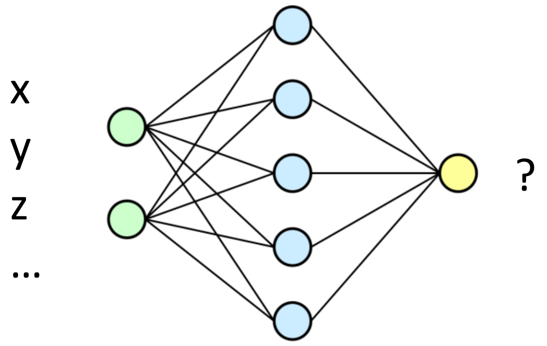
Coordinate Networks

Federico Stella

Coordinate Networks

Spatial coordinates are given as input to a neural network



Neural Radiance Fields (NeRF)

Implicit Surface Representations

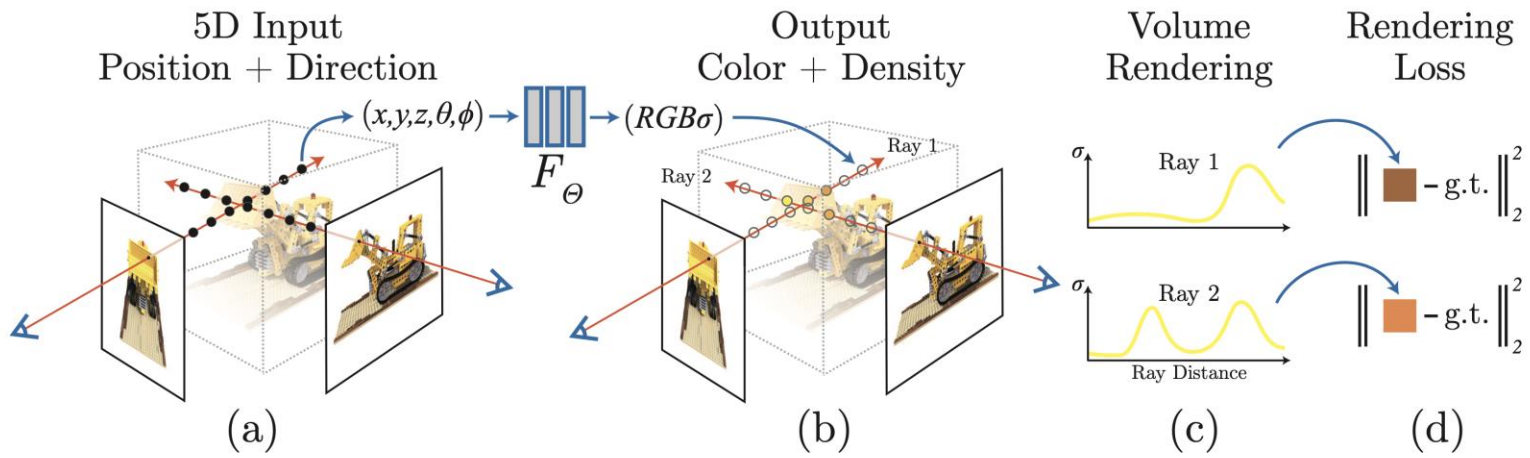
Image Compression

Neural Radiance Fields (NeRF)

Neural Radiance Fields (NeRF)

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

Rendering a radiance field



The volume density $\sigma(\mathbf{x})$ can be interpreted as the differential probability of a ray terminating at an infinitesimal particle at location $\mathbf{x}$. The expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \quad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

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

Positional Encoding

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \cdots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right) \right)$$

https://www.youtube.com/watch?v=JuH79E8rdKc

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

Implicit Surface Representations

3D geometry

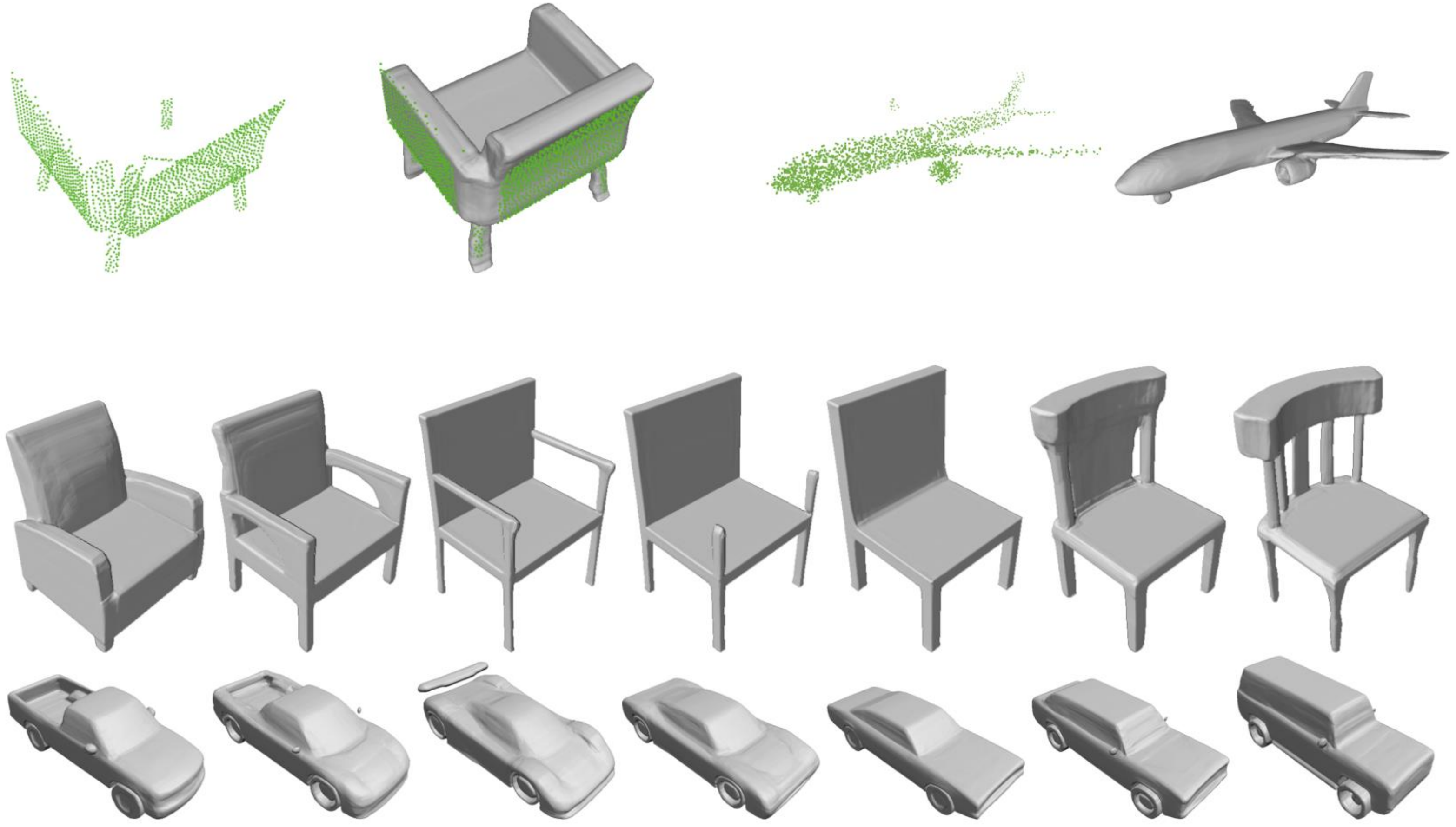
Shape reconstruction and completion from partial or noisy data

Shape optimization

Classification

Segmentation

Rendering



Figures from Park J. J. et al. “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”

How to represent shapes?

Multiple methods exist, with advantages and disadvantages

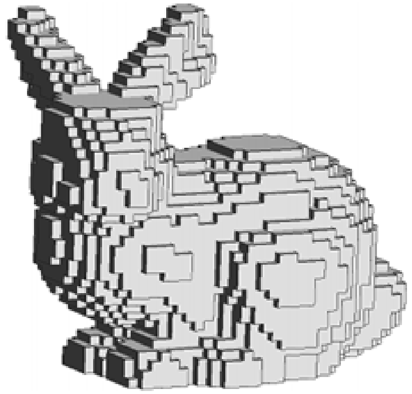
A good model should:

- Encode geometrical information (connectivity, details, etc.)
- Allow different topologies
- Not be limited in resolution
- Be as lightweight as possible
- Be sufficiently easy to further process for applications

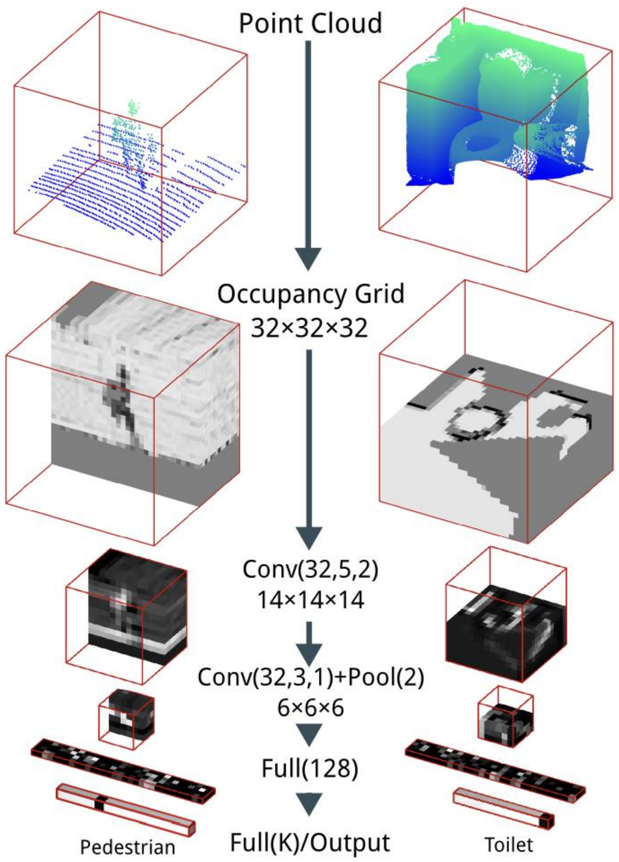
Recently, **implicit functions parametrized with neural networks**
The surface is represented as a level-set of its implicit function

