More Generative Models ☺

Conditional GANs on Videos

- Challenge:
  - Each frame is high quality, but temporally inconsistent

Video-to-Video Synthesis

- Sequential Generator:

$$p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T) = \prod_{t=1}^{T} p(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t}).$$

past L generated frames

past L source frames

(set L = 2)

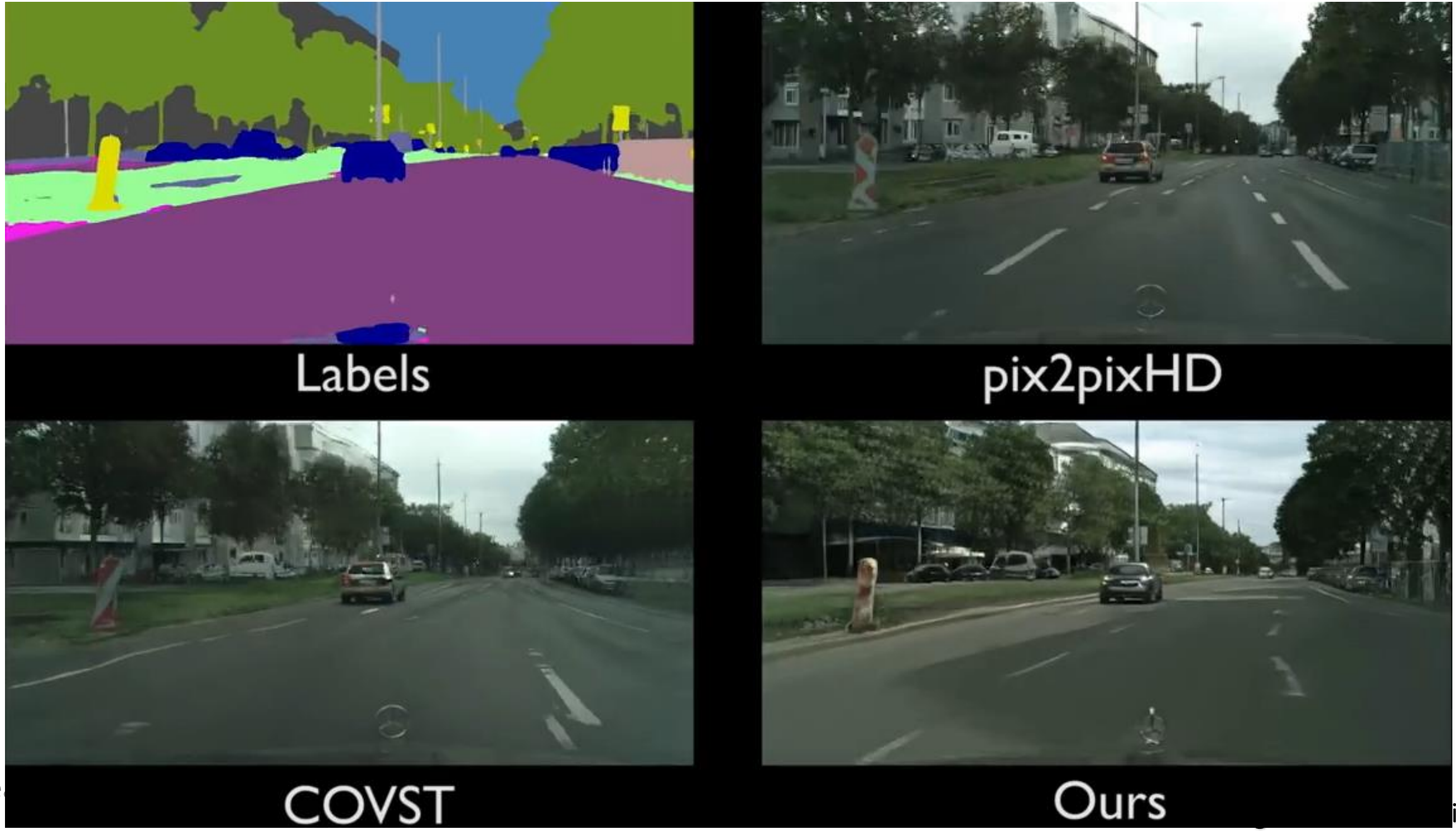
- Conditional Image Discriminator D_i (is it real image)
- Conditional Video Discriminator D_v (temp. consistency via flow)

Full Learning Objective:

$$\min_F \left(\max_{D_I} \mathcal{L}_I(F, D_I) + \max_{D_V} \mathcal{L}_V(F, D_V) \right) + \lambda_W \mathcal{L}_W(F),$$

Wang et al. 18: Vid2Vid

Video-to-Video Synthesis



Video-to-Video Synthesis

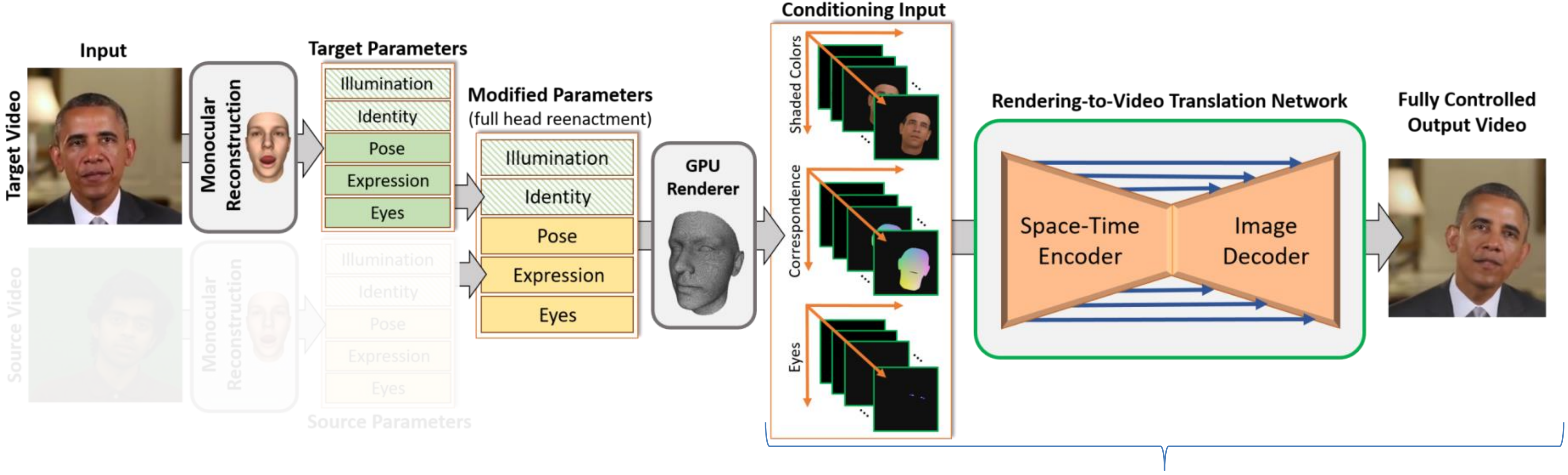
- Key ideas:
  - Separate discriminator for temporal parts
    - In this case based on optical flow
  - Consider recent history of prev. frames
  - Train all of it jointly

Wang et al. 18: Vid2Vid

Deep Video Portraits

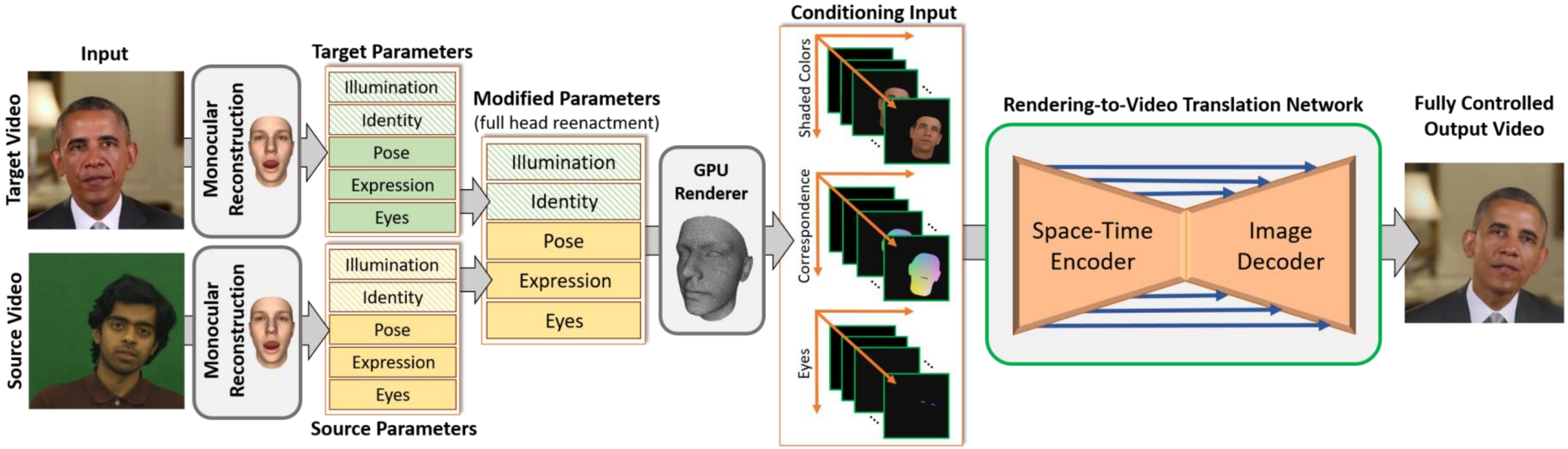
Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits



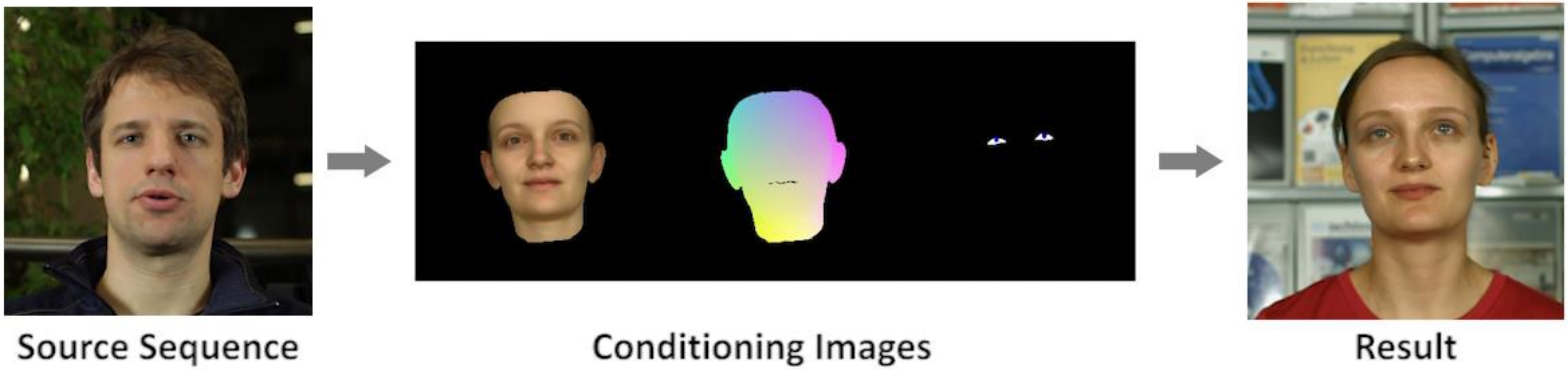
Similar to “Image-to-Image Translation” (Pix2Pix) [Isola et al.]

Deep Video Portraits



Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits



Neural Network converts synthetic data to realistic video

Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits

Source Sequence

Conditioning Images

Result

Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits

Source Sequence | Conditioning Images | Result

Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits

Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits

Interactive Video Editing

2x speed

Siggraph'18 [Kim et al 18]: Deep Portraits

Deep Video Portraits: Insights

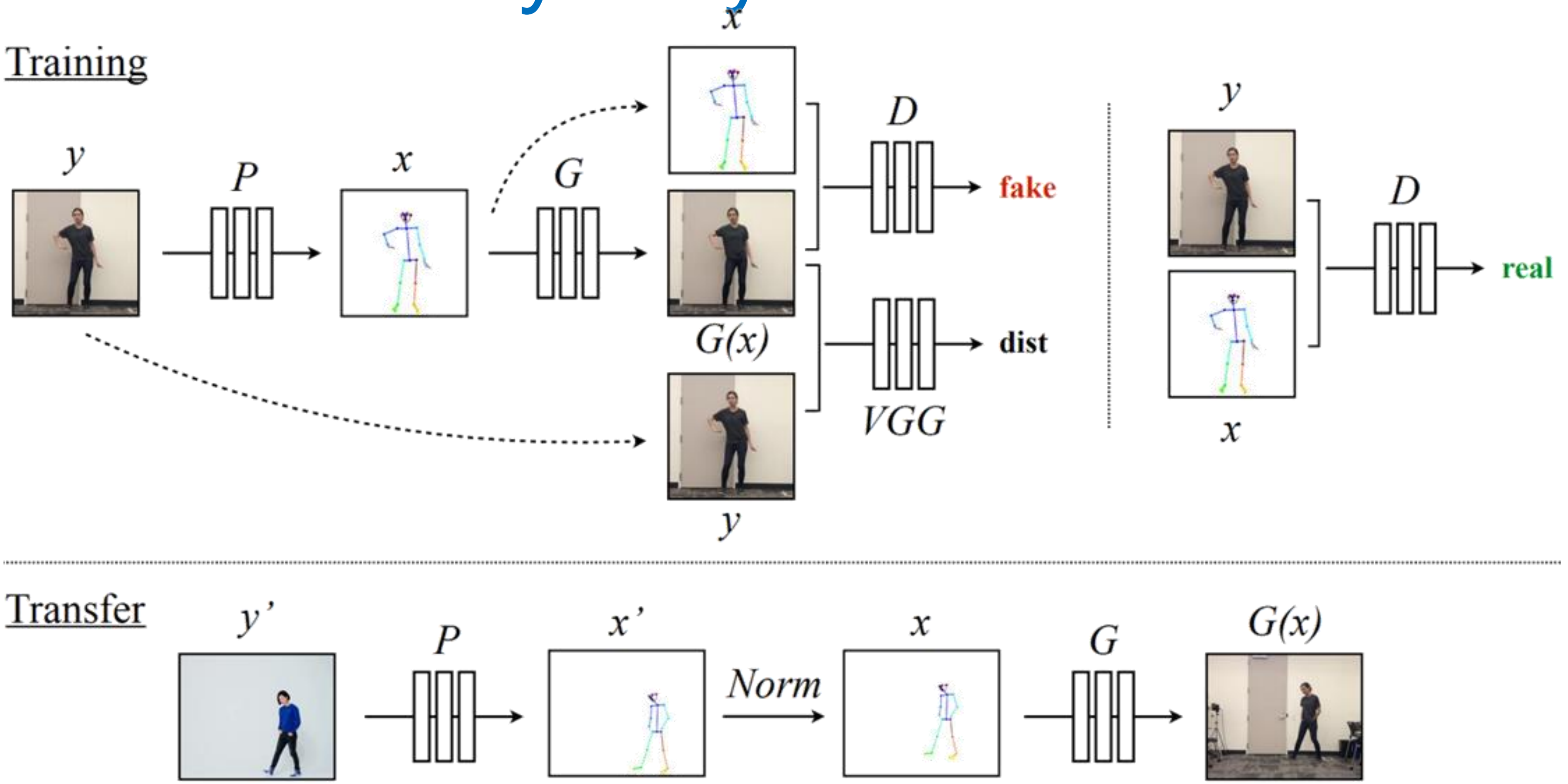
- Synthetic data for tracking is great anchor / stabilizer
- Overfitting on small datasets works pretty well
- Need to stay within training set w.r.t. motions
- No real learning; essentially, optimizing the problem with SGD
  -> should be pretty interesting for future directions

