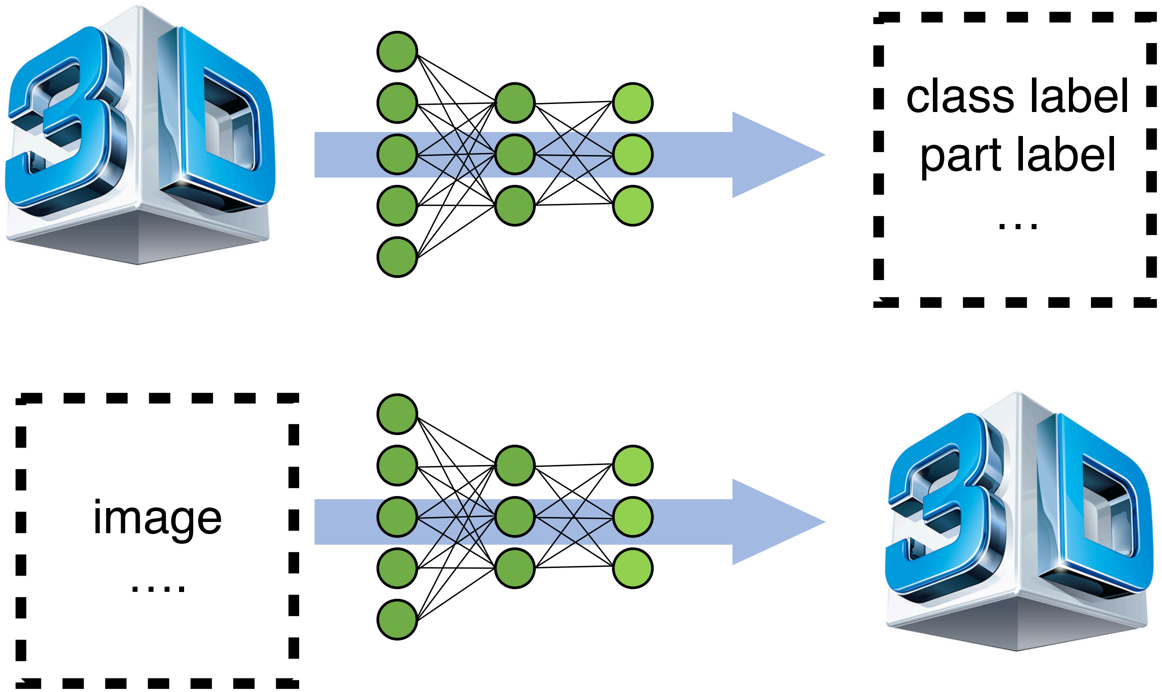
CS468: 3D Deep Learning on Point Cloud Data



Hao Su
Stanford University

May 10, 2017

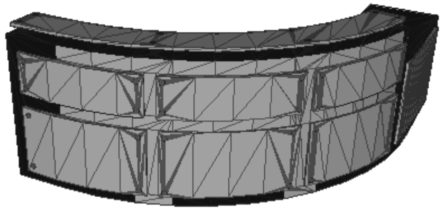
Agenda

- **Point cloud generation**

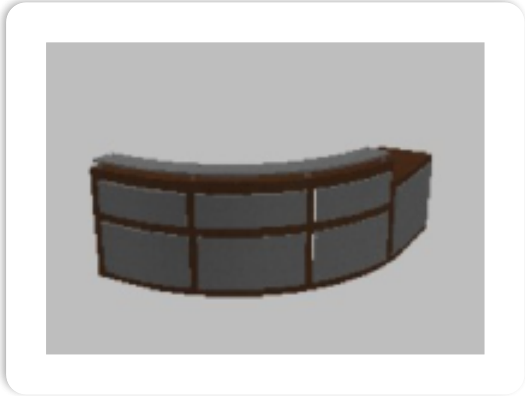
- Point cloud analysis

Pipeline

render



Pipeline

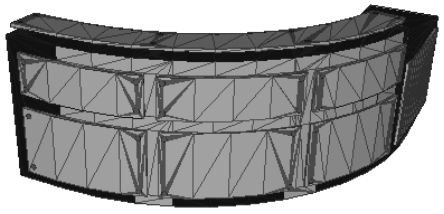


2K object categories

200K shapes

~10M image/point set pairs

render

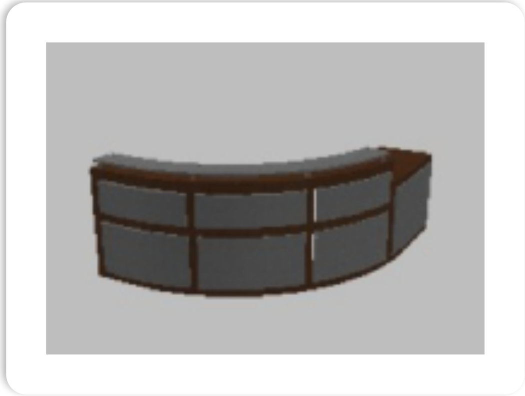


sample

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

Groundtruth point **set**

Pipeline



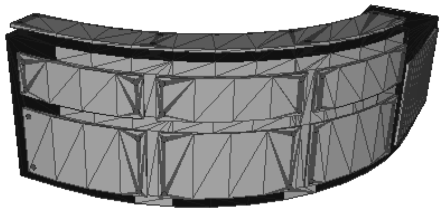
Shape predictor

(f)

Prediction

$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

render

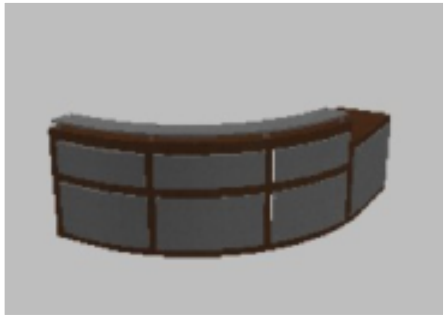


sample

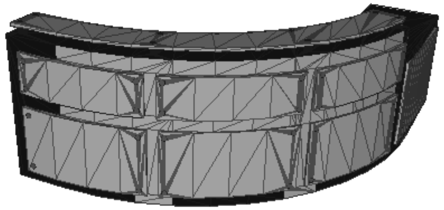
$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{array} \right\}$$

Groundtruth point **set**

Pipeline



render



sample

Shape predictor

(f)

Prediction

$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

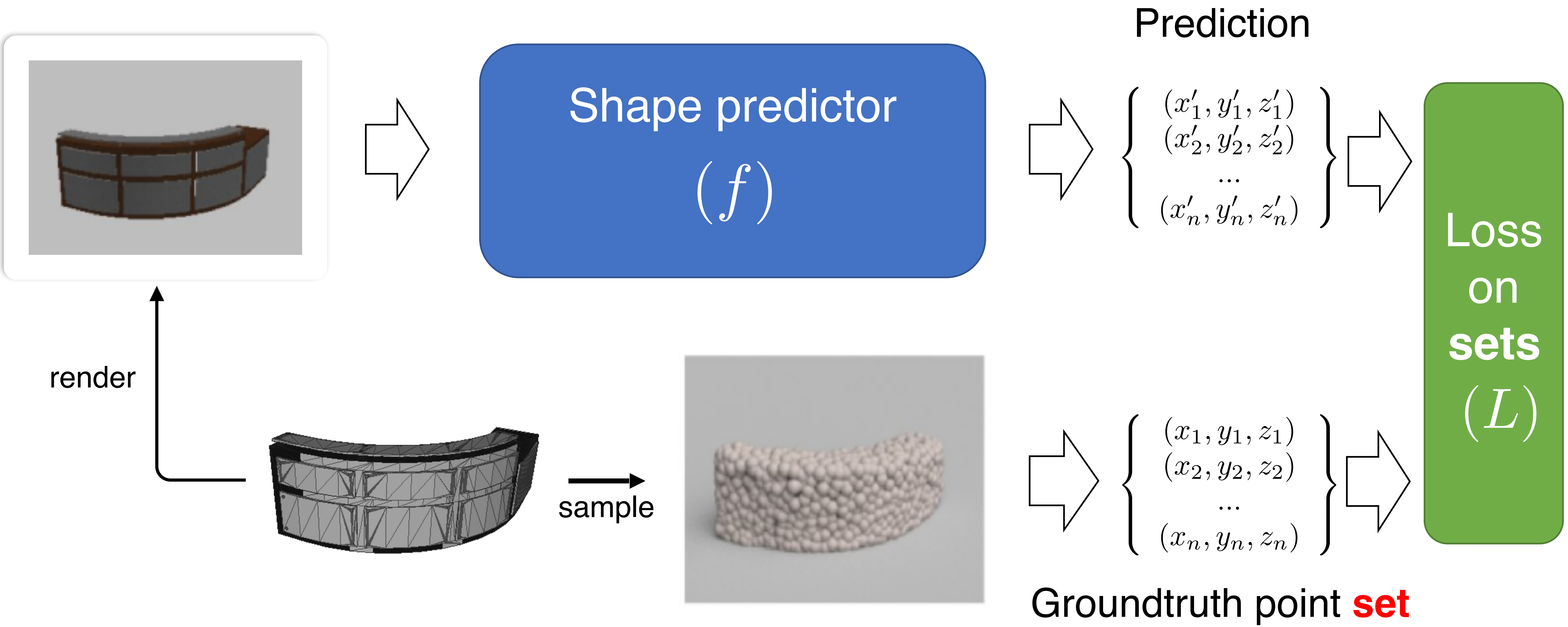
$$\wr\wr$$

A set is invariant up to permutation

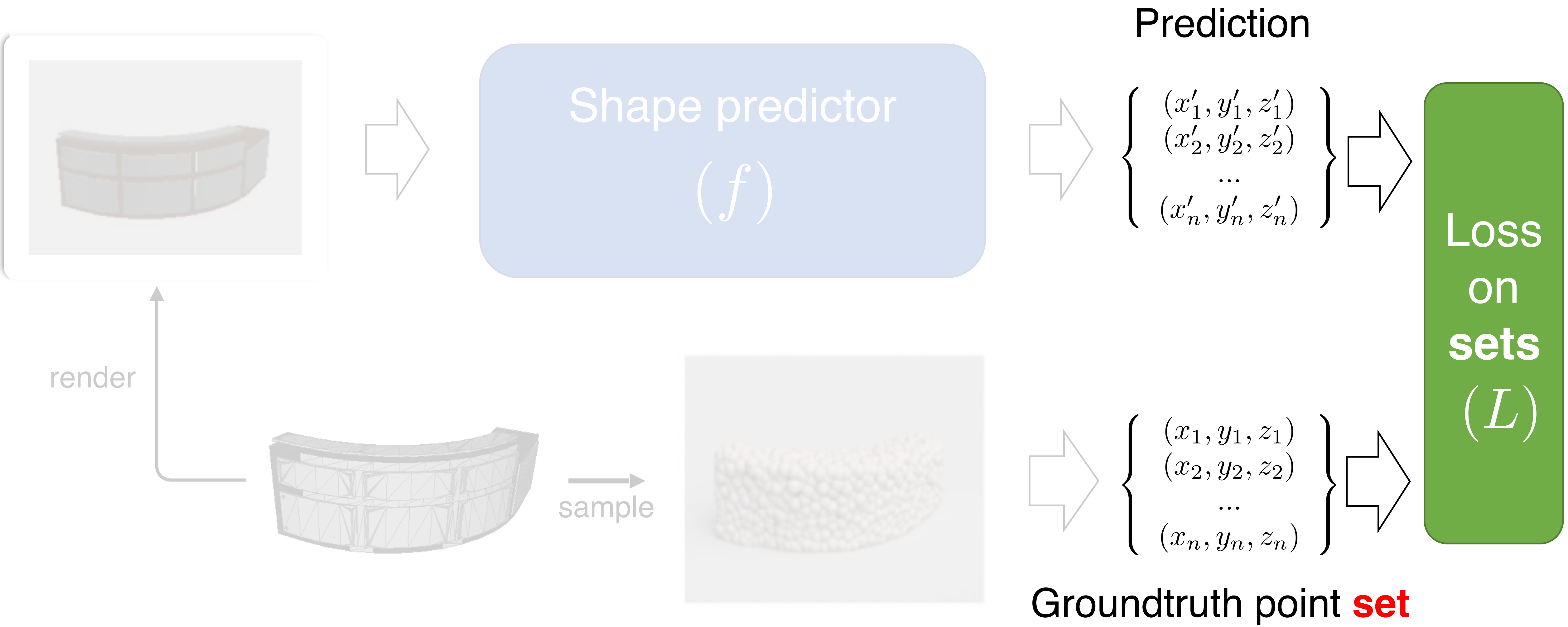
$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{array} \right\}$$

Groundtruth point **set**

Pipeline

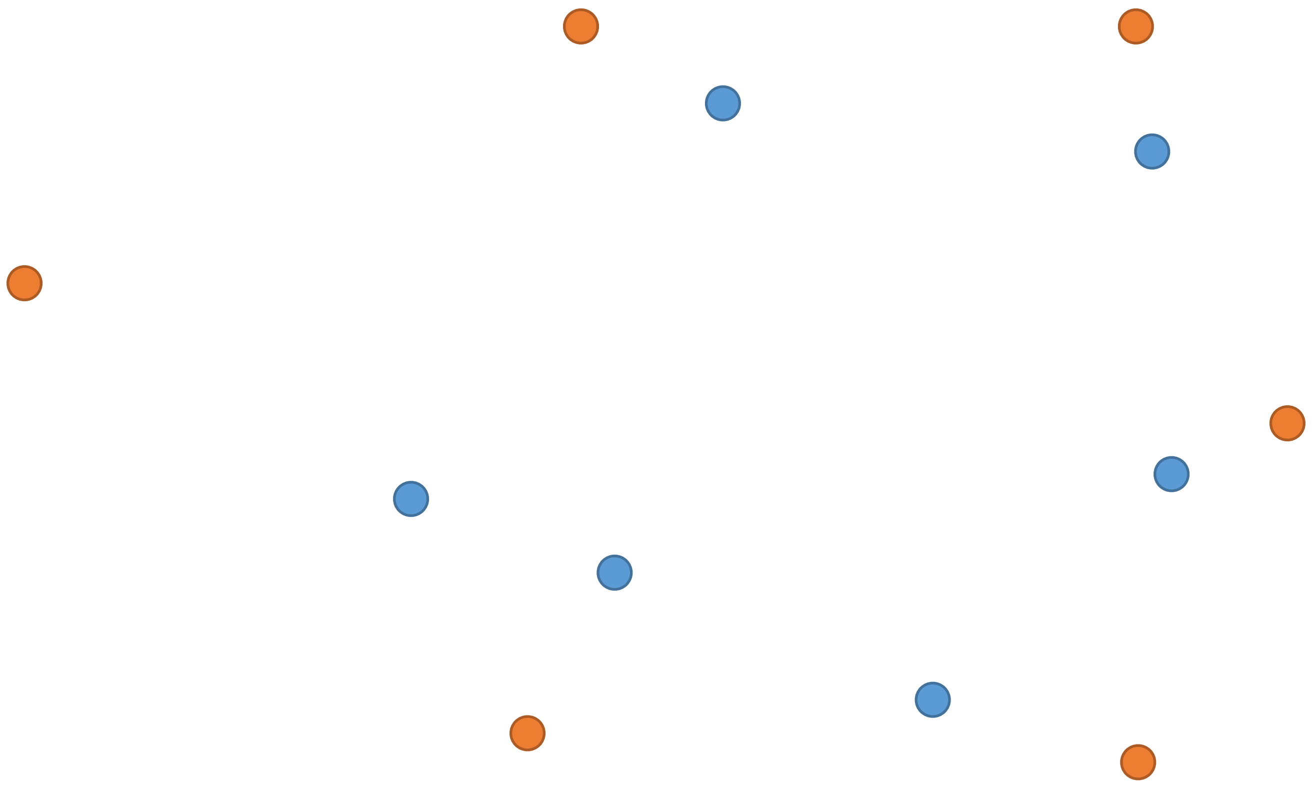

Prediction
Shape predictor
(f)
$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \cdots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$
Loss on **sets** (L)
render
sample
$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \cdots \\ (x_n, y_n, z_n) \end{array} \right\}$
Groundtruth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

Set comparison

Given two sets of points, measure their discrepancy



Set comparison

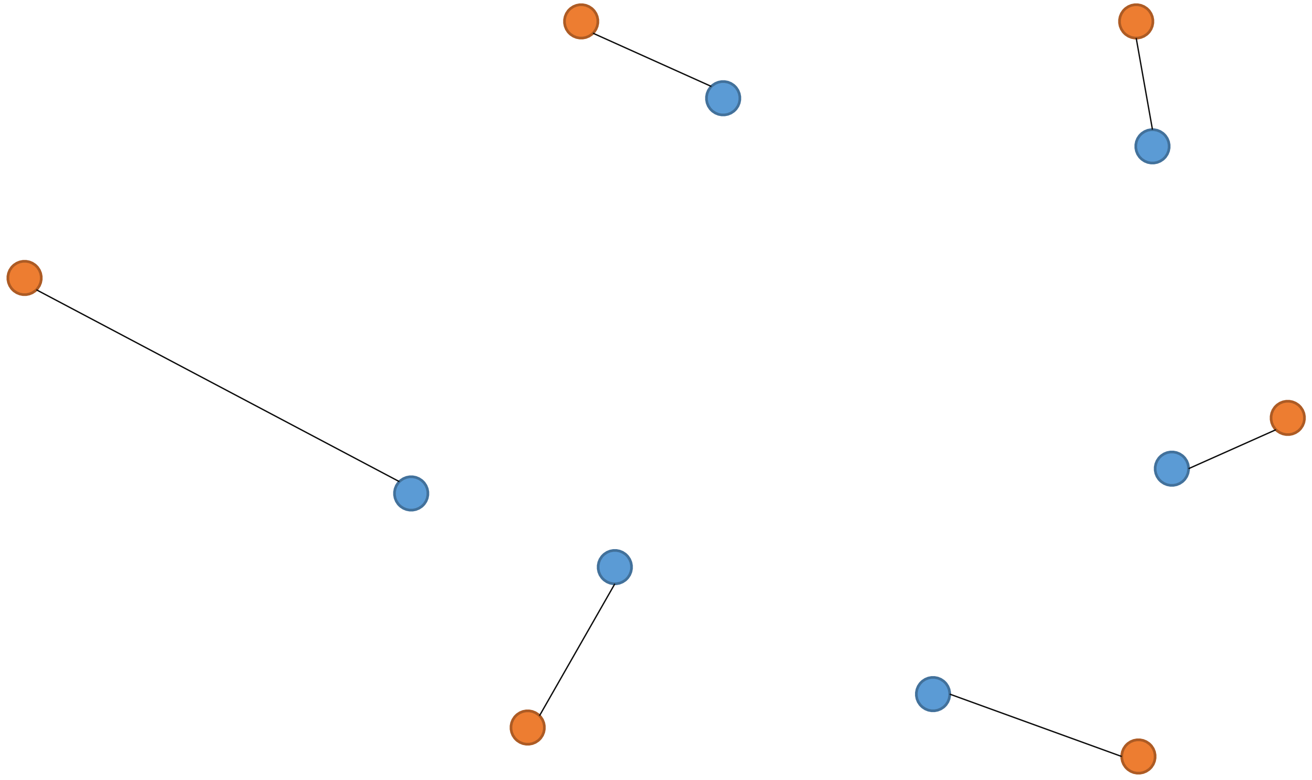
Given two sets of points, measure their discrepancy

Key challenge:

correspondence problem

Correspondence (I): optimal assignment

Given two sets of points, measure their discrepancy



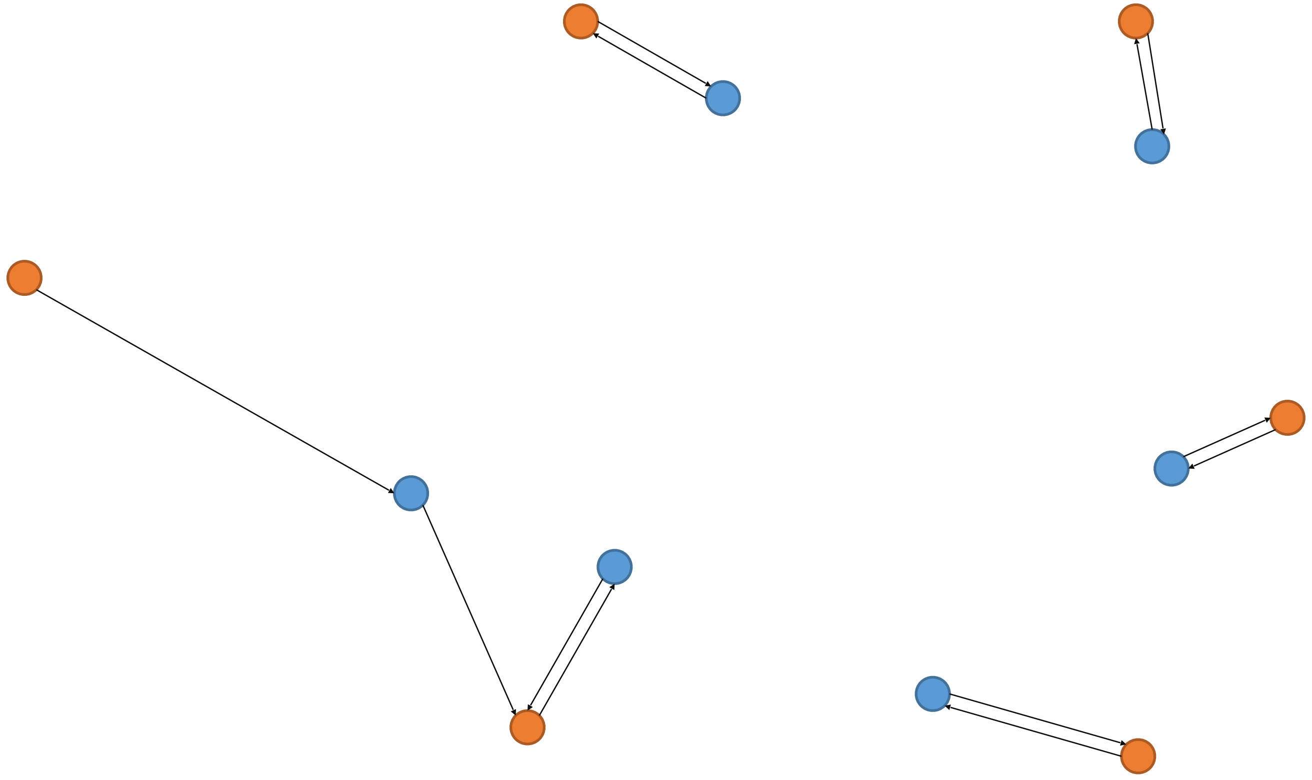
a.k.a Earth Mover's distance (EMD)

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

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***

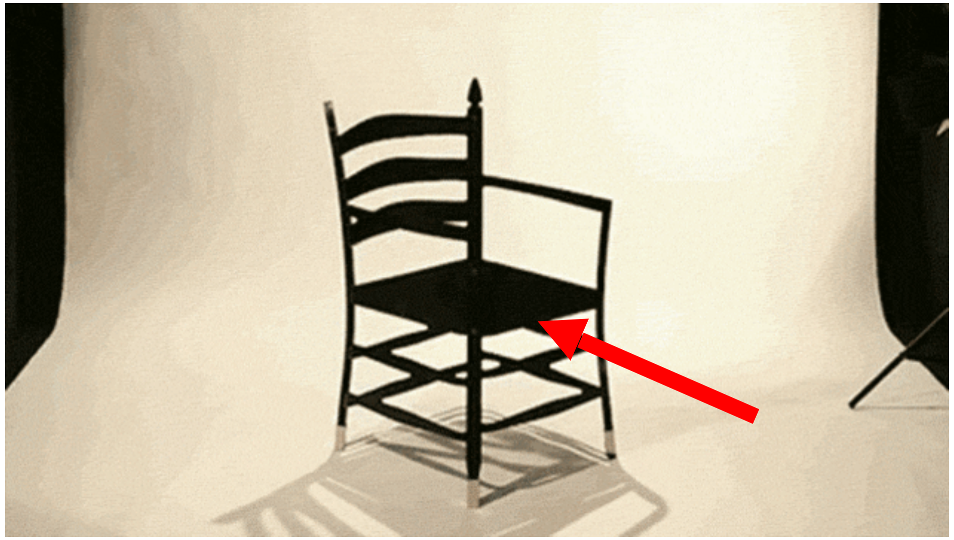
Computational requirement

How distance metric affects learning?

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

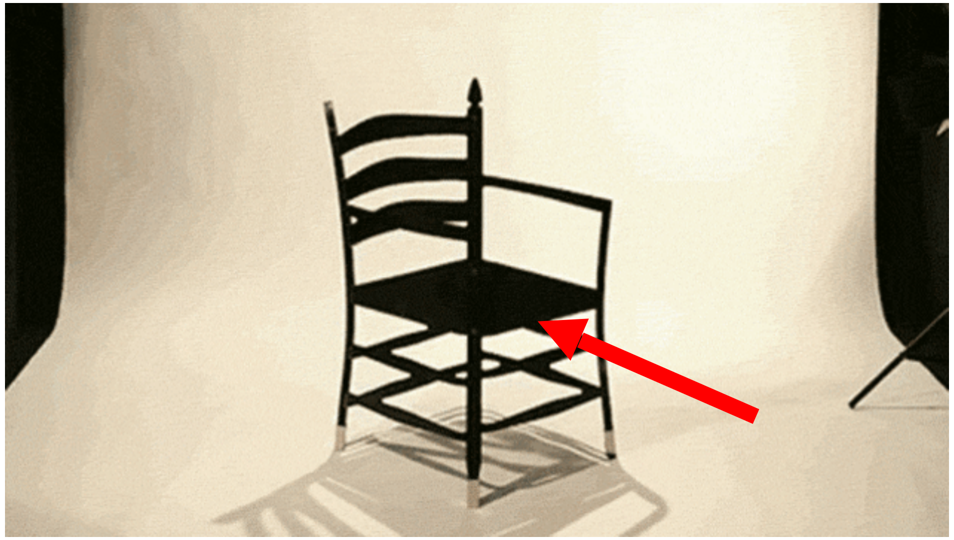
How distance metric affects learning?

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



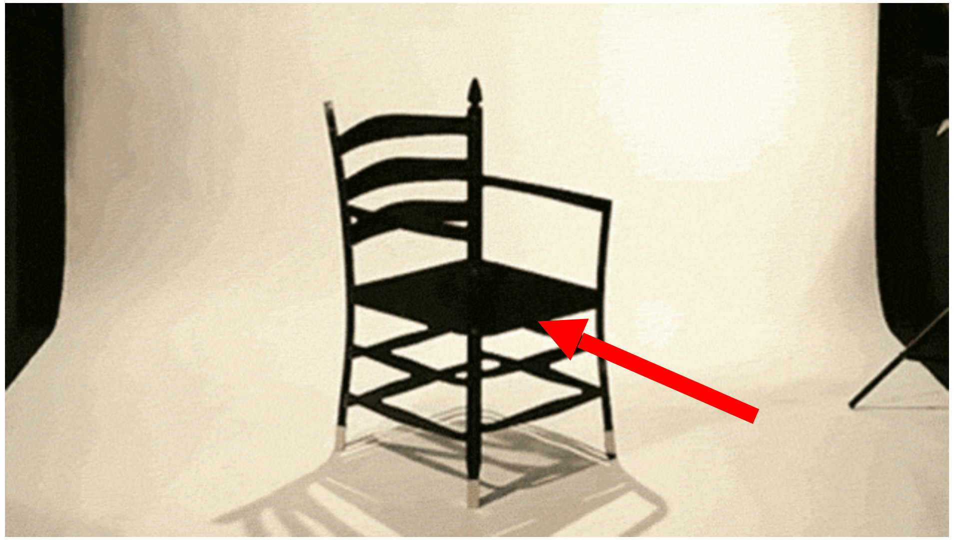
How distance metric affects learning?

