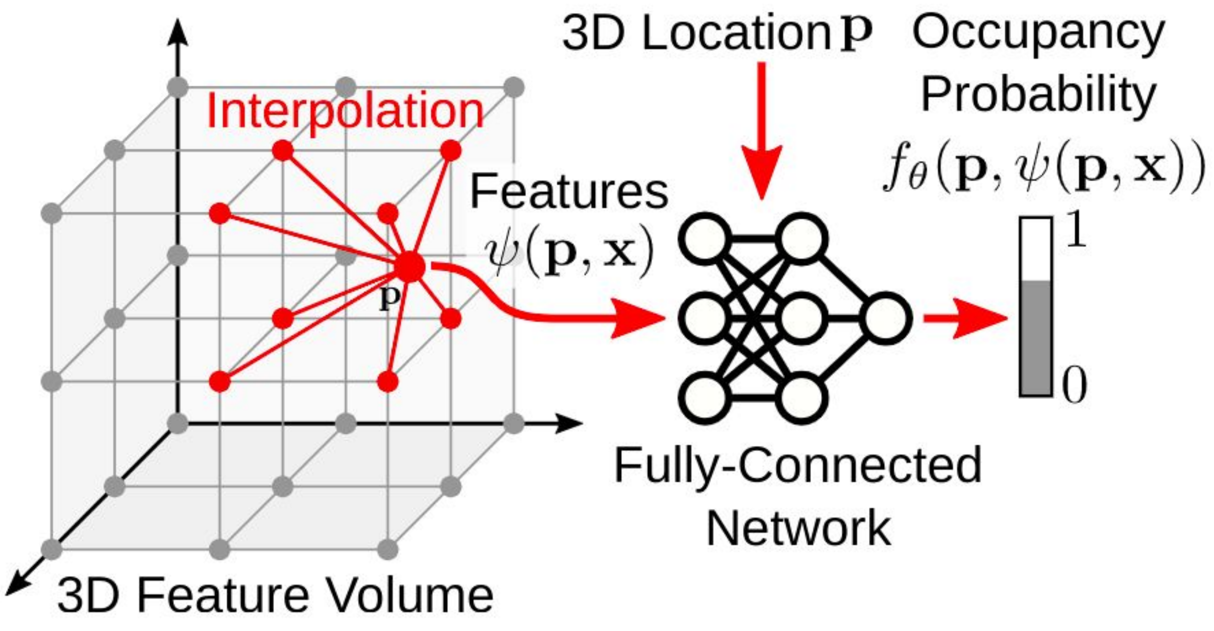
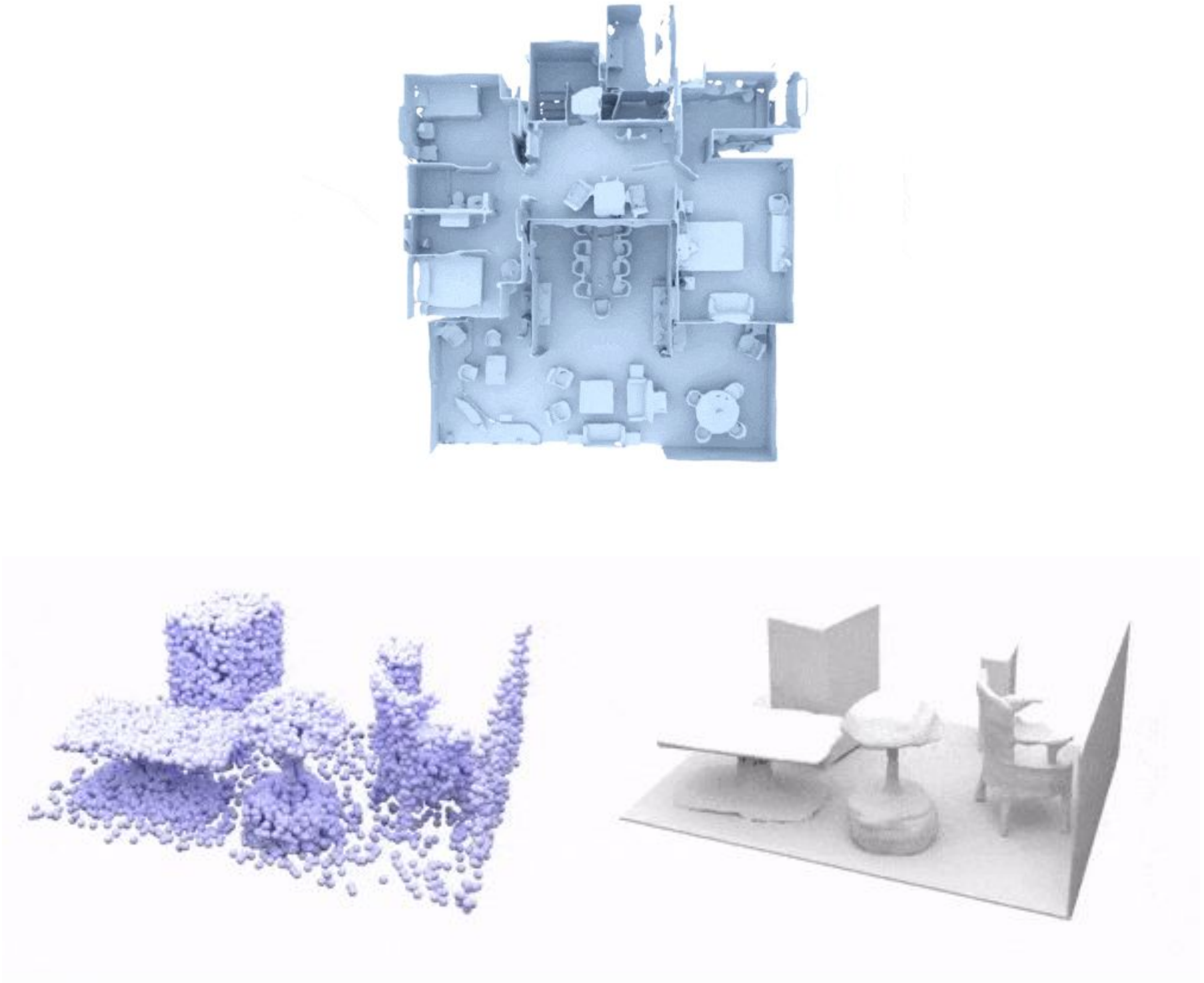
Seminar Presentation: Recent Advances in 3D Computer Vision





Panagiotis Petropoulakis, and Björn Häfner

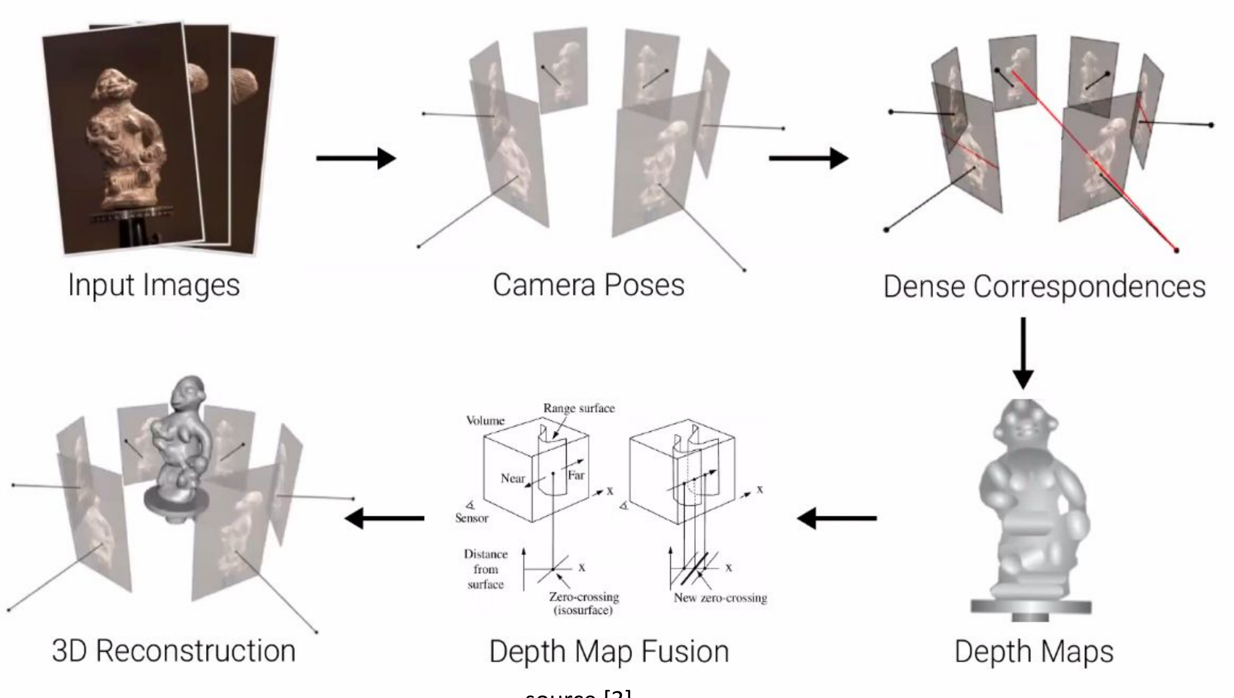
Munich, October 2022

Outline

1. Introduction of the problem
2. Approach
3. Results
4. Personal comments
5. Summary

1. Introduction of the problem

- **Traditional 3D reconstruction**

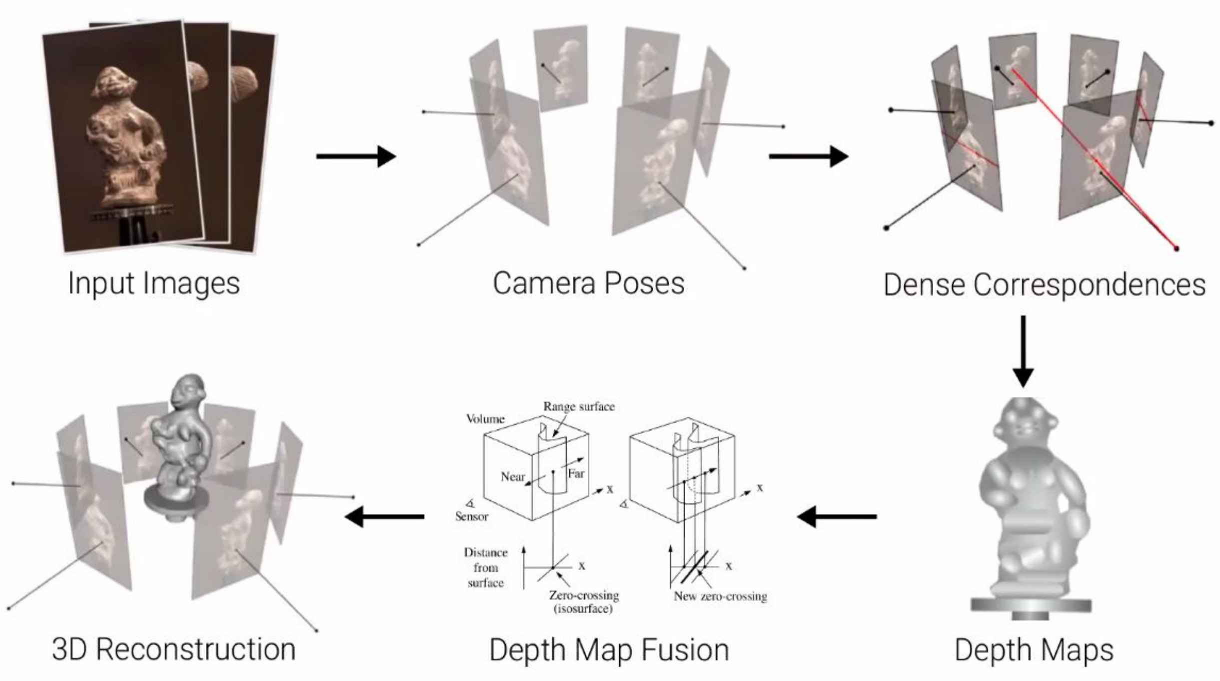


source [3]

-> Multiple images are required as an input at test time

1. Introduction of the problem

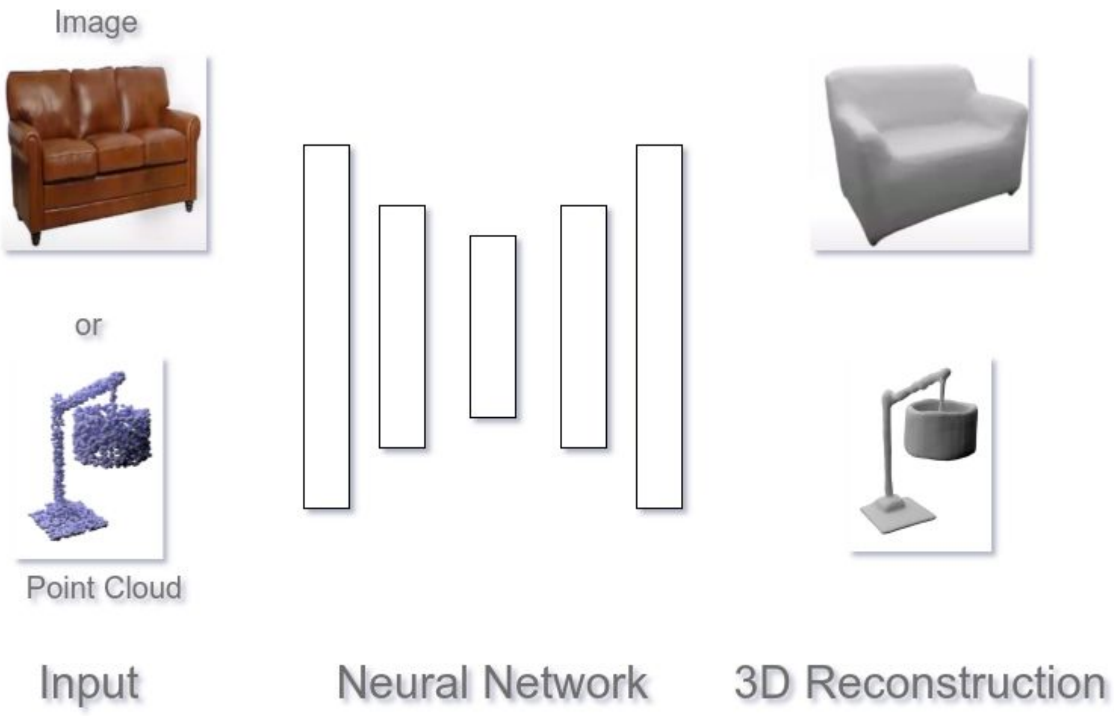
- **Traditional 3D reconstruction**



source [3]

-> Multiple images are required as an input at test time

- **Learning-based 3D reconstruction**



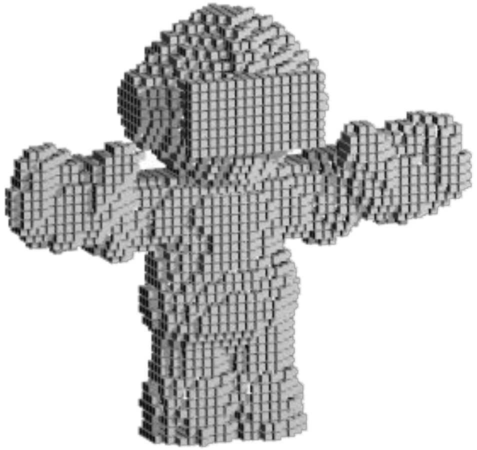
-> Learn the 3D shape

-> 3D reconstruction from a single input

1. Introduction of the problem

Common output representation of Learning-based 3D Reconstruction methods

<u>Voxels:</u>



source [4]

