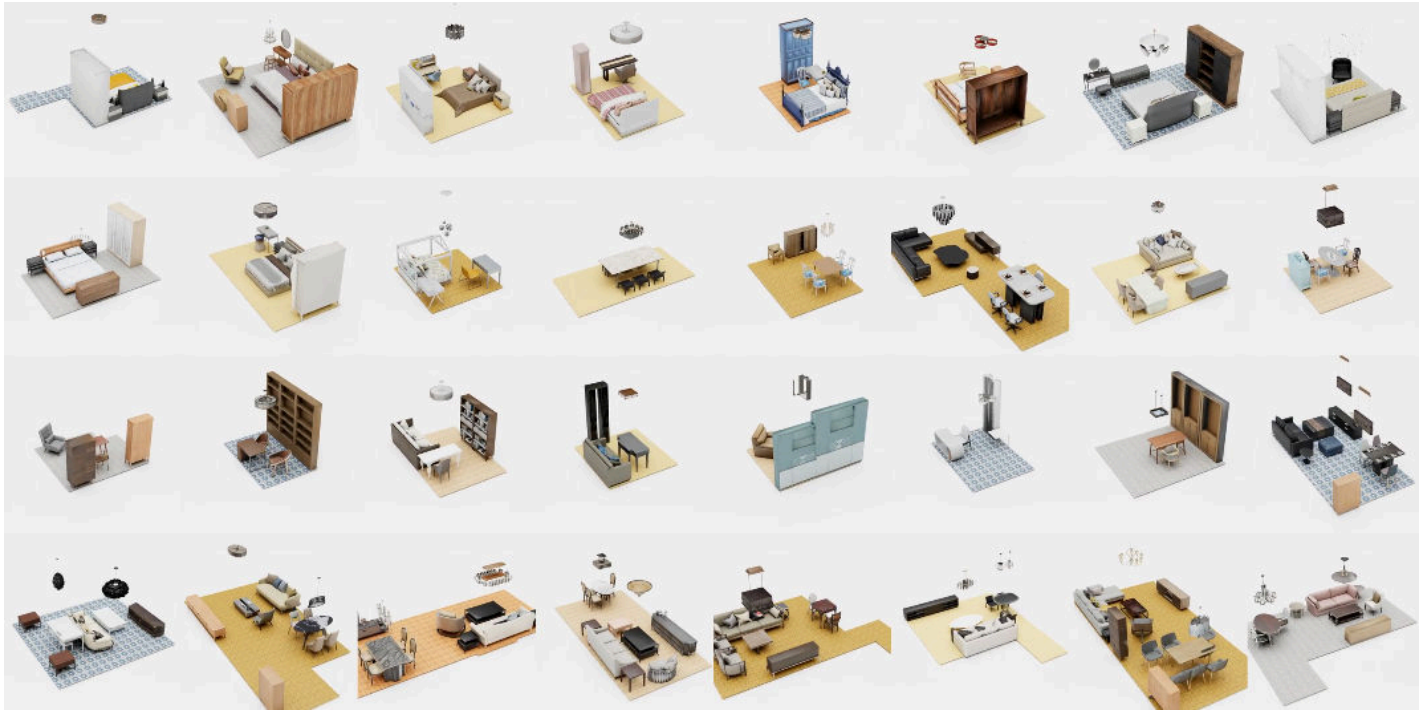
- **Do we really learn semantic parts**?
- We learn primitives by **optimizing the geometry**? Can't we do better?

Conclusions - Scene-Level Understanding



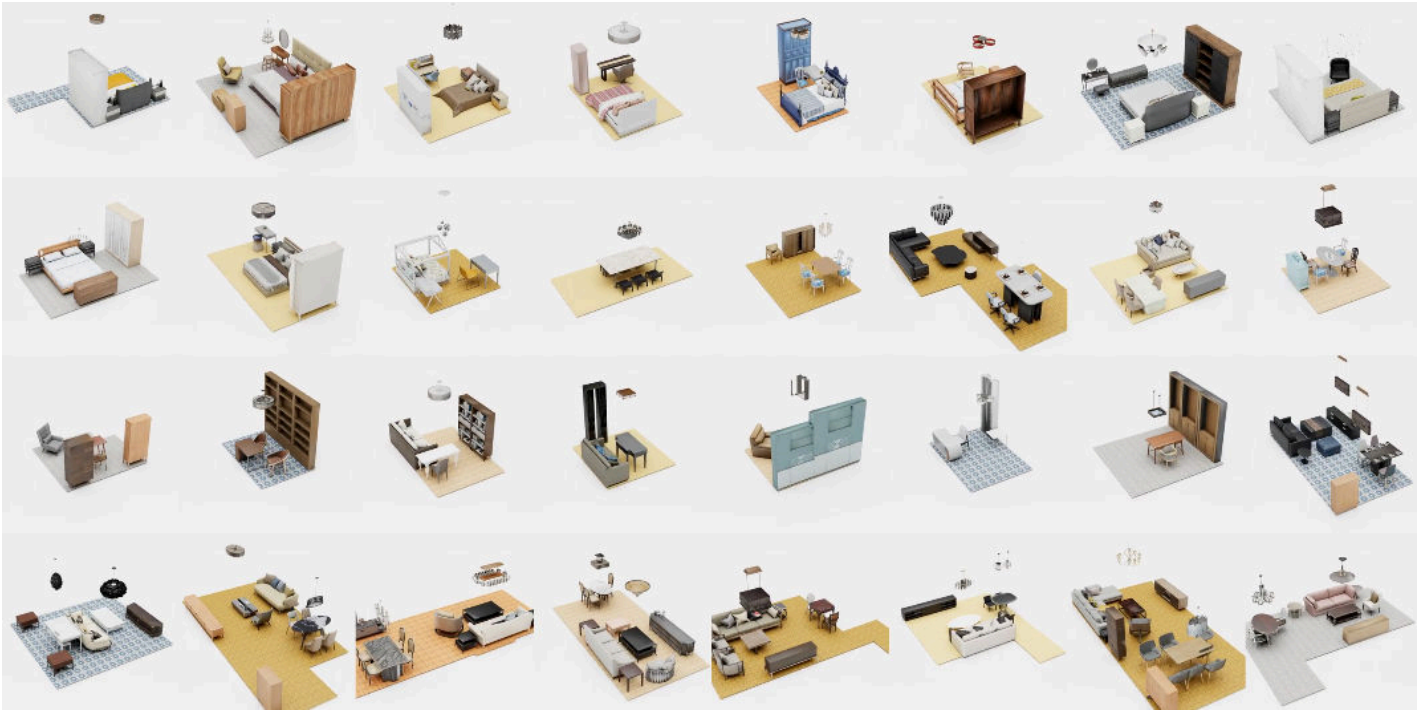
- Our unordered set formulation opens up multiple interactive applications

Conclusions - Scene-Level Understanding



- Our unordered set formulation opens up multiple interactive applications
- A model that has fewer parameters, is simpler and runs up to 8x faster

Conclusions - Scene-Level Understanding

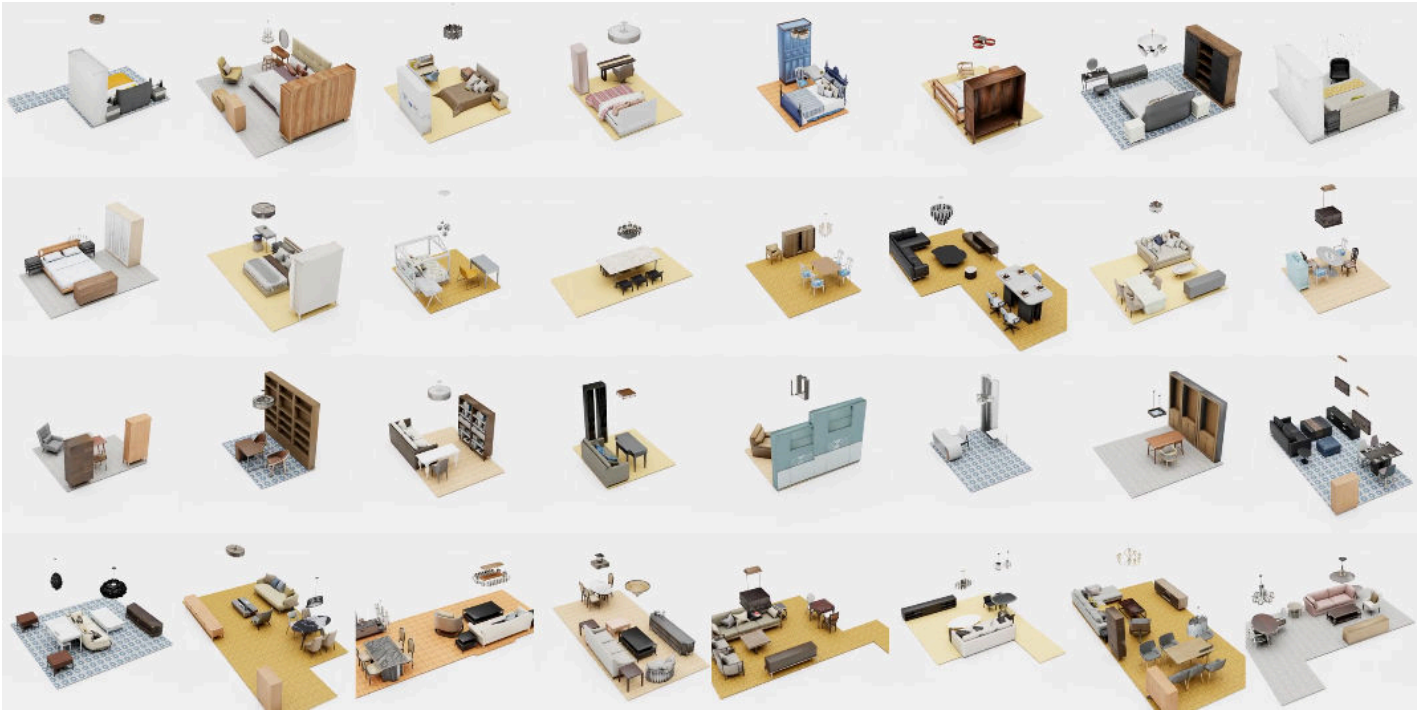


- Our unordered set formulation opens up multiple interactive applications
- A model that has fewer parameters, is simpler and runs up to 8x faster

Open Questions:

- The object retrieval module should not be independent of the scene synthesis pipeline

Conclusions - Scene-Level Understanding



- Our unordered set formulation opens up multiple interactive applications
- A model that has fewer parameters, is simpler and runs up to 8x faster

Open Questions:

- The object retrieval module should not be independent of the scene synthesis pipeline
- Combine scene synthesis pipelines with part-based generation models to allow for more interactive applications

Collaborators



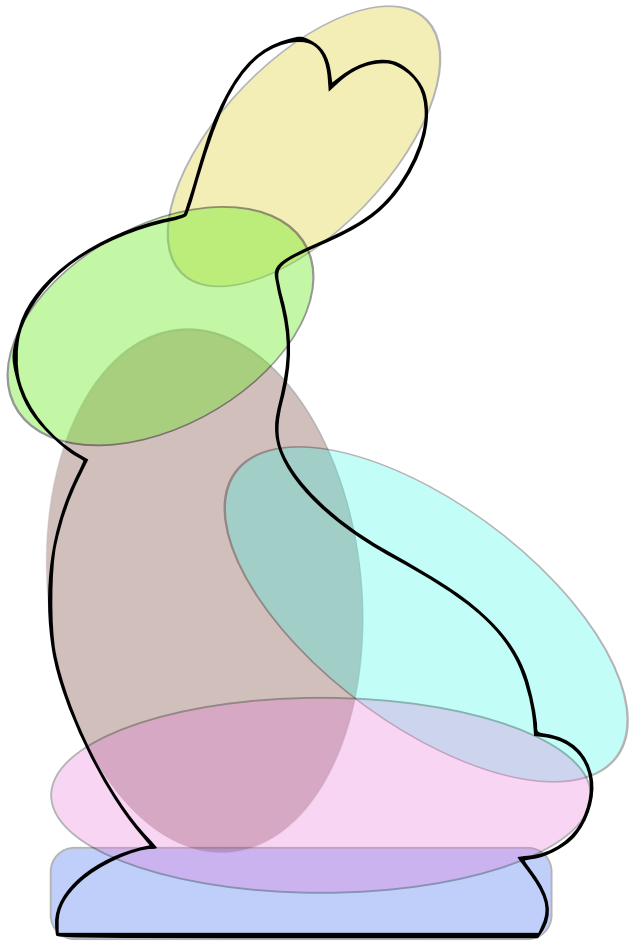




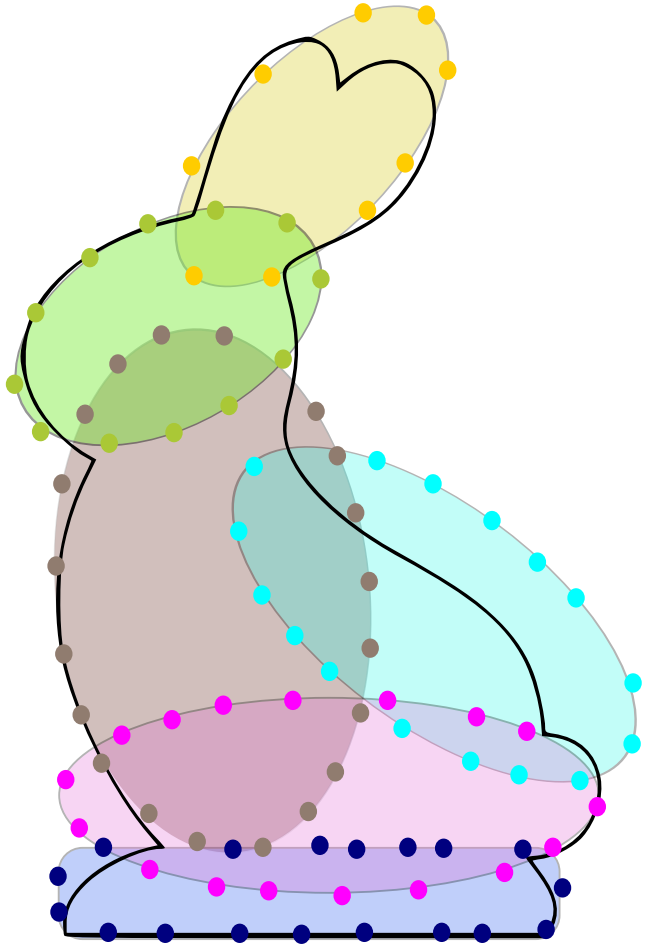



Thank you for your attention!

