ETH zürich

NEURAL INFORMATION PROCESSING SYSTEMS

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

# **Shape As Points**

# A Differentiable Poisson Solver

**Songyou Peng**

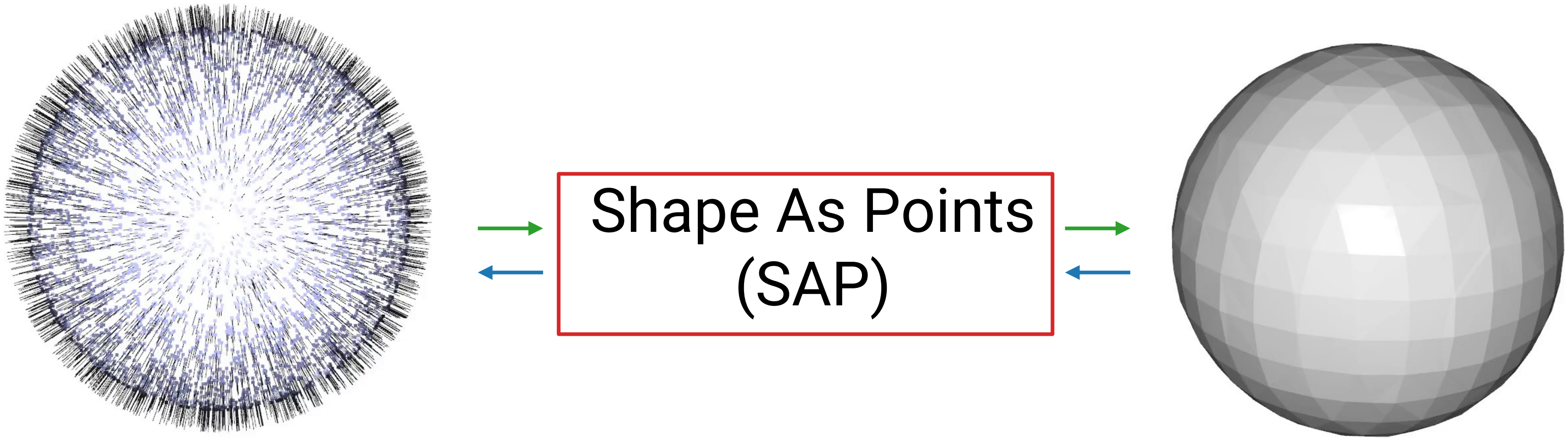
Chiyu “Max” Jiang

Yiyi Liao

Michael Niemeyer

Marc Pollefeys

Andreas Geiger

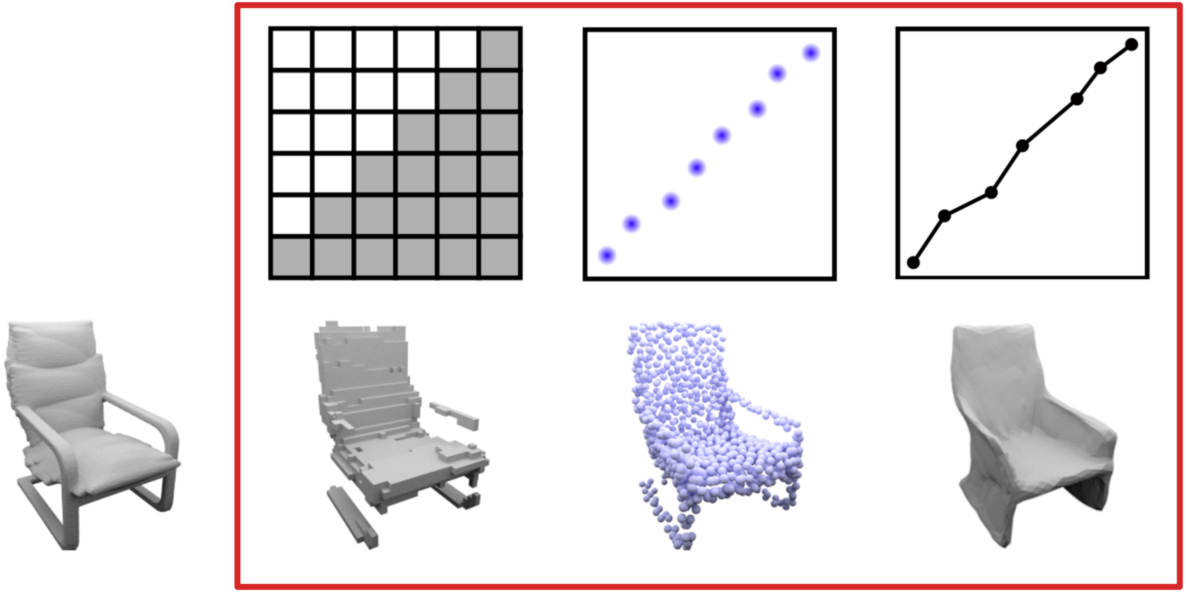


Shape As Points (SAP)

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

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

# 3D Shape Representations



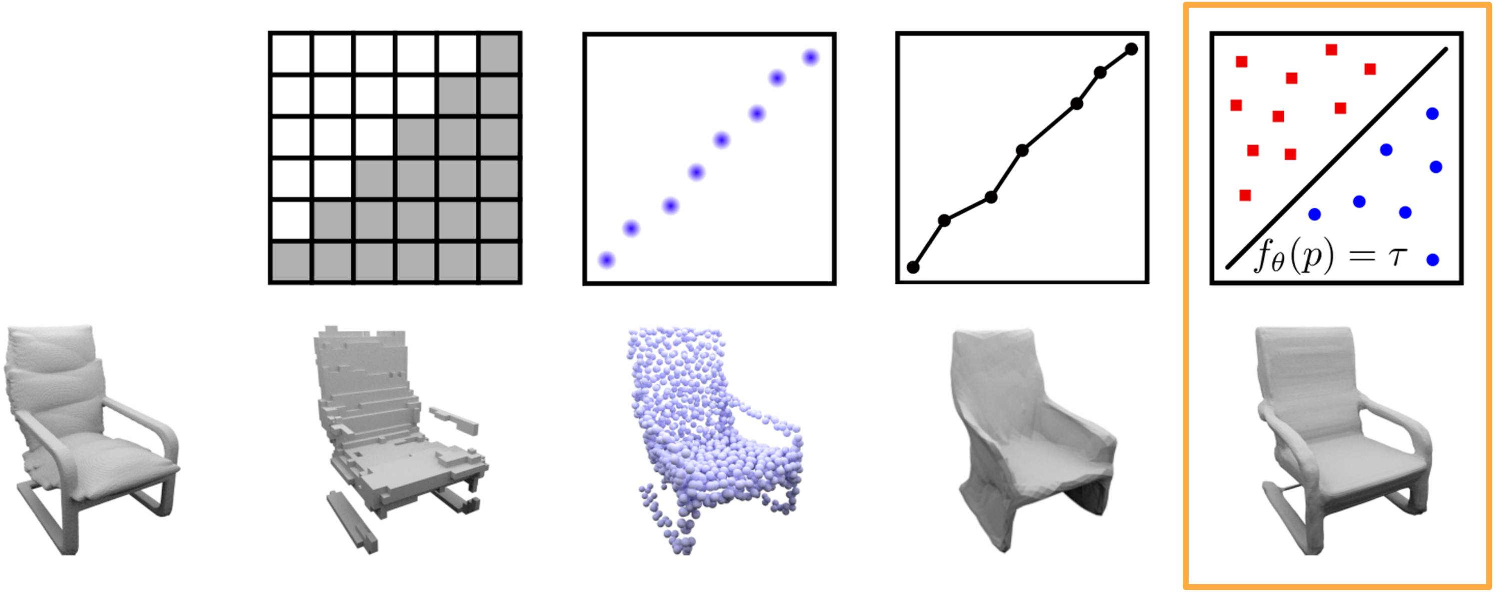
**Traditional Explicit Representations**