Voxel-based

Regular grid with occupancy information



Example: VoxNet



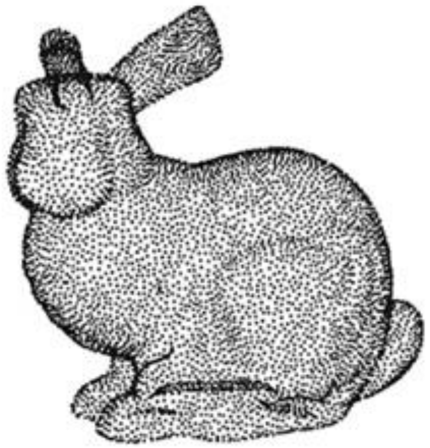
Characteristics:

- CNNs can be easily extended to work on a 3D grid
- High memory footprint
- Higher complexity limiting resolution (20^3- 64^3 typical grid sizes) and results

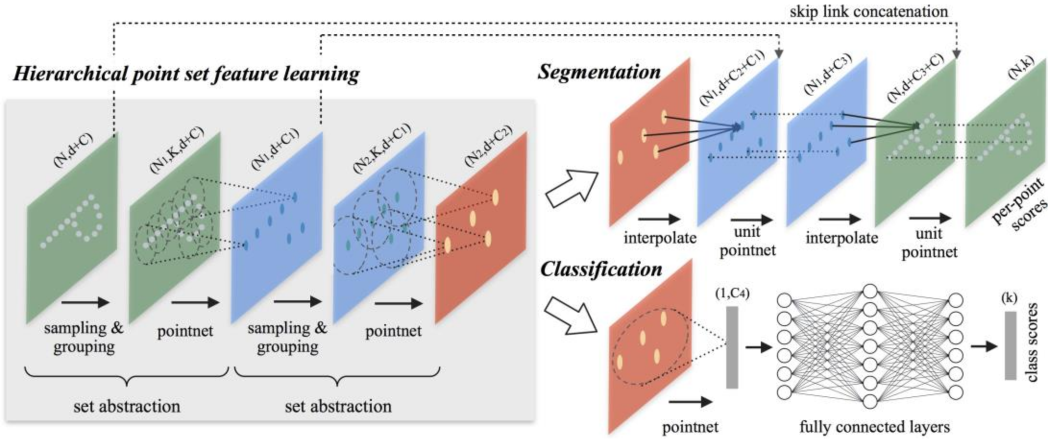
Image from Maturana D. et al. "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition"

Point-based

Unordered set of points in space



Example: PointNet++



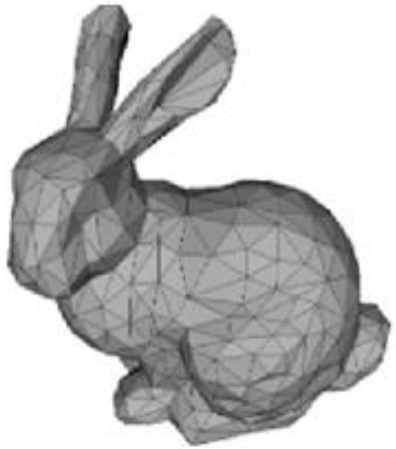
Characteristics:

- Easy to obtain from 3D scanners
- Useful in segmentation
- Memory efficient
- No connectivity information
- Prone to noise and sparsity

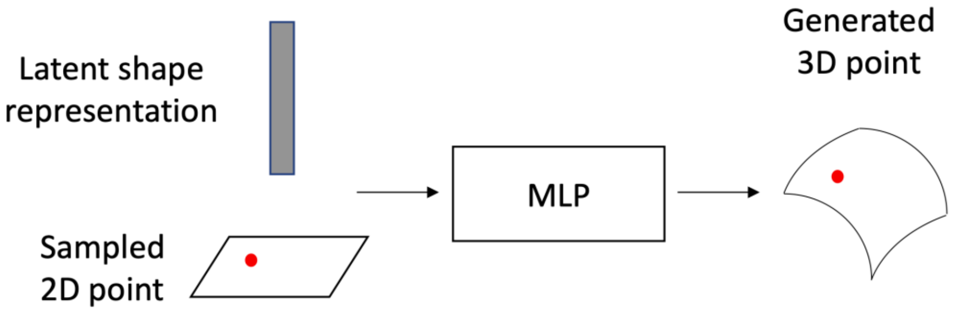
Image from Qi C. R. et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space"

Mesh-based

Set of vertices + set of edges



Example: AtlasNet



Characteristics:

- Explicit 3D surface representation, encoding spatial and connectivity information
- Can be treated as a graph (processed with Graph NN)
- Can be deformed from homeomorphic templates/patches
- Not straightforward to process
- Can represent a single topology

Image from Groueix T. et al. "AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation"

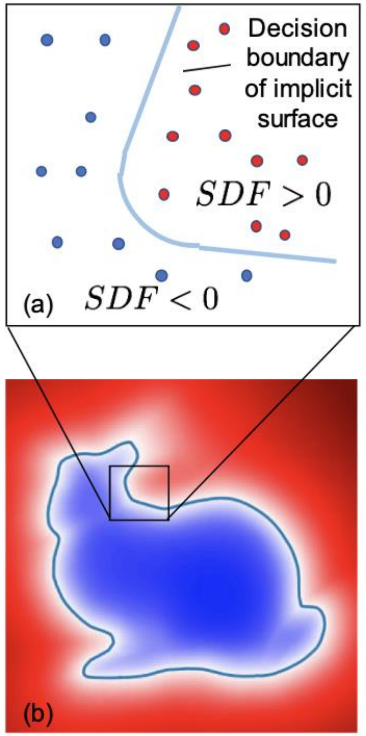
Implicit representations

Occupancy Fields

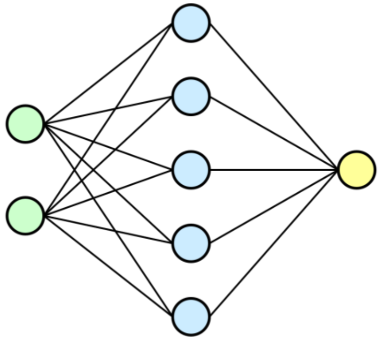
- 1 inside the object
- 0 outside the object

Signed Distance Fields:

- Distance from the closest surface
- Positive outside of the object
- Negative inside of the object
- Unitary-norm gradient



$$SDF(\mathbf{x}) = \min_{\mathbf{p} \in \mathcal{S}} d(\mathbf{x}, \mathbf{p}) \cdot \begin{cases} +1 \text{ if } \mathbf{x} \text{ is outside of } \mathcal{S} \\ -1 \text{ if } \mathbf{x} \text{ is inside of } \mathcal{S} \end{cases}$$



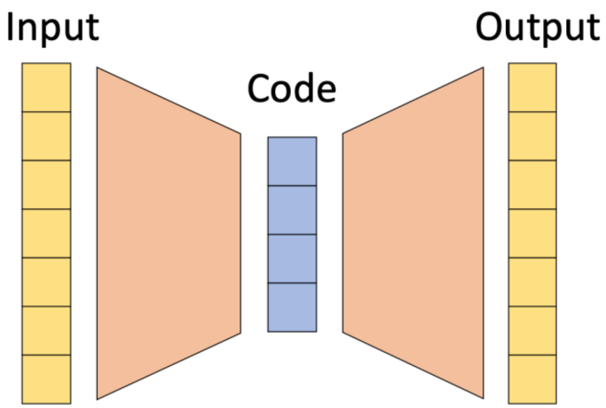
Parametrized with a neural network as a
Neural Implicit Representation
The network itself is the shape representation

Characteristics:

- Typically parametrized with a neural network
- **Continuous**: no resolution limit!
- **Limited memory footprint**
- Can effectively represent detailed models
- **Arbitrary and varying topologies**
- Achieved good experimental results
- Need a meshing algorithm to retrieve an explicit representation

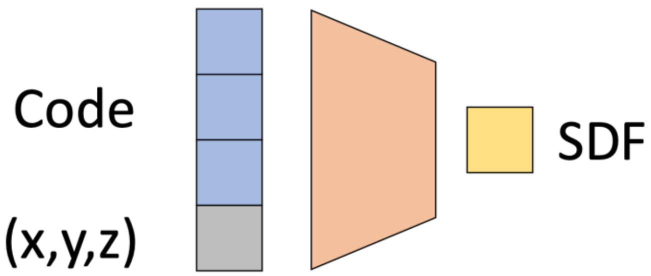
SDF figure from Park J. J. et al. “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”

Auto-decoders for SDF-learning



Auto-encoder

DeepSDF

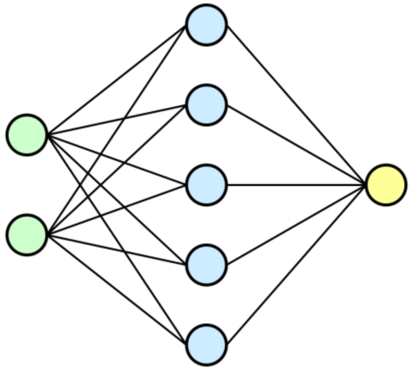


Auto-decoder-based formulation

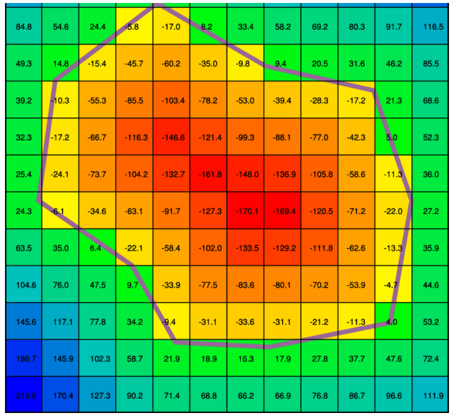
Flat latent code, optimized at inference time

Figures from Park J. J. et al. “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”

