清華大學 Tsinghua University
交叉信息研究院 Institute for Interdisciplinary Information Sciences

Lecture 9: 3D Generative Models

Li Yi

Apr 17, 2025

Recap

- Problem Definition
- Explicit Algorithms
  - Ball-Pivoting
- Implicit Algorithms
  - Estimate an implicit field function from data
  - Extract the zero iso-surface

Radial Basis Functions

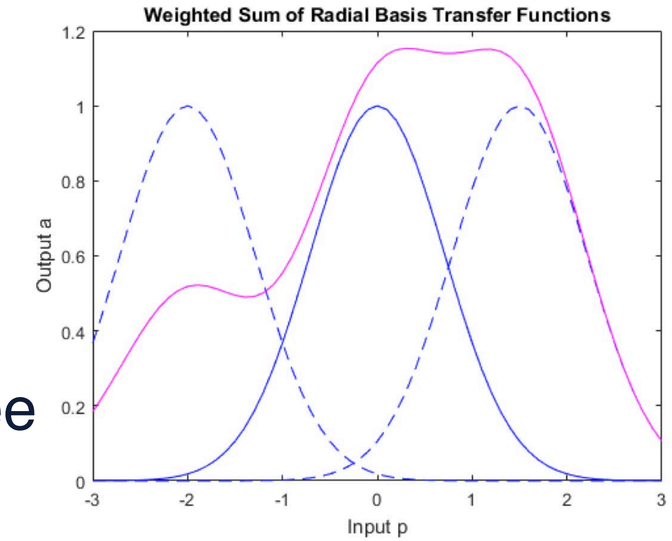
- Radial basis functions (RBF) $\phi_c(\mathbf{x})$: function value depends only on the distance from a center point c

- Use a weighted sum of radial basis functions to approximate the shape:

$$\phi_c(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$$

$$f(\mathbf{x}) = \sum_{i=1}^{n} \omega_i \phi\left(\|\mathbf{x} - \mathbf{x}_i\|\right) + p(x)$$

where p is a polynomial of low degree

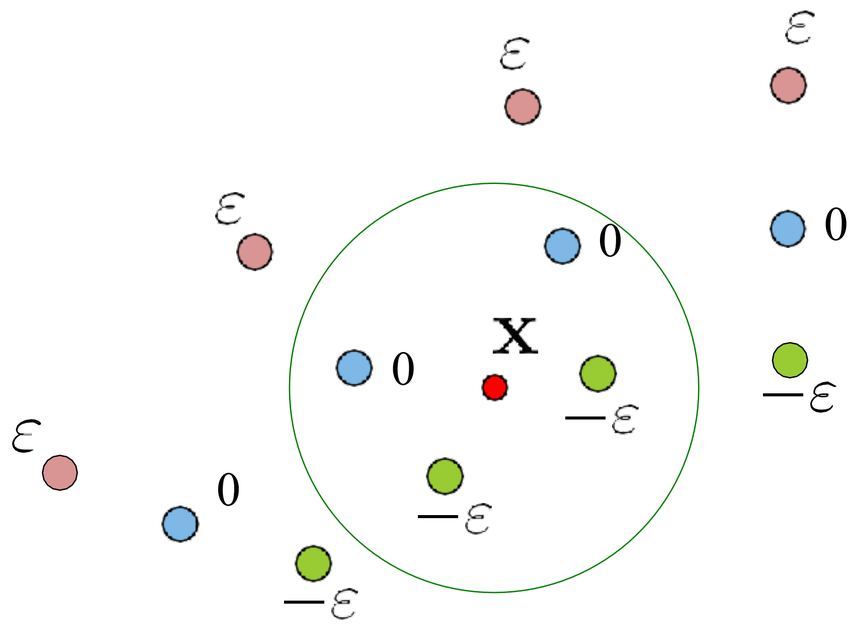


http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Moving Least Squares (MLS)

Do purely **local** approximation of the SDF

Weights change depending on where we are evaluating

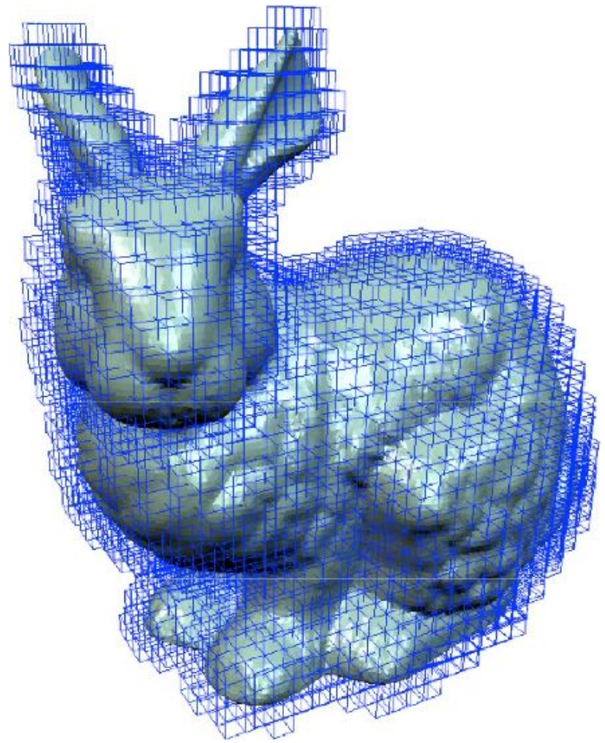


"Interpolating and Approximating Implicit Surfaces from Polygon Soup", Shen et al., ACM SIGGRAPH 2004
http://graphics.berkeley.edu/pap3e/rs1//S2h0e1n8-IAI-2004-08/index.html

Implicit Meshing Algorithm

- Two basic steps:
  1. Estimate an implicit field function from data
  2. **Extract the zero iso-surface**

Input: a signed distance field
(Implicitly assumed knowing the inside/outside of the shape,
often needs to be estimated with **normal** information)



http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

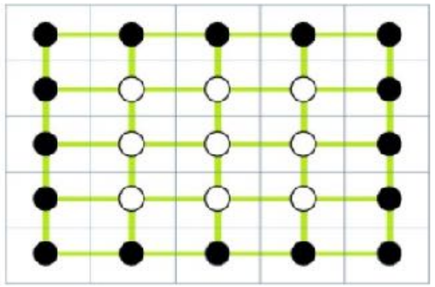
Classical Solution: 2D Marching Square

1	1	1	1	1
1	2	3	2	1
1	3	3	3	1
1	2	3	2	1
1	1	1	1	1

Threshold with iso-value

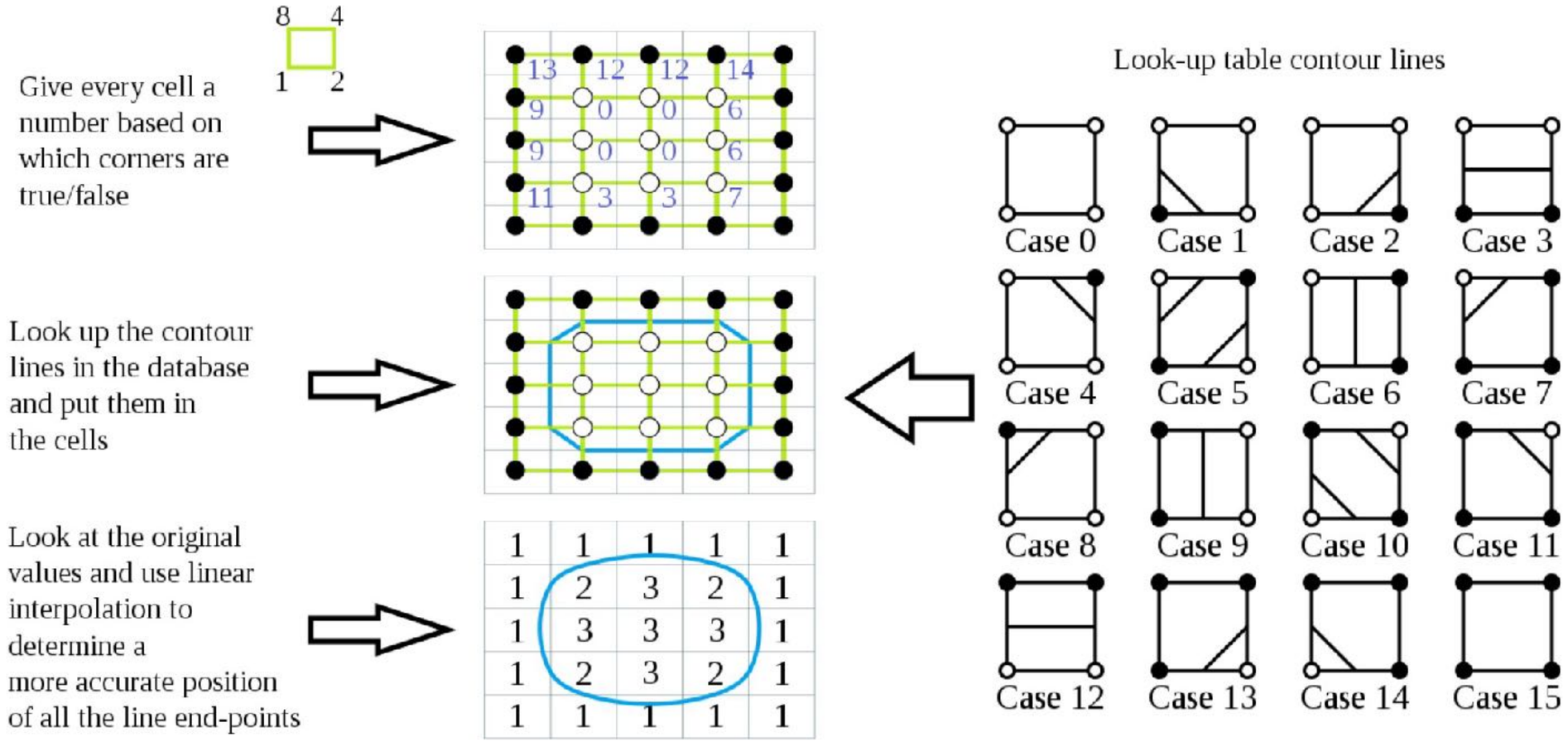
0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Binary image to cells



https://en.wikipedia.org/wiki/Marching_squares

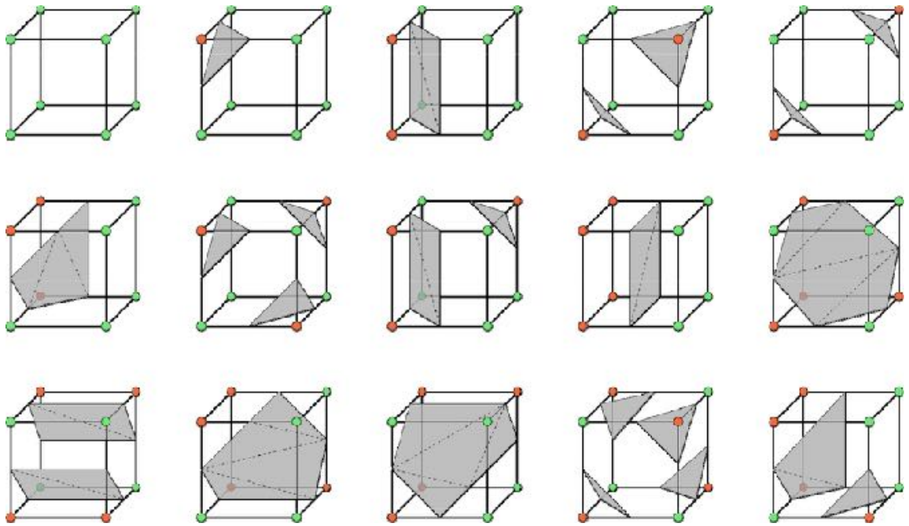
Classical Solution: 2D Marching Square

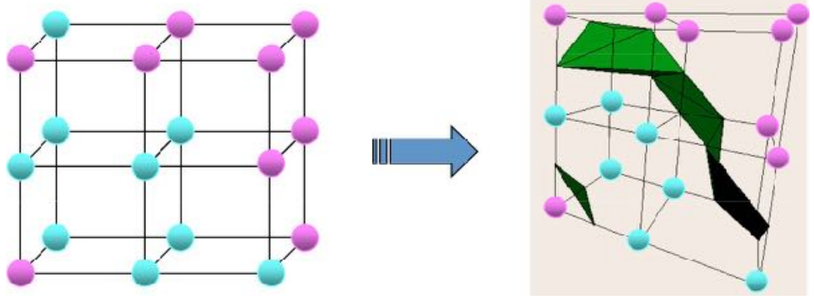


https://en.wikipedia.org/wiki/Marching_squares

Classical Solution: 3D Marching Cube

- $2^8 = 256$ cases
- The first published version exploits rotation and inversion, and only considers 15 unique cases:

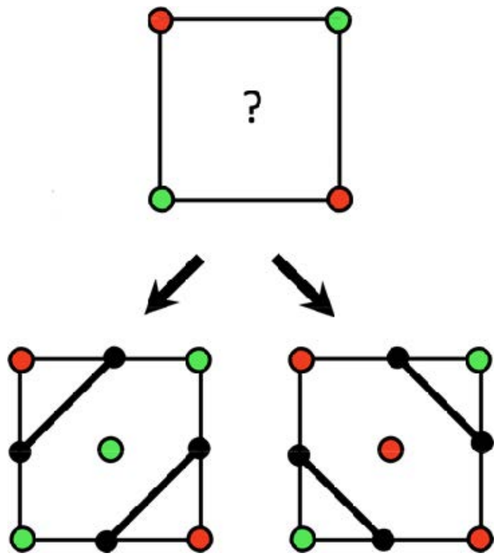


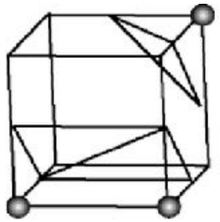


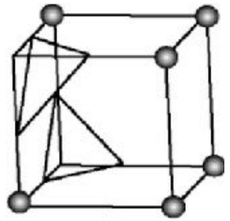
http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

Ambiguity

- Ambiguity leads to holes:



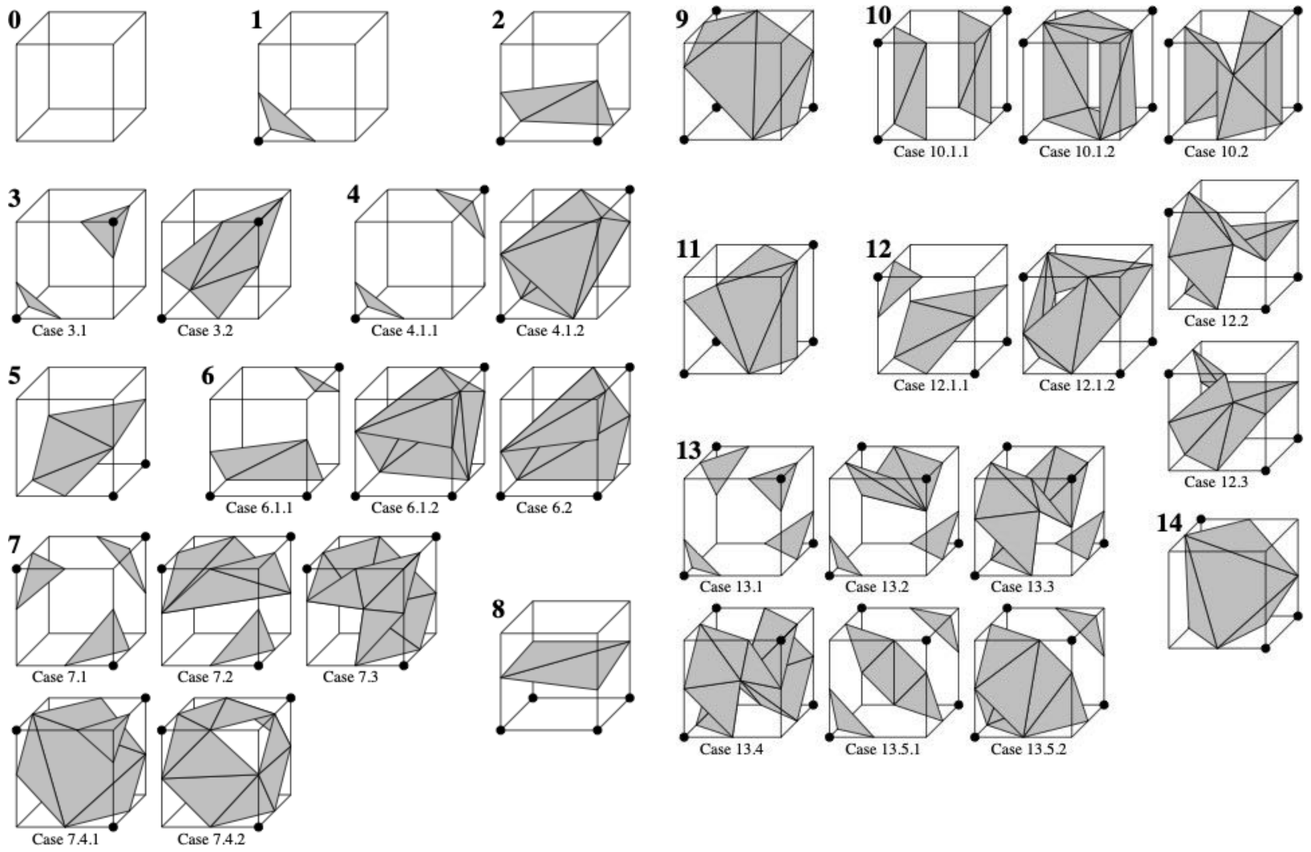


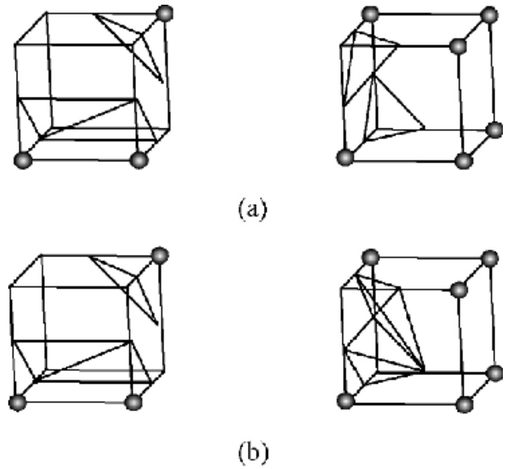


http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

Solution to Ambiguity

- Considering more cases in the look-up table by watching larger context:





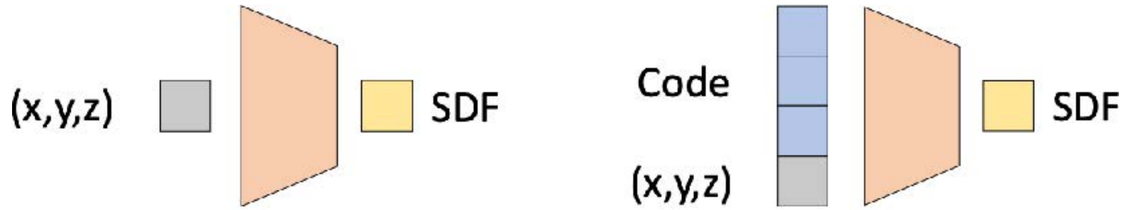
Chernyaev et al., "M**arching cubes 33: Construction of topologically correct isosurfaces.**", 1995

Comparison

	Explicit meshing (e.g., ball-pivoting)	Implicit meshing (e.g., RBF, MLS, Poission)
Sensitive to normals	No	Yes
Watertight manifold	No	Yes, in most cases
Complexity	Linear	• Large-scale equations to estimate implicit function • Marching cubes • Dense voxelization

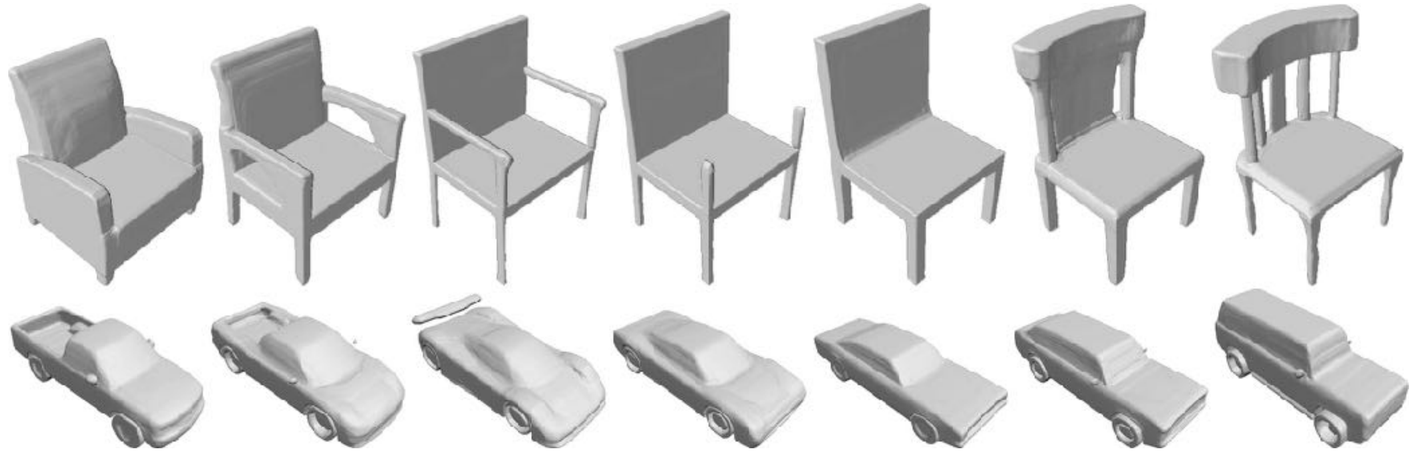
Slides credit: Roi Poranne

Use Neural Network to Approximate Implicit Field Function



(a) Single Shape DeepSDF **(b)** Coded Shape DeepSDF

- (a) use the network to overfit a single shape
- (b) use a latent code to represent a shape, so that the network can be used for multiple shapes



Park et al., "**DeepSDF: Learning continuous signed distance functions for shape representation.**", *CVPR 2019*

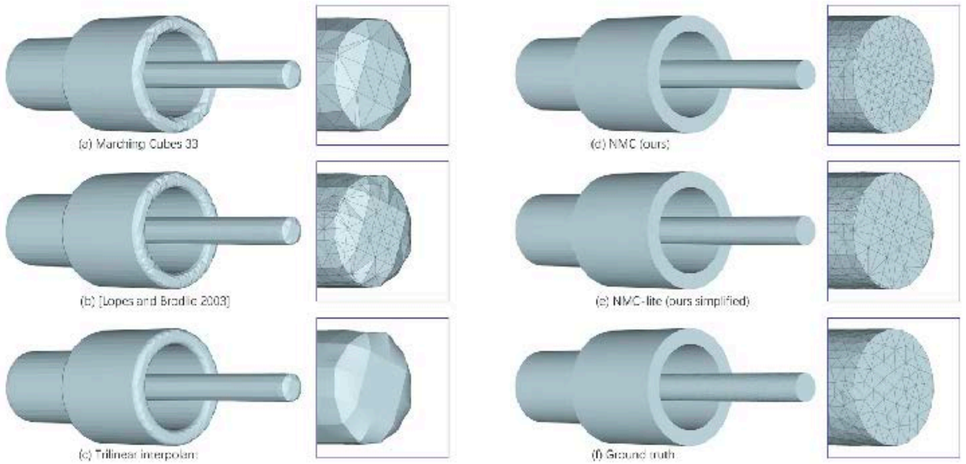
Learning-Based Marching Cube

Deep Marching Cubes: Learning Explicit Surface Representations

Yiyi Liao[1,2] Simon Donné[1,3] Andreas Geiger[1,4]
[1]Autonomous Vision Group, MPI for Intelligent Systems and University of Tübingen
[2]Institute of Cyber-Systems and Control, Zhejiang University
[3]imec - IPI - Ghent University [4]CVG Group, ETH Zürich
{yiyi.liao,simon.donne,andreas.geiger}@tue.mpg.de

Neural Marching Cubes

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



Neural Dual Contouring

ZHIQIN CHEN, Simon Fraser University, Canada
ANDREA TAGLIASACCHI, Google Research, University of Toronto, Canada
THOMAS FUNKHOUSER, Google Research, USA
HAO ZHANG, Simon Fraser University, Canada

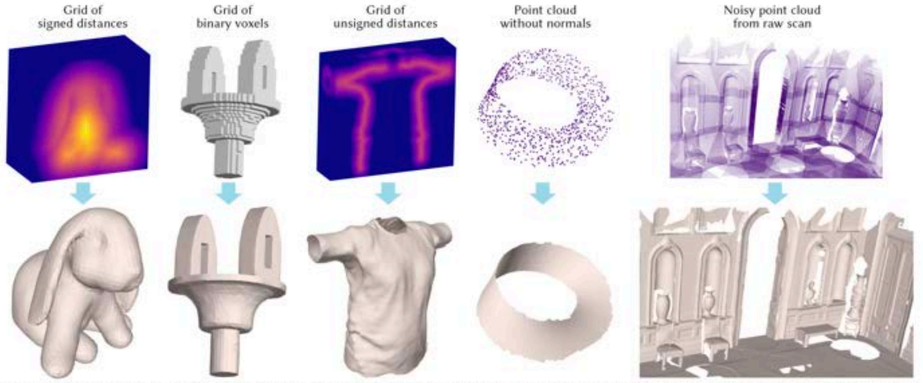


Fig. 1. Neural dual contouring (NDC) is a unified data-driven approach that learns to reconstruct meshes (bottom) from a variety of inputs (top): signed or unsigned distance fields, binary voxels, non-oriented point clouds, and noisy raw scans. Trained on CAD models, NDC generalizes to a broad range of shape types: CAD models with sharp edges, organic shapes, open surfaces for cloths, scans of indoor scenes, and even the non-orientable Mobiüs strip.

Consistent Orientation?