Explicit Representation of Predicted Shape



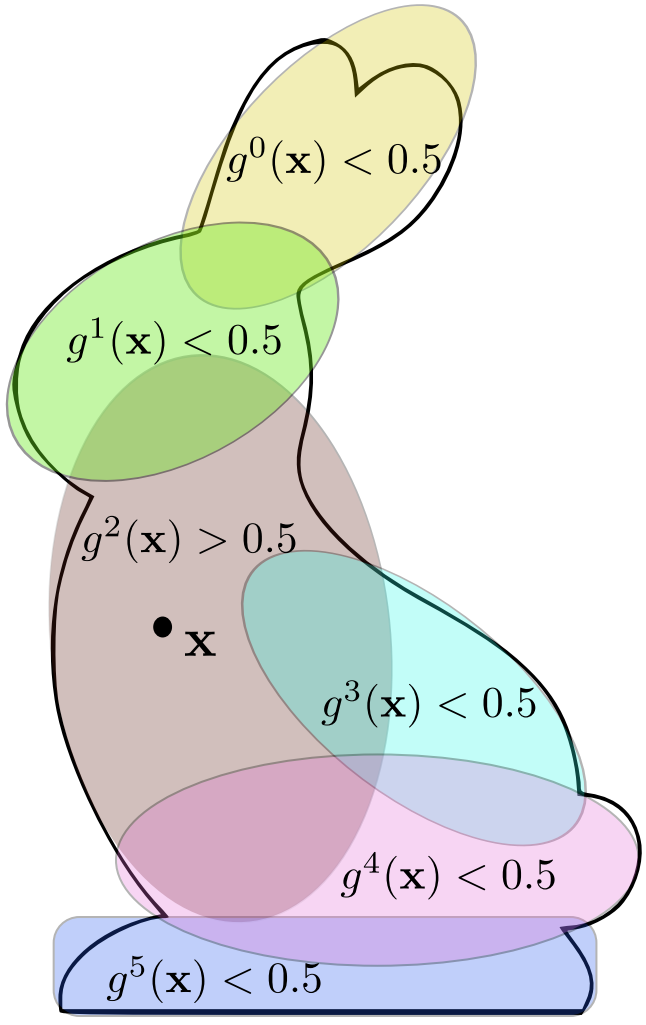
Explicit Representation of Predicted Shape



Explicit Representation:

$$\mathbf{r}(\eta,\omega) = \begin{bmatrix} \alpha_1 \cos^{\epsilon_1}\eta \cos^{\epsilon_2}\omega \\ \alpha_2 \cos^{\epsilon_1}\eta \sin^{\epsilon_2}\omega \\ \alpha_3 \sin^{\epsilon_1}\eta \end{bmatrix} \quad \begin{matrix} -\pi/2 \le \eta \le \pi/2 \\ -\pi \le \omega \le \pi \end{matrix}$$

Implicit Representation of Predicted Shape



Implicit Representation:

$$G^d(\mathbf{x}) = \max_{k \in 0 \ldots 2^d - 1} g_k^d(\mathbf{x})$$

$$g_k^d = \sigma(s(1 - f(\mathbf{x})^{\epsilon_1}))$$

$$f(\mathbf{x}) = \left(\left(\frac{x}{\alpha_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{\alpha_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{\alpha_3} \right)^{\frac{2}{\epsilon_1}}$$

Optimization Objective

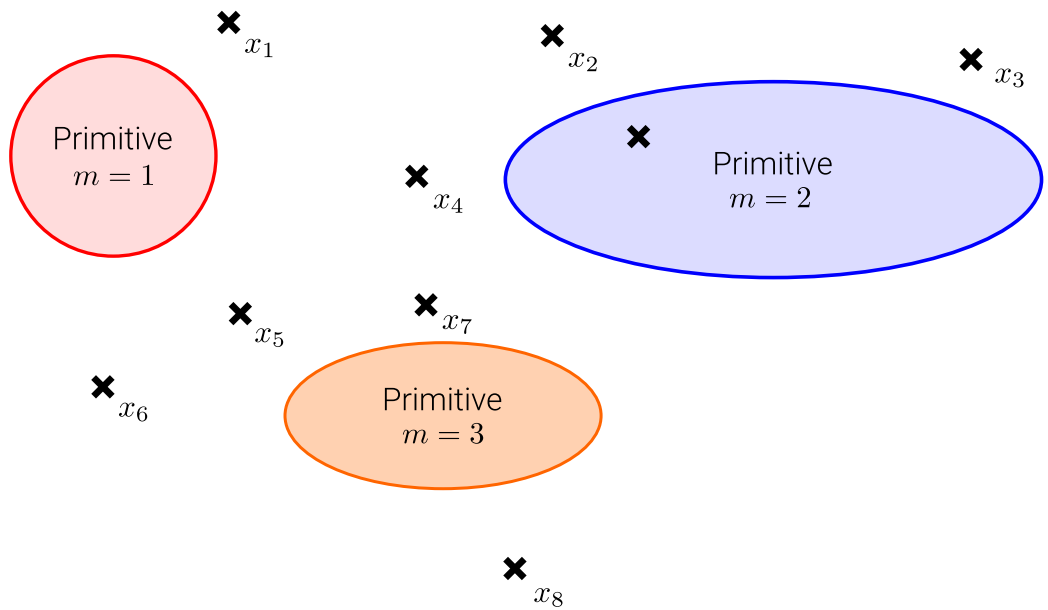
Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

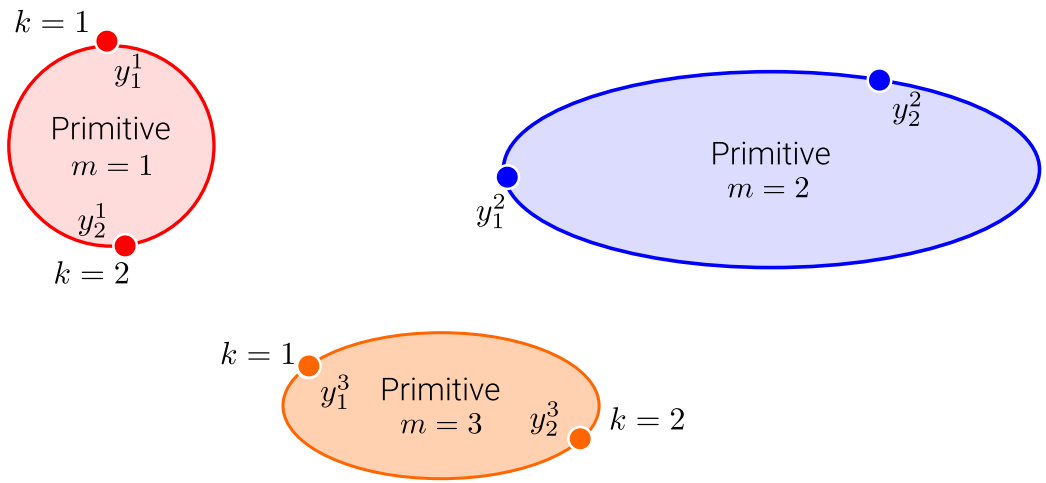


Points on the target: $\mathbf{X} = \{x_i\}_{i=1}^{N}$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$



Points on the m-th primitive: $\mathbf{Y}_m = \{y_k^m\}_{k=1}^K$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

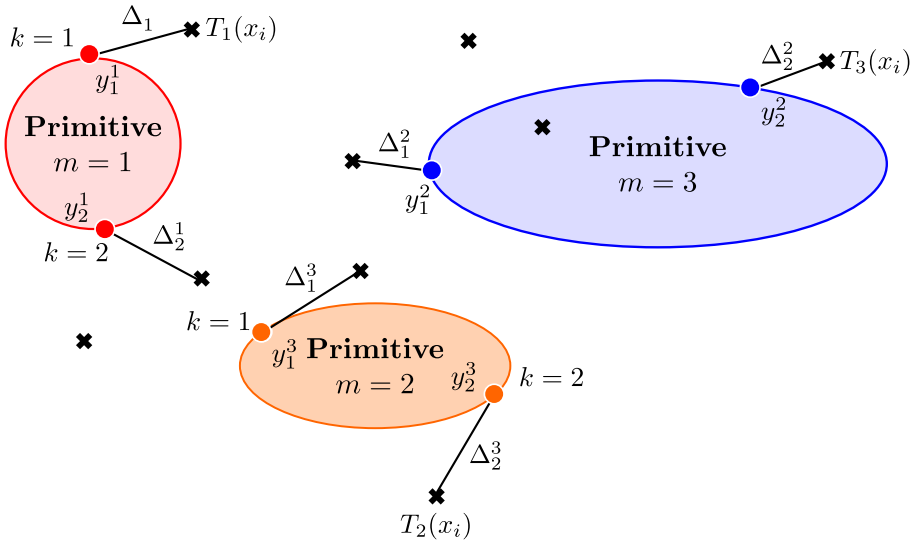
$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Primitive-to-Pointcloud Loss

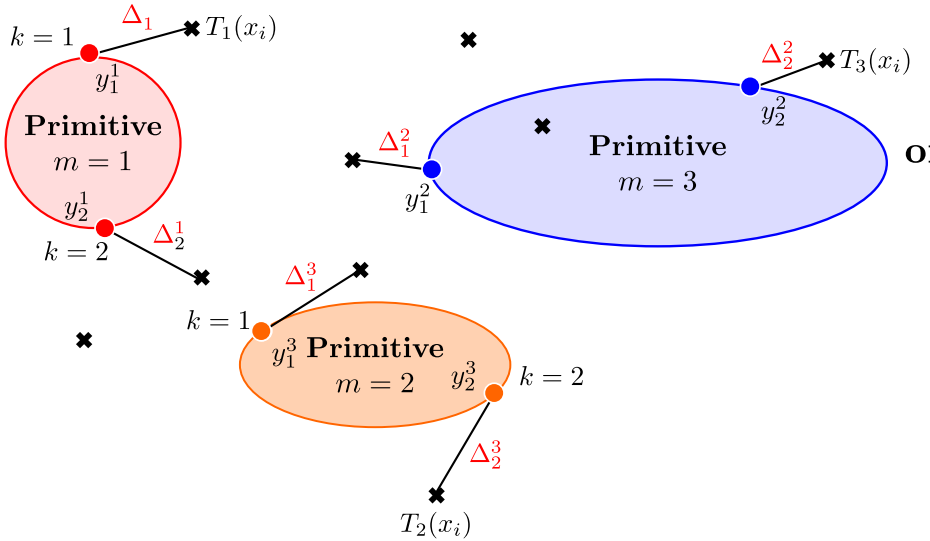


Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Primitive-to-Pointcloud Loss



Minimum distance from point y_k^m on primitive m to the target pointcloud

$$\Delta_k^m = \min_{i=1,..,N} \|\mathcal{T}_m(\mathbf{x}_i) - \mathbf{y}_k^m\|_2$$

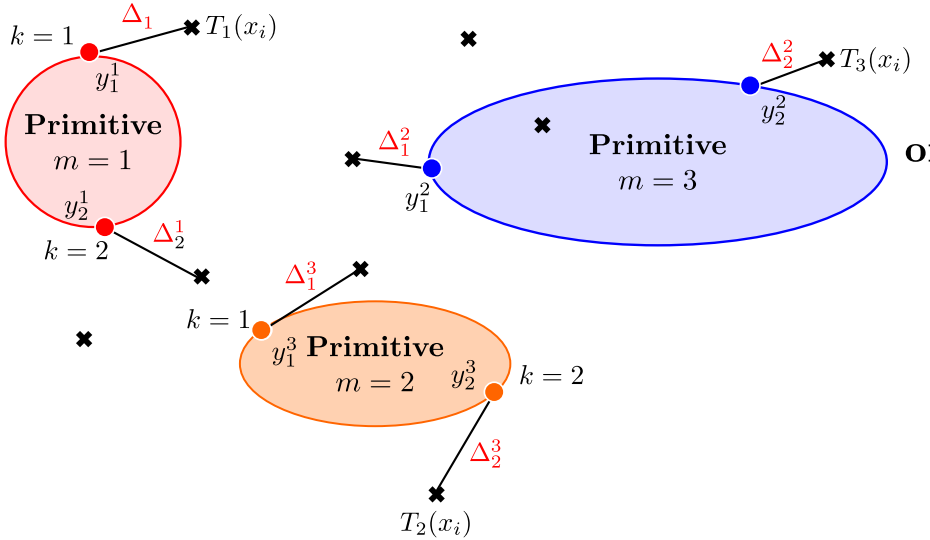
$$\mathcal{L}_{P\to X}^m(\mathbf{P}, \mathbf{X}) = \frac{1}{K}\sum_{k=1}^{K} \Delta_k^m$$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Primitive-to-Pointcloud Loss



Minimum distance from point y_k^m on primitive m to the target pointcloud

$$\Delta_k^m = \min_{i=1,..,N} \|\mathcal{T}_m(\mathbf{x}_i) - \mathbf{y}_k^m\|_2$$

$$\mathcal{L}_{P\to X}^m(\mathbf{P}, \mathbf{X}) = \frac{1}{K}\sum_{k=1}^{K} \Delta_k^m$$

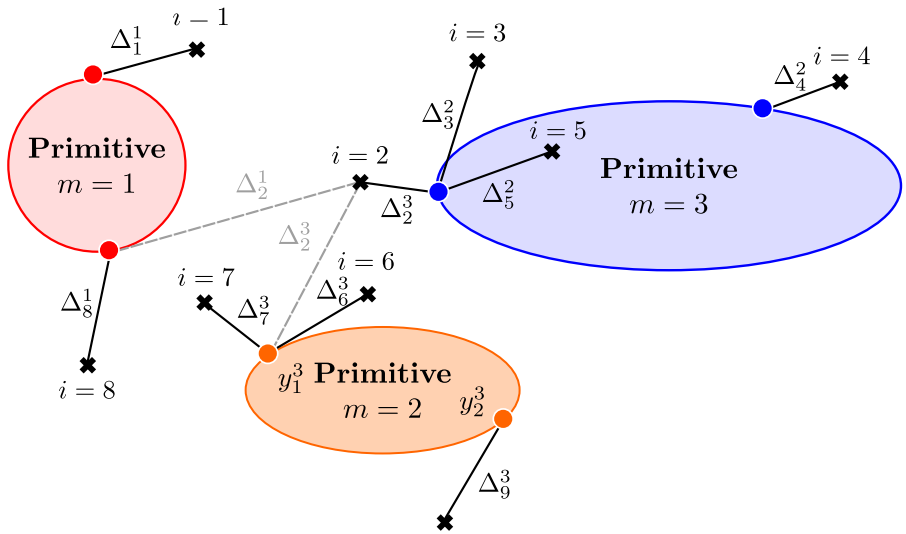
$$\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X}) = \mathbb{E}_{p(\mathbf{z})}\left[\sum_{m|z_m=1} \mathcal{L}_{P\to X}^m(\mathbf{P}, \mathbf{X})\right] = \sum_{m=1}^{M} \gamma_m \mathcal{L}_{P\to X}^m(\mathbf{P}, \mathbf{X})$$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Pointcloud-to-Primitive Loss

