Seeing the unseen

Data-driven 3D Understanding from Single Images

Image world

Shape world

Hao Su

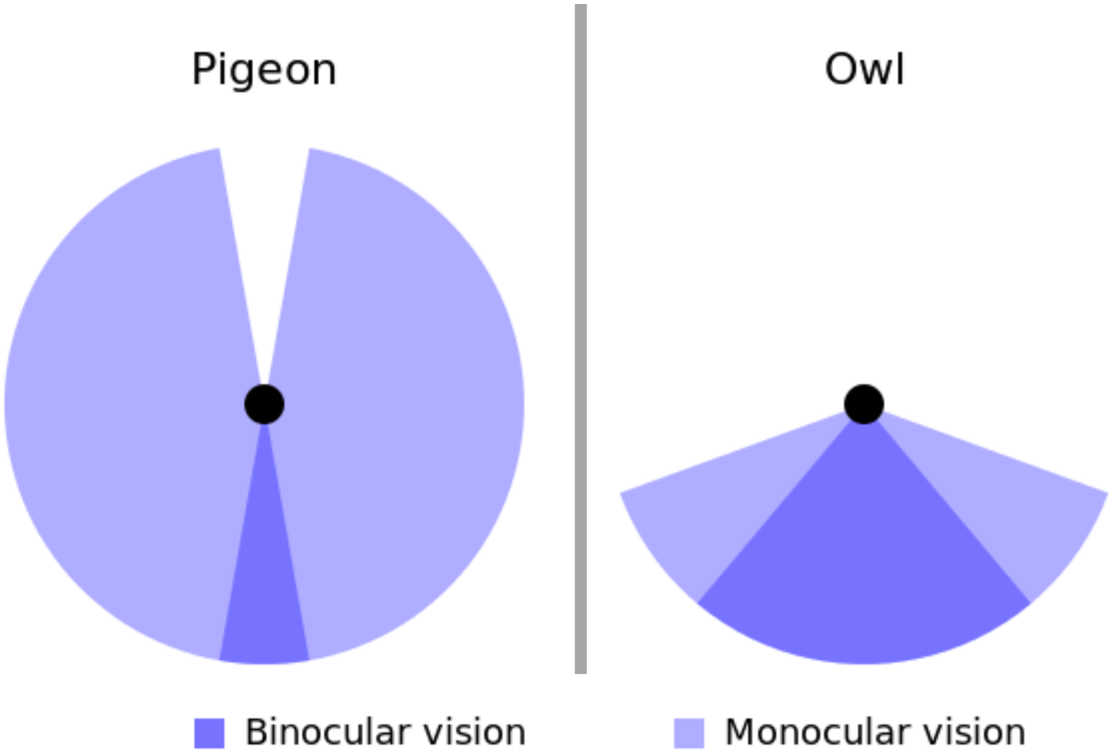
Stanford University

3D perception from a single image

Monocular vision

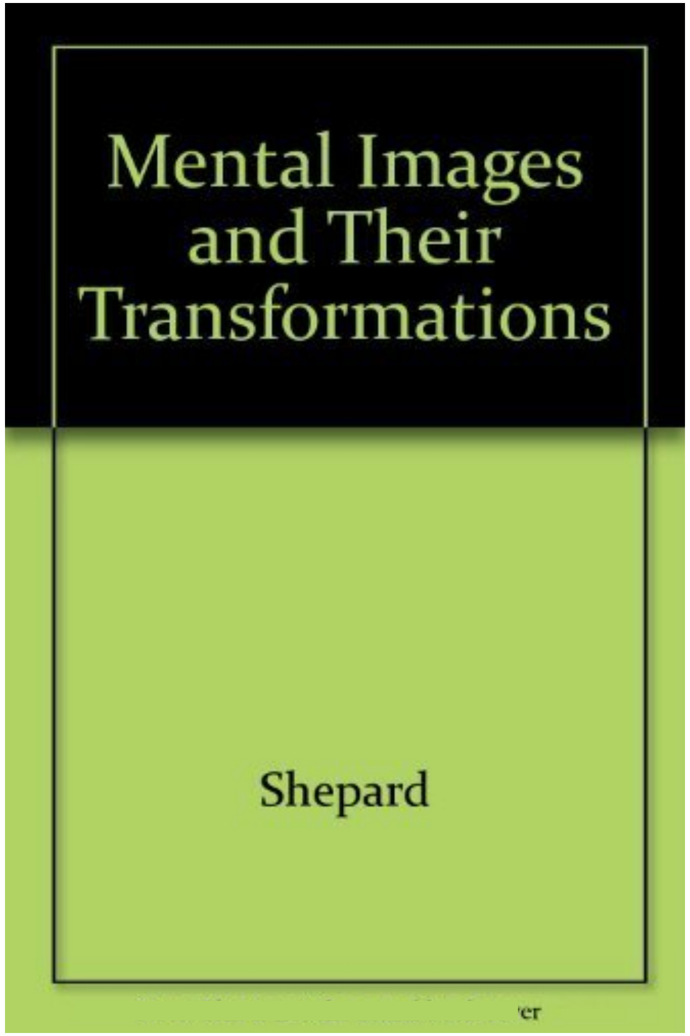
a typical prey

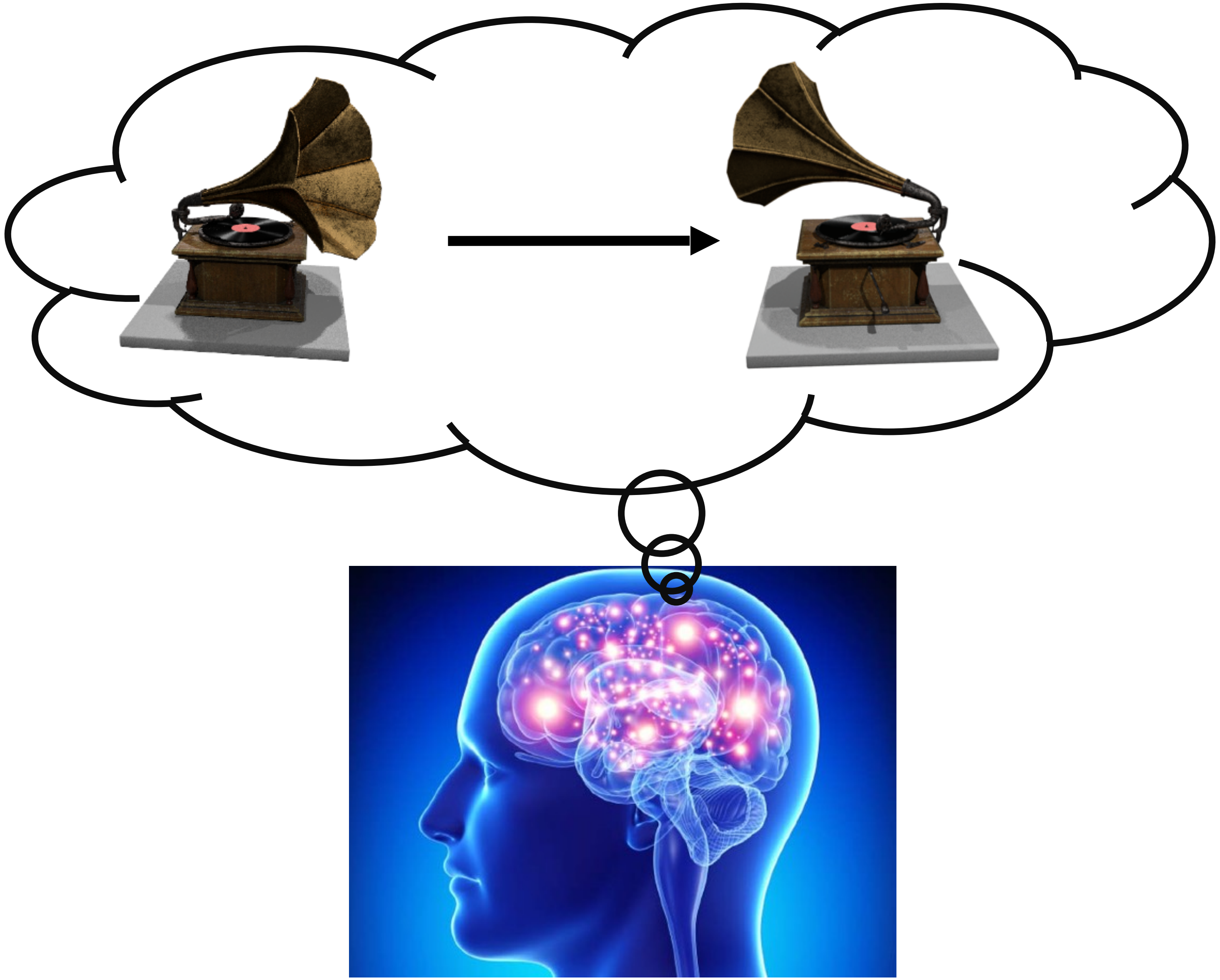
a typical predator



Cited from https://en.wikipedia.org/wiki/Binocular_vision

A psychological evidence – mental rotation





by Roger N. Shepard, National Science Medal Laurate
and Lynn Cooper, Professor at Columbia University

Visual cues are complicated

contrast

color

texture

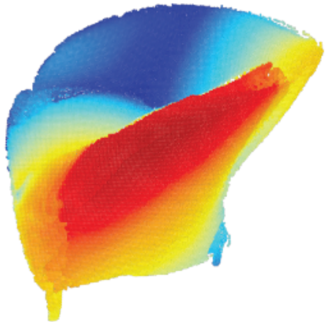
motion

symmetry

part

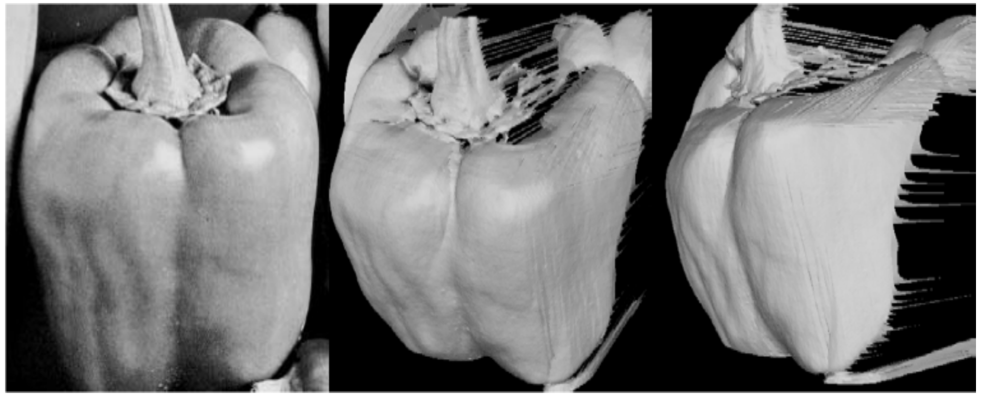
category-specific 3D knowledge

……

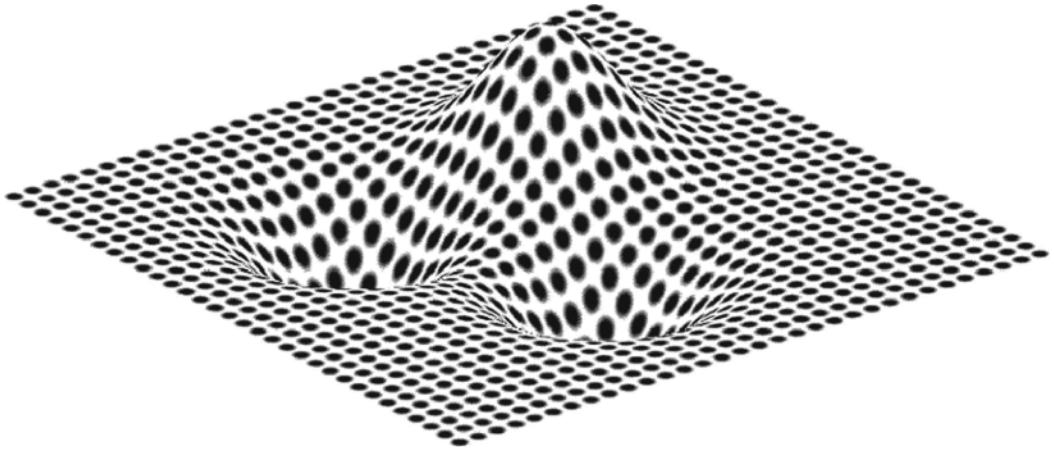


Status review of monocular vision algorithms

- Shape from X (texture, shading, …)



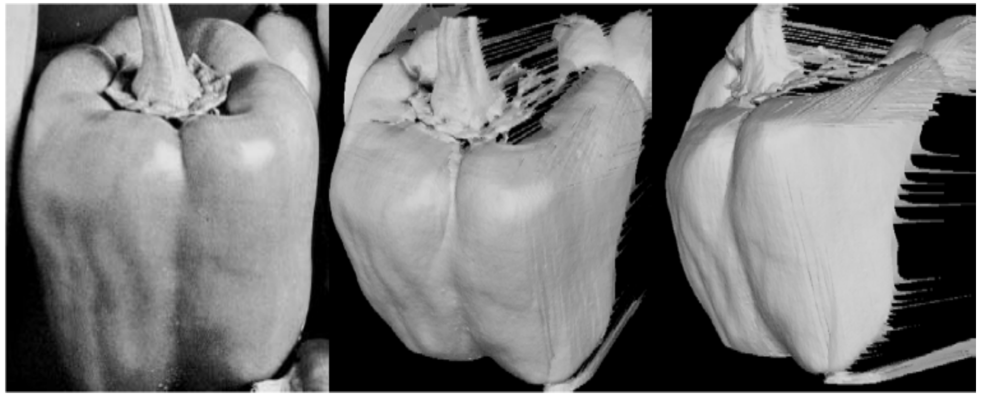
[Horn, 1989]



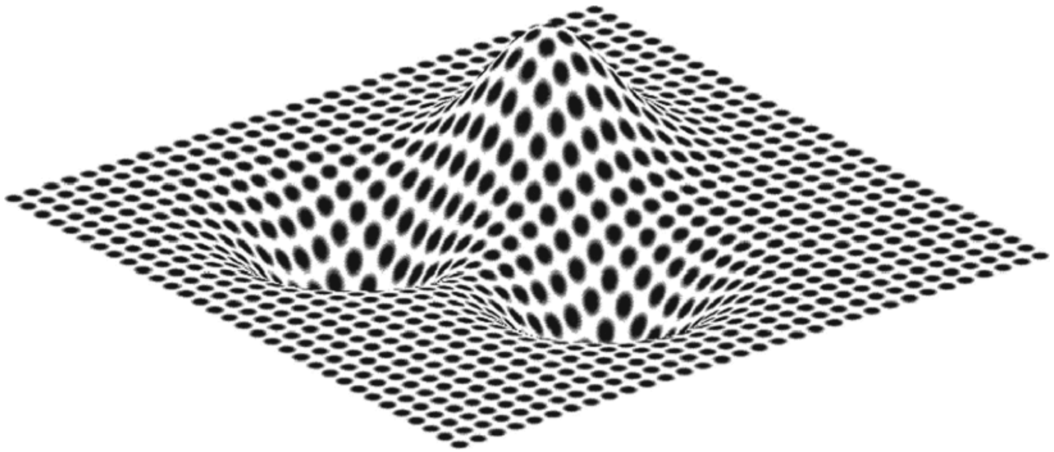
[Kender, 1979]

Status review of monocular vision algorithms

- Shape from X (texture, shading, …)



[Horn, 1989]



[Kender, 1979]

- Learning-based (from small data)

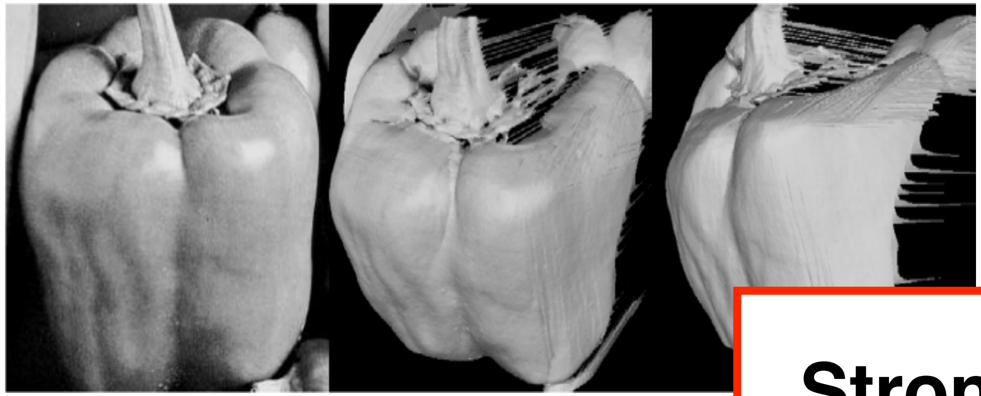
Hoiem et al, ICCV'05
Saxena et al, NIPS'05
…

- large planes

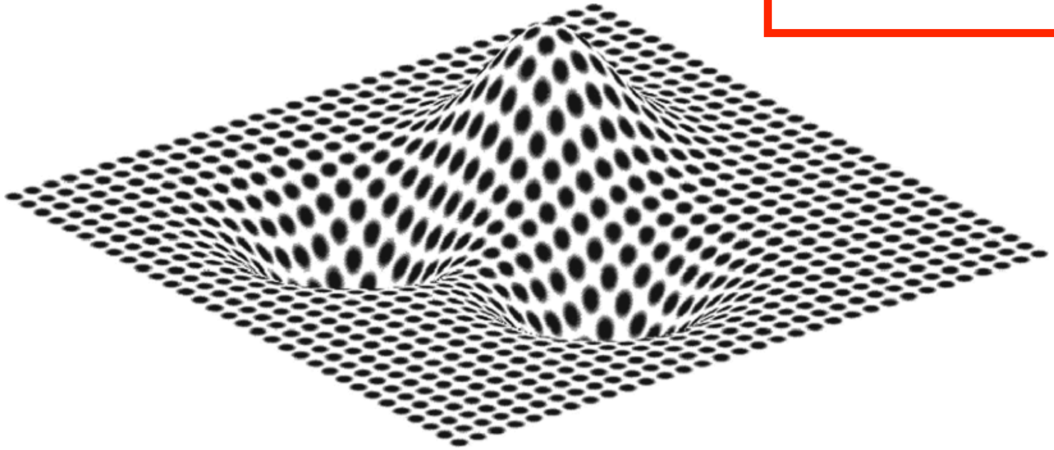
- fine structure
- topological variation
- …

Status review of monocular vision algorithms

- Shape from X (texture, shading, …)



[Horn, 1989]



[Kender, 1979]

- Learning-based (from small data)

Hoiem et al, ICCV’05
Saxena et al, NIPS’05
…

- large planes

Strong assumption
Not robust

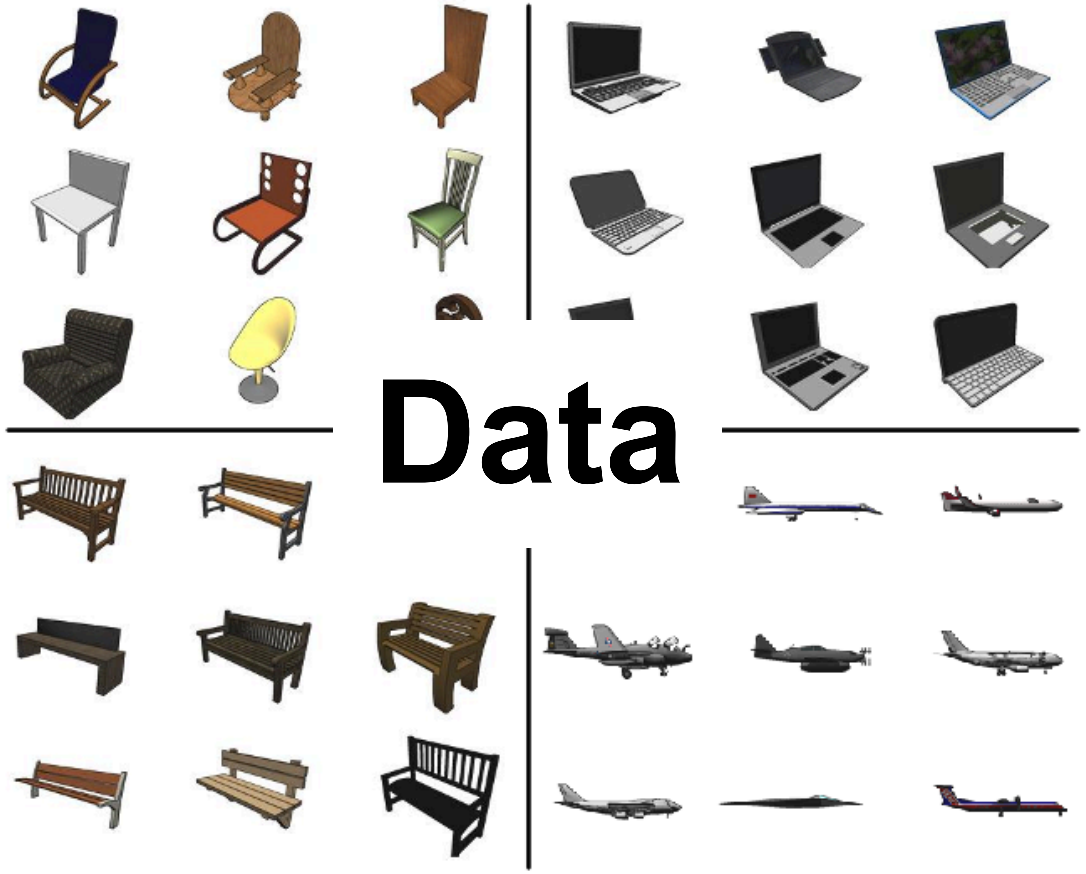
- fine structure
- topological variation
- …

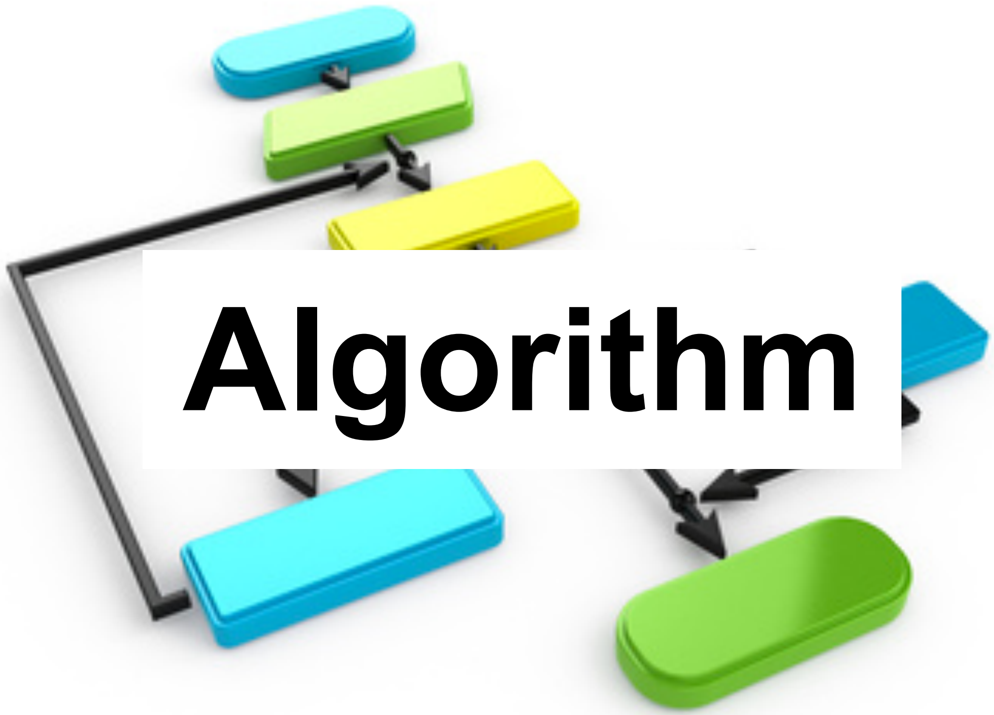
Data-driven 2D-3D lifting

Many 3D objects

A priori knowledge of the 3D world

Two pillars for learning-based 3D understanding

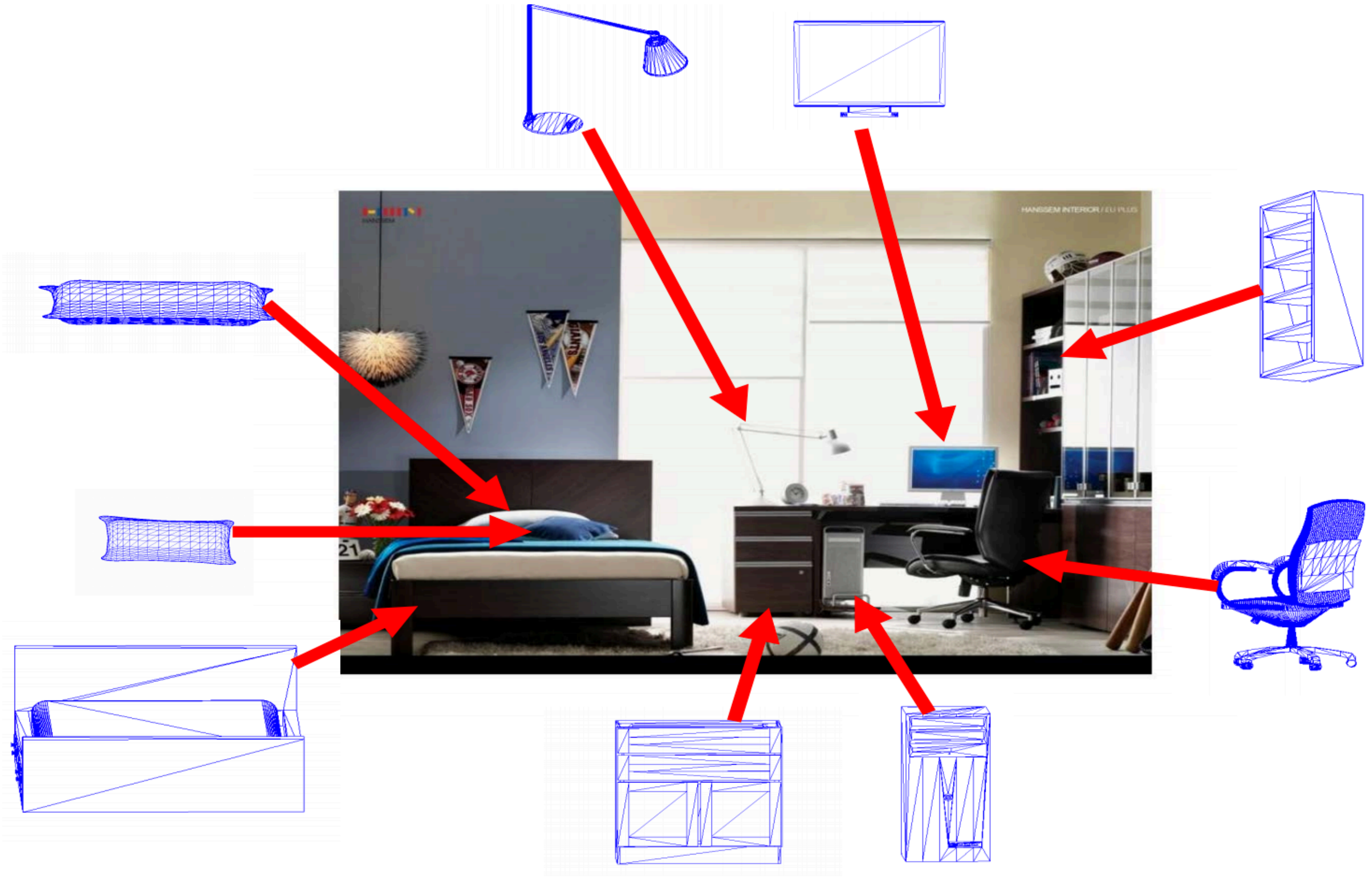




Outline

- Motivation
- **Data**
- Algorithms for 2D-3D lifting
- Algorithm for 3D representation learning

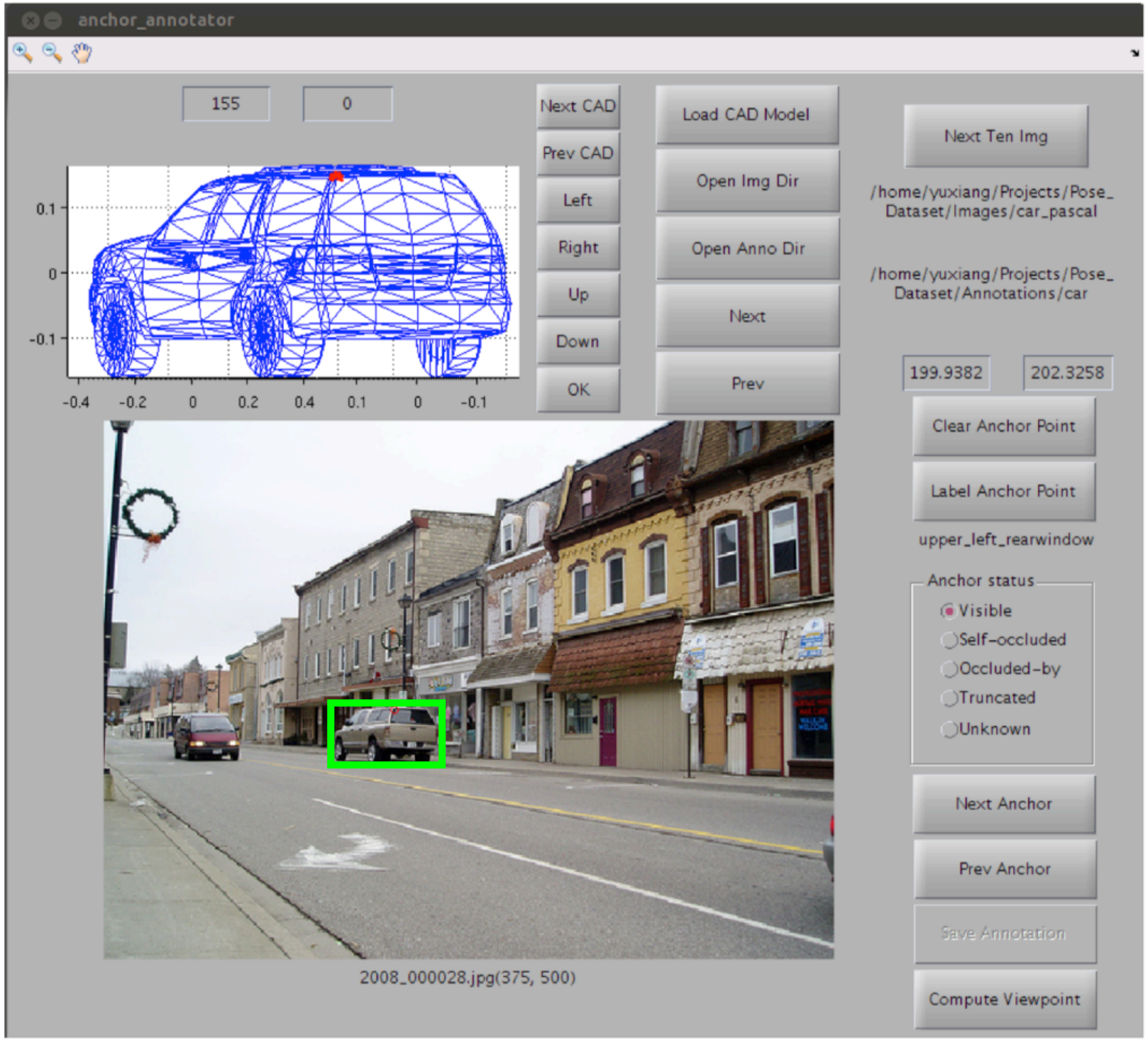
Need many 2D images with 3D labels



Accurate 3D label annotation is expensive

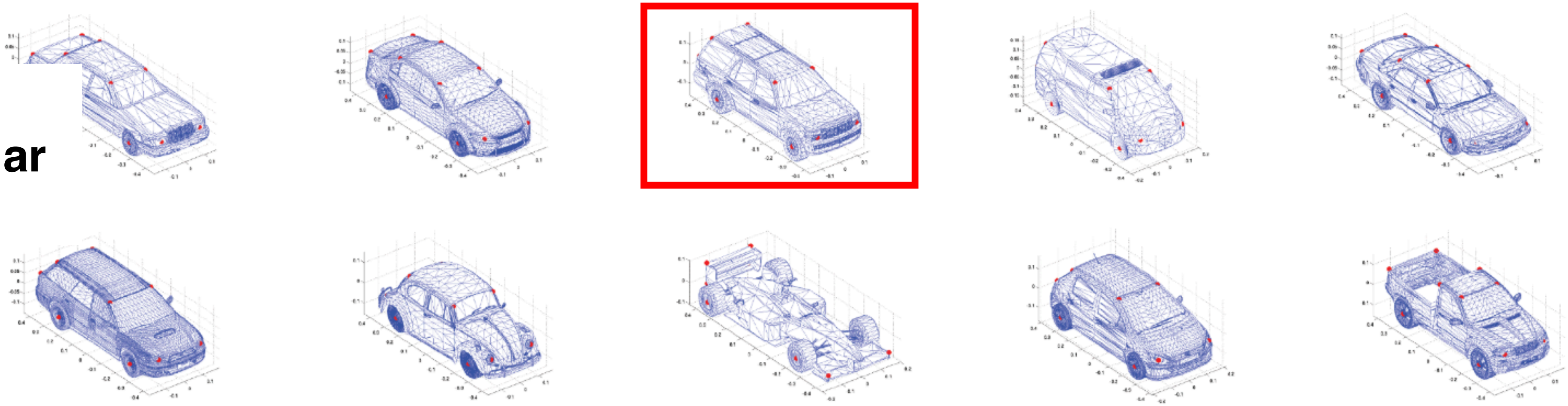
Example: viewpoint estimation

PASCAL3D+ dataset [Xiang et al.]