➕ Fast inference

➖ Discrete

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

# 3D Shape Representations

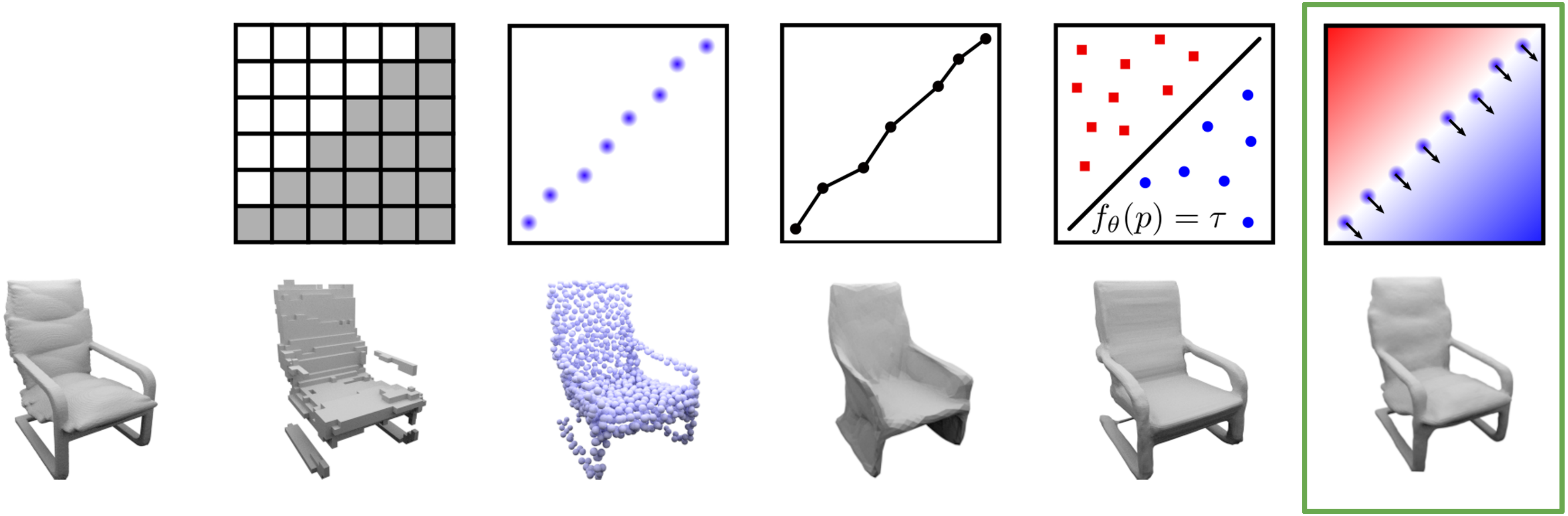


**Neural Implicit Representations**

- + 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**

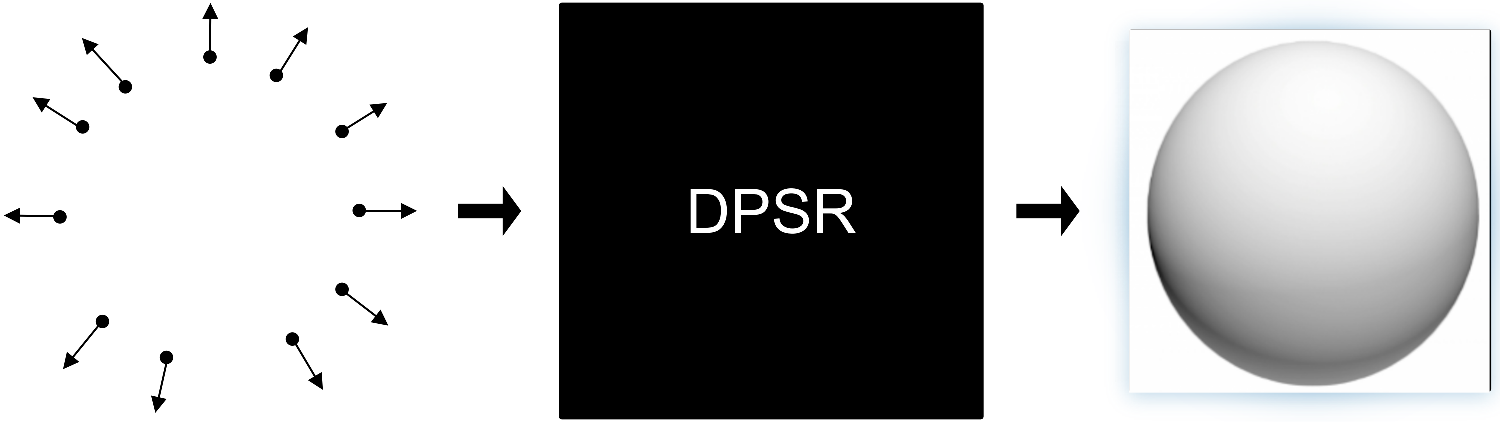
✚ Discrete ⇒ Continuous

✚ Fast inference

✚ Easy initialization

# Method

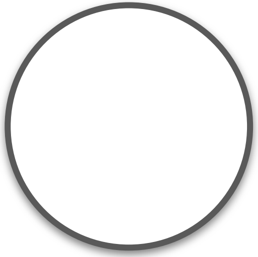
# Differentiable Poisson Solver



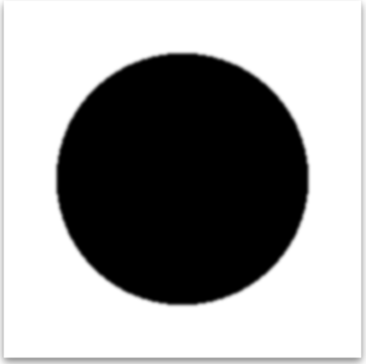
DPSR

# Intuition of Poisson Equation

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

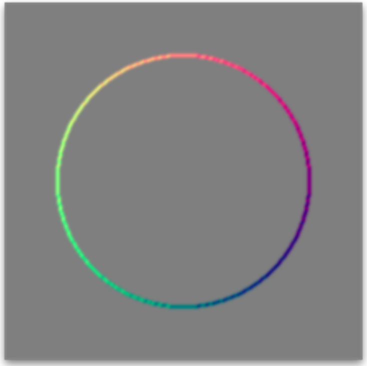


Shape



$\chi$

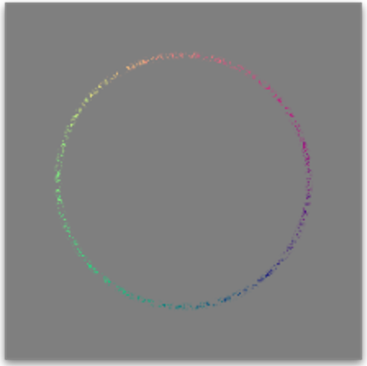
Indicator Function



$\nabla\chi$

Gradient

$\approx$



$\mathbf{v}$

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 code**

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

# Surface Reconstruction from Unoriented Point Clouds

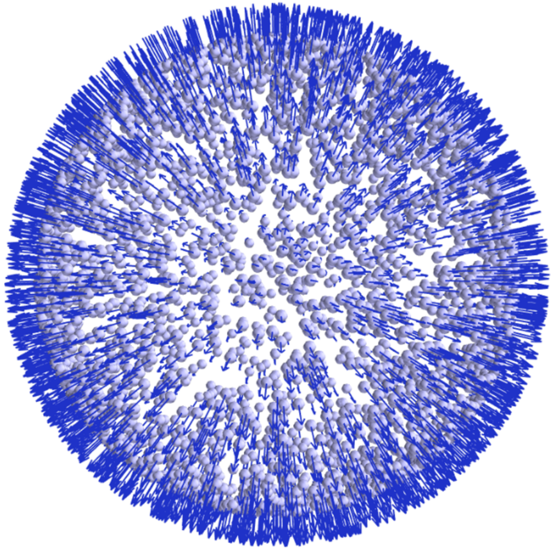
1. SAP for **Optimization-based** 3D Reconstruction
2. SAP for **Learning-based** 3D Reconstruction

