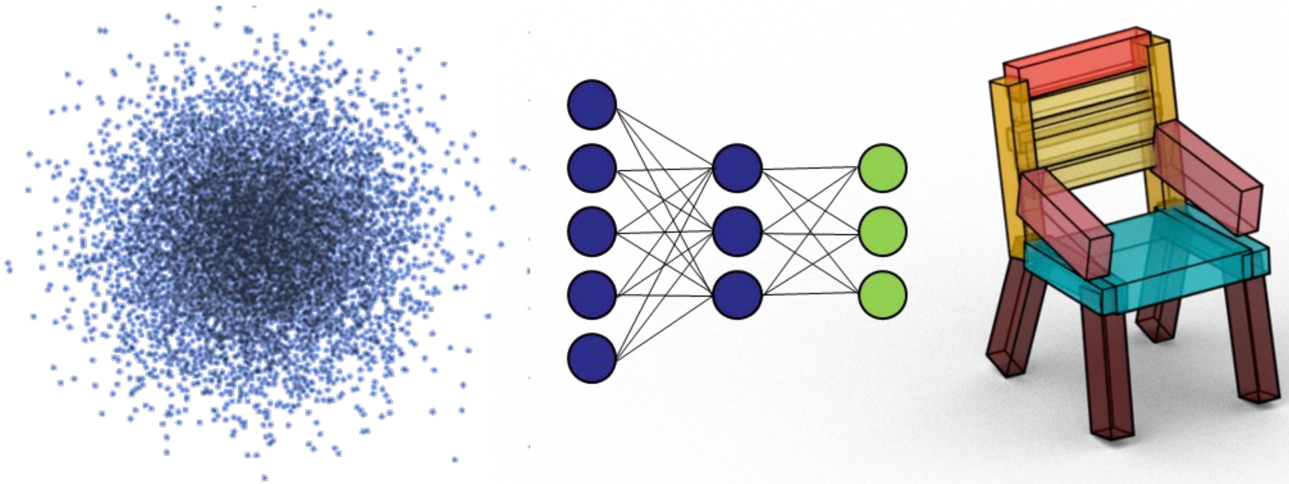
CS348n: Neural Representations and Generative Models for 3D Geometry



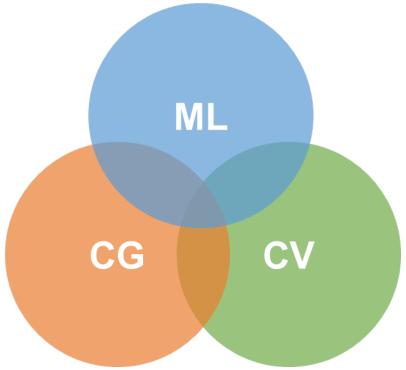
Leonidas Guibas
Computer Science Department
Stanford University

Leonidas Guibas Laboratory

Geometric Computing

Brief Biography

- Education: CS Ph.D. Stanford University
  - Advisor: Donald E. Knuth
- Main Other Employers: Xerox PARC, DEC SRC, MIT
- Current Position: Paul Pigott Professor in the School of Engineering, Stanford University, CS + EE (courtesy)
- Other appointments at: University of Athens (Greece), National University of Singapore, Tokyo National Institute of Informatics, Swiss Federal Institute of Technology (ETH), Google Research, Hong Kong University of Science and Technology, Tsinghua-Berkeley Shenzhen Institute, Facebook AI Research



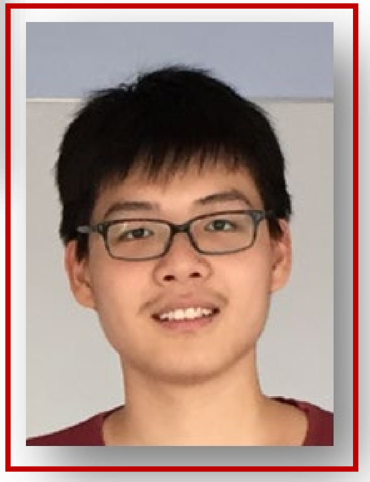
Machine Learning

Computer Vision

Computer Graphics

The Class Principals

- Leonidas (Leo) Guibas (CS & EE)
  - Instructor
- Kaichun Mo (CS)
  - Course Assistant (TA)
- Carrie Petersen (CS)
  - Admin



Also, a number of guest speakers …

Canvas for class videos, etc…

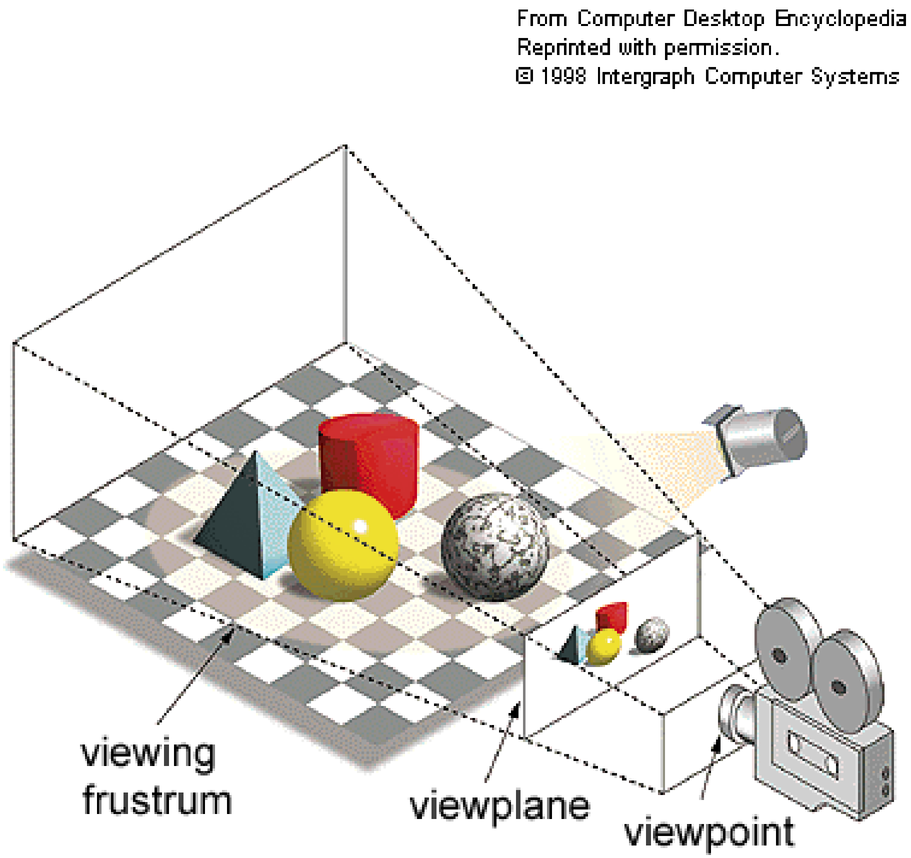
http://cs348n.stanford.edu

Class venue: Zoom, Clark S361, Gates 105, …

The Course:
Deep Generative Models
for 3D

The Classic Graphics Pipeline

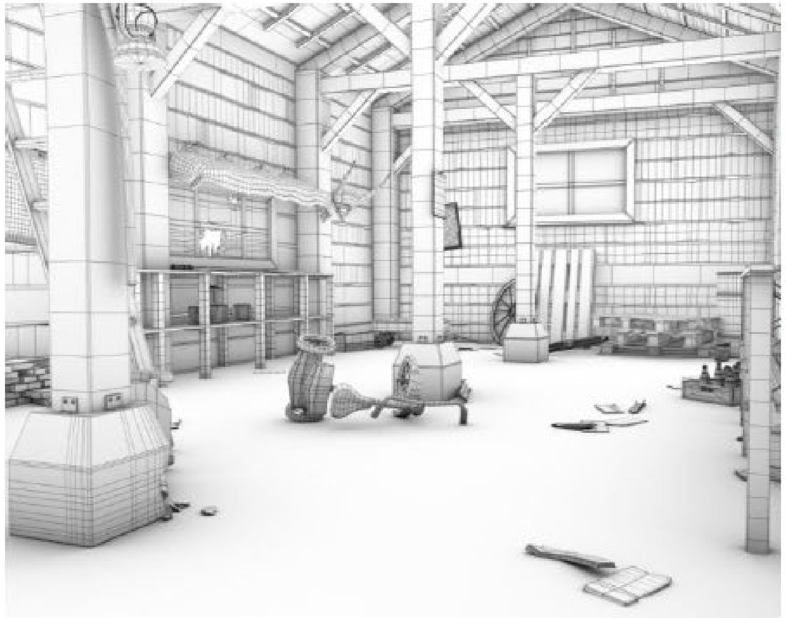
3D Scene:
- Geometry (incl. animation)
- Material
- Lighting



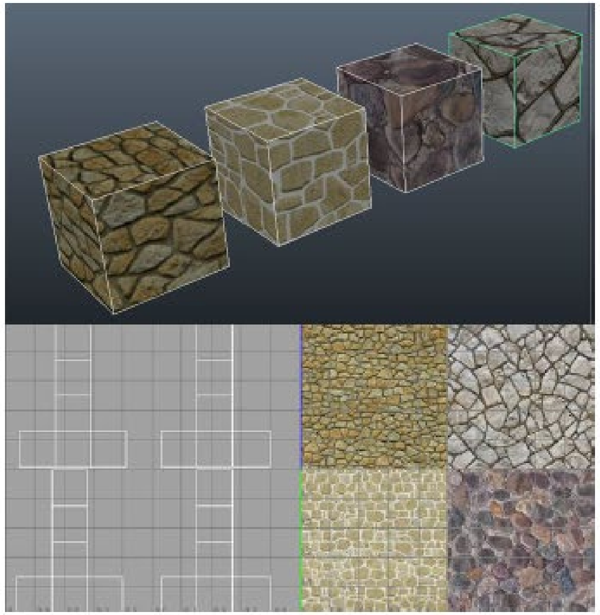
Camera Def.
- Intrinsics
- Often:
    - focal length
    - principal point

Camera View Point
- Extrinsics
- 6 DoF (3 rot, 3 trans)

Need 3D Content for Rendering



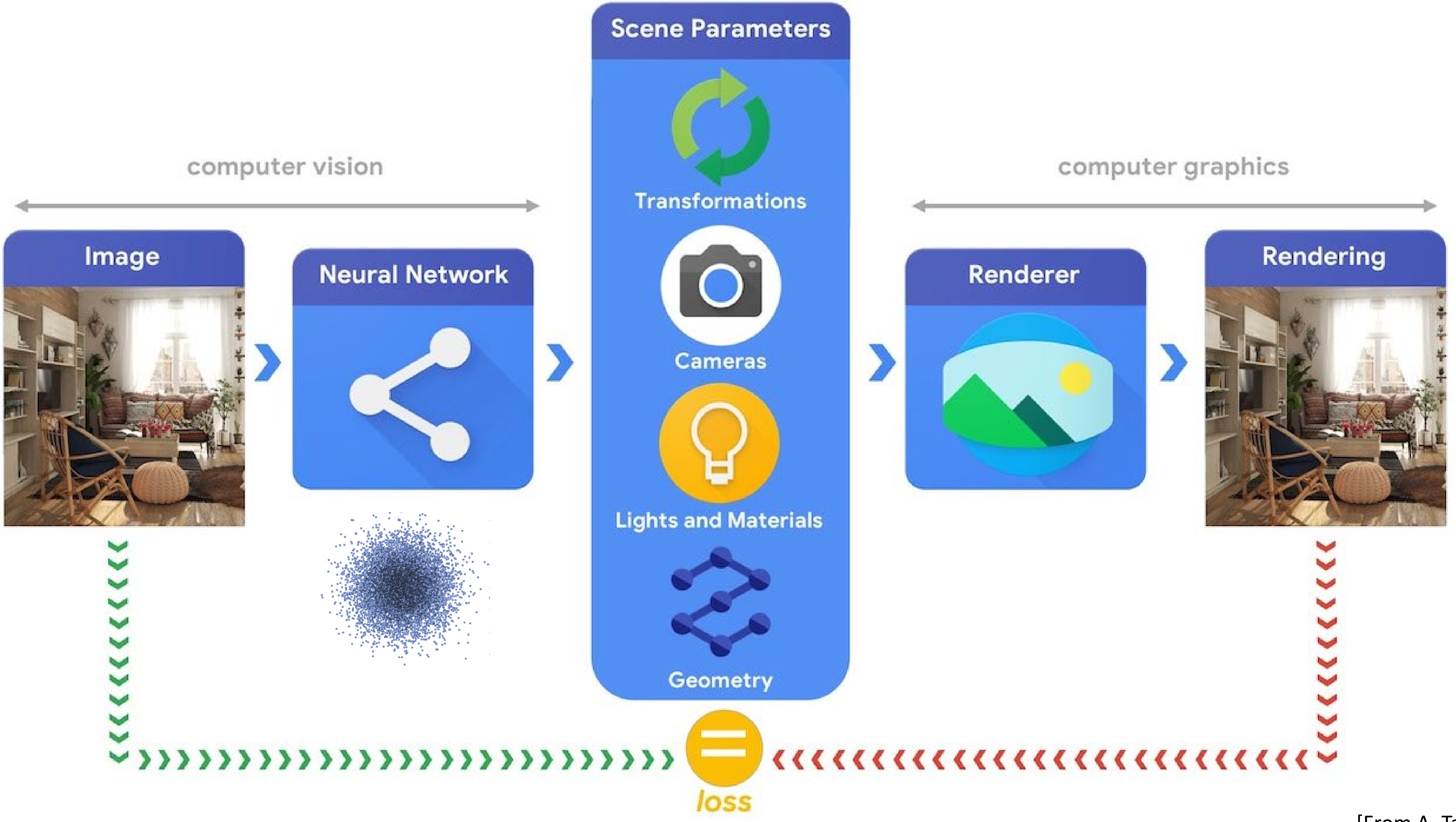
Geometry



Textures and Materials

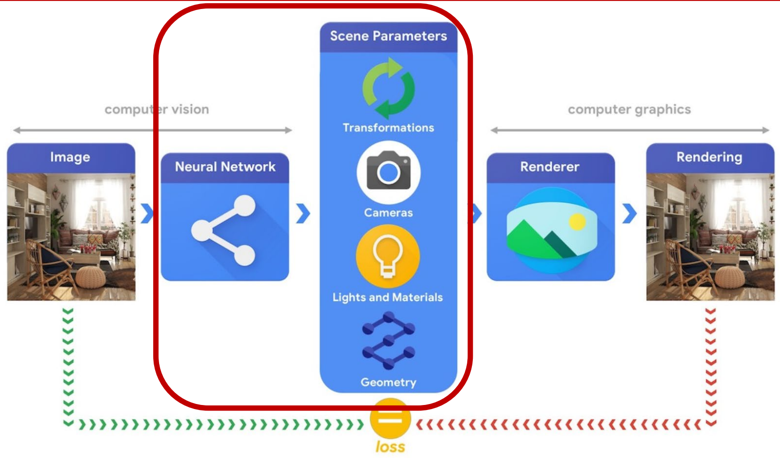
Lighting

Computer Vision vs. Computer Graphics

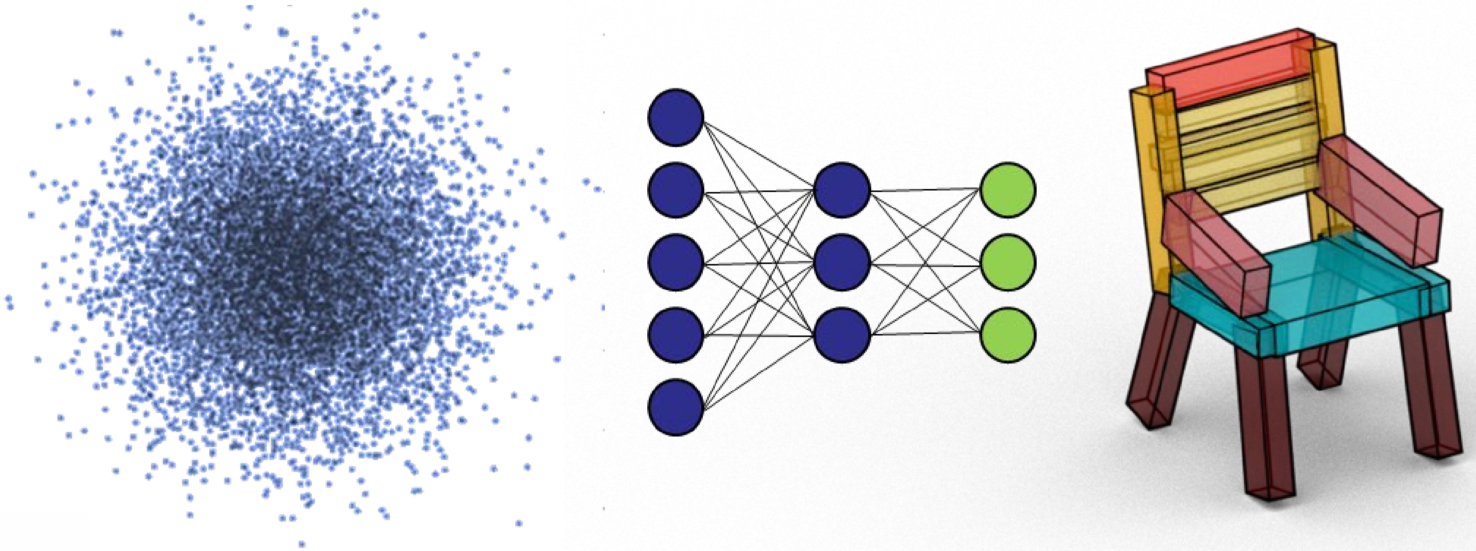


[From A. Tagliasacchi]

Use Neural Networks to Create 3D Objects / Scenes



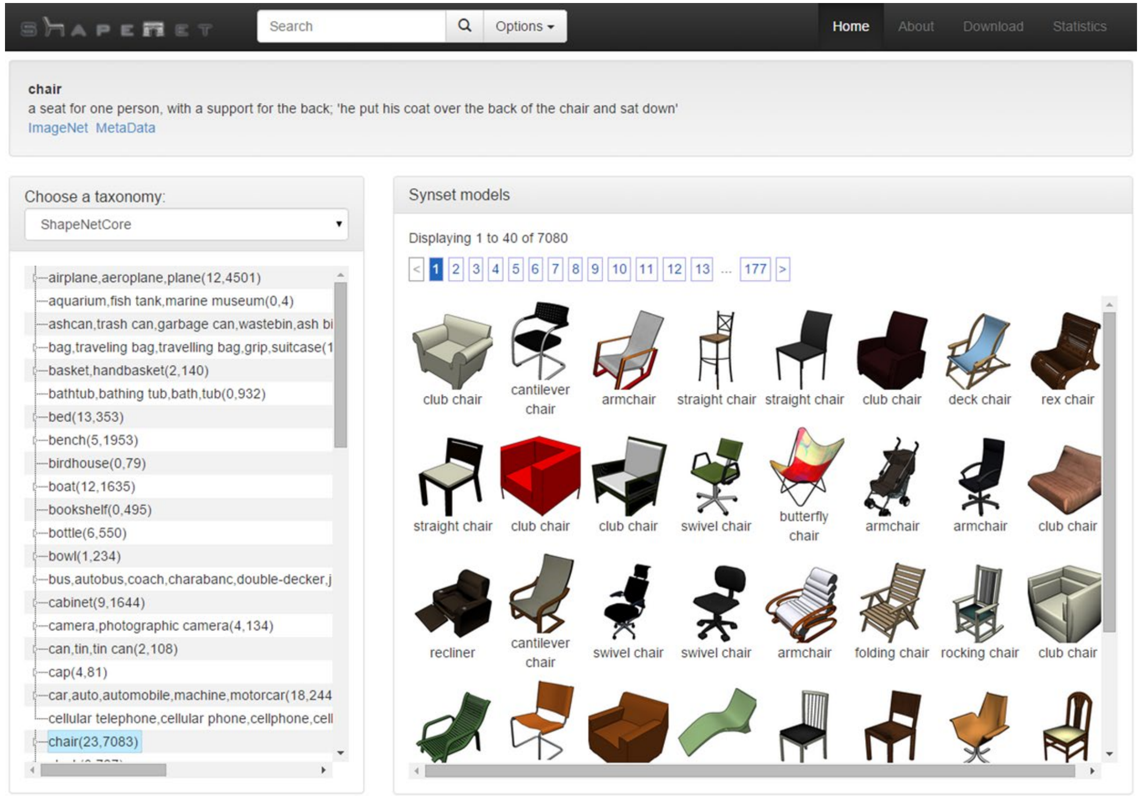
unconditional,
or conditional



Challenges for 3D Machine Learning

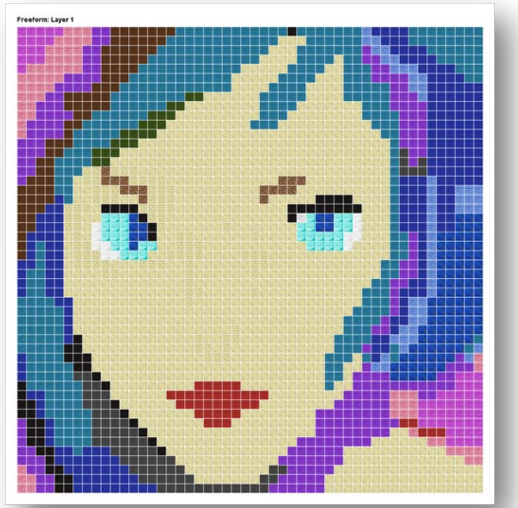
The Role of Supervision, Data Scale: 2D vs 3D

2D —15M Images



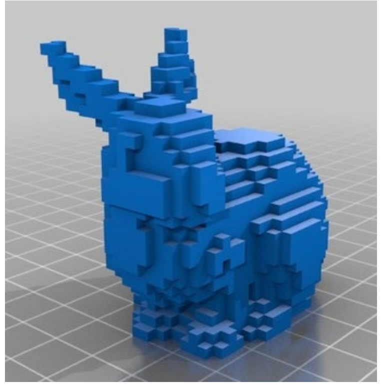
3D —3M Shapes, scarce annotations

Challenge: Unlike 2D, Multiple 3D Representations

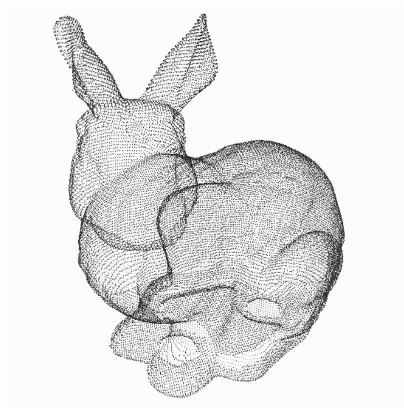


2D

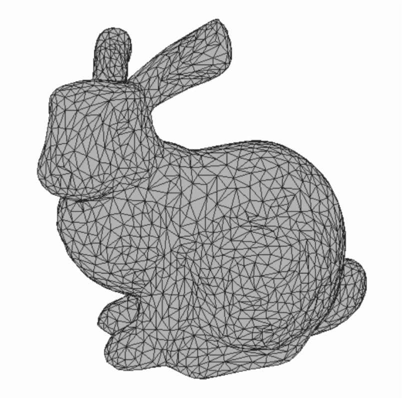
CAD Model



Volumetric



Point Cloud

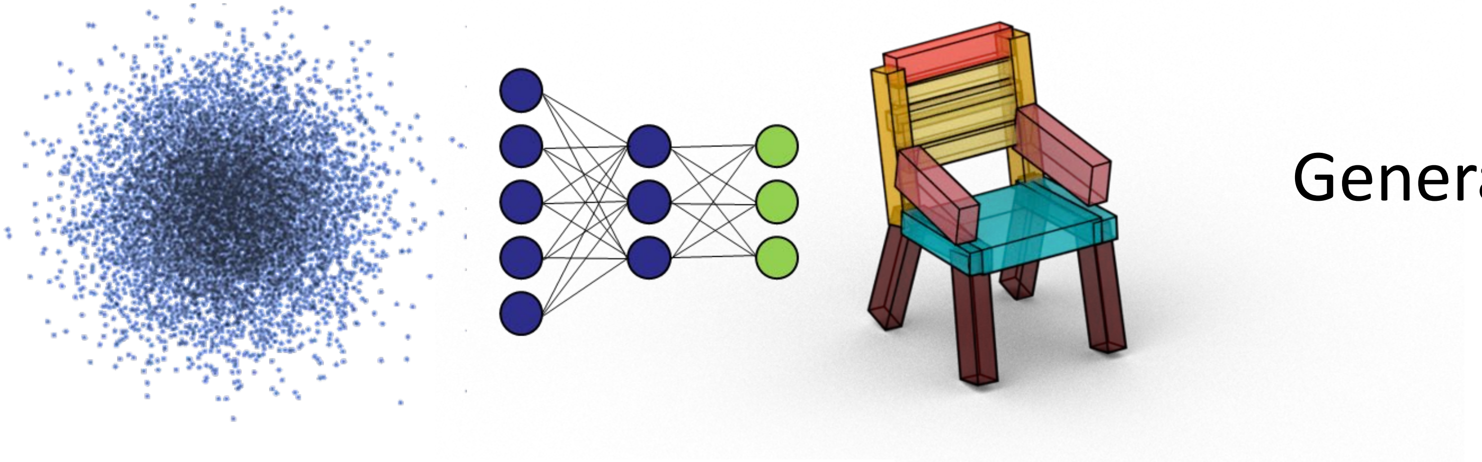


Surface Mesh

Multi-View Images

...

Use NNs to Create Views of 3D Objects / Scenes



Generate 3D Representation

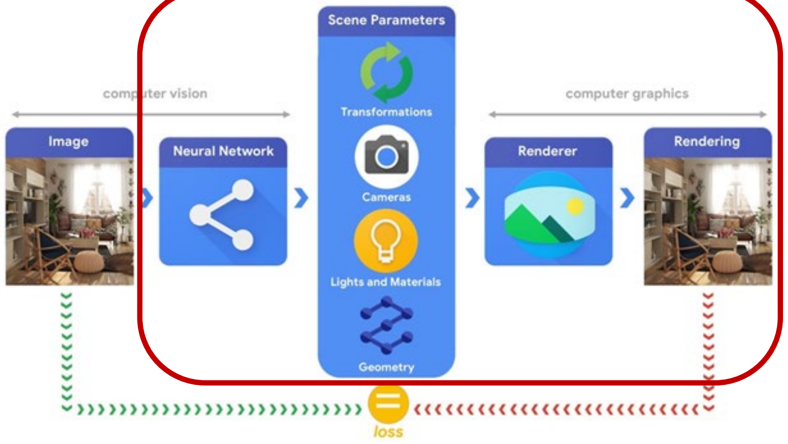
Generate 2D Views

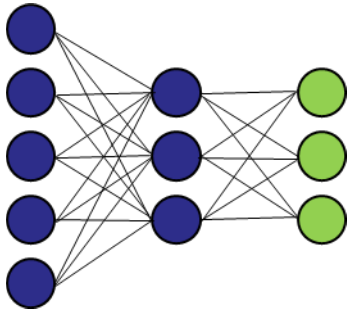
Neural Rendering!

Neural Rendering

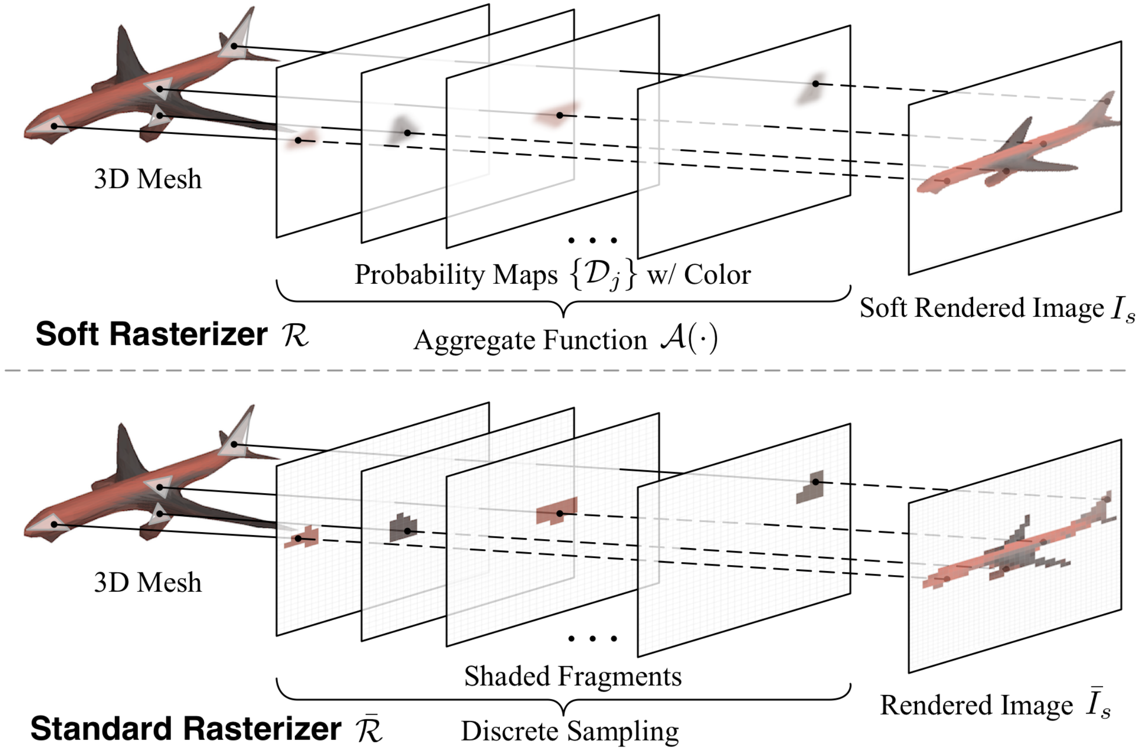
6 DoFCamera
Pose / View Point

Neural Network
-> Encodes entire scene description, lighting, materials, etc.

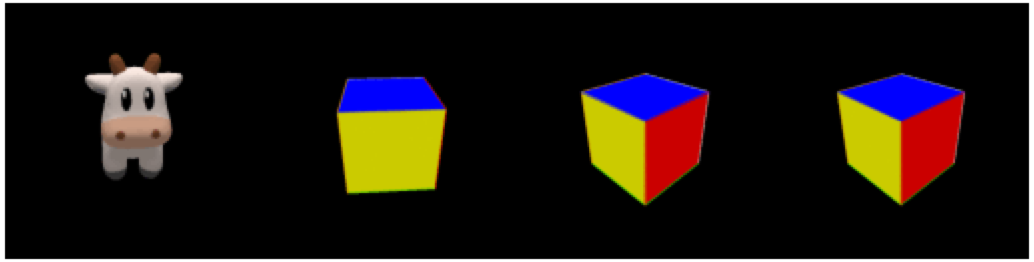




NR Challenges: Differentiability, 3D Consistency



Shichen Liu, Tianye Li, Weikai Chen and Hao Li. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. ICCV'2019.



3D Supervision, from 2D Images!

Where CS348n Fits In

- CS231n – Convolutional Neural Networks for Visual Recognition
- CS233 – Geometric and Topological Data Analysis
- CS236 – Deep Generative Models
- CS348a – Computer Graphics: Geometric Modeling and Processing
- CS348b – Computer Graphics: Image Synthesis Techniques
- CS348i – Computer Graphics in the Era of AI

CS348n is a New Class

There will be rough edges …

Digital 3D Content Creation

Some history

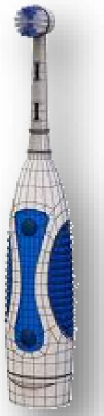
Computer-Aided Shape Design

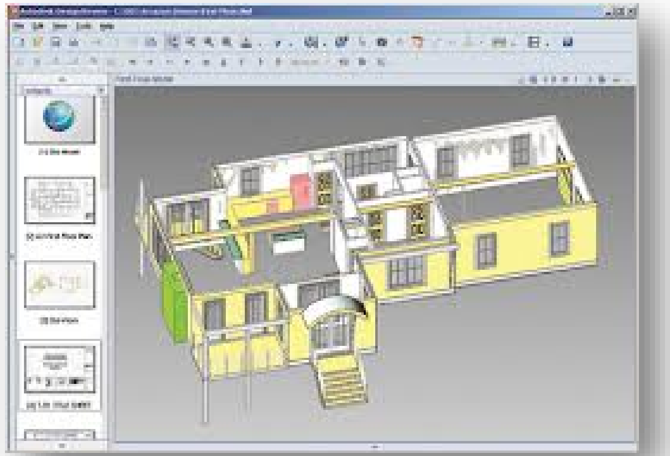
Geometric Modeling and Processing

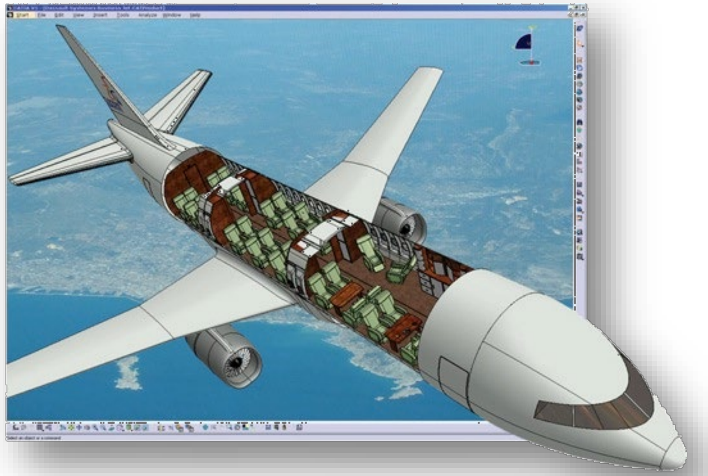
What is computer aided design (CAD), geometric modeling?

