Dive into Neural Explicit-Implicit 3D Representations and Their Applications

Songyou Peng
ETH Zurich and Max Planck Institute for Intelligent Systems

Symposium of Geometry Processing
July 2, 2023

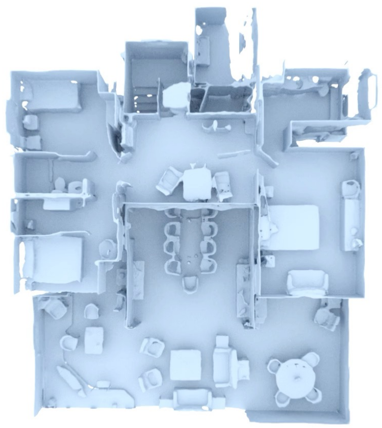

Hardcore Graphics Guys
Me

Who Am I?

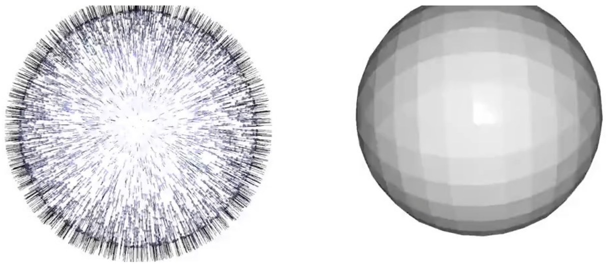
- **Final-year PhD Student**
  - Marc Pollefeys
  - Andreas Geiger
- **Internships during PhD**
  - 2021: Michael Zollhoefer
  - 2022: Tom Funkhouser
- Before PhD, worked in Singapore, and interned at INRIA and TUM

pengsongyou.github.io

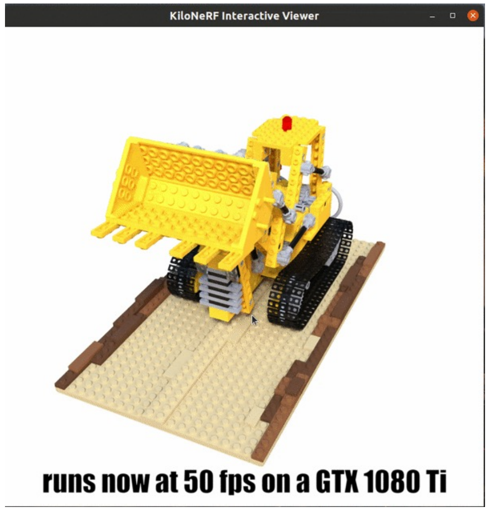
My PhD Topics: Neural Scene Representations
for 3D reconstruction and 3D scene understanding



Convolutional Occupancy Nets
ECCV 2020 (Spotlight)

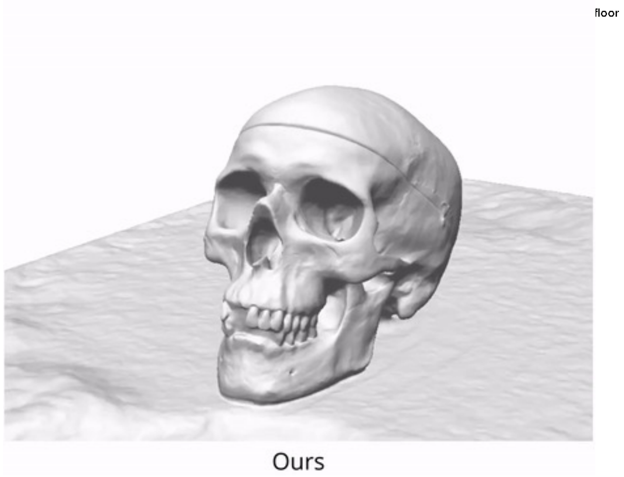


Shape As Points
NeurIPS 2021 (Oral)

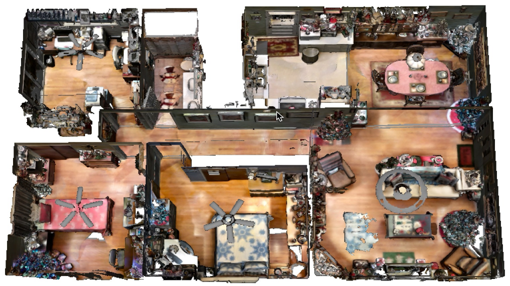


KiloNeRF
ICCV 2021

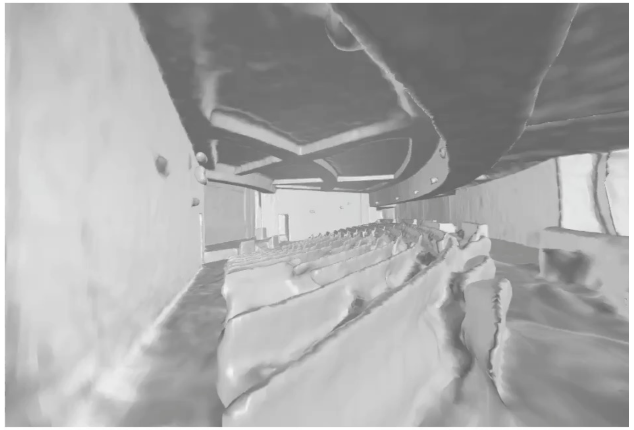
NICE-SLAM
CVPR 2022



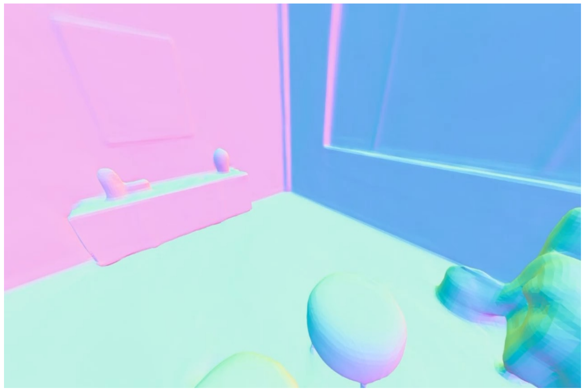
Ours
UNISURF
ICCV 2021 (Oral)



OpenScene
CVPR 2023



Ours
MonoSDF
NeurIPS 2022



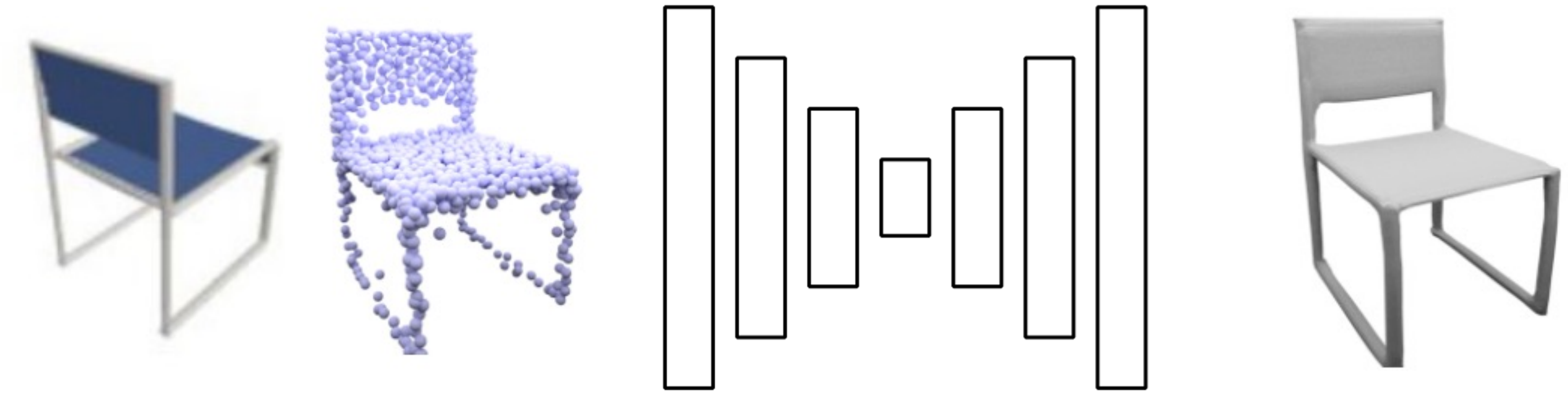
NICER-SLAM
arXiv 2023

In this talk…

- Introduce explicit, implicit, and hybrid 3D scene representations
- Explore the evolution of neural explicit-implicit representations in the field of 3D reconstruction, neural rendering, visual SLAM…
- Discuss seminal works that have advanced the research in computer vision!


ARE YOU
READY ?
makeameme.org

Learning-based 3D Surface Reconstruction



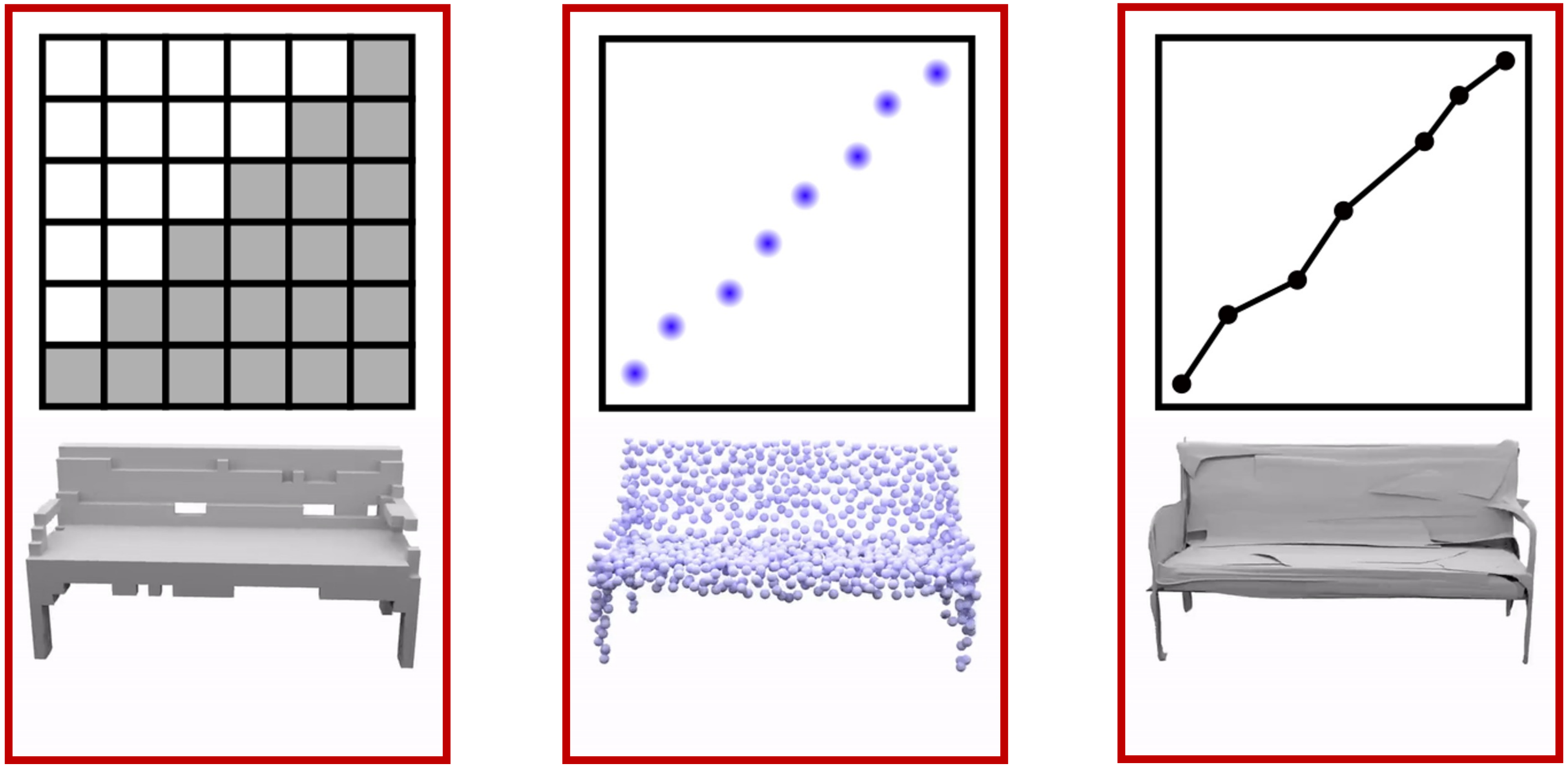
Input
(Images/Point Clouds/…)

Neural Network

3D
Reconstruction

What is a good **3D shape representation?**

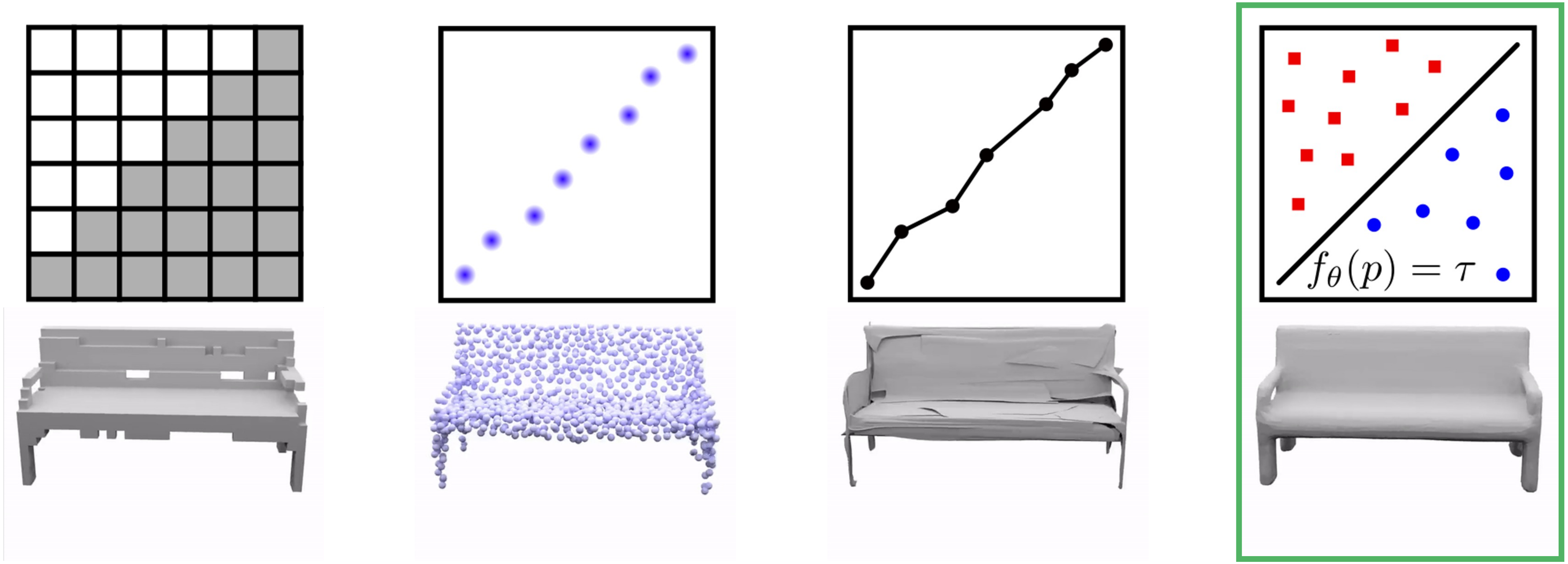
3D Representations



- Traditional Explicit Representations ⇒ **Discrete**

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

3D Representations



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

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

3 seminal papers came out at the same CVPR!

Occupancy Networks: Learning 3D Reconstruction in Function Space

Lars Mescheder[1] Michael Oechsle[1,2] Michael Niemeyer[1] Sebastian Nowozin[3†] Andreas Geiger[1]
[1]Autonomous Vision Group, MPI for Intelligent Systems and University of Tübingen
[2]ETAS GmbH, Stuttgart
[3]Google AI Berlin

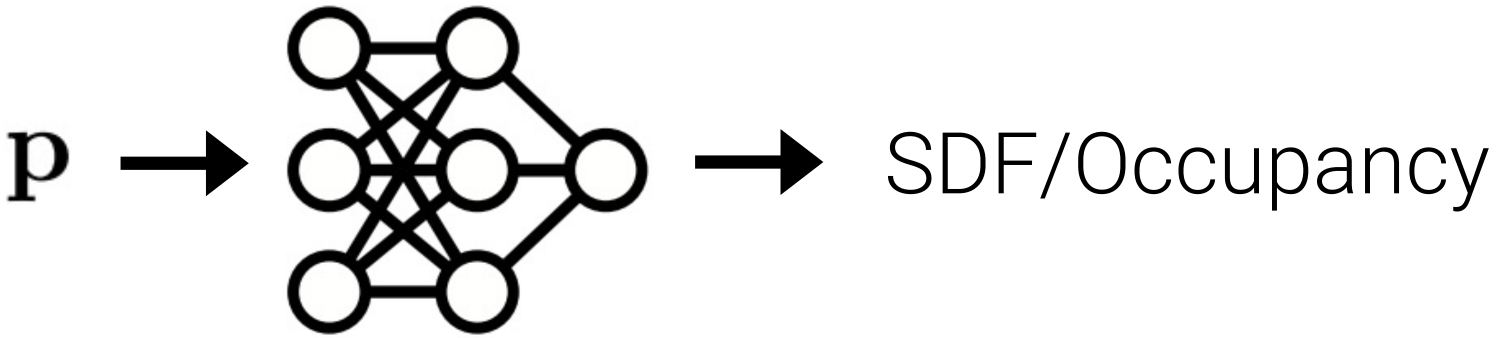
DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation

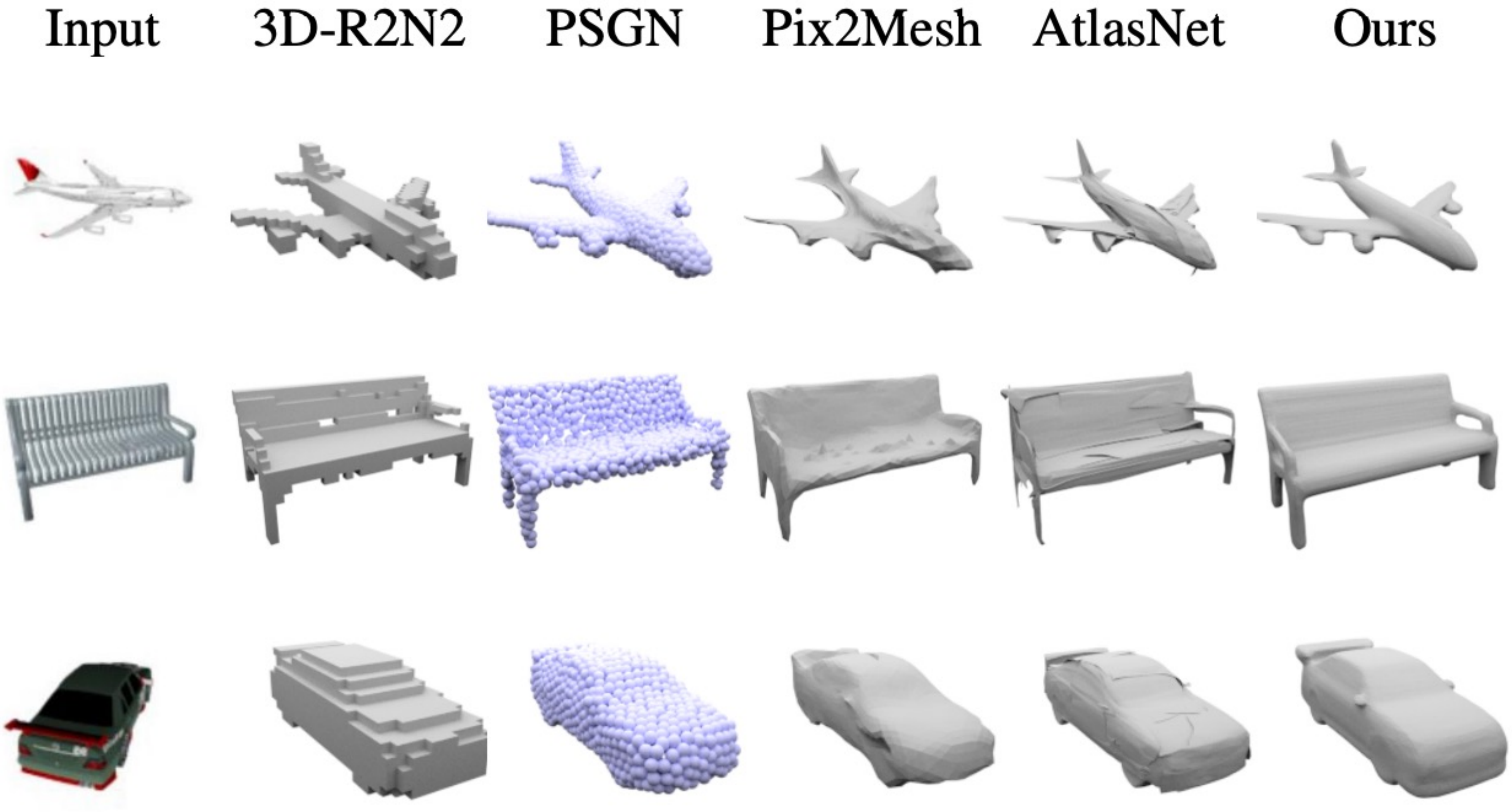
Jeong Joon Park[1,3†] Peter Florence [2,3†] Julian Straub[3] Richard Newcombe[3] Steven Lovegrove[3]
[1]University of Washington [2]Massachusetts Institute of Technology [3]Facebook Reality Labs

Learning Implicit Fields for Generative Shape Modeling

Zhiqin Chen
Simon Fraser University
zhiqinc@sfu.ca

Hao Zhang
Simon Fraser University
haoz@sfu.ca

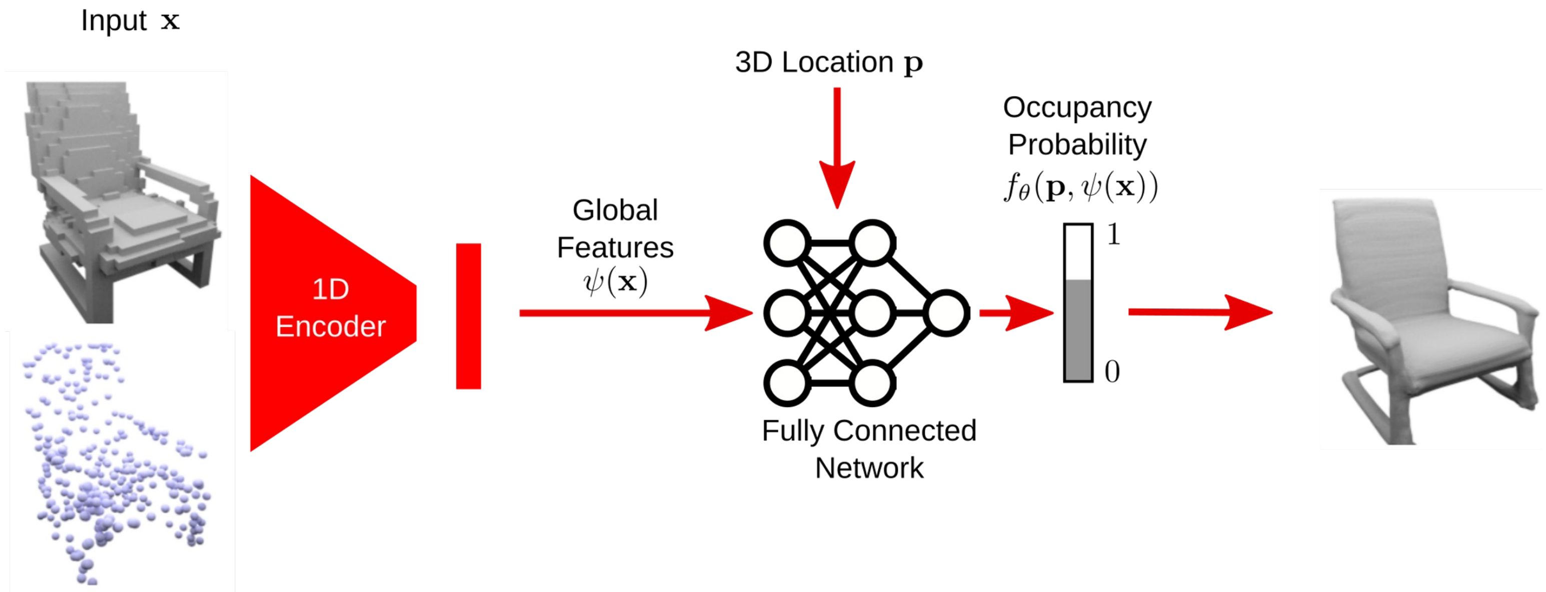




Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

Limitations

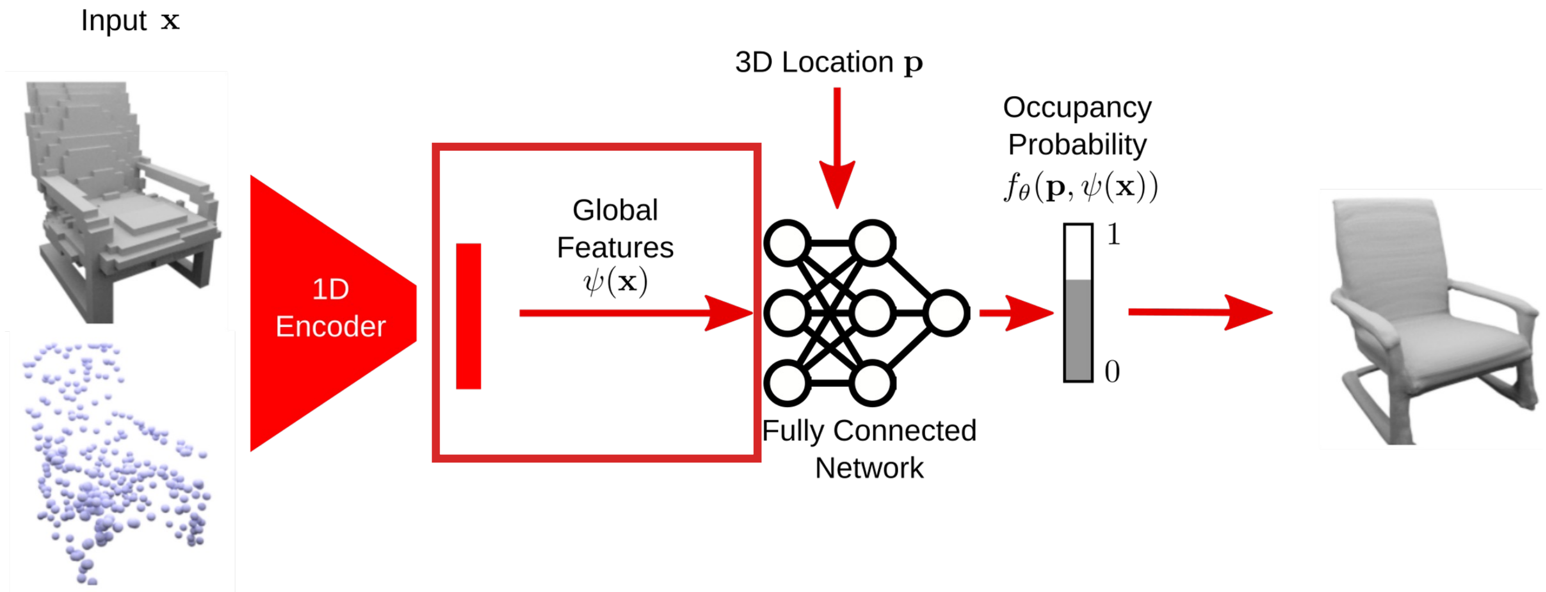
Structure of neural implicit representations:



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

Limitations

Structure of neural implicit representations:

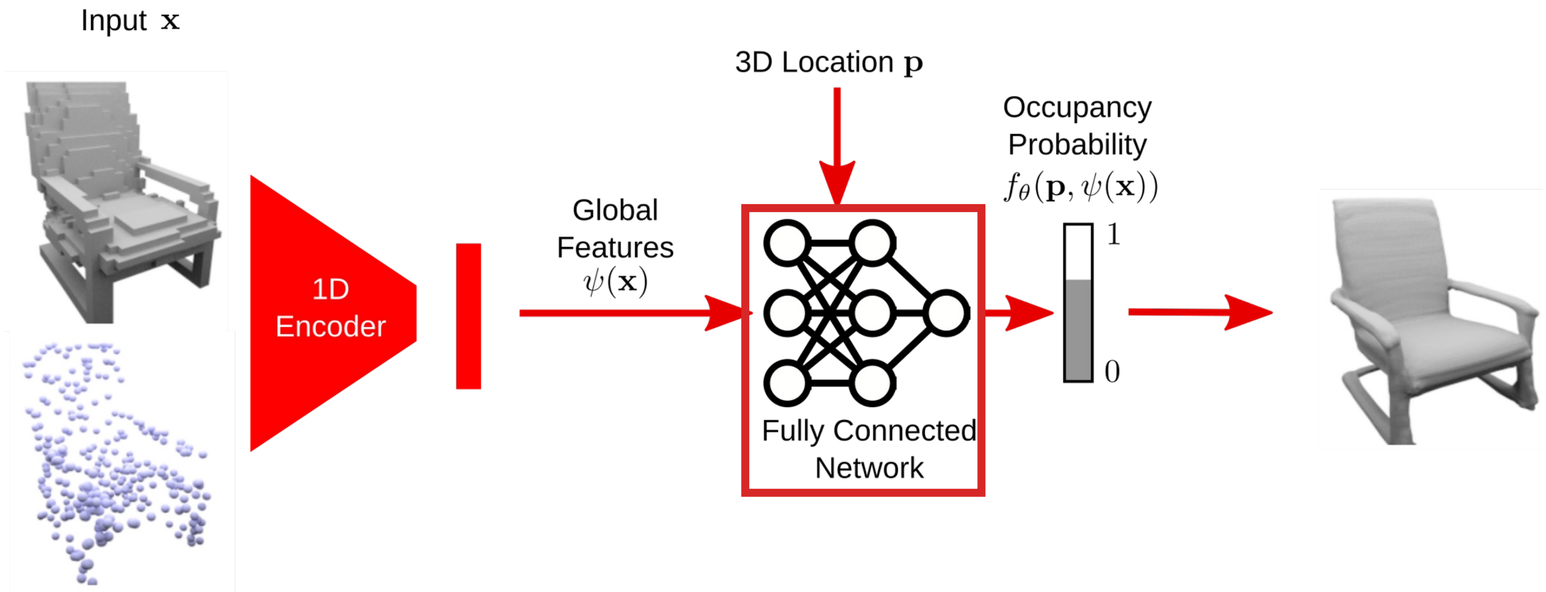


- Global latent code ⇒ **overly smooth geometry**

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

Limitations

Structure of neural implicit representations:

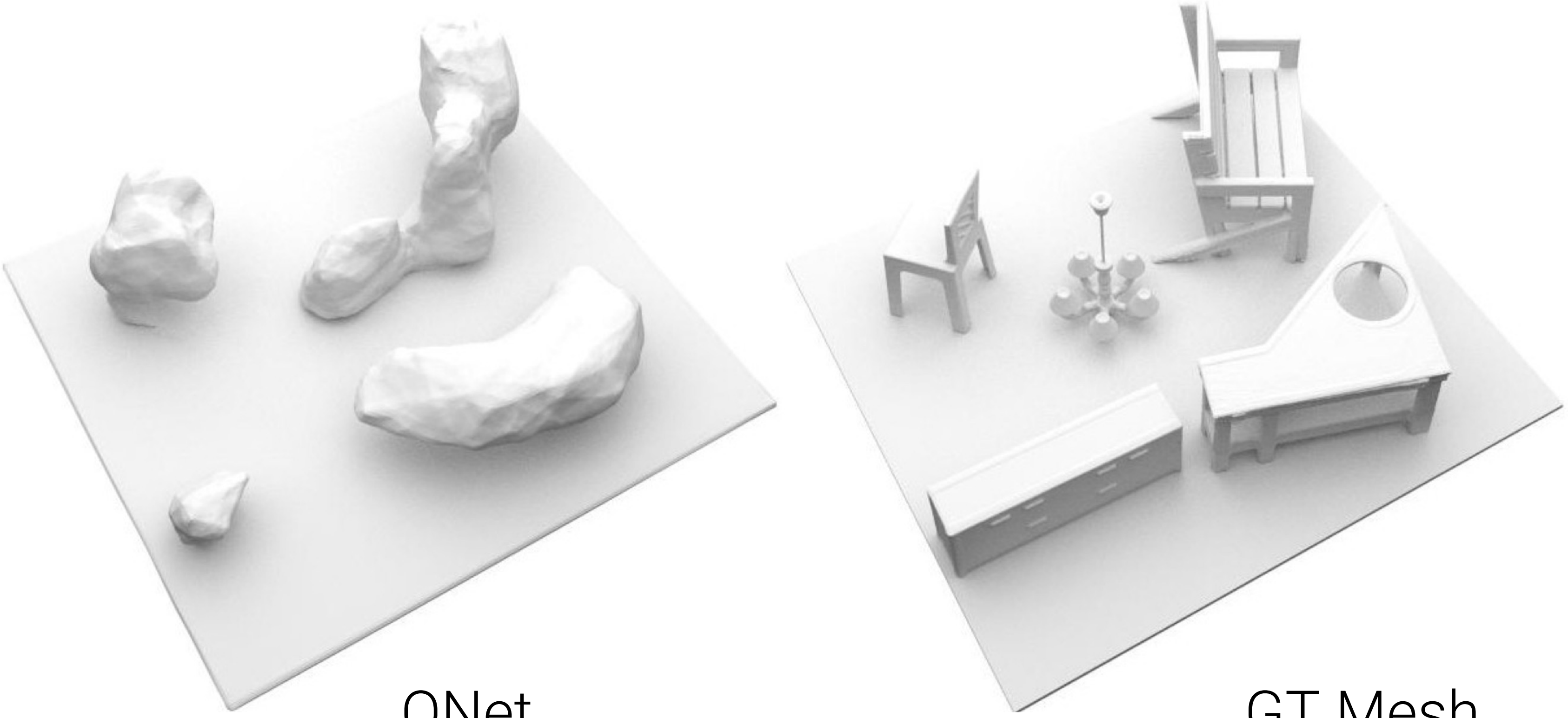


- Global latent code ⇒ **overly smooth geometry**
- Fully-connected architecture ⇒ **no translation equivariance**

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**?