- Require the ground-truth ***signed*** implicit functions during training (e.g., signed distance, occupancy)

- However, 3D raw data, such as point cloud or triangle soup, are not necessarily consistently oriented
  - e.g., getting normal orientation for ShapeNet data is not easy

Sign Agnostic Learning of Shapes from Raw Data

- Unsigned distance is easy to obtain
  - Distance to the point cloud & triangle soup

- Learn ***signed*** distance from ***unsigned*** **distance** groundtruth
  - Require a special loss function

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Sign Agnostic Learning

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim D_\chi} \tau \left(f(\boldsymbol{x}; \boldsymbol{\theta}), h_\chi(\boldsymbol{x}) \right)$$

- $\chi \subset \mathbb{R}^3$: input raw data (*e.g.*, a point cloud or a triangle soup)
- $f(x; \theta) : \mathbb{R}^3 \times \mathbb{R}^m \rightarrow \mathbb{R}$: learned signed function
- D_χ: distribution of the training samples defined by χ
- $h_\chi(\boldsymbol{x})$: some *unsigned* distance measure to χ
- $\tau : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$: a similarity function

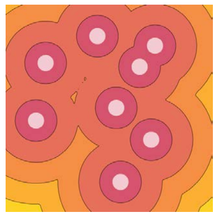
Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Sign Agnostic Learning

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D_\chi}\tau\left(f(\boldsymbol{x};\boldsymbol{\theta}), h_\chi(\boldsymbol{x})\right)$$

- $h_\chi(\boldsymbol{x})$: some *unsigned* distance measure to χ

$$h_2(\boldsymbol{z}) = \min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{z}-\boldsymbol{x}\|_2$$



$$h_0(z) = \begin{cases} 0 & z \in \mathcal{X} \\ 1 & z \notin \mathcal{X} \end{cases}$$

- $\tau : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$: a similarity function

$$\tau_\ell(a,b) = \|a| - b|^\ell$$

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Two Local Minima

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim D_\chi} \tau \left(f(\boldsymbol{x}; \boldsymbol{\theta}), h_\chi(\boldsymbol{x}) \right)$$

- f is an ***unsigned*** function that resembles $h_\chi(\boldsymbol{x})$
- f is a ***signed*** function and $|f|$ resembles $h_\chi(\boldsymbol{x})$
- We prefer the second case to use marching cube

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Two Local Minima

- Pick a special weight initialization θ^0 so that $f\left(\boldsymbol{x};\boldsymbol{\theta}^0\right) \approx \varphi(\|\boldsymbol{x}\| - r)$, where $\varphi(\|\boldsymbol{x}\| - r)$ is the signed distance function to an r-radius sphere
  - $f > 0$ if $\|x\| > r$
  - $f < 0$ if $\|x\| < r$

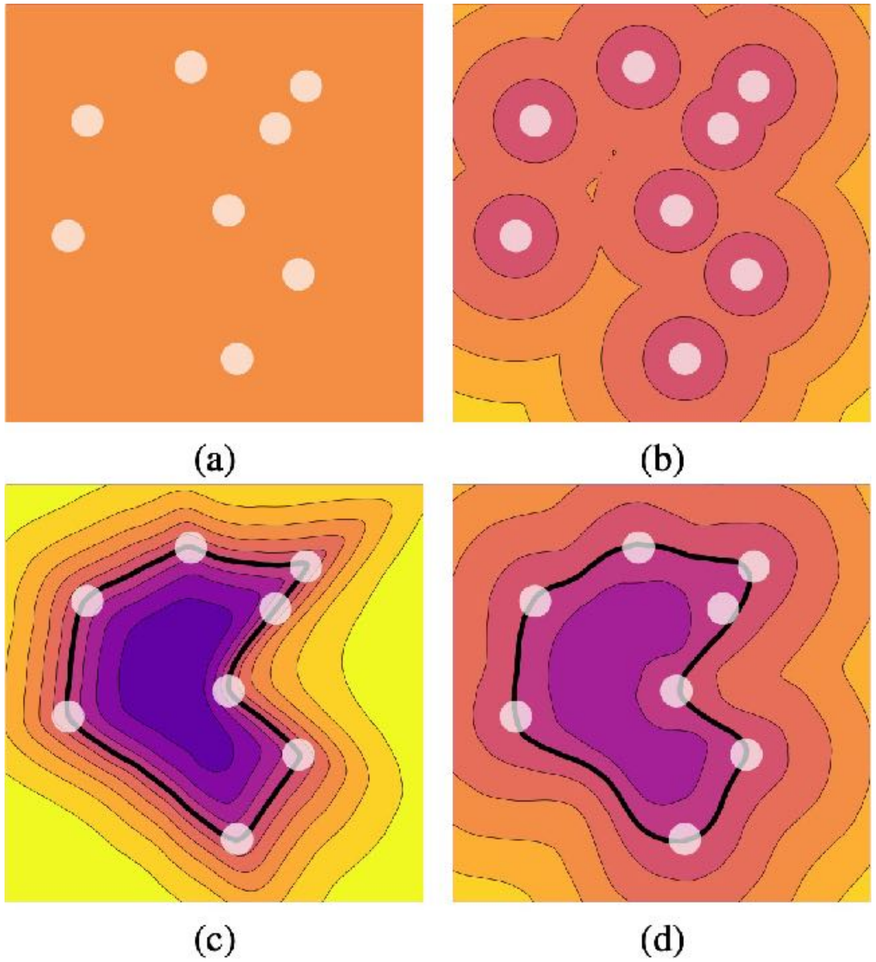
- Under such an initialization, f is not easy to converge to the unsigned local minima.

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

2D Results

Unsigned function as supervision

Learned signed function



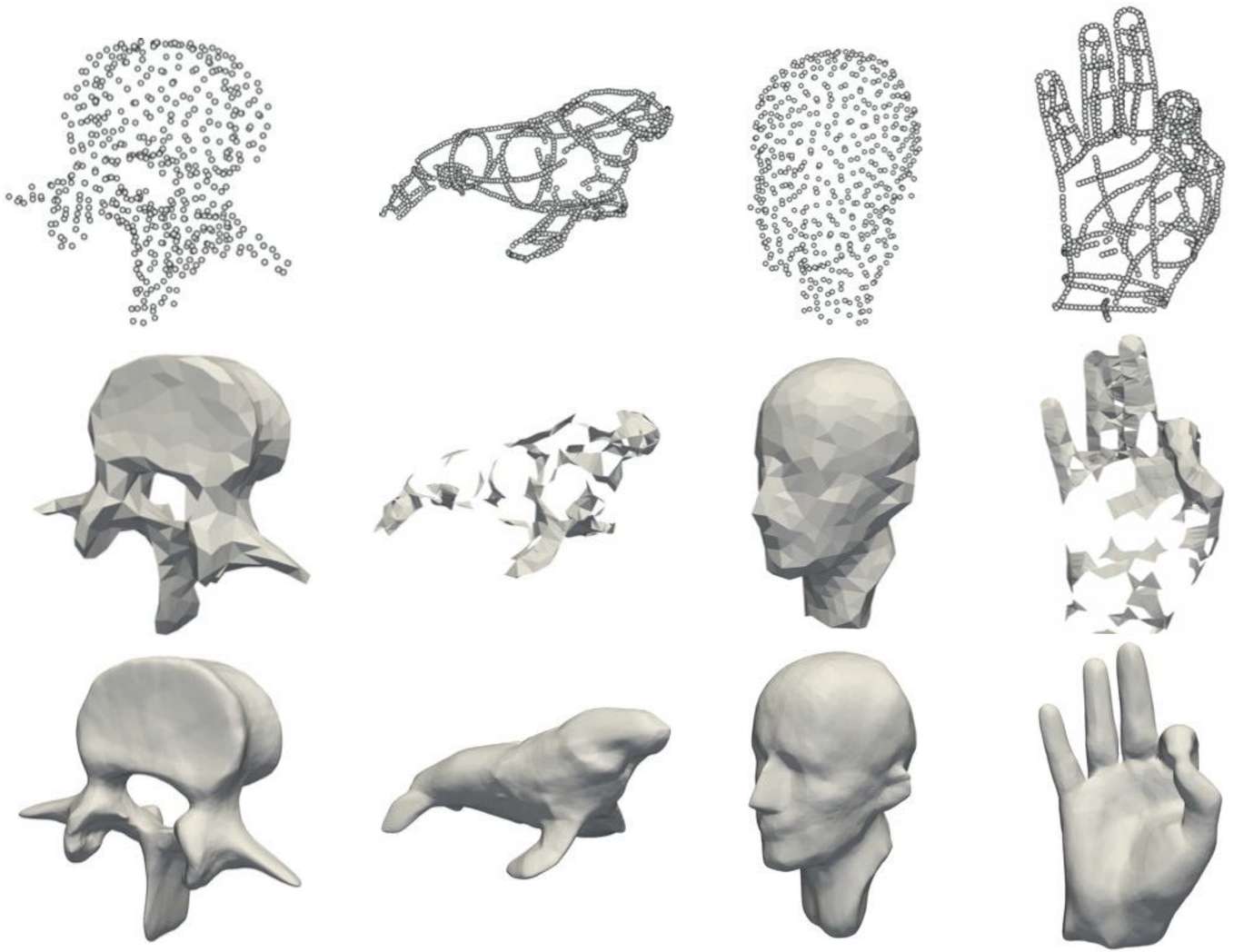
Atzmon et al., “**Sal: Sign agnostic learning of shapes from raw data.**”, *CVPR 2020*

Surface Reconstruction Results from Raw Data

Raw
Point Cloud

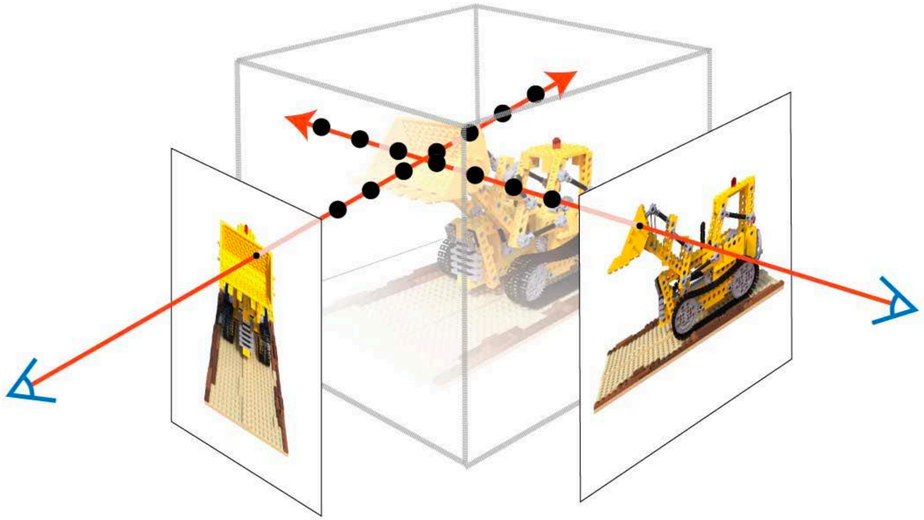
Ball-Pivoting
Algorithm

SAL



Atzmon et al., “**Sal: Sign agnostic learning of shapes from raw data.**”, *CVPR 2020*

Recap: Single-View / Multi-View 3D



From Image(s) to 3D

Today's Focus

a DSLR photo of a squirrel

3D from scratch (Unconditional Generation)

Learn the space of (possibly textured) shapes

Should be able to **sample** from the learned shape space

3D from scratch (Unconditional Generation)

Why is this an important task?

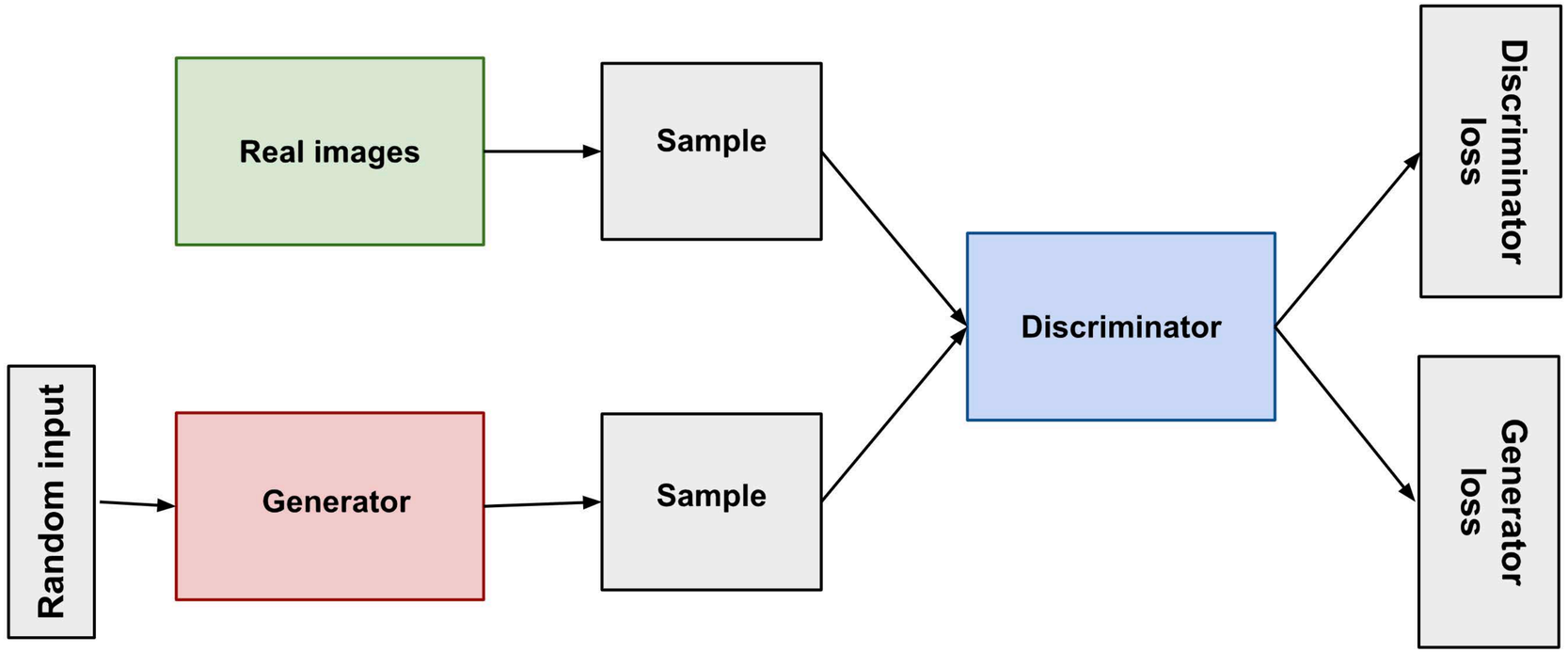
Can re-use the learned shape space for downstream conditional inference

Useful for artists exploring

Outline

- 3D GAN
- 3D Autoregressive Models
- 3D Diffusion Models
- 3D Generation without 3D Training Data
- Part-based 3D Generation

Background: Generative Adversarial Networks



$$L_{GAN}(G, D) = E_y[log\ D(y) + E_{x,z}[log(1 - D(G(x, z))]$$

Learning Objective: Generate output that is indistinguishable from a ‘real’ example

StyleGAN2

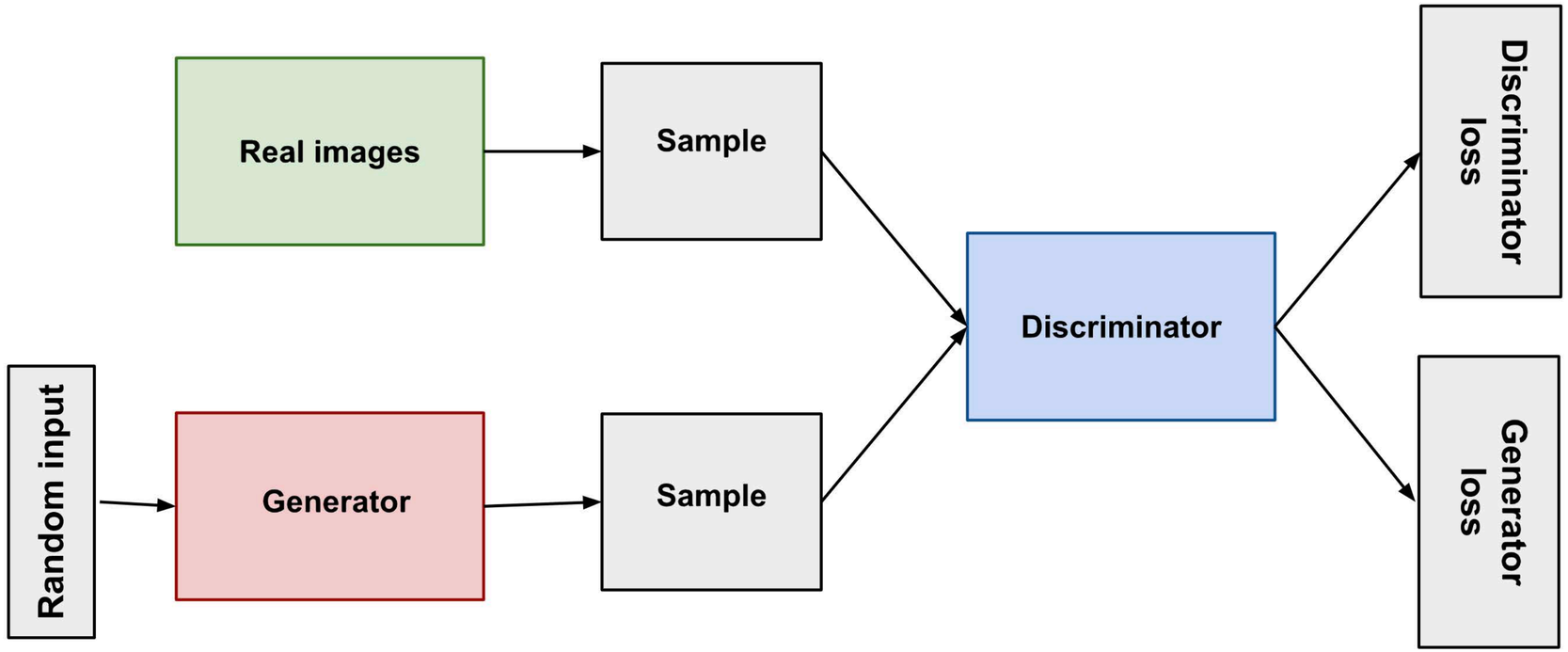
StyleGAN3 (Ours)

Random latent walk using directions from StyleCLIP, GANSpace, and SeFa.


StyleGAN2
StyleGAN3 (Ours)


StyleGAN2
StyleGAN3 (Ours)

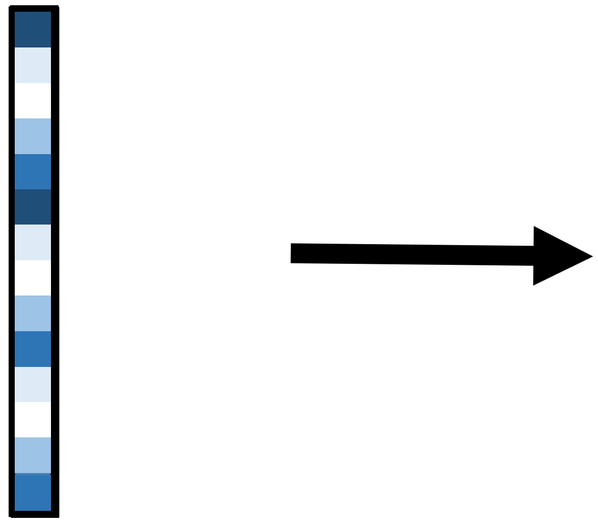
Background: Generative Adversarial Networks



$$L_{GAN}(G, D) = E_y[log\ D(y) + E_{x,z}[log(1 - D(G(x, z))]$$

Can we similarly learn to generate 3D?

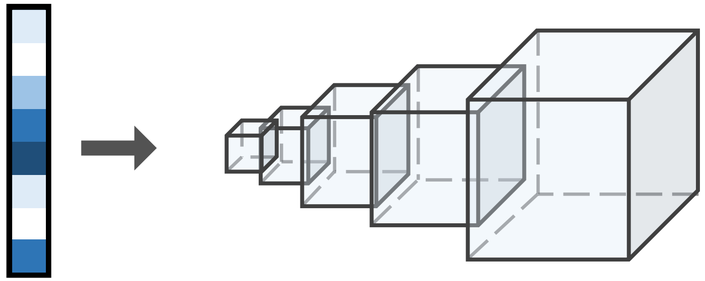
3D GANs



Latent vector

Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

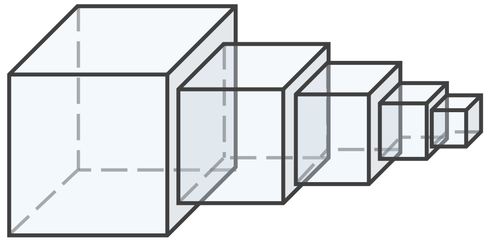
3D GANs



Latent vector

Generator

Generated shape



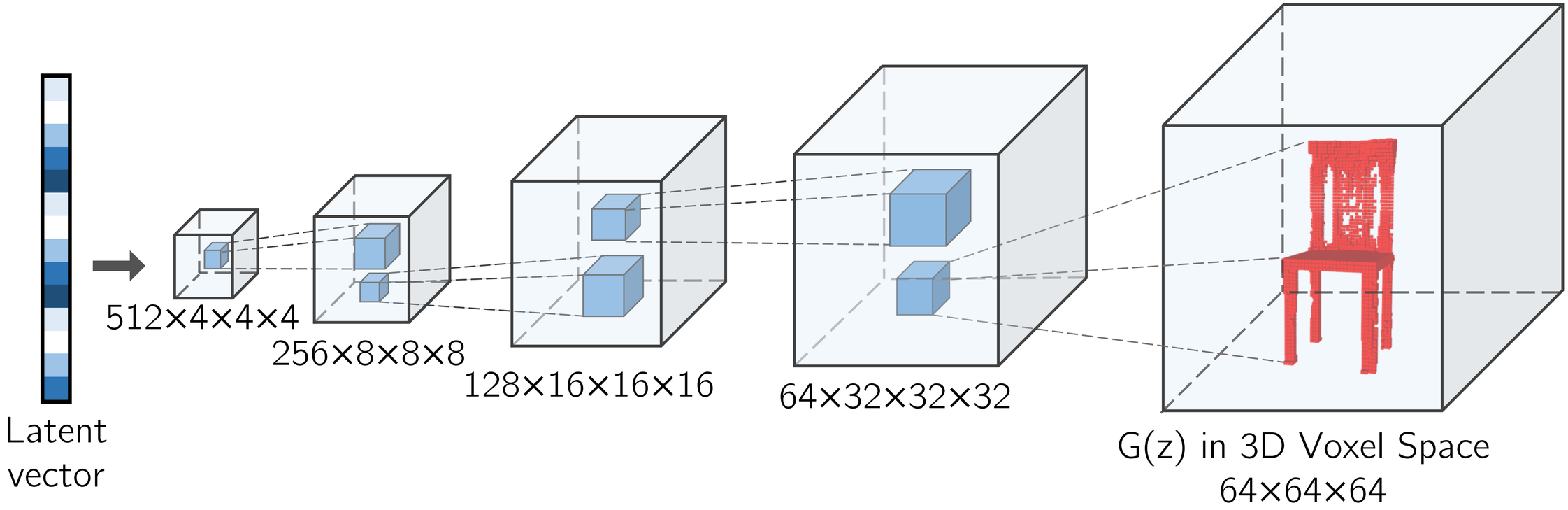
Discriminator

Real?