Broad goal – digital twins of physical objects:

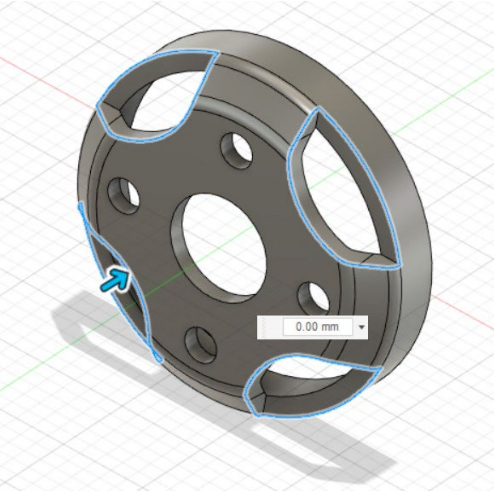
To create mathematical models and practical tools for the digital representation and manipulation of 2D/3D shapes and their physical attributes.

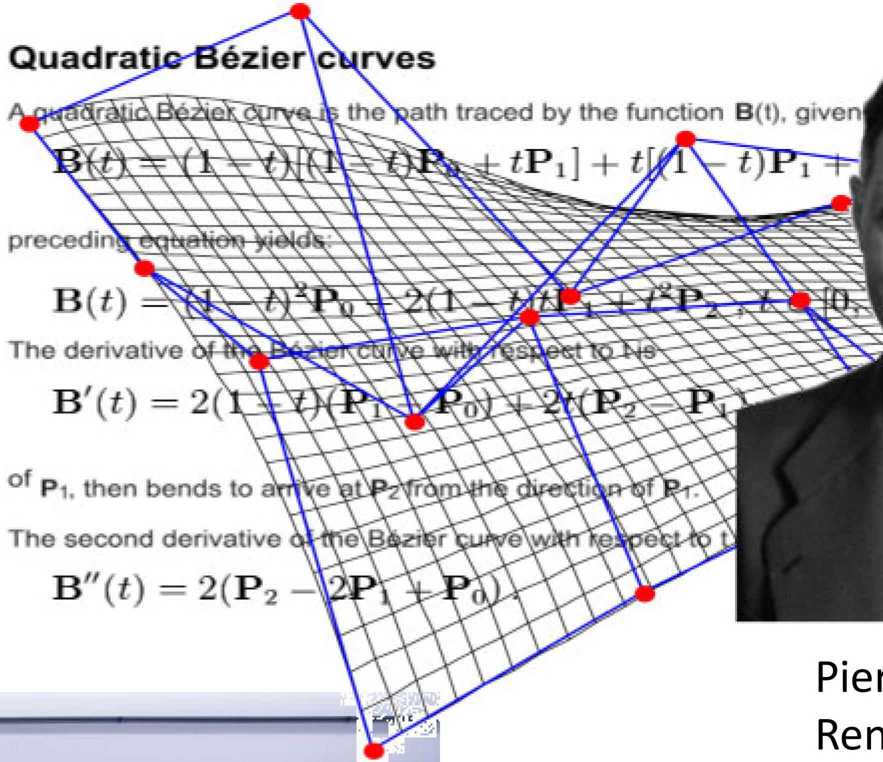


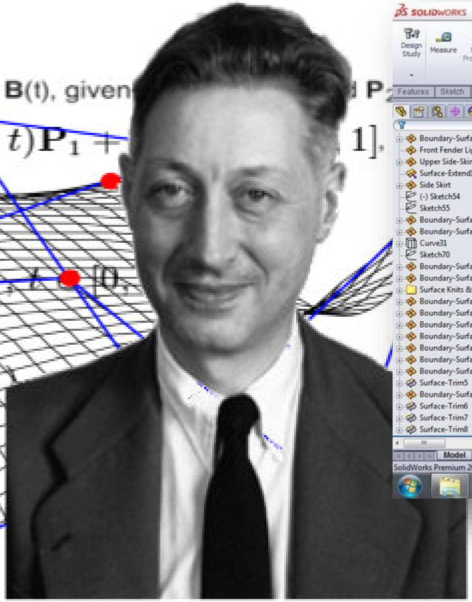




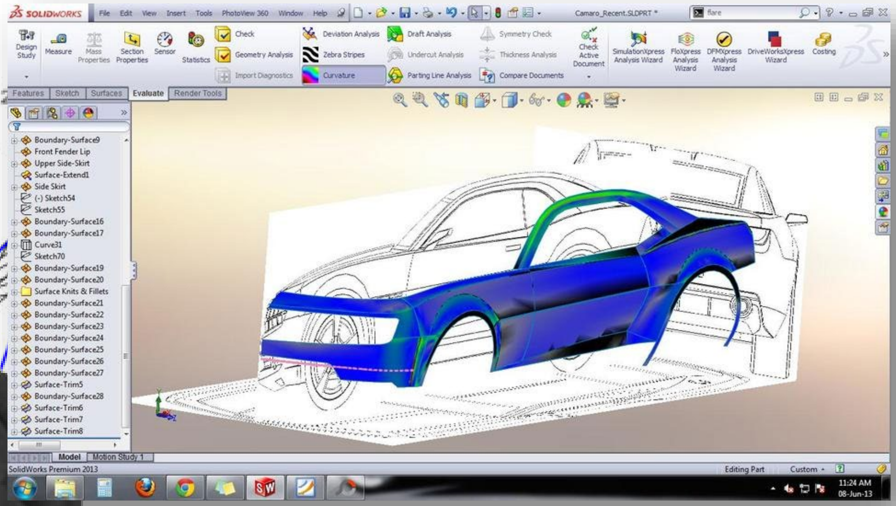
GM Originated in the CAD Industry ~1950

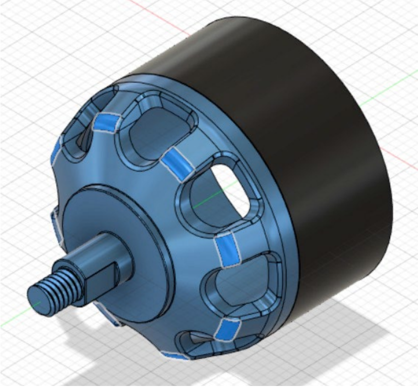


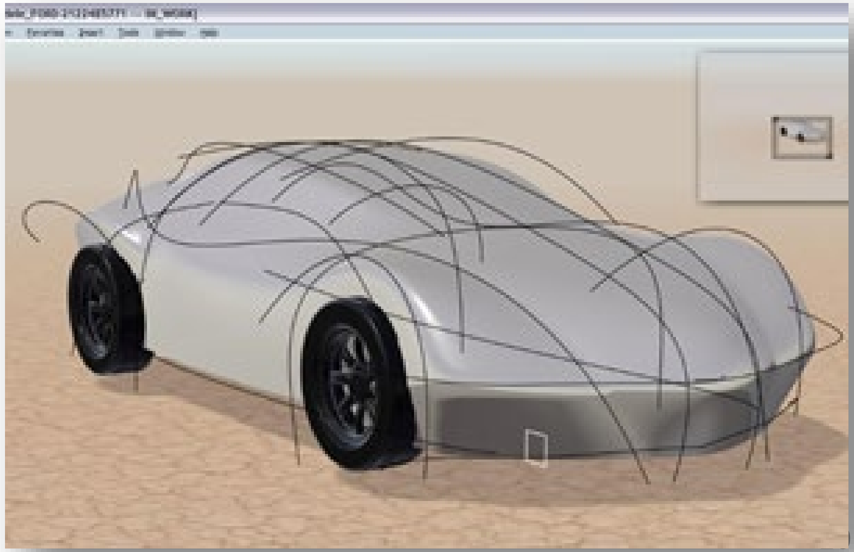




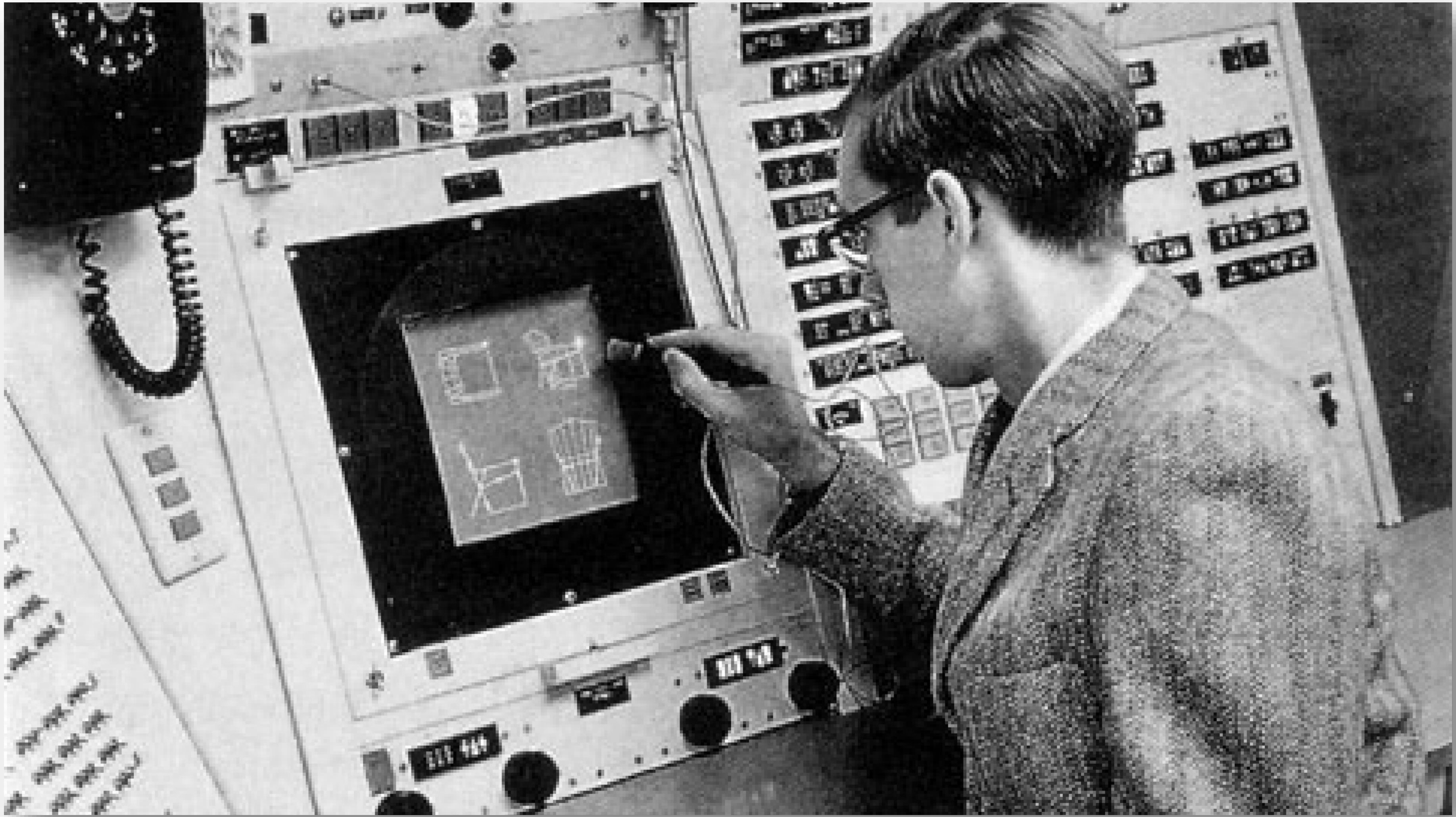
Pierre Bézier
Renault







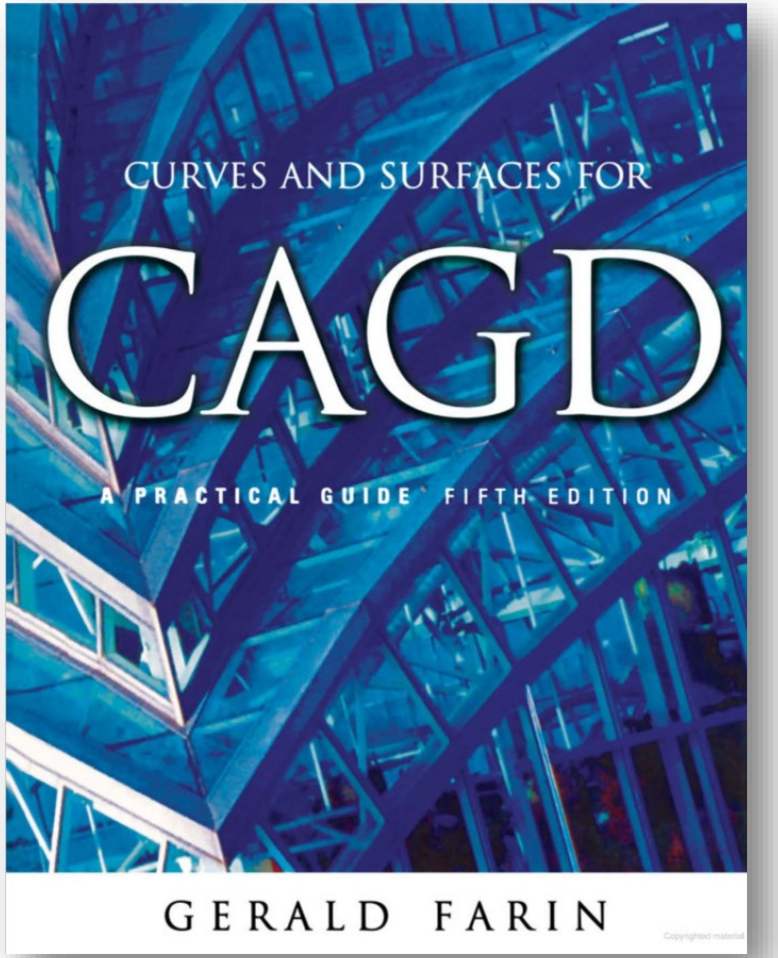
Ivan Sutherland, Sketchpad (1963)

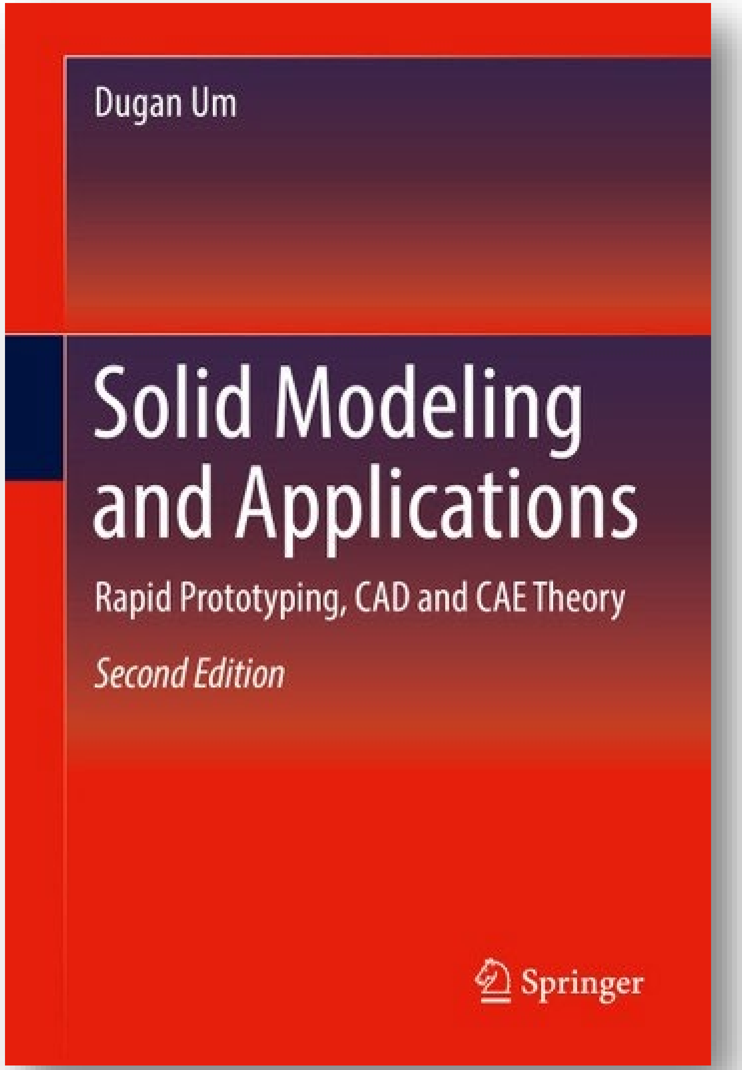


How shape models arise: human design

"A Man-Machine Graphical Communication System"

Many Textbooks





Ab Initio Design: Many Software Environments

Autodesk Fusion 360

SolidWorks

Dassault Systèmes CATIA

Historical Role of 3D Modeling

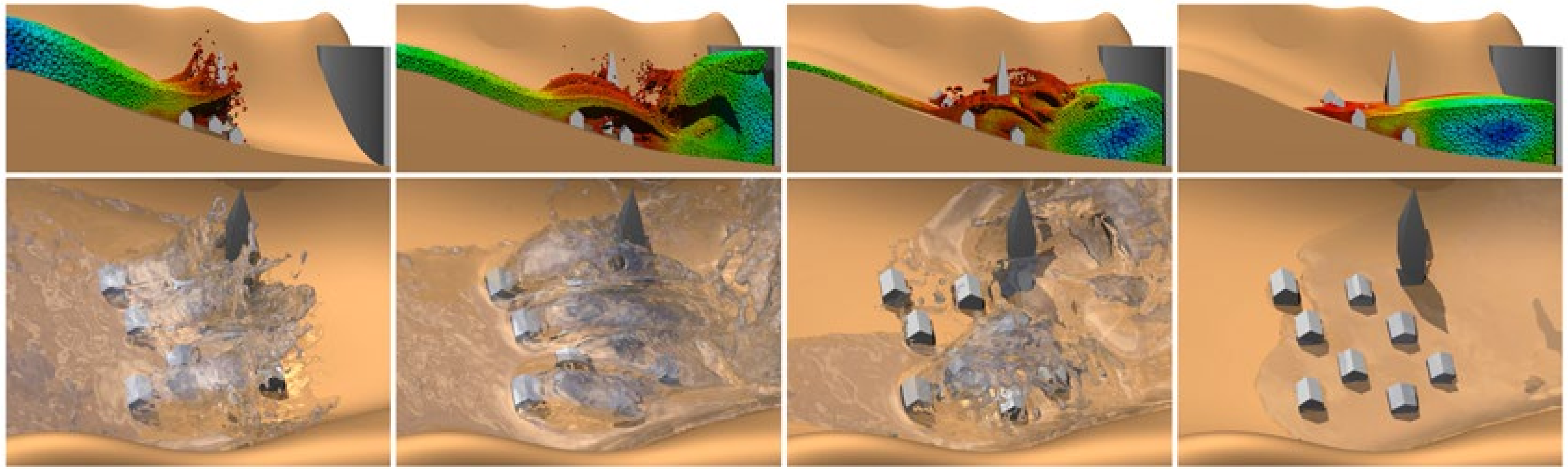

Beautiful synthetic imagery (ads, etc)


Computer games


Movie special effects

Computer Graphics

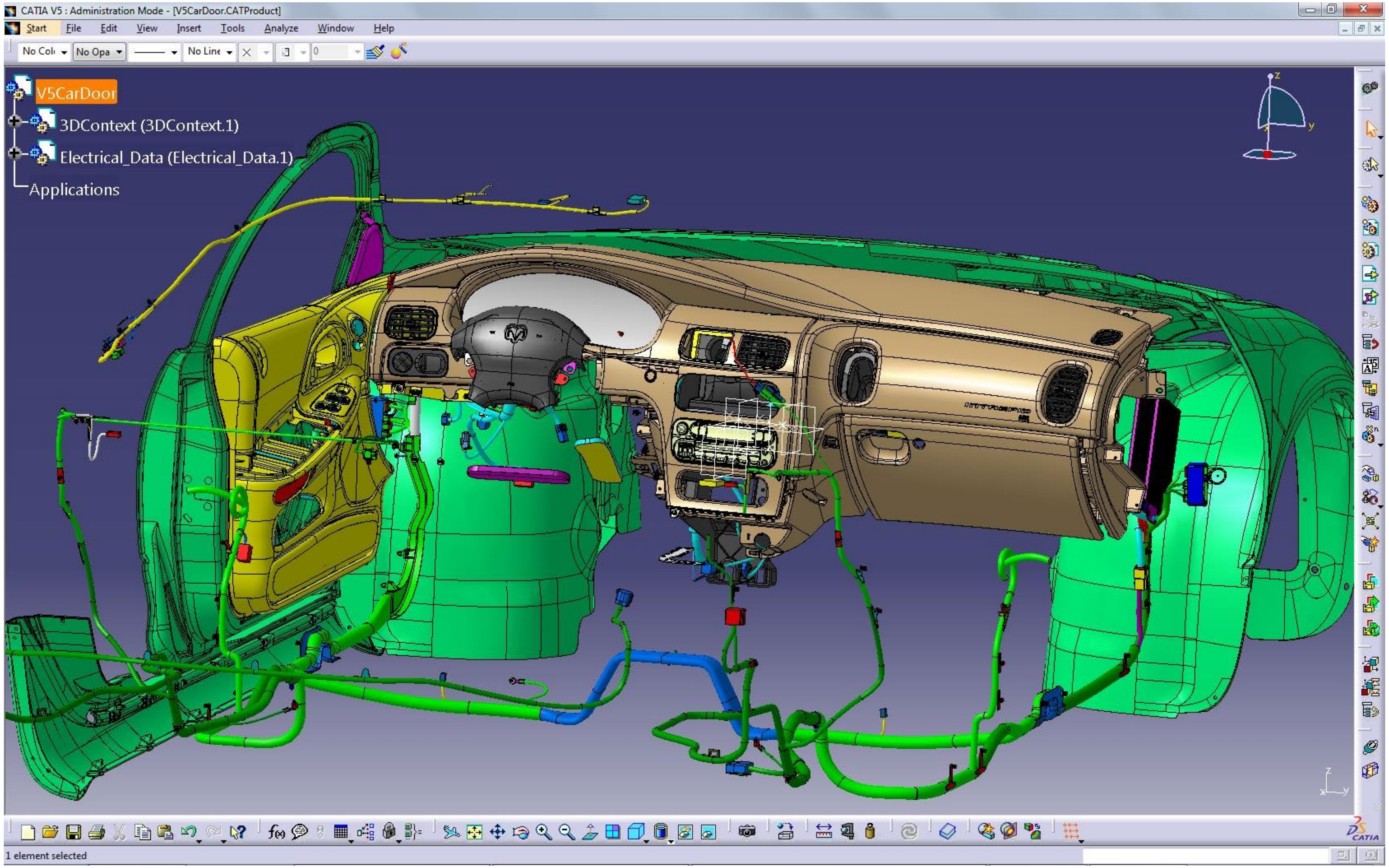
Physically-based simulation



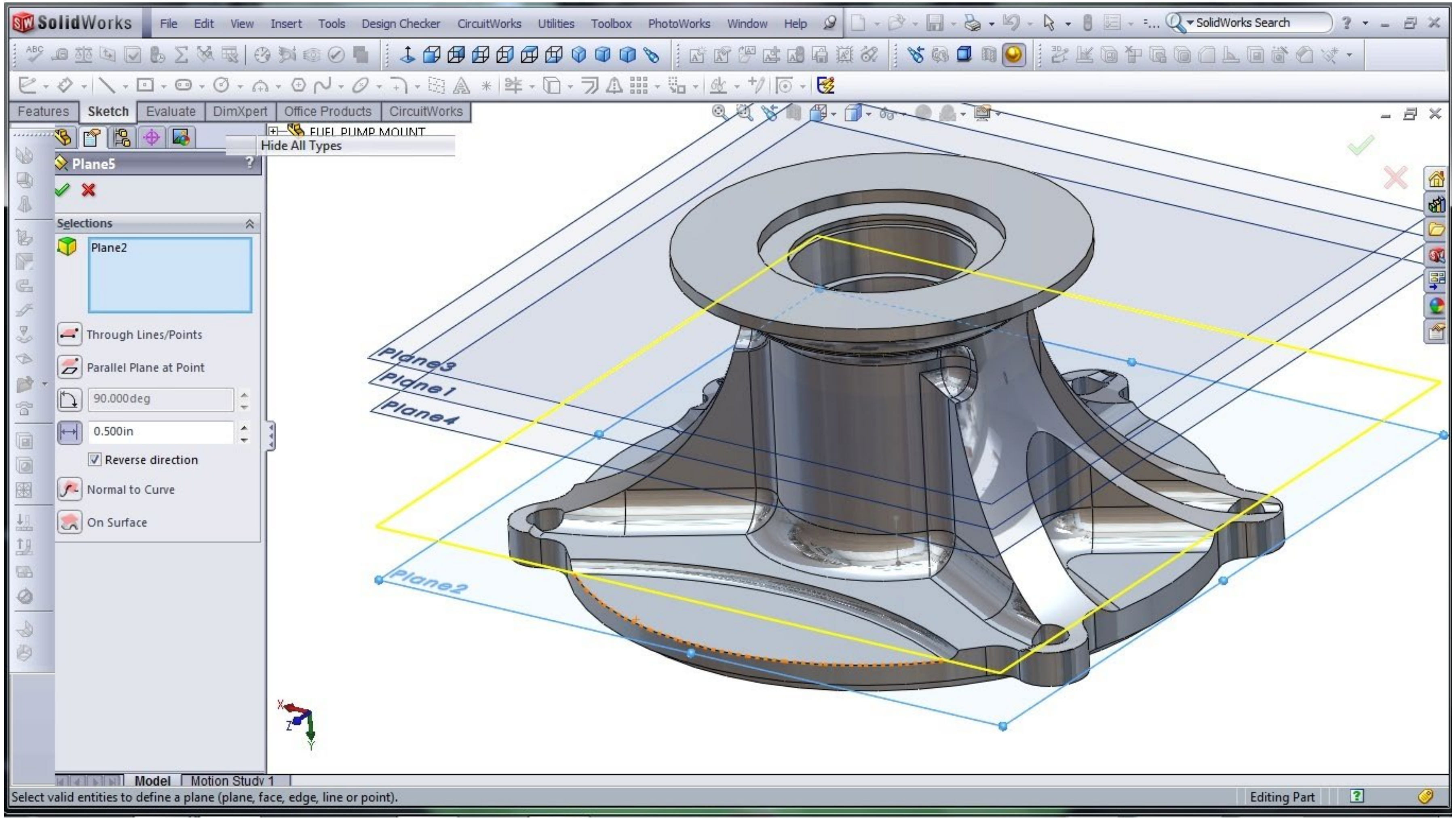
Science

Engineering

CAD Modeling is Hard — Requires Specialists



CAD Modeling is Hard — Requires Specialists

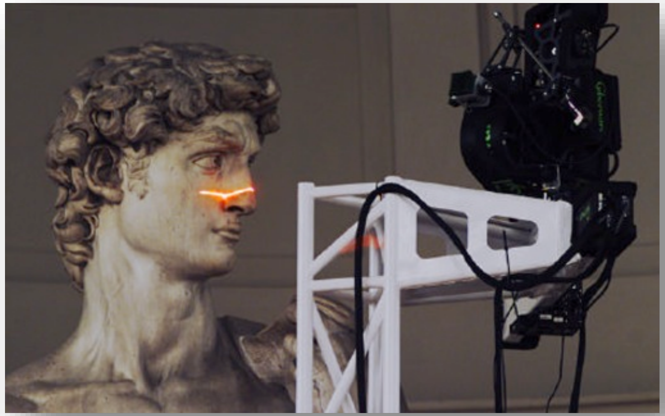


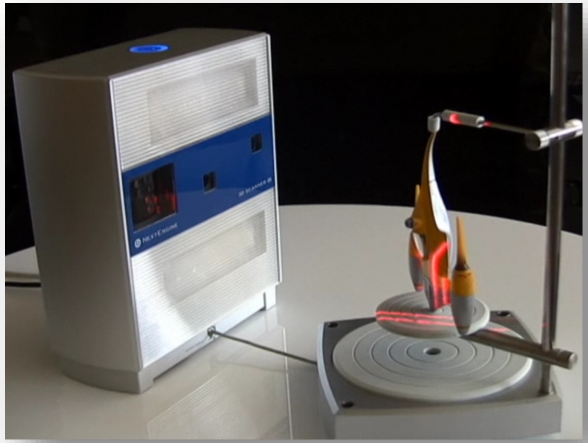
Physical Shape Acquisition

Geometry Capture

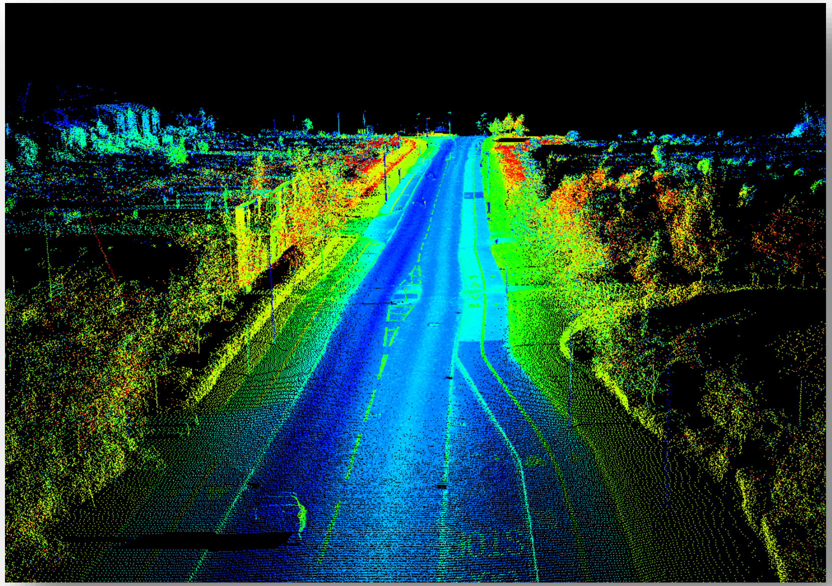
Acquisition by 3D Scanners

- Acquired shapes:

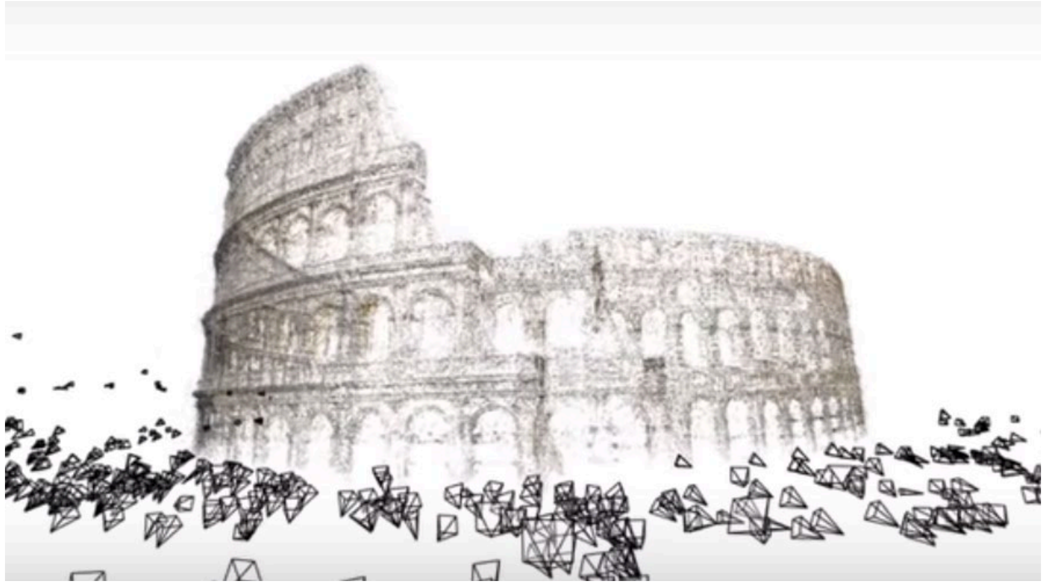




Point Clouds from Many Sensor Types



Lidar point clouds (LizardTech)



