

Neural Network Basics and Representation Learning for 3D Shapes

Richard (Hao) Zhang

CMPT 464/764: Geometric Modeling in Computer Graphics

Lecture 5

Acknowledgment: some images taken from Michael Bronstein's GDL slides; some from Stanford UFLDL Tutorial

Assumed background and level of coverage

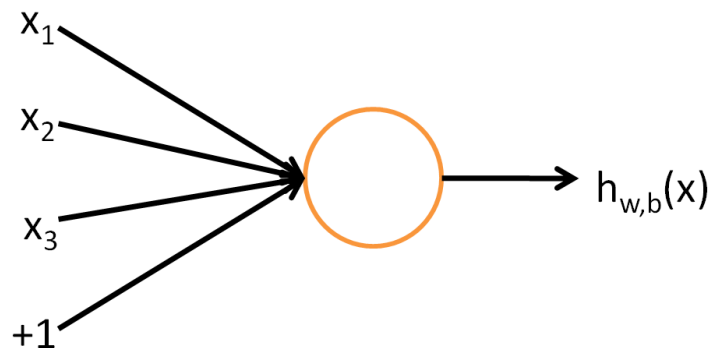
- Not ML class, so only one-class coverage on some basics
- Will not assume much past ML experience, if at all
- But you are expected to go through necessary references and online material (LLM) for assignments and project
- Focus more on representation learning of **3D shapes**
- Focus more on **generative** models, what graphics is about

What sort of learning are we talking about?

- Fundamentally, mimic or simulate how our brain works
 - To start, **neurons** as elementary computational units
 - Use of **artificial neural networks**, which have a long history tracing back to at least early 1990's
- Most ML today utilize neural networks (NNs) in some way: deep learning uses **deep neural networks**
- Modern NNs, e.g., for LLMs, may not be brain-inspired

Start with neural network basics

- Let us start with a single neuron to model a **perceptron**,

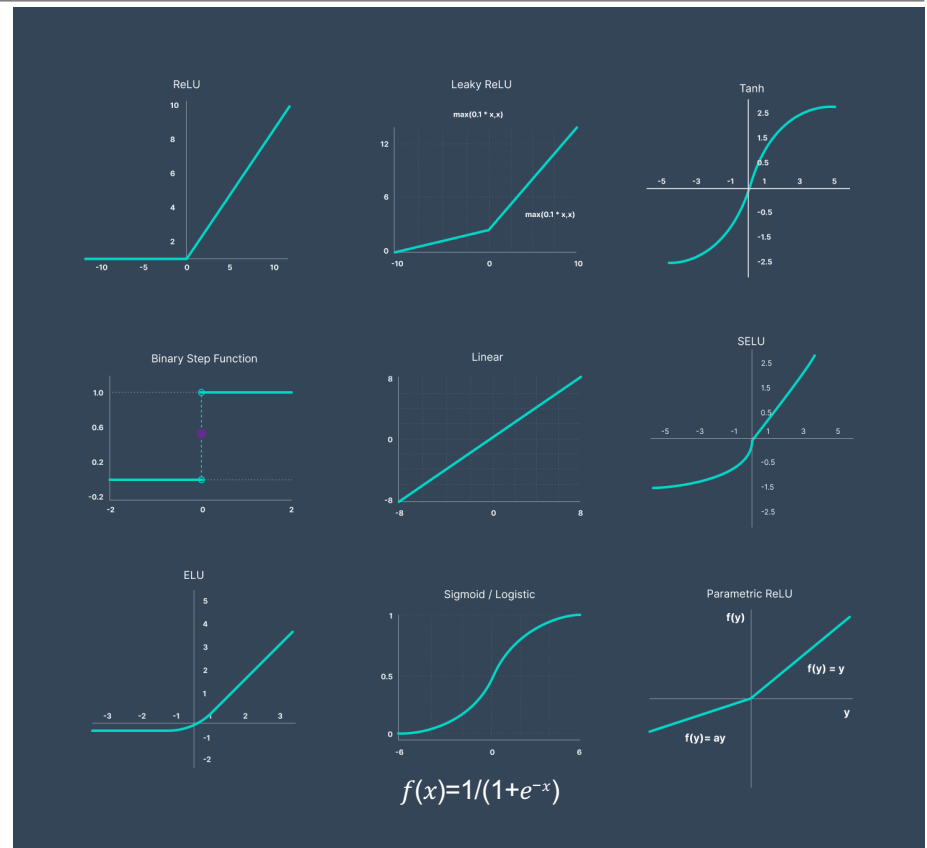


- $h_{w,b}(x) = f(W^T x) = f(\sum_{i=1}^3 W_i x_i + b)$, where $f : \mathfrak{R} \mapsto \mathfrak{R}$ is called an **activation function**, typically **non-linear and non-polynomial**

Activation functions

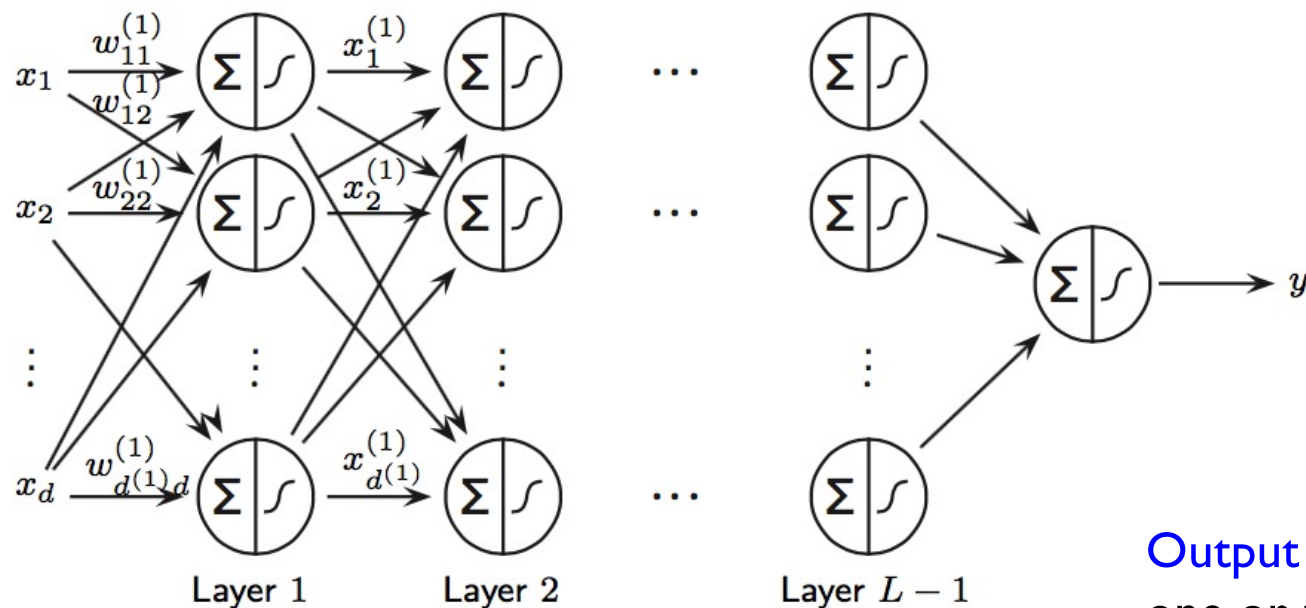
- Desirable properties:
 - **Nonlinearity**: to model real-world phenomena
 - **Non-polynomial**: universal approximation (see Wiki)
 - **Continuity**
 - **Monotonicity**

Among others



Multi-Layer Perceptron (MLP)

- A **feed-forward** network: no loops or cycles

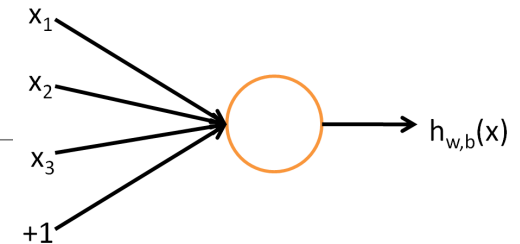


Input layer

Hidden layers

Output layer with
one or more
output values

Computing the network weights



- Using **back-propagation** and **gradient decent**
- Given **training pairs** $\{(x^{(i)}, y^{(i)})\}$, $i=1, \dots, m$, optimize weights W , b to minimize the sum of a **data** term and a **regularization** term

$$\begin{aligned} J(W, b) &= \left[\frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \\ &= \left[\frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \end{aligned}$$

See: <http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/>

<http://neuralnetworksanddeeplearning.com/chap2.html>

Beyond supervised learning

- With the “training pairs”, we have supervised learning
 - The input-output pairs serve as training or ground truth (GT) data
 - Optimize neural network weights to minimize loss against target outputs
- Unsupervised learning
 - Still optimize neural network weights to minimize some loss
 - But loss definition does not need target or GT data: can “self-supervise”
- Weakly supervised learning, e.g., few- or one-shot learning
- Semi-supervised learning, e.g., active learning

Self-supervised learning (SSLs) in LLMs

Masking Tokens (as in BERT)

One of the key techniques in SSL for LLMs is **Masked Language Modeling (MLM)**, prominently used in models like **BERT** (Bidirectional Encoder Representations from Transformers). In MLM, a certain percentage of tokens (words) in the input text are randomly masked, and the model is trained to predict these masked tokens based on the surrounding context.

How it Works:

- **Random Masking:** During training, about 15% of the words in each sequence are replaced with a special `[MASK]` token.
- **Prediction Task:** The model tries to predict the original words that were masked out, using the information from the unmasked words on both sides of the masked token.
- **Bidirectional Context:** Since BERT considers both the left and right context, it effectively learns the relationships and dependencies between words in a bidirectional manner.

Next Token Prediction (as in GPT)

In contrast to BERT's bidirectional approach, **GPT** (Generative Pre-trained Transformer) models utilize an autoregressive method known as **Causal Language Modeling (CLM)**. The model is trained to predict the next word in a sequence, using all the words that come before it.

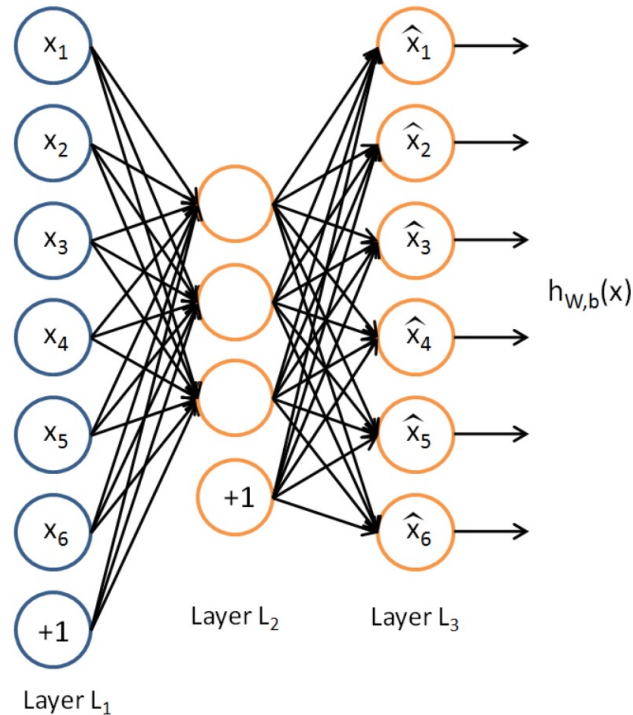
How it Works:

- **Sequential Prediction:** The model processes the input text in a left-to-right manner, predicting the next word at each step.
- **Context Utilization:** By focusing on previous words, the model learns to generate coherent and contextually appropriate continuations of text.



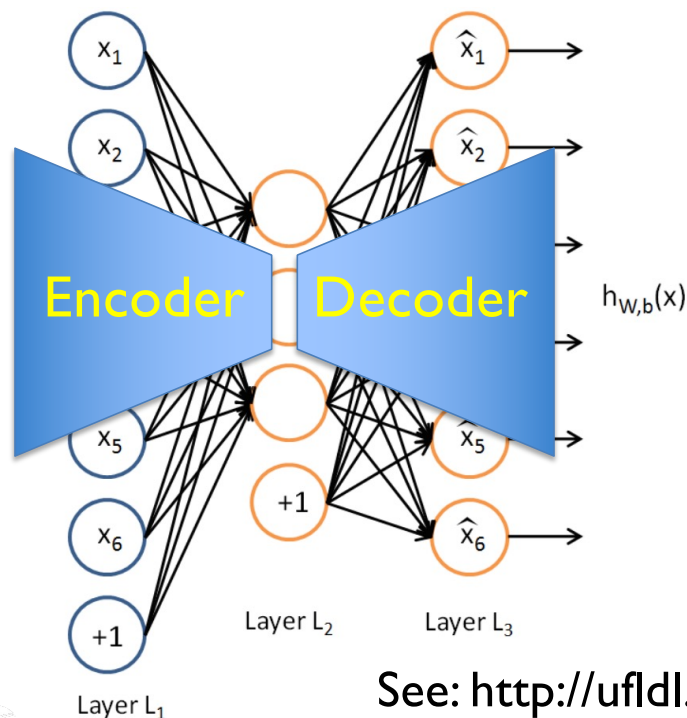
Unsupervised learning: Autoencoders (AEs)

- Network learns to **reconstruct the input**: learn **identity**



Autoencoder for unsupervised learning

- Network learns to **reconstruct** the input: learn **identity**

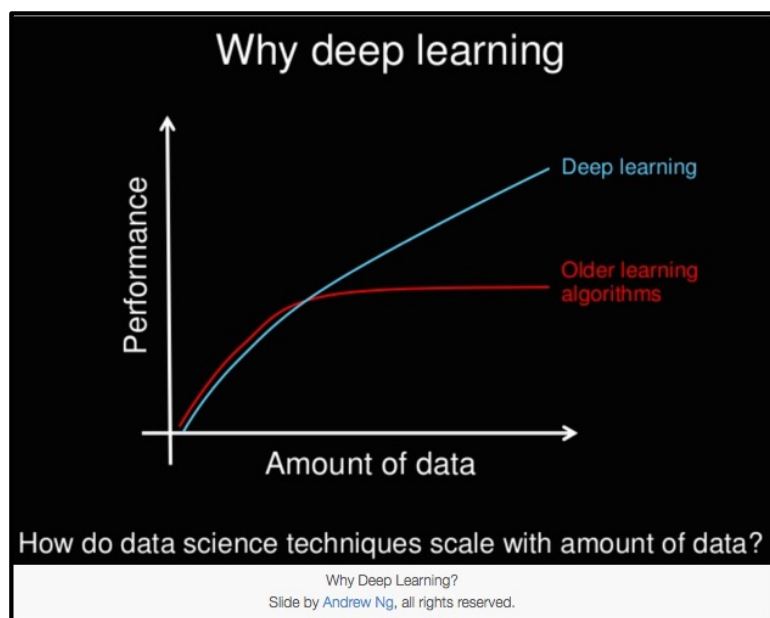


- By limiting the number of hidden units, it is forced to learn a **compressed** representation
- **Representation learning**
- **Dimensionality reduction**

See: <http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/>

What is **deep** learning?

