Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis

Tianchang Shen[1,2,3] Jun Gao[1,2,3] Kangxue Yin[1] Ming-Yu Liu[1] Sanja Fidler[1,2,3]

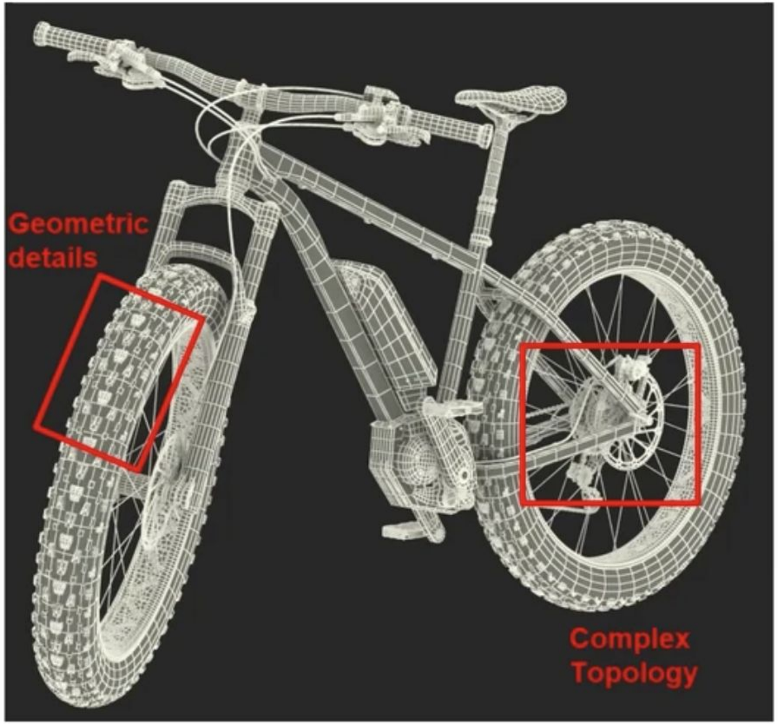
[1]NVIDIA [2]University of Toronto [3]Vector Institute

NeurIPS 2021

Many Field Requires High-Quality 3D Content

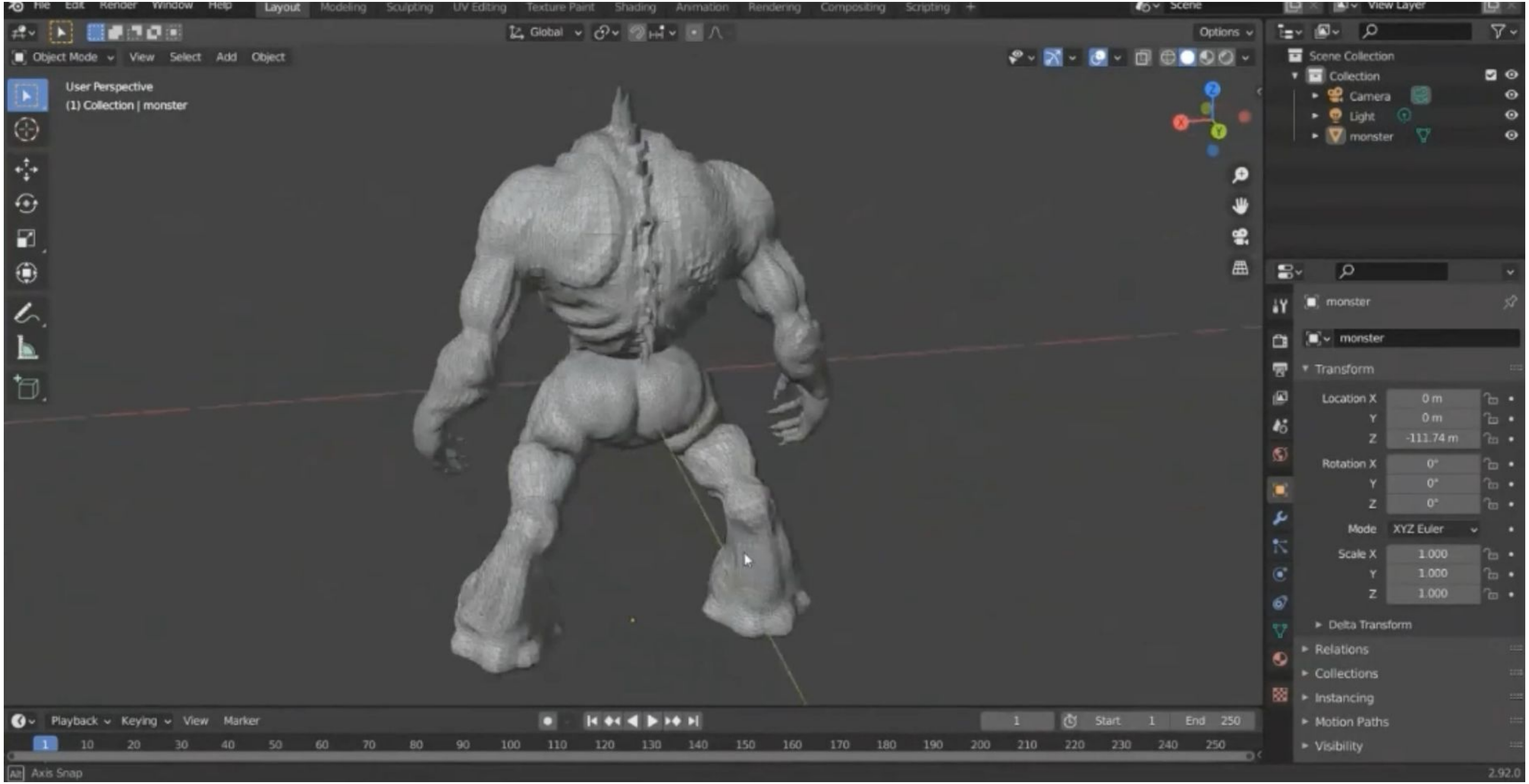
AV Simulation

VR/AR



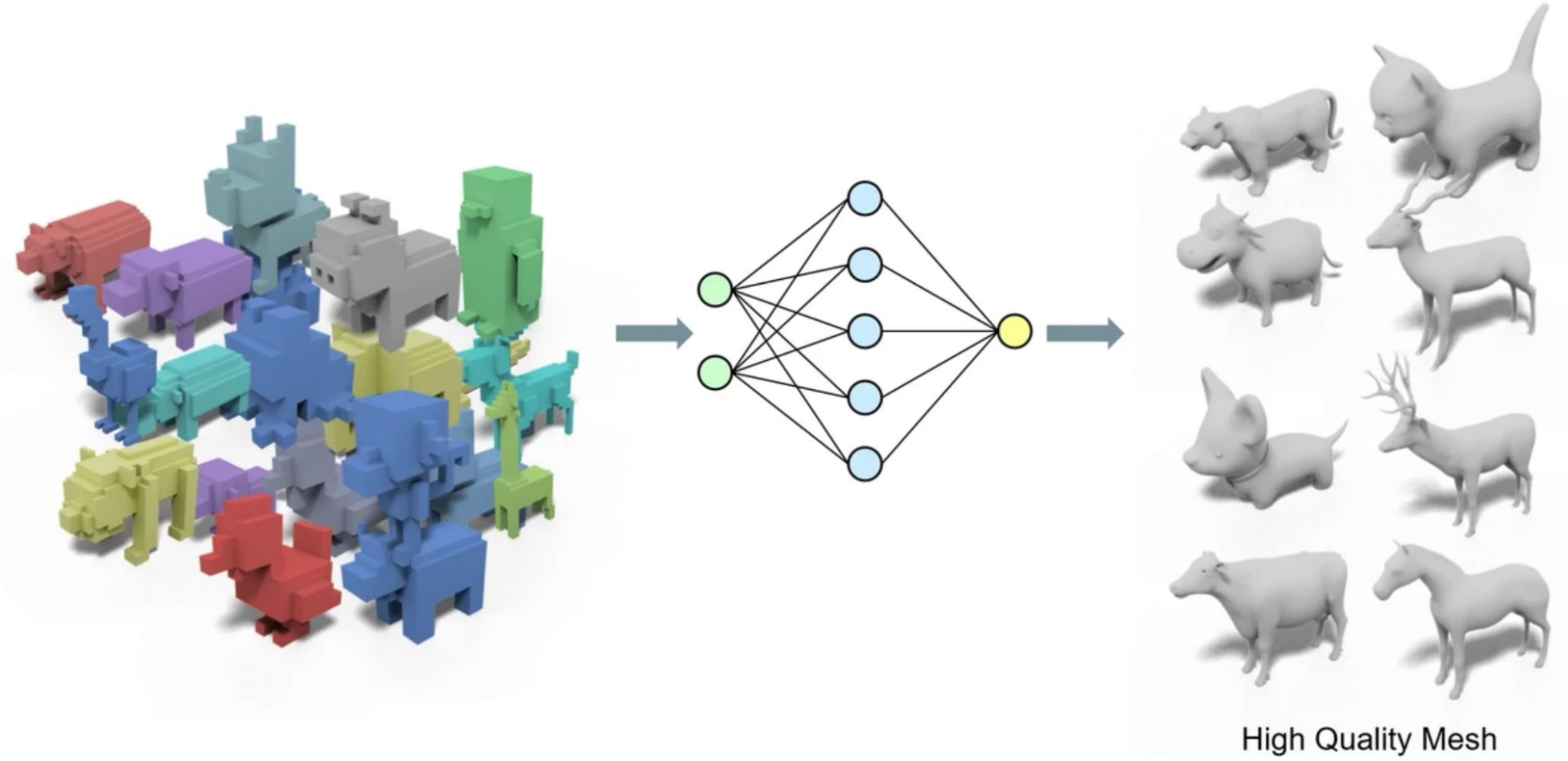
Source: TurboSquid

Creating High-Quality 3D Content Requires Expertise



Rough 3D Shapes

Rough 3D Shapes → Detailed 3D Shapes



Motivation

What kind of 3D representation should we use to represent high-quality 3D contents?

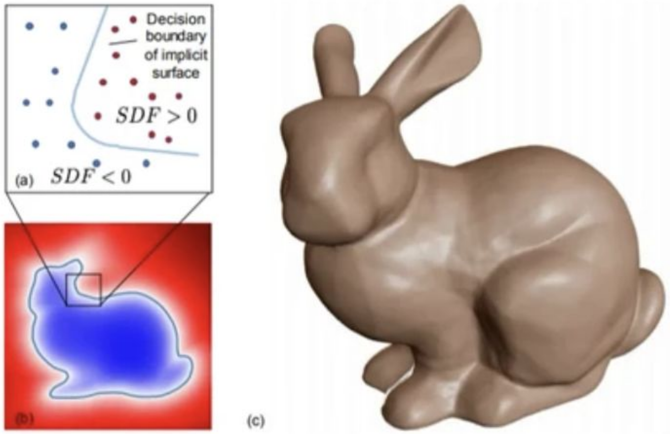
Discrete Representations

Limited to pre-defined resolution or topology.

