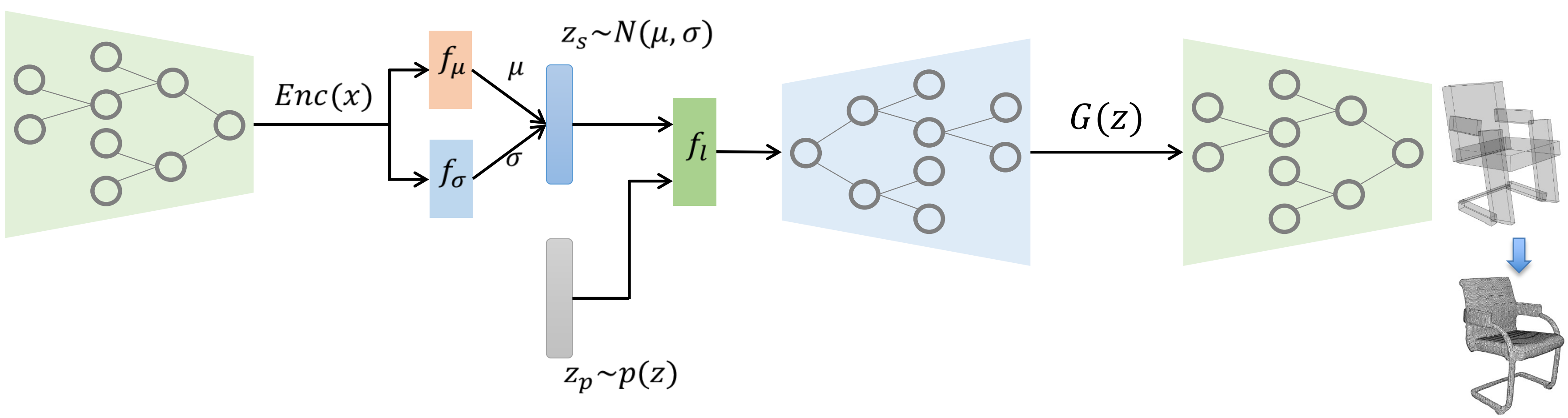
Learning to Generate 3D Shapes

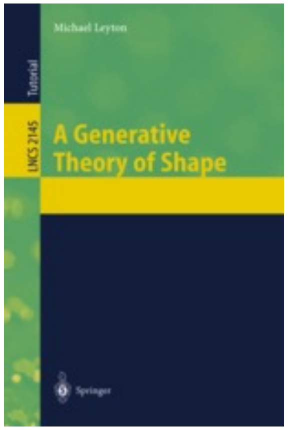


Richard Zhang

CMPT 464/764, Lecture 13

Generative models

- Models that describe and enable generation of intended outcomes, e.g., images, 3D shapes, or scenes sharing some commonalities
  - Procedural models, probabilistic sampling, genetic algorithms, etc.
- A good model should produce diverse yet plausible results



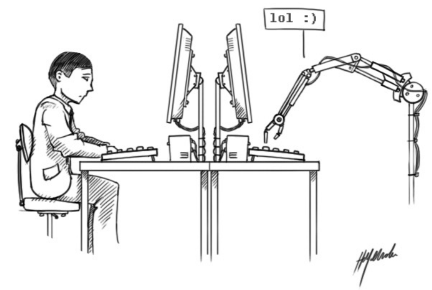
The interesting question is how to recover/infer/learn a generative model from a given set of outcomes

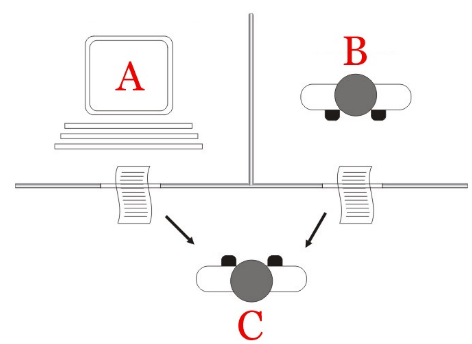
That is, to solve the inverse modeling problem

Learning to generate is at the heart of AI

When does a machine become "human"?

- Turing Test (1950): machine's ability to make human-like conversation ("passed" in 2014)





Learning to generate is at the heart of AI

When does a machine become "human"?

What really separates humans from machines is not the ability to make human-like conversation (the Turing Test), but the ability to be creative or be original!

- Lovelace Test: test machine's creativity
  - To craft a story, painting, 3D shape, or virtual scene
  - How to judge it: when the machine's creator cannot explain machine's creation

[Bringsjord, Bello, and Ferrucci, 2001]

Ada Lovelace (1815 – 1852)

Creativity is hard

Take a step back: from create to generate

- To just generate without requiring creative outcome
- To imitate (i.e., learn from examples) without being original
- Goals: plausibility, realism/quality, and diversity
  - Much recent success on synthesizing speech/face/natural images

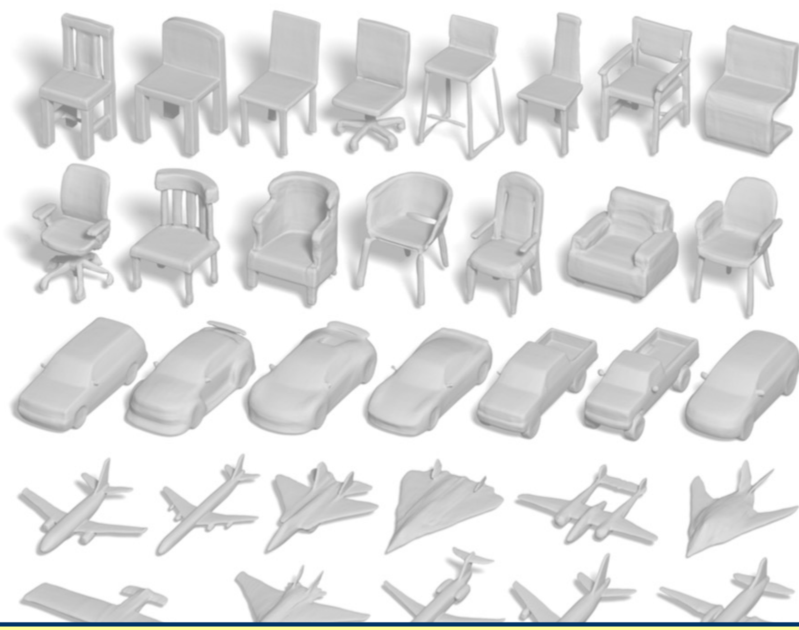
Remarkable progress on image generation

- Progressive GAN (Generative Adversarial Network) [Nvidia, 2016/17]

- BigGAN [Google Deepmind, 2018/19]

400 x 267 image resolution, using class conditionals

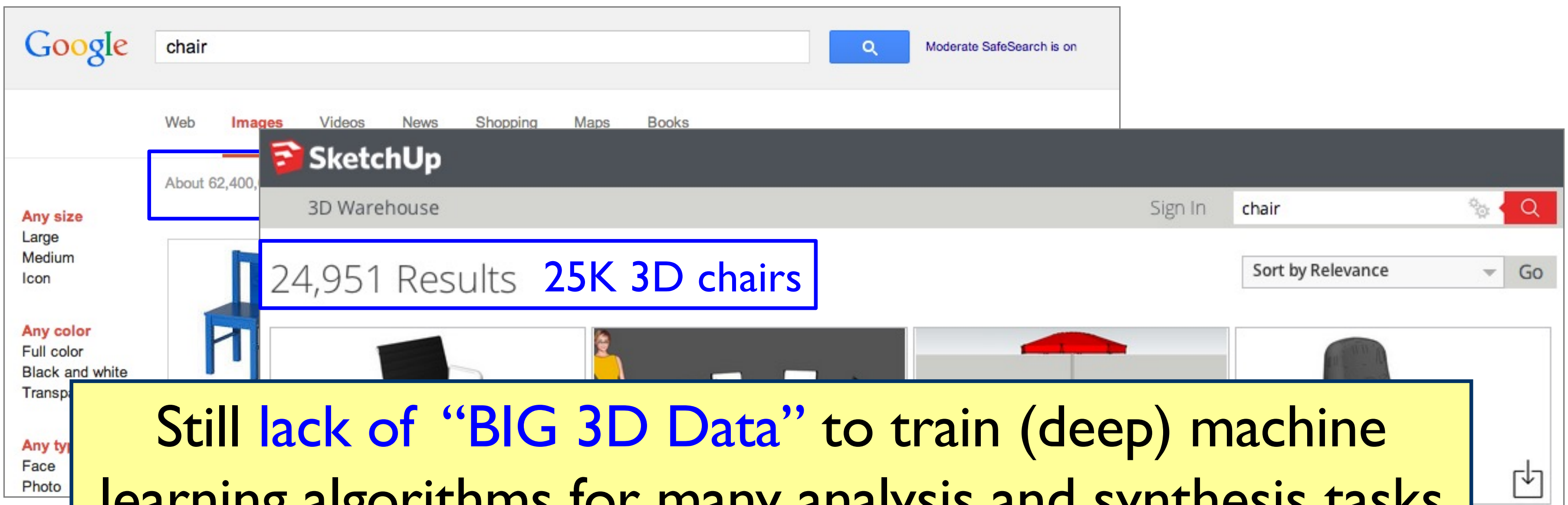
Our focus: neural generation of 3D shapes



There are some unique challenges to training deep neural networks (DNNs) for 3D shape generation and reconstruction …

