Neural Surface Reconstruction

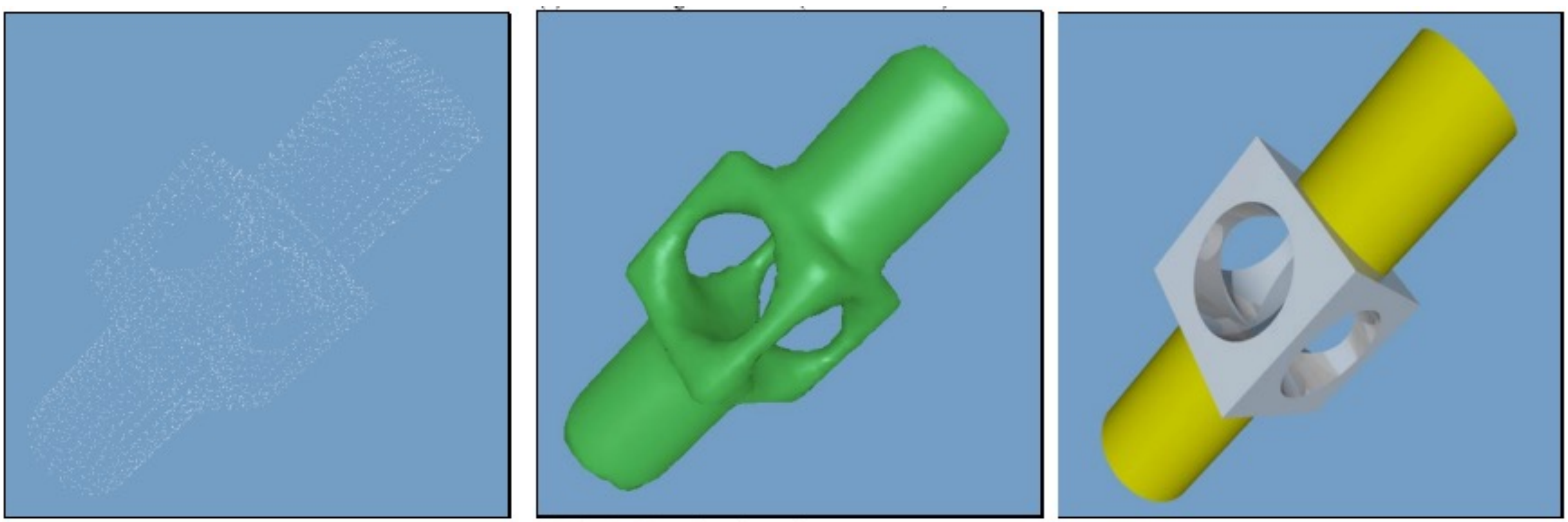
Richard (Hao) Zhang

CMPT 464/764: Geometric Modeling in Computer Graphics

Lecture 9

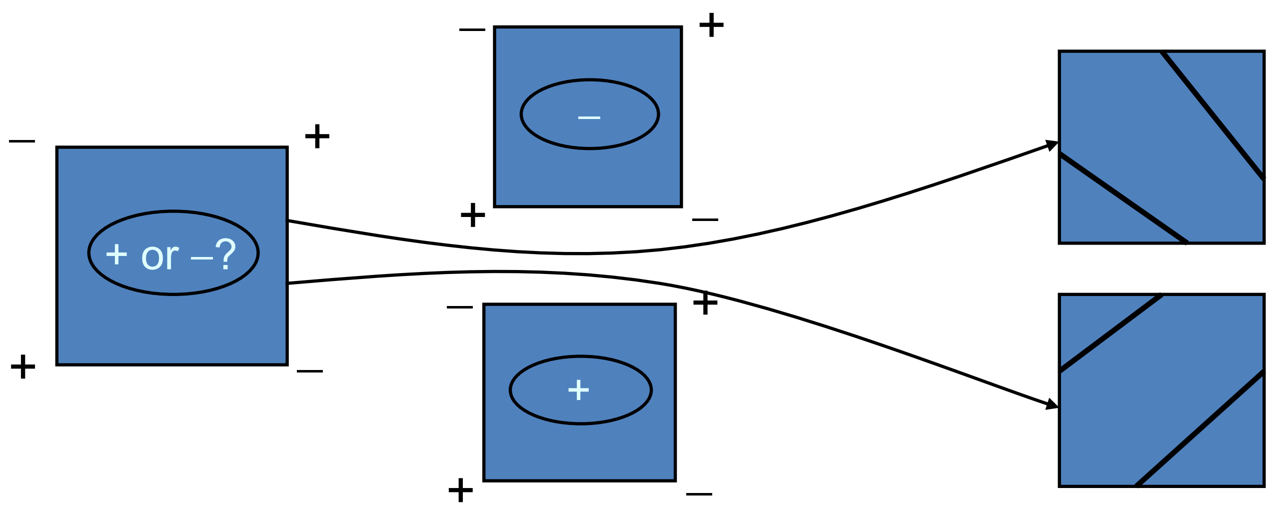
Start with Marching Cubes

- Issue #1: unable to recover sharp features



Start with Marching Cubes

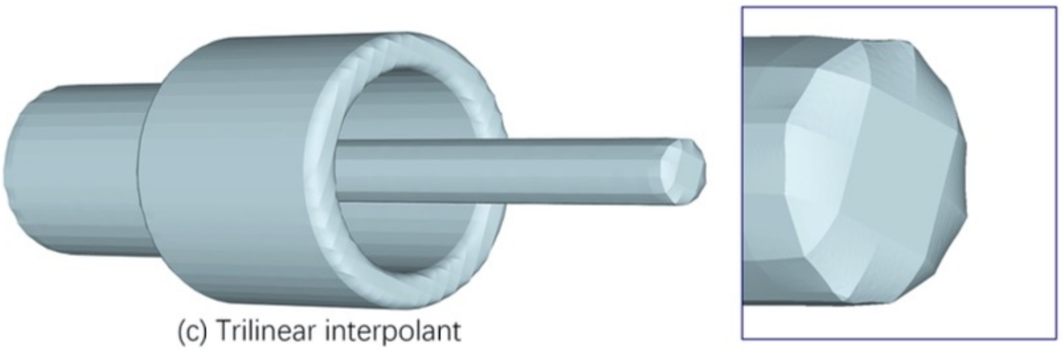
- Issue #2: model-driven — need assumption on unknowns



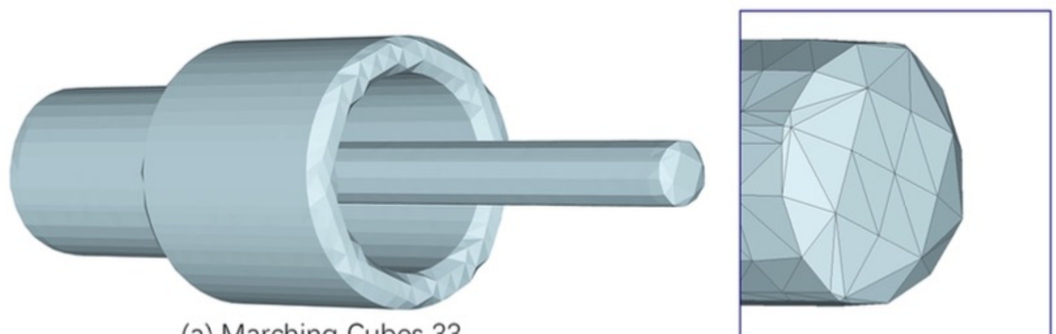
Recall: asymptotic decider uses bilinear interpolation

Natural assumption in 3D: trilinearity

- Results of using trilinear interpolants



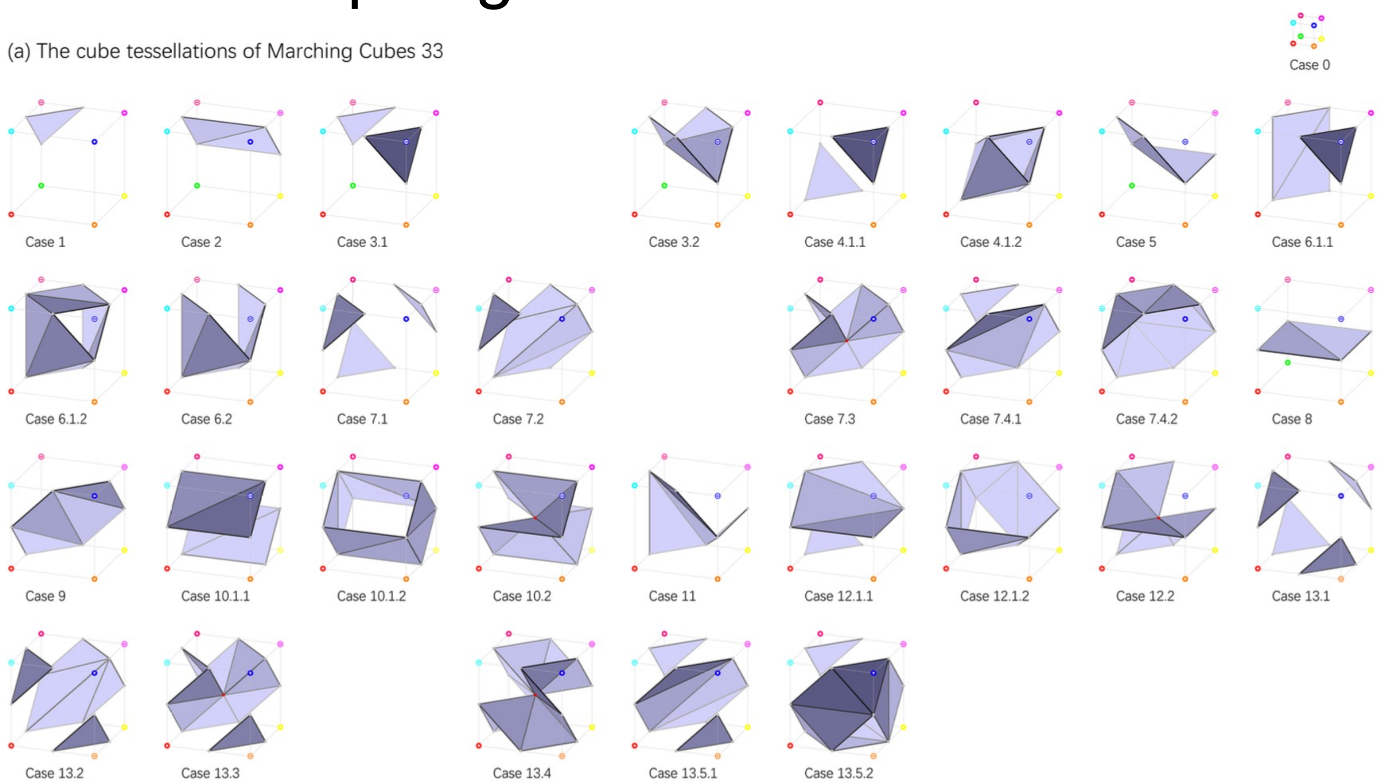
(c) Trilinear interpolant



(a) Marching Cubes 33

Marching Cubes 33 [Chernyaev 1995]

- Enumerated all topological cases based on trilinearity

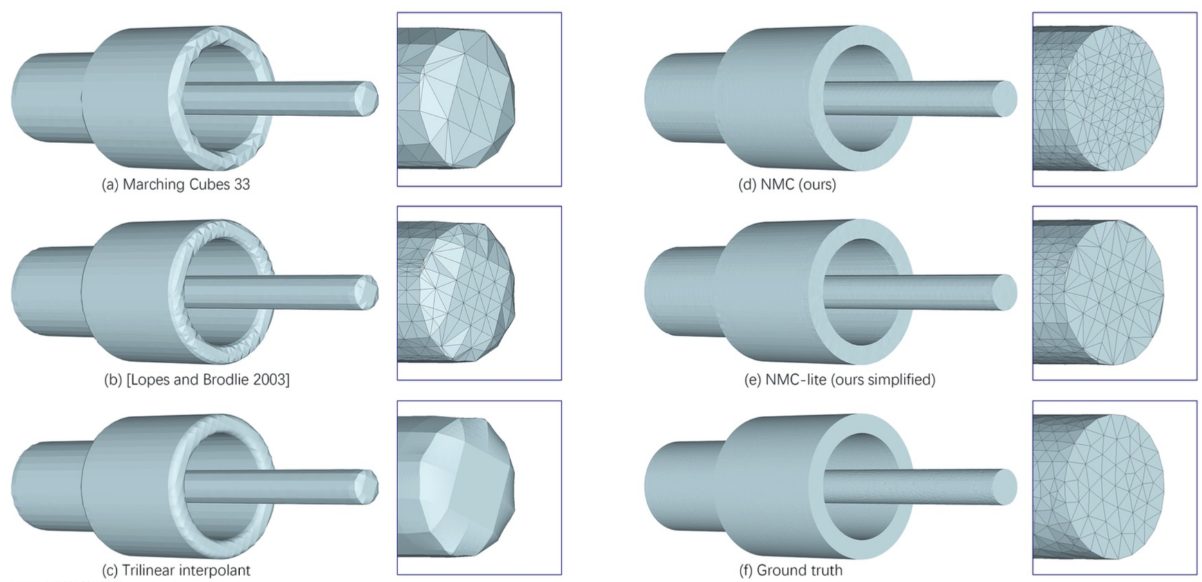


Neural Marching Cubes (NMC)

- Data-driven: learn tessellations from training data

Neural Marching Cubes

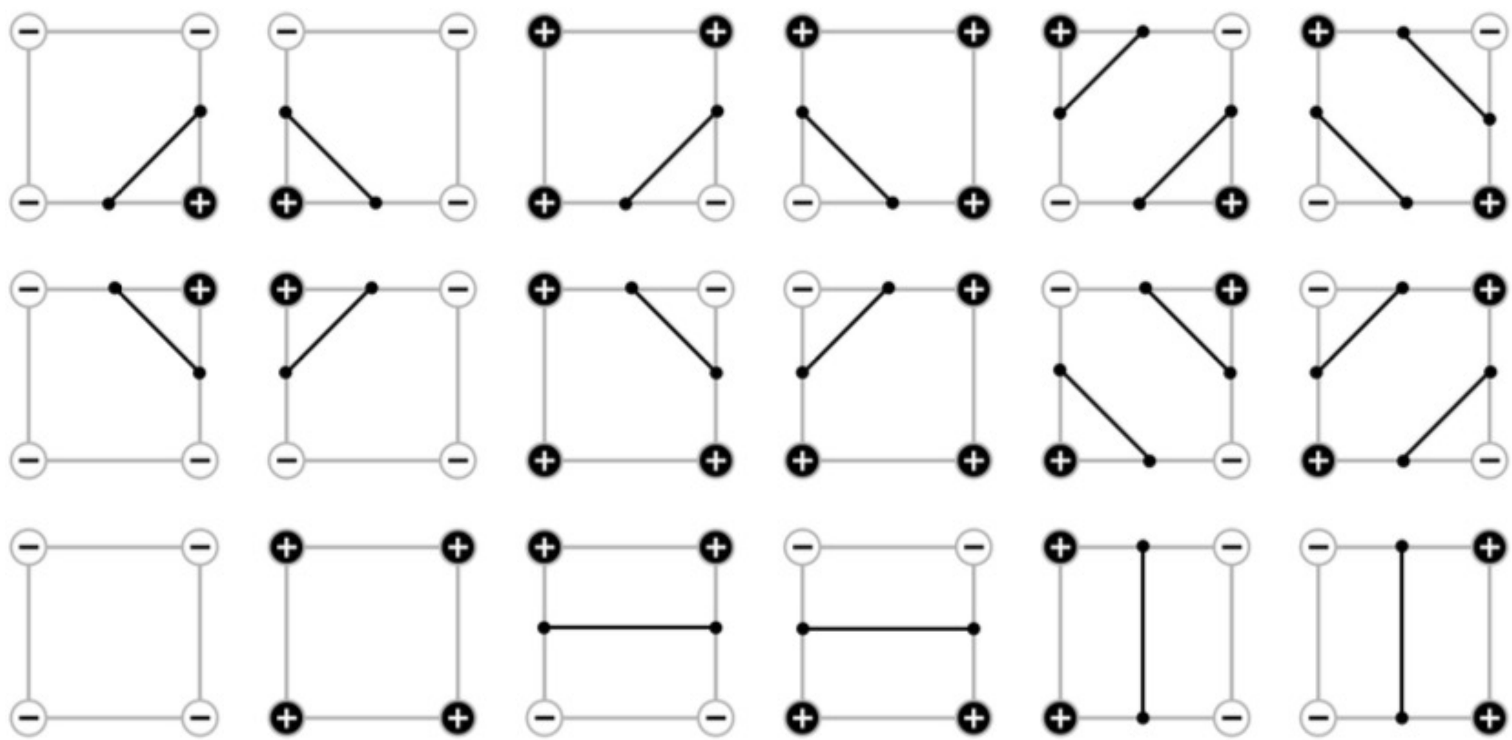
ZHIQIN CHEN, Simon Fraser University, Canada
HAO ZHANG, Simon Fraser University, Canada



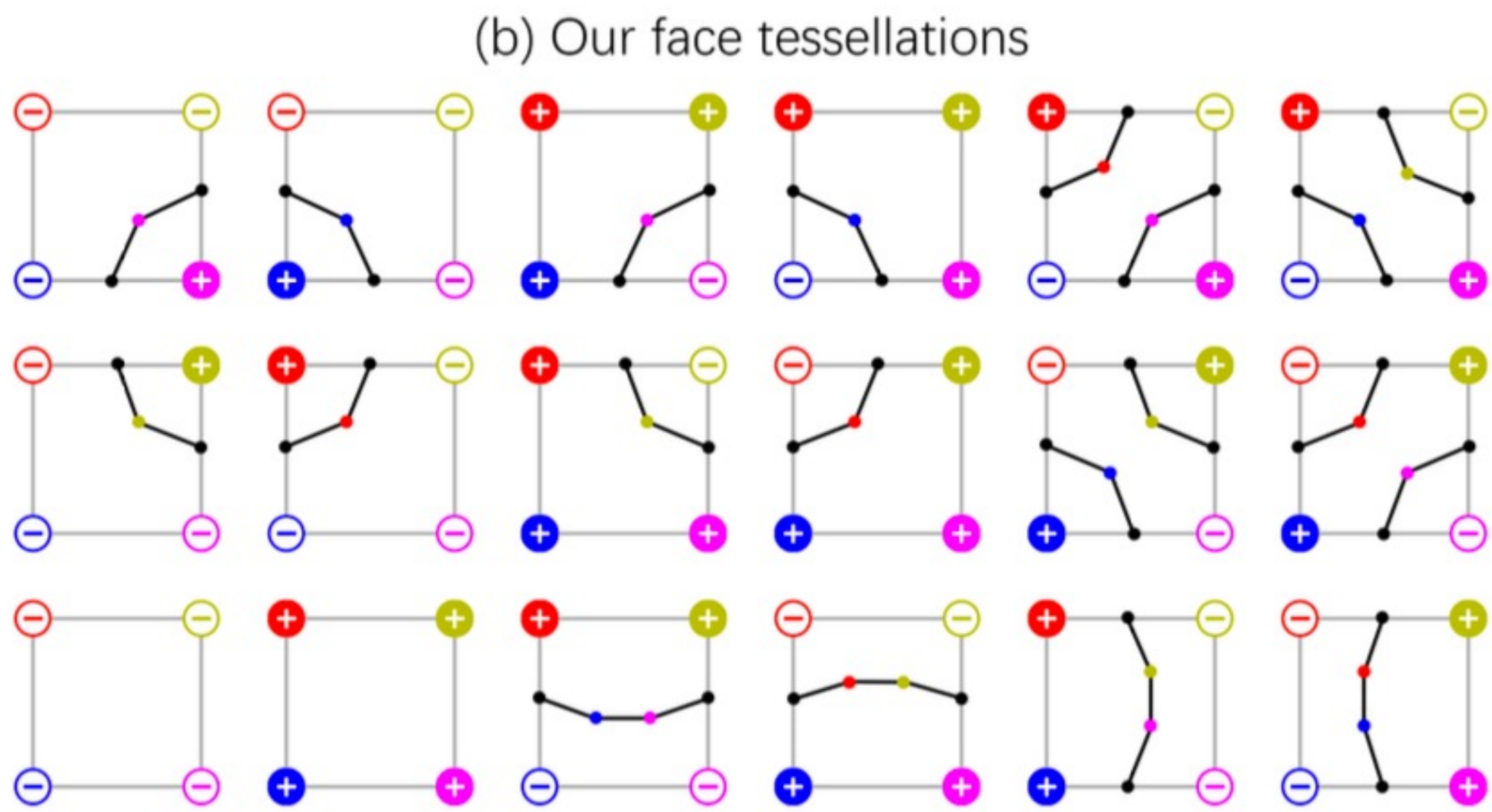
[Chen and Zhang, SIGGRAPH Asia 2021]

