Making 3D Predictions with 2D Supervision

Justin Johnson

UMich / FAIR

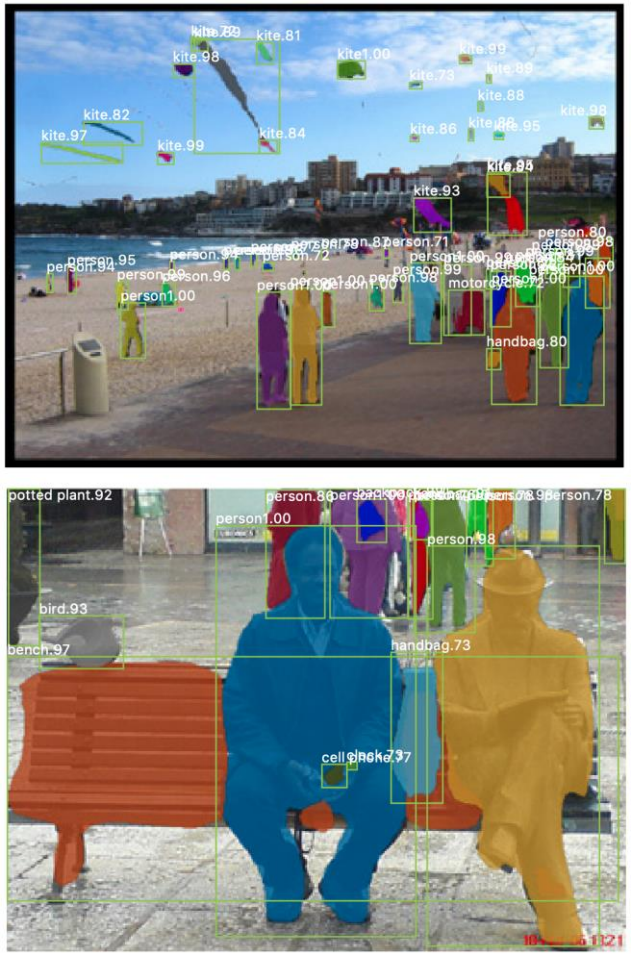
6/15/2020

2D Recognition works very well!

Kirillov et al, “Panoptic Feature Pyramid Networks”, CVPR 2019

A new dimension for recognition

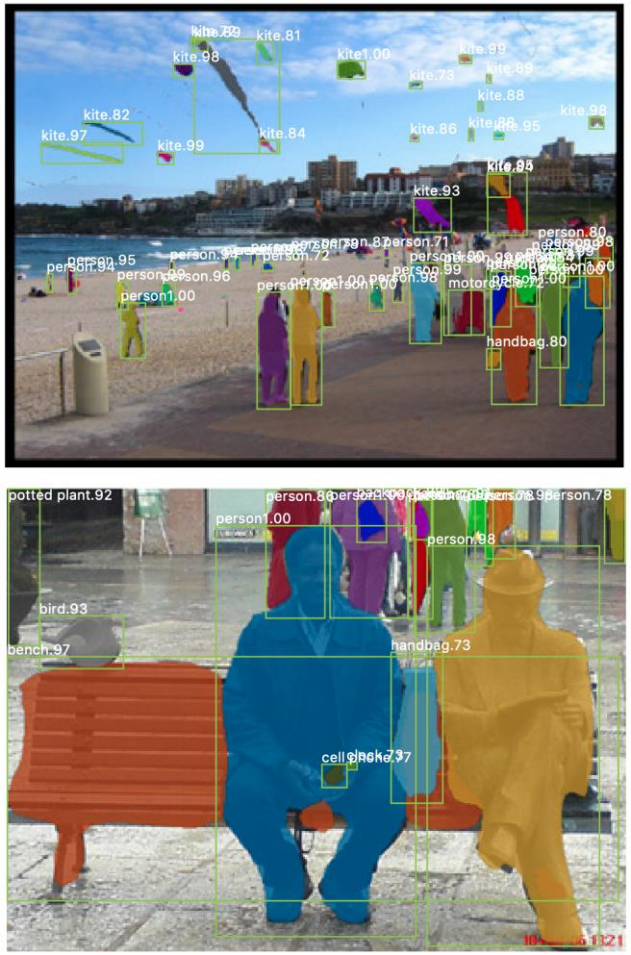
Mask R-CNN:
2D Image -> 2D shapes



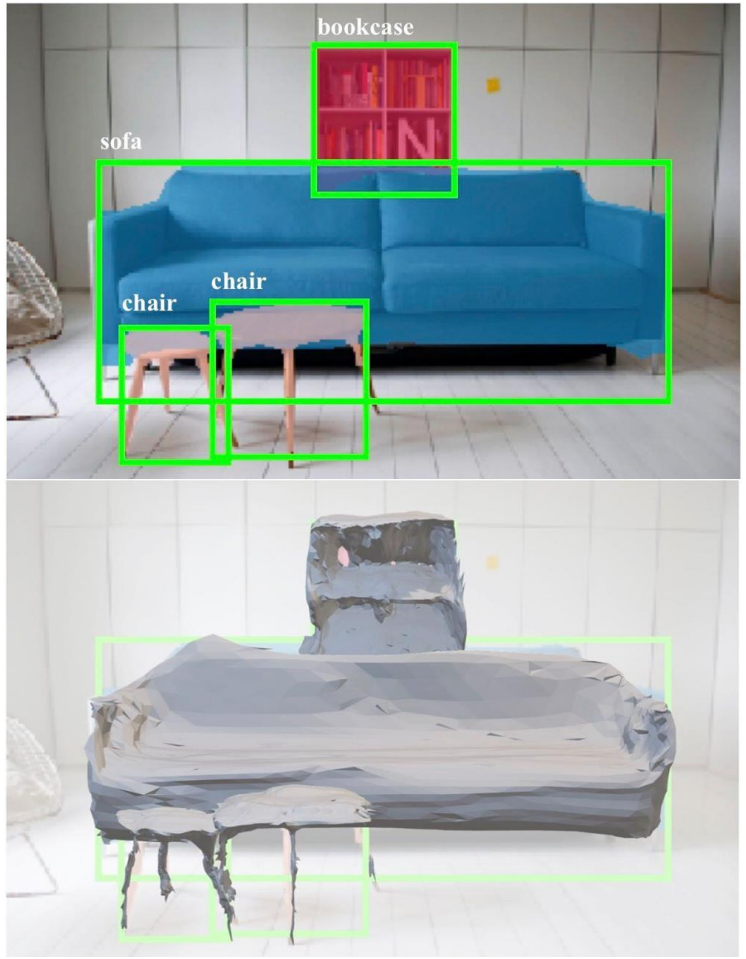
He, Gkioxari, Dollár, and Girshick, “Mask R-CNN”, ICCV 2017

A new dimension for recognition

Mask R-CNN:
2D Image -> 2D shapes



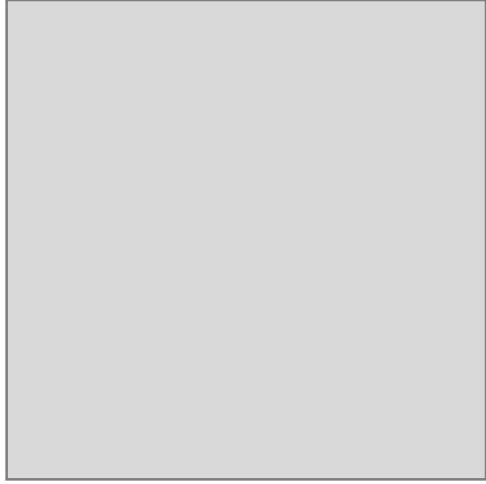
Mesh R-CNN:
2D Image -> **3D** shapes



He, Gkioxari, Dollár, and Girshick, "Mask R-CNN", ICCV 2017

Gkioxari, Malik, and Johnson, "Mesh R-CNN", ICCV 2019

Why care about 3D perception?



Why care about 3D perception?

Autonomous Vehicles

Why care about 3D perception?

Autonomous Vehicles

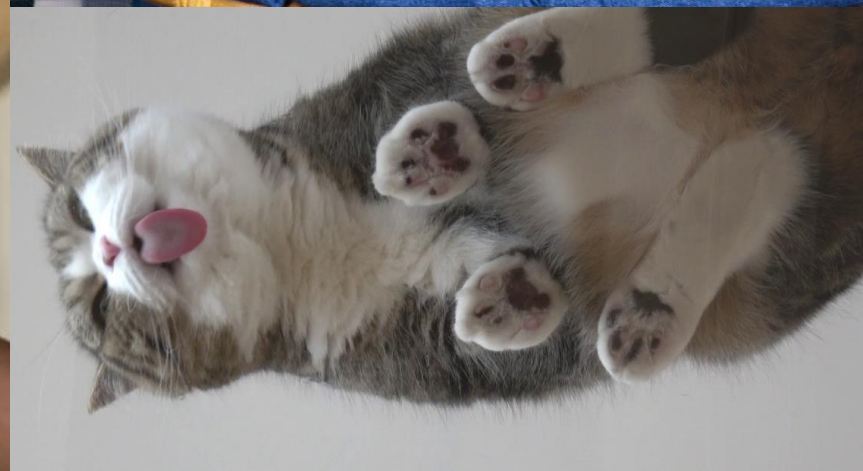
VR / AR

Why care about 3D perception?

Autonomous Vehicles

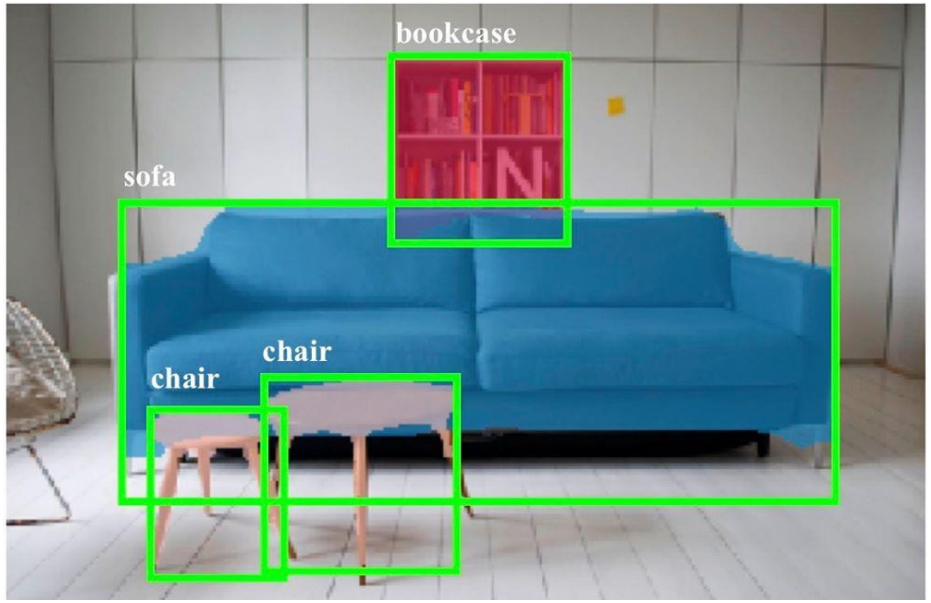
VR / AR

The world is 3D!



Key Challenge for 3D Recognition: Supervision

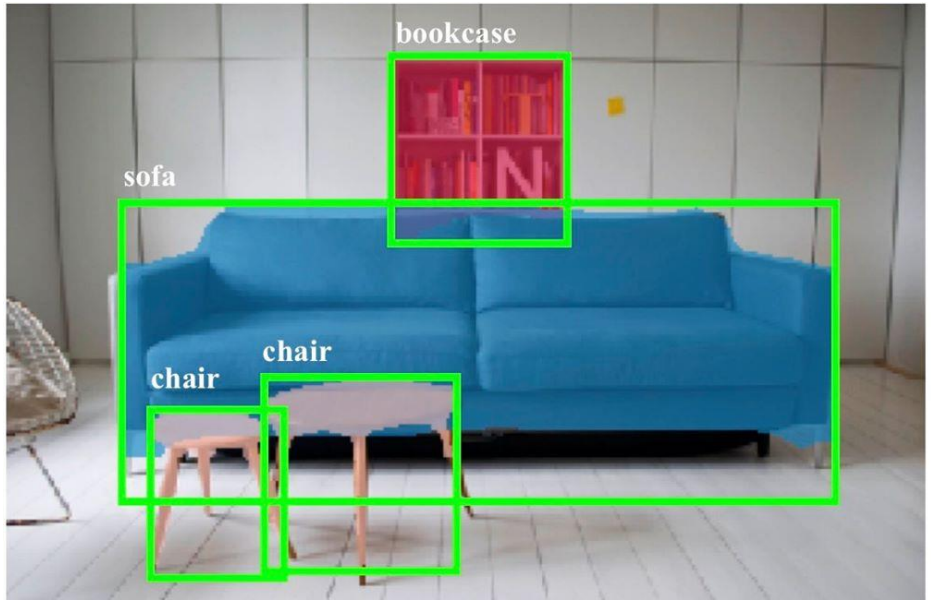
2D supervision:
Category label, bounding box, segmentation



Can be annotated by non-expert humans

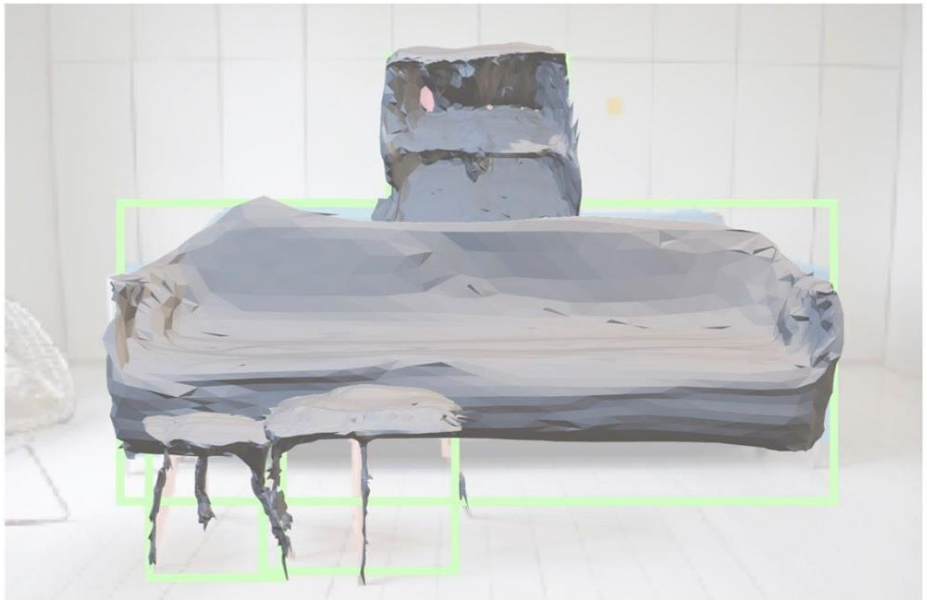
Key Challenge for 3D Recognition: Supervision

2D supervision:
Category label, bounding box, segmentation



Can be annotated by non-expert humans

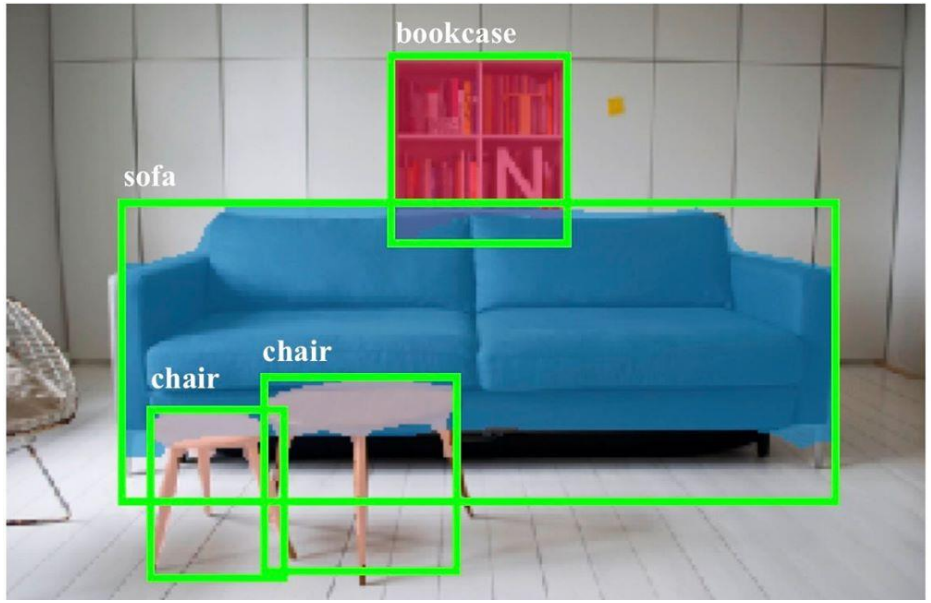
3D supervision:
3D shape, depth, camera pose, etc



Not easily annotated by people!

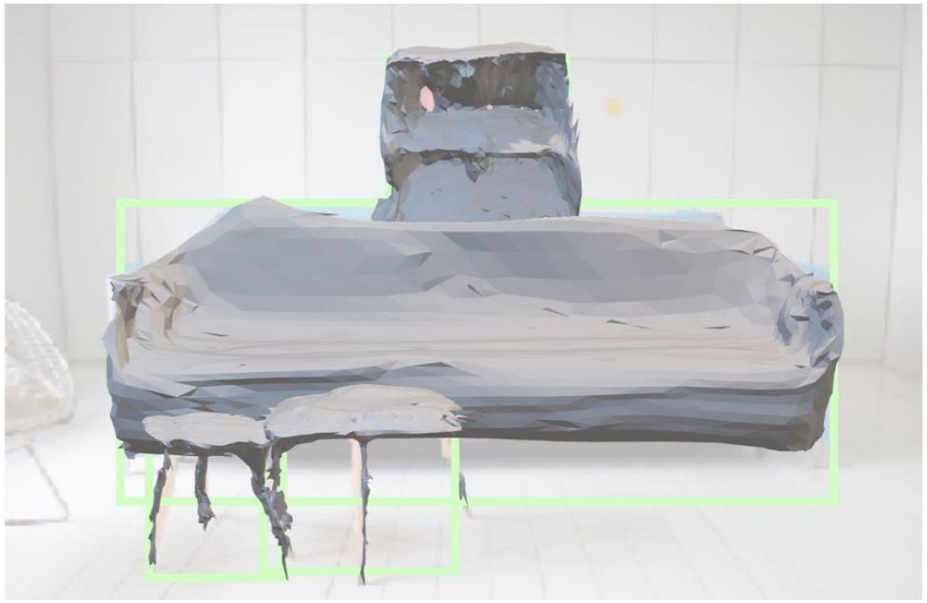
Key Challenge for 3D Recognition: Supervision

2D supervision:
Category label, bounding box, segmentation



Can be annotated by non-expert humans

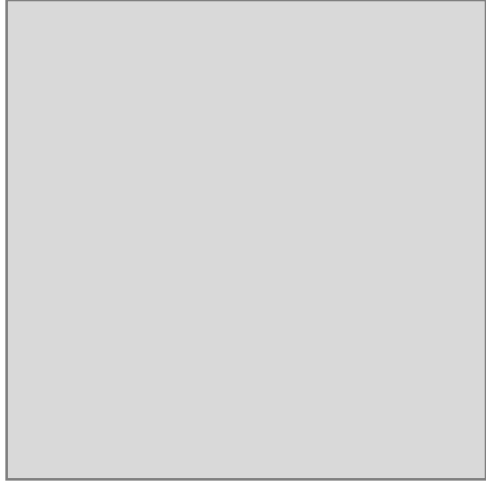
3D supervision:
3D shape, depth, camera pose, etc



Not easily annotated by people!

Claim: For 3D recognition to succeed, we must rely primarily on 2D supervision

3D Predictions from 2D Supervision



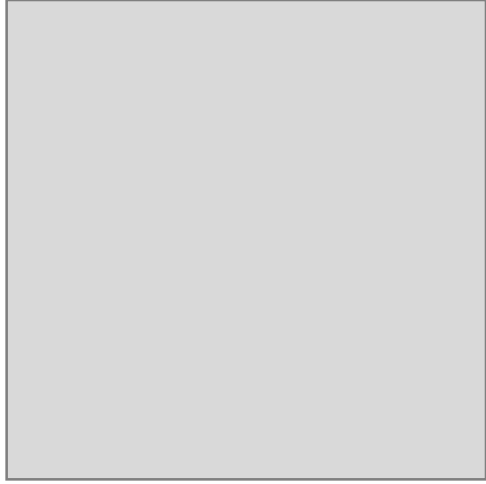
Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

3D Predictions from 2D Supervision



Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

Mesh R-CNN

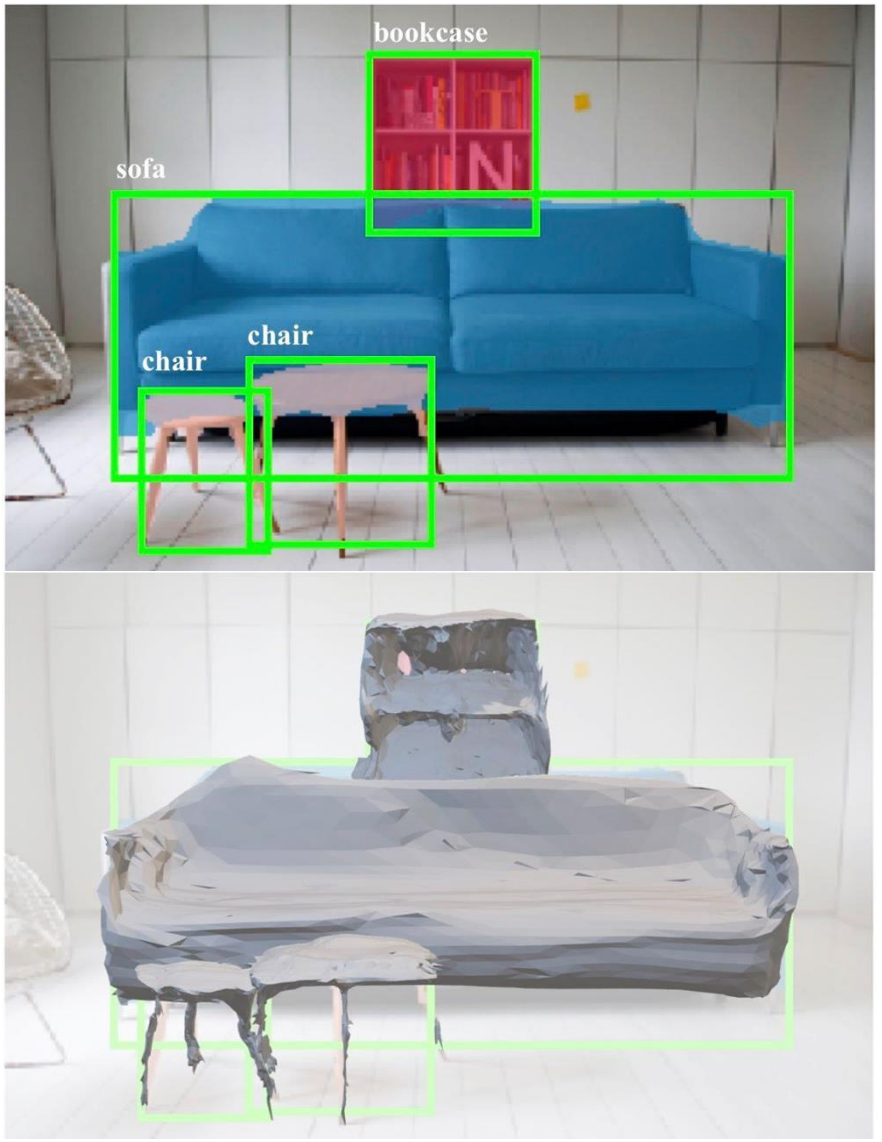
Georgia Gkioxari

Jitendra Malik

Justin Johnson

ICCV 2019

Mesh R-CNN: Task



Input: Single RGB image

Output:

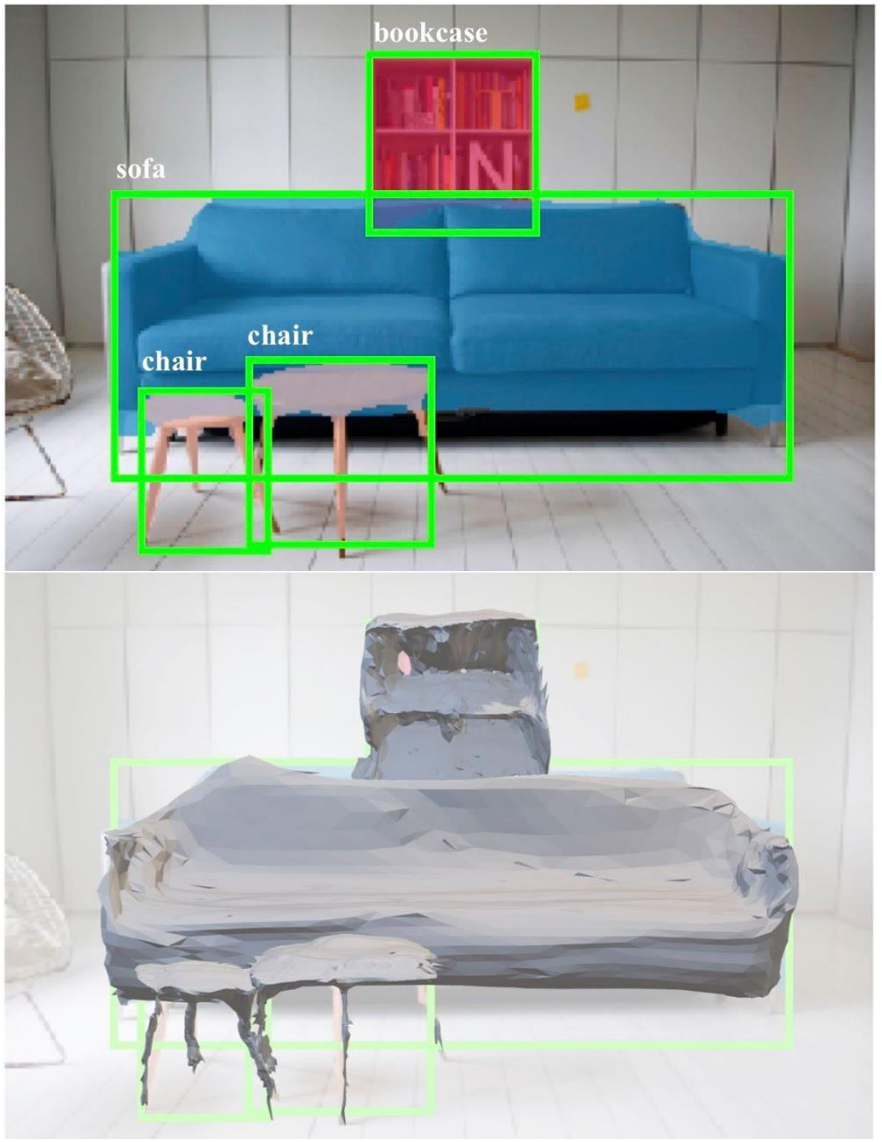
A set of detected objects

For each object:

- Bounding box
- Category label
- Instance segmentation
- 3D triangle mesh

Gkioxari, Malik, and Johnson, “Mesh R-CNN”, ICCV 2019

Mesh R-CNN: Task



Input: Single RGB image

Output:

A set of detected objects

For each object:

- Bounding box
- Category label
- Instance segmentation
- 3D triangle mesh

Mask R-CNN

Mesh head

Gkioxari, Malik, and Johnson, “Mesh R-CNN”, ICCV 2019

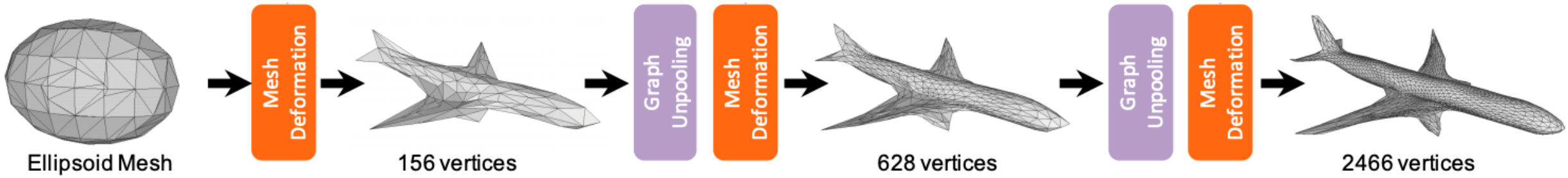
Predicting Meshes with Neural Networks

Iterative mesh refinement:

Start from initial ellipsoid mesh

Network predicts offsets for each vertex

Repeat.



Wang et al, “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images”, ECCV 2018

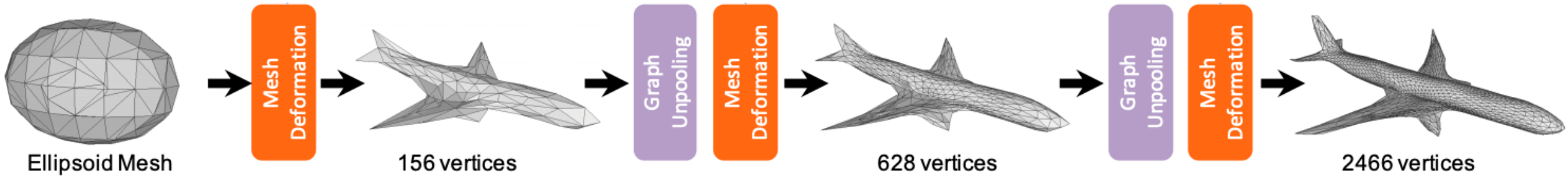
Predicting Meshes with Neural Networks

Iterative mesh refinement:

Start from initial ellipsoid mesh

Network predicts offsets for each vertex

Repeat.



Problem: Can't model objects with holes!

Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

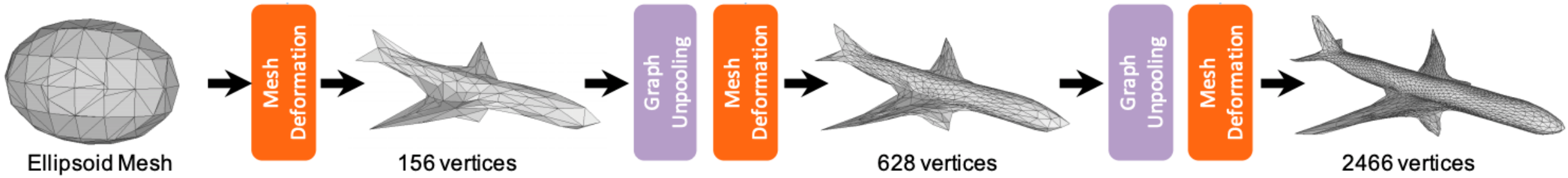
Predicting Meshes with Neural Networks

Iterative mesh refinement:

Start from initial ellipsoid mesh

Network predicts offsets for each vertex

Repeat.



Problem: Can't model objects with holes!

