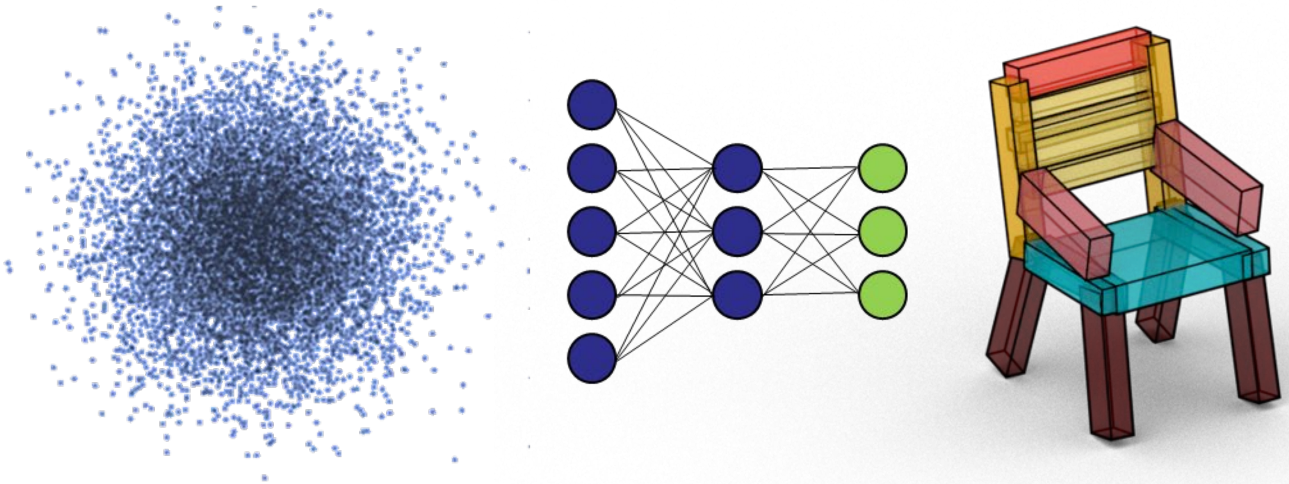
CS348n: Neural Representations and Generative Models for 3D Geometry



Leonidas Guibas
Computer Science Department
Stanford University

Leonidas Guibas Laboratory

Geometric Computing

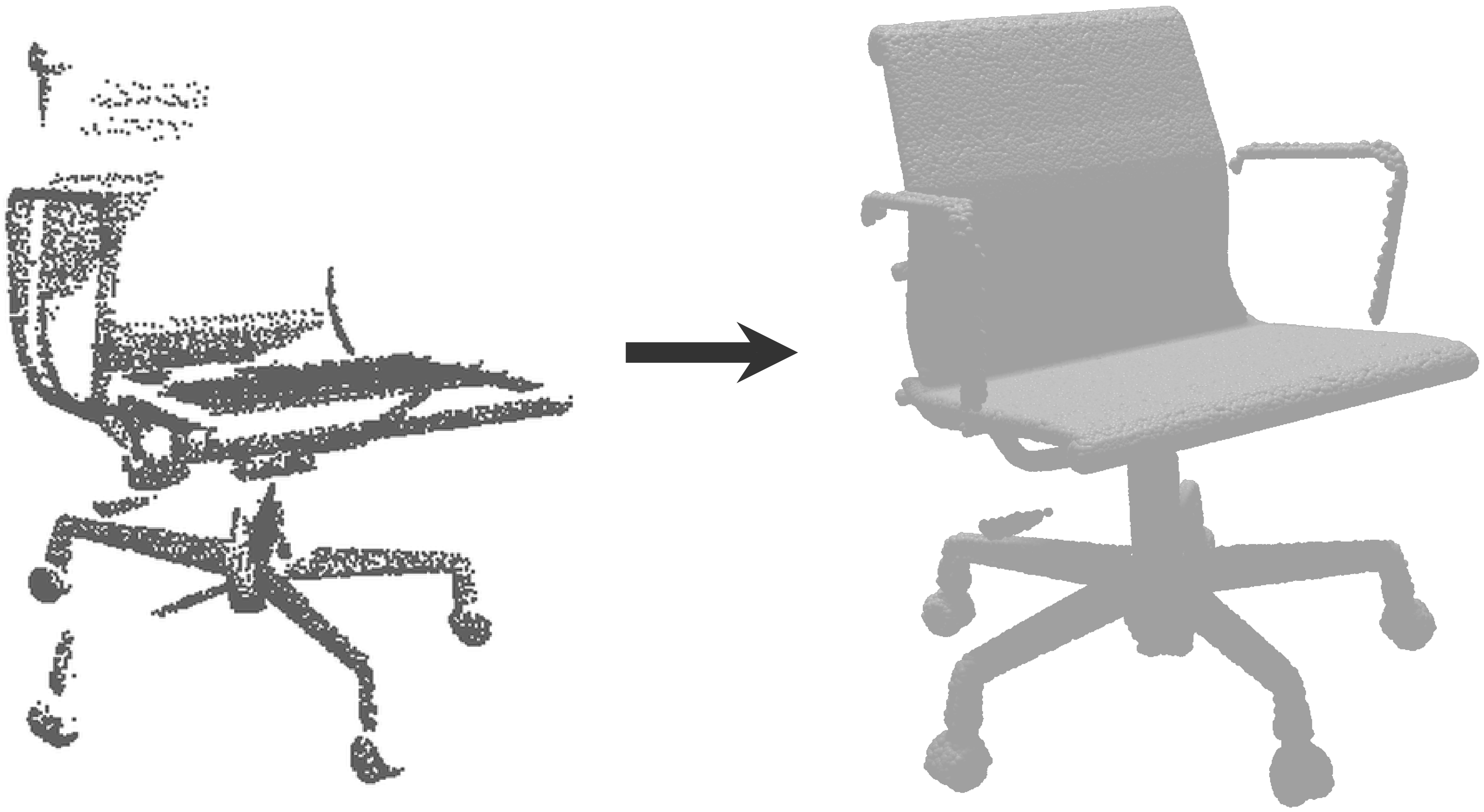
Project Proposal: One Page

- Title and participant names (up to three)
- Brief description of the project goal
  - Relation to class topics
  - Relation to other research projects of the participants (if any)
- Specific experiments or investigations to be conducted
- Evaluation metrics

Last Time: Conditional Shape Generation Based on 3D Data

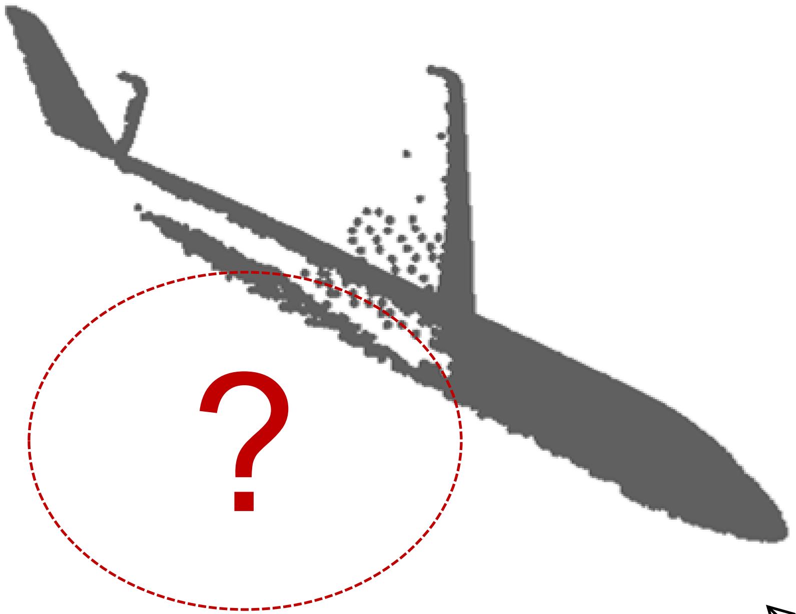
Goal 1: Scan Completion

- Complete or re-generate shape from a single view scan

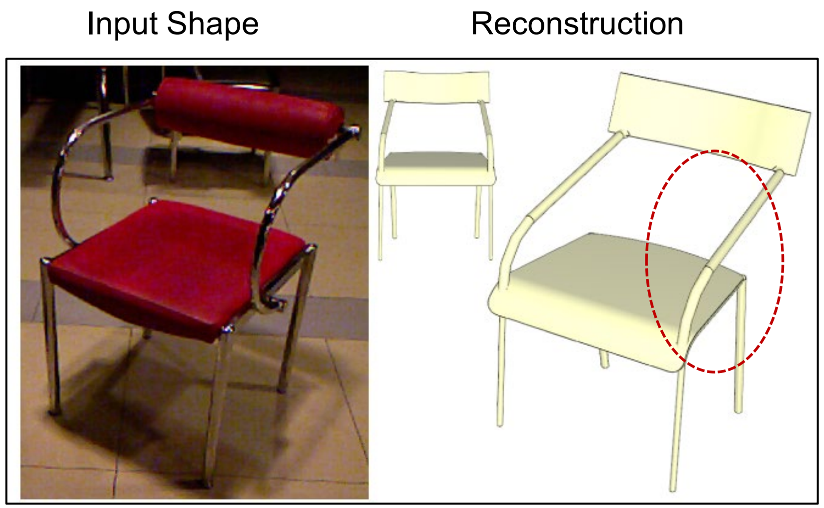


Symmetry & Priors: Each by Itself Has Difficulties

- Symmetry-based
  - Hard to predict from *partial* data.



- Data-based (Priors)
  - Hard to recover the *exact* shape.

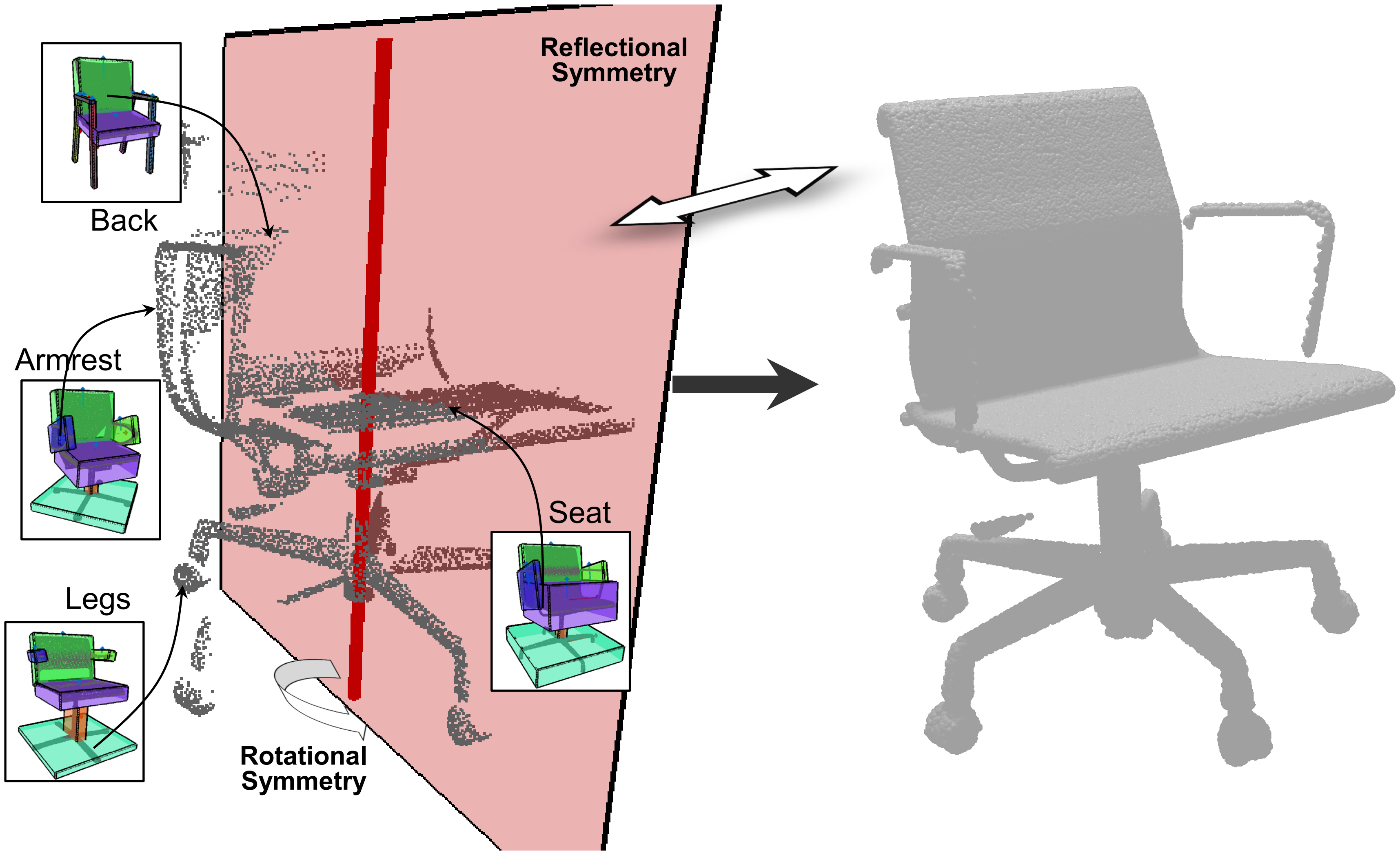


[Shen et. al. 2012]

Complementary!

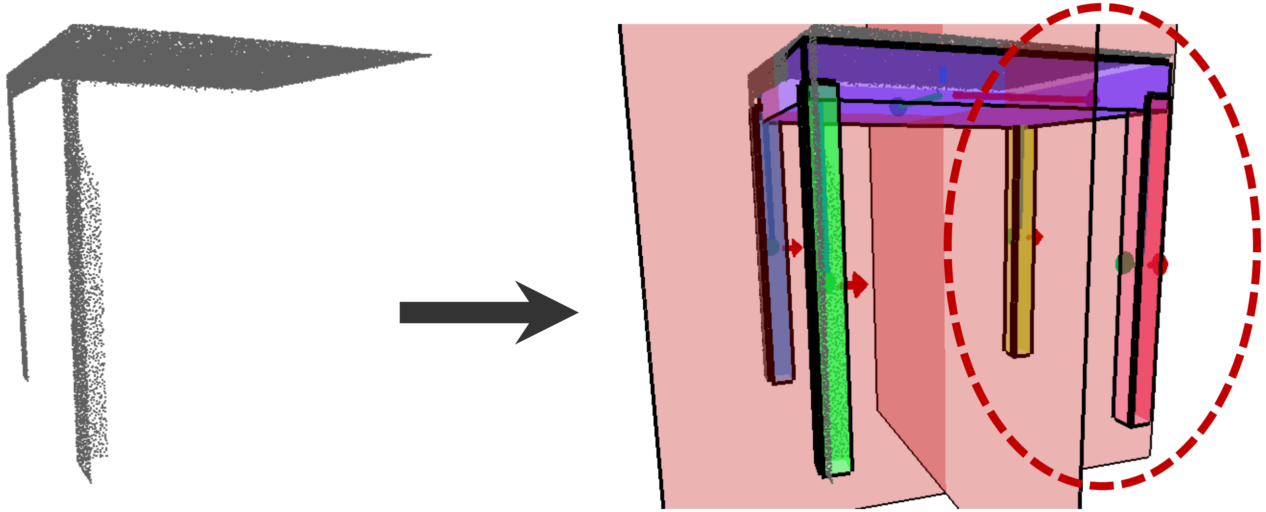
Completion Using Symmetry and Priors (Database)

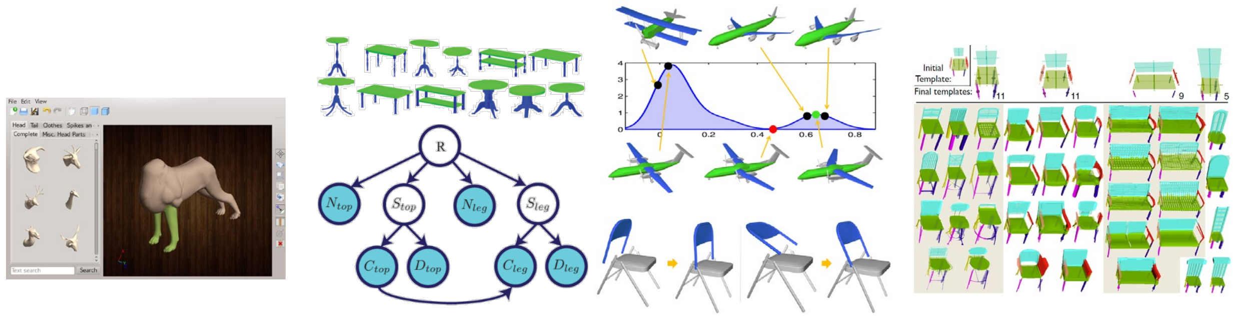
- Combine both symmetry and database sources.



Approach

- Predict *missing* parts based on *part relations*.



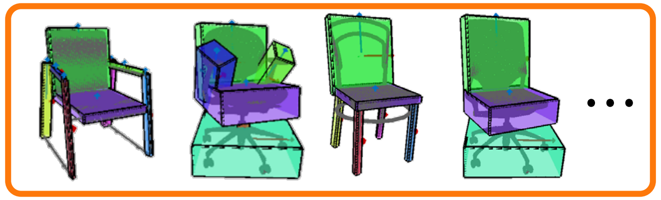


[Chaudhuri et. al. 2011] [Kalogerakis et. al. 2012] [Fish et. al. 2014] [Kim et. al. 2013]

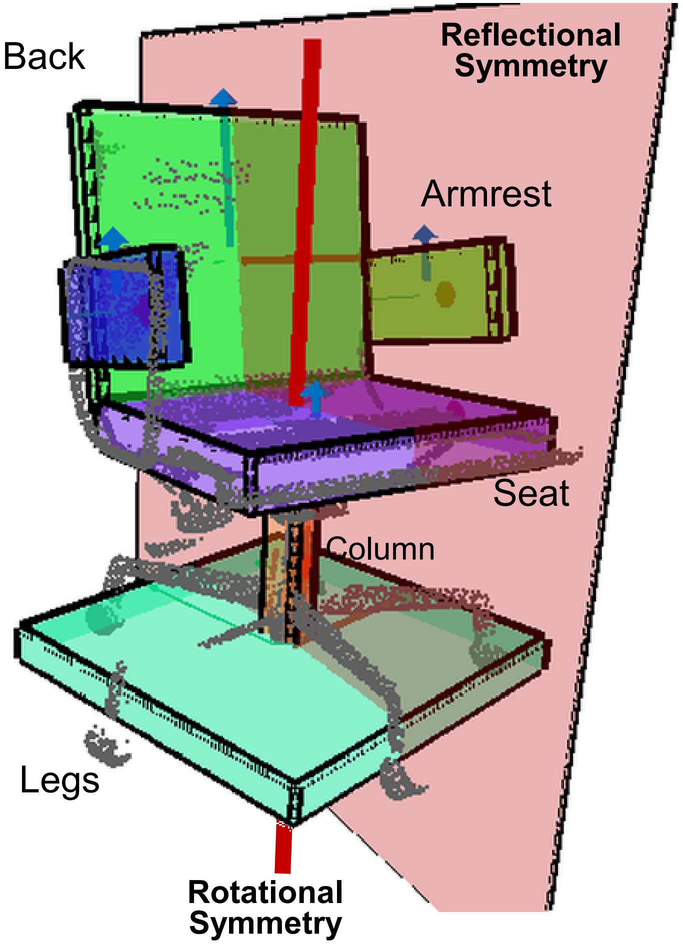
Earlier efforts analyze ***complete*** *shapes only*

Approach

- Estimate part and symmetry structure from the *partial* scan data using data-driven *priors*.

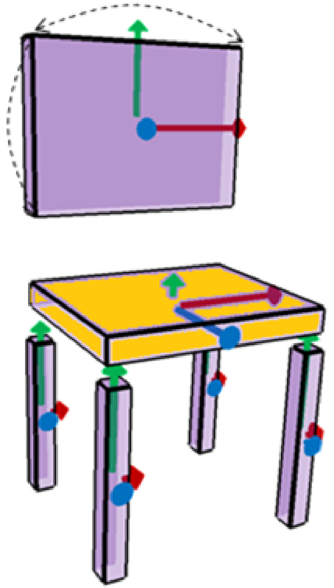


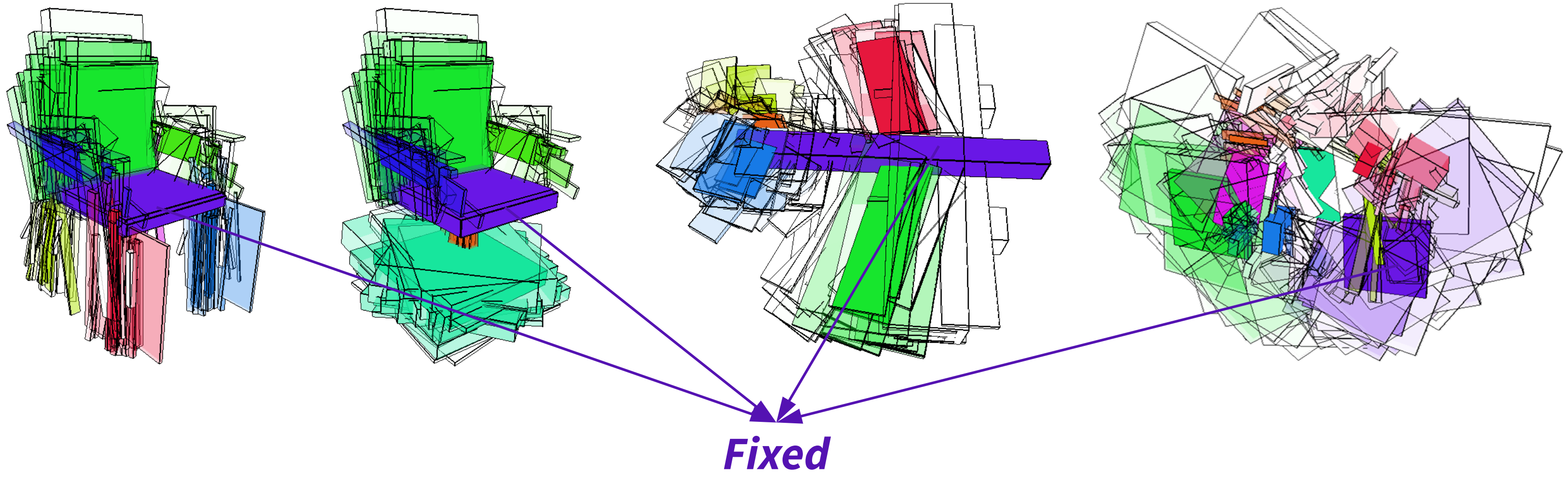
Training Data



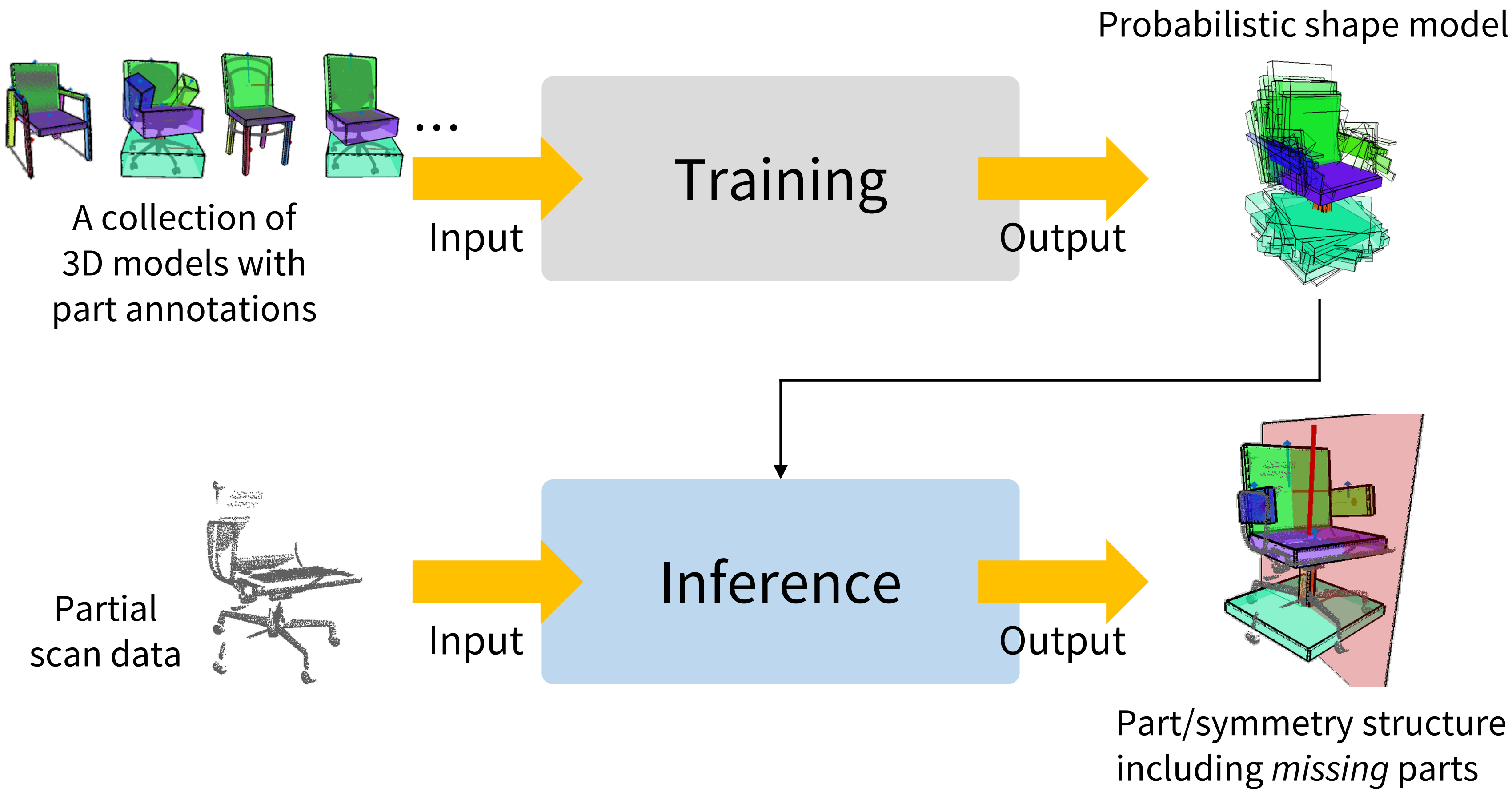
Probabilistic Part Relations

- Part parameters
  - Local coordinates + Scale
- Pairwise relations
  - Gaussian distributions of ***relative*** pose, height, and scale

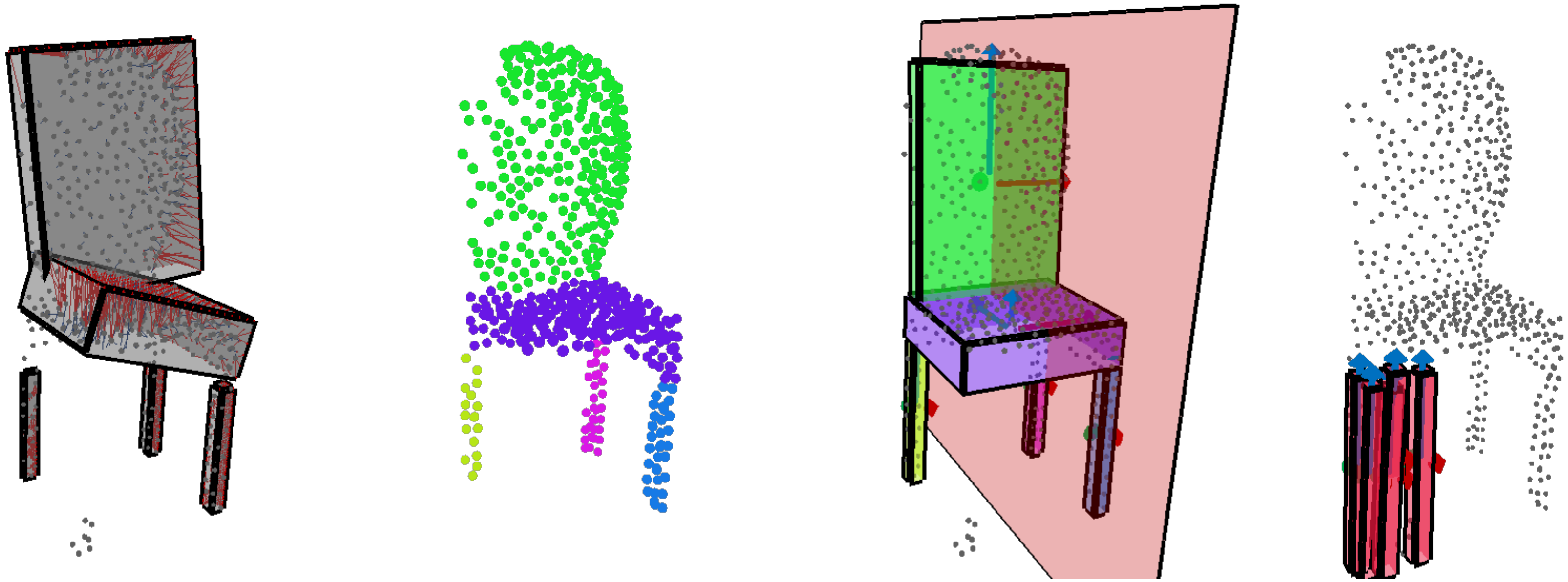




The Pipeline



Inference



Segmentation

Labeling

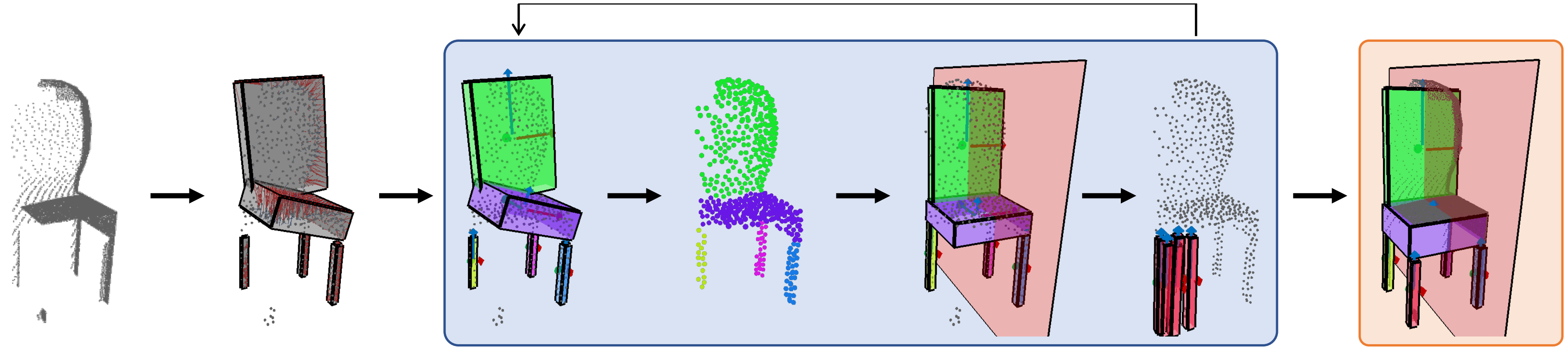
Structure Estimation

Missing Parts Prediction

Discrete

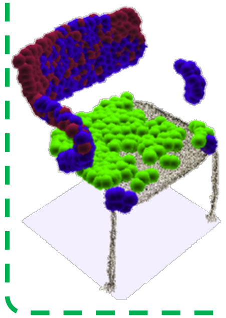
Continuous

Inference Time

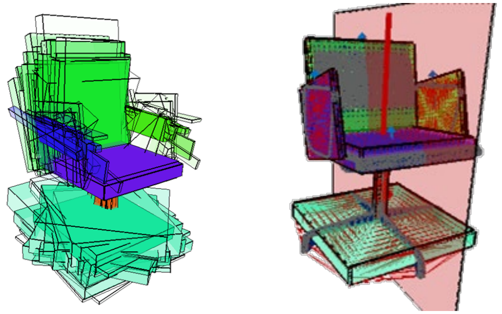


Energy function

$$\boldsymbol{E} = E_{pnt} + E_{smooth} + E_{SMD} + E_{pair} + E_{symm}$$

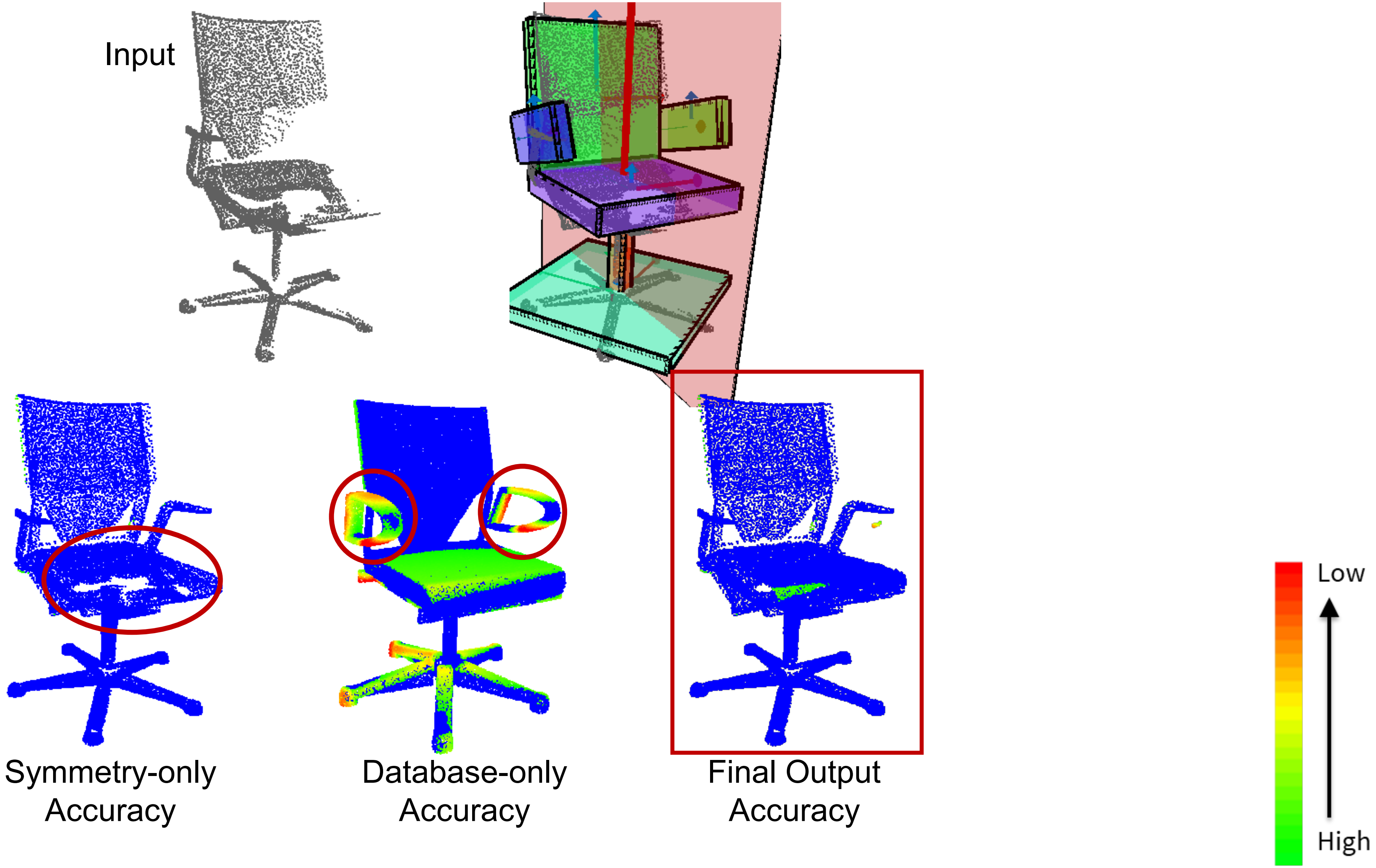


Point-level



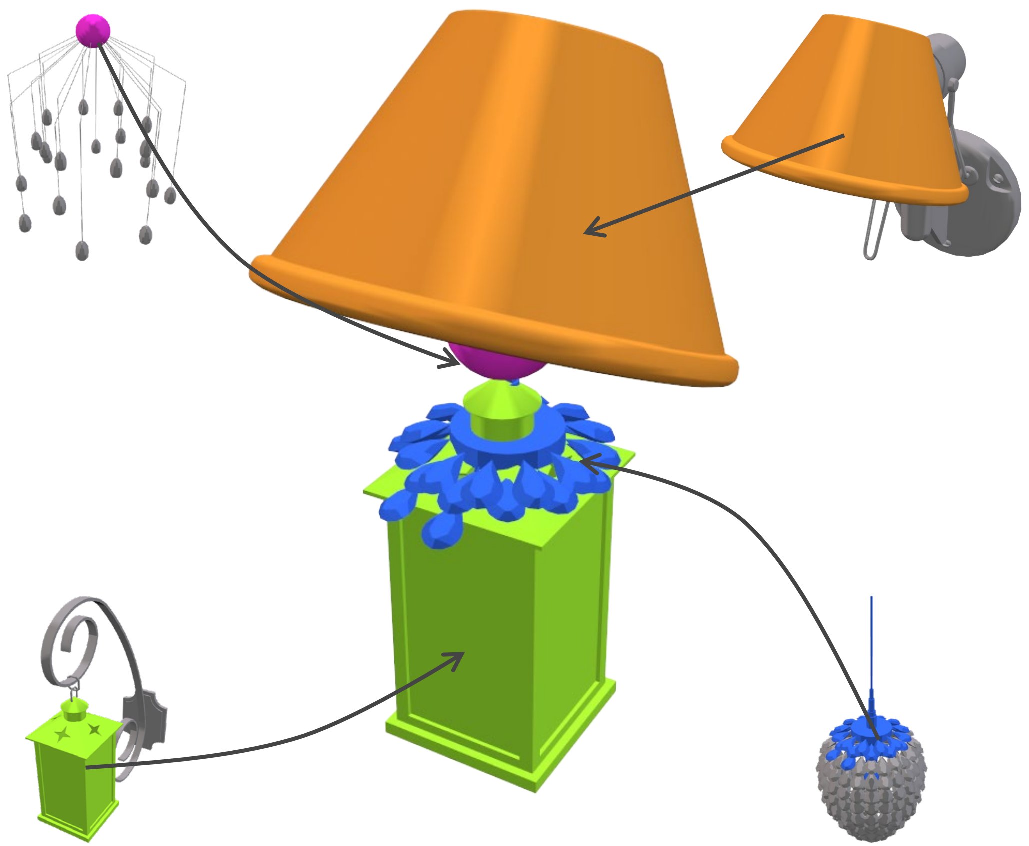
Part-level

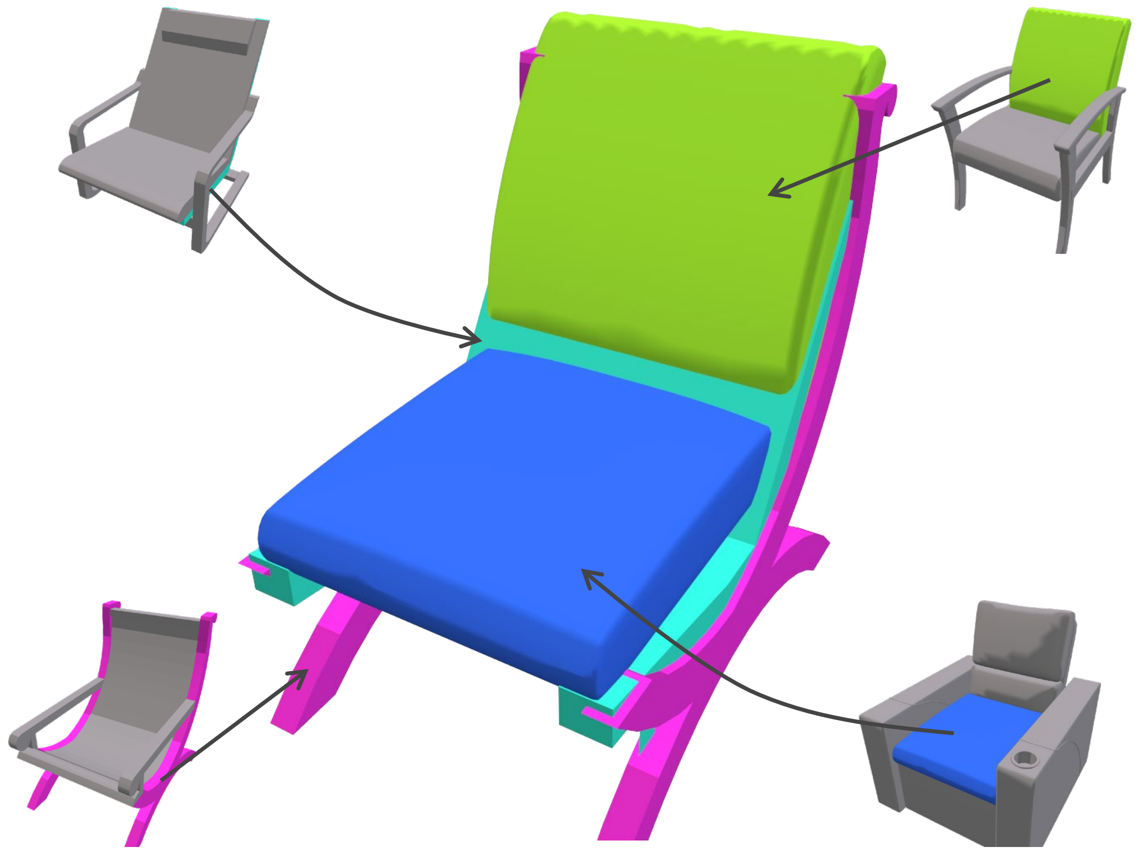
Comparison


Input
Symmetry-only
Accuracy
Database-only
Accuracy
Final Output
Accuracy
Low
High

Goal 2: Composition-Based Modeling

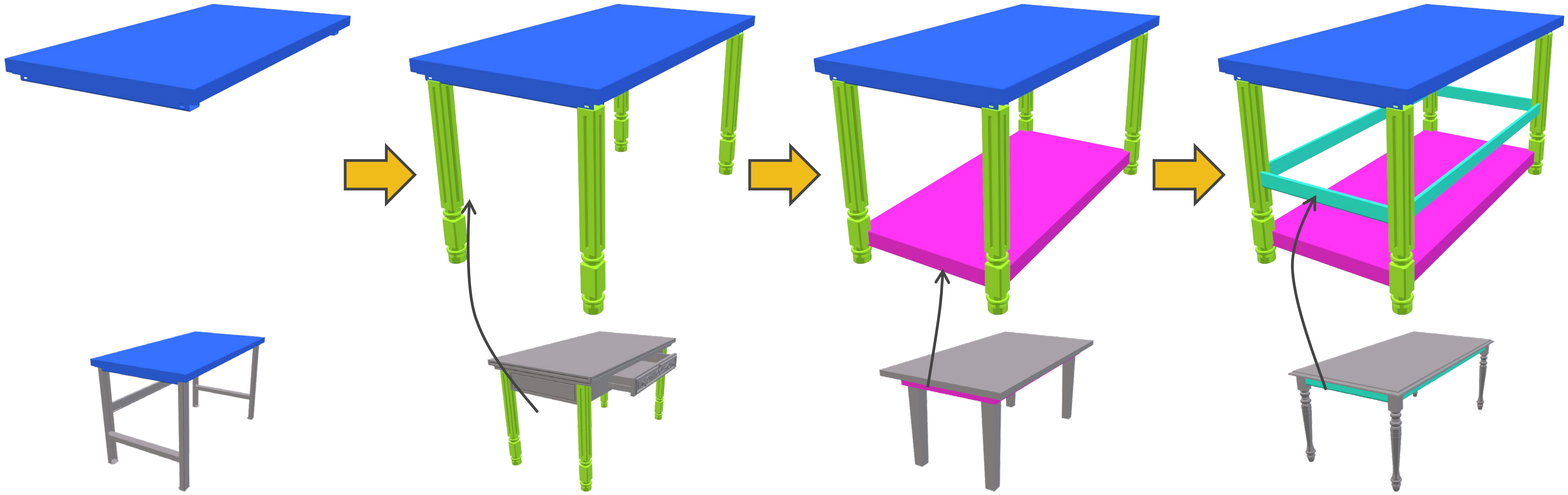
Create a shape by assembling components of 3D models in a large-scale repository.





Composition-Based Modeling

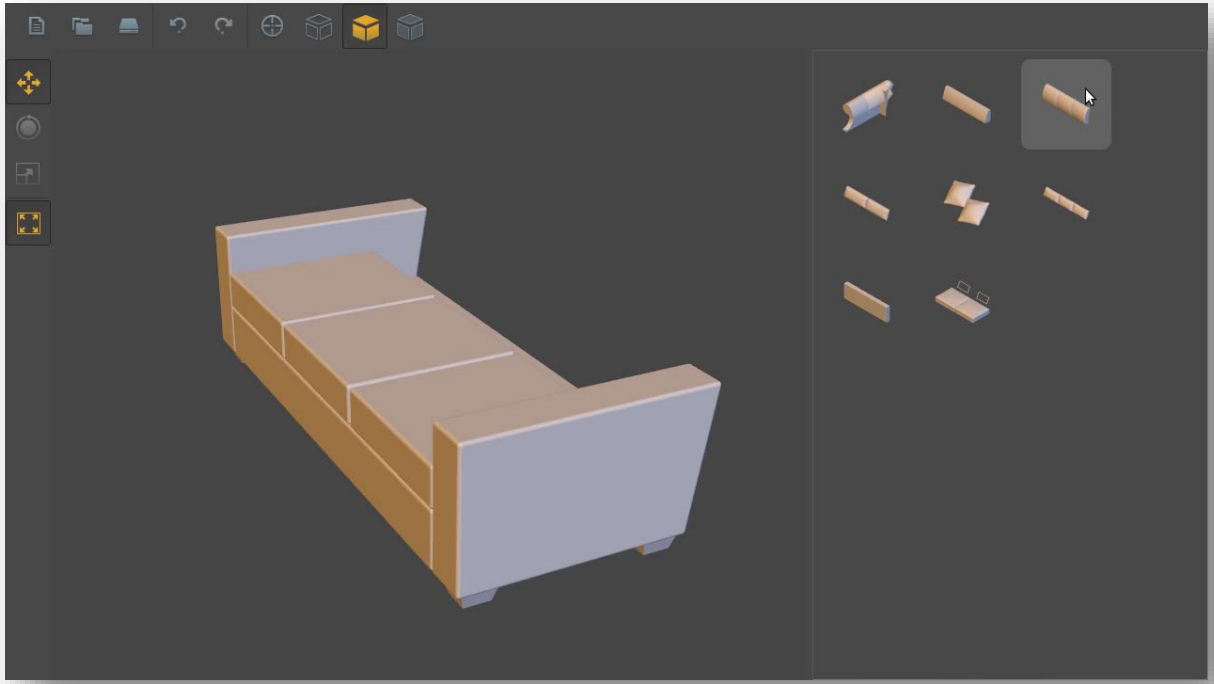
- Propose an iterative *assembly* system.
- Suggest *complementary* parts and their locations at each time.

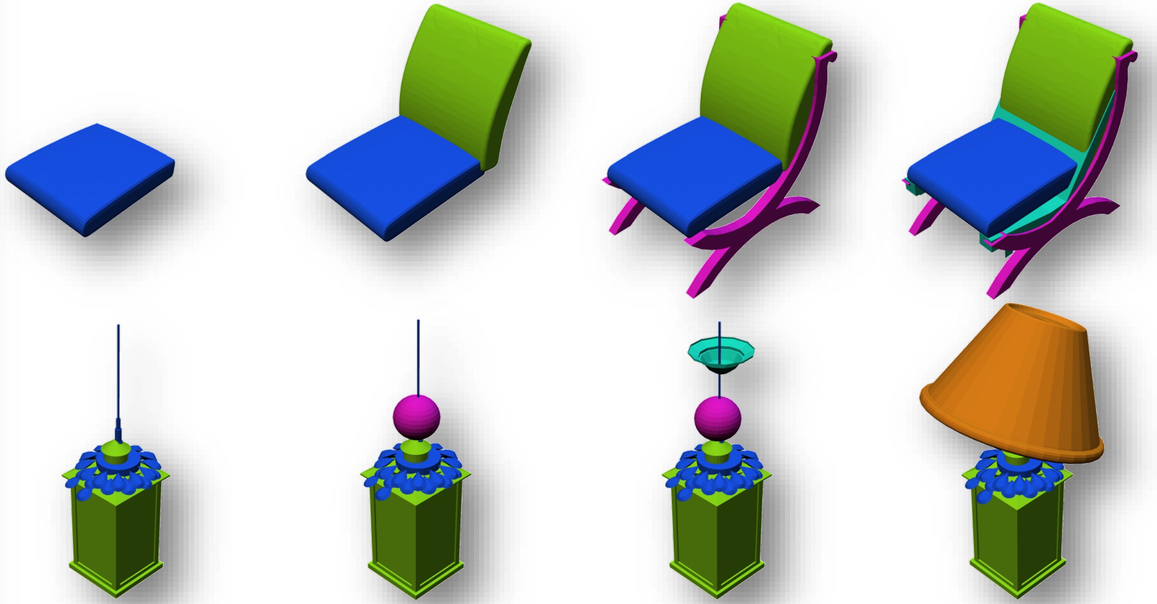


Composition-Based Modeling

Interactive design interface

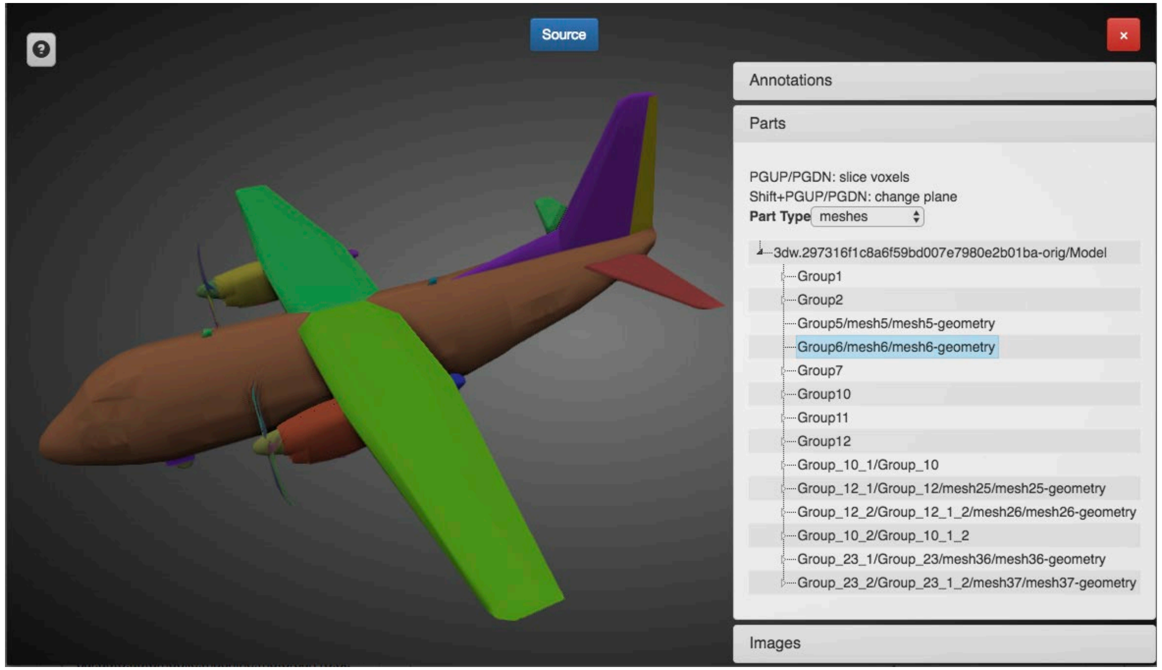
Automatic shape synthesis



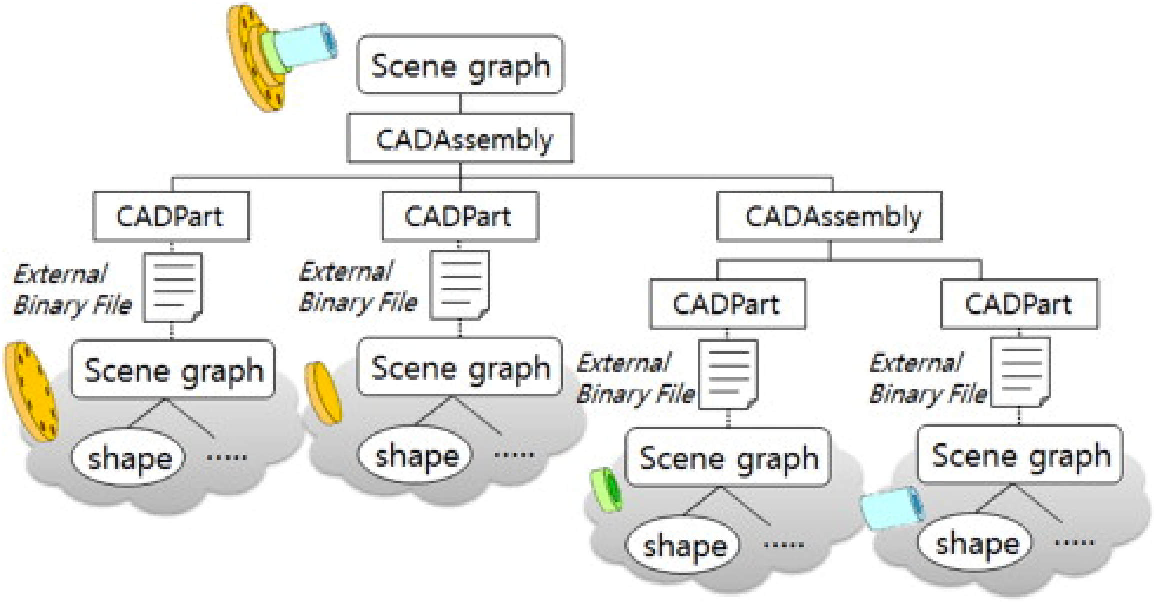


Use Data Sets in the Wild

CAD data include *scene graphs*:
Part geometry + Hierarchical structure



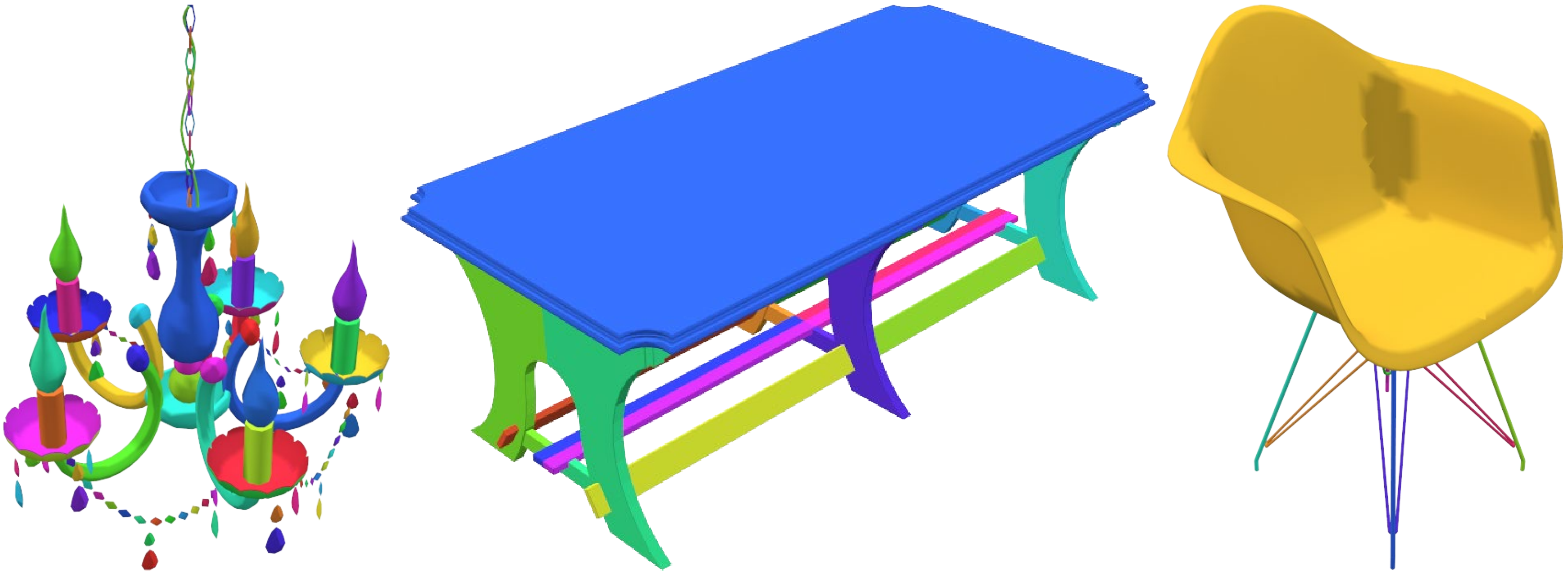
ShapeNet



Kim et al., 2015

Observations

(+) Provides natural part segmentations.



Observations

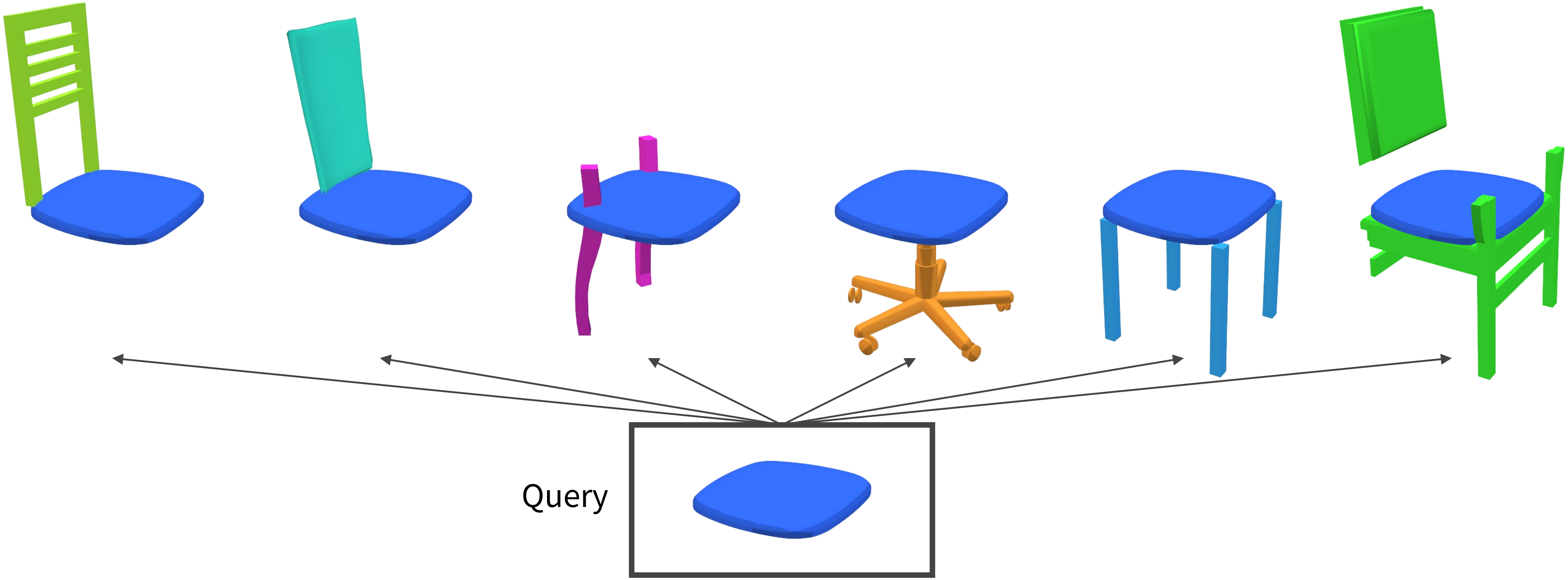
(+) Provides natural part segmentations.