MC vs. NMC cases in 2D

(a) The face tessellations of Marching Cubes

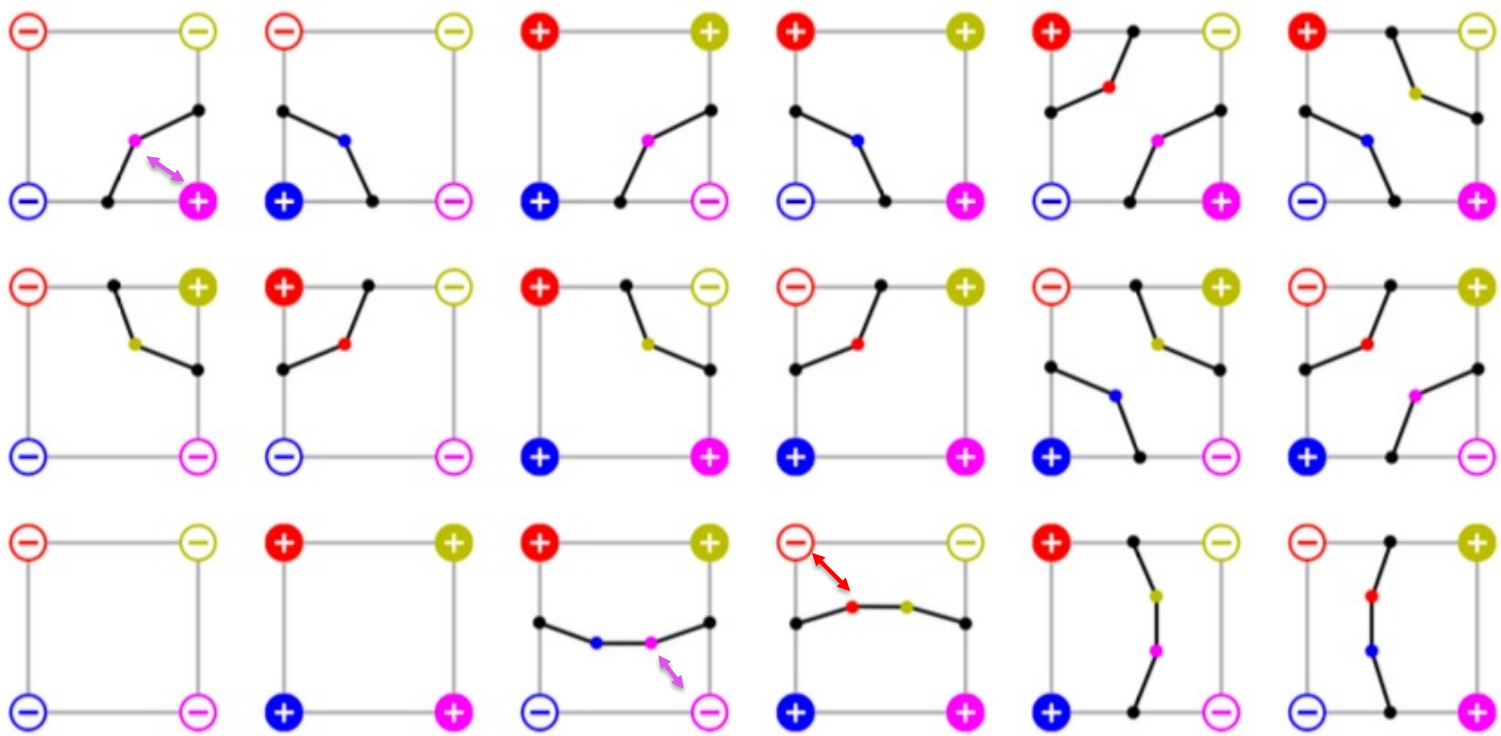


MC vs. NMC cases in 2D



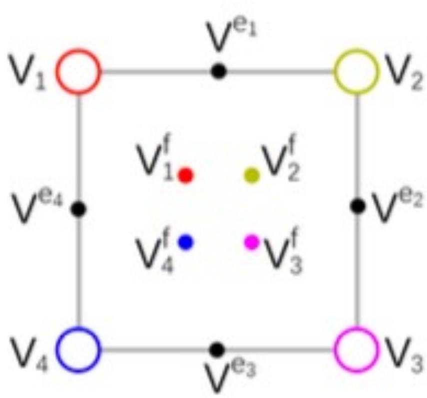
Associating face and corner vertices

(b) Our face tessellations



Fixed-length vectors to store tessellations

(c) Our representation to store each square

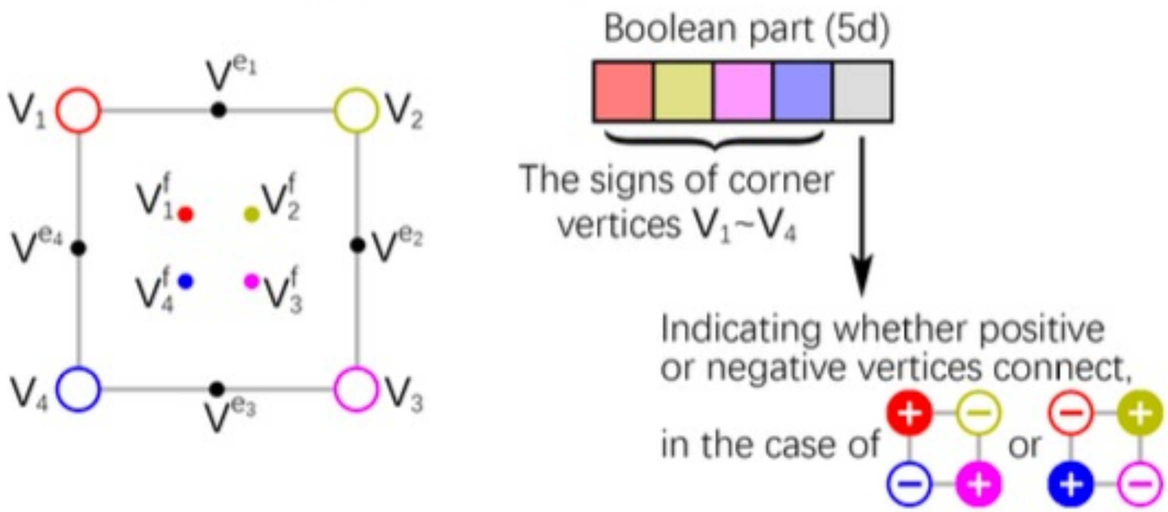


Four edge vertices

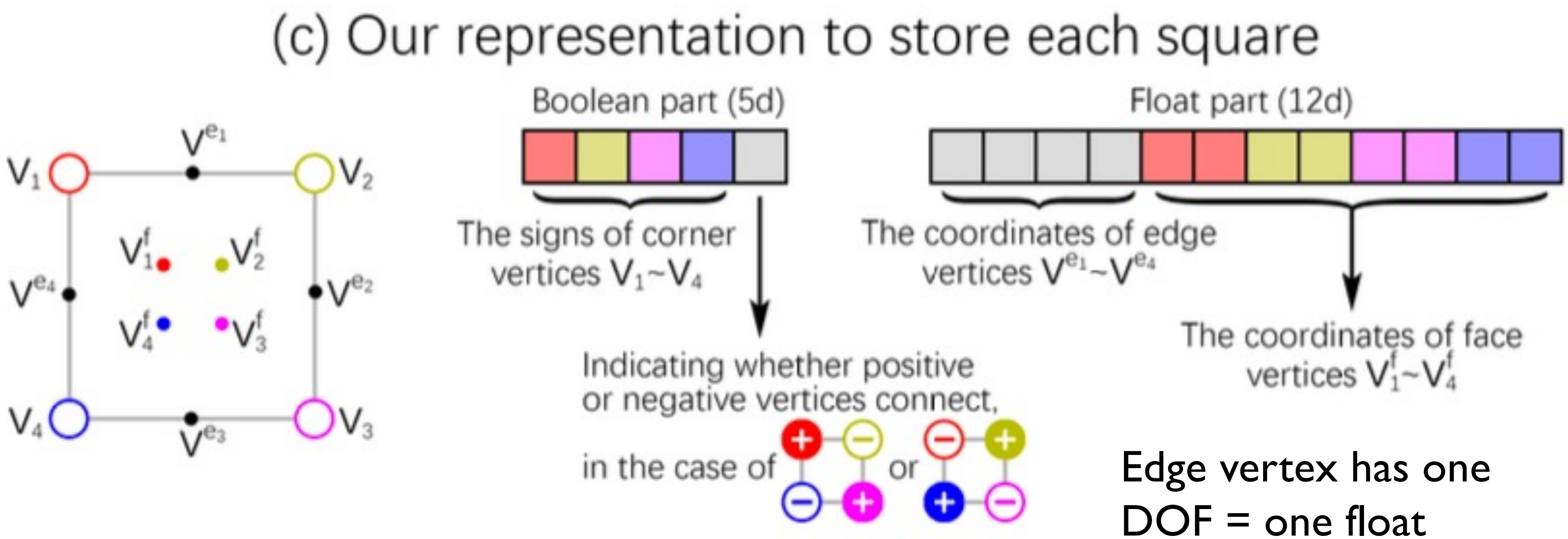
Four face vertices with association to corners

Fixed-length vectors to store tessellations

(c) Our representation to store each square



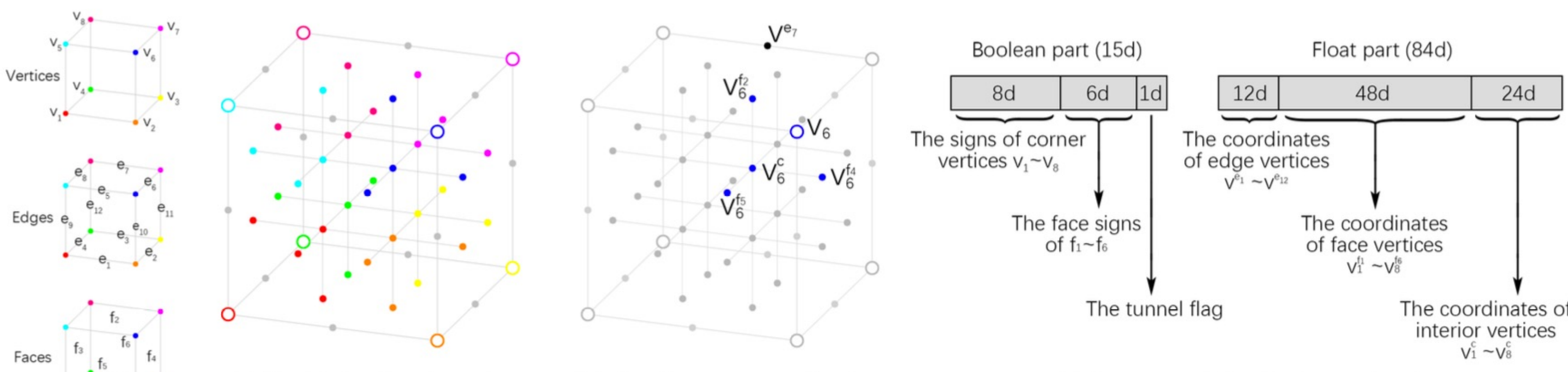
Fixed-length vectors to store tessellations



Edge vertex has one DOF = one float

Face vertex has two DOFs = two floats

3D case to parameterize a cube tessellation



(a) All added vertices in a cube

(b) all vertices corresponding to v_6 and an edge vertex on e_7

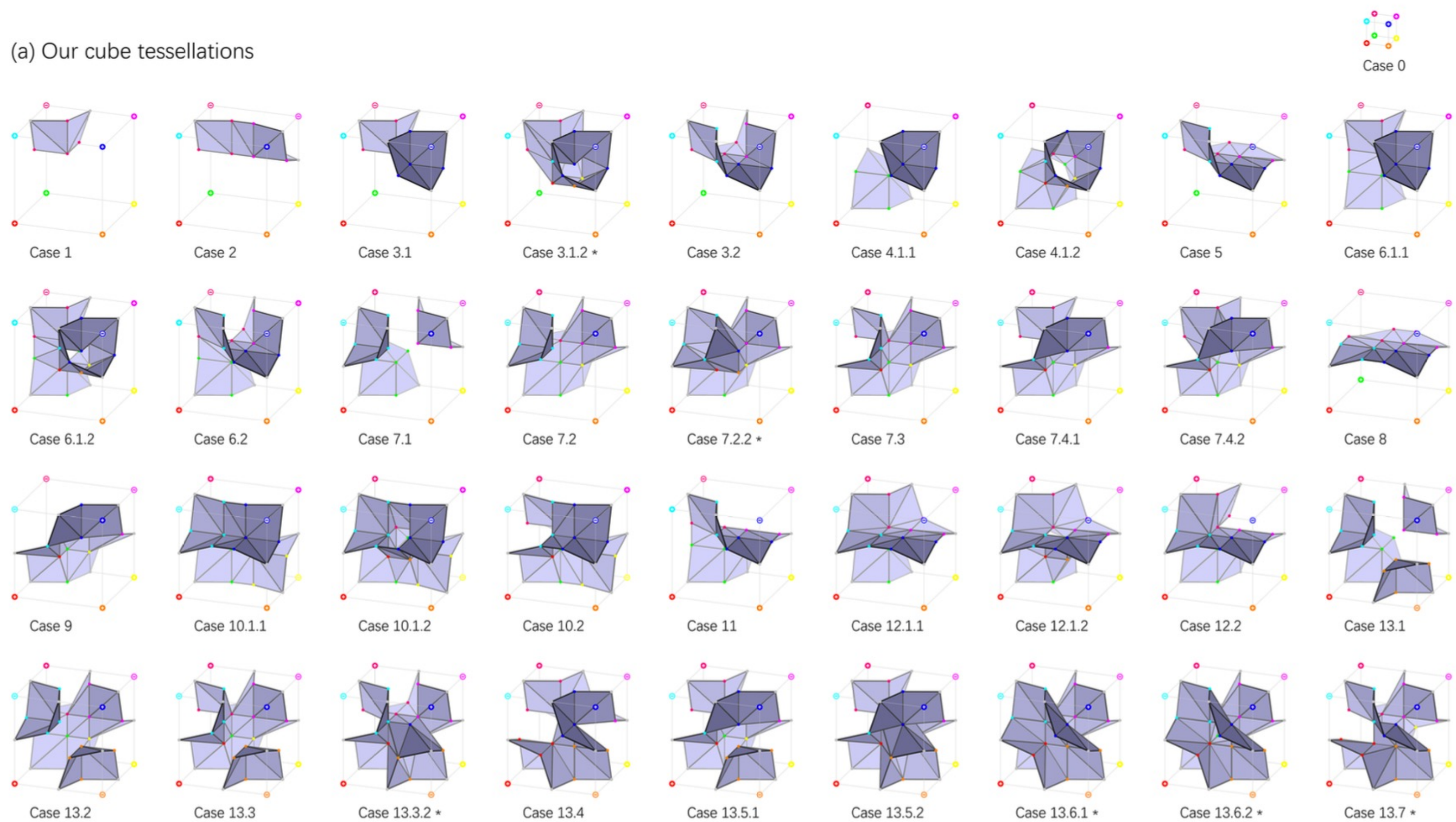
(c) Our representation to store each cube

12 edge vertices

4 face vertices per cube face

8 additional interior vertices

Cube tessellation cases



Neural network to predict tessellations

- A 3D CNN is trained to predict topological cases and vertex positions for all cubes
- Use a 7^3 receptive field to provide local contexts

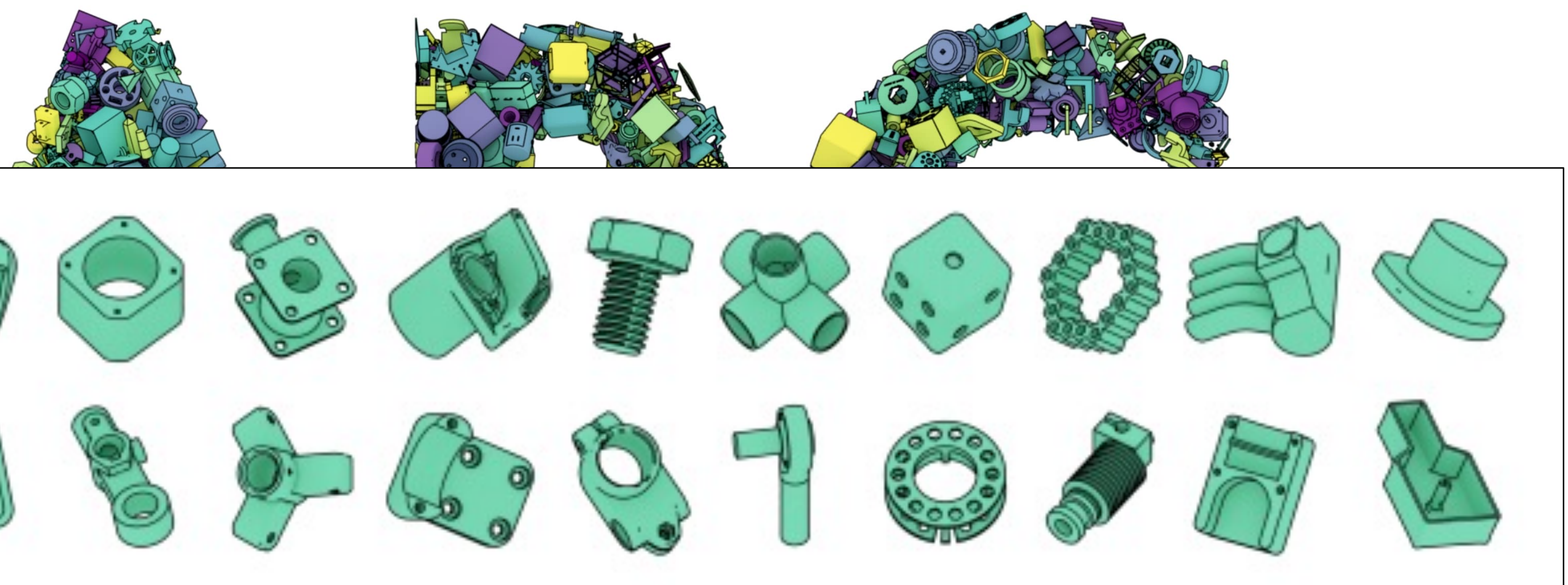
N x M x K field values at grids → 3D Residual Network or ResNet (a 3D CNN formed by residual blocks that learn residual functions) →

N x M x K x 5 Booleans

N x M x K x 51 Floats

Training set: ABC

- A Big Cad dataset: consisting of CAD models



Qualitative results and comparison

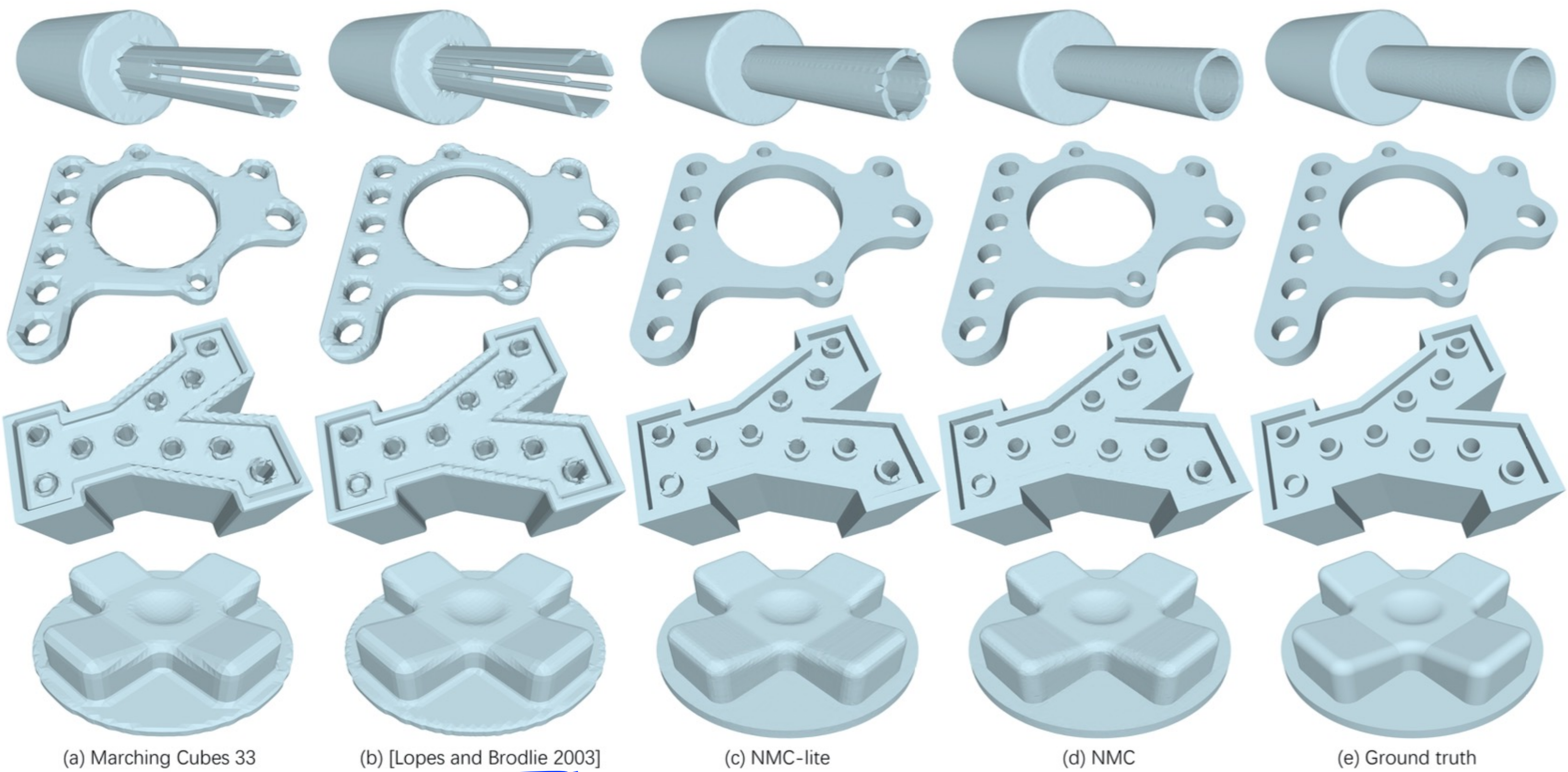


Fig. 9. Results of reconstructing 3D meshes from SDF grid inputs at 64^3 resolution. The shapes in the first two rows are from the ABC test set, and the last two rows from Thingi10K. More results and their mesh tessellations can be found in the supplementary material.

NMC-Lite with simpler tessellations

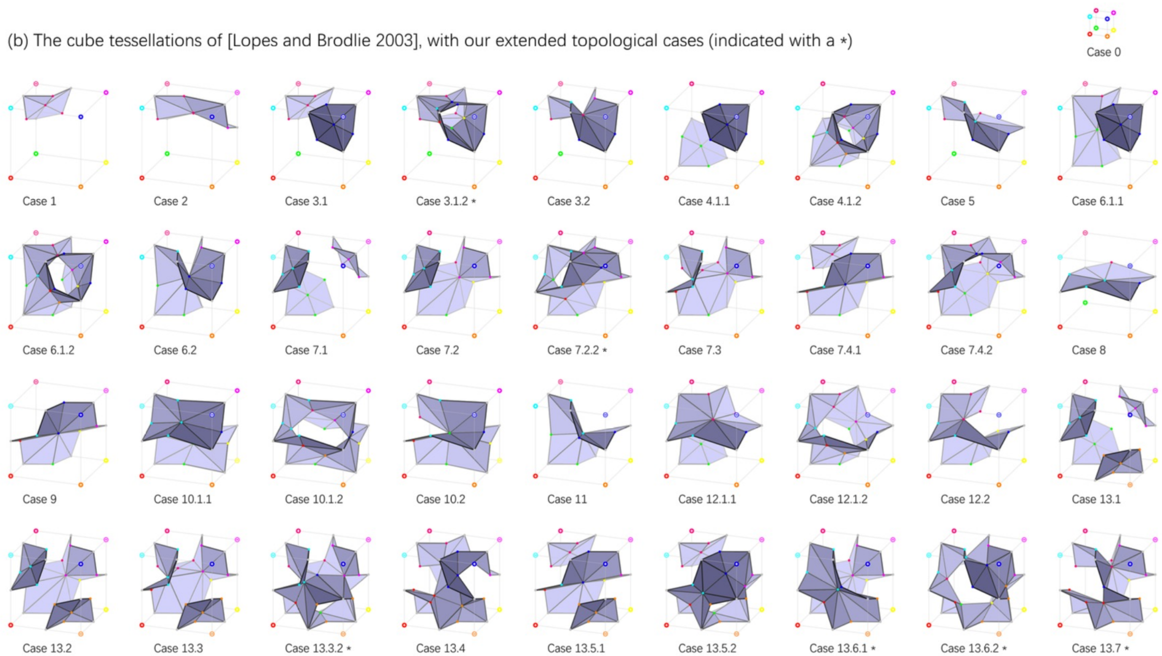
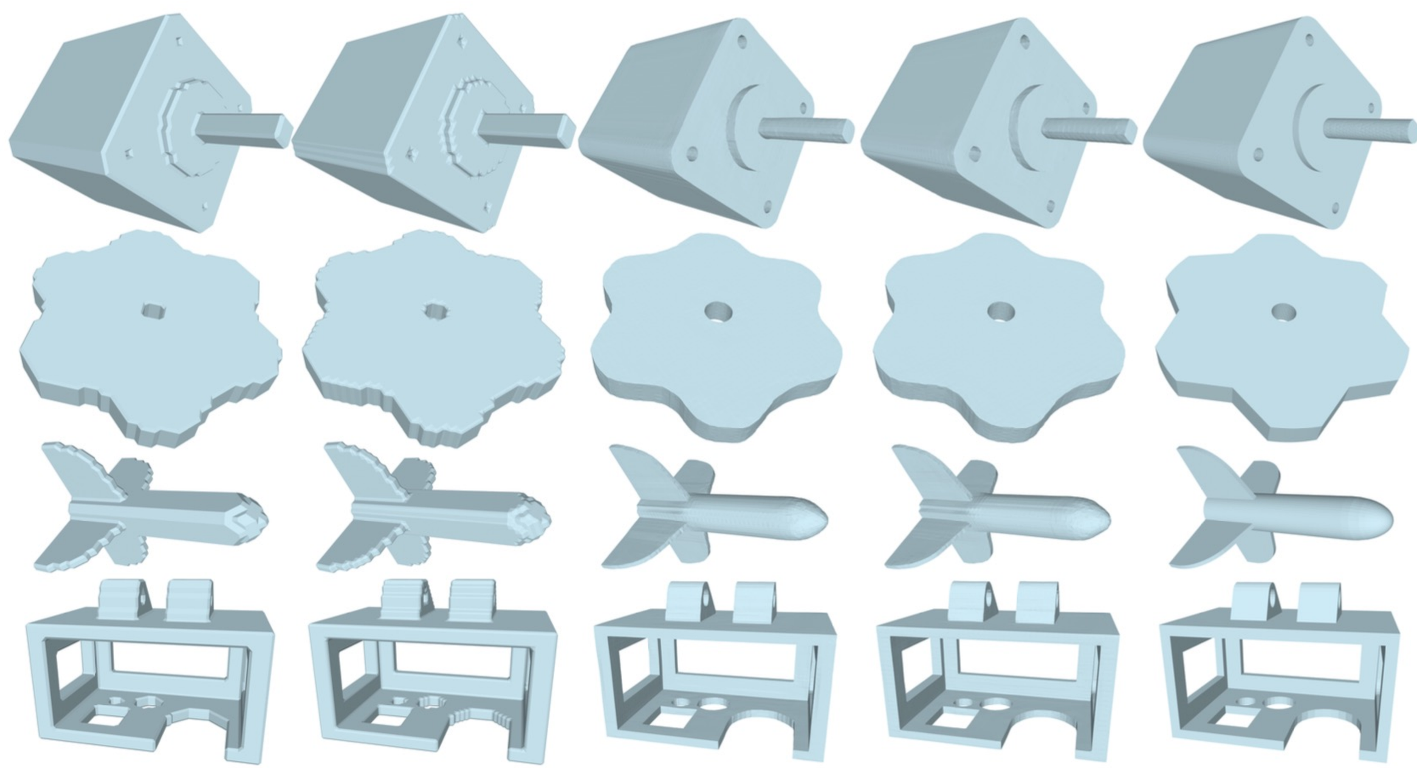


Fig. 4. The 3D cube tessellations of Marching Cubes 33 [Chernyaev 1995] and [Lopes and Brodlie 2003]. Note that they both present 31 cases, since Case 12.3 is equivalent to Case 12.2 and Case 14 is equivalent to Case 11, with respect to rotational and mirroring symmetries. In (b), we also add our extended topological cases to [Lopes and Brodlie 2003], indicated with a *, to form a simplified version of our NMC tessellations, denoted as NMC-lite.