Implicit representations

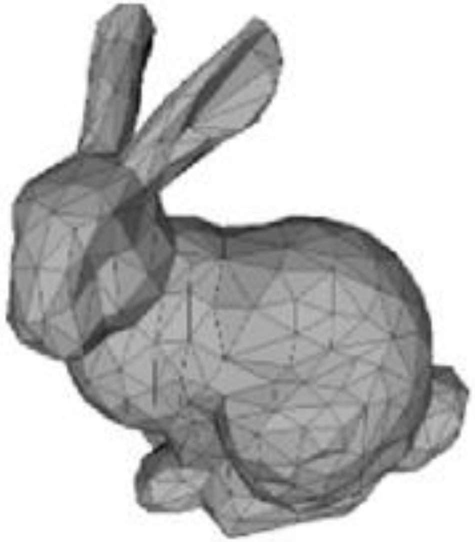


(Neural)
Implicit Representation



3D grid of points

Isosurface extraction algorithms:
Marching Cubes
Extended MC
Dual Contouring
…

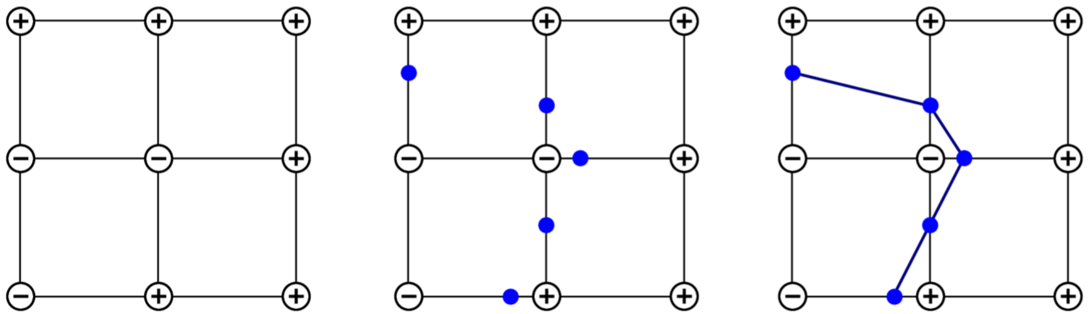


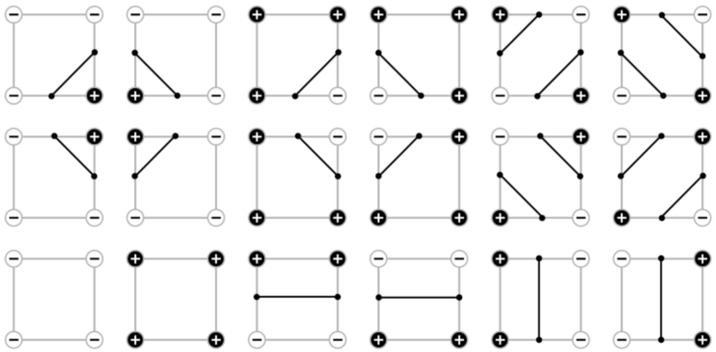
Mesh

Center figure from Zucker M. et al. "An Optimization Approach to Rough Terrain Locomotion"

Marching Cubes vs Dual Contouring

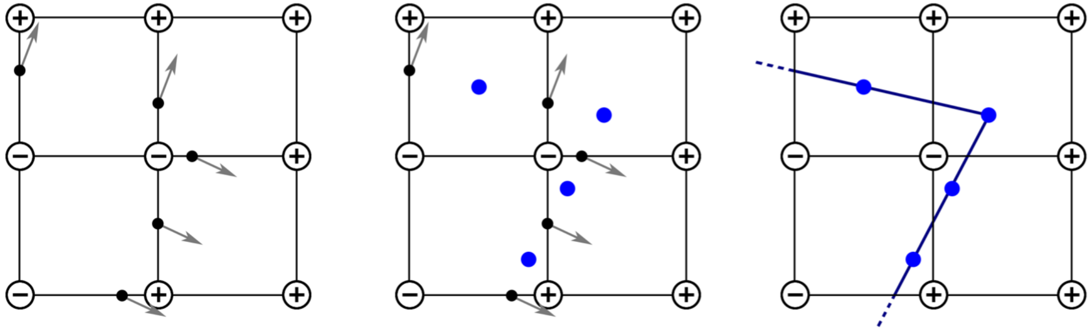
Marching Cubes





Complex but efficient look-up table

Dual Contouring

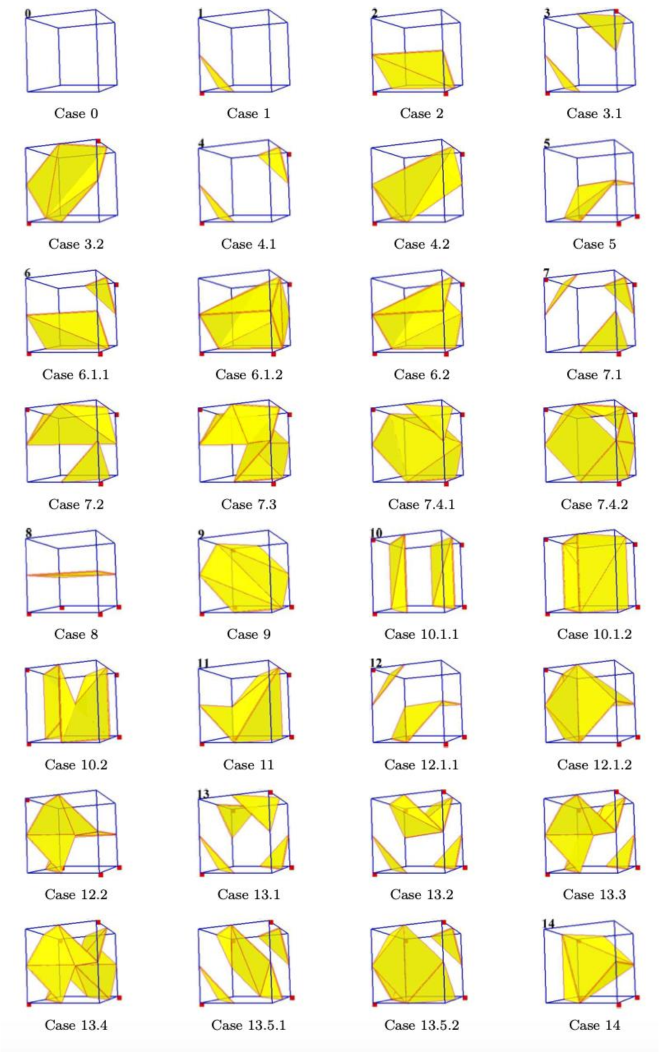


$$\arg\min_{\mathbf{x}} \sum_{e\in\mathcal{E}} \left(\mathcal{N}_e(\mathbf{x}-\mathcal{V}_e)\right)^2$$

- Dual method (one vertex per cell)
- Less vertices and triangles
- Sharp features are preserved
- Needs normals!

Figures from Chen Z. et al. “Neural Dual Contouring”

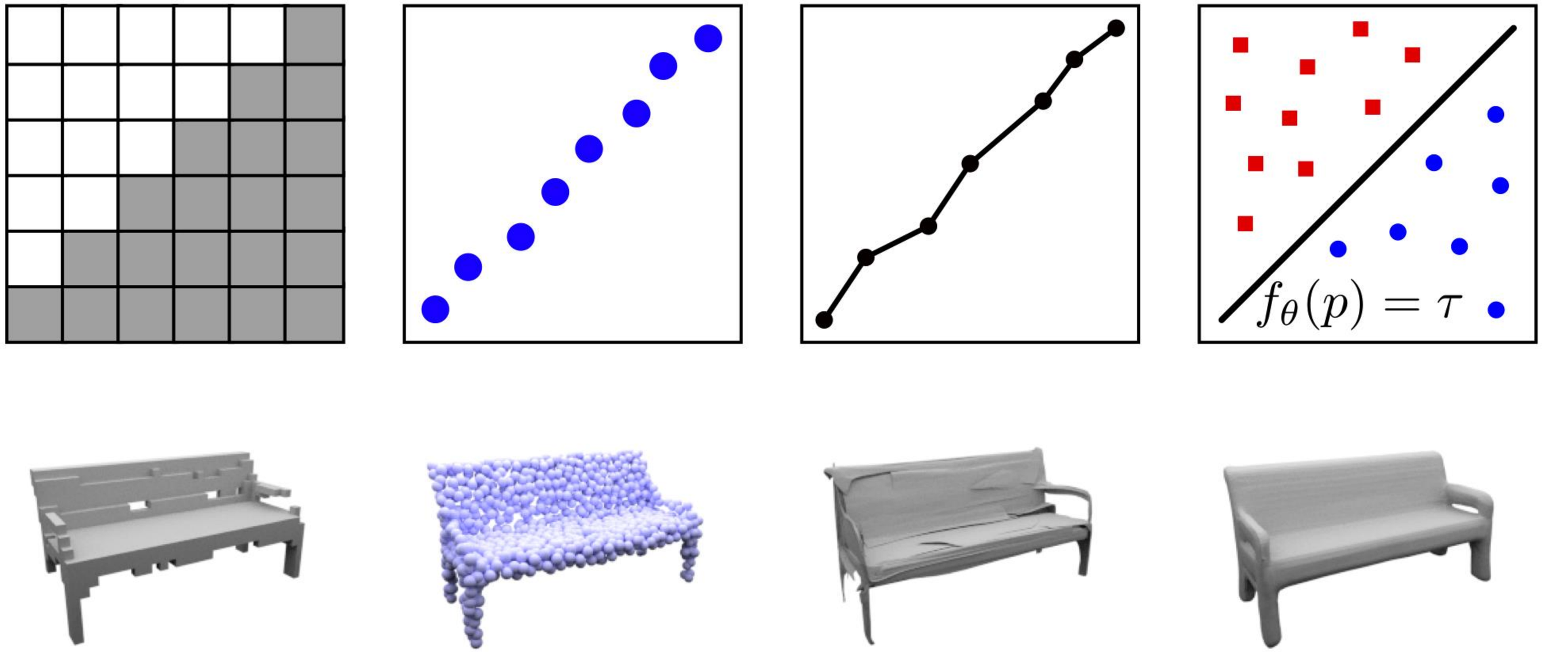
Marching Cubes



- Assumes trilinearity of the field
- Case selection via look-up table
- Vertex positioning via linear interpolation
- Vertices on grid edges
- Possible additional vertices on the faces and inside the cell

MC tessellations from Lewiner T. et al. “Efficient implementation of Marching Cubes’ cases with topological guarantees”