- Use of **large** and **deep** (many layers) neural networks
 - Many hidden layers and many weights: often deep and wide



Important Property of Neural Networks

Results get better with

**more data +
bigger models +
more computation**

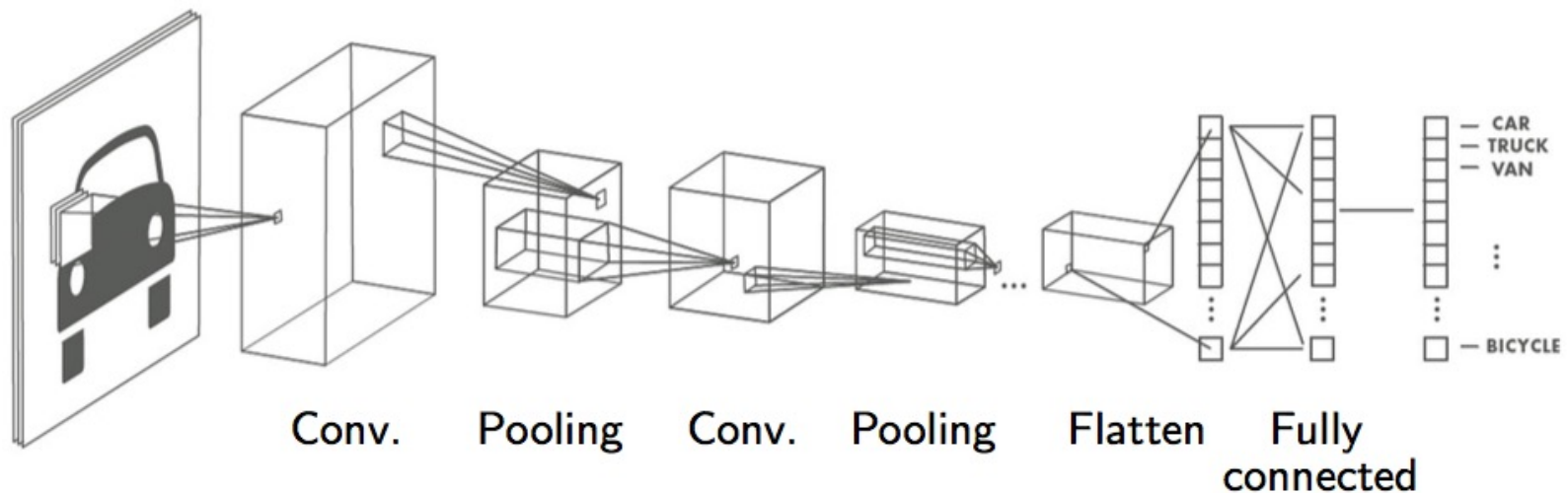
(Better algorithms, new insights and improved techniques always help, too!)



Results Get Better With More Data, Larger Models, More Compute
Slide by Jeff Dean, All Rights Reserved.

Convolutional neural network (CNN)

- A CNN designed for **object classification** from images

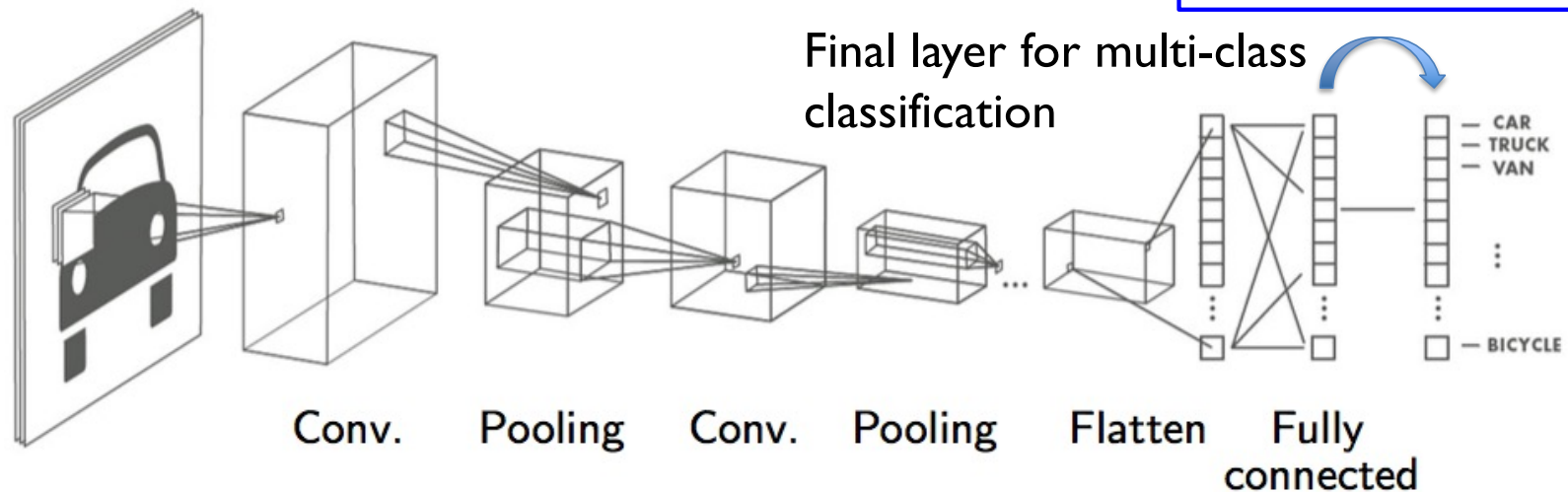


Convolutional neural network (CNN)

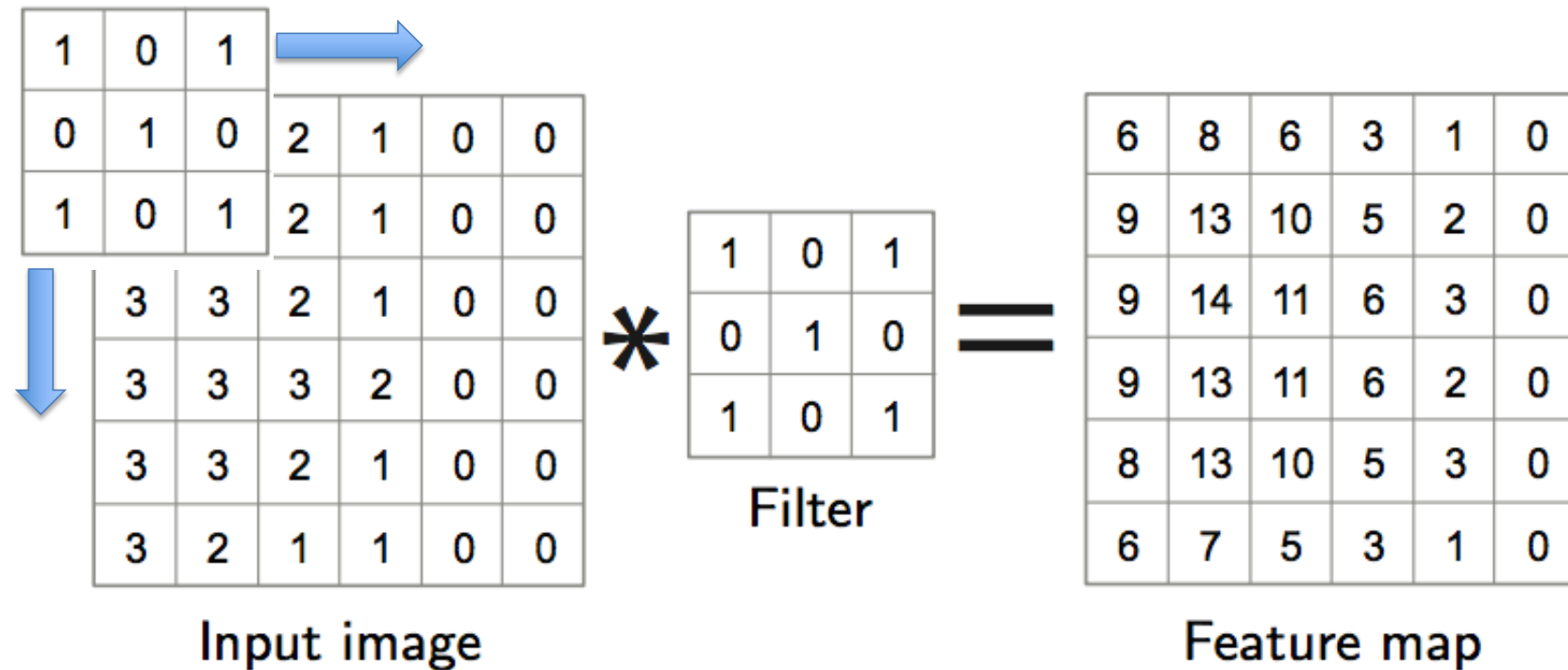
- A CNN designed for **object classification** from images

Softmax activation: converts a vector of numbers into a **probabilistic distribution**

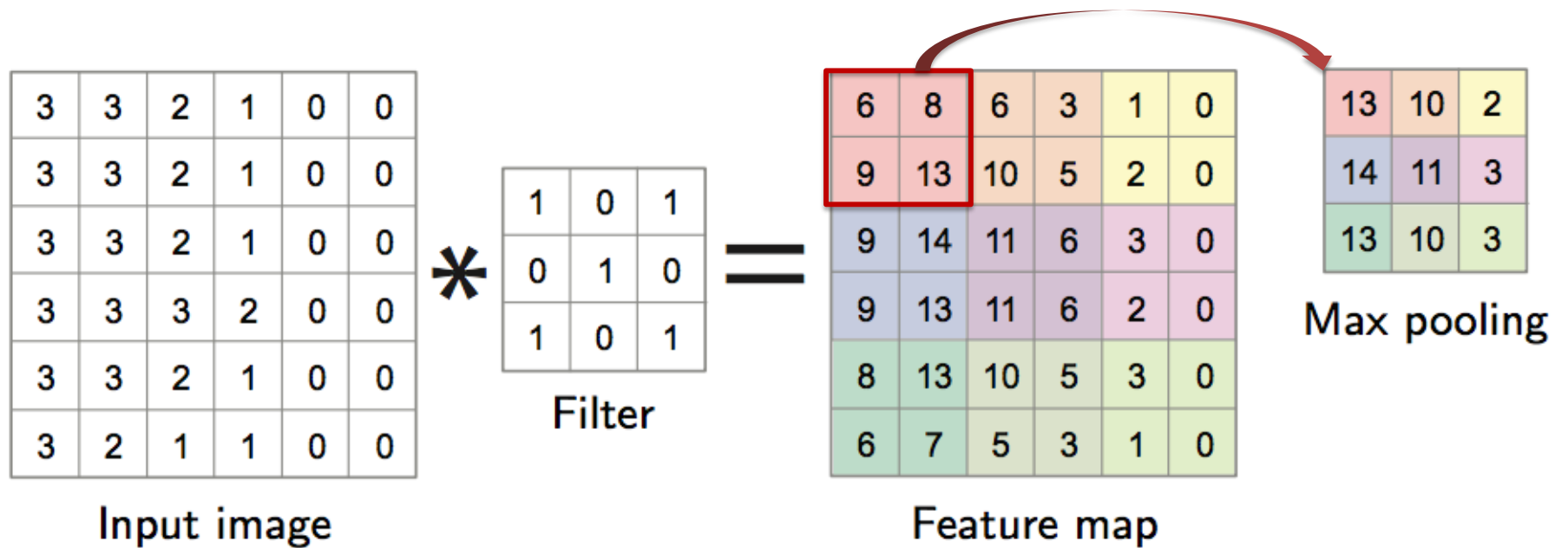
$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$



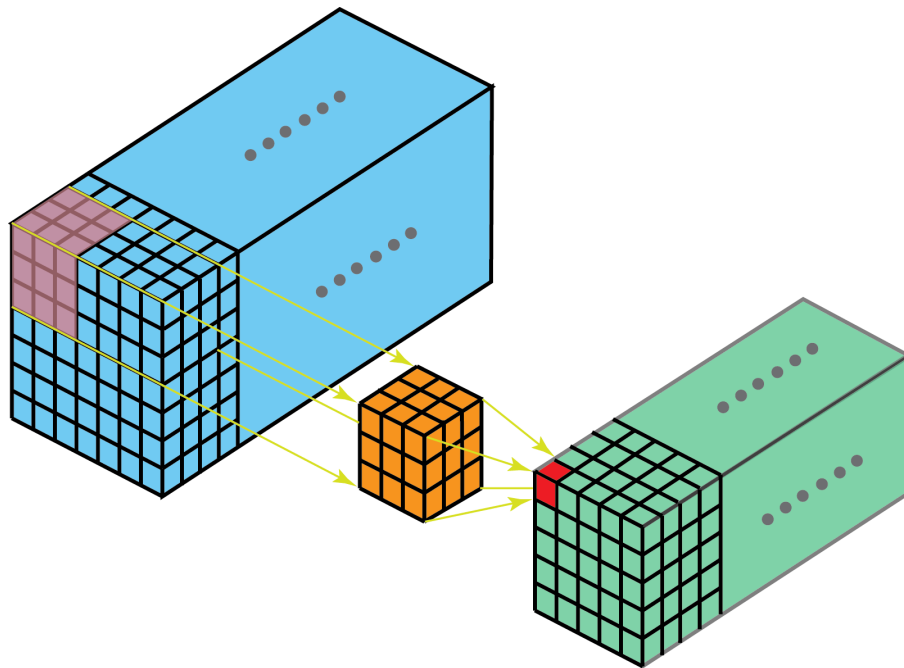
Convolution: a “running” average



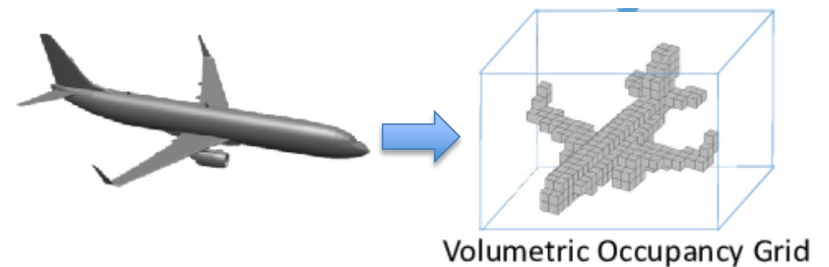
Pooling: summarization to reduce spatial res



