UC San Diego

Deep Learning on Extrinsic Geometry

Instructor: Hao Su

slides credits: Justin Solomon, Chengcheng Tang

Dues

Thu of 3rd week (week of Jan 20): Announcement of projects and start to form project teams

Tue of 4th week (week of Jan 27): Due of casting votes on projects for each team

Thu of 4th week (week of Jan 27): Announcement of project-group alignment

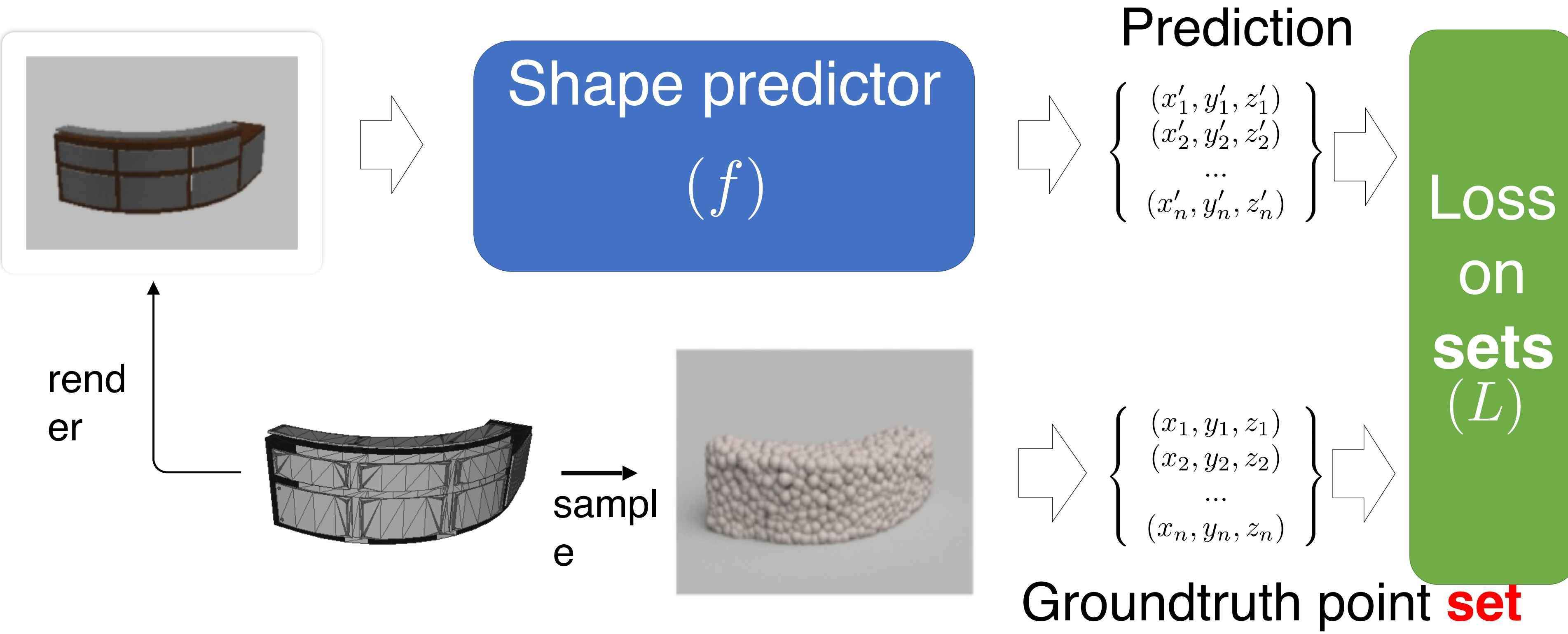
Thu of 5th week (week of Feb 3): Work plan (1 page, template provided)

Thu of 8th week (week of Feb 24): Mid-term report (3 pages, template provided)

Tue/Thu of 10th week (week of Mar 9): Final presentation (15 minutes for each team)

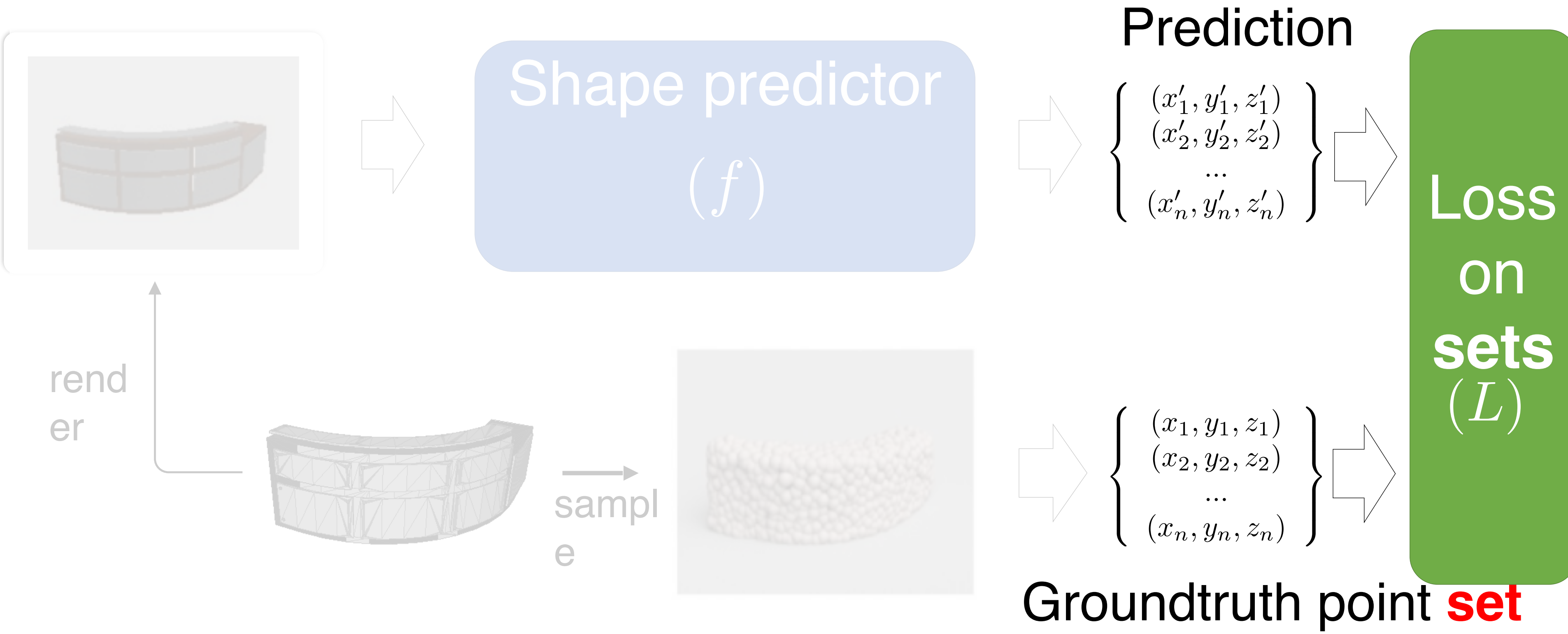
Thu of 11th week (week Mar 16): Final report write-up (6 pages, template provided)

Pipeline



CVPR '17, Point Set Generation

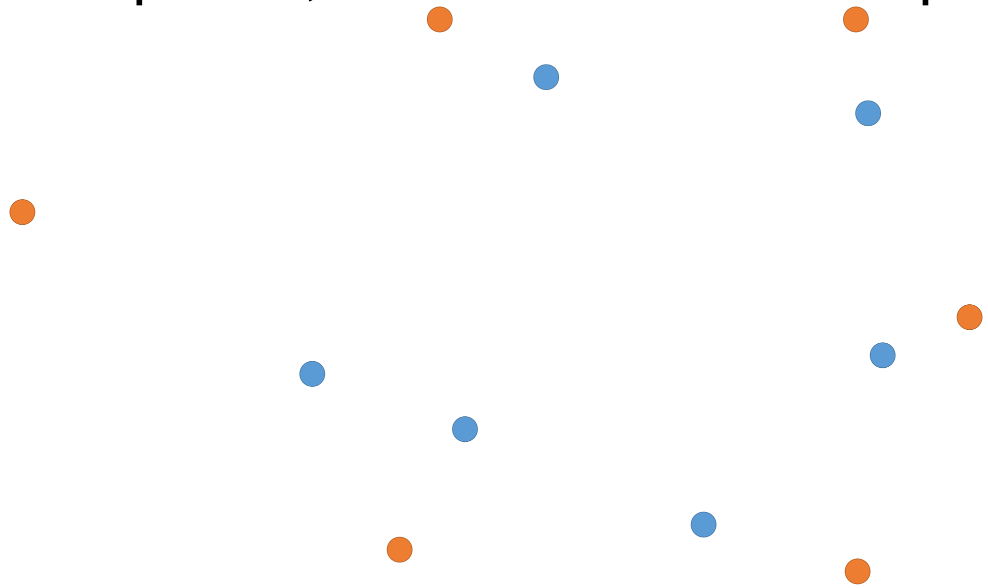
Pipeline



CVPR '17, Point Set Generation

Set comparison

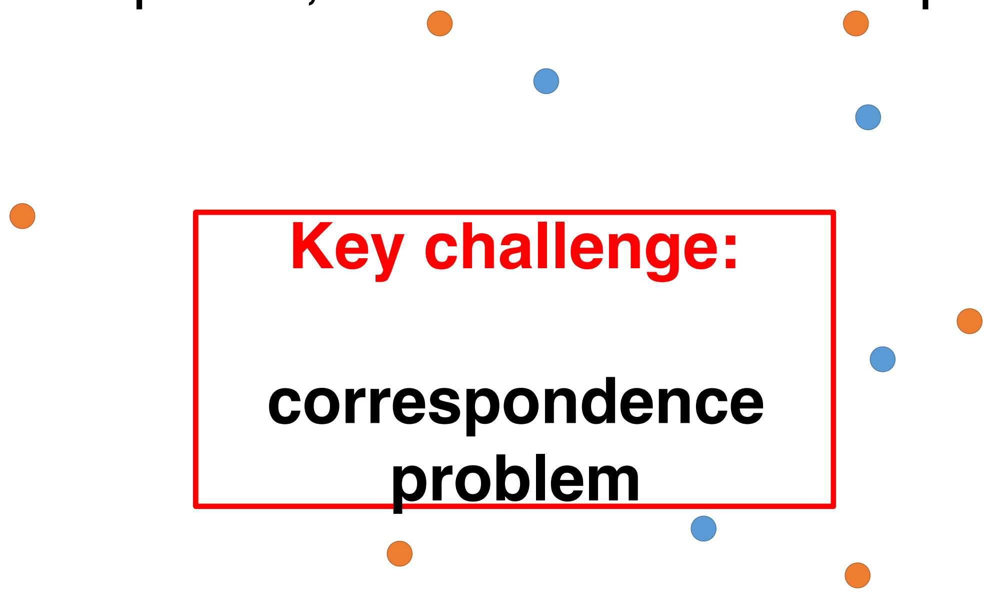
Given two sets of points, measure their discrepancy



CVPR '17, Point Set Generation

Set comparison

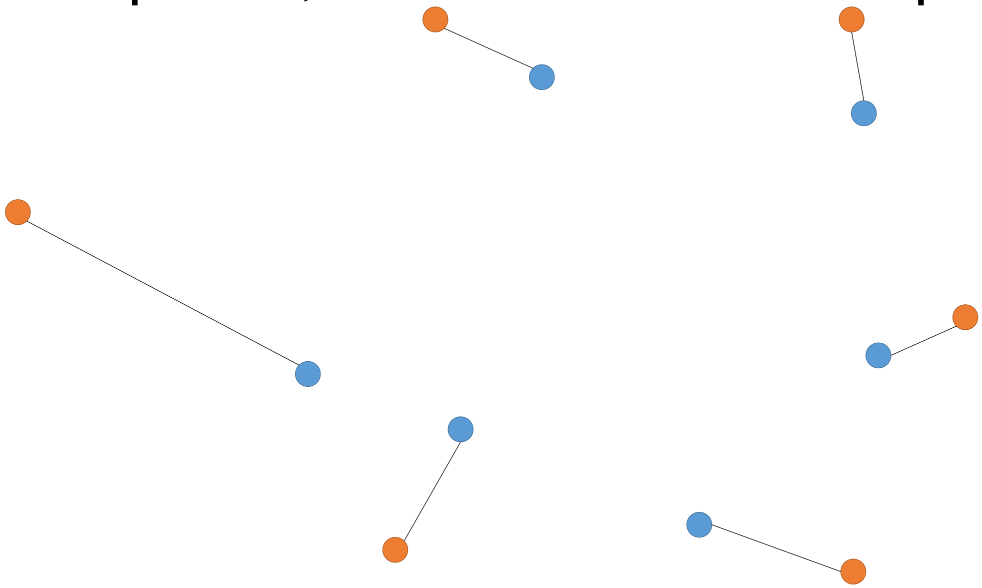
Given two sets of points, measure their discrepancy



CVPR '17, Point Set Generation

Correspondence (I): optimal assignment

Given two sets of points, measure their discrepancy



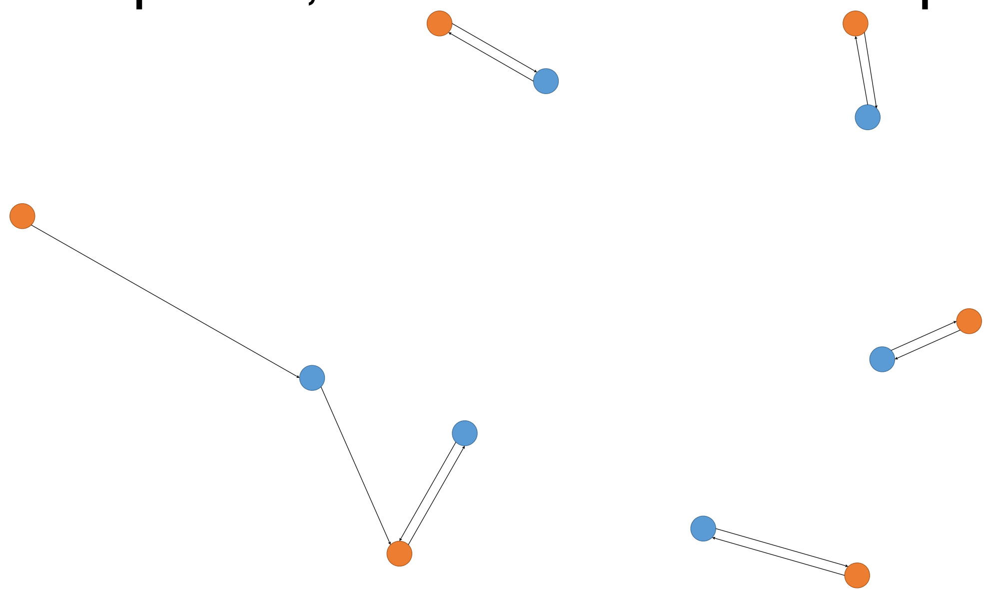
a.k.a Earth Mover's distance (EMD)

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

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

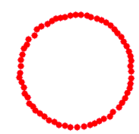
Distance metrics affect mean shapes

The mean shape carries characteristics of the distance metric

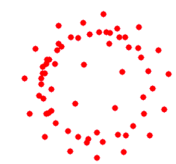
$$\bar{x} = \underset{x}{\operatorname{argmin}}\ \mathbb{E}_{s \sim \mathbb{S}}[d(x, s)]$$

continuous hidden variable (radius)

Input



EMD mean



Chamfer mean

CVPR '17, Point Set Generation

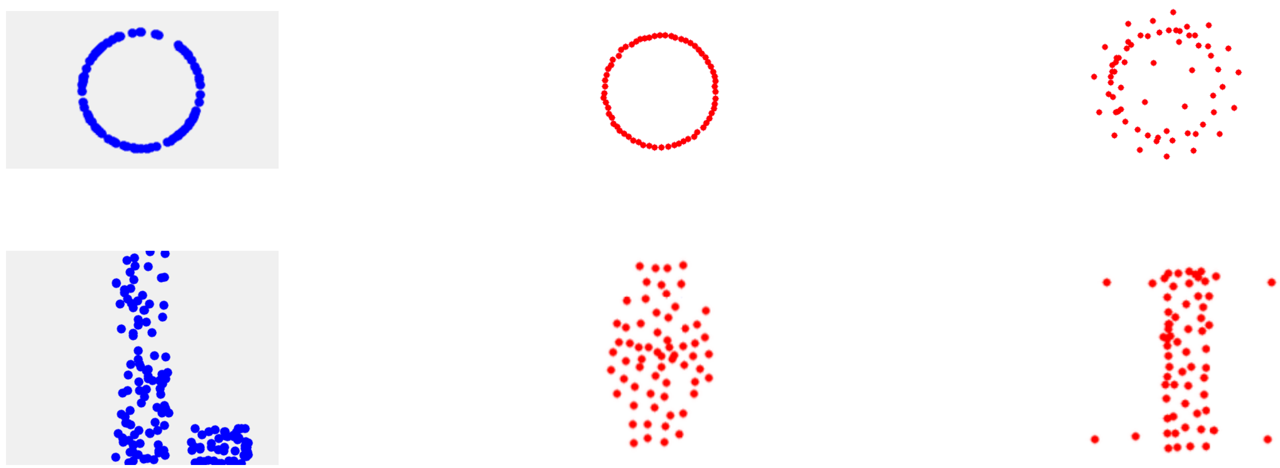
Mean shapes from distance metrics

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)

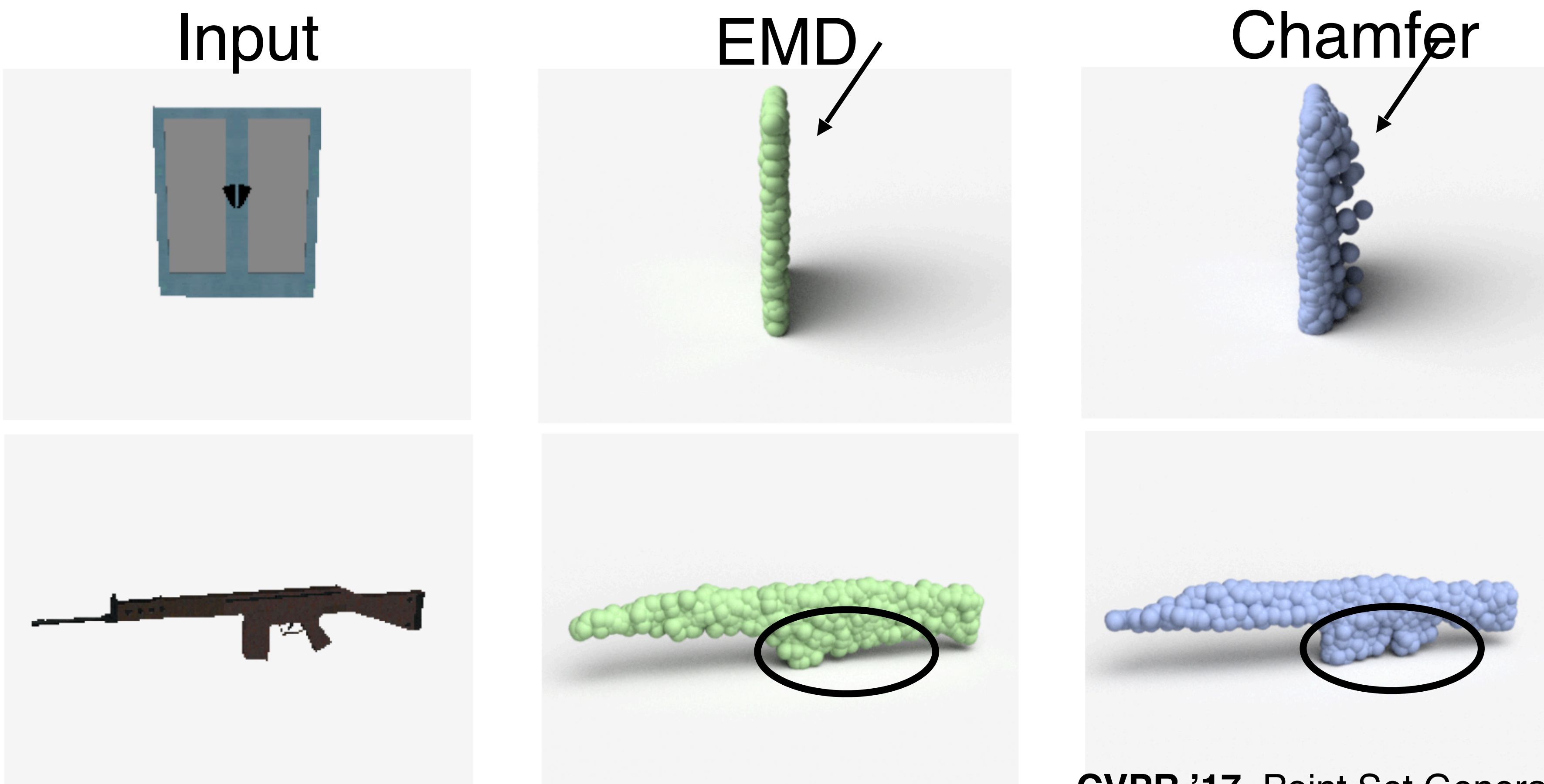
discrete hidden variable (add-on location)



Input | EMD mean | Chamfer mean

CVPR '17, Point Set Generation

Comparison of predictions by EMD versus CD



CVPR '17, Point Set Generation

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.**!