ETH zürich

ECCV'20 ONLINE
23-28 AUGUST 2020

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

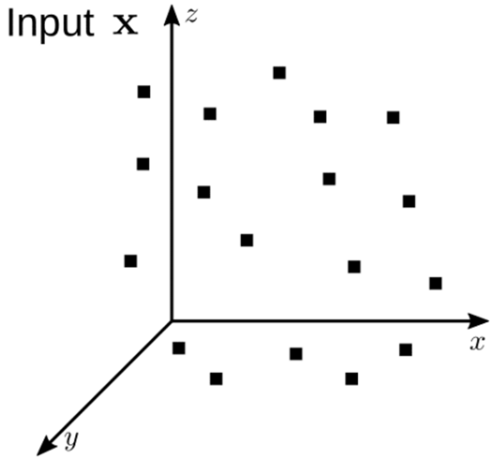
Convolutional Occupancy Networks

Songyou Peng | Michael Niemeyer | Lars Mescheder | Marc Pollefeys | Andreas Geiger

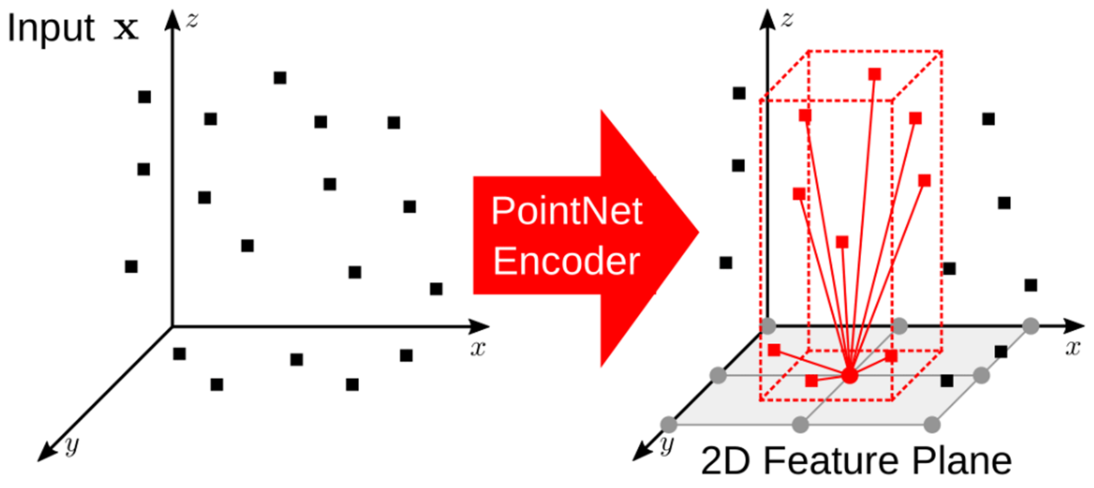




Main Idea

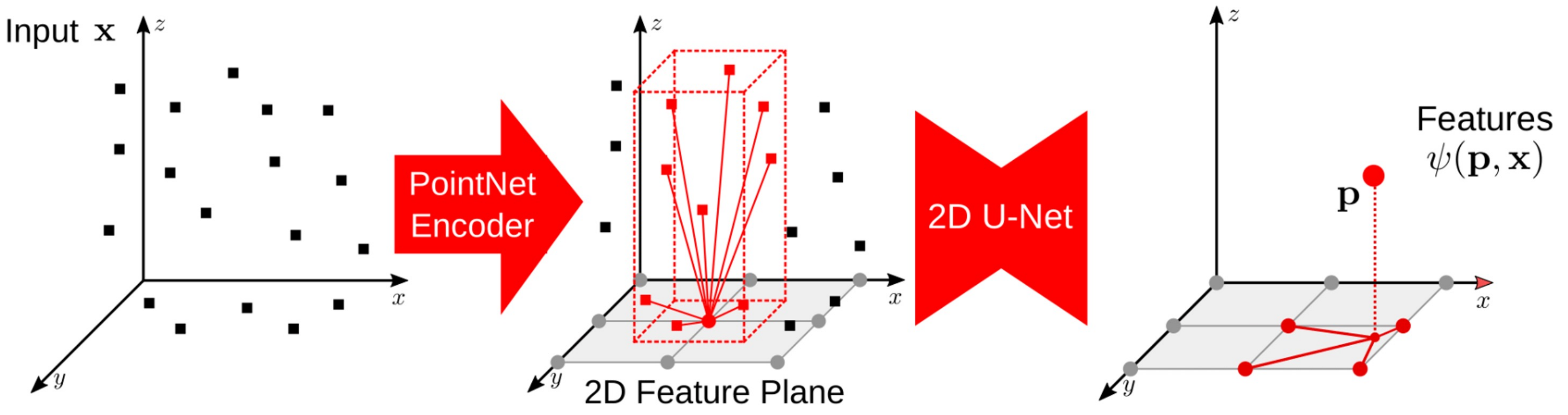


Main Idea



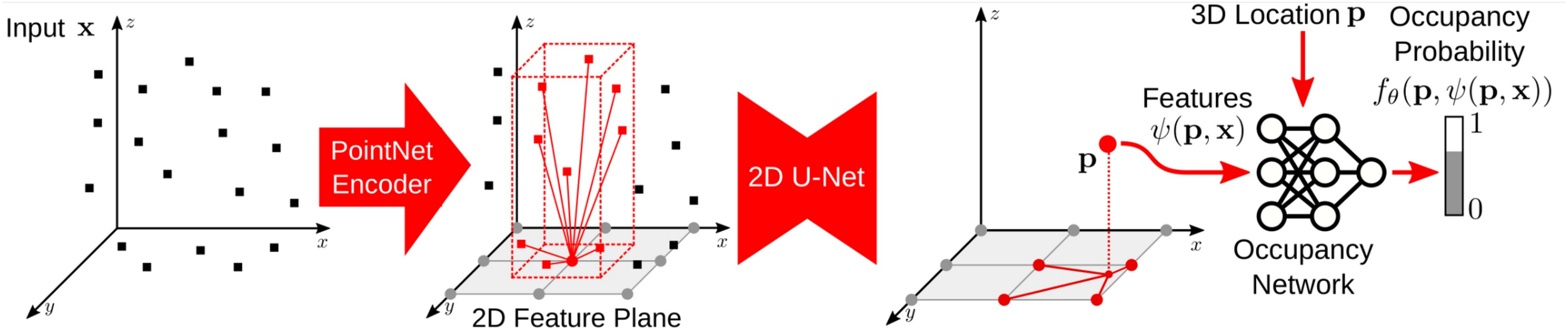
- **2D Plane Encoder**: Use a local PointNet to process input, project onto canonical plane

Main Idea



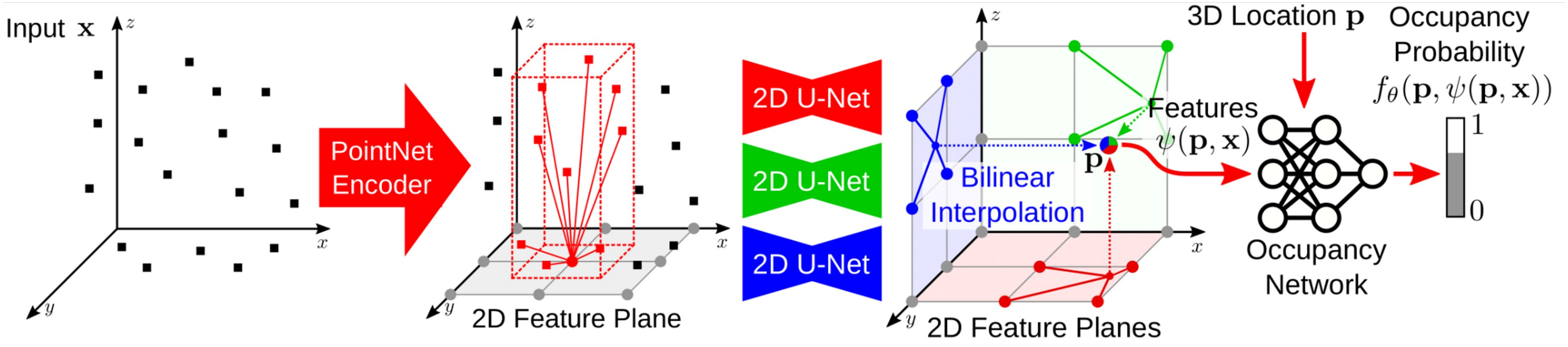
- **2D Plane Encoder**: Use a local PointNet to process input, project onto canonical plane
- **2D Plane Decoder**: Processed by U-Net, query features via bilinear interpolation

Main Idea



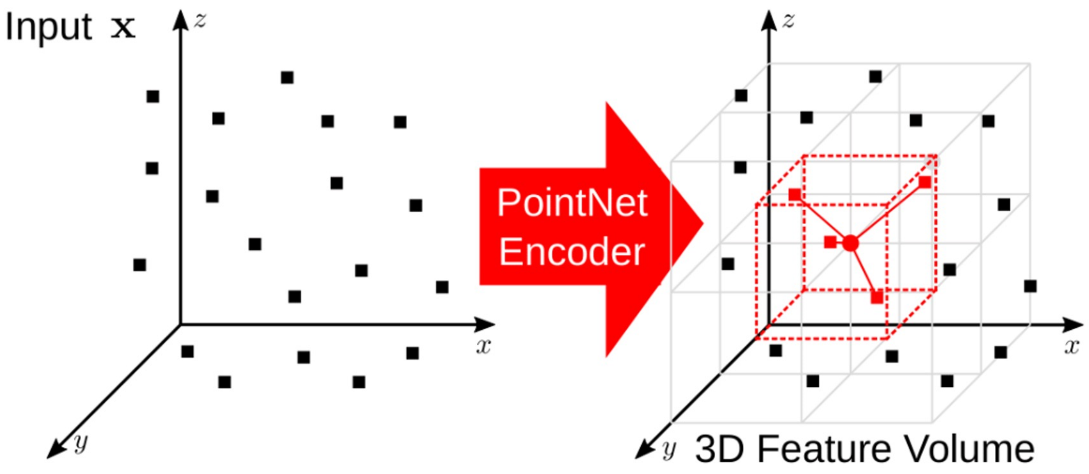
- **2D Plane Encoder**: Use a local PointNet to process input, project onto canonical plane
- **2D Plane Decoder**: Processed by U-Net, query features via bilinear interpolation
- **Occupancy Readout**: Shallow occupancy network $f_\theta(\cdot)$

Main Idea



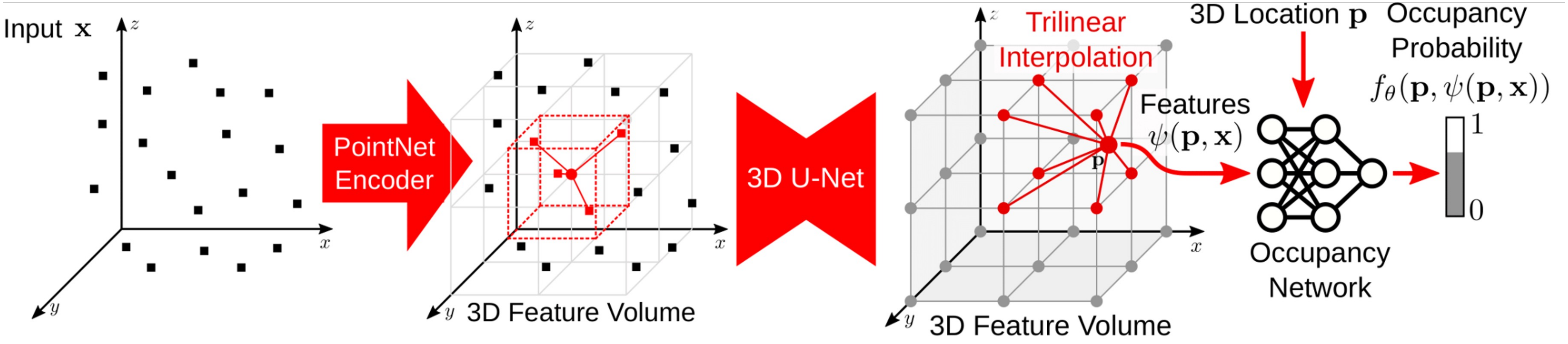
- **2D Plane Encoder**: Use a local PointNet to process input, project onto **3-canonical planes**
- **2D Plane Decoder**: Processed by U-Net, query features via bilinear interpolation
- **Occupancy Readout**: Shallow occupancy network $f_\theta(\cdot)$

Main Idea – 3D



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

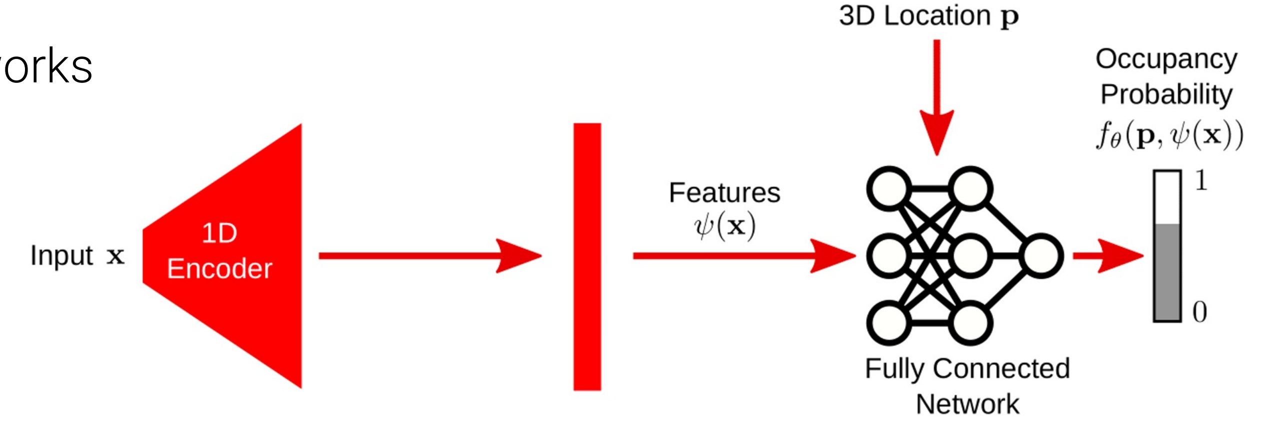
Main Idea – 3D



- **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 $f_\theta(\cdot)$

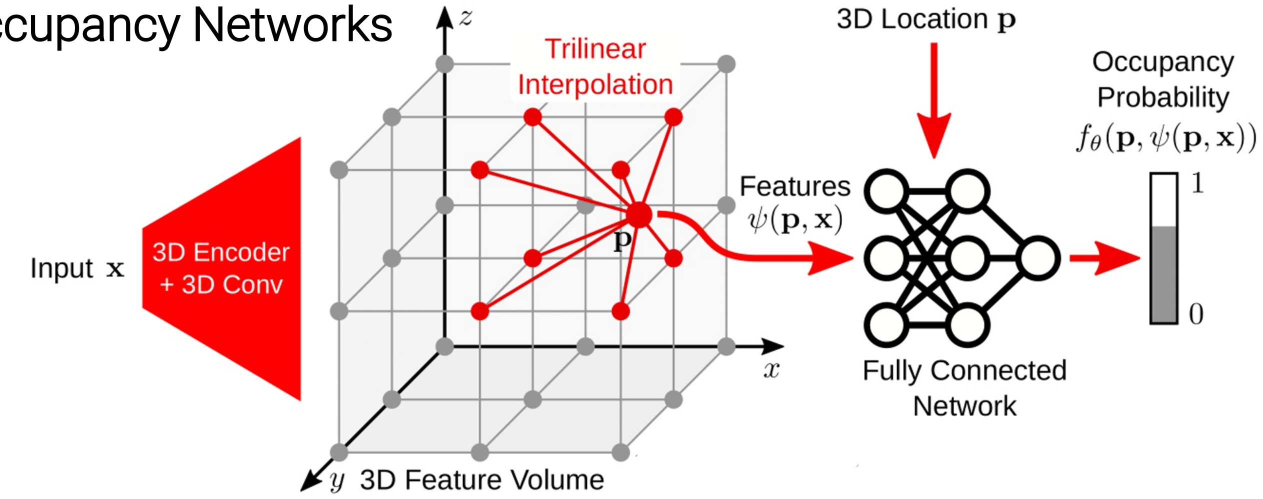
Comparison

Occupancy Networks



- global feature
- heavy FC network
- no translation equivariance

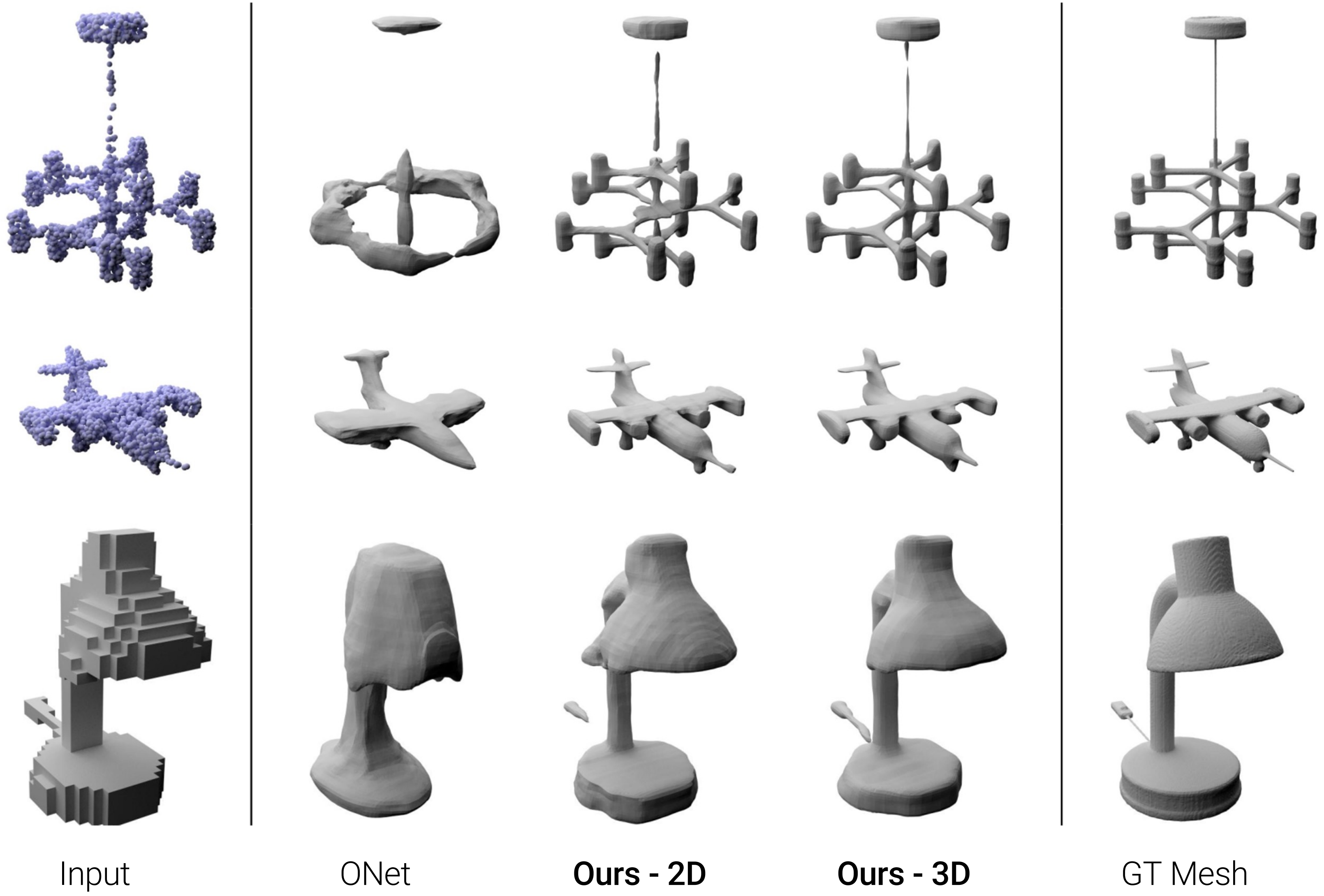
Convolutional Occupancy Networks



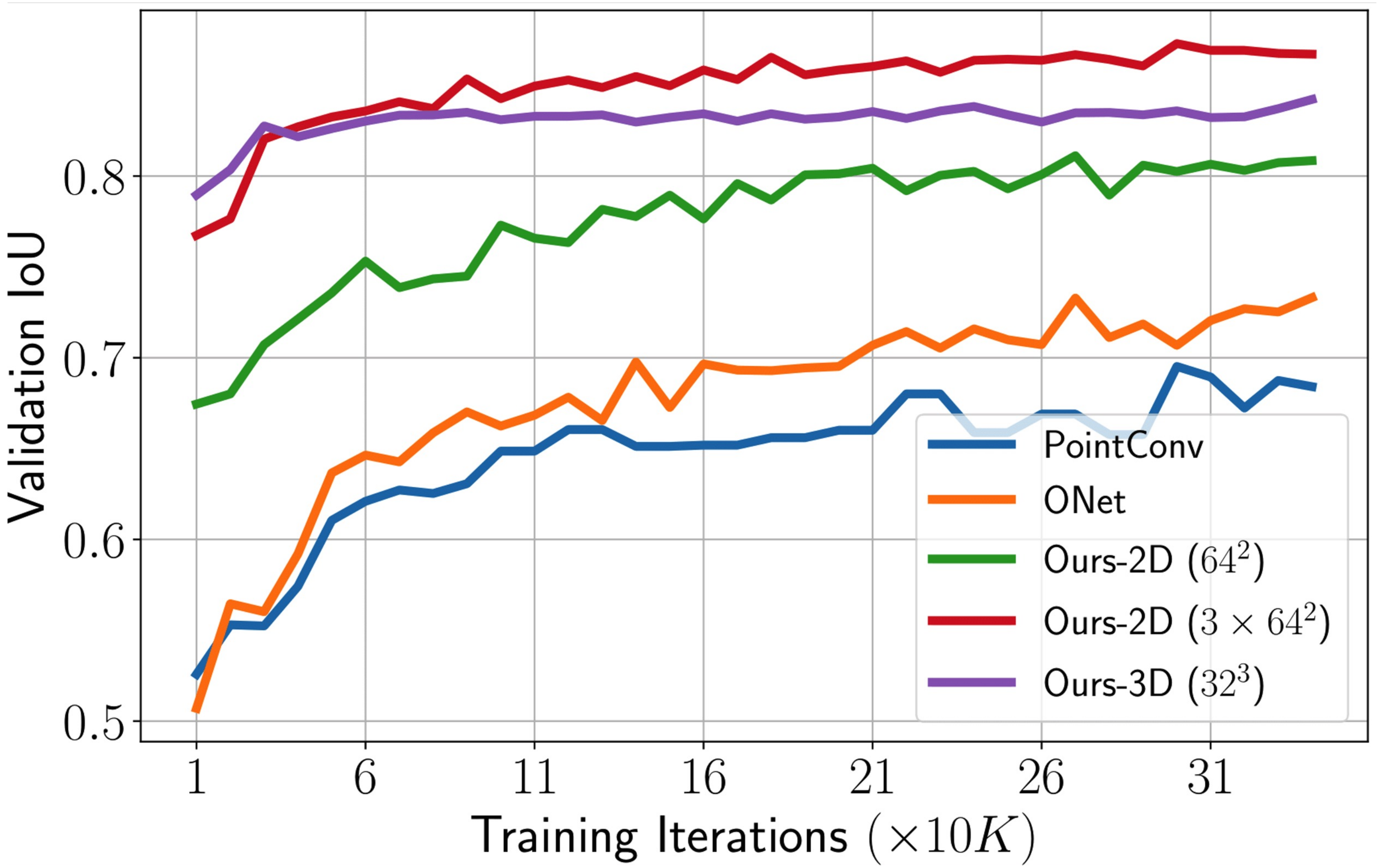
+ local feature
+ shallow FC network
+ translation equivariance

Results

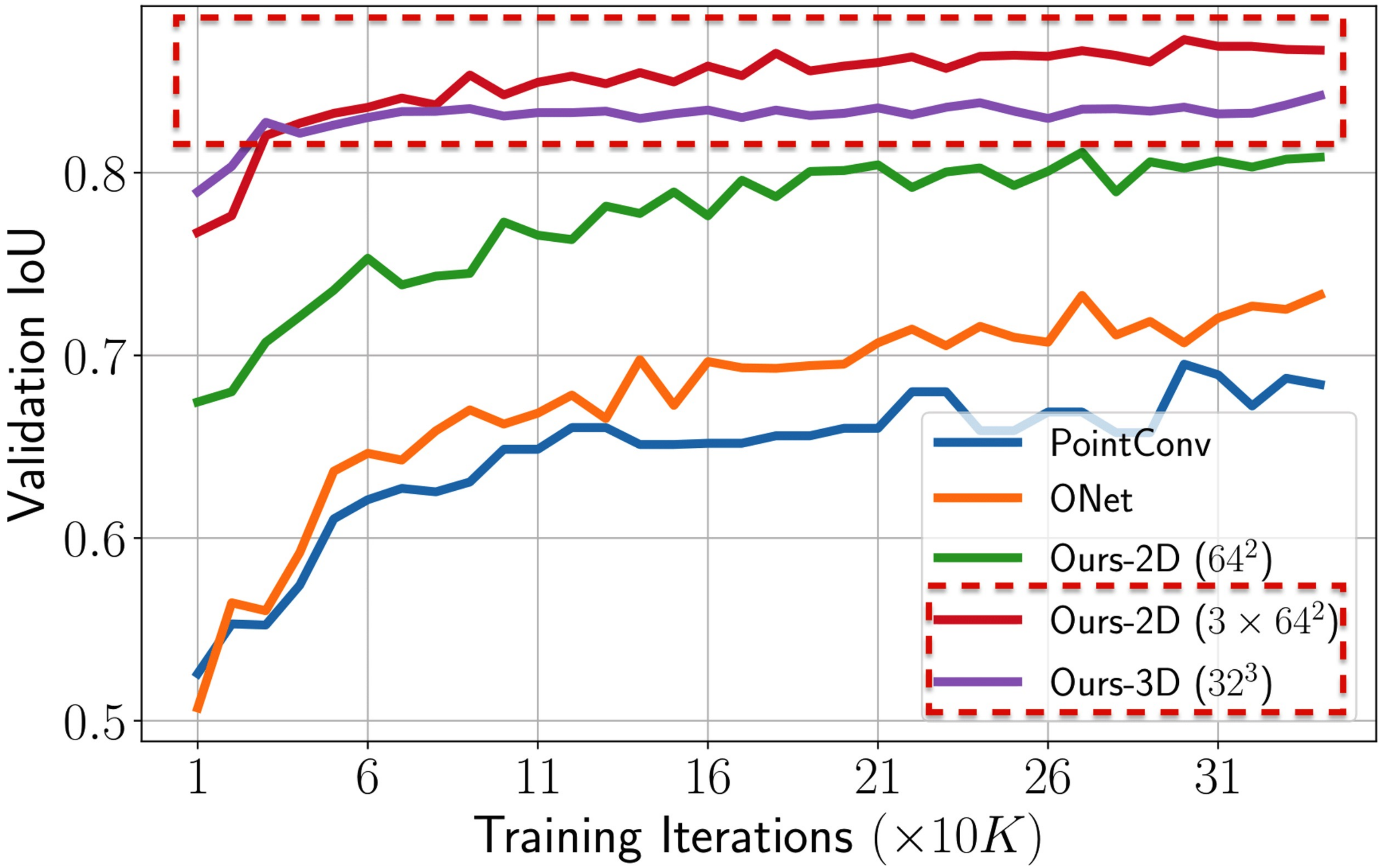
Object-Level Reconstruction



Training Speed

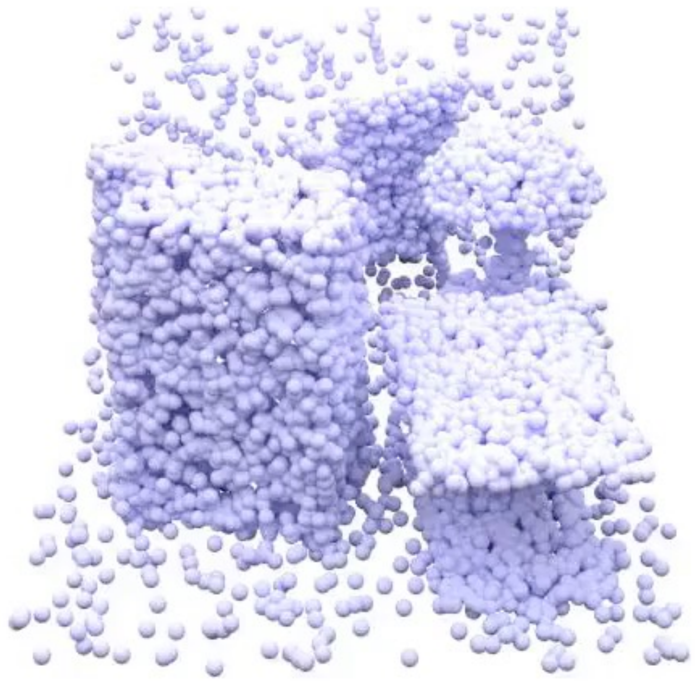


Training Speed



Scene-Level Reconstruction: Synthetic

- Trained and evaluated on synthetic rooms

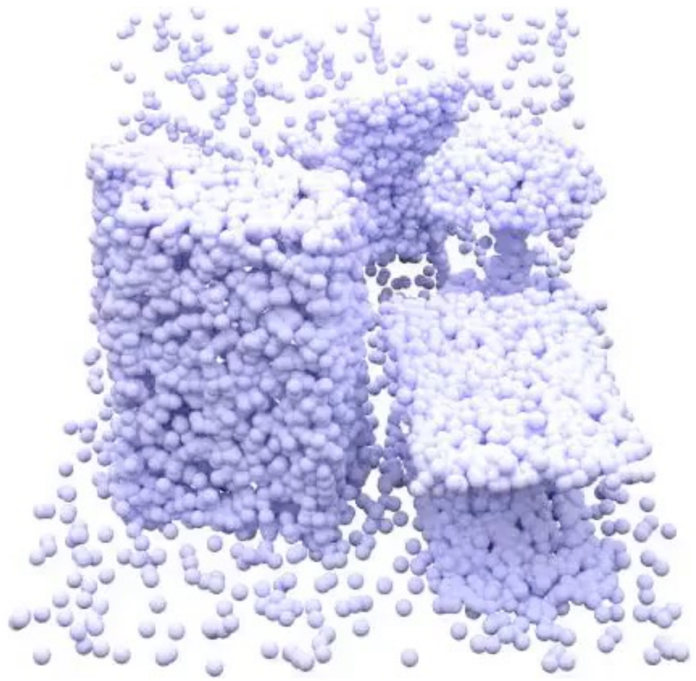


Input

GT Mesh

Scene-Level Reconstruction: Synthetic

- ONet fails on room-level reconstruction

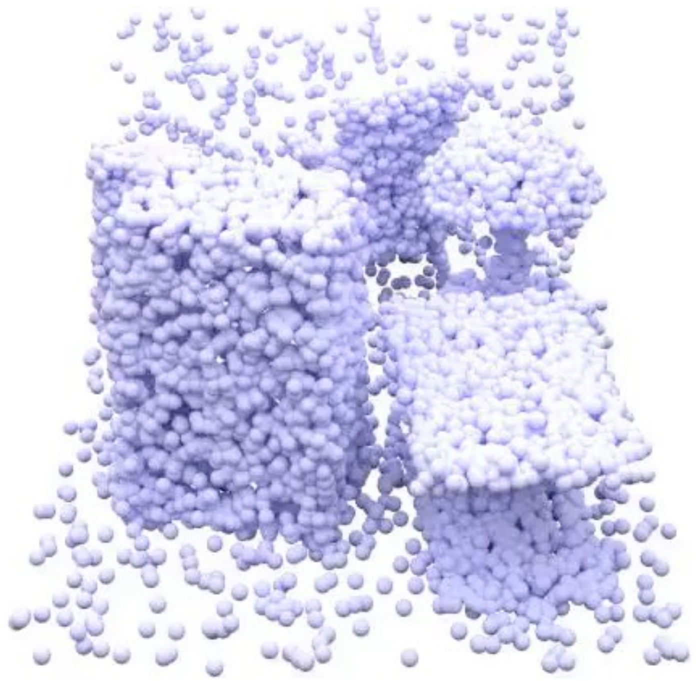


Input

ONet

Scene-Level Reconstruction: Synthetic

- SPSR requires surface normals, output is noisy

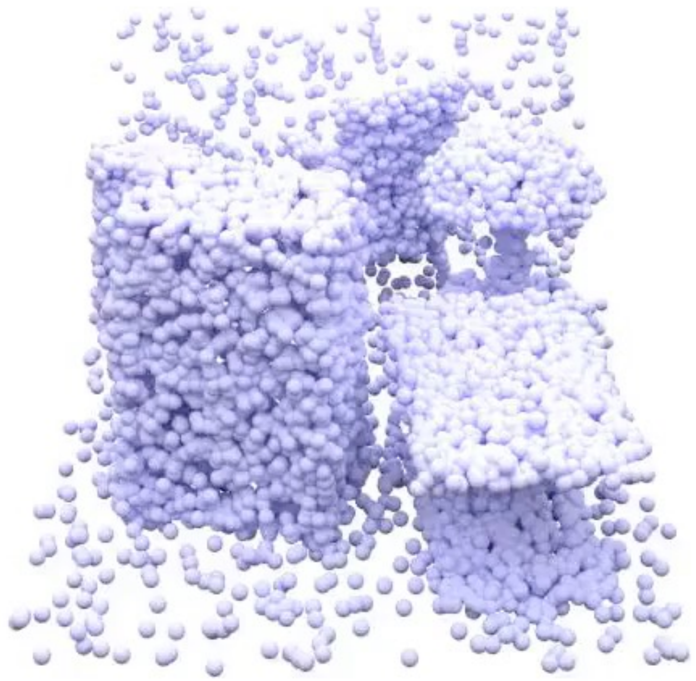


Input

SPSR
(Screened Poisson Surface Reconstruction)

Scene-Level Reconstruction: Synthetic

- Our method preserves better details

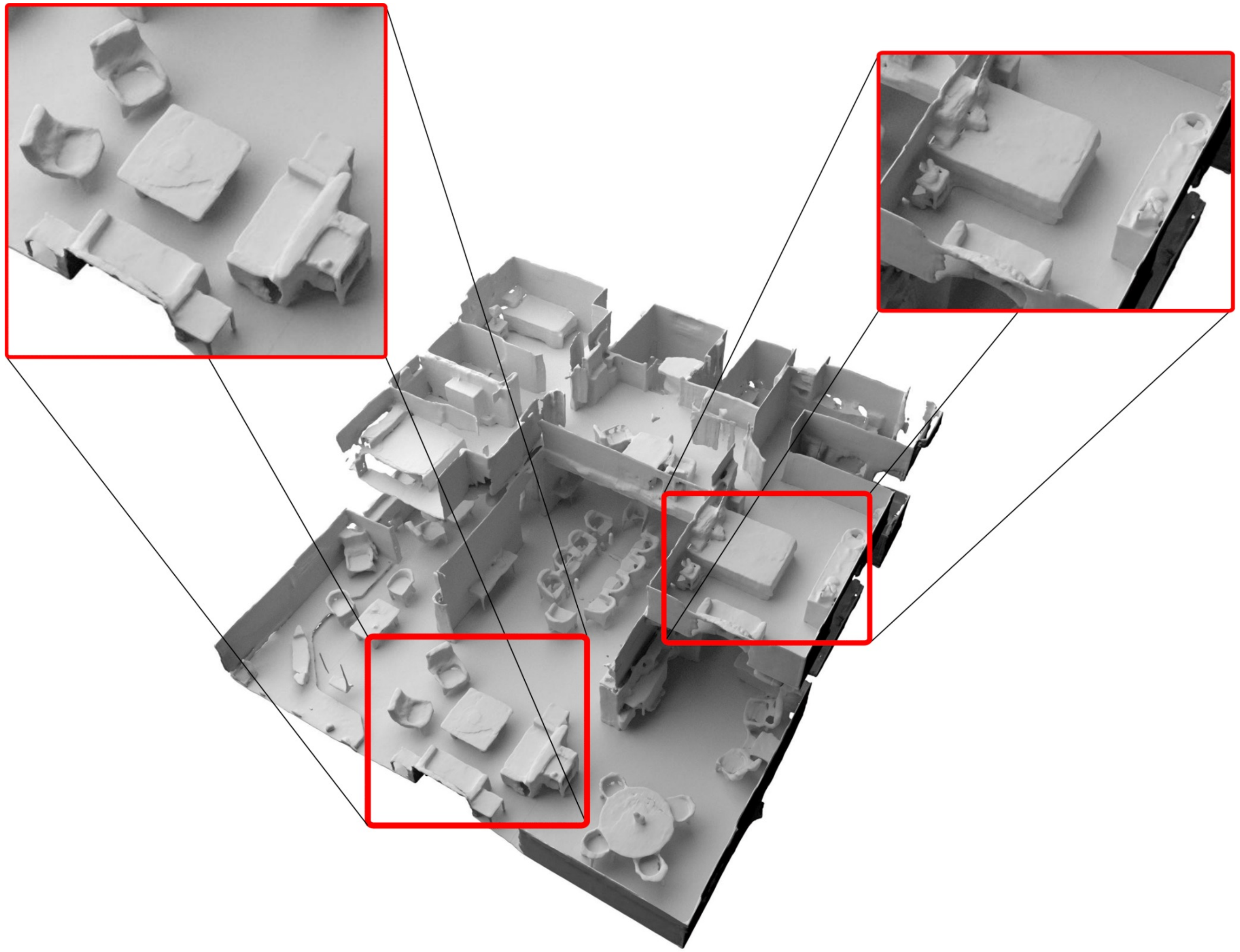


Input

Ours

Large-Scale Reconstruction

Scene size: 15.7m x 12.3m x 4.5m



Our reconstruction output

Results on Matterport3D

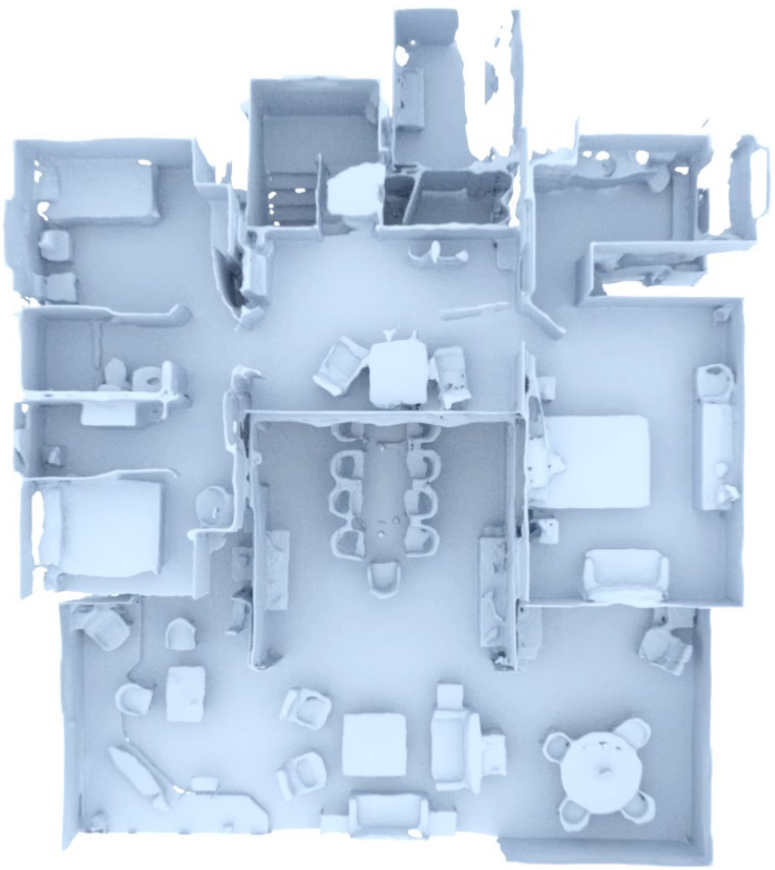
- Fully convolutional model
- Trained on synthetic crops
- Sliding-window evaluation
- Scale to any scene size

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

Take-home Messages

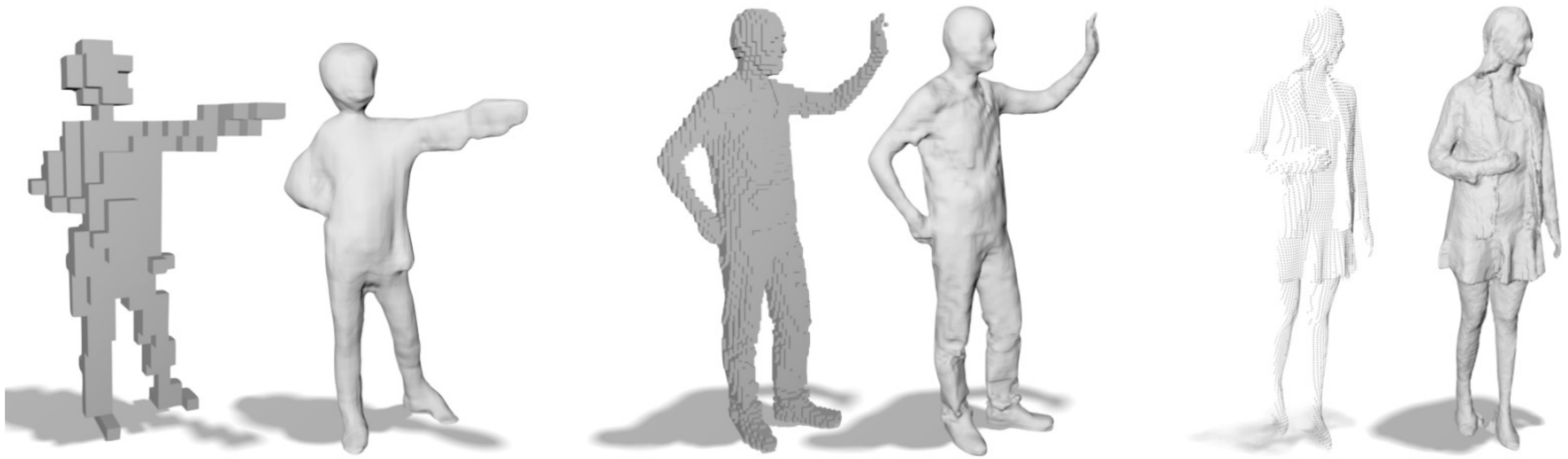
- Introduce 3 different expressive hybrid representations for neural fields
- CNN’s translation equivariance enables to reconstruct large scenes
- The “**tri-plane**” representation became VERY popular
    - Especially in the **NeRF era**, see e.g. EG3D [CVPR’21], TensoRF [ECCV’22]