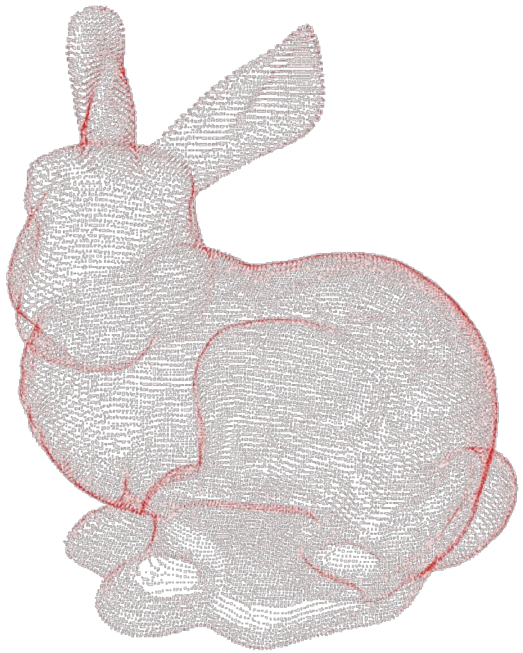
Structure from motion (Microsoft)

Depth camera (Intel)

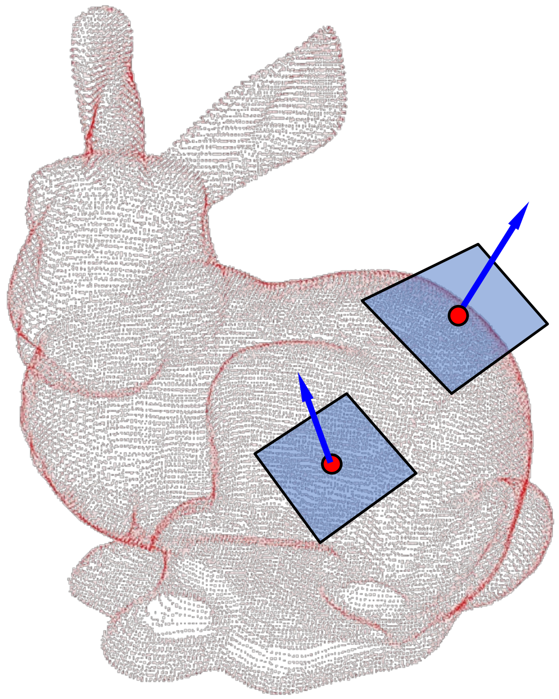
Stanford Bunny

Point Clouds

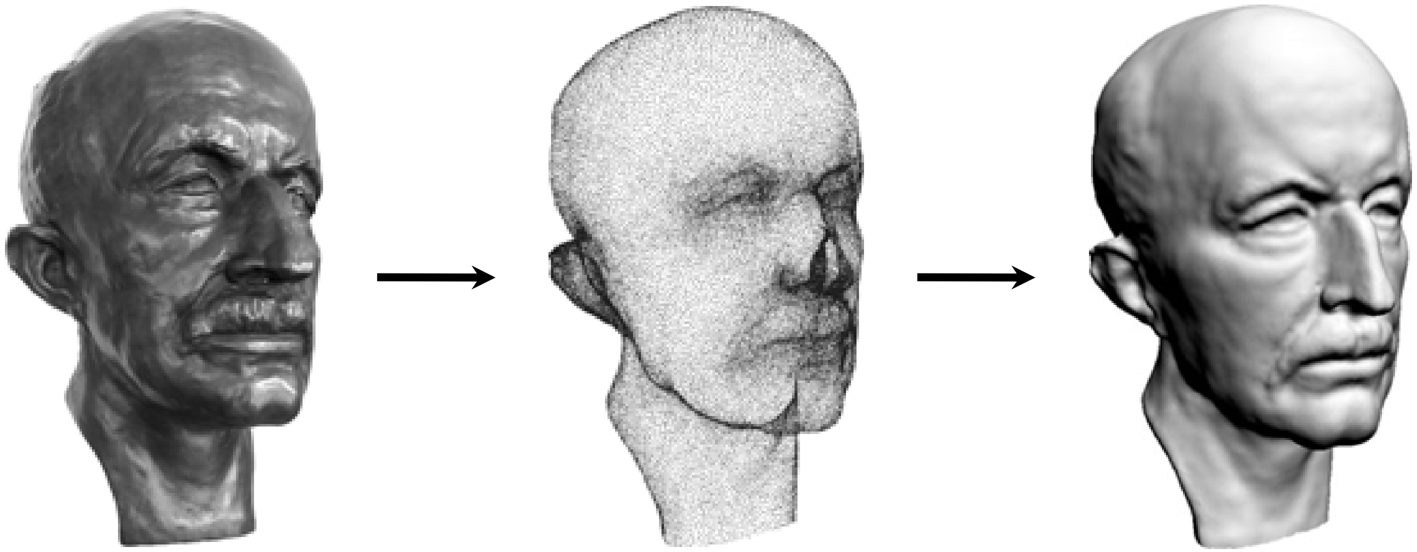
- Simplest representation: **only points,** no connectivity
- Collection of (x,y,z) coordinates, possibly with normals



Stanford bunny



From Point Clouds to Surfaces



physical
model

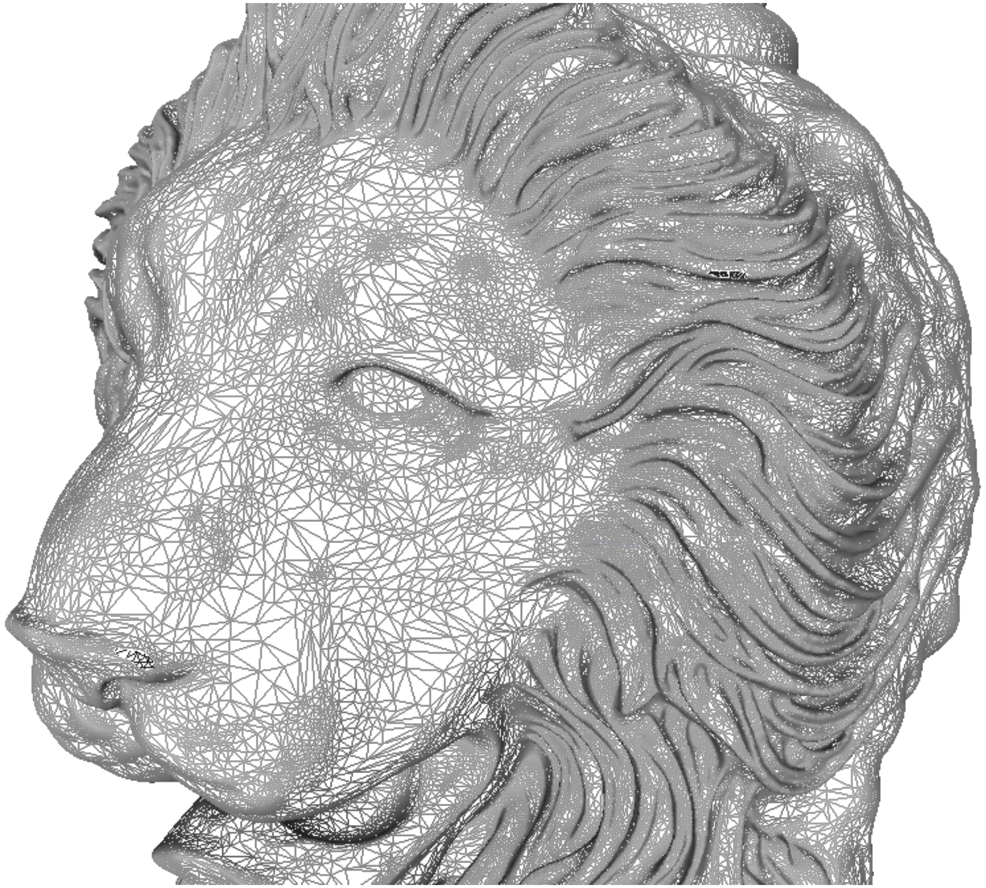
acquired
point cloud

Reconstructed mesh or CAD
3D model

Geometry Processing Pipeline

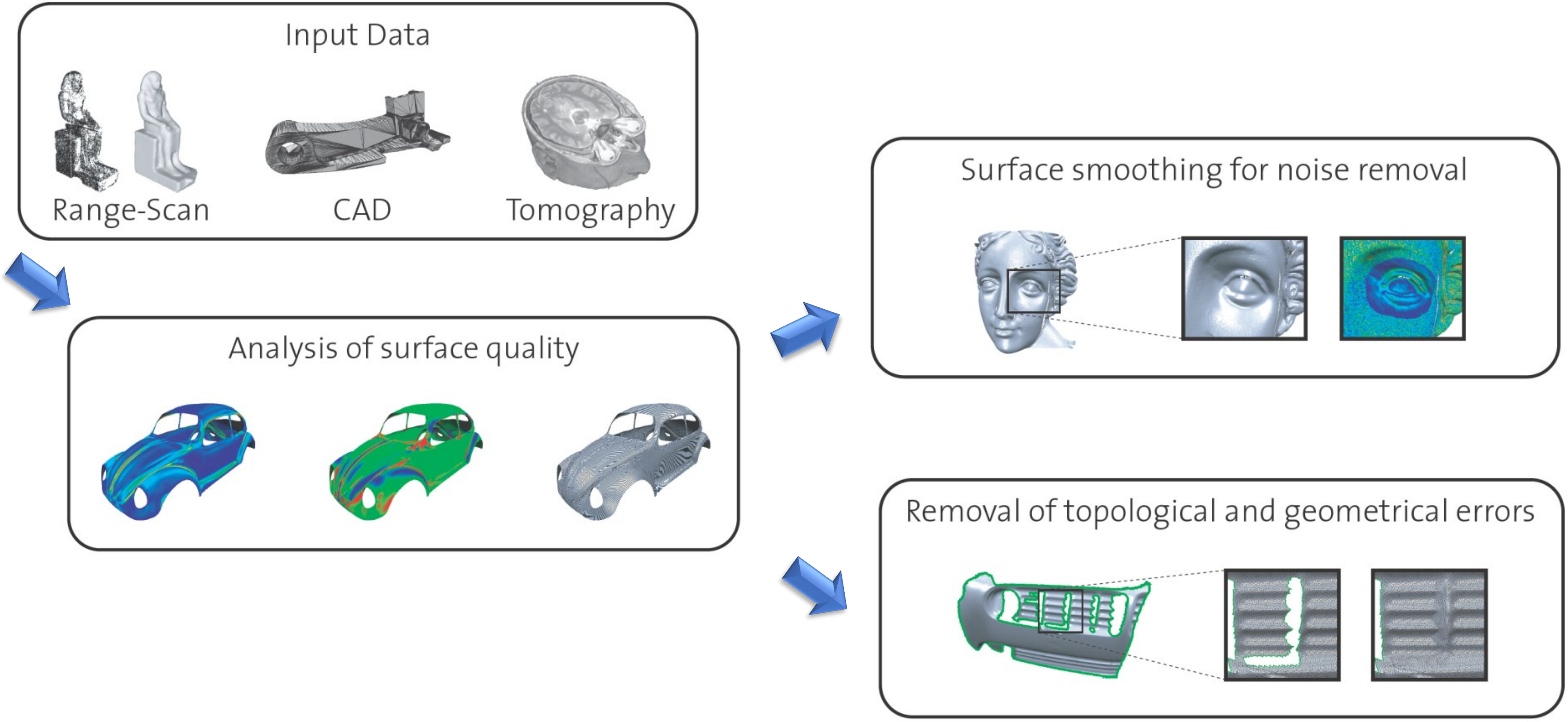
What is Geometry Processing About?

- Acquiring
- Analyzing/Repairing/Improving
- Manipulating

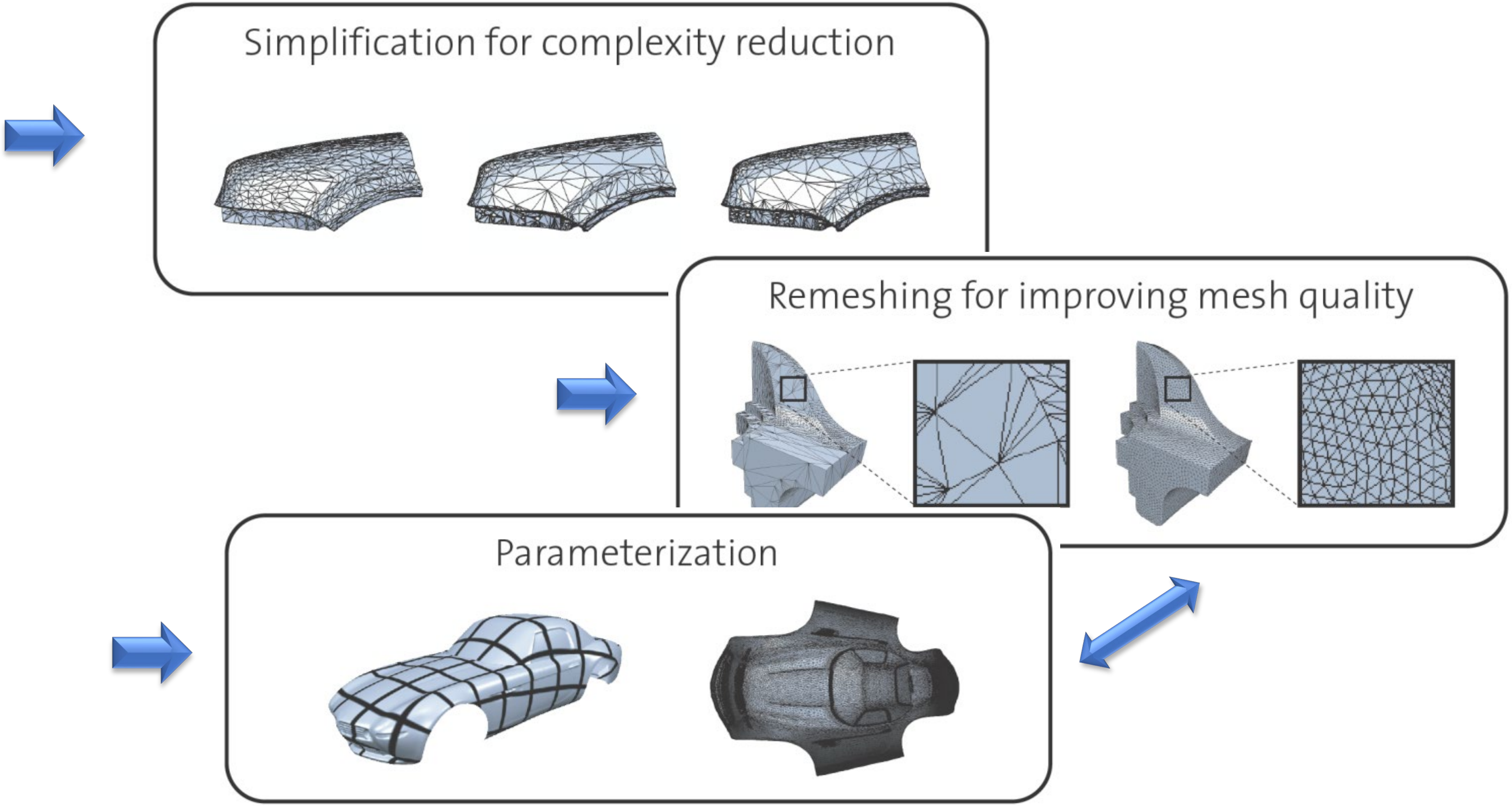


3D Models

A Geometry Processing Pipeline: Low Level Algorithms



A Geometry Processing Pipeline: Intermediate Algorithms

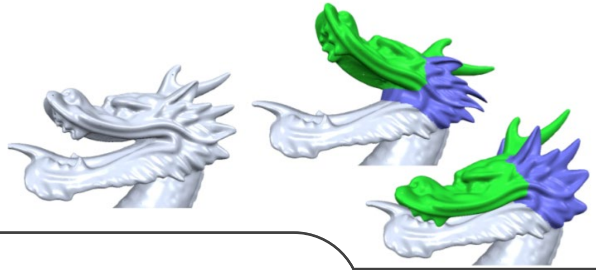


A Geometry Processing Pipeline: High Level Algorithms

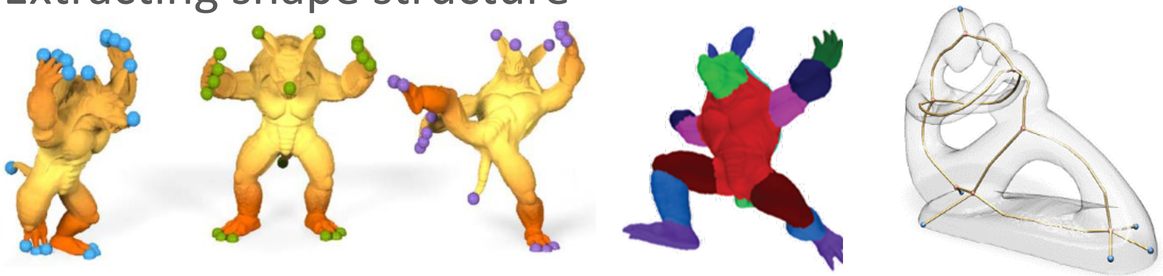
Freeform and multiresolution modeling

CAD model generation

Deformation and editing

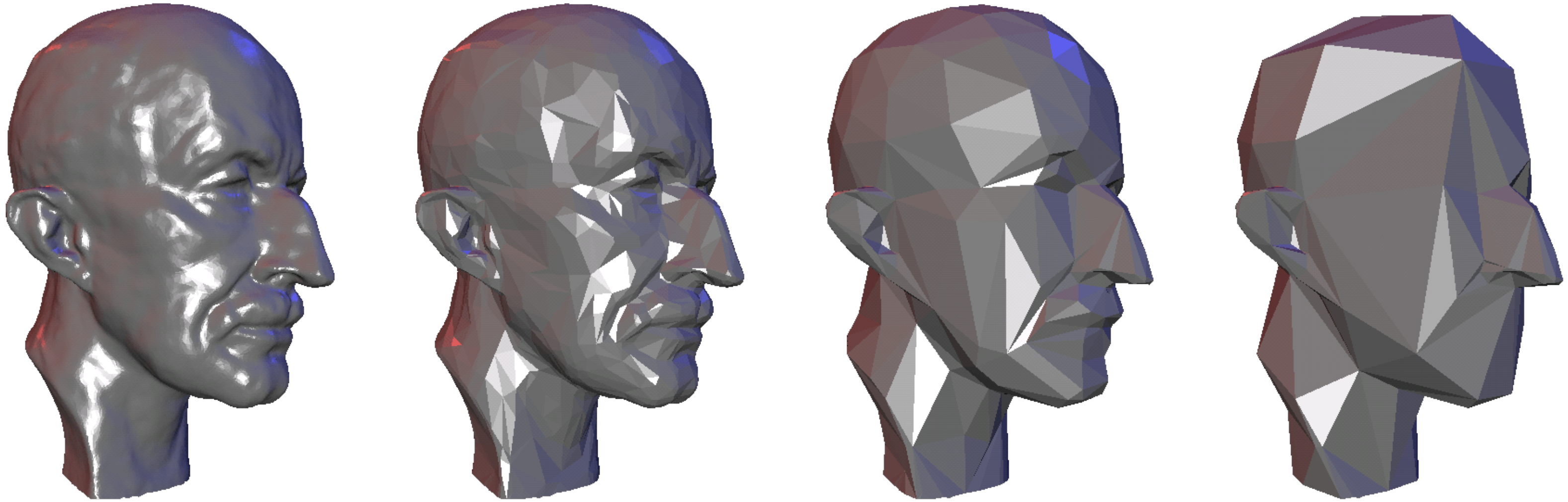


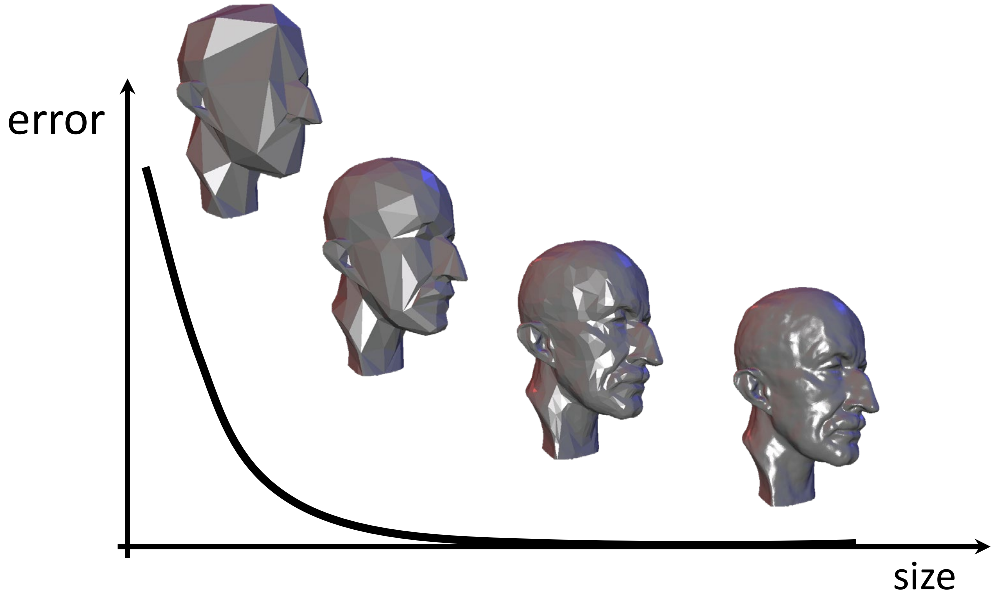
Extracting shape structure



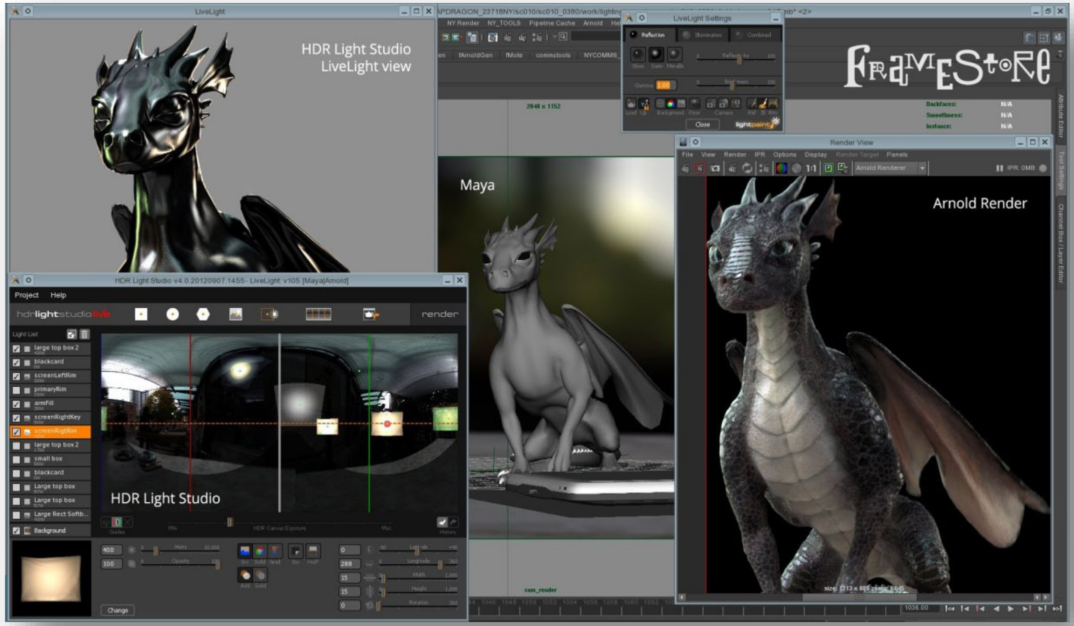
- keypoint extraction
- segmentation
- skeleton inference, more semantic representations

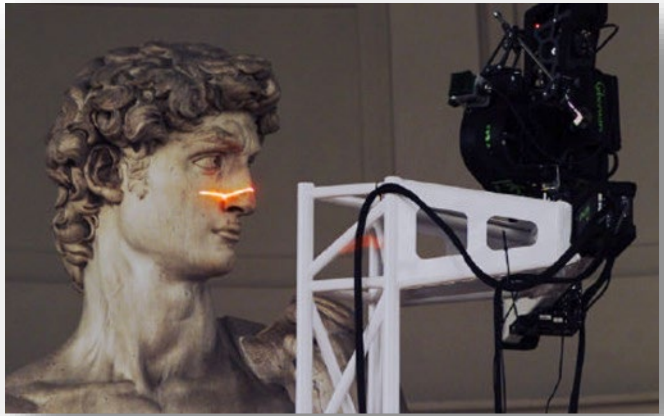
Always Trade-Offs for a Representation





3D Content Creation/Acquisition Is Hard





Autodesk Fusion 360

Autodesk Maya

All This Is Changing …
Better Software, Hardware, and Machine Learning

Democratization of 3D Content Creation

Simpler 3D Modeling Software

House in Istanbul Haluk Hatipoglu

SketchUp

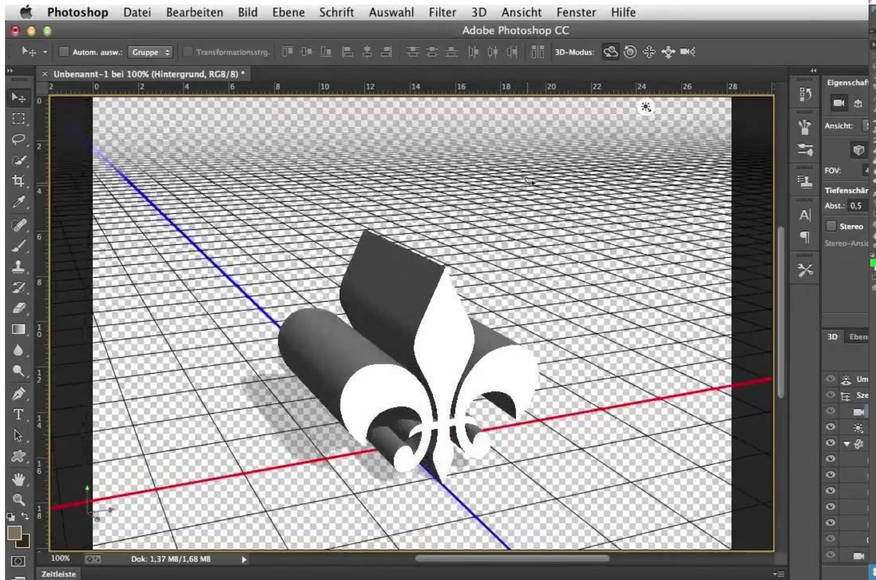
Trimble

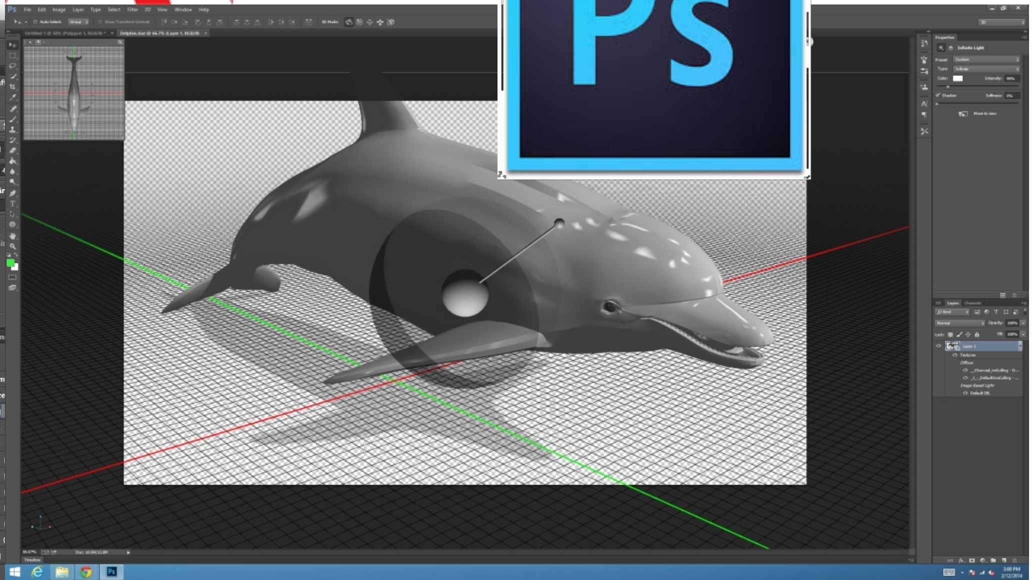
TINKERCAD

AUTODESK®
TINKERCAD®

Adobe

Ps





Affordable 3D Scanners

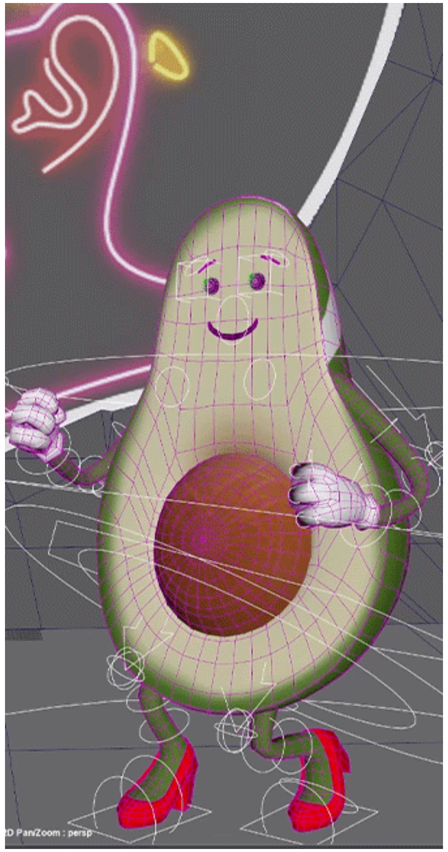
Microsoft Kinect

Google Tango

Intel RealSense

iSense 3D for iPad

SnapChat AR Lenses



Neural Radiance Fields (NeRFs)

Capturing the World in 3D from a Few Photos

The Course

http://cs348n.stanford.edu

Course Topics

January 3	January 5
Introduction. Traditional 3D modeling pipelines. Computer vision as inverse graphics. Neural 3D representations and neural rendering. Democratization of 3D content creation. Synthetic 3D data for ML training pipelines. Lecture Slides: Reading:	**Classical 3D Geometry Representations.** Low level: voxel grids, point clouds, triangular and quad meshes. High level: parametric and implicit boundary representations. Review of some classical geometric concepts (normals, curvature). Lecture Slides: Reading:
January 10	**January 12**
Neural Architectures for Regular Data. Brief review of deep nets and convolutional architectures for images. Sparse convolutions. Transformers. Voxel-based 3D methods for shapes and hierarchical variants. Lecture Slides: Reading:	**Irregular Geometries: Point Clouds.** PointNet and PointNet++. KPConv and other related methods. Sampling issues. Applications to object detection, classification, and segmentation.. Lecture Slides: Reading: PointNet, PointNet++, DGCNN, VoteNet

January 17	January 19
Martin Luther King, Jr., Day (holiday, no classes).	**Introduction to Generative Models (VAEs, deepSDF).** Autoencoders and autodecoders. Variational autoencoders. Deep signed distance functions. Lecture Slides: Reading: VarEncoders1, VarEncoders2, VQVAE, DeepMetafunctionals, IMNet, DeepSDF Student Presentation: KPConv, PointConv Homework 1 out.
January 24	**January 26**
Parametric Models. Generative Adversarial Networks (GANs) for 3D. Disentanglement. AtlasNet. HoloGAN, 3D-GAN, StyleGAN. Lecture Slides: Reading: GAN1, GAN2, AtlasNet, HoloGAN, WassersteinGAN, StyleGAN1 Student Presentation: StructuredImplicits, LocalImplicits	**3D Shape Public Data Sets. Flow and Auto-Regressive Models.** ShapeNet, PartNet, ... PointFlow, PolyGen. Lecture Slides: Reading: Student Presentation: ParSeNet, StyleGAN2, StyleGAN3 Homework 1 due. Homework 2 out.

Course Topics

January 31	February 2
Hierarchical Generation of Structure and Geometry. GRASS, StructureNet, ComplementMe. Lecture Slides: Reading: Student Presentation:	**Vector Graphics, Deep Architectures for Meshes.** Vector graphics generation, convolutions on meshes. MeshCNN. Lecture Slides: Reading: Student Presentation:
February 7	**February 9**
Pose Equivariance and Invariance in 3D Data. Vector neurons. Lecture Slides: Reading: Student Presentation:	**Conditional Generation: 3D Shape Completion.** Lecture Slides: Reading: Student Presentation: Homework 2 due. Homework 3 out.

February 14	February 16
Conditional Generation: From Image to Shape. Lecture Slides: Reading: Student Presentation:	**Learning Discrete and Continuous Shape Edits/Deformations. Shaping Latent Spaces.** Latent shape differences. Neural shape deformations/edits. Lecture Slides: Reading: Student Presentation: Project proposals due.
February 21	**February 23**
Presidents' Day (holiday, no classes).	**Neural Functions: 3D from 2D Supervision.** Neural rendering. Neural radiance fields (NeRFs). GRAF, GIRAFFE. Lecture Slides: Reading: Student Presentation: Homework 3 due.