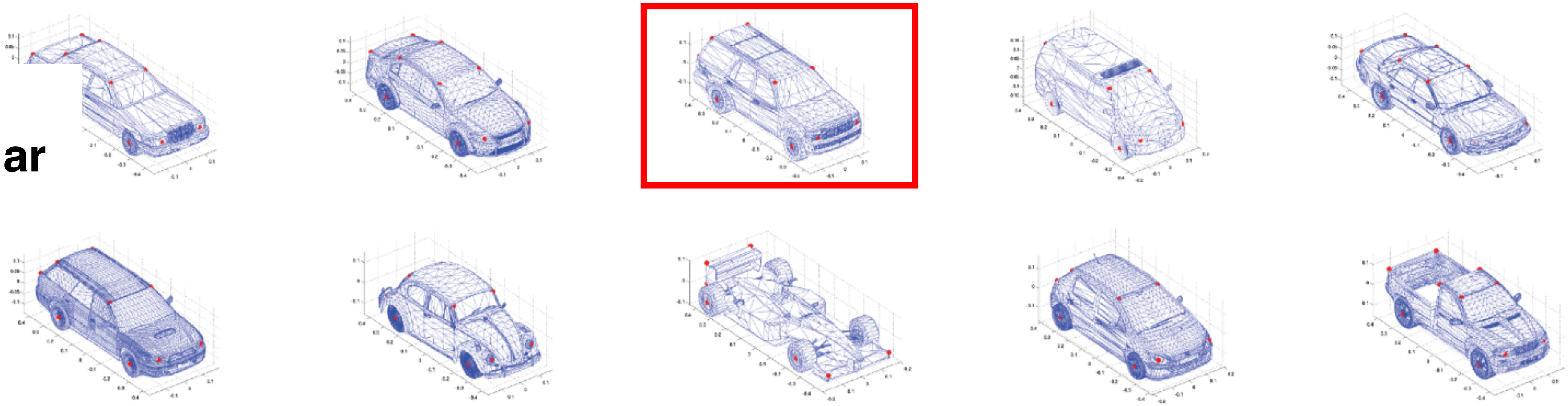
Accurate 3D label annotation is expensive

Step1:
Choose similar model

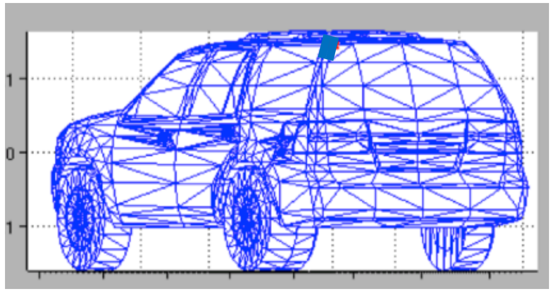


Accurate 3D label annotation is expensive

Step1:
Choose similar model

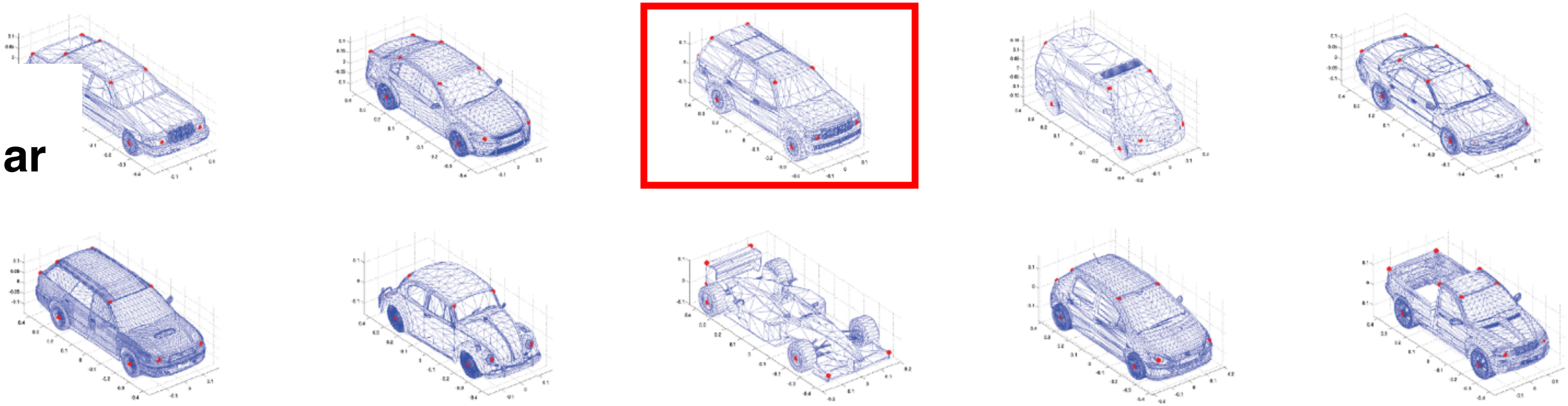


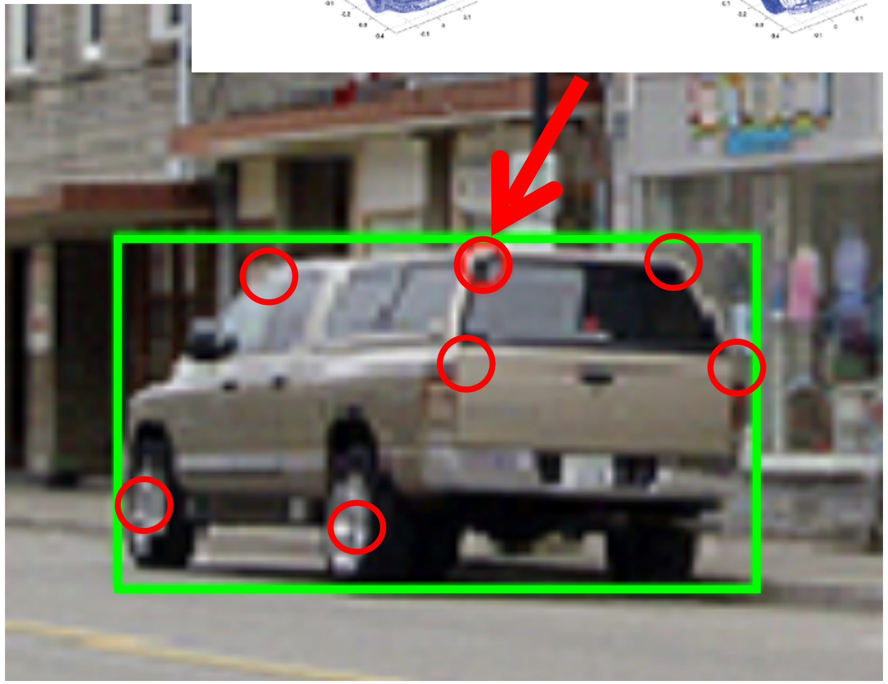
Step2:
Coarse Viewpoint Labeling



Accurate 3D label annotation is expensive

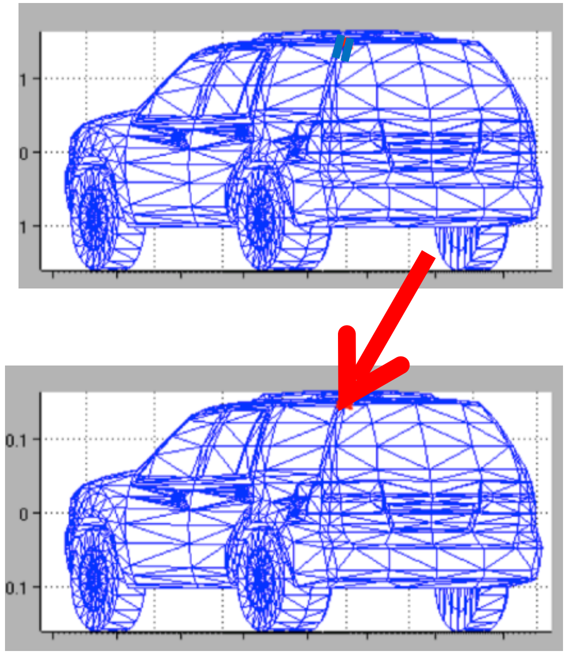
Step1:
Choose similar model





Step2:
Coarse Viewpoint Labeling

Step3:
Label keypoints For alignment



Annotation takes ~1 min per object

High-cost Label Acquisition

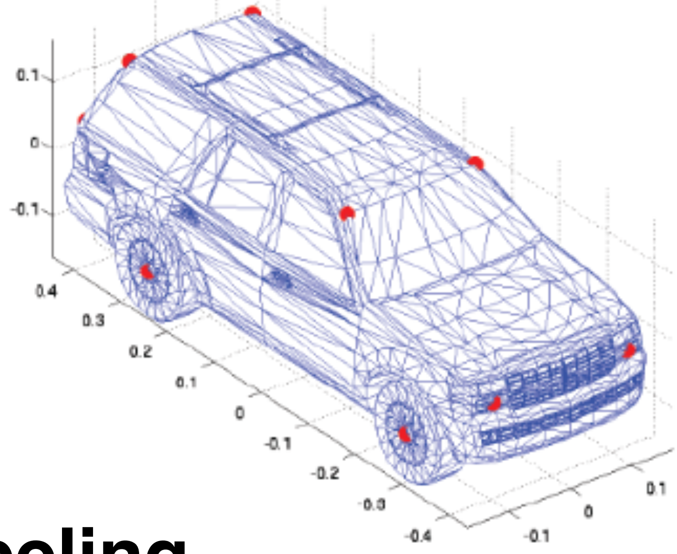
Mismatch!

High-capacity Model

Needs millions of images to train

How to get MORE images with ACCURATE labels?

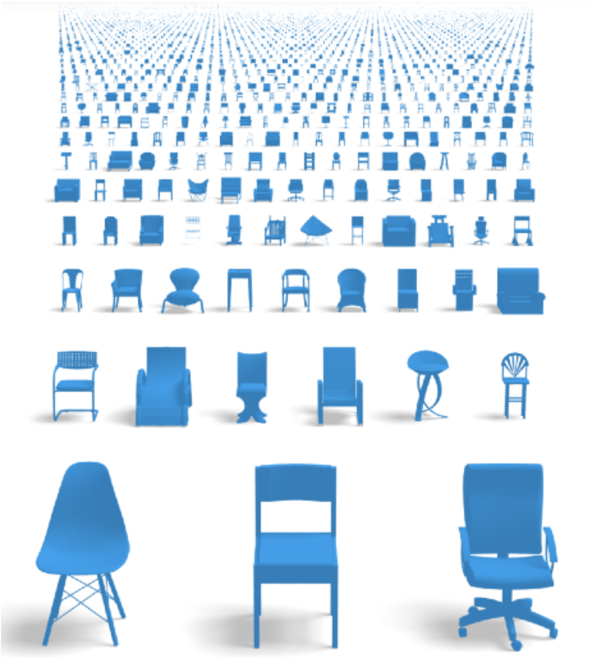
Manual labeling by annotators

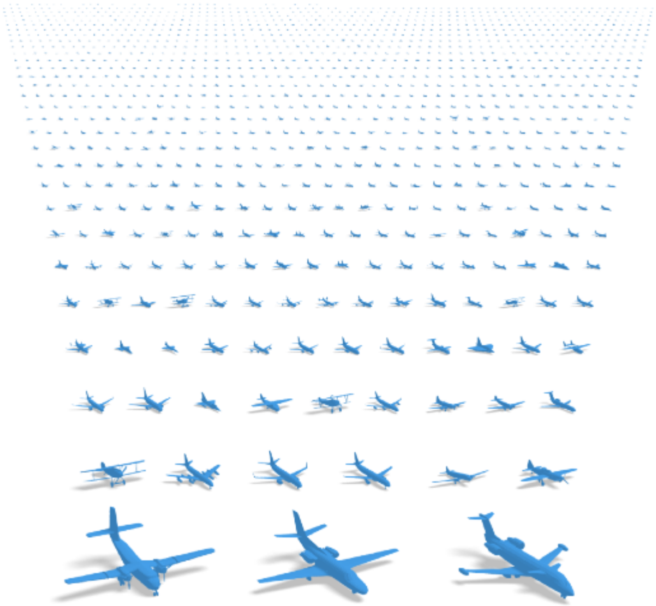


Auto labeling through rendering

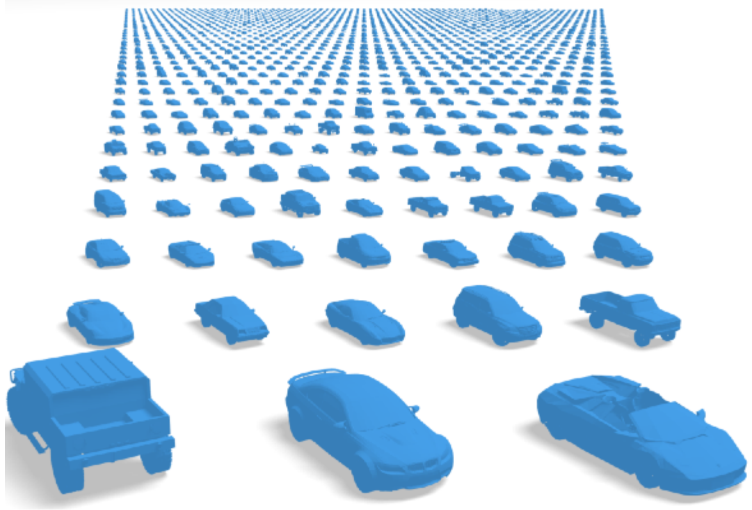
Good News: ShapeNet

http://shapenet.cs.stanford.edu





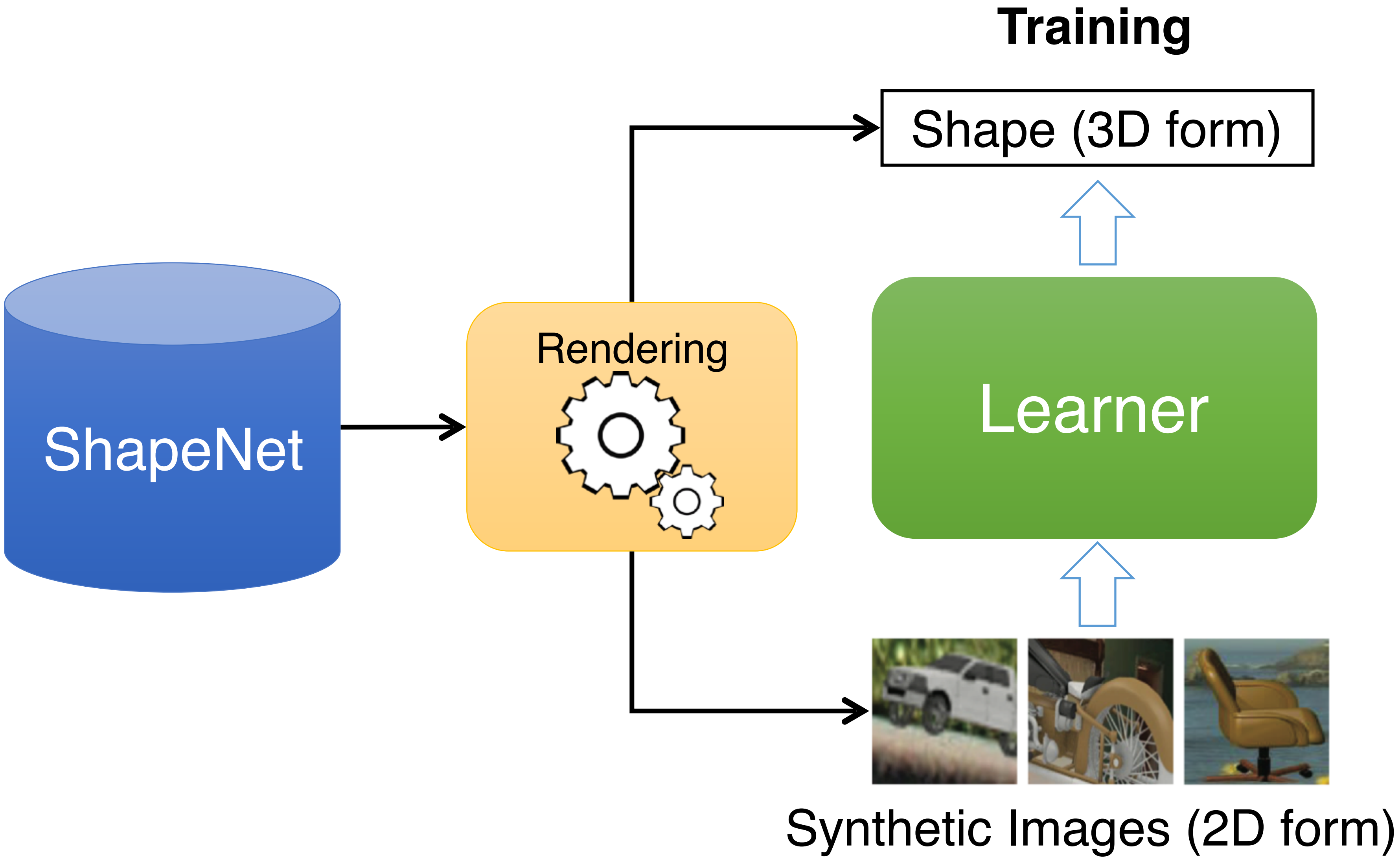
...



millions of shapes with **geometric** and **physical** annotations
(ongoing efforts)

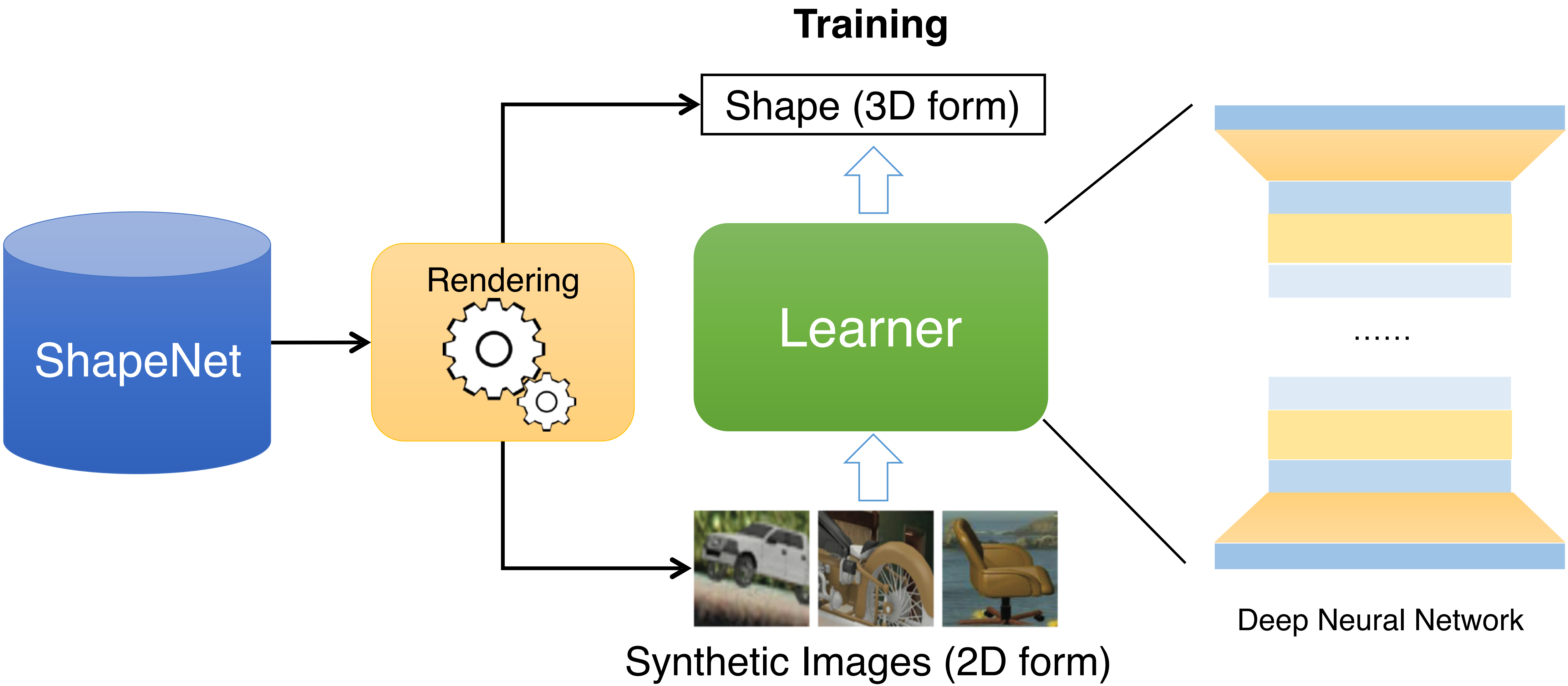
Tech Report ’15, ShapeNet

Synthesize for learning



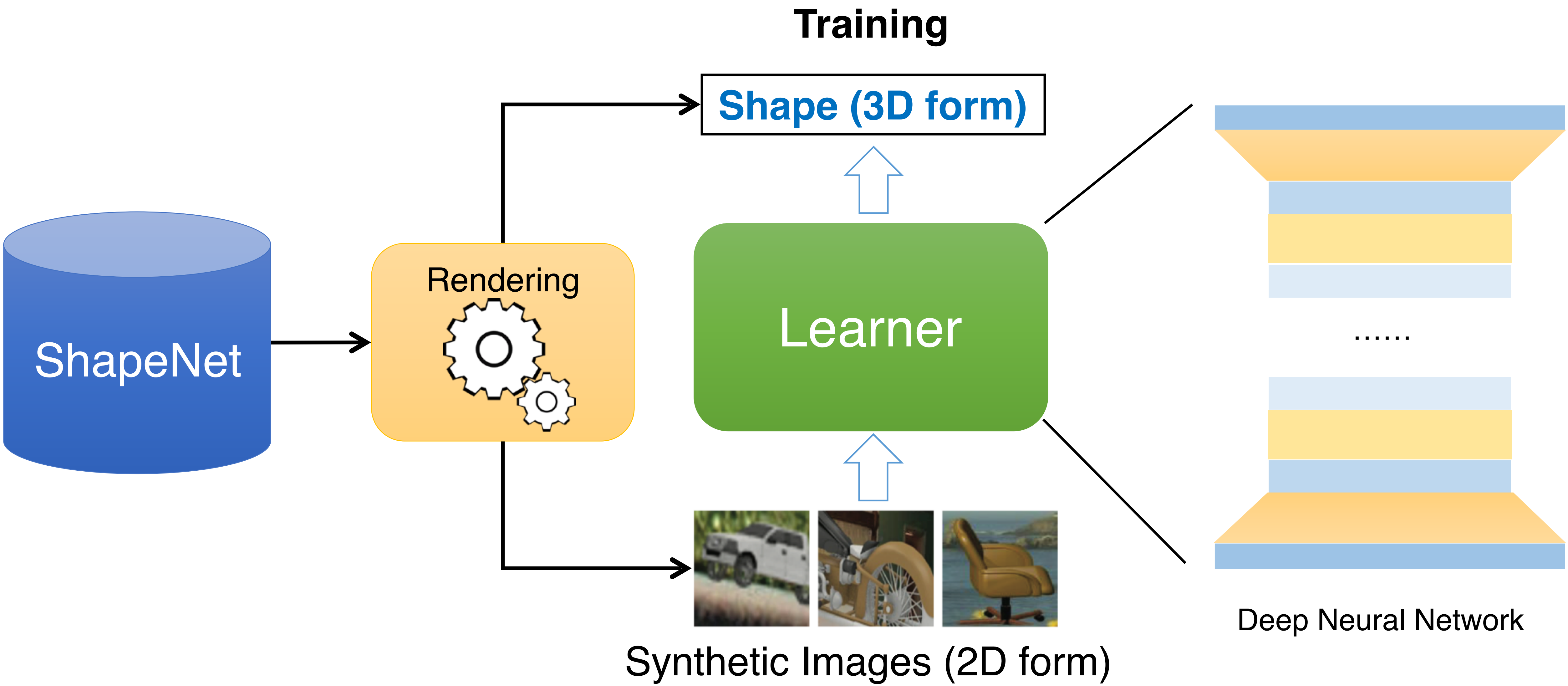
[Su et al., RenderForCNN, ICCV15]

Synthesize for learning



[Su et al., RenderForCNN, ICCV15]

How shall we represent 3D shapes?



[Su et al., RenderForCNN, ICCV15]

Many 3D representations are available

Candidates:

multi-view images

depth map

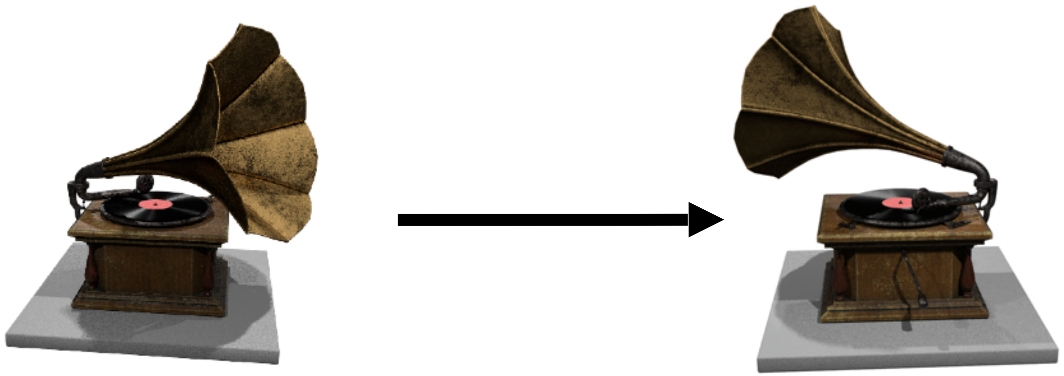
volumetric

polygonal mesh

point cloud

primitive-based CAD models

3D representation



Novel view image synthesis

[Su et al., ICCV15]
[Dosovitskiy et al., ECCV16]

Candidates:

multi-view images

depth map

volumetric

polygonal mesh

point cloud

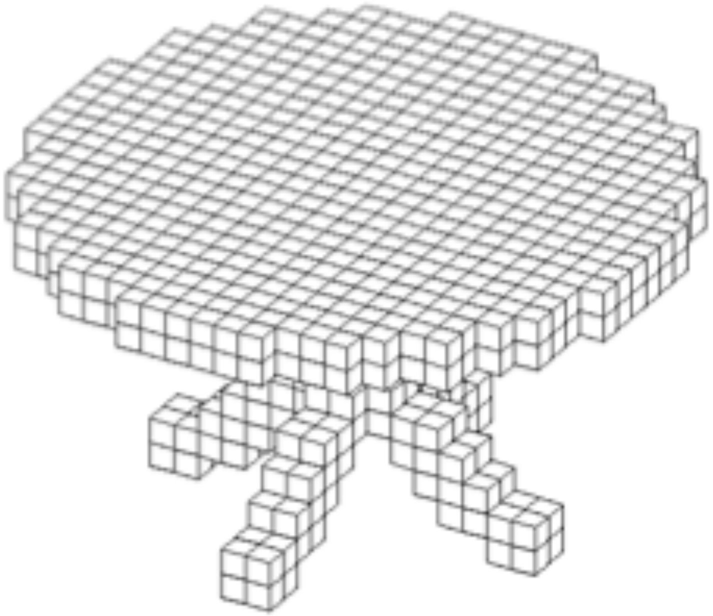
primitive-based CAD models

3D representation

Candidates:

- multi-view images
- **depth map**
- volumetric
- polygonal mesh
- point cloud
- primitive-based CAD models

3D representation



Candidates:

multi-view images

depth map

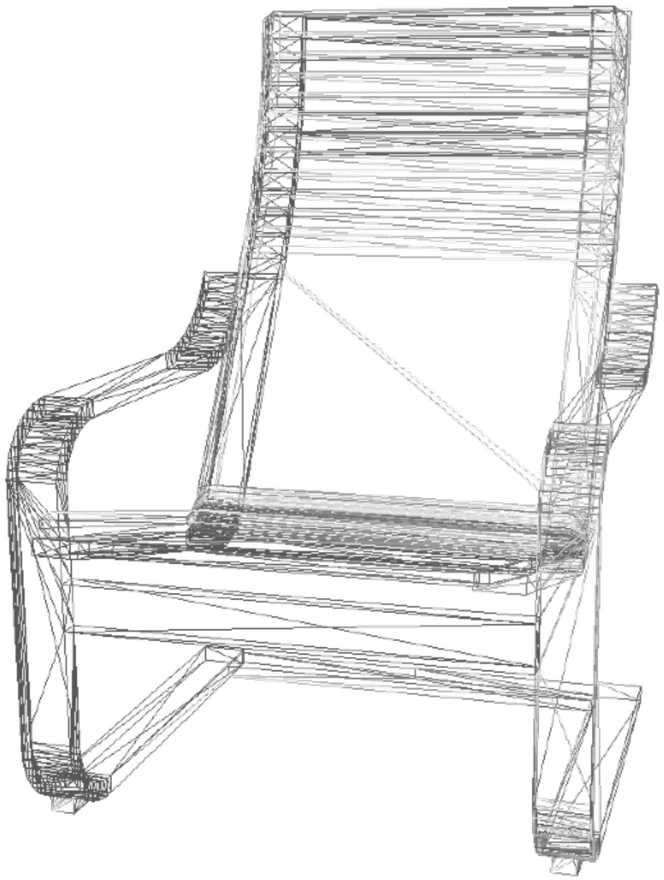
volumetric

polygonal mesh

point cloud

primitive-based CAD models

3D representation



Candidates:

multi-view images

depth map

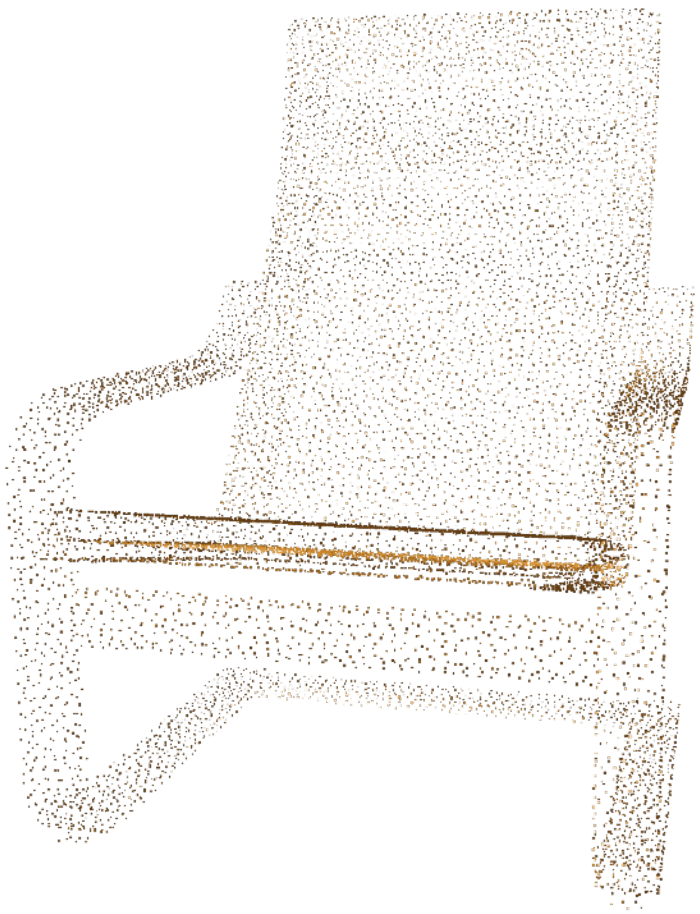
volumetric

polygonal mesh

point cloud

primitive-based CAD models

3D representation



Candidates:

multi-view images

depth map

volumetric

polygonal mesh

point cloud

primitive-based CAD models

3D representation

a chair assembled by cuboids

Candidates:

multi-view images

depth map

volumetric

polygonal mesh

point cloud

primitive-based CAD models

Two groups of representations

Candidates:

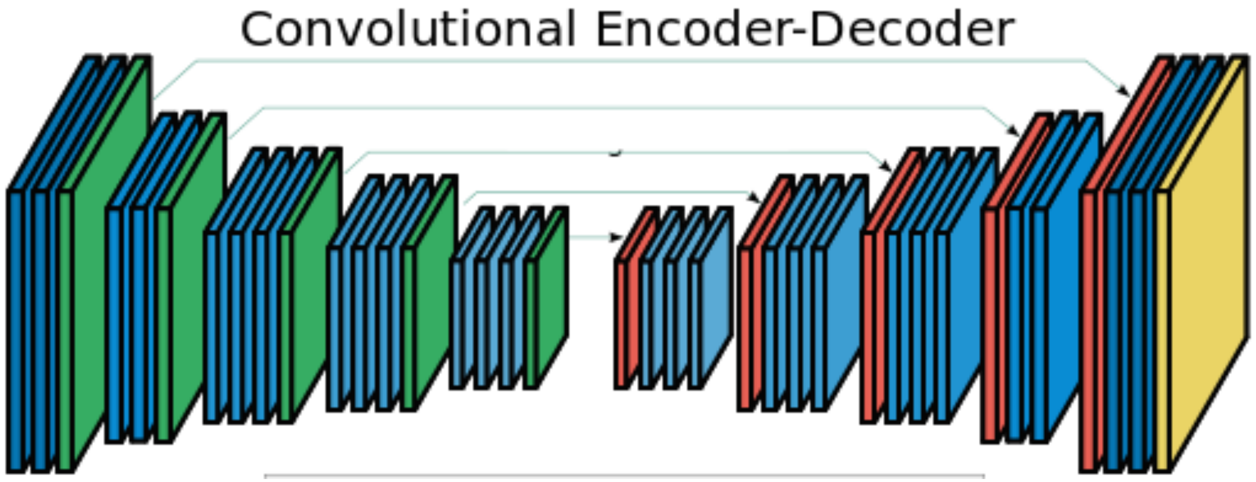
Rasterized form (regular grids)

- multi-view images
- depth map
- volumetric

Geometric form (irregular)

- polygonal mesh
- point cloud
- primitive-based CAD models

Extant 3D DNNs work on grid-like representations



Candidates:

multi-view images

depth map

volumetric

polygonal mesh

point cloud

primitive-based CAD models

Ideally, a 3D representation should be

Friendly to learning

- easily formulated as the output of a neural network
- fast forward-/backward- propagation
- etc.