(–) Inconsistent and unlabeled.

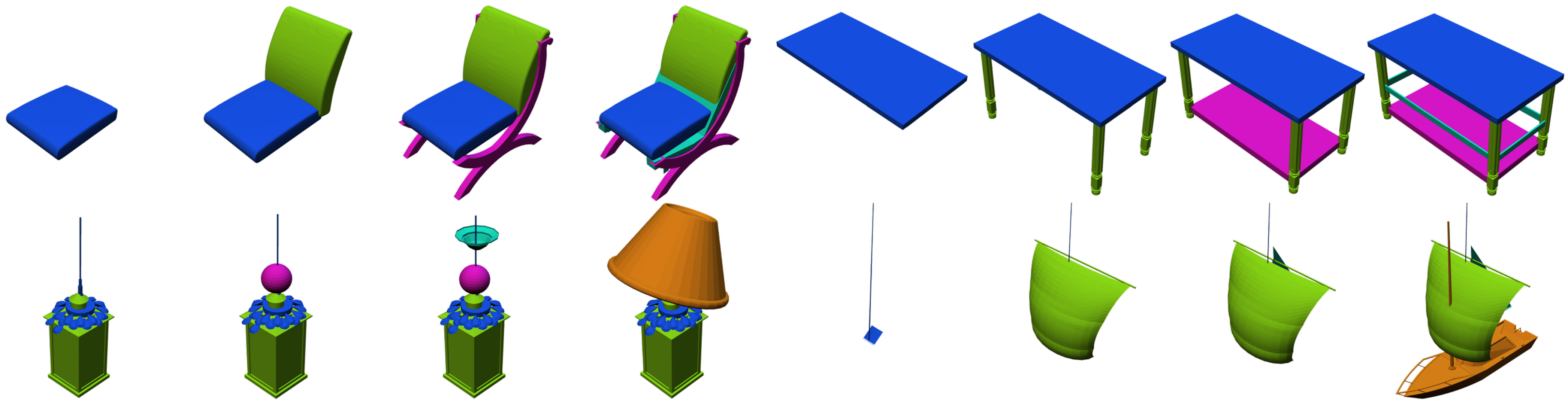
Goal

Predict complementary parts

using only geometric information

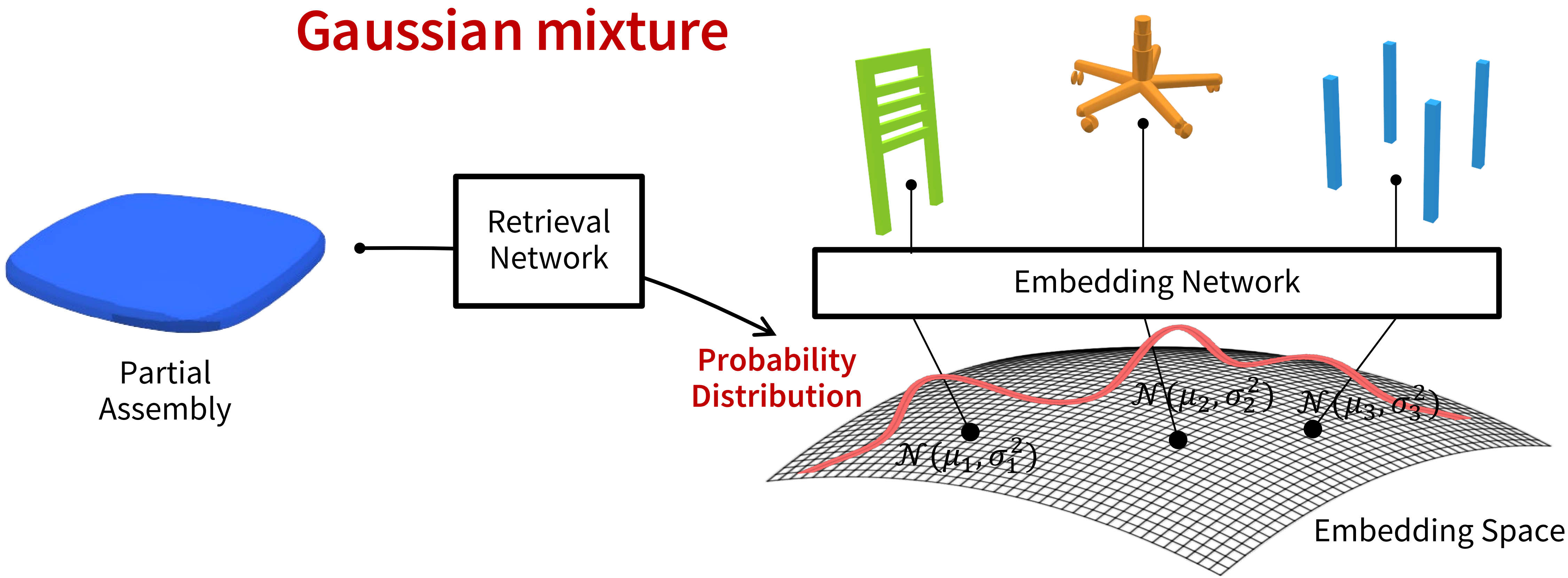


ComplementMe: Weakly-Supervised Component Suggestions

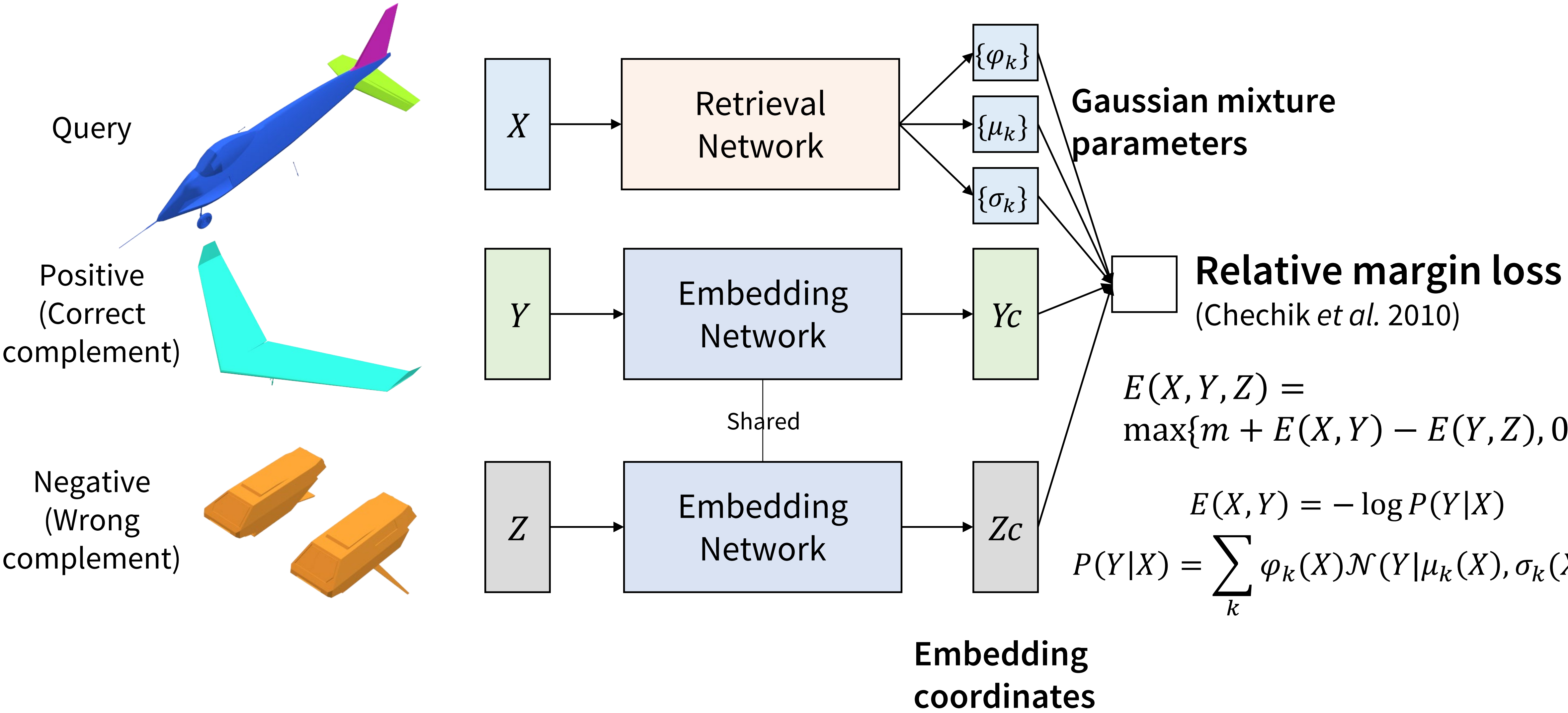


Retrieval and Embedding

Predict a **multimodal** probability distribution (Bishop 1994).



Neural Network Training



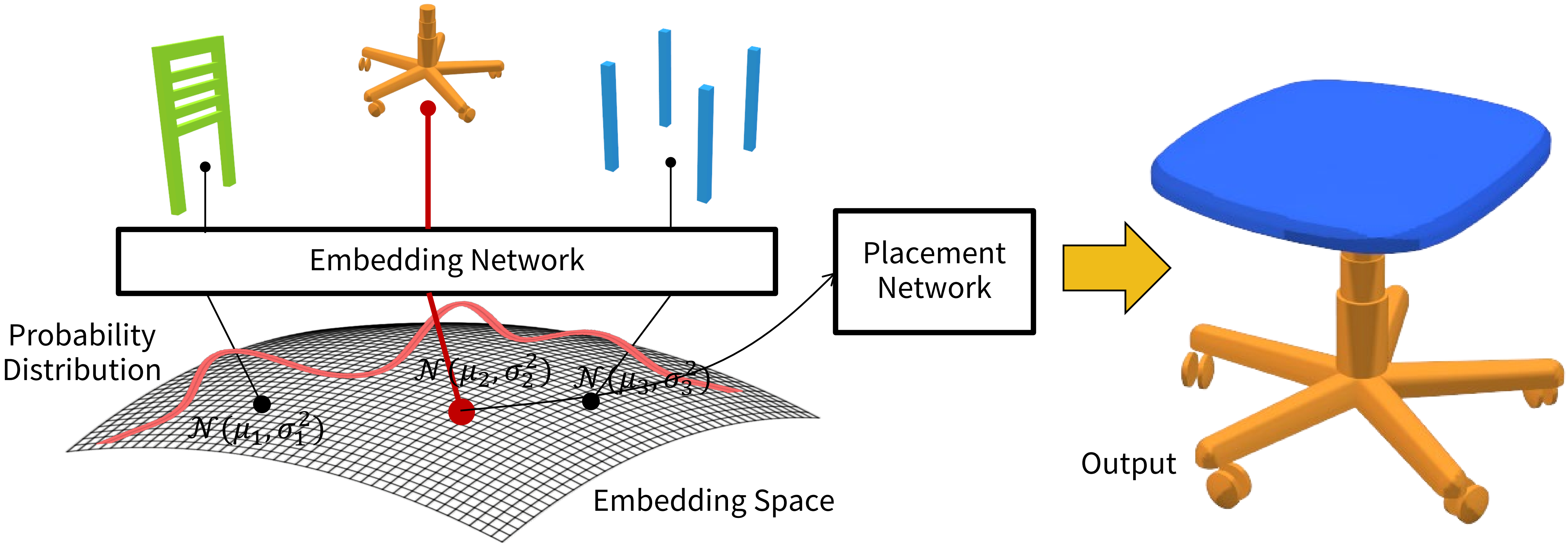
$$E(X, Y, Z) = \max\{m + E(X, Y) - E(Y, Z), 0\}$$

$$E(X, Y) = -\log P(Y|X)$$

$$P(Y|X) = \sum_k \varphi_k(X)\mathcal{N}(Y|\mu_k(X), \sigma_k(X)^2)$$

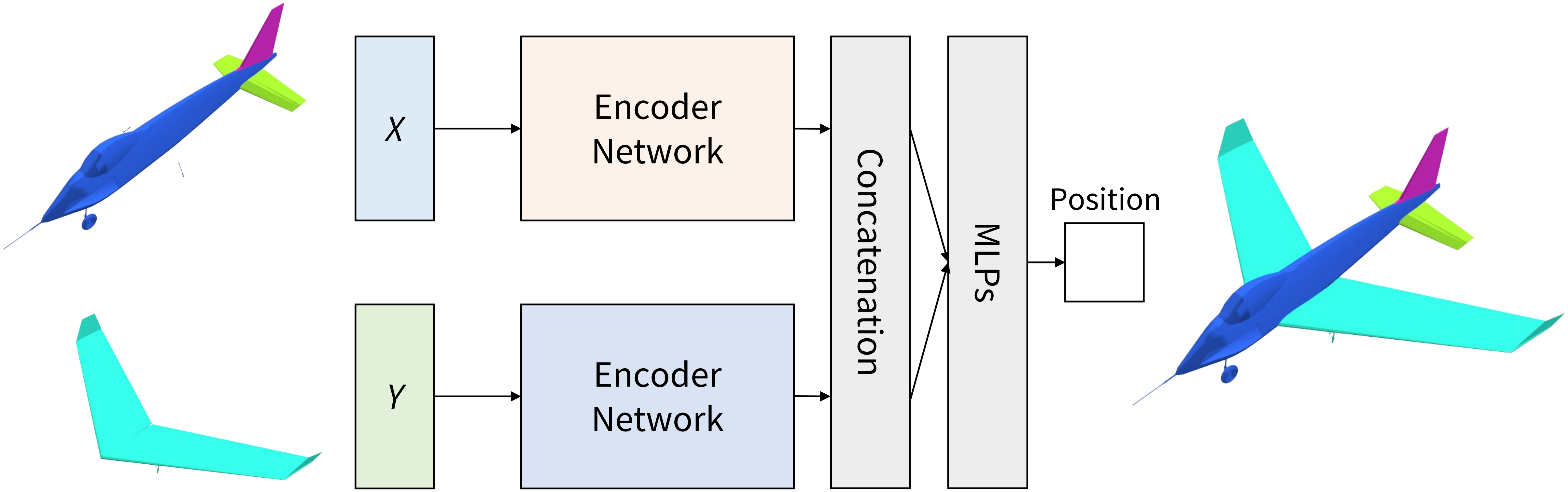
Placement Network

- Sample a complement from the predicted distribution.
- Predict the location of the selected component.



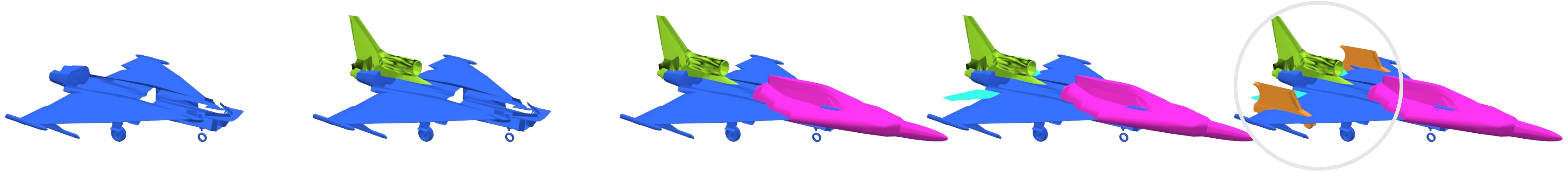
26

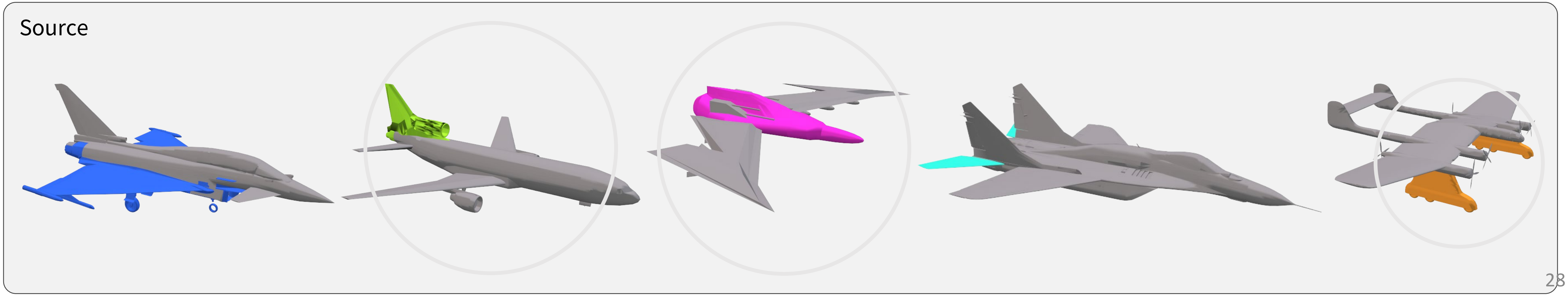
Placement Network



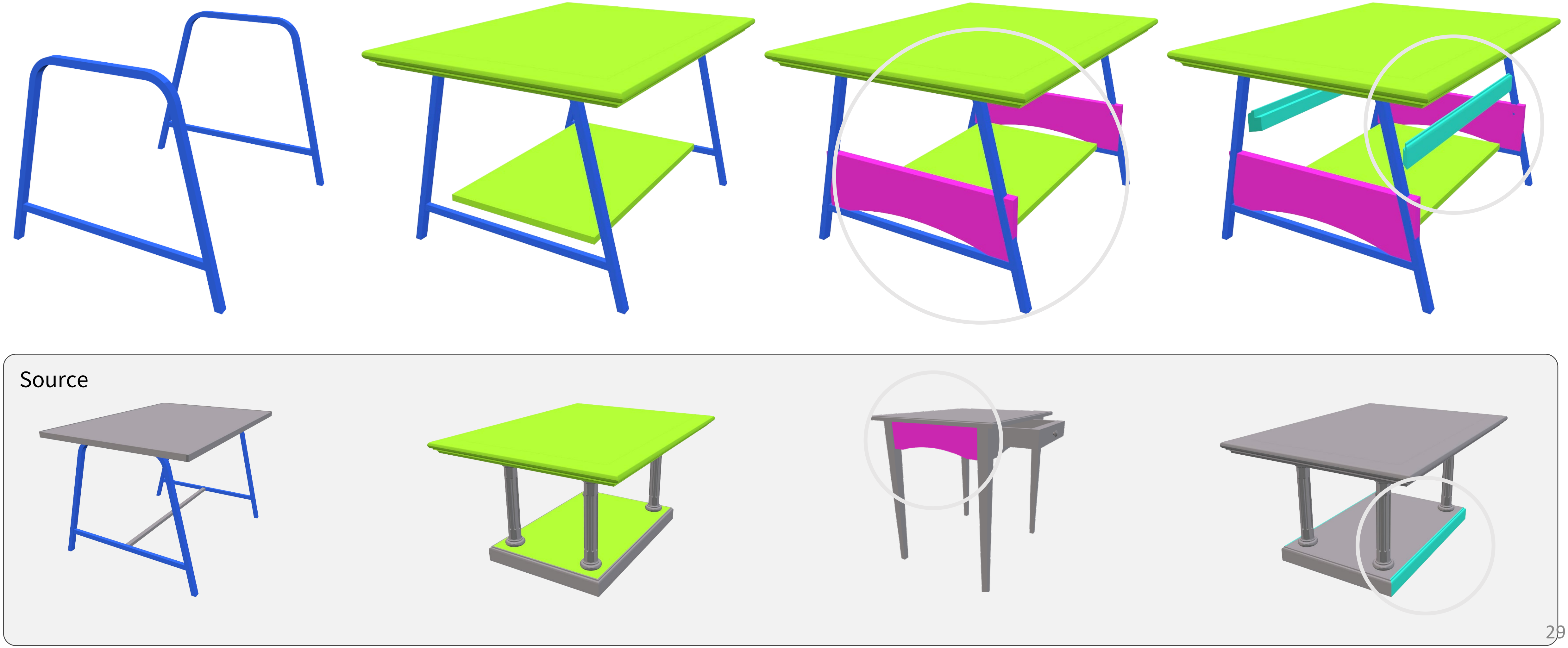
Automatic Shape Synthesis

Add the maximum probability part iteratively.

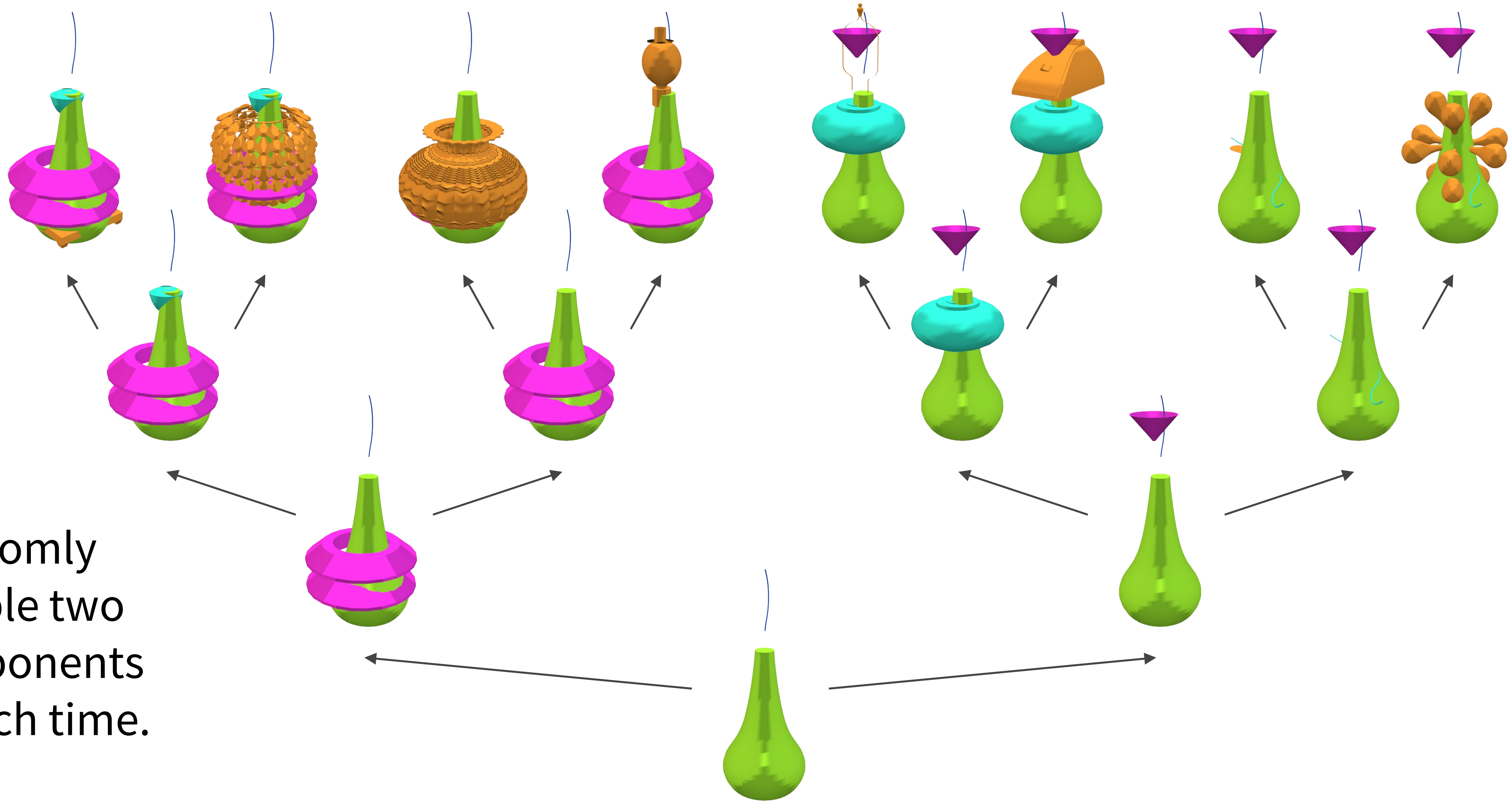




Automatic Shape Synthesis



Automatic Shape Synthesis

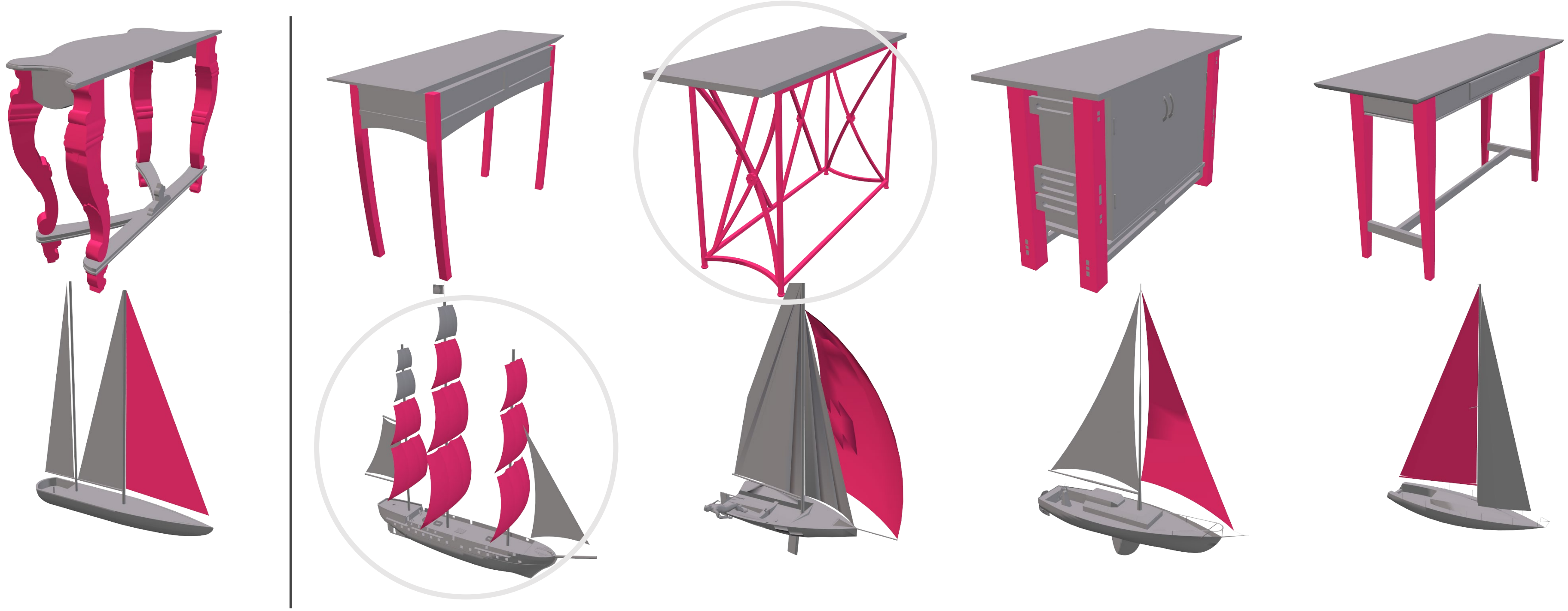


Randomly sample two components at each time.

Observation

The retrieval network discovers *interchangeable* parts.

Query Nearest neighbors in the embedding space



Can discover semantic relationships among parts!

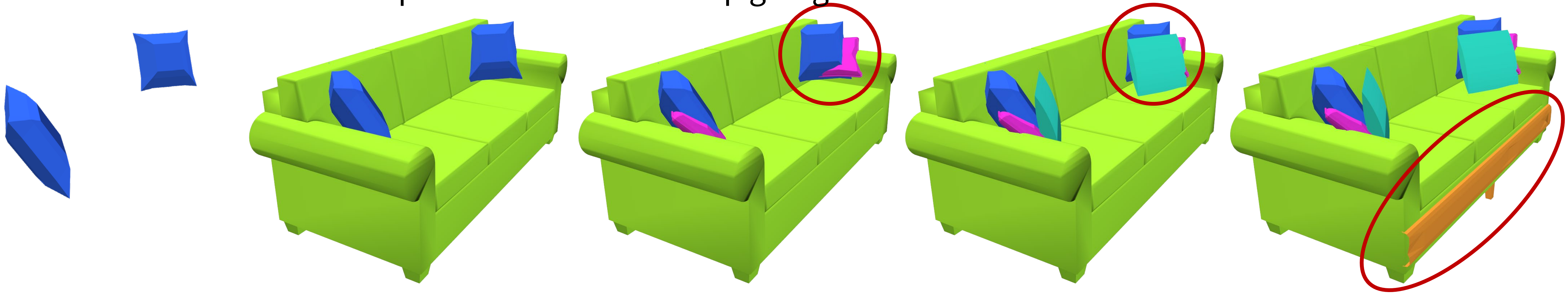
Limitations

Limitations

- Accumulating noise in iterations.
- Missing notion of termination.

Already complete!

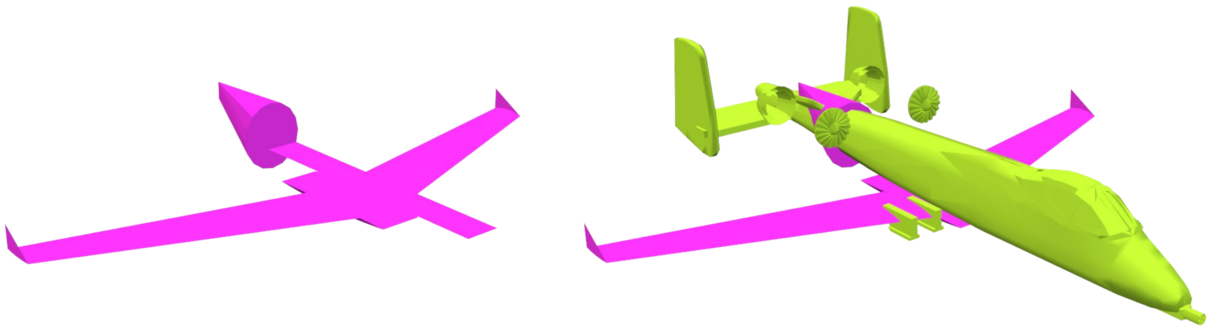
What happens if we keep going…?



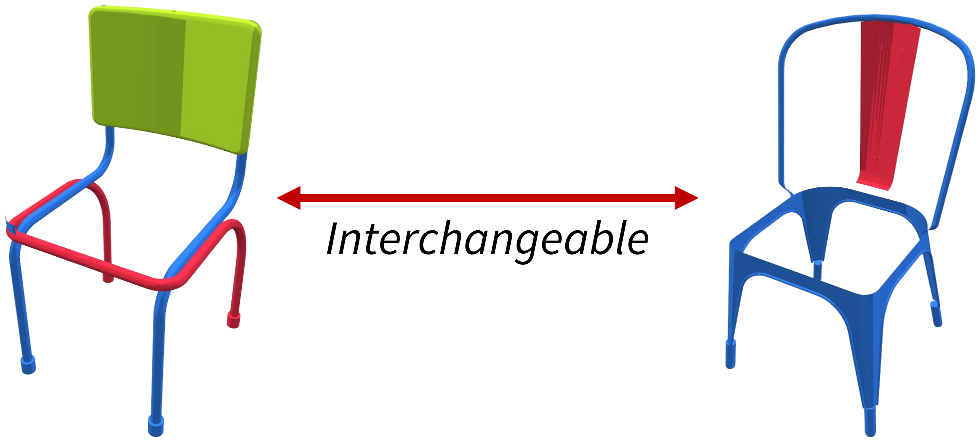
Learn Relations Among Partial Shapes

Learn relations among *partial shapes*.

- Can complete an object with a single retrieval.
- Can discover group-to-group relations.



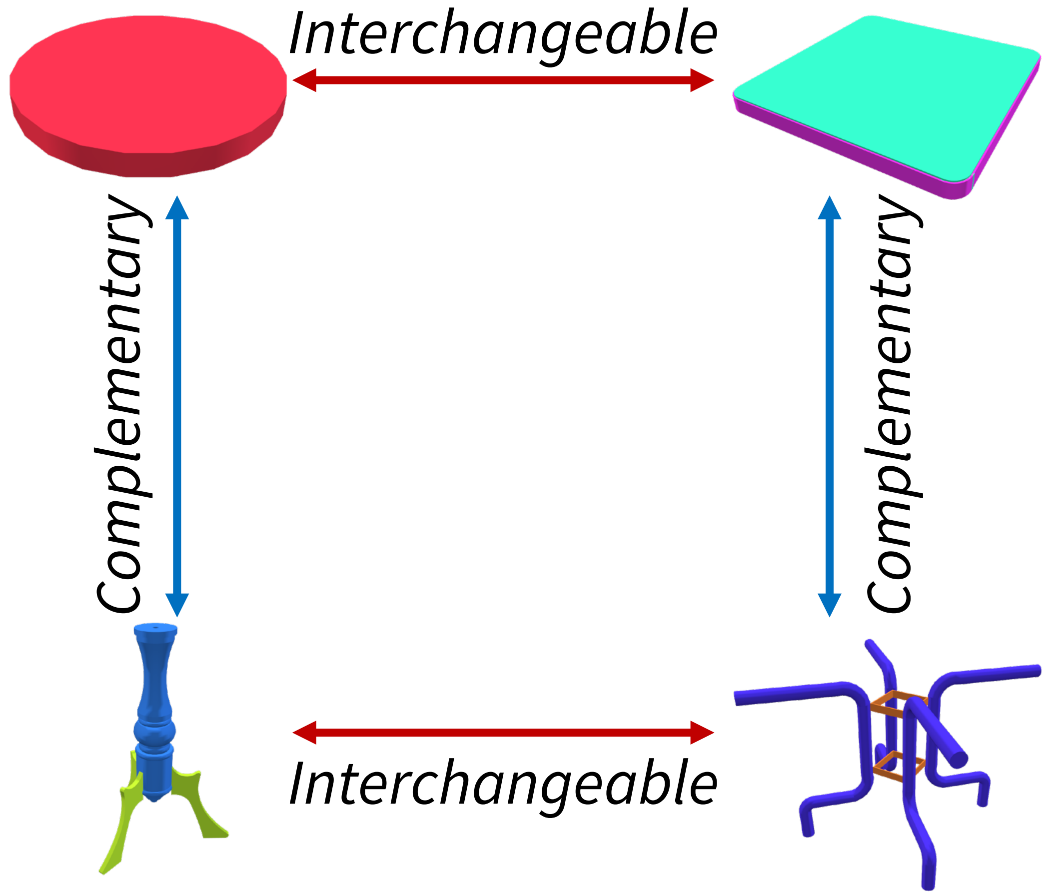
Query Retrieval



Relations Among Partial Shapes

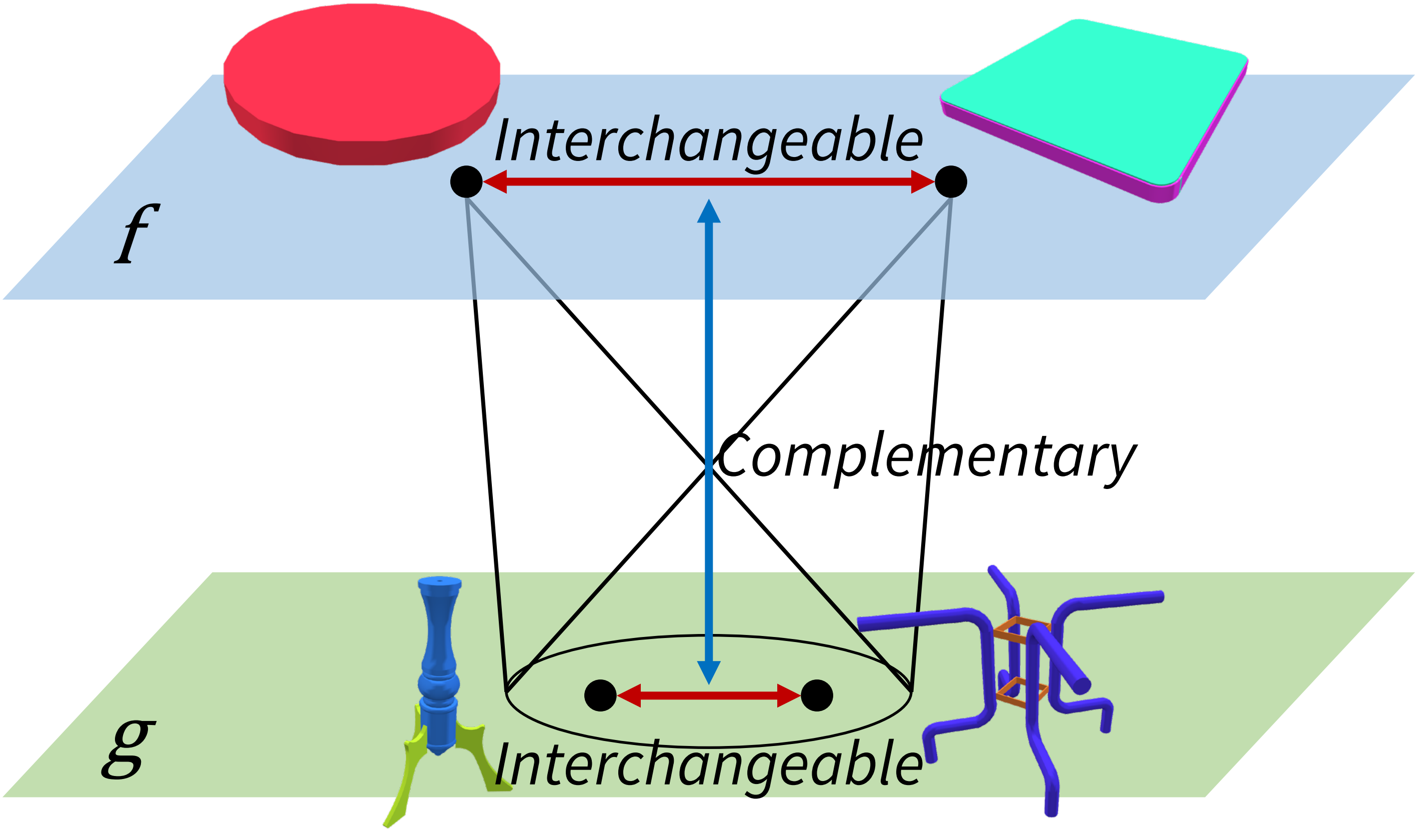
Learn relations among *partial shapes*.

- Complementarity
- Interchangeability



Dual Embedding Spaces

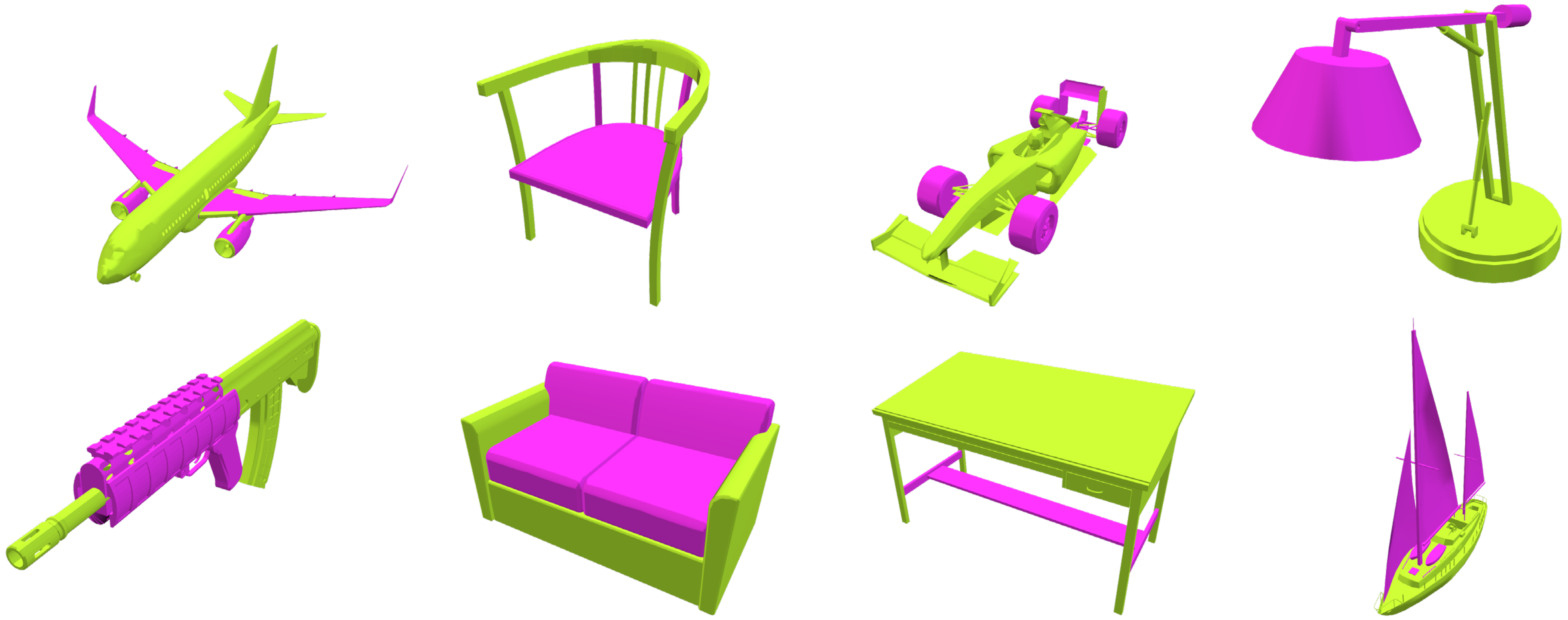
Jointly encode both *complementary* and *interchangeable* relations in a *dual* embedding space.



Learning

Learn *interchangeability* from *complementarity.*

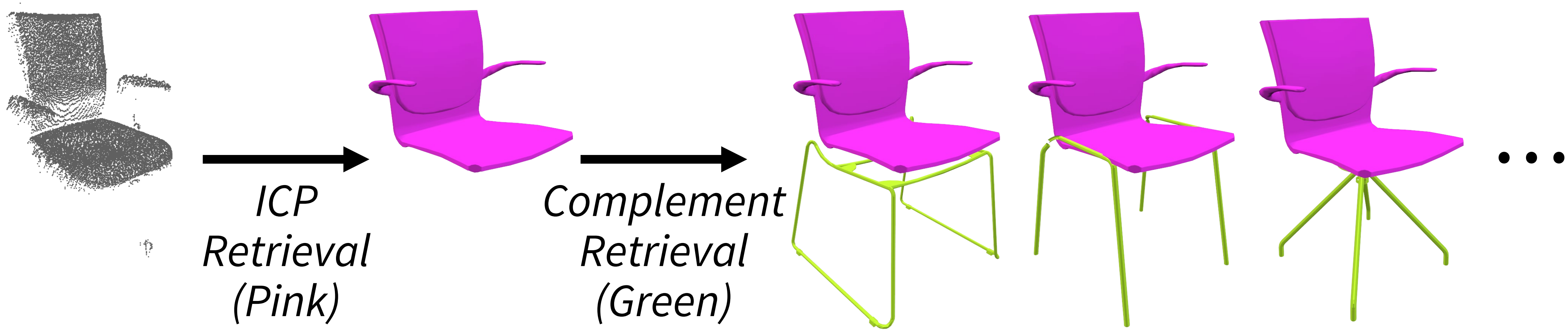
- *Complementary* pairs are created by splitting objects.
- No supervision for *interchangeability* is given.



Applications

- Shape analysis

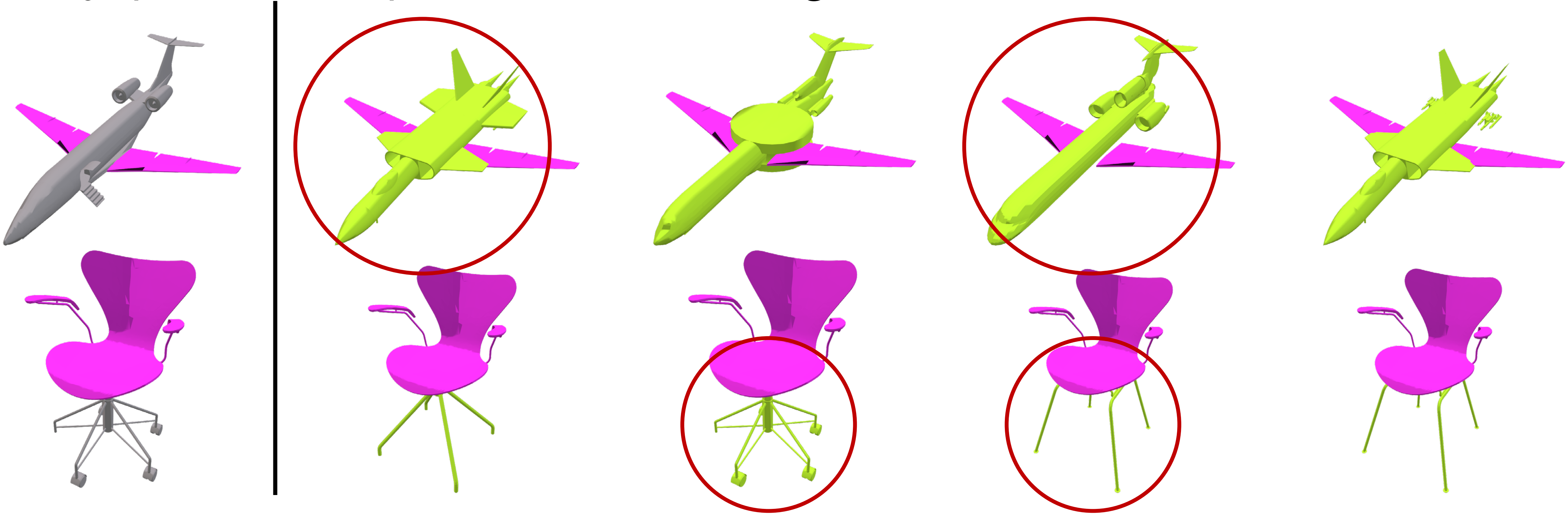
- Shape completion



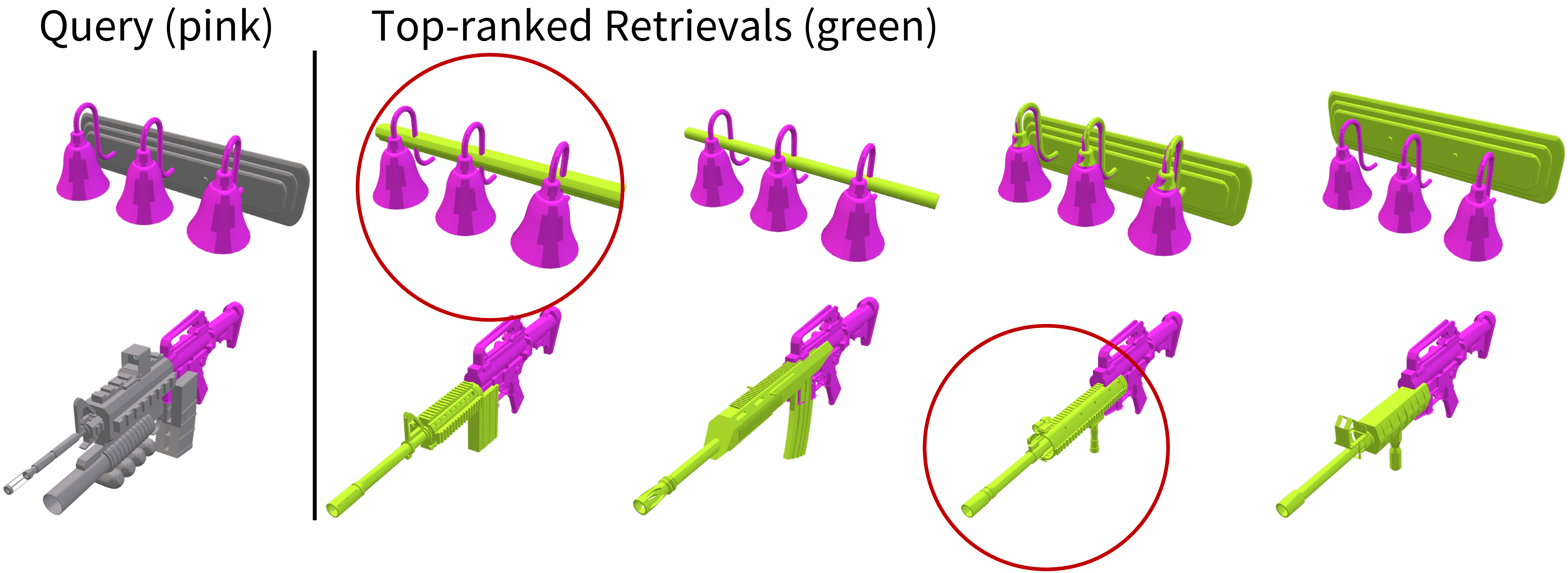
Complementary Shape Retrievals

Query (pink)

Top-ranked Retrievals (green)



Complementary Shape Retrievals

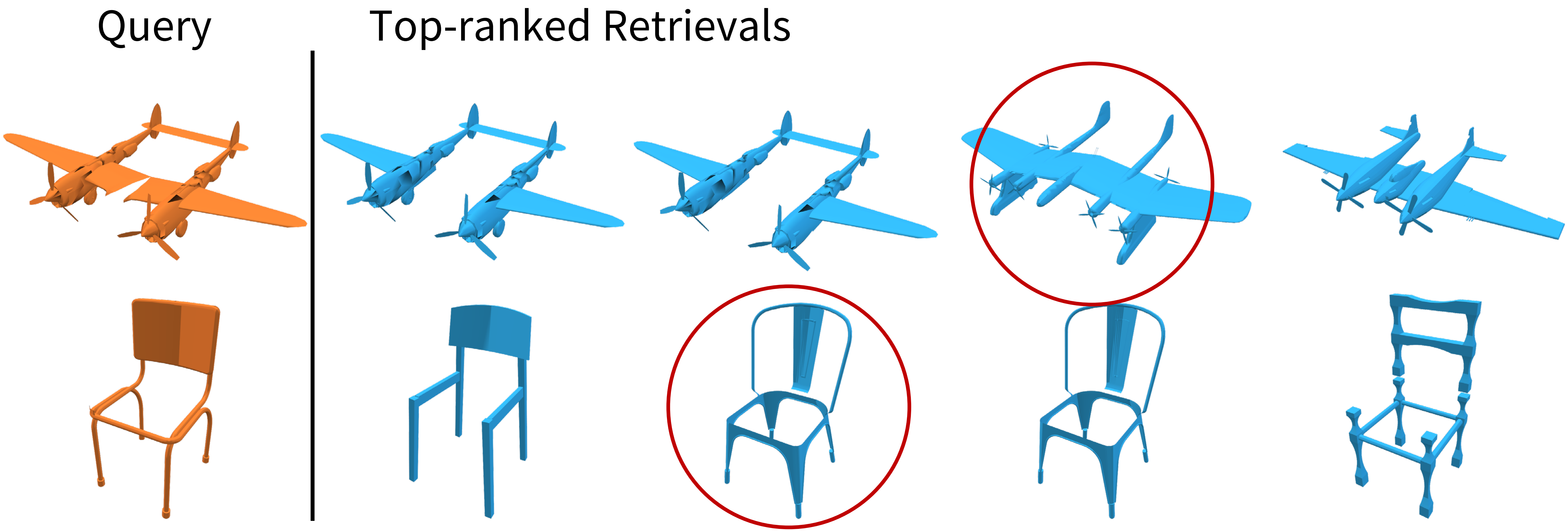


Complementary Shape Retrievals

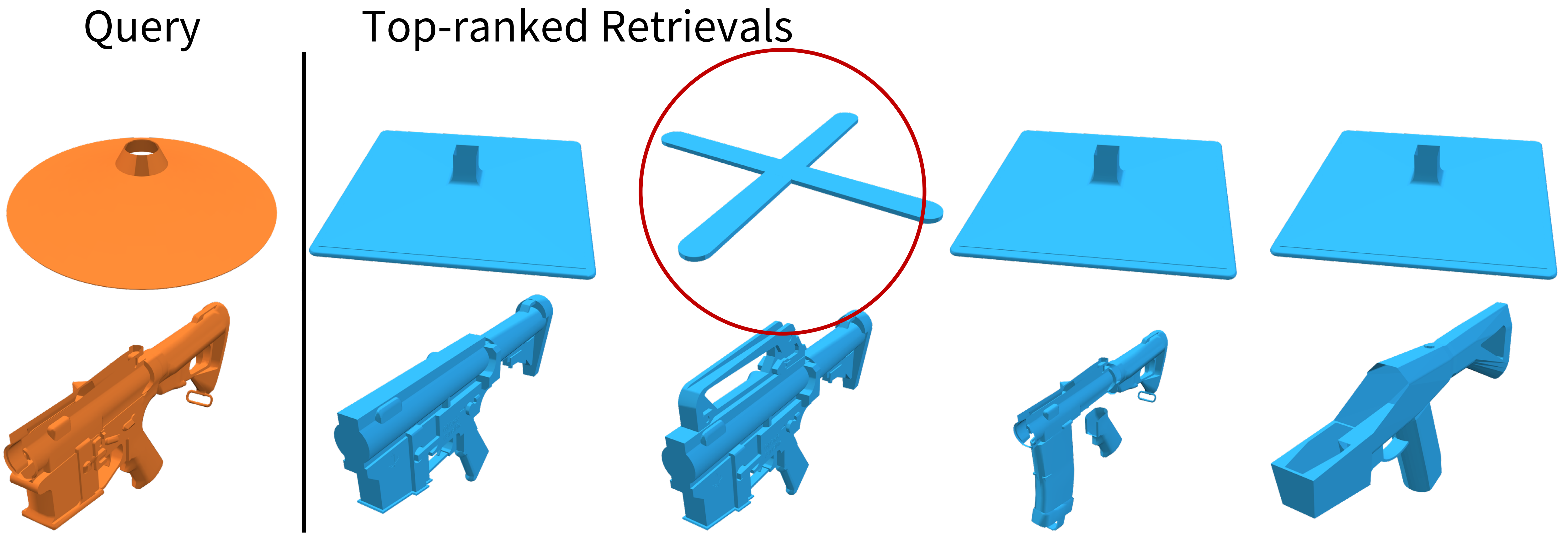
Query (pink)

Top-ranked Retrievals (green)

Interchangeable Shape Retrievals



Interchangeable Shape Retrievals



Conditional Shape Generation Based on Image Data

3D Perception from a Single Image

Visual 3D Cues are Complicated

contrast

color

texture

motion

symmetry

part knowledge

category-specific 3D knowledge

......