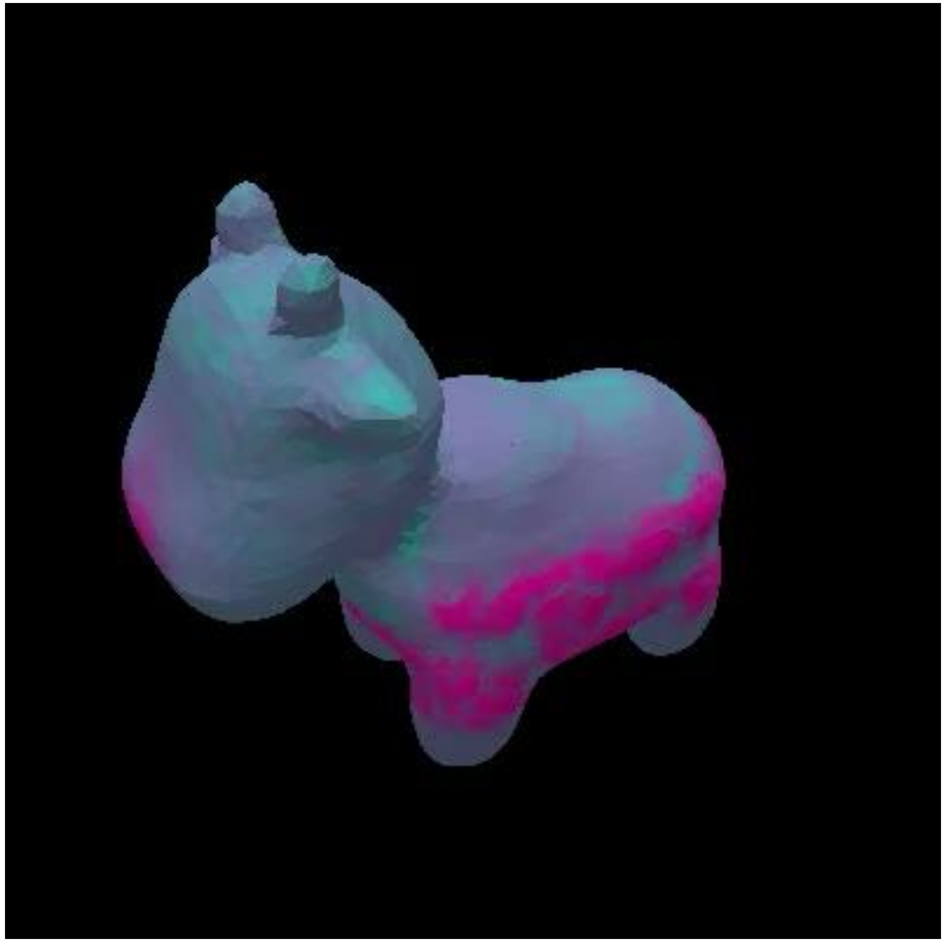
Implicit representations



Voxel | Point cloud | Mesh (AtlasNet) | Implicit (Occupancy)

Image from Mescheder L. et al. “Occupancy Networks: Learning 3D Reconstruction in Function Space”

End-to-End Differentiable Pipeline



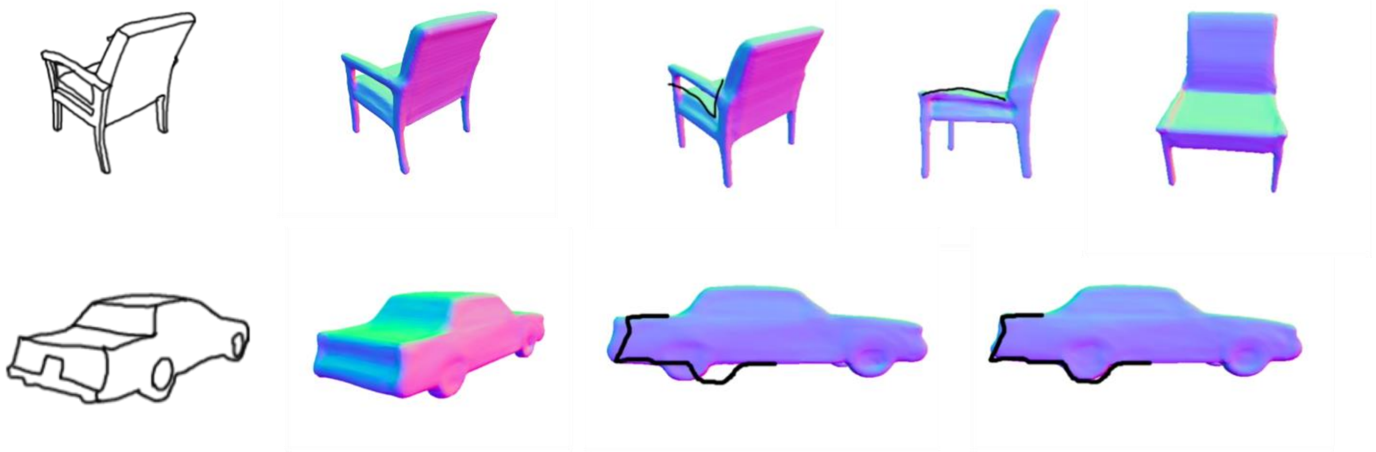
1. Start with a Deep SDF code.
2. Use marching cube to compute vertices and facets.
3. Use them for the forward pass and for backpropagation.
4. Update the SDF code and iterate.

—> We can turn a genus 0 cow into a genus 1 duck by minimizing a differentiable objection function.

[Remelli et al., NeurIPS’20]