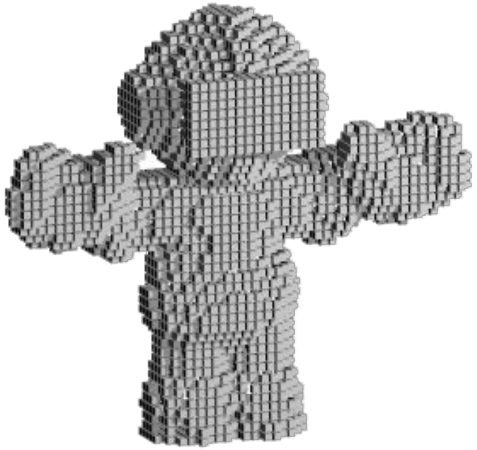
-> Discretize into a grid

- High memory consumption

1. Introduction of the problem

Common output representation of Learning-based 3D Reconstruction methods

Voxels:

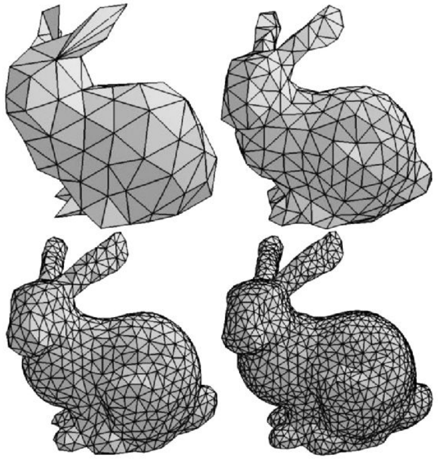


source [4]

-> Discretize into a grid

- High memory consumption

Meshes:



source [5]

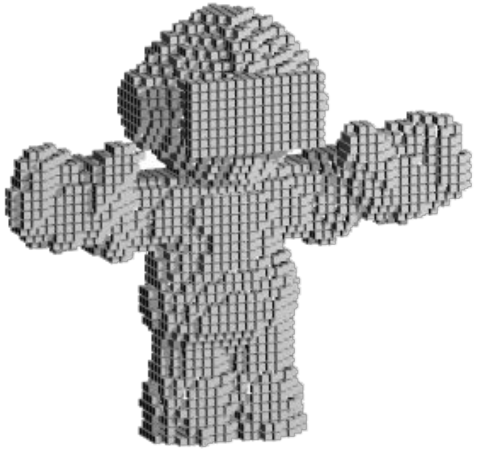
-> Regress into vertices & faces

- Non-watertight reconstructions
- Often require deforming a template

1. Introduction of the problem

Common output representation of Learning-based 3D Reconstruction methods

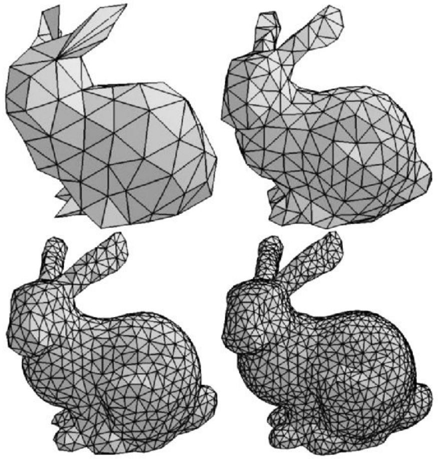
Voxels:



source [4]

-> Discretize into a grid
- High memory consumption

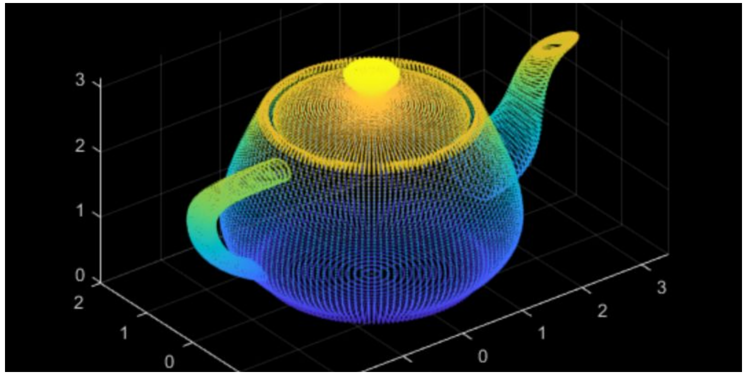
Meshes:



source [5]

-> Regress into vertices & faces
- Non-watertight reconstructions
- Often require deforming a template

Point Clouds:



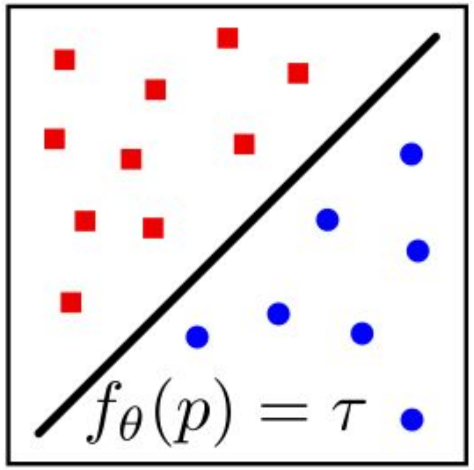
source [6]

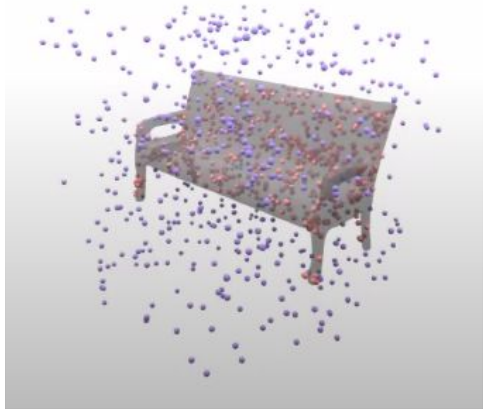
-> Predict the coordinates of 3D points
- Limited number of points
- Topological relations are lost

1. Introduction of the problem

Neural Implicit Representation

- No discretization of the 3D space
- No topological restrictions
- Independent of the camera viewpoint





source [2, 3]

-> Represent the 3D shape implicitly

-> Surface <=> Decision boundary of a non-linear classifier

1. Introduction of the problem

Neural Implicit Representation

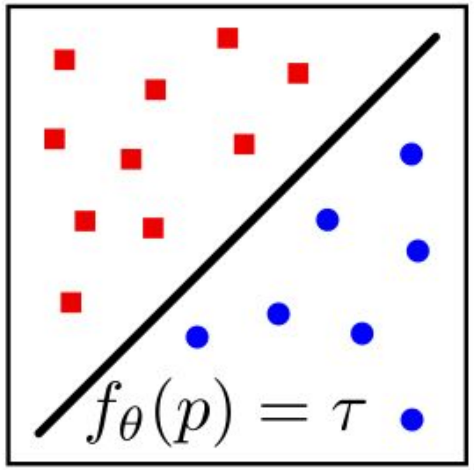
- No discretization of the 3D space
- No topological restrictions
- Independent of the camera viewpoint

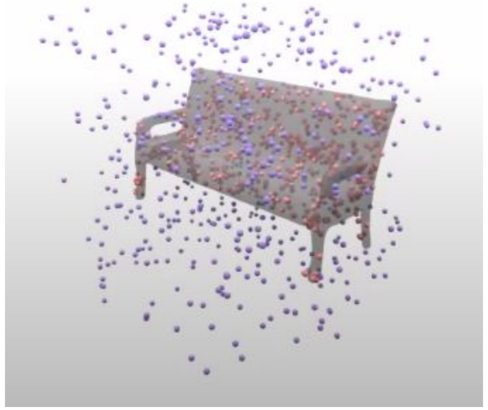
$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$$

3D point   Condition   Occupancy Probability

-> Represent the 3D shape implicitly

-> Surface <=> Decision boundary of a non-linear classifier





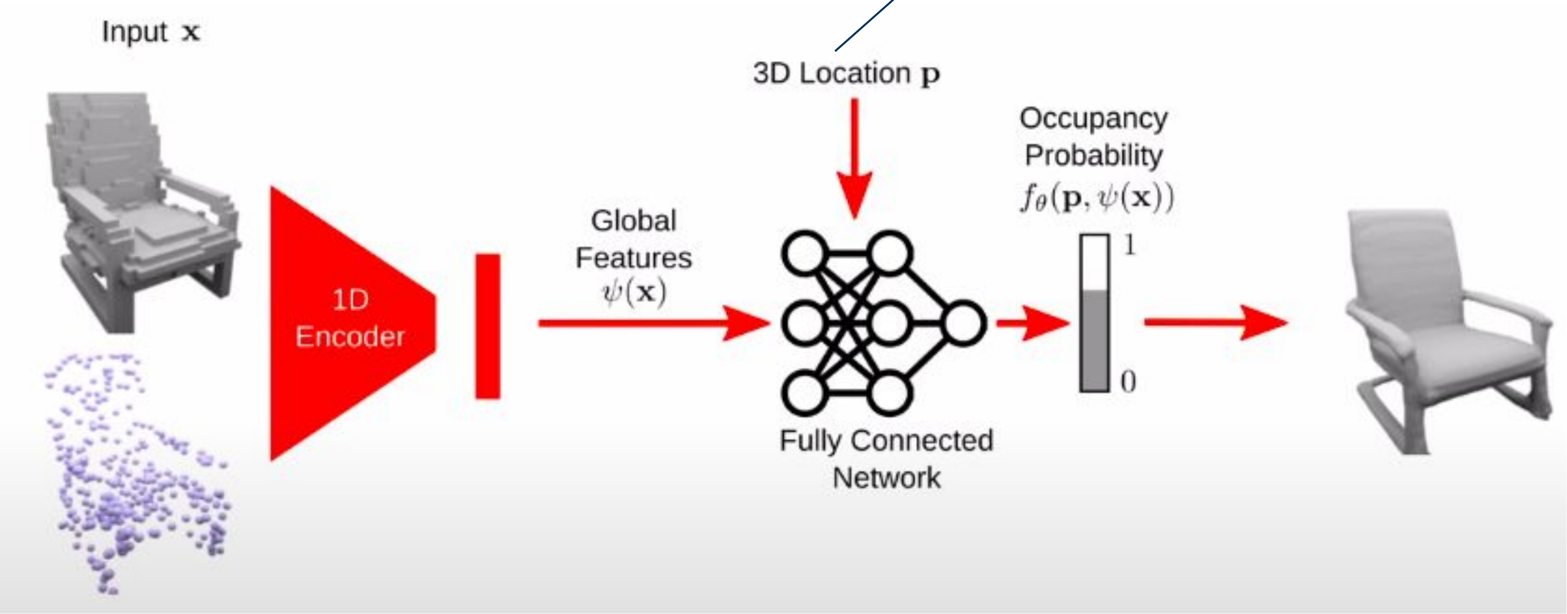
source [2, 3]

1. Introduction of the problem

Problems with previous works

-> Occupancy Networks

Query Point

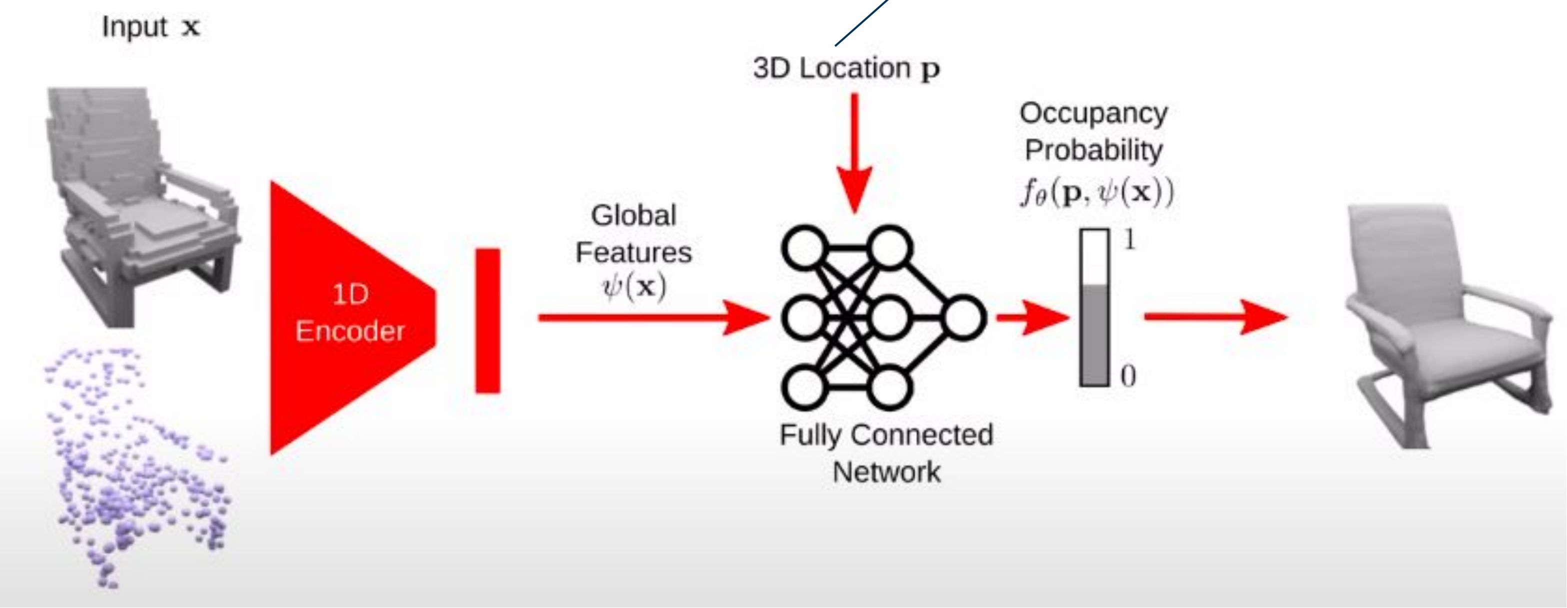


1. Introduction of the problem

Problems with previous works

-> Occupancy Networks

Query Point

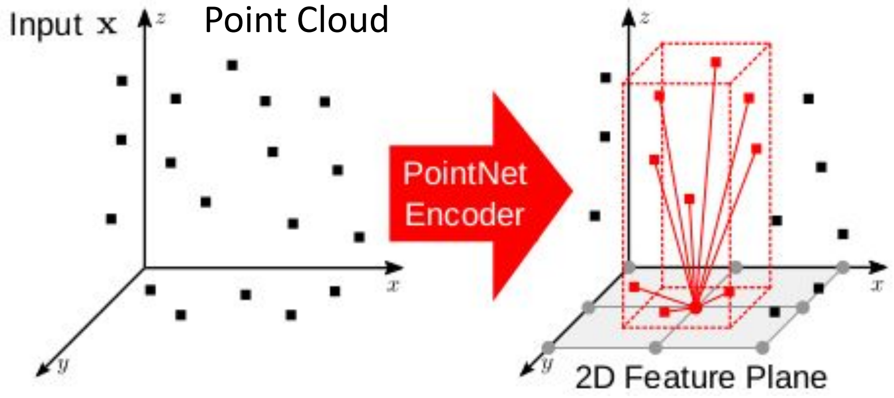


- Local details are not preserved
  - Overly smooth reconstruction
- No Translation Equivariance
- Mainly for simple objects

