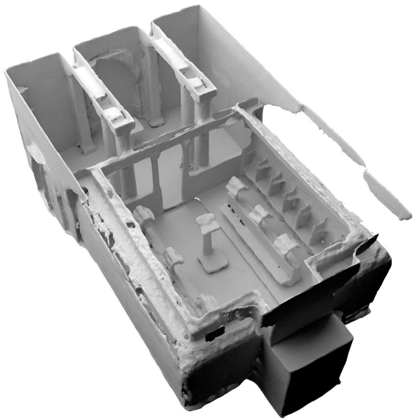
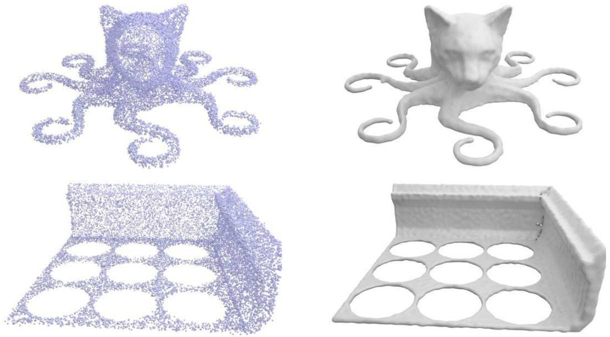
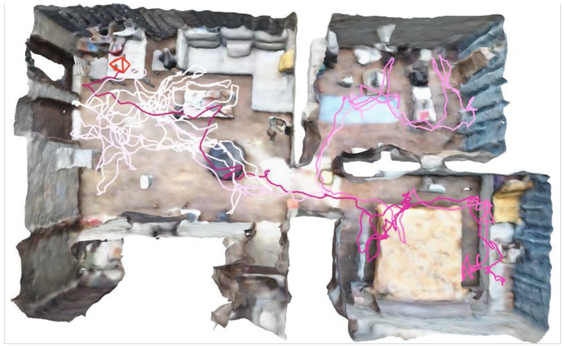
Neural Scene Representations for 3D Reconstruction







Songyou Peng

ETH Zurich & MPI

10.02.2022

Who Am I?

1992 – 2015: Live and study in China

2015 – 2017: Master in Europe

- Internship at Inria
- Master thesis at TUM

2018 – 2019: Research Engineer in Singapore

2019 – Now: **PhD Student at ETH Zurich & MPI**

- With Marc Pollefeys and Andreas Geiger
- Internship at

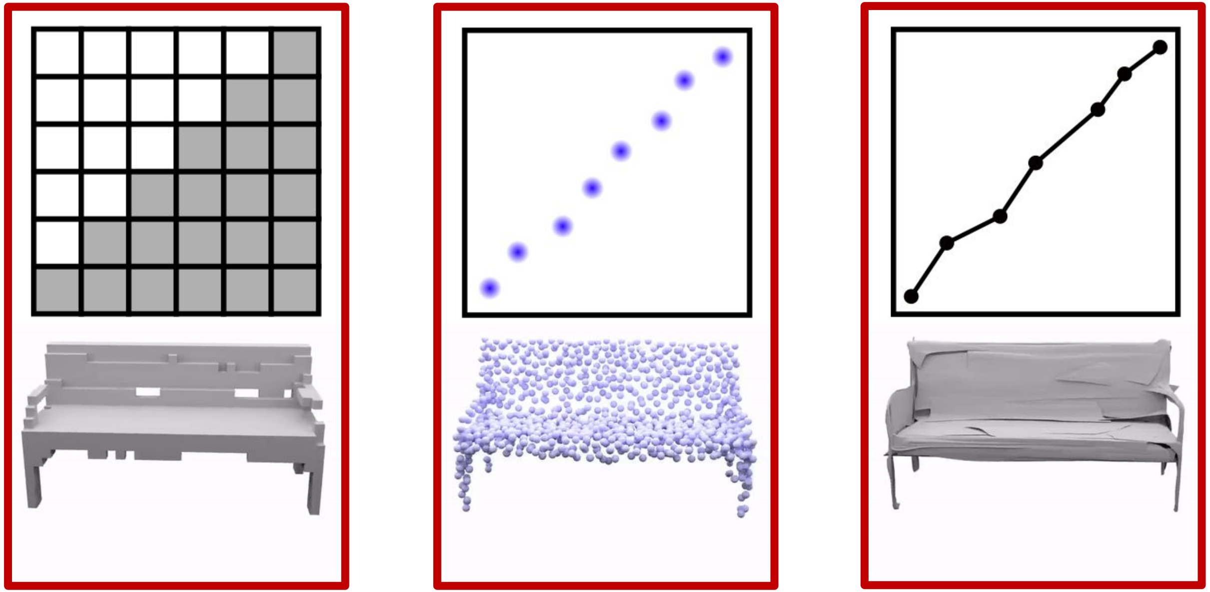
Songyou Peng

https://pengsongyou.github.io/

Agenda

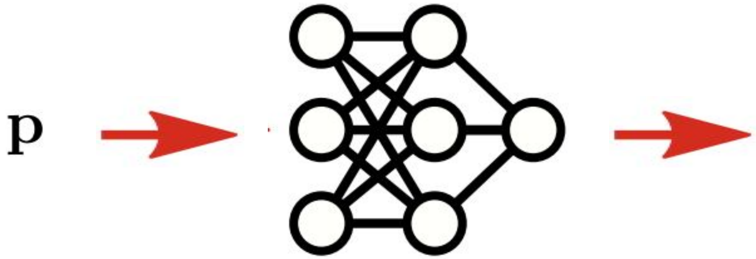
- 3D Scene Representations
- [ECCV'20] Convolutional Occupancy Networks
- [NeurIPS'21] Shape As Points: A Differentiable Poisson Solver
- [arXiv'21]: NICE-SLAM: Neural Implicit Scalable Encoding for SLAM

3D Representations



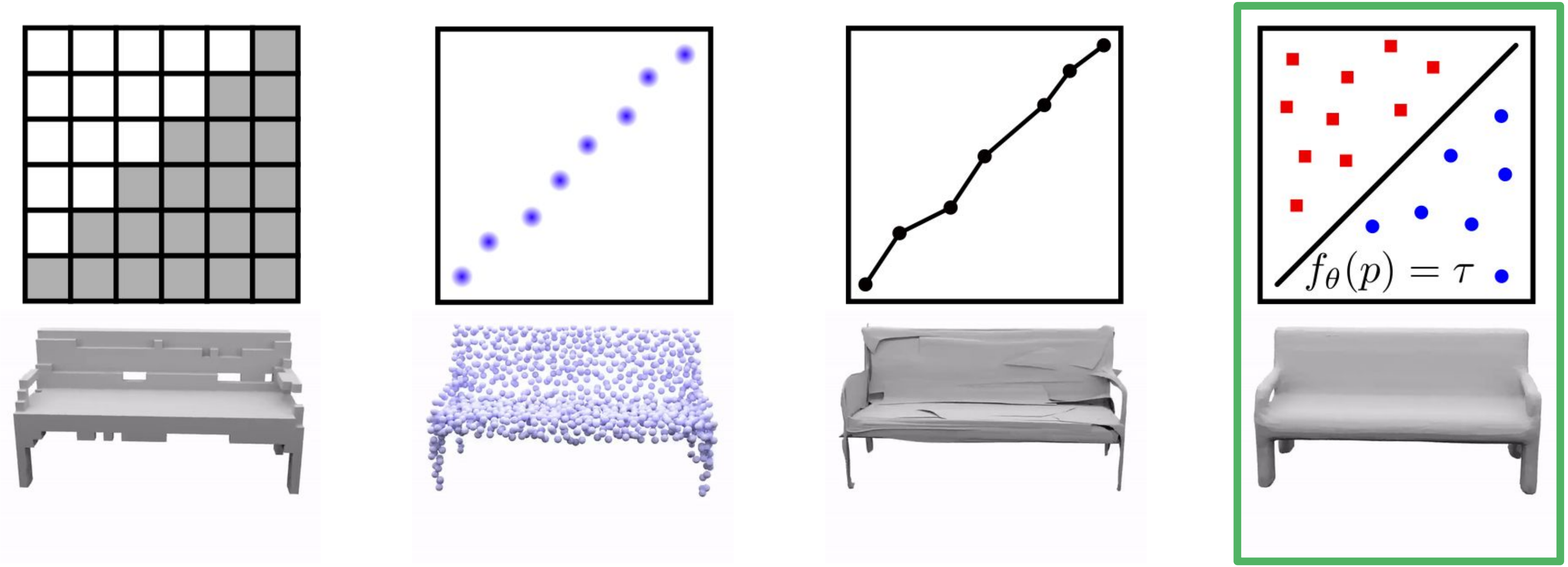
- Traditional Explicit Representations ⇒ **Discrete**

Neural Implicit Representations



SDF
Occupancy
Color
Semantics
……

3D Representations

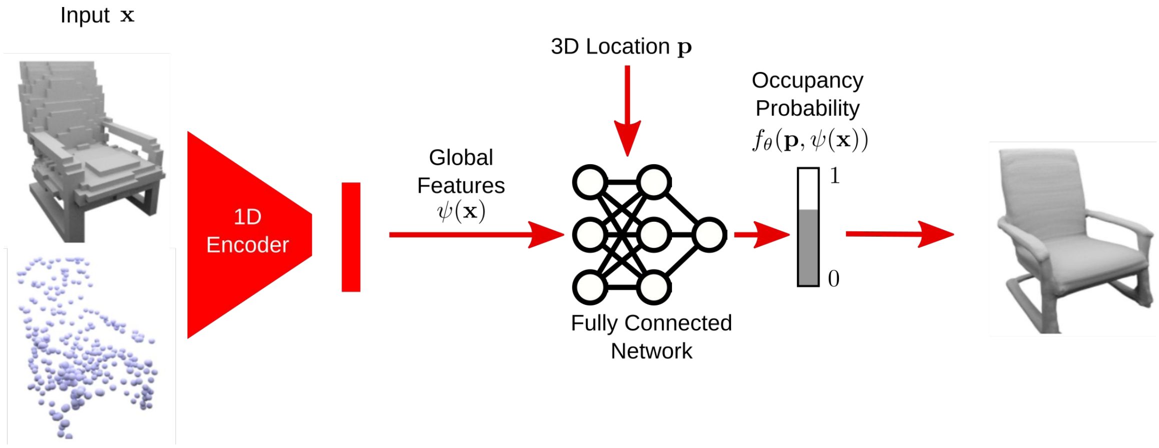


- Traditional Explicit Representations ⇒ **Discrete**
- Neural Implicit Representations ⇒ **Continuous**

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

Limitations

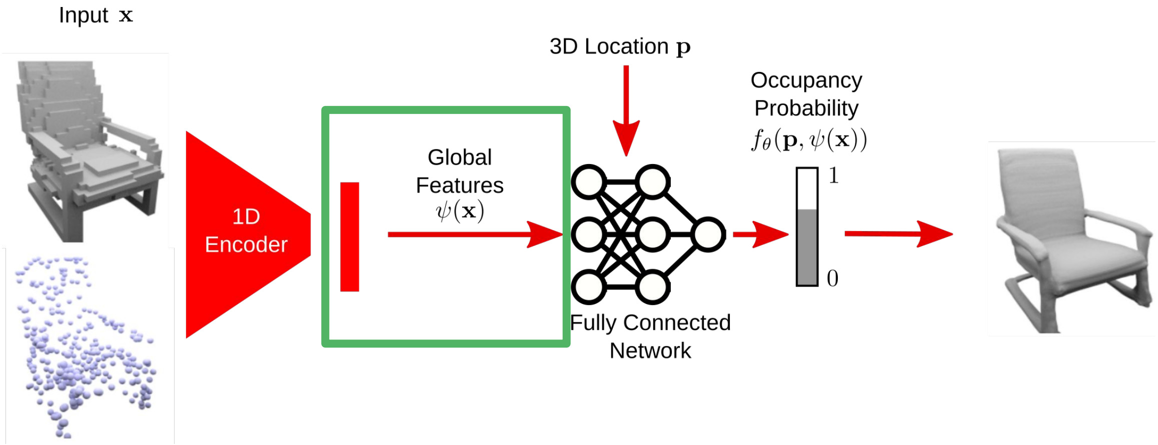
Structure of neural implicit representations:



Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

Limitations

Structure of neural implicit representations:

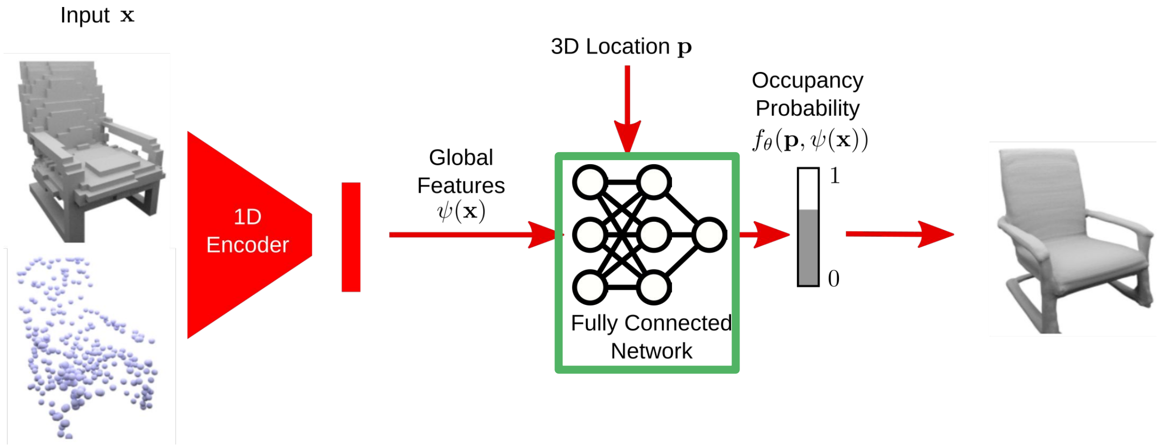


- Global latent code ⇒ overly smooth geometry

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

Limitations

Structure of neural implicit representations:



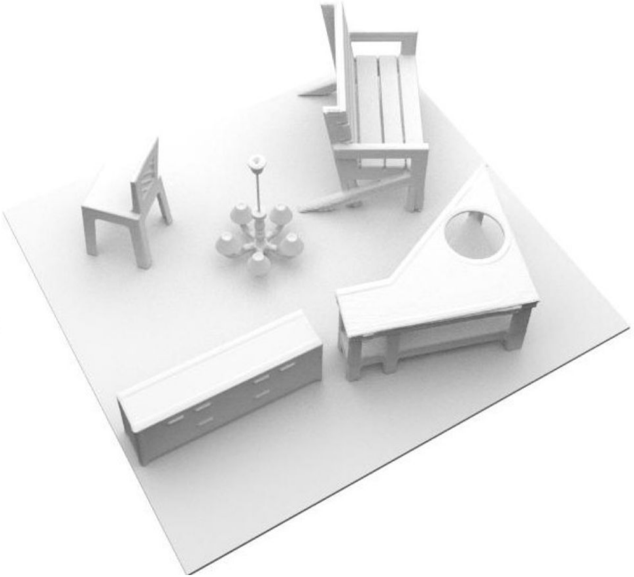
- Global latent code ⇒ overly smooth geometry
- Fully-connected architecture ⇒ no translation equivariance

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

Limitations

Implicit models work well for **simple objects** but poorly on **complex scenes**:

ONet



GT Mesh

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

How to reconstruct large-scale 3D scenes with neural implicit representations?