Limitations

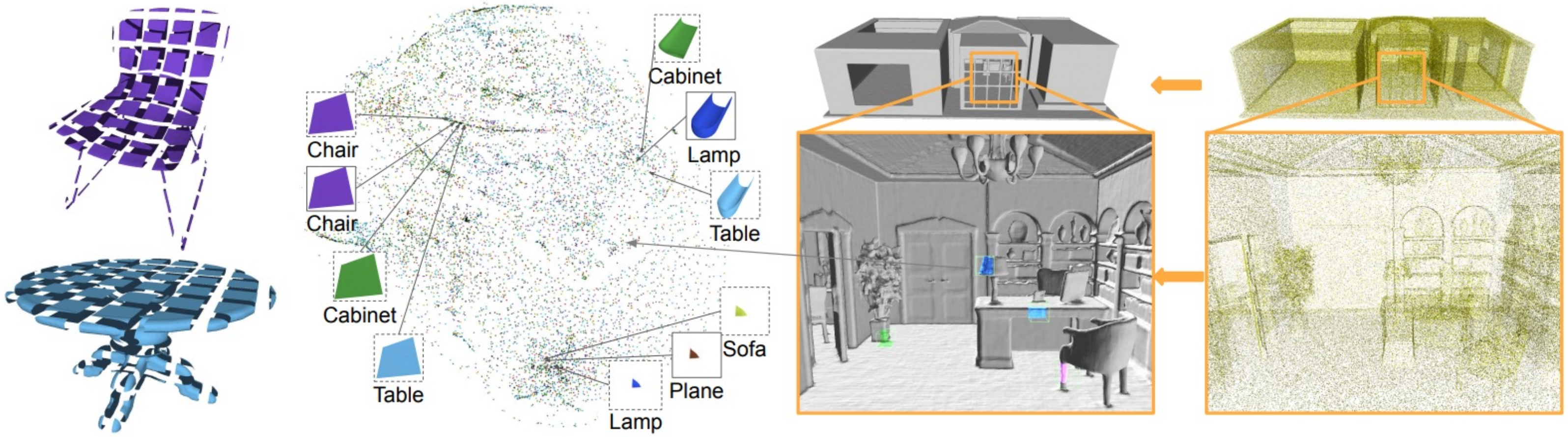
- Not rotational equivariance

Concurrent Works

IF-Net

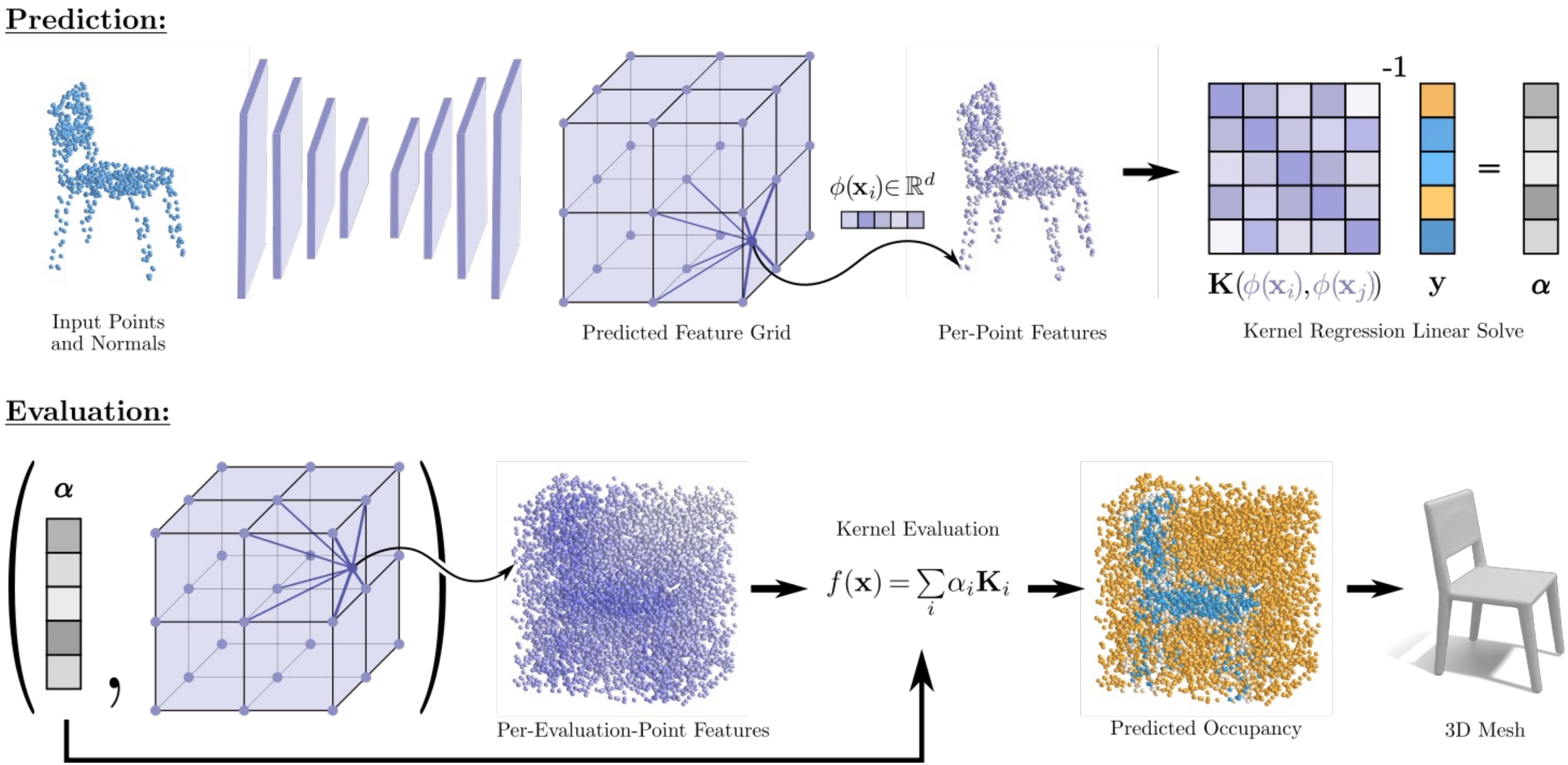


Local implicit Grids



Chibane et al.: Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion. CVPR 2020
Jiang et al.: Local Implicit Grid Representations for 3D Scenes. CVPR 2020

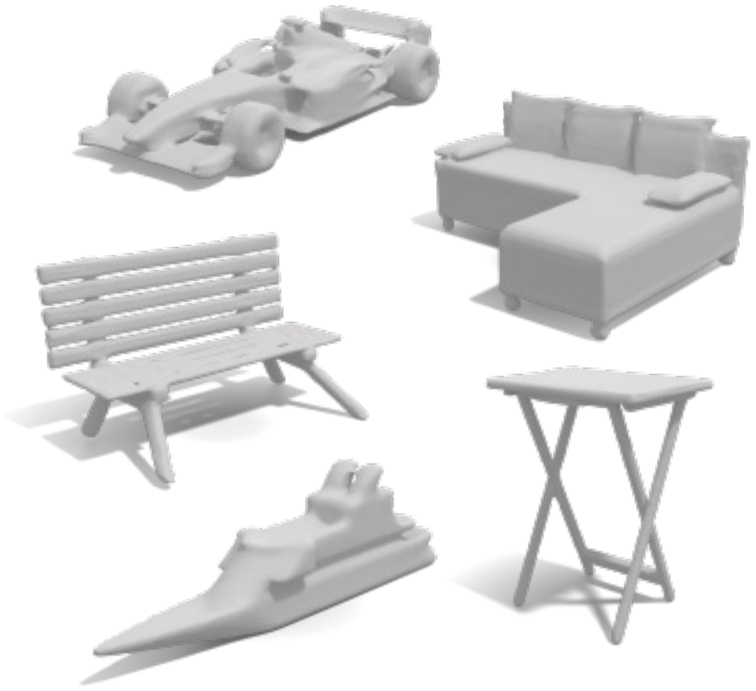
Follow-up works: Neural Kernel Fields (NKF)



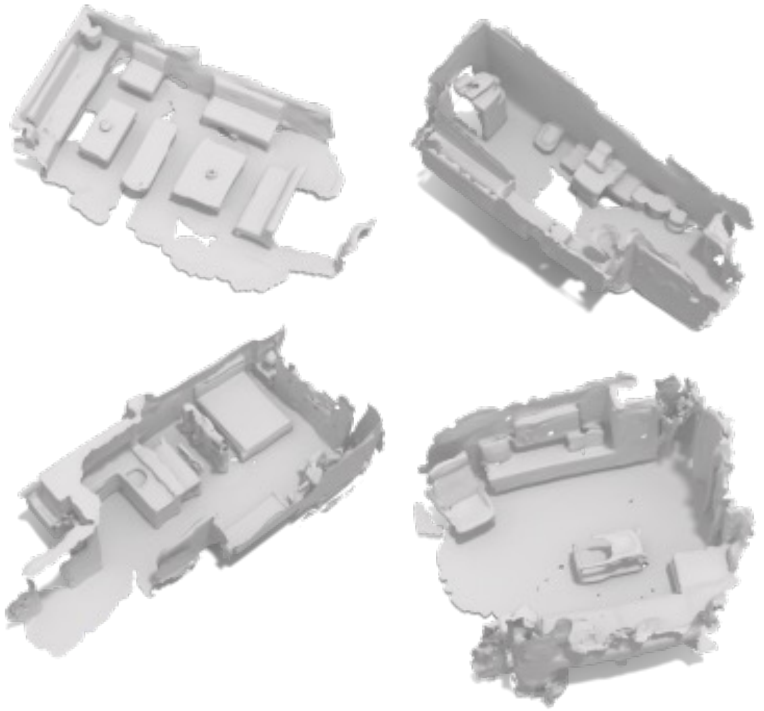
Williams et al.: Neural Fields as Learnable Kernels for 3D Reconstruction. CVPR 2022

Follow-up works: Neural Kernel Fields (NKF)

In-category reconstruction



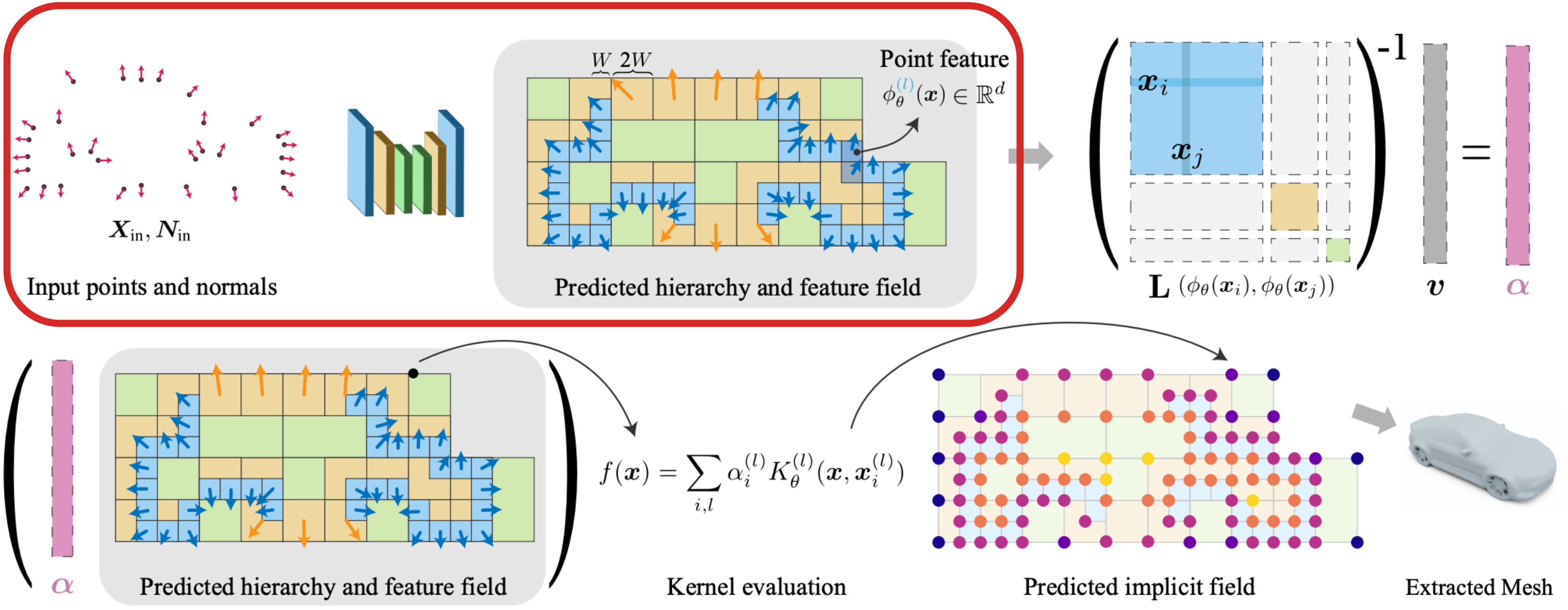
Out-of-category reconstruction



Generalization to scanned scenes

Williams et al.: Neural Fields as Learnable Kernels for 3D Reconstruction. CVPR 2022

Follow-up works: NKSR

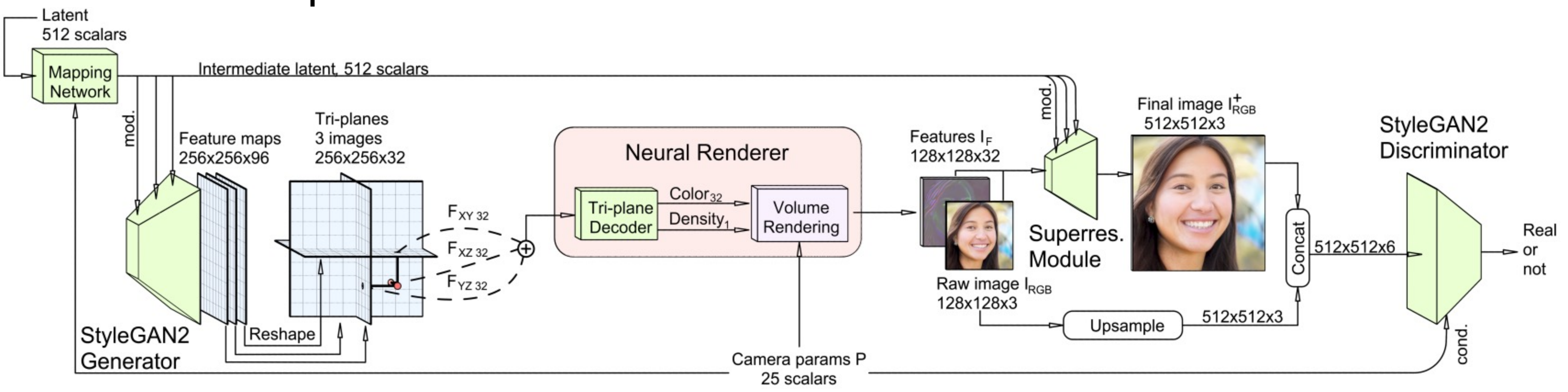


Huang et al.: Neural Kernel Surface Reconstruction. CVPR 2023

Follow-up works: NKSR

Huang et al.: Neural Kernel Surface Reconstruction. CVPR 2023

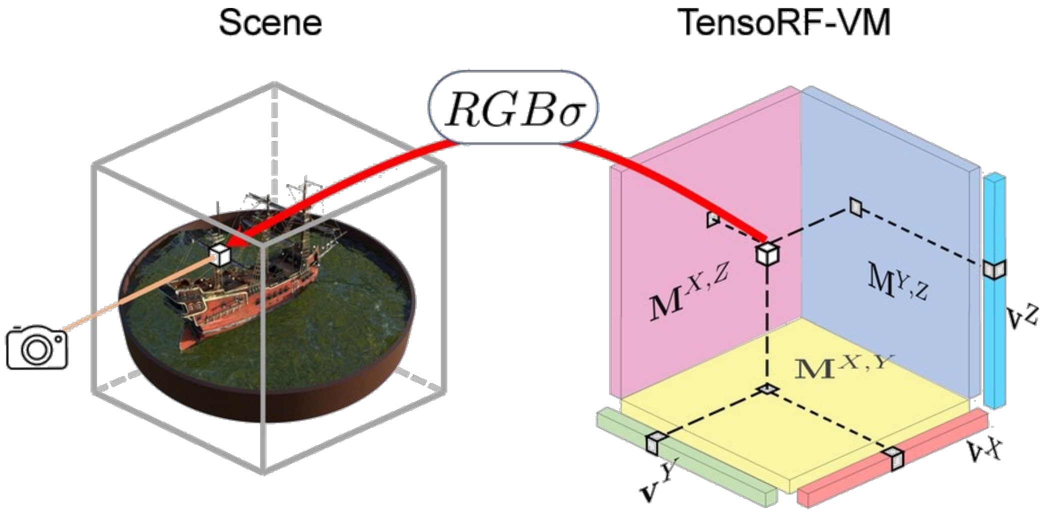
Follow-up works: EG3D

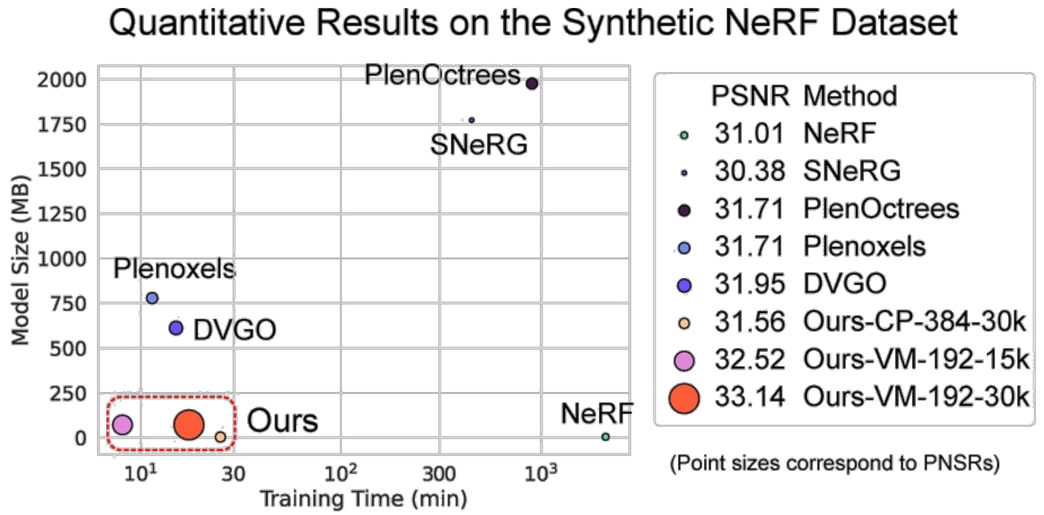


The tri-plane representation enables high-quality 3D-aware view synthesis!

Chan et al.: Efficient Geometry-aware 3D Generative Adversarial Networks. CVPR 2021

Follow-up works: TensoRF



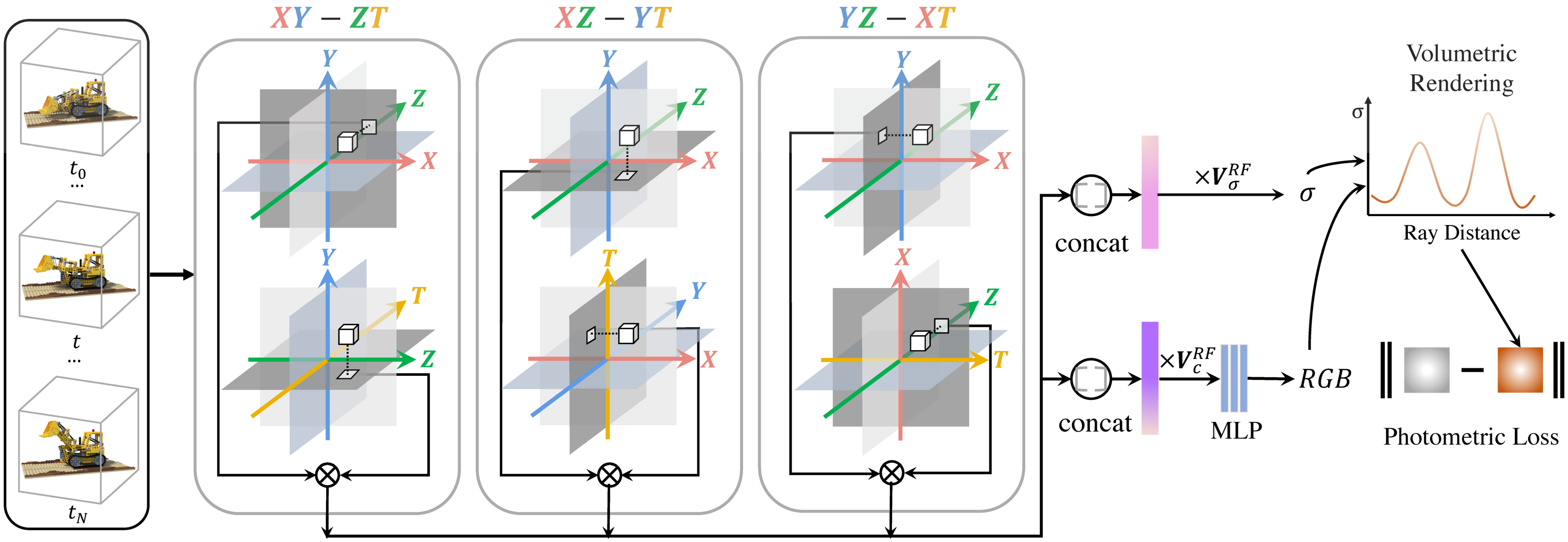





The tri-plane representation speeds up training a memory-efficient NeRF!

Chen et al.: TensoRF: Tensorial Radiance Fields. ECCV 2022

Follow-up works: HexPlane



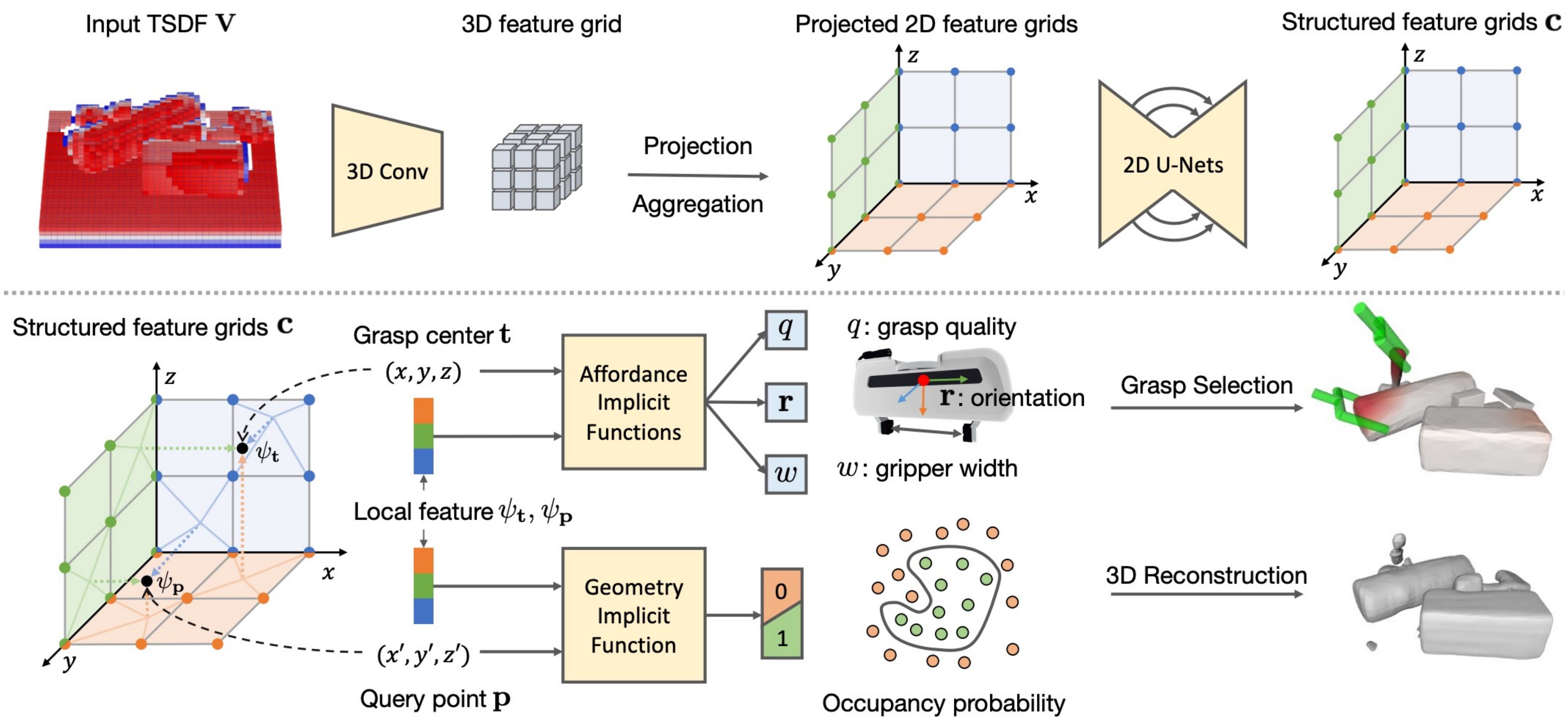
Represent dynamic 3D scenes by decomposing a 4D spacetime grid into six feature planes ⇒ 100x faster training

Cao and Johnson: HexPlane: A Fast Representation for Dynamic Scenes. CVPR 2023

Follow-up works: HexPlane

Cao and Johnson: HexPlane: A Fast Representation for Dynamic Scenes. CVPR 2023

Follow-up works: ACID



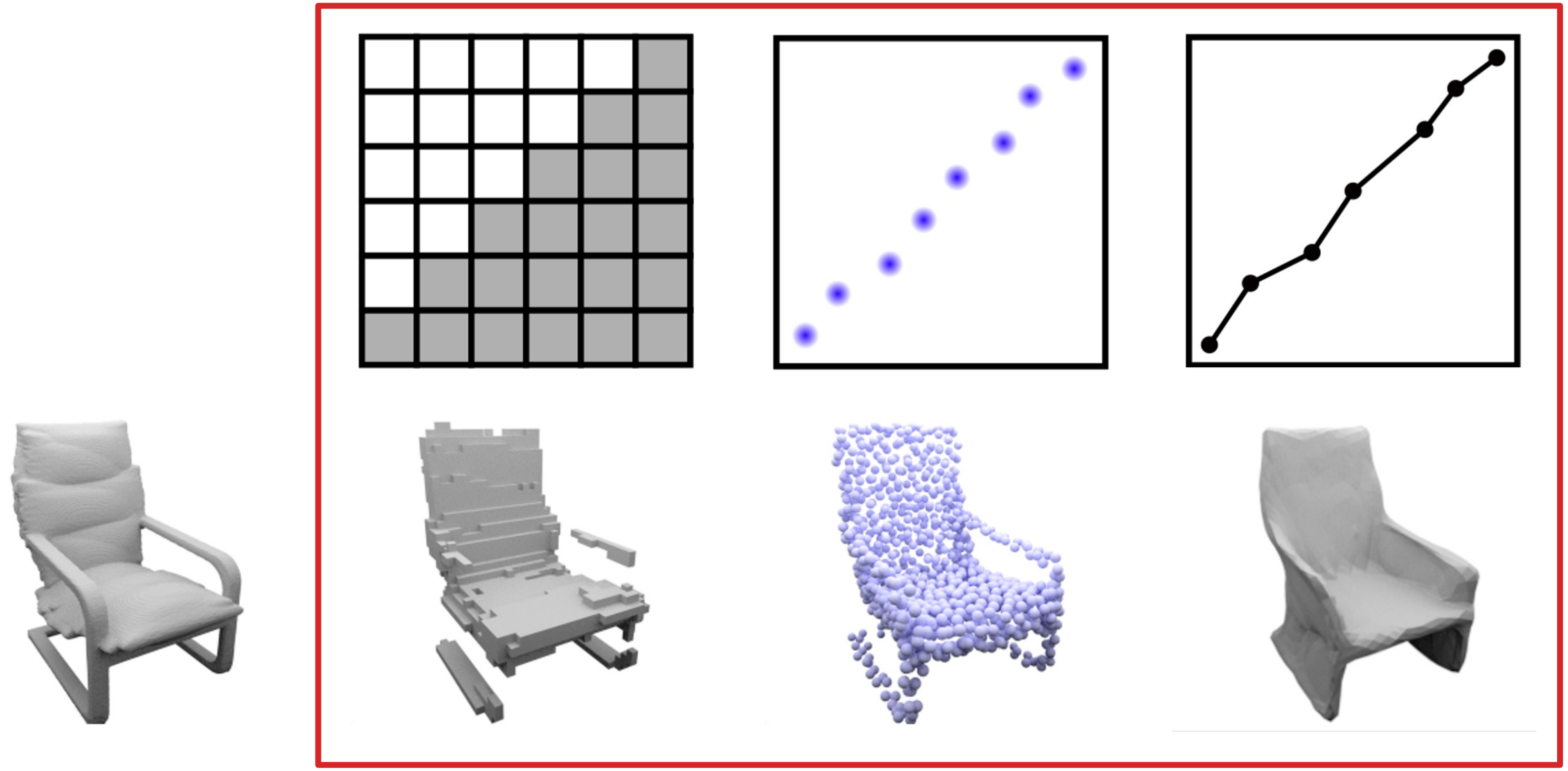
The tri-plane representation is also useful for accurate robot grasping!

Jiang et al.: Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations. RSS 2021
Shen et al.: ACID: Action-Conditional Implicit Visual Dynamics for Deformable Object Manipulation. RSS 2022 (Best Student Paper Finalist)

Let’s take a step back to
3D surface reconstruction…

What is a good **3D shape representation?**

3D Shape Representation

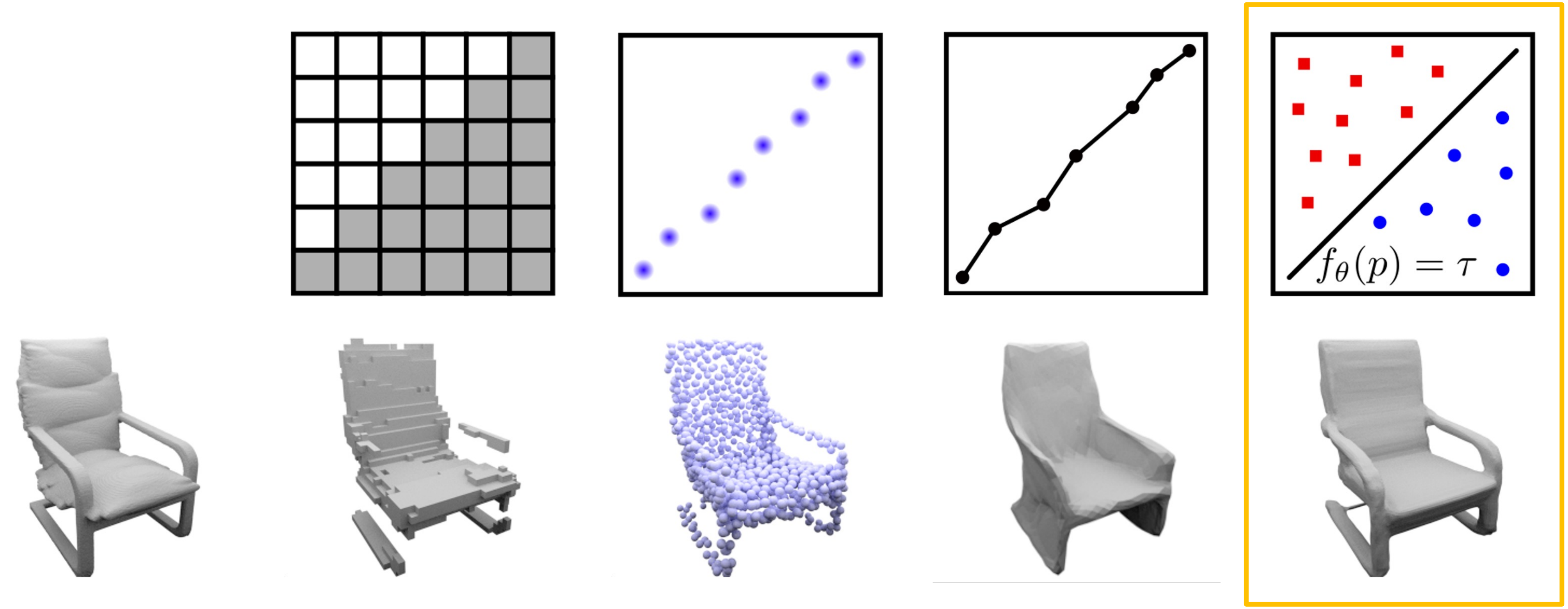


Traditional Explicit Representations

+ Fast inference
− Discrete

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

3D Shape Representation

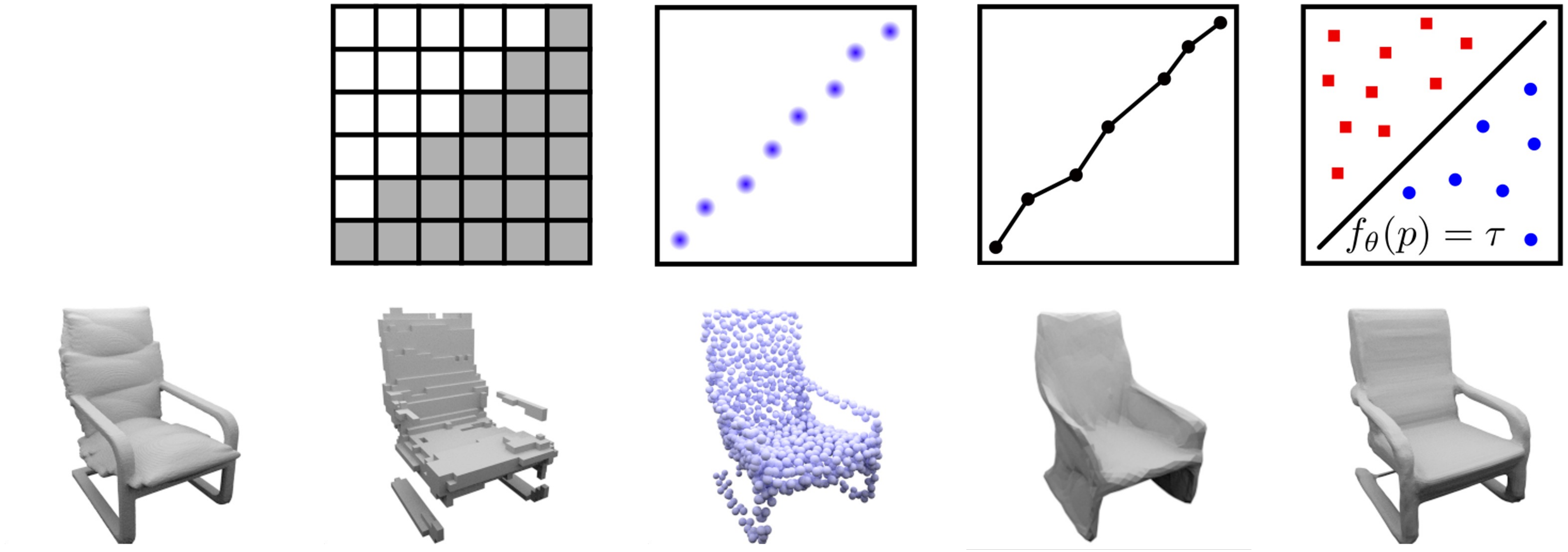


Neural Implicit Representations

- ➕ Continuous, watertight
- ➖ Slow inference
- ➖ Difficult to initialize

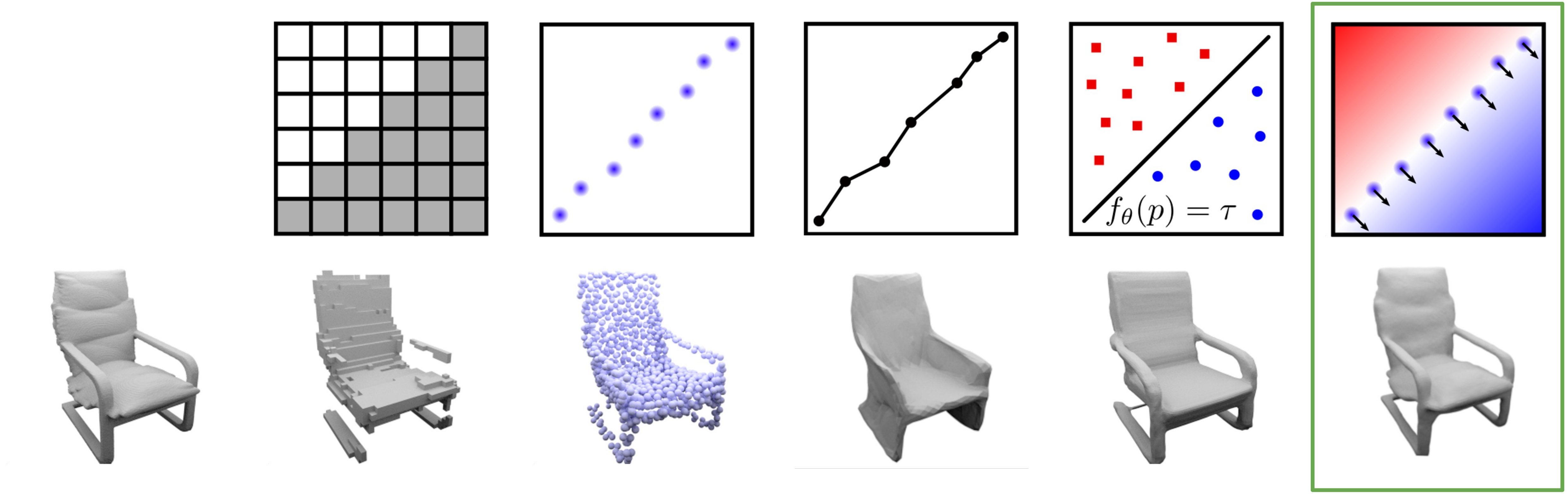
Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

3D Shape Representation



How can we benefit from both worlds?

3D Shape Representation



Shape As Points (SAP) - Hybrid Representation

- ✚ Discrete ⇒ Continuous
- ✚ Fast inference
- ✚ Easy initialization

ETH zürich

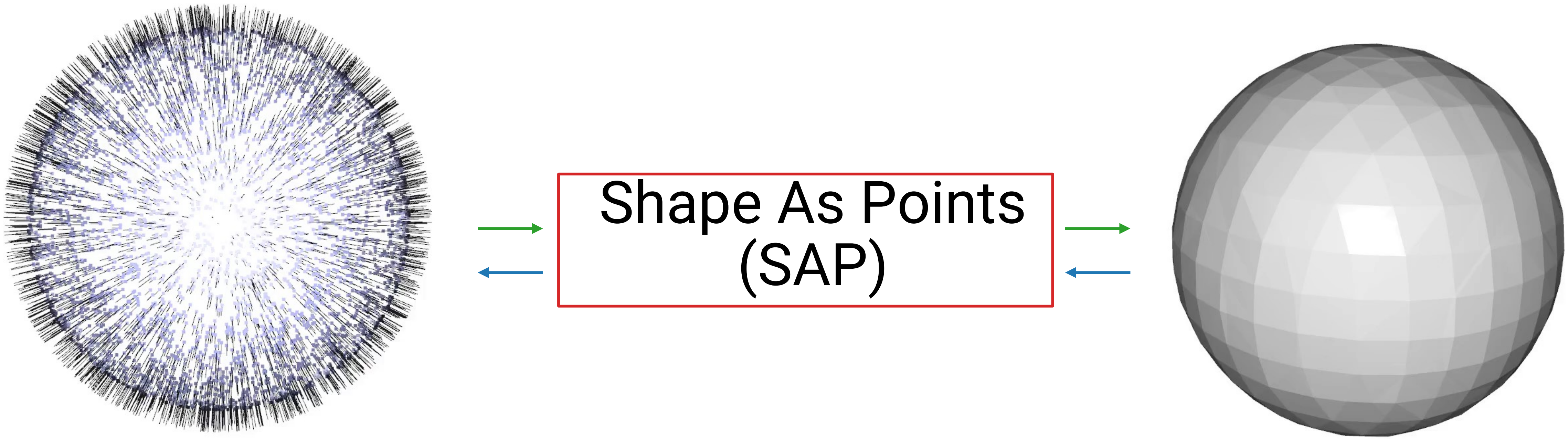
NEURAL INFORMATION PROCESSING SYSTEMS

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Shape As Points

A Differentiable Poisson Solver

Songyou Peng | Chiyu “Max” Jiang | Yiyi Liao | Michael Niemeyer | Marc Pollefeys | Andreas Geiger



Duality between **oriented point clouds** and **3D dense geometry**

Differentiable Poisson Solver

Intuition of Poisson Equation

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

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

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

Our Poisson Solver

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

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

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

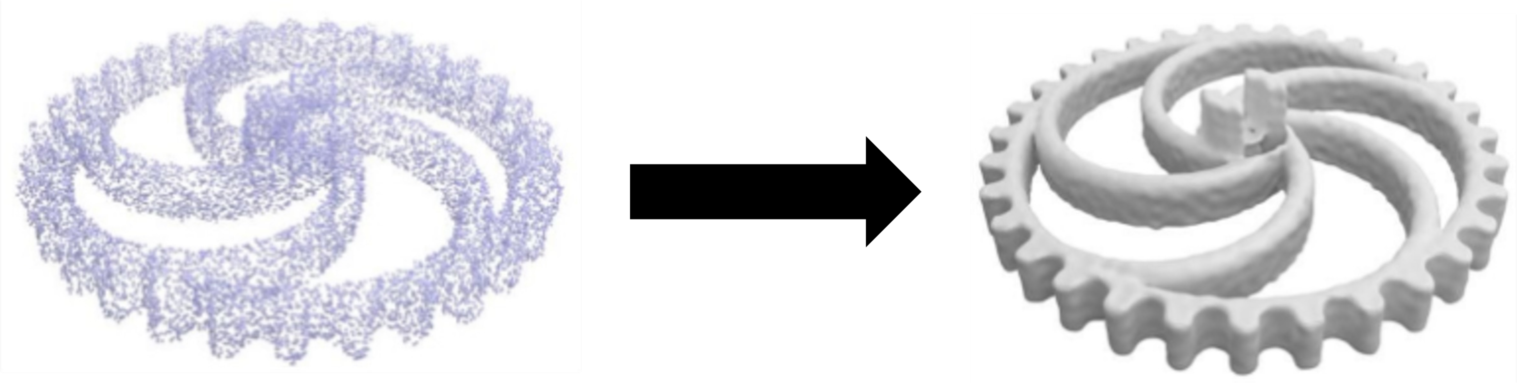
- **Spectral methods** to solve the Poisson equation
  - Derivatives of signals in spectral domain are computed analytically
  - Fast Fourier Transform (FFT) are **highly optimized on GPUs/TPUs**
  - Only **25-line codes**

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

How can we benefit from the differentiablity of DPSR?

