Lecture 4: Generative Models

Next few lectures: Generative models for direct image based rendering.

Current Image

Vision

3D Intrinsic Components

Edit Components

Graphics

New Image under different conditions

Change:

- Viewpoint
- Lighting
- Reflectance
- Background
- Attributes
- Many others…

Implicit: Use a Neural Network (Conditional Generative networks) Often, end-to-end.

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.

Classification

Cat

Supervised vs Unsupervised Learning

Supervised Learning

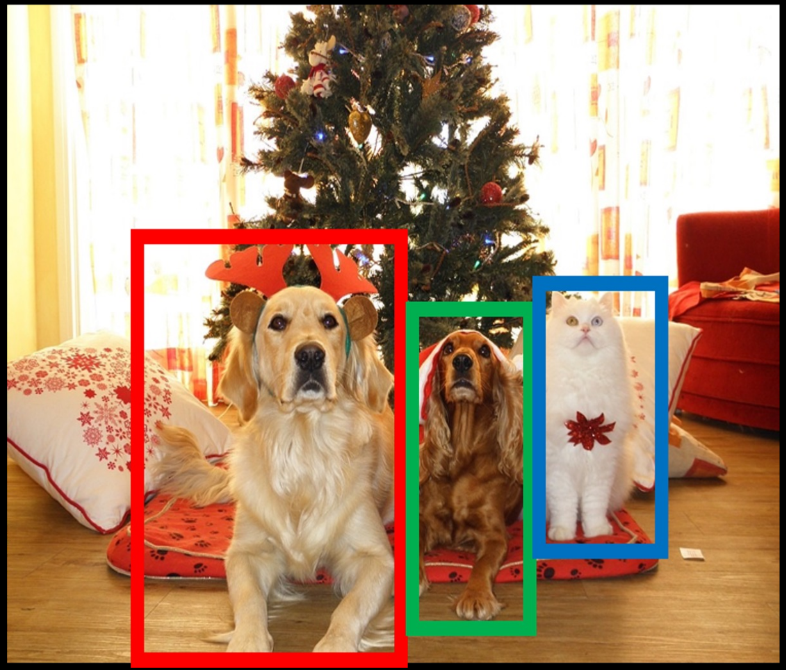
Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map x -> y

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

Object Detection



DOG, DOG, CAT

Supervised vs Unsupervised Learning

Supervised Learning

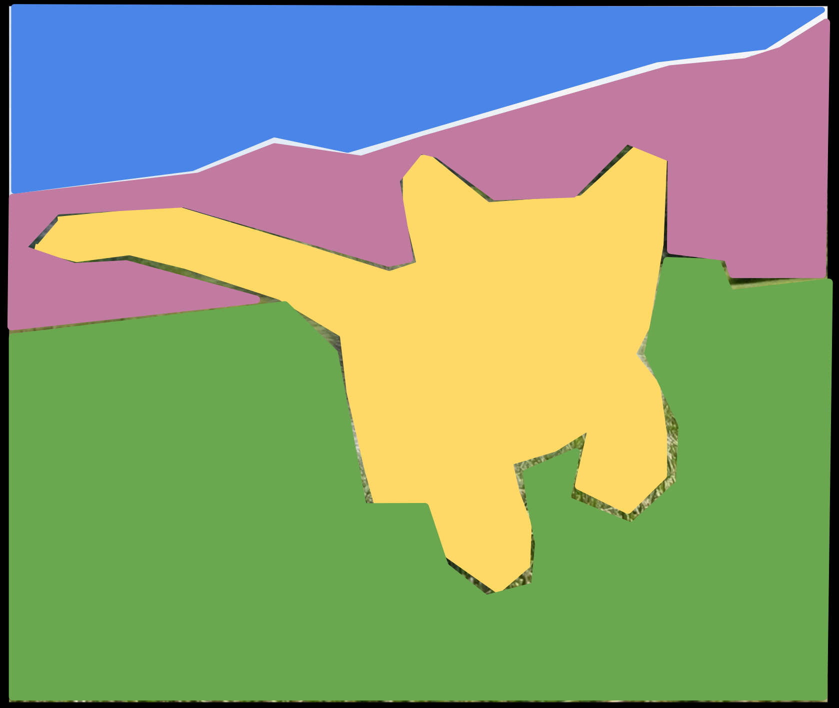
Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map x -> y

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

Semantic Segmentation



GRASS, CAT, TREE, SKY

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.

Image captioning

A cat sitting on a suitcase on the floor

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.

Unsupervised Learning

Data: x

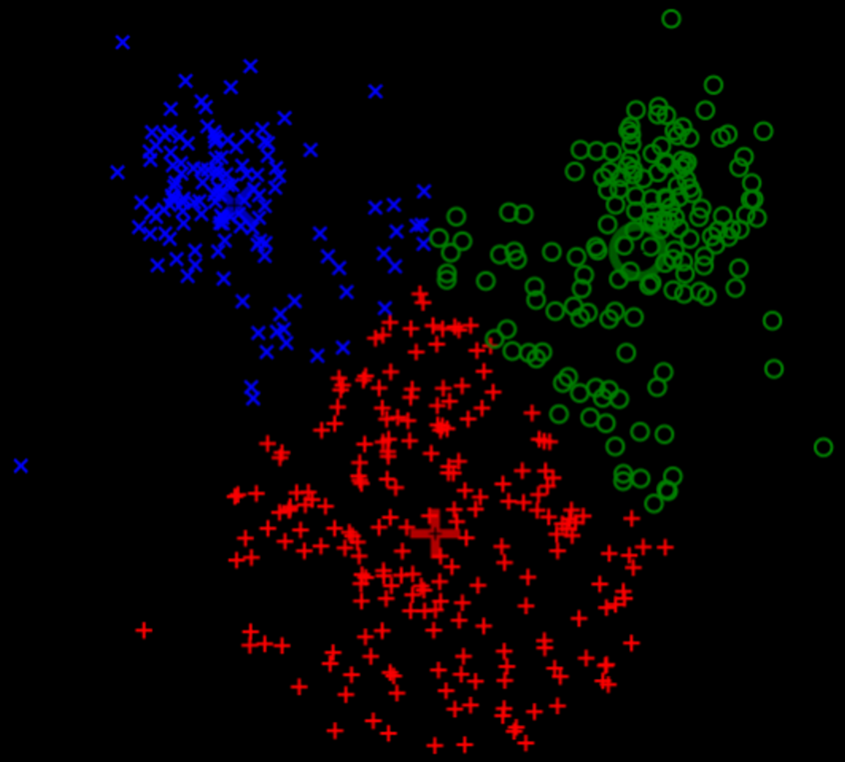
Just data, no labels!

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

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

Supervised vs Unsupervised Learning

Clustering
(e.g. K-Means)



Unsupervised Learning

Data: x

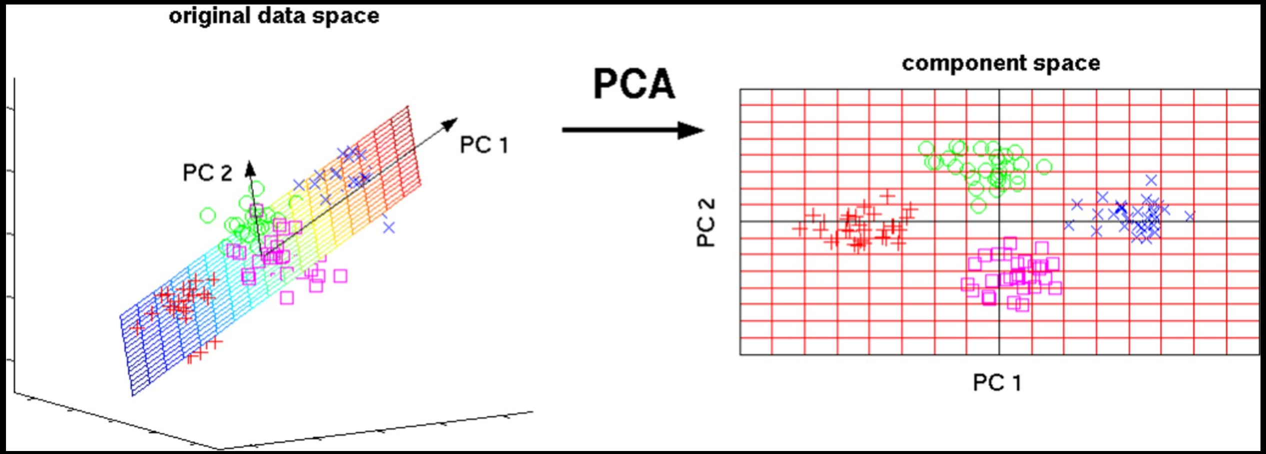
Just data, no labels!

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

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

Supervised vs Unsupervised Learning

Dimensionality Reduction
(e.g. Principal Components Analysis)



3D → 2D

Unsupervised Learning

Data: x

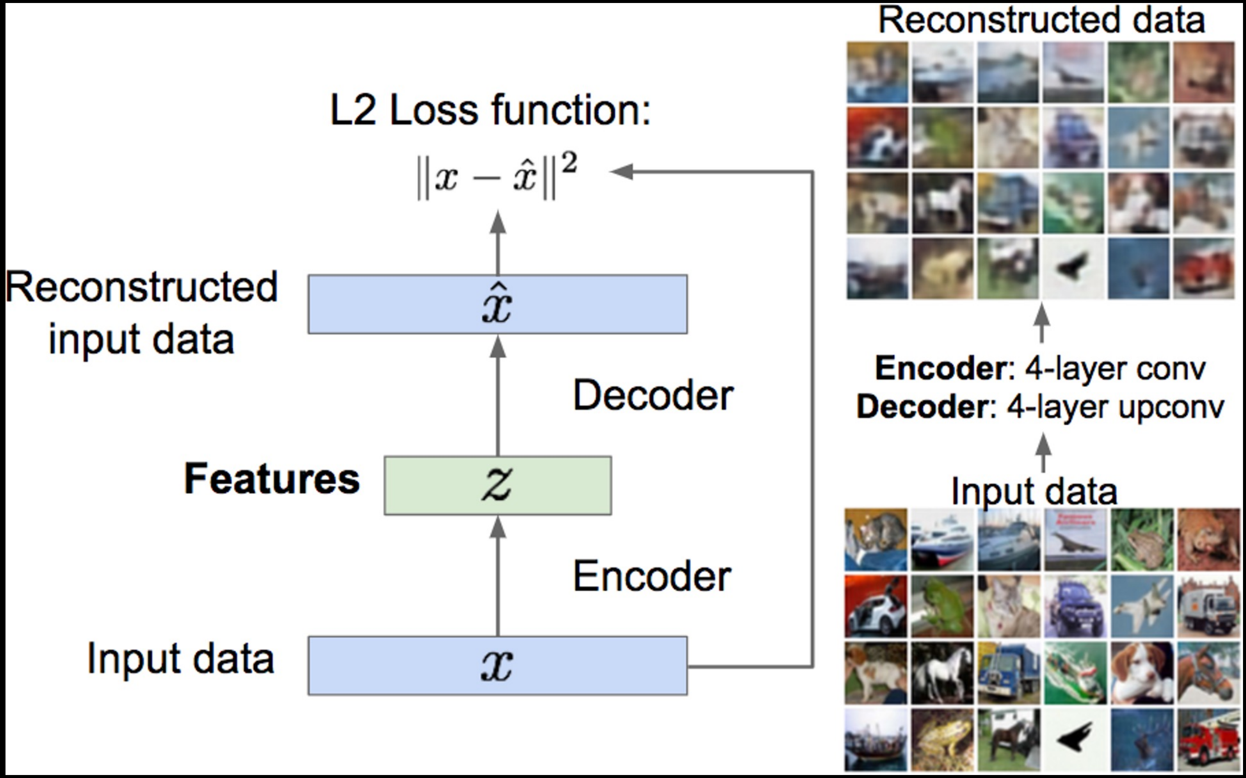
Just data, no labels!

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

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

Supervised vs Unsupervised Learning

Feature Learning
(e.g. autoencoders)



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.

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.

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.

Discriminative vs Generative Models

Discriminative Model:
Learn a probability distribution p(y|x)

Generative Model:
Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Data: x

Label: y

Cat

Discriminative vs Generative Models

Discriminative Model:
Learn a probability distribution $p(y|x)$

Generative Model:
Learn a probability distribution $p(x)$

Conditional Generative Model: Learn $p(x|y)$

Data: x

Label: y

Cat

Probability Recap:

Density Function
$p(x)$ assigns a positive number to each possible x; higher numbers mean x is more likely

Density functions are **normalized**:

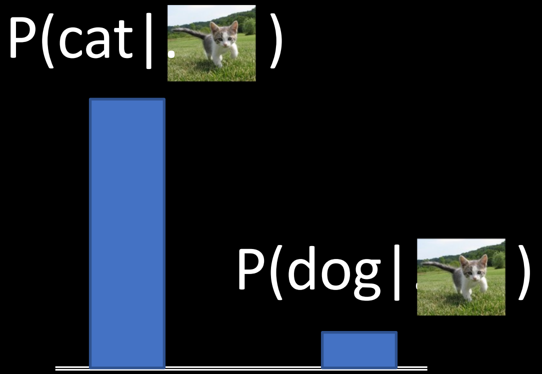
$$\int_X p(x)dx = 1$$

Different values of x **compete** for density

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Data: x

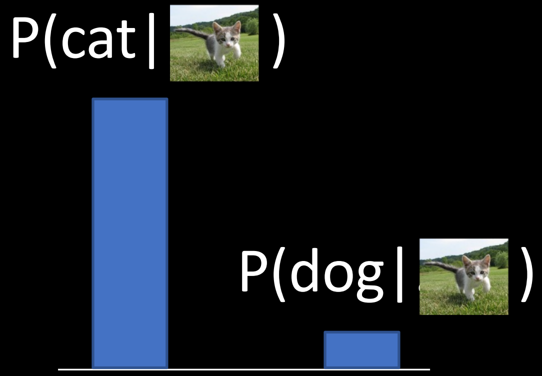


Generative Model: Learn a probability distribution p(x)

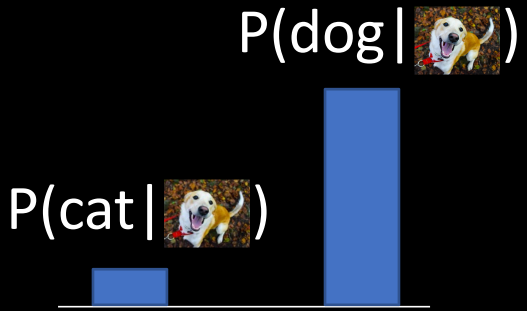
Conditional Generative Model: Learn p(x|y)

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution $p(y|x)$



Generative Model: Learn a probability distribution $p(x)$



Conditional Generative Model: Learn $p(x|y)$

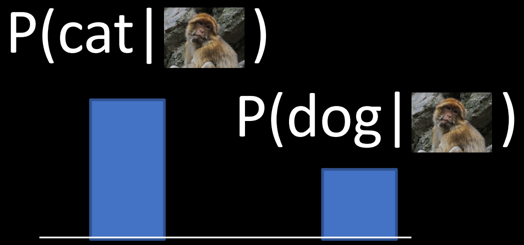
Discriminative model: the possible labels for each input "compete" for probability mass. But no competition between **images**