Real shape (sampled from ShapeNet)

Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

3D GAN: Generator



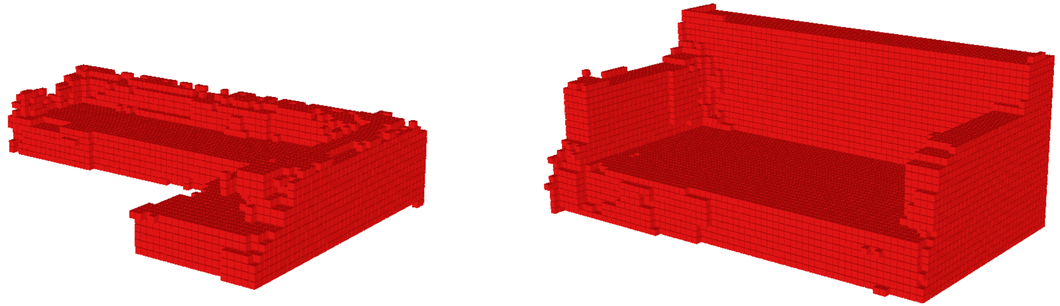
Incrementally increase resolution via convolutions and upsampling layers

Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

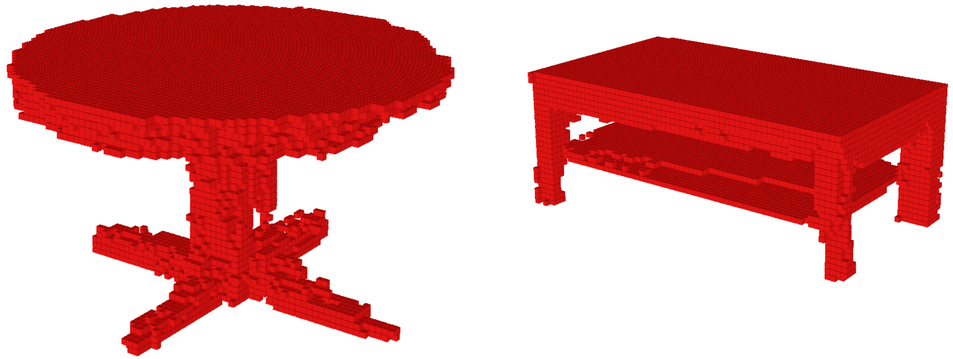
3D GAN: Sampled Shapes

Chairs

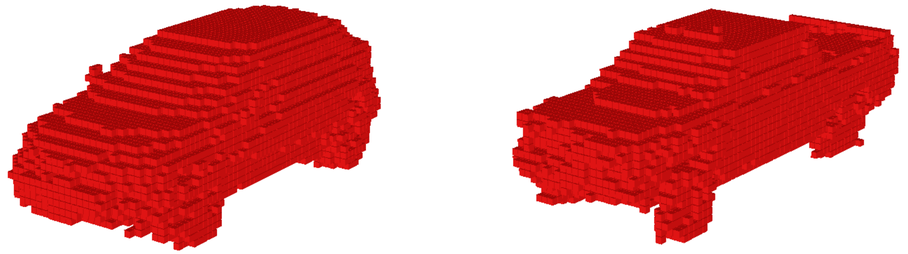
Sofas



Tables



Cars

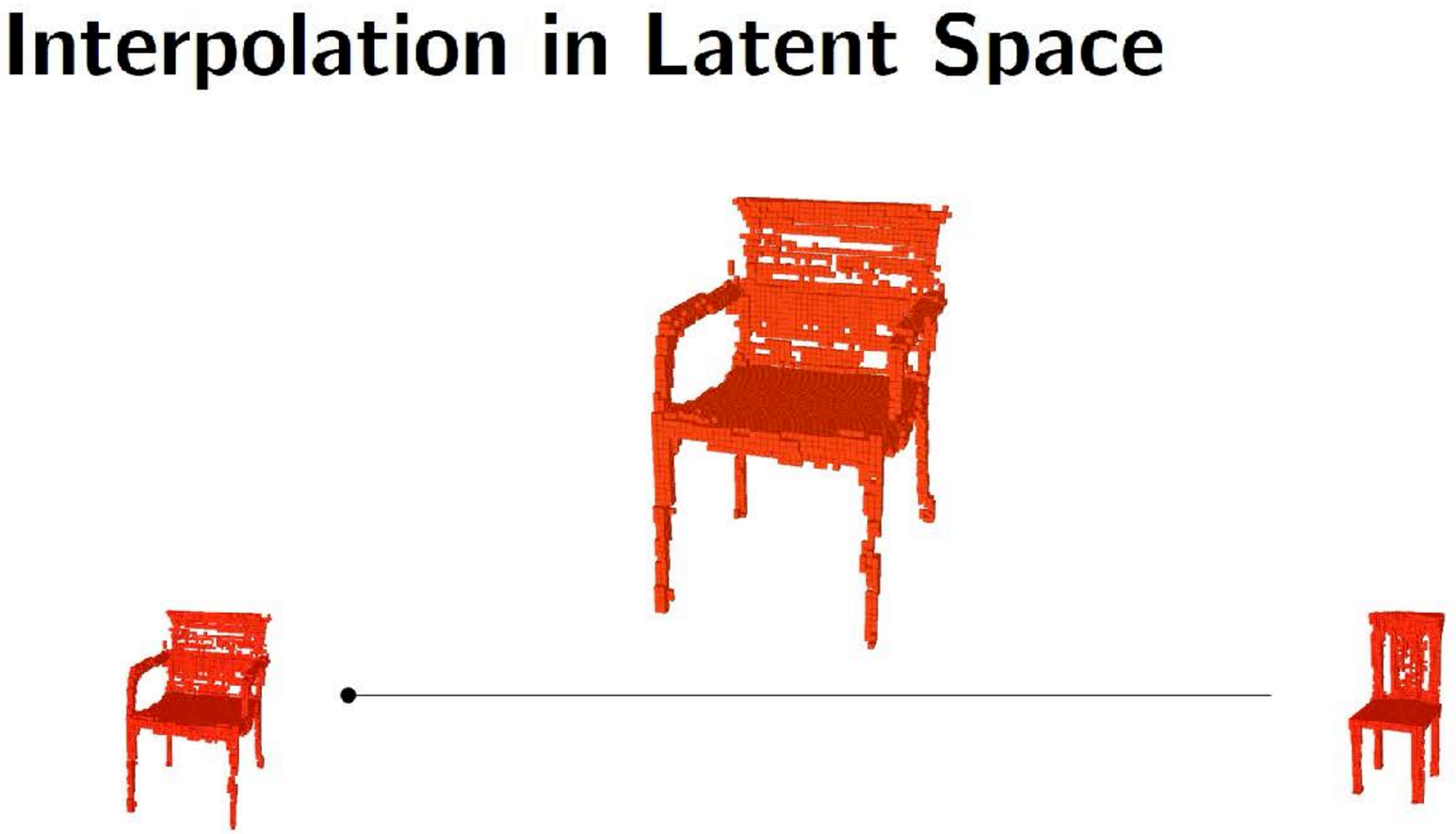


Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

3D GAN: Sampled Shapes

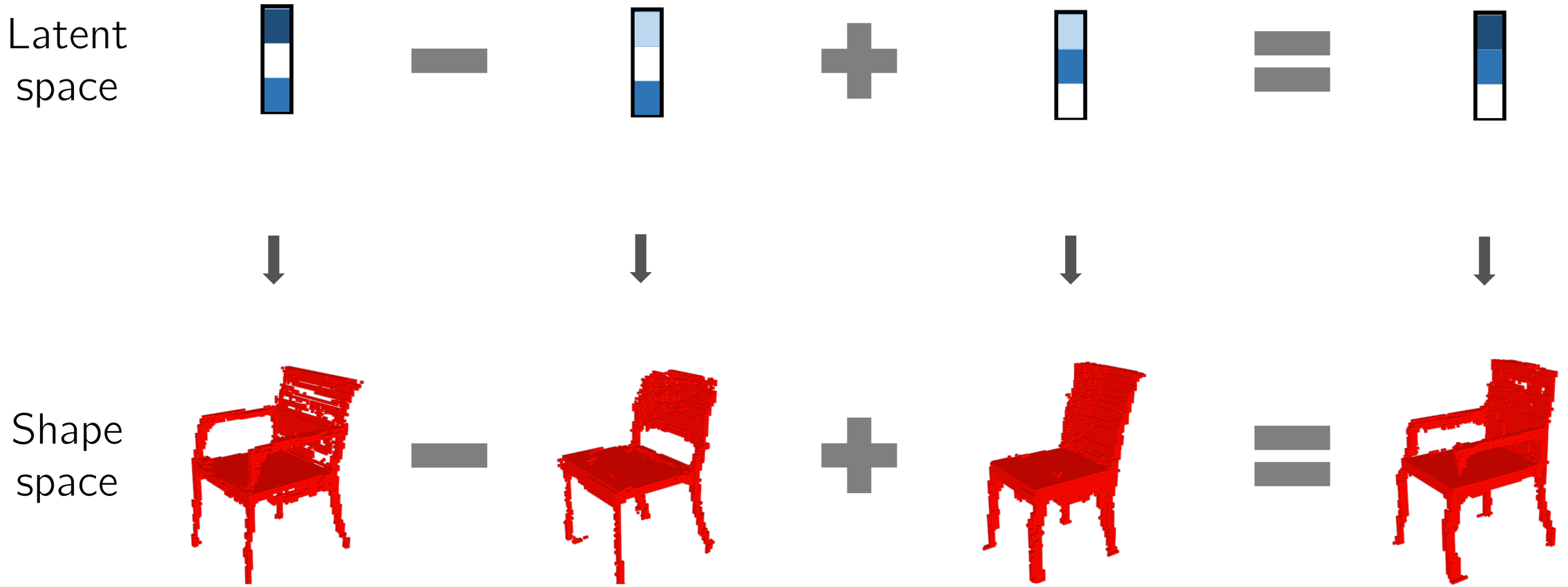
Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

3D GAN: Vector Arithmetic



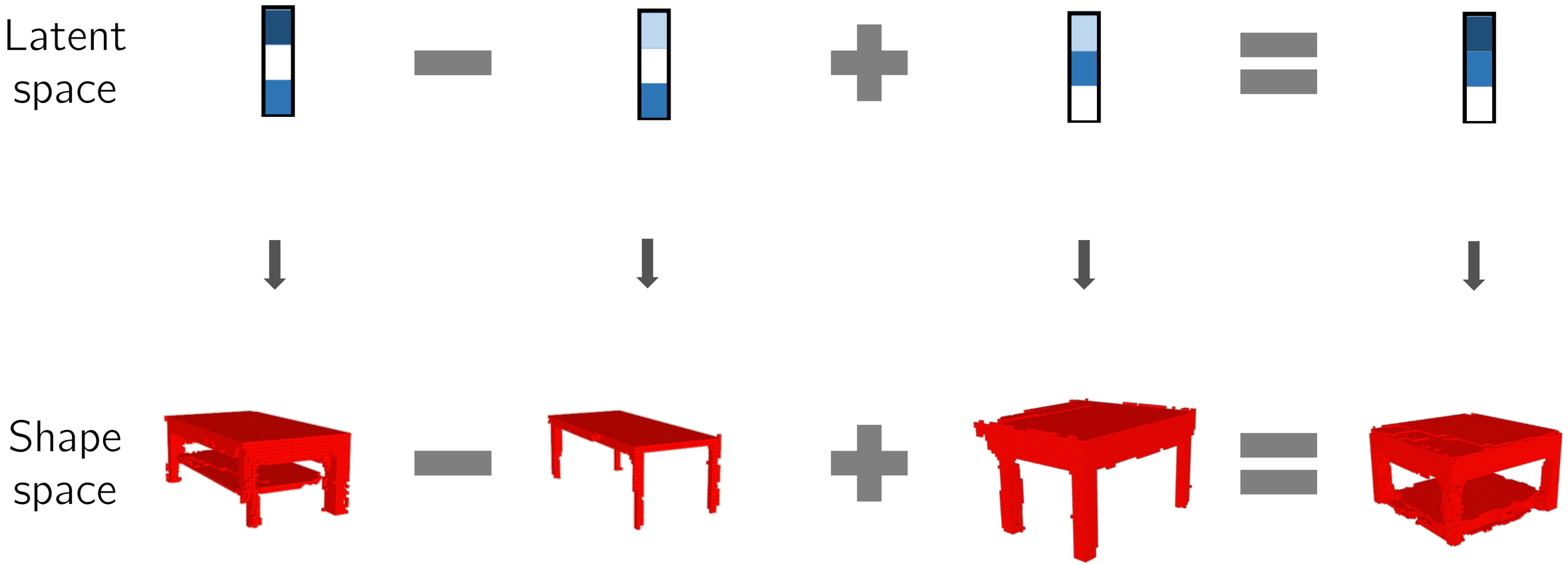
Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

3D GAN: Vector Arithmetic



Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

3D GAN: Vector Arithmetic

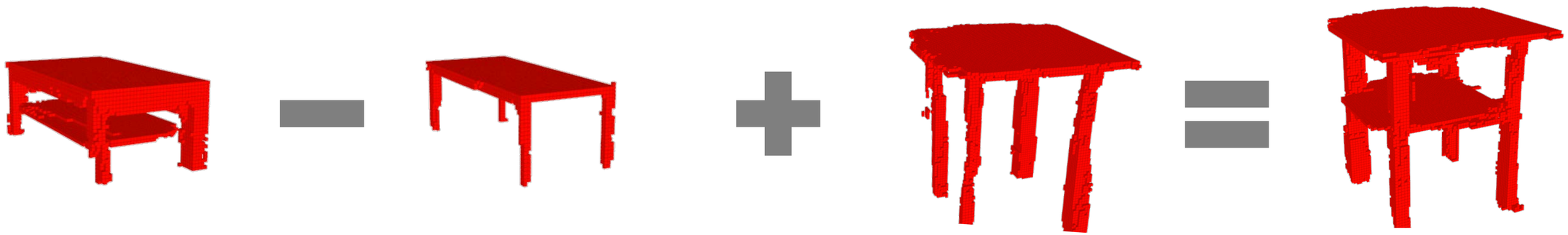


Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

3D GAN: Vector Arithmetic

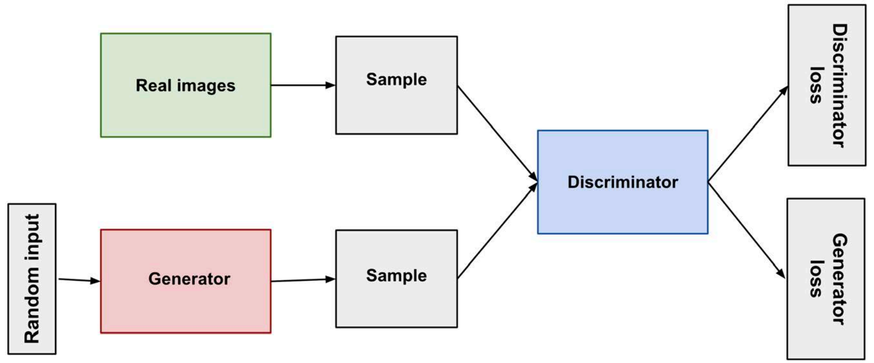
Latent space

Shape space



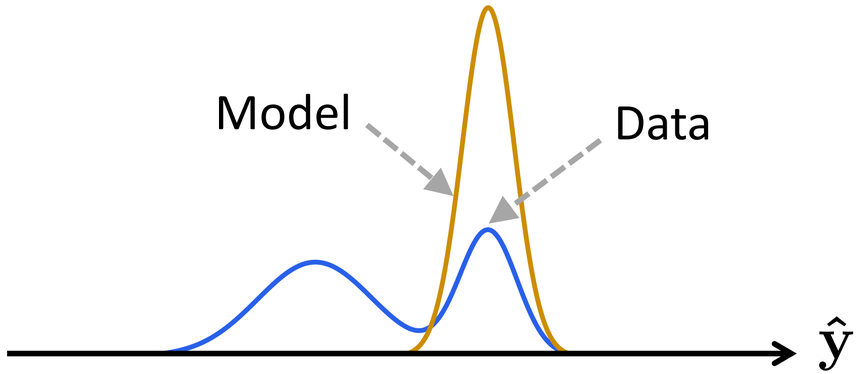
Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Wu et. al.*

GAN Issues



$$L_{GAN}(G, D) = E_y[log\ D(y) + E_{x,z}[log(1 - D(G(x, z))]$$

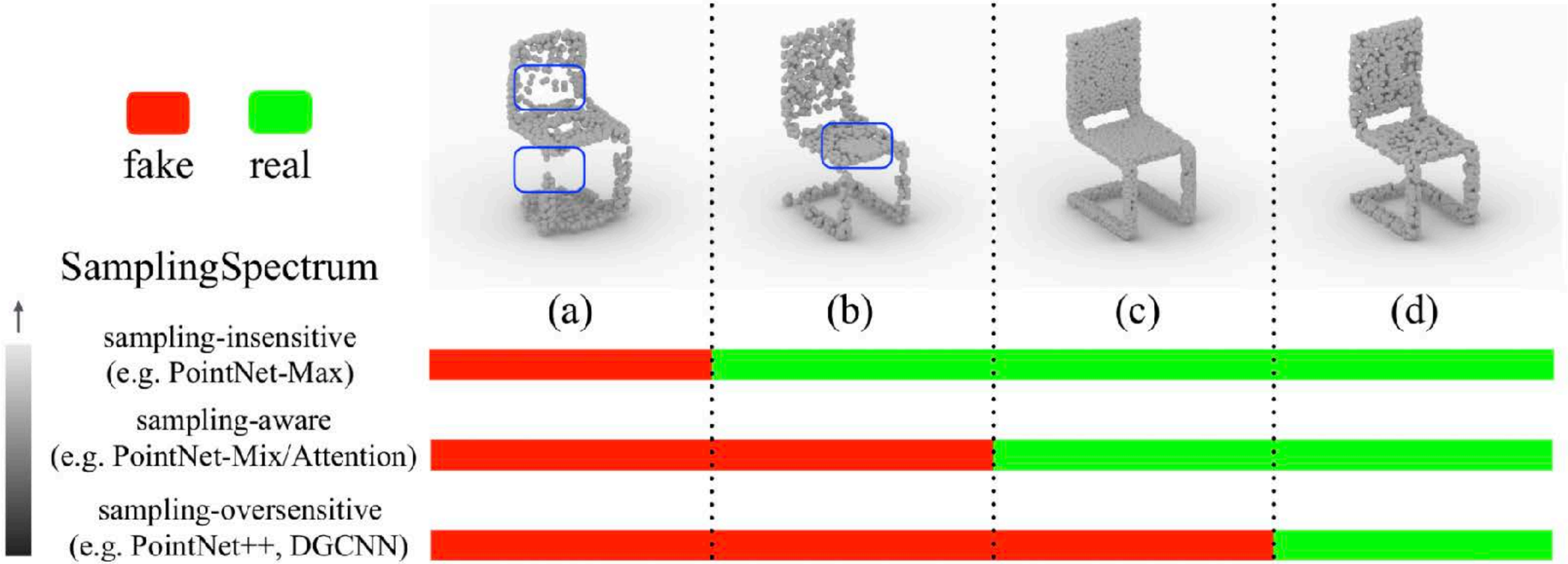
Common issue: **mode collapse**



GANs don’t maximize likelihood of data

Image courtesy: Overcoming Mode Collapse and the Curse of Dimensionality. *Ke Li*

GAN Issues

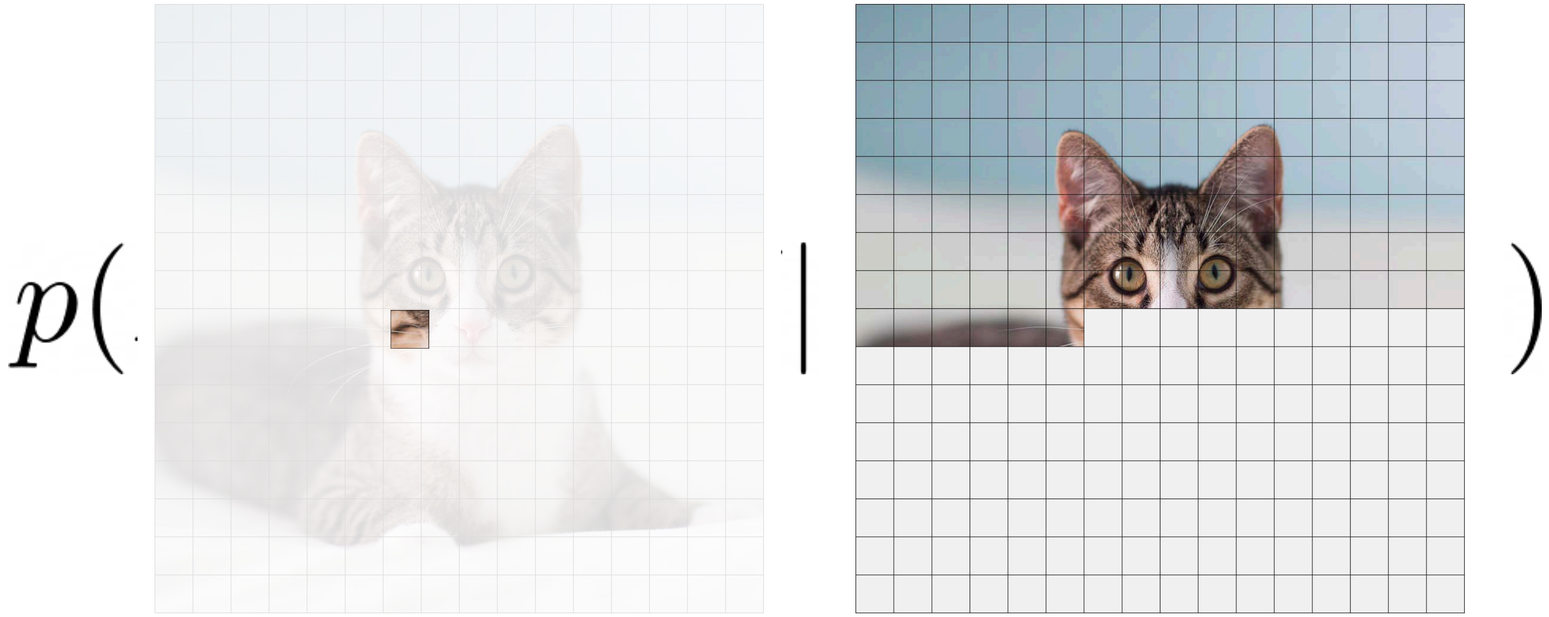


Rethinking Sampling in 3D Point Cloud Generative Adversarial Networks. Wang et al.

Outline

- 3D GAN
- 3D Autoregressive Models
- 3D Diffusion Models
- 3D Generation without 3D Training Data
- Part-based 3D Generation

Background: Autoregressive Models

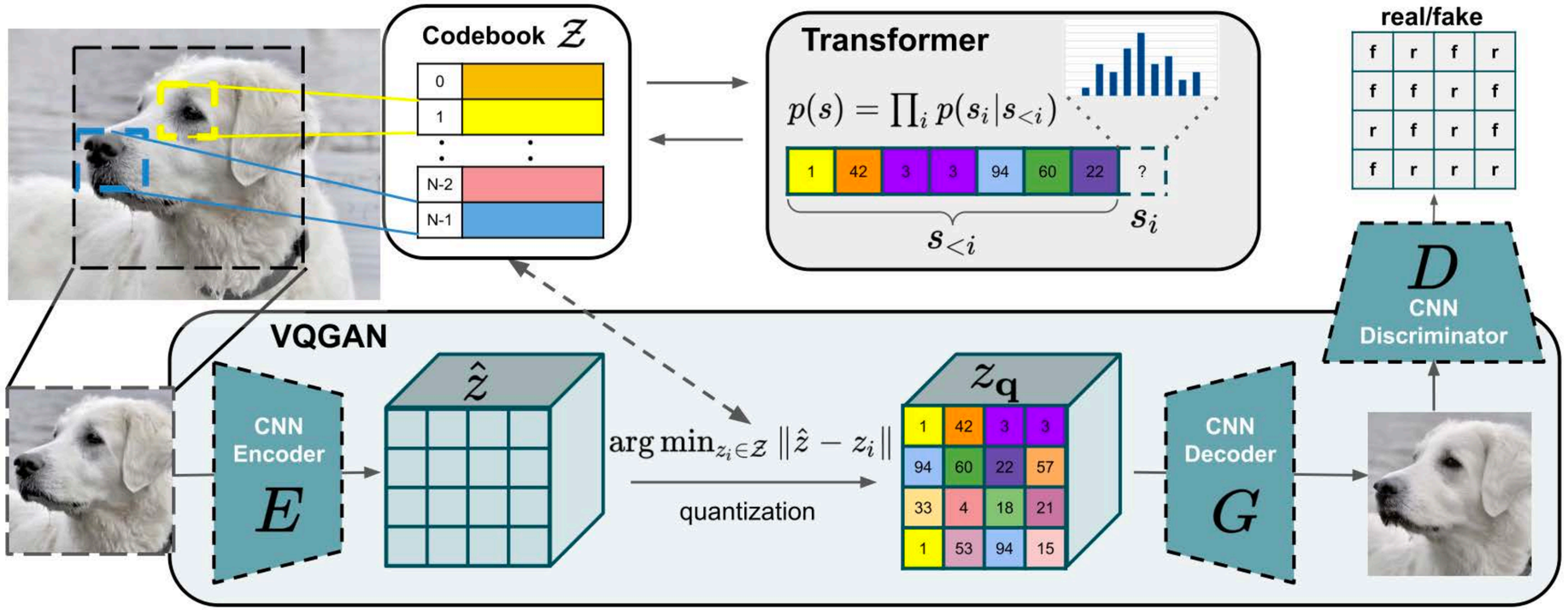


Training: $\max \mathbb{E}_{x \sim \mathcal{D}} \log p_\theta(x)$

Inference: Sample images, one pixel at a time

Image courtesy: PixelTransformer: Sample Conditioned Signal Generation. *Tulsiani et. al.*

Background: Autoregressive Models

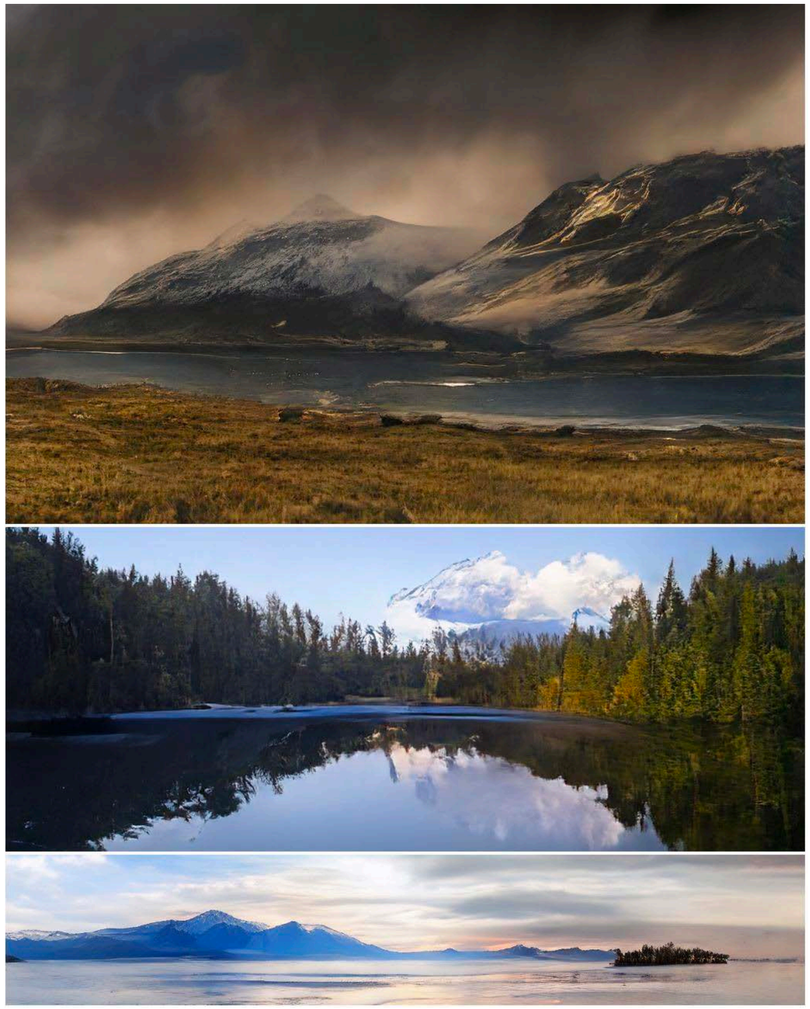


Step 1: Learn a ‘codebook’ of discrete patch representations

Step 2: Learn transformer-based Autoregressive models

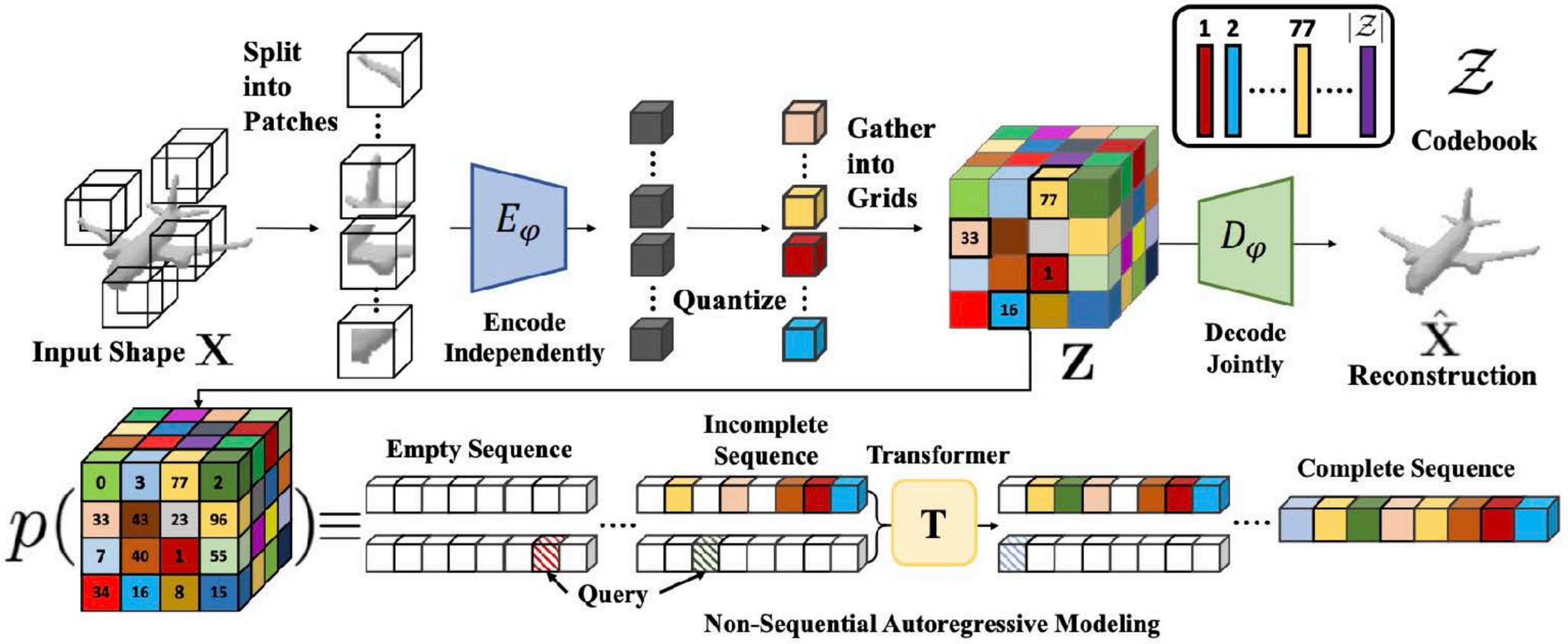
Taming Transformers for High-Resolution Image Synthesis. *Esser et. al.*

