Lecture 5: Machine learning

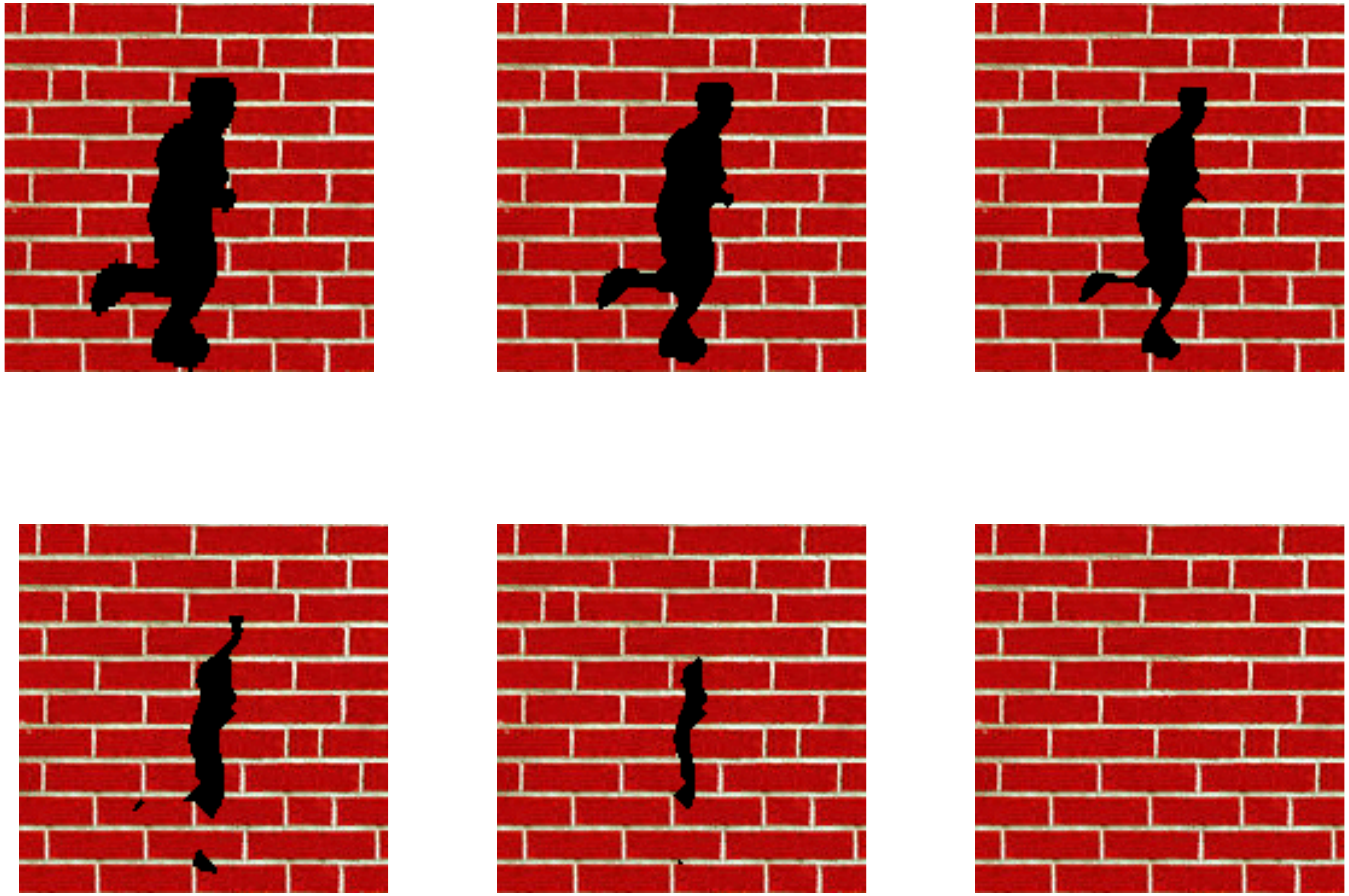
Announcements

- PS1 due tonight
- PS2 out tonight (by midnight).
  - Due 1 week from today!
- Szeliski 5.1
- Optional: Goodfellow Deep Learning “ML Basics

Announcements

- Late days → Late hours
  - You have 5×24 late hours total
  - We no longer "round up" (e.g., turning in 1 hour late doesn't take away a full day).
- Note that Piazza is a discussion platform, not just Q&A with course staff.
  - Thank you for those who have been answering questions!
- Course staff may delay answering Piazza questions and may send you to office hours for complex questions.

Last week: inpainting (hole filling)



Simple inpainting [Efros & Leung 1999]

Source: A. Efros

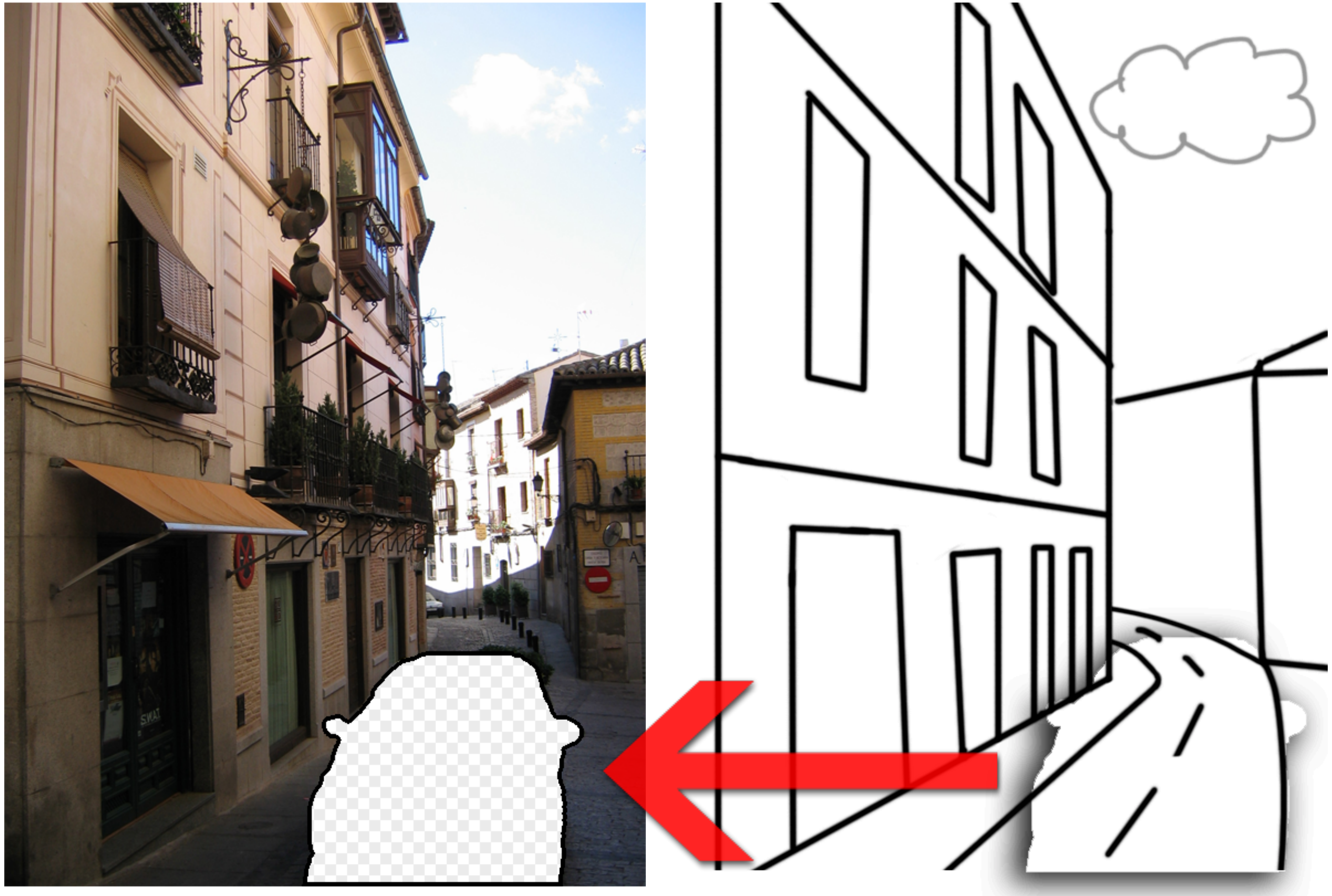
[Hays and Efros “Scene Completion Using Millions of Photographs”, 2007.]

Source: A. Efros

Simple inpainting [Efros & Leung 1999]

Source: A. Efros

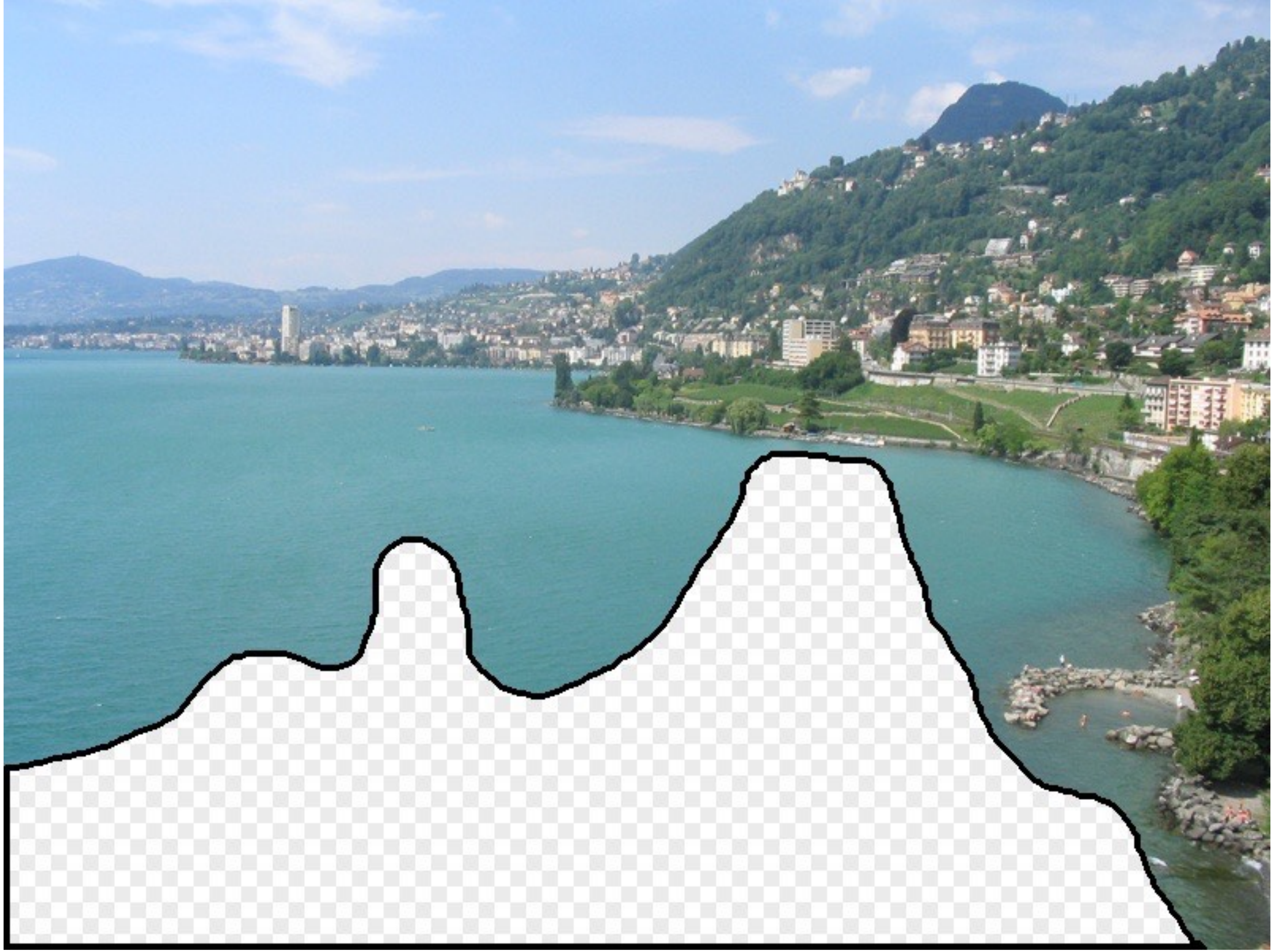
How would a human solve this?