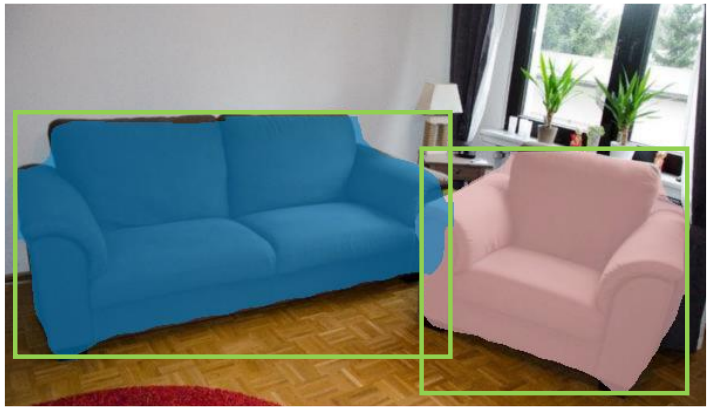
Solution: Hybrid shape representation

Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Mesh R-CNN Pipeline

Input image

2D object recognition



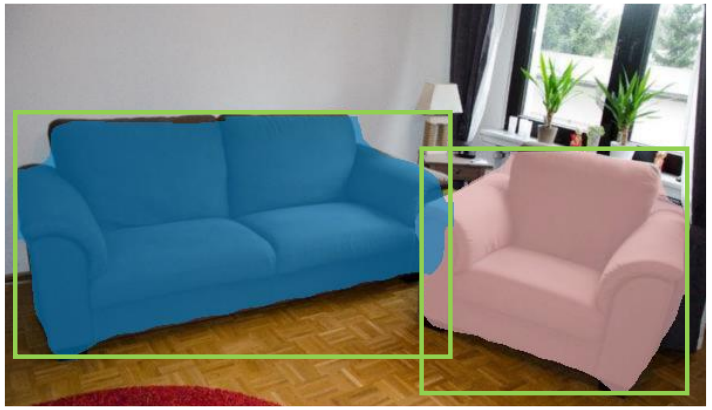
Mesh R-CNN Pipeline

Input image

Mesh R-CNN Pipeline

Input image

2D object recognition

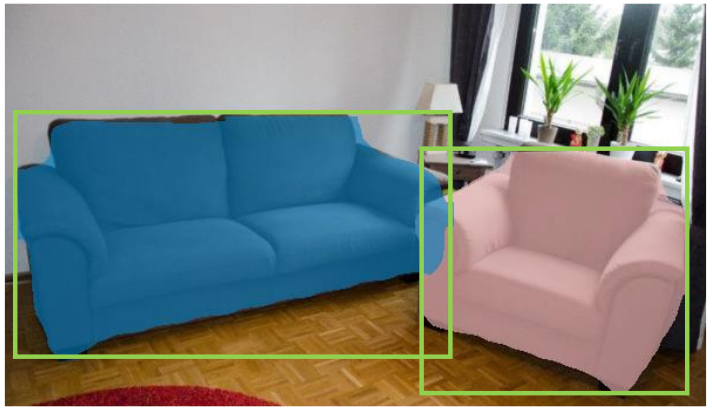


3D object voxels

Mesh R-CNN Pipeline

Input image

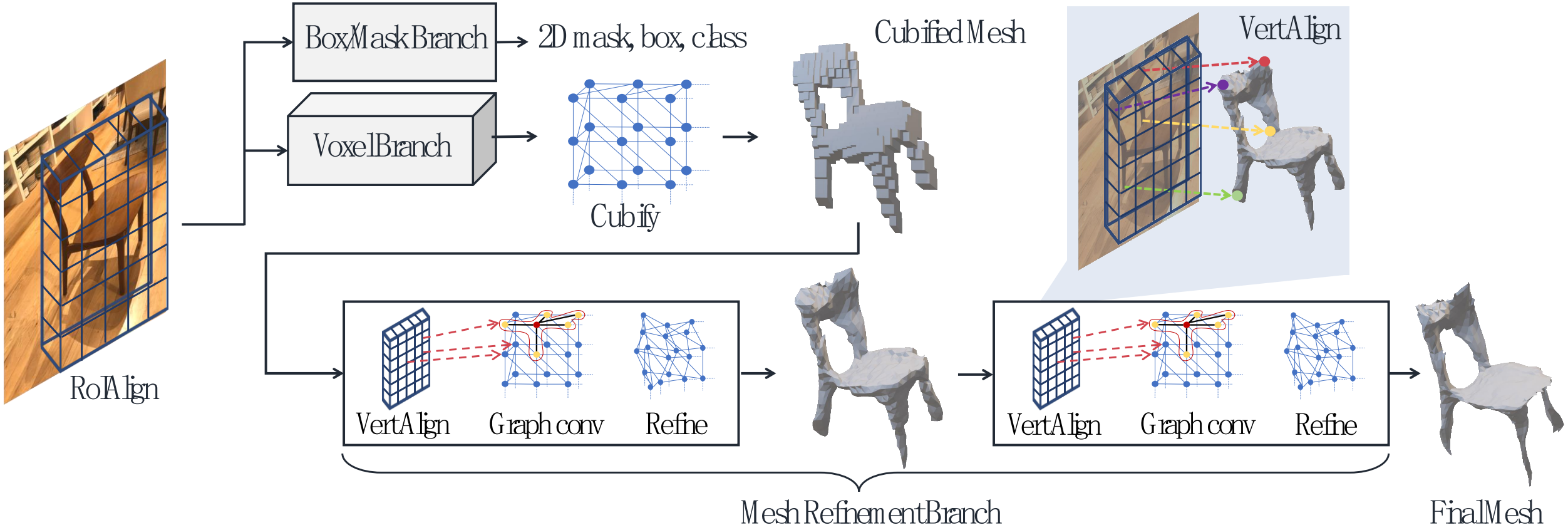
2D object recognition



3D object voxels

3D object meshes

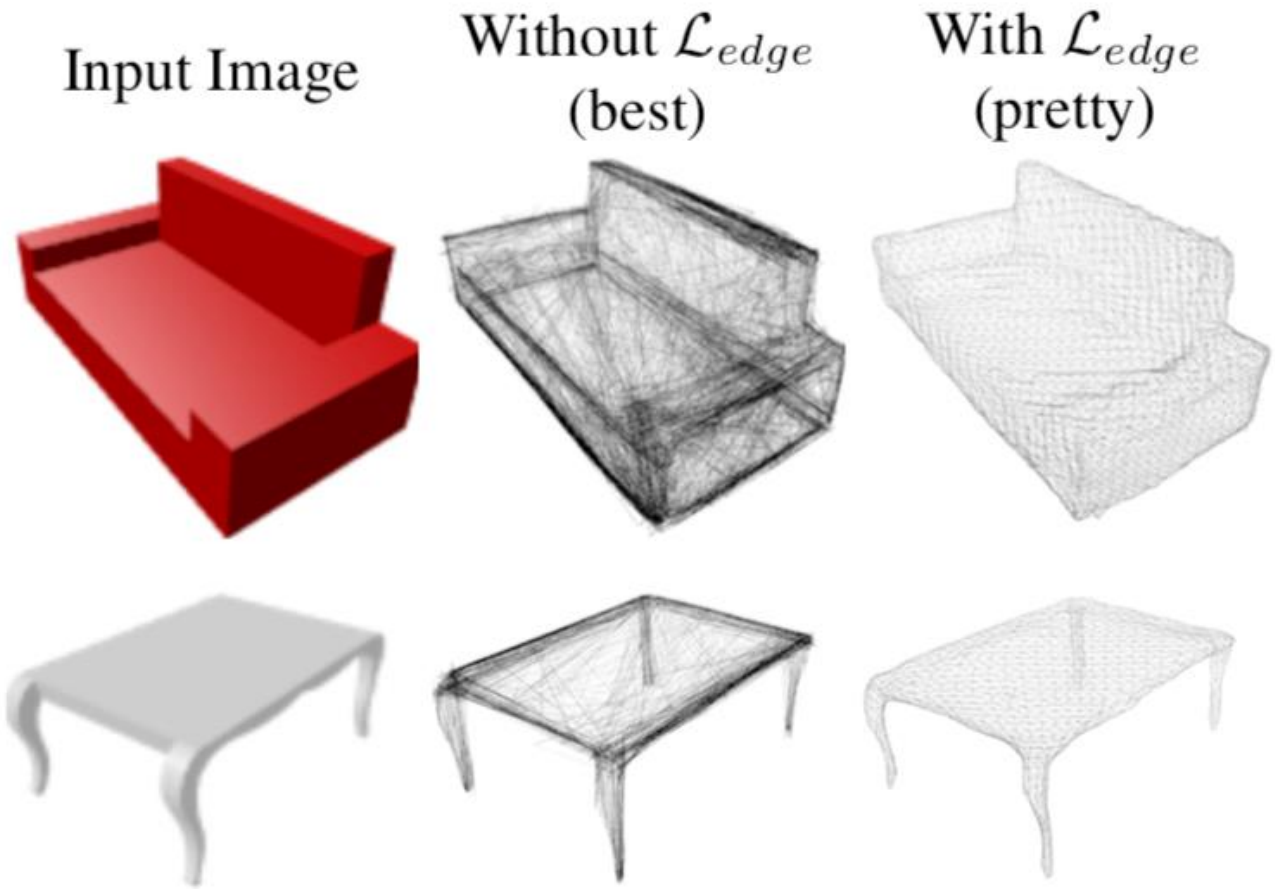
Mesh R-CNN: Architecture



Mesh R-CNN: Training Losses

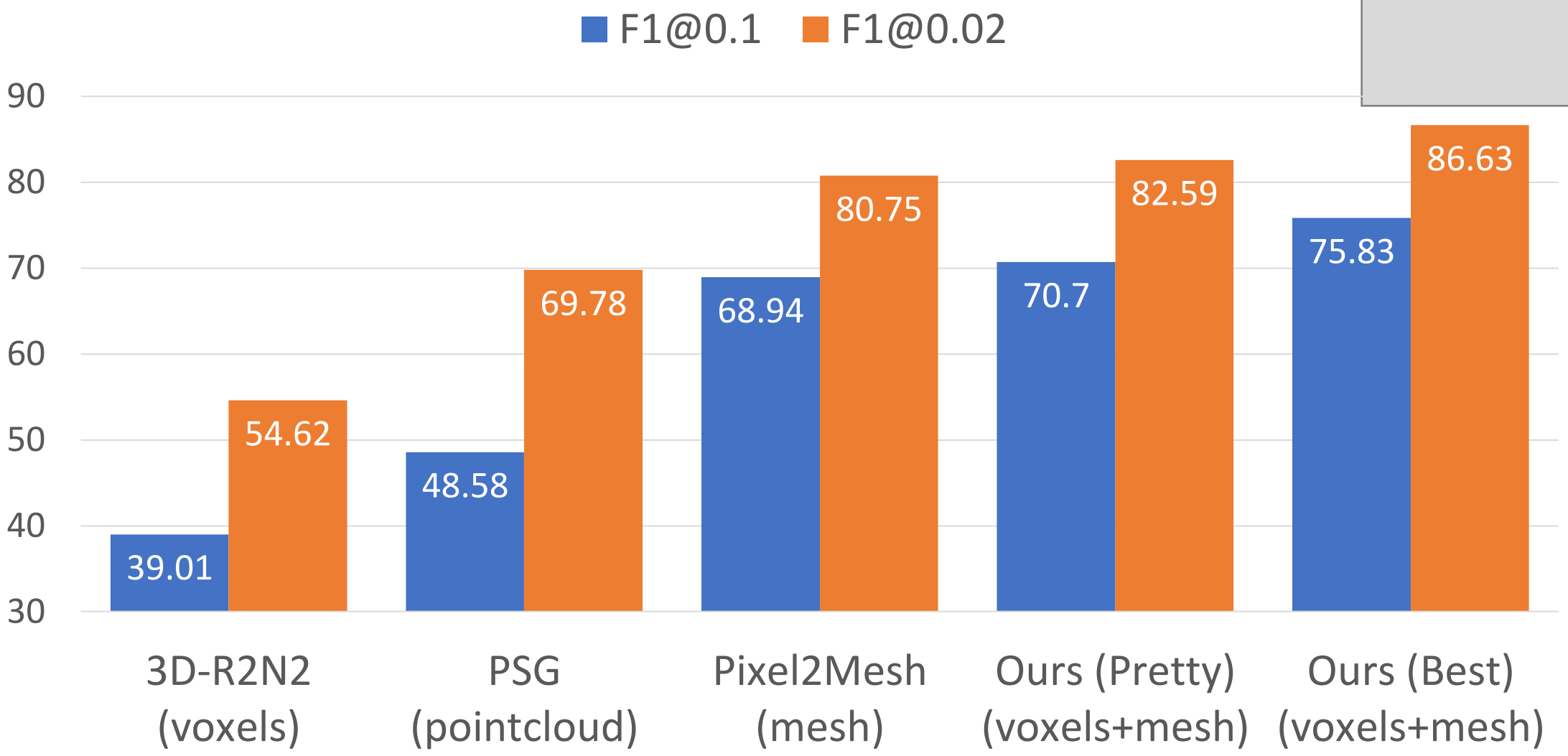
- Instance segmentation losses: Same as Mask R-CNN
  - RPN classification
  - RPN bounding box regression
  - Per-region classification
  - Per-region bounding box regression
  - Per-region instance segmentation mask
- Voxel loss: Binary cross-entropy loss on voxel occupancy
- Mesh loss: Chamfer distance on sampled points at each stage
- Mesh regularizer: Minimize length of edges in mesh

Results on ShapeNet: Best vs Pretty

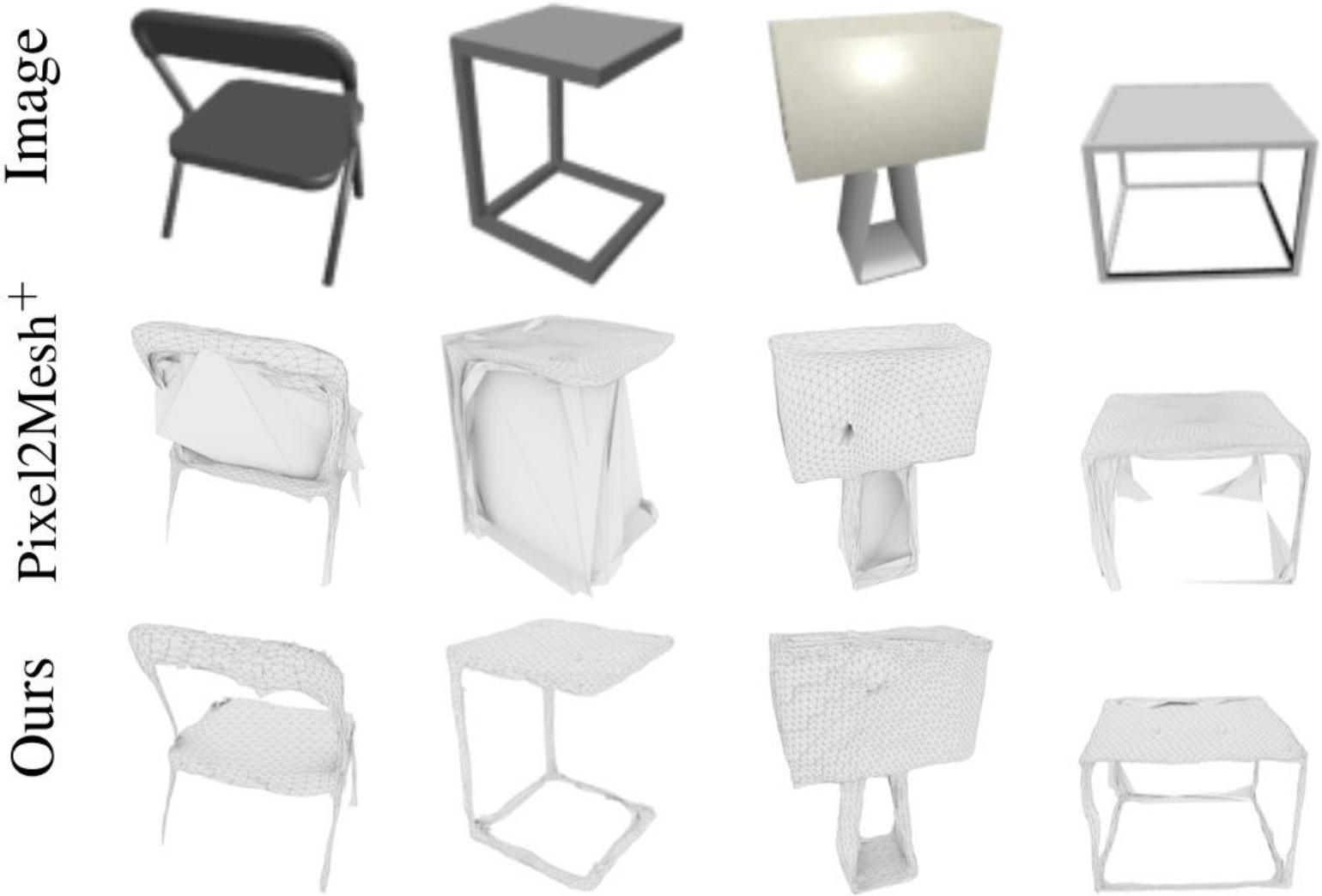


Using regularizers produces smooth shapes but metrics are worse. Note that the metrics do not capture “smoothness”.

Mesh R-CNN: ShapeNet results



Mesh R-CNN: ShapeNet Results



Results on Pix3D

Task & Metrics:

- Detect all objects in the image: AP^{box}
- Predict their instance mask: AP^{mask}
- Predict their 3D shape: AP^{mesh}



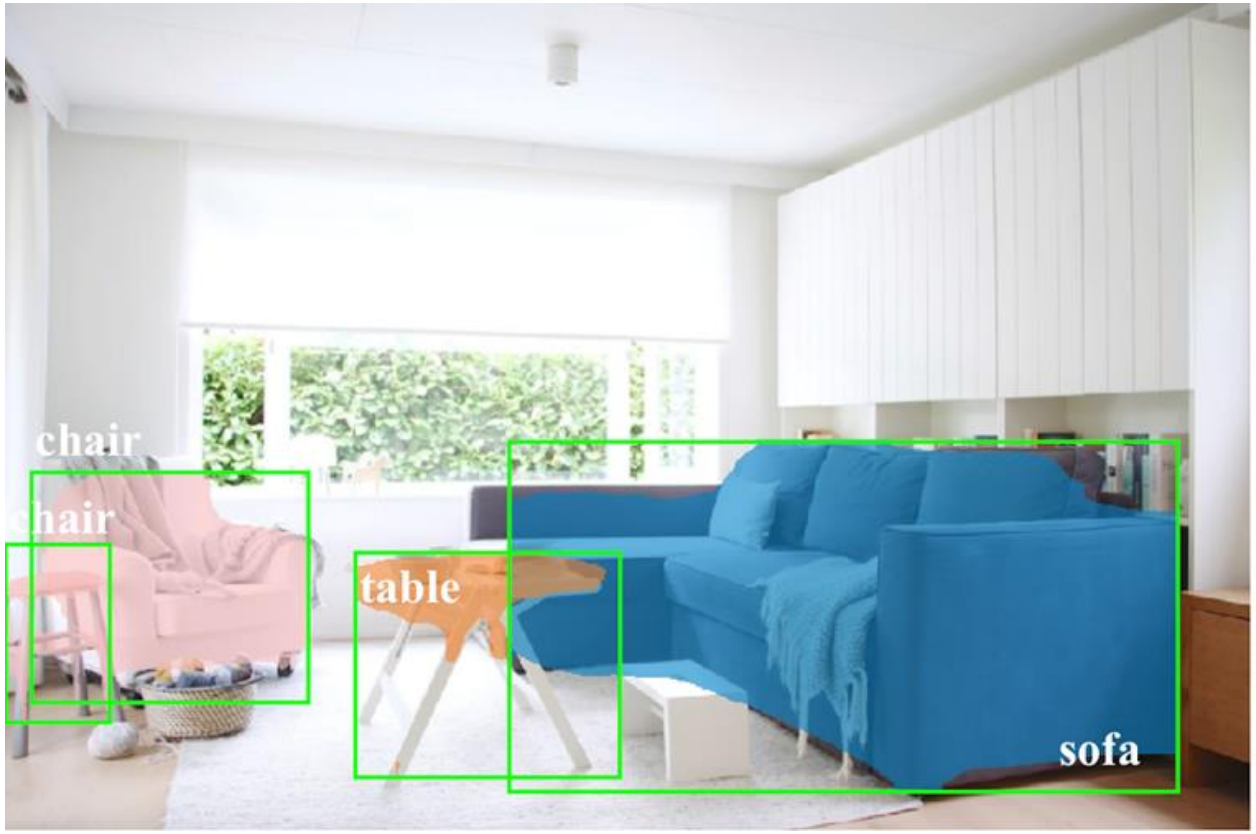
Definition of AP^{mesh}:

- A detection is true positive if the predicted class is correct, it is not a duplicate detection and its $F1^{0.3} > 0.5$

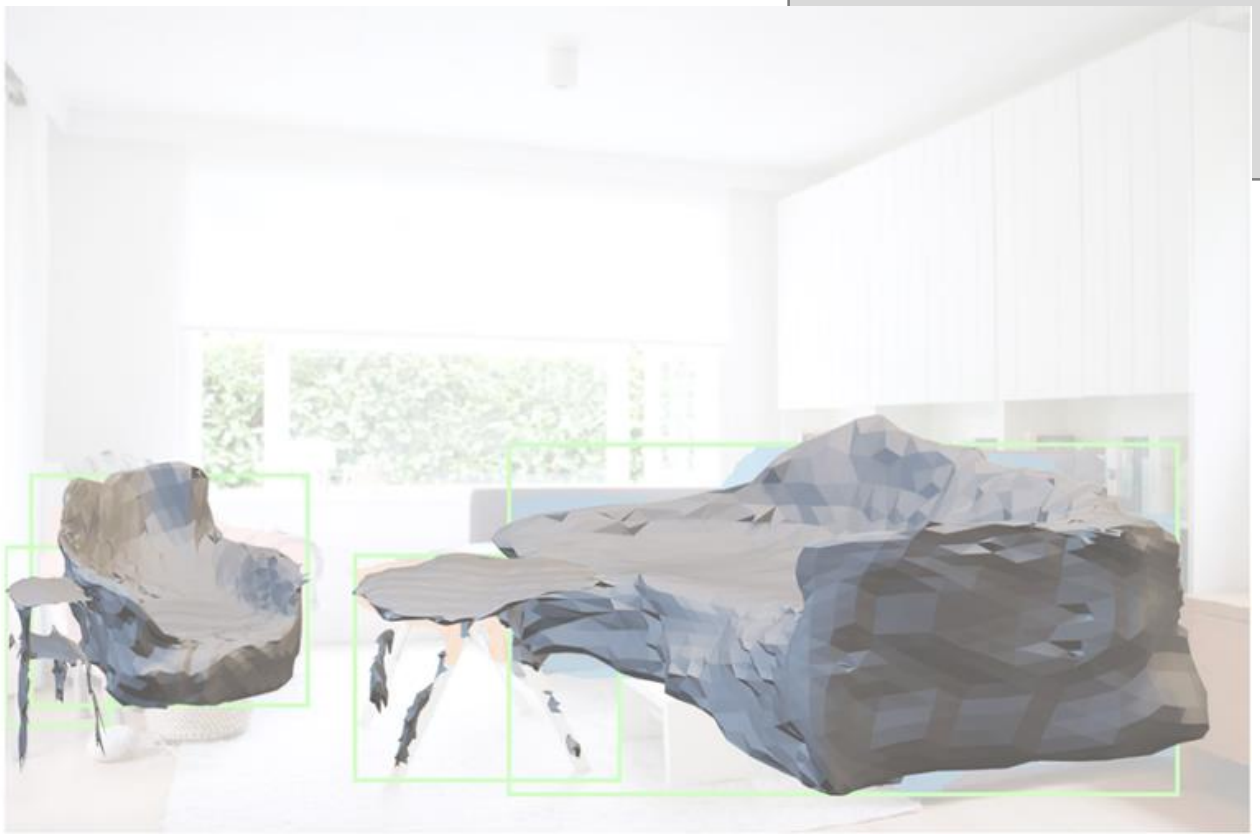
Results on Pix3D (Novel Images)

Results on Pix3D

Predicting many objects per scene



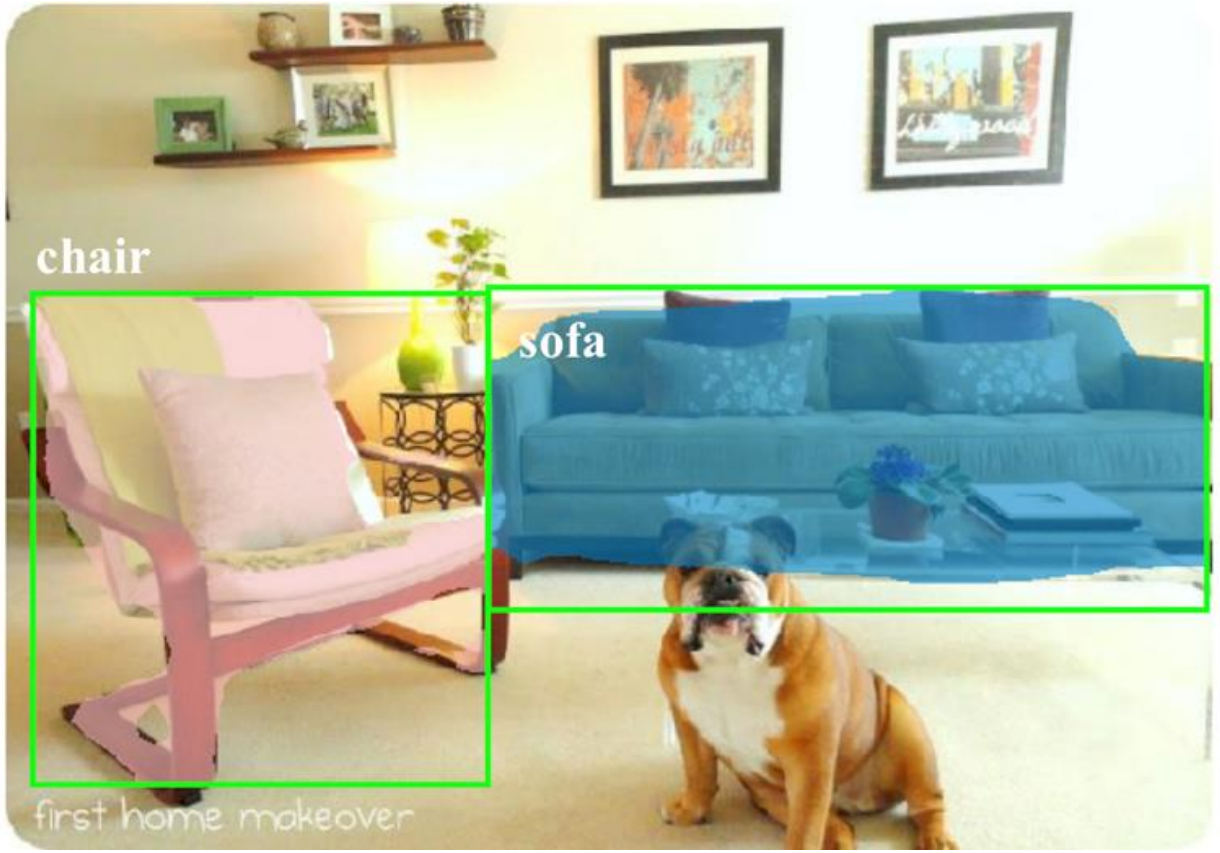
Box & Mask Predictions



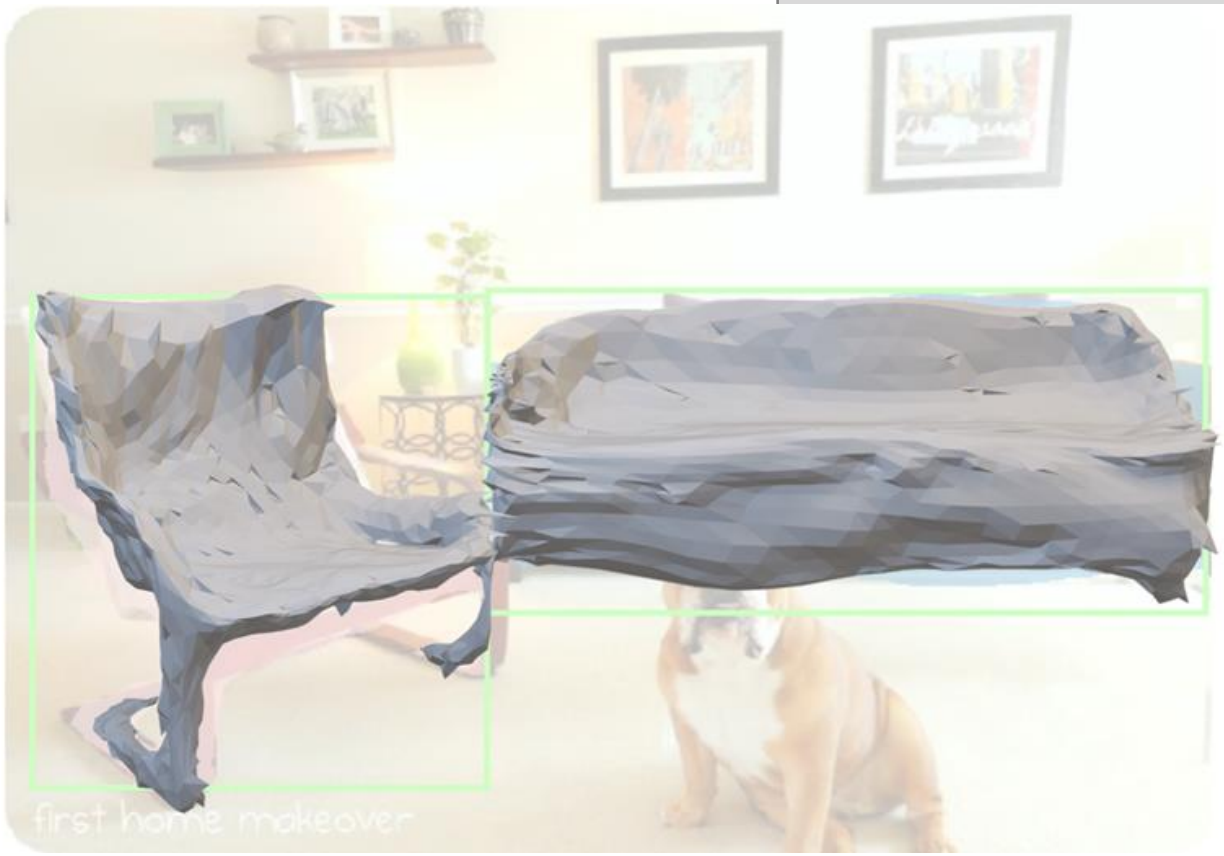
Mesh Predictions

Results on Pix3D (S_1)

Amodal completion: predict occluded parts of objects



Box & Mask Predictions



Mesh Predictions

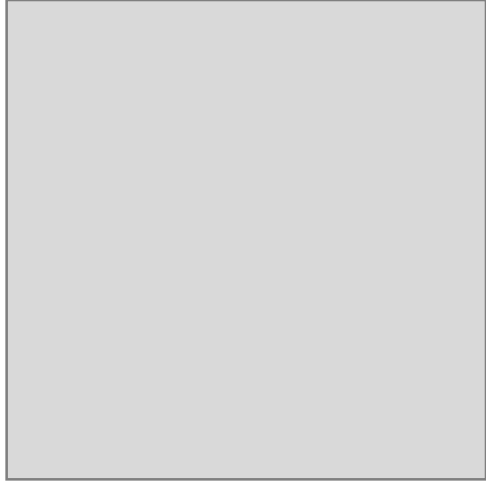
Results on Pix3D (S_1)

Segmentation failures propagate to meshes

Box & Mask Predictions

Mesh Predictions

3D Predictions from 2D Supervision



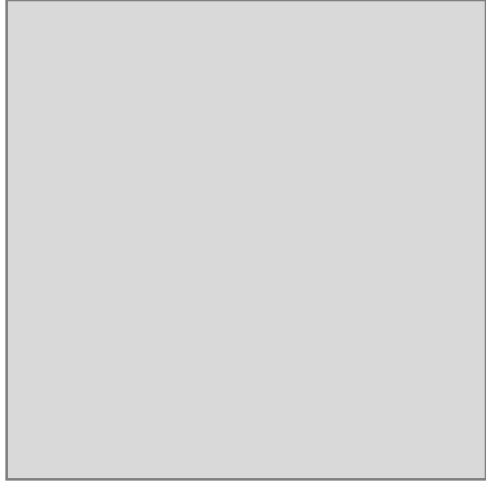
Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

3D Predictions from 2D Supervision



Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

Problem: Mesh R-CNN requires 3D Supervision

Input Image

Model

3D Prediction

Loss Function

Ground-truth 3D

Problem: 3D supervision is expensive to obtain!