Deep Implicit Fields (DIFs)

$$f_\theta(x, y, z) \approx s(x, y, z)$$

Signed distance from (x, y, z) to closest surface



DeepSDF [Park et al. 2019]

Deep Implicit Fields (DIFs)

$$f_\theta(x, y, z) \approx s(x, y, z)$$

Pros:

- Represent arbitrary topology
- Continuous

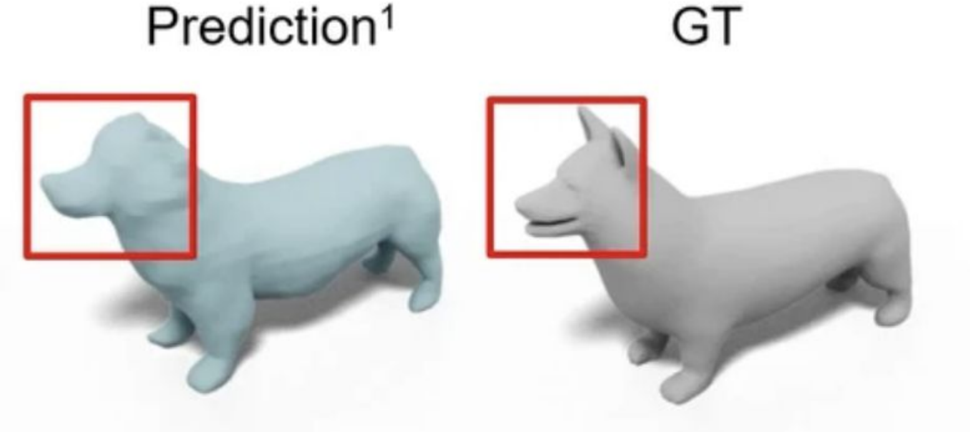
[1]Result of [Peng et al. 2020]

Deep Implicit Fields (DIFs)

$$f_\theta(x, y, z) \approx s(x, y, z)$$

Cons:

- Regressing SDF/OF in generative tasks do not capture geometric details



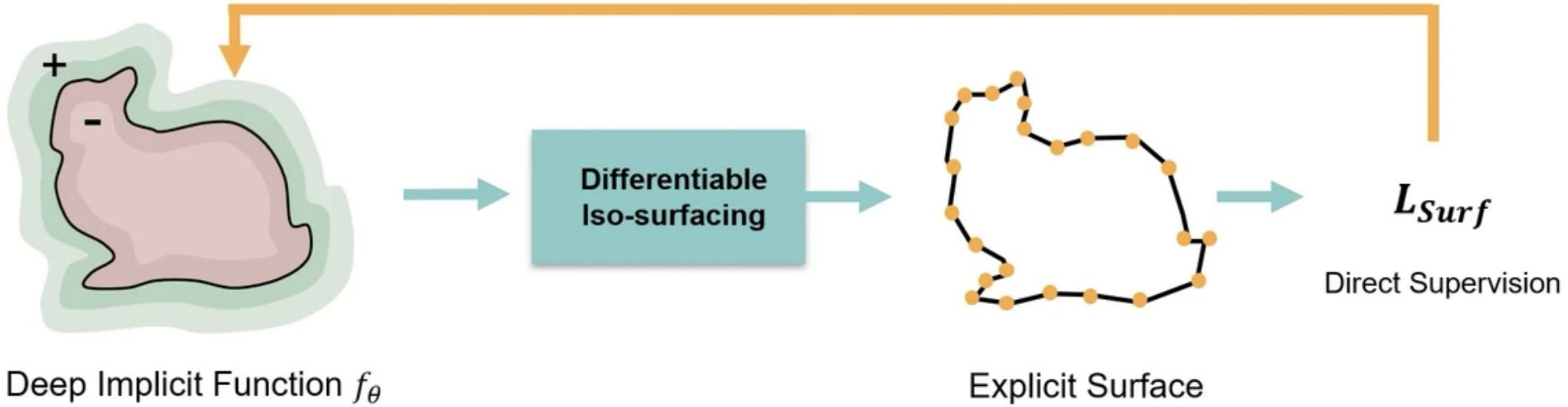
Deep Implicit Fields (DIFs)

$$f_\theta(x, y, z) \approx s(x, y, z)$$

Cons:

- Requires costly, lossy and non-differentiable meshing step (such as marching cube)

Key Idea: Differentiable Iso-surfacing



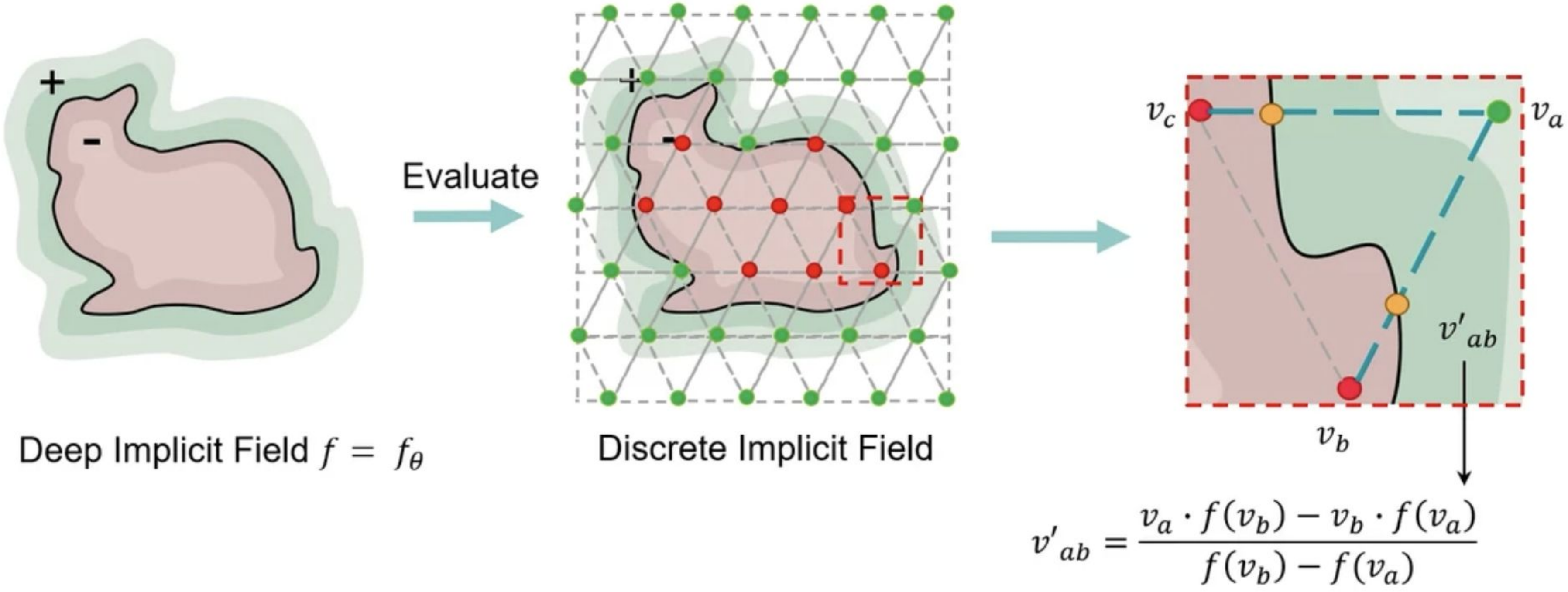
Optimizing f_θ for L_{Surf}

- Aware of **quantization error** from meshing
- Higher quality shapes with **finer geometric details**

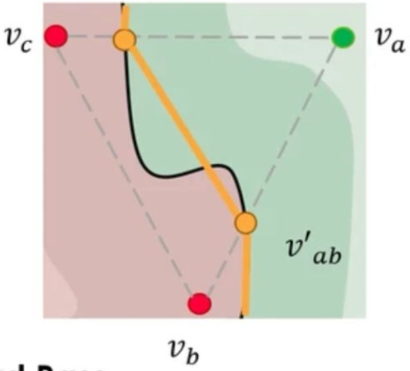
Key Idea: Differentiable Iso-surfacing

Input | Implicit Approach [Peng et al. 2020] | Our result | GT

Marching Tetrahedra



Differentiability of MT

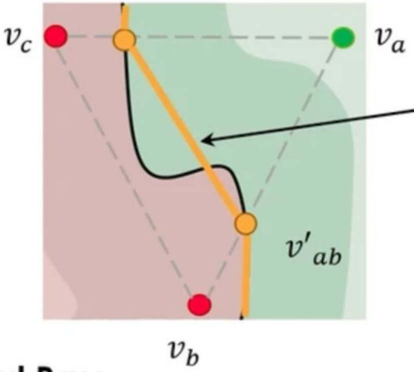


Forward Pass

$$v'_{ab} = \frac{v_a \cdot f(v_b) - v_b \cdot f(v_a)}{\boxed{f(v_b) - f(v_a)}}$$