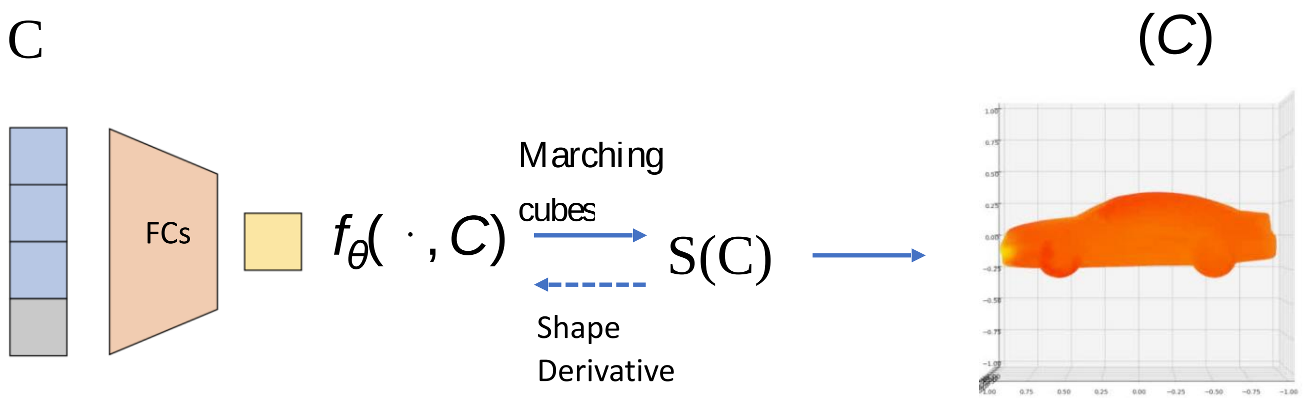
From Silhouettes to 3D Shapes

3D Model from Image



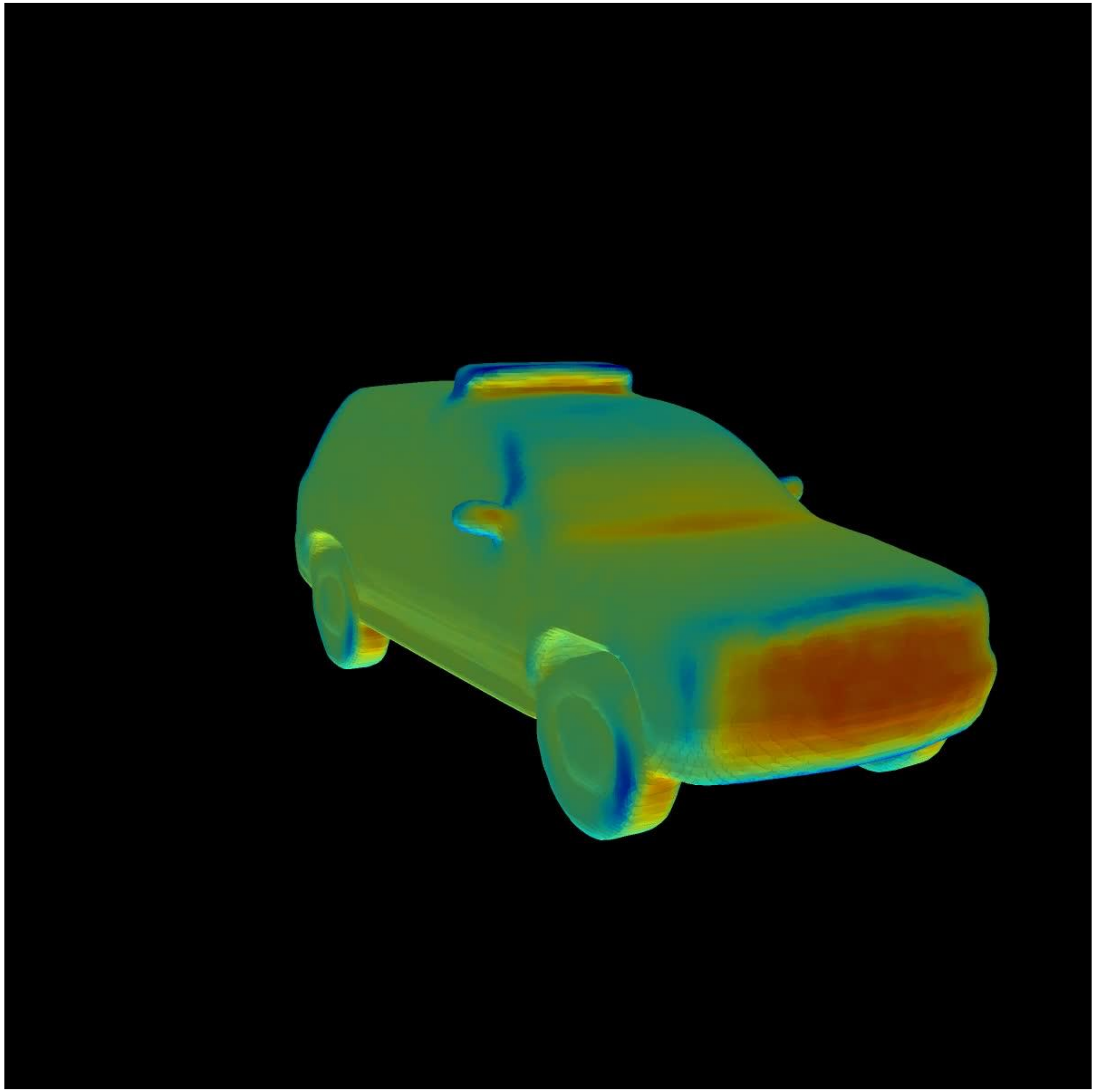
Editable 3D Model from Sketch

Drag Minimization

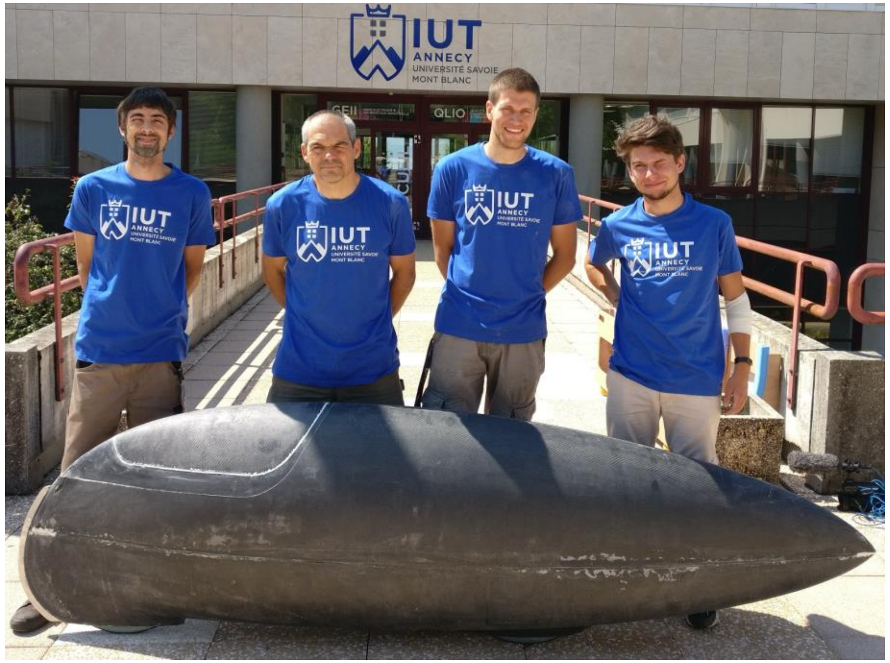


Minimize (C) with respect to C under constraint.

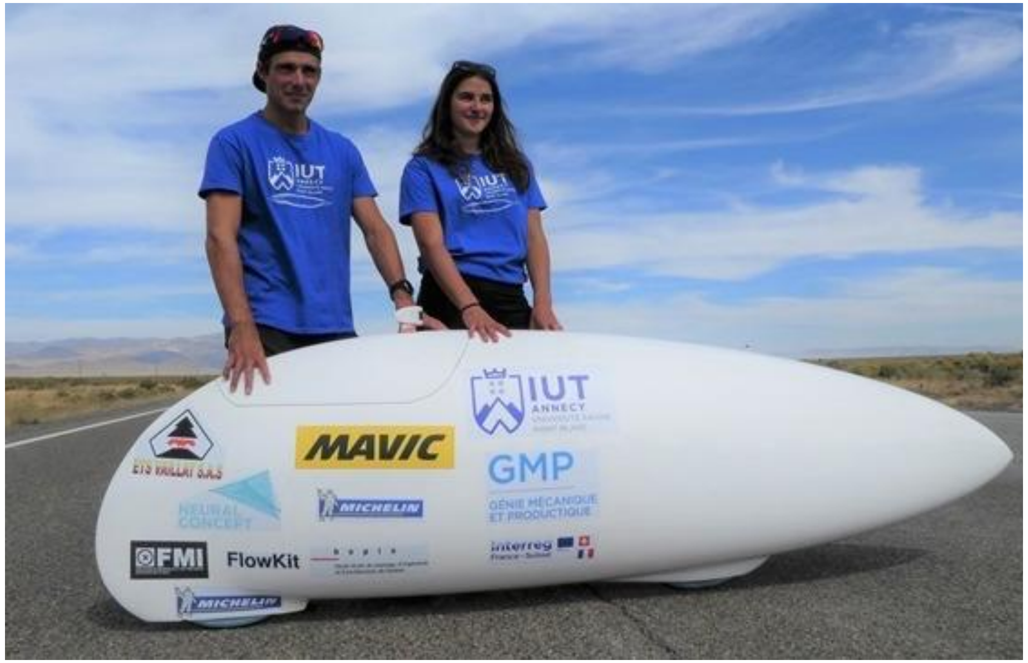
From SUV to Sports Car



Bicycle Shell



Altair 6, IUT Annecy, 2018



World Human Powered Speed Challenge
Battle Mountain Nevada, 2019

Women world record: 126,48 km/h
Men student world record: 136.74 km/h

They are going back in 2024!

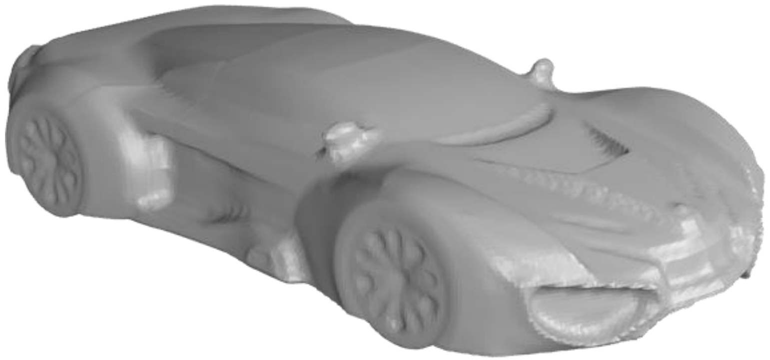
Neural implicit representations

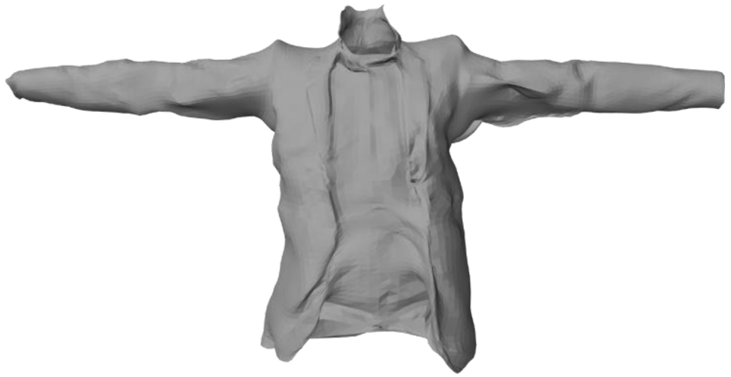
Current methods are based on **Signed Distance Fields** or **Occupancy Fields** and are limited to closed, watertight, surfaces

Common data (i.e. ShapeNet) is not watertight:

- Need for non-trivial preprocessing
- Loss of internal details (e.g. seats in a car)

Garments (and thin surfaces in general) can be more naturally expressed as open surfaces, removing the thickness-resolution requirements





Recently**, Unsigned Distance Fields** have been proposed to deal with **both closed and open surfaces**

Figures from Chibane J. et al. “Neural Unsigned Distance Fields for Implicit Function Learning”

From implicit to explicit

Many applications require explicit surfaces, usually in the form of **meshes:**

- Visualization
- Classification (and other downstream methods)
- Optimization

Meshing an implicit representation is not trivial, but there exist multiple classical algorithms for **watertight** representations

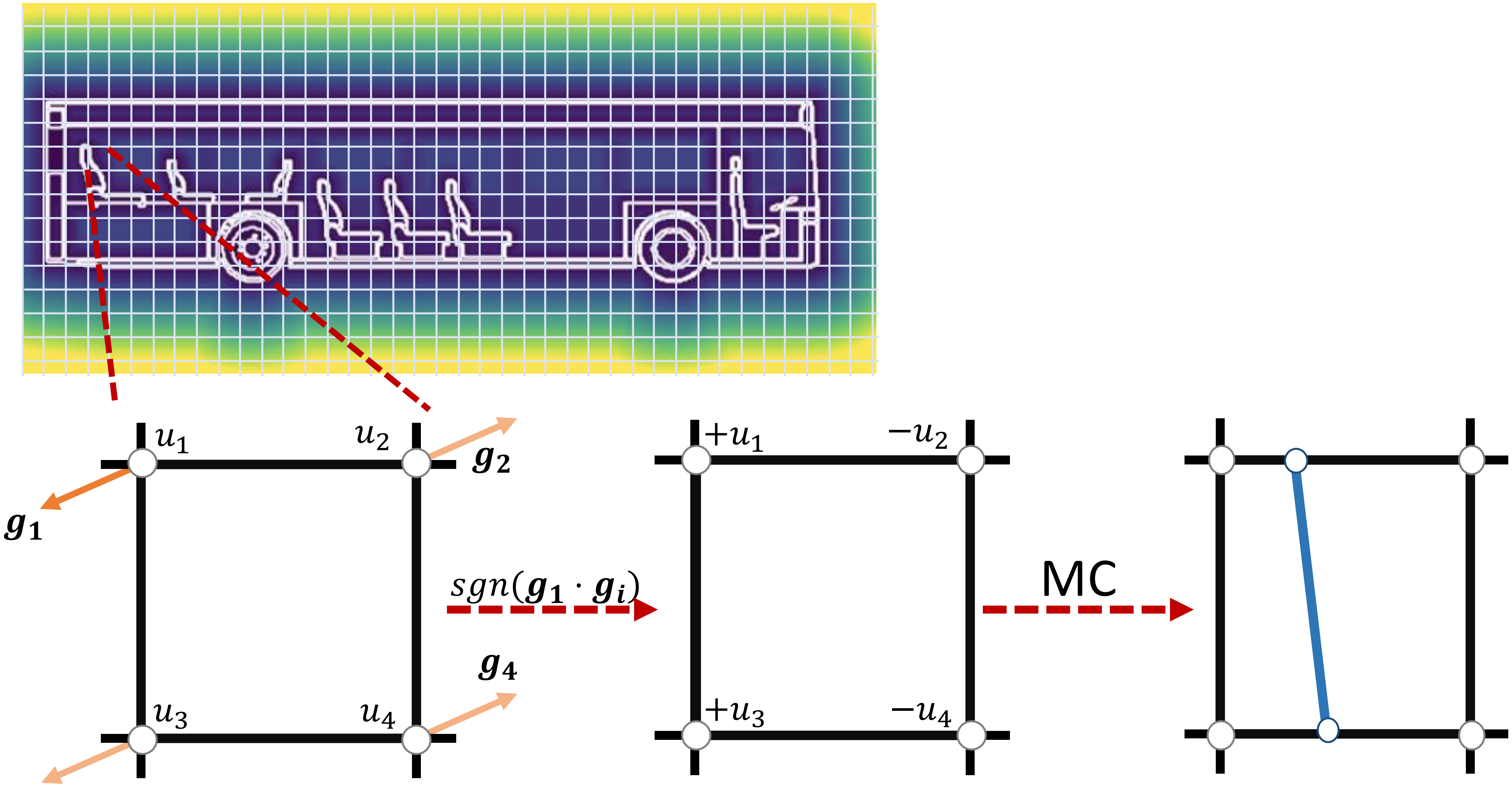
Correctly meshing an open surface from an Unsigned Field is an open problem