Challenge #1: 3D data challenge

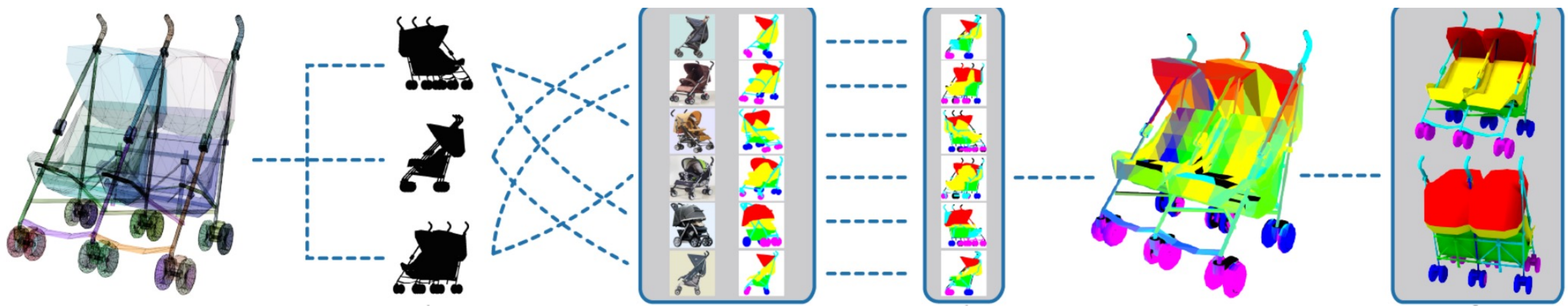
- Acquisition of and interaction with 3D contents are hard



Still lack of "BIG 3D Data" to train (deep) machine learning algorithms for many analysis and synthesis tasks

Useful ideas to address 3D data challenge

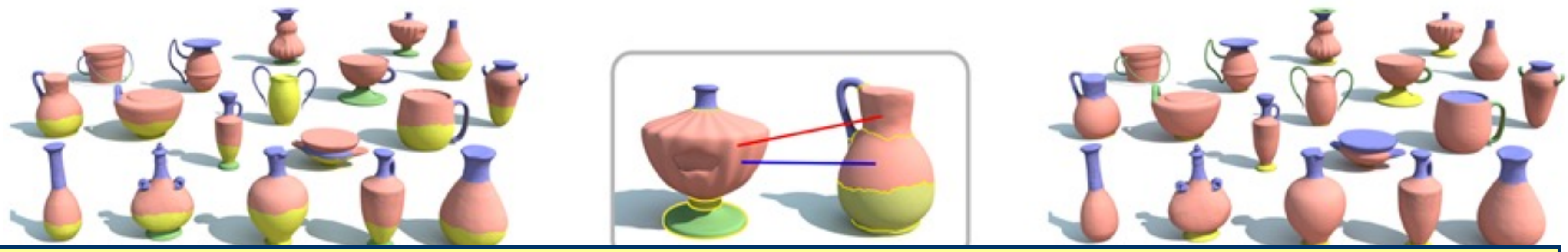
- Projective analysis: use annotated images to train 3D tasks



Projective Analysis for 3D Shape Segmentation [Wang et al. SIG Asia 2013]

Useful ideas to address 3D data challenge

- Projective analysis: use annotated images to train 3D tasks
- Minimizing user annotations: active learning



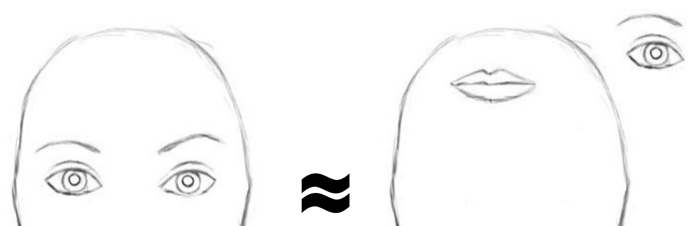
Unsupervised or weakly supervised learning: more challenging, more interesting, and less data bias (learns essence of problem?)

Challenge #2: affordance and functionality

- Why would we design and generate a 3D object?
  - Not just to look at it, but to use it! Not enough to just "look right"

Challenge #2: affordance and functionality

- Why would we design and generate a 3D object?
  - Not just to look at it, but to use it! Not enough to just “look right”
  - It is not about *what* it is, e.g., to have the right parts and be recognizable by a CNN, but what it can do and can afford …



The ultimate goal is not appearance, but functionality!
Generated 3D shapes/scenes need to function properly

CNN with max pooling

Learning functionality is challenging

- Functionality is contextual: defined by interactions between a 3D object and other objects, the agents, e.g., humans

Learning functionality is challenging

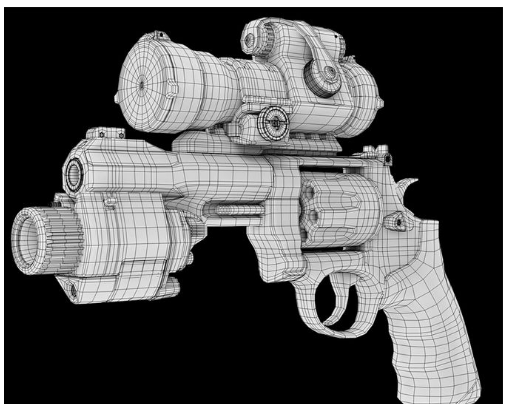
- Functionality is contextual: defined by interactions between a 3D object and other objects, the agents, e.g., humans
  - Interaction contexts harder to collect, describe, and generate
  - Considerably less 3D data have functionality annotations
  - How to define a “differentiable functionality loss”?

Most fundamental: representation challenge

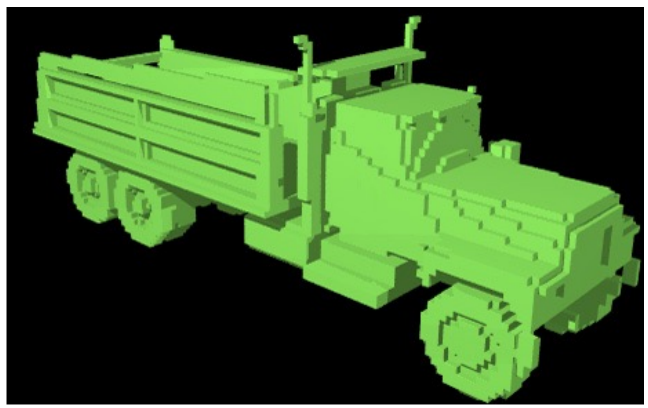
- Unlike images or speech, there is no universally accepted representation or encoding for 3D shapes

Challenge #3: representation challenge

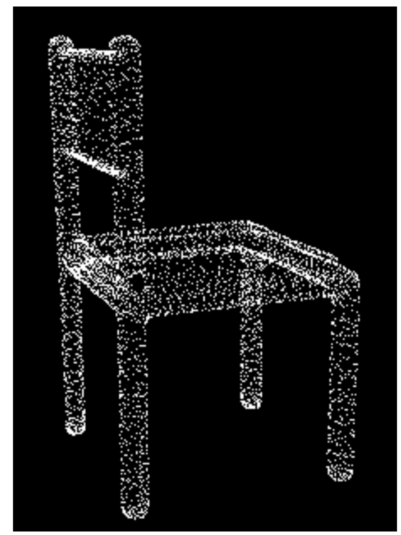
- Unlike images or speech, there is no universally accepted representation or encoding for 3D shapes
- Alternatives: low-level representations