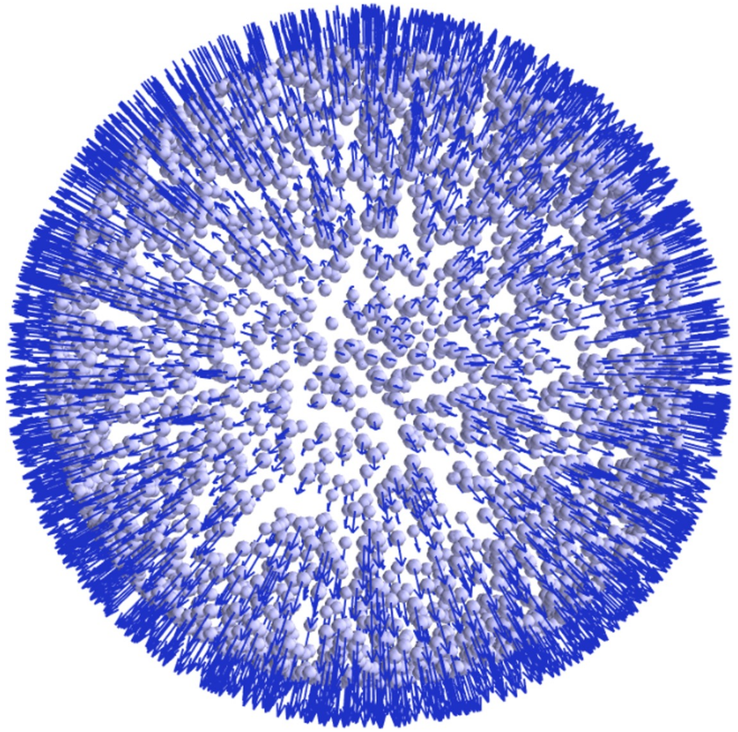
First Application

Optimization-based 3D Surface Reconstruction from unoriented point clouds



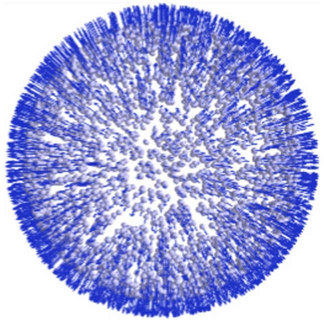
Pipeline - Forward Pass

Input an initial oriented point cloud
(noisy / incomplete observations)

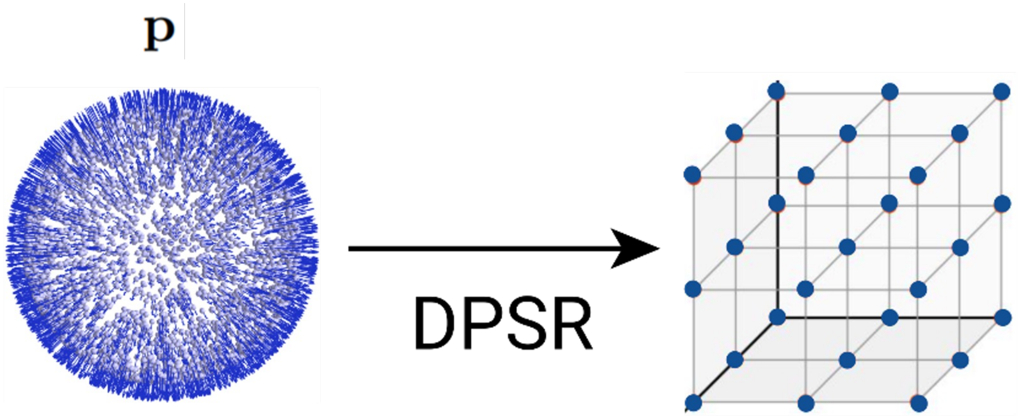


Pipeline - Forward Pass

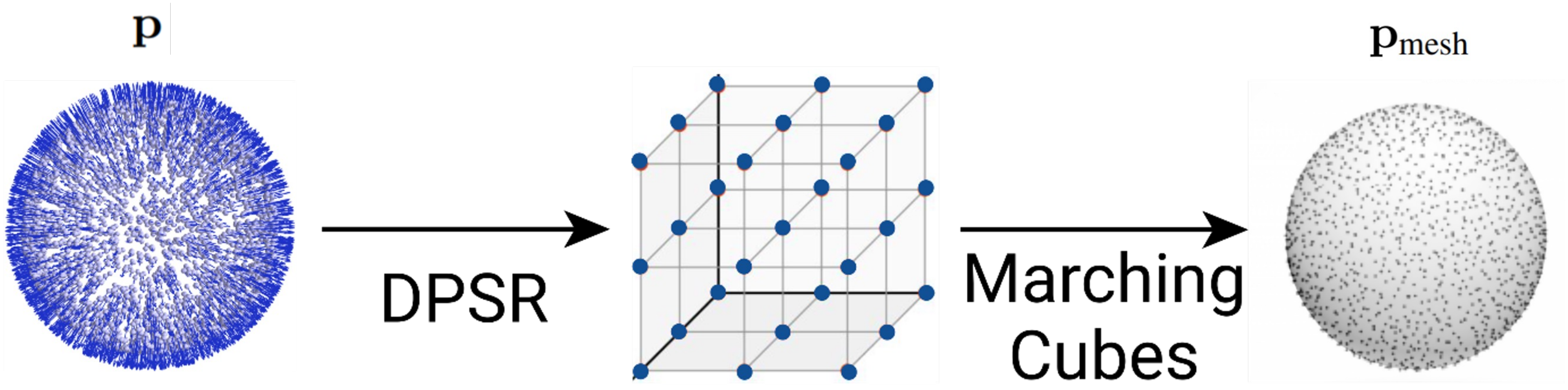
$\mathbf{p}$



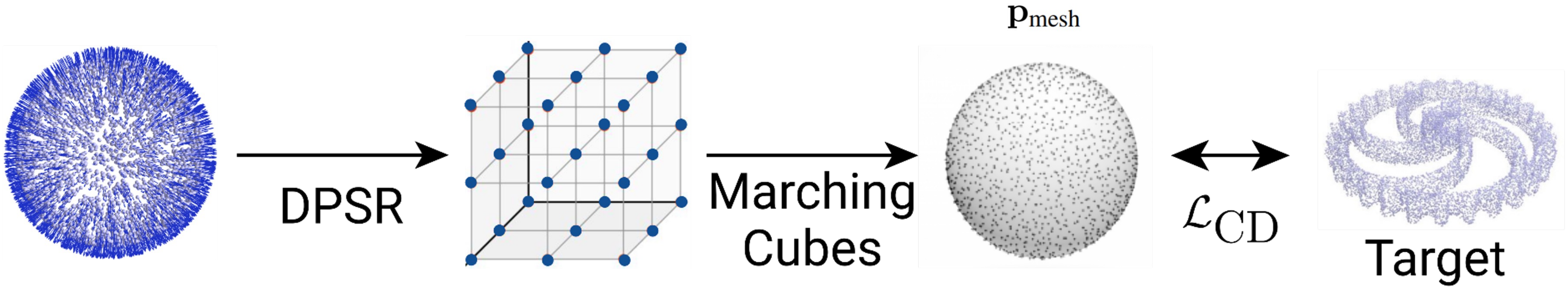
Pipeline - Forward Pass



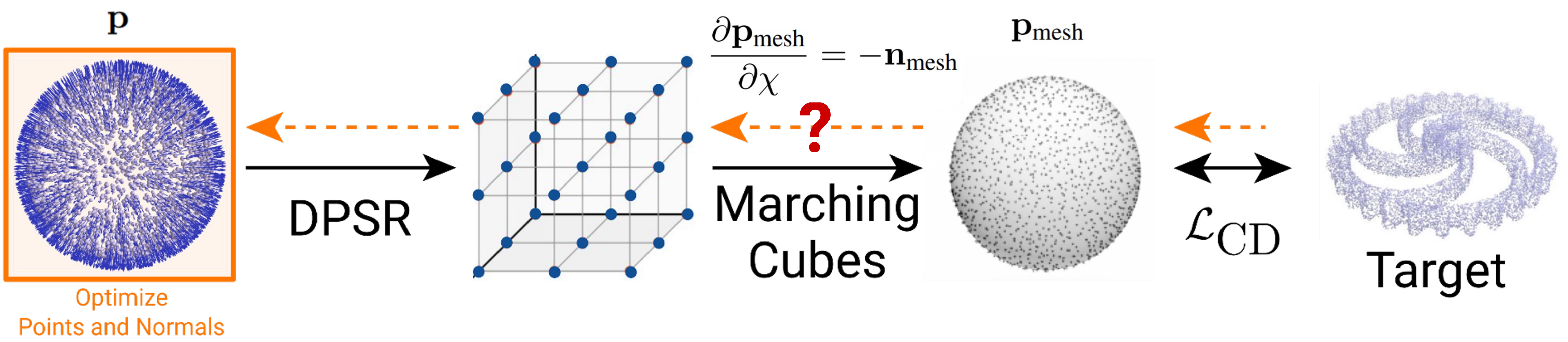
Pipeline - Forward Pass



Pipeline - Forward Pass

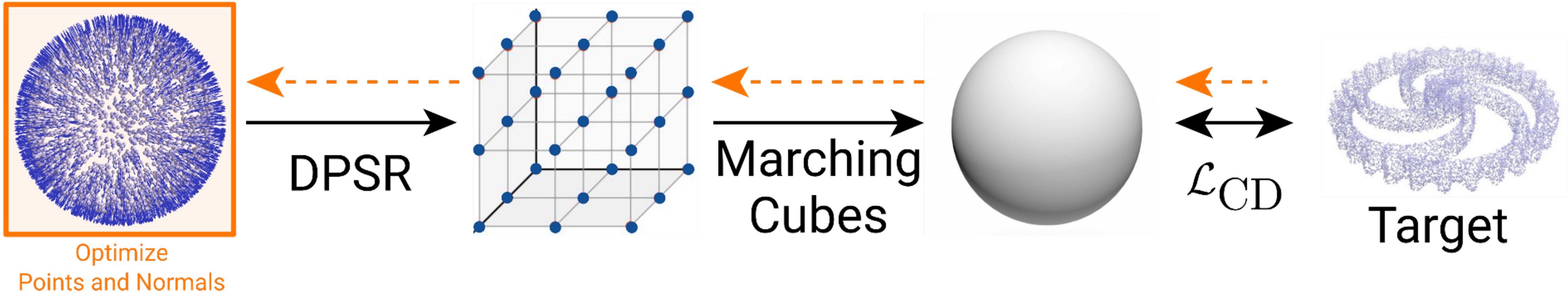


Pipeline - Backward Pass

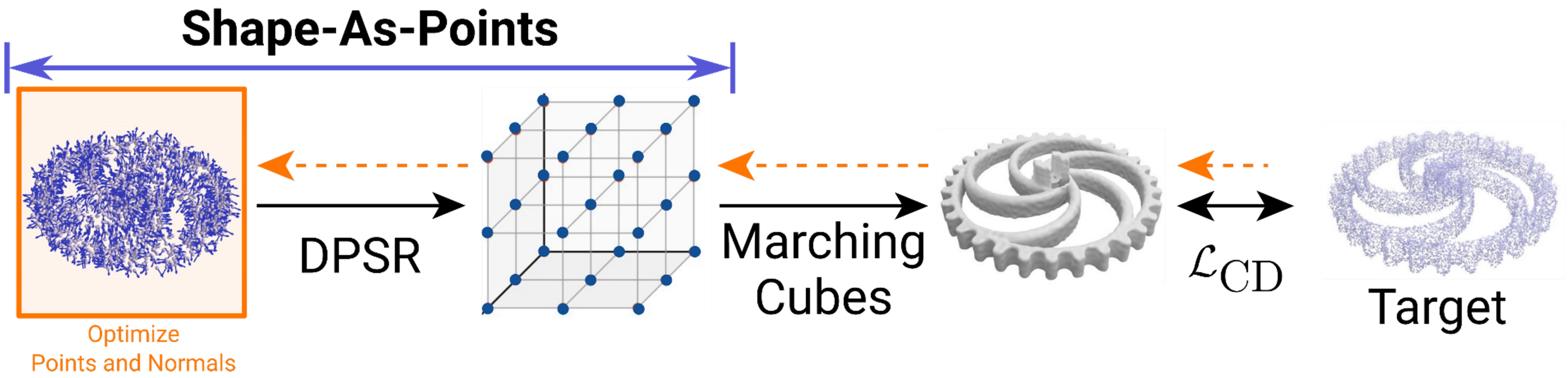


$$\frac{\partial \mathcal{L}_{\text{CD}}}{\partial \mathbf{p}} = \frac{\partial \mathcal{L}_{\text{CD}}}{\partial \mathbf{p}_{\text{mesh}}} \frac{\partial \mathbf{p}_{\text{mesh}}}{\partial \chi} \frac{\partial \chi}{\partial \mathbf{p}}$$

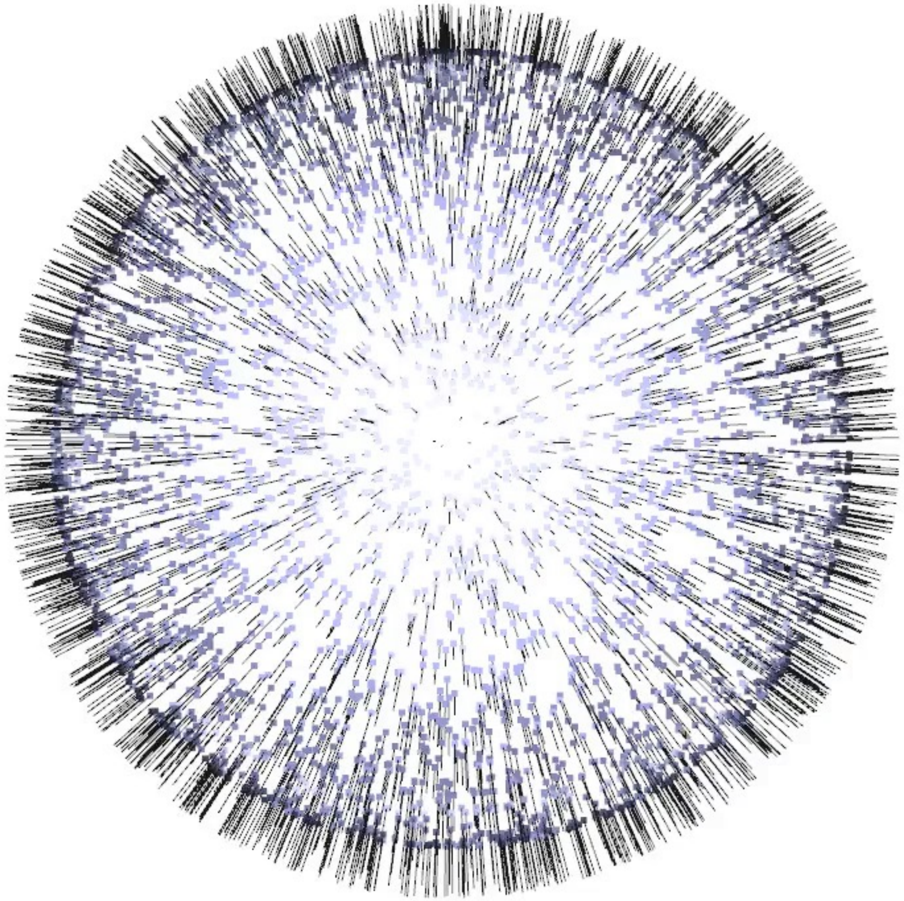
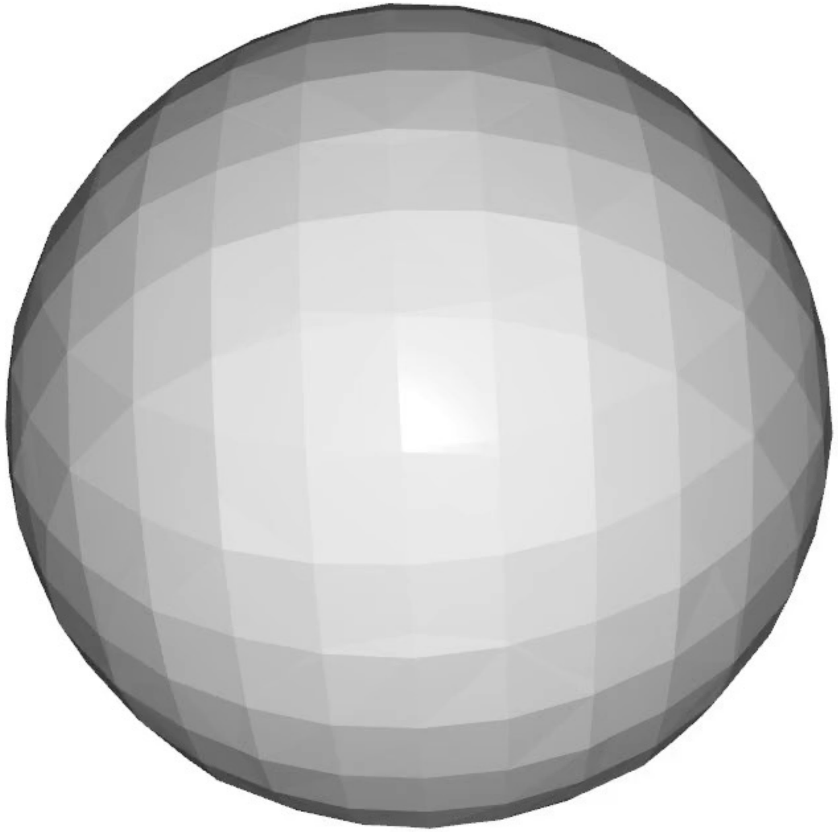
Pipeline


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

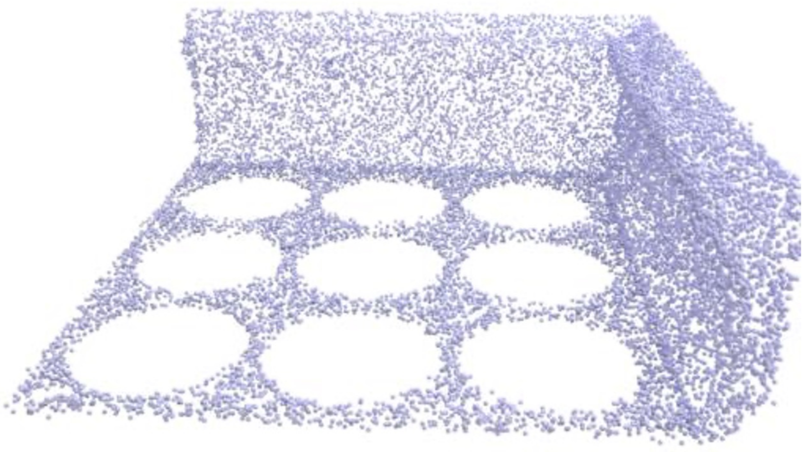
Pipeline



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

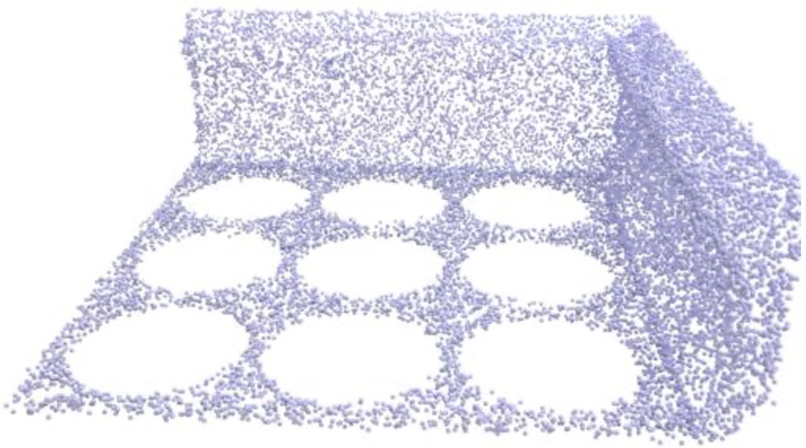
Comparison



Unoriented Point Clouds

GT Mesh

Comparison



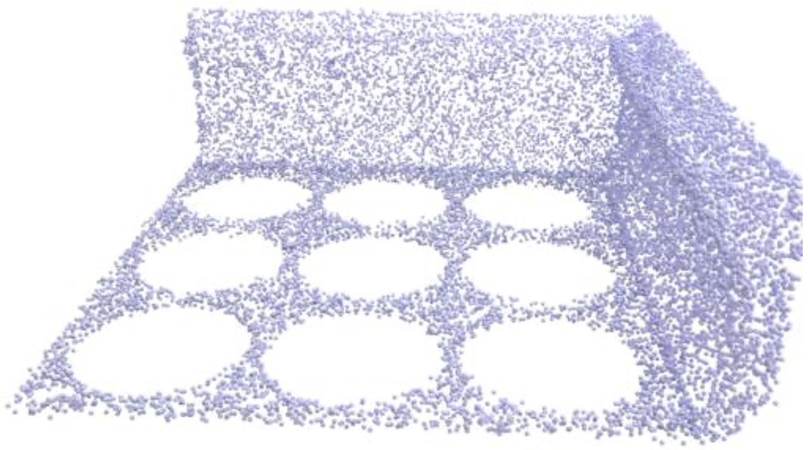
Unoriented Point Clouds

Point2Mesh

Runtime: 62 mins

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

Comparison



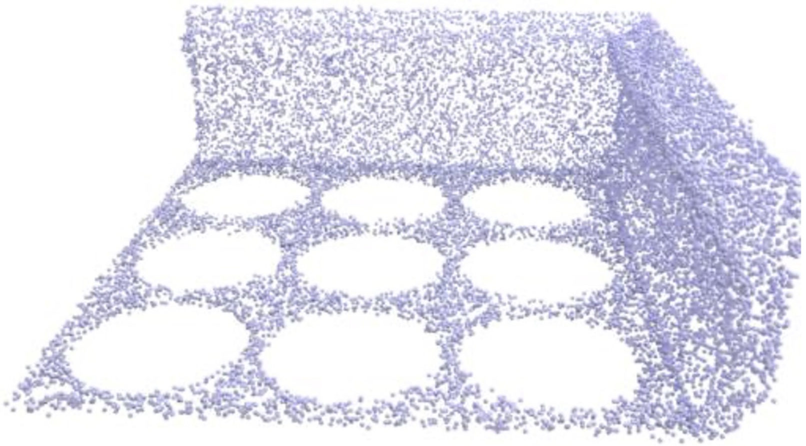
Unoriented Point Clouds

IGR

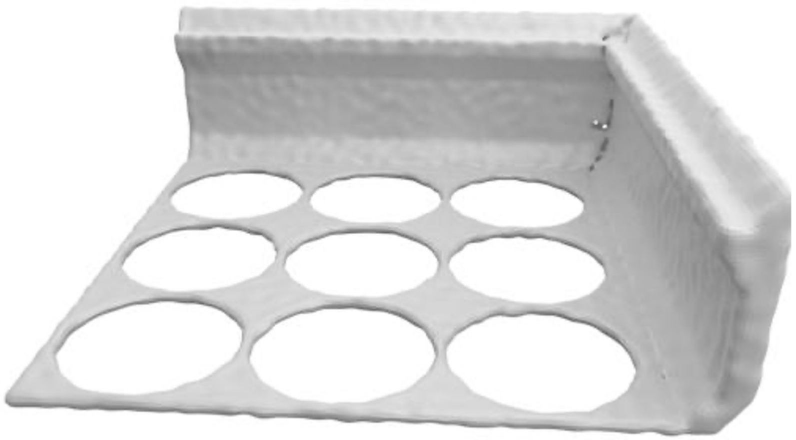
Runtime: 30 mins

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

Comparison



Unoriented Point Clouds



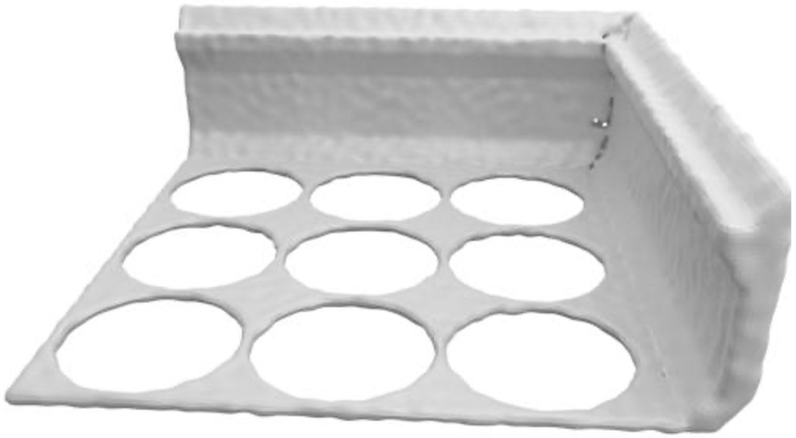
SAP

Runtime: ~6 mins

Comparison

SPSR

Runtime: ~9 sec



SAP

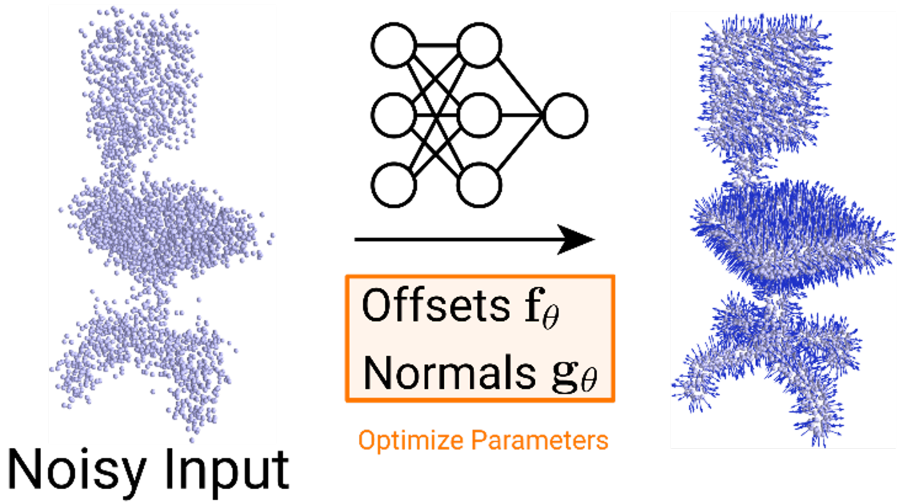
Runtime: ~6 mins

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

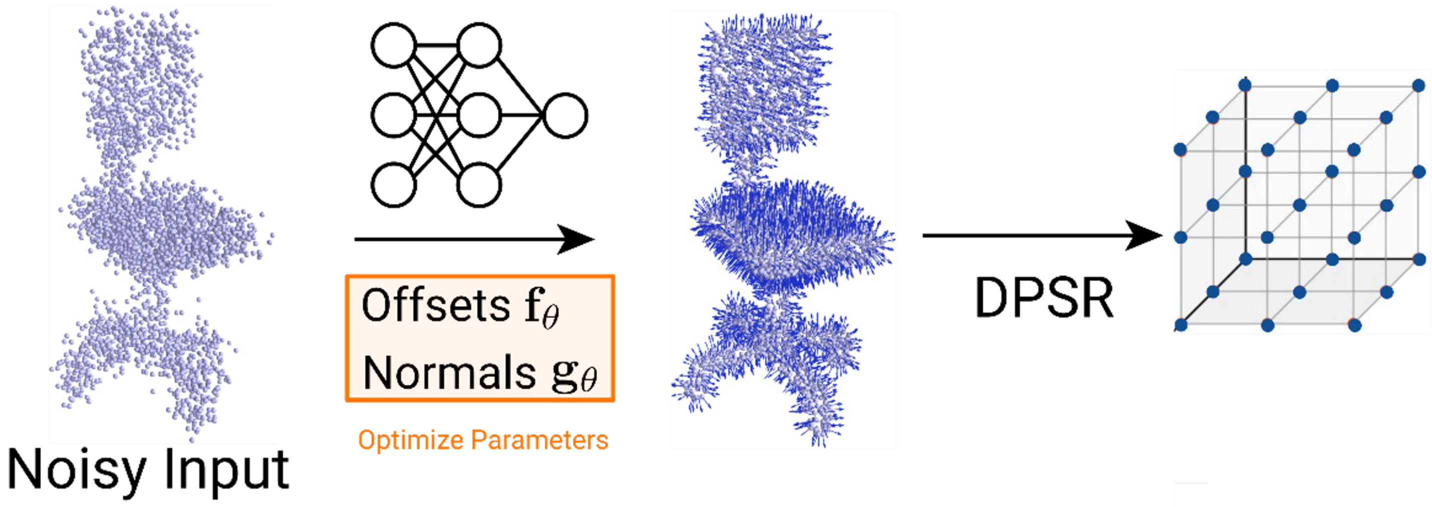
Can we further leverage the differentiability of the Poisson solver for **deep neural networks**?