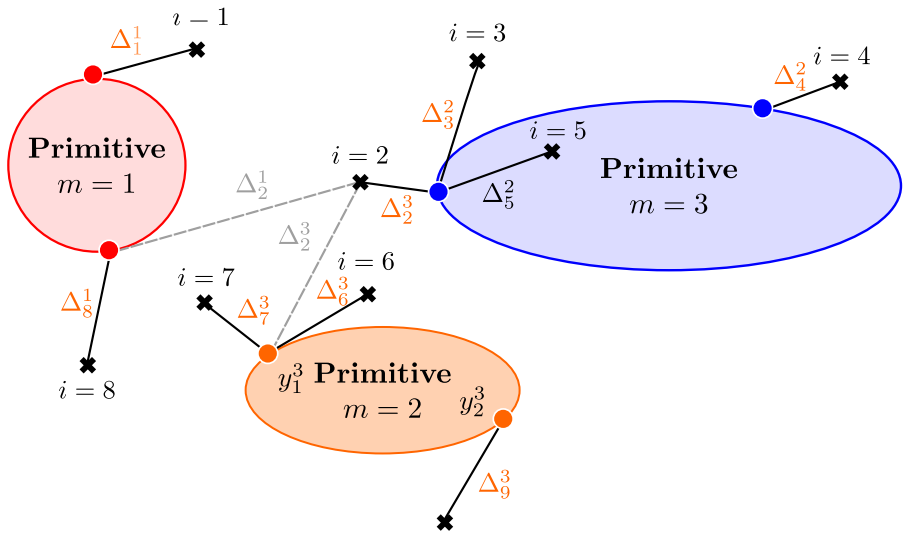


Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Pointcloud-to-Primitive Loss



Minimum distance from point x_i to the predicted shape

$$\Delta_i^m = \min_{k=1,..,K} \|\mathcal{T}_m(\mathbf{x}_i) - \mathbf{y}_k^m\|_2$$

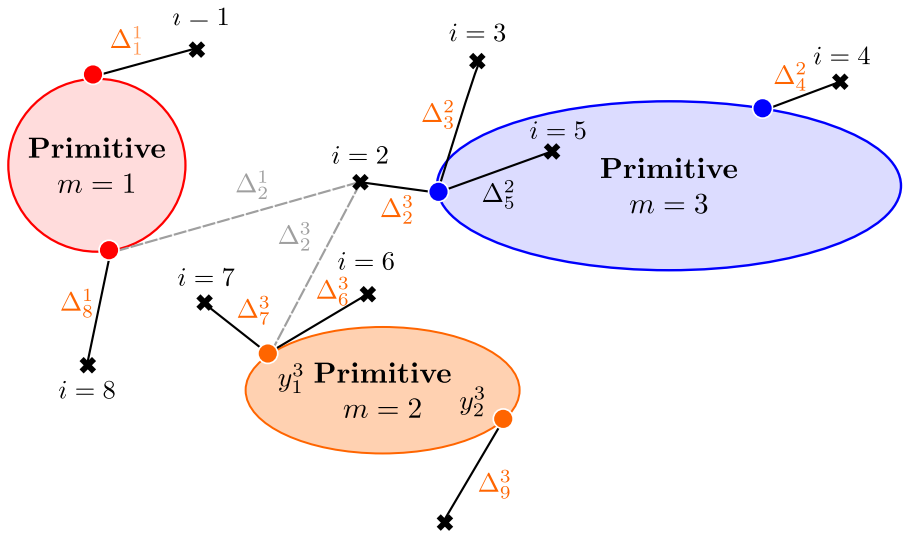
$$\mathcal{L}_{X\to P}^i(\mathbf{X}, \mathbf{P}) = \min_{m|z_m=1} \Delta_i^m$$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_{\gamma}(\mathbf{P})}_{\text{Parsimony}}$$

Pointcloud-to-Primitive Loss



Minimum distance from point x_i to the predicted shape

$$\Delta_i^m = \min_{k=1,..,K} \|\mathcal{T}_m(\mathbf{x}_i) - \mathbf{y}_k^m\|_2$$

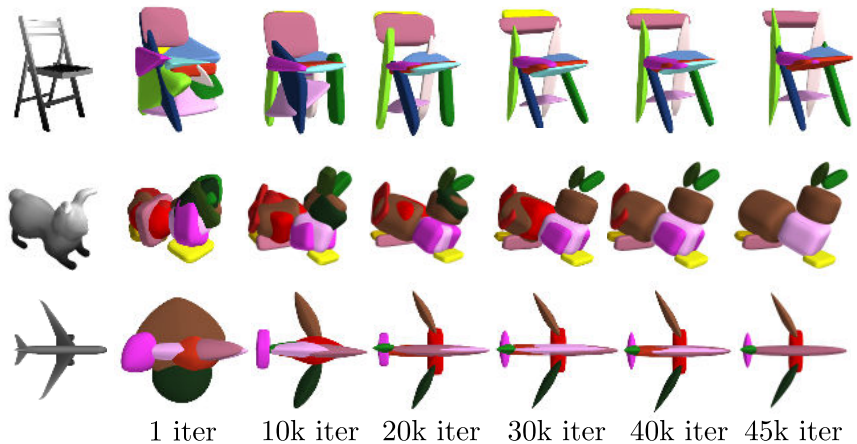
$$\mathcal{L}_{X\to P}^i(\mathbf{X}, \mathbf{P}) = \min_{m|z_m=1} \Delta_i^m$$

$$\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P}) = \mathbb{E}_{p(\mathbf{z})}\left[\sum_{\mathbf{x}_i\in\mathbf{X}} \mathcal{L}_{X\to P}^i(\mathbf{X}, \mathbf{P})\right] = \sum_{\mathbf{x}_i\in\mathbf{X}}\sum_{m=1}^{M} \Delta_i^m \gamma_m \prod_{\bar{m}=1}^{m-1}(1-\gamma_{\bar{m}})$$

Optimization Objective

Overall Loss: Measure the discrepancy between the target and the predicted shape:

$$\mathcal{L}_D(\mathbf{P}, \mathbf{X}) = \underbrace{\mathcal{L}_{P\to X}(\mathbf{P}, \mathbf{X})}_{\text{Primitive-to-Pointcloud}} + \underbrace{\mathcal{L}_{X\to P}(\mathbf{X}, \mathbf{P})}_{\text{Pointcloud-to-Primitive}} + \underbrace{\mathcal{L}_\gamma(\mathbf{P})}_{\text{Parsimony}}$$

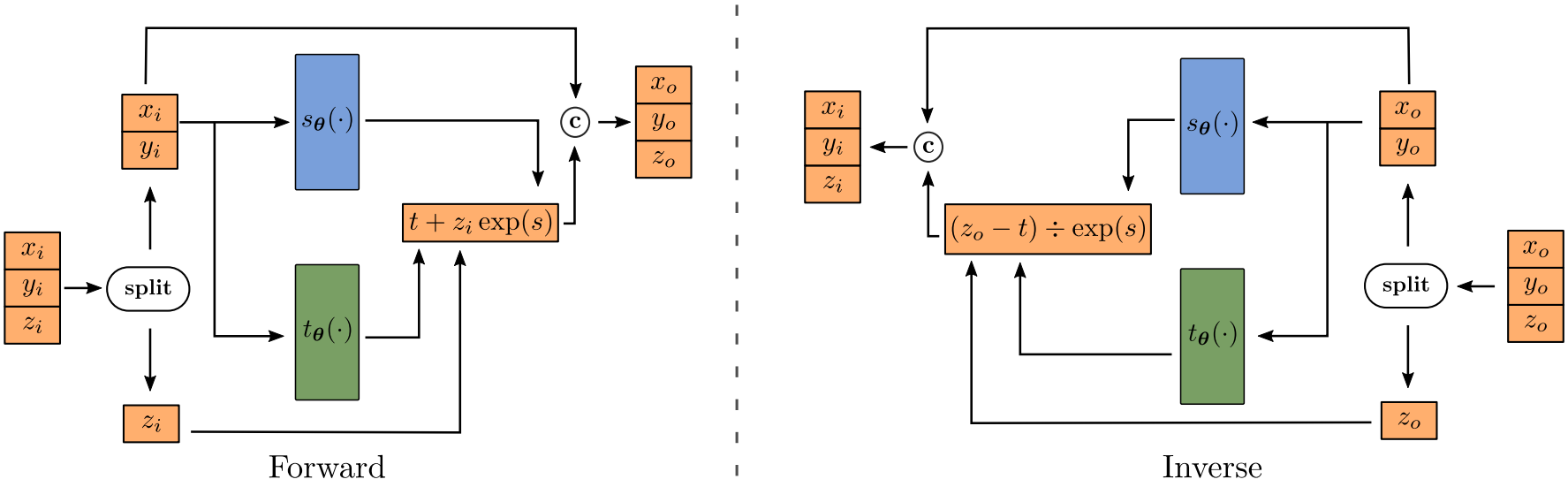


$$\mathcal{L}_\gamma(\mathbf{P}) = \underbrace{\max\left(1 - \sum_{m=1}^{M} \gamma_m, 0\right)}_{\text{Avoid trivial solutions}} + \underbrace{\beta\sqrt{\sum_{m=1}^{M} \gamma_m}}_{\text{Ensure Parsimony}}$$