Only evaluated at $\mathrm{sign}\big(f(v_b)\big) \neq sign(f(v_a))$

Differentiability of MT



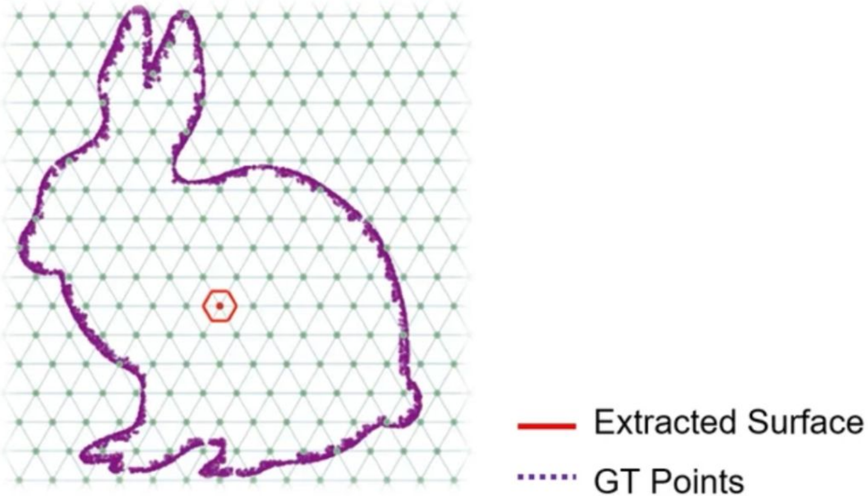
Forward Pass

$$v'_{ab} = \frac{v_a \cdot f(v_b) - v_b \cdot f(v_a)}{f(v_b) - f(v_a)}$$

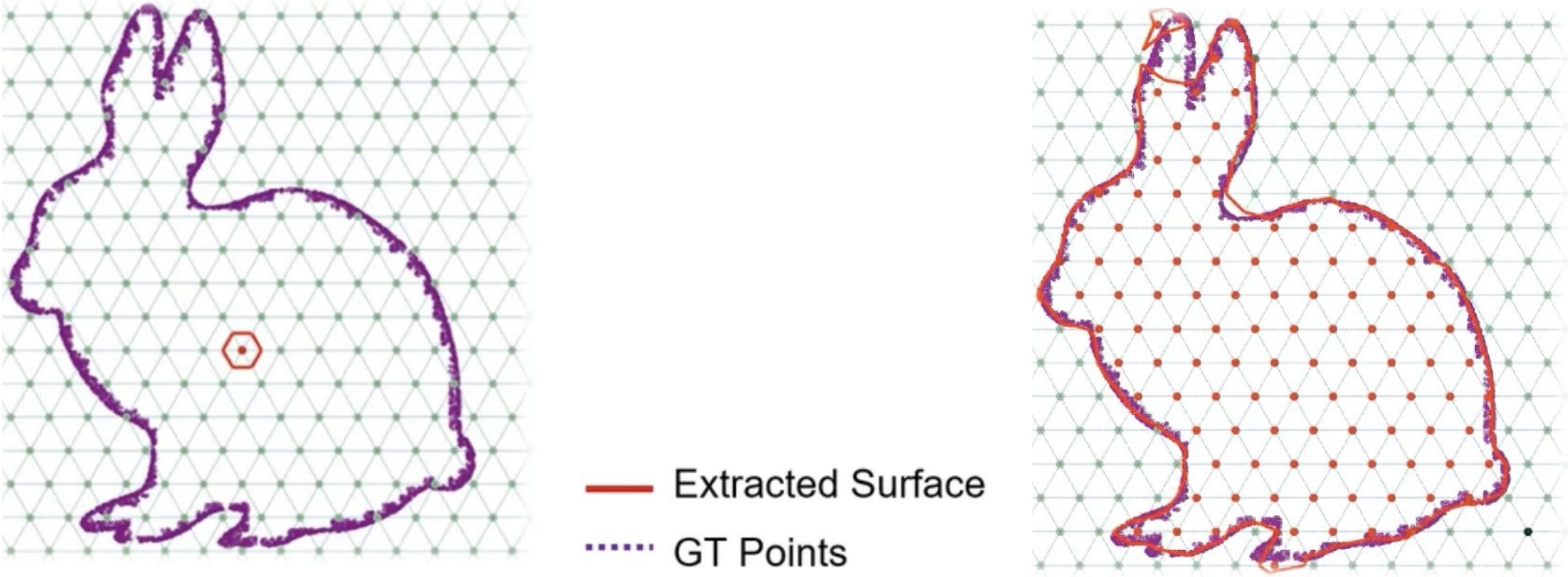
Backward Pass

$$L_{surf} = \textit{Loss defined on extracted surface } \{\mathrm{v_{ij}}\}$$

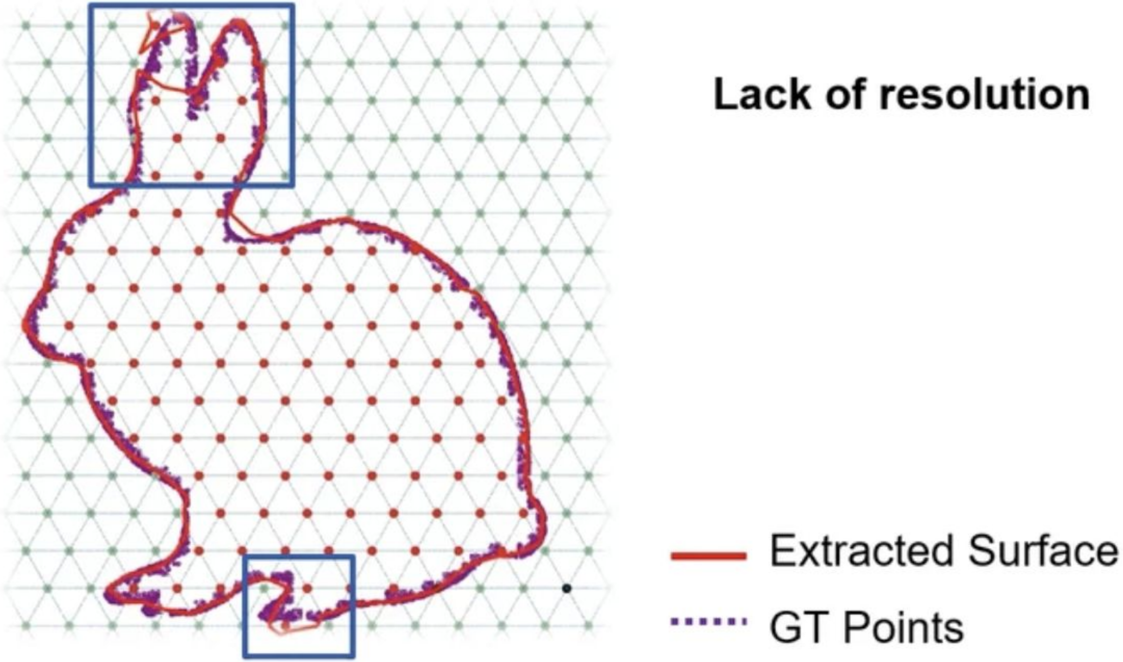
$$\frac{\partial L_{surf}}{\partial f(v_a)} = \frac{\partial L_{surf}}{\partial v'_{ab}} \frac{f(v_b)(v_a - v_b)}{\left(f(v_b) - f(v_a)\right)^2}$$



Differentiability of MT

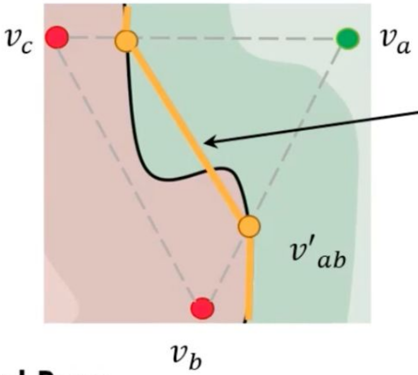


Differentiability of MT



Lack of resolution

Differentiability of MT



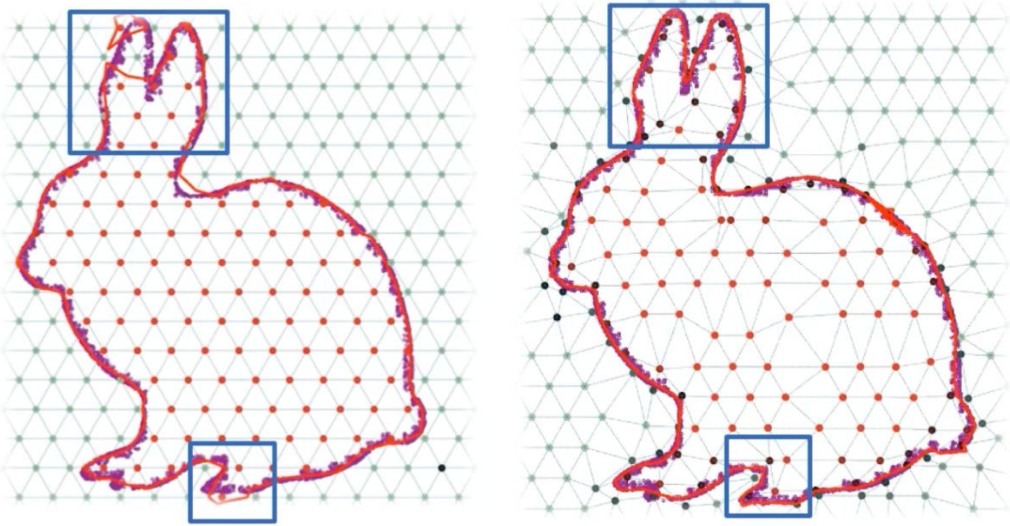
Forward Pass

$$v'_{ab} = \frac{v_a \cdot f(v_b) - v_b \cdot f(v_a)}{f(v_b) - f(v_a)}$$

Backward Pass

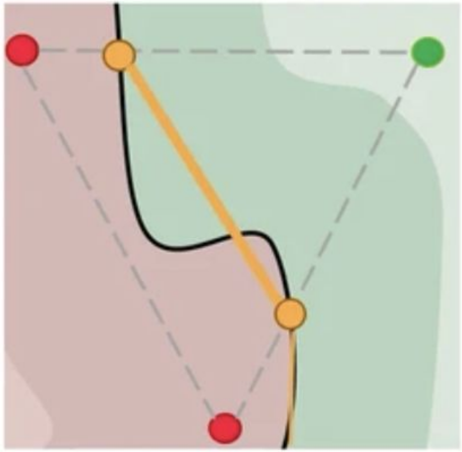
$L_{surf} = $ *Loss defined on extracted surface* $\{\mathrm{v_{ij}}\}$

$$\frac{\partial L_{surf}}{\partial f(v_a)} = \frac{\partial L_{surf}}{\partial v'_{ab}} \frac{f(v_b)(v_a - v_b)}{\left(f(v_b) - f(v_a)\right)^2} \qquad \frac{\partial L_{surf}}{\partial v_a} = \frac{\partial L_{surf}}{\partial v'_{ab}} \frac{f(v_b)}{f(v_b) - f(v_a)}$$

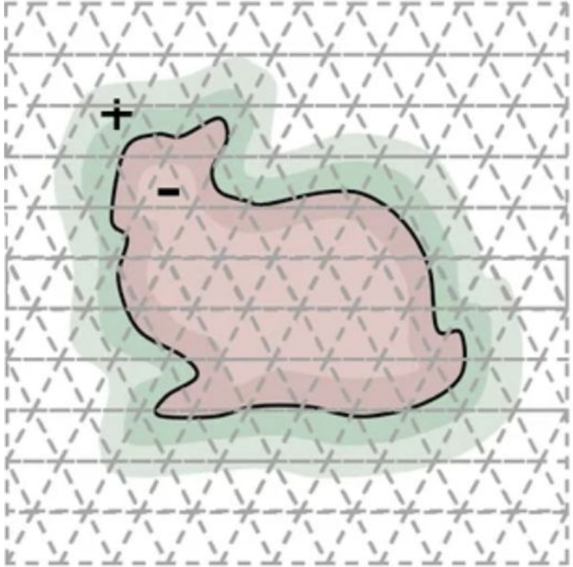


Optimizing grid deformation as in DefTet [Gao et al. 2020] learns better alignment with surface