Course Topics

February 28	March 2
Neural Fields and Surfaces. Surface and Neural field extraction. UNISURF, NeuS. Lecture Slides: Reading: Student Presentation:	**Scene Generation and Object Placement.** MetaSim, MetaSim2. Lecture Slides: Reading: Student Presentation:
March 7	**March 9**
Object/Scene Generation and Language. ShapeGlot, PartGlot. Lecture Slides: Reading: Student Presentation:	**Student Project Presentations.** Lecture Slides: Reading: Project due.

Course Requirements / Mechanics

- 3 programming assignments (1 week, 2 weeks, 2 weeks) using Google Cloud for Education
- 1 small project (BYI, but suggestions also provided, 3 weeks)
- 1 class presentation on research papers from the literature (topics covered in the previous class)
- teams of up to three students allowed

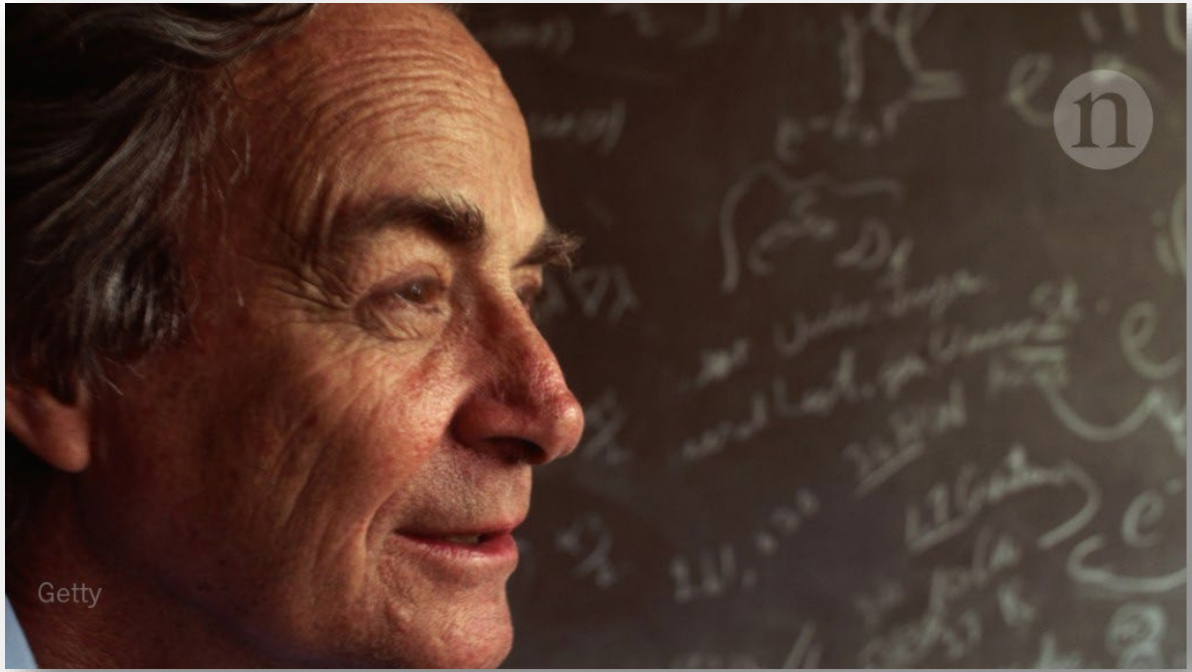
- we'll use Piazza (www.piazza.com) as the class discussion forum, and Gradescope (www.gradescope.com) for assignment submissions

Action Items

- Form collaboration teams, if you so desire (Piazza can help find partners)
- Negotiate the date of your literature paper presentation
- Start thinking about a project (we are here to provide feedback)

First Steps Towards 3D ML

Generative Modeling



Richard Feynman: "*What I cannot create, I do not understand*"

Generative modeling: "*What I understand, I can create*"

Generative Model (Unconditional)

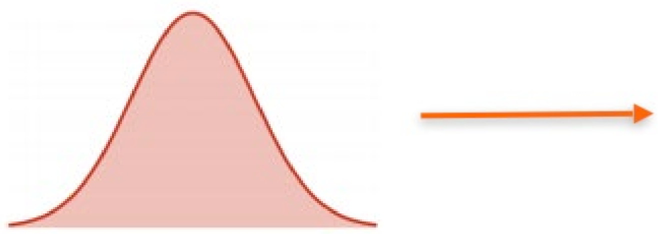
Given training data, generate new samples from the same distribution:

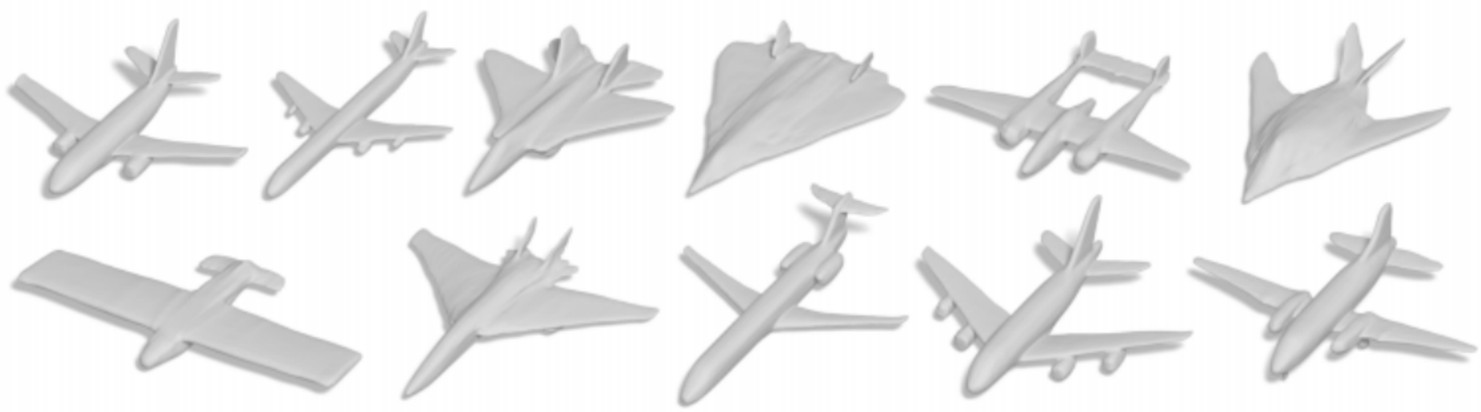
Training data ~ $p_{data}(x)$

Generated samples~ $p_{model}(x)$

Objective: learn a $p_{model}(x)$ that matches $p_{data}(x)$.

Decoding/Generation





Latent vectors *z*

Generated Shapes

Generator/Decoder: generating shapes from latent vectors

How to Learn Generative Models

- Explicitly modeling data probabilistic density, learn a network $p_\theta(x)$ that maximize data probability

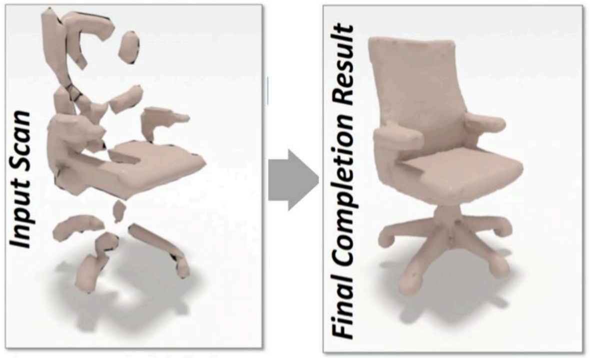
- Implicitly modeling probabilistic density, e.g. learn a network that scores how "real" the generated data is, $f_\theta(x)$

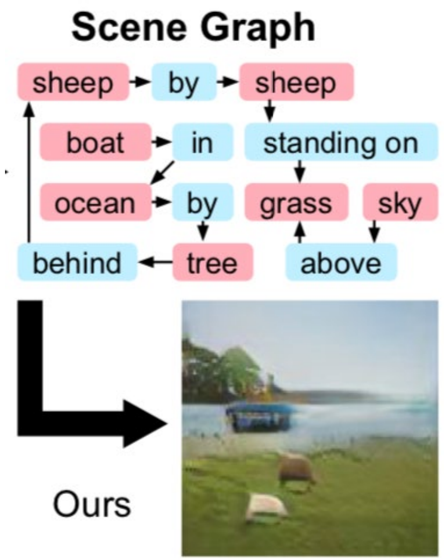
- Markov chain
- Autoregressive models
- Variational autoencoder (VAE)
- Flow-based models
- Structure-based models
- Energy based models
- …

- Generative adversarial network (GAN)
- Score-based generative
- …

Generative Model (Conditional)

- Data: (x, y) where x is a **condition** and y is the corresponding **content.**





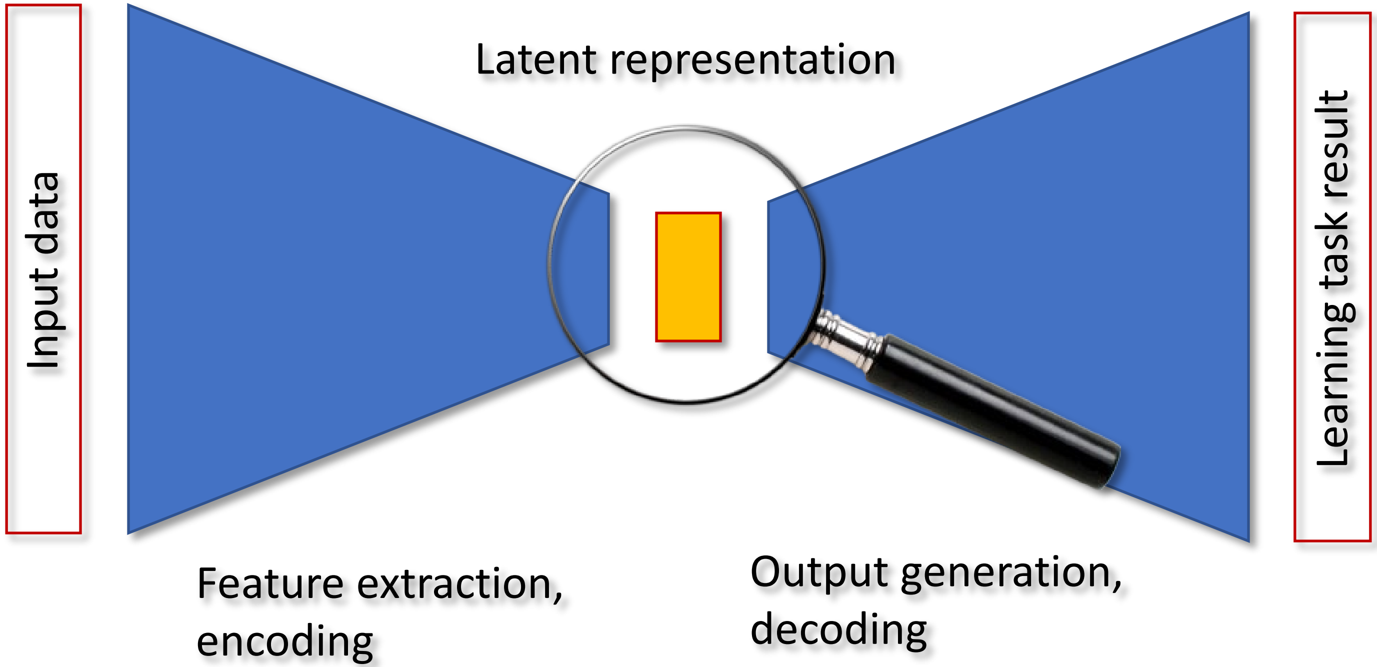
Single-view 3D reconstruction

Shape completion

Image generation based on scene-graph

Objective: learn a $p_{model}(y|x)$ that matches $p_{data}(y|x)$.

Latent Spaces in Deep Learning



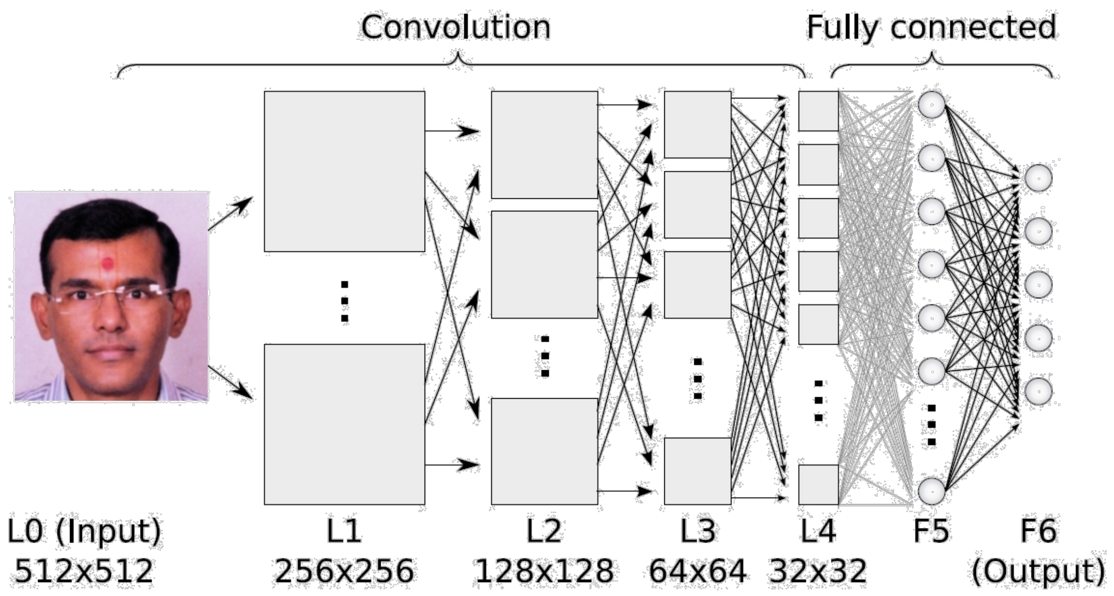
Discriminative tasks, conditional generative tasks

A latent code acts as a low-d proxy for input data w.r.t. a learning task

3D Representations and Learning Frameworks

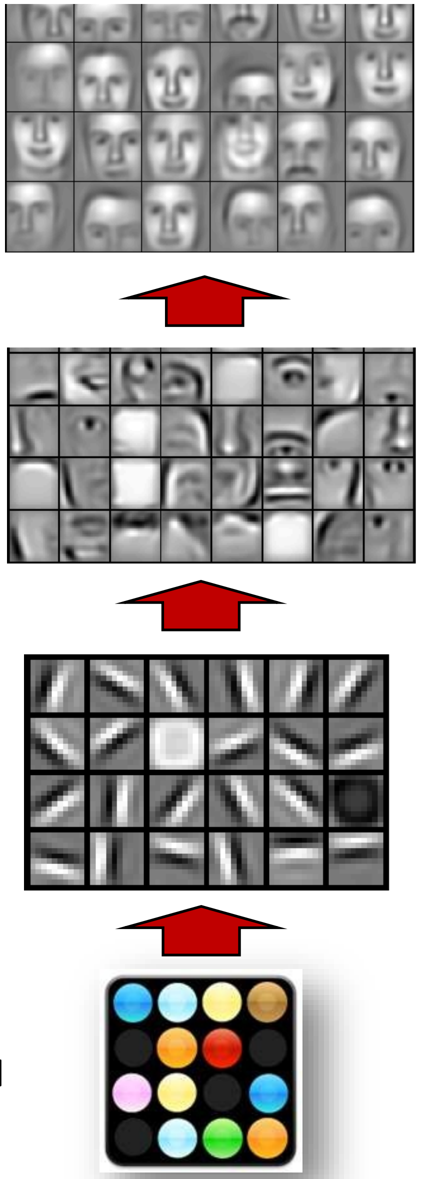
Encoding

Convolutional Image Networks



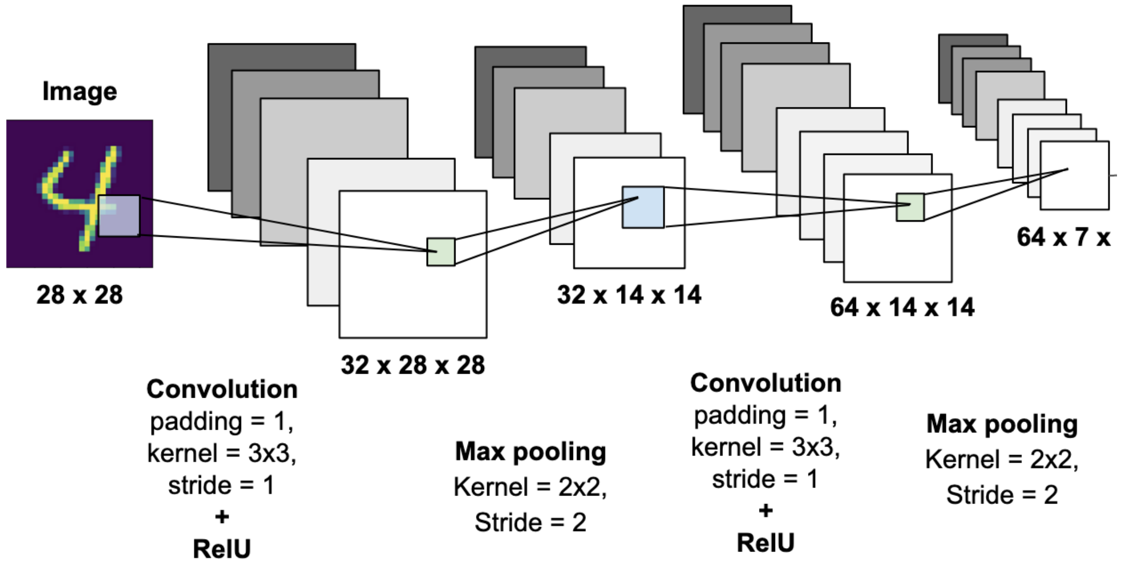
[Makwana, 2016]

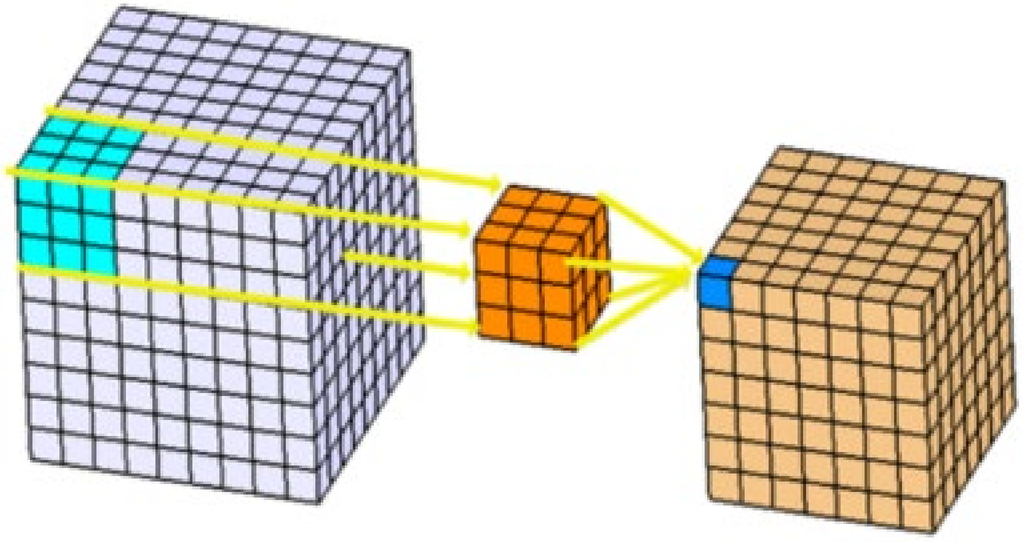
Encoder



[Lee et al., 2009]

From 2D to 3D Convolutions: Pixels to Voxels





2D convolution

Kernel: K_h X K_w

Kernel weight: K_h X K_w X C_1 X C_2

Feature grid: H X W X C

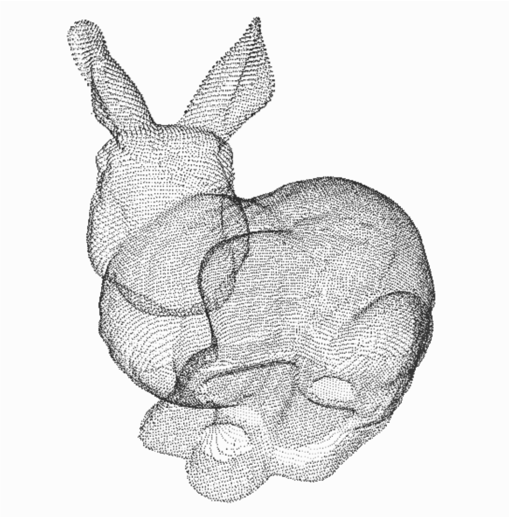
3D convolution

Kernel: K_h X K_w X K_d

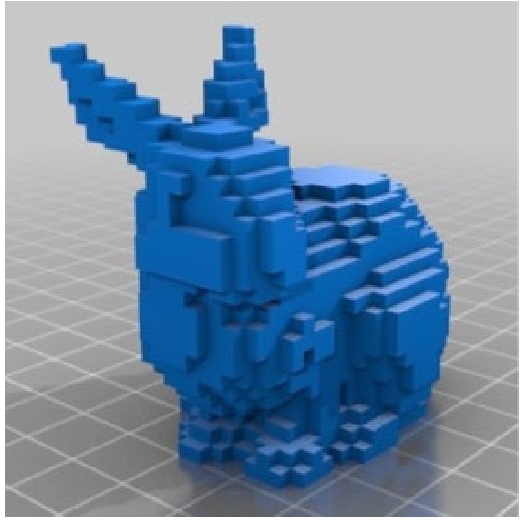
Kernel weight: K_h X K_w X K_d X C_1 X C_2

Feature grid: H X W X D X C

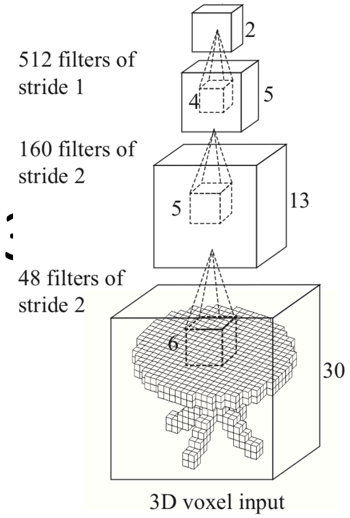
Voxelized 3D Convolution



voxelization →



+



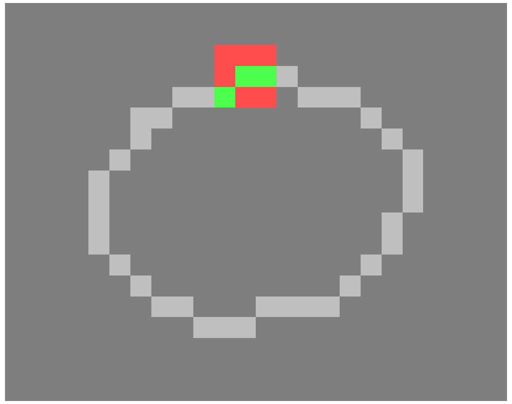
[Wu et al. 2015]

Con: High space complexity -- 3D convolution $O(N^3)$

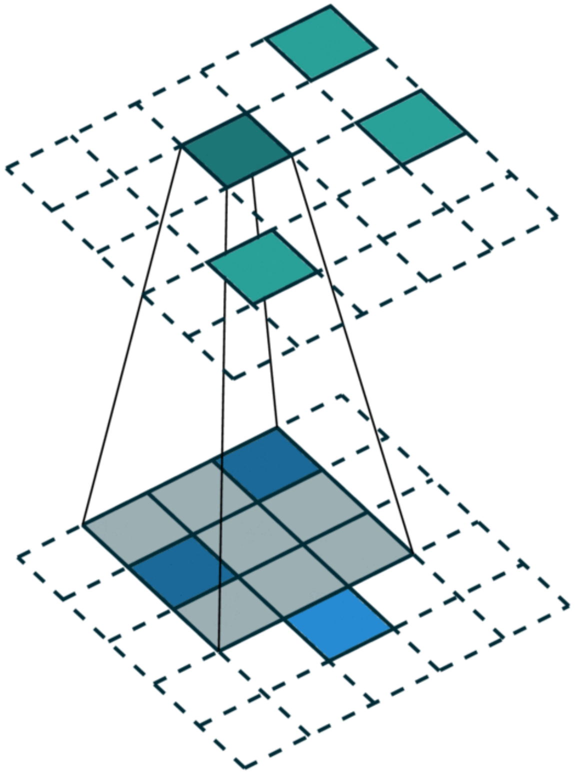
Quantization errors in voxelization

Not very attractive for generative models

Efficiency: Sparse Convolution



Submanifold sparse convolutional network (from FAIR)

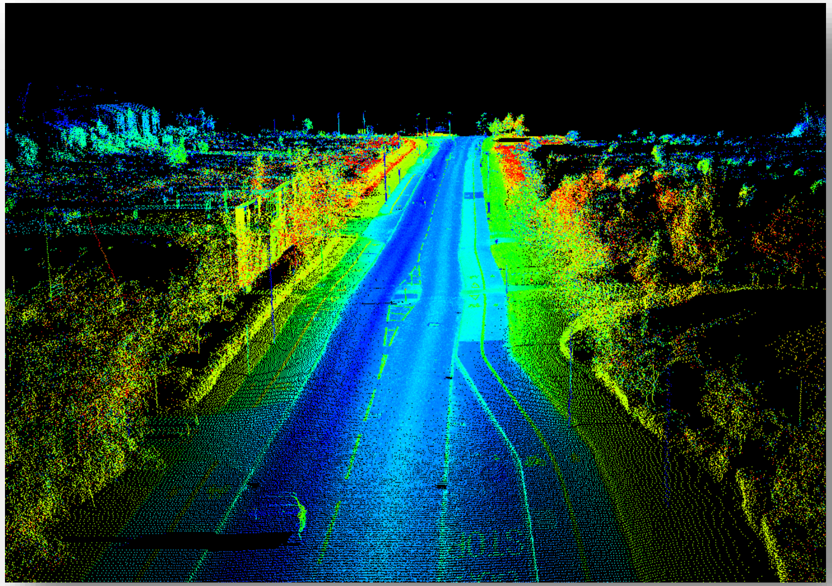


Minkowski Engine (from SVL)

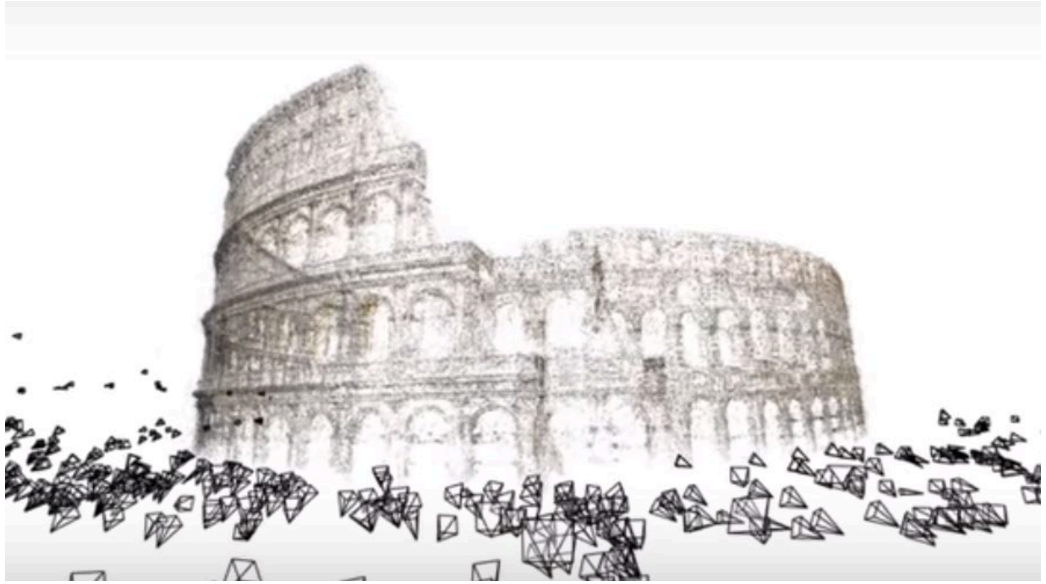
Pro: efficient computation

Con: quantization remains

Point Clouds from Many Sensors



Lidar point clouds (LizardTech)

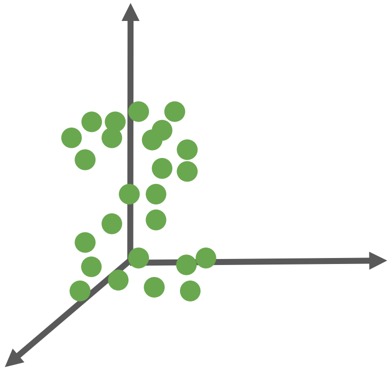


Structure from motion (Microsoft)

Depth camera (Intel)

But irregular!