Idea: Render and Compare

Input Image

Model

3D Prediction

Render

Gradient

2D Image

Loss Function

Solution: Use only **2D supervision** by using a rendering engine!

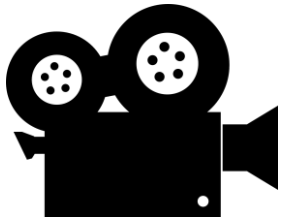
Problem: Need to compute gradients through renderer

Ground-truth **2D**

Differentiable Rendering

$$\frac{\partial L}{\partial (mesh)}$$

$$\frac{\partial L}{\partial (camera)}$$



$$\frac{\partial L}{\partial (texture)}$$

Rendering Engine

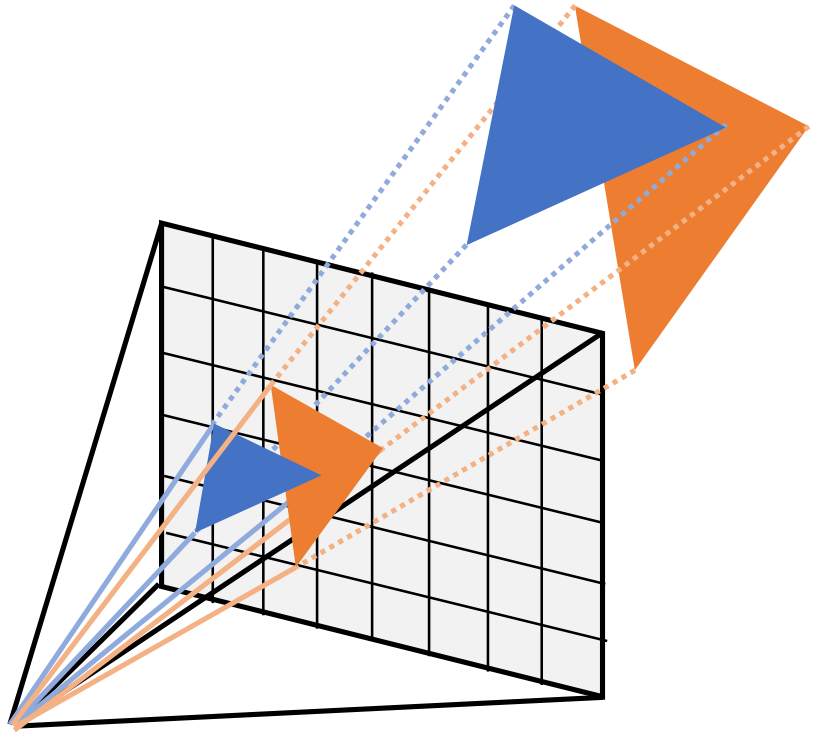
$$\frac{\partial L}{\partial I}$$

Scene Properties

RGB Image

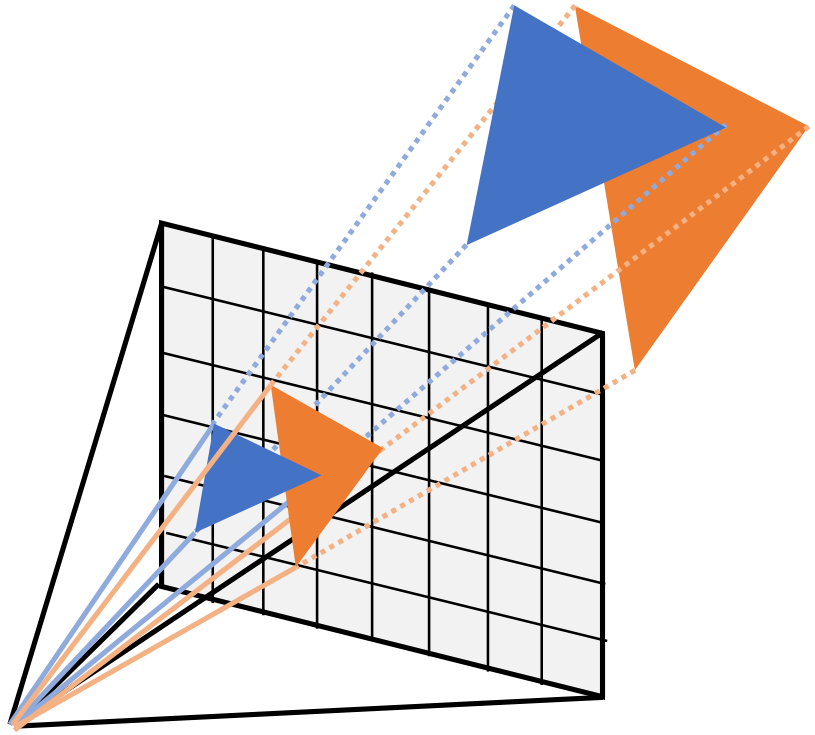
Loper and Black, "OpenDR: An Approximate Differentiable Renderer", ECCV 2014
Kato et al, "Neural 3D Mesh Renderer", CVPR 2018
Li et al, "Differentiable Monte Carlo Ray Tracing through Edge Sampling", SIGGRAHP Asia 2018
Liu et al, "Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning", ICCV 2019
Chen et al, "Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer", NeurIPS 2019

Traditional Rendering

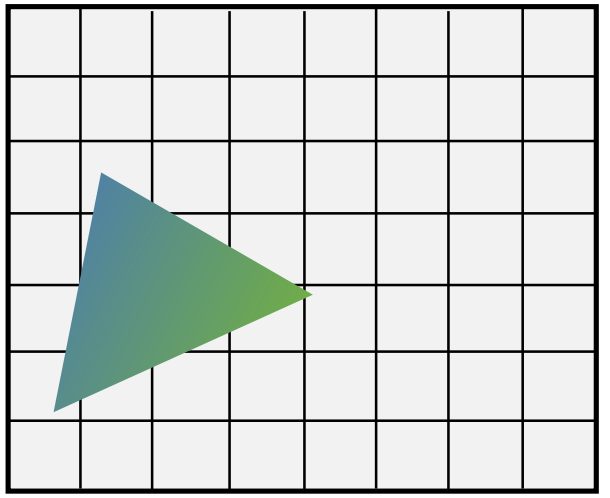


Rasterization: Check which primitives hit each point

Traditional Rendering

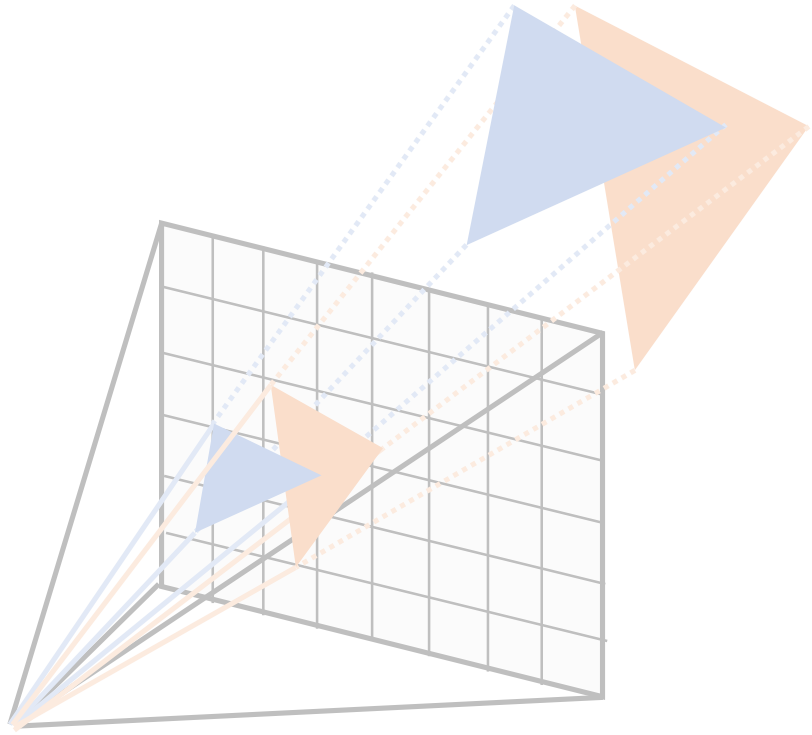


Rasterization: Check which primitives hit each point



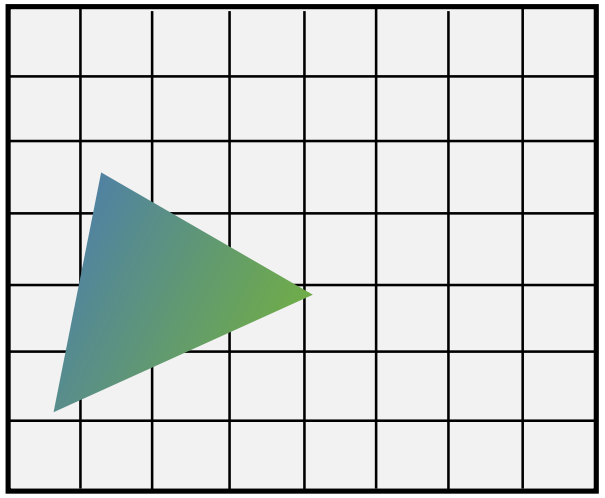
Shading: Compute the color of each pixel

Traditional Rendering



Rasterization: Check which primitives hit each point

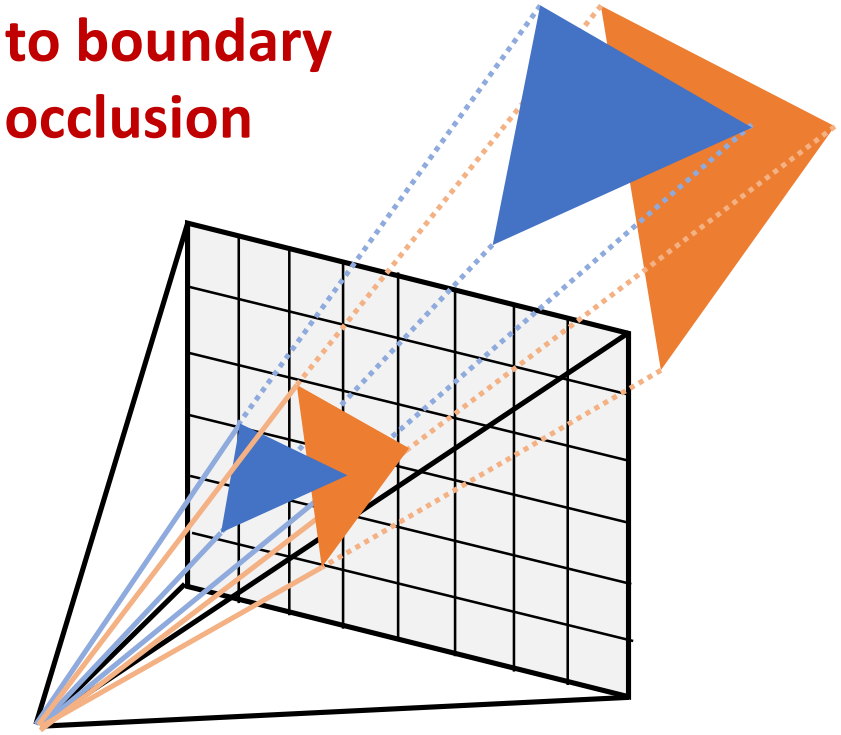
Differentiable! Interpolate vertex data (depth, color, normal, texture, etc) over face interior



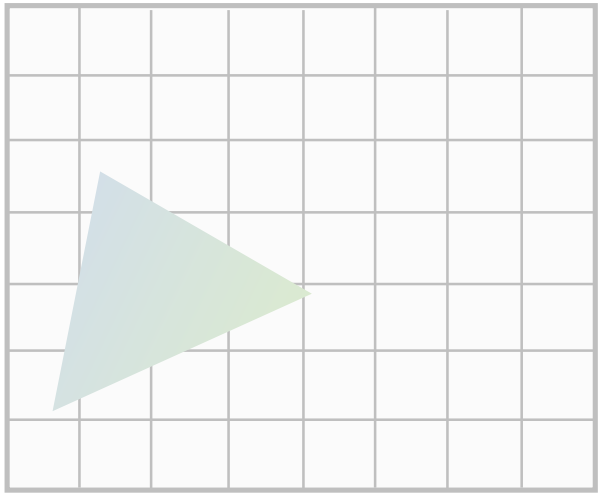
Shading: Compute the color of each pixel

Traditional Rendering

Not differentiable due to boundary and occlusion



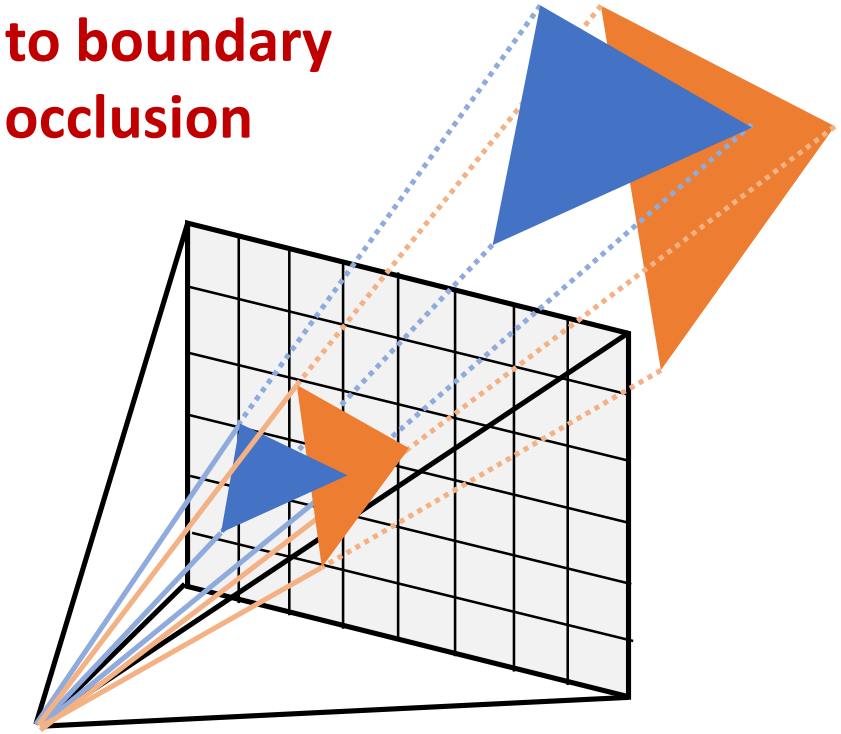
Rasterization: Check which primitives hit each point



Shading: Compute the color of each pixel

Traditional Rendering

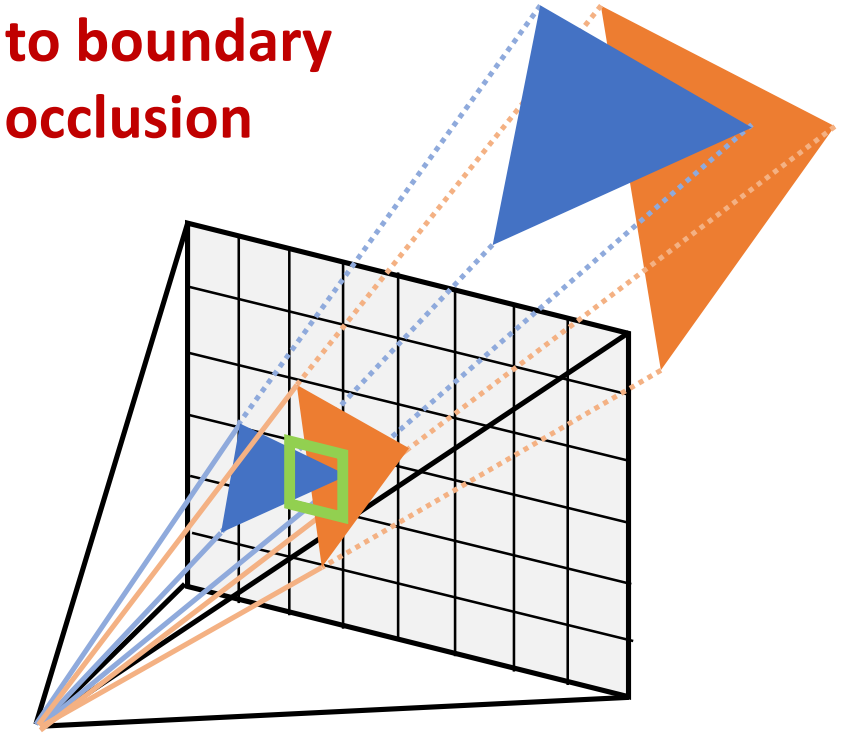
Not differentiable due to boundary and occlusion



Rasterization: Check which primitives hit each point

Traditional Rendering

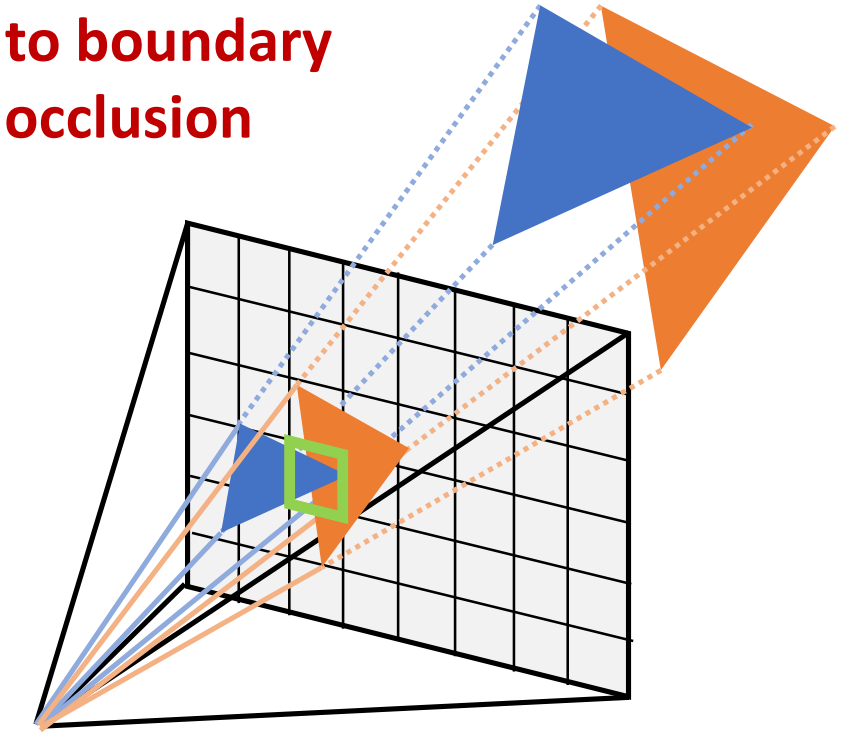
Not differentiable due to boundary and occlusion



Rasterization: Check which primitives hit each point

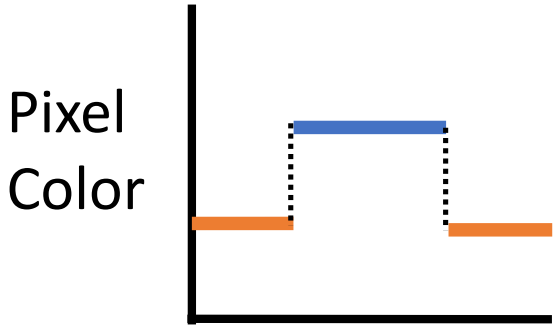
Traditional Rendering

Not differentiable due to boundary and occlusion



Rasterization: Check which primitives hit each point

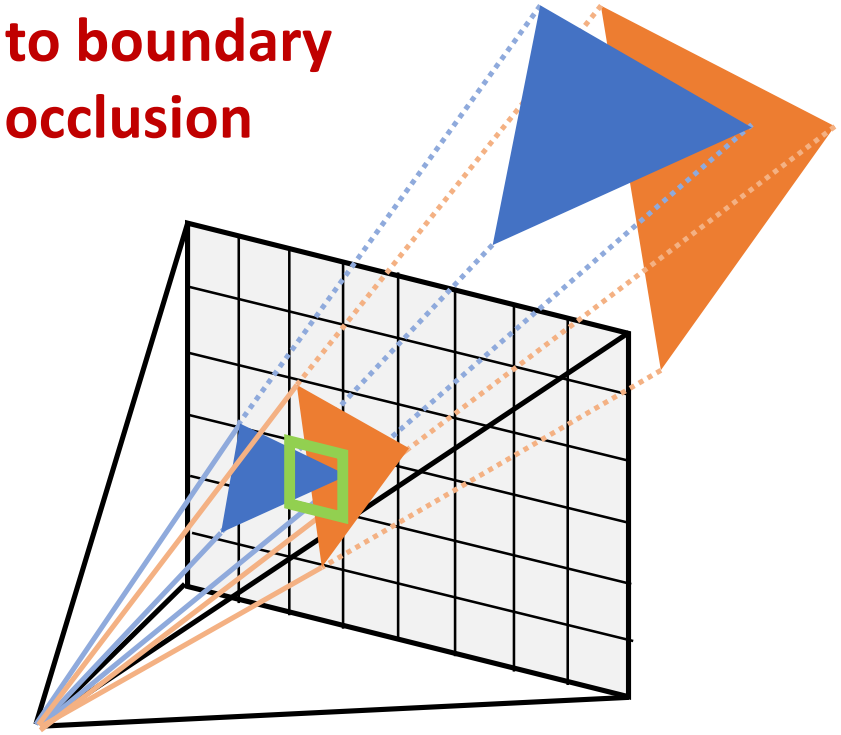
XY non-differentiable due to boundary



X-coord of blue face

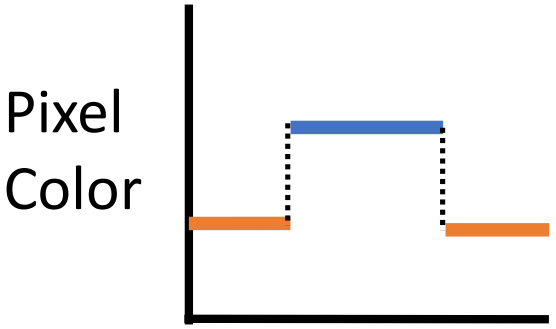
Traditional Rendering

Not differentiable due to boundary and occlusion



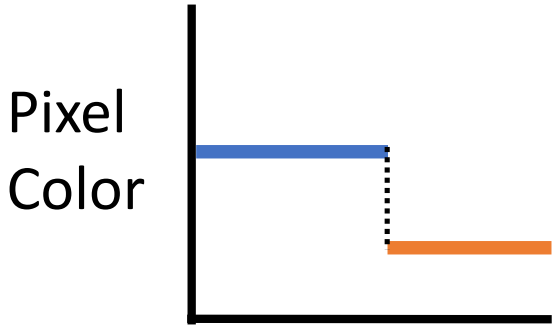
Rasterization: Check which primitives hit each point

XY non-differentiable due to boundary



X-coord of blue face

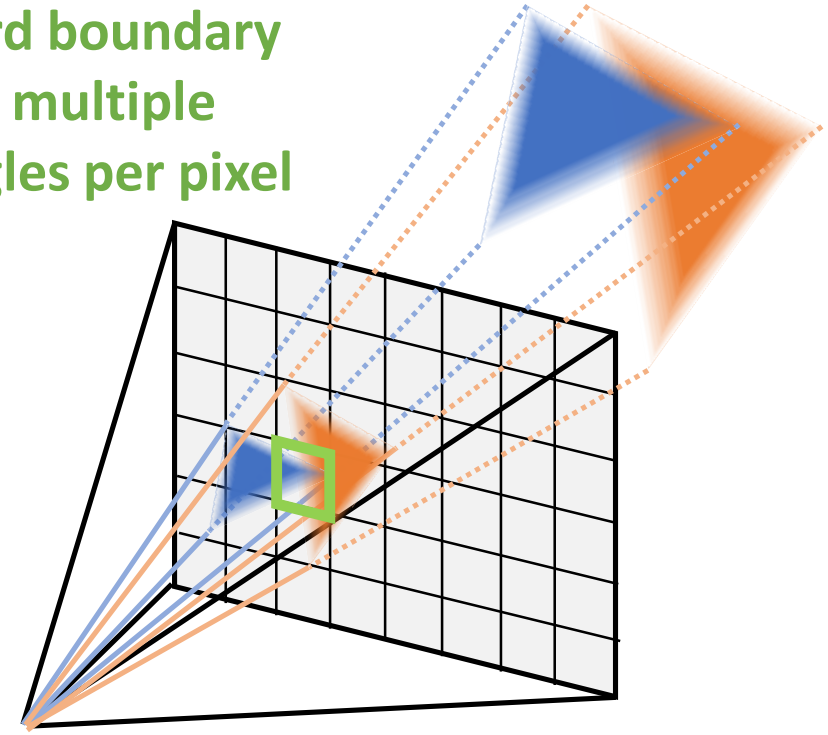
Z non-differentiable due to occlusion



Z-coord of blue face

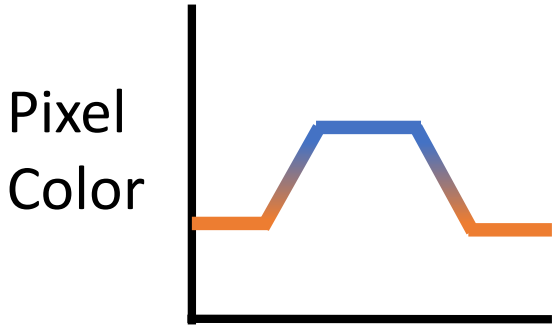
Differentiable Rasterization

1. **Make triangles transparent toward boundary**
2. **Blend multiple triangles per pixel**

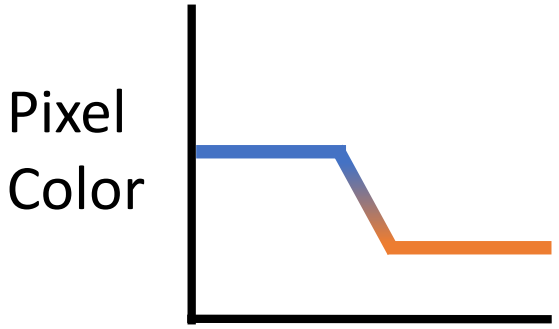


Rasterization: Check which primitives hit each point

Pixel color differentiable w/respect to triangle position!



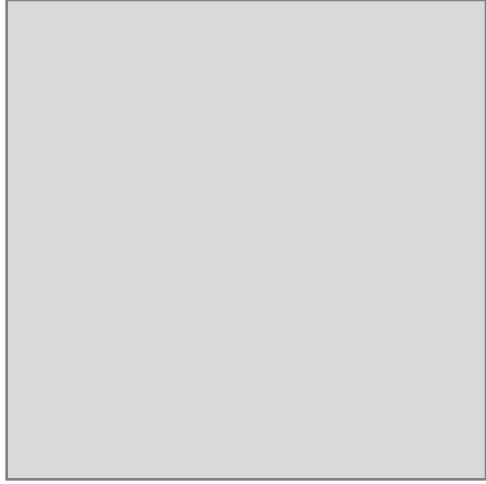
X-coord of blue face



Z-coord of blue face

Liu et al, “Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning”, ICCV 2019

3D Predictions from 2D Supervision



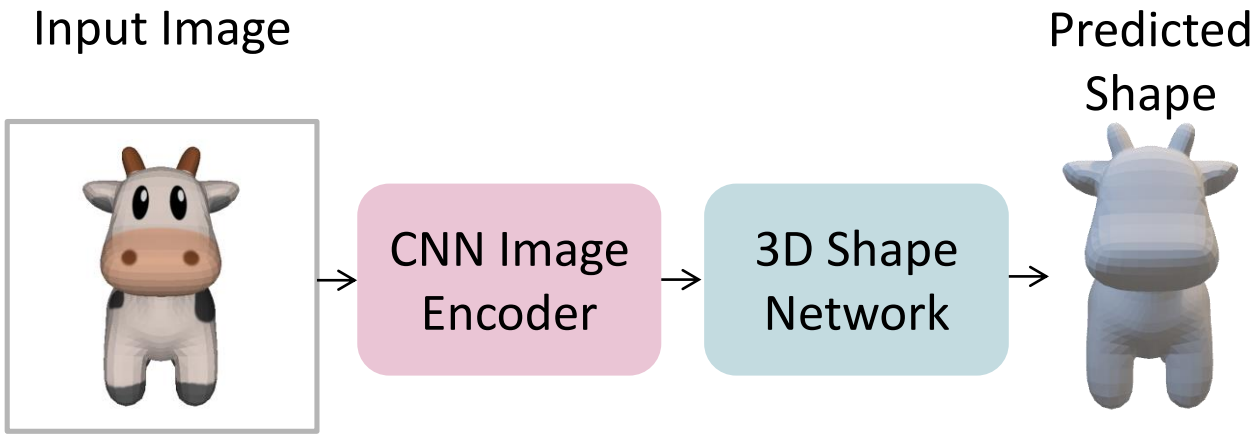
Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

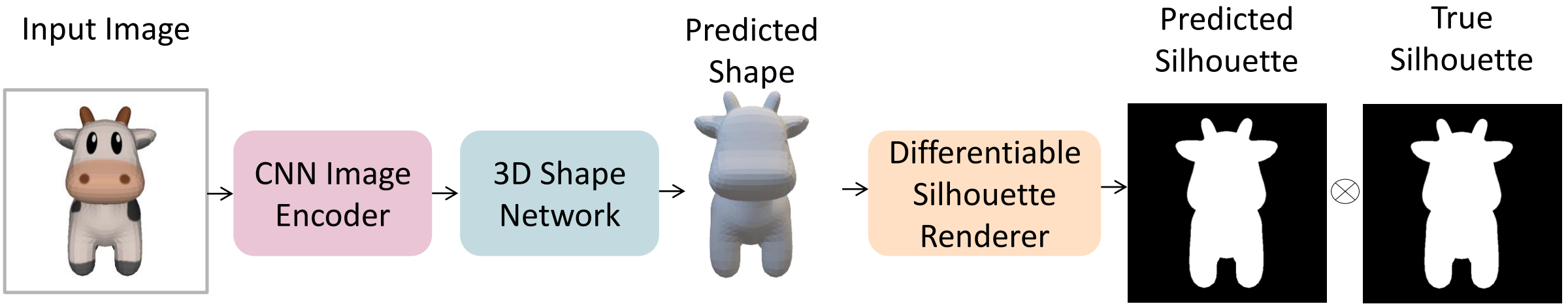
Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