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



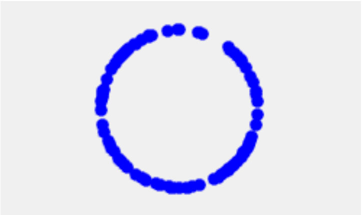
- By loss minimization, the network tends to predict a "**mean shape**" that **averages out** uncertainty

Distance metrics affect mean shapes

The mean shape carries characteristics of the distance metric

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

continuous
hidden variable
(radius)



Input

EMD mean

Chamfer mean

CVPR '17, Point Set Generation

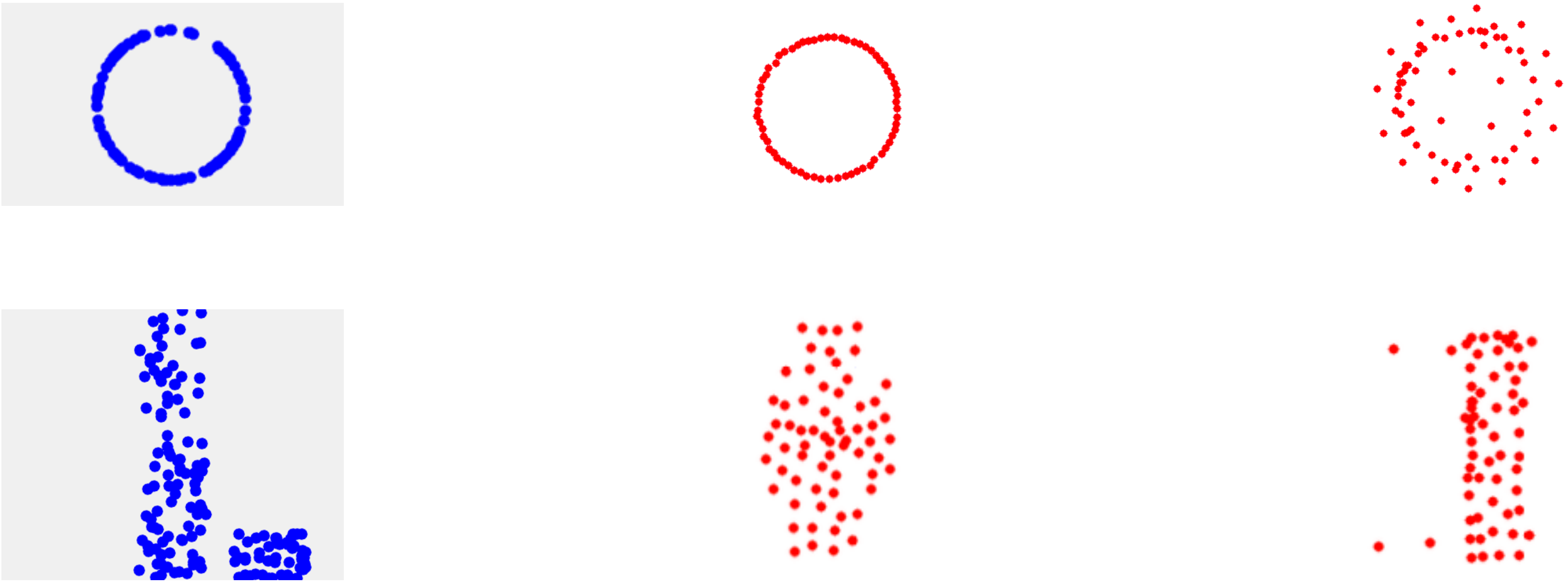
Mean shapes from distance metrics

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)



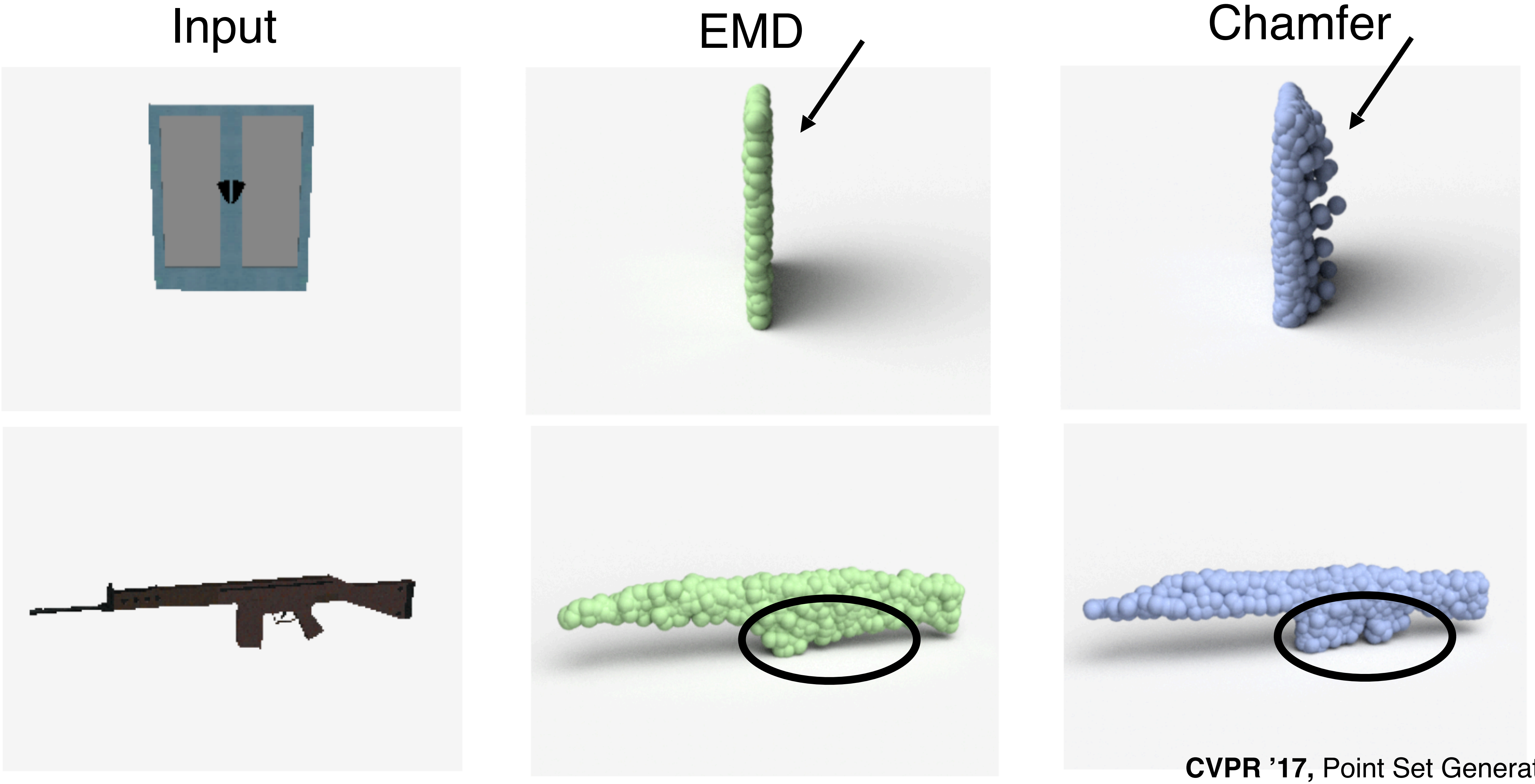
Input

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

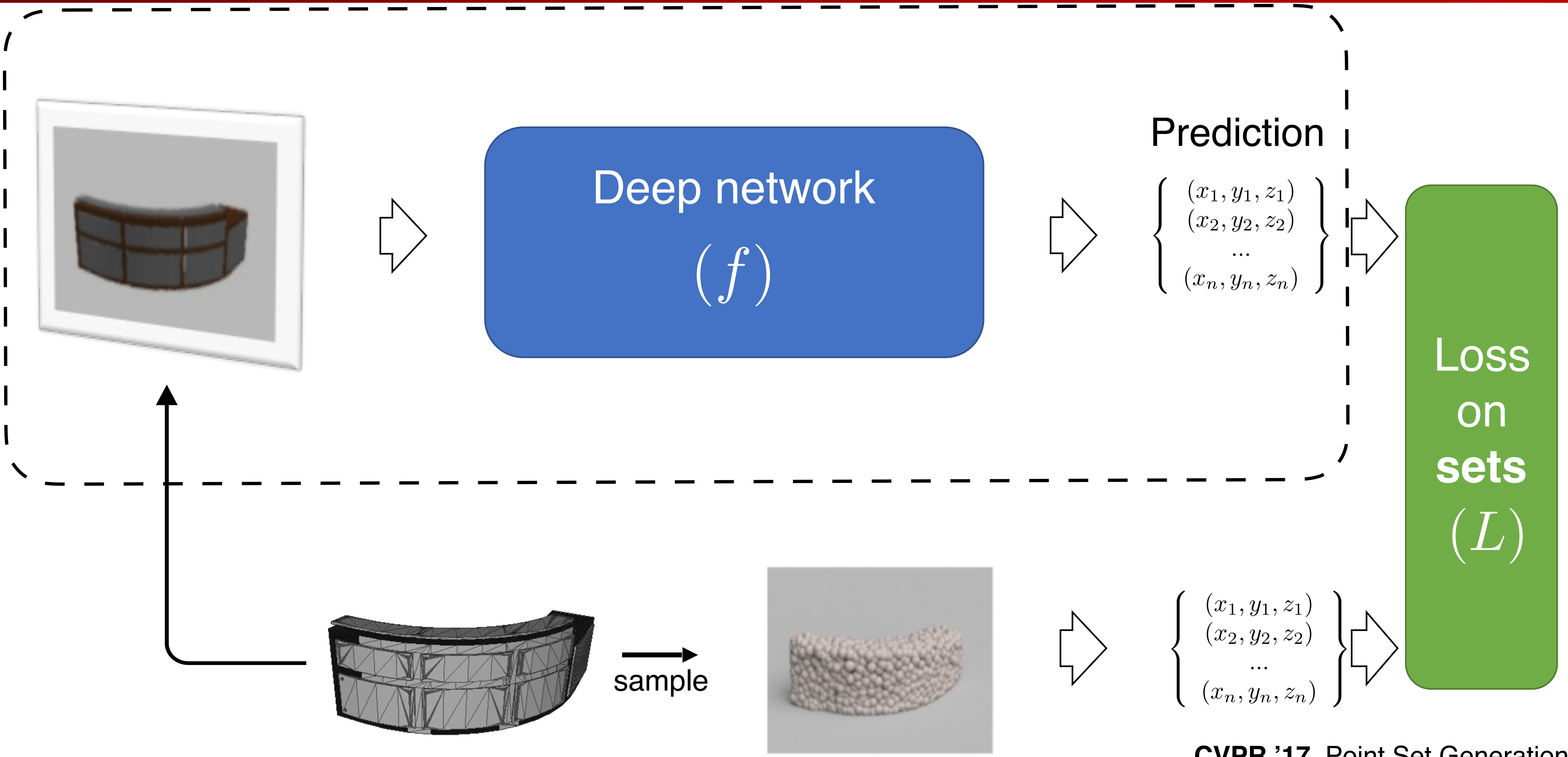
- **Efficient** to compute

Chamfer distance: trivially parallelizable on CUDA

Earth Mover's distance (optimal assignment):

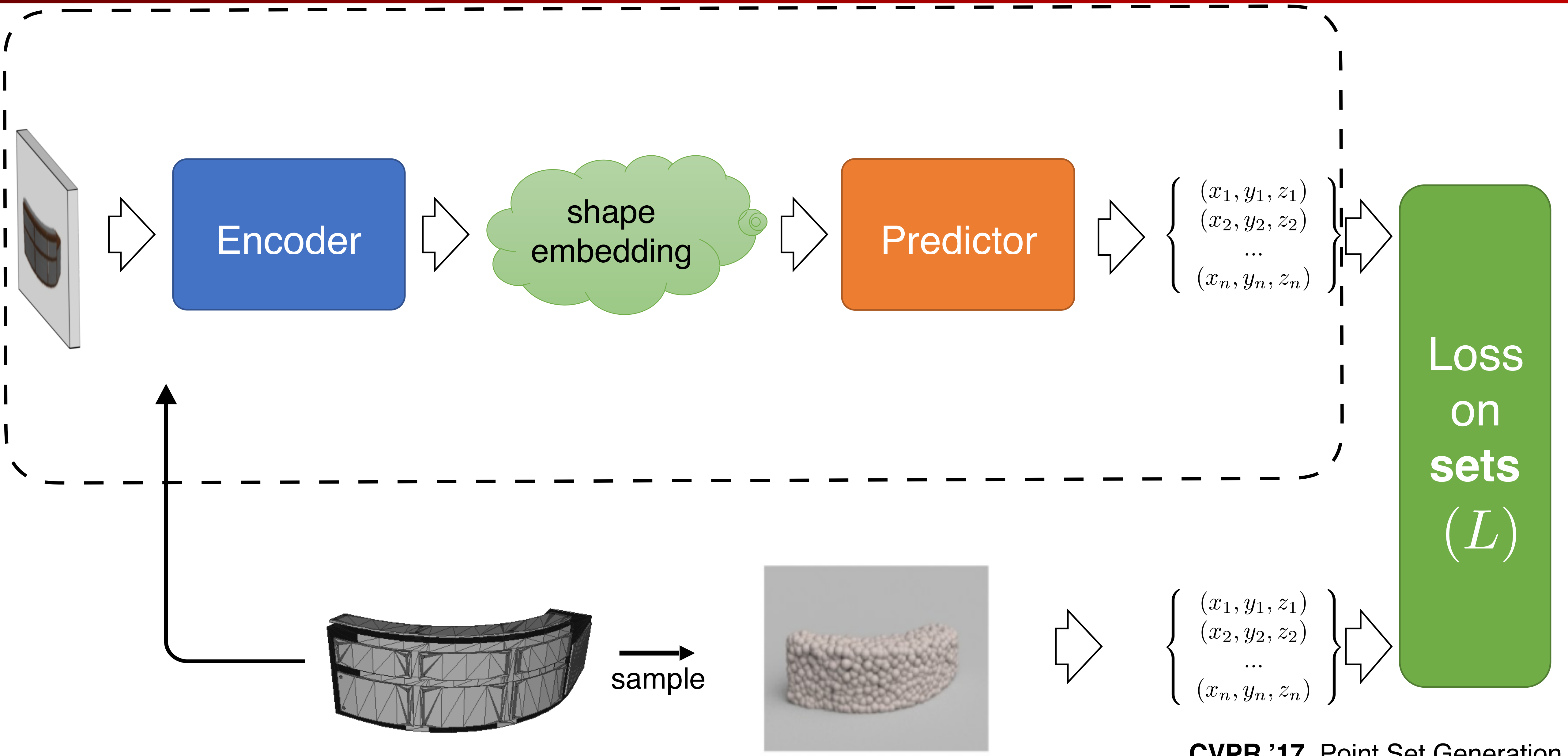
- We implement a **distributed** approximation algorithm on CUDA
- Based upon [Bertsekas, 1985], $(1+\epsilon)$-approximation

Pipeline



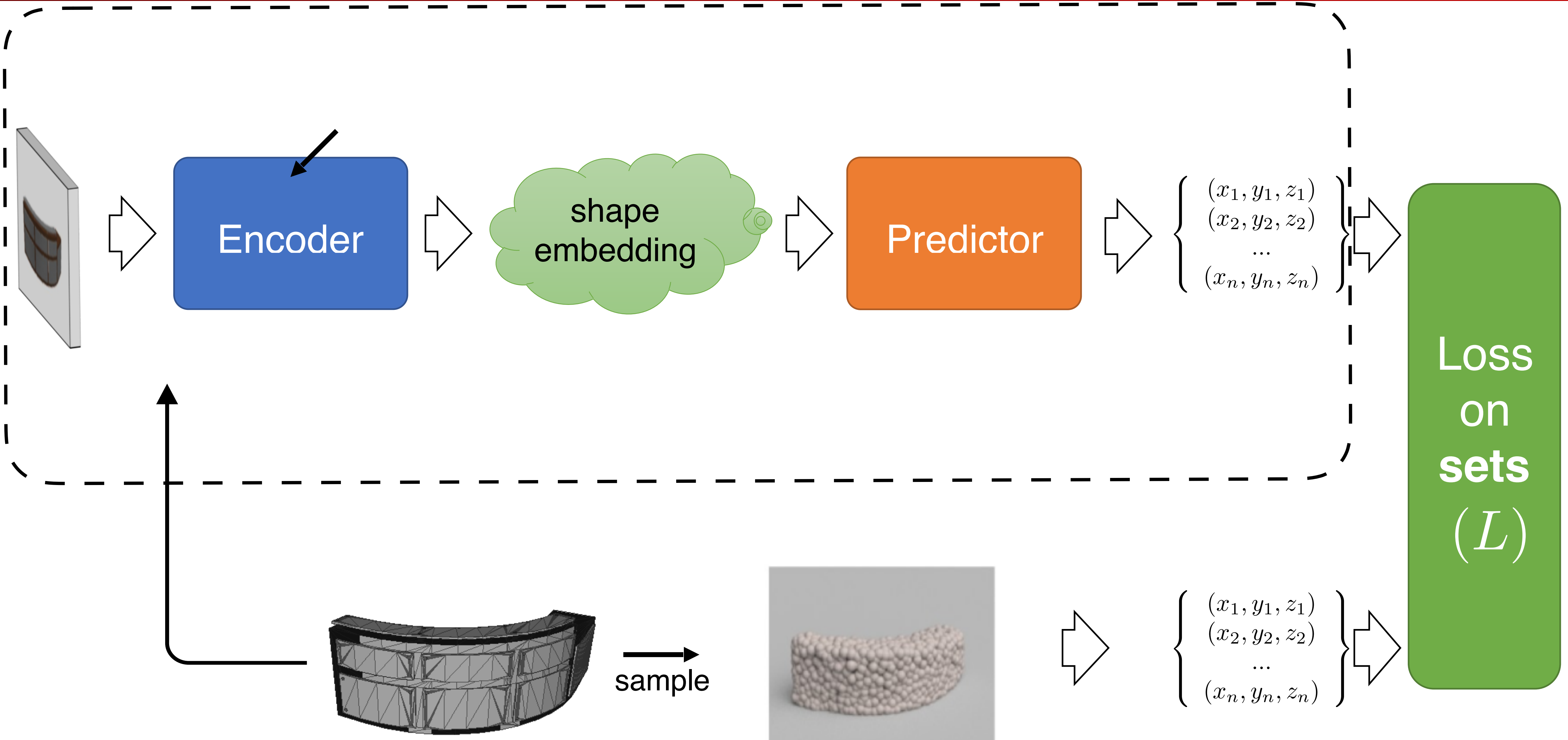
CVPR '17, Point Set Generation

Pipeline



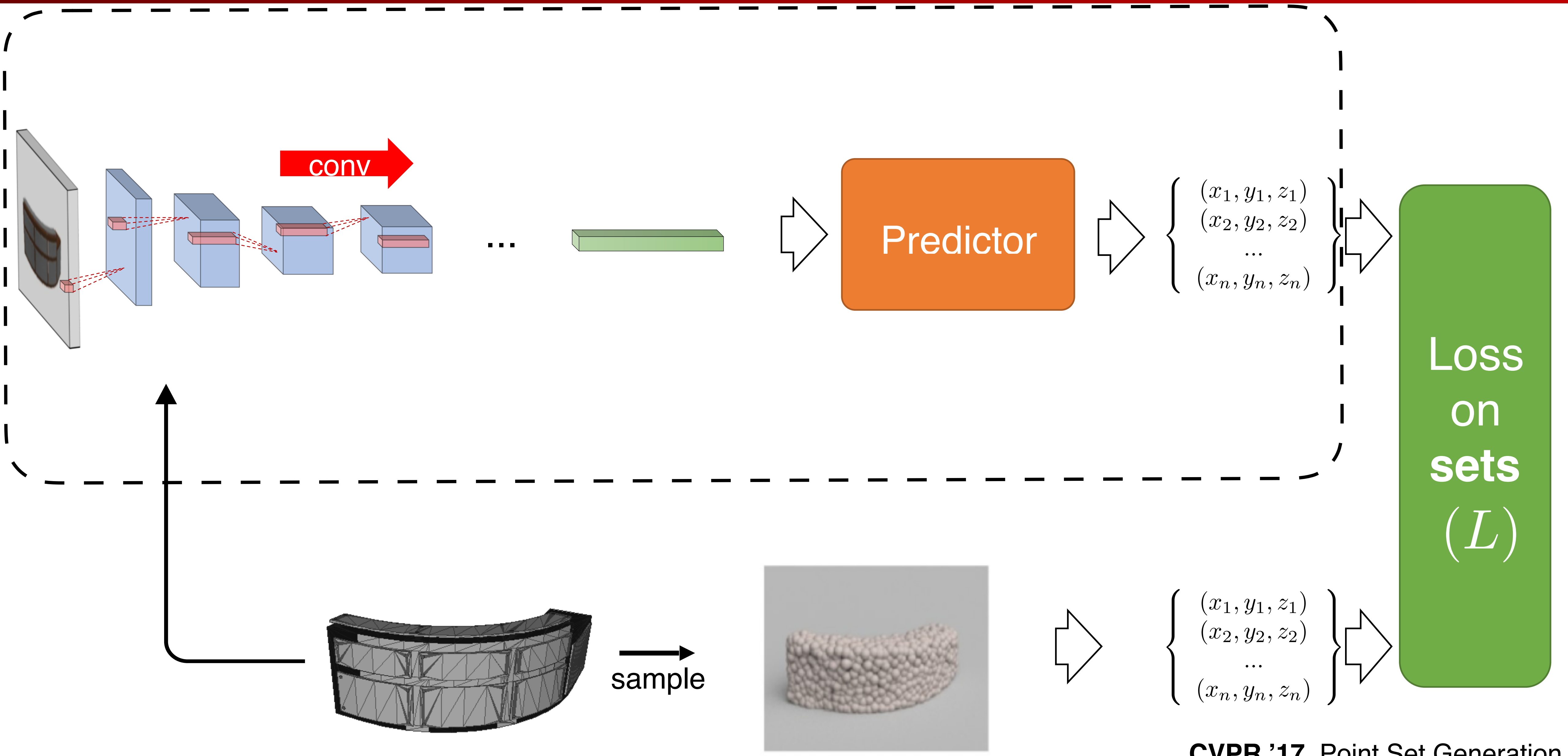
CVPR '17, Point Set Generation

Pipeline



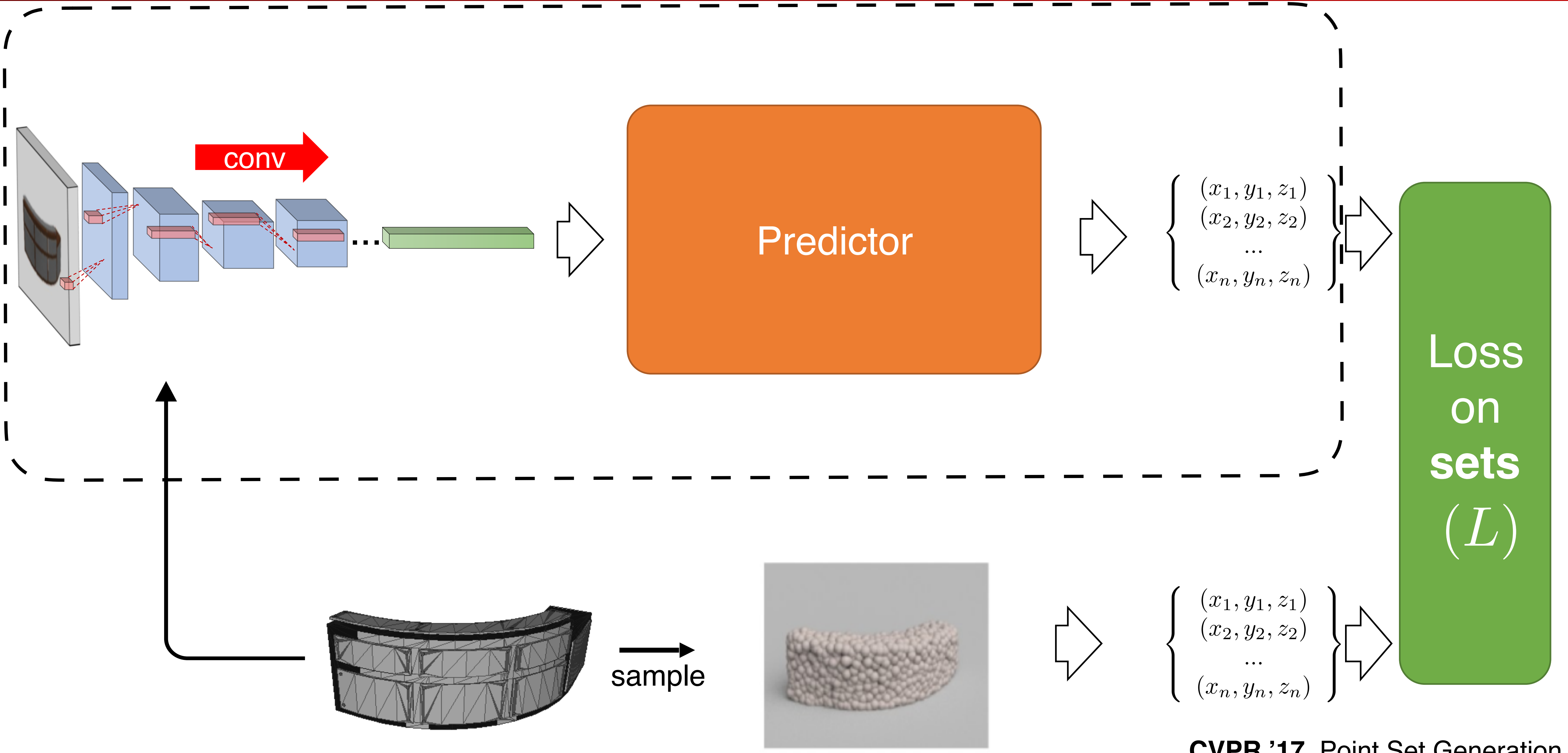
CVPR '17, Point Set Generation

Pipeline



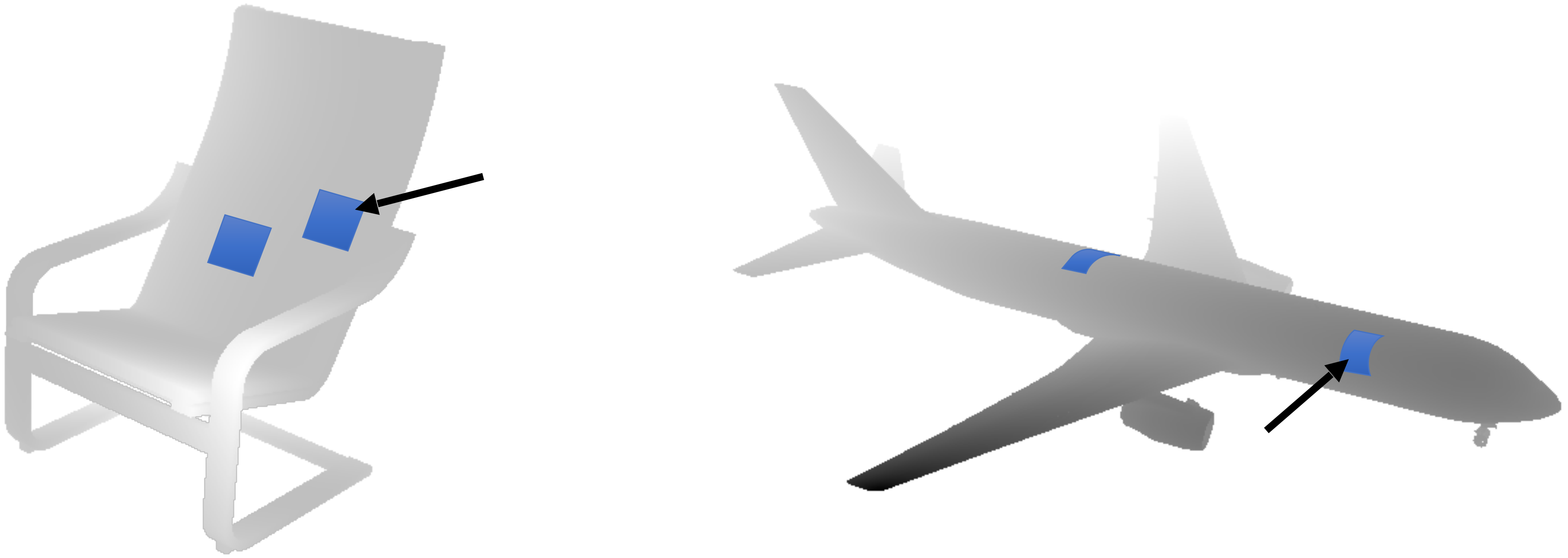
CVPR '17, Point Set Generation

Pipeline



CVPR '17, Point Set Generation

Natural statistics of geometry

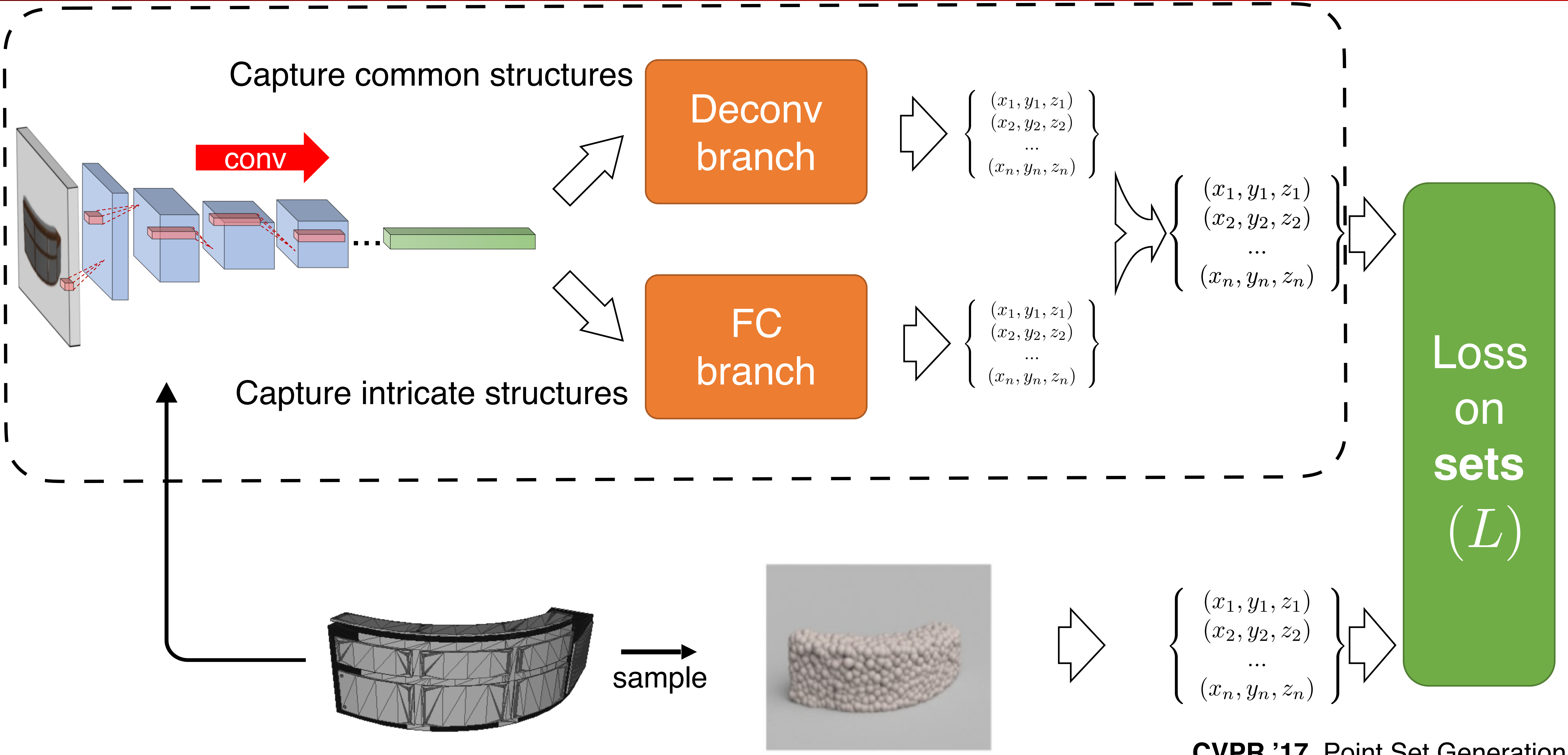


- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates

Natural statistics of geometry

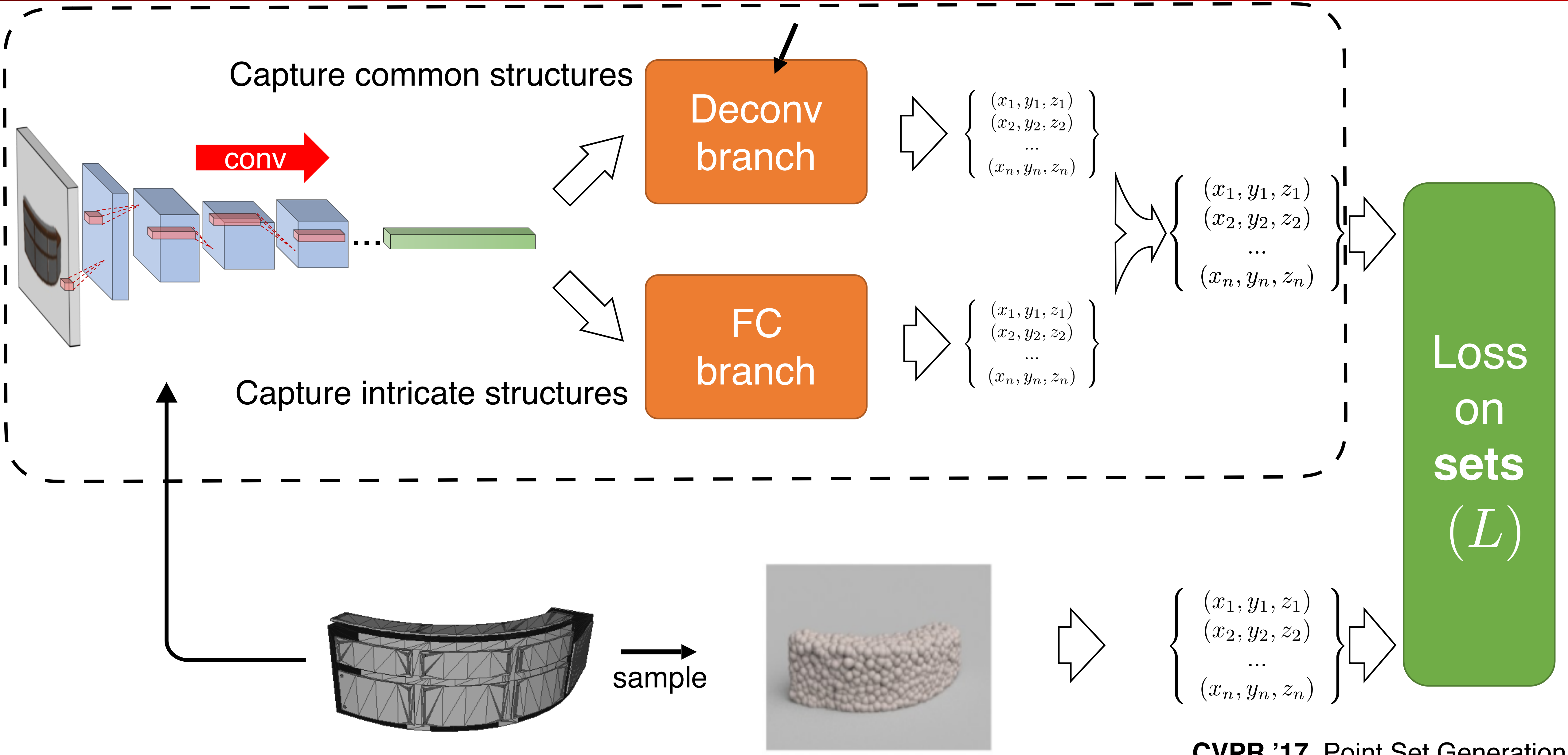
- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates
- Also some intricate structures
  - points have **high local variation**