Siggraph'18 [Kim et al 18]: Deep Portraits

Everybody Dance Now

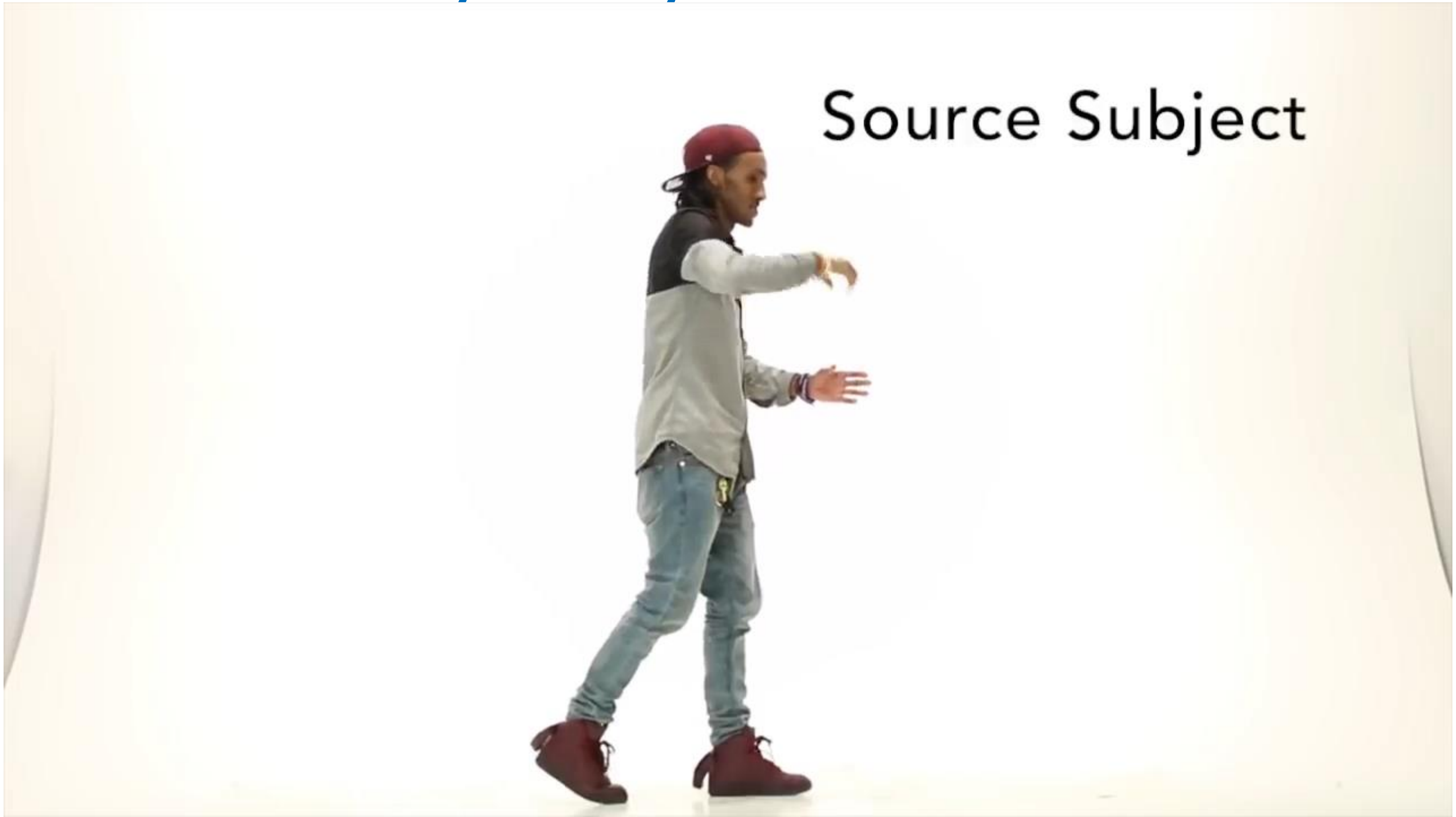
[Chan et al. '18] Everybody Dance Now

Everybody Dance Now



[Chan et al. '18] Everybody Dance Now

Everybody Dance Now



[Chan et al. '18] Everybody Dance Now

Everybody Dance Now: Insights

- Conditioning via tracking seems promising!
  - Tracking quality translates to resulting image quality
  - Tracking human skeletons is less developed than faces
    - Temporally it's not stable… (e.g., OpenPose etc.)
  - Fun fact, there were like 4 papers with a similar same idea that appeared around the same time…

[Chan et al. '18] Everybody Dance Now

Deep Voxels

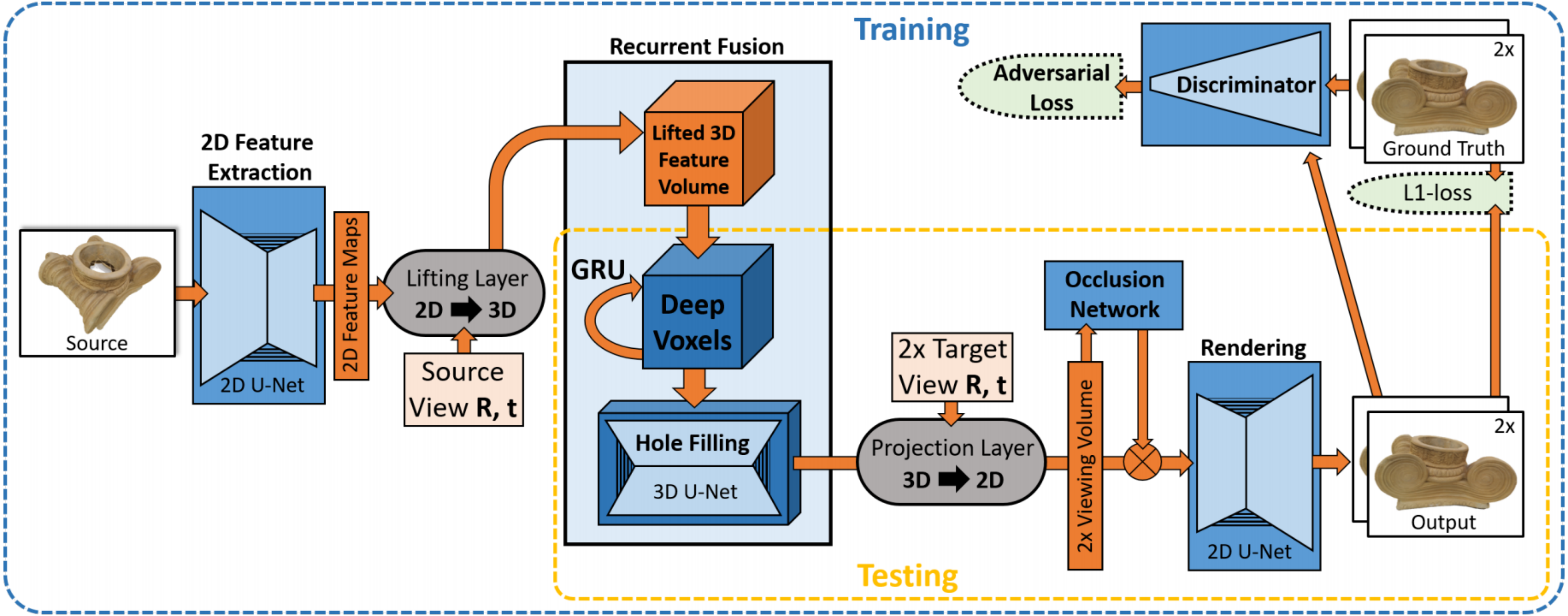
[Sitzmann et al. '18] Deep Voxels

Deep Voxels

- Main idea for video generation:
  - Why learn 3D operations with 2D Convs !?!?
  - We know how 3D transformations work
    - E.g., 6 DoF rigid pose [R | t]
  - Incorporate these into the architectures
    - Need to be differentiable!
  - Example application: novel view point synthesis
    - Given rigid pose, generate image for that view

[Sitzmann et al. '18] Deep Voxels

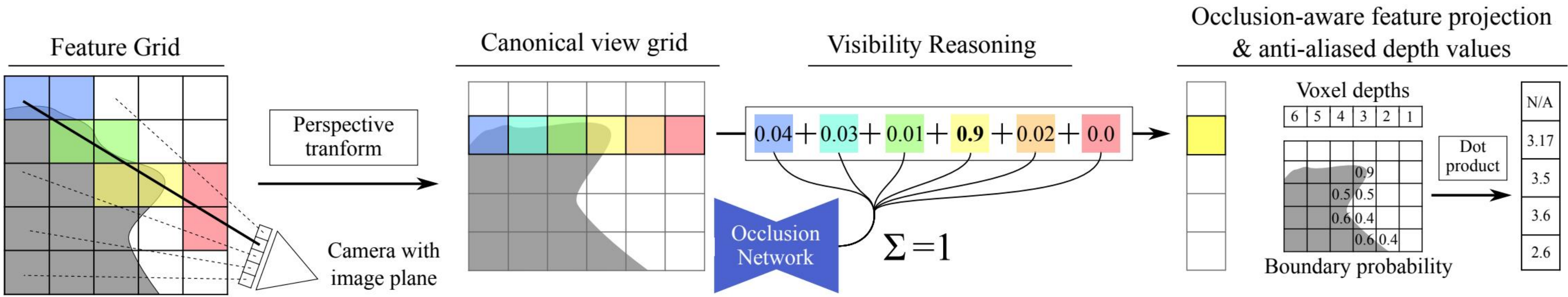
Deep Voxels



[Sitzmann et al. '18] Deep Voxels

Deep Voxels

Occlusion Network:



Issue: we don’ t know the depth for the target!

-> Per-pixel softmax along the ray

-> Network learns the depth

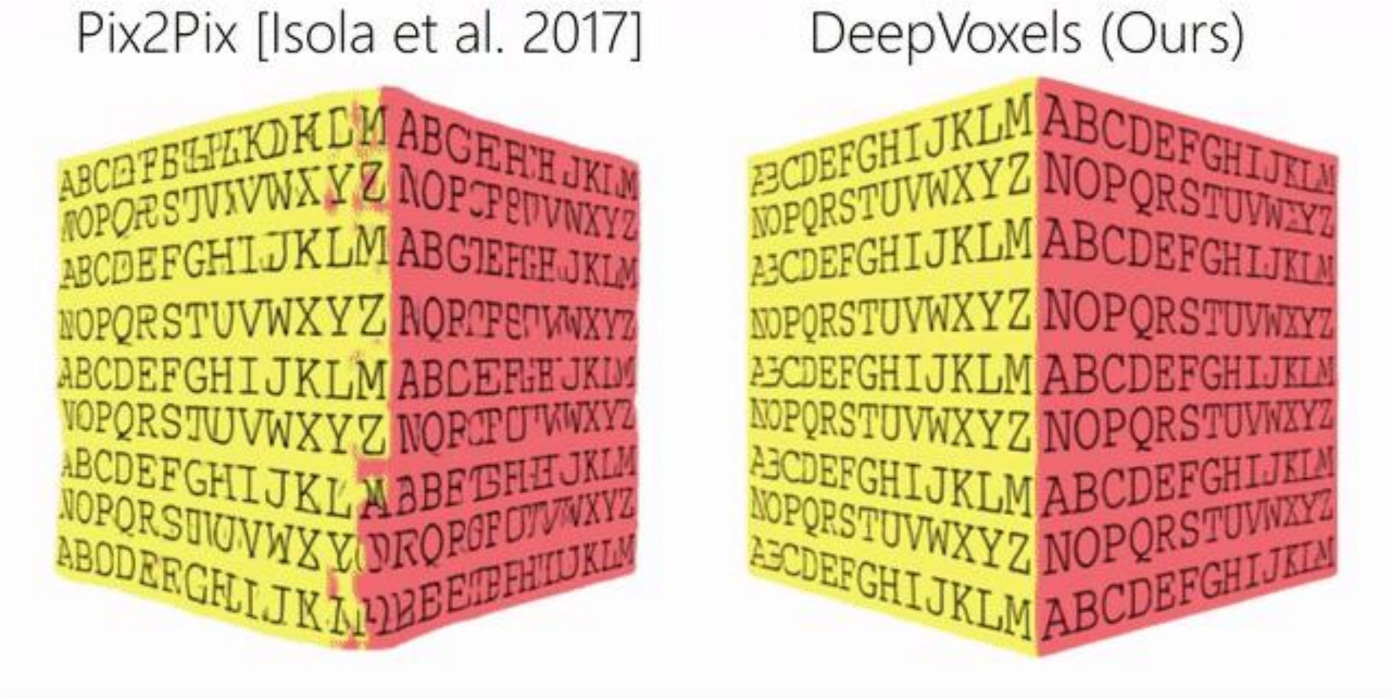
[Sitzmann et al. ’18] Deep Voxels

Deep Voxels

DeepVoxels

Best Baseline: Pix2Pix [Isola et al. 2017]

Deep Voxels



[Sitzmann et al. '18] Deep Voxels

Deep Voxels: Insights

- Lifting from 2D to 3D works great
  - No need to take specific care for temp. coherency!
- All 3D operations are differentiable
- Currently, only for novel view-point synthesis
  - I.e., cGAN for new pose in a given scene

[Sitzmann et al. '18] Deep Voxels

Neural Rendering with Neural Textures

Autoregressive Models

Autoregressive Models vs GANs

- GANs learn implicit data distribution
  - i.e., output are samples (distribution is in model)