SAP for Learning-based 3D Reconstruction

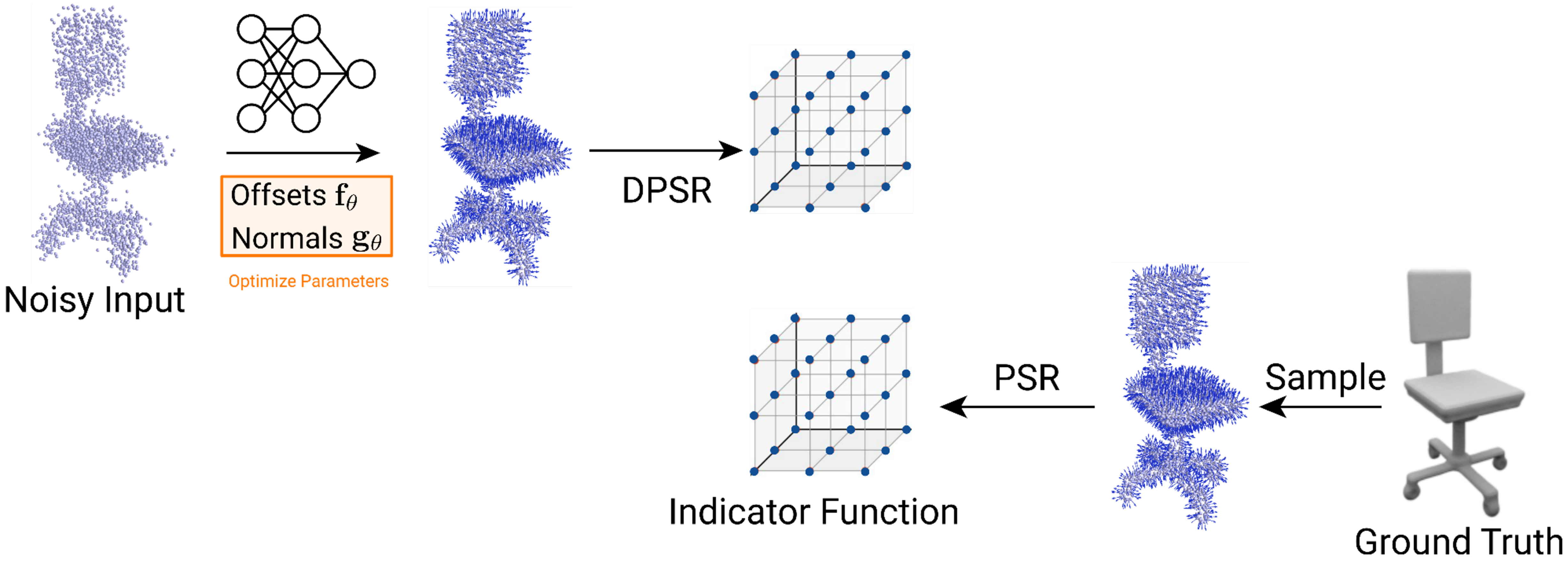
Learning-based Pipeline



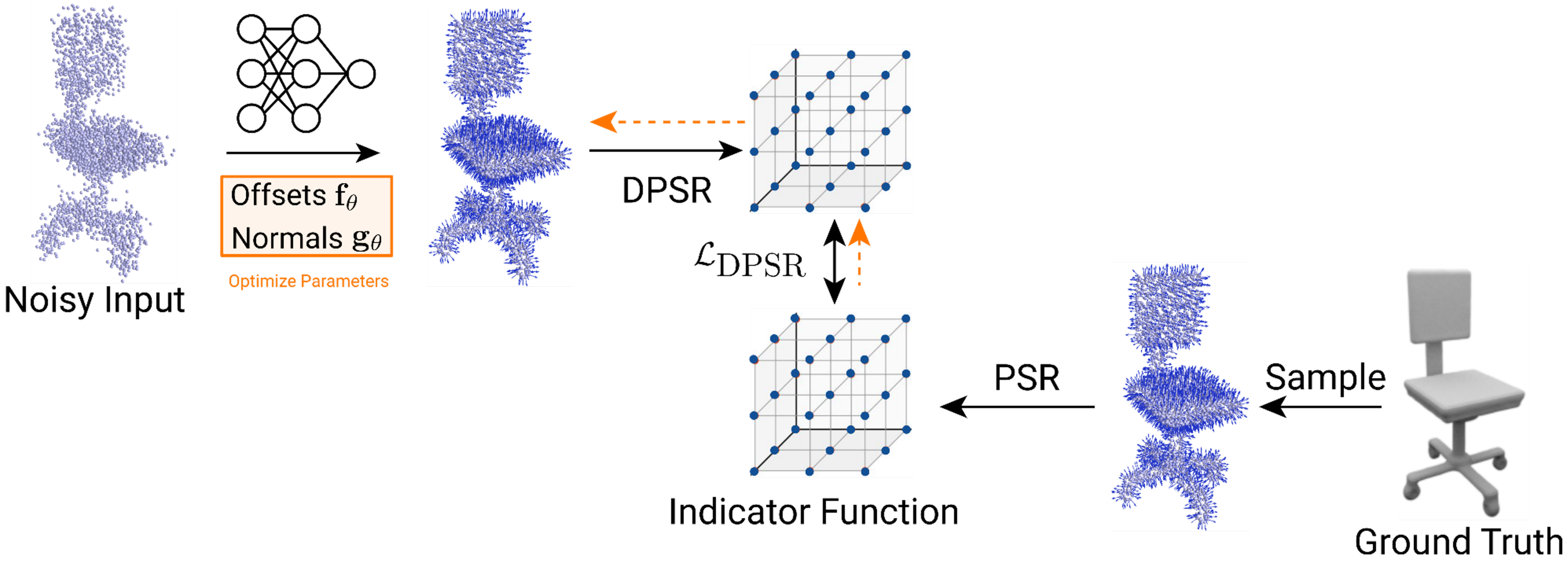
Learning-based Pipeline



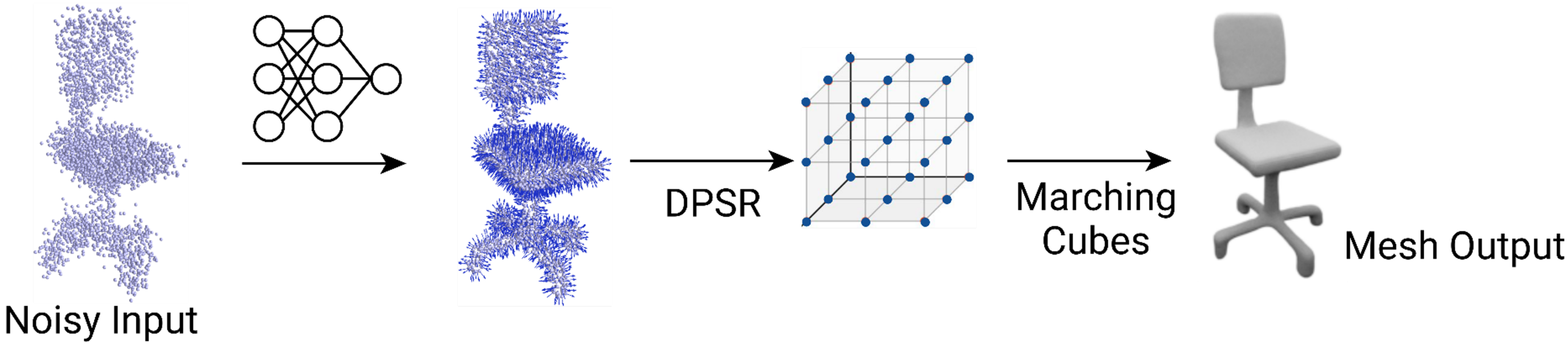
Learning-based Pipeline



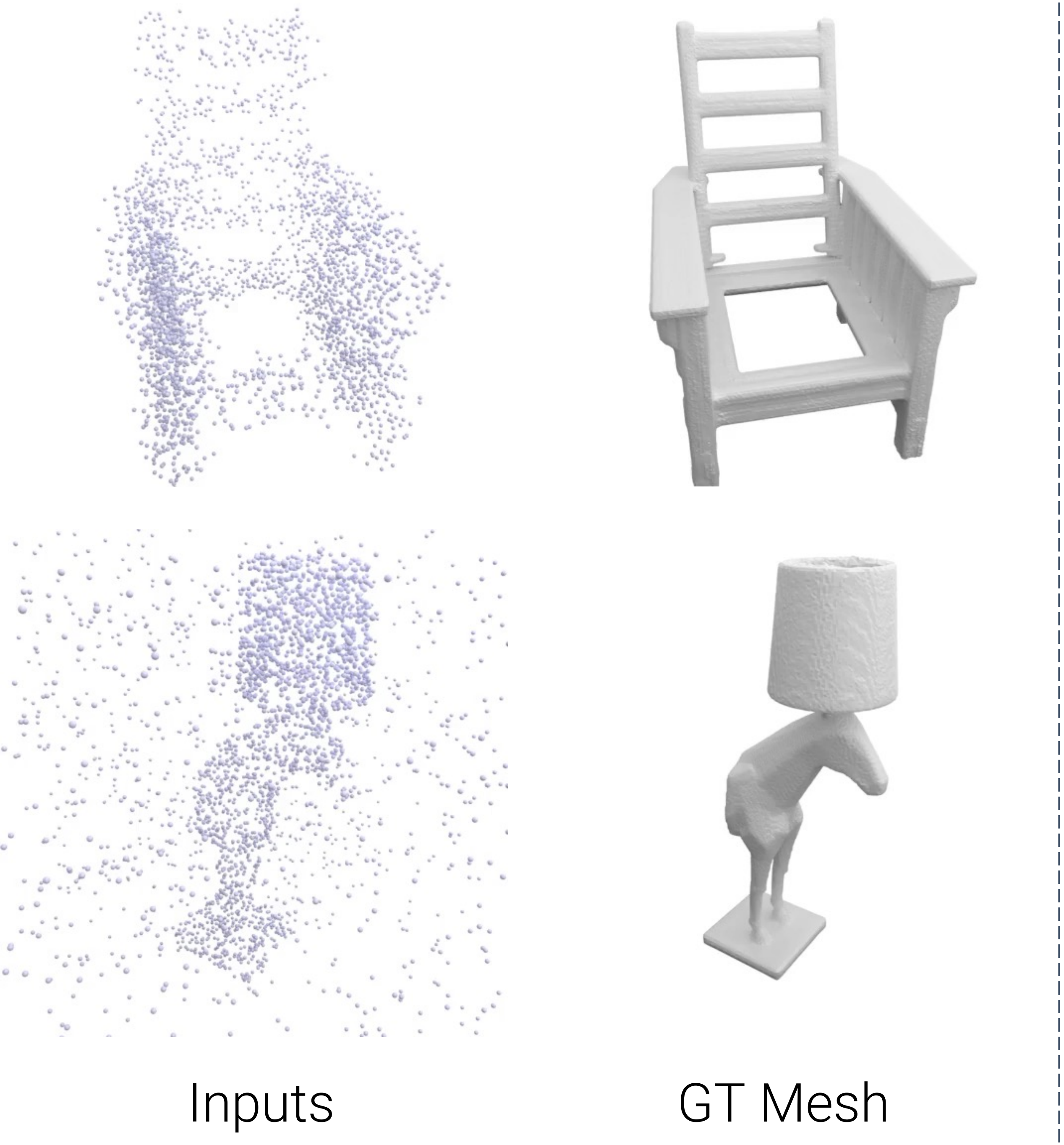
Learning-based Pipeline

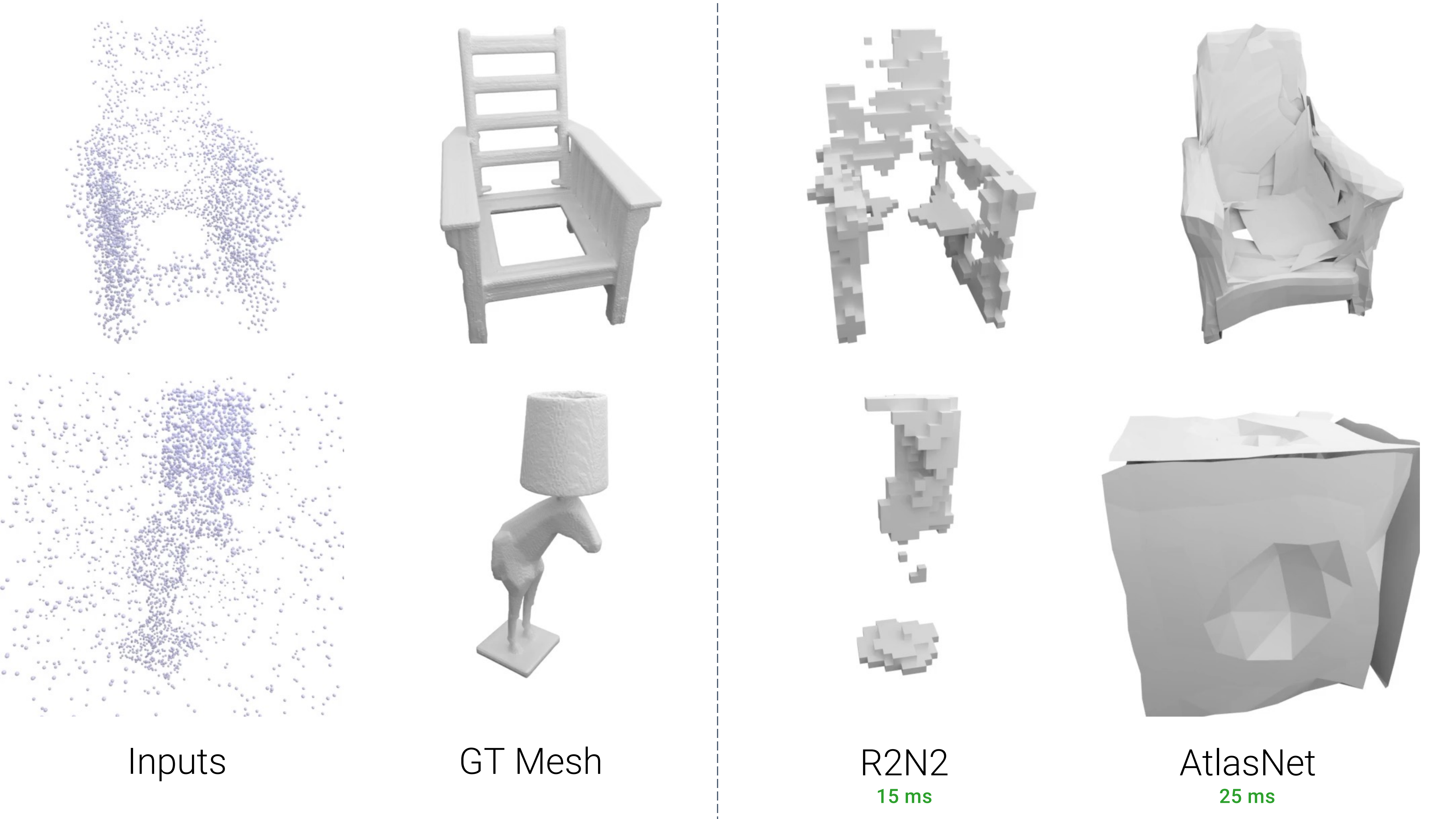


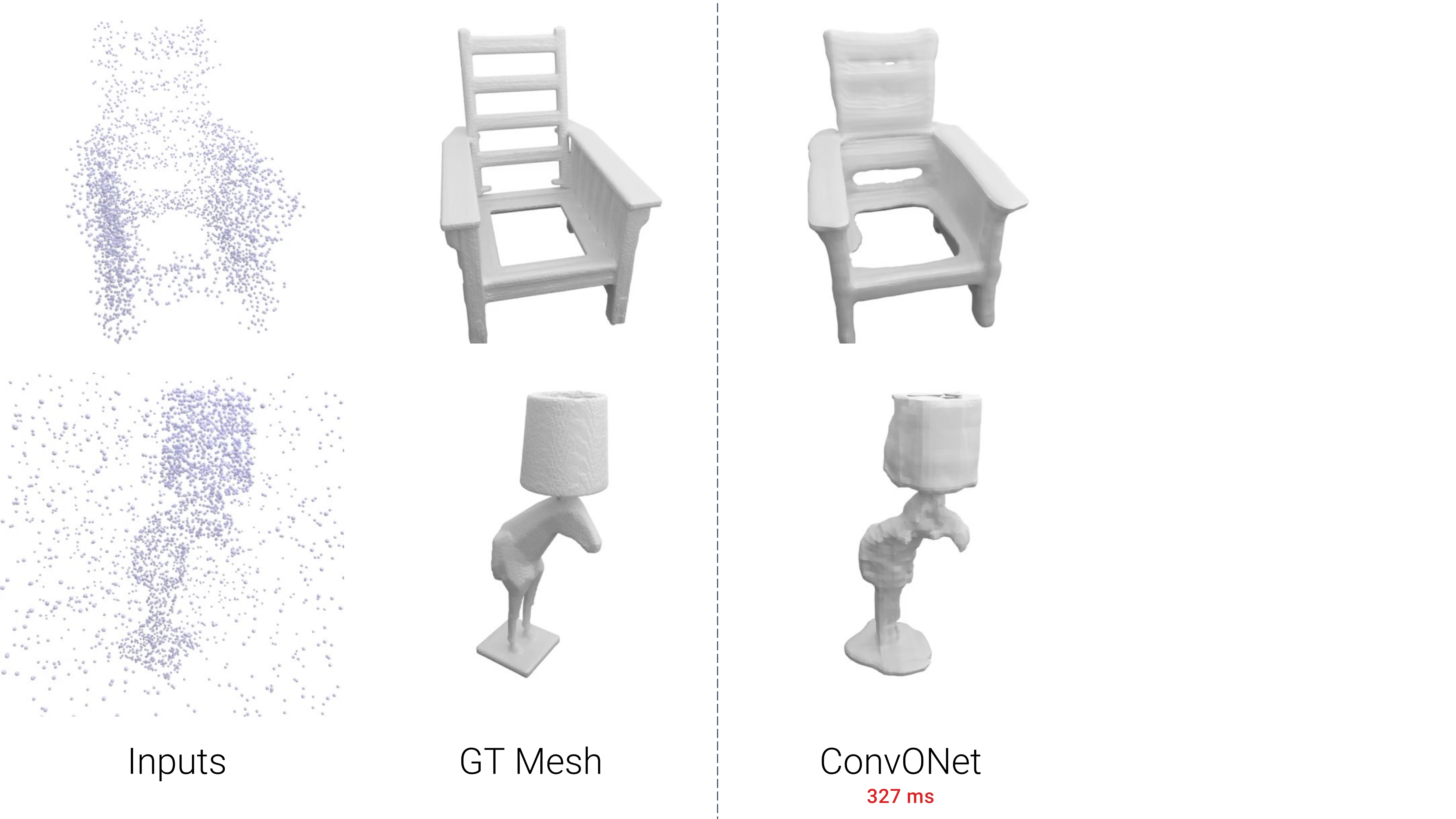
Learning-based Pipeline

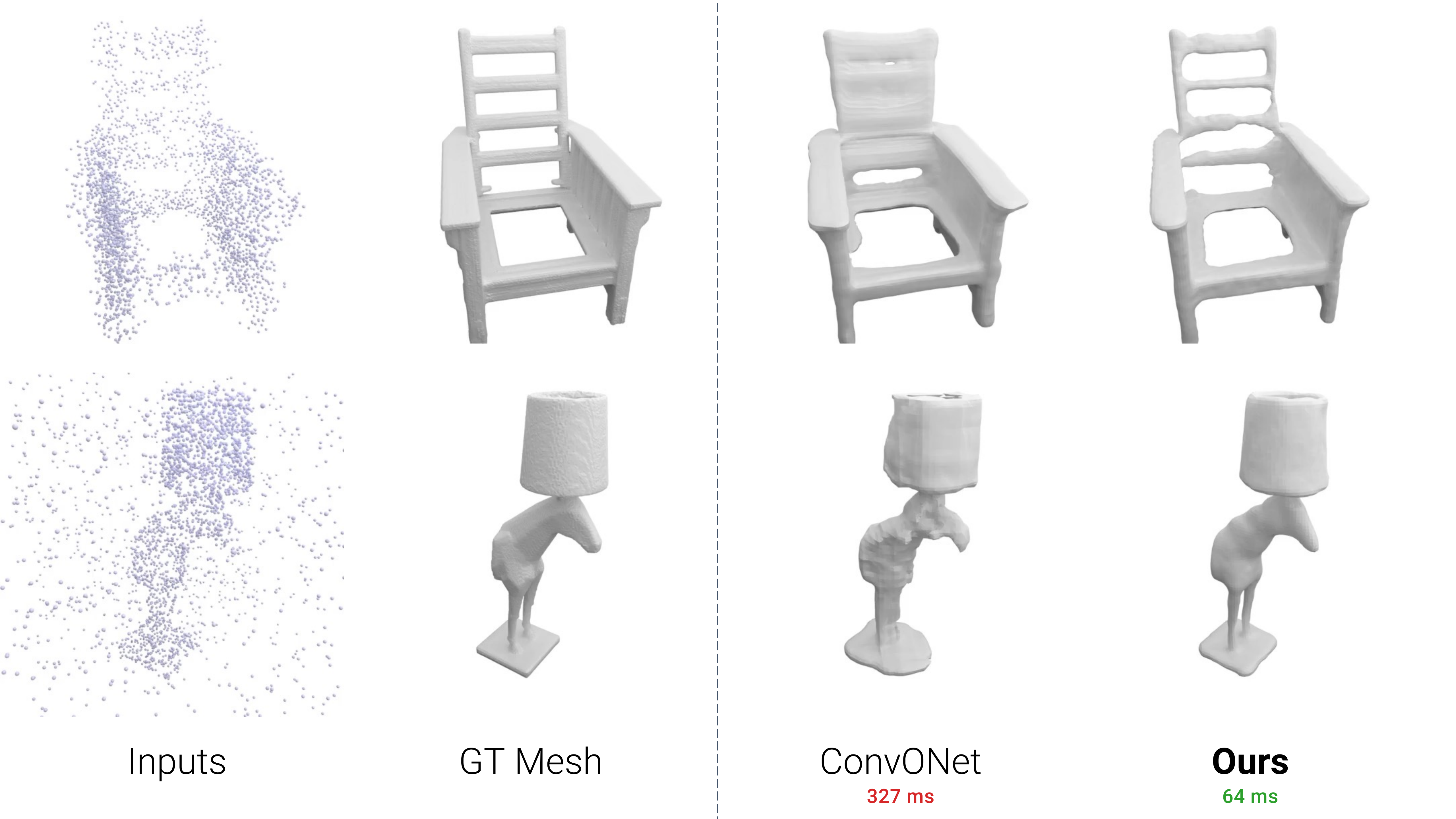


Results


Inputs
GT Mesh


Inputs
GT Mesh
R2N2
15 ms
AtlasNet
25 ms


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**

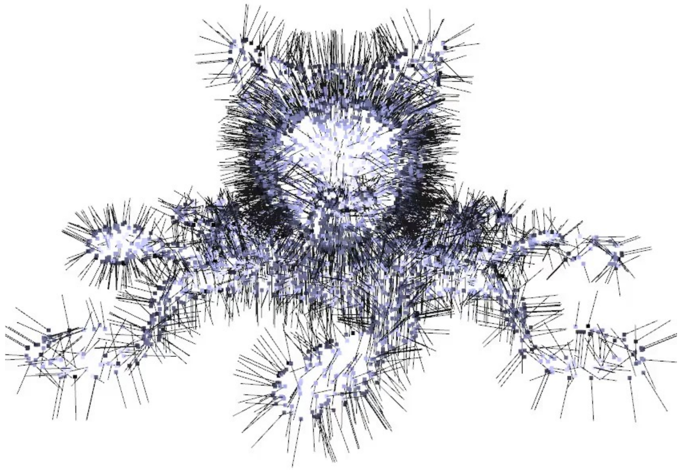
SAP converges much faster!

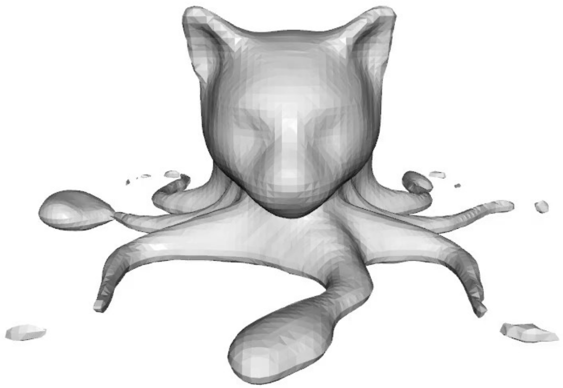
Conclusions

https://pengsongyou.github.io/sap

- SAP is a hybrid representation that is **interpretable, topology agnostic**, and 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...
Neural Radiance Field
(NeRF)
makeameme.org

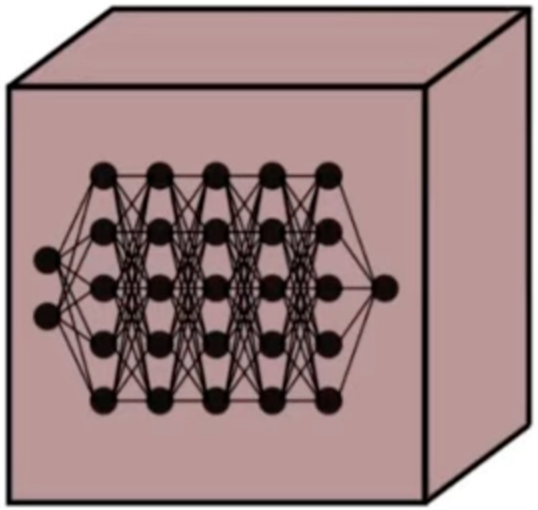
NeRF is awesome!

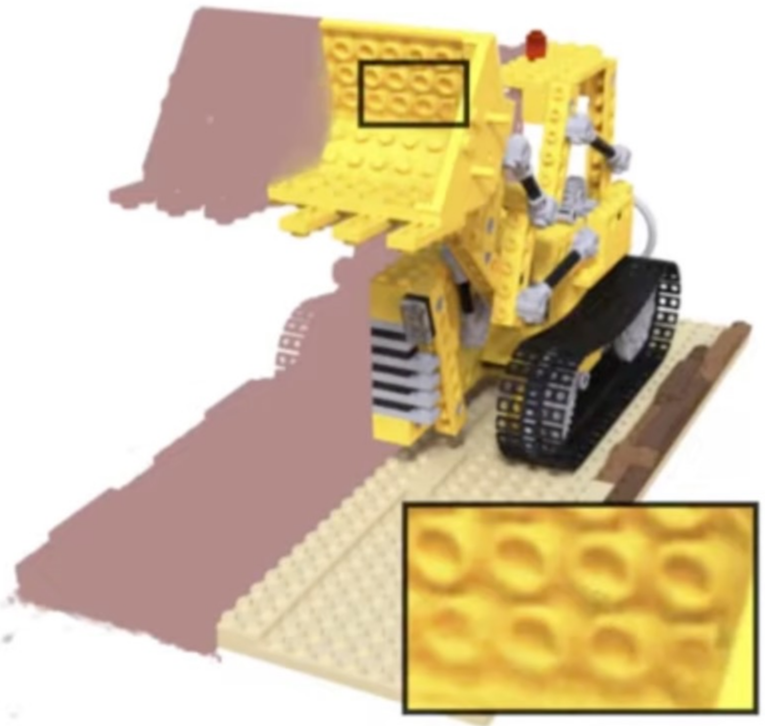
Some existing problems…

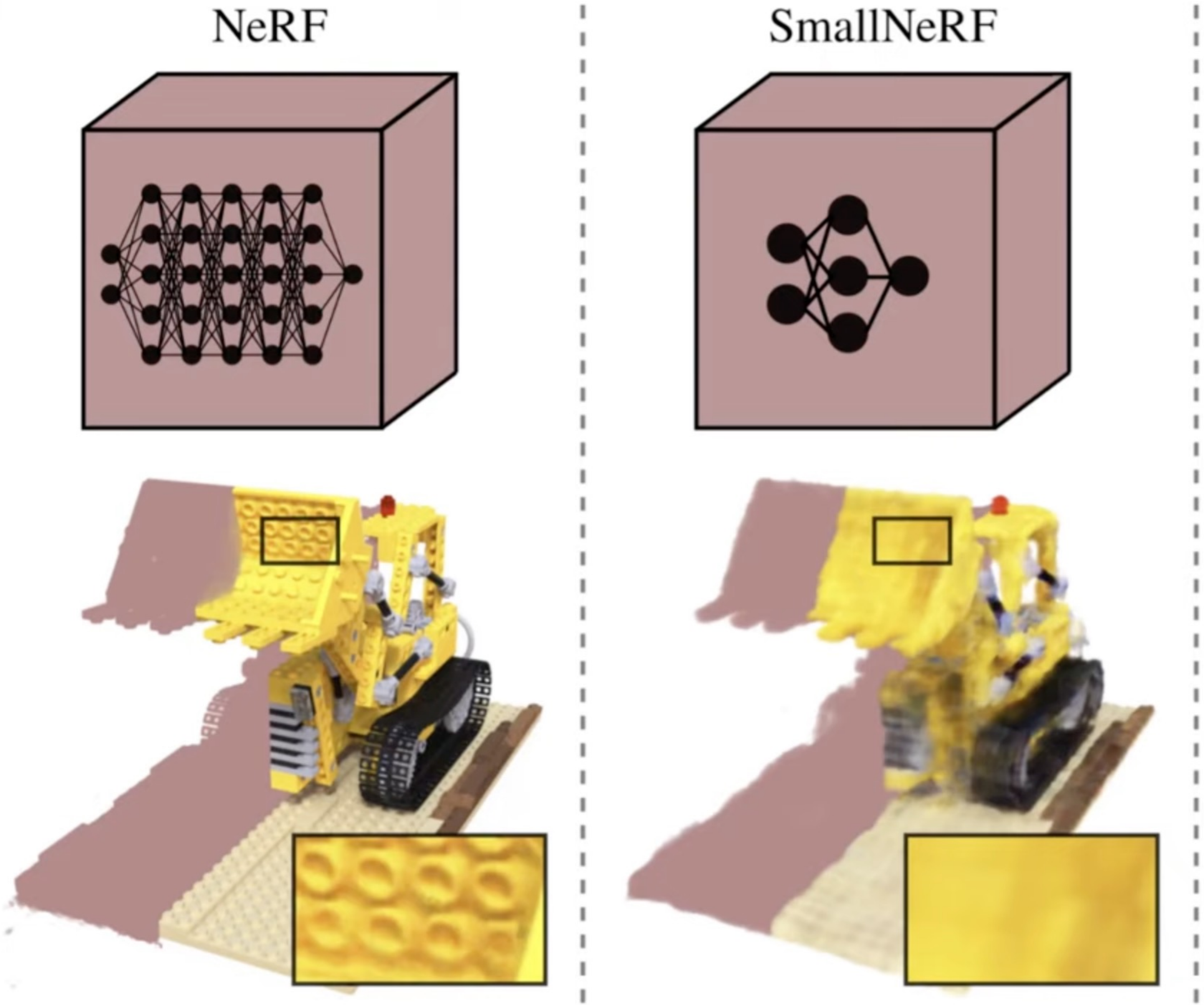
😢 Very slow rendering speed

Mildenhall*, Srinivasan*, Tancik* et al: NeRF : Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV 2020

NeRF

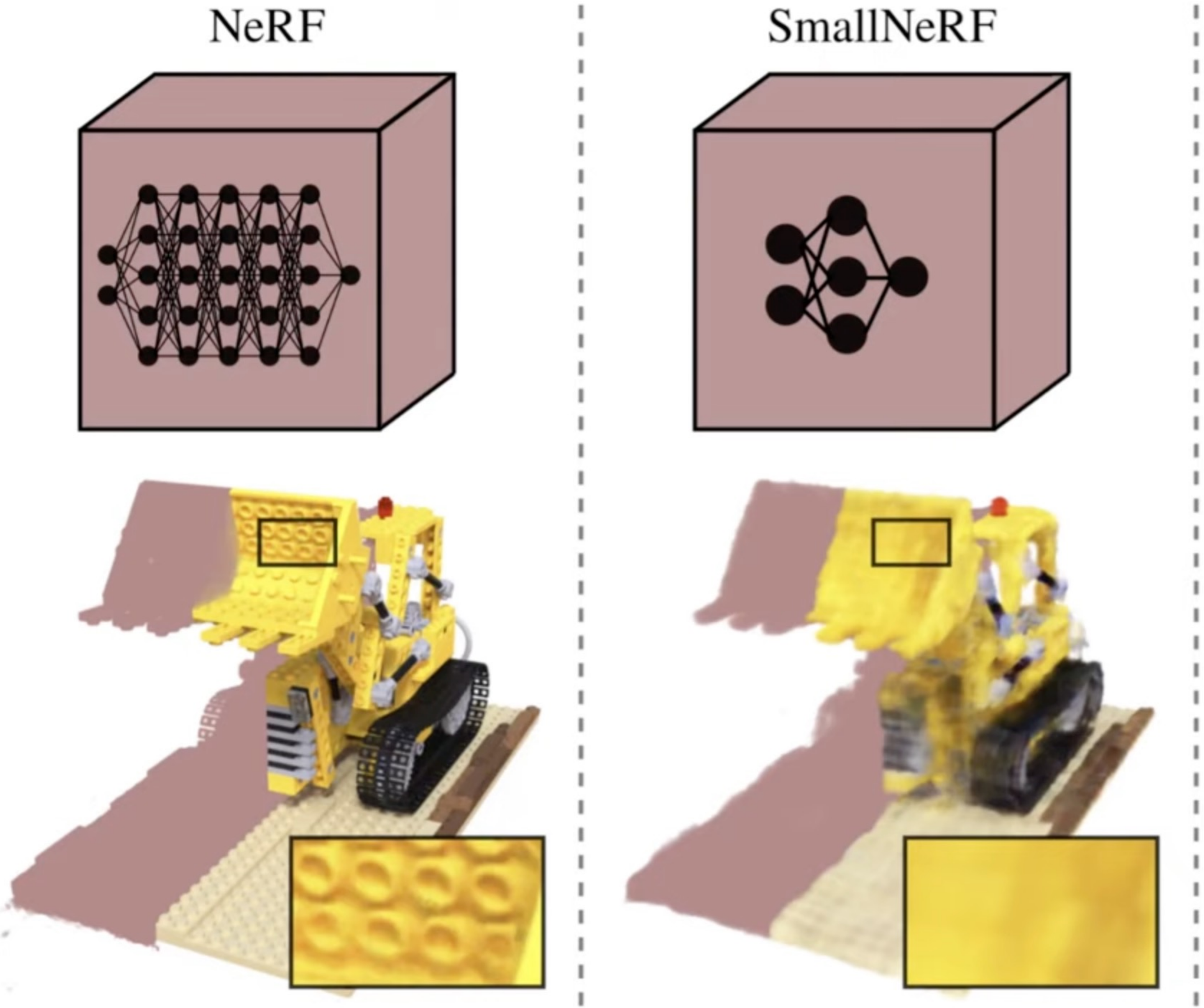


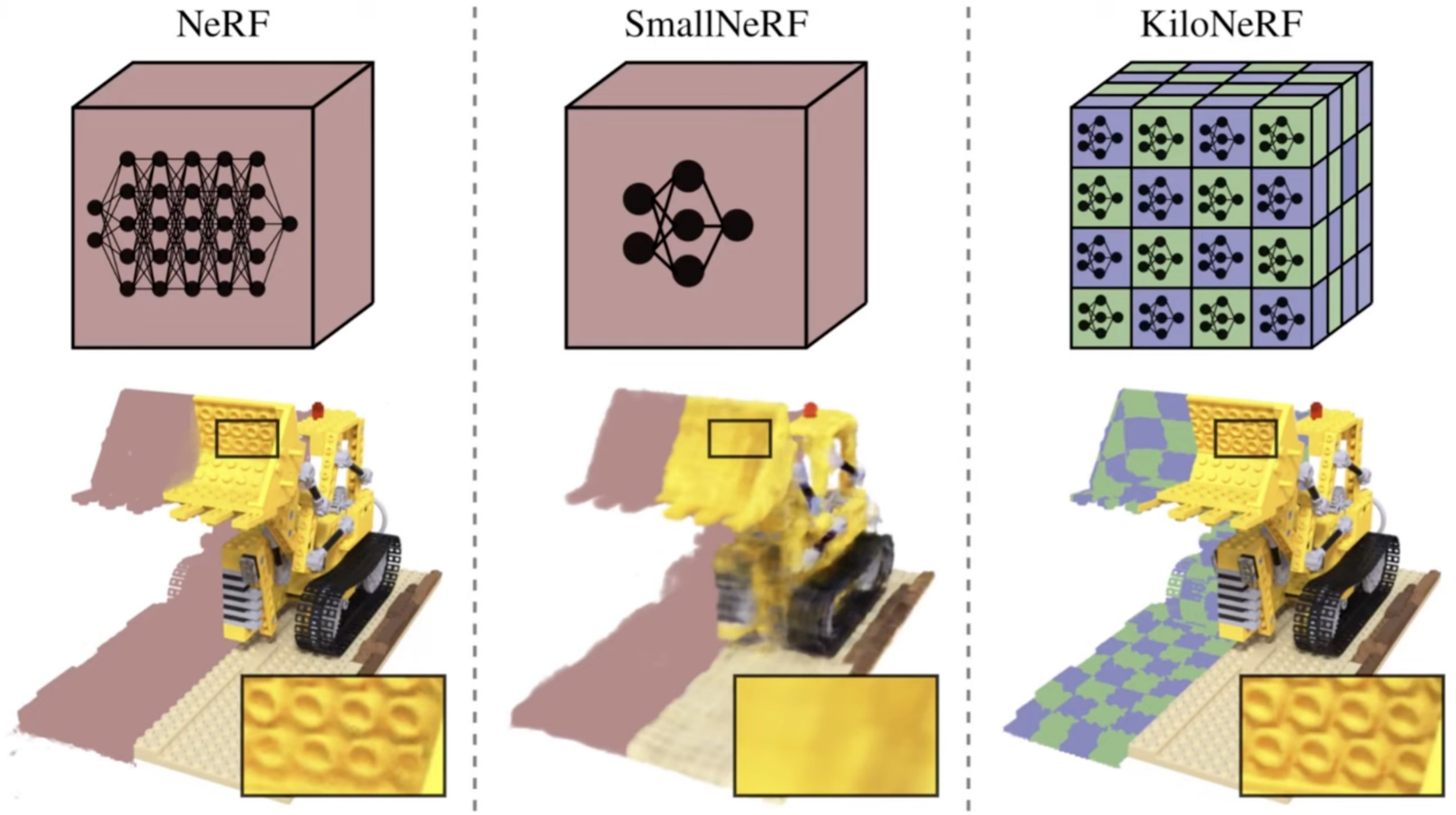


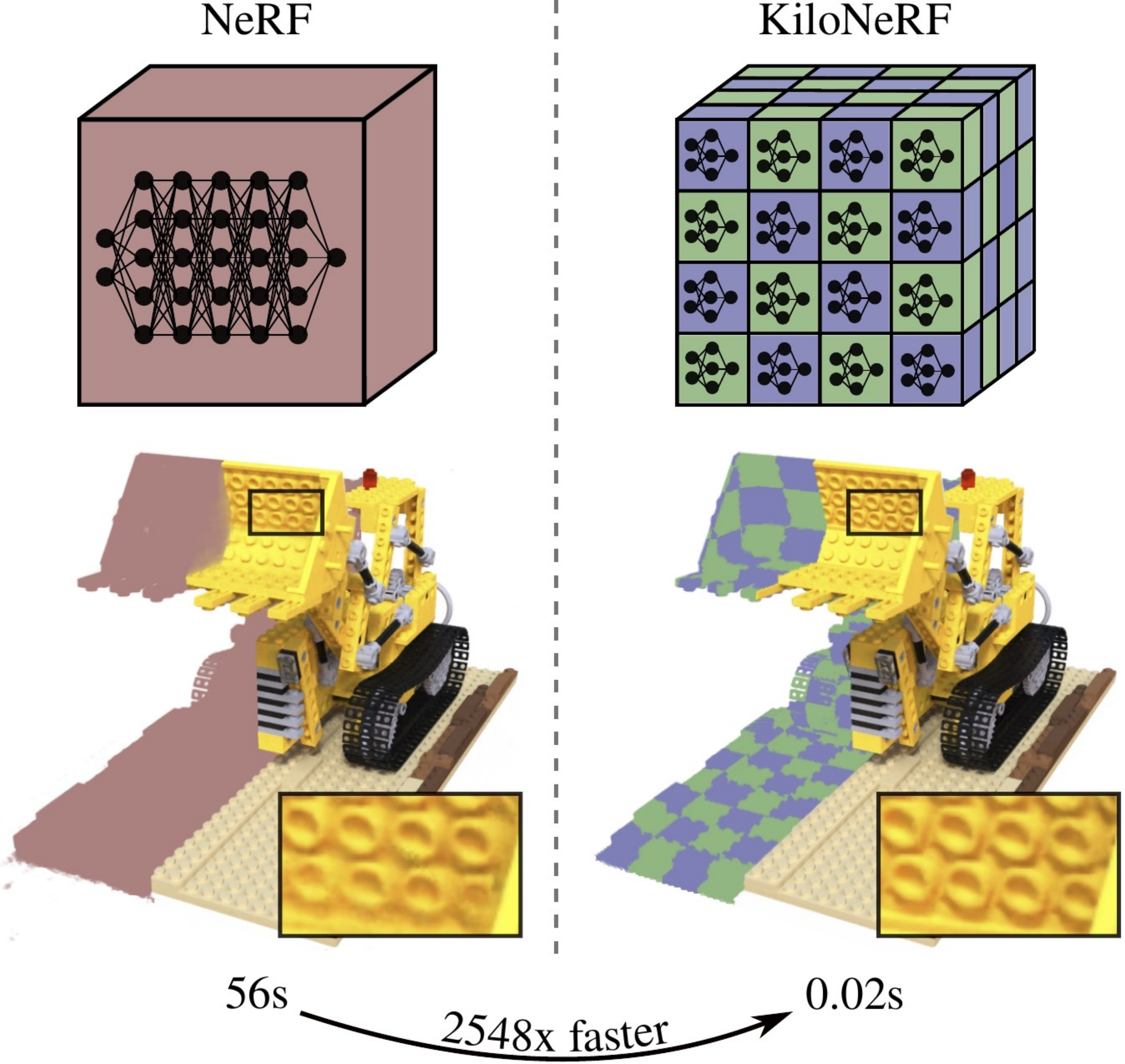

NeRF
SmallNeRF

How to speed up NeRF rendering?

Combine Explicit with Implicit Representations!


NeRF
SmallNeRF

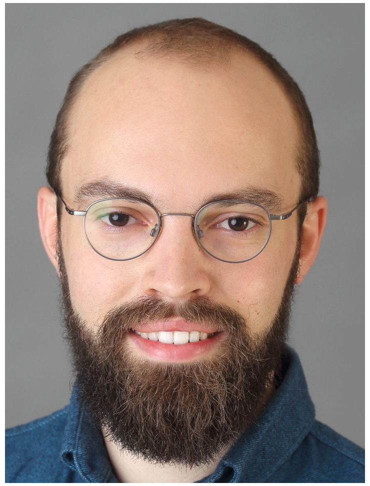

NeRF
SmallNeRF
KiloNeRF

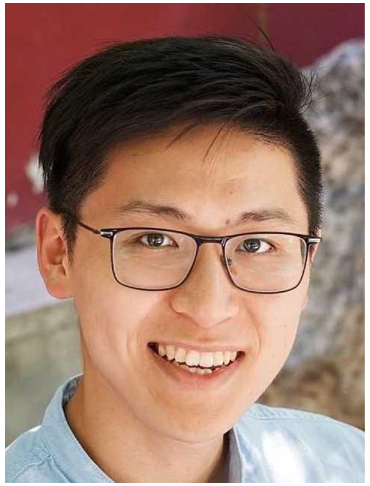

NeRF
KiloNeRF
56s
2548x faster
0.02s

2021 ICCV OCTOBER 11-17 VIRTUAL

KiloNeRF

Speeding up NeRF with Thousands of Tiny MLPs


Christian Reiser


Songyou Peng


Yiyi Liao


Andreas Geiger

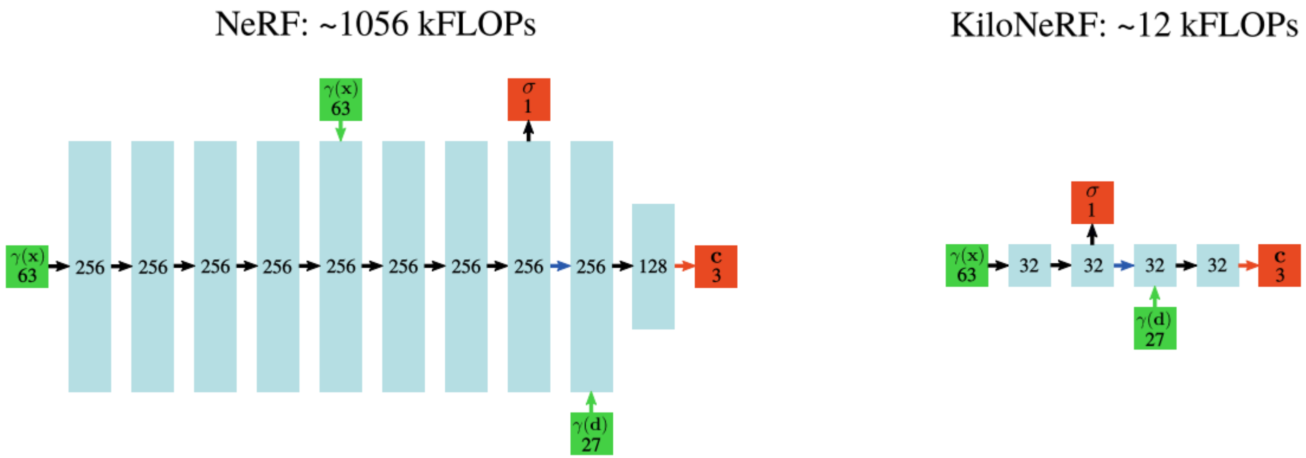
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

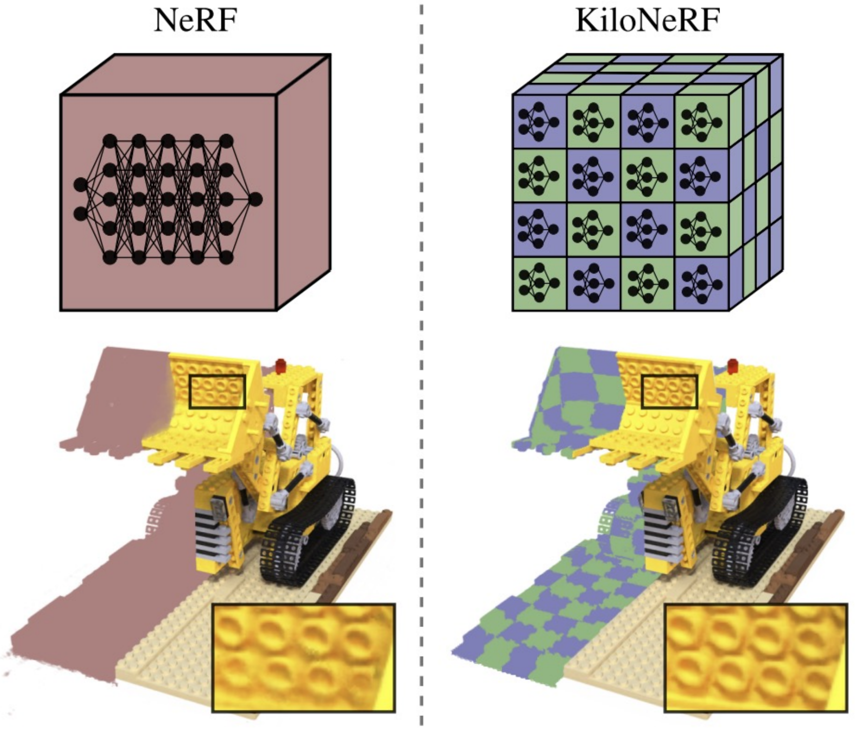
MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

ETH zürich

Key Idea

- Partition a scene into a 16^3 uniform grid
- Each grid cell is represented by a tiny MLP





87x reduction in FLOPs!