Unsupervised Shape Prediction



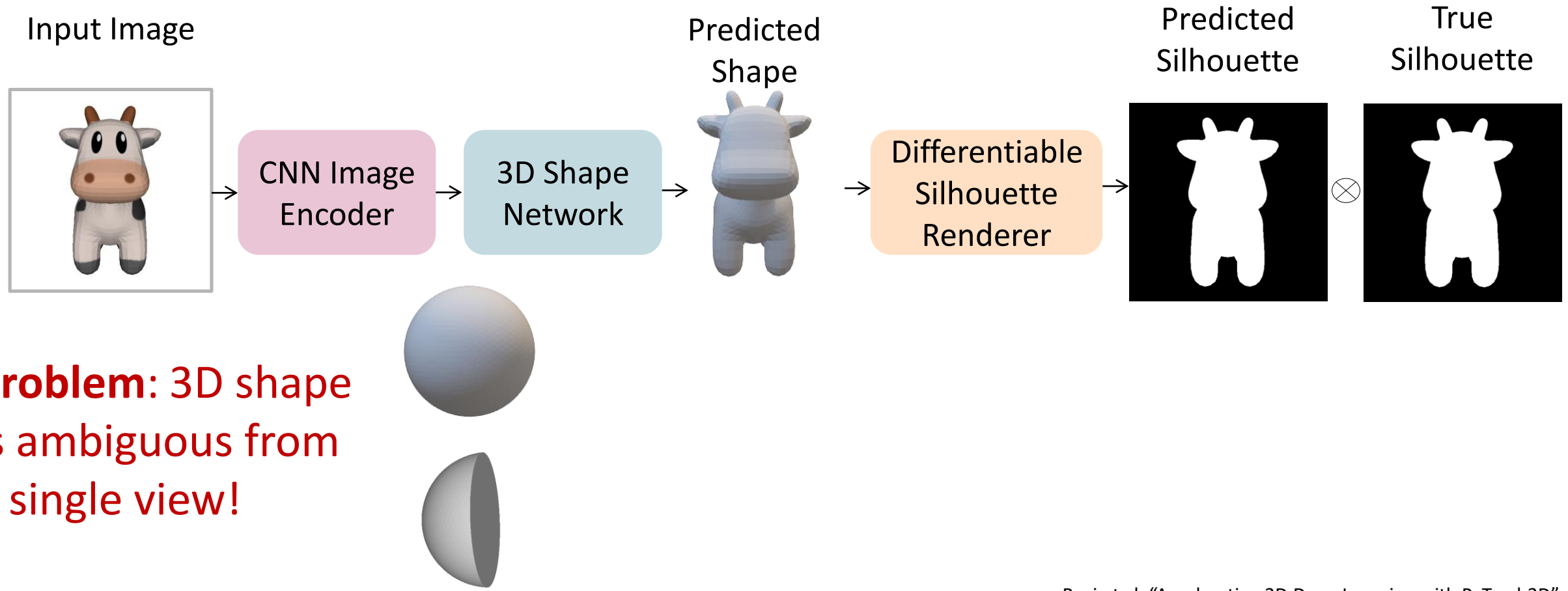
Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Unsupervised Shape Prediction



Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

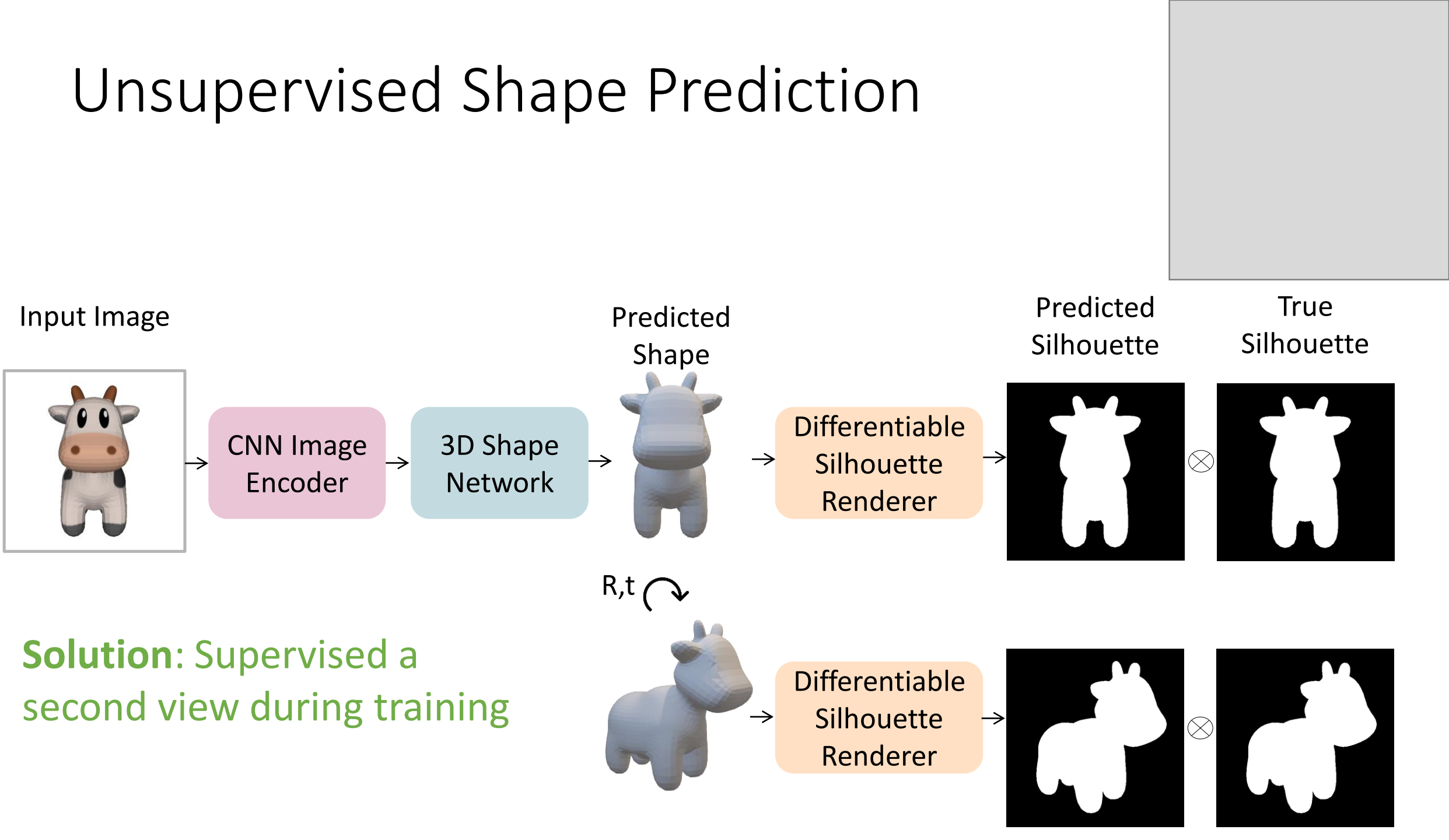
Unsupervised Shape Prediction



Problem: 3D shape is ambiguous from a single view!

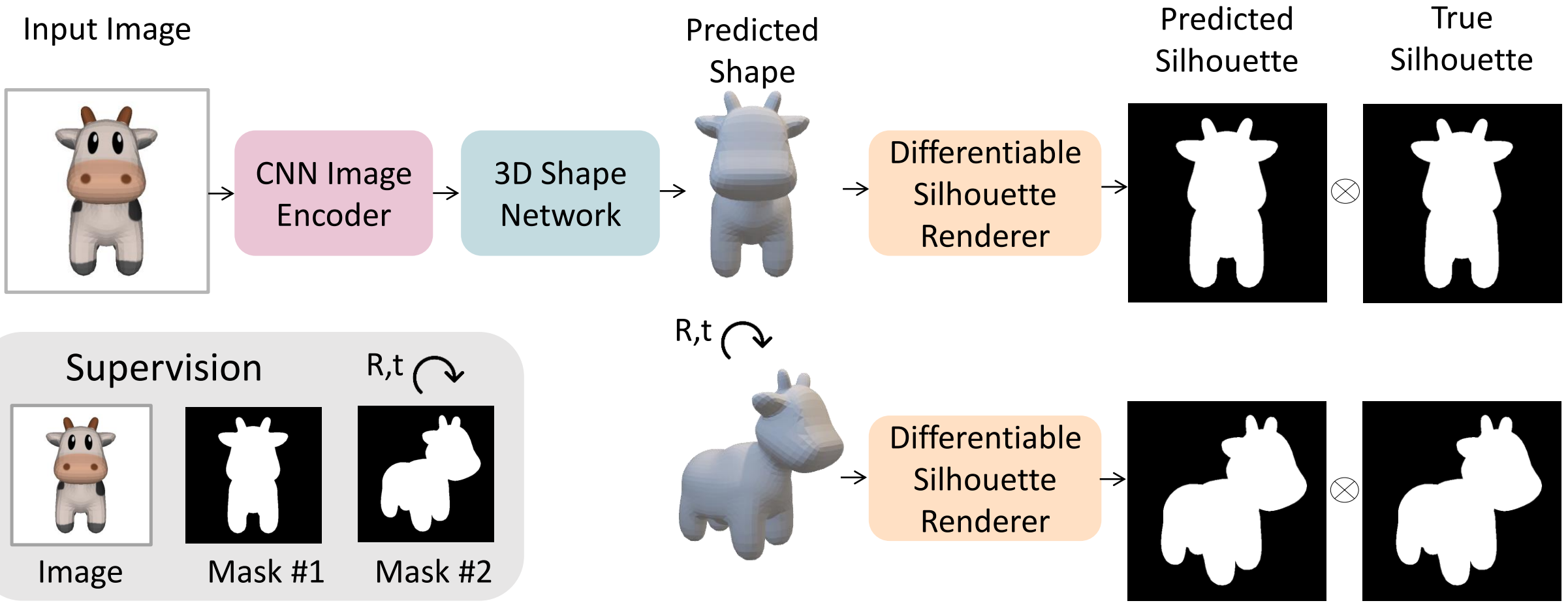
Ravi et al, "Accelerating 3D Deep Learning with PyTorch3D", 2020

Unsupervised Shape Prediction

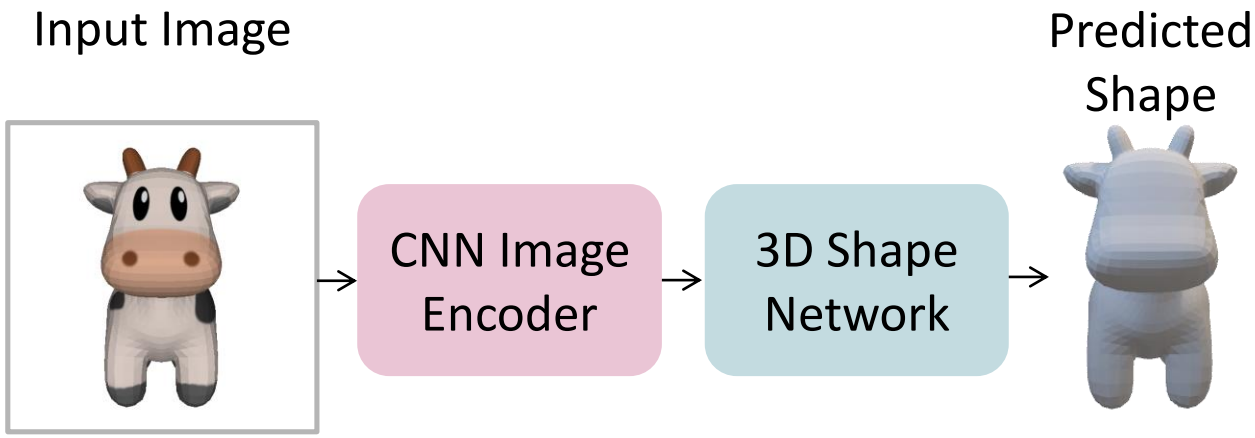


Solution: Supervised a second view during training

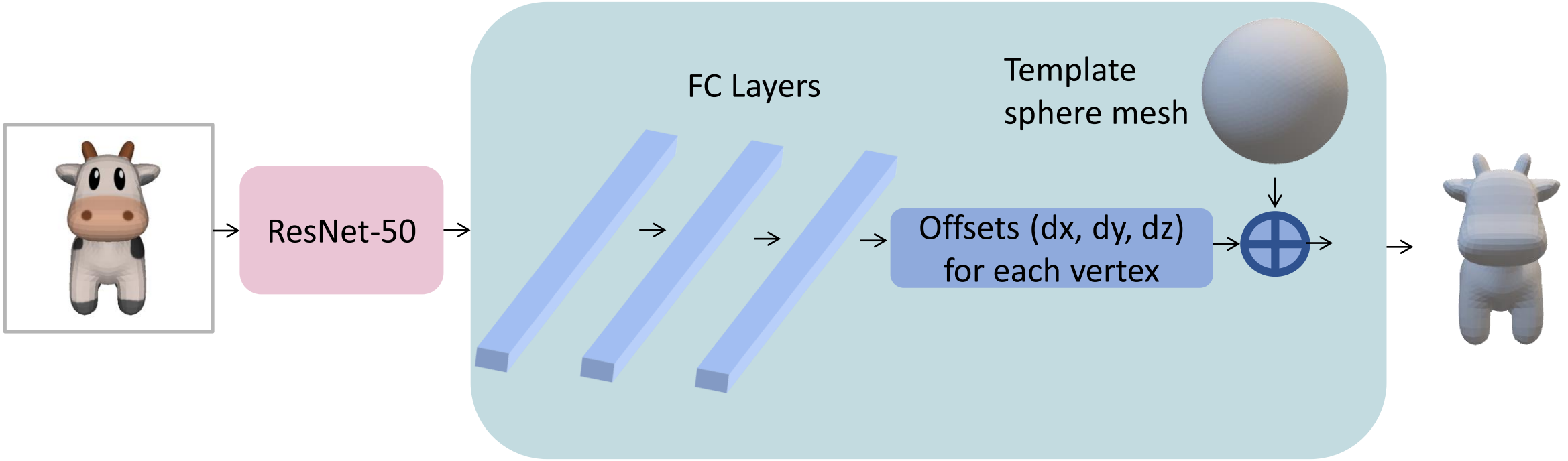
Unsupervised Shape Prediction



3D Shape Prediction Architecture

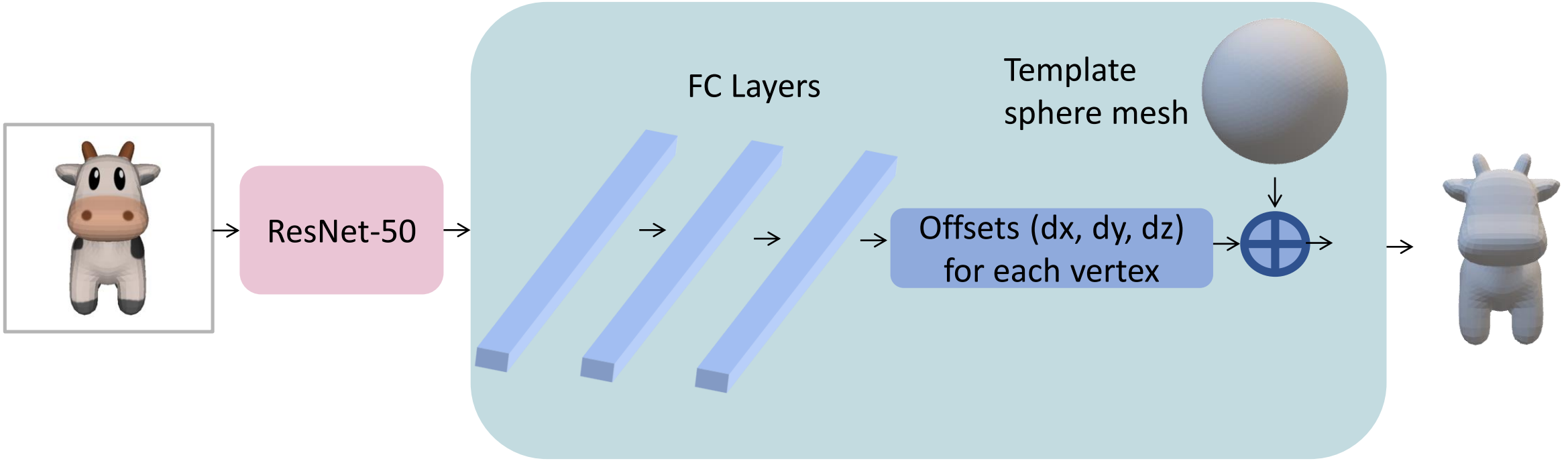


3D Shape Prediction Architecture



Common in prior work: Kato et al, Liu et al, Kanazawa et al., etc. etc.

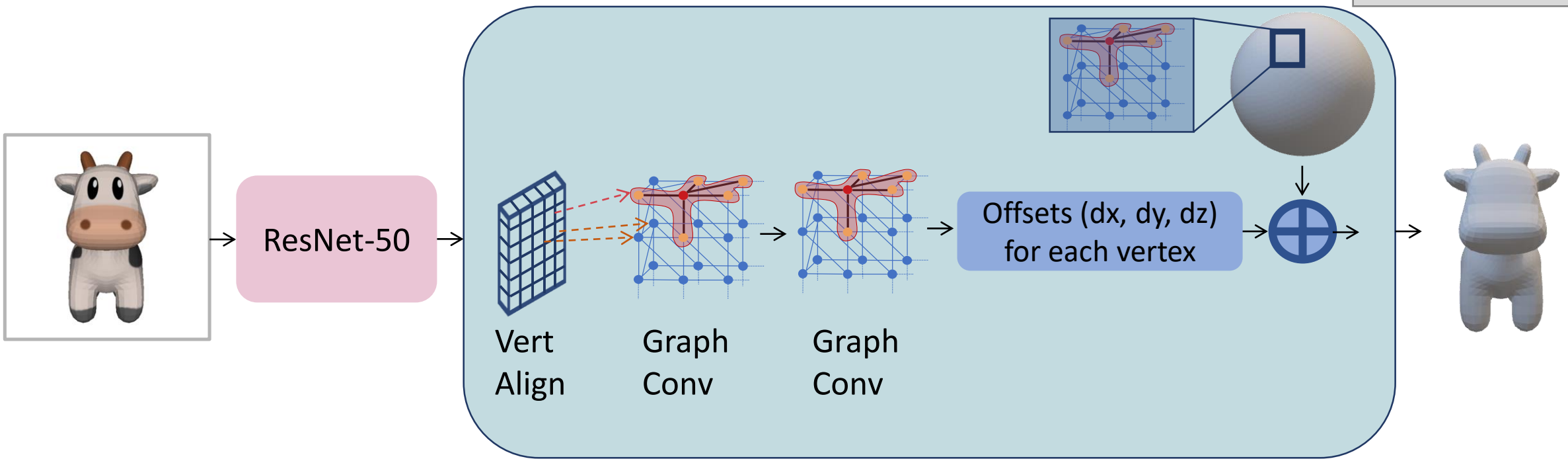
3D Shape Prediction Architecture



Common in prior work: Kato et al, Liu et al, Kanazawa et al., etc. etc.

Problem: FC layers don't respect object shape

3D Shape Prediction Architecture



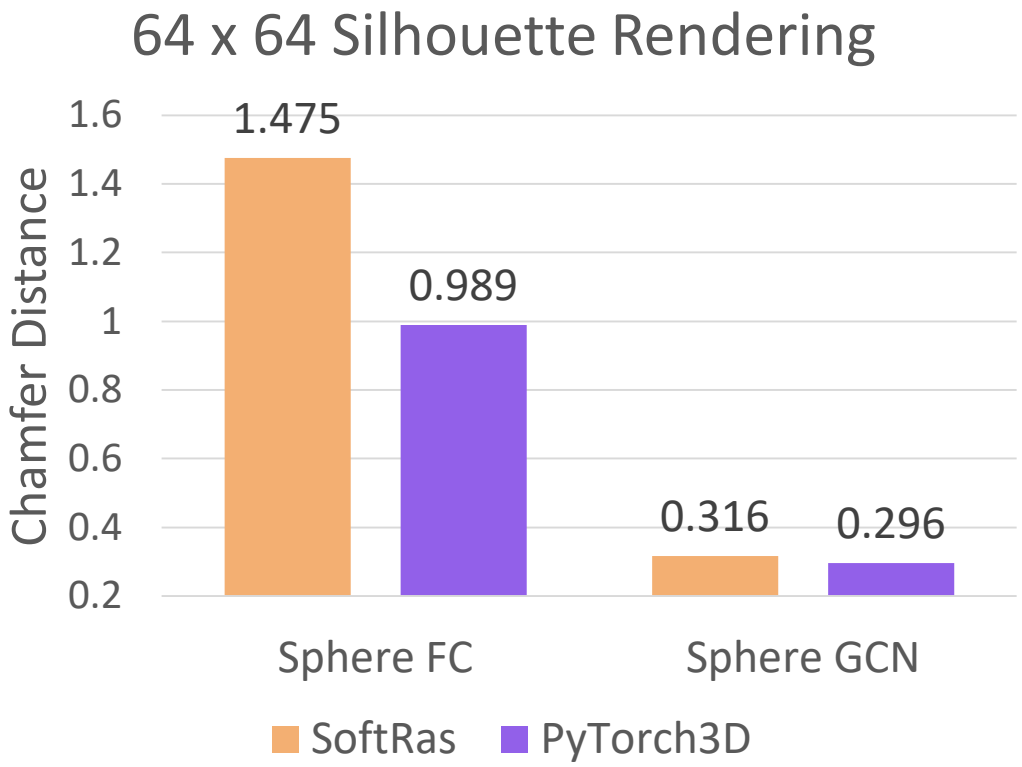
Like Mesh R-CNN, use Vert Align + Graph Conv to respect the mesh structure

Ravi et al, "Accelerating 3D Deep Learning with PyTorch3D", 2020

Experimental Results (ShapeNet)

Main Points:

- Sphere GCN > Sphere FC
- PyTorch3D >= SoftRas

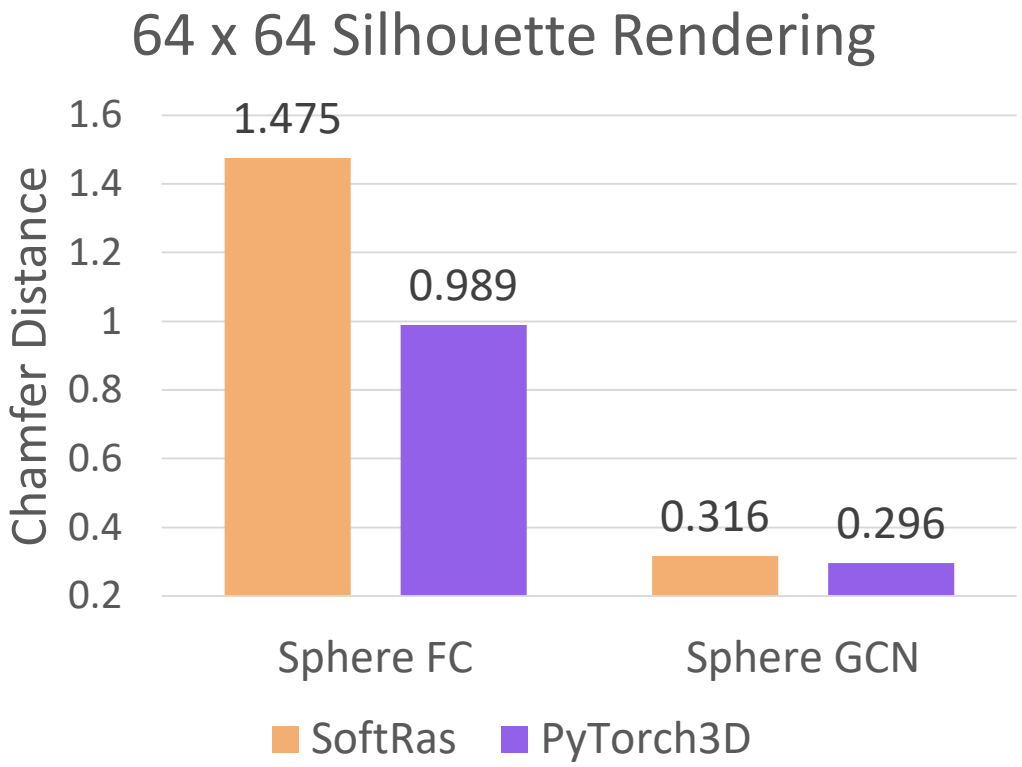


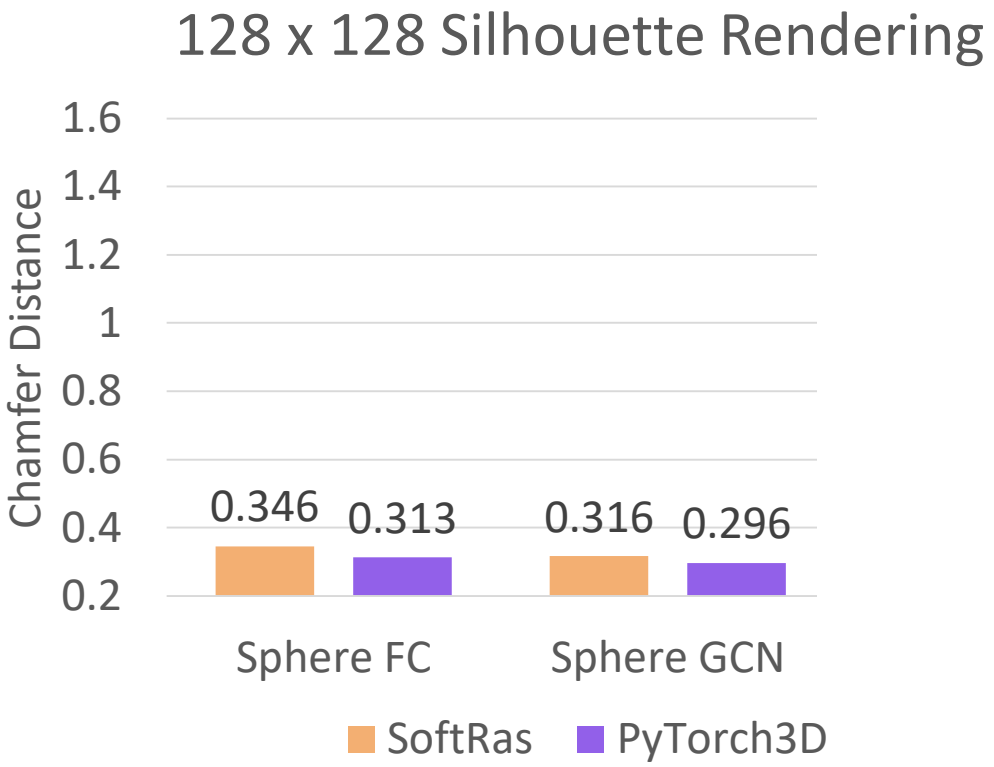
Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Experimental Results (ShapeNet)

Main Points:
- Sphere GCN > Sphere FC
- PyTorch3D >= SoftRas

Main Points:
- Larger images help



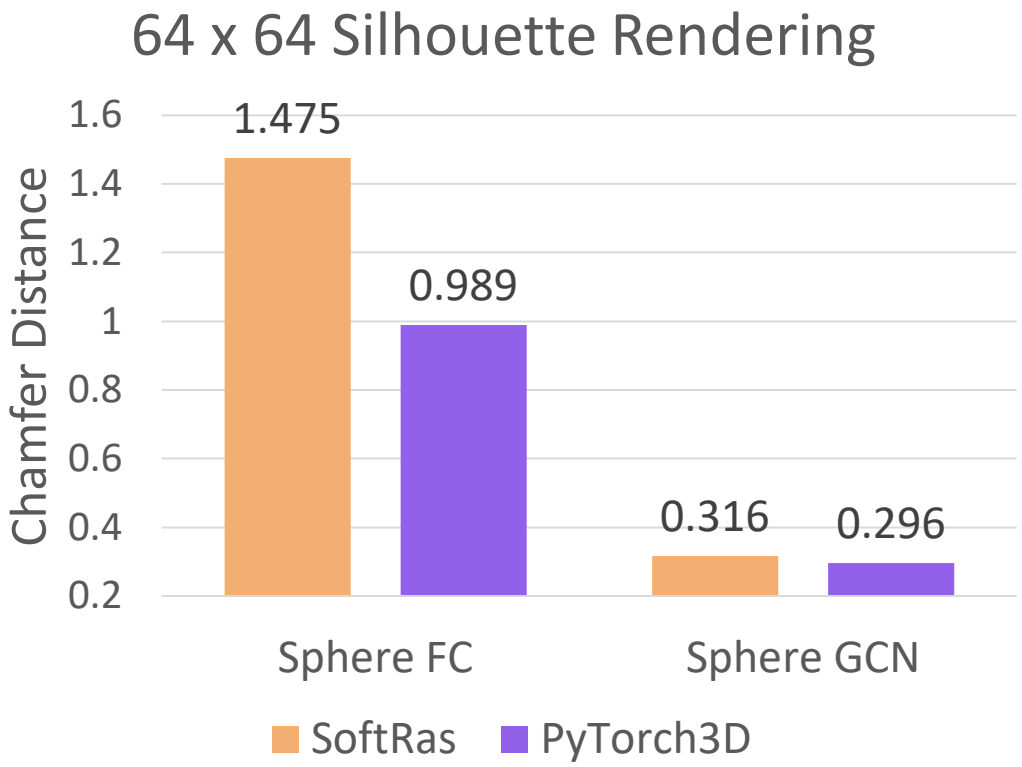


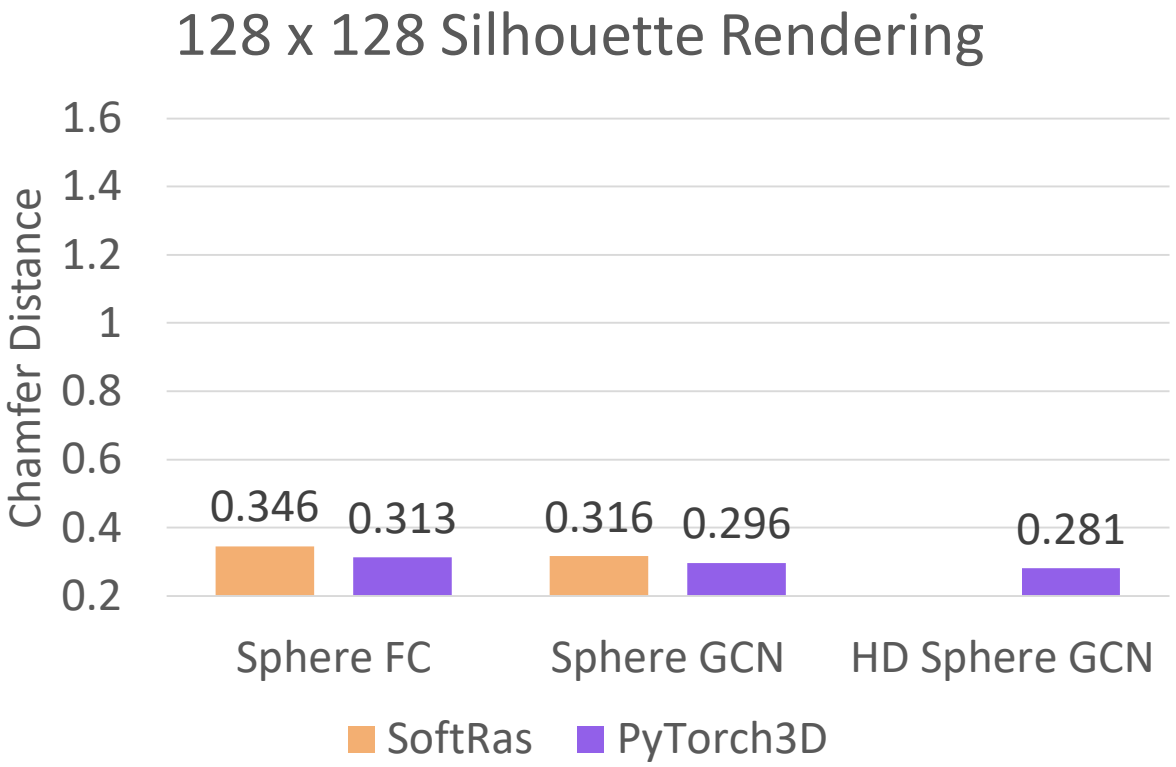
Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Experimental Results (ShapeNet)