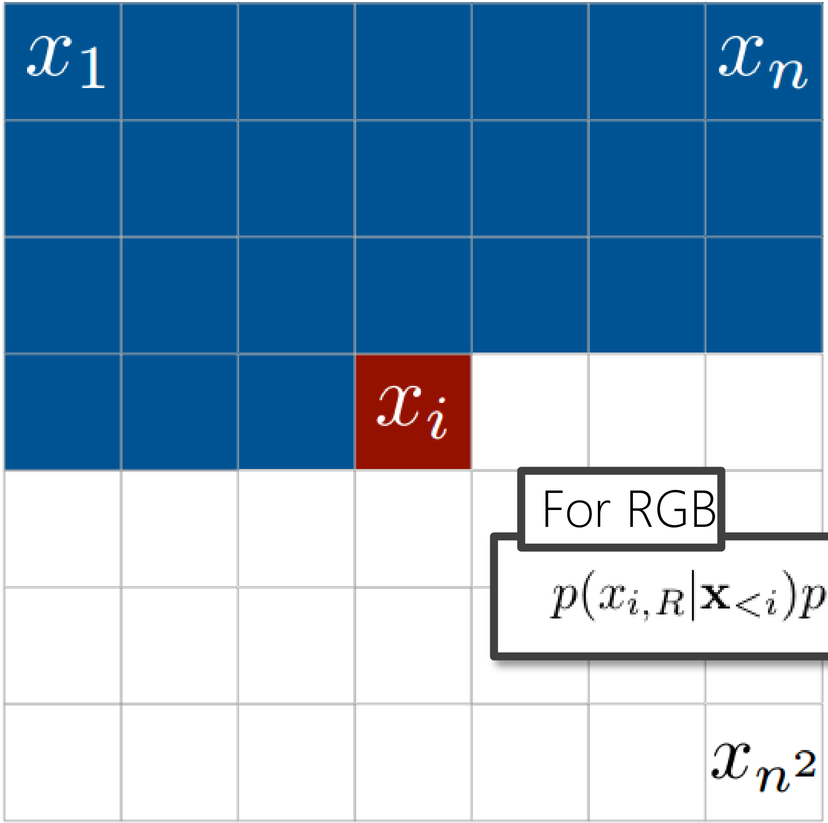
- Autoregressive models learn an explicit distribution governed by a prior imposed by model structure
  - i.e., outputs are probabilities (e.g., softmax)

PixelRNN

- Goal: model distribution of natural images
- Interpret pixels of an image as product of conditional distributions
  - Modeling an image → sequence problem
  - Predict one pixel at a time
  - Next pixel determined by all previously predicted pixels
  - ➢ Use a Recurrent Neural Network

[Van den Oord et al 2016]

PixelRNN



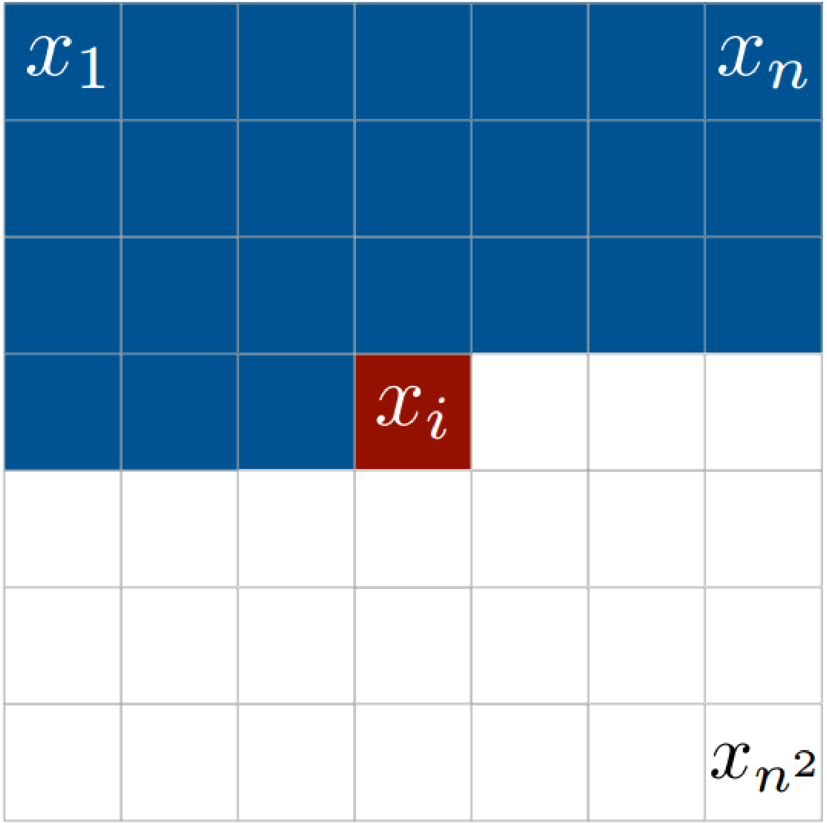
$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

For RGB

$$p(x_{i,R}|\mathbf{x}_{<i})p(x_{i,G}|\mathbf{x}_{<i}, x_{i,R})p(x_{i,B}|\mathbf{x}_{<i}, x_{i,R}, x_{i,G})$$

[Van den Oord et al 2016]

PixelRNN



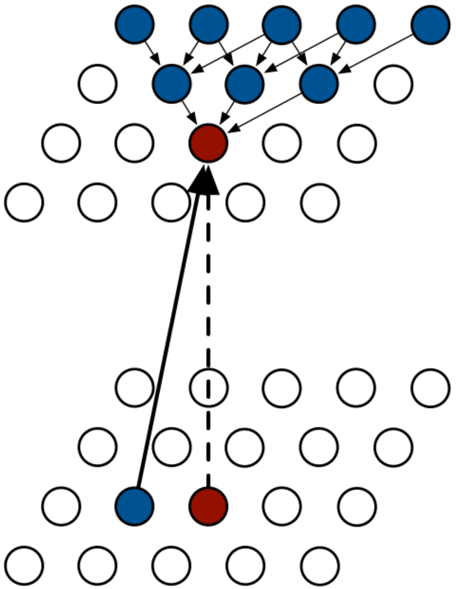
$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

$x_i \in [0,255]$
→ 256-way softmax

[Van den Oord et al 2016]

PixelRNN

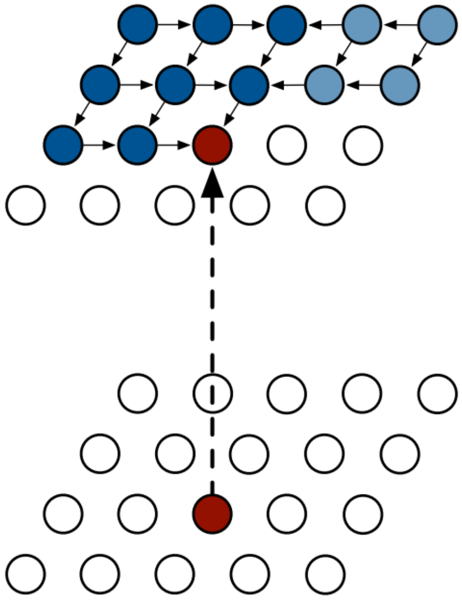
- Row LSTM model architecture
- Image processed row by row
- Hidden state of pixel depends on the 3 pixels above it
  - Can compute pixels in row in parallel
- Incomplete context for each pixel



[Van den Oord et al 2016]

PixelRNN

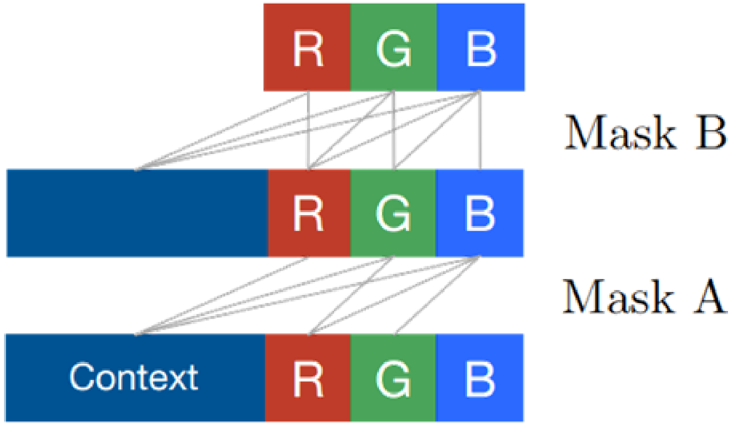
- Diagonal BiLSTM model architecture
- Solve incomplete context problem
- Hidden state of pixel $p_{i,j}$ depends on $p_{i,j-1}$ and $p_{i-1,j}$
- Image processed by diagonals



[Van den Oord et al 2016]

PixelRNN

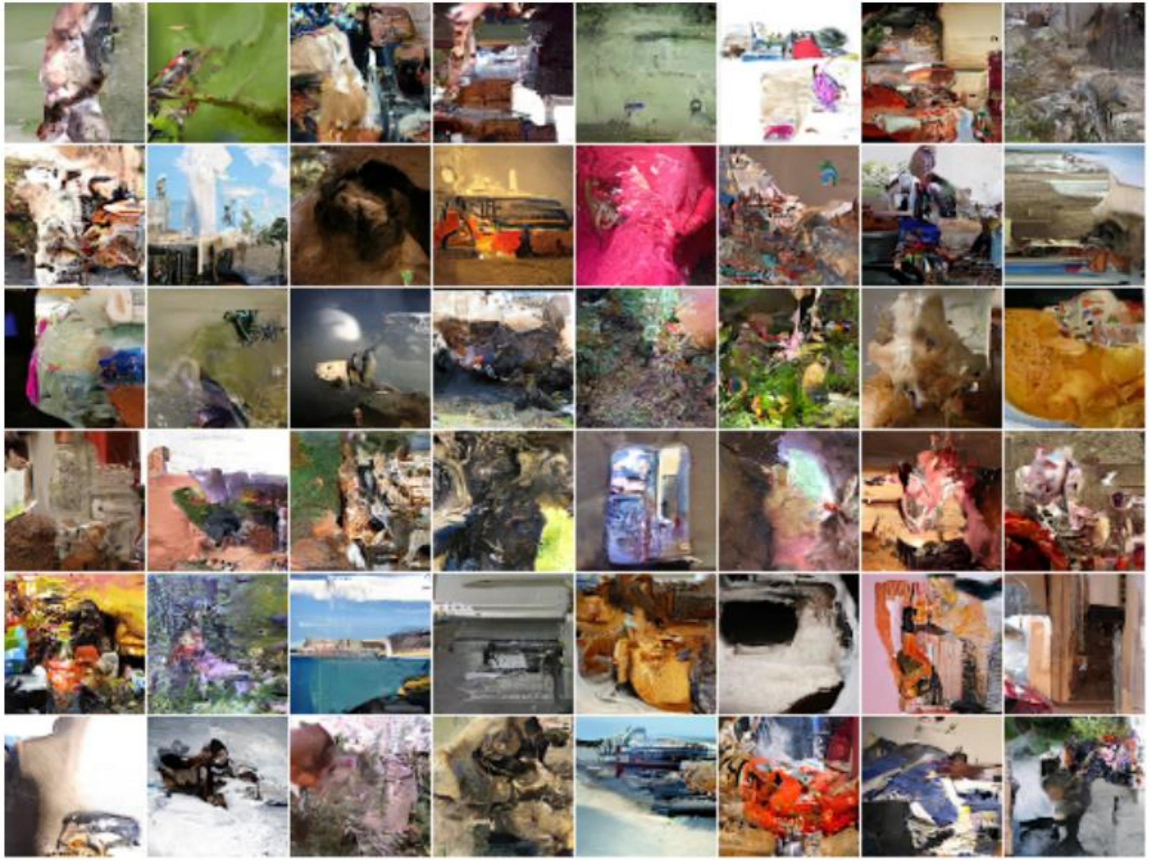
- Masked Convolutions
- Only previously predicted values can be used as context
- Mask A: restrict context during 1st conv
- Mask B: subsequent convs
- Masking by zeroing out values



[Van den Oord et al 2016]

PixelRNN

- Generated 64x64 images, trained on ImageNet



[Van den Oord et al 2016]

PixelCNN

- Row and Diagonal LSTM layers have potentially unbounded dependency range within the receptive field
  - Can be very computationally costly

➢ PixelCNN:
  - standard convs capture a bounded receptive field
  - All pixel features can be computed at once (during training)

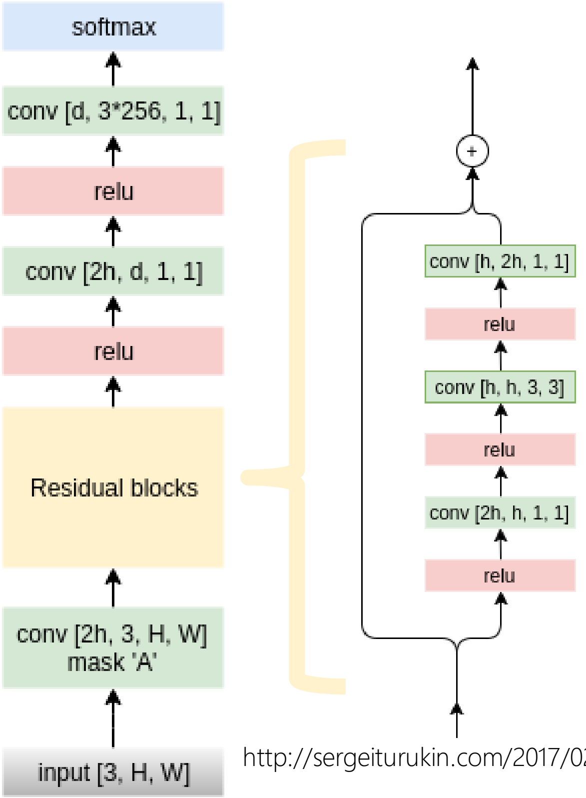
[Van den Oord et al 2016]

PixelCNN

- Model preserves spatial dimensions
- Masked convolutions to avoid seeing future context

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

Mask A



http://sergeiturukin.com/2017/02/22/pixelcnn.htm

[Van den Oord et al 2016]