Source: A. Efros

[Hays and Efros. Scene Completion Using Millions of Photographs. SIGGRAPH 2007.]

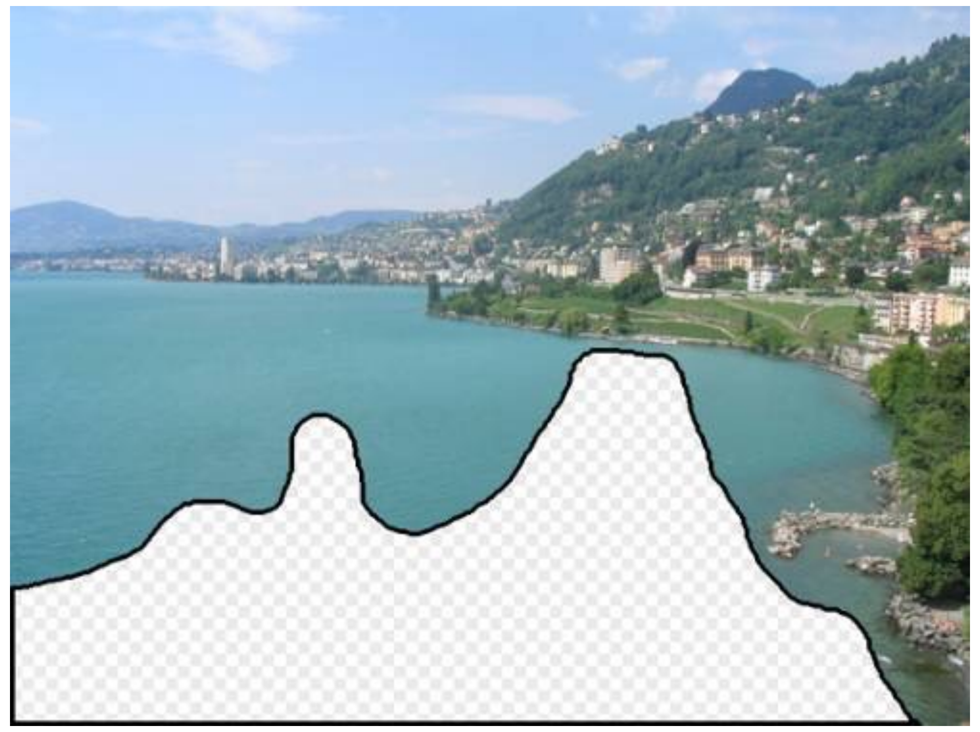
Source: A. Efros

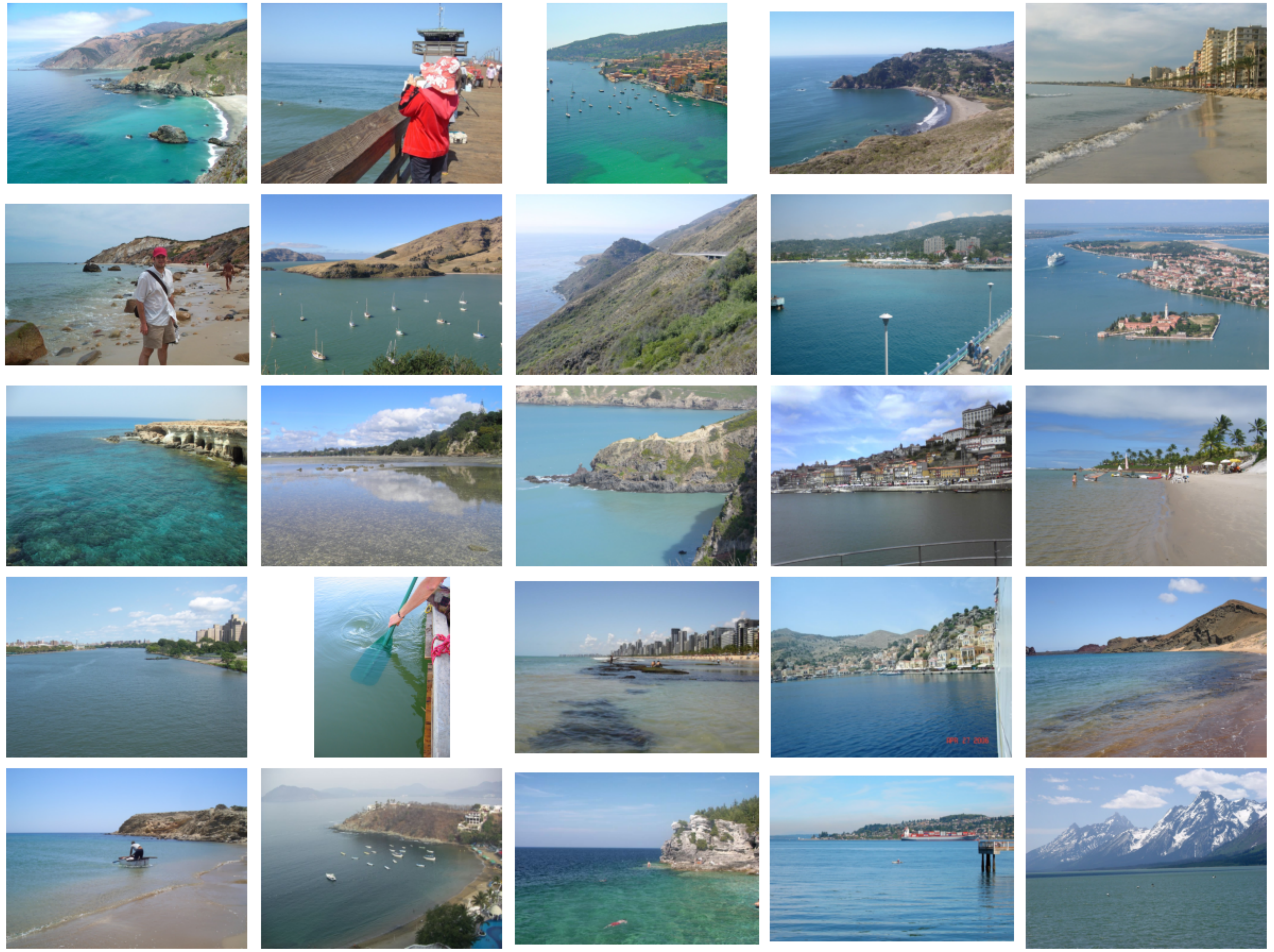


Source: A. Efros

2 Million Flickr Images

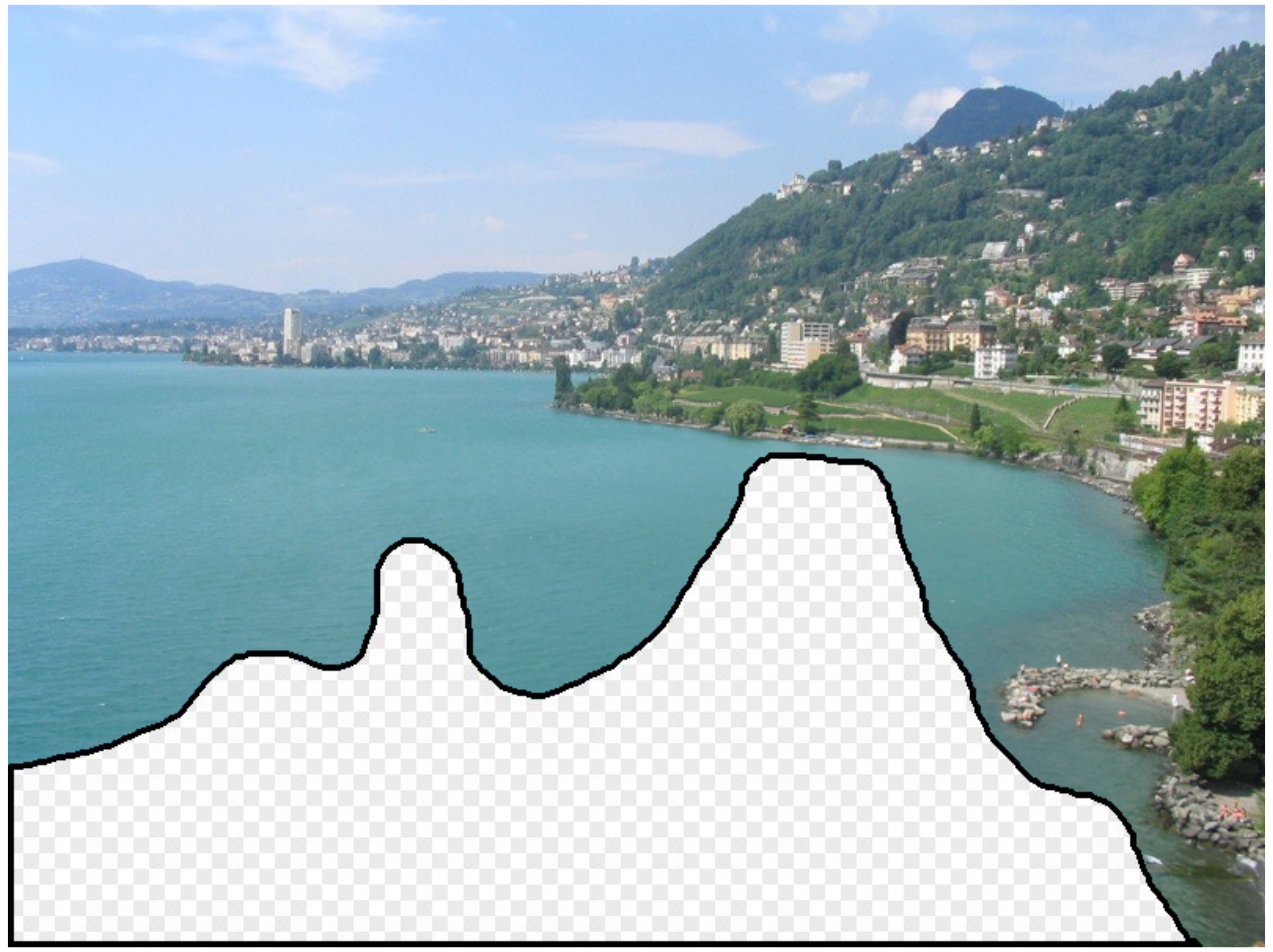
Source: A. Efros





… 200 total

Source: A. Efros



Source: A. Efros

Source: A. Efros