Main Points:
- Sphere GCN > Sphere FC
- PyTorch3D >= SoftRas

Main Points:
- Larger images help
- Larger meshes help



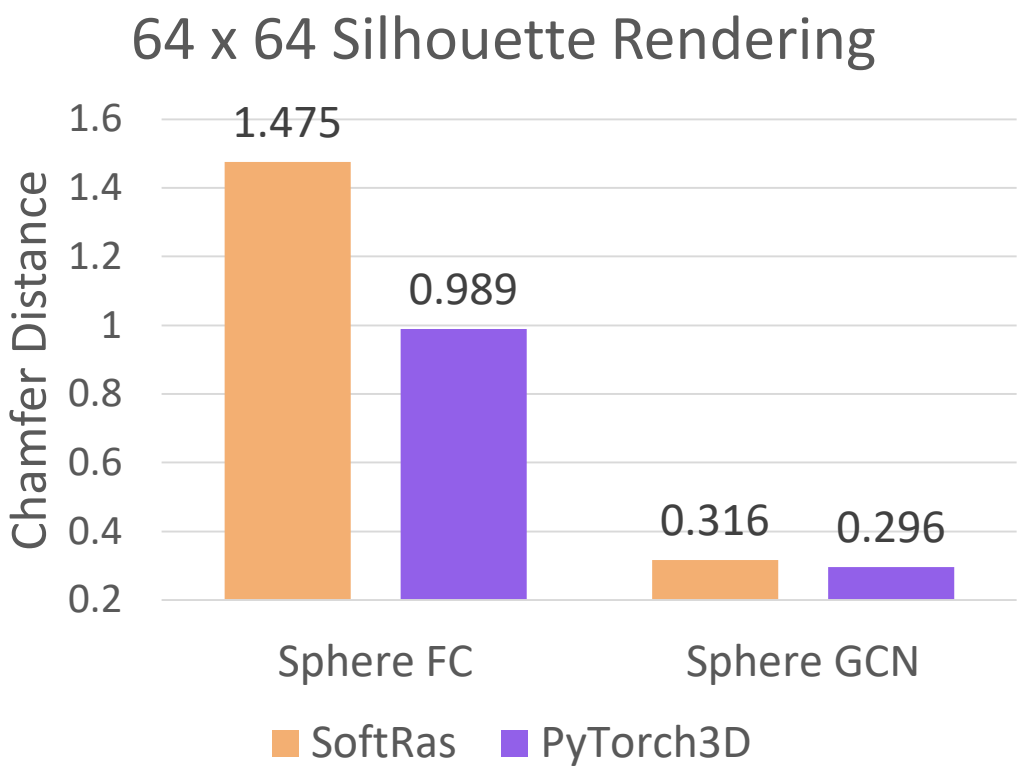


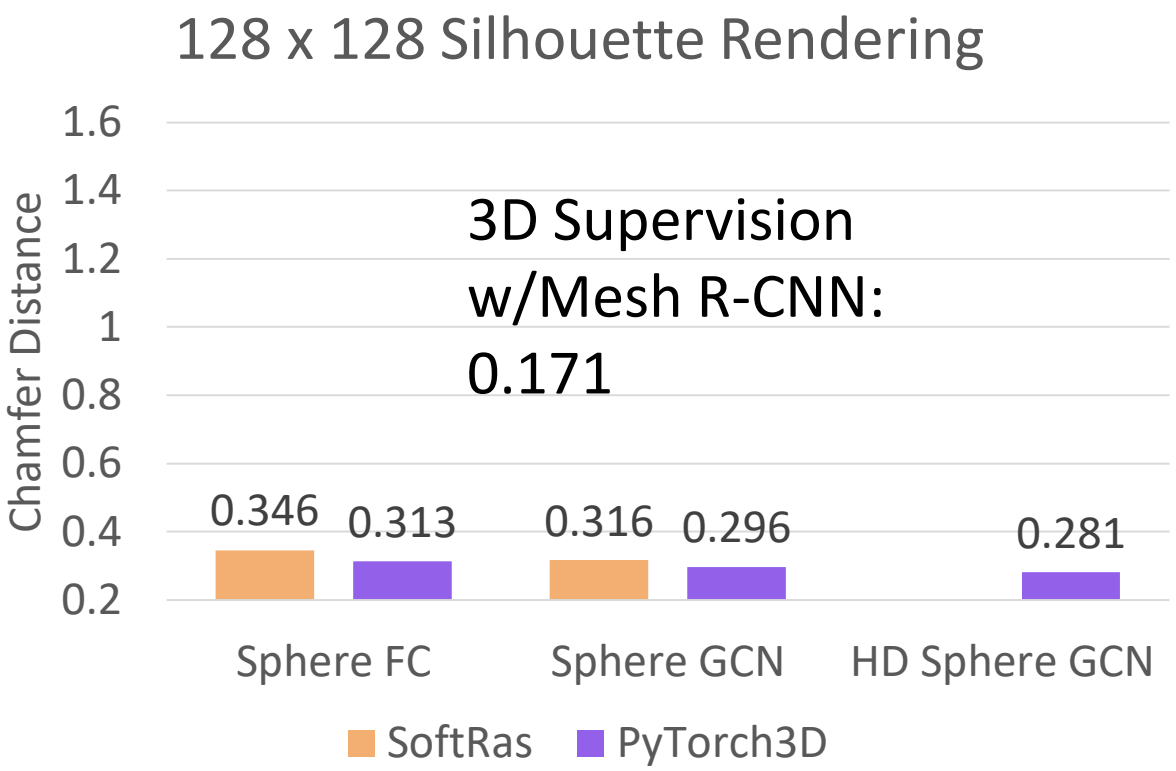
Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Experimental Results (ShapeNet)

Main Points:
- Sphere GCN > Sphere FC
- PyTorch3D >= SoftRas

Main Points:
- Larger images help
- Larger meshes help





Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Experimental Results

642 verts, 1250 faces

Input Image

Sphere FC

Sphere GCN

Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Experimental Results

642 verts, 1250 faces

2562 verts, 5120 faces

Input Image

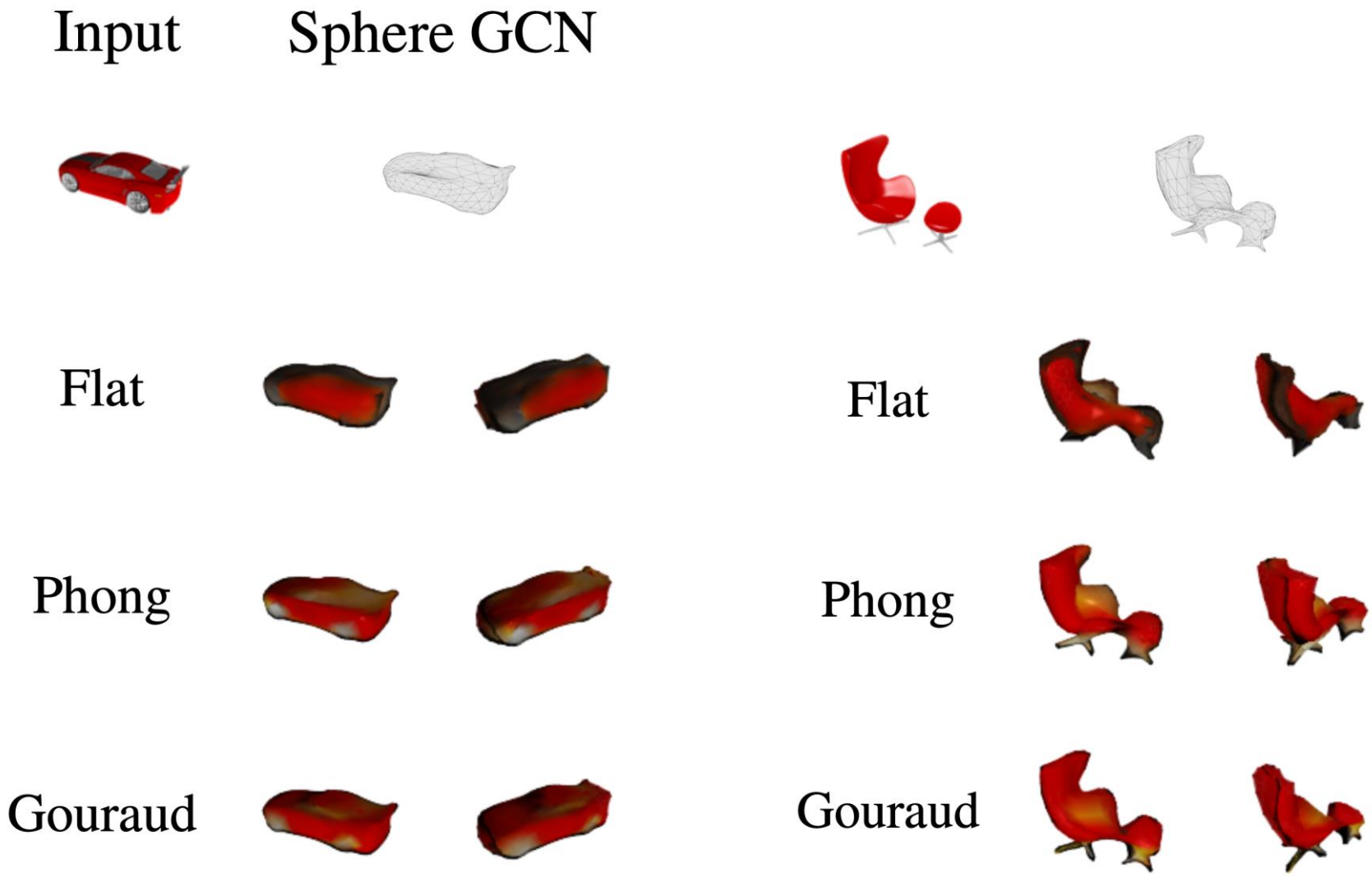
Sphere FC

Sphere GCN

HD Sphere GCN

Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Textured Mesh Rendering



Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

Unsupervised Point Cloud Prediction

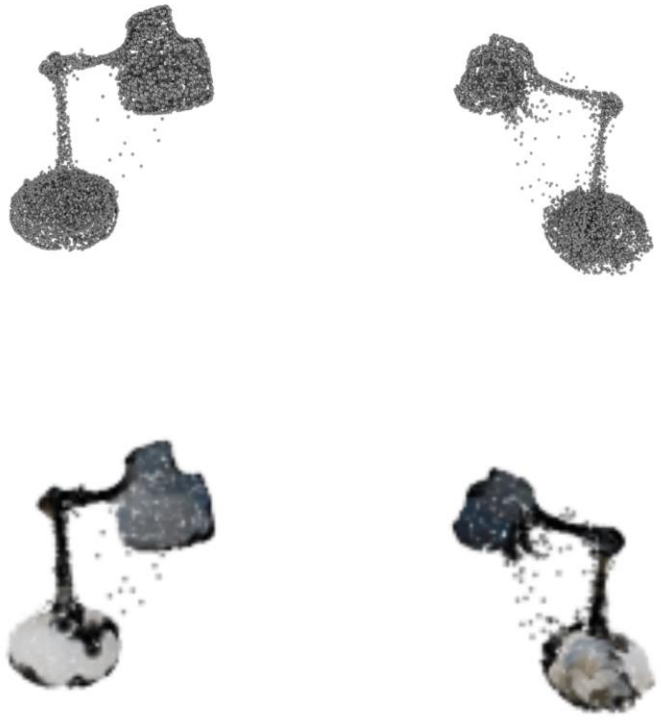
We can use a similar model to predict RGB Point Clouds from two-view supervision:

Input Image

Prediction

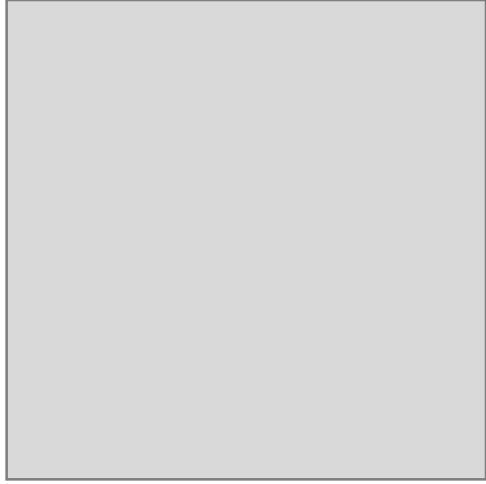
Input Image

Prediction



Ravi et al, “Accelerating 3D Deep Learning with PyTorch3D”, 2020

3D Predictions from 2D Supervision



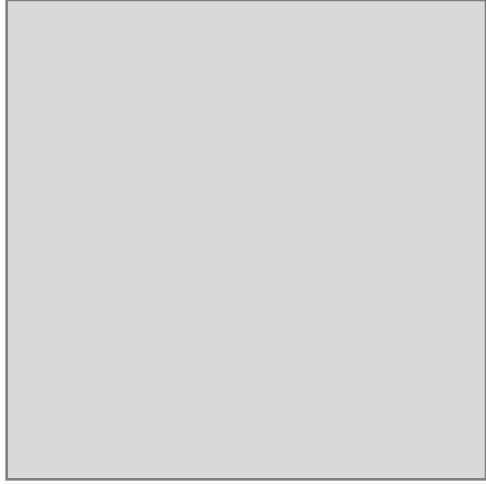
Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

3D Predictions from 2D Supervision



Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

SynSin: End-to-End View **Syn**thesis from a **Sin**gle Image

CVPR 2020

Olivia Wiles
Oxford

Georgia Gkioxari
FAIR

Rick Szeliski
Facebook

Justin Johnson
UMich / FAIR

Single Image View Synthesis

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Single Image View Synthesis

Translation: Step forward

Rotation: Turn left

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

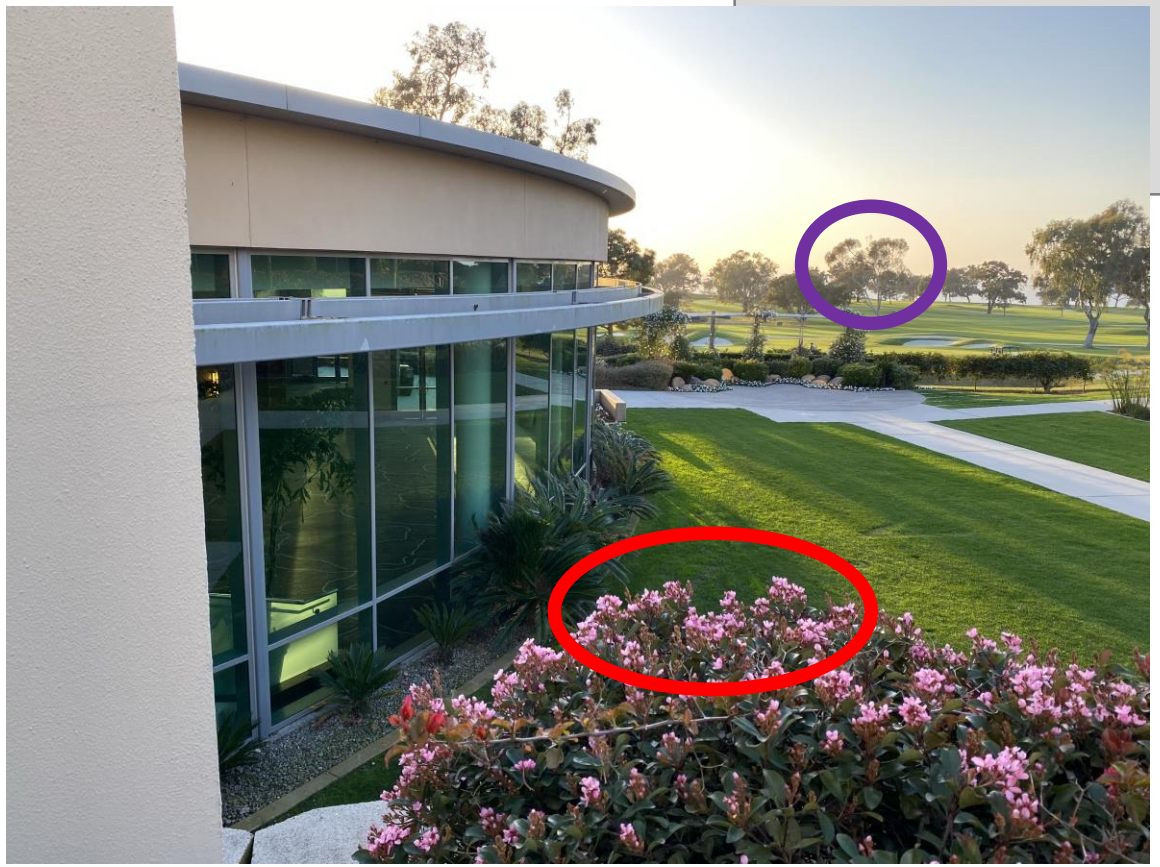
Single Image View Synthesis

Translation: Step forward

Rotation: Turn left

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

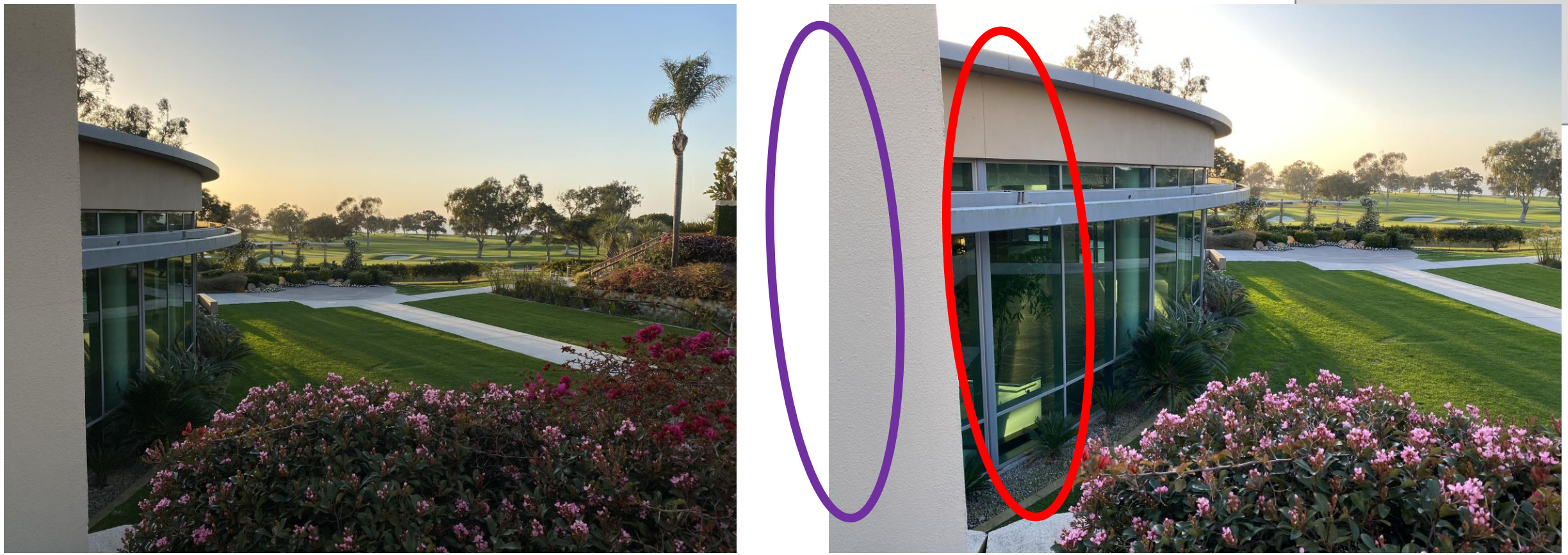
Single Image View Synthesis



Challenge: Need to know depth

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Single Image View Synthesis



Challenge: Inpainting Missing Regions

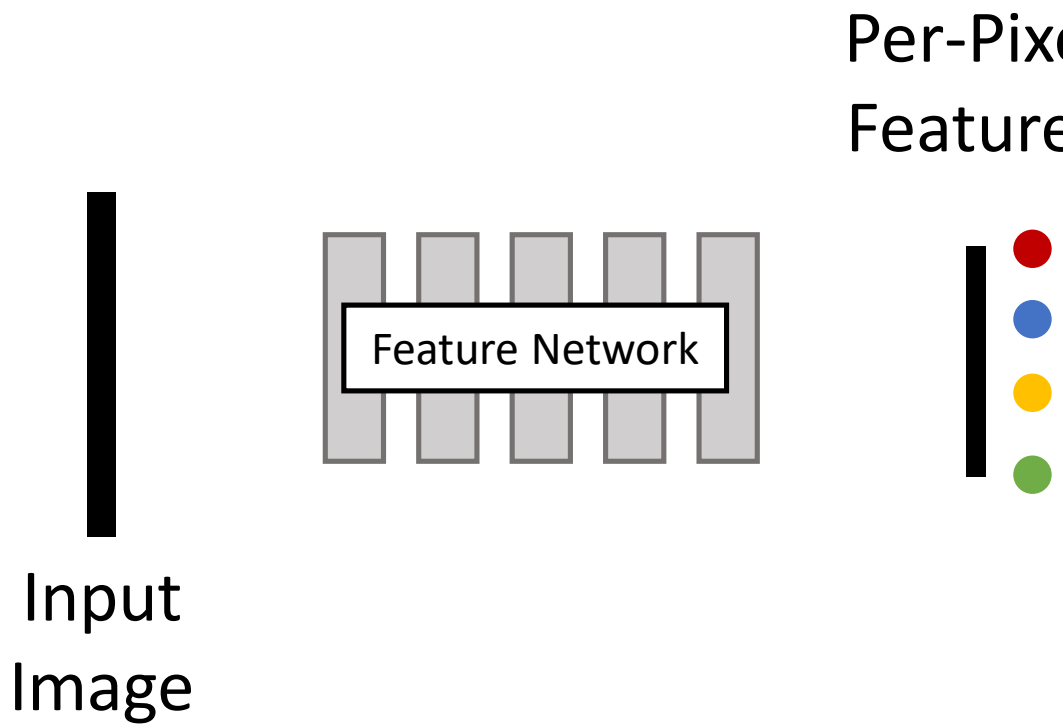
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Single Image View Synthesis: Goals

1. Complex, realistic scenes
2. Train w/o GT Depth;
   only (Image, RT, Image)
3. Test with a single image
   (Image, RT) -> Image
4. End-To-End training

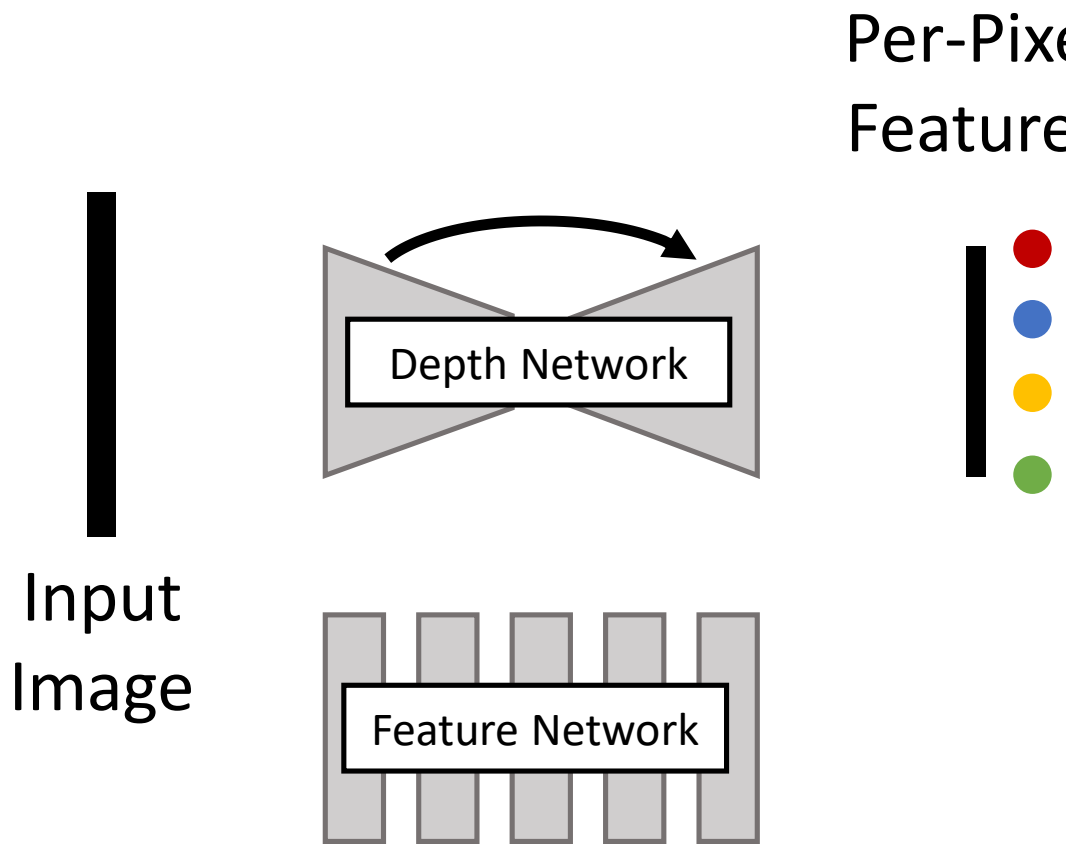
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



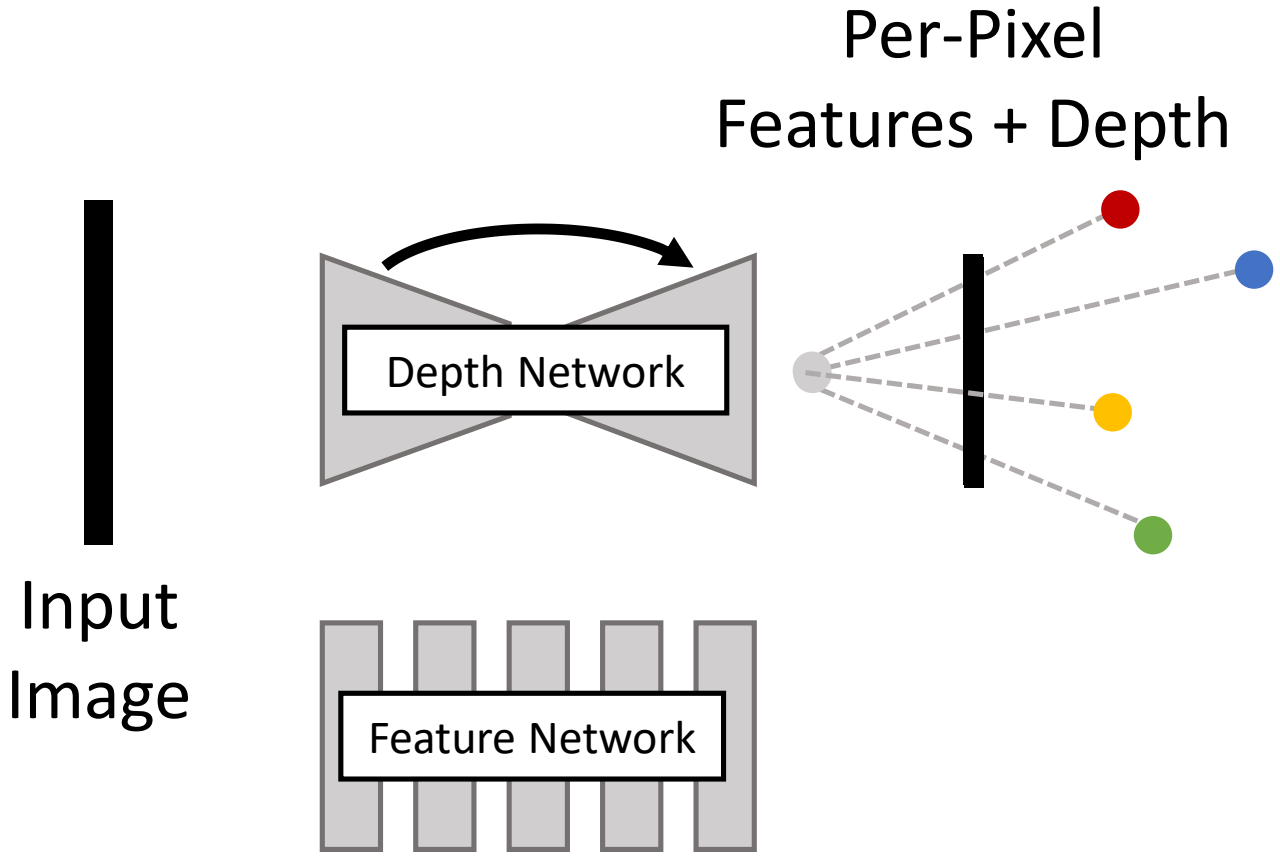
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



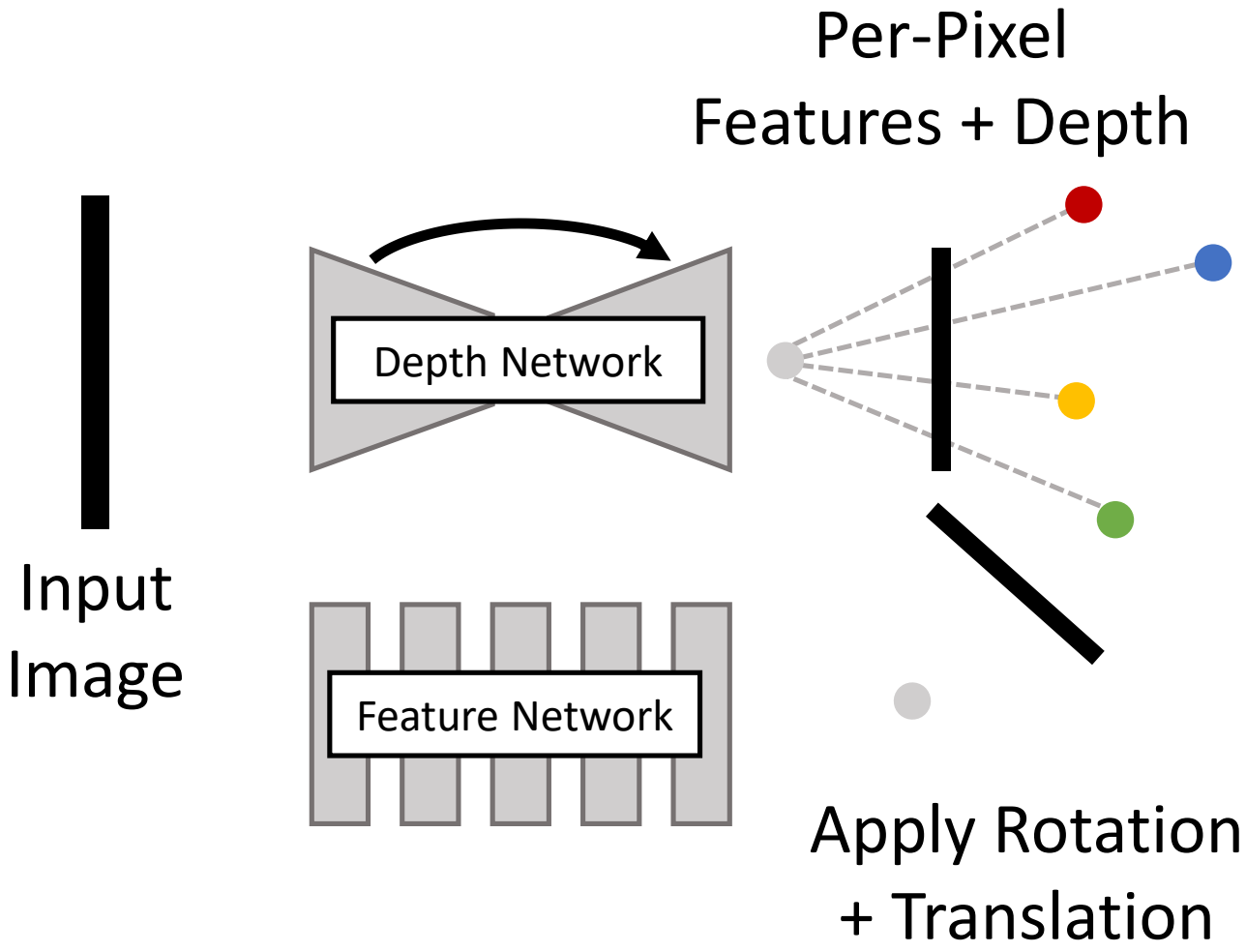
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