* FLOP: floating points operations

KiloNeRF

Training:

1. Distill a trained NeRF model into our KiloNeRF model
   - Randomly sampled points, their predicted alpha & color values should match!
2. Finetune the thousand MLPs on training images

(a) Without Distillation (b) With Distillation

KiloNeRF

Training:

1. Distill a trained NeRF model into our KiloNeRF model
   - Randomly sampled points, their predicted alpha & color values should match!
2. Finetune the thousand MLPs on training images

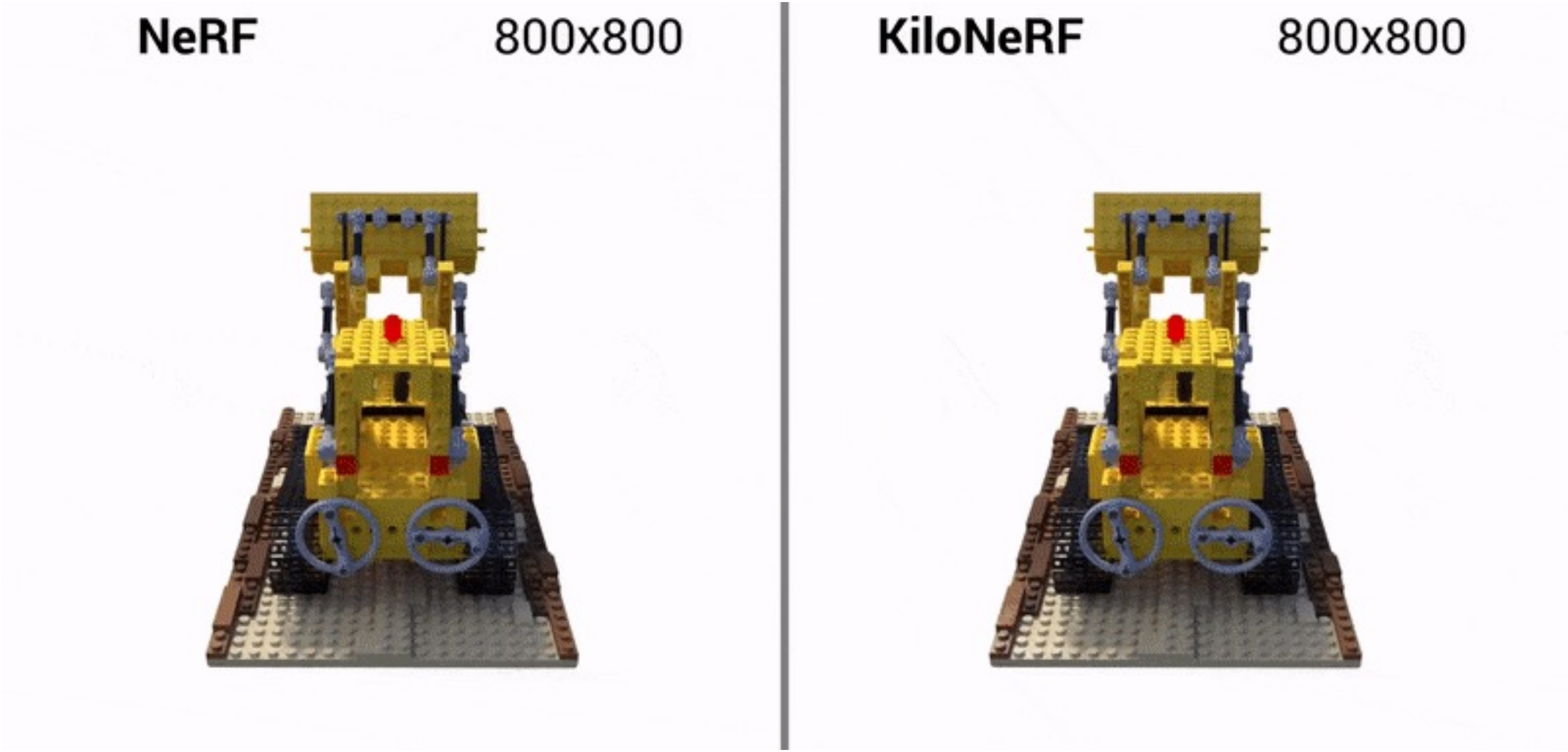
Inference:

1. Empty Space Skipping (ESS) with a pre-computed 256^3 occupancy grid
2. Early Ray Termination (ERT): when transmittance < ε, stop!
3. Evaluate tiny MLPs in parallel

Method	Render time ↓	Speedup ↑
NeRF	56185 ms	–
NeRF + ESS + ERT	788 ms	71
KiloNeRF	**22** ms	**2554**

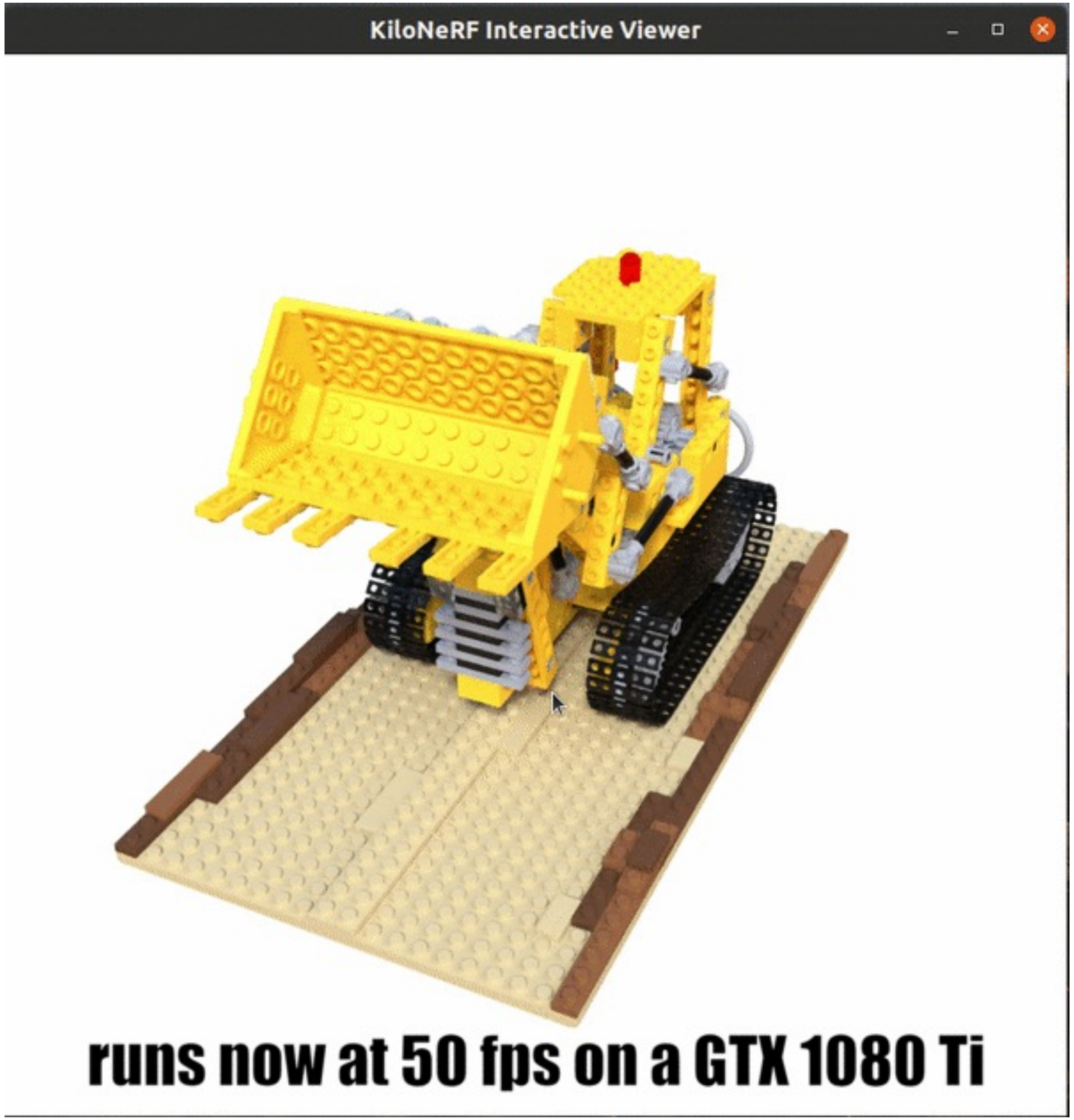
* Tested with NVIDIA GTX 1080 Ti

Results



56 s

0.02 s (50 fps)



https://github.com/creiser/kilonerf

Comparison to Concurrent Works

Type	Neural	Tabulation-based		
Method	**KiloNeRF**	PlenOctree	SNeRG	FastNeRF
GPU Memory	**< 100 MB**	1930 MB	3442 MB	7830 MB

⇒ KiloNeRF has a larger potential for large-scale NVS!

Yu et al.: PlenOctrees For Real-time Rendering of Neural Radiance Fields. ICCV 2021
Hedman et al.: Baking Neural Radiance Fields for Real-Time View Synthesis. ICCV 2021
Garbin et al.: FastNeRF: High-Fidelity Neural Rendering at 200FPS. ICCV 2021

Follow-up Works of KiloNeRF

BlockNeRF applied our idea for city-level NVS ☺

Tancik et al.: Block-NeRF: Scalable Large Scene Neural View Synthesis. CVPR 2022

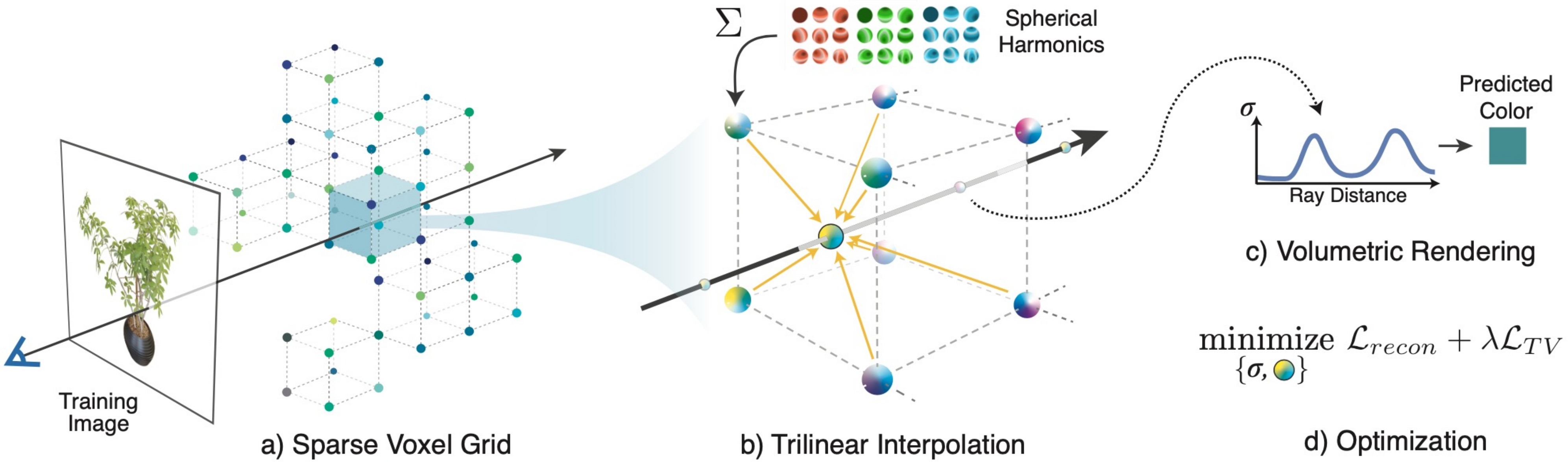
Take-home Message

- Speed up NeRF significantly (~2000x) without loss of quality
- A memory more friendly representation!

Limitations

- Only work on bounded scenes
- **Expensive training time**

Plenoxels



- Directly optimize a view-dependent sparse voxel model
- Train a scene in 11 mins

Fridovich-Keil*, Yu* et al.: Plenoxels: Radiance Fields without Neural Networks. CVPR 2022

Direct Voxel Grid Optimization (DVGO)

- Dense voxel grid for density (geometry), a feature grid with a shallow MLP for appearance
- Train a scene in 15 mins

Sun et al.: Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. CVPR 2022

Instant-NGP

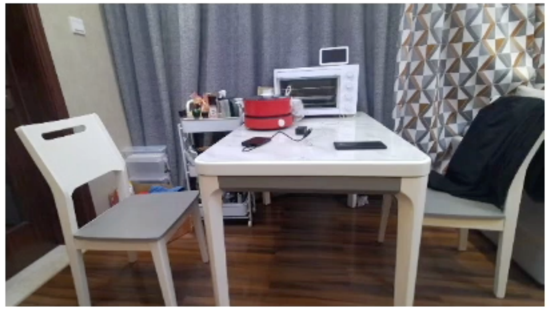
- Multi-res Hash Encoding + shallow MLP + excellent engineering
- Train a scene in <u>**seconds**</u>!

Müller et al.: <u>Instant Neural Graphics Primitives with a Multiresolution Hash Encoding</u>. SIGGRAPH 2022 Best Paper Award

What is still missing for NeRF?

Always assume camera poses given!

RGB-D Sequences



40x Speed

NICE-SLAM

Neural Implicit Scalable Encoding for SLAM

CVPR 2022

Zihan Zhu* Songyou Peng* Viktor Larsson Weiwei Xu Hujun Bao
Zhaopeng Cui Martin R. Oswald Marc Pollefeys

* Equal Contributions

ETH zürich

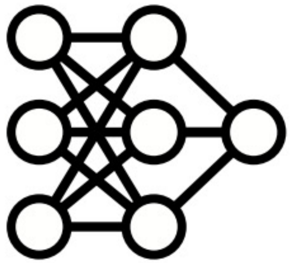
iMAP

[Sucar et al., ICCV'21]

First neural implicit-based online SLAM system

iMAP

[Sucar et al., ICCV'21]



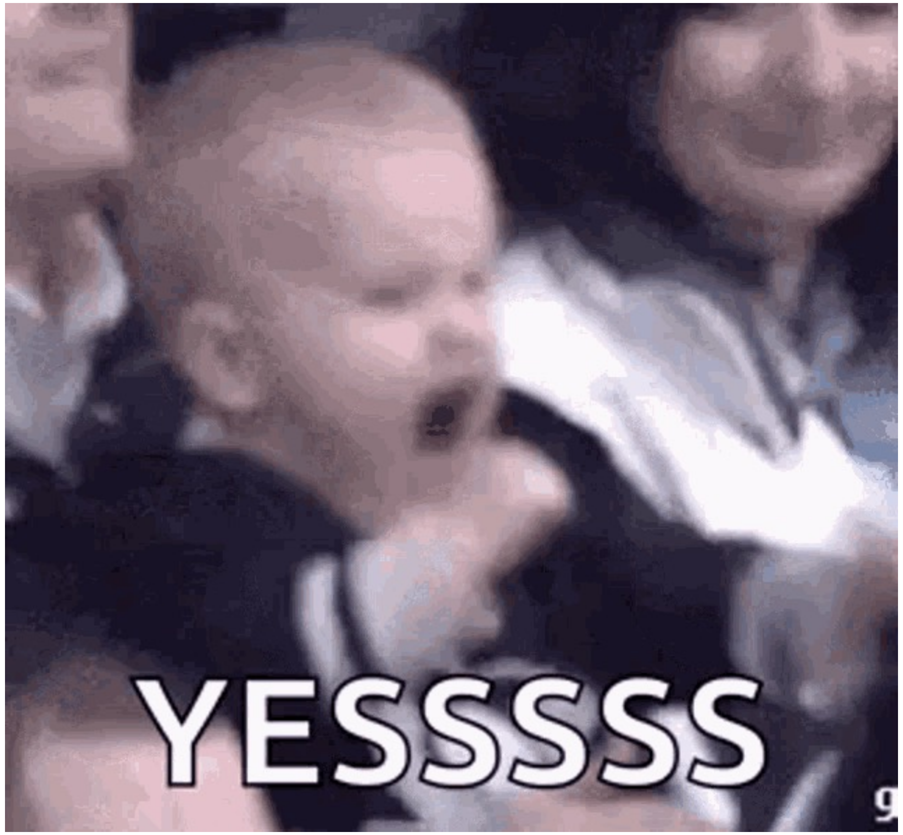
A single MLP

- Fail when scaling up to larger scenes
- Global update → Catastrophic forgetting
- Slow convergence

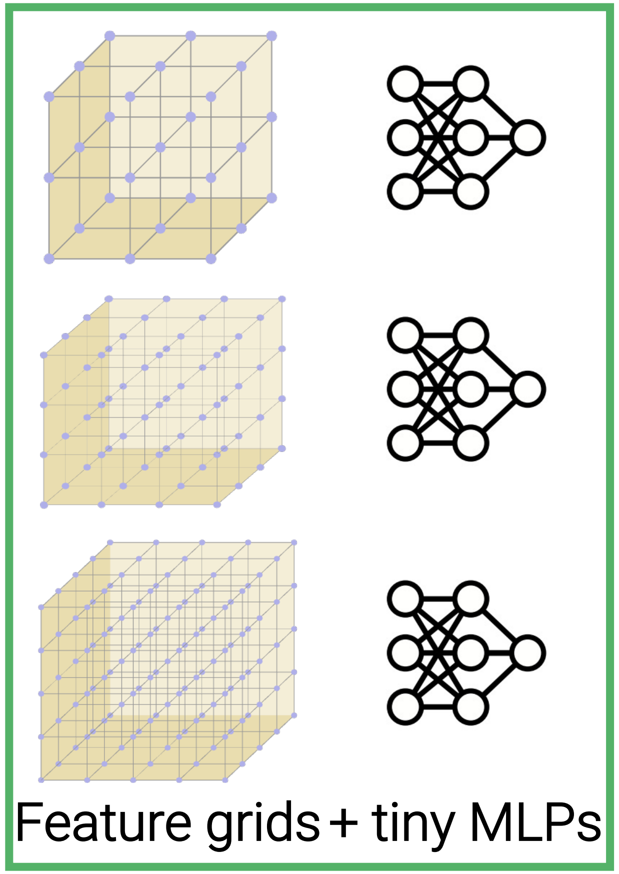
Predicted Poses

GT Poses

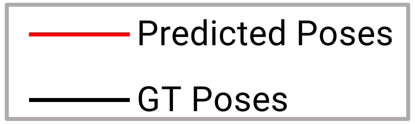
Again, can implicit-explicit representations help?



NICE-SLAM

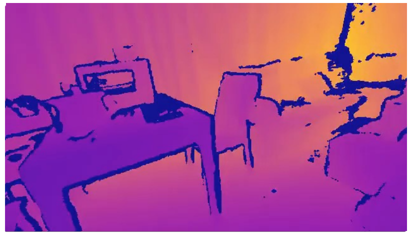


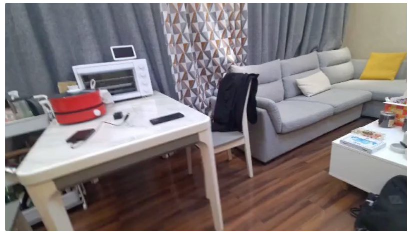
- Applicable to **large-scale scenes**
- Local update → **No forgetting problem**
- **Fast** convergence



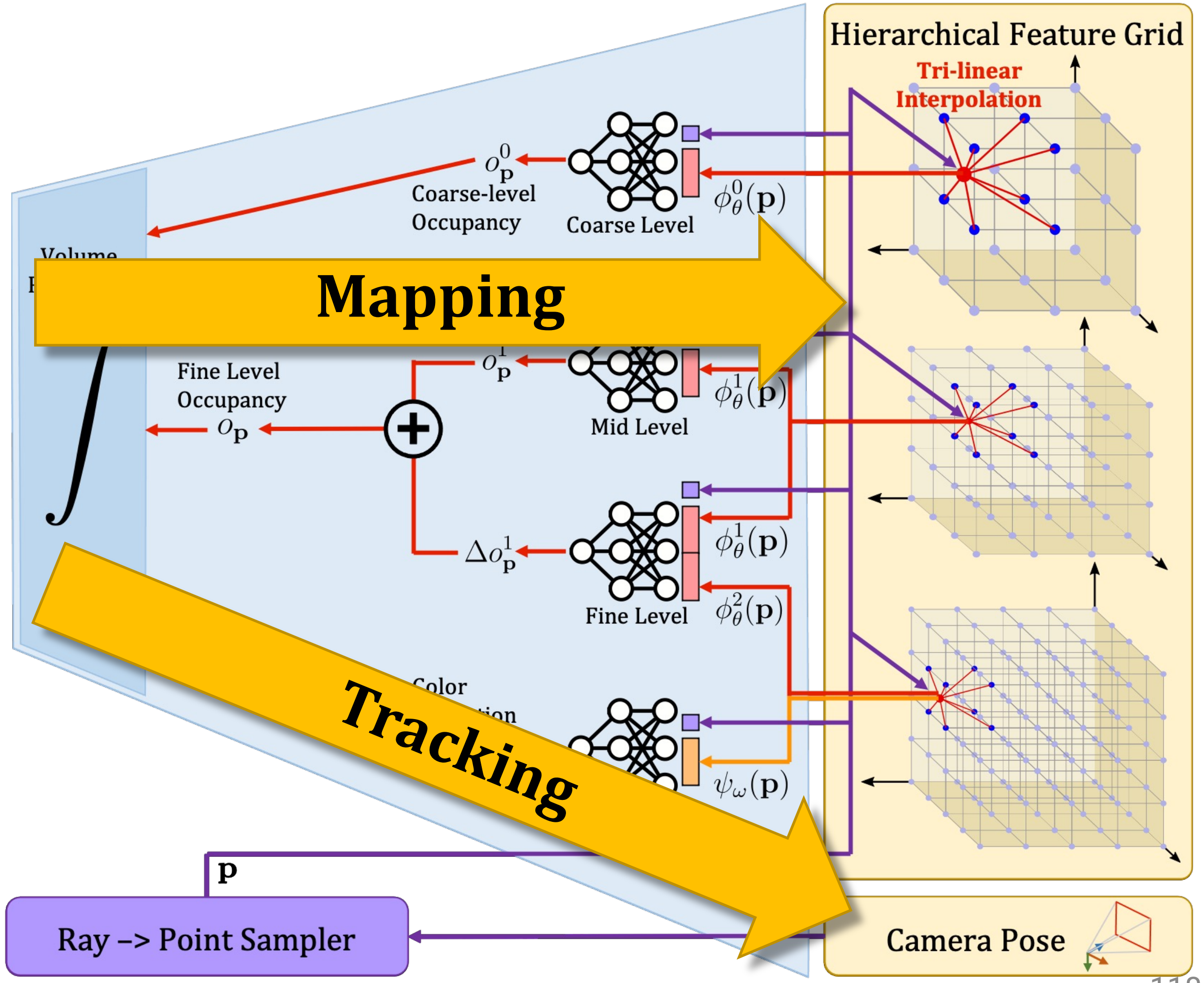
Pipeline

Input Depth





Input RGB



Results

iMAP*

(our re-implementation of iMAP)

NICE-SLAM

4x Speed

Predicted Poses

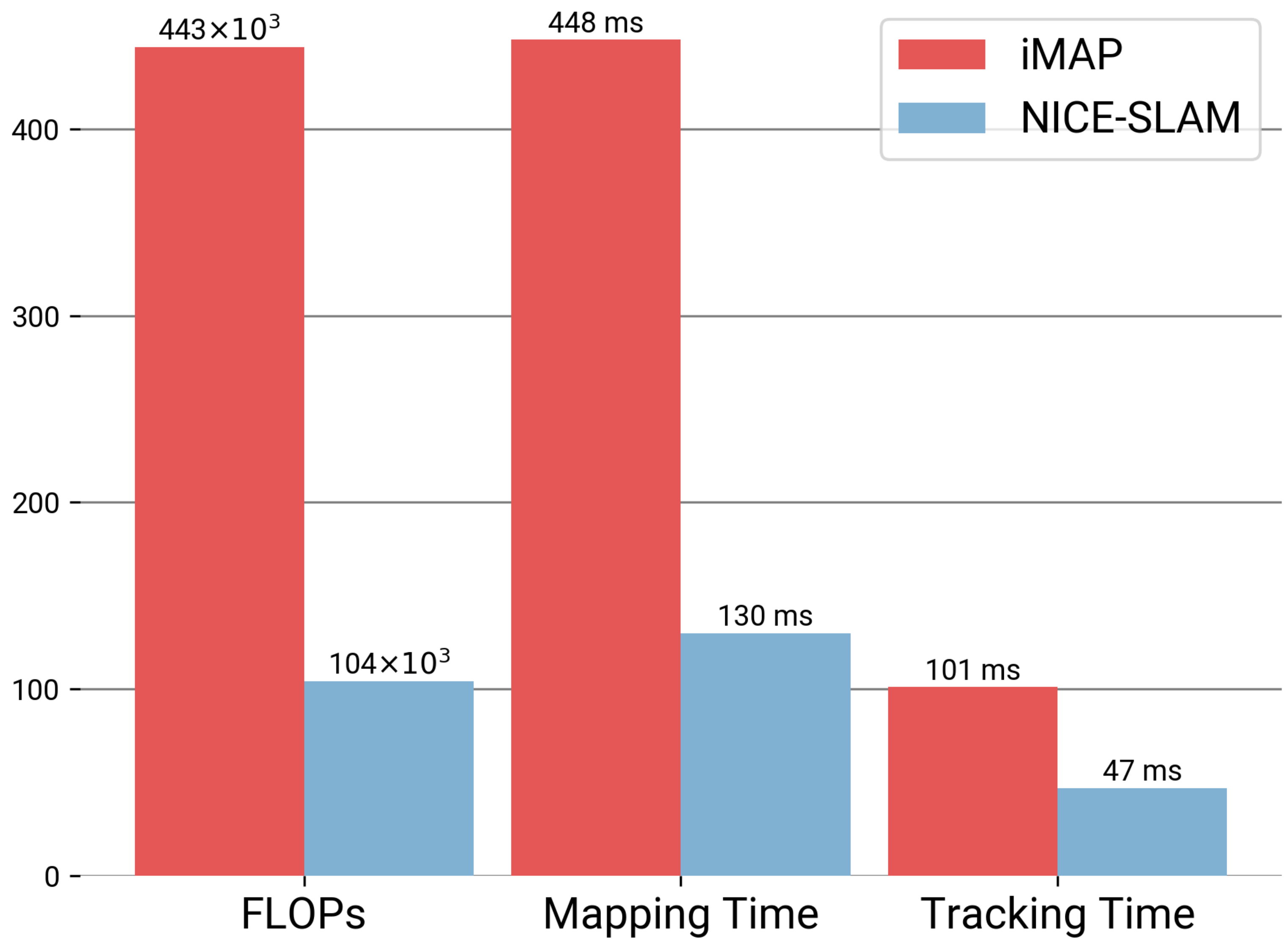
GT Poses

iMAP*

(our re-implementation of iMAP)

NICE-SLAM

10x Speed



Note: Runtime evaluation setting from iMAP paper, not the best-performing setting

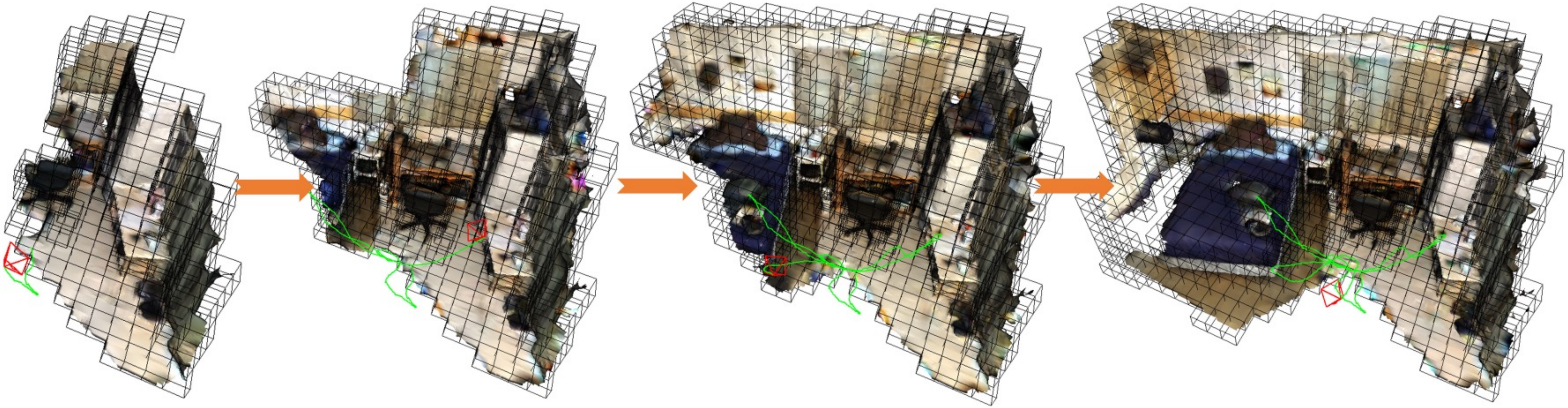
Take-home Message

- Neural explicit-implicit representation again helps!
- Hierarchical feature grids + a tiny MLP seems to be a trend!

Limitations

- Requires depths as input
- Still not real-time

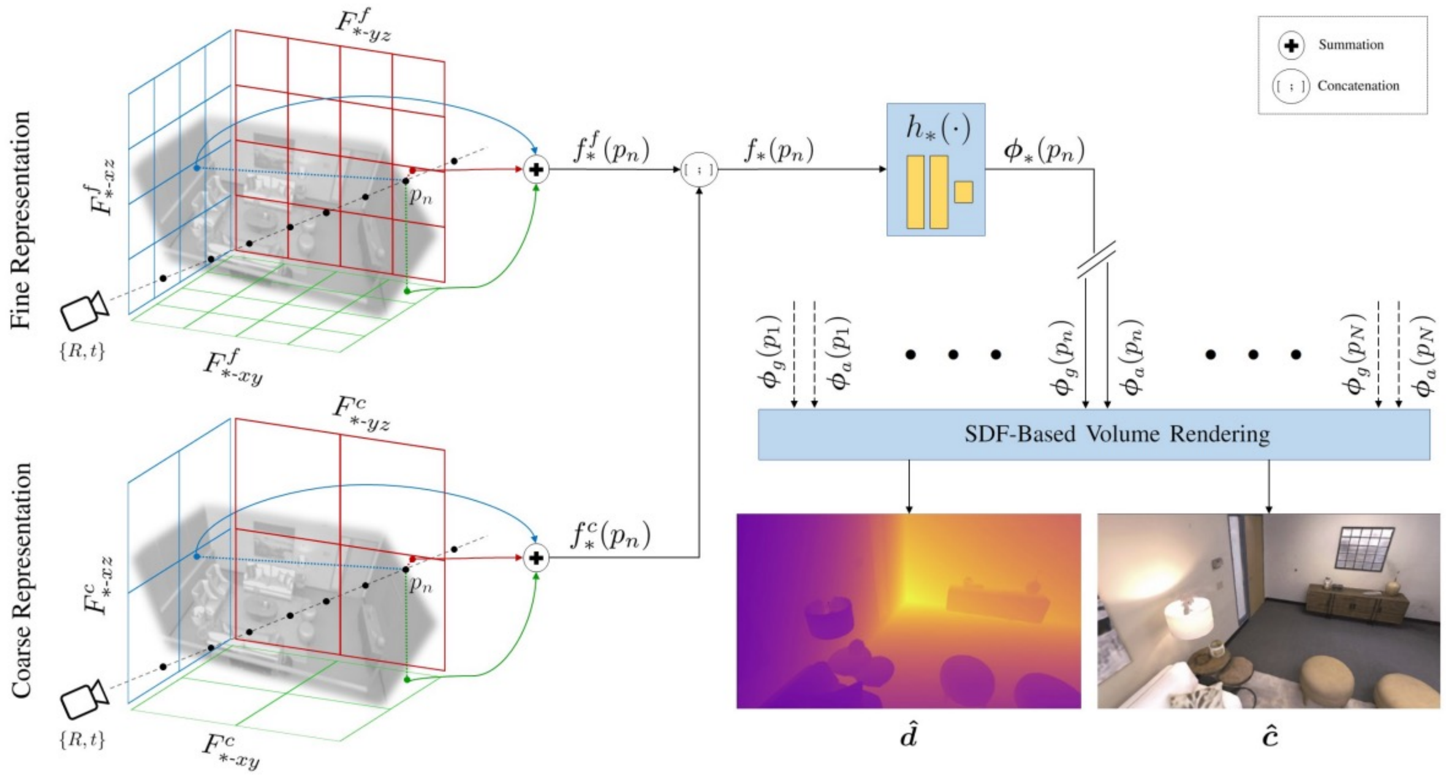
Follow-up Works: **VoxFusion**



- Gradually create voxel feature grids near to the surface
- Also more memory and time efficient

Yang et al.: Vox-Fusion: Dense Tracking and Mapping with Voxel-based Neural Implicit Representation. ISMAR 2022

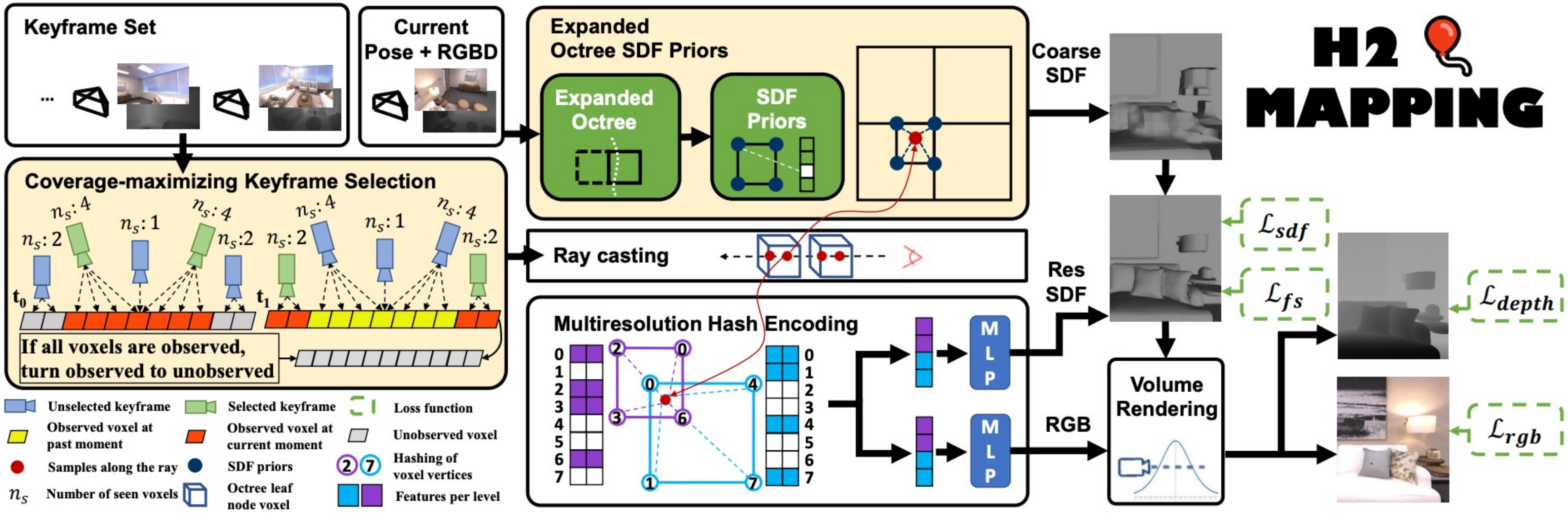
Follow-up Works: **ESLAM**



- My lovely tri-planes as the scene representation!
- Run 10x faster and 10x less memory

Mahdi Johari et al.: ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields. CVPR 2023

Follow-up Works: **H2-Mapping**



- Octree SDF representation + multi-res hash encoding
- Better engineering ⇒ real-time NeRF-based mapping

Yang et al.: H2-Mapping: Real-time Dense Mapping Using Hierarchical Hybrid Representation. Arxiv, June 2023

Related Works: **Neuralangelo**

- SDF representation + multi-res hash encoding
- Great engineering effort ⇒ High-fidelity large-scale outdoor reconstruction

Li et al.: Neuralangelo: High-Fidelity Neural Surface Reconstruction. CVPR 2023

Final Remarks

We introduced many neural explicit-implicit representations:

- Single/multi-res feature grids + MLP
- Tri-plane + MLP
- Feature octrees + MLP
- Multi-res hash encoding + MLP
- Grid of MLPs
- Poisson solver to convert point clouds ⇒ indicator grids

…… (There are sooooooo many forms of neural explicit-implicit representations)

Final Remarks

Neural explicit-implicit representations are AWESOME!!!

- Memory efficiency
- Fast training/testing speed
- Fast convergence
- Scalable, and robust to large scenes

…… Discover more yourself ☺

They truly shine through great engineering efforts!


One more thing...