Different tessellation designs require different Ground Truth (GT) data preparation to supervise the 3D CNN training

Qualitative results and comparison

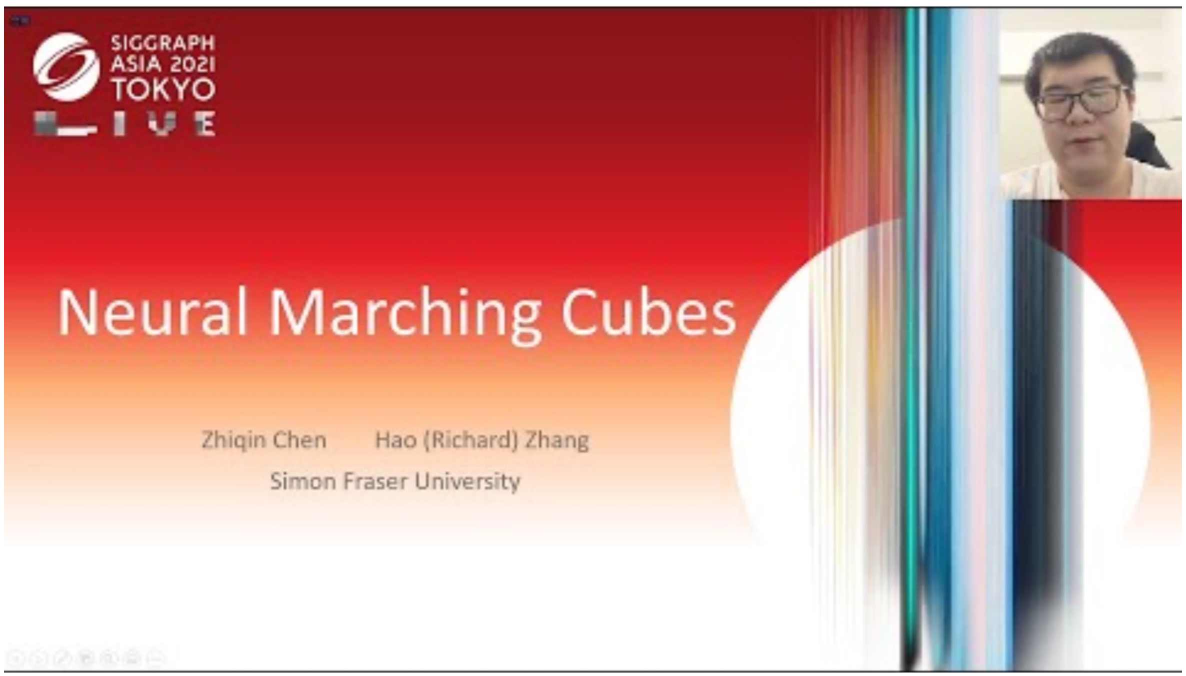


(a) Marching Cubes 33 (b) [Lopes and Brodlie 2003] (c) NMC-lite (d) NMC (e) Ground truth

Fig. 12. Results of reconstructing 3D meshes from binary voxel/occupancy inputs at 64^3 resolution. The shapes in the first two rows are from the ABC test set, and the last two rows from Thingi10K. More results and their mesh tessellations can be found in the supplementary material.

Youtube video on NMC presentation

https://youtu.be/O7NFYN3YzDM?si=a55JVztKUcaxJixw



Pros and cons of NMC

😀 Generalizes well to broad range of shapes: local receptive fields

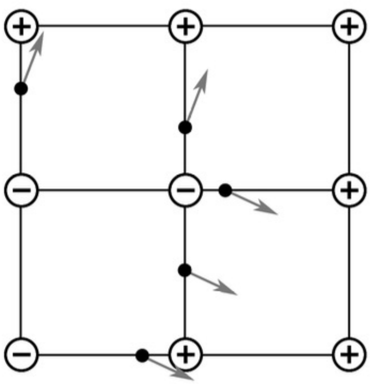
😀 Data-driven and excels at recovering sharp features

Pros and cons of NMC

- 😀 Generalizes well to broad range of shapes: local receptive fields
- 😀 Data-driven and excels at recovering sharp features
- 🙁 Tessellation templates more complex than those of MC/MC33
- 🙁 Output 4-8x more triangles
- 🙁 Incurs 100x computation time to reconstruct a mesh

Follow-up: neural dual contouring (NDC)

❖ First, classical dual contouring (DC)

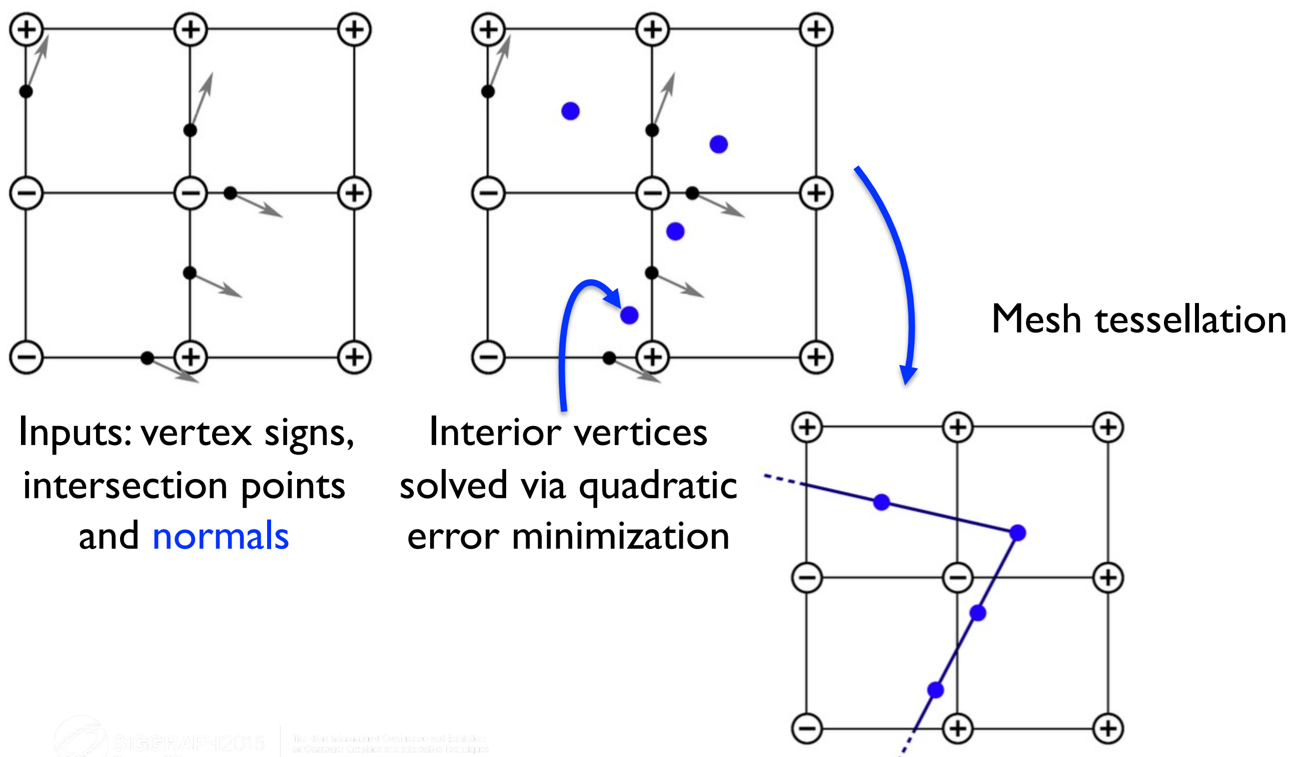


Inputs: vertex signs, intersection points and normals

[Chen et al. SIGGRAPH 2022]

Classical dual contouring: require normals

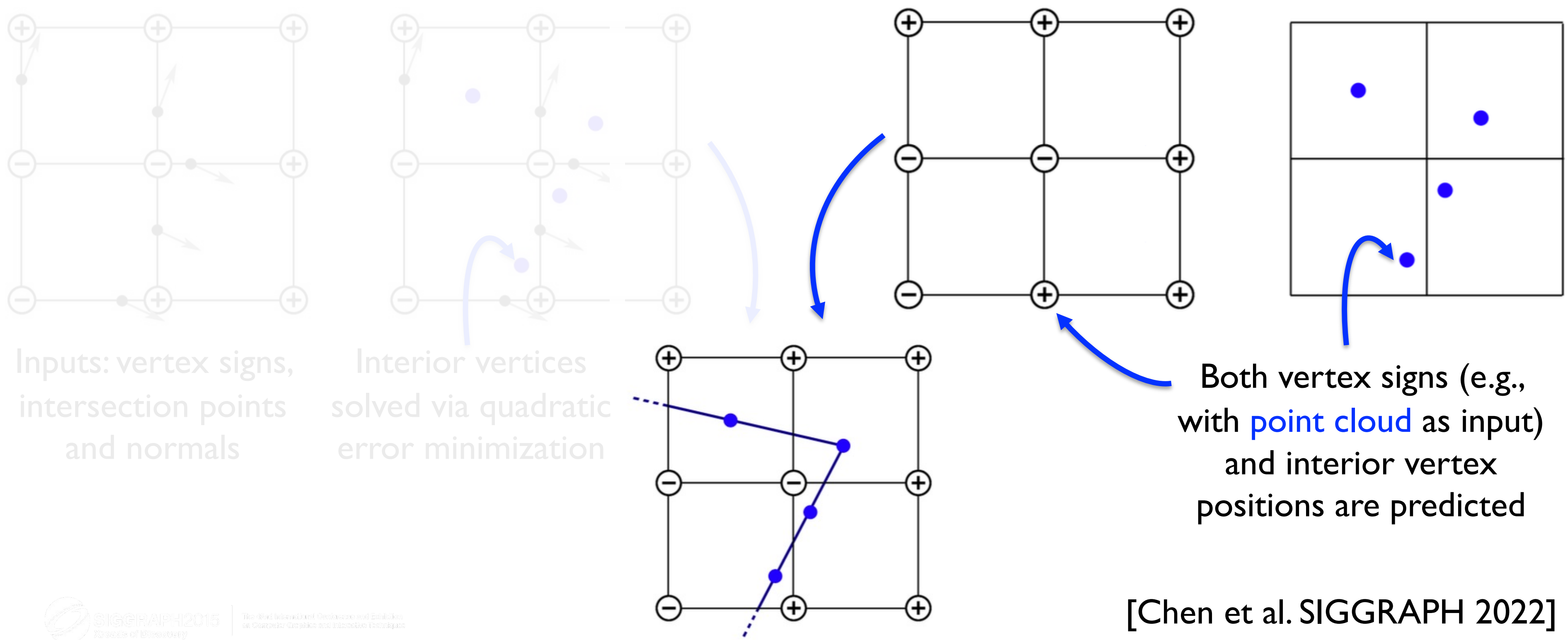
❖ First, classical dual contouring (DC)



[Chen et al. SIGGRAPH 2022]

Neural dual contouring (NDC): no normals

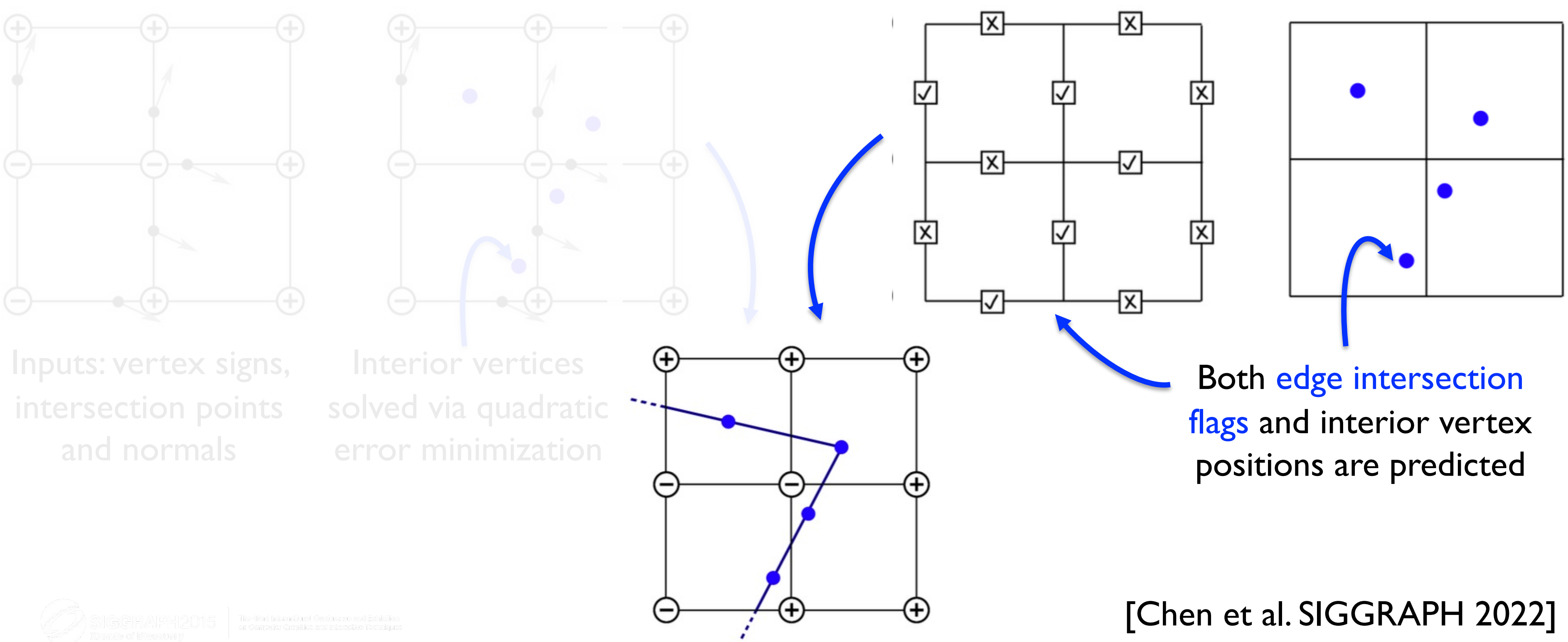
- NDC: train CNNs to directly predict vertex signs and positions



[Chen et al. SIGGRAPH 2022]

Unsigned neural dual contouring (UNDC)

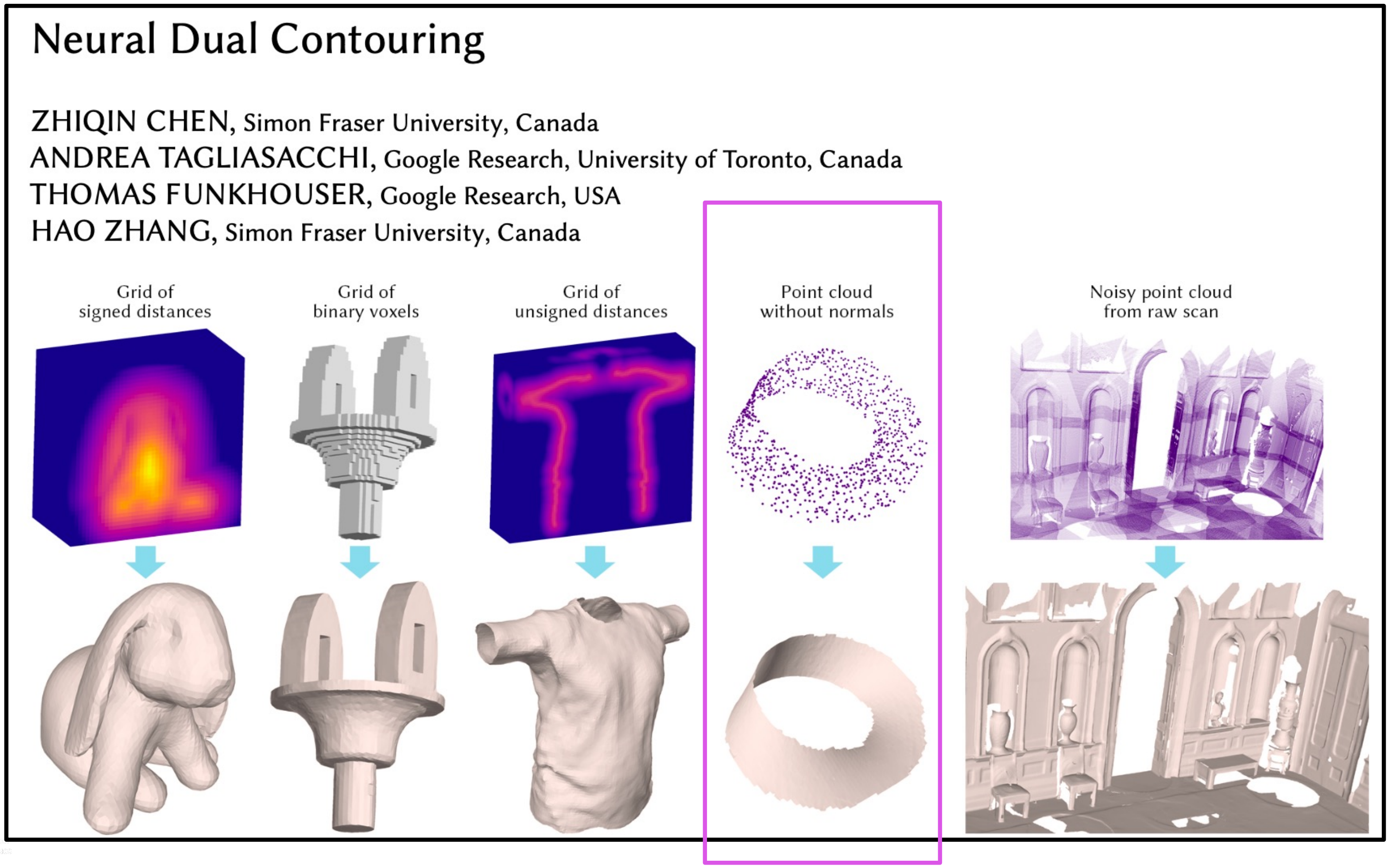
❖ Unsigned NDC: directly predict edge flags and vertex positions



[Chen et al. SIGGRAPH 2022]

Neural dual contouring (NDC)

❖ Can take on various inputs



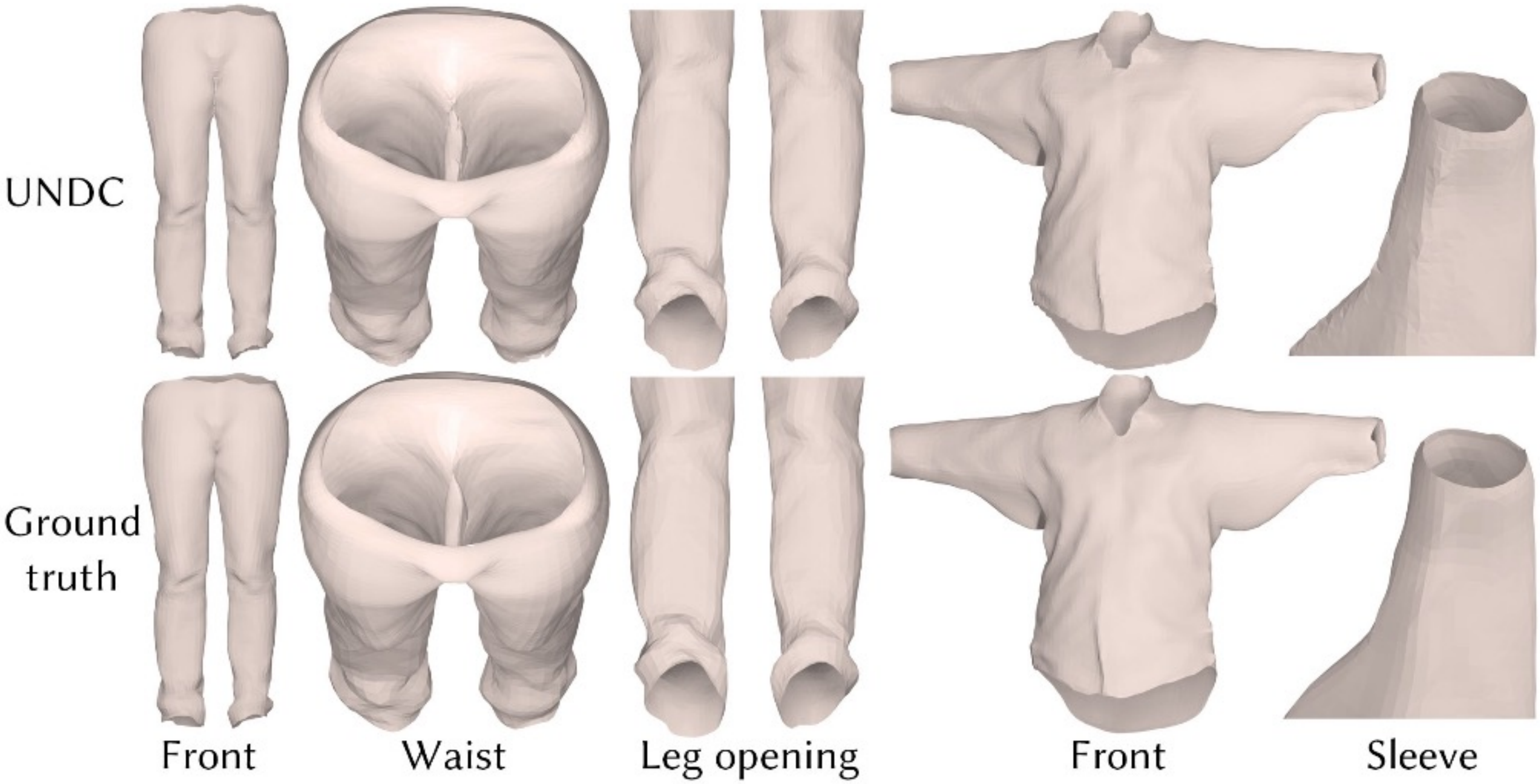
NMC vs. NDC (aside)

	NMC	NDC
Output	5 (bool) + 51 (float) per cube	1 (bool) + 3 (float) per cube
Network	3D ResNet	6-layer 3D CNN
Tessellation	Manually designed, 37 unique cases per cube	$\leq$ 1 vertex per cell; $\leq$ 1 quad per edge; see Figure 3
Output vertex count	$\approx 8\times$ MC	$\approx$ MC
Output triangle count	$\approx 8\times$ MC	$\approx$ MC
Data preparation	Sample dense point cloud in each cube; minimize chamfer distance via back propagation; complex and time-consuming	Sample only vertex signs, intersection points and normals; then apply Dual Contouring; Fast and easy to compute.