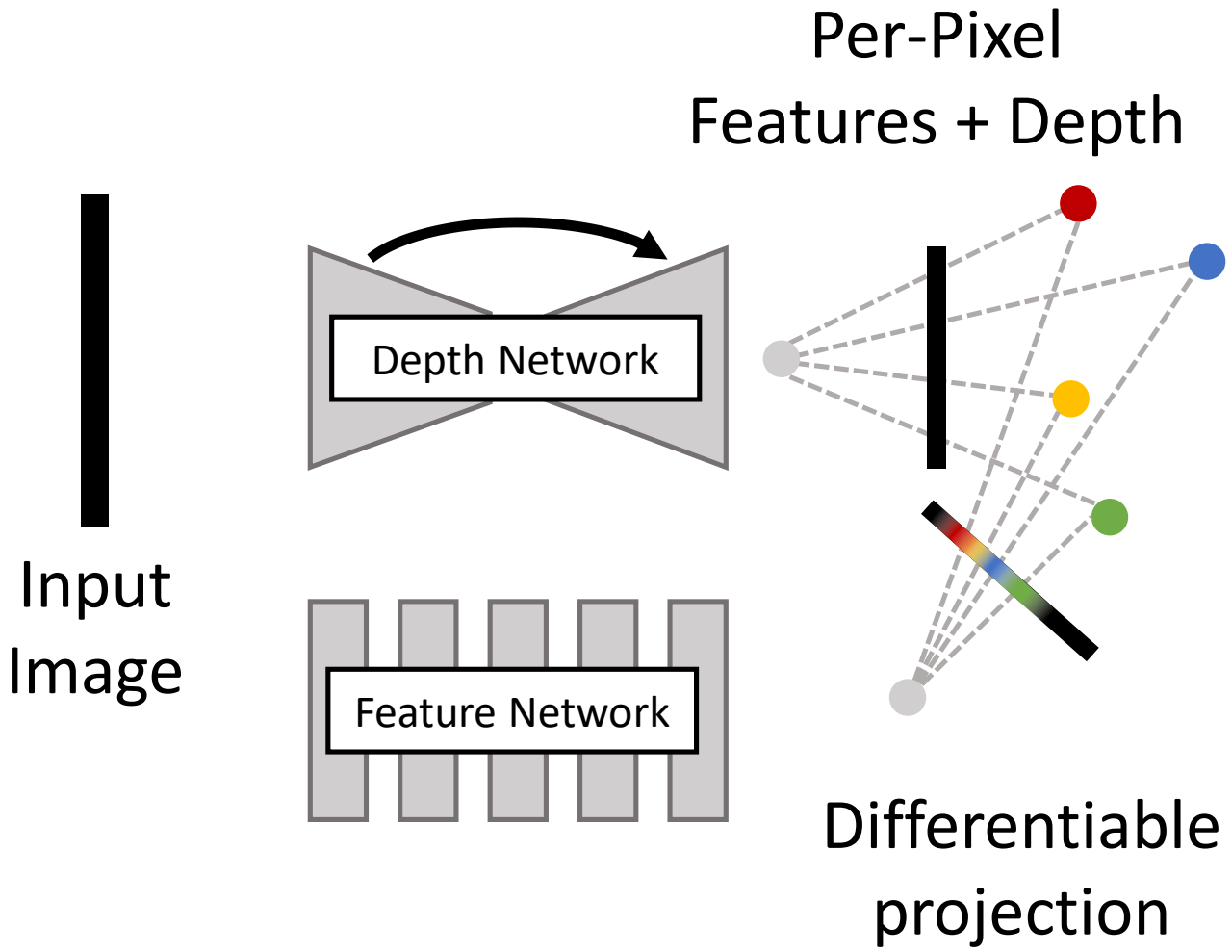
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



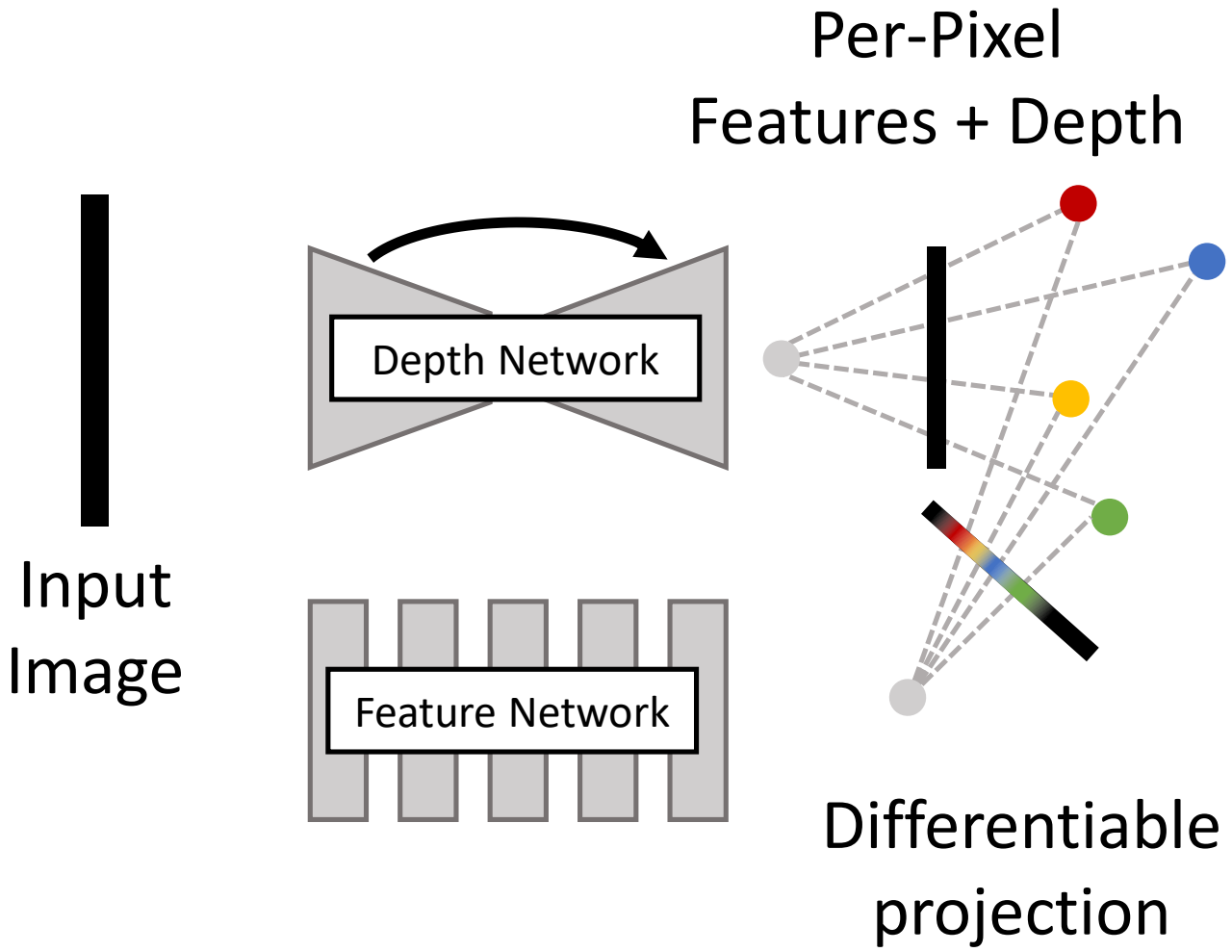
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



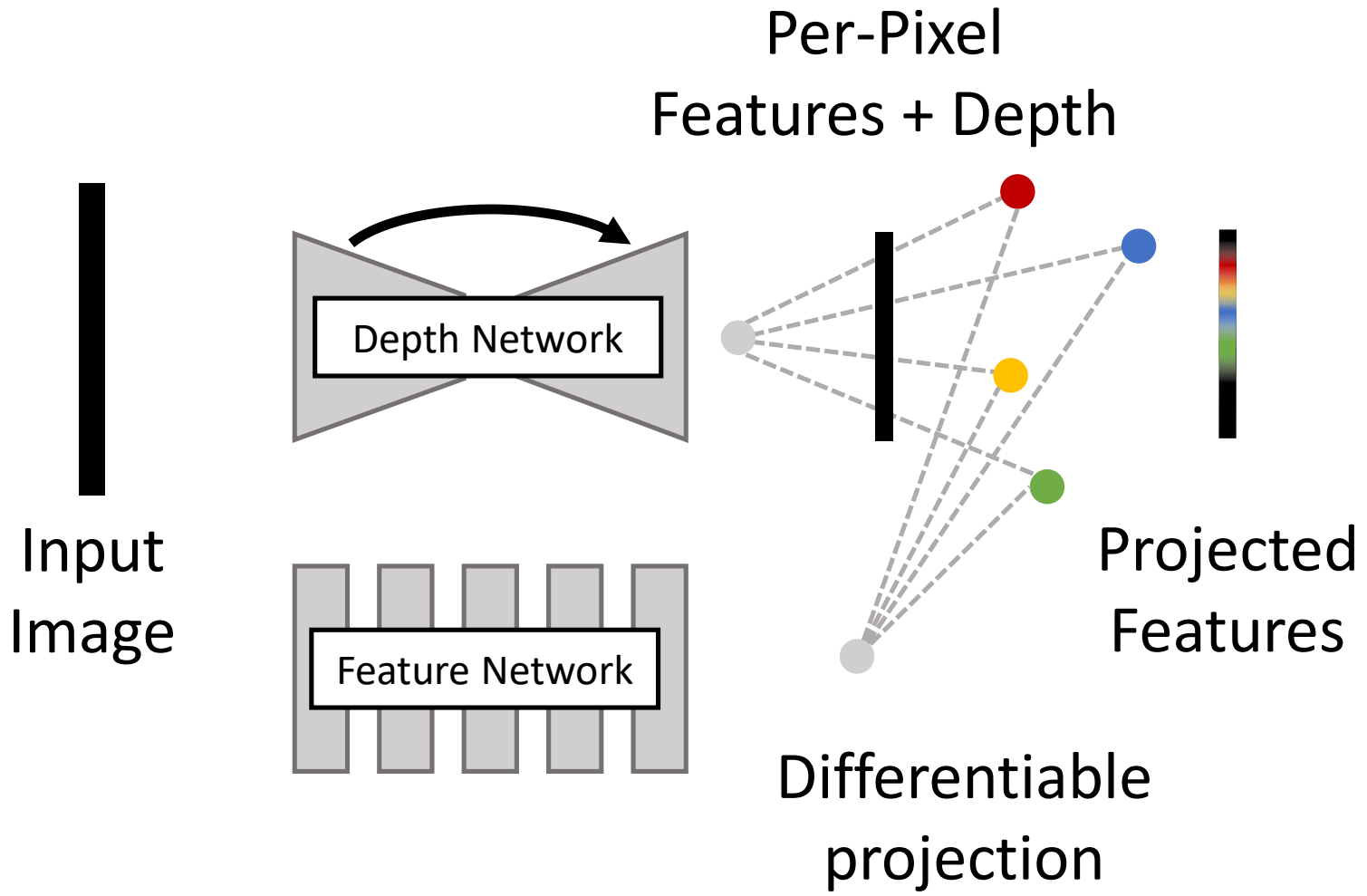
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



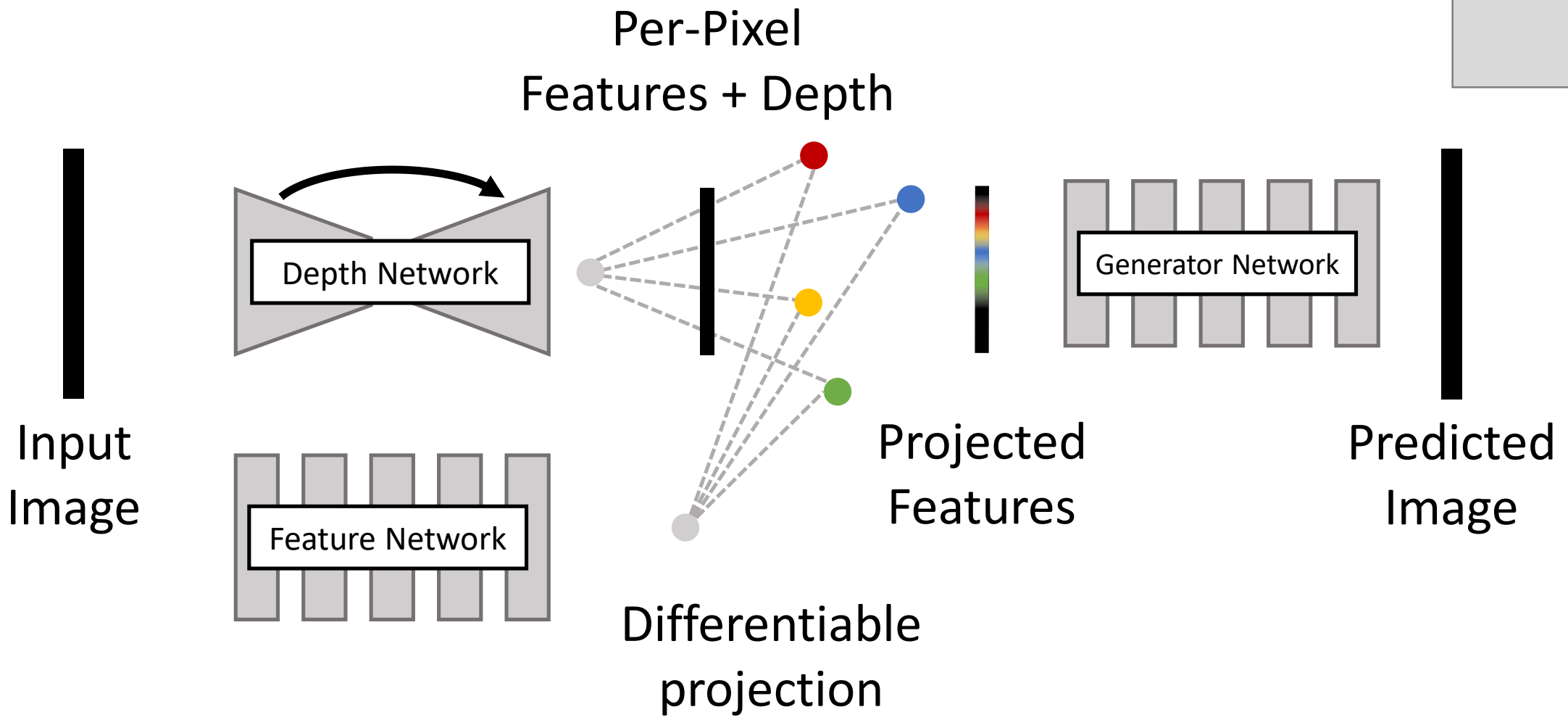
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



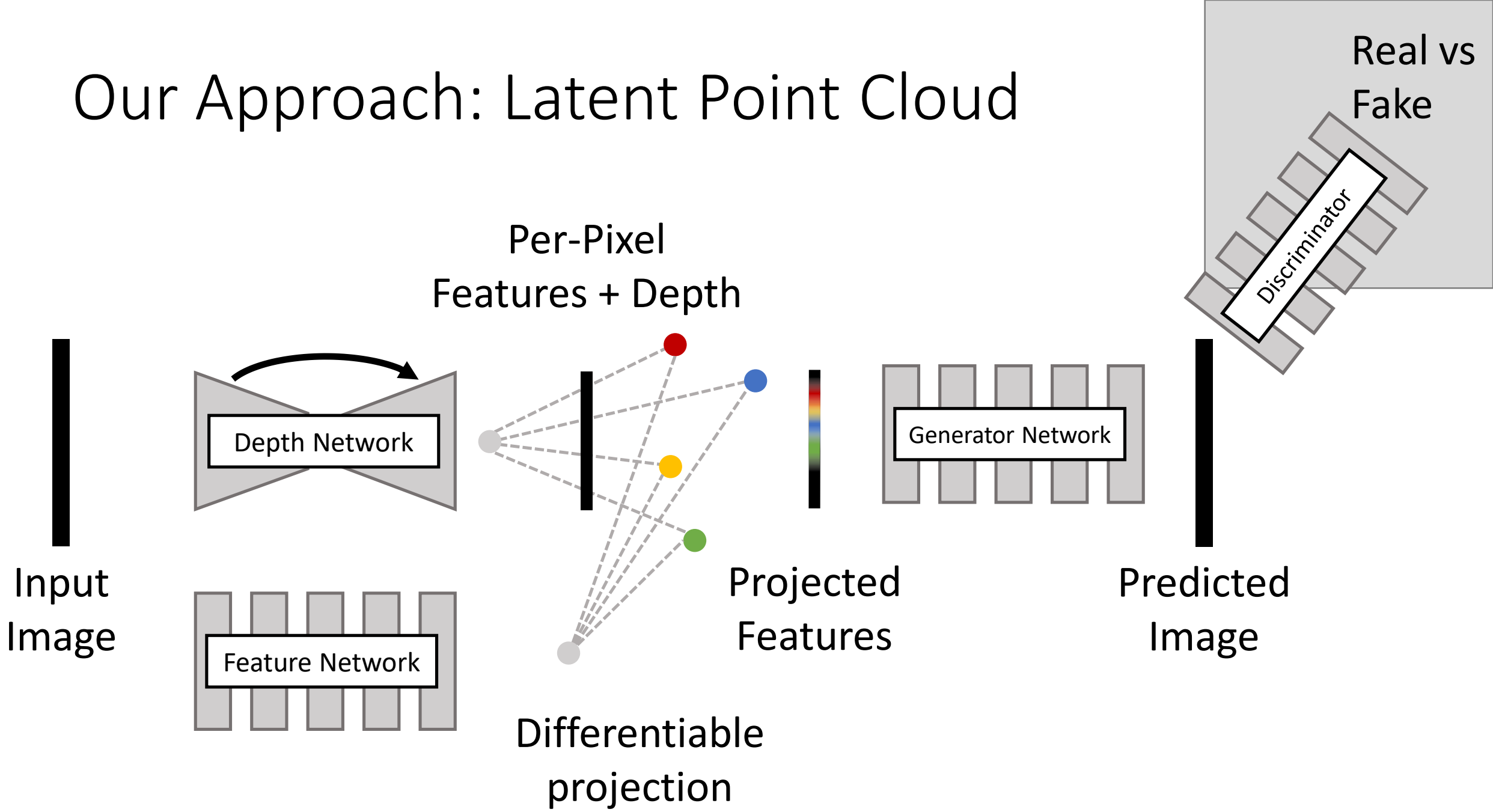
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



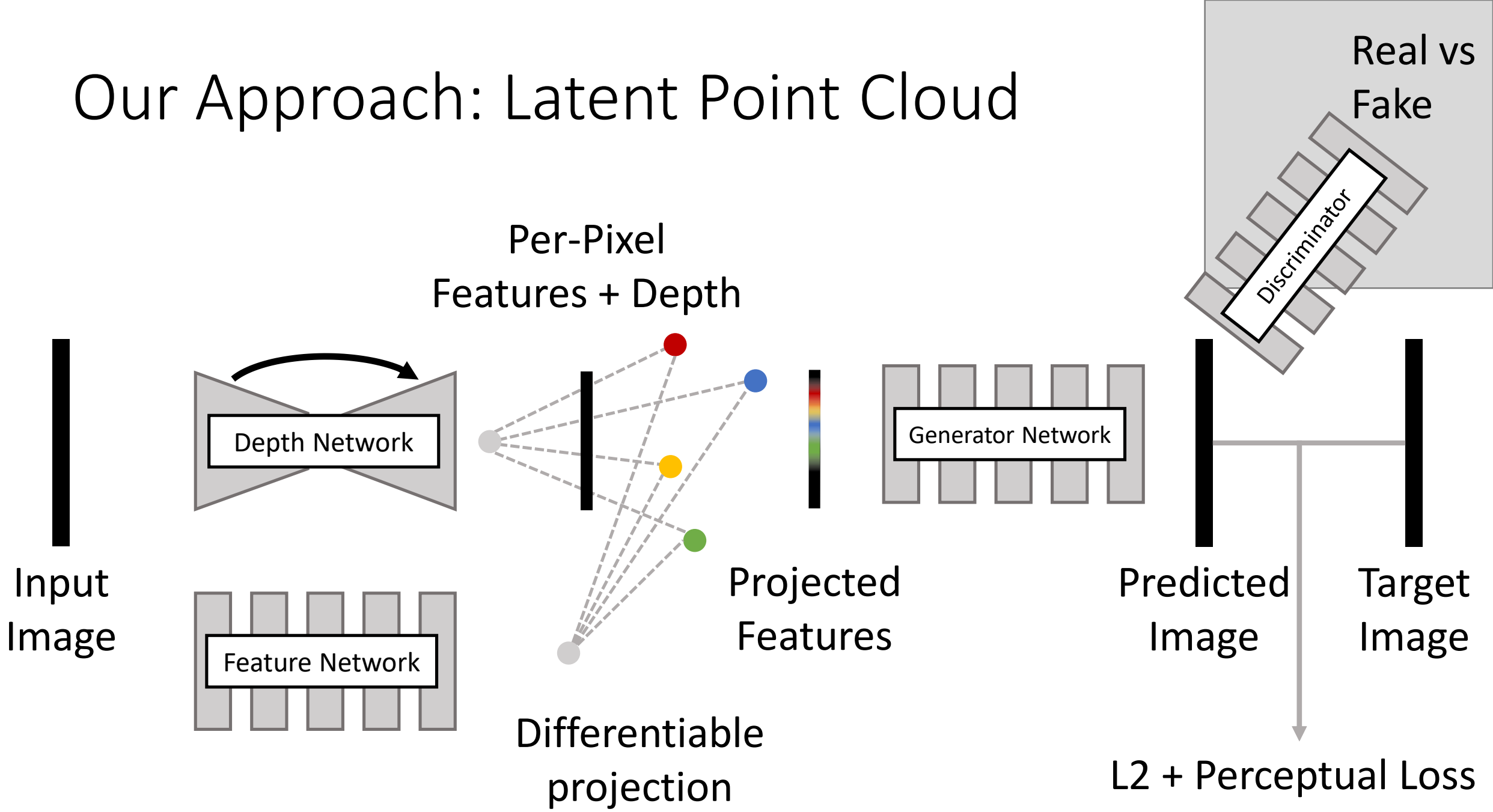
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud



Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Our Approach: Latent Point Cloud

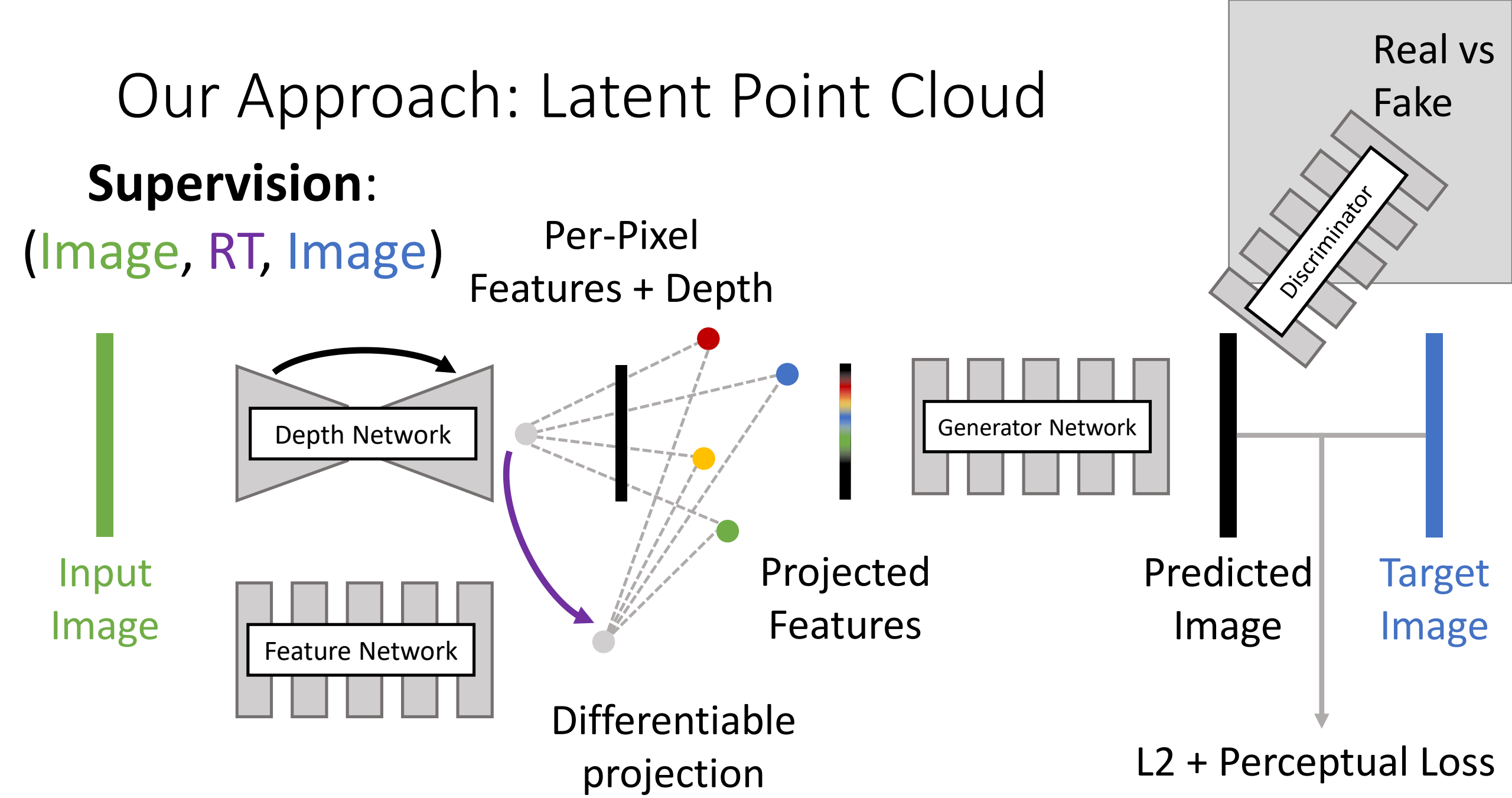


Wiles et al, "SynSin: End-to-End View Synthesis from a Single Image", CVPR 2020

Our Approach: Latent Point Cloud

Supervision:

(Image, RT, Image)

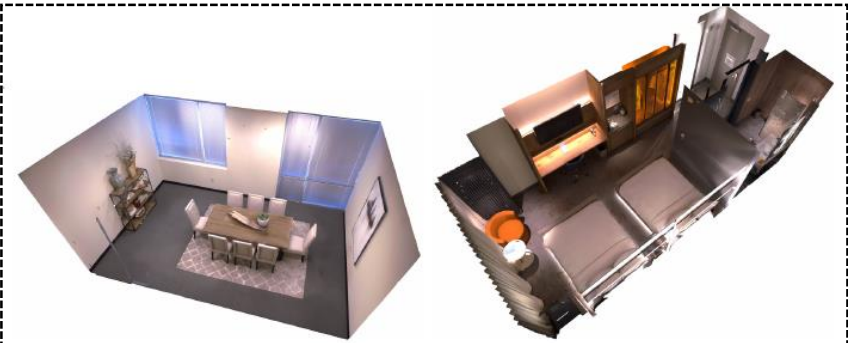


Wiles et al, "SynSin: End-to-End View Synthesis from a Single Image", CVPR 2020

Complex Datasets

Room-Scale 3D Scans

Matterport3D



Replica

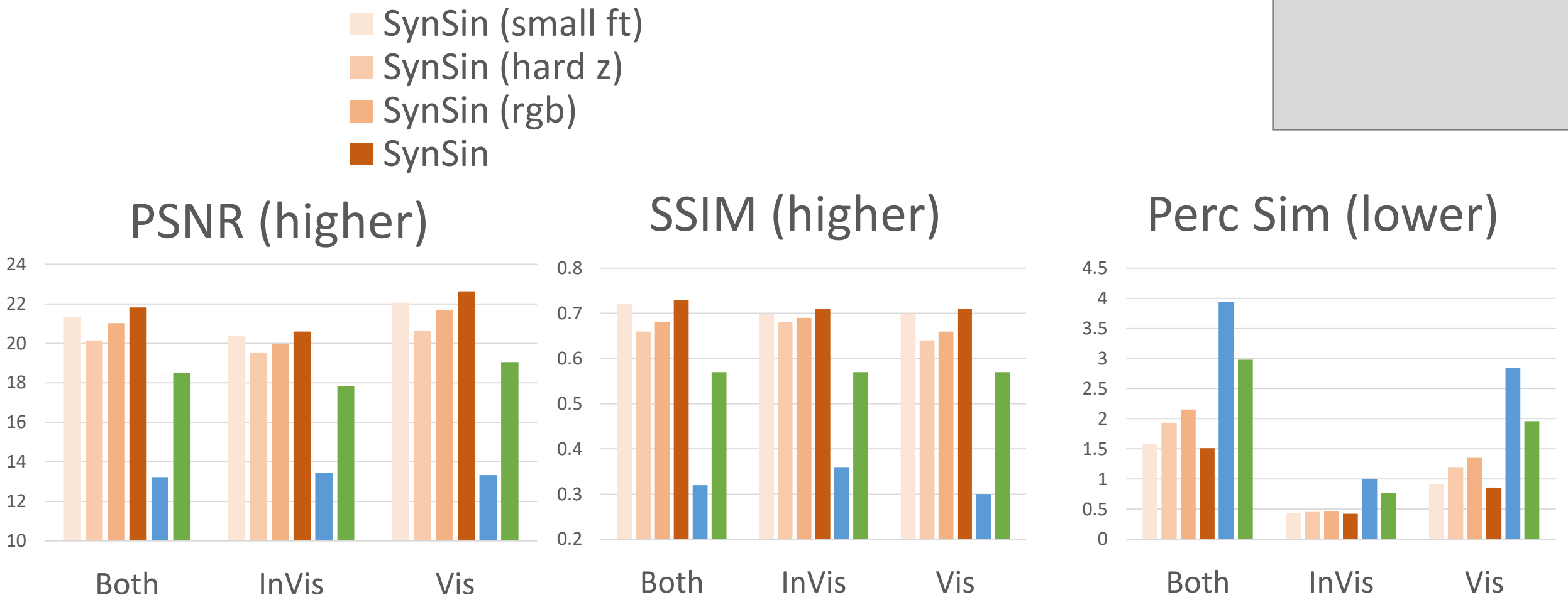
Real-World Video with camera pose

RealEstate10K

RealEstate10K: Zhou et al, "Stereo magnification: Learning view synthesis using multiplane images," SIGGRAPH 2018.
Matterport3D: Chang et al. "Matterport3d: Learning from RGB-D data in indoor environments." 3DV 2017
Replica: Straub et al. "The Replica Dataset: A Digital Replica of Indoor Spaces," arXiv 2019
Habitat: Savva et al, "Habitat: A Platform for Embodied AI Research," ICCV 2019

Wiles et al, "SynSin: End-to-End View Synthesis from a Single Image", CVPR 2020

Ablations: MatterPort3D



Wiles et al, "SynSin: End-to-End View Synthesis from a Single Image", CVPR 2020

Im2Im: Zhou, Tinghui, et al. "View synthesis by appearance flow." ECCV 2016
Vox w/ Unet: Sitzmann, Vincent, et al. "DeepVoxels: Learning persistent 3D feature embeddings." CVPR 2019