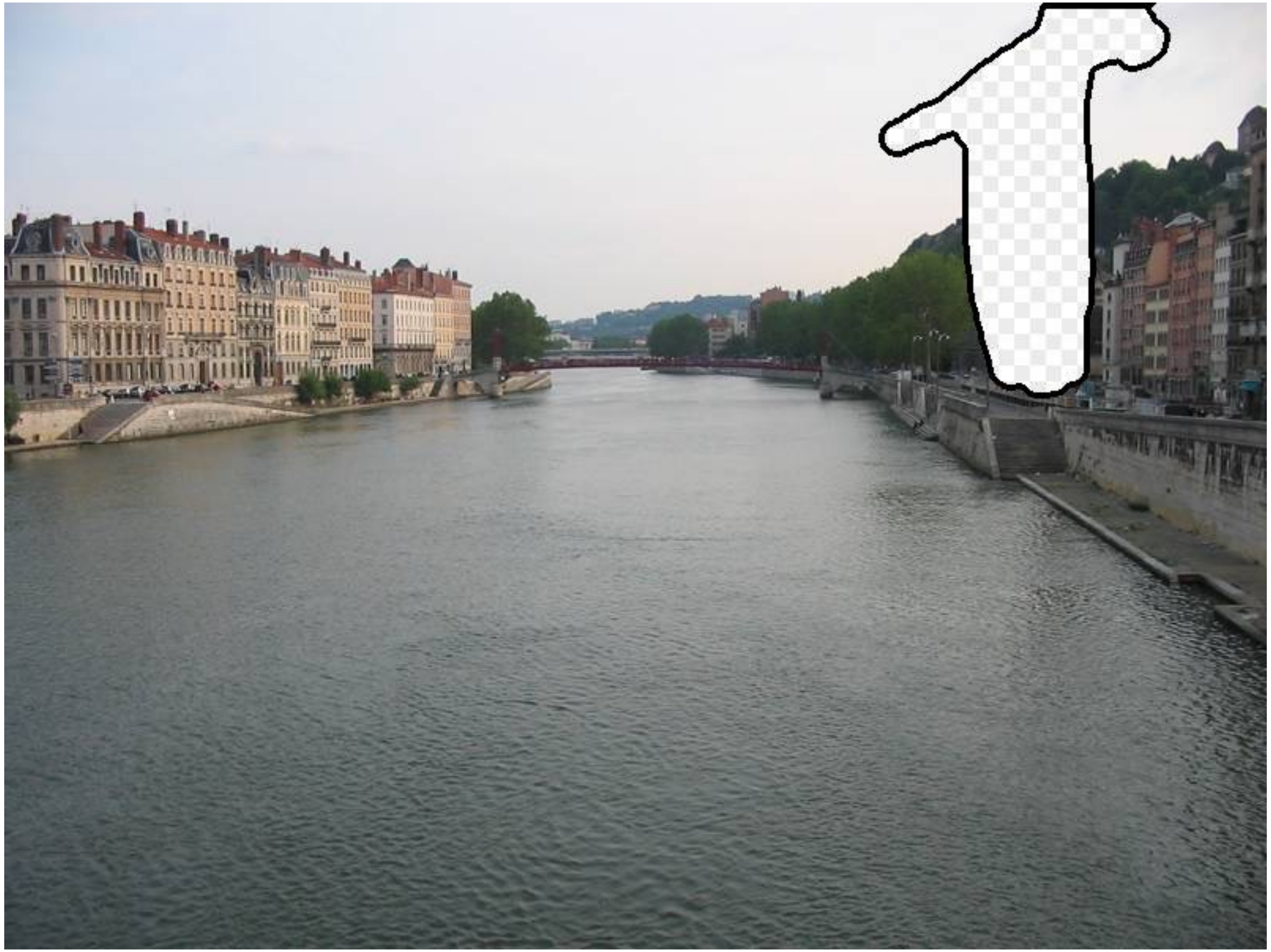
Source: A. Efros

Source: A. Efros

Source: A. Efros

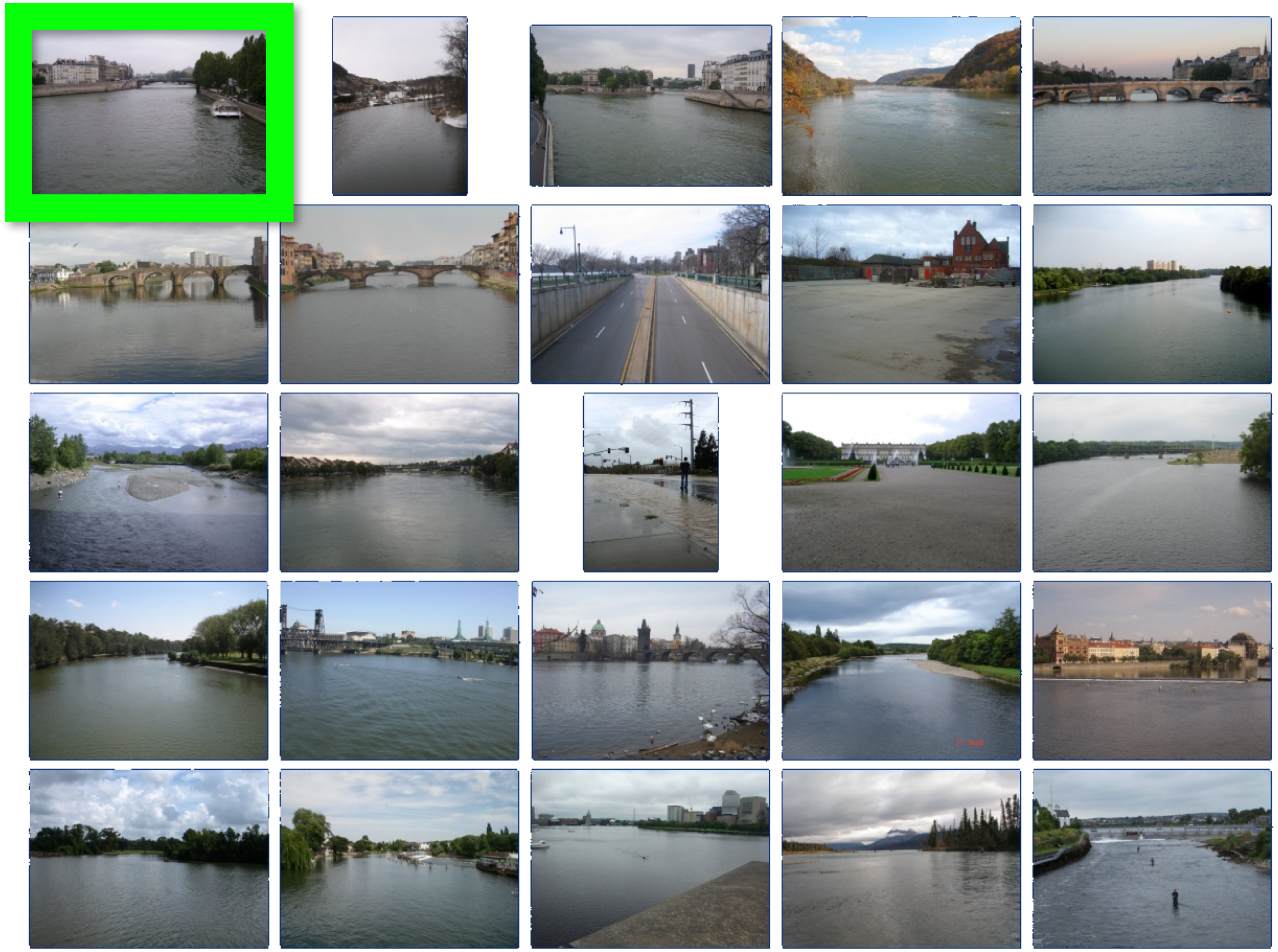
Source: A. Efros

Source: A. Efros



Source: A. Efros

Source: A. Efros



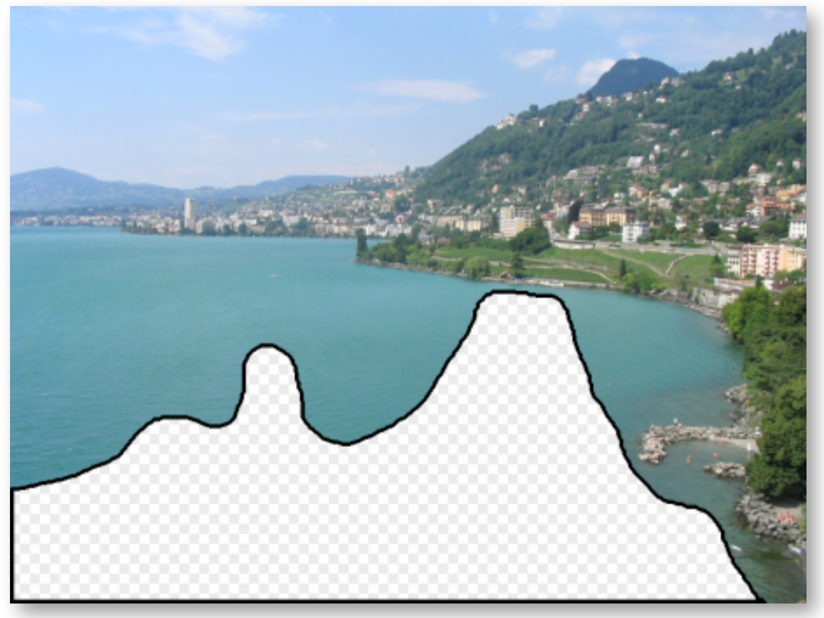
… 200 scene matches

Source: A. Efros

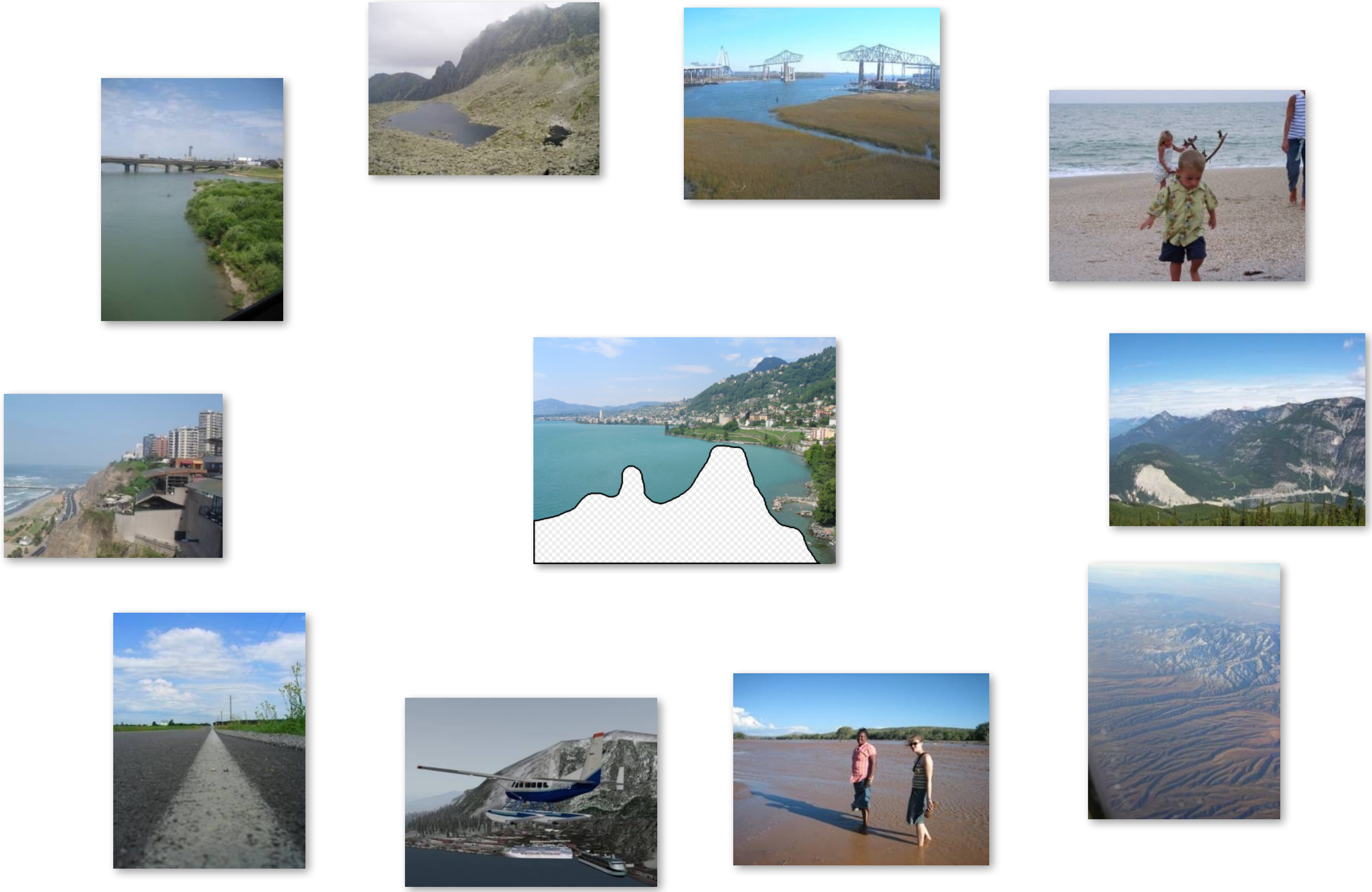
Source: A. Efros

Source: A. Efros

Why does this work?