Gated PixelCNN

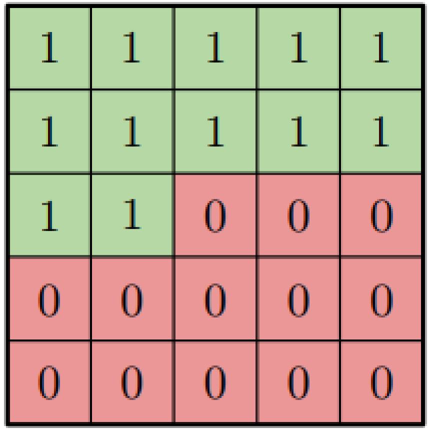
- Gated blocks
- Imitate multiplicative complexity of PixelRNNs to reduce performance gap between PixelCNN and PixelRNN
- Replace ReLU with gated block of sigmoid, tanh

k^th layer sigmoid

$$y = \tanh\left(W_{k,f} * x\right) \odot \sigma(W_{k,g} * x)$$

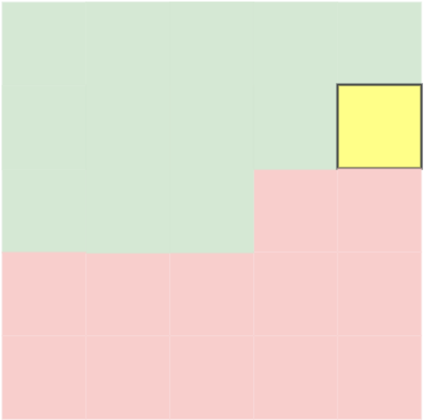
element-wise product convolution

PixelCNN Blind Spot



5x5 image / 3x3 conv

Receptive Field



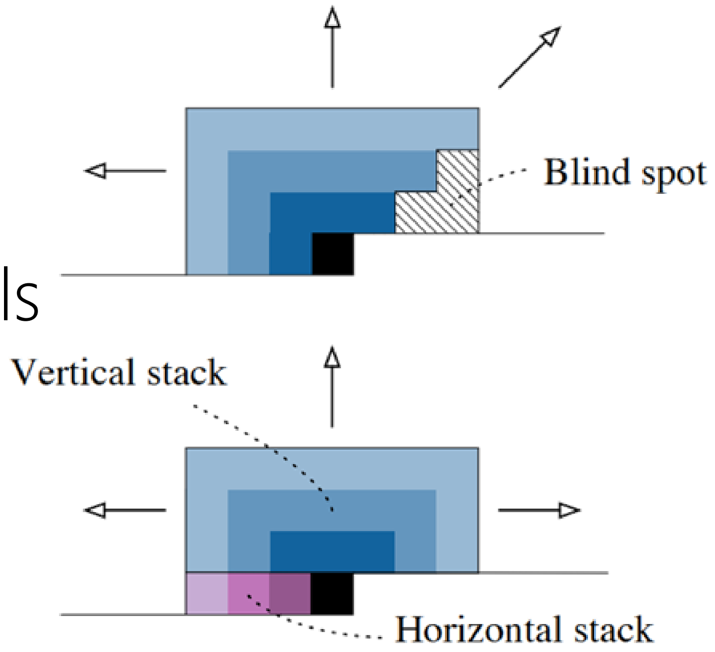
Unseen context

http://sergeiturukin.com/2017/02/24/gated-pixelcnn

[Van den Oord et al 2016]

PixelCNN: Eliminating Blind Spot

- Split convolution to two stacks
- Horizontal stack conditions on current row
- Vertical stack conditions on pixels above



Conditional PixelCNN

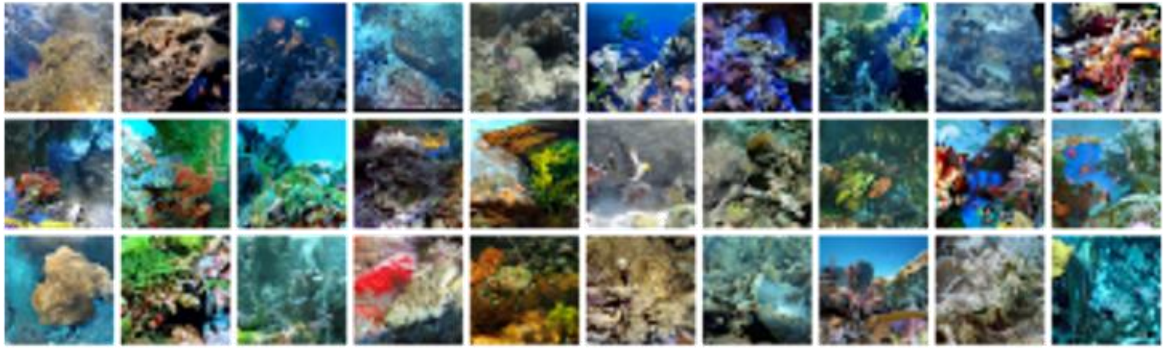
- Conditional image generation
- E.g., condition on semantic class, text description

latent vector to be conditioned on

$$y = \tanh\left(W_{k,f} * x + V_{k,f}^{T} h\right) \odot \sigma(W_{k,g} * x + V_{k,g}^{T} h)$$

[Van den Oord et al 2016]

Conditional PixelCNN



Coral Reef

Sorrel horse

[Van den Oord et al 2016]

Autoregressive Models vs GANs

- Advantages of autoregressive:
  - Explicitly model probability densities
  - More stable training
  - Can be applied to both discrete and continuous data
- Advantages of GANs:
  - Have been empirically demonstrated to produce higher quality images
  - Faster to train

Deep Learning in Higher Dimensions

Multi-Dimensional ConvNets

- 1D ConvNets
  - Audio / Speech
  - Also Point Clouds
- 2D ConvNets
  - Images (AlexNet, VGG, ResNet -> Classification, Localization, etc..)
- 3D ConvNets
  - For videos
  - For 3D data
- 4D ConvNets
  - E.g., dynamic 3D data (Haven' t seen much work there)
  - Simulations

Remember: 1D Convolutions

4	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**

f

g	**1/3**	**1/3**	**1/3**

$f * g$		**3**								

$$4 \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} = 3$$

Remember: 1D Convolutions

4	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**

f

g

1/3	**1/3**	**1/3**

$f * g$

	3	**0**							

$$3 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} + (-5) \cdot \frac{1}{3} = 0$$

Remember: 1D Convolutions

4	3	2	-5	3	5	2	5	5	6

f

g

1/3	1/3	1/3

$f * g$

	3	0	0						

$$2 \cdot \frac{1}{3} + (-5) \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} = 0$$

Remember: 1D Convolutions

4	3	2	-5	3	5	2	5	5	6

f

g

1/3	1/3	1/3

$f * g$

	3	0	0	1					

$$(-5) \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} = 1$$

Remember: 1D Convolutions

4	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**

f

g

1/3	**1/3**	**1/3**

$f * g$

	3	**0**	**0**	**1**	**10/3**				

$$3 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} = \frac{10}{3}$$

Remember: 1D Convolutions

4	3	2	-5	3	5	2	5	5	6

f

g

1/3	1/3	1/3

$f * g$

	3	0	0	1	10/3	4			

$$5 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} = 4$$

Remember: 1D Convolutions

4	3	2	-5	3	5	2	5	5	6

f

g

1/3	1/3	1/3

$f * g$

	3	0	0	1	10/3	4	4		

$$2 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} = 4$$

Remember: 1D Convolutions

4	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**

f

g

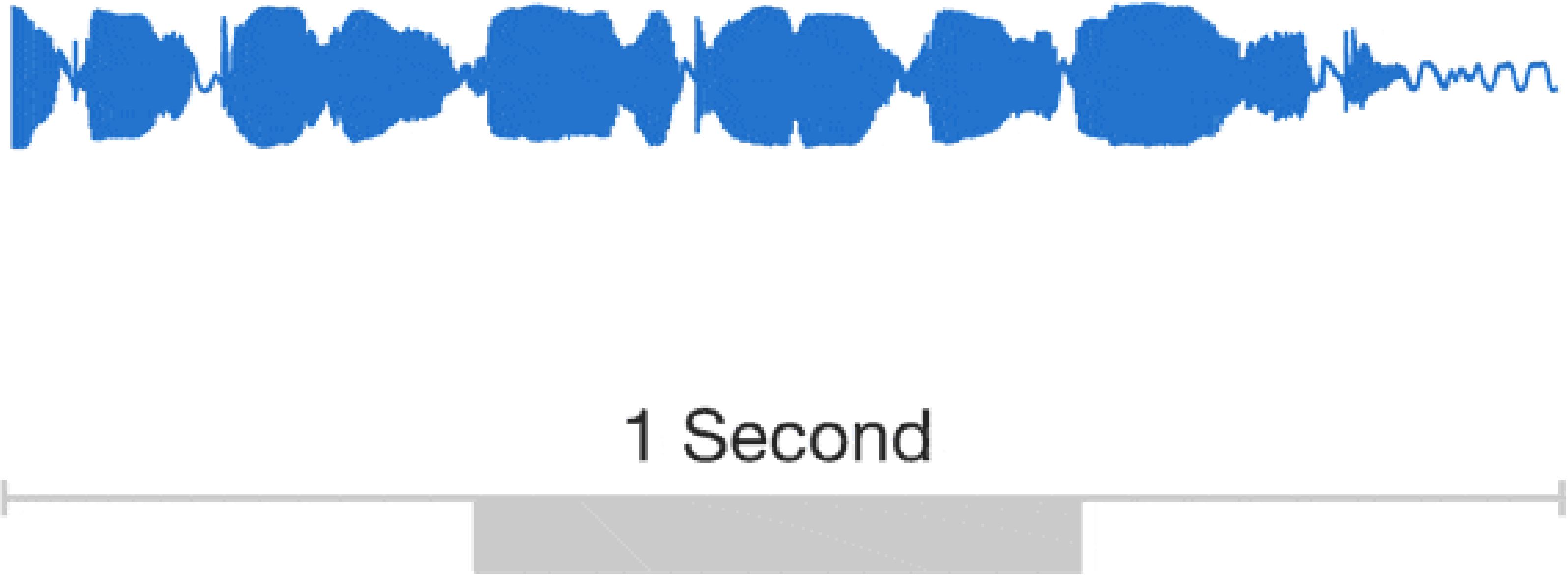
1/3	**1/3**	**1/3**

$f * g$

	3	**0**	**0**	**1**	**10/3**	**4**	**4**	**16/3**	

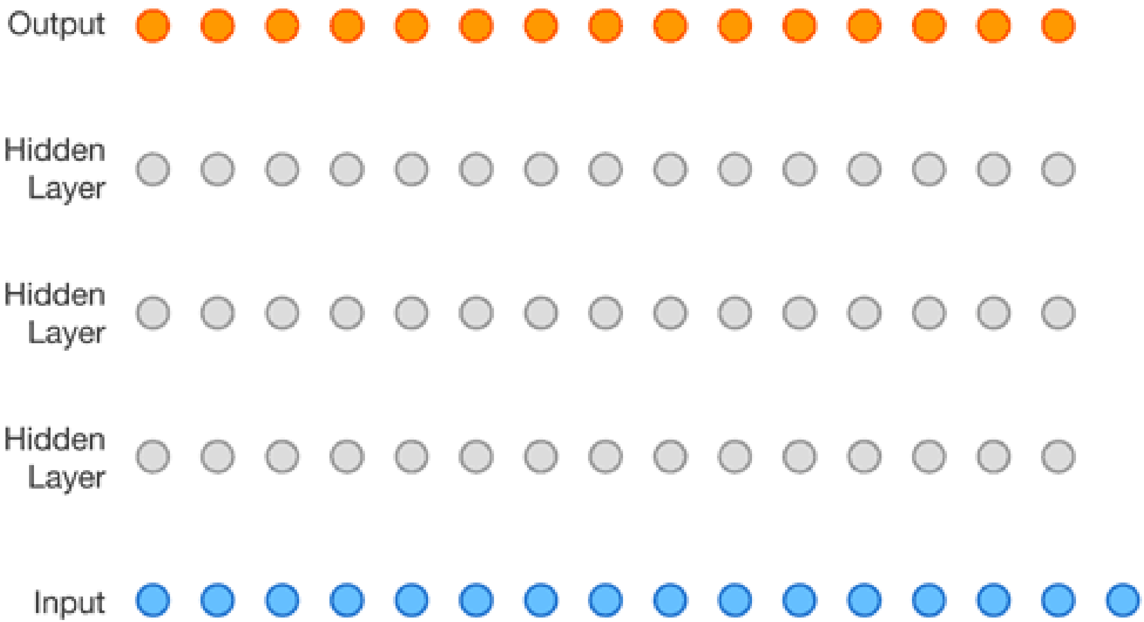
$$5 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 6 \cdot \frac{1}{3} = \frac{16}{3}$$

1D ConvNets: WaveNet



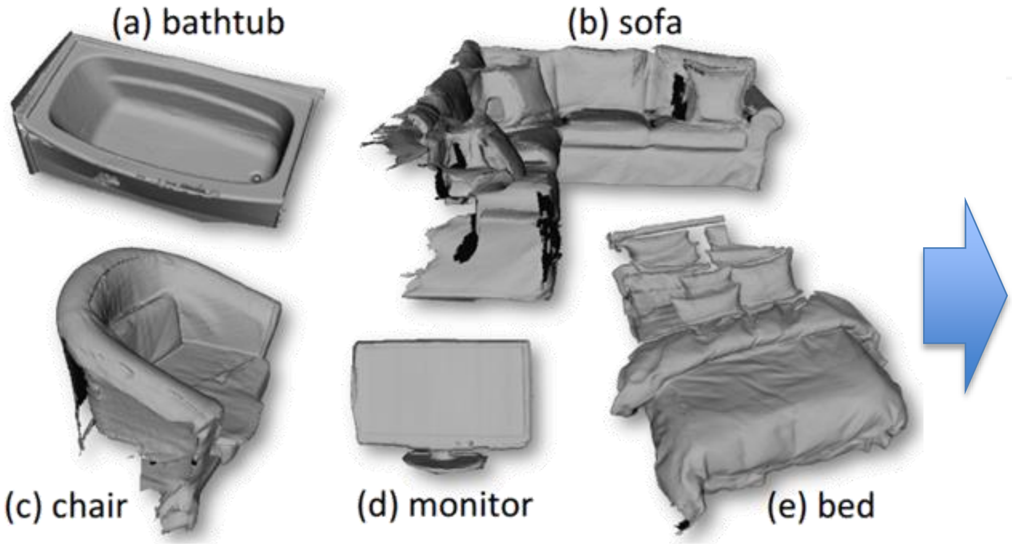
[van der Ooord 16] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