Convolutional Occupancy Networks

ETH zürich

ECCV'20 ONLINE
23-28 AUGUST 2020

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

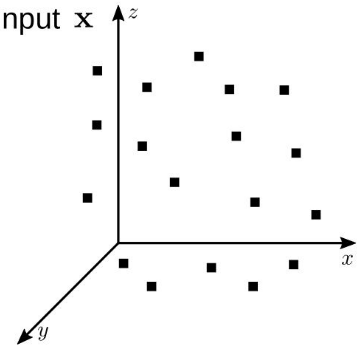
Convolutional Occupancy Networks

Songyou Peng Michael Niemeyer Lars Mescheder Marc Pollefeys Andreas Geiger

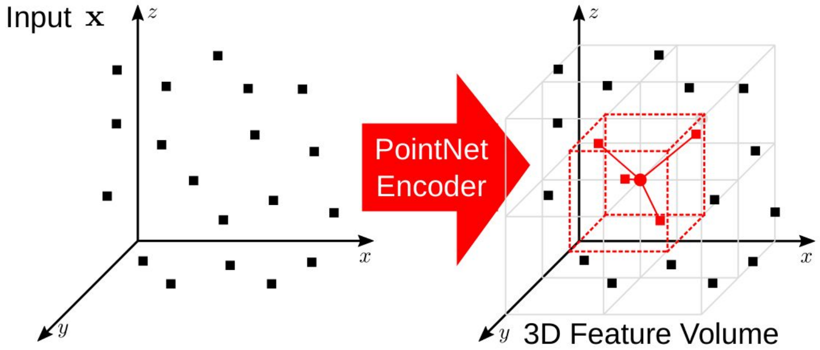
    

Main Idea

Input $\mathbf{x}$

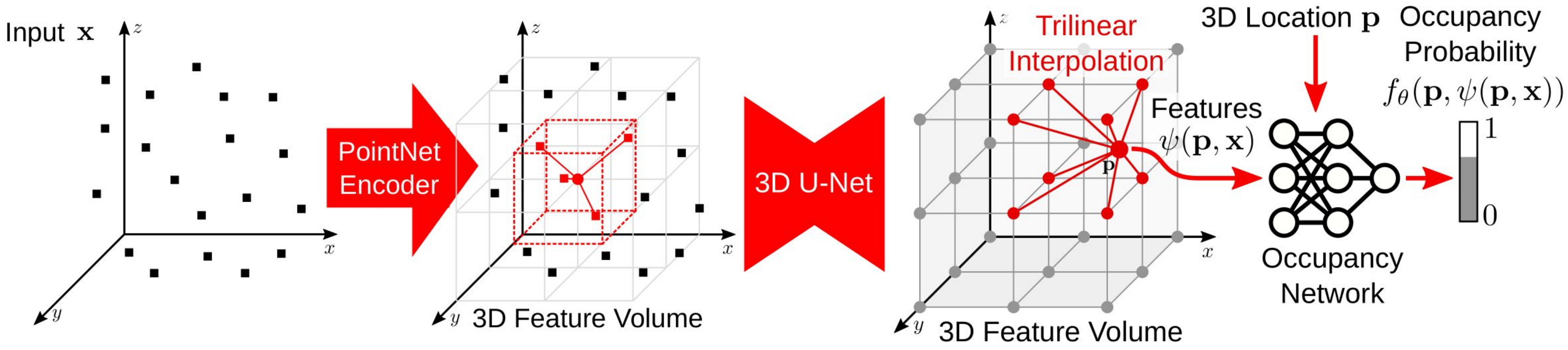


Main Idea



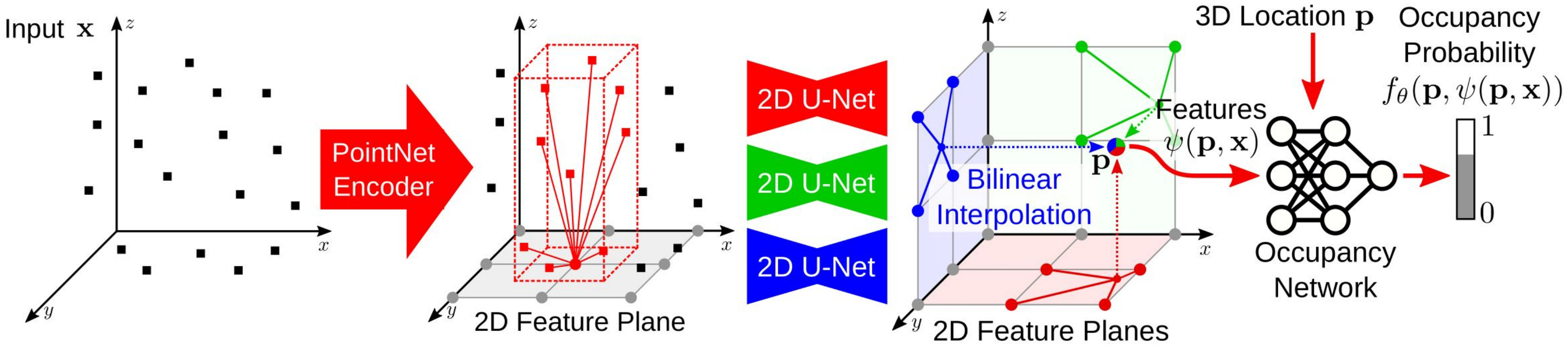
- **3D Volume Encoder**: Use a local PointNet to process input, volumetric feature encoding

Main Idea



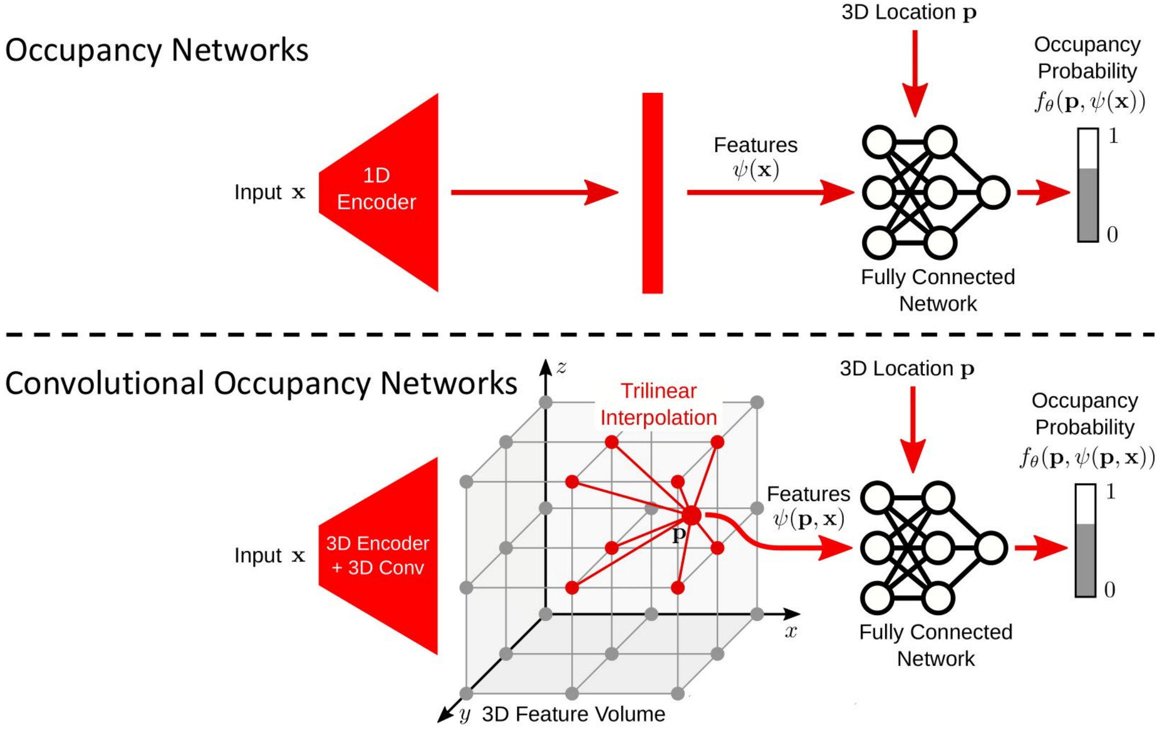
- **3D Volume Encoder**: Use a local PointNet to process input, volumetric feature encoding
- **3D Volume Decoder**: Processed by 3D U-Net, query features via trilinear interpolation
- **Occupancy Readout**: Shallow occupancy network $f_\theta(\cdot)$

Main Idea - 2D



- **2D Plane Encoder**: Use a local PointNet to process input, project onto **3-canonical planes**
- **2D Plane Decoder**: Processed by U-Net, query features via bilinear interpolation
- **Occupancy Readout**: Shallow occupancy network $f_\theta(\cdot)$

Comparison

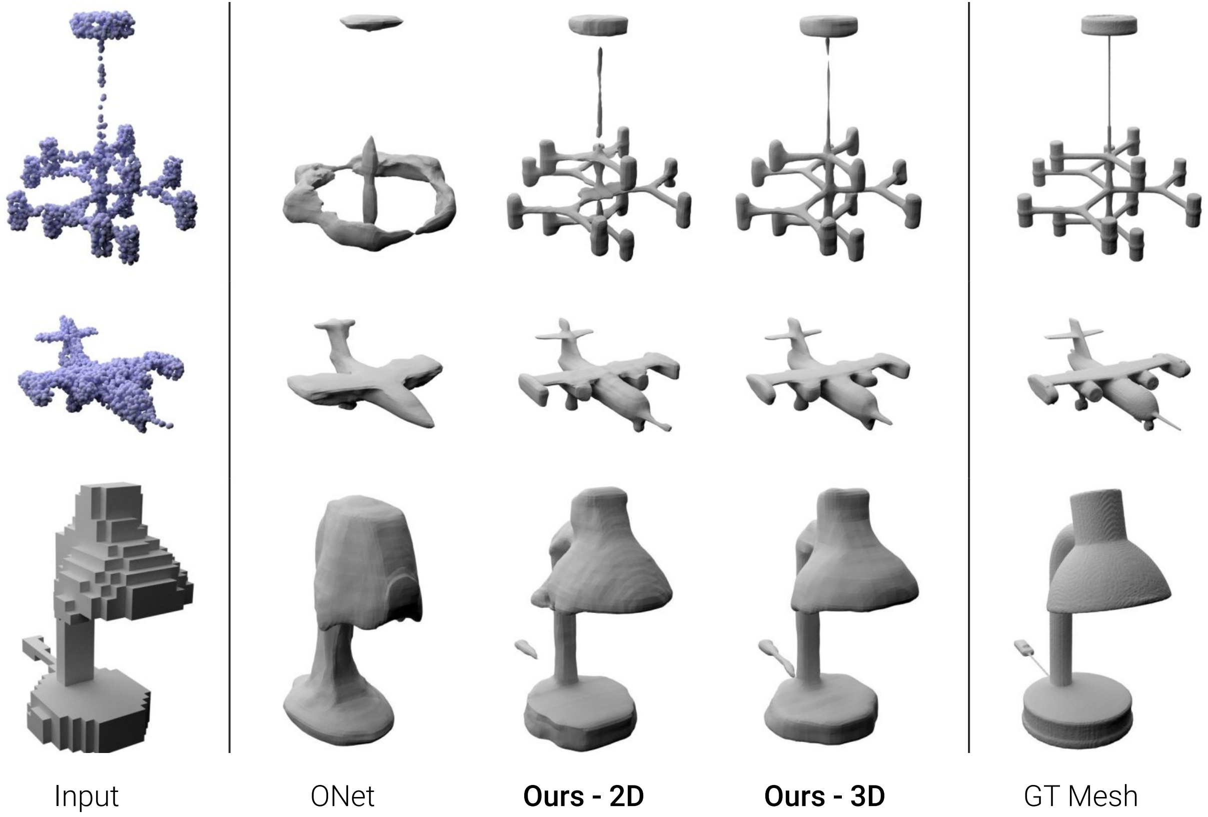


- global feature
- heavy FC network
- no translation equivariance

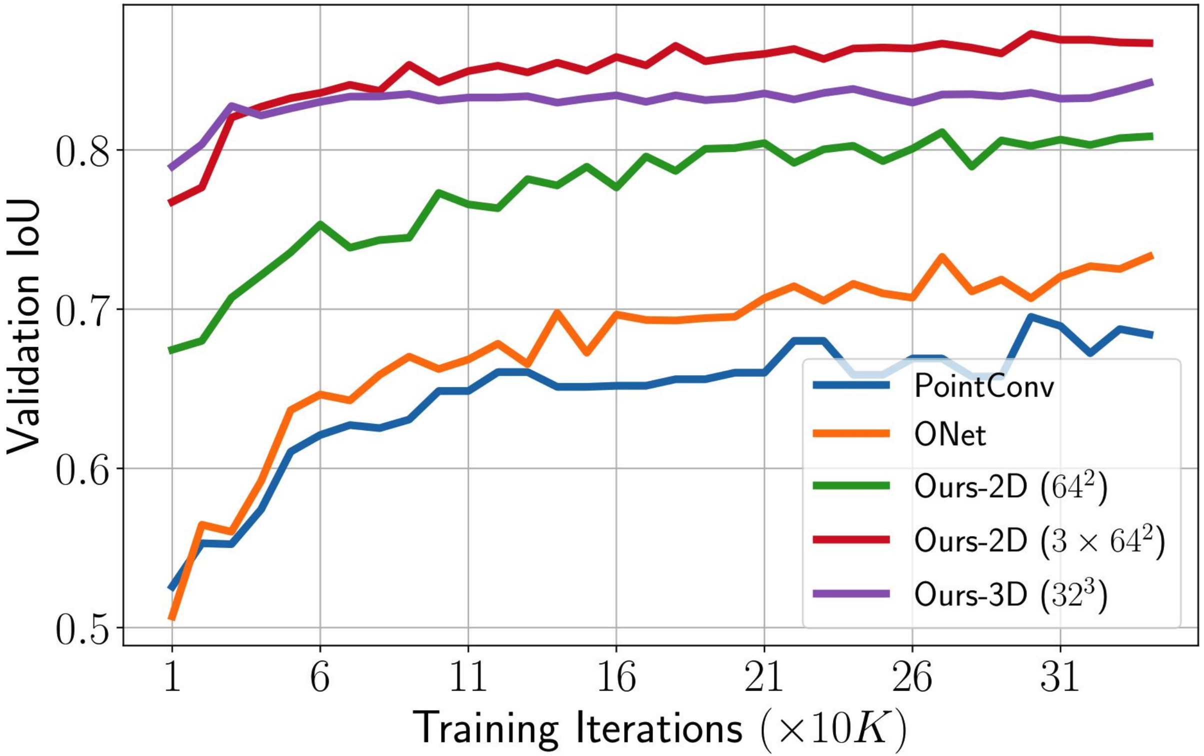
+ local feature
+ shallow FC network
+ translation equivariance