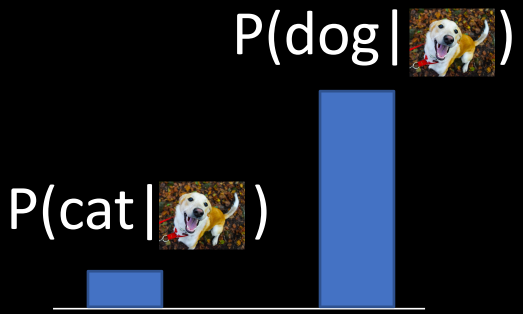
Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

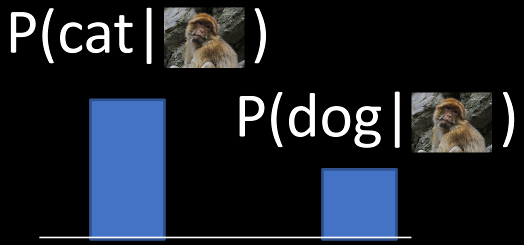




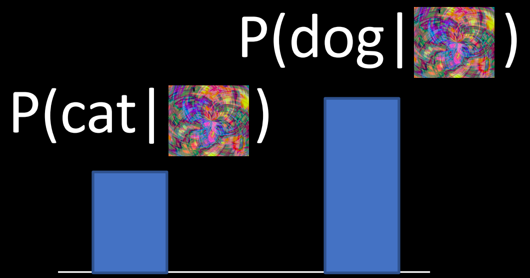
Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)



Generative Model: Learn a probability distribution p(x)



Conditional Generative Model: Learn p(x|y)

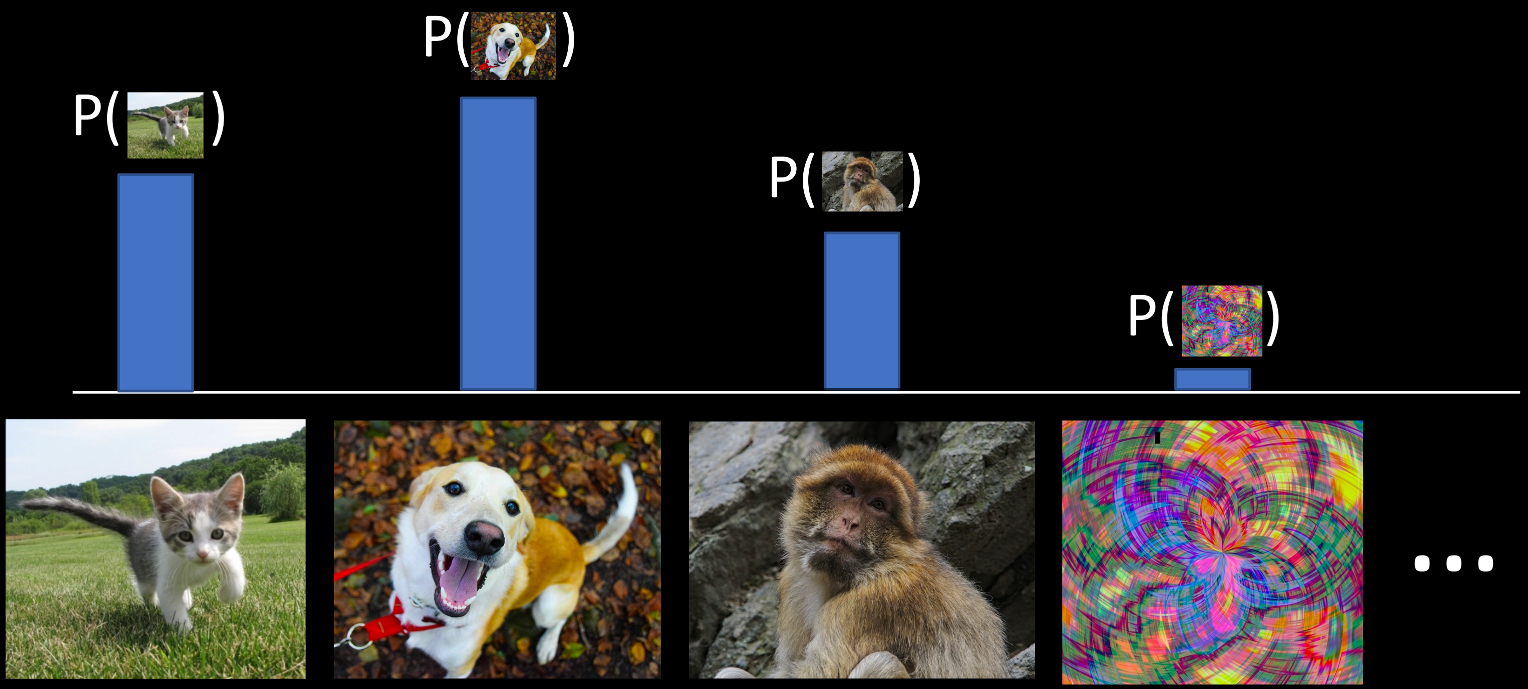
Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution $p(y|x)$

Generative Model: Learn a probability distribution $p(x)$

Conditional Generative Model: Learn $p(x|y)$



Generative model: All possible images compete with each other for probability mass

Requires deep image understanding! Is a dog more likely to sit or stand? How about 3-legged dog vs 3-armed monkey?

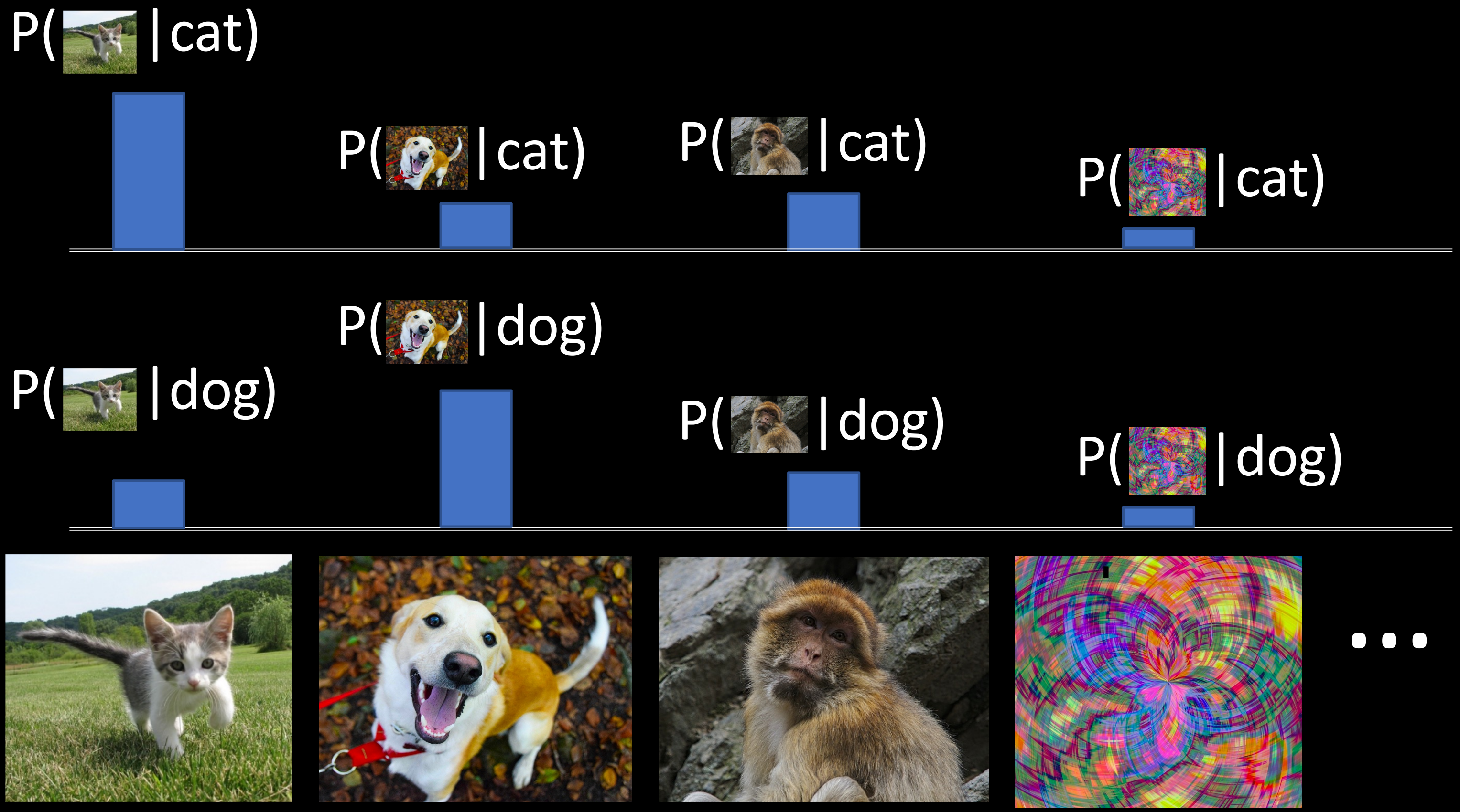
Model can “reject” unreasonable inputs by assigning them small values

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)



Conditional Generative Model: Each possible label induces a competition among all images

Discriminative vs Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Recall **Bayes' Rule:**

$$P(x \mid y) = \frac{P(y \mid x)}{P(y)} P(x)$$

Discriminative vs Generative Models

Discriminative Model:
Learn a probability distribution p(y|x)

Generative Model:
Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Recall **Bayes' Rule:**

$$\underbrace{\boxed{P(x \mid y)}}_{\text{Conditional Generative Model}} = \frac{\overbrace{\boxed{P(y \mid x)}}^{\text{Discriminative Model}}}{\underbrace{\boxed{P(y)}}_{\text{Prior over labels}}} \overbrace{\boxed{P(x)}}^{\text{(Unconditional) Generative Model}}$$

We can build a conditional generative model from other components!

What can we do with a discriminative model?

- **Discriminative Model:** Learn a probability distribution p(y|x) → Assign labels to data; Feature learning (with labels)

- **Generative Model**: Learn a probability distribution p(x)

- **Conditional Generative Model:** Learn p(x|y)

What can we do with a generative model?

- **Discriminative Model:** Learn a probability distribution $p(y|x)$ → Assign labels to data; Feature learning (with labels)

- **Generative Model**: Learn a probability distribution $p(x)$ → Detect outliers; Feature learning (without labels); Sample to **generate** new data

- **Conditional Generative Model:** Learn $p(x|y)$

What can we do with a generative model?

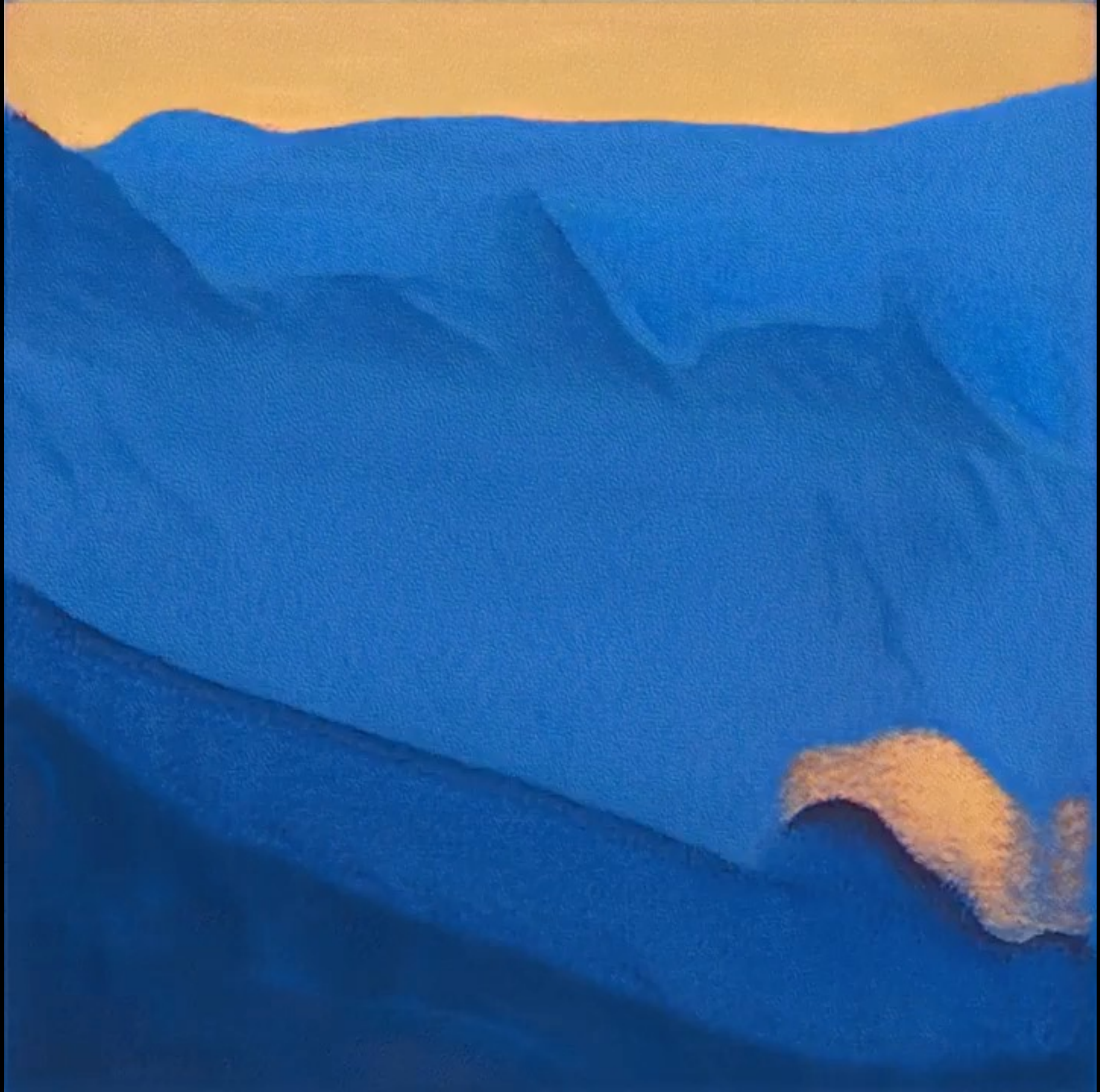
- **Discriminative Model:** Learn a probability distribution $p(y|x)$ → Assign labels to data; Feature learning (with labels)

- **Generative Model**: Learn a probability distribution $p(x)$ → Detect outliers; Feature learning (without labels); Sample to **generate** new data

- **Conditional Generative Model:** Learn $p(x|y)$ → Assign labels, while rejecting outliers!; Generate new data conditioned on input labels

Introduction to Generative Models (Conditional and Unconditional)

What cool things can we do with it?