Ideally, a 3D representation should be

Friendly to learning

- easily formulated as the output of a neural network
- fast forward-/backward- propagation
- etc.

Flexible

- can precisely model a great variety of shapes
- etc.

Ideally, a 3D representation should be

Friendly to learning

- easily formulated as the output of a neural network
- fast forward-/backward- propagation
- etc.

Flexible

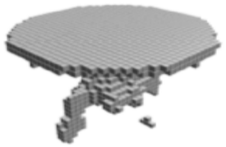
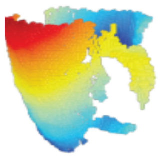
- can precisely model a great variety of shapes
- etc.

Geometrically manipulable

- geometrically deformable, interpolable and extrapolable for networks
- convenient to impose structural constraints
- etc.

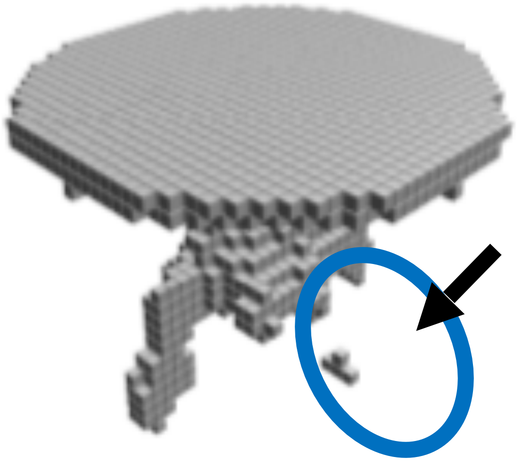
Others

The problem of grid representations

	Affability to learning	Flexibility	Geometric manipulability
Multi-view images 	✓	✓	✗
Volumetric occupancy 	✗ Expensive to compute: $O(N^3)$	✓	✗
Depth map 	✓	✗ Cannot model “back side”	✓

Typical artifacts of volumetric reconstruction

	Affability		Geometric anipulability
Multi-view images			✗
Volumetric occupancy			✗
Depth map			✓



Missing thin structures
due to
improper shape space structure

hard for the network to rotate / deform / interpolate

How about learning to predict geometric forms?

Candidates:

Rasterized form (regular grids)

- multi-view images
- depth map
- volumetric

Geometric form (irregular)

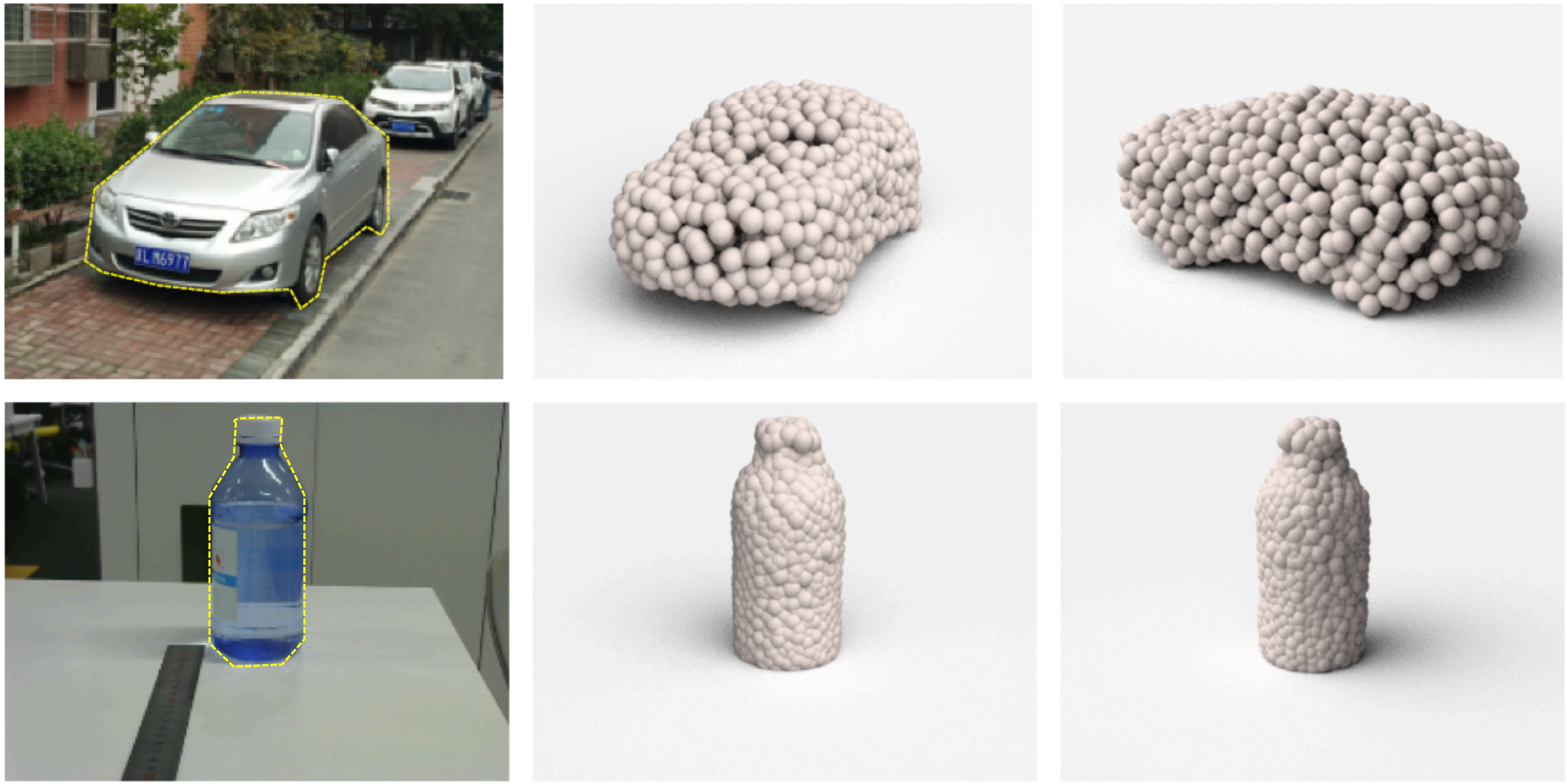
- polygonal mesh
- ★ **point cloud**
- ★ **primitive-based CAD models**

Outline

- Motivation
- Data
- **Algorithms for 2D-3D lifting**
- Shape abstraction by volumetric primitives

Our result: 3D reconstruction from real Images

CVPR 2017, A Point Set Generation Network for 3D Object Reconstruction from a Single Image

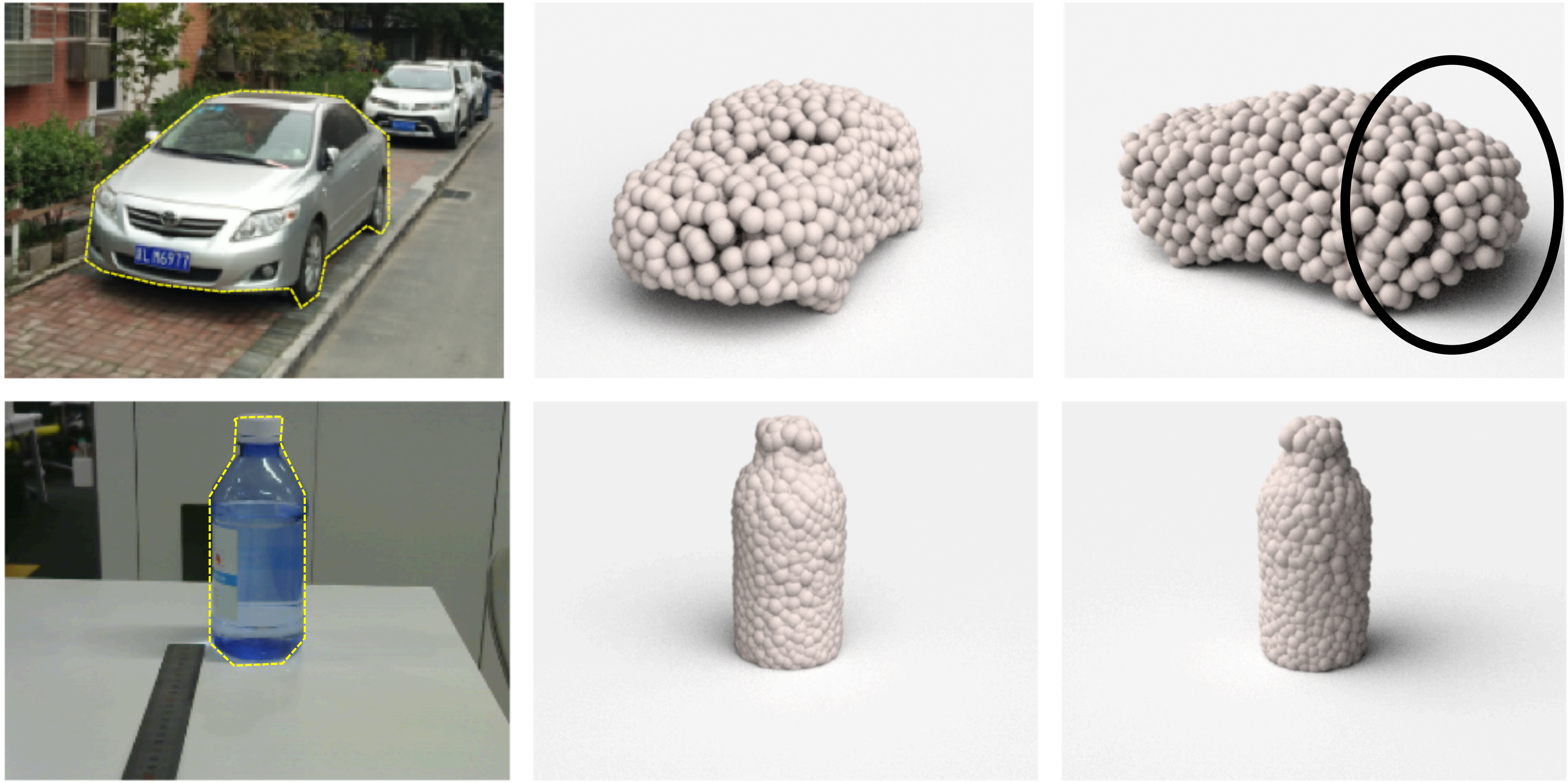


Input

Reconstructed 3D point cloud

Our result: 3D reconstruction from real Images

CVPR 2017, A Point Set Generation Network for 3D Object Reconstruction from a Single Image

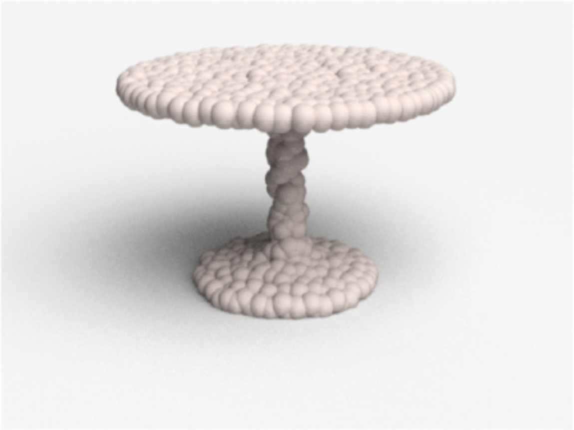


Input　　　　Reconstructed 3D point cloud

3D point clouds

✓ Flexible

- a few thousands of points can precisely model a great variety of shapes



CVPR '17, Point Set Generation

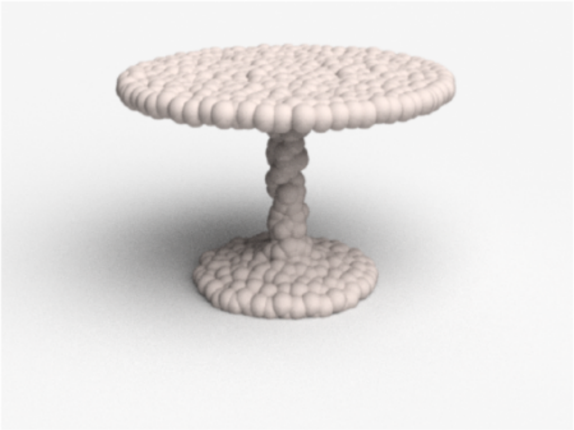
3D point clouds

✓ Flexible

- a few thousands of points can precisely model a great variety of shapes

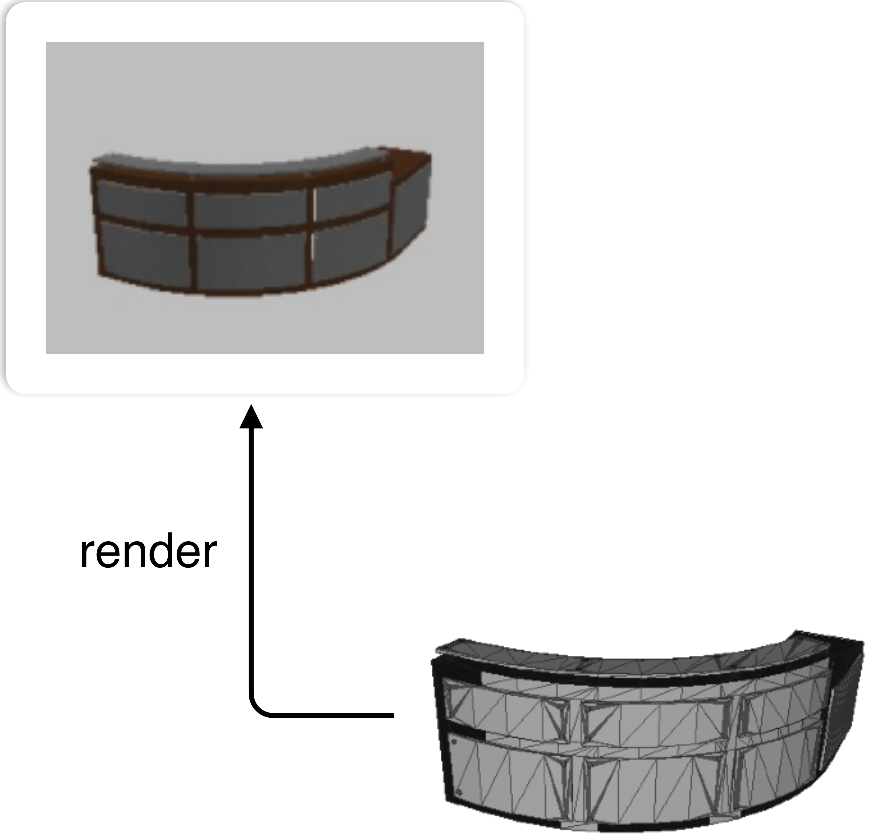
✓ Geometrically manipulable

- deformable
- interpolable, extrapolable
- convenient to impose structural constraints



CVPR ’17, Point Set Generation

Pipeline



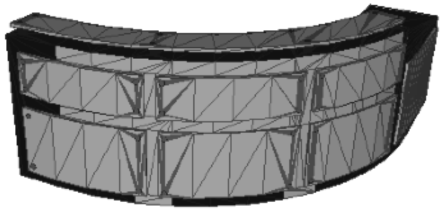
Pipeline

2K object categories

200K shapes

~10M image/point set pairs

render

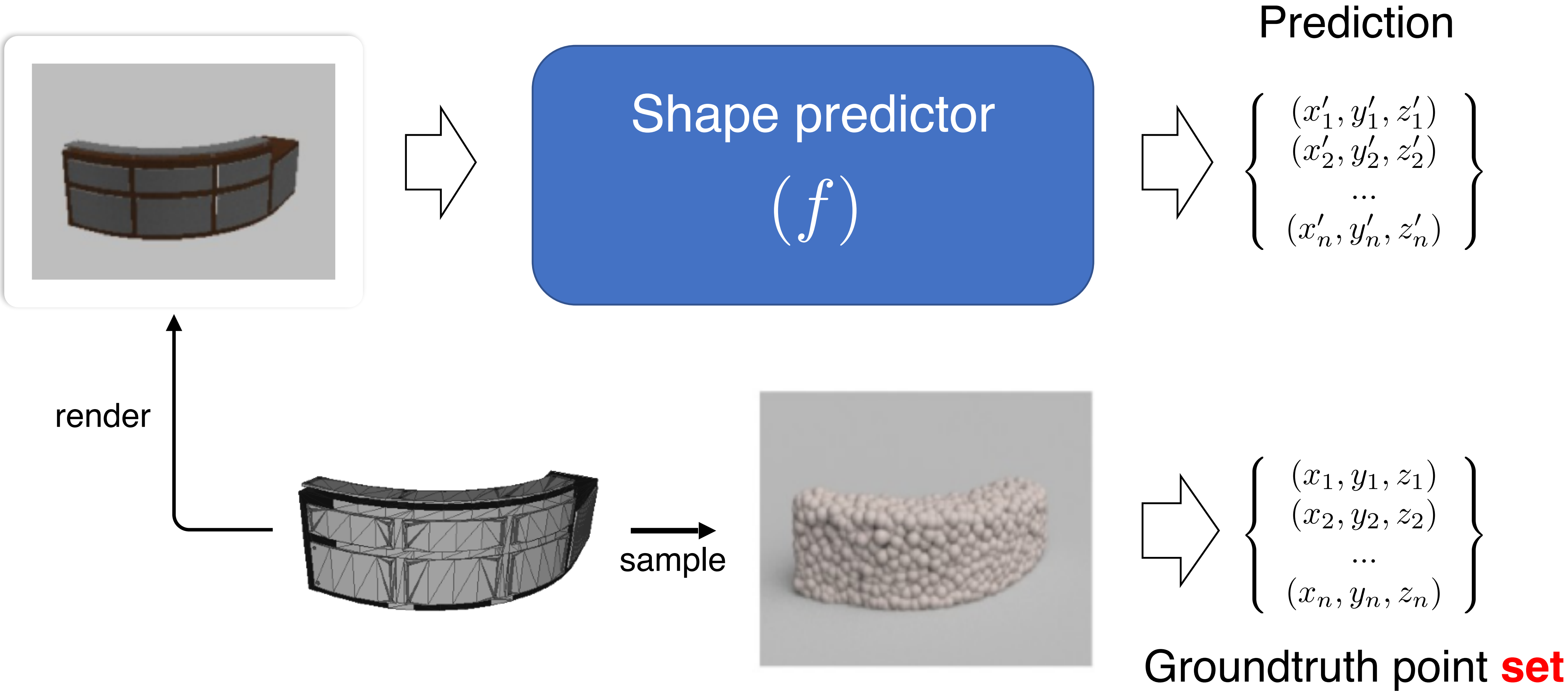


sample

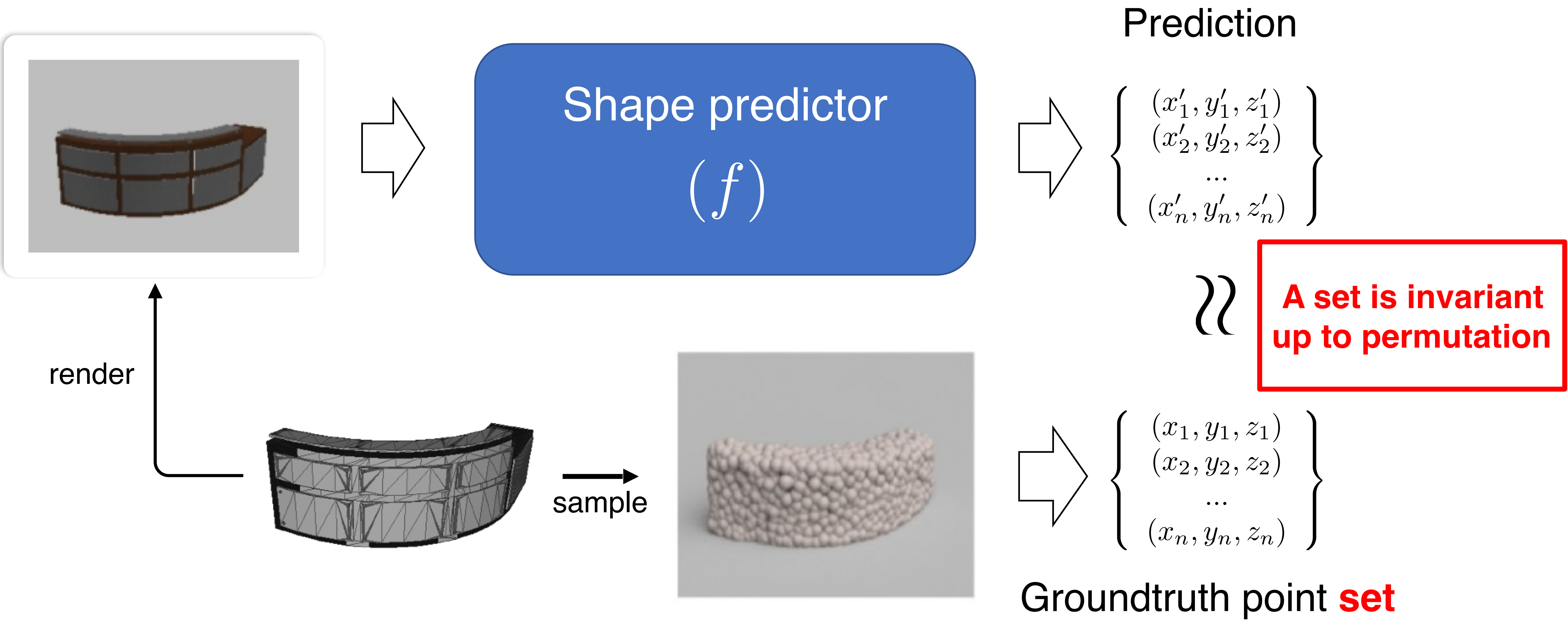
$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{array} \right\}$$

Groundtruth point **set**

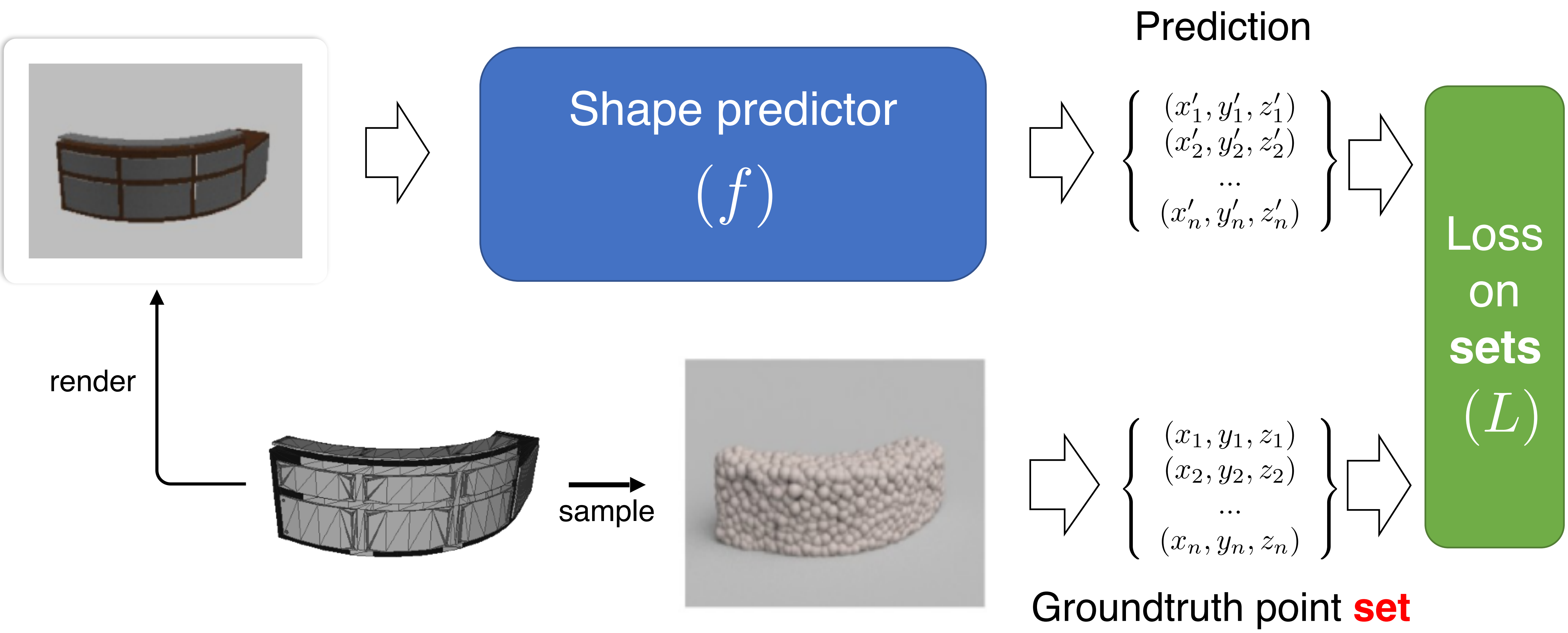
Pipeline


Prediction
Shape predictor
(f)
$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$
render
sample
$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{array} \right\}$
Groundtruth point **set**

Pipeline

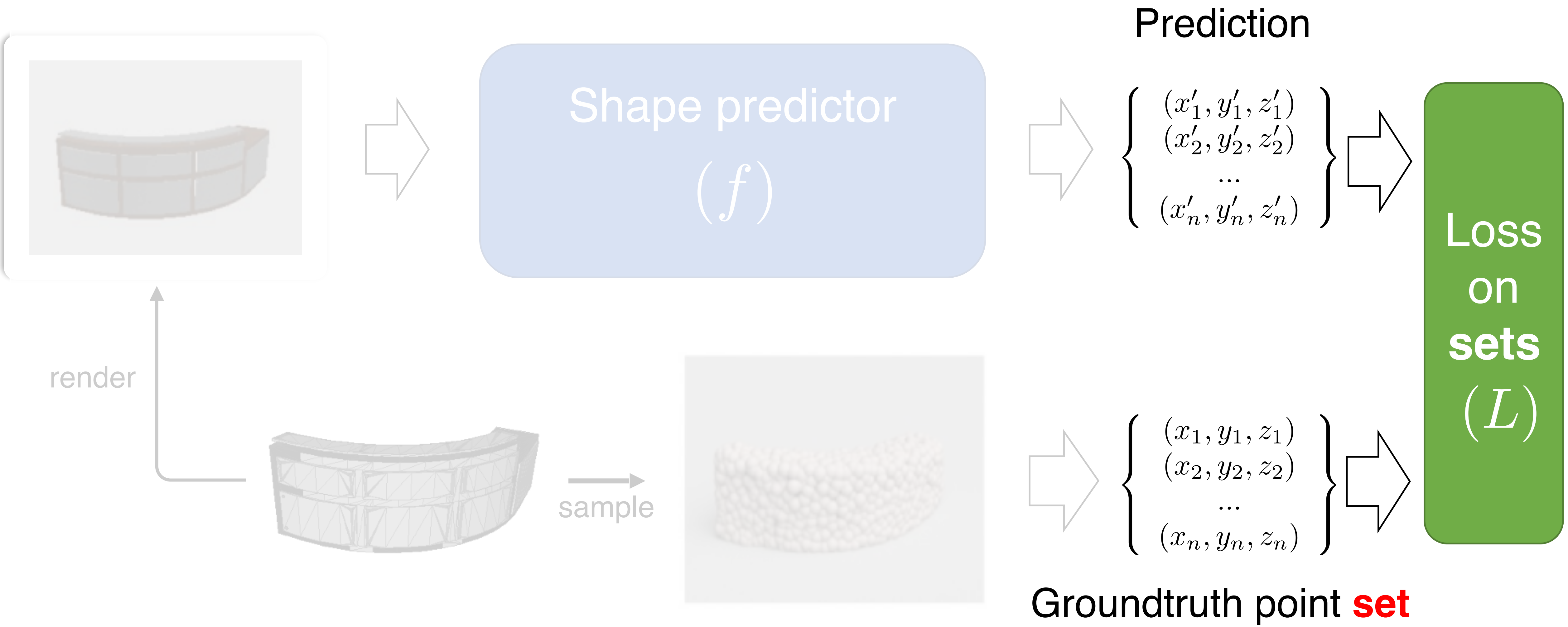

Prediction
Shape predictor
(f)
$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$
$\wr\wr$
A set is invariant up to permutation
render
sample
$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{array} \right\}$
Groundtruth point set

Pipeline



Prediction
Shape predictor
(f)
$\left\{ \begin{matrix} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{matrix} \right\}$
Loss on **sets**
(L)
render
sample
$\left\{ \begin{matrix} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{matrix} \right\}$
Groundtruth point **set**

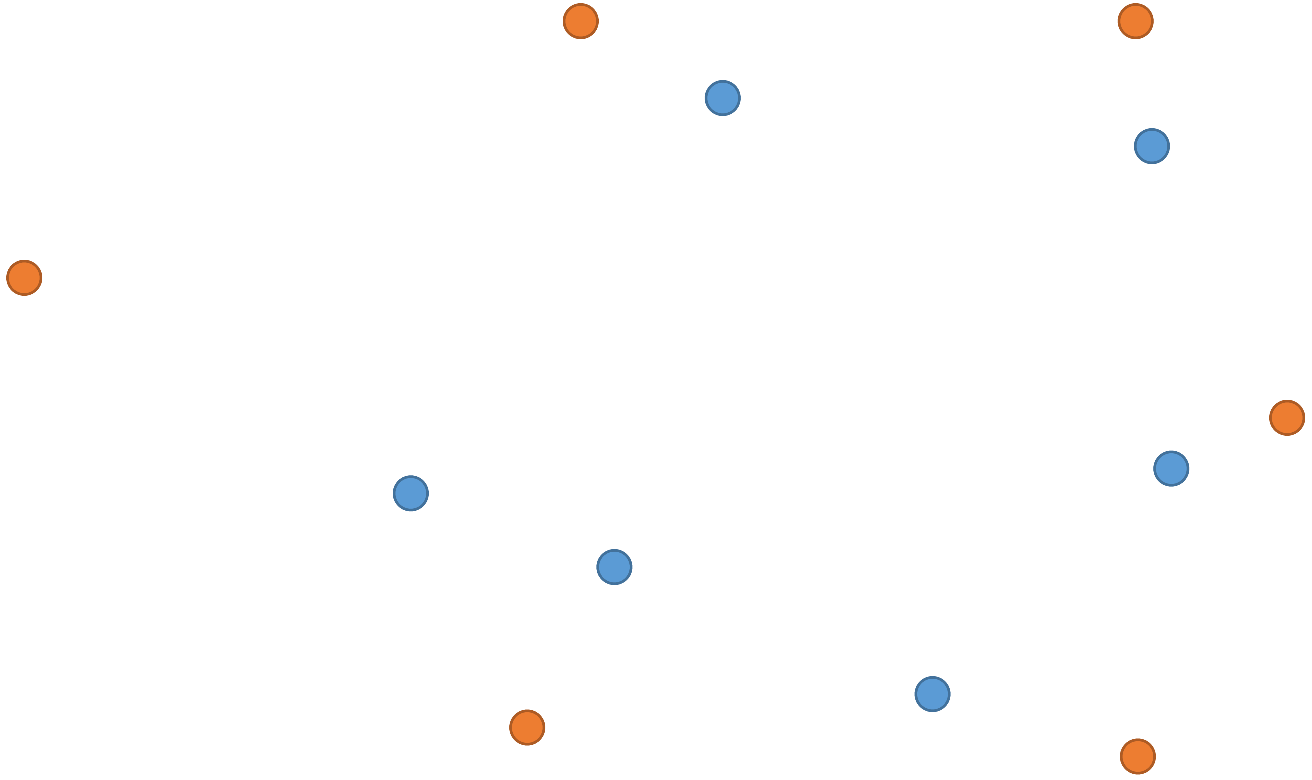
Pipeline



Prediction
Shape predictor
(f)
$\left\{ \begin{matrix} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{matrix} \right\}$
Loss on **sets** (L)
render
sample
$\left\{ \begin{matrix} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{matrix} \right\}$
Groundtruth point **set**