Co **ere**

CVPR '17, Point Set Generation

Computational requirement of metrics

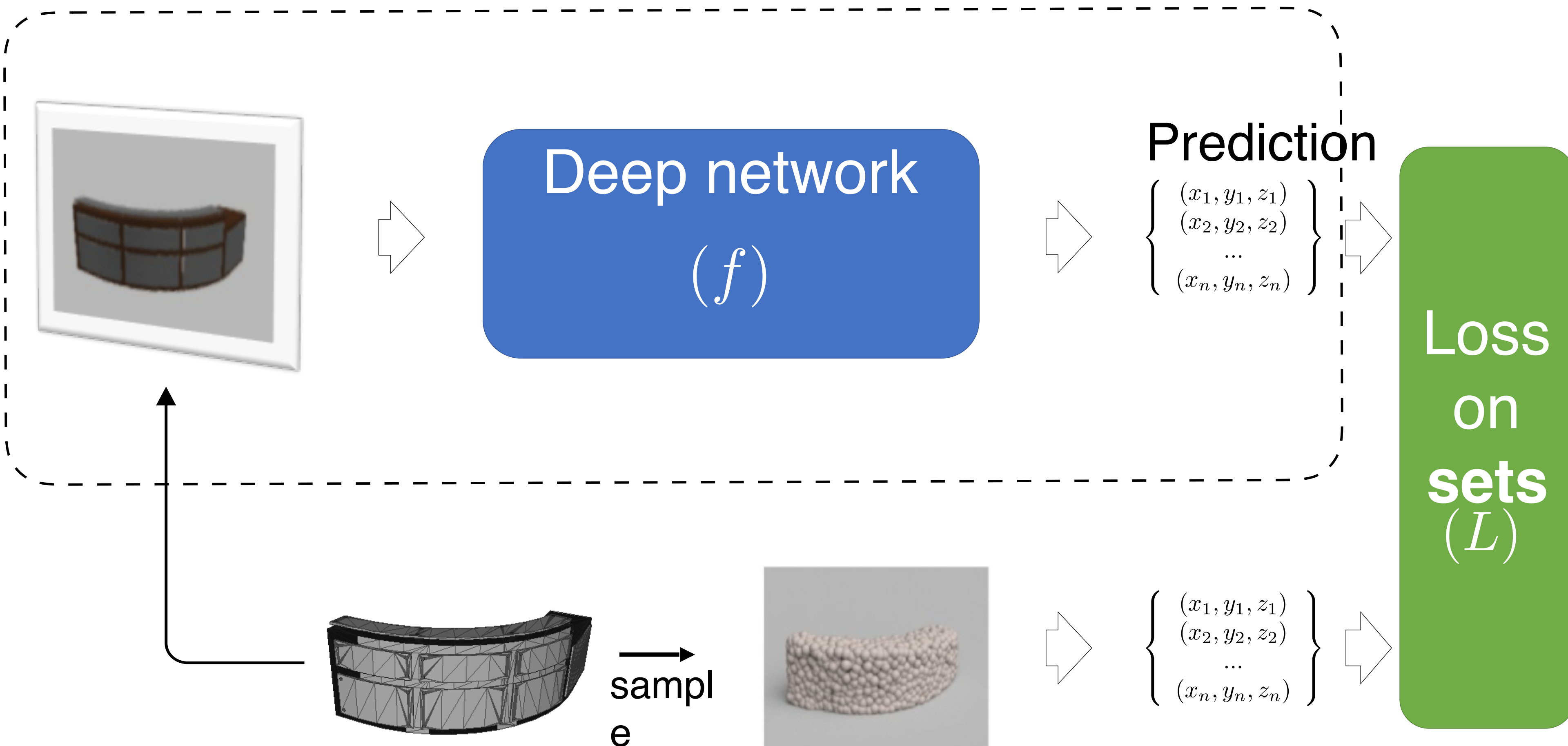
- **Efficient** to compute

Chamfer distance: trivially parallelizable on CUDA

Earth Mover's distance (optimal assignment):

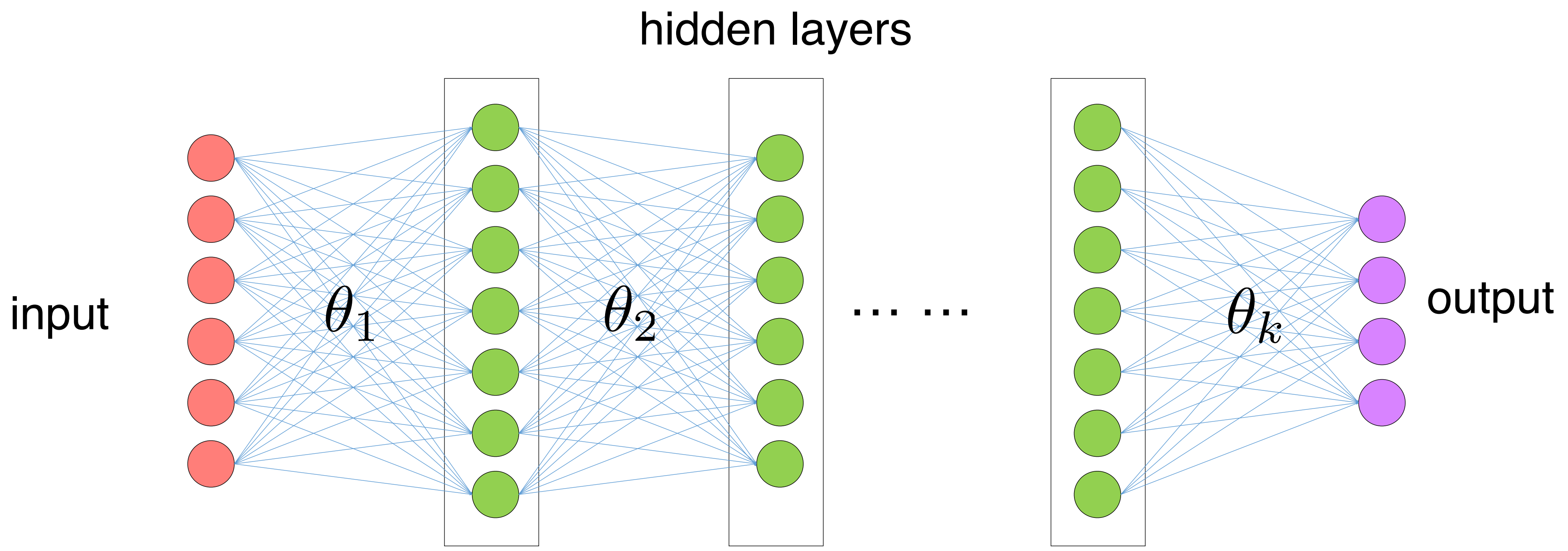
- We implement a **distributed** approximation algorithm on CUDA
- Based upon [Bertsekas, 1985], $(1+\epsilon)$-approximation