Qualitative Results

Input Image

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Qualitative Results

Input Image

Ground Truth

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Qualitative Results

Input Image

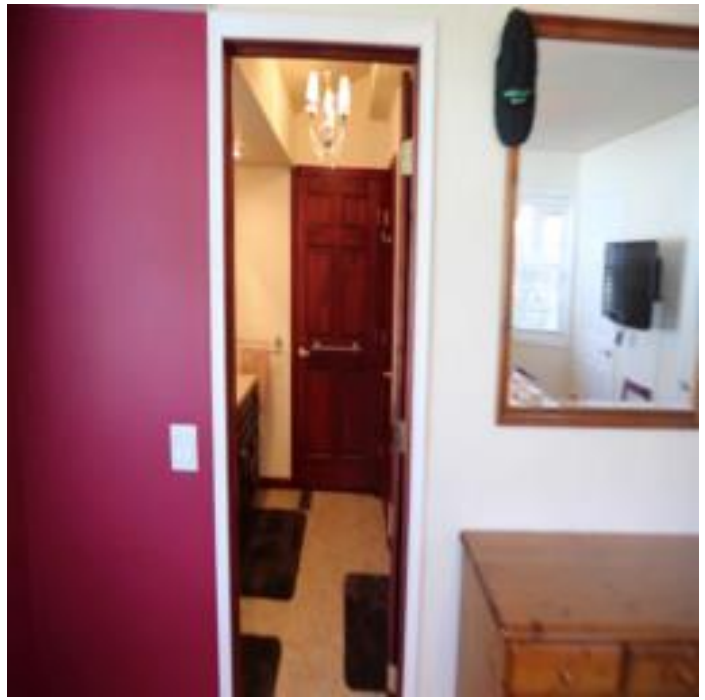
Ground Truth

Ours

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

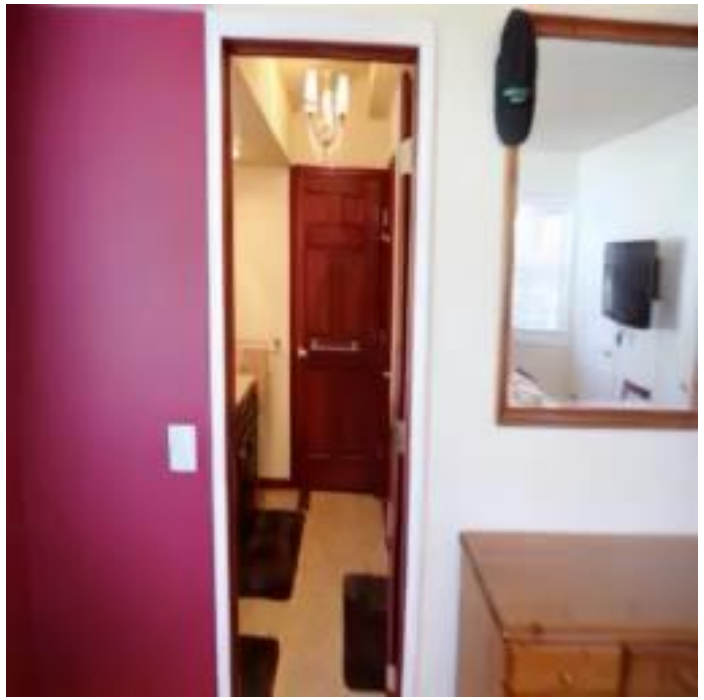
Qualitative Results

Input Image



Ground Truth

Ours



Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

Qualitative Results

Input Image

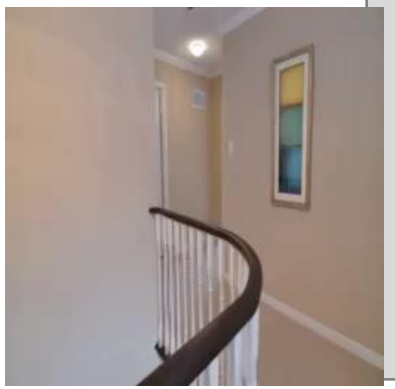
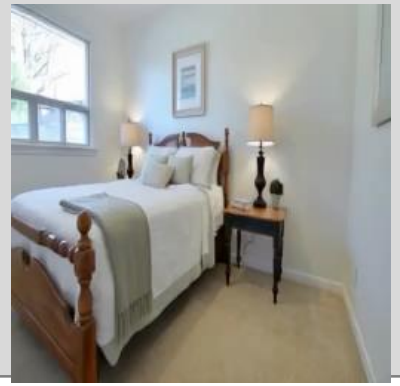
Ground Truth

Ours

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

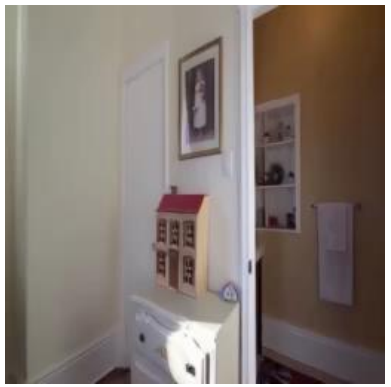
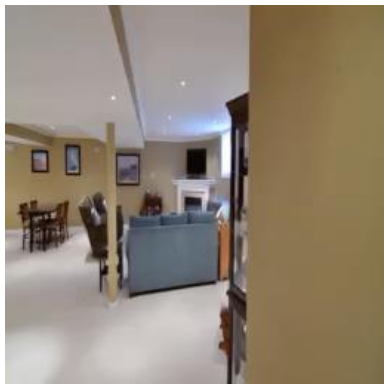






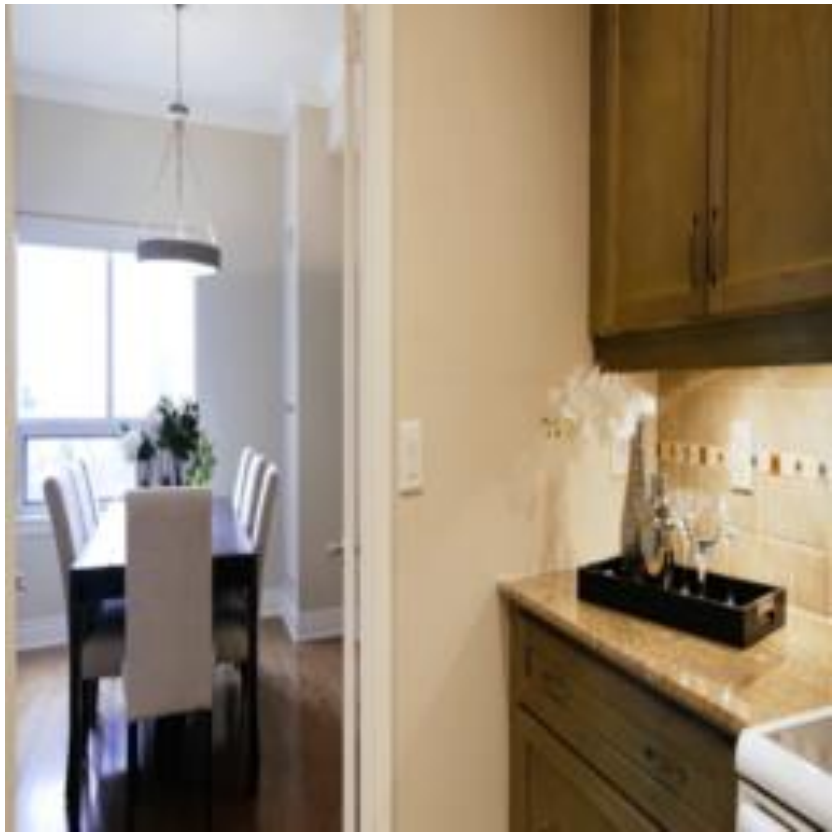


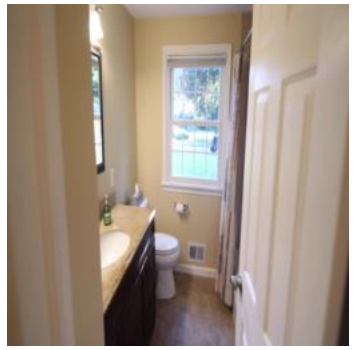


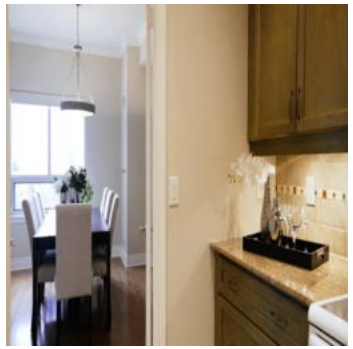

Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020

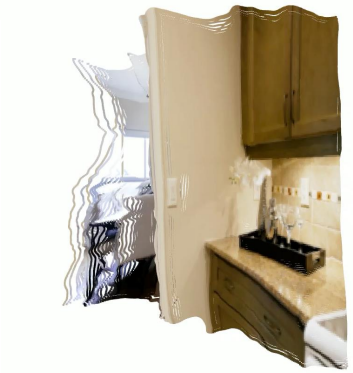
Predicted Point Clouds

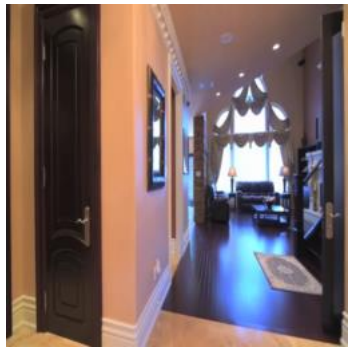


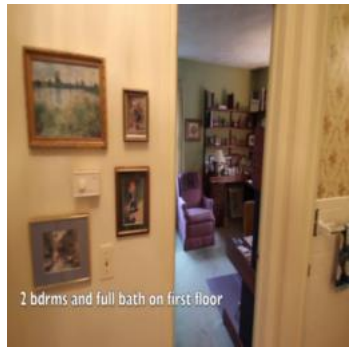
Wiles et al, ”SynSin: End-to-End View Synthesis from a Single Image”, CVPR 2020











Input Image

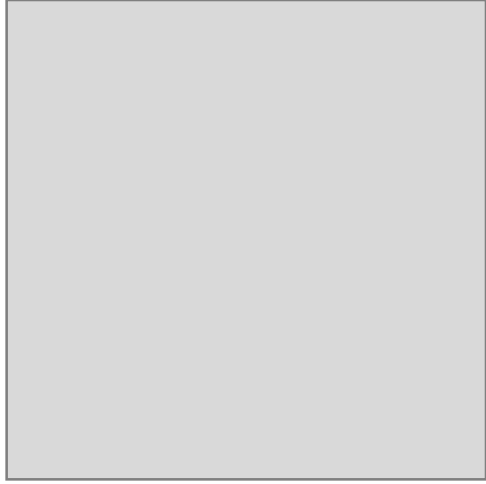
SynSin's predicted 3D point cloud

Input Image

SynSin's predicted 3D point cloud

Wiles et al, "SynSin: End-to-End View Synthesis from a Single Image", CVPR 2020

3D Predictions from 2D Supervision



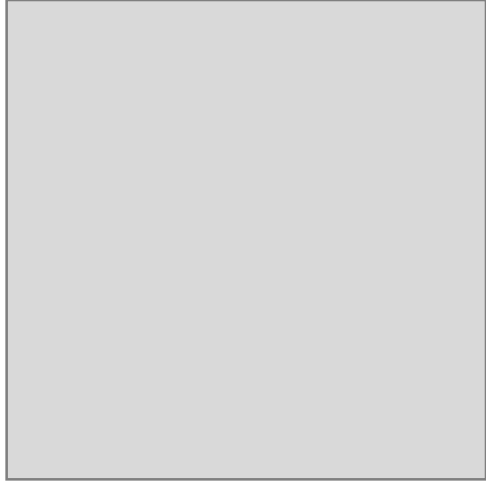
Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

3D Predictions from 2D Supervision

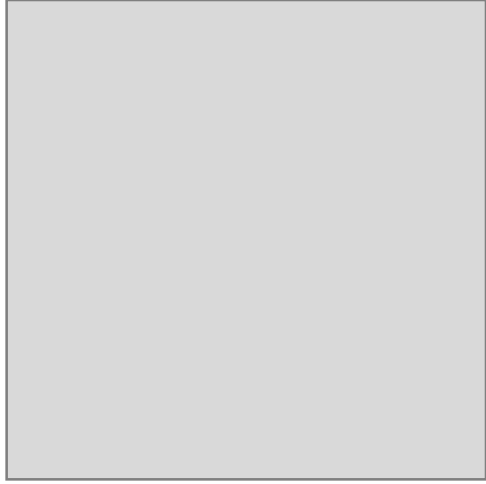


Task: Supervised Shape Prediction
Gkioxari et al, ICCV 2019

Tool: Differentiable Rendering + PyTorch3D
Ravi et al, 2020

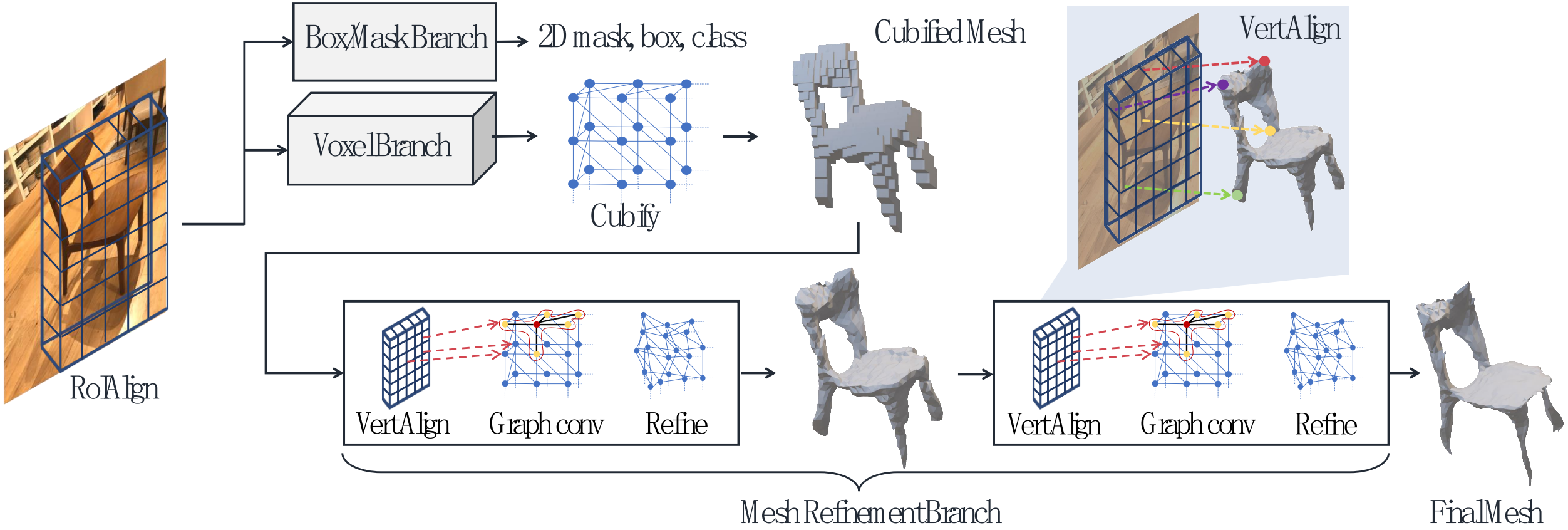
Task: Unsupervised Shape Prediction
Ravi et al, 2020

Task: Single-Image View Synthesis
Wiles et al, CVPR 2020

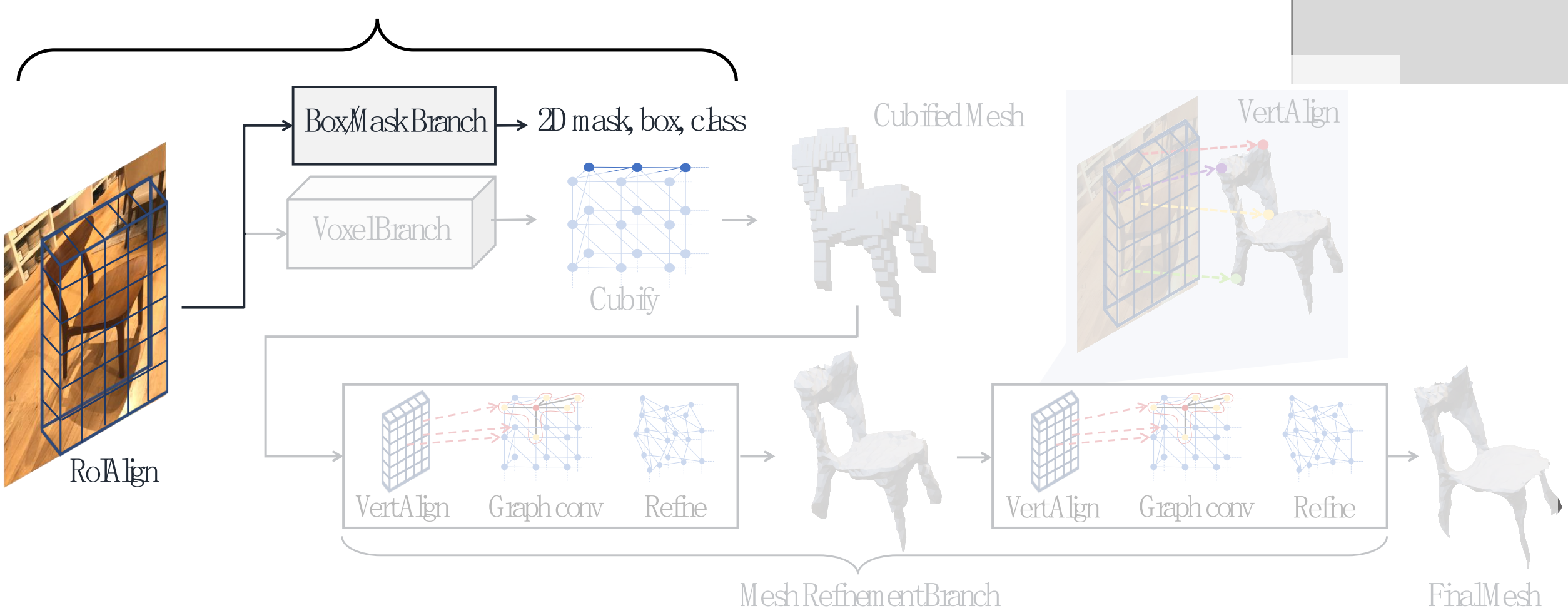


Thank You!

Mesh R-CNN: Architecture

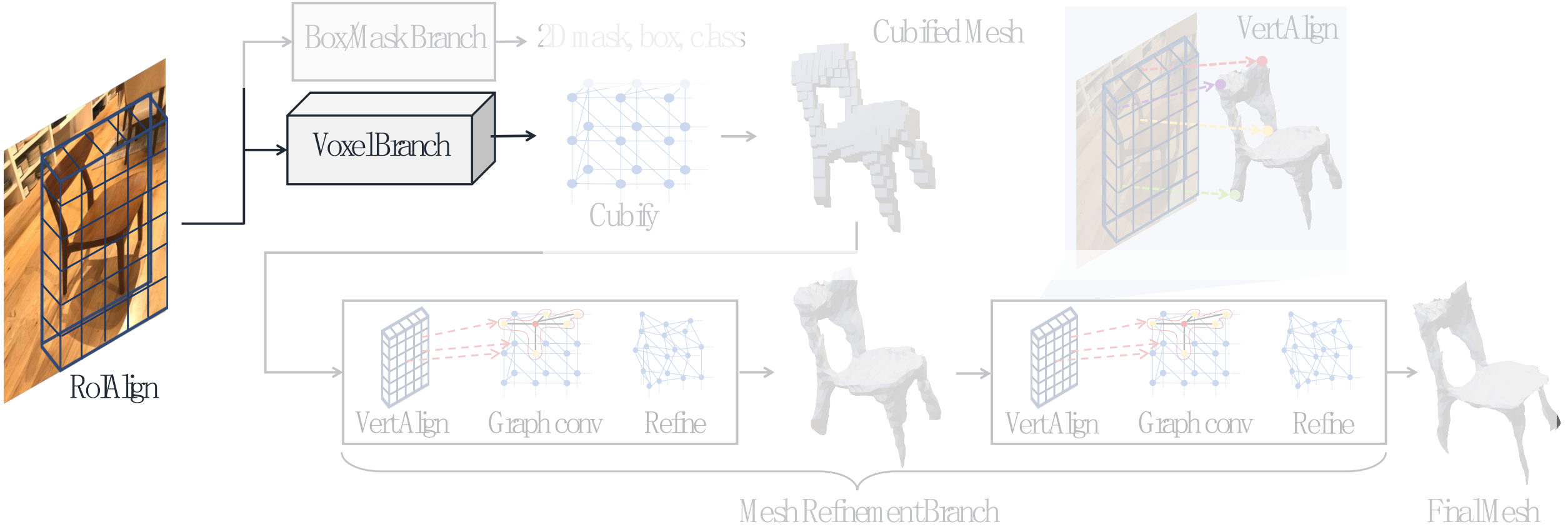


Mesh R-CNN: Architecture

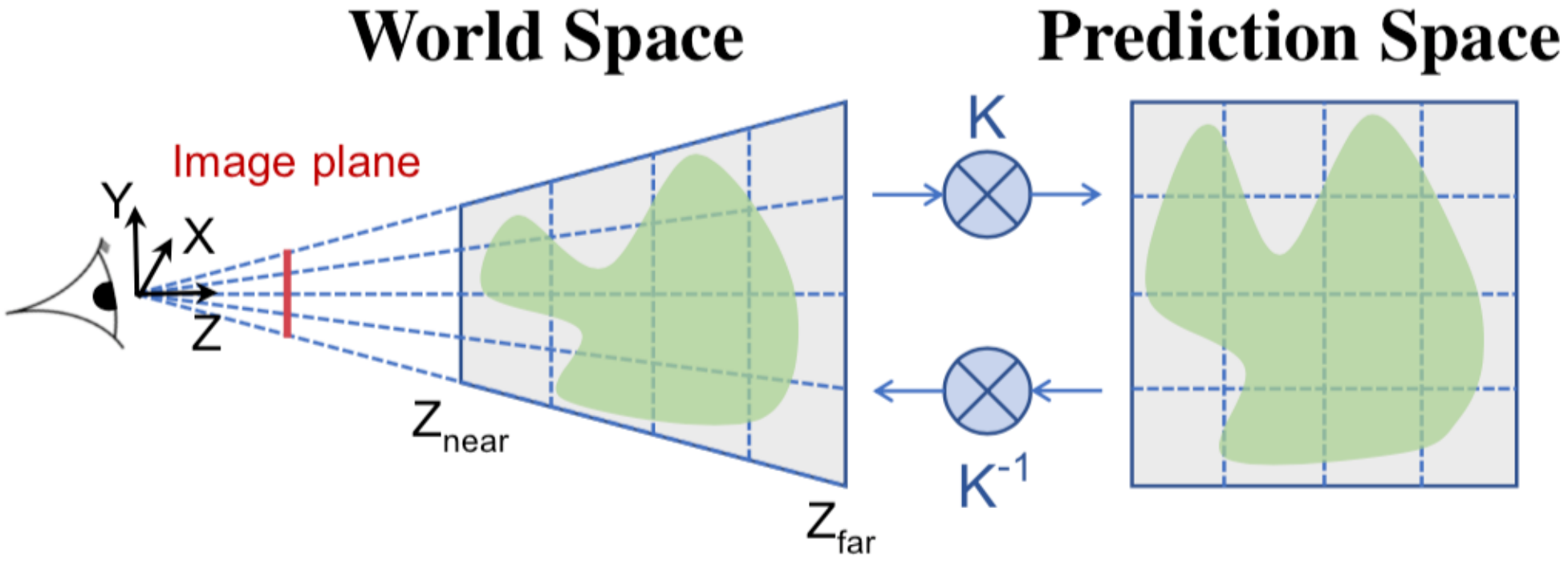


Mesh R-CNN: Architecture

Voxel branch: Predict voxels with CNN



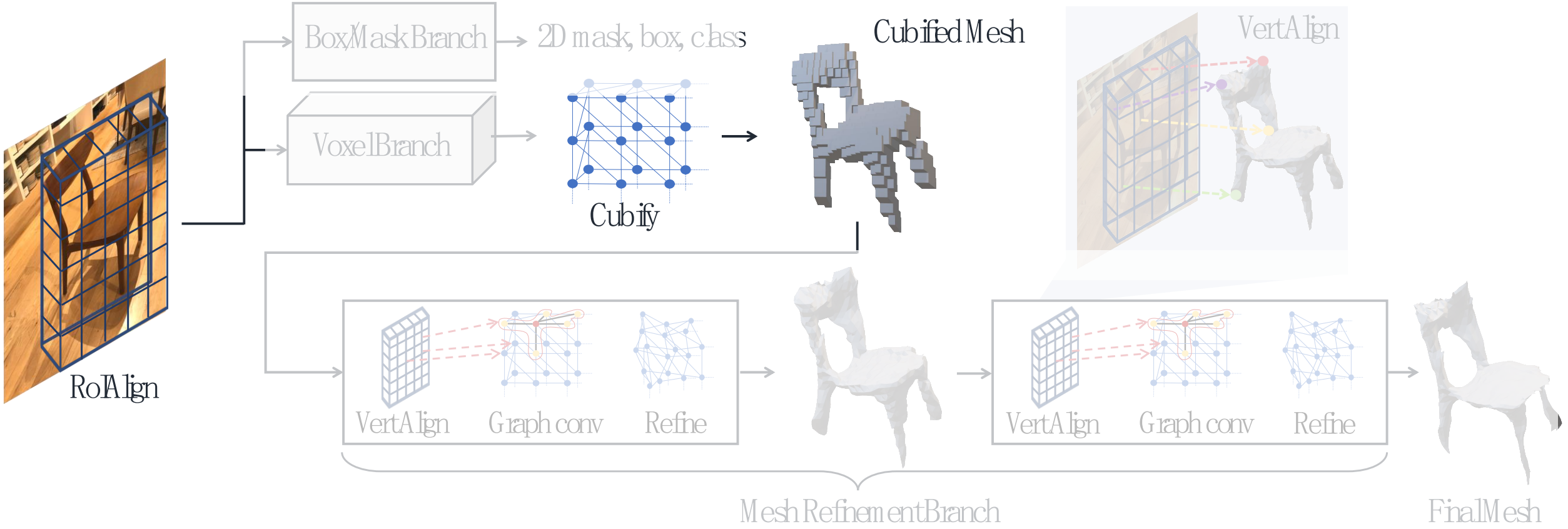
Mesh R-CNN: View-Centric Voxels



View-centric predictions! Voxels take perspective camera into account, so our “voxels” are actually frustums

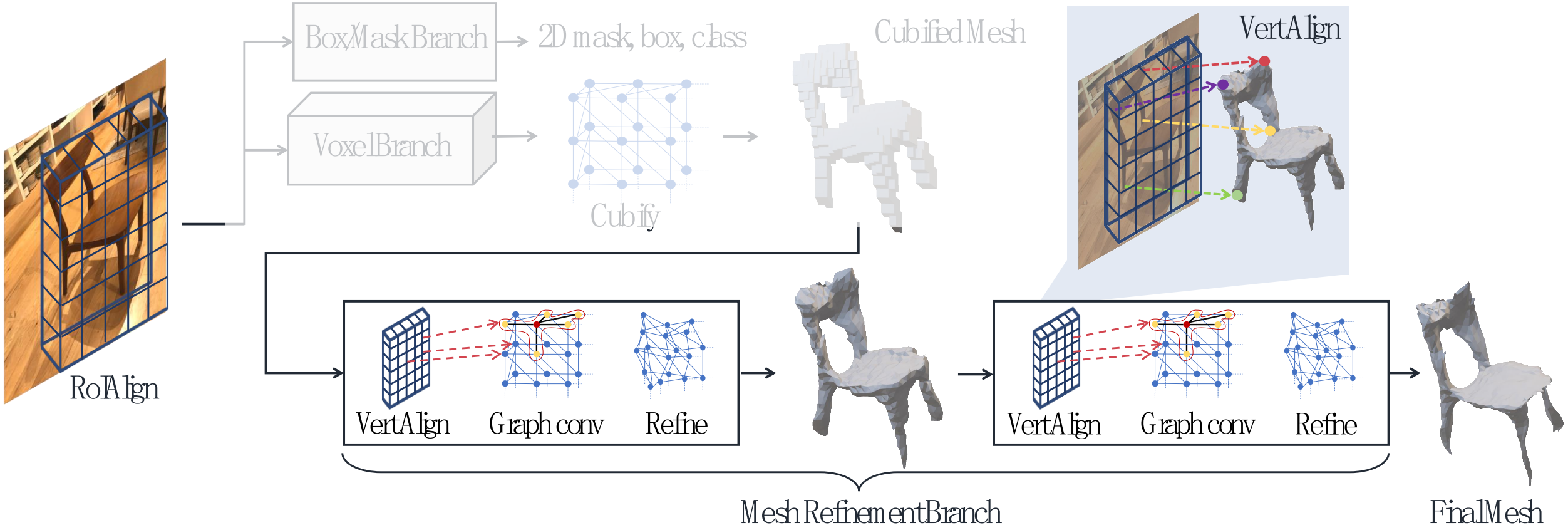
Mesh R-CNN: Architecture

Cubify: Threshold voxels,
convert to cube-like mesh



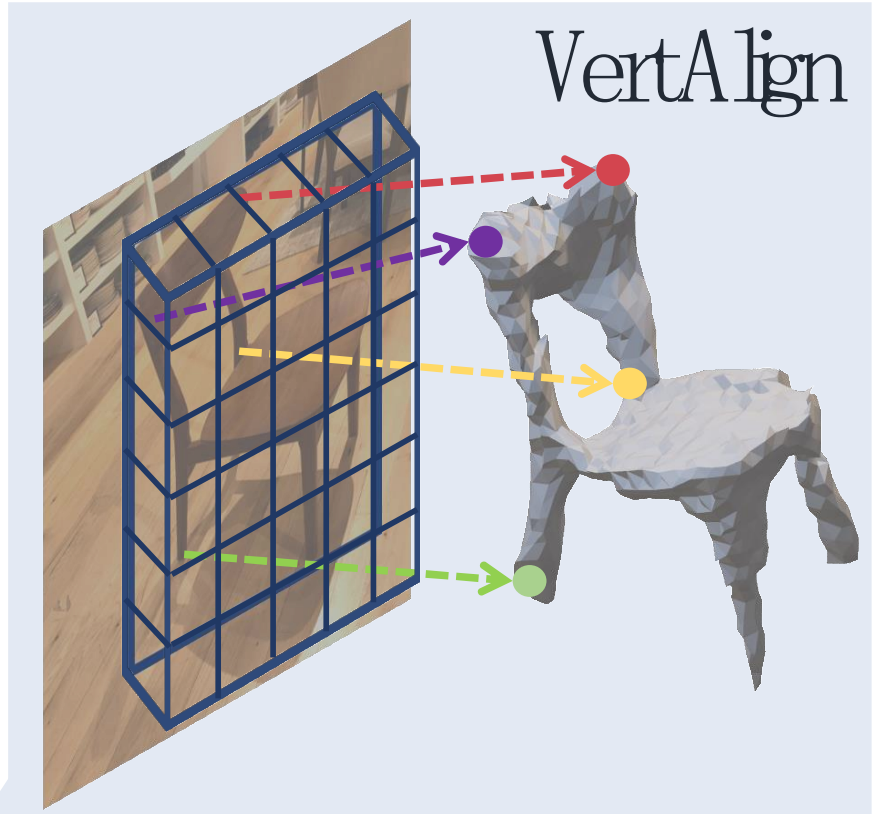
Mesh R-CNN: Architecture

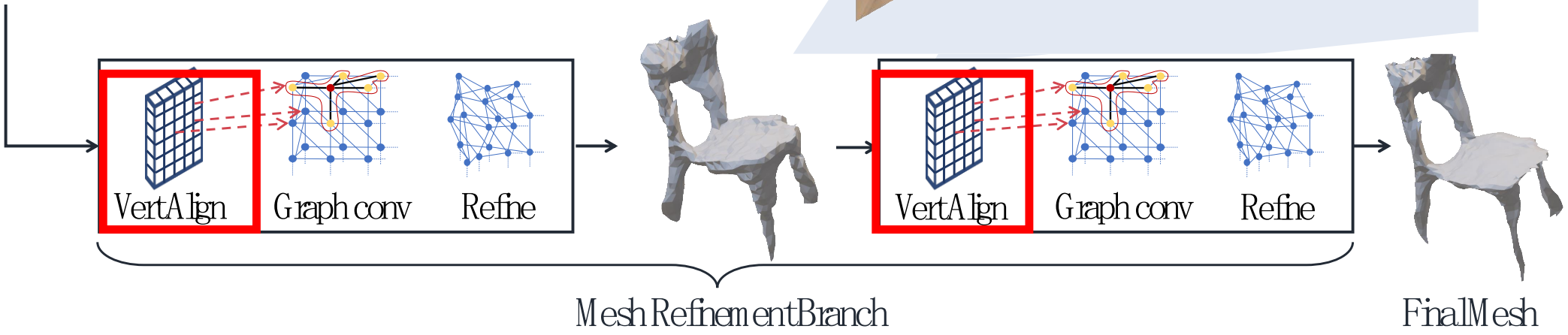
Mesh refinement branch: Iteratively refine mesh through stages of vertex feature alignment, graph convolution, and vertex offset