Credits: Neural Synesthesia

PLAY ABOUT METHODS LEARN PRESS CONTACT BOOK CALLING BS

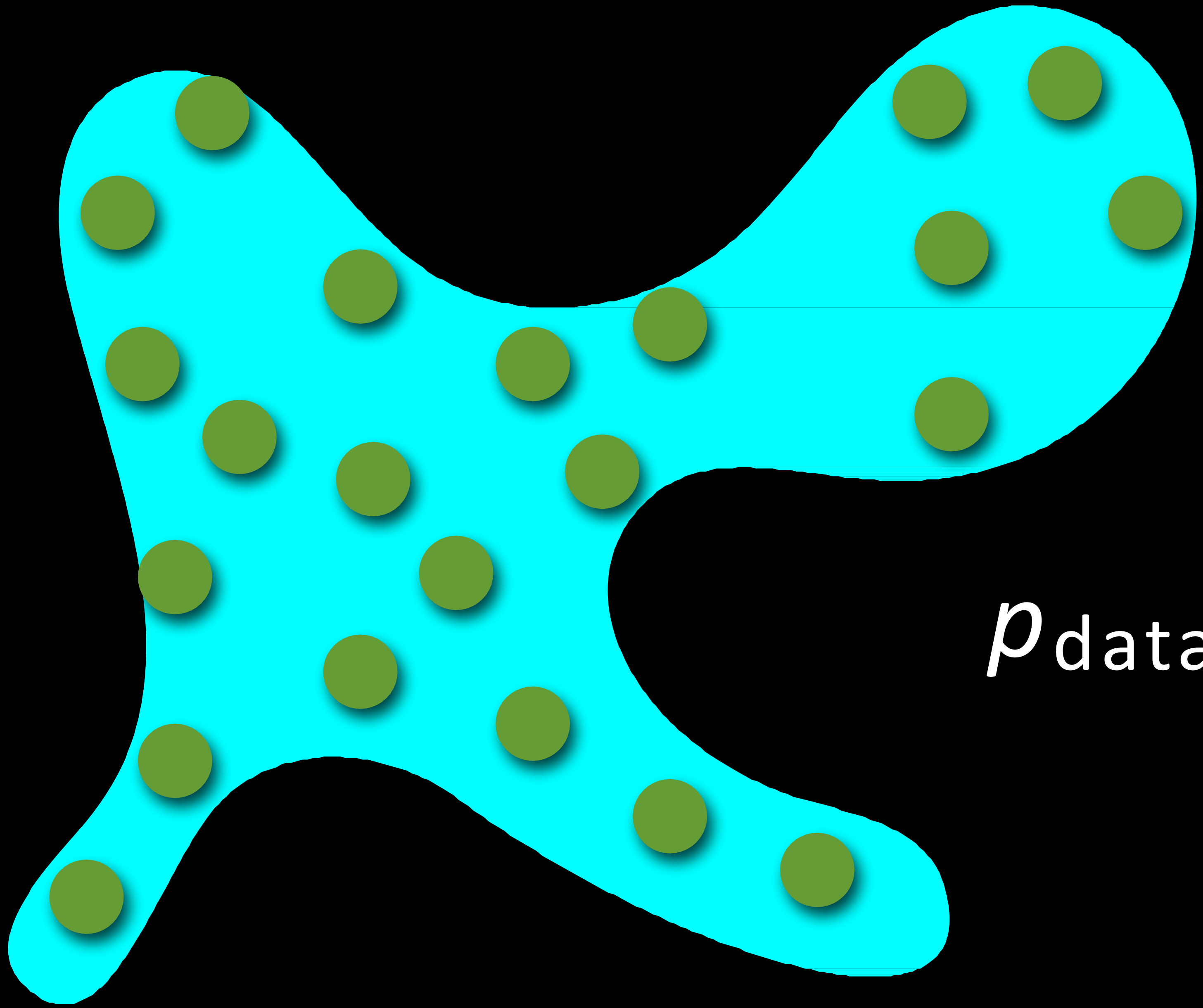
Click on the person who is real.

https://www.whichfaceisreal.com/index.php

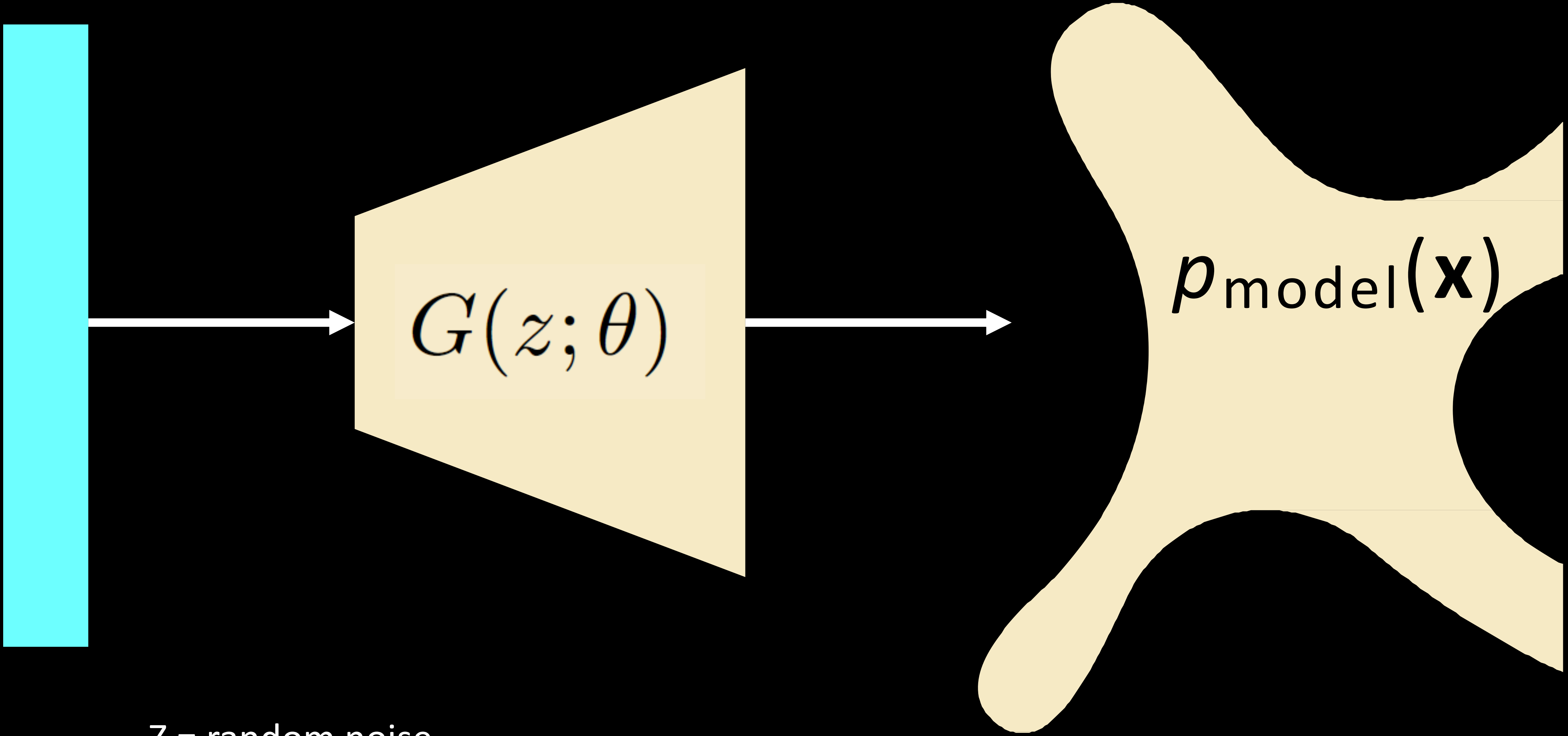
$$p_{\text{data}}(\mathbf{x})$$

$$p_{\mathrm{data}}(\mathbf{x}) \approx p_{\mathrm{model}}(\mathbf{x})$$

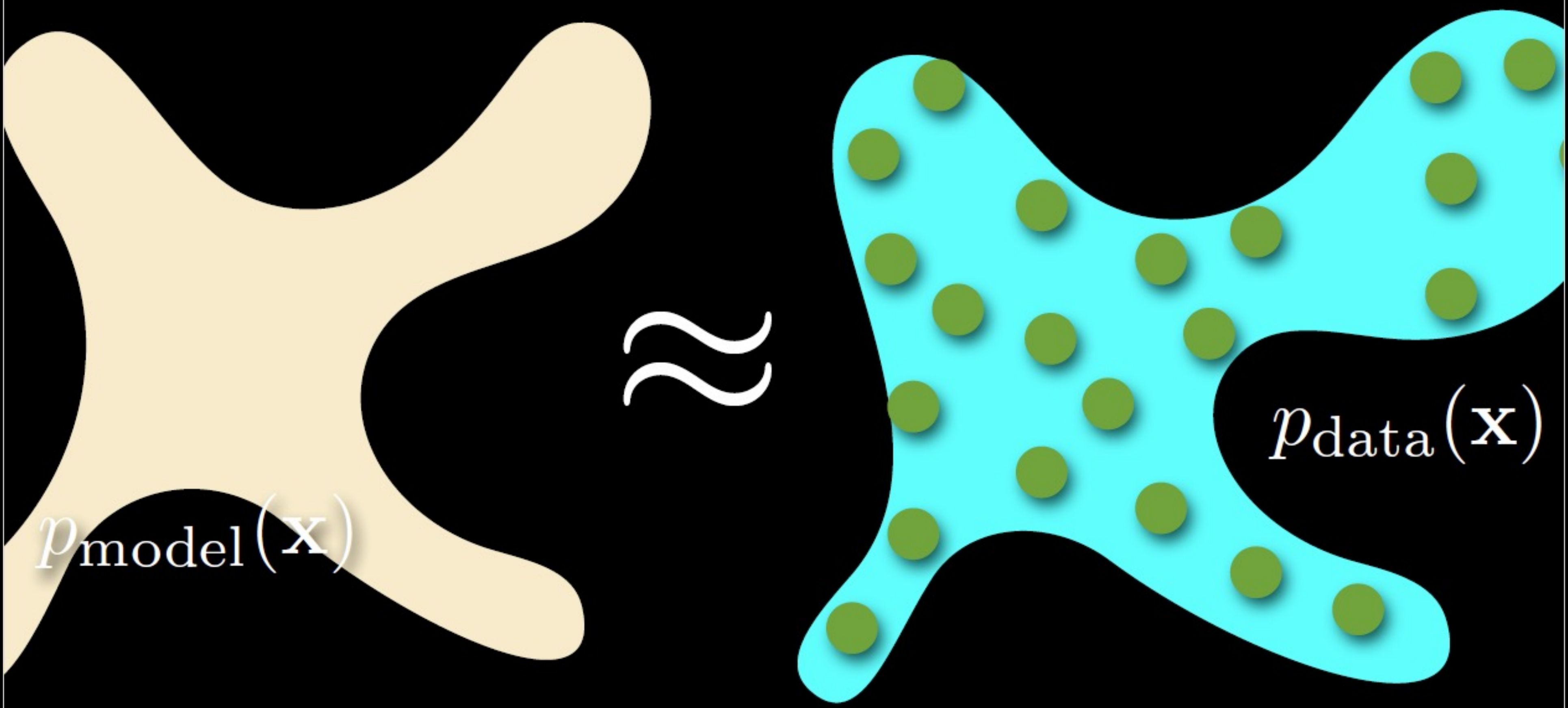

TNT



$p_{\text{data}}(\mathbf{x})$



Z = random noise
(samples from latent space)



$p_{\text{model}}(\mathbf{x})$
$\approx$
$p_{\text{data}}(\mathbf{x})$

Taxonomy of Generative Models

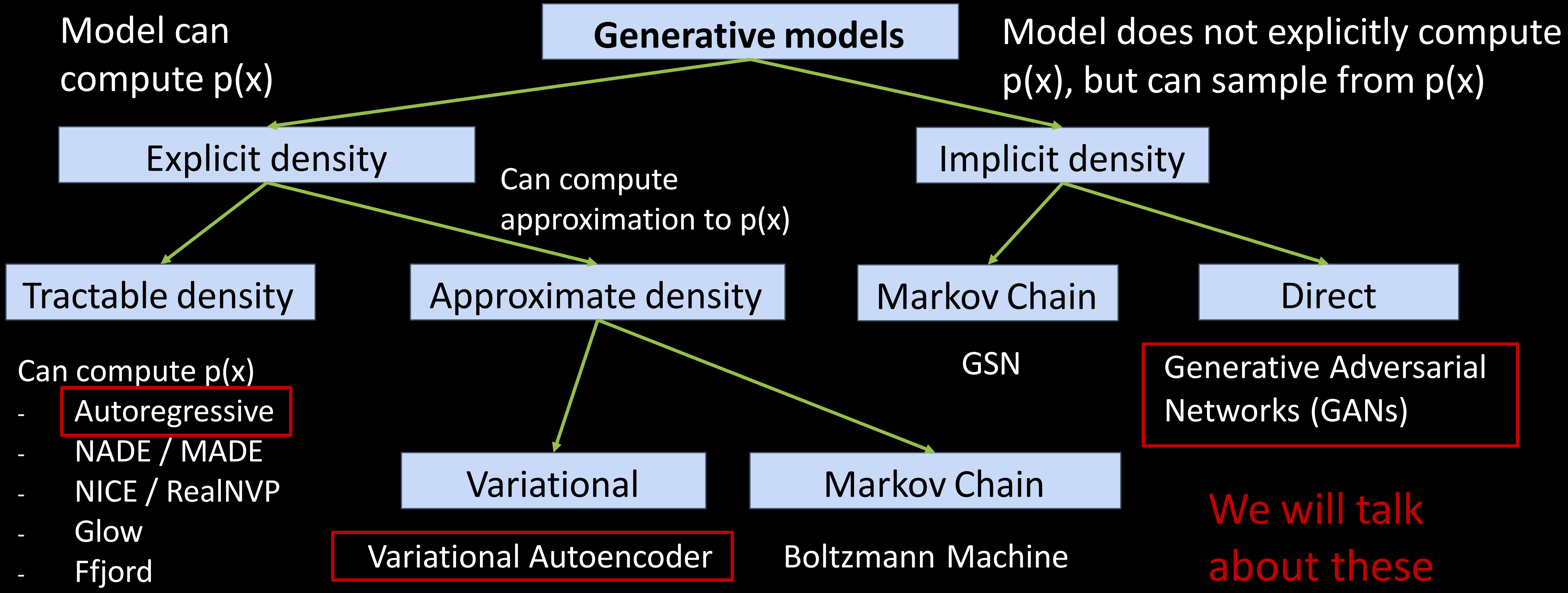


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Autoregressive models: PixelRNN & PixelCNN

Explicit Density Estimation

Learn weights W

Goal: Write down an explicit function for $p(x) = f(x, \boxed{W})$

Given dataset $x^{(1)}, x^{(2)}, \ldots x^{(N)}$, train the model by solving:

$$W^* = \arg\max_{W} \prod_i p(x^{(i)})$$

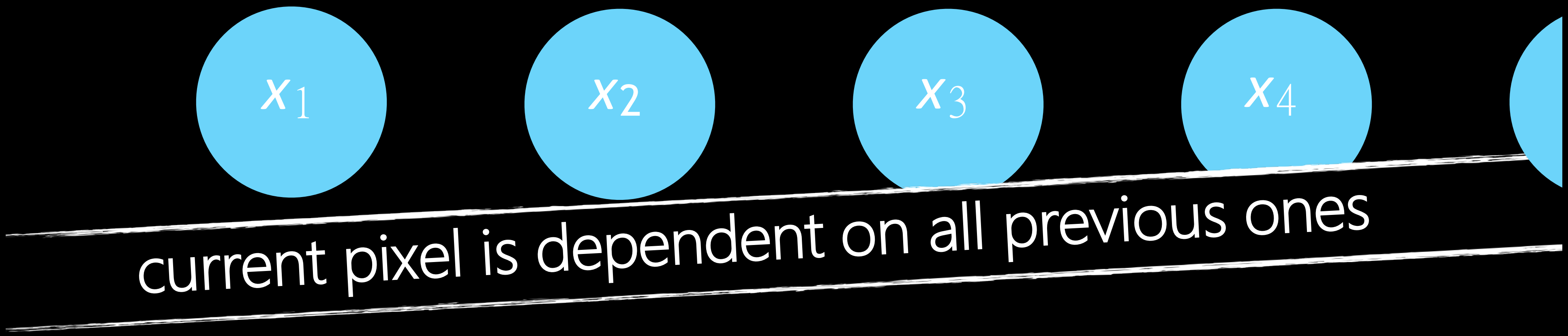
Maximize probability of training data
(Maximum likelihood estimation)