2. Approach

2D Method

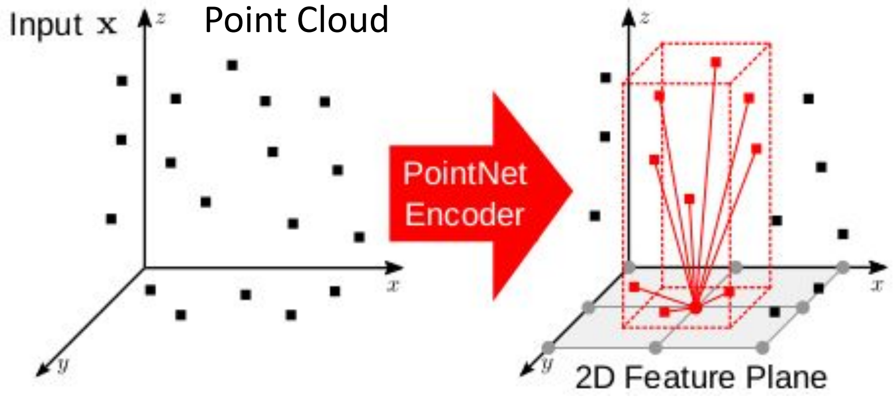
Encoder



2. Approach

2D Method

Encoder

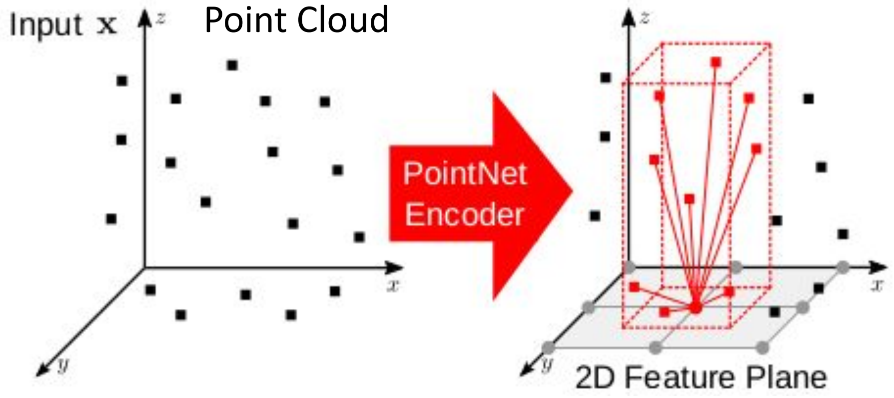


1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Approach

2D Method

Encoder



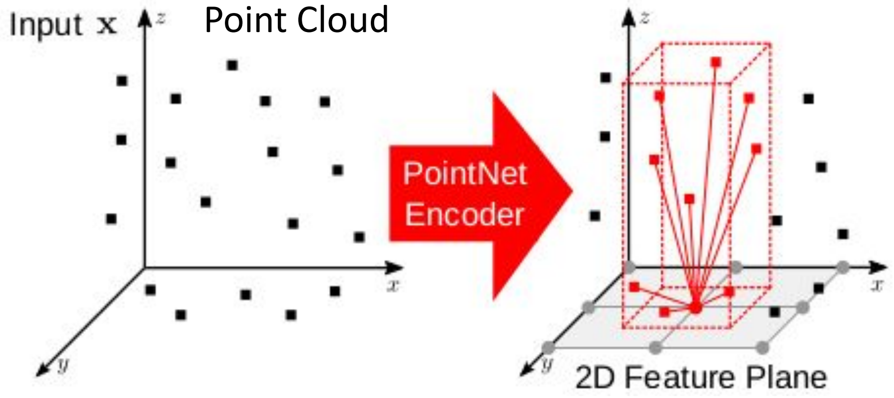
1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

2. Approach

2D Method

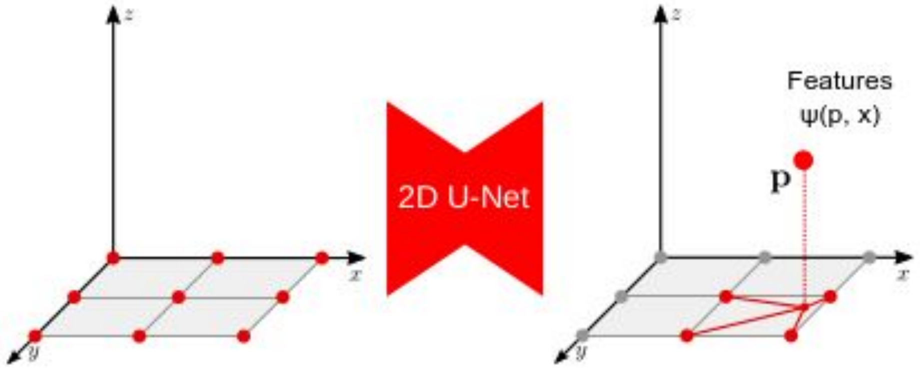
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

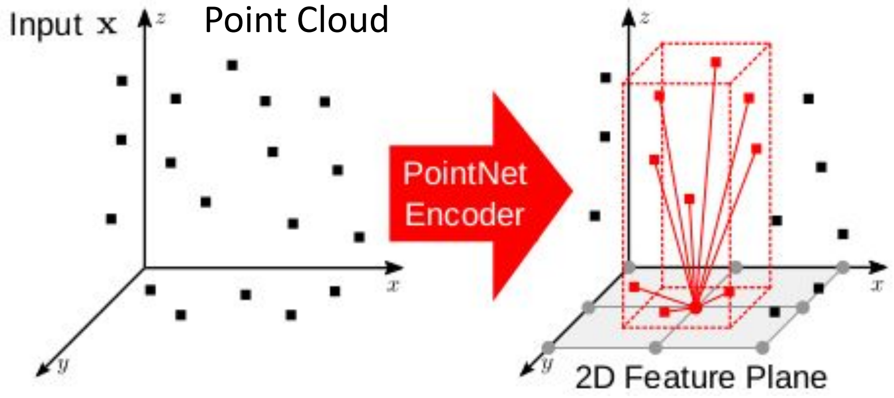
Decoder



2. Approach

2D Method

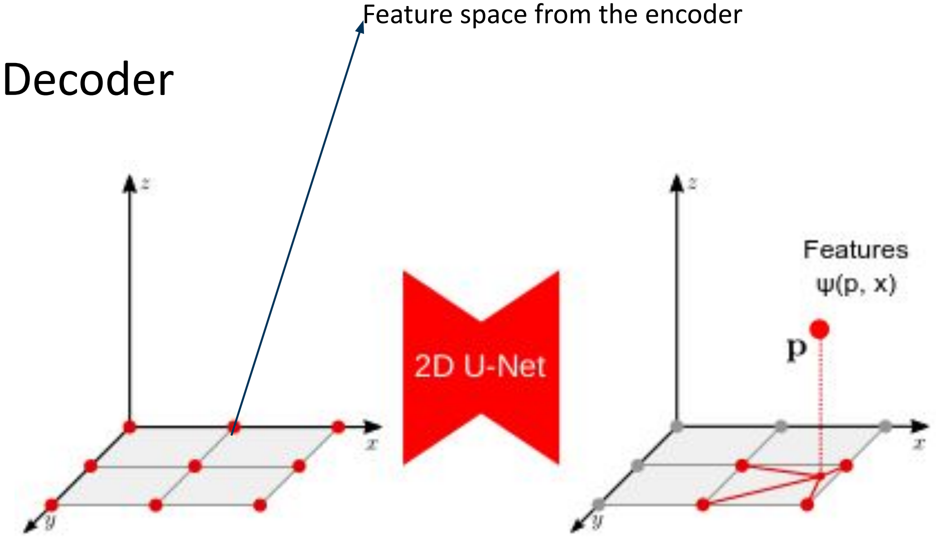
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

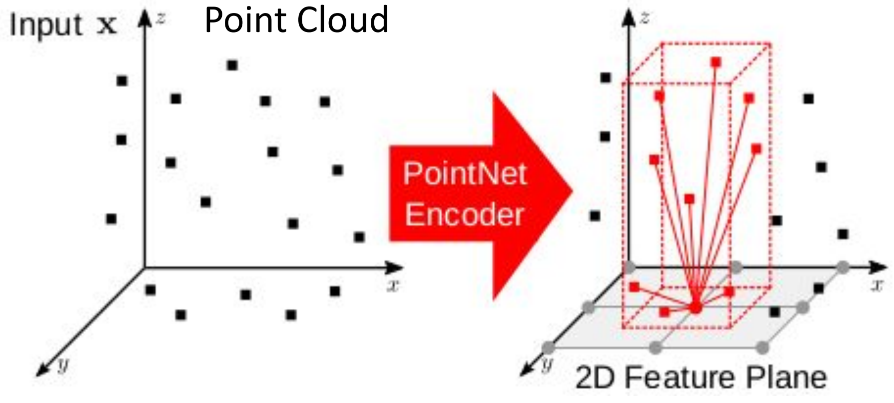
Decoder



2. Approach

2D Method

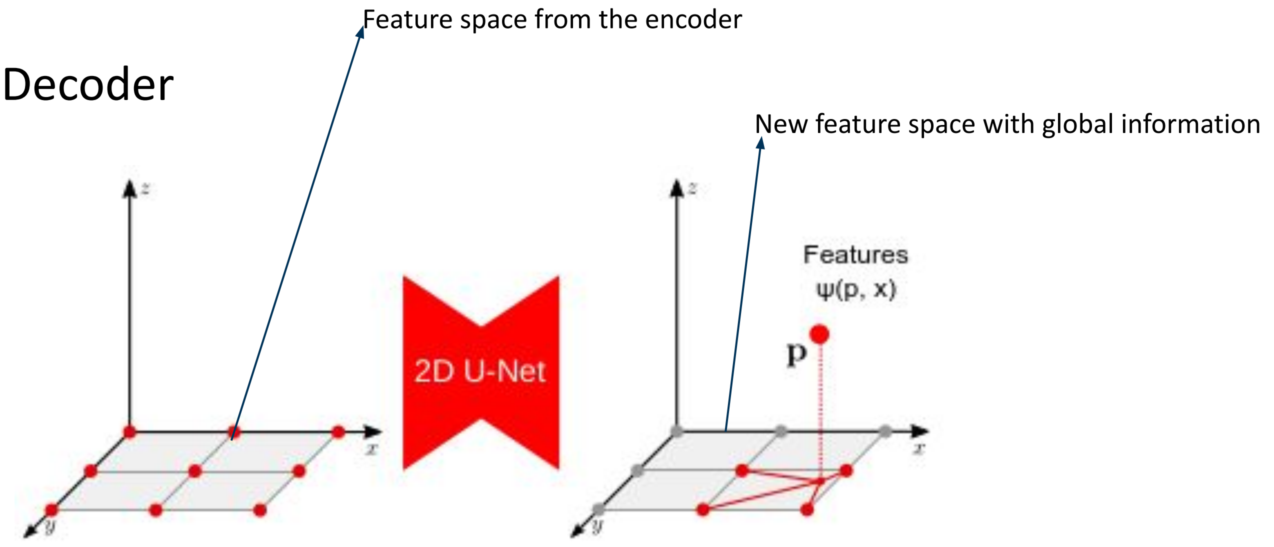
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

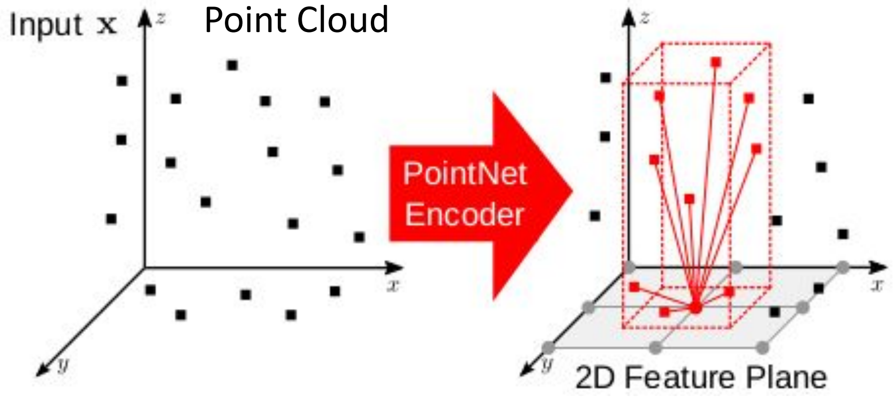
Decoder



2. Approach

2D Method

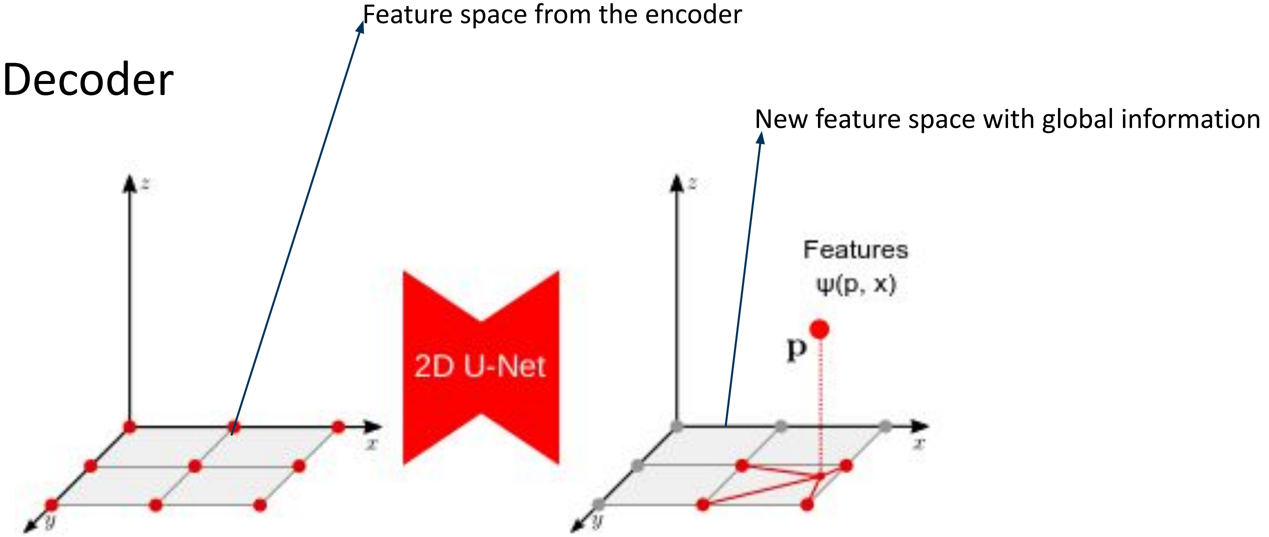
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