NMC vs. NDC (aside)

	NMC	NDC
Implementation	Need to consider all cube tessellation cases; difficult to implement	Could be a nice undergraduate assignment
Regularization	Need a complex regularization term for voxel input	No regularization term needed
Trainging time	(On ABC training set) 4 days per network	(Same setting) < 12 hours per network
Inference speed	(64^3 SDF input) > 1 second per shape	(Same setting) 30+ shapes per second
Inherent issues	Self-intersections, thin triangles with small angles	Non-manifold edges and vertices

UNDC results with open surfaces



Comparing predicted tessellations

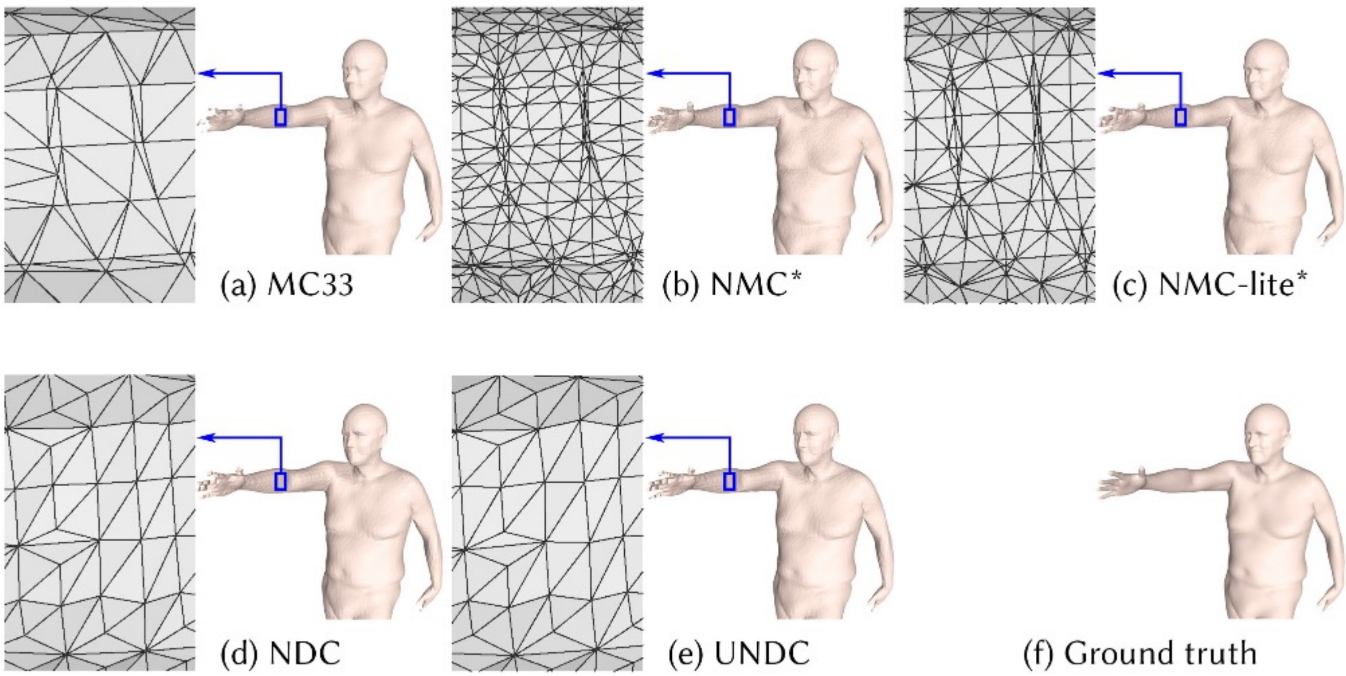
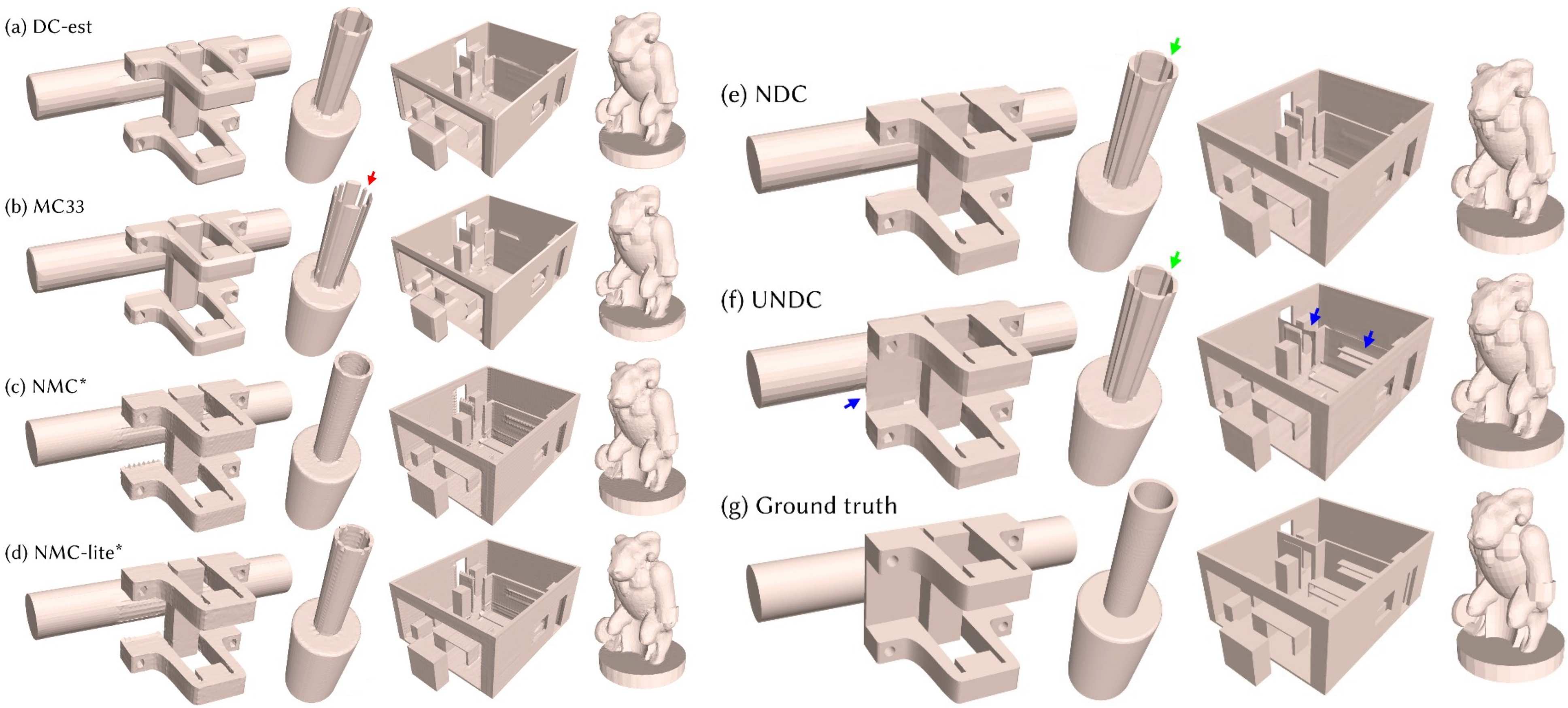
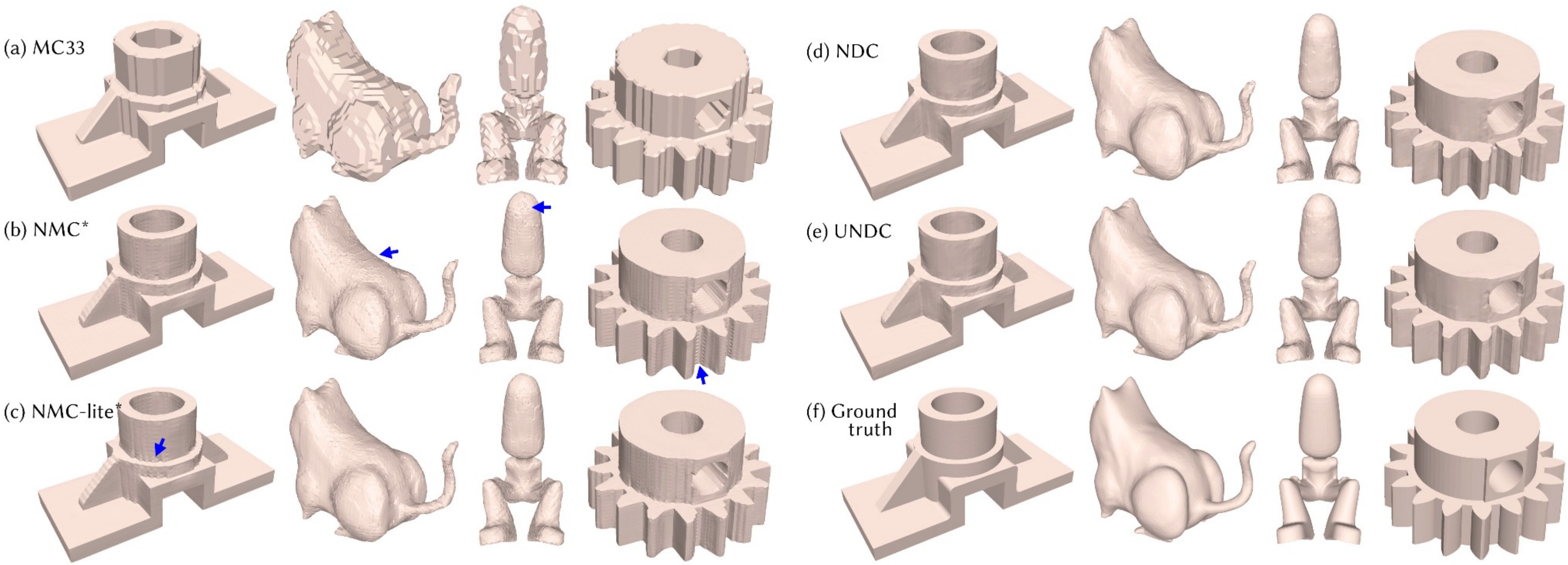


Fig. 9. Mesh reconstruction results from **SDF** grid inputs at 128^3 resolution on the **FAUST** dataset; see insets to compare triangle quality.

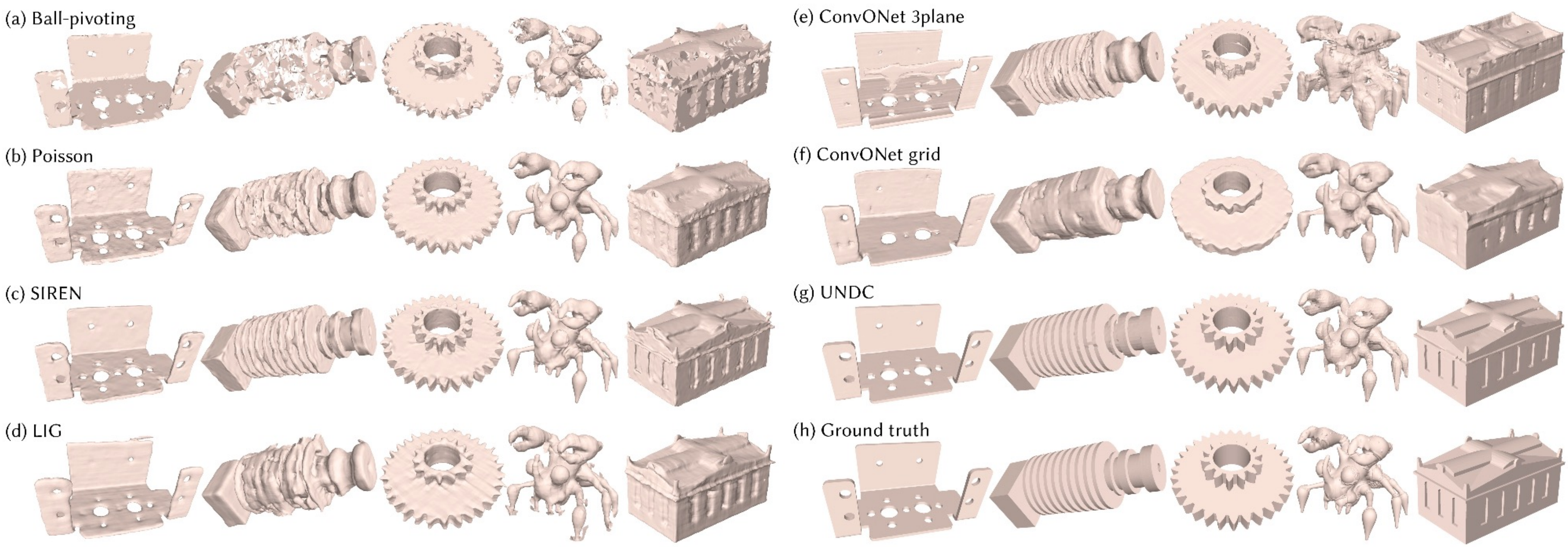
Comparison on SDF inputs



Comparison on binary voxel inputs

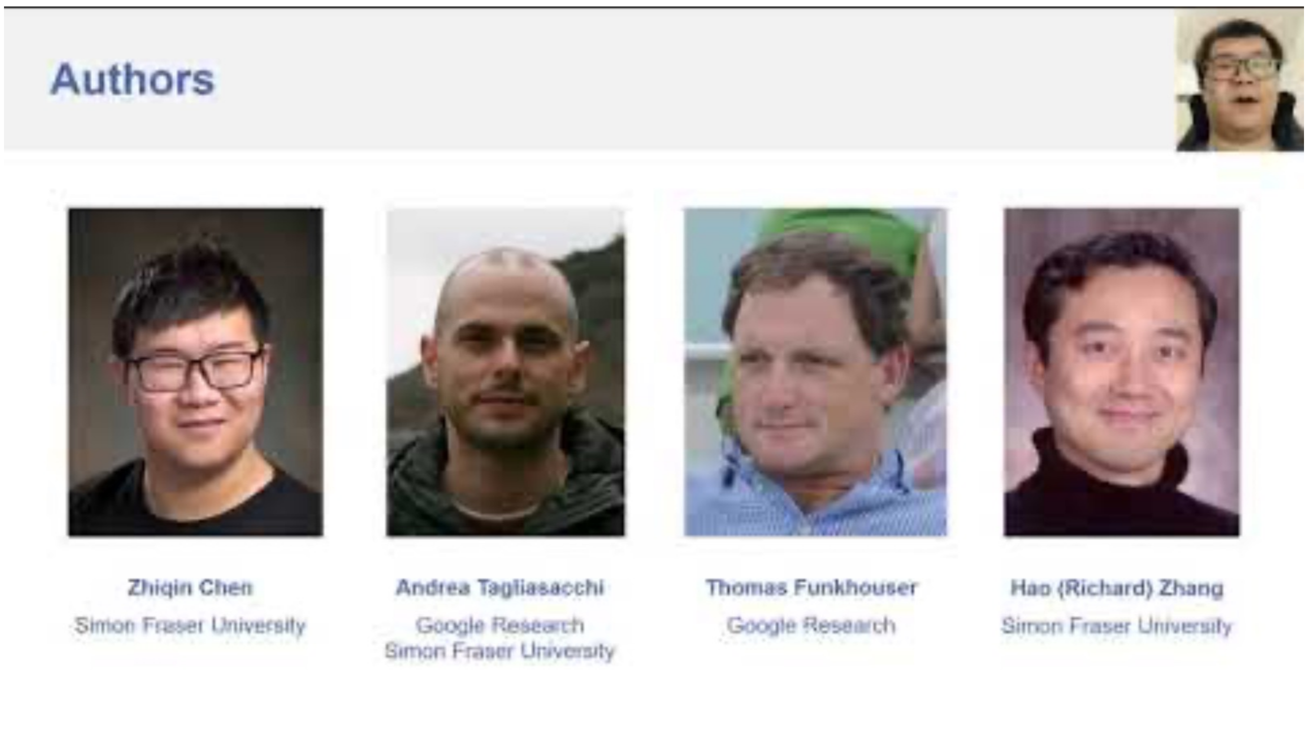


Comparison on unoriented point clouds



Youtube video on NDC presentation (aside)

https://www.youtube.com/watch?app=desktop&v=uQV9GqeKaQg



Overview of neural surface reconstruction

- Perhaps the most popular problem in geometric DL

Learning Implicit Fields for Generative Shape Modeling
Zhiqin Chen, Hao Zhang

Occupancy Networks: Learning 3D Reconstruction in Function Space
Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation
Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, Steven Lovegrove

Three CVPR 2019 papers with a combined citation counts of 8,300+

NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction

Peng Wang[†], Lingjie Liu[‡*], Yuan Liu[†], Christian Theobalt[‡], Taku Komura[†], Wenping Wang[◇*]
[†]The University of Hong Kong [‡]Max Planck Institute for Informatics
[◇]Texas A&M University
[†]{pwang3,yliu,taku}@cs.hku.hk [‡]{lliu,theobalt}@mpi-inf.mpg.de
[◇]wenping@tamu.edu

NeurIPS 2021 paper with 1,400+ citations

Overview of neural surface reconstruction

- Perhaps the most popular problem in geometric DL

Learning Implicit Fields for Generative Shape Modeling
Zhiqin Chen, Hao Zhang

Occupancy Networks: Learning 3D Reconstruction in Function Space
Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation
Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, Steven Lovegrove

Three CVPR 2019 papers with a combined citation counts of 8,300+

NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction

Peng Wang[†], Lingjie Liu[‡*], Yuan Liu[†], Christian Theobalt[‡], Taku Komura[†], Wenping Wang[◇*]
[†]The University of Hong Kong [‡]Max Planck Institute for Informatics
[◇]Texas A&M University
[†]{pwang3,yliu,taku}@cs.hku.hk [‡]{lliu,theobalt}@mpi-inf.mpg.de
[◇]wenping@tamu.edu

NeurIPS 2021 paper with 1,400+ citations

- Most produce a neural field, e.g., SDF, NeRF (for NVS)
  - NMC and NDC are exceptions
- Many inputs: point clouds (early) and images (recent)

Typical approach

Input: point cloud, image(s), even text → An Encoder → Some latent representation or code

First critical question:

What features do we encode from the inputs?

Global, local, local+global?

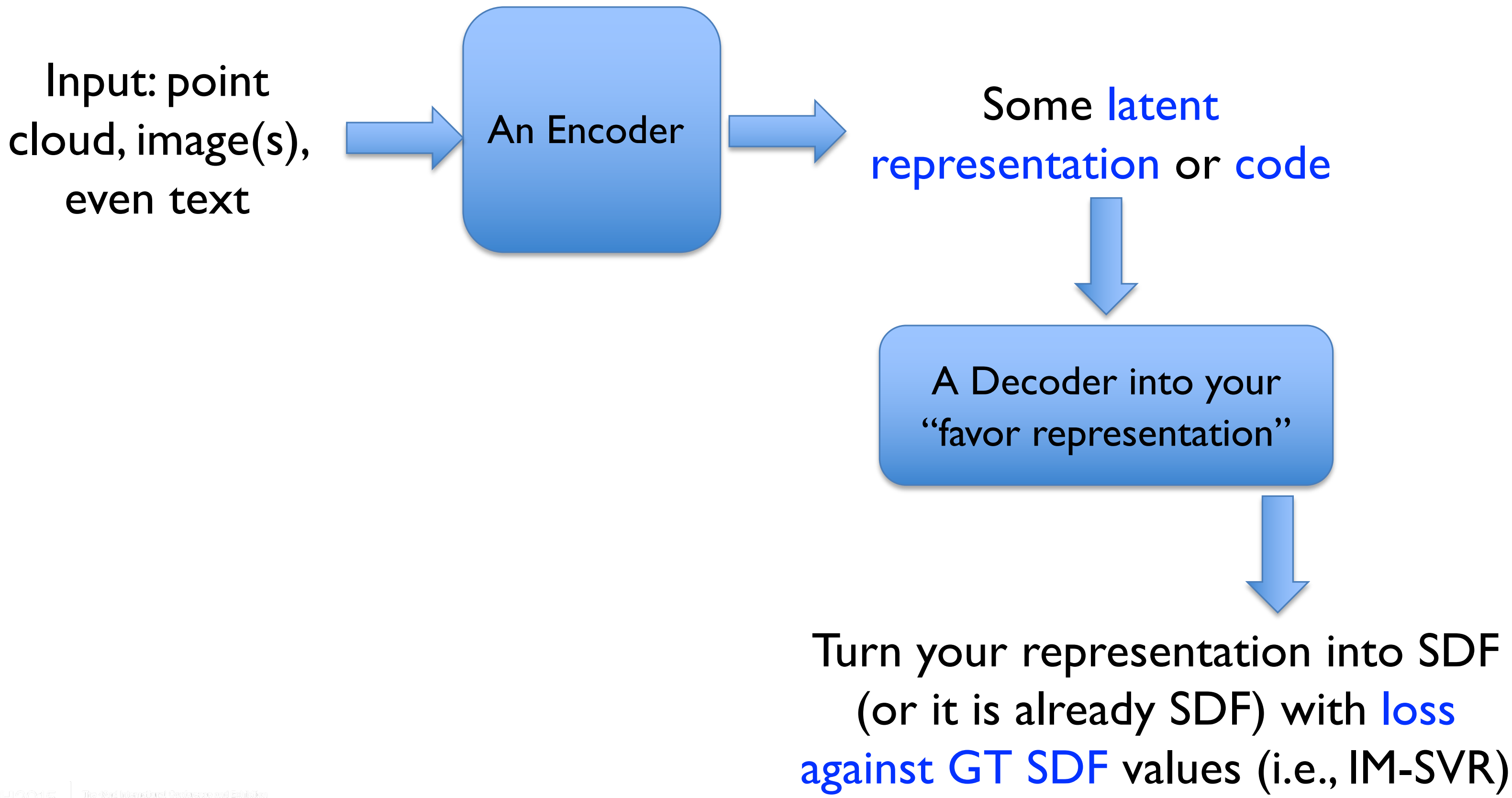
Early works: local image/point cloud features, using ResNet, VGG, PointNet, etc.