Volumetric convolution: 3D CNN

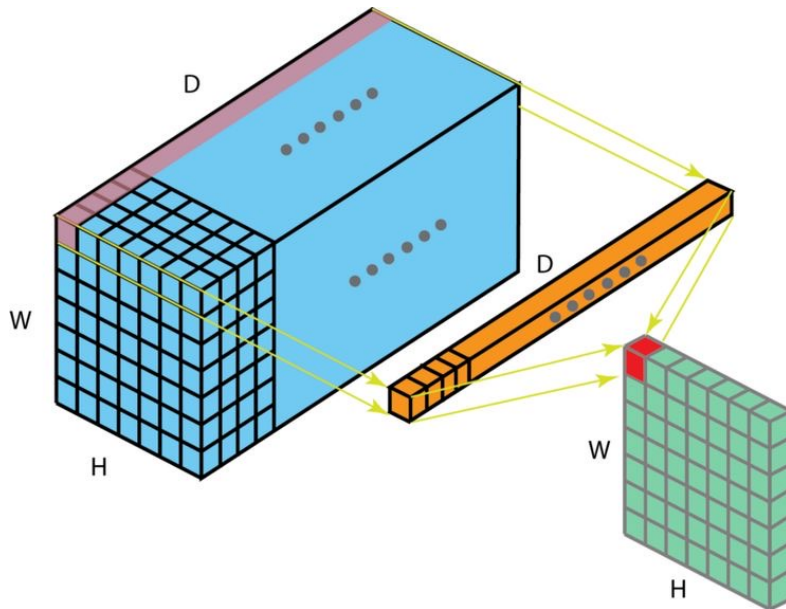


- Straightforward extension from image convolution
- Processes volumetric data, e.g., a 3D shape, or **multi-channel** image data



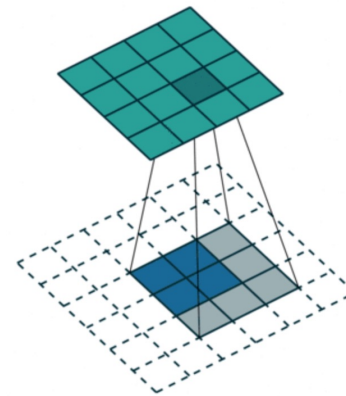
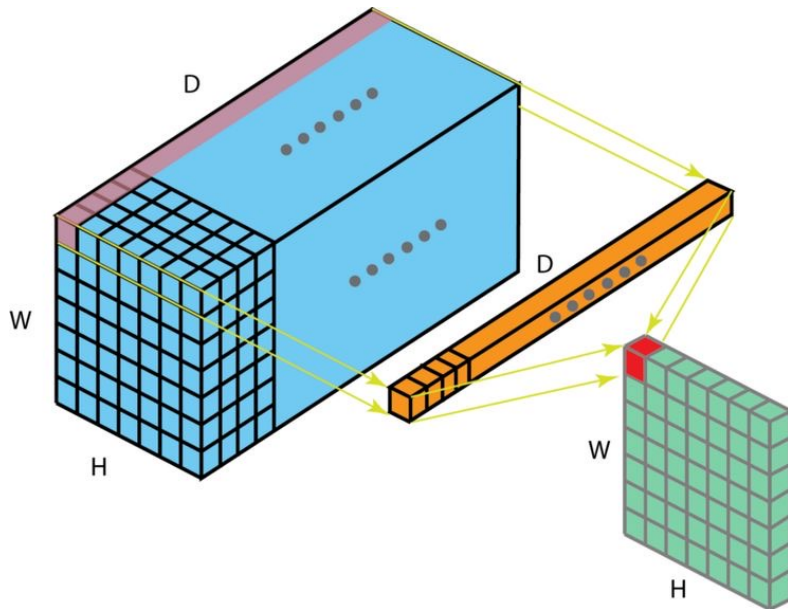
Other variants of convolution

- 1x1 convolution: reducing depth/channel resolution



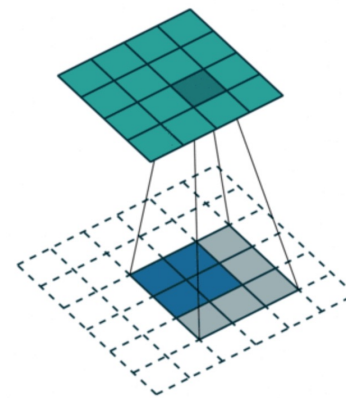
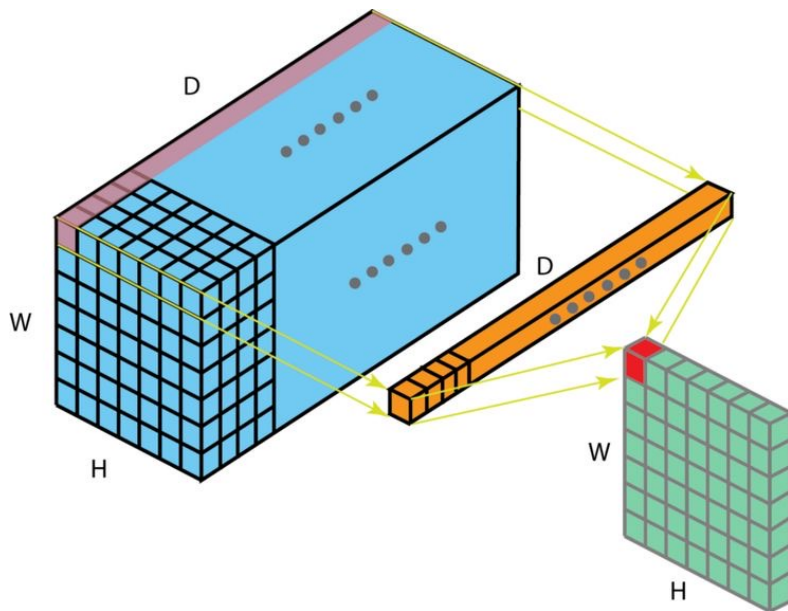
Other variants of convolution

- 1x1 convolution: reducing depth/channel resolution
- Deconvolution: **upsampling** by **multiply-and-accumulate**

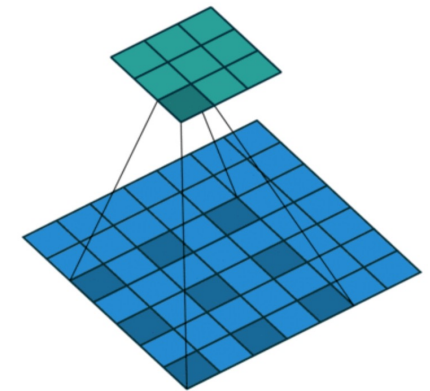


Other variants of convolution

- 1x1 convolution: reducing depth/channel resolution
- Deconvolution: upsampling by multiply-and-accumulate

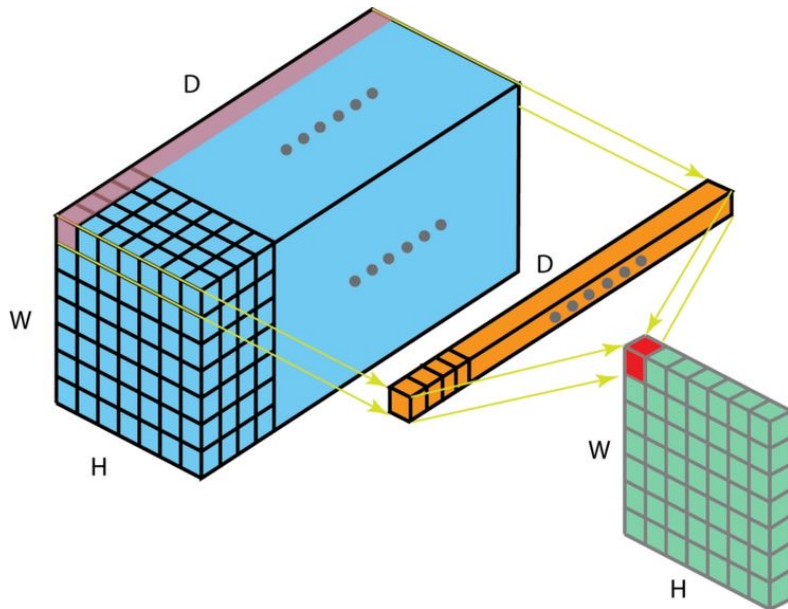


- Dilated convolution

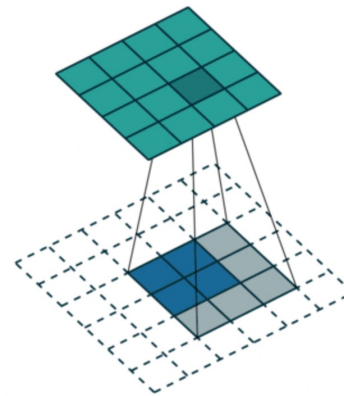


Other variants (check more details yourself)

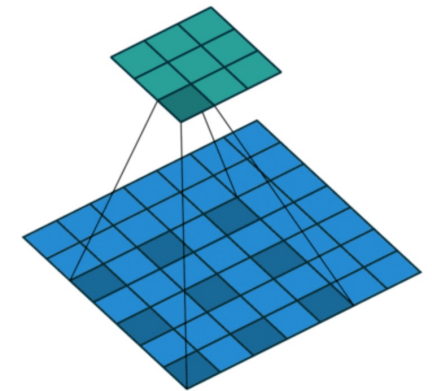
- 1x1 convolution: reducing depth/channel resolution



- Deconvolution: upsampling by multiply-and-accumulate



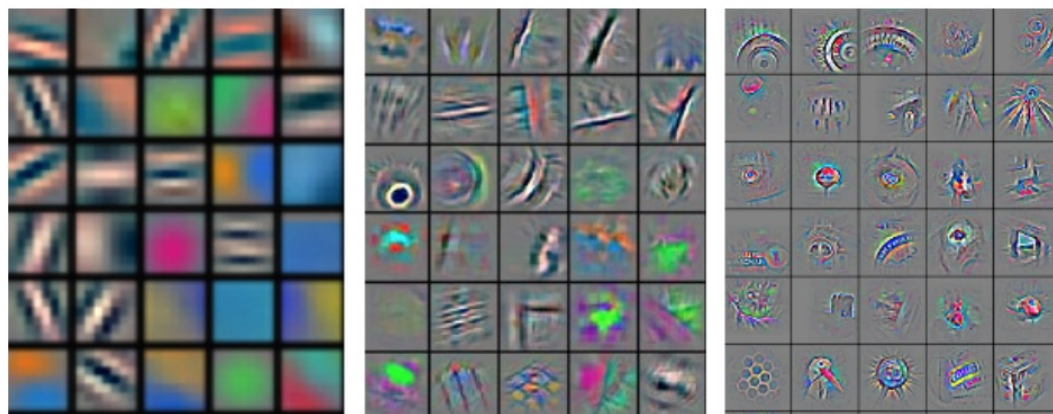
- Dilated convolution



- Graph convolution

Hand-crafted features vs. learned features

- Hand-crafted: e.g., total curvature, normal distance, etc.
- CNNs start with **raw images** and perform seemingly “uneducated” operations ...
- Learned features are reflected in **network weights: sensitivity or activation** w.r.t. certain patterns in the images



Typical features learned by a CNN becoming increasingly complex at deeper layers

Some key questions about CNNs/DNNs

- How to design the **network architecture**?
 - From feature hand-crafting to hand-crafting network architecture

Some key questions about CNNs/DNNs

- How to design the **network architecture**?
 - From feature hand-crafting to hand-crafting network architecture
- How to design networks that **generalize well to new data**?
 - Avoid **overfitting**?

Some key questions about CNNs/DNNs

- How to design the **network architecture**?
 - From feature hand-crafting to hand-crafting network architecture
- How to design networks that **generalize well to new data**?
 - Avoid **overfitting**?
- How to ensure there is enough data?
 - Going to **weak supervision** or **data augmentation**

Some key questions about CNNs/DNNs

- How to design the **network architecture**?
 - From feature hand-crafting to hand-crafting network architecture
- How to design networks that **generalize well to new data**?
 - Avoid **overfitting**?
- How to ensure there is enough data?
 - Going to **weak supervision** or **data augmentation**
- How to improve training efficiency?

Some key questions about CNNs/DNNs

- How to design the **network architecture**?
 - From feature hand-crafting to hand-crafting network architecture
- How to design networks that **generalize well to new data**?
 - Avoid **overfitting**?
- How to ensure there is enough data?
 - Going to **weak supervision** or **data augmentation**
- How to improve training efficiency?
- **Interpretability** or explainability of the networks

Geometric deep learning

- Learn to discriminate or **generate geometric data**
- Apply deep learning to **3D** data
- Can we replicate success of generative DNNs for images?

Remarkable progress on image generation

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

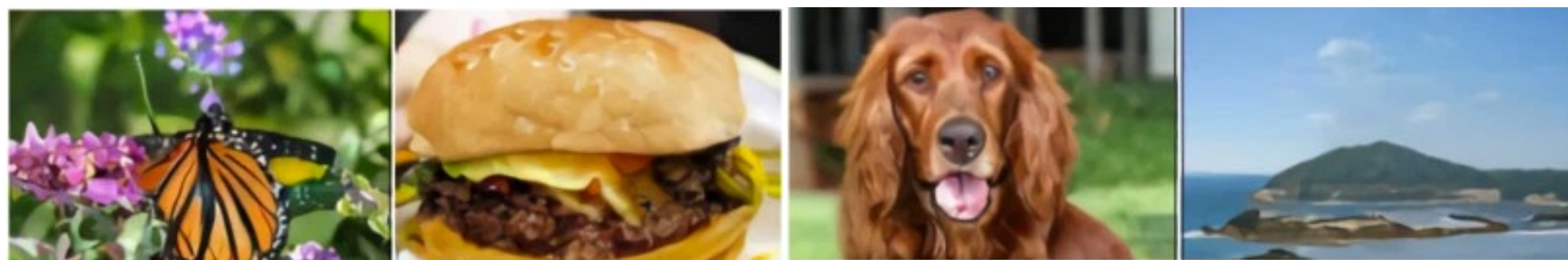


Remarkable progress on image generation

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

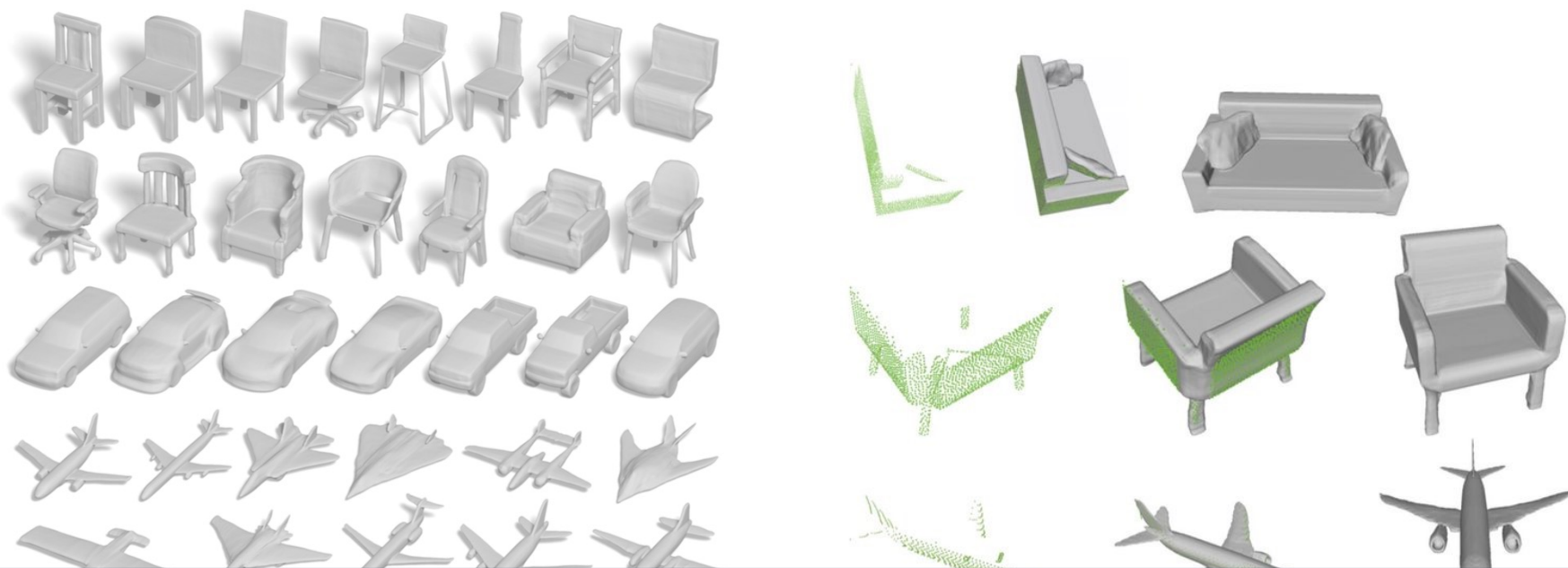


- Latest: BigGAN [Google Deepmind, 2018/19]



400 x 267 image resolution, using class conditionals

SOTA for 3D shape generation at the time



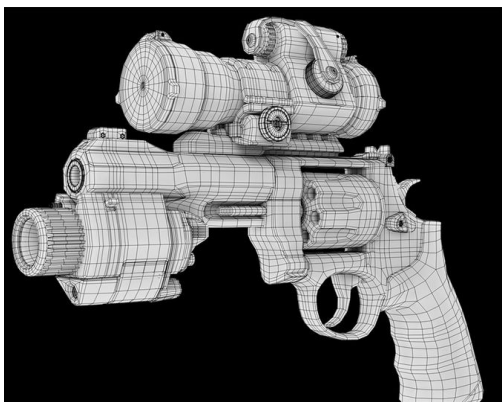
Some **unique challenges** to training deep neural networks (DNNs) for **3D shape generation** ...