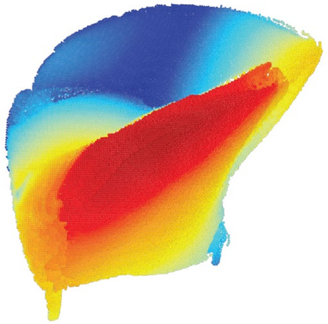
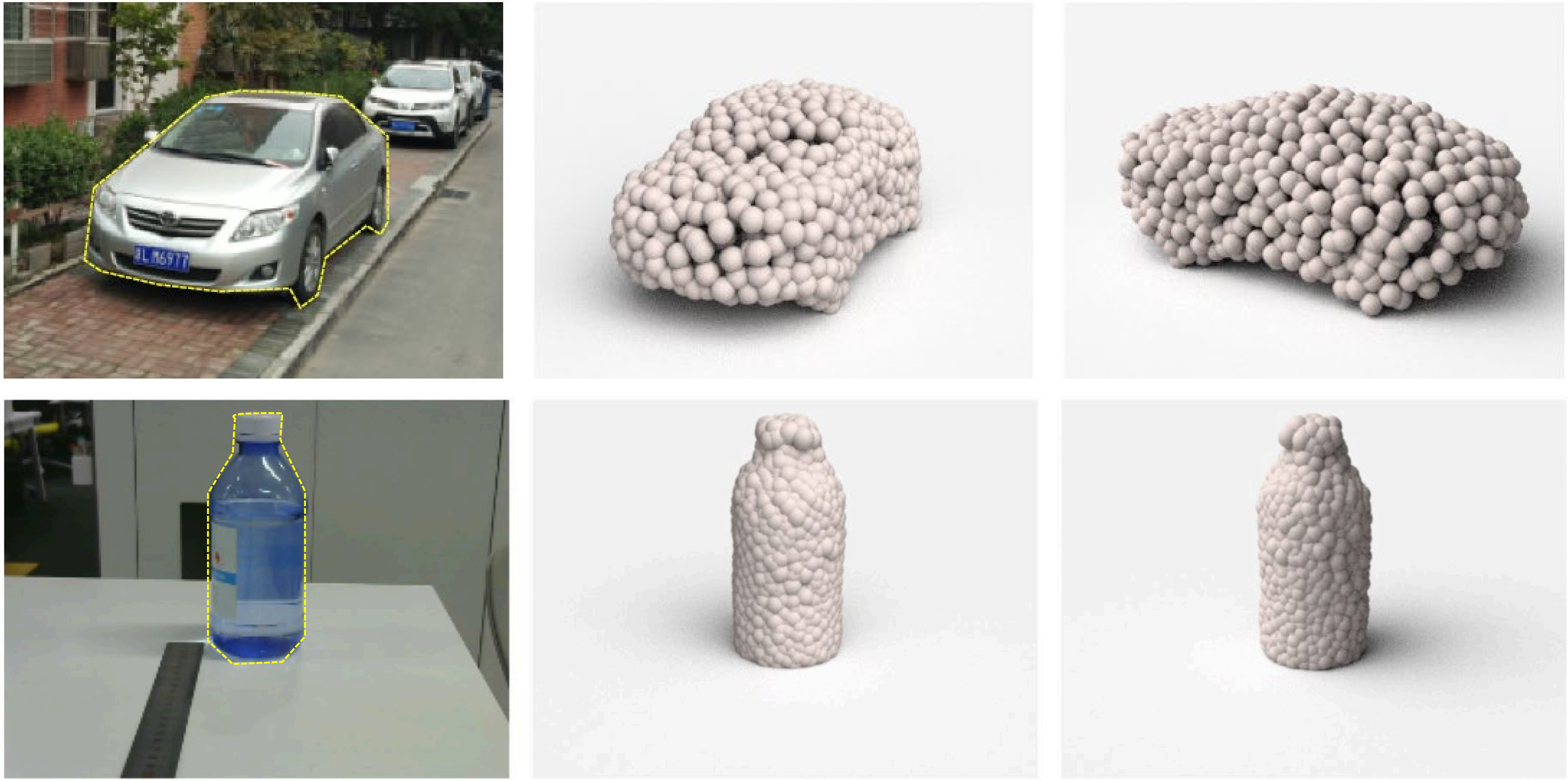


Image 2 PointCloud

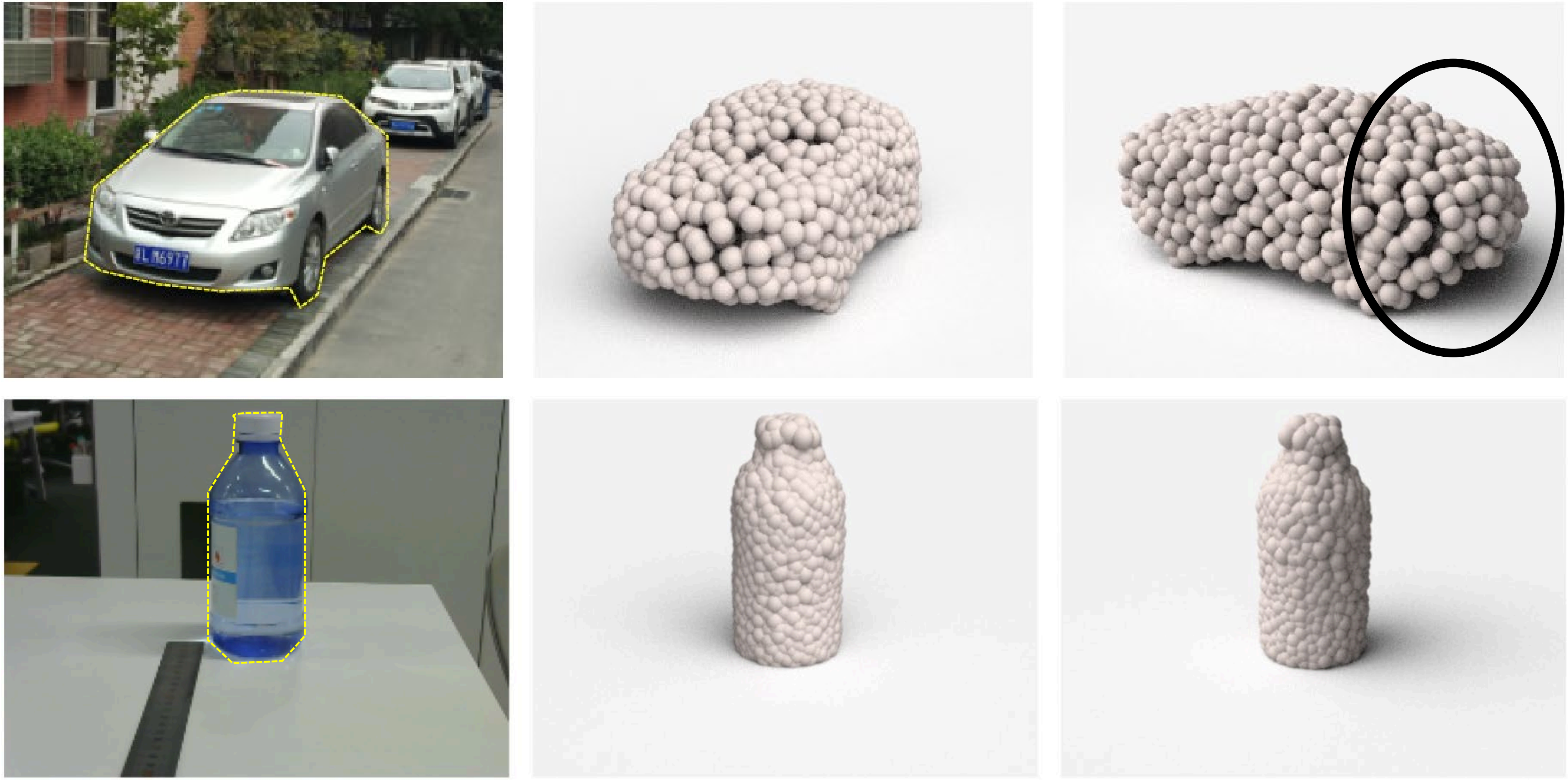
Fan, H., Su, H. and Guibas, L.J., 2017. A point set generation network for 3D object reconstruction from a single image. CVPR 2017

Point Cloud Synthesis from a Single Image



Input | Reconstructed 3D point cloud

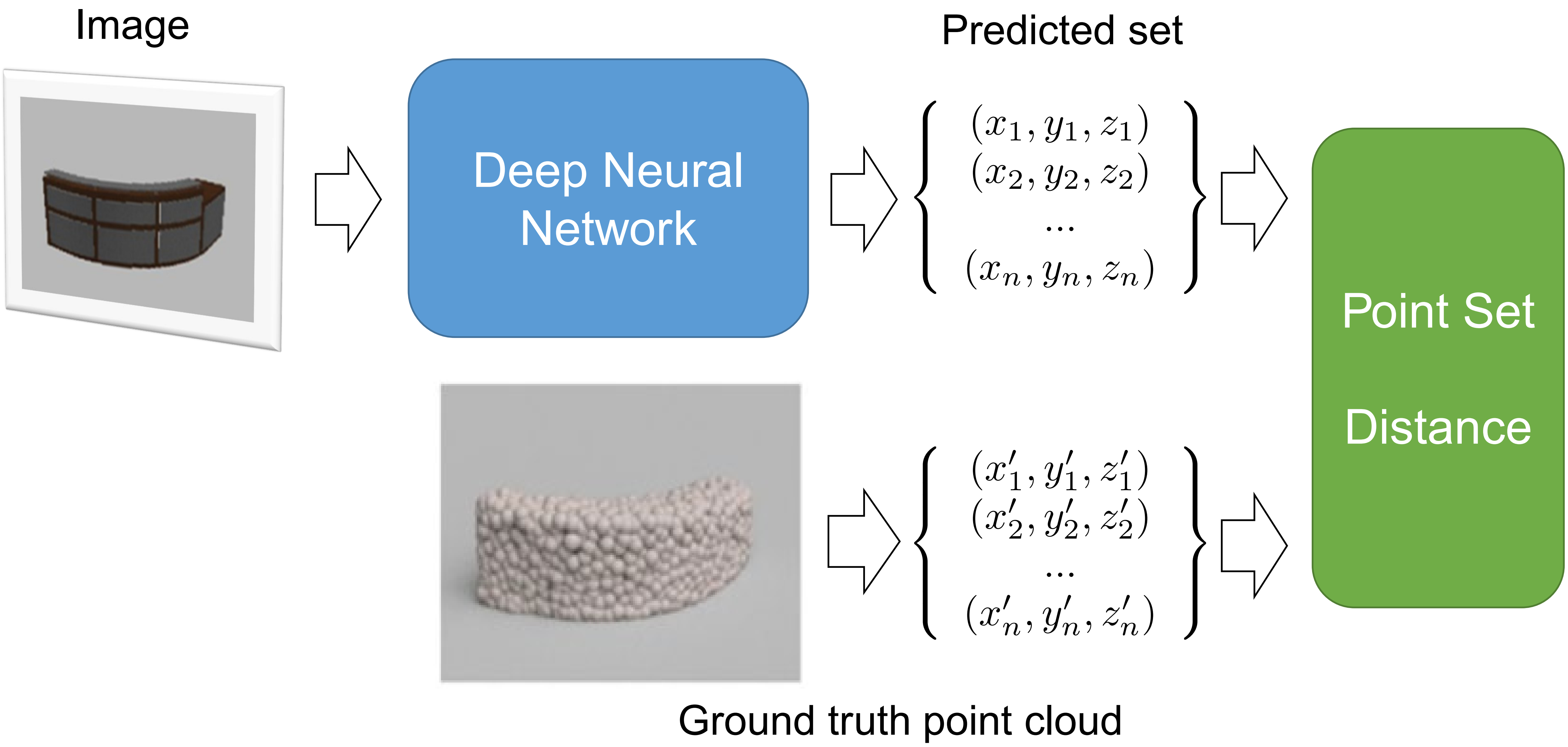
Point Cloud Synthesis from a Single Image



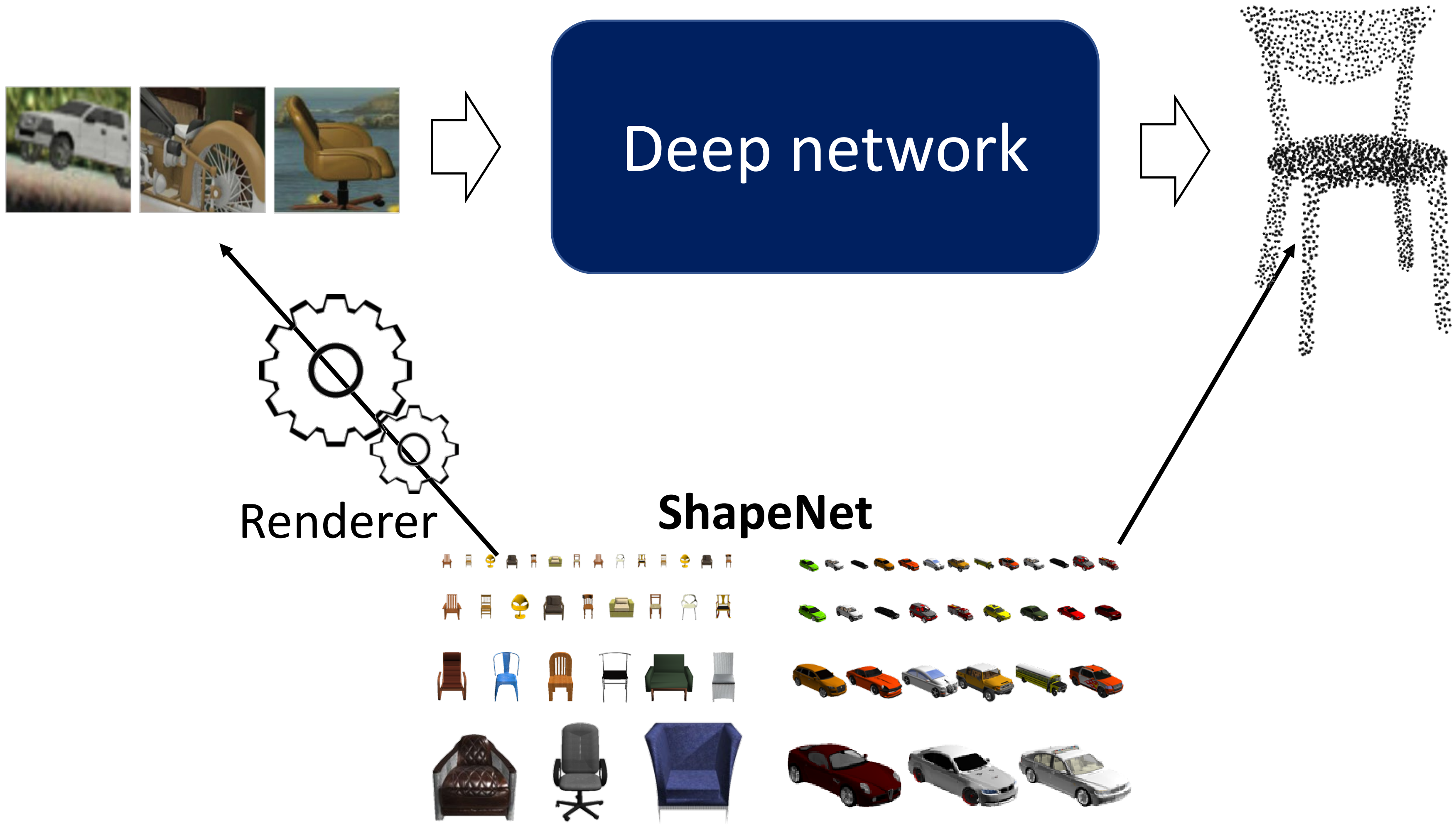
Input

Reconstructed 3D point cloud

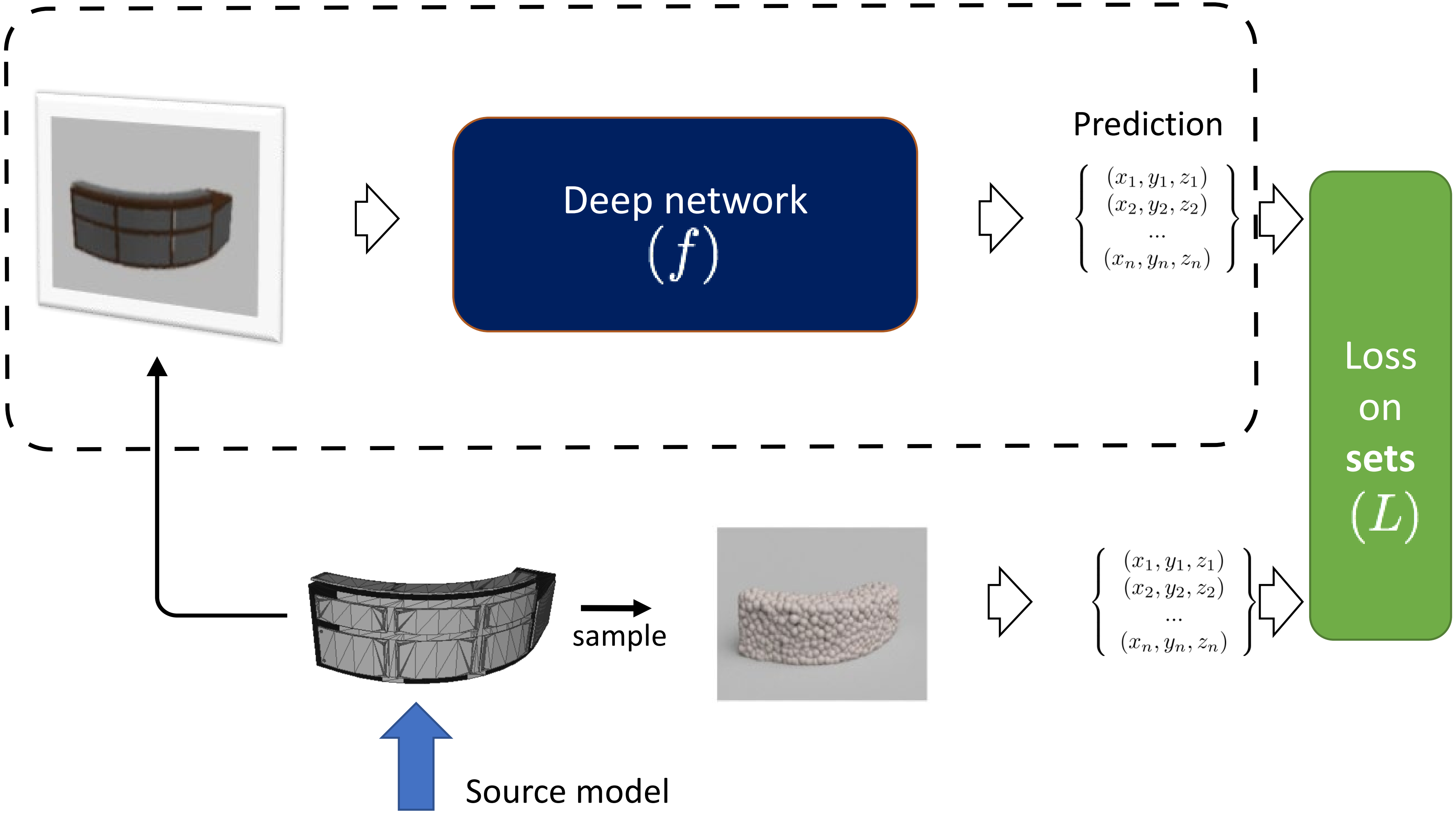
End-to-End Learning



Synthesize for Learning

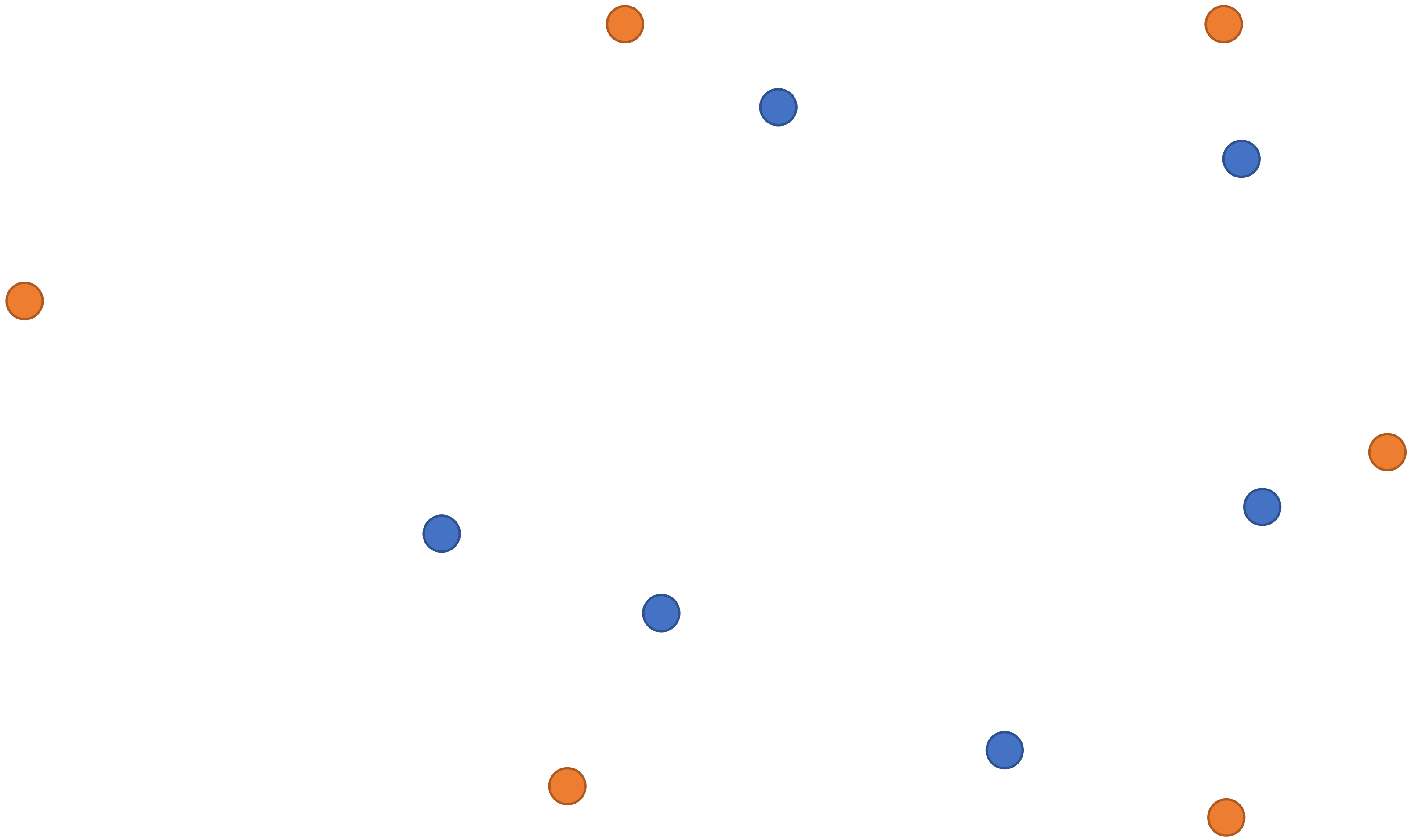


End-to-End Learning



Point Cloud Distance Metrics

Given two sets of points, measure their discrepancy



Point Cloud Distance Metrics

Worst case: Hausdorff distance (HD)

$$d_{\mathrm{HD}}(S_1, S_2) = \max\{\max_{x_i \in S_1} \min_{y_j \in S_2} \|x_i - y_j\|, \max_{y_j \in S_2} \min_{x_i \in S_1} \|x_i - y_j\|\}$$

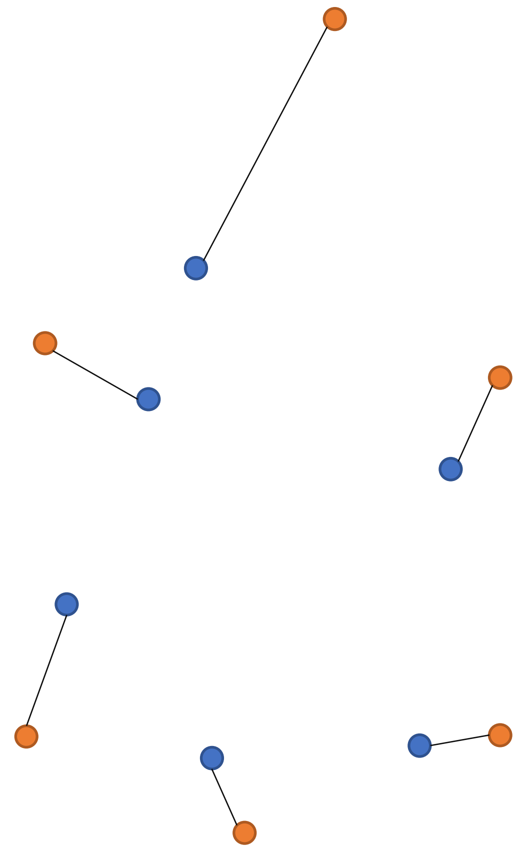
Average case: Chamfer distance (CD)

$$d_{CD}(S_1, S_2) = \frac{1}{n} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{m} \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

Optimal case: Earth Mover's distance (EMD)

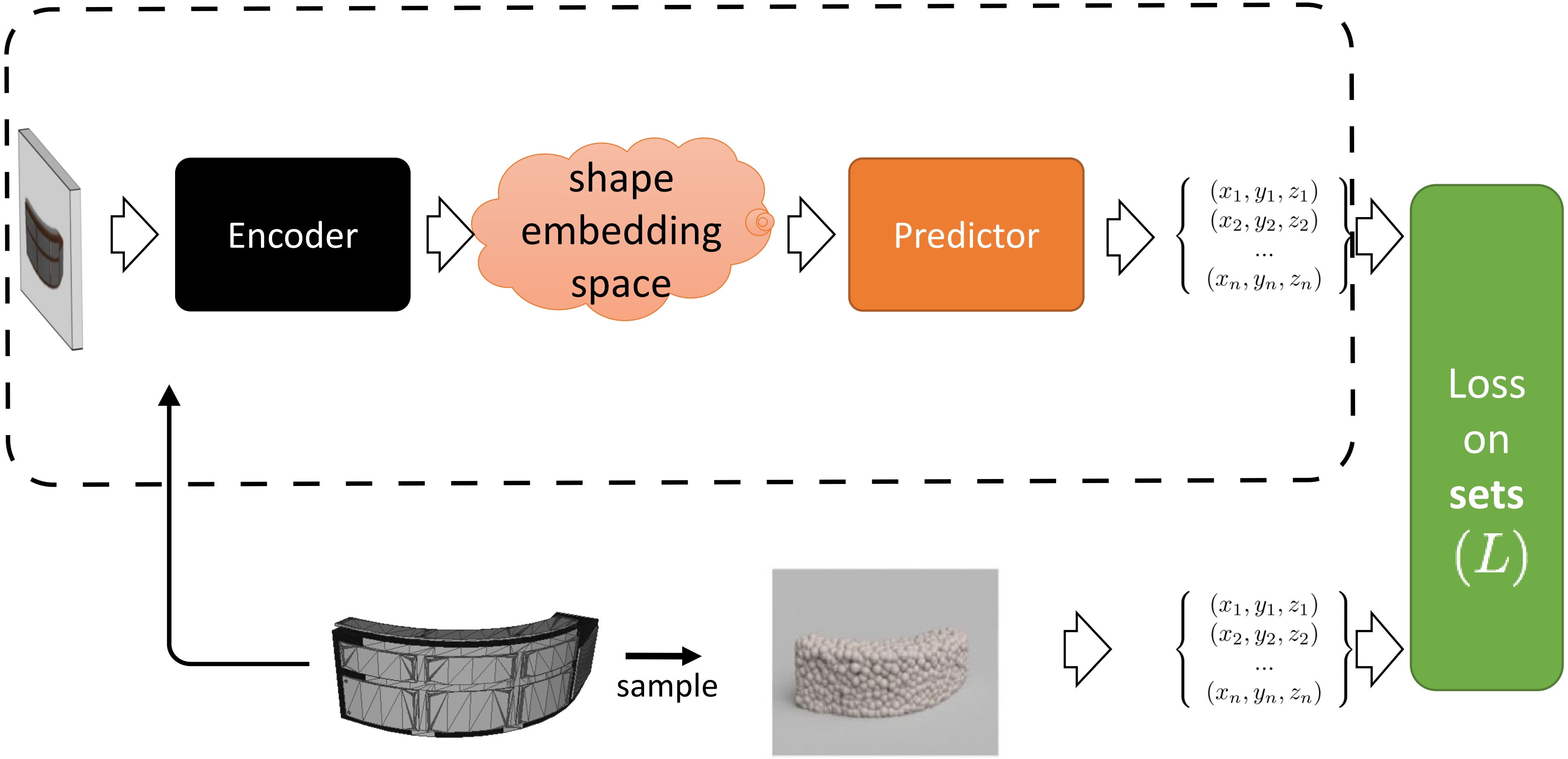
$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where $\phi : S_1 \to S_2$ is a bijection.

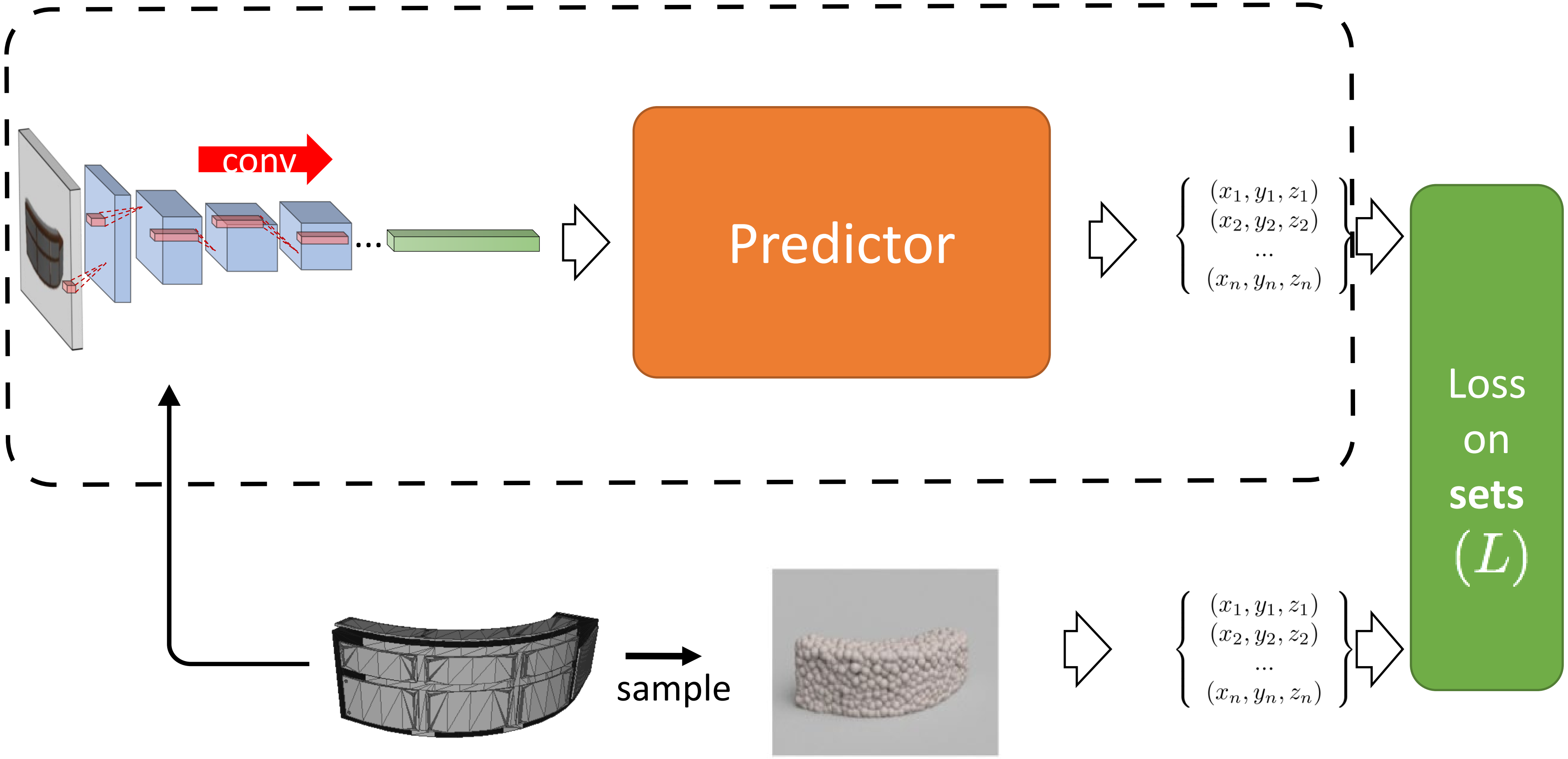


Solves the optimal transportation (bipartite matching) problem!

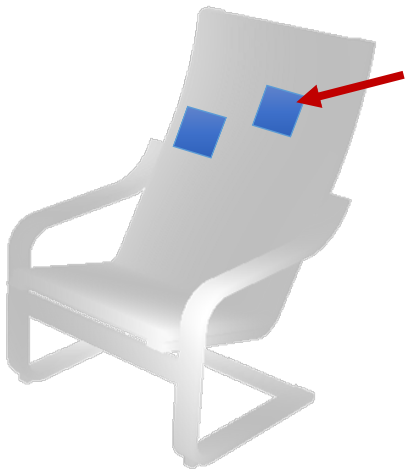
End-to-End Learning Architecture

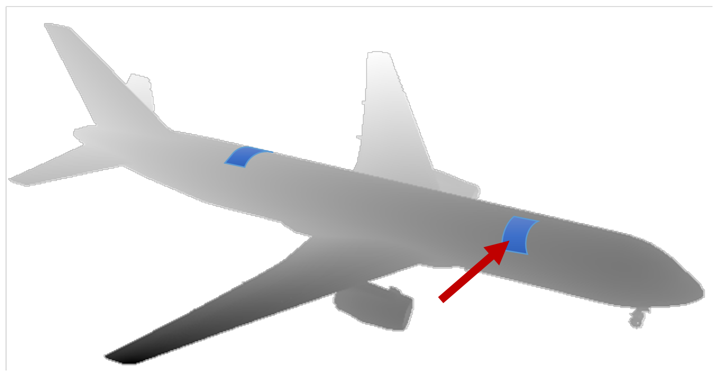


End-to-End Learning Architecture



Natural Statistics of Object Geometry

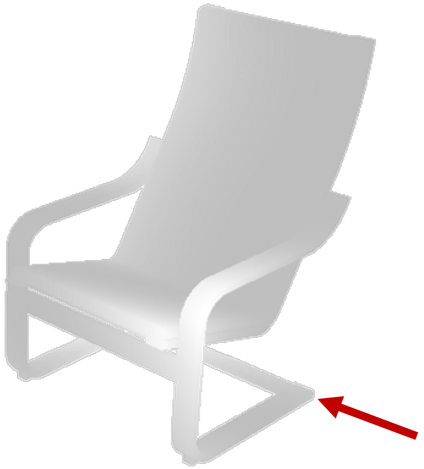


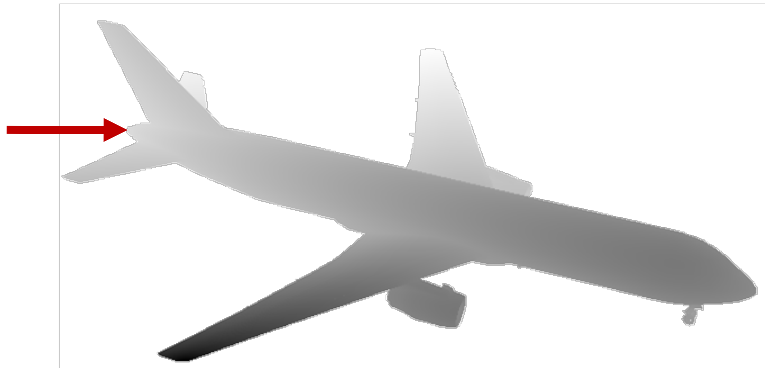


- Many smooth local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates

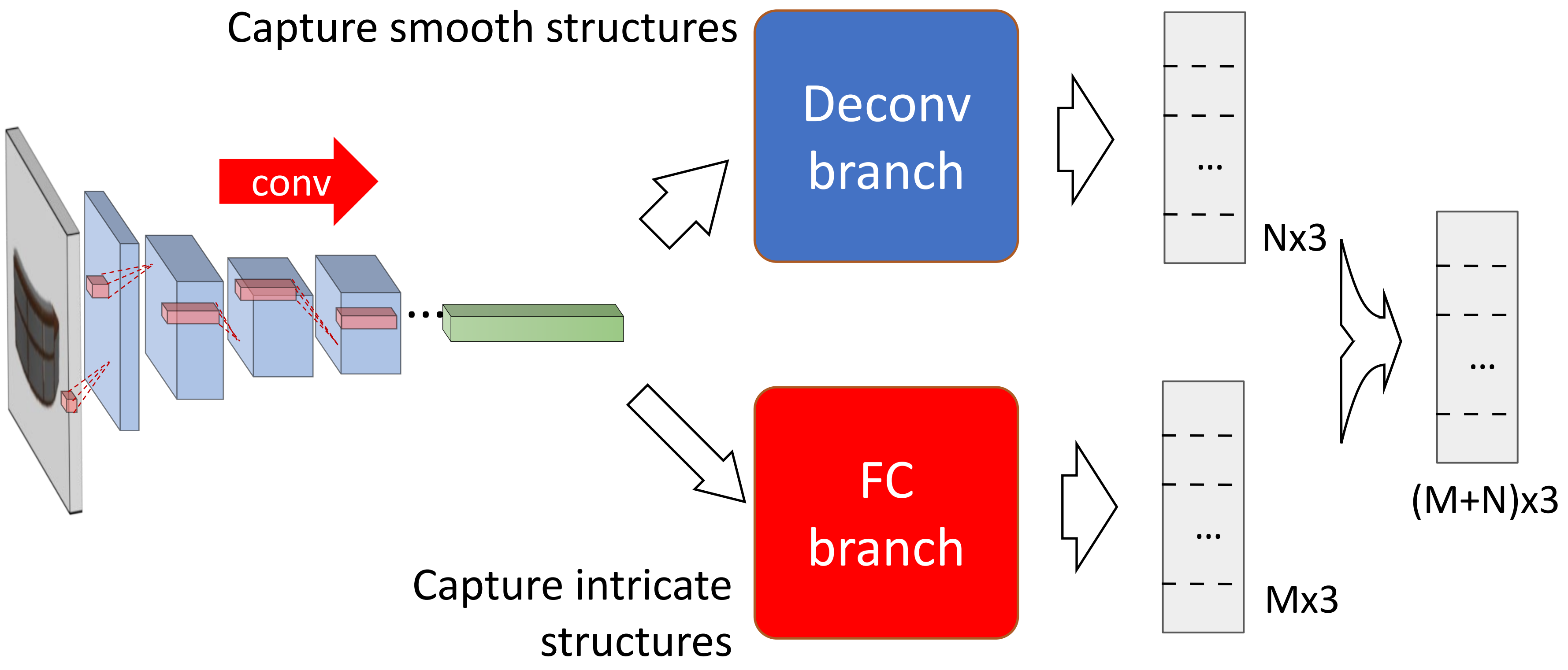
Natural Statistics of Object Geometry

- Many local structures are common/shared
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates
- But also some intricate local structures
  - **some points have high variability neighborhoods**



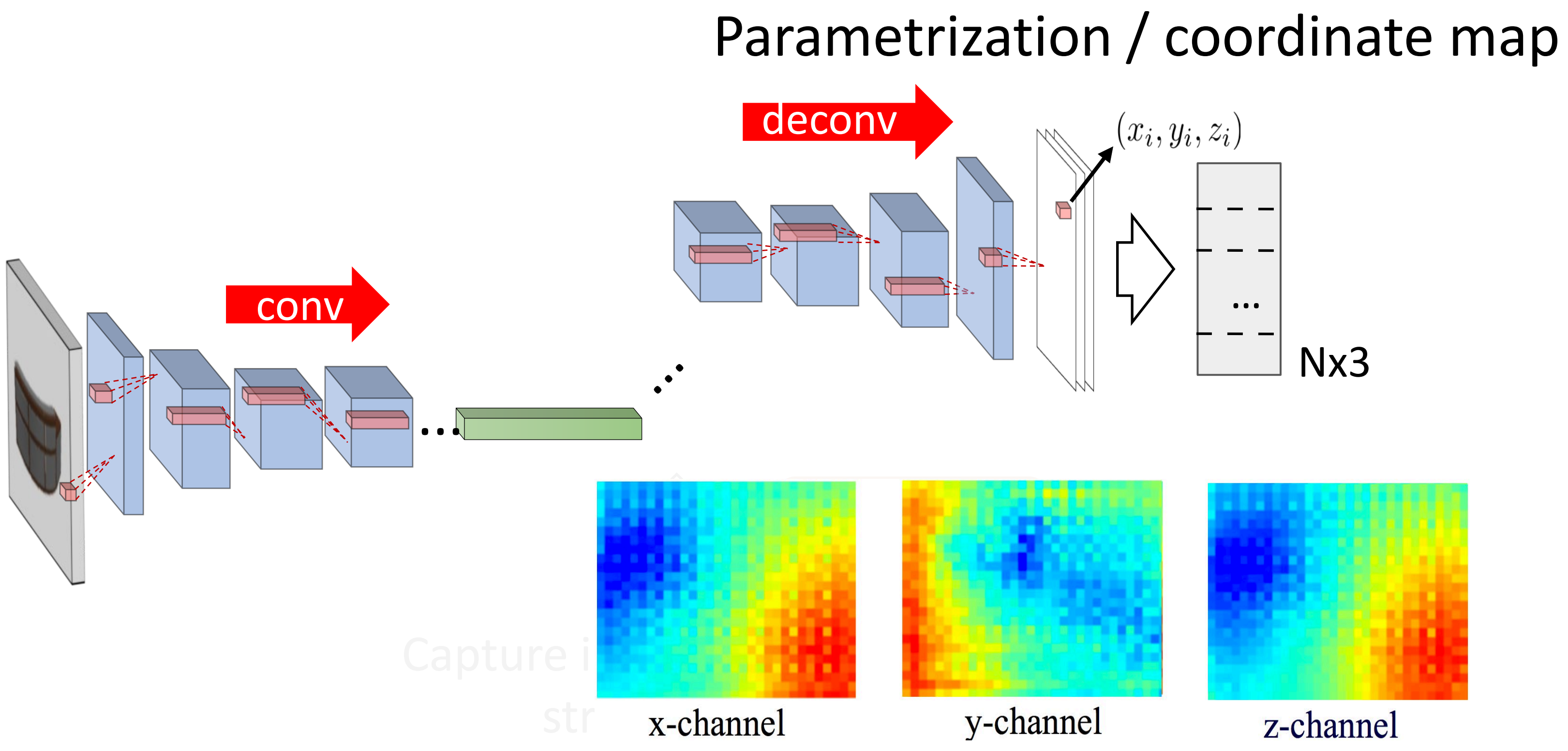


Two-Branch Architecture



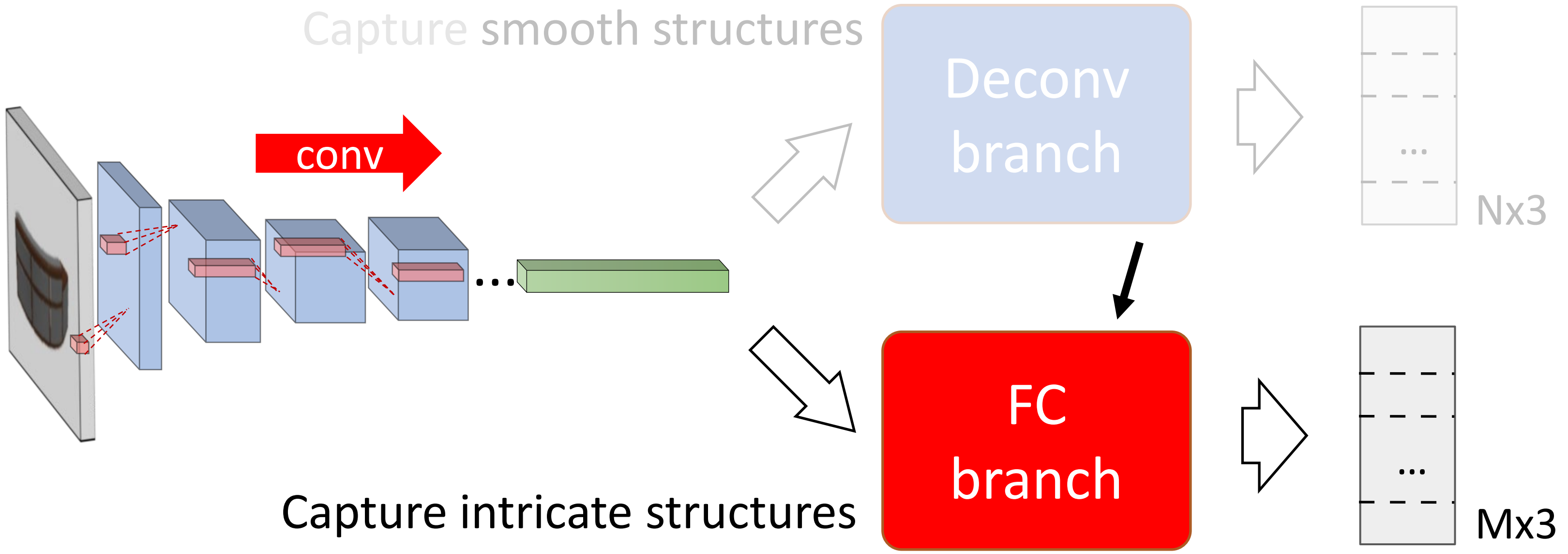
Set union by array concatenation

Deconvolution Branch



- Deconvolution induces a smooth coordinate map
- Geometrically, learns a smooth parameterization

Fully Connected Branch



The Two Branches

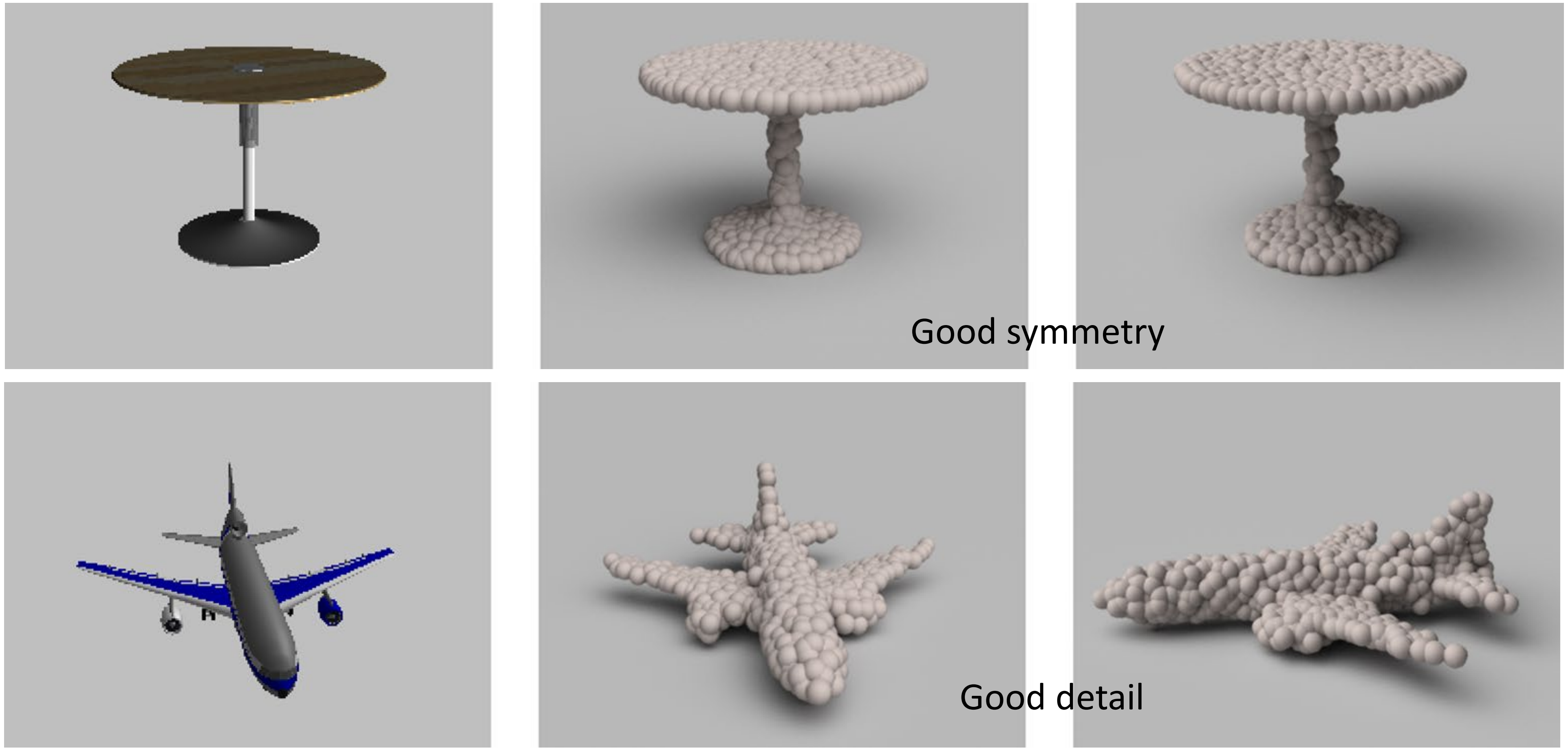
blue: deconv branch – **large, consistent, smooth** structures

red: fully-connected branch – **more intricate** structures

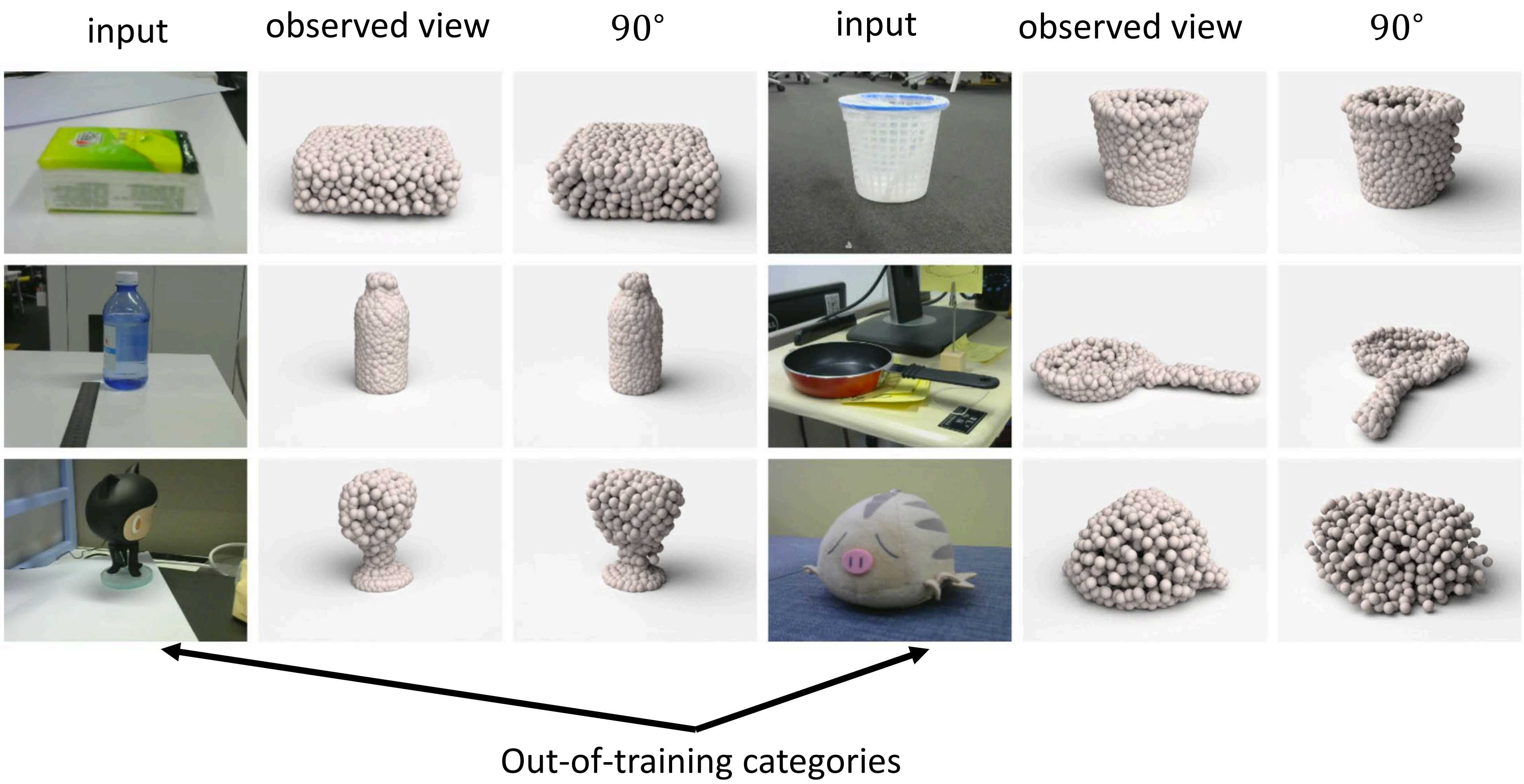
Example Results

Same view

New view

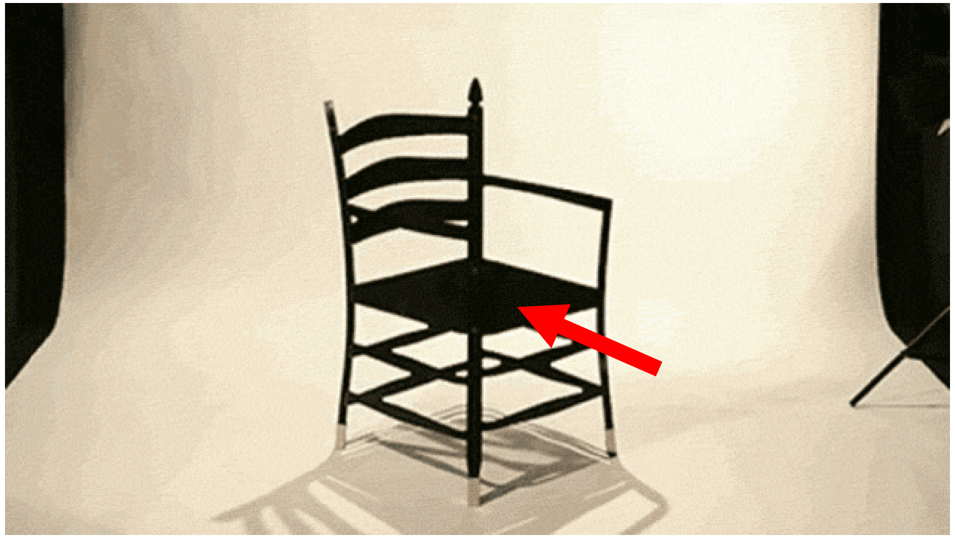


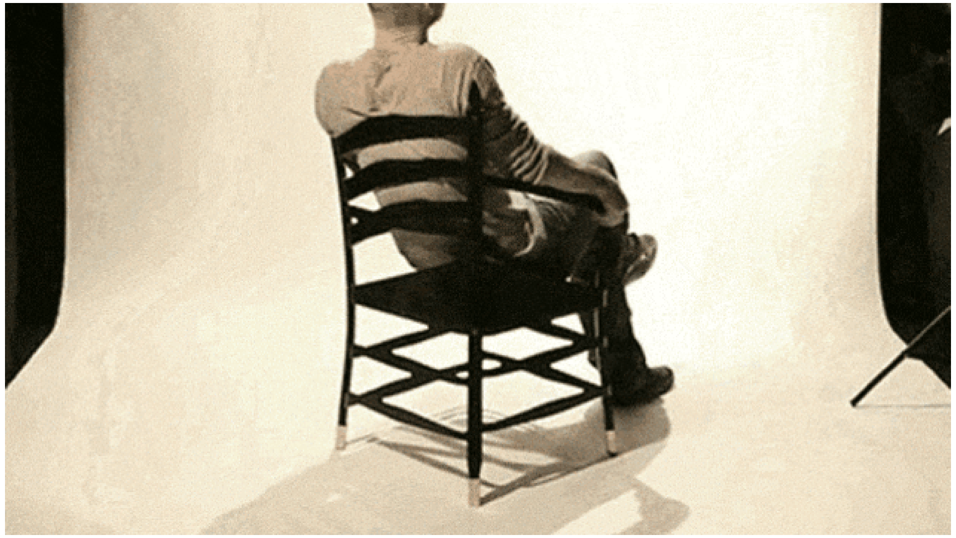
From Real Images



Ambiguity in Object Views

- A fundamental issue: inherent ambiguity in prediction



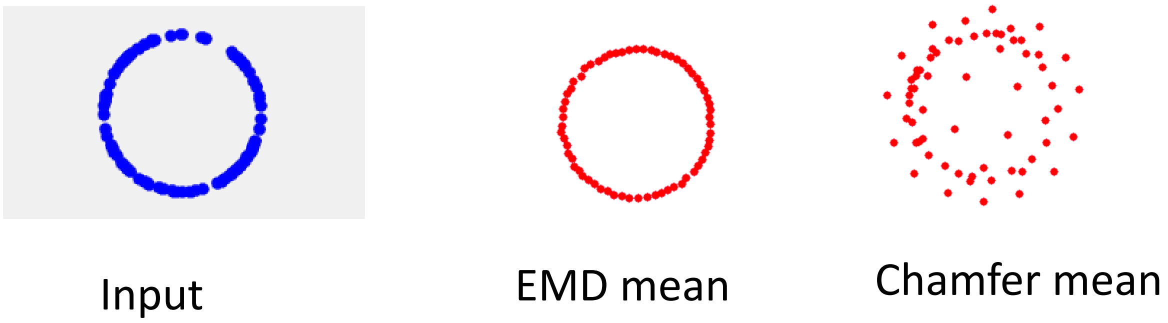


- By loss minimization, the network tends to predict a "**mean shape**" that **averages out** uncertainty

Distance Metrics Affect Mean Shapes

$$\bar{x} = \underset{x}{\operatorname{argmin}} \, \mathbb{E}_{s \sim \mathbb{S}}[d(x, s)]$$

continuous
hidden variable
(radius)

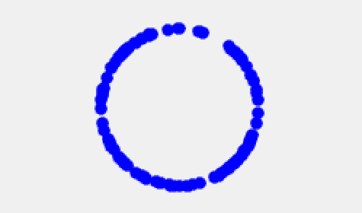


The mean shape carries characteristics of the distance metric

Distance Metrics Affect Mean Shapes

The mean shape carries characteristics of the distance metric

continuous
hidden variable
(radius)

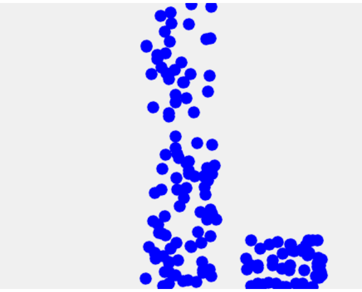


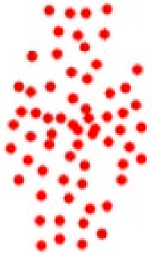
Input

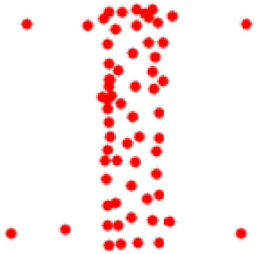
EMD mean

Chamfer mean

discrete
hidden variable
(add-on location)





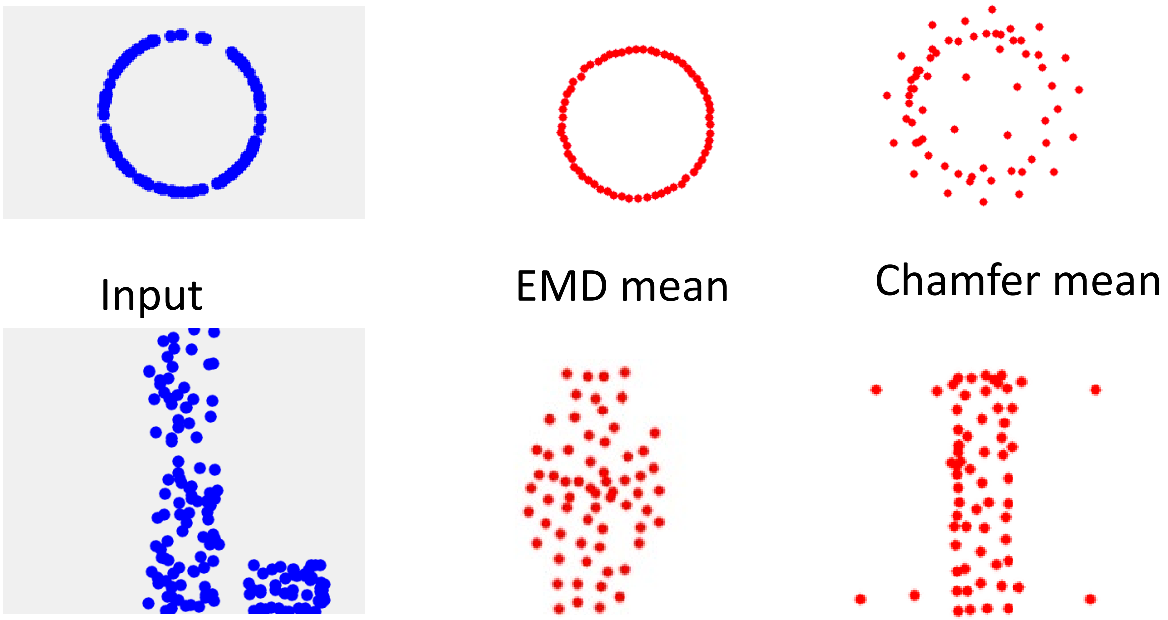


Distance Metrics Affect Mean Shapes

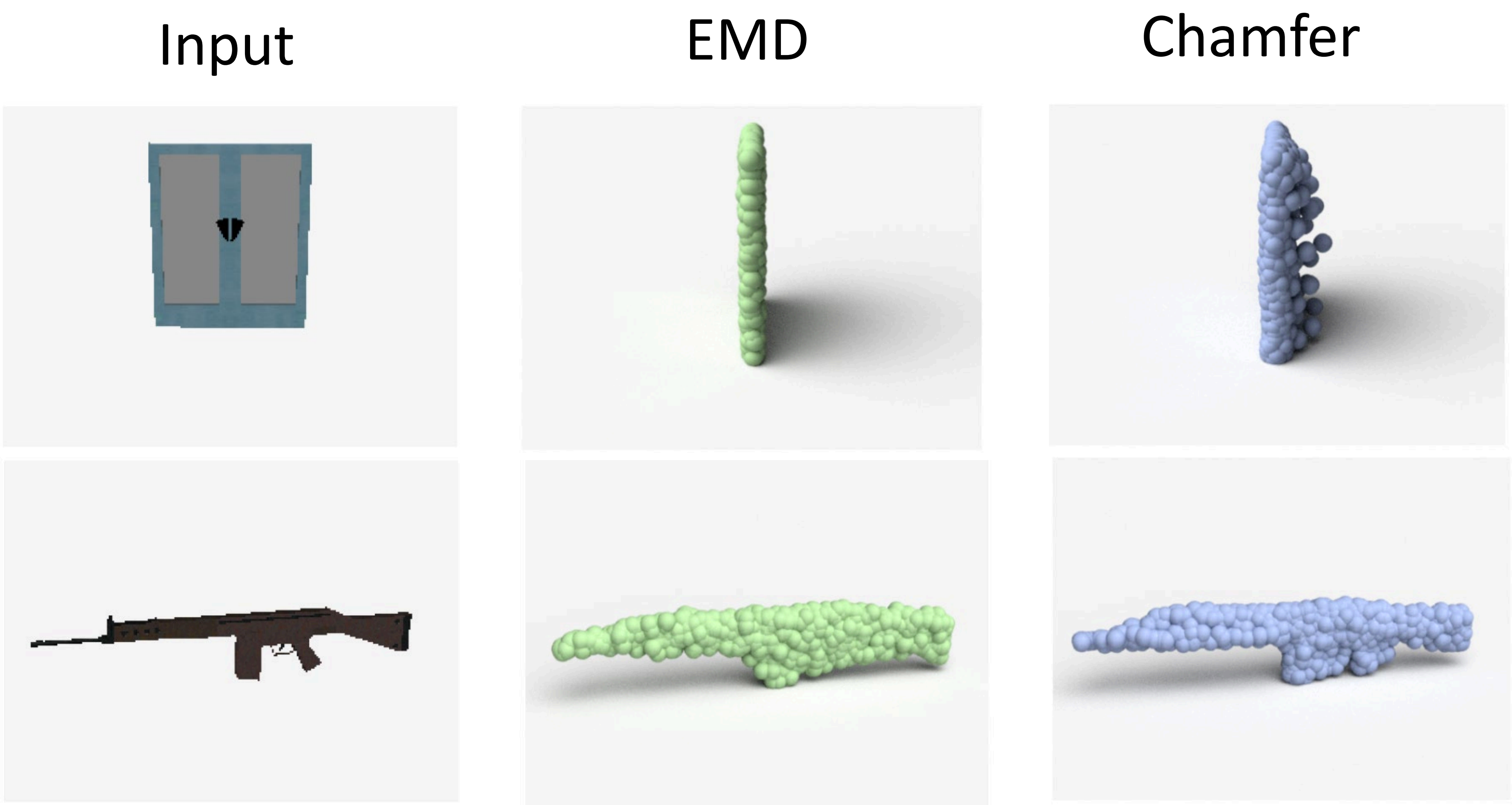
The mean shape carries characteristics of the distance metric

continuous
hidden variable
(radius)

discrete
hidden variable
(add-on location)



EMD vs CD Predictions



Additional Views Reduce Uncertainty

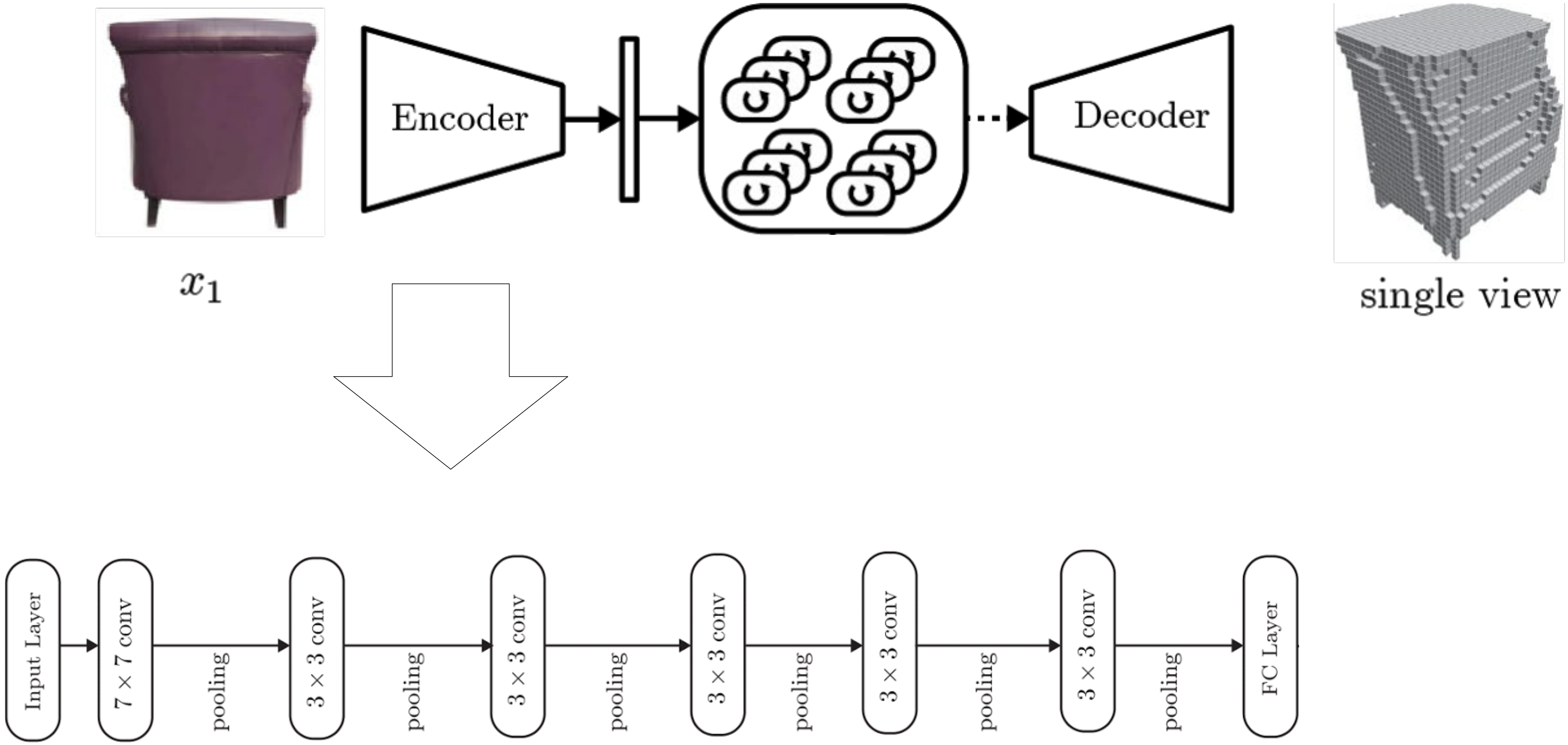
Can we reduce prediction uncertainty by

factoring out the inherent ambiguity of groundtruth?