Typical approach

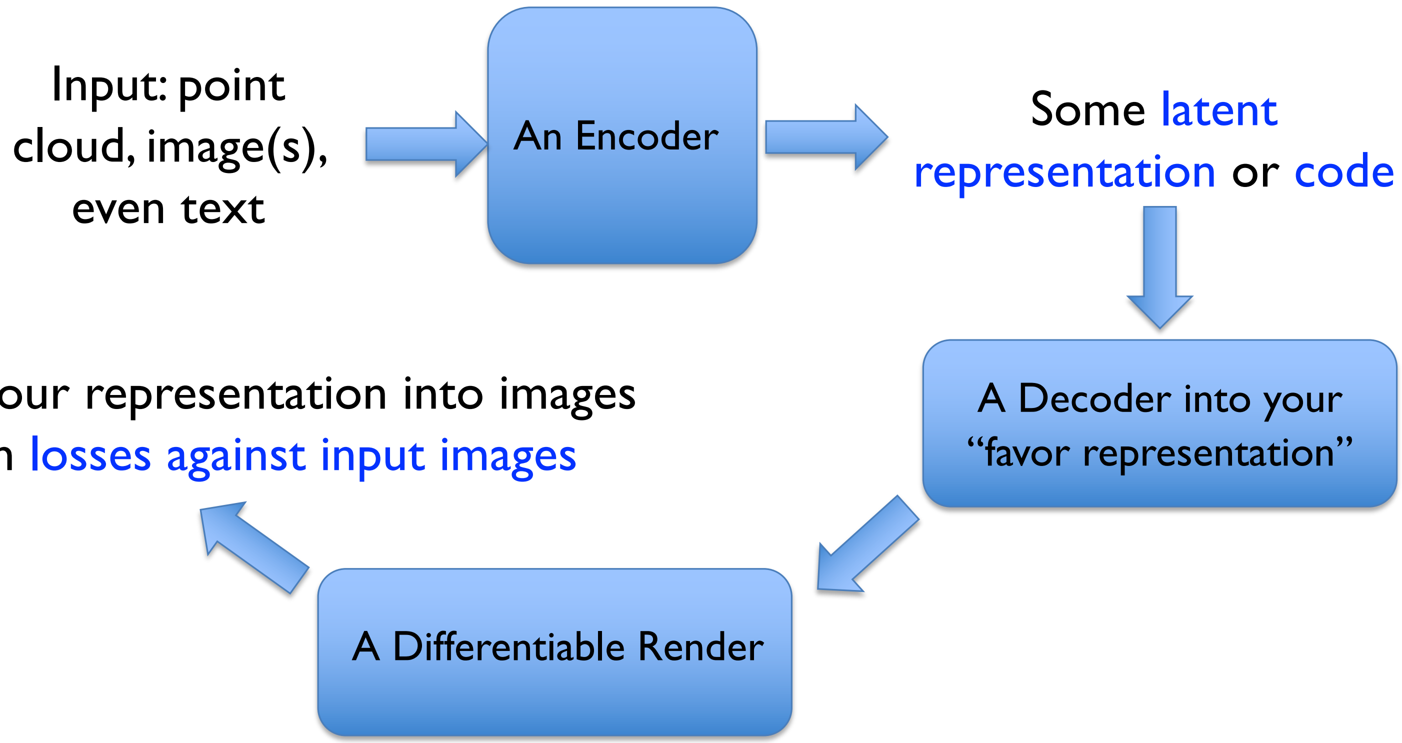
Input: point cloud, image(s), even text → An Encoder → Some latent representation or code → A Decoder into your "favor representation"

Next critical question:
What is this "representation?"

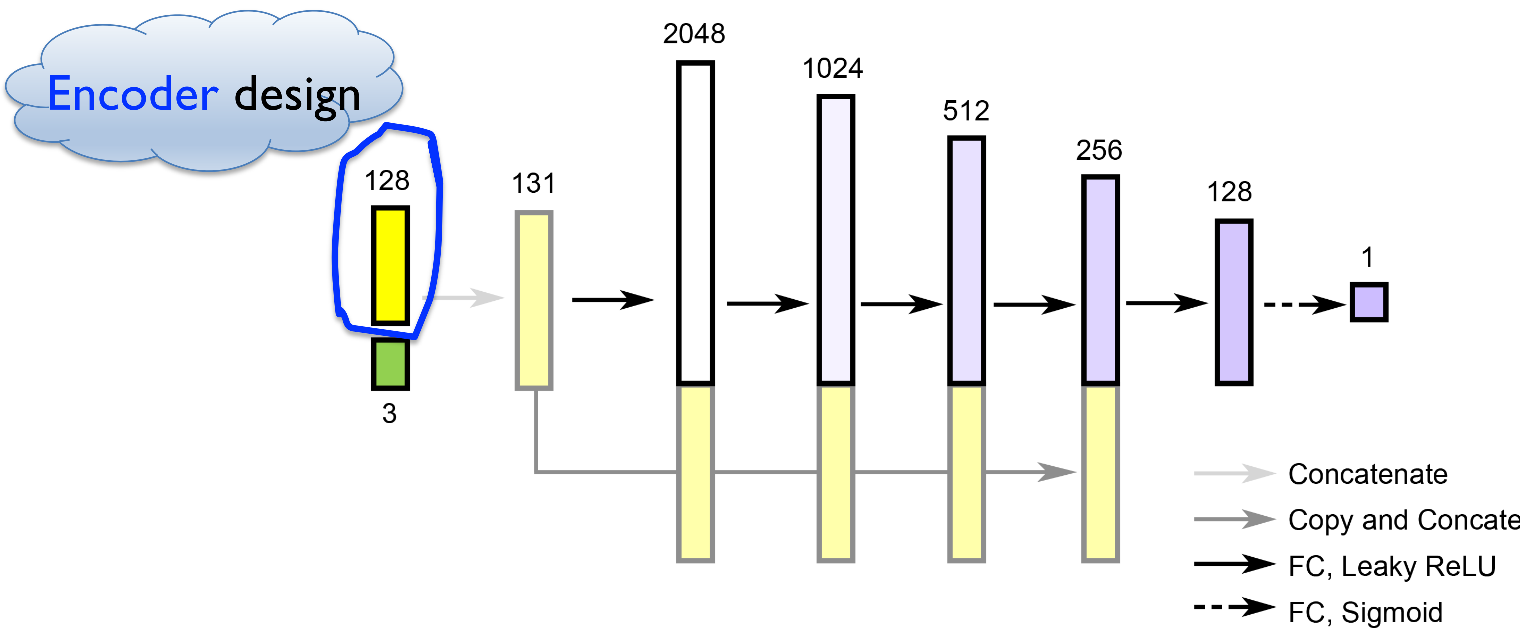
Typical approach



Typical approach



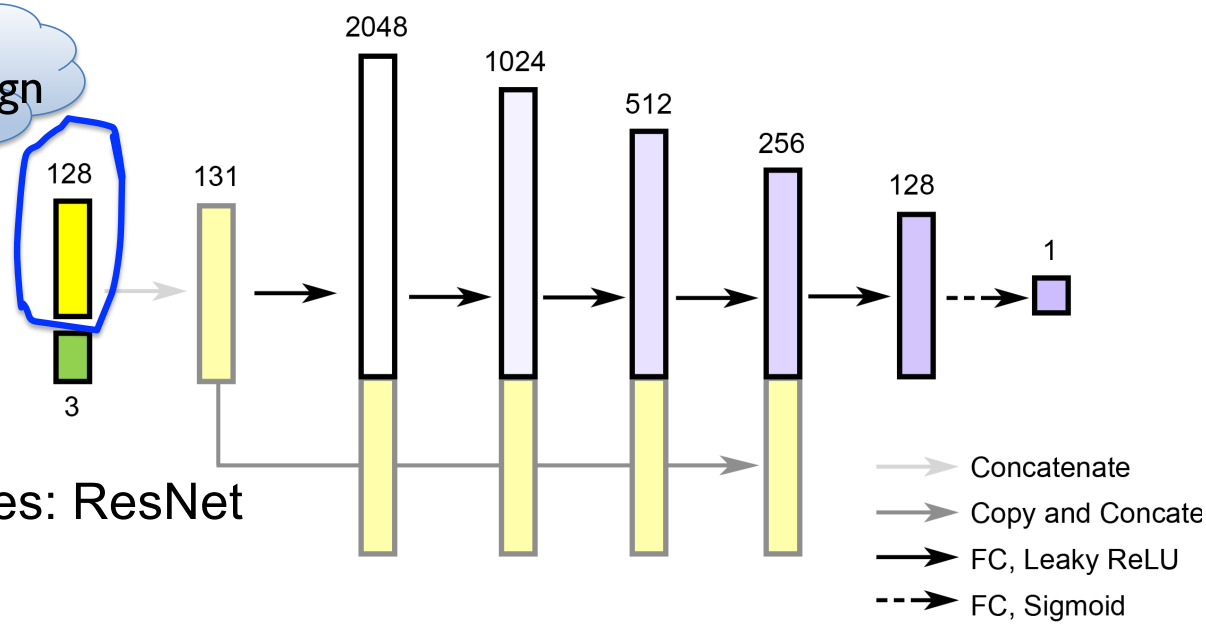
Examples: start from IM-Net



IM-Net [Chen and Zhang, CVPR 2019]

IM-SVR (single-view reconstruction)

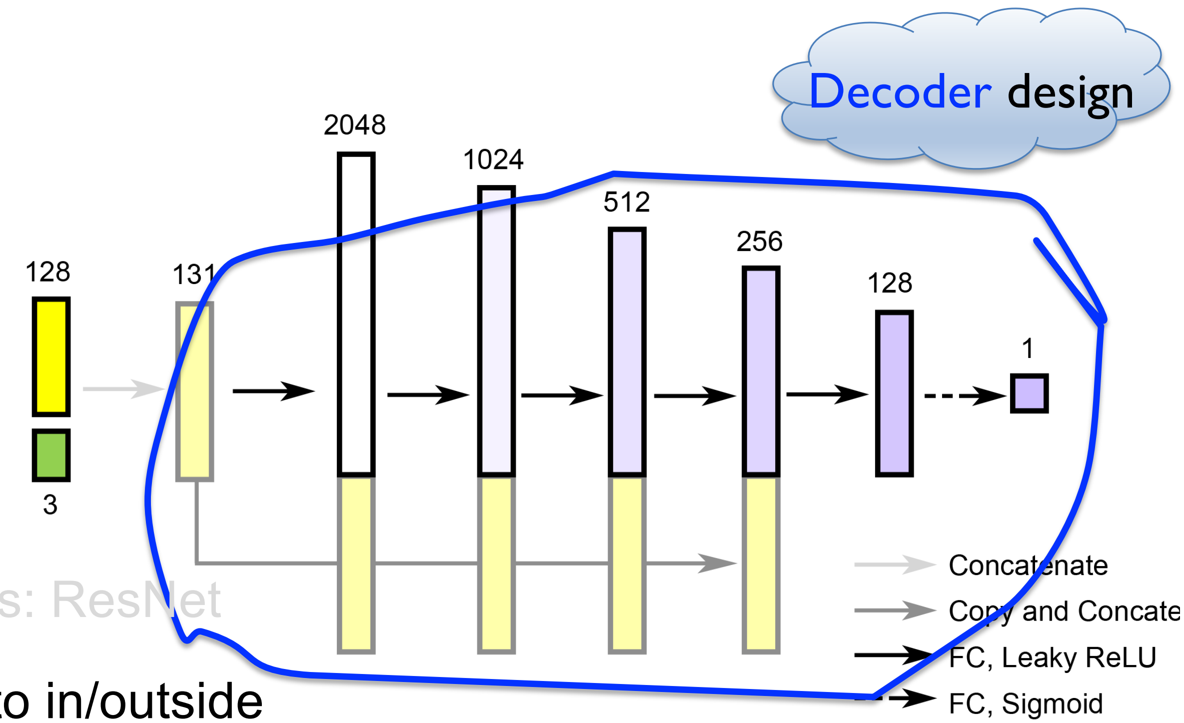
Encoder design



- Global features: ResNet

IM-Net [Chen and Zhang, CVPR 2019]

IM-SVR (single-view reconstruction)



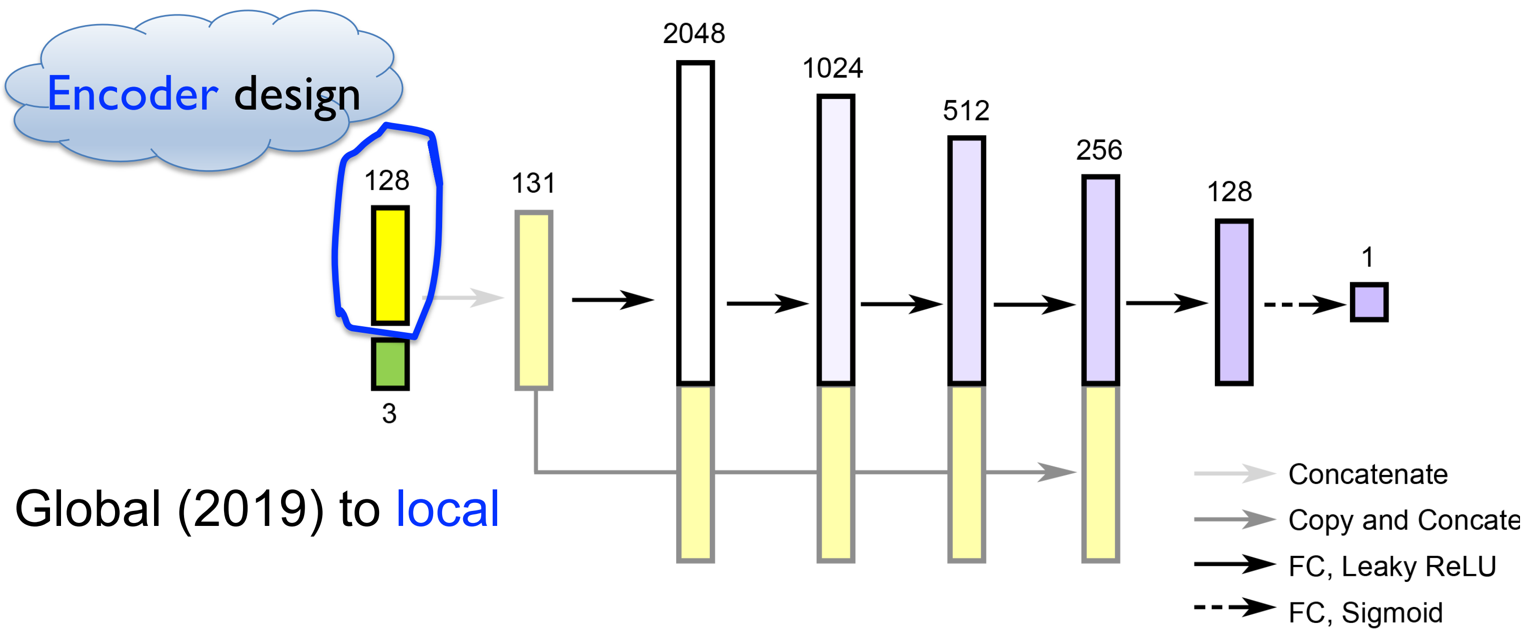
❖ Global features: ResNet

❖ IM-decoder into in/outside

❖ Marching Cubes to mesh

IM-Net [Chen and Zhang, CVPR 2019]

An evolution of neural implicit (2019 - now)

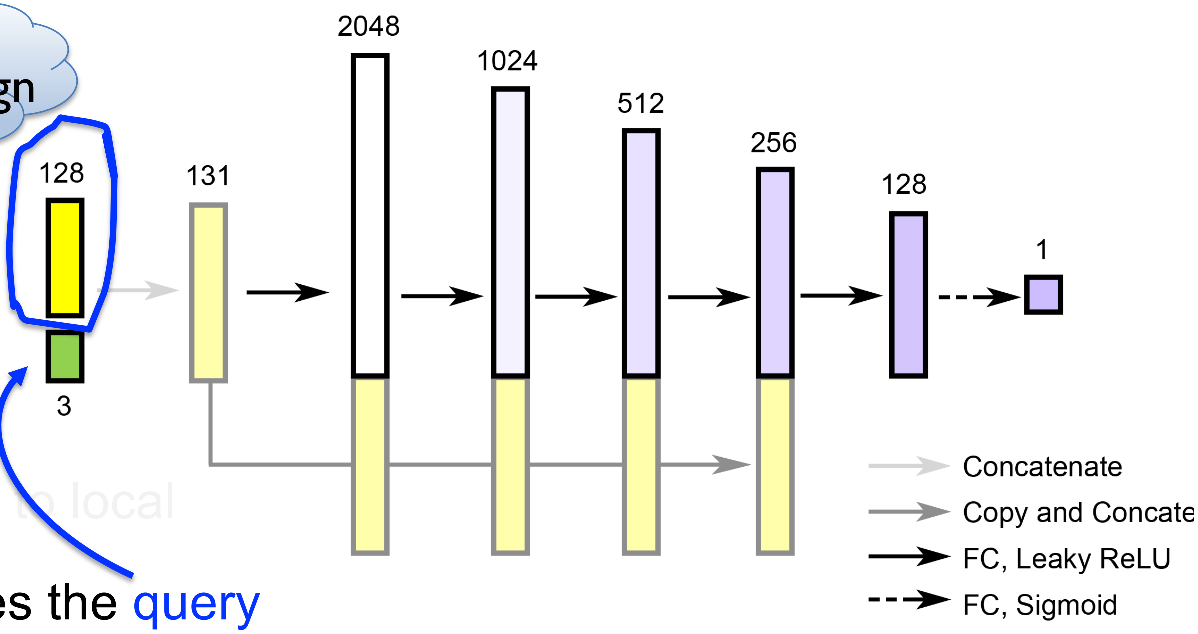


❖ Global (2019) to local

IM-Net [Chen and Zhang, CVPR 2019]

An evolution of neural implicit (2019 - now)

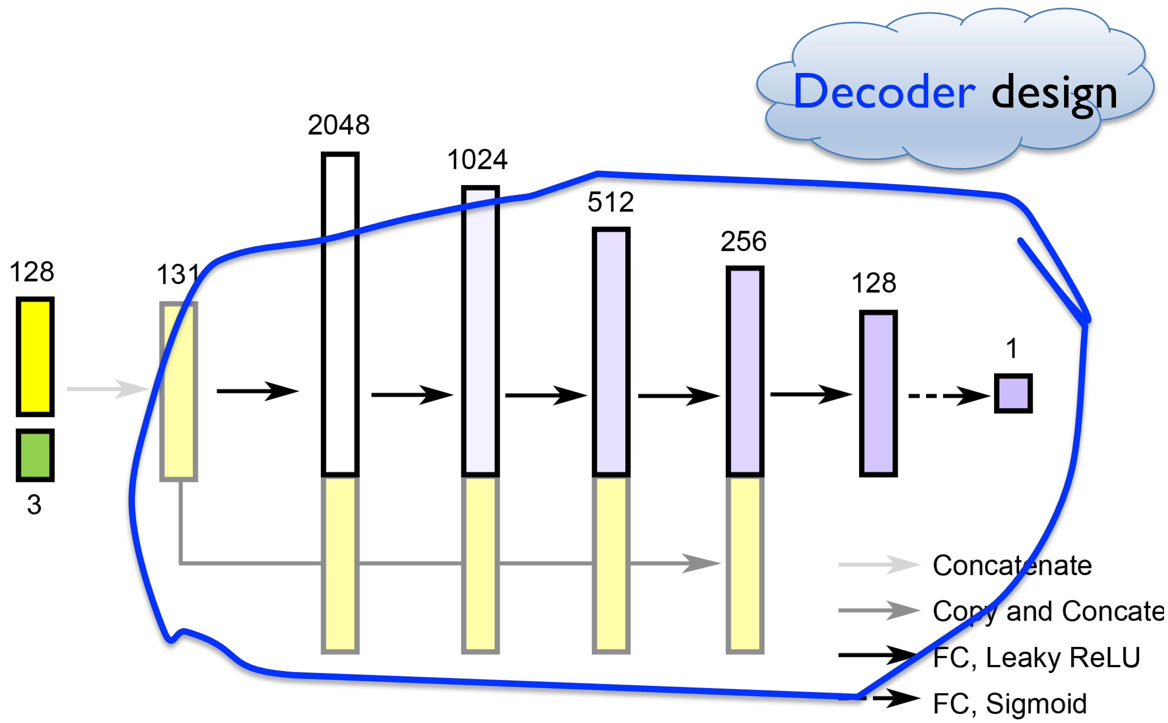
Encoder design



- Global (2019) to local
- What role does the query point (coordinate) play?

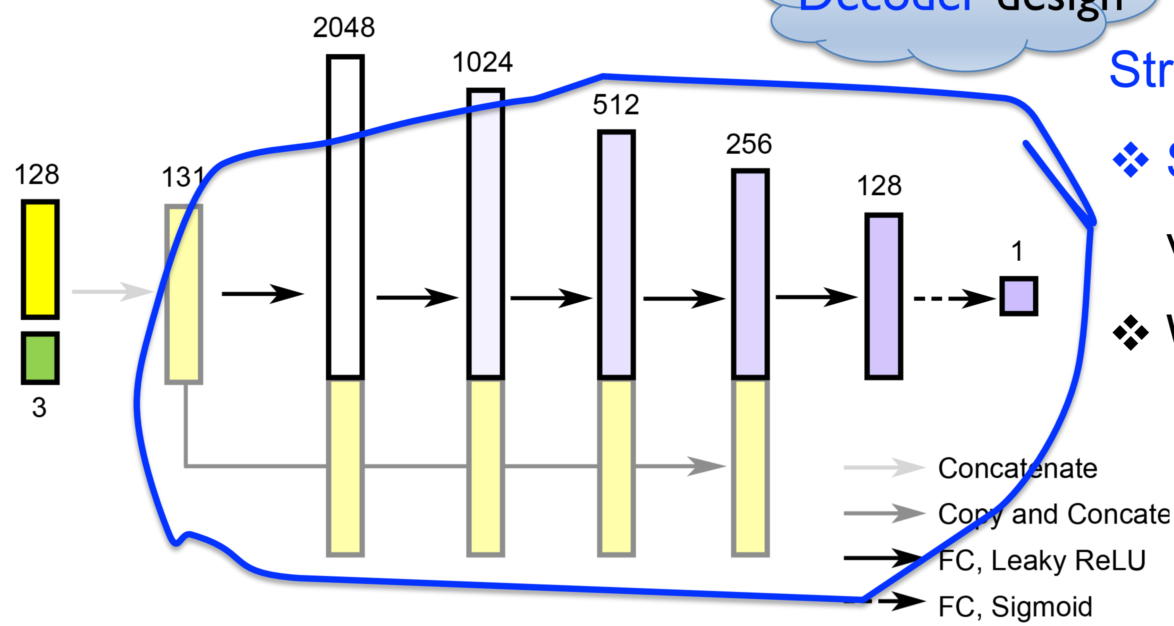
IM-Net [Chen and Zhang, CVPR 2019]

An evolution of neural implicit (2019 - now)



IM-Net [Chen and Zhang, CVPR 2019]

An evolution of neural implicit (2019 - now)



Decoder design

Structured models

- Semantic parts vs. primitives?
- What primitives?