Pipeline



CVPR '17, Point Set Generation

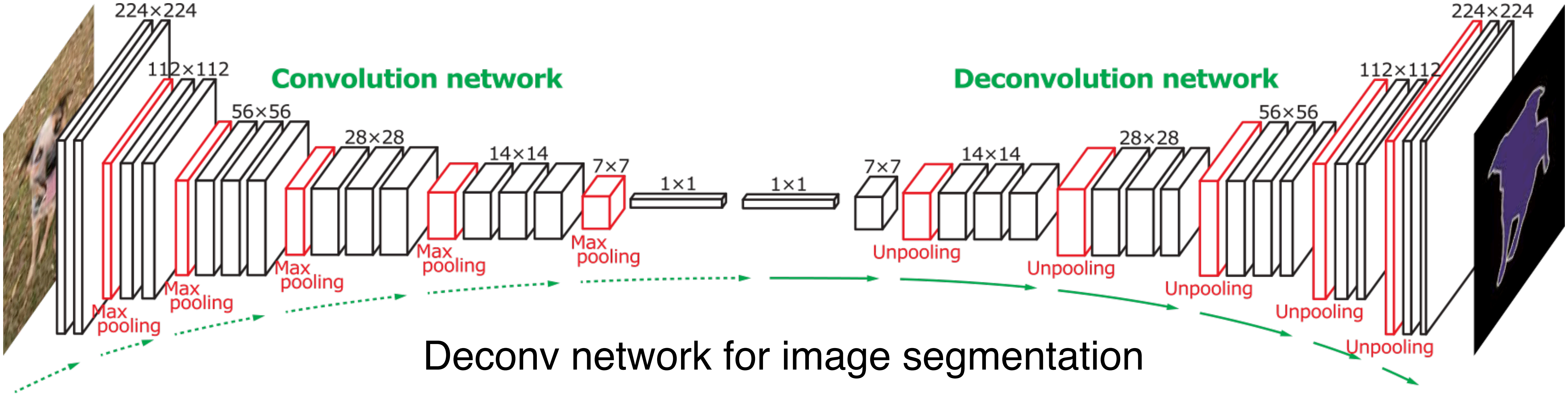
Pipeline



Capture common structures
conv
Deconv branch
FC branch
Capture intricate structures
$\{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}$
sample
Loss on sets (L)

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

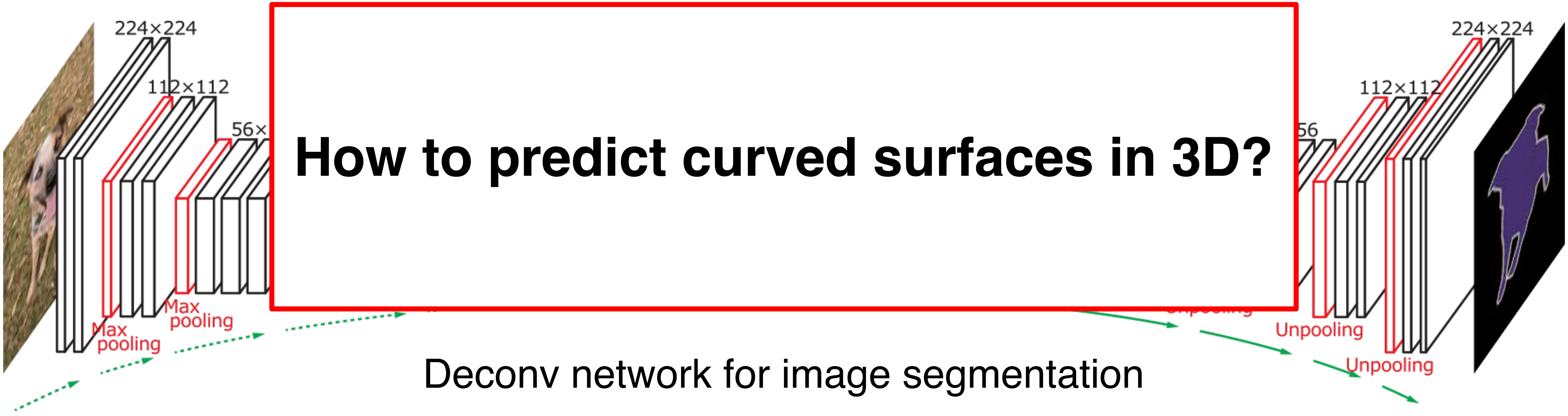


Deconv network for image segmentation

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

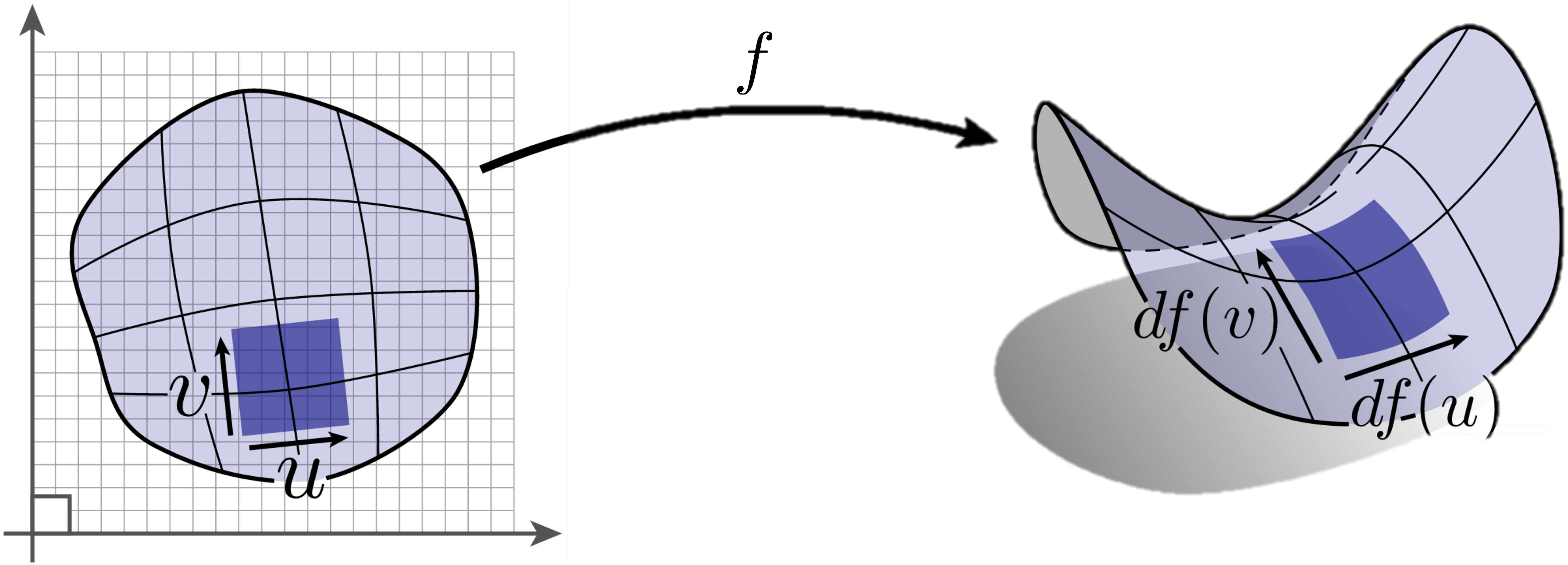


Deconv network for image segmentation

Credit: FCNN, Long et al.

Prediction of curved 2D surfaces in 3D

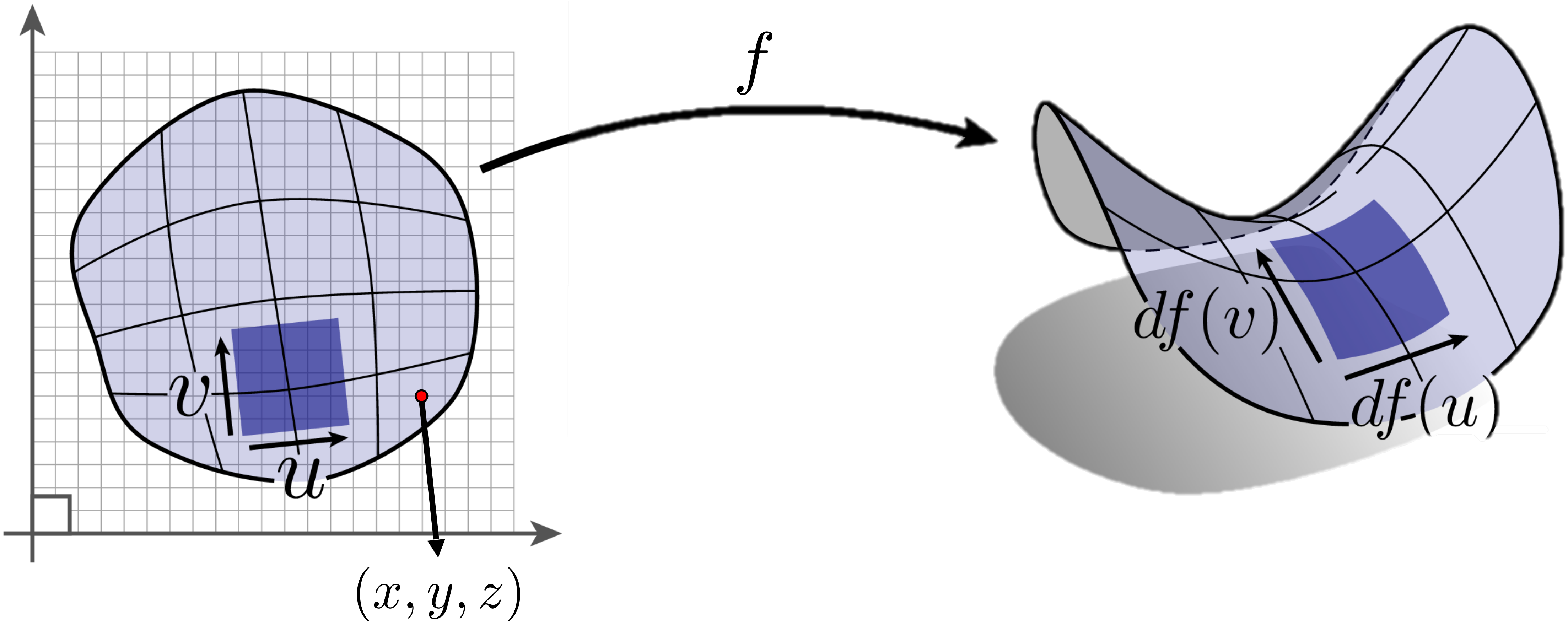
- Surface parametrization (2D$\leftrightarrow$3D mapping)



Credit: Discrete Differential Geometry, Crane et al.

Prediction of curved 2D surfaces in 3D

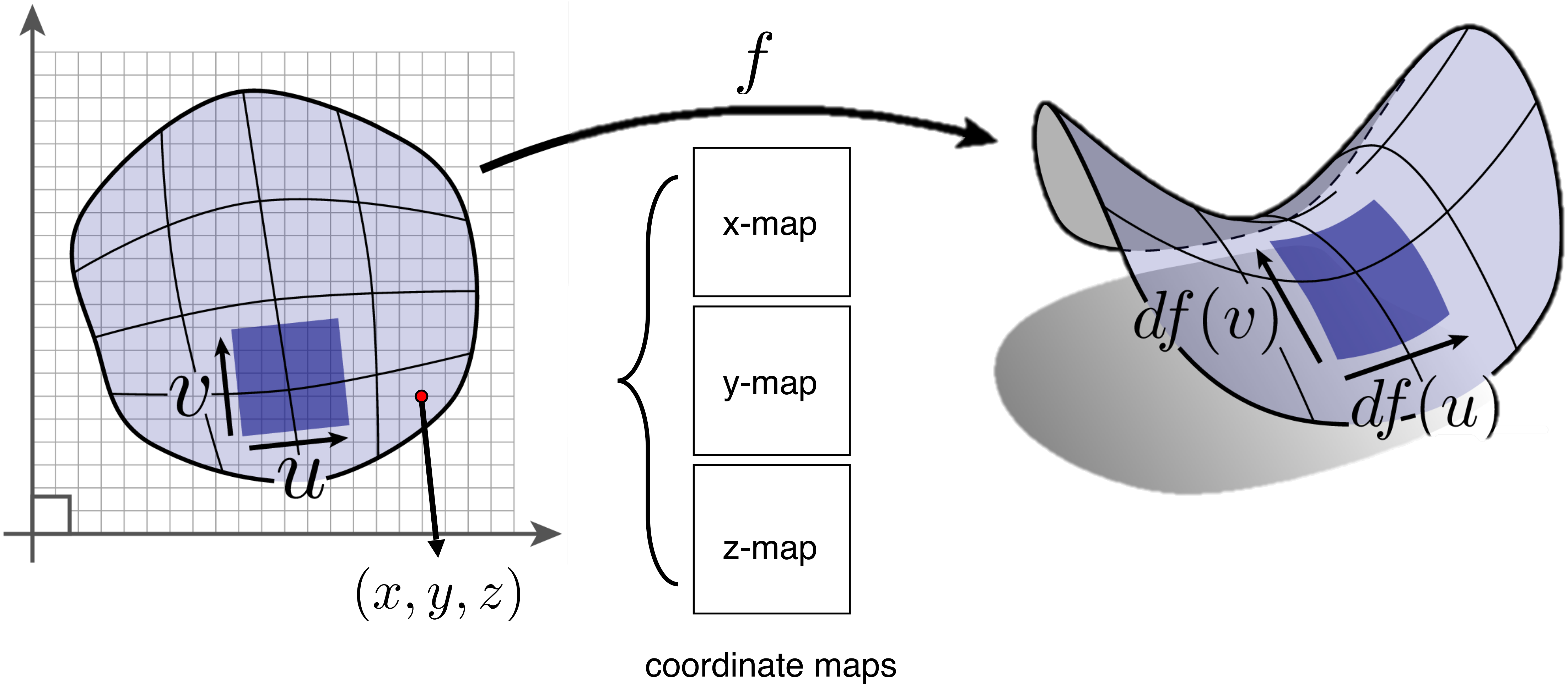
- Surface parametrization (2D$\leftrightarrow$3D mapping)



Credit: Discrete Differential Geometry, Crane et al.

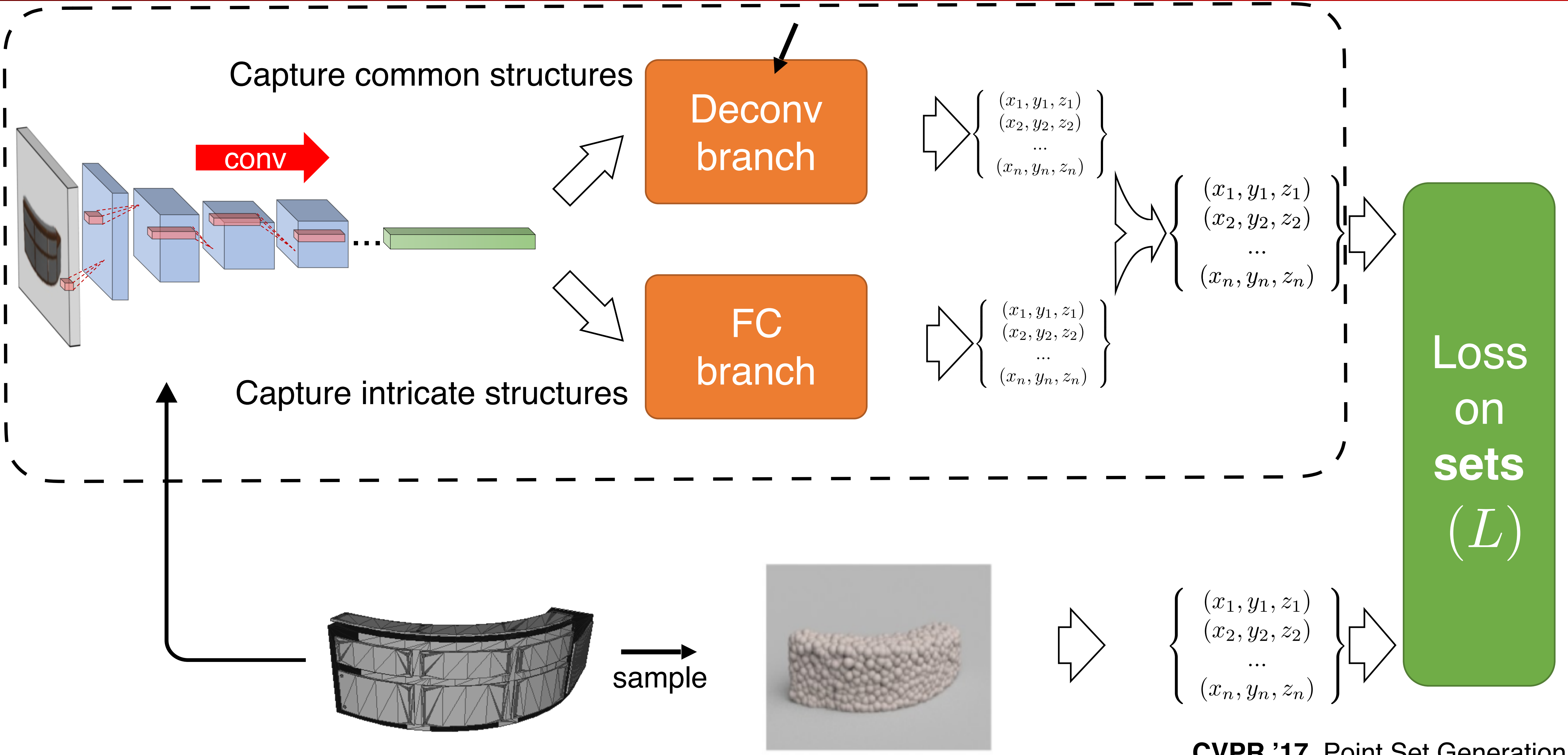
Prediction of curved 2D surfaces in 3D

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



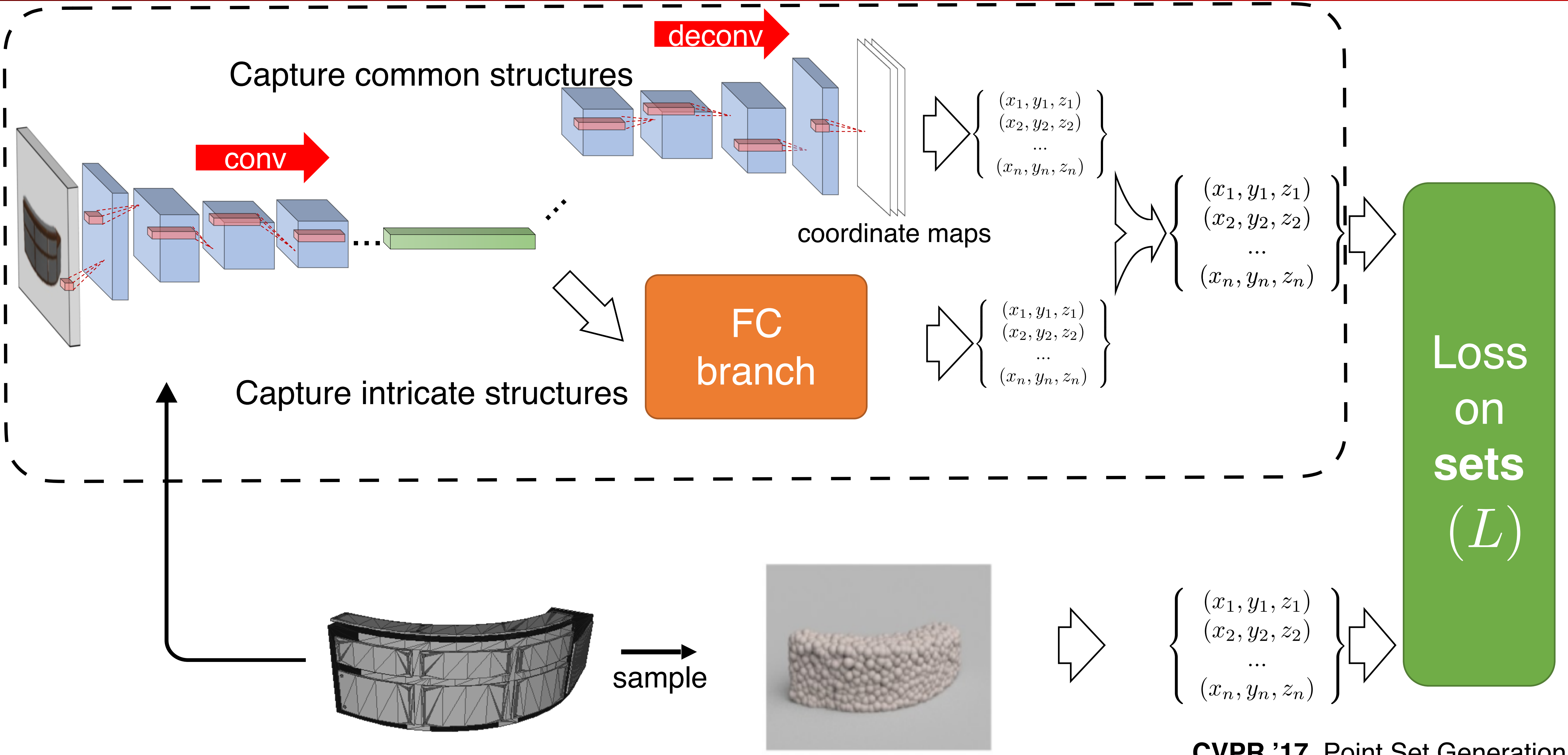
Credit: Discrete Differential Geometry, Crane et al.

Parametrization prediction by deconv network



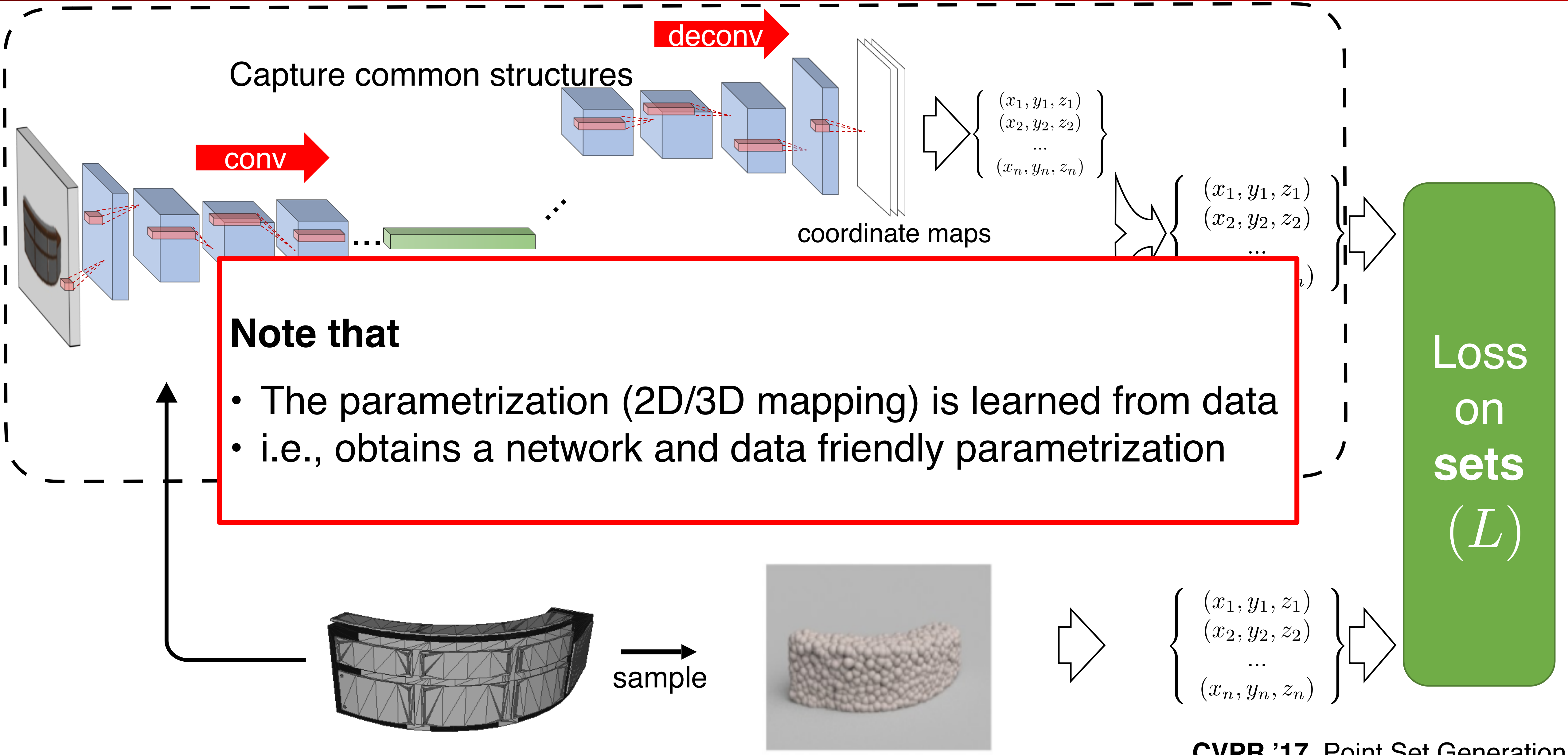
CVPR '17, Point Set Generation

Parametrization prediction by deconv network



CVPR '17, Point Set Generation

Parametrization prediction by deconv network



Note that

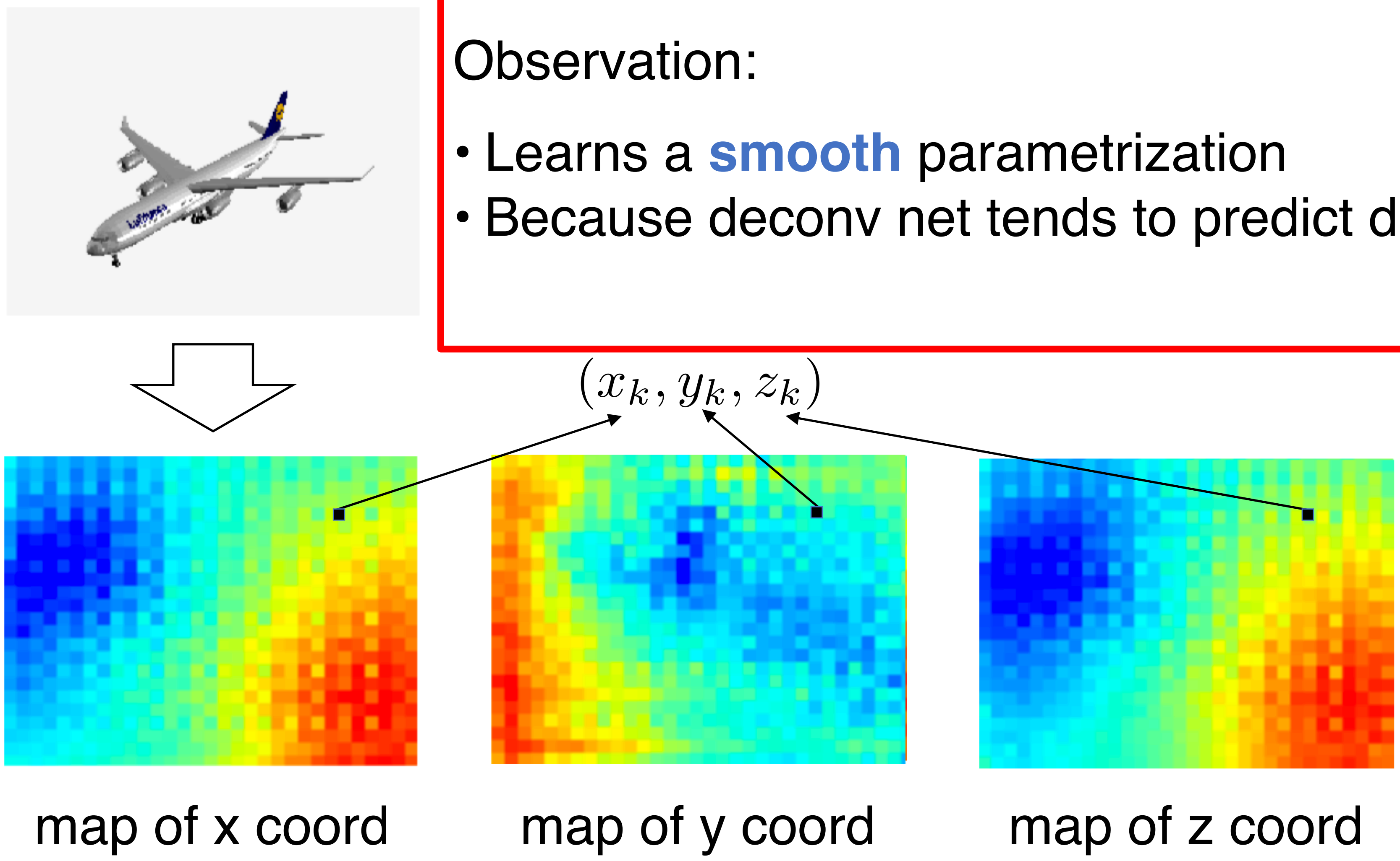
- The parametrization (2D/3D mapping) is learned from data
- i.e., obtains a network and data friendly parametrization

CVPR '17, Point Set Generation

Visualization of the learned parameterization

Observation:

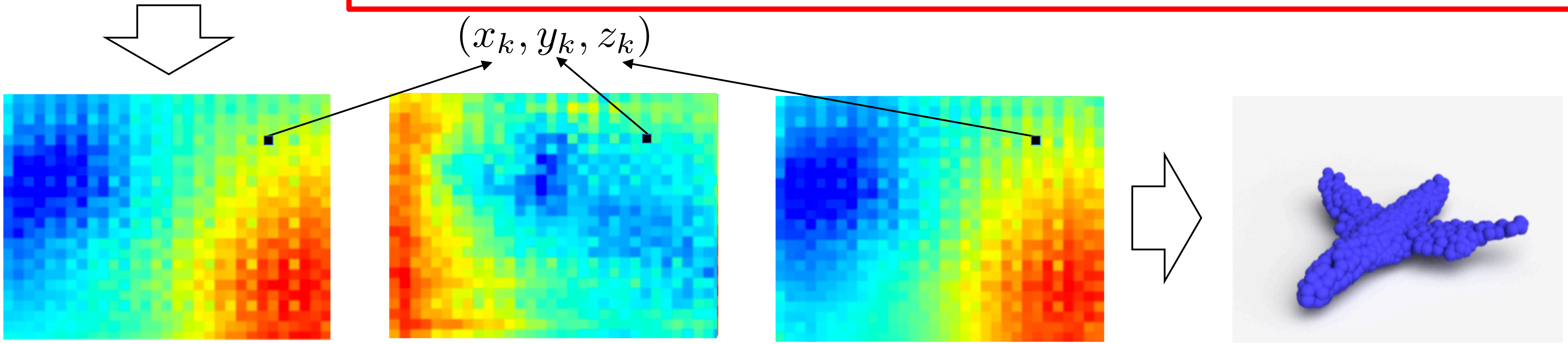
- Learns a **smooth** parametrization
- Because deconv net tends to predict data with local correlation