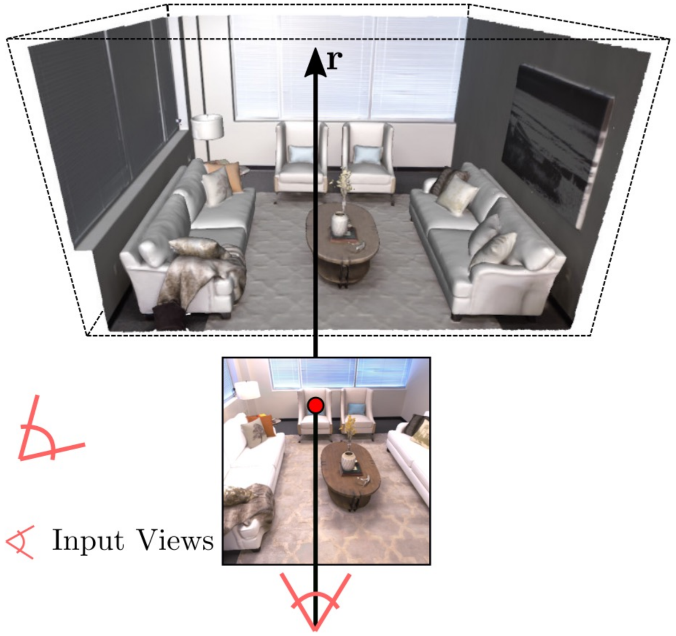
SDF Studio

A Unified Framework for Surface Reconstruction

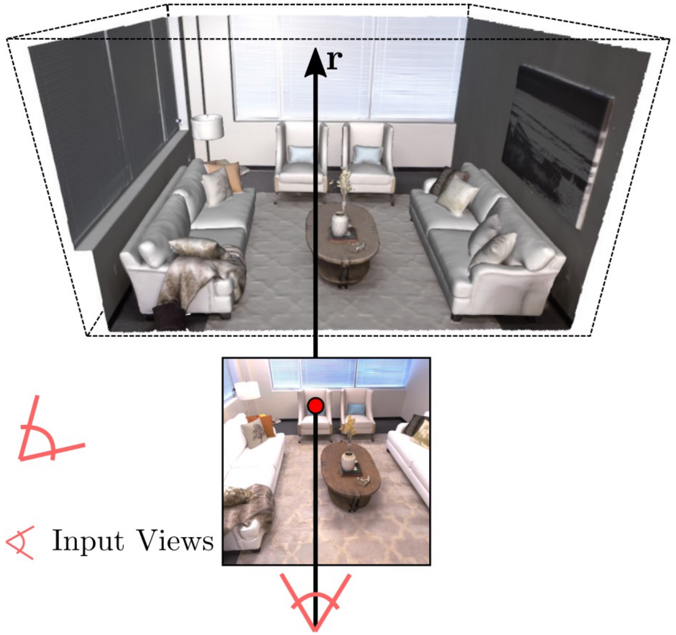
Zehao Yu[1] Anpei Chen[1,2] Bozidar Antic[1] Songyou Peng[2,3] Apratim Bhattacharyya[1]
Michael Niemeyer[1,3] Siyu Tang[2] Torsten Sattler[4] Andreas Geiger[1,3]
[1]University of Tübingen [2]ETH Zurich [3]MPI for Intelligent Systems, Tübingen
[4]Czech Technical University in Prague

https://github.com/autonomousvision/sdfstudio

Overview of SDFStudio

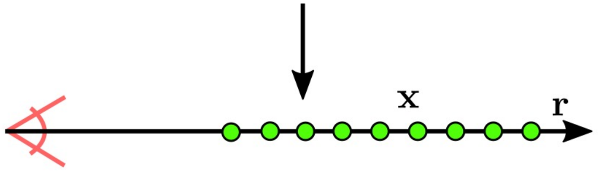


Overview of SDFStudio

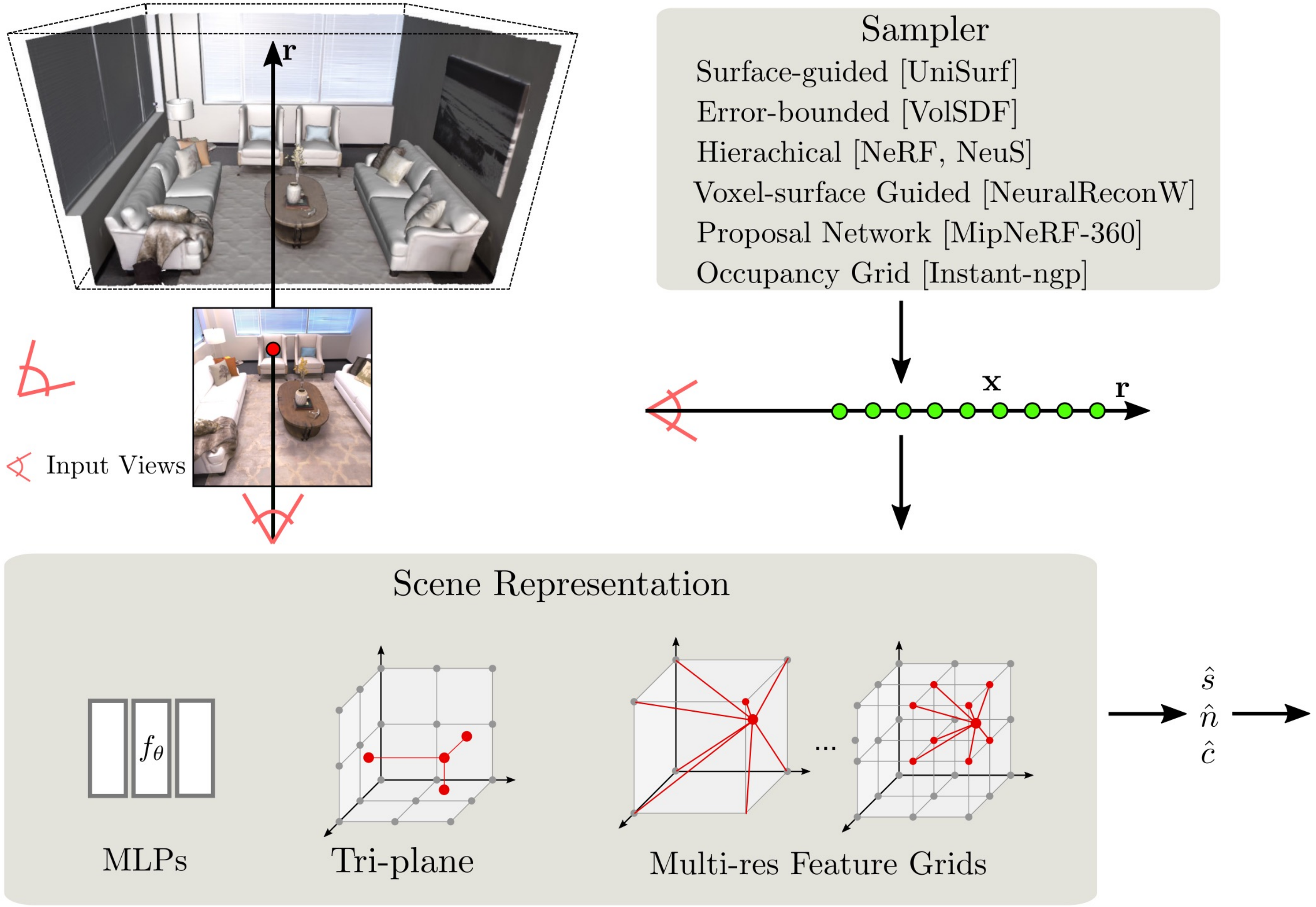


Sampler

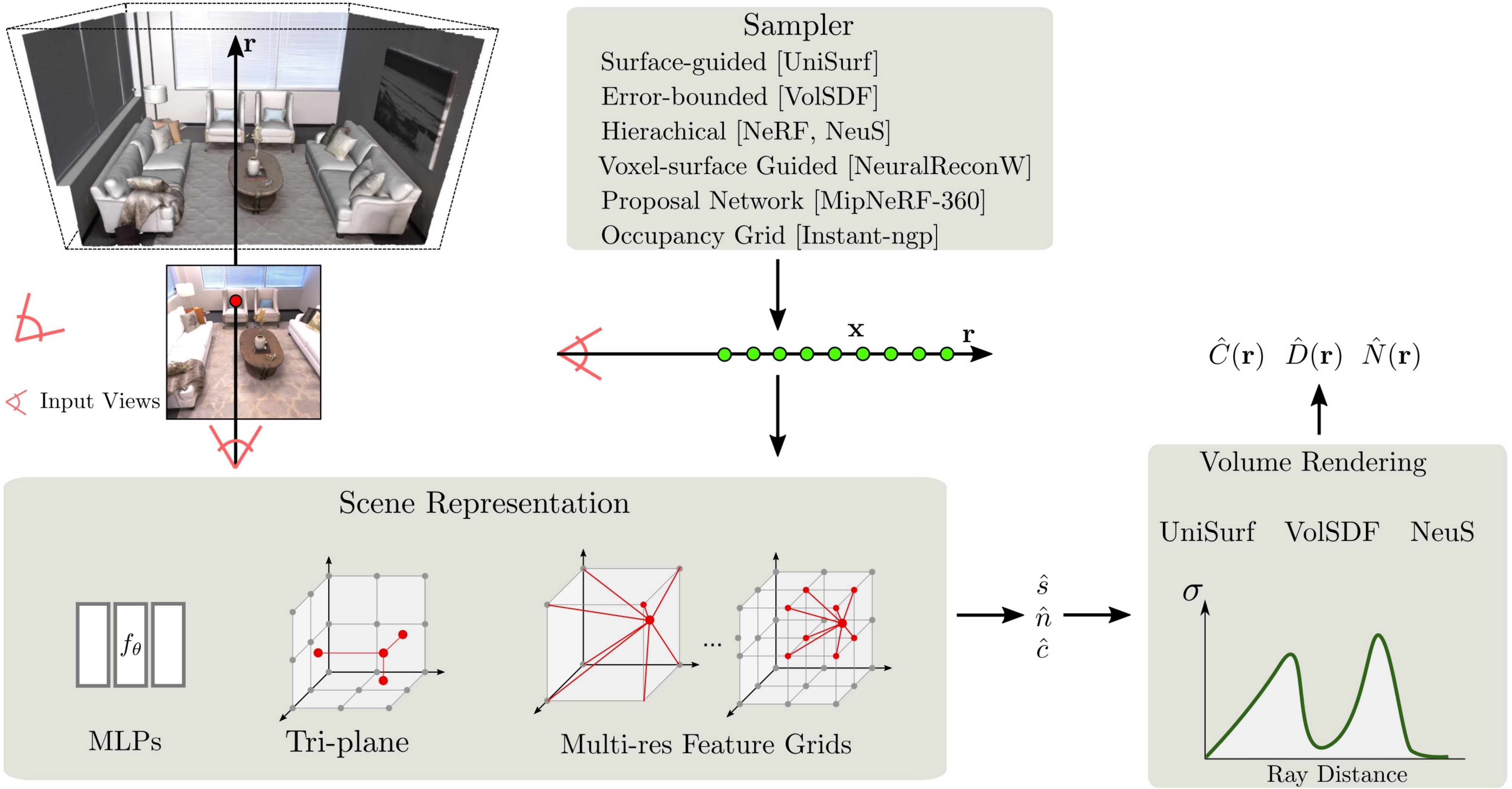
Surface-guided [UniSurf]
Error-bounded [VolSDF]
Hierachical [NeRF, NeuS]
Voxel-surface Guided [NeuralReconW]
Proposal Network [MipNeRF-360]
Occupancy Grid [Instant-ngp]



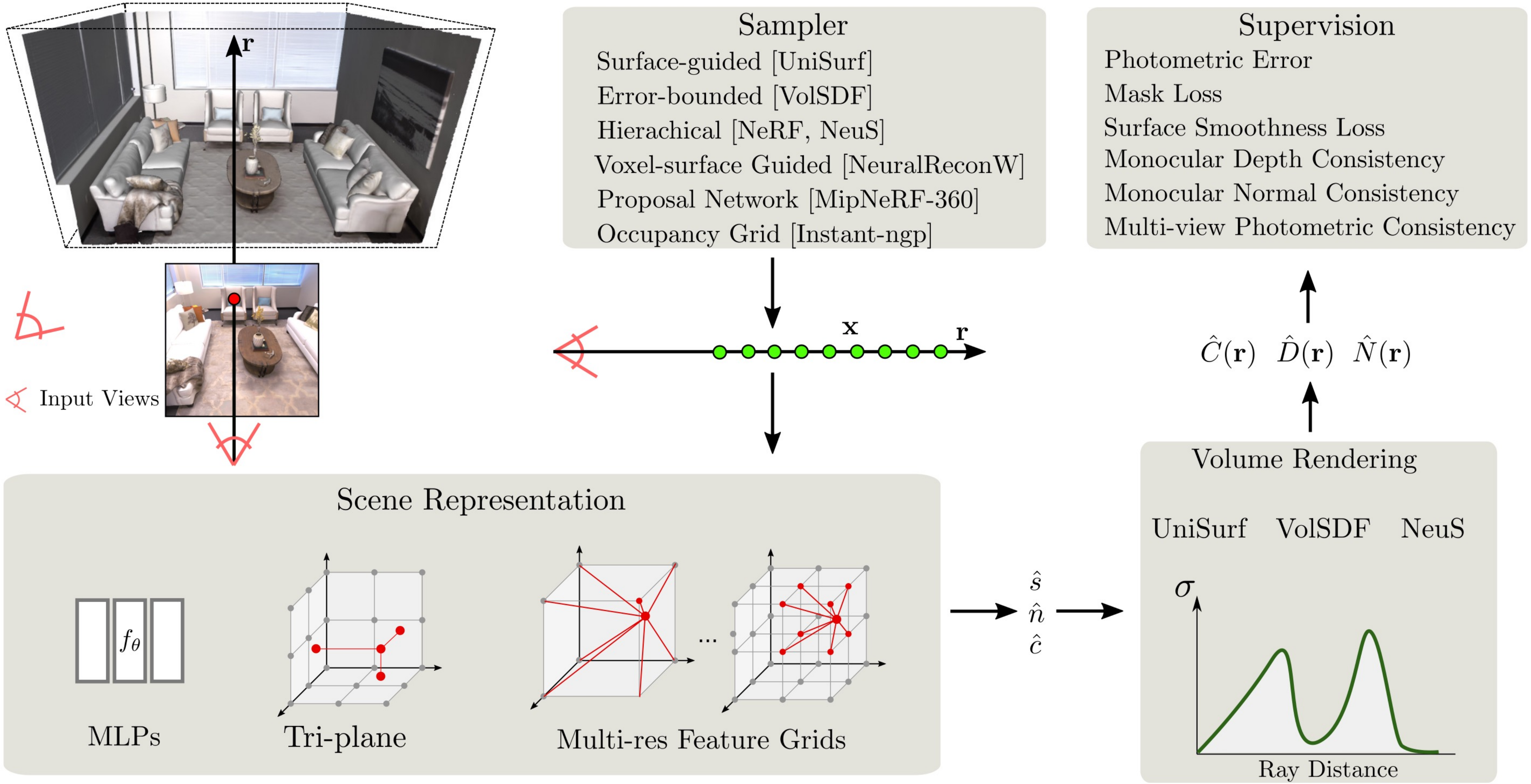
Overview of SDFStudio

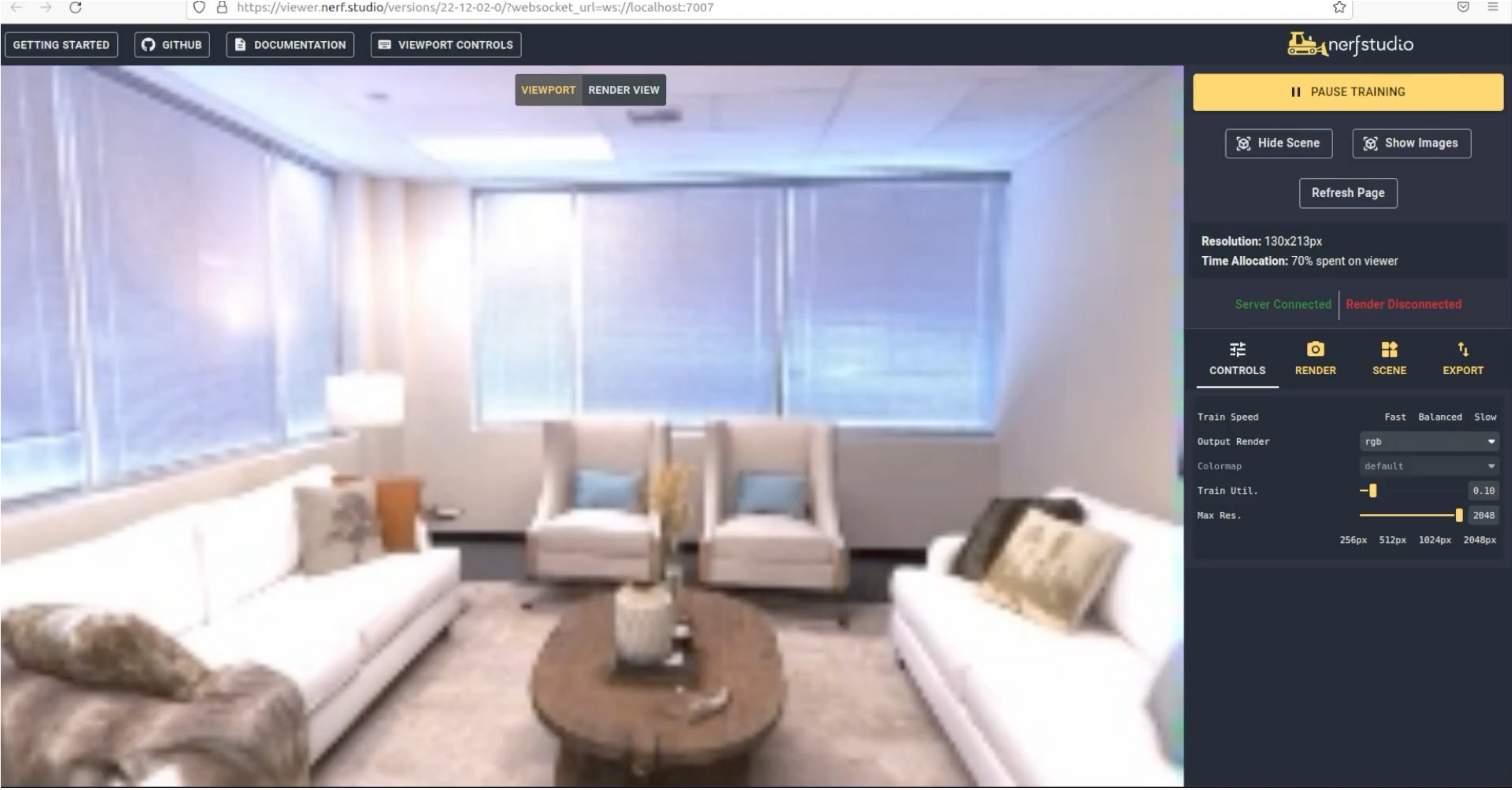


Overview of SDFStudio



Overview of SDFStudio

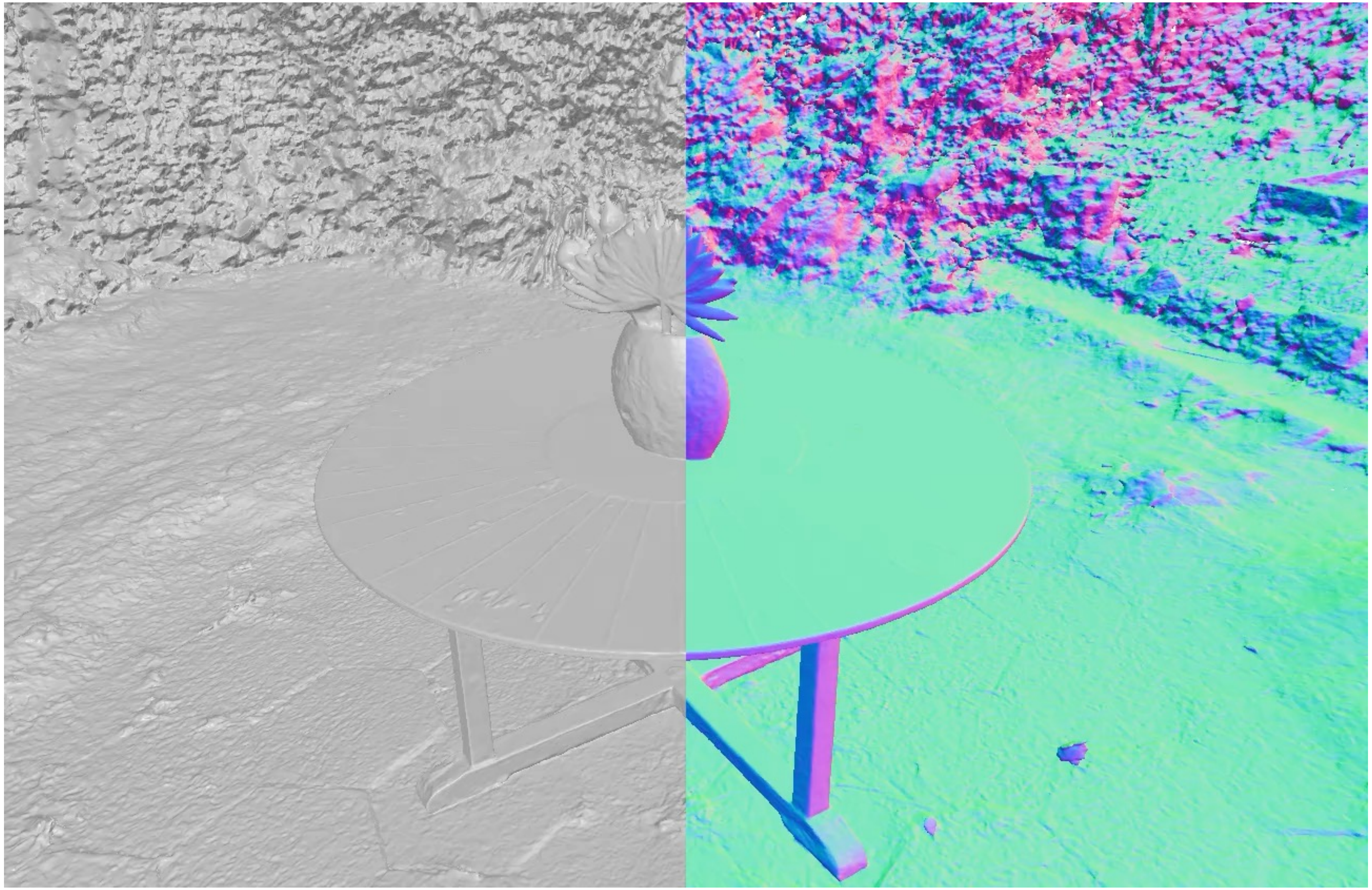




We build on top of the amazing NeRFStudio!

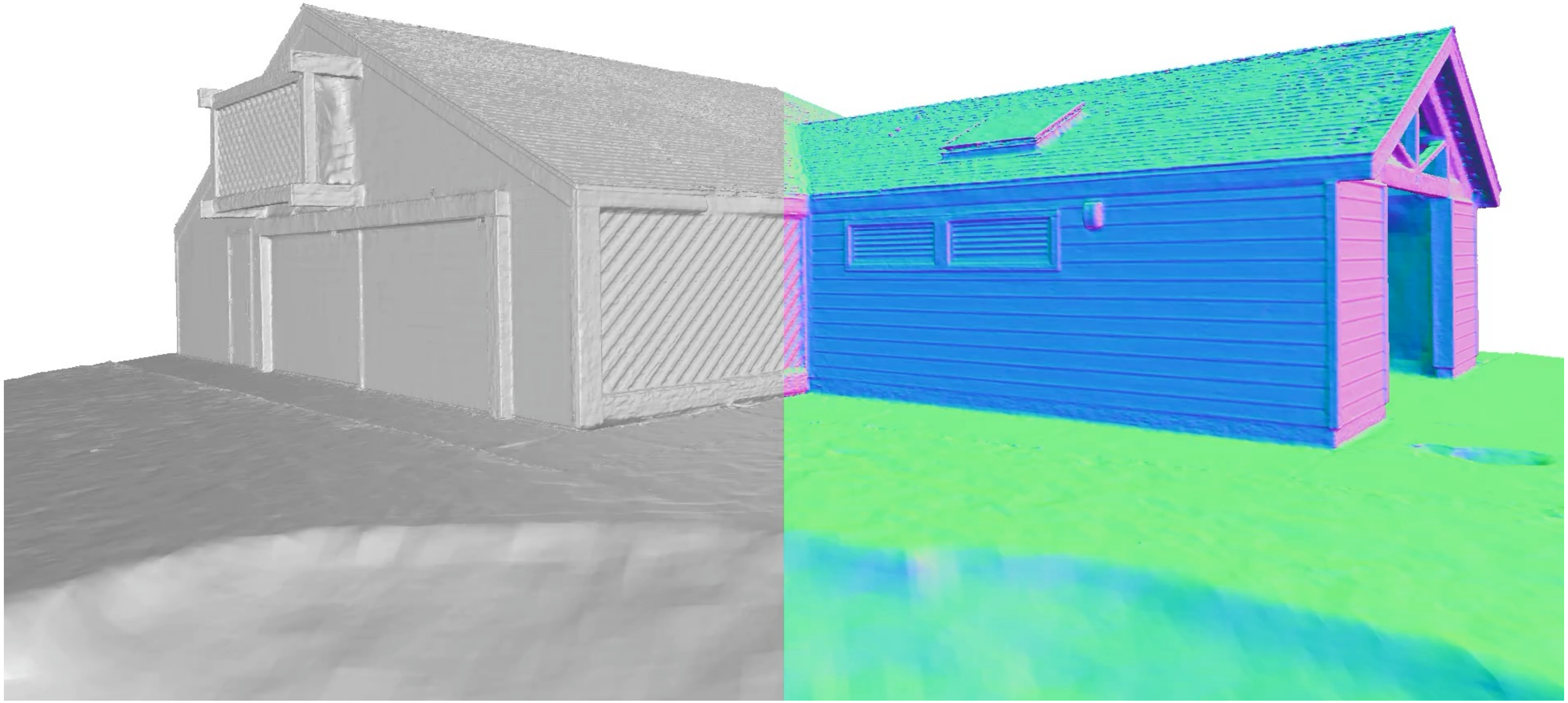
Results on outdoor scenes: Neus-facto

Results on outdoor scenes: BakedSDF

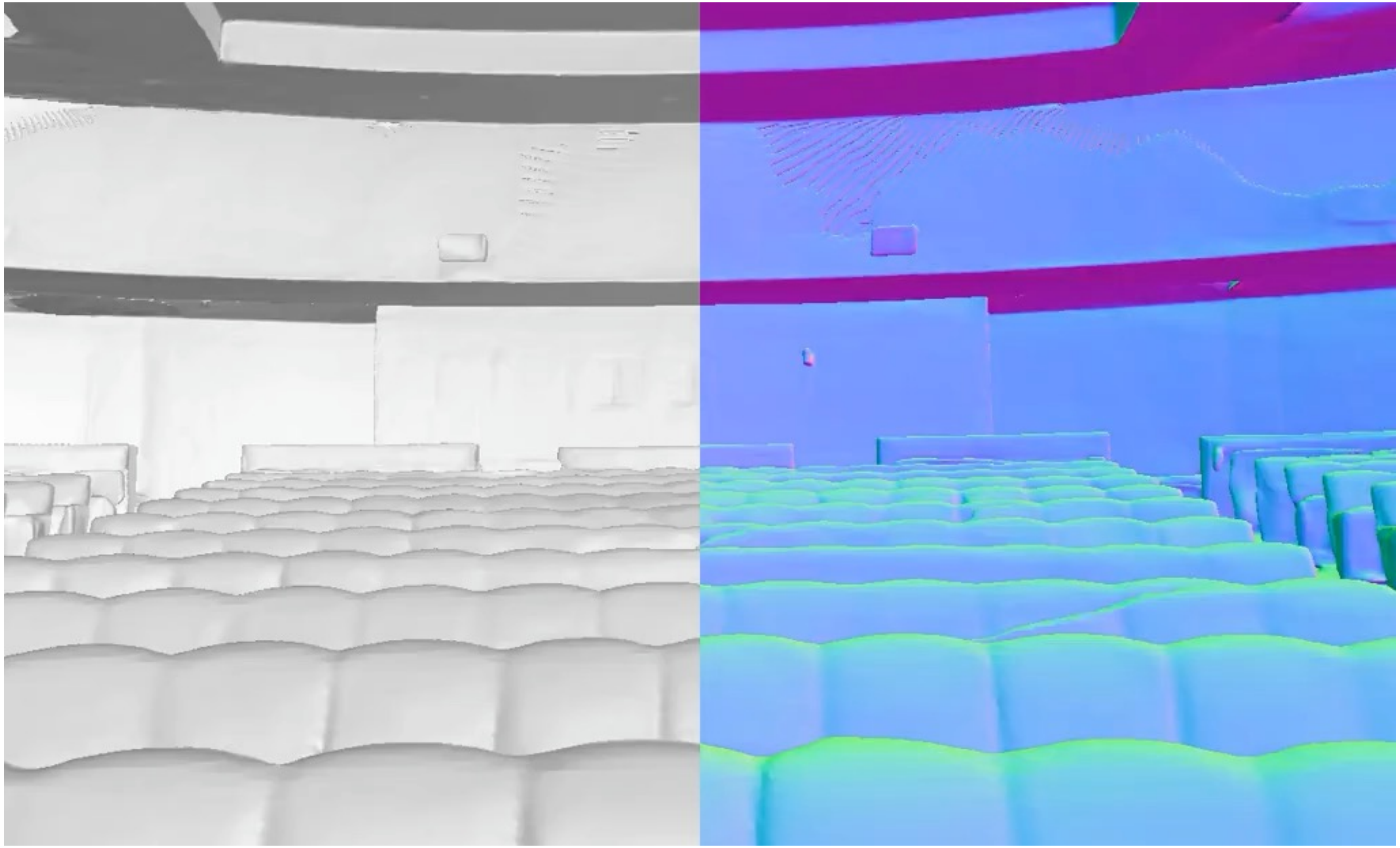


Results on outdoor scenes: Bakedangelo

Results on outdoor scenes: Bakedangelo

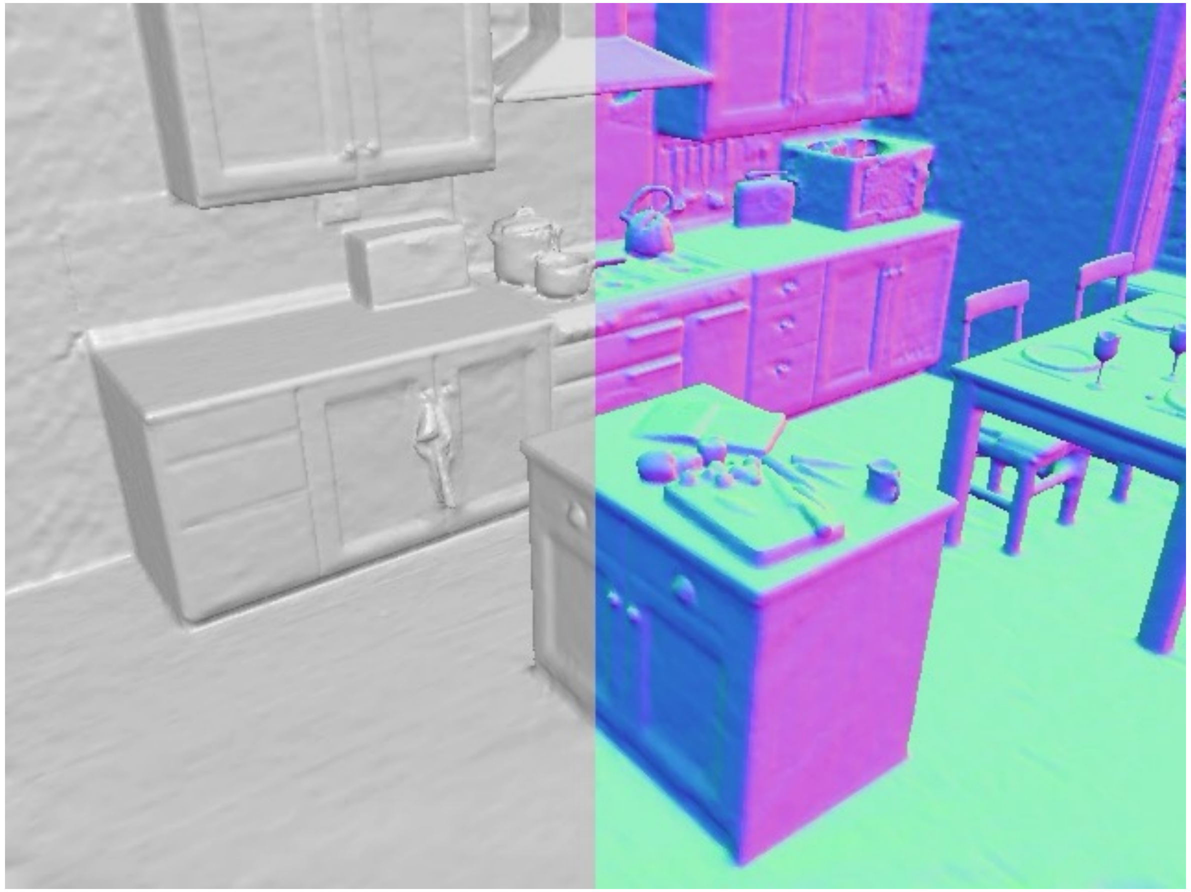


Results on indoor scenes: Mono-NeuS

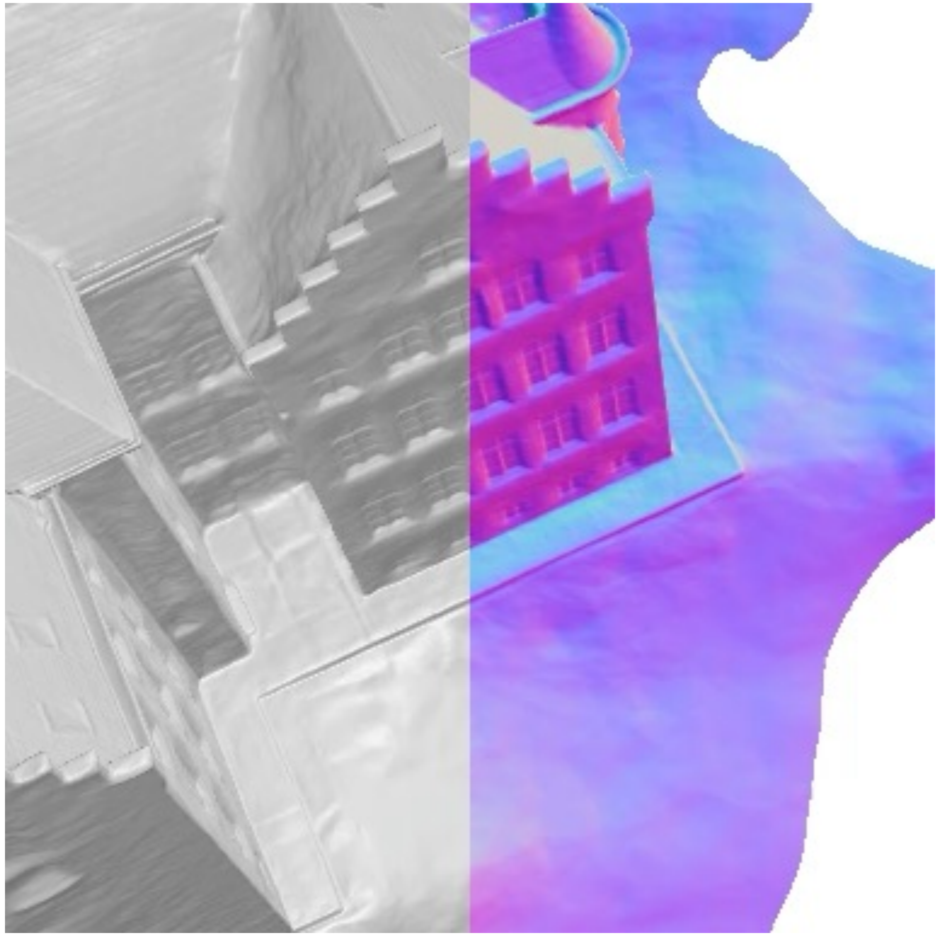


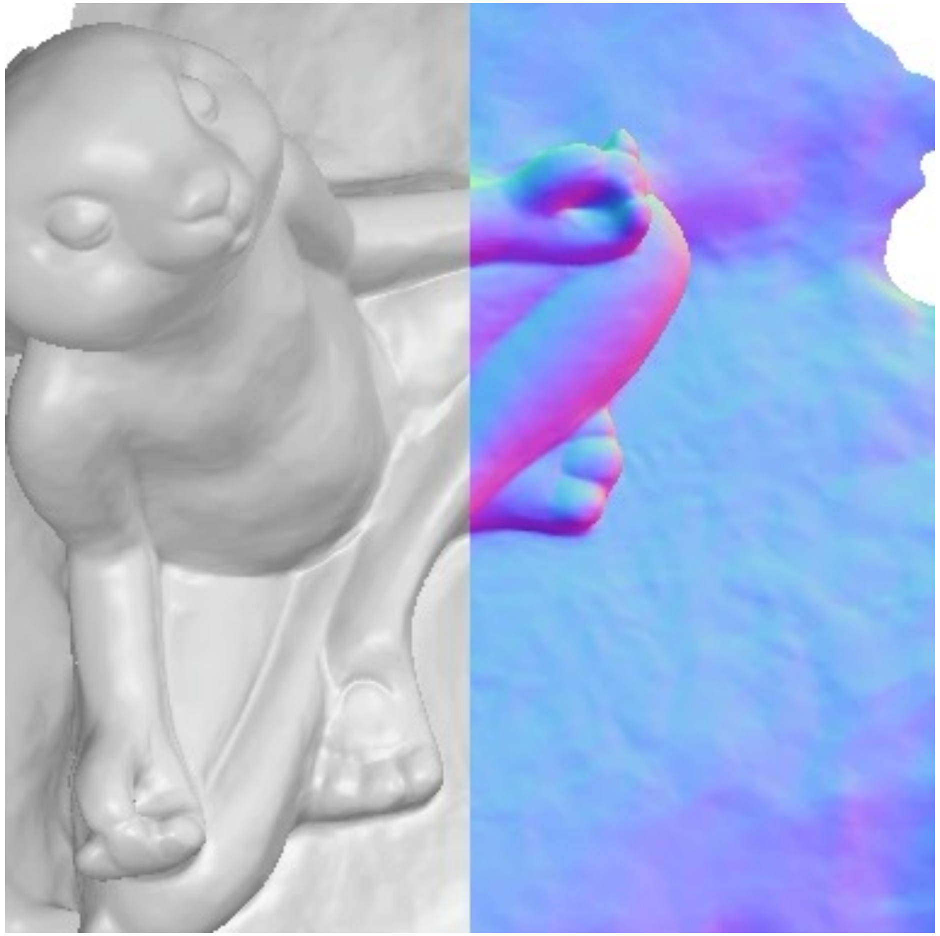
Results on indoor scenes: MonoSDF

Results on RGBD data



Results on object dataset





Dive into Neural Implicit-Explicit 3D Representations and Their Applications

Songyou Peng

pengsongyou.github.io

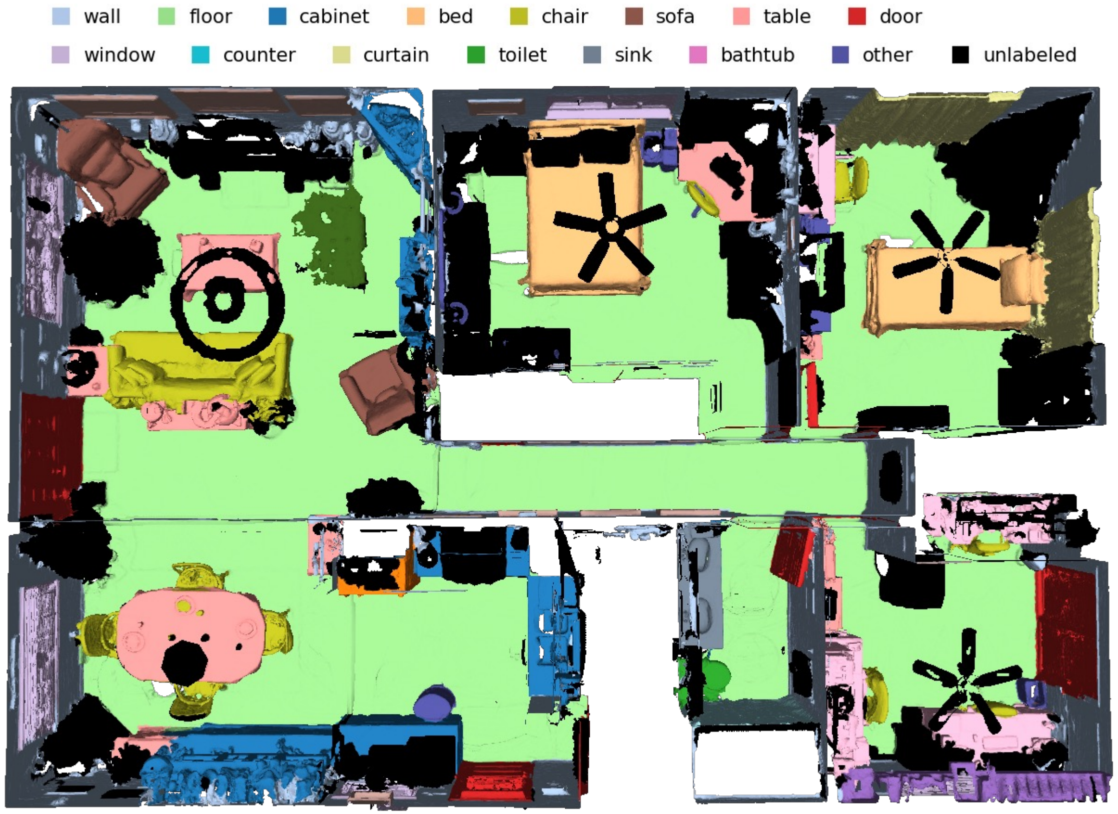
ETH Zurich and Max Planck Institute for Intelligent Systems

Symposium of Geometry Processing

July 2, 2023

Input 3D Geometry

Input 3D Geometry



Traditional Semantic Segmentation

Only train and test on a few common classes

Input 3D Geometry

- Affordance prediction
- Material identification
- Physical property estimation
- Rare object retrieval
- Activity site prediction
- Fine-grained semantic segmentation
- Many more…