PointNet++: Convolutions on Point Clouds



PointNet/
PointNet++

Object Classification

Object Part Segmentation

Semantic Scene Parsing

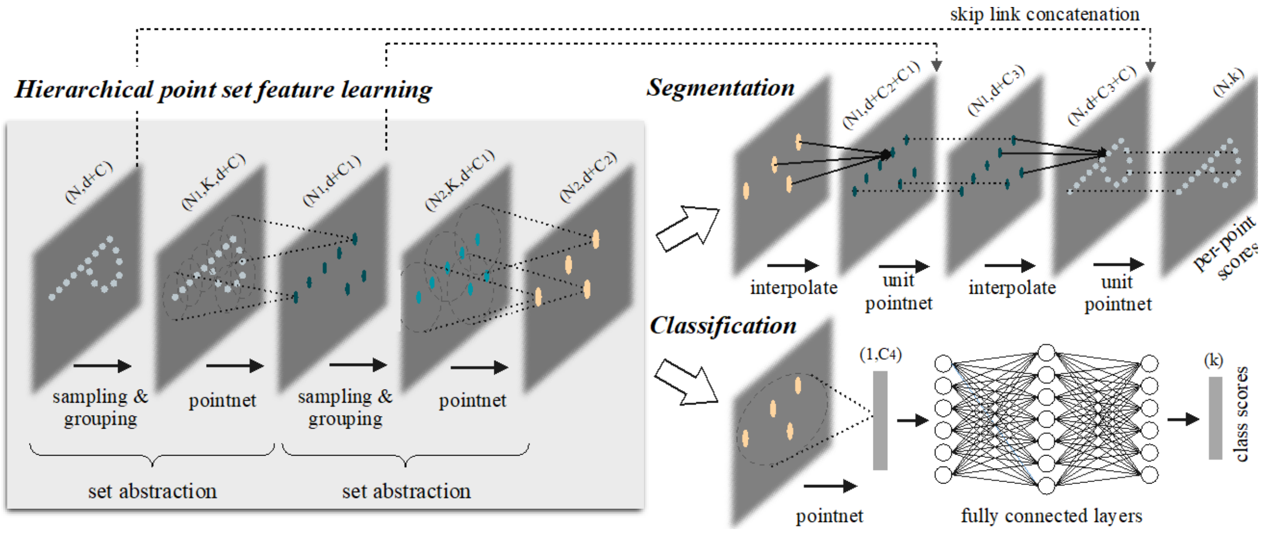
…

End-to-end learning for irregular point data

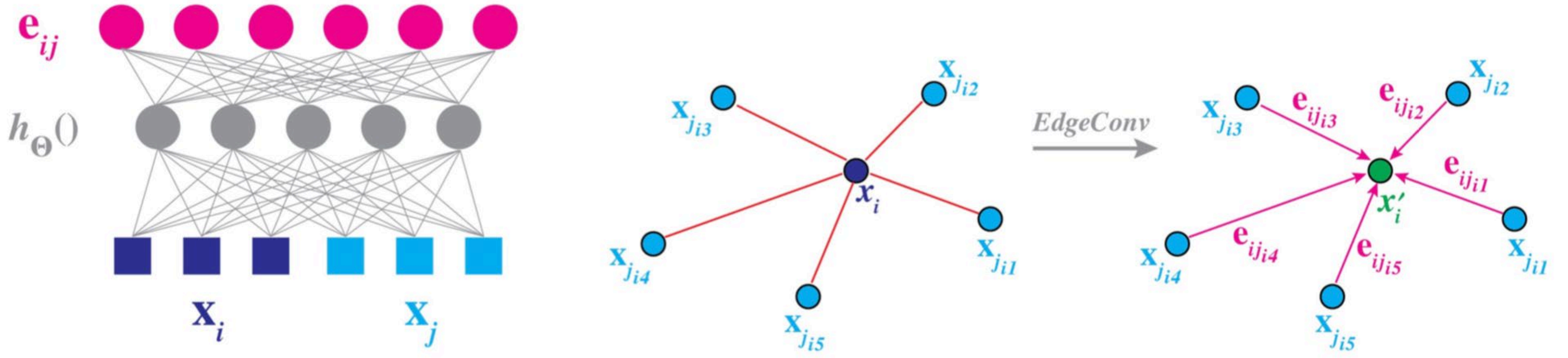
Unified framework for various tasks

Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation (CVPR'17)

Charles R. Qi, Li Yi, Hao Su, Leonidas Guibas. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space* (NeurIPS 2017)



Convolution on Mesh/Graph



Message passing: The output of EdgeConv at the i-th vertex is thus given by

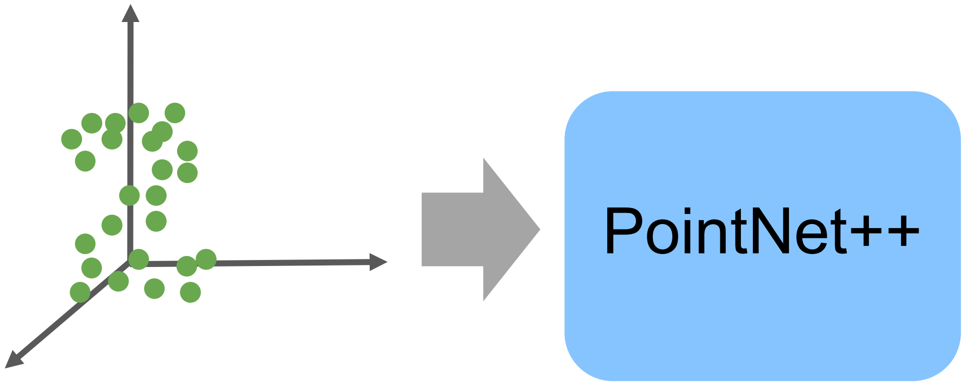
$$\mathbf{x}'_i = \square_{j:(i,j)\in\mathcal{E}} h_\Theta(\mathbf{x}_i, \mathbf{x}_j). \tag{1}$$

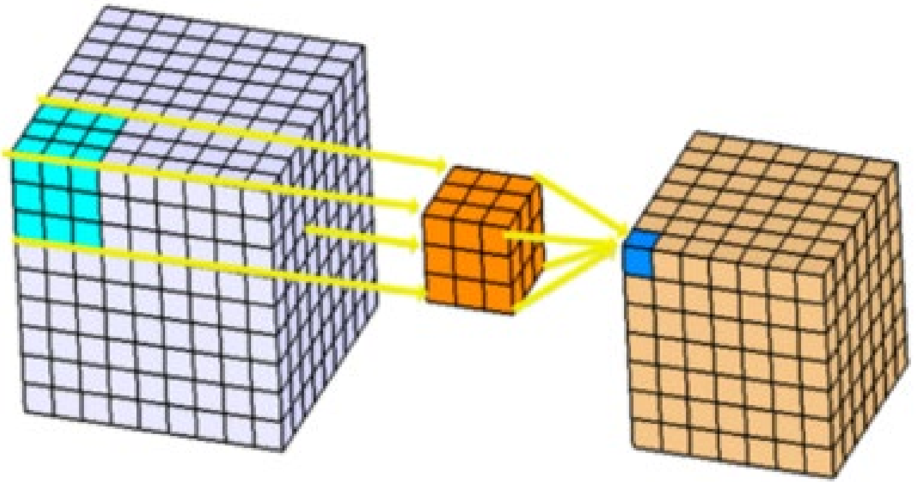
Wang, et.al., Dynamic Graph CNN for Learning on Point Clouds, ToG 2019

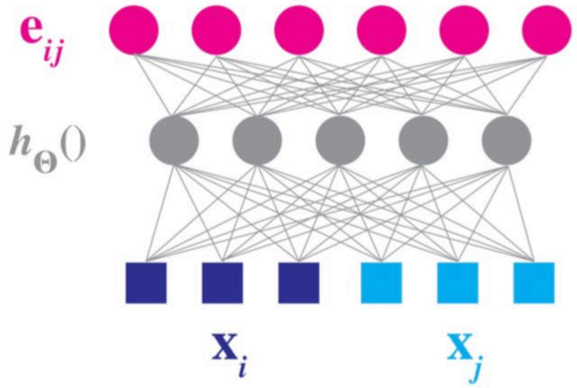
3D Representations and Learning Frameworks

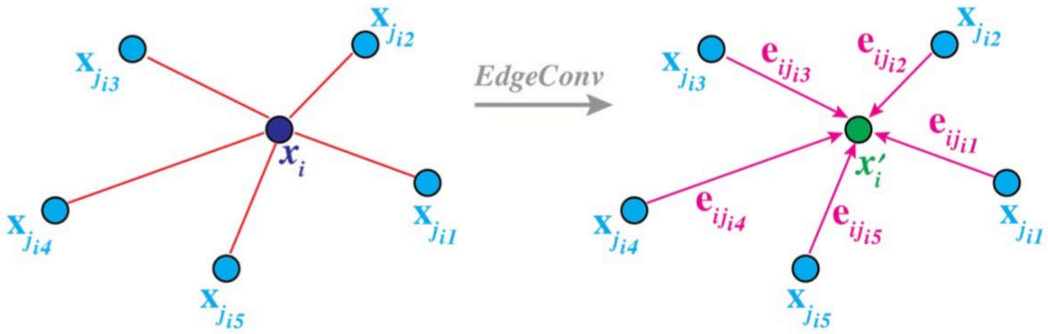
Decoding

Encoding 3D Using Convolution



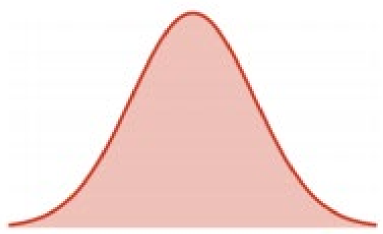


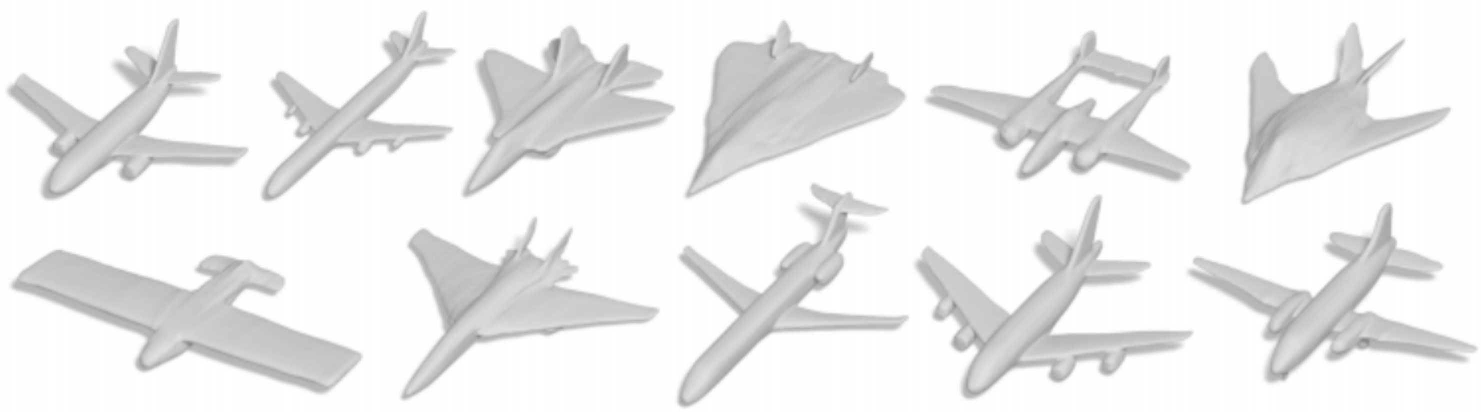




Encoding: Convolution networks can transform a 3D data into a vector in latent space.

Decoding/Generation





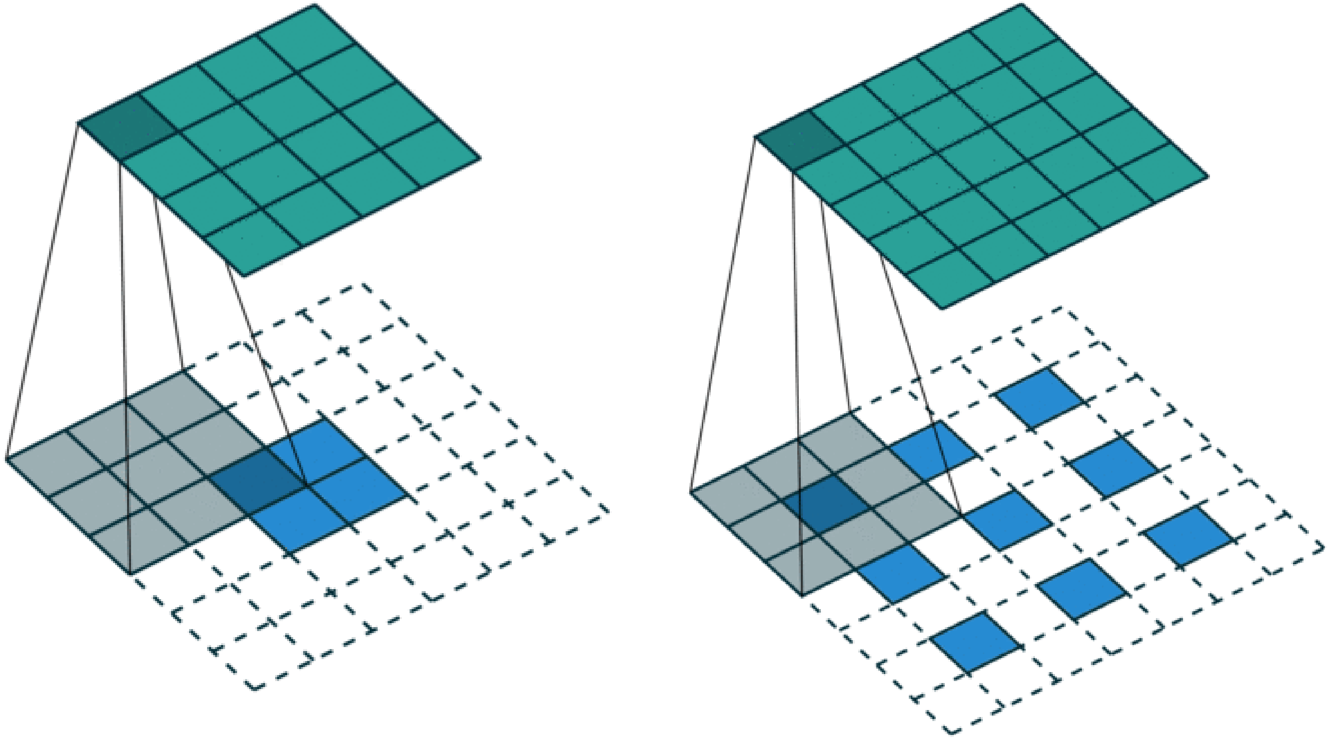
Latent vectors *z*

Generated Shapes

Generator/Decoder: generating shapes from latent vectors

Upsampling and Deconvolution (Transposed Conv)

Stride = 1,
Padding = 0



Stride = 2,
Padding = 1

•**Padding *(p):*** The number of zeros padded around the original input increasing the size to ***(i+2*p)x(i+2*p)***
•**Stride *(s):*** The amount by which the kernel is shifted when sliding across the input image.

Image credit:
https://github.com/vdumoulin/conv_arithmetic

Decoders Really Matter for Generative Models

Springer Link

Published: 14 March 2019

The Devil is in the Decoder: Classification, Regression and GANs

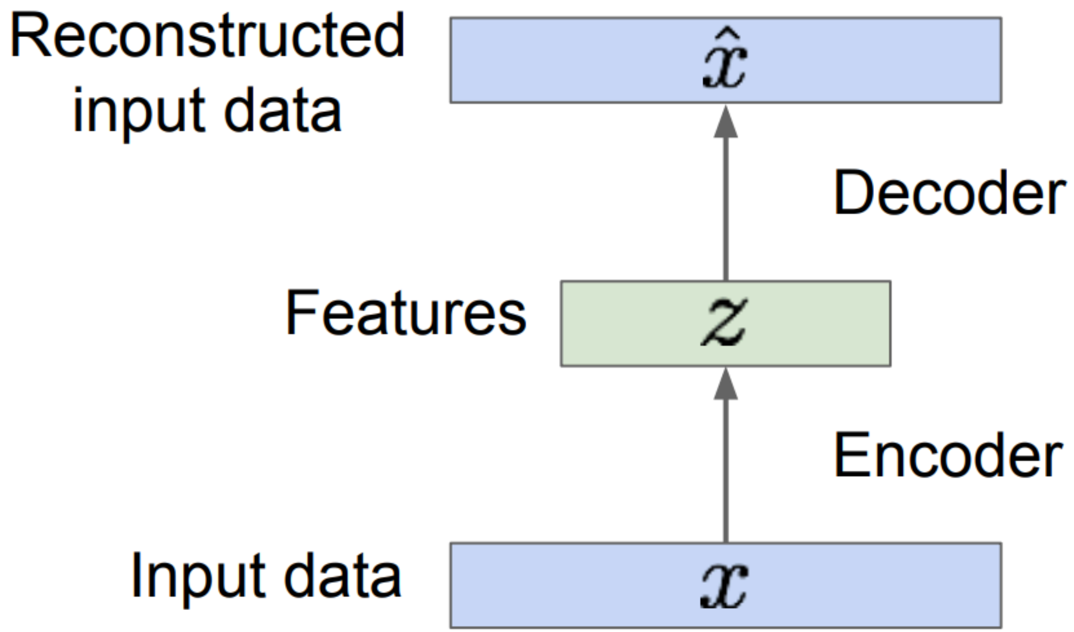
Zbigniew Wojna, Vittorio Ferrari, Sergio Guadarrama, Nathan Silberman, Liang-Chieh Chen, Alireza Fathi & Jasper Uijlings

International Journal of Computer Vision **127**, 1694–1706 (2019) | Cite this article

Abstract

Many machine vision applications, such as semantic segmentation and depth prediction, require predictions for every pixel of the input image. Models for such problems usually consist of encoders which decrease spatial resolution while learning a high-dimensional representation, followed by decoders who recover the original input resolution and result in low-dimensional predictions. While encoders have been studied rigorously, relatively few studies address the decoder side. This paper presents an extensive comparison of a variety of decoders for a variety of pixel-wise tasks ranging from classification, regression to synthesis.

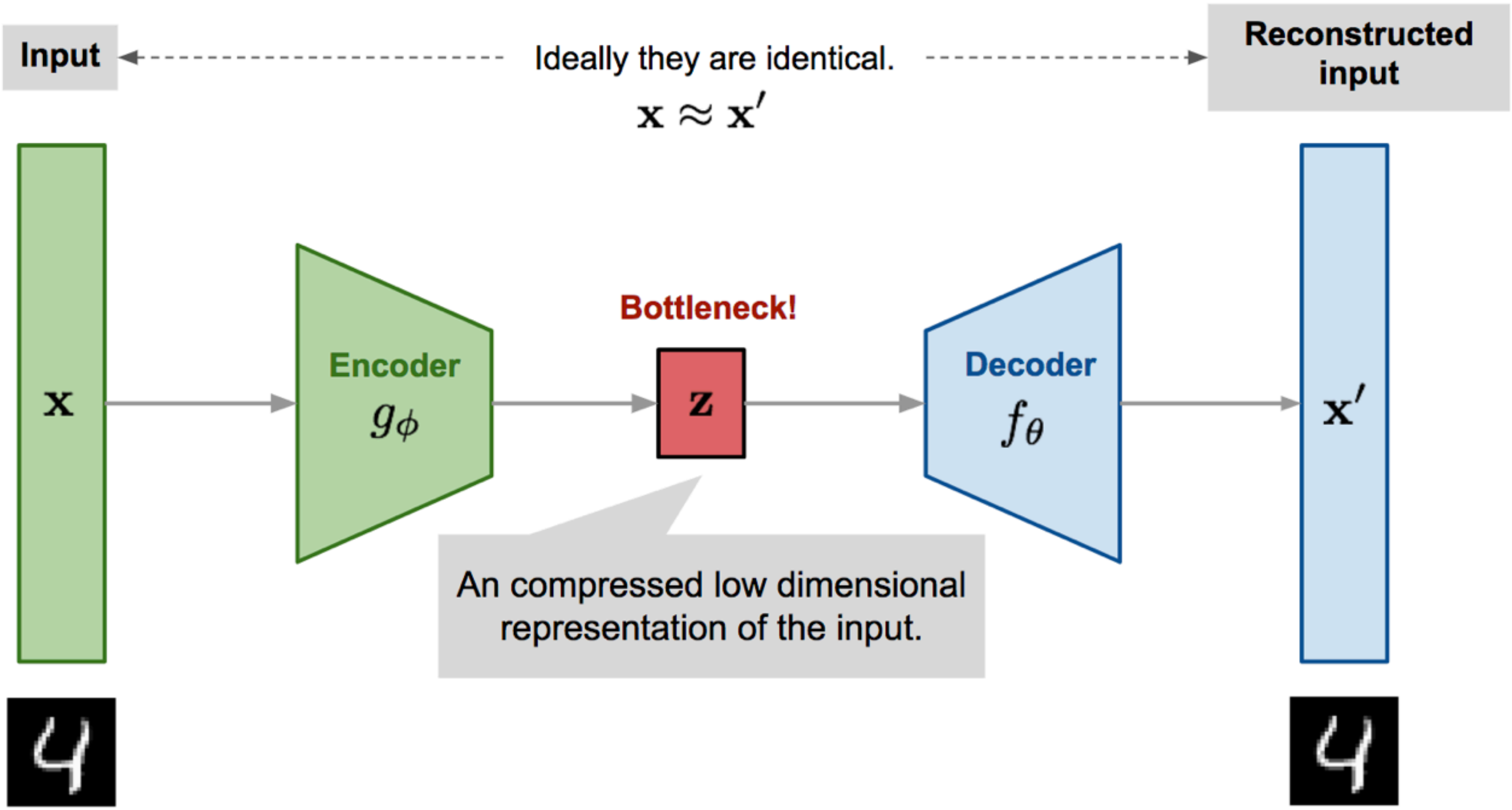
Auto-Encoder



- AE encodes itself into a latent z
- AE then decodes the latent z back to itself
- Understanding AE is the first step to understand generative models.

Image Credit: Stanford CS231N

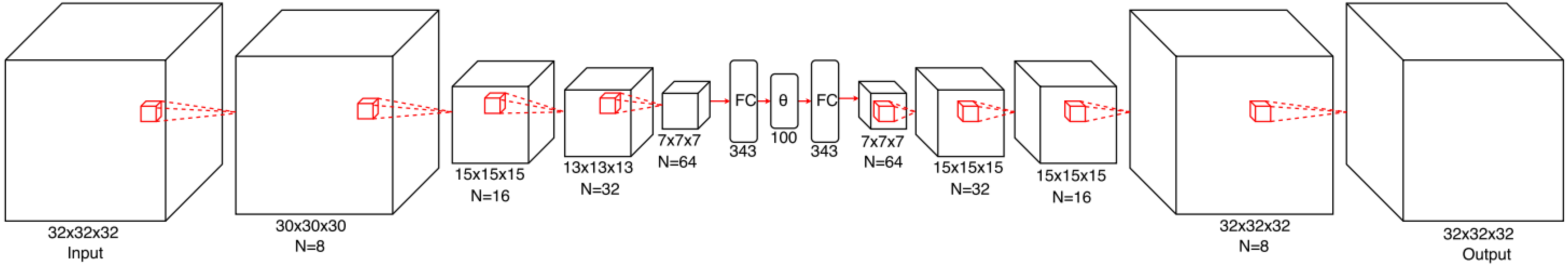
Auto-Encoder Training (Self-Supervised)



Task: Learn to encode the input and decode itself

Reconstruction loss: measuring the distance between the input/output

Volumetric AE



Binary Cross-Entropy Loss: $\mathcal{L} = -t\,log(o) \; - \; (1-t)\,log(1-o)$

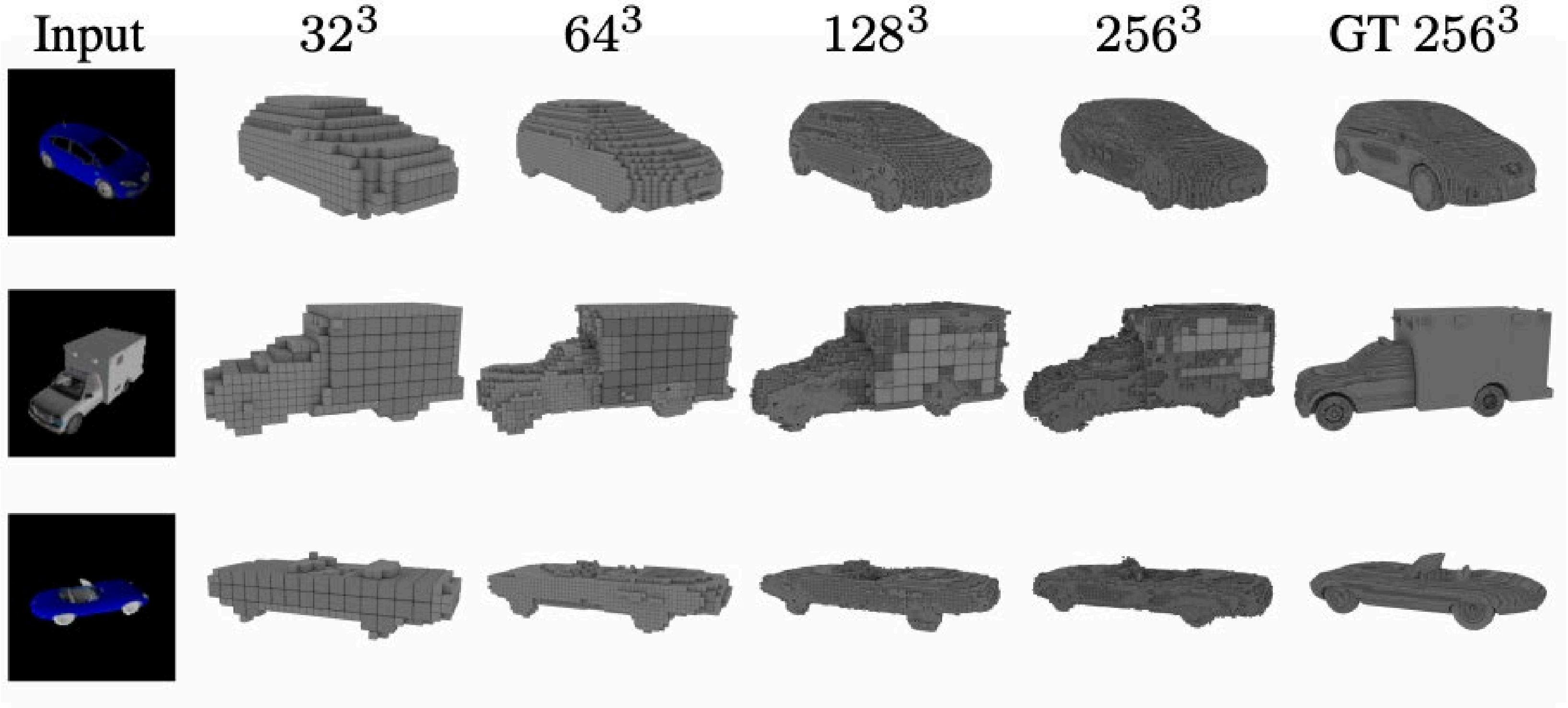
CoRR 2016

Generative and Discriminative Voxel Modeling with Convolutional Neural Networks

Auto-Encoder Connecting 2D and 3D

Encoder: 2D Conv

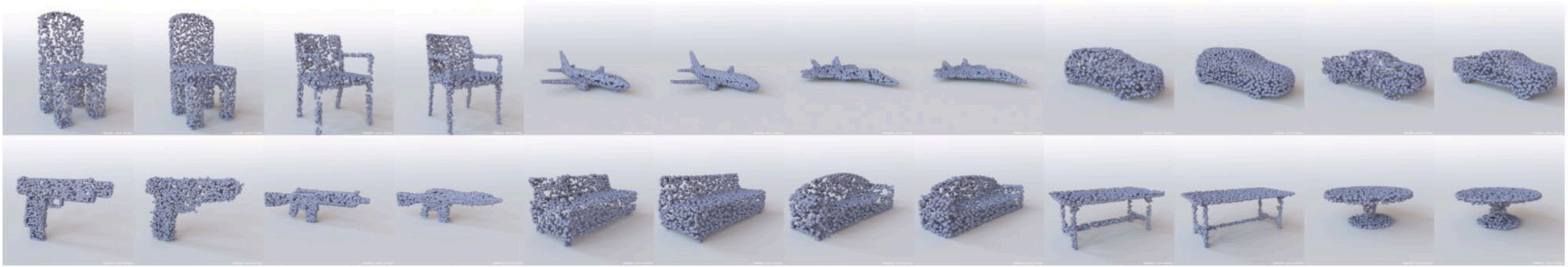
Decoder: 3D Deconv (Octree decoder)



Tatarchenko et al., **"Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs"**, *ICCV 2017*

Point Cloud AE

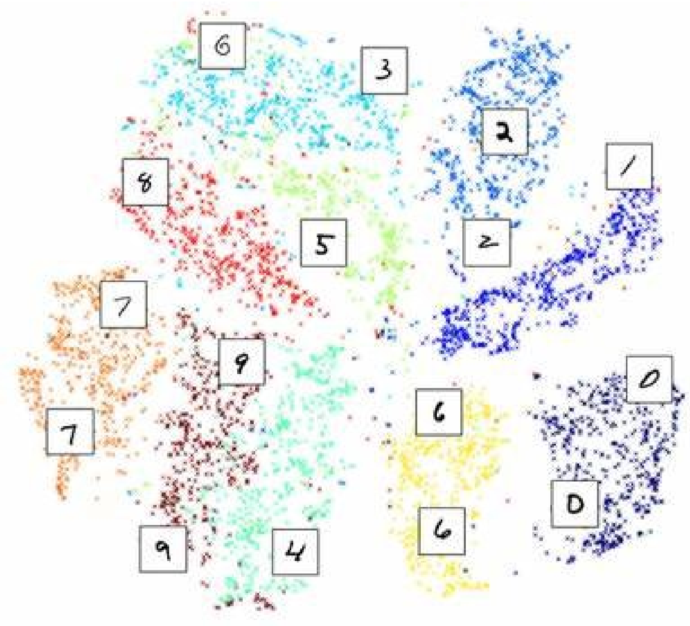
Encoder: PointNet (N*3 → L)
Decoder: MLP (L → 3N → N*3)

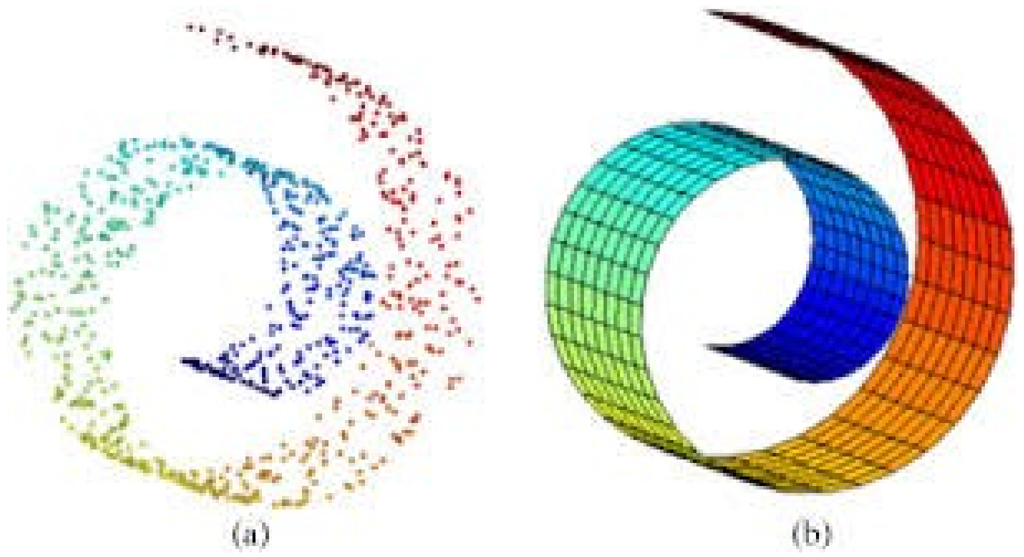


ICML, 2018, Learning Representations and Generative Models for 3D Point Clouds, Panos Achlioptas, et. al.

Auto-encoder Latent Space

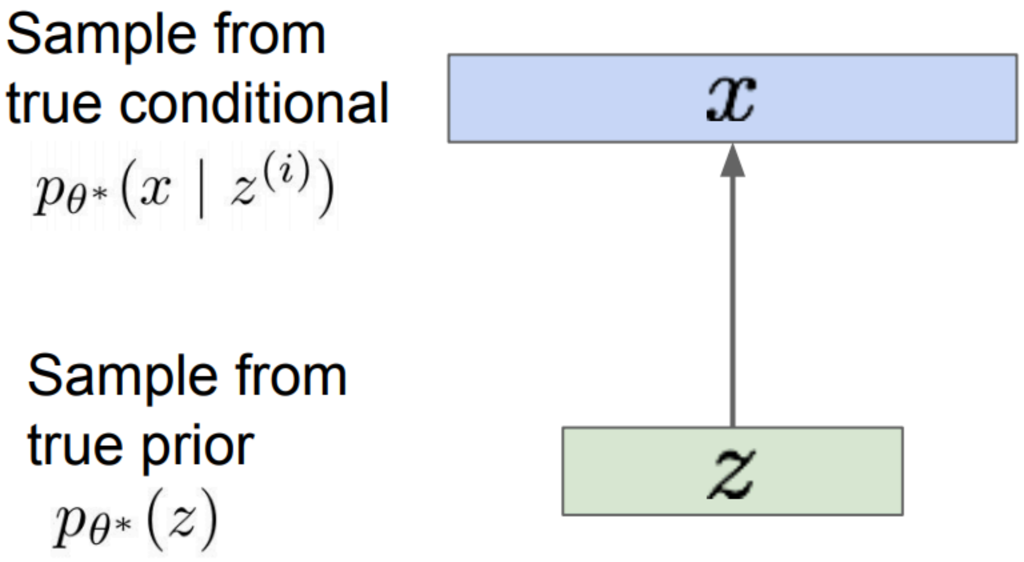
- Where is the data manifold in the latent space?
- Is a vanilla autoencoder a generative model?





Need to “structure” the latent space

Variational Auto-Encoder

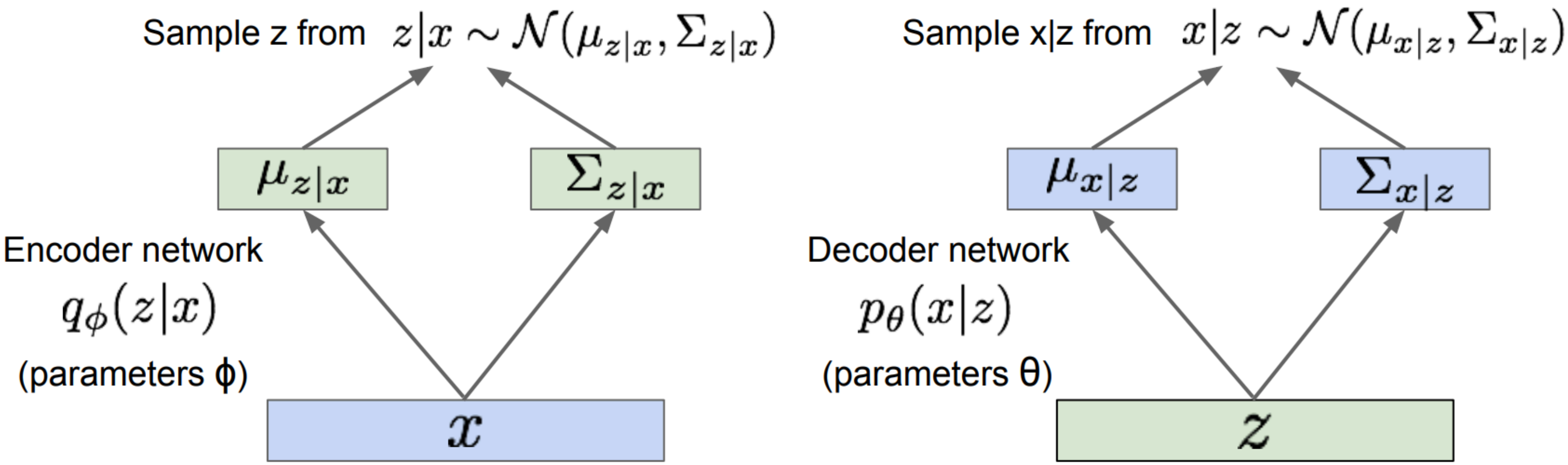


We can assume z follows a distribution.