Set comparison

Given two sets of points, measure their discrepancy



Set comparison

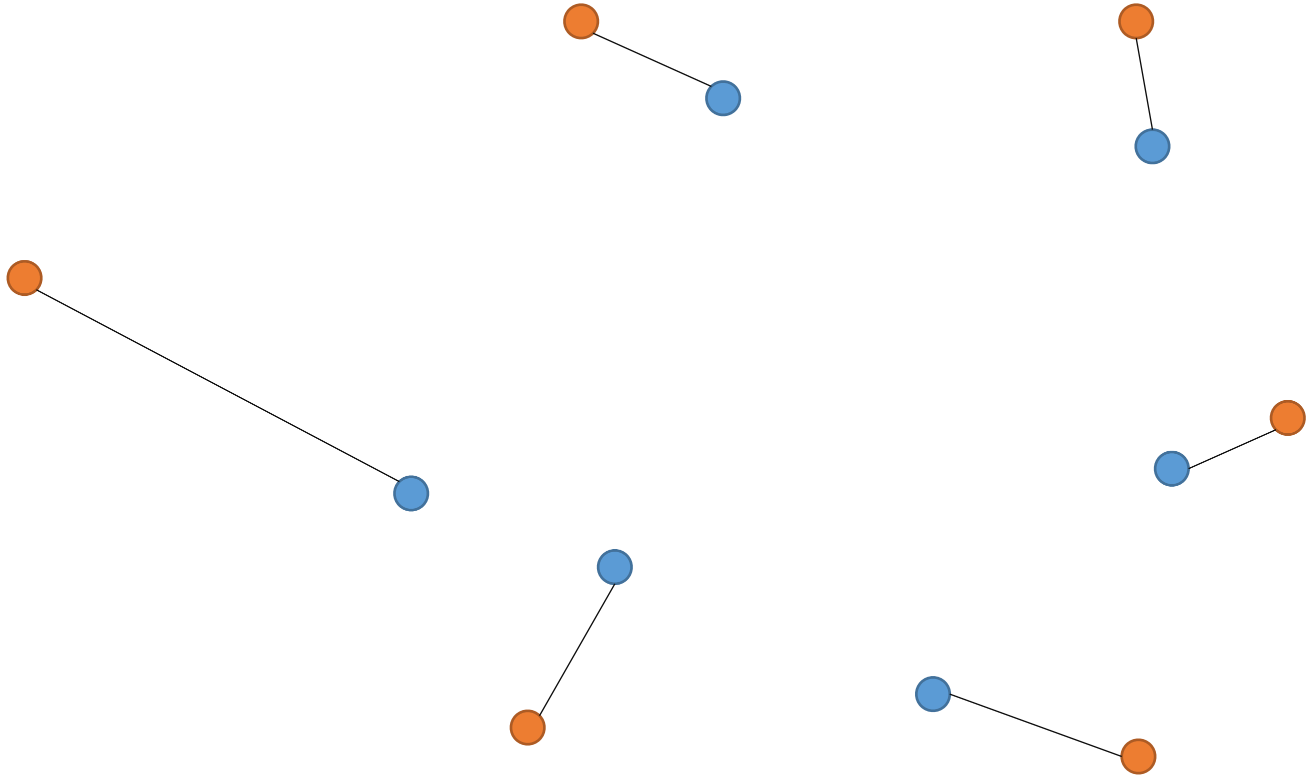
Given two sets of points, measure their discrepancy

Key challenge:

correspondence problem

Correspondence (I): optimal assignment

Given two sets of points, measure their discrepancy



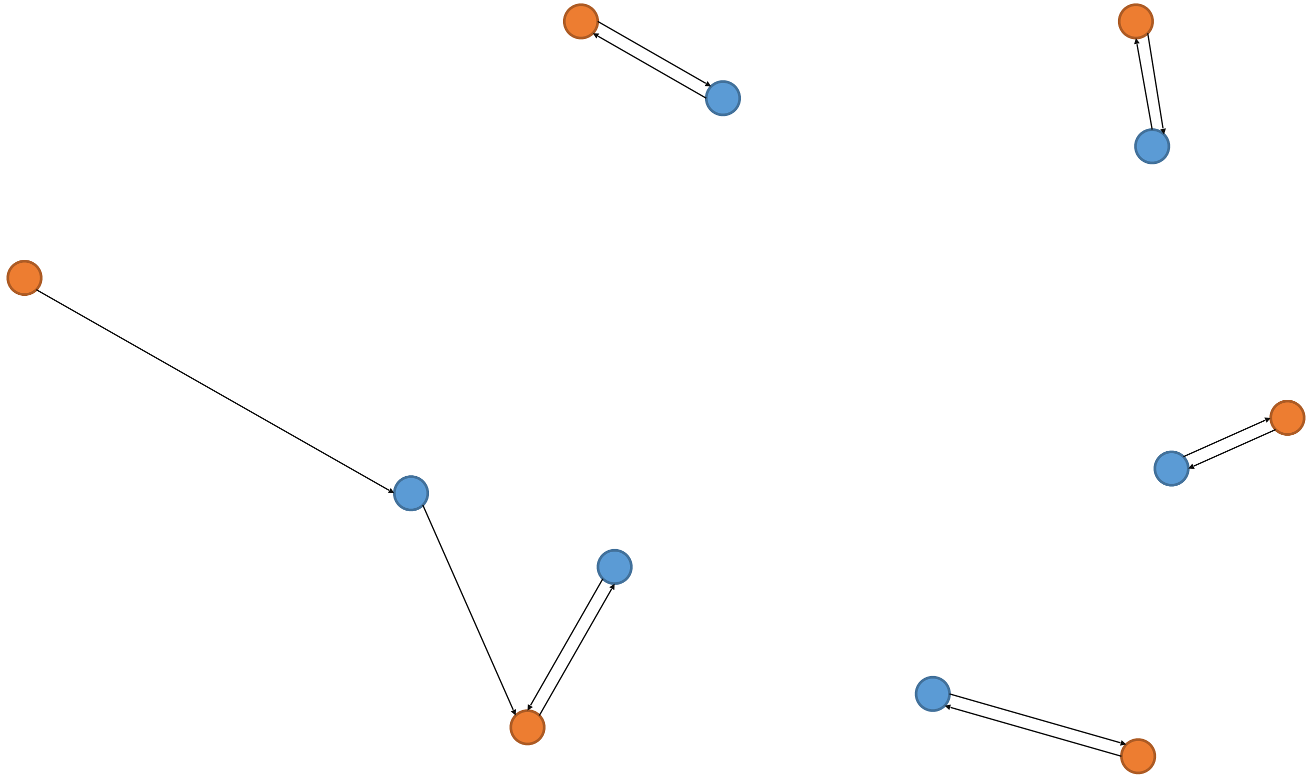
a.k.a Earth Mover's distance (EMD)

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2 \quad \text{where } \phi : S_1 \to S_2 \text{ is a bijection.}$$

CVPR '17, Point Set Generation

Correspondence (II): closest point

Given two sets of points, measure their discrepancy



a.k.a Chamfer distance (CD)

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

CVPR '17, Point Set Generation

Required properties of distance metrics

Geometric requirement

Computational requirement

Required properties of distance metrics

Geometric requirement

- Reflects natural shape differences
- Induce a nice space for ***shape interpolations***

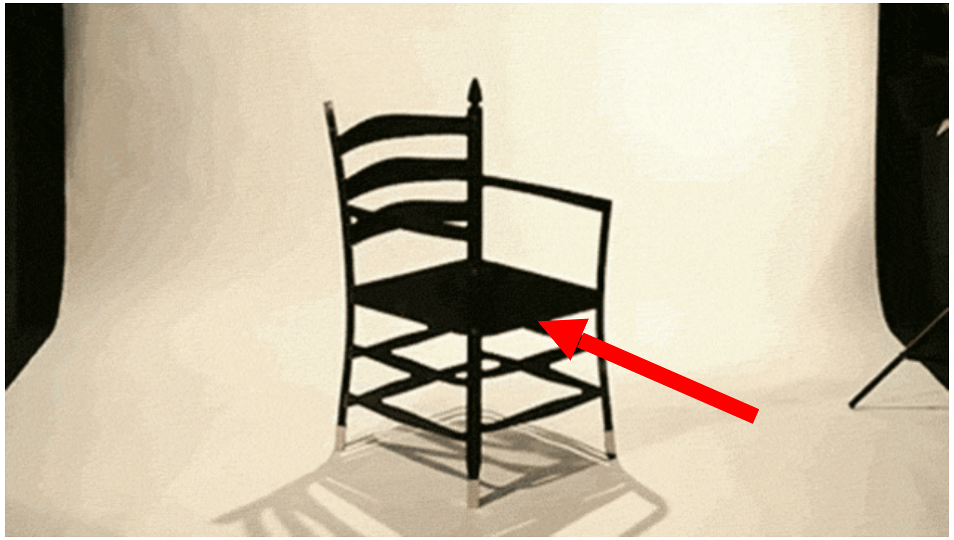
Computational requirement

How distance metric affects learning?

A fundamental issue: inherent ambiguity in 2D-3D dimension lifting

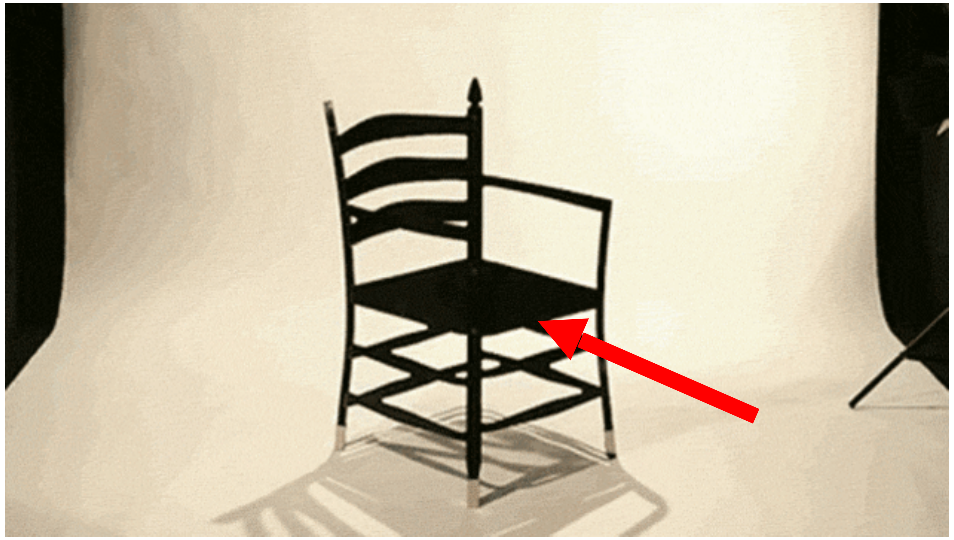
How distance metric affects learning?

A fundamental issue: inherent ambiguity in 2D-3D dimension lifting



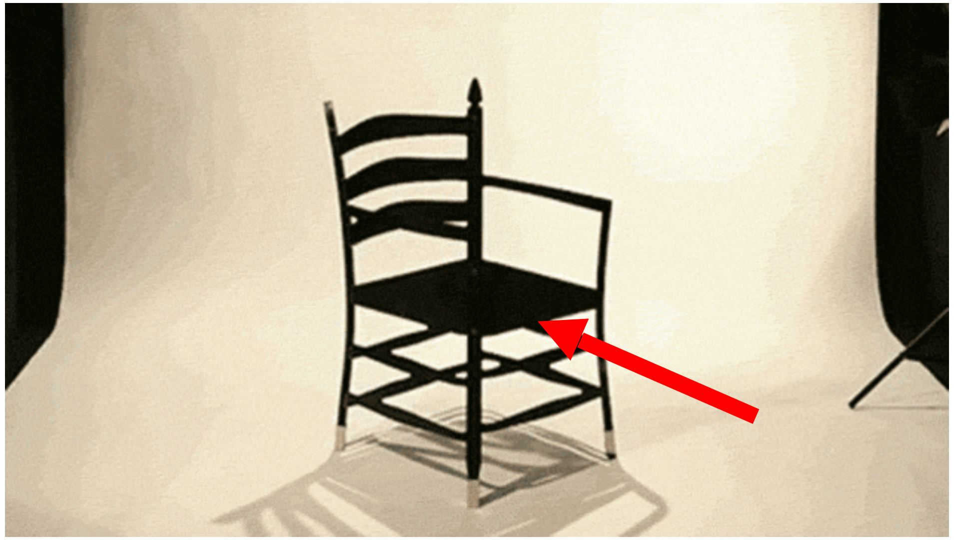
How distance metric affects learning?

A fundamental issue: inherent ambiguity in 2D-3D dimension lifting



How distance metric affects learning?

A fundamental issue: inherent ambiguity in 2D-3D dimension lifting



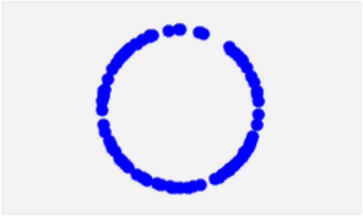
- By loss minimization, the network tends to predict a “**mean shape**” that **averages out** uncertainty

Distance metrics affect mean shapes

The mean shape carries characteristics of the distance metric

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

continuous
hidden variable
(radius)



Input

EMD mean

Chamfer mean

CVPR '17, Point Set Generation

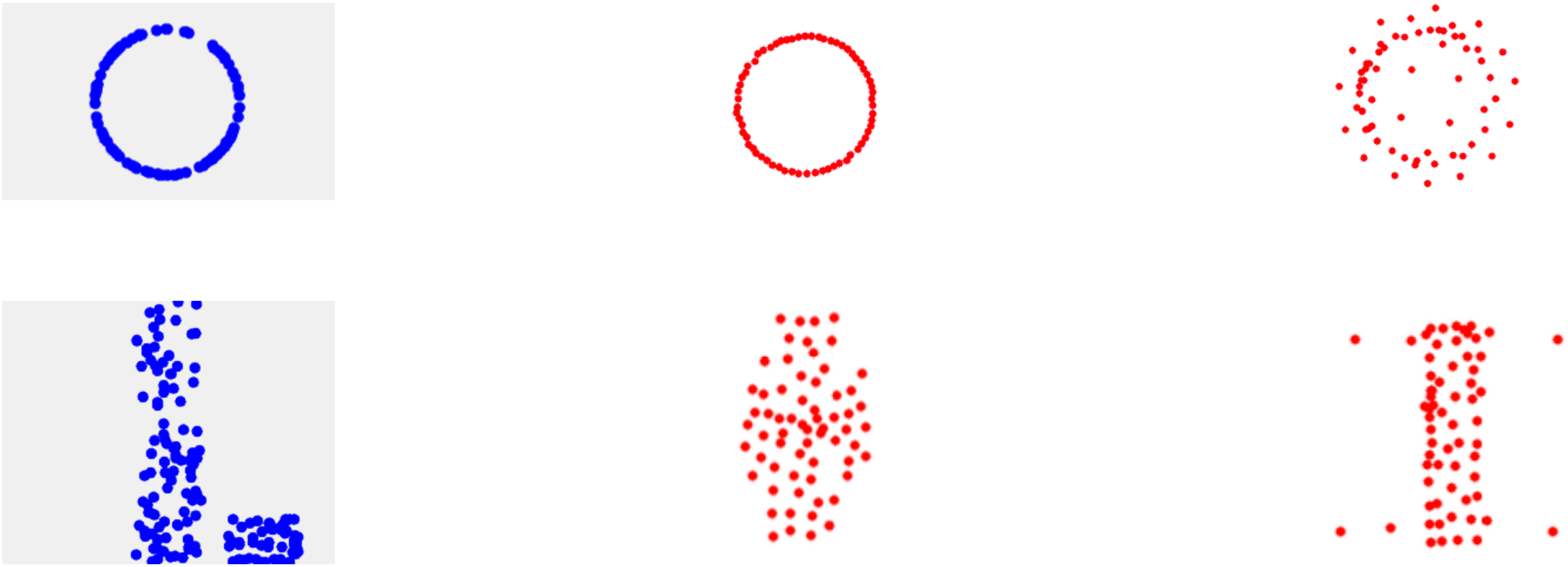
Mean shapes from distance metrics

The mean shape carries characteristics of the distance metric

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

continuous
hidden variable
(radius)

discrete
hidden variable
(add-on location)



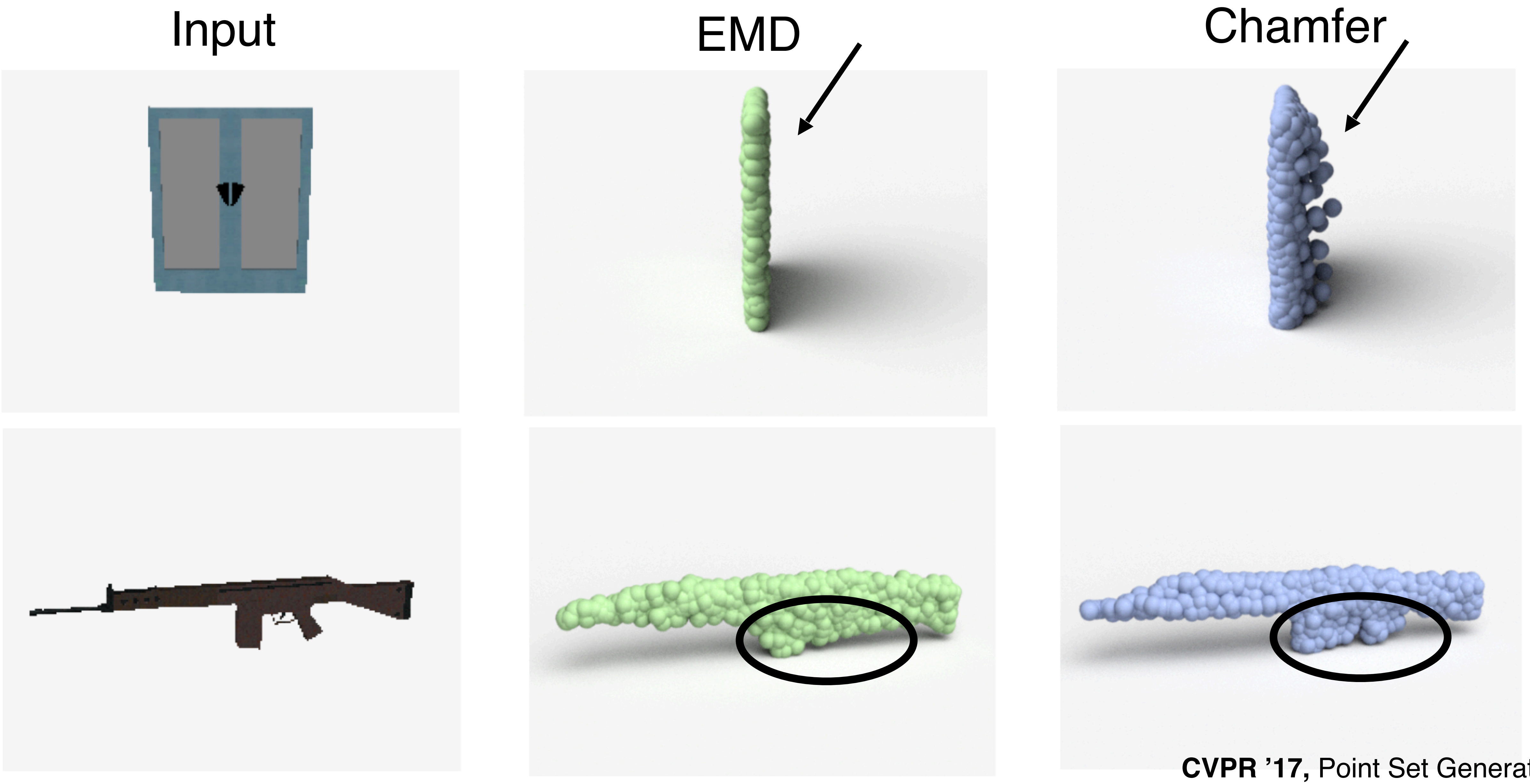
Input

EMD mean

Chamfer mean

CVPR '17, Point Set Generation

Comparison of predictions by EMD versus CD



CVPR '17, Point Set Generation

Required properties of distance metrics

Geometric requirement

- Reflects natural shape differences
- Induce a nice space for shape interpolations

Computational requirement

- Defines a loss function that is numerically easy to optimize

Computational requirement of metrics

To be used as a loss function, the metric has to be

- **Differentiable** with respect to point locations
- **Efficient** to compute

Computational requirement of metrics

- **Differentiable** with respect to point location

Chamfer distance

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

Earth Mover's distance

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2 \quad \text{where } \phi : S_1 \to S_2 \text{ is a bijection.}$$

- Simple function of coordinates
- In general positions, the correspondence is unique
- **With infinitesimal movement, the correspondence does not change**

Conclusion: differentiable almost everywhere

Computational requirement of metrics

- **Differentiable** with respect to point location

- For many **algorithms** (sorting, shortest path, network flow, …),
- an infinitesimal change to model parameters (almost) does not change solution structure,

leads to **differentiable a.e.!**

Computational requirement of metrics

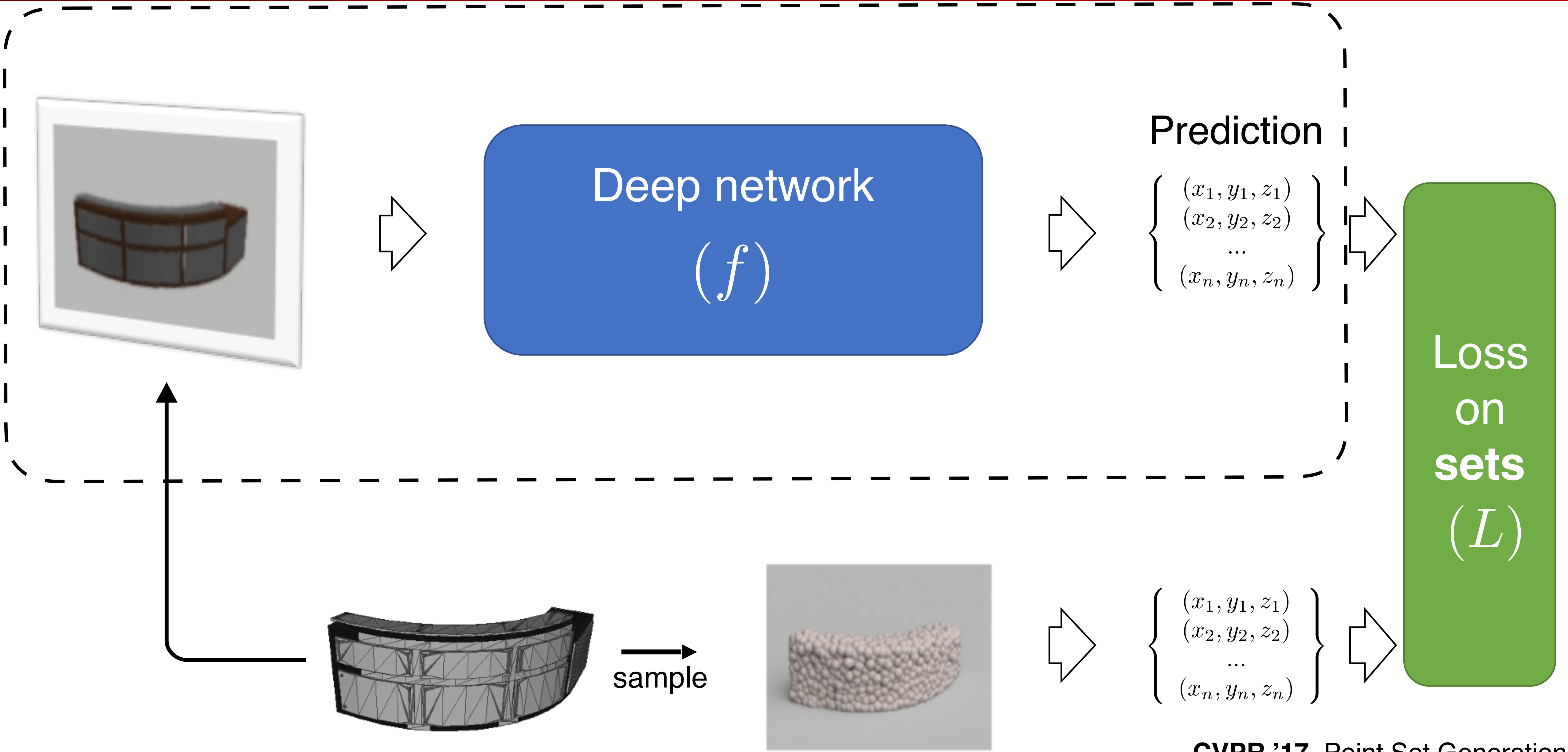
- **Efficient** to compute

  Chamfer distance: trivially parallelizable on CUDA

  Earth Mover's distance (optimal assignment):

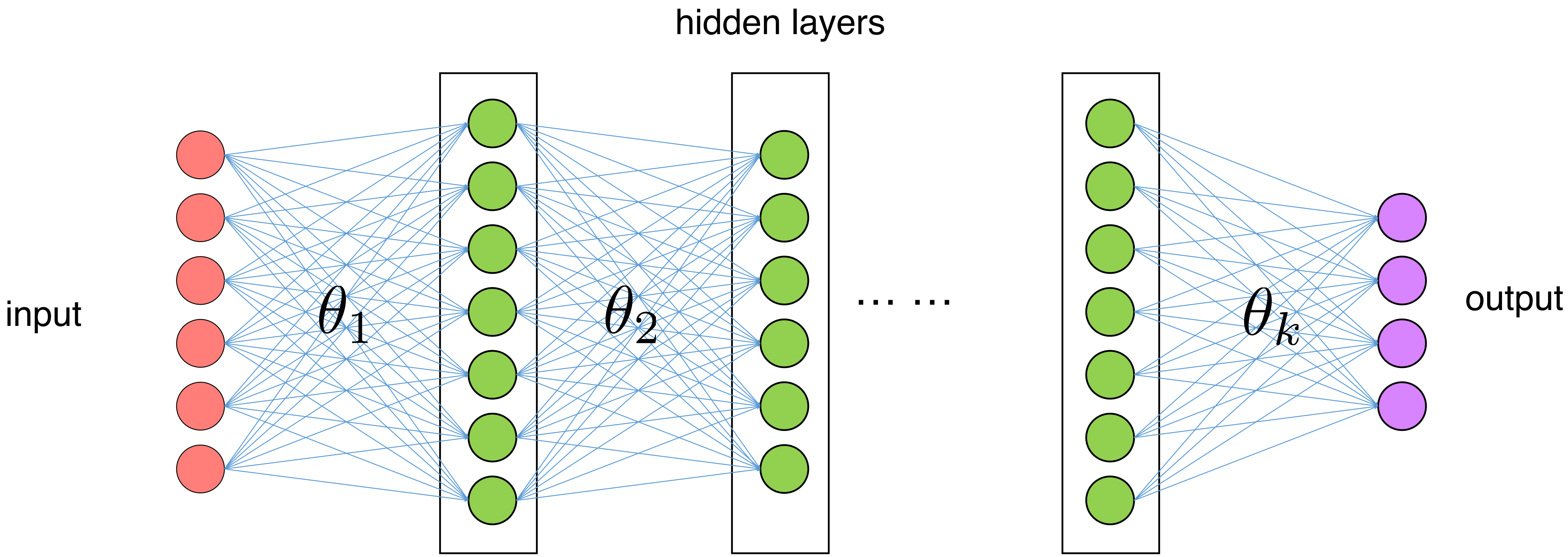
  - We implement a **distributed** approximation algorithm on CUDA
  - Based upon [Bertsekas, 1985], $(1+\epsilon)$-approximation

Pipeline



CVPR '17, Point Set Generation

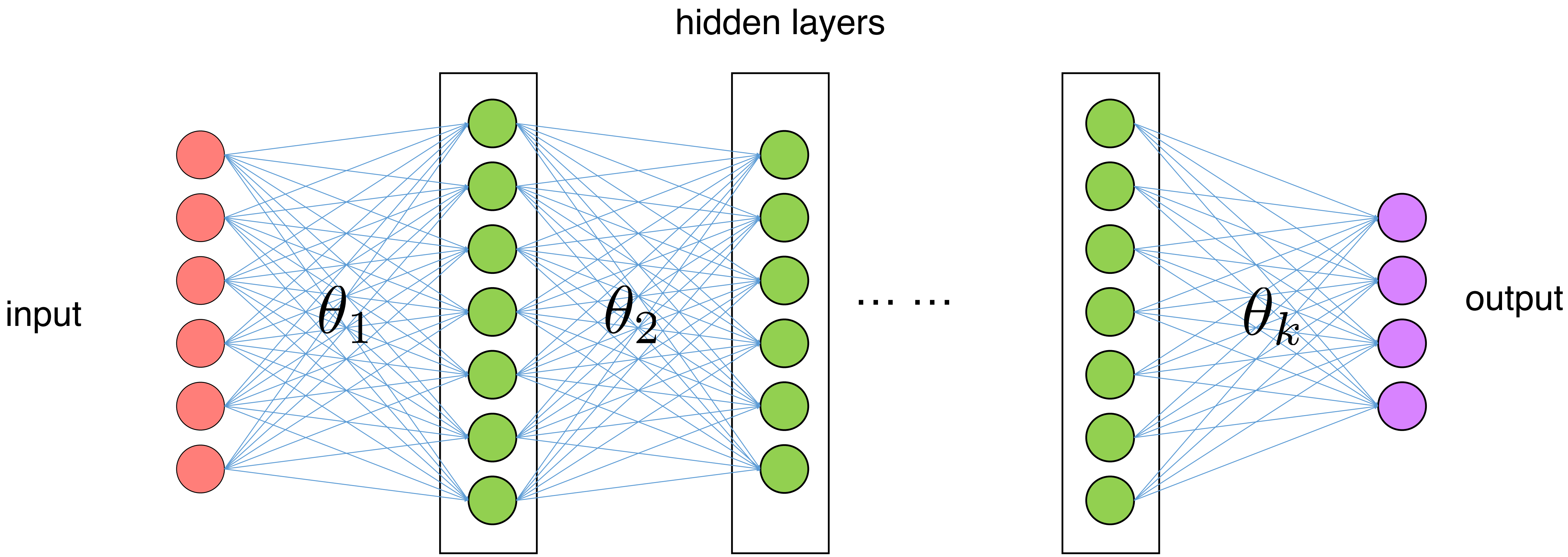
Deep neural network



Universal function approximator

- A cascade of layers

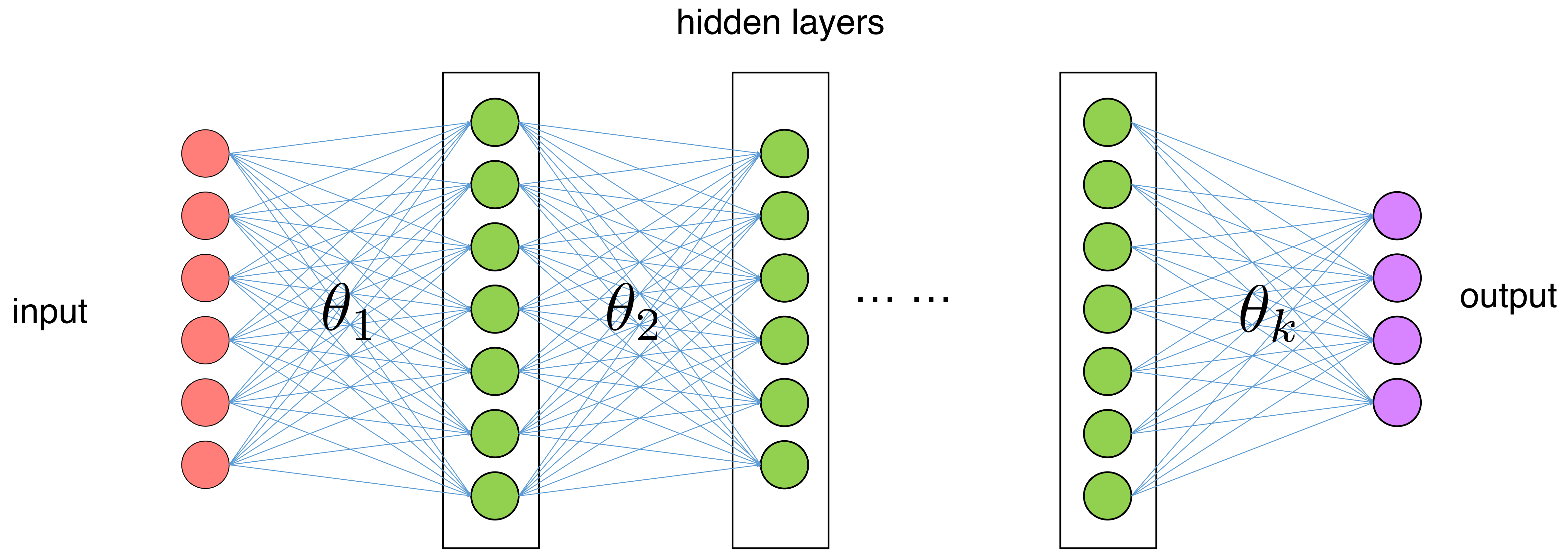
Deep neural network



Universal function approximator

- A cascade of layers
- Each layer conducts a simple transformation (parameterized)