Volume Subdivision

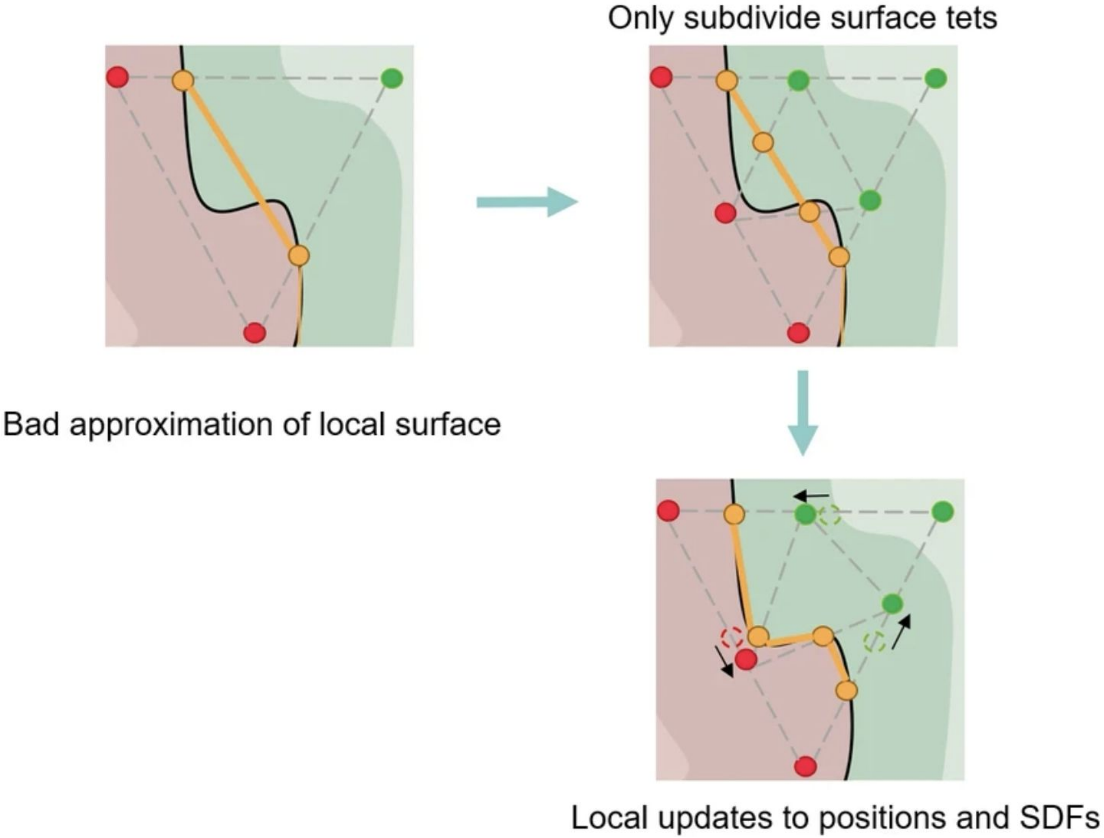


Bad approximation of local surface

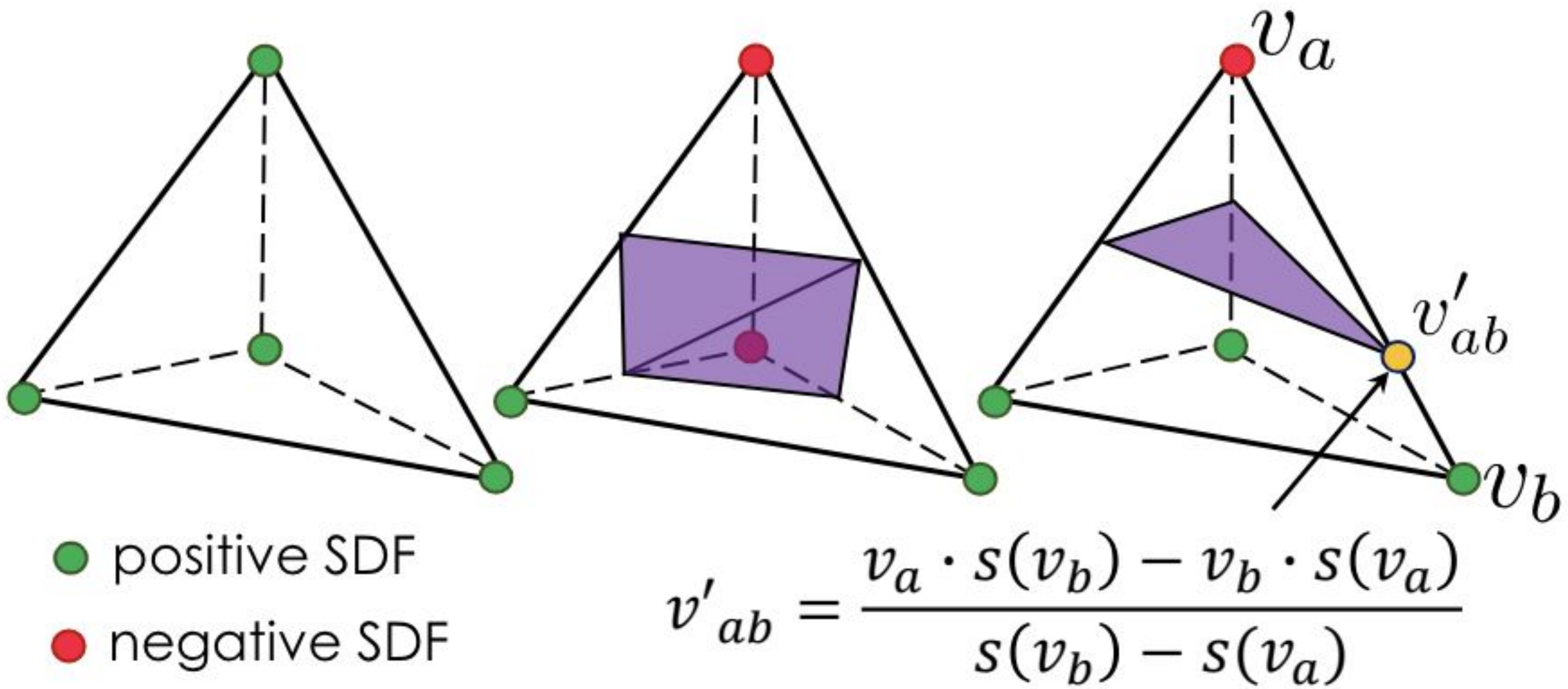


Higher Resolution Grid?
Too costly!

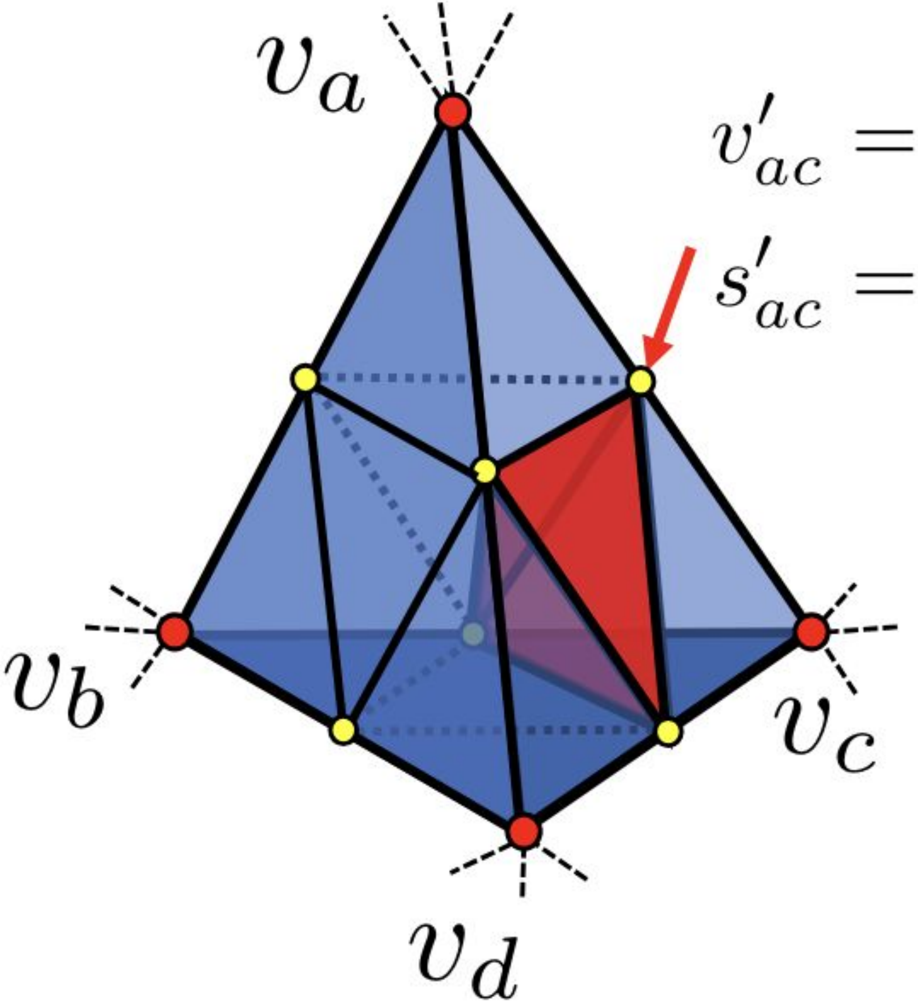
Volume Subdivision



Selective Volume Subdivision to replace the global high resolution grid, which is computationally inefficient



v_a
v'_{ab}
v_b
positive SDF
negative SDF
$v'_{ab} = \frac{v_a \cdot s(v_b) - v_b \cdot s(v_a)}{s(v_b) - s(v_a)}$

