THE UNIVERSITY OF BRITISH COLUMBIA

Topics in AI (CPSC 532S):
Multimodal Learning with Vision, Language and Sound

Lecture 15: Generative Models

Supervised vs. **Unsupervised** Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x → y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, *etc.*

→ Cat

Classification

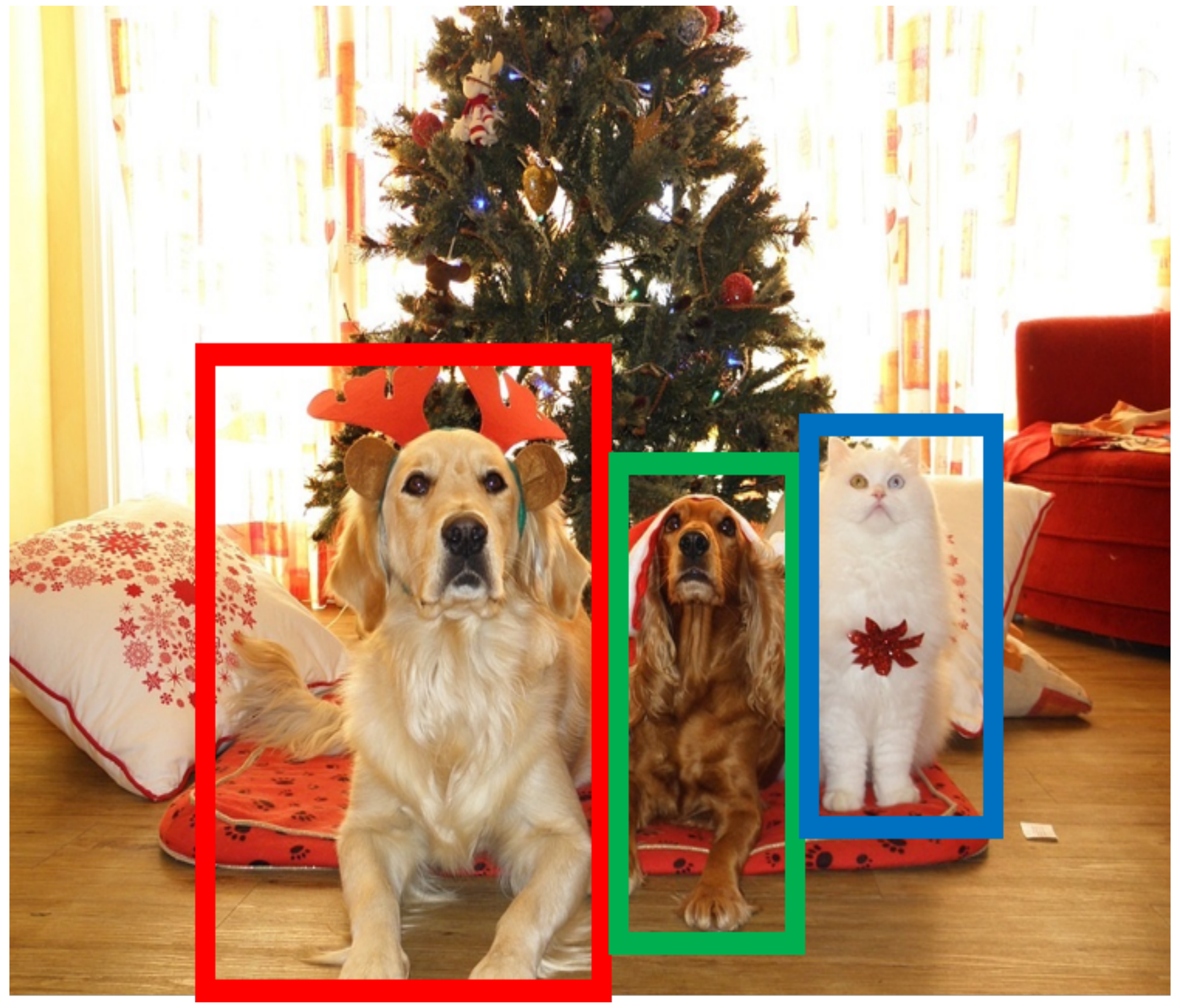
Supervised vs. **Unsupervised** Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x→y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, *etc.*



DOG, DOG, CAT

Object Detection

This image is CC0 public domain

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

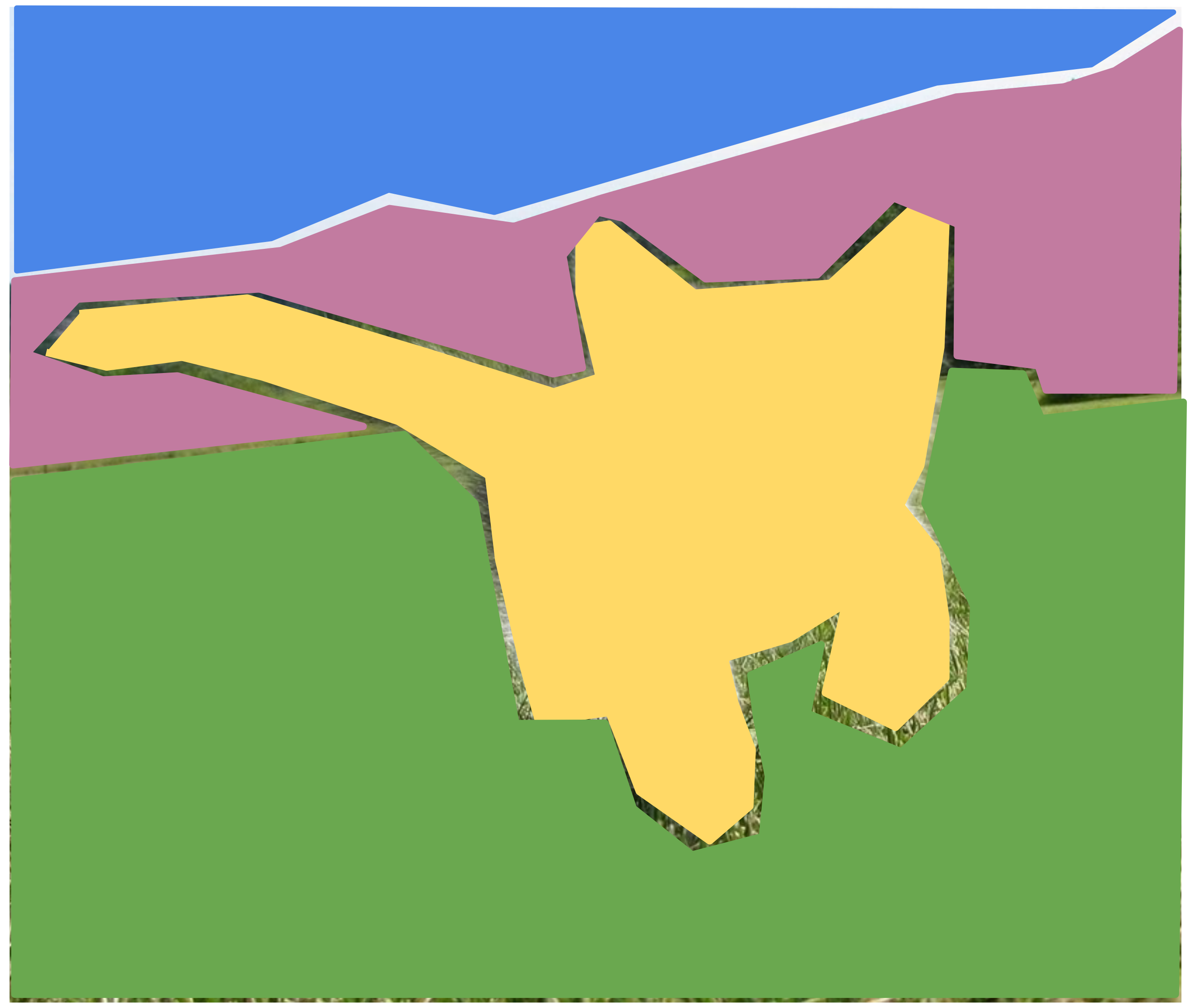
Supervised vs. **Unsupervised** Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x→y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, *etc.*



GRASS, CAT, TREE, SKY

Semantic Segmentation

Supervised vs. **Unsupervised** Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x→y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, *etc.*

A cat sitting on a suitcase on the floor

Image Captioning

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Supervised vs. **Unsupervised** Learning

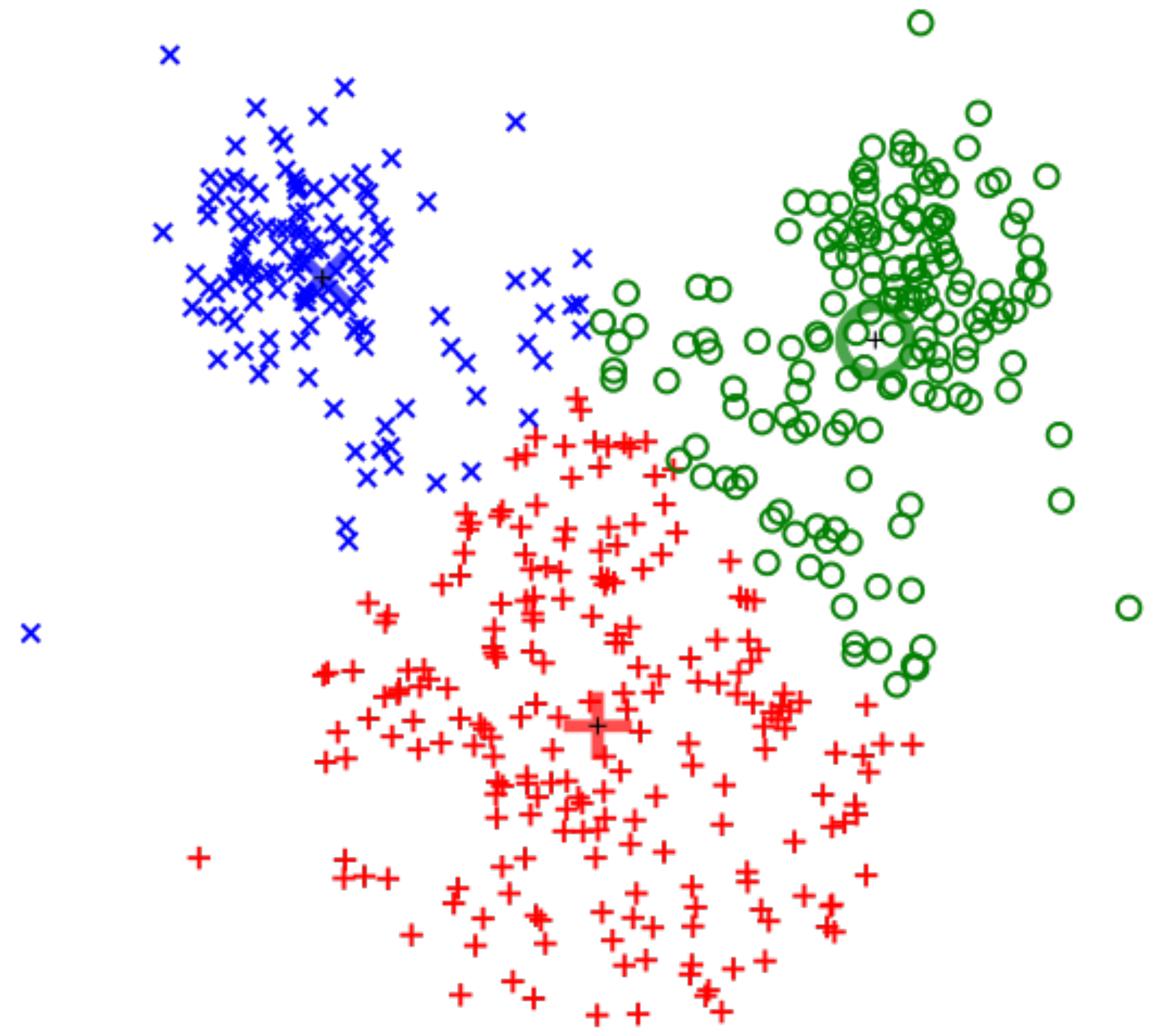
Unsupervised Learning

Data: x

Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, *etc.*



k-means clustering

This image is CC0 public domain

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Supervised vs. **Unsupervised** Learning