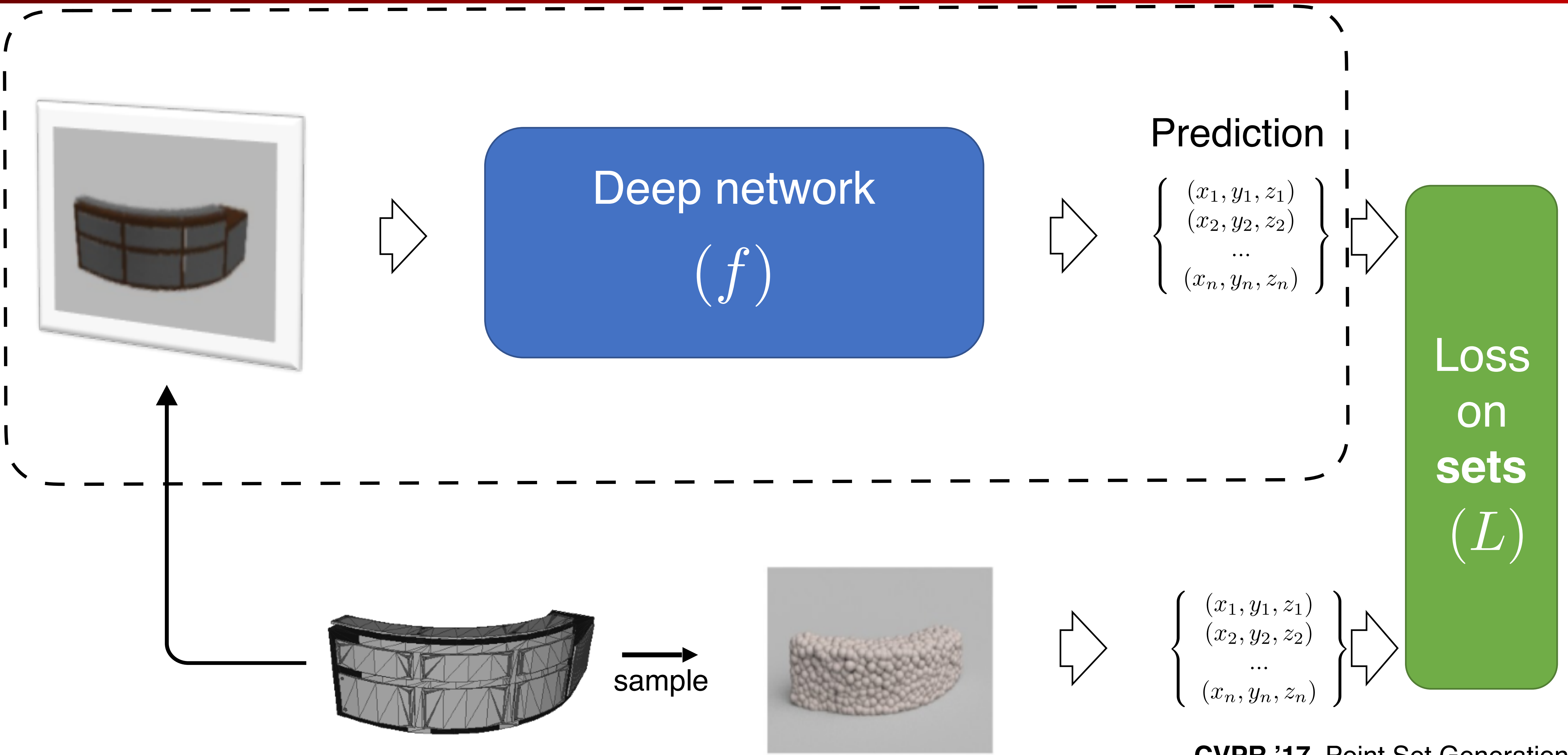
Deep neural network



Universal function approximator

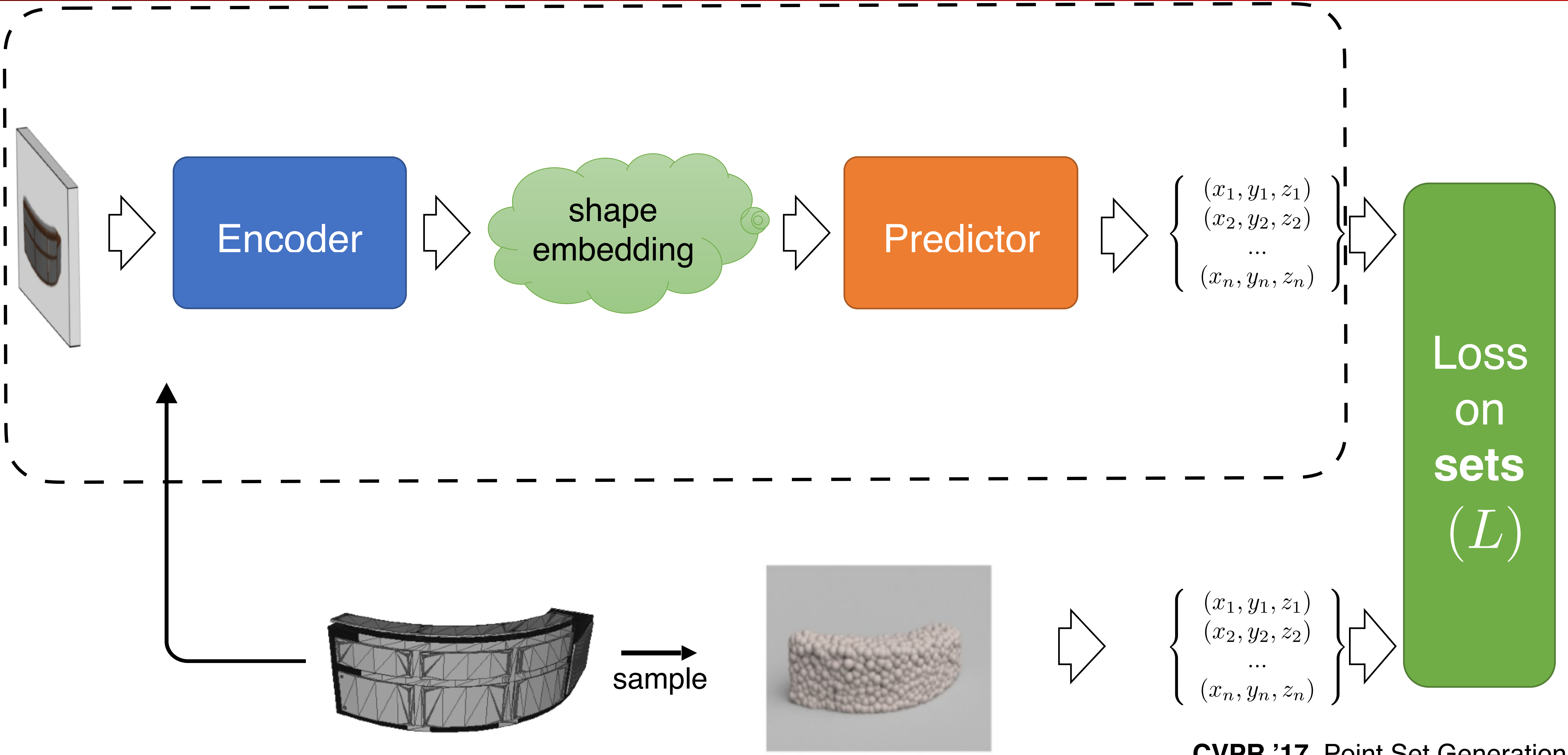
- A cascade of layers
- Each layer conducts a simple transformation (parameterized)
- Millions of parameters, has to be fitted by many data

Pipeline



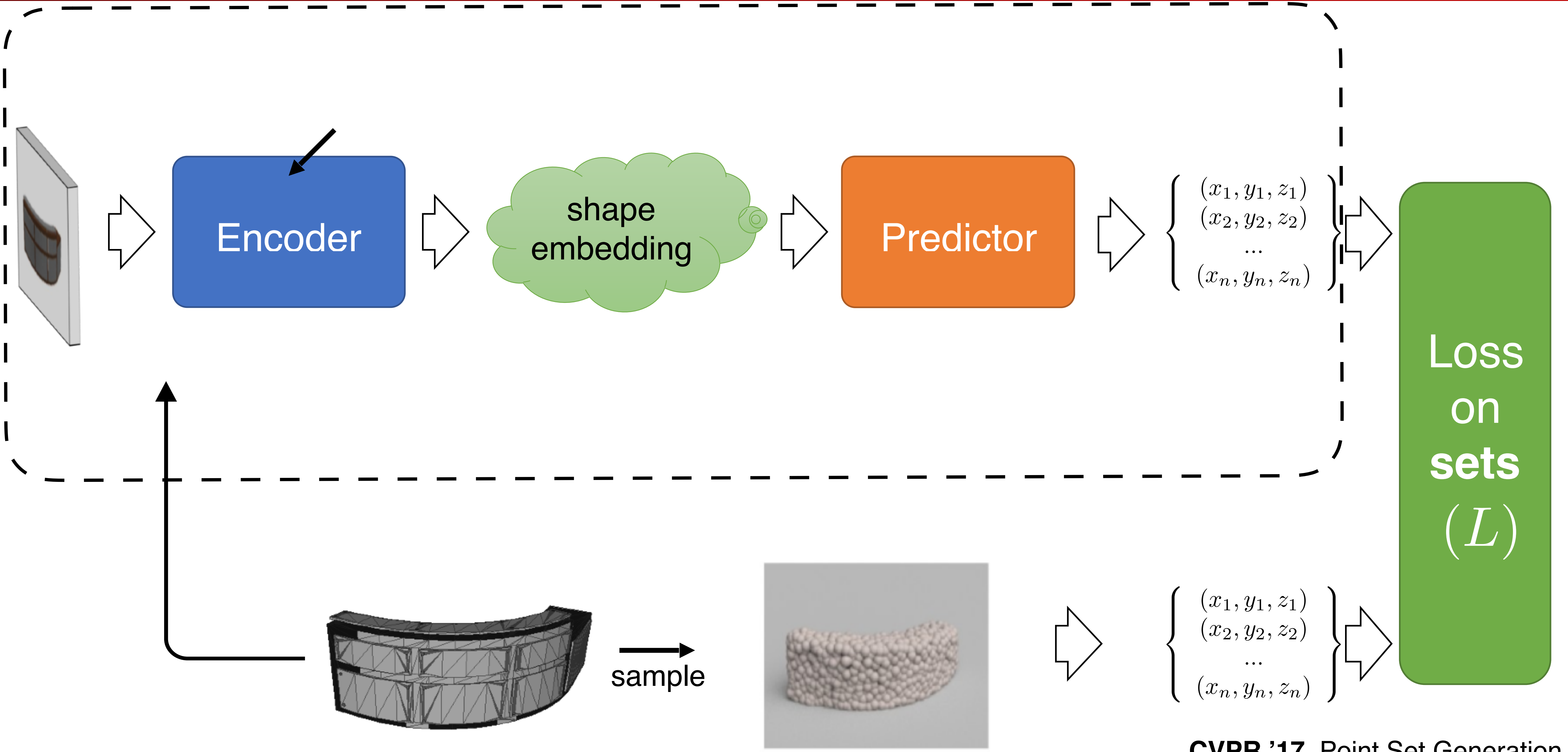
CVPR '17, Point Set Generation

Pipeline



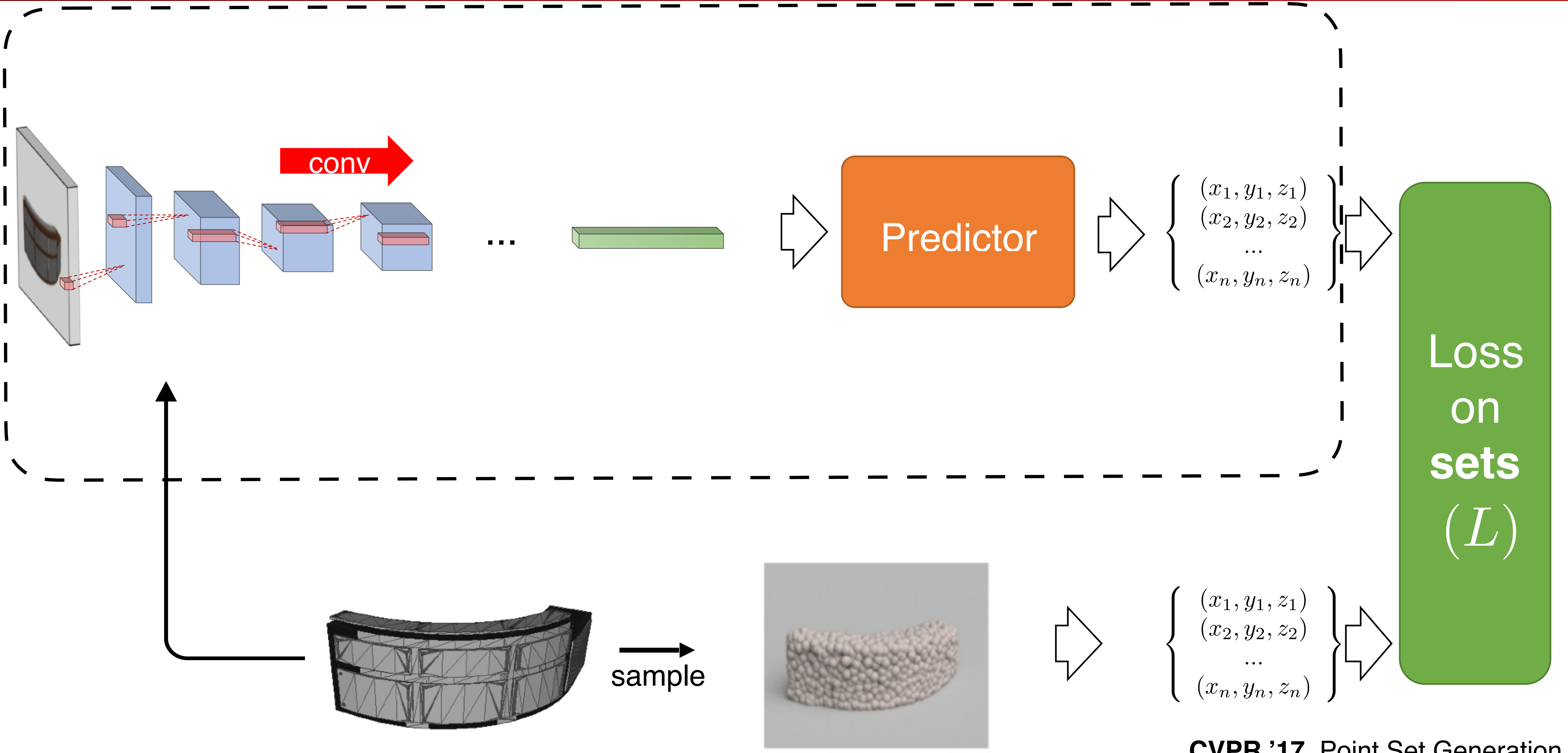
Encoder
shape embedding
Predictor
(x_1, y_1, z_1)
(x_2, y_2, z_2)
$\dots$
(x_n, y_n, z_n)
Loss on **sets** (L)
sample
(x_1, y_1, z_1)
(x_2, y_2, z_2)
$\dots$
(x_n, y_n, z_n)

Pipeline



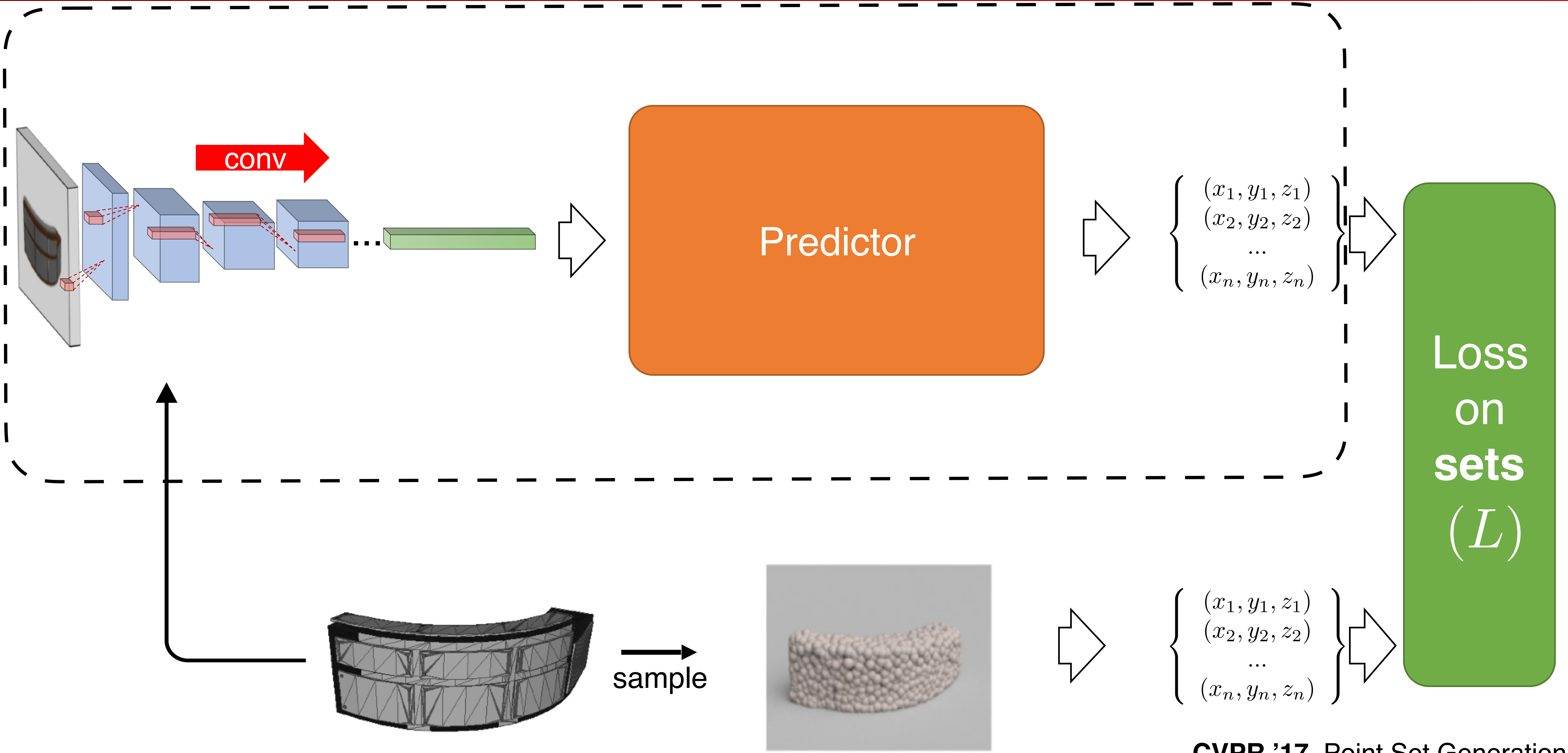
CVPR '17, Point Set Generation

Pipeline



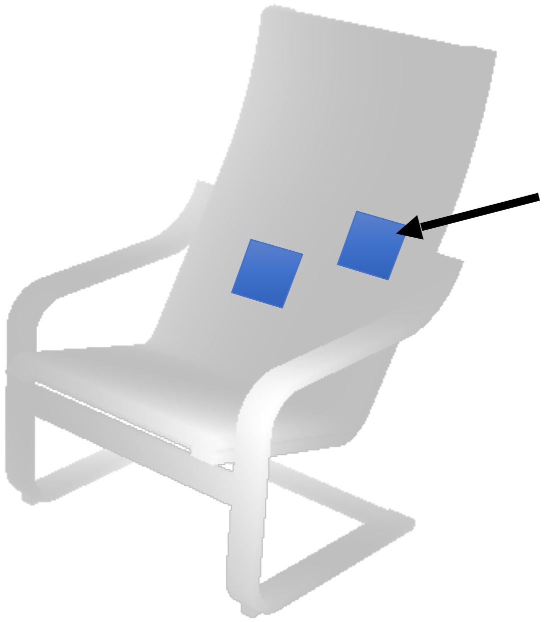
CVPR '17, Point Set Generation

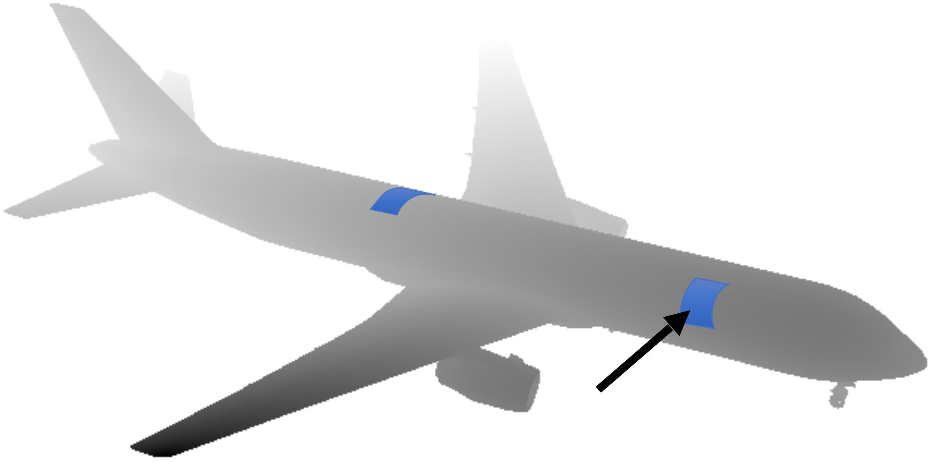
Pipeline



CVPR '17, Point Set Generation

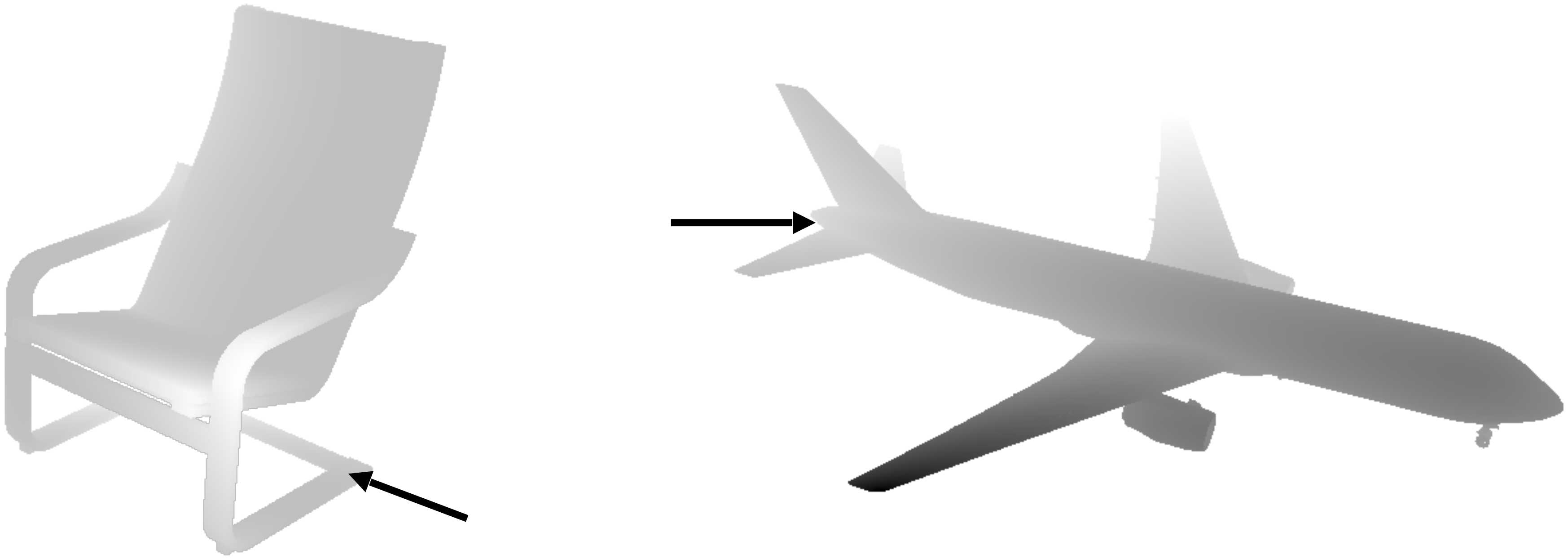
Natural statistics of geometry





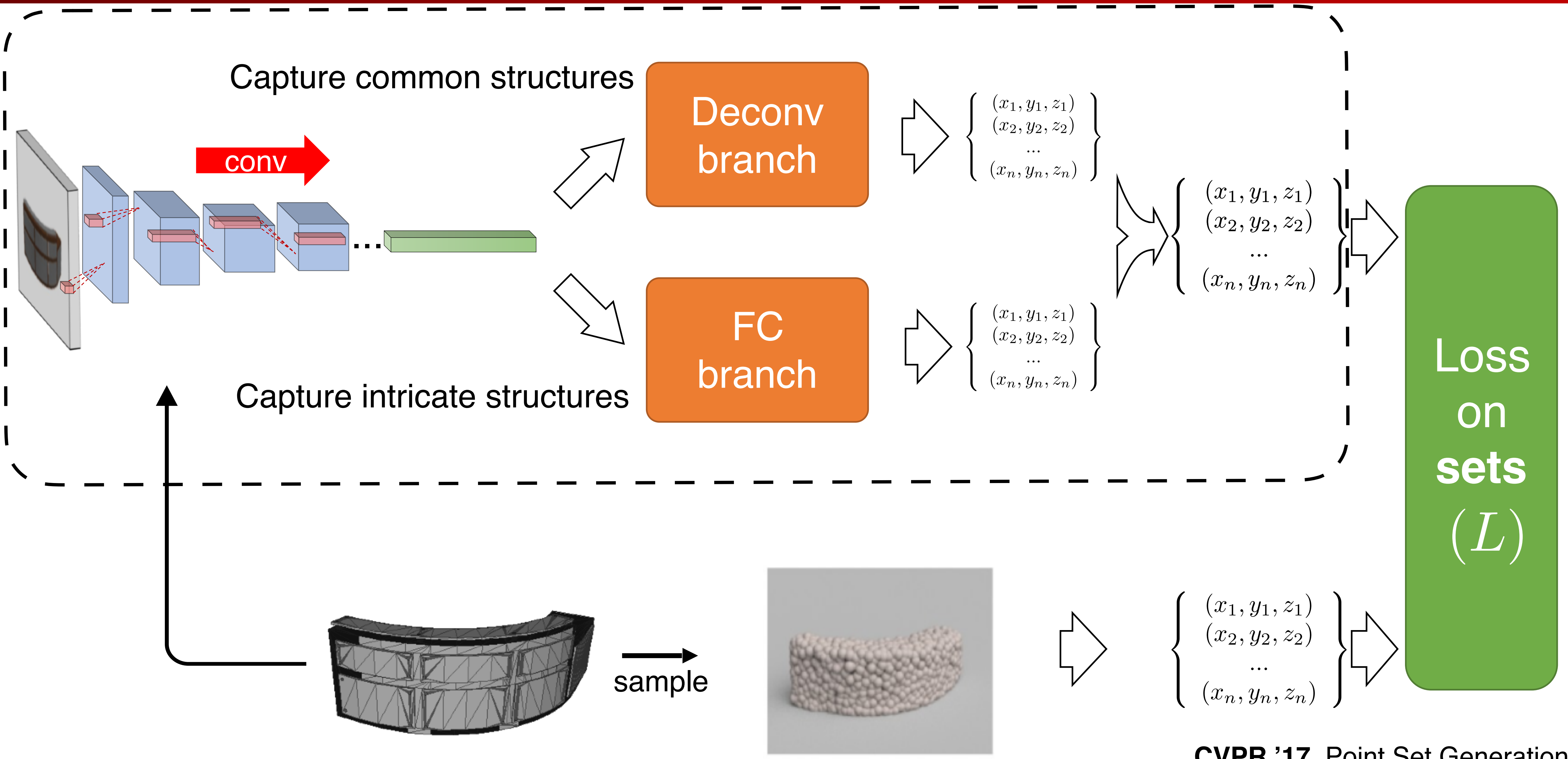
- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates

Natural statistics of geometry

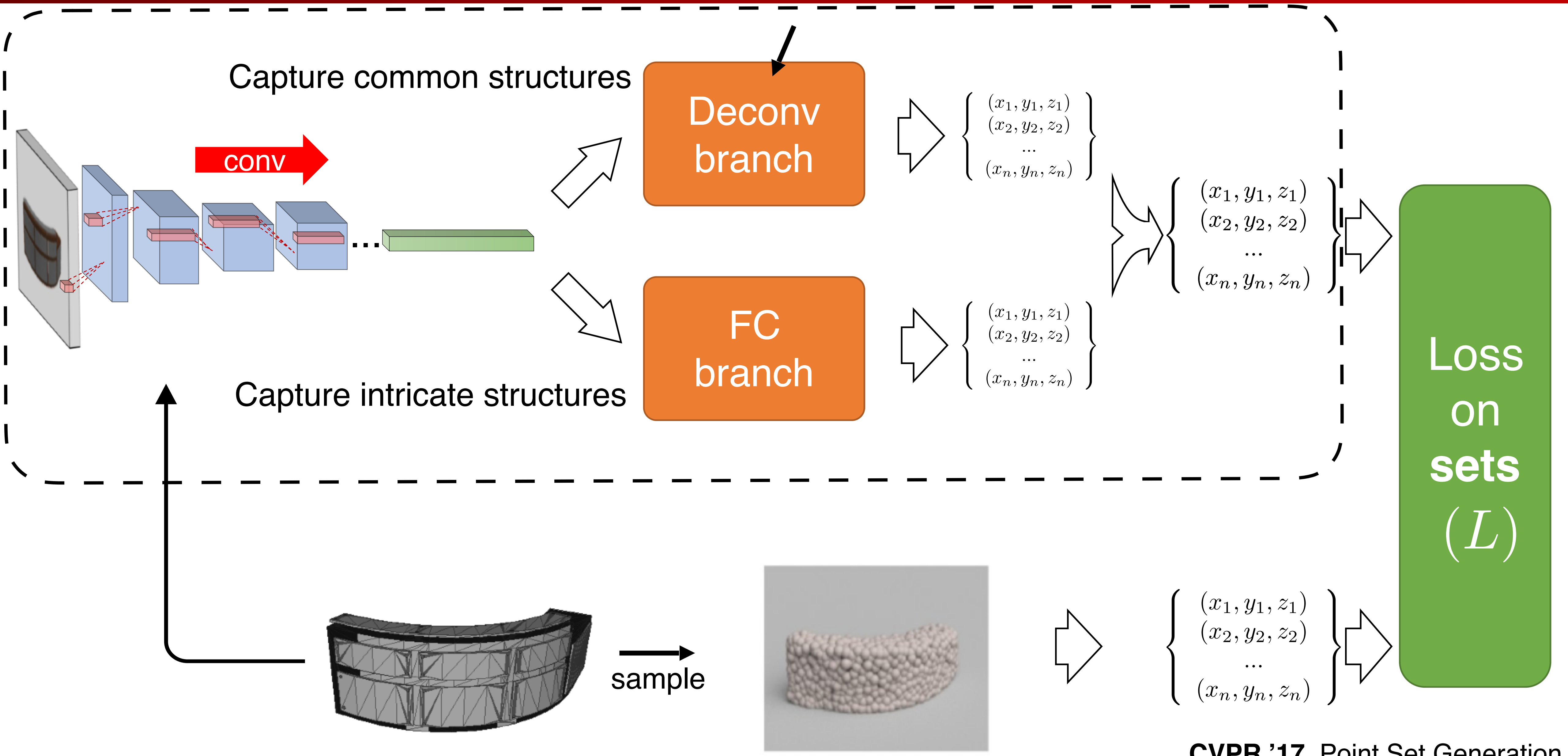


- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates
- Also some intricate structures
  - points have **high local variation**