Background: Autoregressive Models



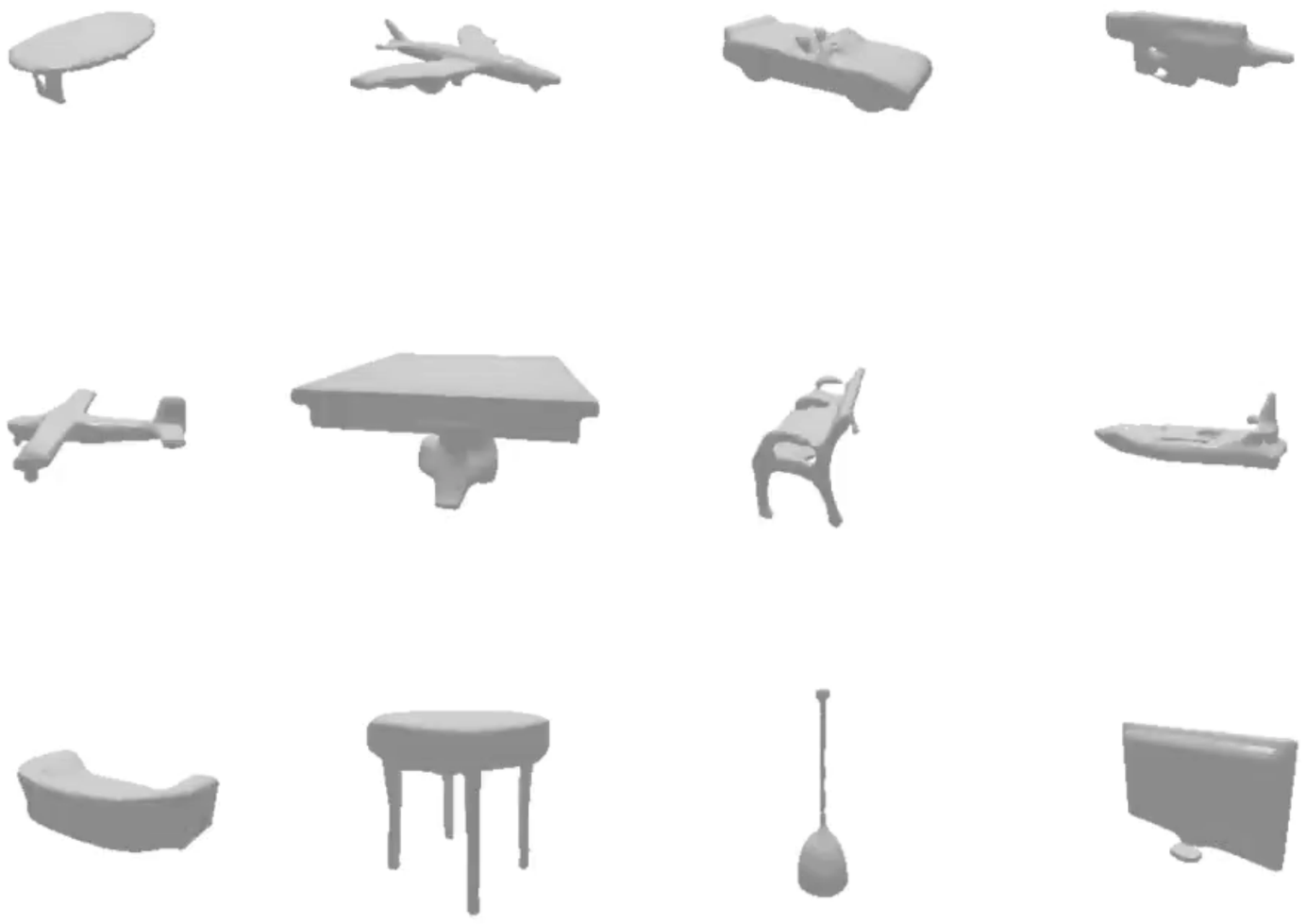
Taming Transformers for High-Resolution Image Synthesis. *Esser et. al.*

Autoregressive Generation of 3D Shapes



AutoSDF: Shape Priors for 3D Completion, Reconstruction and Generation. *Mittal*, Cheng* et. al.*

Autoregressive Generation of 3D Shapes



Unconditional generation from model trained over 50 categories

AutoSDF: Shape Priors for 3D Completion, Reconstruction and Generation. *Mittal*, Cheng* et. al.*

Autoregressive Generation of 3D Shapes

Trivially allows shape completion (missing regions in red)

AutoSDF: Shape Priors for 3D Completion, Reconstruction and Generation. *Mittal*, Cheng* et. al.*

Autoregressive Generation of 3D Shapes

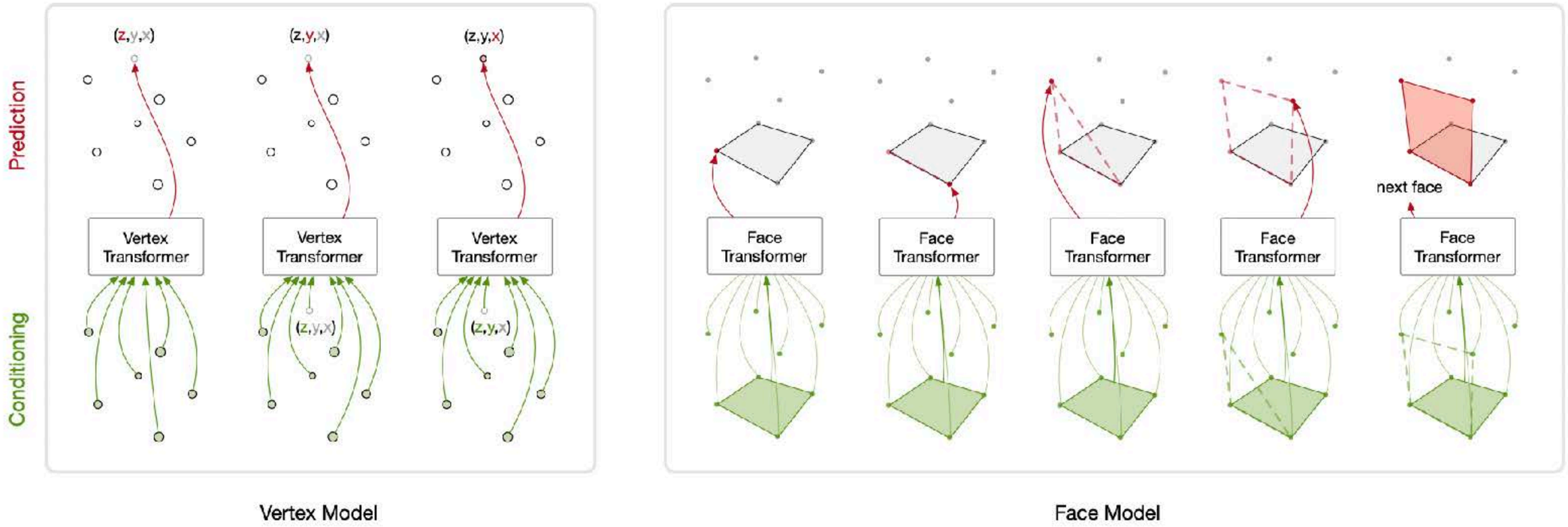
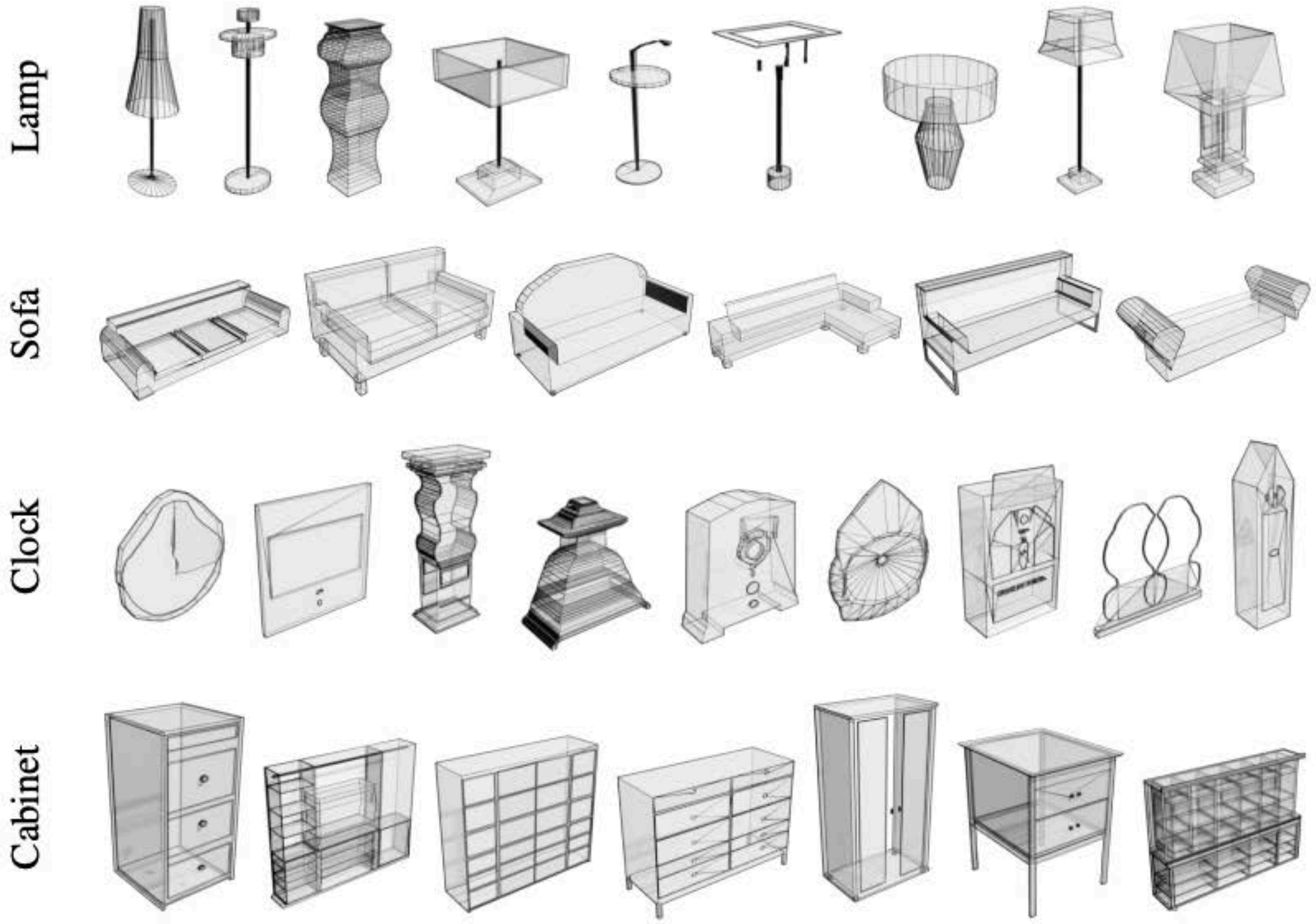


Figure 2. PolyGen first generates mesh vertices (left), and then generates mesh faces conditioned on those vertices (right). Vertices are generated sequentially from lowest to highest on the vertical axis. To generate the next vertex the current sequence of vertex coordinates is passed as context to a vertex Transformer, which outputs a predictive distribution for the next vertex coordinate. The face model takes as input a collection of vertices, and the current sequence of face indices, and outputs a distribution over vertex indices.

PolyGen: An Autoregressive Generative Model of 3D Meshes. Nash et al.

Autoregressive Generation of 3D Shapes

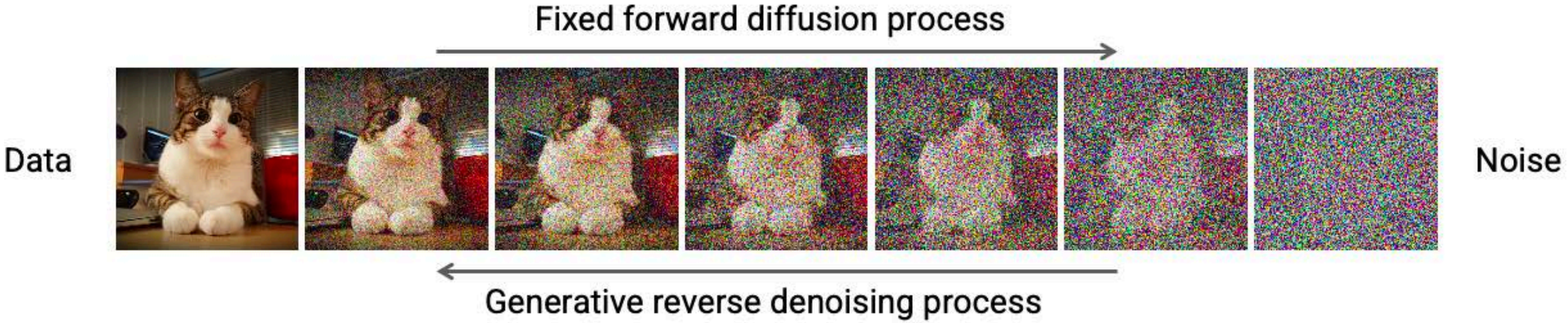


PolyGen: An Autoregressive Generative Model of 3D Meshes. Nash et al.

Outline

- 3D GAN
- 3D Autoregressive Models
- 3D Diffusion Models
- 3D Generation without 3D Training Data
- Part-based 3D Generation

Background: Denoising Diffusion Models

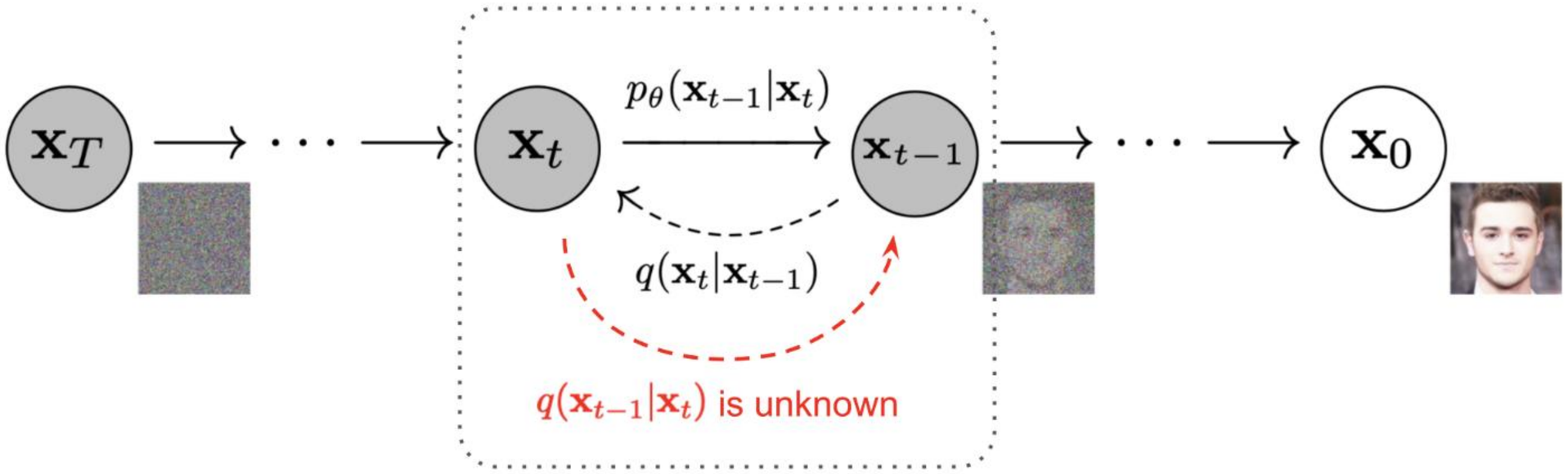


synthesis by progressive denoising

Train a neural network to learn the reverse process

Tutorial: https://cvpr2022-tutorial-diffusion-models.github.io/

Background: Denoising Diffusion Models



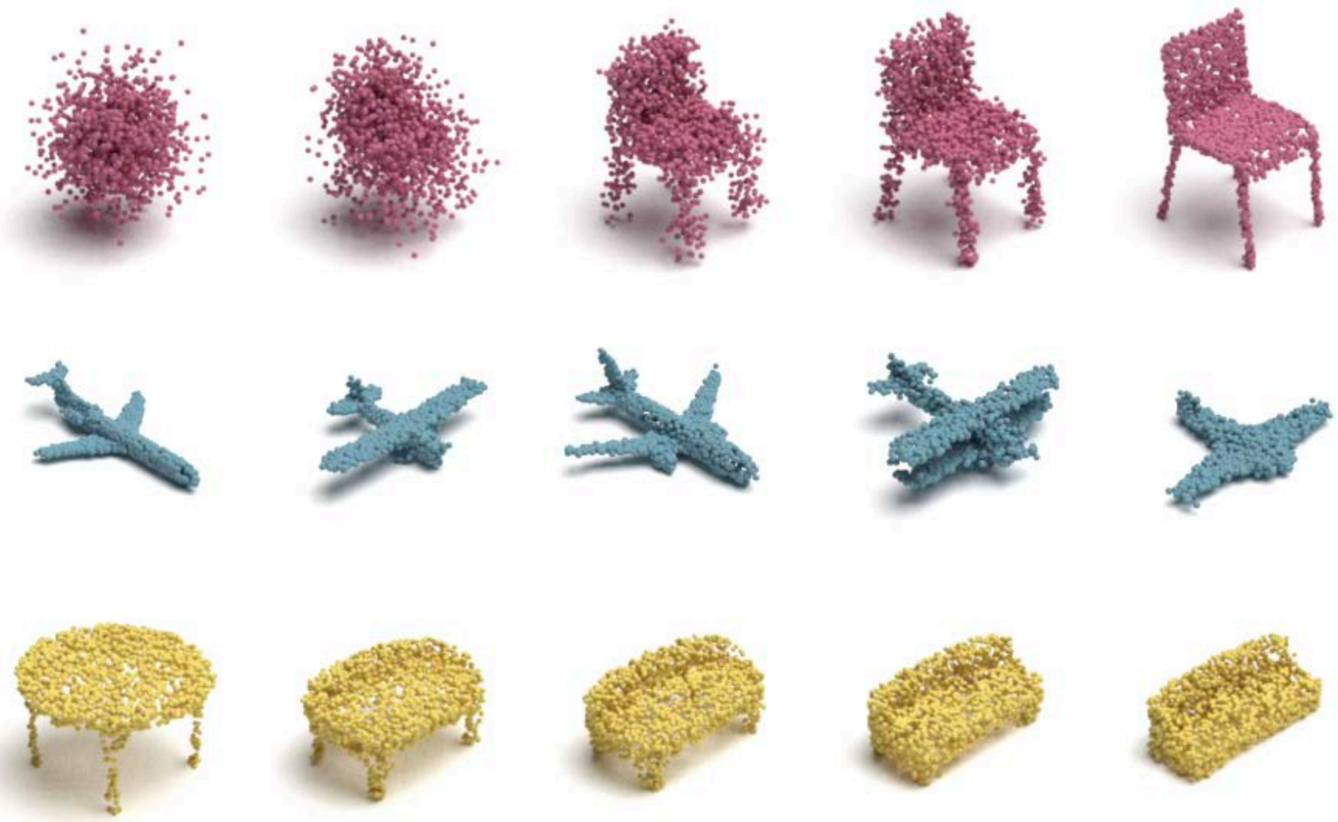
Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Point Cloud Diffusion



Algorithm 1 Training (Simplified)

repeat
  Sample $\boldsymbol{X}^{(0)} \sim q_{\text{data}}(\boldsymbol{X}^{(0)})$
  Sample $\boldsymbol{z} \sim q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})$
  Sample $t \sim \text{Uniform}(\{1, \dots, T\})$
  Sample $\boldsymbol{x}_1^{(t)}, \dots, \boldsymbol{x}_N^{(t)} \sim q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(0)})$
  $L_t \leftarrow \sum_{i=1}^{N} D_{\text{KL}}\left(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)}) \middle\| p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})\right)$
  $L_{\boldsymbol{z}} \leftarrow D_{\text{KL}}(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)}) \| p(\boldsymbol{z}))$
  Compute $\nabla_{\boldsymbol{\theta}}(L_t + \frac{1}{T} L_{\boldsymbol{z}})$. Then perform gradient descent.
until converged

Diffusion Probabilistic Models for 3D Point Cloud Generation. Luo et al.