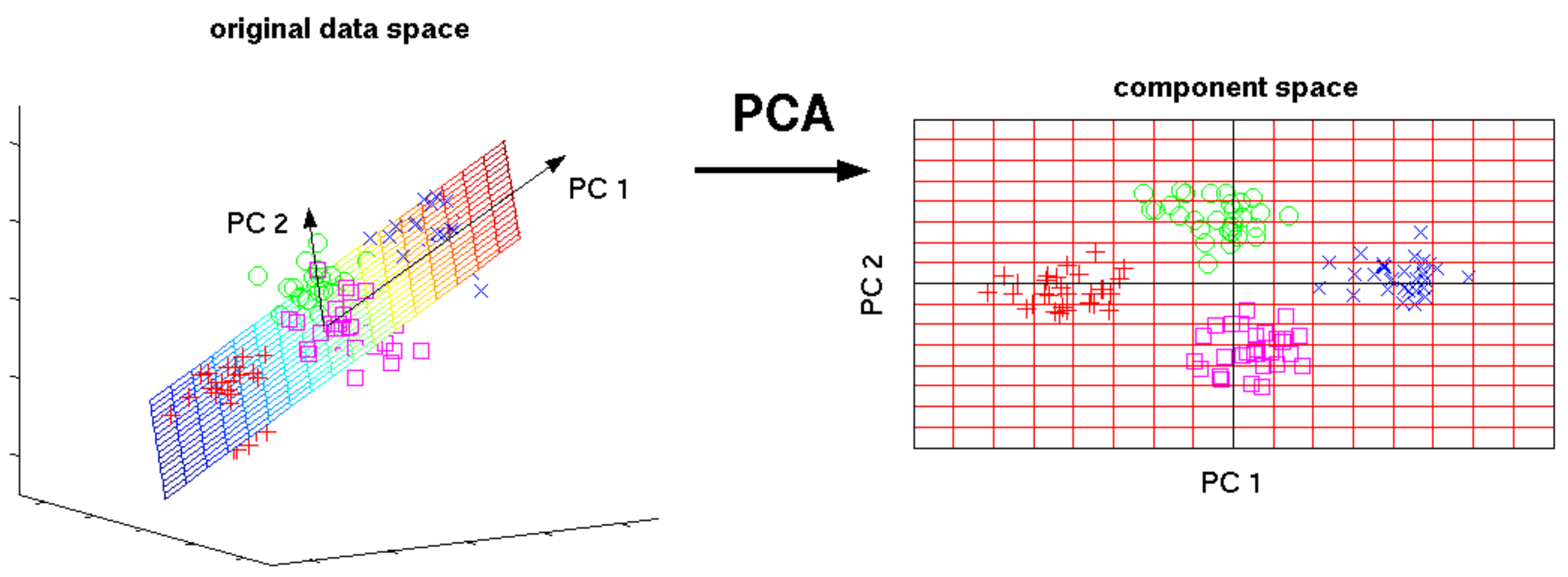
Unsupervised Learning

Data: x

Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, *etc.*



dimensionality reduction

Supervised vs. **Unsupervised** Learning

Unsupervised Learning

Data: x

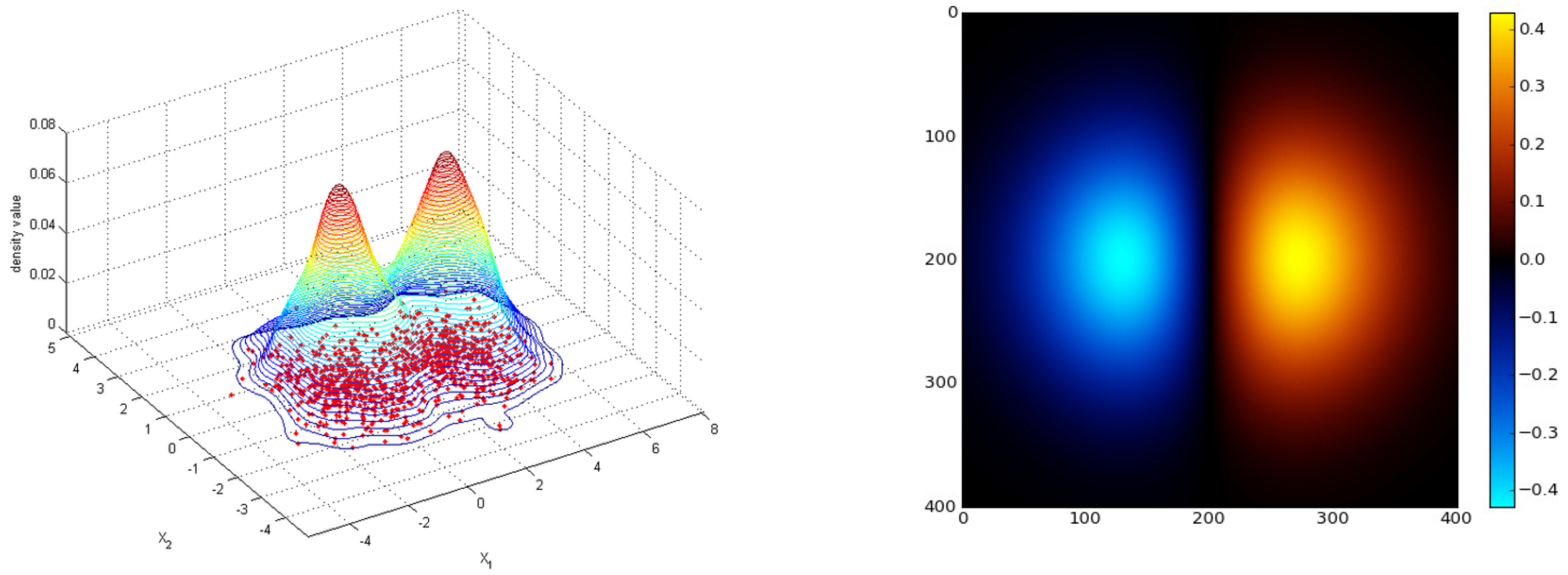
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, *etc.*

Figure copyright Ian Goodfellow, 2016. Reproduced with permission.

1-dim density estimation



2-dim density estimation

2-d density images left and right are CC0 public domain

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Supervised vs. **Unsupervised** Learning

Supervised Learning	**Unsupervised** Learning
Data: (x, y) x is data, y is label **Goal:** Learn a *function* to map x→y **Examples:** Classification, regression, object detection, semantic segmentation, image captioning, *etc.*	**Data:** x Just data, no labels! **Goal:** Learn some underlying hidden *structure* of the data **Examples:** Clustering, dimensionality reduction, feature learning, density estimation, *etc.*

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Generative Models

Given training data, generate new samples from the same distribution

Training data ~ $p_{\text{data}}(\boldsymbol{x})$

Generated samples ~ $p_{\text{model}}(\boldsymbol{x})$

Want to learn $p_{\text{model}}(\boldsymbol{x})$ similar to $p_{\text{data}}(\boldsymbol{x})$

Generative Models

Given training data, generate new samples from the same distribution

Training data ~ $p_{\text{data}}(\boldsymbol{x})$

Generated samples ~ $p_{\text{model}}(\boldsymbol{x})$

Want to learn $p_{\text{model}}(\boldsymbol{x})$ similar to $p_{\text{data}}(\boldsymbol{x})$

Addresses **density estimation**, a core problem in unsupervised learning

— **Explicit** density estimation: explicitly define and solve for $p_{\text{model}}(\boldsymbol{x})$

— **Implicit** density estimation: learn model that can sample from $p_{\text{model}}(\boldsymbol{x})$ w/o explicitly defining it

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Taxonomy of Generative Models

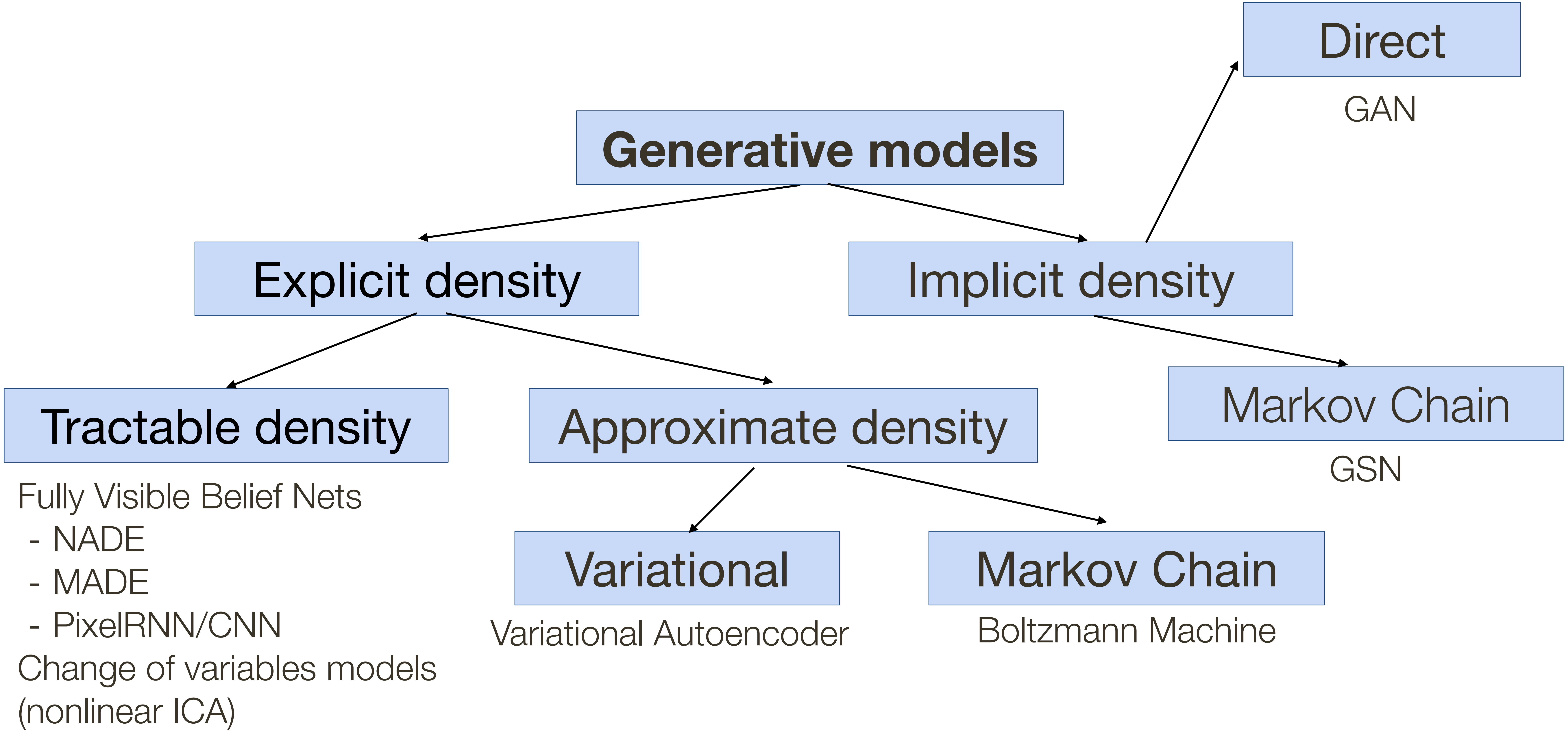


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Taxonomy of Generative Models