# SAP for Optimization-based 3D Reconstruction

# Pipeline - Forward Pass

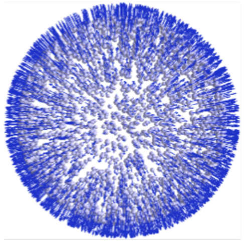
**Input an initial oriented point cloud**

(noisy / incomplete observations)

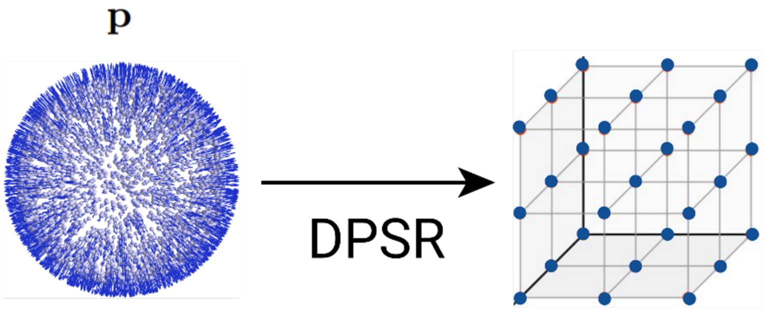


# Pipeline - Forward Pass

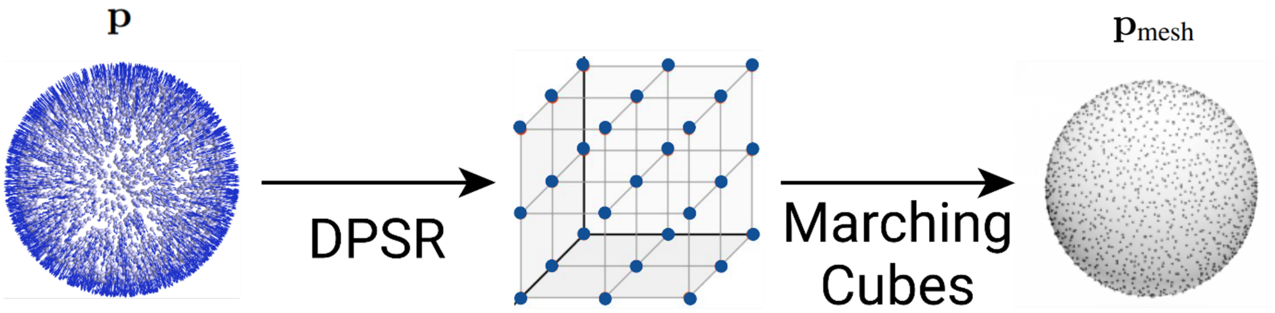
$\mathbf{p}$



# Pipeline - Forward Pass



# Pipeline - Forward Pass



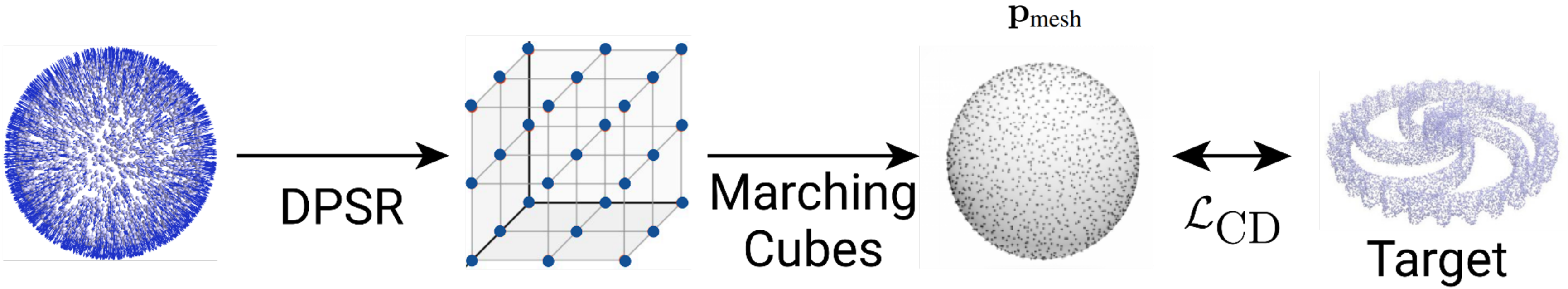
$\mathbf{p}$

DPSR

Marching Cubes

$\mathbf{p}_{\text{mesh}}$

# Pipeline - Forward Pass



$\mathbf{p}_{\text{mesh}}$

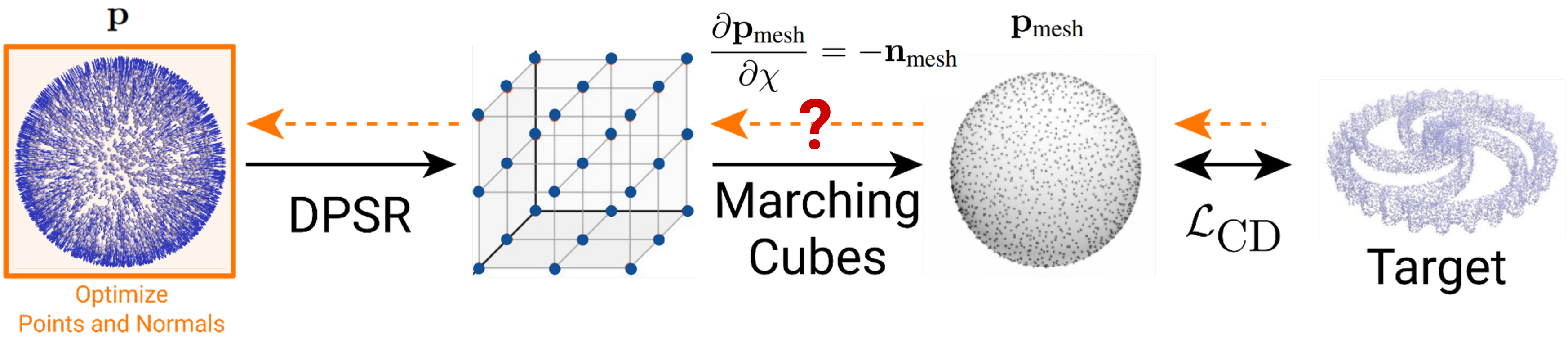
DPSR

Marching Cubes

$\mathcal{L}_{\text{CD}}$