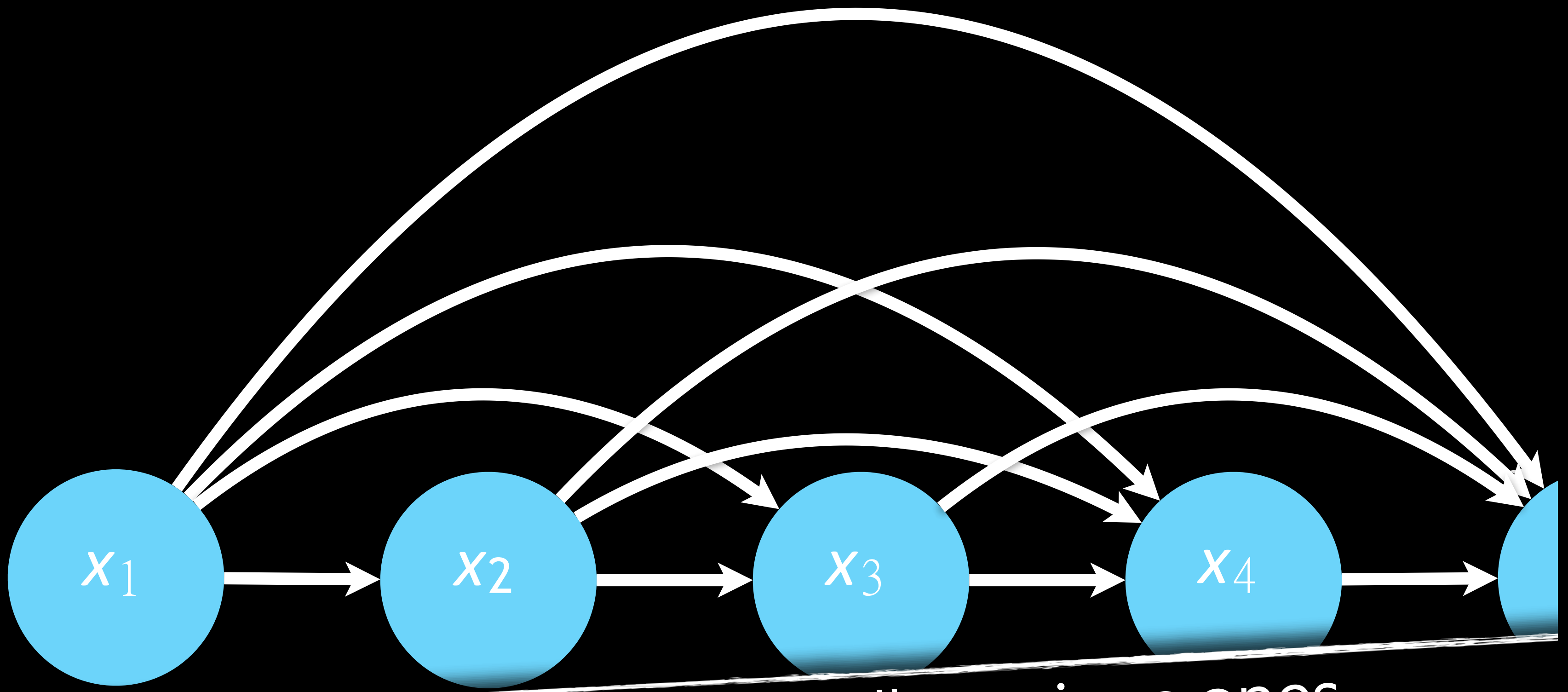
$$= \arg\max_{W} \sum_i \log p(x^{(i)})$$

Log trick to exchange product for sum

$$= \arg\max_{W} \sum_i \log f(x^{(i)}, W)$$

This will be our loss function!
Train with gradient descent


x_1
x_2
x_3
x_4
current pixel is dependent on all previous ones



current pixel is dependent on all previous ones

$$p_{\mathrm{model}}(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid x_1, \ldots, x_{i-1})$$

use chain rule to decompose

$$p_{\mathrm{model}}(\mathbf{x}) = \prod_{i=1}^{N} p(x_i | x_1, \ldots, x_{i-1})$$

$$p_{\mathrm{model}}(\mathbf{x}) = \prod^{N} p(x_i | x_1, \ldots, x_{i-1})$$

probability of ith pixel given all previous pixels

$$p_{\text{model}}(\mathbf{x}) = \prod^{N} p(x_i|x_1, \ldots, x_{i-1})$$

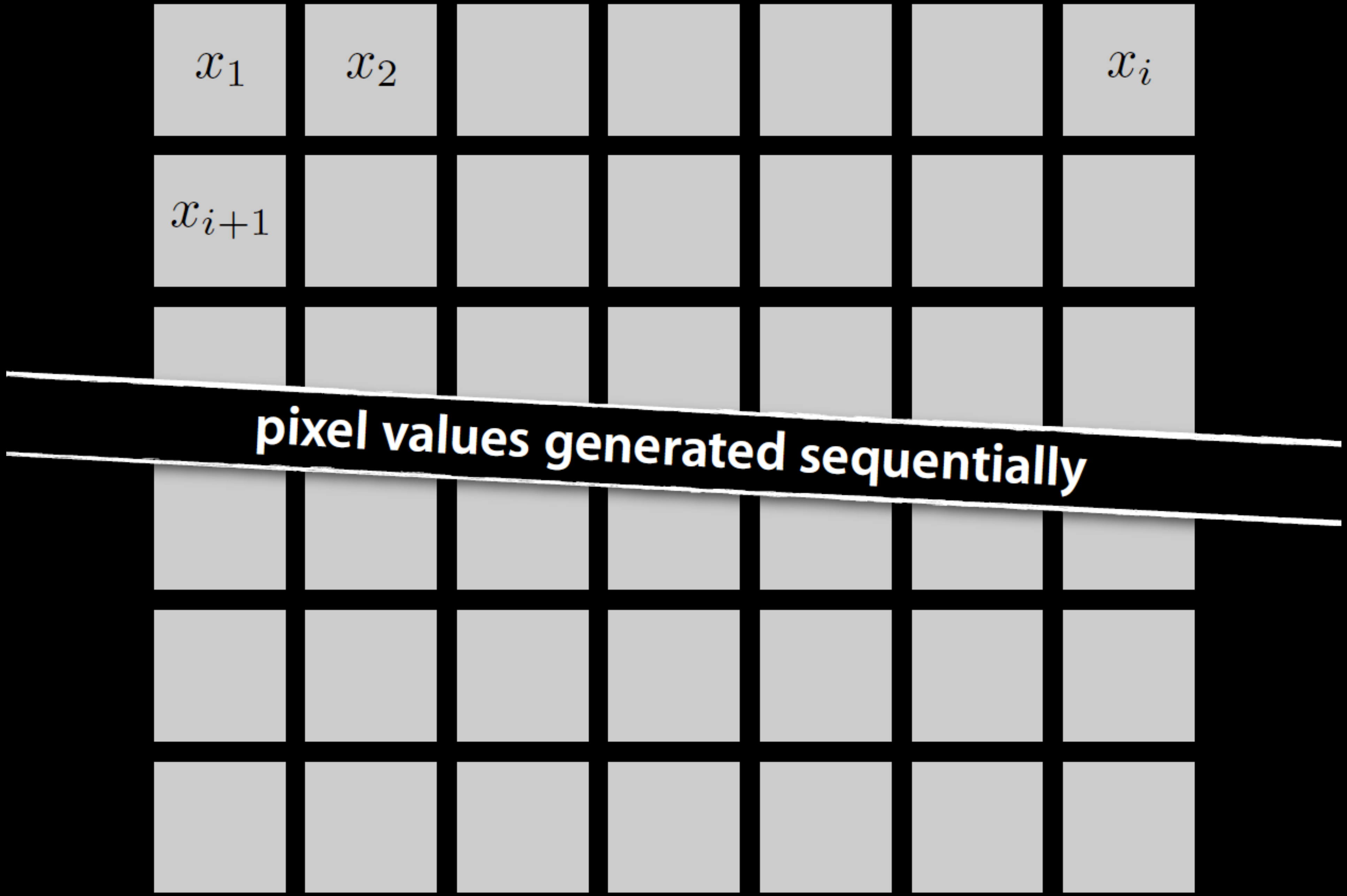
express distribution over pixel values as neural network

$$p_{\text{model}}(\mathbf{x}) = \prod_{i=1}^{N} p(x_i | x_1, \ldots, x_{i-1})$$

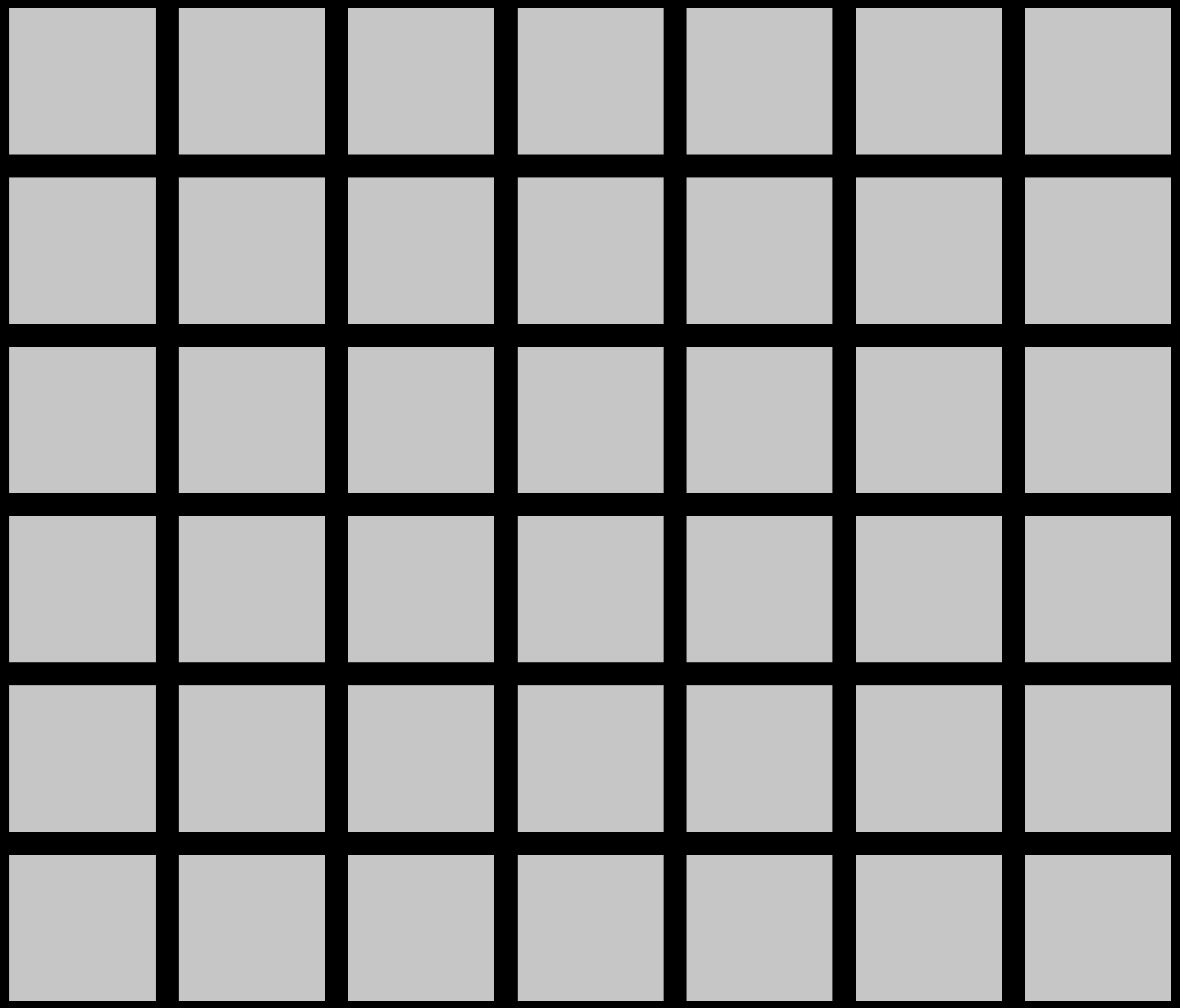
maximize likelihood of training data with respect to network weights

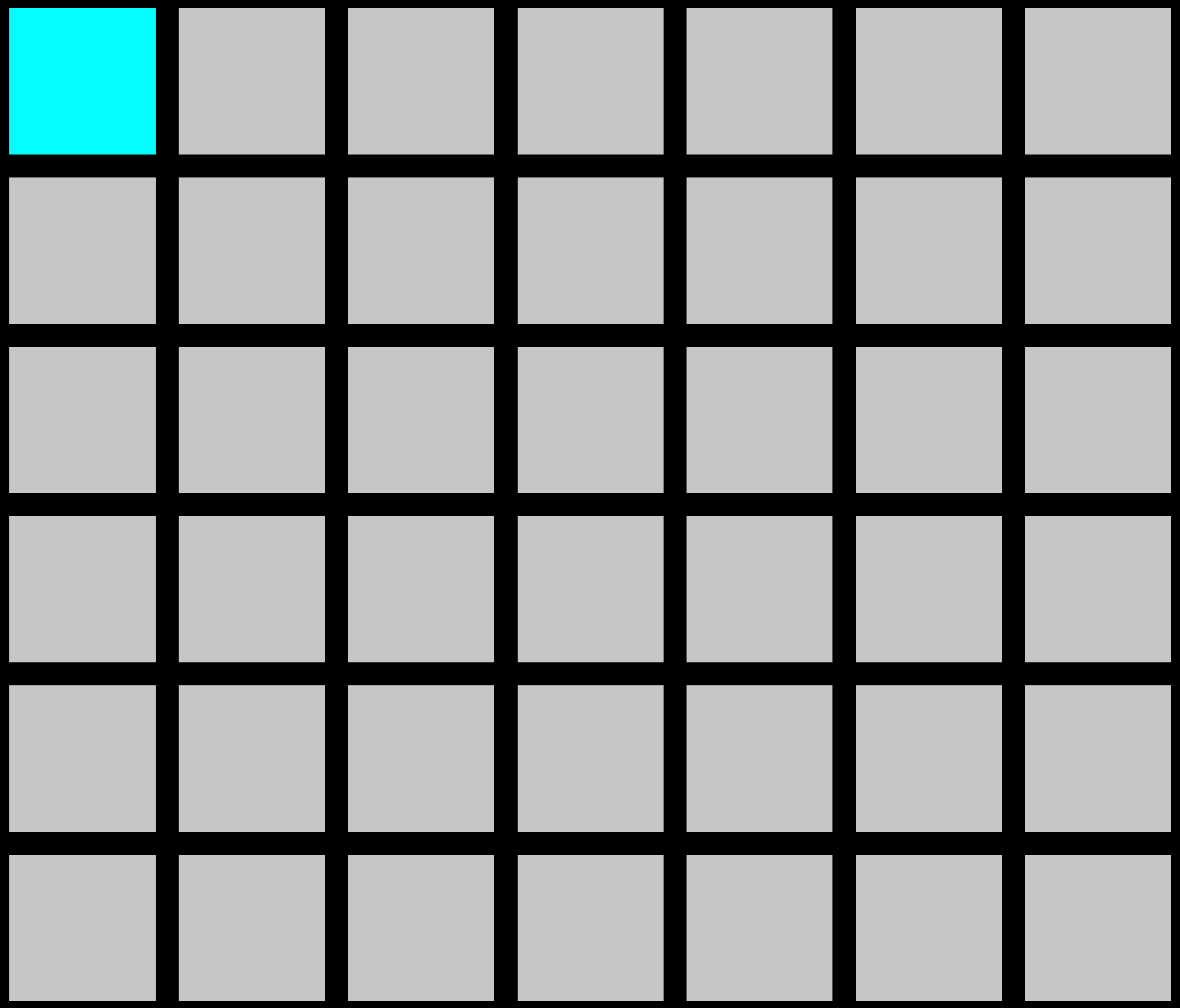
Maximum Likelihood estimate: $$arg \max_{\theta} \; log f_{model}(x;\theta)$$