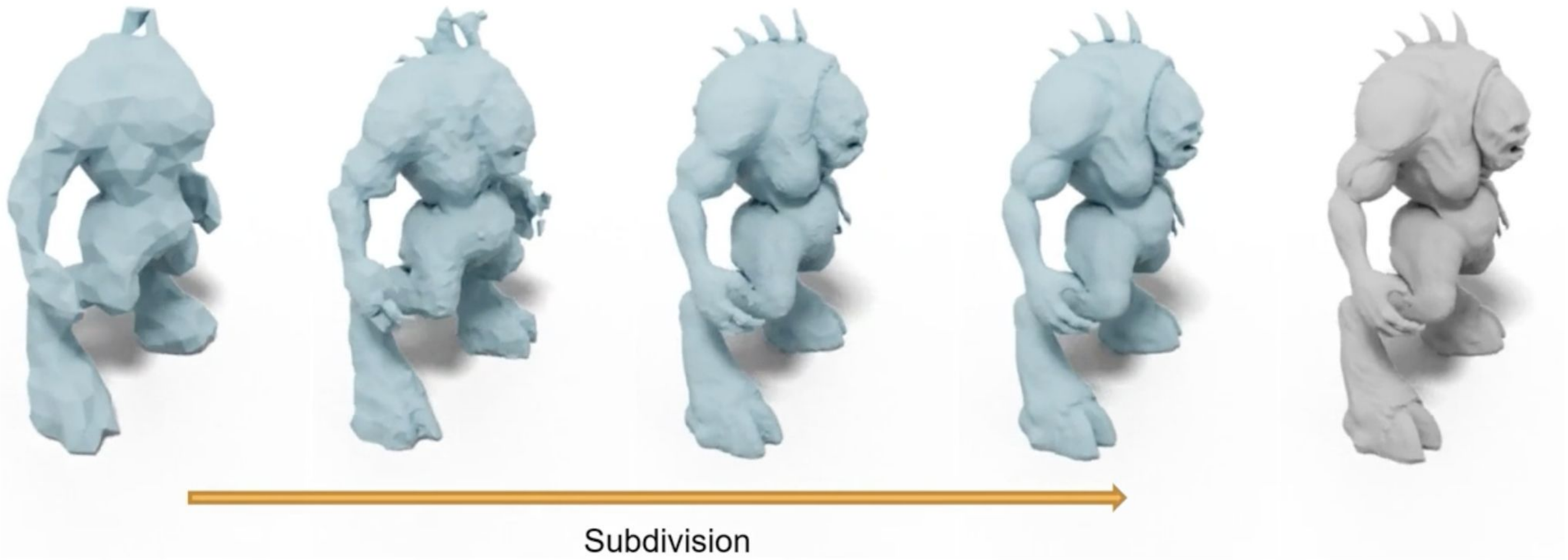


$$v'_{ac} = \frac{1}{2}(v_a + v_c)$$

$$s'_{ac} = \frac{1}{2}\big(s(v_a) + s(v_c)\big)$$

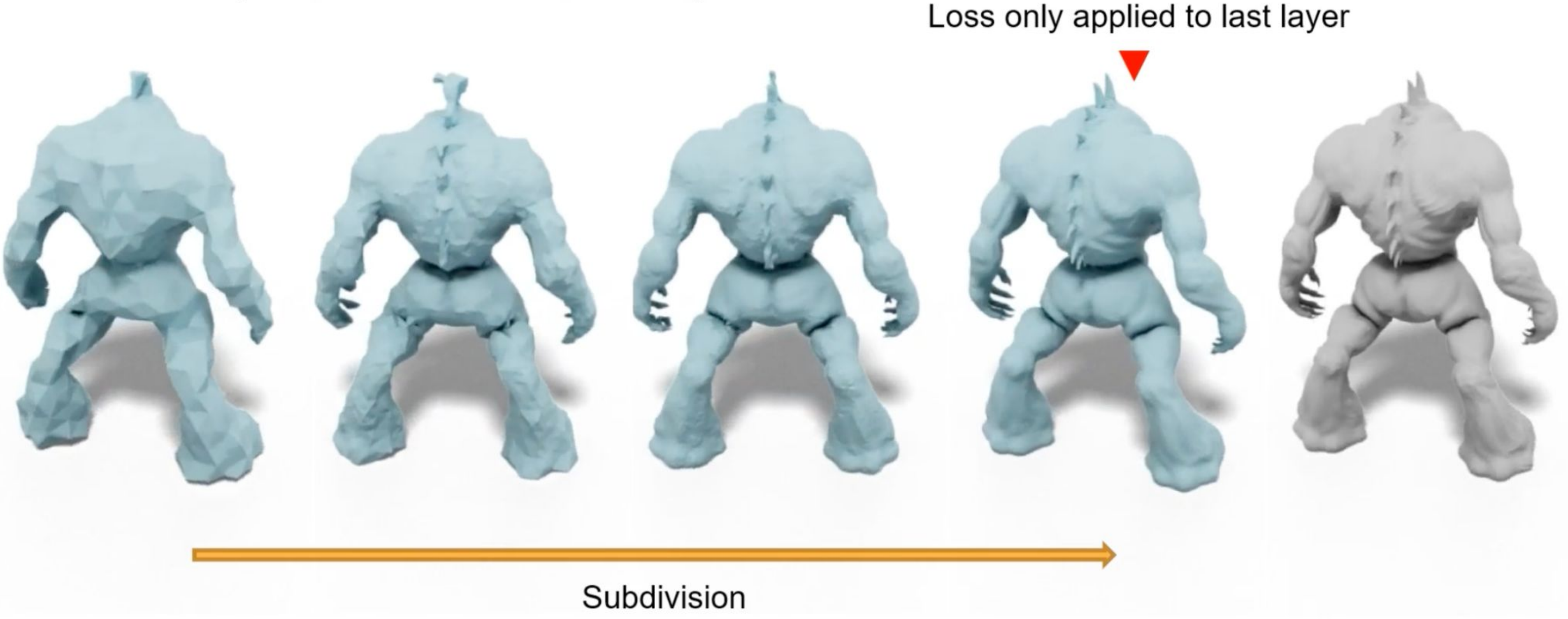
Volume Subdivision

Automatically learns the subdivision hierarchy

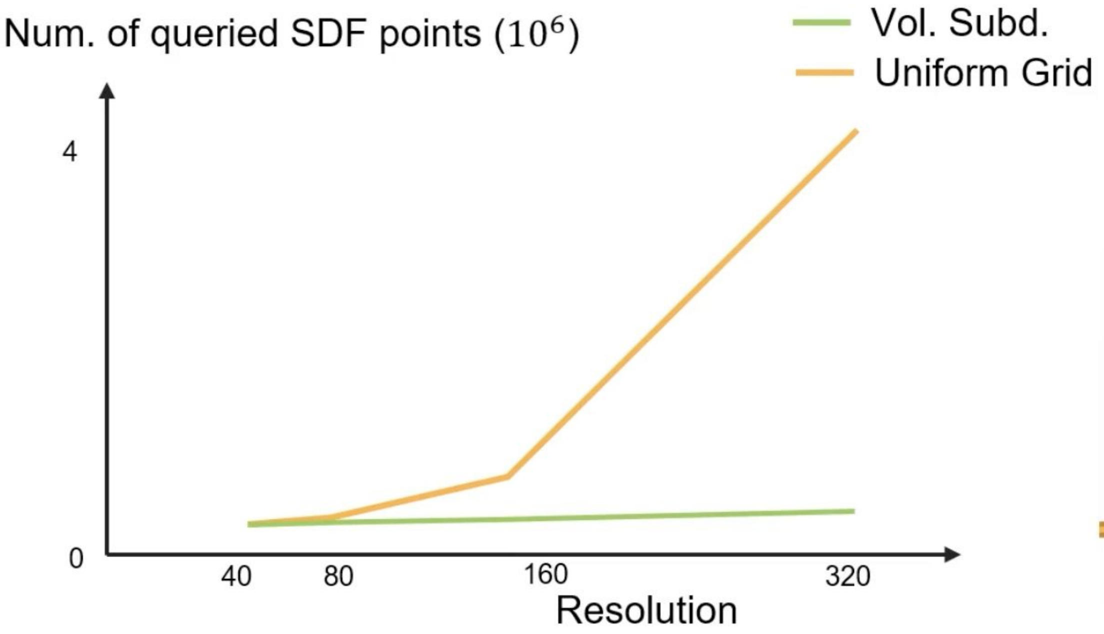


Volume Subdivision

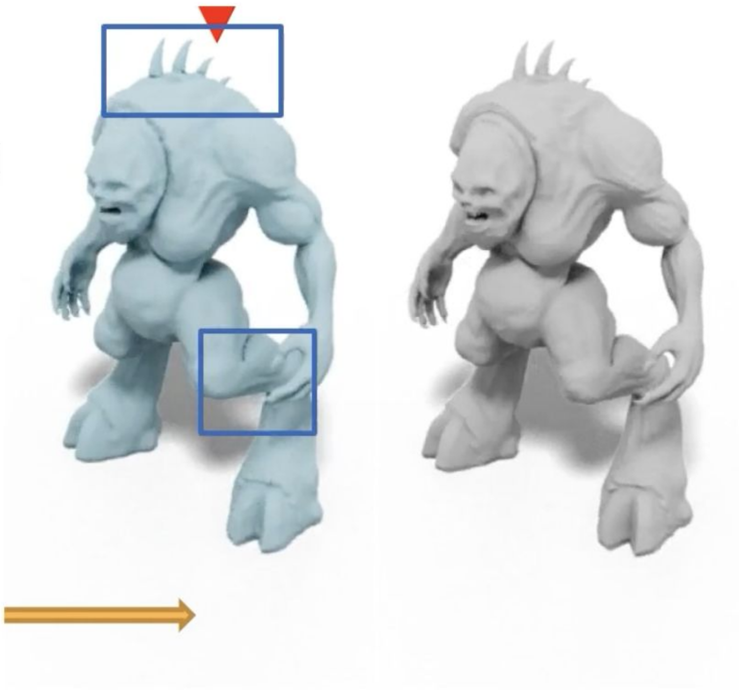
Automatically learns the subdivision hierarchy