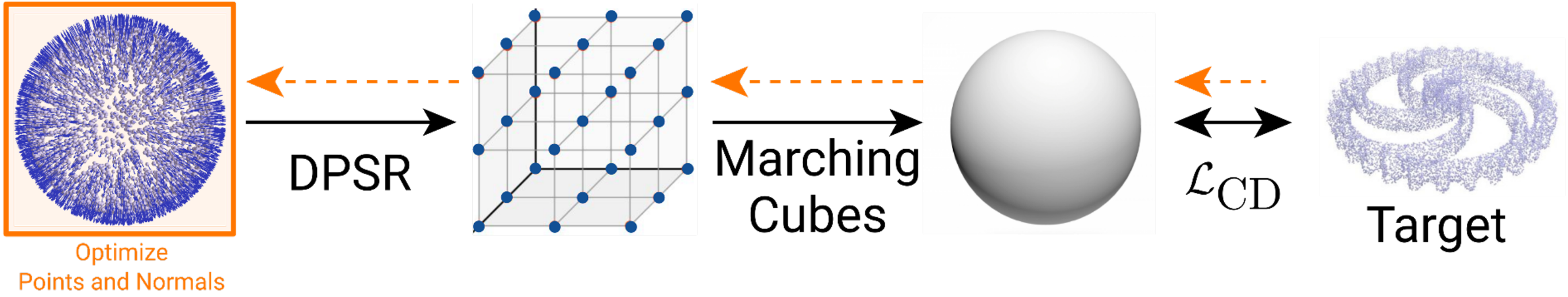
Target

# Pipeline - Backward Pass

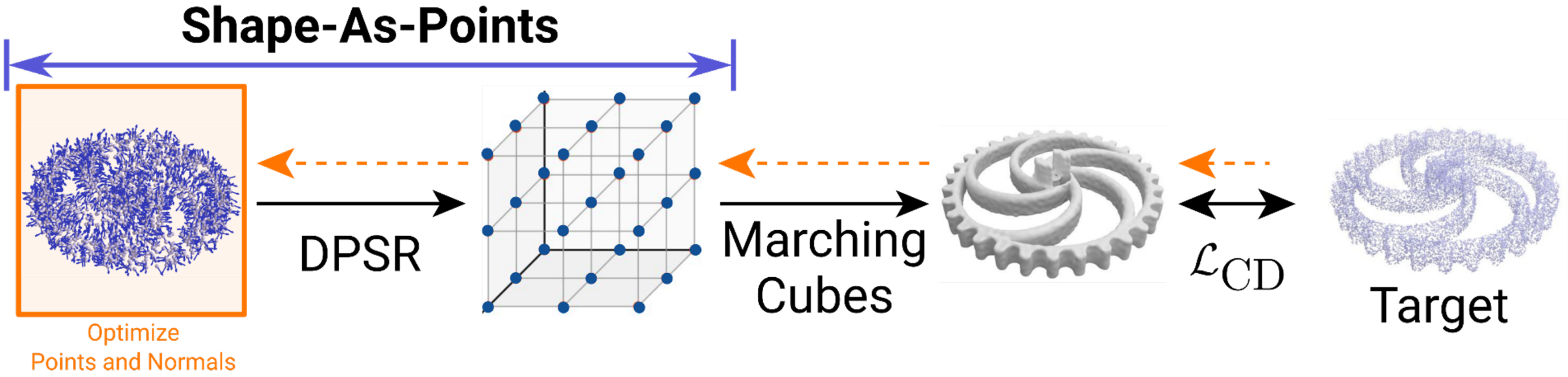

$\mathbf{p}$
Optimize
Points and Normals
DPSR
$\frac{\partial \mathbf{p}_{\text{mesh}}}{\partial \chi} = -\mathbf{n}_{\text{mesh}}$
?
Marching
Cubes
$\mathbf{p}_{\text{mesh}}$
$\mathcal{L}_{\text{CD}}$
Target

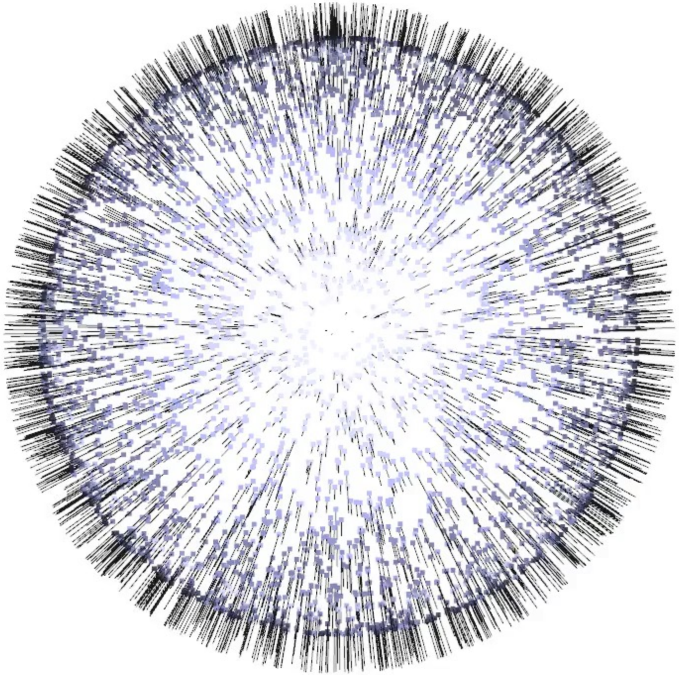
$$\frac{\partial \mathcal{L}_{\text{CD}}}{\partial \mathbf{p}} = \frac{\partial \mathcal{L}_{\text{CD}}}{\partial \mathbf{p}_{\text{mesh}}} \overset{?}{\frac{\partial \mathbf{p}_{\text{mesh}}}{\partial \chi}} \frac{\partial \chi}{\partial \mathbf{p}}$$