Results

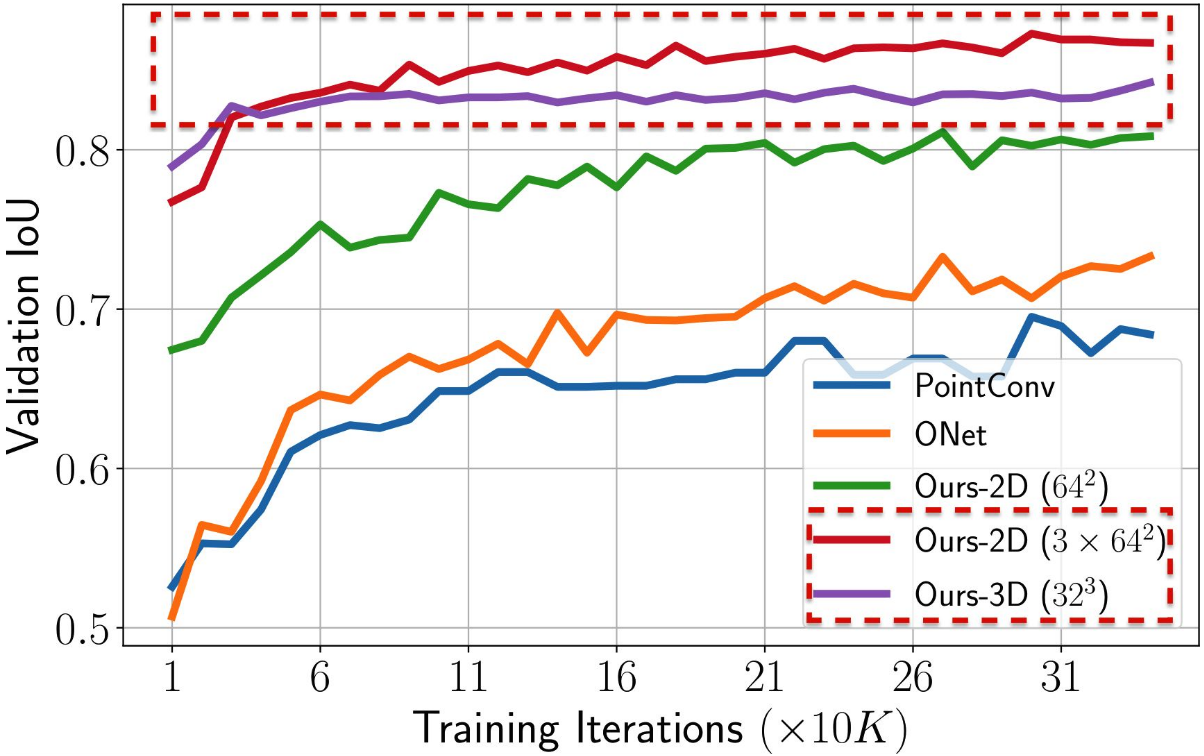
Object-Level Reconstruction



Training Speed

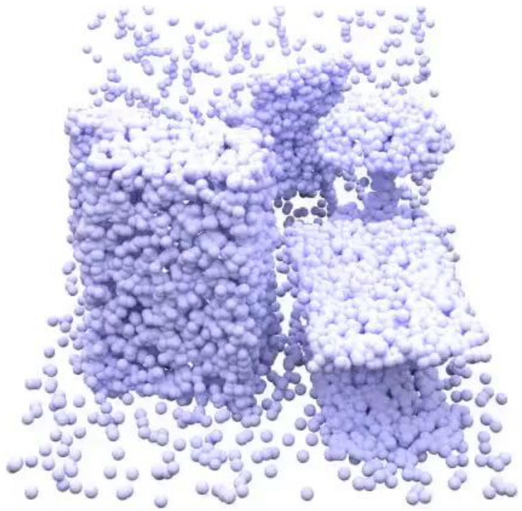


Training Speed



Scene-Level Reconstruction: Synthetic

- Trained and evaluated on synthetic rooms

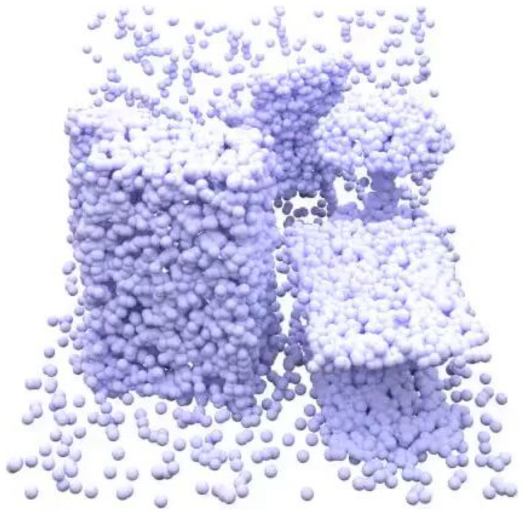


Input

GT Mesh

Scene-Level Reconstruction: Synthetic

- ONet fails on room-level reconstruction

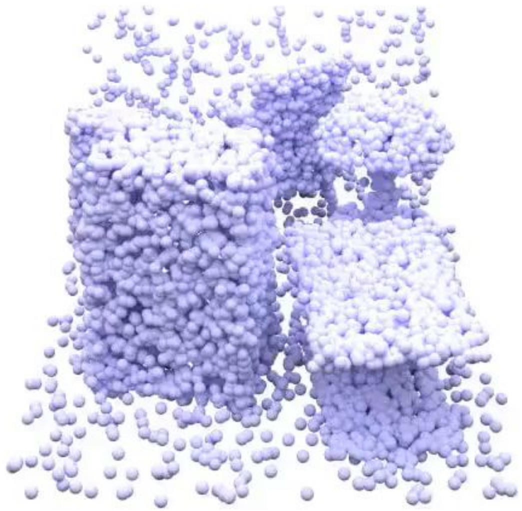


Input

ONet

Scene-Level Reconstruction: Synthetic

- SPSR requires surface normals, output is noisy

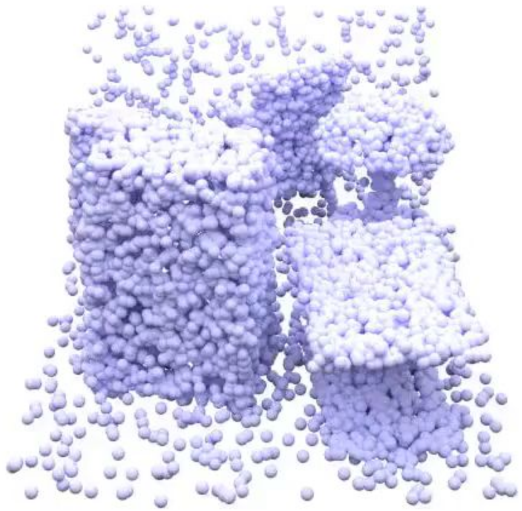


Input

SPSR
(Screened Poisson Surface Reconstruction)

Scene-Level Reconstruction: Synthetic

- Our method preserves better details



Input

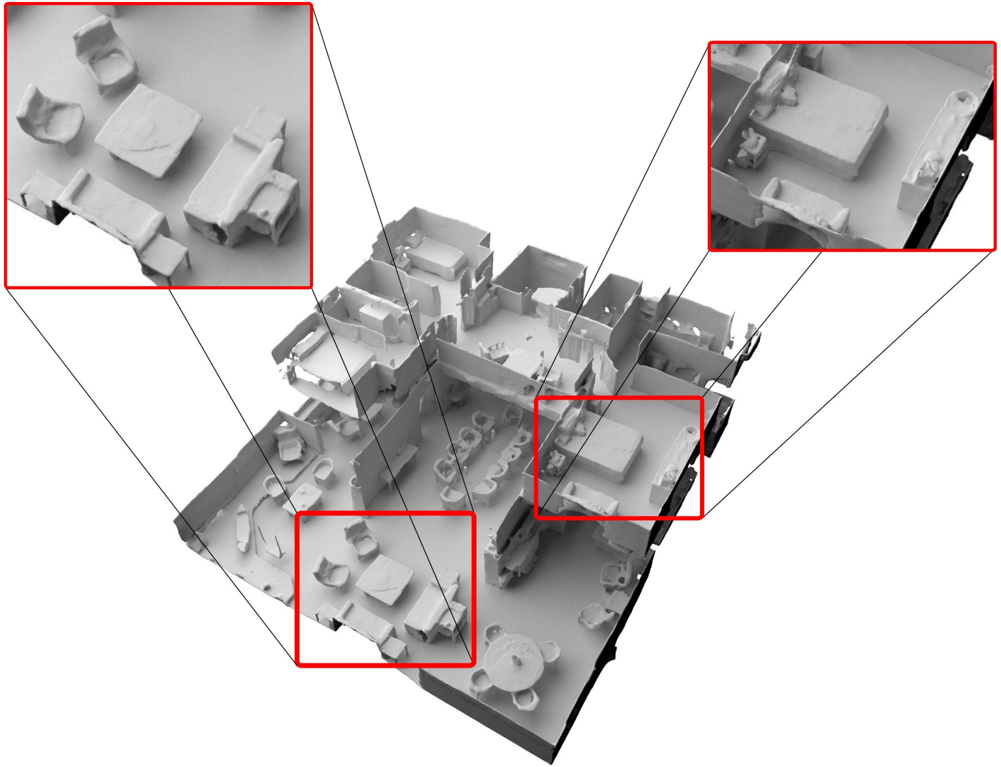
Ours

Large-Scale Reconstruction

Scene size: 15.7m x 12.3m x 4.5m

Results on Matterport3D

- Fully convolutional model
- Trained on synthetic crops
- Sliding-window evaluation
- Scale to any scene size



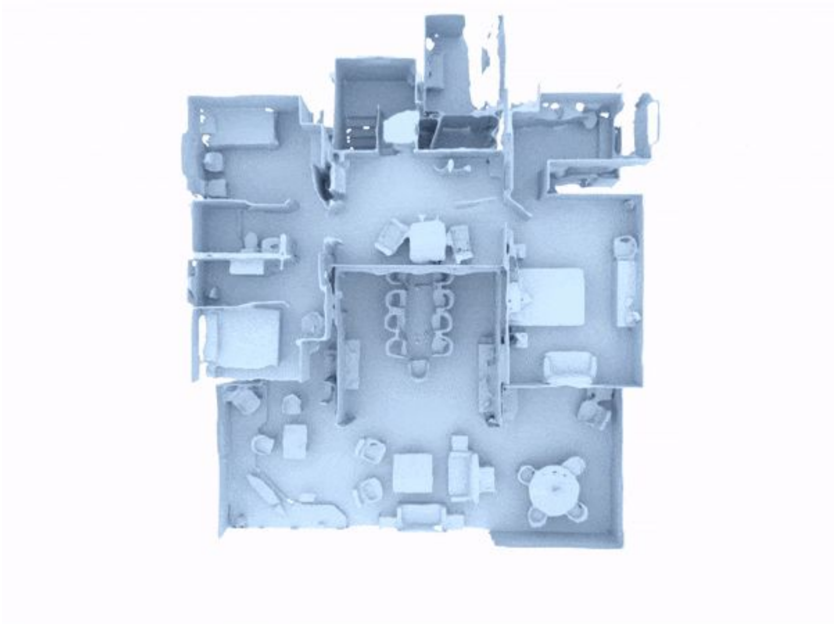
Our reconstruction output

Large-Scale Reconstruction

Scene size: 15.7m x 12.3m x 4.5m

Results on Matterport3D

- Fully convolutional model
- Trained on synthetic crops
- Sliding-window evaluation
- Scale to any scene size



Our reconstruction output

Conclusions

- ConvONet allows for scaling to large-scale scenes
- ConvONet generalizes well from synthetic to real scenes
- ConvONet trains faster than original ONet

Limitations

- **Slow inference** due to dense grid evaluation
- **Difficult to initialize**

ETH zürich

NEURAL INFORMATION PROCESSING SYSTEMS

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

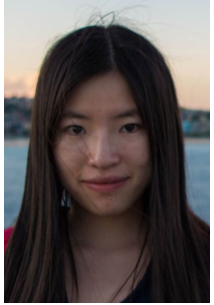
Shape As Points

A Differentiable Poisson Solver

Songyou Peng

Chiyu “Max” Jiang

Yiyi Liao

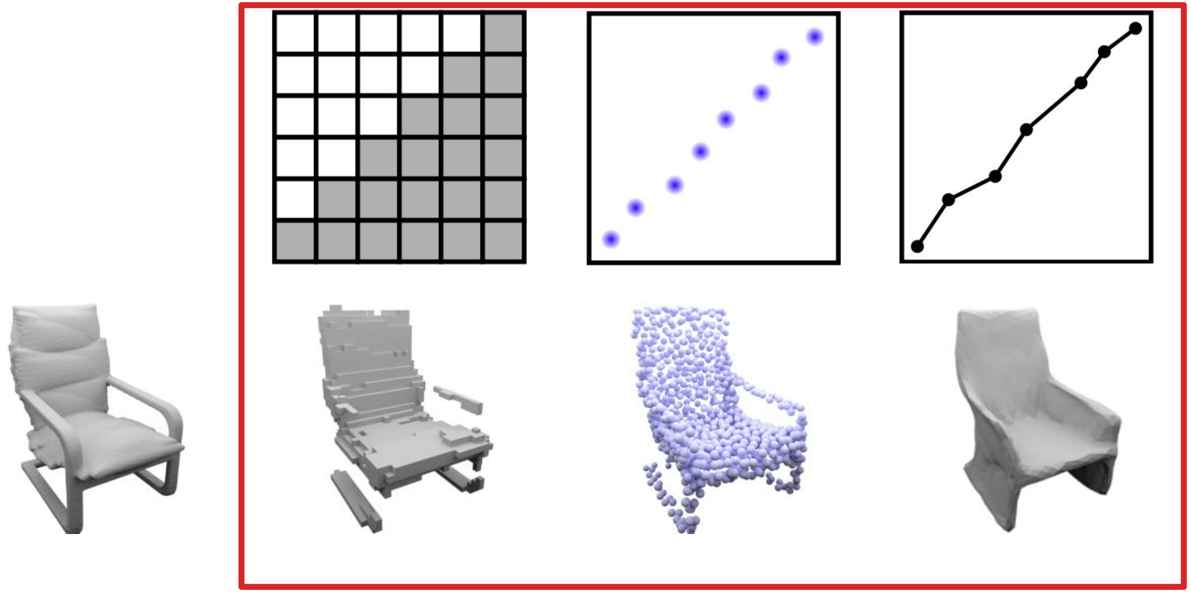


Michael Niemeyer

Marc Pollefeys

Andreas Geiger

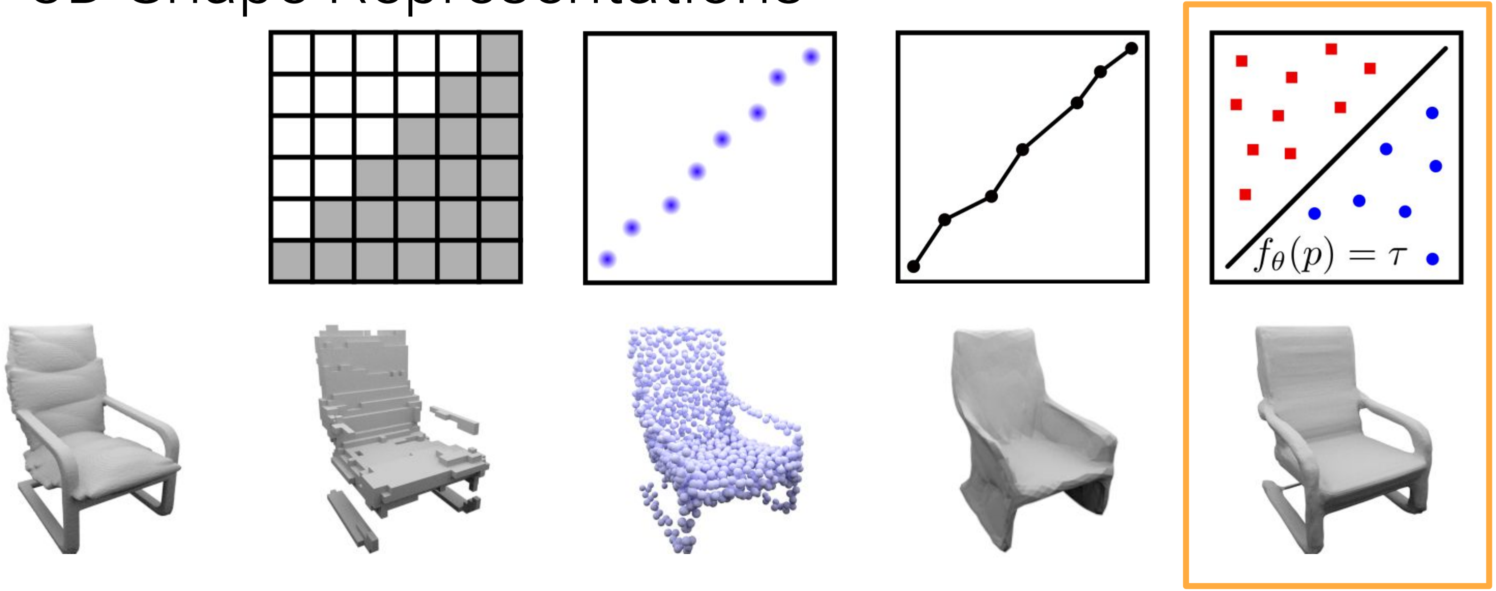
3D Representations



Traditional Explicit Representations

- Discrete
+ Fast inference

3D Shape Representations

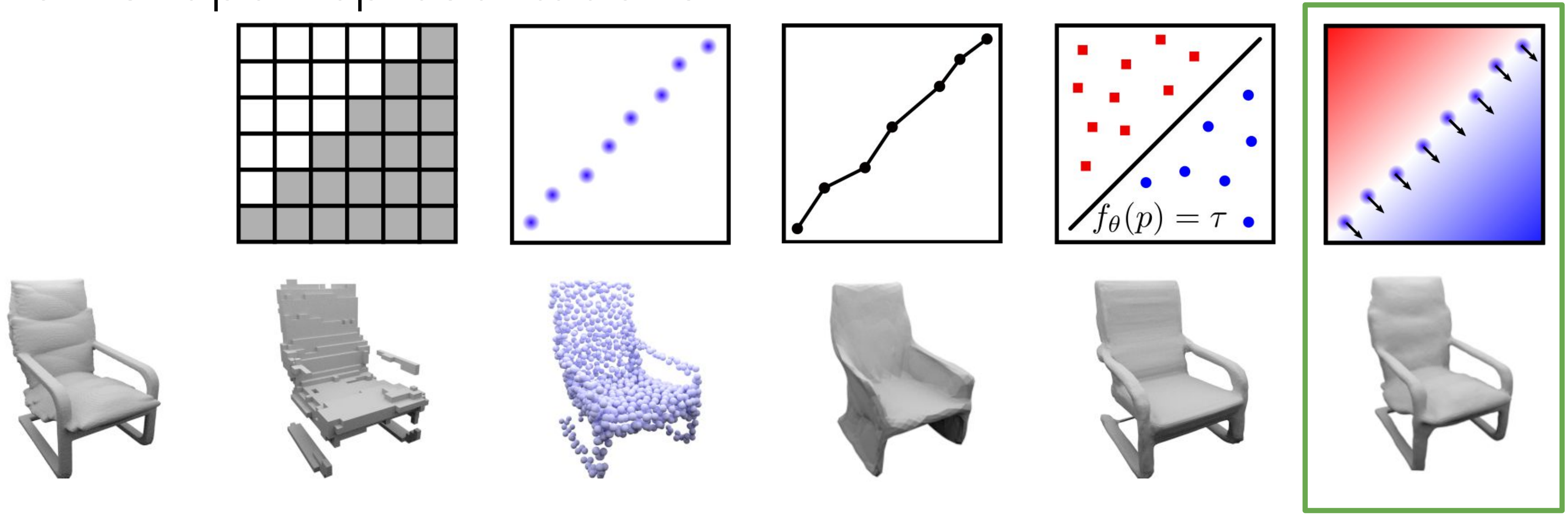


Neural Implicit Representations

- \+ Continuous, watertight
- − Slow inference
- − Difficult to initialize

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

3D Shape Representations



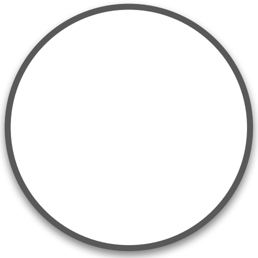
Shape As Points (SAP) - Hybrid Representation

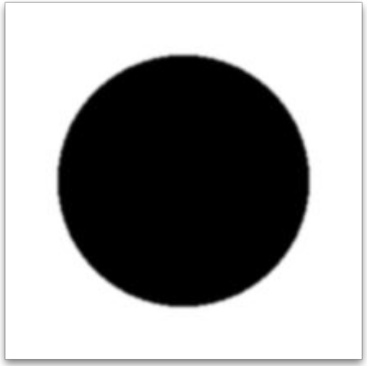
- Discrete (Oriented point clouds) ⇒ Continuous (Implicit indicator grid)
- Fast inference
- Easy initialization