Pipeline


Capture common structures
conv
Deconv branch
FC branch
Capture intricate structures
(x_1, y_1, z_1) (x_2, y_2, z_2) ... (x_n, y_n, z_n)
sample
Loss on **sets** (L)

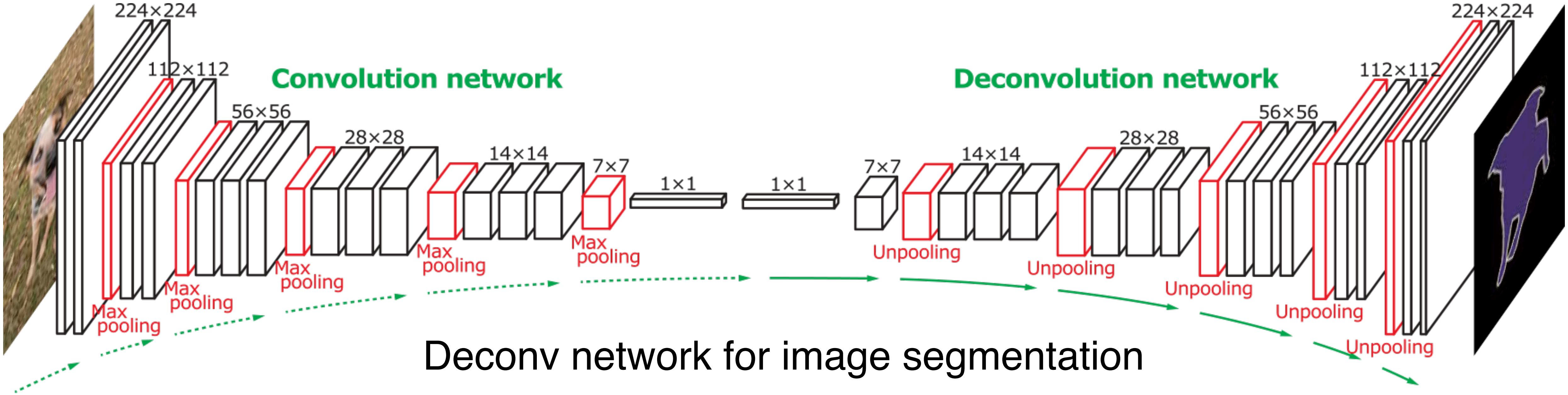
Pipeline



CVPR ’17, Point Set Generation

Review: deconv network

- Output nD arrays, e.g., 2D segmentation map
- **Common local patterns** are **learned from data**
- Predict **locally correlated** data well
- Weight sharing reduces the number of params

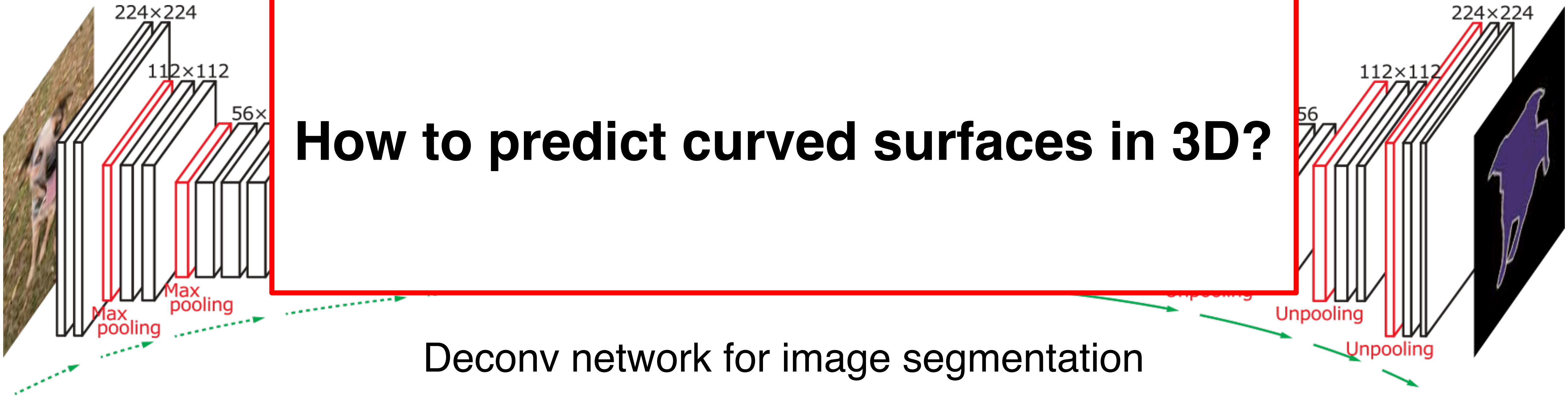


Deconv network for image segmentation

Credit: FCNN, Long et al.

Review: deconv network

- Output nD arrays, e.g., 2D segmentation map
- **Common local patterns** are **learned from data**
- Predict **locally correlated** data well
- Weight sharing reduces the number of params



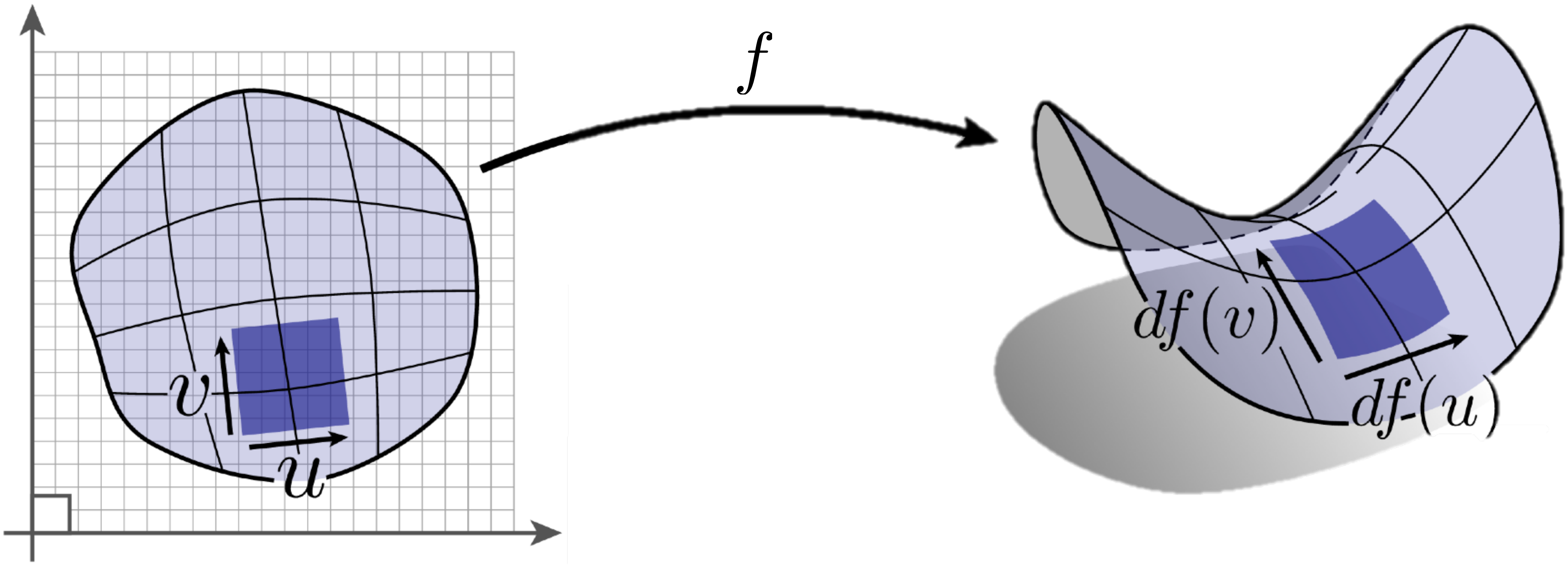
How to predict curved surfaces in 3D?

Deconv network for image segmentation

Credit: FCNN, Long et al.

Prediction of curved 2D surfaces in 3D

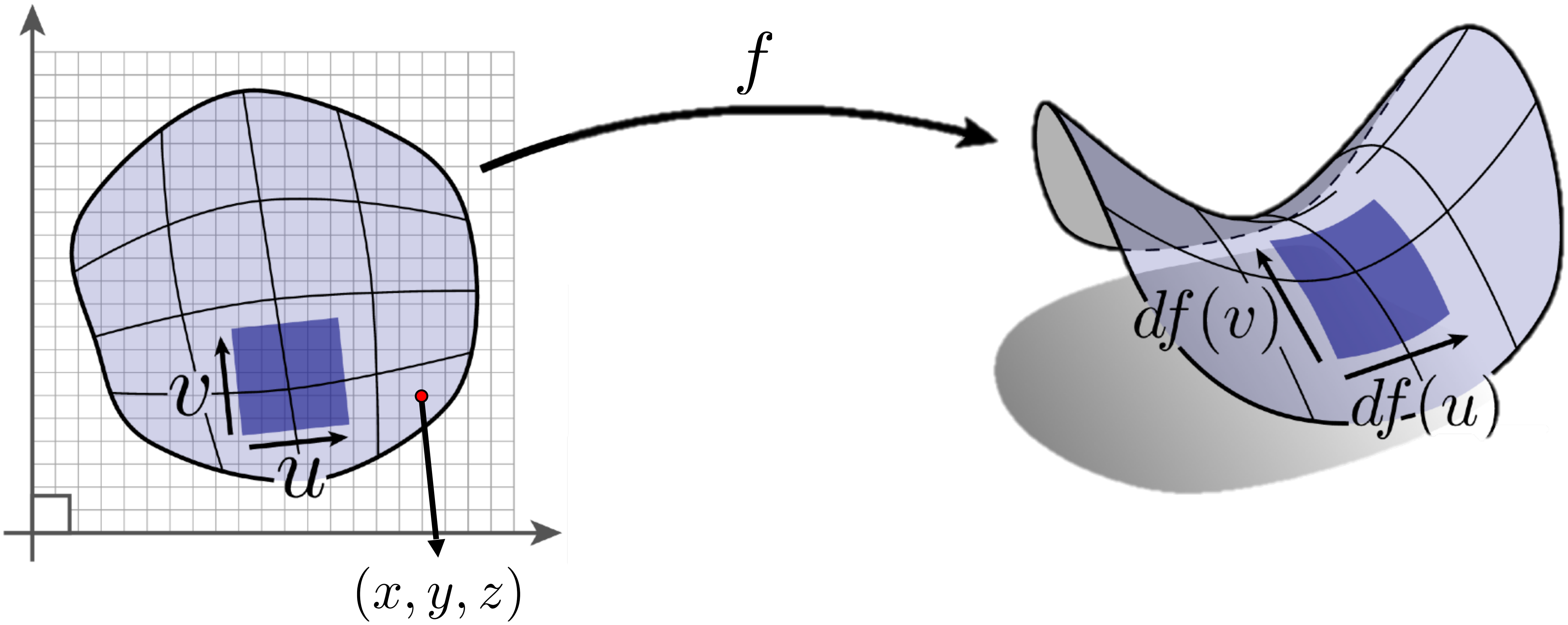
- Surface parametrization (2D↔3D mapping)



Credit: *Discrete Differential Geometry, Crane et al.*

Prediction of curved 2D surfaces in 3D

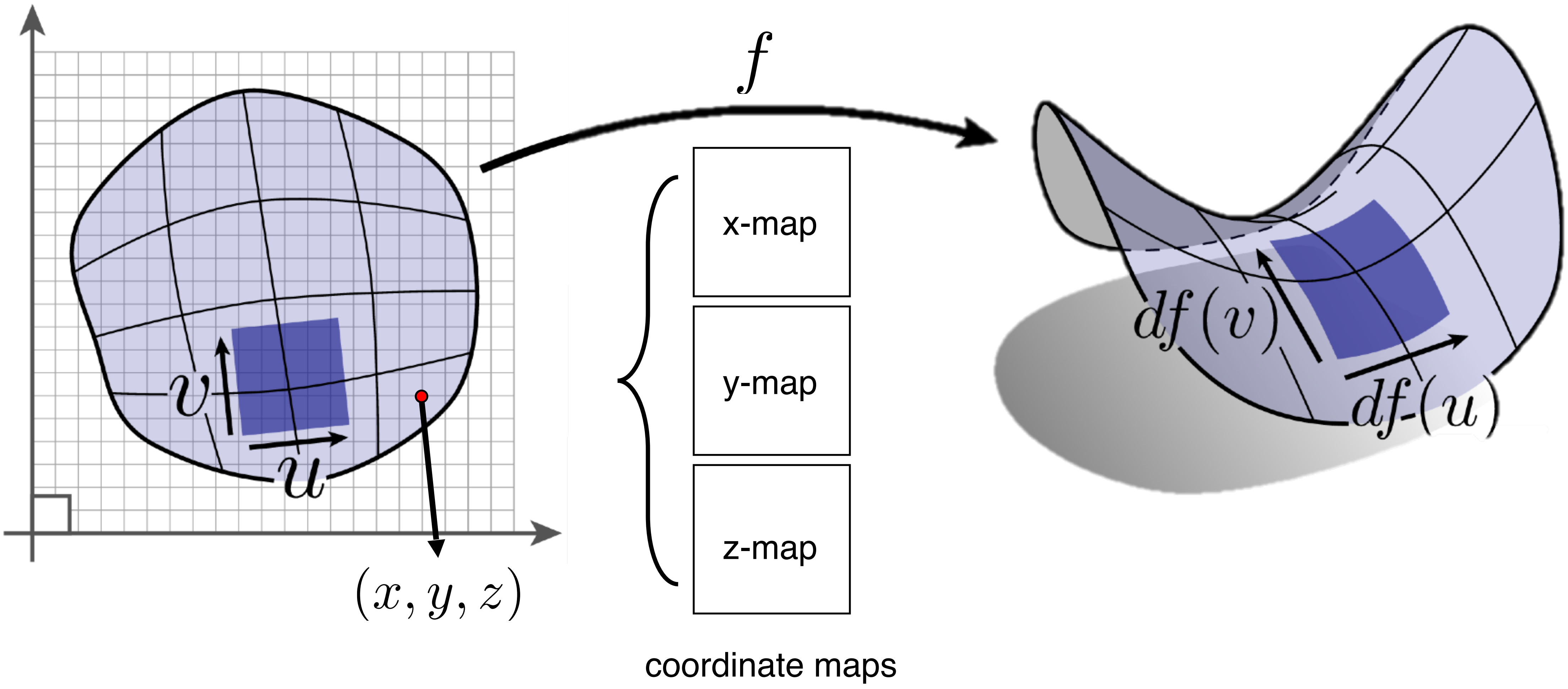
- Surface parametrization (2D$\leftrightarrow$3D mapping)



Credit: Discrete Differential Geometry, Crane et al.

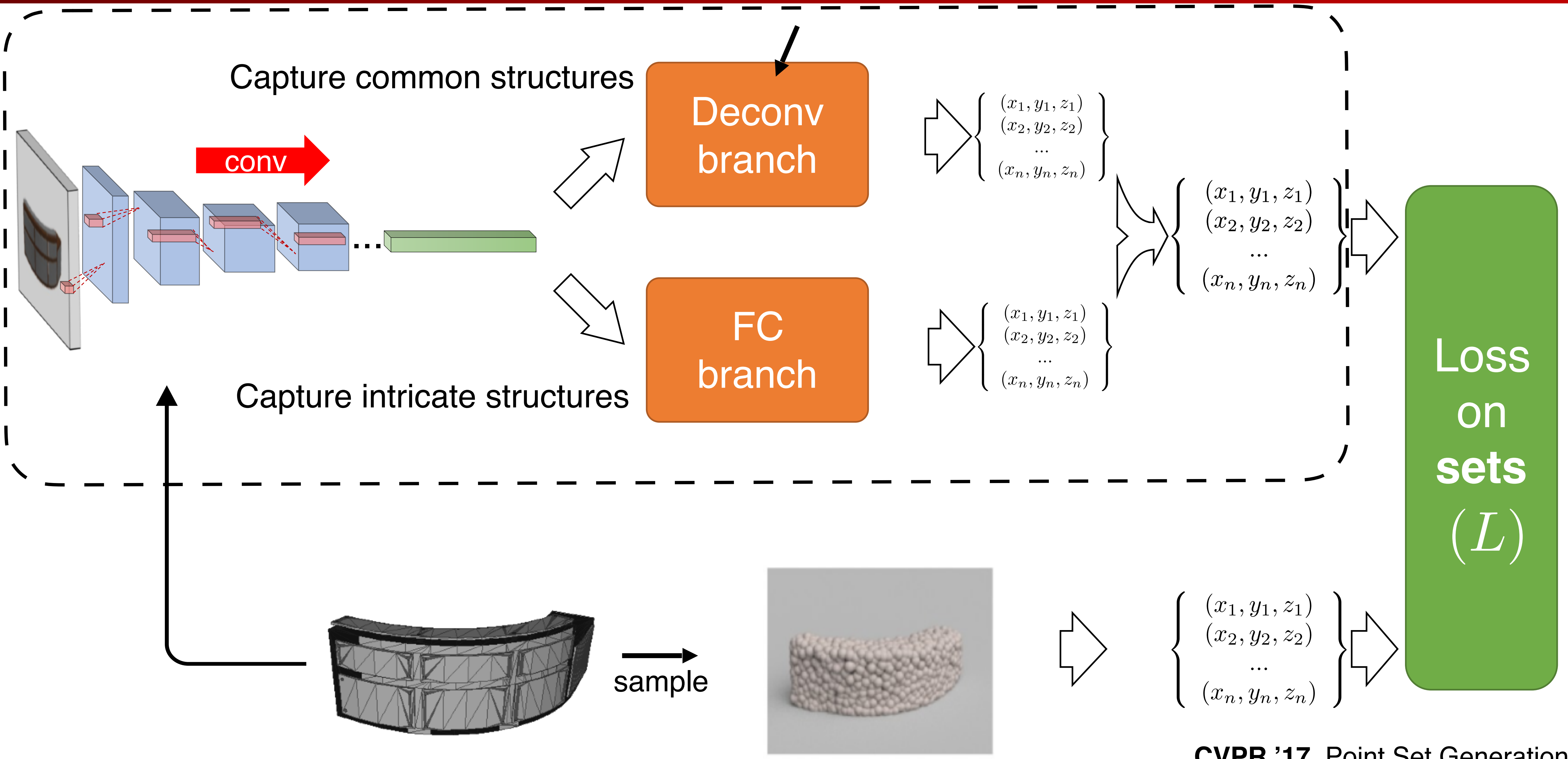
Prediction of curved 2D surfaces in 3D

- Surface parametrization (2D$\leftrightarrow$3D mapping)



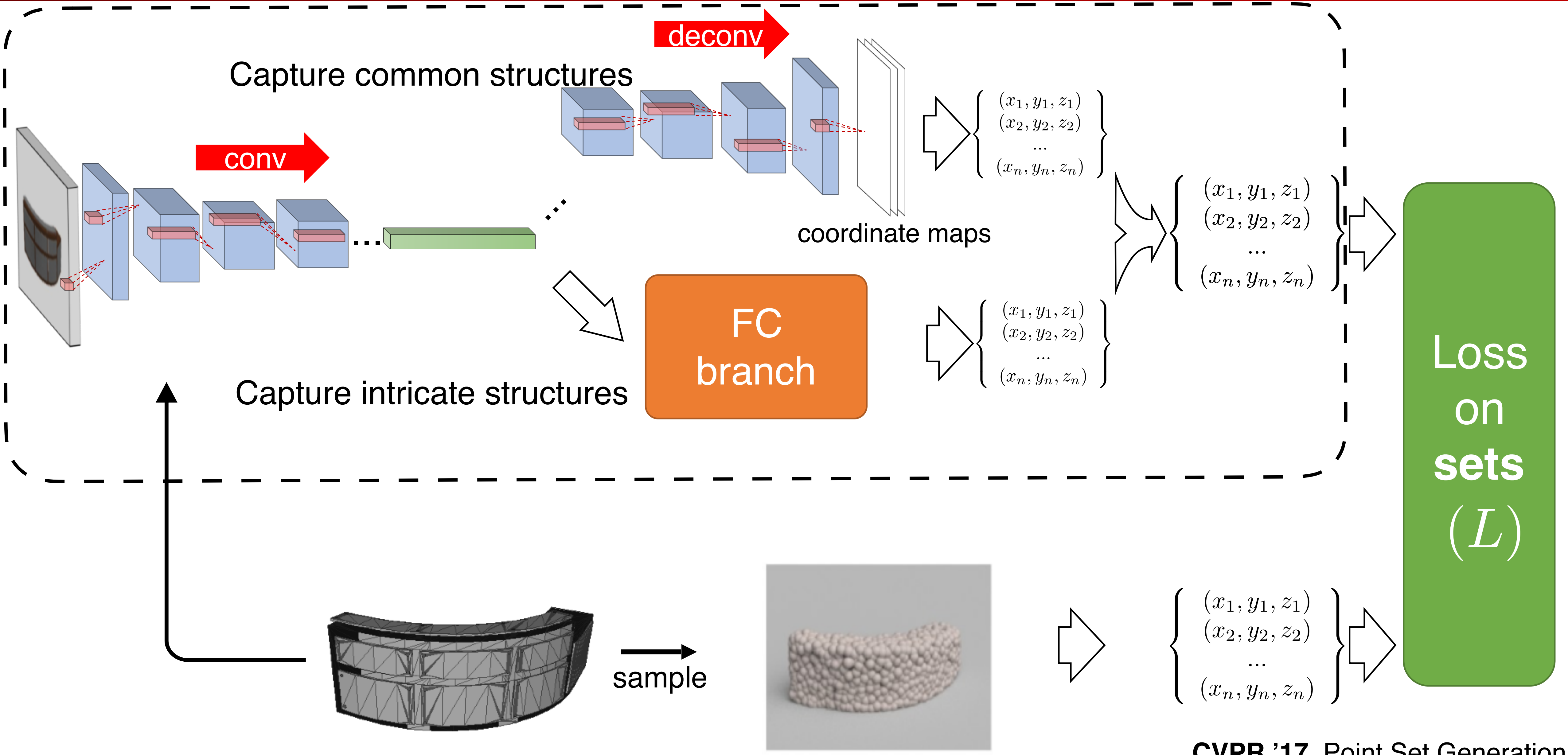
Credit: Discrete Differential Geometry, Crane et al.

Parametrization prediction by deconv network



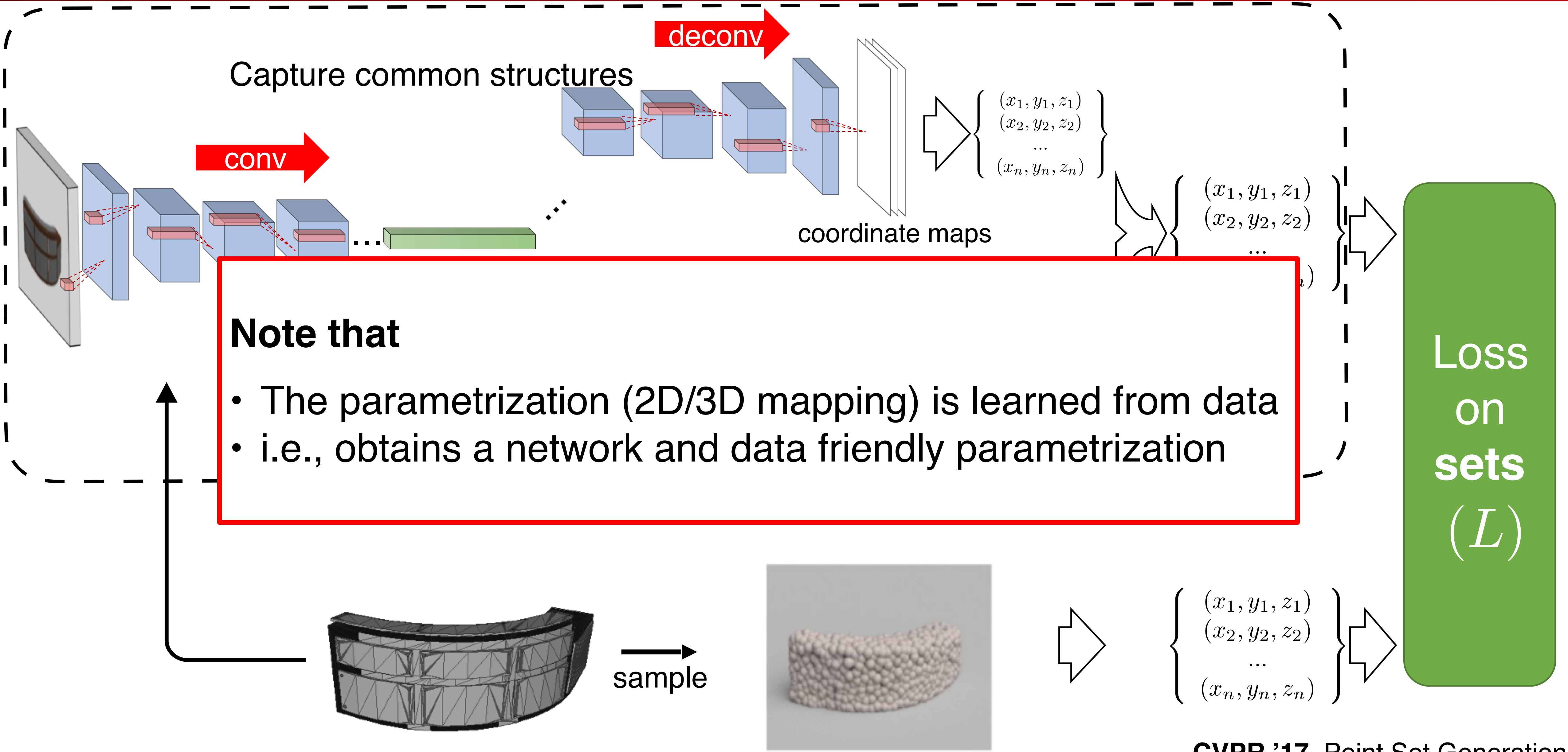
CVPR '17, Point Set Generation

Parametrization prediction by deconv network



CVPR '17, Point Set Generation

Parametrization prediction by deconv network



Note that

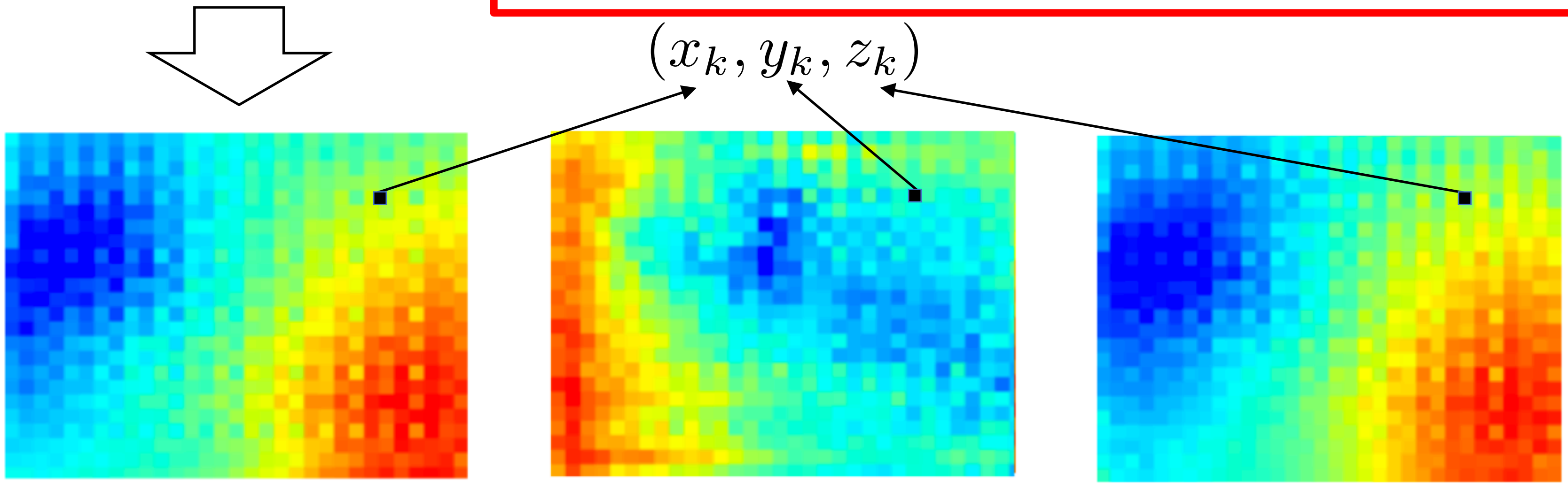
- The parametrization (2D/3D mapping) is learned from data
- i.e., obtains a network and data friendly parametrization

CVPR '17, Point Set Generation

Visualization of the learned parameterization

Observation:

- Learns a **smooth** parametrization
- Because deconv net tends to predict data with local correlation



(x_k, y_k, z_k)

map of x coord

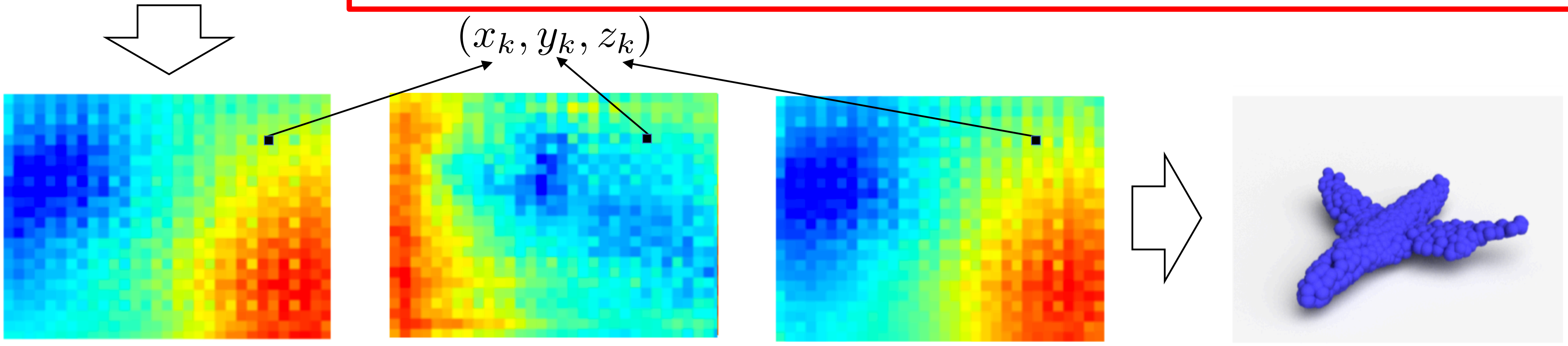
map of y coord

map of z coord

Visualization of the learned parameterization

Observation:

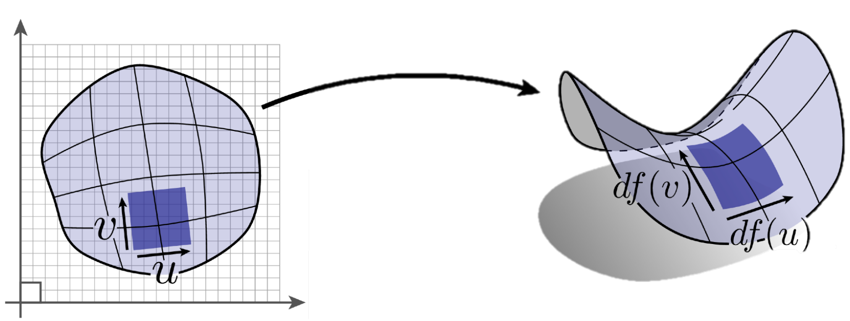
- Learns a **smooth** parametrization
- Because deconv net tends to predict data with local correlation
- Corresponds to **smooth surfaces!**