Visualization of the learned parameterization

Observation:

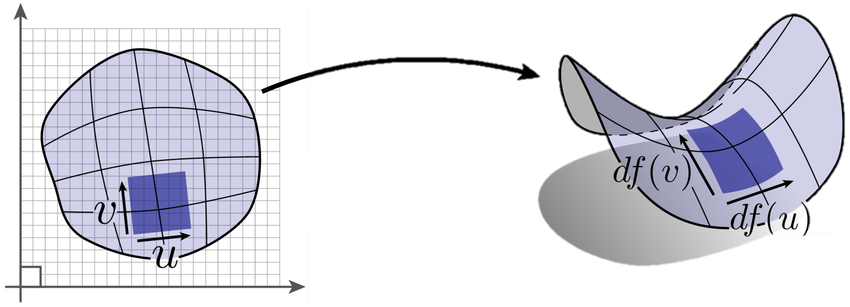
- Learns a **smooth** parametrization
- Because deconv net tends to predict data with local correlation
- Corresponds to **smooth surfaces!**

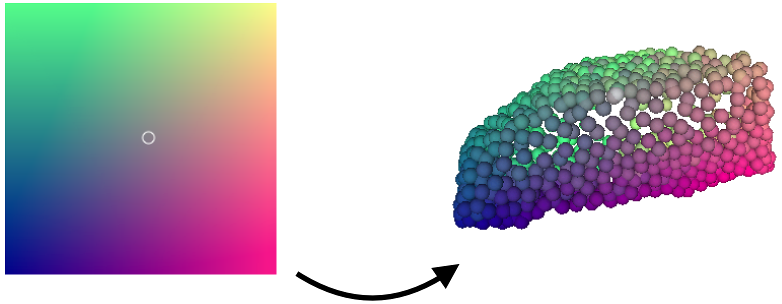


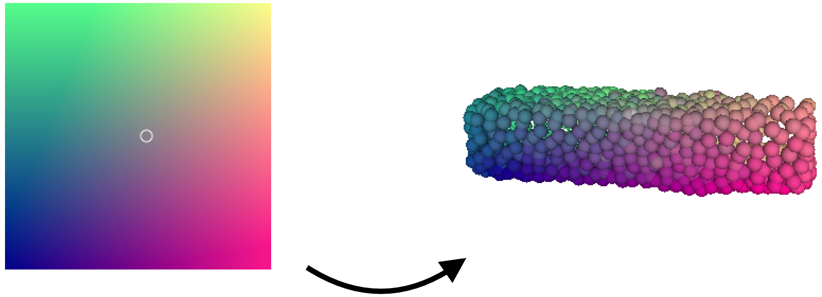
map of x coord

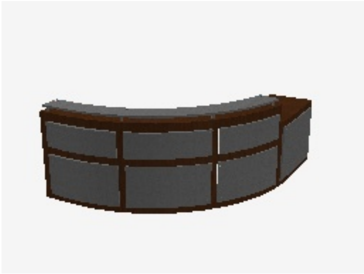
map of y coord

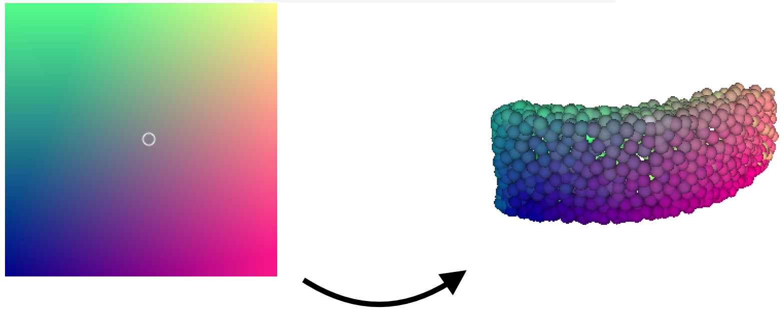
map of z coord


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





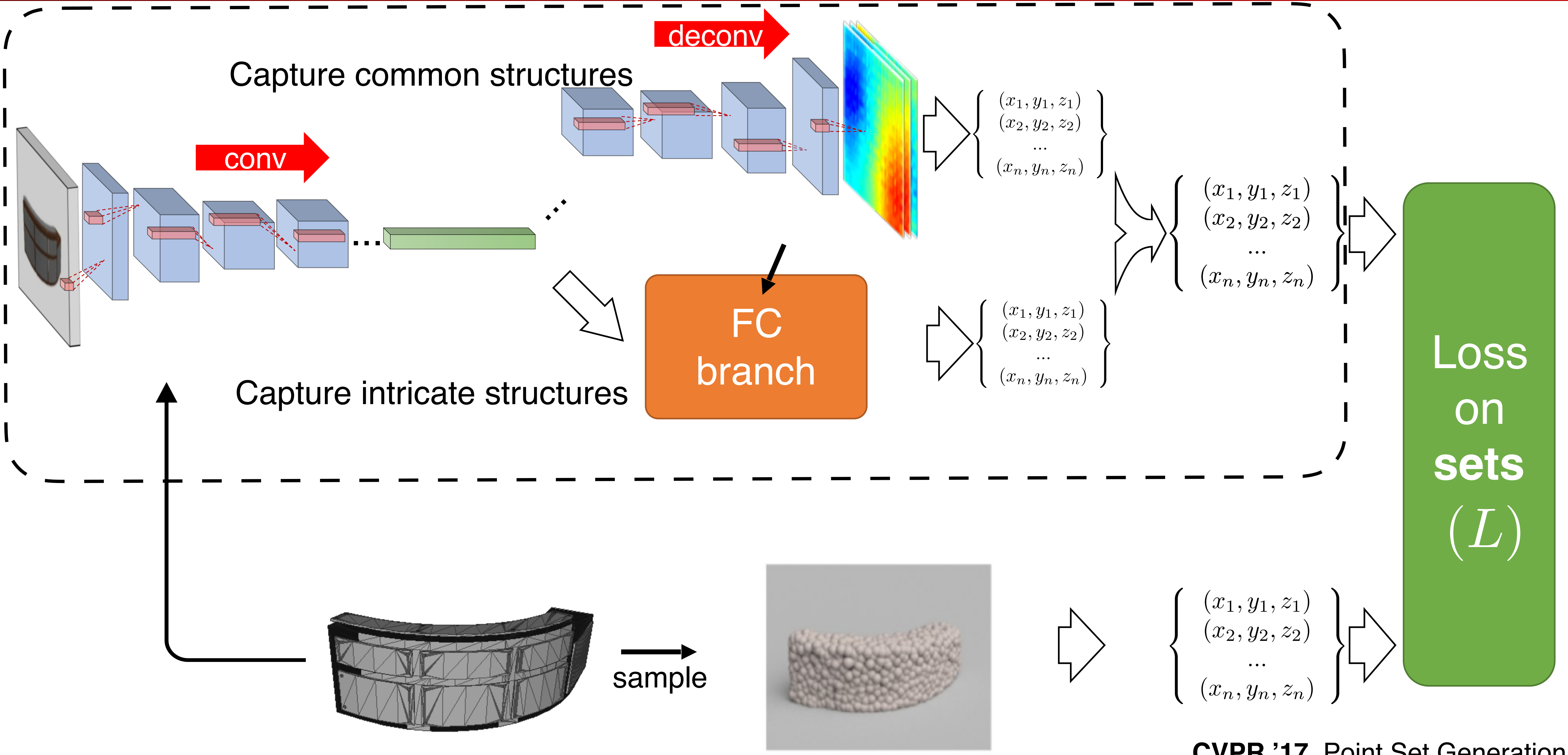




Natural statistics of geometry

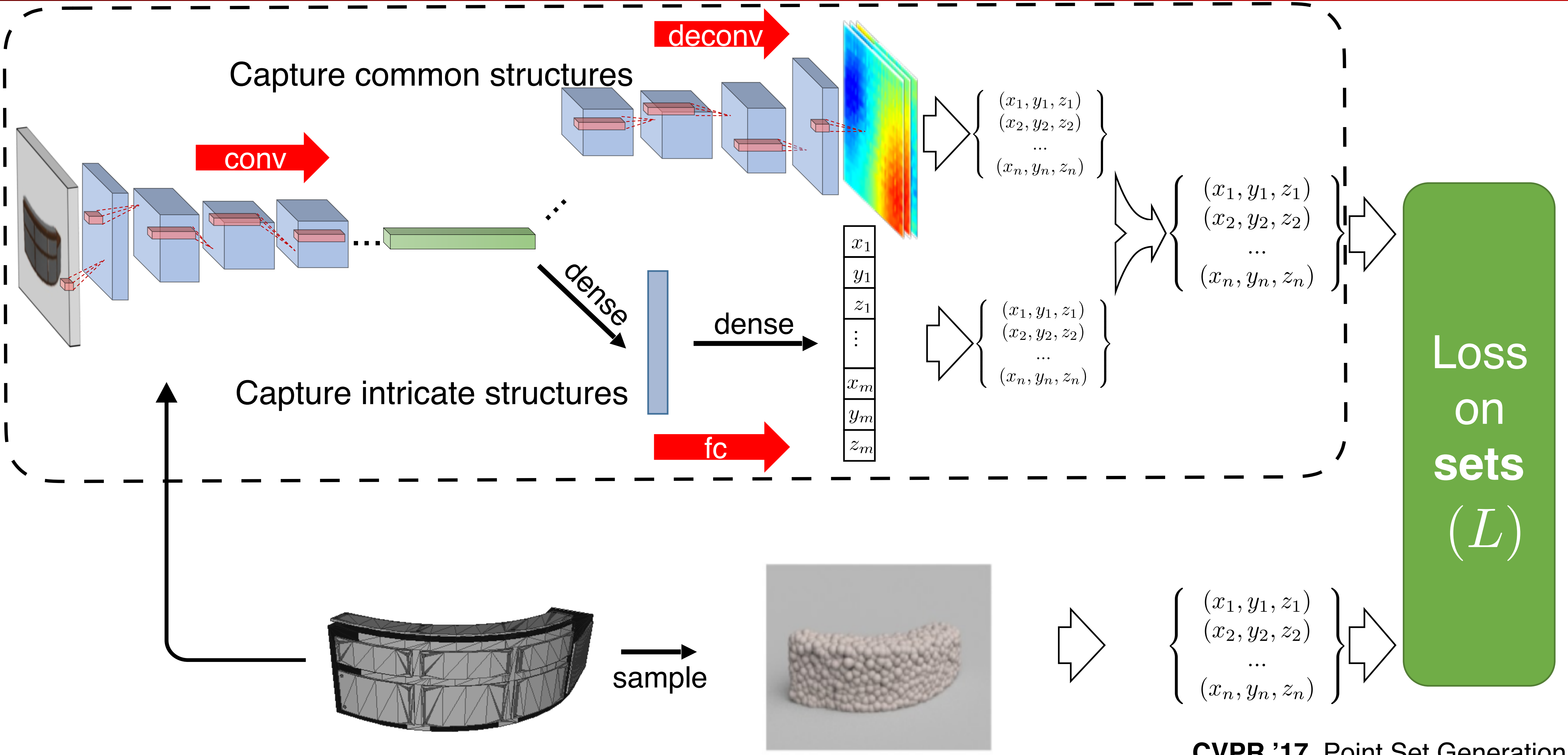
- Many local structures are common
    - e.g., planar patches, cylindrical patches
    - **strong local correlation** among point coordinates
- Also some intricate structures
    - points have **high local variation**

Pipeline



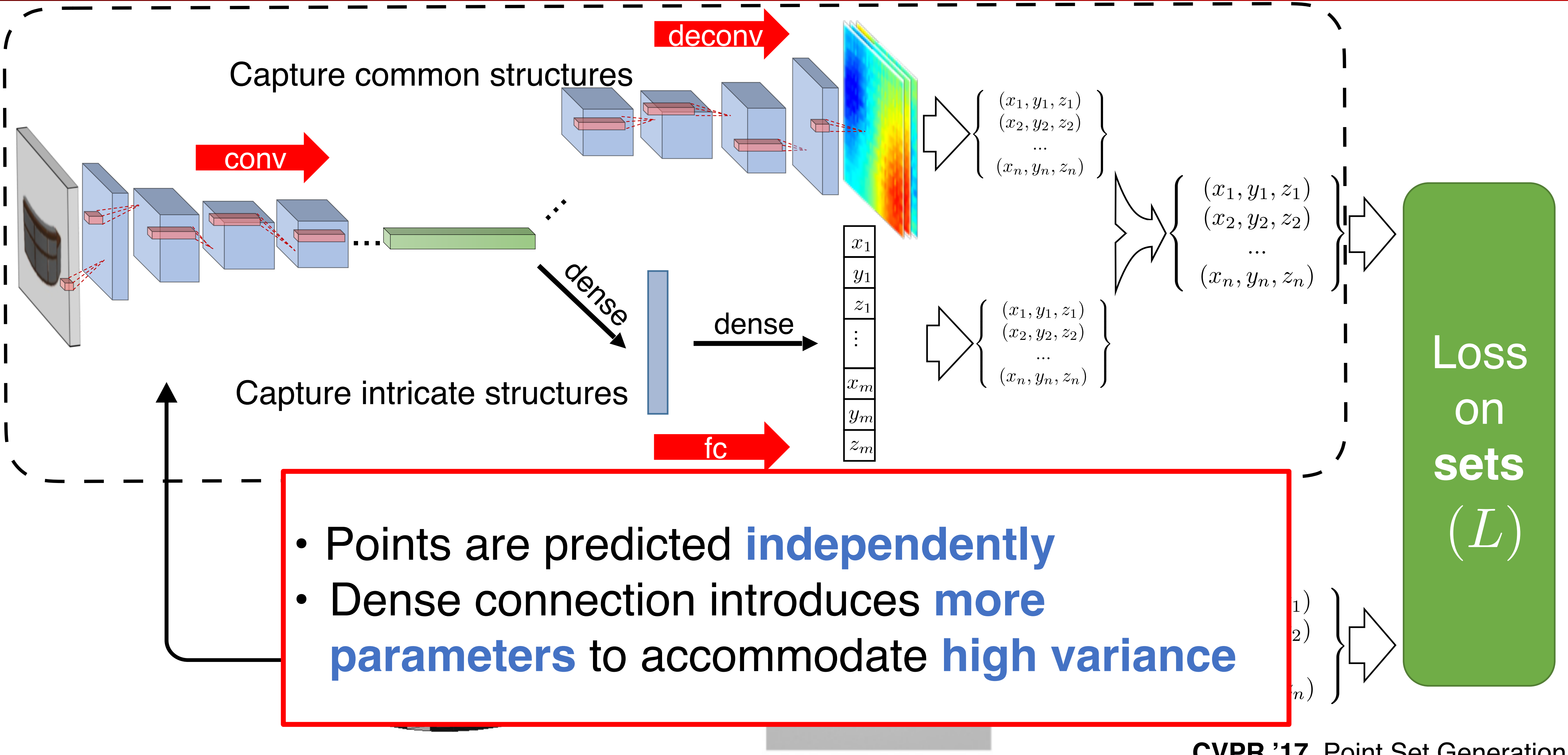
CVPR '17, Point Set Generation

Pipeline



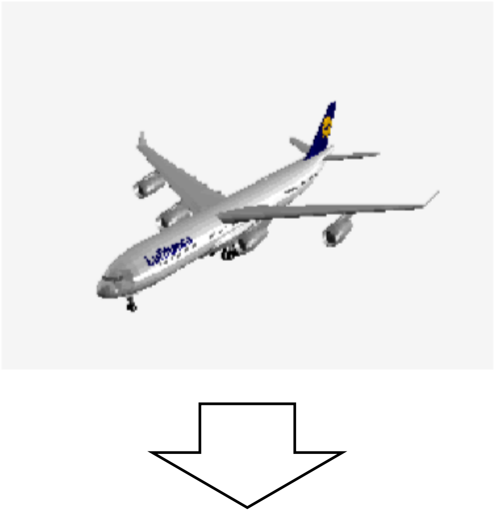
CVPR '17, Point Set Generation

Pipeline



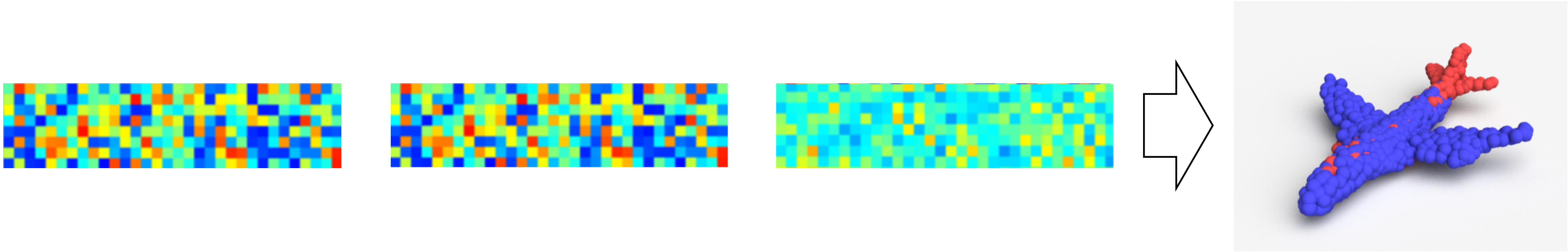
- Points are predicted **independently**
- Dense connection introduces **more parameters** to accommodate **high variance**

Visualization of the effect of FC branch



Observation:

- The arrangement of predicted points are uncorrelated



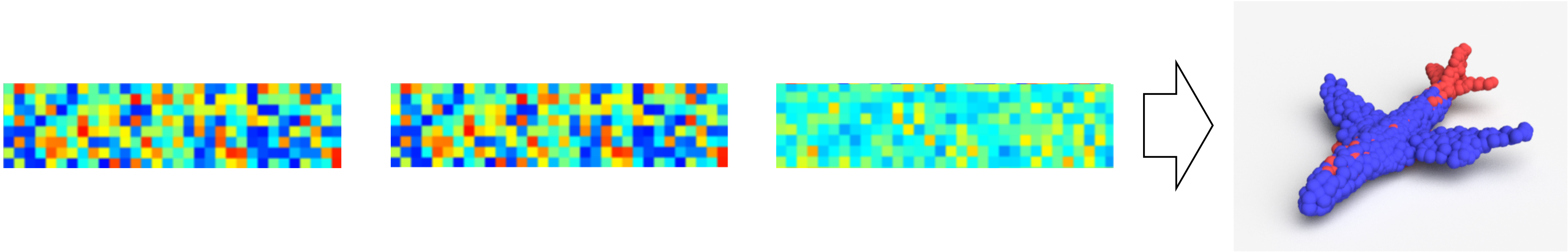
x-coord y-coord z-coord **red**

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?

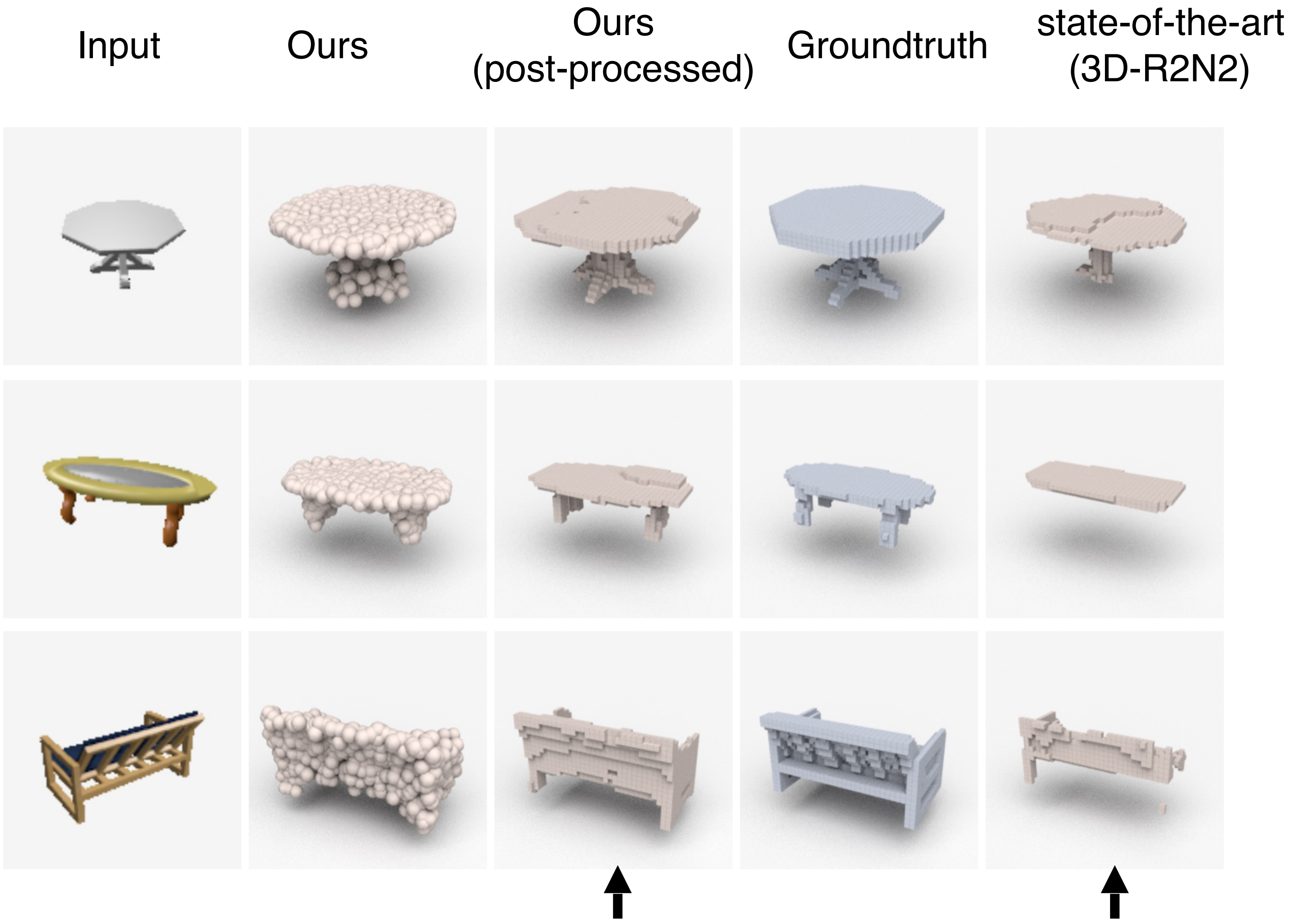
blue: deconv branch – **large, smooth** structures

red: FC branch – **intricate** structures

CVPR '17, Point Set Generation

Comparison to state-of-the-art

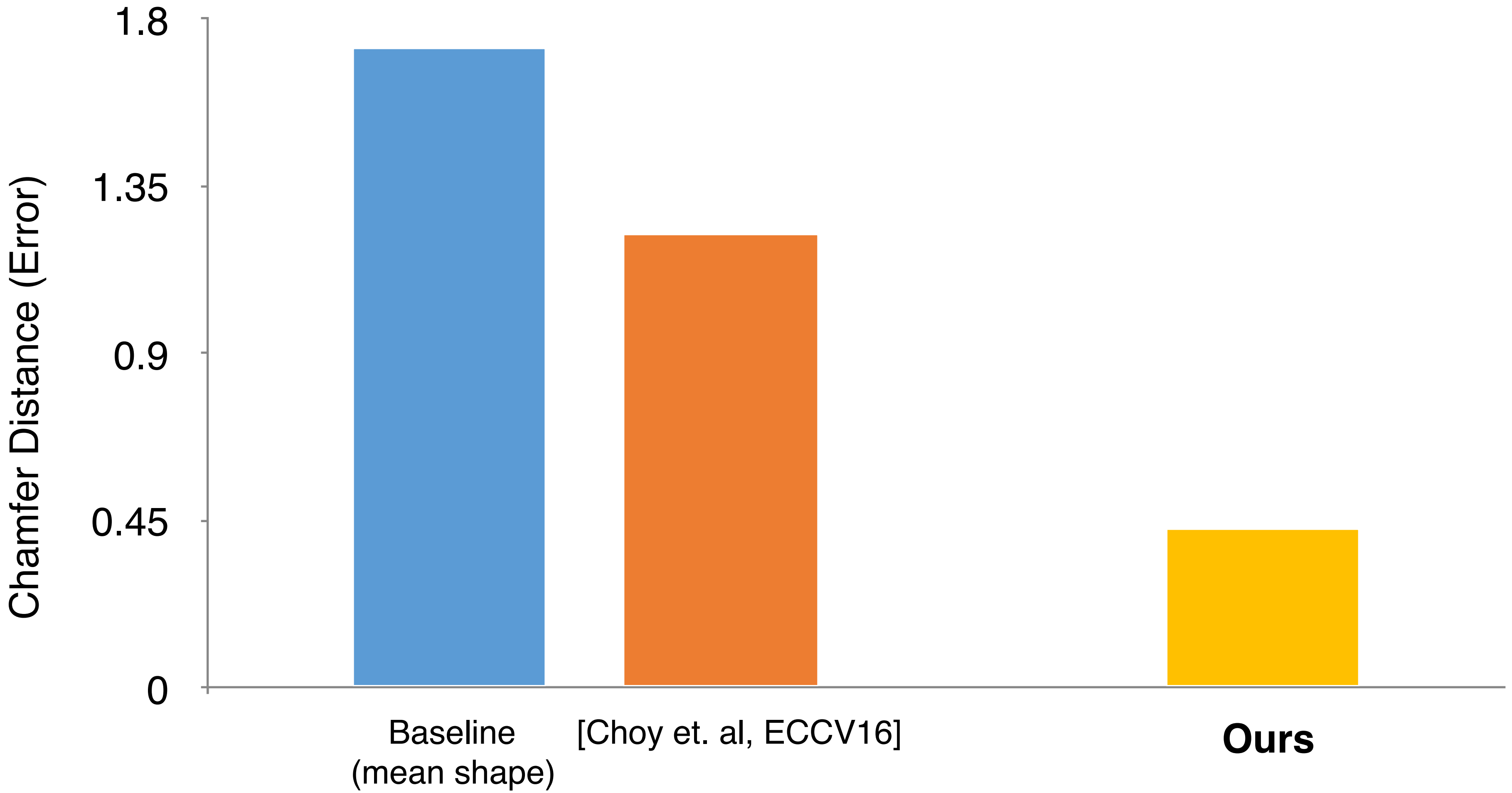
- Better global structure
- Better details



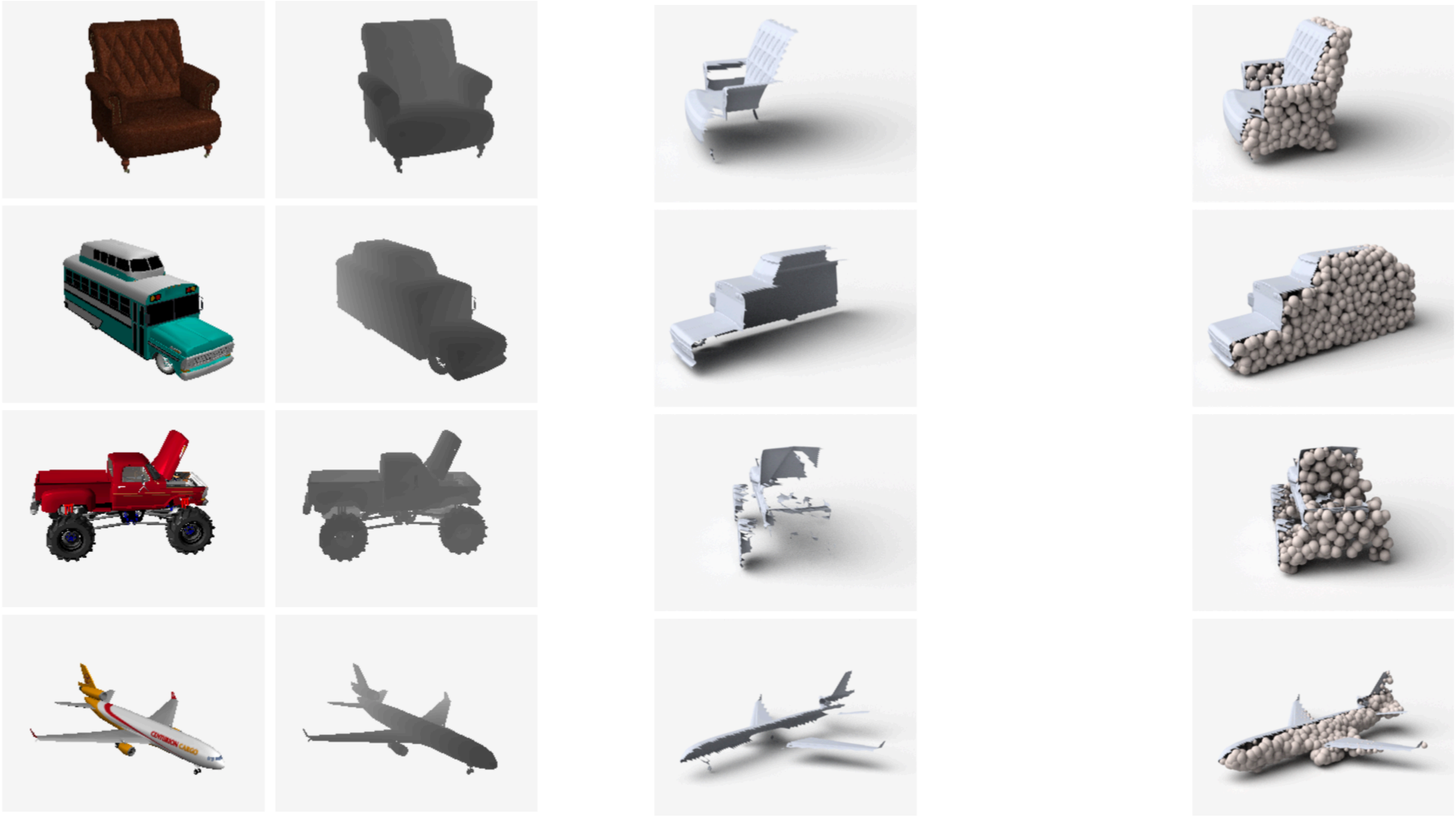
CVPR '17, Point Set Generation

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

How about learning to predict geometric forms?