Latent Diffusion for 3D Shape Generation

Diffusion-SDF: Conditional Generative Modeling of Signed Distance Functions

Gene Chou
Princeton University

Yuval Bahat
Princeton University

Felix Heide
Princeton University

SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation

Yen-Chi Cheng[1] Hsin-Ying Lee[2] Sergey Tulyakov[2] Alexander Schwing[1*] Liangyan Gui[1*]
[1]University of Illinois Urbana-Champaign [2]Snap Research

Diffusion-SDF: Text-to-Shape via Voxelized Diffusion

Muheng Li[1], Yueqi Duan[†,2], Jie Zhou[1], Jiwen Lu[1]
[1]Department of Automation, Tsinghua University
[2]Department of Electronic Engineering, Tsinghua University

Several (concurrent) works exploring this, with essentially similar approach…

Background: Stable Diffusion

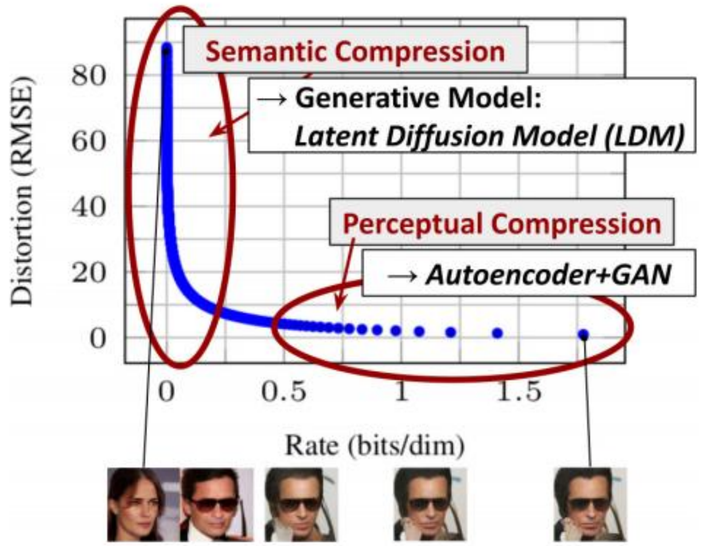
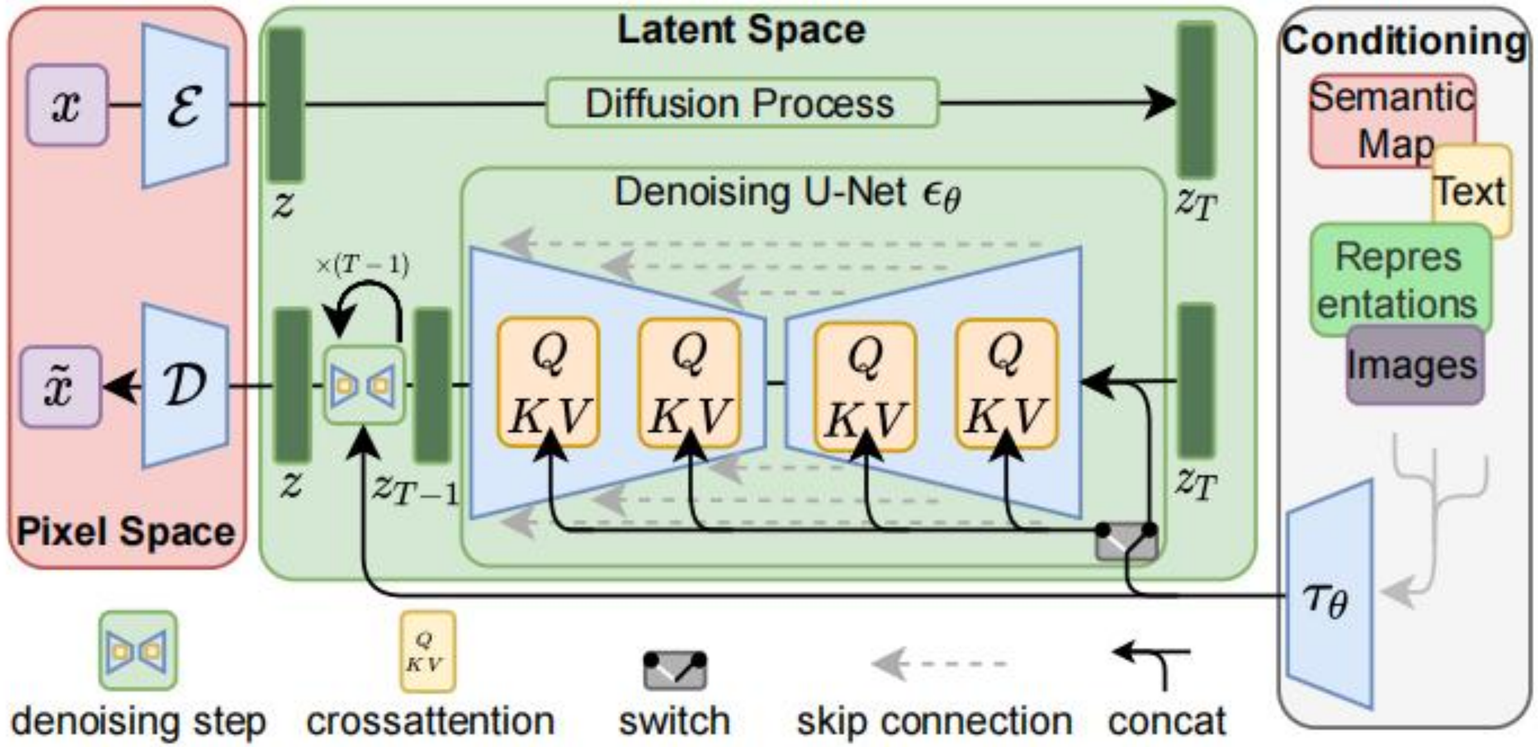


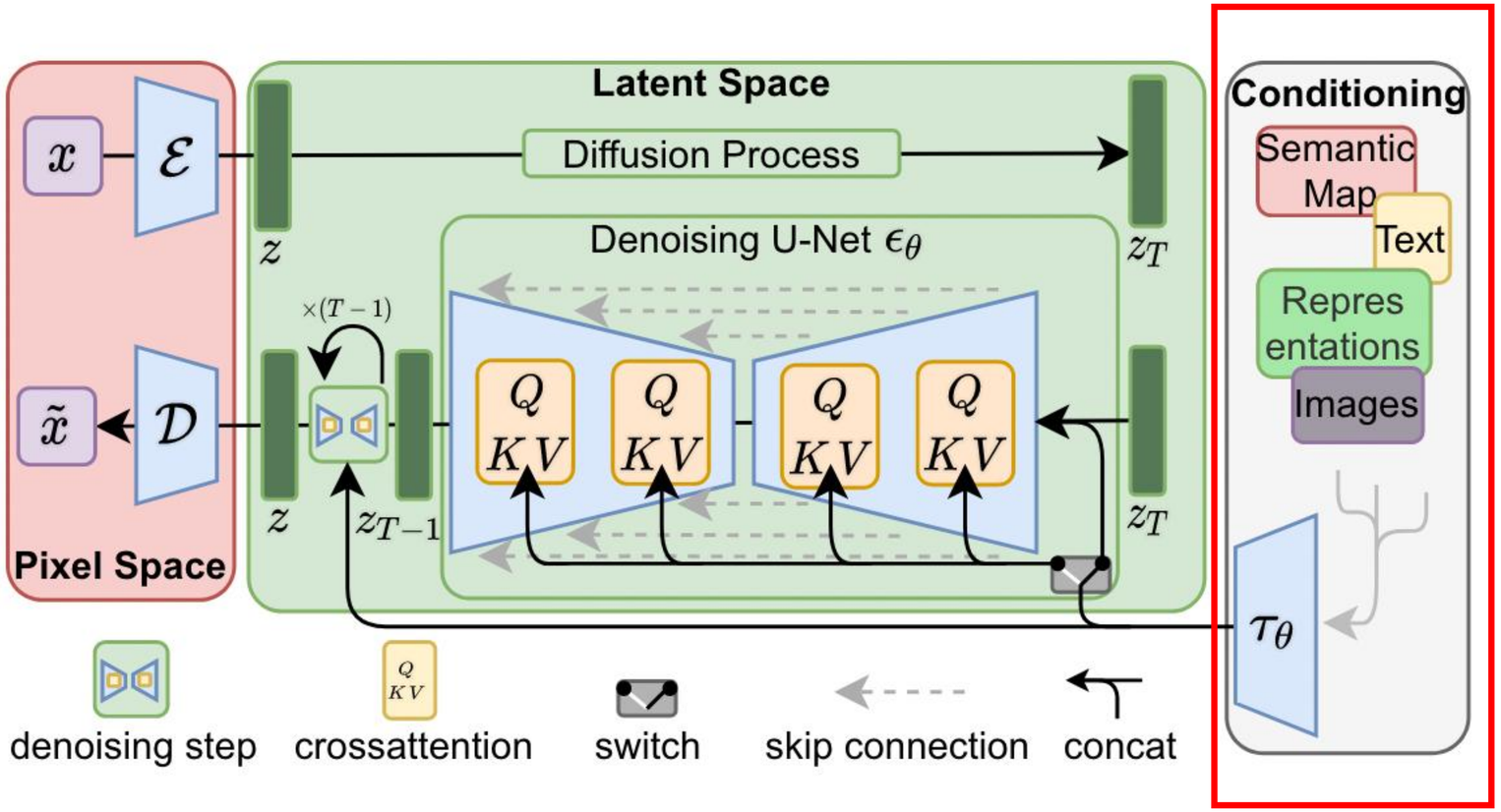
Figure 2. Illustrating perceptual and semantic compression: Most bits of a digital image correspond to imperceptible details. While DMs allow to suppress this semantically meaningless information by minimizing the responsible loss term, gradients (during training) and the neural network backbone (training and inference) still need to be evaluated on all pixels, leading to superfluous computations and unnecessarily expensive optimization and inference. We propose *latent diffusion models (LDMs)* as an effective generative model and a separate mild compression stage that only eliminates imperceptible details. Data and images from [30].

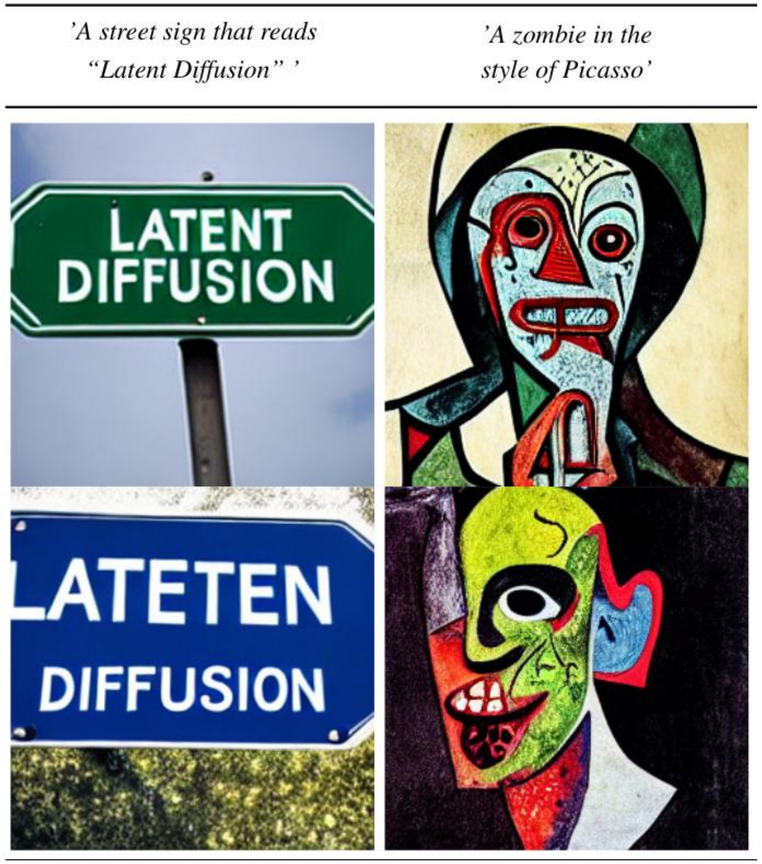


- VAE-encoder + diffusion model in latent space + VAE-decoder = Stable Generation

High-Resolution Image Synthesis with Latent Diffusion Models. Rombach et al.

Background: Conditional Diffusion



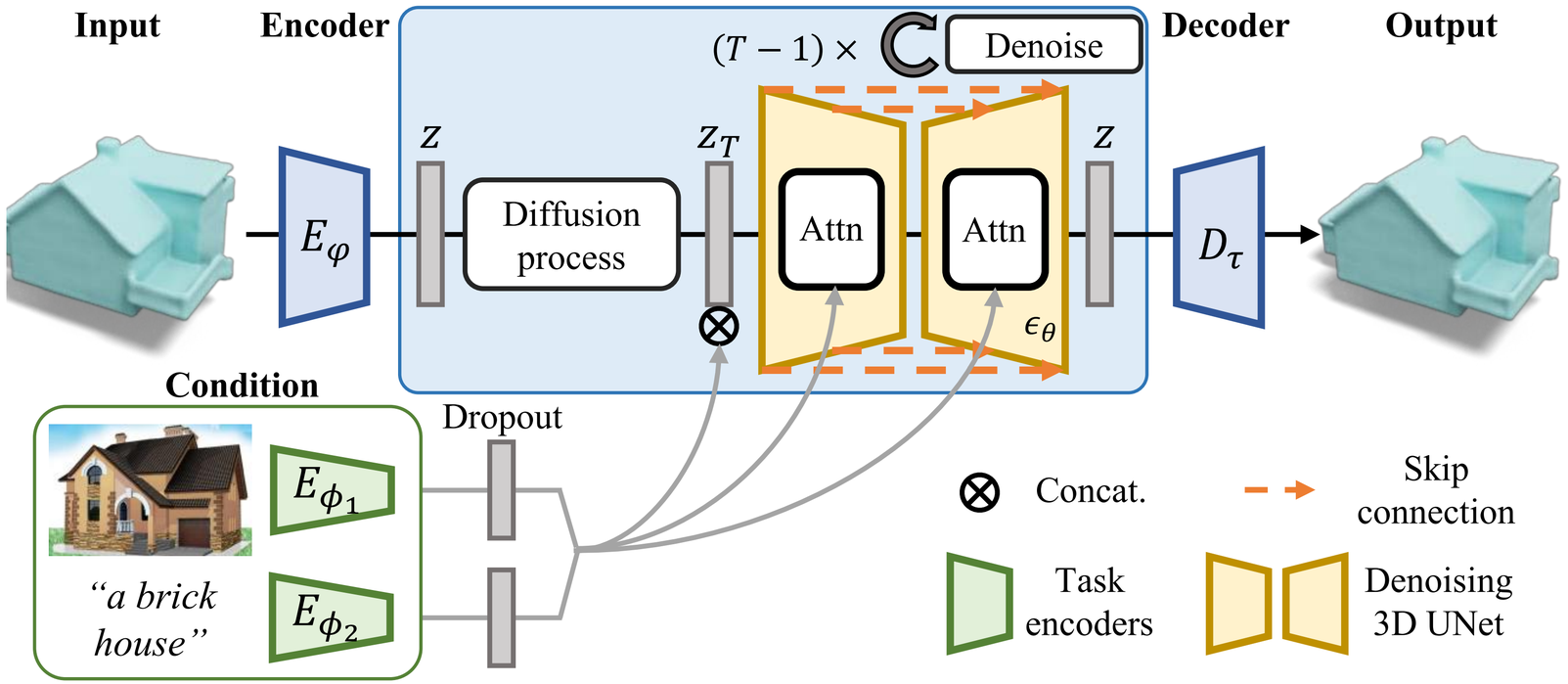


$$\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{y}\sim\tilde{p}(\boldsymbol{x}_0,\boldsymbol{y}),\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\left\|\boldsymbol{\varepsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\bar{\beta}_t\boldsymbol{\varepsilon},\boldsymbol{y},t)\right\|^2\right] \quad (22)$$

feed condition information y
To de-noise network in training

High-Resolution Image Synthesis with Latent Diffusion Models. Rombach et al.

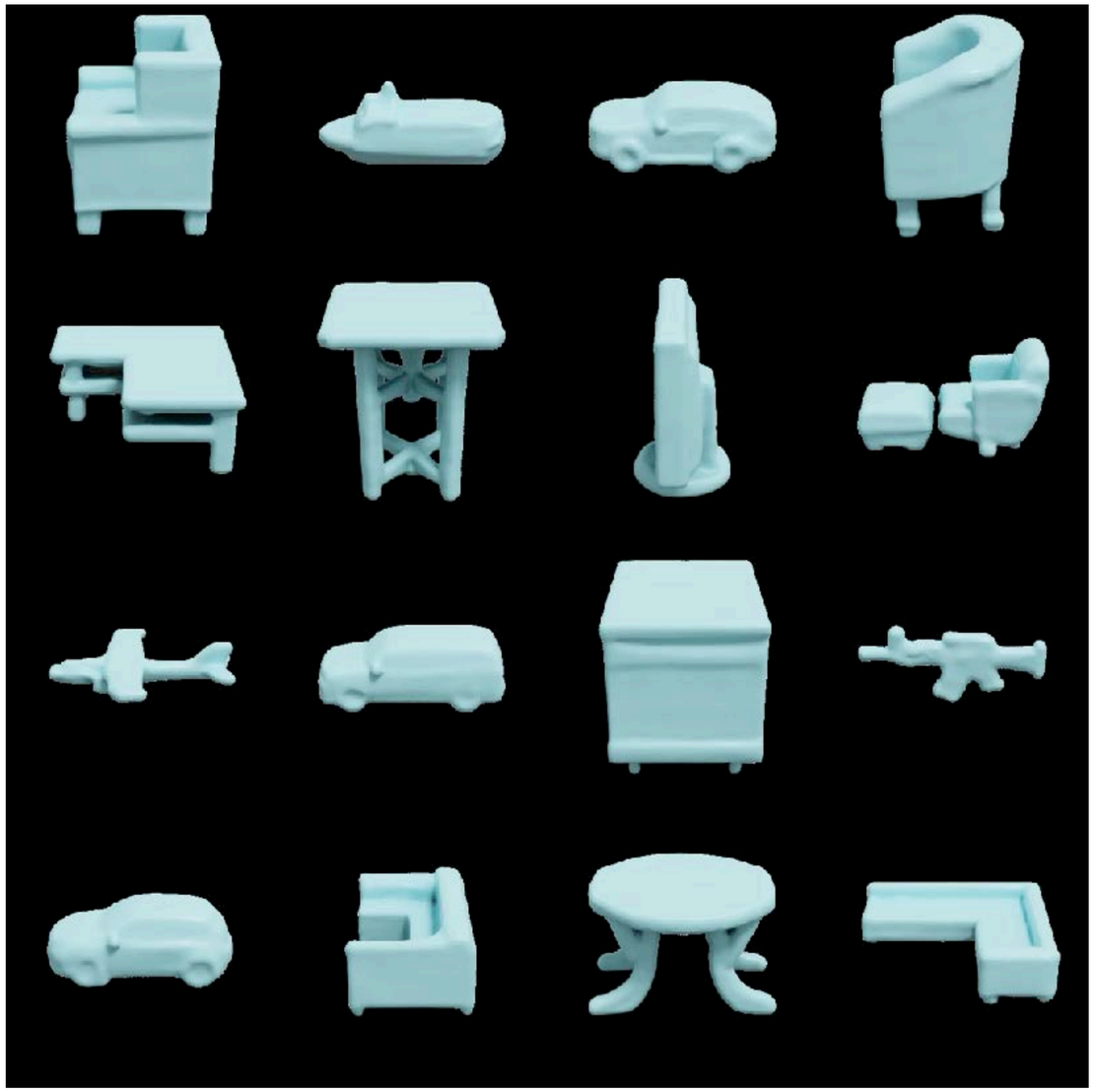
Latent Diffusion for 3D Shape Generation

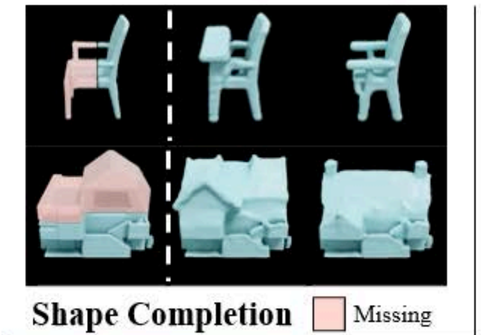


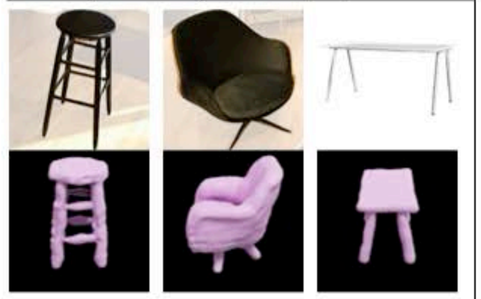
3D Diffusion model in latent space, with conditioning via attention layers

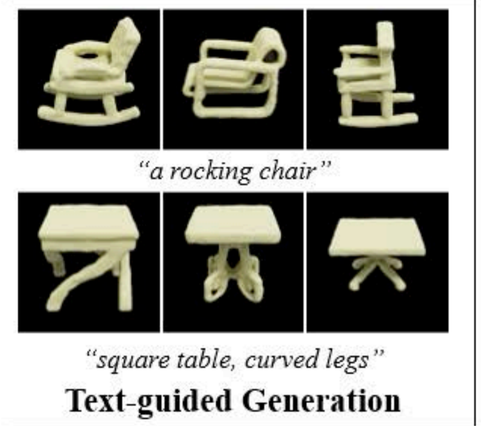
SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation.*Cheng et. al.*

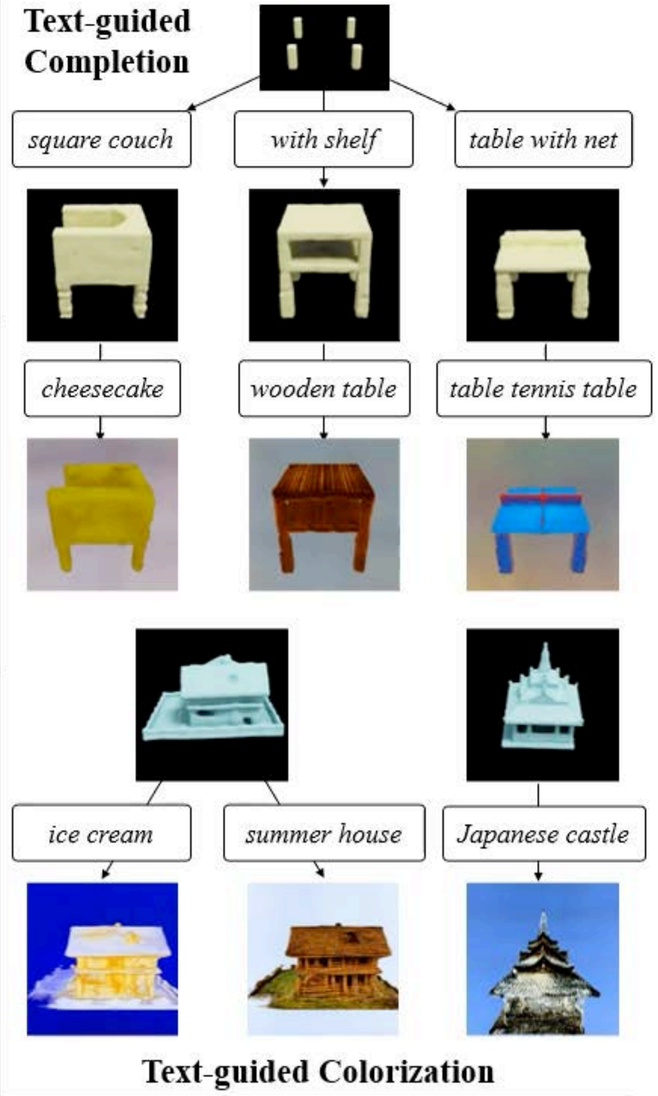
Latent Diffusion for 3D Shape Generation











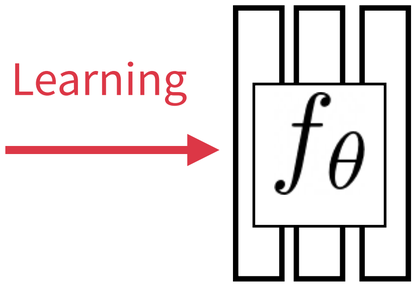
SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation.*Cheng et. al.*

Outline

- 3D GAN
- 3D Autoregressive Models
- 3D Diffusion Models
- 3D Generation without 3D Training Data
- Part-based 3D Generation

Generating 3D Shapes without 3D training data

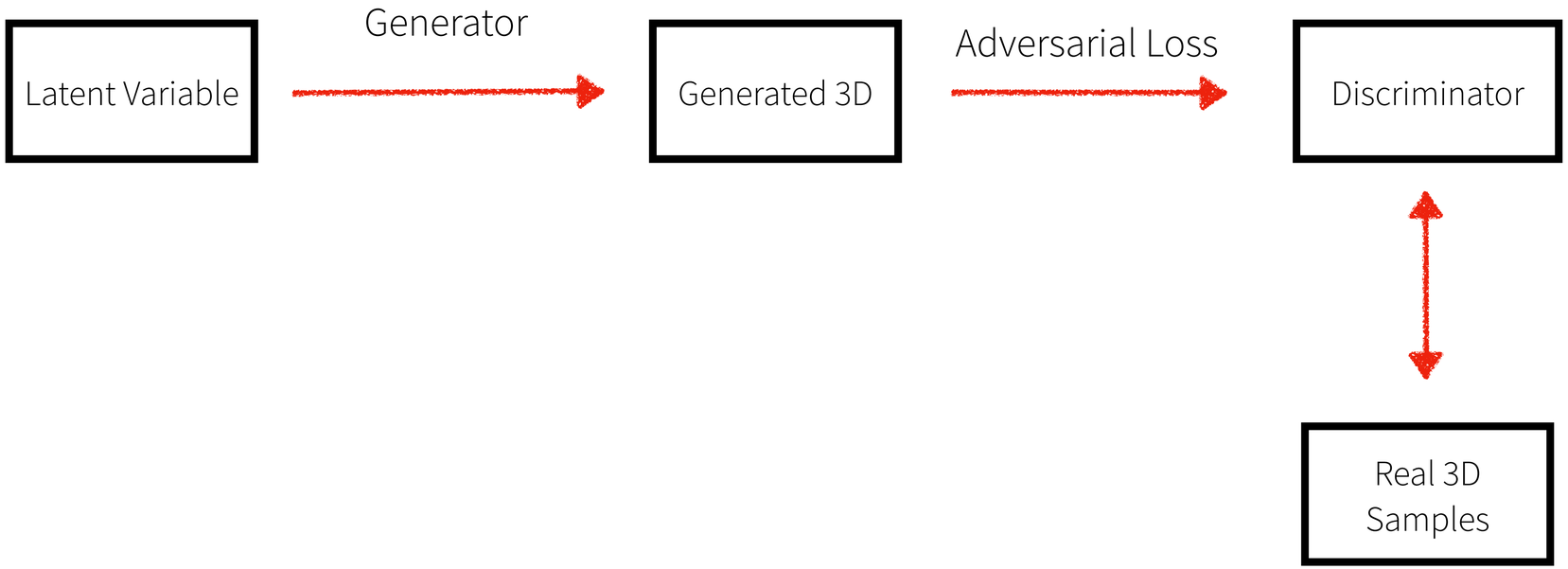
Learning



Inference

Can we learn a generative 3D model from 2D images?