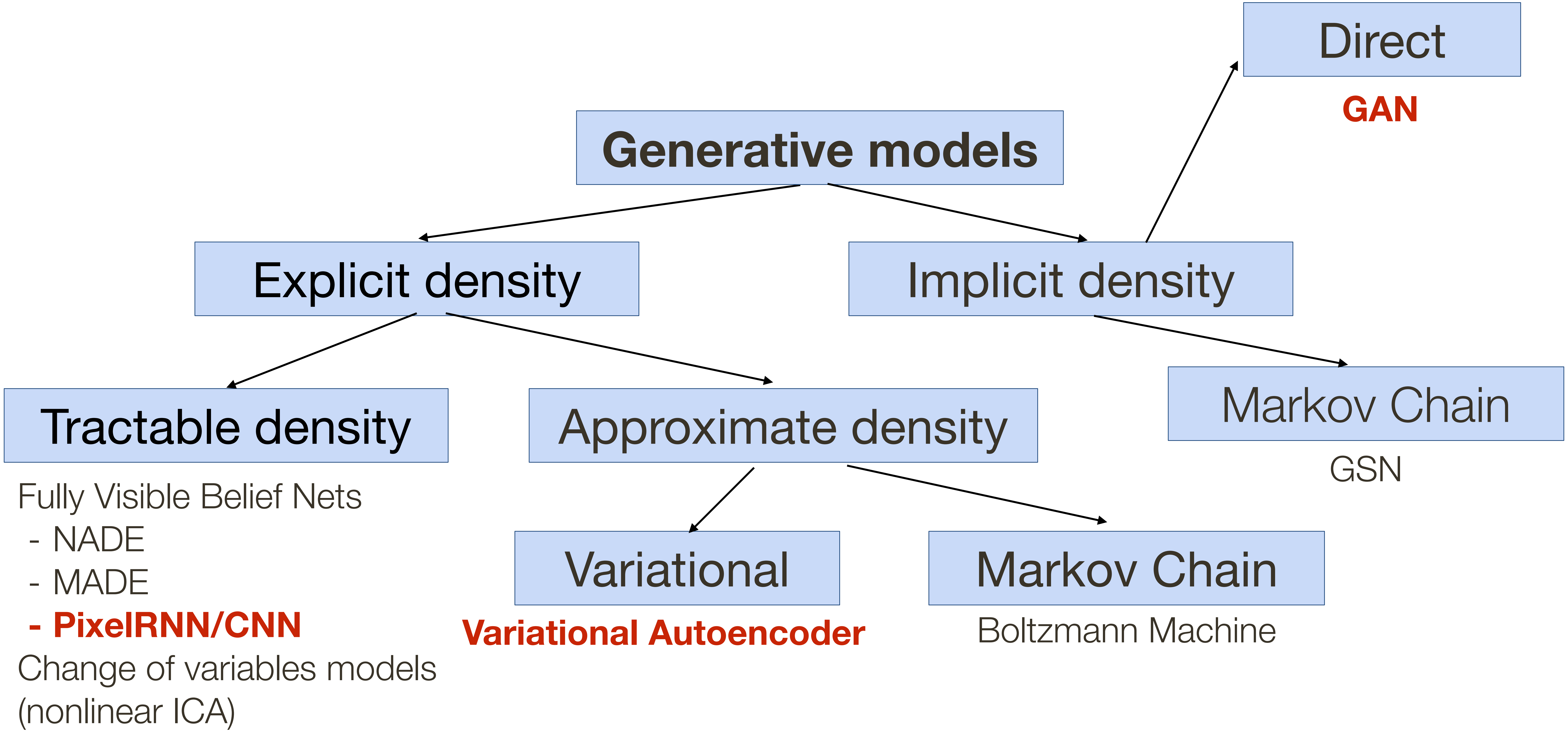


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Why **Generative** Models?

- Realistic **samples** for artwork, super-resolution, colorization, *etc.*

Why **Generative** Models?

— Realistic **samples** for artwork, super-resolution, colorization, *etc.*



— Generative models of time-series data can be used for **simulation**, **predictions** and planning (reinforcement learning applications)

— Training generative models can also enable inference of latent representation that can be useful as **general features**

— **Dreaming** / hypothesis visualization

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

PixelRNN and PixelCNN

PixelRNN

[van der Oord et al., 2016]

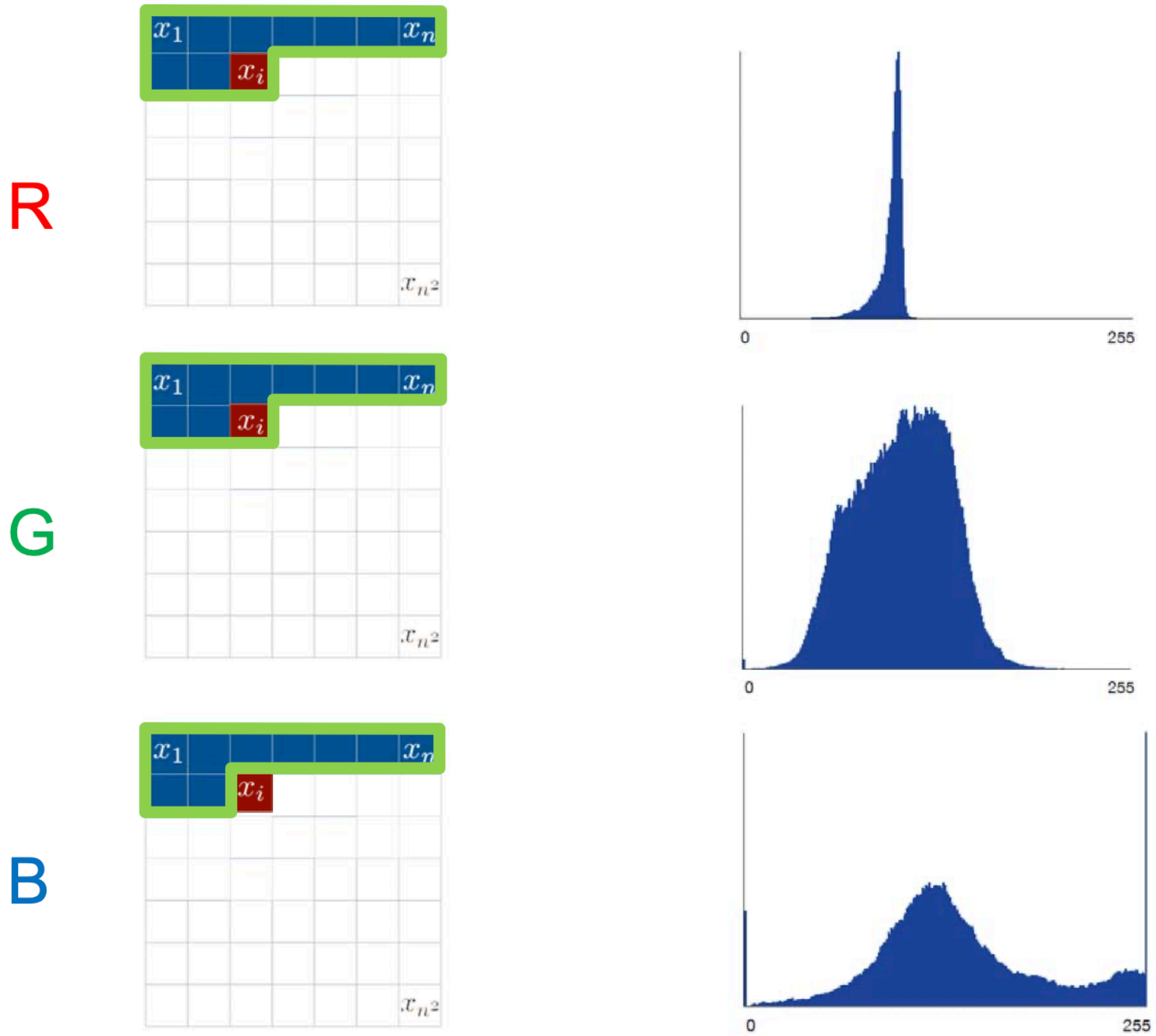
Explicit Density model

Use chain rule to decompose likelihood of an image $\boldsymbol{x}$ into product of (many) 1-d distributions

$$\underbrace{p(x)}_{\text{Likelihood of image } \boldsymbol{x}} = \prod_{i=1}^{n} \underbrace{p(x_i | x_1, ..., x_{i-1})}_{\text{Probability of i'th pixel value given all previous pixels}}$$

then maximize likelihood of training data

PixelRNN



PixelRNN

[van der Oord et al., 2016]

Explicit Density model

Use chain rule to decompose likelihood of an image $\boldsymbol{x}$ into product of (many) 1-d distributions

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

Likelihood of image $\boldsymbol{x}$

Probability of i'th pixel value given all previous pixels

Complex distribution over pixel values, so lets model using **neural network**

then maximize likelihood of training data

PixelRNN

[van der Oord et al., 2016]

Explicit Density model

Use chain rule to decompose likelihood of an image $\boldsymbol{x}$ into product of (many) 1-d distributions

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

Likelihood of image $\boldsymbol{x}$

Probability of i'th pixel value given all previous pixels

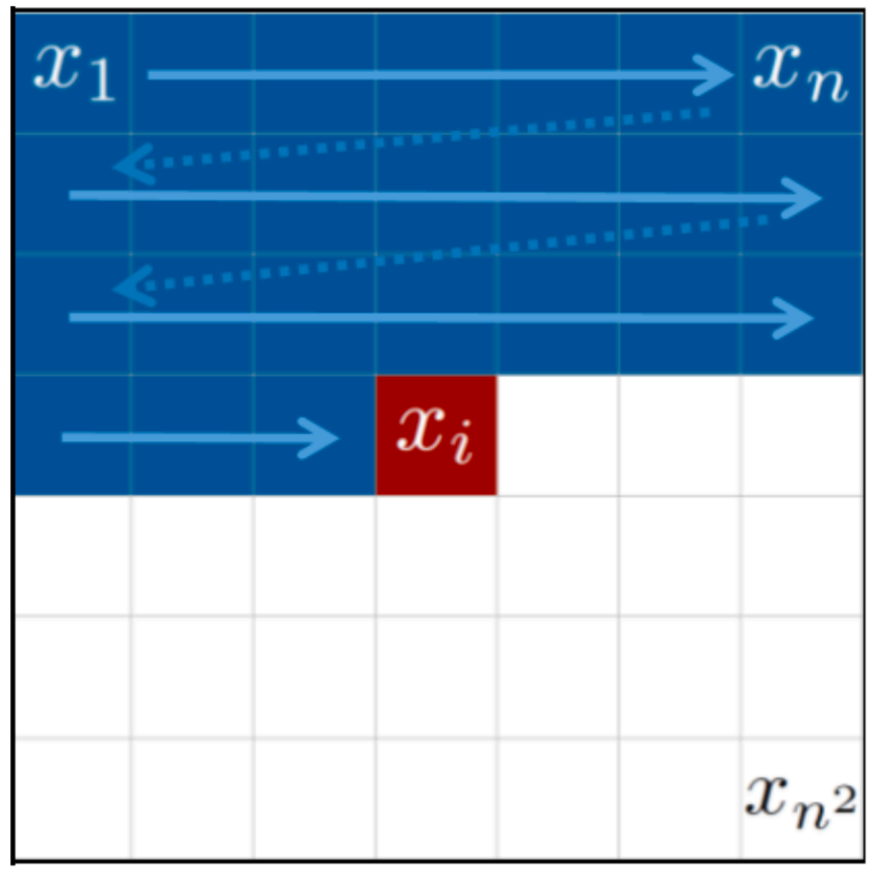
then maximize likelihood of training data

Complex distribution over pixel values, so lets model using **neural network**

Also requires defining **ordering** of "previous pixels"

PixelRNN

[van der Oord et al., 2016]