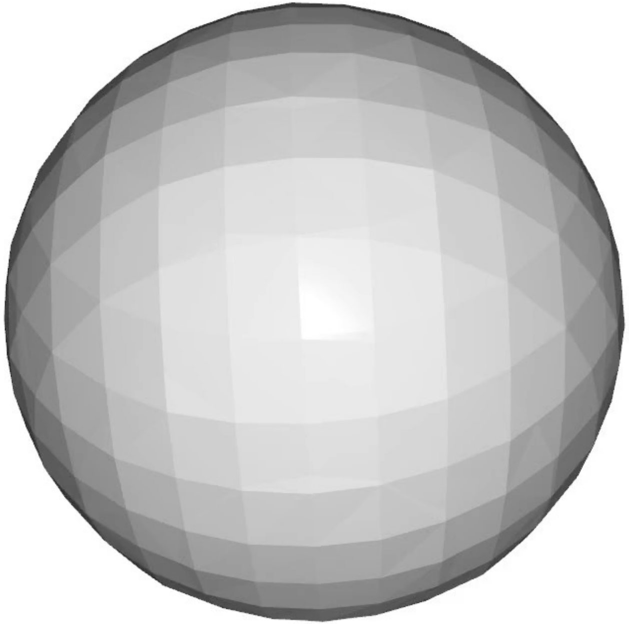
# Pipeline


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

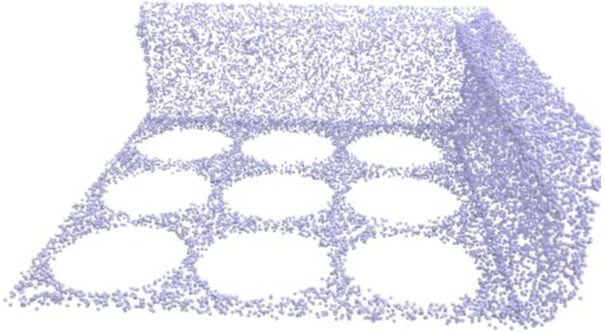
# Pipeline


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

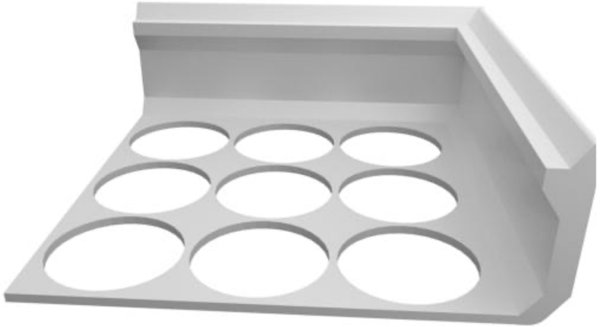




# Comparison

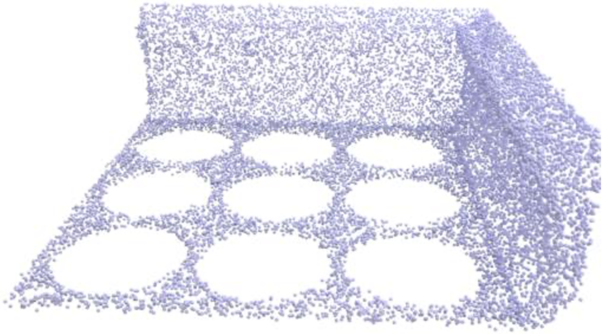


Unoriented Point Clouds



GT Mesh

# Comparison



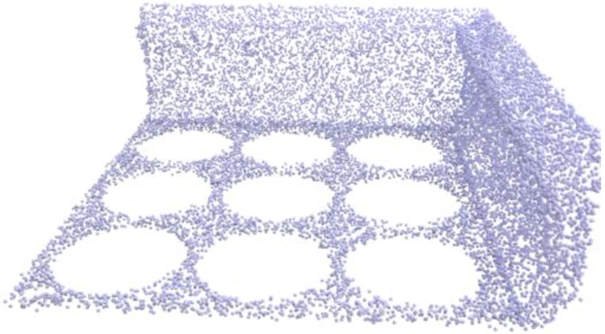
Unoriented Point Clouds

Point2Mesh

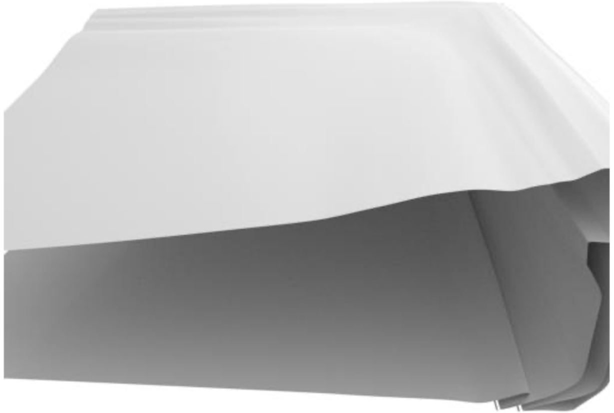
**Runtime**: 62 mins

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

# Comparison



Unoriented Point Clouds

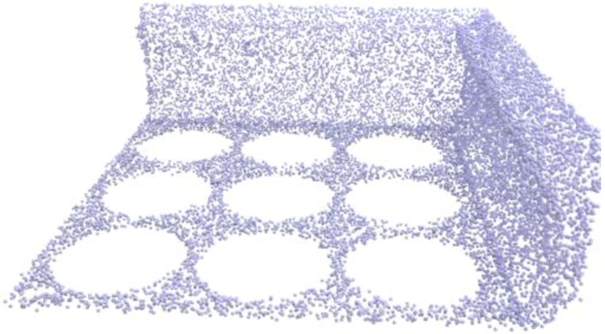


IGR

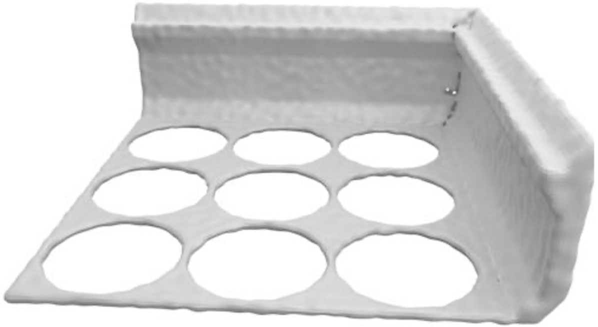
**Runtime**: 30 mins

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

# Comparison



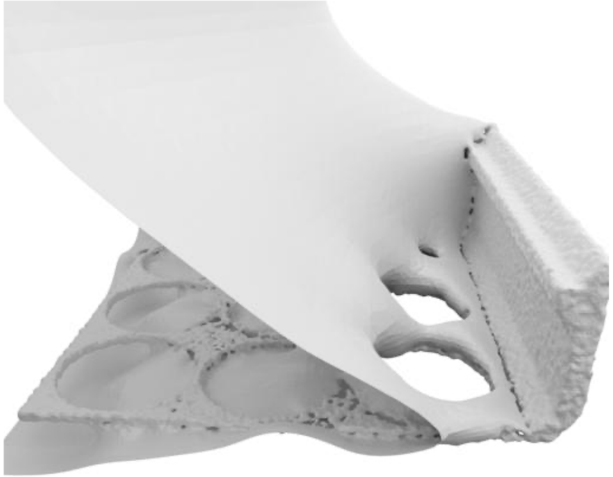
Unoriented Point Clouds



**SAP**

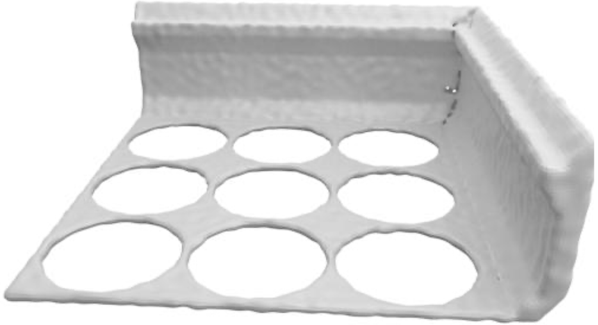