Candidates:

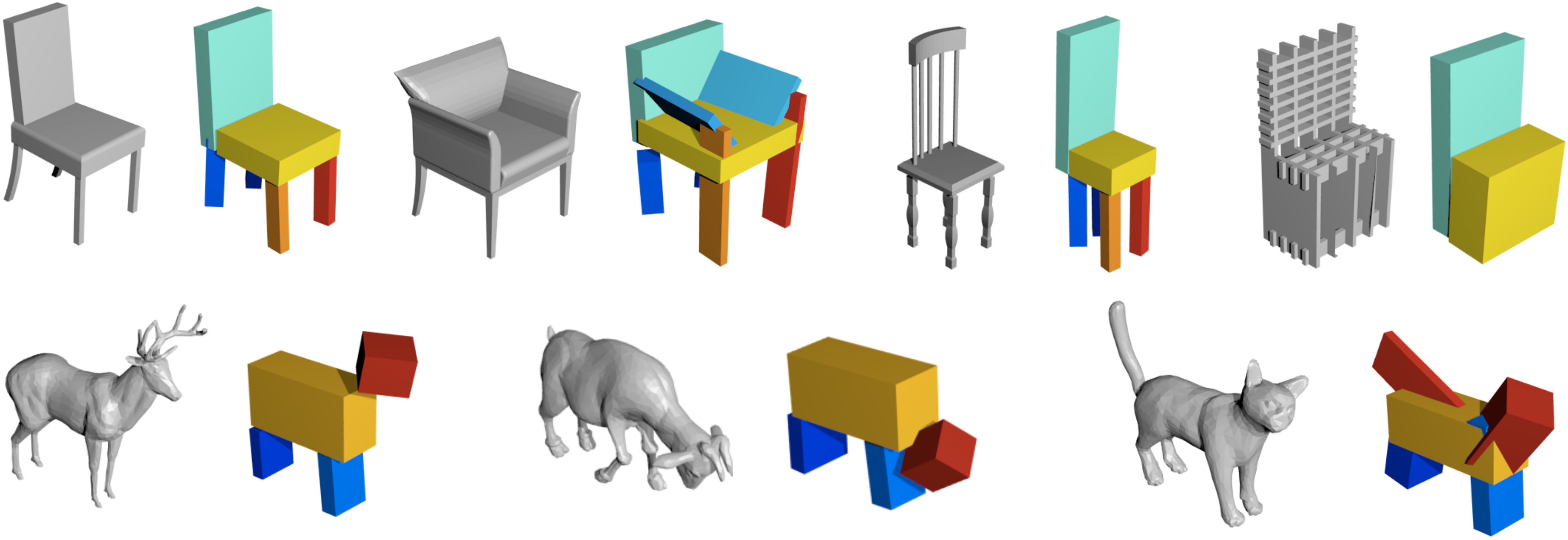
Rasterized form (regular grids)

- multi-view images
- depth map
- volumetric

Geometric form (irregular)

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

Primitive-based assembly

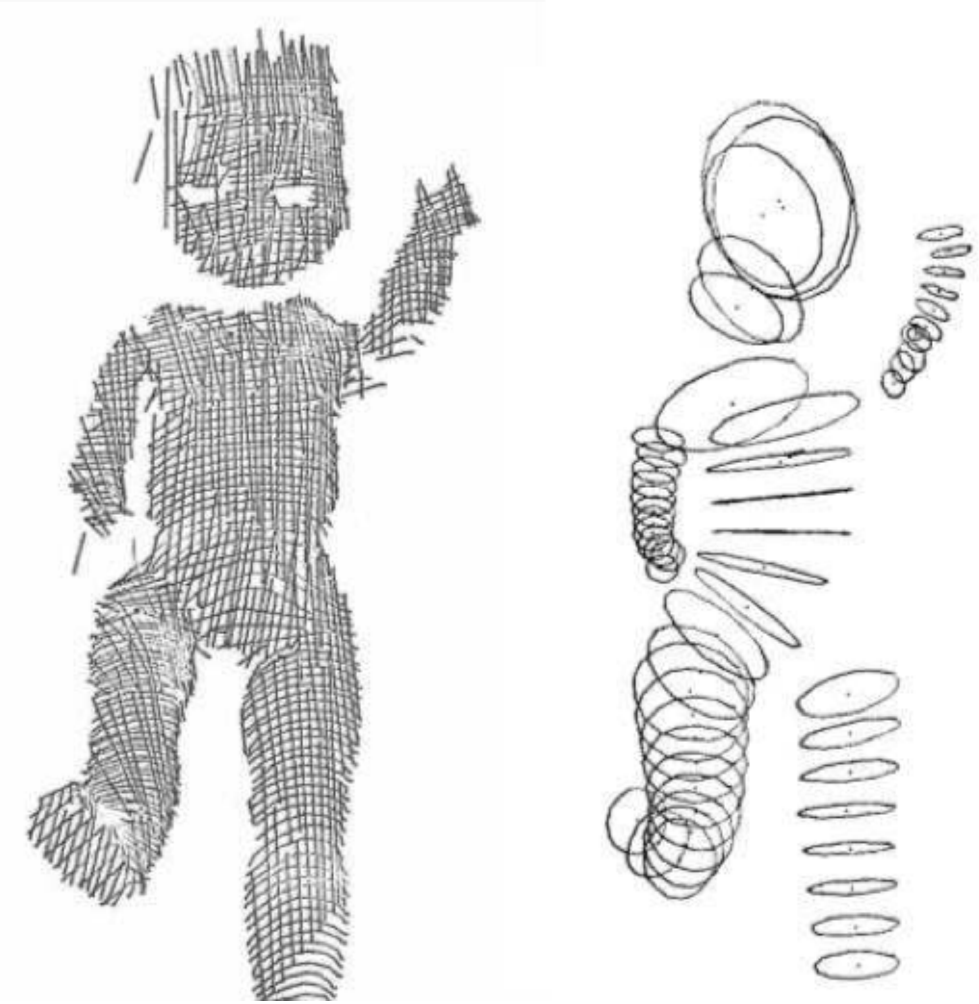


We **learn** to predict a corresponding shape composed by primitives.
It allows us to predict **consistent** compositions across objects.

Unsupervised parsing

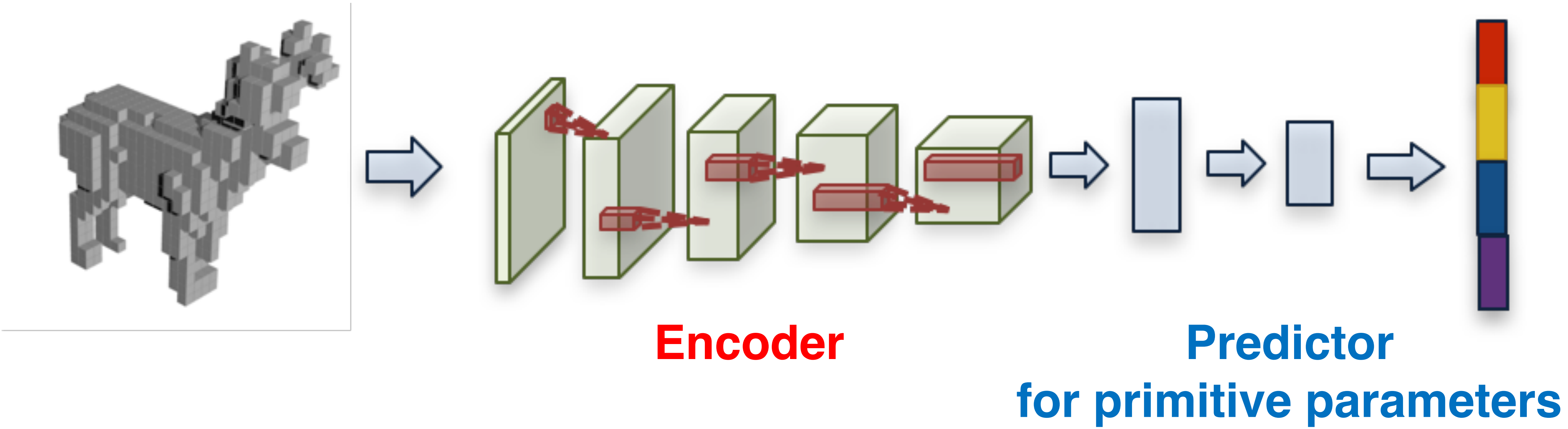
Each point is colored according to the assigned primitive

A historical overview



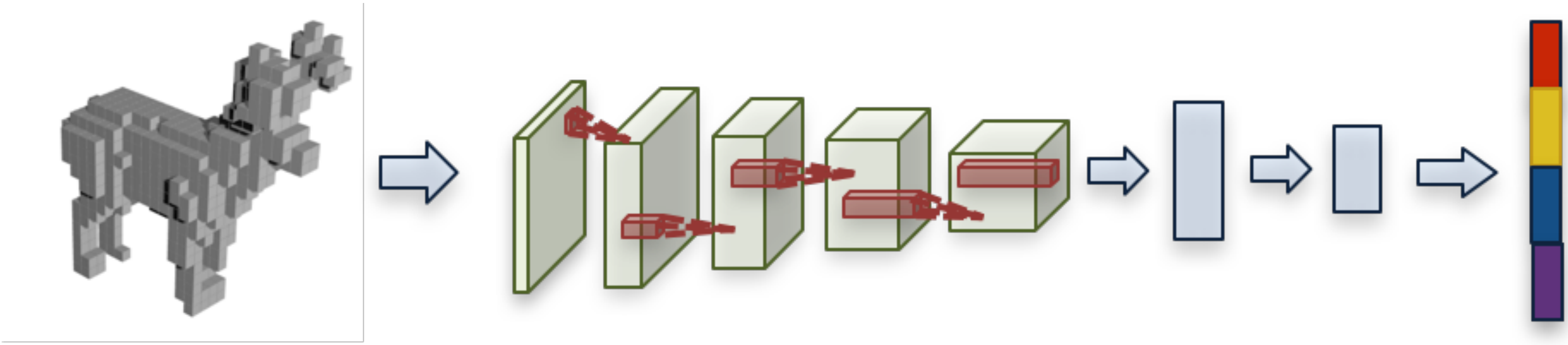
Generalized Cylinders, Binford (1971)

Approach



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

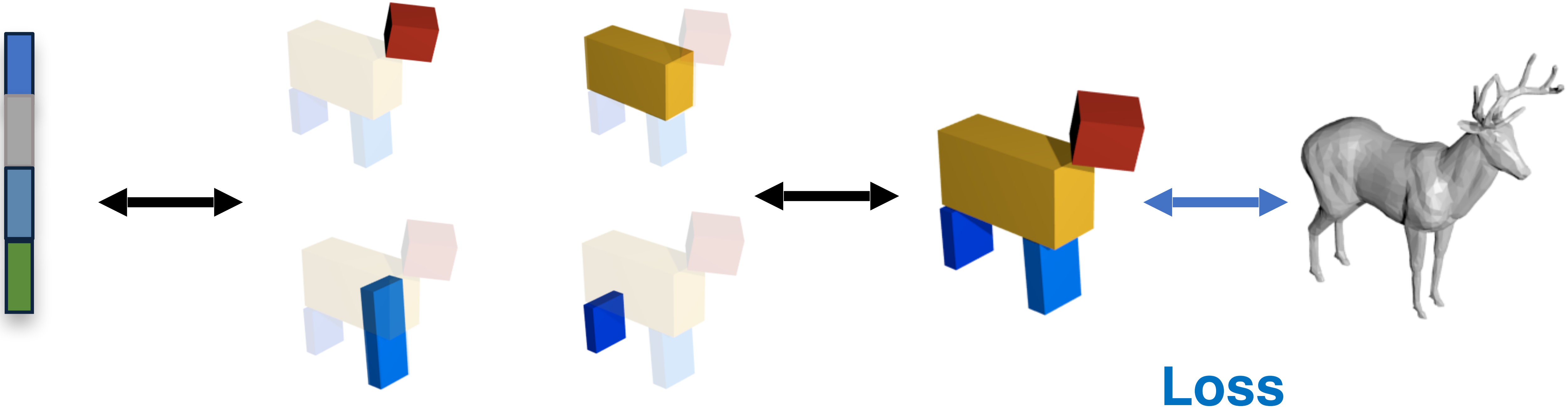
Approach



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

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

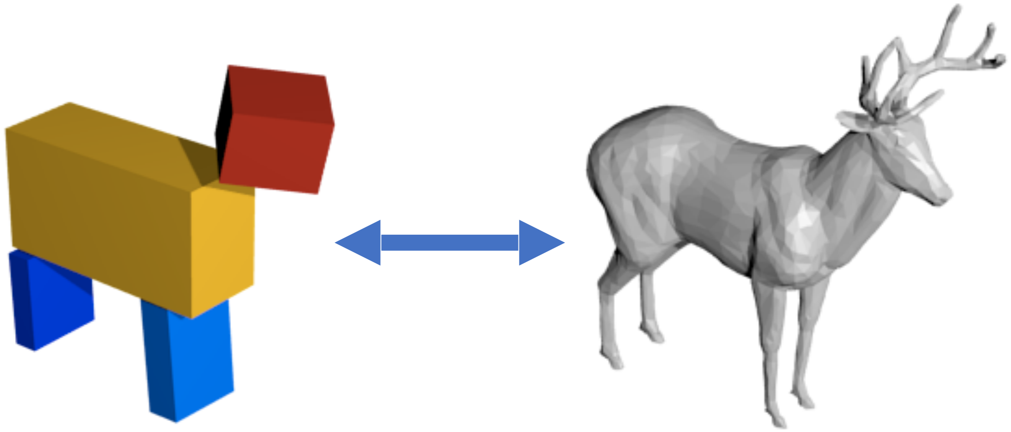
Loss function



Loss function construction

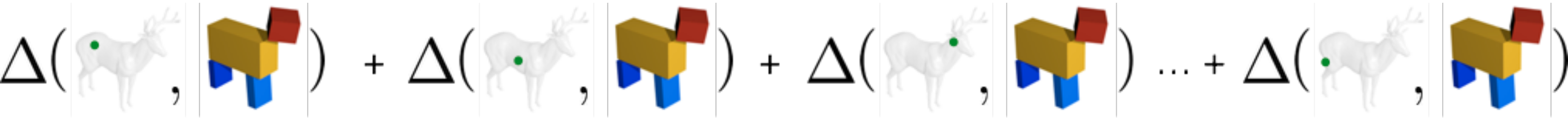
Basic idea: **Chamfer distance!**

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



Loss function construction

Sample points on the groundtruth mesh and predicted assembly

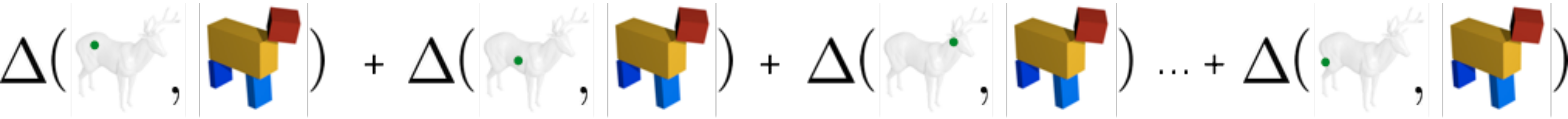


Each point is a **linear function** of mesh/primitive vertex coordinates

Differentiable!

Loss function construction

Sample points on the groundtruth mesh and predicted assembly

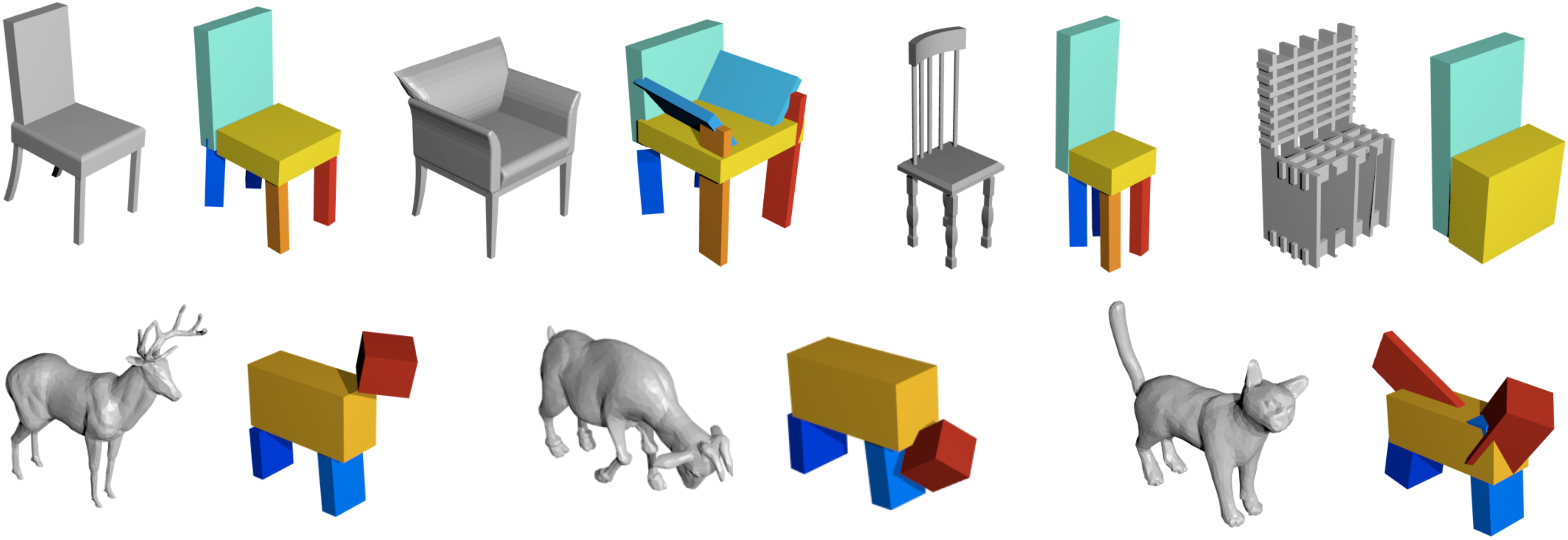


Each point is a **linear function** of mesh/primitive vertex coordinates

Differentiable!

Speed up the computation leveraging parameterization of primitives

Consistent primitive configurations



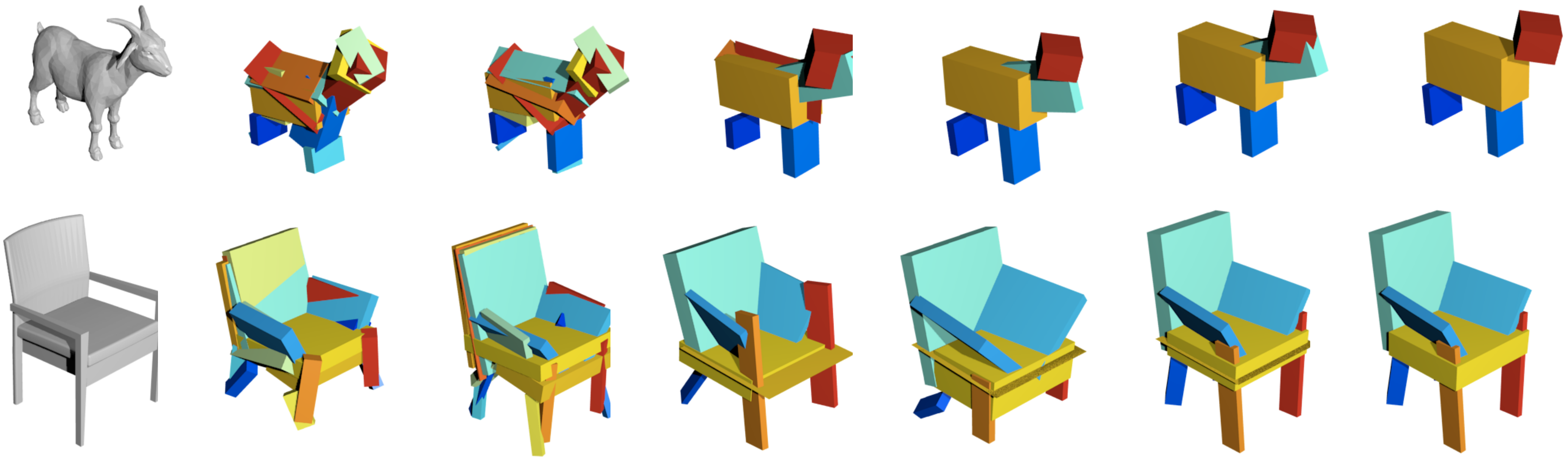
Primitive locations are **consistent** due to
the **smoothness** of primitive prediction network

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

Agenda

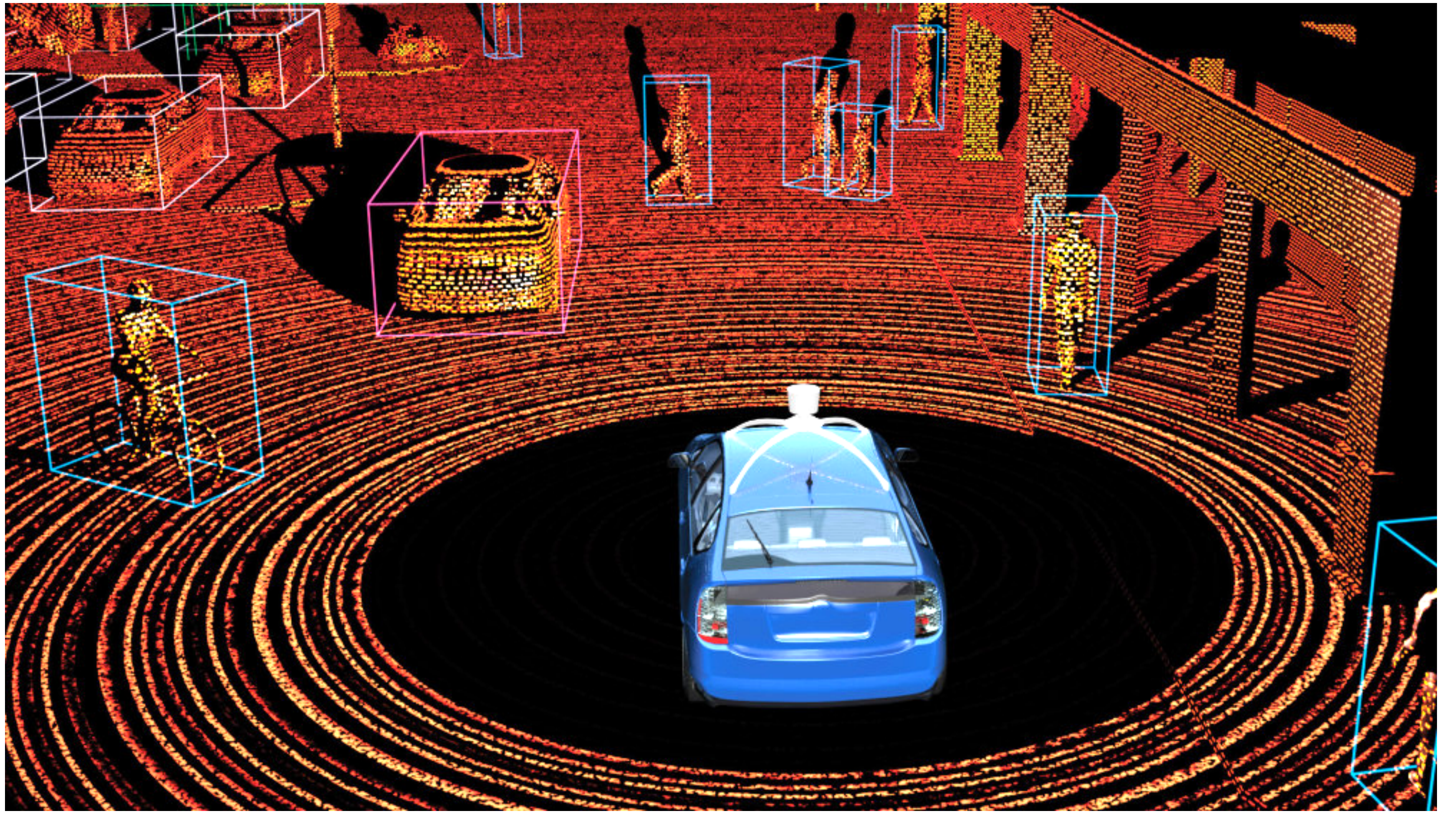
- Point cloud generation

- **Point cloud analysis**

Applications of Point Set Learning

- **Robot Perception**

What and where are the objects in a LiDAR scanned scene?

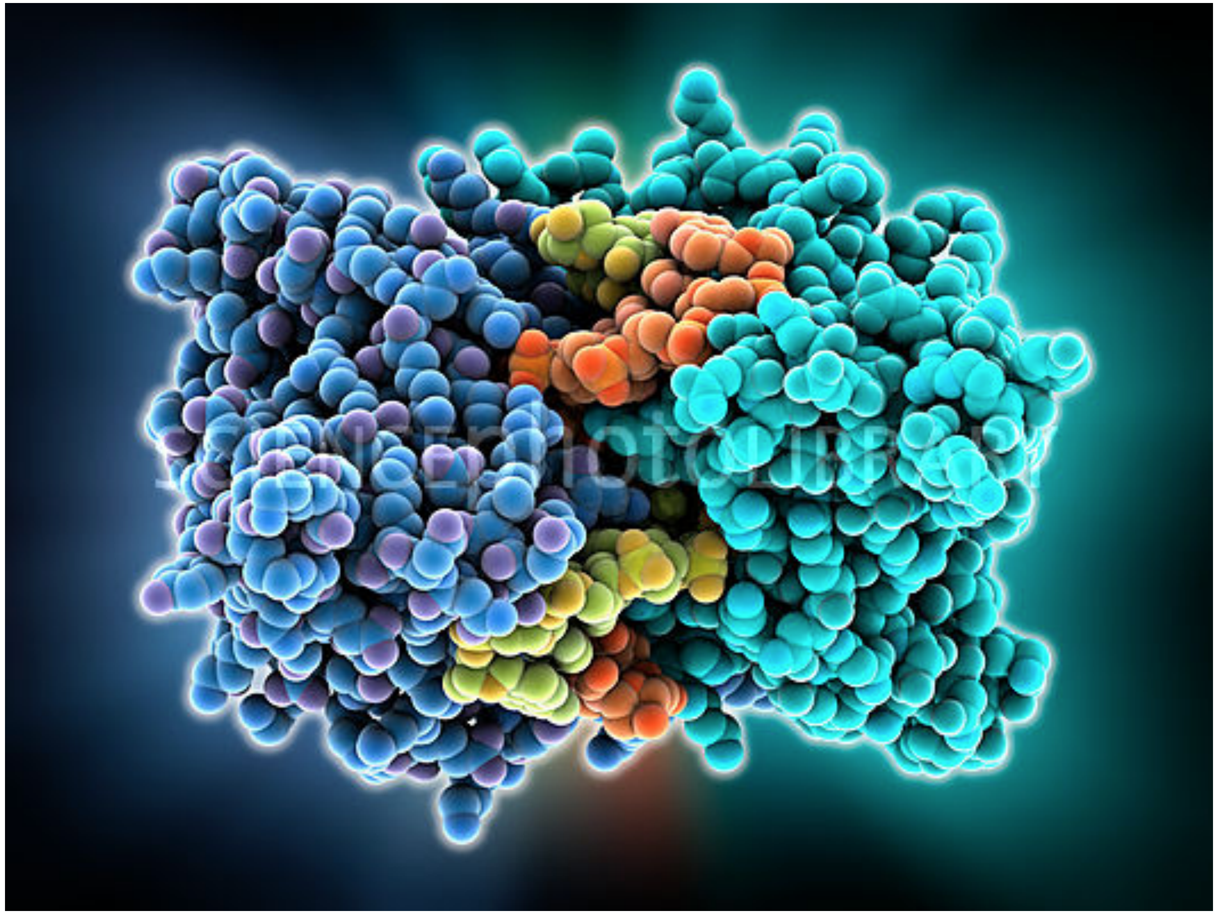


https://3dprint.com/116569/self-driving-cars-privacy/

Applications of Point Set Learning

- **Molecular Biology**

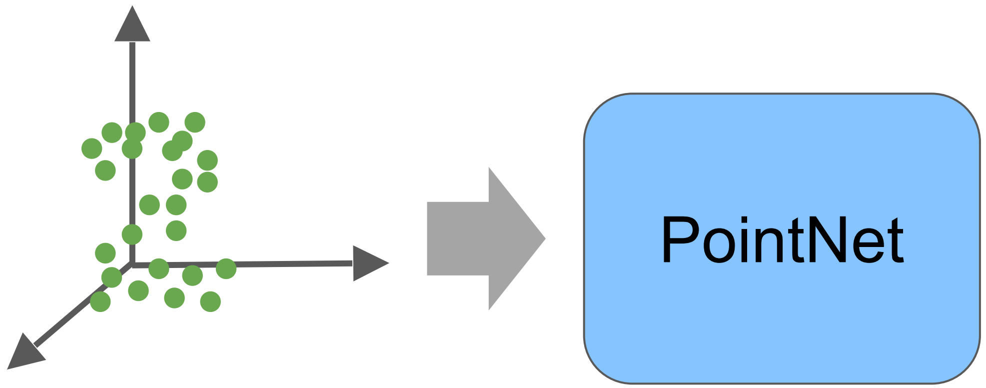
Can we infer an enzyme's category (reactions they catalyze) from its structure?



EcoRV restriction enzyme molecule, LAGUNA DESIGN/SCIENCE PHOTO LIBRARY

Directly process point cloud data

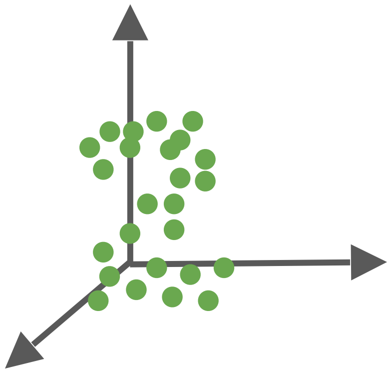
End-to-end learning for **unstructured, unordered** point data



Directly process point cloud data

End-to-end learning for **unstructured, unordered** point data

Unified framework for various tasks



PointNet

Object Classification

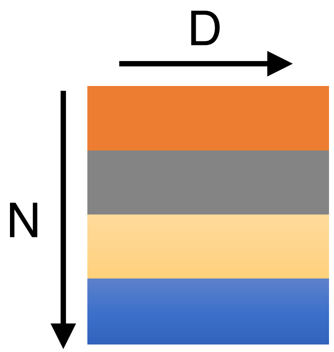
Part Segmentation

Scene Parsing

...