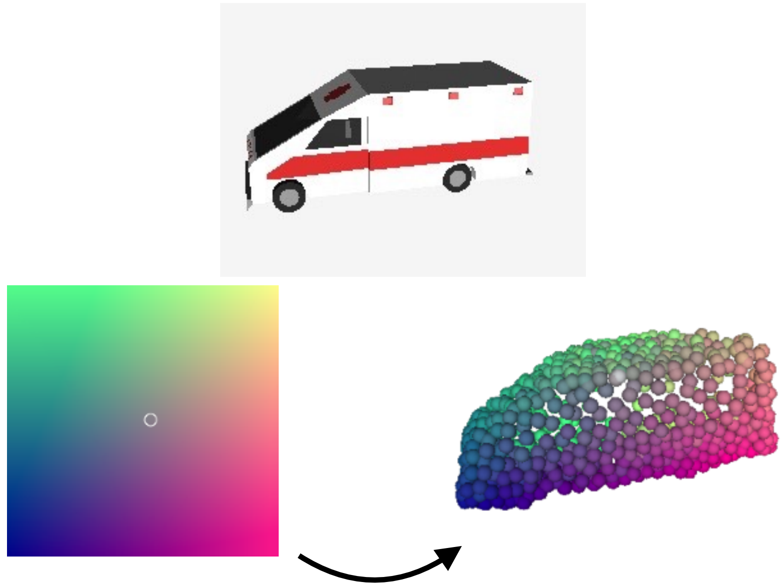
map of x coord

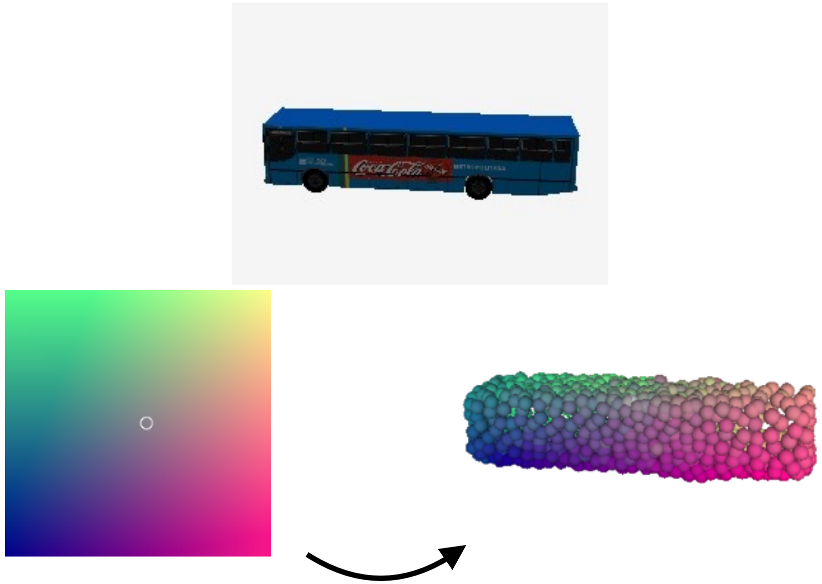
map of y coord

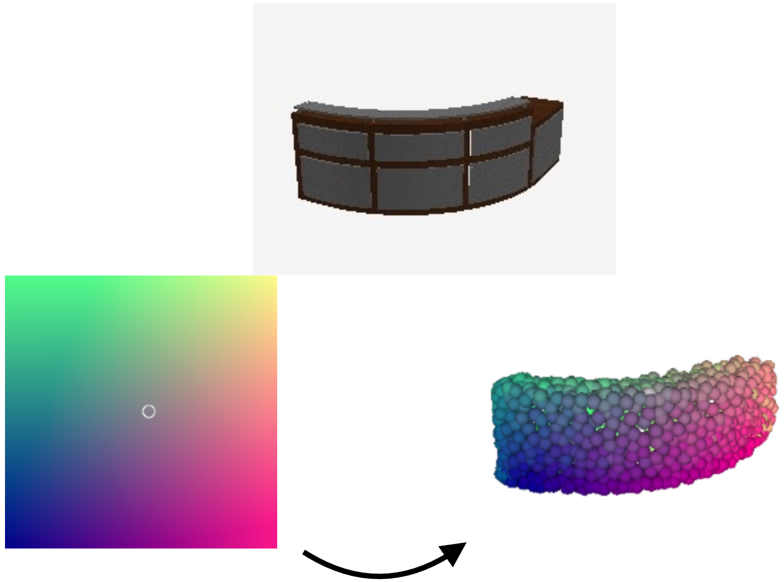
map of z coord



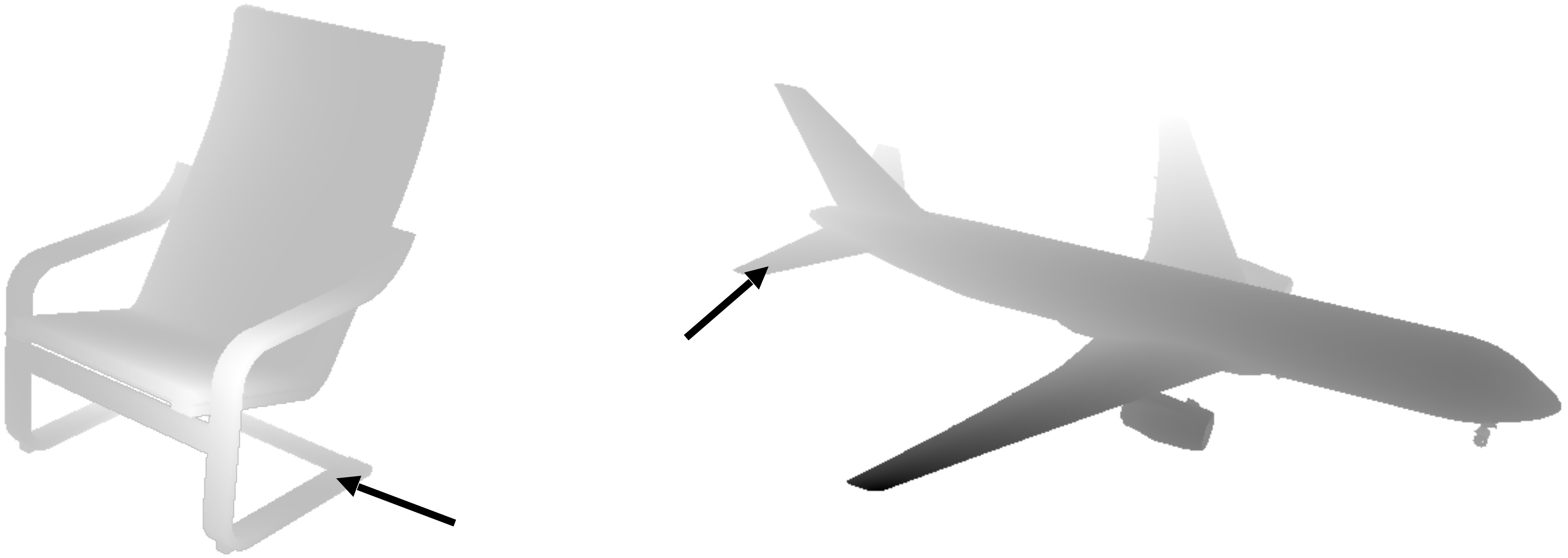
v
u
$df(v)$
$df(u)$





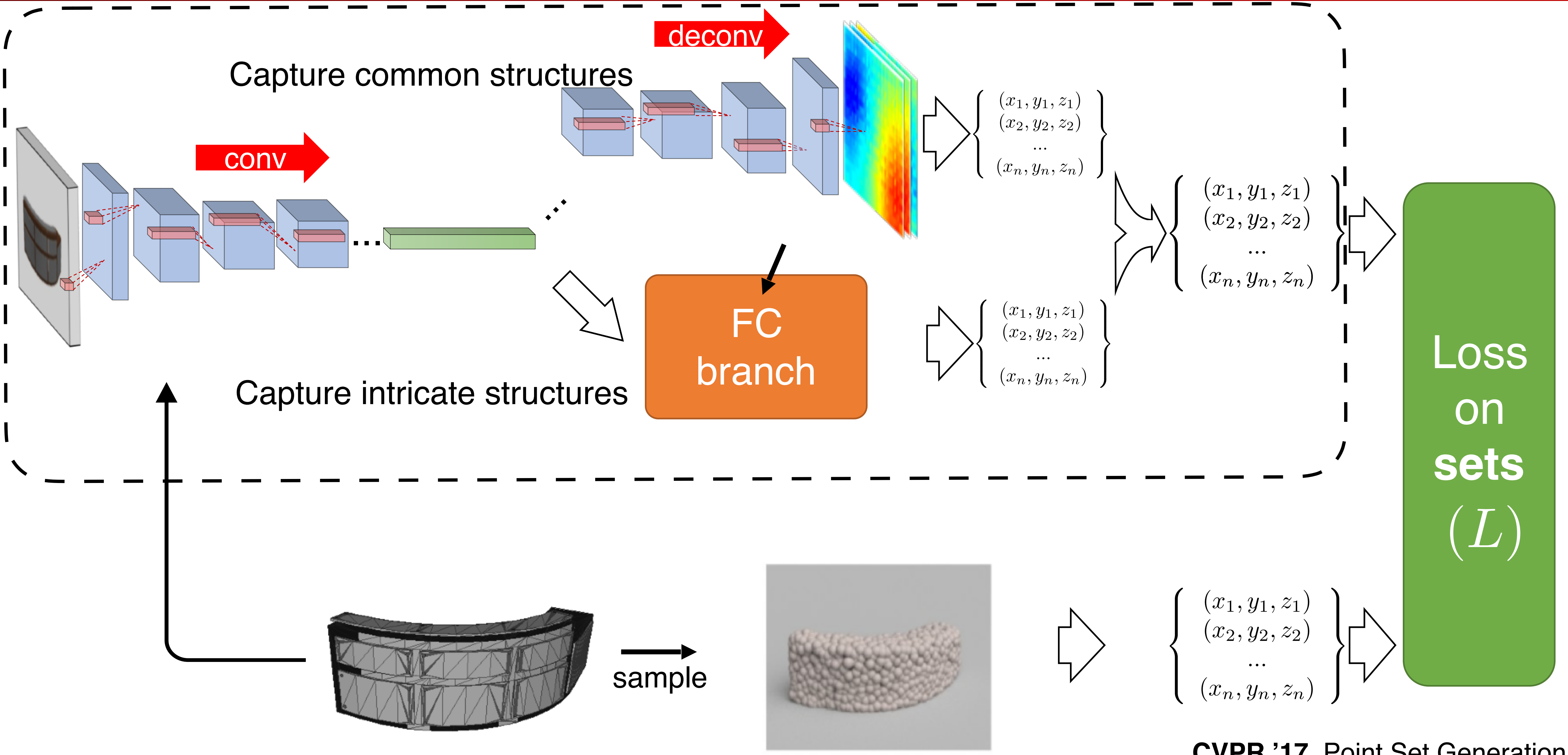


Natural statistics of geometry



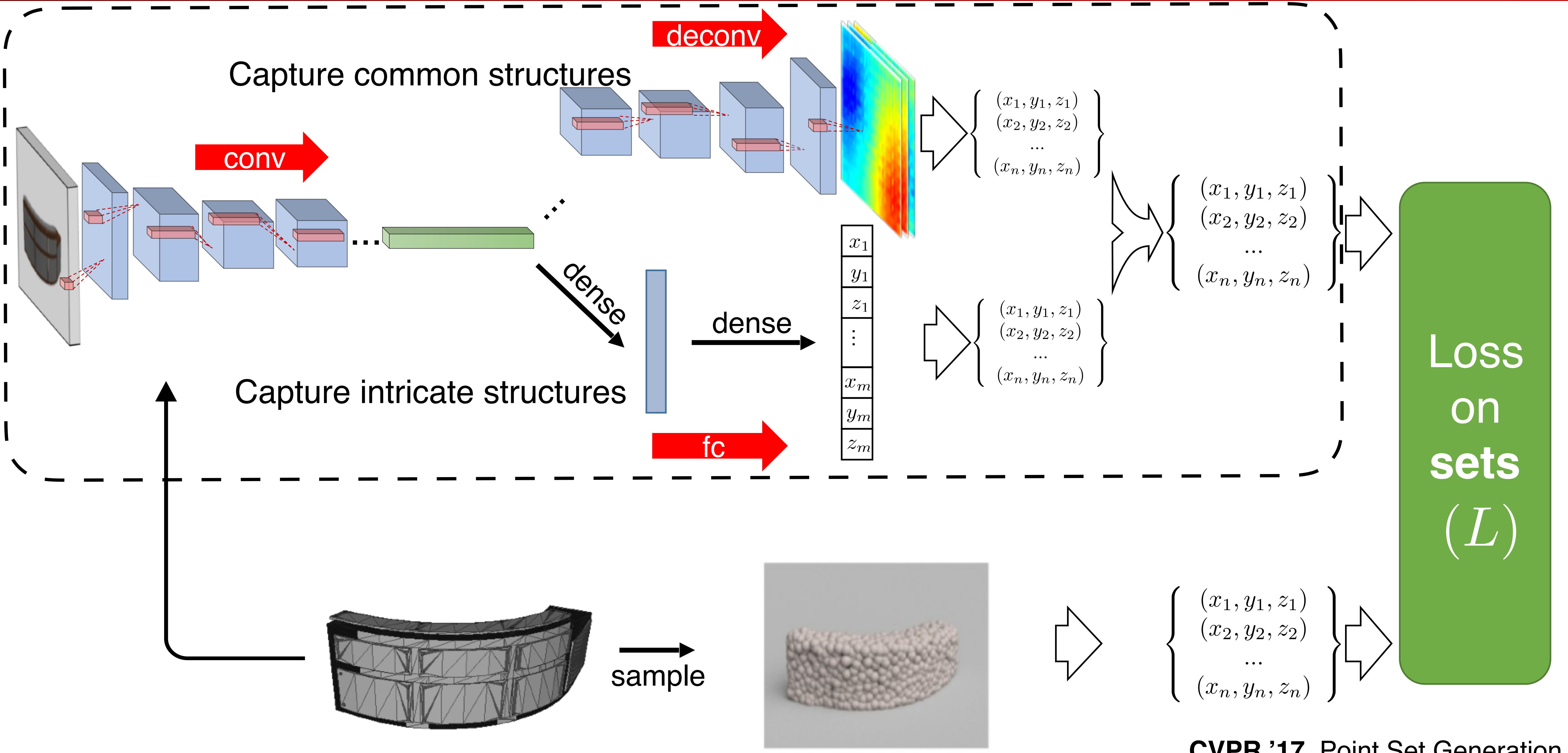
- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates
- Also some intricate structures
  - points have **high local variation**

Pipeline



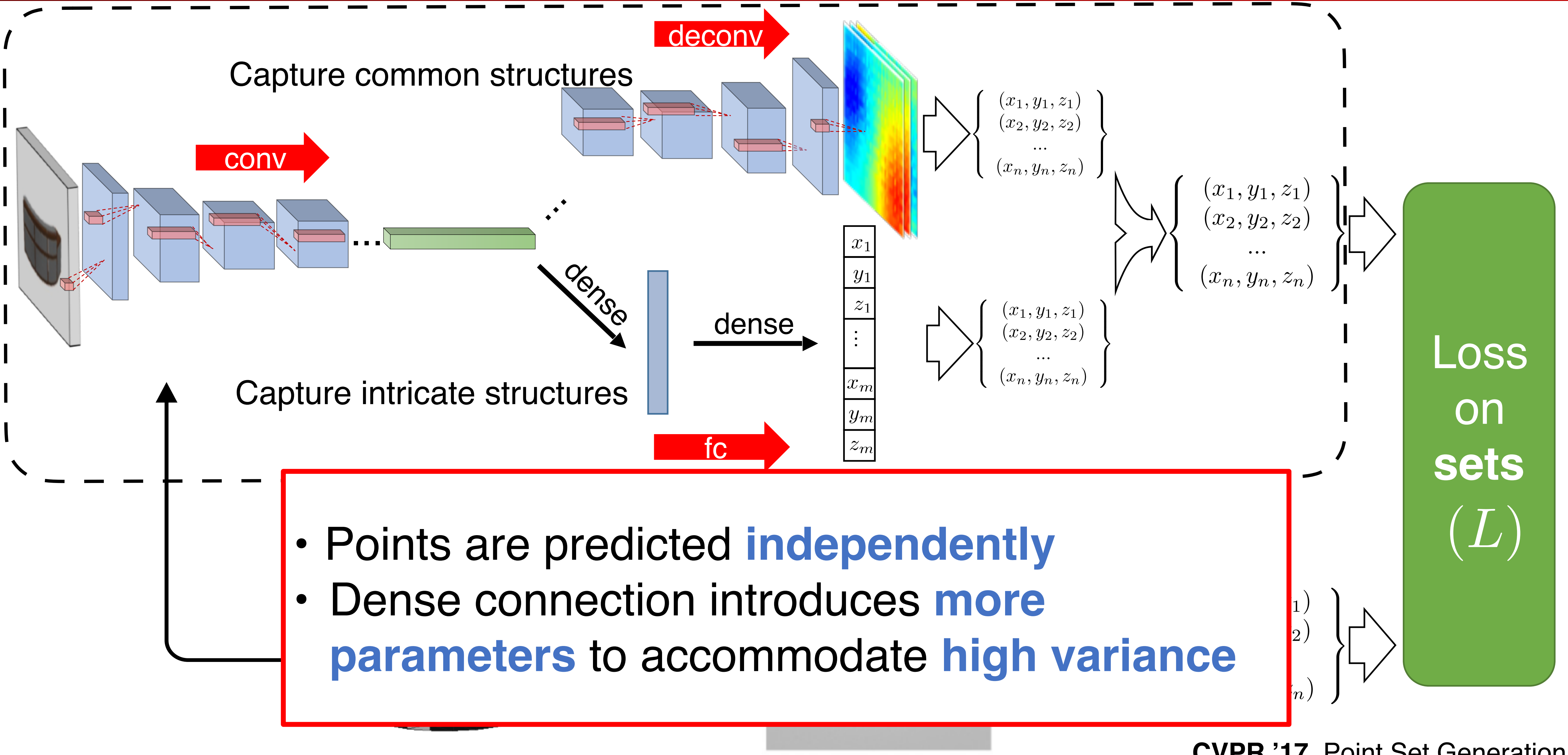
CVPR '17, Point Set Generation

Pipeline



CVPR '17, Point Set Generation

Pipeline

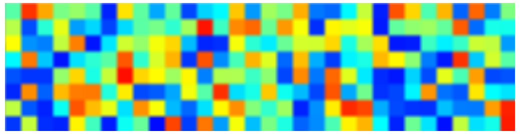


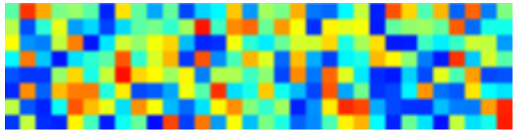
- Points are predicted **independently**
- Dense connection introduces **more parameters** to accommodate **high variance**

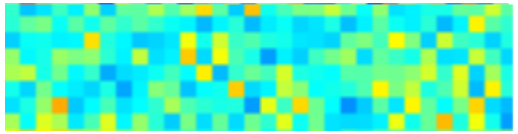
Visualization of the effect of FC branch

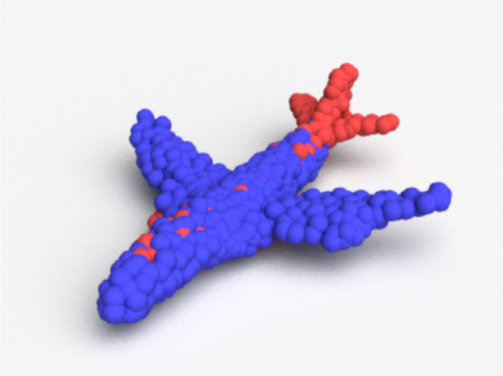
Observation:

• The arrangement of predicted points are uncorrelated









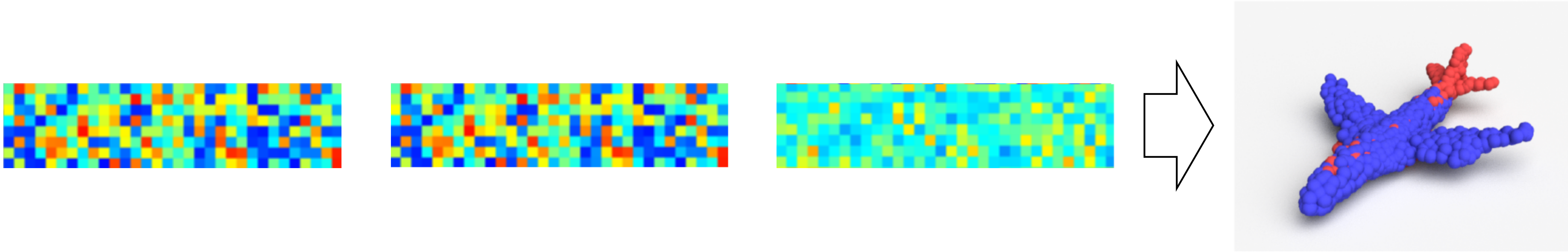
x-coord y-coord z-coord **red**

CVPR '17, Point Set Generation

Visualization of the effect of FC branch

Observation:

- The arrangement of predicted points are uncorrelated
- Located at **fine** structures



x-coord y-coord z-coord **red**

Q: Which color corresponds to the deconv branch? FC branch?

CVPR '17, Point Set Generation

Q: Which color corresponds to the deconv branch? FC branch?

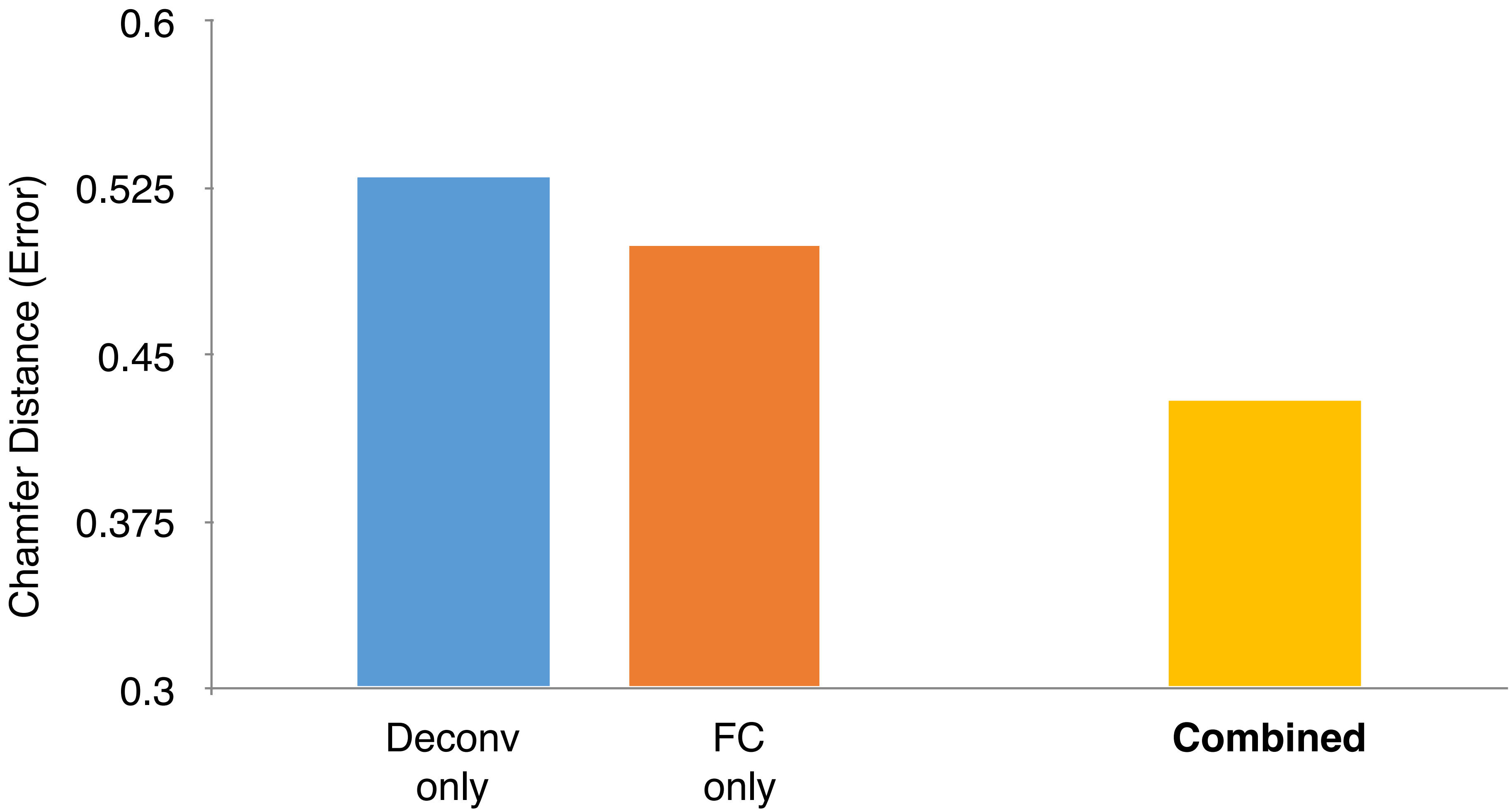
blue: deconv branch – **large, smooth** structures