Pipeline



CVPR '17, Point Set Generation

Deep neural network

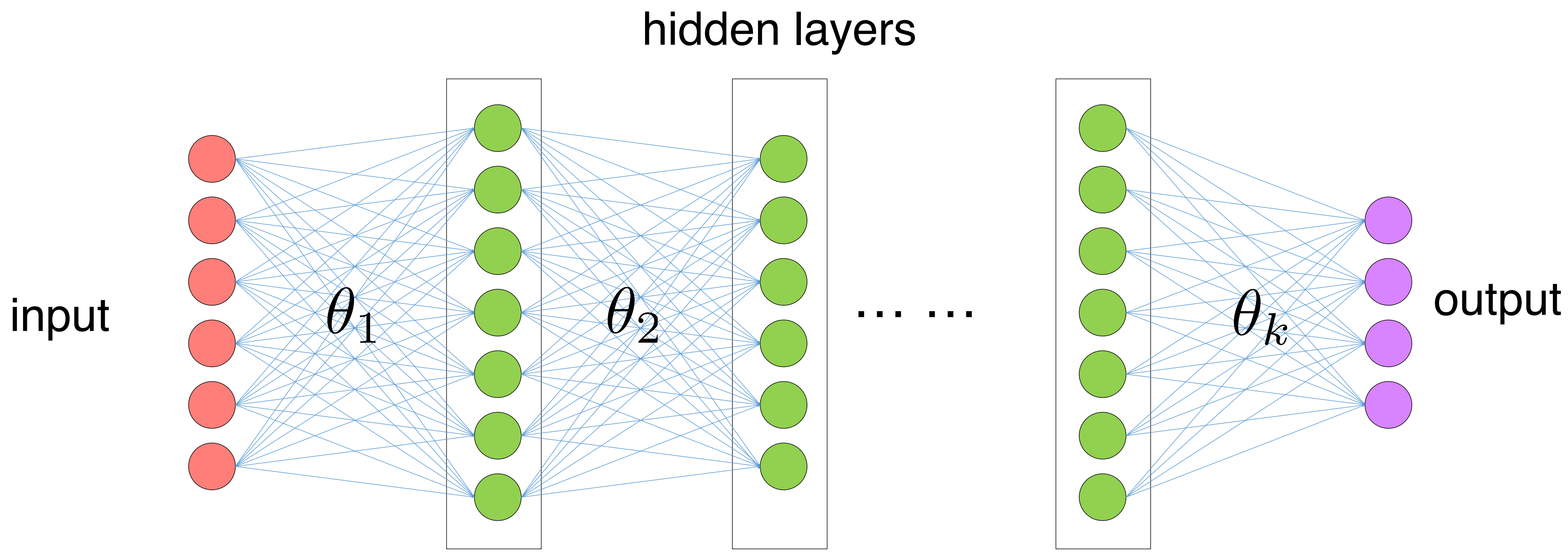


Universal function approximator

- A cascade of layers

CVPR '17, Point Set Generation

Deep neural network

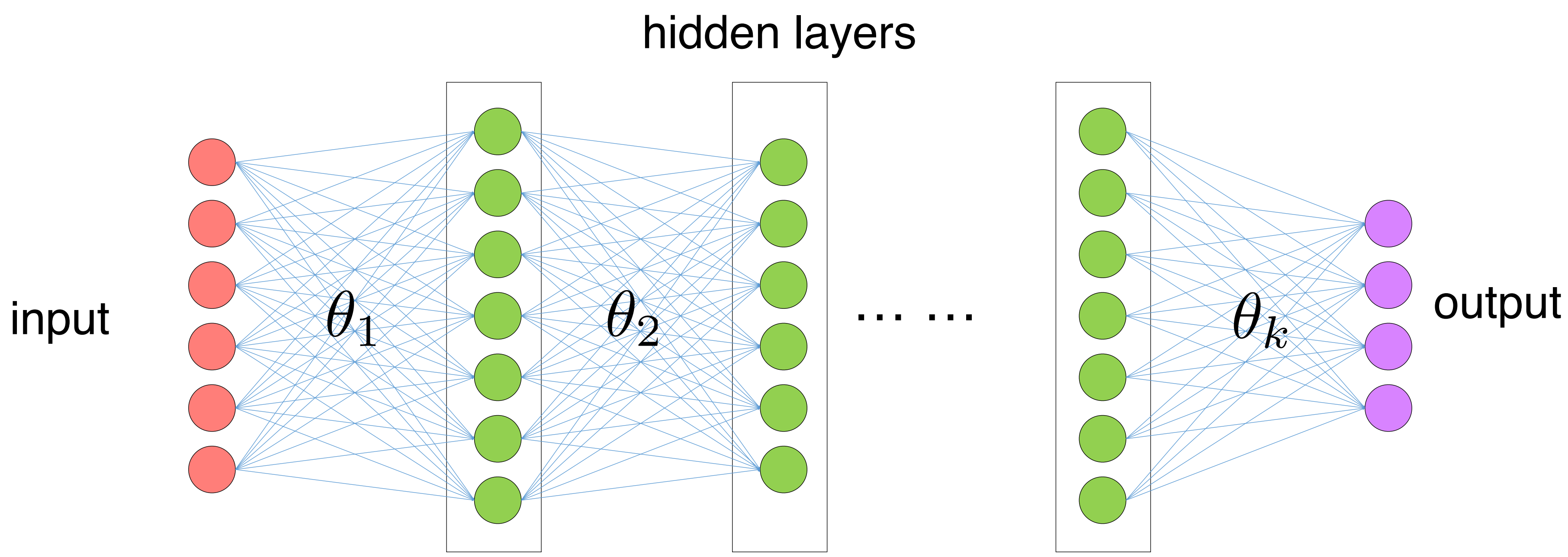


Universal function approximator

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

CVPR '17, Point Set Generation

Deep neural network

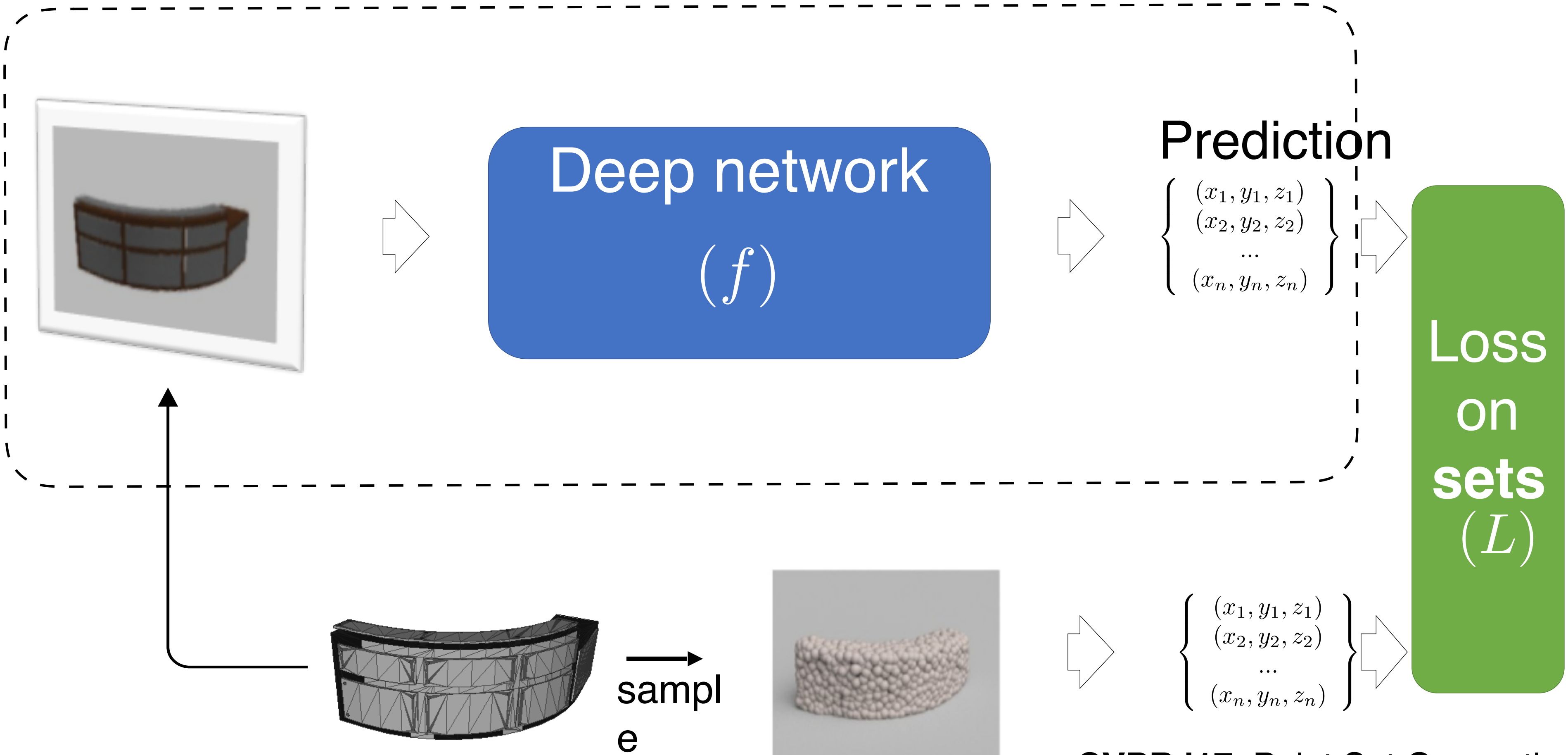


Universal function approximator

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

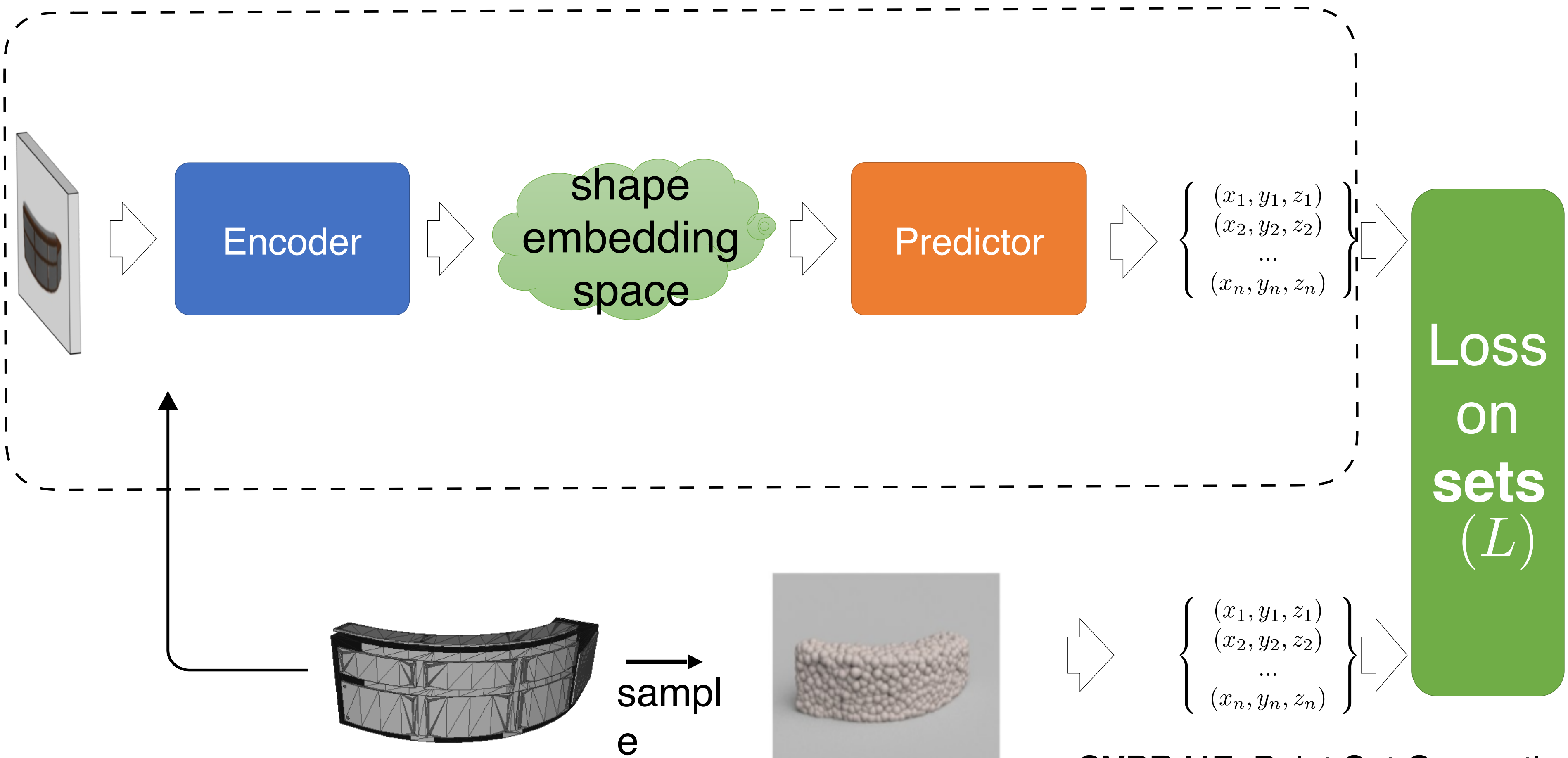
CVPR '17, Point Set Generation

Pipeline



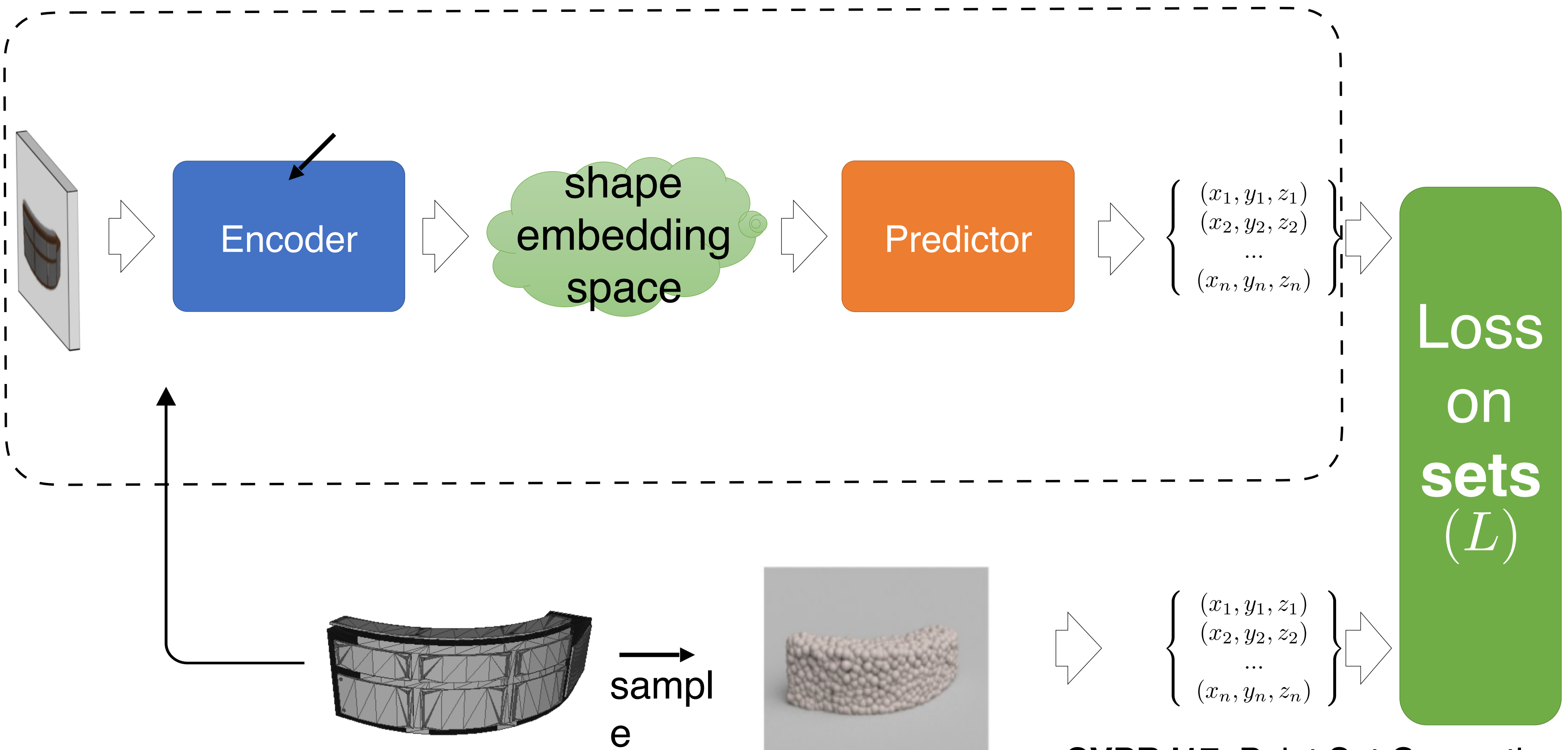
CVPR '17, Point Set Generation

Pipeline



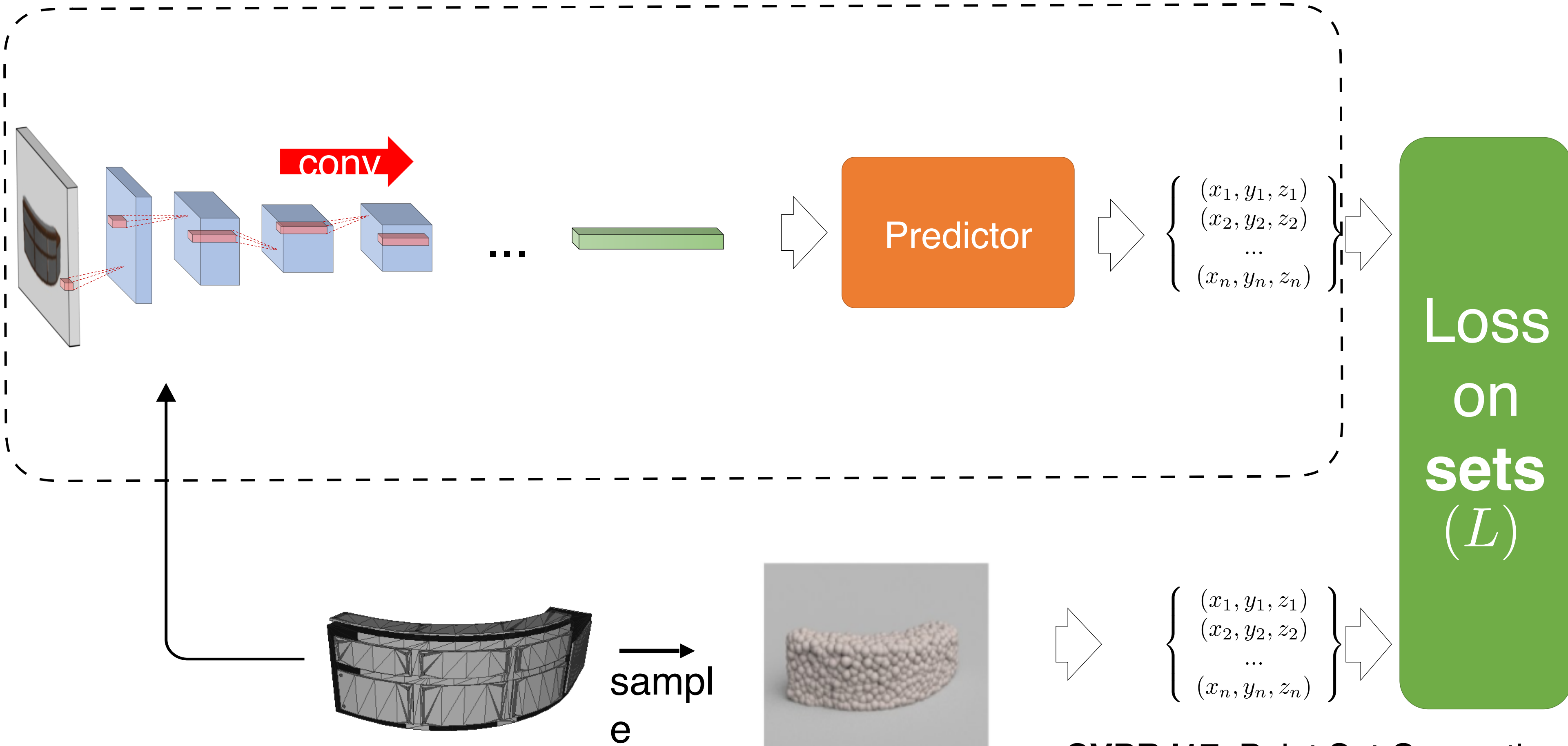
CVPR '17, Point Set Generation

Pipeline



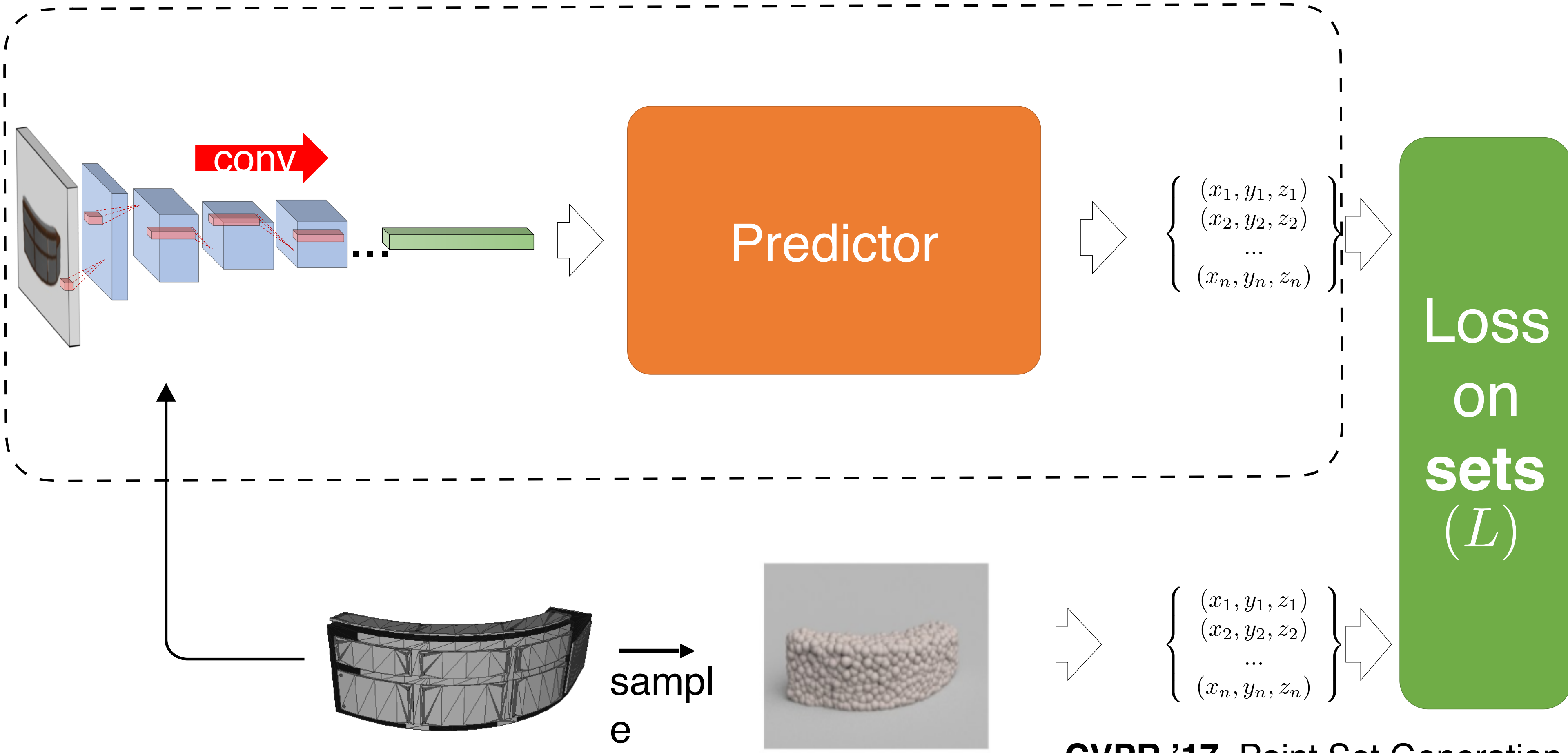
CVPR '17, Point Set Generation

Pipeline



CVPR '17, Point Set Generation

Pipeline



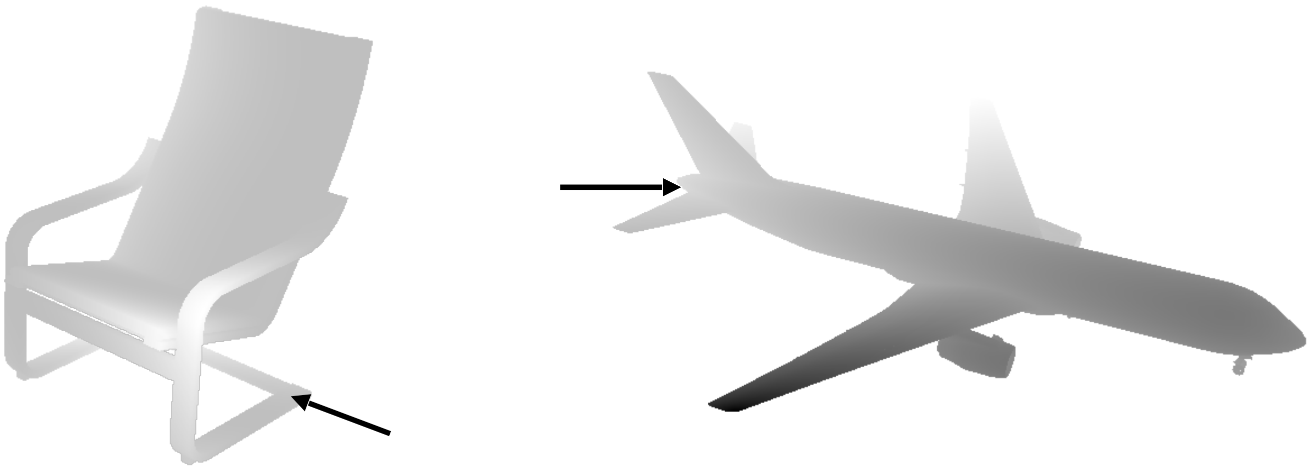
CVPR '17, Point Set Generation

Natural statistics of geometry

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