IM-Net [Chen and Zhang, CVPR 2019]

An evolution of neural implicit (aside)

IM-Net, OCC-Net, DeepSDF, CVPR 2019

Encoder design

Global + local feature encoding

- PIFu - 2019
- Deep implicit surface network (DISN) – 2020
- Deep local shape - 2020
- Local implicit grid representations - 2020
- Local deep implicit functions (LDIF) – 2020
- PatchNets – 2020
- ❖ D^{2}IM-Net – 2022, …

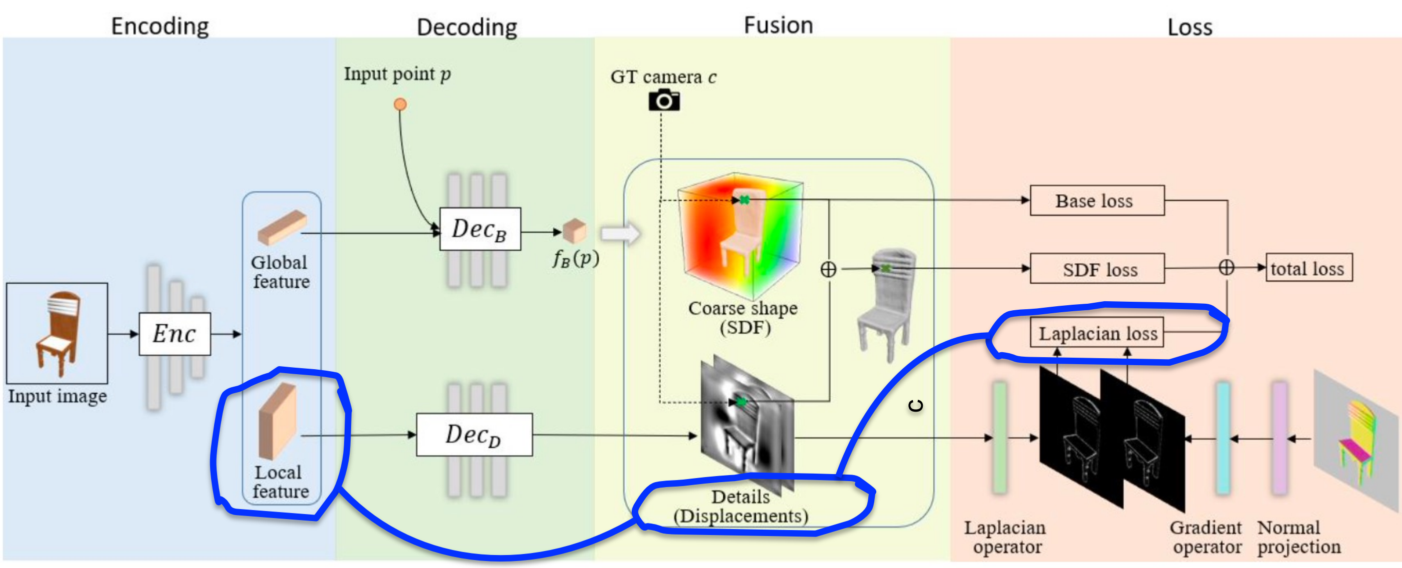
Learning to recover shape/surface details

- Encode both global and local features for single-view reconstruction
- Generally, only global encoding leads to coarse/blurry shapes

Recover shape/surface details (aside)

- Encode both global and local features for single-view reconstruction
- Generally, only global encoding leads to coarse/blurry shapes

D²IM-Net: Learning Detail Disentangled Implicit Fields from Single Images [Li and Zhang CVPR 2021]



An evolution of neural implicit (aside)

Encoder design

IM-Net, OCC-Net, DeepSDF, CVPR 2019

Global + local feature encoding

Query-specific feature encoding

- ❖ Convolutional occupancy network (ConvONet) – 2020
- IF-Net – 2020
- Point2Surf – 2020
- AIR-Net – 2021
- SA-ConvONet - 2021
- POCO – 2022
- 3DILG – 2022
- ❖ ARO-Net via Anchored Radial Observations - 2023

Why encode query-specific shape features?

- Network is trained to predict occupancy/SDF *at* a query point
- Local feature encoding is better than global, learning features with respect to the query point is even better

Query-specific, contextual, shape features

[Wang et al. CVPR 2023]

ARO-Net: Learning Implicit Fields from Anchored Radial Observations

Yizhi Wang[1,2]*, Zeyu Huang[1]*, Ariel Shamir[3], Hui Huang[1], Hao Zhang[2], Ruizhen Hu[1]†
[1]Shenzhen University [2]Simon Fraser University [3]Reichman University

Encode features with respect to a fixed set of anchors

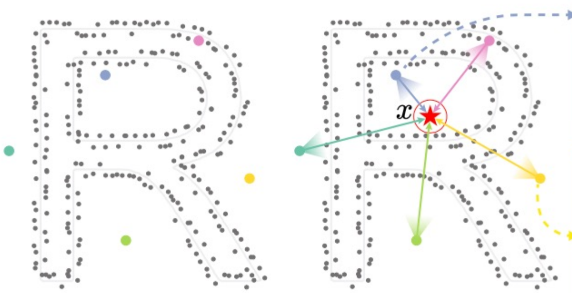
Task: 3D reconstruction from sparse point clouds

Query-specific, contextual, shape features

[Wang et al. CVPR 2023]

ARO-Net: Learning Implicit Fields from Anchored Radial Observations

Yizhi Wang[1,2]*, Zeyu Huang[1]*, Ariel Shamir[3], Hui Huang[1], Hao Zhang[2], Ruizhen Hu[1]†
[1]Shenzhen University [2]Simon Fraser University [3]Reichman University



Key idea: encode query-specific and contextual (local-to-global) features by making observations from the anchors towards the query point

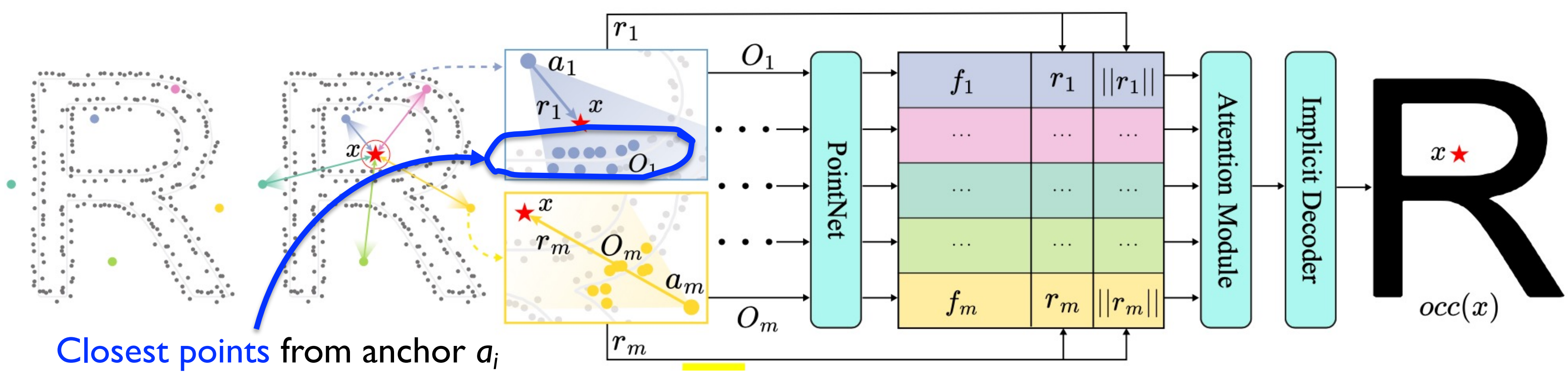
"What does the shape look like from the perspectives of the anchors towards the query point?" — from a perceptual point of view

Query-specific, contextual, shape features

[Wang et al. CVPR 2023]

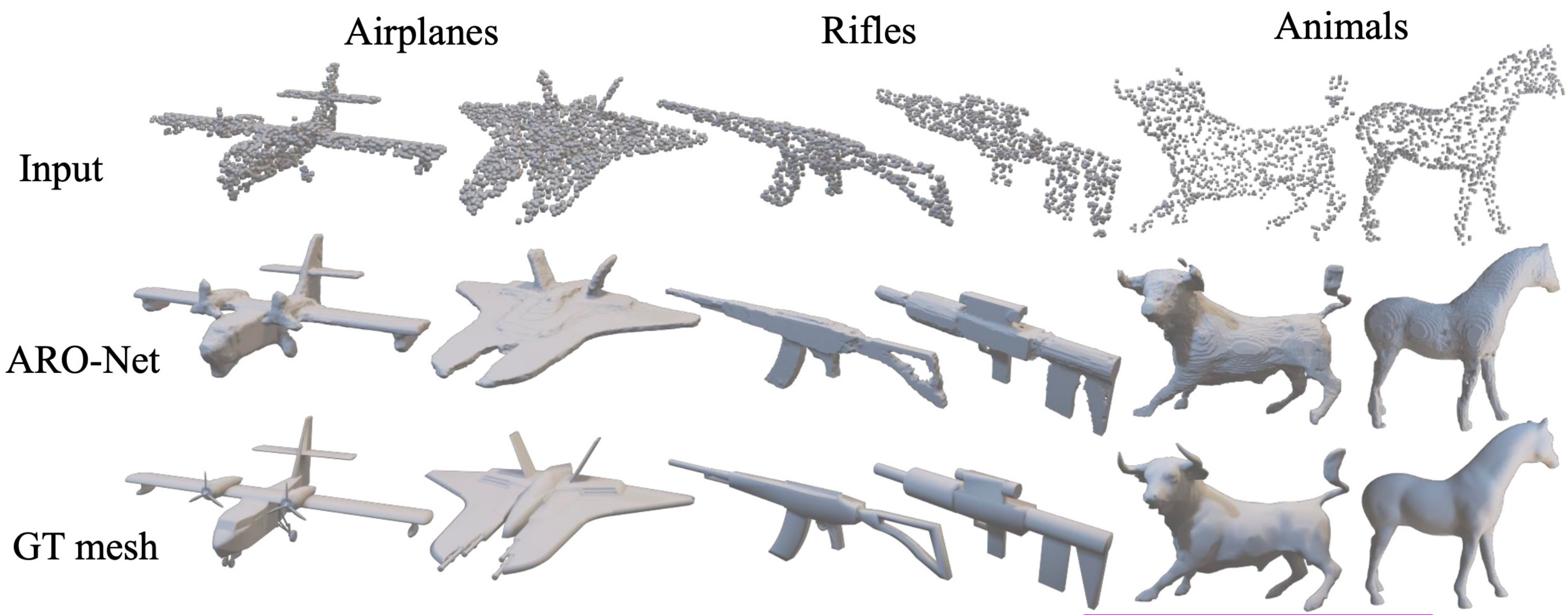
ARO-Net: Learning Implicit Fields from Anchored Radial Observations

Yizhi Wang[1,2]*, Zeyu Huang[1]*, Ariel Shamir[3], Hui Huang[1], Hao Zhang[2], Ruizhen Hu[1]†
[1]Shenzhen University [2]Simon Fraser University [3]Reichman University



Closest points from anchor a_i

Generalizability and quality of reconstruction

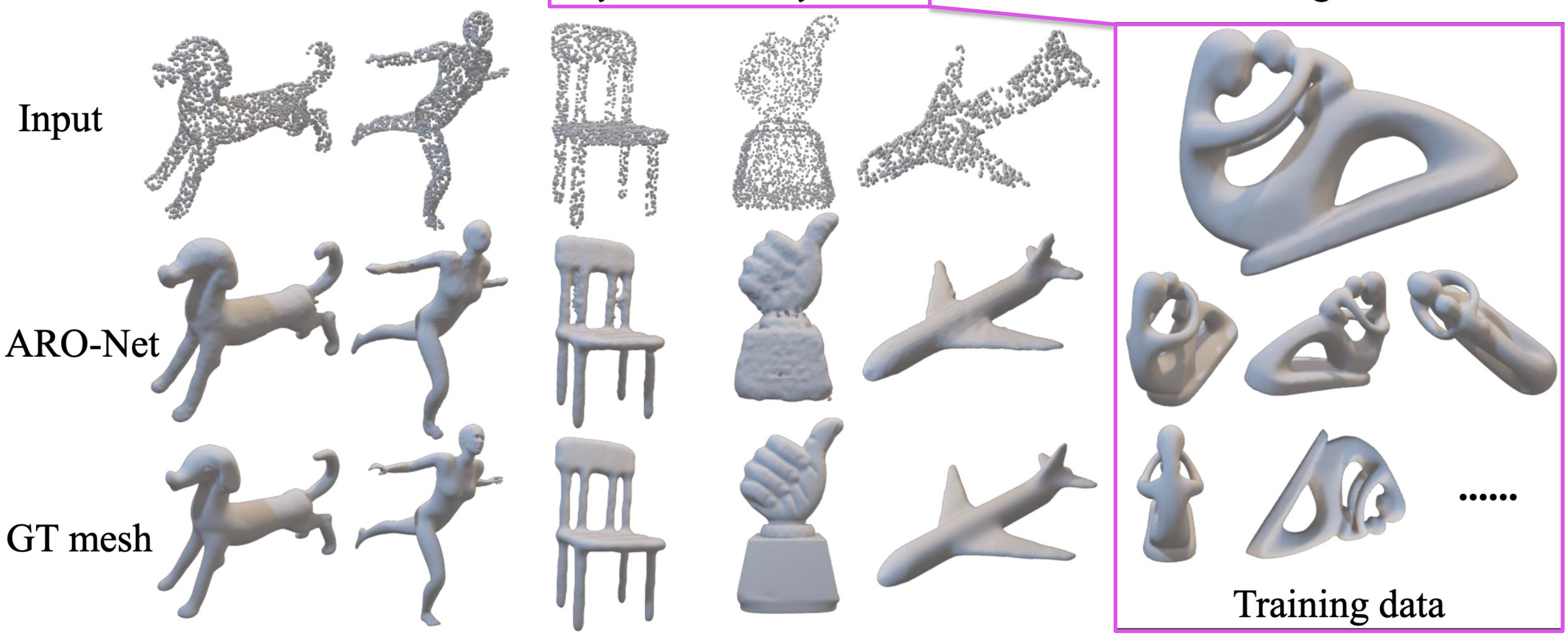


3D Reconstruction from sparse point clouds by ARO-Net *trained on 4K chairs*

[Wang et al. CVPR 2023]

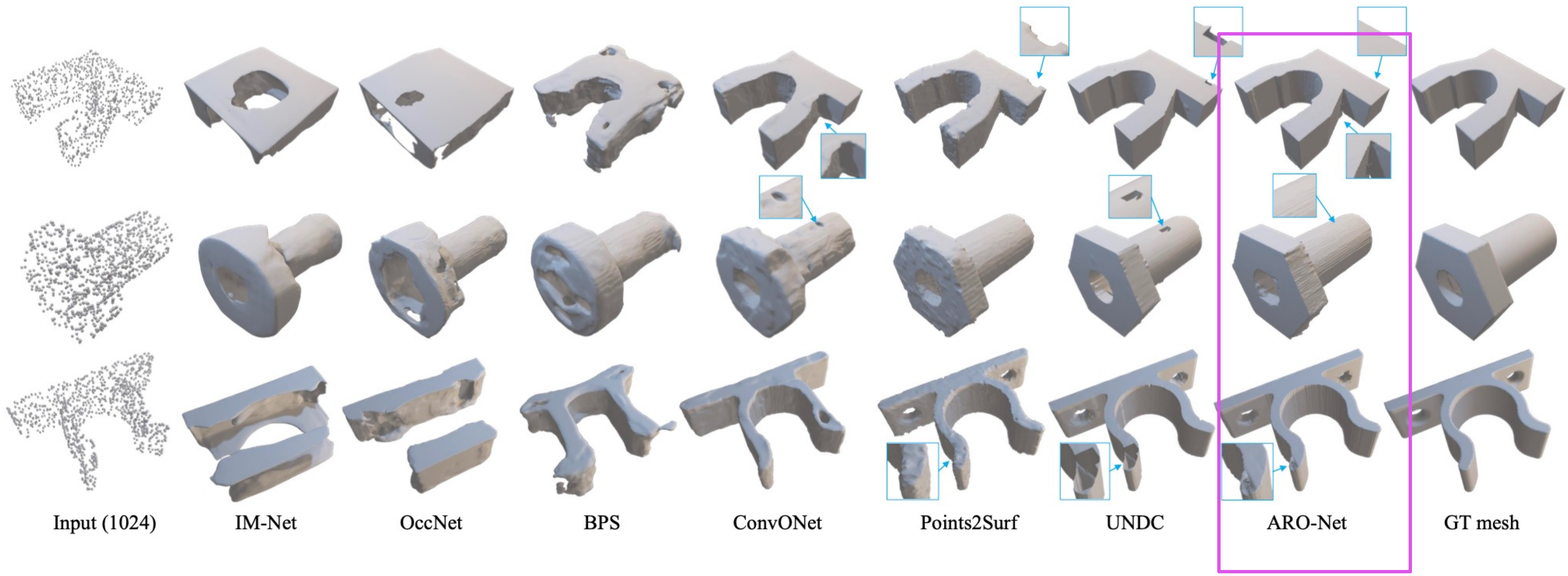
Generalizability and quality of reconstruction

ARO-Net trained on *only the Fertility model* with rotation and scaling



[Wang et al. CVPR 2023]

Generalizability and quality of reconstruction



[Wang et al. CVPR 2023]

An evolution of neural implicit (2019 - now)

Encoder design

IM-Net, OCC-Net, DeepSDF, CVPR 2019

Decoder design

Global + local feature encoding

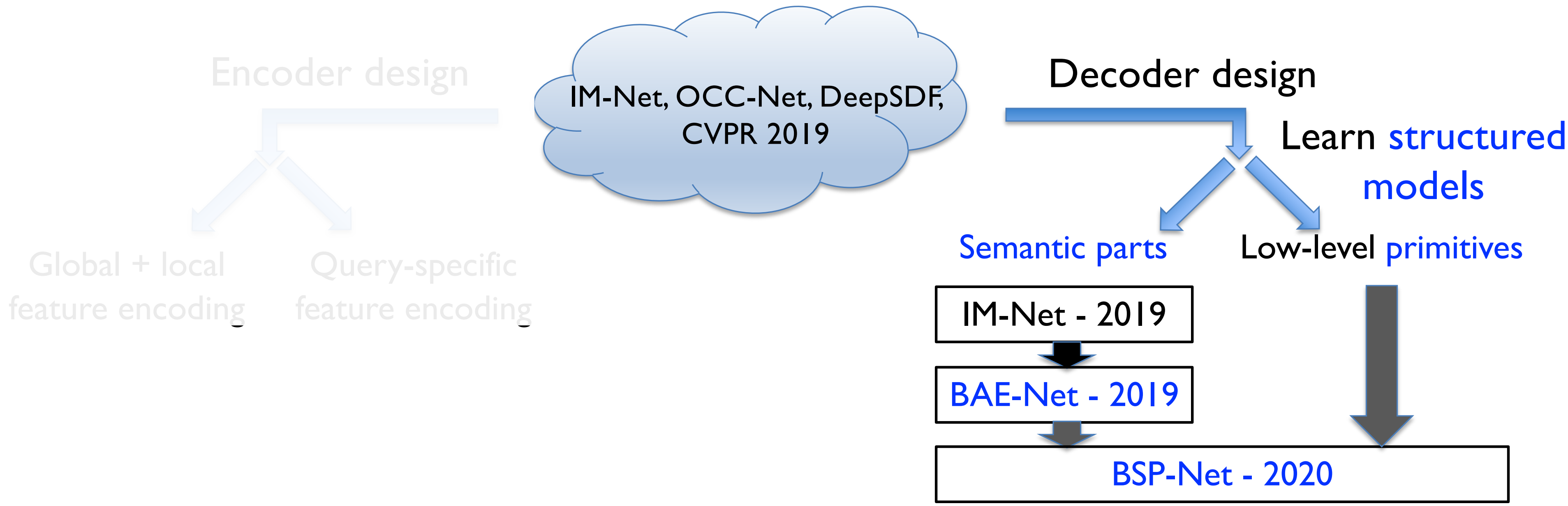
Query-specific feature encoding

Learn structured models

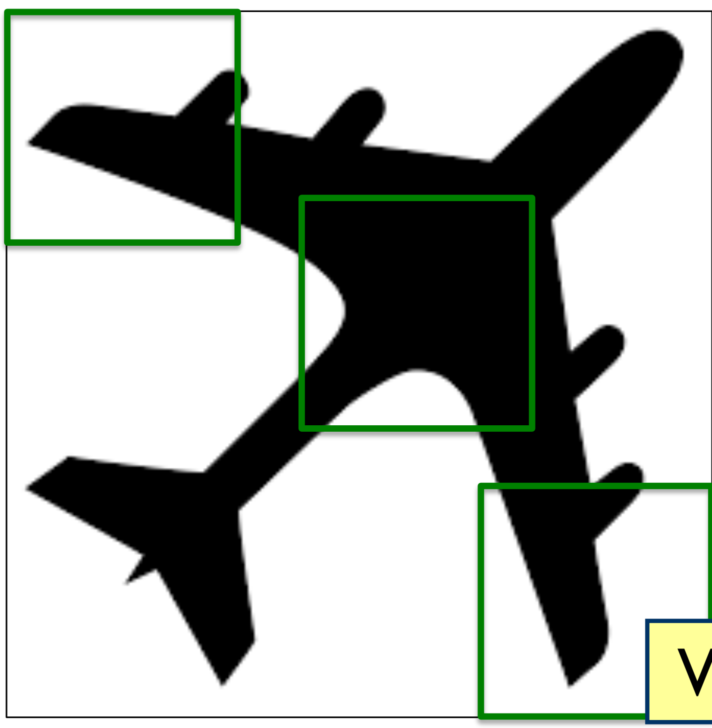
Semantic parts

Low-level primitives

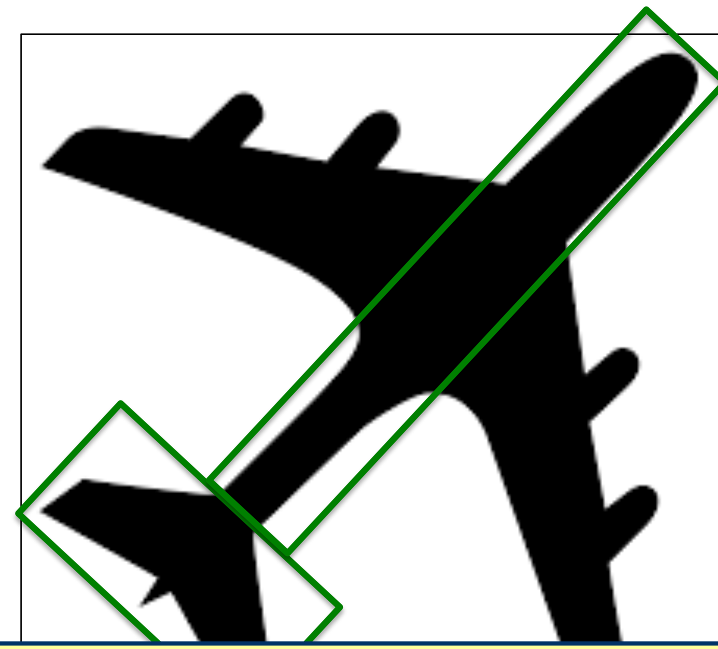
An evolution of neural implicit (2019 - now)



Can IM-NET learn shape parts (the right boxes)



Wrong boxes?