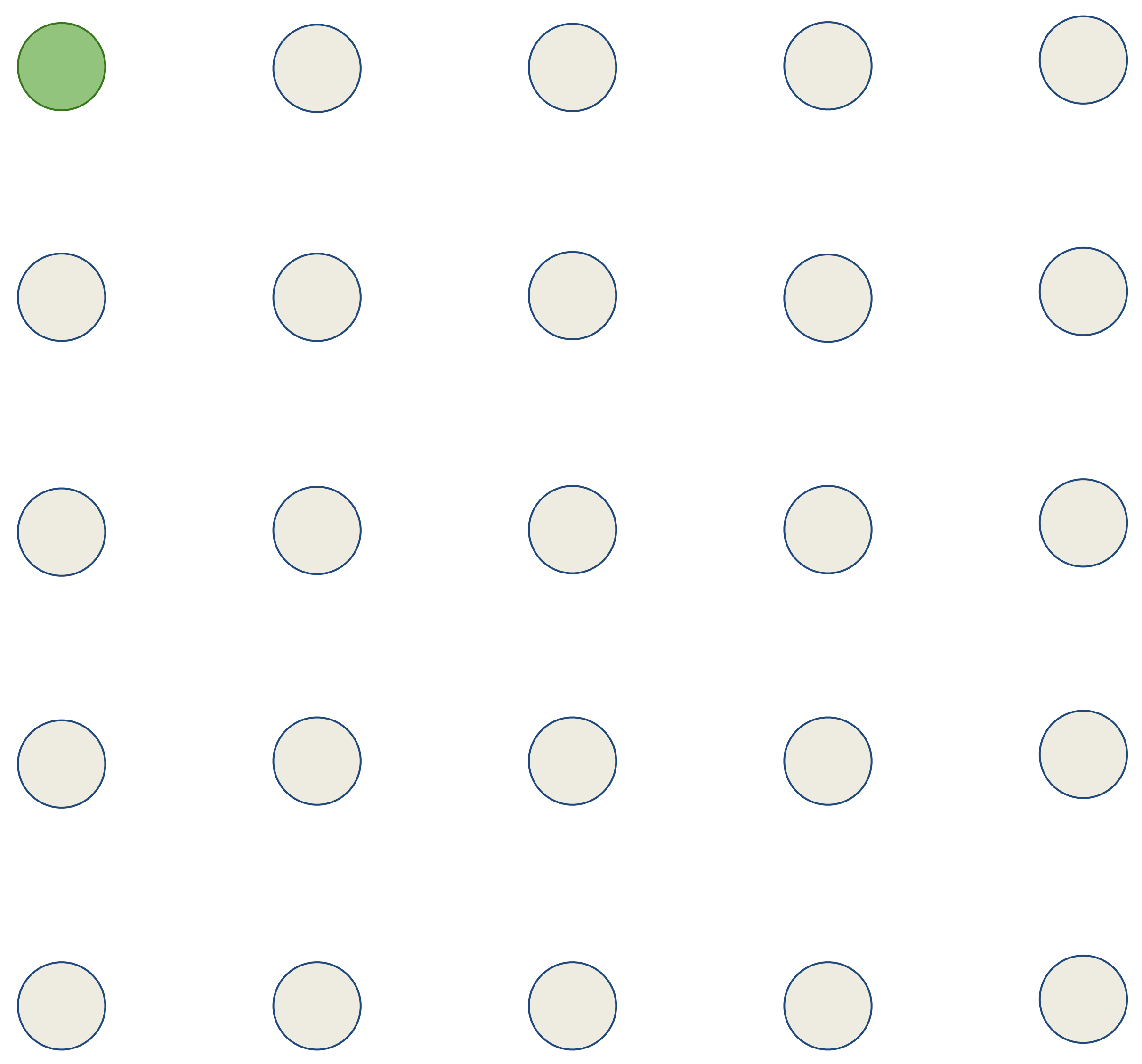


PixelRNN

[van der Oord et al., 2016]

Generate image pixels starting from the corner

Dependency on previous pixels model using an RNN (LSTM)

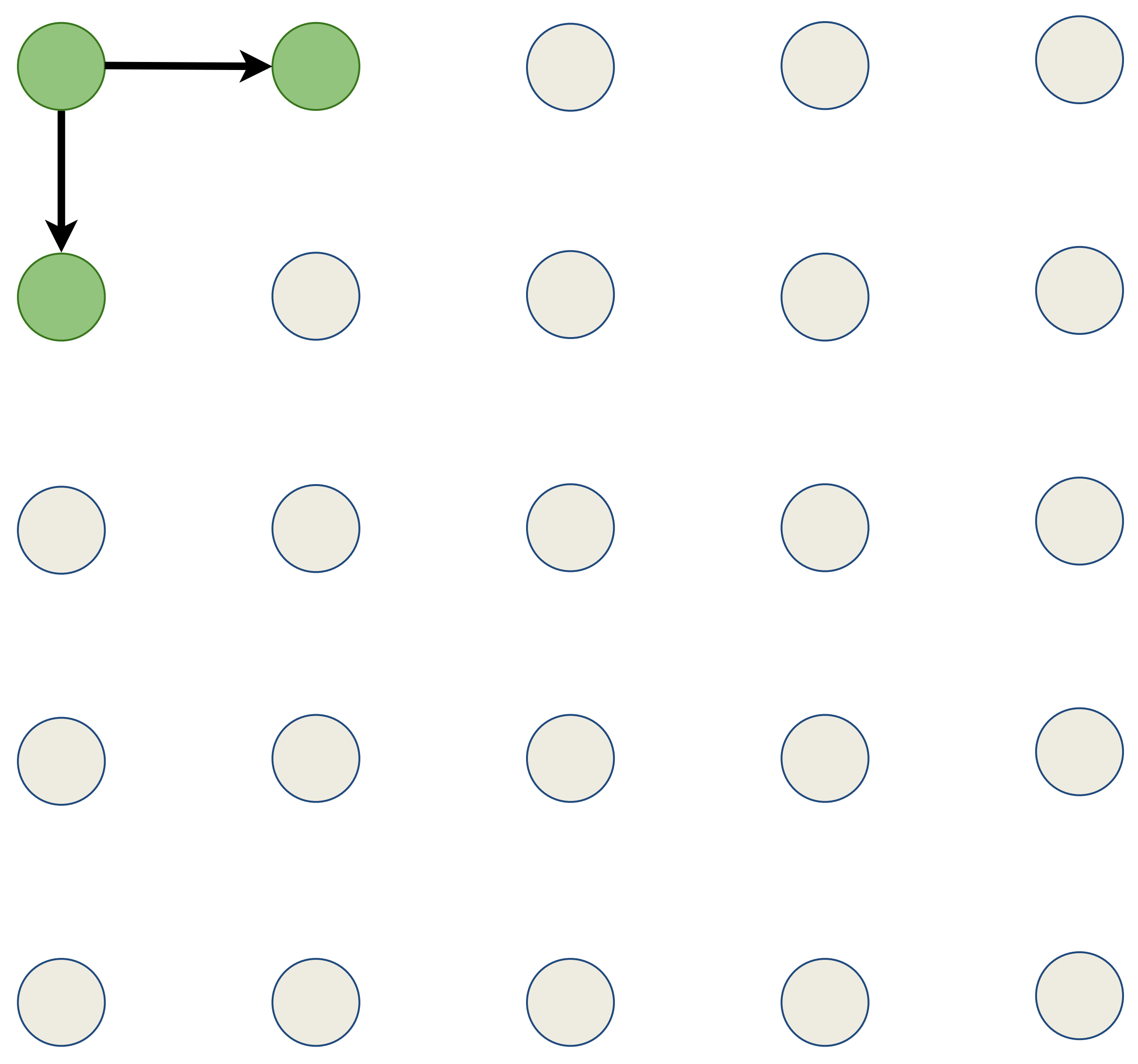


PixelRNN

[van der Oord et al., 2016]

Generate image pixels starting from the corner

Dependency on previous pixels model using an RNN (LSTM)

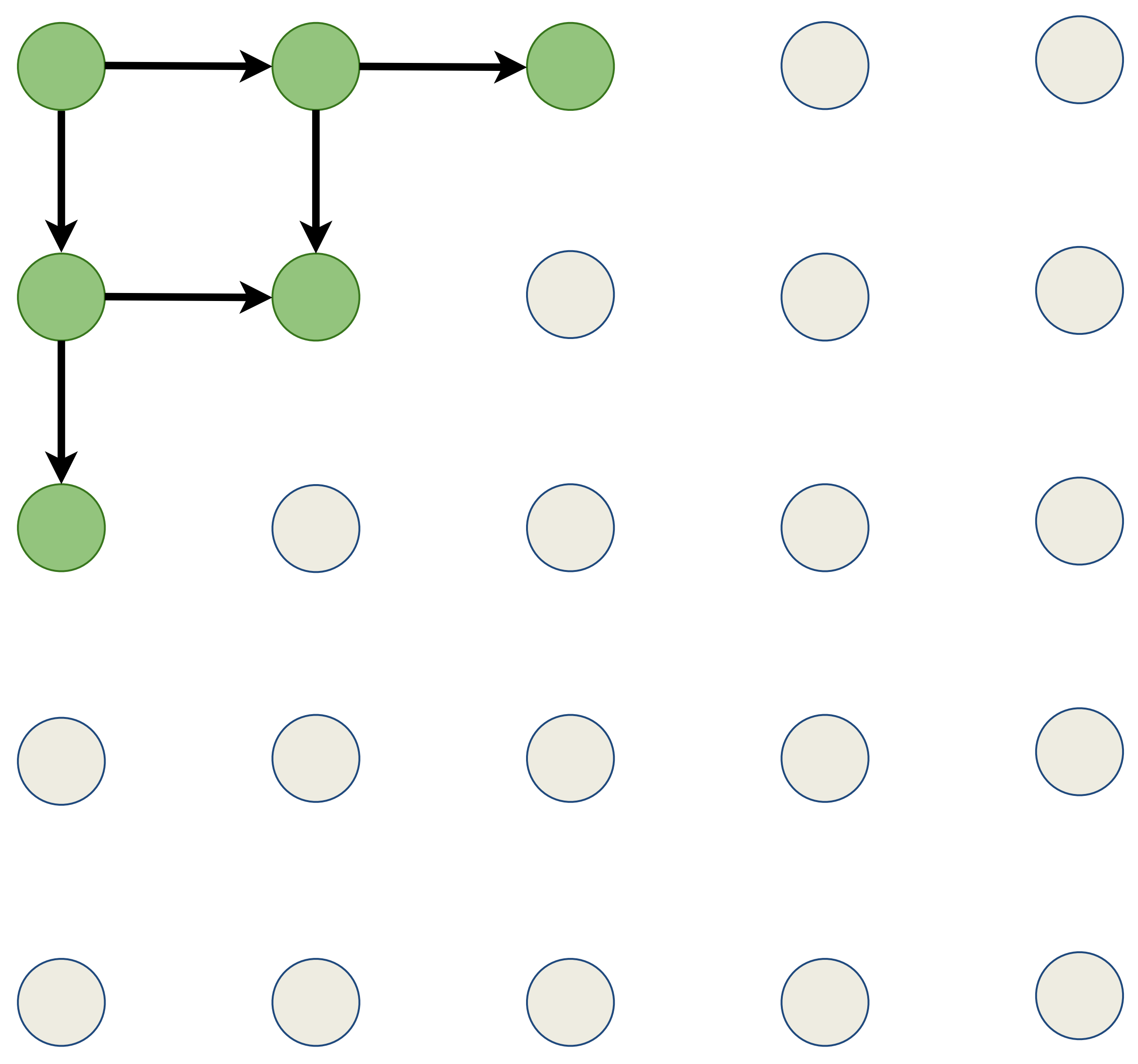


PixelRNN

[van der Oord et al., 2016]