Properties of a desired neural network on point clouds

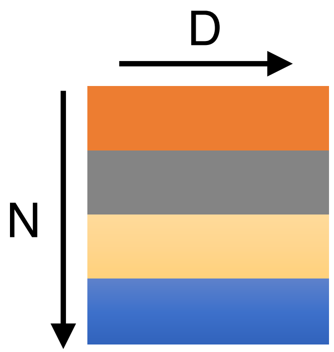
Point cloud: N **orderless** points, each represented by a D dim coordinate



2D array representation

Properties of a desired neural network on point clouds

Point cloud: N **orderless** points, each represented by a D dim coordinate



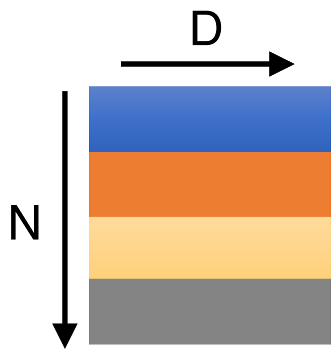
2D array representation

Permutation invariance

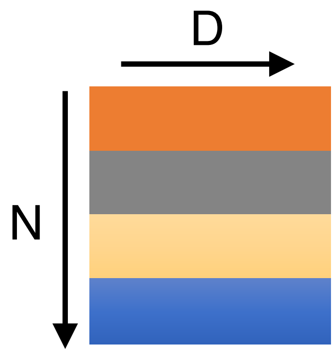
Transformation invariance

Properties of a desired neural network on point clouds

Point cloud: N **orderless** points, each represented by a D dim coordinate



represents the same **set** as



2D array representation

Permutation invariance

Permutation invariance: Symmetric function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

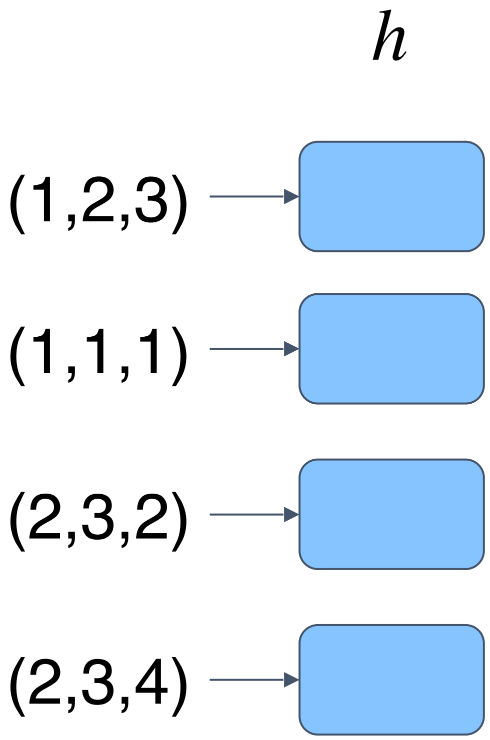
$\ldots$

Construct symmetric function family

Observe: $f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric

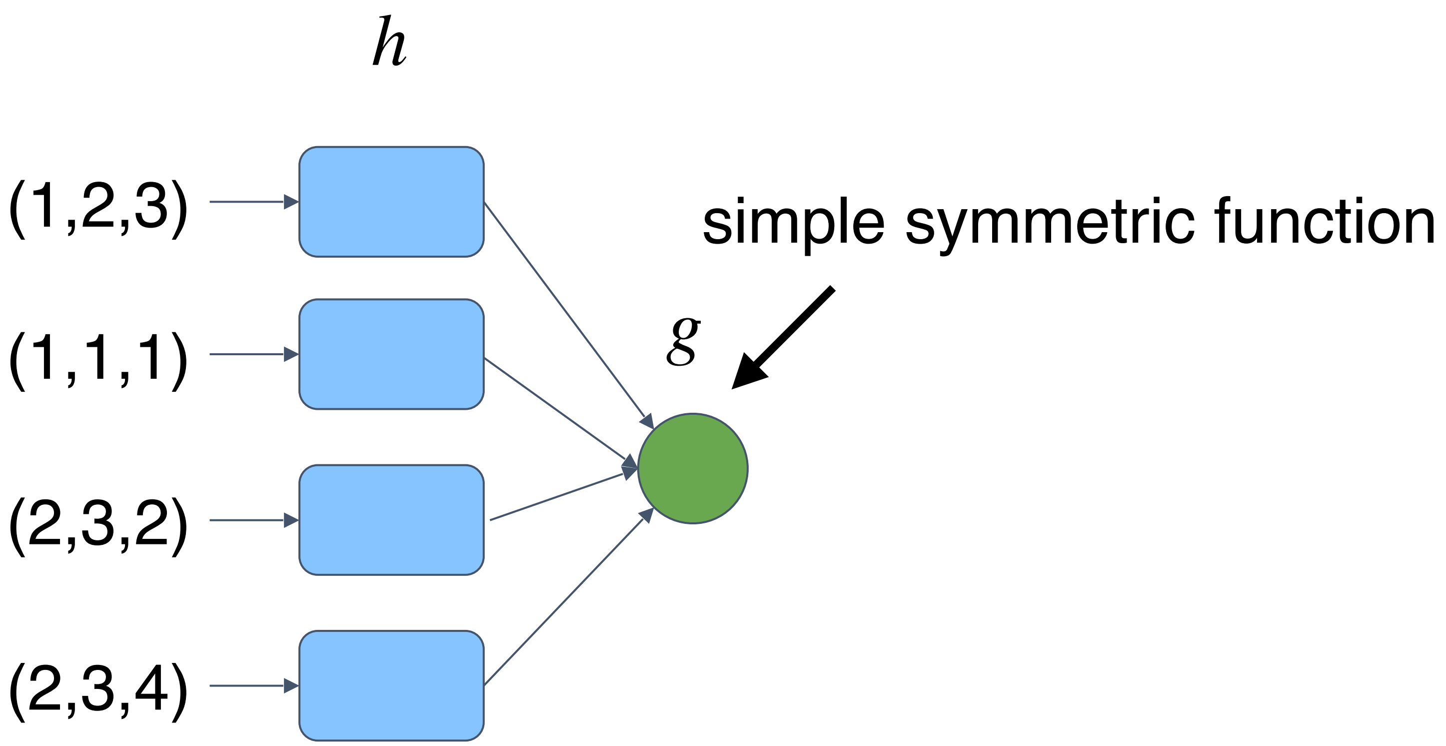
Construct symmetric function family

Observe: $f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



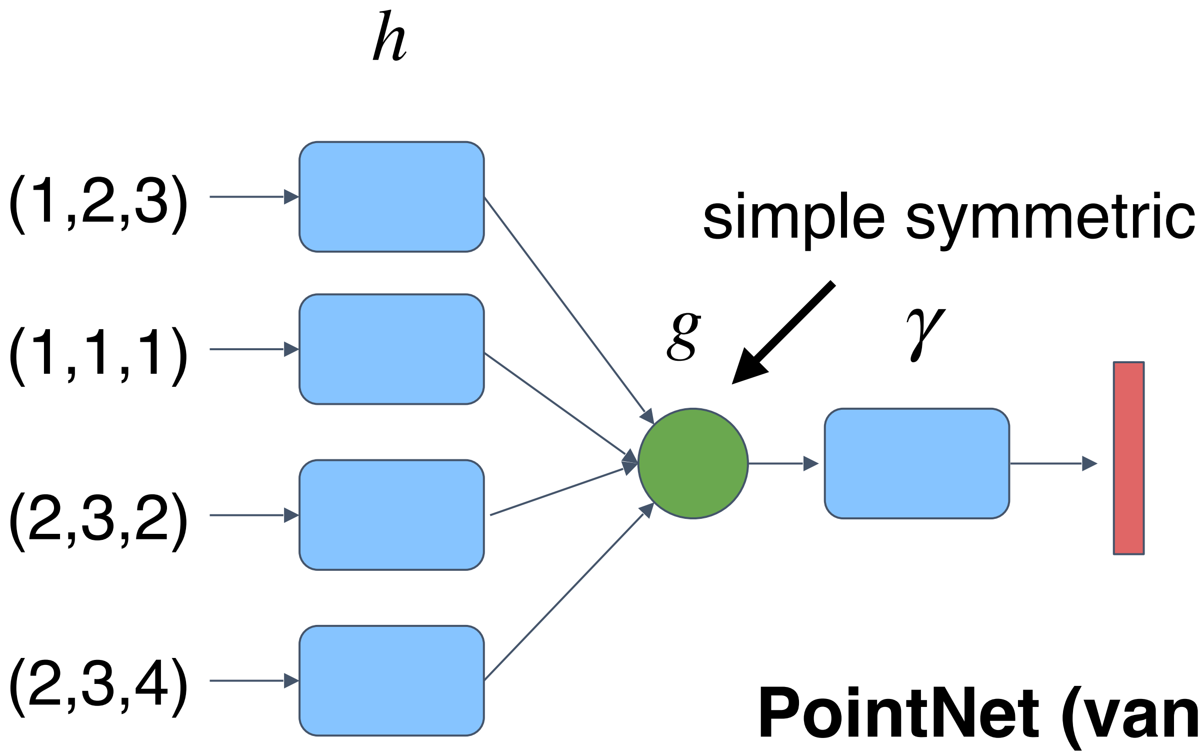
Construct symmetric function family

Observe: $f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



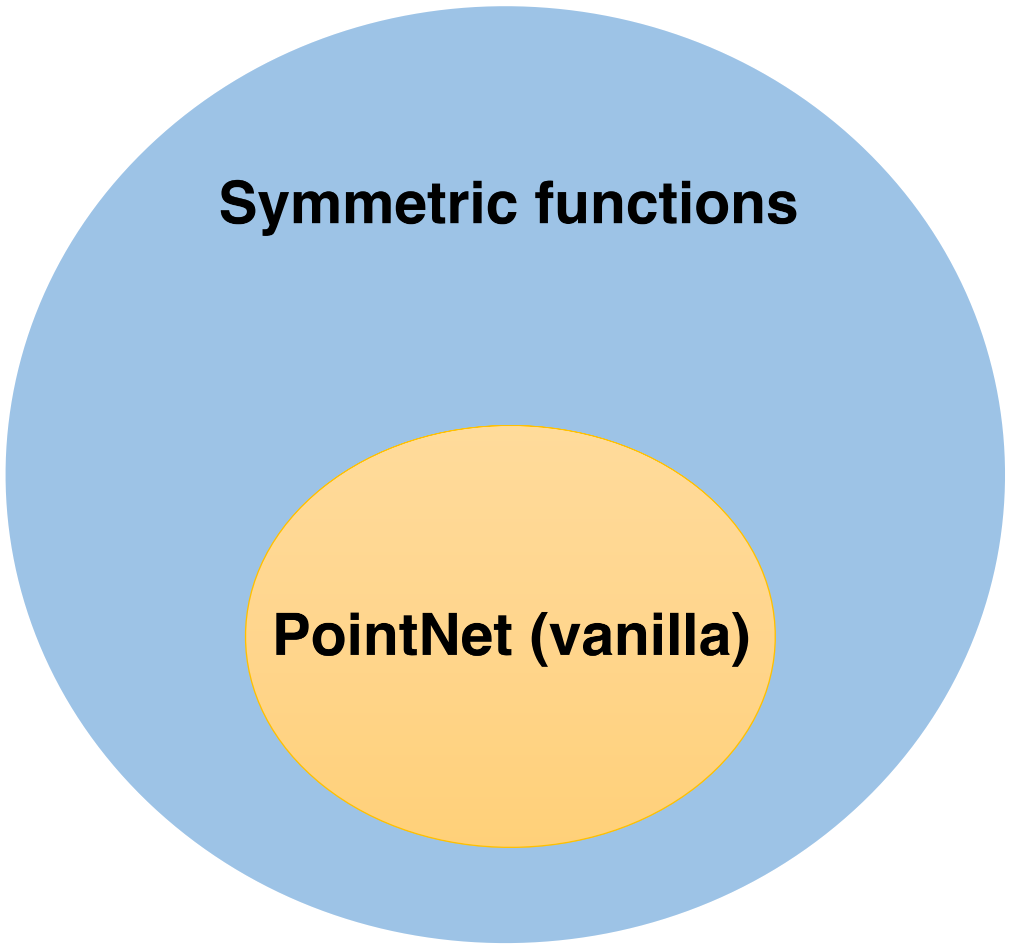
Construct symmetric function family

Observe: $f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet (vanilla)

Q: What symmetric functions can be constructed by PointNet?



A: Universal approximation to **continuous** symmetric functions

Theorem:

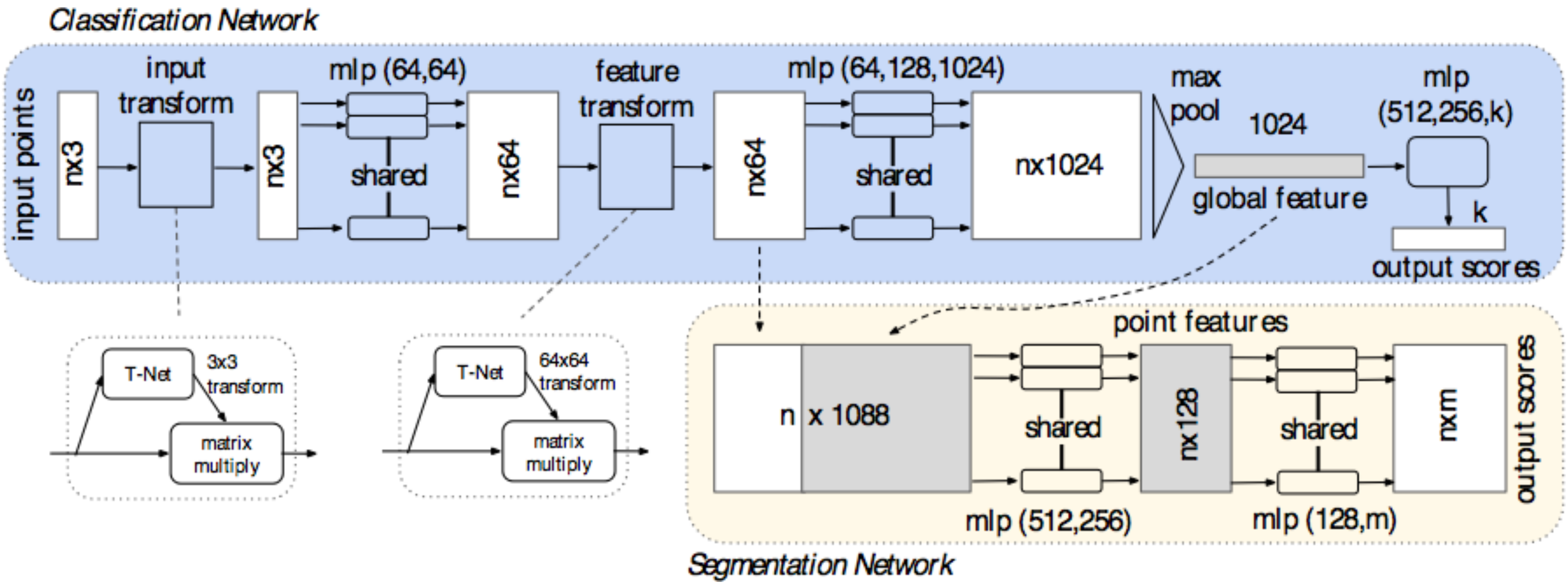
A Hausdorff continuous symmetric function $f: 2^{\mathcal{X}} \rightarrow \mathbb{R}$ can be arbitrarily approximated by PointNet.

$$\left| f(S) - \boxed{\gamma \left(\underset{x_i \in S}{\mathrm{MAX}} \{ h(x_i) \} \right)} \right| < \epsilon$$

$S \subseteq \mathbb{R}^d$,

PointNet (vanilla)

PointNet Architecture

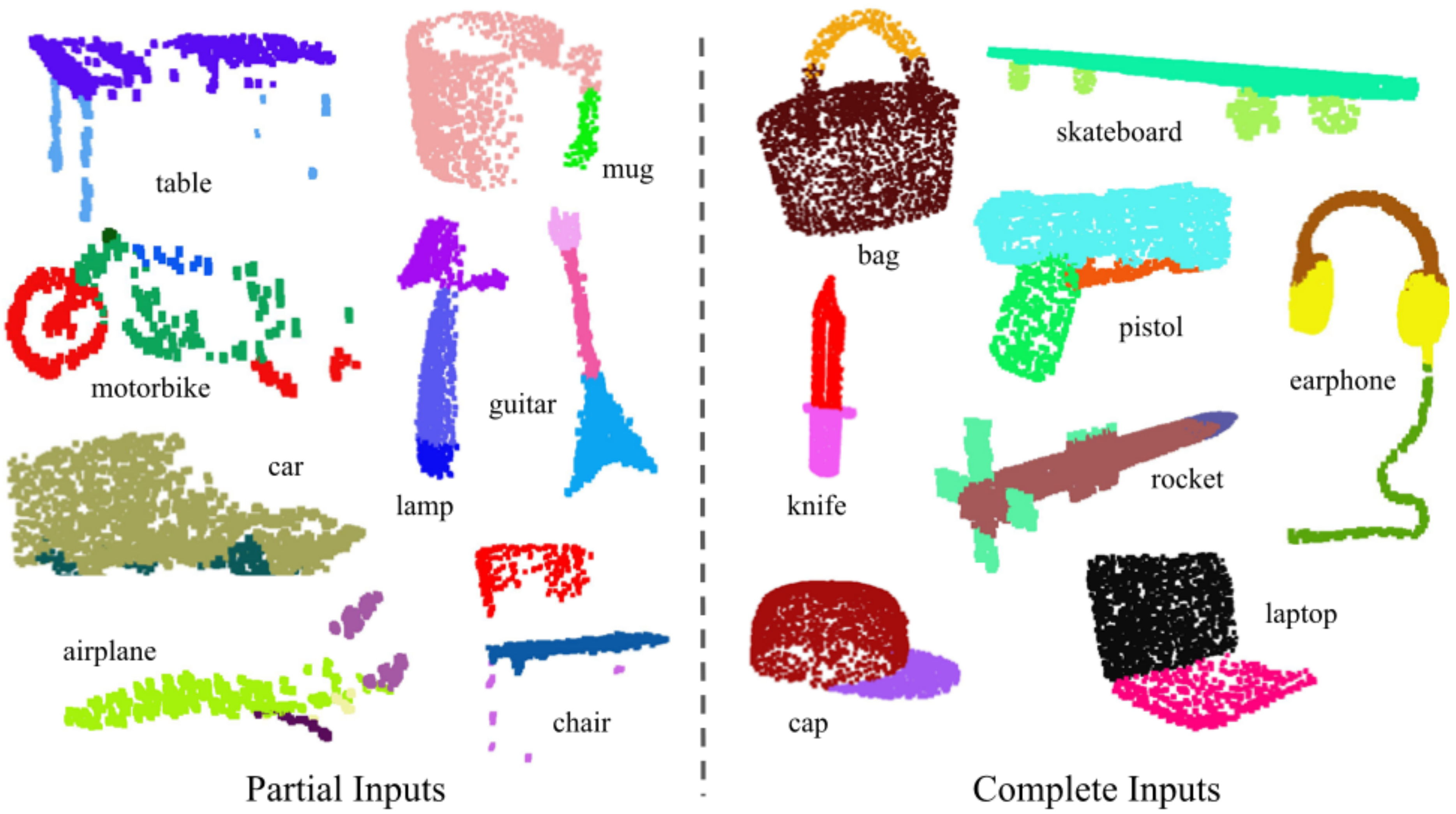


Results on Object Classification

Object Classification Accuracy on ModelNet40

	input	#views	accuracy avg. class	accuracy overall
SPH [12]	mesh	-	68.2	-
3DShapeNets [29]	volume	1	77.3	84.7
VoxNet [18]	volume	12	83.0	85.9
Subvolume [19]	volume	20	86.0	**89.2**
LFD [29]	image	10	75.5	-
MVCNN [24]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

Results on Object Part Segmentation



Results on Object Part Segmentation

Part Segmentation mIoU on ShapeNet Part Dataset

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [28]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [30]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**	91.5	**85.9**	80.8	95.3	65.2	**93.0**	81.2	**57.9**	**72.8**	**80.6**

Results on Semantic Scene Segmentation

Results on Semantic Scene Parsing

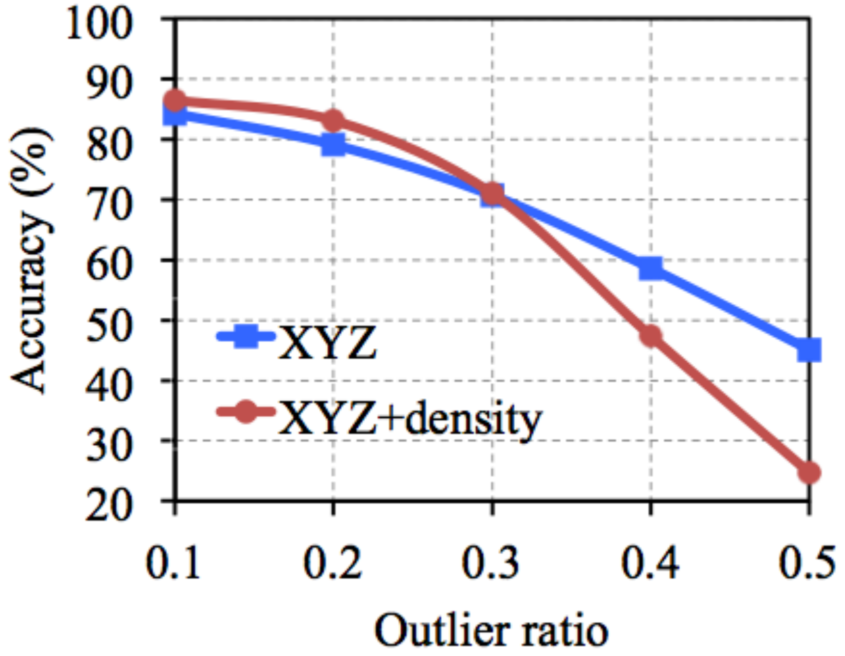
Semantic Segmentation (point based)
on Stanford Semantic Parsing dataset

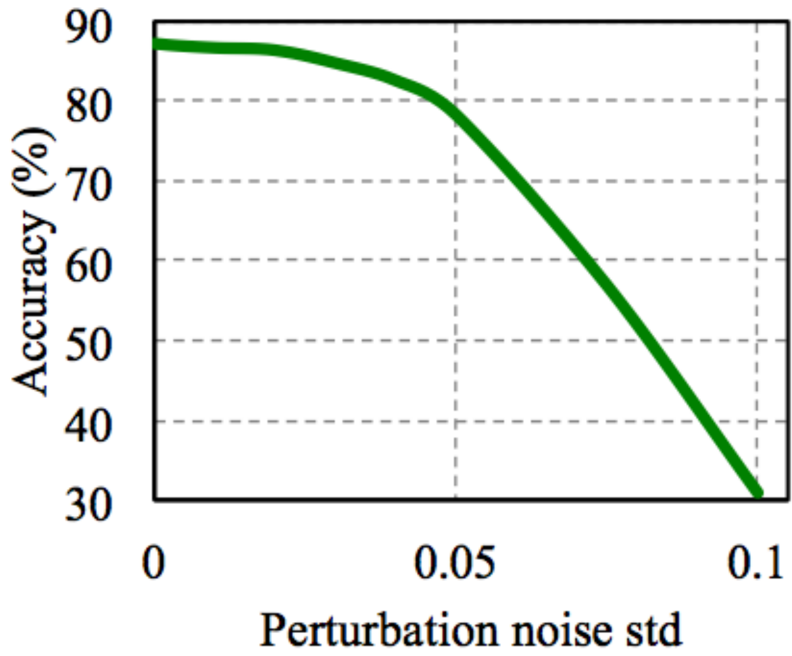
	mean IoU	overall accuracy
Ours baseline	20.12	53.19
Ours PointNet	**47.71**	**78.62**

3D Object Detection (bounding box based)

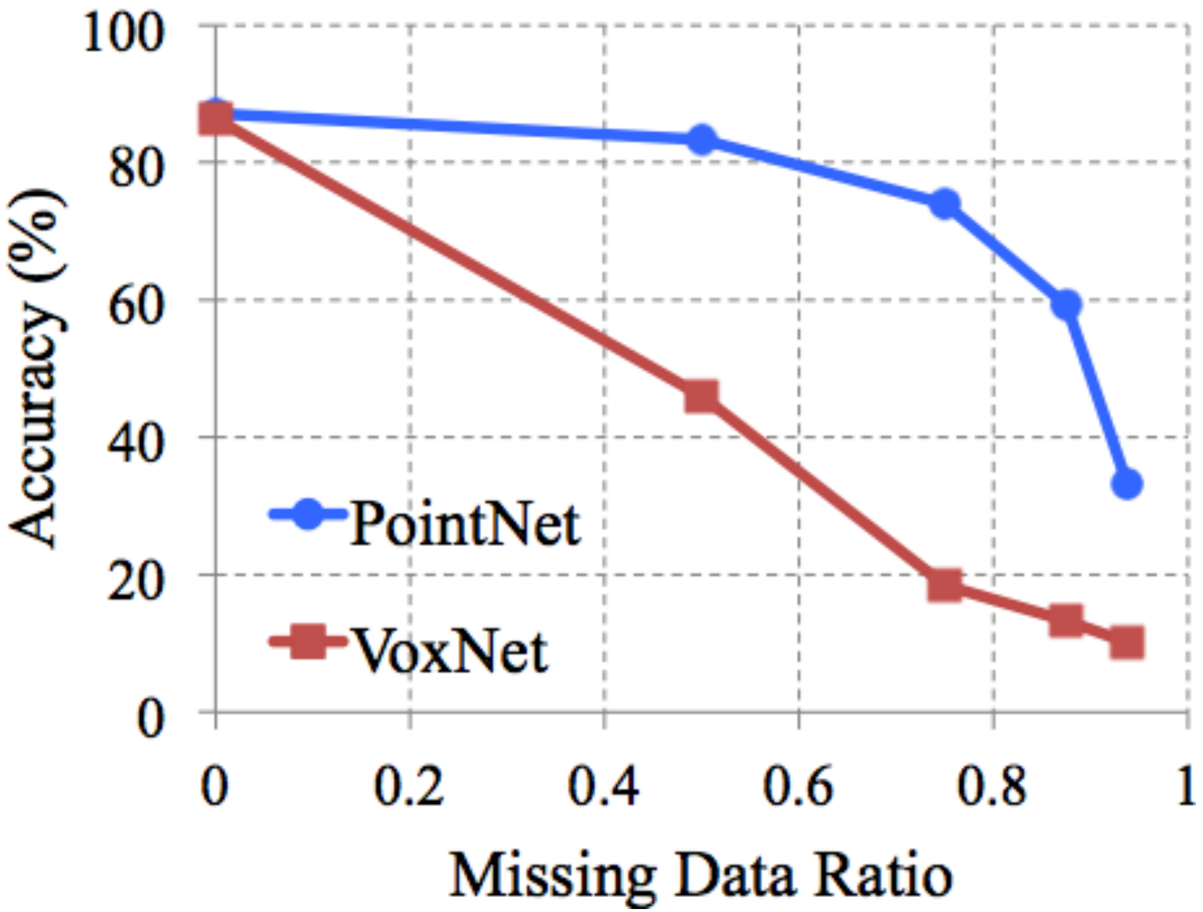
	table	chair	sofa	board	mean
# instance	455	1363	55	137	
Armeni et al. [2]	46.02	16.15	**6.78**	3.91	18.22
Ours	**46.67**	**33.80**	4.76	**11.72**	**24.24**

Robustness to Data Corruption

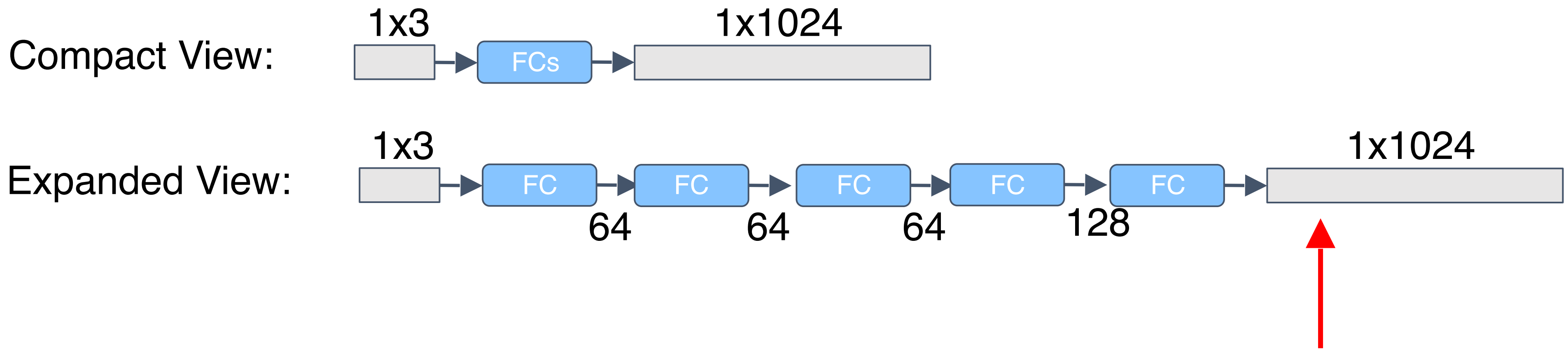




Robustness to Data Corruption



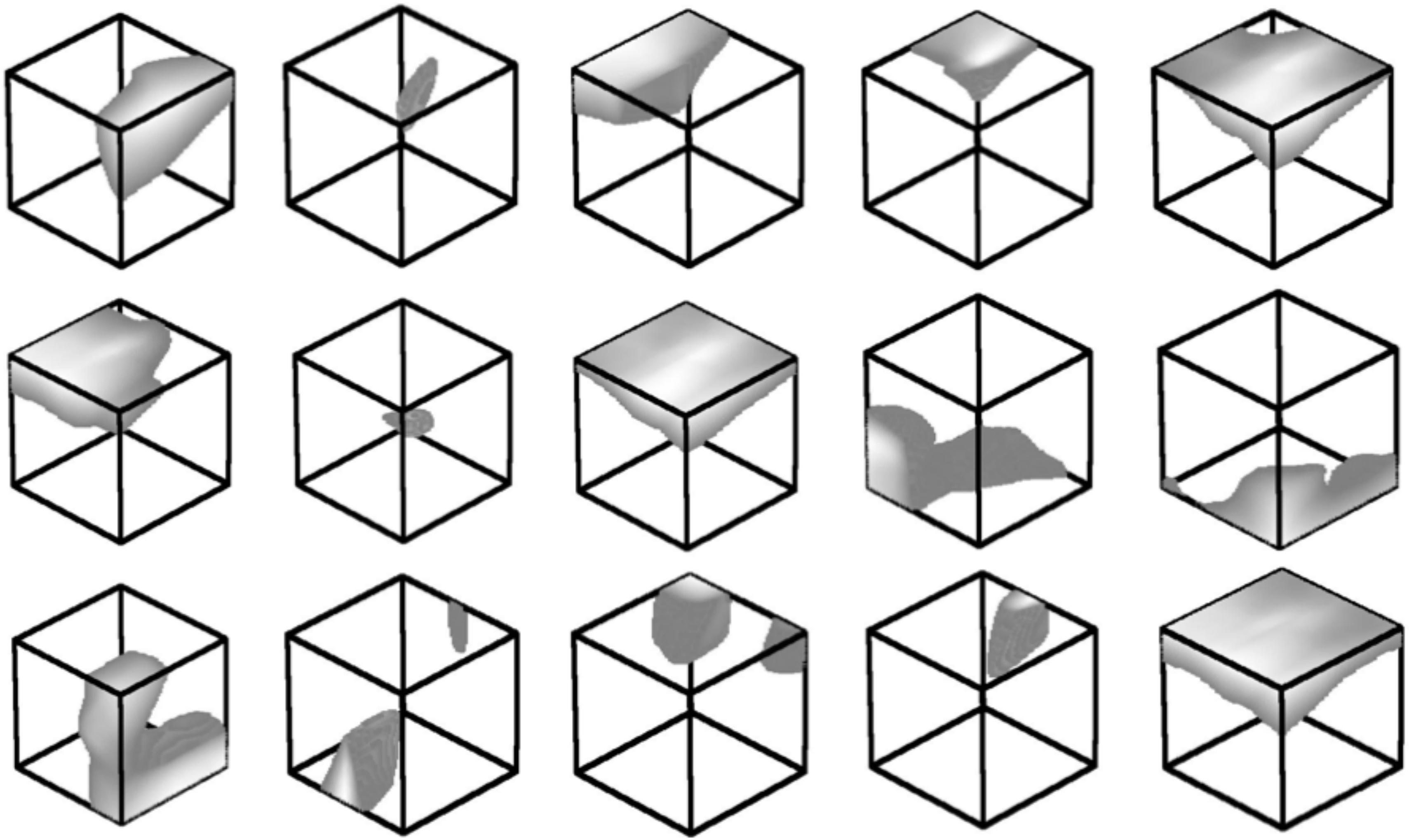
Visualizing Point Functions



Which input point will activate neuron j?