Mesh: a set of triangles



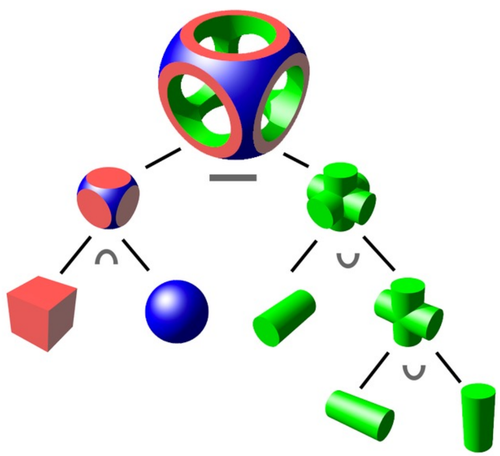
Volume: a grid of voxels



Point cloud: a set of points

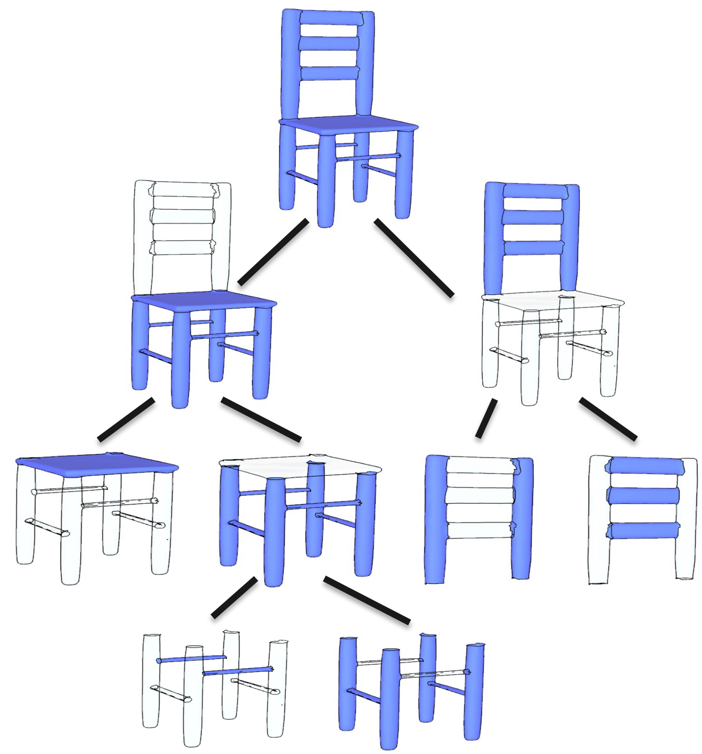
Challenge #3: representation challenge

- Higher-level representations



Procedural: e.g., CSG rep

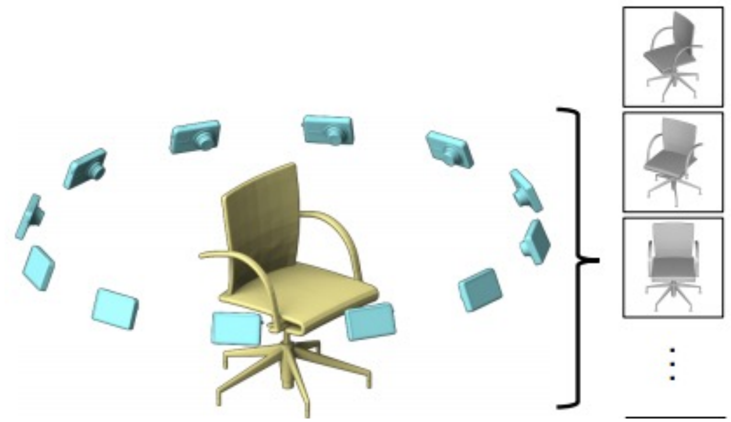
Structural: a set of parts



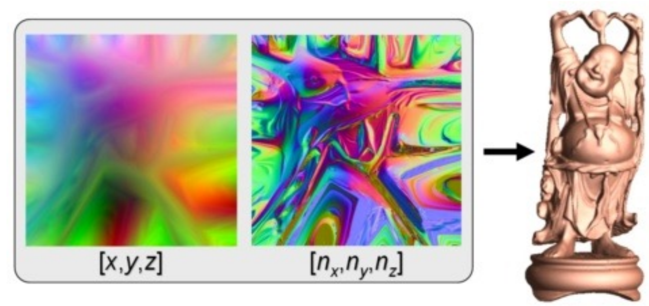
Hierarchical
organization of parts

Challenge #3: representation challenge

- Parameterized representation through mapping



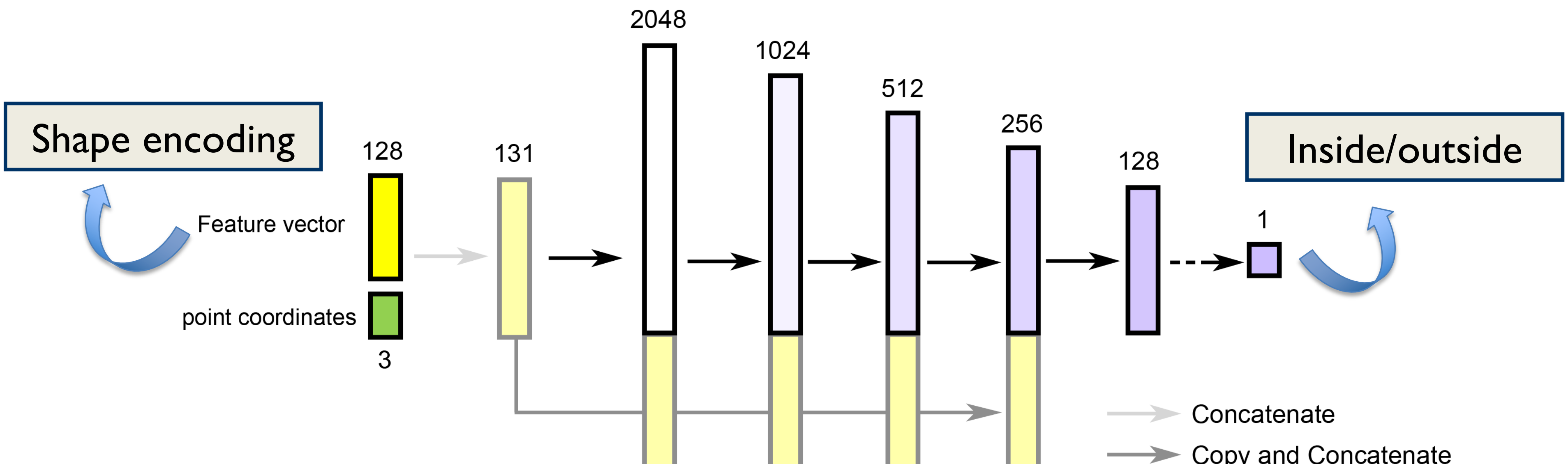
Multi-view images
in MVCNN [Su et al. 2015]



Geometry images [Sinha et al. 2016]

Recent wave of neural models for implicit reps

- Learn mapping from a 3D point (x, y, z) to inside/outside status or signed distance function (SDF) with respect to a 3D shape



The implicit field decoder learns shape boundaries

How has 3D shape generation been done?

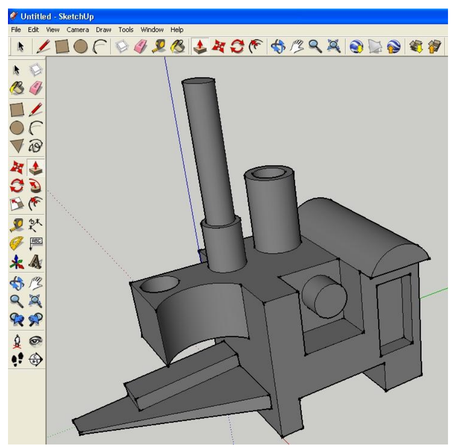
Traditional modeling paradigms in graphics

- Model-driven and interactive (human-in-the-loop)
- Human defines/influences the rules/procedures

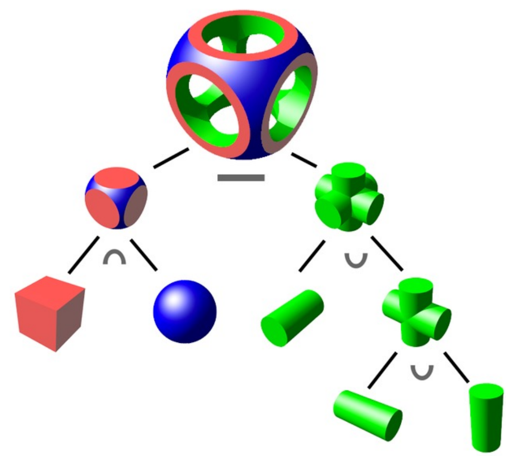
Traditional modeling paradigms in graphics

- Model-driven and interactive (human-in-the-loop)
- Human defines/influences the rules/procedures
- Examples:

Sketch-based modeling
[Igarashi et al. SIG 1999]



+Extrusion (SketchUp)



Procedural modeling

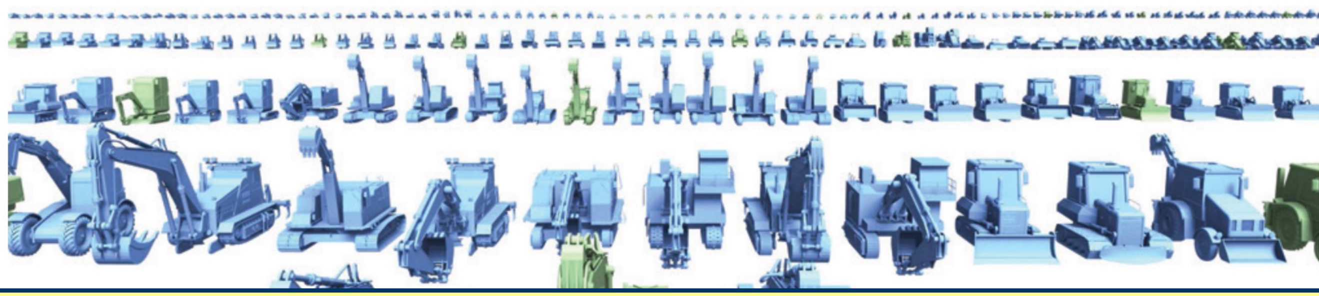
Can machines learn to generate 3D shapes?

Where does the machine learn from?

- Learn model generation from data or examples
- Shifting from model-driven to data-driven
- Two basic model generation paradigms

Paradigm #1: "more of the same"

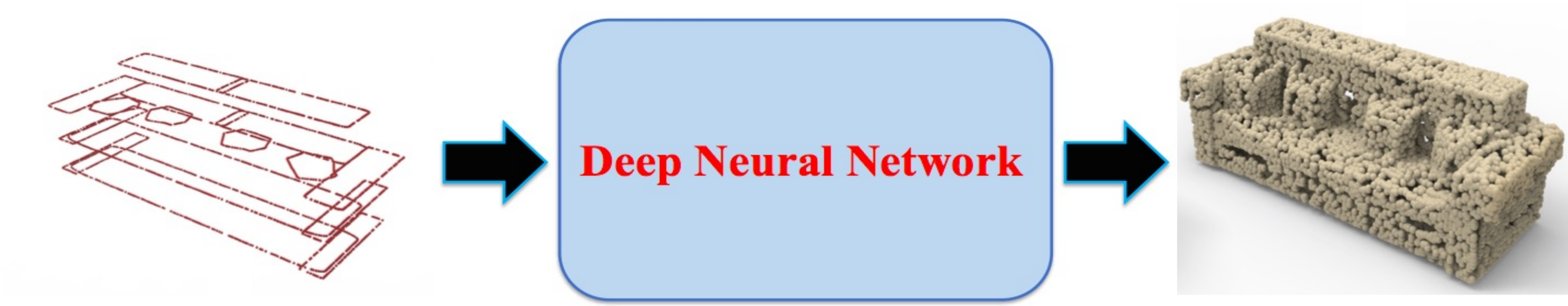
- Input: set of examples with commonality, e.g., all tractors
- Learn to generate more of the "same" (but with novelty)



Key: learn a space or manifold or distribution spanned by the examples. Then sample or traverse the space to generate novel 3D shapes.