red: FC branch – **intricate** structures

CVPR ’17, Point Set Generation

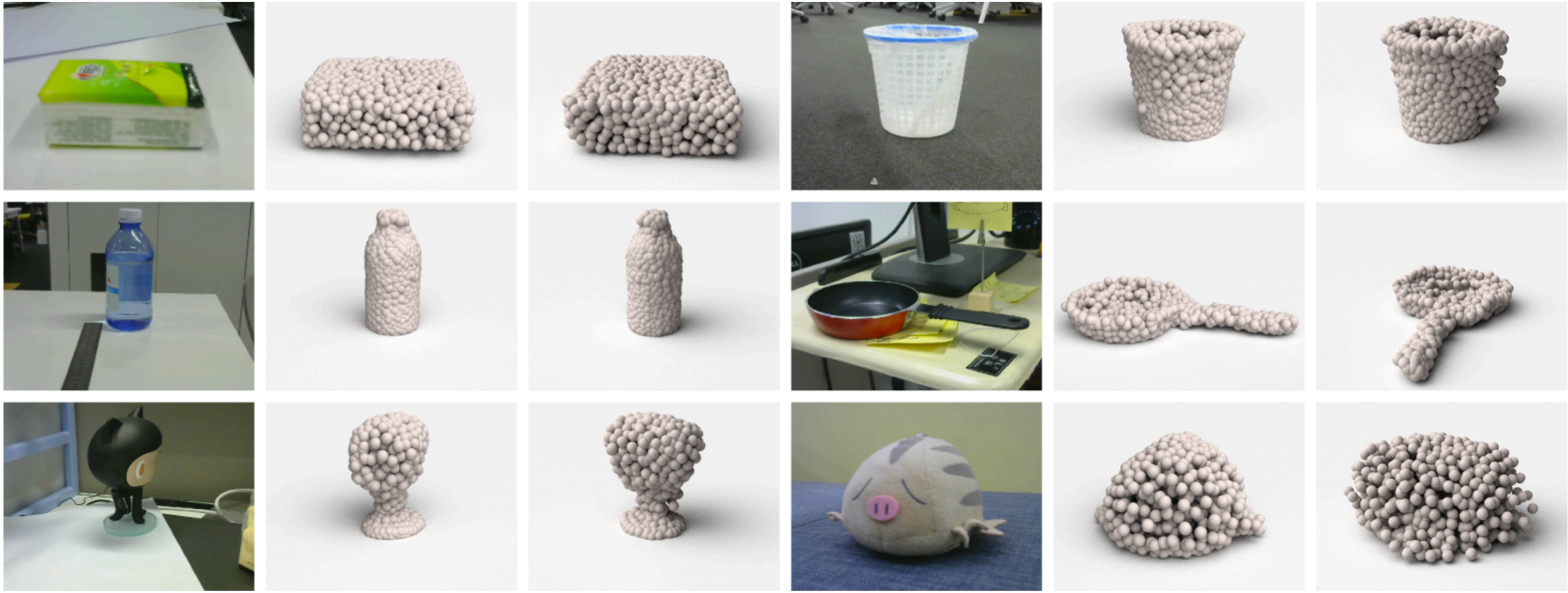
Effect of combining two branches

Train/tested on 2K object categories

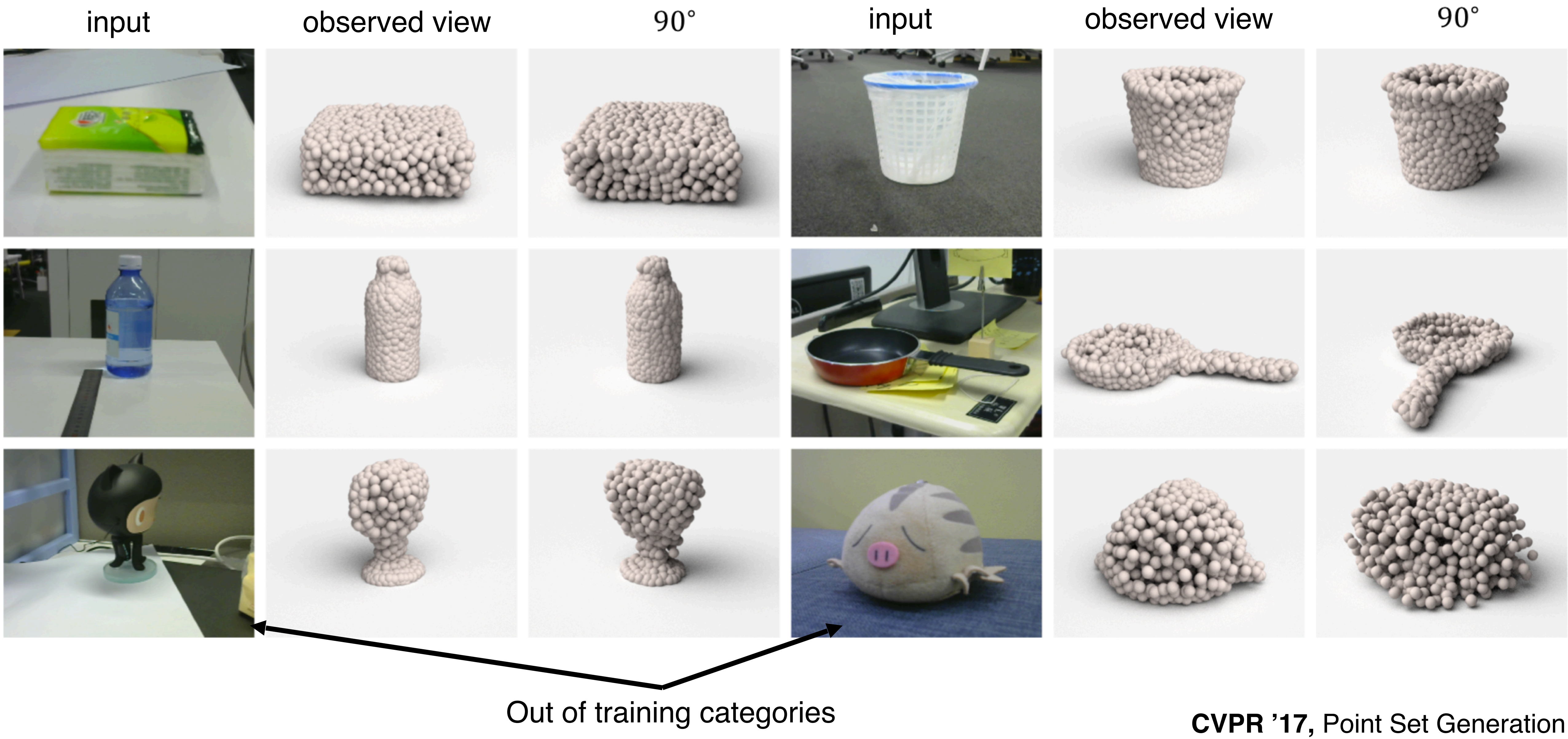


Real-world results

input | observed view | 90° | input | observed view | 90°



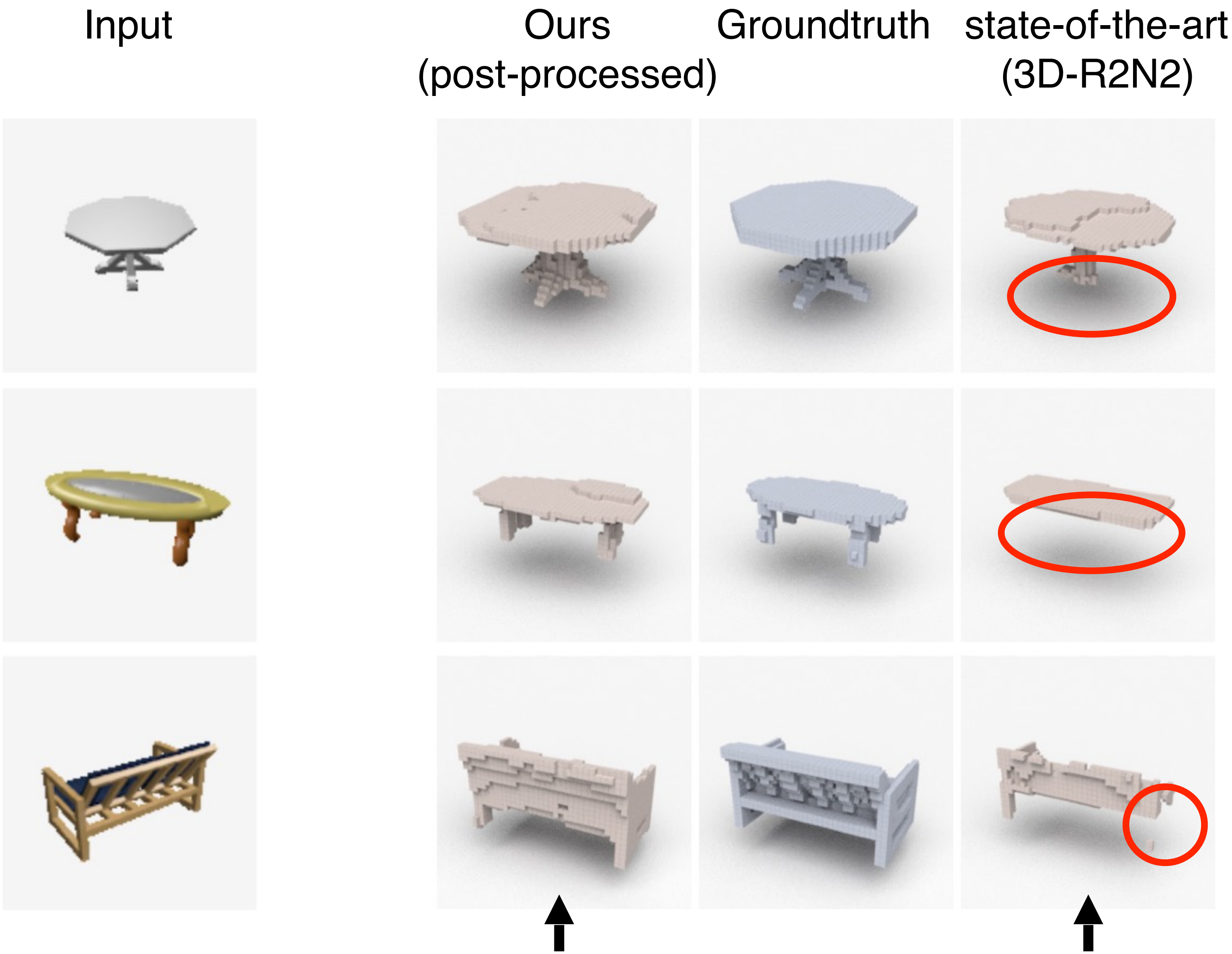
Generalization to unseen categories



CVPR '17, Point Set Generation

Comparison to state-of-the-art

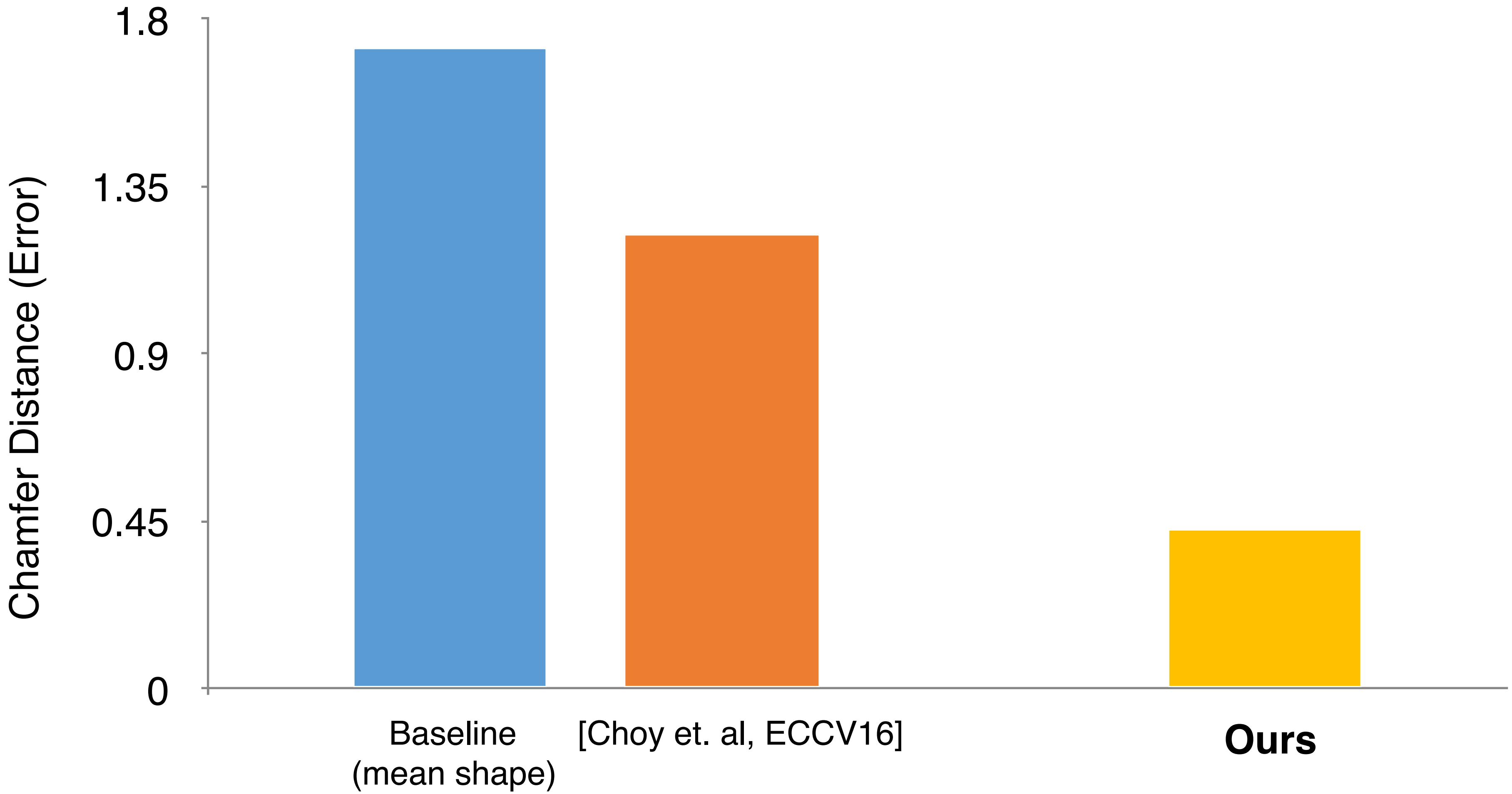
- Better global structure
- Better details



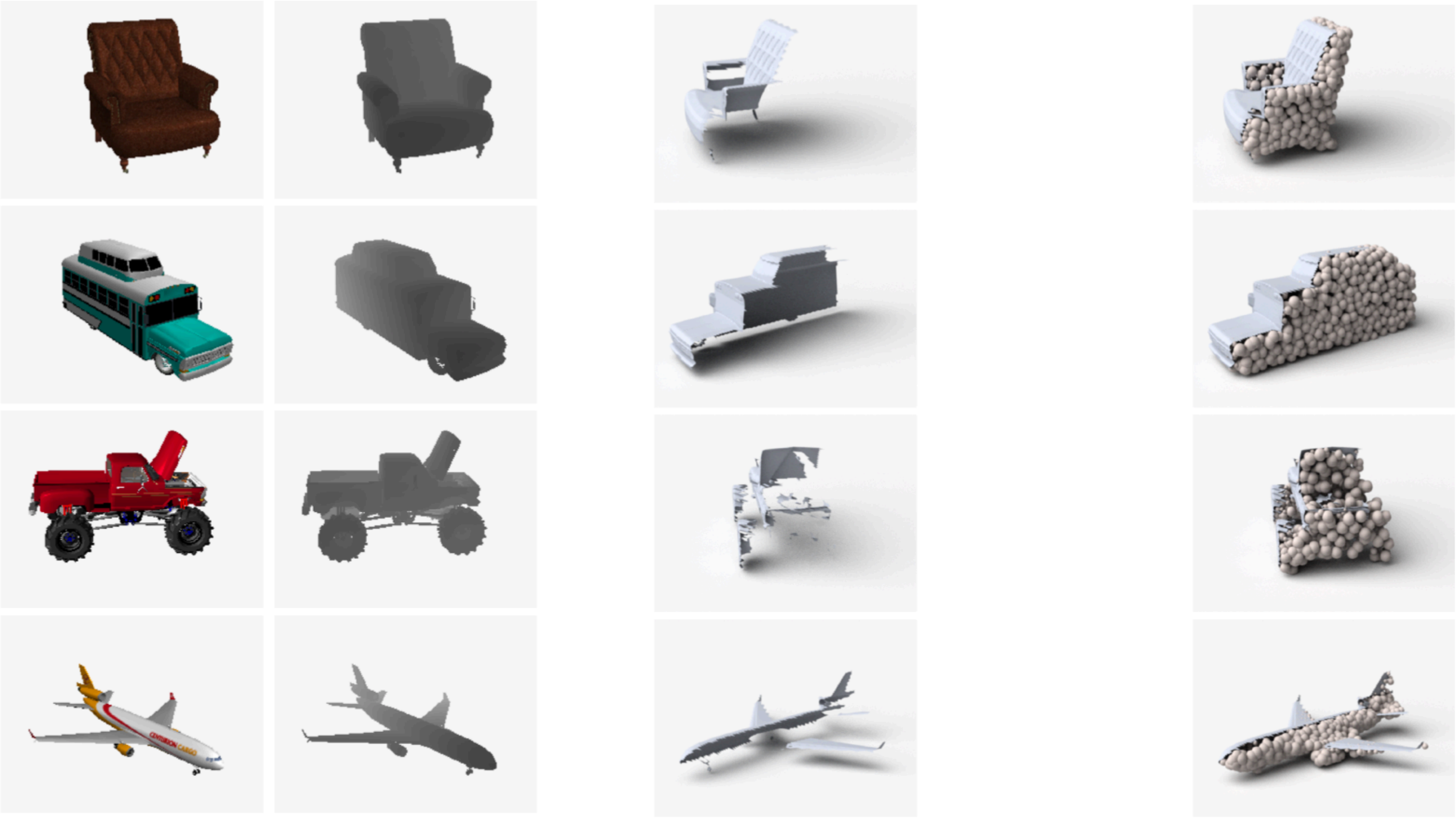
CVPR '17, Point Set Generation

Comparison to state-of-the-art

Trained/tested on 2K object categories



Extension: shape completion for RGBD data



RGBD map (input) 90° view of input output: completed point cloud

CVPR '17, Point Set Generation

Open problems

A better metric that takes the best of Chamfer and EMD?

How to add further structure constraints?

How to extend the pipeline to scene level?

How generalizable the method is?

In principle, what is the generalizability of a geometry estimator? To what extend is 3D perception ability innate or learned?

Outline

- Motivation
- Data
- Algorithms for 2D-3D lifting
- **Shape abstraction by volumetric primitives**

How about learning to predict geometric forms?

Candidates:

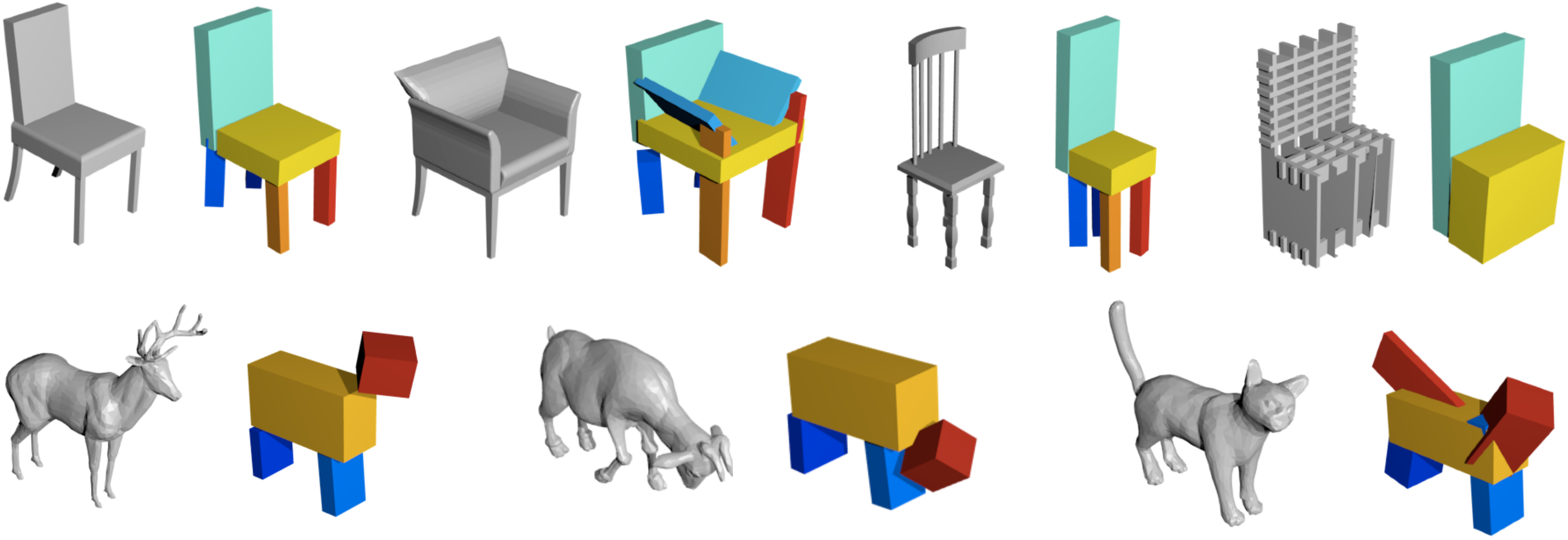
Rasterized form (regular grids)

- multi-view images
- depth map
- volumetric

Geometric form (irregular)

- polygonal mesh
- **point cloud**
- ★ **primitive-based CAD models**

Primitive-based assembly

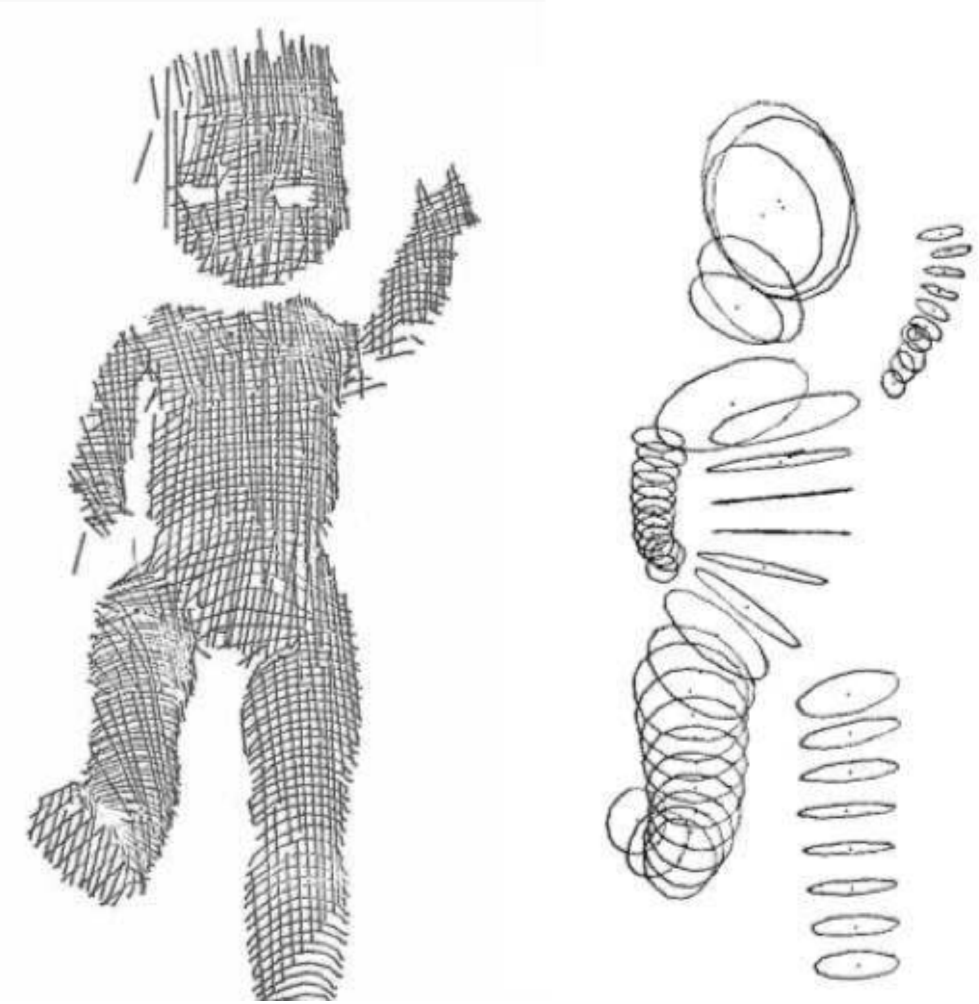


We **learn** to predict a corresponding shape composed by primitives.
It allows us to predict **consistent** compositions across objects.

Unsupervised parsing

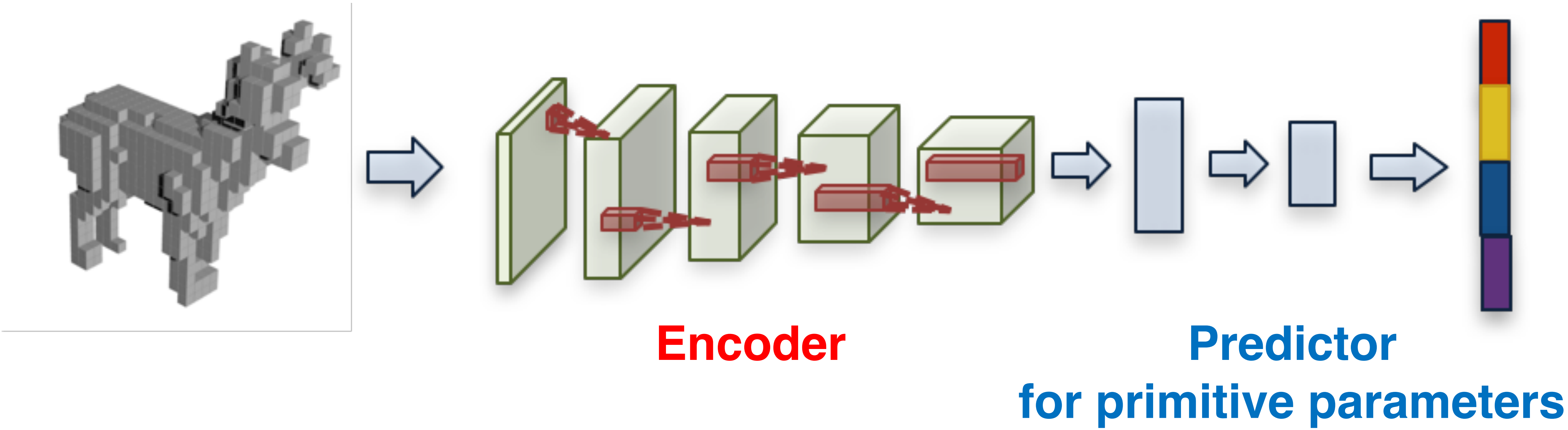
Each point is colored according to the assigned primitive

A historical overview



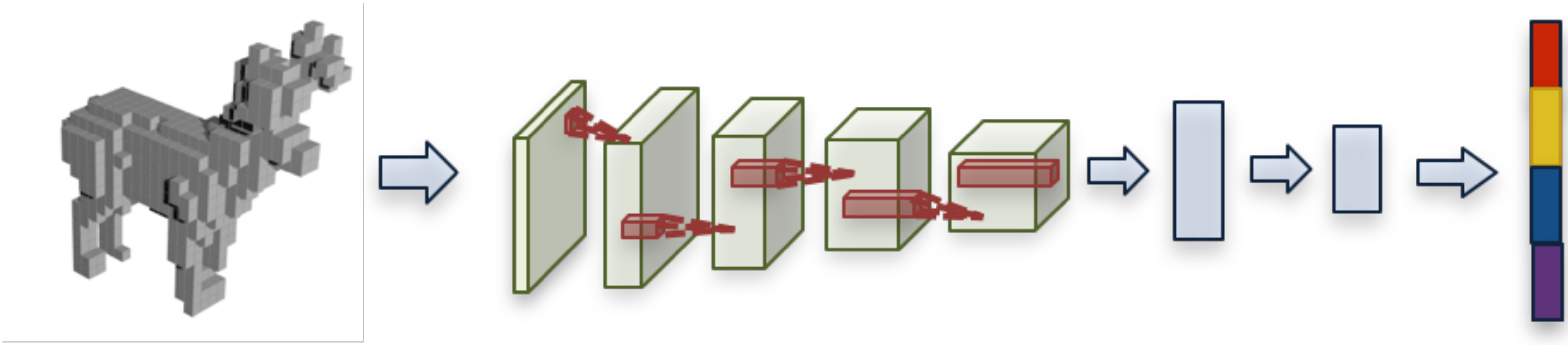
Generalized Cylinders, Binford (1971)

Approach



We predict primitive parameters: size, rotation, translation of M cuboids.

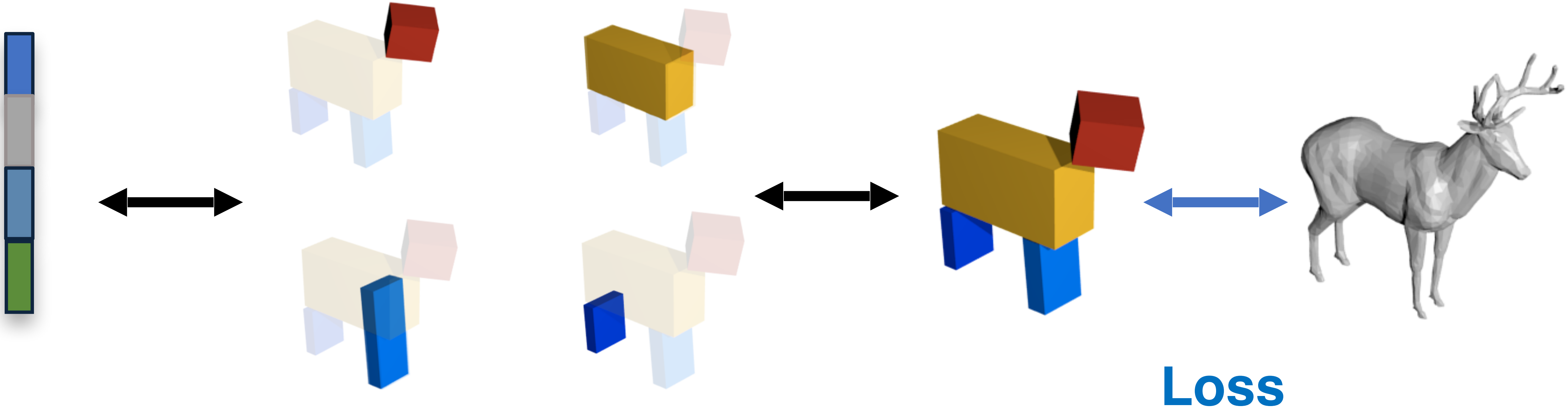
Approach



We predict primitive parameters: size, rotation, translation of M cuboids.

Variable number of parts? We predict “primitive existence probability”

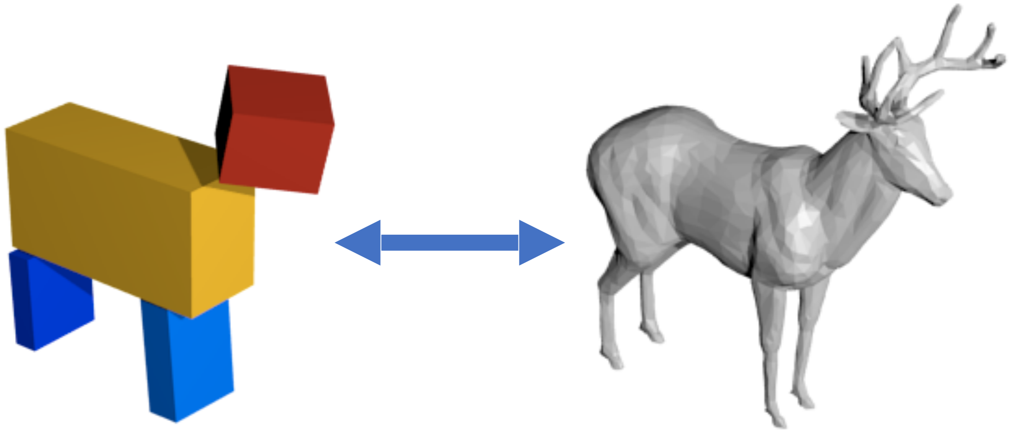
Loss function



Loss function construction

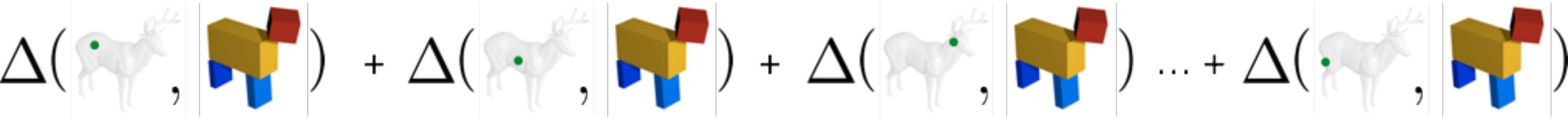
Basic idea: **Chamfer distance!**

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



Loss function construction

Sample points on the groundtruth mesh and predicted assembly

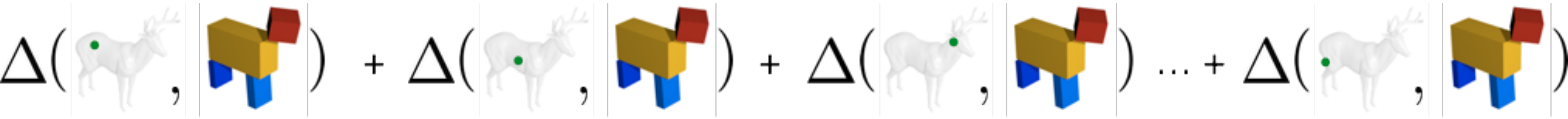


Each point is a **linear function** of mesh/primitive vertex coordinates

Differentiable!

Loss function construction

Sample points on the groundtruth mesh and predicted assembly

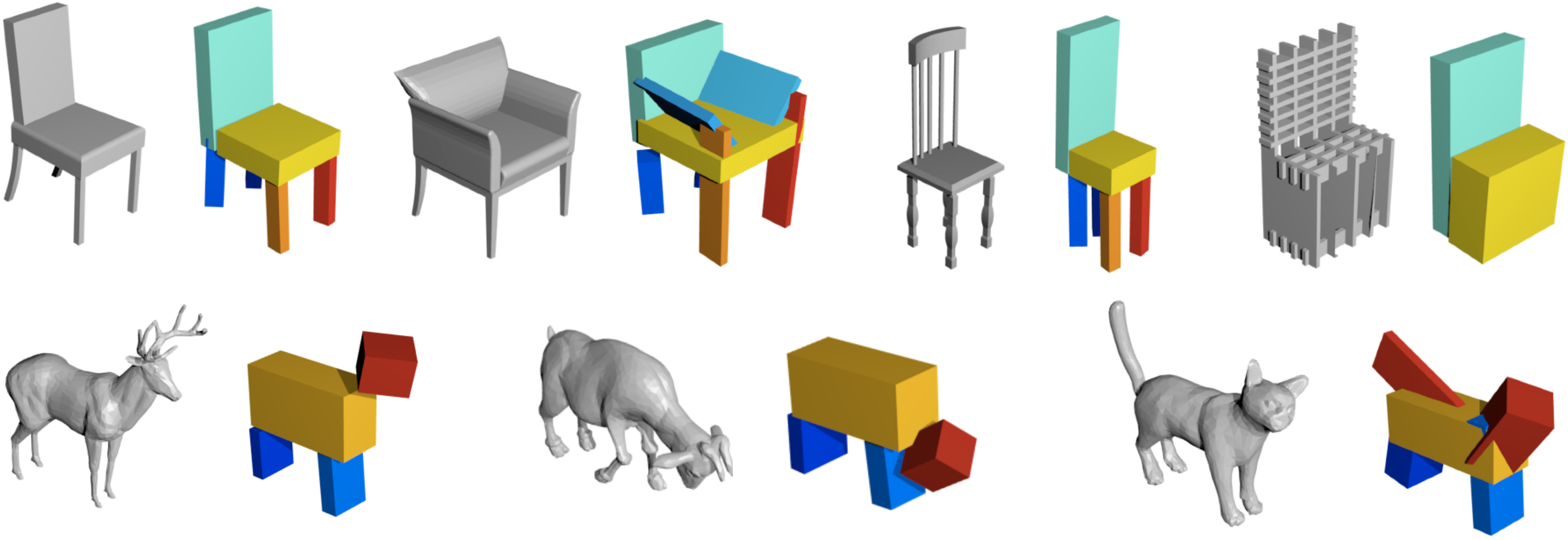


Each point is a **linear function** of mesh/primitive vertex coordinates

Differentiable!

Speed up the computation leveraging parameterization of primitives

Consistent primitive configurations



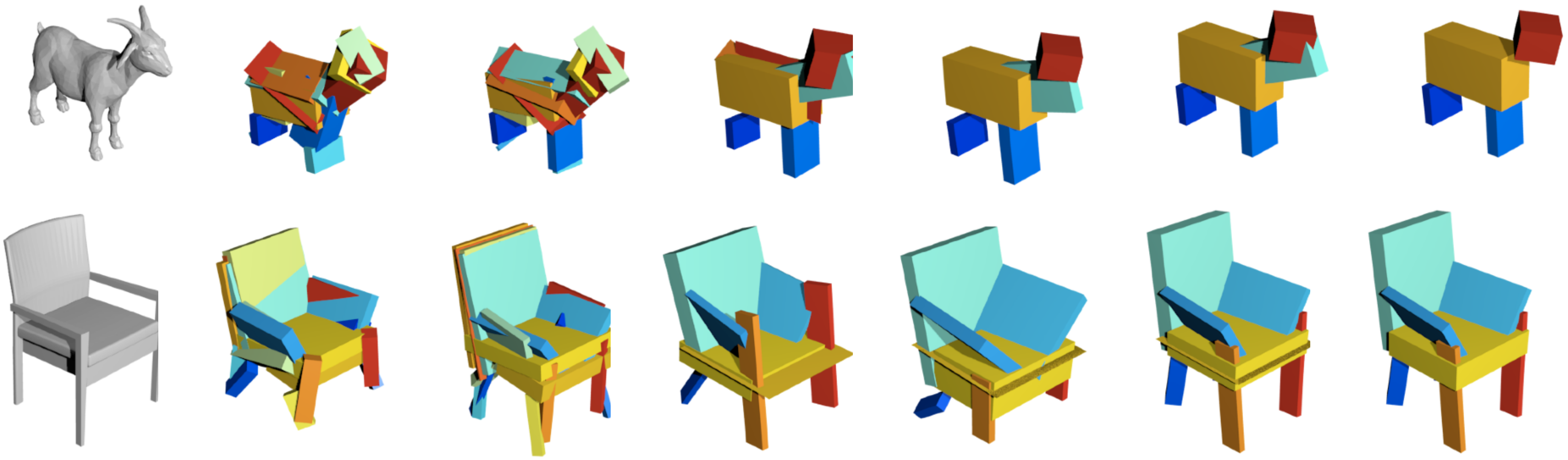
Primitive locations are **consistent** due to
the **smoothness** of primitive prediction network

Unsupervised parsing

Method	[31] (initial)	[31] (refined)	Ours
Accuracy	78.6	84.8	89.0

Mean accuracy (face area) on Shape COSEG chairs.

Analysis



Shapes become more parsimonious as training progresses (due to our parsimony reward)

Image-based modeling

Open problems

How to introduce other primitives types?

Towards image based modeling, how to add more operations to edit those primitives?

- e.g., Deform? Extrude? Loop cut?

How to use it for design purposes? For example, with certain structural and functional constraints.

Ultimately, we expect to automate the modeling process from images, as artists do.

Resources

- For both works, paper and source codes are available.

- Haoqiang Fan*, Hao Su*, Leonidas Guibas, *A Point Set Generation Network for 3D Object Reconstruction from a Single Image*, CVPR2017 (oral)

- Shubham Tulsiani, Hao Su, Leonidas Guibas, Alexei Efros, Jitendra Malik, *Learning Shape Abstractions by Assembling Volumetric Primitives*, CVPR2017 (oral)

To sum up

We explore to generate geometric representations by neural networks

Point cloud based reconstruction has better quality than state-of-the-art volumetric shape generators

Primitive-based CAD models can be generated to abstract polygonal meshes in an unsupervised manner

Keys:

- network structure leveraging geometric natural statistics
- loss function design