Choose prior p(z) to be simple, e.g. Gaussian.

Image Credit: Stanford CS231N

Variational Auto-Encoder

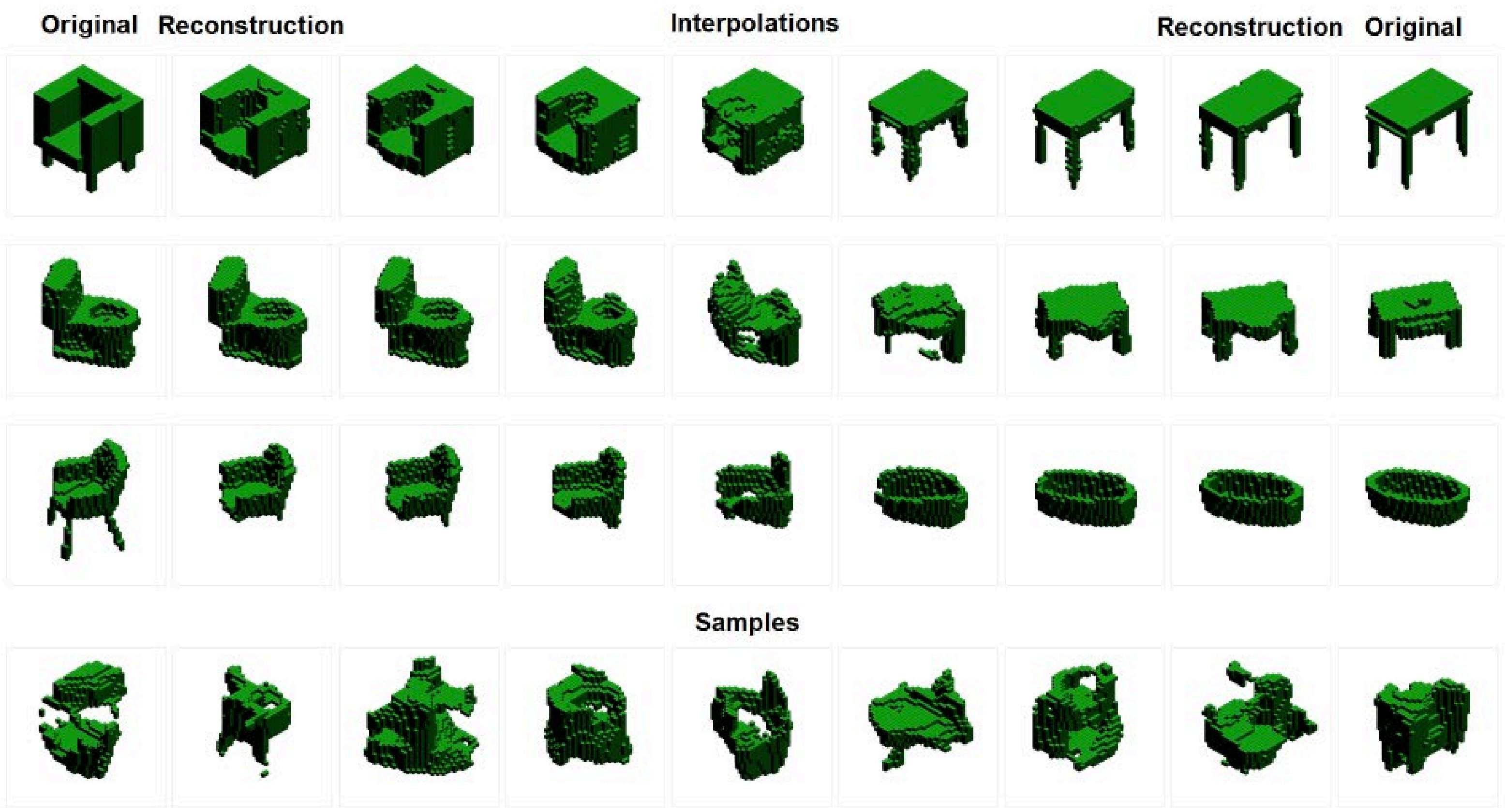


Encoder

Decoder

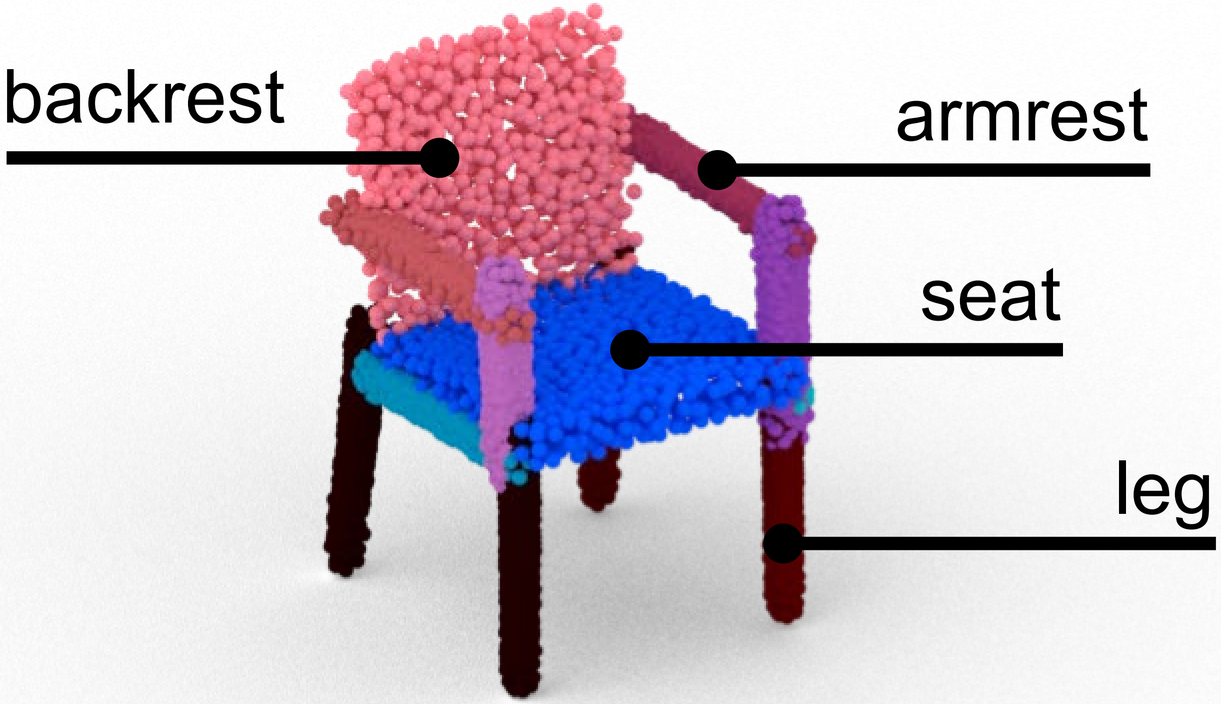
Image Credit: Stanford CS231N

Generating New Samples & Interpolation



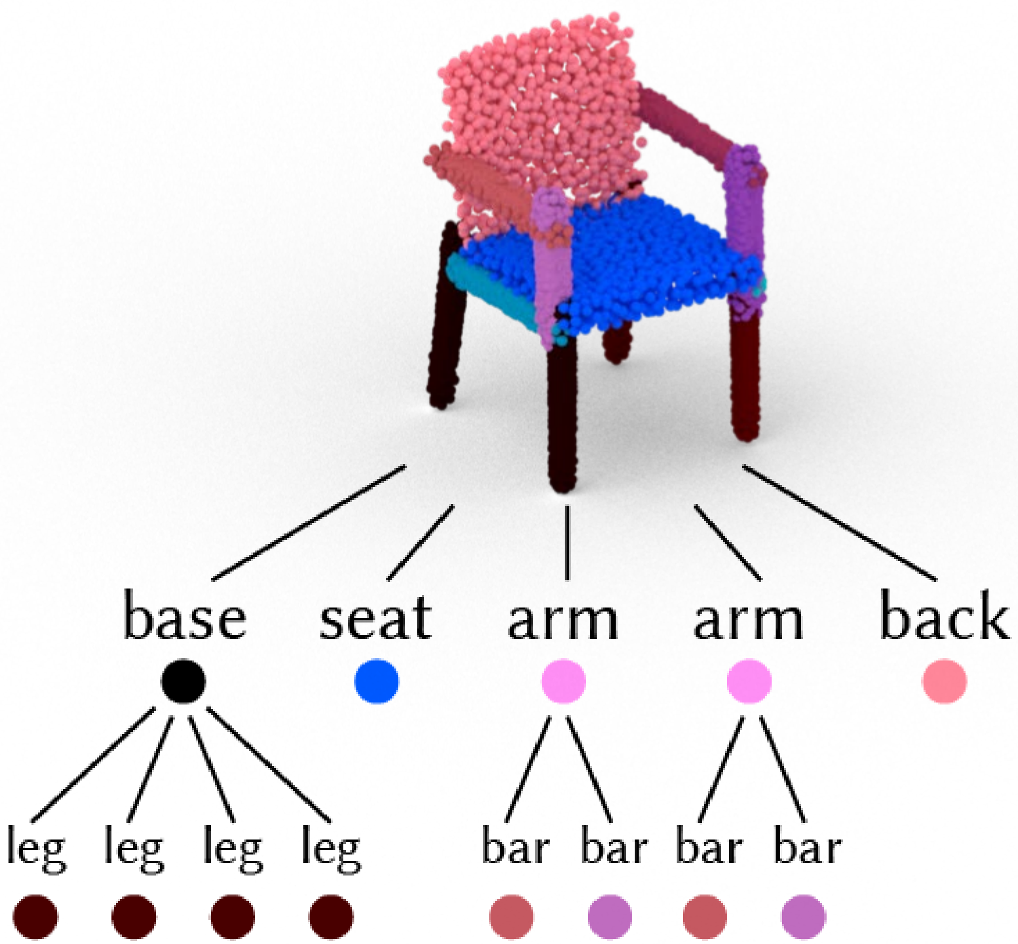
CoRR 2016
Generative and Discriminative Voxel Modeling with Convolutional Neural Networks

Geometry and Structure

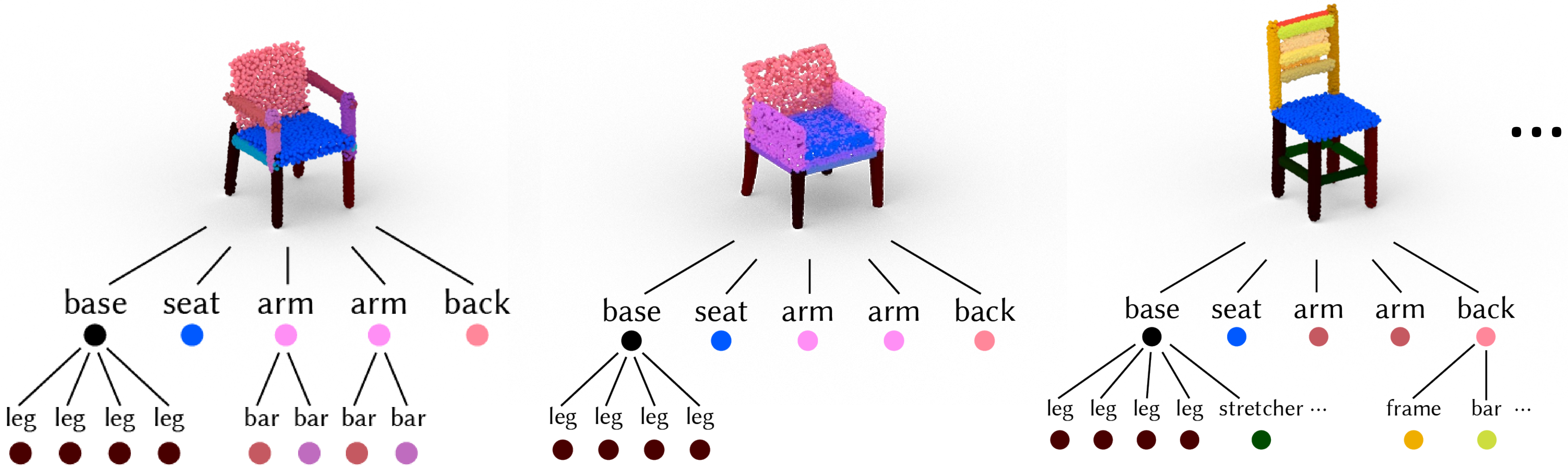


Now with point clouds for the geometry

Structure: Part Hierarchy

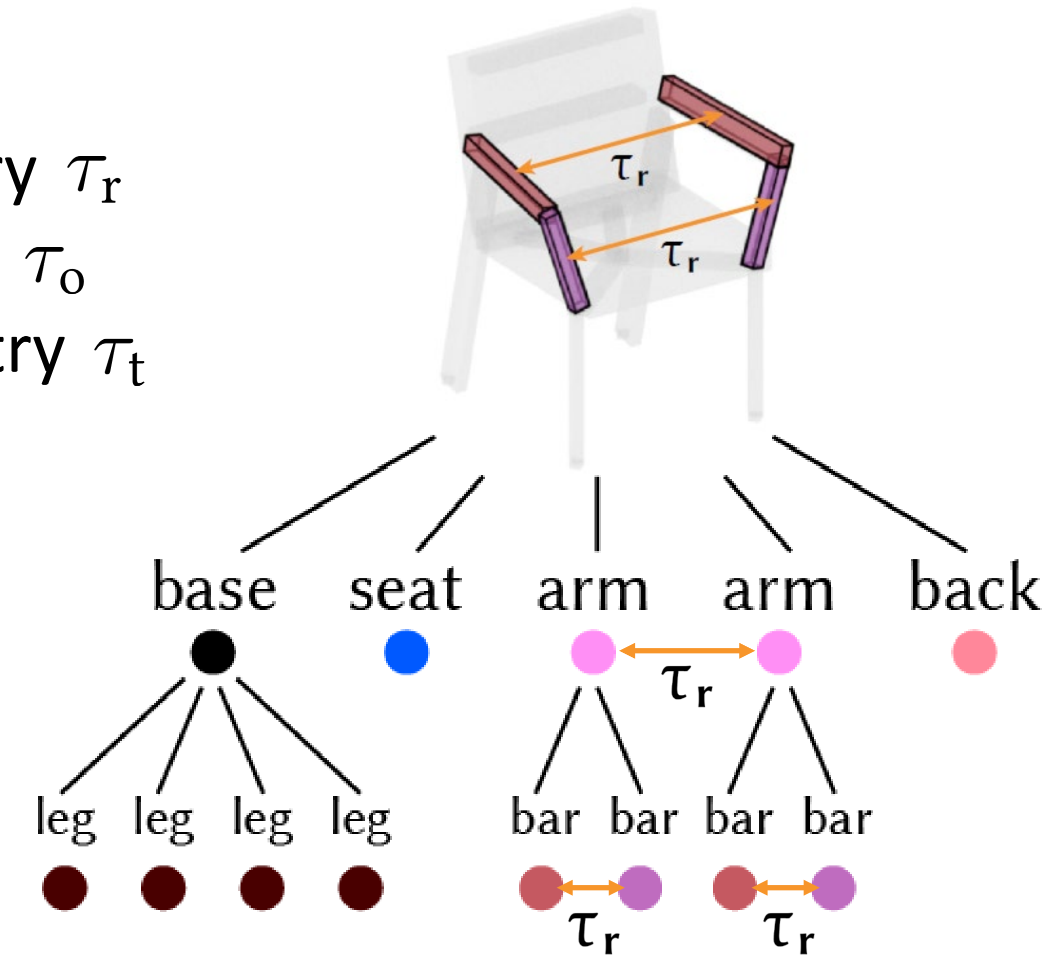


Structural Consistency

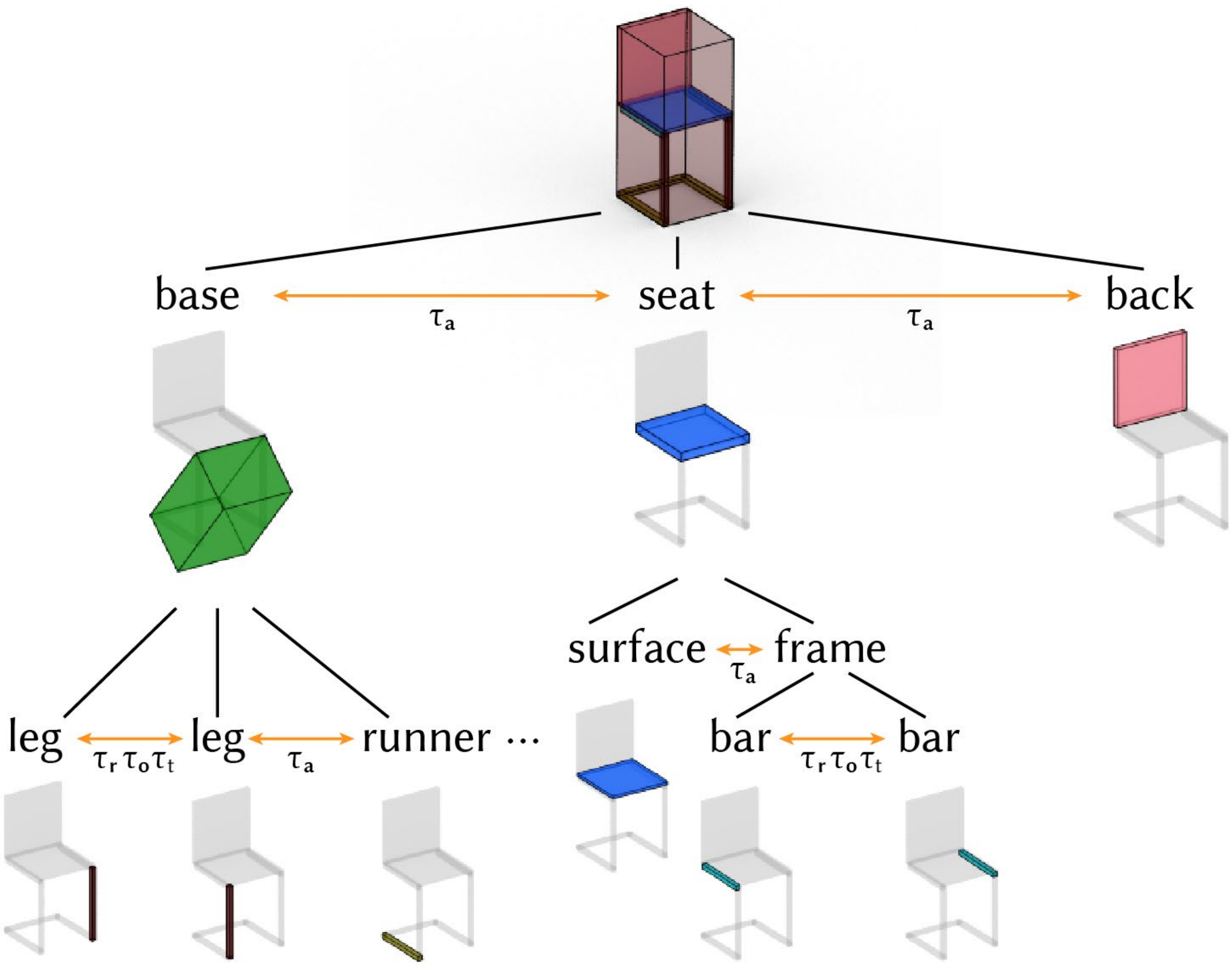


Object Representation: Sibling Relationships

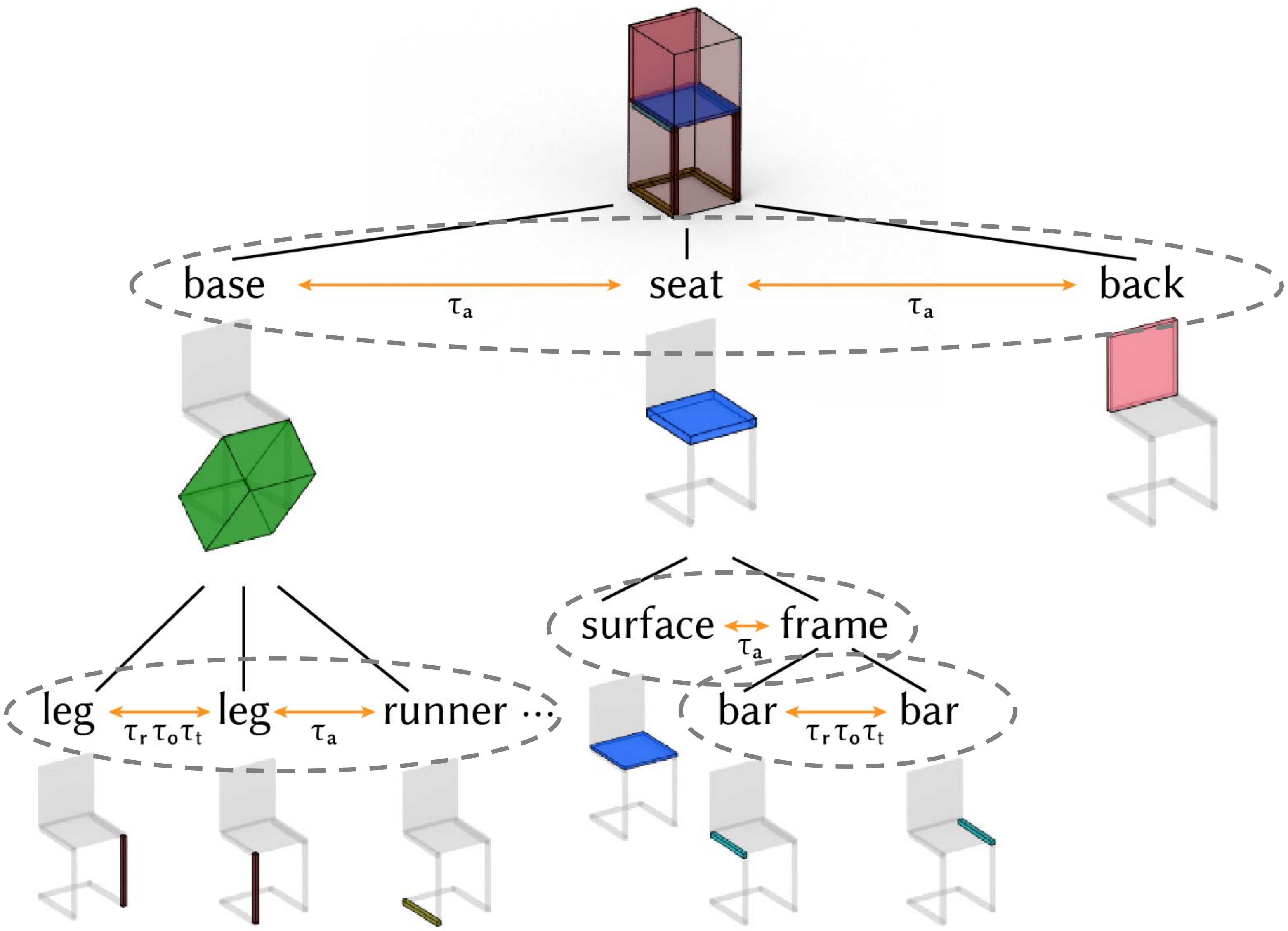
Reflectional Symmetry τ_r
Rotational Symmetry τ_o
Translational Symmetry τ_t
Adjacency τ_a