Generate image pixels starting from the corner

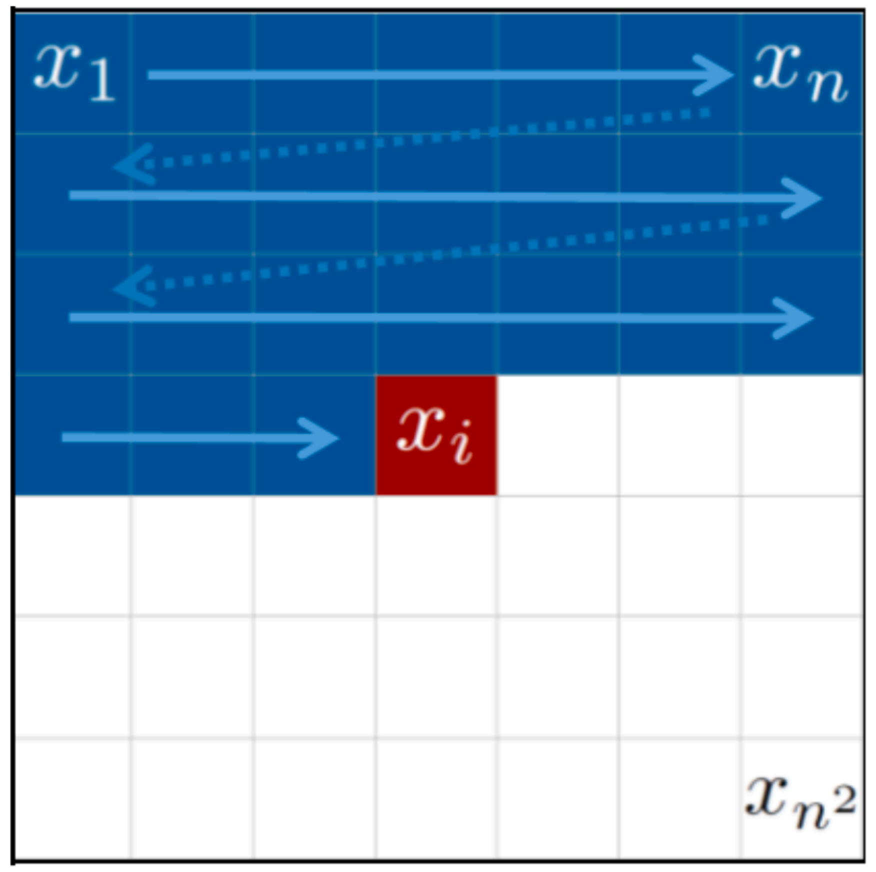
Dependency on previous pixels model using an RNN (LSTM)

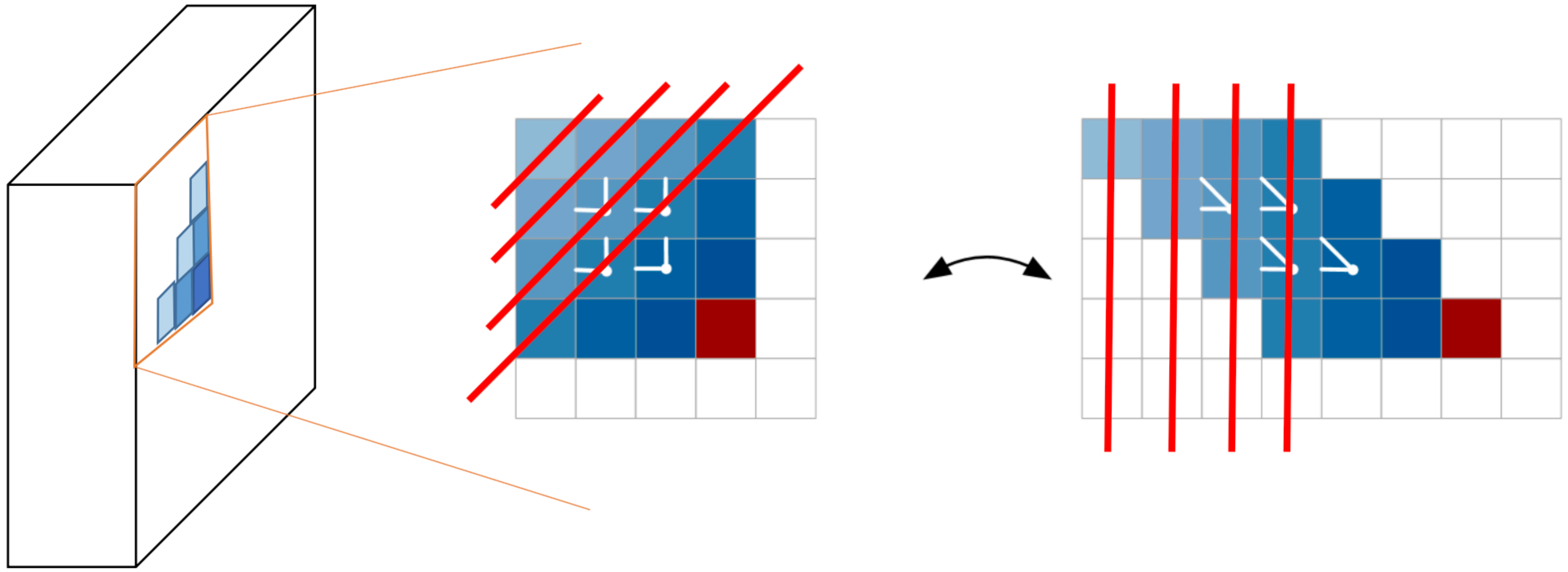


* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

PixelRNN

[van der Oord et al., 2016]





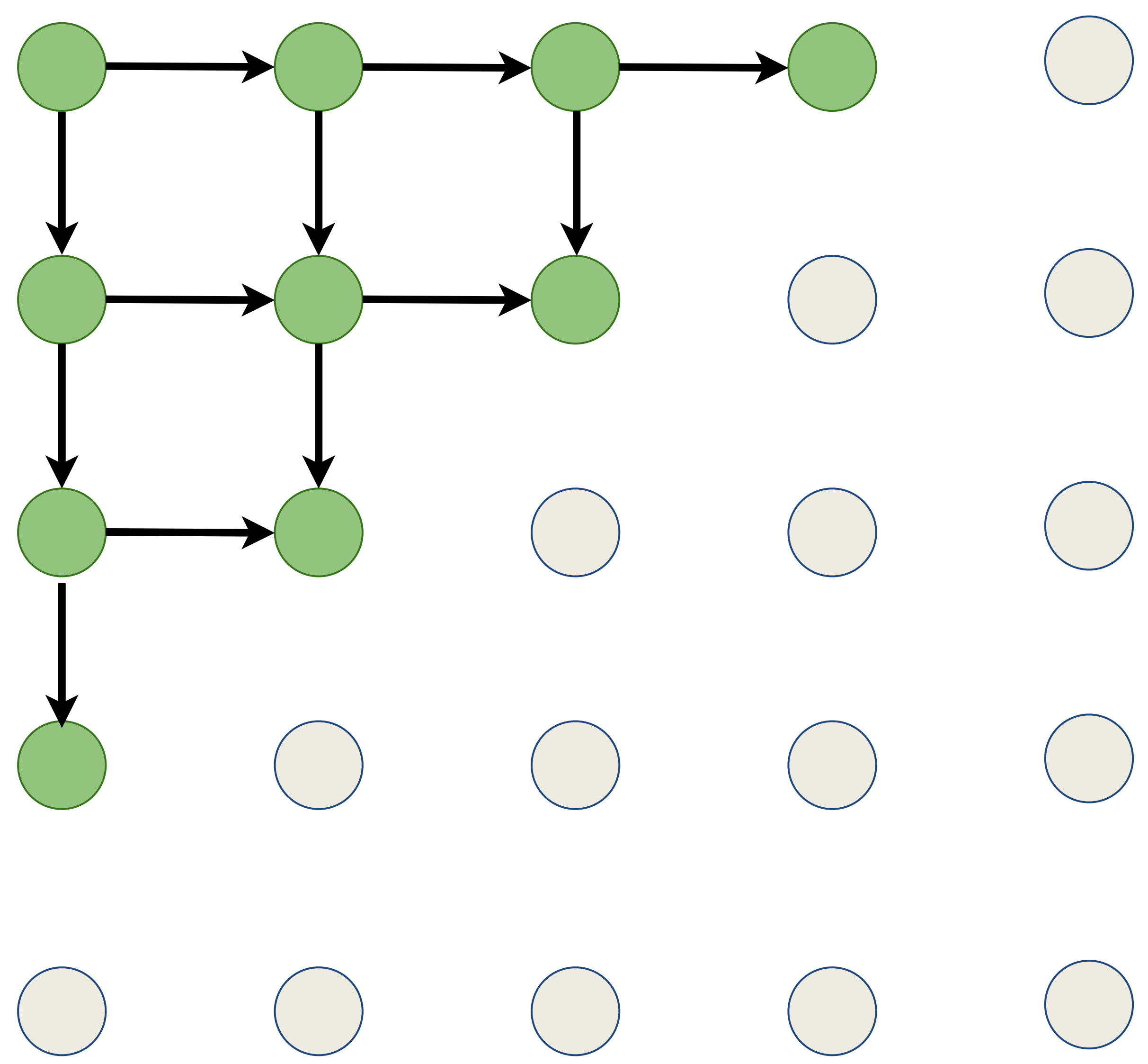
* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

PixelRNN

[van der Oord et al., 2016]

Generate image pixels starting from the corner

Dependency on previous pixels model using an RNN (LSTM)

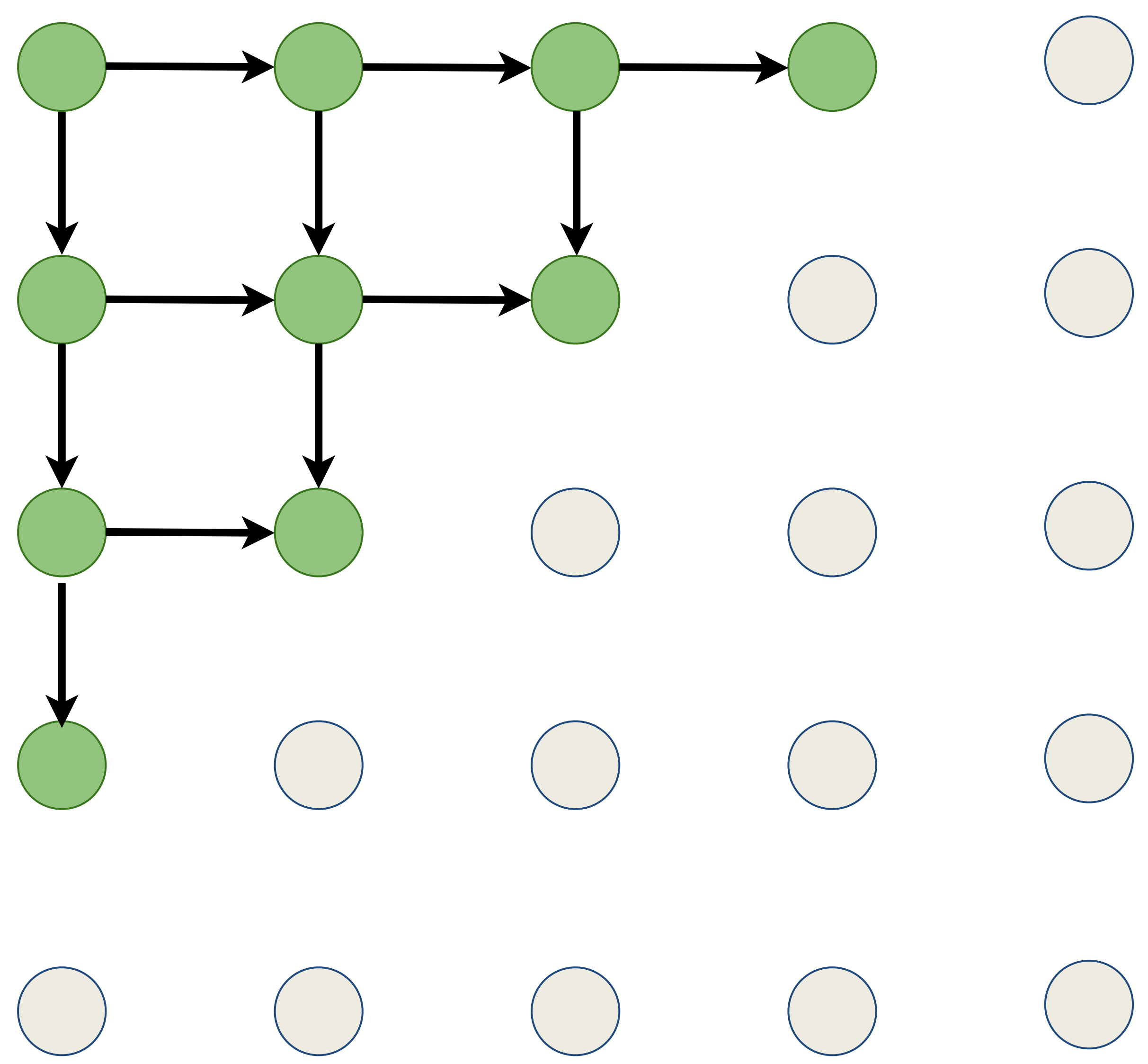


PixelRNN

[van der Oord et al., 2016]