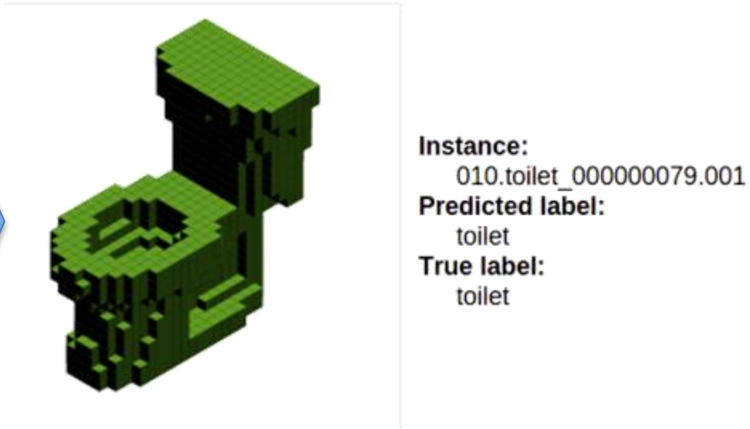
1D ConvNets: WaveNet



[van der Ooord 16] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

3D Classification

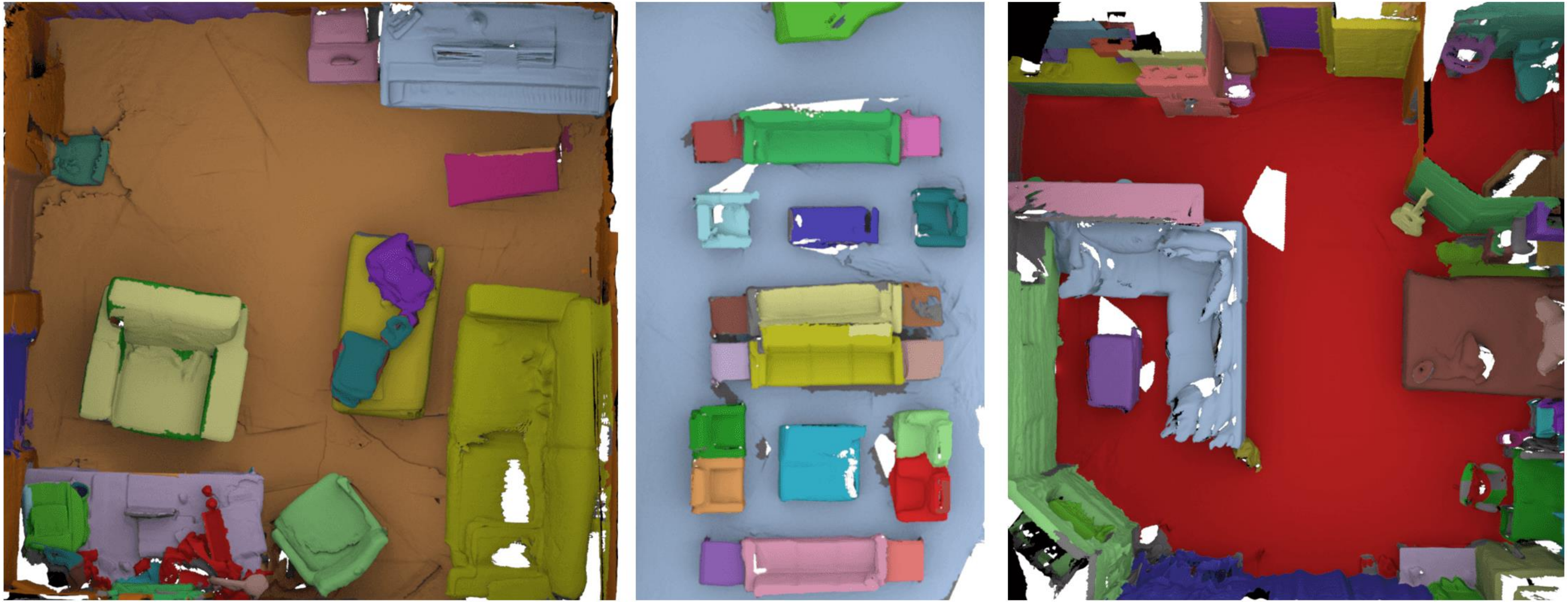




Class from 3D model (e.g., obtained with Kinect Scan)

[Maturana et al. 15] & [Qi et al. 16] 3D vs Multi-view

3D Semantic Segmentation



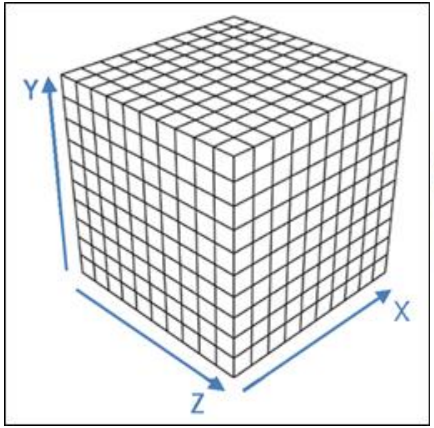
1500 densely annotated 3D scans; 2.5 mio RGB-D frames

[Dai et al. 17] ScanNet

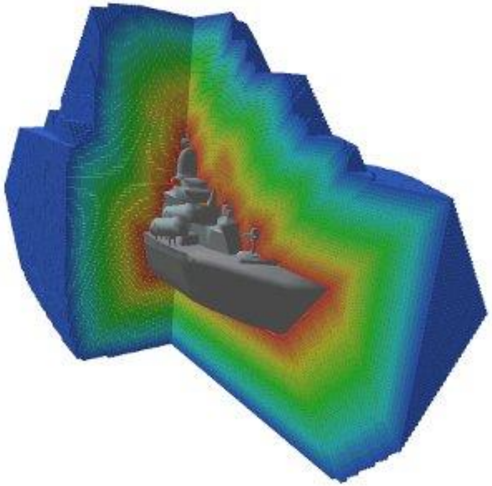
Volumetric Grids

Volumetric Grids

Volumetric Data Structures
- Occupancy grids
- Ternary grids
- Distance Fields
- Signed Distance fields



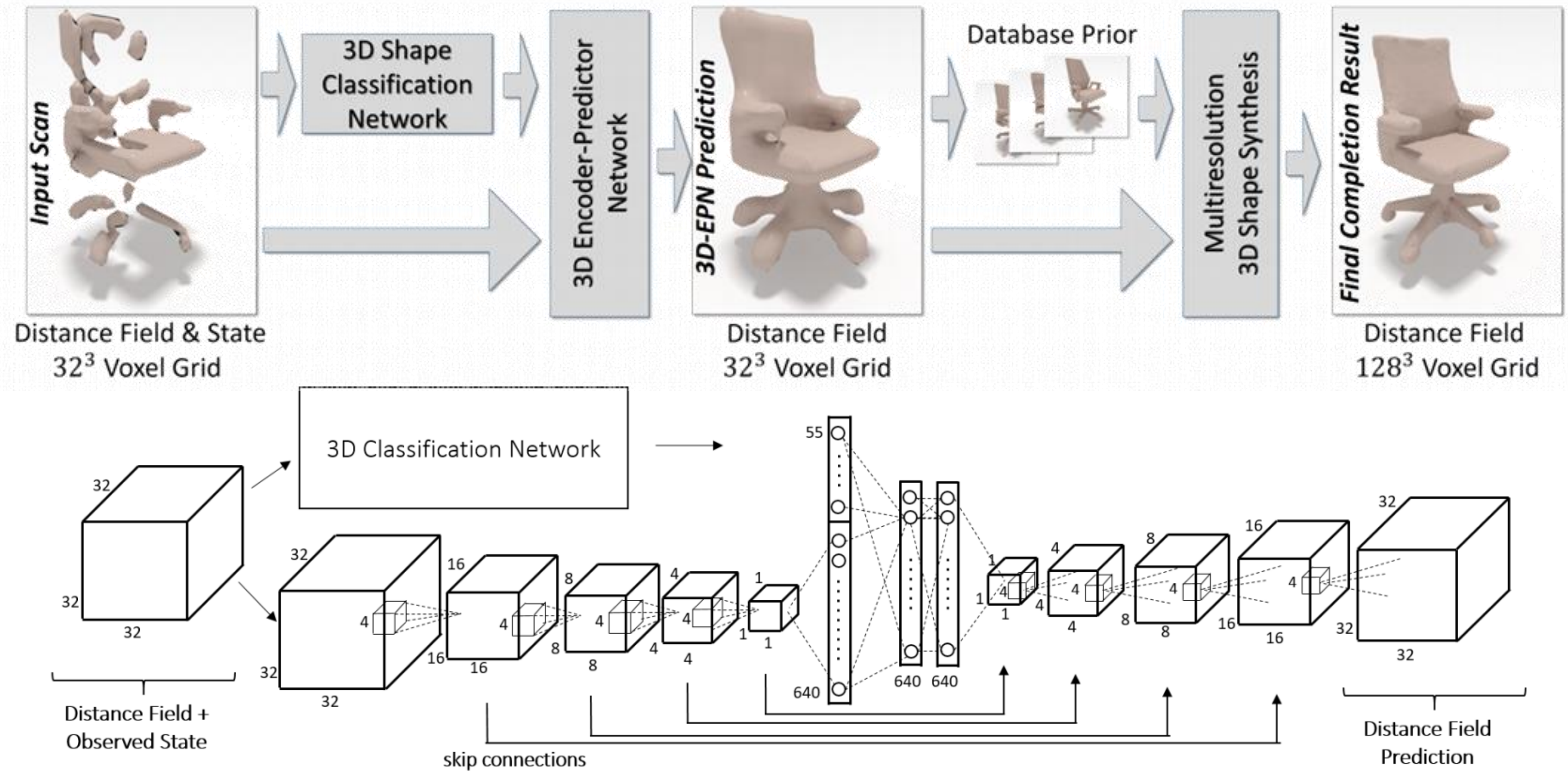
(binary) Voxel Grid



Method	ℓ_1-Err (32^3)	ℓ_1-Err (128^3)
Ours (3D-EPN + synth)	0.382	1.94
Ours (3D-EPN-class + synth)	0.376	1.93
Ours (3D-EPN-unet + synth)	0.310	1.82
Ours (final) 3D-EPN-unet-class + synth	**0.309**	**1.80**

Shape completion error (higher == better)

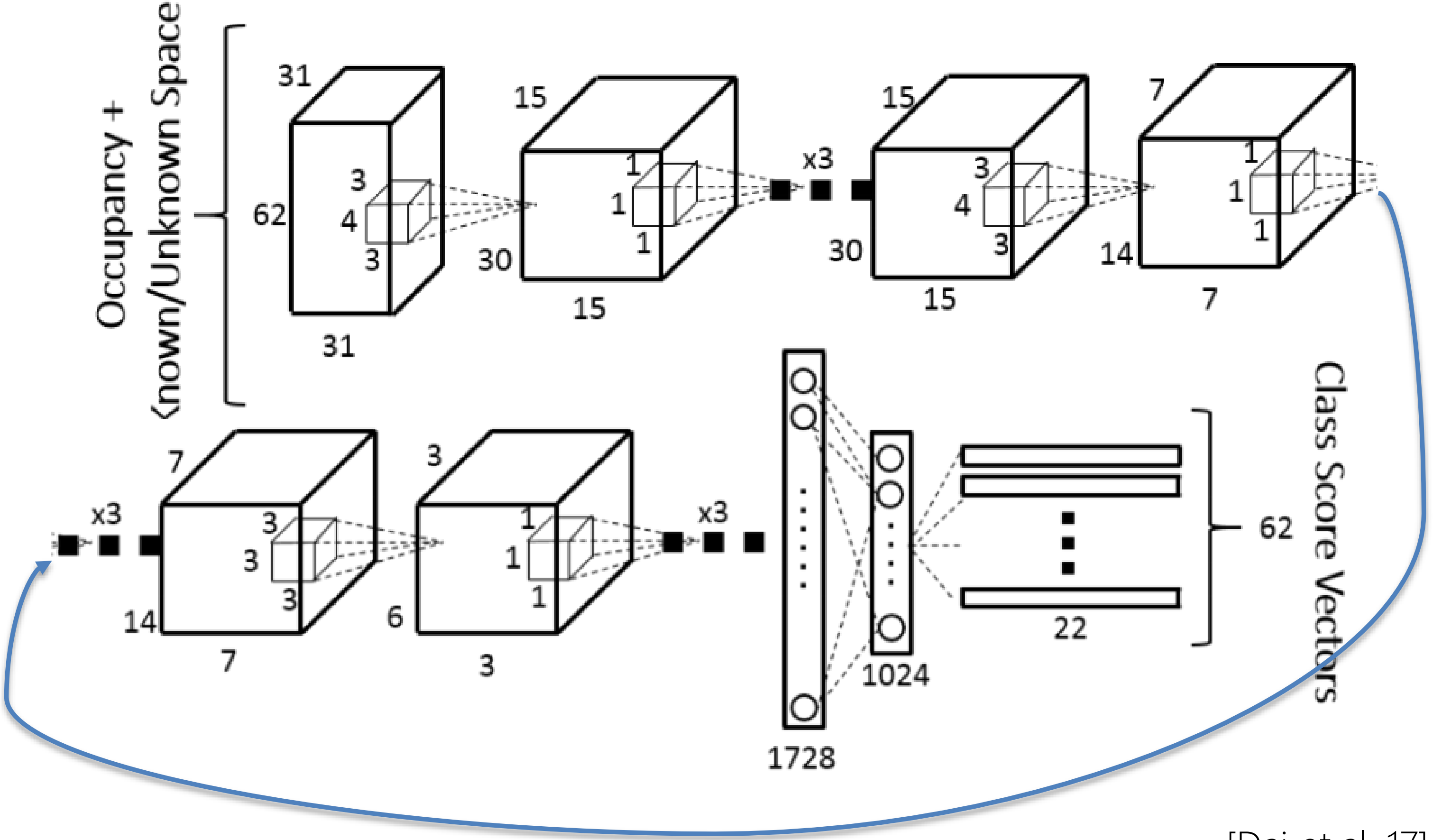
3D Shape Completion on Grids



Works with 32 x 32 x 32 voxels…

[Dai et al. 17] CNNComplete

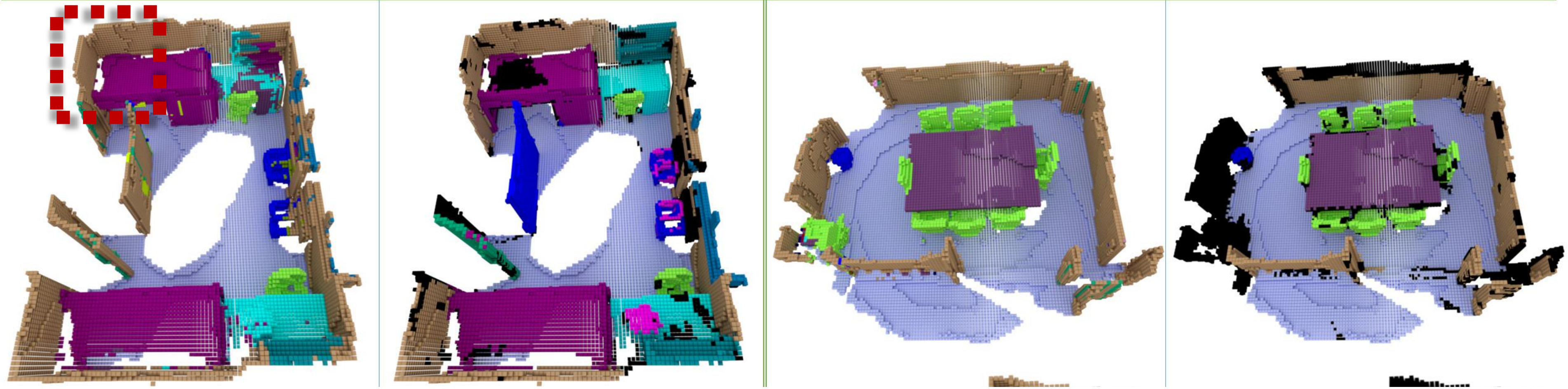
ScanNet: Semantic Segmentation in 3D



[Dai et al. 17] ScanNet

ScanNet: Sliding Window

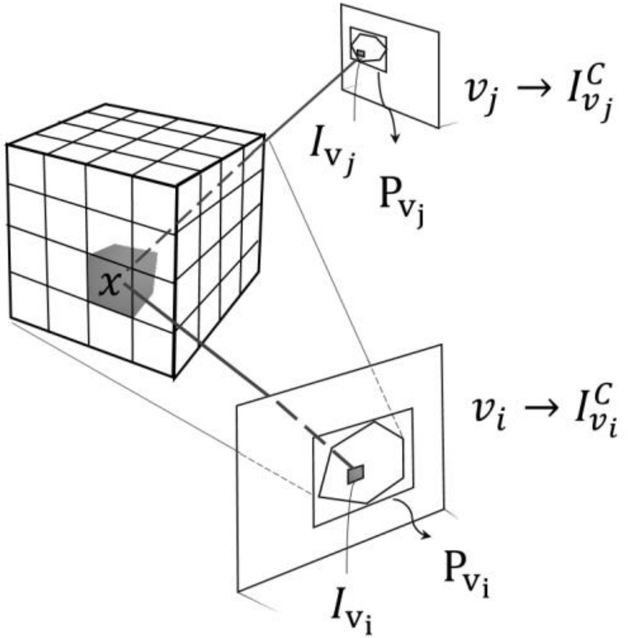
Voxel Predictions	Ground Truth	Voxel Predictions	Ground Truth