CVPR '17, Point Set Generation

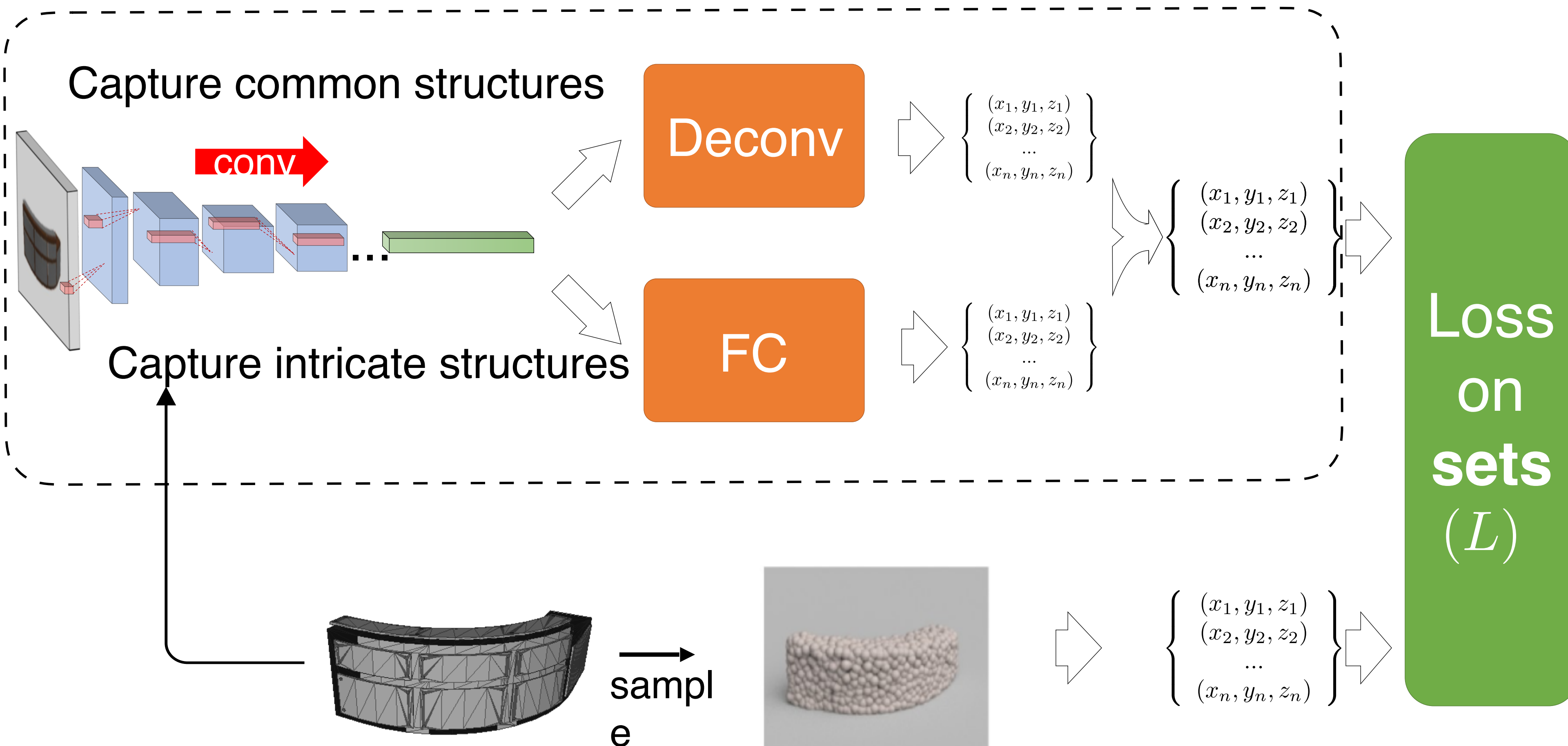
Natural statistics of geometry



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

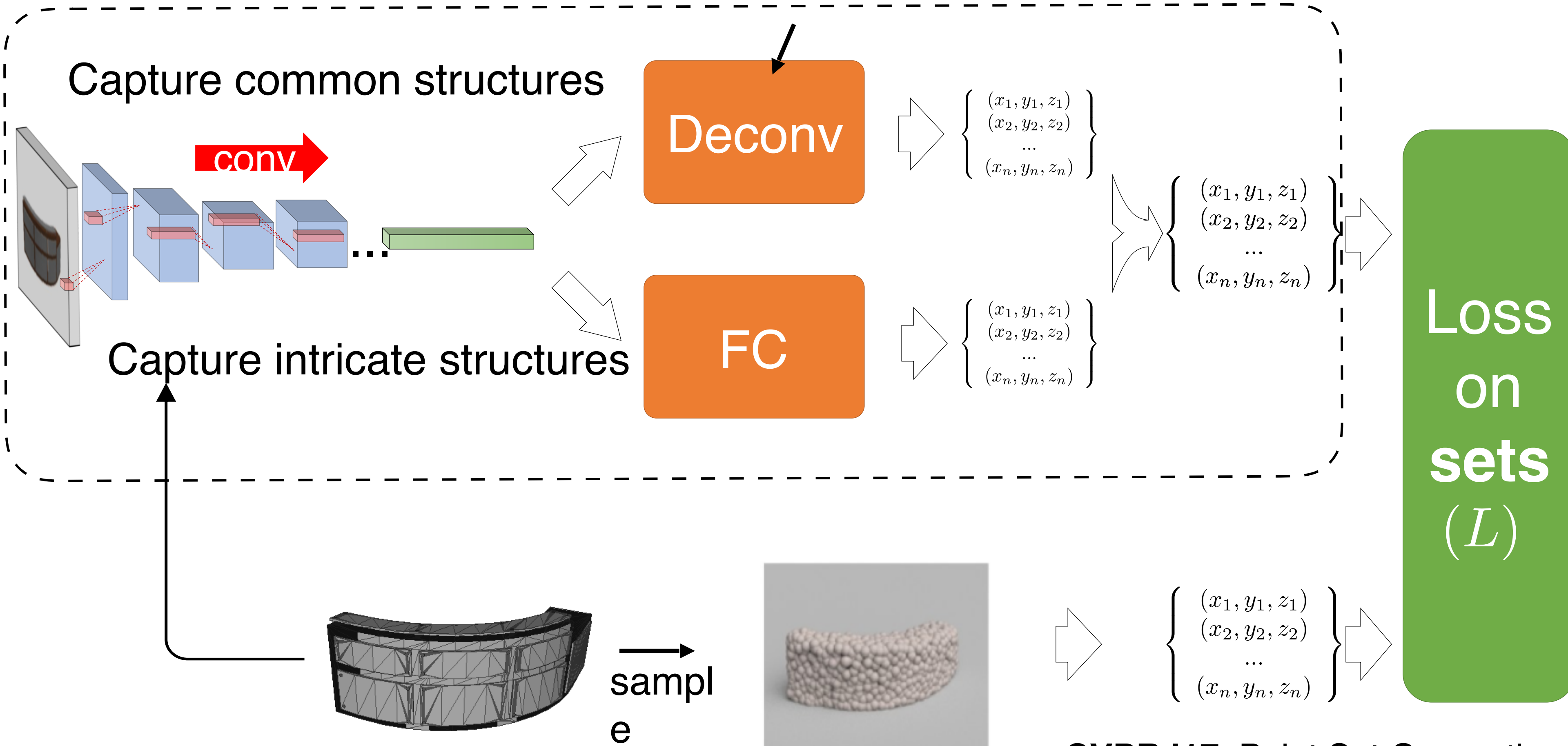
CVPR '17, Point Set Generation

Pipeline



CVPR '17, Point Set Generation

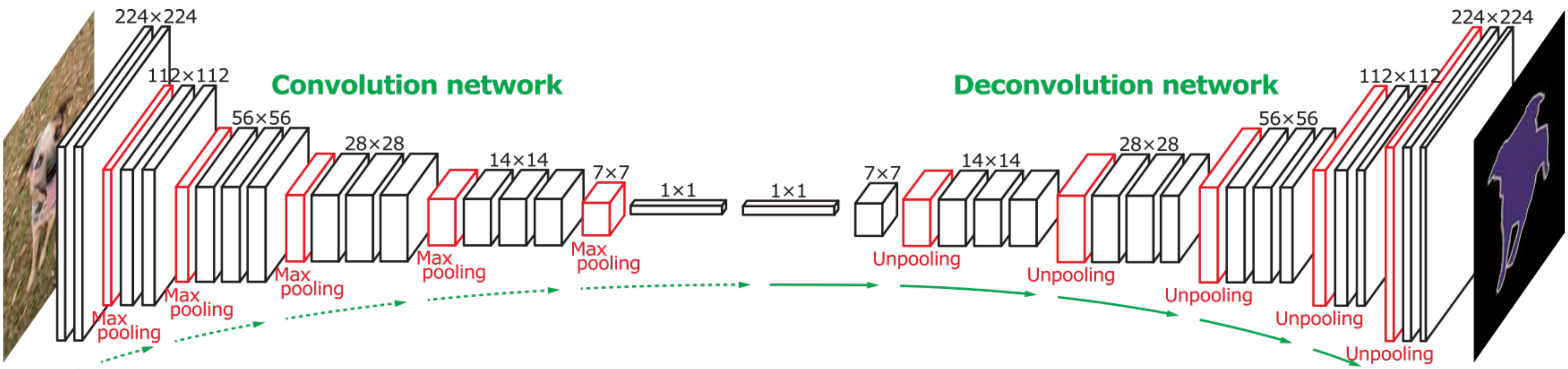
Pipeline



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

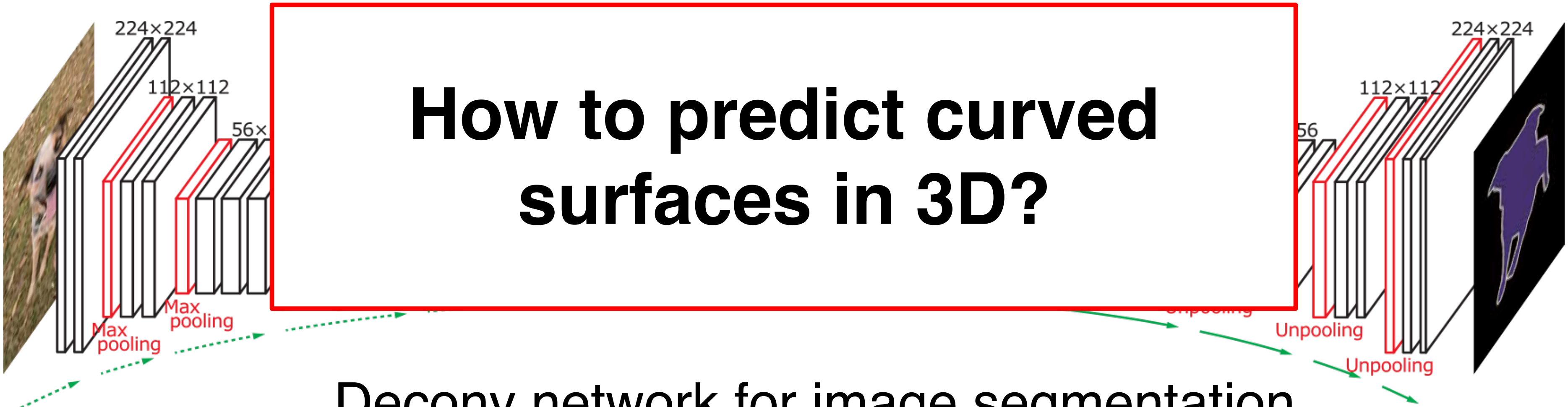


Deconv network for image segmentation

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?

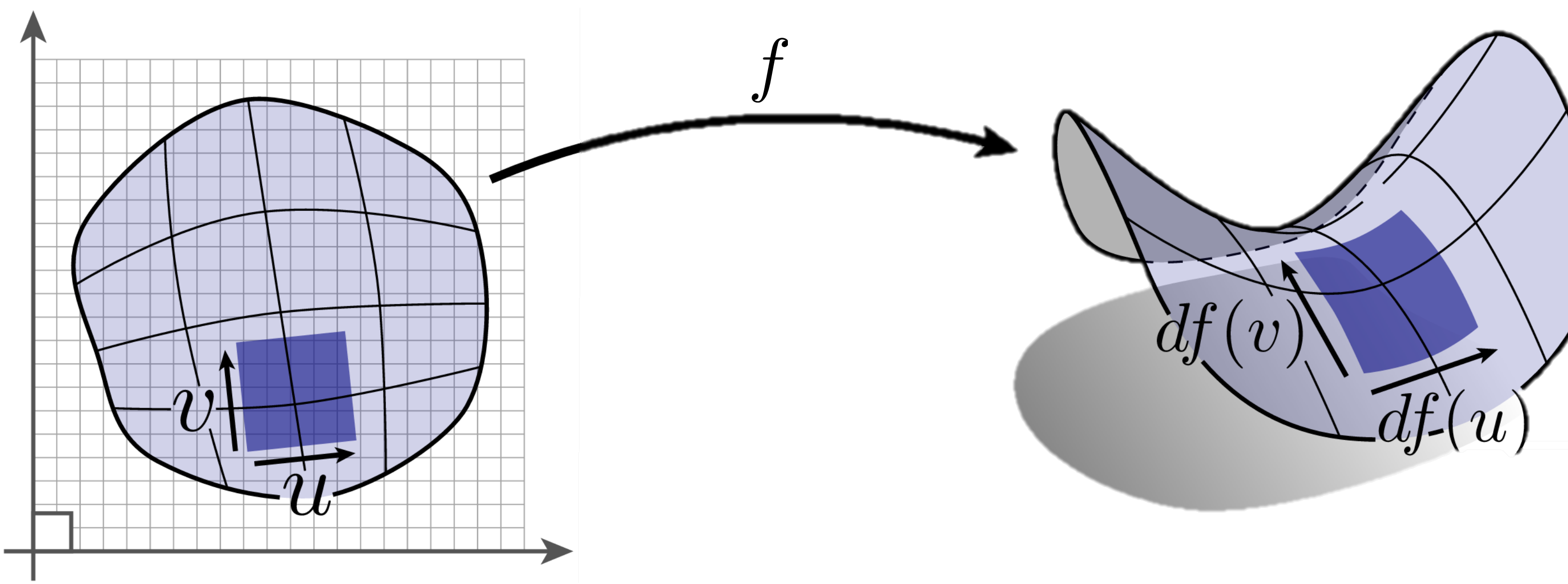
Deconv network for image segmentation

Credit: FCNN, Long et al.

Prediction of curved 2D surfaces in 3D

- Surface parametrization (2D   3D mapping)

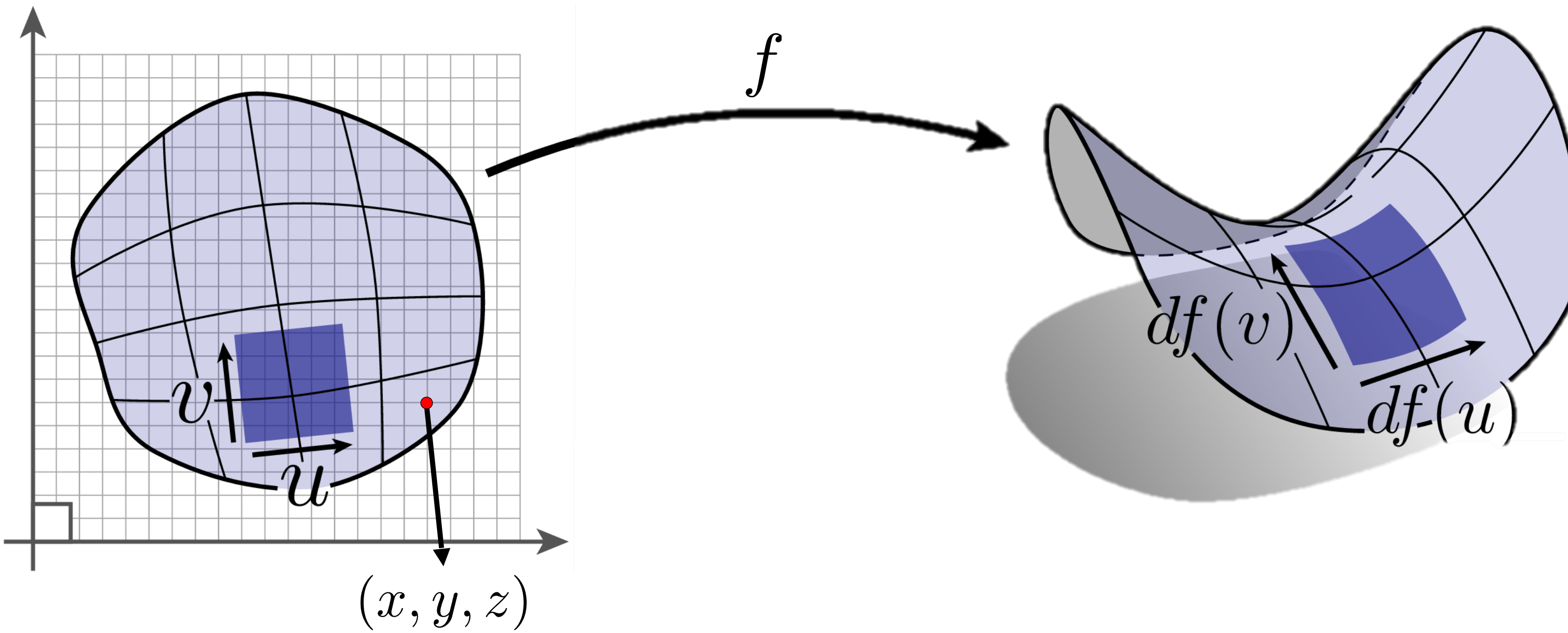
$\leftrightarrow$



Credit: Discrete Differential Geometry, Crane et al.

Prediction of curved 2D surfaces in 3D

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

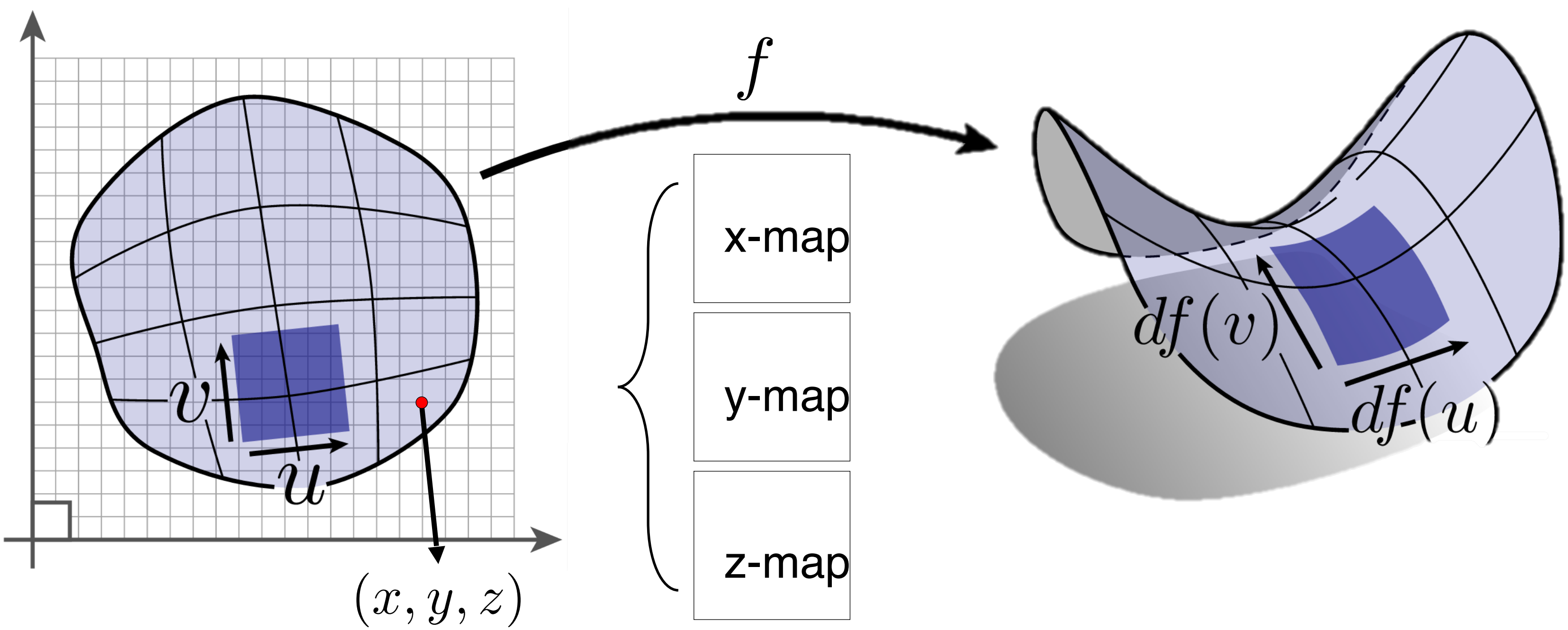


Credit: Discrete Differential Geometry, Crane et al.