Generate image pixels starting from the corner

Dependency on previous pixels model using an RNN (LSTM)



Problem: sequential generation is slow

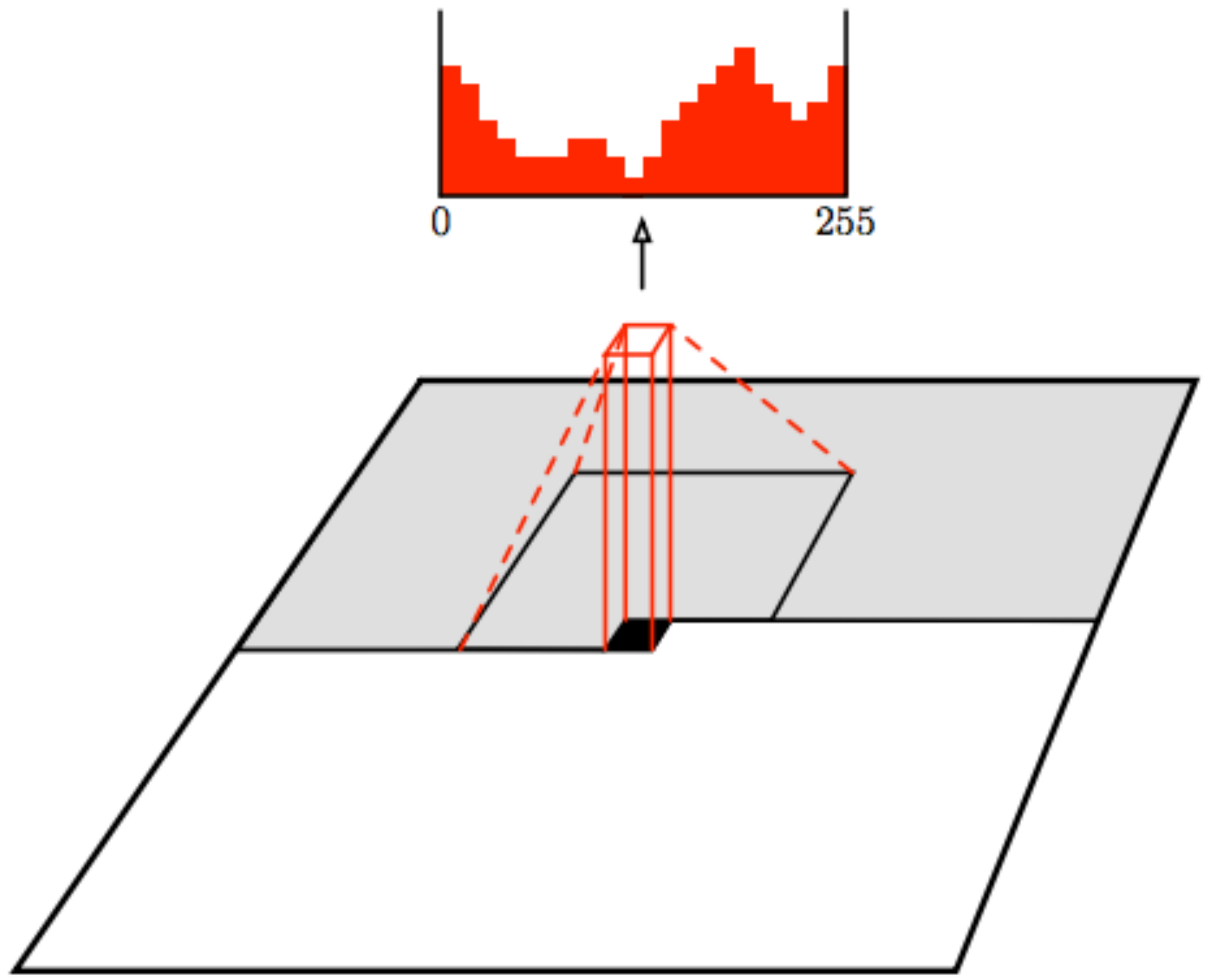
* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

PixelCNN

[van der Oord et al., 2016]

Still generate image pixels starting from the corner

Dependency on previous pixels now modeled using a CNN over context region



PixelCNN

[van der Oord et al., 2016]

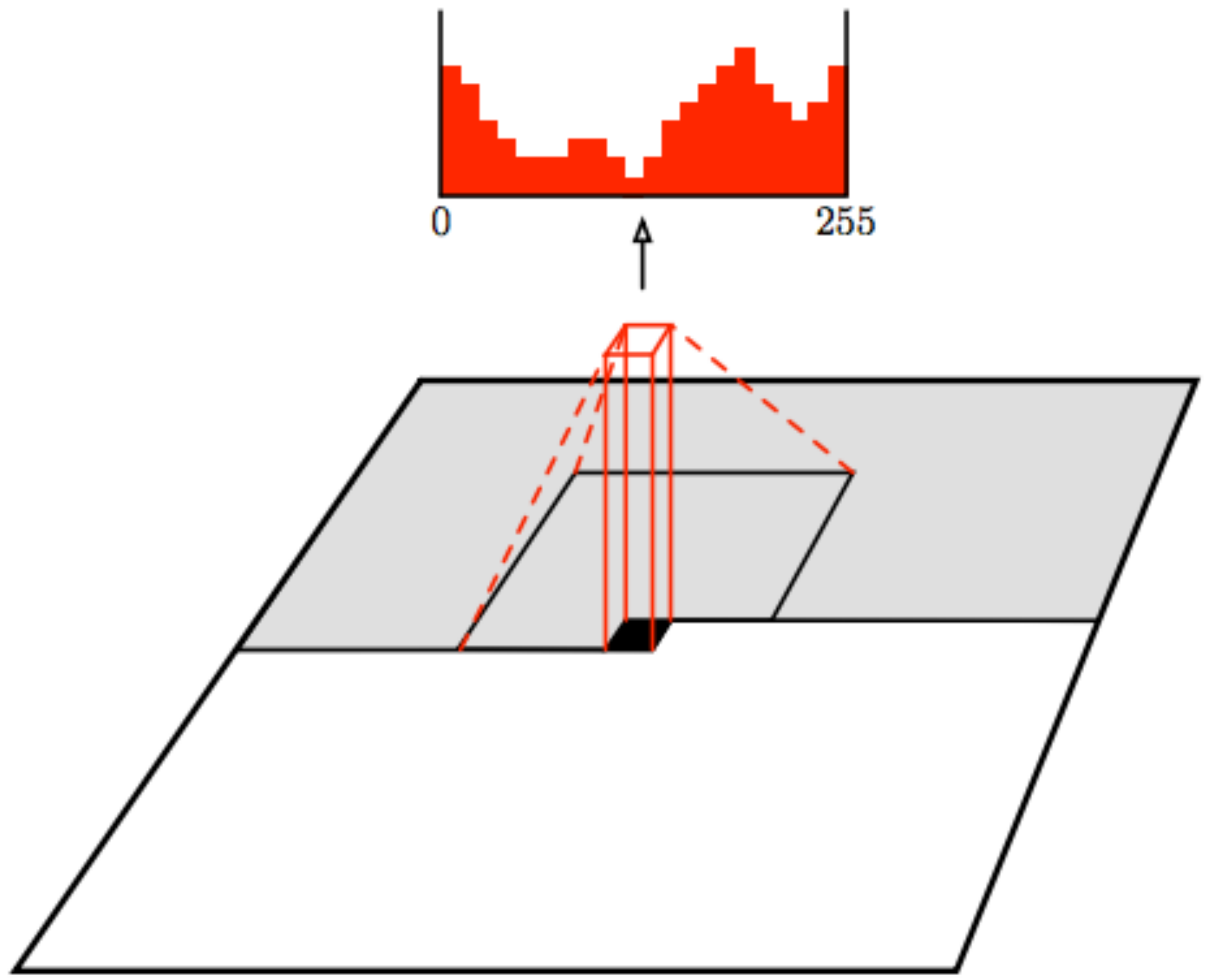
Still generate image pixels starting from the corner

Dependency on previous pixels now modeled using a CNN over context region

Training: maximize likelihood of training images

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

Softmax loss at each pixel



PixelCNN

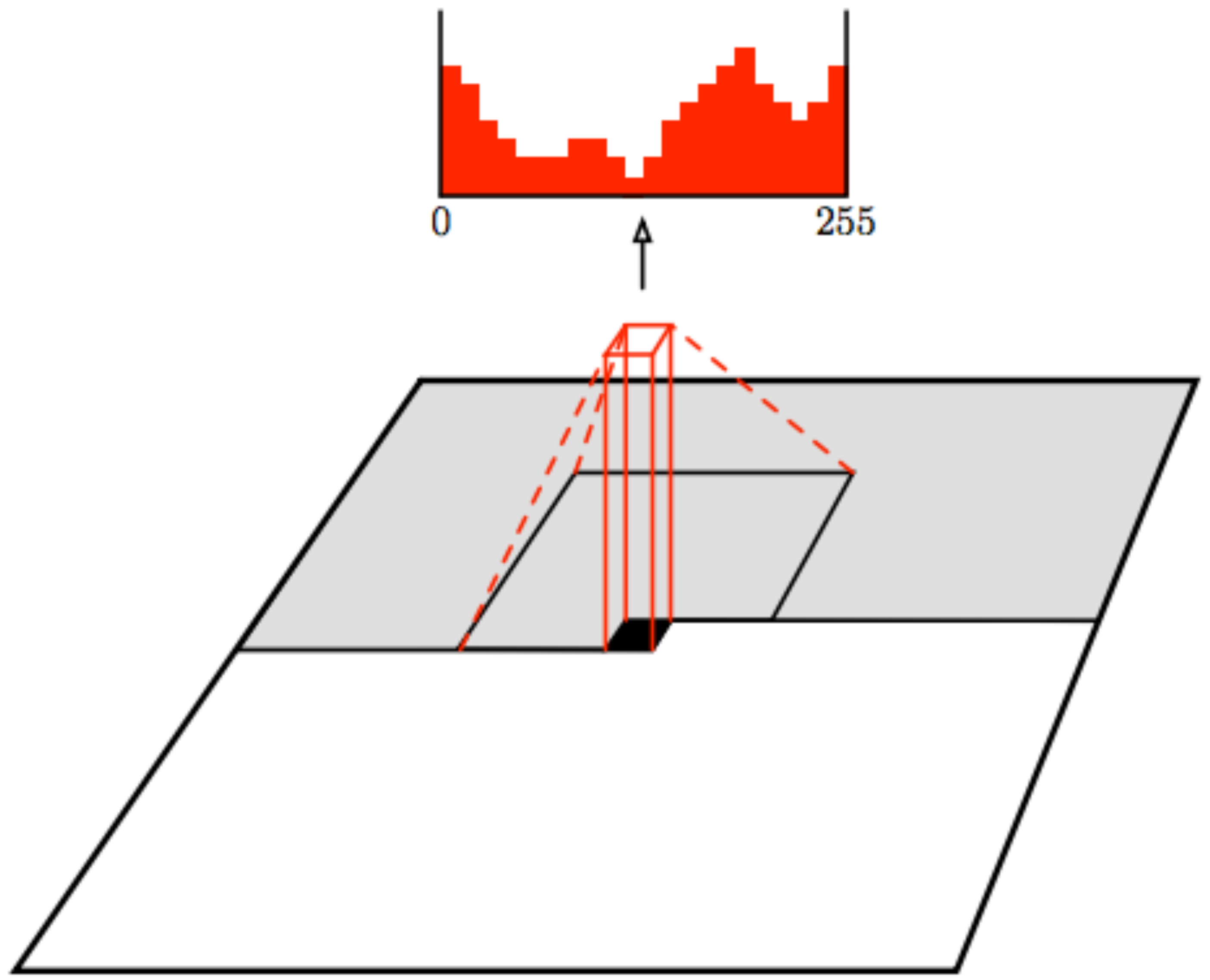
[van der Oord et al., 2016]