Differentiable Poisson Solver

Intuition of Poisson Equation

$$\boxed{\nabla^2 \chi := \nabla \cdot \nabla \chi = \nabla \cdot \mathbf{v}}$$





χ $\quad$ $\nabla \chi \approx \mathbf{v}$

Shape $\quad$ Indicator Function $\quad$ Gradient $\quad$ Point Normals

Our Poisson Solver

$$\nabla^2 \chi := \nabla \cdot \nabla \chi = \nabla \cdot \mathbf{v}$$

- **Discretization** allows to invert the divergence operator

$$\chi = (\nabla^2)^{-1} \nabla \cdot \mathbf{v}$$

- **Spectral methods** to solve the Poisson equation efficiently
  - Derivatives of signals in spectral domain are computed analytically
  - Fast Fourier Transform (FFT) are **highly optimized on GPUs/TPUs**
  - Only **25-line code**

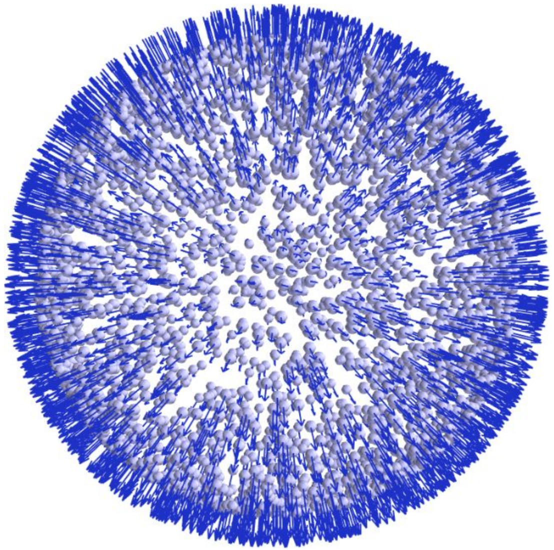
$$\tilde{\mathbf{v}} = \mathrm{FFT}(\mathbf{v}) \quad \longrightarrow \quad \tilde{\chi} = \tilde{g}_{\sigma,r}(\mathbf{u}) \odot \frac{i\mathbf{u} \cdot \tilde{\mathbf{v}}}{-2\pi\|\mathbf{u}\|^2} \quad \longrightarrow \quad \chi' = \mathrm{IFFT}(\tilde{\chi})$$

SAP for Optimization-based 3D Reconstruction

Pipeline - Forward Pass

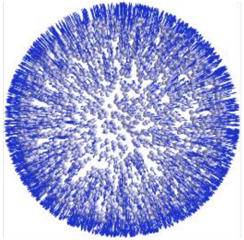
Input an initial oriented point cloud

(noisy / incomplete observations)

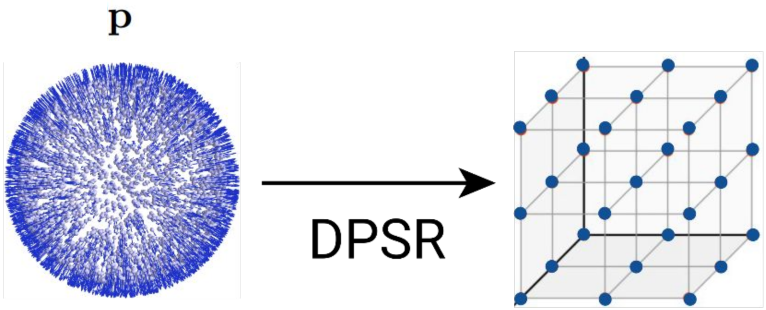


Pipeline - Forward Pass

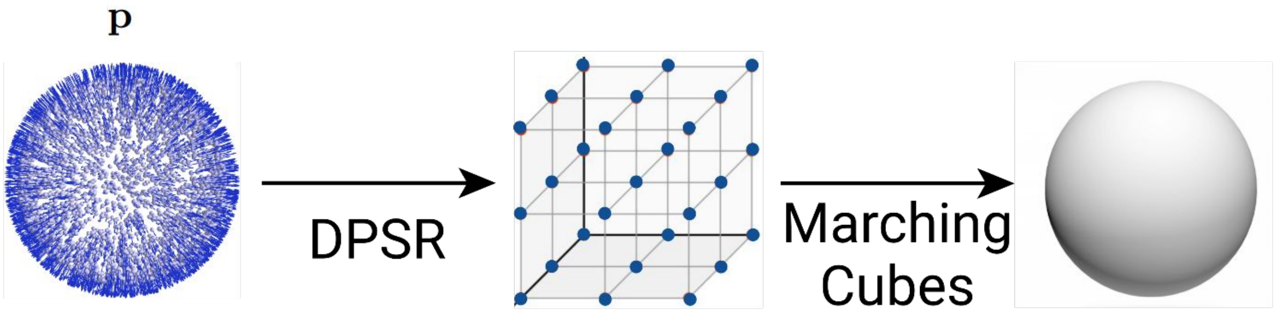
$\mathbf{p}$



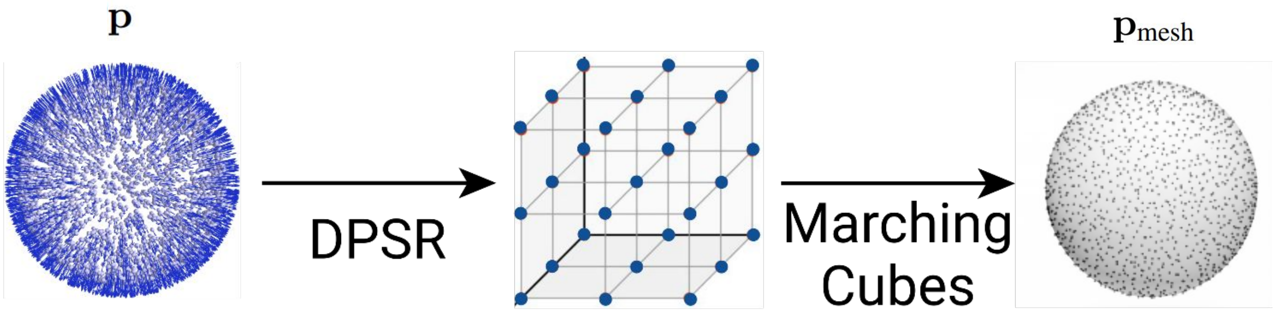
Pipeline - Forward Pass



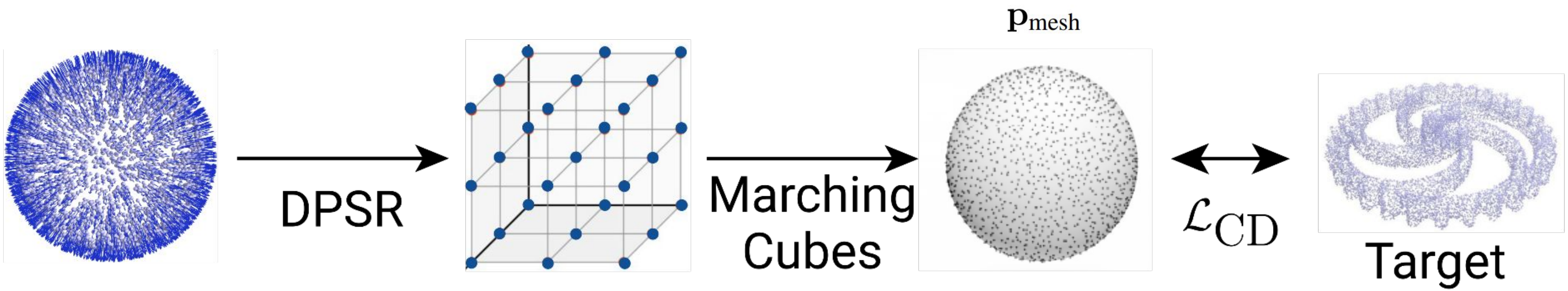
Pipeline - Forward Pass



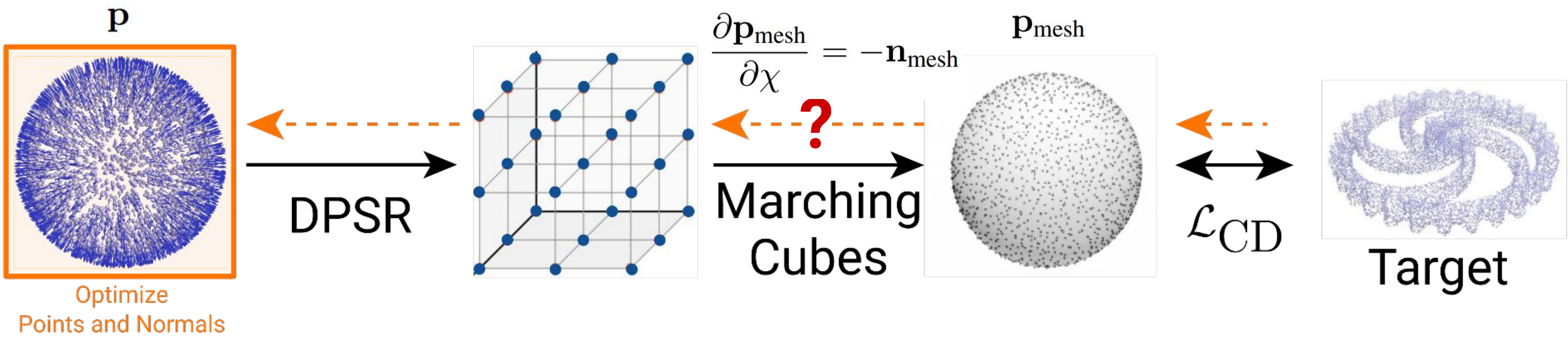
Pipeline - Forward Pass



Pipeline - Forward Pass



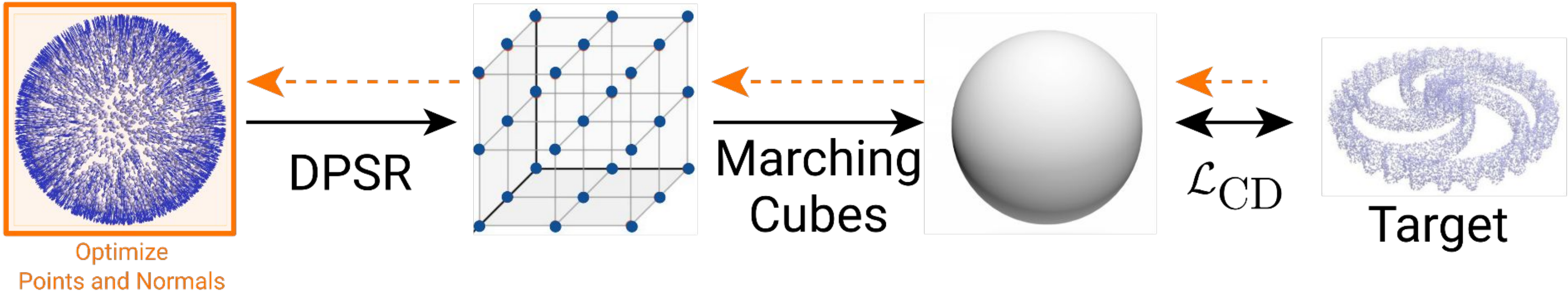
Pipeline - Backward Pass



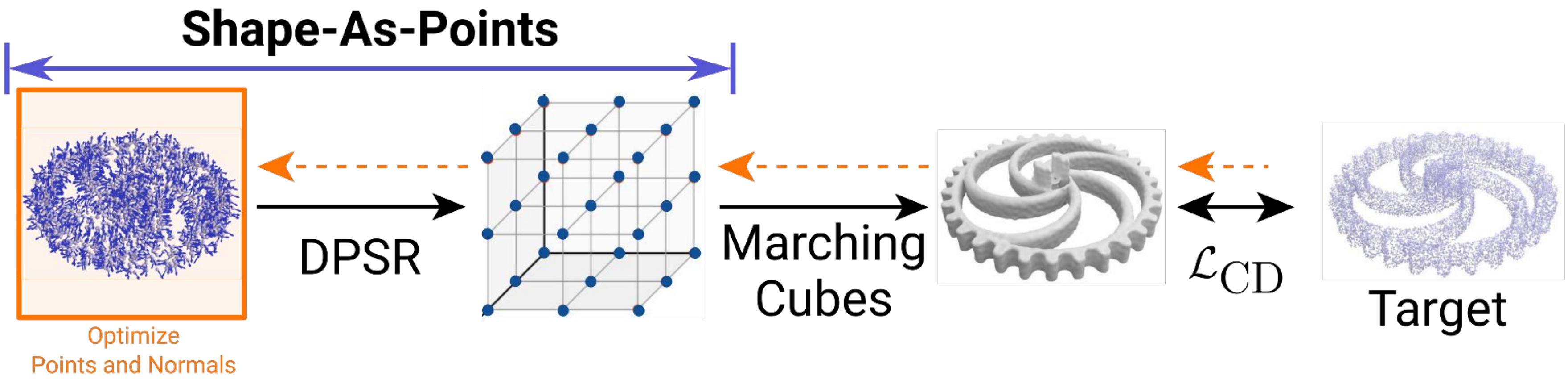
$$\frac{\partial \mathcal{L}_{\text{CD}}}{\partial \mathbf{p}} = \frac{\partial \mathcal{L}_{\text{CD}}}{\partial \mathbf{p}_{\text{mesh}}} \frac{\partial \mathbf{p}_{\text{mesh}}}{\partial \chi} \frac{\partial \chi}{\partial \mathbf{p}}$$

?

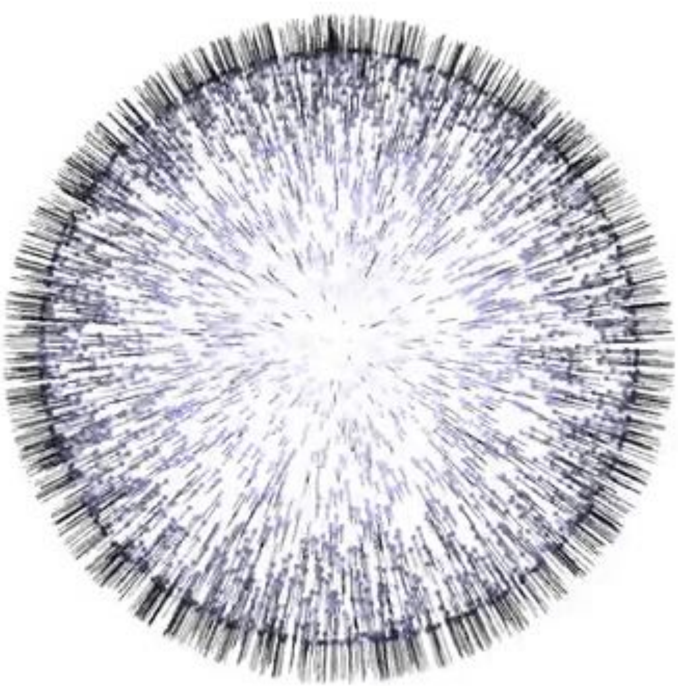
Pipeline

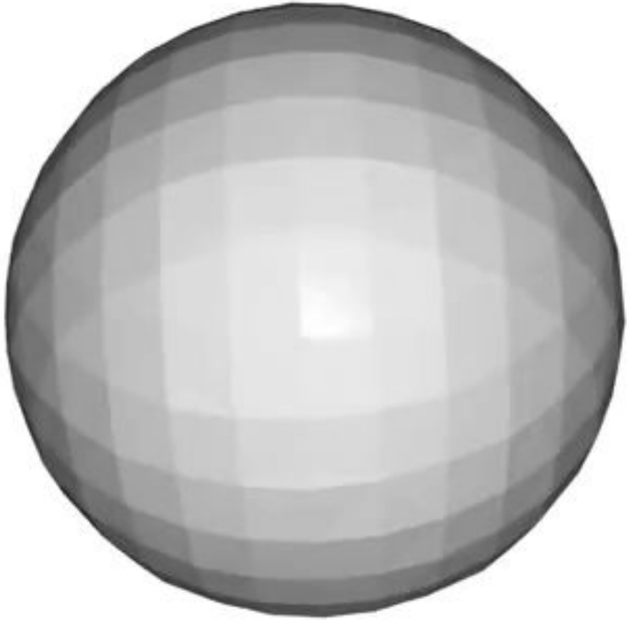

DPSR
Marching Cubes
$\mathcal{L}_{\mathrm{CD}}$
Target
Optimize
Points and Normals

Pipeline

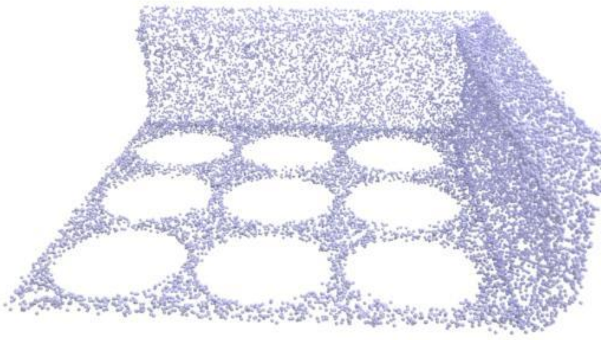


Shape-As-Points
DPSR
Marching Cubes
$\mathcal{L}_{\mathrm{CD}}$
Target
Optimize
Points and Normals