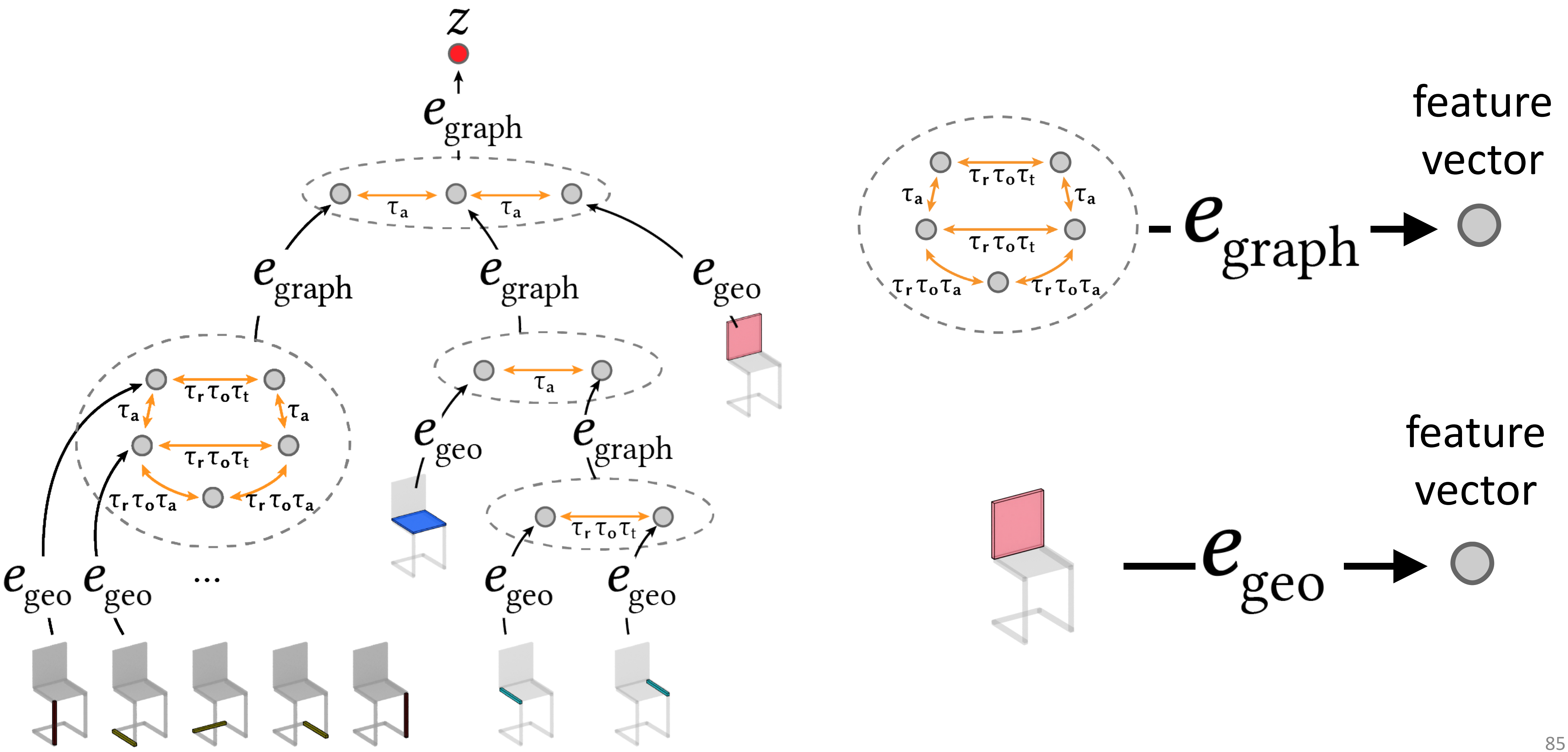
Object Representation: Example



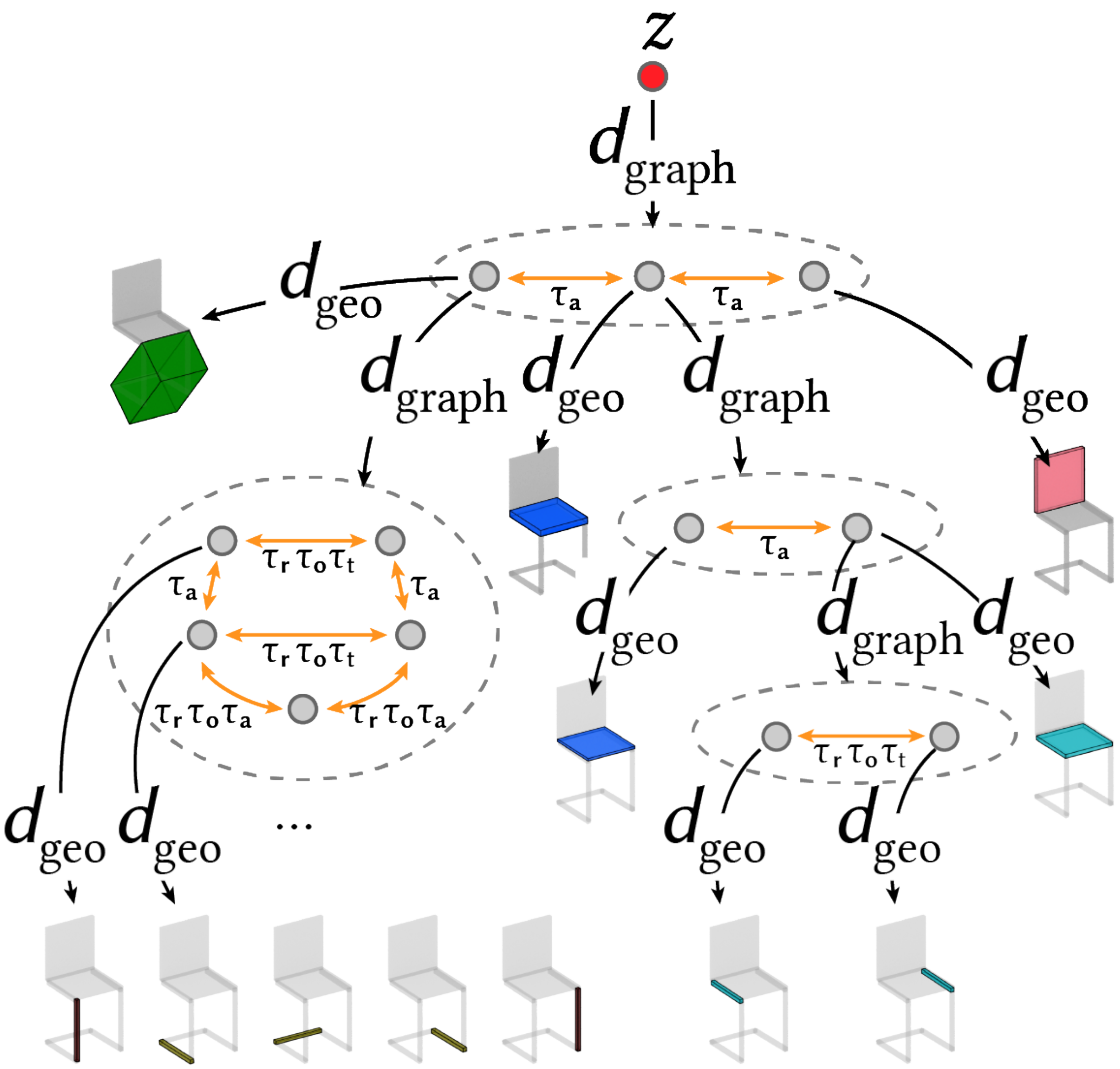
A Hierarchy of Graphs

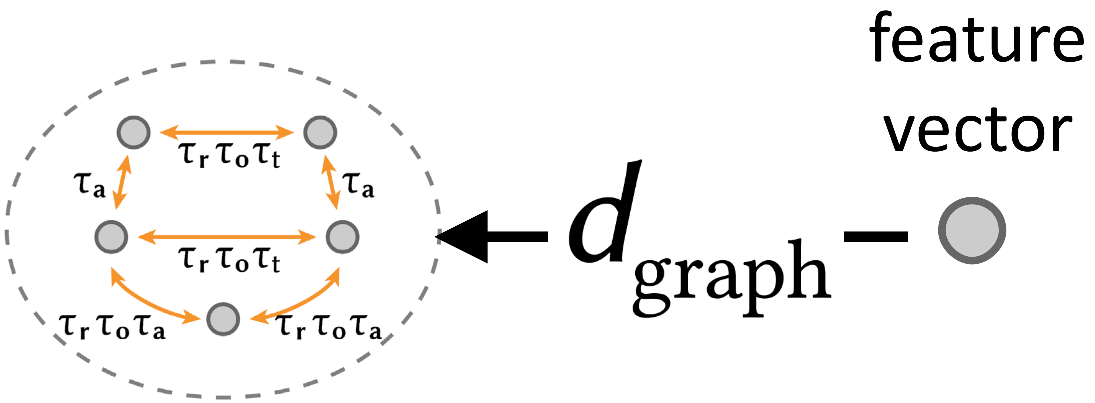


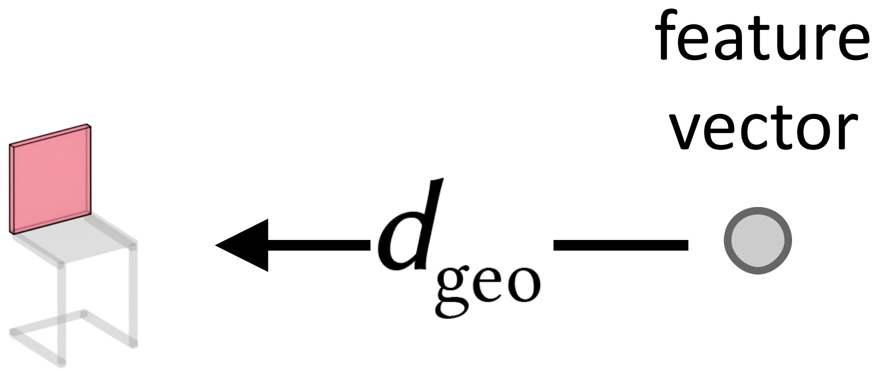
Hierarchical Graph Encoder



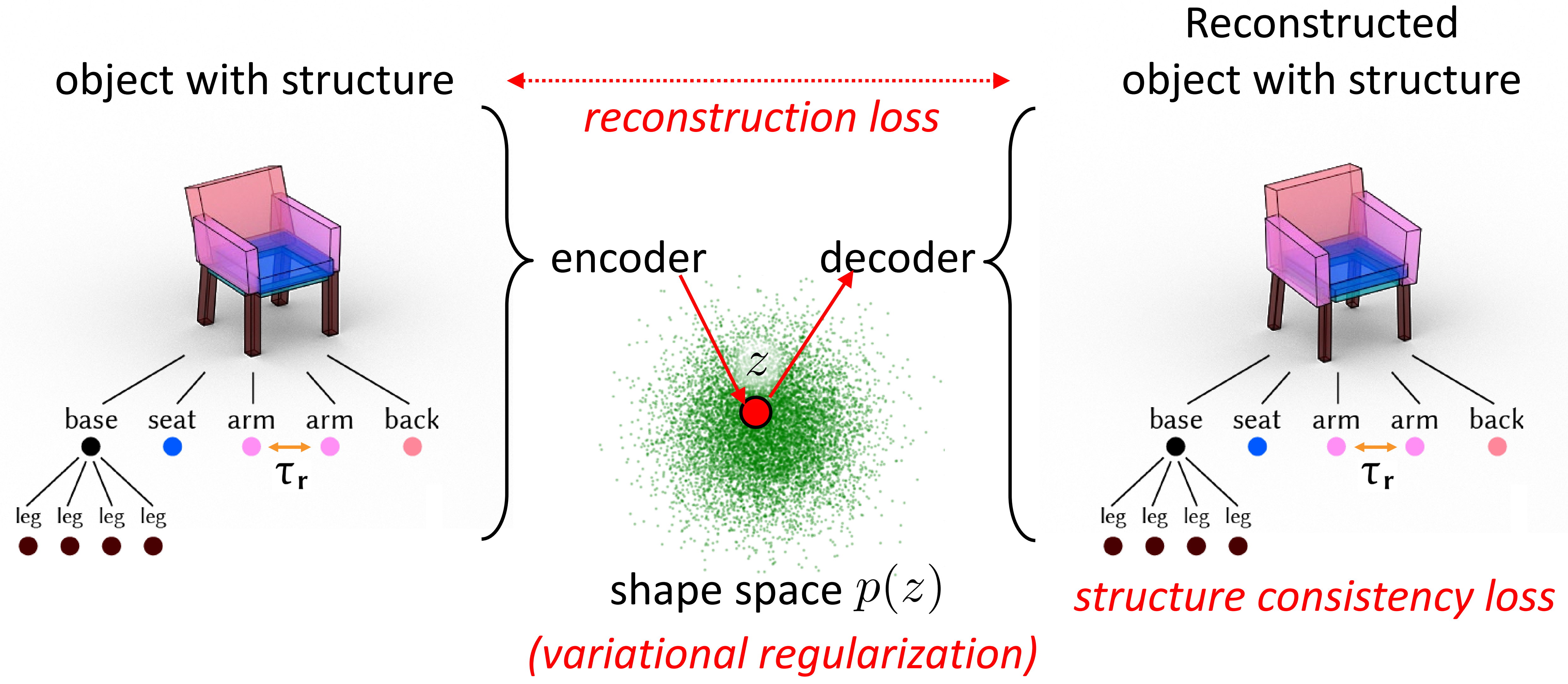
Hierarchical Graph Decoder



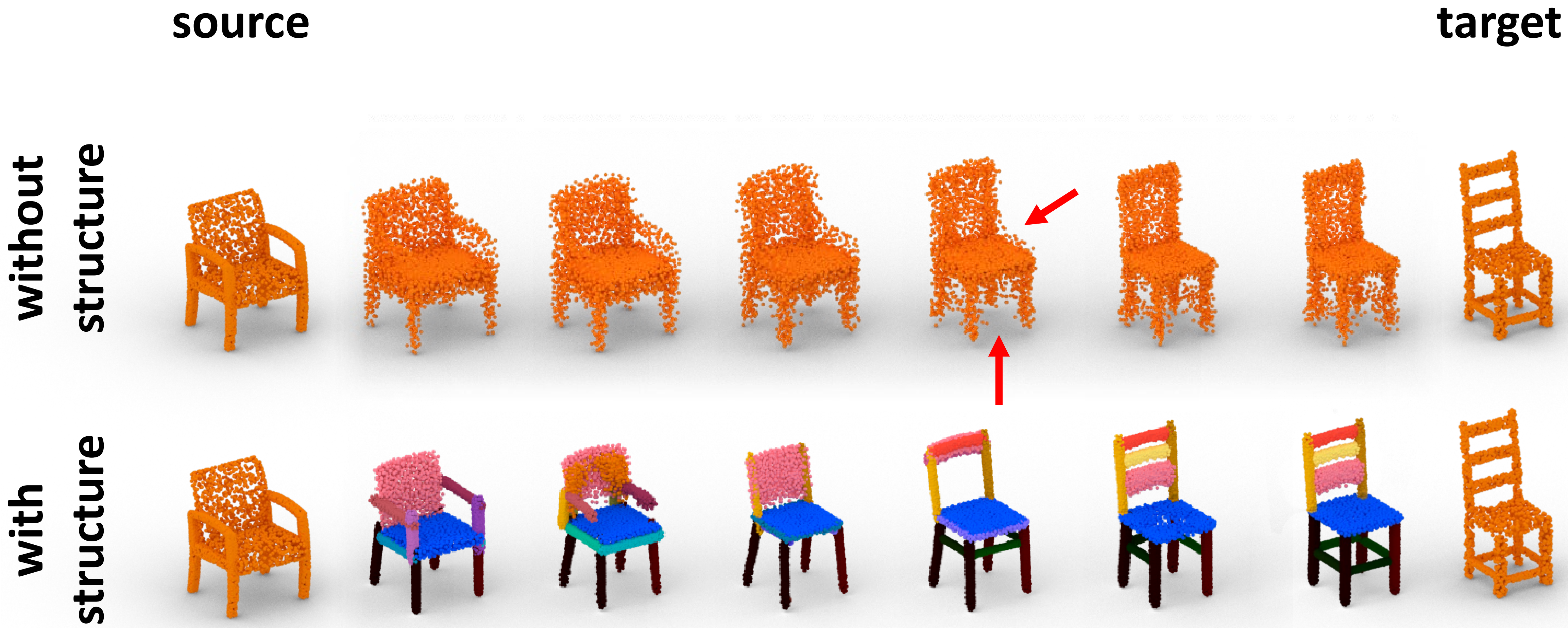




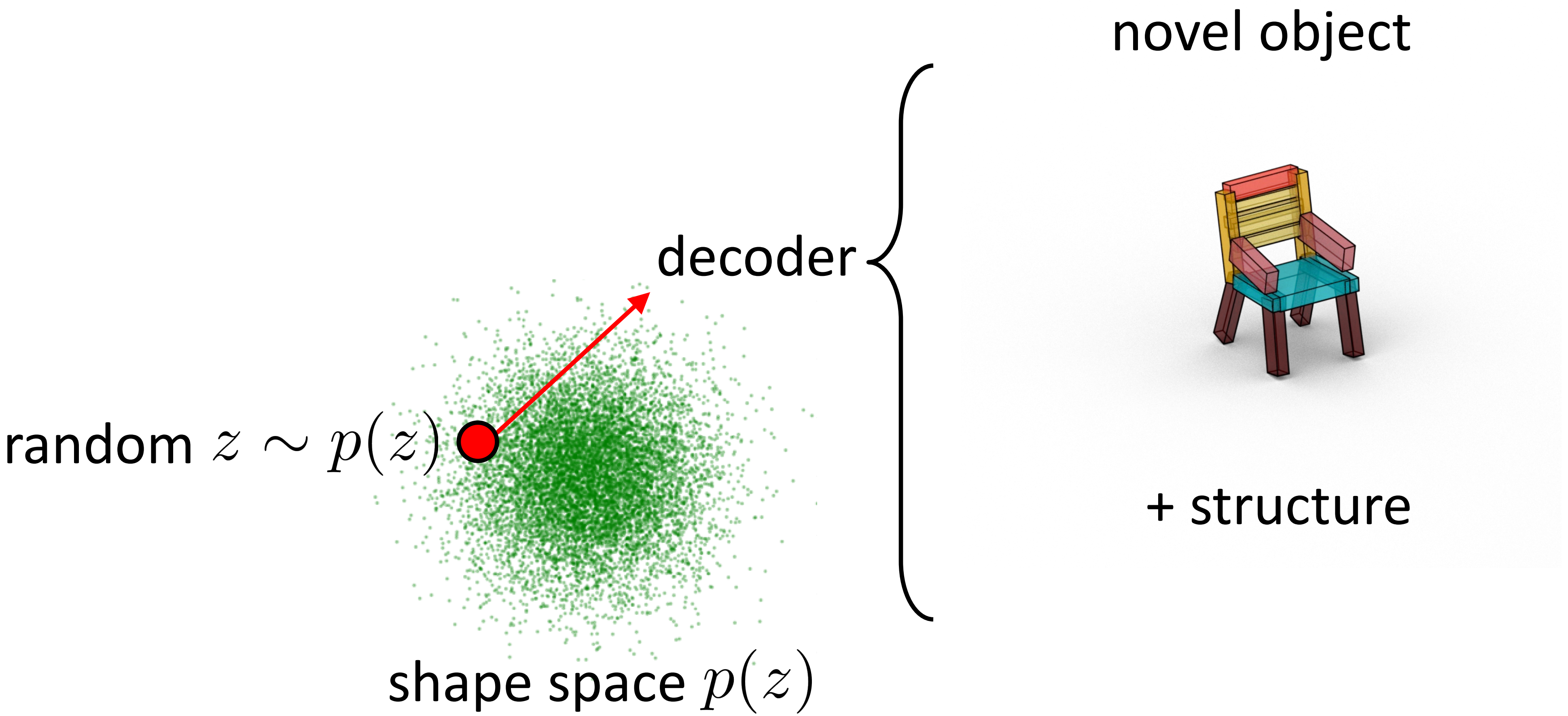
Architecture Overview: VAE Training



Interpolation With vs. Without Structure



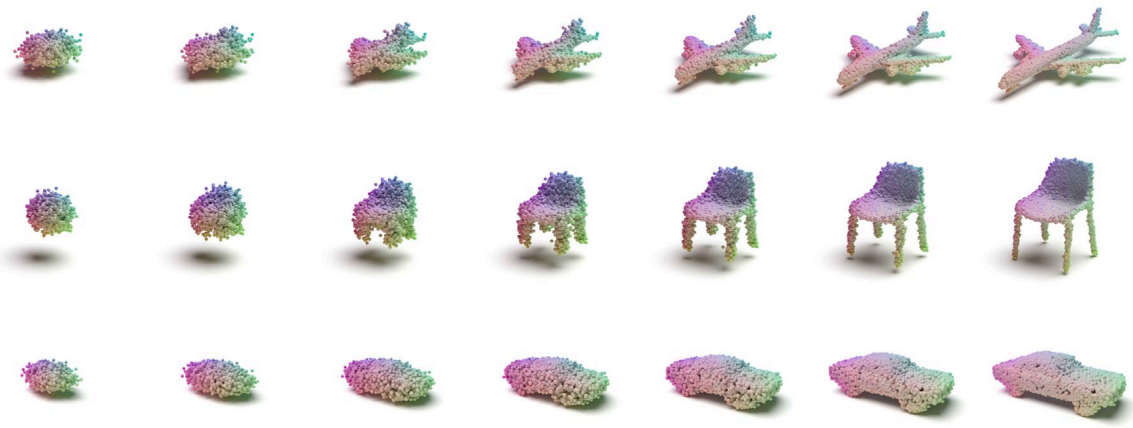
Application: Generation



Generation

Flow-Based 3D Generative Model

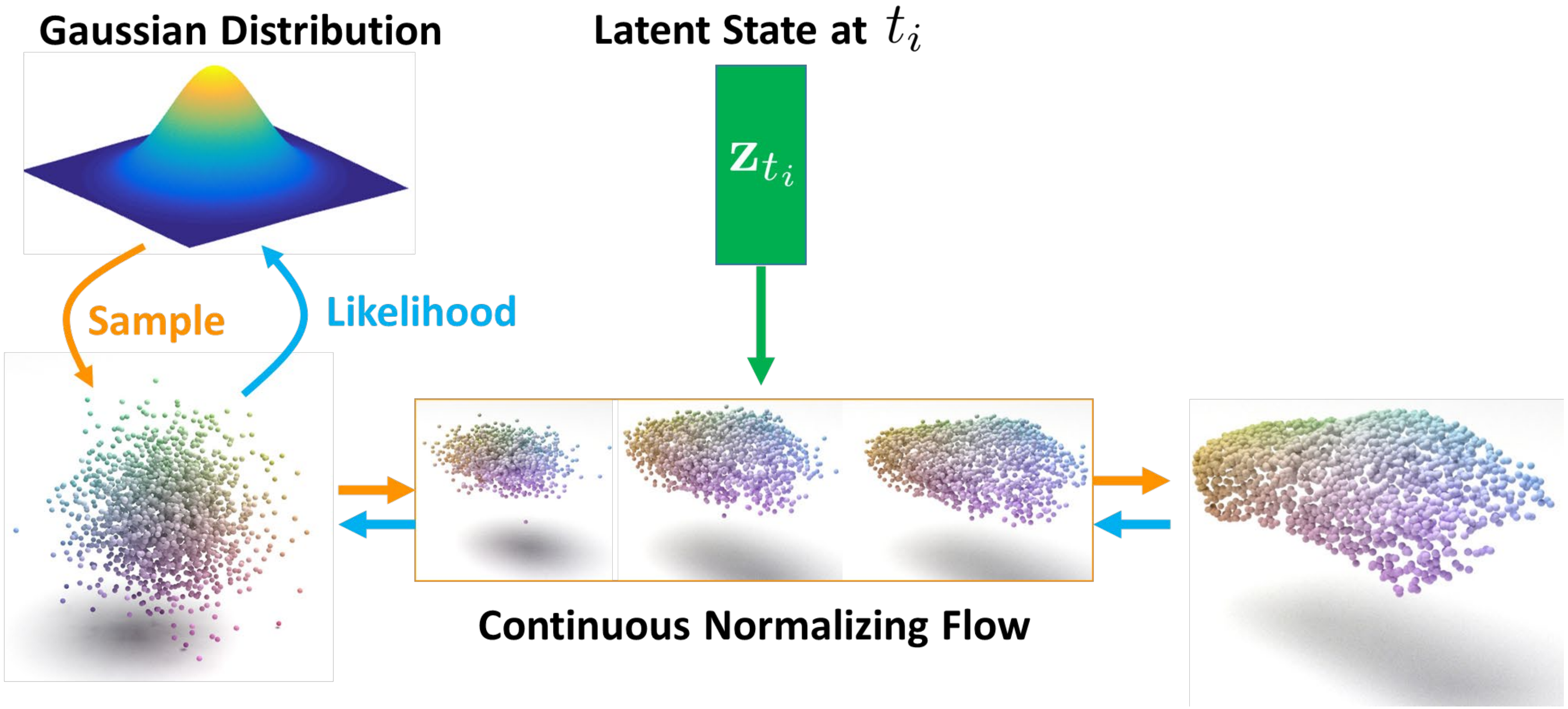
Discrete Point Flow Networks
From Univ. Grenoble Alpes



PointFlow (continuous normalizing flow)
From Cornell

Note that bijectivity requires same dimensionality.
From left to right: latent points to generated points

Flow-Based Generative Model



Flow-based model is constructed by a sequence of **invertible transformations.**

Explicitly modeling probability. Loss: negative loglikelihood of z = f(x)

Image credit: [Lil'log]

How to Learn Generative Models

- Explicitly modeling data probabilistic density, learn a network $p_\theta(x)$ that maximize data probability

- Implicitly modeling probabilistic density, e.g. learn a network that scores how "real" the generated data is, $f_\theta(x)$

→

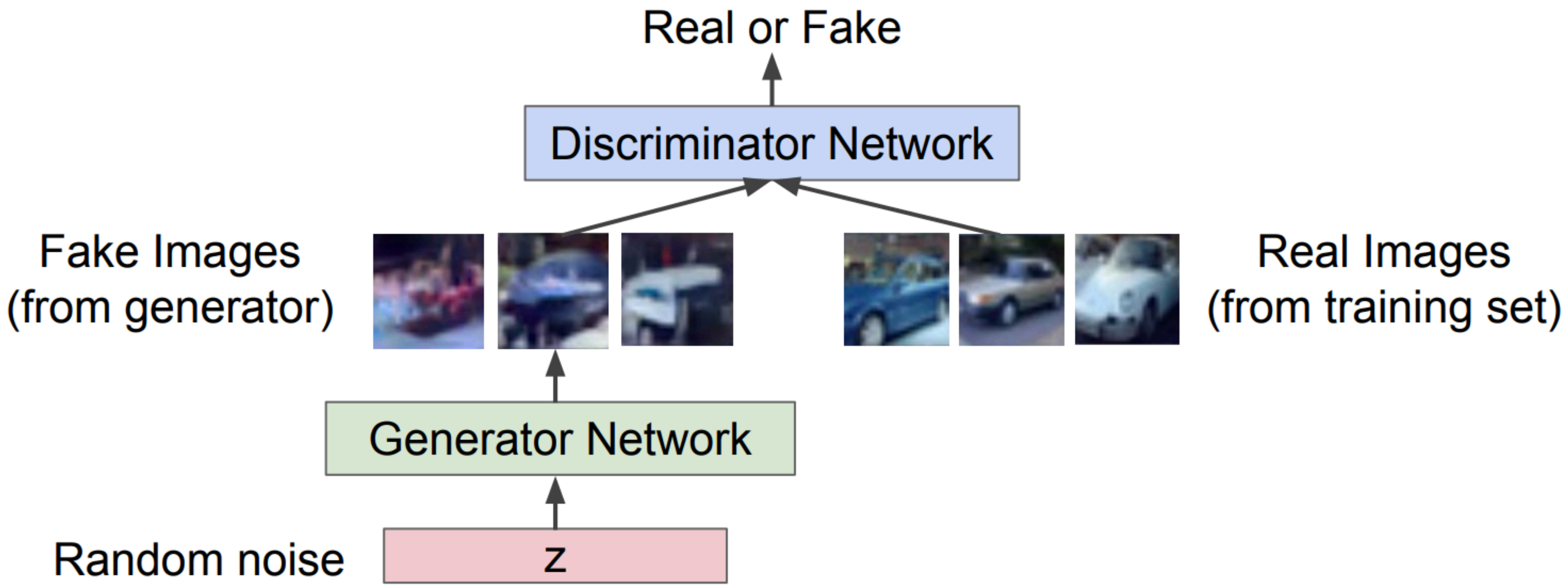
- Markov chain
- Autoregressive models
- Variational autoencoder (VAE)
- Flow-based models
- Structure-based models
- Energy based models
- …

→

- Generative adversarial network (GAN)
- Score-based generative
- …

GAN

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



Credit: Stanford CS231N

Voxel GAN

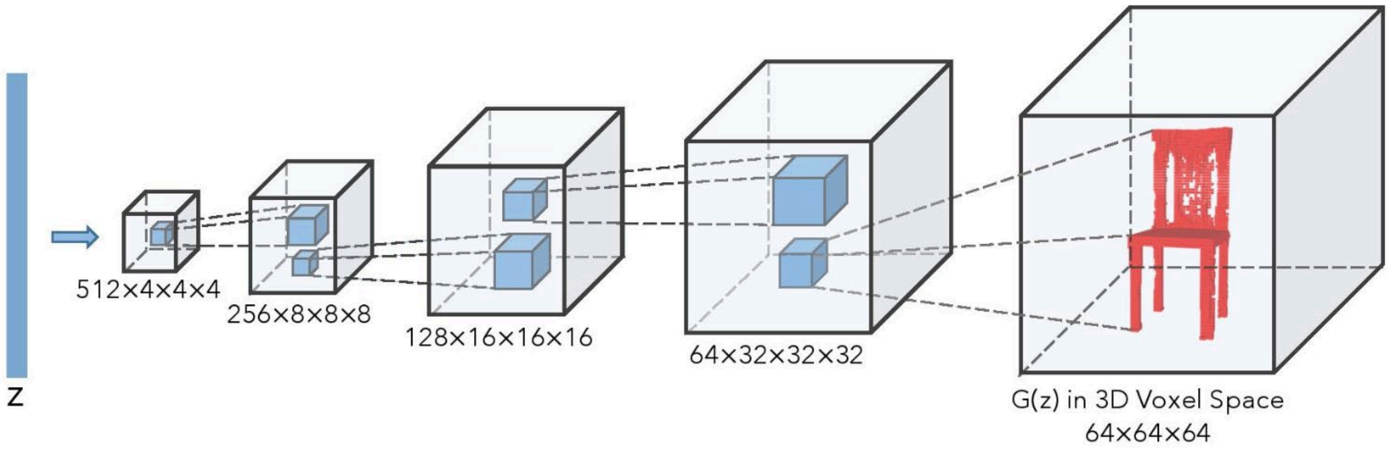
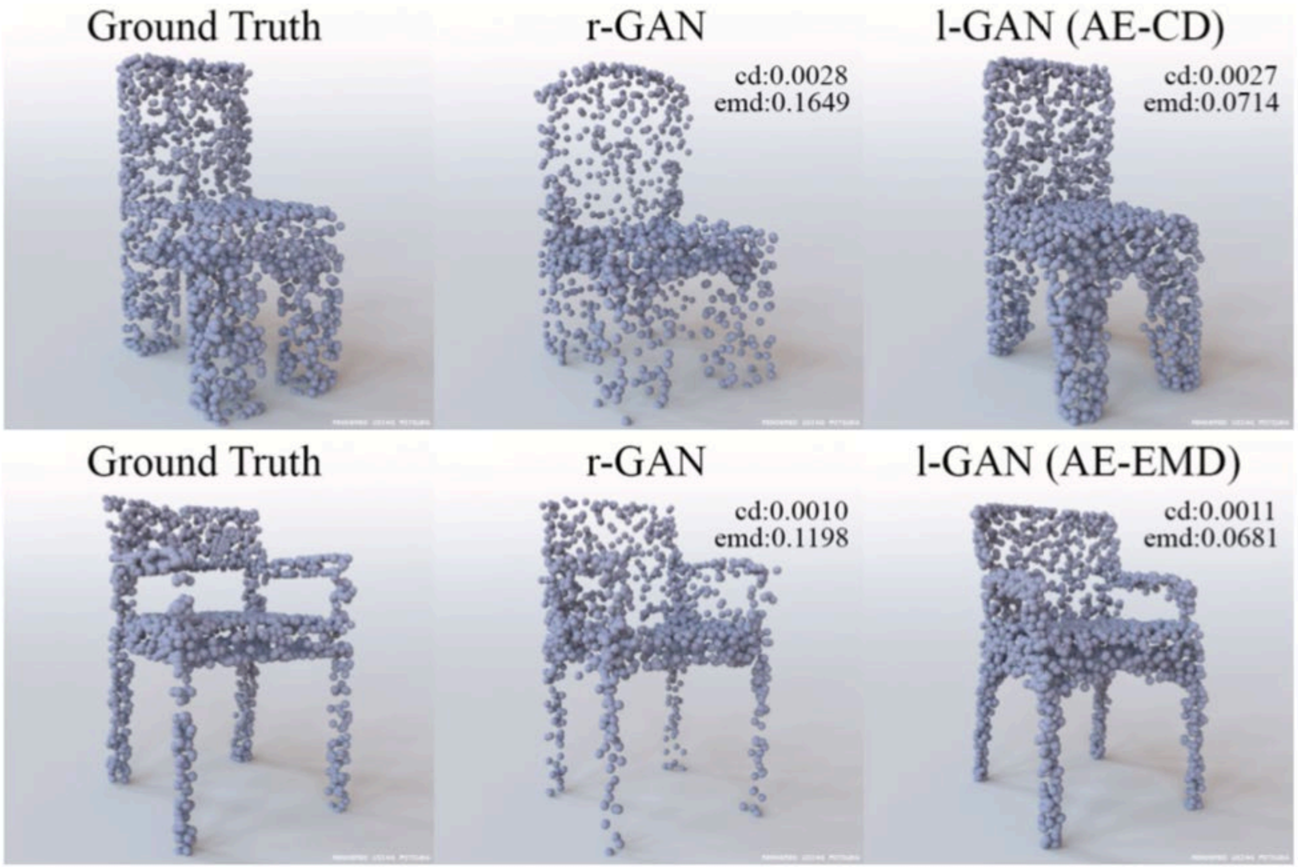


Figure 1: The generator of 3D Generative Adversarial Networks (3D-GAN)

Figure 2: Shapes synthesized by 3D-GAN

Wu et. al., **Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling**, NeurIPS 2016

Point Cloud GANs



ICML, 2018, Learning Representations and Generative Models for 3D Point Clouds, Panos Achlioptas, et. al.

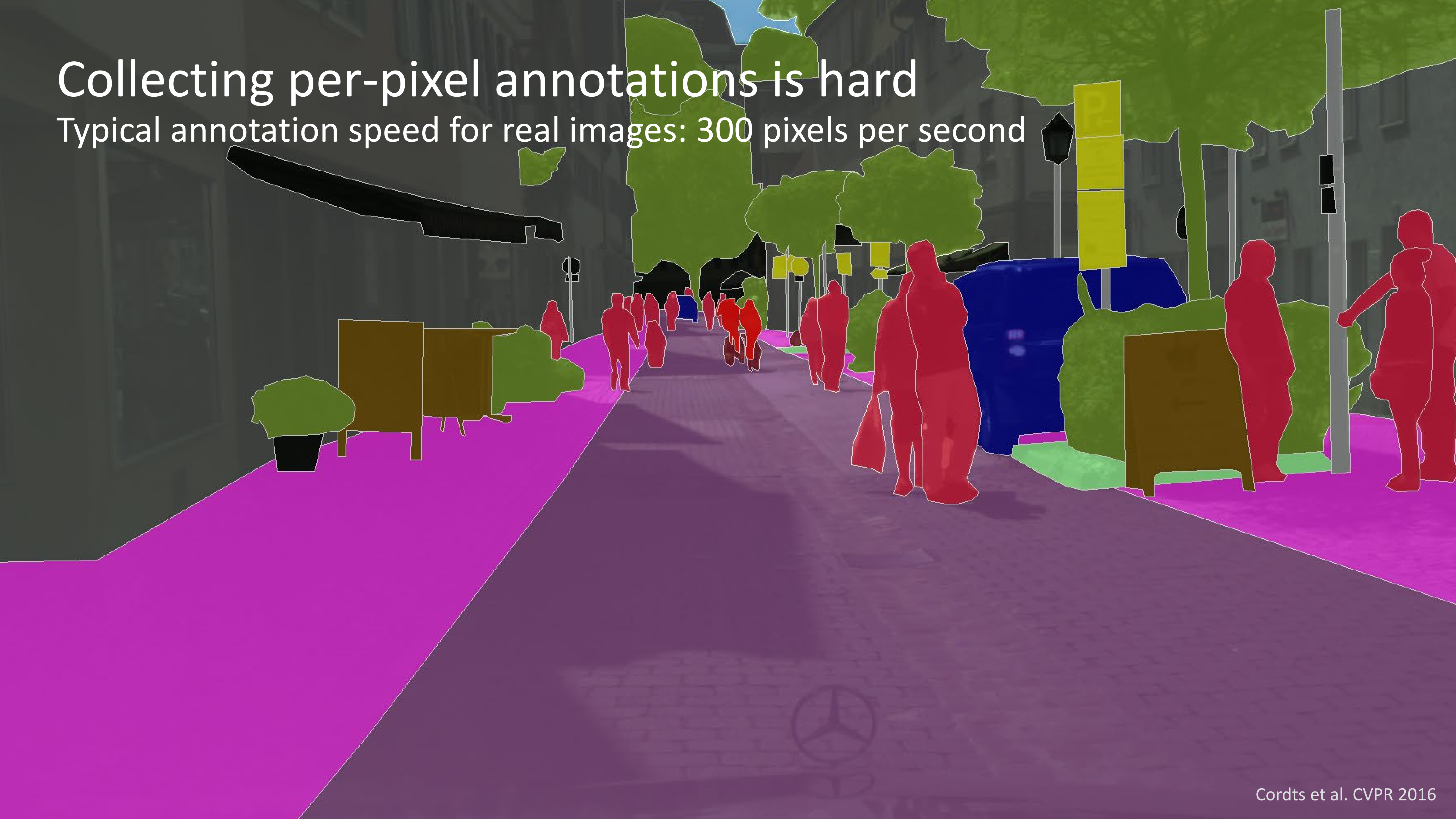
In 3D, Not Close to Direct 2D Results Using GANs

https://thispersondoesnotexist.com

Synthetic 3D for ML

Collecting per-pixel annotations is hard

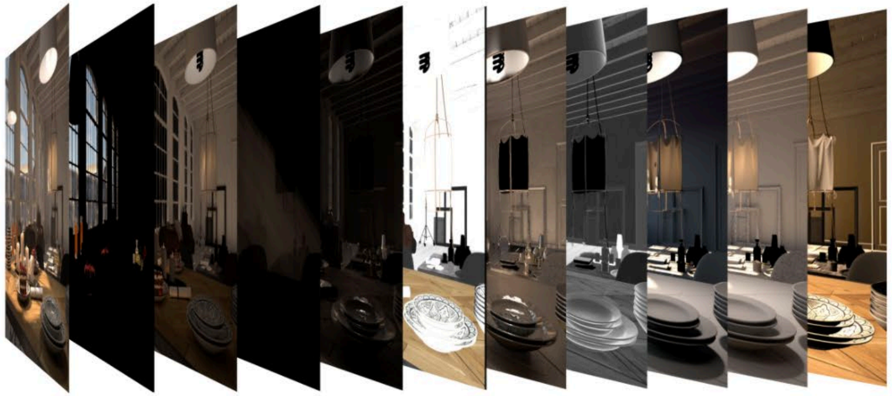
Typical annotation speed for real images: 300 pixels per second



Cordts et al. CVPR 2016

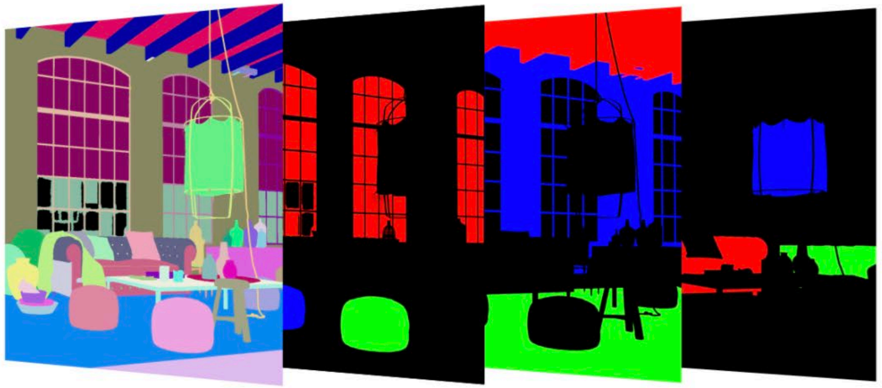
Synthetic Data Easily Provides Multi-Modal Annotations

Beauty Render Elements



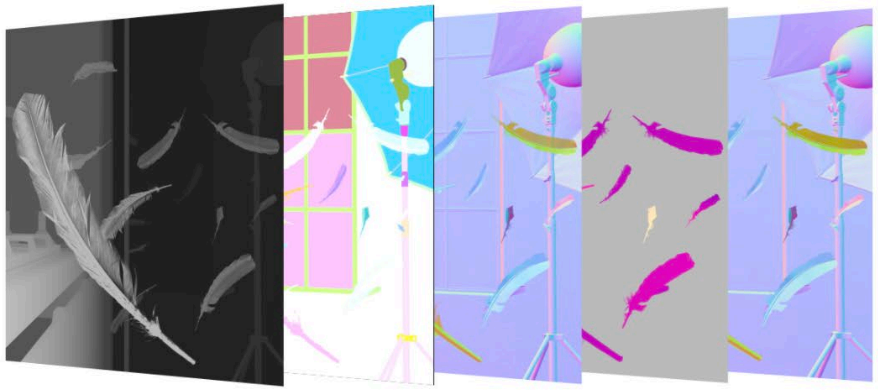
The Beauty Render Elements are the primary render elements that form V-Ray's pre-composited final render pass Beauty RGB_Color. These Include the Lighting, Global Illumination, Reflection, Refraction, Specular, etc.