Still generate image pixels starting from the corner

Dependency on previous pixels now modeled using a CNN over context region

Training: maximize likelihood of training images

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

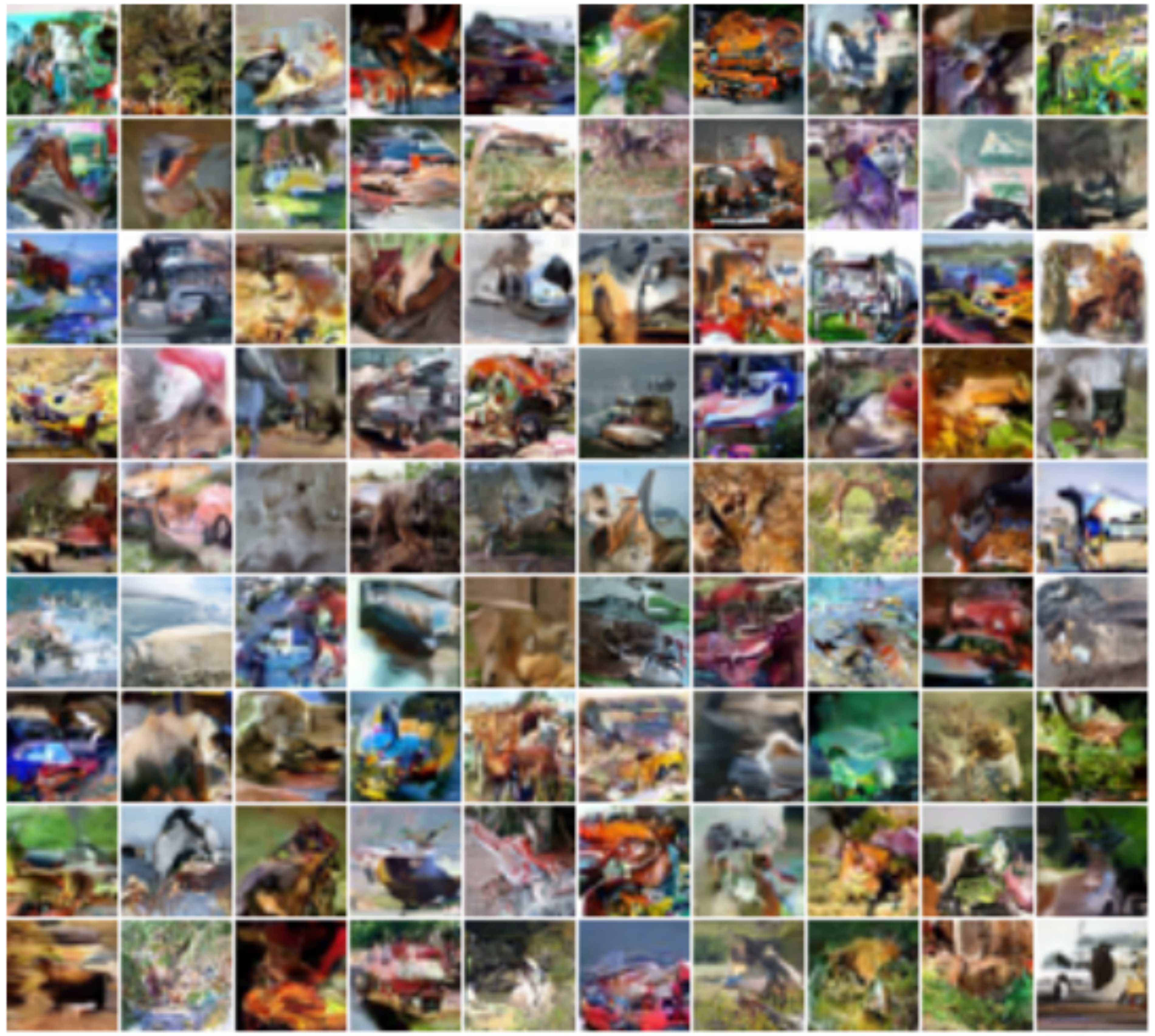


Generation is still slow (sequential), but learning is faster

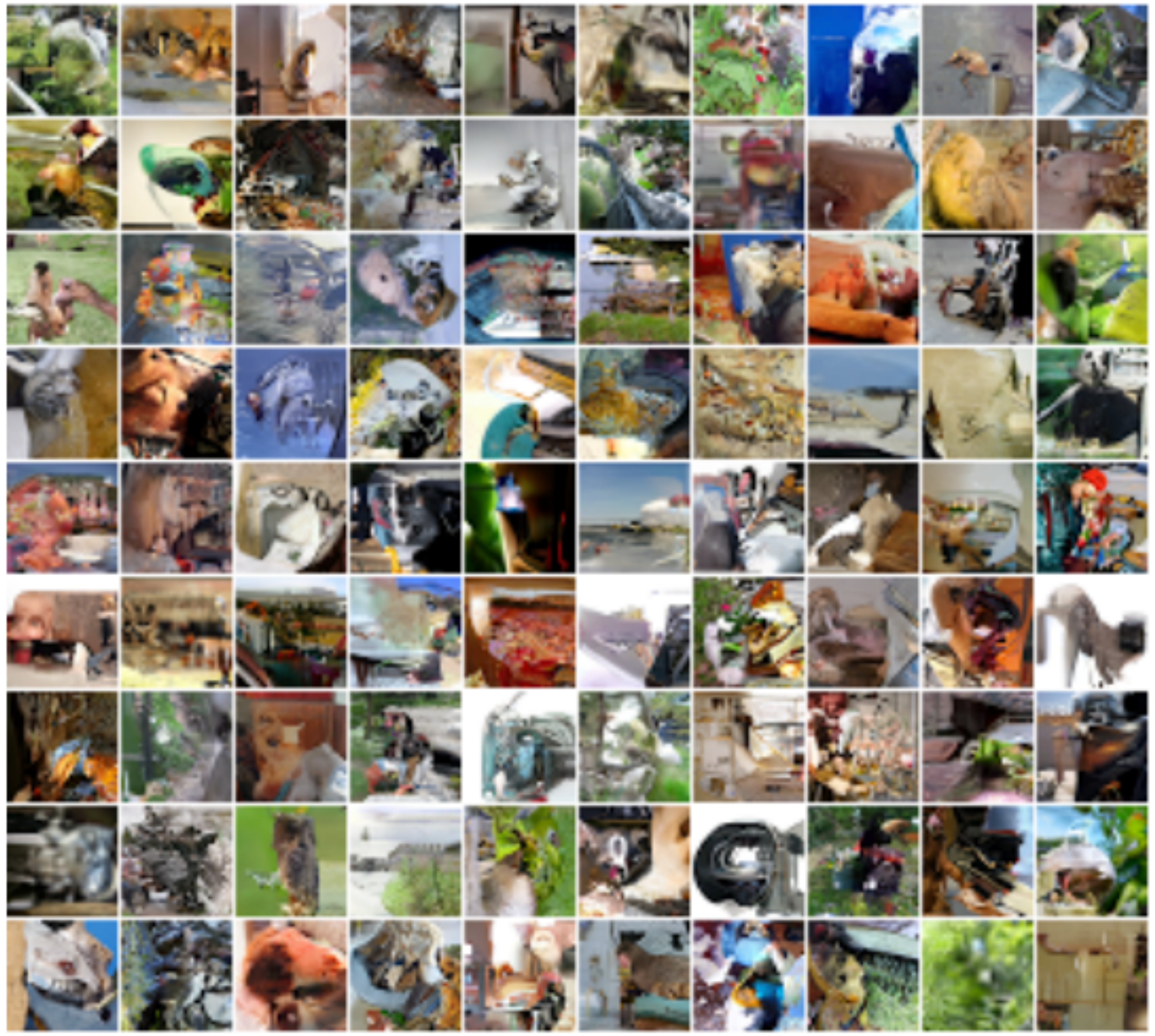
* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Generated Samples

[van der Oord et al., 2016]



32x32 **CIFAR-10**



32x32 **ImageNet**

PixelRNN and PixelCNN

Pros:

— Can explicitly compute likelihood p(x)

— Explicit likelihood of training data gives good evaluation metric

— Good samples

Con:

— Sequential generation => slow

Improving PixelCNN performance

— Gated convolutional layers

— Short-cut connections

— Discretized logistic loss

— Multi-scale

— Training tricks

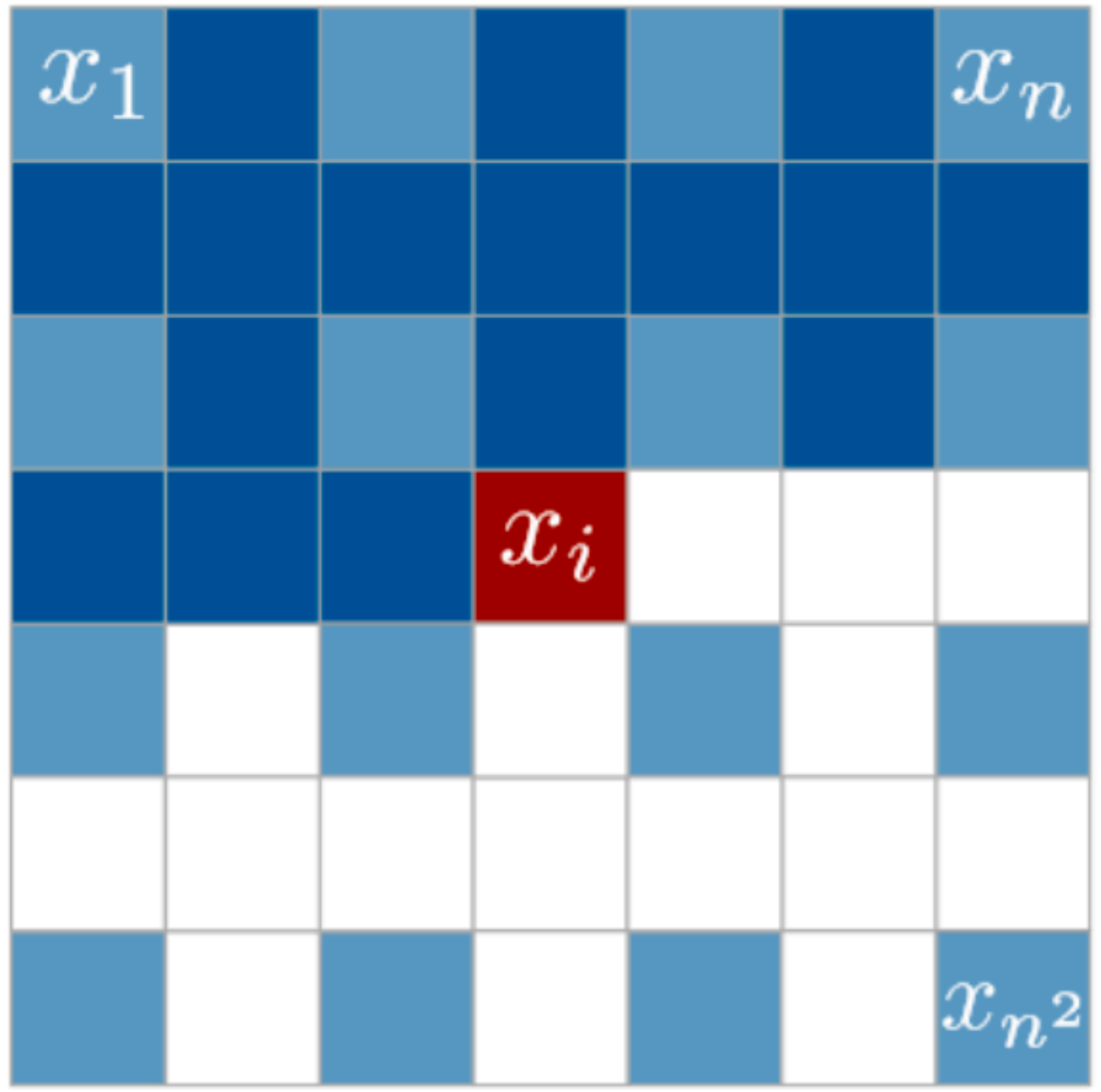
— Etc…

Multi-scale PixelRNN

[van der Oord et al., 2016]