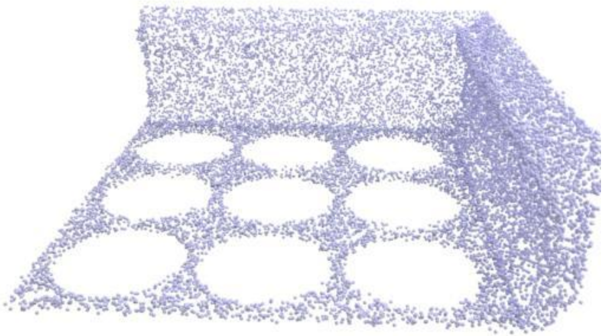
Comparison



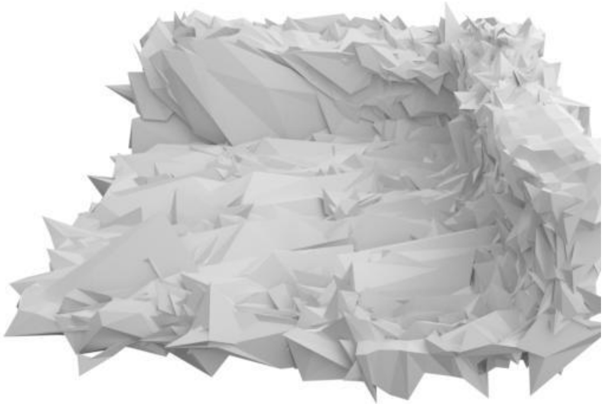
Unoriented Point Clouds

GT Mesh

Comparison



Unoriented Point Clouds

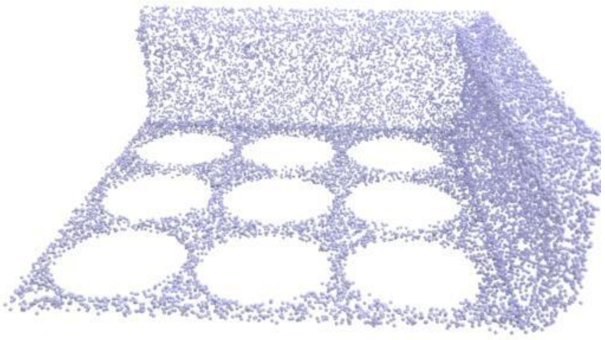


Point2Mesh

Runtime: 62 mins

Hanocka, Metzer, Giryes, Cohen-Or: Point2Mesh: A Self-Prior for Deformable Meshes. SIGGRAPH, 2020

Comparison



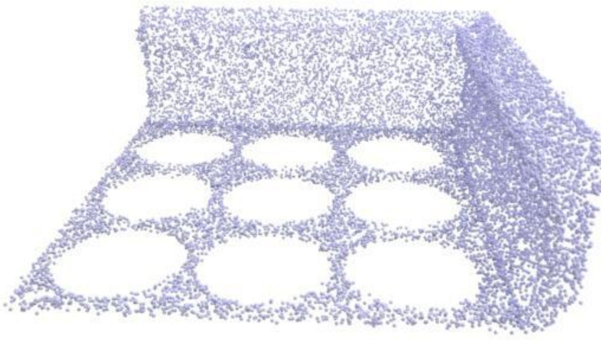
Unoriented Point Clouds

IGR

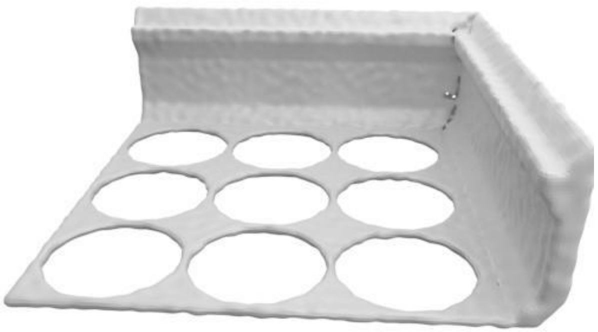
Runtime: 30 mins

Gropp, Yariv, Haim, Atzmon and Lipman: Implicit Geometric Regularization for Learning Shapes. ICML, 2020

Comparison



Unoriented Point Clouds



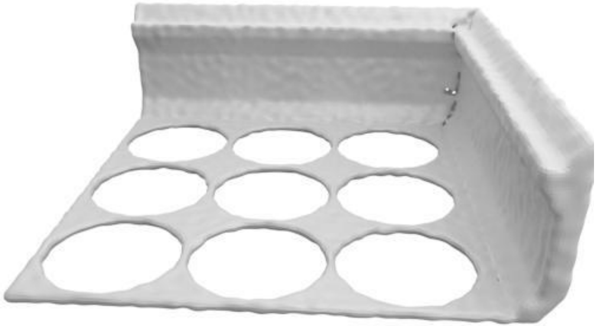
SAP

Runtime: ~6 mins

Comparison

SPSR

Runtime: ~9 sec



SAP

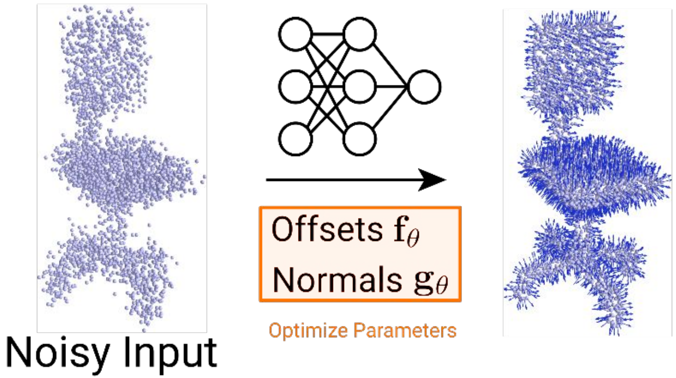
Runtime: ~6 mins

Kazhdan and Hoppe: Screened Poisson Surface Reconstruction. SIGGRAPH, 2013

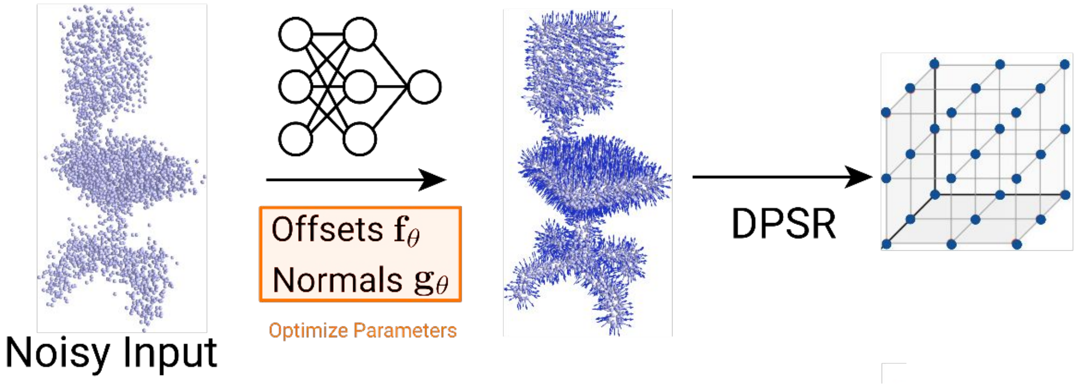
Can we further leverage the **differentiability** of the Poisson solver for **deep neural networks**?