Input

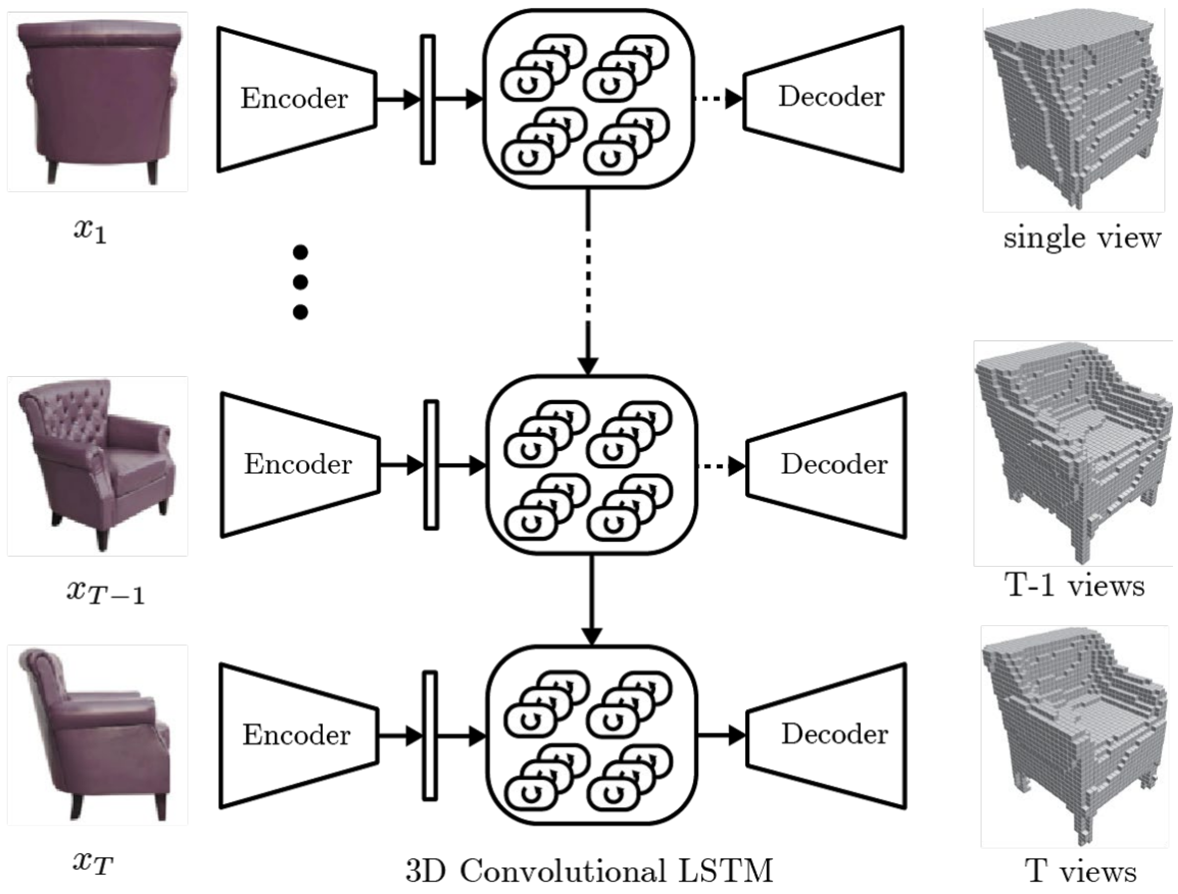
Possible observations from a novel viewpoint

3D Voxel CNNs: Reconstruction from Views



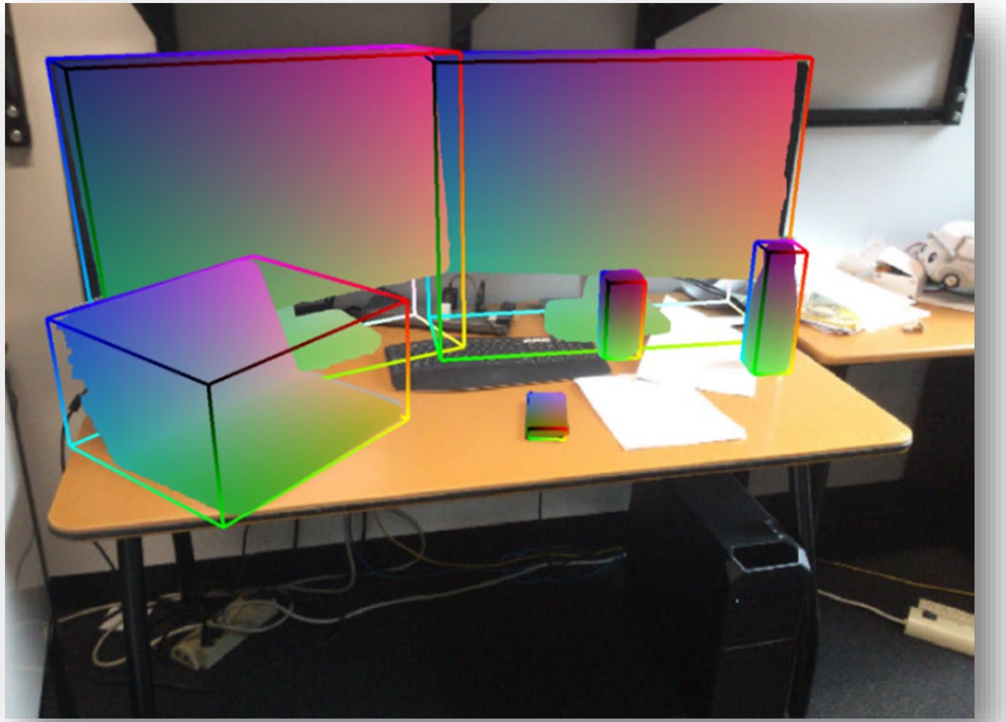
3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction
Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese
ECCV 2016

3D Voxel CNNs: Reconstruction



3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction
Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese
ECCV 2016

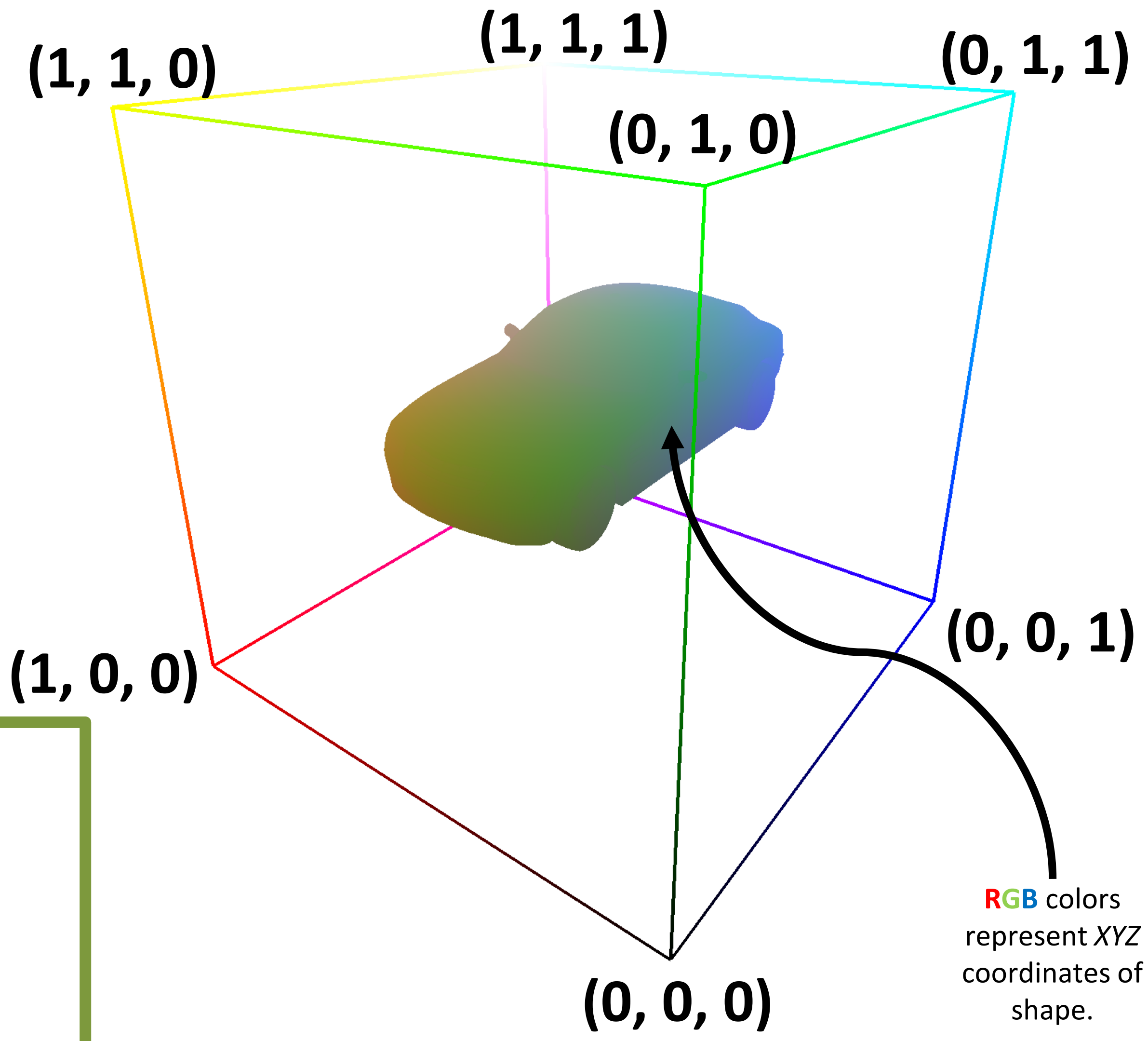
Pose Estimation and View Aggregation



Canonical “Containers” for Object Categories

He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, Leonidas J. Guibas. *Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation*. CVPR 2019.

Normalized Object Coordinate Spaces (NOCS)

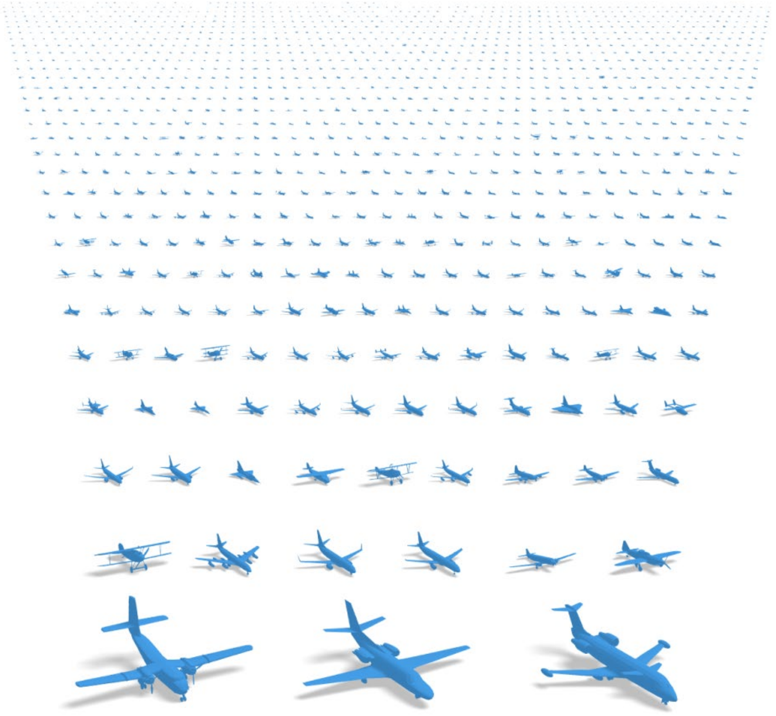


RGB colors represent *XYZ* coordinates of shape.

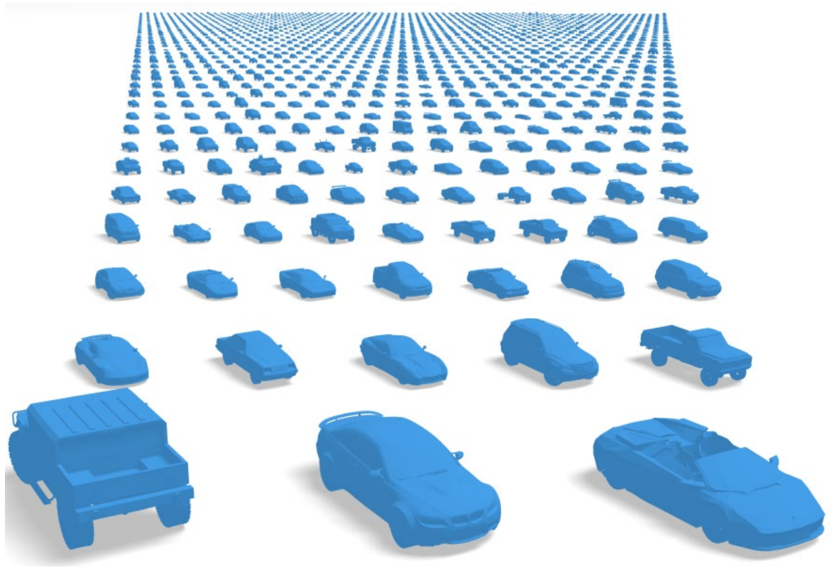
Canonicalize

- Position
- Orientation
- Size

Exploit ShapeNet Category Co-Alignments

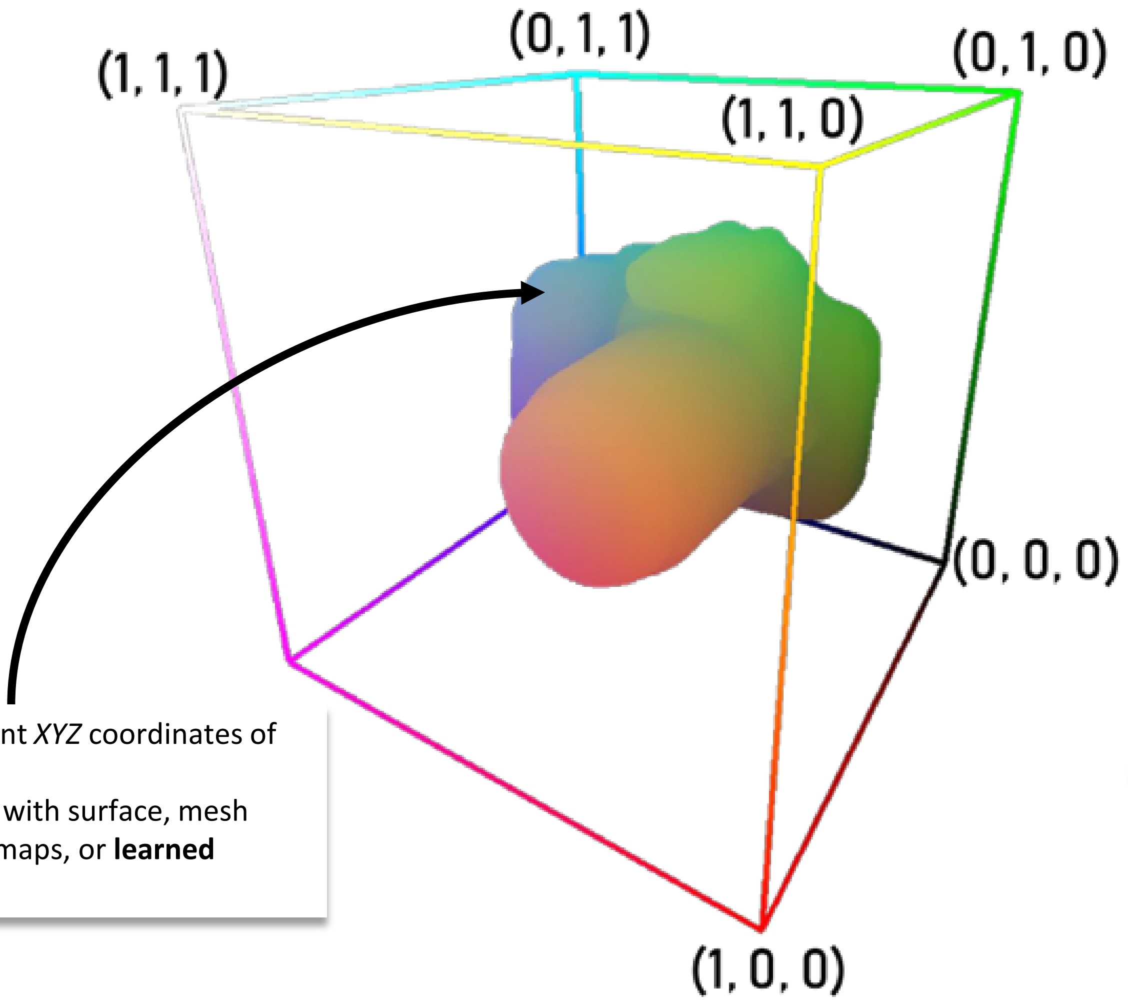


...



Abstract away, pose, size, some intra-category variation

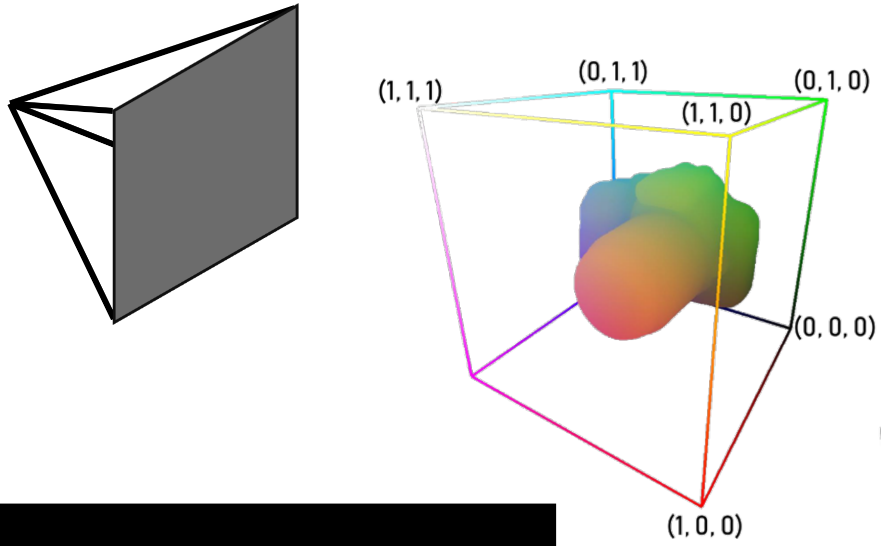
Normalized Object Coordinate Spaces (NOCS)



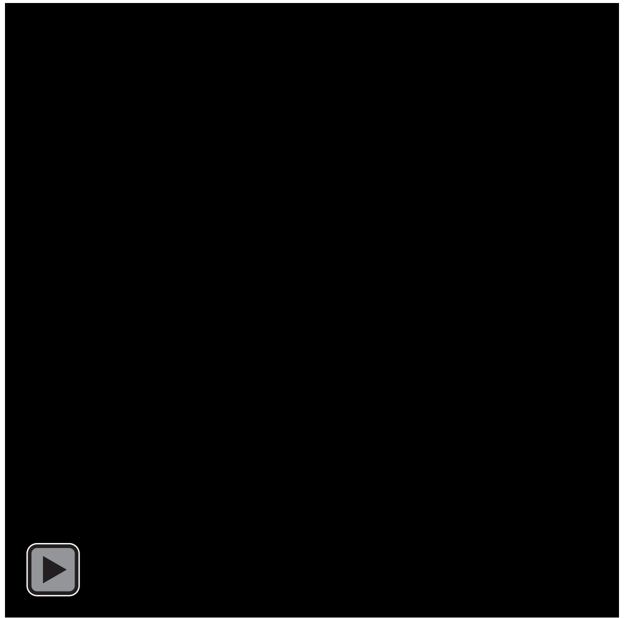
- RGB colors represent *XYZ* coordinates of shape.
- Can be augmented with surface, mesh colors, affordance maps, or **learned features.**

He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, Leonidas J. Guibas. *Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation*. CVPR 2019.

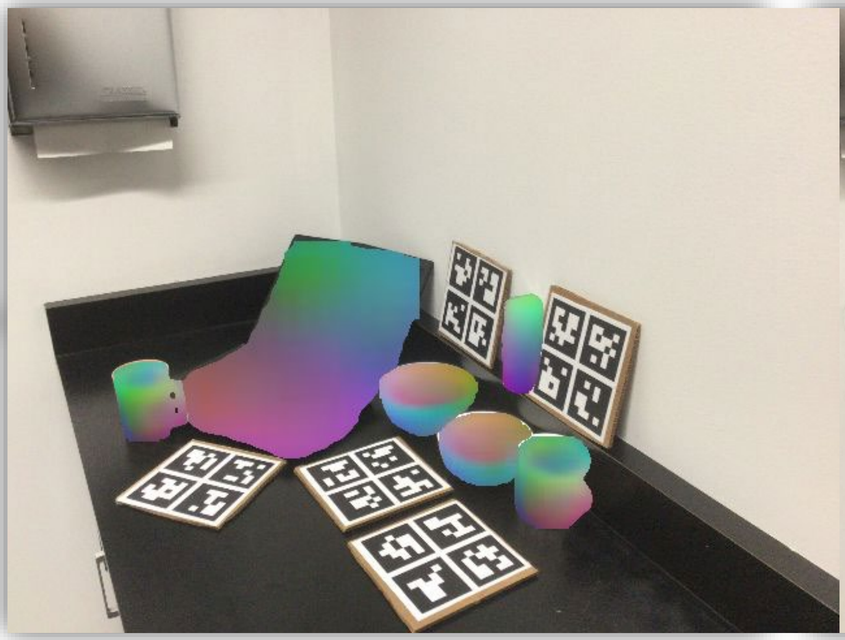
NOCS Lifting Map

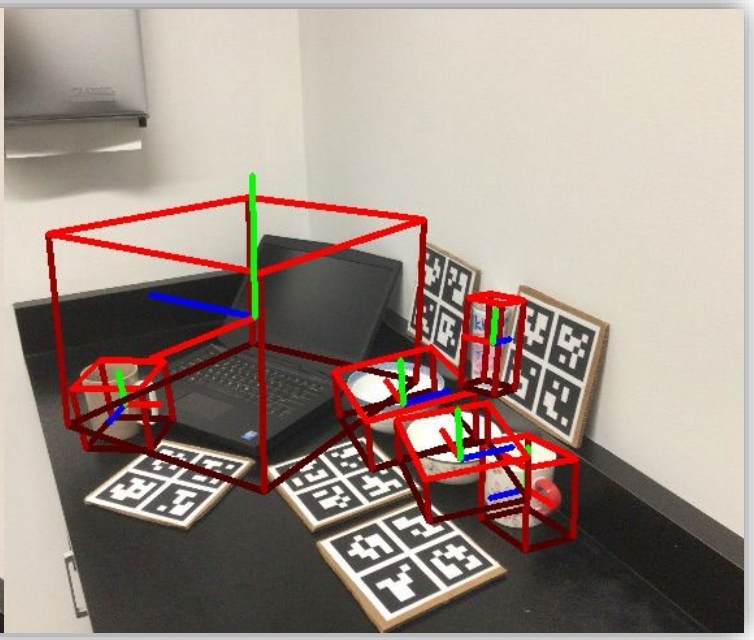


Readout



NOCS for 9D Object Pose and Size Estimation





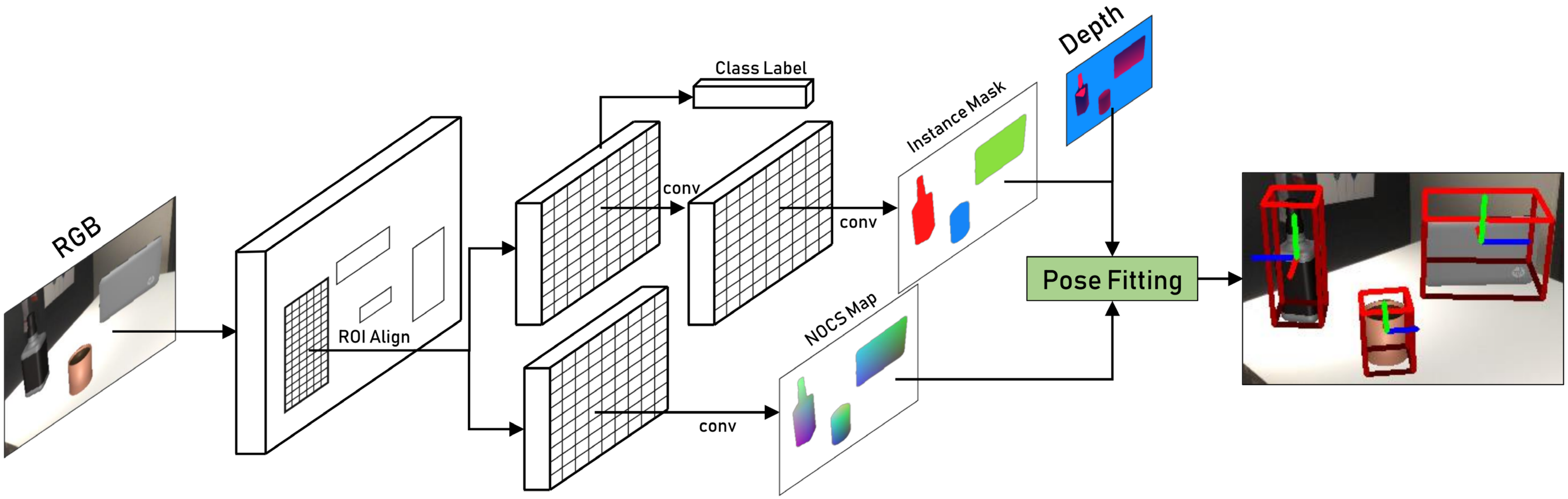
<u>Input</u>: RGB-D Image + Category-Level CAD Model Repository

<u>Output</u>: 3 degrees Translation + 3 Rotation + 3 Size (9 DoF)

<u>No object-specific CAD model</u>

An Image-Centric Approach: Build on Mask-RCNN

Include depth info

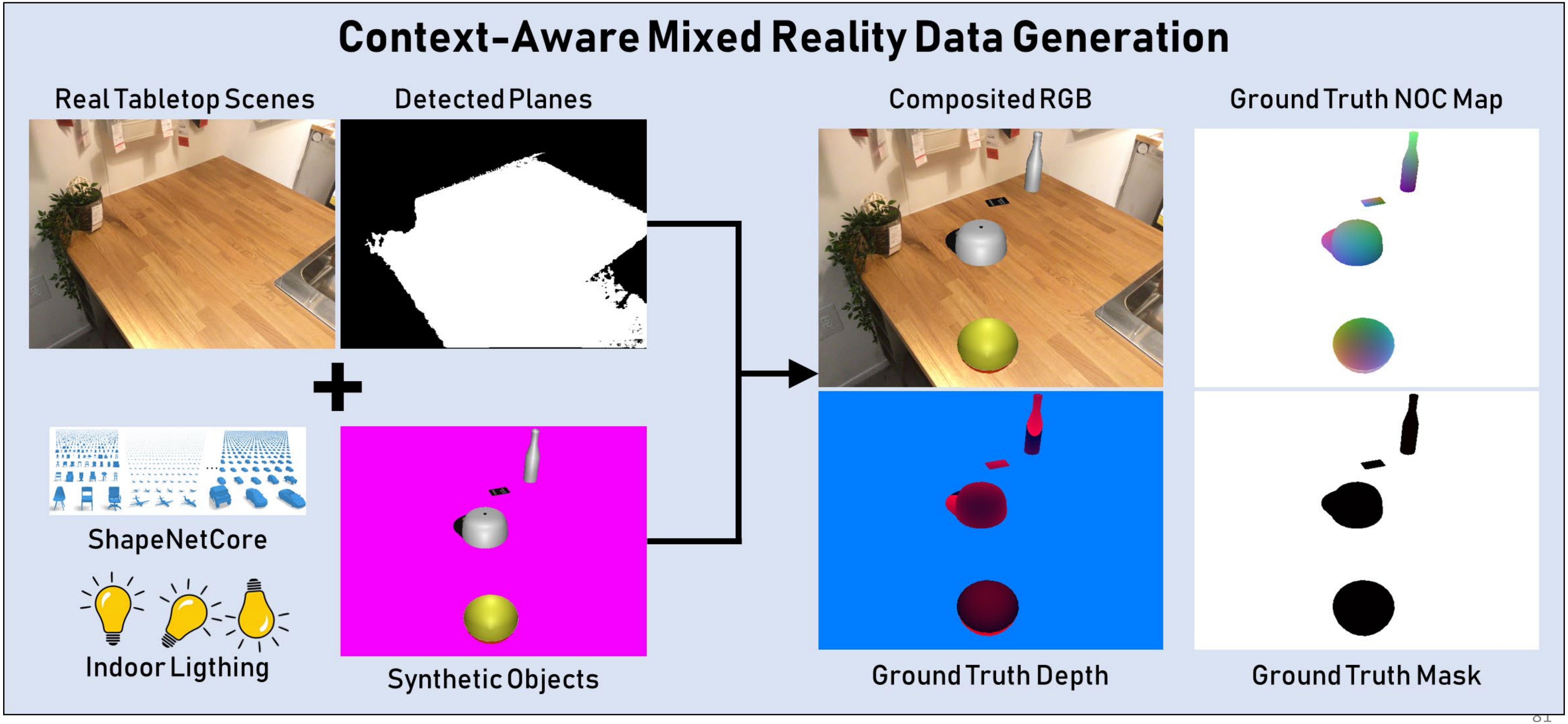


A Mask-RCNN-based backbone

Category-Level Pose

Category-level object pose can be defined for each category up to the limit of global symmetry in the category.

Context-Aware MixEd ReAlity (CAMERA) Dataset



CAMERA Dataset

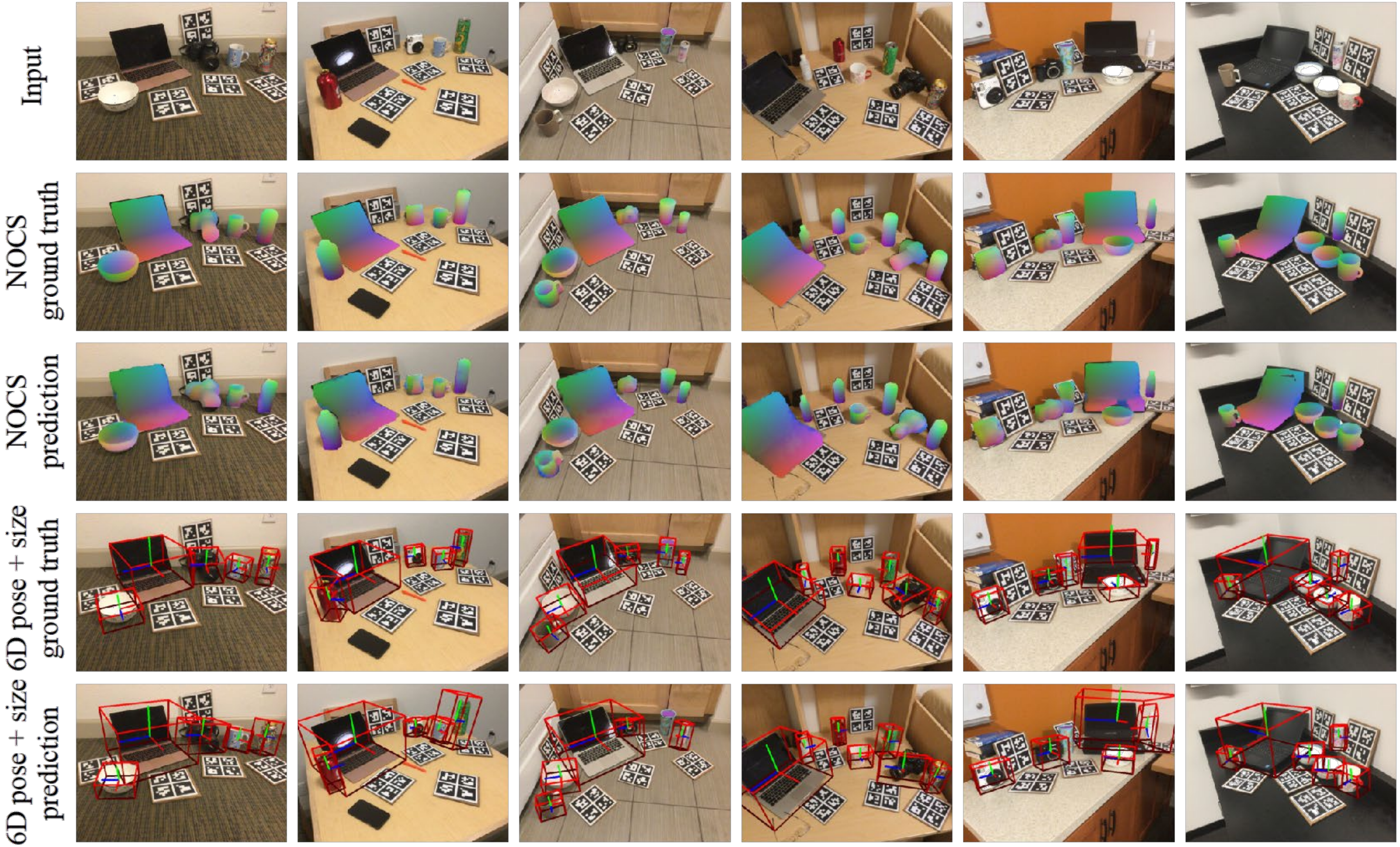
- **300K** mixed reality images are generated
  - 275K training
  - 25K validation
- **31 scenes** captured from IKEA as real backgrounds
  - 27 scenes for training
  - 4 scene for validation
  - 553 images
- **6 object categories**— bottle, bowl, camera, can, laptop and mug
  - 1085 models, 184 for validation
- Distractor objects

Real Dataset

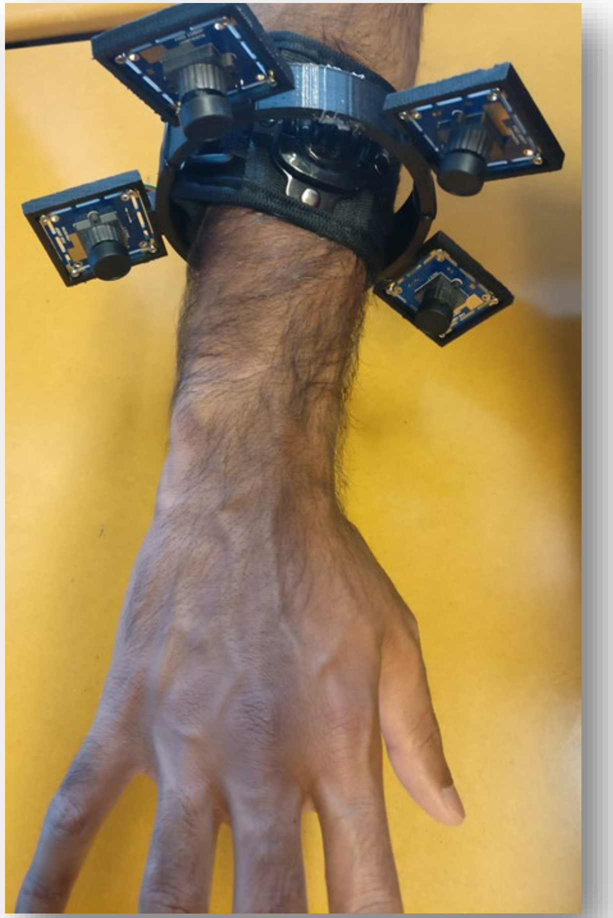
- 8K RGB-D frames
  - training/validation/testing
  - 4300/950/2750
- 18 different real scenes
  - training/validation/testing
  - 7/5/6
- 42 unique instances
  - 7 per category
  - training/validation/testing
  - 3/1/3

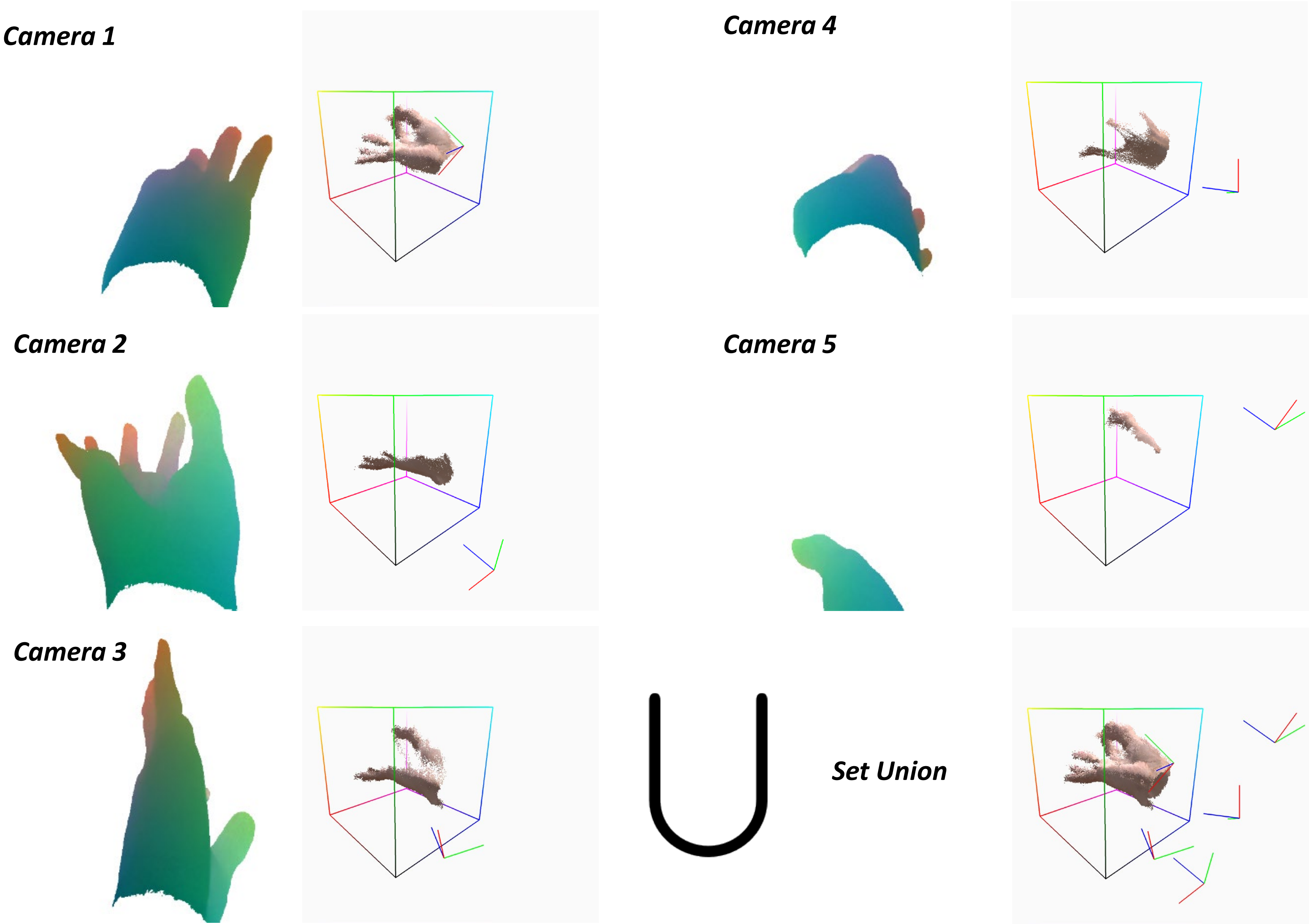
Plus COCO images without pose annotation

Qualitative Results: Real Data

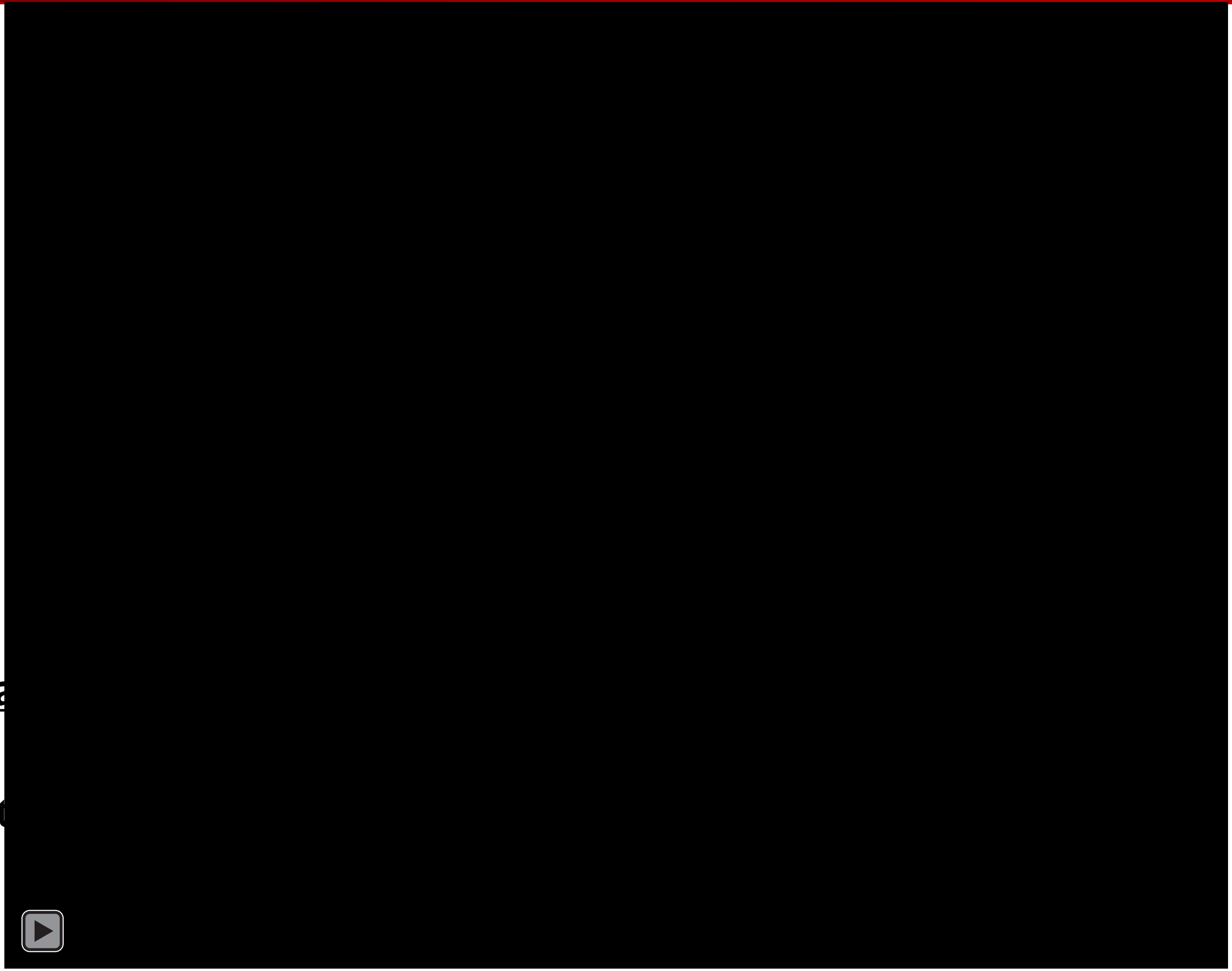


Multi-View Aggregation as Set Union





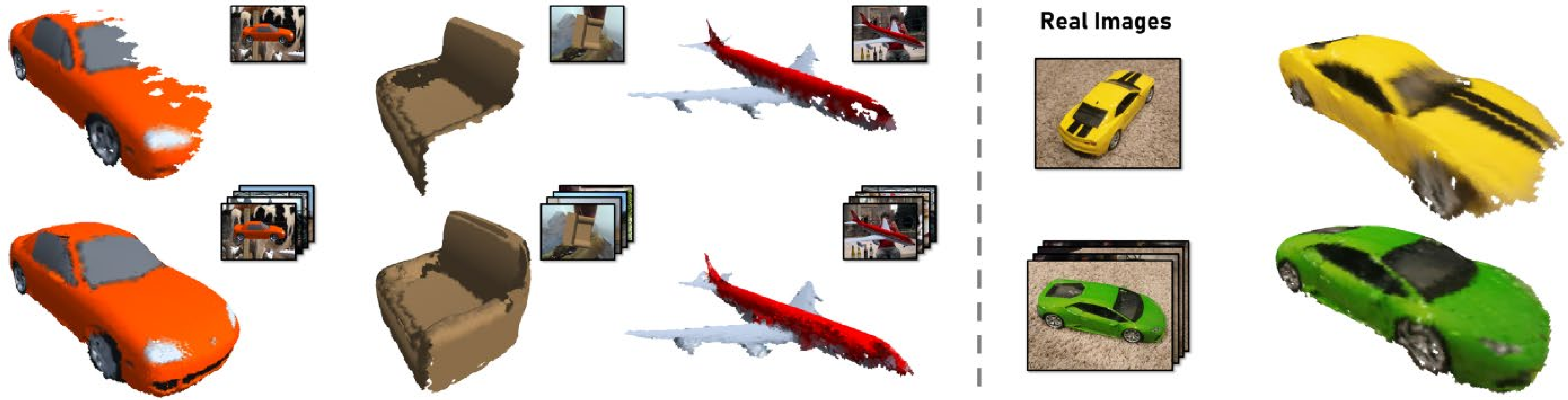
Multi-View NOCS Aggregation



Limita

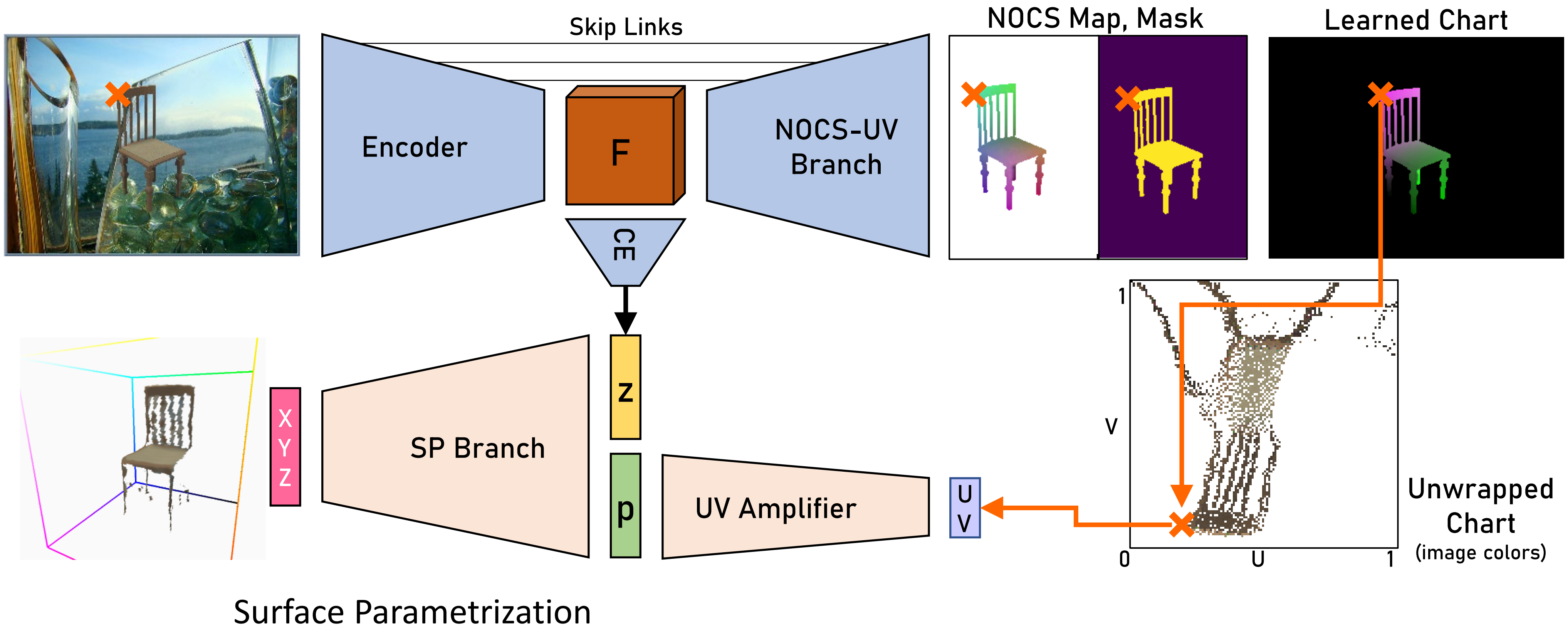
- Set union of t
- No surface

Pix2Surf: Single-View Single-Chart



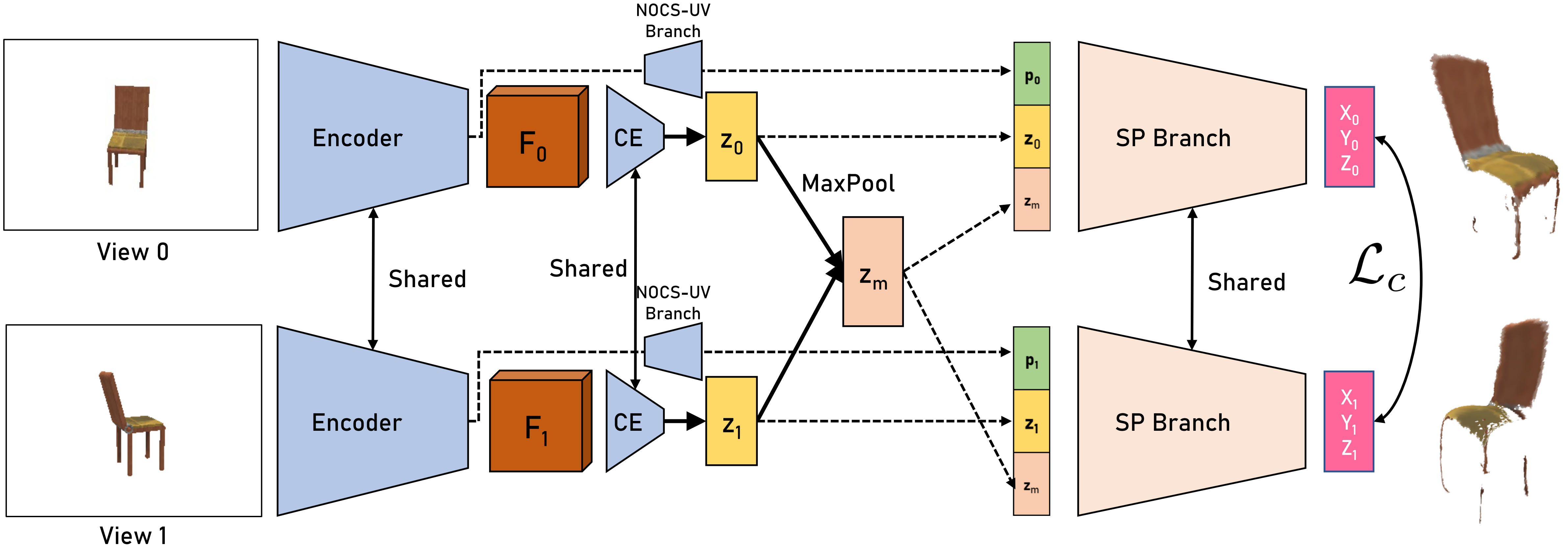
Lei, J., Sridhar, S., Guerrero, P., Sung, M., Mitra, N. and Guibas, L.J. Pix2surf: Learning parametric 3D surface models of objects from images. ECCV 2020.

Pix2Surf: Single-View Single-Chart



Lei, J., Sridhar, S., Guerrero, P., Sung, M., Mitra, N. and Guibas, L.J. Pix2surf: Learning parametric 3D surface models of objects from images. ECCV 2020.