Take sub-sampled pixels as additional input pixels

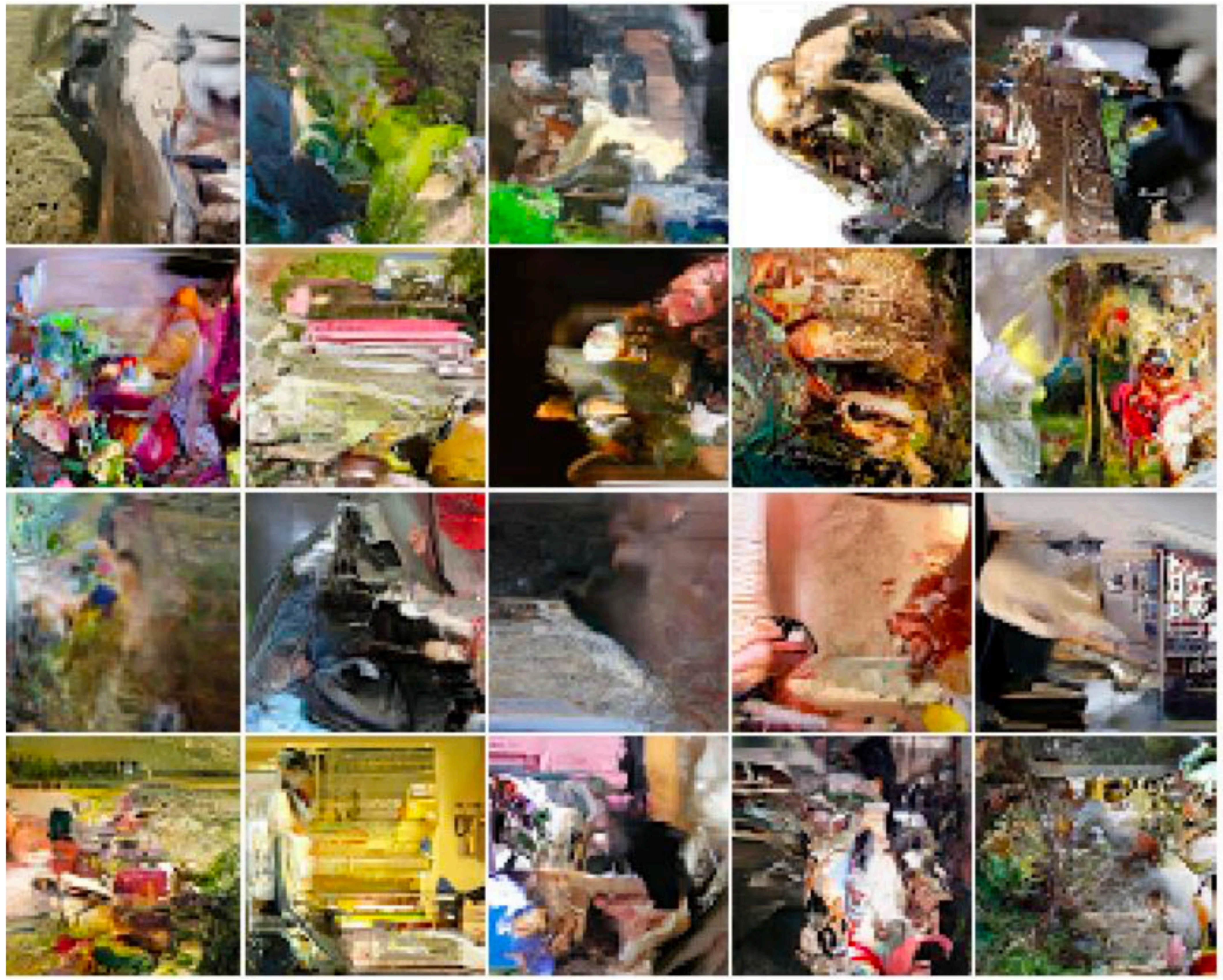
Can capture better global information (more visually coherent)

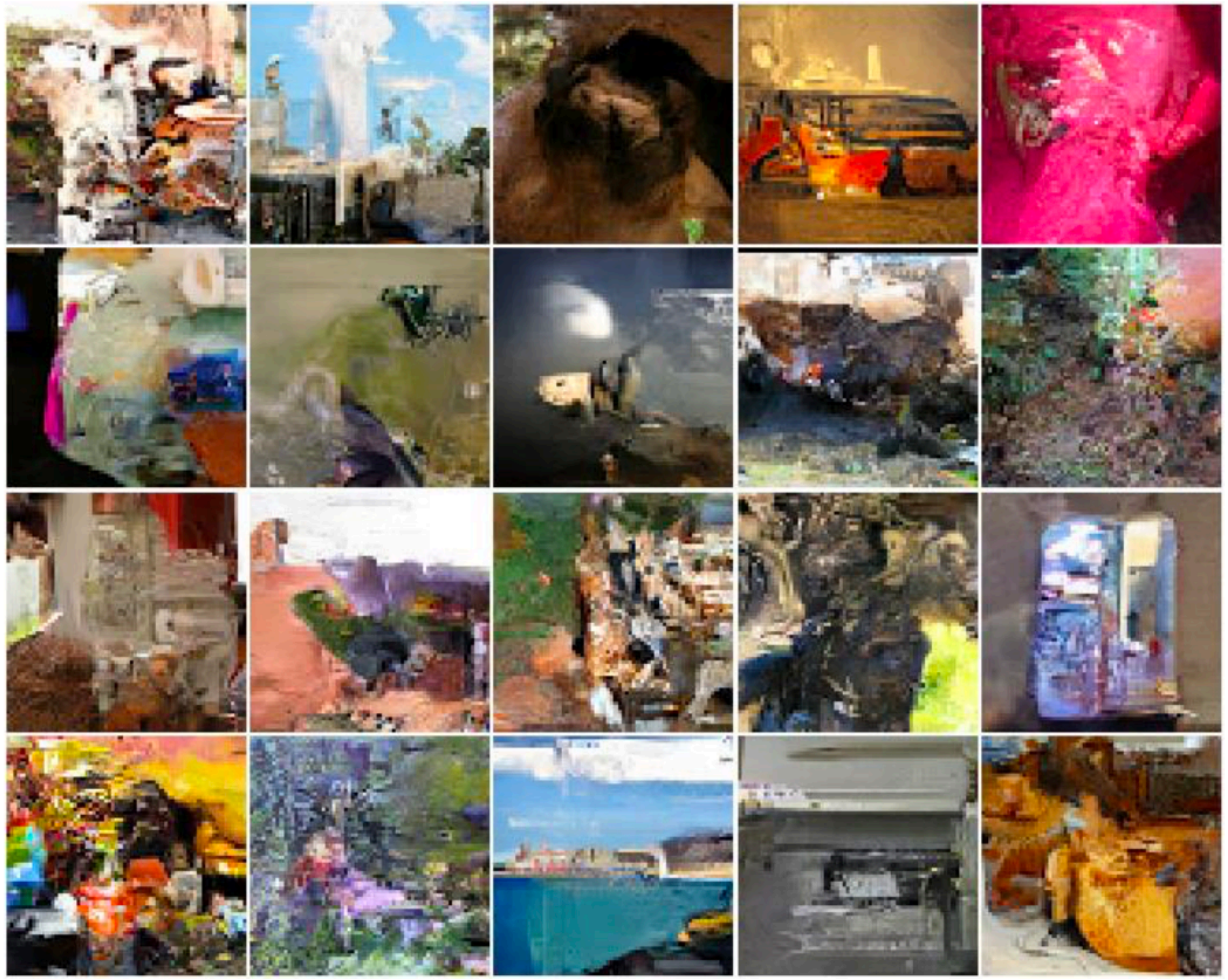


* slide from Hsiao-Ching Chang, Ameya Patil, Anand Bhattad

Multi-scale PixelRNN

[van der Oord et al., 2016]





* slide from Hsiao-Ching Chang, Ameya Patil, Anand Bhattad

Conditional Image Generation

[van der Oord et al., 2016]

Similar to PixelRNN/CNN but conditioned on a high-level image description vector $\mathbf{h}$

$$p(\mathbf{x}) = p(x_1, x_2, \ldots, x_{n^2})$$

$$\downarrow$$

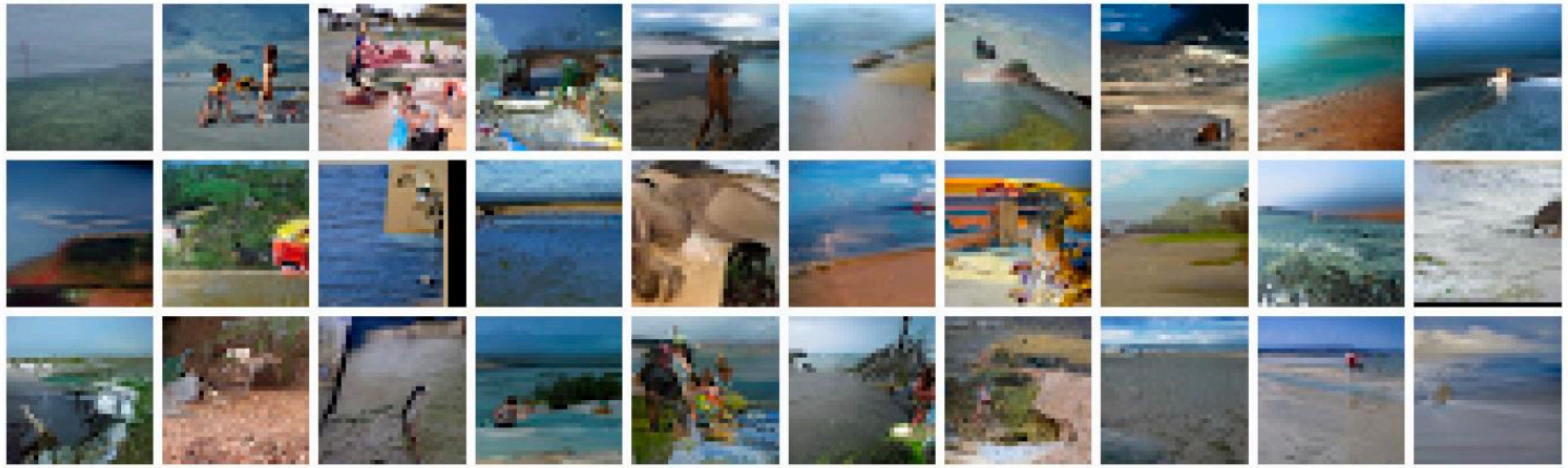
$$p(\mathbf{x}|\mathbf{h}) = p(x_1, x_2, \ldots, x_{n^2}|\mathbf{h})$$

* slide from Hsiao-Ching Chang, Ameya Patil, Anand Bhattad

Conditional Image Generation

[van der Oord et al., 2016]

African elephant



Sandbar

* slide from Hsiao-Ching Chang, Ameya Patil, Anand Bhattad