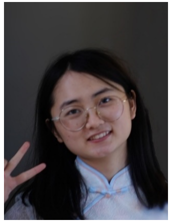
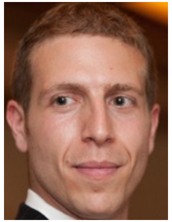
MeshDiffusion: Score-based Generative 3D Mesh Modeling

Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, Weiyang Liu

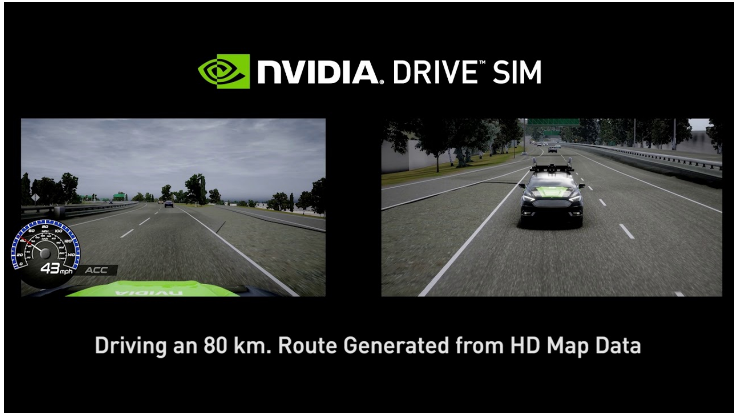


Why 3D Generation?

Creating realistic but diverse set of 3D assets is hard

- Games & movies
- Digital avatar design
- Synthetic environments for robotics

Most 3D assets are built with meshes

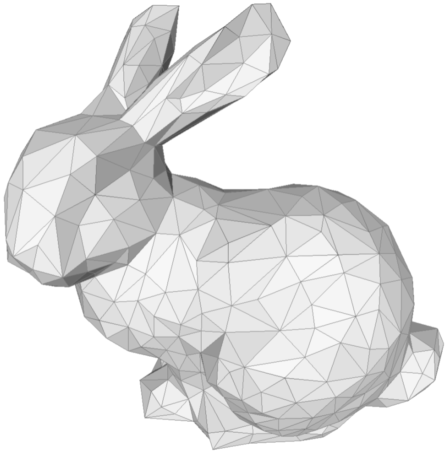


Source: Genshin Impact; https://www.youtube.com/watch?v=fiRcLUuWDD8

3D Meshes

Discretized surfaces with triangles / polygons

+ Easy manipulation (geometry, light, motion)
+ Fast and reliable physics-based rendering



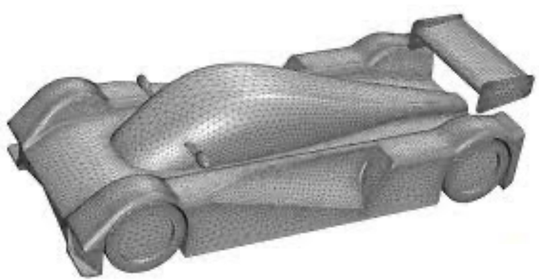
1st citizen in modern graphics pipelines

Goal: to build a **diffusion model** to directly generate **3D meshes**

Source: Stanford bunny, mesh figure from https://arxiv.org/abs/1904.05537

Challenges with Meshes

- No predefined **topology**
- Varying numbers of vertices and faces

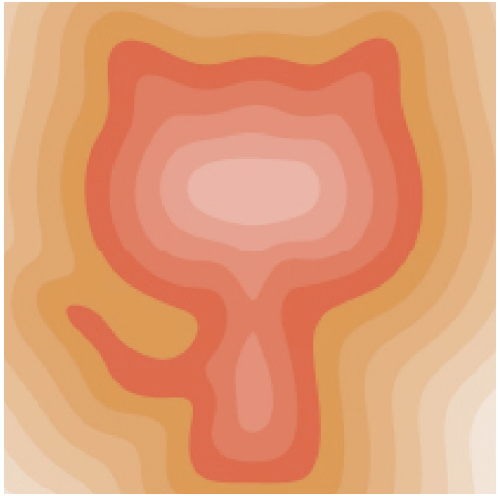


Source: ShapeNet Dataset, https://shapenet.org/

Capture Topology with SDFs

Signed distance field (SDF):