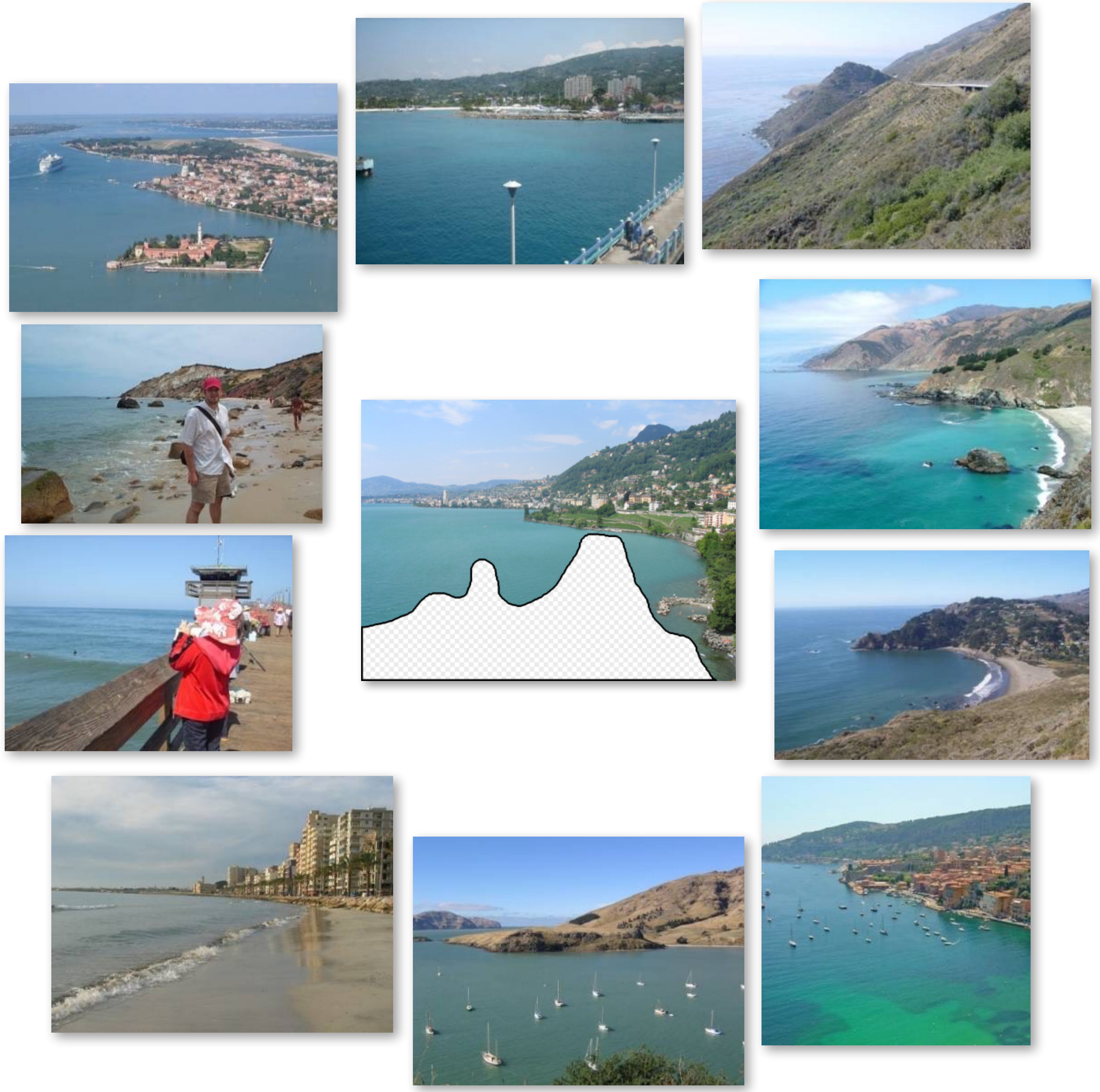


Source: A. Efros



Nearest neighbors from a collection of 20 thousand images

Source: A. Efros



Nearest neighbors from a
collection of 2 million images

Source: A. Efros

“Unreasonable Effectiveness of Data”

[Halevy, Norvig, Pereira 2009]

Parts of our world can be explained by elegant mathematics

physics, chemistry, astronomy, etc.