Single View 3D Reconstruction on SURREAL

Parametrizing a Homeomorphism with an INN

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



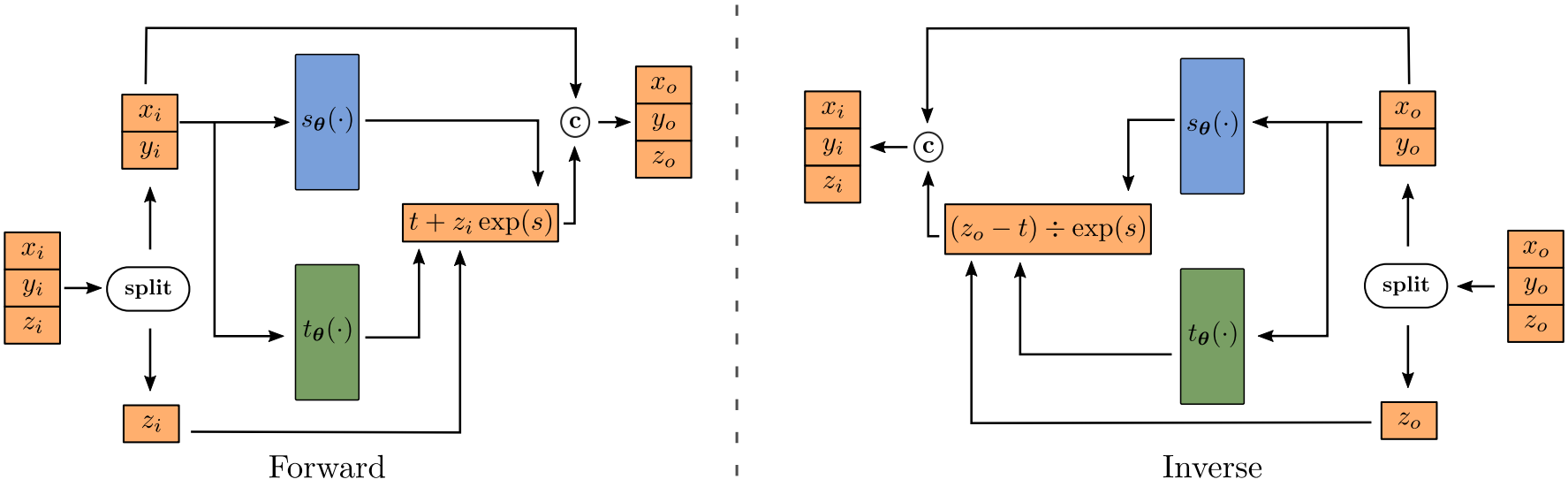
$$x_o = x_i$$

$$y_o = y_i$$

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

Parametrizing a Homeomorphism with an INN

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



$$x_o = x_i$$

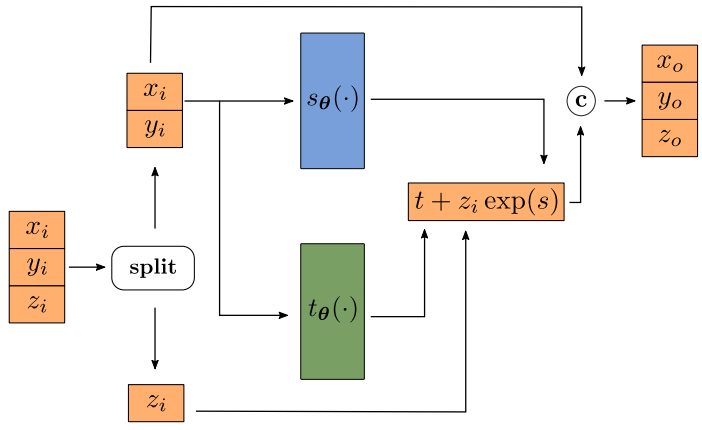
$$y_o = y_i$$

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

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

Conditional Homeomorphism

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



Original Coupling Layer

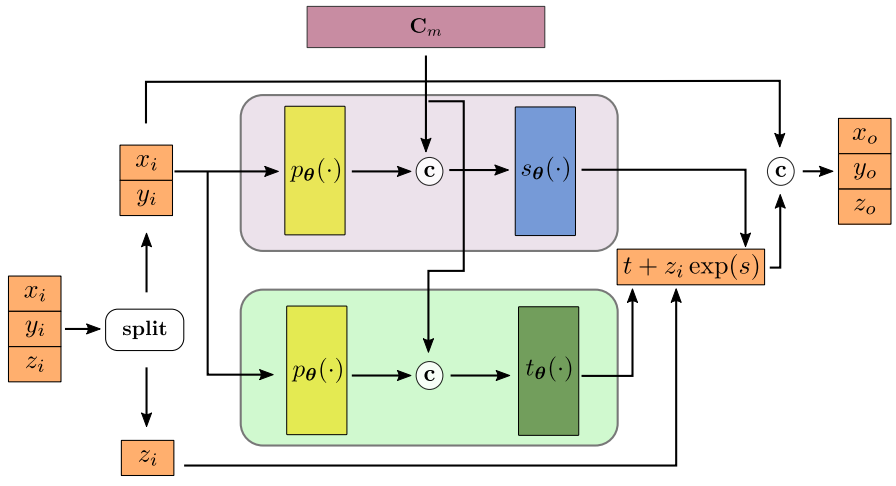
$$x_o = x_i$$

$$y_o = y_i$$

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

Conditional Homeomorphism

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



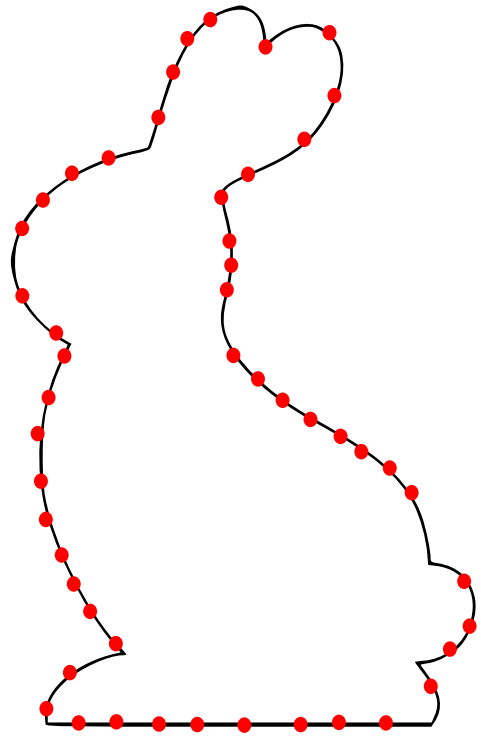
Conditional Coupling Layer

$$x_o = x_i$$

$$y_o = y_i$$

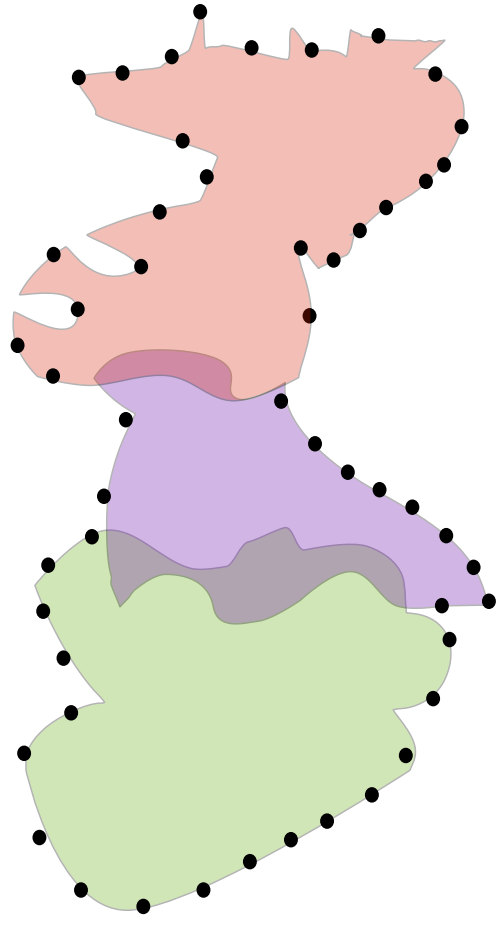
$$z_o = z_i \exp\left(s_{\theta}\left([\mathbf{C}_m; p_{\theta}\left(x_i, y_i\right)]\right)\right) + t_{\theta}\left([\mathbf{C}_m; p_{\theta}\left(x_i, y_i\right)]\right)$$

Reconstruction Loss



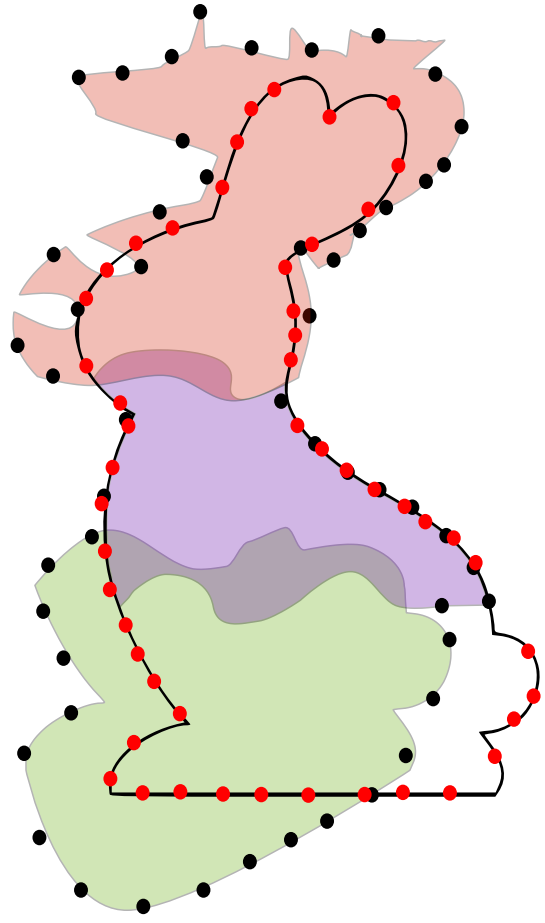
Target Surface Samples: $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$

Reconstruction Loss



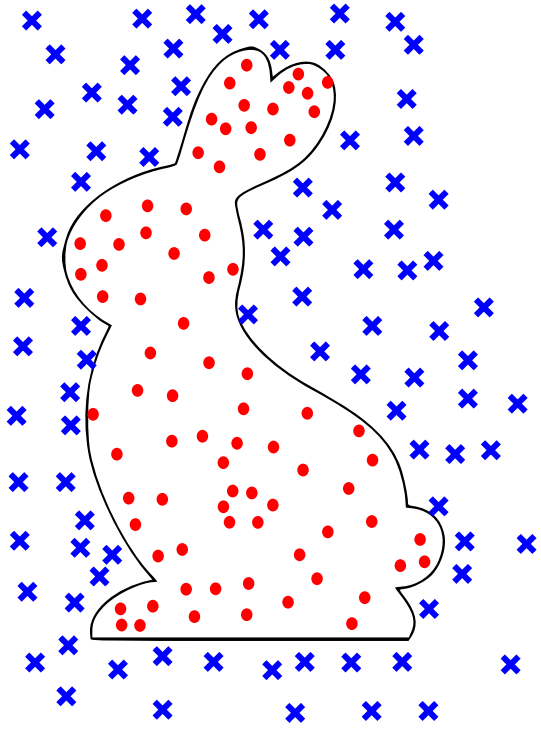
Predicted Surface Samples: $\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}$

Reconstruction Loss



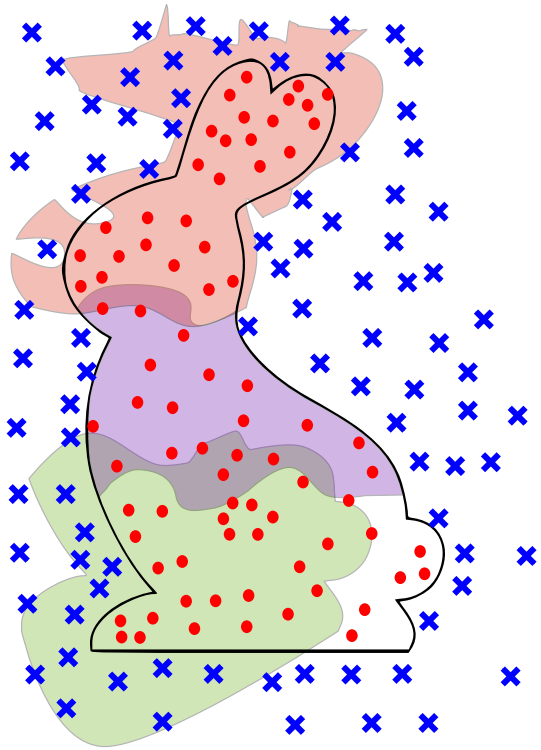
$$\mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) = \frac{1}{|\mathcal{X}_t|} \sum_{\mathbf{x}_i \in \mathcal{X}_t} \min_{\mathbf{x}_j \in \mathcal{X}_p} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_j \in \mathcal{X}_p} \min_{\mathbf{x}_i \in \mathcal{X}_t} \|\mathbf{x}_i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

Occupancy Loss

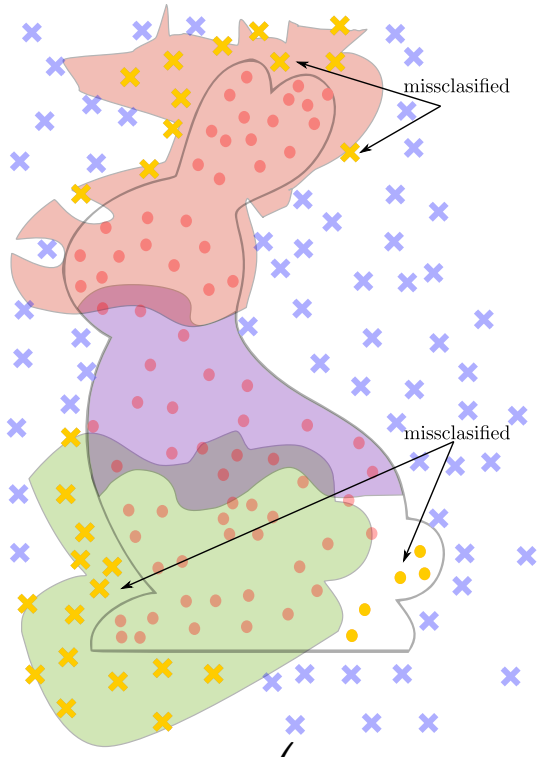


Target Volumetric Samples: $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Occupancy Loss

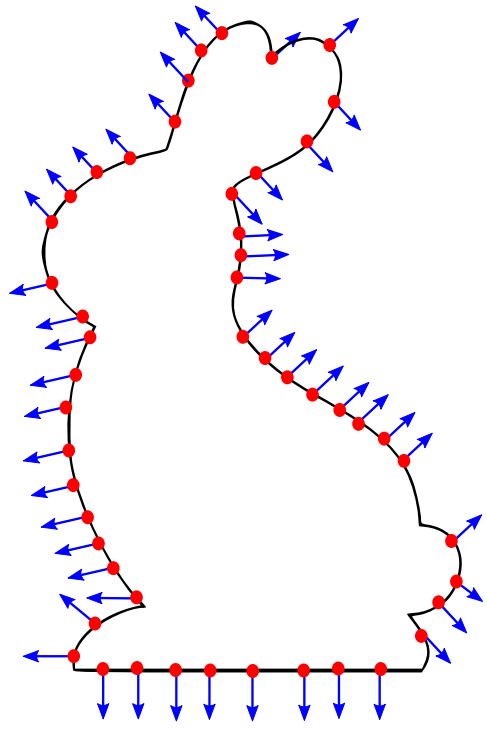


Occupancy Loss



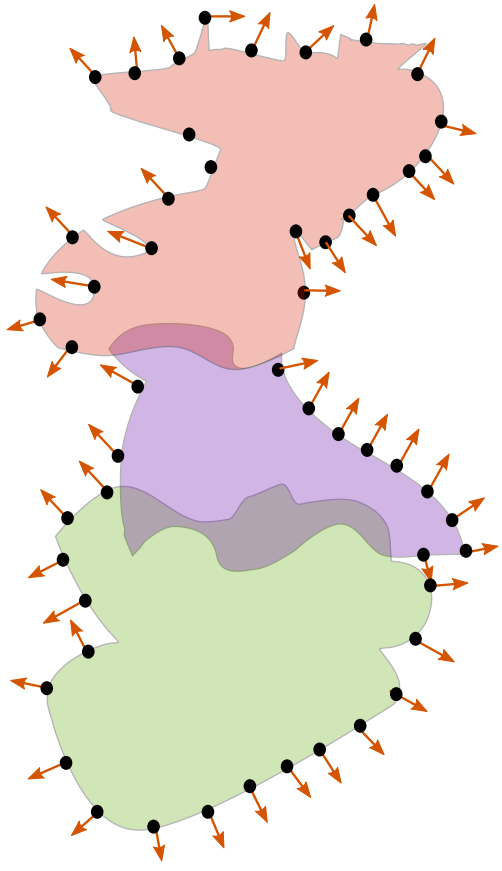
$$\mathcal{L}_{occ}(\mathcal{X}_o) = \sum_{(\mathbf{x},o)\in\mathcal{X}_o} \mathcal{L}_{ce}\left(\underbrace{\sigma\left(\frac{-G(\mathbf{x})}{\tau}\right)}_{\substack{>1 \text{ when } \mathbf{x} \\ \text{inside the predicted shape}}}, o\right)$$

Normal Consistency Loss



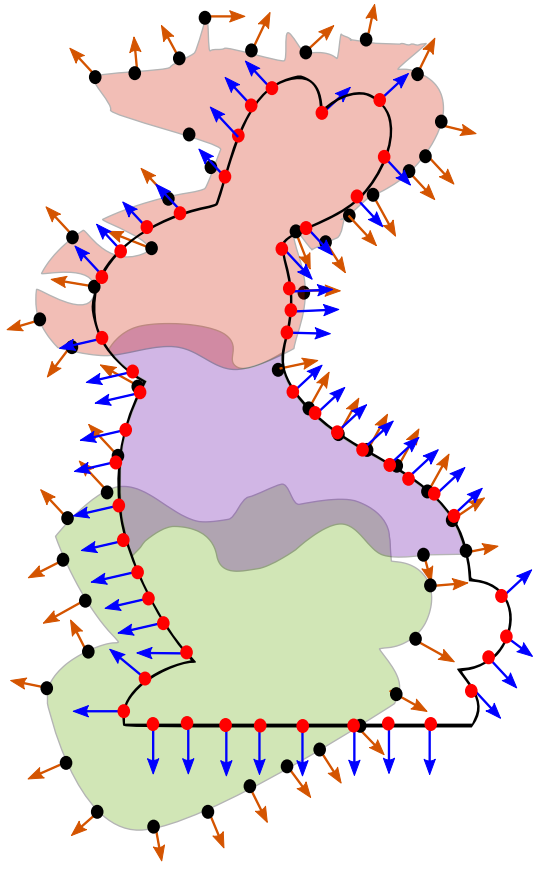
Target Surface Samples: $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^{N}$