3D Scene Understanding Tasks w/o Labels

OpenScene

3D Scene Understanding with Open Vocabularies

CVPR 2023

Songyou Peng

Kyle Genova

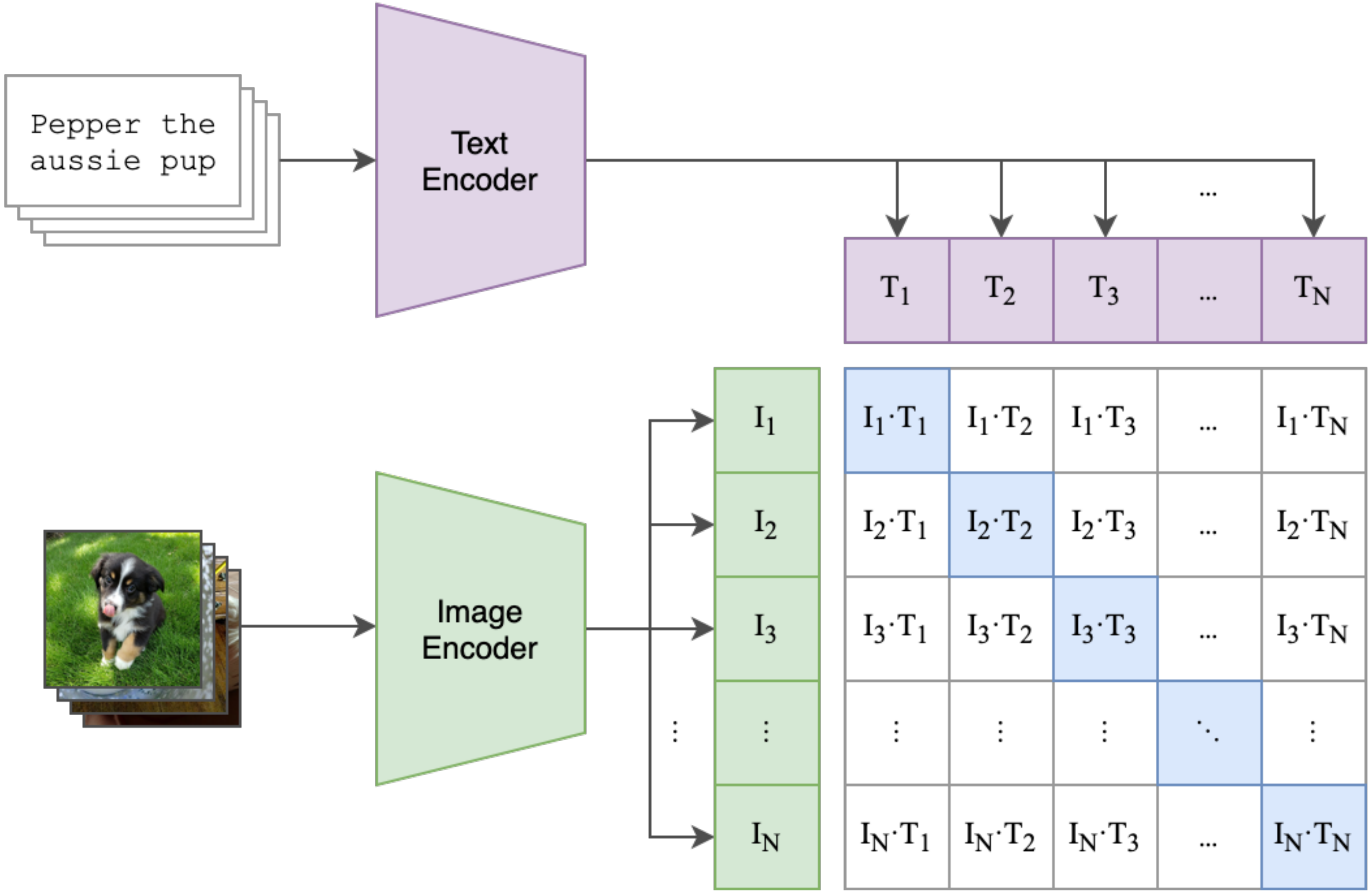
Chiyu “Max” Jiang

Andrea Tagliasacchi

Marc Pollefeys

Tom Funkhouser

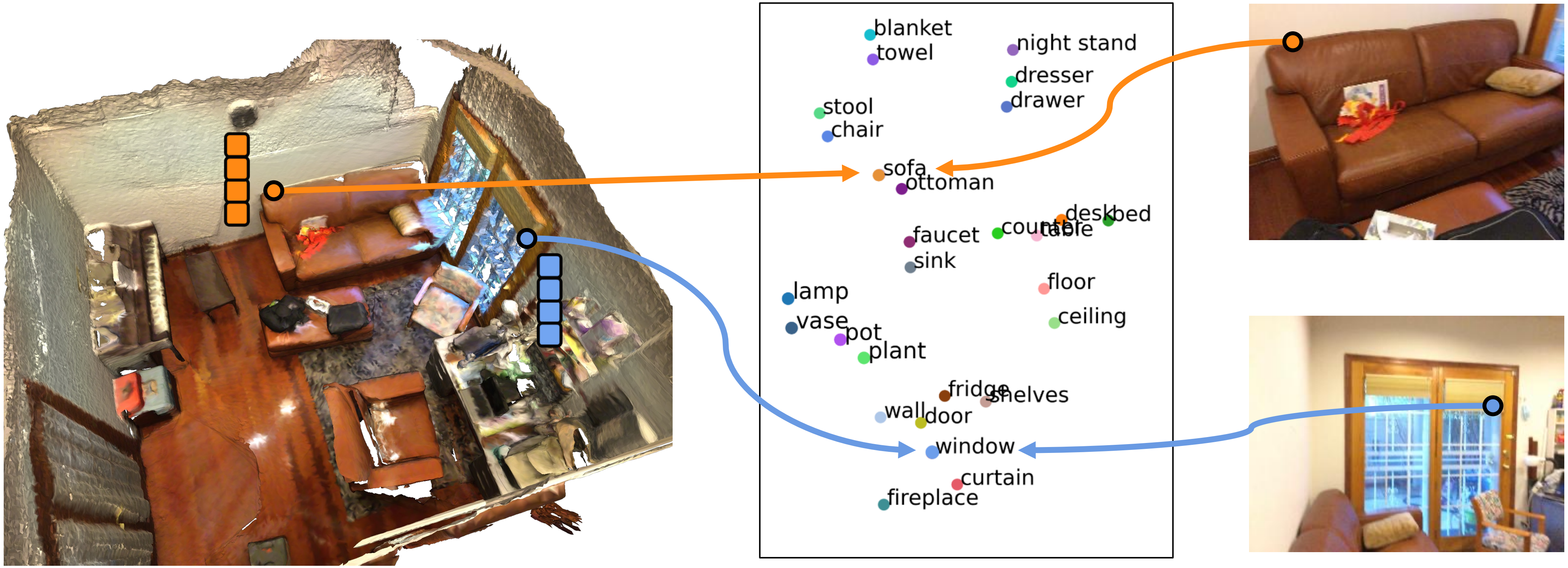
Key Idea: Co-embed 3D features with CLIP features



CLIP: Contrastive Language-Image Pre-Training

Radford et al.: Learning Transferable Visual Models From Natural Language Supervision. ICML 2021

Key Idea: Co-embed 3D features with CLIP features

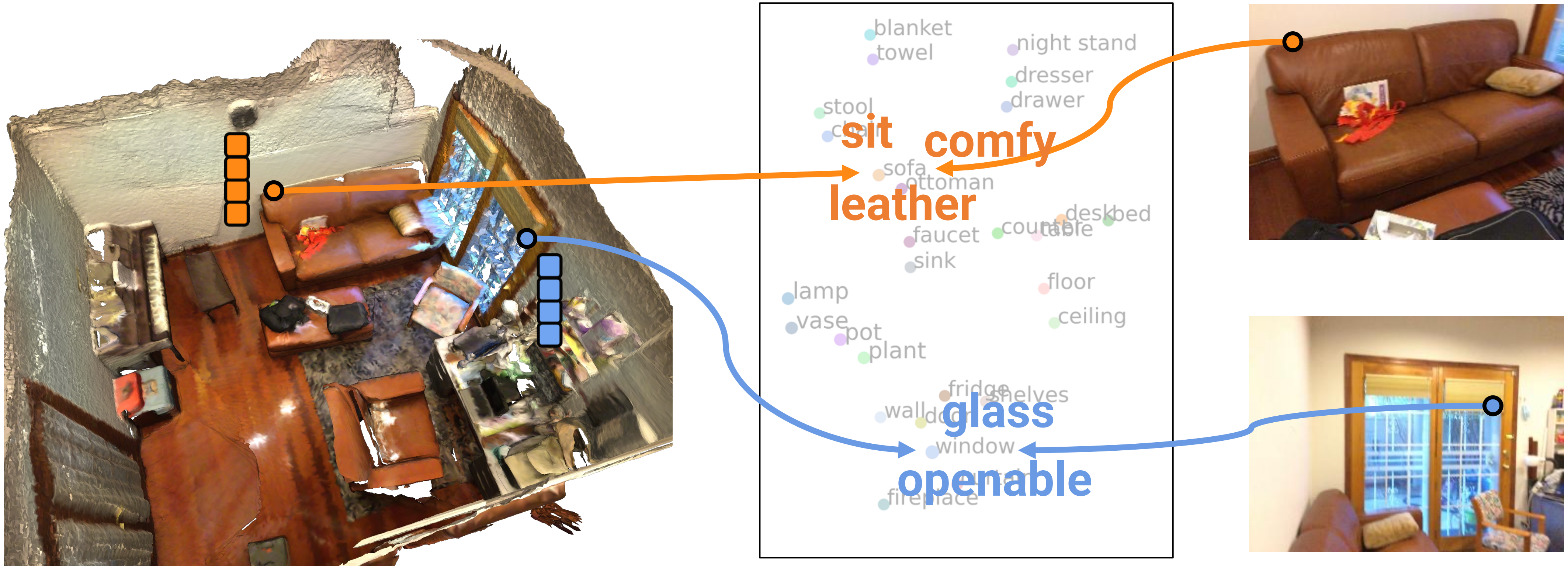


3D Geometry

CLIP Text Features
(visualize with T-SNE)

RGB Images

Key Idea: Co-embed 3D features with CLIP features



3D Geometry

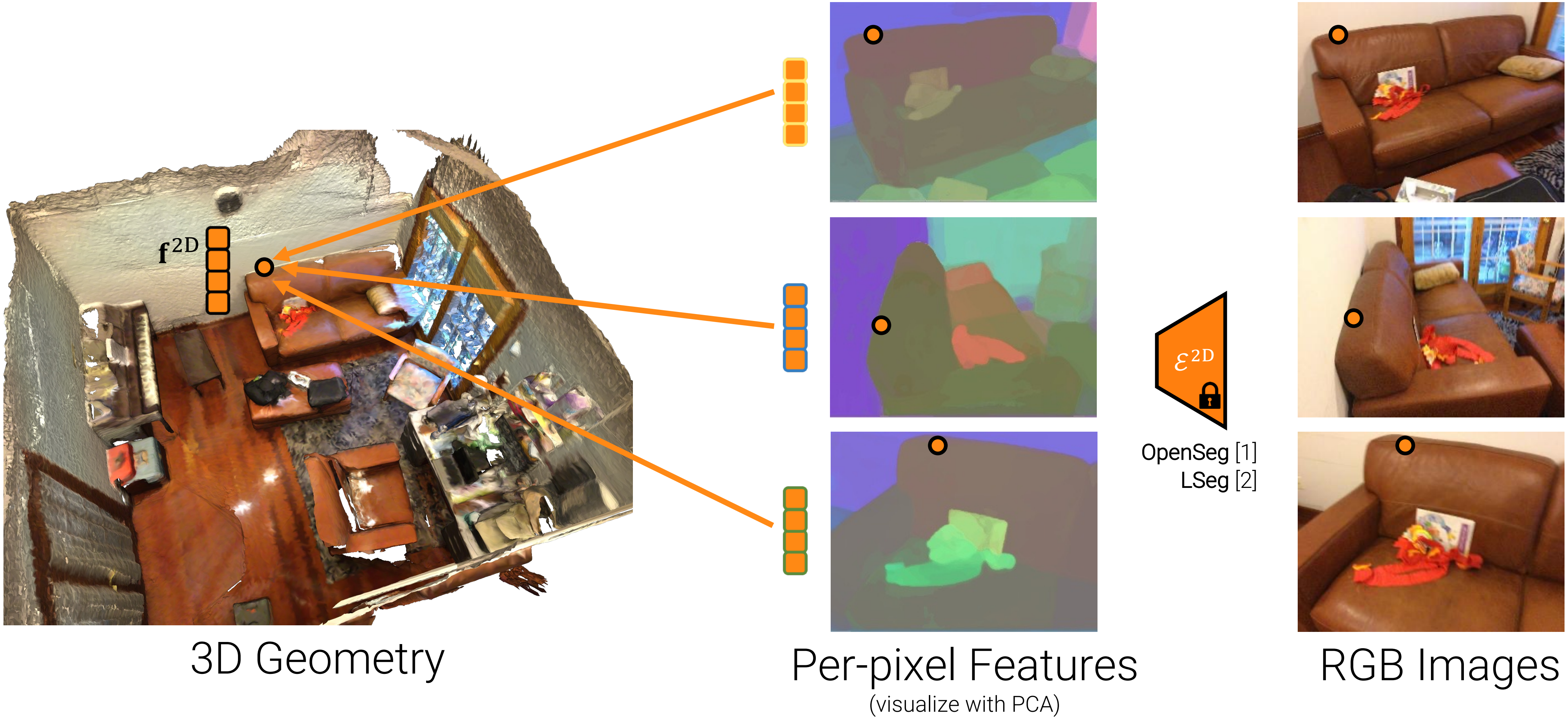
CLIP Text Features
(visualize with T-SNE)

RGB Images

Note: bold word embeddings are approximate

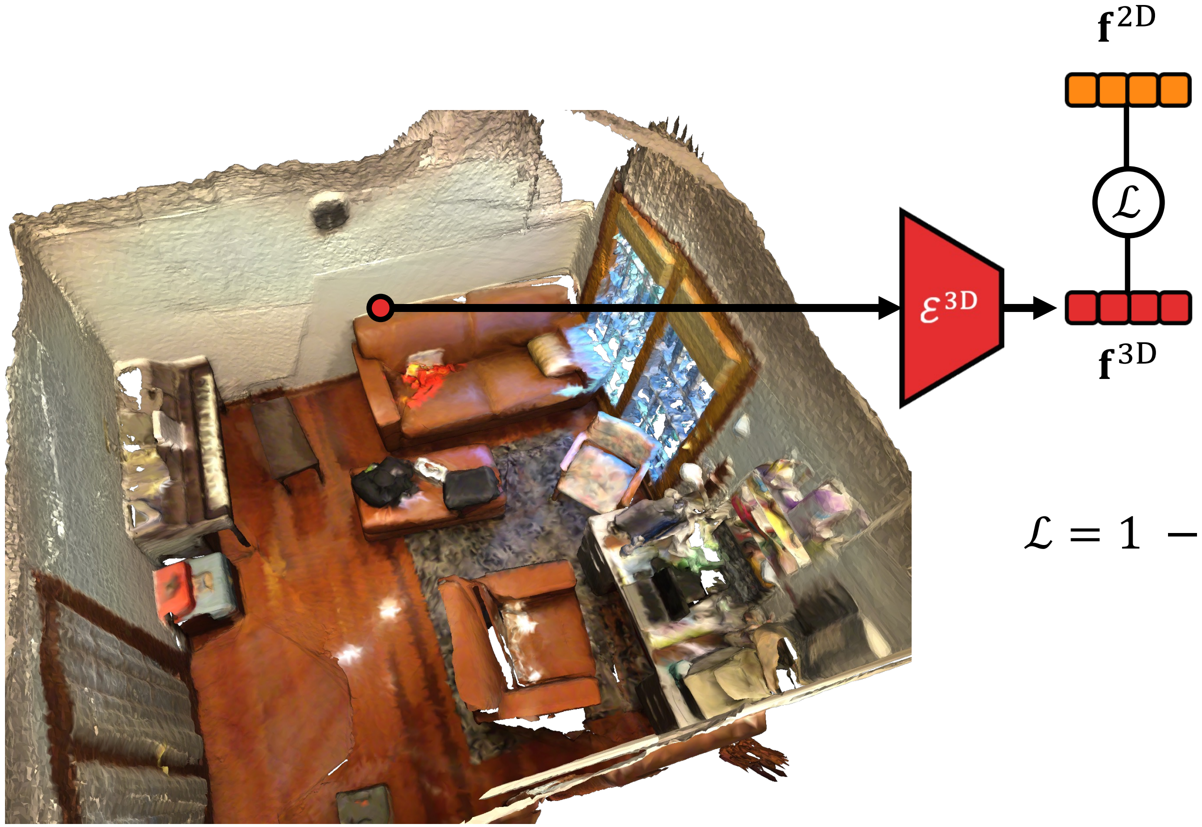
How to Learn Such Text-Image-3D Co-Embeddings?

Step 1: Multi-view Feature Fusion



[1] Ghiasi, Gu, Cui, Lin: Scaling Open-Vocabulary Image Segmentation with Image-Level Labels. ECCV 2022
[2] Li, Weinberger, Belongie, Koltun, Ranftl: Language-driven Semantic Segmentation. ICLR 2022

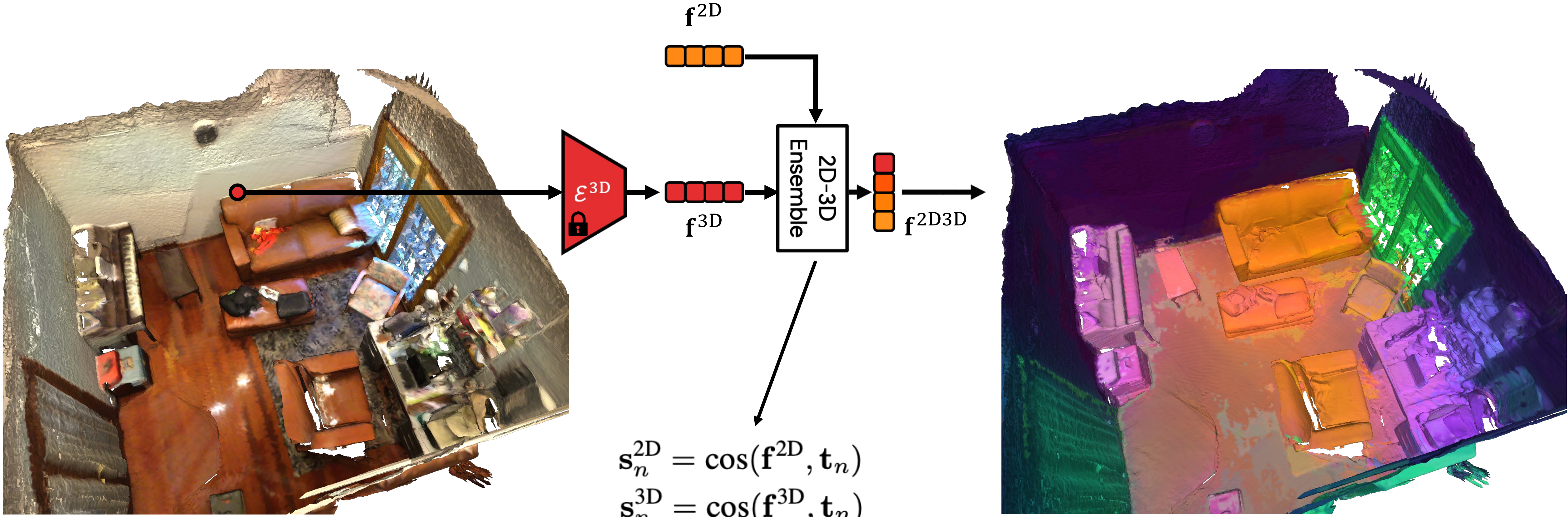
Step 2: 3D Distillation



3D Geometry

$$\mathcal{L} = 1 - \cos(\mathbf{f}^{2\mathrm{D}} - \mathbf{f}^{3\mathrm{D}})$$

Step 3: 2D-3D Ensemble



$$\mathbf{s}_n^{2D} = \cos(\mathbf{f}^{2D}, \mathbf{t}_n)$$
$$\mathbf{s}_n^{3D} = \cos(\mathbf{f}^{3D}, \mathbf{t}_n)$$

3D Geometry

2D-3D Ensemble Features
(visualize with PCA)

Choose the feature with
the highest max score among all prompts

Open-Vocabulary, Zero-shot

3D Semantic Segmentation

Input 3D Geometry

Our Zero-shot 3D Segmentation
(20 classes)

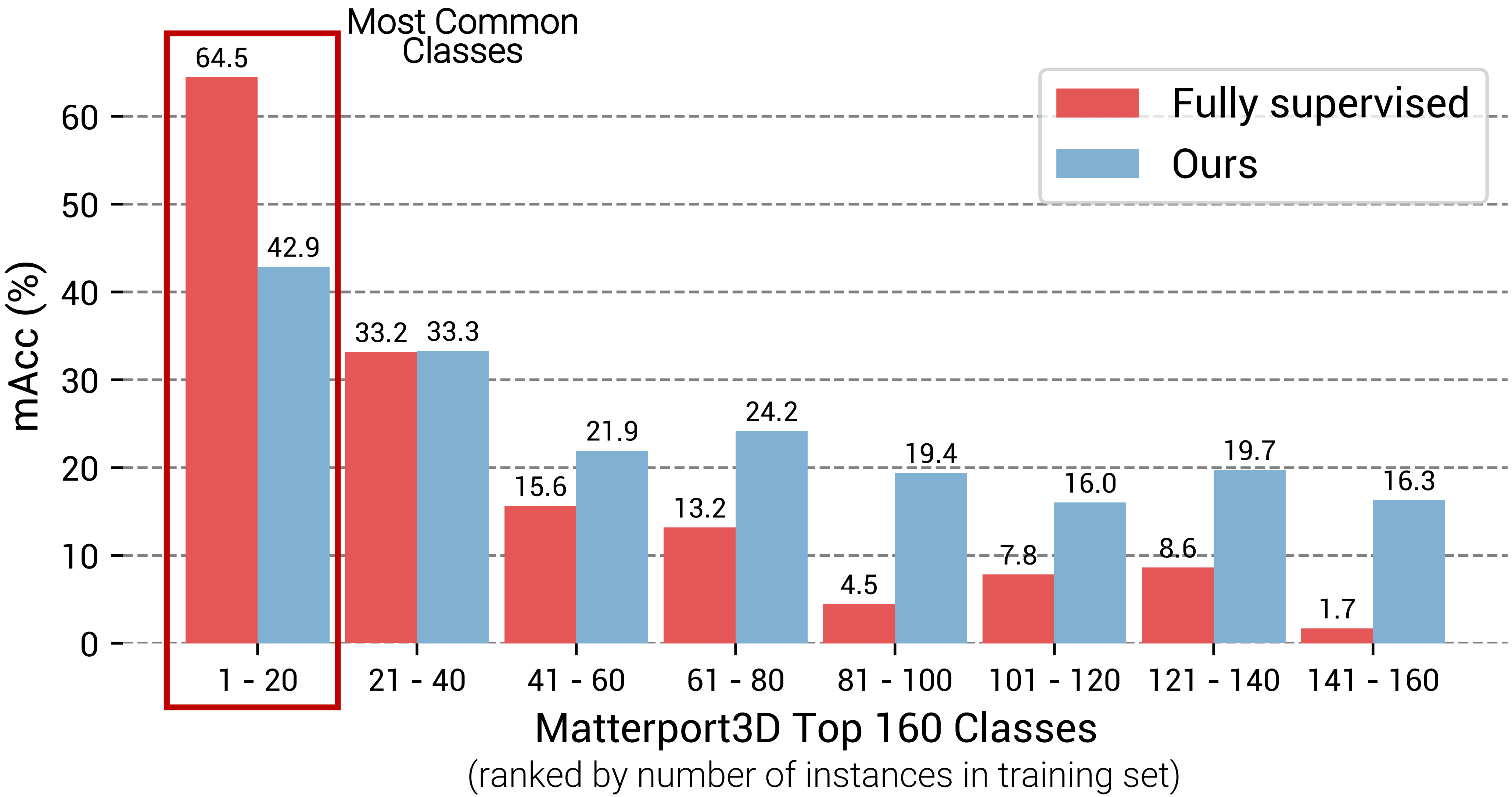
wall floor cabinet bed chair sofa table door window bookshelf picture counter desk curtain refrigerator shower curtain toilet sink bathtub other

Our Zero-shot 3D Segmentation

(160 classes)

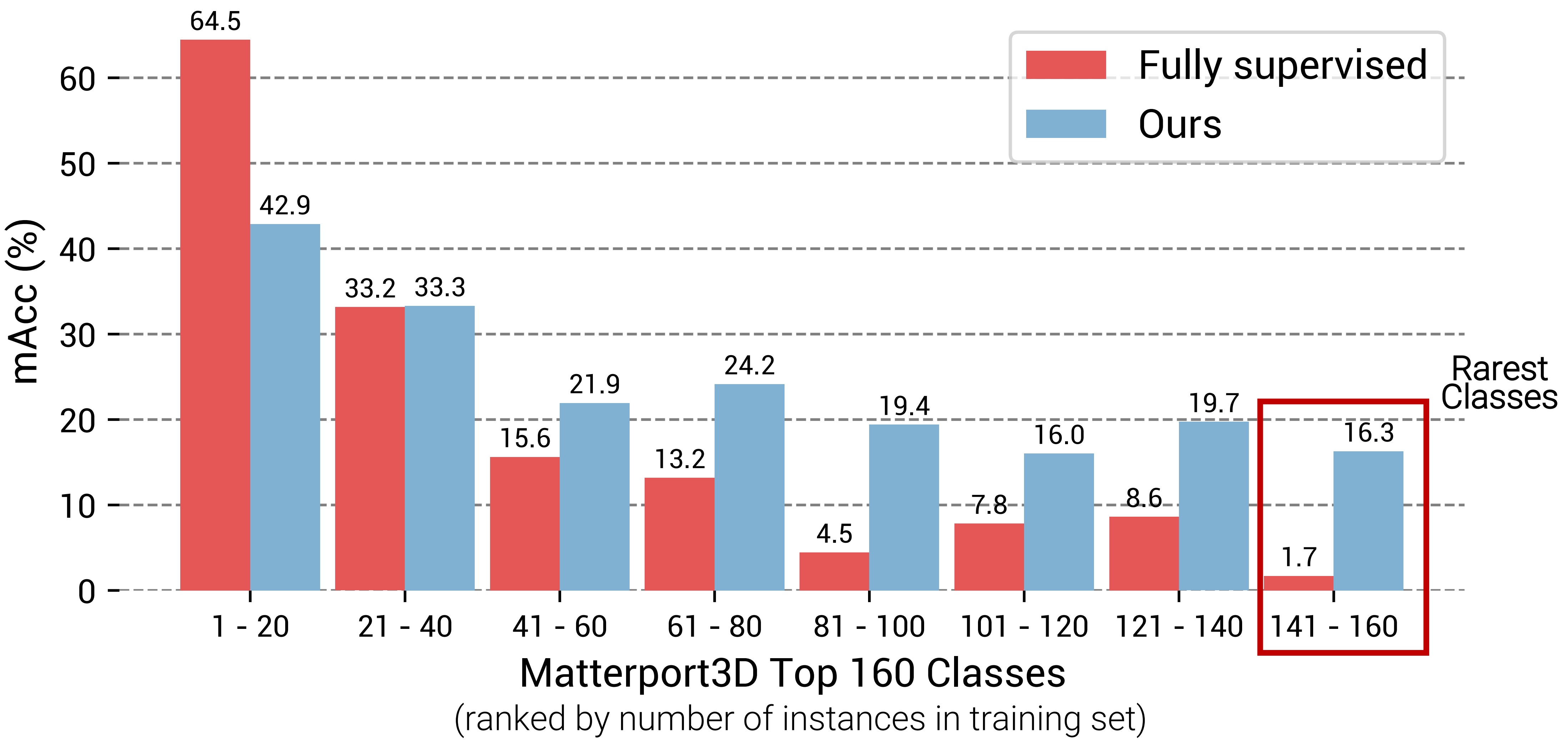
wall	cabinet	bed	pot	bathtub	dresser	stand	clock	tissue box	furniture	soap	cup	hanger	urn	paper towel dispenser	toy
door	curtain	night stand	desk	book	rug	drawer	stove	tv stand	air conditioner	thermostat	ladder	candlestick	decorative plate	lamp shade	foot rest
ceiling	table	toilet	box	air vent	ottoman	container	washing machine	shoe	fire extinguisher	radiator	garage door	light	pool table	car	soap dish
floor	plant	column	coffee table	faucet	bottle	light switch	shower curtain	heater	curtain rod	kitchen island	piano	scale	jacket	toilet brush	cleaner
picture	mirror	banister	counter	photo	refridgerator	purse	bin	headboard	printer	paper towel	board	bag	bottle of soap	drum	computer
window	towel	stairs	bench	toilet paper	bookshelf	door way	chest	bucket	telephone	sheet	rope	display case	water cooler	whiteboard	knob
chair	sink	stool	garbage bin	fan	wardrobe	basket	microwave	candle	blanket	glass	ball	toilet paper holder	tea pot	range hood	paper
pillow	shelves	vase	fireplace	railing	pipe	chandelier	blinds	flower pot	handle	dishwasher	excercise equipment	tray	stuffed animal	candelabra	projector

Comparison



Most Common Classes
64.5
42.9
33.2
33.3
15.6
21.9
13.2
24.2
4.5
19.4
7.8
16.0
8.6
19.7
1.7
16.3
Fully supervised
Ours
mAcc (%)
0
10
20
30
40
50
60
1 - 20
21 - 40
41 - 60
61 - 80
81 - 100
101 - 120
121 - 140
141 - 160
Matterport3D Top 160 Classes
(ranked by number of instances in training set)

Comparison


Fully supervised
Ours
mAcc (%)
0
10
20
30
40
50
60
64.5
42.9
33.2
33.3
15.6
21.9
13.2
24.2
4.5
19.4
7.8
16.0
8.6
19.7
1.7
16.3
Rarest Classes
1 - 20
21 - 40
41 - 60
61 - 80
81 - 100
101 - 120
121 - 140
141 - 160
Matterport3D Top 160 Classes
(ranked by number of instances in training set)

Ablation

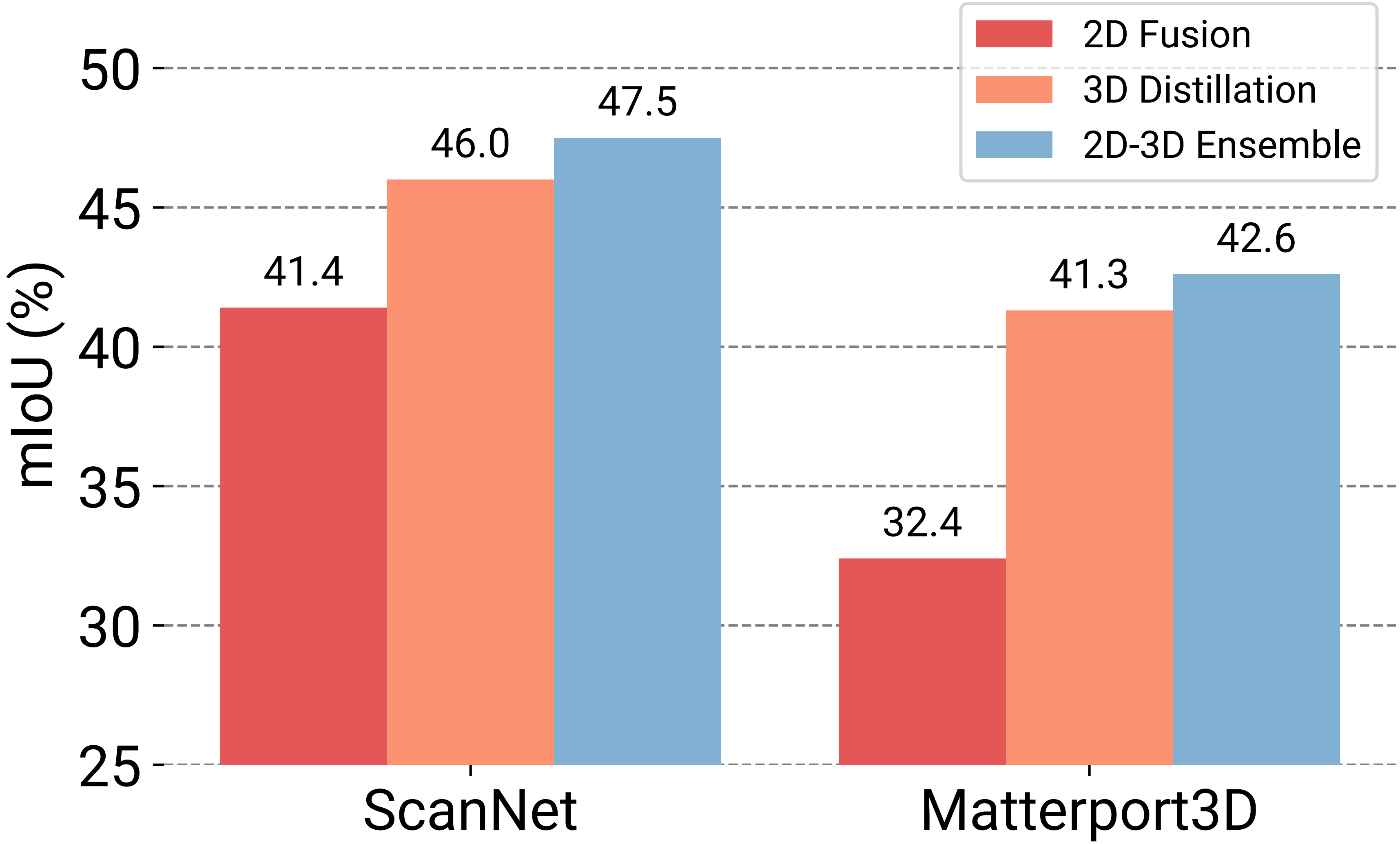

2D Fusion
3D Distillation
2D-3D Ensemble
50
45
40
35
30
25
mIoU (%)
41.4
46.0
47.5
32.4
41.3
42.6
ScanNet
Matterport3D

Image-based 3D Scene Query

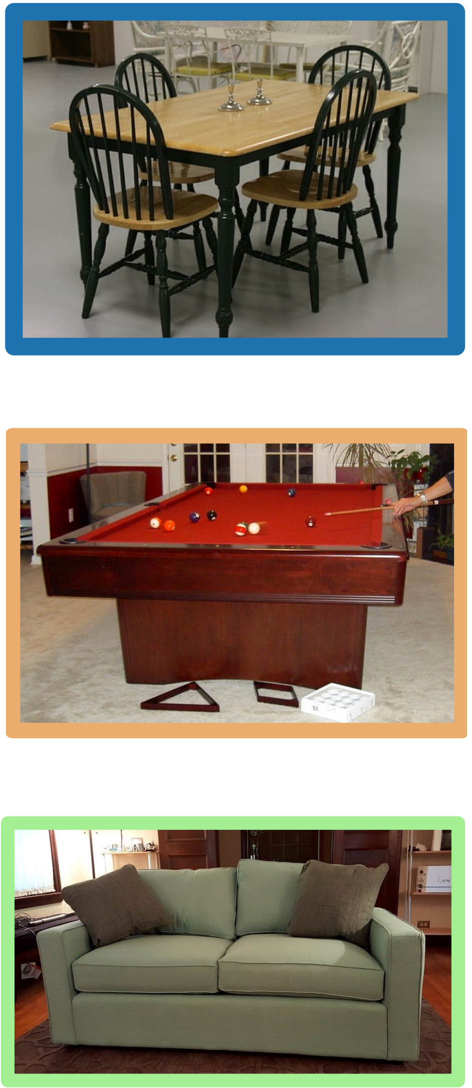


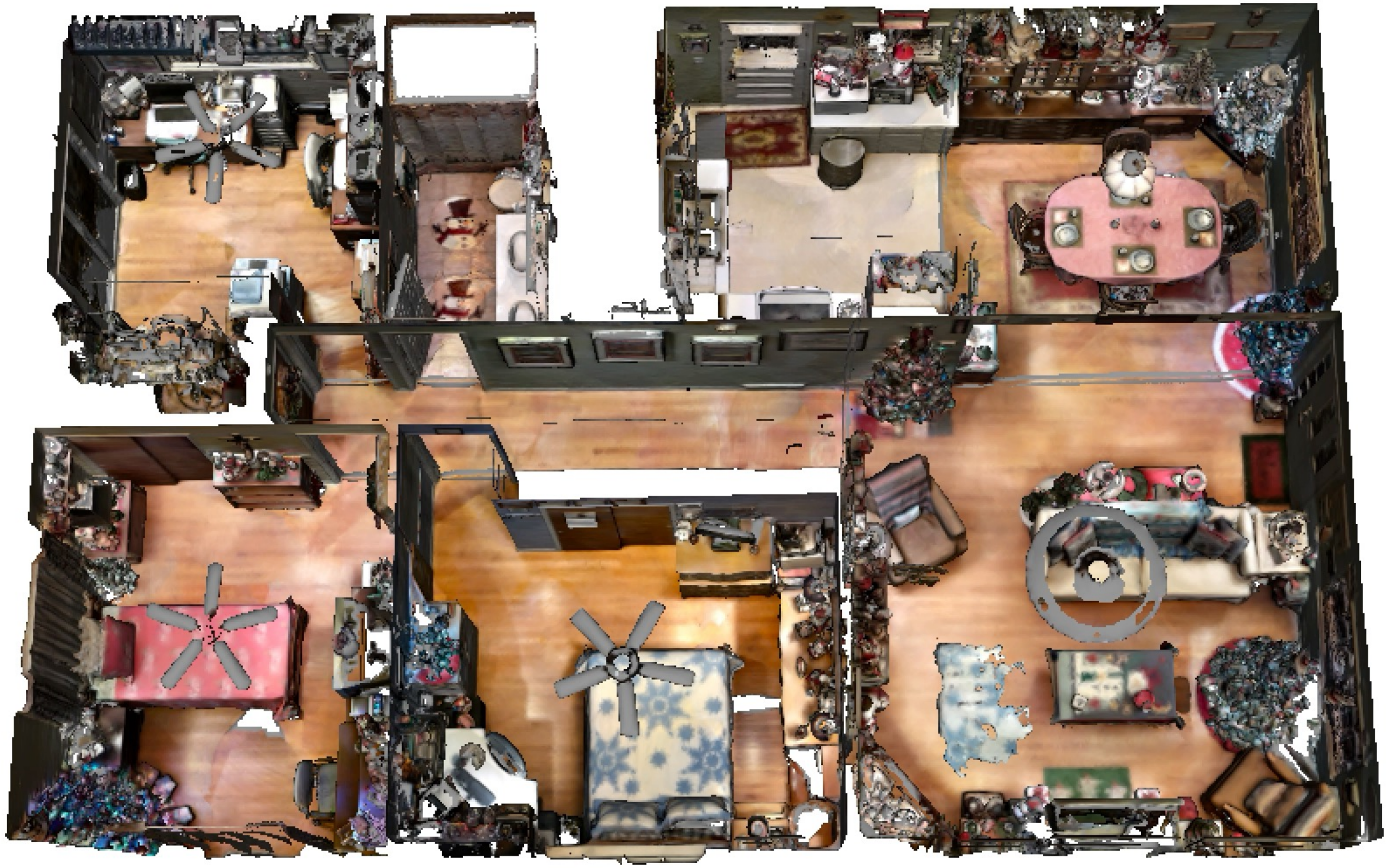
Image Queries

Given 3D Geometry

Interactive Demo

Open-vocabulary 3D Scene Exploration

Text queries:



Take-home Message

- We enable a **wide range of applications** by open-vocabulary queries
- This can hopefully influence how people train 3D scene understanding systems in the future
- Our real-time demo already shows the **possibility to directly apply to AR/VR**