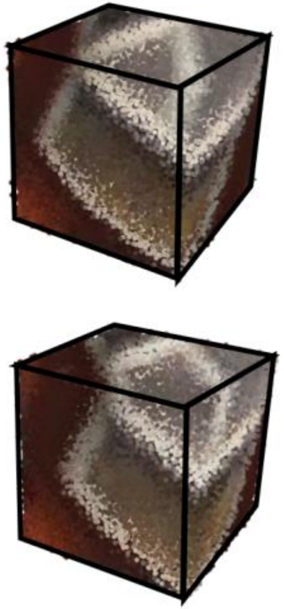


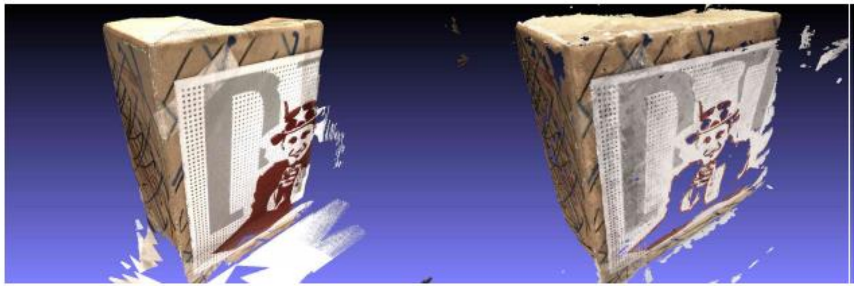
unannotated	wall	floor	chair	table	desk	bed	bookshelf	sofa	sink	bathtub	toilet
curtain	counter	door	window	shower curtain	refrigerator	picture	cabinet	otherfurniture			

[Dai et al. 17] ScanNet

SurfaceNet: Stereo Reconstruction





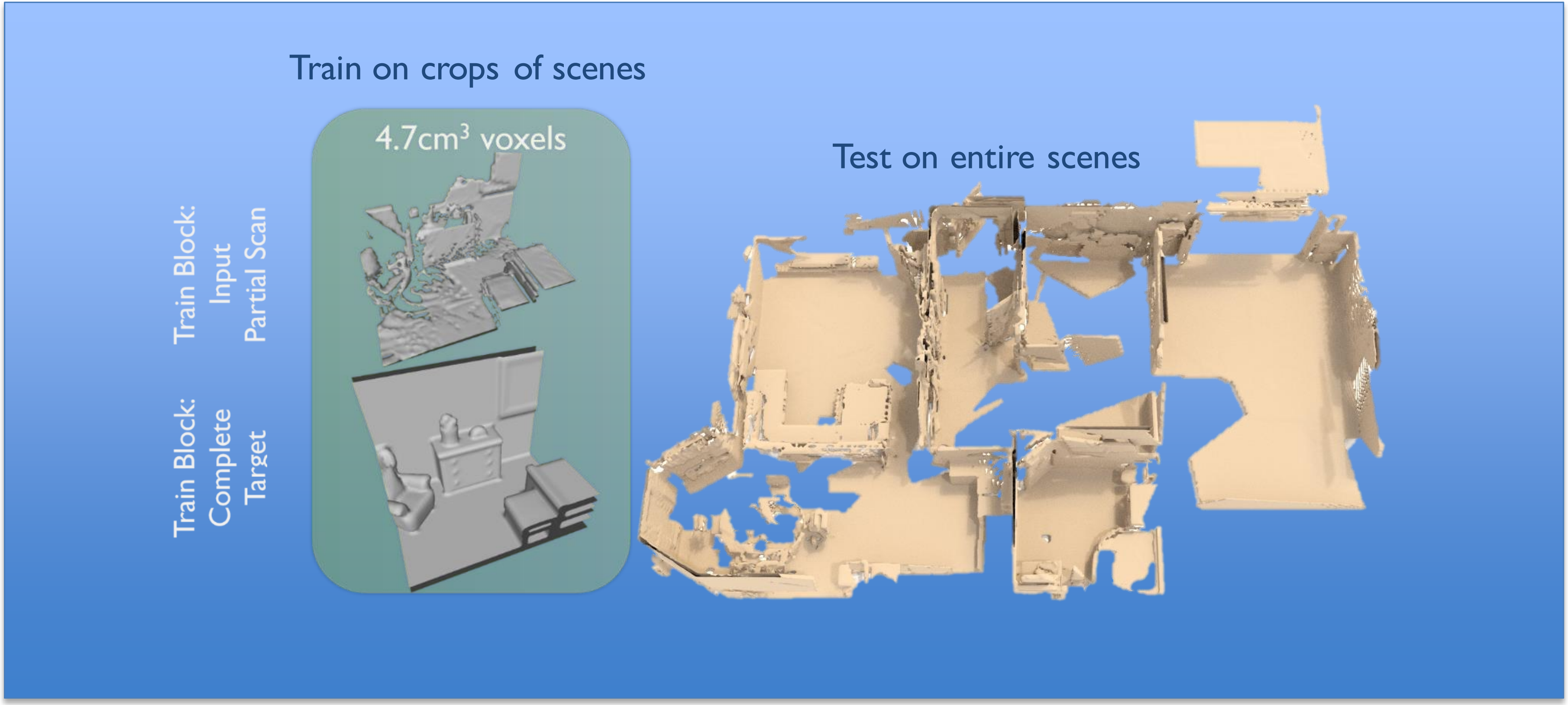


(a) reference model (b) **SurfaceNet**

Run on 32 x 32 x 32 blocks -> takes forever...