Prediction of curved 2D surfaces in 3D

- Surface parametrization (2D-3D mapping)

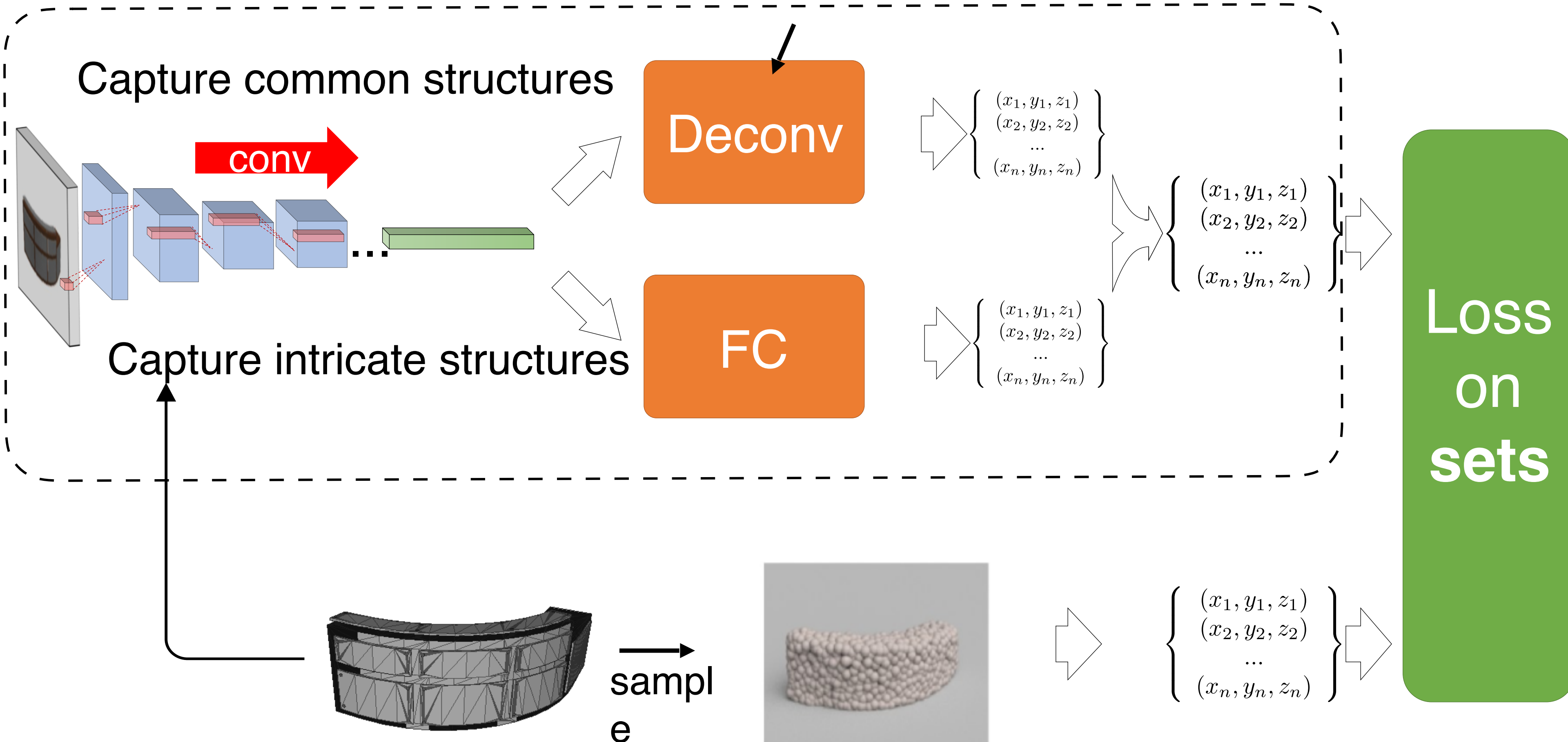
$\leftrightarrow$



coordinate maps

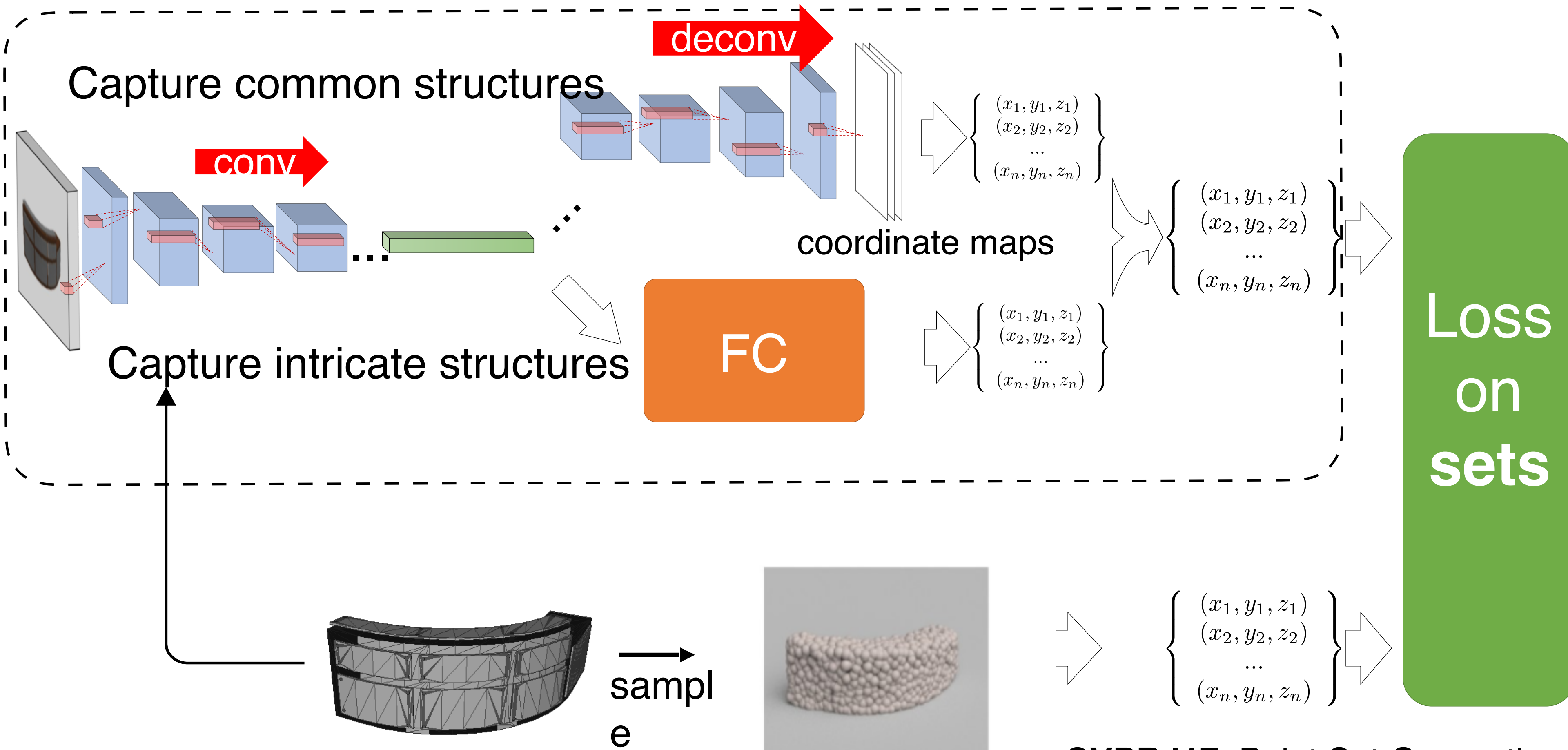
Credit: Discrete Differential Geometry, Crane et al.

Parametrization prediction by deconv network



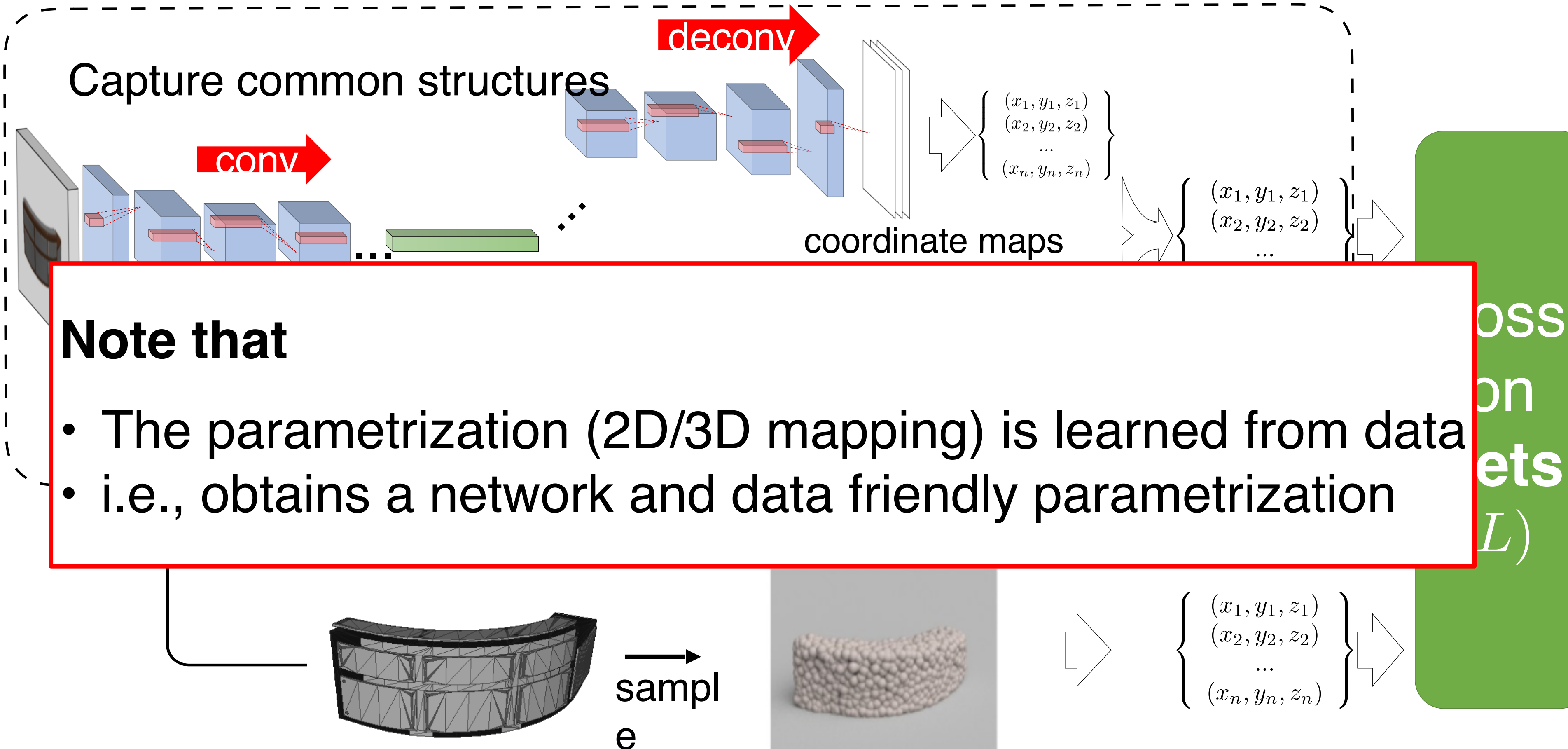
CVPR '17, Point Set Generation

Parametrization prediction by deconv network



CVPR '17, Point Set Generation

Parametrization prediction by deconv network



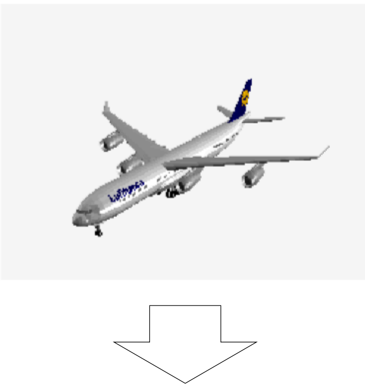
Note that

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

CVPR ’17, Point Set Generation

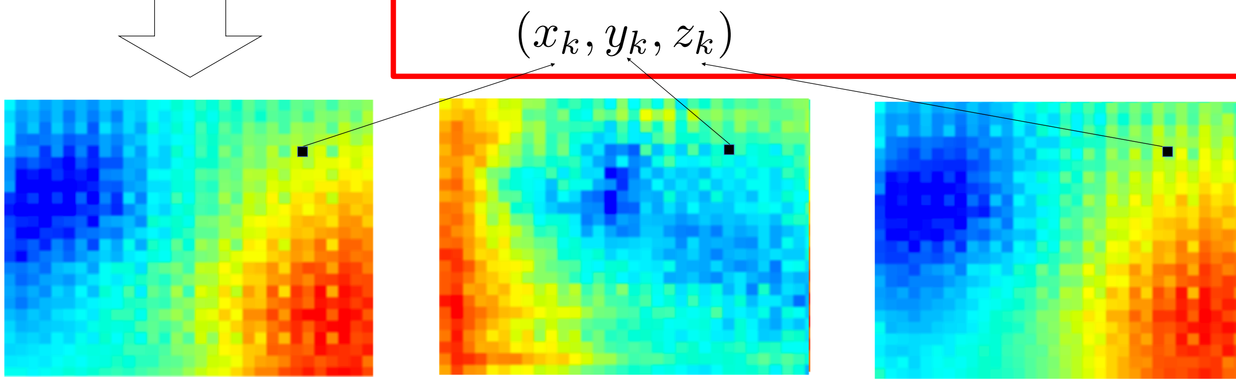
Visualization of the learned parameterization

- Surface parametrization (2D 3D mapping)



Observation:

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



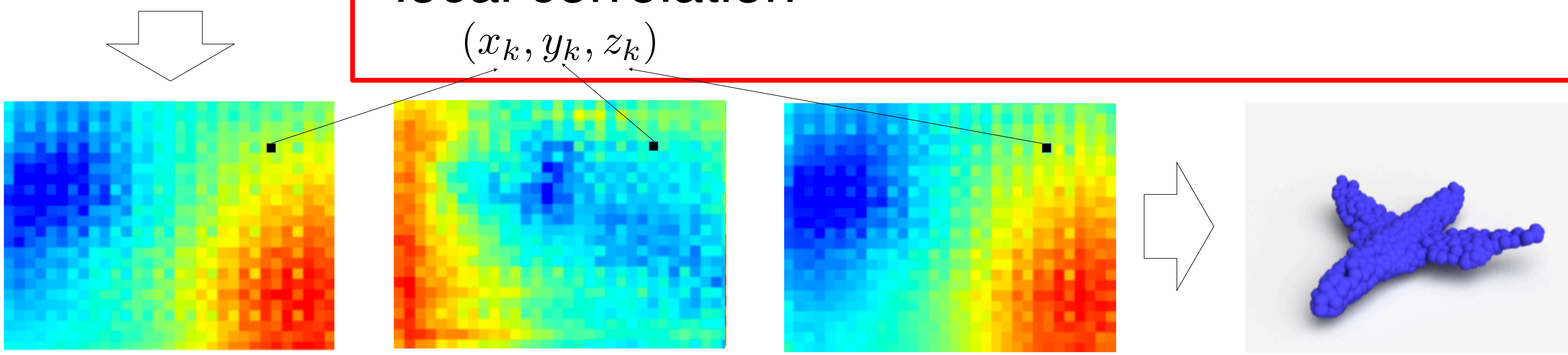
map of x coord map of y coord map of z coord

Visualization of the learned parameterization

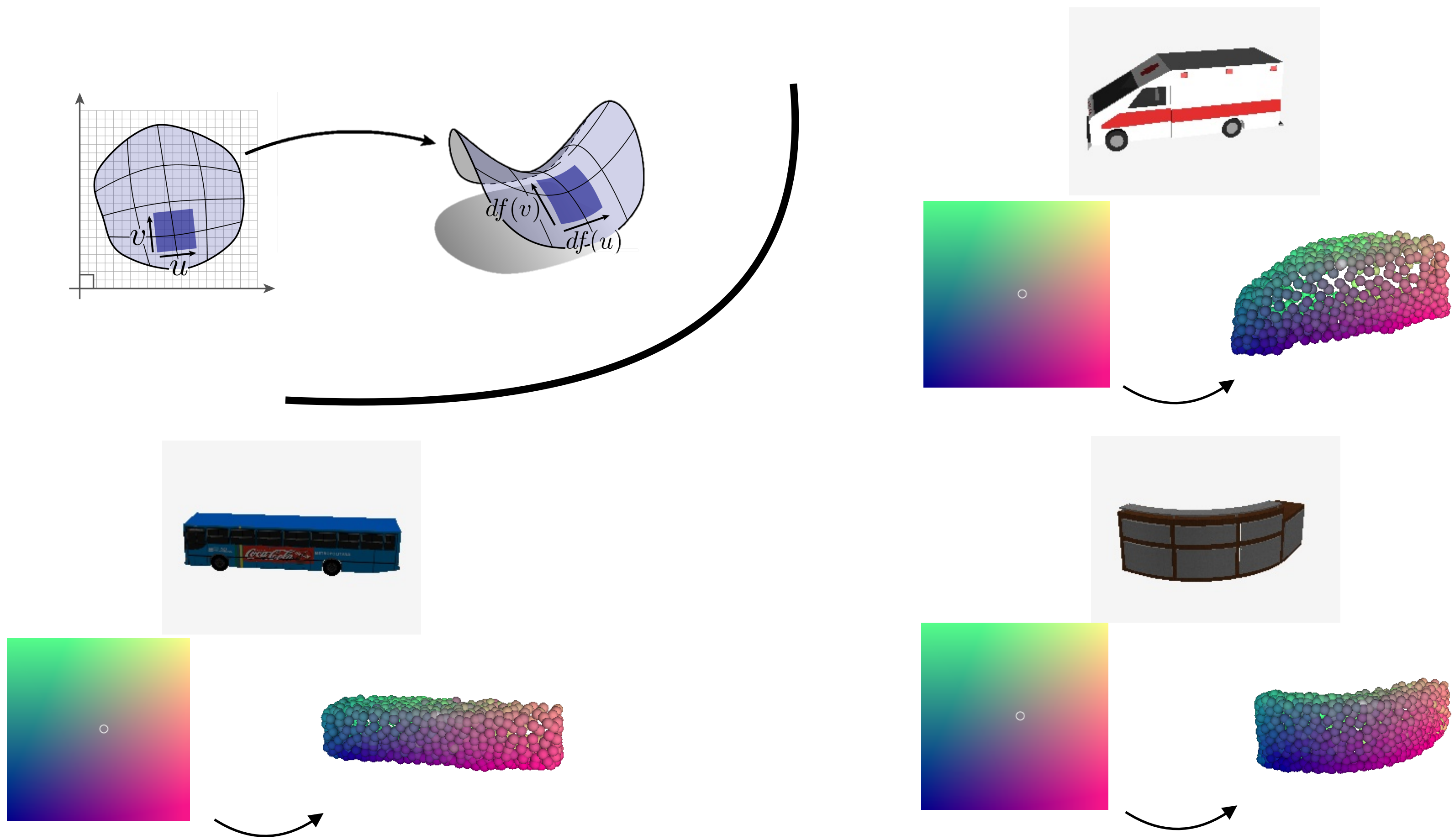
- Surface parametrization (2D 3D mapping)

Observation:

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

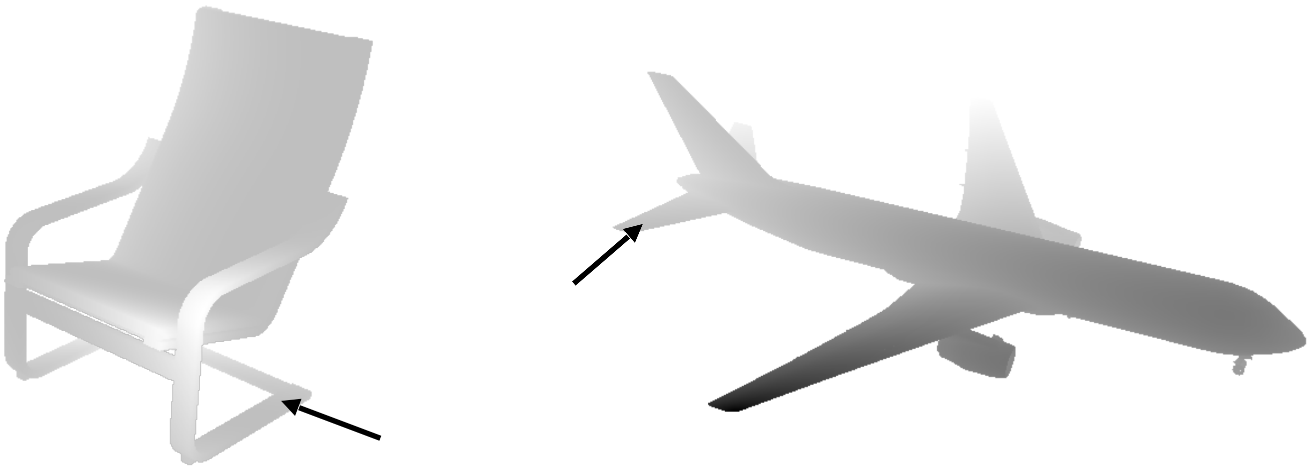


map of x coord map of y coord map of z coord



v
u
df(v)
df(u)