Unique challenge #1: representation

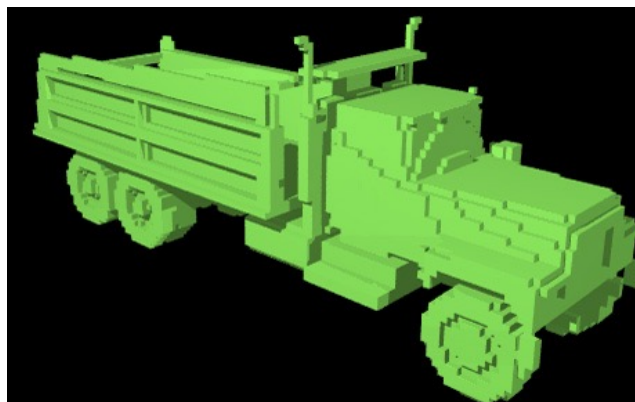
- Unlike images or speech, there is **no universally accepted representation or encoding** for 3D shapes

Unique challenge #1: representation

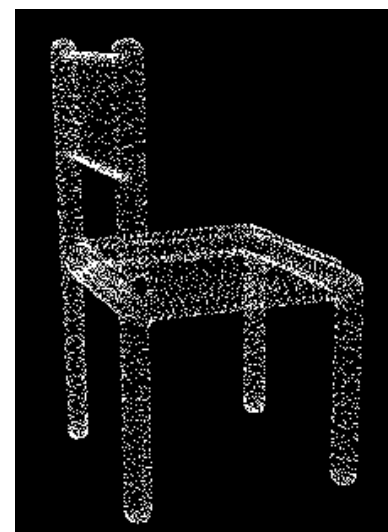
- Unlike images or speech, there is **no universally accepted representation or encoding** for 3D shapes
- Alternatives: low-level representations



Mesh: a set of triangles



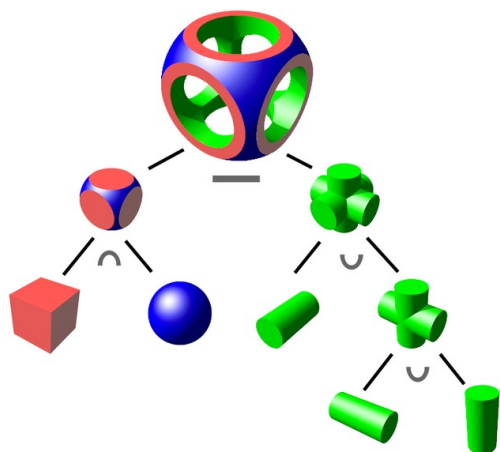
Volume: a grid of voxels



Point cloud: a set of points

Unique challenge #1: representation

- Many other representations



Procedural: e.g., CSG rep



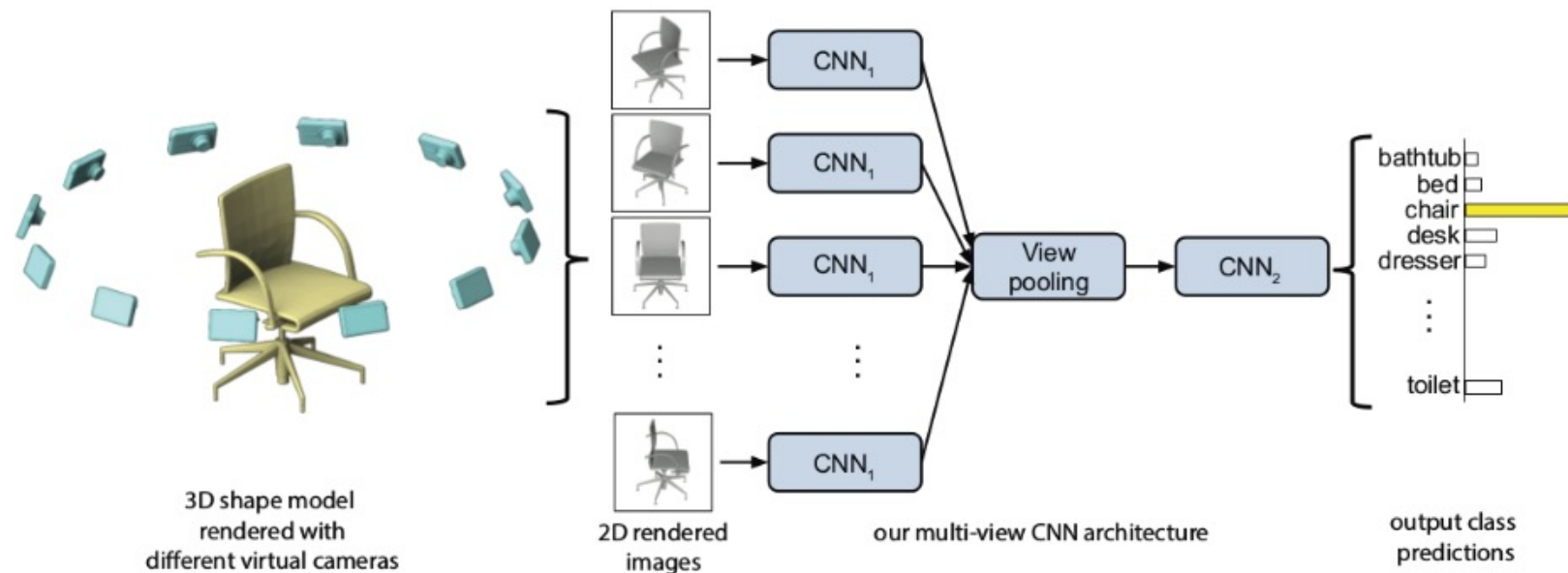
Structural: a set of parts



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

DNN examples: multi-view CNN (MVCNN)

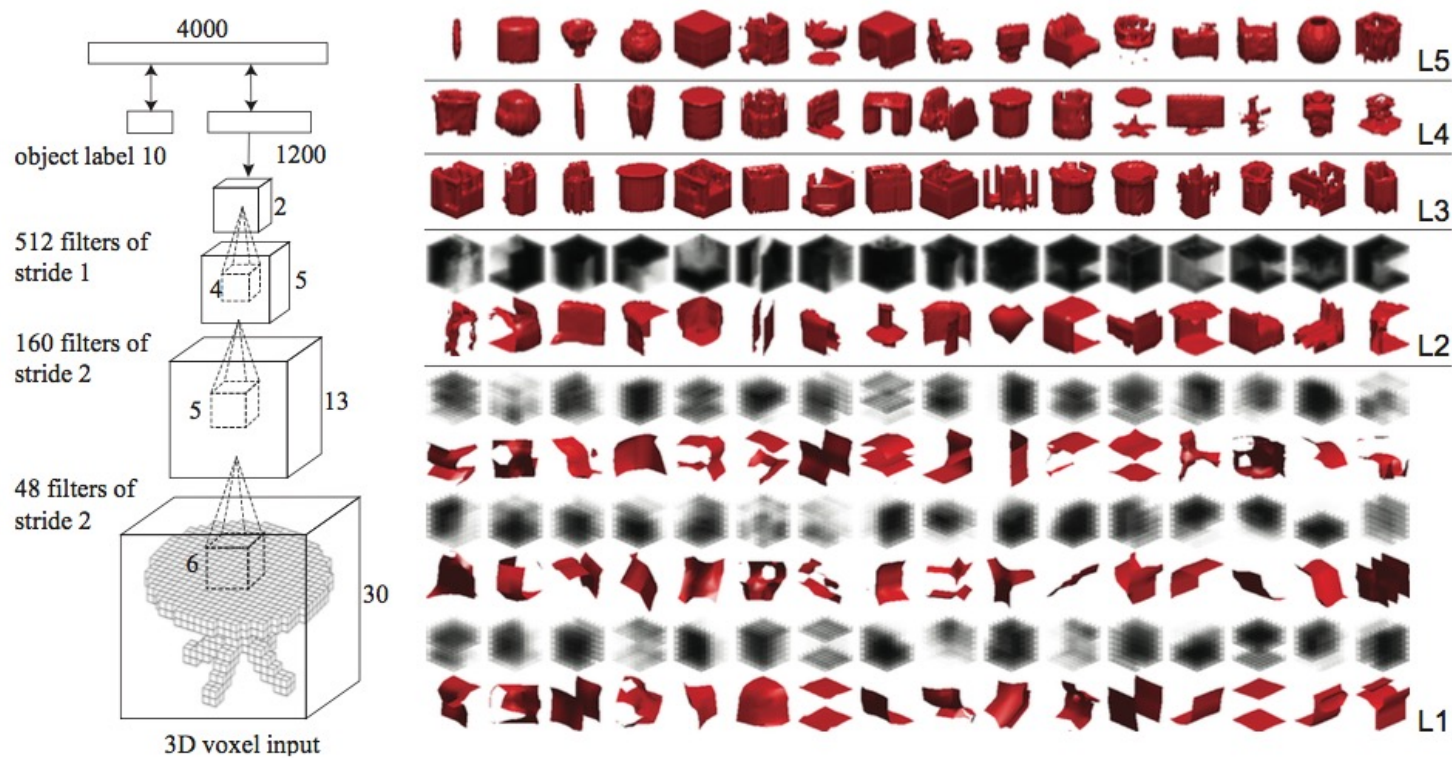
- Reuse standard components of image-based CNNs
- Not geometric; designed for classification not generation



3D ShapeNet: voxel representations

[Wu et al. 2015]

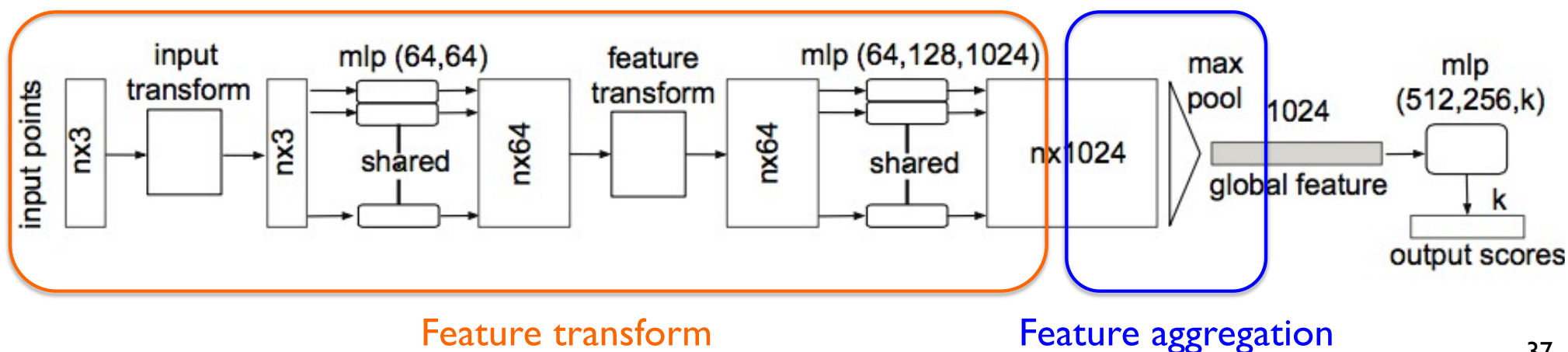
- Straightforward generation of image CNNs, for classification



PointNet (more later)

[Qi et al. 2017]

- First transform each 3D point into a high(1,024)-D feature.
- Then aggregate features into a signature for classification
- The max pooling ensures **permutation invariance**



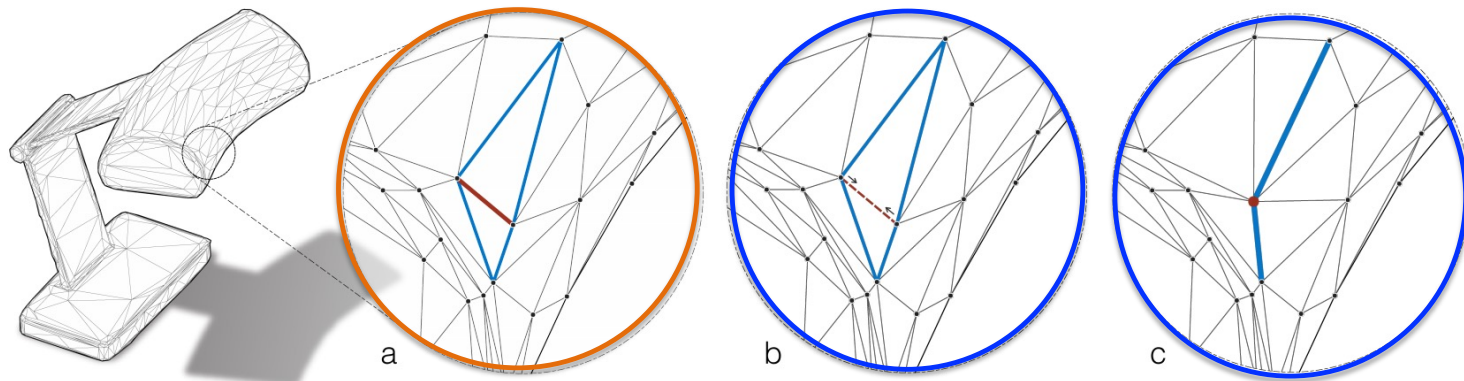
MeshCNN

MeshCNN: A Network with an Edge

RANA HANOCKA, Tel Aviv University
AMIR HERTZ, Tel Aviv University
NOA FISH, Tel Aviv University
RAJA GIRYES, Tel Aviv University
SHACHAR FLEISHMAN, Amazon
DANIEL COHEN-OR, Tel Aviv University