UDF Challenges

- How to extract the 0-level isosurface
- Not differentiable on the surface
- Need suitable networks to support the different characteristics of this field (sharpness around 0 and the gradient's abrupt orientation change)

MeshUDF

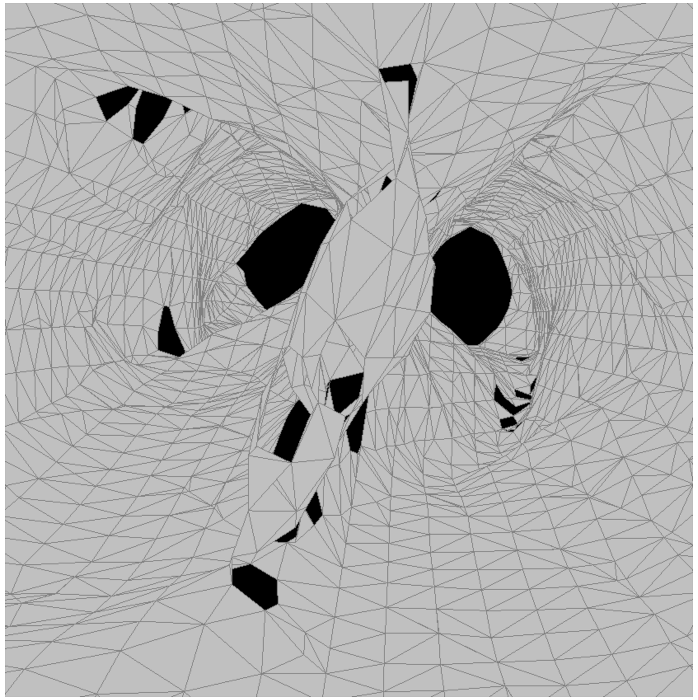
Local pseudo-sign retrieval via gradient dot product



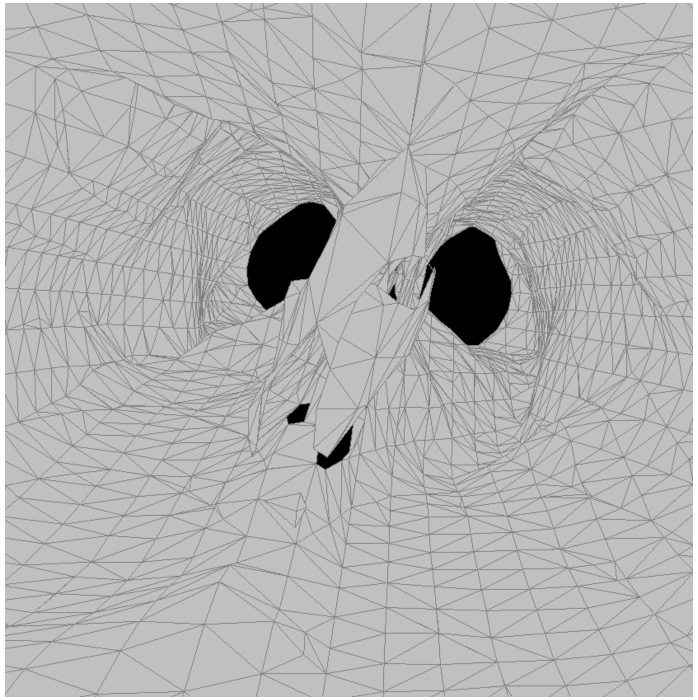
Guillard B., Stella F., Fua P., MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks, ECCV 2022

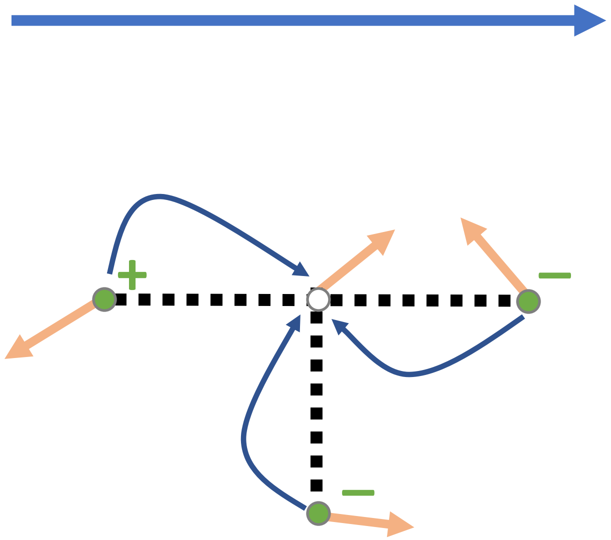
MeshUDF

Voting scheme to increase robustness



Undesirable holes



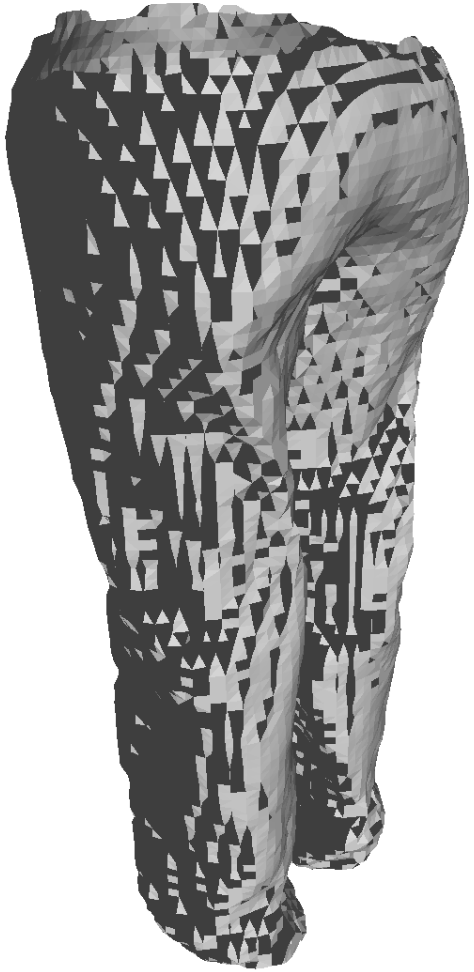


Fewer holes

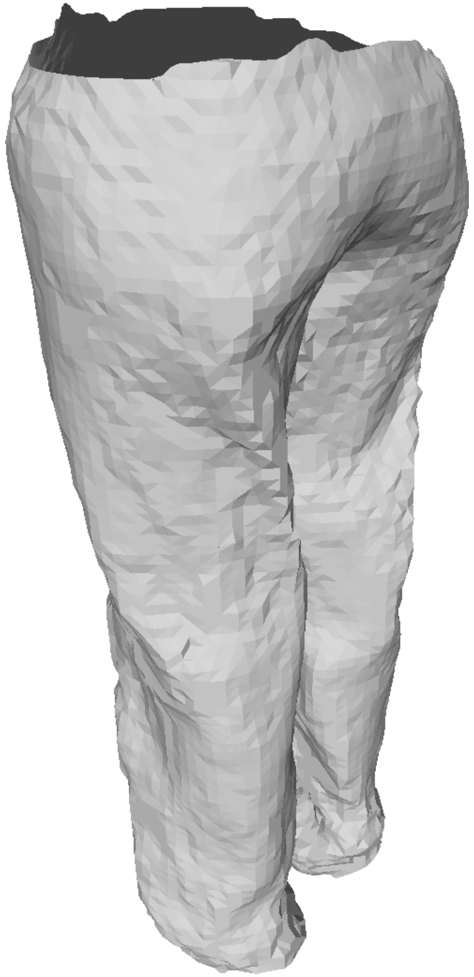
Guillard B., Stella F., Fua P., MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks, ECCV 2022

MeshUDF

Breadth-first exploration



Inconsistent normal orientations



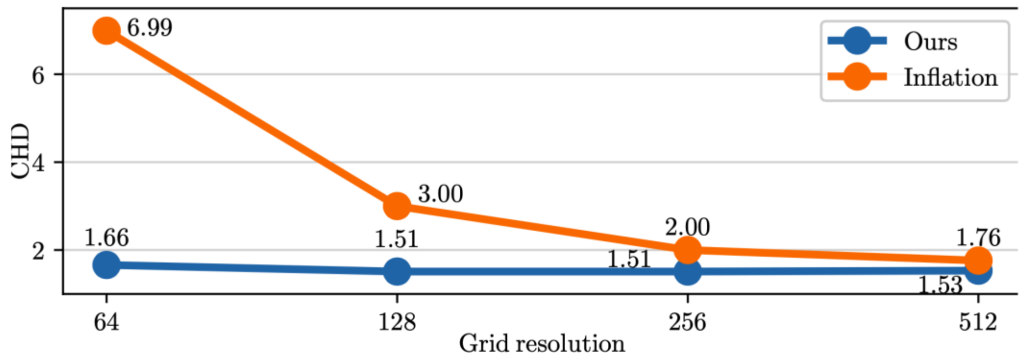
Better consistency

Guillard B., Stella F., Fua P., MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks, ECCV 2022

MeshUDF

Results: meshing garments

	BP	*Inflation*	*Ours*
CHD (↓)	1.62	3.00	**1.51**
IC (%, ↑)	92.51	88.48	**92.80**
NC (%, ↑)	89.50	94.16	**95.50**
Time (↓)	16.5s + 3000s	**1.0 sec.**	1.2 sec.