Decoder

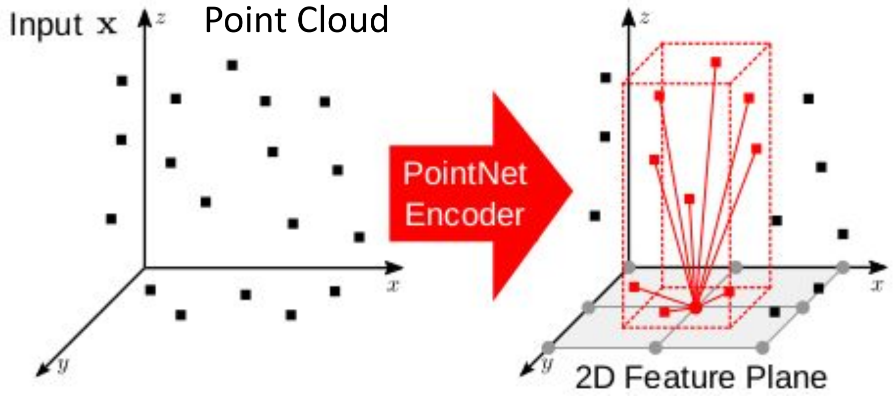


1. Process the Feature Plane (space)
   -> 2D U-Net
   +Integrate global information
   +Translation equivariance

2. Approach

2D Method

Encoder

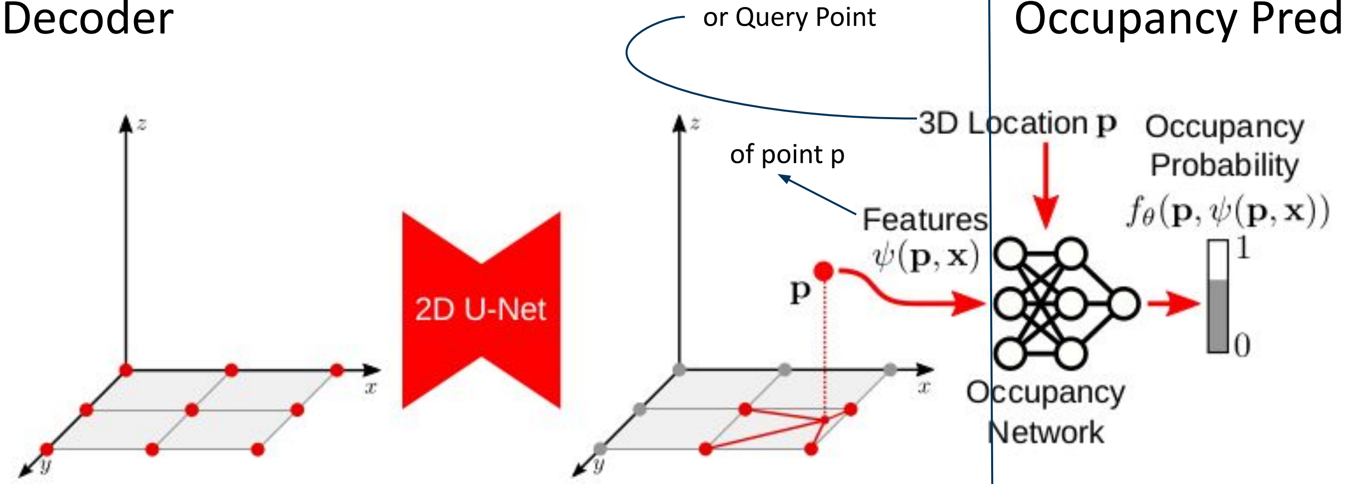


1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

Decoder

Occupancy Prediction

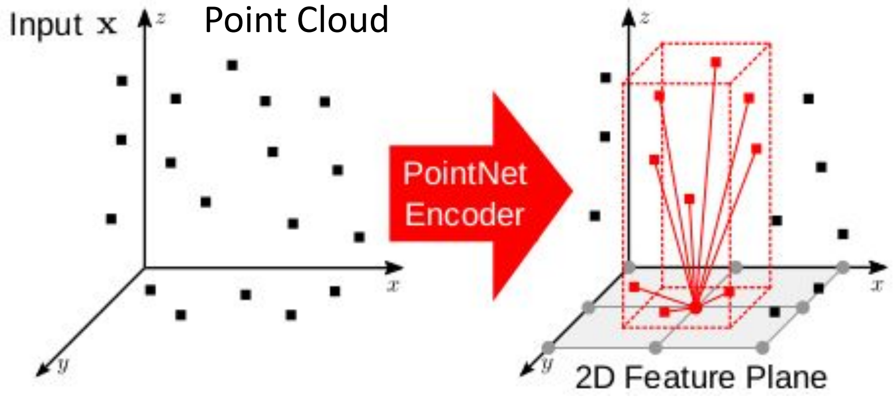


1. Process the Feature Plane (space)
   -> 2D U-Net
   +Integrate global information
   +Translation equivariance

2. Approach

2D Method

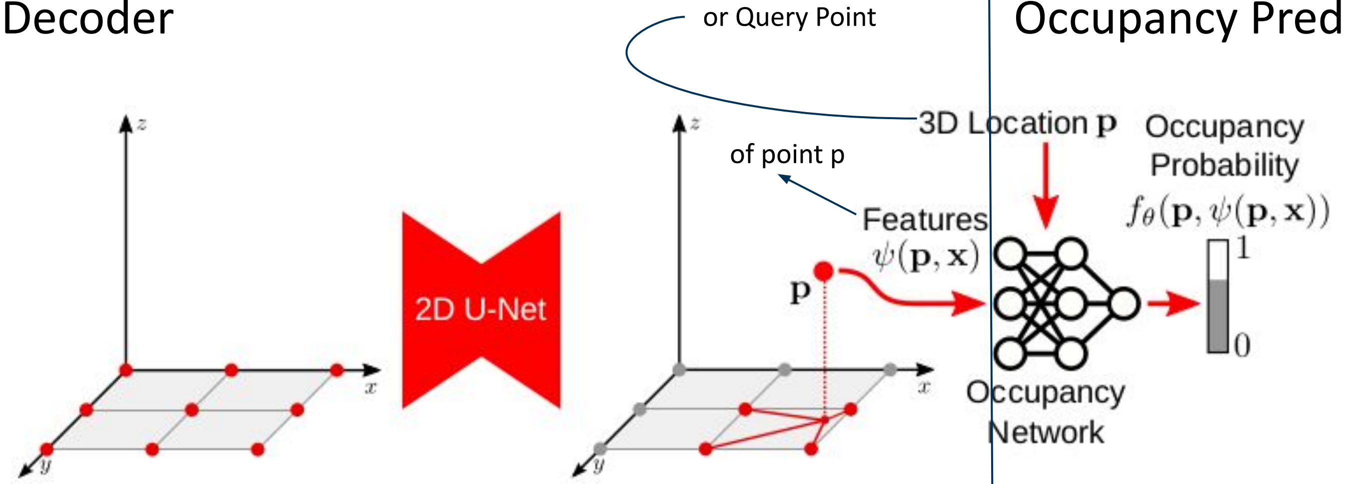
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

Decoder



1. Process the Feature Plane (space)
   -> 2D U-Net
   +Integrate global information
   +Translation equivariance

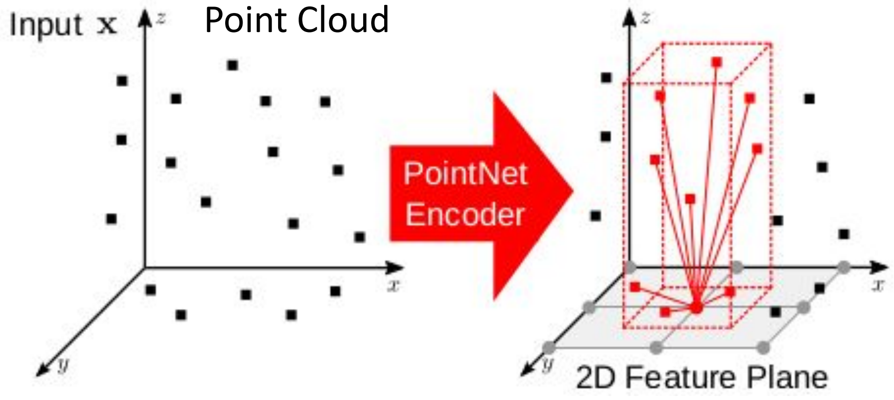
Occupancy Prediction

1. Query a 3D point
   -> Use interpolation
   -> Predict the
   Occupancy Prob.

2. Approach

2D Method

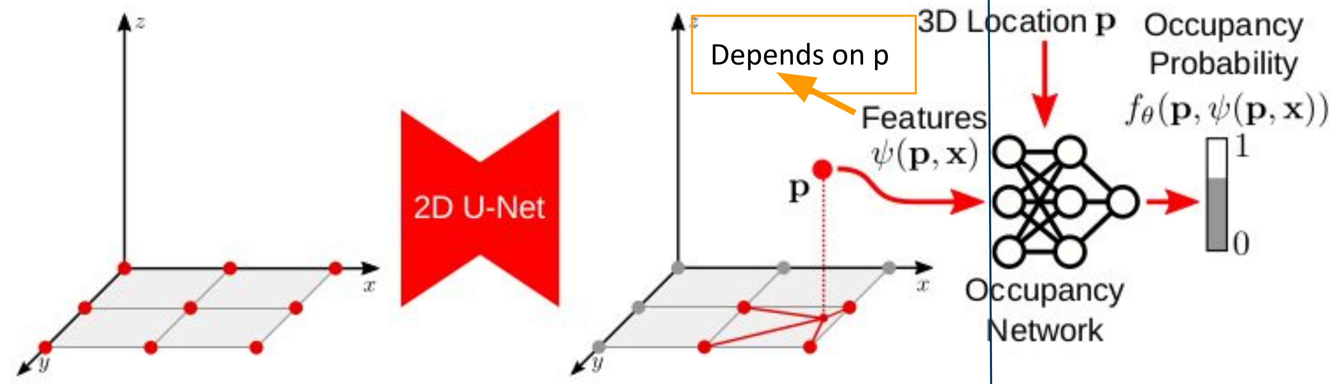
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

Decoder



1. Process the Feature Plane (space)
   -> 2D U-Net
   +Integrate global information
   +Translation equivariance

Occupancy Prediction

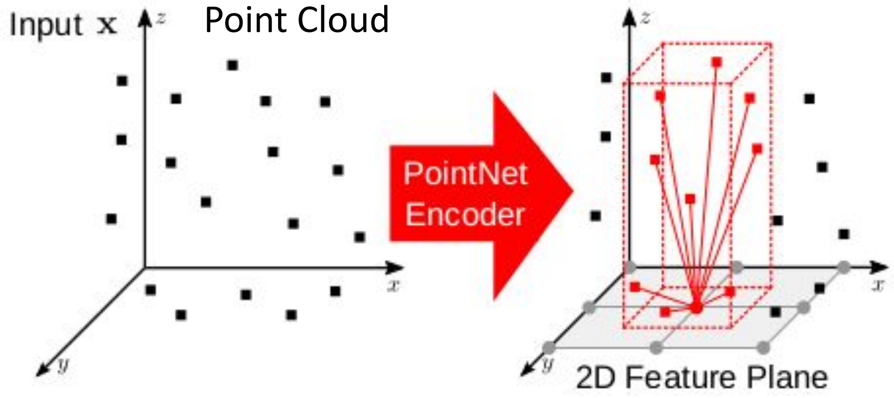
1. Query a 3D point
   -> Use interpolation
   -> Predict the Occupancy Prob.

2. Approach

2D Method

Multiple Canonical Planes

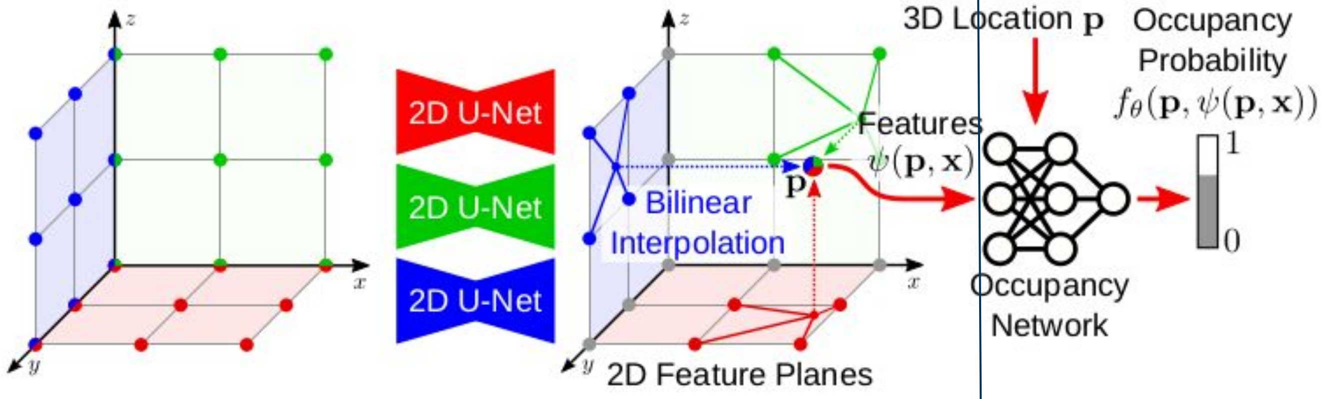
Encoder



1. Refine features
   -> 2D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

Decoder



1. Process the Feature Plane (space)
   -> 2D U-Net
   +Integrate global information
   +Translation equivariance

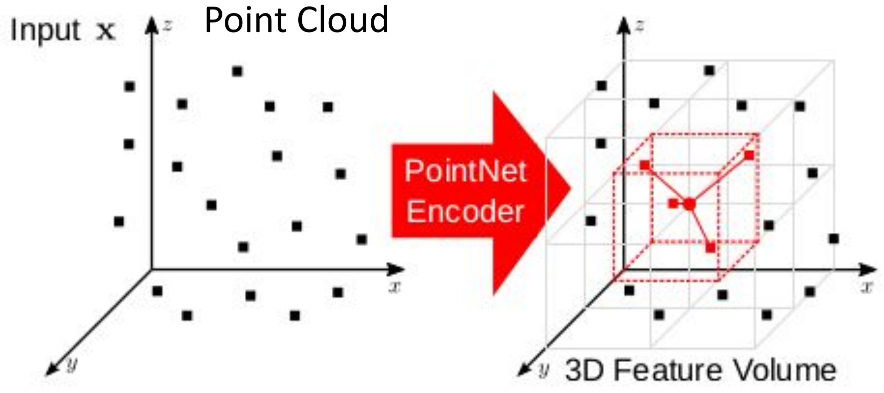
Occupancy Prediction

1. Query a 3D point
   -> Use interpolation
   -> Predict the Occupancy Prob.

2. Approach

3D Method - Volumetric Repr.

Encoder



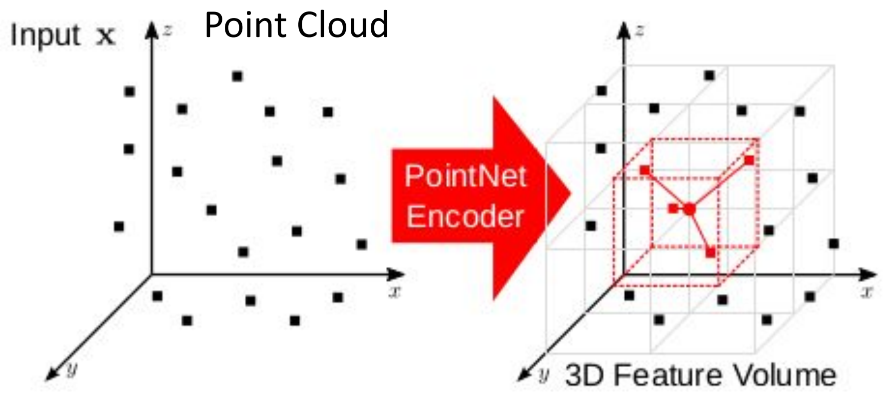
1. Refine features
   -> 3D PointNet
   +Preserves local information

2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

2. Approach

3D Method - Volumetric Repr.