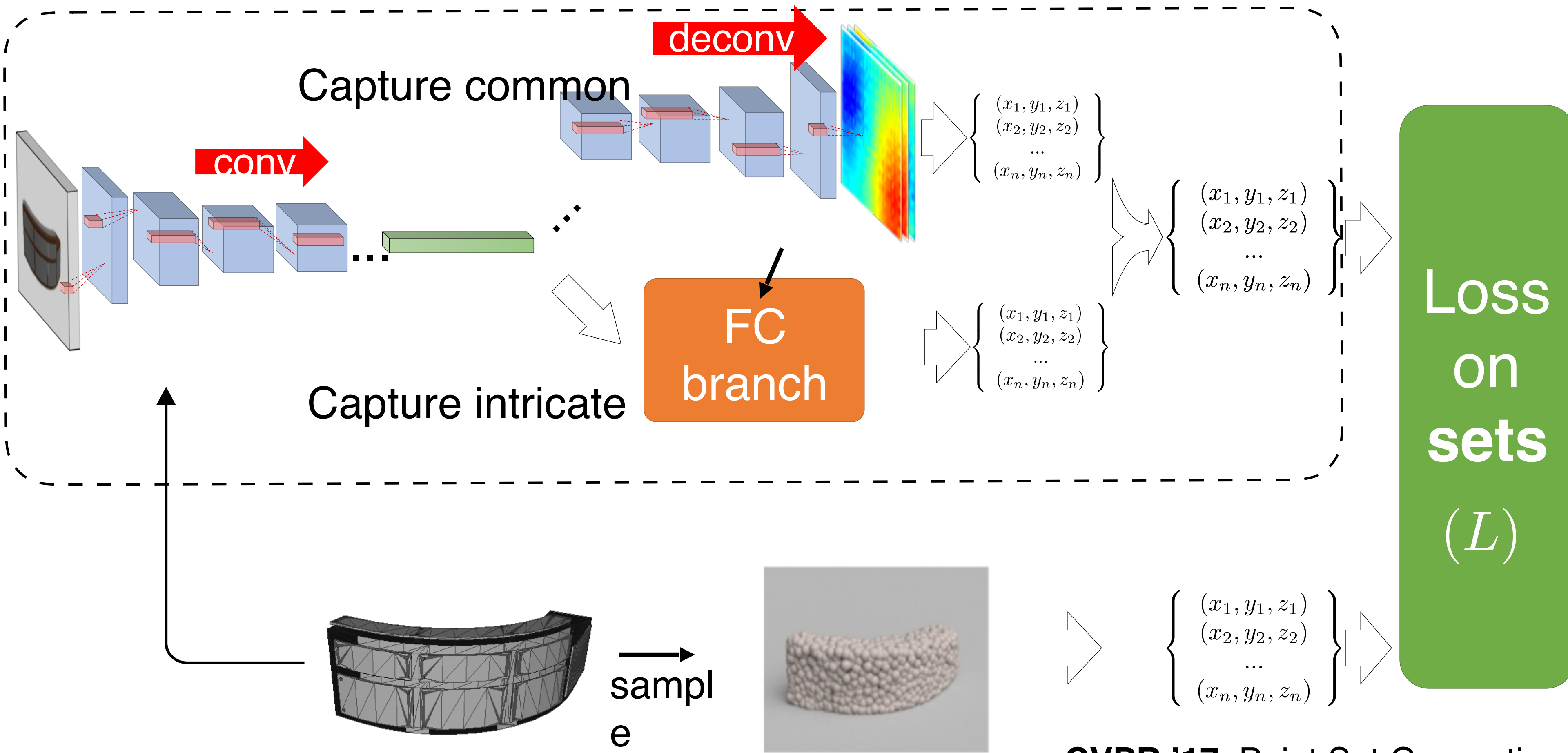
Natural statistics of geometry



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

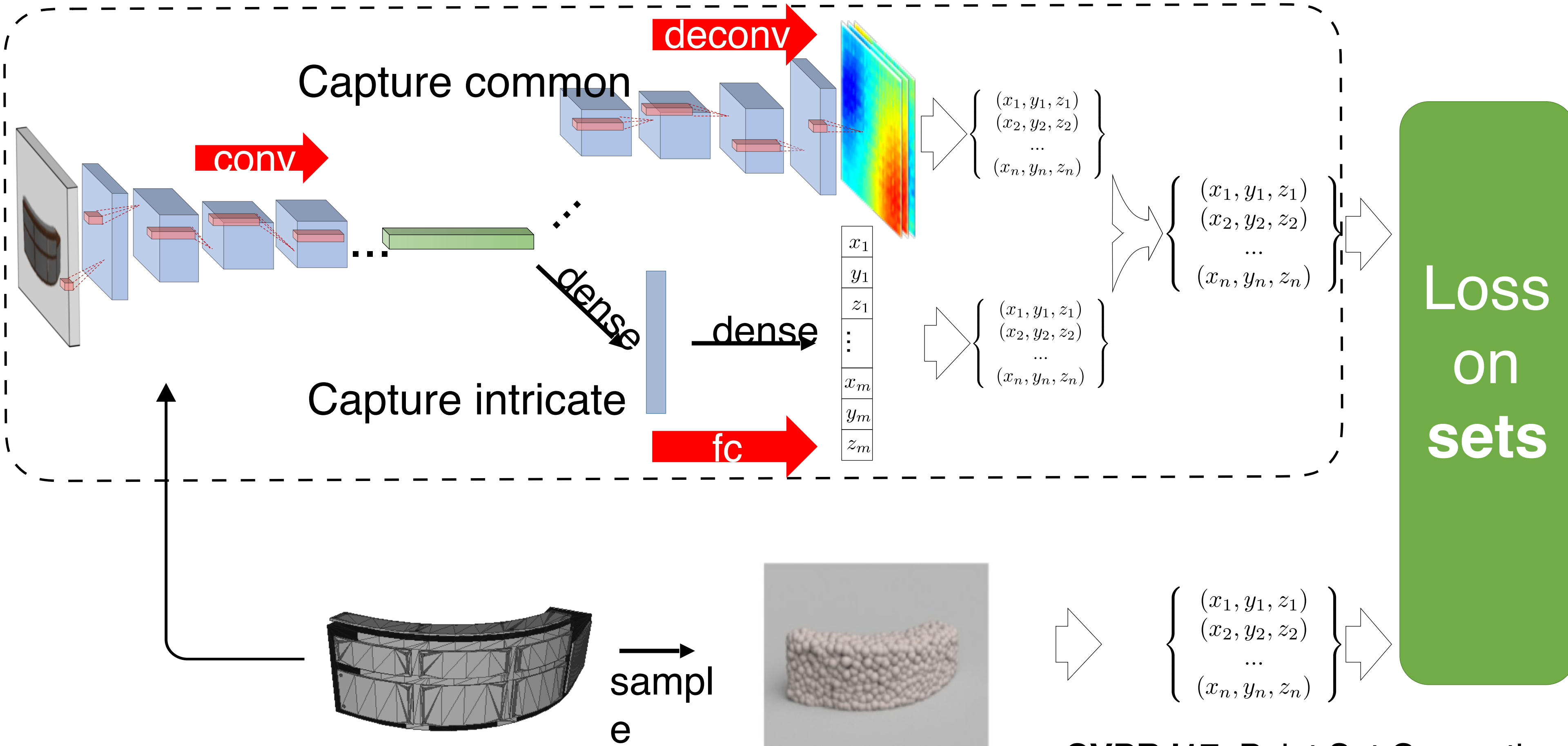
CVPR '17, Point Set Generation

Pipeline



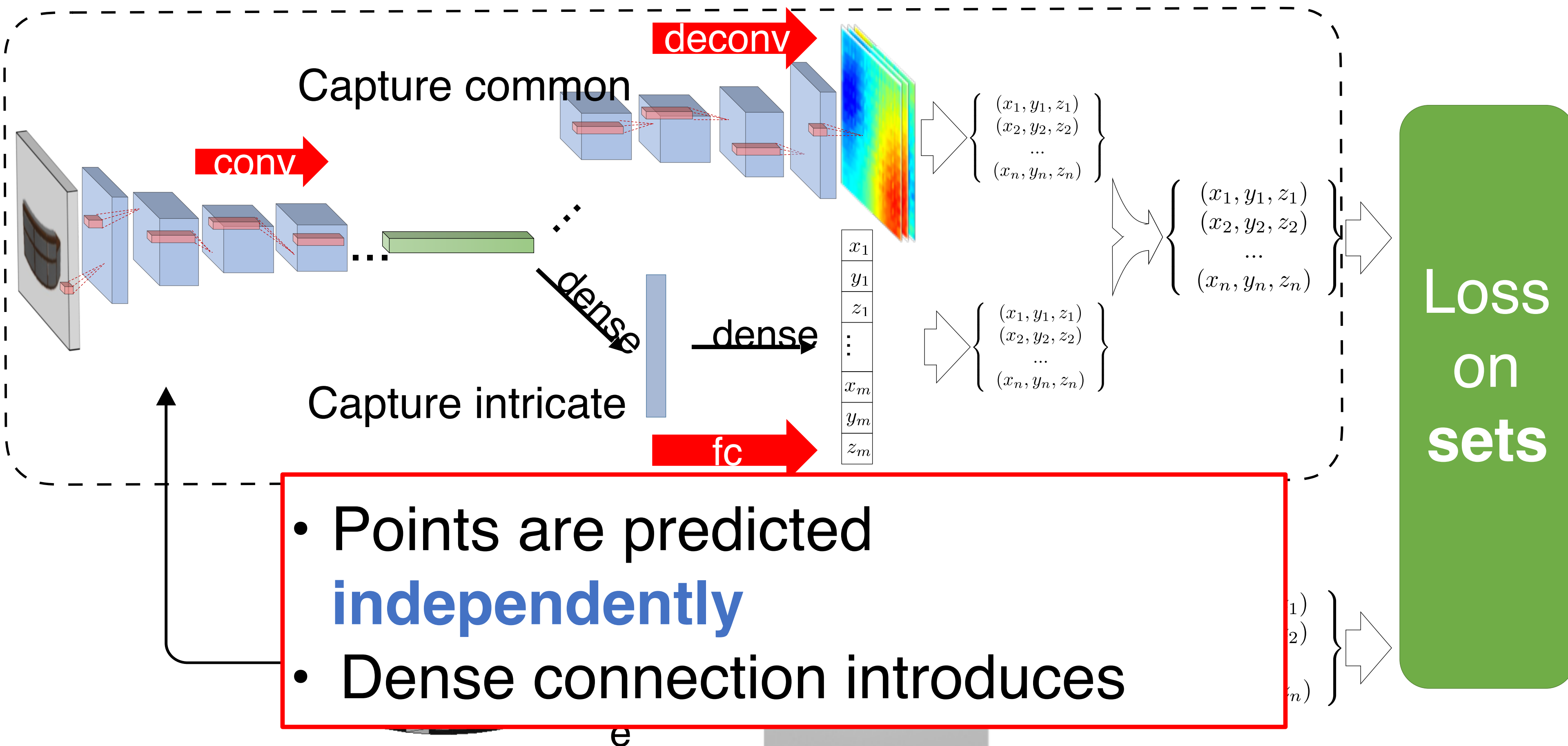
CVPR '17, Point Set Generation

Pipeline



CVPR '17, Point Set Generation

Pipeline

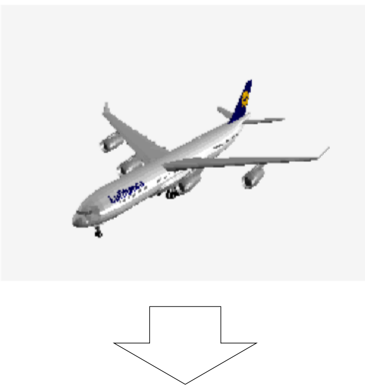


- Points are predicted **independently**
- Dense connection introduces

CVPR '17, Point Set Generation

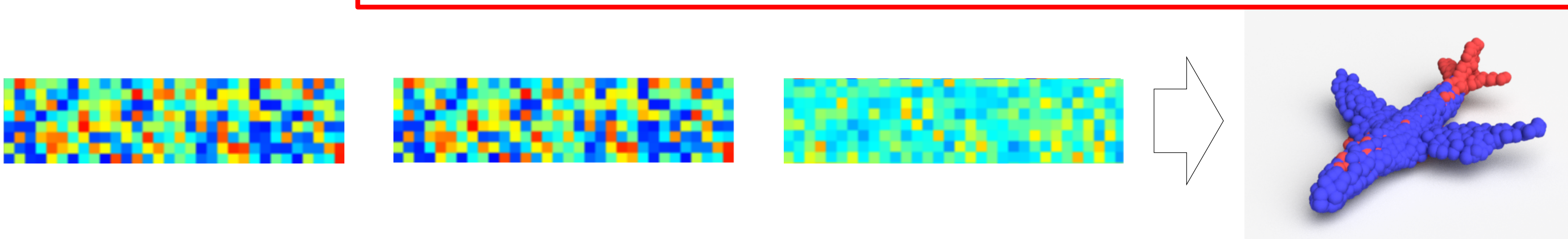
Visualization of the effect of FC branch

- Surface parametrization (2D   3D mapping)



Observation:

- The arrangement of predicted points are uncorrelated



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

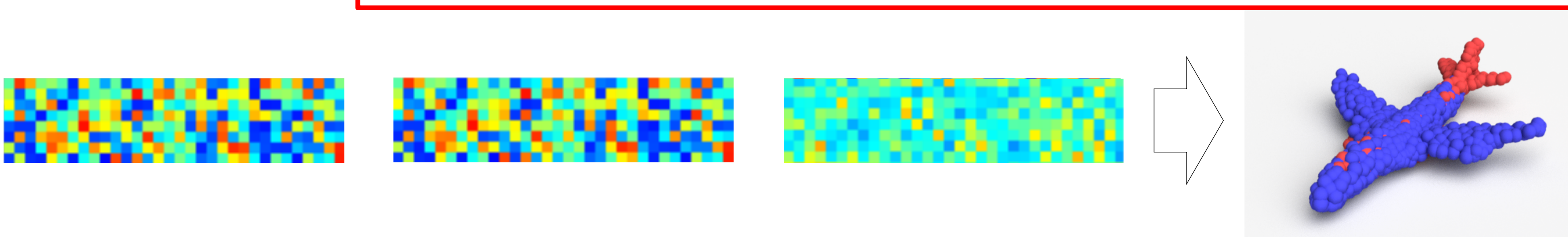
CVPR '17, Point Set Generation

Visualization of the effect of FC branch

- Surface parametrization (2D   3D mapping)

Observation:

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



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

CVPR ’17, Point Set Generation

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

CVPR '17, Point Set Generation

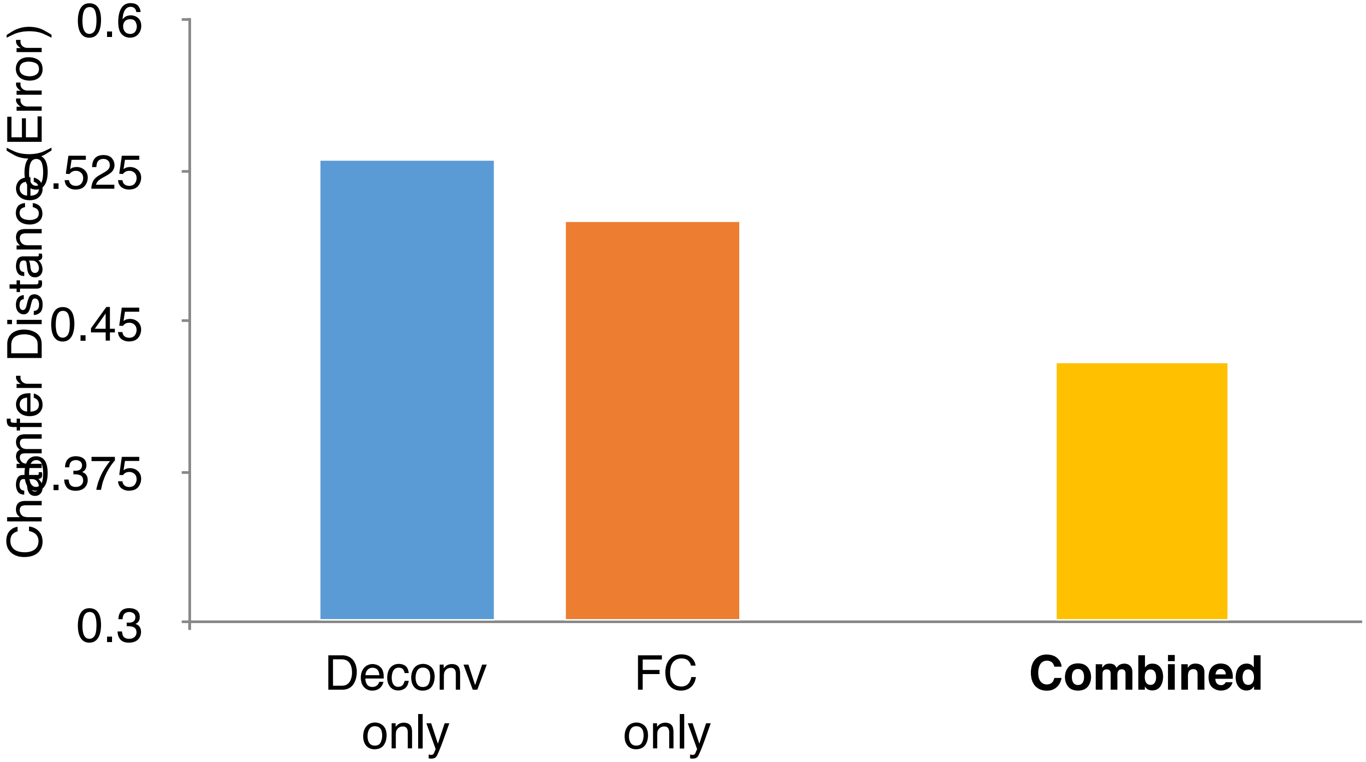
Q: Which color corresponds to the deconv branch? FC branch?

blue: deconv branch – **large, smooth** structures
red: FC branch – **intricate** structures