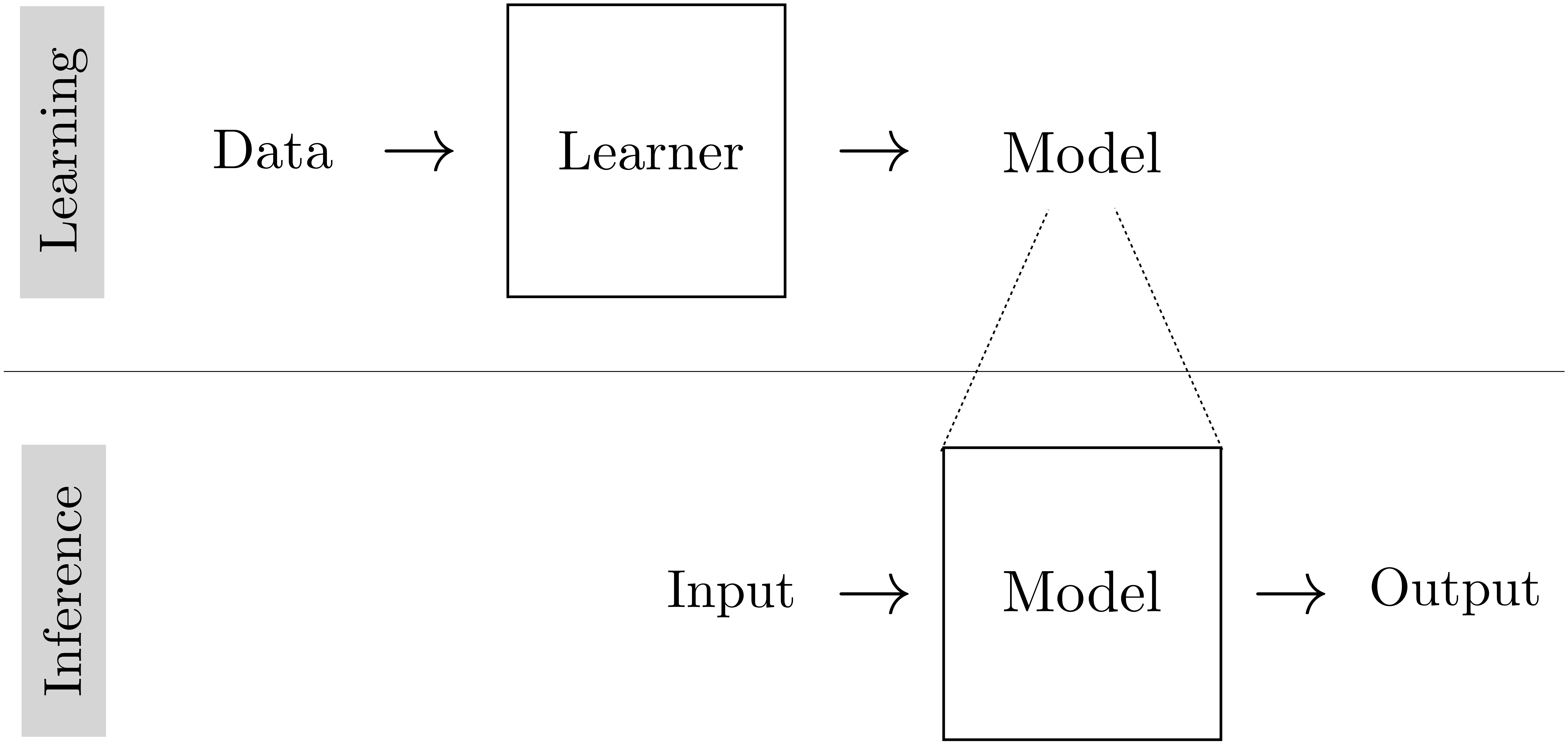
But much cannot

psychology, economics, genetics, etc.

Enter <u>The Data!</u>

> “For many tasks, once we have a billion or so examples, we essentially have a closed set that represents (or at least approximates) what we need…”

Source: A. Efros



Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

$$\{x_1, y_1\}$$
$$\{x_2, y_2\} \quad \rightarrow \quad \boxed{\text{Learner}} \quad \rightarrow \quad f : X \rightarrow Y$$
$$\{x_3, y_3\}$$
$$\ldots$$

Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

X Y

$\{$  , "bowling alley" $\}$

$\{$  , "desert" $\}$

$\vdots$

$\rightarrow$ Learner $\rightarrow$ $f : X \rightarrow Y$

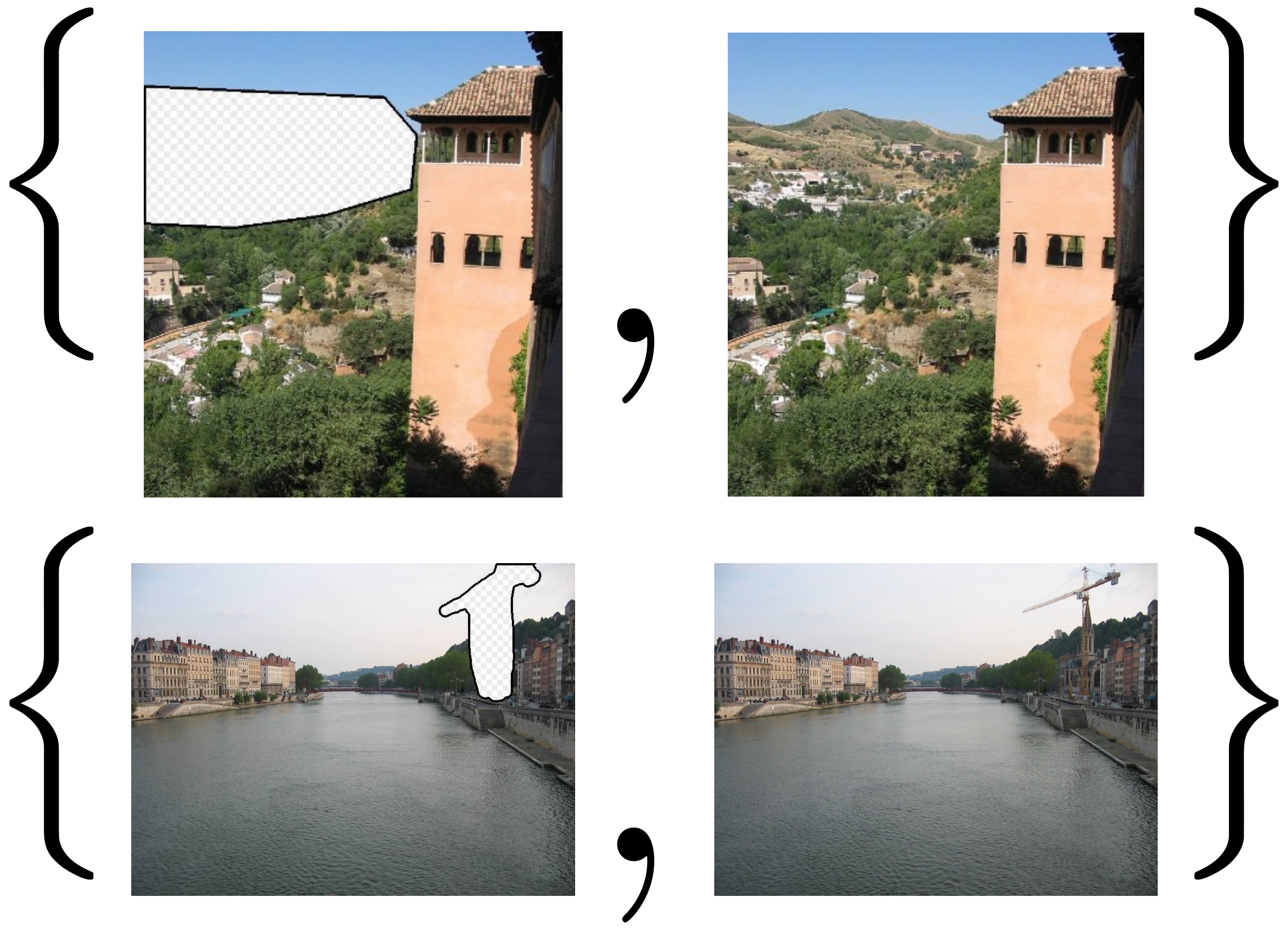
Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

X $\quad$ Y



$$\left\{ X, Y \right\} \;\; \vdots \;\; \rightarrow \boxed{\text{Learner}} \rightarrow \quad f : X \rightarrow Y$$

Source: Isola, Torralba, Freeman

Case study #1: Nearest neighbor

Nearest neighbor

Input feature vector $\mathbf{x}_q$

Input image

X Y

{ , "bowling alley" }

{ , "beach" }

⋮

{ , "desert" }

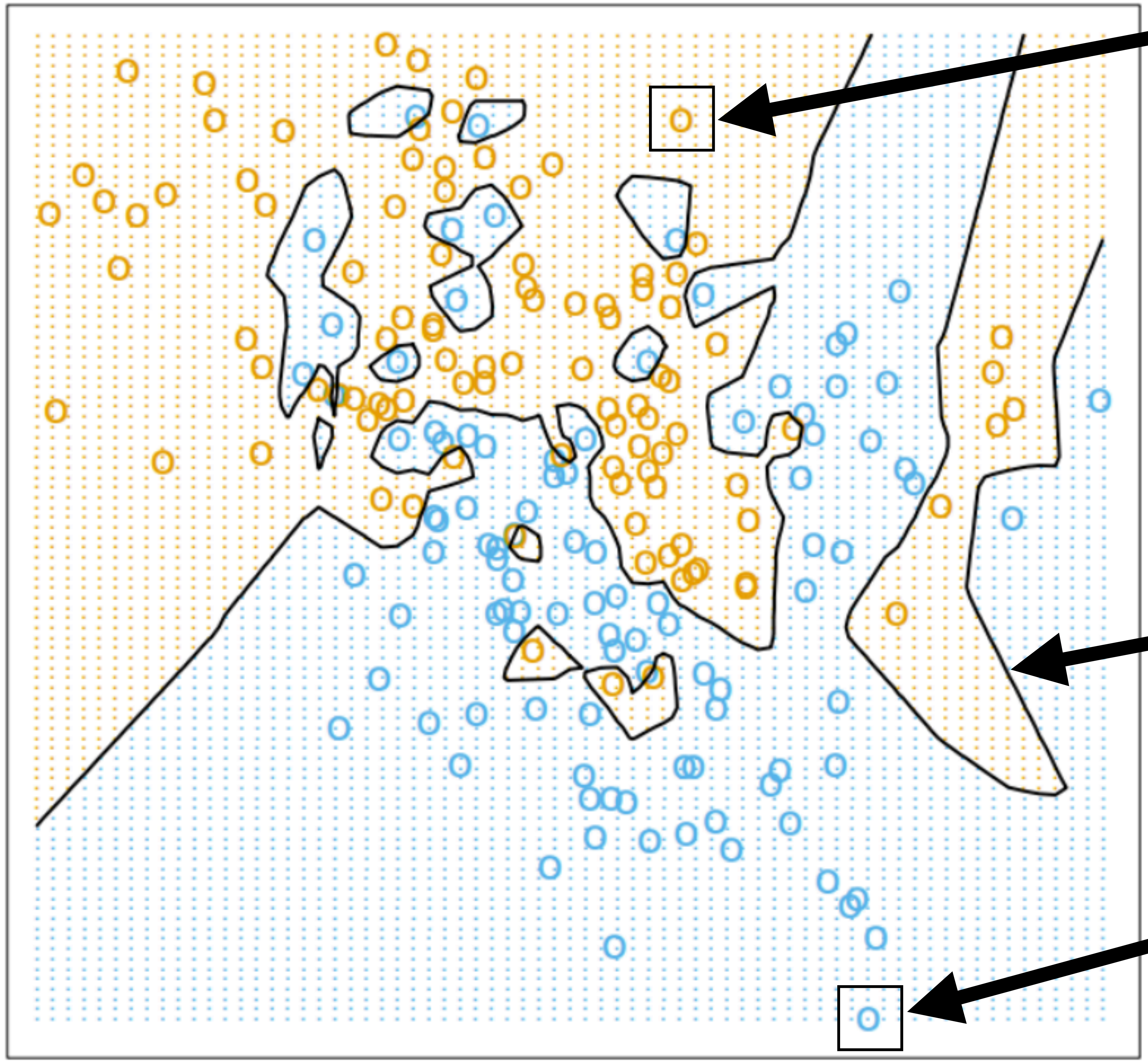
Take label from:
$\text{argmin}_i \; ||\mathbf{x}_q - \mathbf{x}_i||^2$