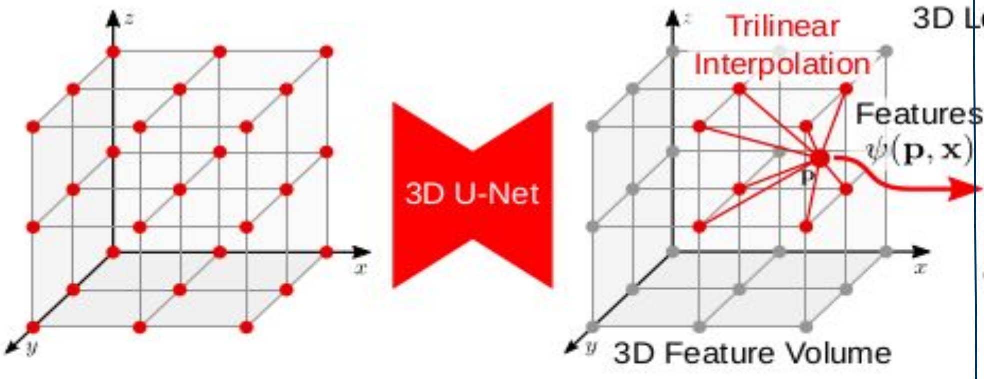
Encoder



1. Refine features
   -> 3D PointNet
   +Preserves local information

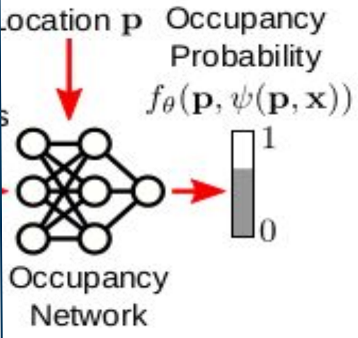
2. Project to canonical plane
   -> Aggregate local neighbors
   +Preserves local information
   +Not depend on a global frame

Decoder



1. Process the Feature Plane (space)
   -> 3D U-Net
   +Integrate global information
   +Translation equivariance

Occupancy Prediction

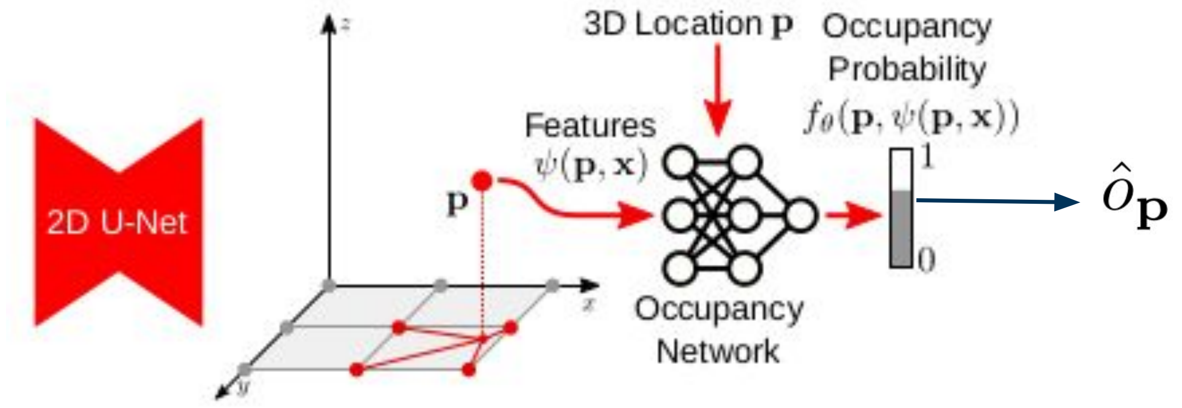


1. Query a 3D point
   -> Use interpolation
   -> Predict the
   Occupancy Prob.

2. Approach

Training

- Train the Occupancy network
  - Sample query points p from 3D objects using the train set



<u>Binary Cross-Entropy Loss</u>

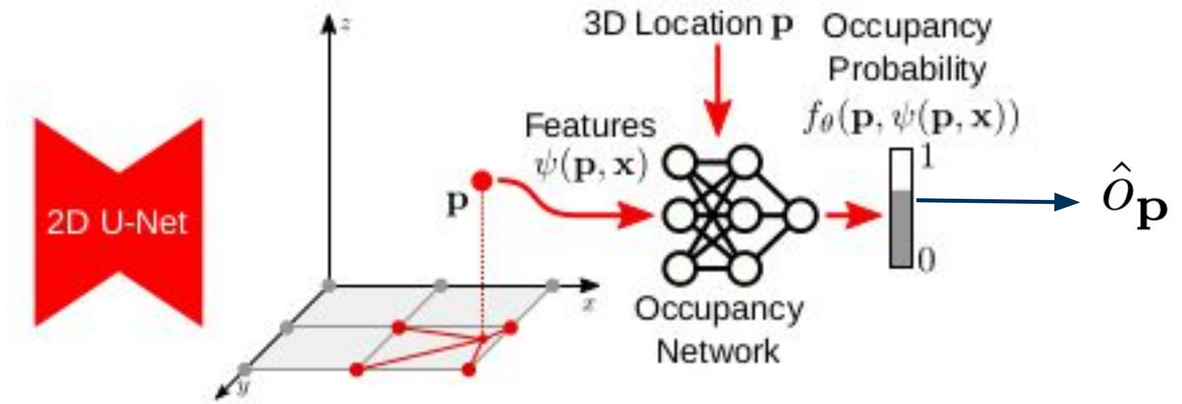
$$\mathcal{L}(\hat{o}_{\mathbf{p}}, o_{\mathbf{p}}) = -[o_{\mathbf{p}} \cdot \log(\hat{o}_{\mathbf{p}}) + (1 - o_{\mathbf{p}}) \cdot \log(1 - \hat{o}_{\mathbf{p}})]$$

True occupancy prob. Predicted occupancy prob.

2. Approach

Training

- Train the Occupancy network
  - Sample query points p from 3D objects using the train set
  - The Encoder is pre-trained / task-specific: classf. & segm.
    - feature space is ready to use (ψ)



<u>Binary Cross-Entropy Loss</u>

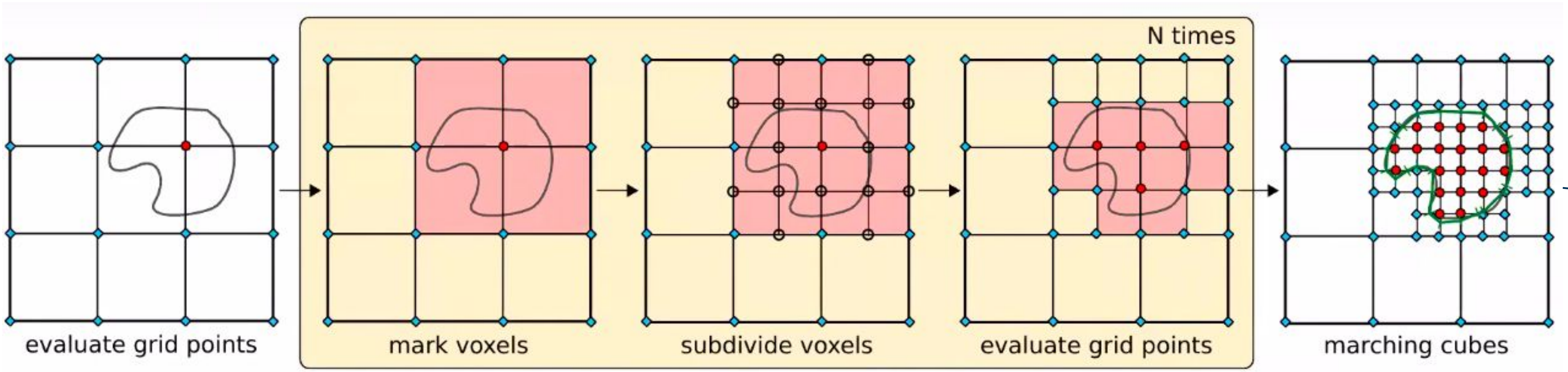
$$\mathcal{L}(\hat{o}_\mathbf{p}, o_\mathbf{p}) = -[o_\mathbf{p} \cdot \log(\hat{o}_\mathbf{p}) + (1 - o_\mathbf{p}) \cdot \log(1 - \hat{o}_\mathbf{p})]$$

True occupancy prob. Predicted occupancy prob.

2. Approach

Rendering - Generate a 3D Mesh

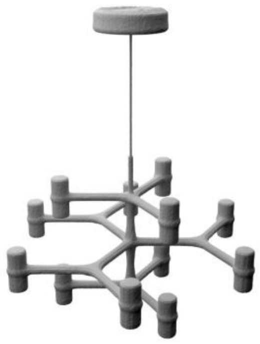
Multiresolution IsoSurface Extraction (MISE)



Simplified example - 2D “projection”

Triangular Mesh

Expected 3D example



1. Partition the 3D space
   - Build octree incrementally

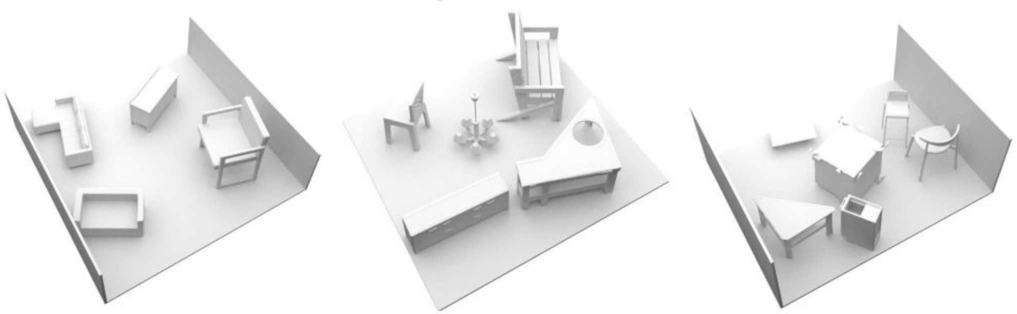
2. Query the occupancy network

3. Results

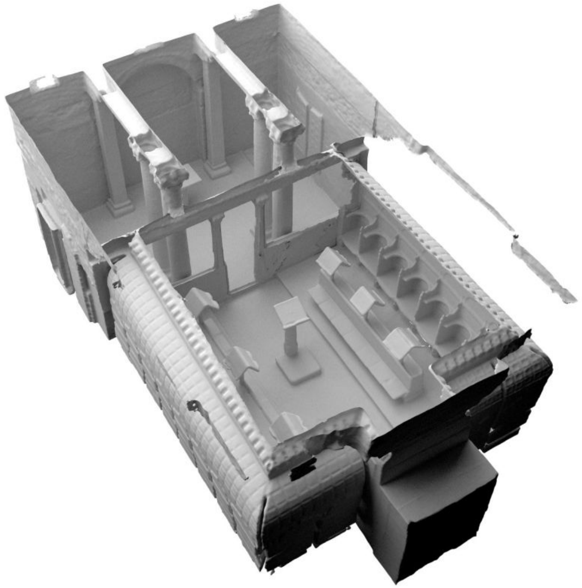
Datasets - 4 in total

ShapeNet

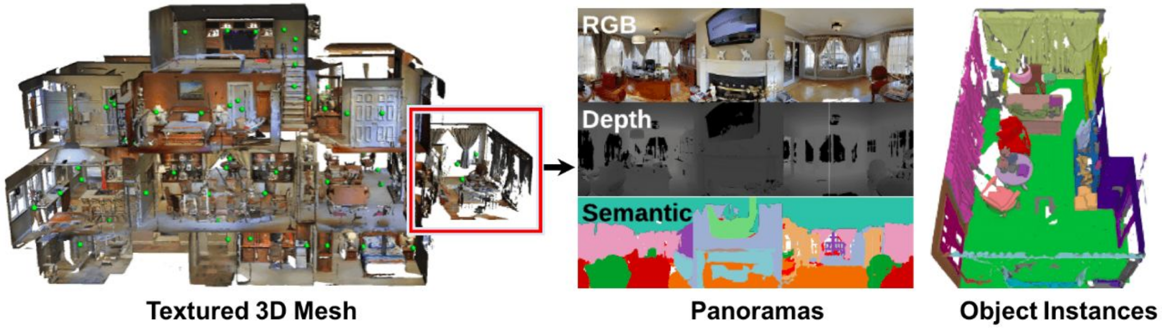
Synthetic Indoor Scene Dataset



Matterport3D



ScanNet v2



3. Results

Metrics

-> Volumetric IoU

$$\mathrm{IoU}(\mathcal{M}_{\mathrm{pred}}, \mathcal{M}_{\mathrm{GT}}) \equiv \frac{|\mathcal{M}_{\mathrm{pred}} \cap \mathcal{M}_{\mathrm{GT}}|}{|\mathcal{M}_{\mathrm{pred}} \cup \mathcal{M}_{\mathrm{GT}}|}$$

-> Chamfer Distance

$$\mathrm{Accuracy}(\mathcal{M}_{\mathrm{pred}}|\mathcal{M}_{\mathrm{GT}}) \equiv \frac{1}{|\partial\mathcal{M}_{\mathrm{pred}}|} \int_{\partial\mathcal{M}_{\mathrm{pred}}} \min_{\mathbf{q}\in\partial\mathcal{M}_{\mathrm{GT}}} \|\mathbf{p}-\mathbf{q}\| d\mathbf{p}$$

$$\mathrm{Completeness}(\mathcal{M}_{\mathrm{pred}}|\mathcal{M}_{\mathrm{GT}}) \equiv \frac{1}{|\partial\mathcal{M}_{\mathrm{GT}}|} \int_{\partial\mathcal{M}_{\mathrm{GT}}} \min_{\mathbf{p}\in\partial\mathcal{M}_{\mathrm{pred}}} \|\mathbf{p}-\mathbf{q}\| d\mathbf{q}$$

$$\text{Chamfer-}L_1(\mathcal{M}_{\mathrm{pred}}, \mathcal{M}_{\mathrm{GT}}) = \frac{1}{2}\left(\mathrm{Accuracy}(\mathcal{M}_{\mathrm{pred}}|\mathcal{M}_{\mathrm{GT}}) + \mathrm{Completeness}(\mathcal{M}_{\mathrm{pred}}|\mathcal{M}_{\mathrm{GT}})\right)$$

-> Normal Consistency

$$\text{Normal-Con.}(\mathcal{M}_{\mathrm{pred}}, \mathcal{M}_{\mathrm{GT}}) \equiv \frac{1}{2\,|\partial\mathcal{M}_{\mathrm{pred}}|} \int_{\partial\mathcal{M}_{\mathrm{pred}}} |\langle n(\mathbf{p}), n(\mathrm{proj}_2(\mathbf{p}))\rangle| d\mathbf{p} + \frac{1}{2\,|\partial\mathcal{M}_{\mathrm{GT}}|} \int_{\partial\mathcal{M}_{\mathrm{GT}}} |\langle n(\mathrm{proj}_1(\mathbf{q})), n(\mathbf{q})\rangle| d\mathbf{q}$$