Matte Render Elements



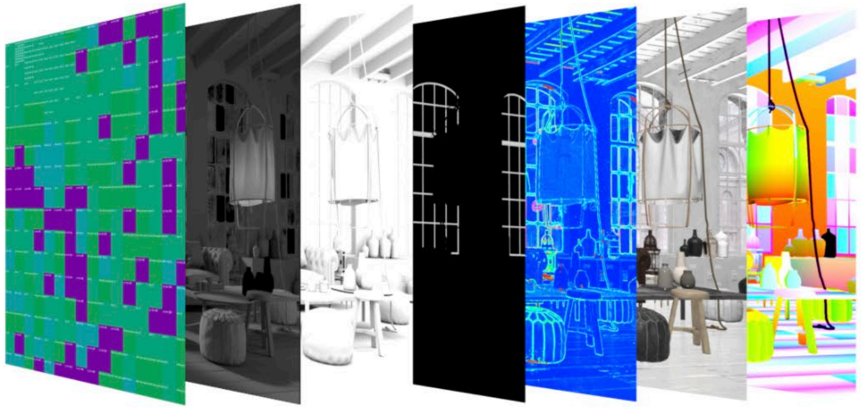
The Matte Render Elements aid selection masking in the compositing process. These include the Material ID, Multimatte, Object ID, and Render ID render elements.

Geometry Render Elements



The Geometry Render Elements rely on geometry to generate geometry data for the compositing process. These include the Velocity, Z-Depth, Wire Color, and Normals render elements.

Utility Render Elements



The Utility Render Elements give insight into how V-Ray is running and extra functionality for compositing. These include the Distributed Render ID, Sample Rate, Denoiser, Extra texture, Sampler Info, etc

Image credit: V-Ray

Synthetic Data Improves Performance

Richter et al. ECCV 2016 | Johnson et al. ICCV 2019 | Li et al. ICCV 2019 | Ehsani et al. CVPR 2018

Real-World Tasks

Need Quantity, Quality, Diversity in Generation

Zhang et al. CVPR 2017

Li and Snavely ECCV 2018

Less photorealistic

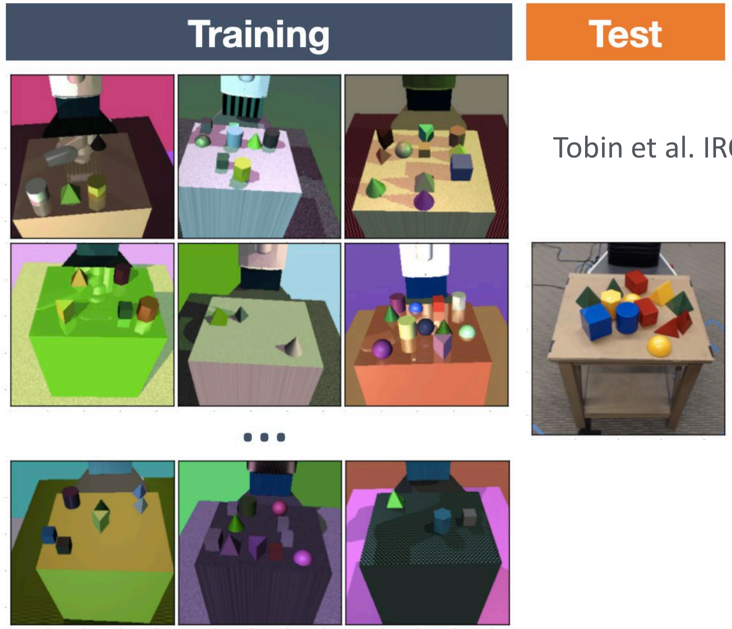
More photorealistic

More photorealism is better

Zhang et al. CVPR 2017, Li and Snavely ECCV 2018

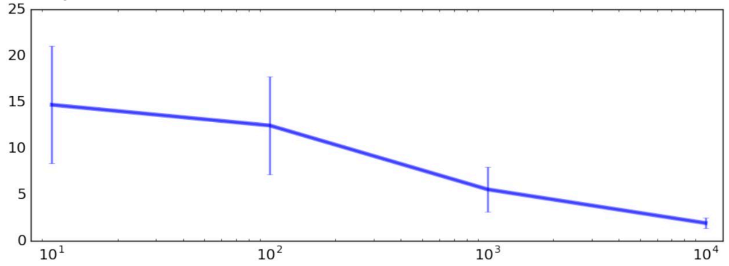
"Sim2Real"

[Roberts and Paczan]



Tobin et al. IROS 2017

Object localization error (cm)



Number of unique textures seen during training (log scale)

Greater diversity is better

Want all plausible scenes, views, lighting conditions, materials, ...

Sadeghi and Levine RSS 2017, Tobin et al. IROS 2017

Synthetic Indoor Environments

Hypersim: A Photorealistic Synthetic Dataset for Holistic Scene Understanding

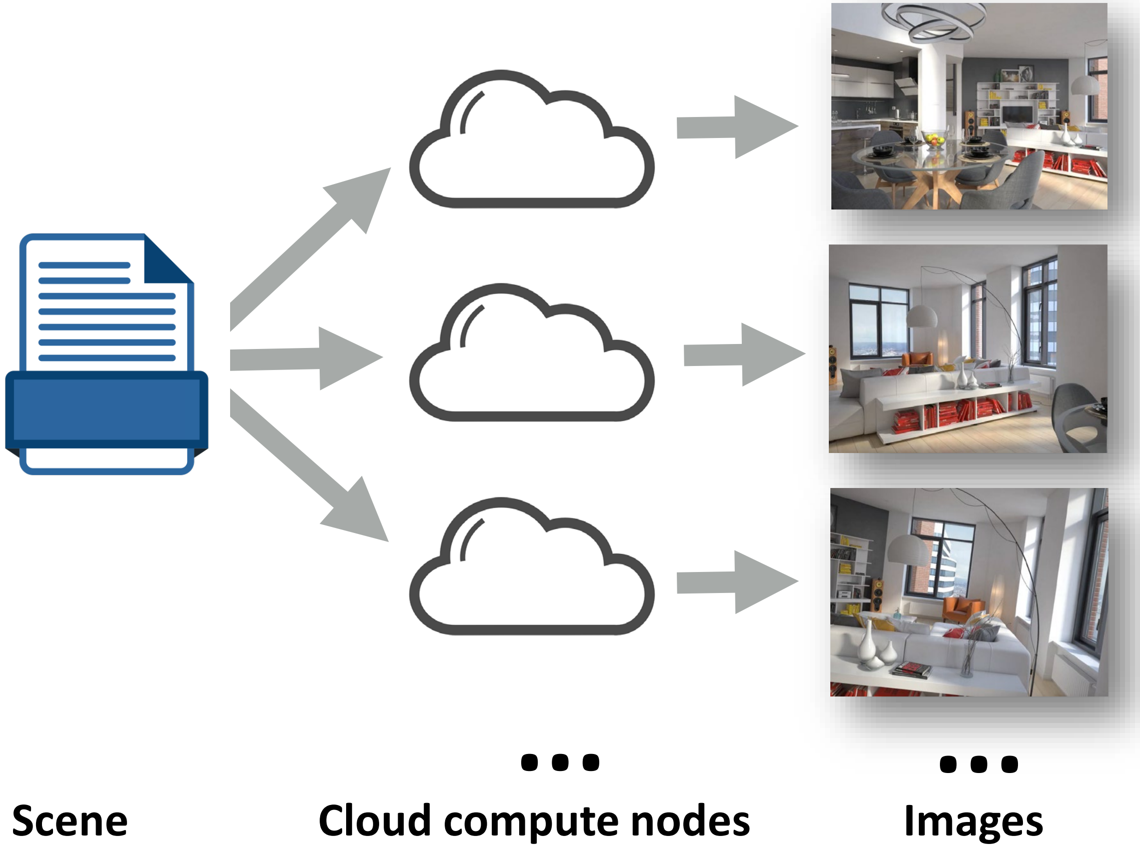
461 indoor scenes, 77k images

Images ✓ 3D assets ✓ Segmentation ✓ Intrinsic ✓

Roberts and Paczan (under review)

Synthetic Imagery Has Its Own Costs

Specialist design time/effort



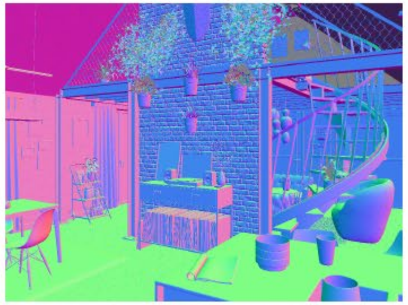
Photorealistic rendering in the cloud is very costly

$51K for 77K images @ 1024x768 resolution

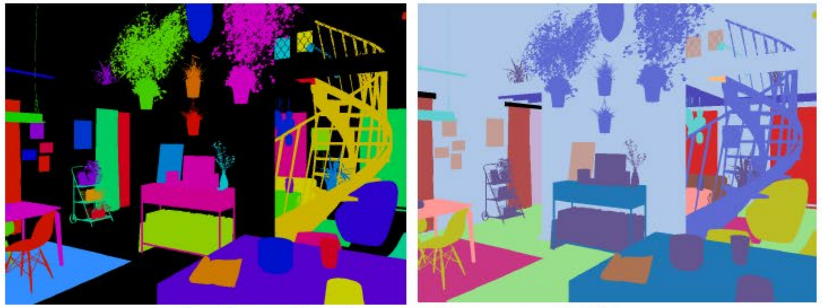
Intermediate-Level Annotations

RGB image

Depth



Surface normal



Instance-level semantic segmentation

Non-Lambertian intrinsic image decomposition

Hypersim: A Photorealistic Synthetic Dataset for Holistic Scene Understanding

461 indoor scenes, 77k images

Images ✓ 3D assets ✓ Segmentation ✓ Intrinsic ✓

Roberts and Paczan (under review)

Opportunity: Deferred Shading

Diffuse reflectance | Diffuse illumination | Non-diffuse residual | Final image

can randomize

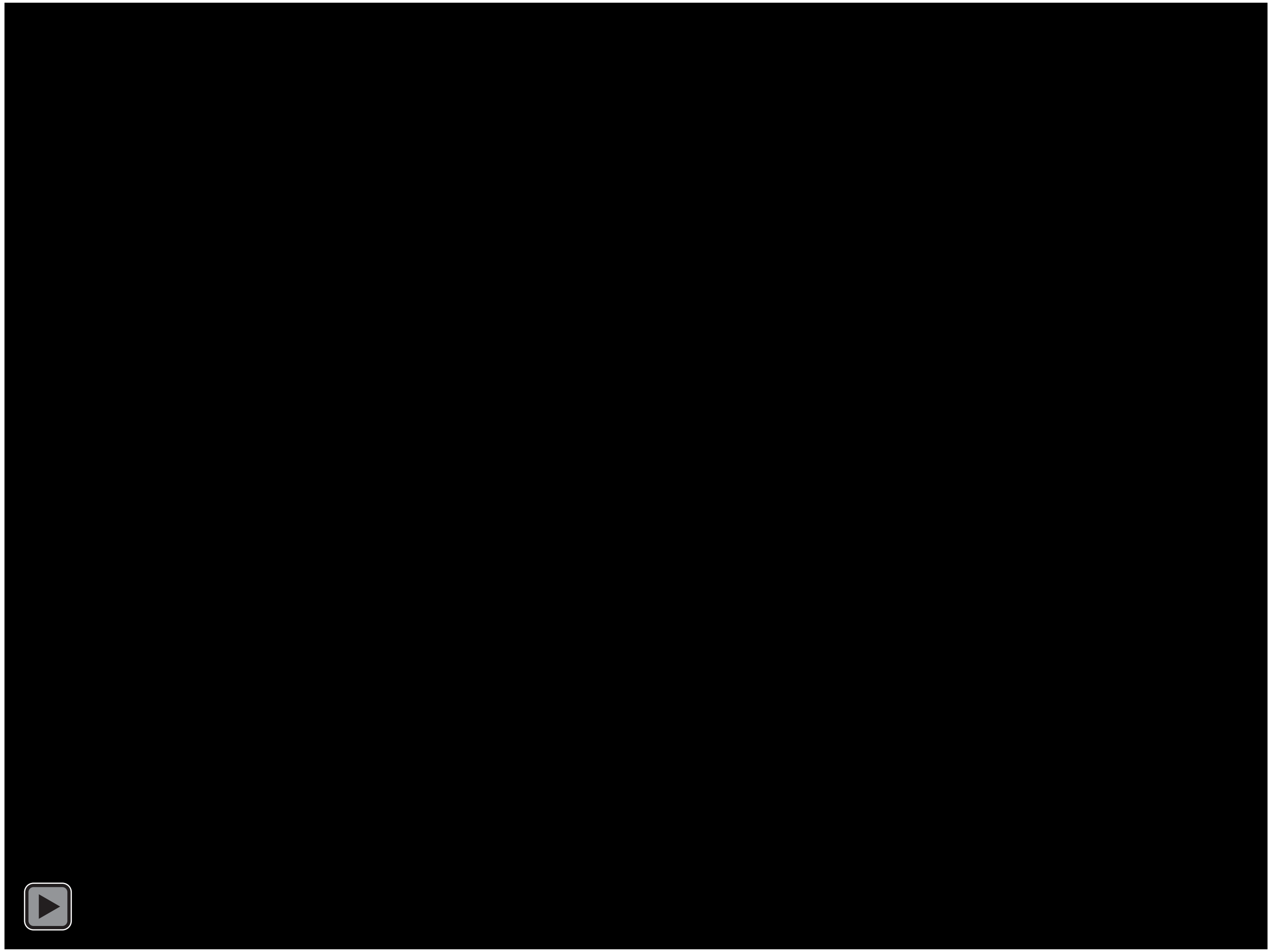
Fixed camera, fixed lighting, dynamic materials

Faster Shading

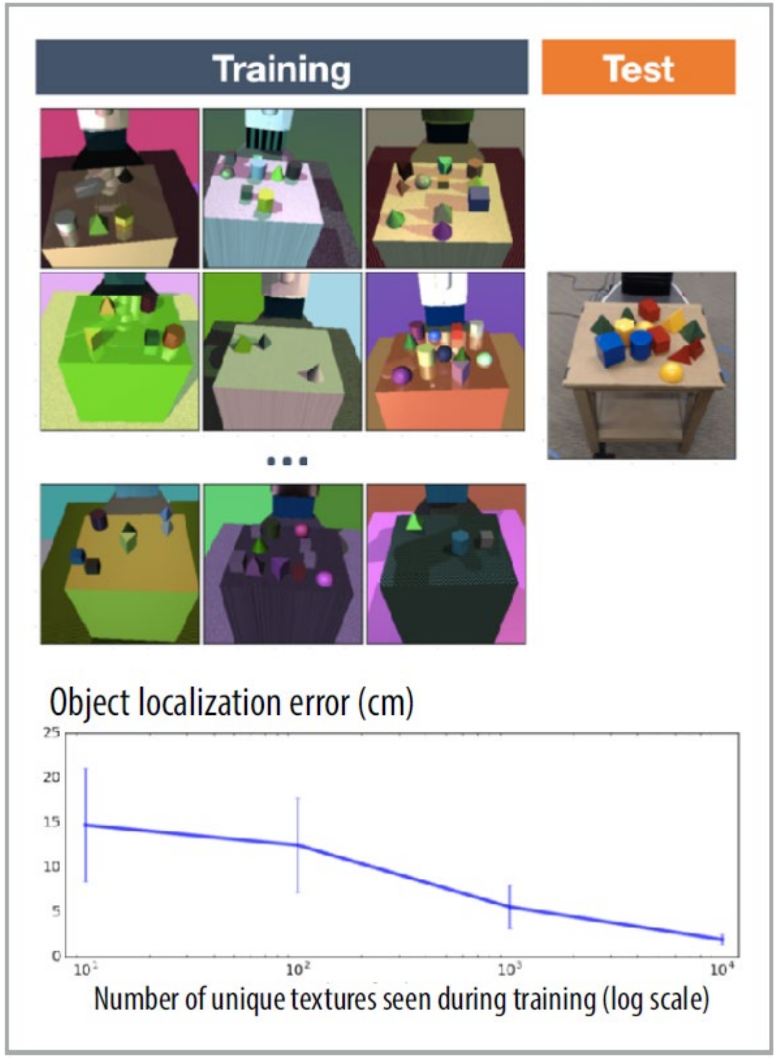
Original Image
1.5 hours to render

Random Variations
10 milliseconds to render

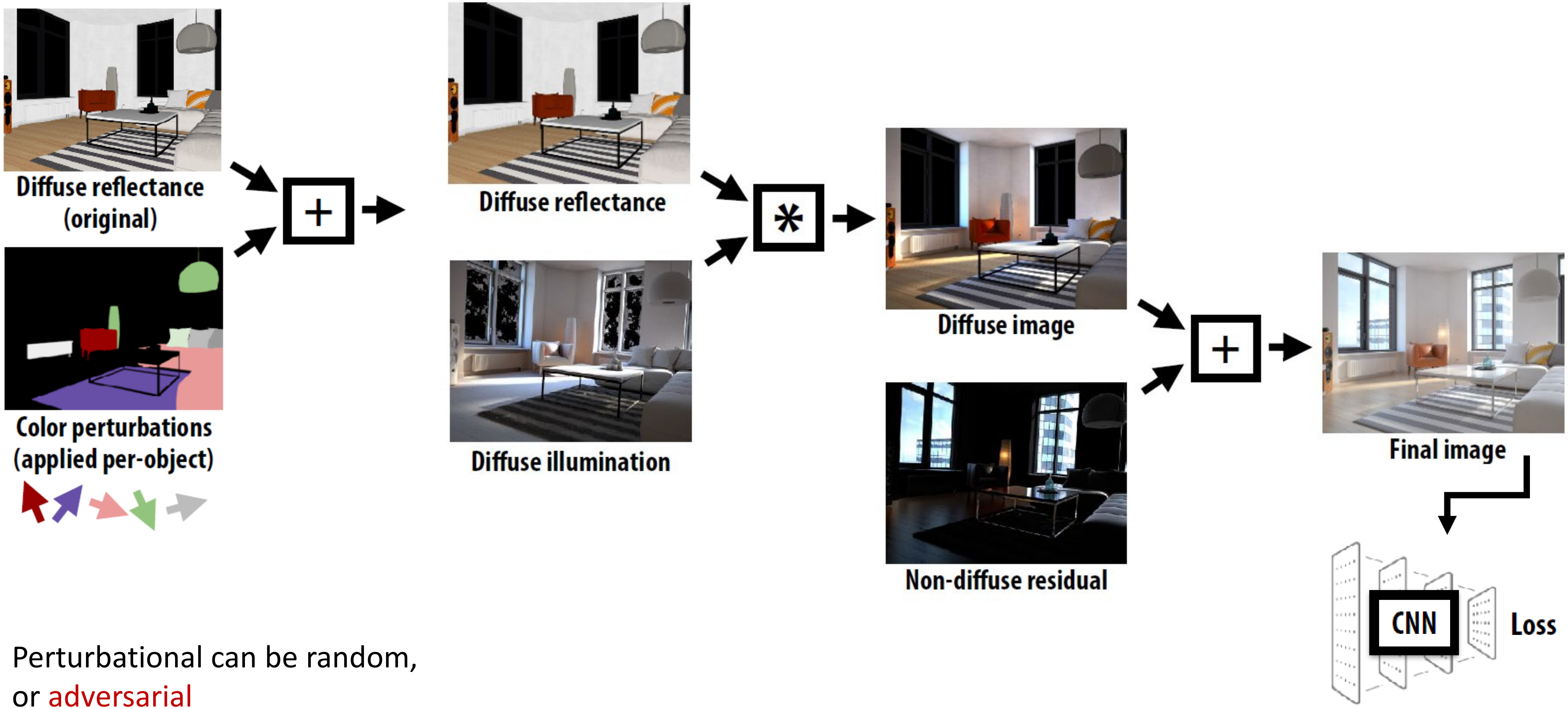
Faster Shading



Tobin et al. IROS 2017



Deferred Shading is Differentiable

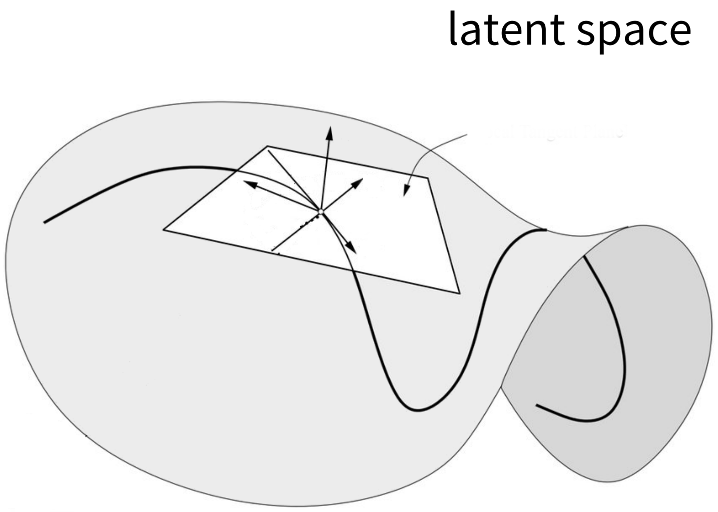


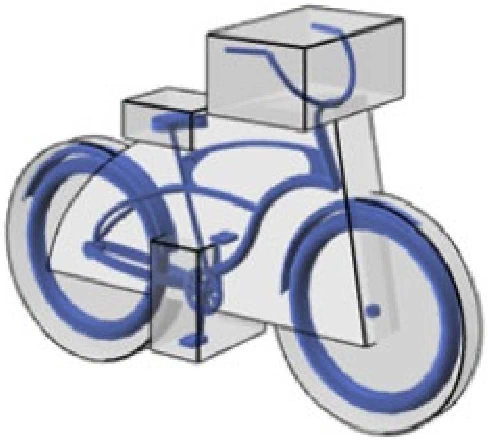
Perturbational can be random,
or adversarial

Fine-grained integration of data synthesis and ML training

Generative Model for Geometry Deformations/Edits

Learn possible variations of an input shape, discrete or continuous, meeting semantic constraints.





Learning Variation Generation

Learning to Vary

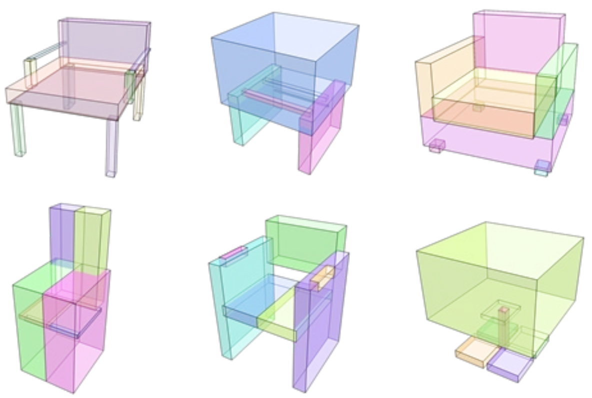
- Re-use what we already have
- Populate sparsely sampled regions

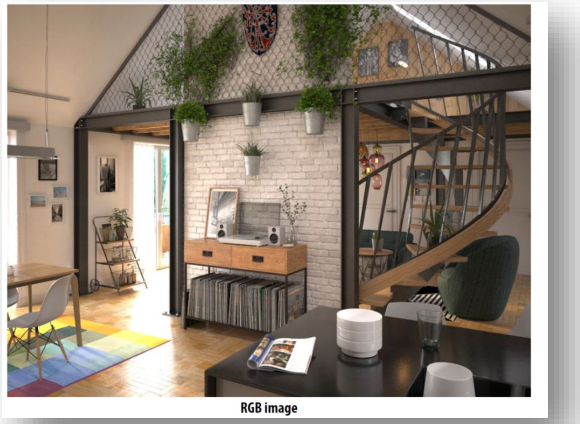
Geometry,
Arrangement,
Appearance,
Motion

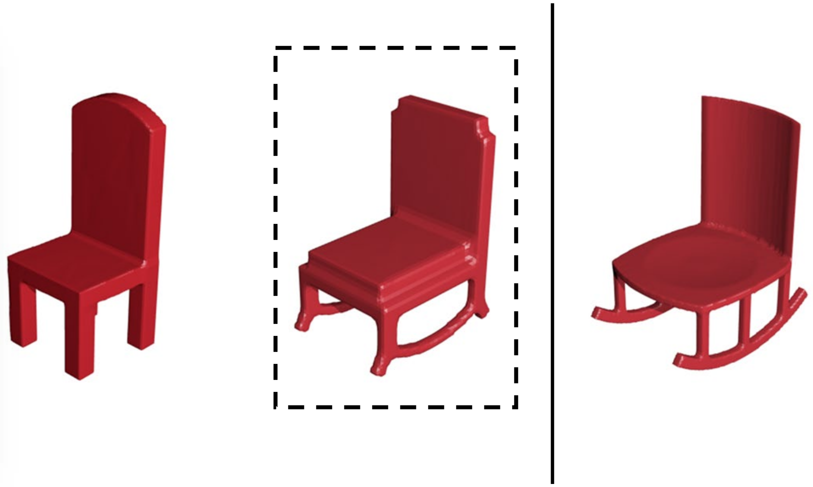
for Objects and Scenes

Varying to Learn

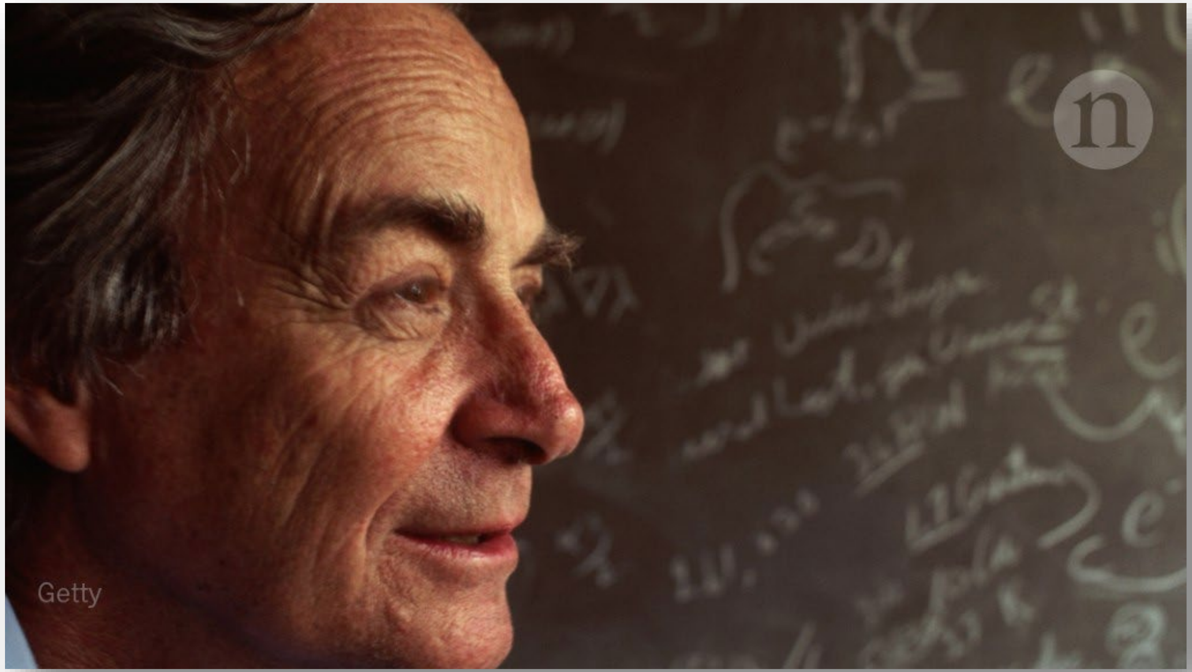
- Provide generation diversity
- Create training data tailored for hard concepts







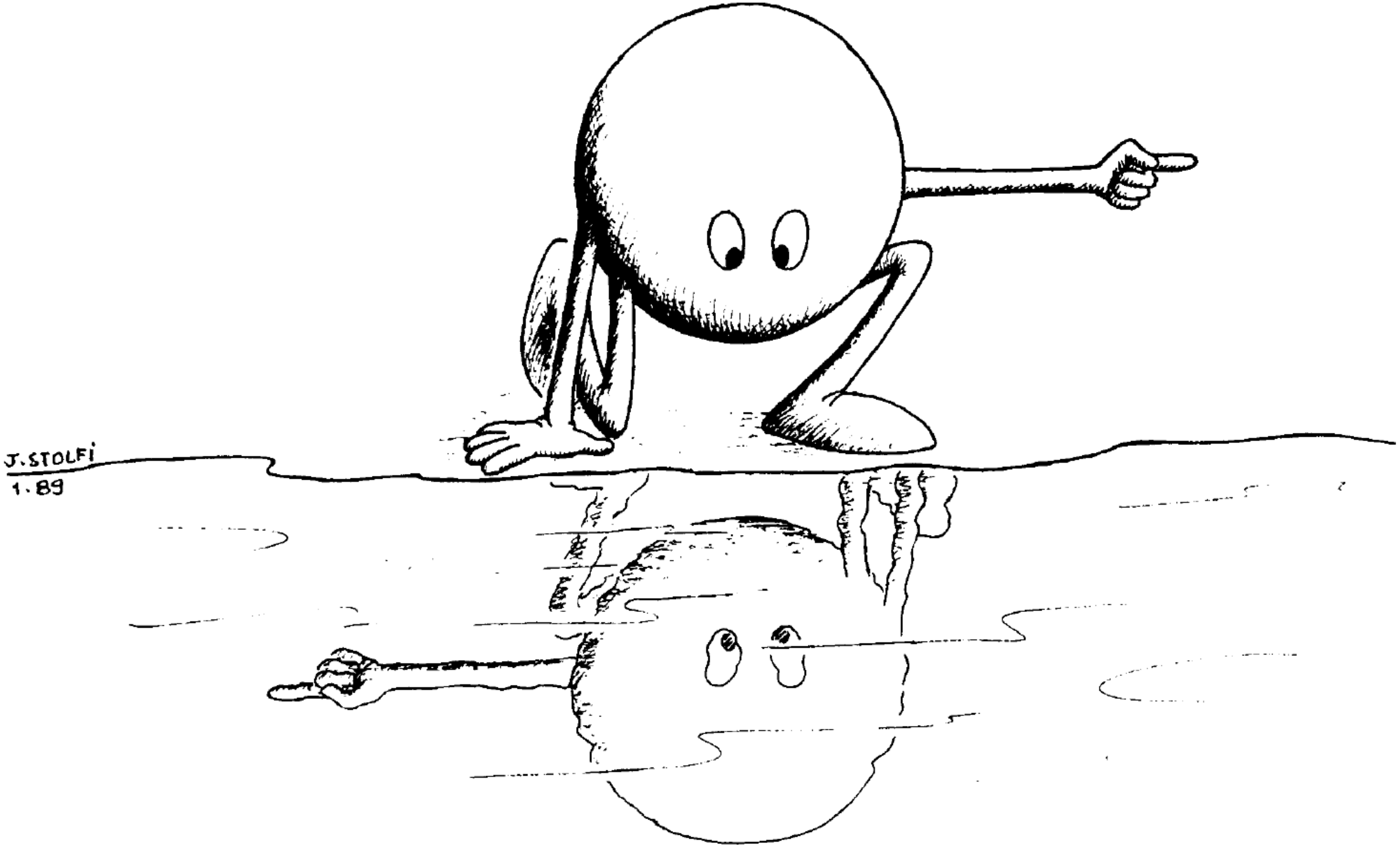
Generative Modeling



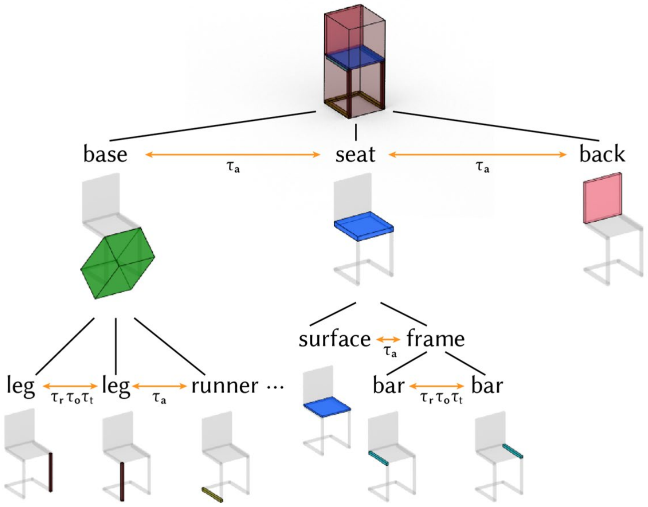
Richard Feynman: "*What I cannot create, I do not understand*"

Generative modeling: "*What I understand, I can create*"

That's All



CS348n: Neural Generative Models for 3D Geometry



Winter 2021-22