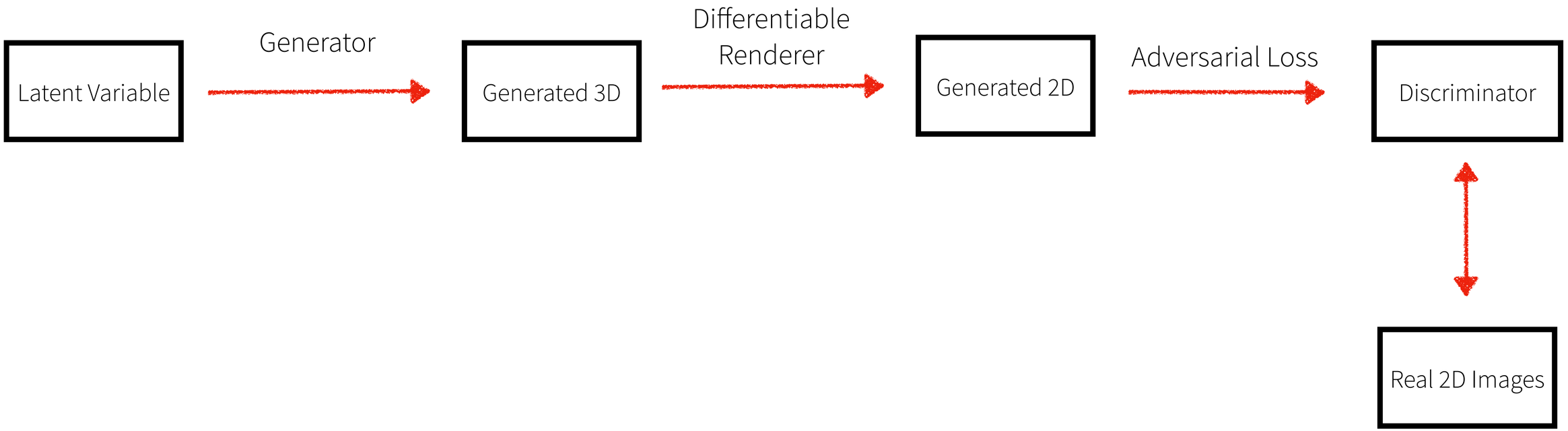
Generating 3D Shapes without 3D training data



Key Idea: Generate 3D representations such that —

they are indistinguishable from real samples

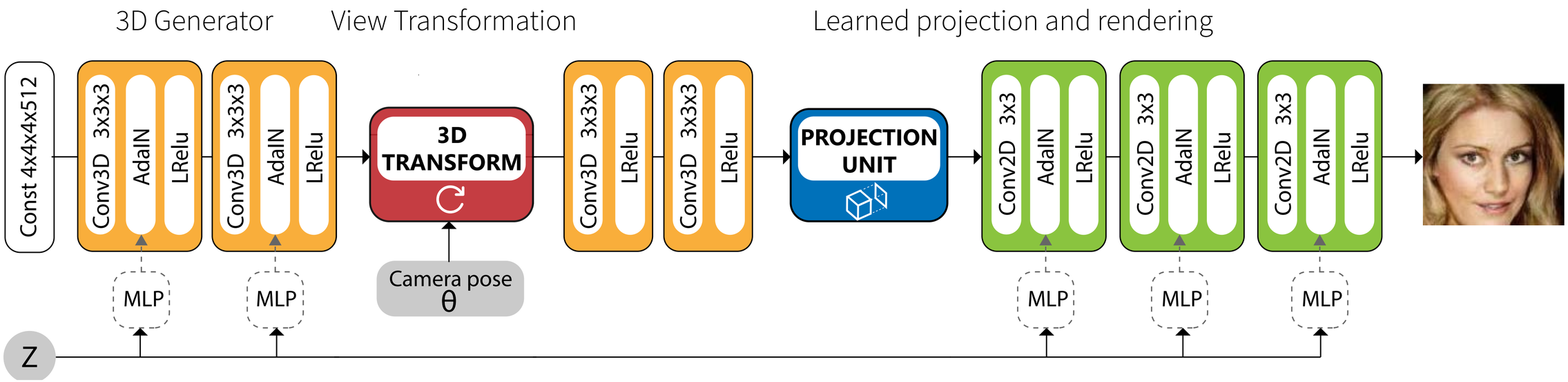
Generating 3D Shapes without 3D training data



Key Idea: Generate 3D representations such that —

their renderings are indistinguishable from real samples

Generating 3D using 2D Adversarial Nets



Key Idea: Generate 3D representations such that —

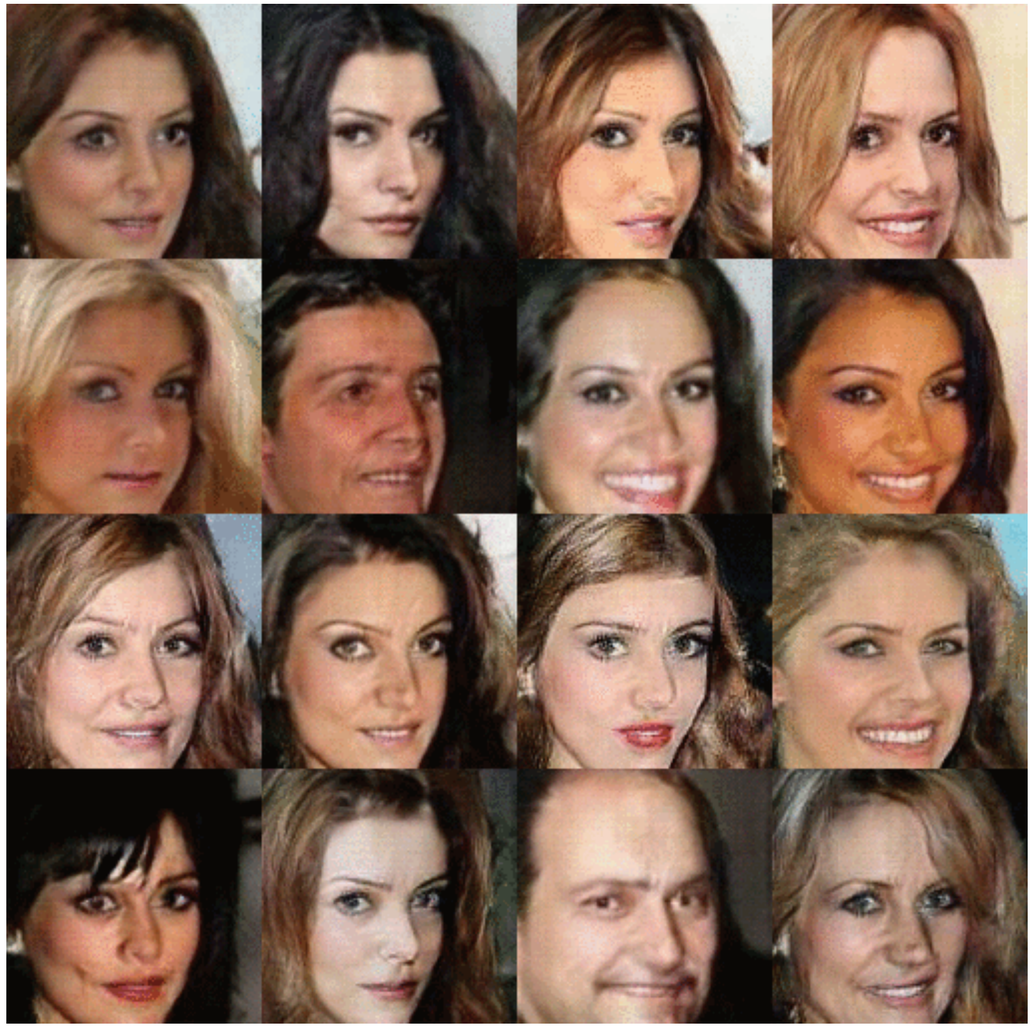
their renderings are indistinguishable from real samples

HoloGAN: Unsupervised learning of 3D representations from natural images. *Nguyen-Phuoc et. al.*

Generating 3D using 2D Adversarial Nets

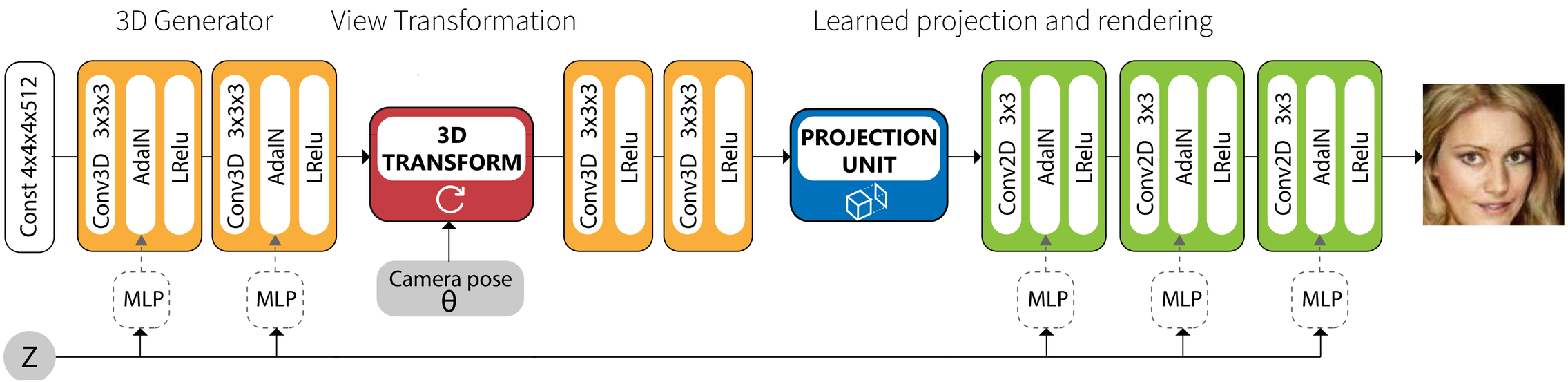
HoloGAN: Unsupervised learning of 3D representations from natural images. *Nguyen-Phuoc et. al.*

Generating 3D using 2D Adversarial Nets



HoloGAN: Unsupervised learning of 3D representations from natural images. *Nguyen-Phuoc et. al.*

Generating 3D using 2D Adversarial Nets

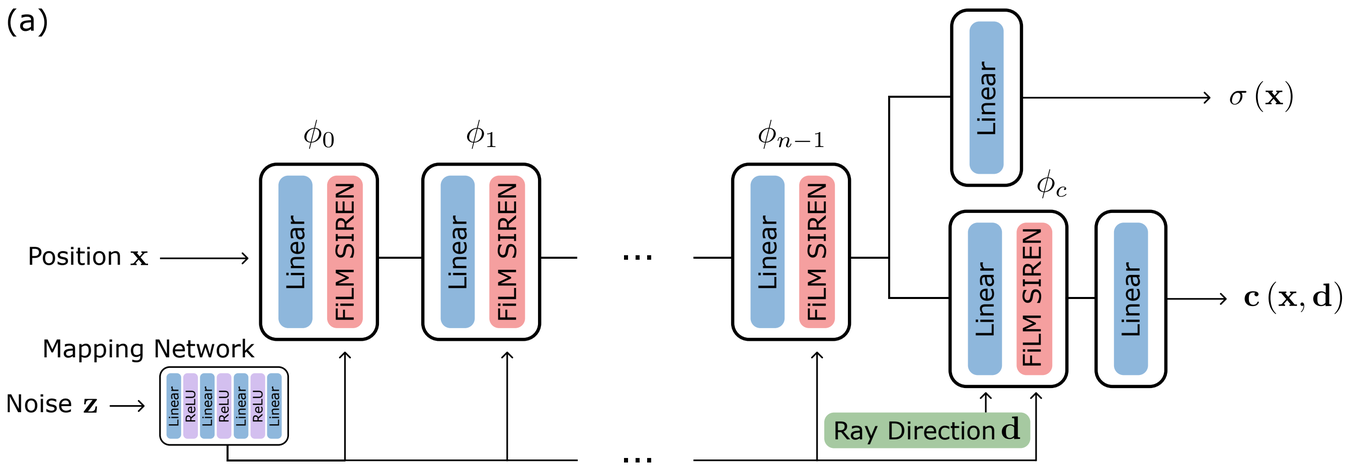


Some inconsistency across views

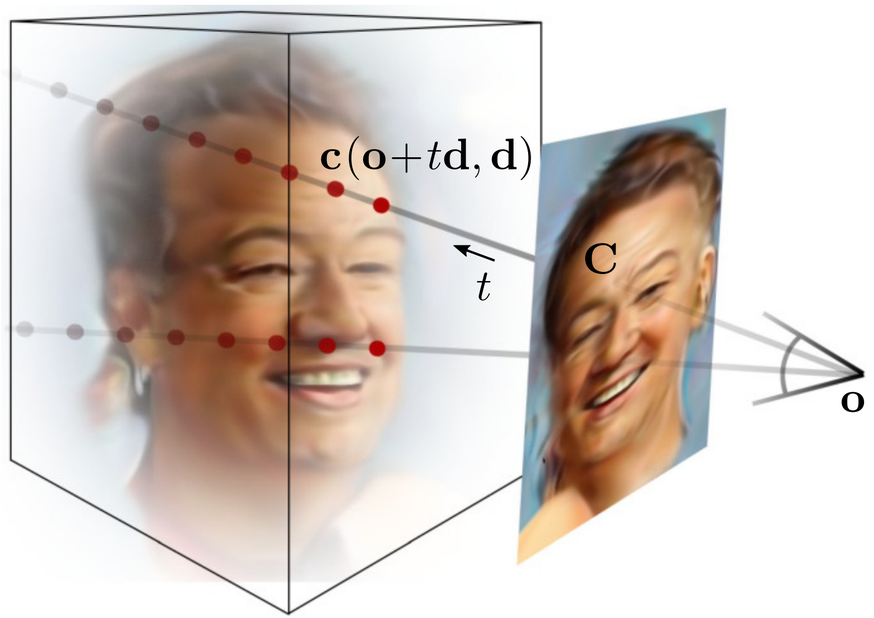
Some 3D inducting biases, but not enough (e.g. learned instead of volumetric rendering)

HoloGAN: Unsupervised learning of 3D representations from natural images. *Nguyen-Phuoc et. al.*

Generating 3D using 2D Adversarial Nets



Generator: A conditional NeRF model

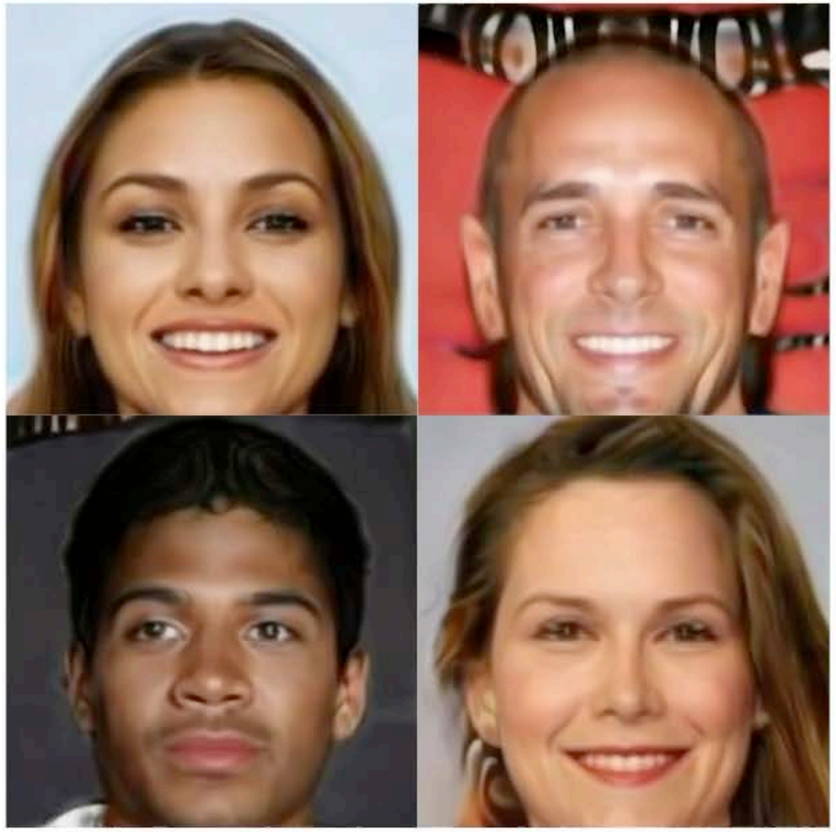


Renderer: Volume rendering

pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. *Chan et. al.*

Generating 3D using 2D Adversarial Nets

CelebA



Cats

CARLA

pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. *Chan et. al.*

Generating 3D using 2D Adversarial Nets

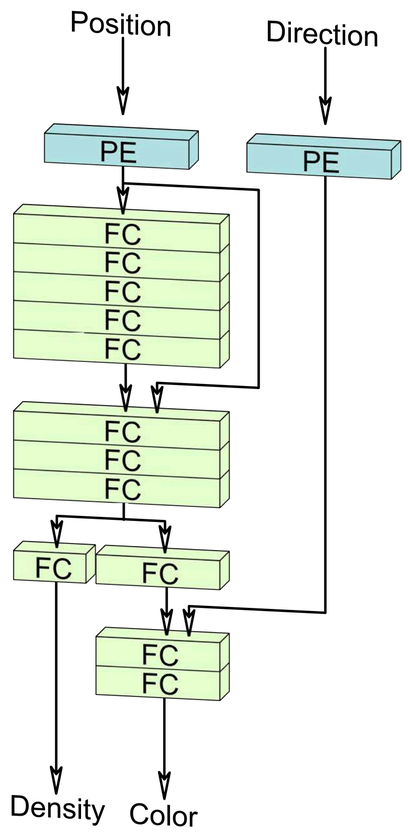
Also possible to extract some geometry

Figure 12: In a failure case reminiscent of the hollow-face illusion, our model sometimes generates objects with inverted sections.

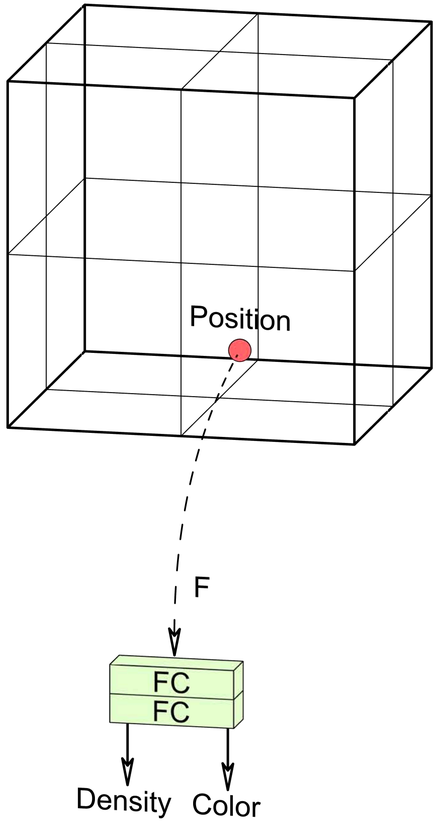
Although not always great

pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. *Chan et. al.*

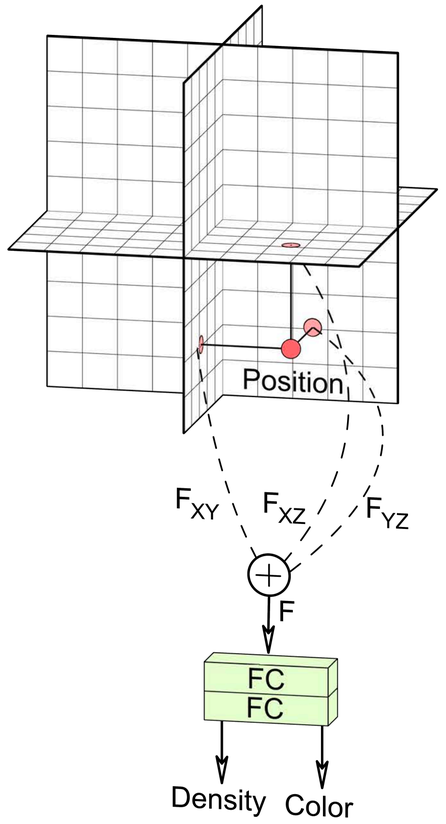
Generating 3D using 2D Adversarial Nets



(a) NeRF (Implicit)



(b) Voxels (Explicit or Hybrid)



(c) Ours (Hybrid)

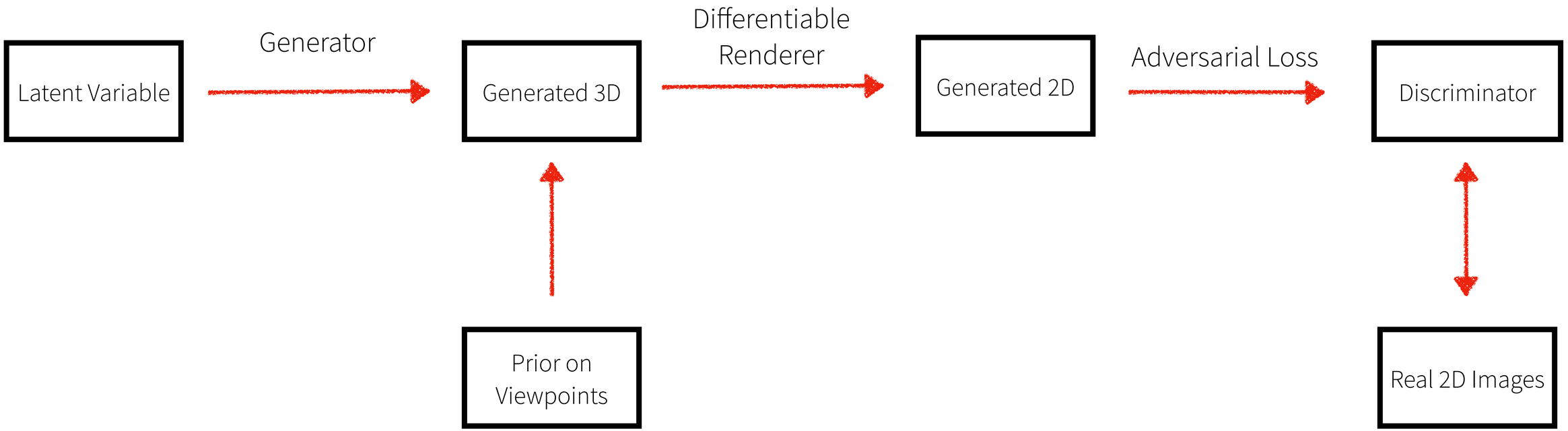
Replace NeRFs via hybrid representations

EG3D: Efficient Geometry-aware 3D Generative Adversarial Networks. *Chan et. al.*

Generating 3D using 2D Adversarial Nets

EG3D: Efficient Geometry-aware 3D Generative Adversarial Networks. *Chan et. al.*

Generating 3D Shapes without 3D training data



Crucial to leverage the 3D structure of the problem

Assumed known view distribution e.g. faces seen frontally

Typical success stories on ‘nice’ categories (lots of clean images, known views…)

Optimizing Text-conditioned 3D

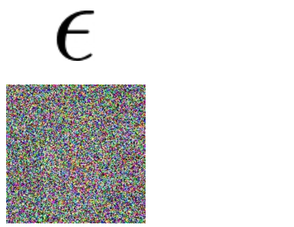
Let’s see more combinations:
https://dreamfusion3d.github.io/

“A squirrel wearing a kimono and riding a motorbike”

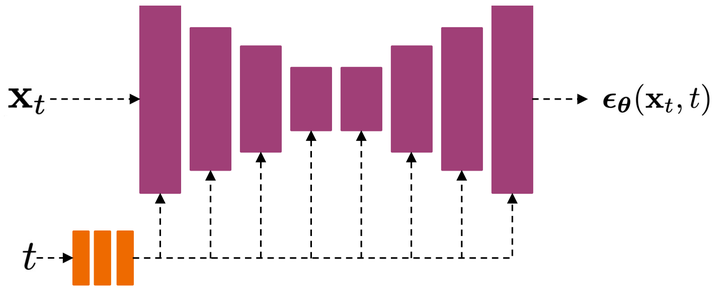
DreamFusion: Text-to-3D using 2D Diffusion. *Poole et. al.*

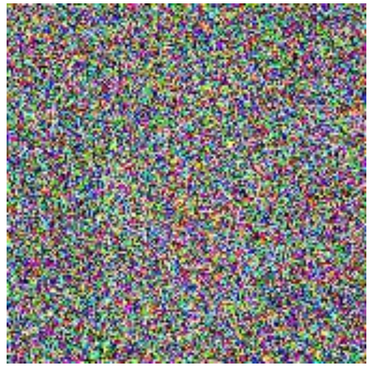
Background: Likelihood in Diffusion Models

$\mathbf{x}_0$



$\mathbf{x}_t$



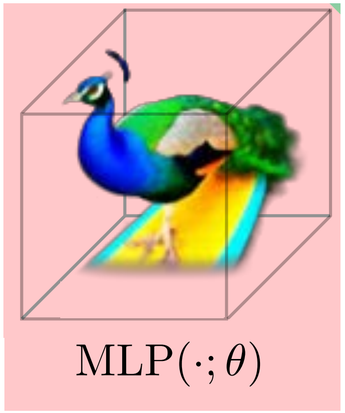


$\epsilon_\theta(\mathbf{x}_t, t)$

Training Objective: Noise prediction error $\|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon\|^2$

Prediction error also be correlated to **p(x)** (images with high likelihood yield lower error)

Optimizing Text-conditioned 3D



$\pi \sim \Pi$

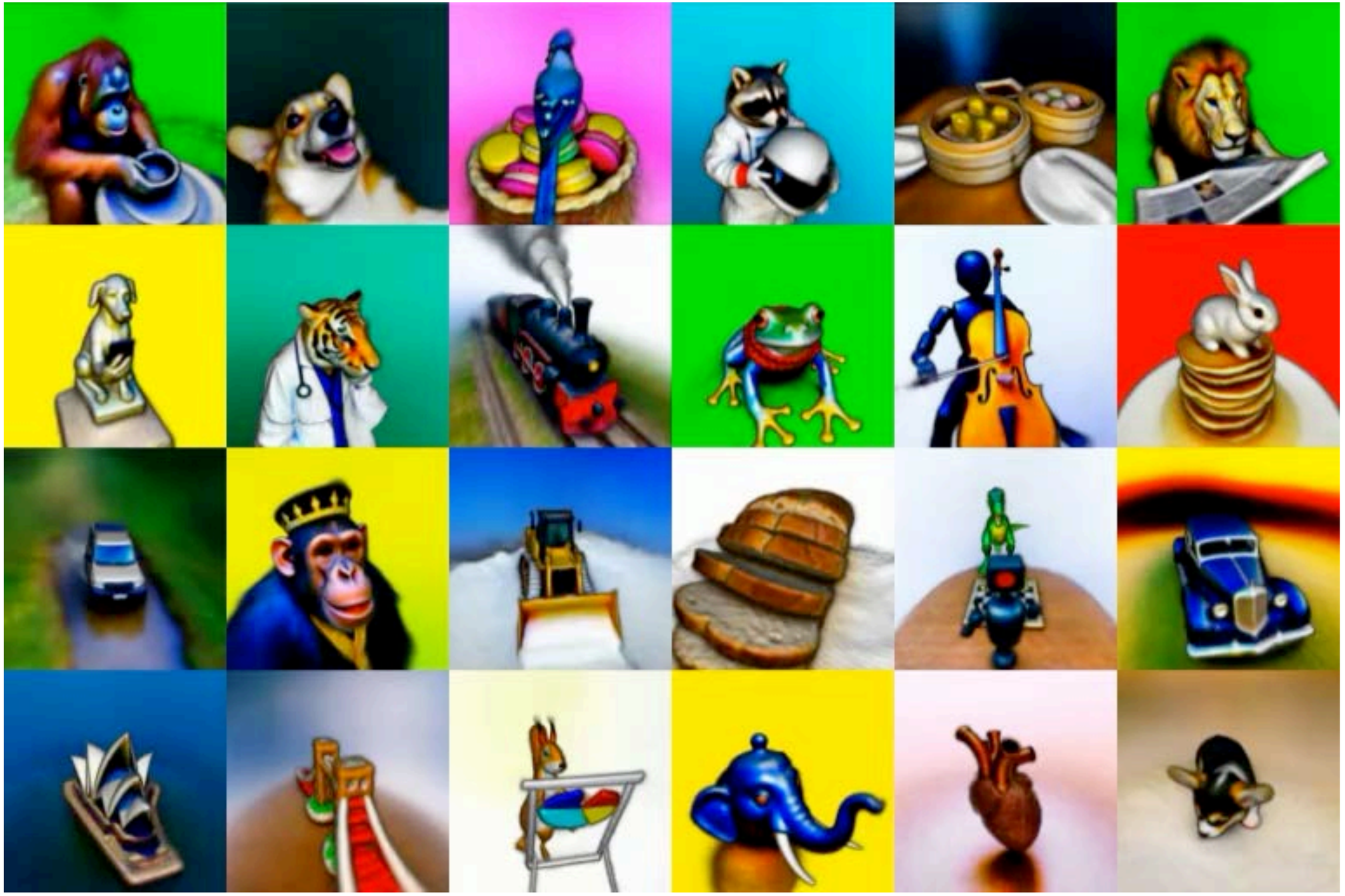
rendering

Stable Diffusion

DALL·E 2

"A high quality image of a peacock"