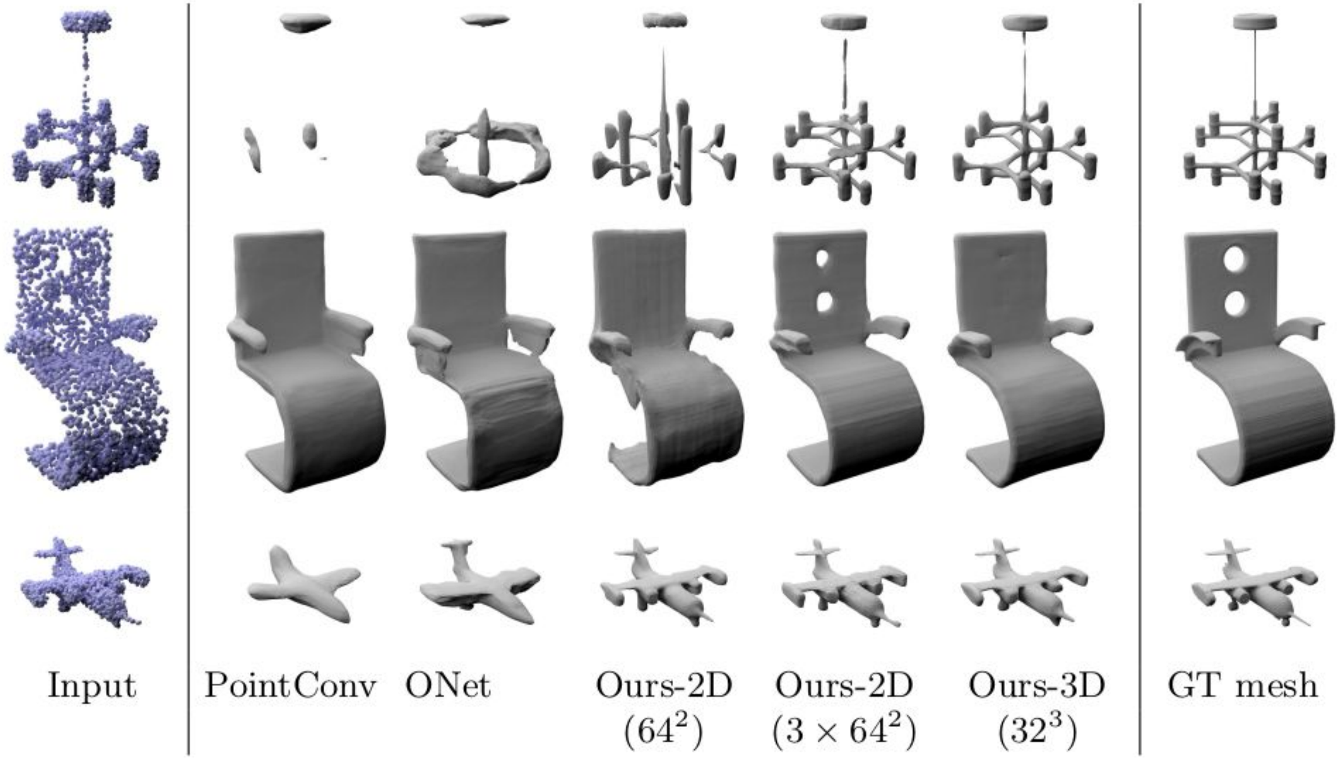
-> F-Score

$$\text{F-Score} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

3. Results

Object-Level 3D Reconstruction

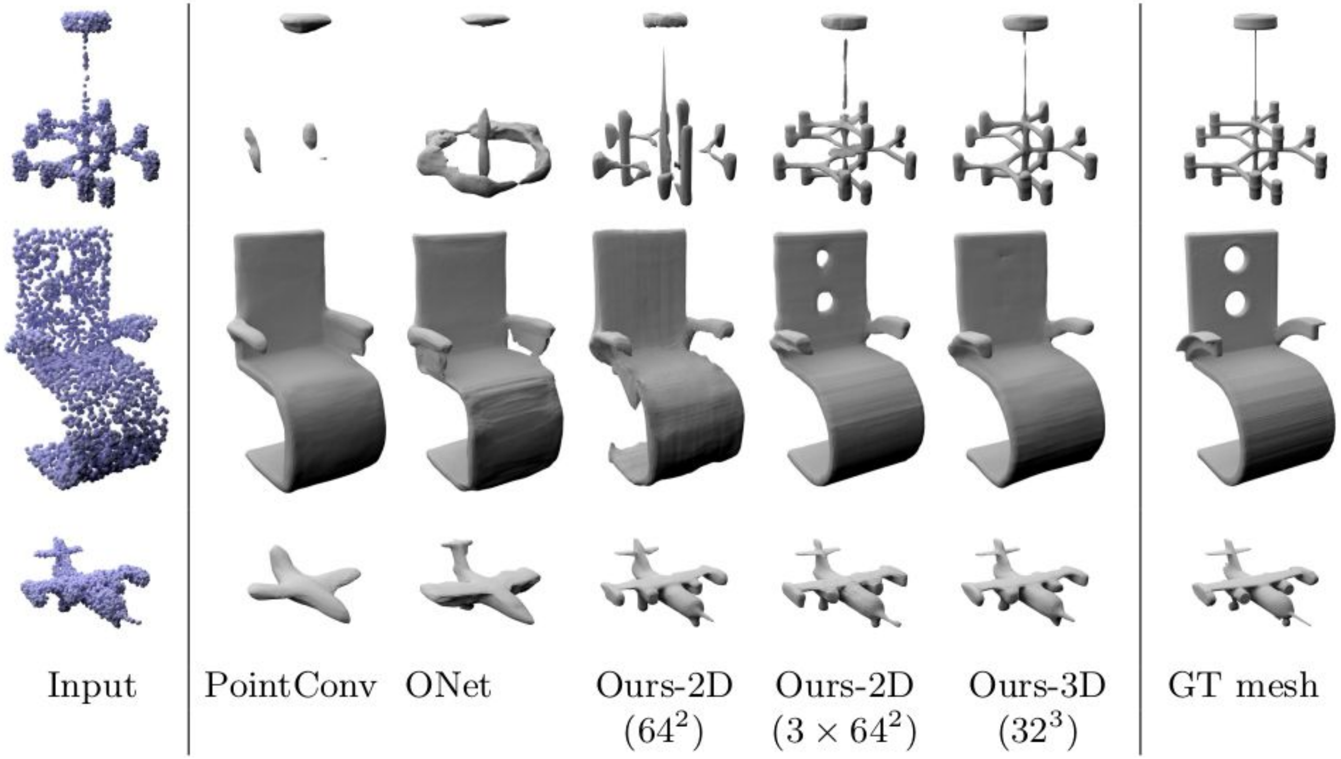
Reconstruction from Point Clouds



3. Results

Object-Level 3D Reconstruction

Reconstruction from Point Clouds

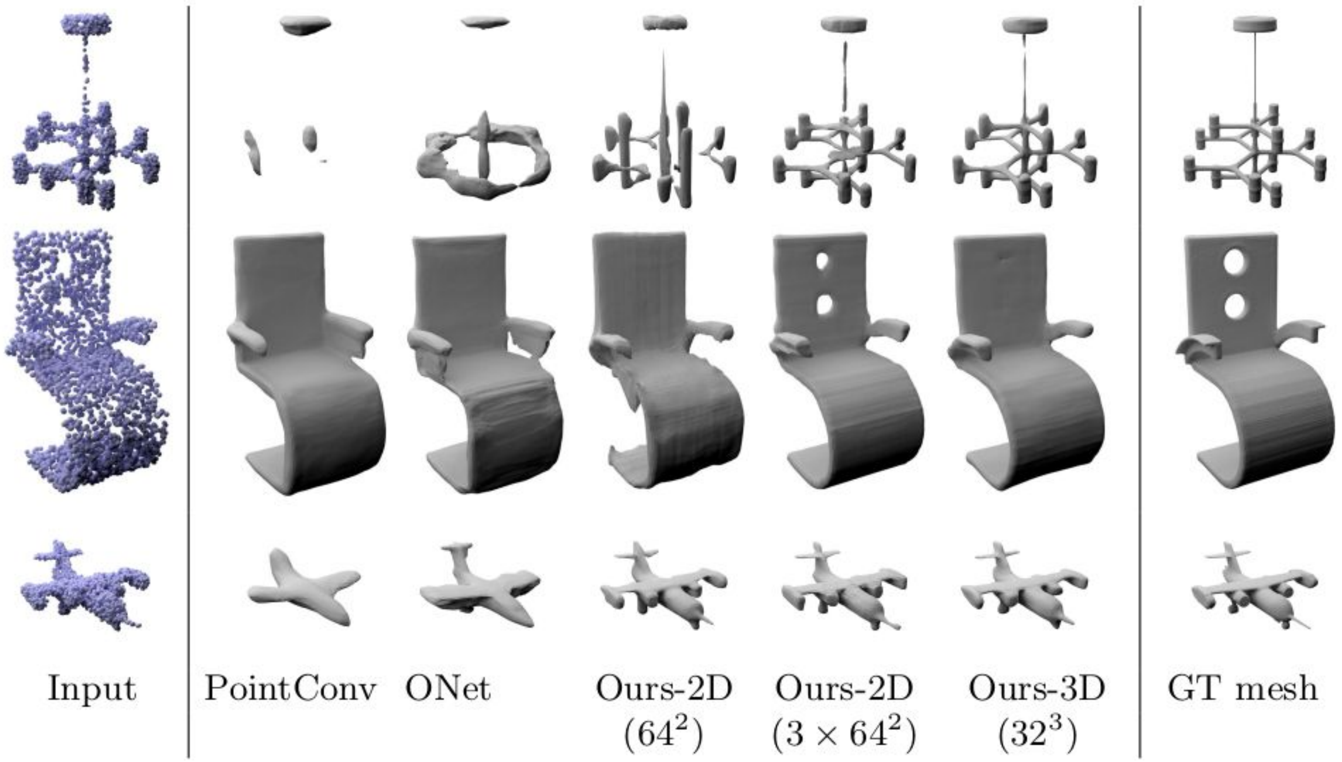


- Baseline: PointConv
  - PointNet++ encoder
    - Remove canonical planes
  - Instead of the 2D decoder and interpolation
    - Gaussian regression

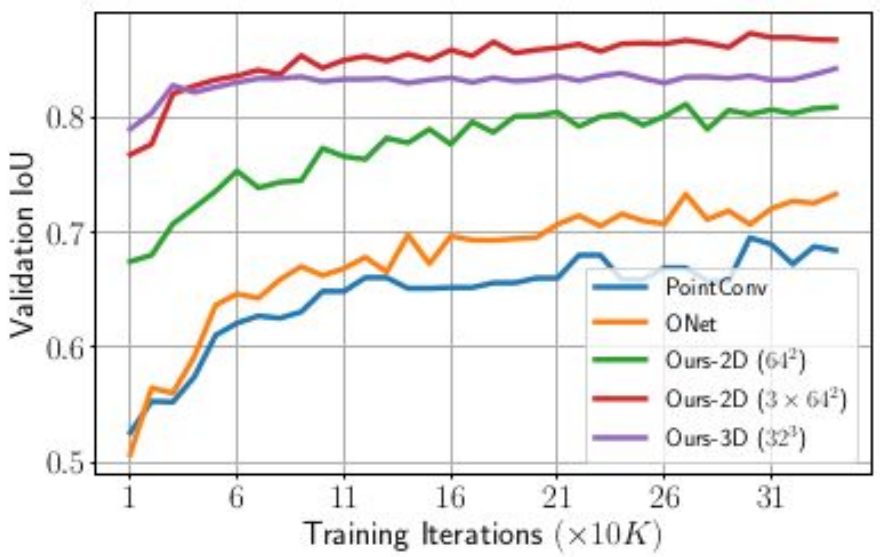
3. Results

Object-Level 3D Reconstruction

Reconstruction from Point Clouds



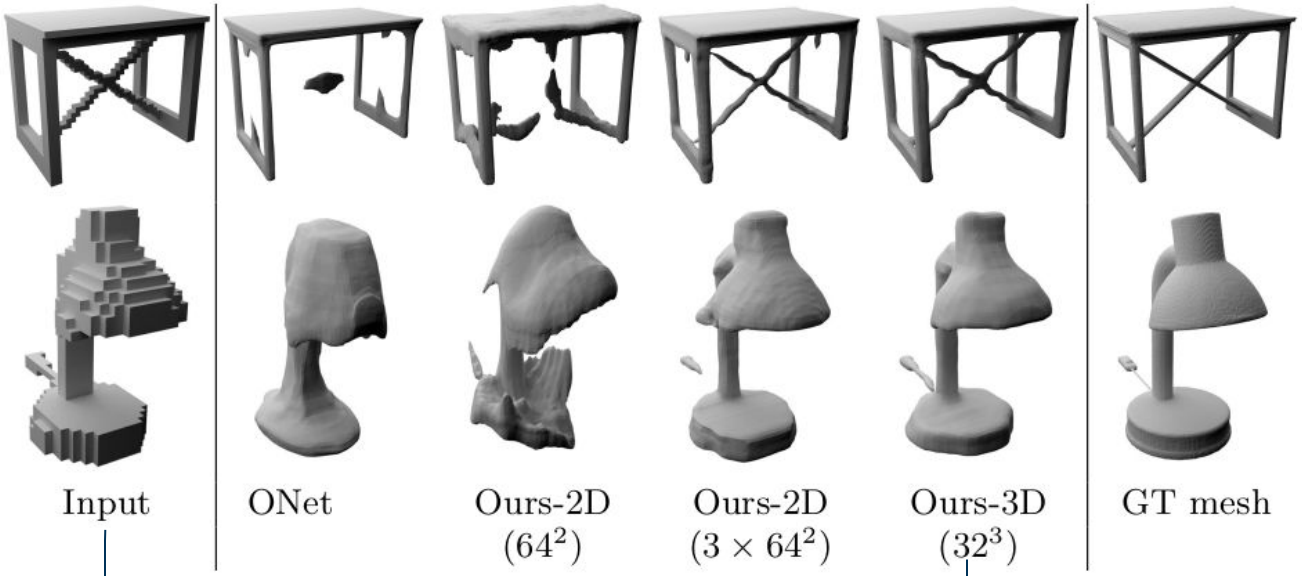
- Convolutional Occupancy Networks
  - Reconstruction of complex shapes
  - Faster convergence



3. Results

Object-Level 3D Reconstruction

Voxel Super-Resolution



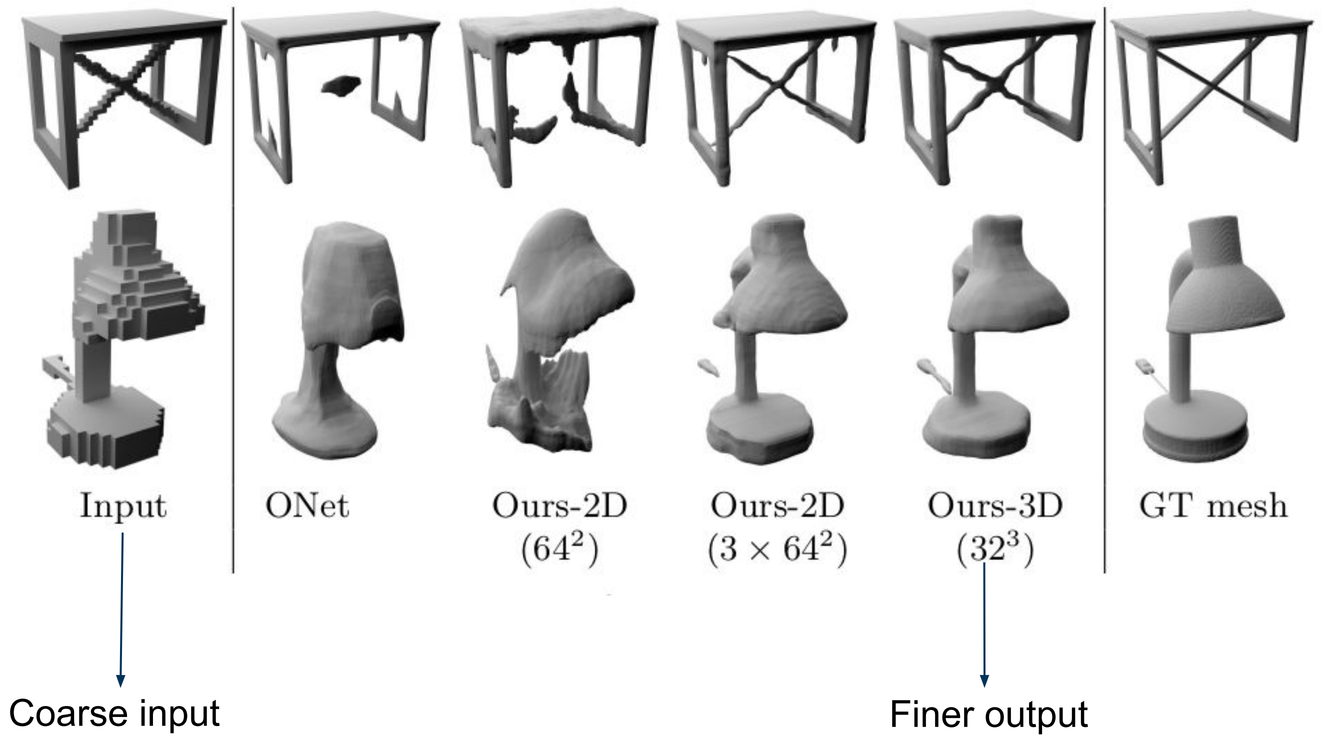
Coarse input

Finer output

3. Results

Object-Level 3D Reconstruction

Voxel Super-Resolution

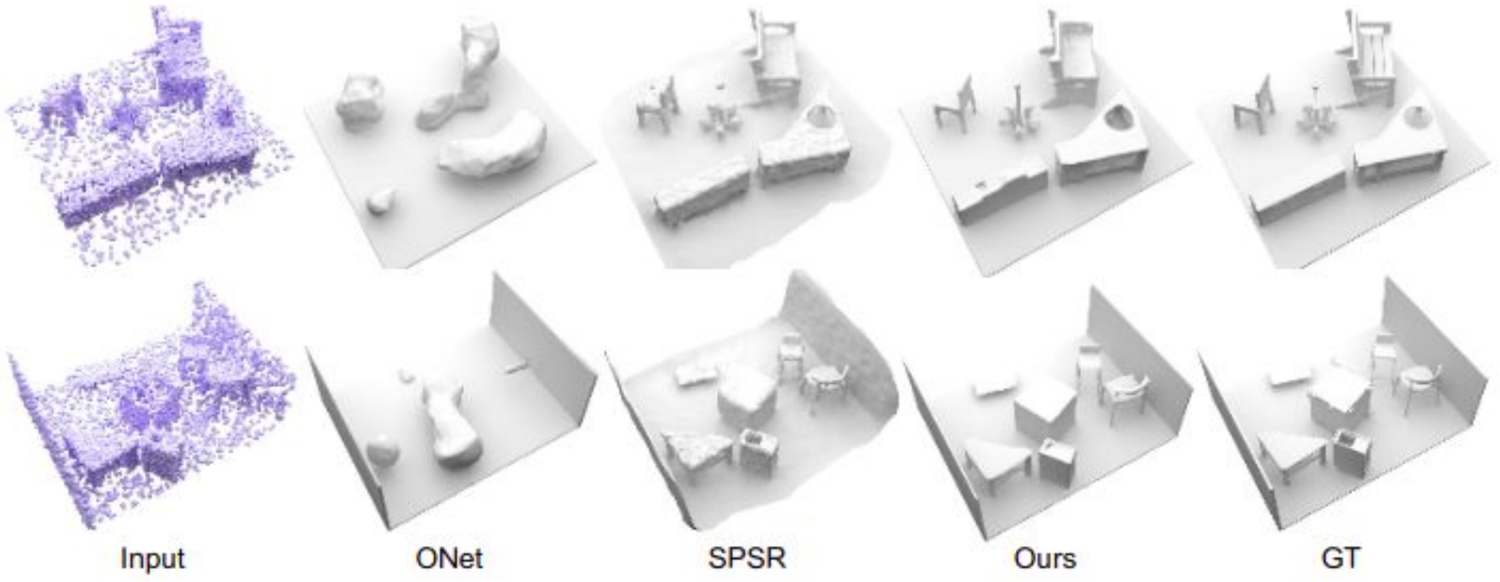


- Convolutional Occupancy Networks
  - Recover high-resolution details
  - Three planes perform similar to the volumetric encoder while consuming 37% of the GPU
  - The single-plane approach is not powerful