Mesh R-CNN: Architecture

Vertex Alignment:
Project each vertex onto the image plane, and (bilinearly) sample from the RoI features

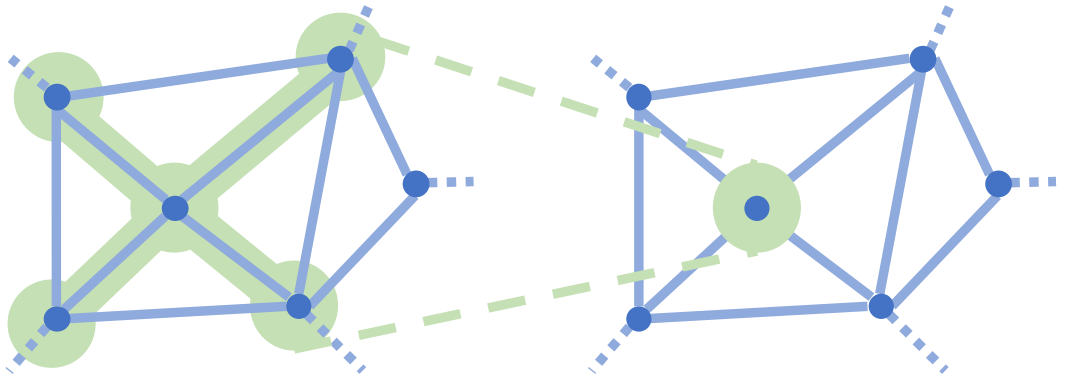


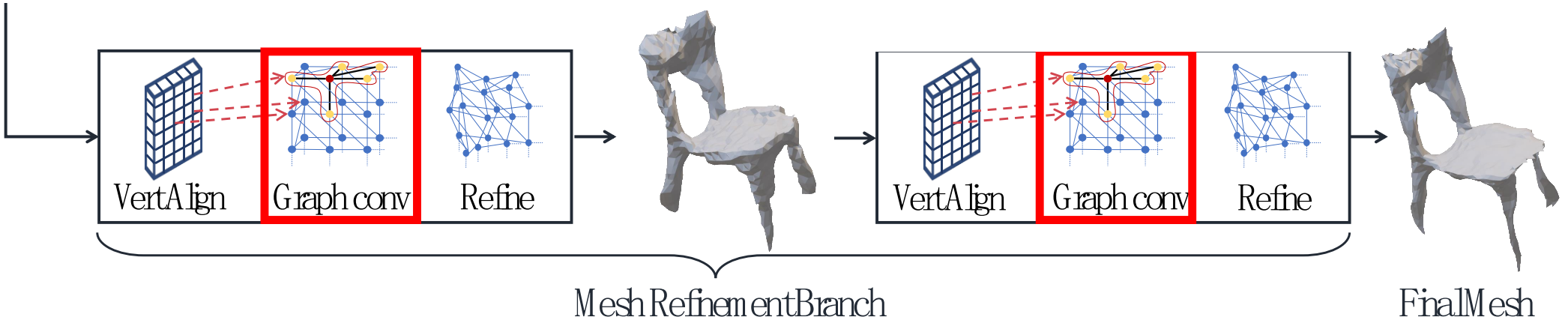


Mesh R-CNN: Architecture

$$W_0 f_i + \sum_{j \in \mathcal{N}(i)} W_1 f_j$$

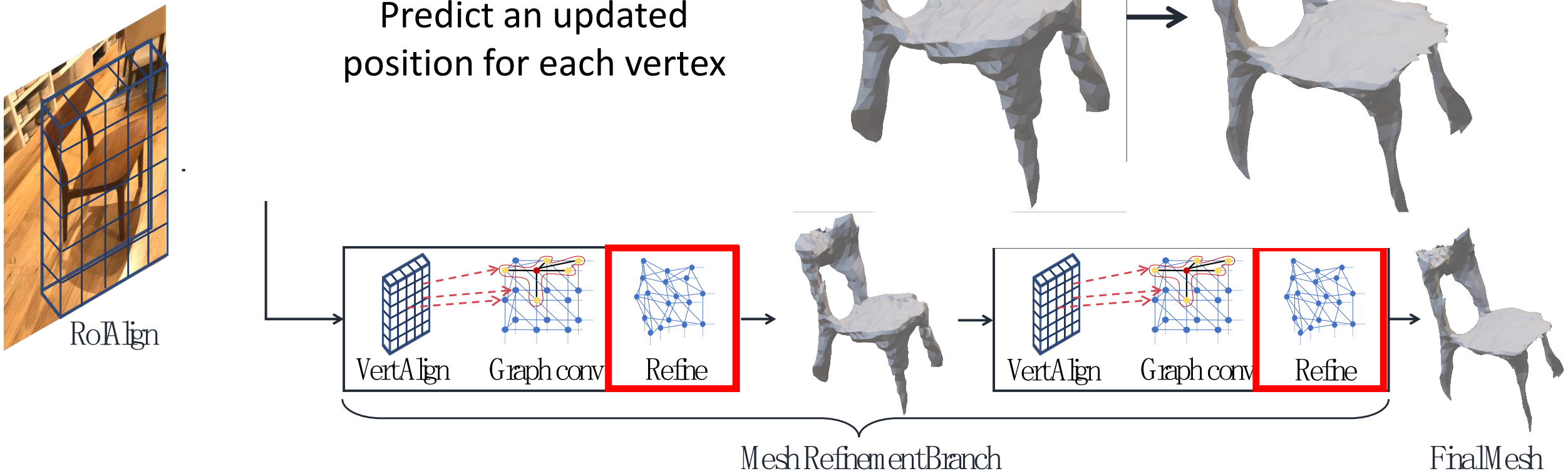
Graph Convolution:
Compute a new feature for each vertex by exchanging information among adjacent vertices



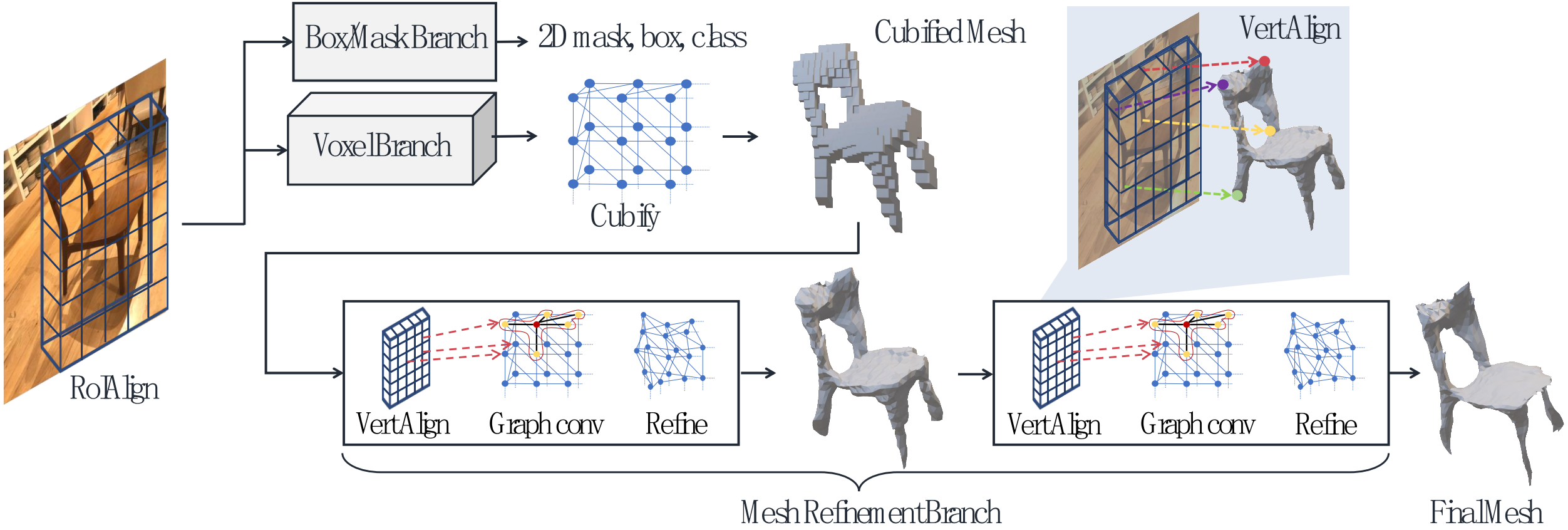


Mesh R-CNN: Architecture

Mesh Refinement:
Predict an updated position for each vertex



Mesh R-CNN: Architecture



Mesh R-CNN: Training Losses

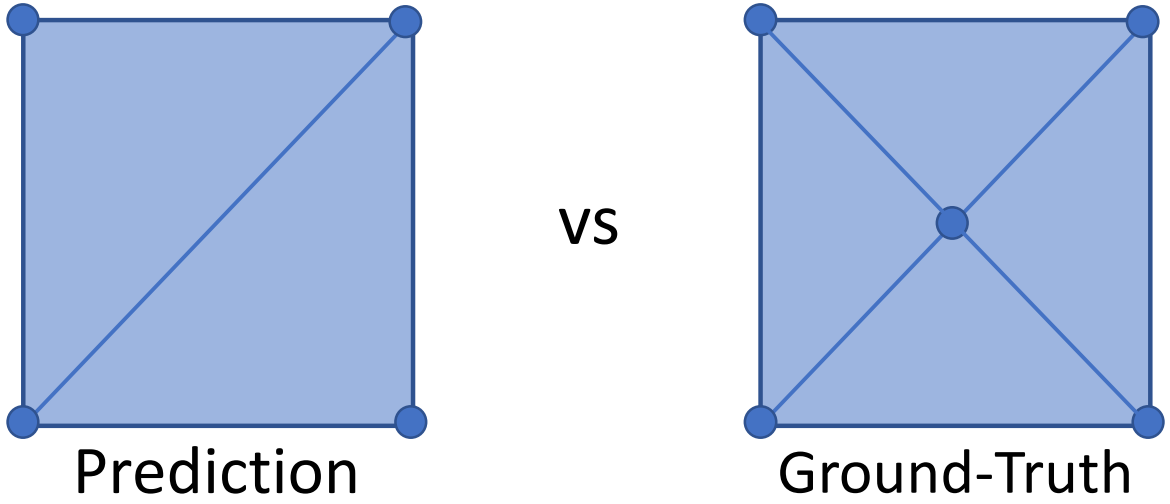
- Instance segmentation losses: Same as Mask R-CNN
  - RPN classification
  - RPN bounding box regression
  - Per-region classification
  - Per-region bounding box regression
  - Per-region instance segmentation mask
- Voxel loss: Binary cross-entropy loss on voxel occupancy
- Mesh loss: Chamfer distance on sampled points at each stage

Mesh R-CNN: Training Losses

- Instance segmentation losses: Same as Mask R-CNN
  - RPN classification
  - RPN bounding box regression
  - Per-region classification
  - Per-region bounding box regression
  - Per-region instance segmentation mask
- Voxel loss: Binary cross-entropy loss on voxel occupancy
- **Mesh loss: Chamfer distance on sampled points at each stage**

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

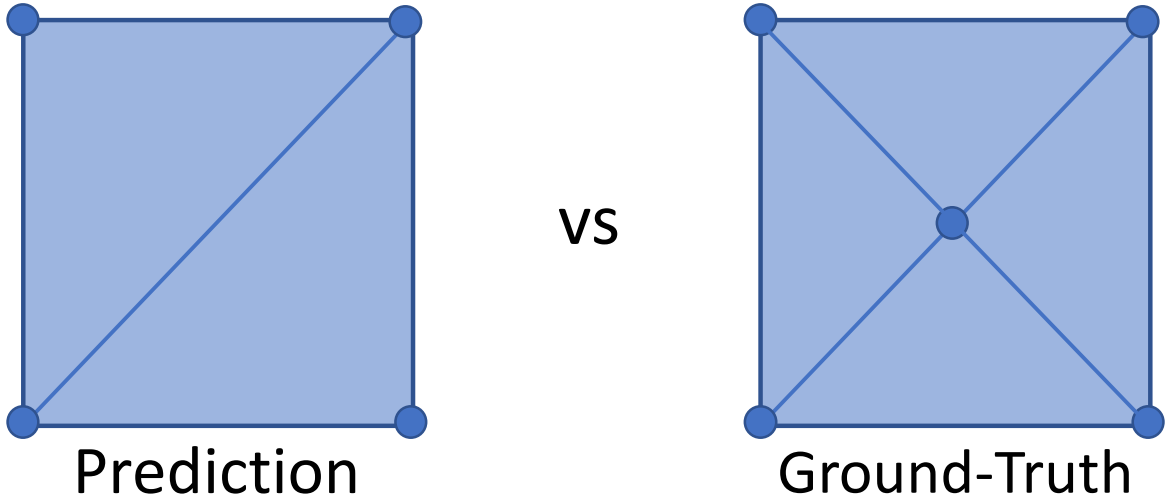


Wang et al, “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images”, ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Idea: Convert meshes into **point clouds**, and then compare

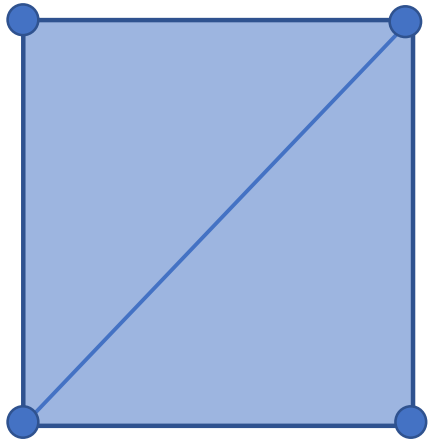


Wang et al, “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images”, ECCV 2018

Predicting Meshes: Loss Function

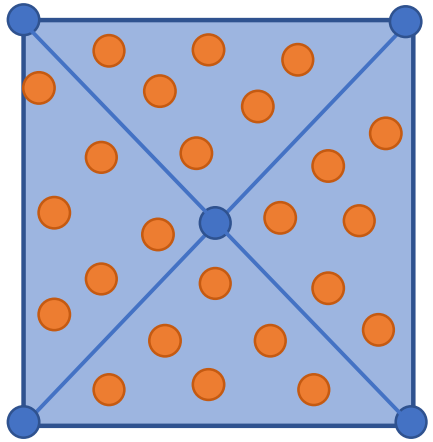
The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Idea: Convert meshes into **point clouds**, and then compare



Prediction

vs



Ground-Truth

Sample points from the surface of the ground-truth mesh (offline)

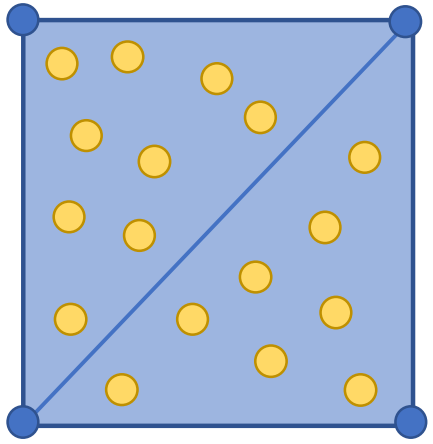
Wang et al, “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images”, ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

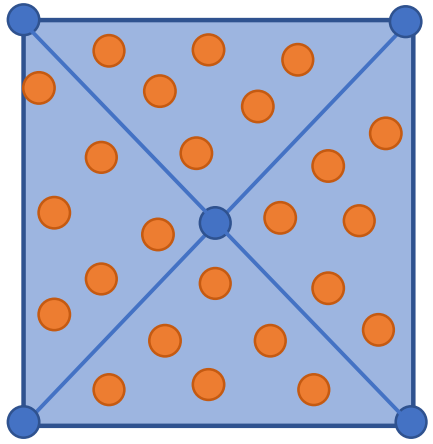
Idea: Convert meshes into **point clouds**, and then compare

Sample points from the surface of the predicted mesh (online!)



Prediction

vs



Ground-Truth

Sample points from the surface of the ground-truth mesh (offline)

Wang et al, “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images”, ECCV 2018

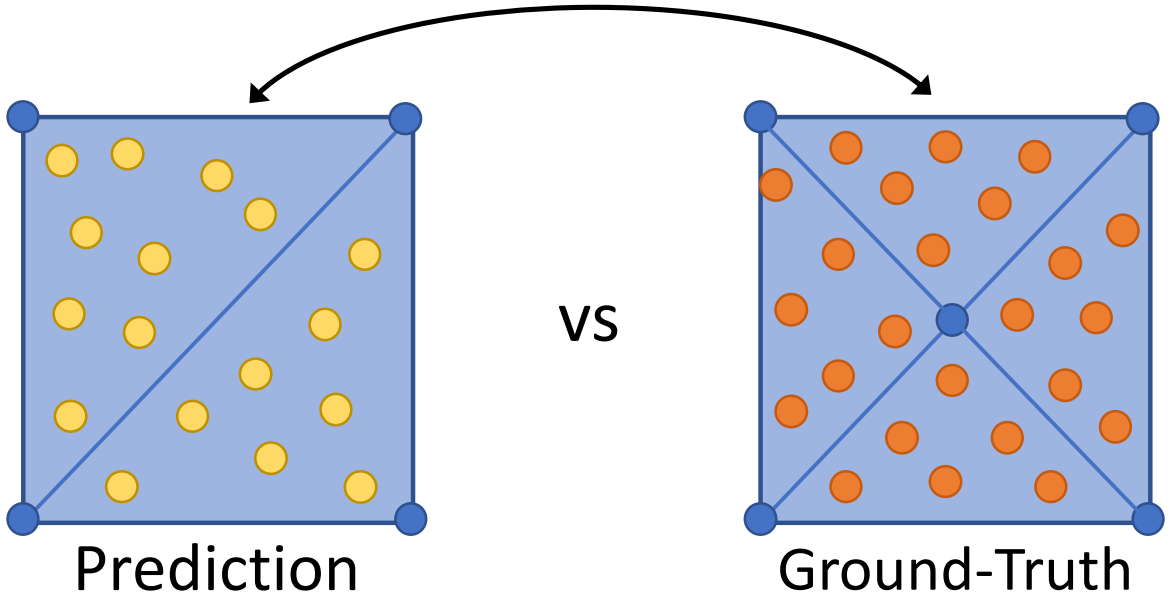
Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Idea: Convert meshes into **point clouds**, and then compare

Loss: Compare **samples** from prediction and **samples** from ground-truth

Sample points from the surface of the predicted mesh (online!)



Sample points from the surface of the ground-truth mesh (offline)

Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

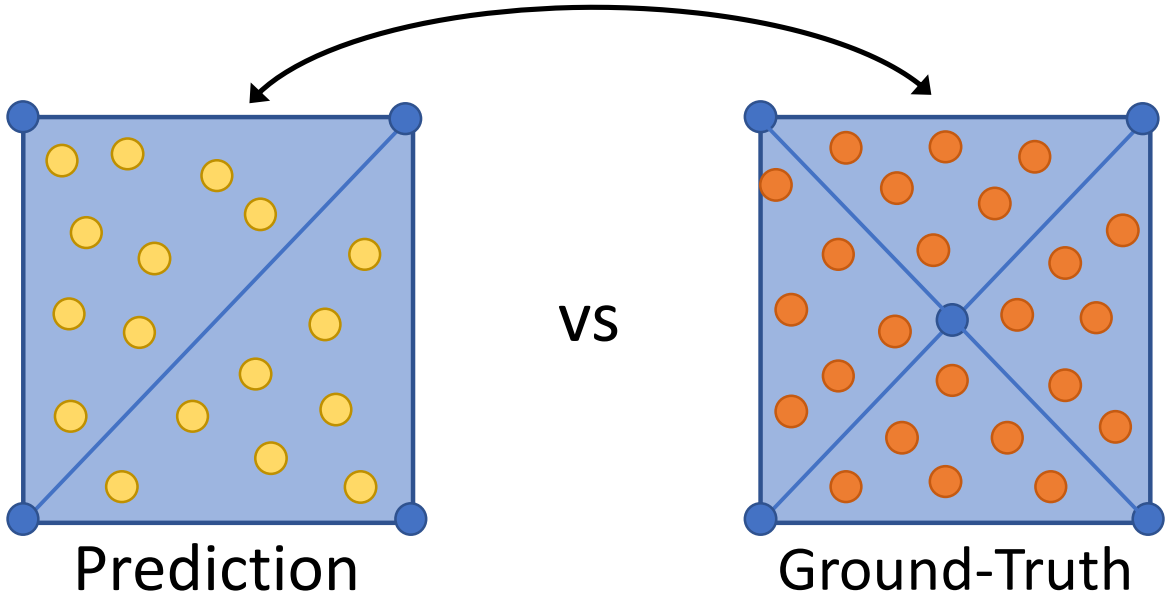
Predicting Meshes: Loss Function

Problem: Need to sample online! Must be efficient!

Problem: Need to backprop through sampling!

Loss: Compare **samples** from prediction and **samples** from ground-truth

Sample points from the surface of the predicted mesh (online!)



Sample points from the surface of the ground-truth mesh (offline)

Smith et al, “GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects”, ICML 2019

Comparing Point Clouds: Chamfer Distance

We need a (differentiable) way to compare pointclouds **as sets**!

Chamfer distance is the sum of L2 distance to each point's nearest neighbor in the other set

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

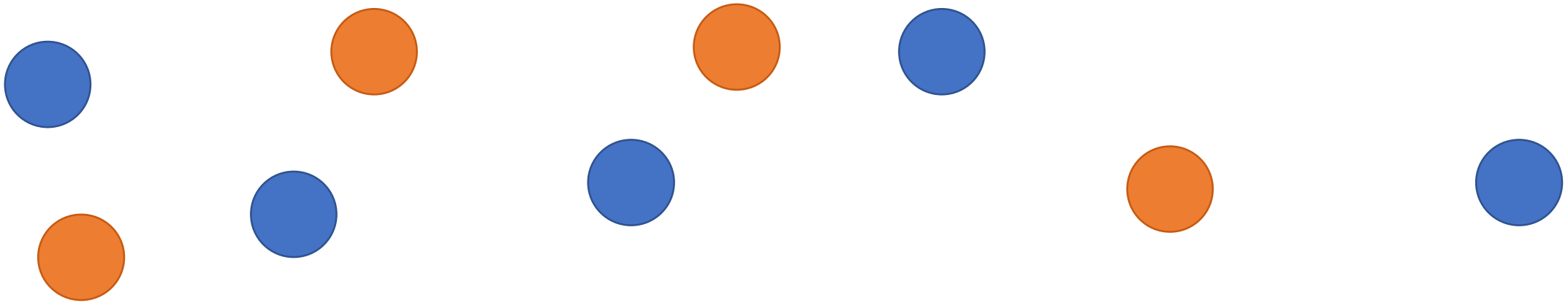
Fan et al, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image", CVPR 2017

Comparing Point Clouds: Chamfer Distance

We need a (differentiable) way to compare pointclouds **as sets**!

Chamfer distance is the sum of L2 distance to each point's nearest neighbor in the other set

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



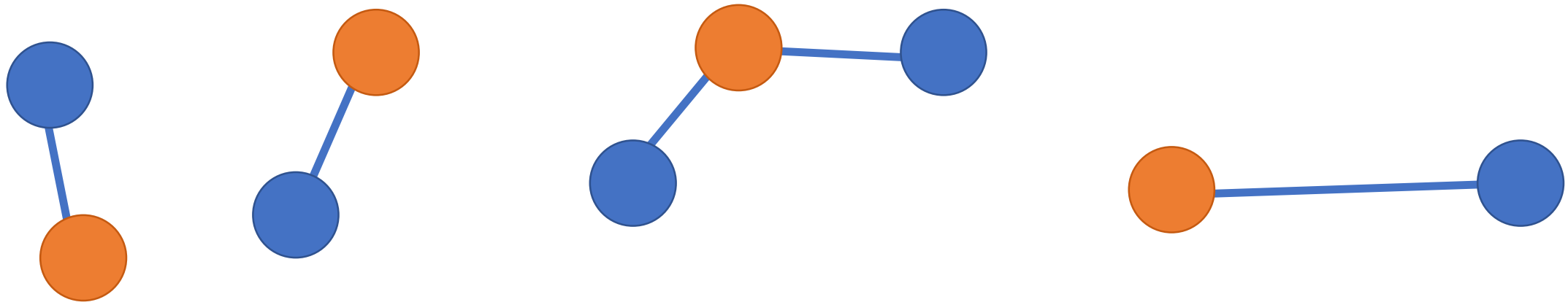
Fan et al, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image", CVPR 2017

Comparing Point Clouds: Chamfer Distance

We need a (differentiable) way to compare pointclouds **as sets**!

Chamfer distance is the sum of L2 distance to each point's nearest neighbor in the other set

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



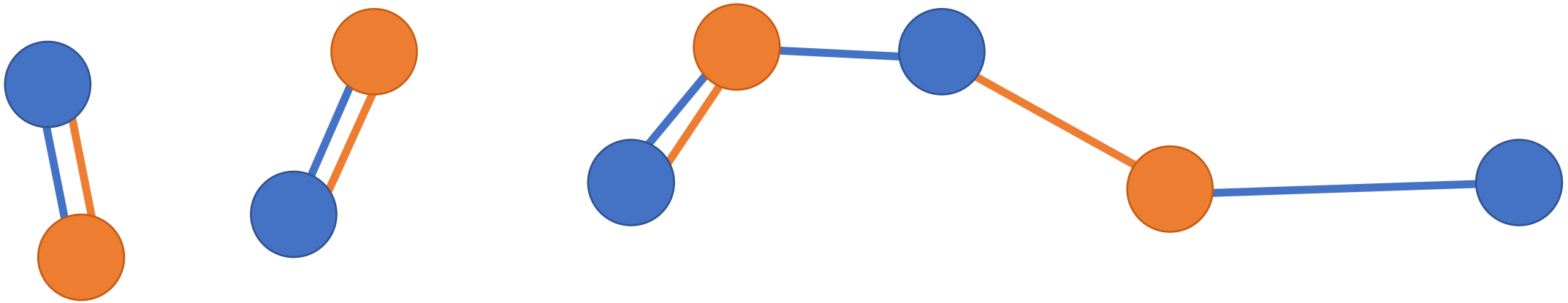
Fan et al, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image", CVPR 2017

Comparing Point Clouds: Chamfer Distance

We need a (differentiable) way to compare pointclouds **as sets**!

Chamfer distance is the sum of L2 distance to each point's nearest neighbor in the other set

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



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

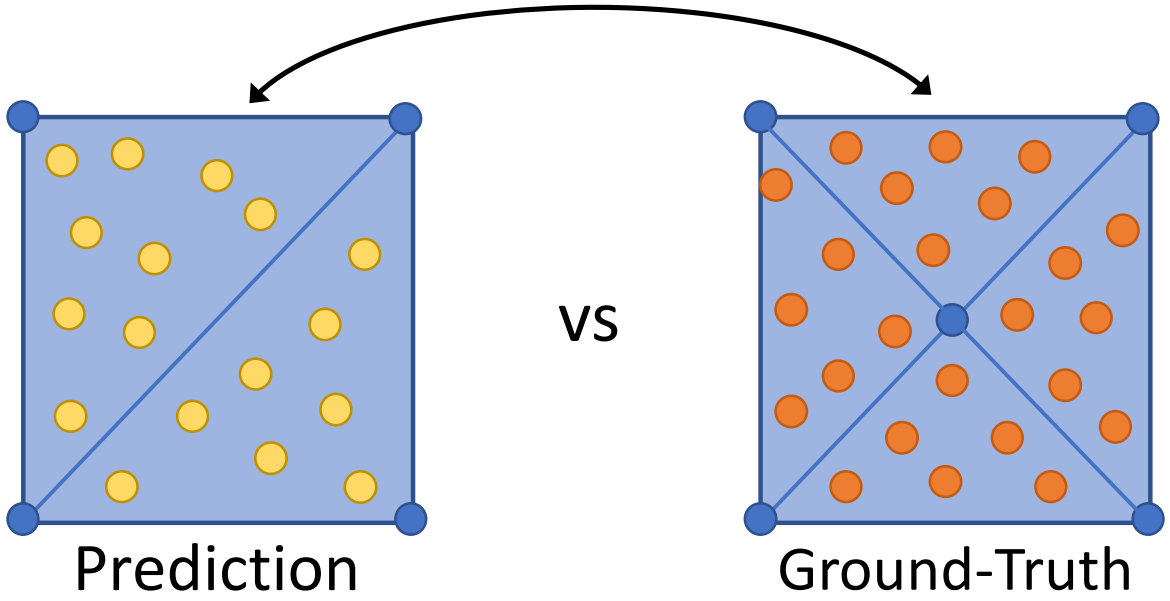
Predicting Meshes: Loss Function

Problem: Need to sample online! Must be efficient!

Problem: Need to backprop through sampling!

Loss: Chamfer distance between samples from prediction and GT

Sample points from the surface of the predicted mesh (online!)



Sample points from the surface of the ground-truth mesh (offline)

Smith et al, “GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects”, ICML 2019

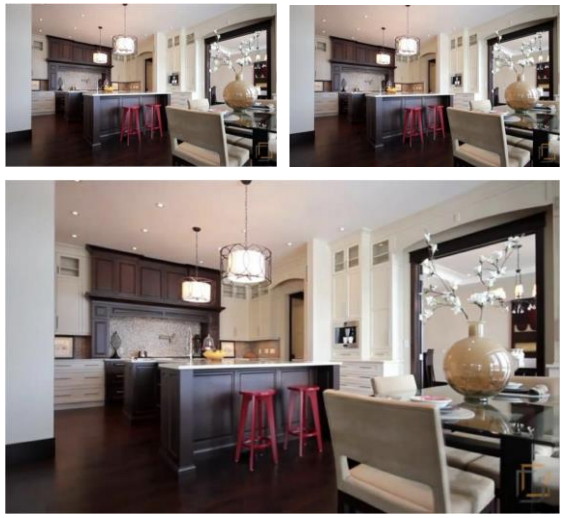
Previous Work

2D Image to Image

(-) Best at synthetic
(+) No GT Depth
(+) Single Image
(+) End-to-End

Zhou et al, ECCV 2016; Tatarchenko et al, ECCV 2016; Park et al, CVPR 2017; Worrall et al, ICCV 2017; Sun et al, ECCV 2018

Multi-View Inputs



(+) Real-World Images
(+) No GT depth
(-) Stereo Images
(+) End-to-End

Zhou et al, SIGGRAPH 2018; Srinivasan et al, CVPR 2019

Depth + Inpainting

(+) Real-World Images
(-) Train with GT Depth
(+) Single Image
(-) Not End-to-End

Niklaus et al, SIGGRAPH Asia 2019

Deep Learning with Latent 3D

(-) Synthetic data
(+) No GT Depth
(+) Single Image
(+) End-to-End

Sitzmann et al, CVPR 2019; Sitzmann et al, NeurIPS 2019; Chen et al, ICCV 2019; Nguyen-Phuoc et al, ICCV 2019

Comparison with State-of-the-Art

System comparison on RealEstate10K [71]			
	PSNR ↑	SSIM ↑	Perc Sim ↓
SynSin	$22.78_{5.25}$	$0.74_{0.17}$	$0.95_{0.61}$
3DView	$21.88_{8.43}$	$0.64_{0.24}$	$1.29_{0.88}$
StereoMag [71]	$25.34_{9.48}$	$0.82_{0.14}$	$1.10_{0.73}$

3DView : Model similar to: Niklaus, Simon, et al. “3D Ken Burns Effect from a Single Image.” *ACM Transactions on Graphics.* 2019.
StereoMag: Zhou, Tinghui, et al. "Stereo magnification: Learning view synthesis using multiplane images.” *ACM Transactions on Graphics.* 2018.

Comparison with State-of-the-Art

3DView : Model similar to: Niklaus, Simon, et al. “3D Ken Burns Effect from a Single Image.” *ACM Transactions on Graphics.* 2019.
StereoMag: Zhou, Tinghui, et al. "Stereo magnification: Learning view synthesis using multiplane images.” *ACM Transactions on Graphics.* 2018.