DreamFusion: Text-to-3D using 2D Diffusion. *Poole et. al.*



Optimizing Text-conditioned 3D

Allows 3D generation for complex structures

Not a ‘generative’ model in a probabilistic sense — no distribution over 3D is inferred

(relatively) compute intensive — per instance optimization

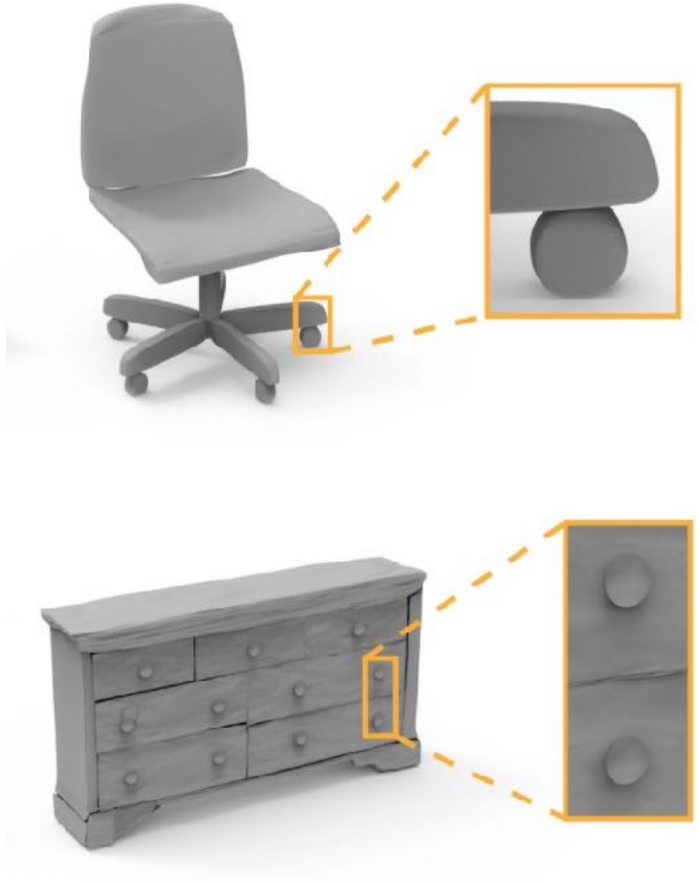
DreamFusion: Text-to-3D using 2D Diffusion. *Poole et. al.*

Outline

- 3D GAN
- 3D Autoregressive Models
- 3D Diffusion Models
- 3D Generation without 3D Training Data
- Part-based 3D Generation

Part-based 3D Shape Generation

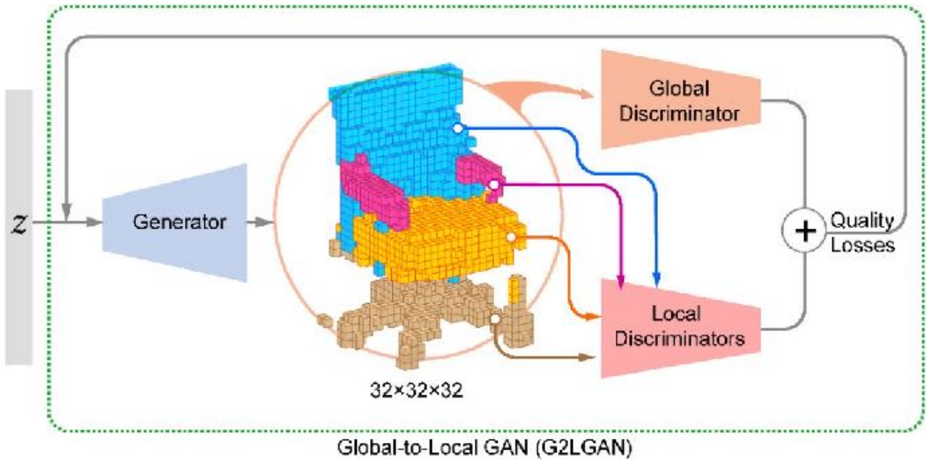
- Detailed Part Geometry
- Clear Shape Structures
- Sharp Part Boundaries
- With Semantic Part Labels

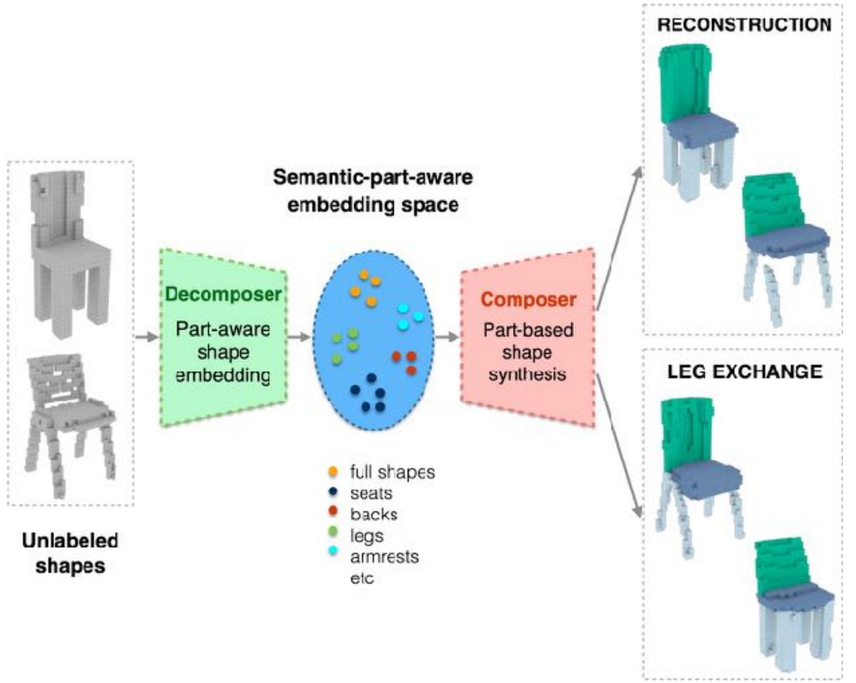


Yang and Mo et al., "**DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation**", *ACM ToG 2021*

Semantic-level Synthesis and Assembly

Per-part Geometry Generation and Part Assembly



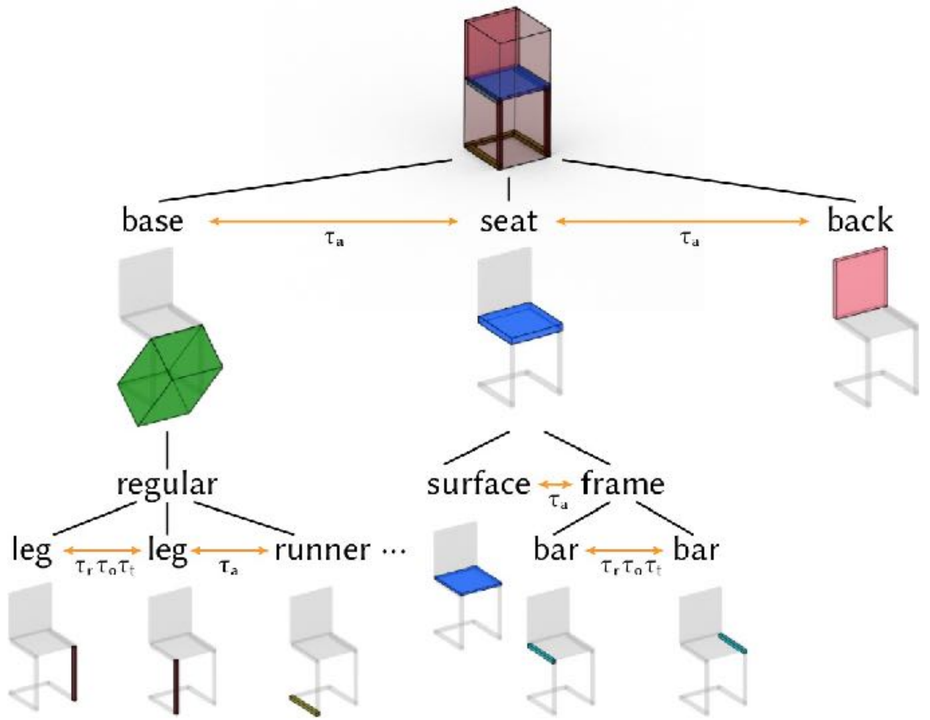


Wang and Schor et al., "**Global-to-Local Generative Model for 3D Shapes**", Siggraph Asia *2018*

Dubrovina et al., "**Composite Shape Modeling via Latent Space Factorization**", ICCV 2019

Hierarchical Generation

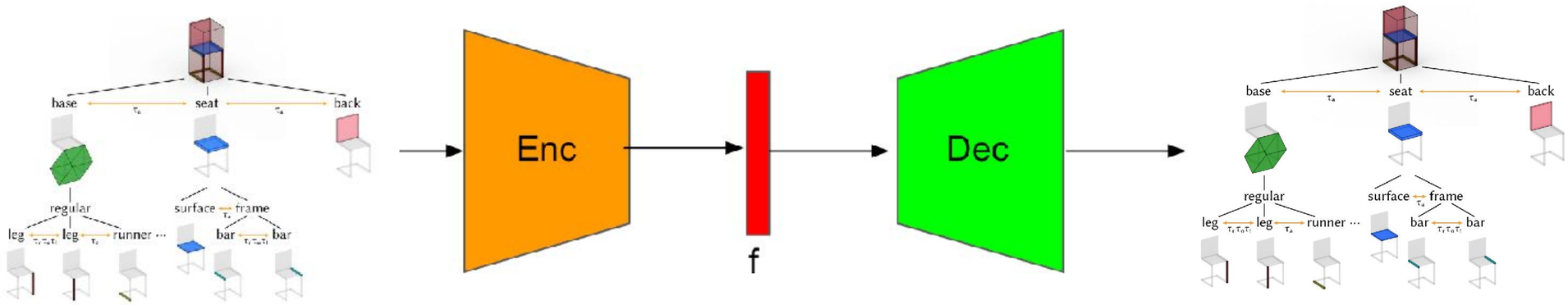
For fine-grained parts, we need coarse-to-fine generation



Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

Hierarchical Generation

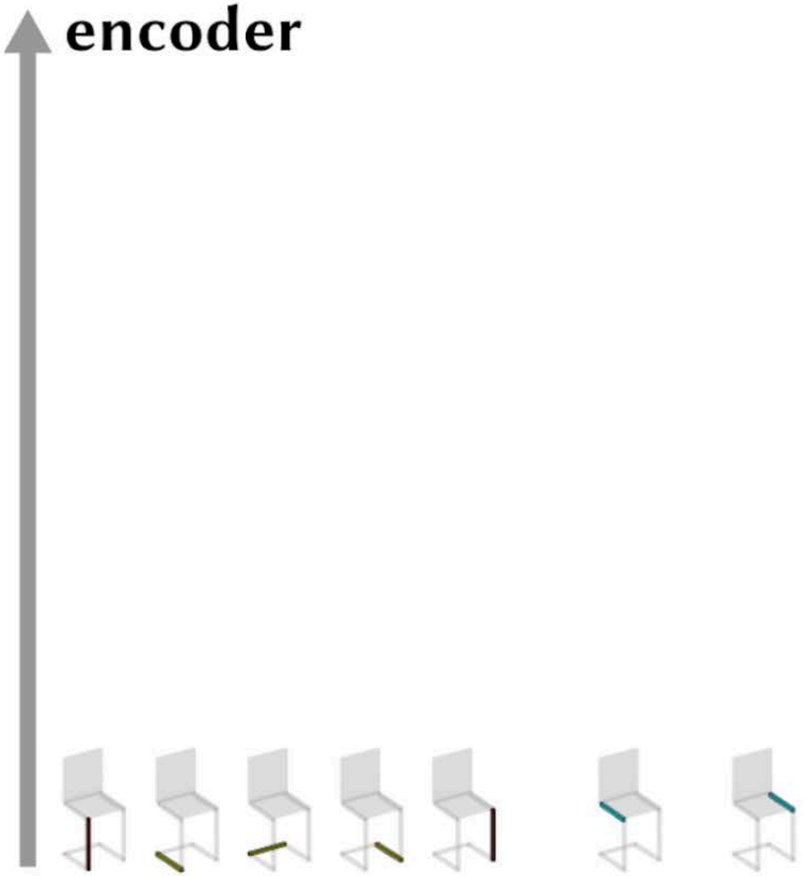
We use Variational Autoencoder (VAE)



$$f \sim N(0, I)$$

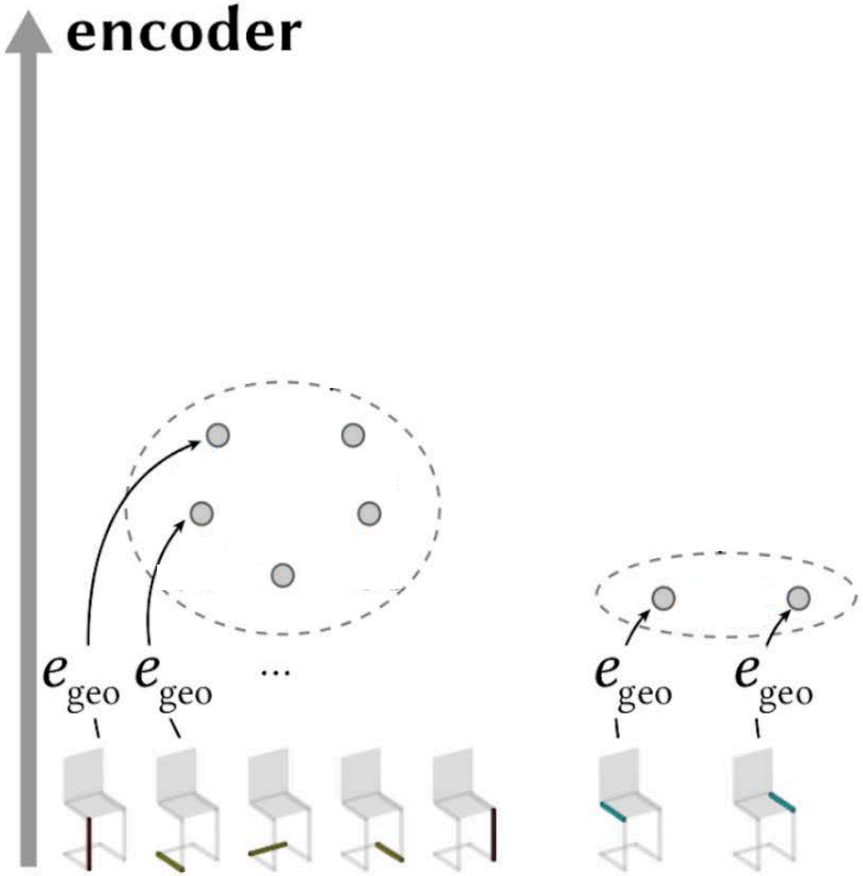
Mo et al., “**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**”, *Siggraph Asia 2019*

Hierarchical Generation



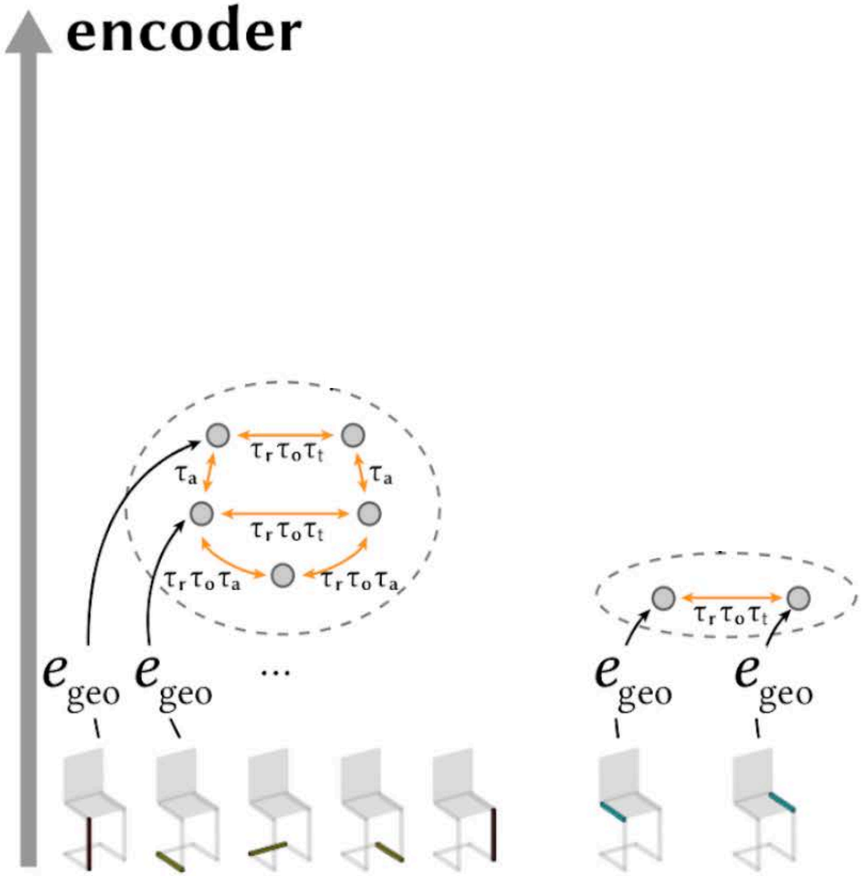
Mo et al., “**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**”, *Siggraph Asia 2019*

Hierarchical Generation



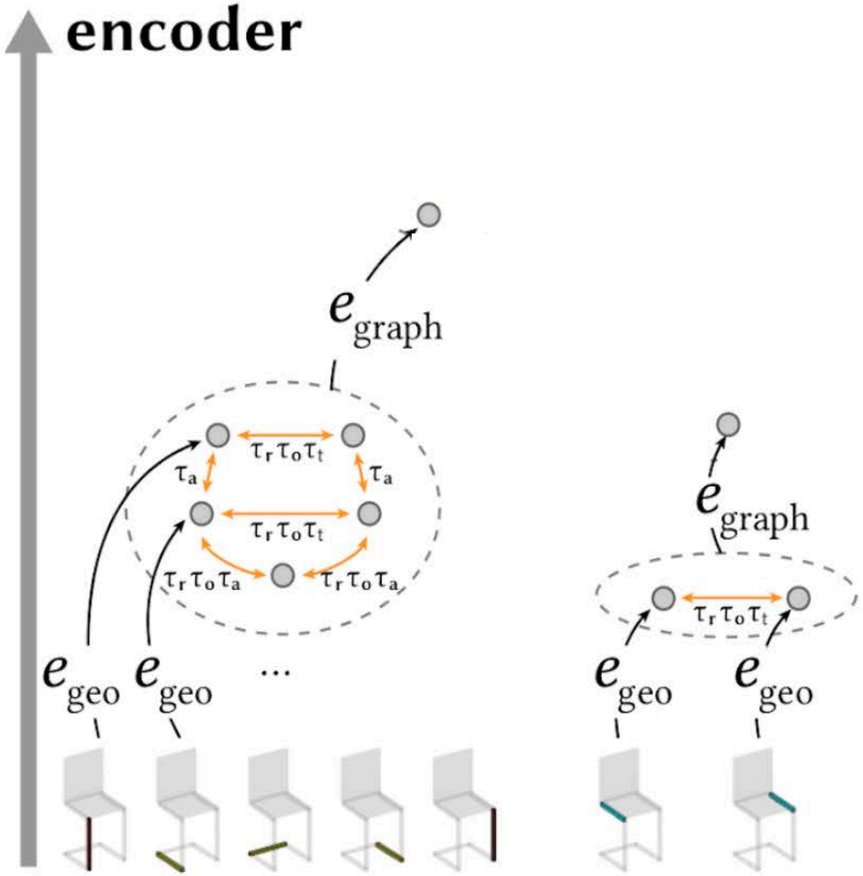
Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

Hierarchical Generation



Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

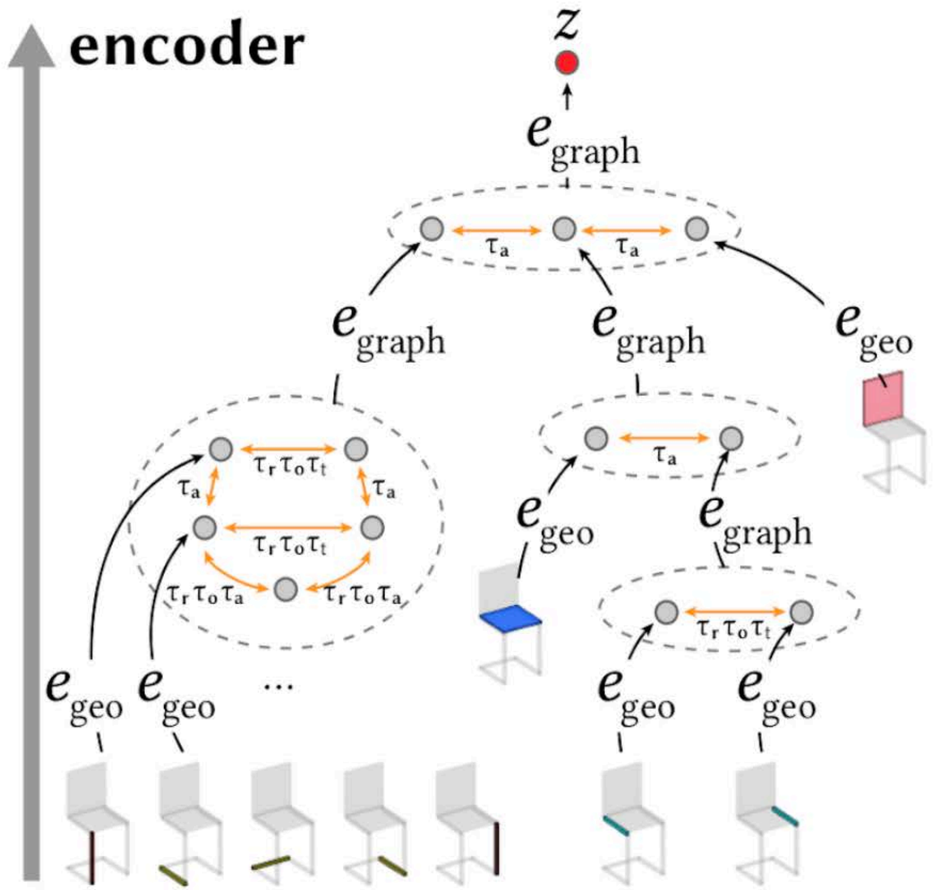
Hierarchical Generation



Graph Neural Network

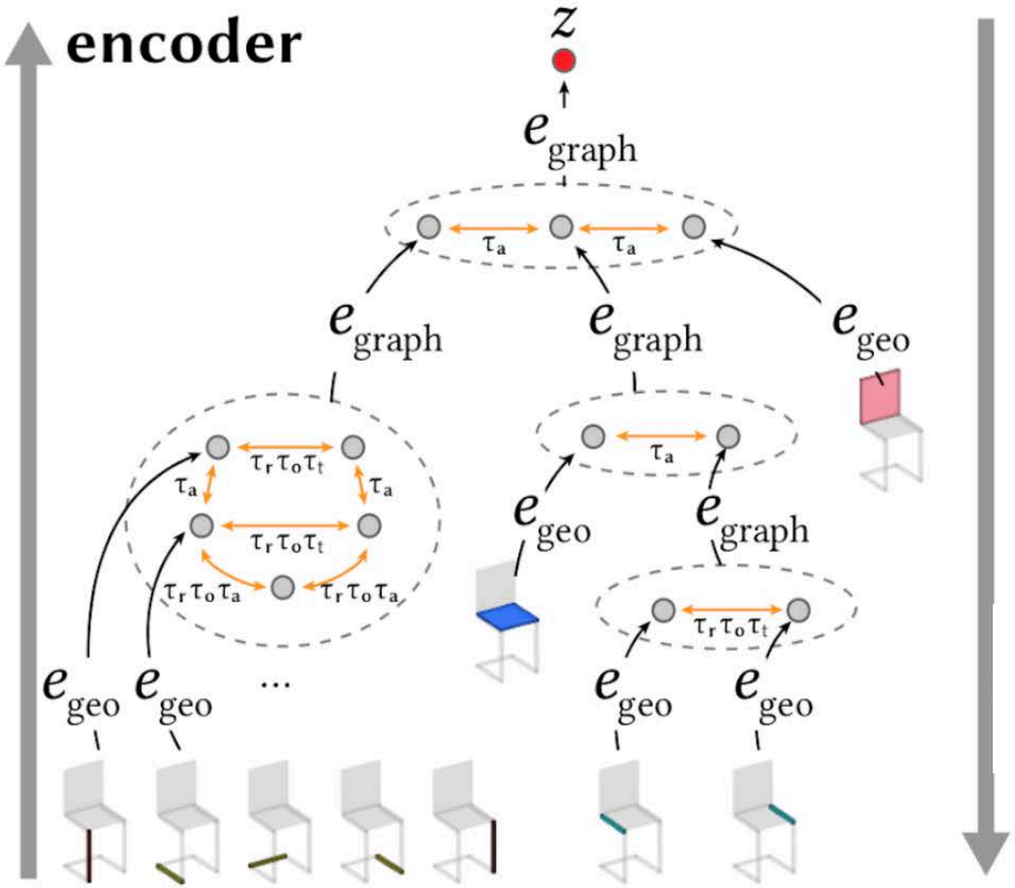
Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