- Scale = Distance to the nearest surface
- Sign = Inside/outside the object

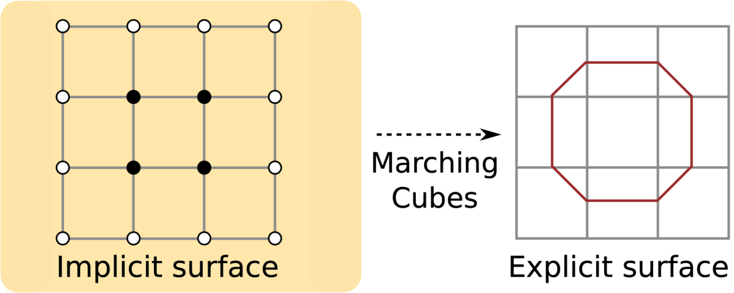


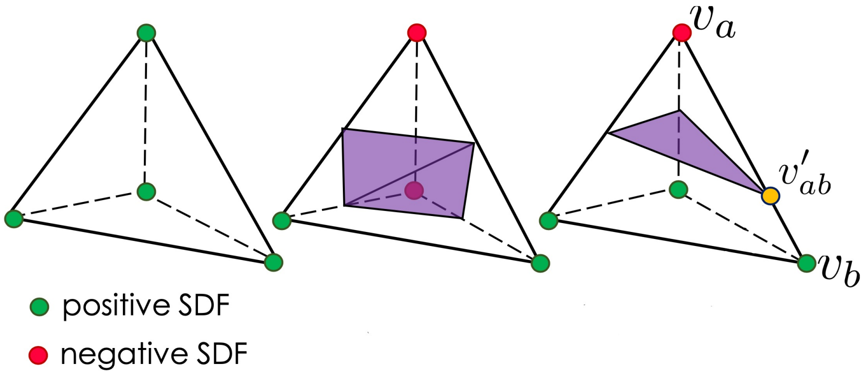
Surface = Zero levelset

Source: https://arxiv.org/abs/2202.08345

SDFs to Meshes

Marching cubes / Marching tetrahedra: 1-to-1 mapping from SDFs to meshes





Source: https://openaccess.thecvf.com/content_cvpr_2018/CameraReady/1704.pdf, https://nv-tlabs.github.io/DMTet/

Parametrizing Meshes

Deep Marching Tetrahedra (DMTet): Parametrize meshes with deformable tetrahedral grids

- Deformation = details without higher resolution
- Deformed tetrahedra are still tetrahedra