K-nearest neighbor:
Take most common label in k closest examples

example from orange class

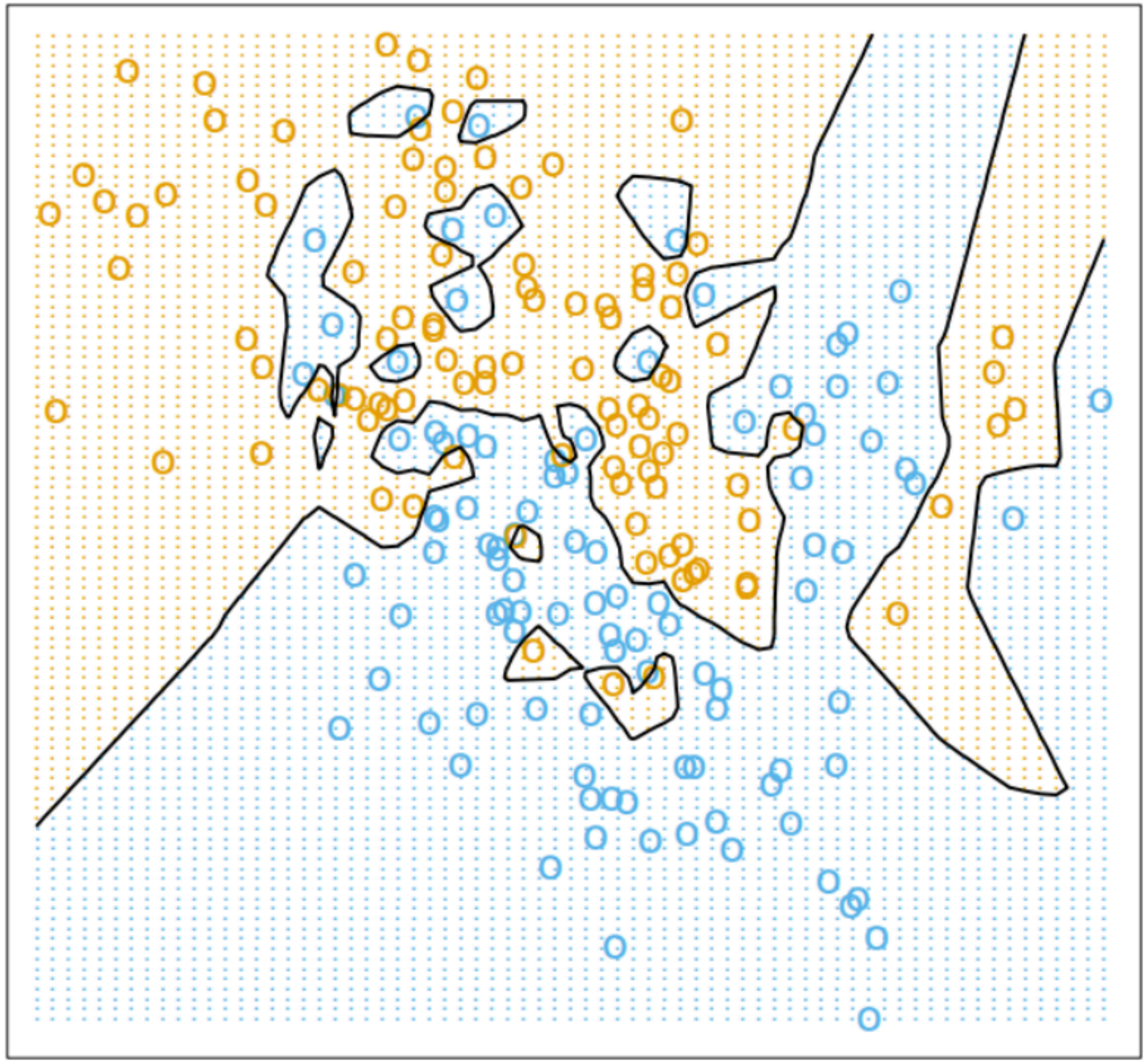
a boundary between
blue/orange classifier predictions

example from blue class

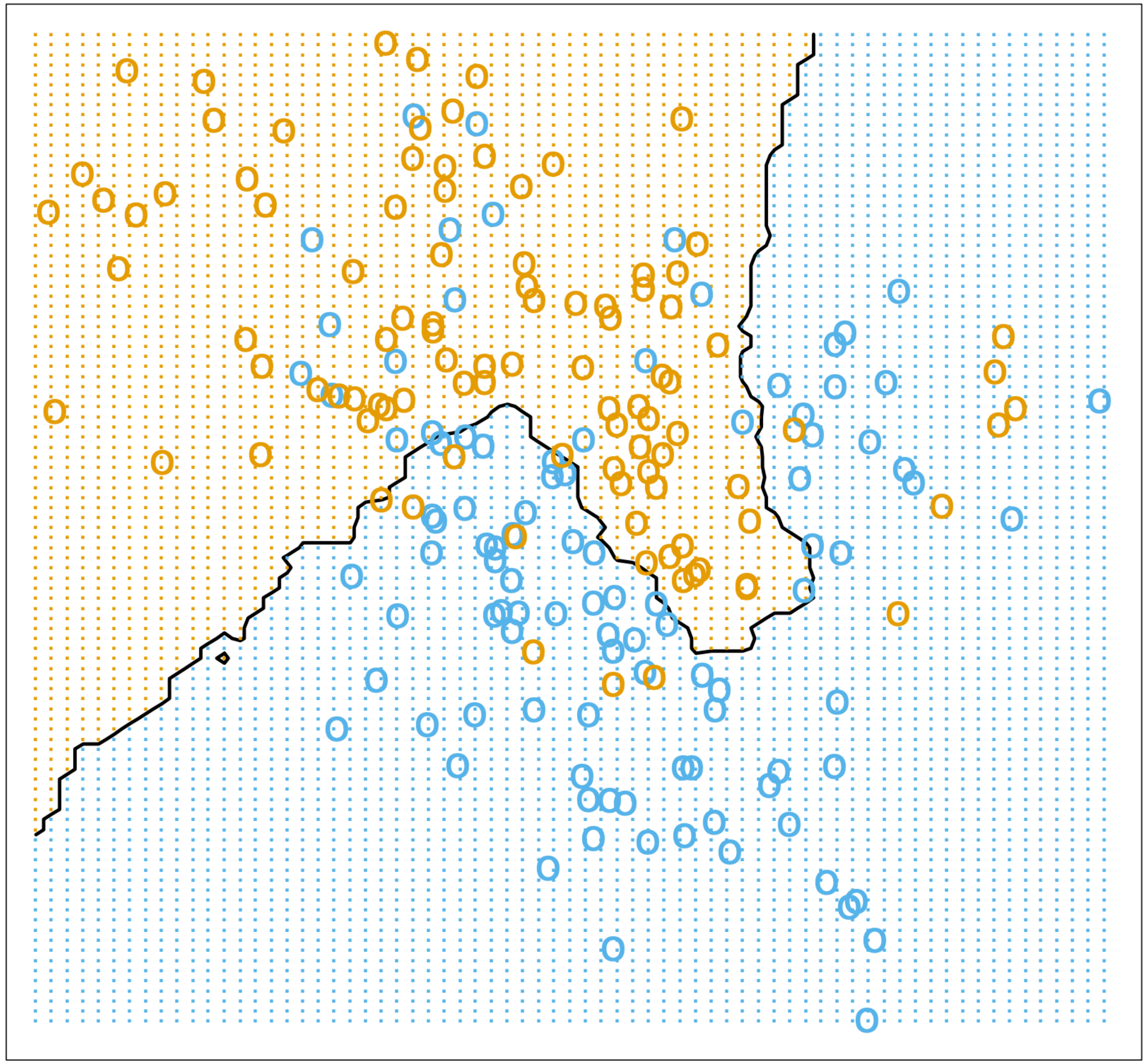


$k = 1$

Source: [Hastie et al. “Elements of Statistical Learning”, 2001]