Inflation   Ball-Pivoting   MeshUDF

Guillard B., Stella F., Fua P., MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks, ECCV 2022

Have a neural network do the meshing

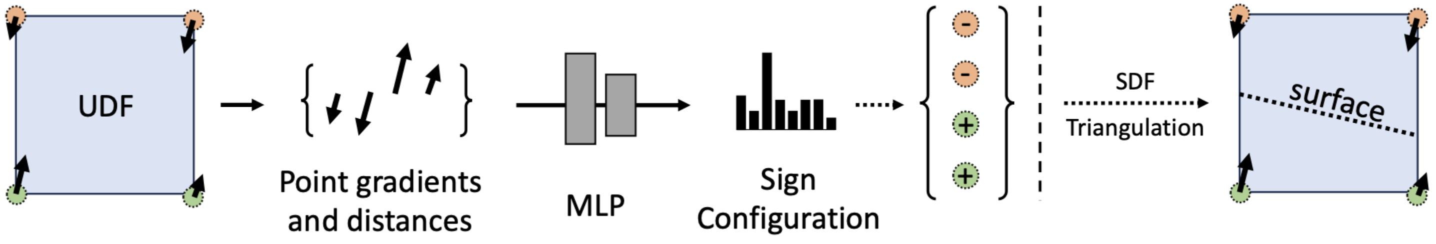
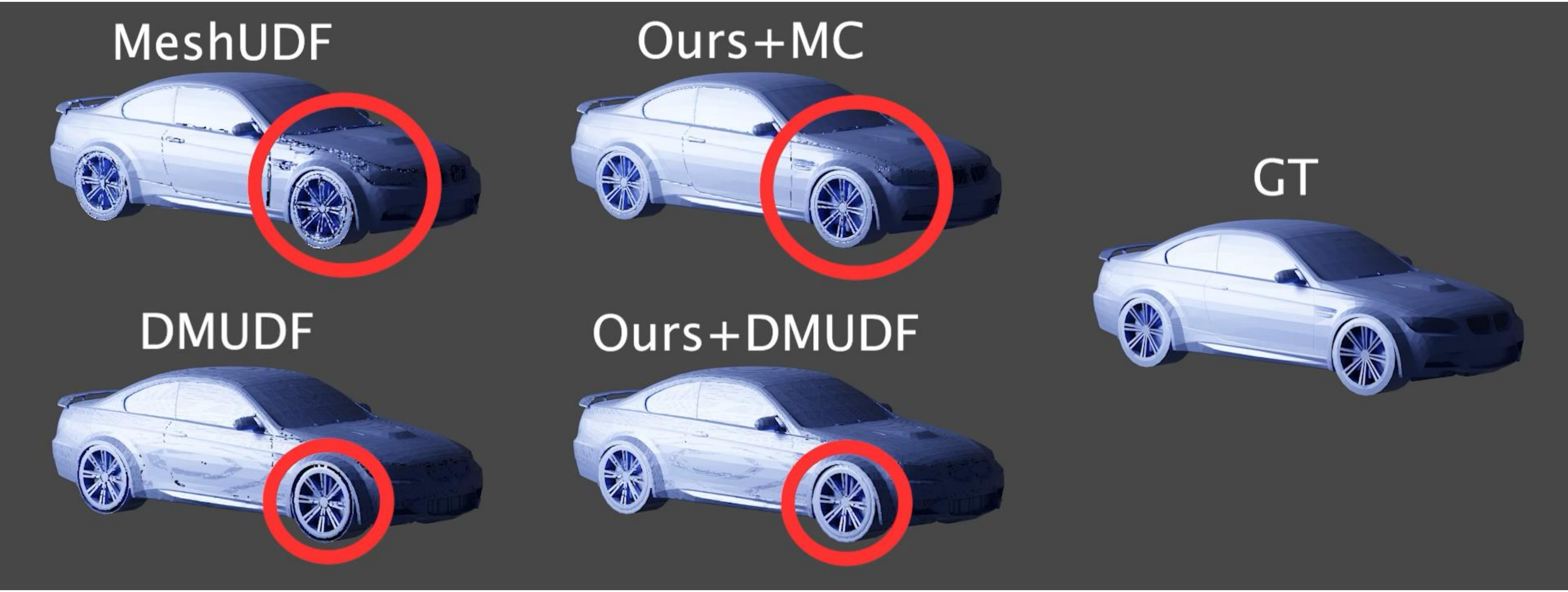


Fig. 2: Neural Surface Detection. We formulate the surface detection problem as a per-cell classification task. In each cell, we map point distances and gradients to a sign configuration of the cell vertices, which can be used to mesh the surface via Marching Cubes [15] or Dual Contouring [12].

Stella F., Talabot N., Le H., Fua P., Neural Surface Detection for Unsigned Distance Fields, to appear at ECCV 2024

Have a neural network do the meshing



Stella F., Talabot N., Le H., Fua P., Neural Surface Detection for Unsigned Distance Fields, to appear at ECCV 2024

Image Compression

Image Compression

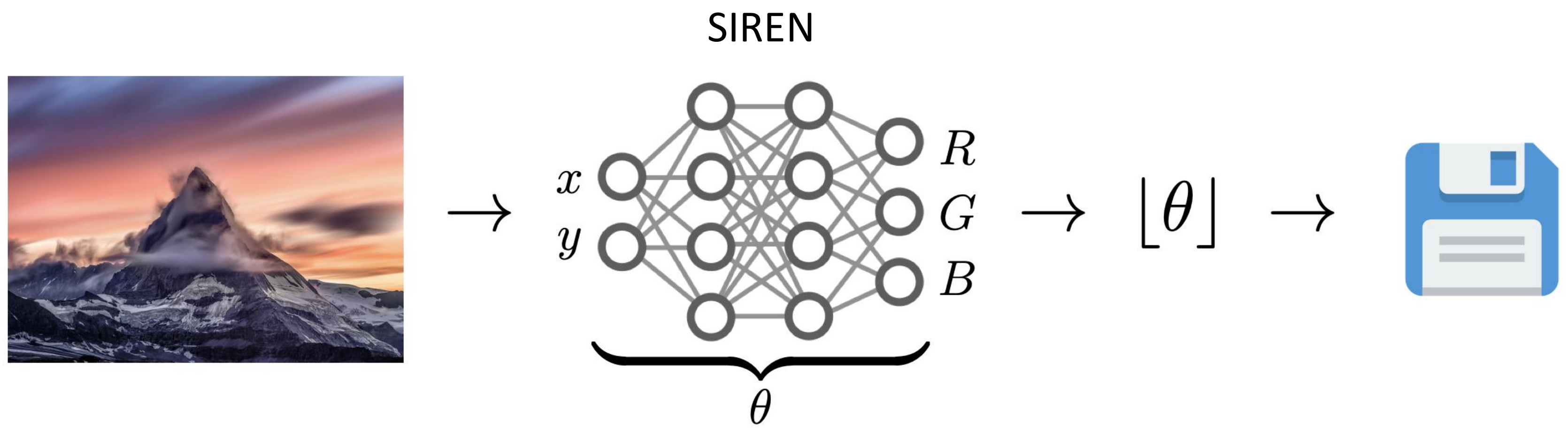


Figure 1: Compressed implicit neural representations. We overfit an image with a neural network mapping pixel locations (x, y) to RGB values (often referred to as an implicit neural representation). We then quantize the weights θ of this neural network to a lower bit-width and transmit them.

Sitzmann V., et al., Implicit Neural Representations with Periodic Activation Functions, NeurIPS 2020 Oral
Dupont E., et al., COIN: Compression with Implicit Neural Representations, Arxiv preprint 2021

COIN: Compression with Implicit Neural Representations

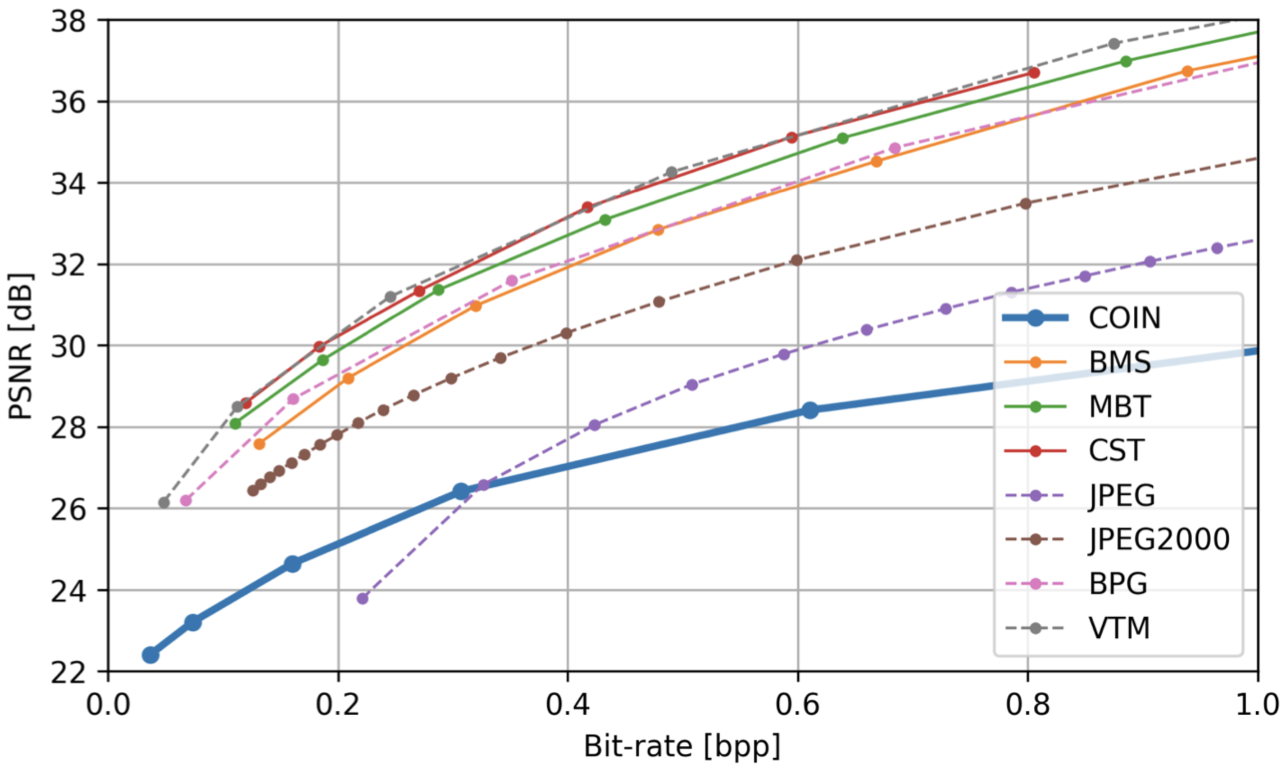


Figure 2: Rate distortion plots on the Kodak dataset.