Model Objective

Multiview Images

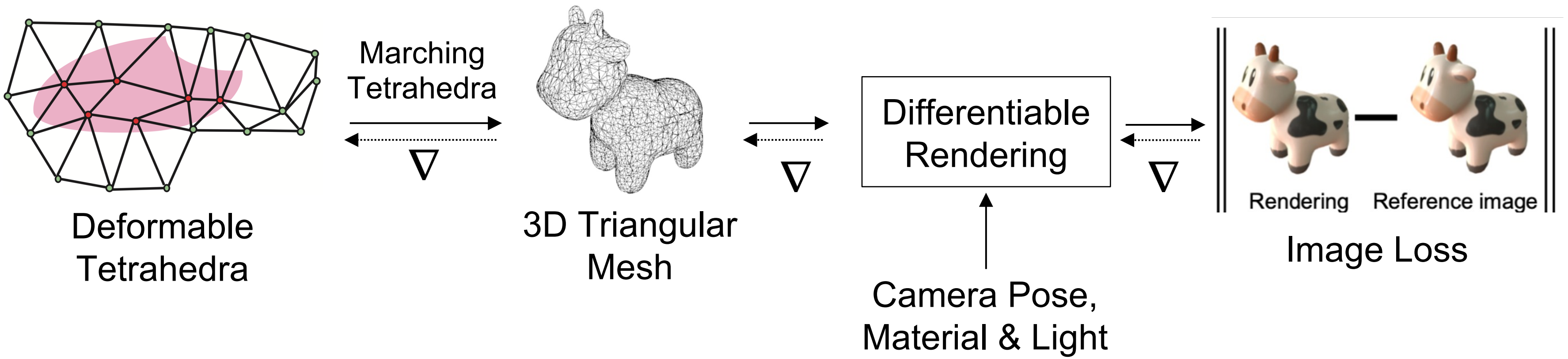
$$\mathcal{L} = \mathop{\mathbb{E}}_{i\in[N]} \mathcal{L}_{\text{diffusion}}(x_i) \quad s.t. \quad x_i = \arg\min_{x} \underbrace{\mathcal{L}_{\text{Render}}(x, \{y_i^{(k)}\})}_{\text{Render Loss}}$$

DMTet

For simplicity, follow a two-stage process:

Create a DMTet dataset ⟶ Train a diffusion model

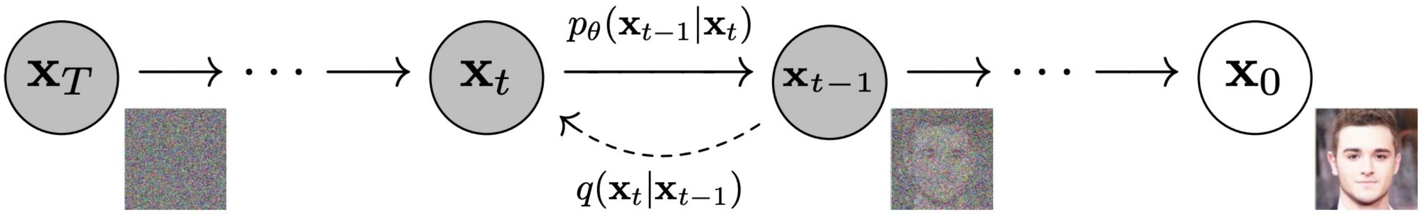
Create a DMTet dataset



Source: Adapted from https://nv-tlabs.github.io/nvdiffrec/

Recap: Diffusion Model

Key idea: model the generation process as a denoising process



Learning objective: denoising autoencoder

$$\mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

Noisy input

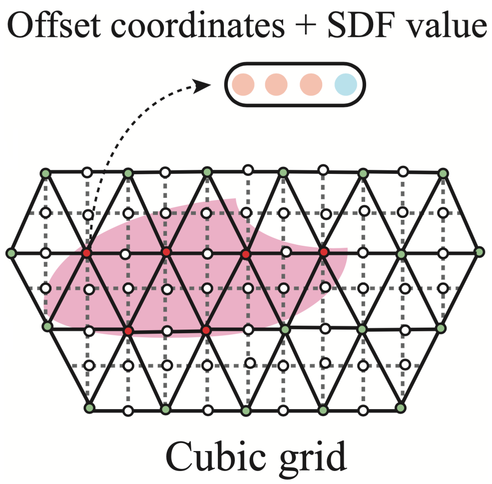
Noise prediction U-Net

Source: Denoising Diffusion Probabilistic Models, https://arxiv.org/abs/2006.11239