CVPR '17, Point Set Generation

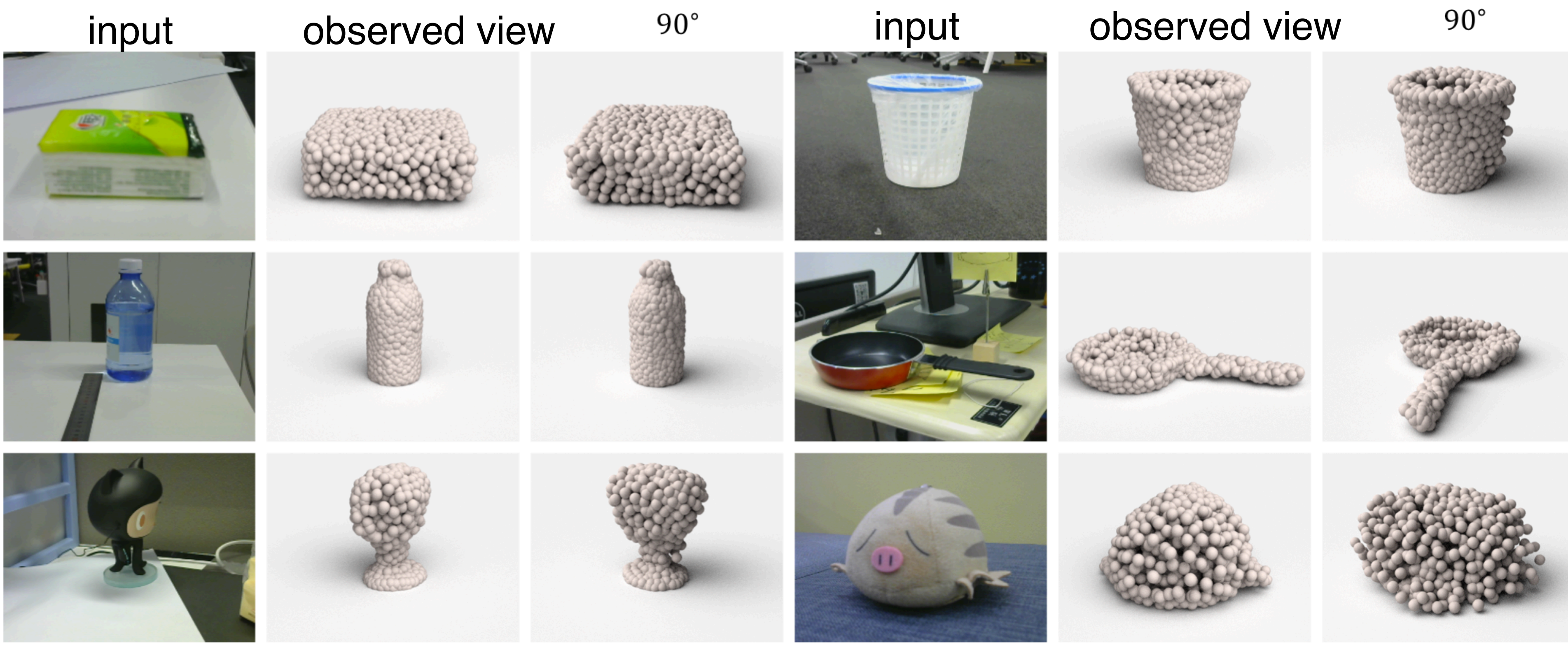
Effect of combining two branches

Train/tested on 2K object categories



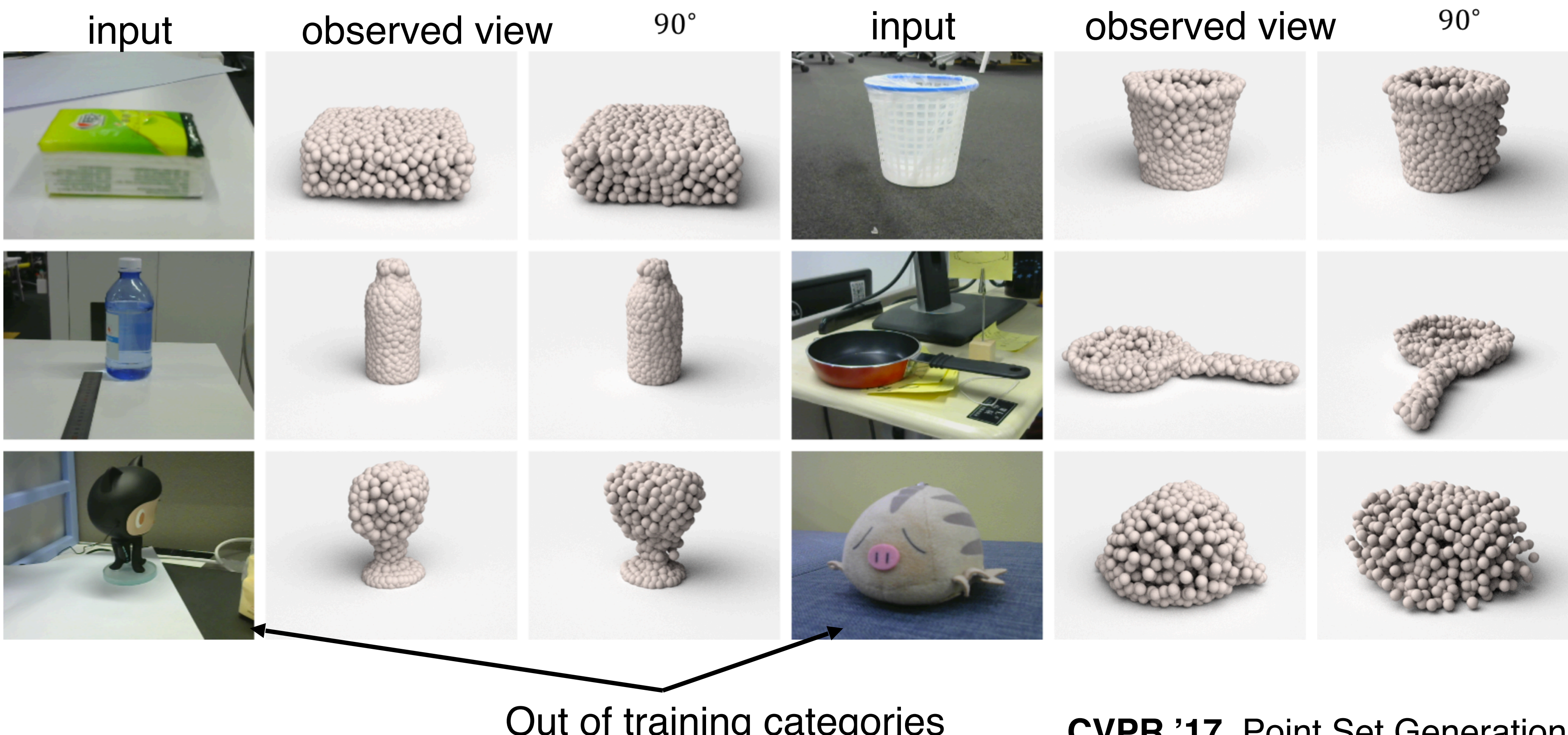
CVPR '17, Point Set Generation

Real-world results



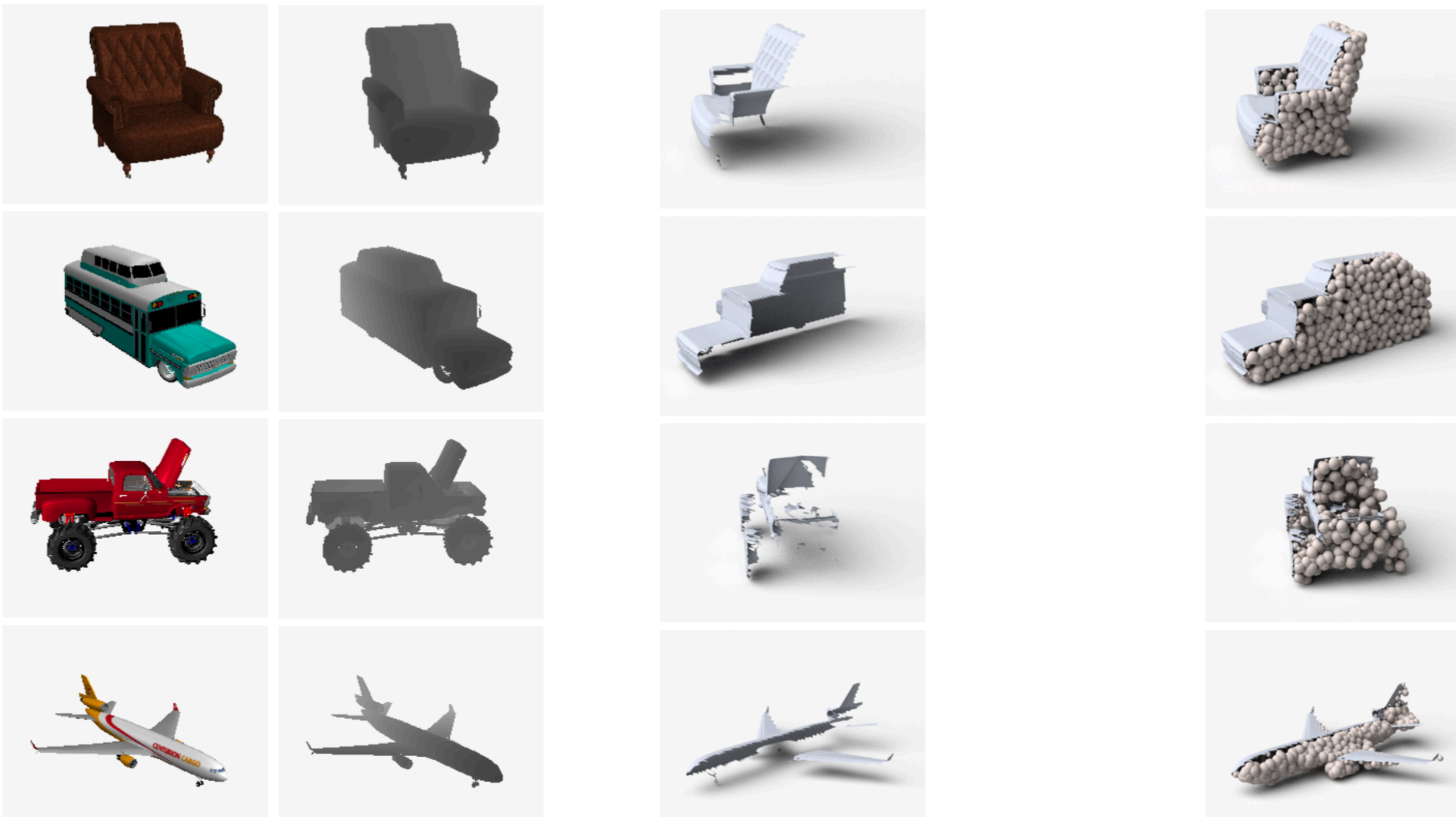
CVPR ’17, Point Set Generation

Generalization to unseen categories



CVPR '17, Point Set Generation

Extension: shape completion for RGBD data



RGBD map (input)

90° view of input

output: completed point cloud

CVPR ’17, Point Set Generation

Generation of Multiple Plausible Shapes

Ambiguity of the prediction arises at test time, the depth for visible parts is under-determined, and the geometry for invisible parts has to be hallucinated by guessing.

Min-of-N Loss (MoN):

$$\underset{\Theta}{\text{minimize}} \sum_{k} \min_{\substack{r_j \sim \mathbb{N}(\mathbf{0},\mathbf{I}) \\ 1 \leq j \leq n}} \{d(\mathbb{G}(I_k, r_j; \Theta), S_k^{gt})\}$$

Min-of-N Loss (MoN)

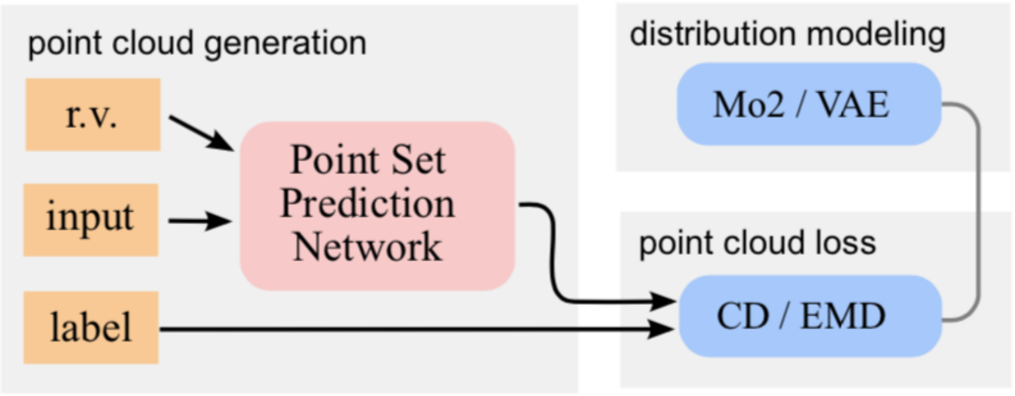


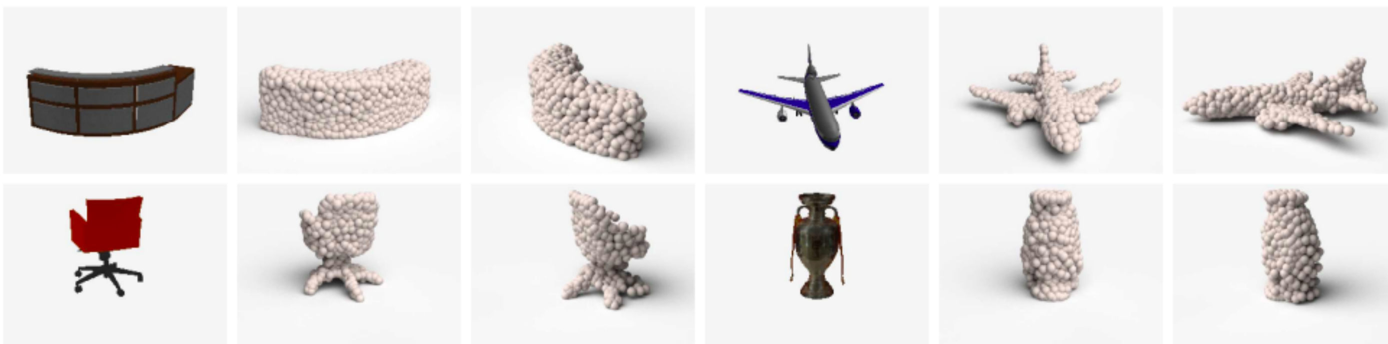
Figure 4. System structure. By plugging in distributional modeling module, our system is capable of generating multiple predictions.

SURFACE DEFORMATION-BASED RECONSTRUCTION

UC San Diego

Generating points : PointSetGen

Another approach is to sample points on the surface of the 3D shape and work with an

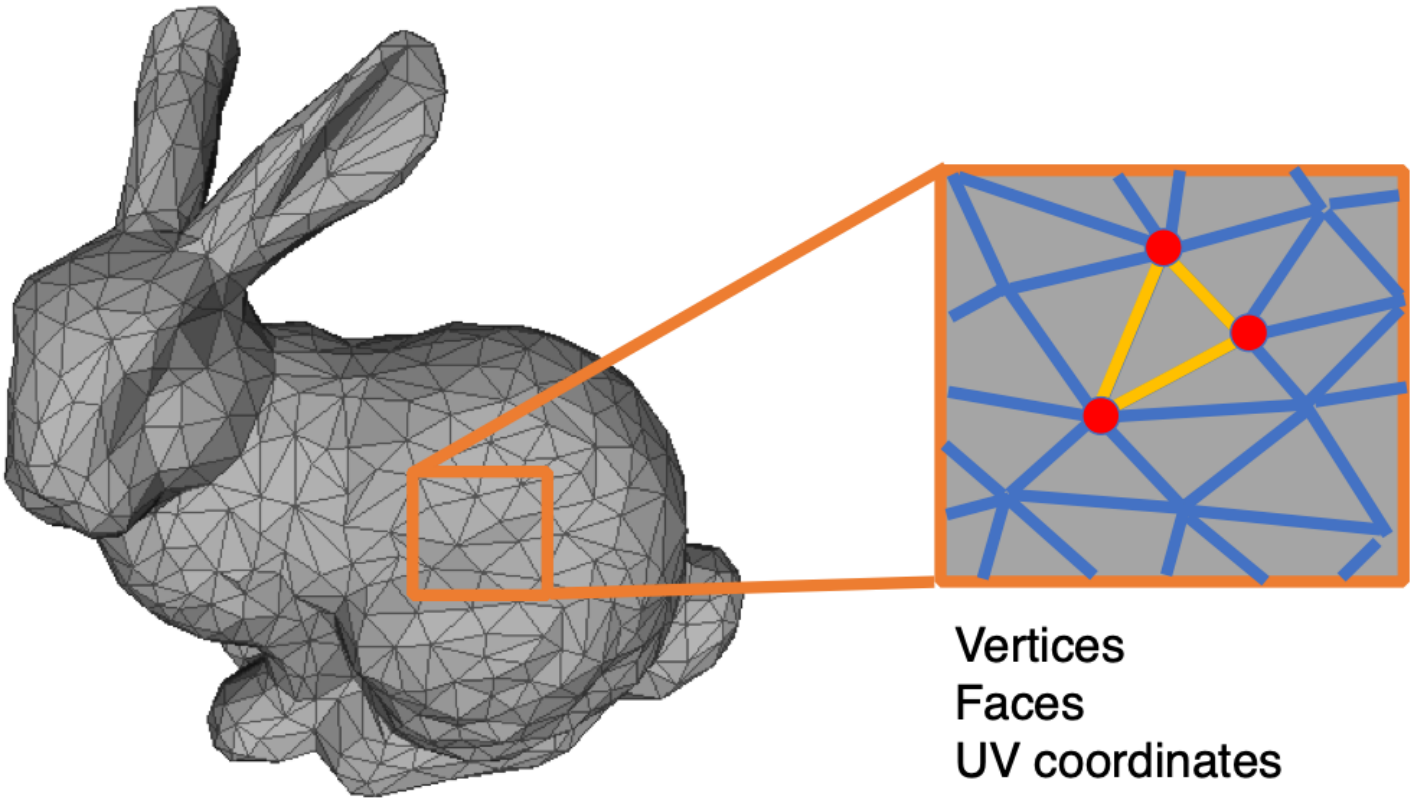


Fan, H., Su, H., & Guibas, L. A point set generation network for 3d object reconstruction from a single image. CVPR 2017

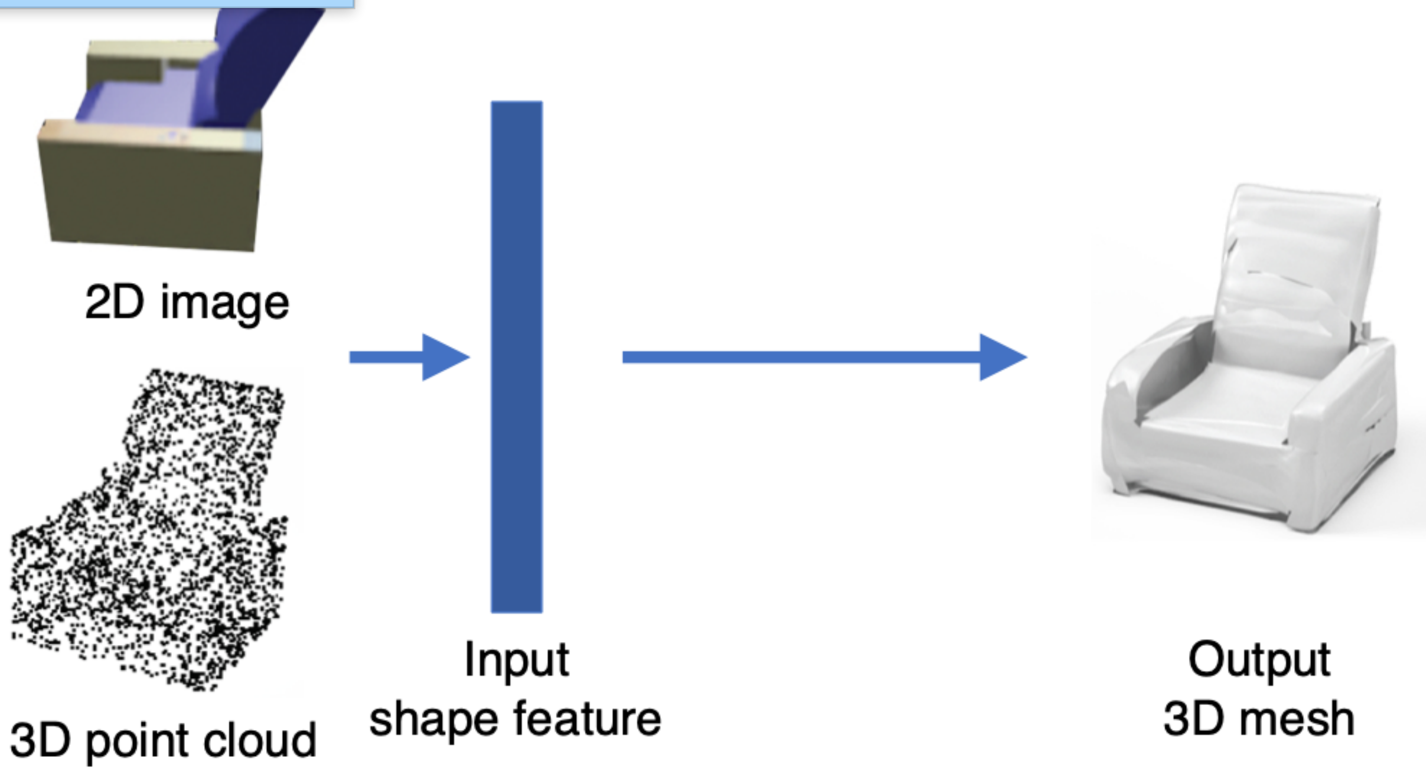
- Simple
- Unstructured point cloud

In fact, that’s our goal :
generating a set of 3D points
and the connectivity between

meshes and atlases



From an input object (on the left), we use existing methods to extract a latent vector, and