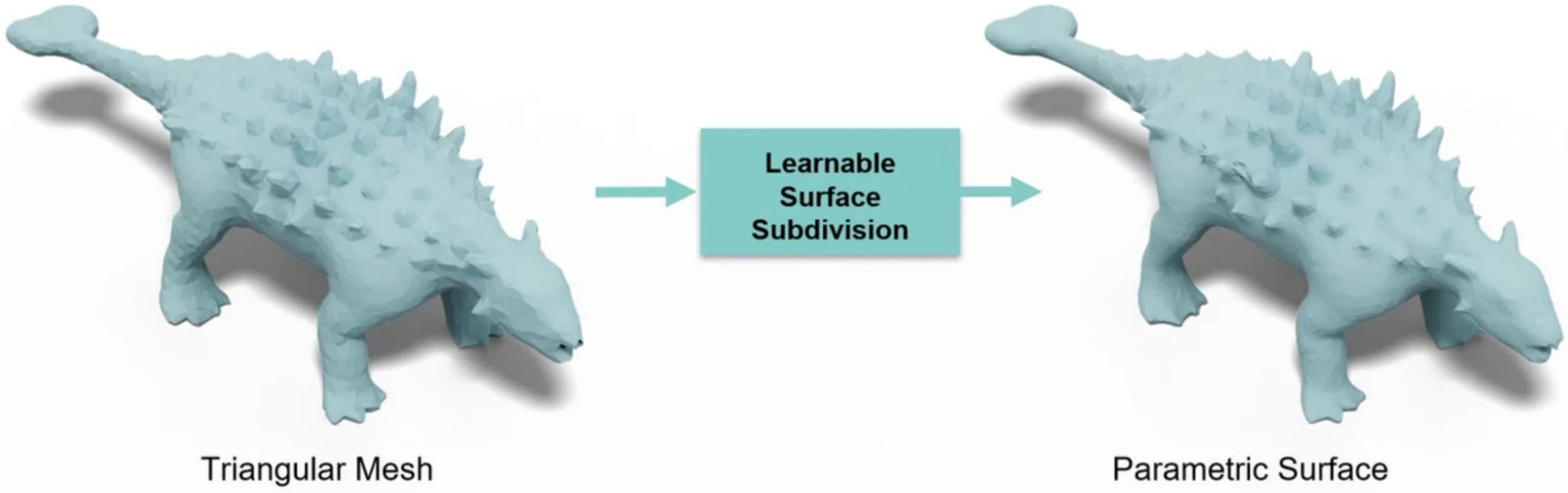
Volume Subdivision



Loss only applied to last layer

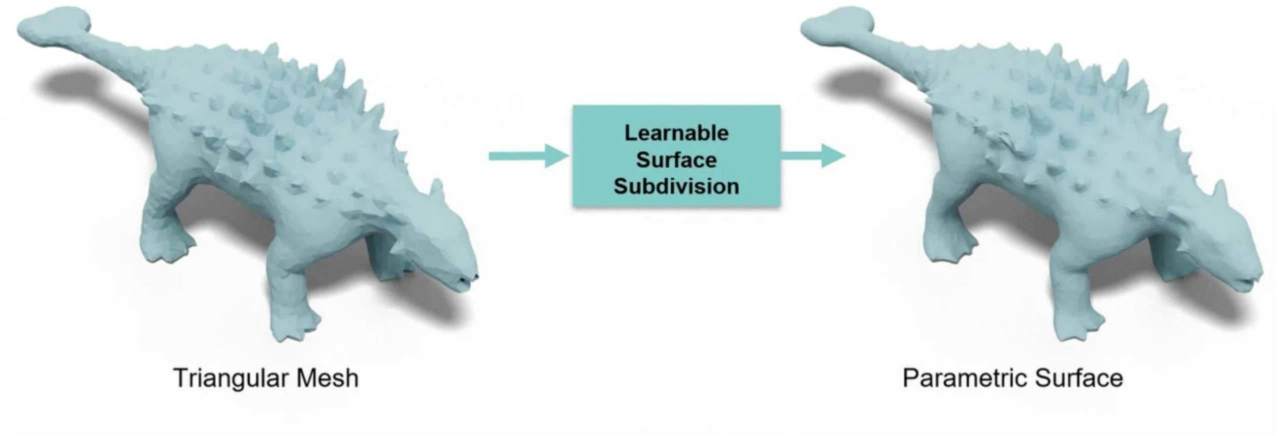


Surface Subdivision

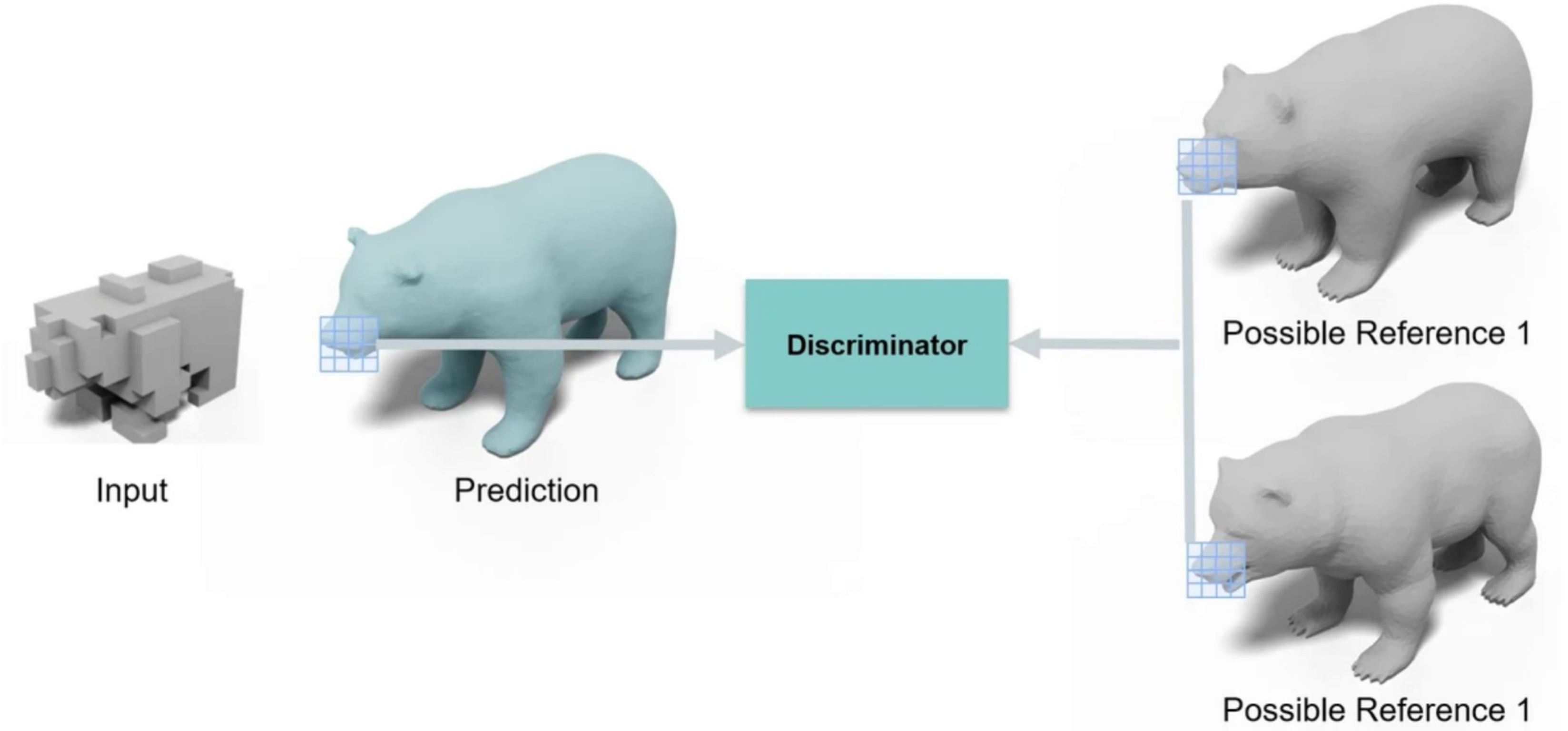


Surface Subdivision

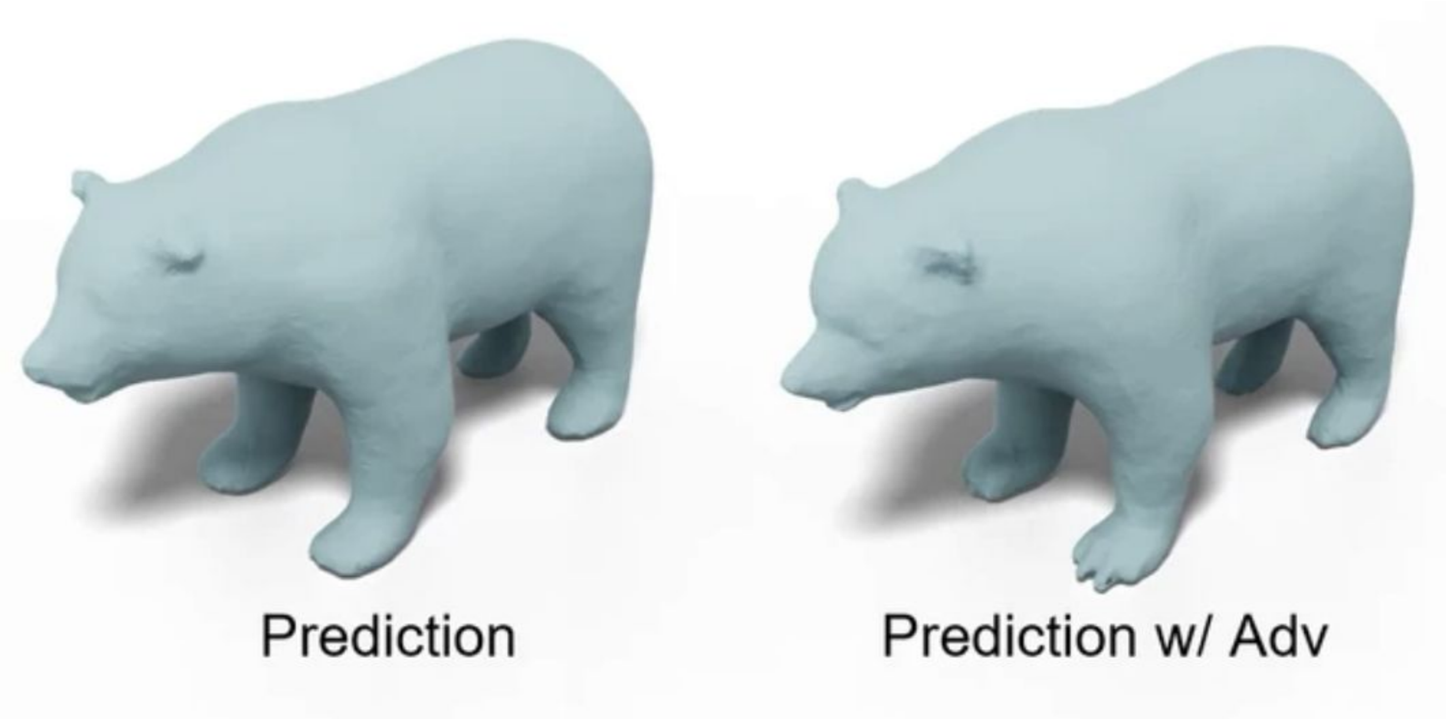
A learnable, fully differentiable surface subdivision algorithm.