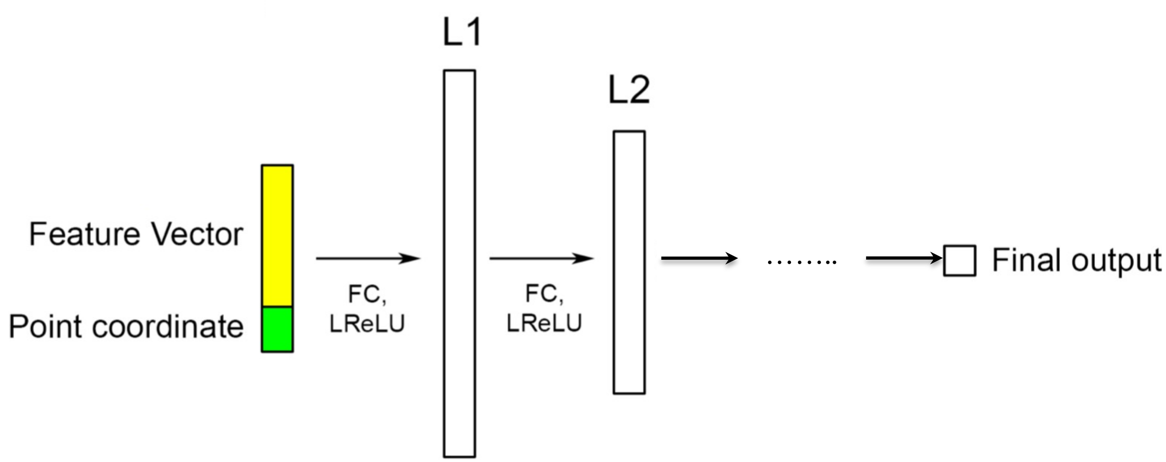


The right boxes

We care about shape parts = structures

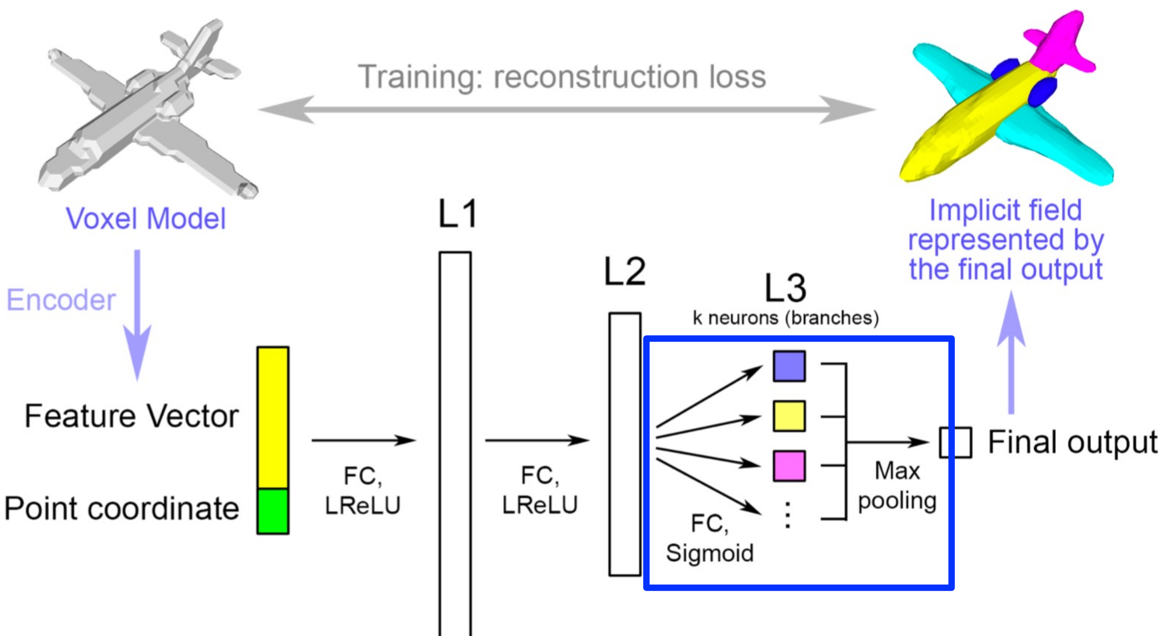
Can IM-NET learn shape parts?

❖ Original IM-Net trained with reconstruction loss



Key: add a branching layer

❖ Same reconstruction loss, with no part label as supervision

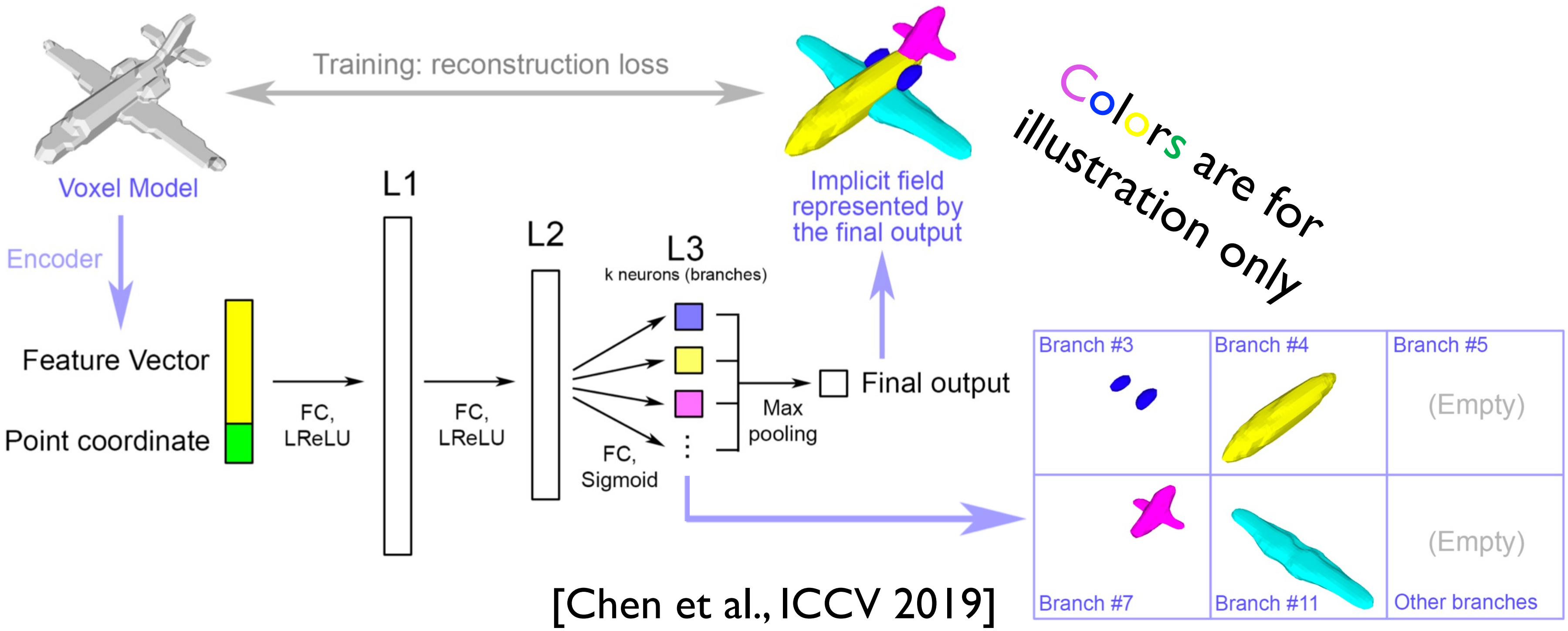


[Chen et al., ICCV 2019]

Branching in last layer
is the only change!

Branched IM-NET to learn shape parts

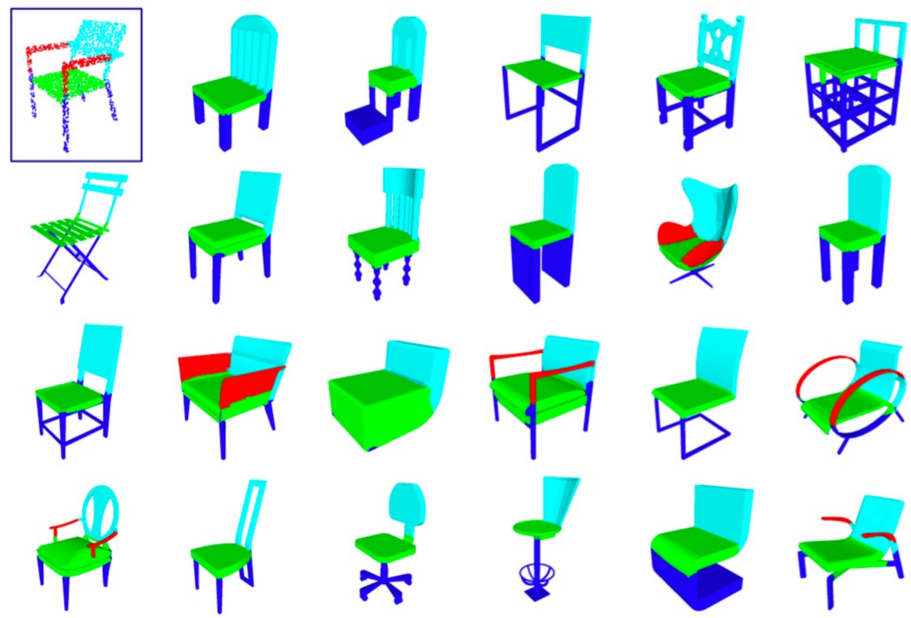
❖ Same reconstruction loss, with no part label as supervision



[Chen et al., ICCV 2019]

Unsupervised and 1-shot co-segmentation

- ❖ Unsupervised BAE-NET = Branched Autoencoder
- ❖ Repeatedly train on a set of unlabeled shapes with only shape reconstruction loss
- ❖ One-shot learning with just 1, or 2, or 3 labeled shapes, via label reconstruction loss

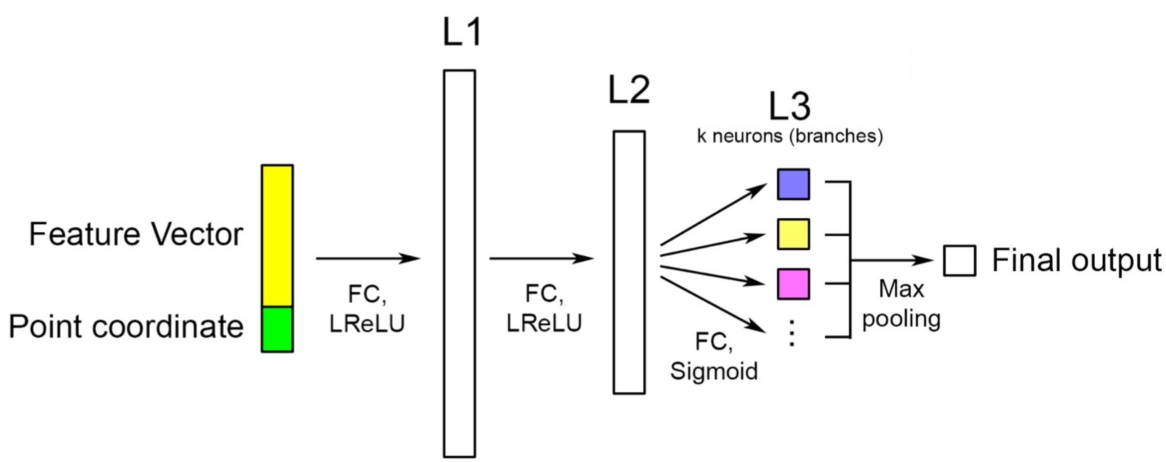


One-shot learning by BAE-NET on chair co-segmentation

[Chen et al., ICCV 2019]

Why does this work?

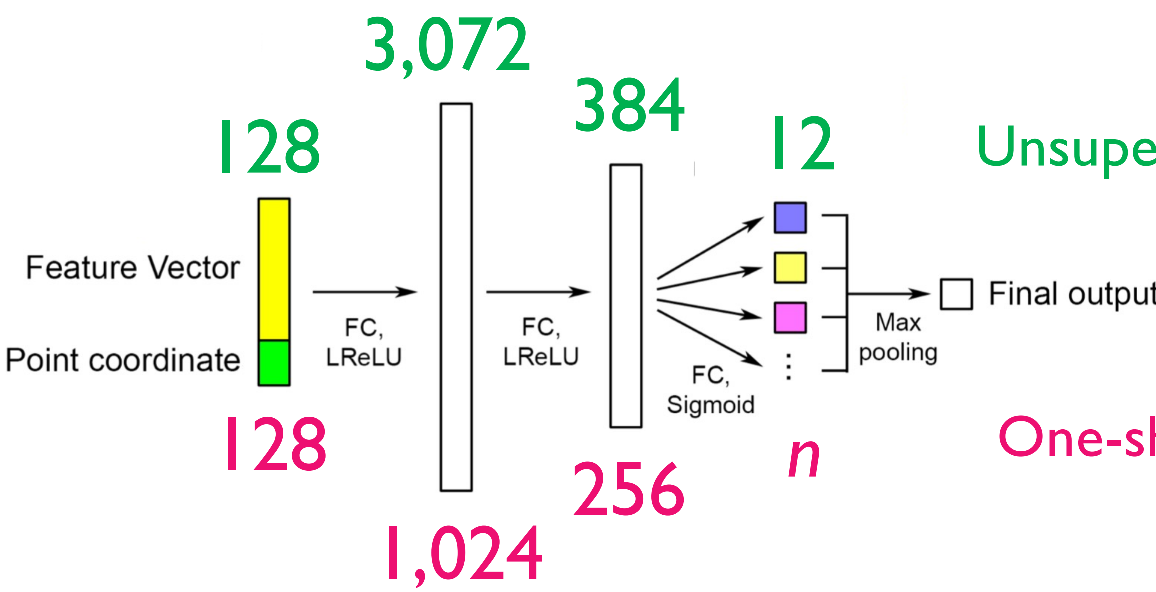
❖ Exploit the structure of small implicit field network (IM-Net)



[Chen et al., ICCV 2019]

Why does this work?

- Exploit the structure of small implicit field network (IM-Net)
- Small = shallow network (3 layers only) and few neurons



Unsupervised learning

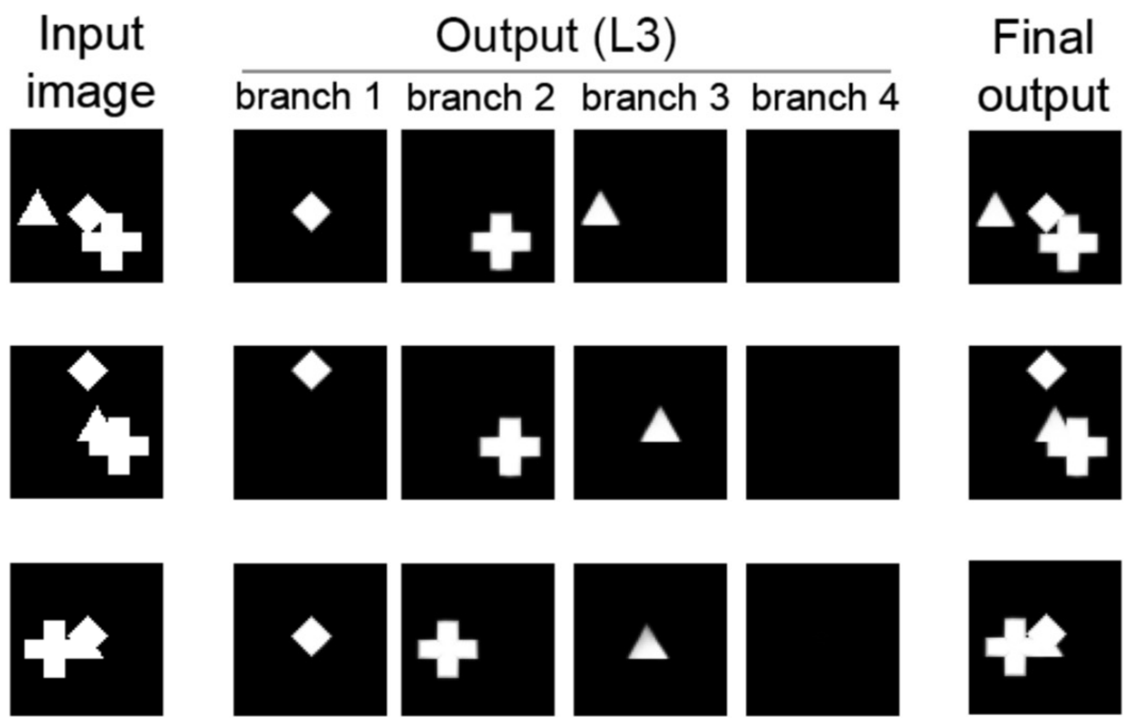
One-shot learning

[Chen et al., ICCV 2019]

Consequence of a compact network

❖ Must find compact and hence consistent reps ⇒ parts

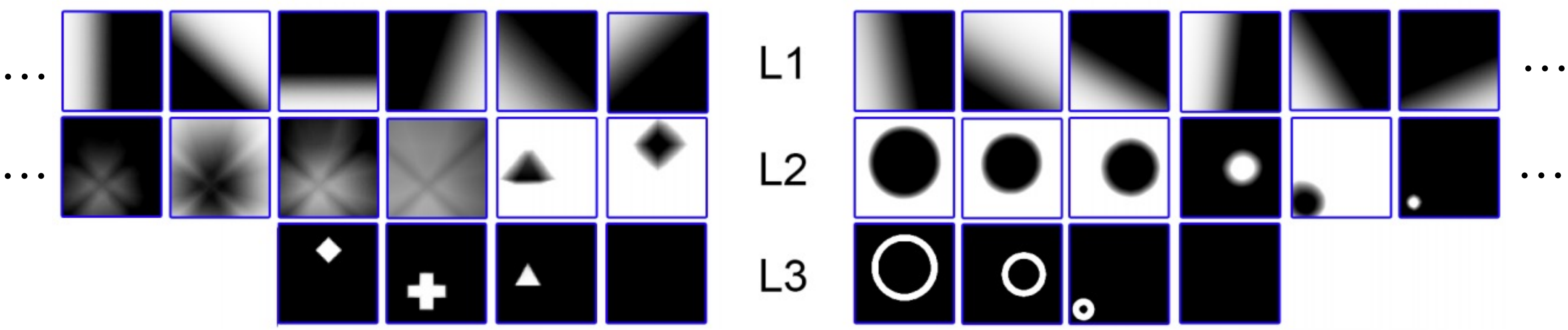
Trained on "Elements", synthetic 2D pattern images consisting of a cross, a ▲ , and a ◆ , randomly placed.



Interpret what each layer learns …

❖ A closer look at neural activation maps in each layer

Layer 1 learn linear gradient fields

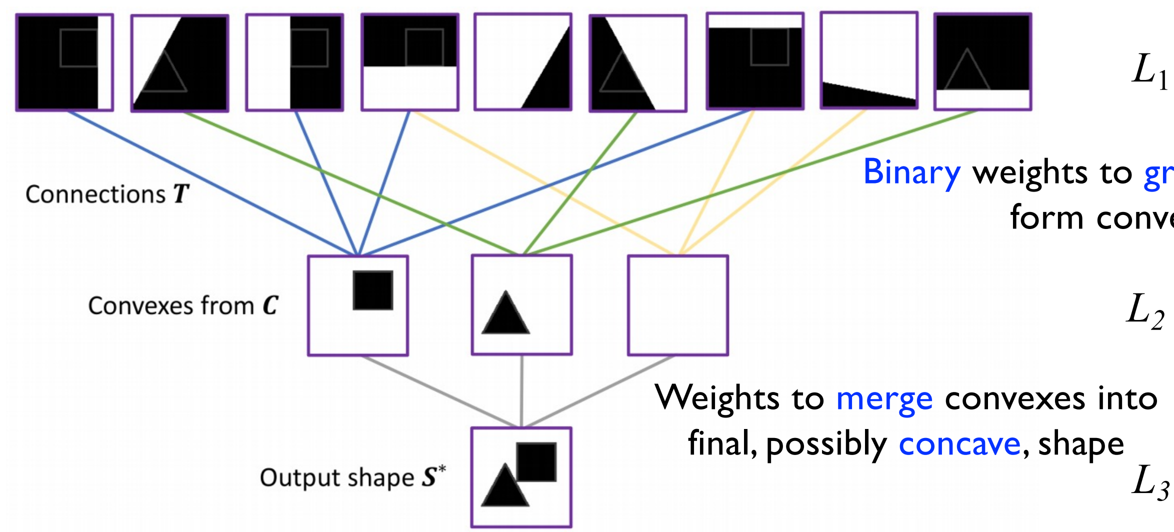


Trained on the "Elements" images

Trained on the "Three rings" images

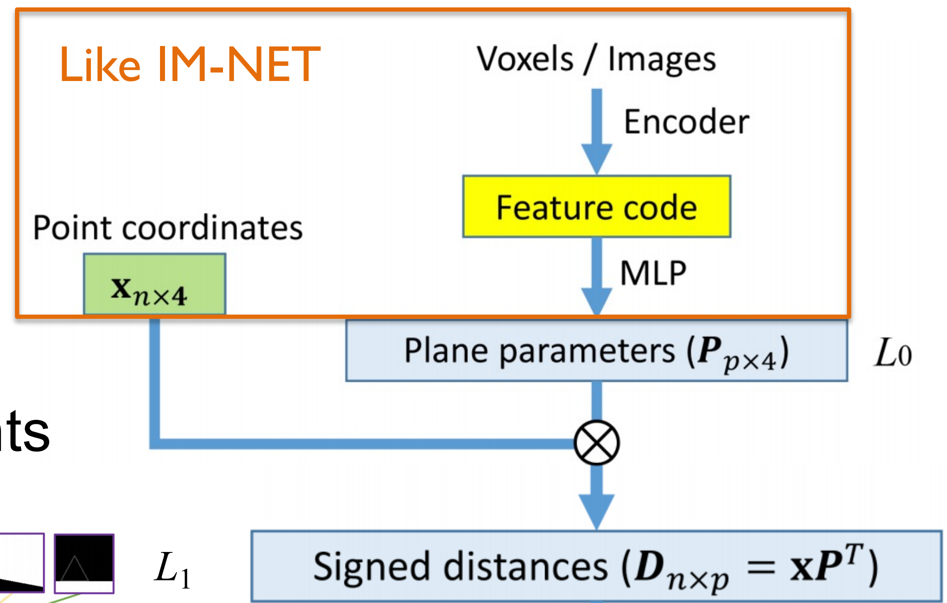
Make this "interpretability" explicit

- Let neurons in L_1 layer represent planes explicitly
- Layer L_2 combines planes into convex shape primitives

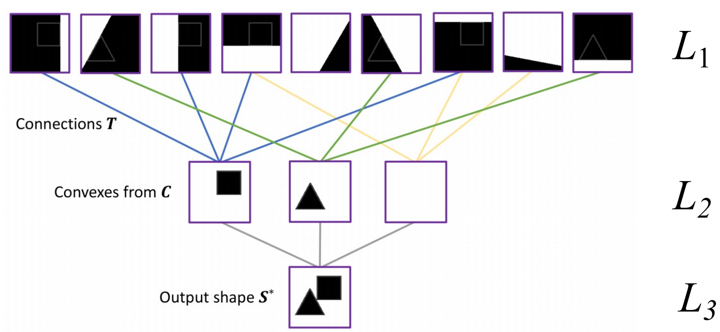


BSP-NET: another neural implicit model

- I/O the same as IM-NET, with same reconstruction loss
- NEW: shapes are formed via binary space partitioning (BSP)
- Planes defined by learned weights



Each neuron a BSP



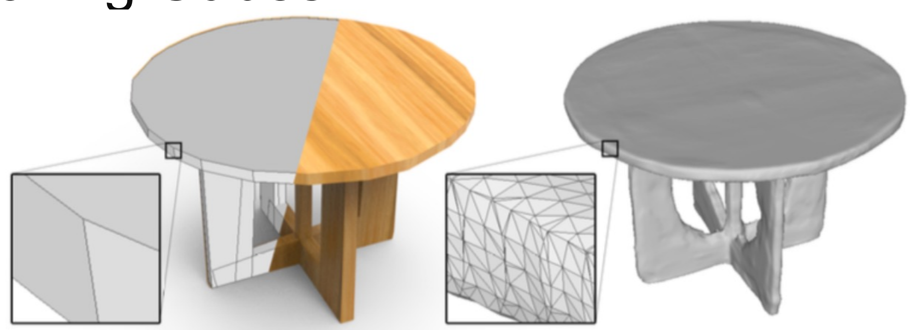
[Chen et al., CVPR 2020]
Best Student Paper Award

BSP-NET: directly produce compact meshes

- The network learns how to best reconstruct the set of training 3D shapes using *N* (e.g., 4,096) planes