SAP for Learning-based 3D Reconstruction

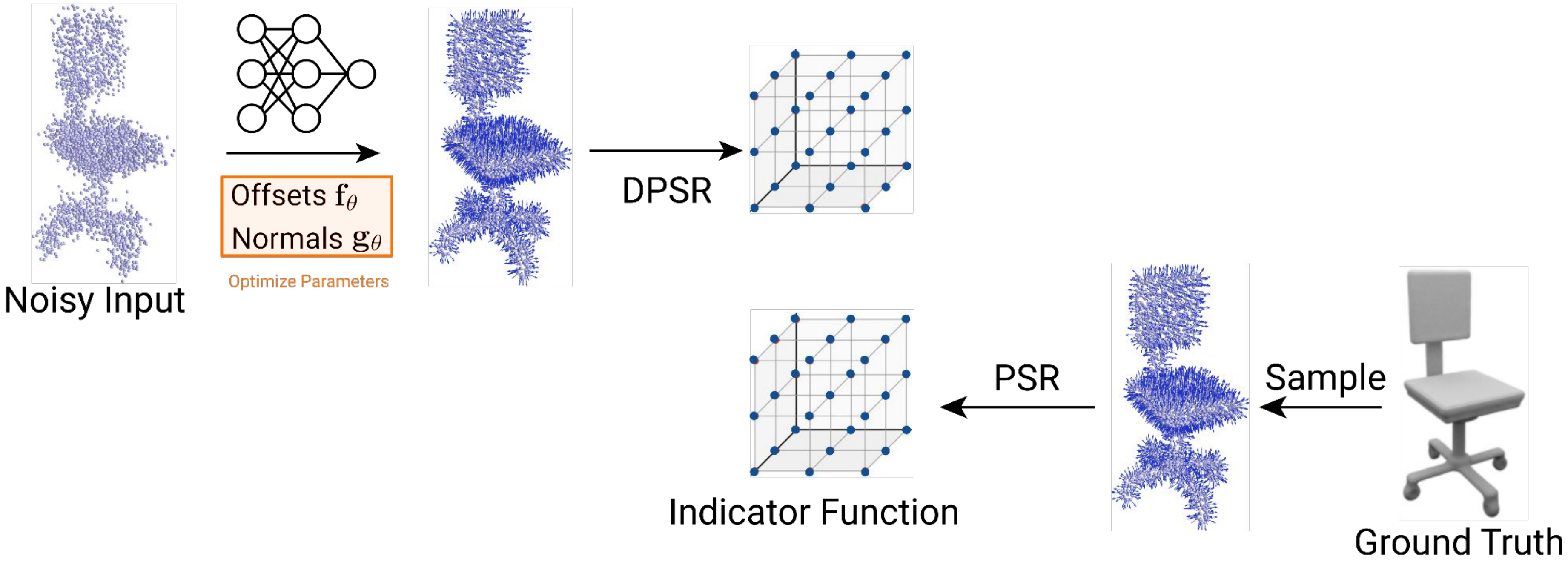
Learning-based Pipeline



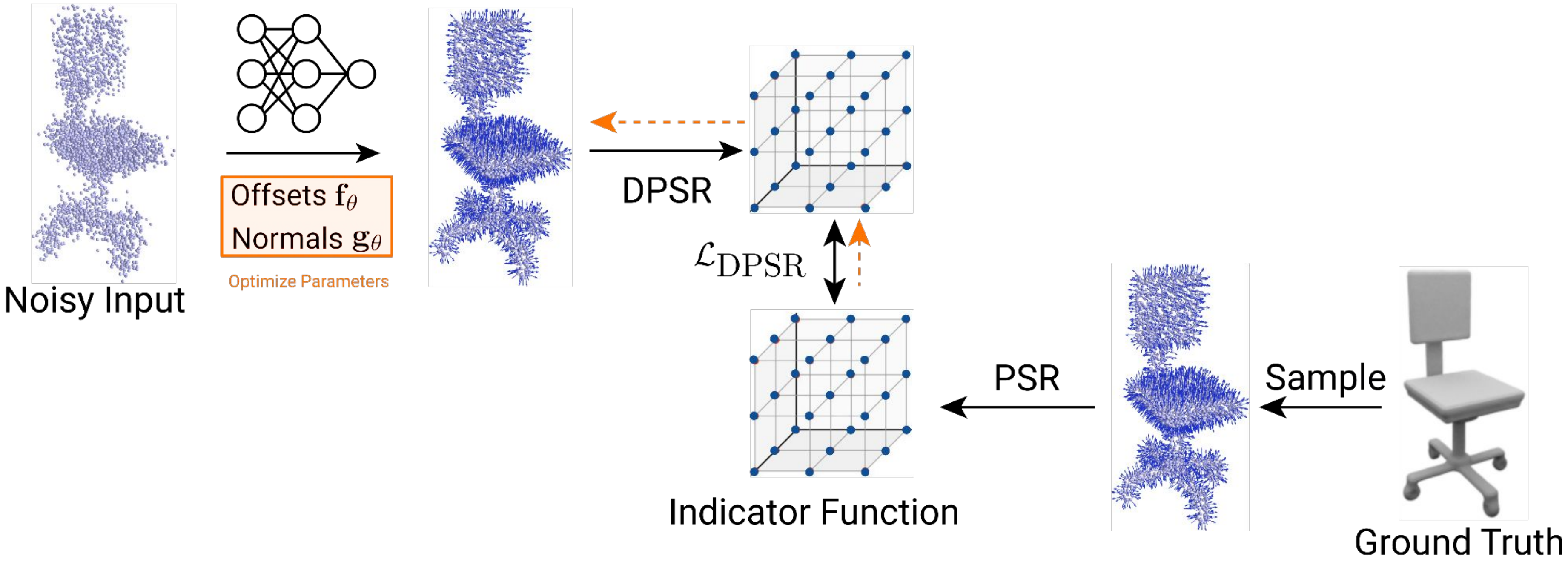
Learning-based Pipeline



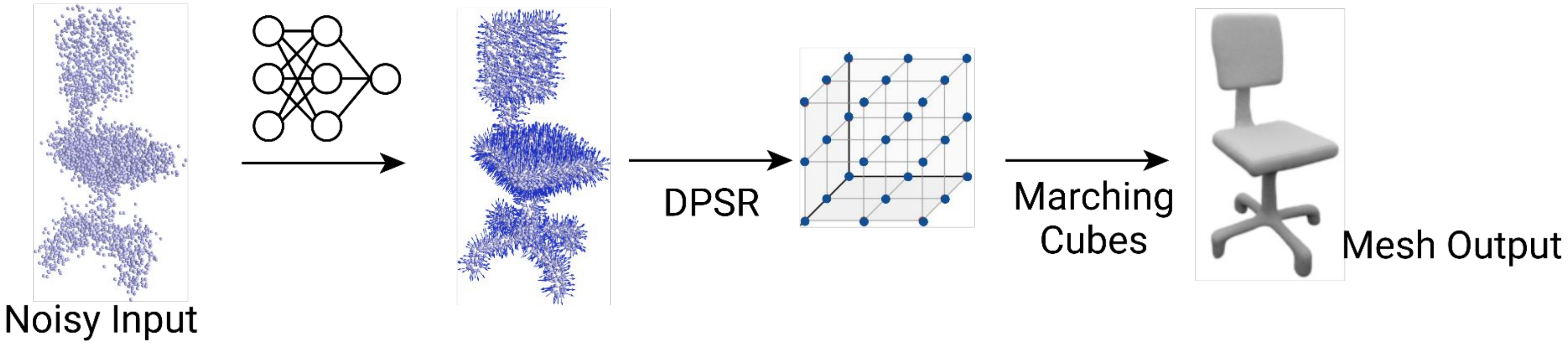
Learning-based Pipeline



Learning-based Pipeline

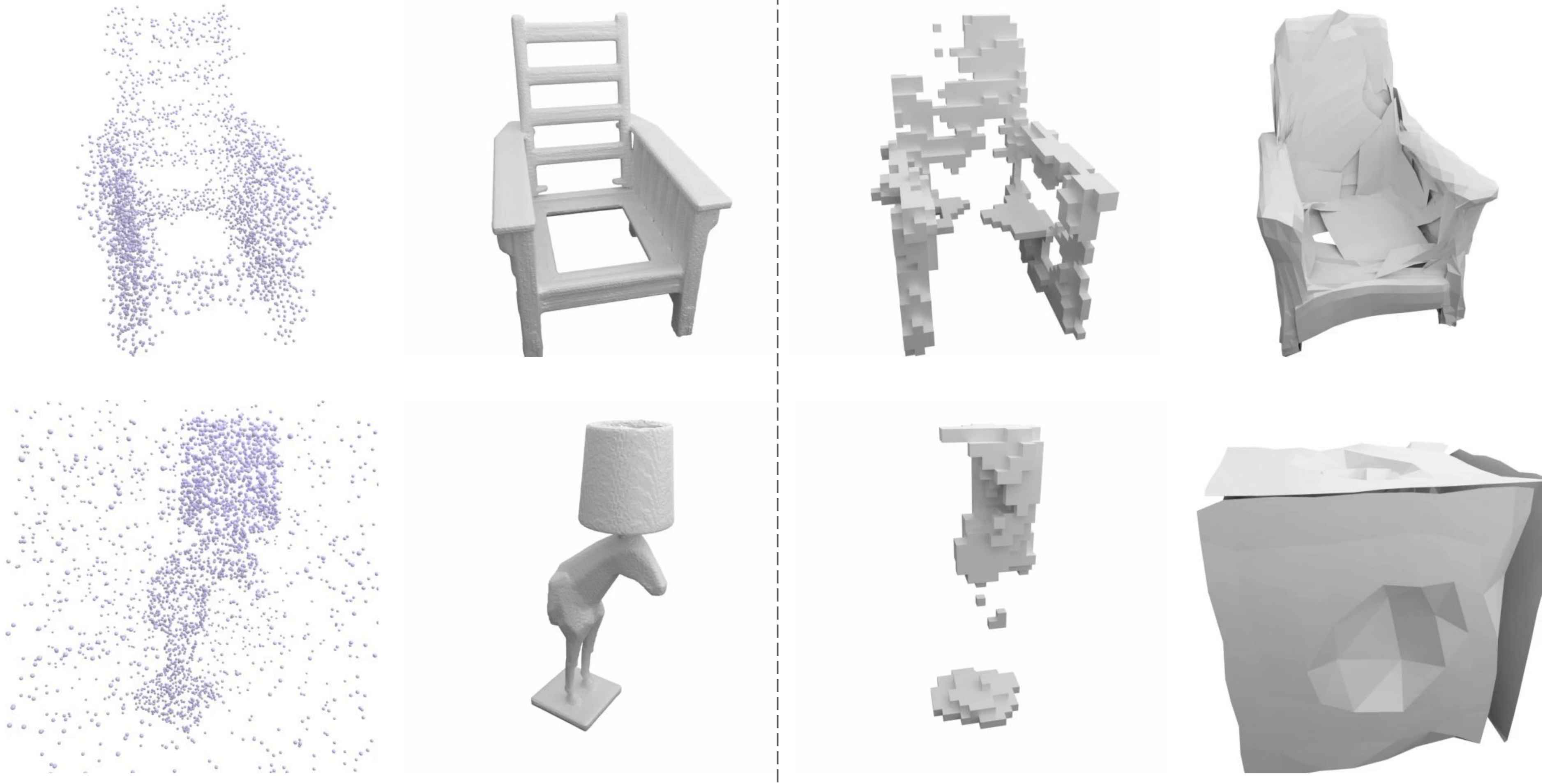


Learning-based Pipeline



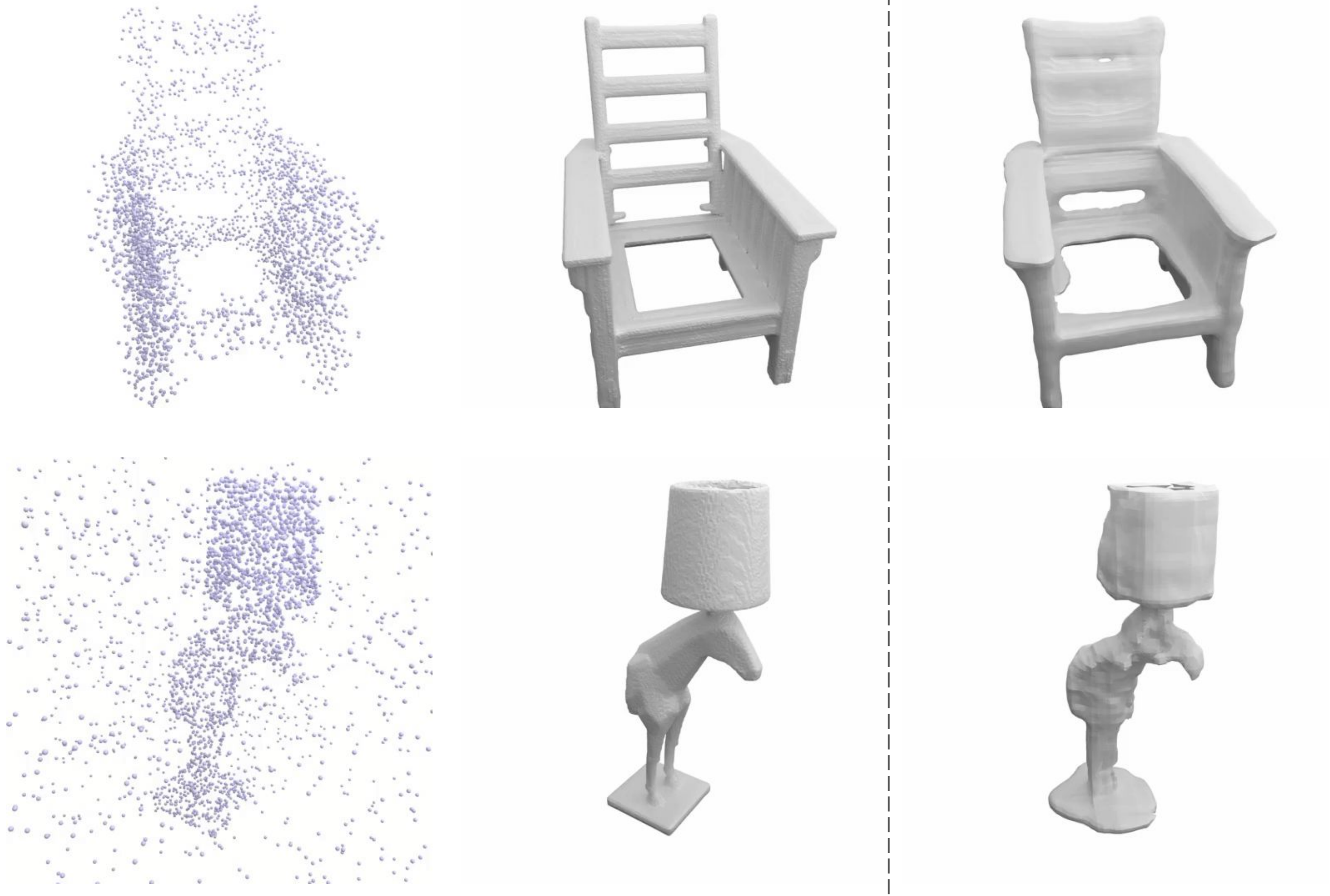
Inputs

GT Mesh



Inputs | GT Mesh | R2N2 | AtlasNet

15 ms (R2N2) | **25 ms** (AtlasNet)

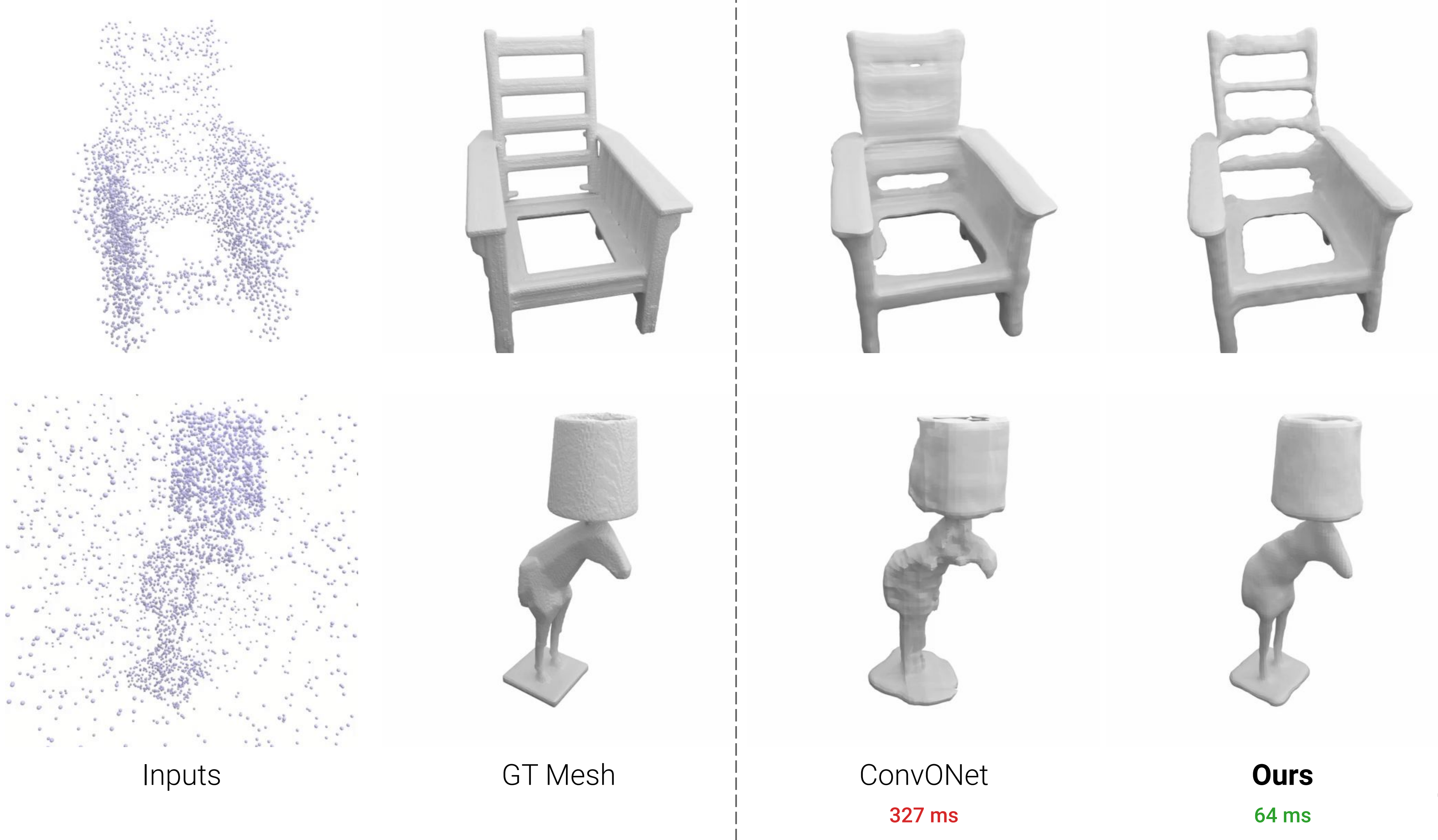


Inputs

GT Mesh

ConvONet

327 ms


Inputs
GT Mesh
ConvONet
327 ms
Ours
64 ms

Benefit of Geometric Initialization

Chamfer distance over the training process

Iterations	10K	50K	100K	200K	Best
ConvONet	0.082	0.058	0.055	0.050	0.044
Ours	**0.041**	**0.036**	**0.035**	**0.034**	**0.034**

SAP converges much faster!

Conclusions

- SAP is **interpretable**, **lightweight** and guarantees **HQ watertight meshes**
- SAP is also **topology agnostic**, enables **fast inference**
- Our Poisson solver is **differentiable** and **GPU-accelerated**

Limitation: Cubic memory requirements limits SAP for small scenes


YOU HAVING FUN
YET?
makeameme.org

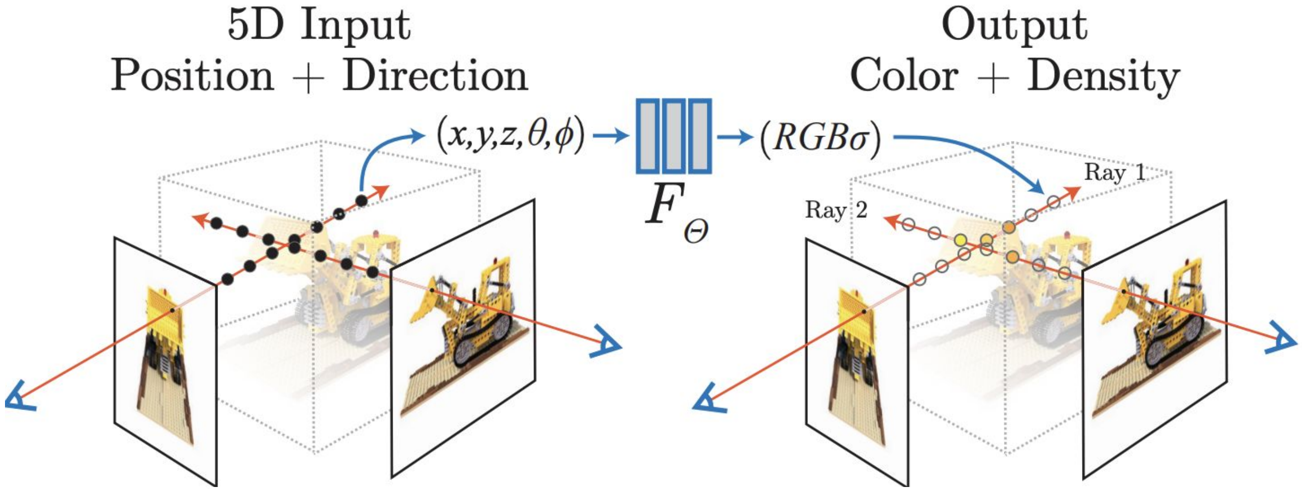

SO WHAT'S NEXT...
makeameme.org

Neural Radiance Field (NeRF) **Heat**

Learning Scene Textures

Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV 2020

Neural Radiance Field (NeRF)



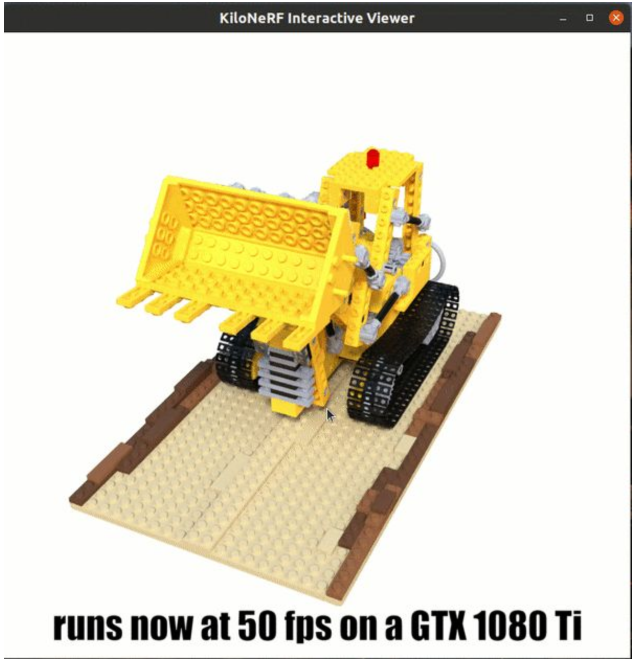
$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{c}_i\,,\ \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi, Ng: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV 2020

KiloNeRF: Speeding up NeRF with Thousands of Tiny MLPs

Christian Reiser, Songyou Peng, Yiyi Liao, Andreas Geiger

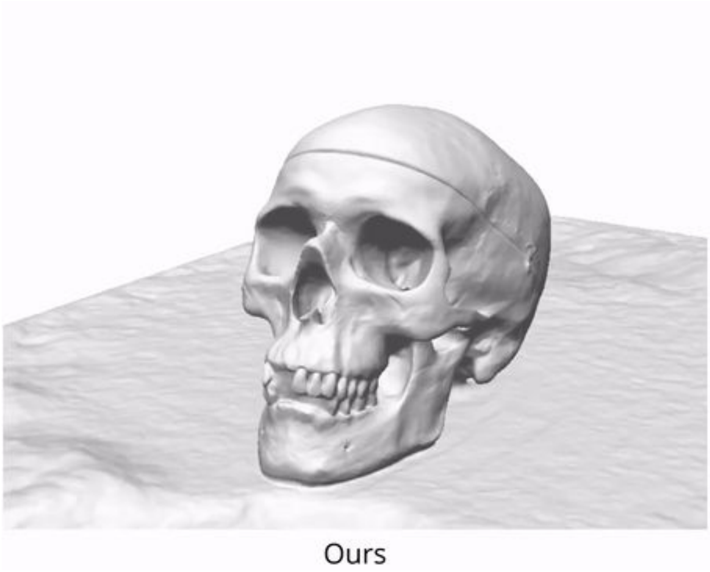
ICCV 2021



UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction

Michael Oechsle, Songyou Peng, Andreas Geiger

ICCV 2021 (Oral)



Given an RGB-D sequence, jointly represent **geometry & color**?

What if the camera poses are also unknown?