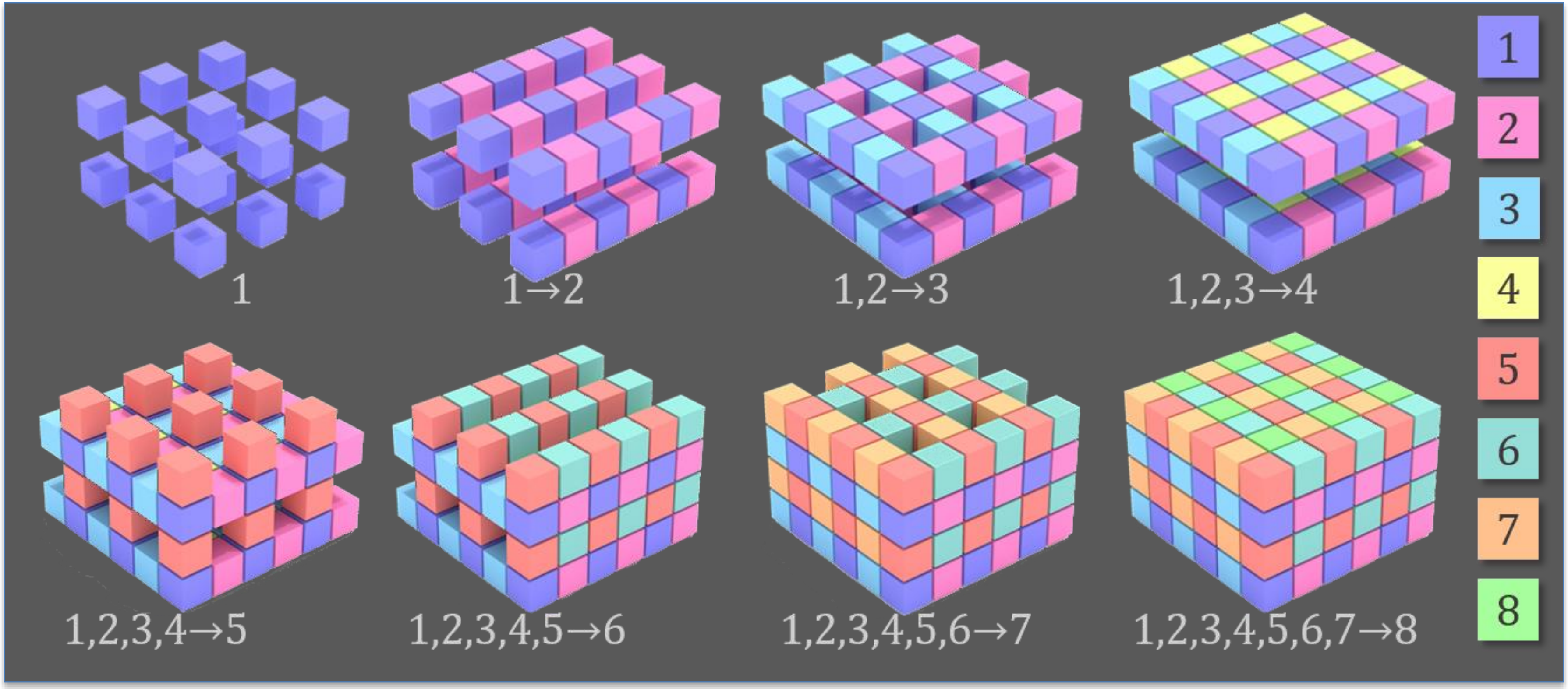
[Ji et al. 17] SurfaceNet

ScanComplete: Fully Convolutional



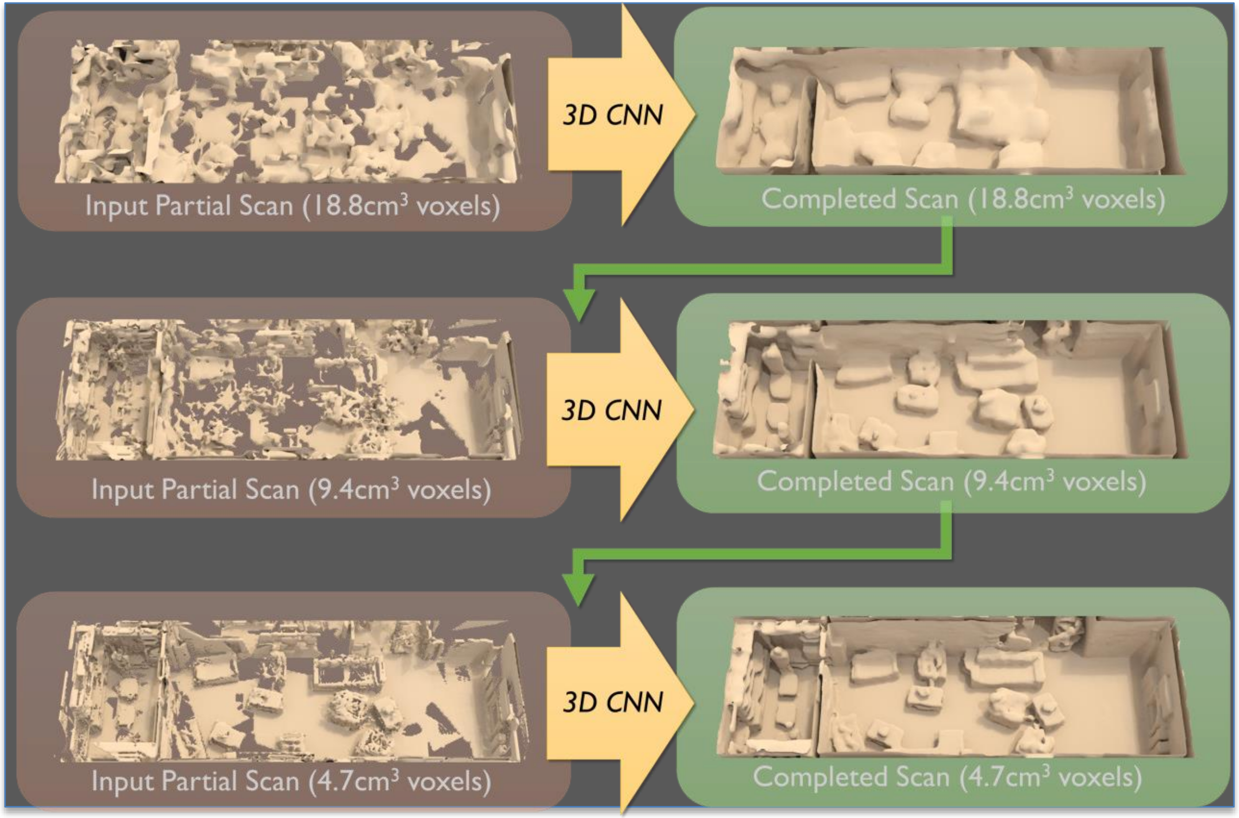
[Dai et al. 18] ScanComplete

Dependent Predictions: Autoregressive Neural Networks



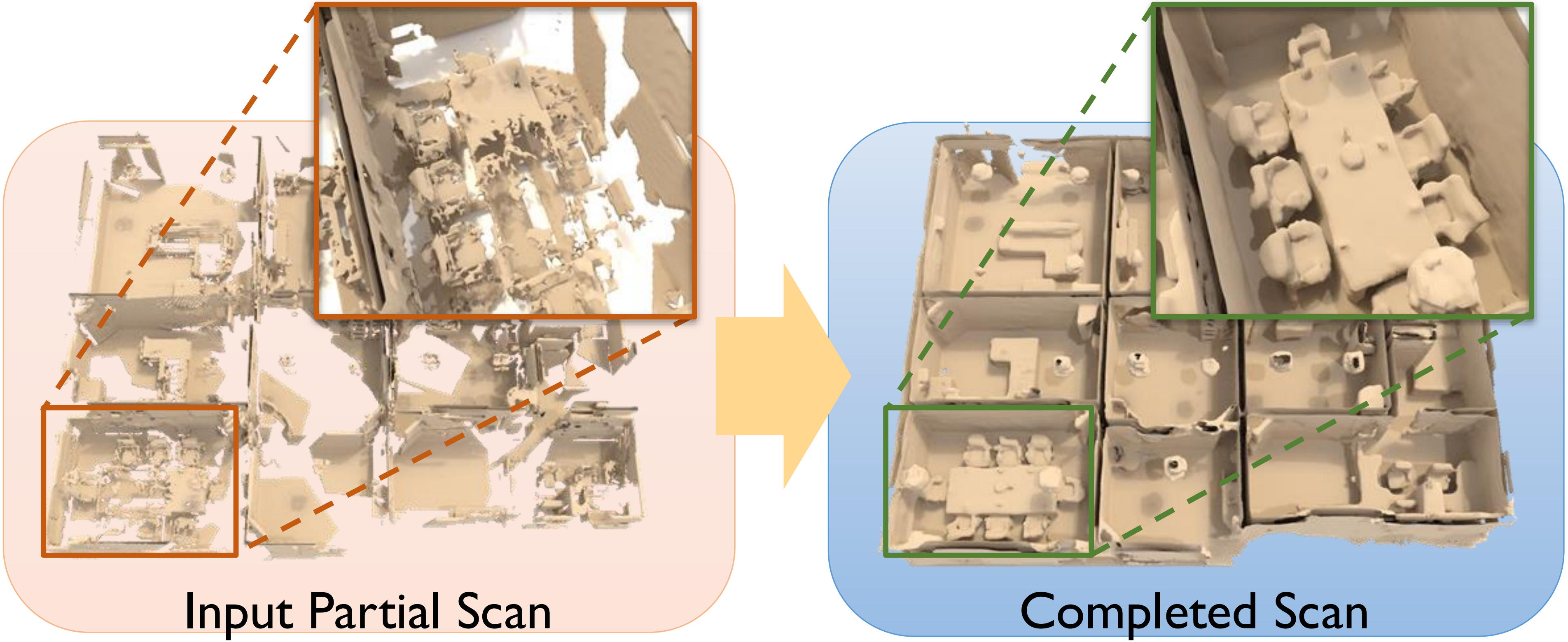
[Dai et al.]: ScanComplete

Spatial Extent: Coarse-to-Fine Predictions



[Dai et al.]: ScanComplete

ScanComplete: Fully Convolutional

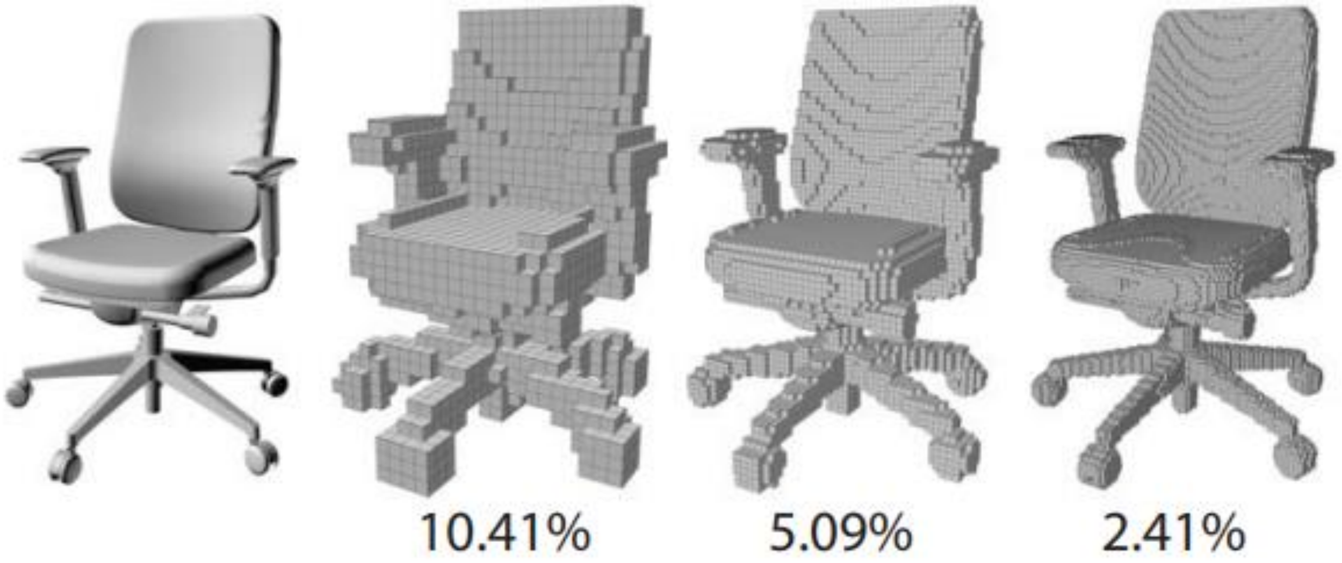


[Dai et al. 18] ScanComplete

Conclusion so far

- Volumetric Grids are easy
  - Encode free space
  - Encode distance fields
  - Need a lot of memory
  - Need a lot of processing time
  - But can be used sliding window or fully-conv.

Conclusion so far



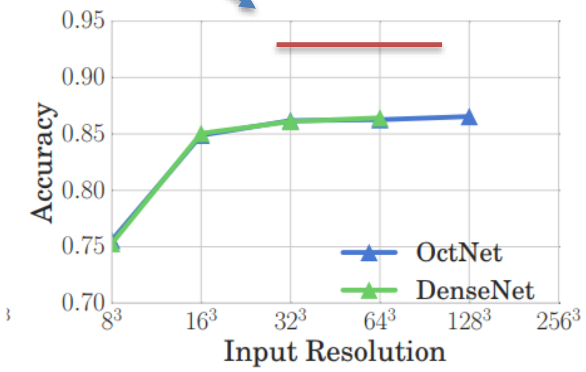
Surface occupancy gets smaller with higher resolutions

Volumetric Hierarchies

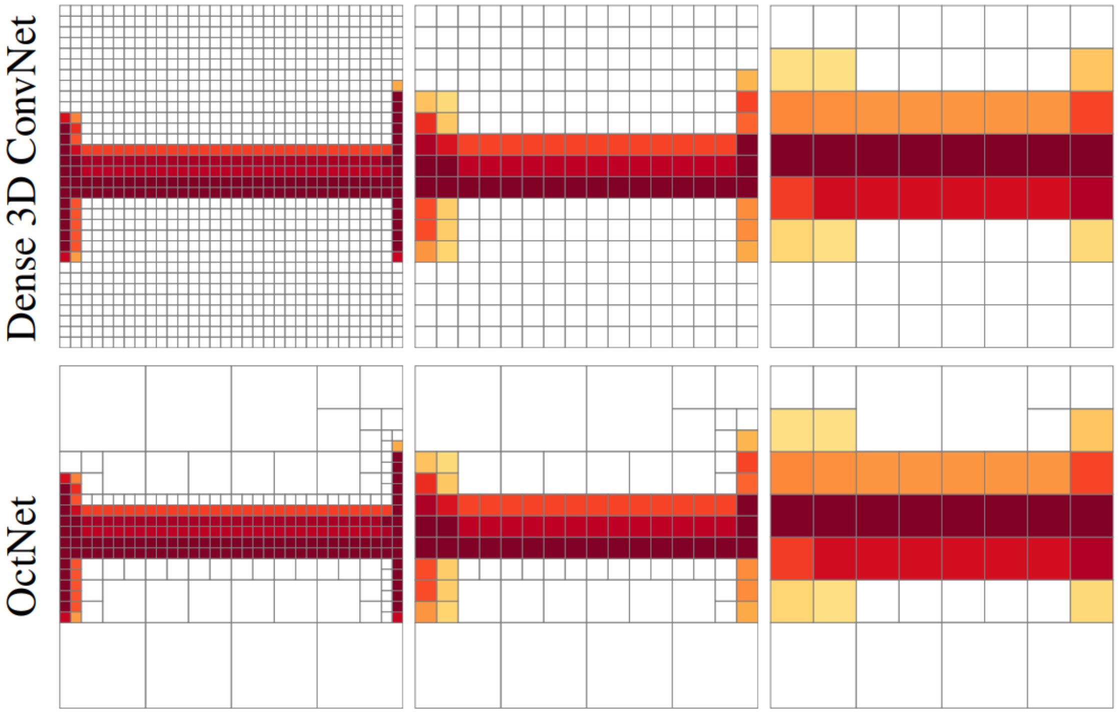
Discriminative Tasks

Structure is known in advance!

State of the art is somewhere here...



(b) Accuracy



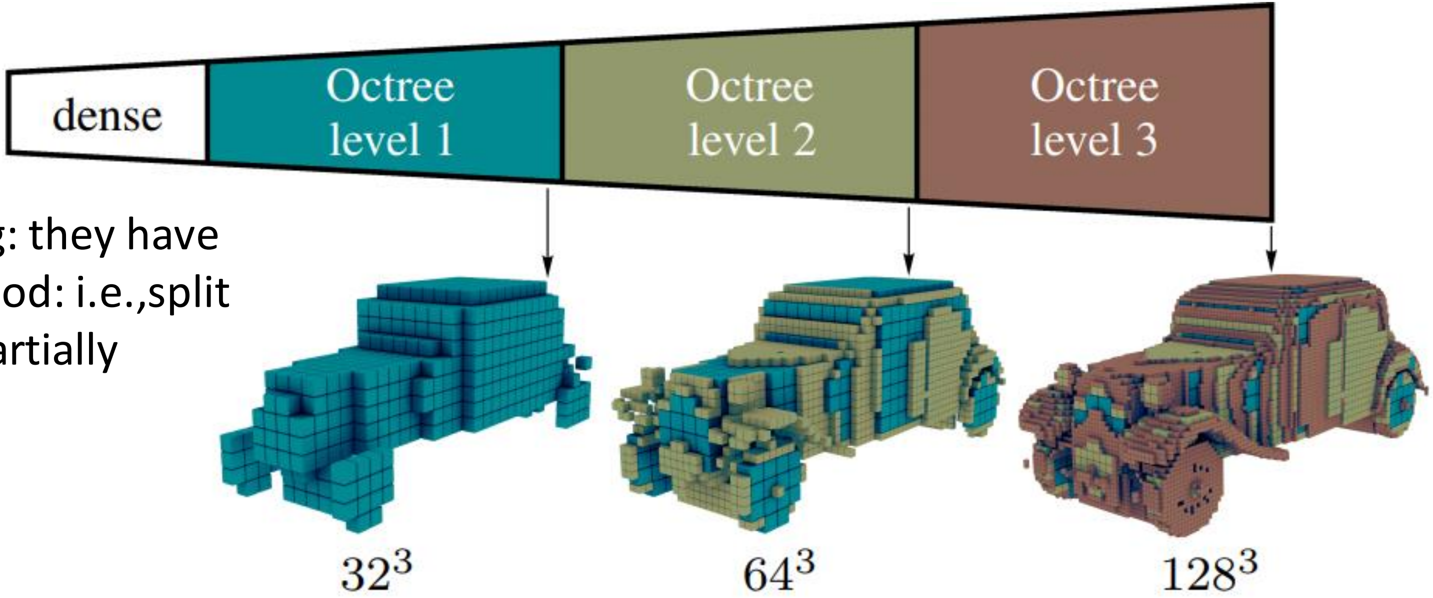
(a) Layer 1: 32^3 (b) Layer 2: 16^3 (c) Layer 3: 8^3

[OctNet: Learning Deep 3D Representations at High Resolutions] (CVPR 2017)

[O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis] (SIG17)

Generative Tasks

Need to infer structure!



Pretty interesting: they have end-to-end method: i.e.,split voxels that are partially occupied

[Octree Generating Networks: Efficient Convolutional Architectures for High-resolution Outputs]()

[OctNetFusion: Learning Depth Fusion from Data]() (that one not end to end)

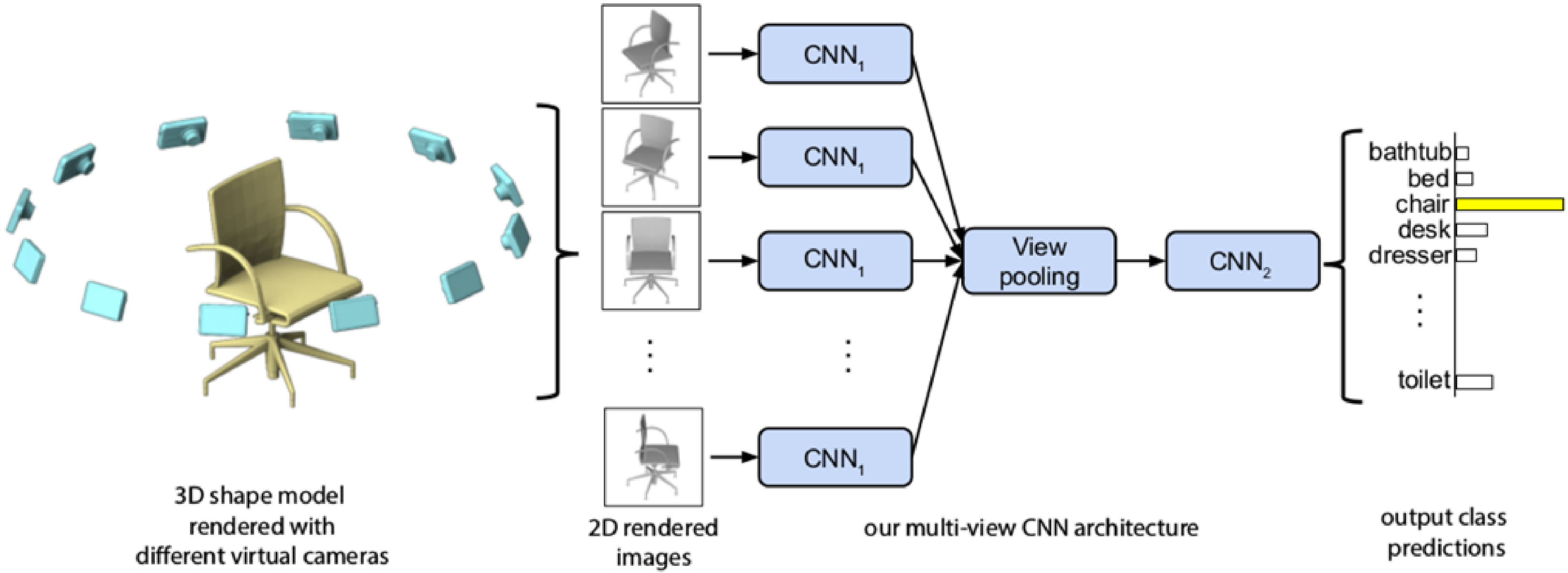
Conclusion so far

- Hierarchies
  - are great for reducing memory and runtime
  - Comes at a performance hit
  - Easier for discriminative tasks when structure is known

Multi-view

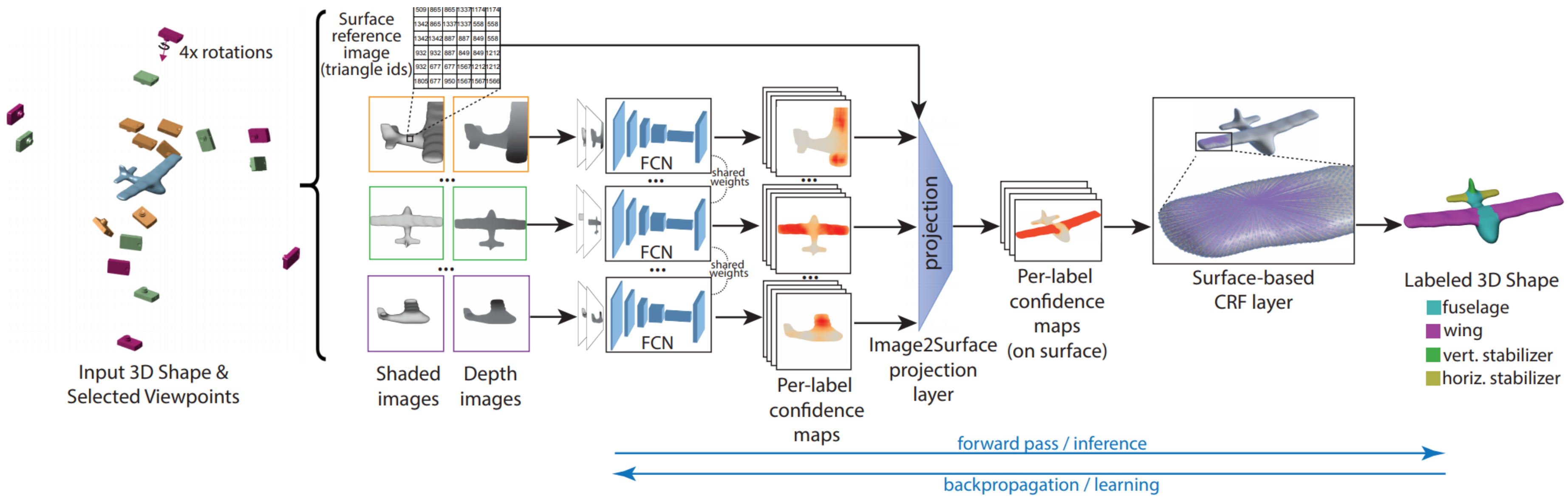
Multiple Views: Classification

- RGB images from fixed views around object:
  - view pooling for classification (only RGB; no spatial corr.)