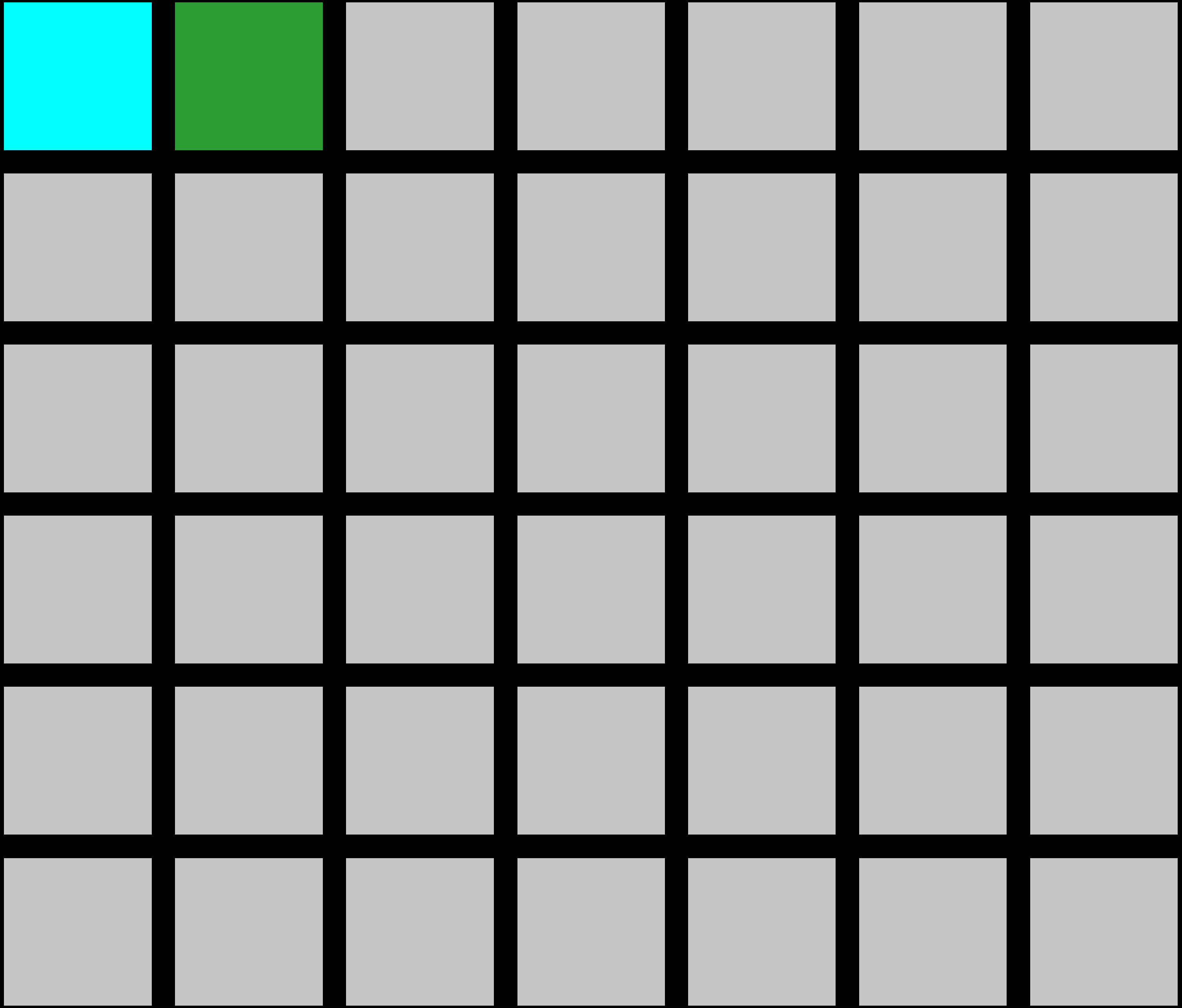
- Maximize this estimator w.r.t. neural network parameter $\boldsymbol{\theta}$.
- In practice we minimize negative of the above estimator as our loss function.