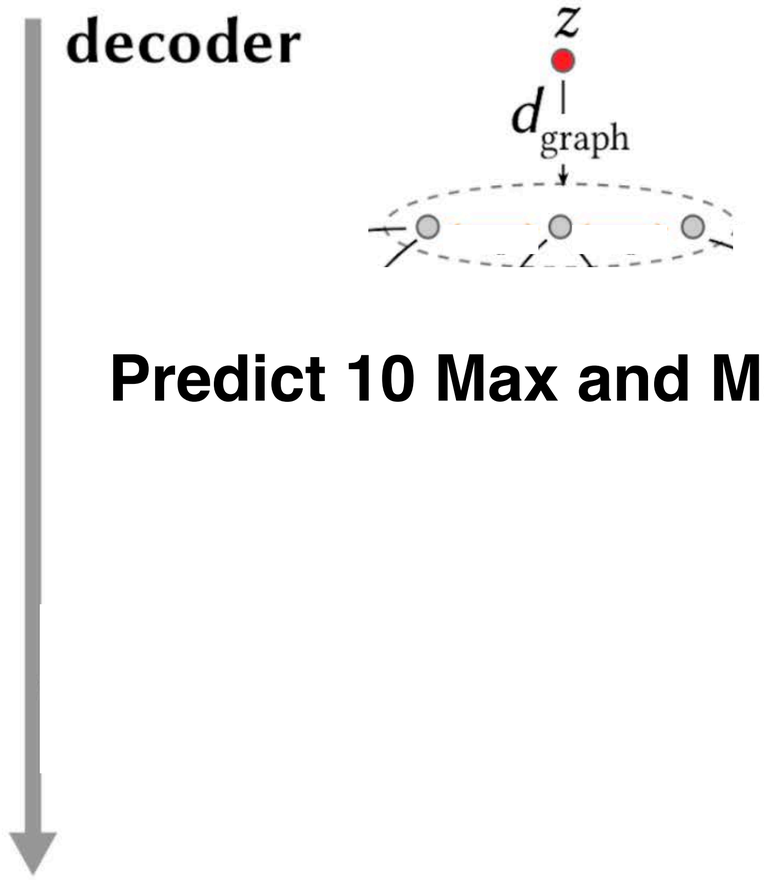
Hierarchical Generation



Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

Hierarchical Generation

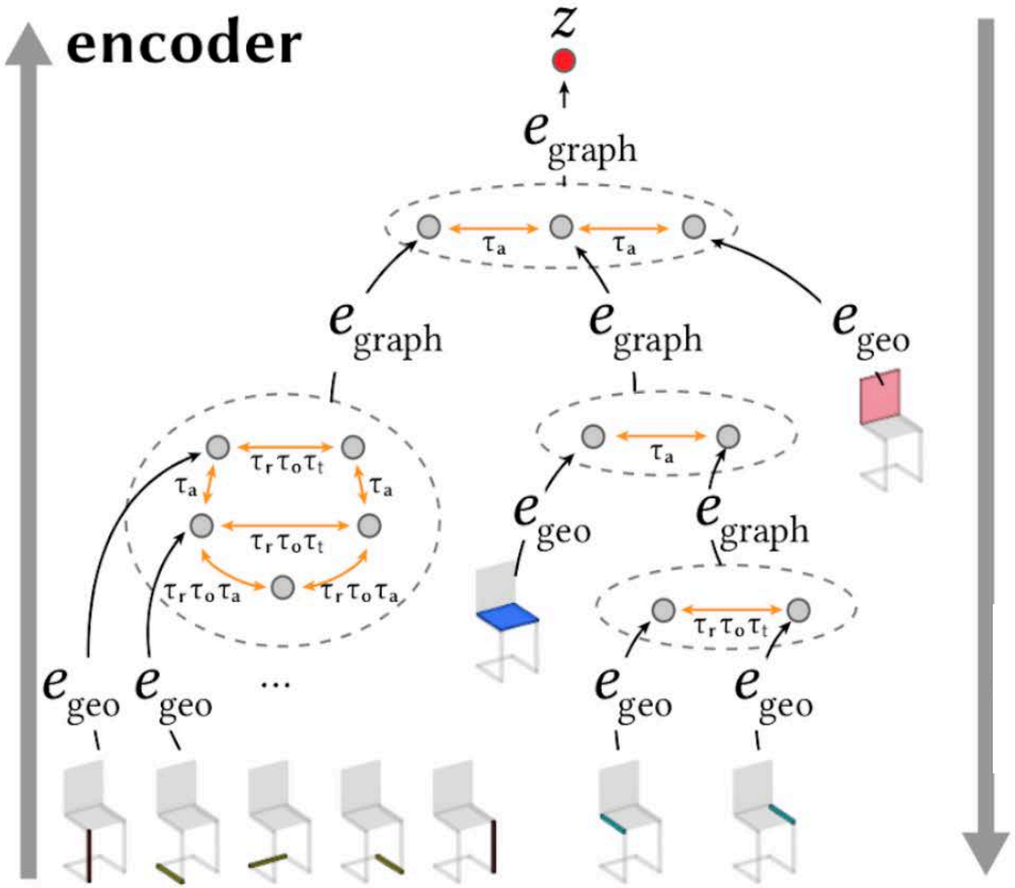


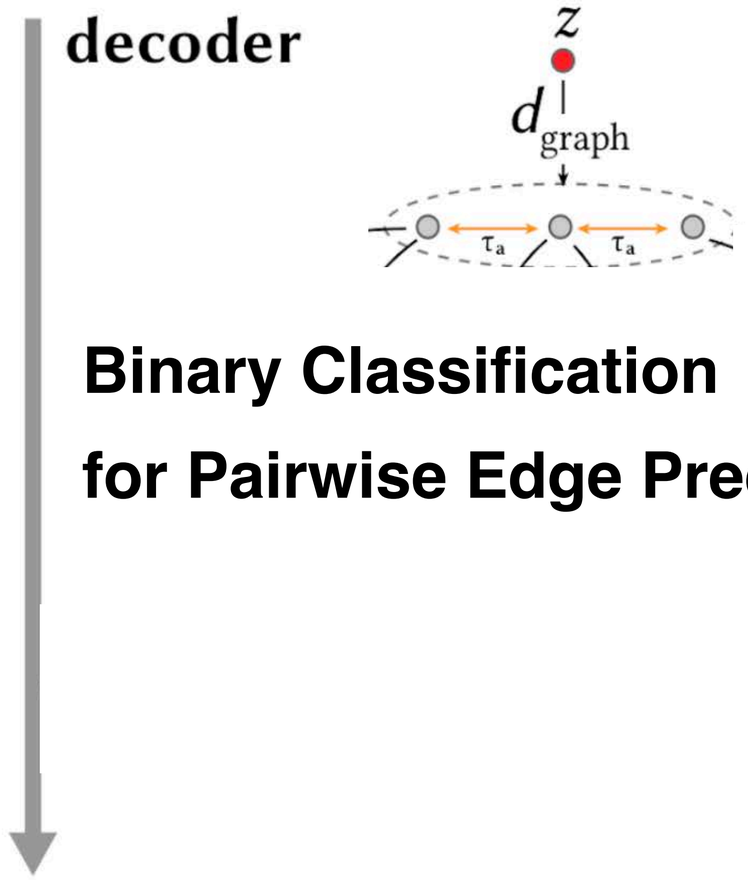


Predict 10 Max and Match

Mo et al., “**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**”, *Siggraph Asia 2019*

Hierarchical Generation



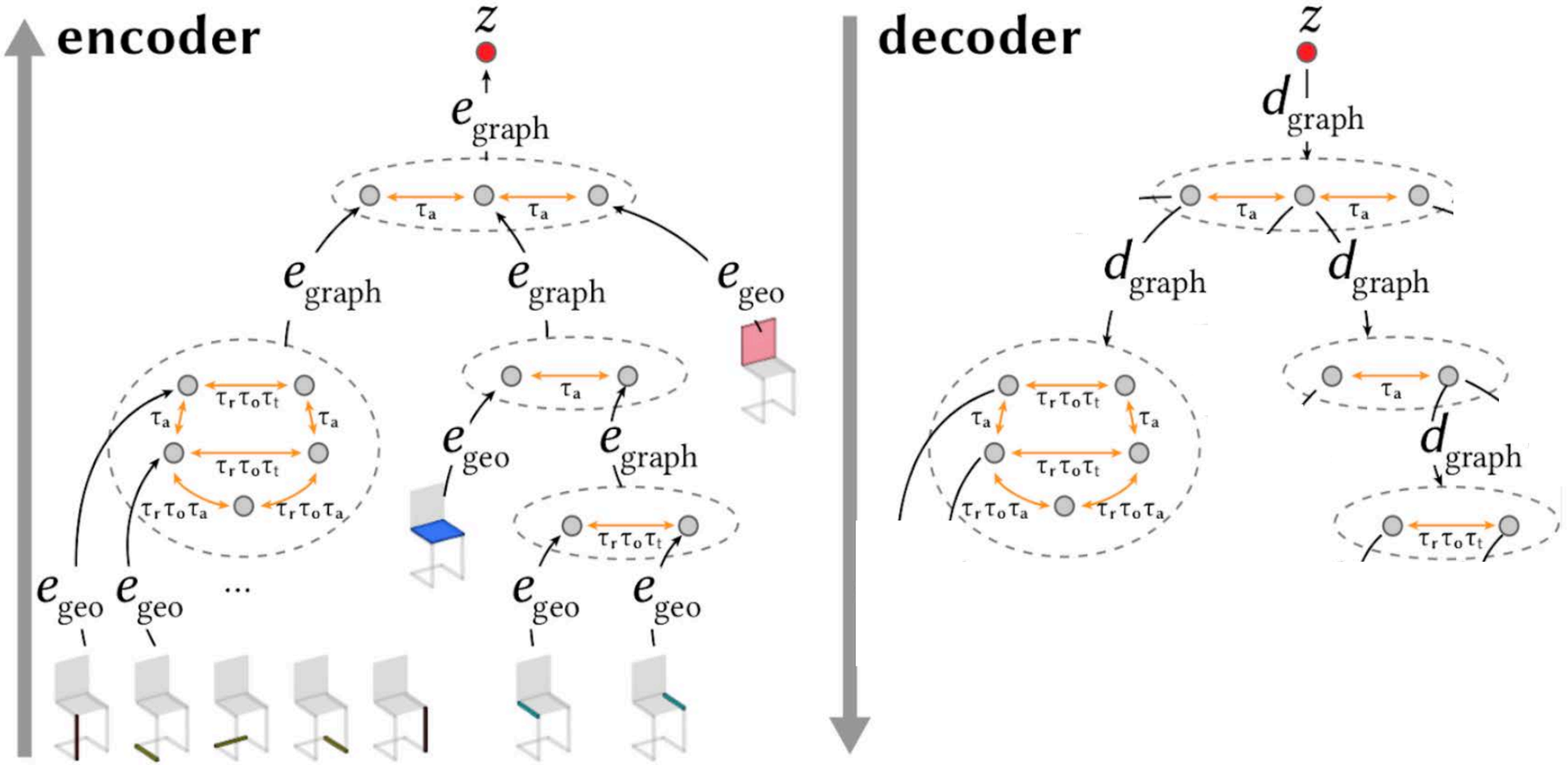


Binary Classification

for Pairwise Edge Prediction

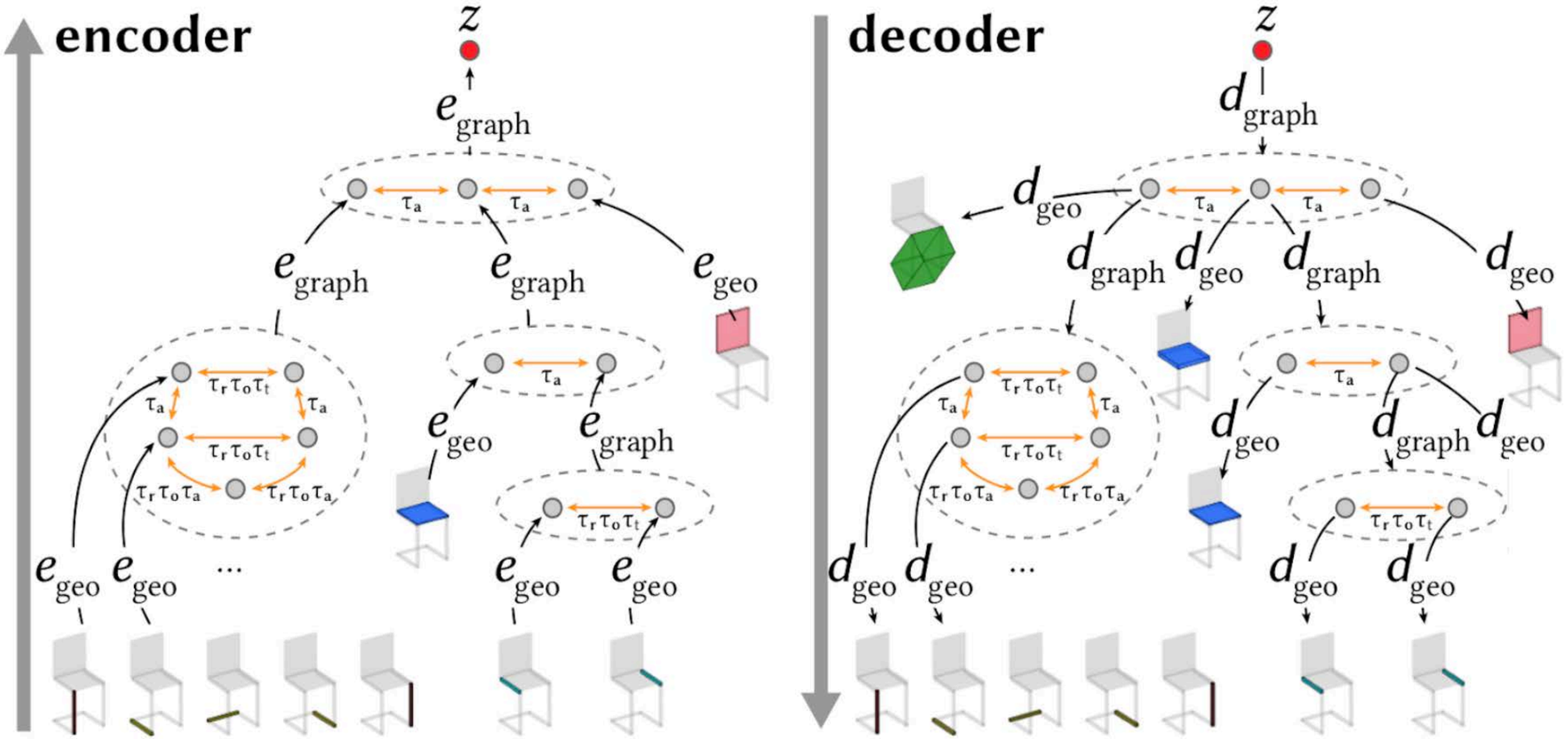
Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

Hierarchical Generation



Mo et al., "**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**", *Siggraph Asia 2019*

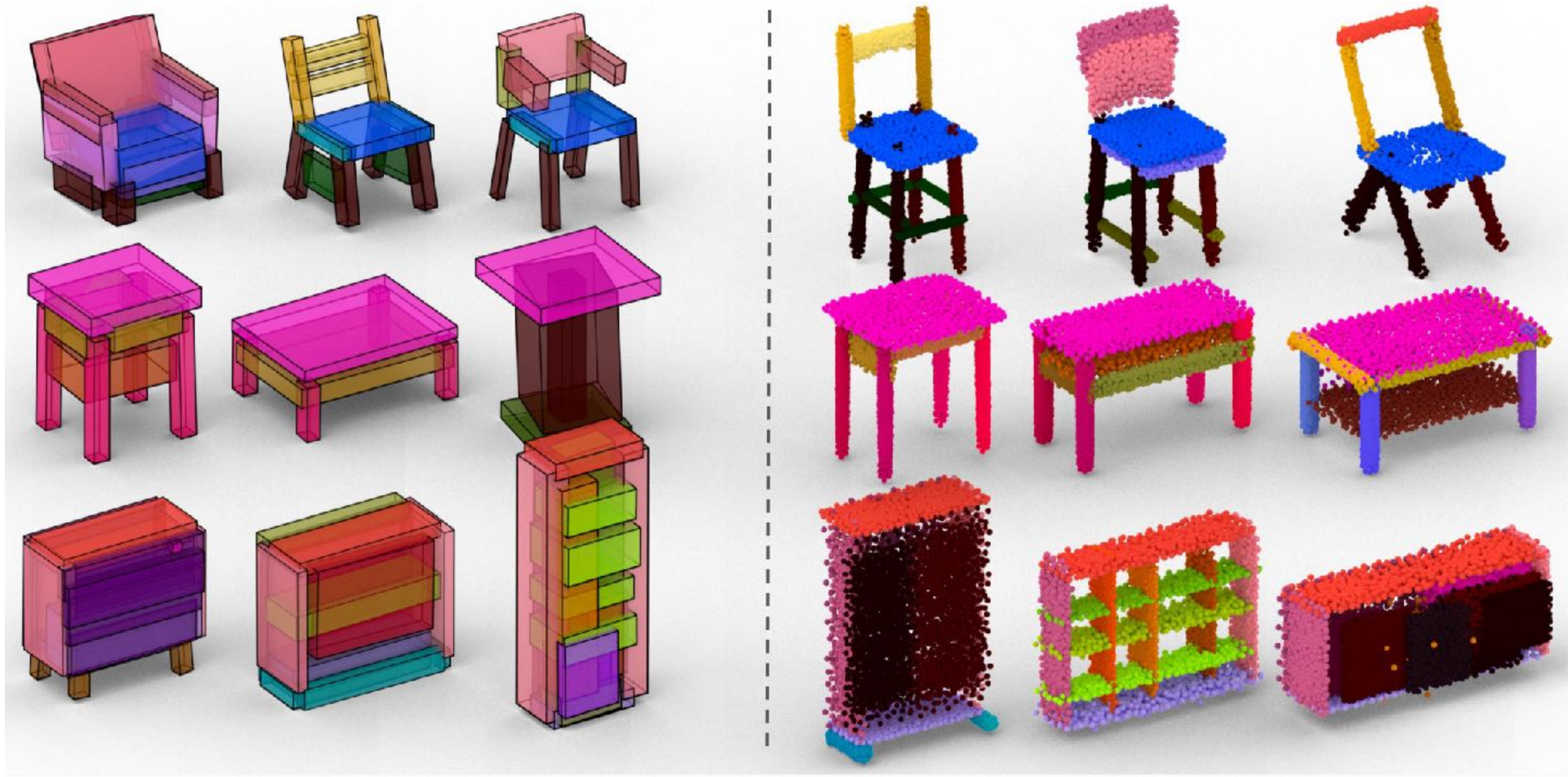
Hierarchical Generation



Recursive Neural Network

Mo et al., “**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**”, *Siggraph Asia 2019*

Hierarchical Generation: Results



Mo et al., “**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**”, *Siggraph Asia 2019*

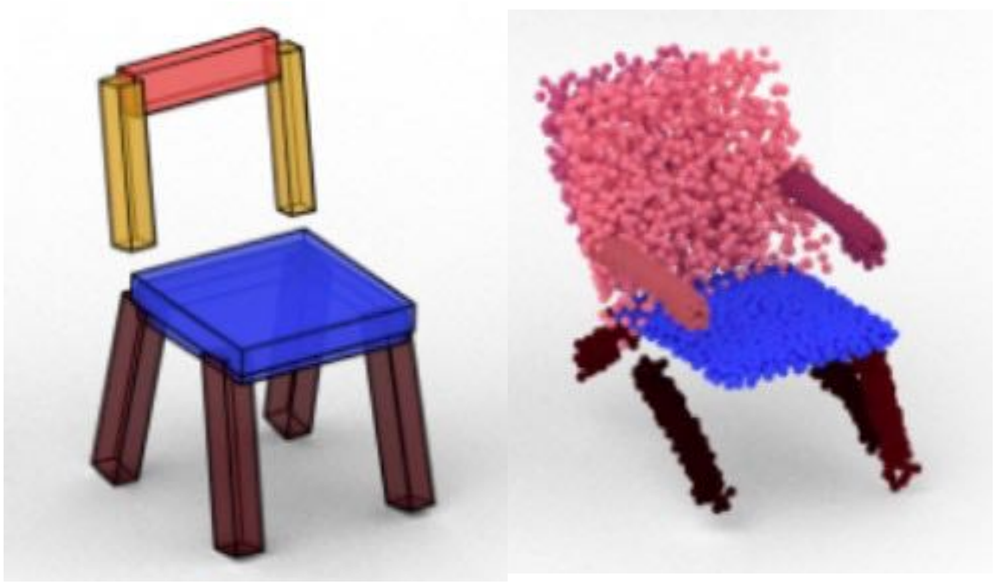
Hierarchical Generation: Results

Use mesh generation for leaf nodes

Yang and Mo et al., “**DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation**”, *ACM ToG 2021*

Part Connection Issue

There could be floating parts



Mo et al., “**StructureNet: Hierarchical Graph Networks for 3D Shape Generation**”, *Siggraph Asia 2019*

Part Connection Issue

Post-processing

Optimize part poses using the predicted part adjacency relationships

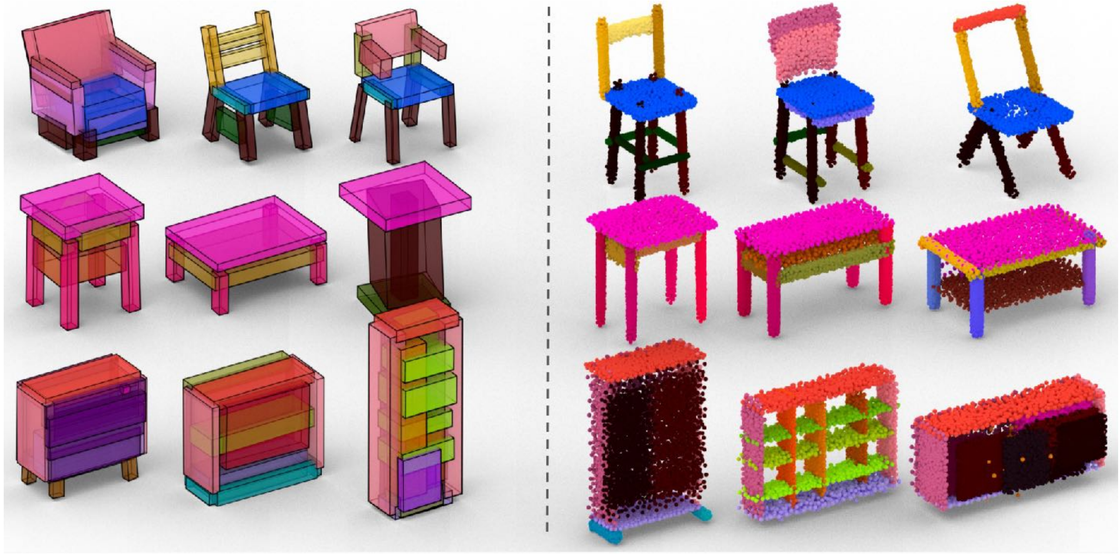
Yang and Mo et al., "**DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation**", *ACM ToG 2021*

Joint Re-synthesis

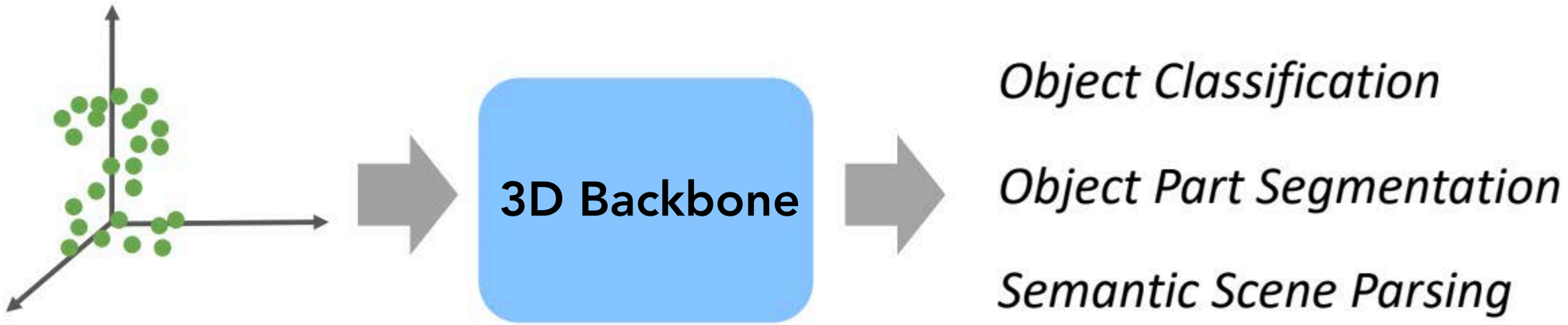
Erode the part connection regions and re-generate the local geometry

Yin et al., "**COALESCE: Component Assembly by Learning to Synthesize Connections**", *3DV 2020*

Summary



Next Time


3D Backbone
Object Classification
Object Part Segmentation
Semantic Scene Parsing