Normal Consistency Loss



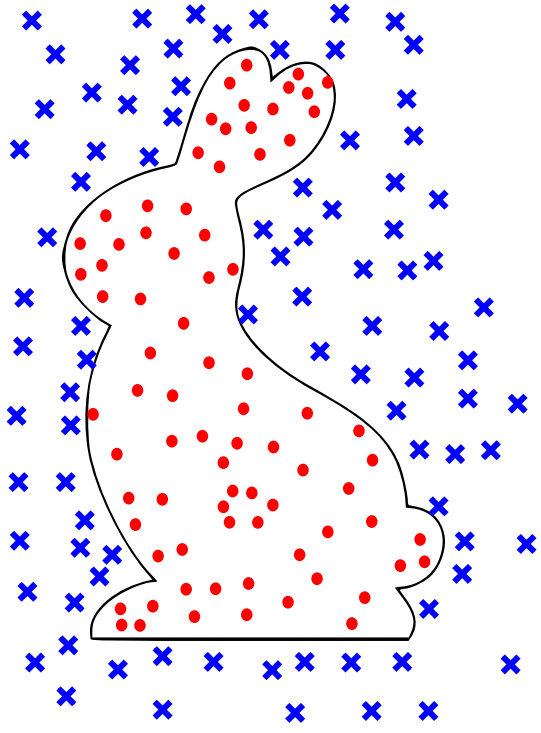
Predicted Surface Normals: $\frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}$

Normal Consistency Loss



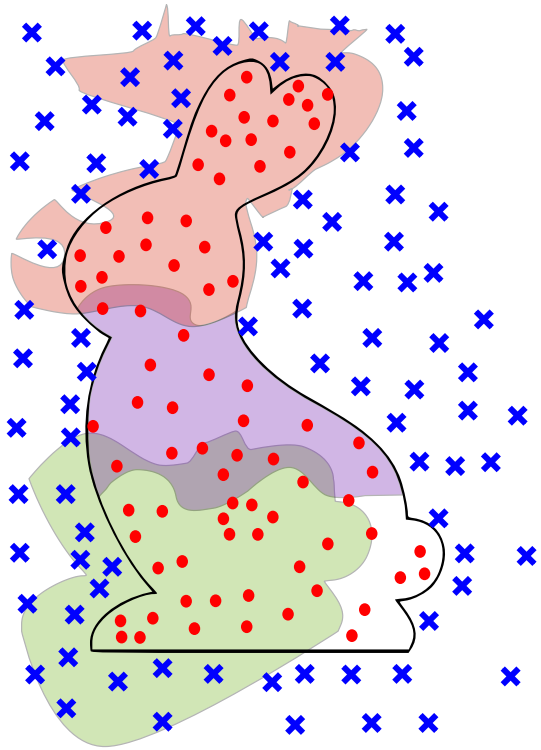
$$\mathcal{L}_{norm}(\mathcal{X}_t) = \frac{1}{|\mathcal{X}_t|} \sum_{(\mathbf{x},\mathbf{n})\in\mathcal{X}_t} \left(1 - \left\langle \frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}, n \right\rangle\right)$$

Overlapping Loss

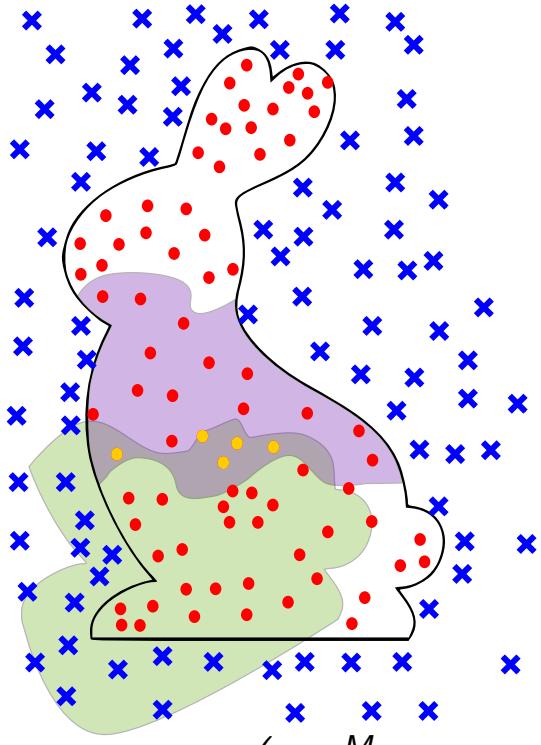


Target Volumetric Samples: $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

Overlapping Loss

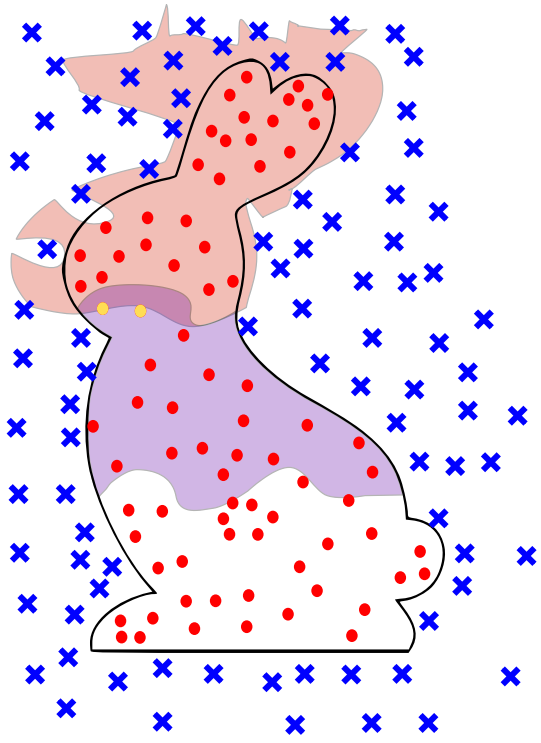


Overlapping Loss



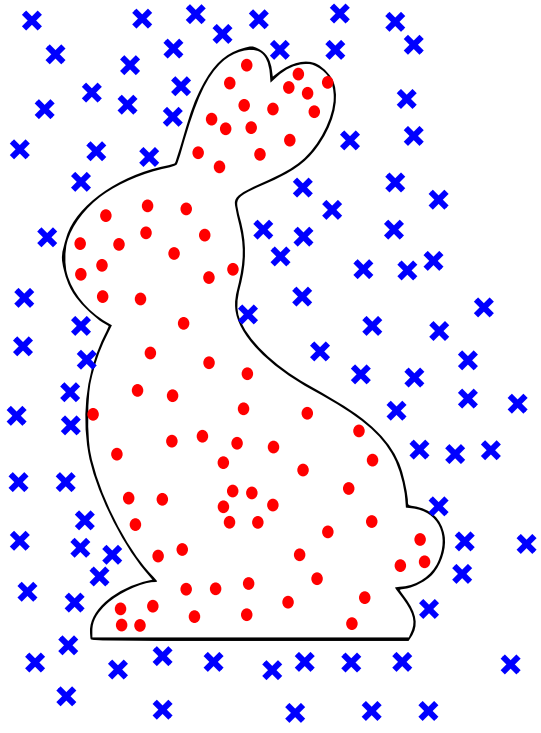
$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-g^m(\mathbf{x})}{\tau}\right) - \lambda\right)$$

Overlapping Loss



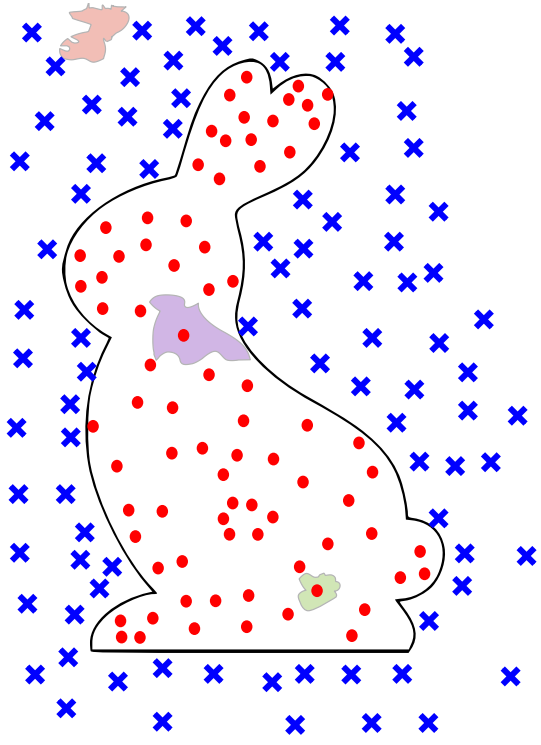
$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-g^m(\mathbf{x})}{\tau}\right) - \lambda\right)$$

Coverage Loss



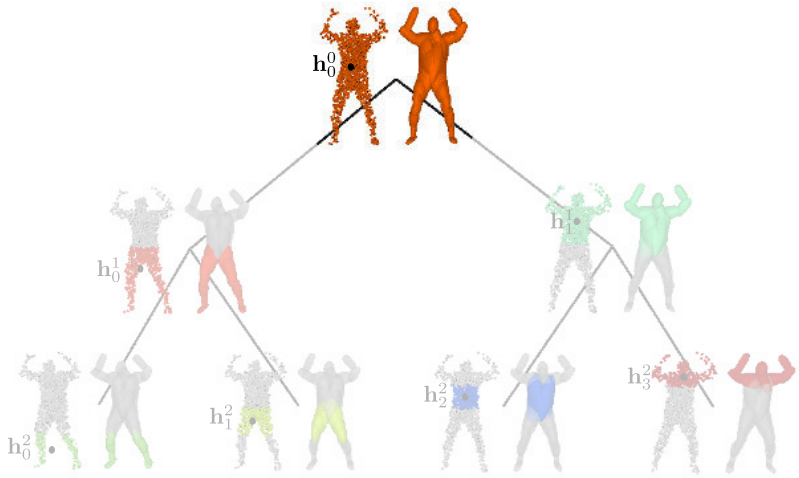
Target Volumetric Samples: $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^{V}$

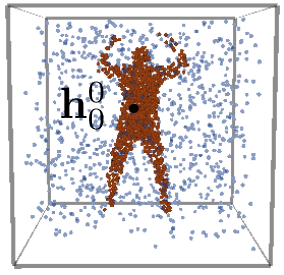
Coverage Loss



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

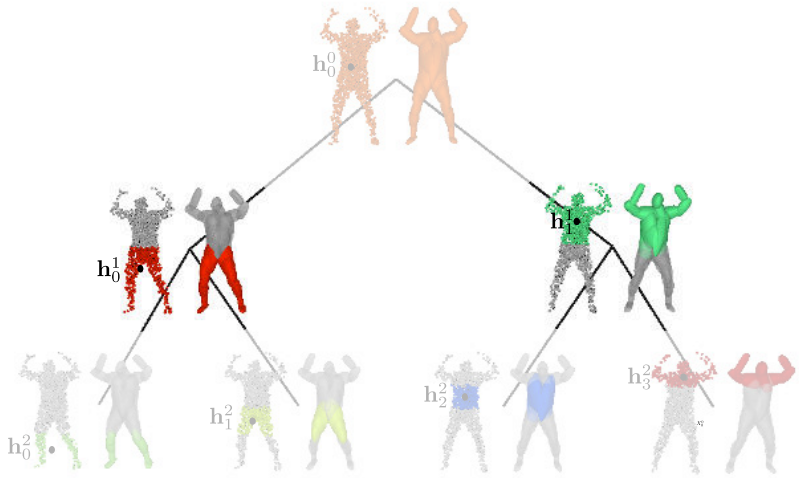
Structure Loss

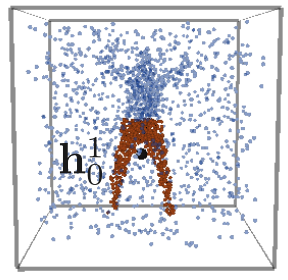




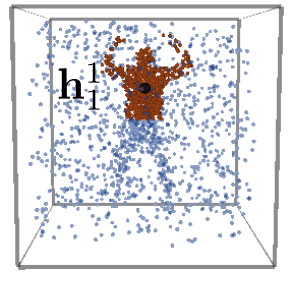
$\mathcal{X}_0^0$

Structure Loss



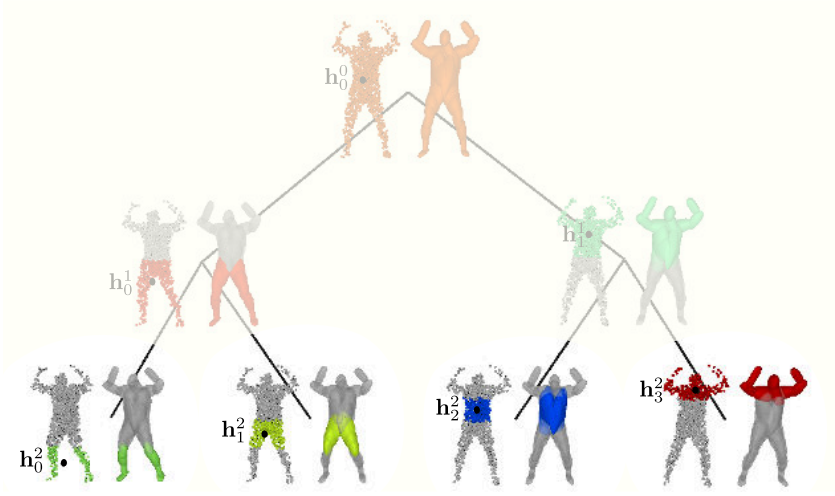


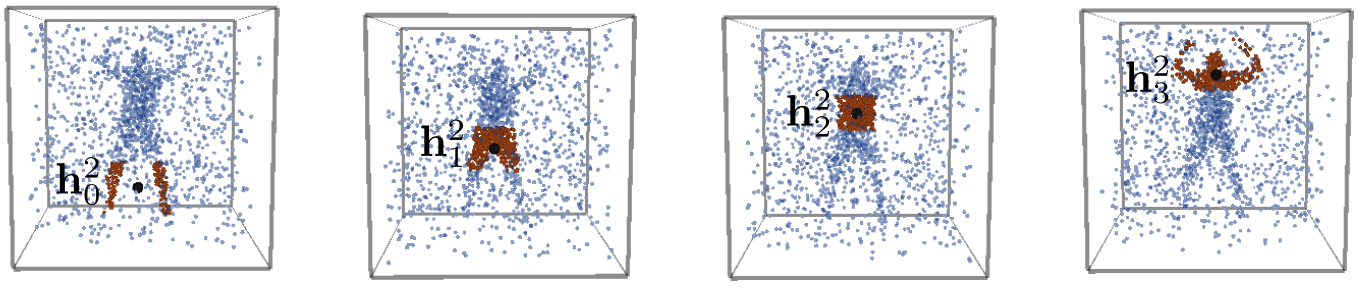
$\mathcal{X}_0^1$



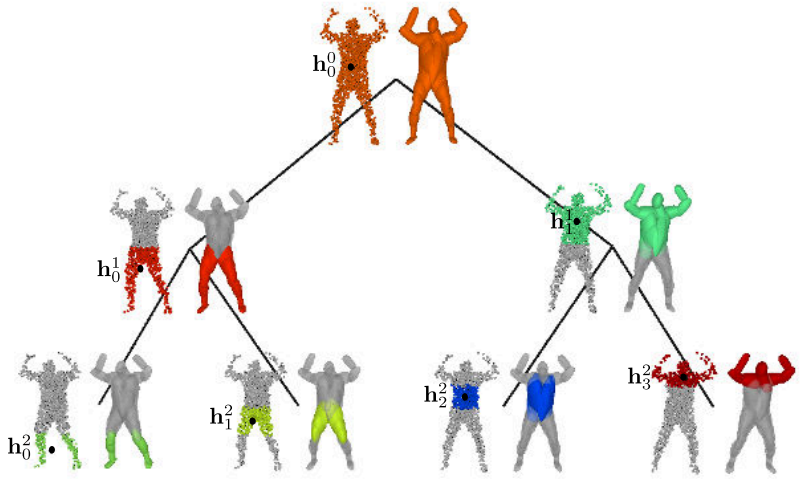
$\mathcal{X}_1^1$

Structure Loss





Structure Loss



$$\mathcal{L}_{str}(\mathcal{H};\mathcal{X}) = \sum_{h_k^d \in \mathcal{H}} \frac{1}{2^d - 1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} o \left\|\mathbf{x} - \mathbf{h}_k^d\right\|_2$$