Implicit Neural Representations for Image Compression

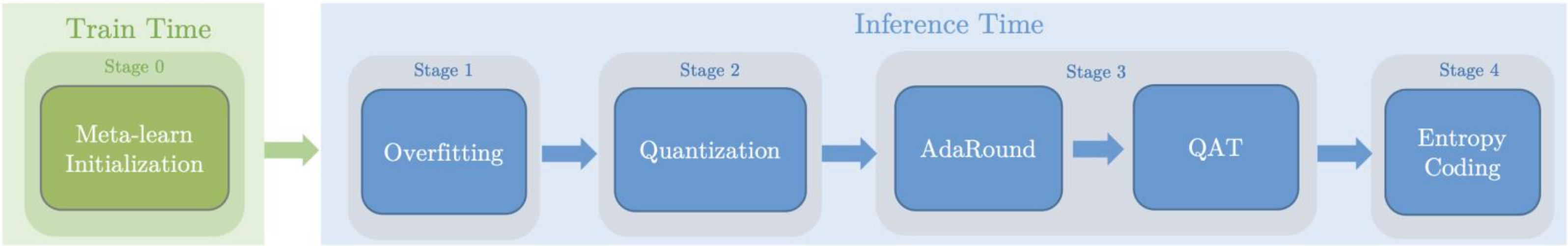
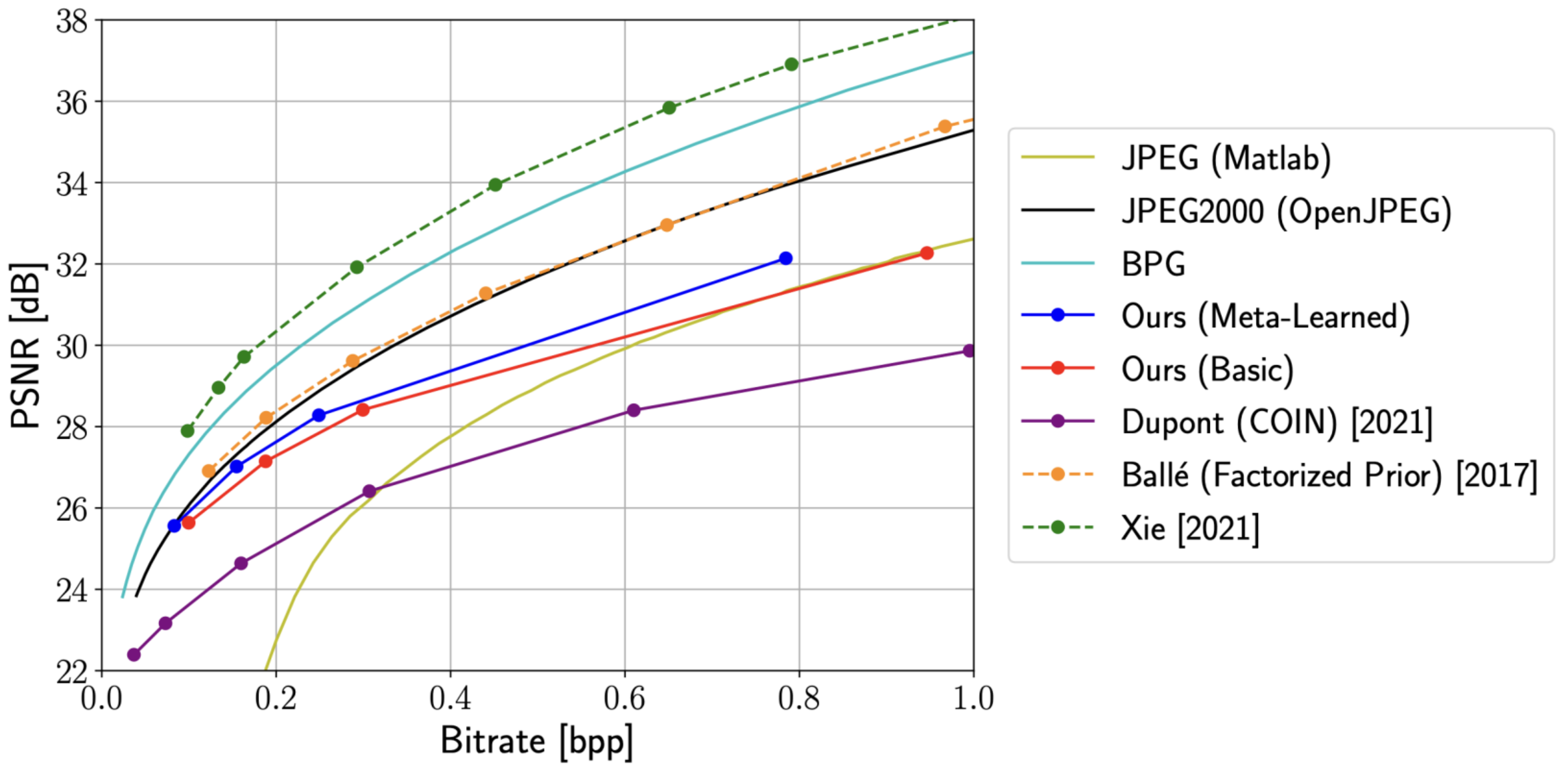


Fig. 2: Overview of INR-based compression pipeline. Blue: The basic compression pipeline comprising overfitting, quantization, AdaRound, QAT and entropy coding. Green: Additional meta-learning of initializations at training time.

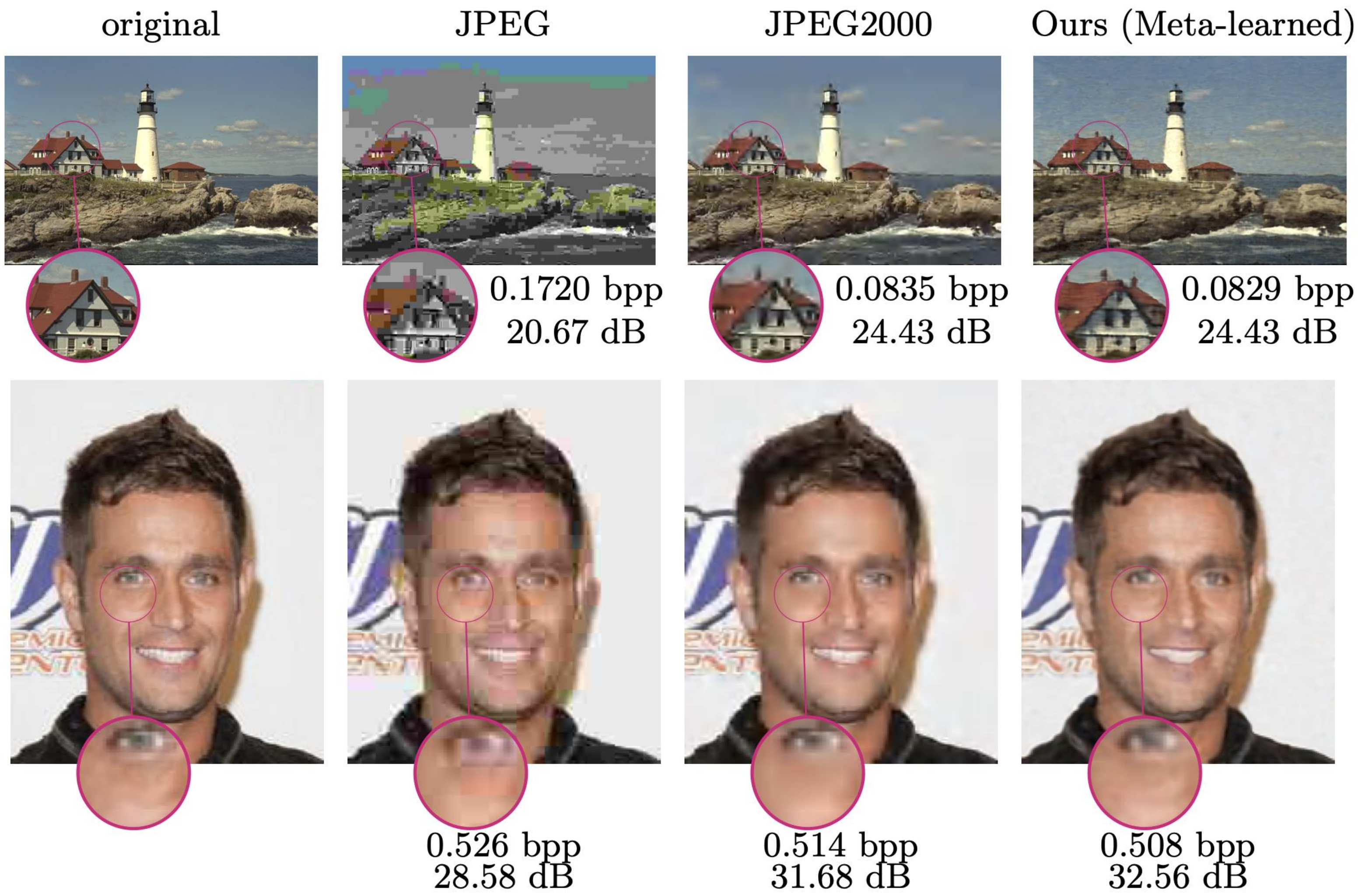
Strümpler Y., et al., Implicit Neural Representations for Image Compression, ECCV 2022

Implicit Neural Representations for Image Compression



Strümpler Y., et al., Implicit Neural Representations for Image Compression, ECCV 2022

Implicit Neural Representations for Image Compression



Strümpler Y., et al., Implicit Neural Representations for Image Compression, ECCV 2022