[Hanocka et al. 2019]

- Direct convolution on **irregular mesh connectivity**
- Mesh **edges** act as pixels in an image to store features
- Mesh pooling reduces mesh resolution via **edge collapse**

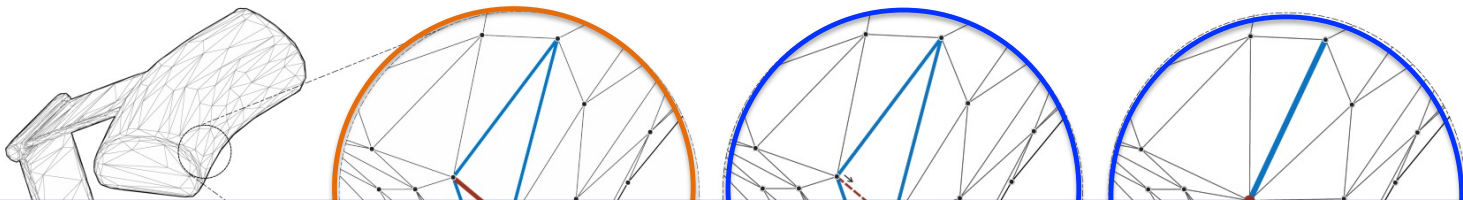


Convolve edge features

Mesh pooling via edge collapse

MeshCNN

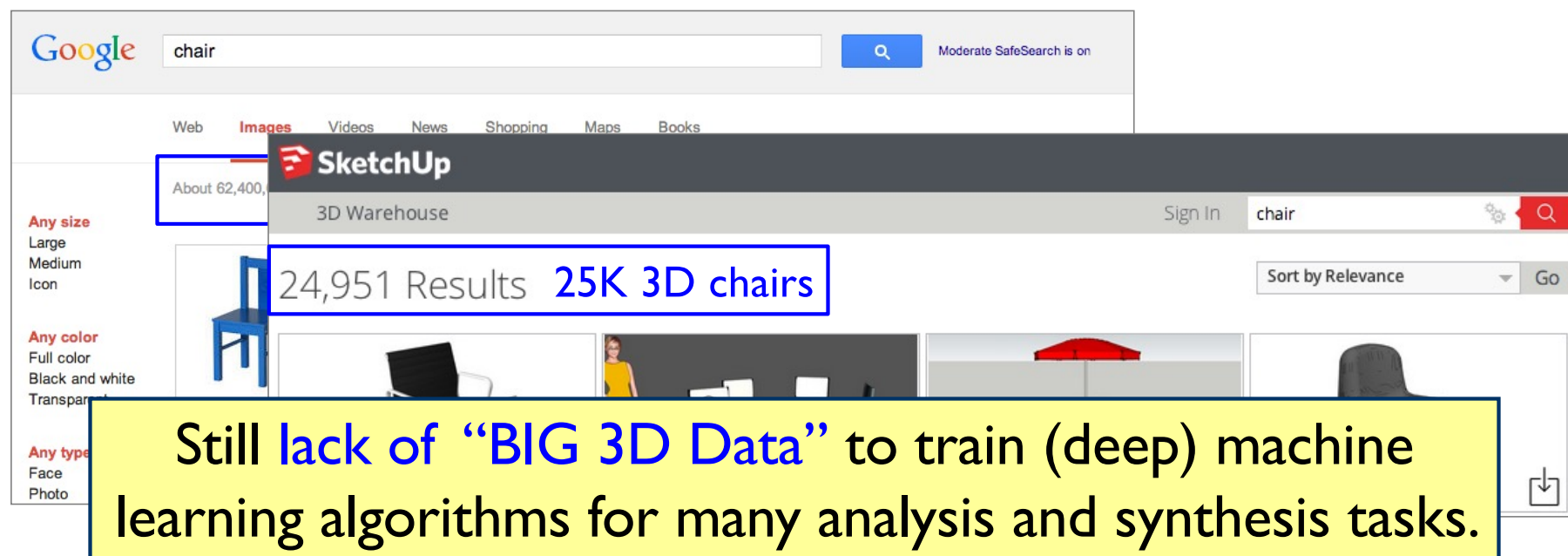
- Direct convolution on **irregular mesh connectivity**
- Mesh **edges** act as pixels in an image to store features
- Mesh pooling reduces mesh resolution via **edge collapse**



Most networks developed for these reps, in particular, those using **convolutional neural networks (CNNs)**, are designed for discriminative analysis and **recognition, not generation.**

Unique challenge #2: 3D data challenge

- Acquisition of and interaction with 3D contents are hard



Unique challenge #3: affordance/functionality

- 3D objects or designs are meant to be **used in real life**
 - Not enough to just have the right parts



Unique challenge #3: affordance/functionality

- 3D objects or designs are meant to be **used in real life**
 - Not enough to just have the right parts
 - Not enough to just “look right” as an image or rendering



Ultimate goal of 3D shape generation

- We live in 3D world to **interact** with our surroundings
 - We do not just see and observe, we **use** and we **act** ...

Ultimate goal of 3D shape generation

- We live in 3D world to **interact** with our surroundings
 - We do not just see and observe, we use and we act ...
- Our understanding does not stop at **what things are**
- Ultimately, the understanding is about
 - **how** things are
 - **how** to use them

That is **functionality!**

3D shapes need to **function properly**

Early work: theory of affordance (aside)

- Affordance is what the environment **offers or affords** the individual
 - It presents opportunities for **actions (e.g., sit) afforded** by a specific object or environment
 - Actions by agents, e.g., humans/hands



J.J. Gibson, "*The Ecological Approach to Visual Perception*", 1979

Affordance analysis in vision (aside)

CVPR 2011

What Makes a Chair a Chair?

Helmut Grabner¹

Juergen Gall¹

Luc Van Gool^{1,2}

¹Computer Vision Laboratory
ETH Zurich

{grabner,gall,vangool}@vision.ee.ethz.ch

²ESAT - PSI / IBBT
K.U. Leuven

luc.vangool@esat.kuleuven.be

Abstract

Many object classes are primarily defined by their functions. However, this fact has been left largely unexploited by visual object categorization or detection systems. We propose a method to learn an affordance detector. It identifies locations in the 3d space which “support” the particular function. Our novel approach “imagines” an actor performing an action typical for the target object class, instead of relying purely on the visual object appearance. So, function is handled as a cue complementary to appearance, rather than being a consideration after appearance-based detection. Experimental results are given for the functional category “sitting”. Such affordance is tested on a 3d representation of the scene, as can be realistically obtained through SfM or depth cameras. In contrast to appearance-based object detectors, affordance detection requires only very few training examples and generalizes very well to other sittable objects like benches or sofas when trained on a few chairs.

1. Introduction

“An object is first identified as having important functional relations, [...] perceptual analysis is derived of the functional concept [...]”

Nelson, 1974, [17]

“Affordances relate the utility of things, events, and places to the needs of animals and their actions in fulfilling them [...]. Affordances themselves are perceived and, in fact, are the essence of what we perceive.”

Gibson, 1982, [8, p. 60]

“There’s little we can find in common to all chairs – except for their intended use.”

Minsky, 1986, [16, p. 123]

“[...] objects like coffee cups are artifacts that were created to fulfill a function. The function of an object plays a critical role in processing that object [...] for categorization and naming.”

Carlson-Radvansky et al., 1999, [4]



Figure 1. The “chair-challenge” by I. and H. Bülthoff [3] (reprint with the author’s permission).

These quotes emphasize that functional properties or affordances¹ are essential for forming concepts and learning object categories. Experiments (e.g. [18, 4]) have demonstrated that both appearance and function are strong cues for learning by infants. Initially they attend only to the form of an object. Later they use form and function and finally (by the age of 18 months) they attend to the relationships between form and function. Furthermore, Booth and Waxman [2] have identified two salient cues that facilitate categorization in infancy, namely (i) object functions and (ii) object names. Moreover, names of objects most often evolve on the basis of function².

Whereas all this is well known for a long time, it has been left mostly unused for object detection in computer vision. Taking a look at the results of the recent Pascal VOC Challenge [5], the performance still strongly depends

¹“Affordance: A situation where an object’s sensory characteristics intuitively imply its functionality and use. [...] A chair, by its size, its curvature, its balance, and its position, suggests sitting on it”, <http://www.usabilityfirst.com/glossary/affordance>, 2010/07/28. Introduced in 1979 by Gibson [9, p. 127] based on the verb *afford*.

²When considering the evolution of a word for an object, most of the time it is based on its function. For example the word “chair”: PIE base **sed-* (to sit) → Latin *sedentarius* (sitting, remaining in one place) → *sedentary* (meaning “not in the habit of exercise”) → *cathedral* → *chair*. <http://www.etymonline.com>, 2010/10/02.

- Fit canonical human poses into 3D scenes to detect sitting affordances

Affordance analysis in vision (aside)

CVPR 2011

What Makes a Chair a Chair?

Helmut Grabner¹

Juergen Gall¹

Luc Van Gool^{1,2}

¹Computer Vision Laboratory
ETH Zurich
{grabner,gall,vangool}@vision.ee.ethz.ch

²ESAT - PSI / IBBT
K.U. Leuven
luc.vangool@esat.kuleuven.be

Abstract

Many object classes are primarily defined by their functions. However, this fact has been left largely unexploited by visual object categorization or detection systems. We propose a method to learn an affordance detector. It identifies locations in the 3d space which "support" the particular function. Our novel approach "imagines" an actor performing an action typical for the target object class, instead of relying purely on the visual object appearance. So, function is handled as a cue complementary to appearance, rather than being a consideration after appearance-based detection. Experimental results are given for the functional category "sitting". Such affordance is tested on a 3d representation of the scene, as can be realistically obtained through SfM or depth cameras. In contrast to appearance-based object detectors, affordance detection requires only very few training examples and generalizes very well to other sitable objects like benches or sofas when trained on a few chairs.

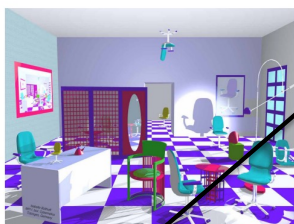


Figure 1. The "chair-challenge" by I. and H. Bülthoff [3] (reprint with the author's permission)

These quotes emphasize that functional properties or affordances¹ are essential for forming concepts and learning object categories. Experiments (e.g. [18, 4]) have demonstrated that both appearance and function are strong cues for learning by infants. Initially they attend only to the form of an object. Later they use form and function and finally (by the age of 18 months) they attend to the relationships between form and function. Furthermore, Booth and Waxman [2] have identified two salient cues that facilitate categorization in infancy, namely (i) object functions and (ii) object names. Moreover, names of objects most often evolve on the basis of function².

Whereas all this is well known for a long time, it has been left mostly unused for object detection in computer vision. Taking a look at the results of the recent Pascal VOC Challenge [5], the performance still strongly depends

1. Introduction

"An object is first identified as having important functional relations, [...] perceptual analysis is derived of the functional concept [...]."

Nelson, 1974, [17]

Affordances relate the utility of things, events, and places to the needs of animals and their actions in fulfilling them [...]. Affordances themselves are perceived and, in fact, are the essence of what we perceive."

Gibson, 1982, [8, p. 60]

"There's little we can find in common to all chairs – except for their intended use."

Minsky, 1986, [16, p. 123]

"[...] objects like coffee cups are artifacts that were created to fulfill a function. The function of an object plays a critical role in processing that object [...] for categorization and naming."

Carlson-Radvansky et al., 1999, [4]

- Fit canonical human poses into 3D scenes to detect sitting affordances

"An object is first identified as having important functional relations, [...], perceptual analysis is derived of the functional concept [...]"

Nelson [1974]

¹"Affordance: A situation where an object's sensory characteristics intuitively imply its functionality and use. [...] A chair, by its size, its curvature, its balance, and its position, suggests sitting on it", <http://www.usabilityfirst.com/glossary/affordance>, 2010/07/28. Introduced in 1979 by Gibson [9, p. 127] based on the verb *afford*.

²When considering the evolution of a word for an object, most of the time it is based on its function. For example the word "chair": PIE base **sed-* (to sit) → Latin *sedentarius* (sitting, remaining in one place) → *sedentary* (meaning "not in the habit of exercise") → *cathedral* → *chair*. <http://www.etymonline.com>, 2010/10/02.

Affordance analysis in vision (aside)

CVPR 2011

What Makes a Chair a Chair?

Helmut Grabner¹

Juergen Gall¹

Luc Van Gool^{1,2}

¹Computer Vision Laboratory
ETH Zurich

{grabner,gall,vangool}@vision.ee.ethz.ch

²ESAT - PSI / IBBT
K.U. Leuven

luc.vangool@esat.kuleuven.be

Abstract

Many object classes are primarily defined by their functions. However, this fact has been left largely unexploited by visual object categorization or detection systems. We propose a method to learn an affordance detector. It identifies locations in the 3d space which “support” the particular function. Our novel approach “imagines” an actor performing an action typical for the target object class, instead of relying purely on the visual object appearance. So, function is handled as a cue complementary to appearance, rather than being a consideration after appearance-based detection. Experimental results are given for the functional category “sitting”. Such affordance is tested on a 3d representation of the scene, as can be realistically obtained through SfM or depth cameras. In contrast to appearance-based object detectors, affordance detection requires only very few training examples and generalizes very well to other sittable objects like benches or sofas when trained on a few chairs.

1. Introduction

“An object is first identified as having important functional relations, [...] perceptual analysis is derived of the functional concept [...]”

Nelson, 1974, [17]

“Affordances relate the utility of things, events, and places to the needs of animals and their actions in fulfilling them [...]. Affordances themselves are perceived and, in fact, are the essence of what we perceive.”

Gibson, 1982, [8, p. 60]

“There’s little we can find in common to all chairs – except for their intended use.”

Minsky, 1986, [16, p. 123]

“[...] objects like coffee cups are artifacts that were created to fulfill a function. The function of an object plays a critical role in processing that object [...] for categorization and naming.”

Carlson-Radvansky et al., 1999, [4]



Figure 1. The “chair-challenge” by I. and H. Bülthoff [3], reprint with the author’s permission).

These quotes emphasize that functional properties or affordances¹ are essential for forming concepts and learning object categories. Experiments (e.g. [18, 4]) have demonstrated that both appearance and function are strong cues for learning by infants. Initially they attend only to the form of an object. Later they use form and function and finally (by the age of 18 months) they attend to the relationships between form and function. Furthermore, Booth and Waxman [2] have identified two salient cues that facilitate categorization in infancy, namely (i) object functions and (ii) object names. Moreover, names of objects most often evolve on the basis of function².