**Runtime**: ~6 mins

# Comparison



SPSR

**Runtime**: ~9 sec



**SAP**

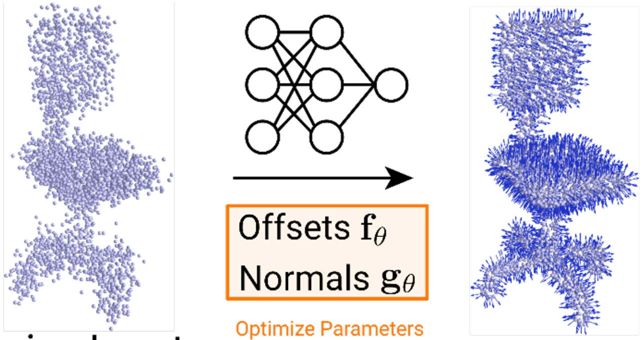
**Runtime**: ~6 mins

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



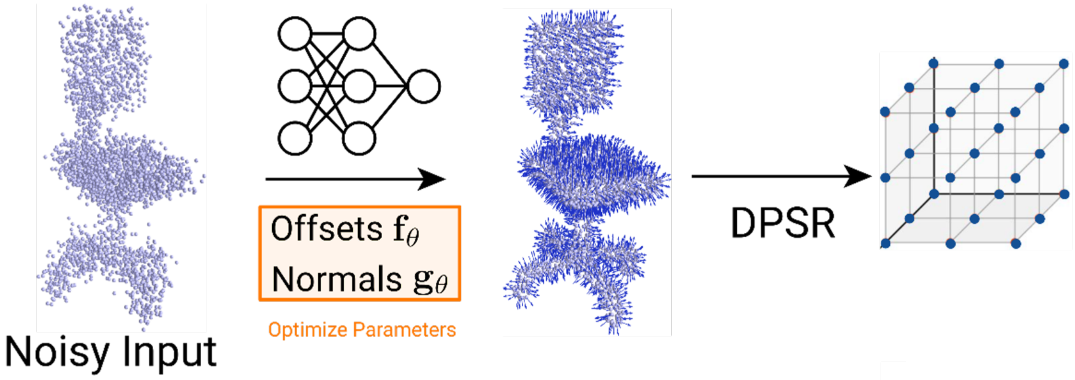
Noisy Input

Offsets $\mathbf{f}_\theta$

Normals $\mathbf{g}_\theta$

Optimize Parameters

# Learning-based Pipeline



Noisy Input

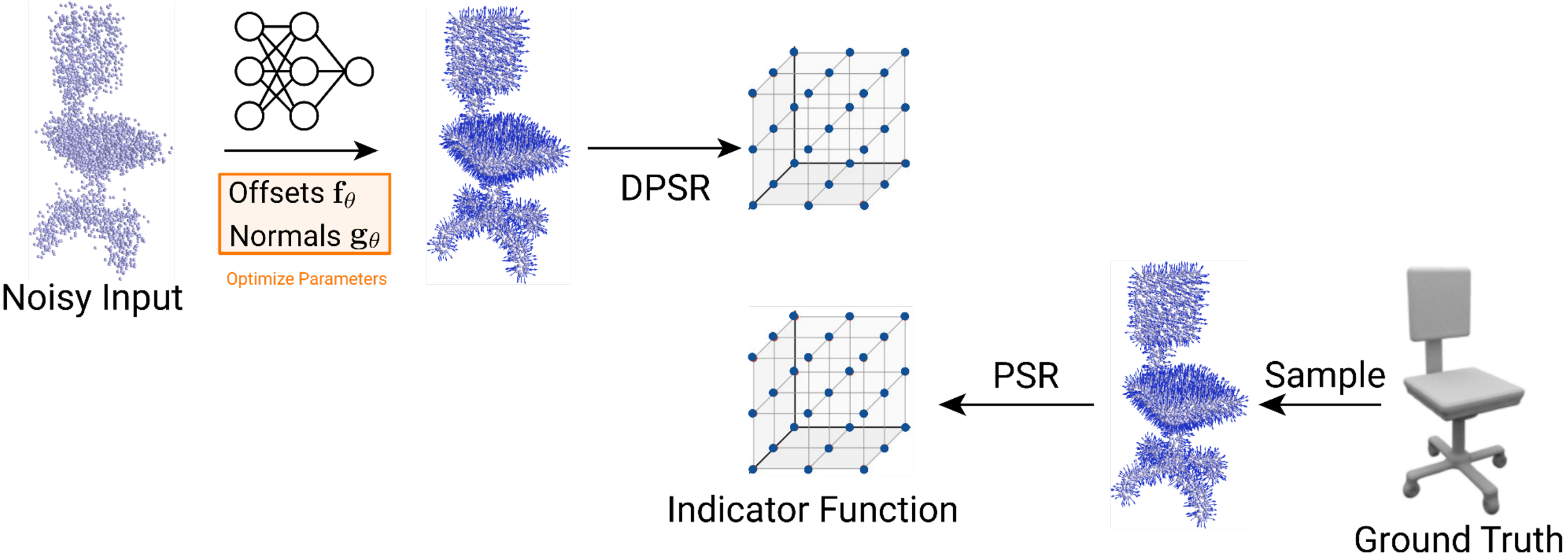
Offsets $f_\theta$

Normals $g_\theta$

Optimize Parameters

DPSR

# Learning-based Pipeline



Noisy Input

Offsets $\mathbf{f}_\theta$

Normals $\mathbf{g}_\theta$

Optimize Parameters

DPSR

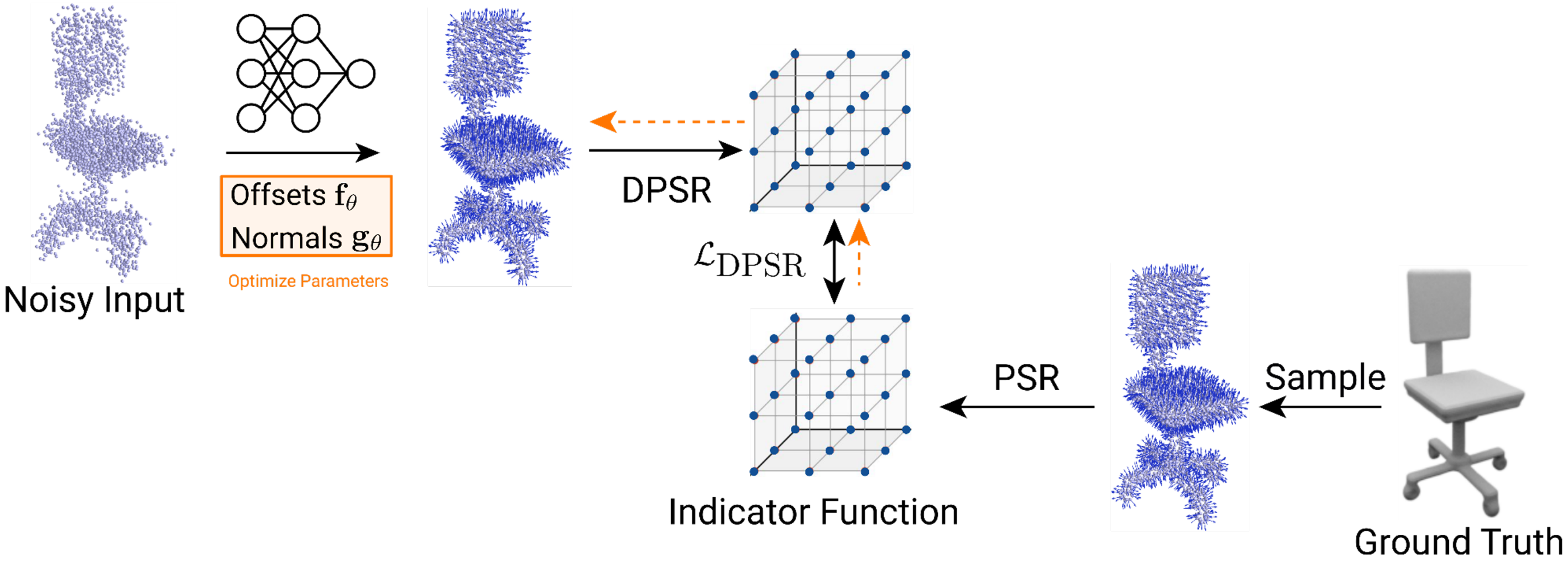
Indicator Function

PSR

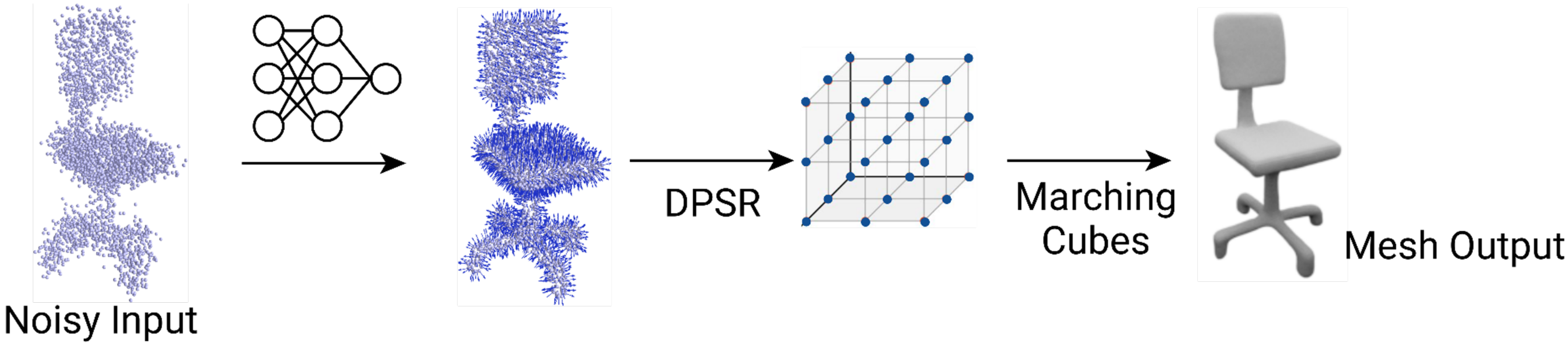
Sample

Ground Truth

# Learning-based Pipeline


Noisy Input
Offsets $\mathbf{f}_\theta$
Normals $\mathbf{g}_\theta$
Optimize Parameters
DPSR
$\mathcal{L}_{\text{DPSR}}$
Indicator Function
PSR
Sample
Ground Truth