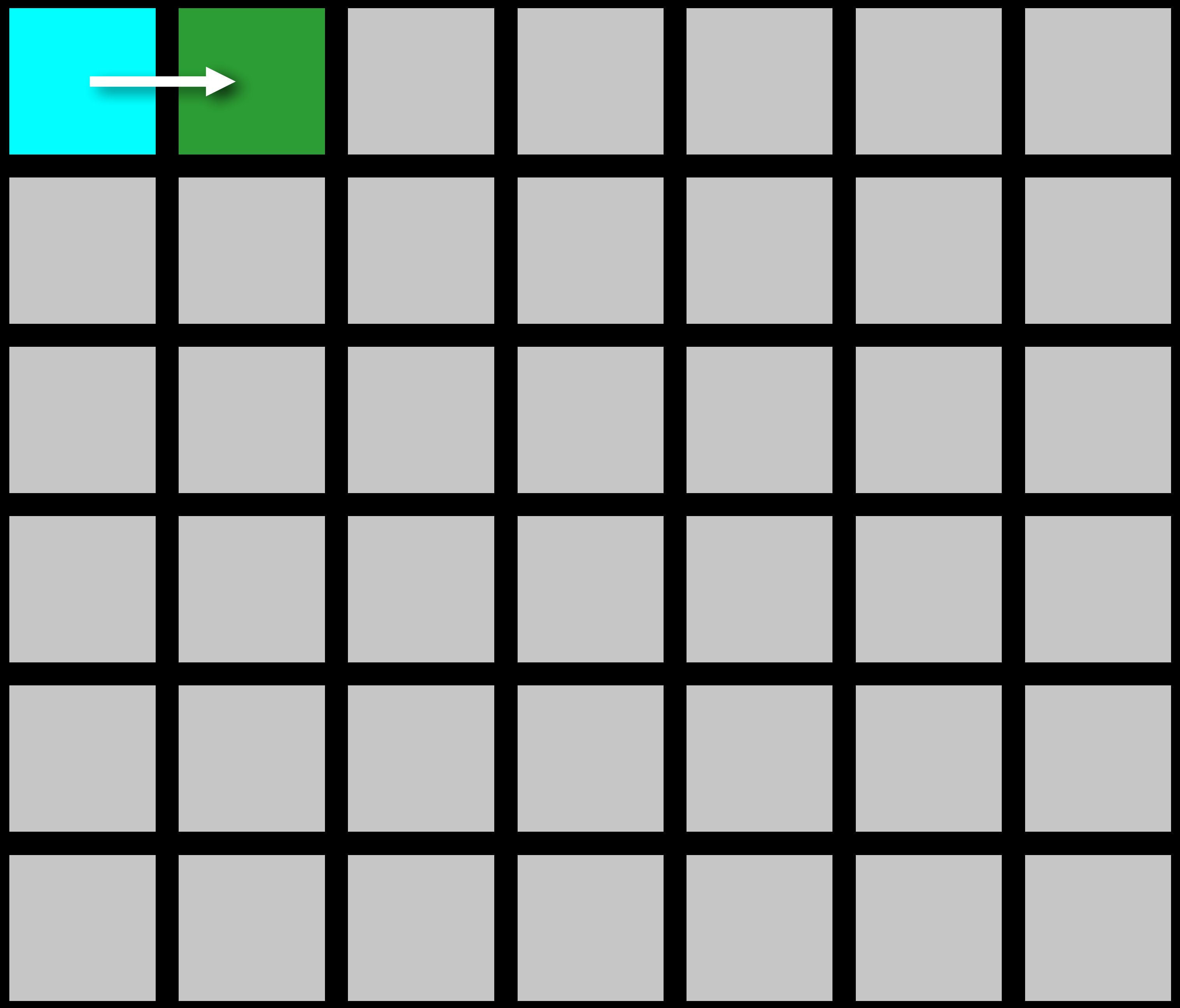

x_1
x_2
x_i
x_{i+1}
pixel values generated sequentially

PixelRNN

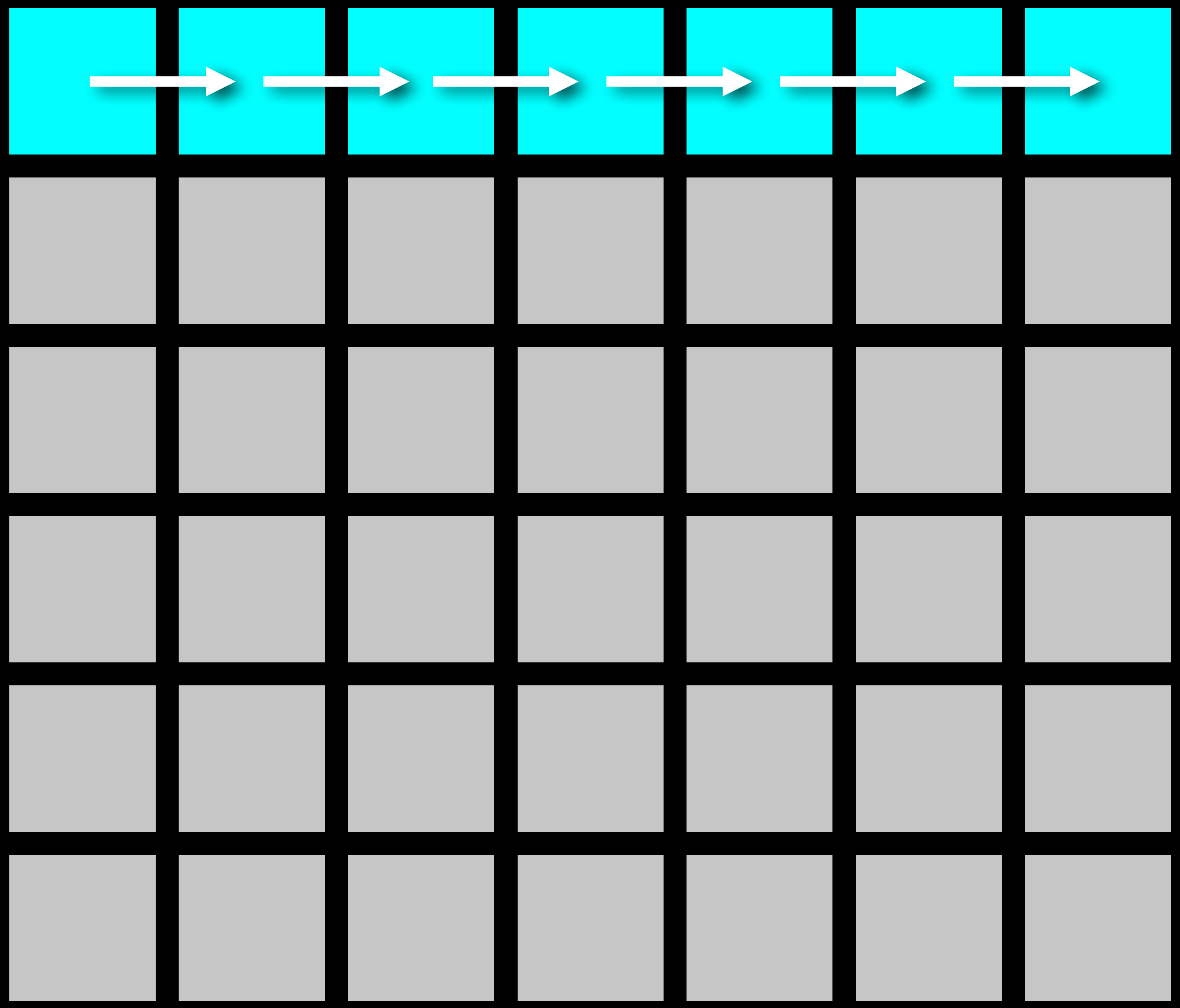


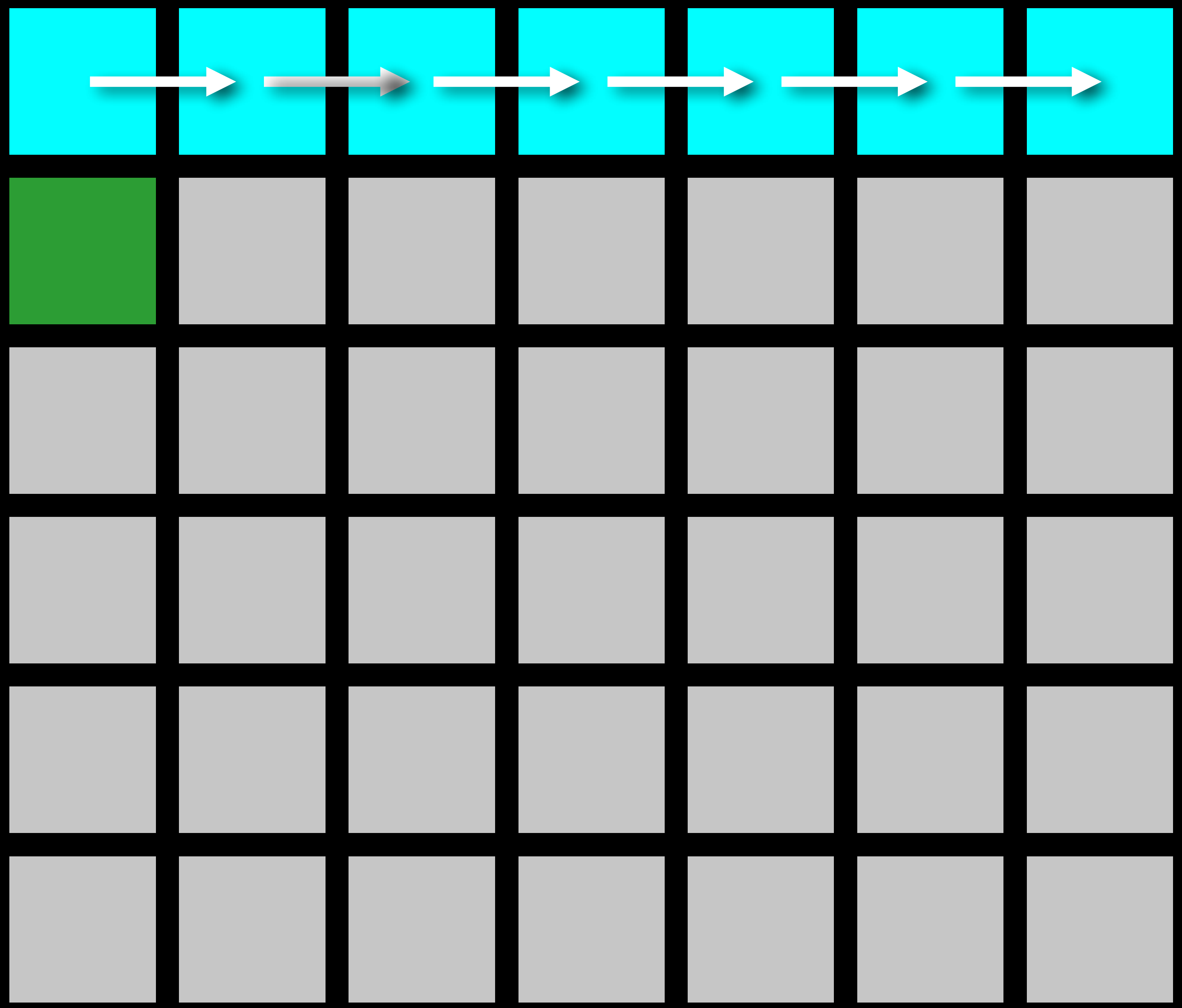


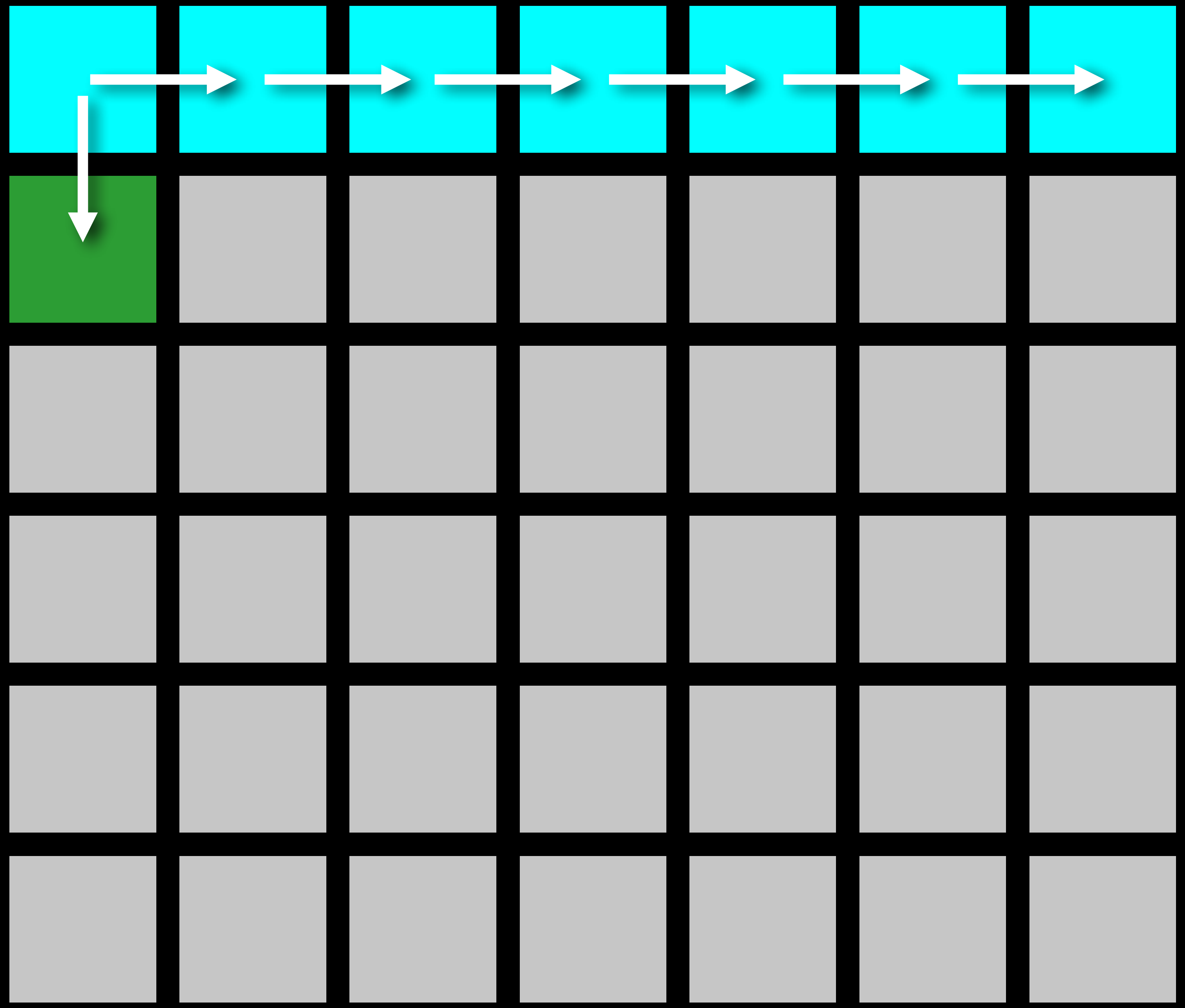


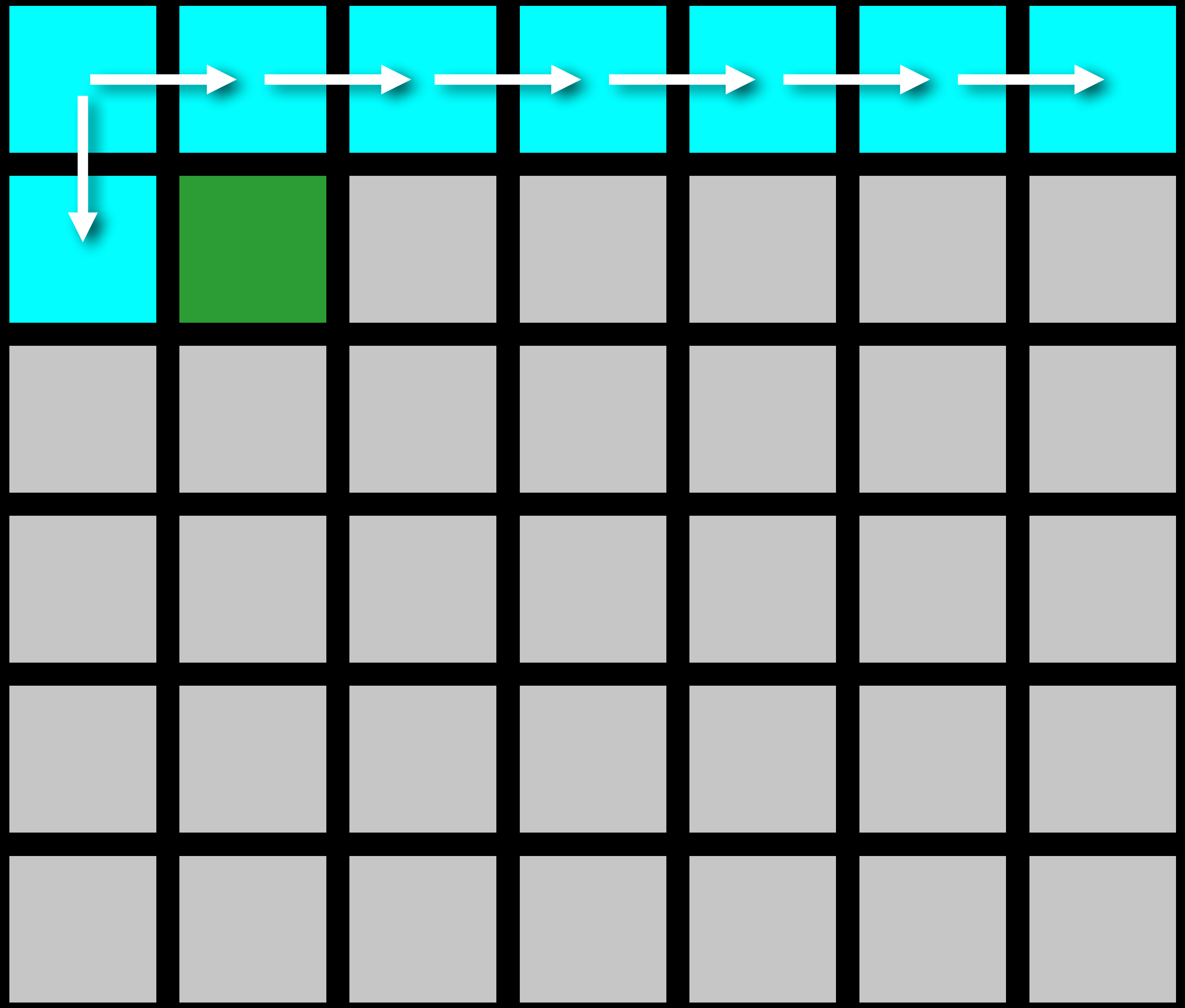


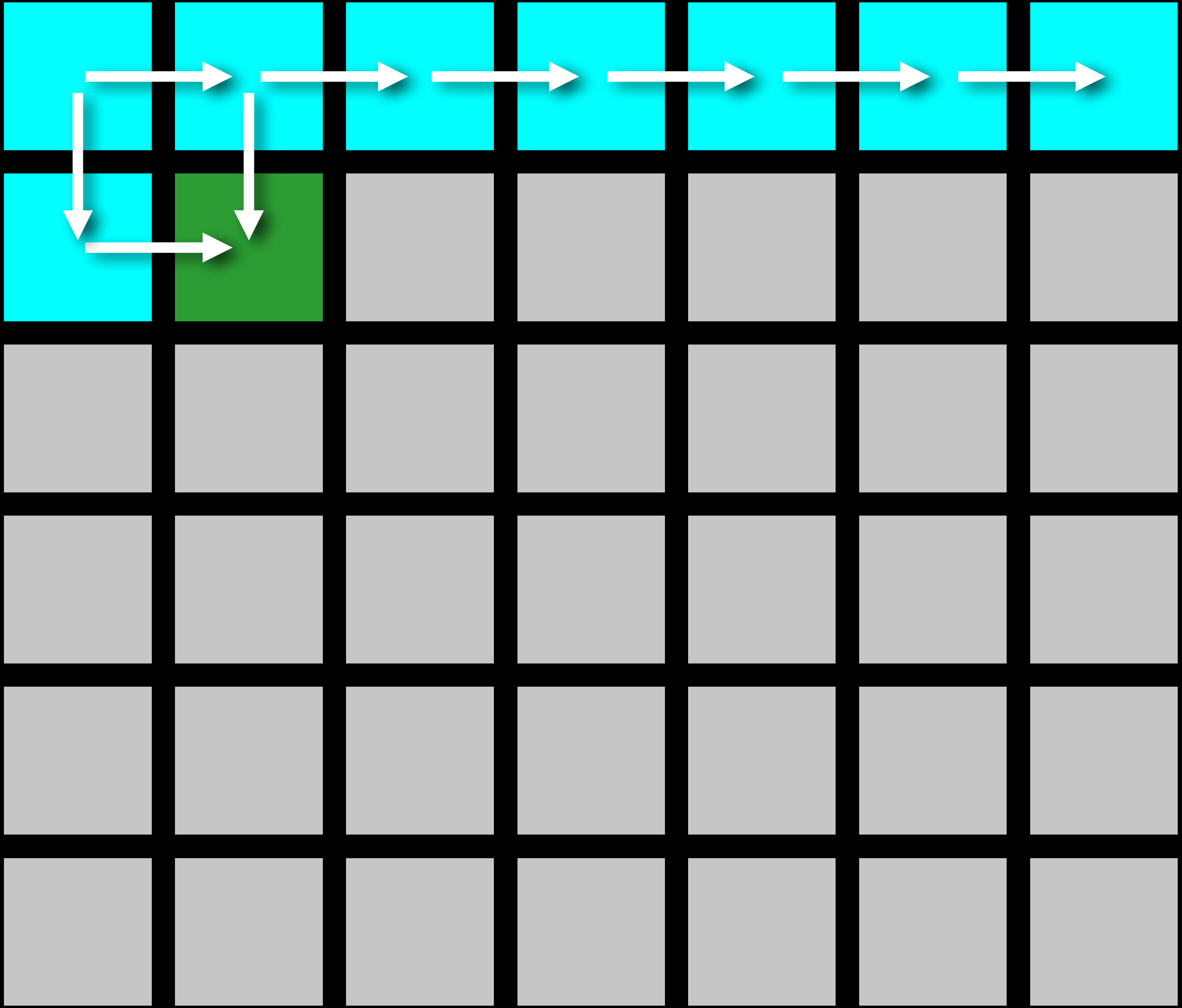

dependency on previous pixels modelled using an RNN



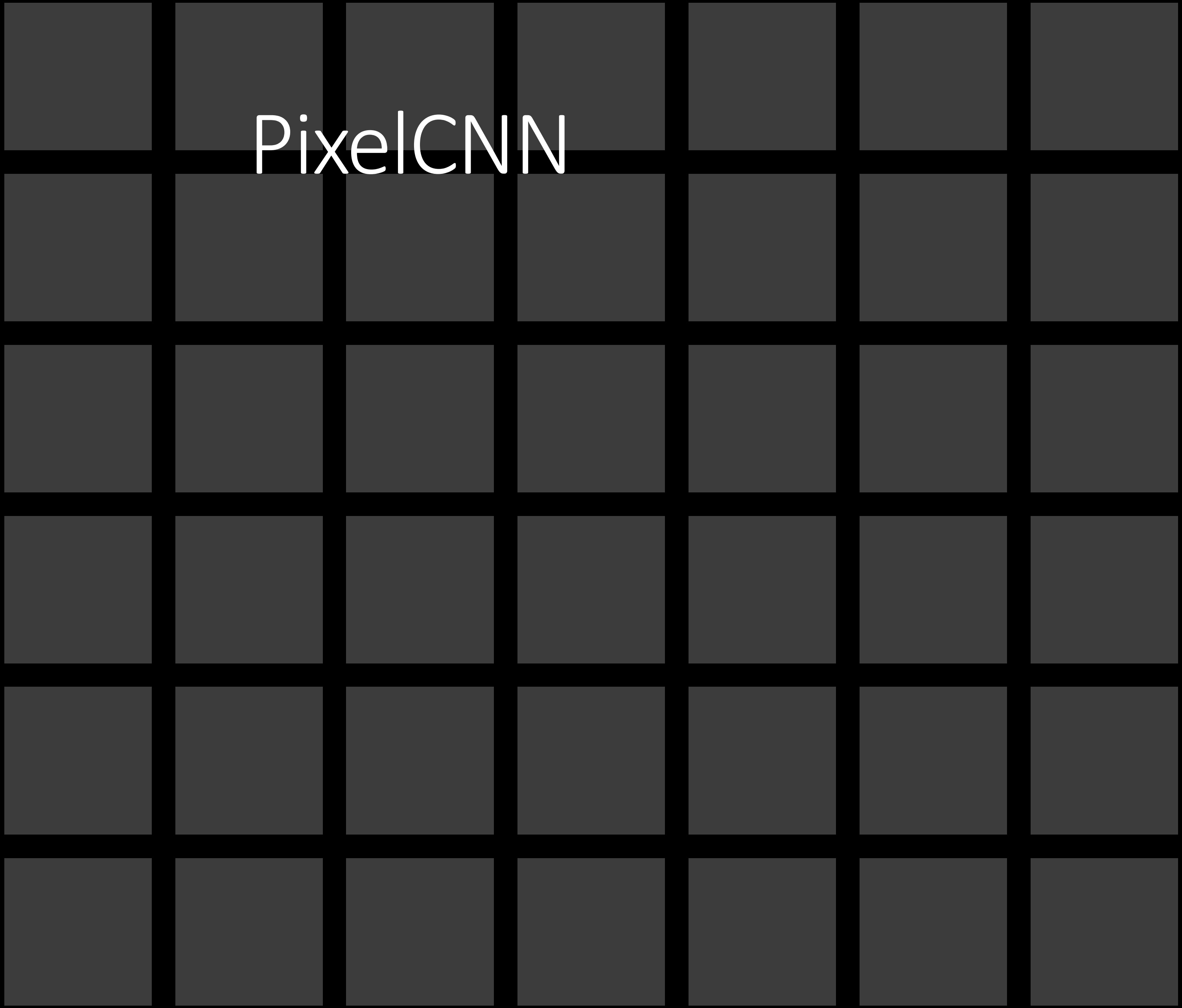


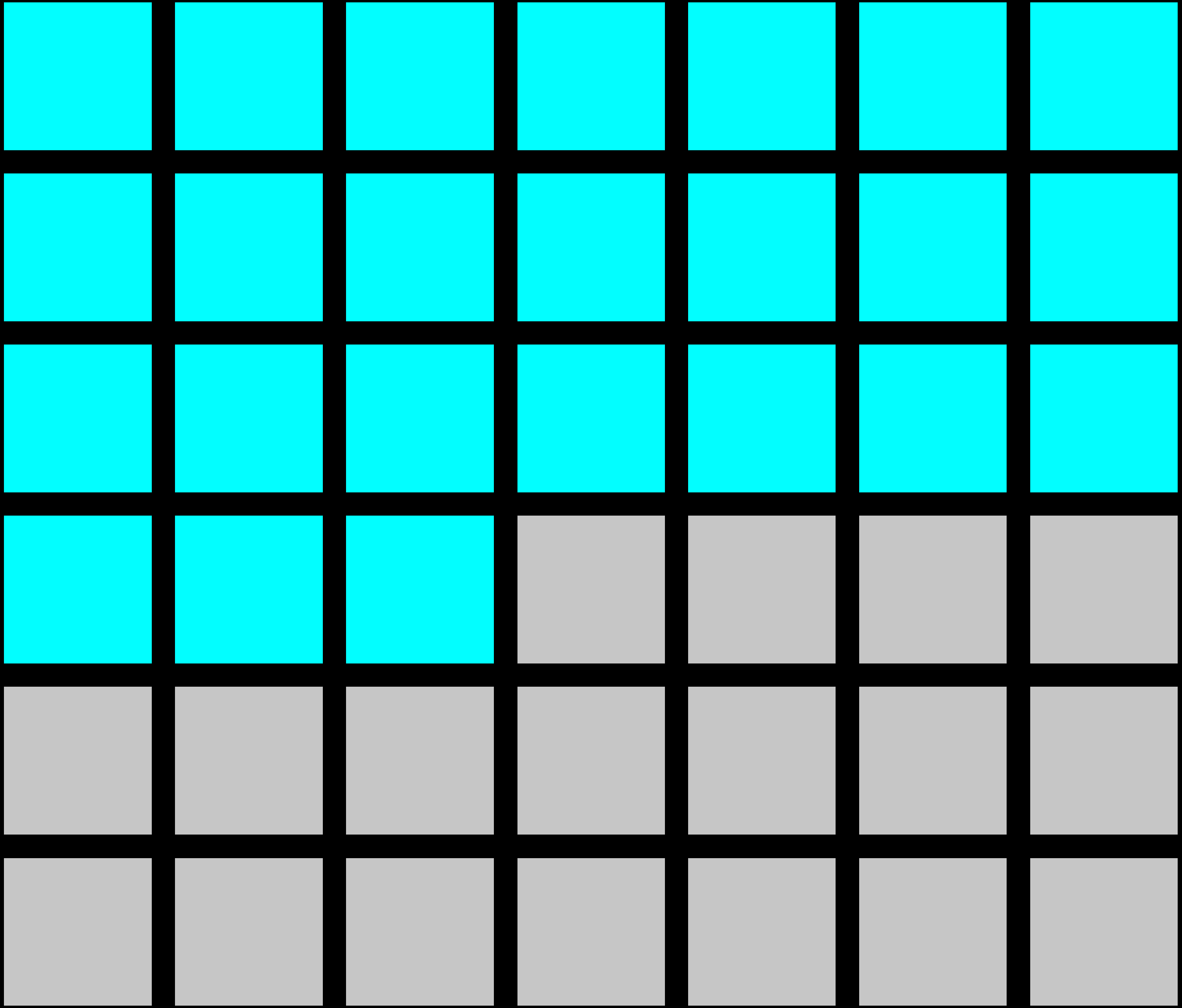


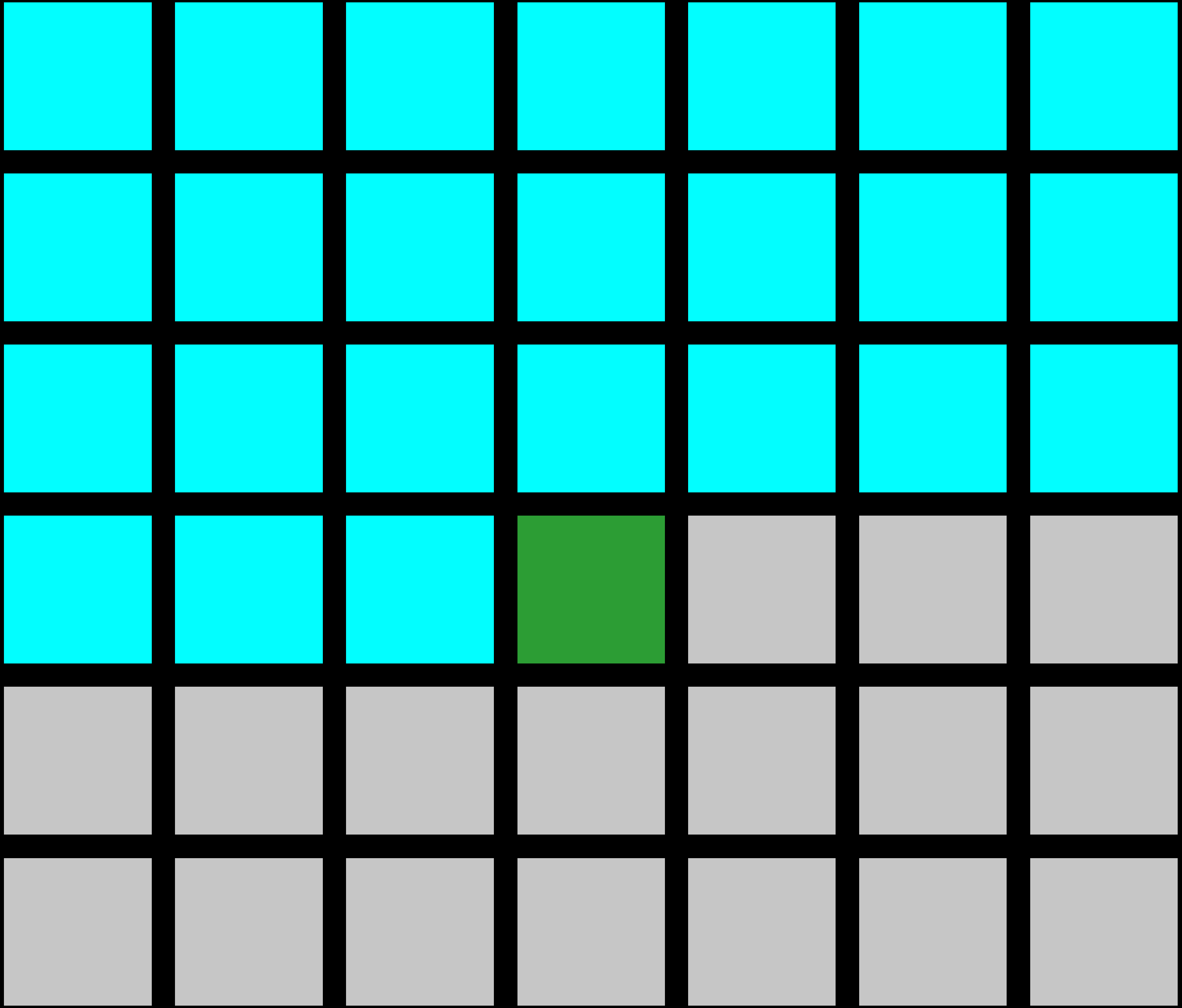


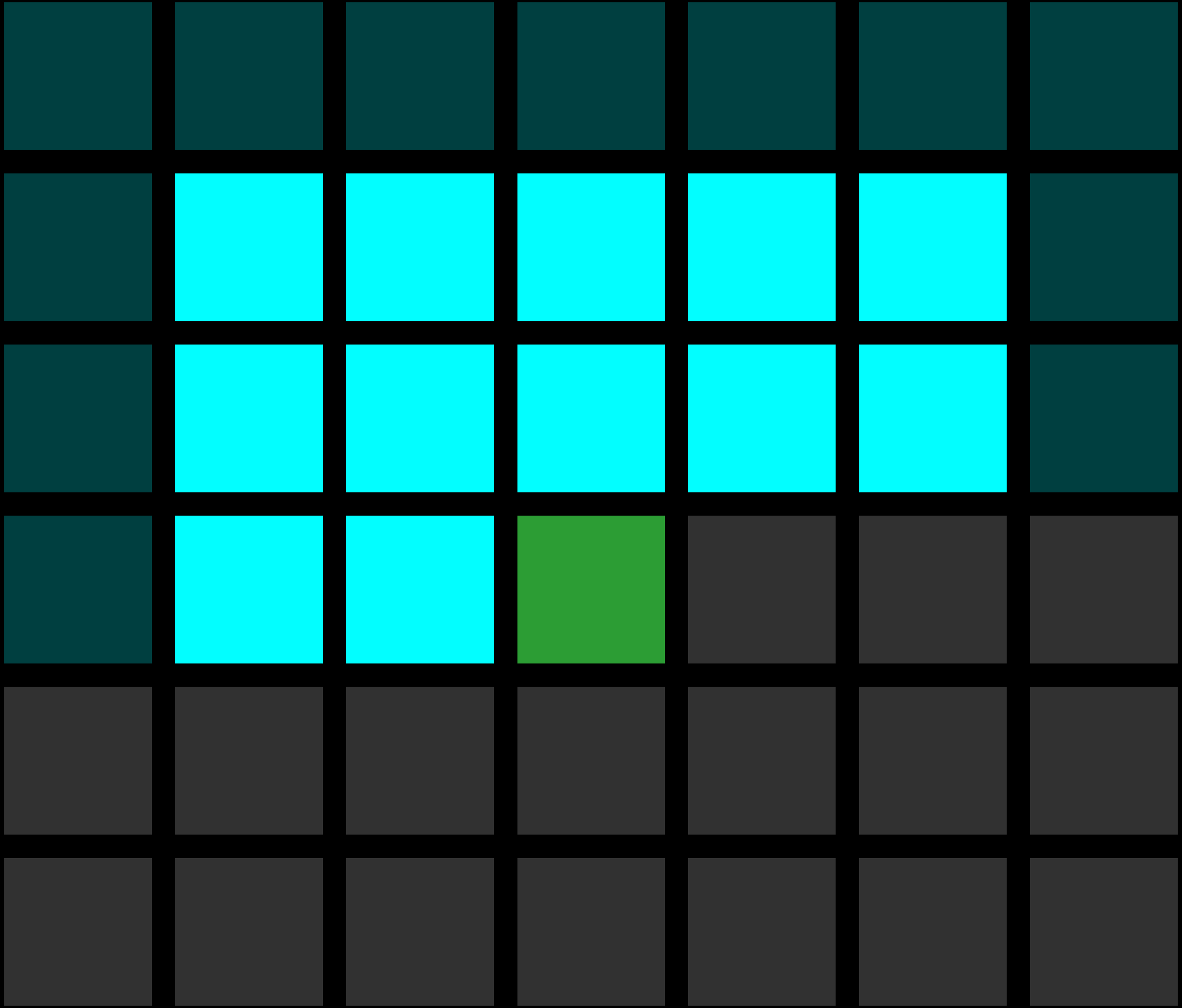


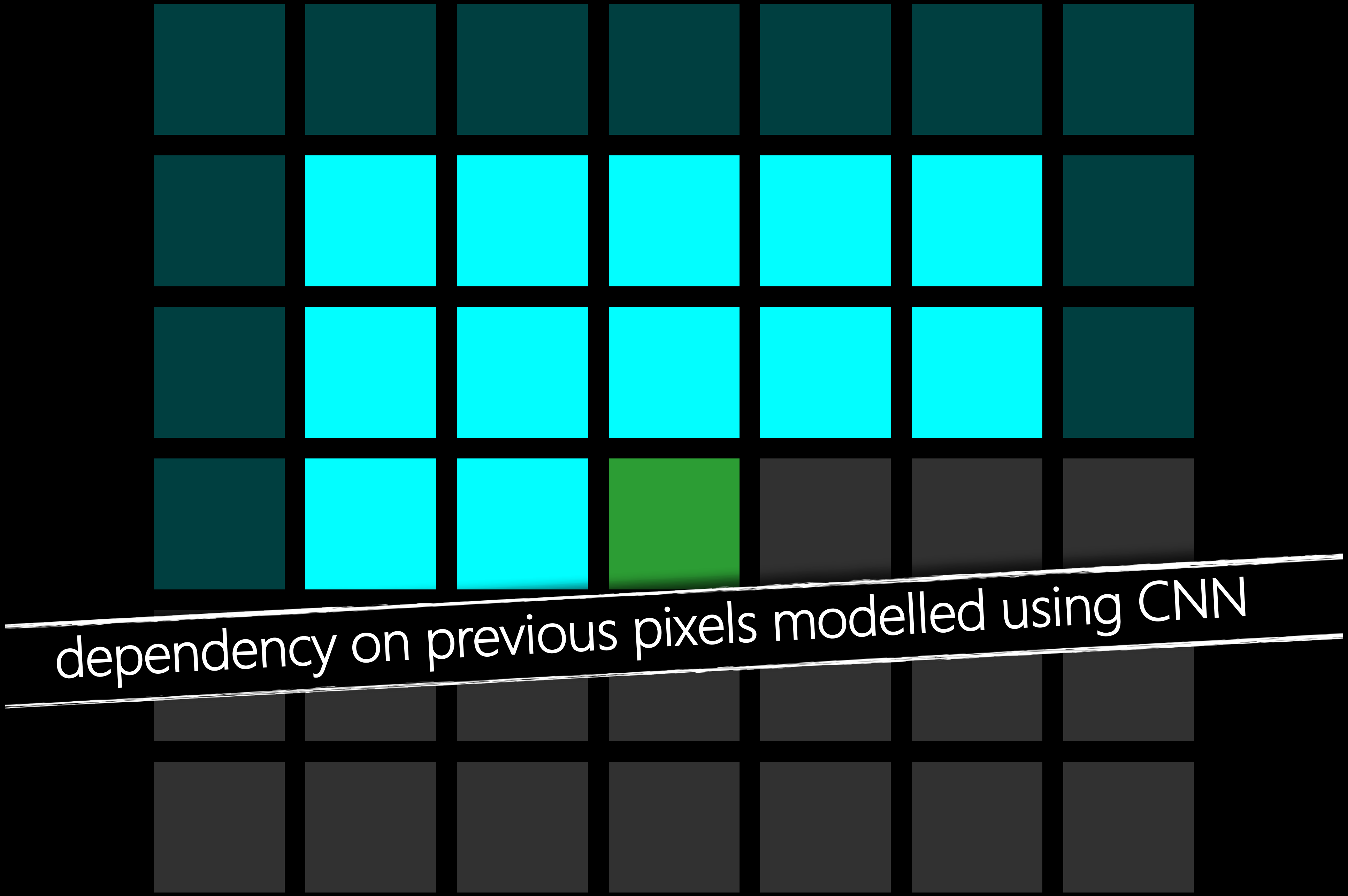
PixelCNN

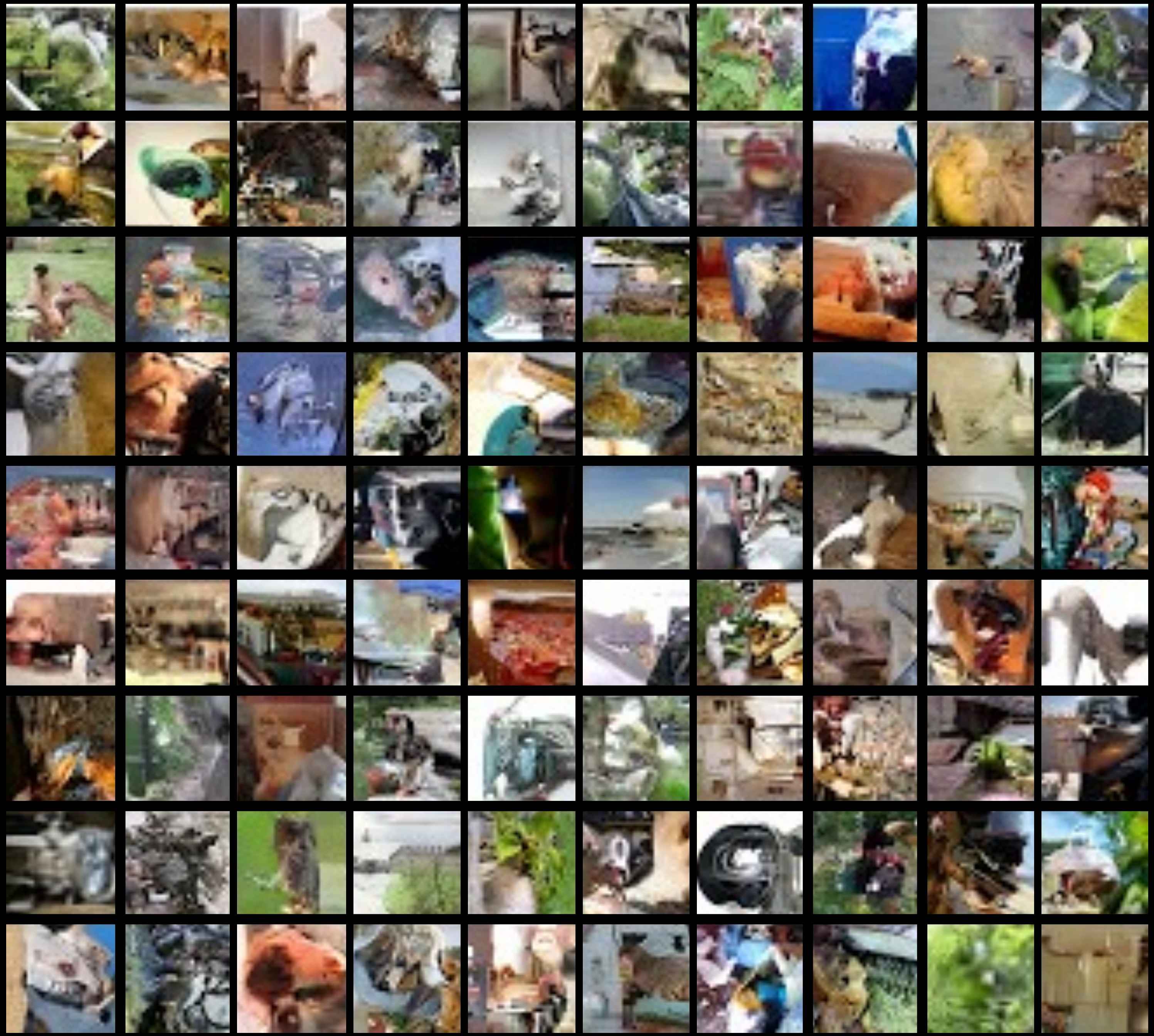


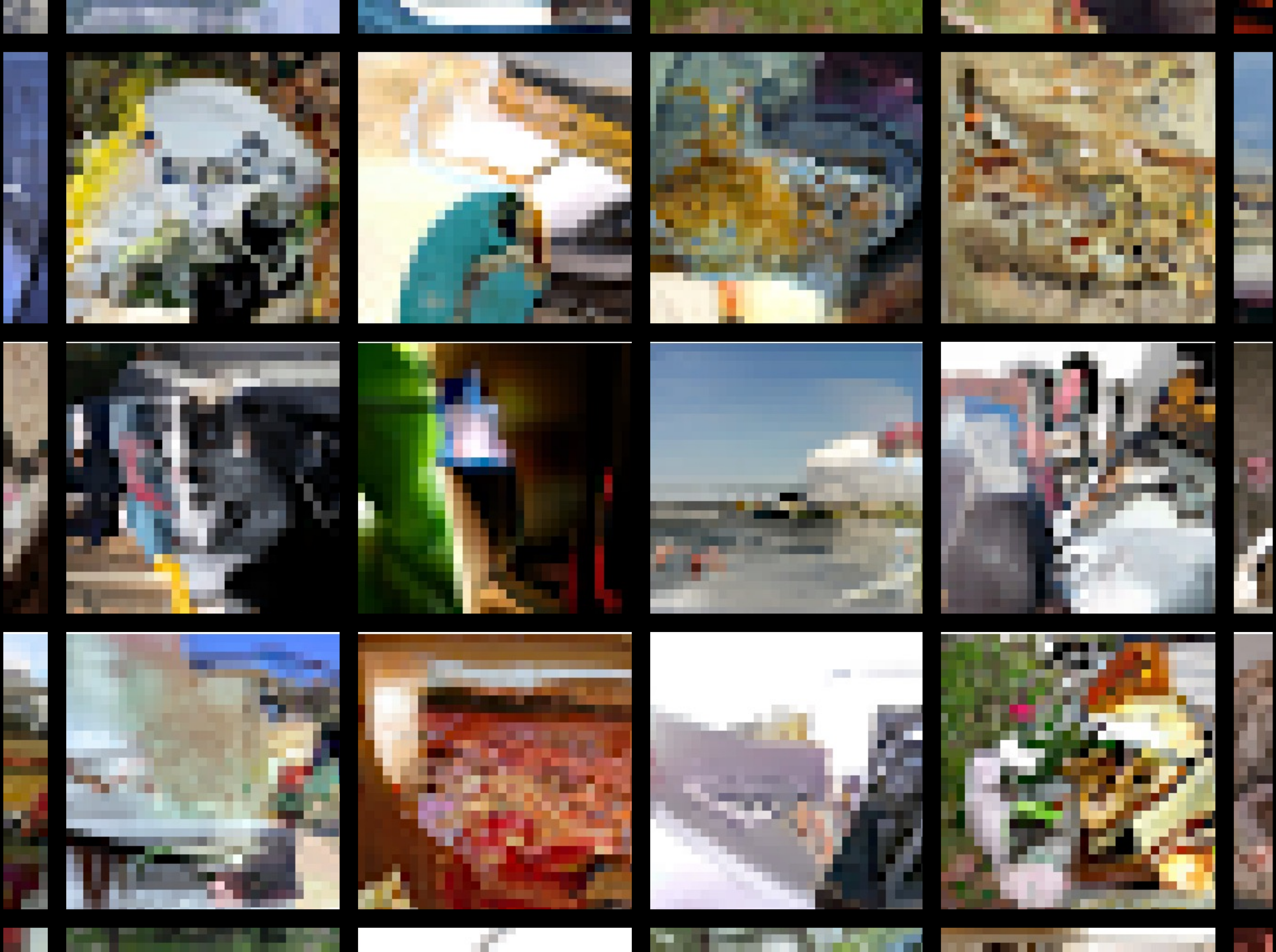




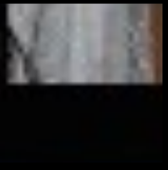


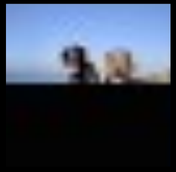

dependency on previous pixels modelled using CNN



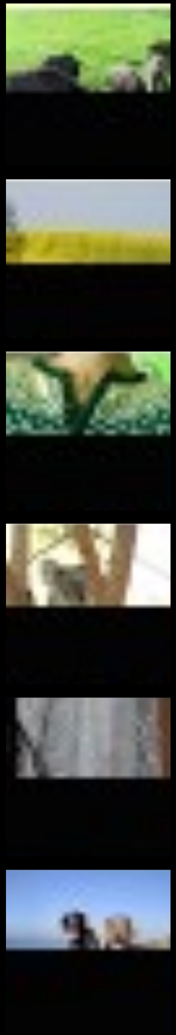


occluded