3. Results

Scene-Level Reconstruction

Synthetic dataset evaluation

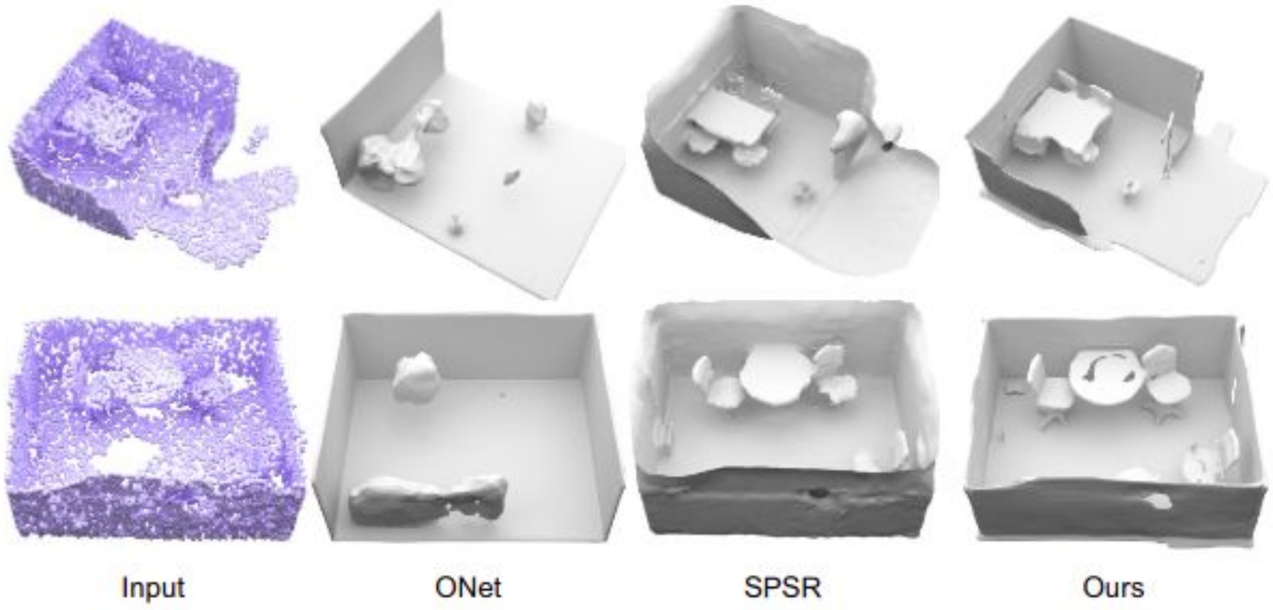


- Occupancy Networks
  - Can not scale to bigger scenes
- SPSR
  - Requires the normals of the points
  - Noisy results

3. Results

Scene-Level Reconstruction

Trained on synthetic and transfer to ScanNet v2

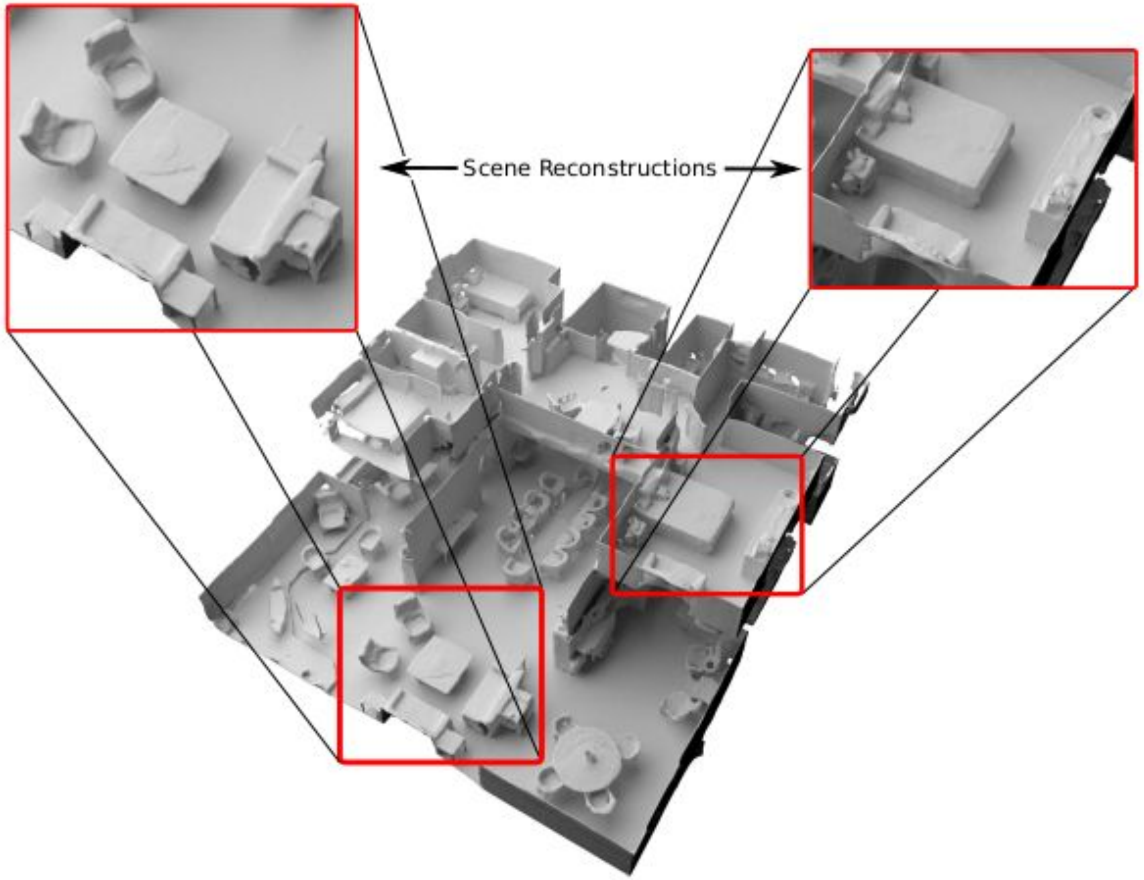


-> All previous methods mostly fail on this task

3. Results

Large-Scale Reconstruction

Trained on synthetic and transfer to Matterport3D



-> Trained on synthetic crops

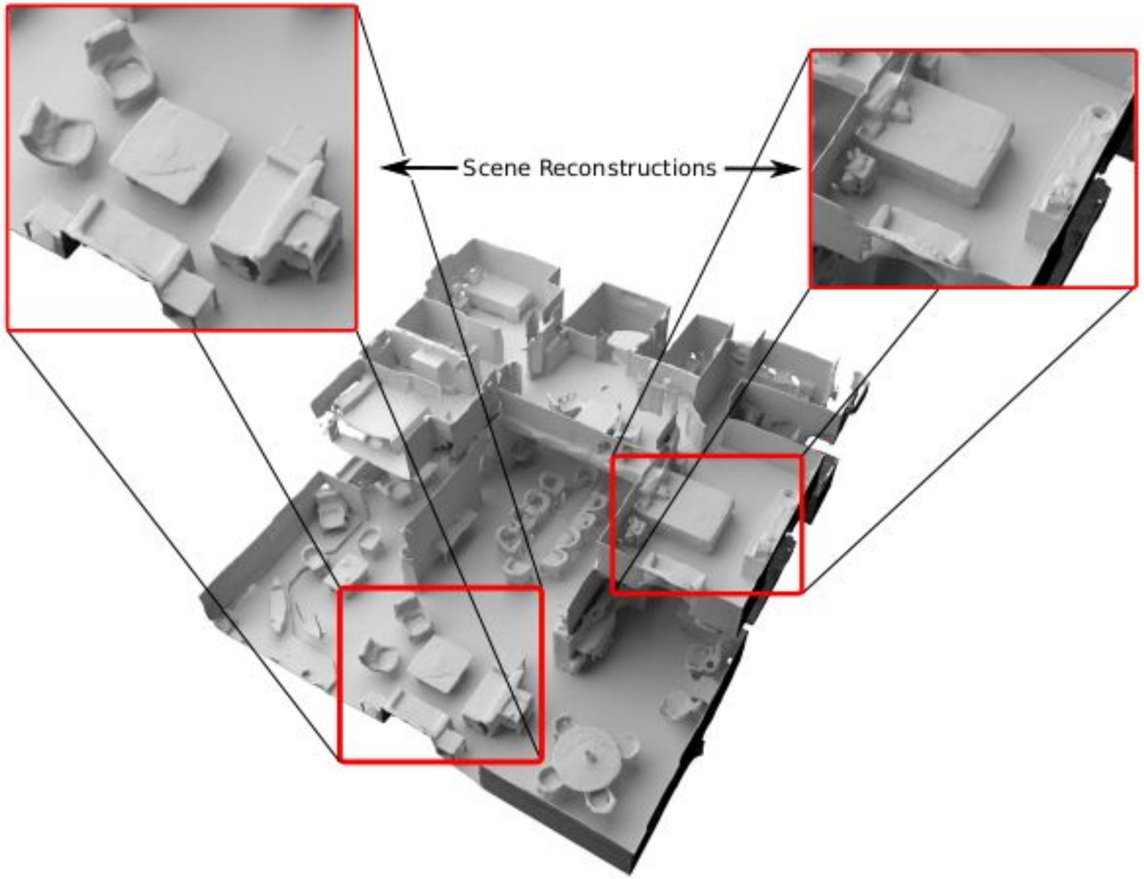
-> During inference, use sliding window

-> 3D CNN performed the best

3. Results

Large-Scale Reconstruction

Trained on synthetic and transfer to Matterport3D



-> Trained on synthetic crops

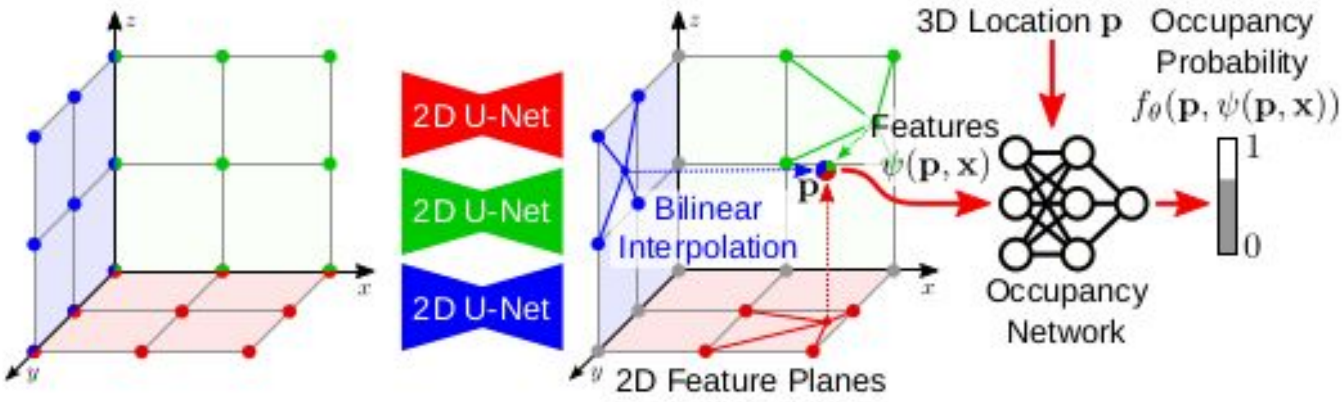
-> During inference, use sliding window

-> 3D CNN performed the best

-> The authors do not explain how to merge the patches
- What happens with the artifacts of the overlapping windows?

4. Personal comments

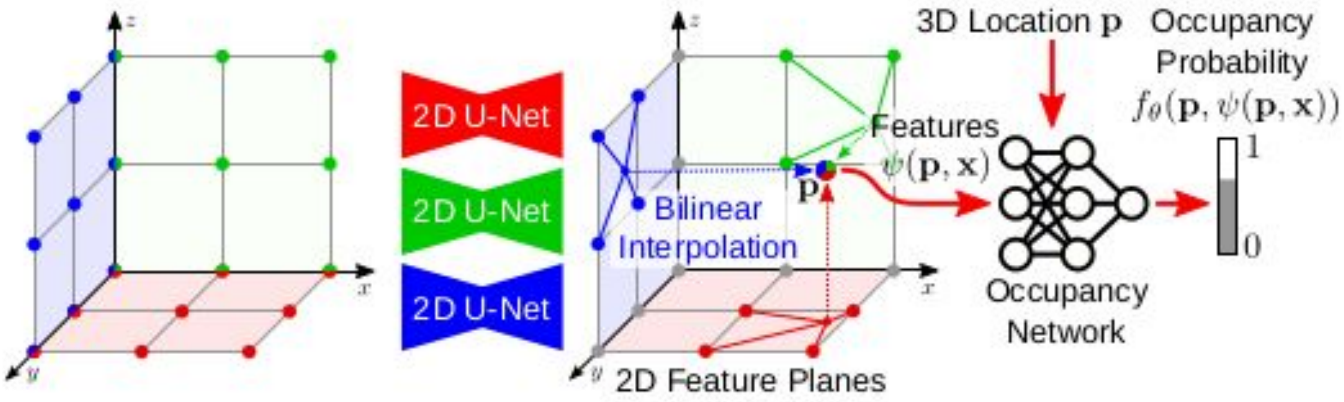
1. Shared 2D U-Nets?