Reconstruction Loss

$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$

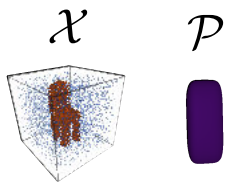
Reconstruction Loss

$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$

Reconstruction Loss

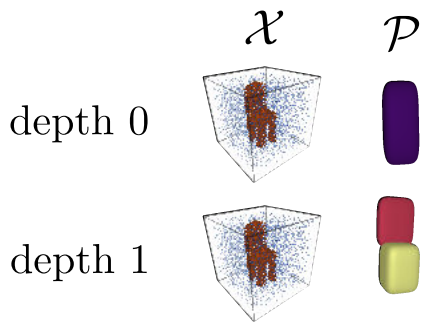
$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$

depth 0



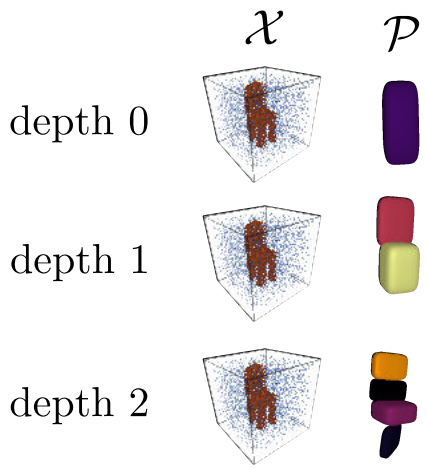
Reconstruction Loss

$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$



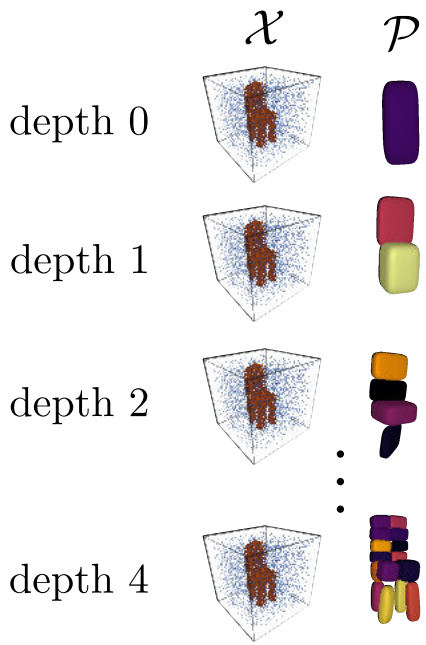
Reconstruction Loss

$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$



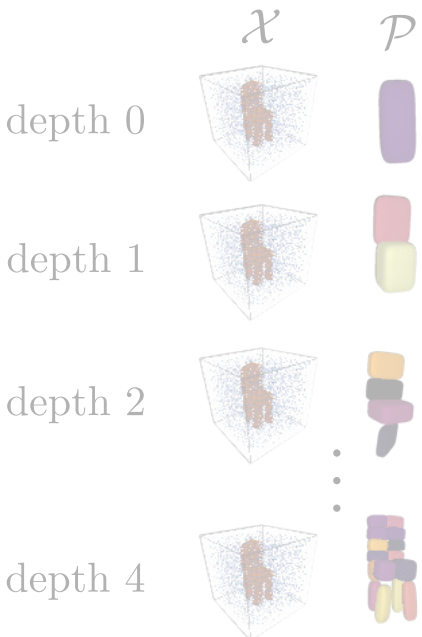
Reconstruction Loss

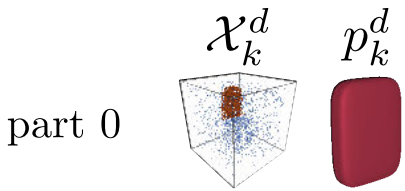
$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$



Reconstruction Loss

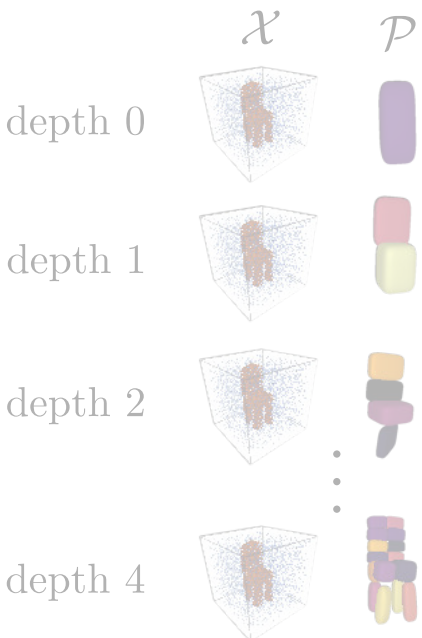
$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$

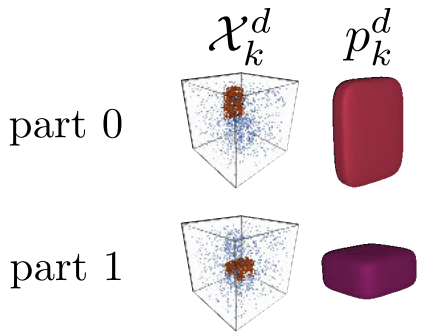




Reconstruction Loss

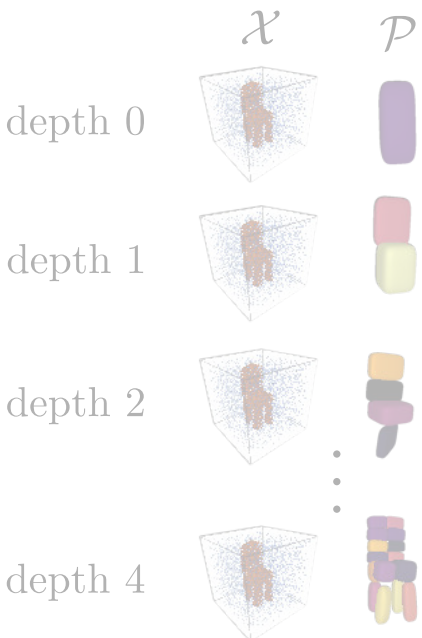
$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$

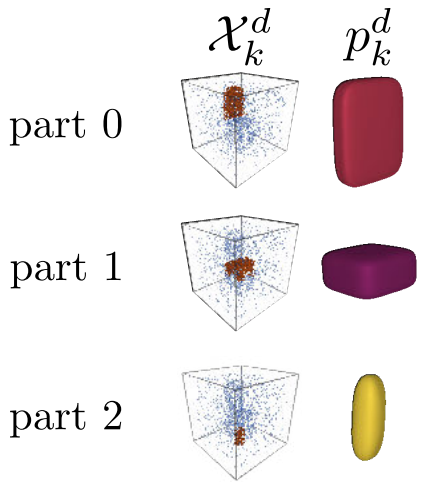




Reconstruction Loss

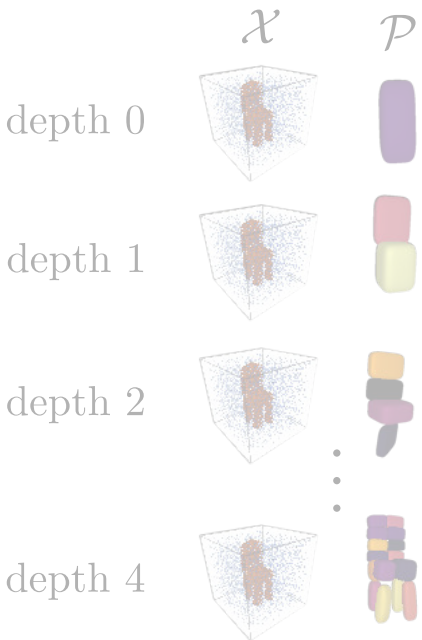
$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$

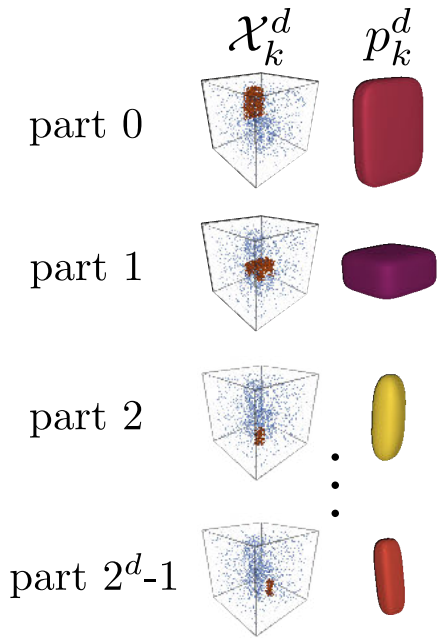




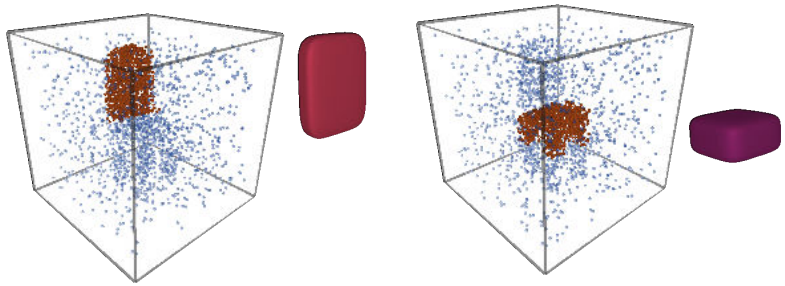
Reconstruction Loss

$$\mathcal{L}_{rec}(\mathcal{P};\mathcal{X}) = \underbrace{\sum_{(\mathbf{x},o)\in\mathcal{X}} \sum_{d=0}^{D} L\left(G^d(\mathbf{x}), o\right)}_{\text{Object Reconstruction}} + \underbrace{\sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \sum_{(\mathbf{x},o)\in\mathcal{X}_k^d} L\left(g_k^d\left(\mathbf{x};\lambda_k^d\right), o\right)}_{\text{Part Reconstruction}}$$





Compatibility Loss



$$\mathcal{L}_{comp}(\mathcal{P}) = \sum_{d=0}^{\mathcal{D}} \sum_{k=0}^{2^d-1} \left(q_k^d - \mathsf{IoU}(p_k^d, \mathcal{X}_k^d) \right)^2$$

Proximity Loss

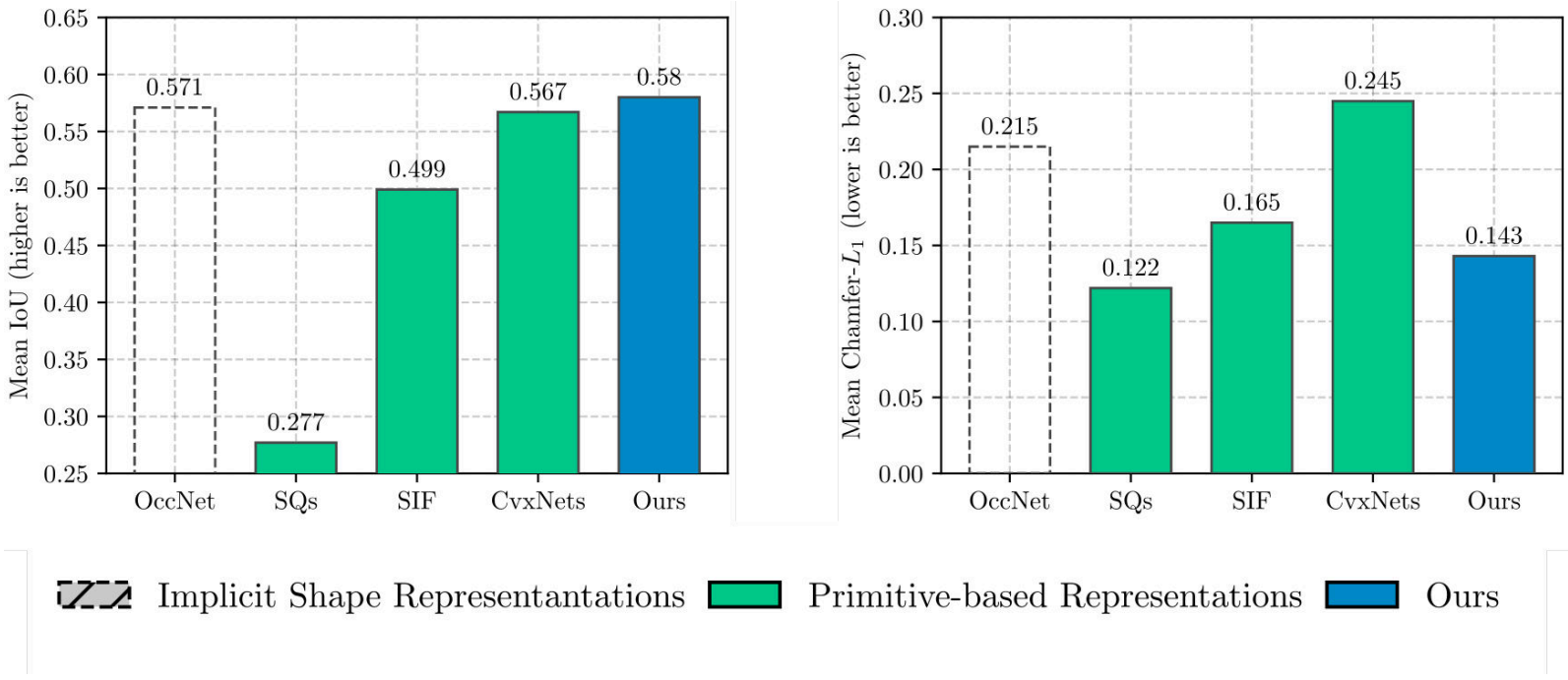
(a) **Input**

(b) **without**

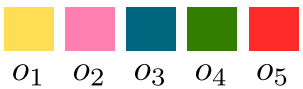
(c) **Ours**

$$\mathcal{L}_{prox}(\mathcal{P}) = \sum_{d=0}^{D} \sum_{k=0}^{2^d-1} \|\mathbf{t}(\lambda_k^d) - \mathbf{h}_k^d\|_2$$