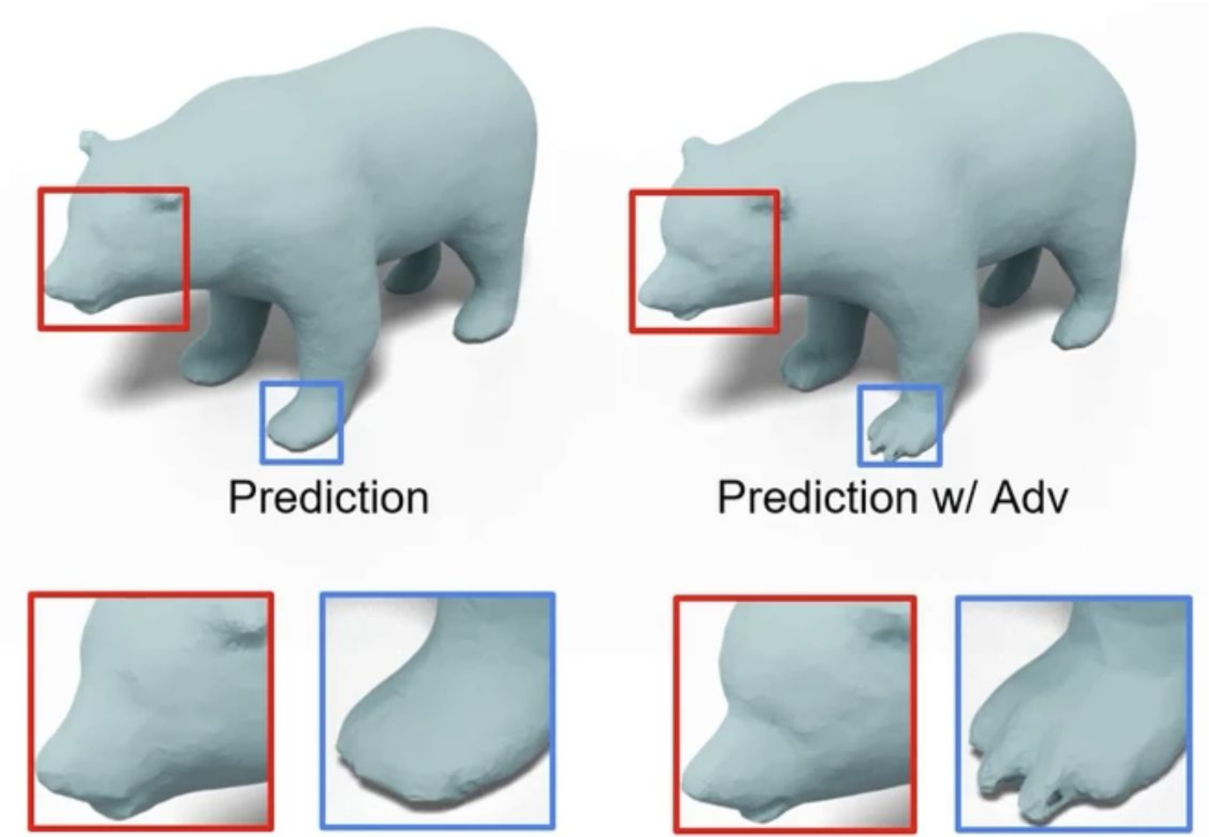
Reconstruction Loss Produces Mean Shape



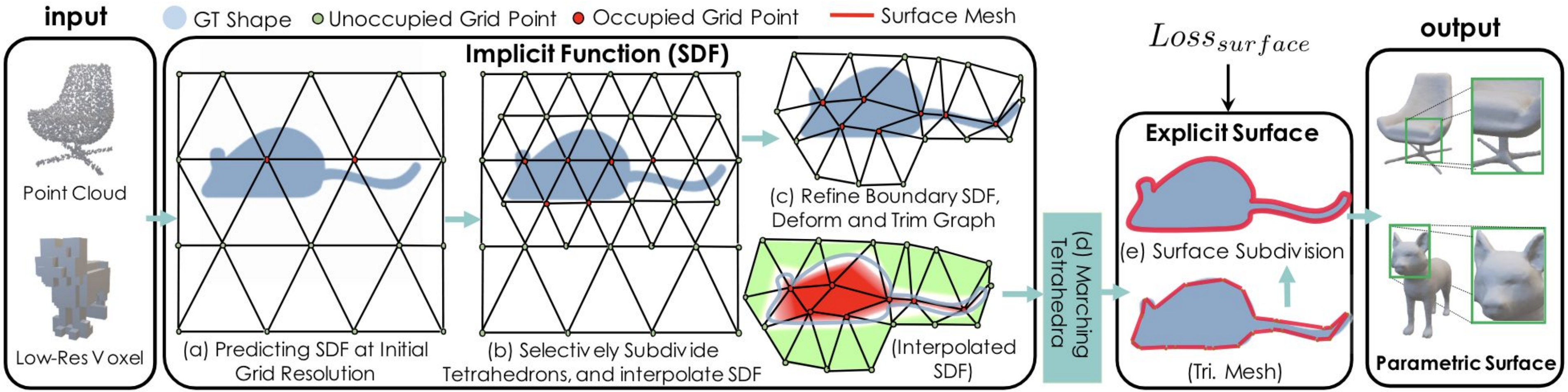
Reconstruction Loss Produces Mean Shape



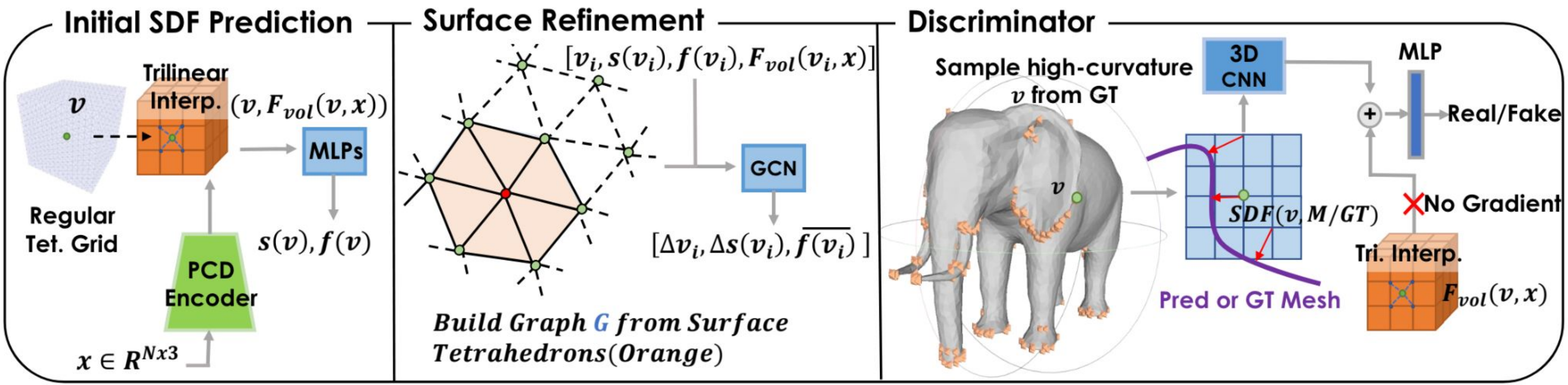
Reconstruction Loss Produces Mean Shape



Deep Marching Tetrahedra



Deep Marching Tetrahedra



Loss Function

The loss function contains three different terms:

- a surface alignment loss to encourage the alignment with ground truth surface,
- an adversarial loss to improve realism of the generated shape
- regularizations to regularize the behavior of SDF and vertex deformations

Surface Alignment Loss

$$L_{\text{cd}} = \sum_{p \in P_{pred}} \min_{q \in P_{gt}} ||p - q||_2 + \sum_{q \in P_{gt}} \min_{p \in P_{pred}} ||q - p||_2, L_{\text{normal}} = \sum_{p \in P_{pred}} (1 - |\vec{\mathbf{n}}_p \cdot \vec{\mathbf{n}}_{\hat{q}}|)$$

Adversarial Loss

$$L_{\text{D}} = \frac{1}{2}[(D(M_{gt}) - 1)^2 + D(M_{pred})^2], \; L_{\text{G}} = \frac{1}{2}[(D(M_{pred}) - 1)^2].$$

Regularization

$$L_{\text{SDF}} = \sum_{v_i \in V_T} |s(v_i) - SDF(v_i, M_{gt})|^2$$

L2 Regularization Loss

$$L_{\text{def}} = \sum_{v_i \in V_T} ||\Delta v_i||_2$$

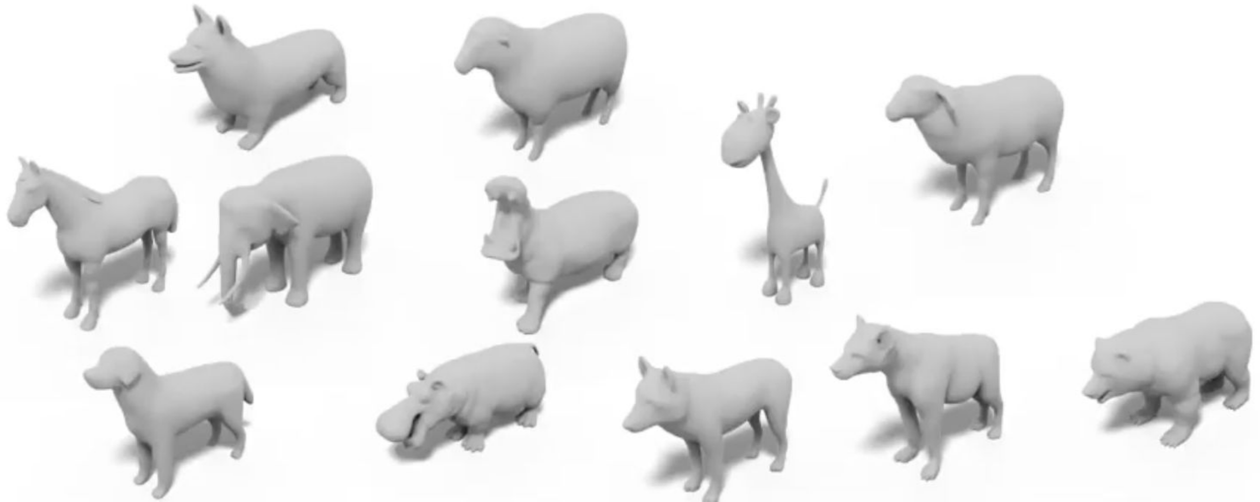
Final Loss

$$L = \lambda_{\text{cd}} L_{\text{cd}} + \lambda_{\text{normal}} L_{\text{normal}} + \lambda_{\text{G}} L_{\text{G}} + \lambda_{\text{SDF}} L_{\text{SDF}} + \lambda_{\text{def}} L_{\text{def}}$$

3D Shape Synthesis from Coarse Voxels

Animal Dataset collected from TurboSquid:

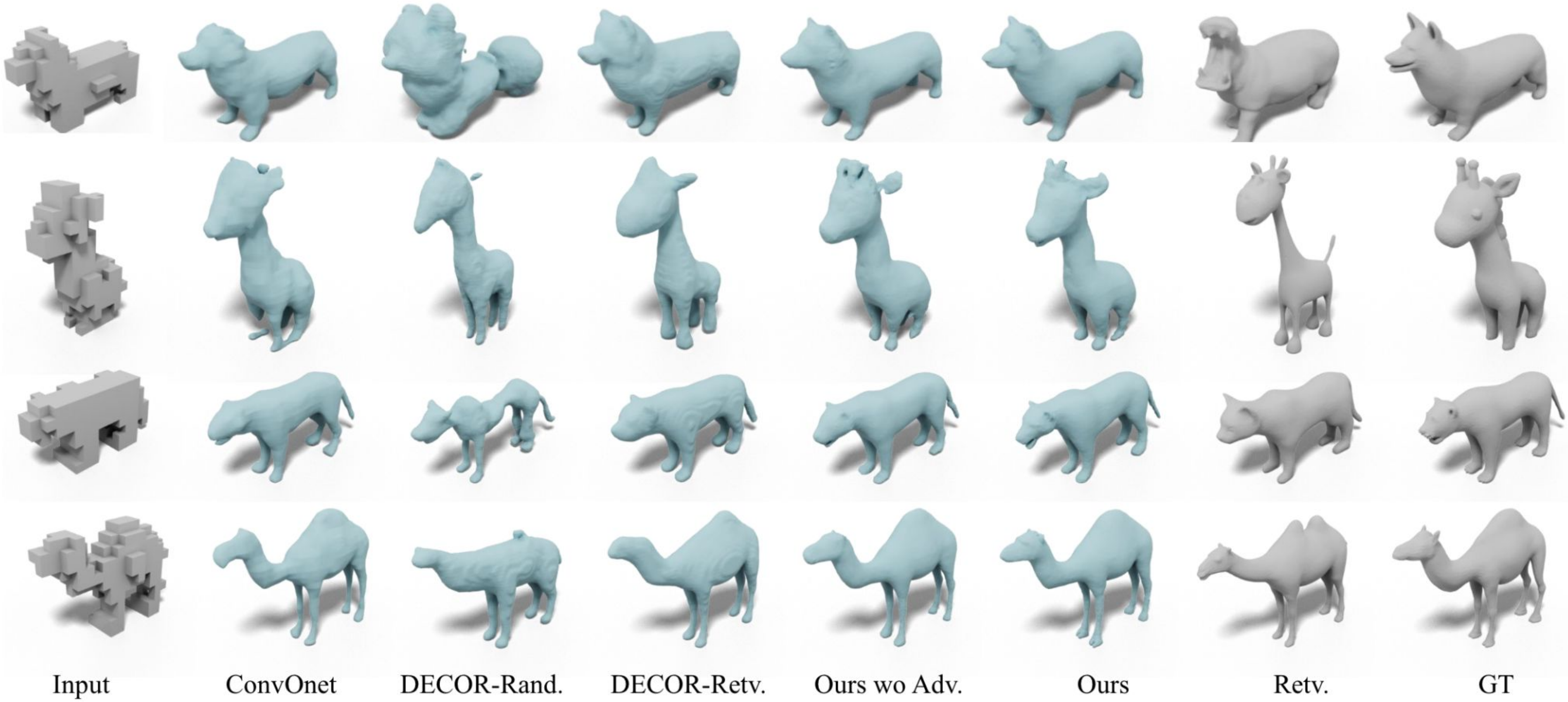
- 1562 high-quality animal models (1120 for training)
- Input: 16^3 voxel downsampled from Mesh