$k = 1$

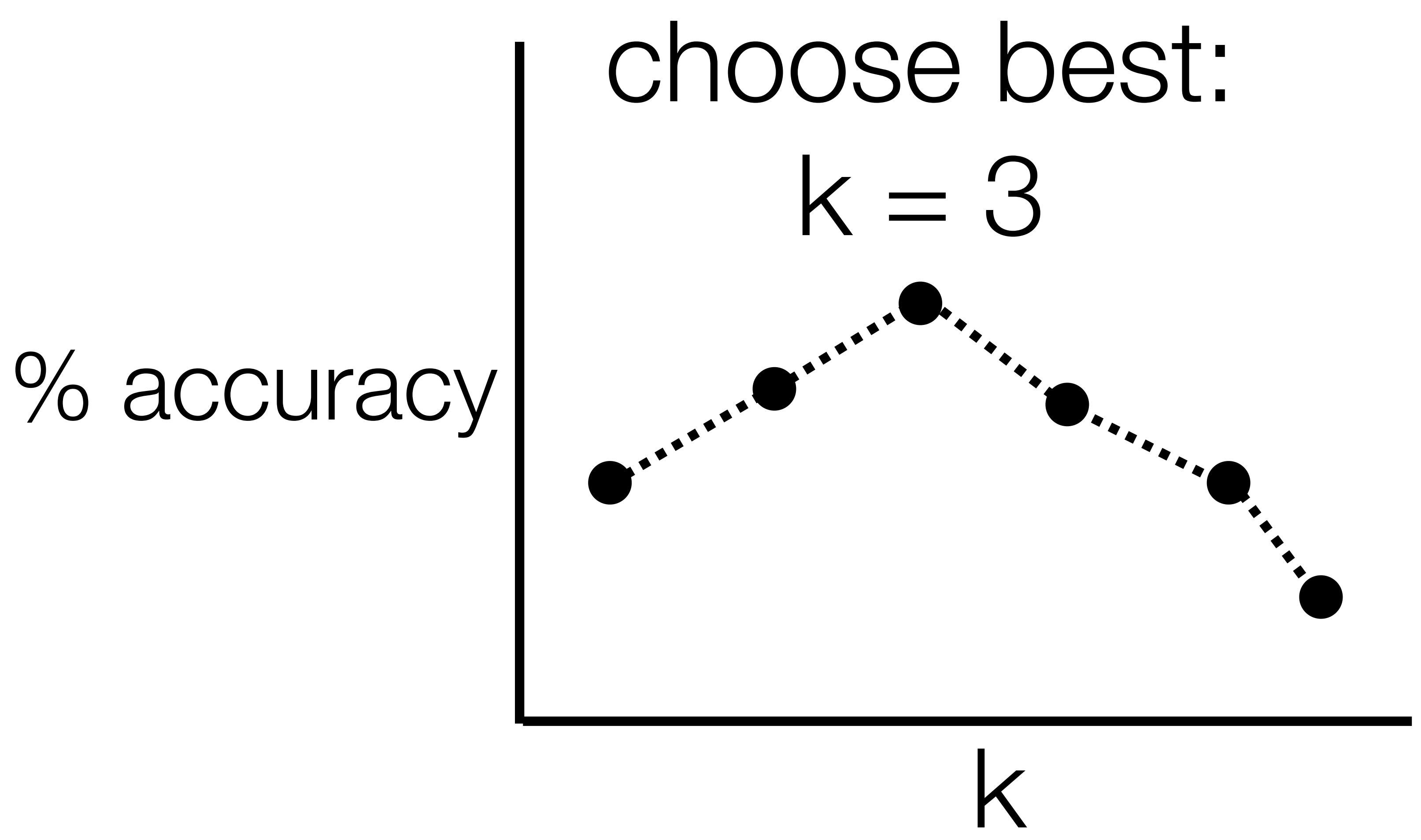


$k = 15$

Source: [Hastie et al. “Elements of Statistical Learning”, 2001]

How do we choose k?

How do we choose k?



Pool of nearest neighbors

Training Set	Validation Set	Test Set

Measures generalization.
Ideally only test once at end!

Choose **hyperparameters** like k, feature space, similarity function, etc.

Can work well when datasets are very big!

Input image

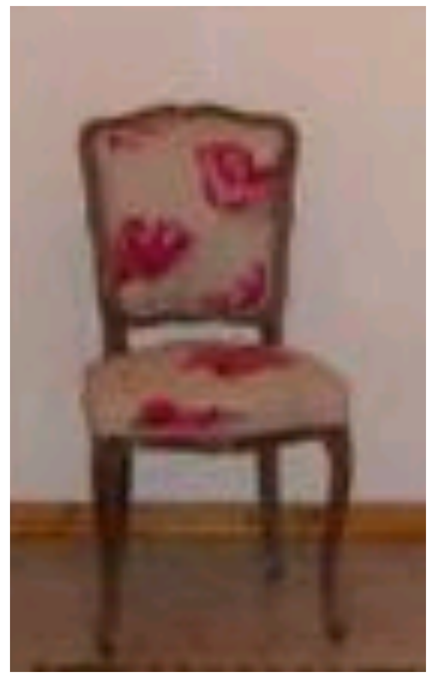
Colorized output

Nearest Neighbors

[Torralba et al. “80 million tiny images”, 2008]

How do you represent the image?

Idea: L2 distance between pixels

$\|$  $-$  $\|_2$

Spot the difference

Not Blurred

Blurred

Source: Daniel Geng

Spot the difference

Source: Daniel Geng

Spot the difference

Shifted

Source: Daniel Geng

Spot the difference

Original

L2 distance: 51.1

Shifted

L2 distance: 51.6

Blurred

Almost the same distance!

Source: Daniel Geng

How do you represent the image?

Input image

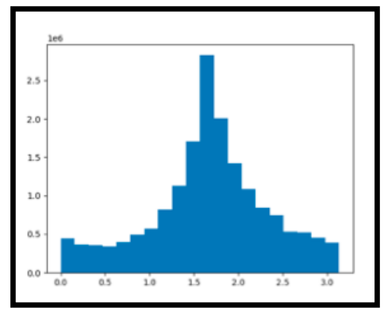
$$\longrightarrow \begin{bmatrix} 0.5 \\ 0.1 \\ -0.4 \\ \cdots \\ 0.9 \end{bmatrix}$$

Feature vector x_q

Some feature ideas:

tiny image

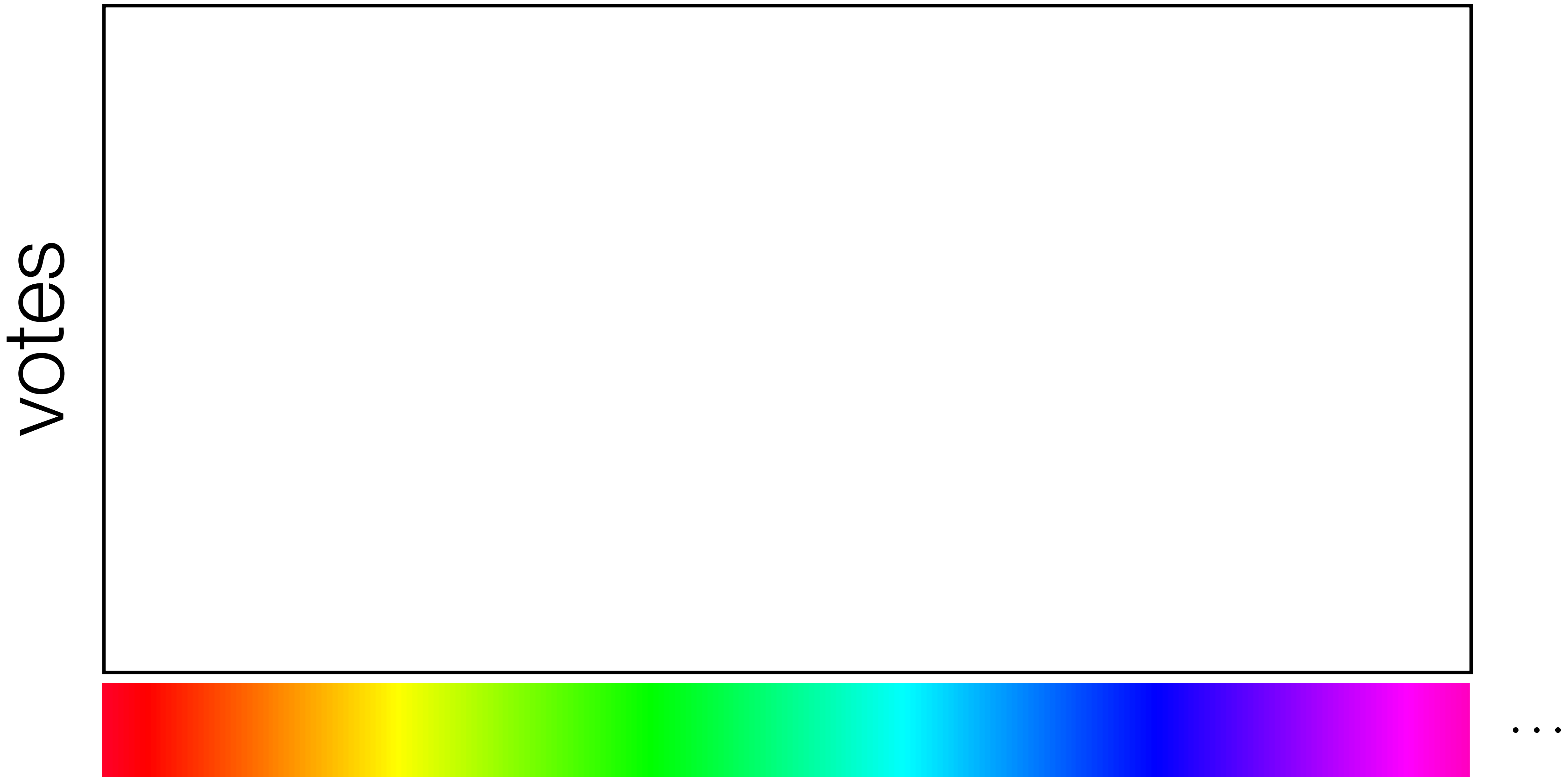
normalized tiny image $\dfrac{x - \mu}{\sigma}$