Whereas all this is well known for a long time, it has been left mostly unused for object detection in computer vision. Taking a look at the results of the recent Pascal VOC Challenge [5], the performance still strongly depends

¹“Affordance: A situation where an object’s sensory characteristics intuitively match its functionality and use. [...] A chair, by its size, its curvature, its balance, and its position, suggests sitting on it”, <http://www.usabilityfirst.com/glossary/affordance>, 2010/07/28. Introduced in 1979 by Gibson [9, p. 127] based on the verb *afford*.

²When considering the evolution of a word for an object, most of the time it is based on its function. For example the word “chair”: PIE base *sed- (to sit) → Latin *sedentarius* (sitting, remaining in one place) → *sedentary* (meaning “not in the habit of exercise”) → *cathedral* → *chair*. <http://www.etymonline.com>, 2010/10/02.

- Fit canonical human poses into 3D scenes to detect sitting affordances

“There’s little we can find in common to all chairs – except for their **intended use**.”

Minsky [1986]

How to define affordance/functionality?

- **Interactions** between a 3D object with other objects (the **agents**) in a given **scene context** reflects its functionality
- Agents can be
 - Humans or hands
 - Other 3D objects

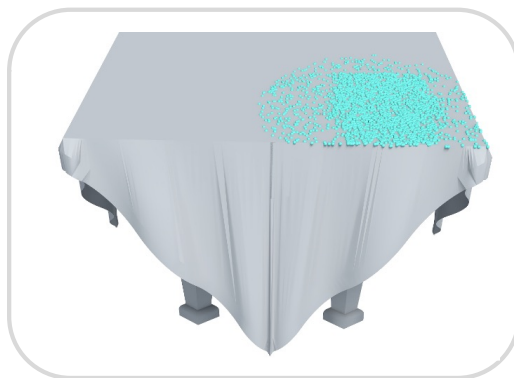


How to represent object-object interactions

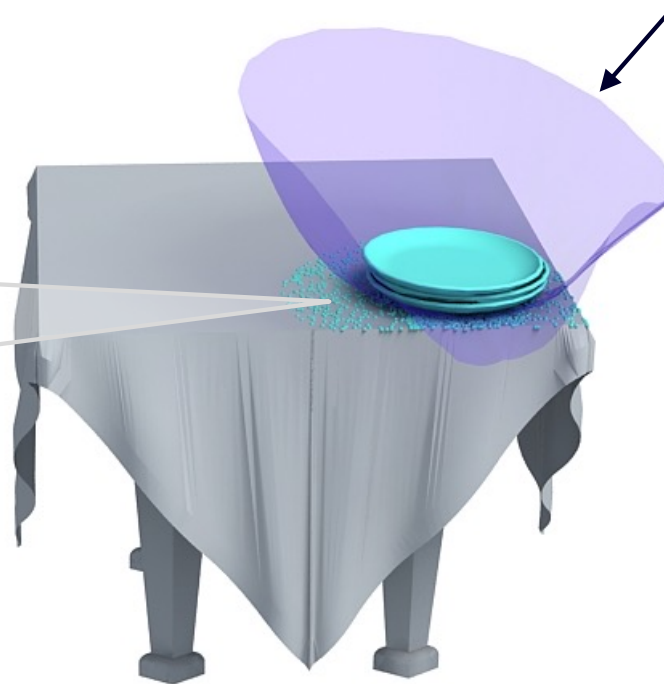
IBS: Intersection Bisector Surface (to describe the **interaction**)

[Zhao et al. TOG 2014]

[Hu et al. SIGGRAPH 2015]

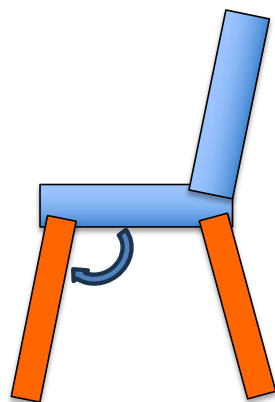


IR: Interaction Region (to describe the **object geometry**)

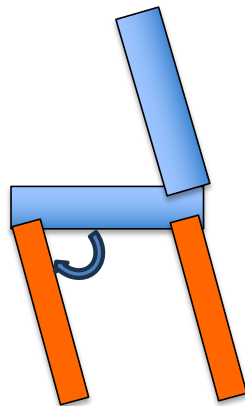


More generally ...

- Object functions are characterized by object **structures**, i.e., **parts + relations**,



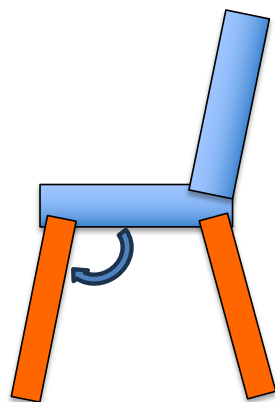
Functional



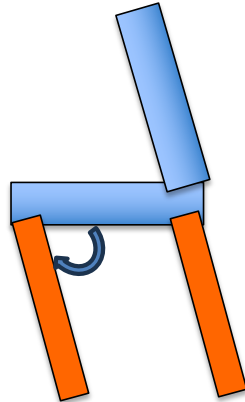
Non-functional

More generally ...

- Object functions are characterized by object structures, i.e., parts + relations, realized by **motion**,



Functional

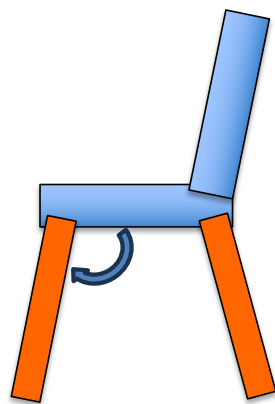


Non-functional

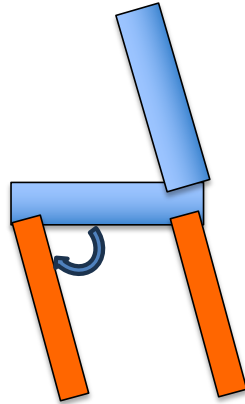


More generally ...

- Object functions are characterized by object structures, i.e., parts + relations, realized by motion, via **interactions**



Functional



Non-functional



Human and bicycle



Hanger and jackets



Hand and racket



Human and backpack

Before functionality ...

- Results of 3D generative NNs **still visually unsatisfactory**
 - Low resolution and geometric/structural/topological noise



It is important to find **the right shape representations** for training DNNs to generate quality 3D **shapes**

What is a shape?



shape

/SHāp/

noun

1. the external form or appearance characteristic of someone or something, the outline of an area or figure.

"she liked the shape of his nose"

synonyms: form, appearance, configuration, formation, structure; [More](#)

Shape vs. image



shape

/SHāp/

noun



vs.



1. the external form or appearance characteristic of someone or something, the outline of an area or figure.

"she liked the shape of his nose"

synonyms: [form](#), [appearance](#), [configuration](#), [formation](#), [structure](#); [More](#)

Shape vs. image



shape

/SHāp/

noun

1. the external form or appearance characteristic of someone or something, the outline of an area or figure.
"she liked the shape of his nose"
synonyms: form, appearance, configuration, formation, structure; [More](#)

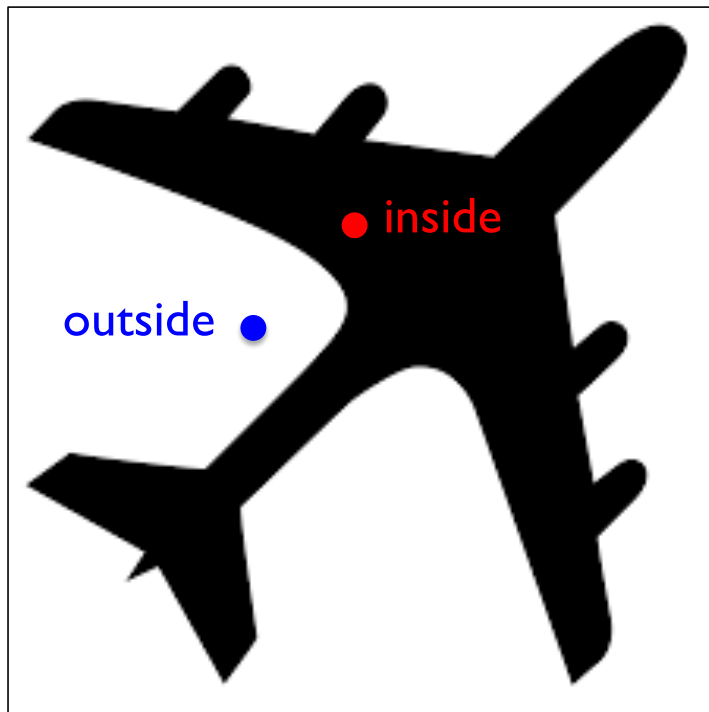


vs.

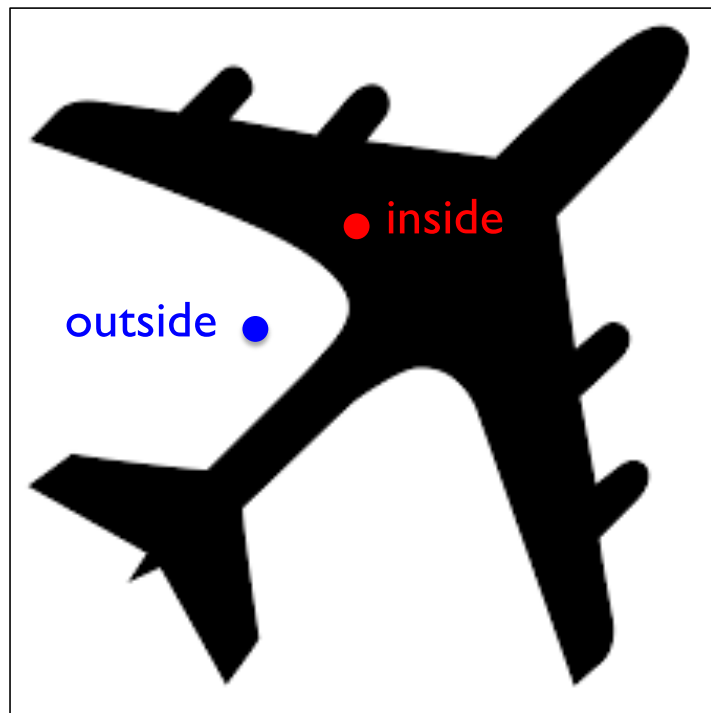


- A shape is **defined**/characterized by its **boundary/outline**
- Image boundary is artificial: it is because we had to crop

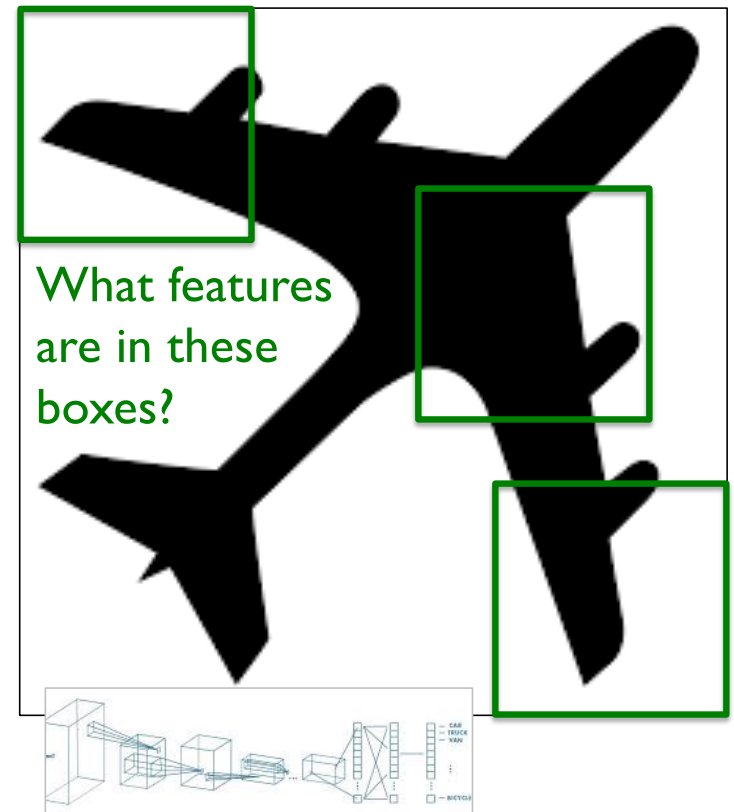
Shape boundary is about what is **inside/outside**



It is **not** really about feature inference



vs. image
convolution
via CNNs



CNNs “see” textures, humans see shapes (aside)



(a) Texture image

81.4%	Indian elephant
10.3%	indri
8.2%	black swan



(b) Content image

71.1%	tabby cat
17.3%	grey fox
3.3%	Siamese cat



(c) Texture-shape cue conflict

63.9%	Indian elephant
26.4%	indri
9.6%	black swan

“ImageNet-trained CNNs are biased towards texture; Increasing shape bias improves accuracy and robustness” [Geirhos et al. ICLR 2019]

To learn to generate shapes, we should ask ...

Is this point inside the shape?



Inside 

Outside

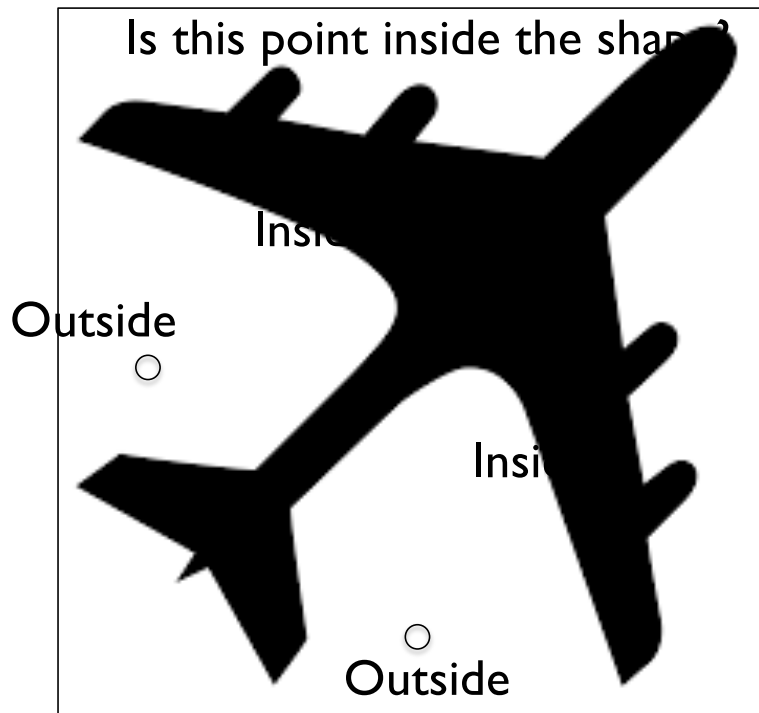


Inside 



Outside

To learn to generate shapes, we should ask ...