Let’s try this on an arbitrary shape : me :)

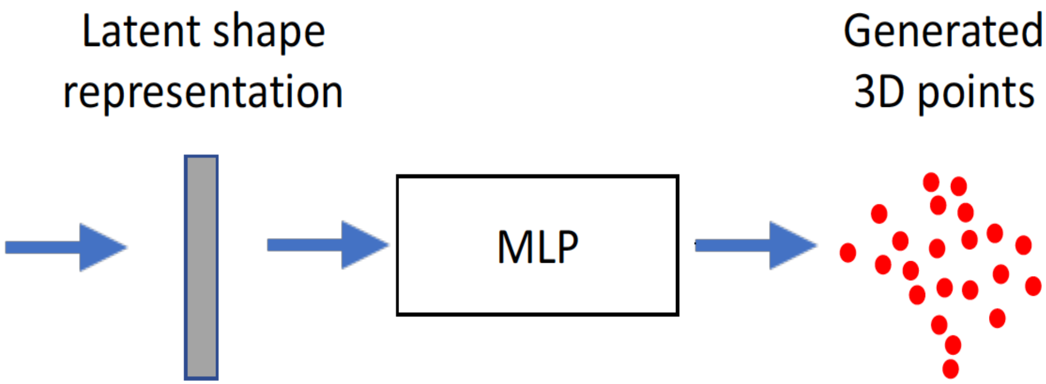
Test Shape

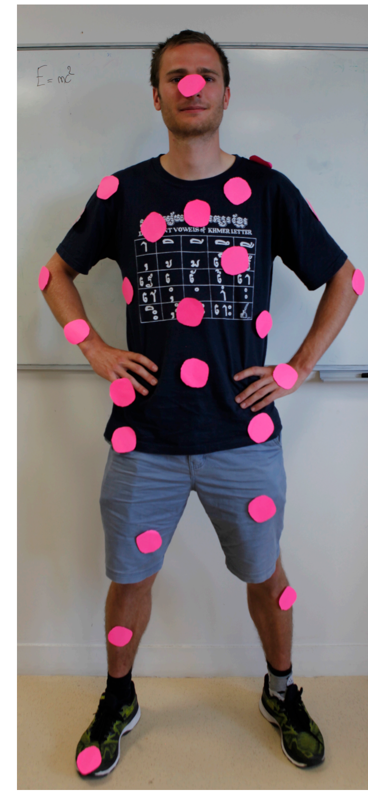
Generating points

We build on PointSetGen, and its point cloud representation. In its simplest form, the latent

results

a of surface, thin structures

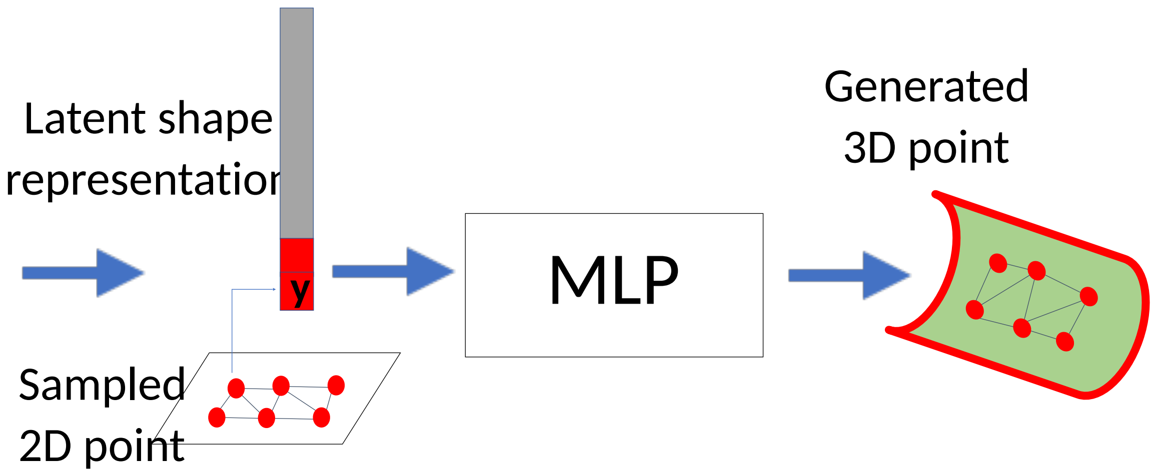




Key idea 1: deform a surface

We observed that and build on this work by adding in the decoder architecture the prior

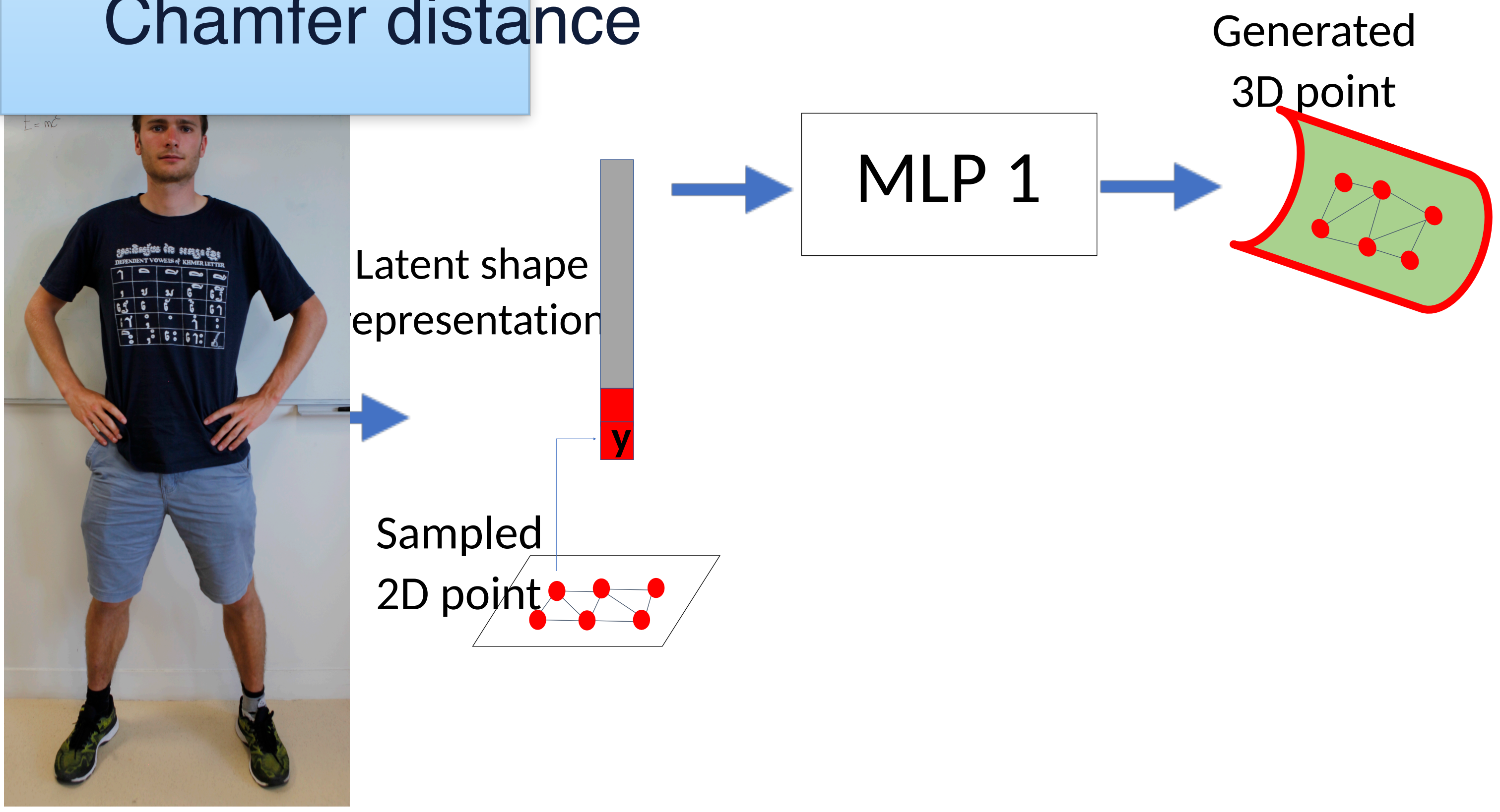
Learnt simply by sampling many points and minimizing Chamfer distance



Key idea 2: learn an atlas

To solve this issue, instead of learning a single deformation, we learn K deformations

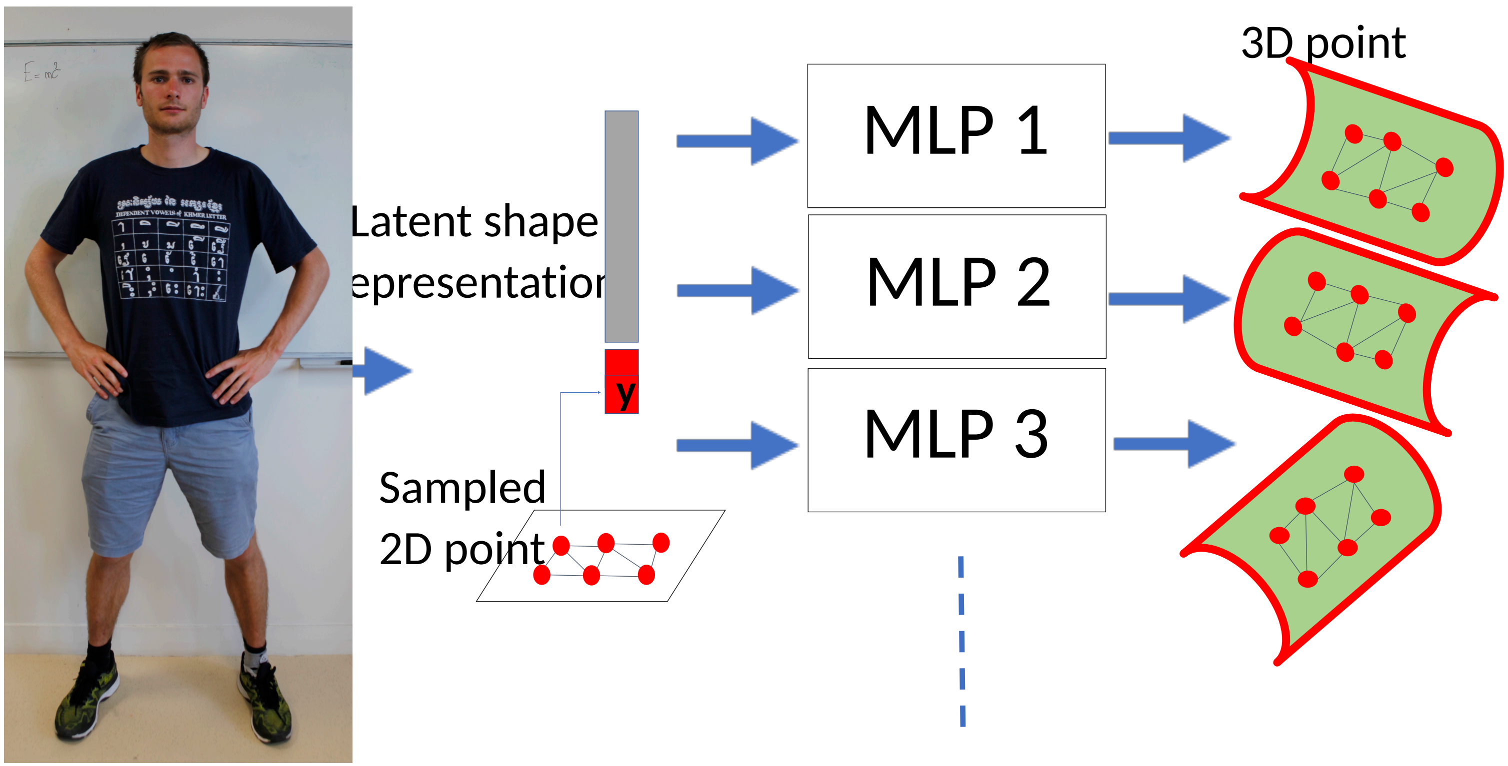
Learnt simply by sampling many points and minimizing Chamfer distance



ed object.

y idea 2: learn an atlas

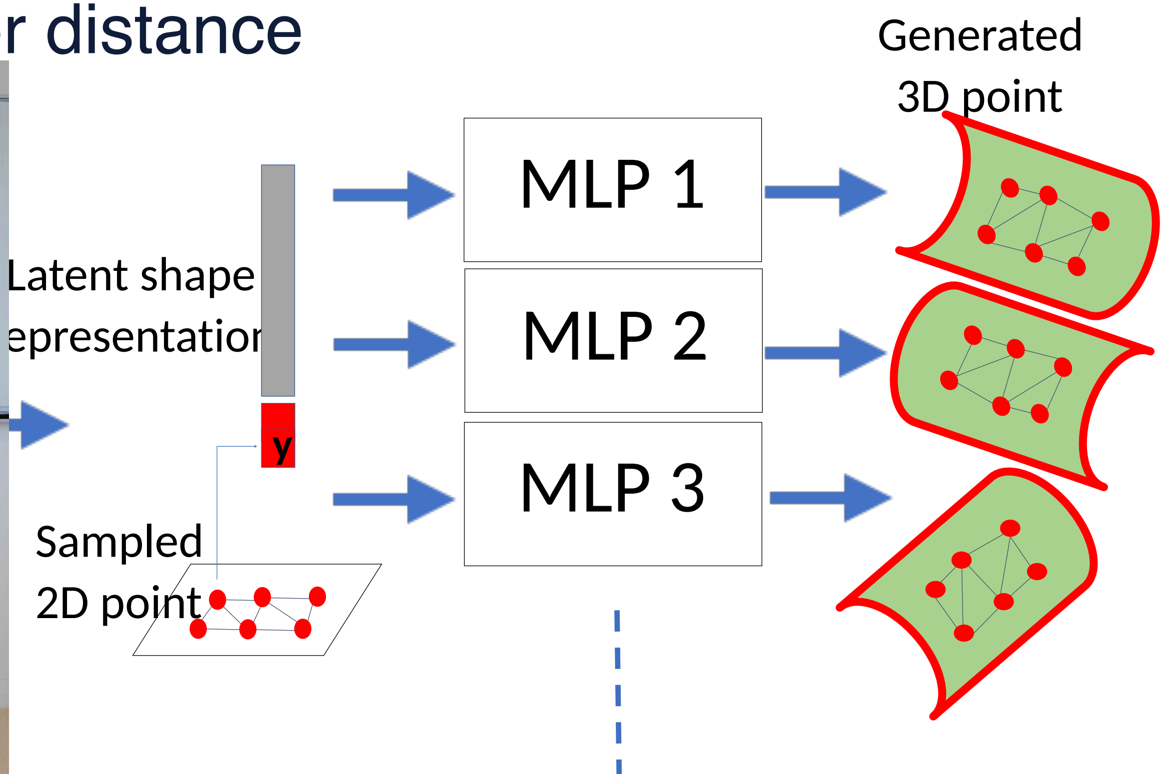
- Learnt simply by sampling many points and minimizing Chamfer distance



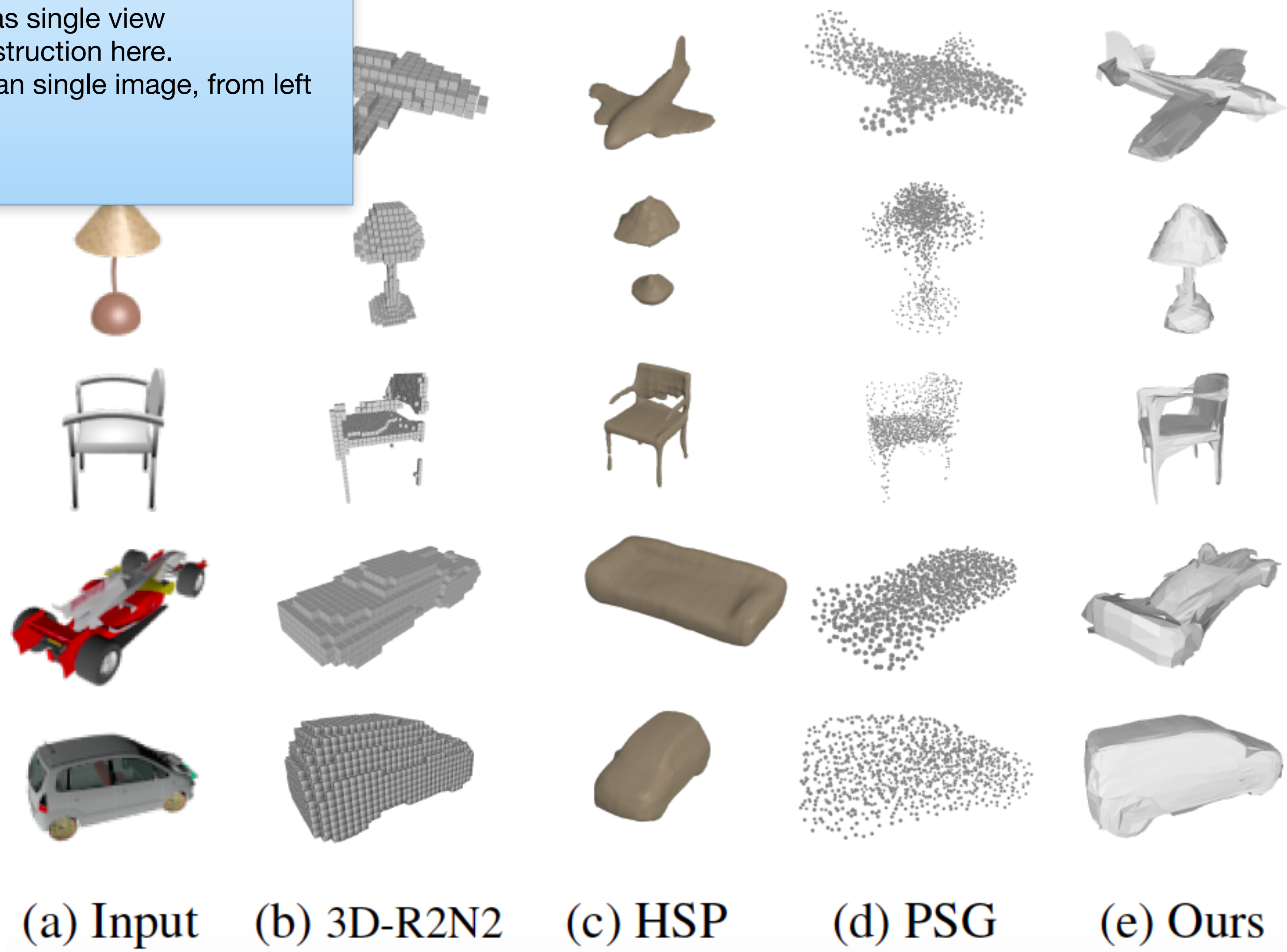
es back to the

idea 2: learn an atlas

Learnt simply by sampling many points and minimizing Chamfer distance



such as single view
reconstruction here.
From an single image, from left



NEXT LECTURE: LAPLACIAN