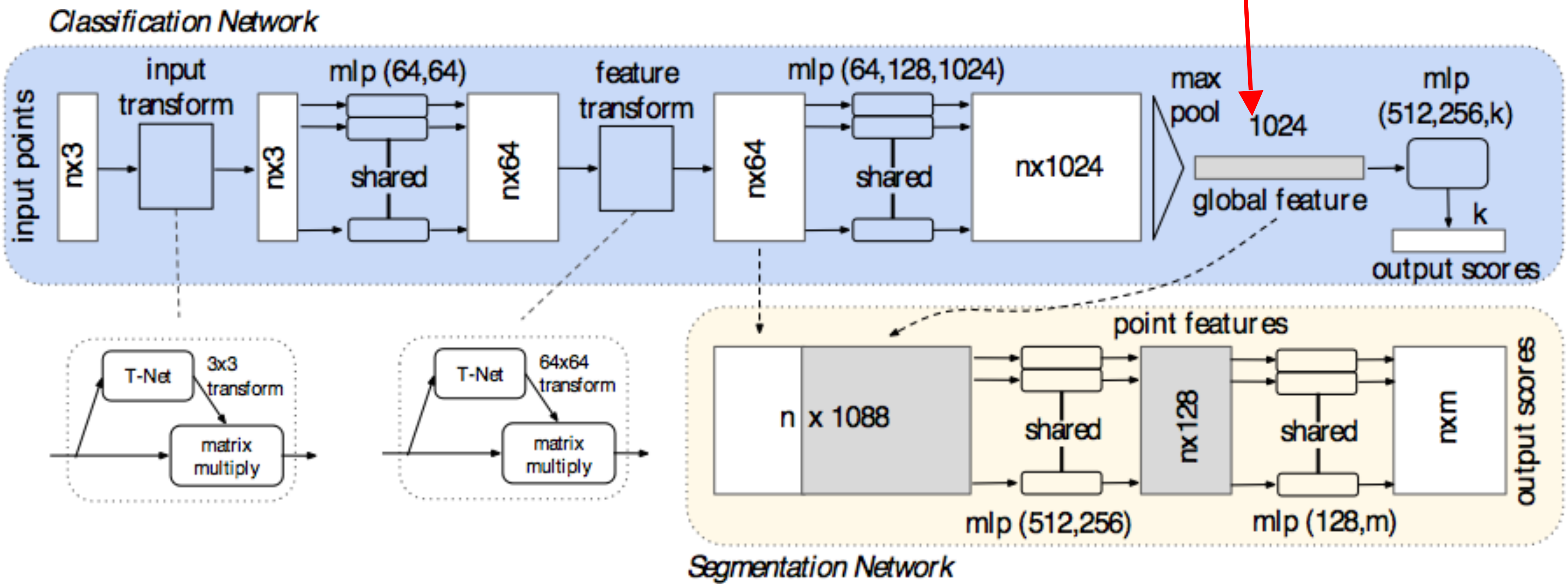
Find the top-K points in a dense volumetric grid that activates neuron j.

Visualizing Point Functions

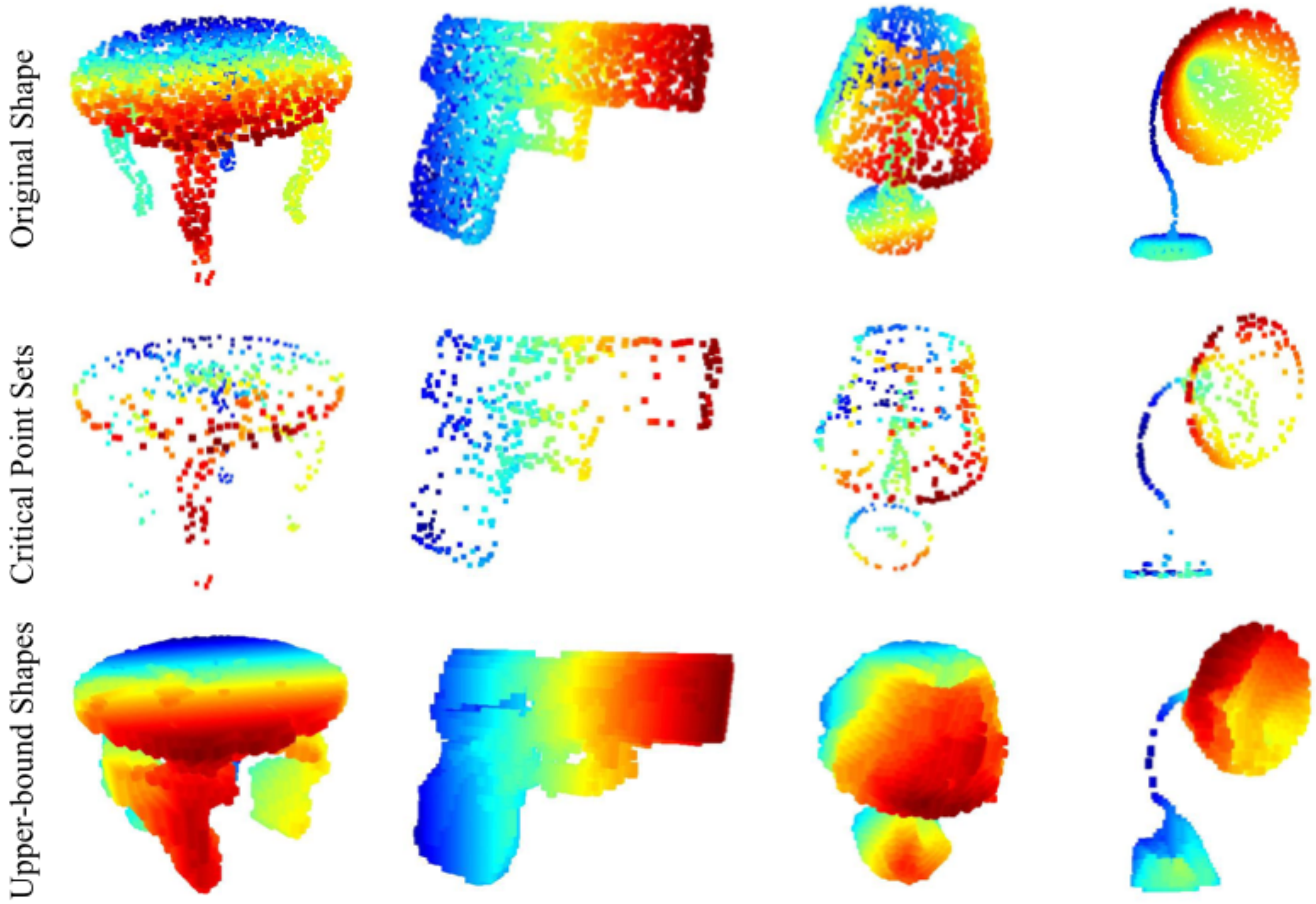


Visualizing Global Point Cloud Features

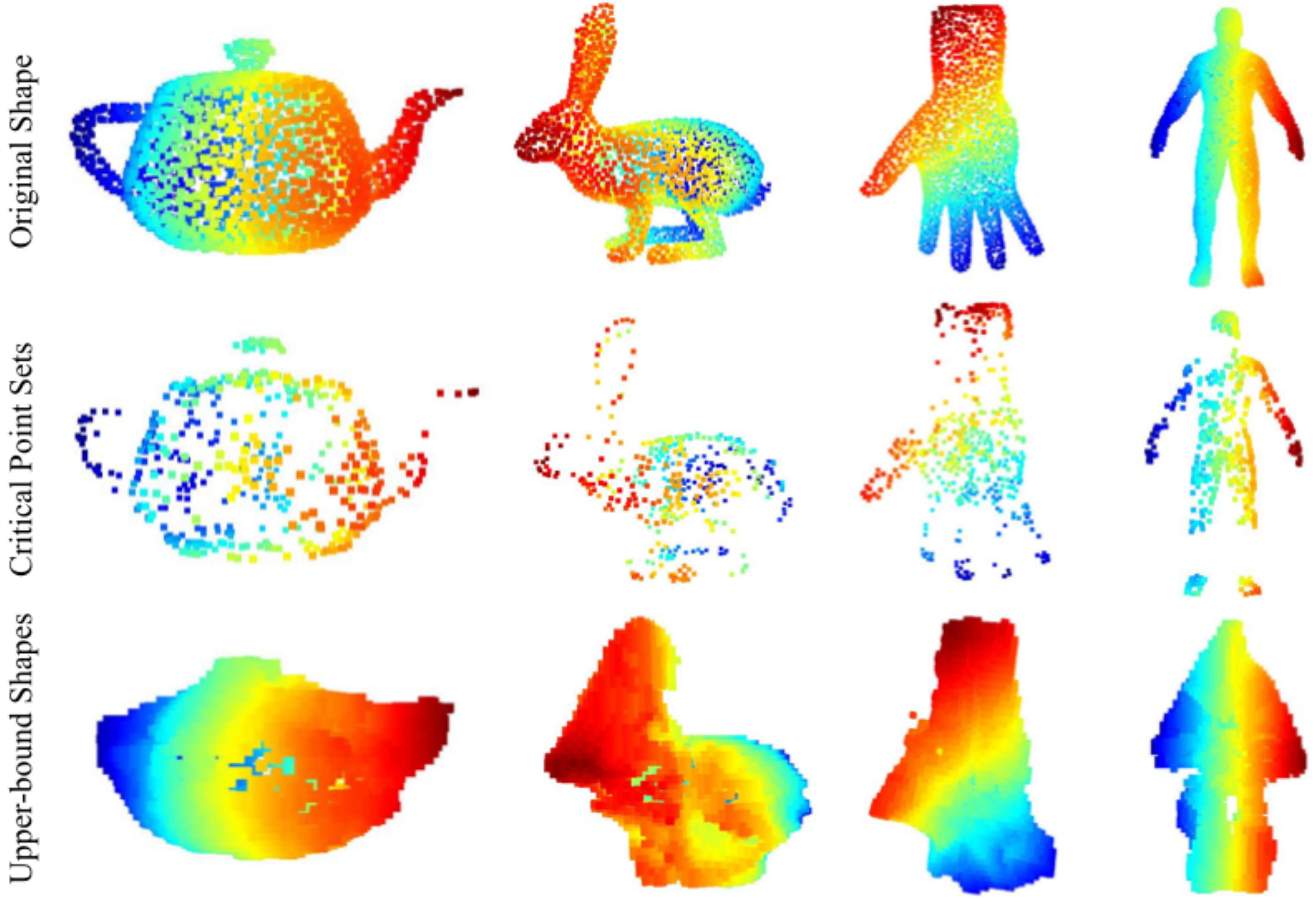
What's captured and left out here?



Visualizing Global Point Cloud Features

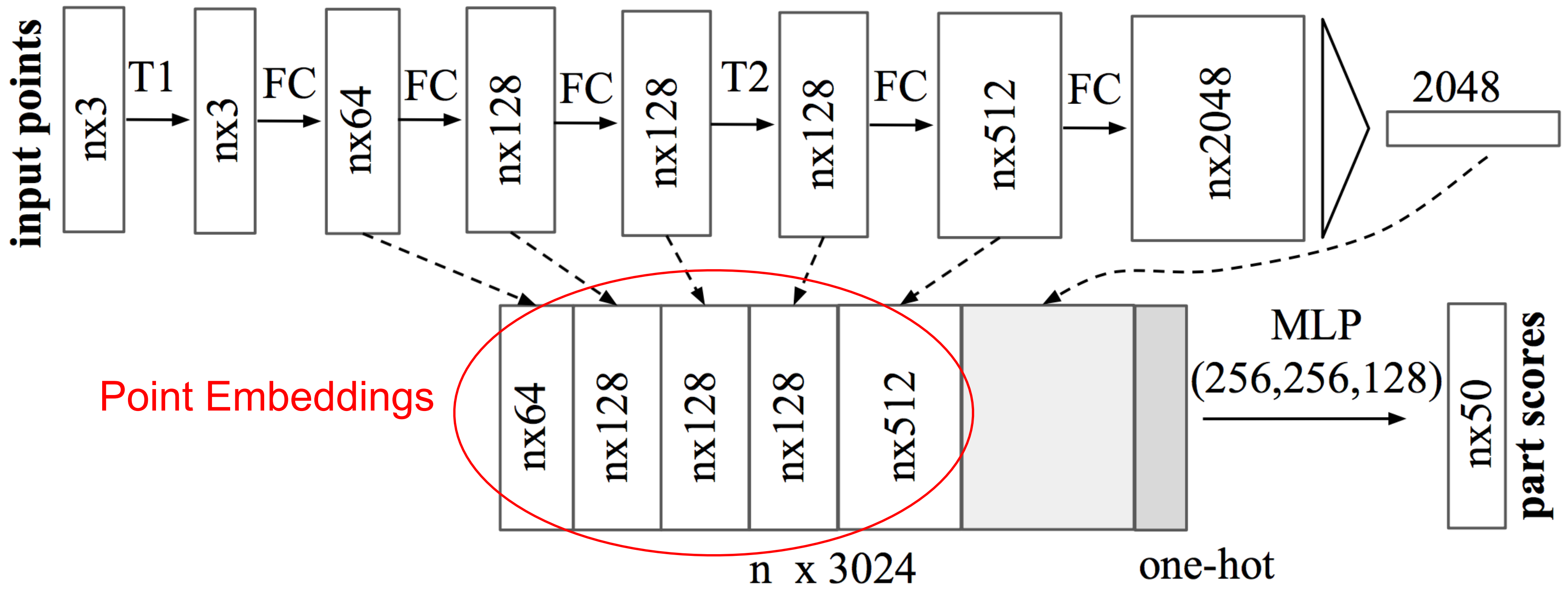


Visualizing Global Point Cloud Features (OOS)



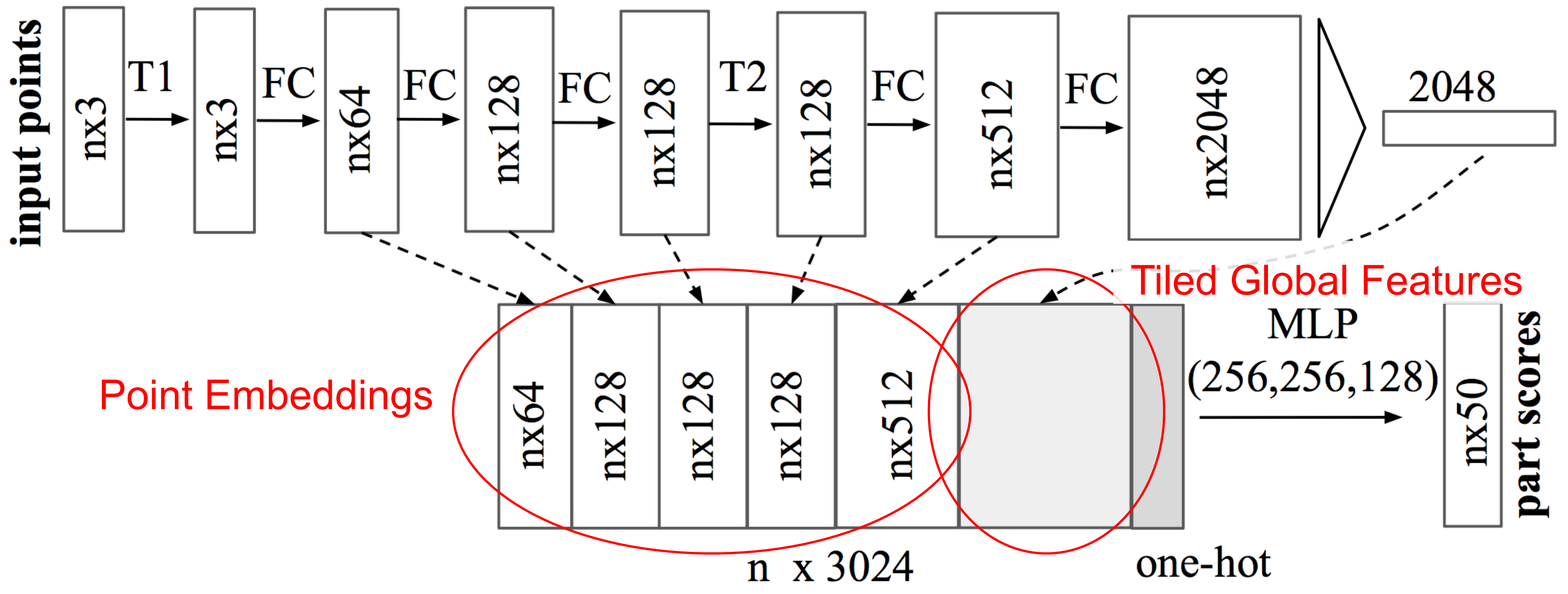
Previous Work: PointNet v1.0

Segmentation Network



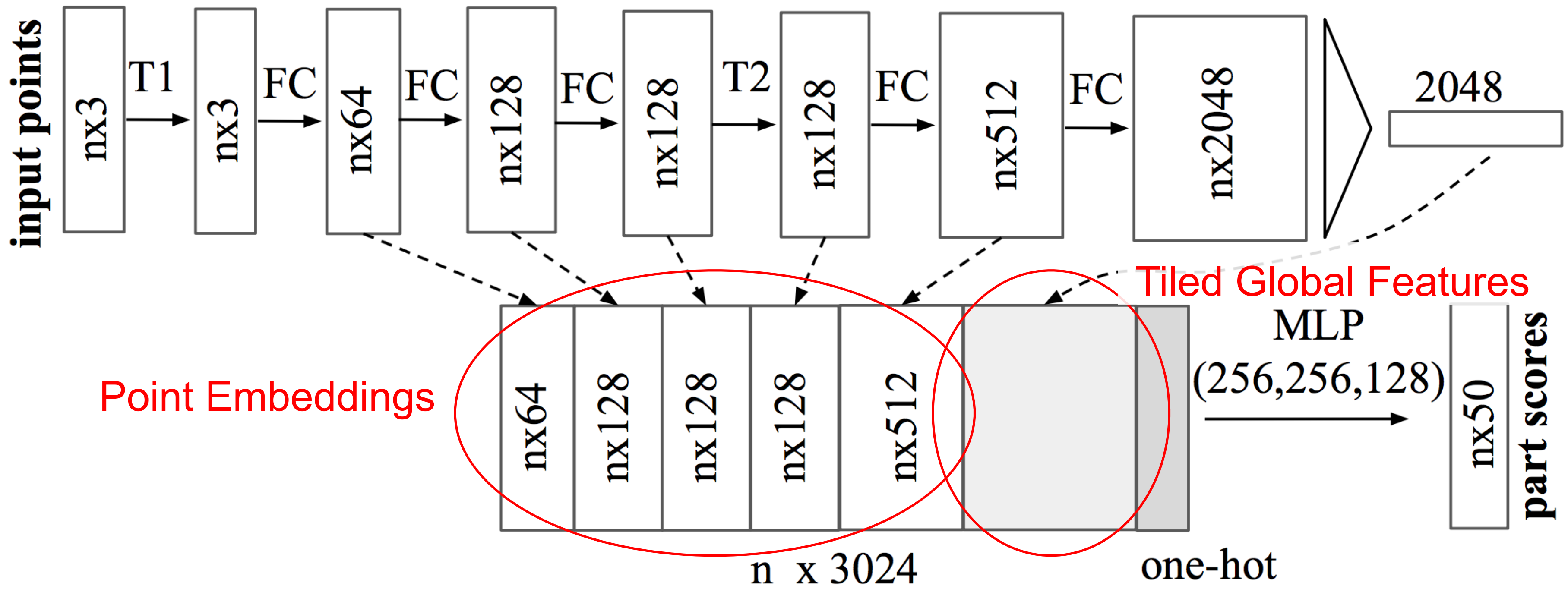
Previous Work: PointNet v1.0

Segmentation Network



Previous Work: PointNet v1.0

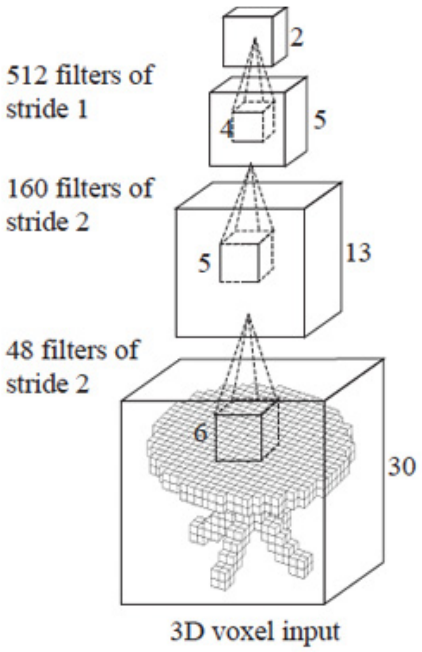
Segmentation Network



- No local context for each point!

Limitations of PointNet v1.0

- Hierarchical Feature Learning
- Increasing receptive field

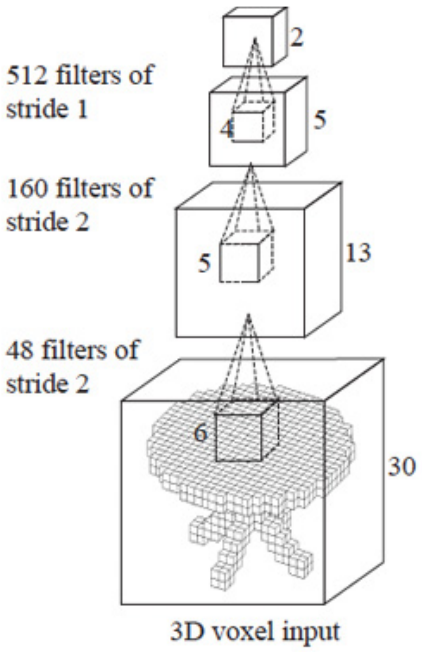


3D CNN (Wu et al.)

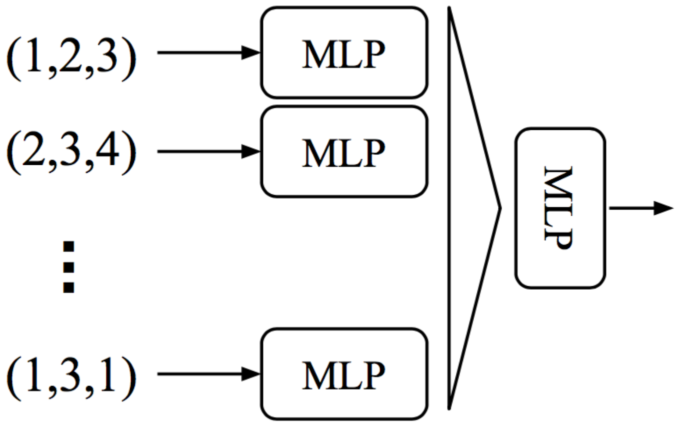
Limitations of PointNet v1.0

- Hierarchical Feature Learning
- Increasing receptive field

Global Feature Learning
Receptive field:
one point OR all points



v.s.

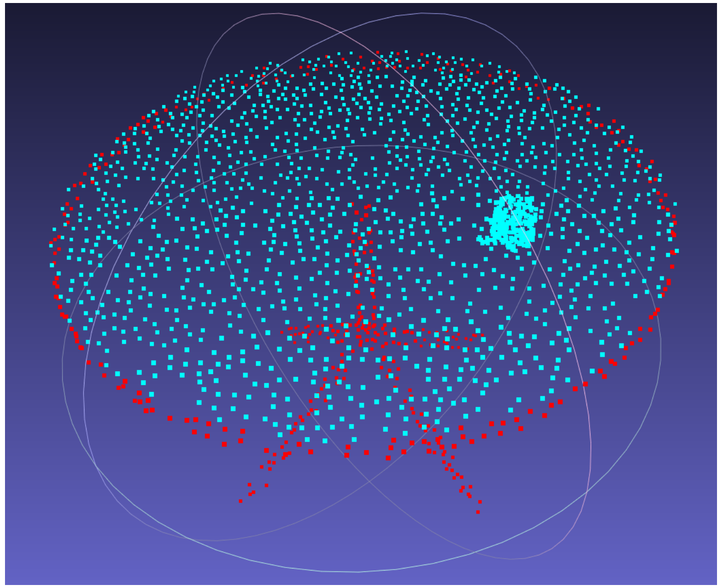


3D CNN (Wu et al.)

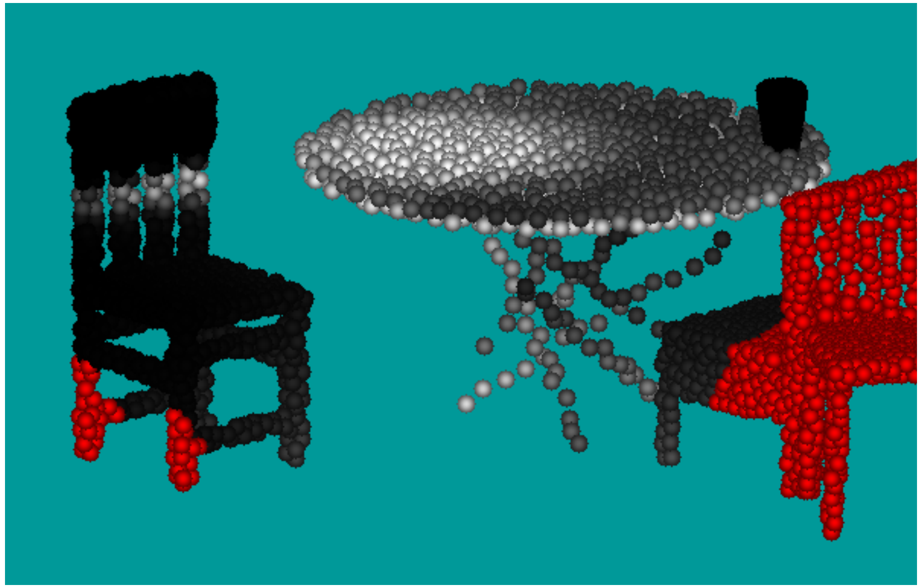
PointNet (vanilla) (Qi et al.)

Limitations of PointNet v1.0

Artifacts in segmentation tasks:



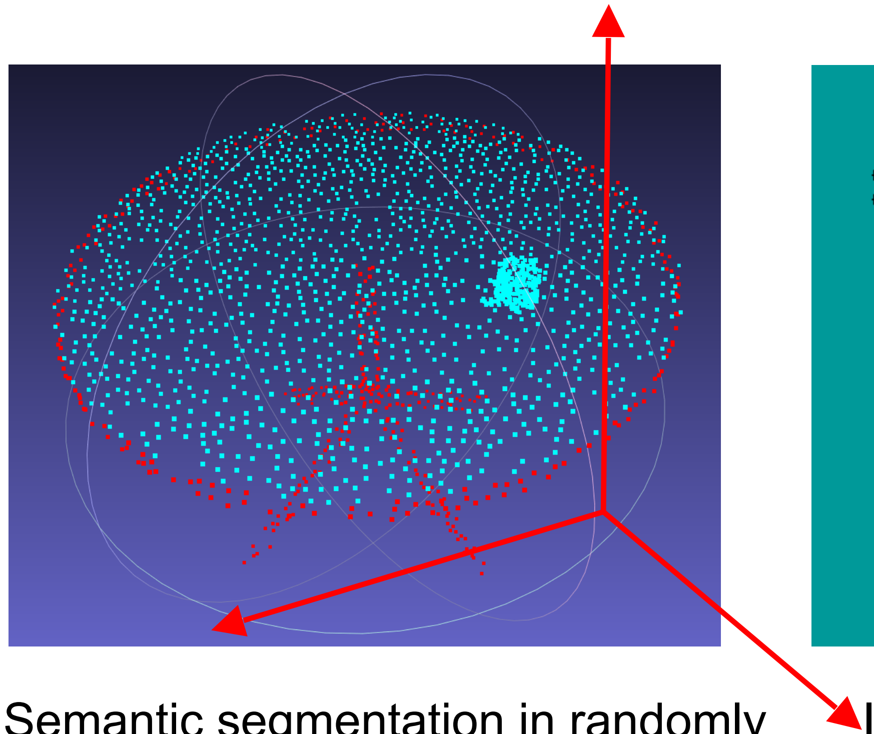
Semantic segmentation in randomly translated table-cup scene.



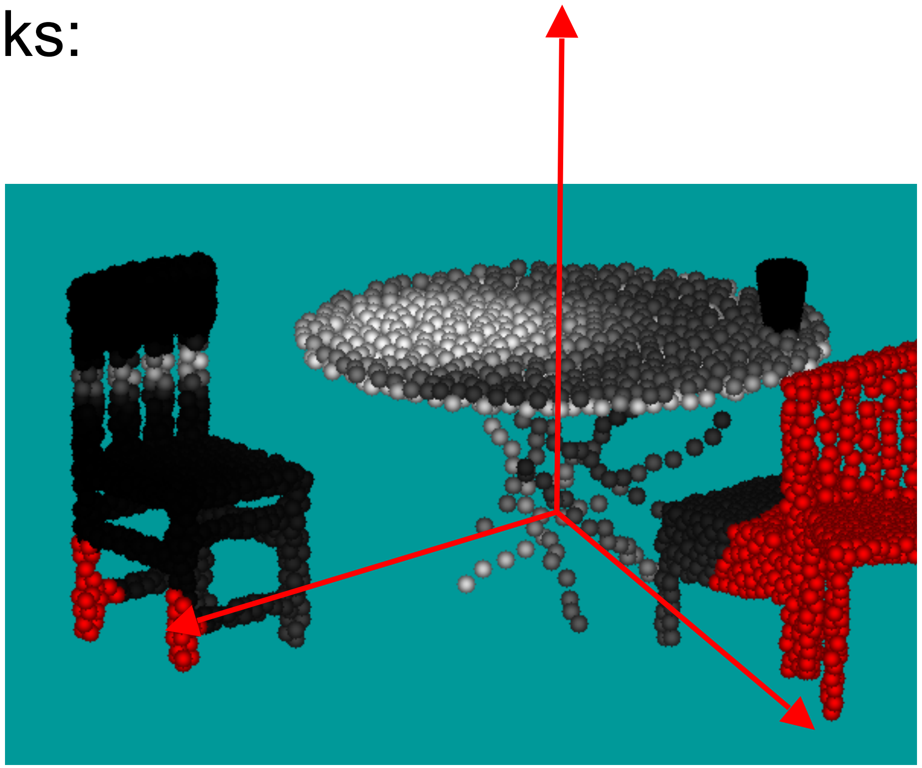
Instance segmentation in table-chair-cup scene

Limitations of PointNet v1.0

Artifacts in segmentation tasks:



Semantic segmentation in randomly translated table-cup scene.



Instance segmentation in table-chair-cup scene

- Global feature depends on absolute XYZ!
- Hard to generalize to unseen point configurations

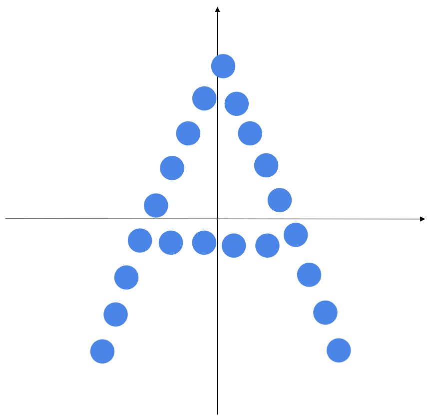
Question

- How to learn local context feature for points?