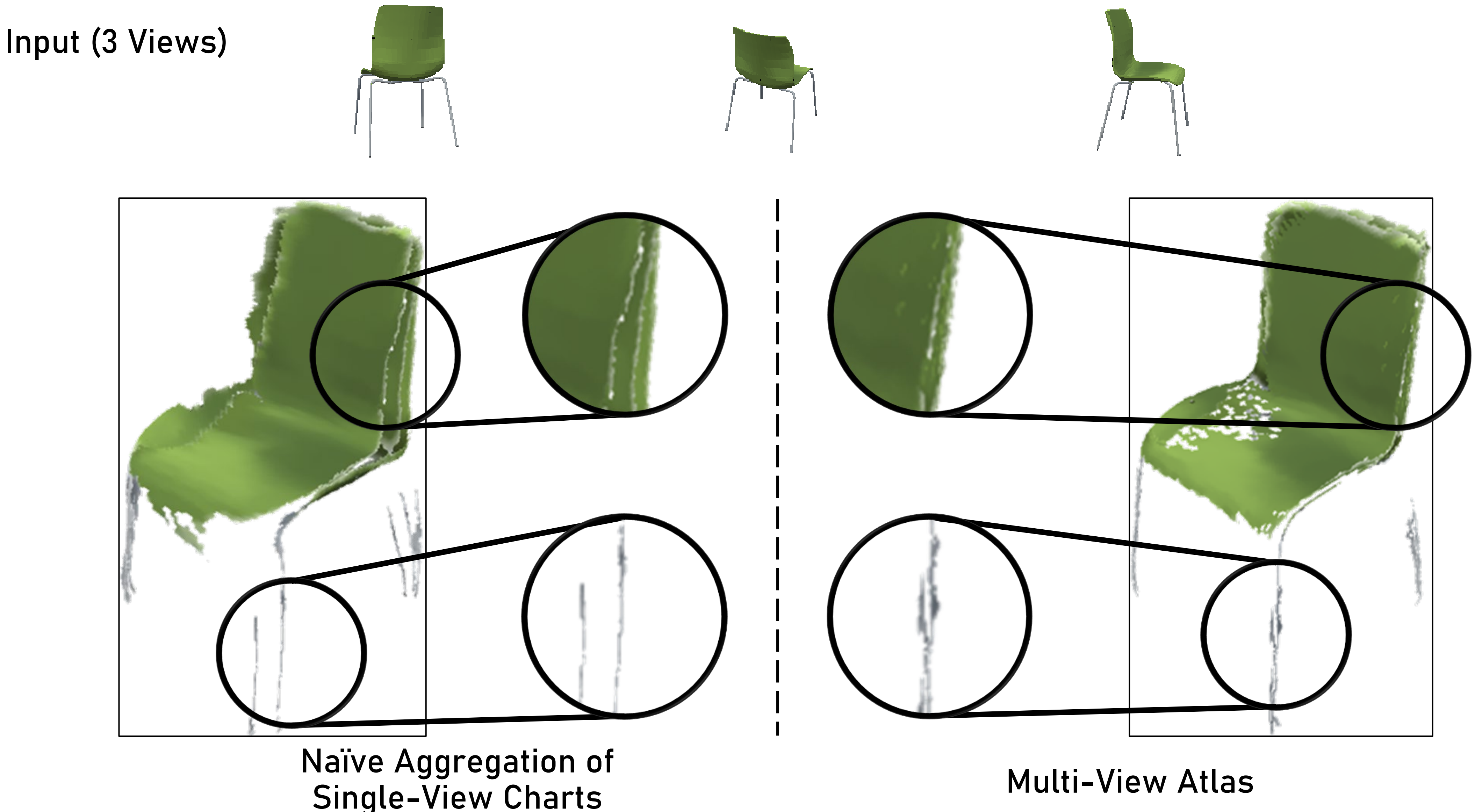
Pix2Surf: Multi-View Atlas



Multi-View consistency loss

$$L_c = \frac{1}{|P|}\sum_{(i,j)\in P} ||x_i - x_j||_2$$

Multi-View Consistency Loss



Emergent Properties

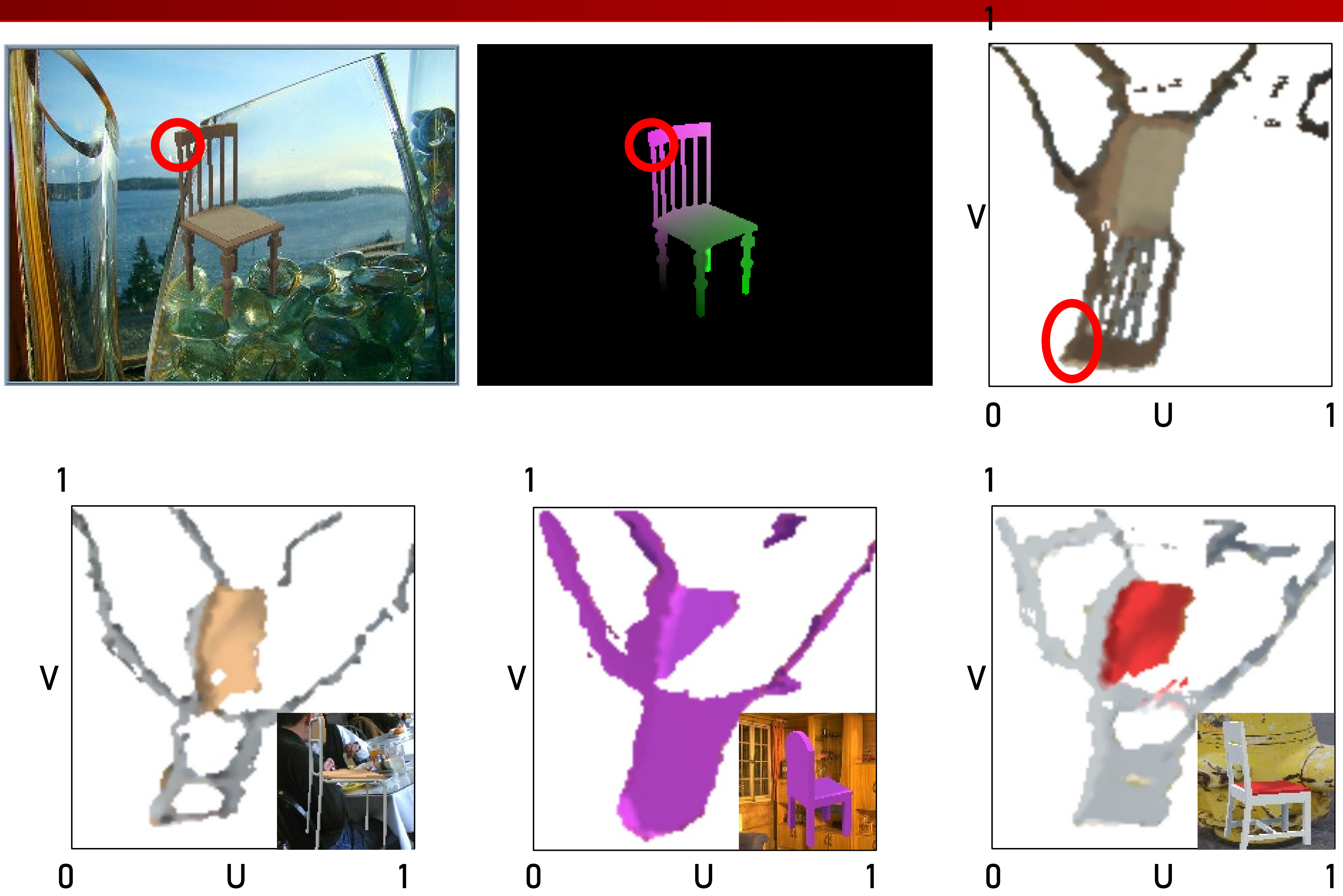


Image 2 Shape from Unseen Classes

Learning to Reconstruct Shapes from Unseen Classes






Xiuming Zhang[1]*

Zhoutong Zhang[1]*

Chengkai Zhang[1]

Joshua B. Tenenbaum[1]

William T. Freeman[1,2]

Jiajun Wu[1]

MIT CSAIL

[1] MIT CSAIL

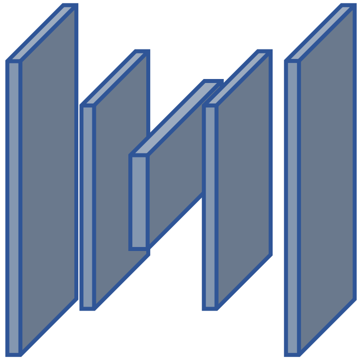
[2] Google Research

NeurIPS 2018

* indicates equal contribution

Formulation

Image I



Neural Network f_θ

Shape V

Training: $\text{argmin}_\theta \text{ Loss}(f_\theta(I), V)$

DRC* [Tulsiani et al., CVPR ’17]

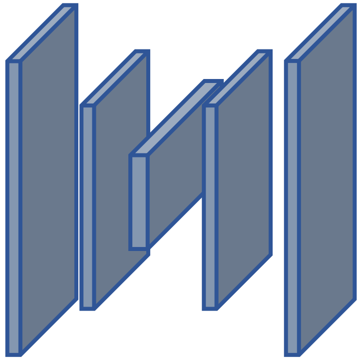
[Differentiable Ray Consistency]

AtlasNet* [Groueix et al., CVPR ’18]

*Trained on cars, chairs, airplanes

Formulation

Image I



Neural Network f_θ

Shape V

Training: $\mathrm{argmin}_\theta \; \mathrm{Loss}(f_\theta(I), V)$

Directly regularize $f_\theta(I)$ by adding inductive biases

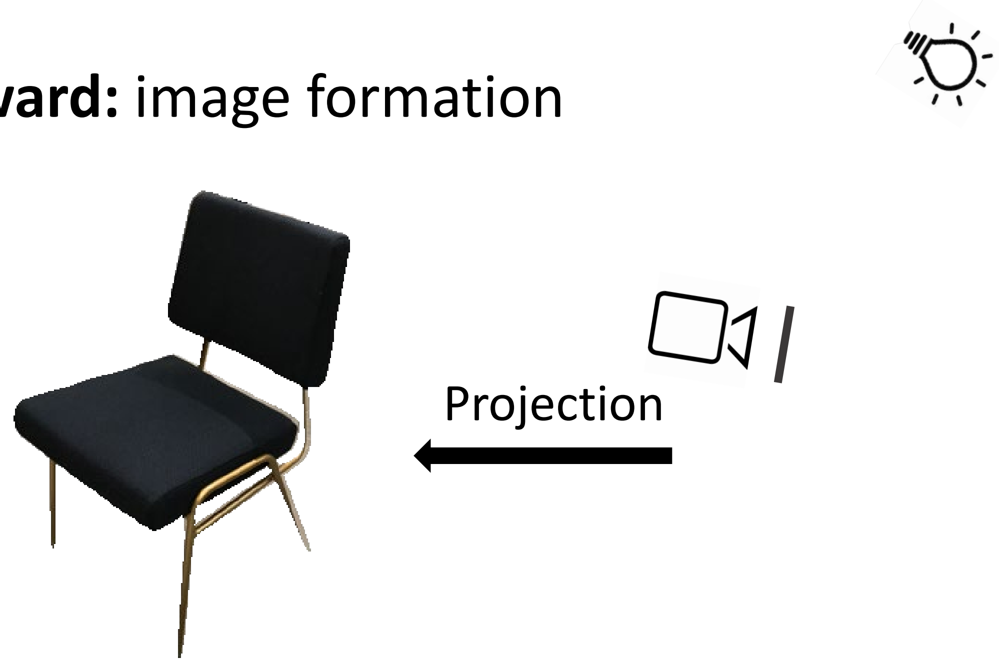
DRC* [Tulsiani et al., CVPR '17]

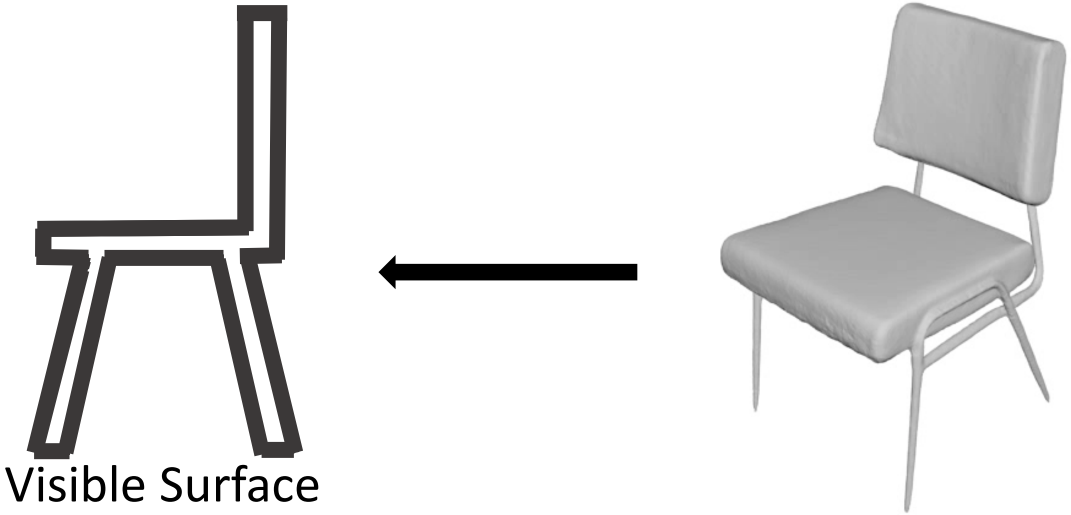
AtlasNet* [Groueix et al., CVPR '18]

*Trained on cars, chairs, airplanes

What is the proper inductive bias of $f_\theta(I)$?

Forward: image formation

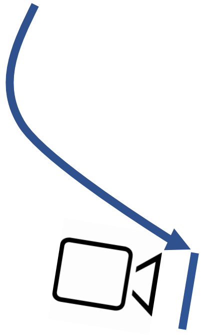




Inverse: shape estimation

Depth to Shape?

Image I | Neural Network f_θ | Depth $\boldsymbol{D}$ | Neural Network g_θ | Shape V



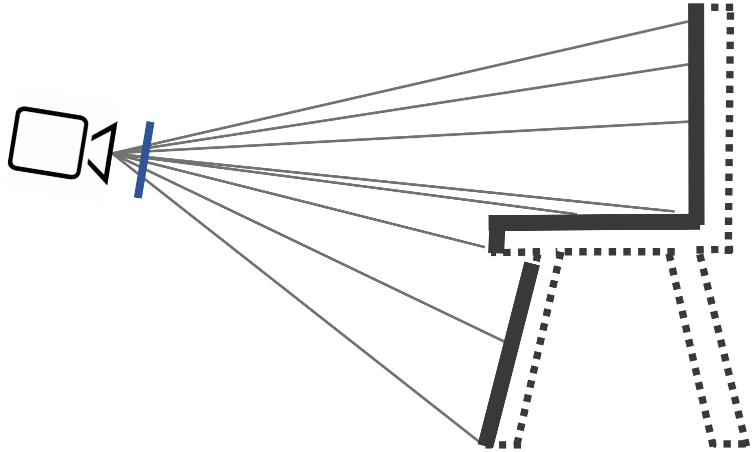
Neural network g_θ is over-parameterized:
g_θ has to learn a deterministic mapping!

Projecting depth into 3D is a deterministic, fully differentiable process!

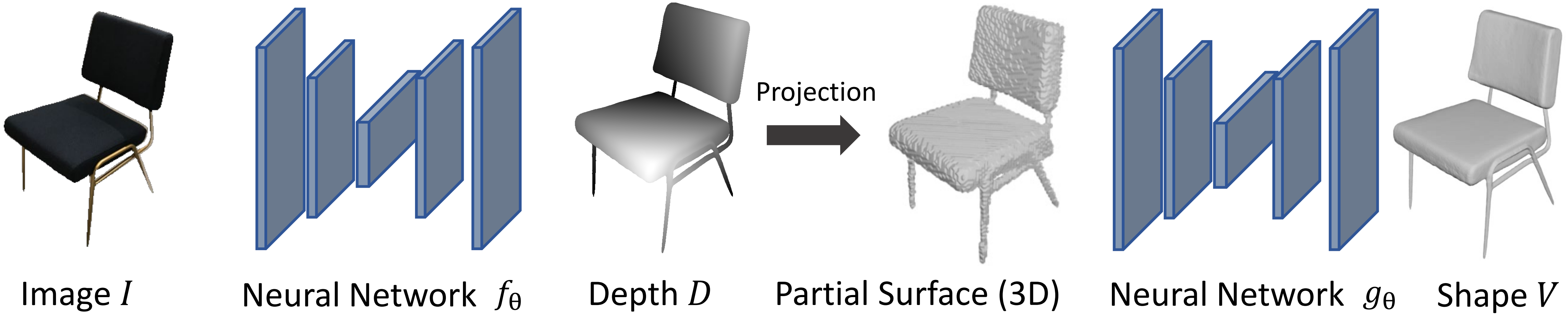
MarrNet [Wu et at., NIPS '17]

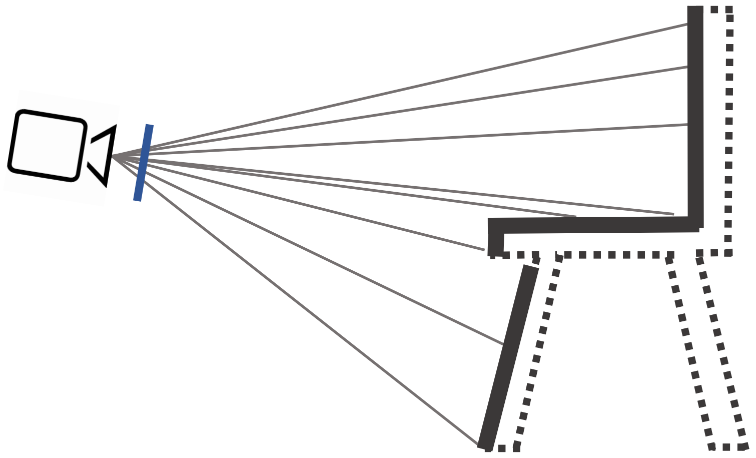
Depth to Shape?

Image I Neural Network f_θ Depth D Neural Network g_θ Shape V

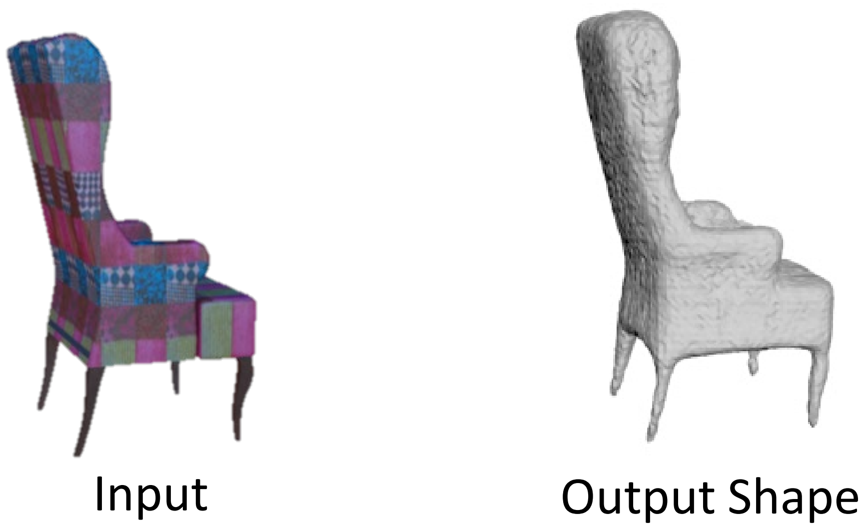


Our approach: Generalizable Reconstruction (GenRe)

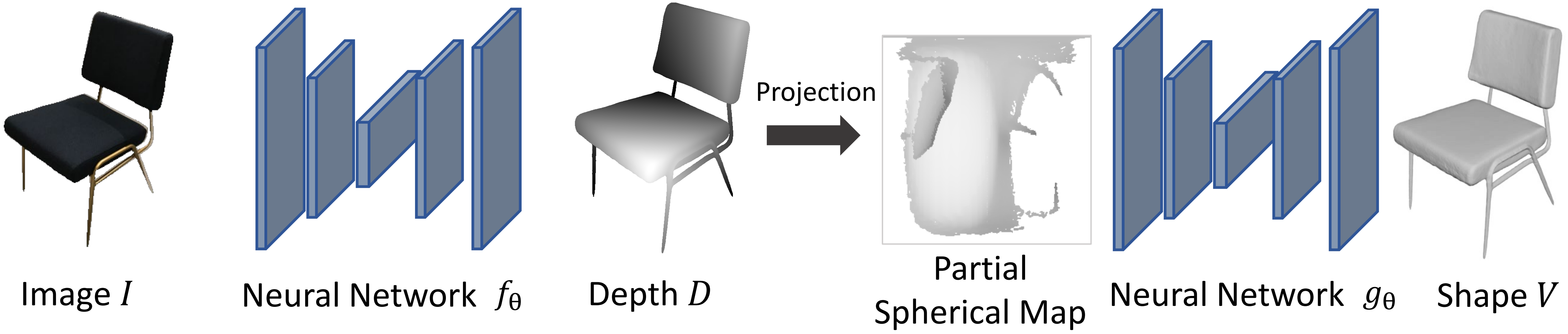




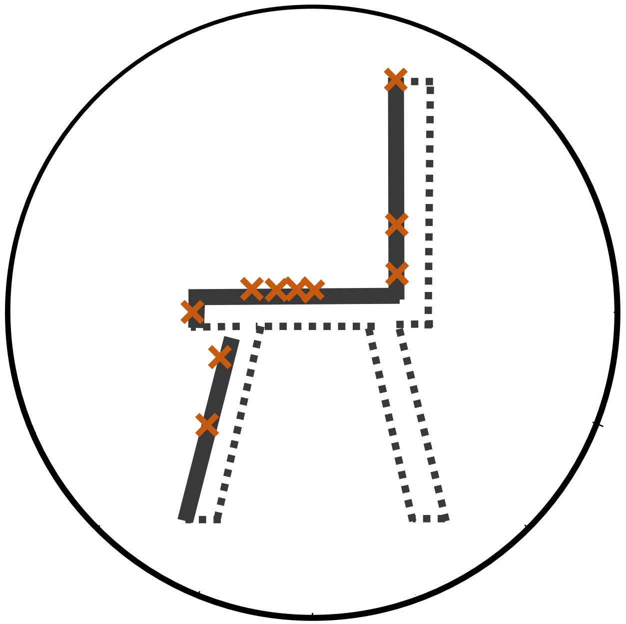
Partial surface in 3D is very sparse, which makes it hard for g_θ to capture surface features.



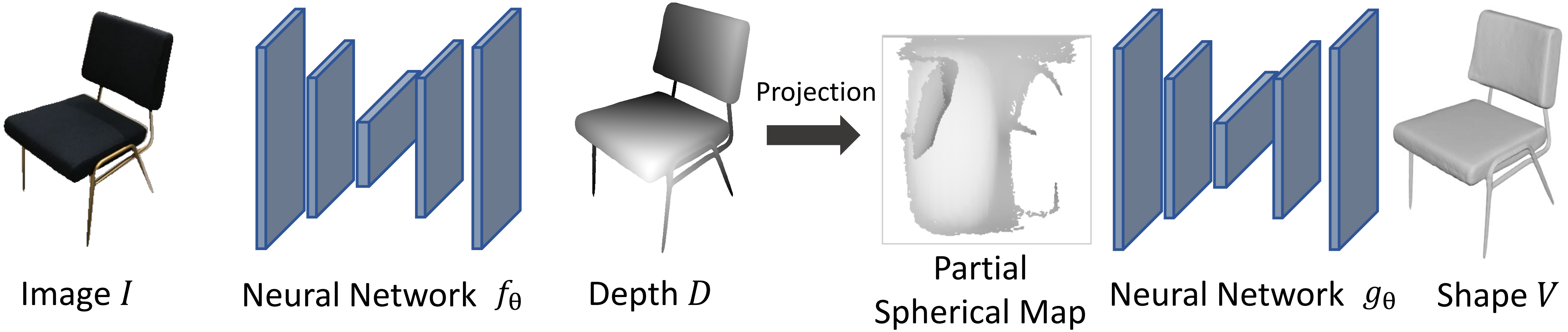
Our approach: **Gen**eralizable **Re**construction (**GenRe**)



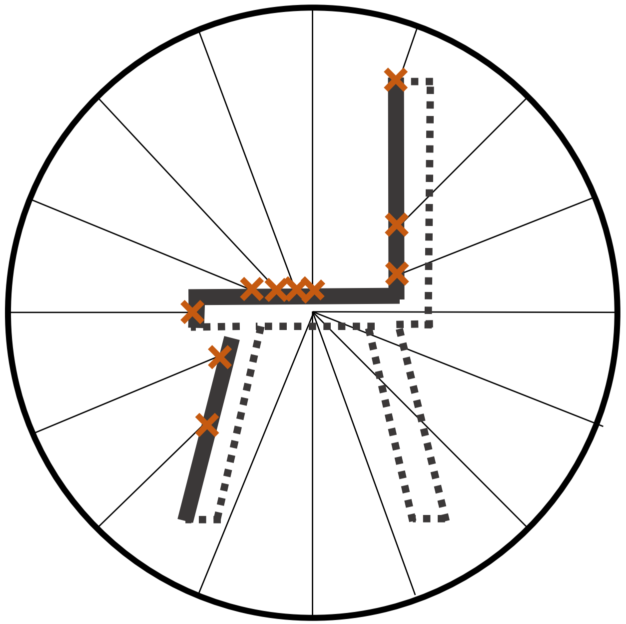
Spherical map as a surrogate representation for surfaces in 3D



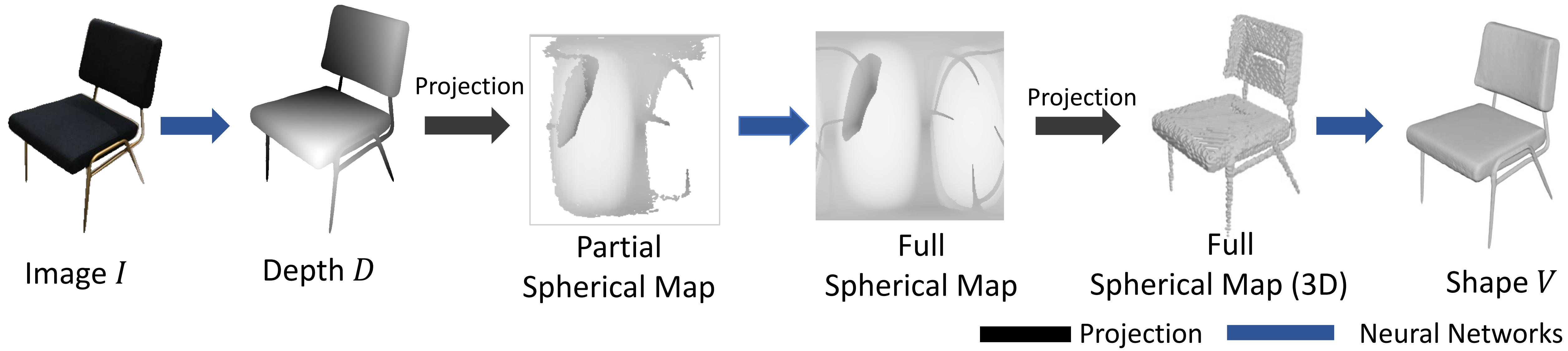
Our approach: Generalizable Reconstruction (GenRe)

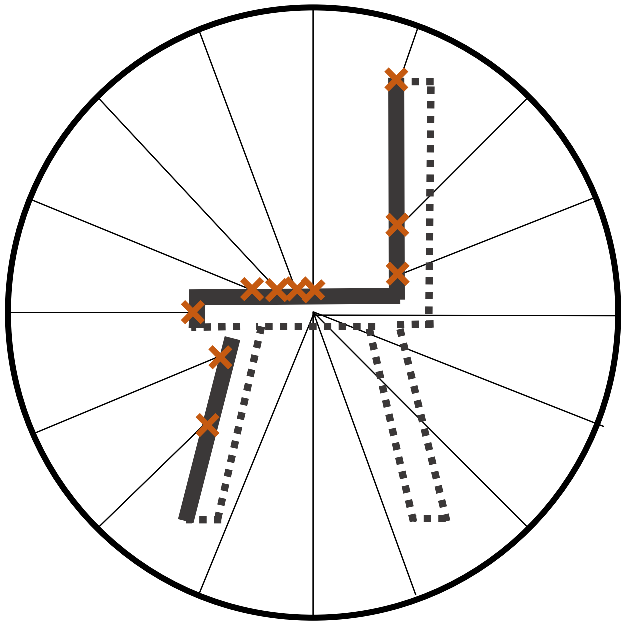


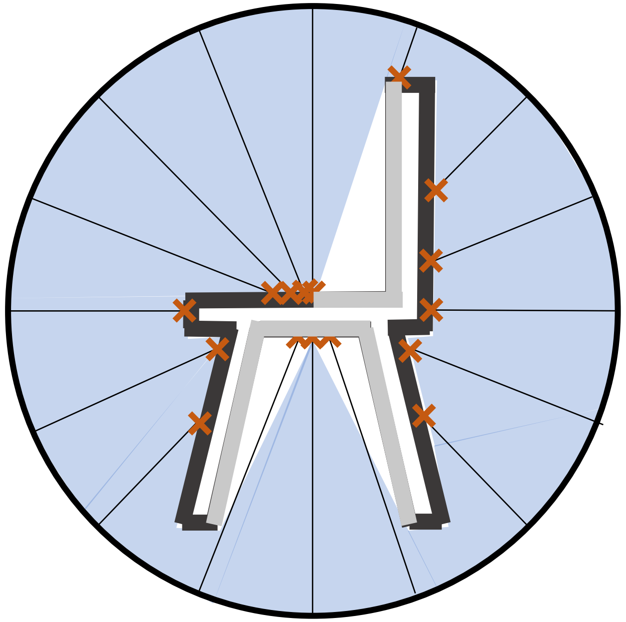
Spherical map as a surrogate representation for surfaces in 3D



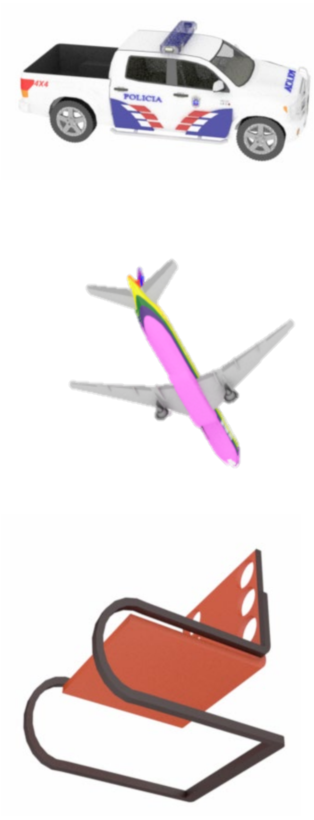
Our approach: Generalizable Reconstruction (GenRe)



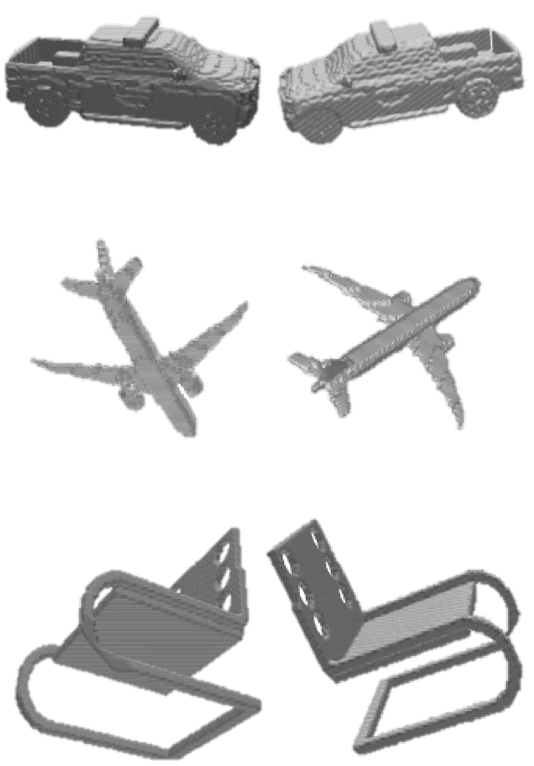




Results: Testing on the Training Classes



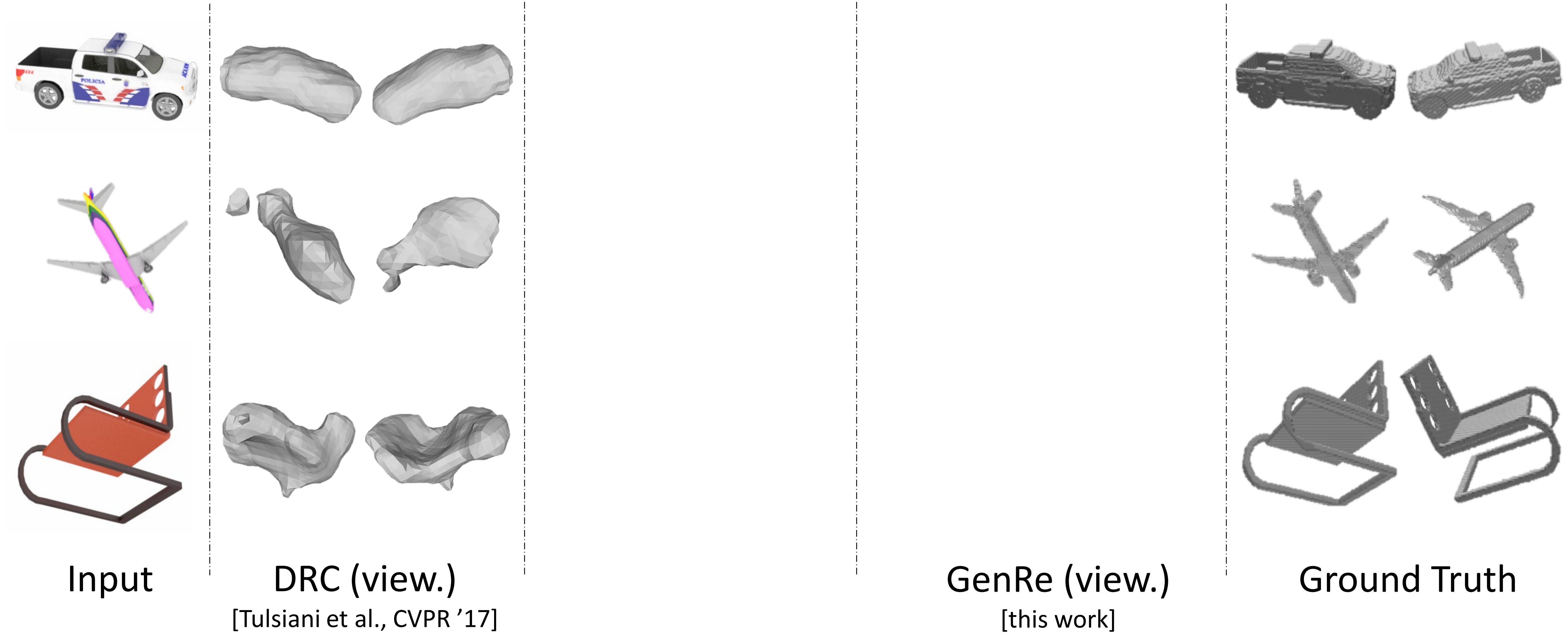
Input



Ground Truth

Training & Testing: cars, chairs, airplanes

Results: Testing on the Training Classes



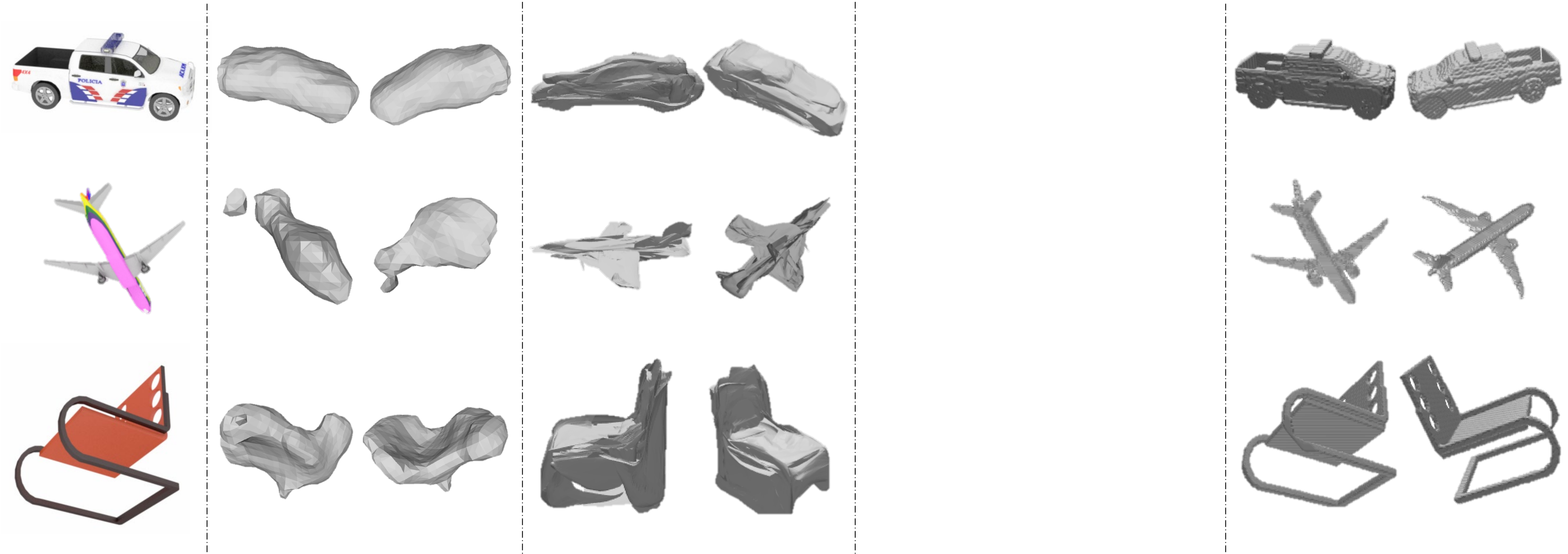
Training & Testing: cars, chairs, airplanes

Results: Testing on the Training Classes

Input | DRC (view.) [Tulsiani et al., CVPR '17] | AtlasNet (obj.) [Groueix et al., CVPR '18] | GenRe (view.) [this work] | Ground Truth

Training & Testing: cars, chairs, airplanes

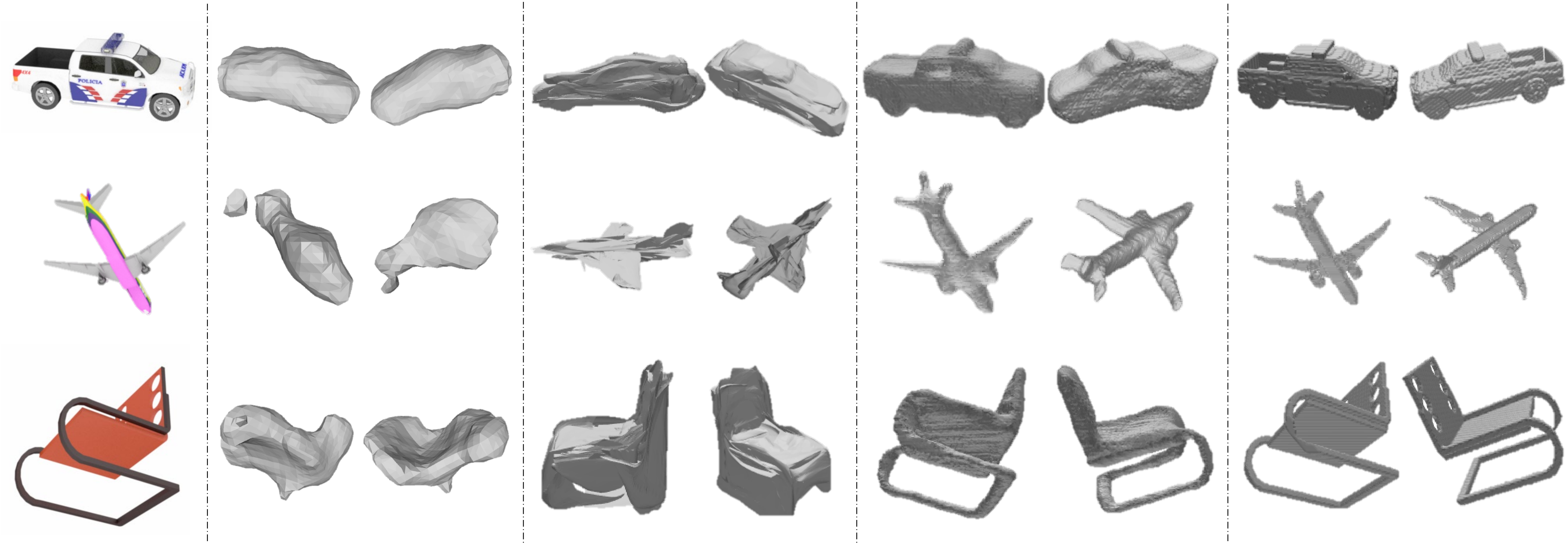
Results: Testing on the Training Classes



Input | DRC (view.) [Tulsiani et al., CVPR '17] | AtlasNet (obj.) [Groueix et al., CVPR '18] | GenRe (view.) [this work] | Ground Truth

Training & Testing: cars, chairs, airplanes

Results: Testing on the Training Classes



Input

DRC (view.)
[Tulsiani et al., CVPR '17]

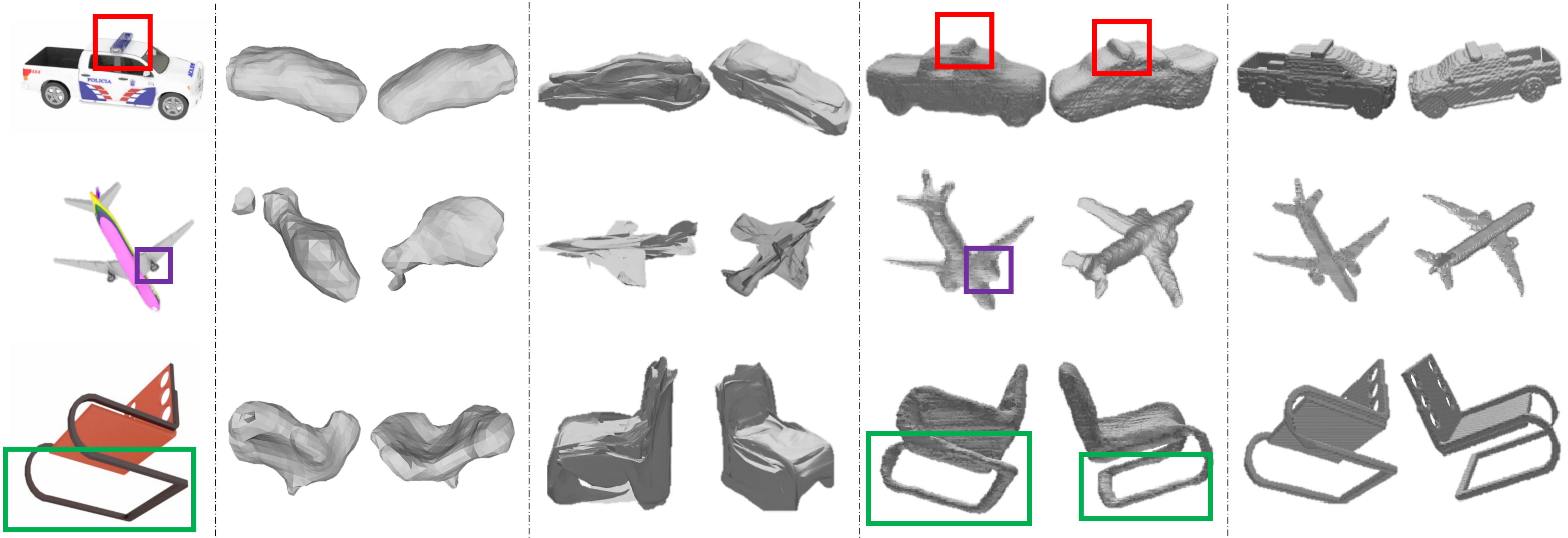
AtlasNet (obj.)
[Groueix et al., CVPR '18]

GenRe (view.)
[this work]

Ground Truth

Training & Testing: cars, chairs, airplanes

Results: Testing on the Training Classes



Input

DRC (view.)
[Tulsiani et al., CVPR '17]

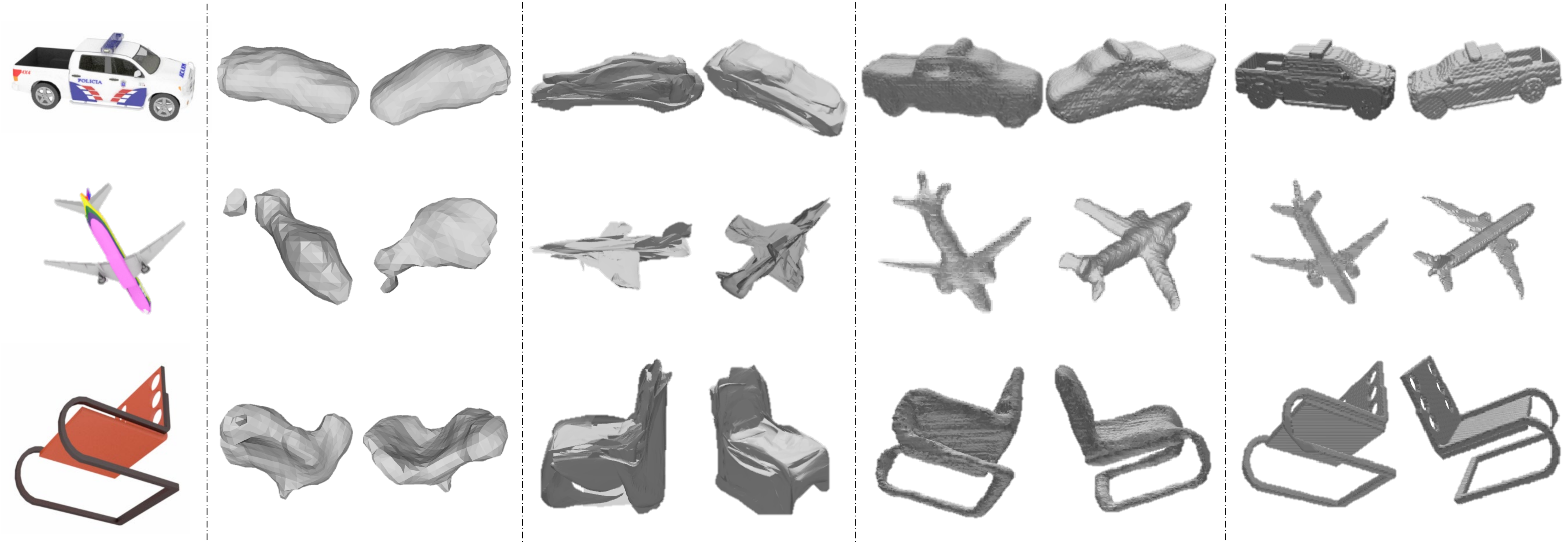
AtlasNet (obj.)
[Groueix et al., CVPR '18]

GenRe (view.)
[this work]

Ground Truth

Training & Testing: cars, chairs, airplanes

Results: Testing on the Training Classes

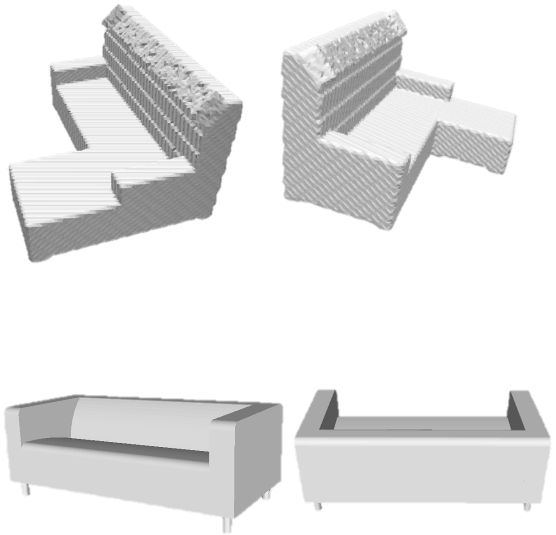


Input | DRC (view.) [Tulsiani et al., CVPR '17] | AtlasNet (obj.) [Groueix et al., CVPR '18] | GenRe (view.) [this work] | Ground Truth

AtlasNet (obj.) is 8% better in Chamfer distance than GenRe

Results: Generalizing to Unseen Classes

Input

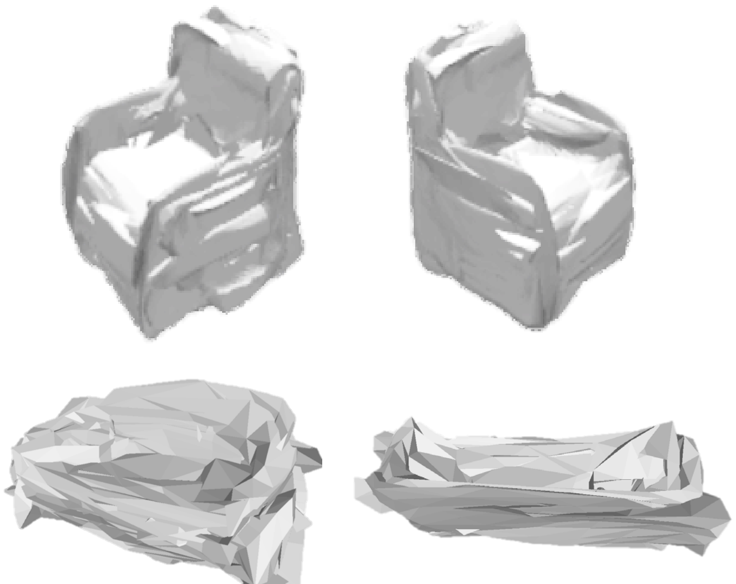


Ground Truth

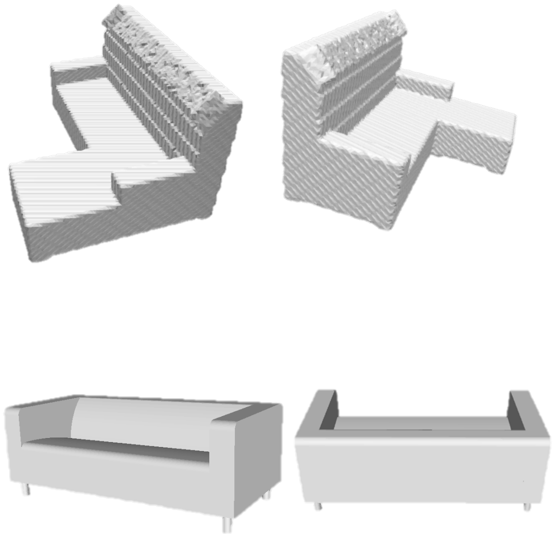
Training: cars, chairs, airplanes
Testing: sofas

Results: Generalizing to Unseen Classes

Input



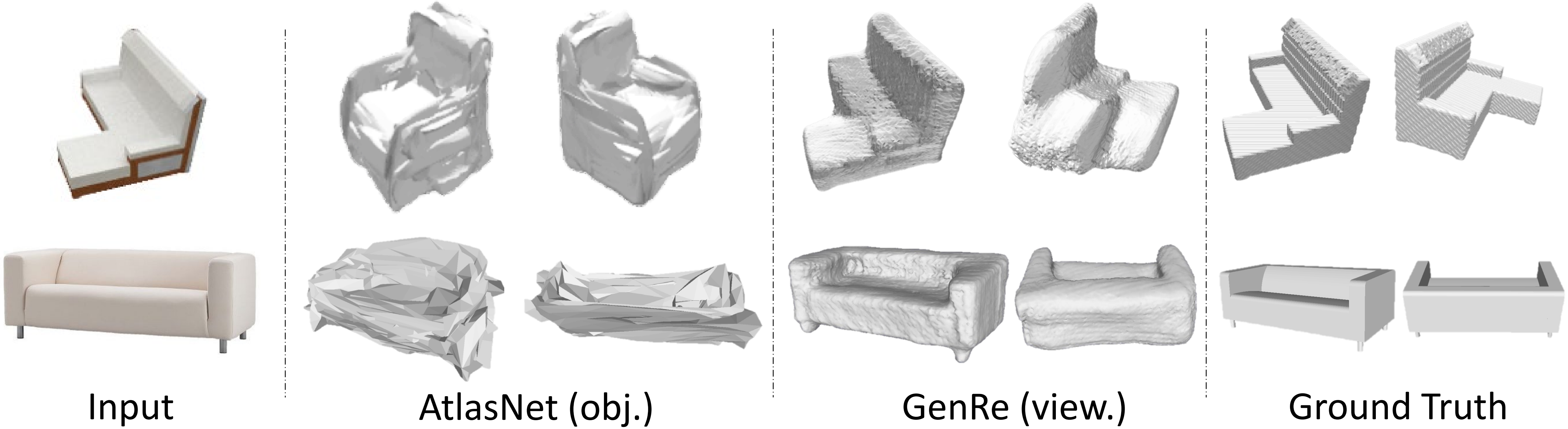
AtlasNet (obj.)