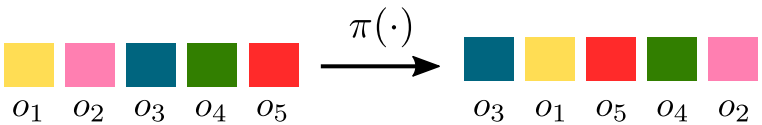
Single-view 3D Reconstruction on ShapeNet



Training Overview

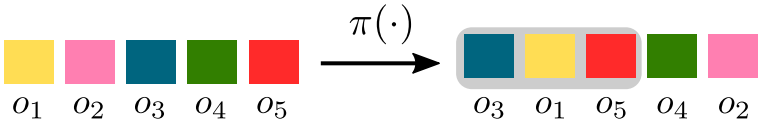


Training Overview



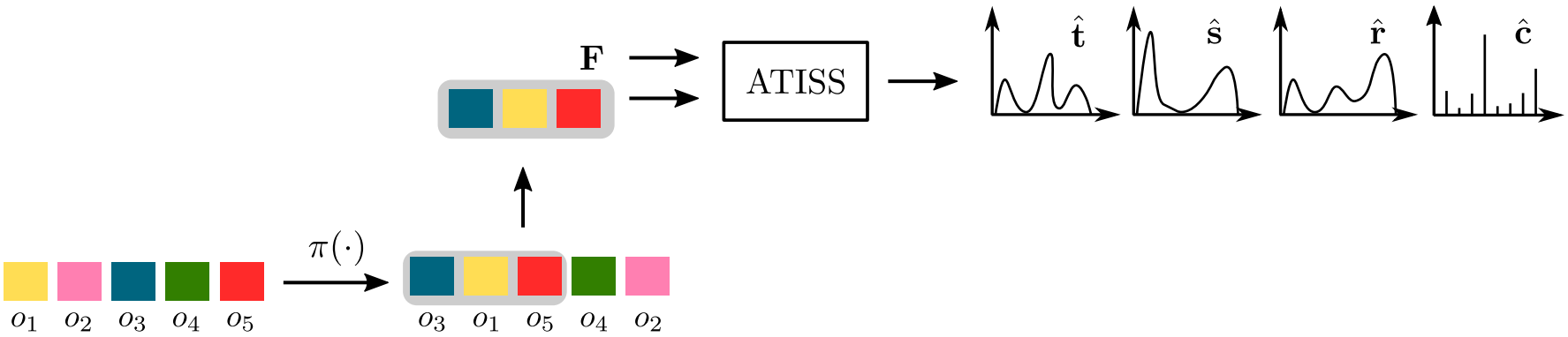
- Randomly permute the M objects of a scene.

Training Overview



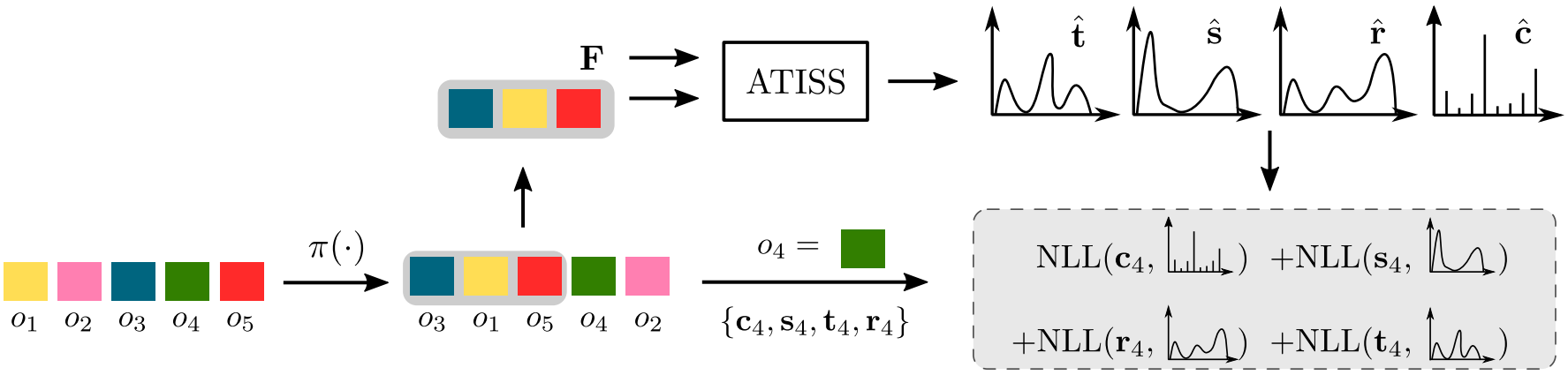
- Randomly permute the M objects of a scene.
- Randomly select the first T objects to compute the context embedding $\mathbf{C}$.

Training Overview



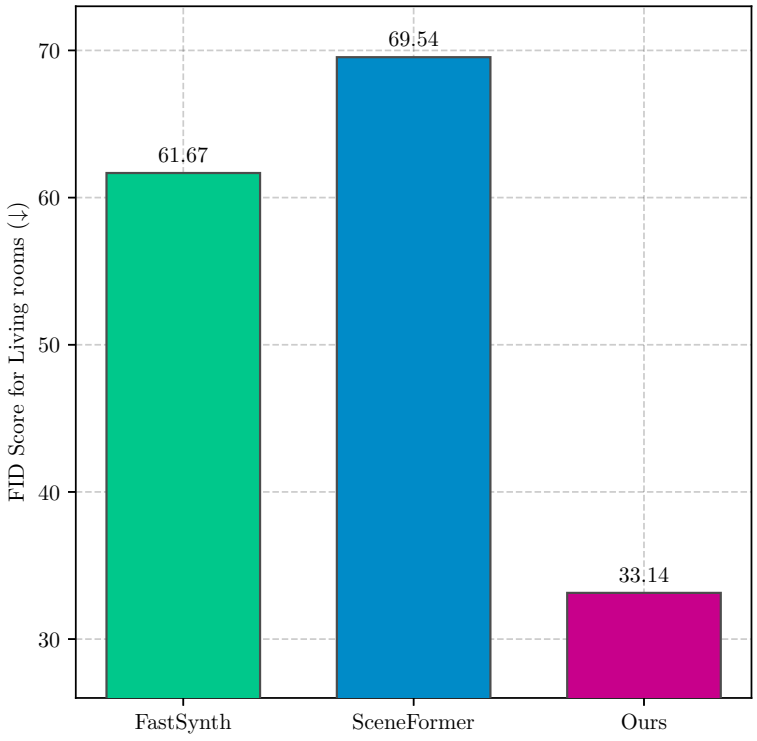
- Randomly permute the M objects of a scene.
- Randomly select the first T objects to compute the context embedding $\mathbf{C}$.
- Conditioned on the $\mathbf{C}$ and $\mathbf{F}$, ATISS **predicts the attribute distributions of the next object**.

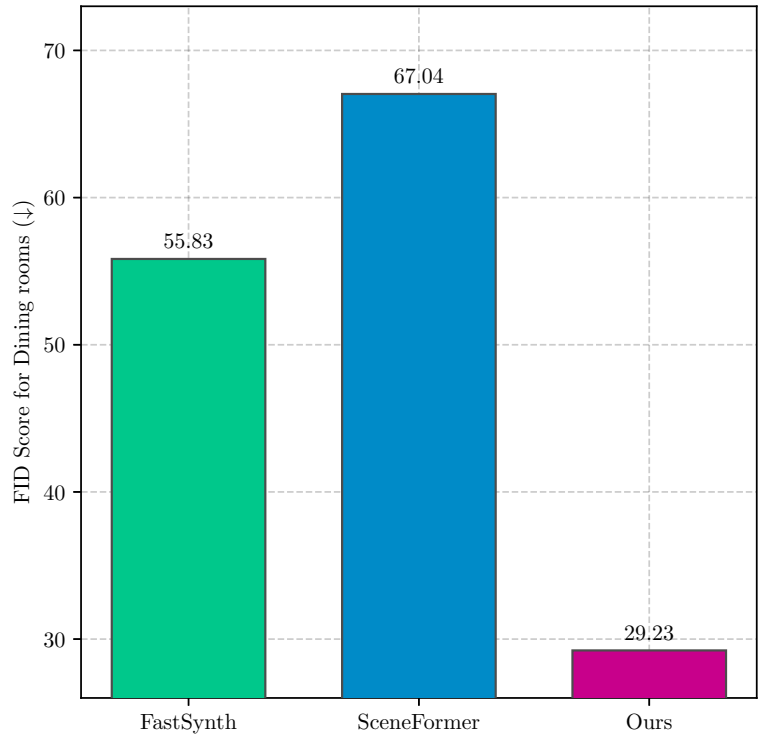
Training Overview



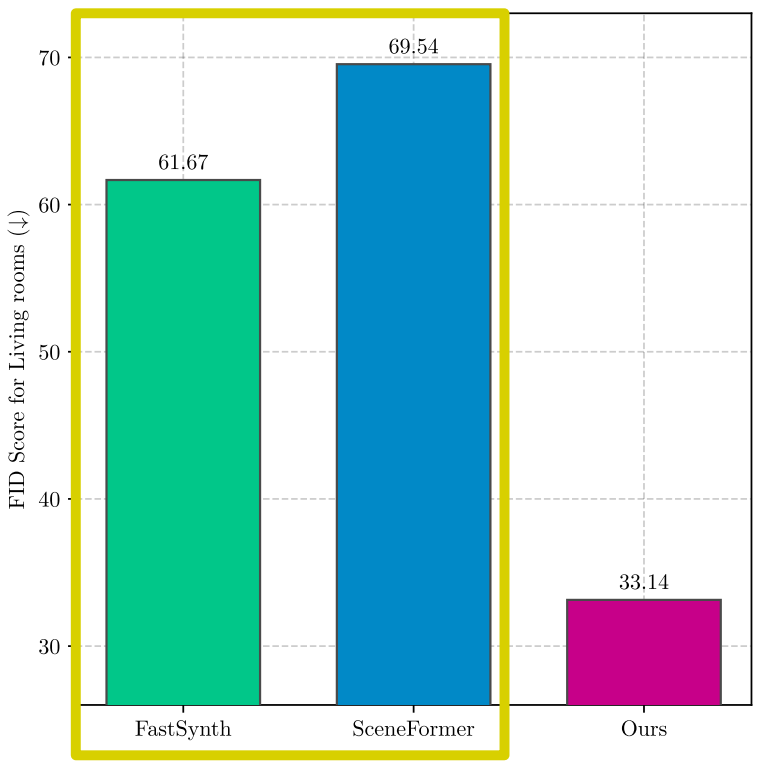
- Randomly permute the M objects of a scene.
- Randomly select the first T objects to compute the context embedding $\mathbf{C}$.
- Conditioned on the $\mathbf{C}$ and $\mathbf{F}$, ATISS **predicts the attribute distributions of the next object**.
- ATISS is trained to maximize the log likelihood of the $T+1$ object from the permuted set of objects.