[Multi-view Convolutional Neural Networks for 3D Shape Recognition]

Multiple Views: Segmentation



[3D Shape Segmentation with Projective Convolutional Networks]

This one is interesting in a sense that it does 3D shape segmentation (only on CAD models)

But it uses multi-view and has a spatial correlation on top of the mesh surface

Fun thing...

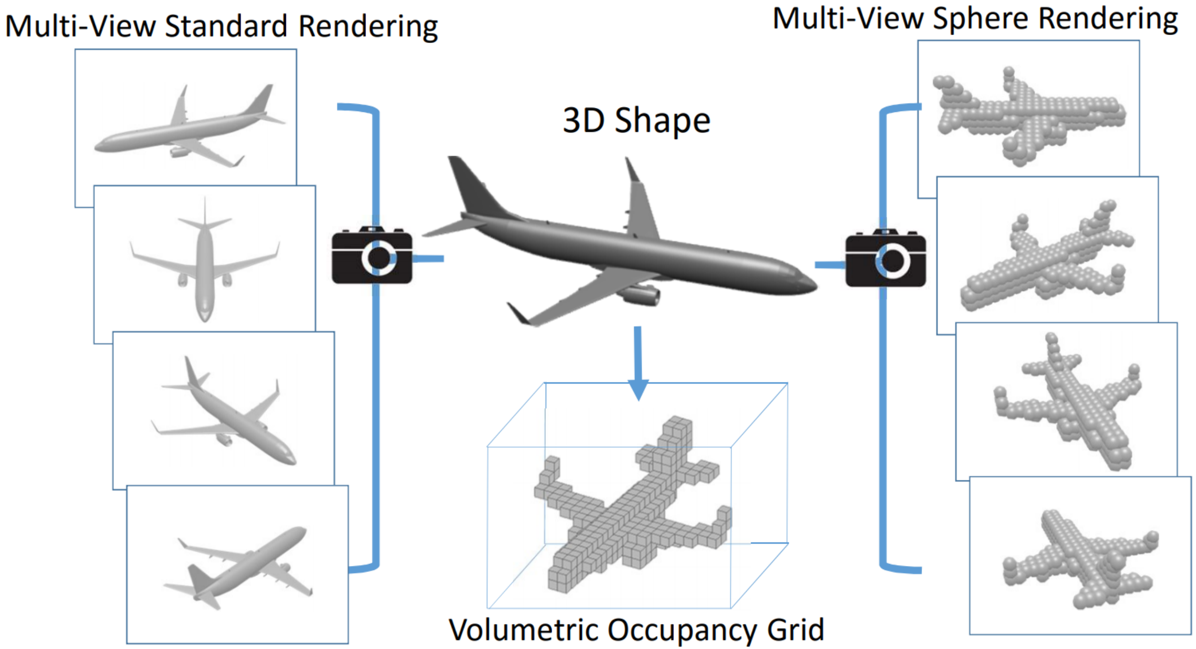


Figure 1. 3D shape representations.

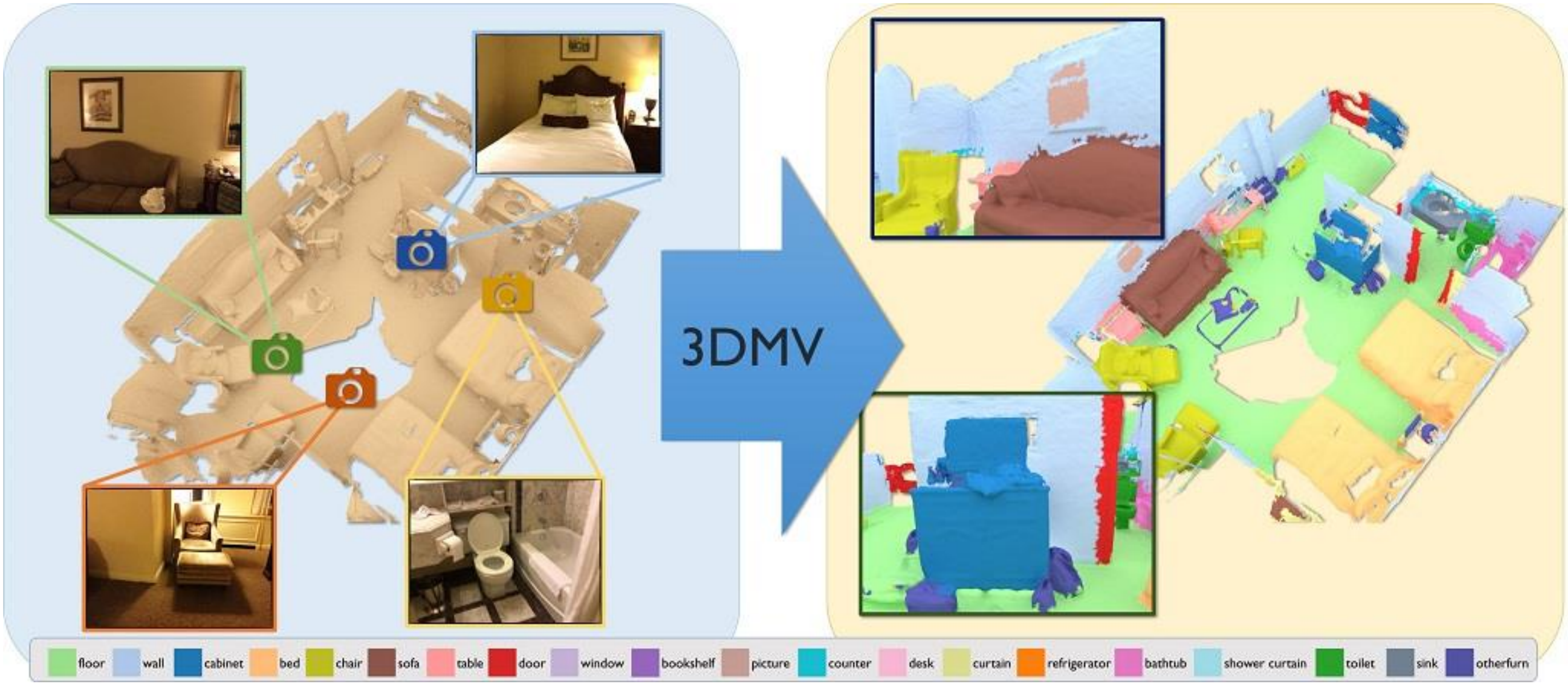
Method	#Views	Accuracy (class)	Accuracy (instance)
SPH (reported by [33])	-	68.2	-
LFD (reported by [33])	-	75.5	-
FV (reported by [32])	12	84.8	-
Su-MVCNN [32]	80	90.1	-
PyramidHoG-LFD	20	87.2	90.5
Ours-MVCNN	20	89.7	92.0
Ours-MVCNN-MultiRes	20	**91.4**	**93.8**

Table 3. Comparison of multi-view based methods. Numbers reported are classification accuracy (class average and instance average) on ModelNet40.

Volumetric and Multi-View CNNs for Object Classification on 3D Data

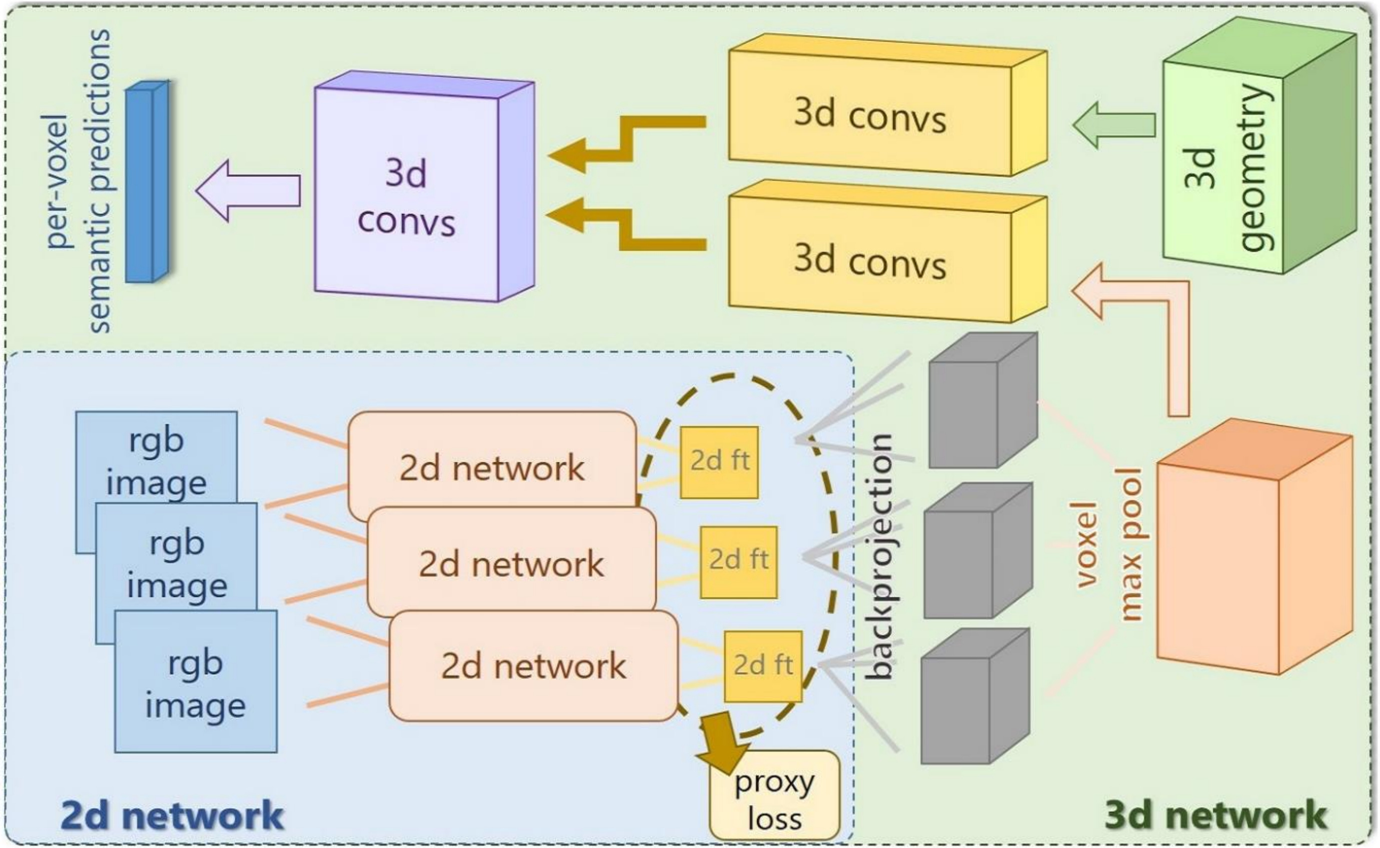
Hybrid: Volumetric + Multi-view

3D Volumetric + Multi-view



[Dai & Niessner 18] 3DMV

3D Volumetric + Multi-view



[Dai & Niessner 18] 3DMV

3D Volumetric + Multi-view

	wall	floor	cab	...	bath	other	avg
ScanNet [1]	70.1	90.3	49.8		74.3	19.5	50.8
ScanComplete [12]	87.2	96.9	44.5		85.1	26.9	52.8
PointNet++ [24]	**89.5**	**97.8**	39.8		86.1	30.7	60.2
3DMV (ours)	73.9	95.6	**69.9**		**94.7**	**58.5**	**75.0**

[Dai & Niessner 18] 3DMV

3D Volumetric + Multi-view

	wall	floor	cab		bath	other	avg
2d only (1 view)	37.1	39.1	26.7		36.3	20.4	27.1
2d only (3 views)	58.6	62.5	40.8		61.5	34.3	44.2
Ours (no geo input)	76.2	92.9	59.3		80.8	9.3	58.2
Ours (3d geo only)	60.4	95.0	54.4		87.0	20.6	54.4
Ours (3d geo+voxel color)	58.8	94.7	55.5		85.4	20.5	55.9
Ours (1 view, fixed 2d)	77.3	96.8	**70.0**	…	87.0	58.5	69.1
Ours (1 view)	70.7	96.8	61.4		81.6	51.7	70.1
Ours (3 view, fixed 2d)	**81.1**	96.4	58.0		92.5	**60.7**	72.8
Ours (3 view)	75.2	**97.1**	66.4		89.9	57.2	73.0
Ours (5 view, fixed 2d)	77.3	95.7	68.9		93.5	59.6	74.5
Ours (5 view)	73.9	95.6	69.9		**94.7**	58.5	**75.0**

[Dai & Niessner 18] 3DMV

Conclusion so far

- Hybrid:
  - Nice way to combine color and geometry
  - Great performance (best so far for segmentation)
  - End-to-end helps less than we hoped for
  - Could be faster…

Next Lectures

- Next Lecture -> Jan 28th
  - Domain Adaptation and Transfer Learning
  - Possible graphs if time permits

- Keep working on the projects!