Ground Truth

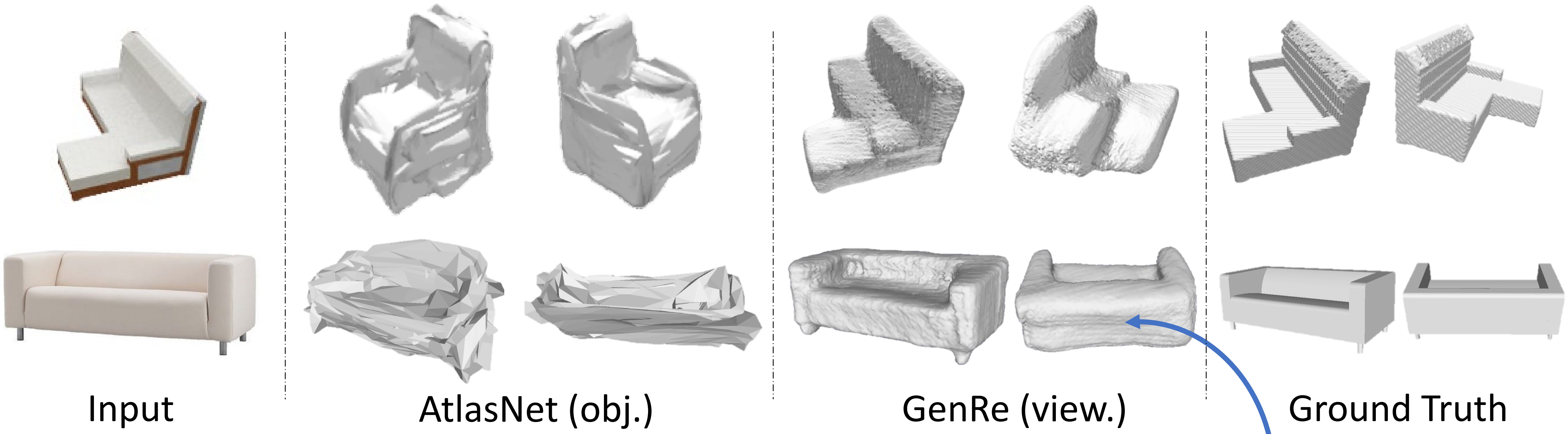
Training: cars, chairs, airplanes
Testing: sofas

Results: Generalizing to Unseen Classes

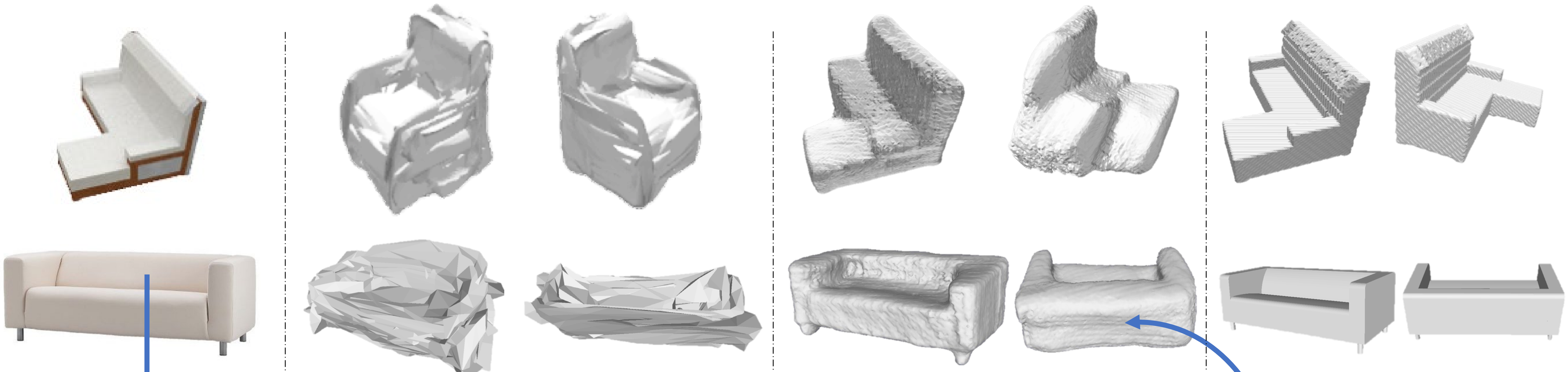


Training: cars, chairs, airplanes
Testing: sofas

Results: Generalizing to Unseen Classes

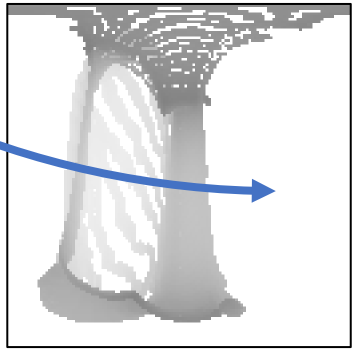


Results: Generalizing to Unseen Classes



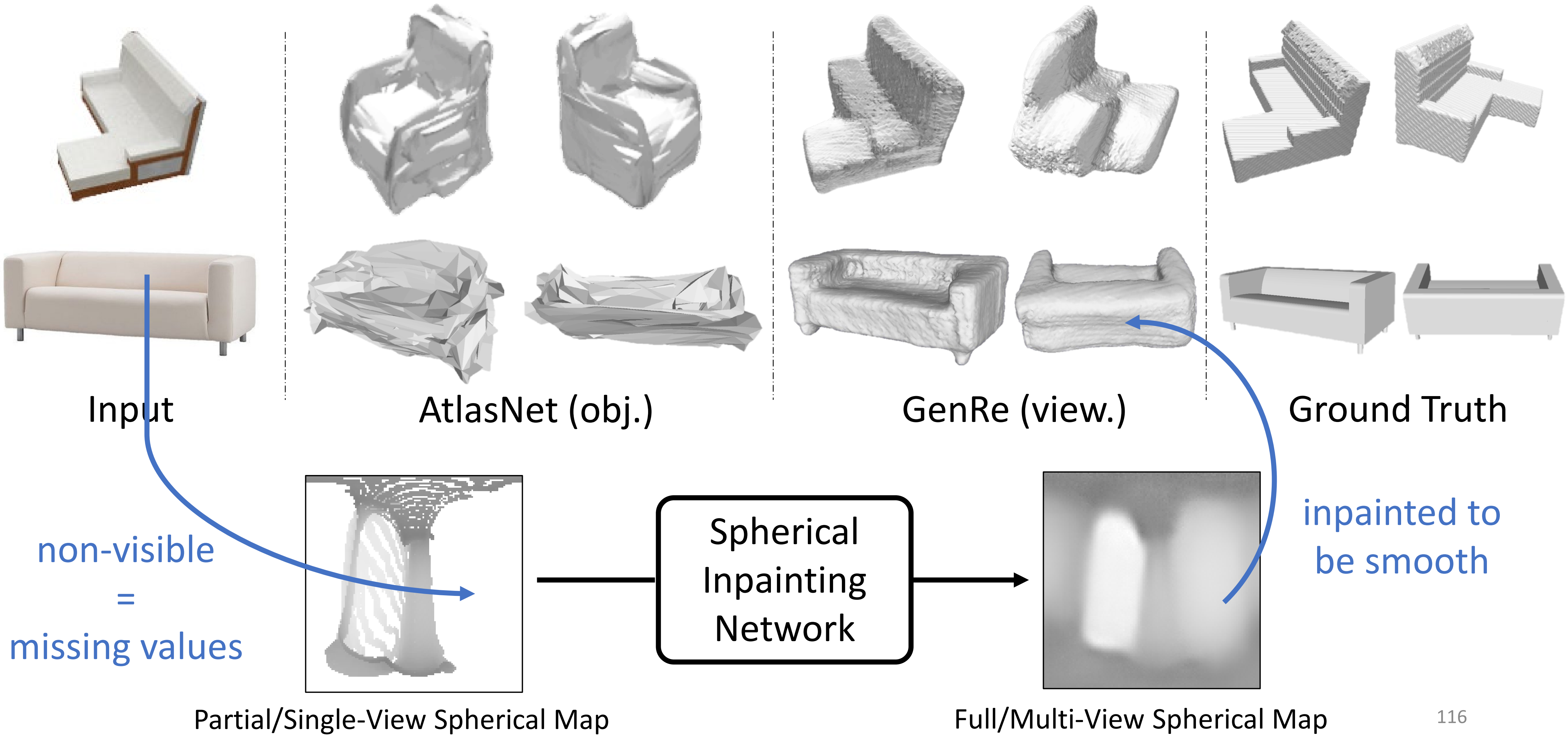
Input | AtlasNet (obj.) | GenRe (view.) | Ground Truth

non-visible
=
missing values

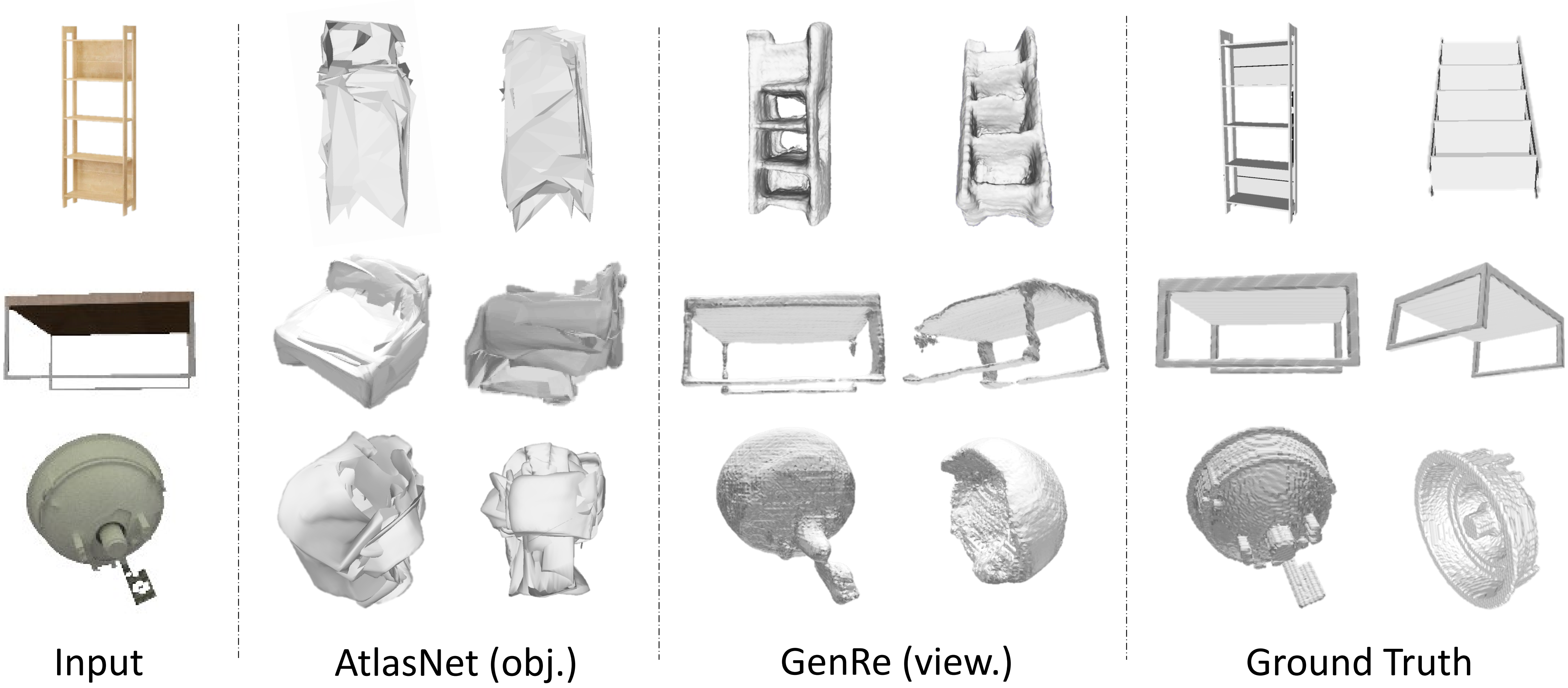


Partial/Single-View Spherical Map

Results: Generalizing to Unseen Classes

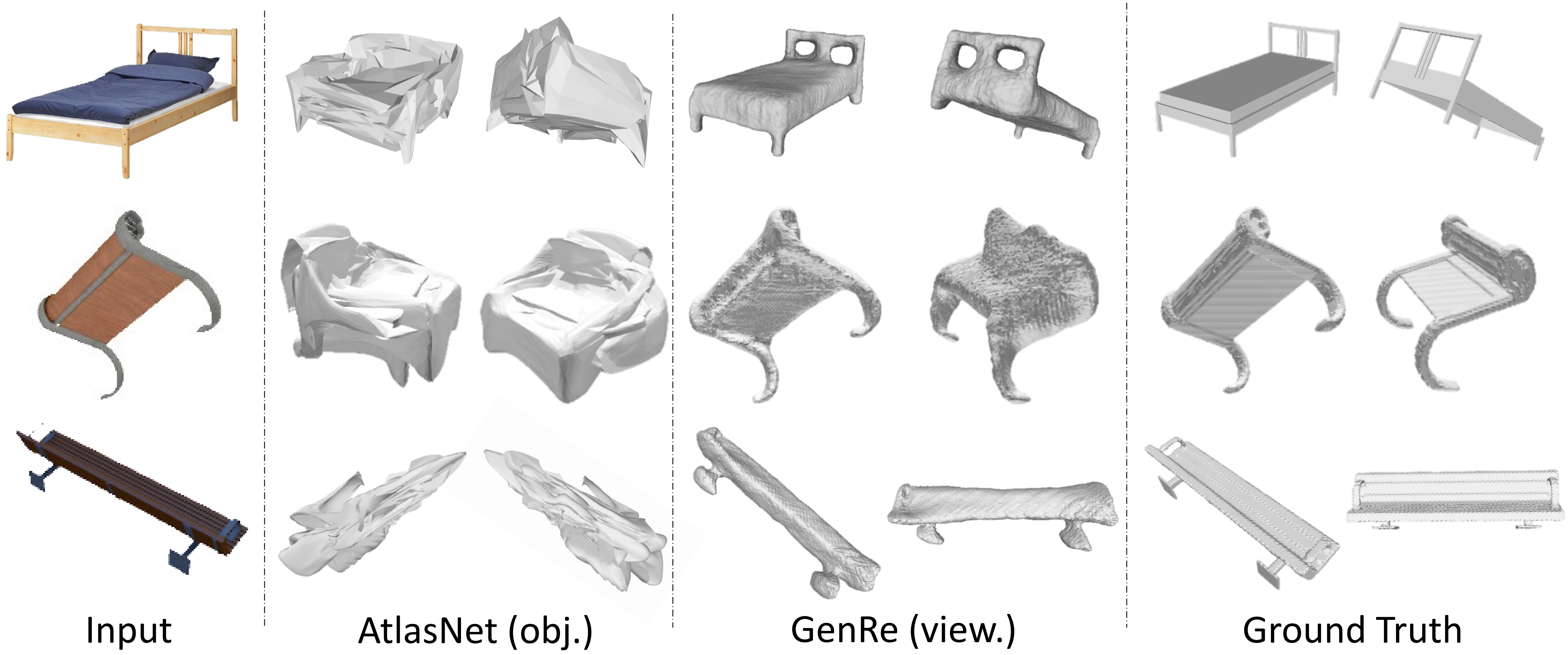


Results: Generalizing to Unseen Classes



Training: cars, chairs, airplanes
Testing: bookcases, tables, loudspeakers

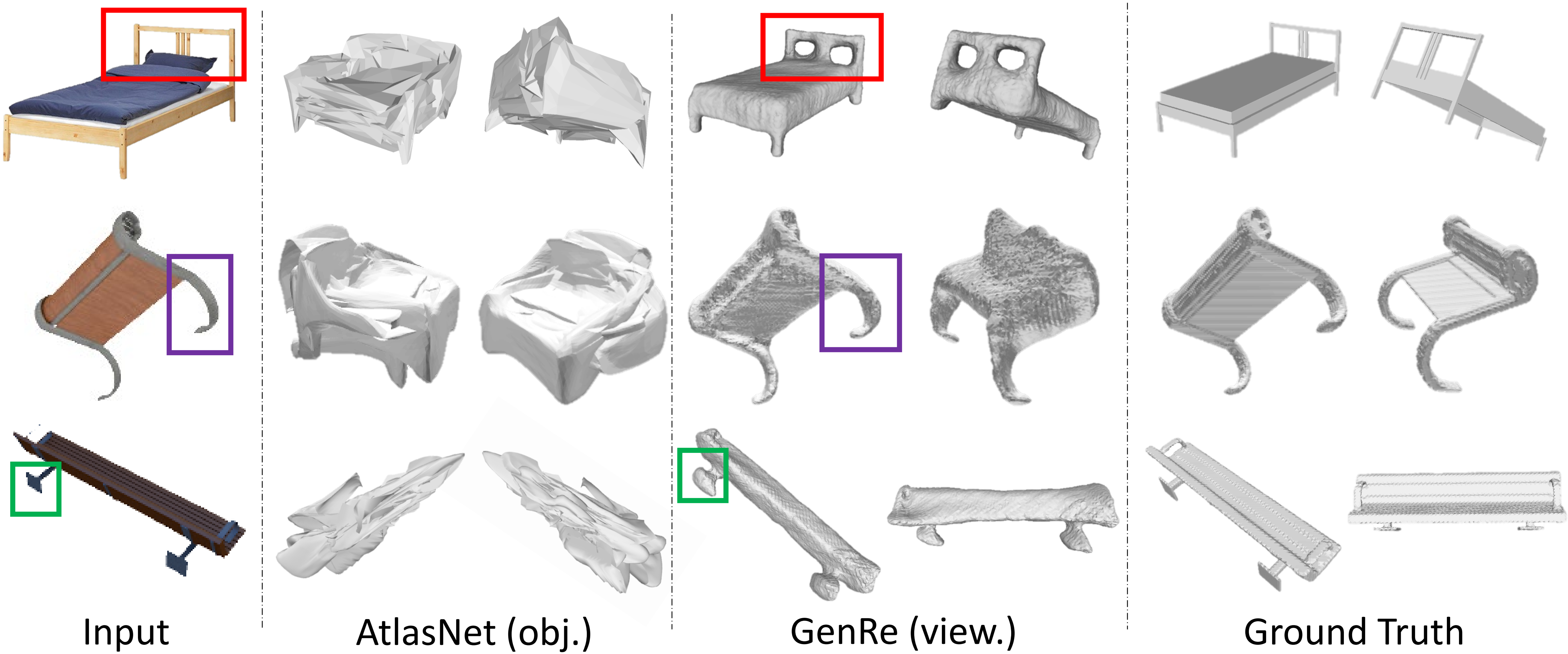
Results: Generalizing to Unseen Classes



Training: cars, chairs, airplanes

Testing: beds, benches

Results: Generalizing to Unseen Classes

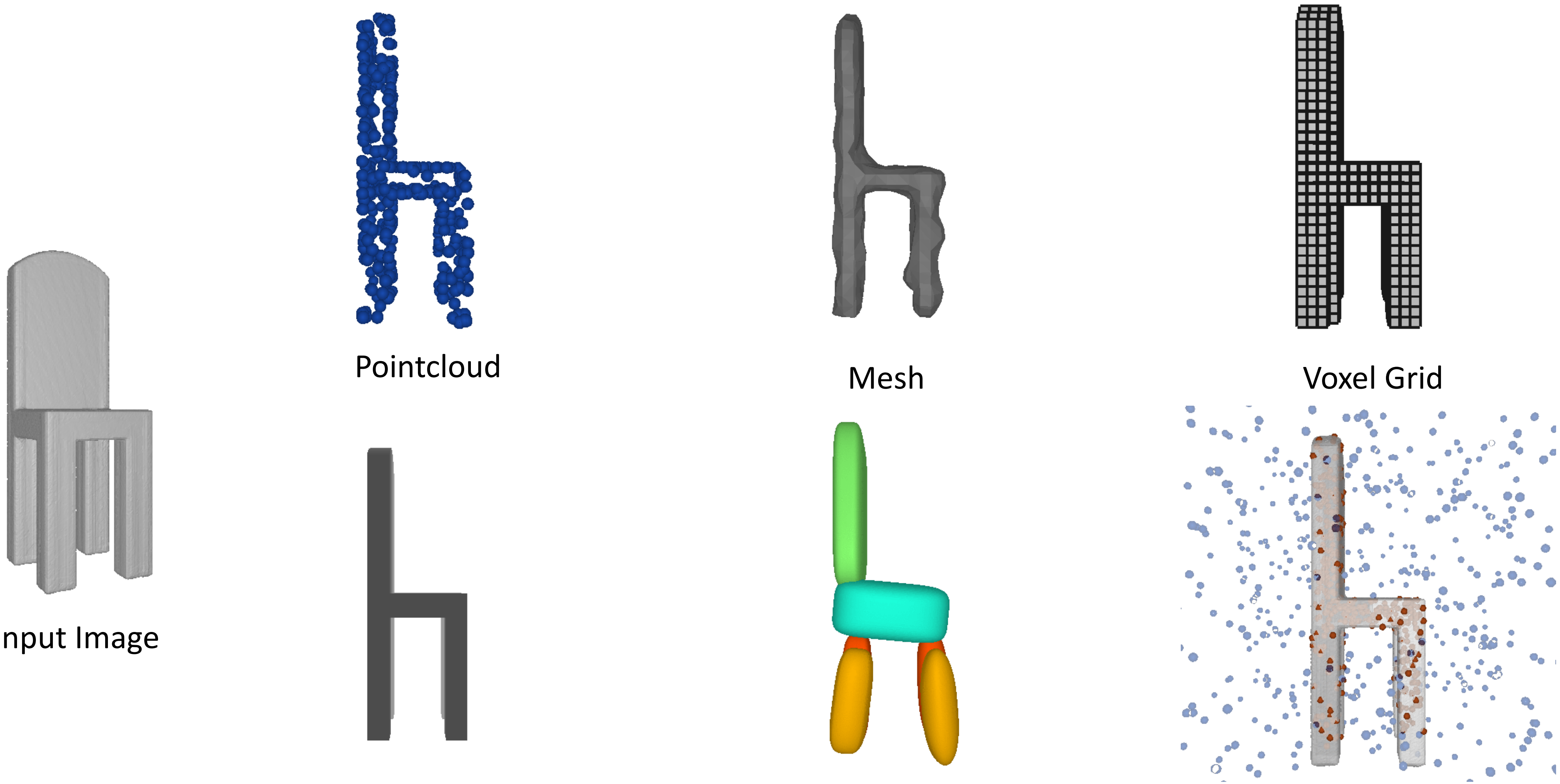


Training: cars, chairs, airplanes

Testing: beds, benches

Image 2 Shape Representation Considerations

What is the best 3D Representation?



Input Image

Pointcloud

Mesh

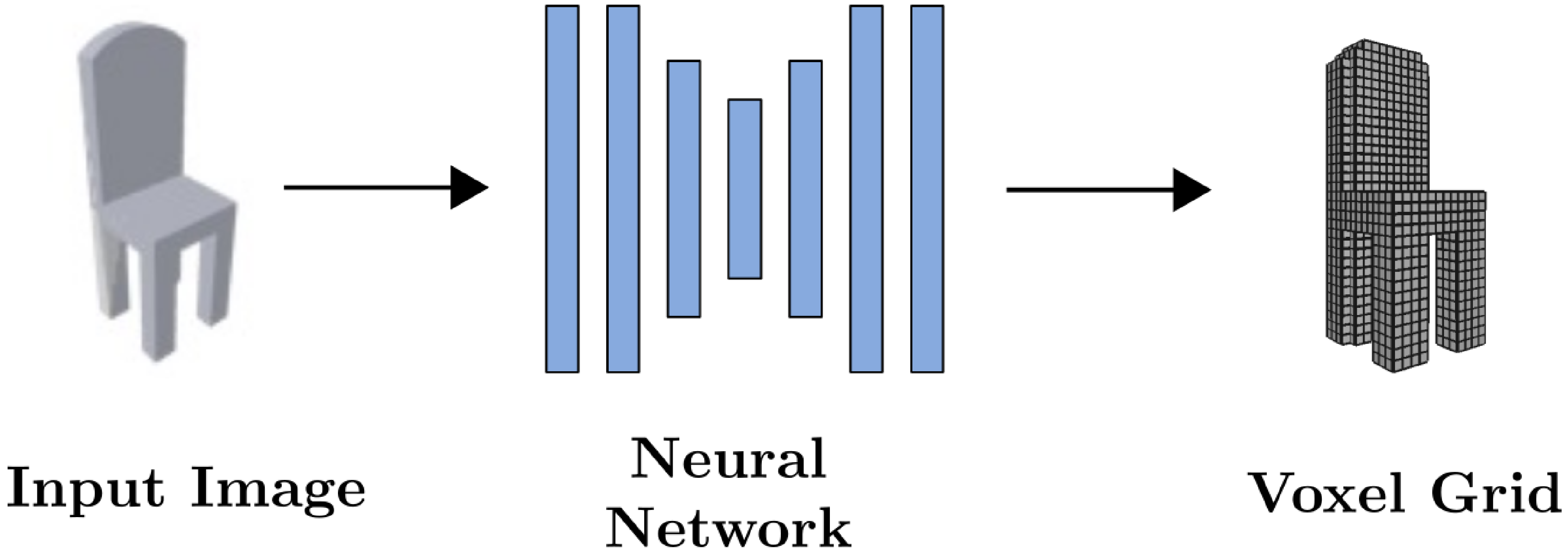
Voxel Grid

Depth

Primitives

Implicit Surface

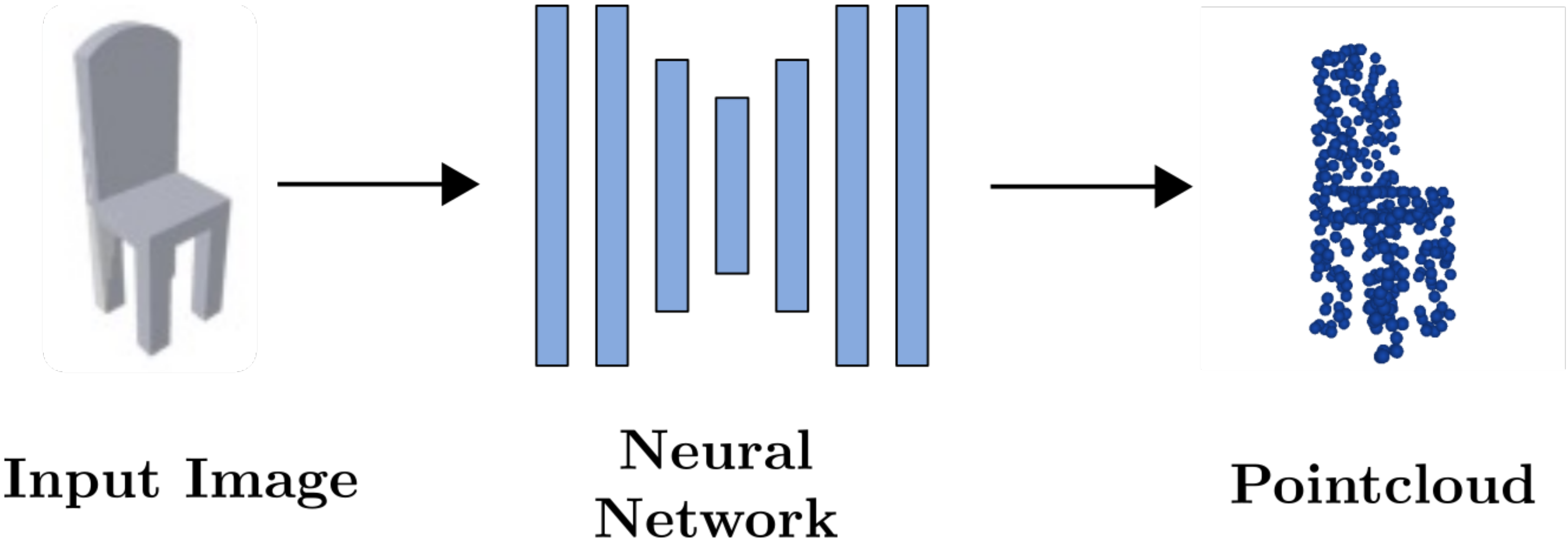
Voxel-based 3D Representations



Discretization of a 3D surface with a voxel grid:

- Can **accurately capture shape details**.
- The **parametrization size** is proportional to the r**econstruction quality**.
- **Cannot** yield **smooth reconstructions**.
- **Cannot** convey **semantic information**.

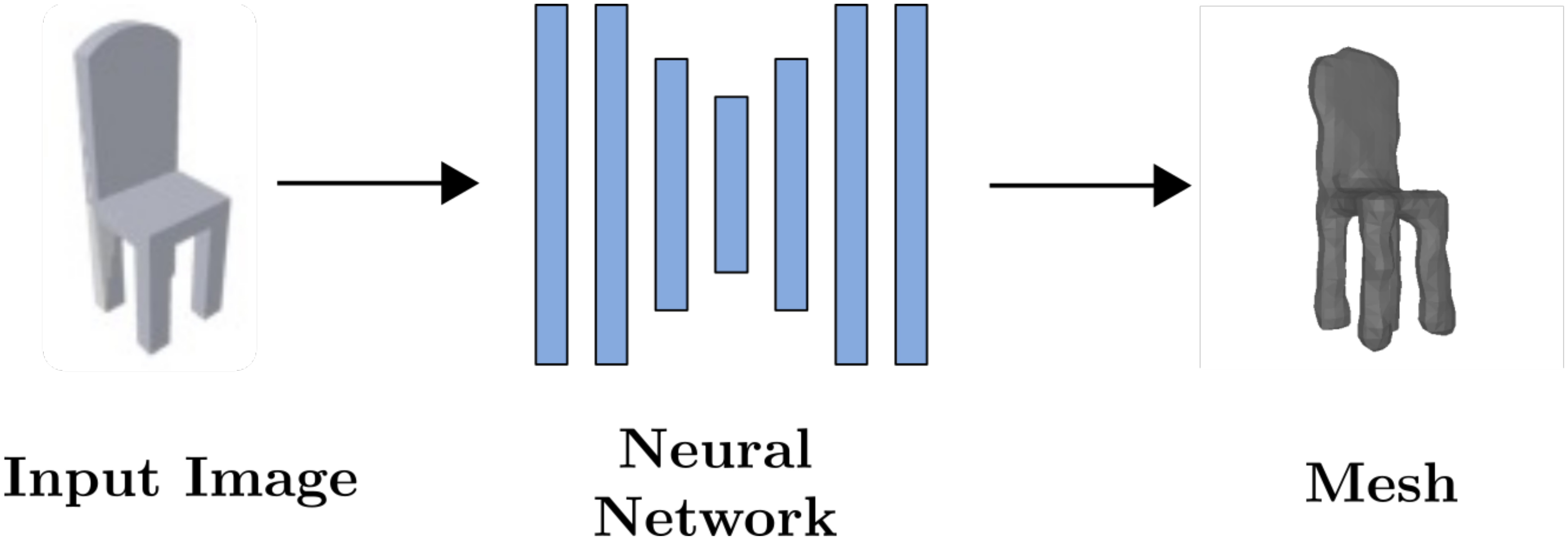
Point-based 3D Representations



Discretization of a 3D surface with 3D points:

- Can **accurately capture shape details**.
- The **parametrization size** is proportional to the r**econstruction quality**.
- **Cannot** yield **smooth reconstructions**.
- **Cannot** convey **semantic information**.
- **Lacks surface connectivity** and assumes a fixed number of points.

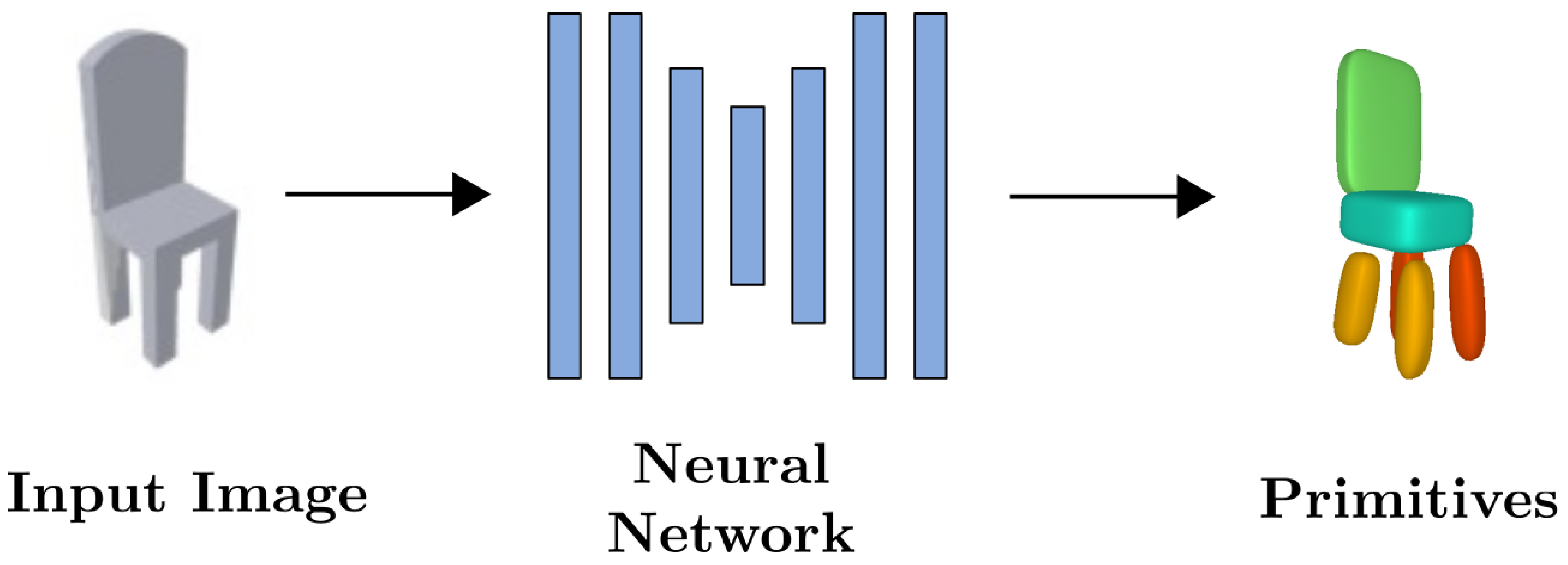
Mesh-based 3D Representations



Discretization of a 3D surface with vertices and faces:

- Can **accurately capture shape details**.
- Yields **smooth reconstructions.**
- Imposes a **large parametrization size.**
- Typically requires **class-specific template topology**.
- **Cannot** convey **semantic information**.

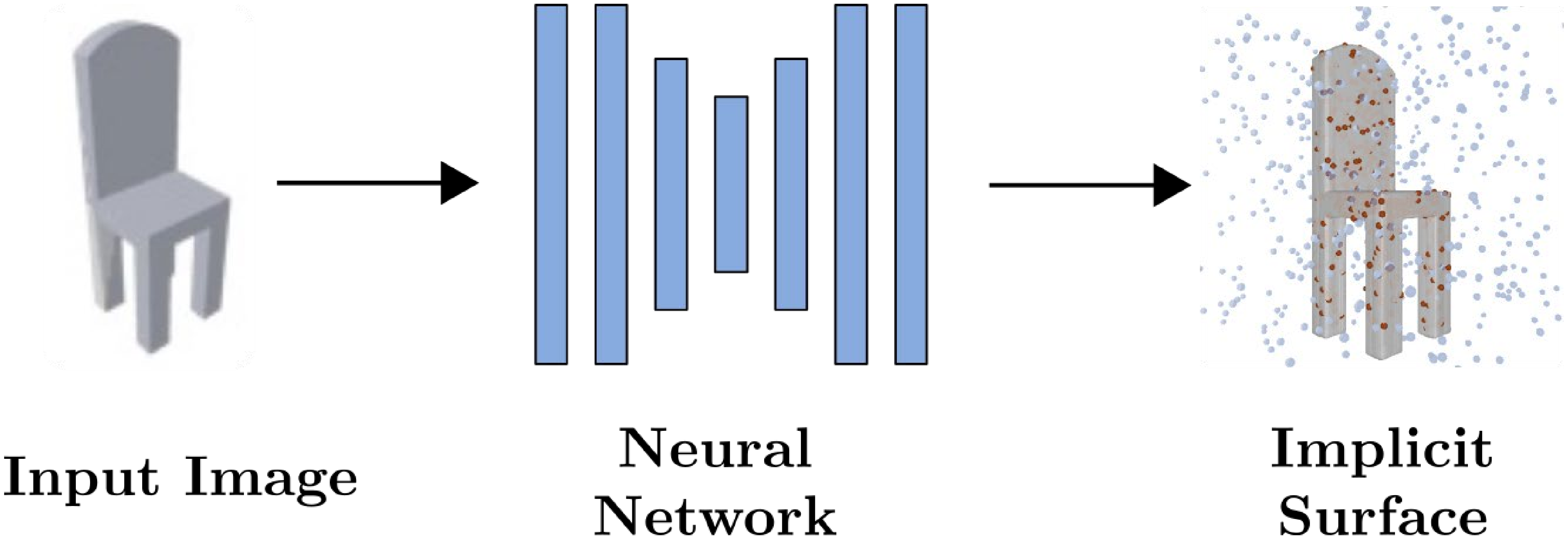
Primitive-based 3D Representations



Discretization of a 3D surface with parts:

- Can **accurately capture shape details**.
- Yields **smooth reconstructions.**
- Imposes a **small parametrization size.**
- Requires post-processing**.**
- Typically **fails to reconstruct fine shape details**.

Implicit-based 3D Representations



No Discretization:

- Can **convey semantic information**.
- Yields **smooth reconstructions.**
- Imposes a **small parametrization size.**
- Ensures **inter-object coherence.**
- **Cannot** convey **semantic information**.

Occupancy Networks: Learning 3D Reconstruction in Function Space

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger

CVPR 2019

Occupancy Networks

- **Key Idea**:
  - Do not represent the 3D shape explicitly
  - Consider the surface **implicitly**, as **the decision boundary of a non-linear classifier,** parametrized by the neural network:

$$\underbrace{f_\theta : \mathbb{R}^3}_{\text{3D Location}} \times \underbrace{\mathcal{X}}_{\text{Condition (eg, Image)}} \rightarrow \underbrace{[0,1]}_{\text{Occupancy Probability}}$$

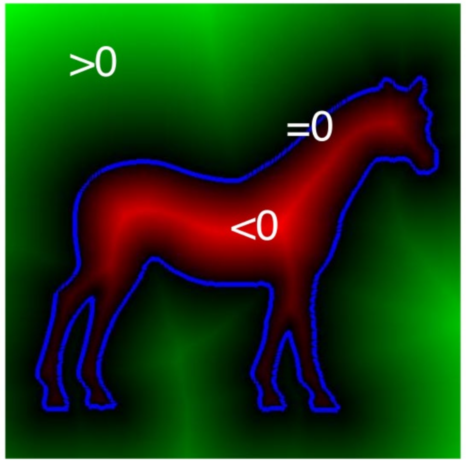
  - **Space partitioning**:
    - Outside the surface: f(x) = 0
    - Inside the surface: f(x) = 1

Occupancy Networks

- **Key Idea**:
  - Do not represent the 3D shape explicitly
  - Consider the surface **implicitly**, as **the decision boundary of a non-linear classifier,** parametrized by the neural network:

$$f_\theta : \underset{\text{3D Location}}{\mathbb{R}^3} \times \underset{\text{Condition (eg, Image)}}{\mathcal{X}} \to \underset{\text{Occupancy Probability}}{[0, 1]}$$

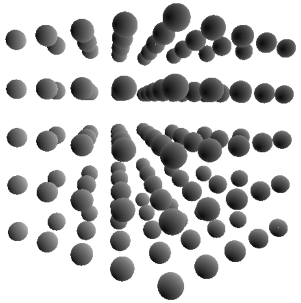
- **Space partitioning**:
  - Outside the surface: f(x) = 0
  - Inside the surface: f(x) = 1
  - Alternatively, we can use **the level set of a signed distance function**.



Occupancy Networks

- **Key Idea**:
  - Do not represent the 3D shape explicitly
  - Consider the surface implicitly, as **the decision boundary of a non-linear classifier,** parametrized by the neural network:

Condition

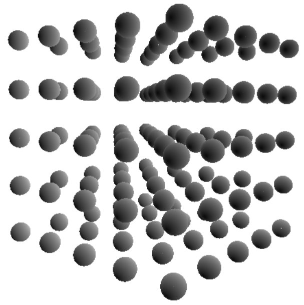


3D Locations

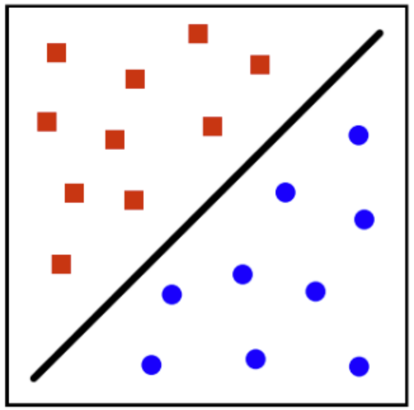
Occupancy Networks

- **Key Idea**:
  - Do not represent the 3D shape explicitly
  - Consider the surface implicitly, as **the decision boundary of a non-linear classifier,** parametrized by the neural network:

Condition



3D Locations

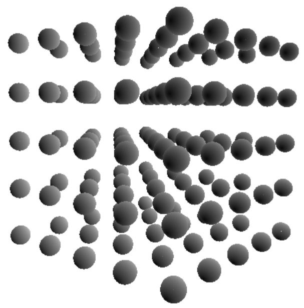


The **decision boundary of the classifier** models the **occupancy field.**

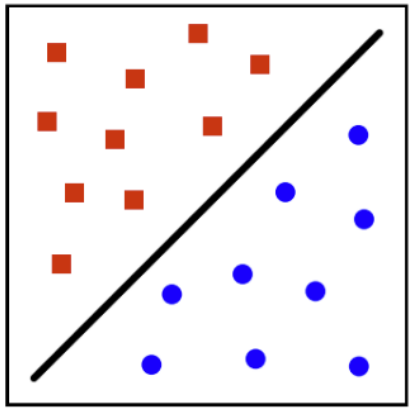
Occupancy Networks

- **Key Idea**:
  - Do not represent the 3D shape explicitly
  - Consider the surface implicitly, as **the decision boundary of a non-linear classifier,** parametrized by the neural network:

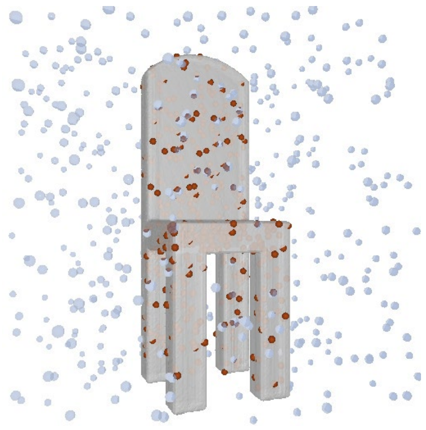
Condition



3D Locations



The **decision boundary of the classifier** models the **occupancy field.**



Occupancy Probability

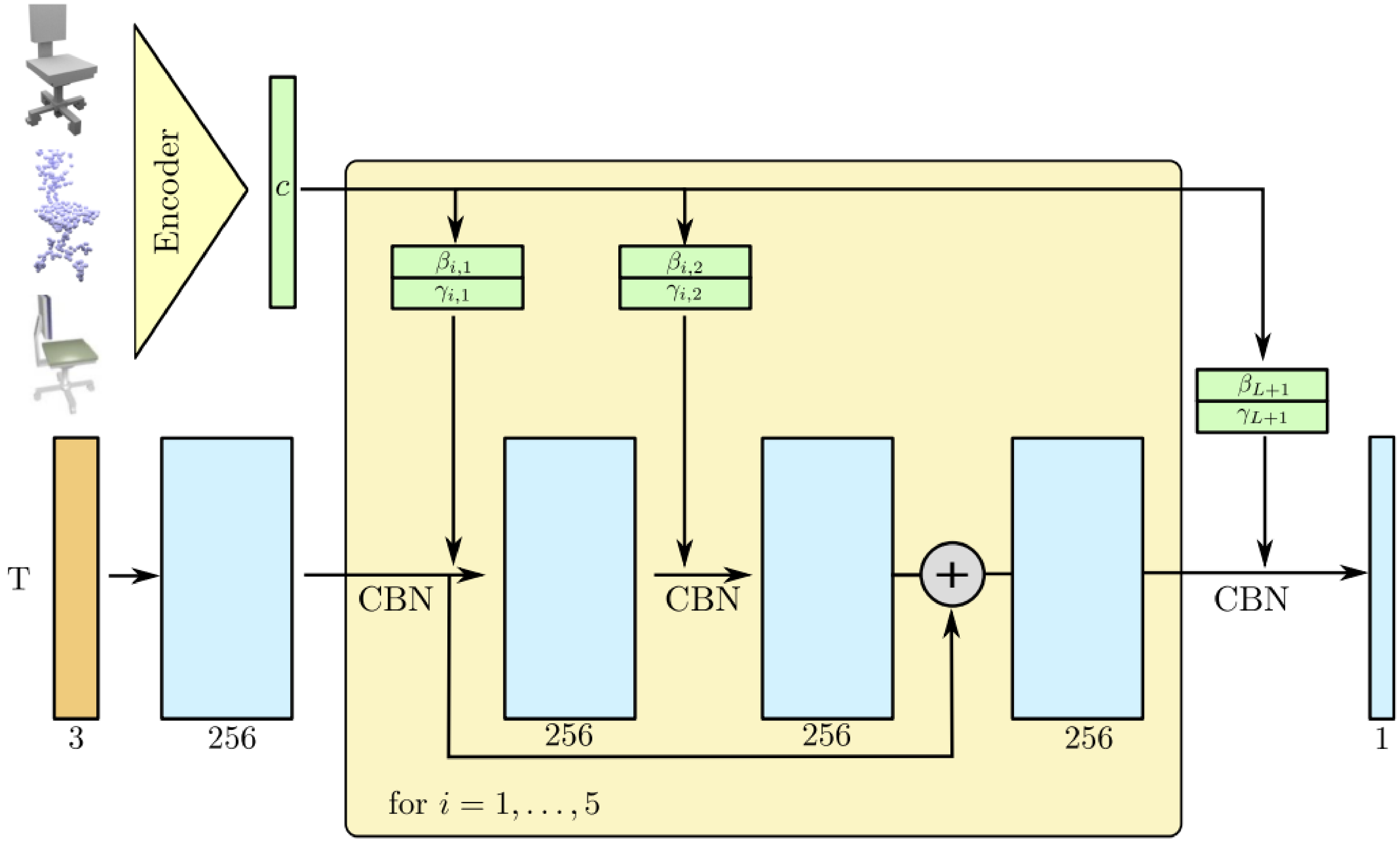
Occupancy Networks

- **Key Idea**:
  - Do not represent the 3D shape explicitly
  - Consider the surface **implicitly**, as **the decision boundary of a non-linear classifier,** parametrized by the neural network:

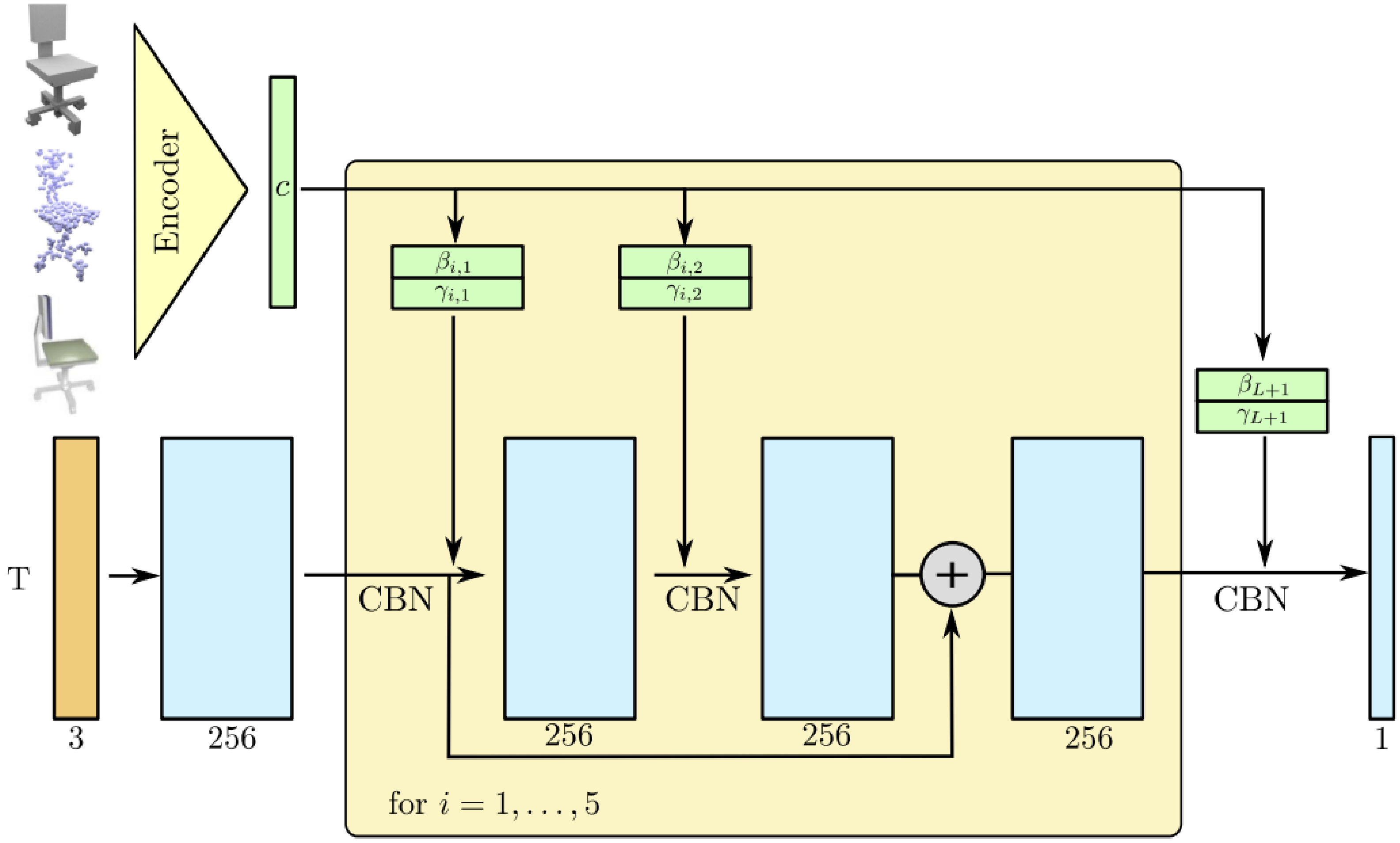
$$f_\theta : \underset{\substack{\uparrow \\ \text{3D} \\ \text{Location}}}{\mathbb{R}^3} \times \underset{\substack{\uparrow \\ \text{Condition} \\ \text{(eg, Image)}}}{\mathcal{X}} \to \underset{\substack{\uparrow \\ \text{Occupancy} \\ \text{Probability}}}{[0, 1]}$$

  - **Benefits**:
    - Can generate 3D shapes of infinite high resolutions.
    - Can capture arbitrary topologies.

Network Architecture



Network Architecture



How can we train Occupancy Networks?

Optimization Objective

Assume **supervision in the form of a watertight mesh**, parametrized as **a set of occupancy pairs**, denoting whether **a 3D point lies inside or outside the target mesh**.

$$\mathcal{L}(\theta, \psi) = \sum_{j=1}^{K} \mathrm{BCE}(f_\theta(p_{ij}, z_i), o_{ij}) + KL\left[q_\psi(z|(p_{ij}, o_{ij})_{j=1:K}) \,\|\, p_0(z)\right]$$

Optimization Objective

Assume **supervision in the form of a watertight mesh**, parametrized as **a set of occupancy pairs**, denoting whether **a 3D point lies inside or outside the target mesh**.

$$\mathcal{L}(\theta, \psi) = \sum_{j=1}^{K} \underbrace{\boxed{\text{BCE}(f_\theta(p_{ij}, z_i), o_{ij})}}_{\text{Binary Cross Entropy}} + KL\left[q_\psi(z | (p_{ij}, o_{ij})_{j=1:K}) \,\|\, p_0(z)\right]$$

Optimization Objective

Assume **supervision in the form of a watertight mesh**, parametrized as **a set of occupancy pairs**, denoting whether **a 3D point lies inside or outside the target mesh**.

Binary Cross Entropy

$$\mathcal{L}(\theta, \psi) = \boxed{\sum_{j=1}^{K}} \boxed{\mathrm{BCE}(f_\theta(p_{ij}, z_i), o_{ij})} + KL\left[q_\psi(z | (p_{ij}, o_{ij})_{j=1:K}) \,\|\, p_0(z)\right]$$

Sample K random points Within the bounding box that contains the target object.

Optimization Objective

Assume **supervision in the form of a watertight mesh**, parametrized as **a set of occupancy pairs**, denoting whether **a 3D point lies inside or outside the target mesh**.

Binary Cross Entropy

$$\mathcal{L}(\theta, \psi) = \sum_{j=1}^{K} \mathrm{BCE}(f_\theta(p_{ij}, z_i), o_{ij}) + KL\left[q_\psi(z | (p_{ij}, o_{ij})_{j=1:K}) \,\|\, p_0(z)\right]$$

Sample K random points Within the bounding box that contains the target object.

Supervision

Optimization Objective

Assume **supervision in the form of a watertight mesh**, parametrized as **a set of occupancy pairs**, denoting whether **a 3D point lies inside or outside the target mesh**.

Binary Cross Entropy

Encoder

$$\mathcal{L}(\theta, \psi) = \boxed{\sum_{j=1}^{K}} \boxed{\mathrm{BCE}(f_\theta(p_{ij}, z_i), \boxed{o_{ij}})} + KL\left[\boxed{q_\psi(z|(p_{ij}, o_{ij})_{j=1:K})} \| p_0(z)\right]$$

Sample K random points Within the bounding box that contains the target object.

Supervision

Optimization Objective

Assume **supervision in the form of a watertight mesh**, parametrized as **a set of occupancy pairs**, denoting whether **a 3D point lies inside or outside the target mesh**.

$$\mathcal{L}(\theta,\psi) = \underbrace{\sum_{j=1}^{K}}_{\text{Sample K random points}} \overbrace{\mathrm{BCE}(f_\theta(p_{ij}, z_i), \underbrace{o_{ij}}_{\text{Supervision}})}^{\text{Binary Cross Entropy}} + KL\left[\overbrace{q_\psi(z|(p_{ij}, o_{ij})_{j=1:K})}^{\text{Encoder}} \,\|\, \underbrace{p_0(z)}_{\text{Normal Distribution}}\right]$$

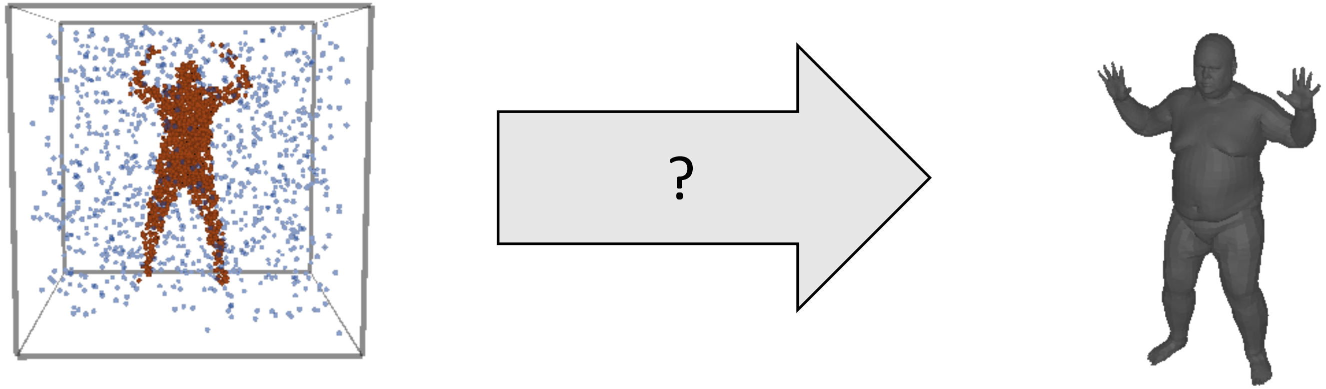
Sample K random points Within the bounding box that contains the target object.

Supervision

Normal Distribution

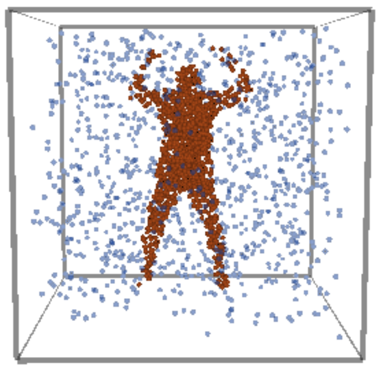
Can we convert the implicit surface to mesh?

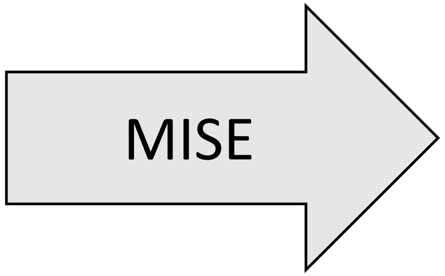
We need to extract the isosurface corresponding to the predicted implicit surface:



Can we convert the implicit surface to mesh?

We need to extract the isosurface corresponding to the predicted implicit surface:

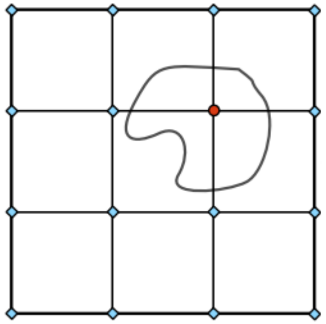




Multiresolution IsoSurface Extraction (MISE): Iteratively build an octree by incrementally quering the occupancy network.

Can we convert the implicit surface to mesh?

Multiresolution IsoSurface Extraction (MISE): **Iteratively build an octree** by incrementally quering the occupancy network.

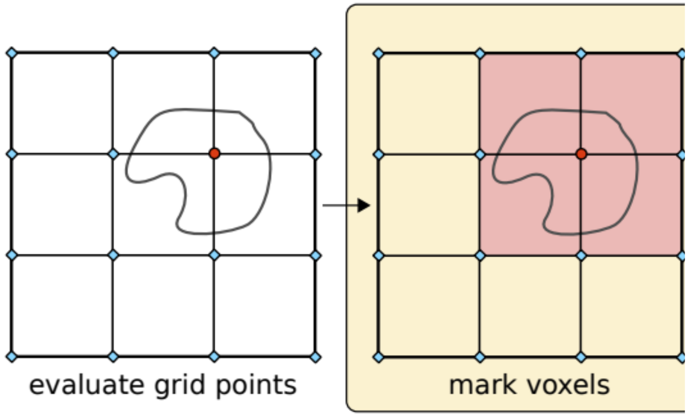


evaluate grid points

- In a grid of points and find the points that are **occupied** and points that are **unoccupied**.

Can we convert the implicit surface to mesh?

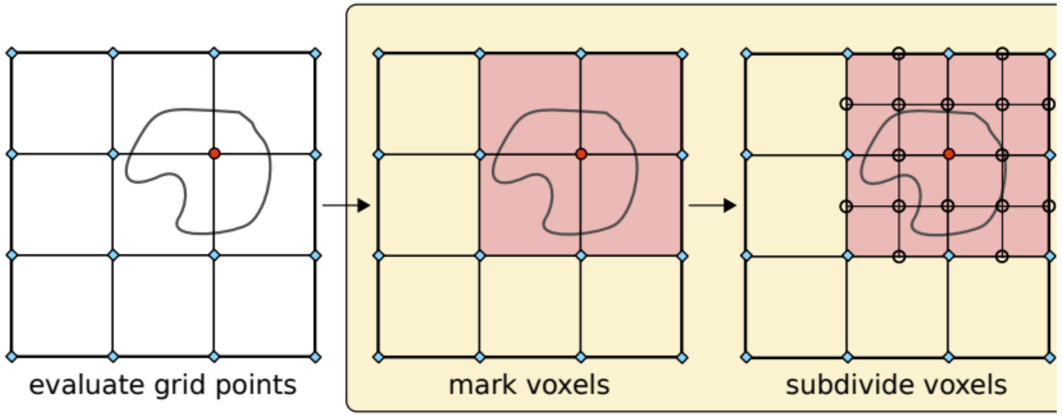
Multiresolution IsoSurface Extraction (MISE): **Iteratively build an octree** by incrementally quering the occupancy network.



- In a grid of points and find the points that are **occupied** and points that are **unoccupied**.
- **Mark the voxels between occupied and unoccupied** points as **voxels that require further investigation.**

Can we convert the implicit surface to mesh?

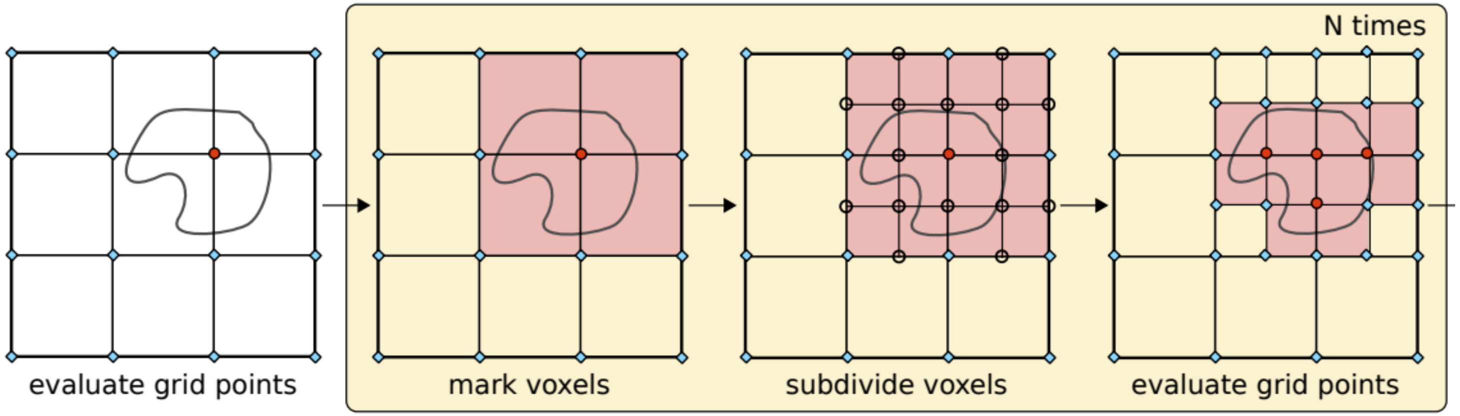
Multiresolution IsoSurface Extraction (MISE): **Iteratively build an octree** by incrementally quering the occupancy network.



- In a grid of points and find the points that are **occupied** and points that are **unoccupied**.
- **Mark the voxels between occupied and unoccupied** points as **voxels that require further investigation.**
- **Further subdivide these voxels.**

Can we convert the implicit surface to mesh?

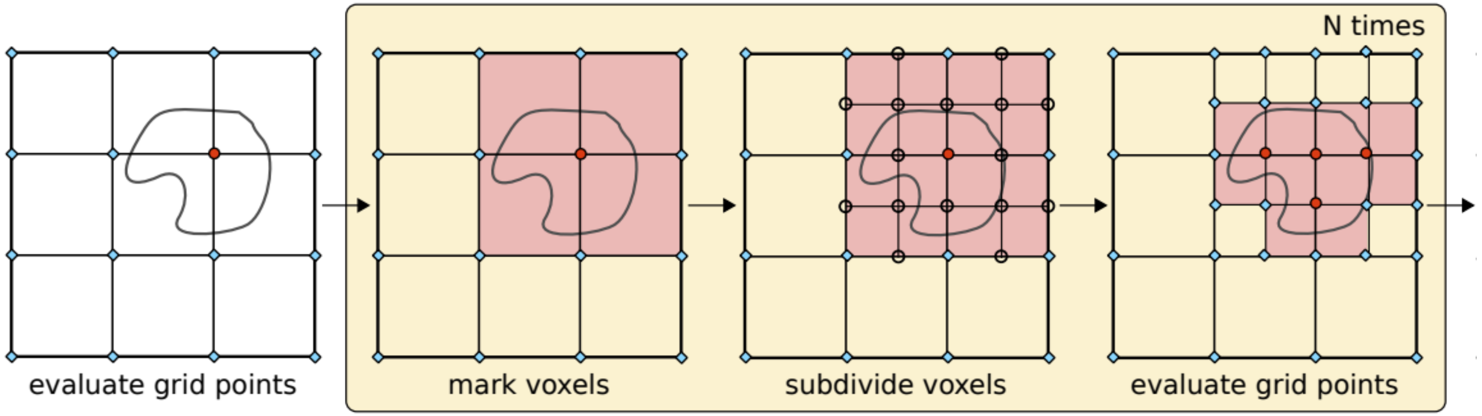
Multiresolution IsoSurface Extraction (MISE): **Iteratively build an octree** by incrementally quering the occupancy network.