Paradigm #2: "generate from X"

- Input: sets of examples from two domains X and Y
- Learn to generate target 3D shapes in Y from inputs in X



Key: learn a mapping or regression model

Approaches to "generate from X"

- Earlier data-driven methods: retrieve-and-adjust

[Xu et al. SIG 2013]

Sketch-to-scene

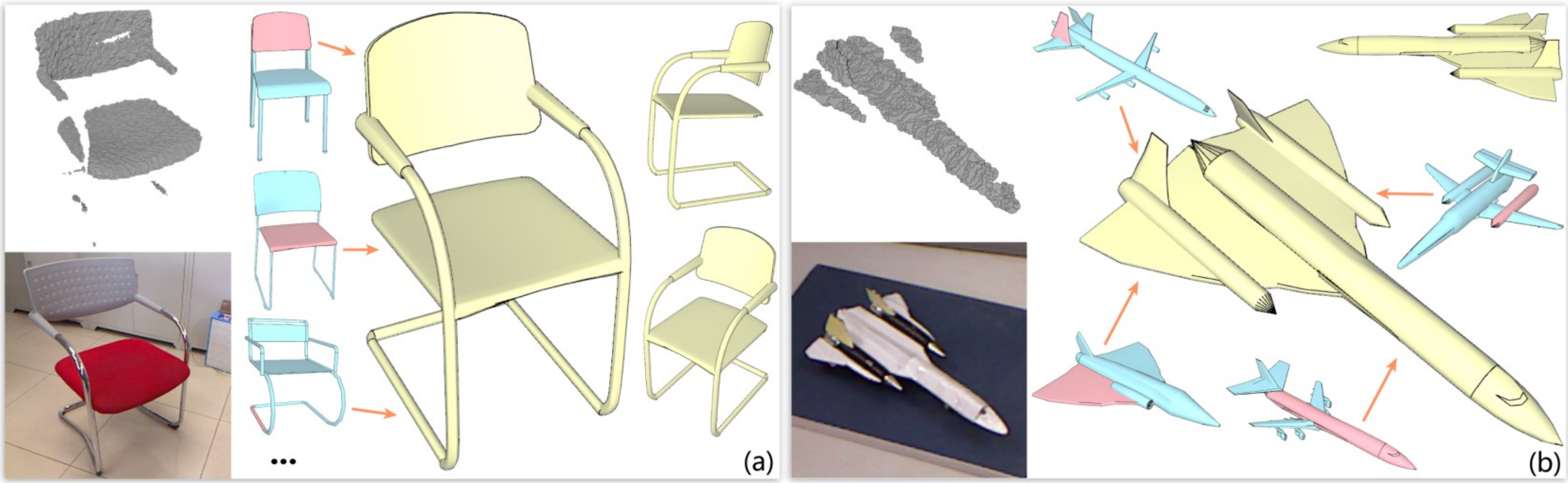
Approaches to "generate from X"

- 3D model generation from a single photograph

[Xu et al. SIG 2011]

Approaches to “generate from X”

- Model generation from single depth scan + RGB image
- 3D model built by assembling parts from different shapes



[Shen et al. SIG Asia 2012]

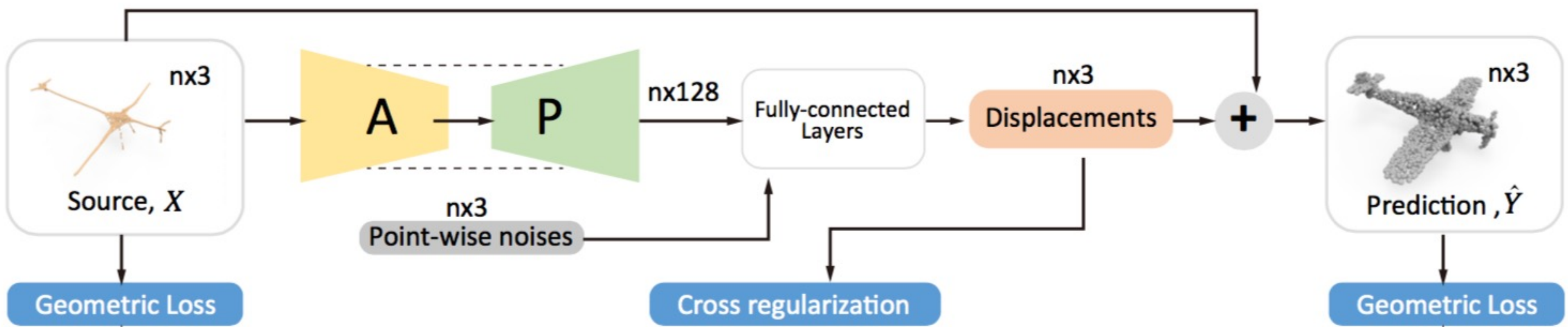
Retrieve-and-adjust approaches

- Similarity-driven retrieval followed by fitting and assembly
- Program does not really learn a general mapping
- Lack of novelty: generations do not deviate too much from database models

[Xu et al. SIG 2012]

Deep learning based methods

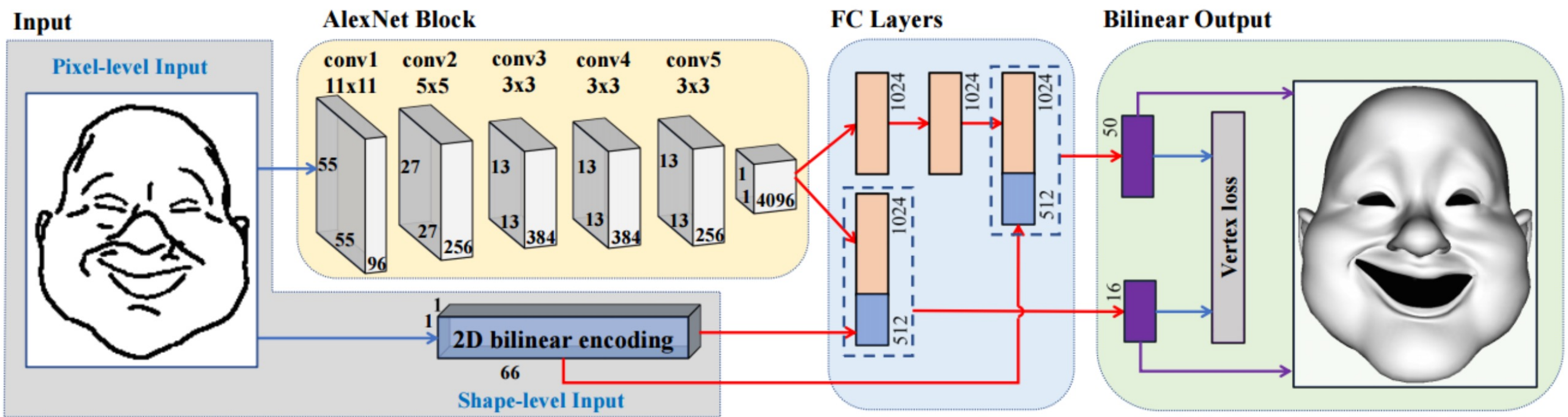
- Example: learn a general-purpose, non-linear mapping between two point sets, trained with paired data



P2P-NET [Yin et al. SIG 2018]

Another example: DeepSketch2Face

- Also trained with paired data: sketches and face meshes



DeepSketch2Face [Han et al. SIG 2017]

New challenge: unpaired training data

- Only available data are examples from domains X and Y
- Examples in X and Y are not matched up
- A more general setting as paired data can be unavailable

Photograph Monet Van Gogh

CycleGAN [Zhu et al. ICCV 2017]

“Shooting two birds with one stone”

- With same framework, train two mappings simultaneously
- Two translators (X→Y and Y→X): duals and form a cycle

"Shooting two birds with one stone"

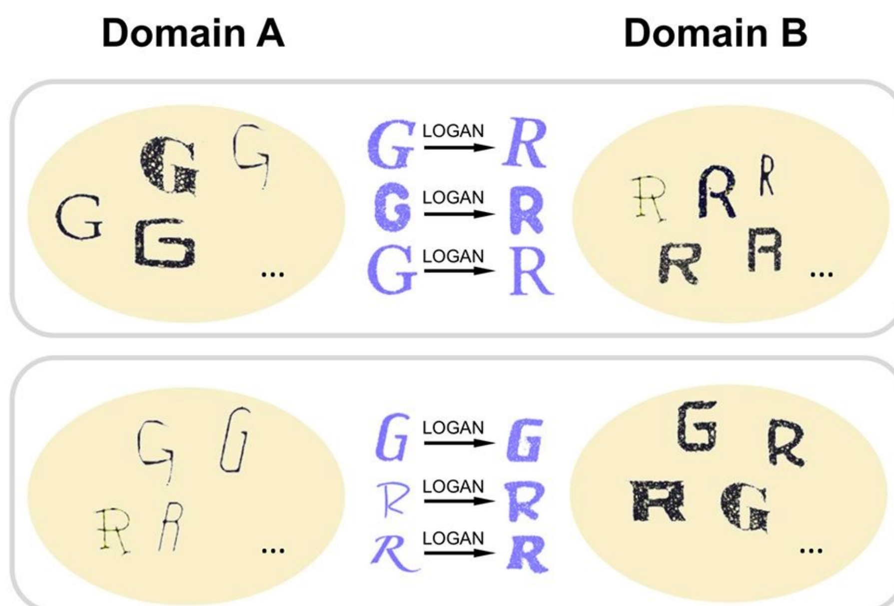
- With same framework, train two mappings simultaneously
- Two translators (X→Y and Y→X): duals and form a cycle
- Map from X to Y and back to X: loss to be measured in only one domain, e.g., a cycle consistency loss

Exciting new direction: unsupervised or weakly supervised domain translation with unpaired data. Most works on image-to-image translation and mainly style transfer.