- Use PointNet in local regions, aggregate local region features by PointNet again..
-
- Hierarchical feature learning!

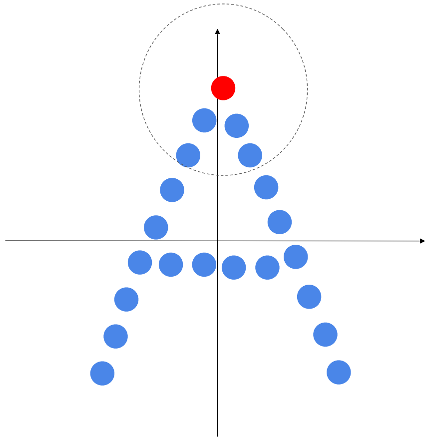
Multi-Scale PointNet for Hierarchical Feature Learning

PointNet v2.0: Multi-Scale PointNet

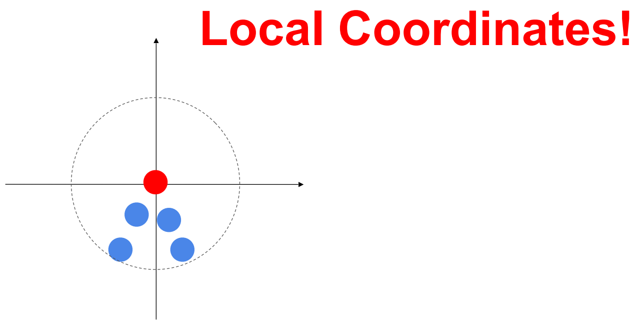


N points in (x,y)

PointNet v2.0: Multi-Scale PointNet

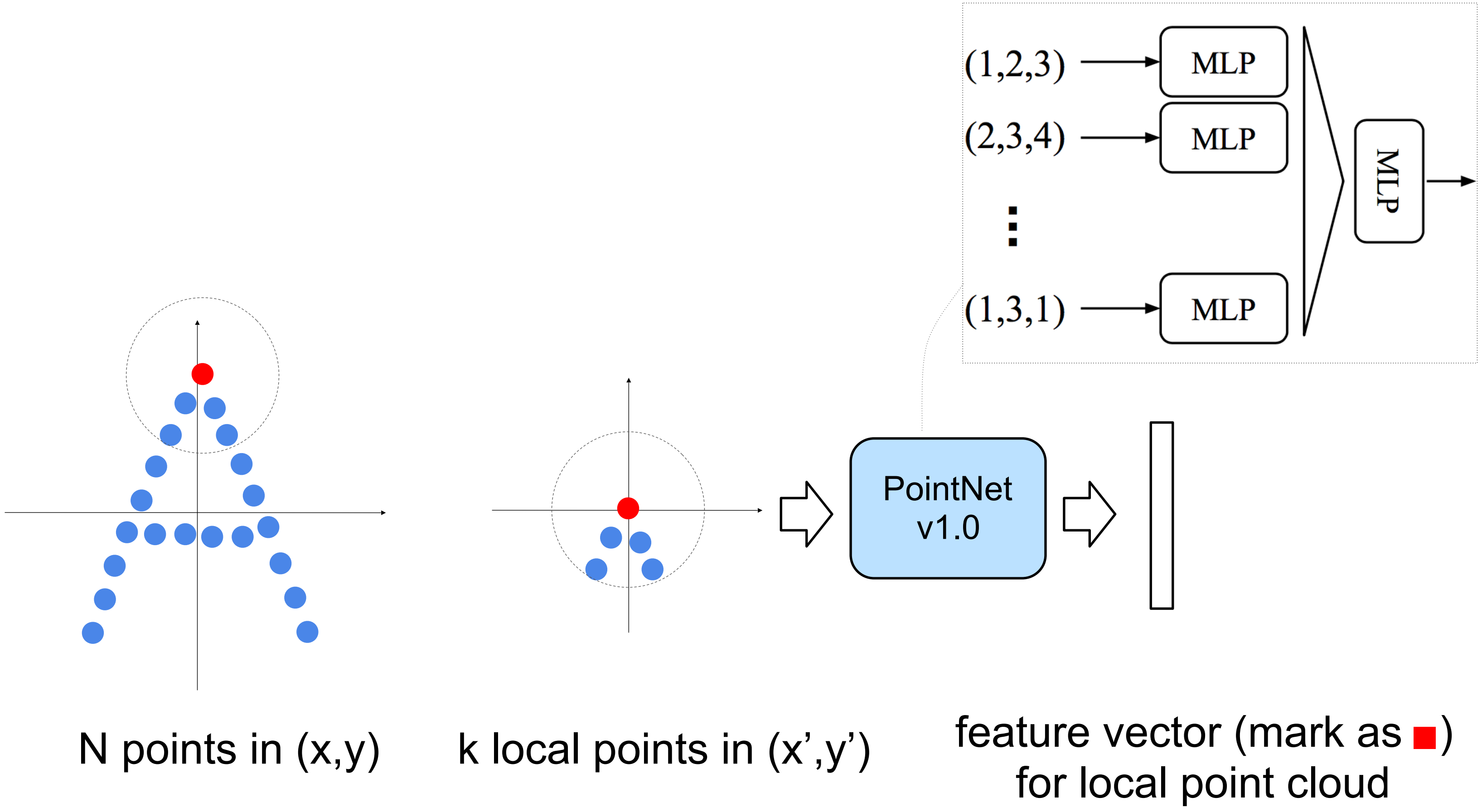


N points in (x,y)



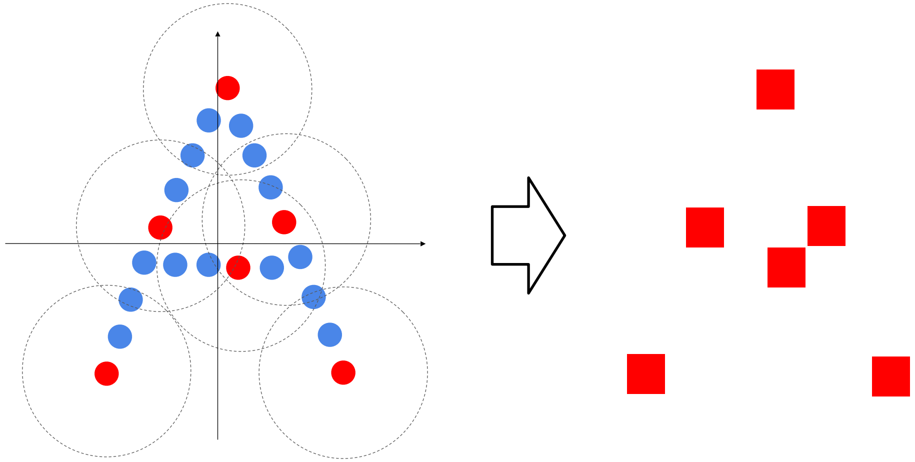
k local points in (x',y')

PointNet v2.0: Multi-Scale PointNet



PointNet v2.0: Multi-Scale PointNet

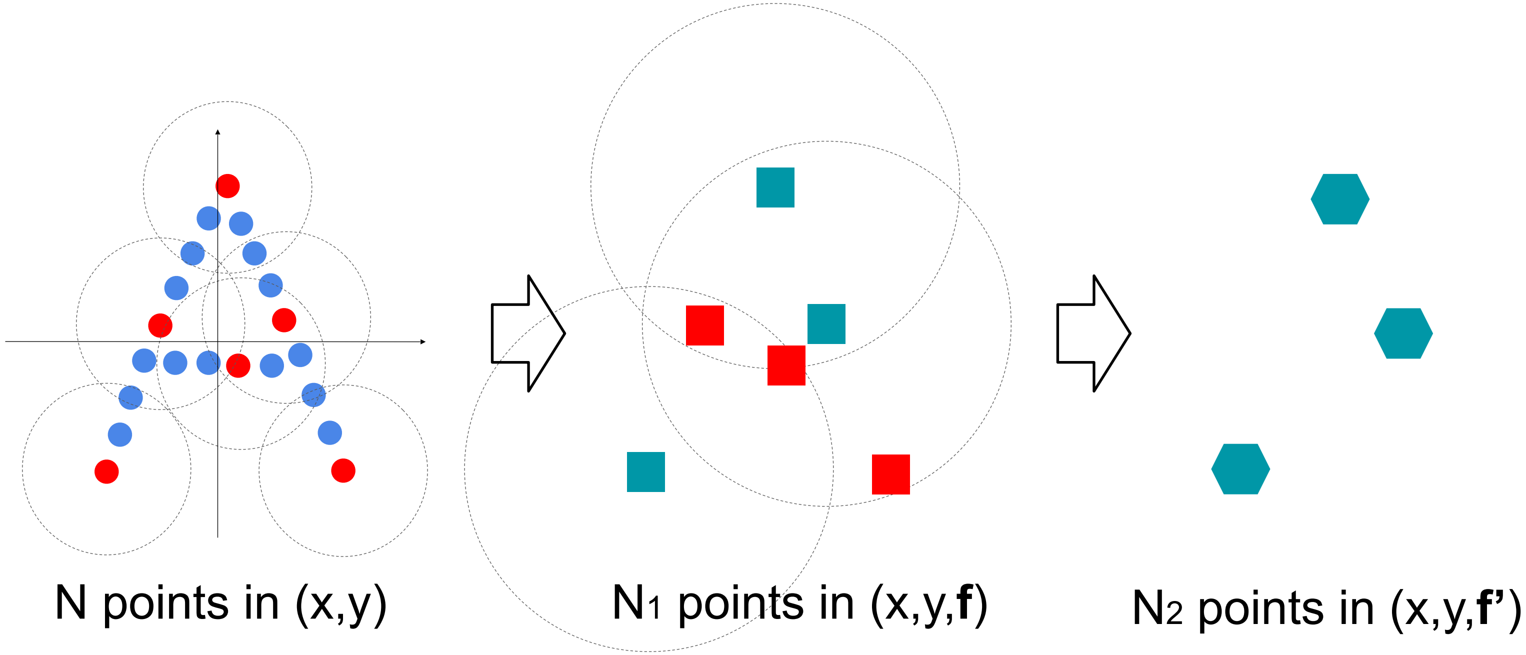
PointNet Module/Layer: Farthest Point Sampling + Grouping + PointNet v1.0



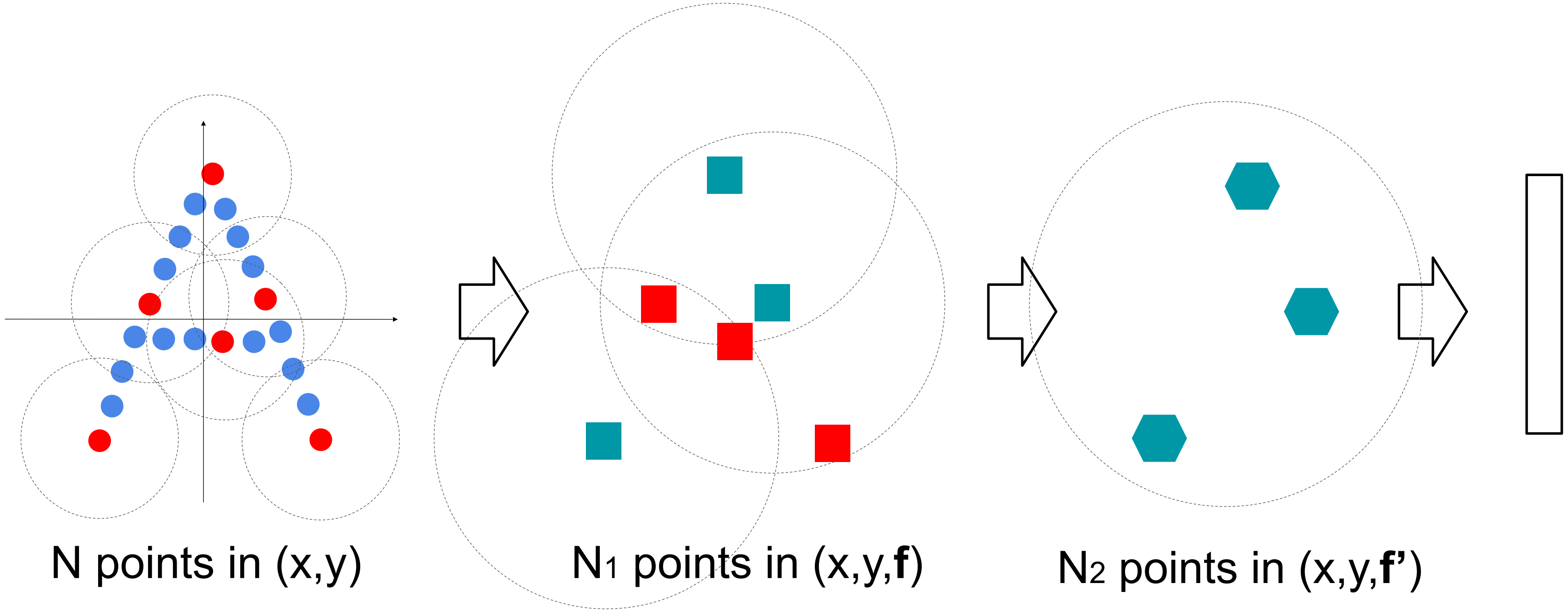
N points in (x,y)

N_1 points in (x,y,**f**)

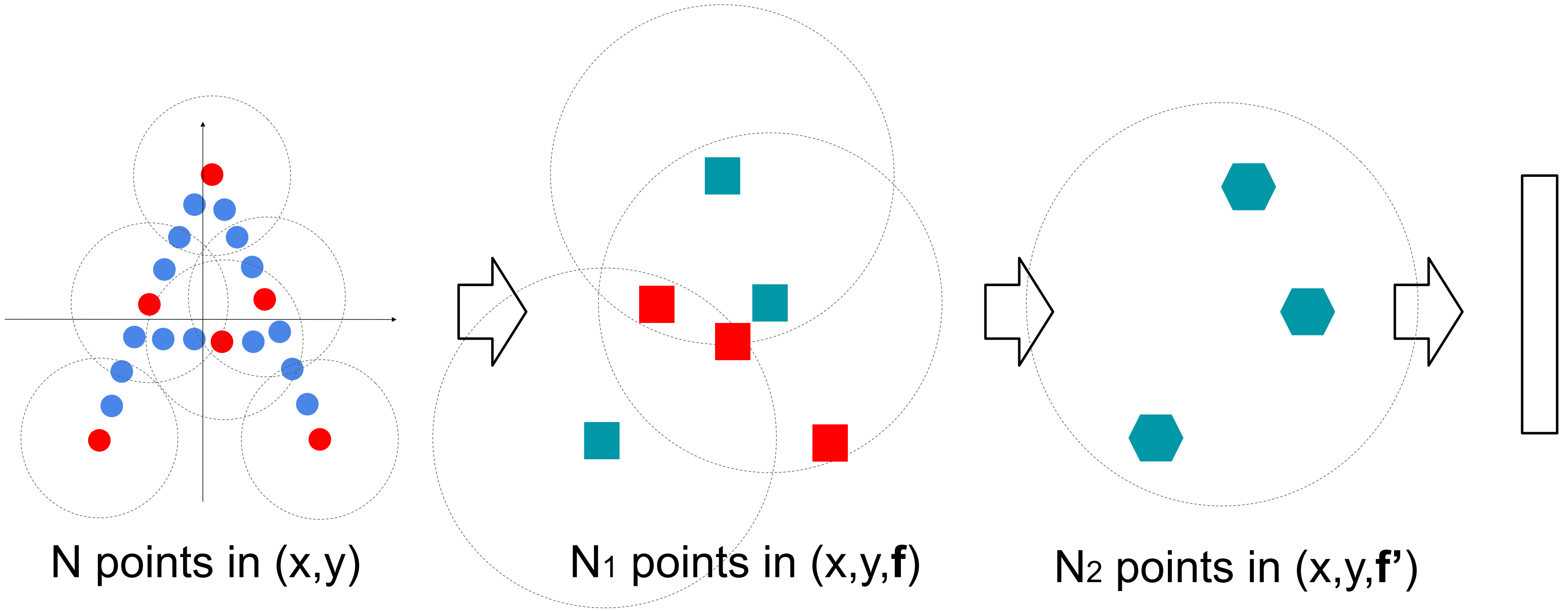
PointNet v2.0: Multi-Scale PointNet



PointNet v2.0: Multi-Scale PointNet



PointNet v2.0: Multi-Scale PointNet

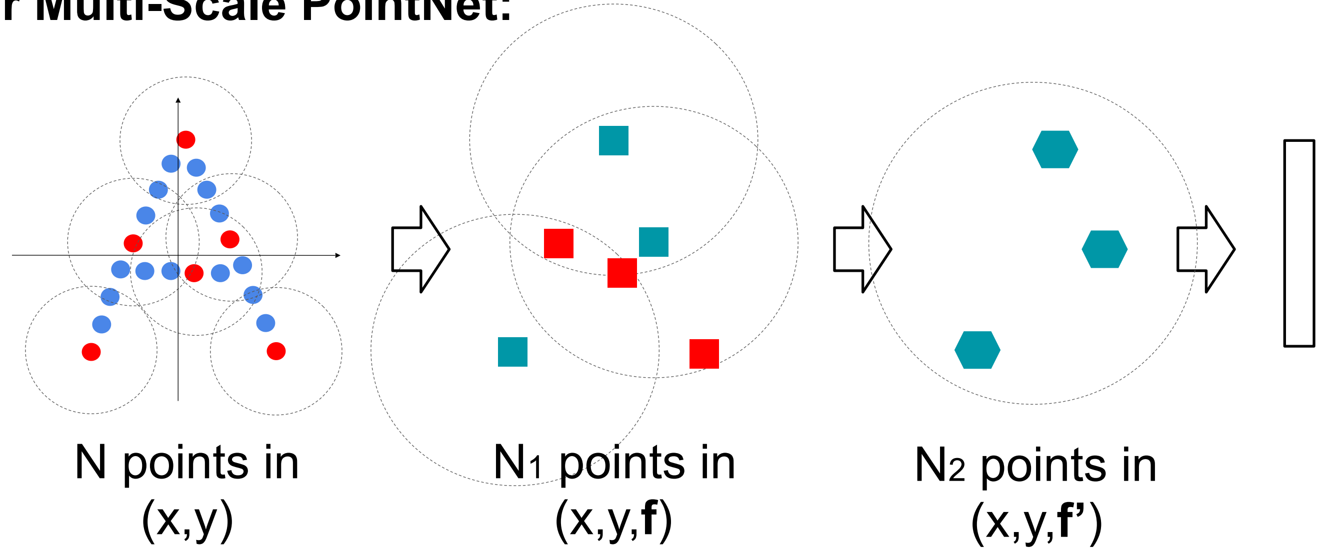


1. Larger receptive field in higher layers ✓
2. Less points in higher layers (more scalable) ✓
3. Weight sharing ✓
4. Translation invariance (local coordinates in local regions) ✓

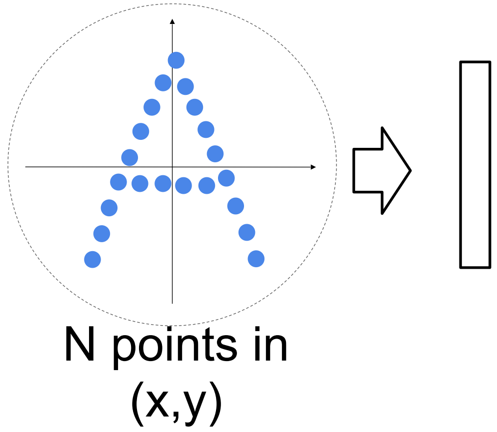
Discussions on Multi-Scale PointNet

Multi-Scale PointNet v.s. PointNet v1.0

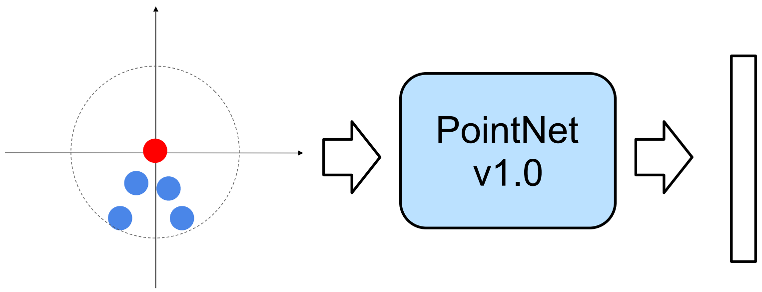
Three-layer Multi-Scale PointNet:

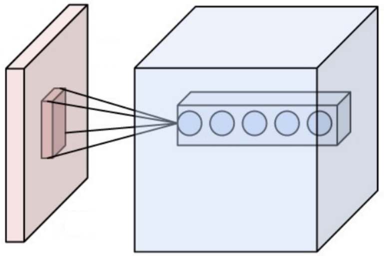


One-layer Multi-Scale PointNet <=> PointNet v1.0



PointNet Layer v.s. Convolution Layer

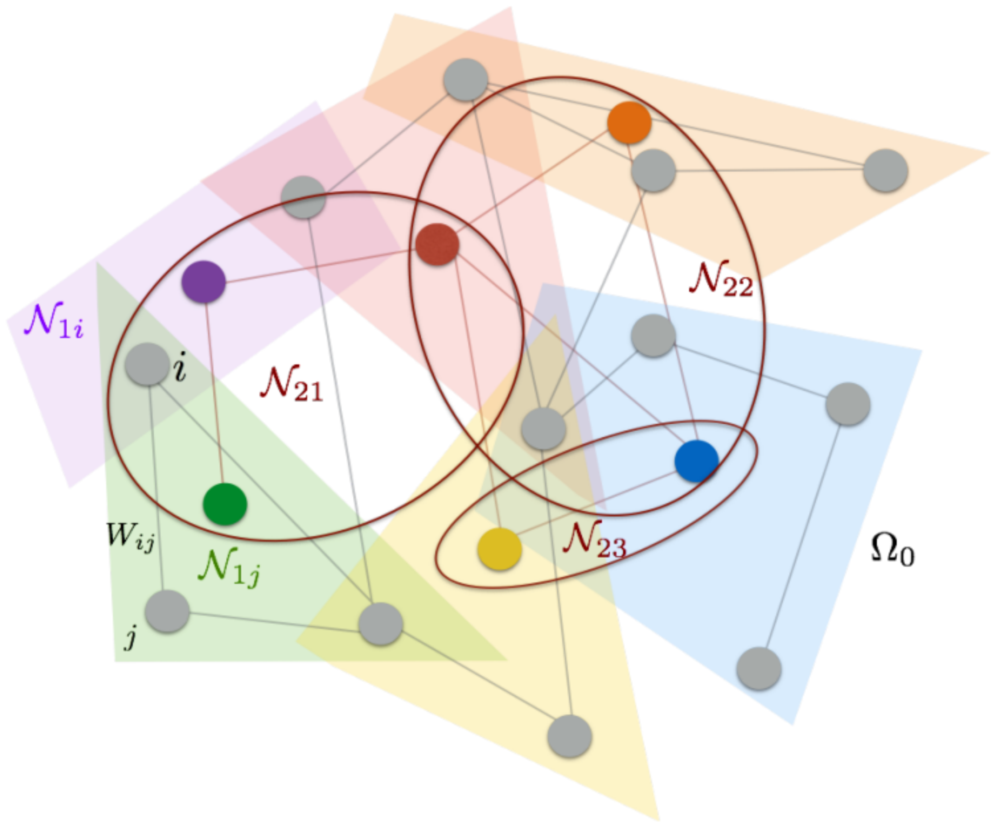




	PointNet Layer	Convolution Layer
Input:	Point set	Dense array
Operation:	MLP + max pooling	Multiply and add
Neighbor -hood:	Distance query	Array index

Multi-Scale PointNet v.s. Graph CNN

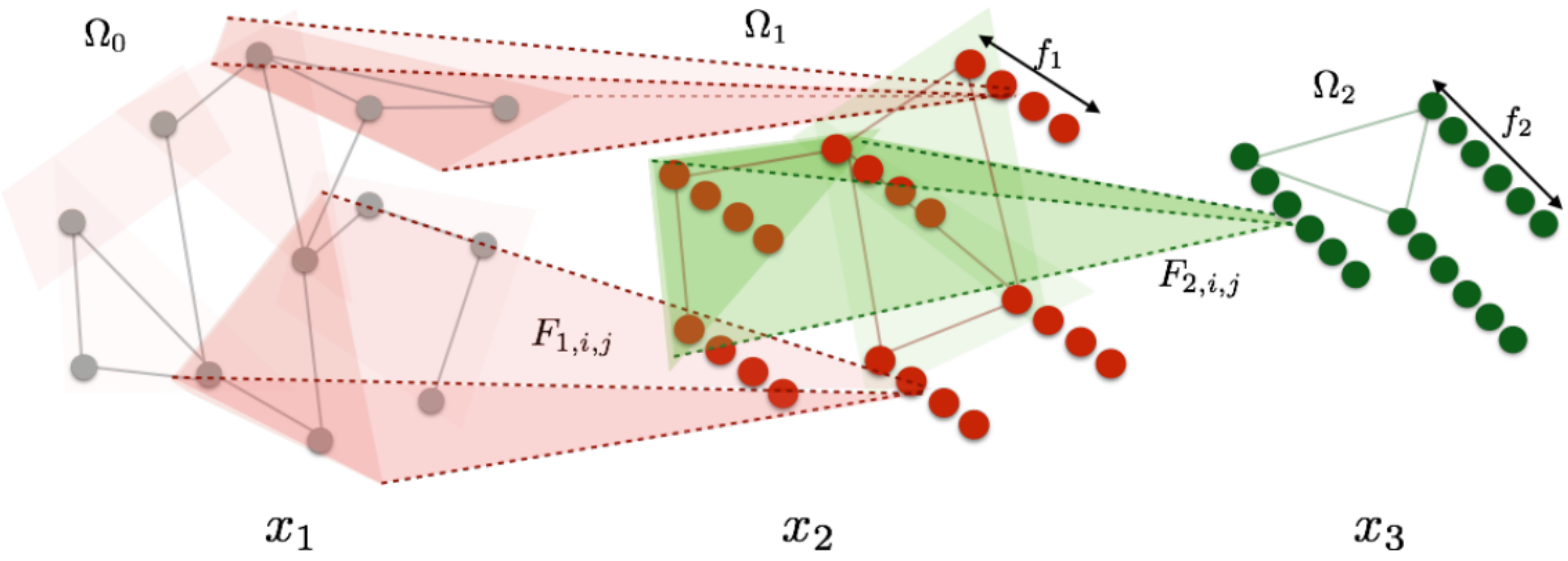
- Unexpectedly strong relation with Graph CNN:



Joan Bruna et al. Spectral Networks and Deep Locally Connected Networks on Graphs. ICLR 2014

Multi-Scale PointNet v.s. Graph CNN

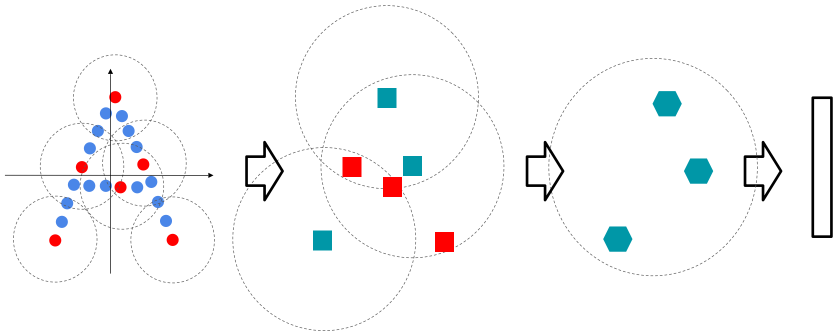
- Local feature extraction, graph coarsening, repeat..



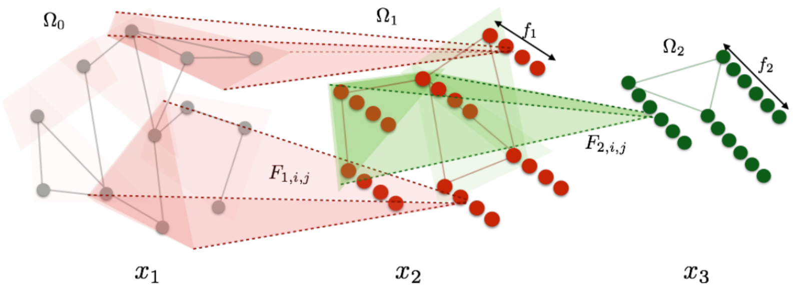
Joan Bruna et al. Spectral Networks and Deep Locally Connected Networks on Graphs. ICLR 2014

Multi-Scale PointNet v.s. Graph CNN

- In Graph CNN’s perspective:
- Multi-Scale PointNet defines
  1. Graph connectivity through Euclidean distance
  2. Graph coarsening by farthest point sampling
  3. Local feature extraction with PointNet (v1.0)



Multi-scale PointNet

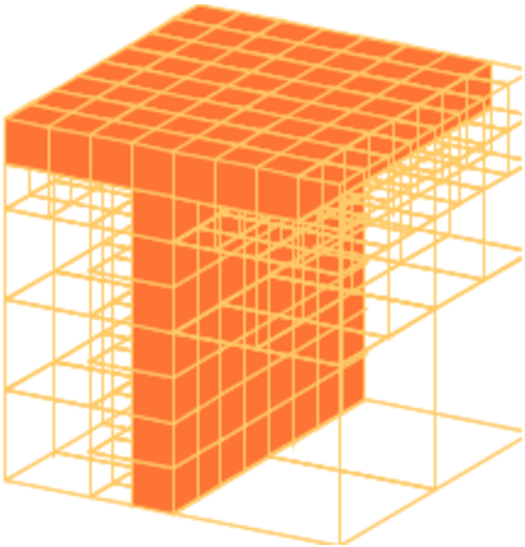


Graph CNN

Relations to OctNet (Octree based 3D CNN)

OctNet in Graph CNN’s perspective:

1. Both connectivity and graph coarsening are defined by the Octree.
2. Local feature extraction by convolution layer.



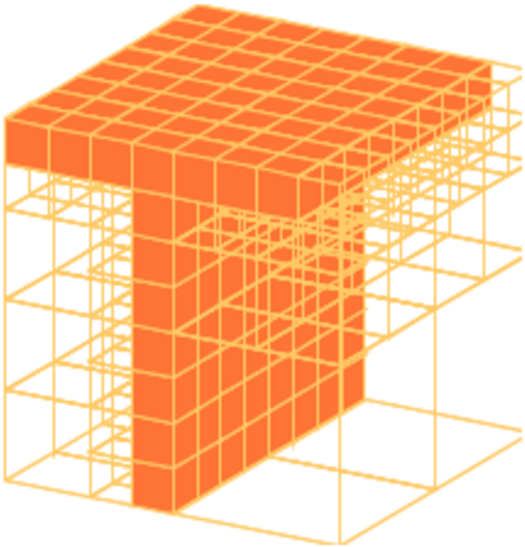
OctNet: Learning Deep 3D Representations at High Resolutions
Gernot Riegler, Ali Osman Ulusoy and Andreas Geiger

Relations to OctNet (Octree based 3D CNN)

OctNet in Graph CNN's perspective:

In Multi-Scale PointNet

1. Both connectivity and graph coarsening are defined by the Octree. By ground distance
2. Local feature extraction by convolution layer. By PointNet (v1.0)



OctNet: Learning Deep 3D Representations at High Resolutions
Gernot Riegler, Ali Osman Ulusoy and Andreas Geiger