Generalization

Minecraft Shapes

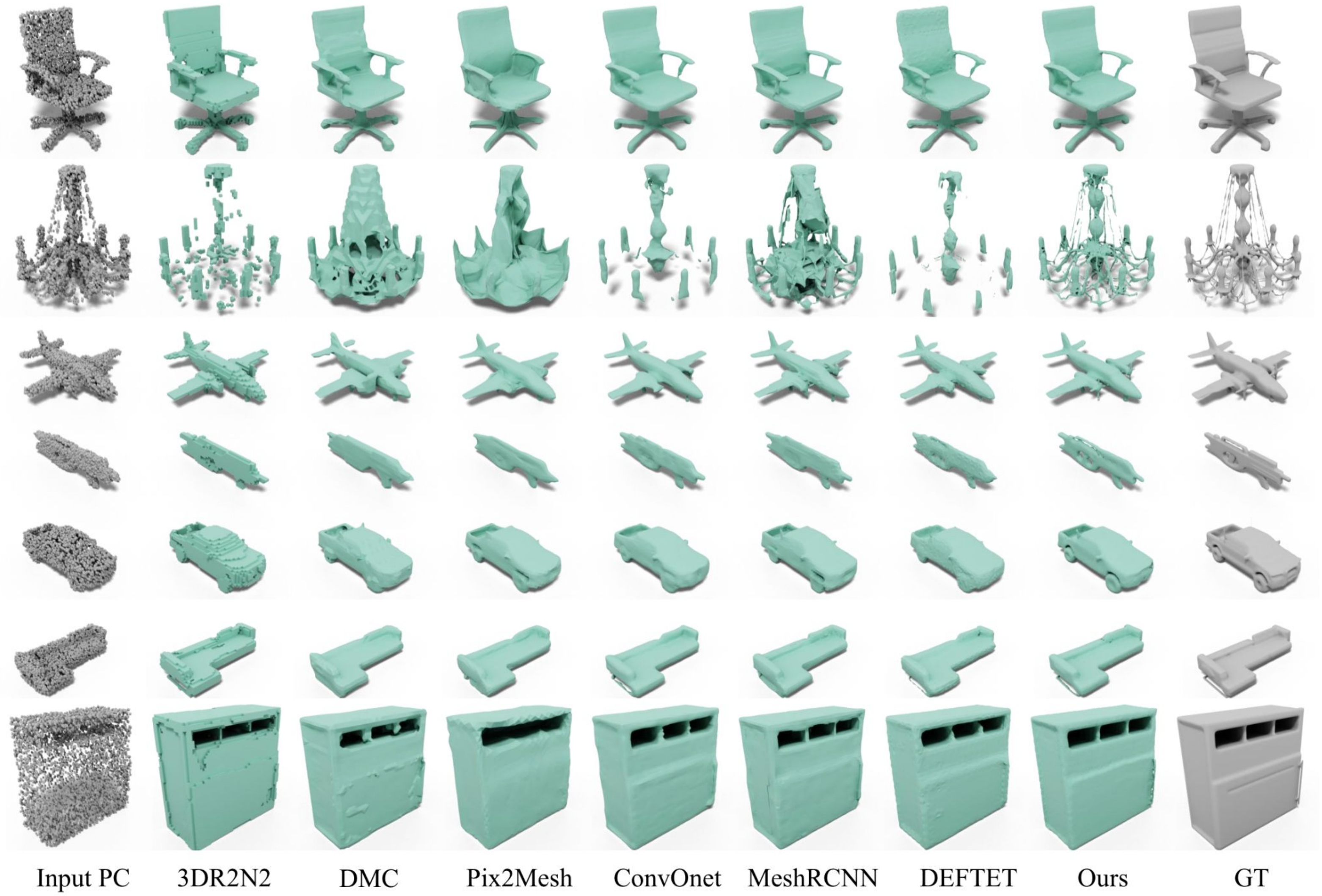
Super-resolution of Animal Shapes



Super-resolution of Animal Shapes

	L2 Chamfer ↓	L1 Chamfer ↓	Norm. Cons. ↑	LFD ↓	Cls ↓
ConvOnet [44]	0.83	2.41	0.901	3220	0.63
DECOR [6]-Retv.	1.32	3.81	0.876	3689	0.66
DECOR [6]-Rand.	2.38	6.85	0.797	5338	0.67
DMTET wo Adv.	0.76	2.20	**0.916**	2846	0.58
DMTET	**0.75**	**2.19**	**0.918**	**2823**	**0.54**

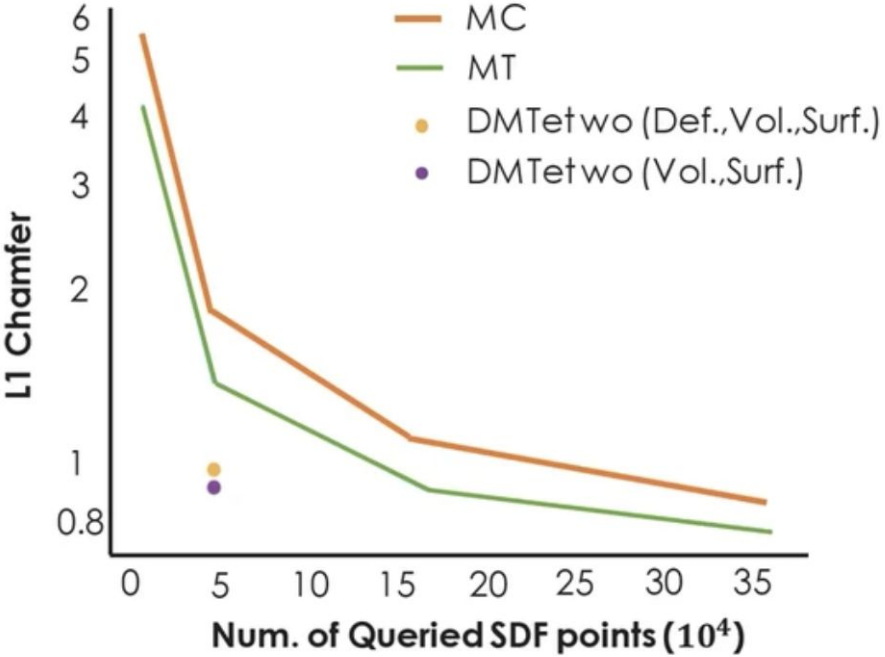
Super Resolution of Animal Shapes: DMTET significantly outperforms all baselines in all metrics.

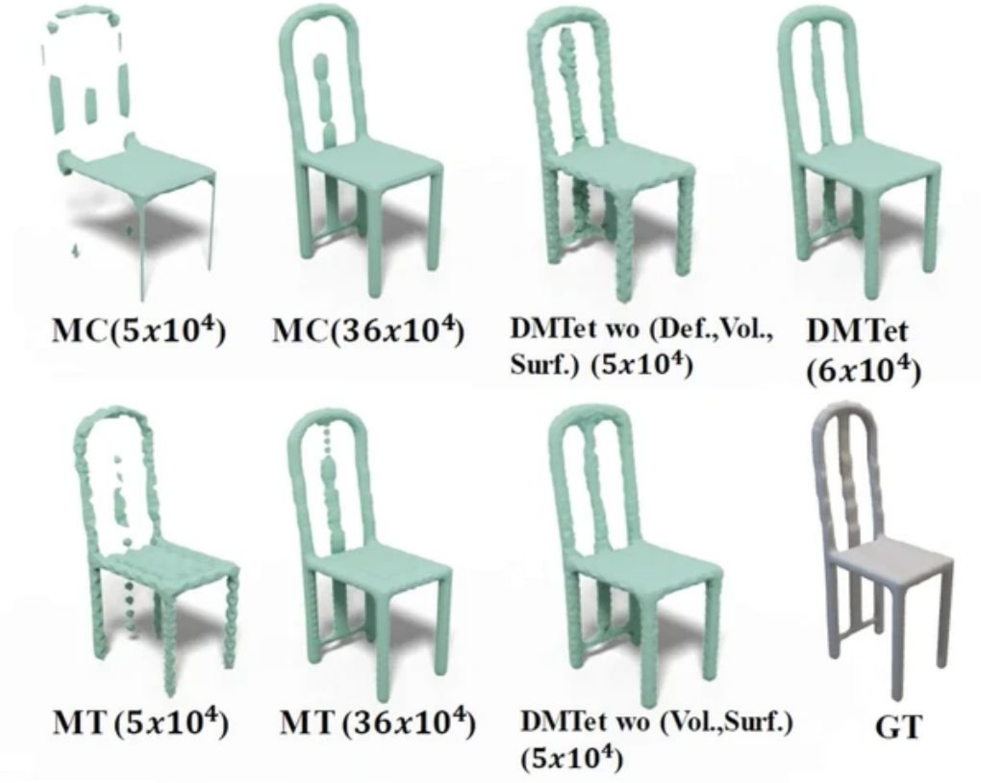

Input PC
3DR2N2
DMC
Pix2Mesh
ConvOnet
MeshRCNN
DEFTET
Ours
GT

Quantitative Results on Point Cloud Reconstruction

Category	Airplane	Bench	Dresser	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Vessel	**Mean**↓	**Time**(ms)↓
3D-R2N2 [10]	1.48	1.59	1.64	1.62	1.70	1.66	1.74	1.74	1.37	1.60	1.78	1.55	1.51	1.61	174
DMC [31]	1.57	1.47	1.29	1.67	1.44	1.25	2.15	1.49	1.45	1.19	1.33	0.88	1.70	1.45	349
Pixel2mesh [54]	0.98	1.28	1.44	1.19	1.91	1.25	2.07	1.61	0.91	1.15	1.82	0.83	1.12	1.35	**30**
ConvOnet [44]	0.82	0.95	0.96	1.12	1.03	0.93	1.22	1.12	0.79	0.91	0.94	0.67	0.99	0.95	866
MeshRCNN [22]	0.88	1.01	1.05	1.14	1.10	0.99	1.20	1.21	0.83	0.96	1.00	0.71	1.03	1.01	228
DEFTET [18]	0.85	0.94	0.97	1.13	1.04	0.92	1.28	1.17	0.85	0.90	0.93	0.65	0.99	0.97	61
DMTET wo (Def, Vol., Surf.)	0.82	0.96	0.94	0.98	0.99	0.90	1.04	1.03	0.80	0.86	0.93	0.65	0.89	0.91	52
DMTET wo (Vol., Surf.)	0.69	0.82	0.88	0.92	0.92	0.82	0.89	0.97	0.65	0.81	0.84	0.61	0.80	0.81	52
DMTET wo Vol.	0.65	0.78	0.84	0.89	0.89	0.79	0.86	0.95	0.61	0.78	0.79	0.60	0.78	0.79	67
DMTET wo Surf.	0.63	0.77	0.84	0.88	**0.88**	0.79	**0.84**	**0.94**	0.60	0.78	0.79	0.59	**0.76**	0.78	108
DMTET	**0.62**	**0.76**	**0.83**	**0.87**	**0.88**	**0.78**	**0.84**	**0.94**	**0.59**	**0.77**	**0.78**	**0.57**	**0.76**	**0.77**	129

Comparison with Oracle Performance of MC/MT





Broad Impact

Many fields such as AR/VR, robotics, architecture, gaming and film rely on high-quality 3D content.

Creating such content, however, requires human experts, i.e., experienced artists, and a significant amount of development time. In contrast, platforms like Minecraft enable millions of users around the world to carve out coarse shapes with simple blocks. This work aims at creating A.I. tools that would enable even novice users to upscale simple, low-resolution shapes into high resolution, beautiful 3D content.

Reference and Further Reading

- Towards Generative Modeling of 3D Objects Learned from Images | Toronto AIR Seminar
    - https://www.youtube.com/watch?v=whXTP08XMYA
- DMTet Kaolin Implementation
    - https://github.com/NVIDIAGameWorks/kaolin/blob/master/examples/tutorial/dmtet_tutorial.ipynb
- Project Page
    - https://research.nvidia.com/labs/toronto-ai/DMTet/
- Paper
    - https://research.nvidia.com/labs/toronto-ai/DMTet/assets/dmtet.pdf