LOGAN: unpaired shape-shape transform

LOGAN: Unpaired Shape Transform in Latent Overcomplete Space

LOGAN [Yin et al. SIG Asia 2019]

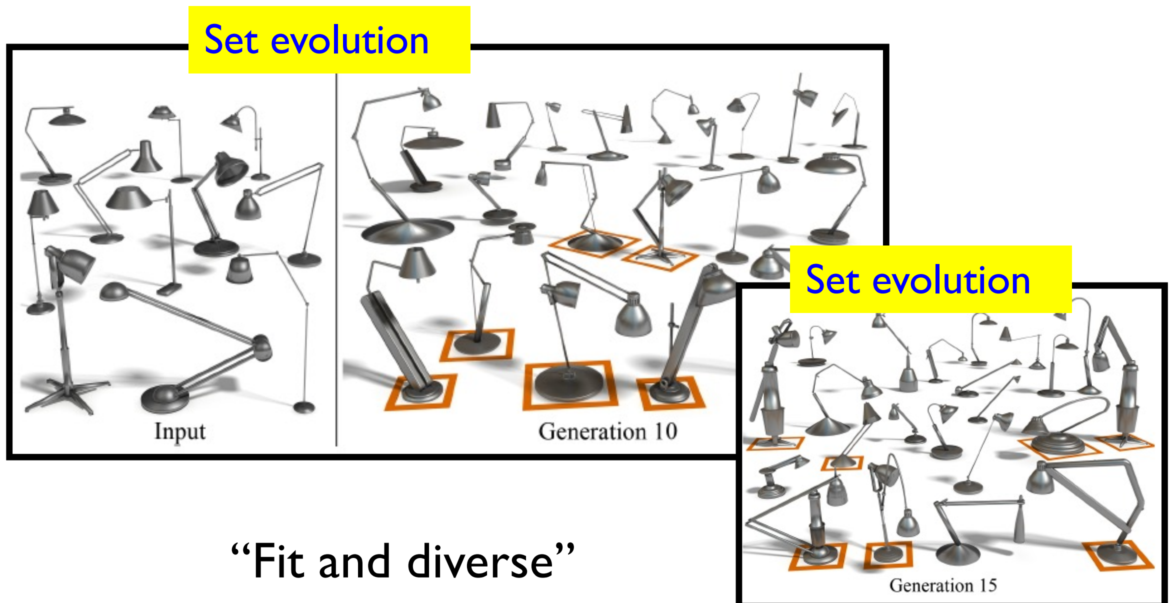


Style or content completely determined by the two input domains

Approaches to "more of the same"

- Earlier methods: mix-and-match or part (re-)composition

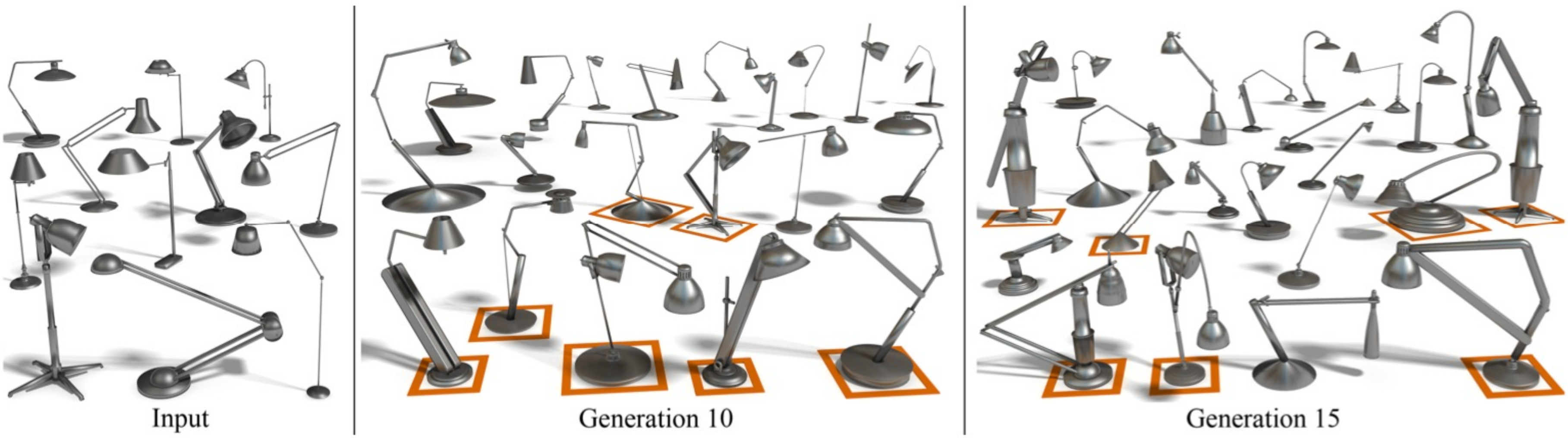
Modeling by example
[Funkhouser et al. 2004]



"Fit and diverse"
[Xu et al. SIG 2012]

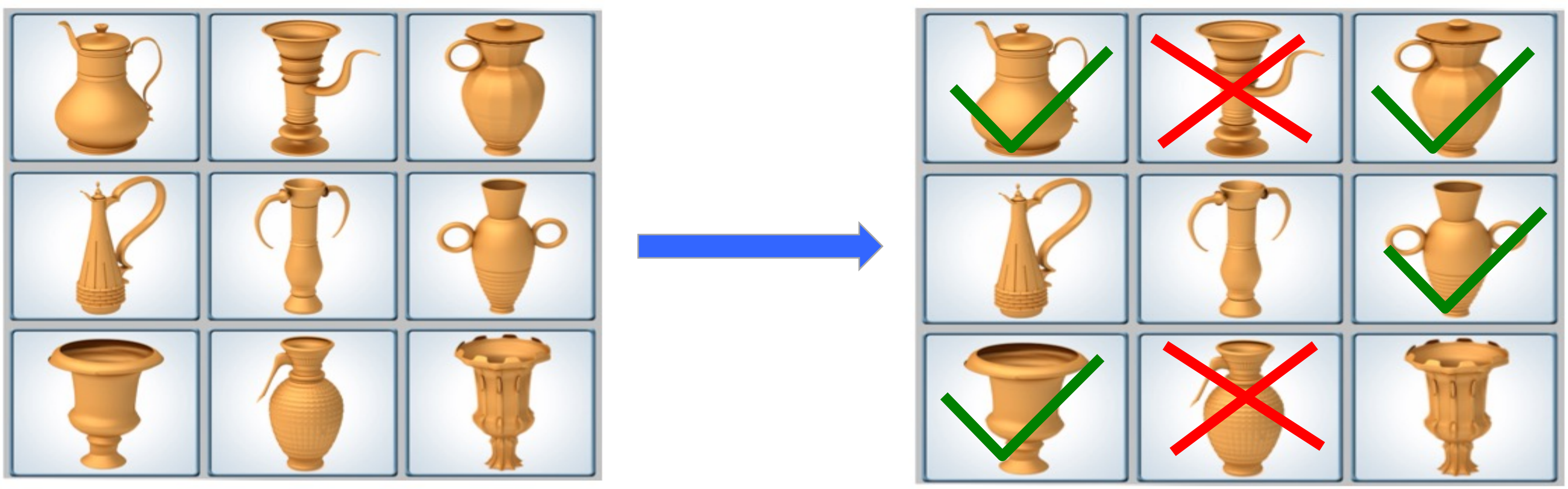
“Fit and diverse” for creative modeling

- Evolves an entire set of 3D models to obtain generations of fit and diverse new offsprings



“Fit and diverse” for creative modeling

- Creativity: machines stochastically generate models
- Control: by humans operating on a “design gallery”



Creative 3D modeling: evolution

- Fit = plausibility, e.g., from chairs to chair-like shapes
- Diversity = "surprising" designs to not stuck in an elite population — the elites do not survive well

Creative 3D modeling: evolution

- Fit = plausibility, e.g., from chairs to chair-like shapes
- Diversity = “surprising” designs to not stuck in an elite population — the elites do not survive well
- Executed via stochastic cross-over (part exchange)

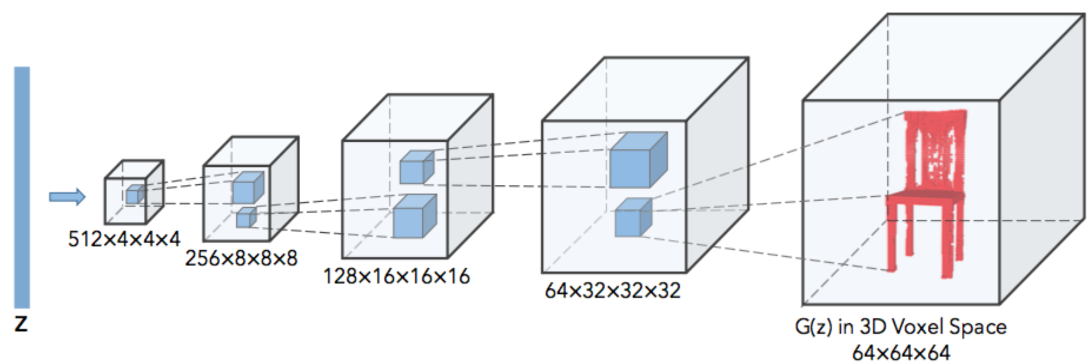
Mix-and-match approaches

- Similarity-driven part substitution within a shape collection
- Machine does not really learn any space/manifold

“Fit and diverse” [Xu et al. SIG 2012]

3D generative adversarial network (3D-GAN)