Convolutional U-Net on DMTet

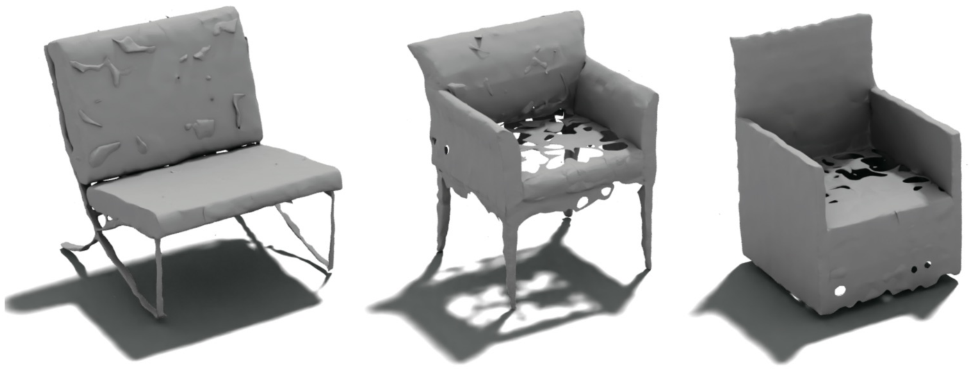
Translational invariance in DMTet → use convolution

- Reimplementing convolutions for tetrahedral grids
- Augment tetrahedral grids to cubic grids → 3D CNN



Uneven Surfaces due to Nonlinearity

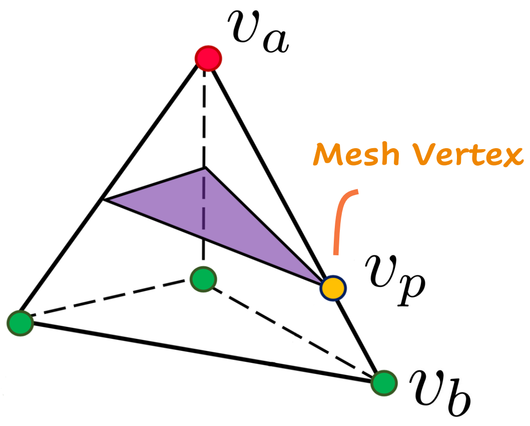
A naïve implementation results in uneven or broken generated surfaces



Uneven Surfaces due to Nonlinearity

Create v_p if s_a and s_b (the SDFs of v_a and v_b) have different signs

$$v_p = \frac{v_a|s_b| + v_b|s_a|}{|s_a| + |s_b|}$$

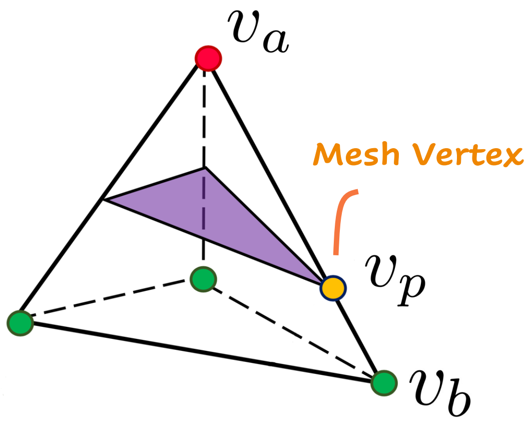


Source: Figure adapted from https://nv-tlabs.github.io/DMTet/assets/dmtet.pdf

Uneven Surfaces due to Nonlinearity

Create v_p if s_a and s_b (the SDFs of v_a and v_b) have different signs

$$v_p = \frac{v_a|s_b| + v_b|s_a|}{|s_a| + |s_b|}$$



Suppose $s_b < 0 < s_a$. With an identical noise on both s_a and s_b :

$$v_{p,\text{noisy}} - v_p = \underbrace{\frac{\epsilon}{|s_a| + |s_b|}}_{\text{Unknown Scale}}(v_b - v_a) \qquad (0 < \epsilon < |s_b|)$$