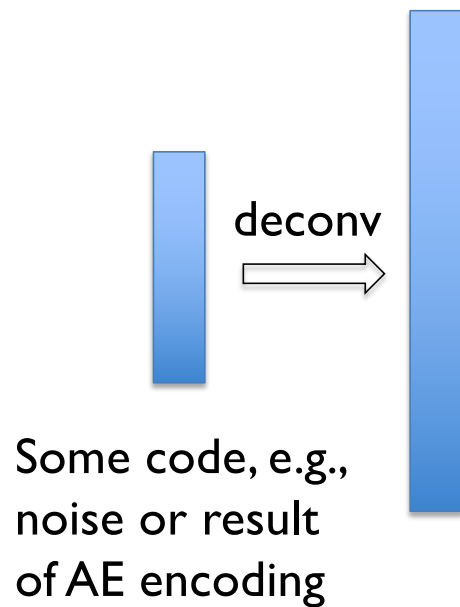


A typical CNN-based shape generator

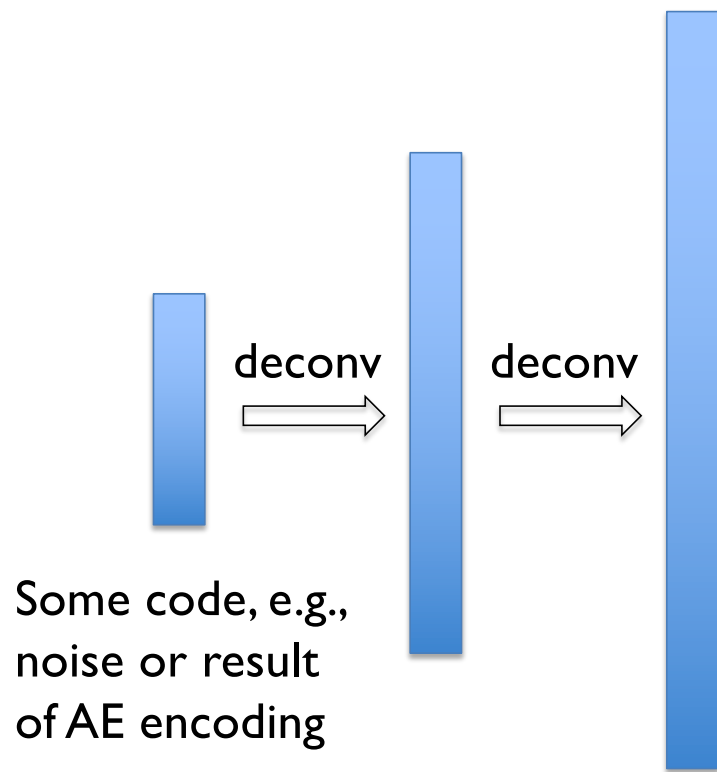


Some code, e.g.,
noise or result
of AE encoding

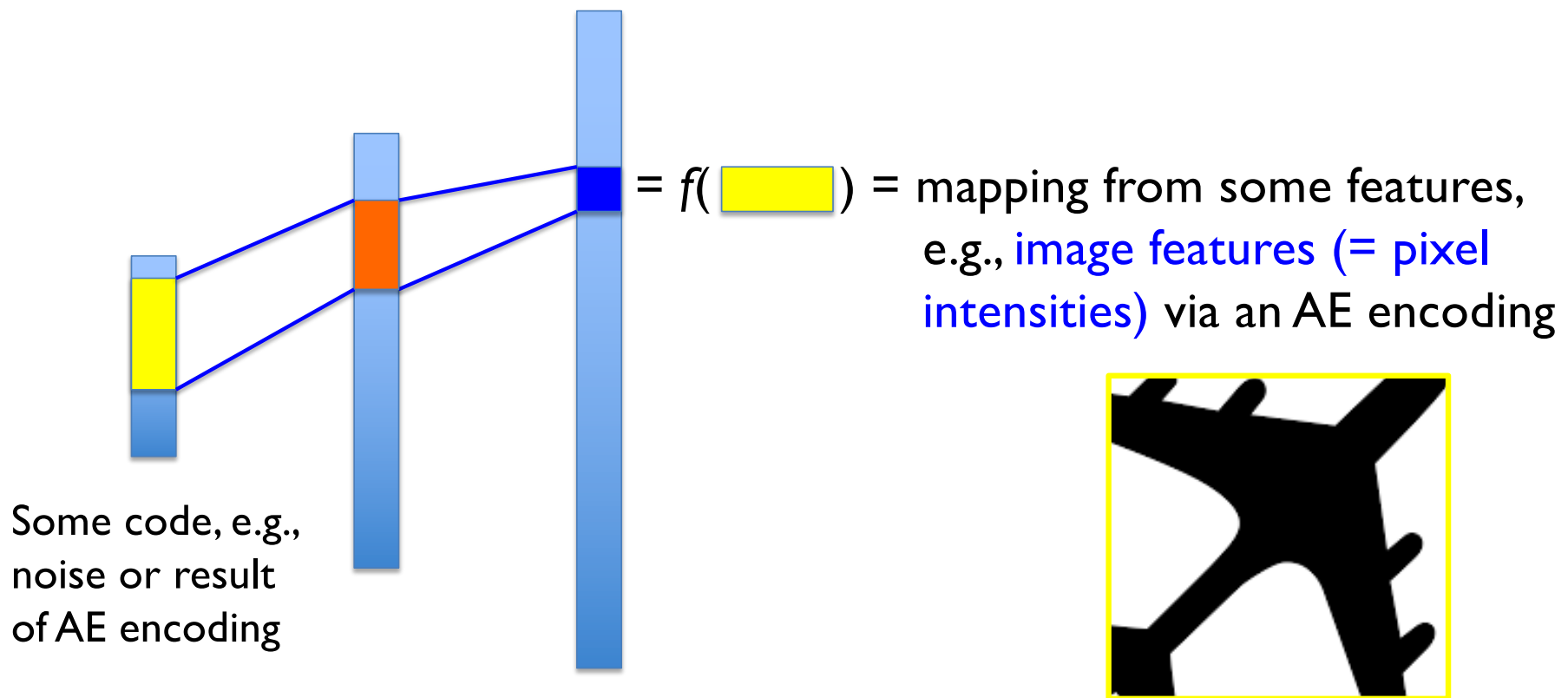
A typical CNN-based shape generator



A typical CNN-based shape generator

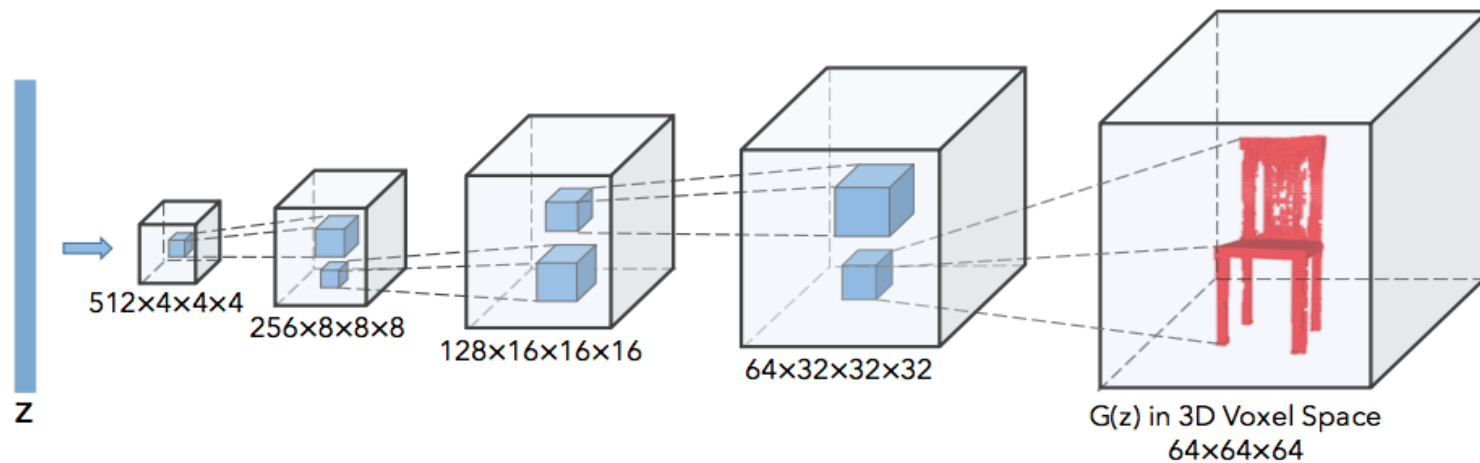


A mapping from features to voxel values



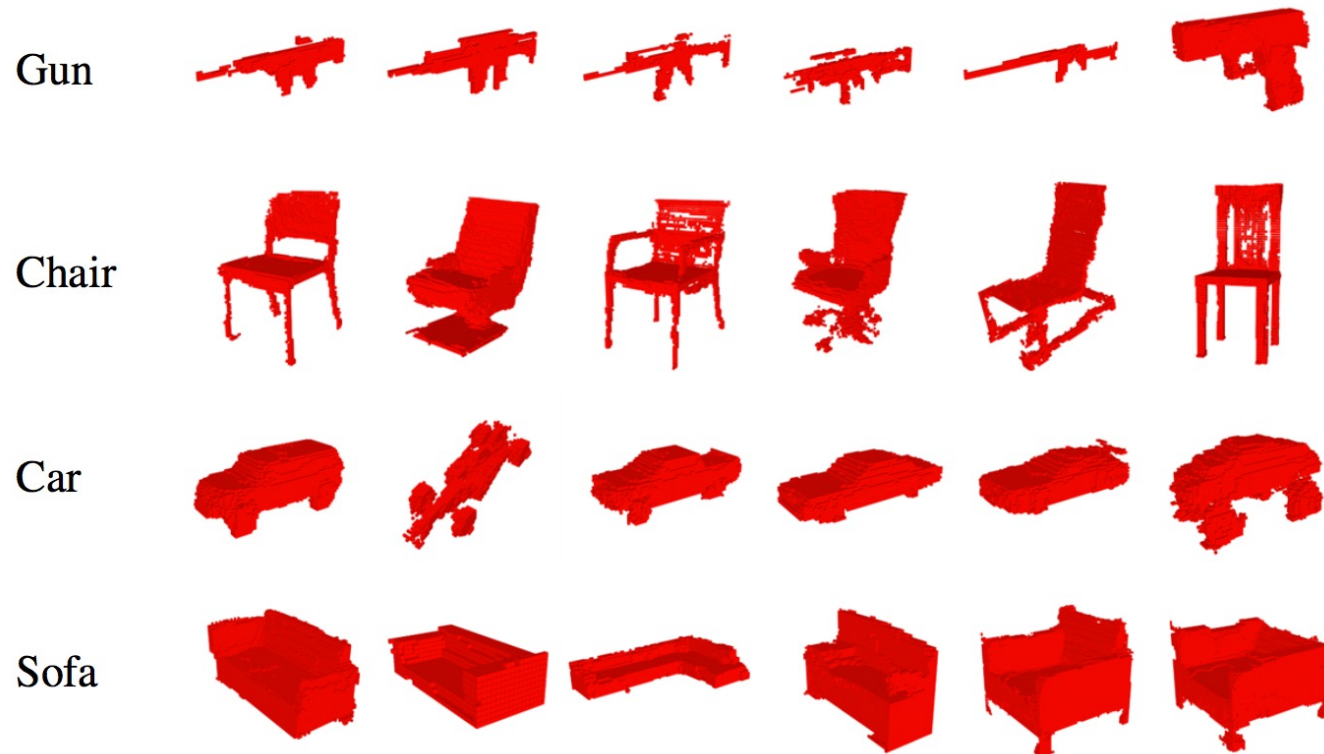
3D generative adversarial network (3D-GAN)

- 3D shape as **voxels**: combine volumetric CNN and GAN
- Generative network maps 200d vector to 64^3 volume



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

Shape generation results by 3D-GAN



3D volumetric shapes generated from random latent vectors

Let us learn the right mapping ...

$f(p, S)$: is point p inside or outside shape S

Let us learn the right mapping ...

$f(p, S)$: is point p inside or outside shape S

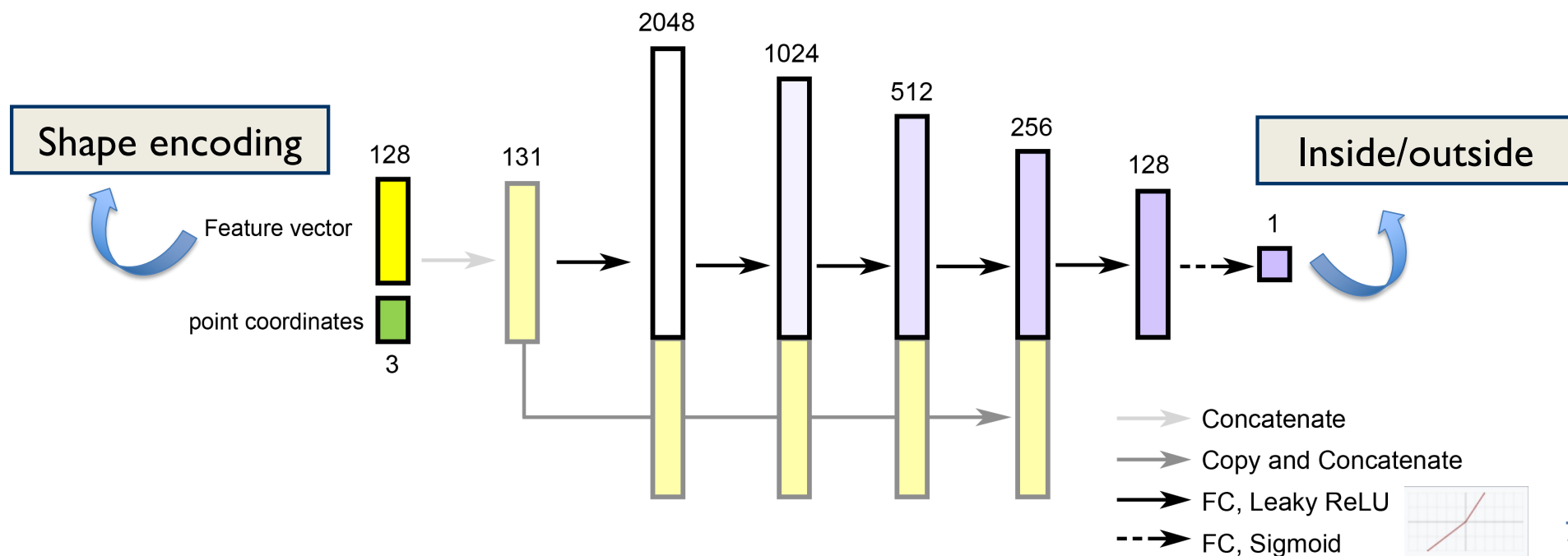
e.g.,

$$\begin{aligned} f(p, S) &= 1, \text{ if } p \text{ is outside } S \\ &= 0, \text{ otherwise} \end{aligned}$$

- This is an **implicit representation**: a shape is composed of the set of all points satisfying an equation $f(x) = 0$
- Point p can be in $\mathbb{R}^3$, so it is a **continuous** representation