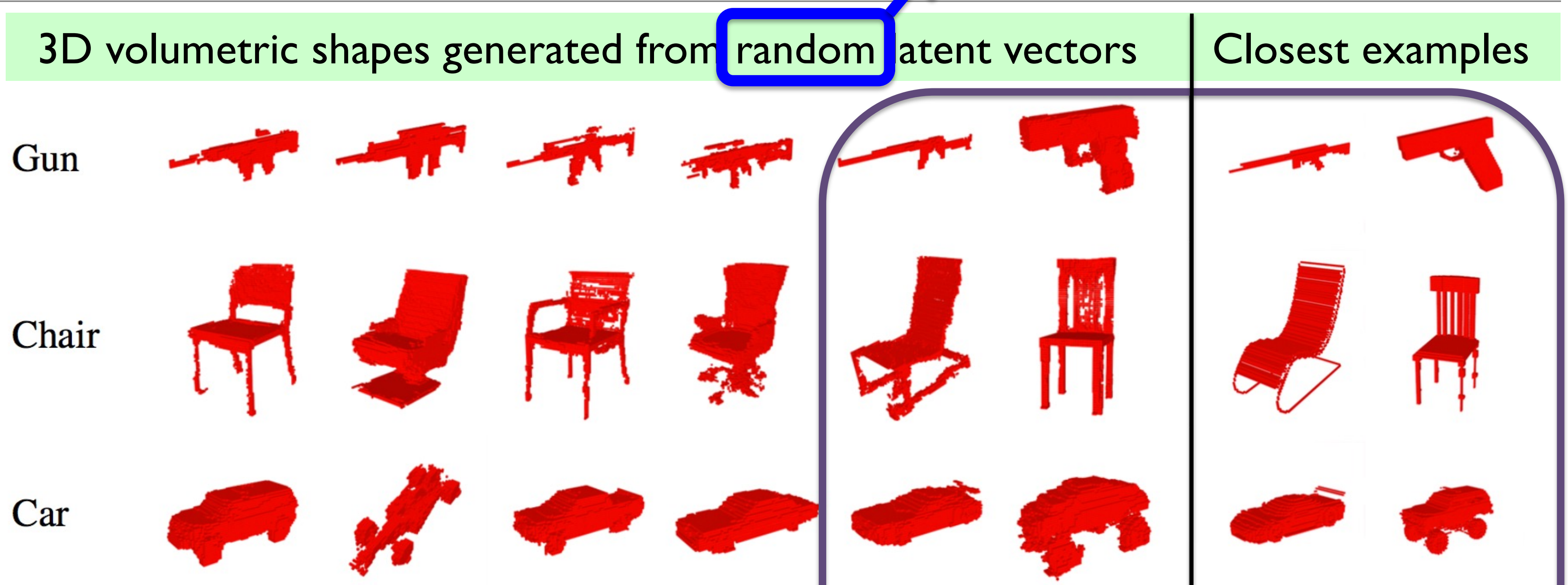
- 3D shape as voxels: combine volumetric CNN and GAN
- Generator maps 200D latent vector to 64x64x64 volume
- Discriminator classifies real objects vs. generator outputs



3D-GAN [Wu et al. NIPS 2016]

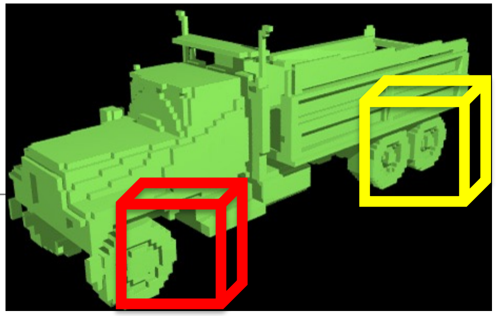
3D-GAN results

Generate novel shapes



“More of the same”: fooling the discriminator = making the generated outputs similar to training examples

3D-GAN



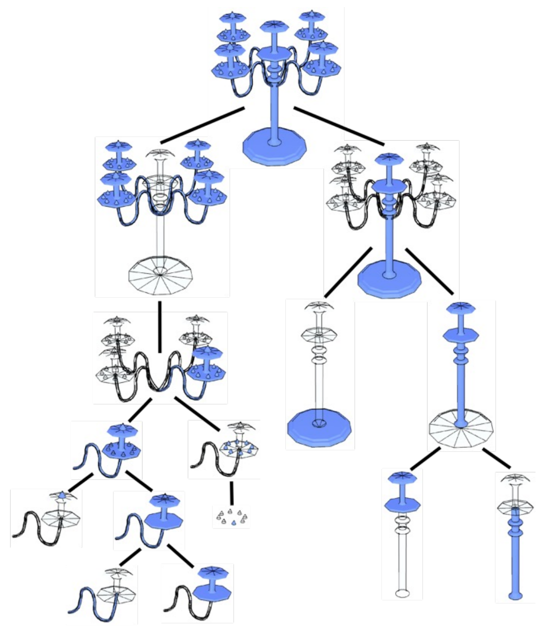
- Volumetric CNN is not structure-aware
- Results: low-res “blobs” of voxels; no clean separation of object parts; not reusable for subsequent modeling

- In real life, 3D objects are not build at voxel (but part) level
    - Think IKEA furniture or most current manufacture process

Symmetry hierarchies (SYMH)

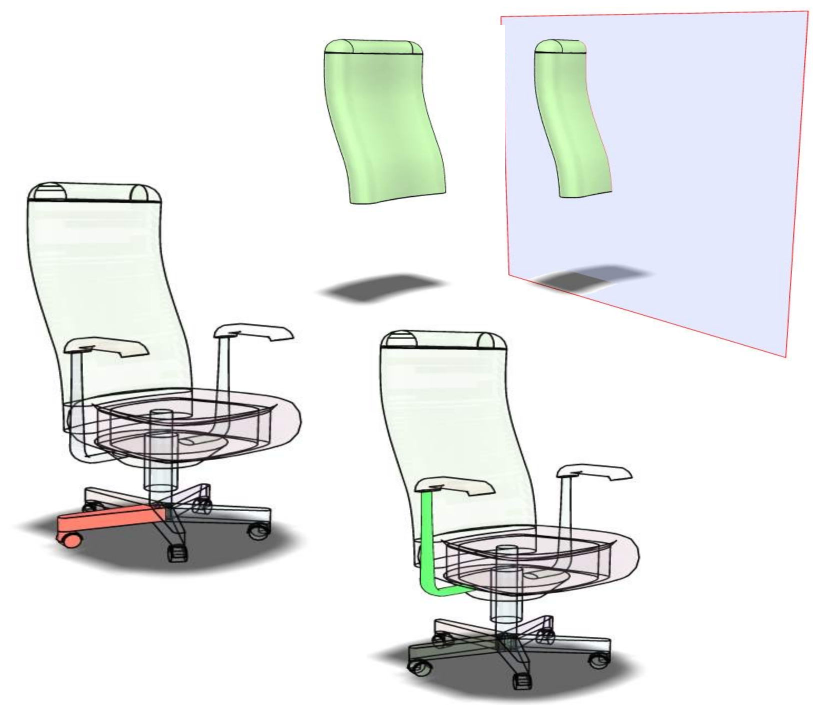
[Wang et al. EG 2011]

Symmetry hierarchy: symmetry guides grouping and assembly of shape parts to form a meaningful hierarchical part organization.