- In a grid of points and find the points that are **occupied** and points that are **unoccupied**.
- **Mark the voxels between occupied and unoccupied** points as **voxels that require further investigation.**
- **Further subdivide these voxels.**
- **Query these voxels again** and find the **occupied** and **unoccupied** points.

Can we convert the implicit surface to mesh?

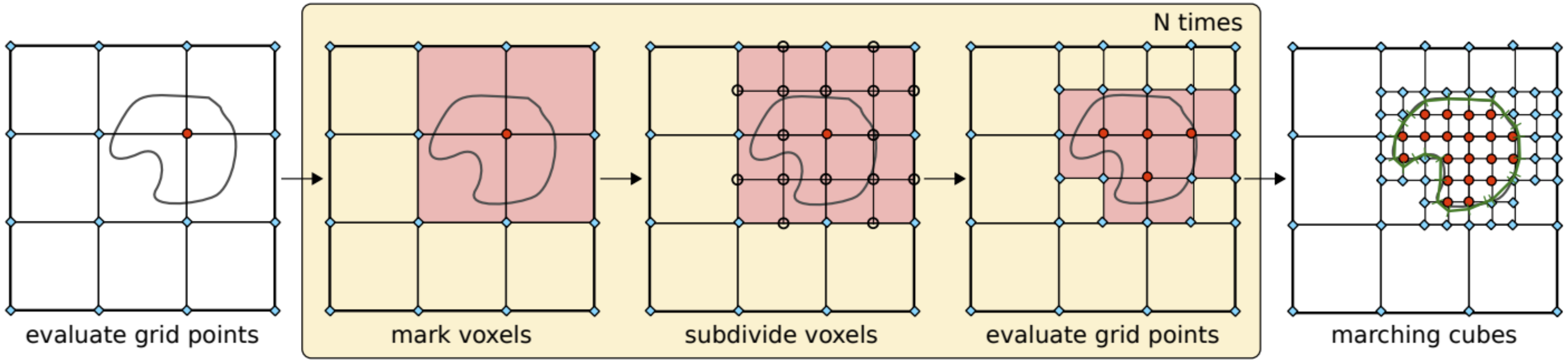
To **extract an isosurface** corresponding to a new observation given a trained occupancy network, use **Multiresolution IsoSurface Extraction (MISE)**:



- Repeat this process N times until we reach the desired resolution.

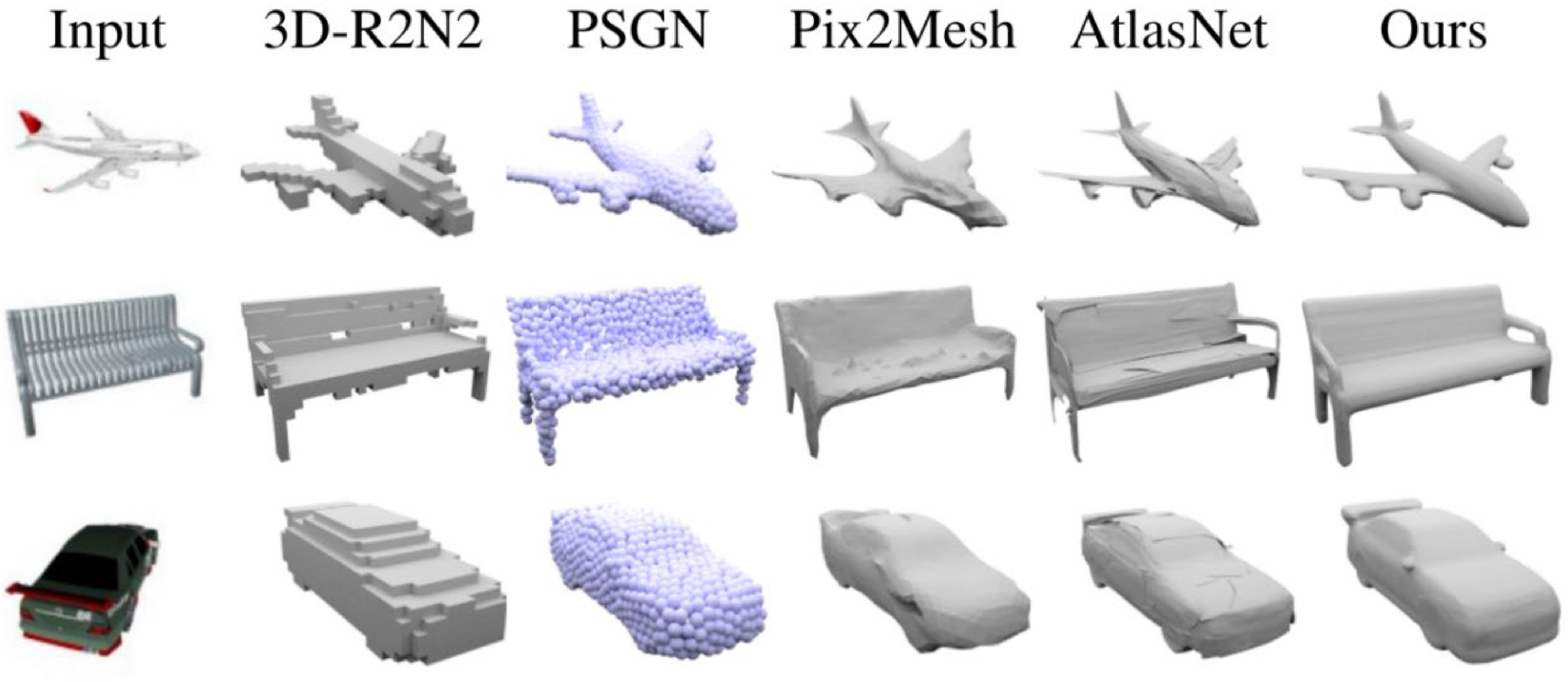
Can we convert the implicit surface to mesh?

To **extract an isosurface** corresponding to a new observation given a trained occupancy network, use **Multiresolution IsoSurface Extraction (MISE)**:



- Repeat this process N times until we reach the desired resolution.
- Extract triangular mesh using **Marching Cubes**.

How well does it work?



What about Generative Models?

Car

Sofa

Chair

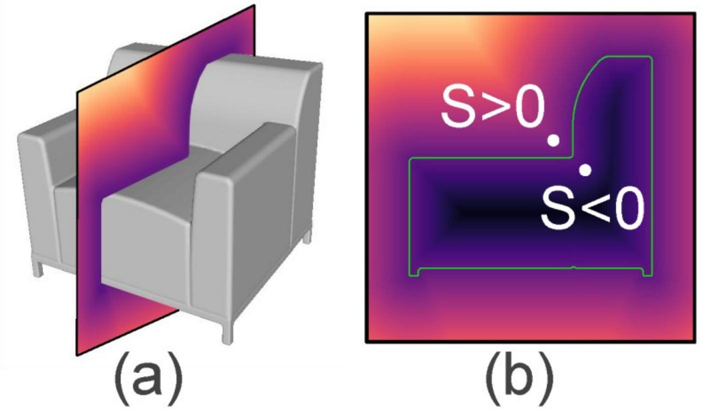
DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction

Weiyue Wang*, Qiangeng Xu*, Duygu Ceylan, Radomir Mech, Ulrich Neumann

NeurIPS 2019

Implicit 3D surfaces for recovering fine details

- Deep Implicit Surface Network (DISN) DISN predicts Signed Distance Function (SDF) for each 3D point. SDFs **do not impose any constraints on the output topology and resolution.**

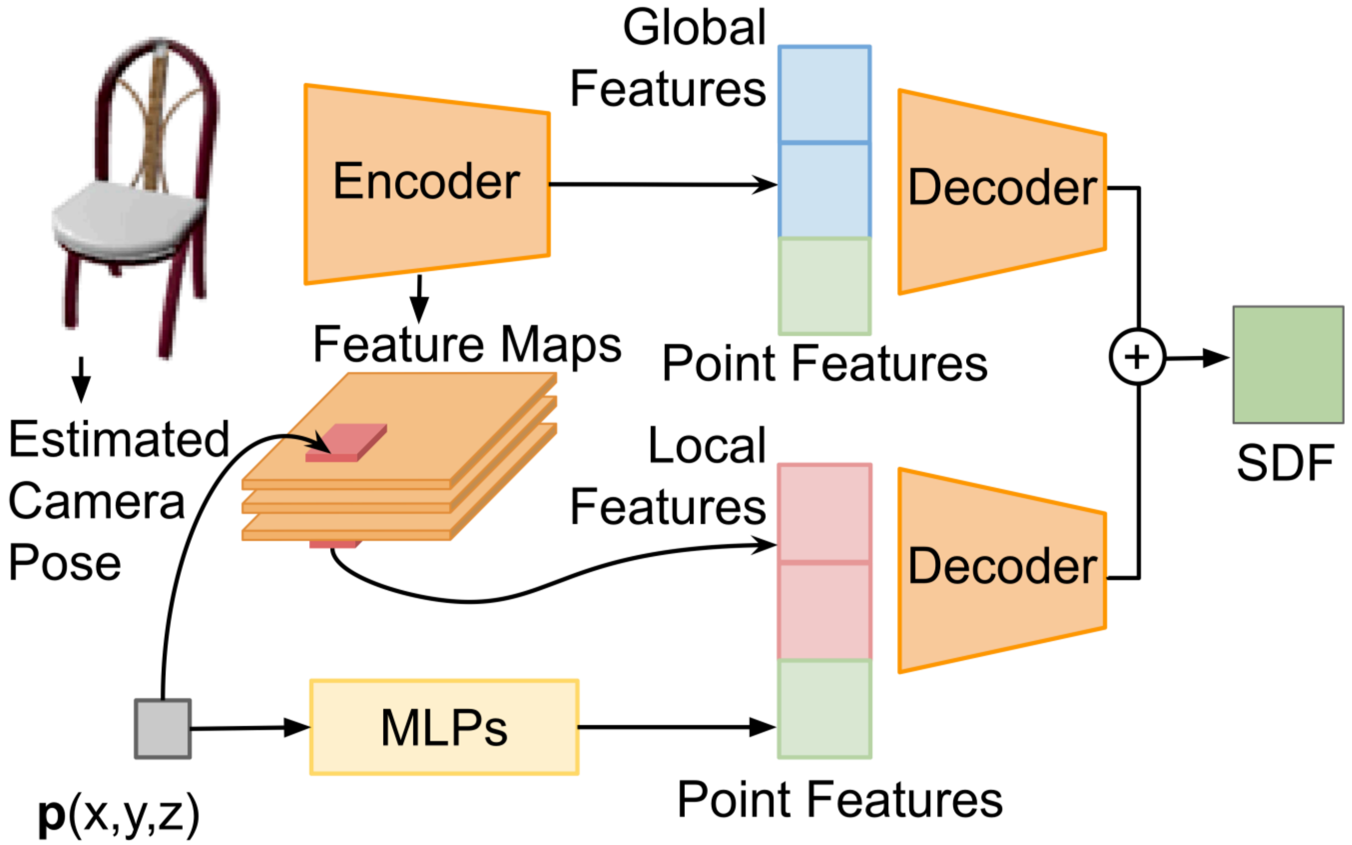


SDF is a continuous function that maps a given 3D point $\mathbf{p} = (x, y, z) \in \mathbb{R}^3$ to a real value $s \in \mathbb{R} : s = SDF(\mathbf{p})$.

An isosurface $S_0 = \{p | SDF(p) = 0\}$ implicitly represents the underlying 3D shape.

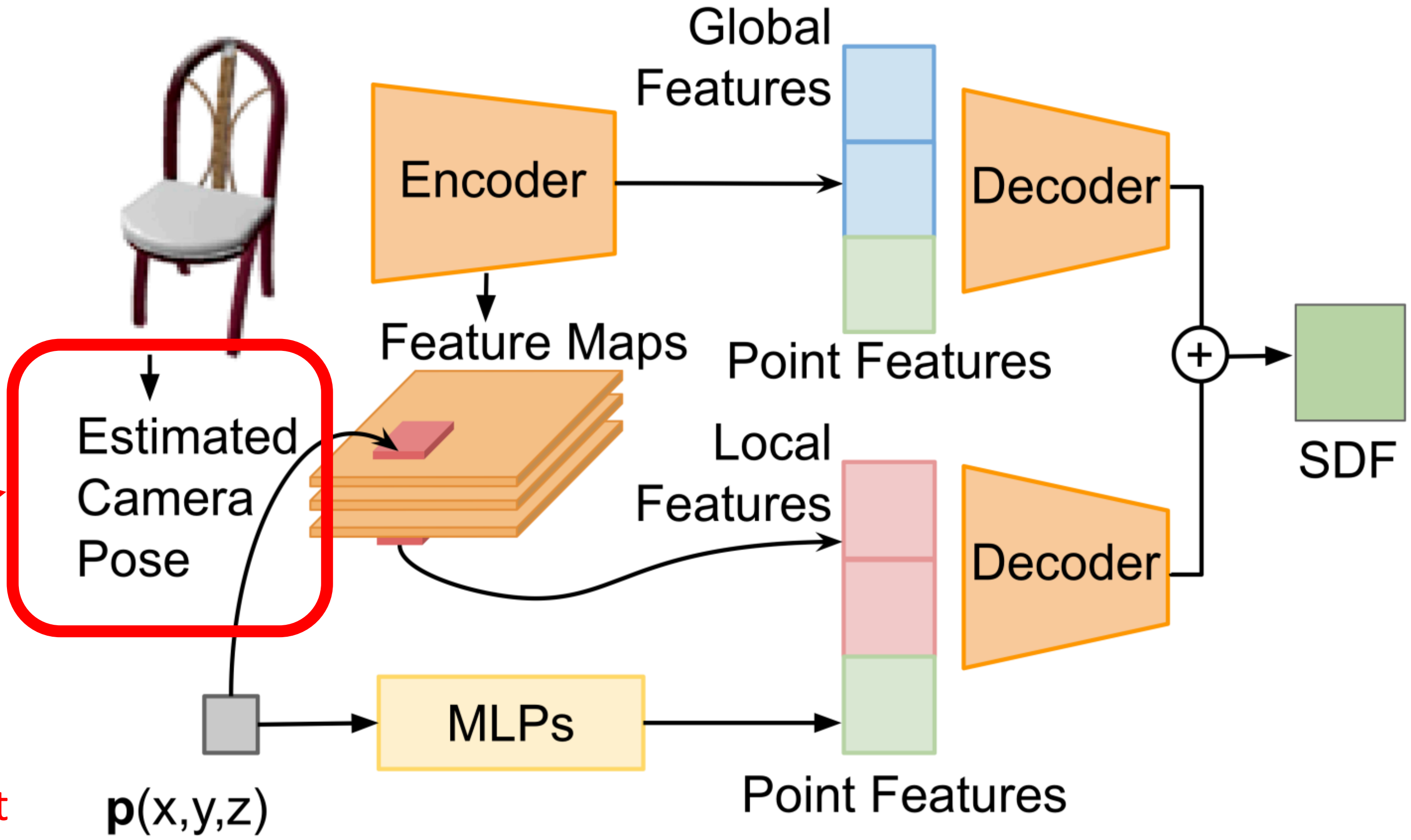
DISN – Model Overview

Key Idea: Use both global and local features for capturing both the overall shape and the fine-grained details.



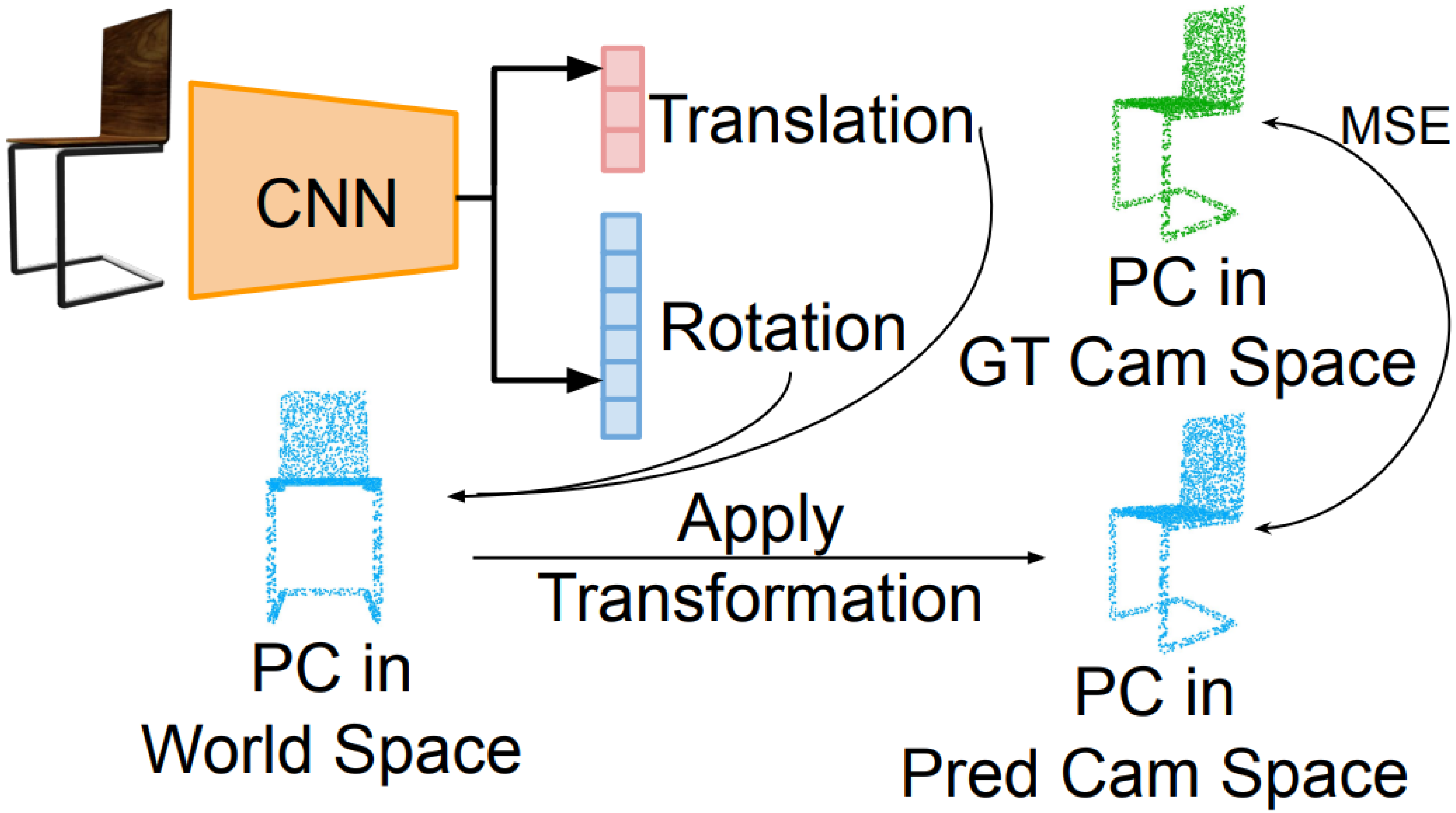
DISN – Model Overview

Key Idea: Use both global and local features for capturing both the overall shape and the fine-grained details.



Given an image and a **3D point p in world coordinates**, we first need to **predict the camera parameters** that project the 3D point to the image plane.

DISN – Camera Pose Estimation



DISN – Camera Pose Estimation

- **Loss:** MSE between the transformed point cloud and the ground truth point cloud **in the camera space**.

$$L_{cam} = \frac{\sum_{\mathbf{p}_w \in PC_w} ||\mathbf{p}_G - (\mathbf{R}\mathbf{p}_w + \mathbf{t}))||_2^2}{\sum_{\mathbf{p}_w \in PC_w} 1}$$

DISN – Camera Pose Estimation

- **Loss:** MSE between the transformed point cloud and the ground truth point cloud **in the camera space**.

Ground truth pointcloud location in camera space.

$$L_{cam} = \frac{\sum_{\mathbf{p}_w \in PC_w} \|\boxed{\mathbf{p}_G} - (\mathbf{R}\mathbf{p}_w + \mathbf{t}))\|_2^2}{\sum_{\mathbf{p}_w \in PC_w} 1}$$

DISN – Camera Pose Estimation

- **Loss:** MSE between the transformed point cloud and the ground truth point cloud **in the camera space**.

Ground truth pointcloud location in camera space.

$$L_{cam} = \frac{\sum_{\mathbf{p}_w \in PC_w} \|\boxed{\mathbf{p}_G} - (\mathbf{R}\boxed{\mathbf{p}_w} + \mathbf{t})\|_2^2}{\sum_{\mathbf{p}_w \in PC_w} 1}$$

Point in world space coordinates.

DISN – Camera Pose Estimation

- **Loss:** MSE between the transformed point cloud and the ground truth point cloud **in the camera space**.

Ground truth pointcloud location in camera space.

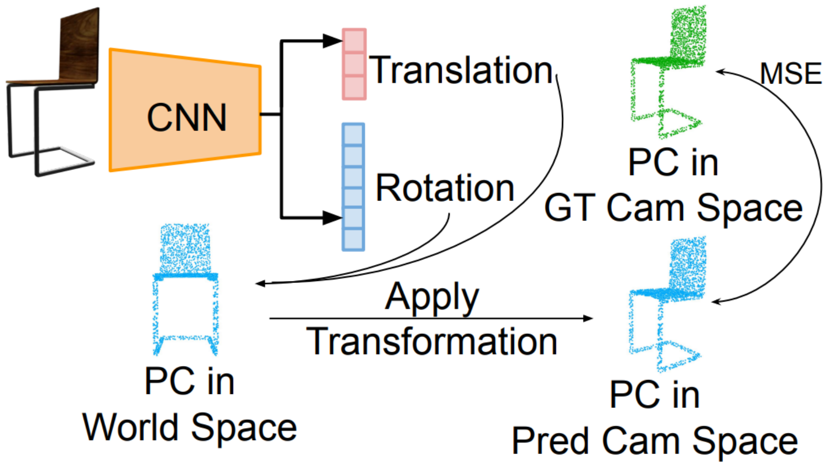
$$L_{cam} = \frac{\sum_{\mathbf{p}_w \in PC_w} \|\boxed{\mathbf{p}_G} - (\mathbf{R}\boxed{\mathbf{p}_w} + \mathbf{t}))\|_2^2}{\sum_{\mathbf{p}_w \in PC_w} 1}$$

Point in world space coordinates.

- The **rotation matrix R** and the **translation vector t** are directly predicted from the network.

DISN – Model Overview

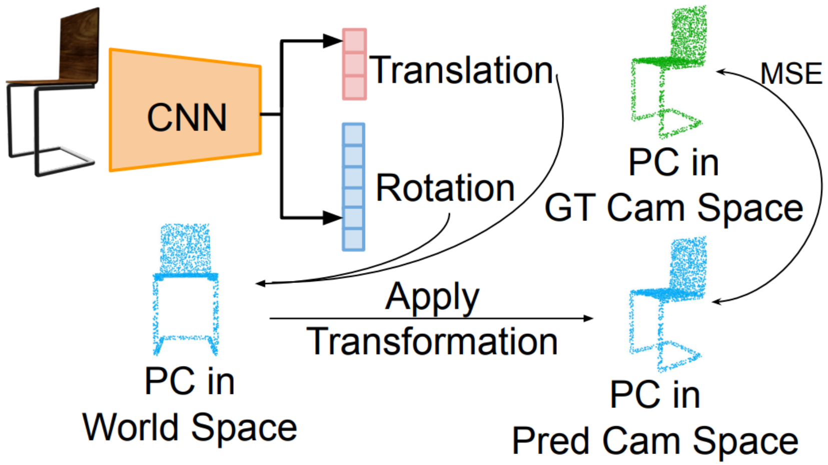
- **Camera Pose Network:** Estimate the camera pose, the 6 DoF transformation from the camera coordinate to world coordinate.



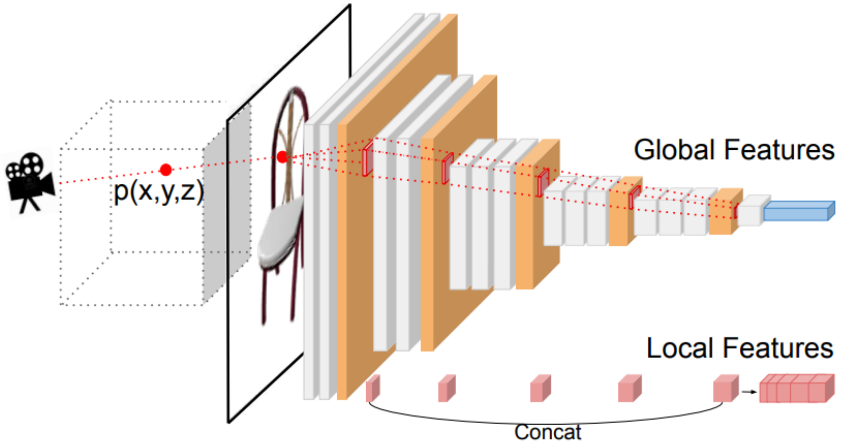
Camera Pose Network

DISN – Model Overview

- **Camera Pose Network:** Estimate the camera pose, the 6 DoF transformation from the camera coordinate to world coordinate.
- **Local Feature Extraction Network:** Using the camera pose find a 3D point's 2D location on the image and **extract local feature patches from multiple network layers.**



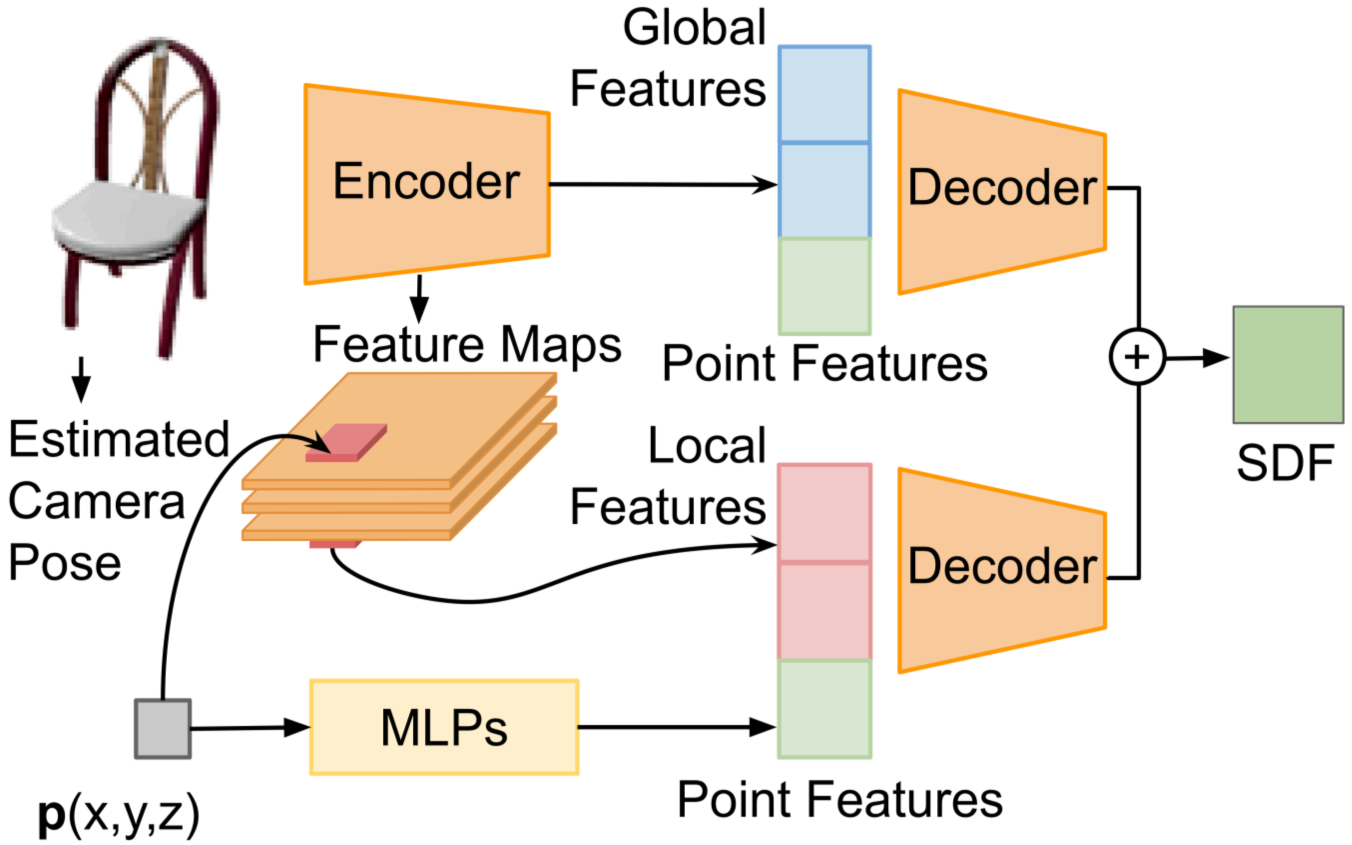
Camera Pose Network



Local Feature Extraction Network

DISN – Model Overview

Key Idea: Use both global and local features for capturing both the overall shape and the fine-grained details.

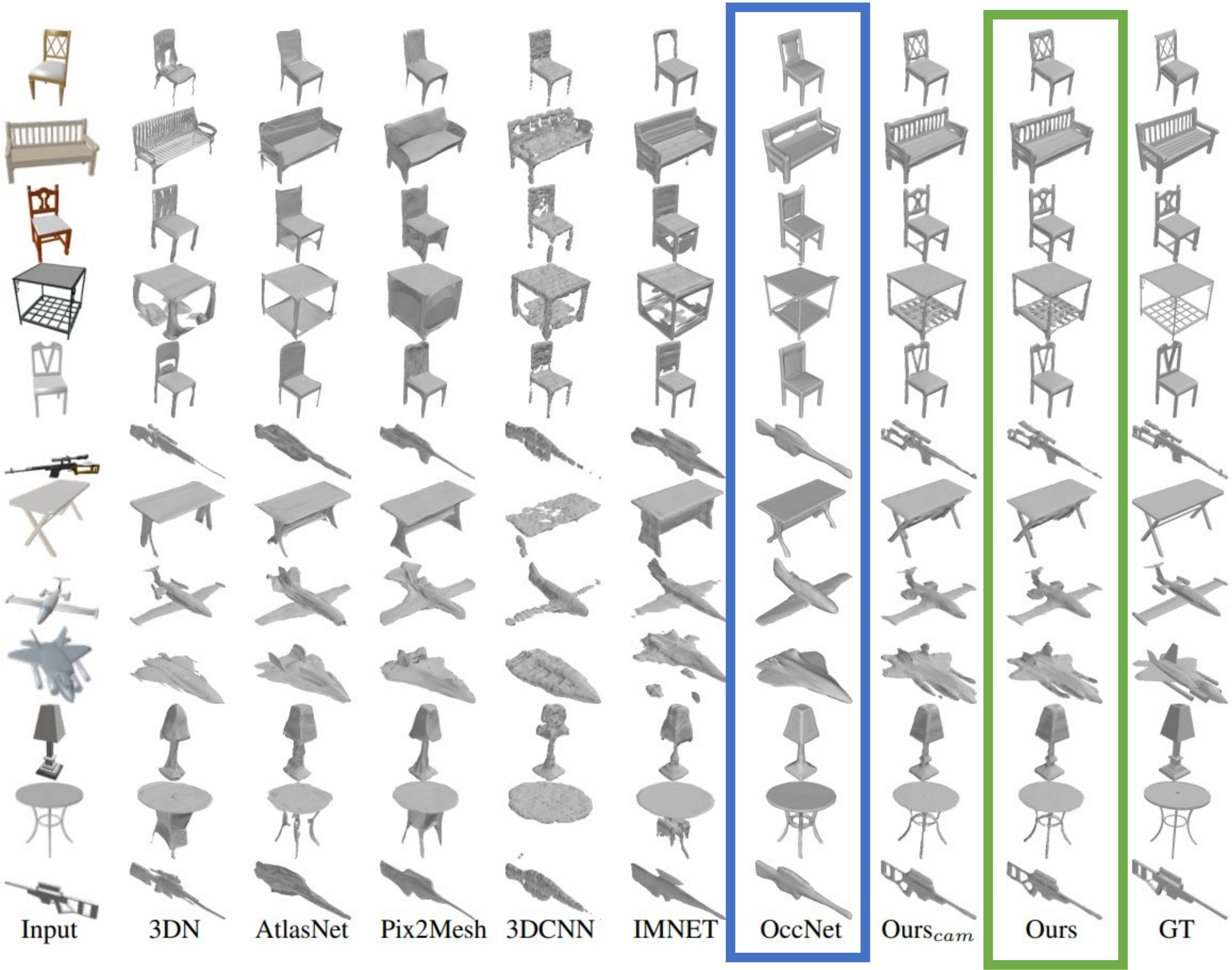


Optimization Objective

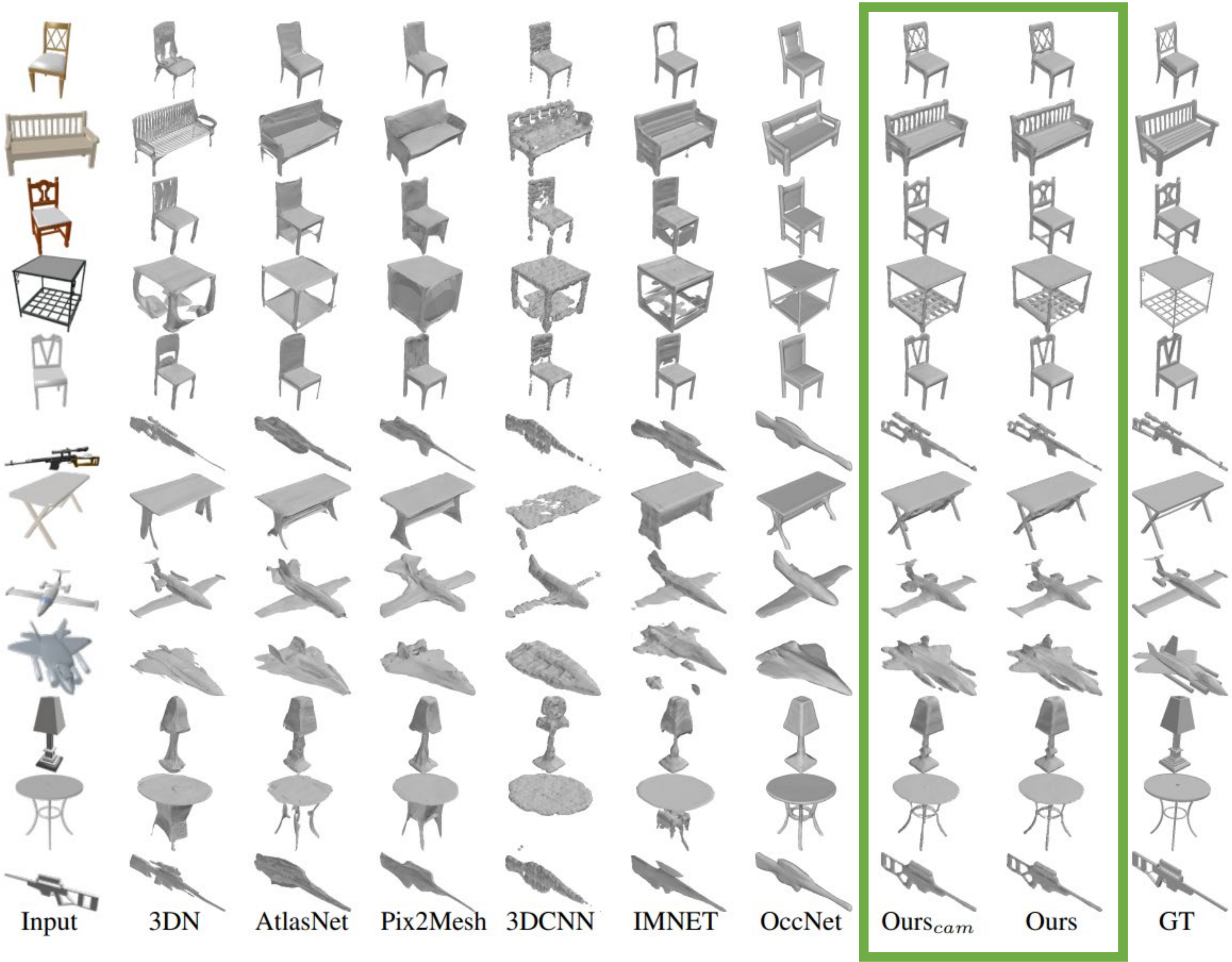
$$L_{SDF} = \sum_{\mathbf{p}} m|f(I, \mathbf{p}) - SDF^I(\mathbf{p})|,$$

$$m = \begin{cases} m_1, & \text{if } SDF^I(\mathbf{p}) < \delta, \\ m_2, & \text{otherwise,} \end{cases}$$

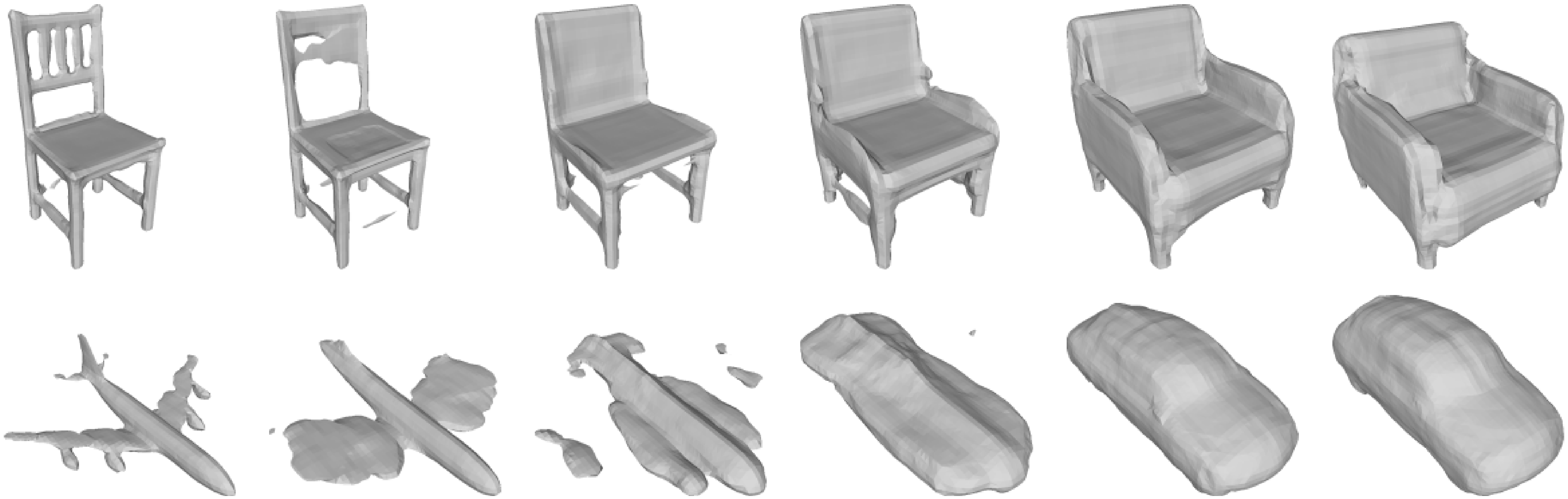
How well does it work?



How well does it work?



What about generative models?



PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization

Shunsuke Saito∗, Zeng Huang∗, Ryota Natsume∗, Shigeo Morishima,

Angjoo Kanazawa,Hao Li

ICCV 2019

PIFu: Pixel-Aligned Implicit Function

Key Idea: The **pixel-aligned implicit function** consists of a fully convolutional **image encoder** $g(\cdot)$ and a **continuous implicit function** $f(\cdot)$ represented by multi-layer perceptrons (MLPs), where the surface is defined as a level set of

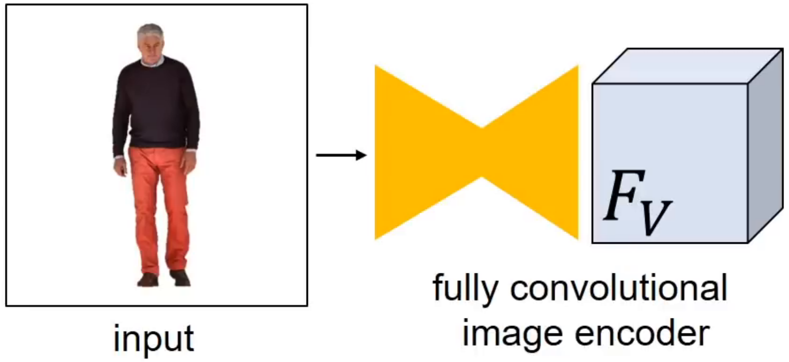
$$f(F(x), z(X)) = s : s \in \mathbb{R}$$

where for a 3D point X, $x = \pi(X)$ is its 2D projection, $z(X)$ is the **depth value in the camera coordinate space**, $F(x) = g(I(x))$ is **the image feature at x**.

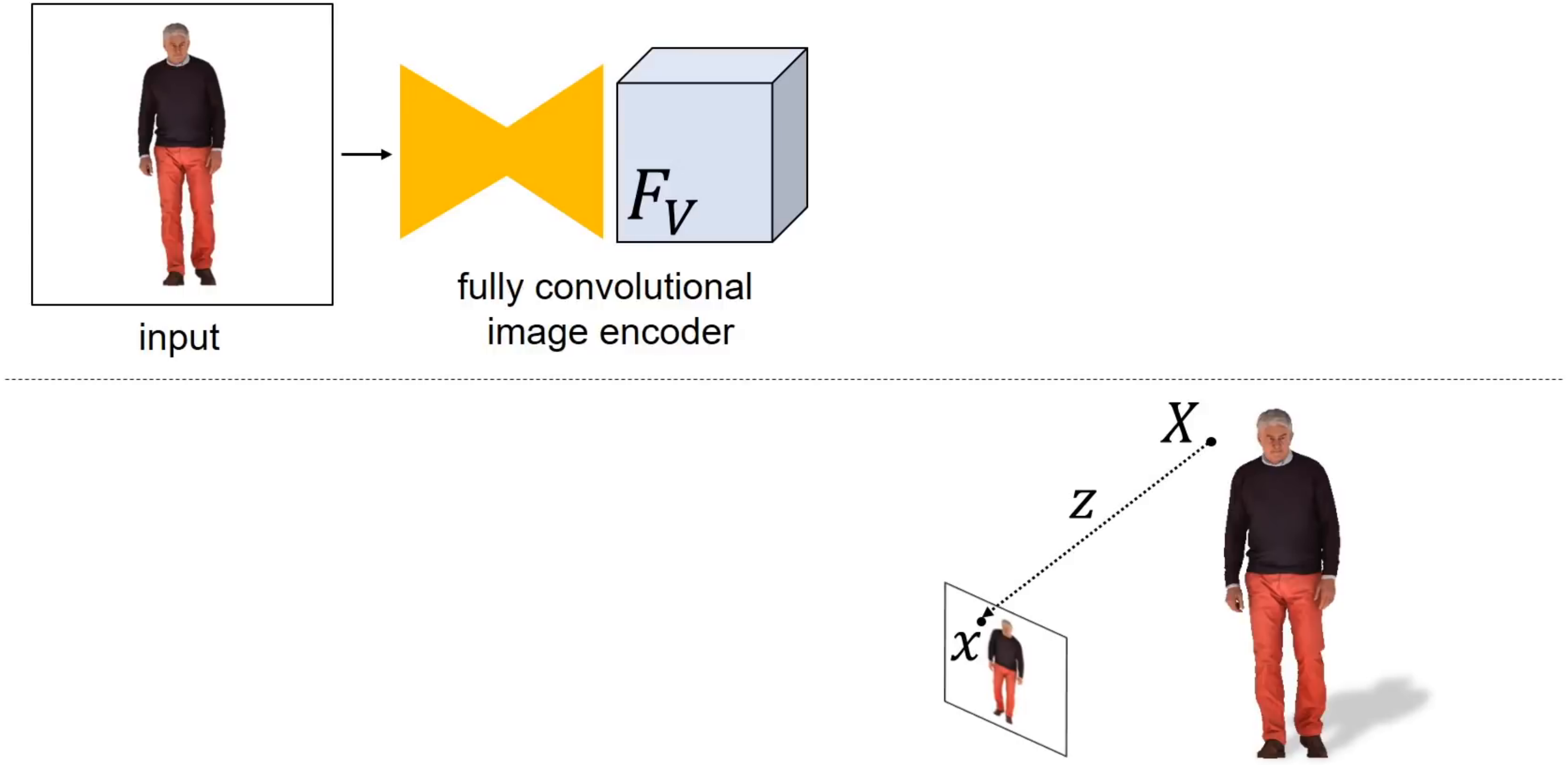
PIFu: Pixel-Aligned Implicit Function

input

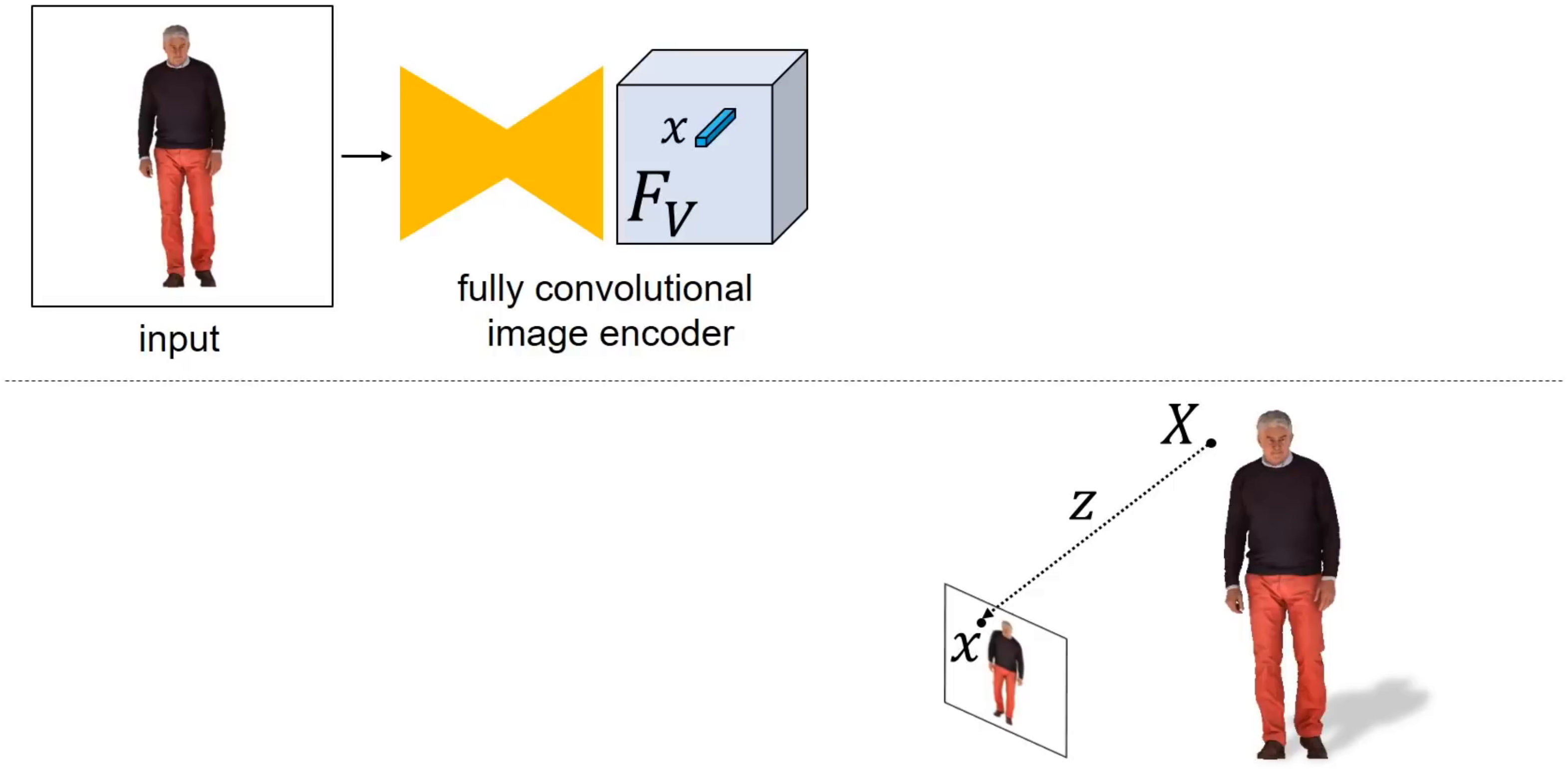
PIFu: Pixel-Aligned Implicit Function



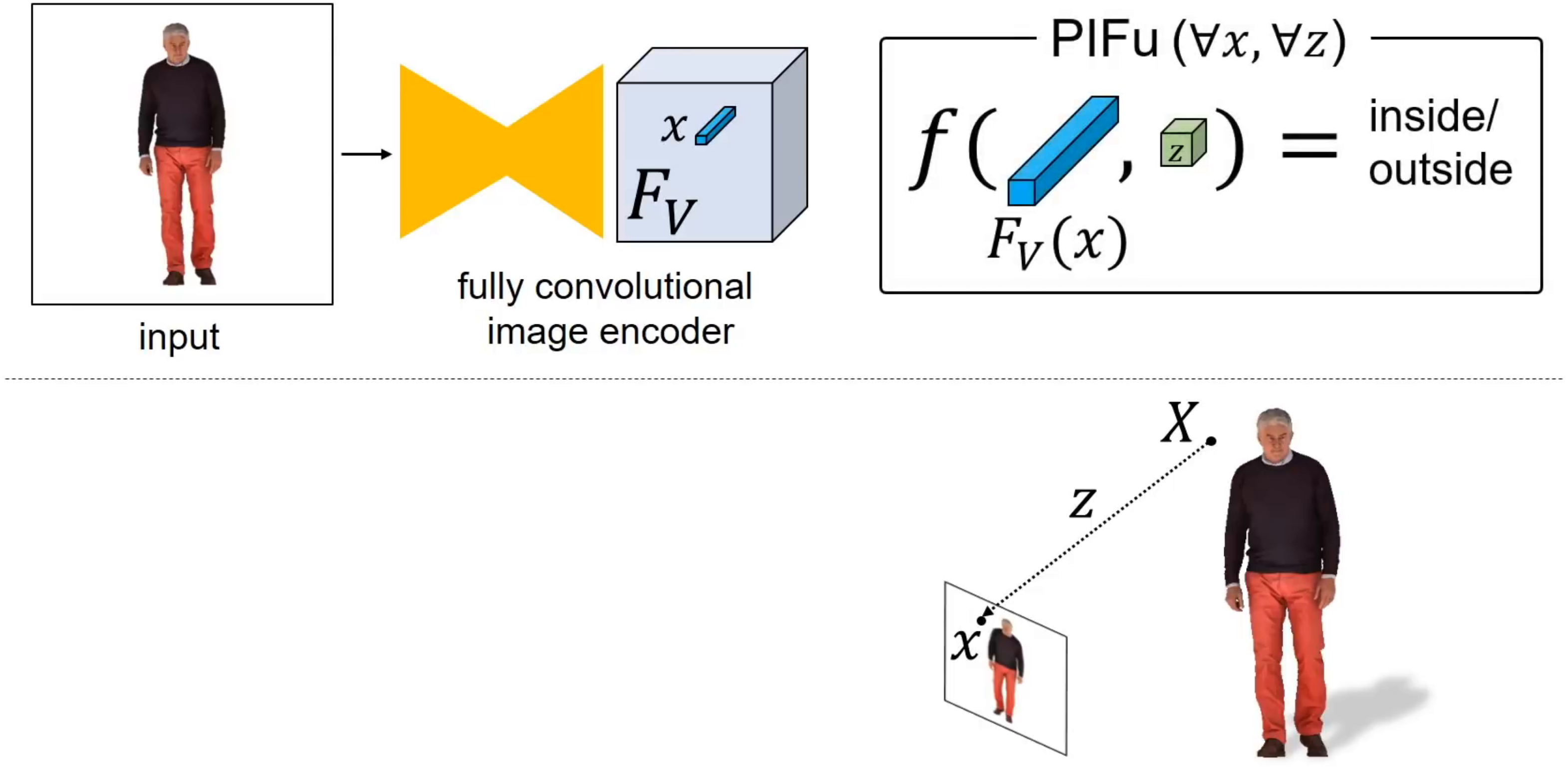
PIFu: Pixel-Aligned Implicit Function



PIFu: Pixel-Aligned Implicit Function



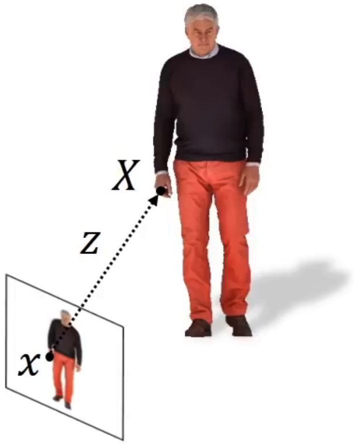
PIFu: Pixel-Aligned Implicit Function



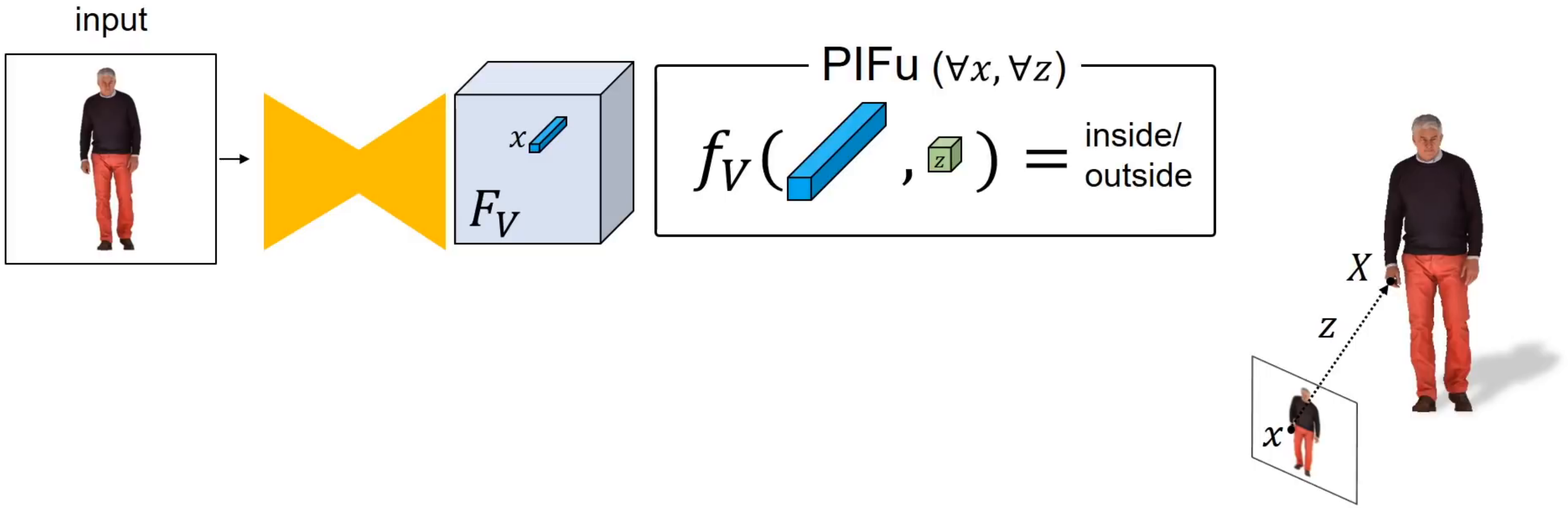
What about texture?