# Learning-based Pipeline



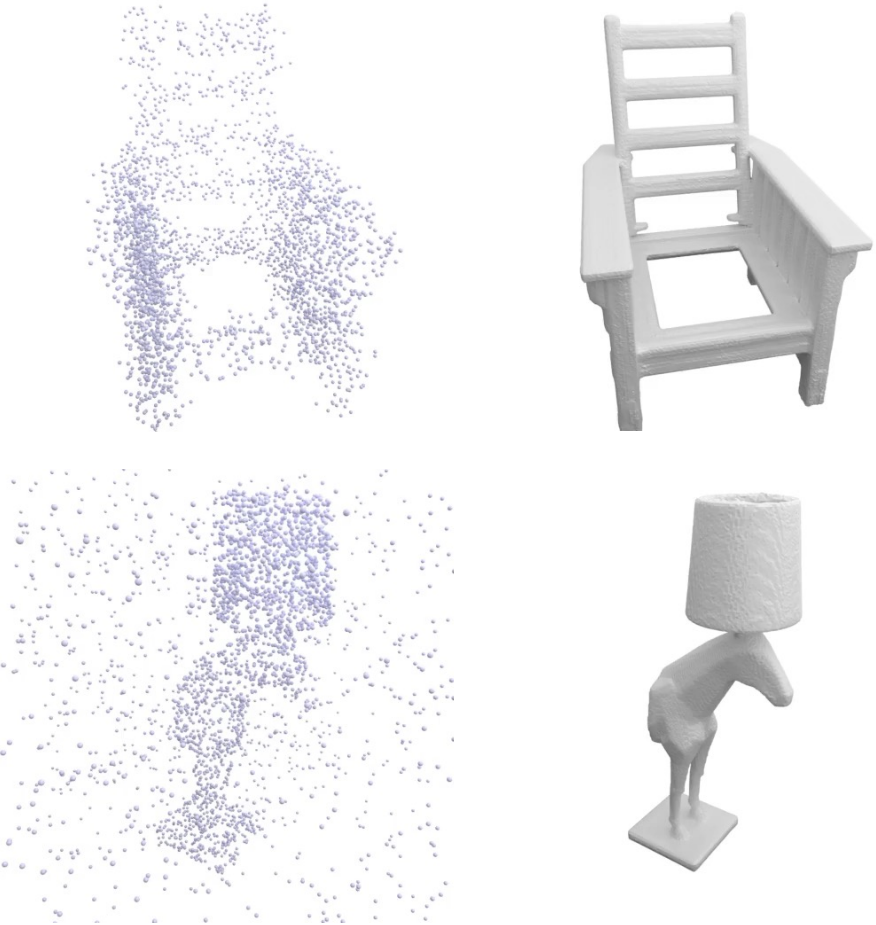
Noisy Input

DPSR

Marching Cubes

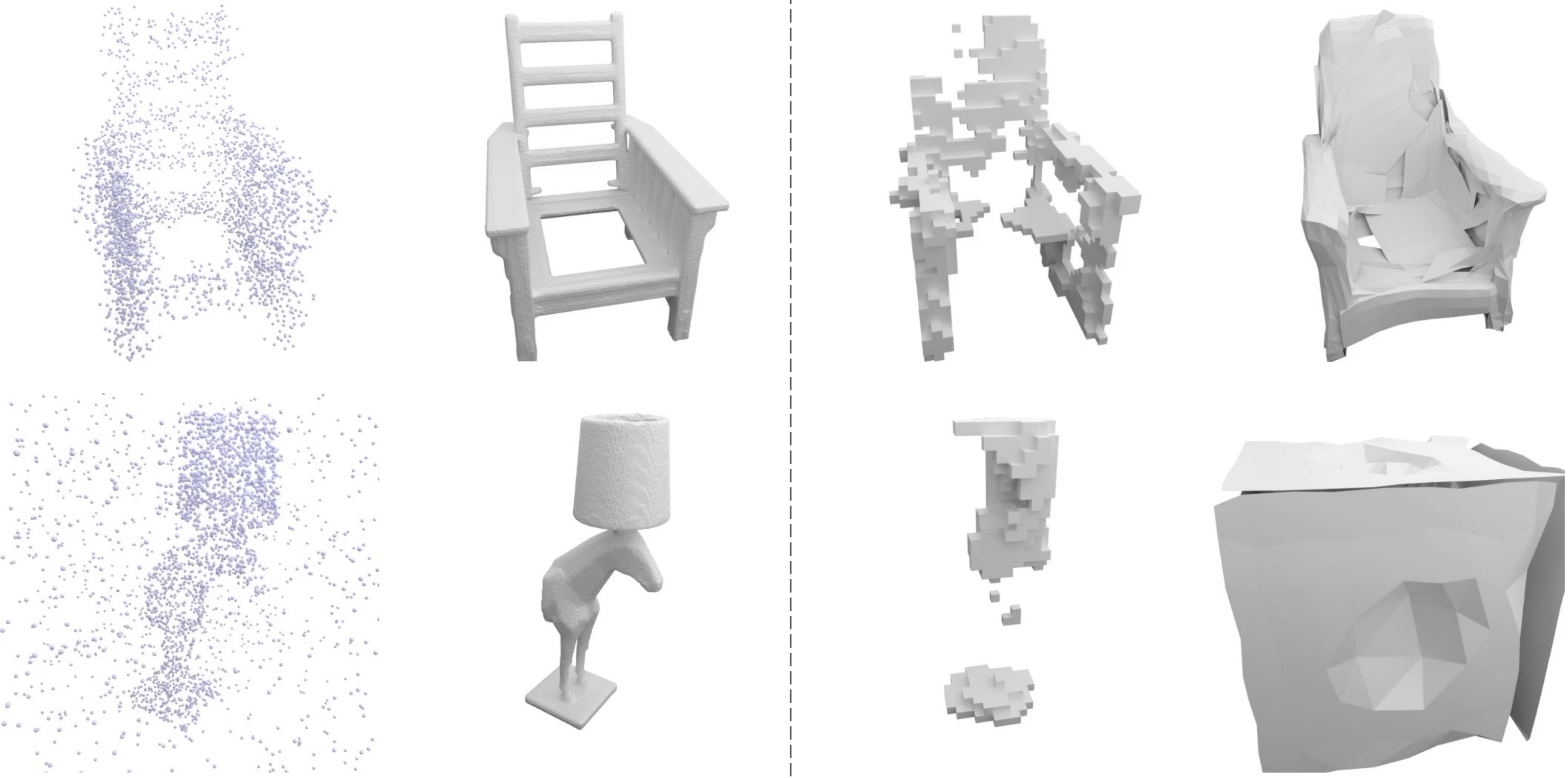
Mesh Output

# Results



Inputs

GT Mesh



Inputs

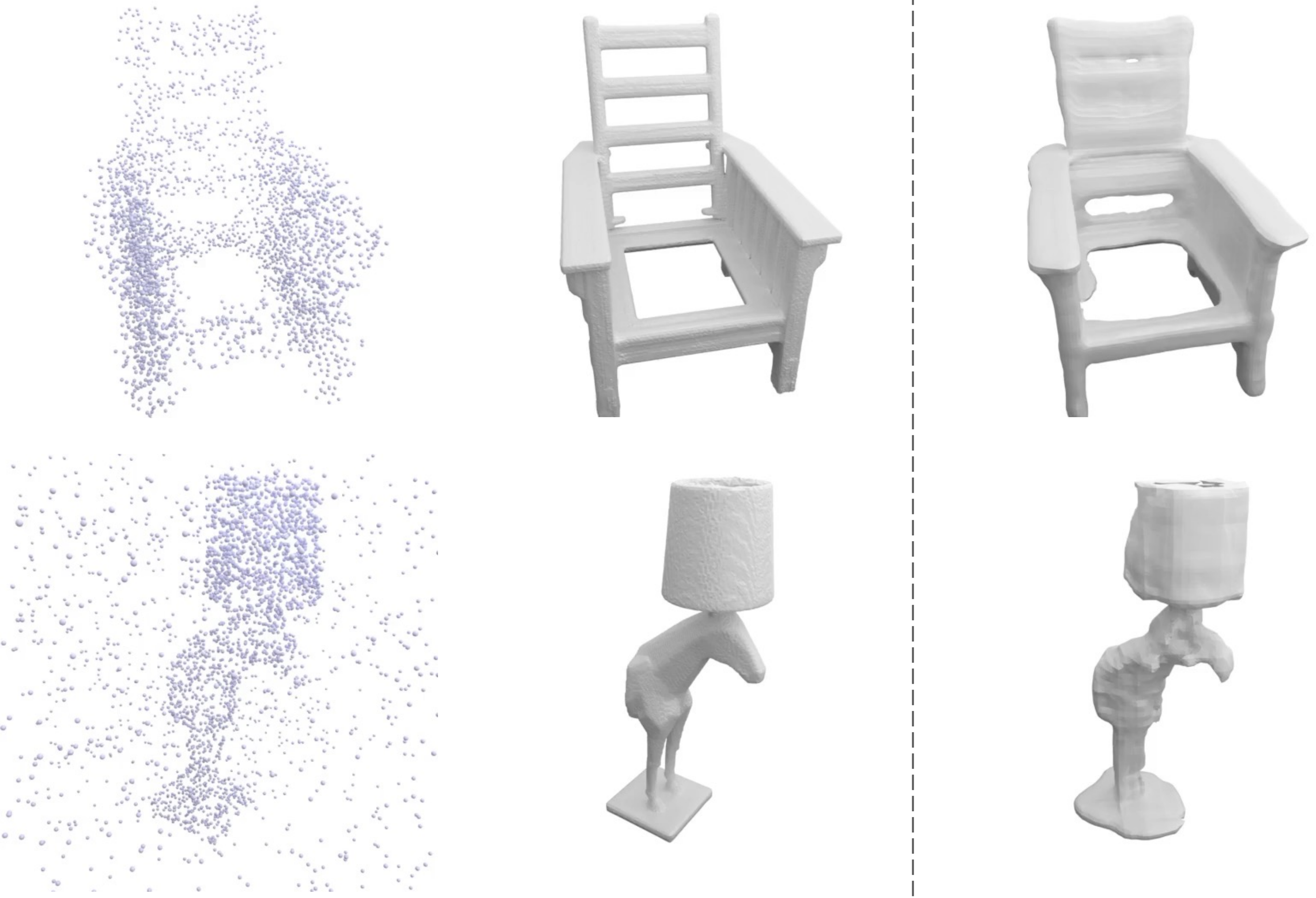
GT Mesh

R2N2

15 ms

AtlasNet