Unknown Scale

→ Varies at different locations in different data points

Source: Figure adapted from https://nv-tlabs.github.io/DMTet/assets/dmtet.pdf

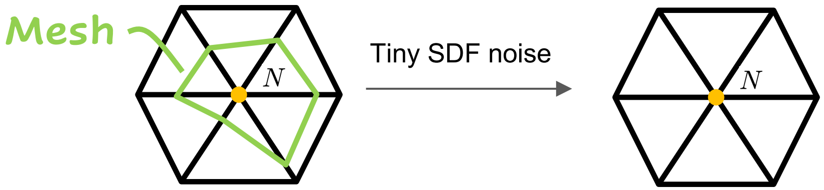
Uneven Surfaces due to Nonlinearity

Similarly, consider:

- A vertex N with a negative SDF value s_N close to zero, but
- All surrounding vertices with large positive SDF values

A small perturbation on s_N

$\rightarrow$ a topological change but negligible L2 loss



Uneven Surfaces due to Nonlinearity

Lower denoising loss on SDFs $\neq$ Lower prediction loss on mesh vertex positions

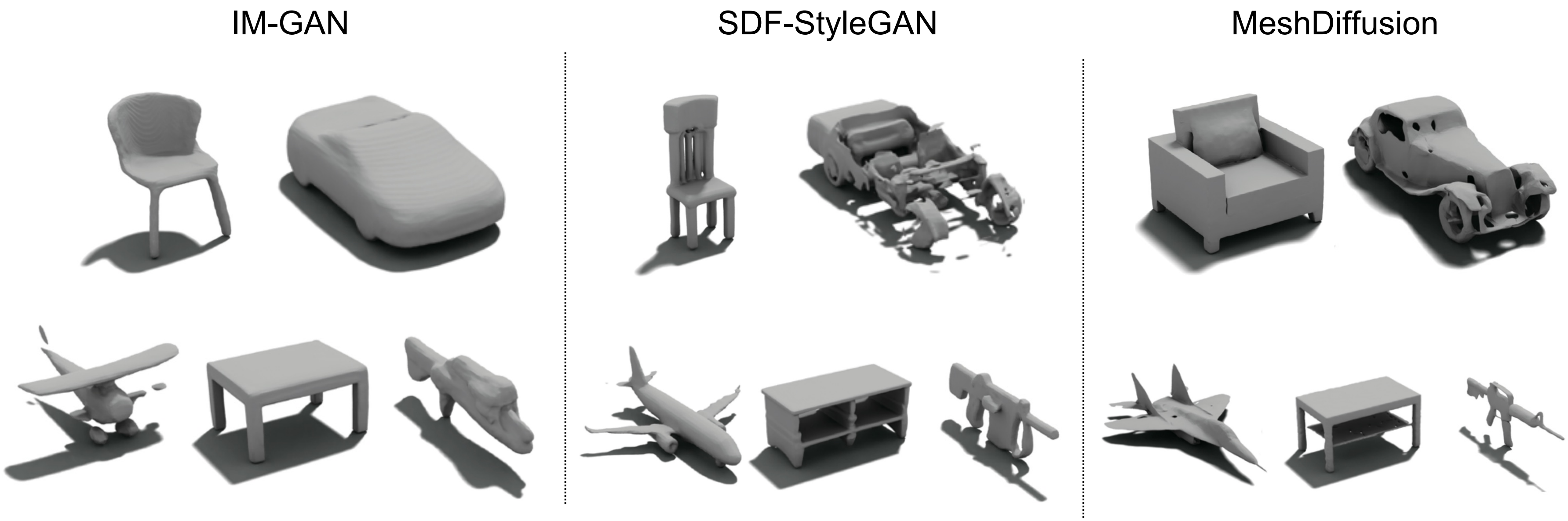
$\neq$ Good topological prediction

Solution: Normalize SDF values on all tetrahedral vertices to ± 1 by rounding

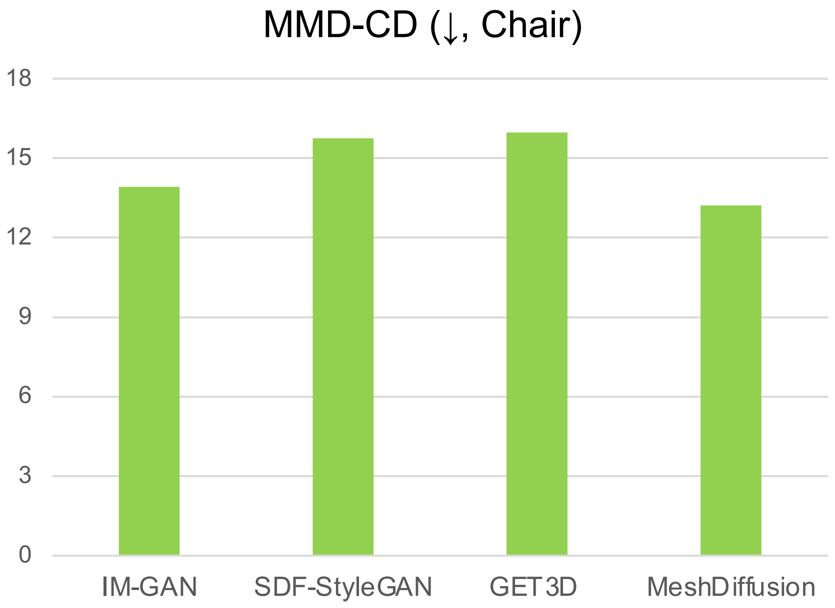
- Finetune offsets in the DMTet dataset after normalization

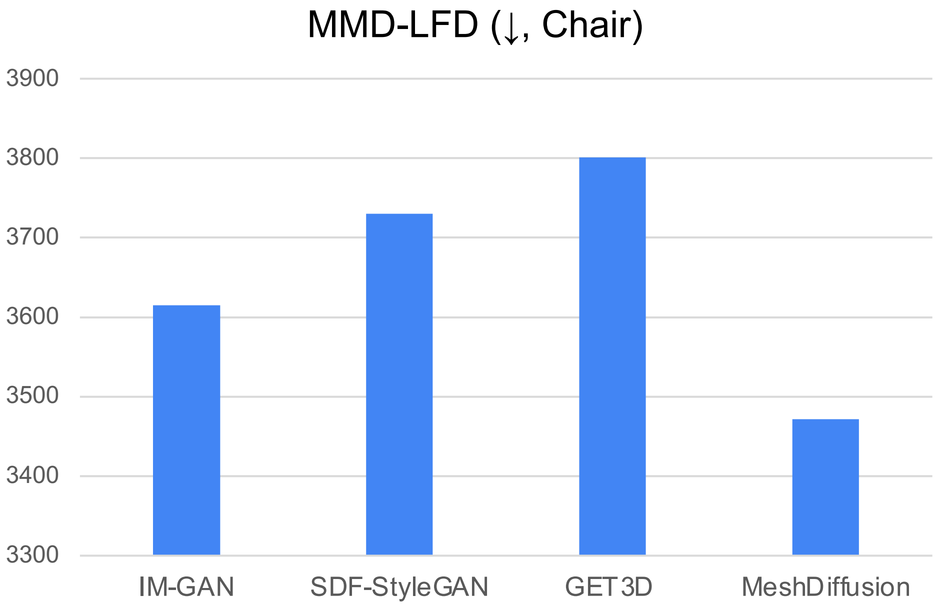
Unconditional Generation

MeshDiffusion 1) produces sharper edges and 2) is less prone to catastrophic failures