4. Personal comments

1. Shared 2D U-Nets?

2. Use shallow Neural Net instead of sum?

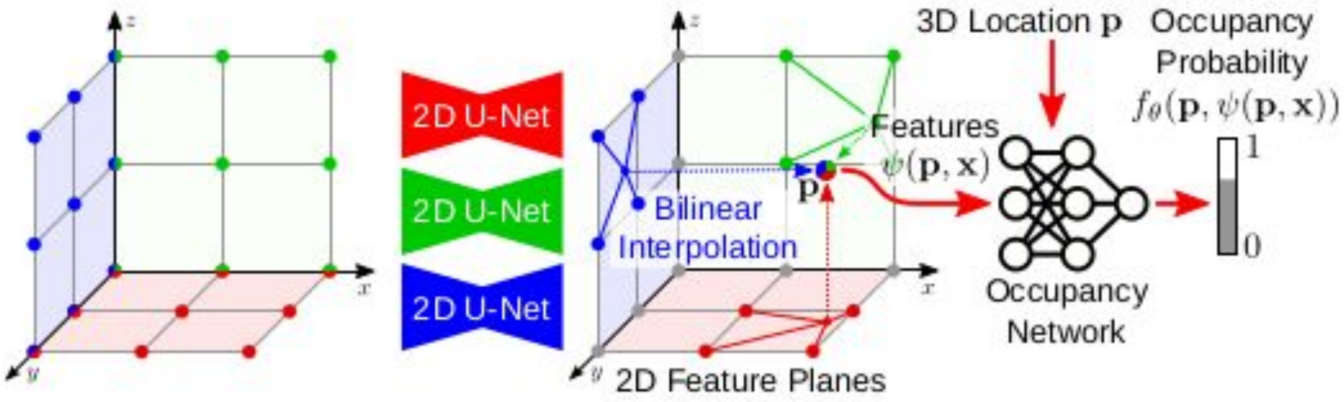


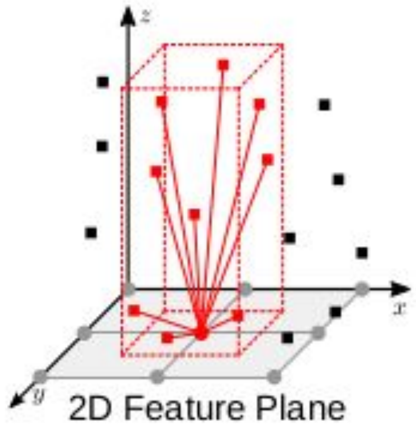
4. Personal comments

1. Shared 2D U-Nets?

2. Use shallow Neural Net instead of sum?

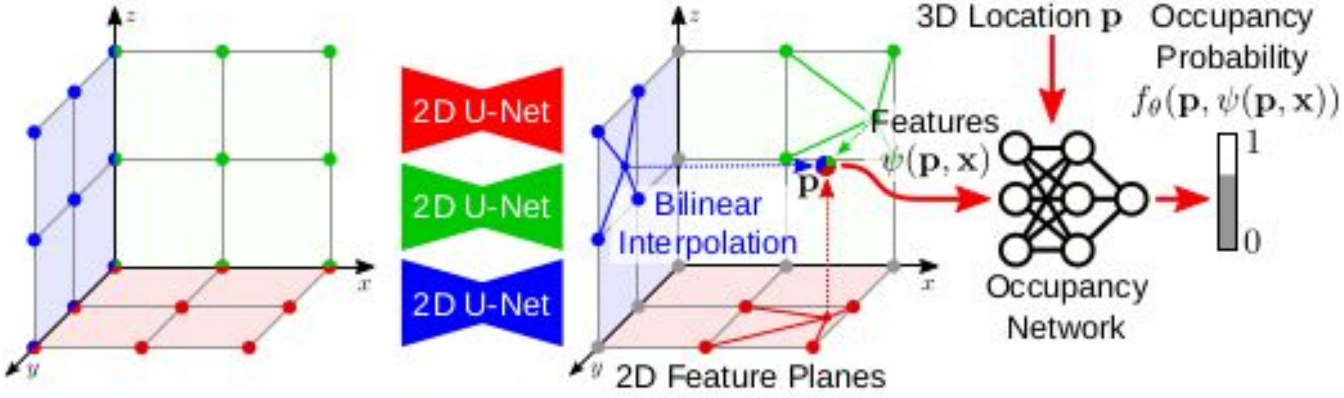
3. Average or max pooling aggregation?

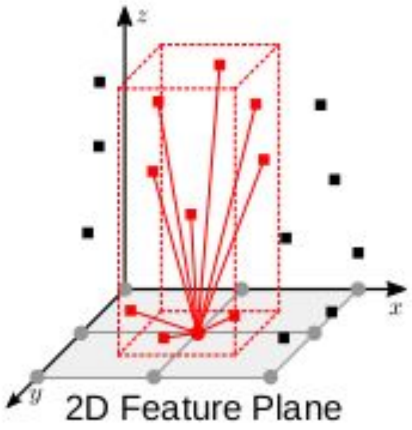




4. Personal comments

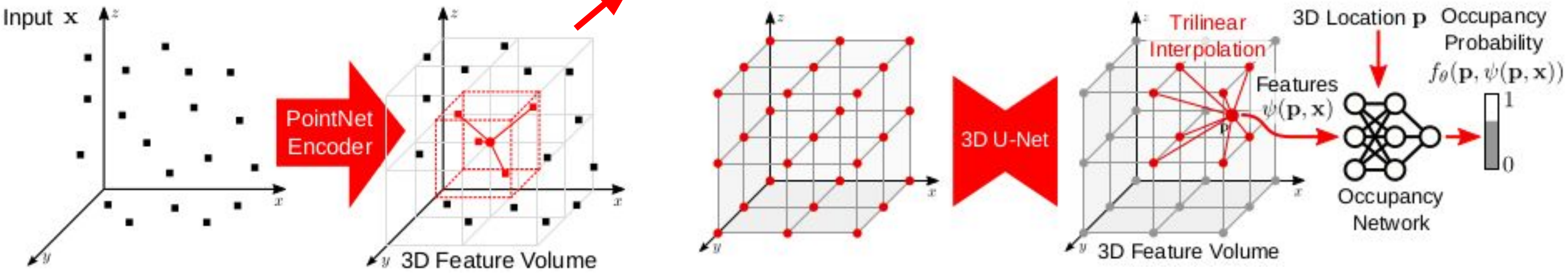
1. Shared 2D U-Nets?
2. Use shallow Neural Net instead of sum?
3. Average or max pooling aggregation?





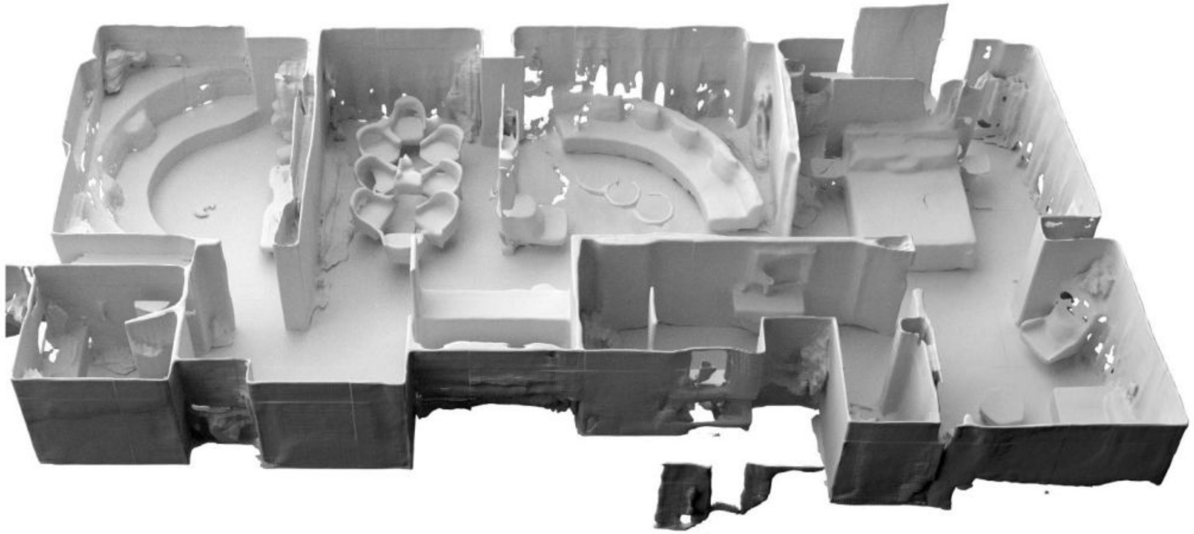
4. Formulate as End to End Learning?

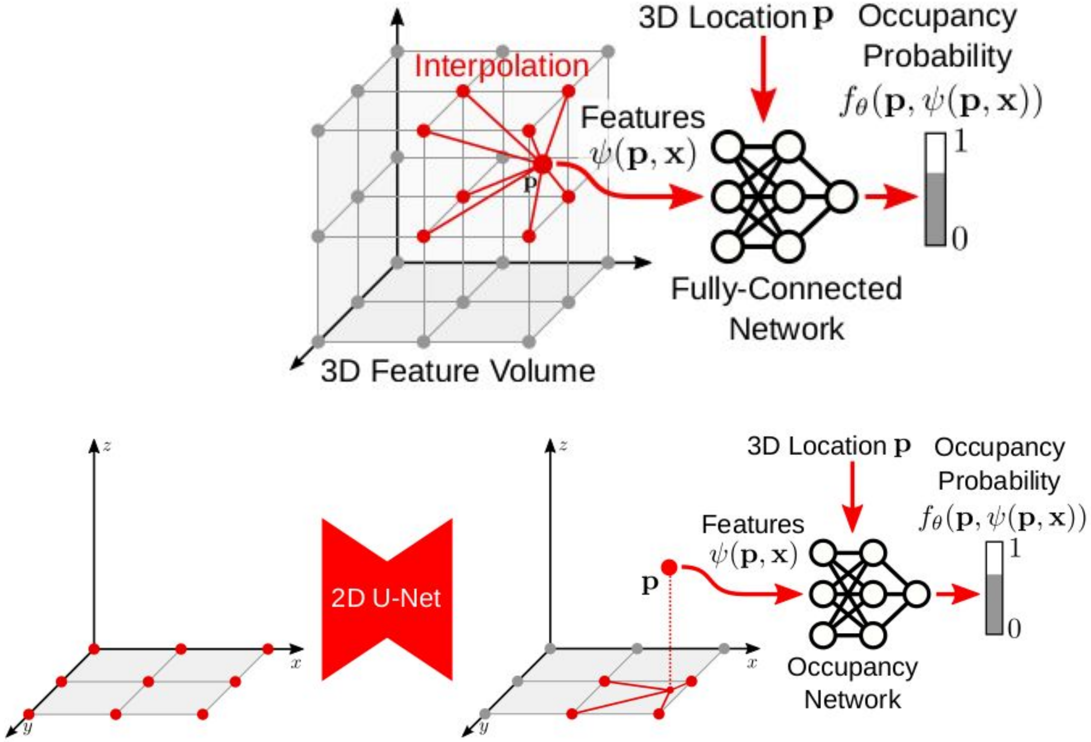
Pre-trained



5. Summary

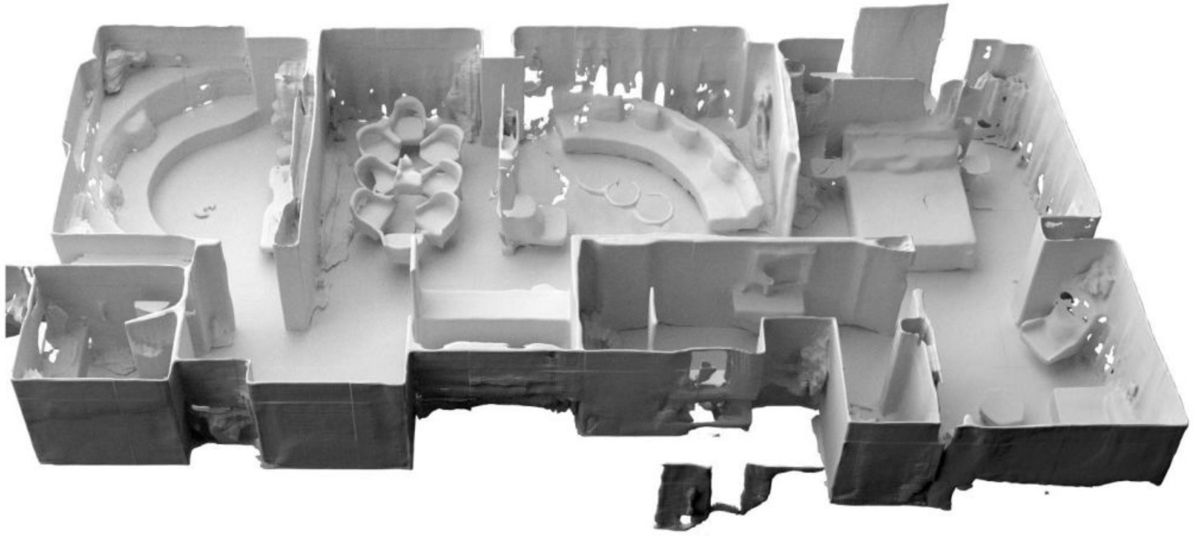
- Conv. Occupancy Networks can be transferred to noisy real large-scale scenes
- Incorporate global and local information
- Faster training

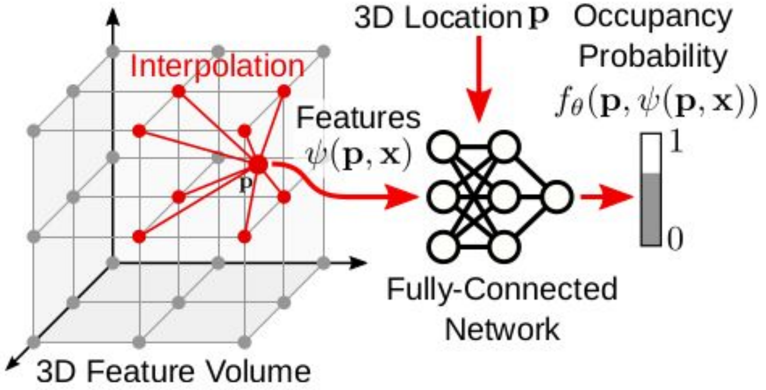


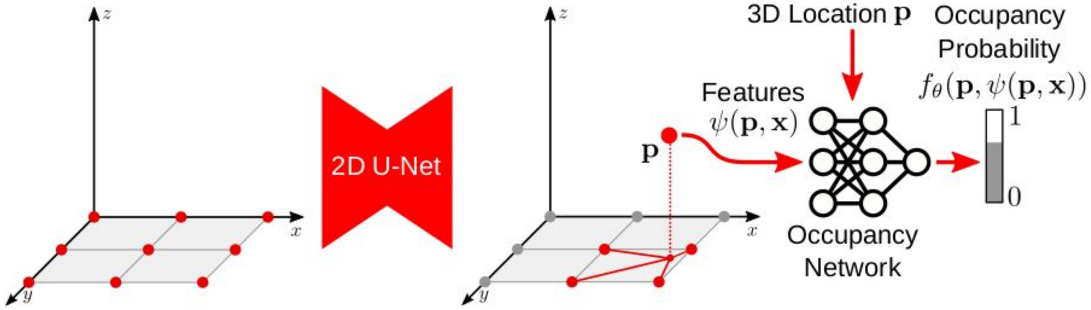


5. Summary

- Conv. Occupancy Networks can be transferred to noisy real large-scale scenes
- Incorporate global and local information
- Faster training
- **But**
  - Translation equivariant w.r.t to multiples translations of the voxel size
  - No rotation equivariant
  - Reality gap is still present







Thank you for the attention!

Questions?

References

[1] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional Occupancy Networks Computer”, in ECCV, 2020

[2] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space”, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019

[3] https://www.cvlibs.net/talks/talk_oxford_2020_04_27.pdf

[4] https://www.gamersnexus.net/gg/762-voxels-vs-vertexes-in-games

[5]https://www.researchgate.net/figure/3D-mesh-triangles-with-different-resolution-3D-Modelling-for-programmers-Available-at_fig2_322096576

[6] https://www.revopoint3d.com/point-cloud-and-3d-image/

[7] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “Shapenet: An information-rich 3d model repository”, in arXiv.org 1512.03012, 2015

[8] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, ”Scannet: Richly-annotated 3d reconstructions of indoor scenes”, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017

[9] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3D: Learning from RGB-D data in indoor environments”, in Proc. of the International Conf. on 3D Vision (3DV), 2017