Neural Implicit Representations

for

Simultaneous Localization and Mapping
(SLAM)

NICE-SLAM
Neural Implicit Scalable Encoding for SLAM

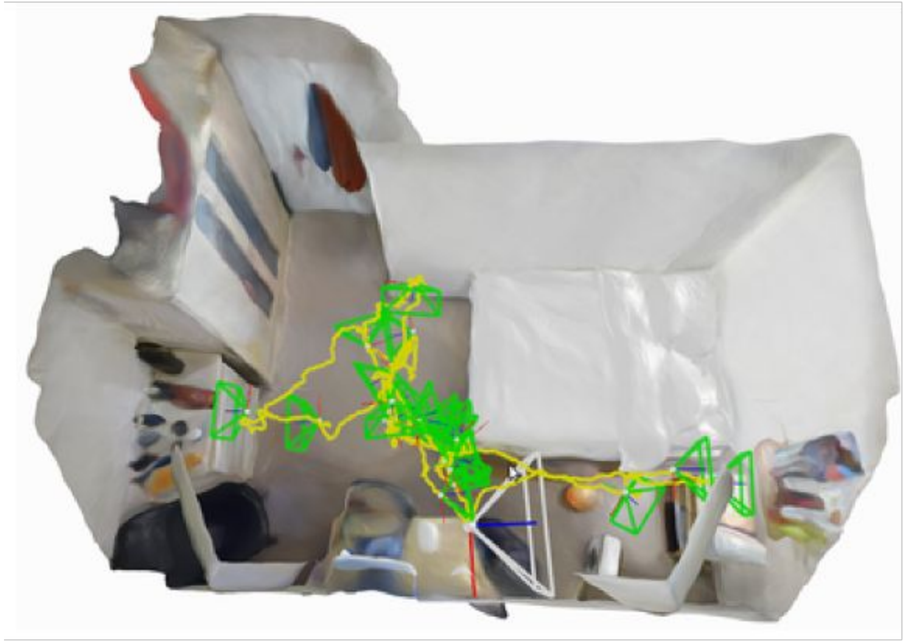
Zihan Zhu[1,2 *] Songyou Peng[1,3 *] Viktor Larsson[1] Weiwei Xu[2] Hujun Bao[2]
Zhaopeng Cui[2 #] Martin R. Oswald[1,4] Marc Pollefeys[1,5]

(* equal contribution)

[1]ETH Zurich [2]State Key Lab of CAD&CG, Zhejiang University
[3]MPI for Intelligent Systems, Tübingen [4]University of Amsterdam [5]Microsoft

https://pengsongyou.github.io/nice-slam

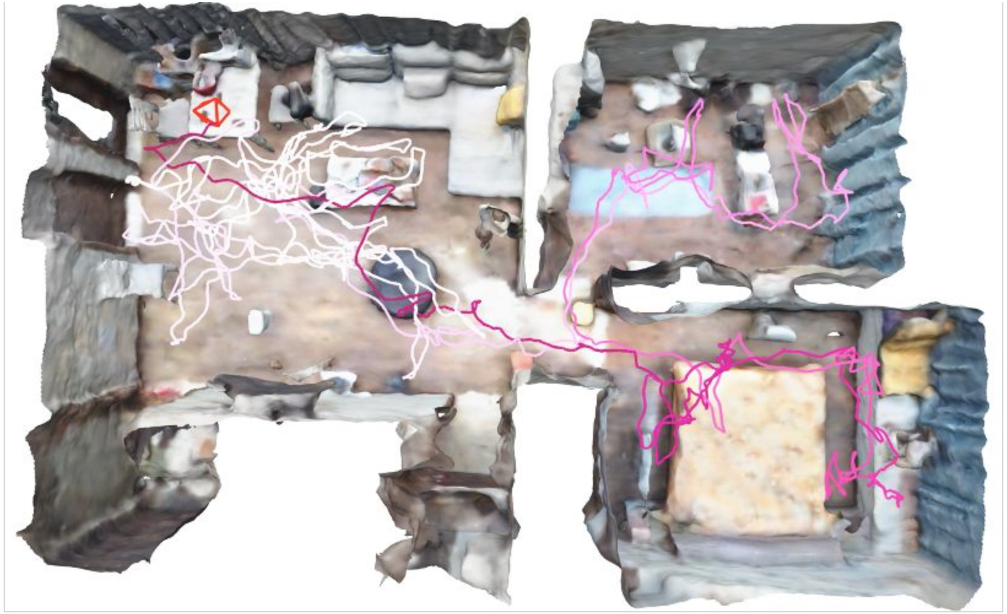
iMAP



- First dense SLAM system that uses a neural scene representation
- Use a single fully-connected network to represent the entire scene
- Low-quality geometry, fail in larger scenes

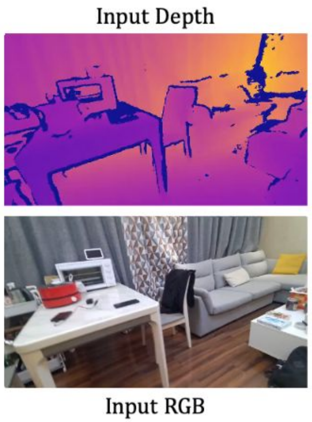
Sucar, Liu, Ortiz, Davison: iMAP: Implicit Mapping and Positioning in Real-Time. ICCV 2021

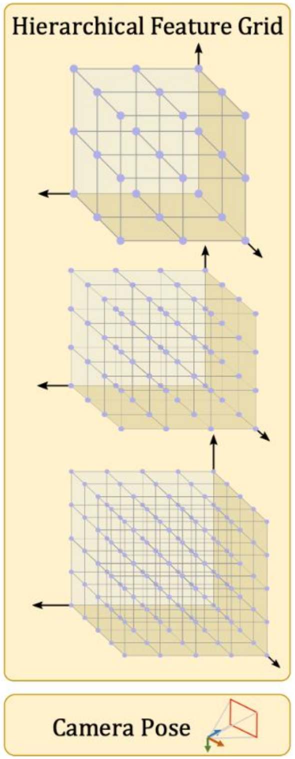
NICE-SLAM



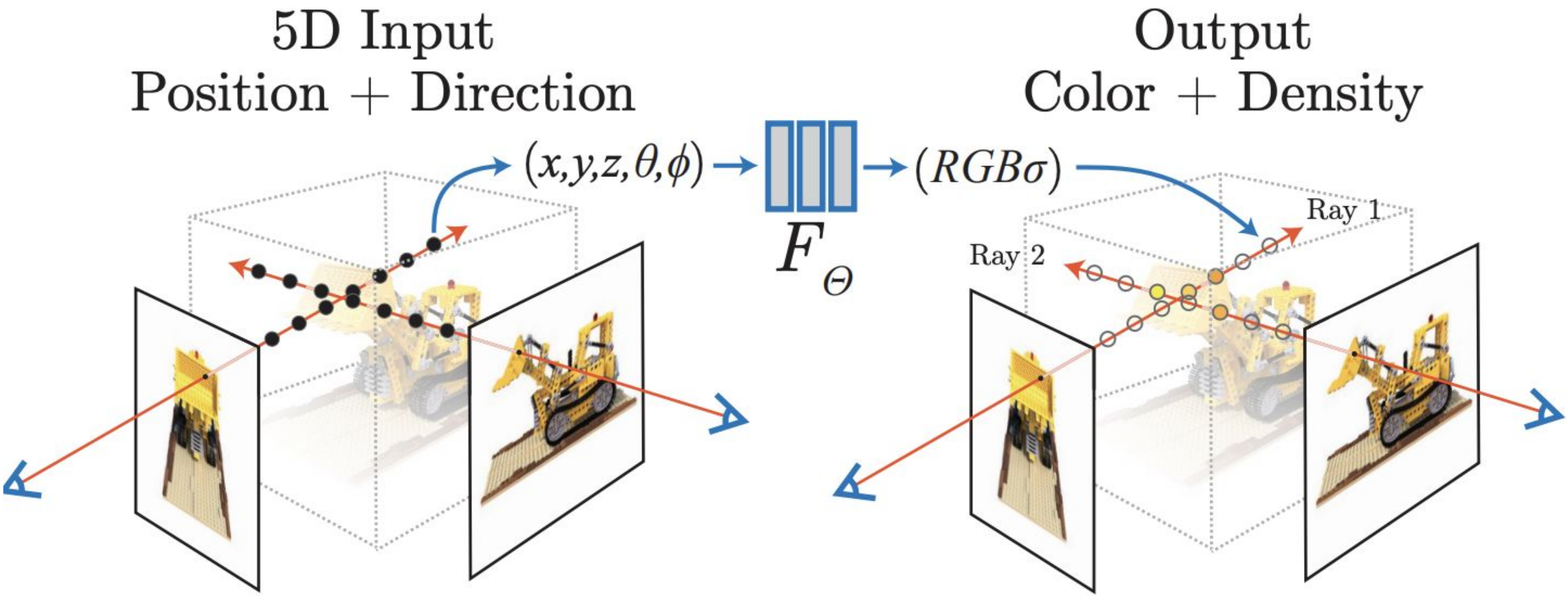
- Hierarchical grid-based encoding
- High-quality scene geometry & camera tracking on large-scale scene
- Local updates -> No forgetting problem for geometry

NICE-SLAM Overview



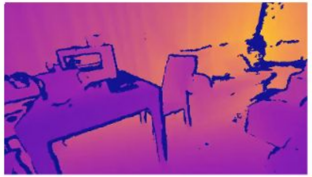


NICE-SLAM Overview

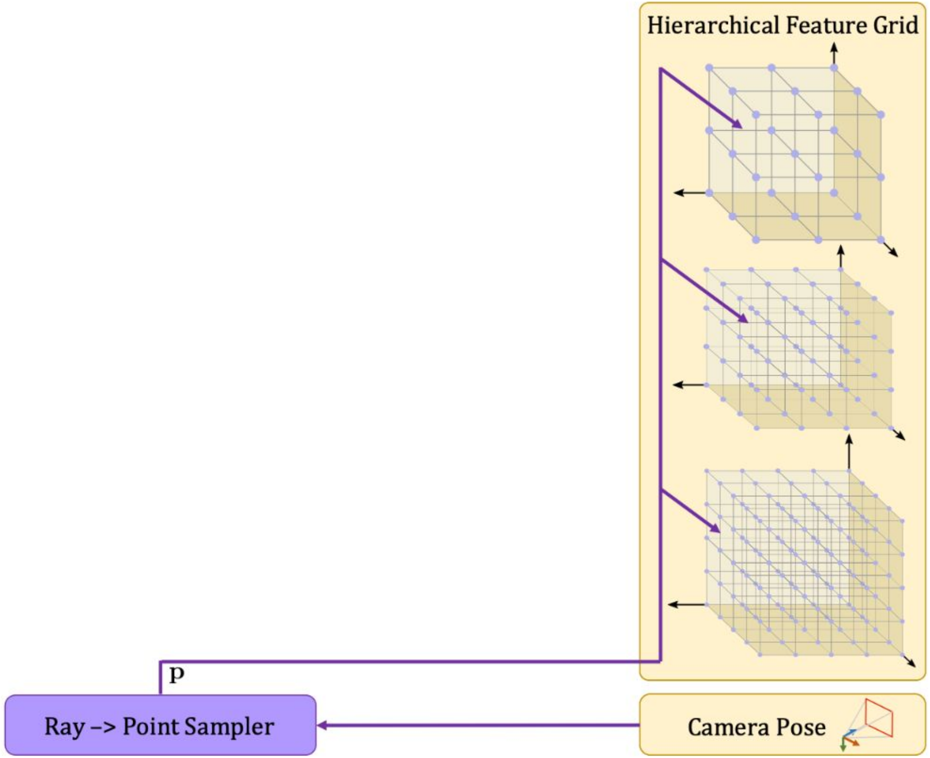


NICE-SLAM Overview

Input Depth

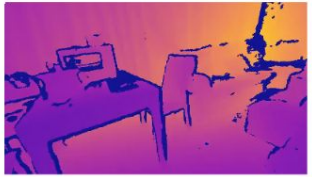


Input RGB

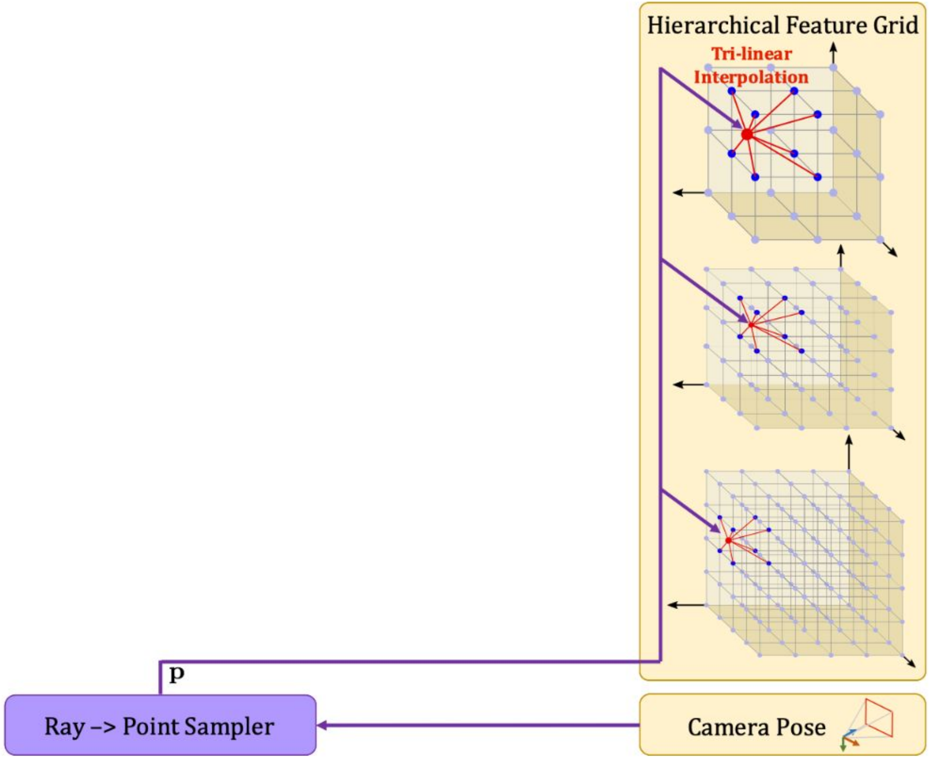


NICE-SLAM Overview

Input Depth

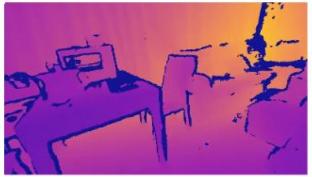


Input RGB

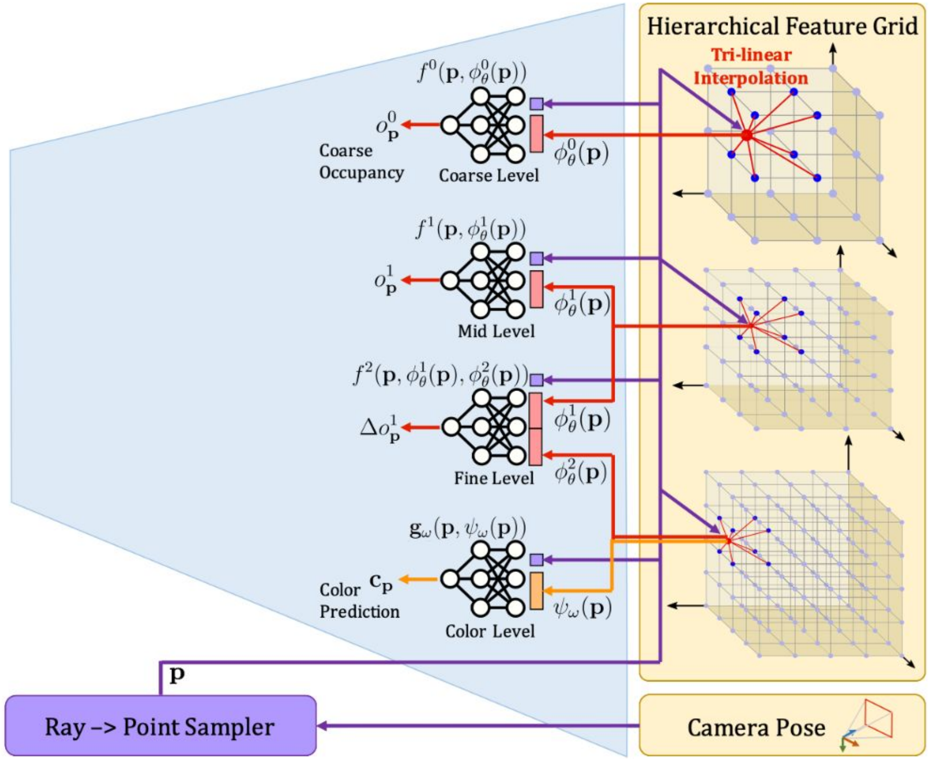


NICE-SLAM Overview

Input Depth

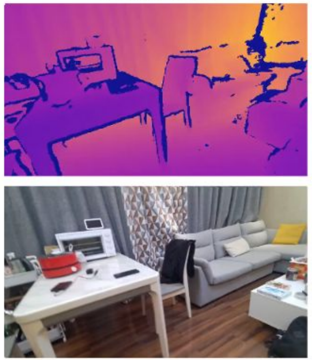


Input RGB



NICE-SLAM Overview

Input Depth

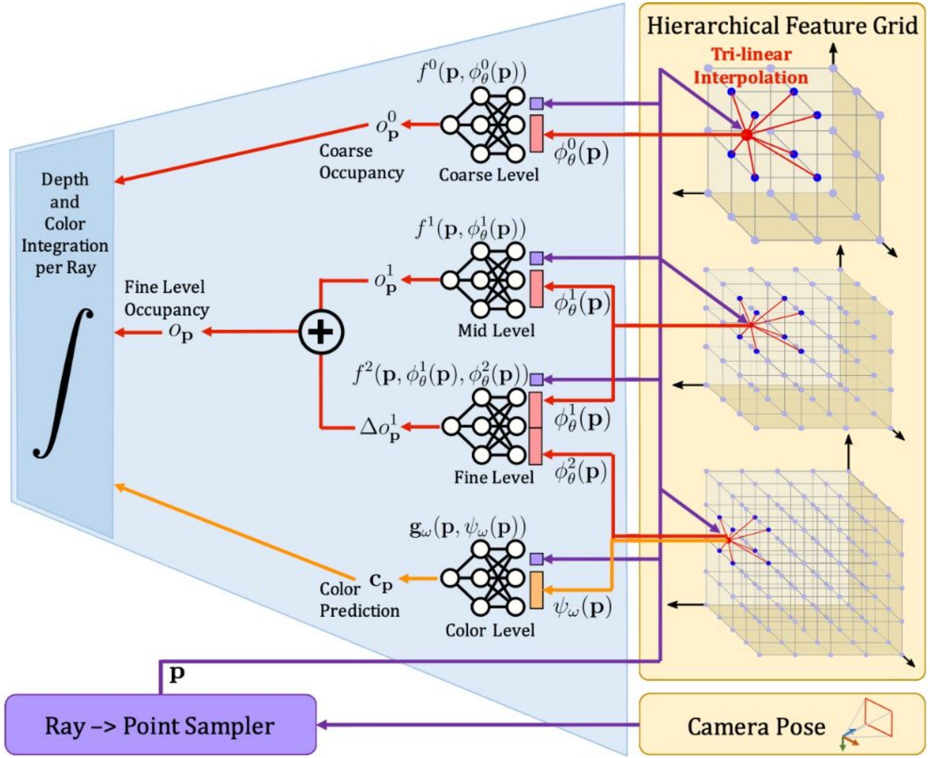


Input RGB

Reformulate the volume rendering equation along each ray as [1]:

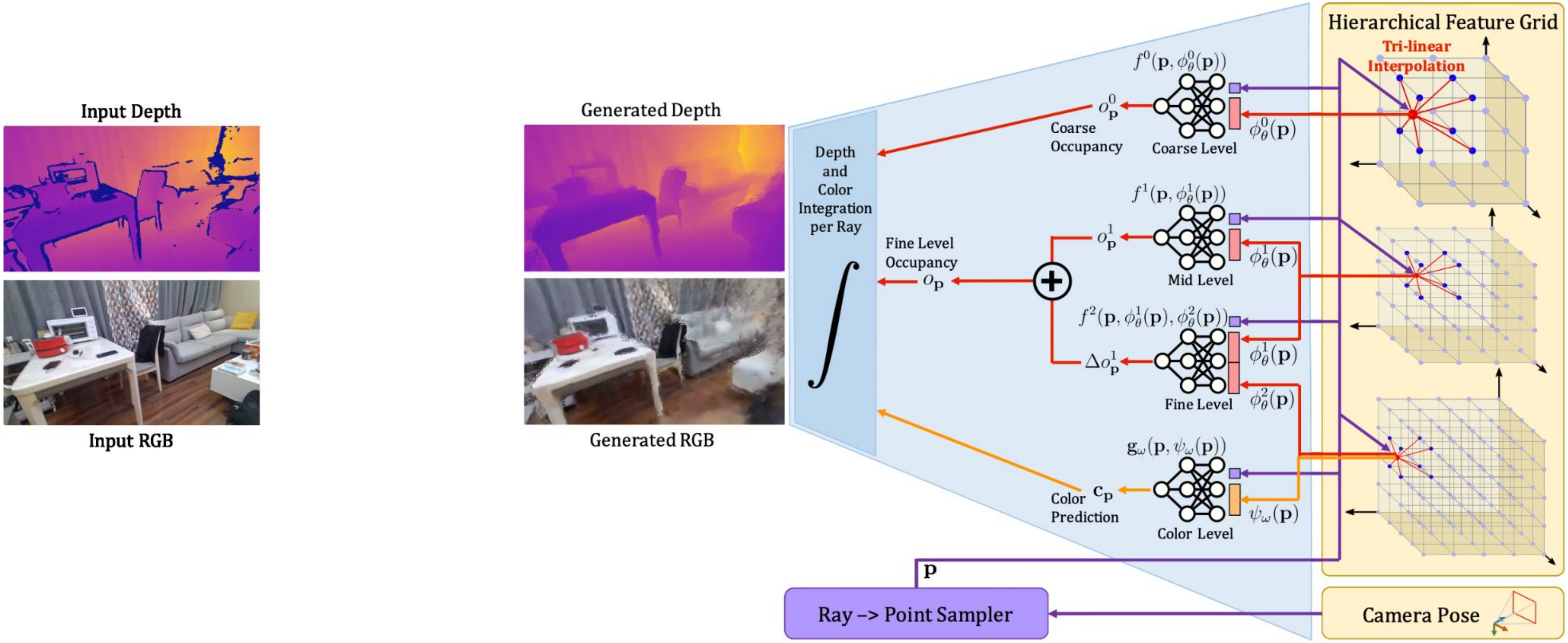
$$\hat{D}^f = \sum_{i=1}^{N} w_i^f d_i, \quad \hat{I} = \sum_{i=1}^{N} w_i^f \mathbf{c}_i$$

$$w_i^f = o_{\mathbf{p}_i} \prod_{j=1}^{i-1} (1 - o_{\mathbf{p}_j})$$

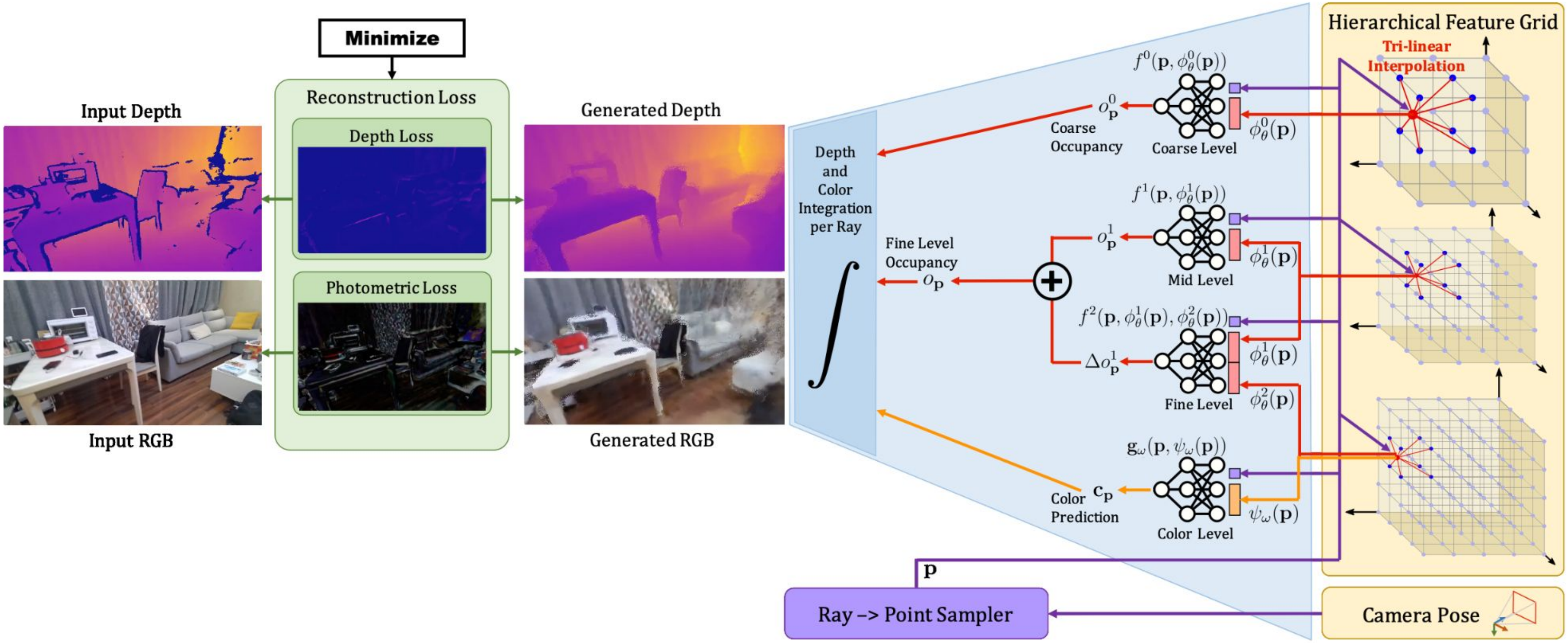


[1] Oechsle, Peng, Geiger: UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. ICCV 2021

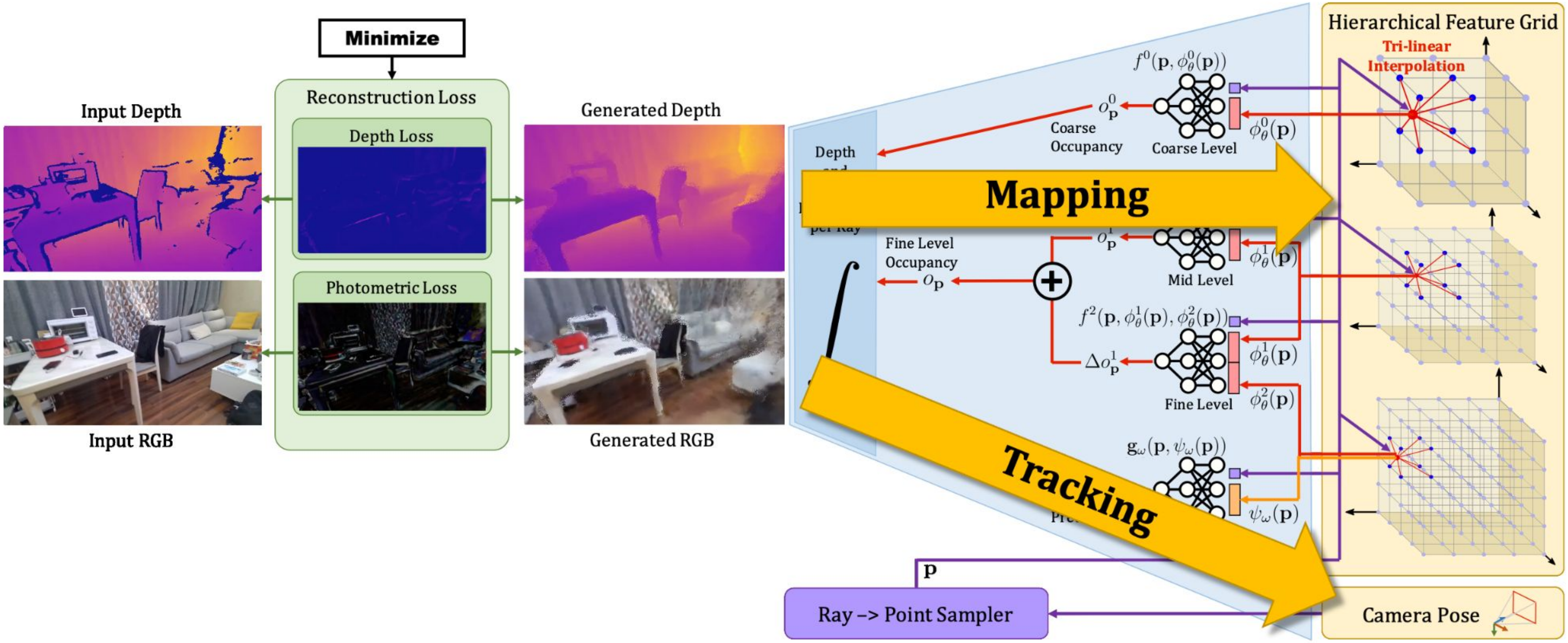
NICE-SLAM Overview



NICE-SLAM Overview



NICE-SLAM Overview



Results

ScanNet Dataset

iMAP NICE-SLAM

	FLOPs [$\times 10^3$]↓	Tracking [ms]↓	Mapping [ms]↓
iMAP [42]	443.91	101	448
NICE-SLAM (Ours)	**104.16**	**47**	**130**

Computations & Runtime

Results

Large Apartment with Multiple Rooms

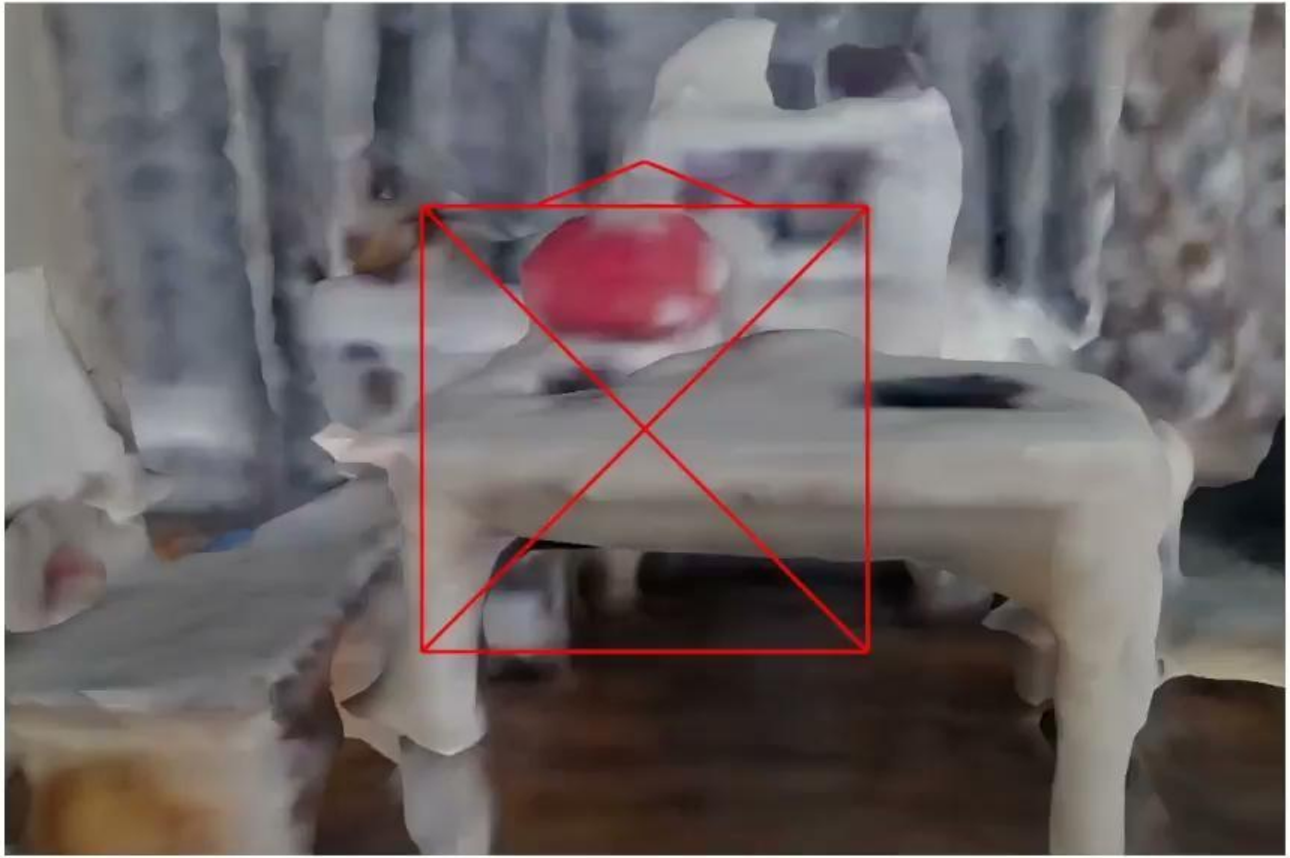
Input Depth Generated Depth

Residual

Input RGB Generated RGB

Residual

2X SPEED



Conclusion

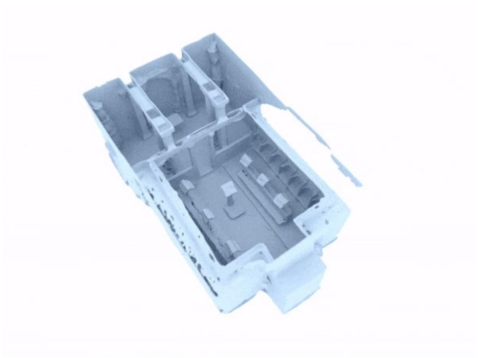
- Hierarchical Encoding + Implicit decoders -> Scalable scene representations
- Show decent results on large indoor scenes
- Do not have forgetting problems

Limitation

- Geometry & colors are OK, but far from satisfactory
- Camera tracking is worse than traditional methods
- Cannot scale up to outdoor scenes

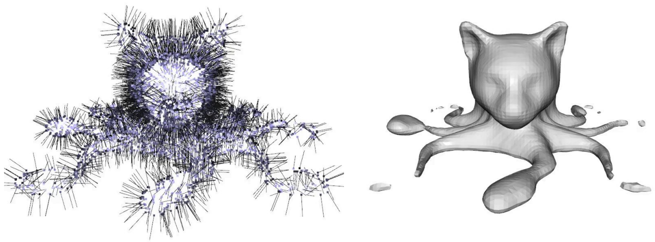
Thank You!

ConvONet
ECCV'20 (Spotlight)



https://pengsongyou.github.io/conv_onet

Shape As Points
NeurIPS'21 (Oral)



https://pengsongyou.github.io/sap

NICE-SLAM
arXiv'21

https://pengsongyou.github.io/nice-slam