BSP-NET: directly produce compact meshes

- The network learns how to best reconstruct the set of training 3D shapes using N (e.g., 4,096) planes
- At inference, obtain planes and convexes based on input
  - Output mesh directly, no Marching Cubes
  - Compact: small # planes
  - Sharp features



(a) BSP-Net output
(1,038 vertices, 219 polygons or 600 triangles)

(b) IM-NET output
(sampled at 256^3, 91,542 vertices, 183,096 triangles)

An evolution of neural implicit (2019 - now)

IM-Net, OCC-Net, DeepSDF, CVPR 2019

Encoder design

- Global + local feature encoding
- Query-specific feature encoding

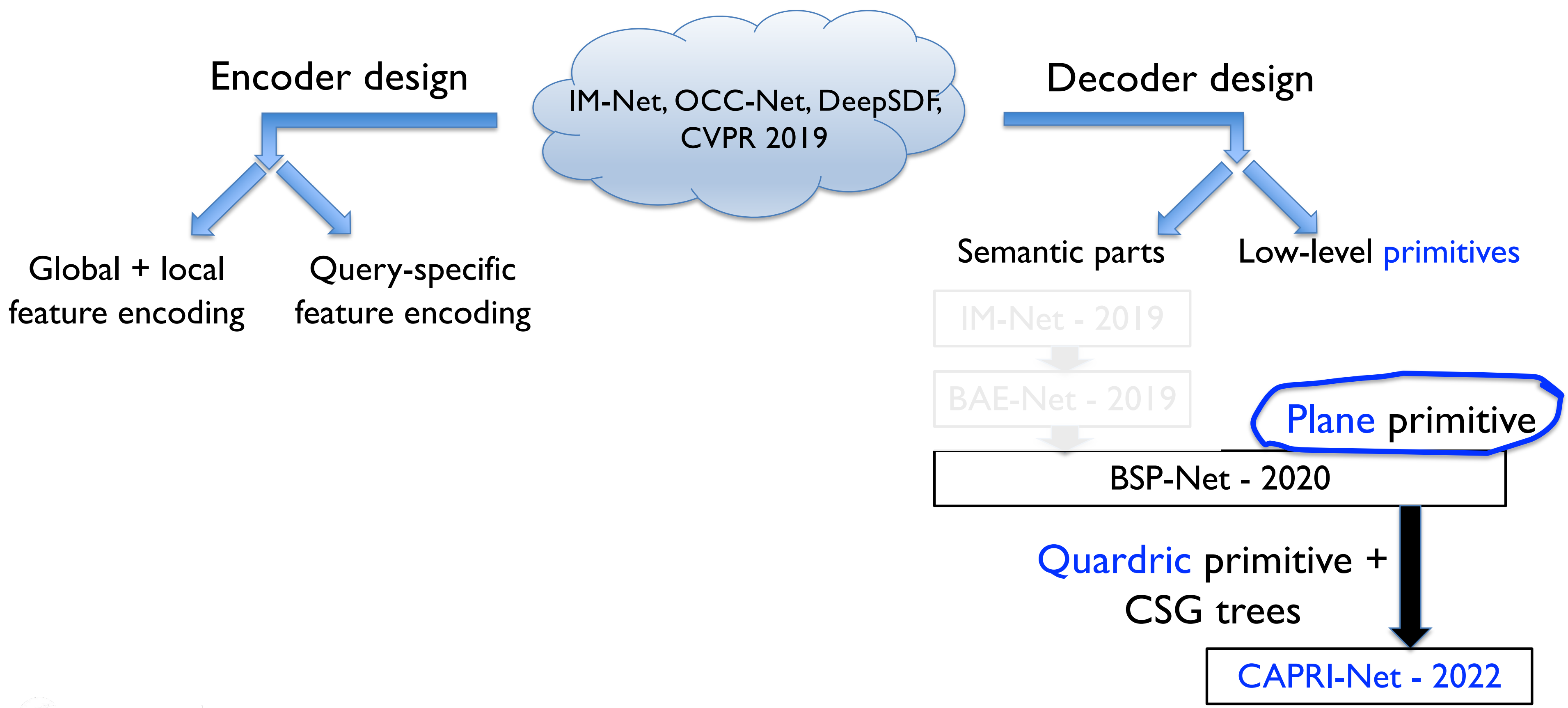
Decoder design

- Semantic parts
  - IM-Net - 2019
  - BAE-Net - 2019
- Low-level primitives
  - BSP-Net - 2020

Direct mesh outputs

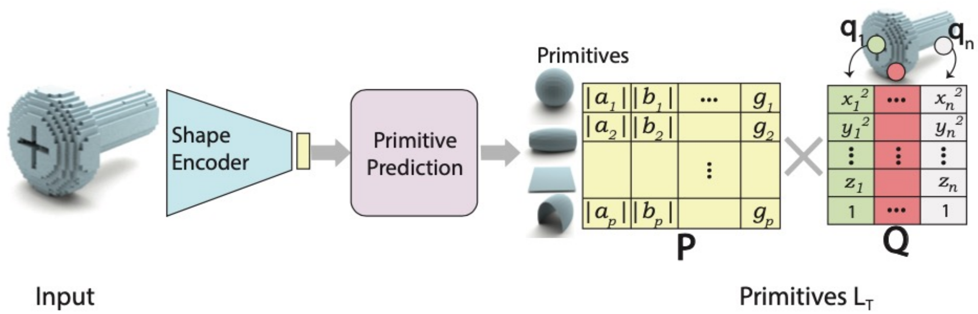
- Neural Marching Cubes (NMC) - 2020
- Neural Dual Contouring (NDC) - 2022
- MobileNeRF - 2023

An evolution of neural implicit (2019 - now)



Learning primitive assemblies (aside)

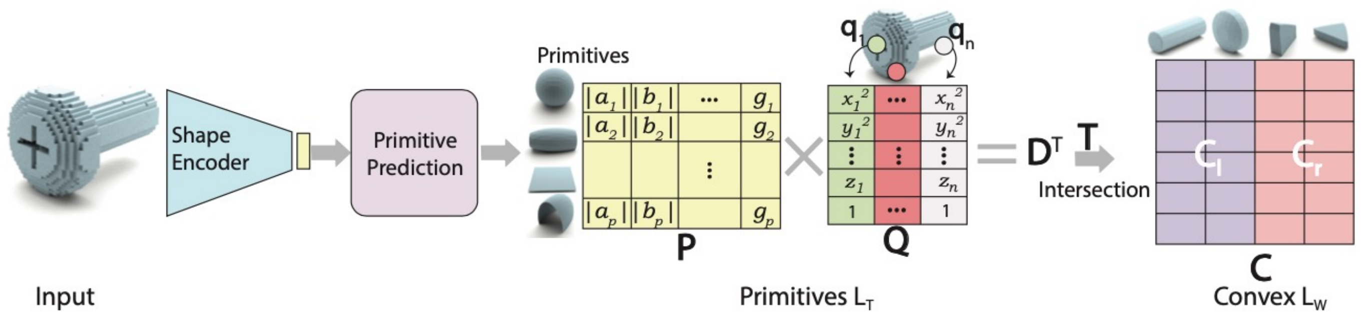
- BSP-Net extension with quadric primitives and difference operation
- Goal: to produce compact CSG trees, without GT supervision



CAPRI-Net: [Yu et al., CVPR 2022]

Learning primitive assemblies (aside)

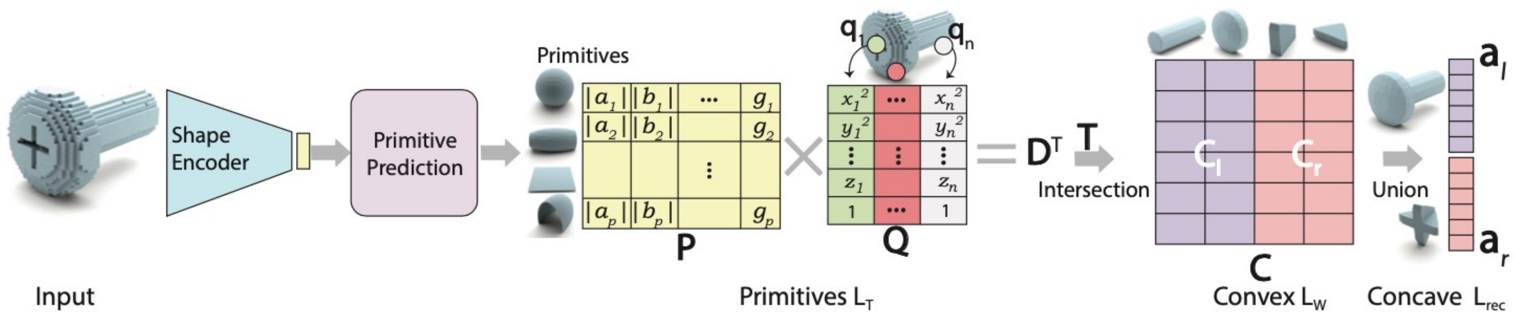
- BSP-Net extension with quadric primitives and difference operation
- Goal: to produce compact CSG trees, without GT supervision



CAPRI-Net: [Yu et al., CVPR 2022]

Learning primitive assemblies (aside)

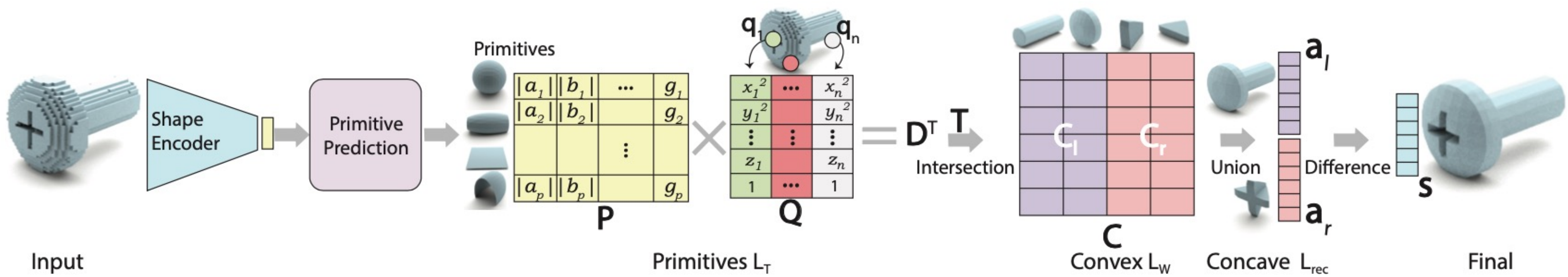
- BSP-Net extension with quadric primitives and difference operation
- Goal: to produce compact CSG trees, without GT supervision



CAPRI-Net: [Yu et al., CVPR 2022]

Learning primitive assemblies (aside)

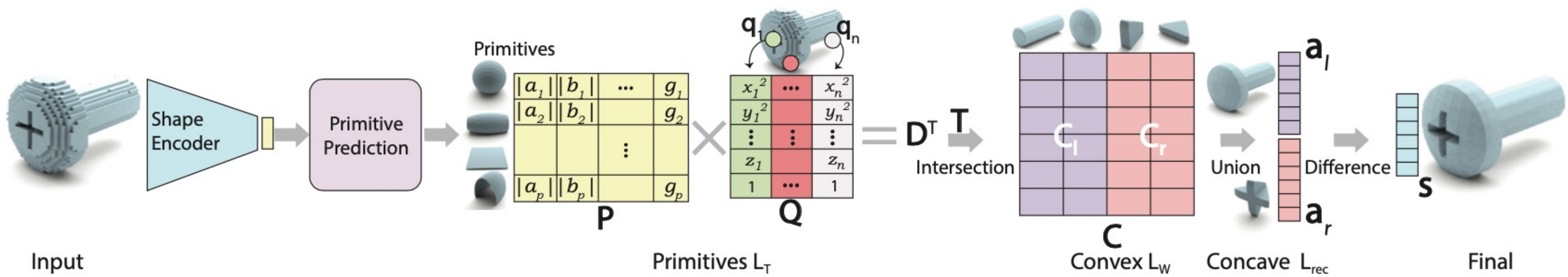
- BSP-Net extension with quadric primitives and difference operation
- Goal: to produce compact CSG trees, without GT supervision



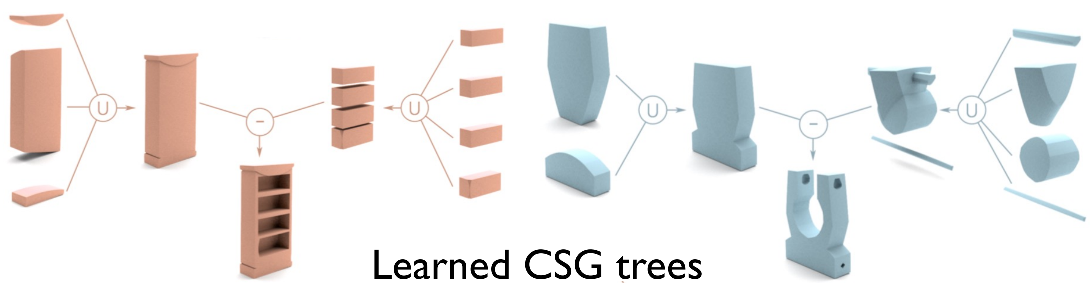
CAPRI-Net: [Yu et al., CVPR 2022]

Learning primitive assemblies (aside)

- BSP-Net extension with quadric primitives and difference operation
- Goal: to produce compact CSG trees, without GT supervision



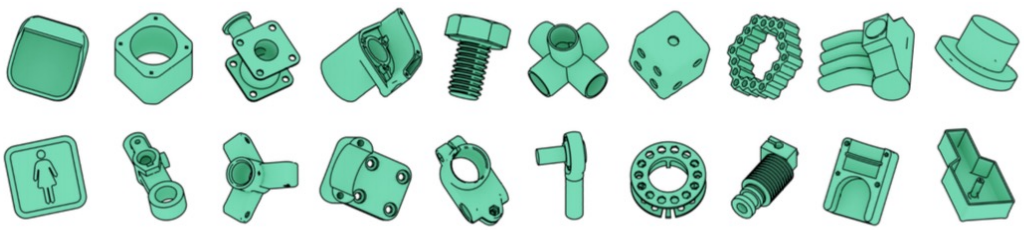
CAPRI-Net: [Yu et al., CVPR 2022]

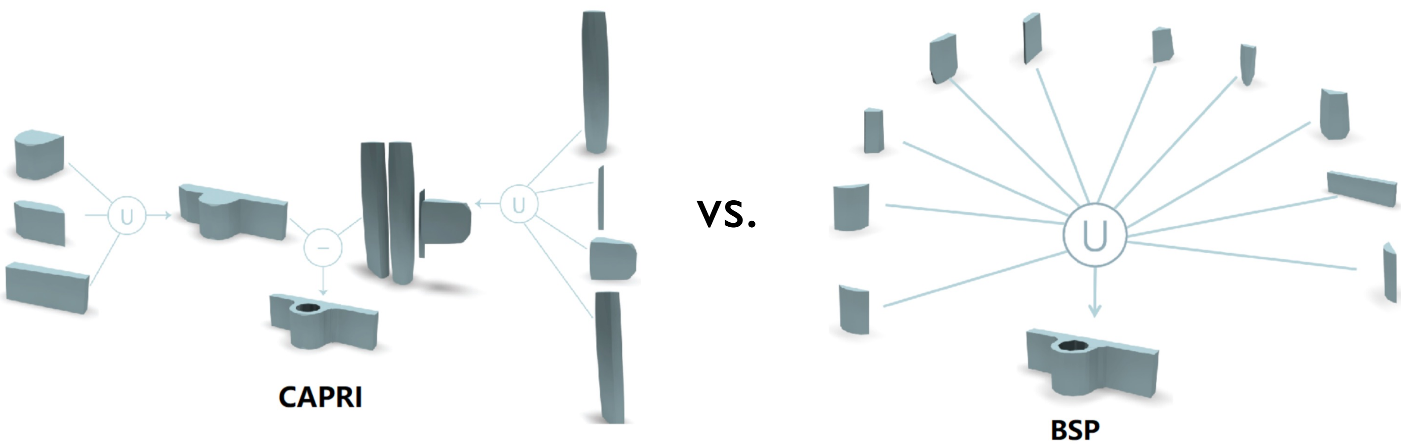


Learned CSG trees

CAPRI-Net: compact primitive assembly (aside)

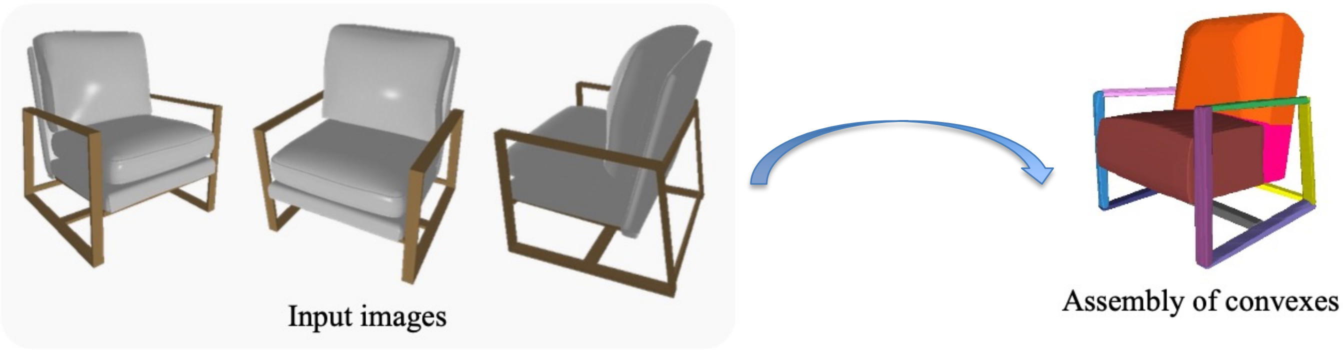
❖ Trainable over ABC dataset (without class labels)





CAPRI-Net from multi-view images (aside)

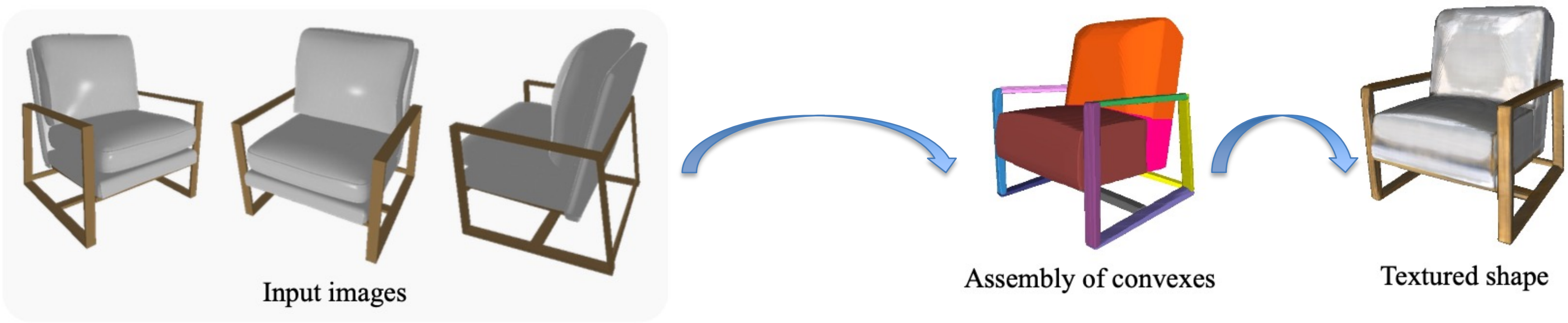
❖ 3D primitive assemblies from sparse and wide-baseline views



[Yu et al. ECCV 2024]

DPA-Net: differentiable primitive assembly (aside)

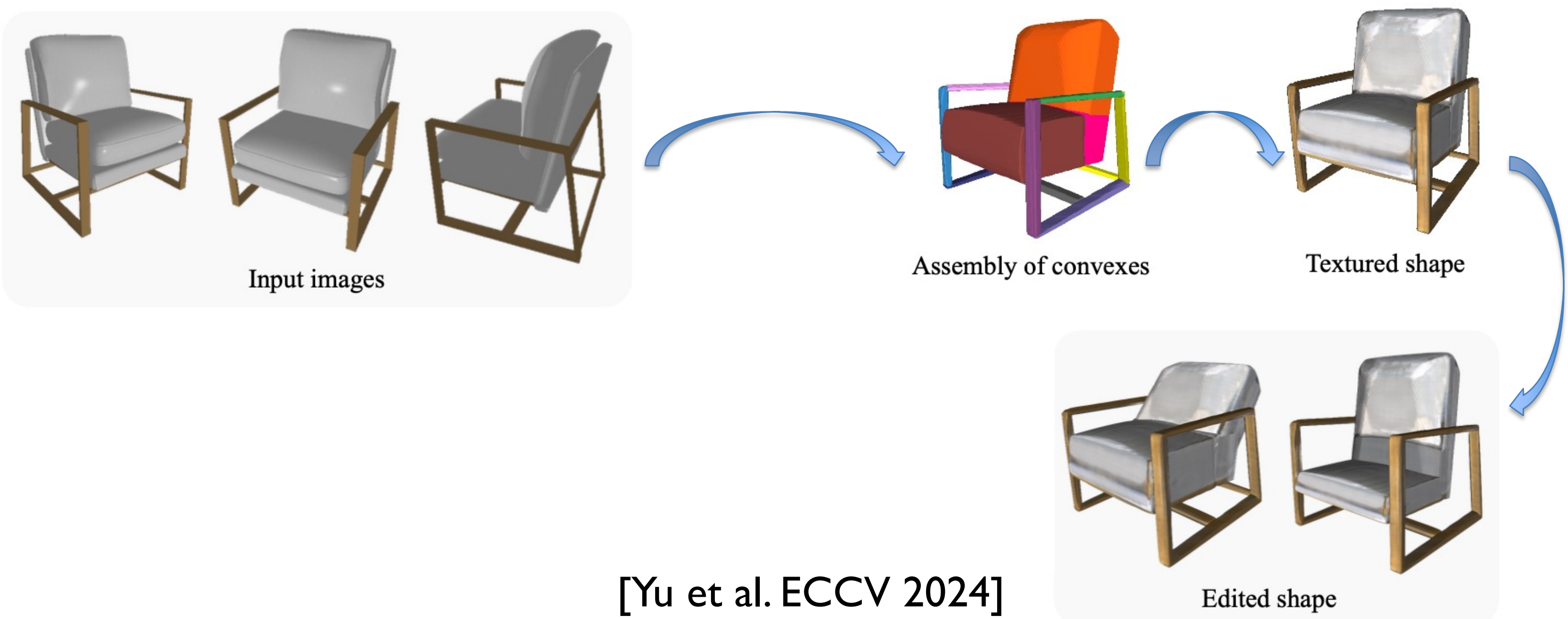
❖ Primitive assembly is differentiable, without 3D supervision



[Yu et al. ECCV 2024]

DPA-Net: differentiable primitive assembly (aside)

❖ Generated primitive assembly directly supports editing



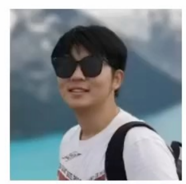
[Yu et al. ECCV 2024]

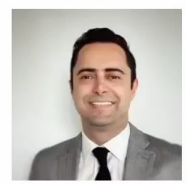
Final example: a “smart” representation (aside)

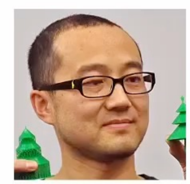
Slice3D:
Multi-Slice, Occlusion-Revealing, Single View 3D Reconstruction


Yizhi Wang


Wallace Lira


Wenqi Wang


Ali Mahdavi-Amiri


Hao (Richard) Zhang

GrUVi Lab, Simon Fraser University

Project website: https://yizhiwang96.github.io/Slice3D/