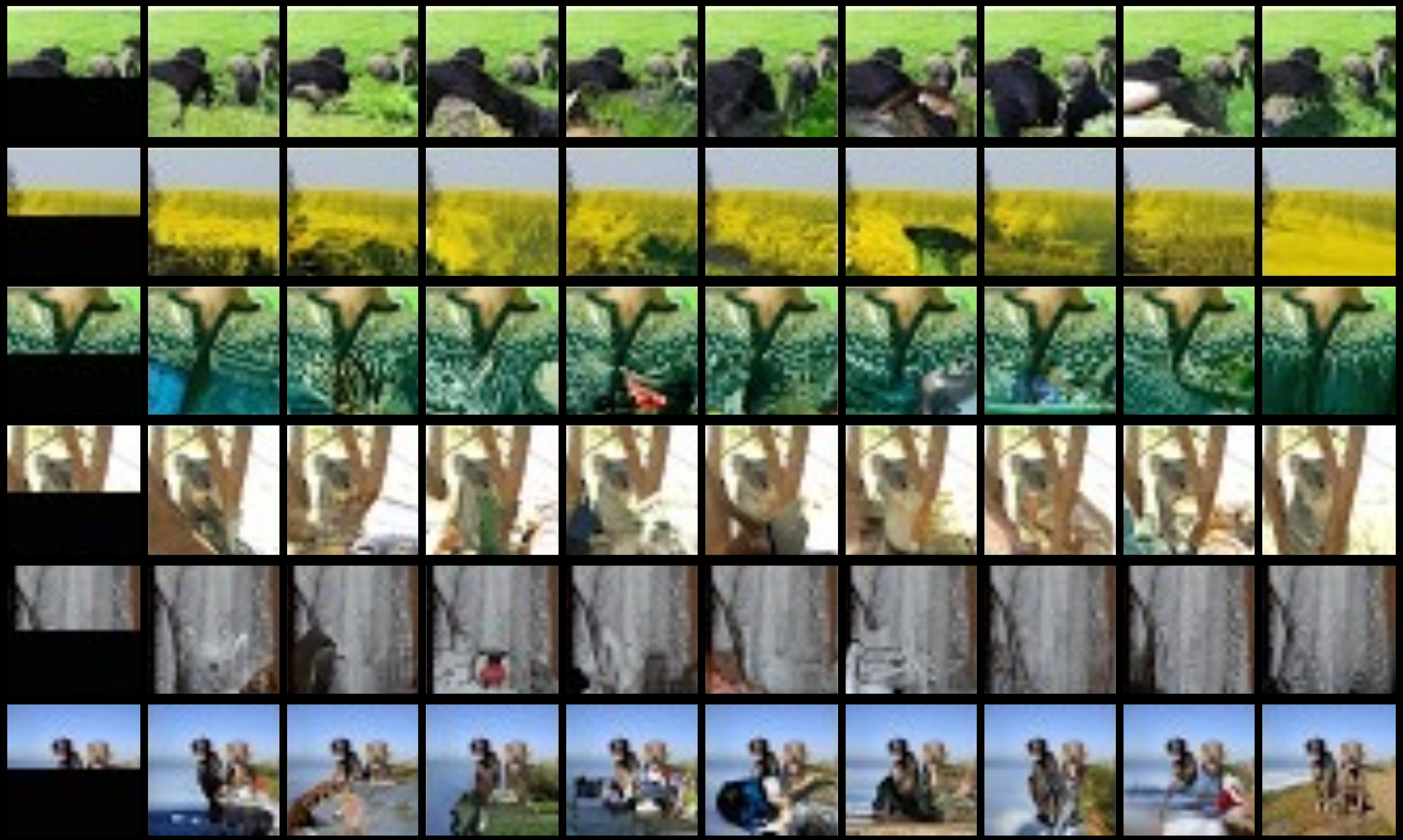


occluded



original

occluded completions original



sequential pixel generation is SLOW

pixel generation order matters

Explicit Density Estimation

Learn weights W

Goal: Write down an explicit function for $p(x) = f(x, \boxed{W})$

Given dataset $x^{(1)}, x^{(2)}, \dots x^{(N)}$, train the model by solving:

$$W^* = \arg\max_{W} \prod_i p(x^{(i)})$$

Maximize probability of training data
(Maximum likelihood estimation)

$$= \arg\max_{W} \sum_i \log p(x^{(i)})$$

Log trick to exchange product for sum

$$= \arg\max_{W} \sum_i \log f(x^{(i)}, W)$$

This will be our loss function!
Train with gradient descent

Explicit Density: Autoregressive Models

Goal: Write down an explicit function for $p(x) = f(x, W)$

Assume x consists of multiple subparts:

$$x = (x_1, x_2, x_3, \ldots, x_T\ )$$

Break down probability using the chain rule:

$$\begin{aligned} p(x) &= p(x_1, x_2, x_3, \ldots, x_T\ ) \\ &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots \\ &= \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}) \end{aligned}$$

Probability of the next subpart given all the previous subparts

$$\begin{tikzcd} p(x_1) & p(x_2) & p(x_3) & p(x_4) \\ h_1 \arrow[u] \arrow[r] & h_2 \arrow[u] \arrow[r] & h_3 \arrow[u] \arrow[r] & h_4 \arrow[u] \\ x_0 \arrow[u] & x_1 \arrow[u] & x_2 \arrow[u] & x_3 \arrow[u] \end{tikzcd}$$

We've already seen this! Language modeling with an RNN!

Slide Credits

- EECS 6322 Deep Learning for Computer Vision, Kosta Derpanis (York University)
- EECS 498 Deep Learning for Computer Vision, Justin Johnson (U. Michigan)
- Many amazing research papers!

Extra Reading:

https://towardsdatascience.com/auto-regressive-generative-models-pixelrnn-pixelcnn-32d192911173