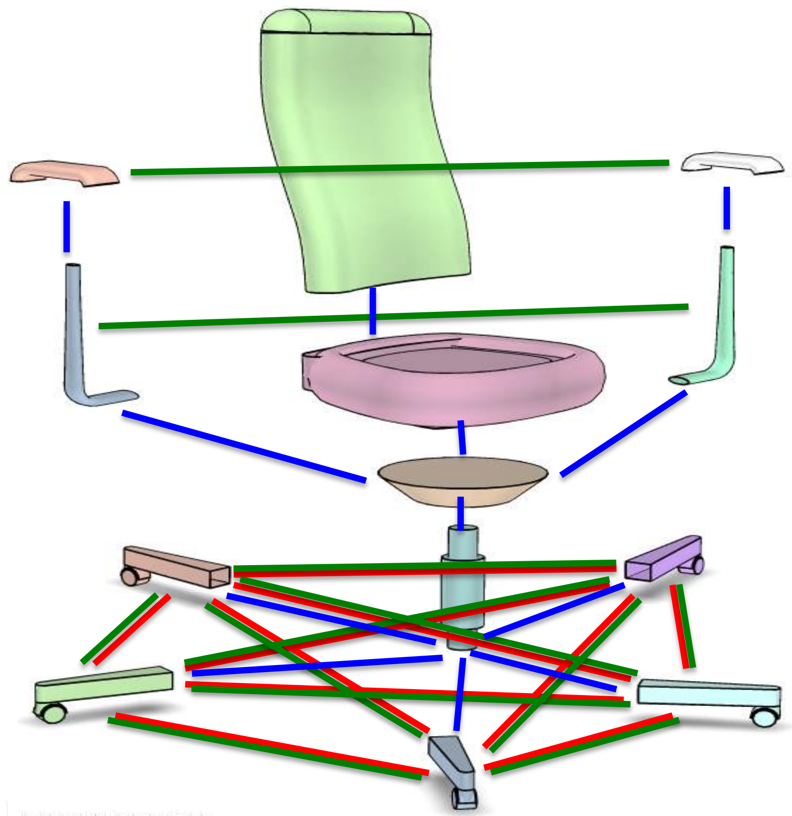
SYMH construction

Part segmentation



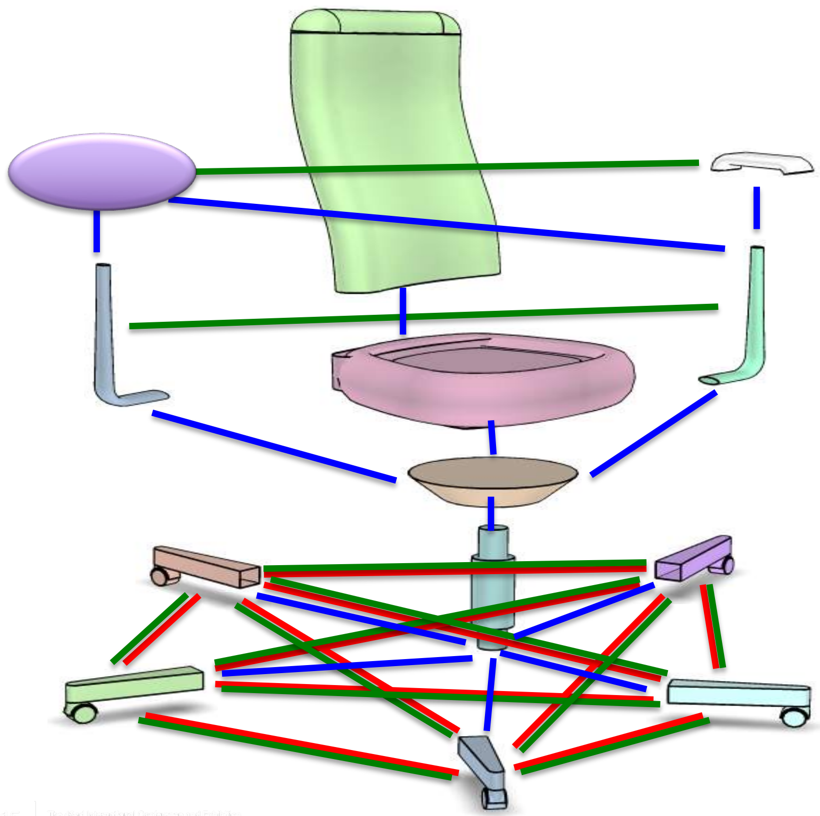
Symmetry detection

Initial graph



Rotational symmetry
Reflection symmetry
Connectivity

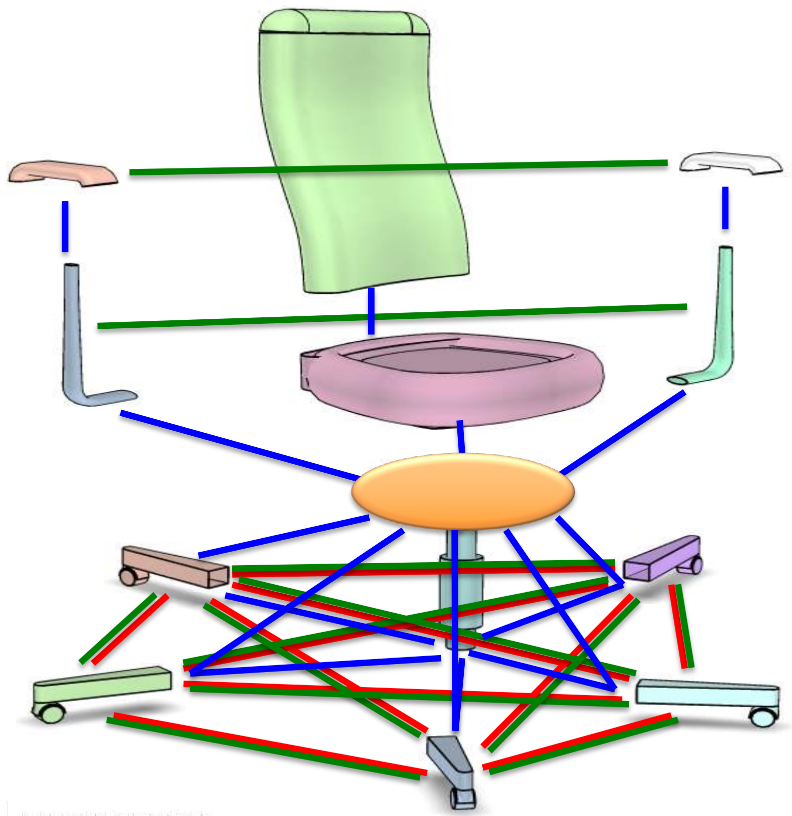
Bottom-up graph contraction



Two operations:

Grouping by symmetry

Bottom-up graph contraction



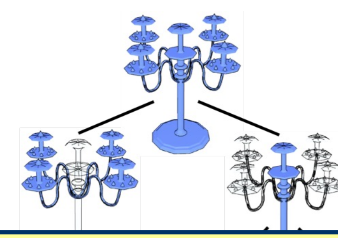
Two operations:

Grouping by symmetry

Assembly by proximity

SYMH: a fundamental shape representation

- Structure-aware: hierarchical part organization
- Functionality-aware (just a bit): symmetric parts tend to perform the same function
- Decouples structure and (part) geometry



Can SYMH be generative?

A good idea: with SYMH, we can decouple the learning and generation of shape structure and part geometry

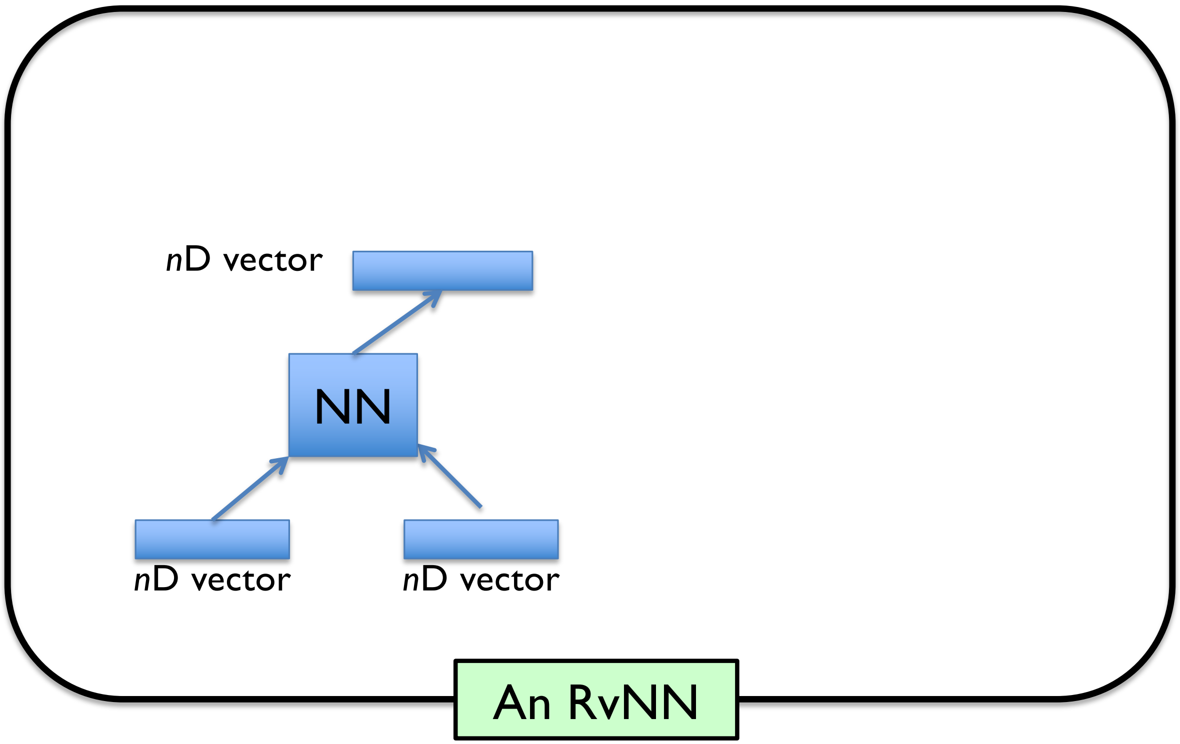
Can neural nets (NNs) be trained to learn SYMH?

- Can we encode or “vectorize” SYMHs for NN processing?
- Can traditional convolutional NNs work for SYMHs?

SYMH is a structural shape representations (an organization of parts). We need a different kind of NN.

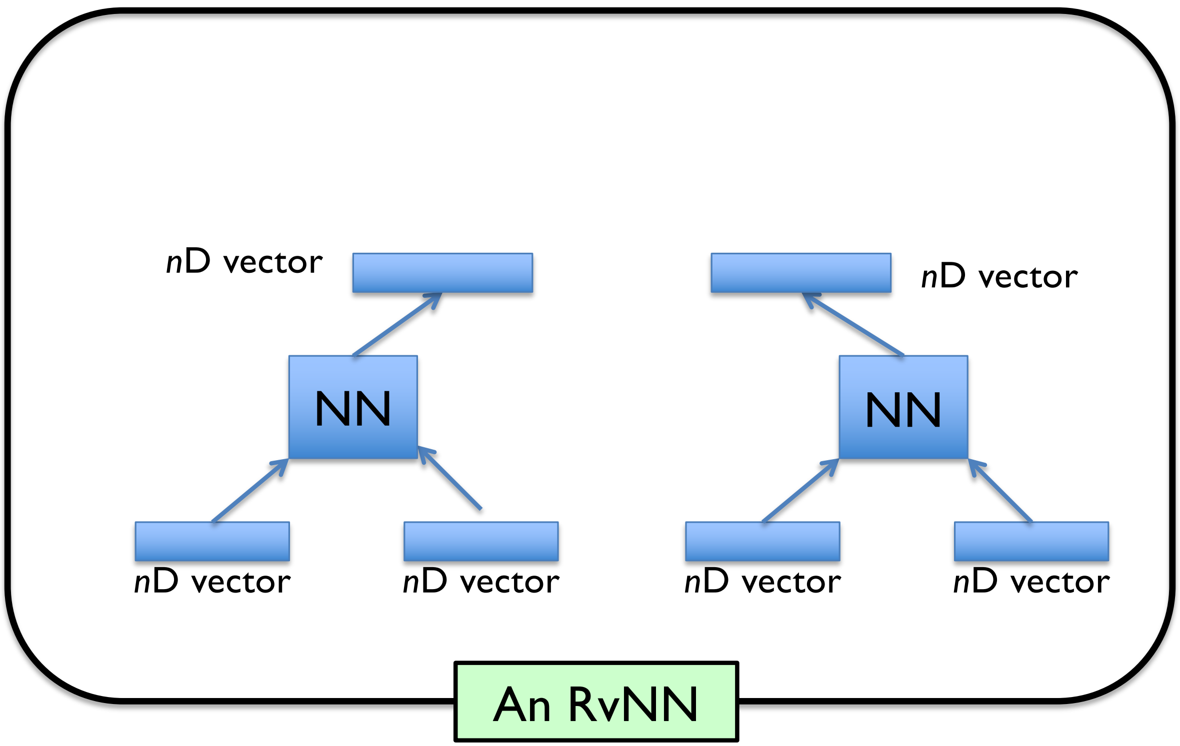
Recursive neural network (RvNN ≠ Recurrent NN)