PiFu: Training Overview

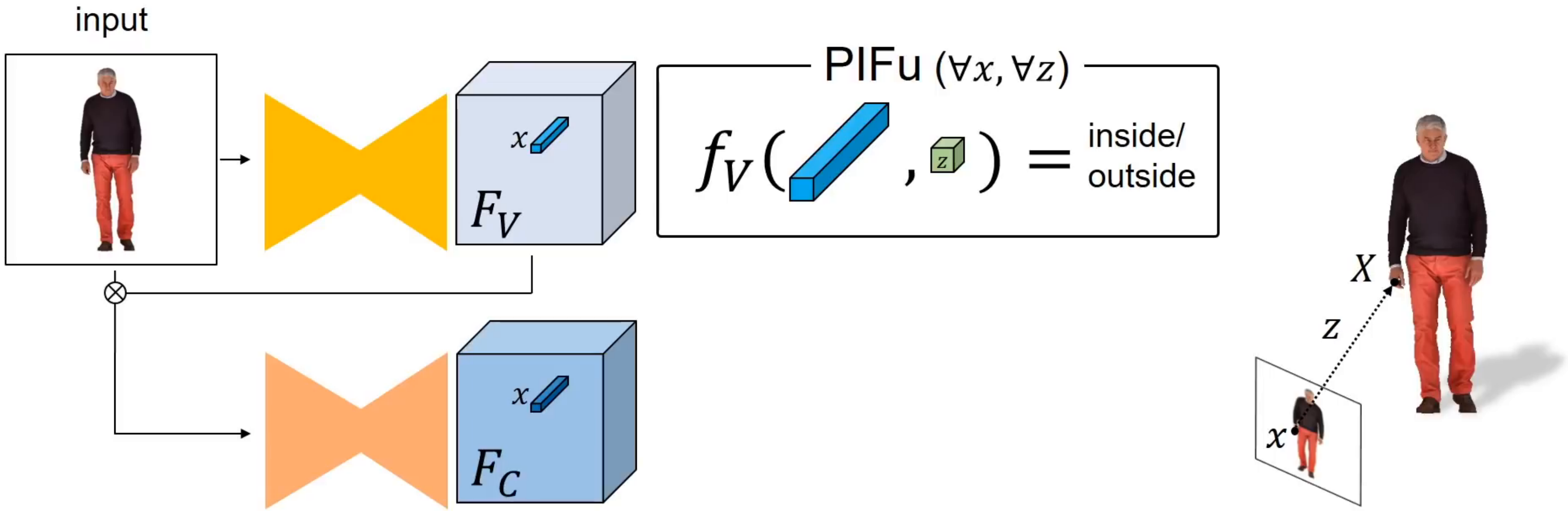
input



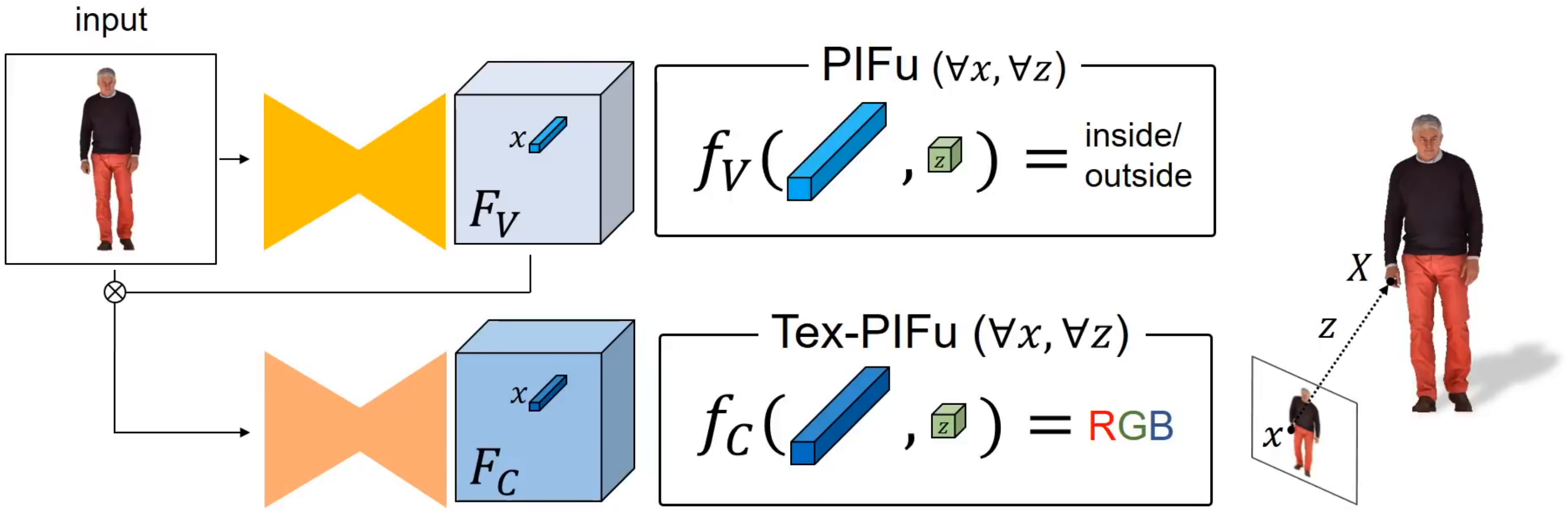
PiFu: Training Overview



PiFu: Training Overview



PiFu: Training Overview



Optimization Objective

- **Surface Reconstruction:**

$$\mathcal{L}_V = \frac{1}{n}\sum_{i=1}^{n} |f_v(F_V(x_i), z(X_i)) - \boxed{f_v^*(X_i)}|^2$$

Groundtruth surface: 1 if X is inside the mesh, 0 otherwise.

Optimization Objective

- **Surface Reconstruction:**

$$\mathcal{L}_V = \frac{1}{n}\sum_{i=1}^{n} |f_v(F_V(x_i), z(X_i)) - \boxed{f_v^*(X_i)}|^2$$

Groundtruth surface: 1 if X is inside the mesh, 0 otherwise.

- **Texture Reconstruction:**

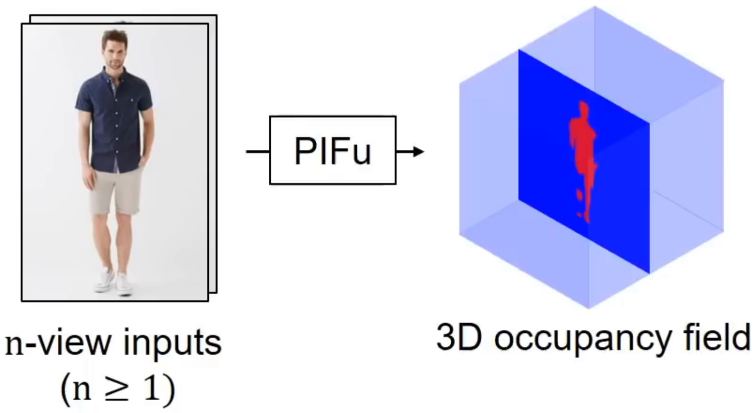
$$\mathcal{L}_C = \frac{1}{n}\sum_{i=1}^{n} |f_c(F_C(x_i), z(X_i)) - \boxed{C(X_i)}|$$

Groundtruth RGB value on the surface point

Inference

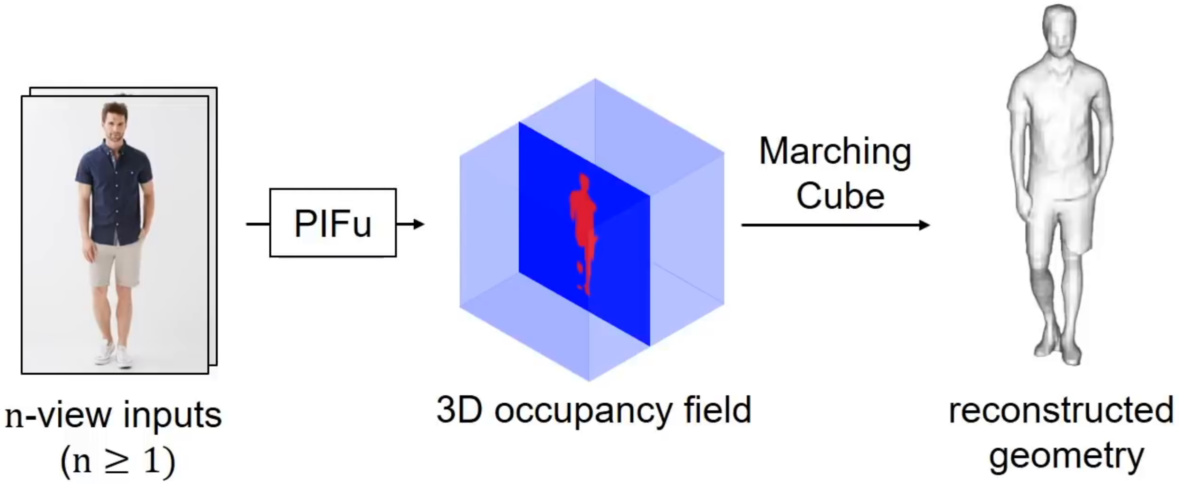
n-view inputs
$(n \geq 1)$

Inference



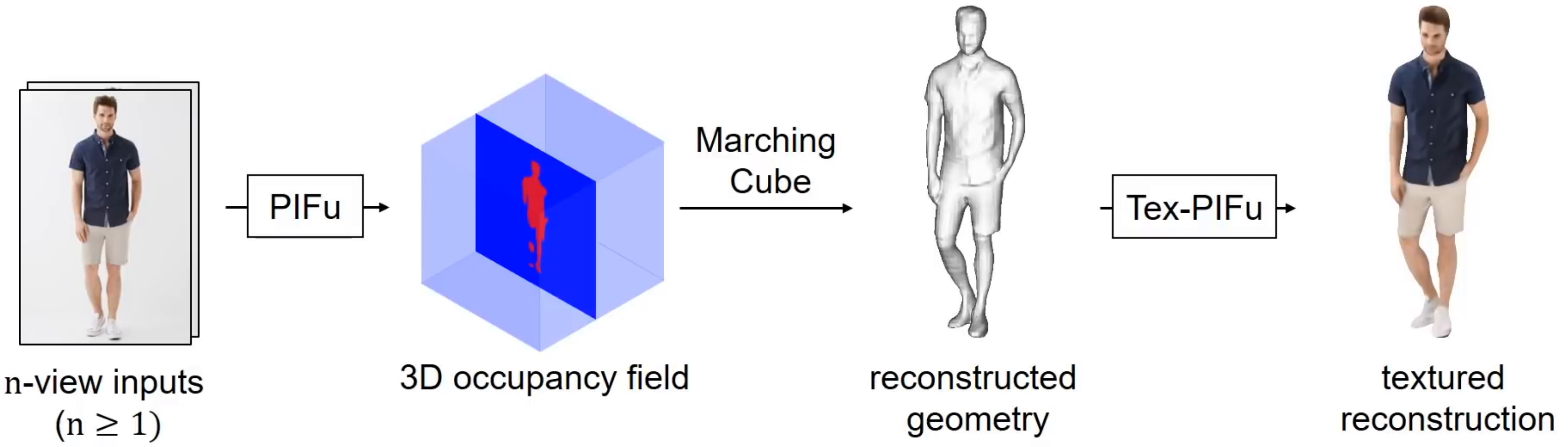
- PiFU maps the input image into a continuous occupancy field.

Inference



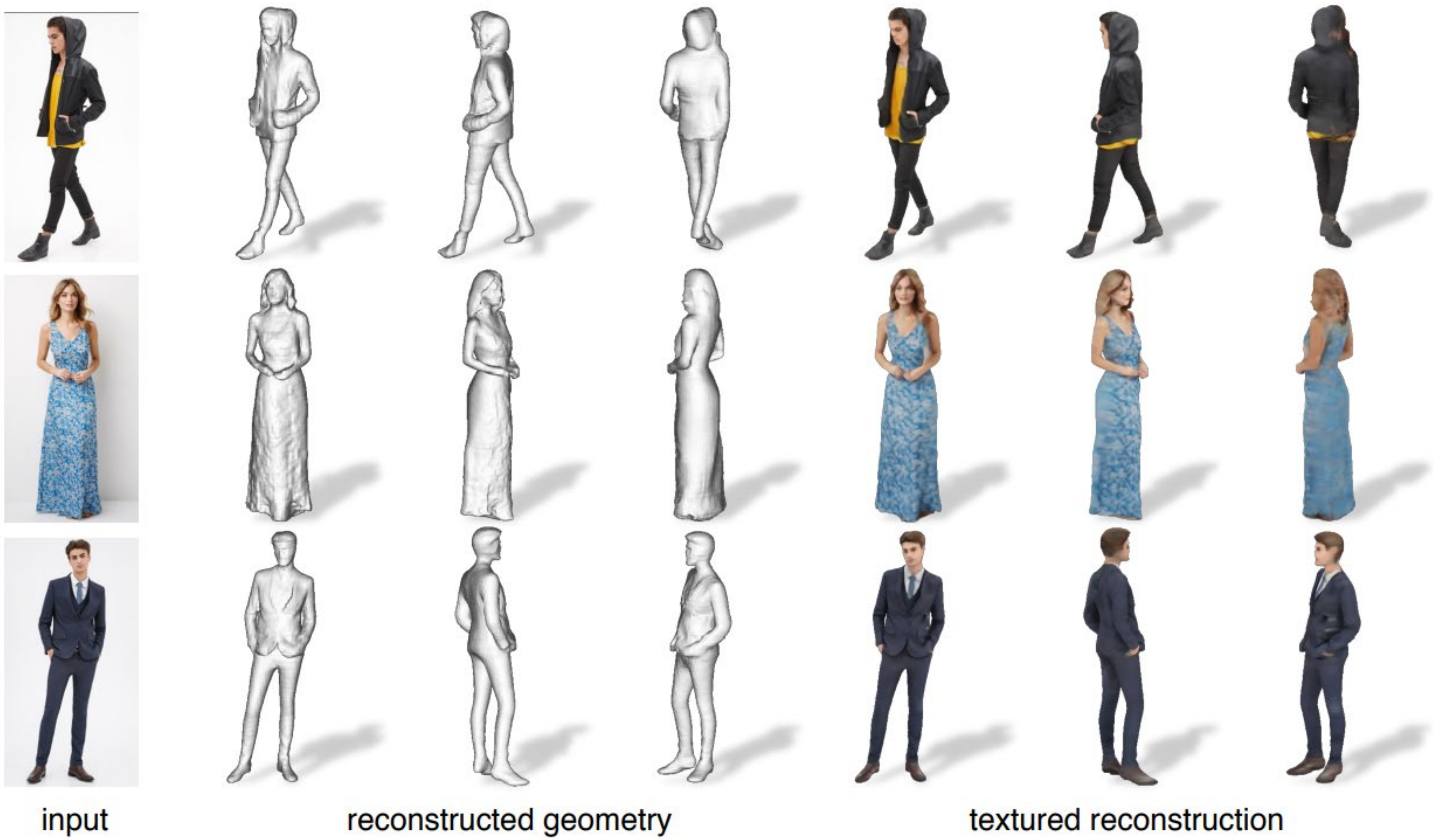
- PiFU maps the input image into a continuous occupancy field.
- Using Marching Cubes we can recover the surface of the object.

Inference

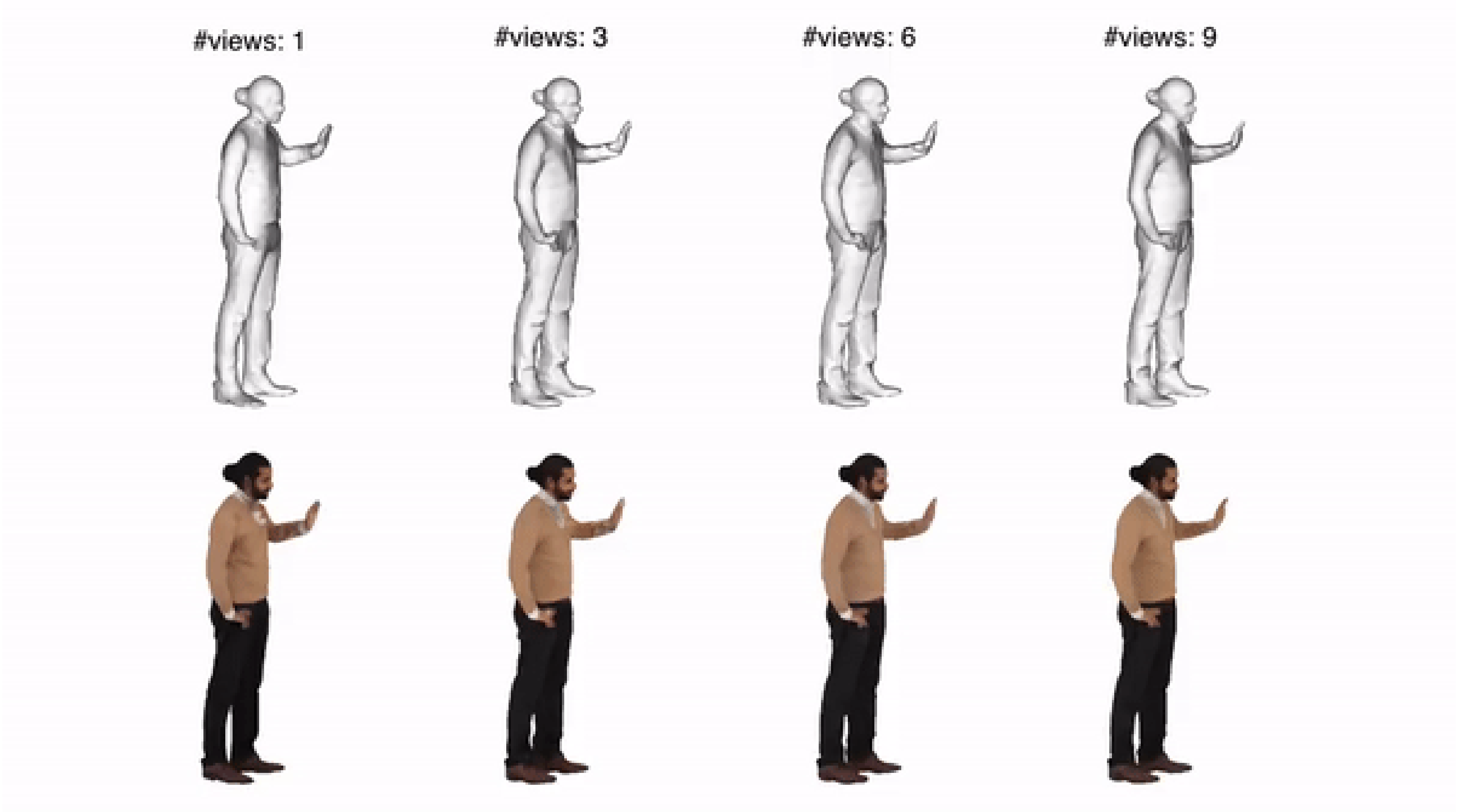


- PiFU maps the input image into a continuous occupancy field.
- Using Marching Cubes we can recover the surface of the object.
- For every point in the reconstructed surface, we use Text-PiFu to estimate its corresponding colour.

How well does it work?



How well does it work?



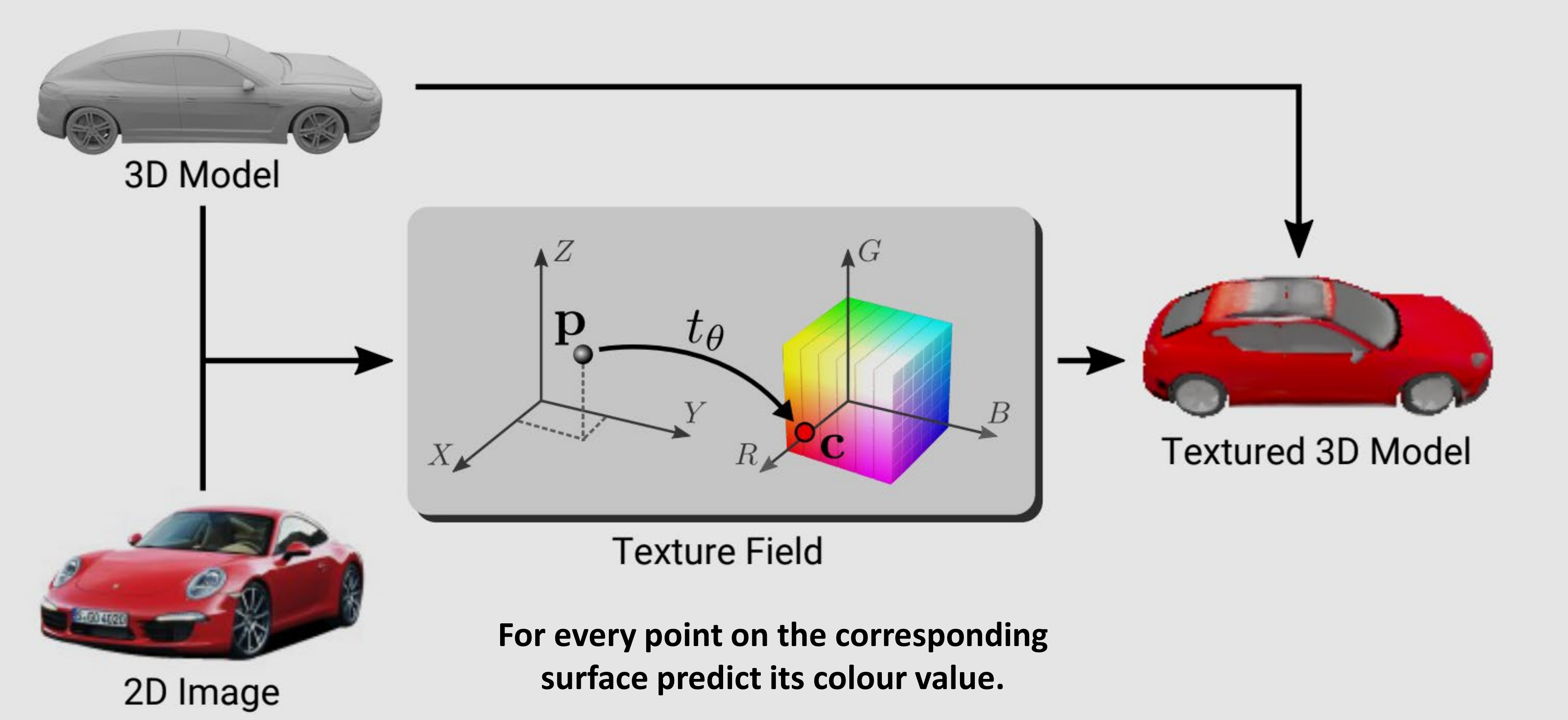
Texture Fields: Learning Texture Representations in Function Space

Michael Oechsle, Lars Mescheder, Michael Niemeyer,

Thilo Strauss, Andreas Geiger

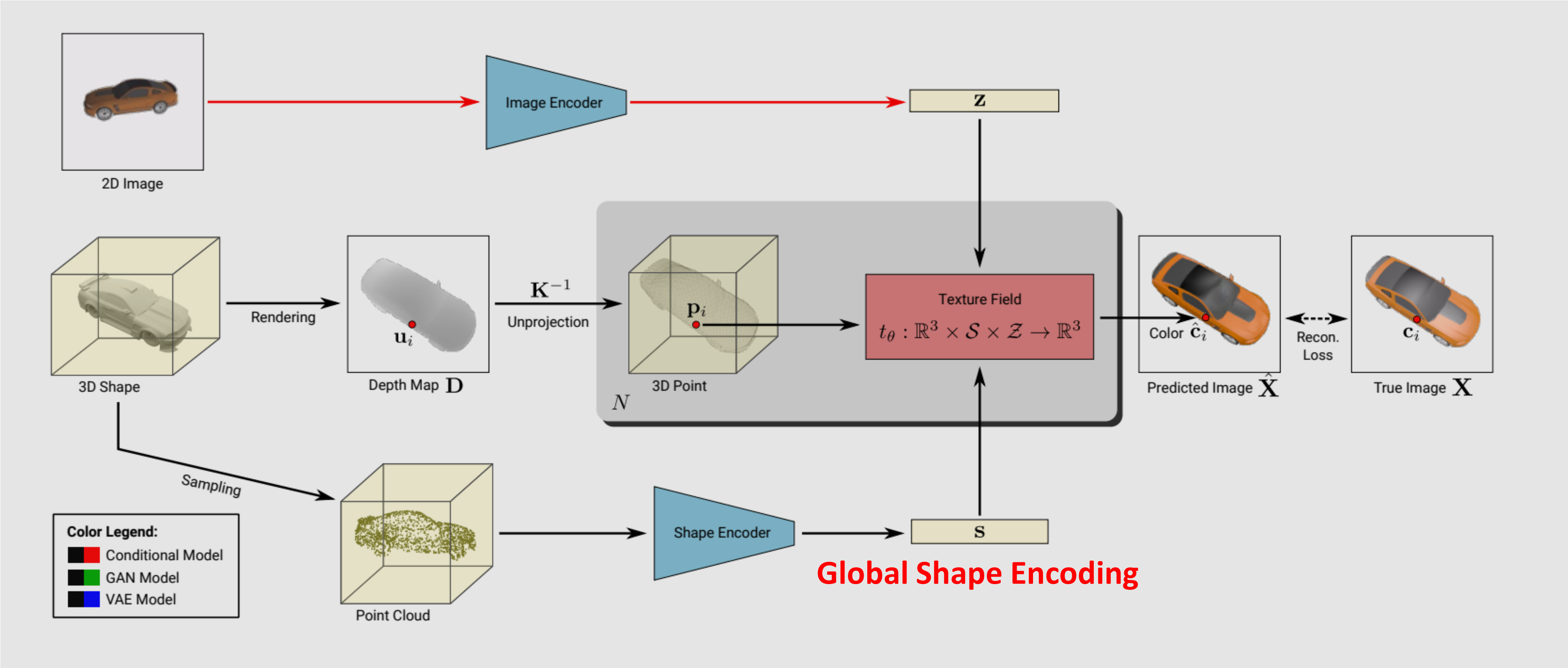
ICCV 2019

A similar idea: Texture Fields



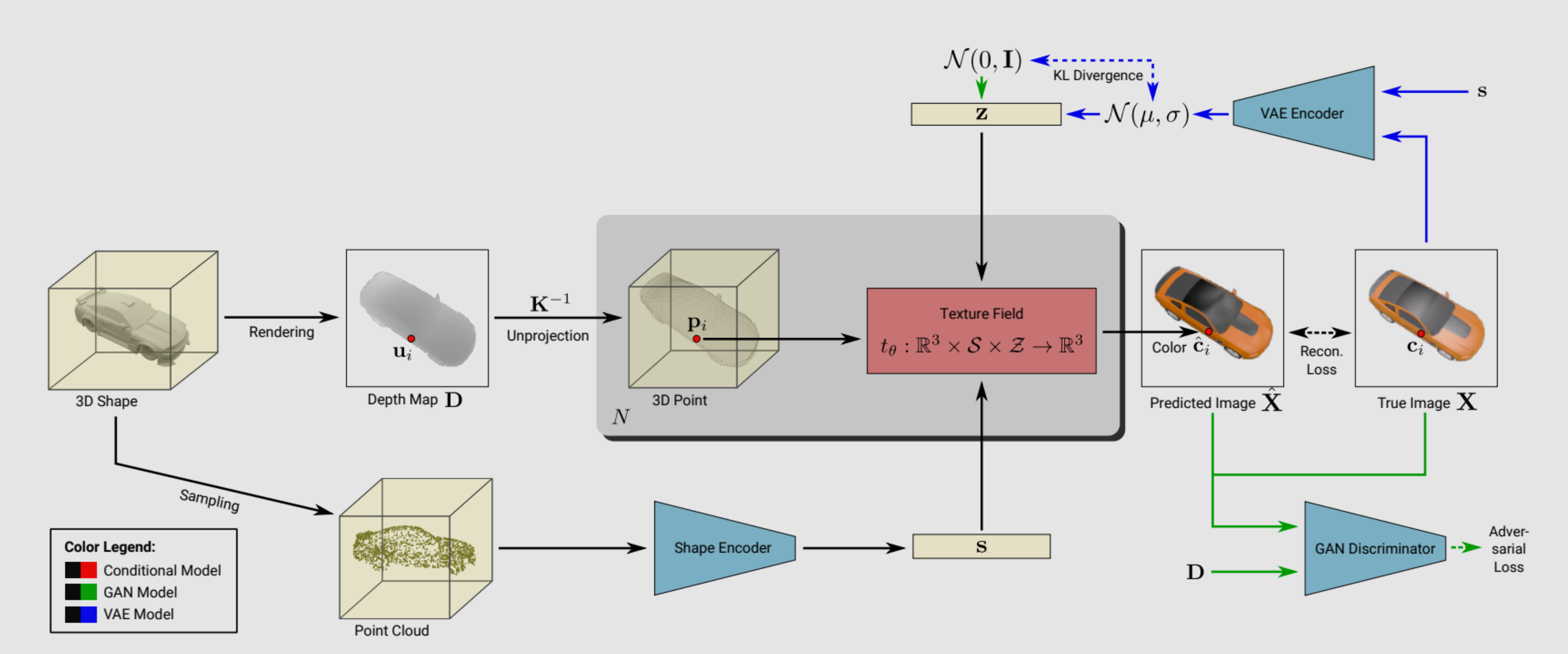
For every point on the corresponding surface predict its colour value.

A similar idea: Texture Fields



Global Shape Encoding

Texture Fields as Generative Models



What about Generative Models?

GAN

VAE

Quick Recap

Occupancy Networks:

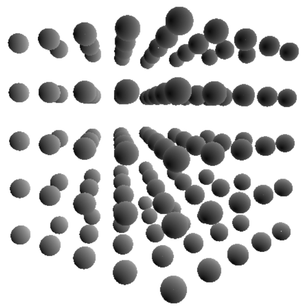
$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \rightarrow [0, 1]$$

3D Location

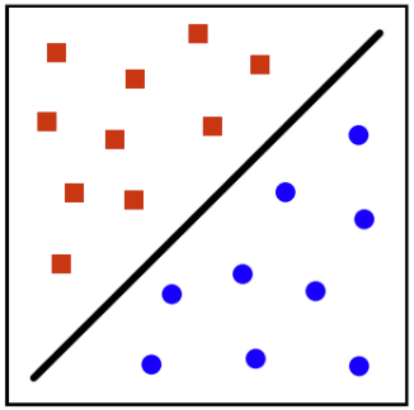
Condition (eg, Image)

Occupancy Probability

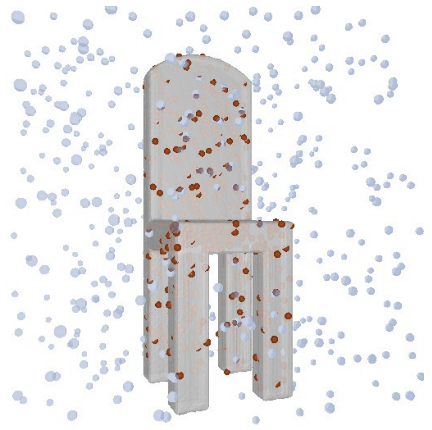
Condition



3D Locations



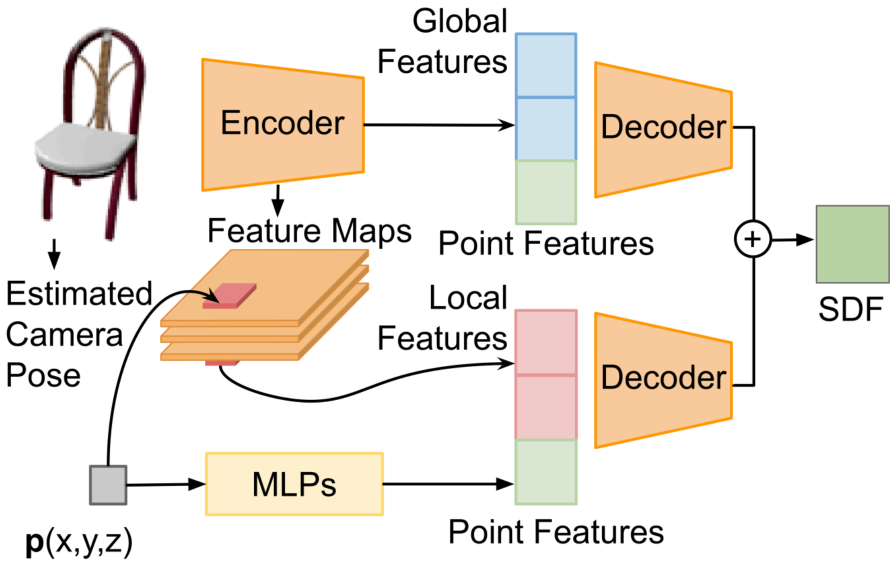
The **decision boundary of the classifier** models the **occupancy field.**



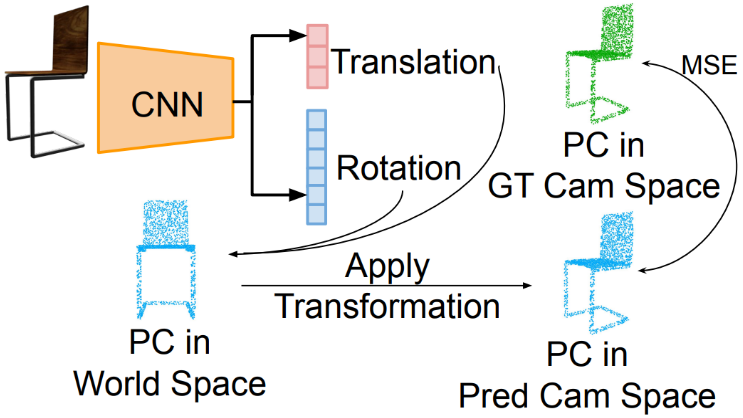
Occupancy Probability

Quick Recap

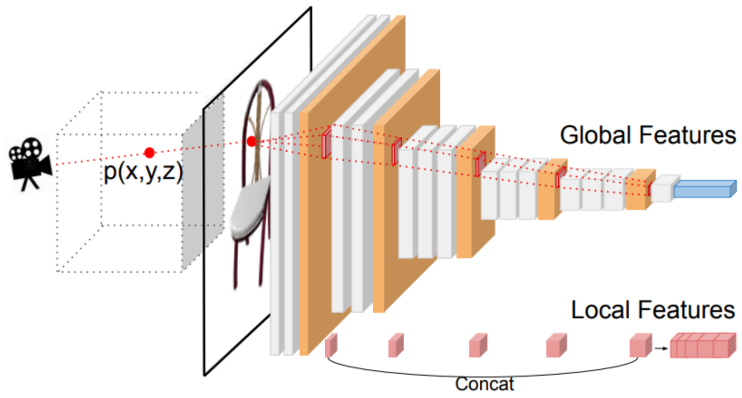
DISN:



Model Overview



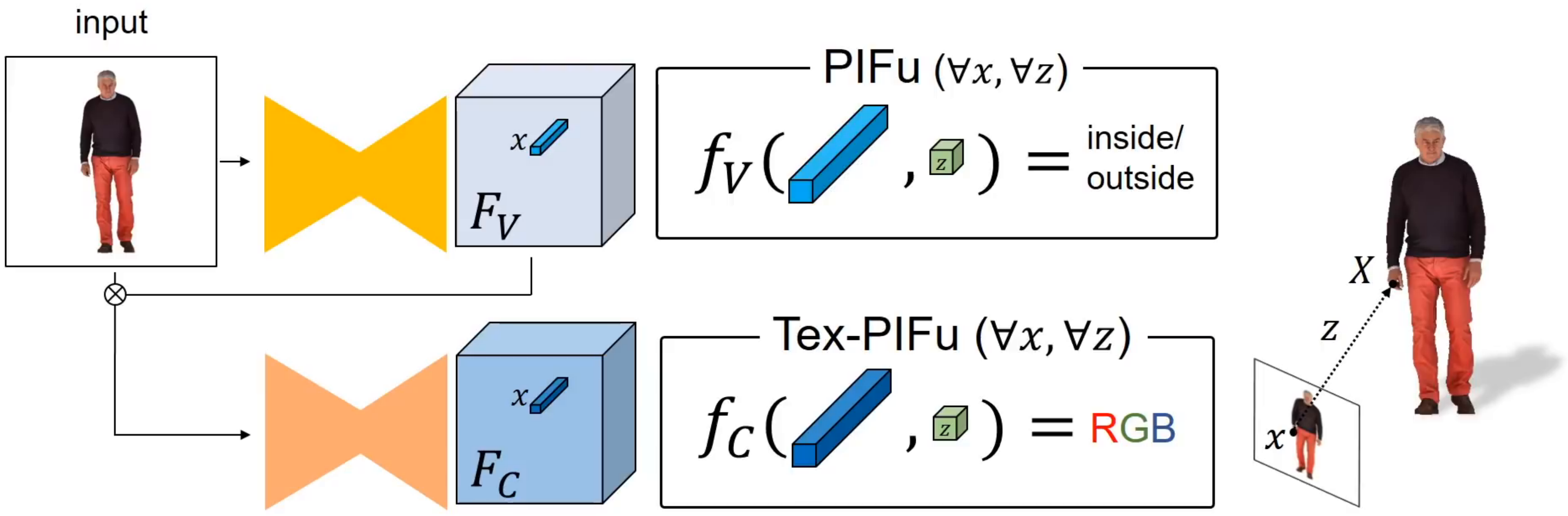
Camera Pose Estimation Network



Local Feature Extractor

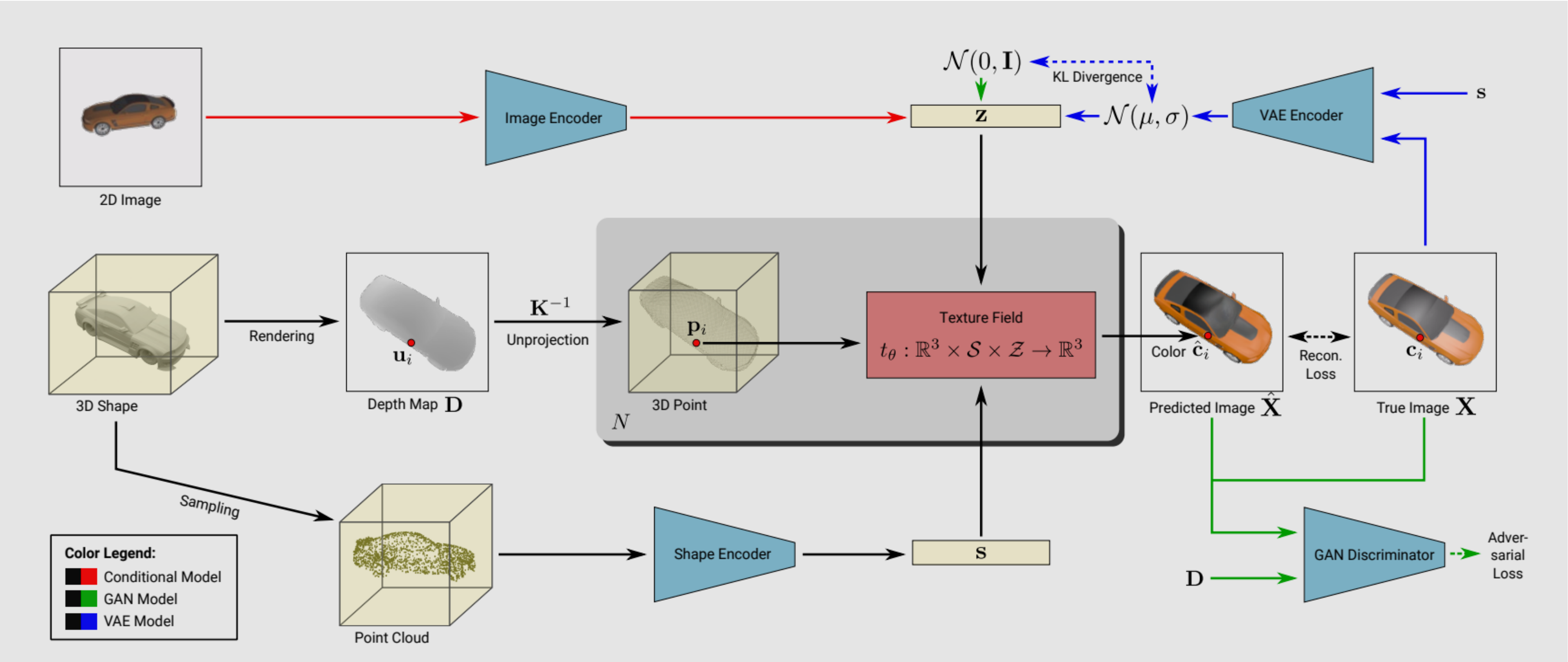
Quick Recap

- **PiFU:**



Quick Recap

- **Texture Fields:**



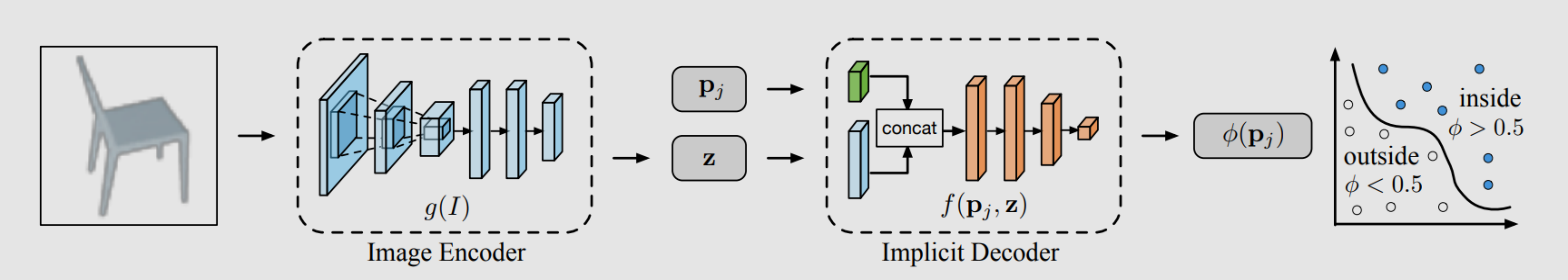
Learning to Infer Implicit Surfaces without 3D Supervision

Shichen Liu , Shunsuke Saito, Weikai Chen (B), Hao Li

NeurIPS 2019

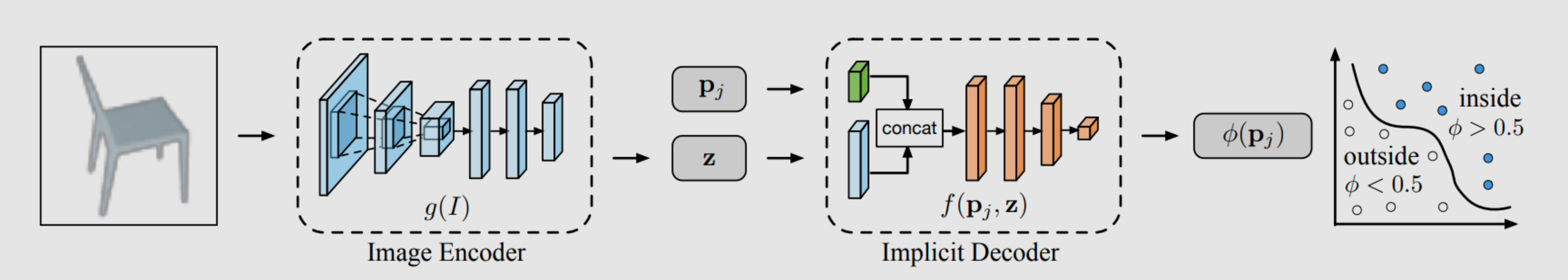
Infer Implicit Surfaces without 3D Supervision

- How can we learn to **infer implicit surfaces solely from images**, without any 3D supervision?



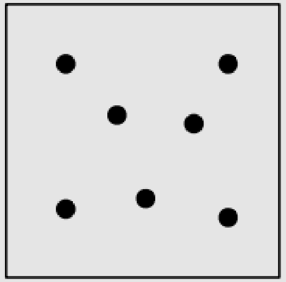
Infer Implicit Surfaces without 3D Supervision

- How can we learn to **infer implicit surfaces solely from images**, without any 3D supervision?



How do we define our loss function?

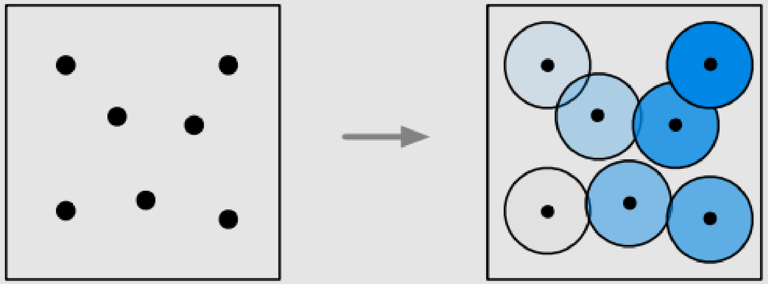
Ray-based Field Probing



(a) 3D anchor points

- Sample a sparse set of 3D points.

Ray-based Field Probing

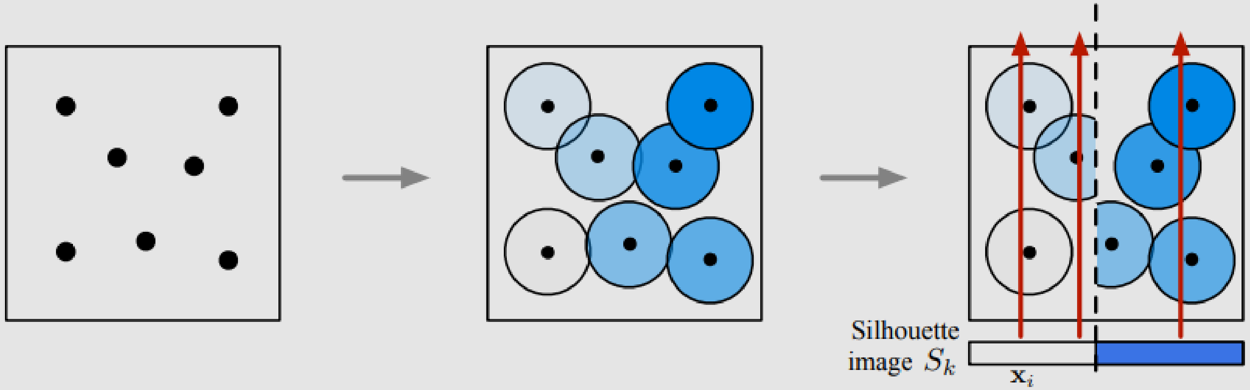


(a) 3D anchor points

(b) Occupancy field evaluation

- Sample a sparse set of 3D points.
- For each 3D point compute its occupancy value. Each point is assigned a support region to enable ray point intersection.

Ray-based Field Probing



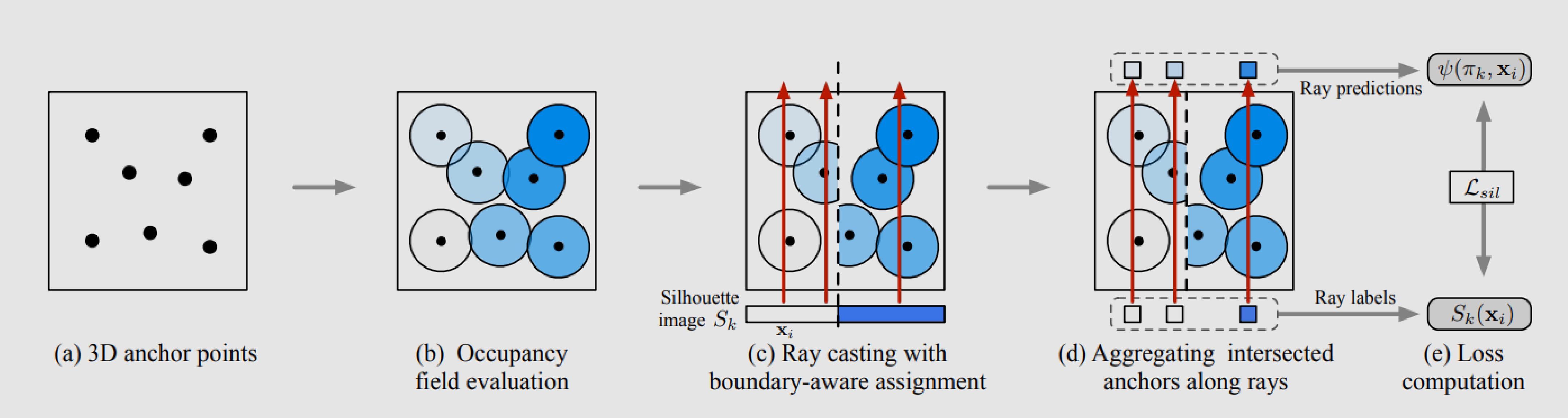
(a) 3D anchor points

(b) Occupancy field evaluation

(c) Ray casting with boundary-aware assignment

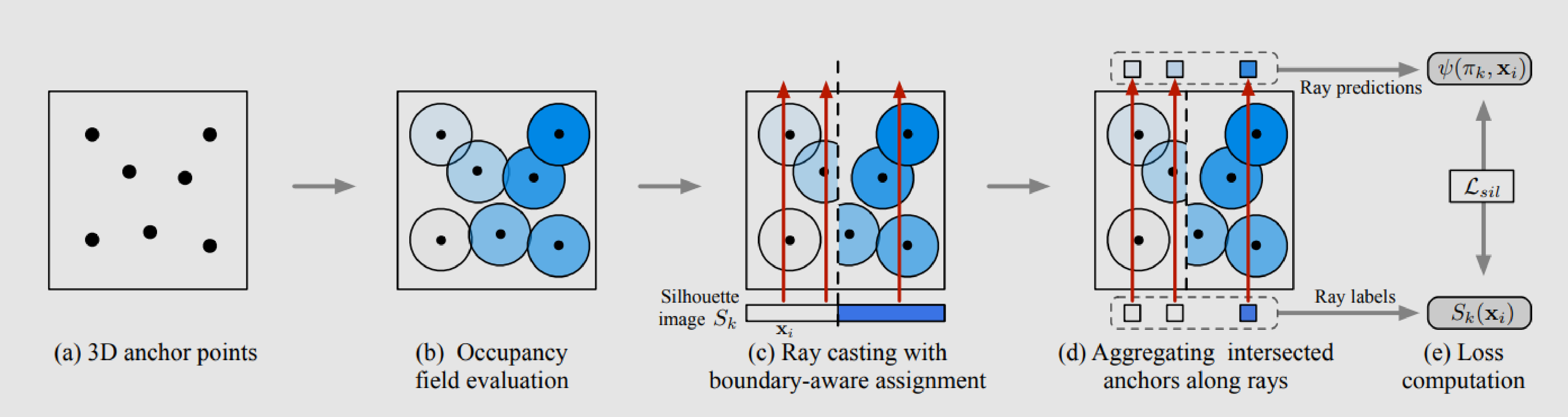
- Sample a sparse set of 3D points.
- For each 3D point compute its occupancy value. Each point is assigned a support region to enable ray point intersection.
- **Cast rays through the 3D points to the 2D silhouette** under a fixed camera view.

Ray-based Field Probing



- Sample a sparse set of 3D points.
- For each 3D point compute its occupancy value. Each point is assigned a support region to enable ray point intersection.
- **Cast rays through the 3D points to the 2D silhouette** under a fixed camera view.
- **Aggregate the information** from the intersected points along each ray and **get a per ray prediction**.

Ray-based Field Probing



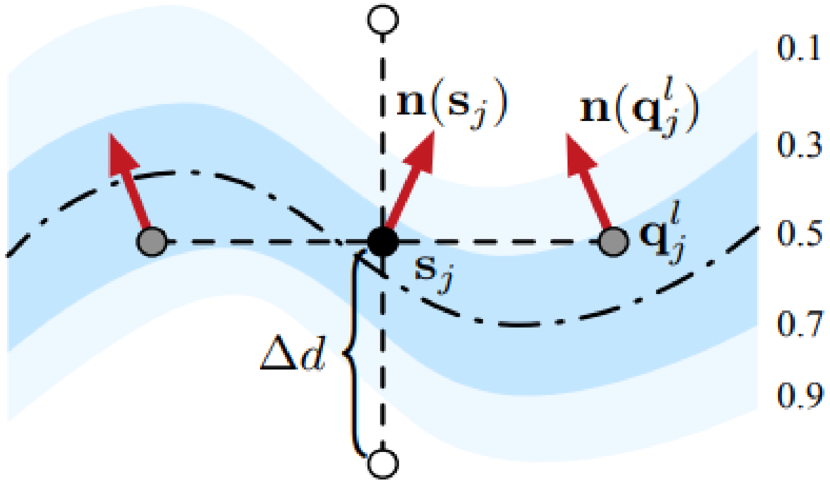
- Sample a sparse set of 3D points.
- For each 3D point compute its occupancy value. Each point is assigned a support region to enable ray point intersection.
- **Cast rays through the 3D points to the 2D silhouette** under a fixed camera view.
- **Aggregate the information** from the intersected points along each ray and **get a per ray prediction**.

$$\mathcal{L}_{sil} = \frac{1}{N_r} \sum_{i=1}^{N_r} \sum_{k=1}^{N_K} \|\psi(\pi_k, \mathbf{x}_i) - S_k(\mathbf{x}_i)\|^2$$

Geometric Regularization on Implicit Surfaces

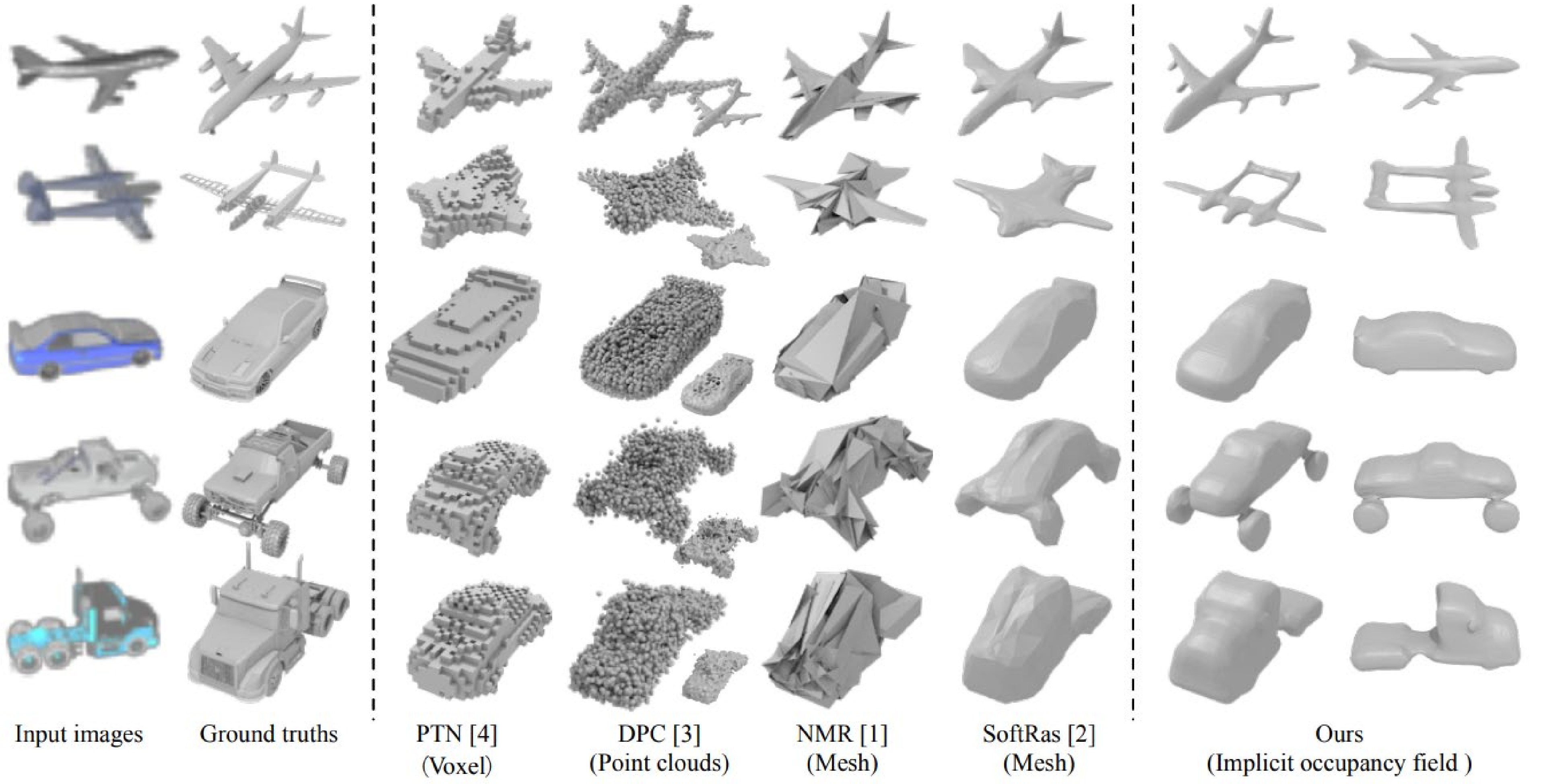
$$\mathcal{L}_{geo} = \frac{1}{N_p} \sum_{j=1}^{N_p} W(\phi(\mathbf{s}_j)) \frac{\sum_{l=1}^{6} W(\phi(\mathbf{q}_j^l)) \|\mathbf{n}(\mathbf{s}_j) - \mathbf{n}(\mathbf{q}_j^l)\|_p^p}{\sum_{l=1}^{6} W(\phi(\mathbf{q}_j^l))}$$

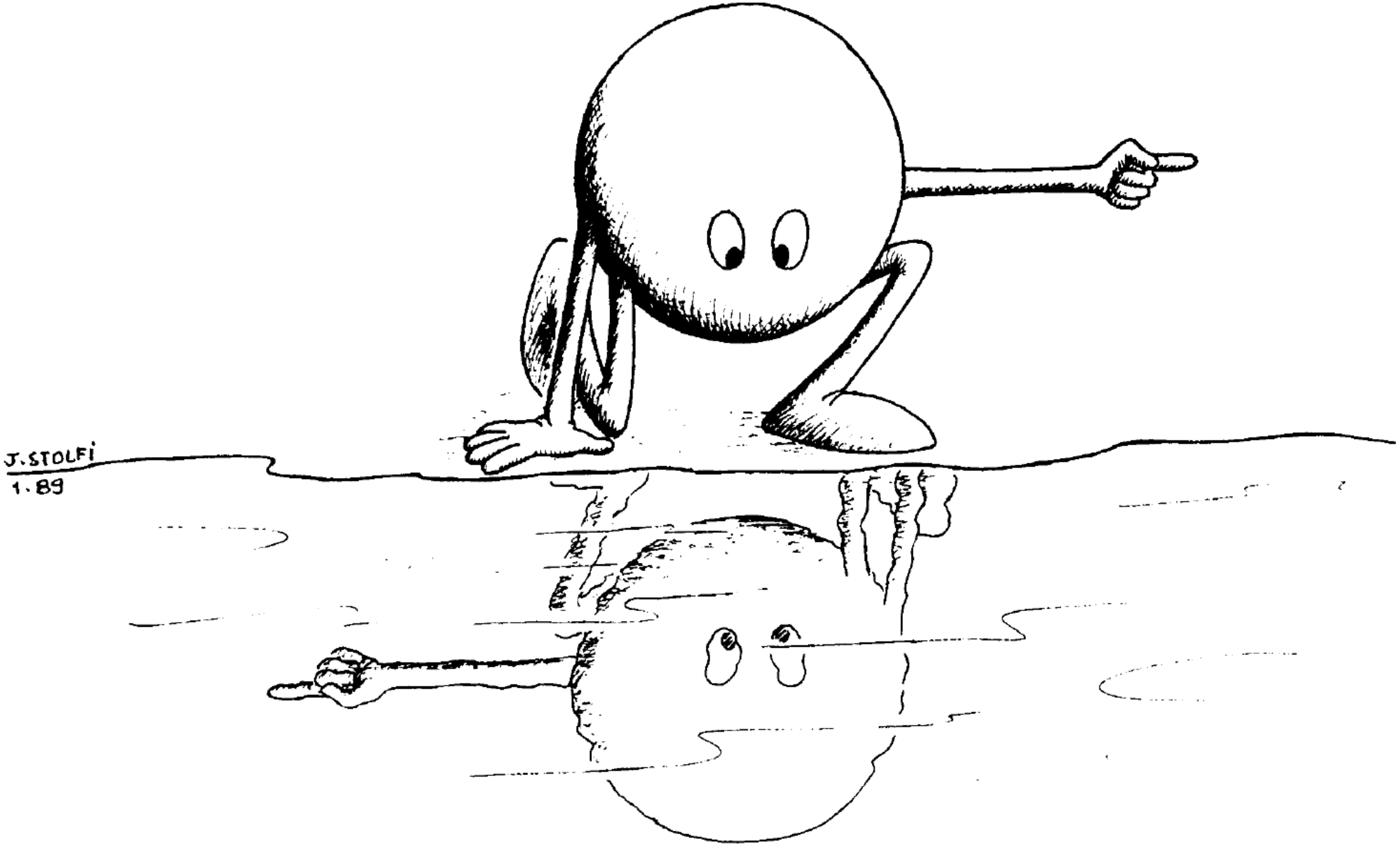
$$W(x) = \mathbb{I}(|x - 0.5| < \epsilon)$$



$$\mathbf{n}(\mathbf{p}_j) = \frac{\delta \phi}{\delta \mathbf{p}_j} \Big/ \left| \frac{\delta \phi}{\delta \mathbf{p}_j} \right|$$

How well does it work?




J. STOLFI
1.89