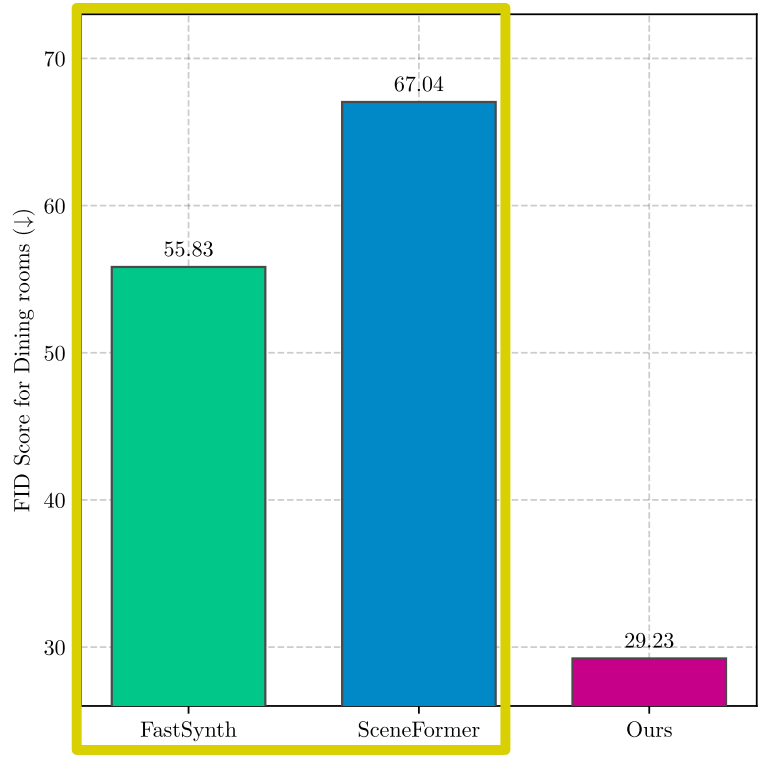
Scene Synthesis



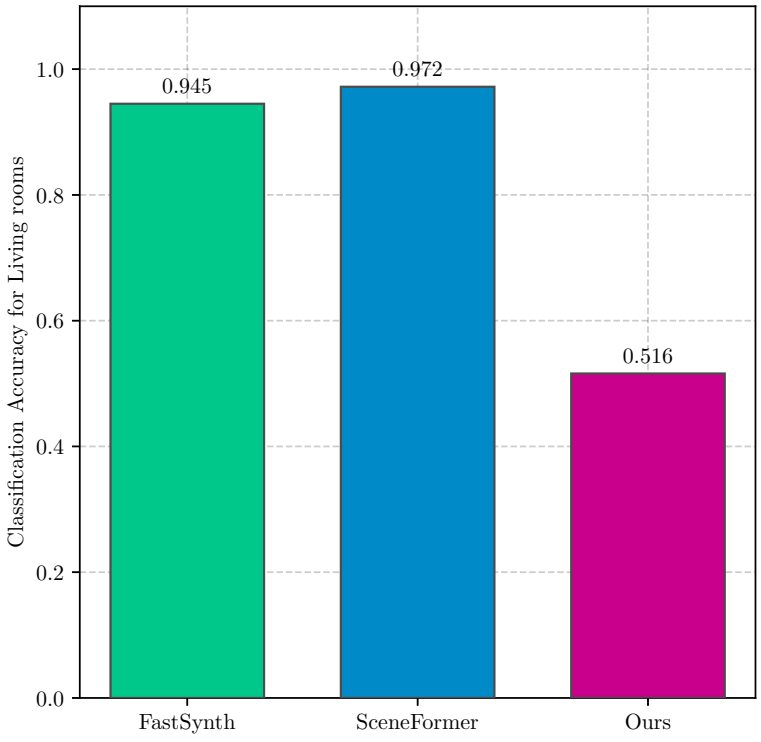


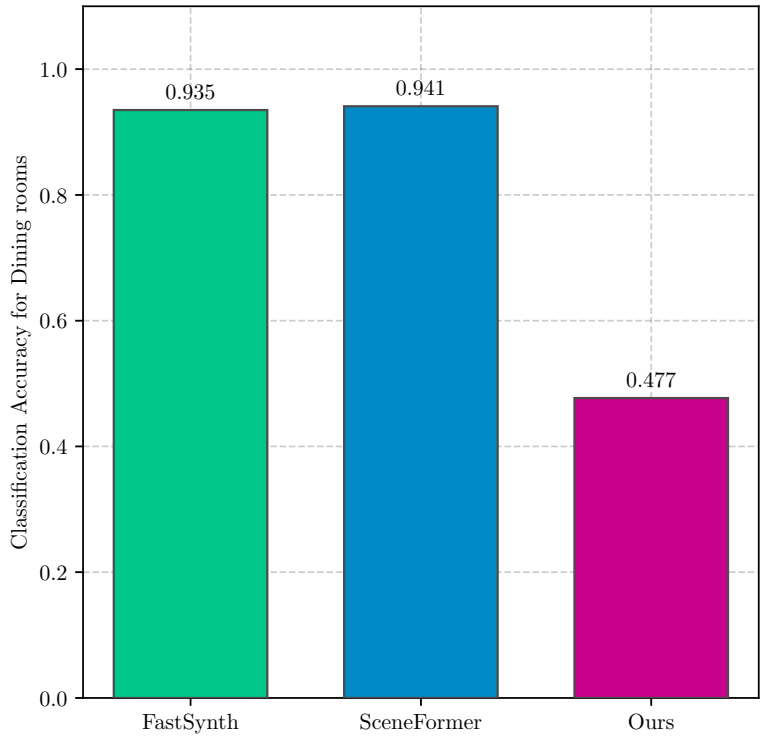
Scene Synthesis



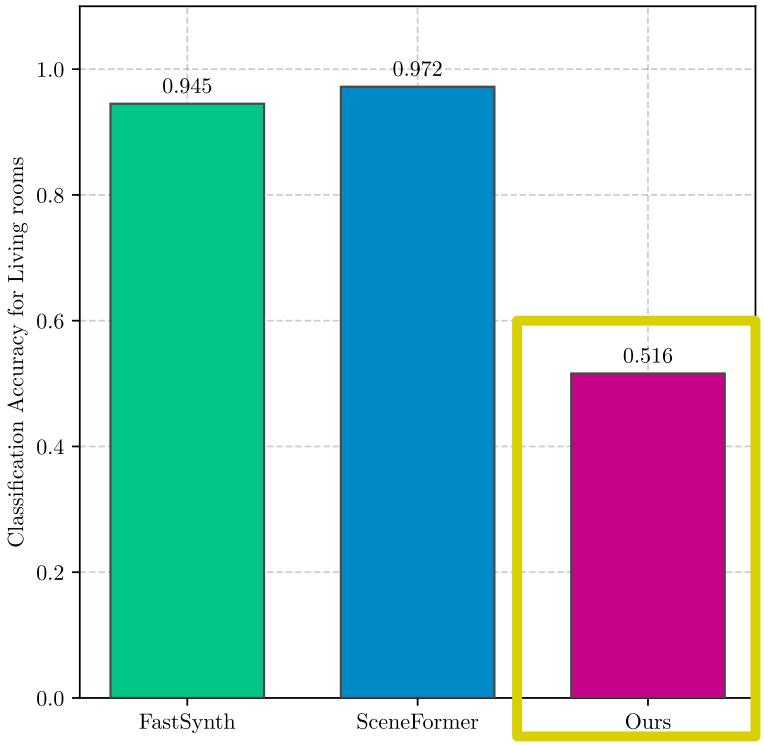


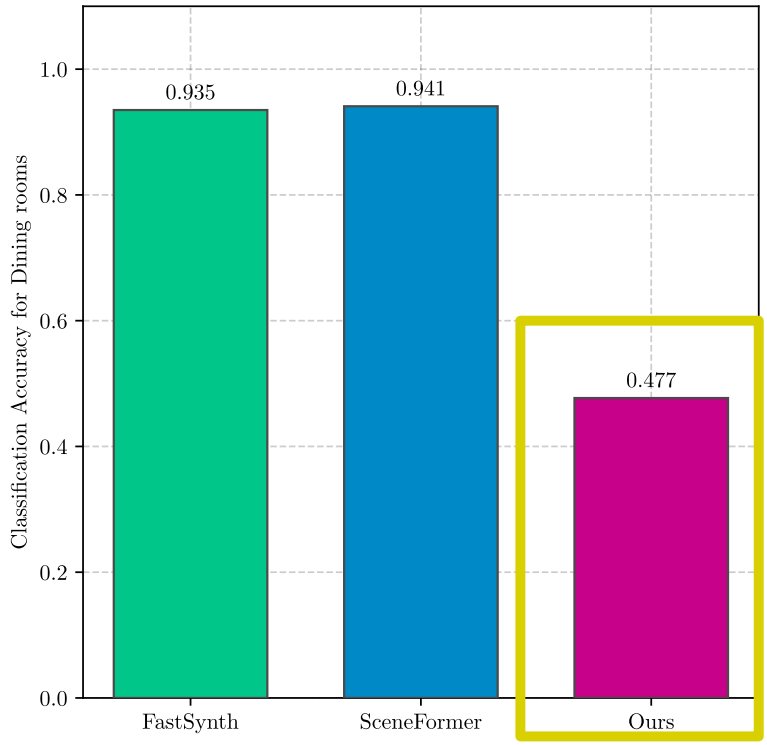
Scene Synthesis





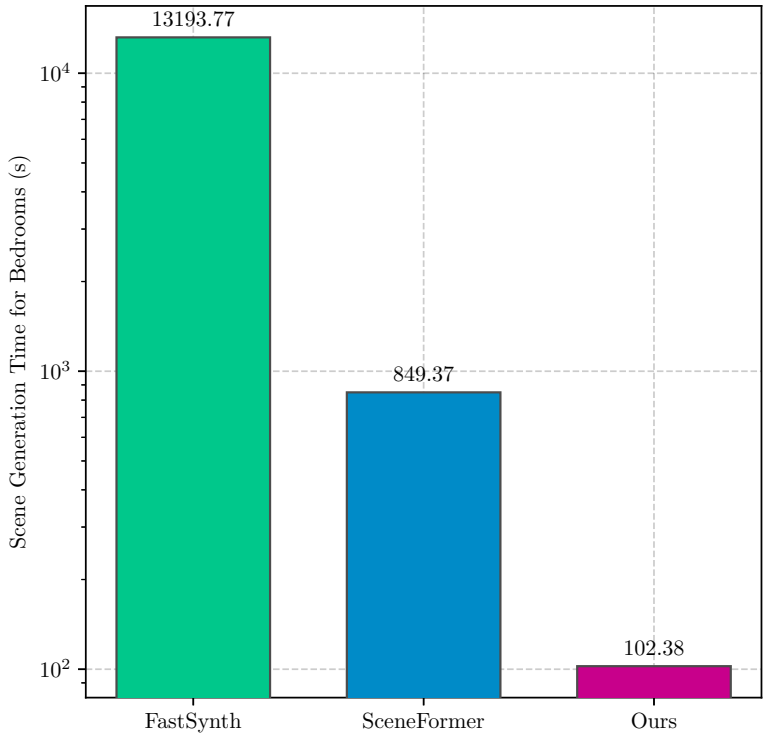
Scene Synthesis

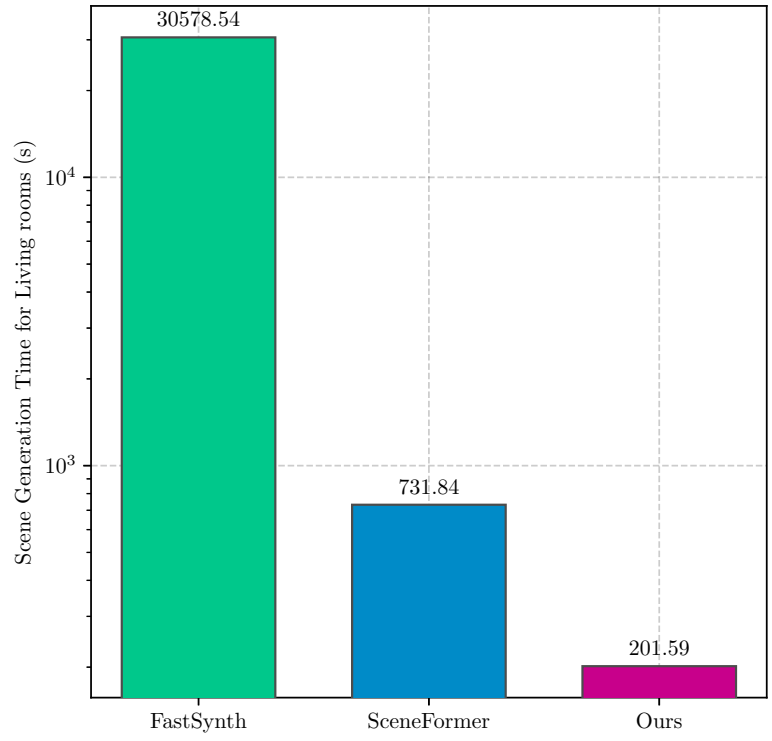




Scene Completion

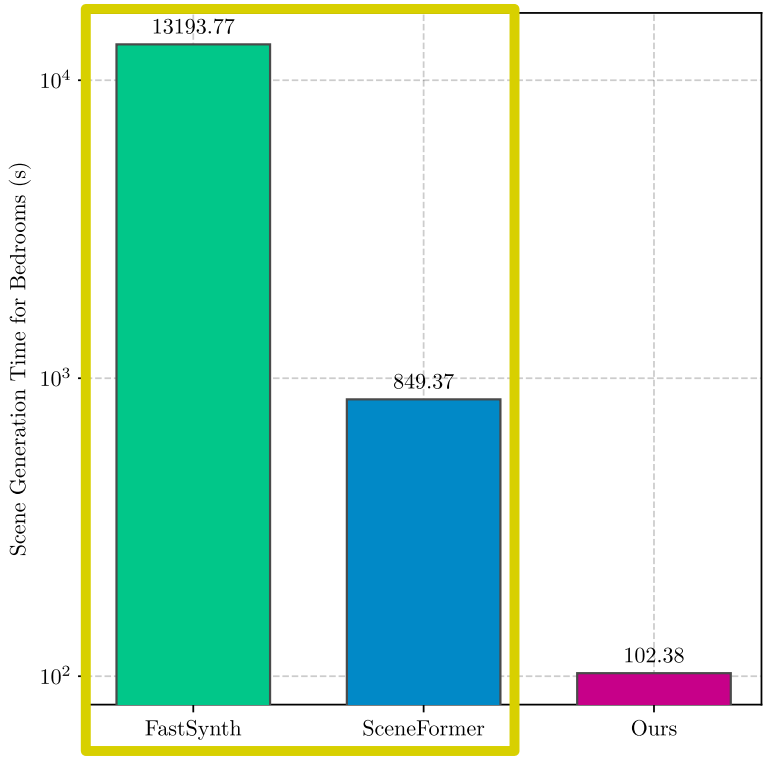
Generation Time

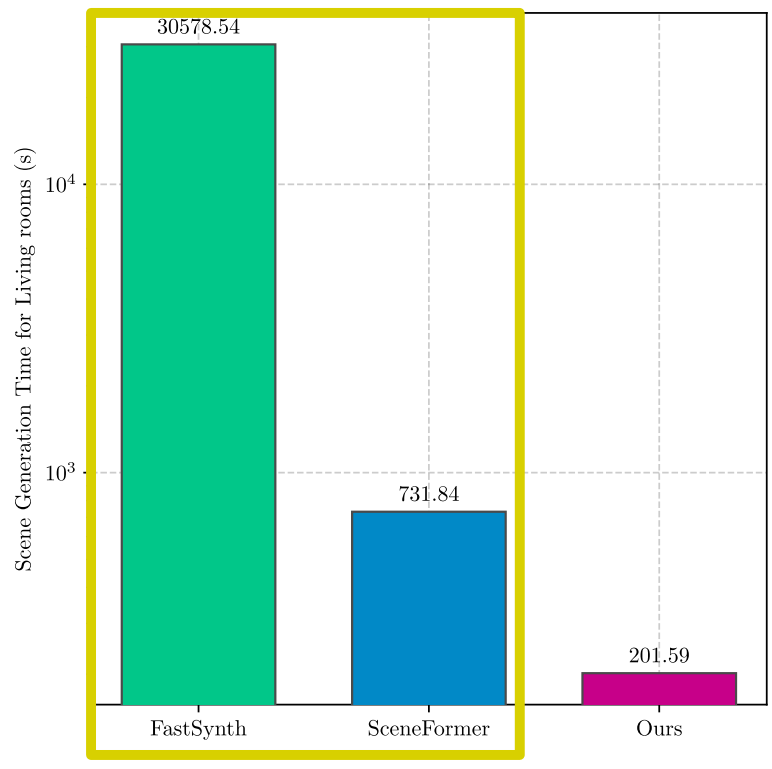




- At least $100\times$ faster than the CNN-based FastSynth for all room types.
- At least $4\times$ faster than the Transformer-based SceneFormer for all room types.

Generation Time





- 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.