25 ms

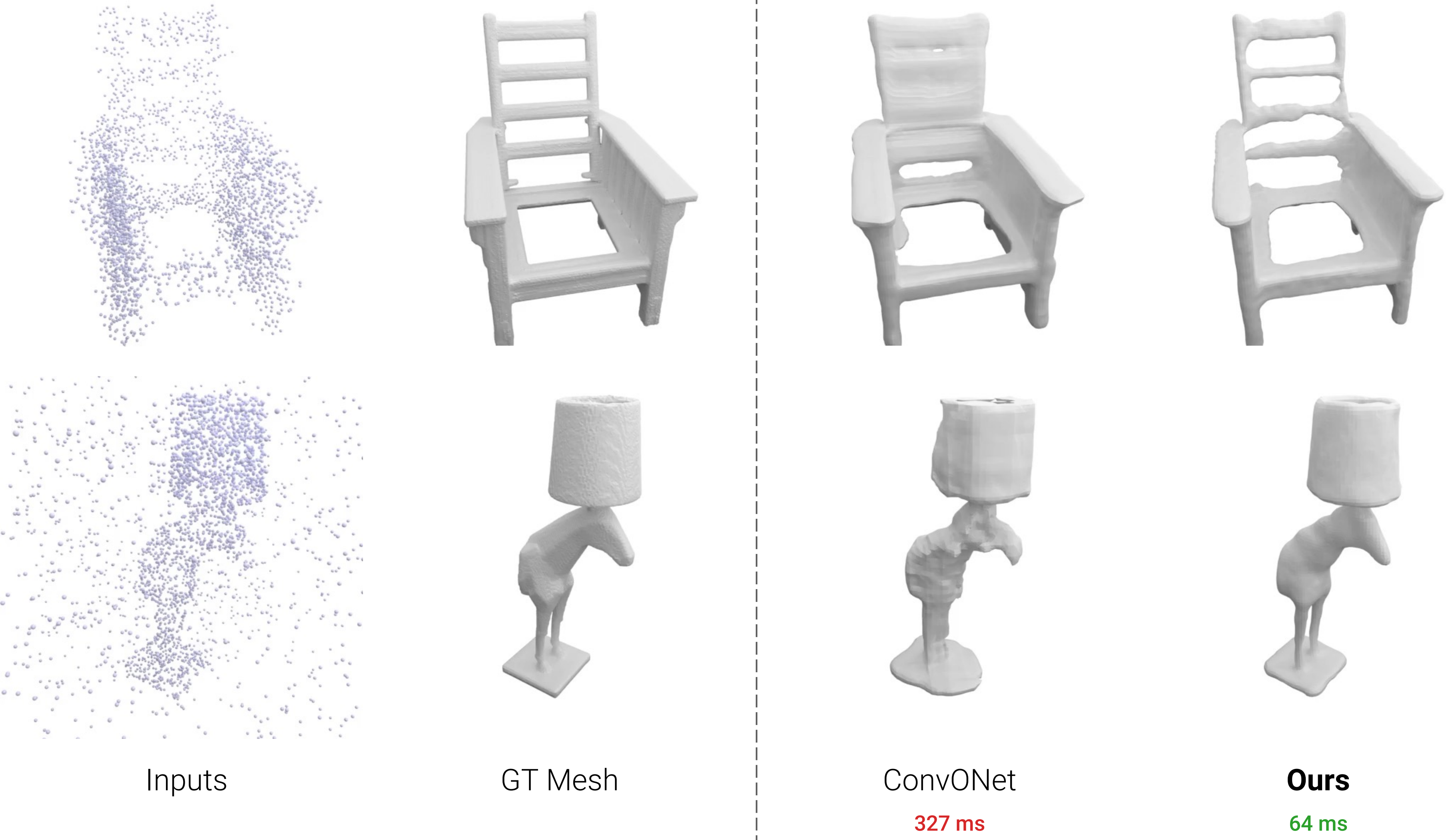


Inputs

GT Mesh

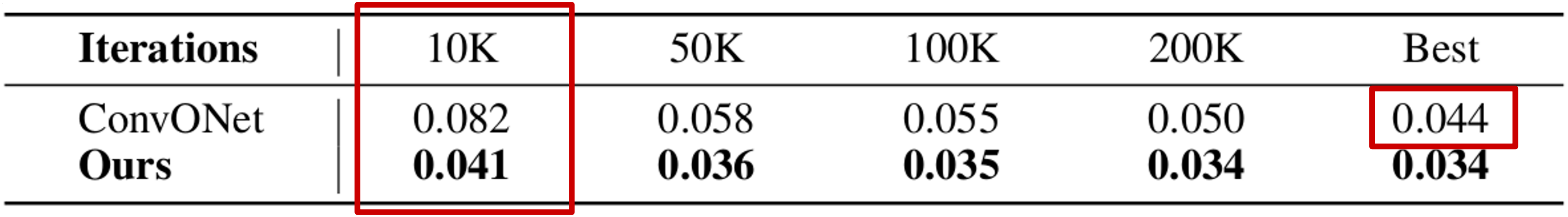
ConvONet

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

- 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

# Thank You!

https://pengsongyou.github.io/sap