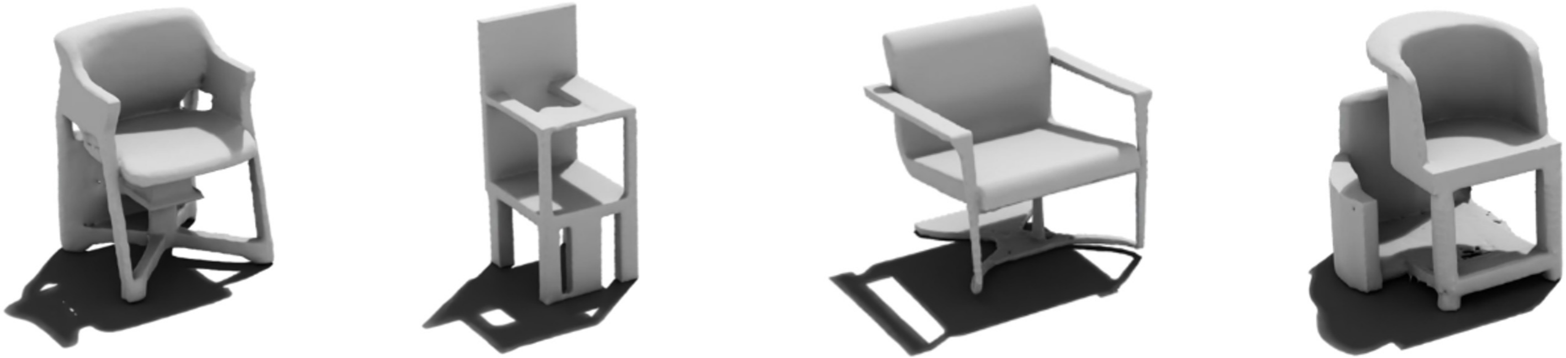
Quantitative Results





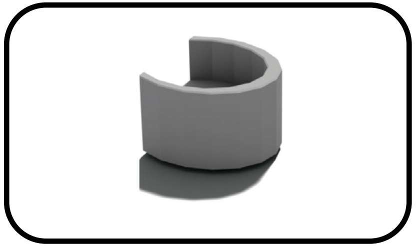
Hallucinated Samples

Not reasonable in the sense of affordance but geometry



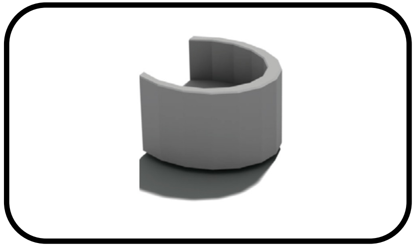
Single-view Conditional Generation

Given RGBD view



Single-view Conditional Generation

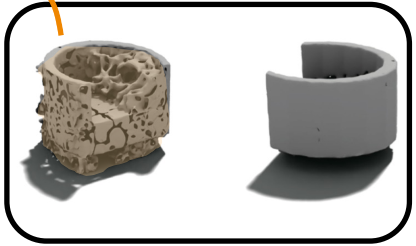
Given RGBD view



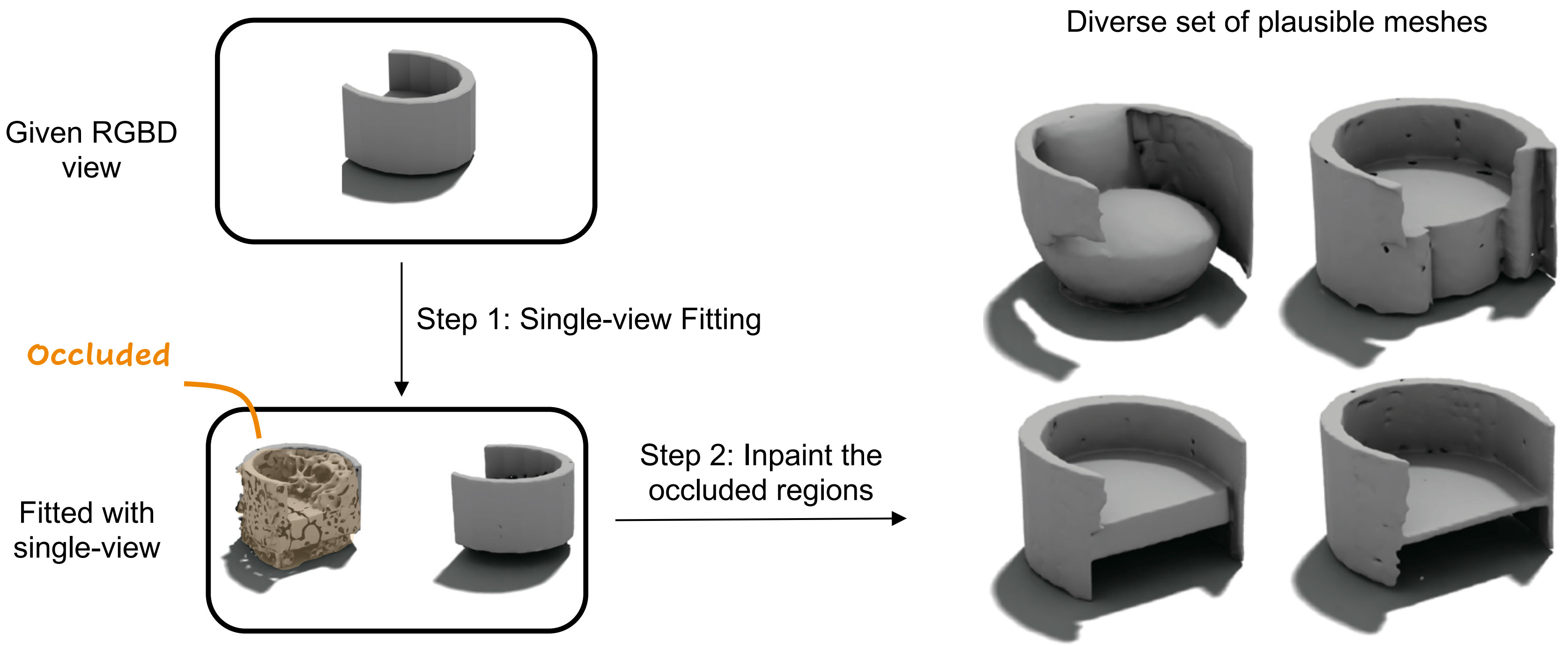
Step 1: Single-view Fitting

Occluded

Fitted with single-view



Single-view Conditional Generation