- A tree structure where each node is a neural network



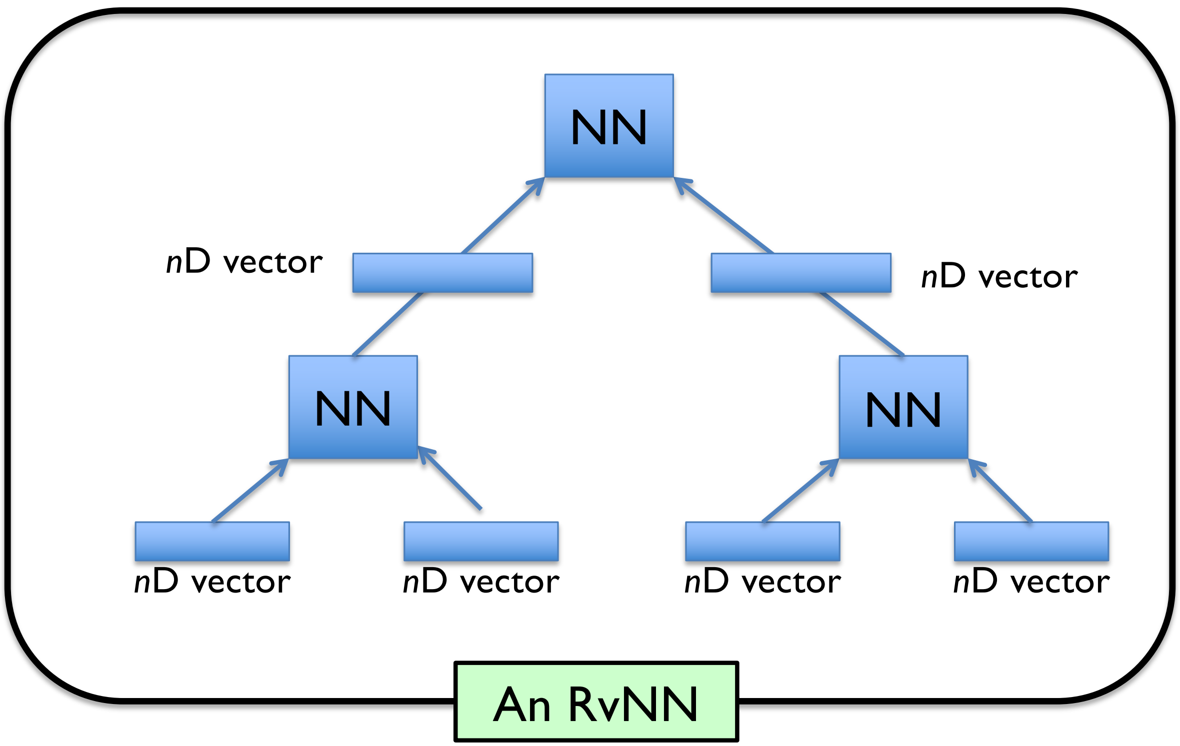
Recursive neural network (RvNN ≠ Recurrent NN)

- A tree structure where each node is a neural network



Recursive neural network (RvNN ≠ Recurrent NN)

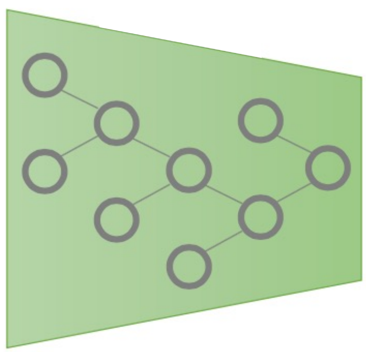
- A tree structure where each node is a neural network



Learn SYMH using RvNNs

- A shape structure is represented by an arrangement of boxes
- Each box is encoded as a fix-dim vector: leaves of the SYMH

Box arrangement

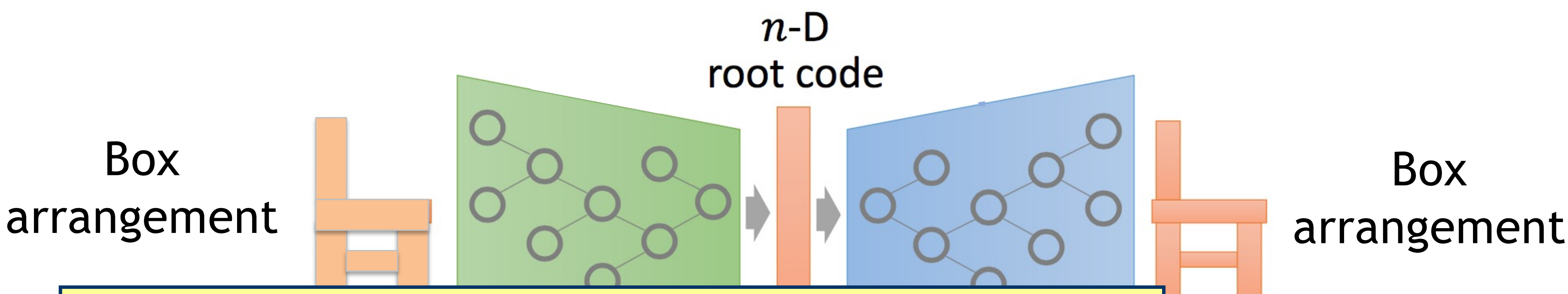


RvNN

How to define the network loss?

RvNNs into recursive autoencoder (RAE)

- RAE encoder turns box arrangements into a root code, recursively
- RAE decoder turns a code into a SYMH, recursively
- Network loss is the reconstruction loss summed over boxes

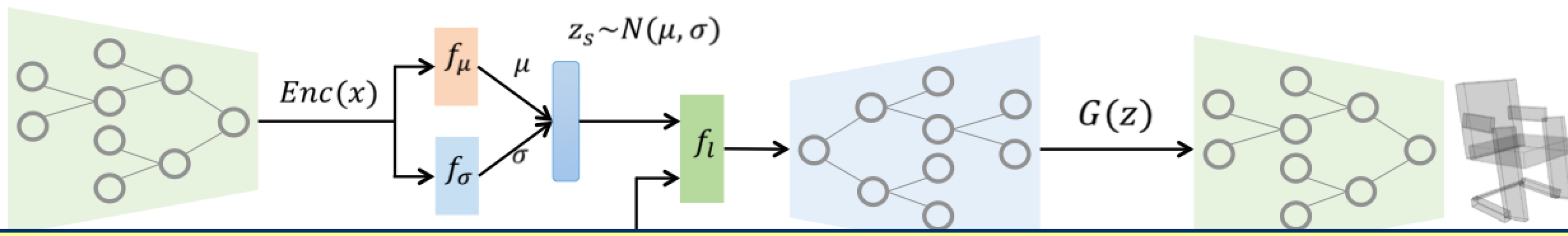


RAE decoder can be refined into a generative model, allowing generation of new SYMHs

[Li et al. SIG 2017]

Generative Recursive Autoencoder: GRASS

- Change AE loss to GAN loss to learn a manifold of plausible codes
- Part geometry is learned by yet another neural network
- Generation: sample root code → SYMH → fills in part geometry

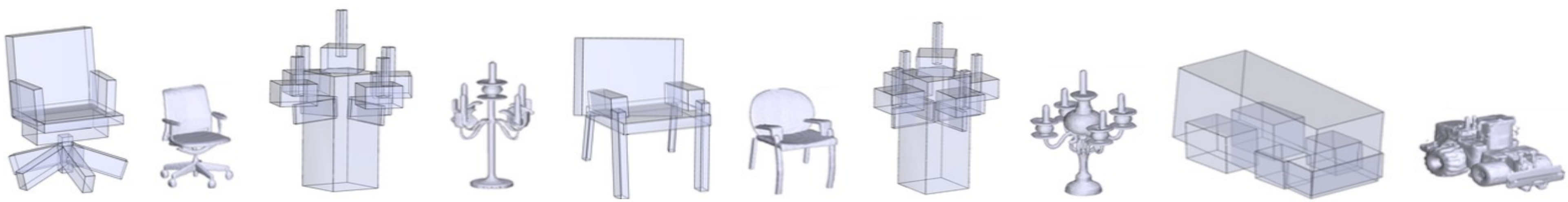


Key idea again: de-couple generation of shape structures (SYMHs) and generation of shape geometries.

3D shape generation results

[Li et al. SIG 2017]

- First neural network to learn multi-attribute structural graphs
- Coarse-to-fine synthesis: structure-aware; high-res; clean parts



Results from voxel-based 3D-GAN [Wu et al. NIPS 2016]

Main works covered

[Wang et al. 2011] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong, "Symmetry Hierarchy of Man-Made Objects", *Eurographics* 2011.

[Chen and Zhang 2019] Zhiqin Chen and Hao Zhang, "Learning Implicit Fields for Generative Shape Modeling", submitted to *CVPR* 2019.

[Xu et al. 2012] Kai Xu, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen, "Fit and Diverse: Set Evolution for Inspiring 3D Shape Galleries", *SIGGRAPH* 2012.

[Li et al. 2017] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas, "GRASS: Generative Recursive Autoencoders for Shape Structures", *SIGGRAPH 2017*.

[Yi et al. 2017] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong, "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation", to appear in *ICCV* 2017.

[Li, Patil, et al. 2018] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owis Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang, "GRAINS: Generative Recursive Autoencoder for INdoor Scenes", *accepted with minor revision, ACM TOG* 2019.

[Yin et al. 2018] Kangxue Yin, Hui Huang, Daniel Cohen-Or, and Hao Zhang, "P2P-NET: Bi-Directional Point Displacement Net for Shape Transform", *SIGGRAPH* 2018.

[Yin et al. 2019] Kangxue Yin, Zhiqin Chen, Hui Huang, Daniel Cohen-Or, and Hao Zhang, "LOGAN: Unpaired Shape Transform in Latent Overcompelte Space", *SIGGRAPH Asia* 2019.