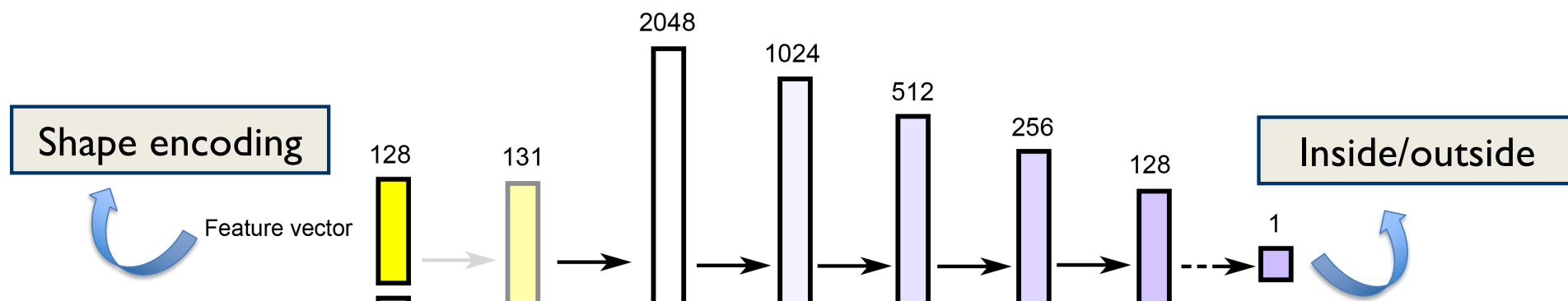
IM-NET: an implicit field generator

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



IM-NET: an implicit field generator

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

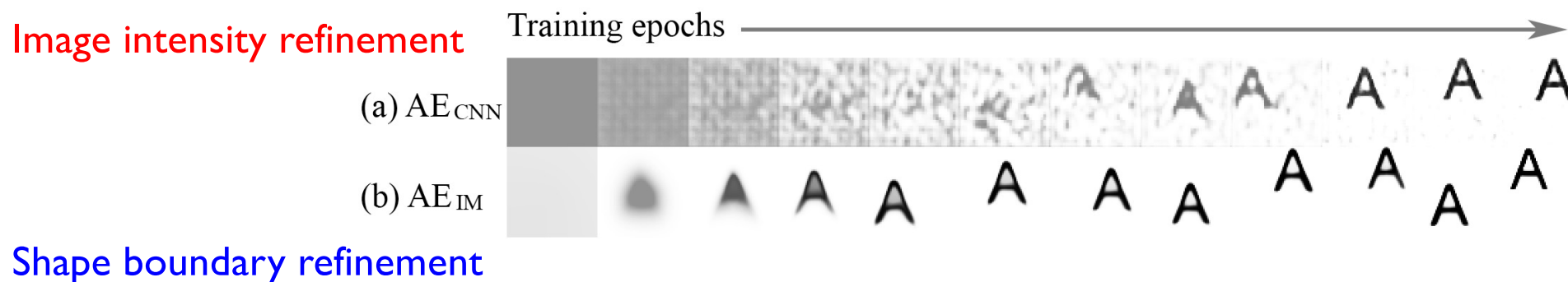


IM-NET: trained on point-value pairs and **learns shape boundaries**

Traditional CNN-based decoders **learn feature-to-voxel mappings**.
The features are **voxel intensity distributions** inside various boxes

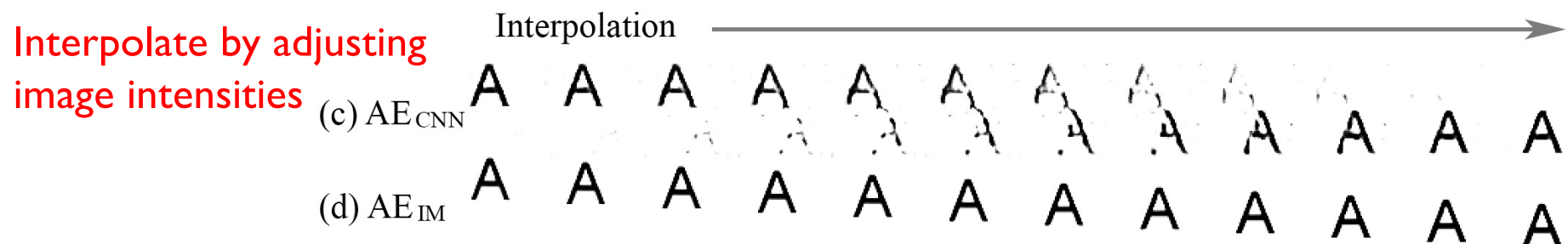
CNN vs. IM-NET for shape generations

- CNN vs. implicit decoders on learning to generate A's
- Networks were trained on same letter shape A, with white background, in different locations



CNN vs. IM-NET for shape interpolation

- Networks were trained in the same way
- In-between results were generated from **linearly interpolated latent codes** between source and target



Interpolate by moving
the shapes: the right way

Comparing 3D shape generation results

[Chen and Zhang, CVPR 2019]

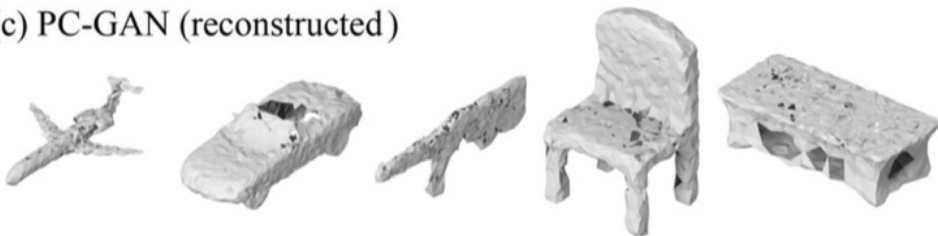
(a) 3DGAN



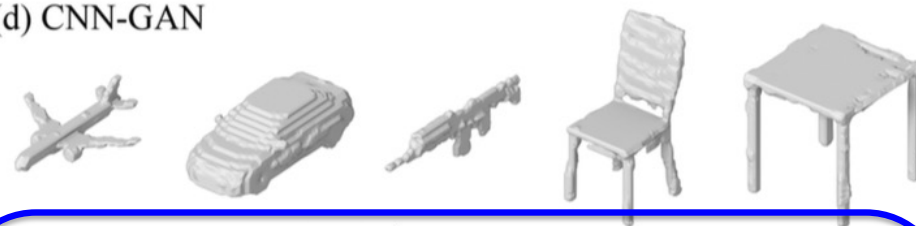
(b) PC-GAN



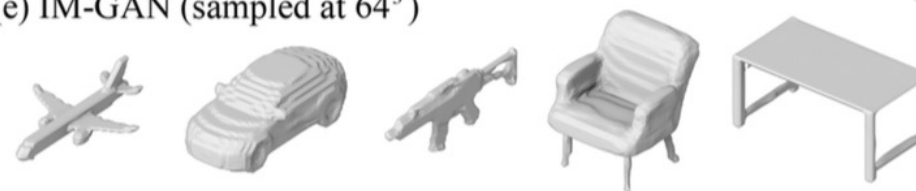
(c) PC-GAN (reconstructed)



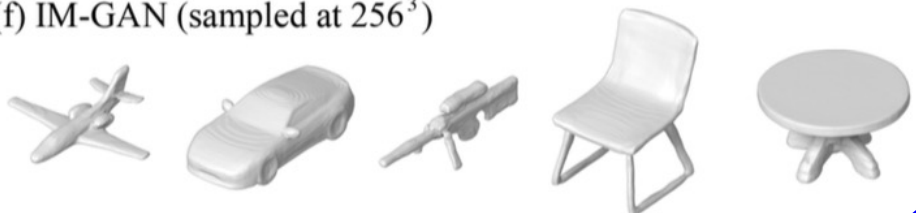
(d) CNN-GAN



(e) IM-GAN (sampled at 64^3)



(f) IM-GAN (sampled at 256^3)



Comparing 3D shape interpolation results

[Chen and Zhang, CVPR 2019]



2019: the start of neural implicit

Learning Implicit Fields for Generative Shape Modeling

Zhiqin Chen, Hao Zhang

(Submitted on 6 Dec 2018 (v1), last revised 5 Apr 2019 (this version, v3))

Occupancy Networks: Learning 3D Reconstruction in Function Space

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger

(Submitted on 10 Dec 2018)

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, Steven Lovegrove

(Submitted on 16 Jan 2019)

Deep Level Sets: Implicit Surface Representations for 3D Shape Inference

Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, Anders Eriksson

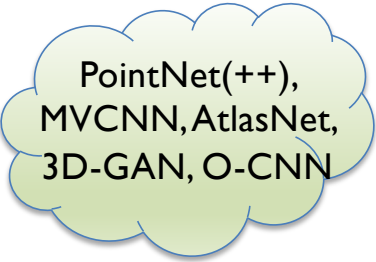
(Submitted on 21 Jan 2019)

Learning Shape Templates with Structured Implicit Functions

Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, Thomas Funkhouser

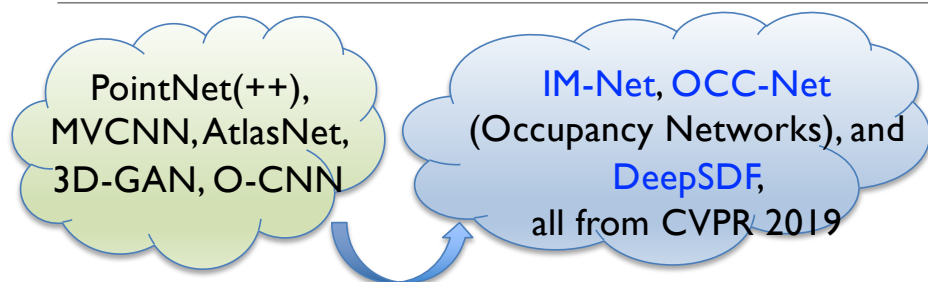
(Submitted on 12 Apr 2019)

Evolution of 3D rep learning: up to 2018



PointNet(++),
MVCNN, AtlasNet,
3D-GAN, O-CNN

Neural fields since 2019



Learning Implicit Fields for Generative Shape Modeling

Zhiqin Chen, Hao Zhang

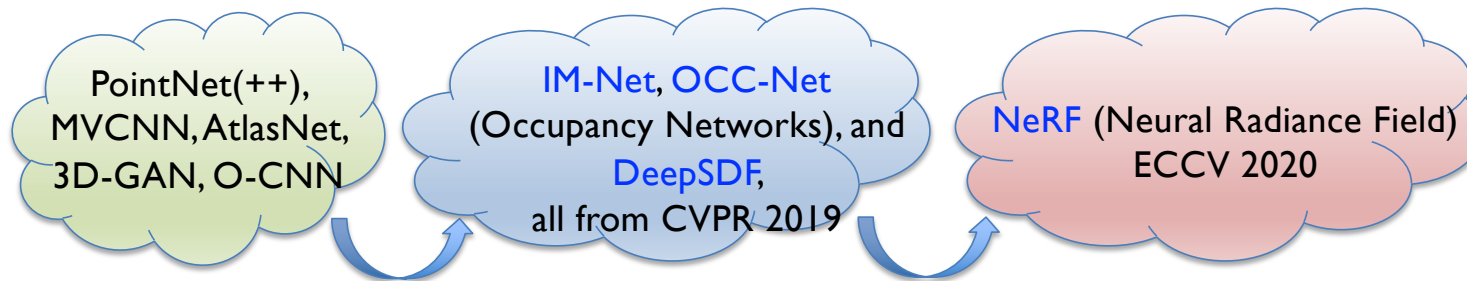
Occupancy Networks: Learning 3D Reconstruction in Function Space

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, Steven Lovegrove

2020: NeRF = multi-view to 3D via NVS



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

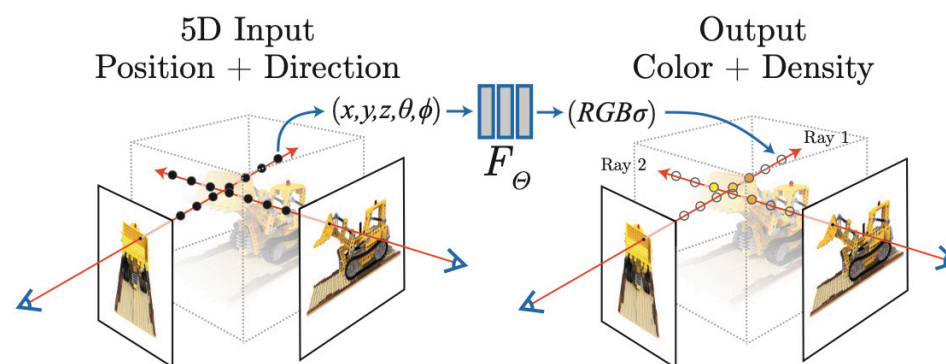
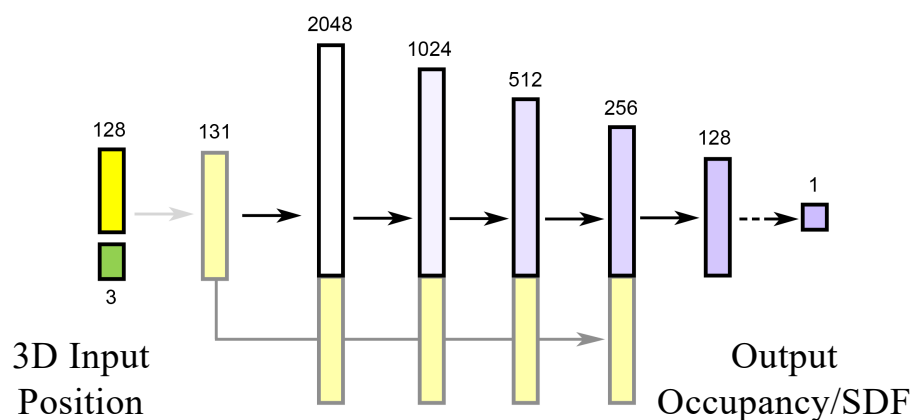
Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng

- NVS = novel view synthesis: rendering the 3D scene from another view direction, to still produce an image
- NeRF is not an explicit 3D entity: it is a neural network

NeRF: still a coordinate-based representation

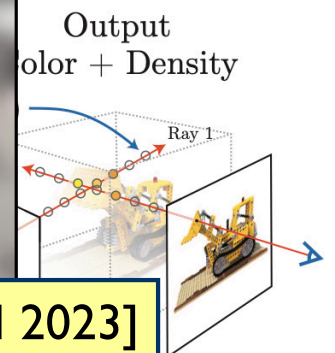
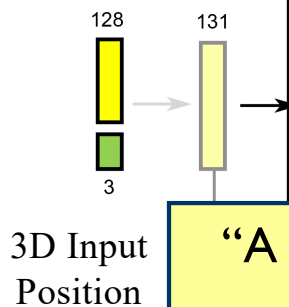
IM-Net, OCC-Net
(Occupancy Networks), and
DeepSDF,
all from CVPR 2019

NeRF (Neural Radiance Field)
ECCV 2020



NeRF:

Representation



“A radical new approach was proposed by Kerbl et al. [SIGGRAPH 2023] using 3D gaussian splats (3DGS) for their [point primitives](#).

Ravi Ramamoorthi

“NeRFs: The Search for the Best 3D Representation”, arXiv, August 2023