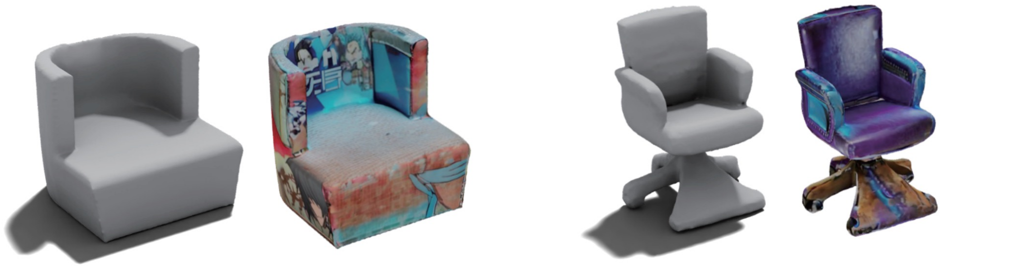


Interpolation

Using DDIM inference, we can treat the initial noises as latent codes

Text-to-Texture

May use SOTA methods for text-to-texture synthesis



A sofa with an anime character   A blue and purple leather swivel chair

A StarWars jet   A WWI style British plane

Method used: https://texturepaper.github.io/TEXTurePaper/

Thank you!

Project Page GitHub

Project page:

https://meshdiffusion.github.io

Github repo:

https://github.com/lzzcd001/MeshDiffusion/

Check our poster @ MH1-2-3-4 #161