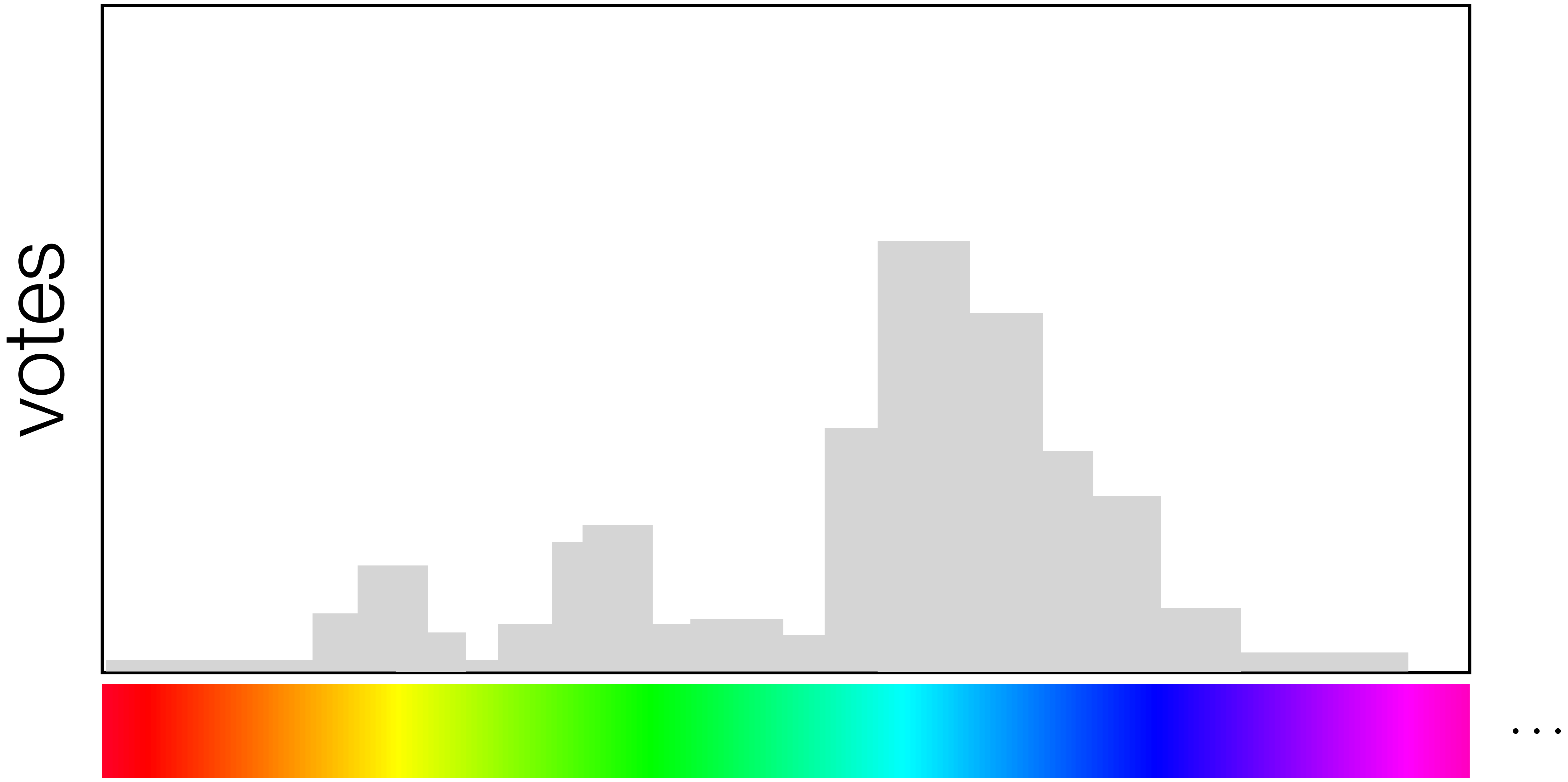
θ

edge orientation histogram

Color histograms

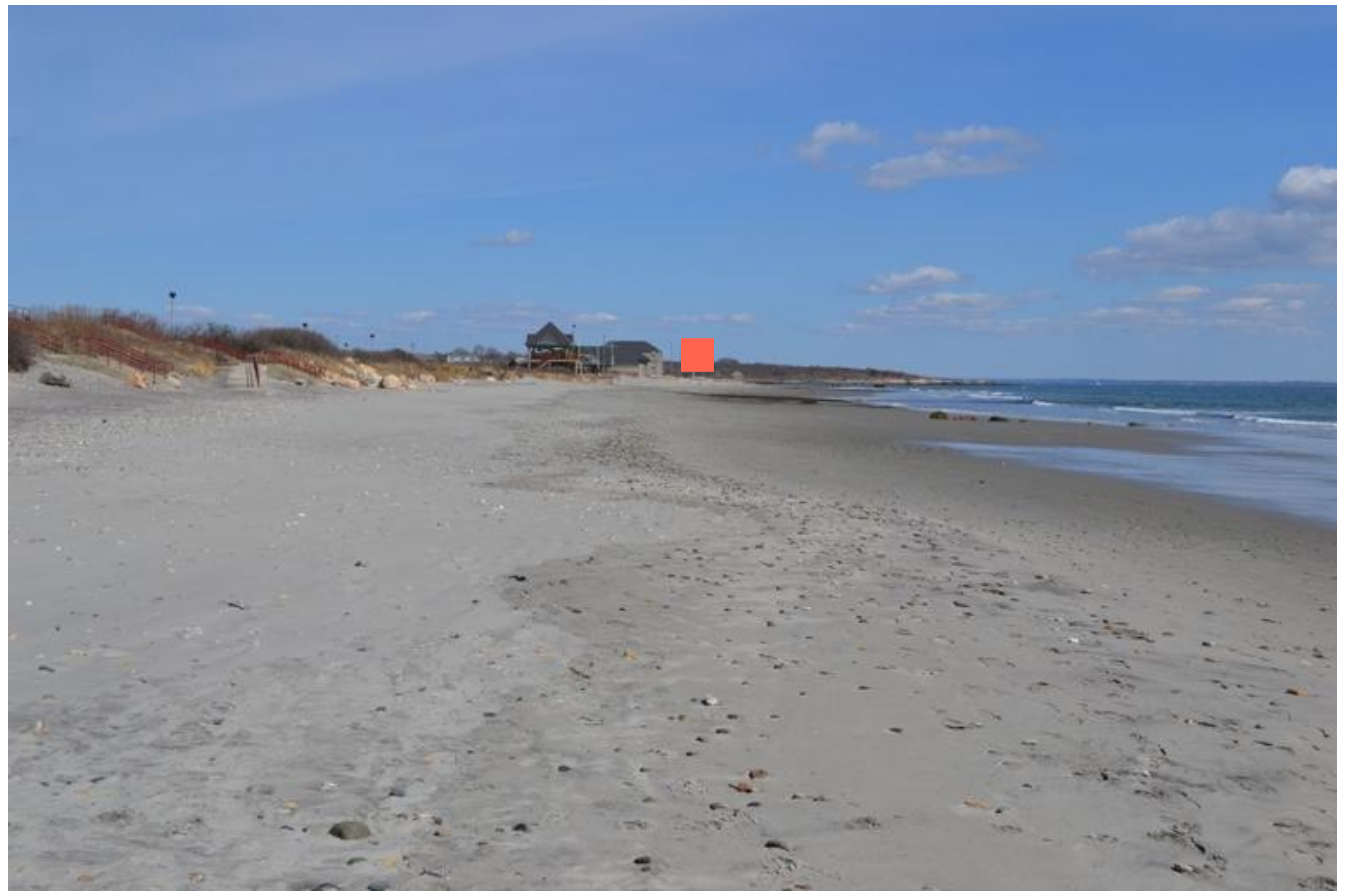


Color histograms



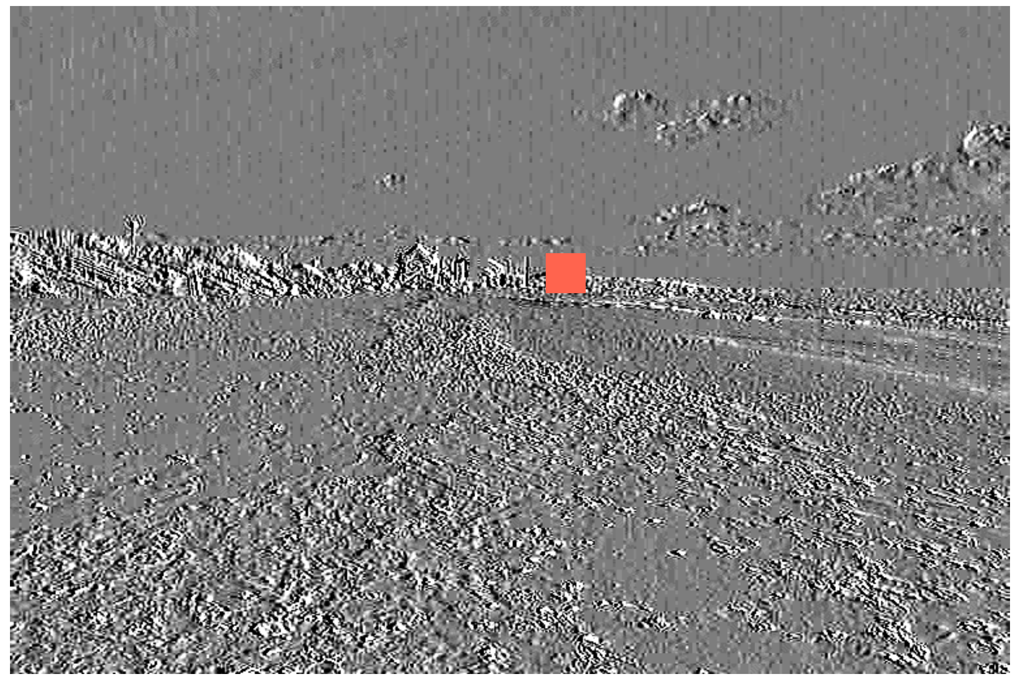
Typically create a coarse grid over the color channels, such as $10 \times 10 \times 10$.

Edge orientation histograms

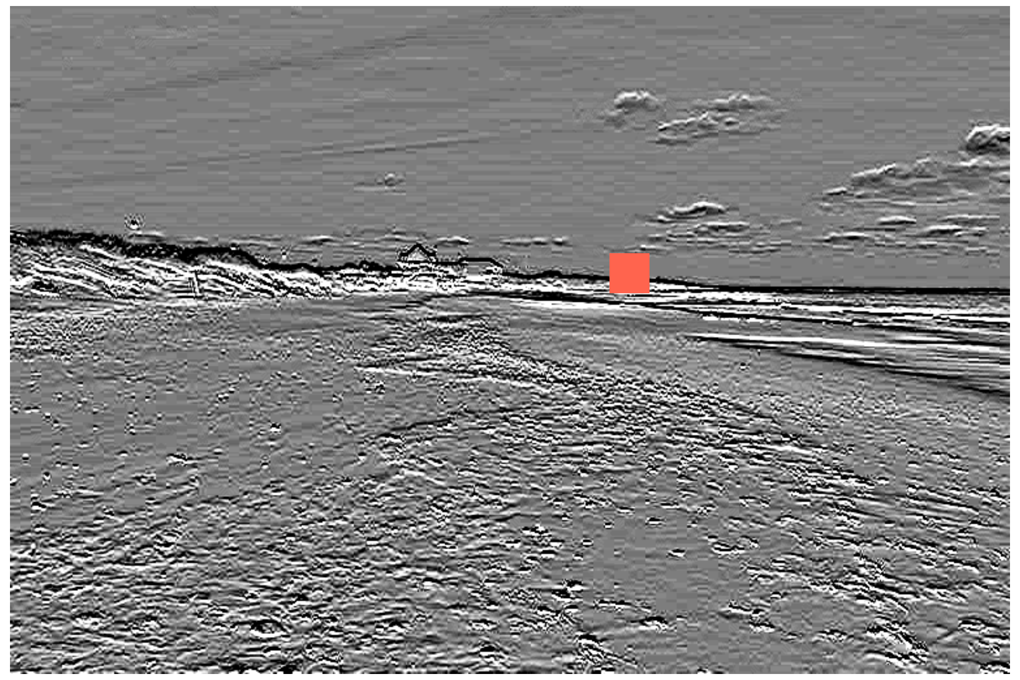


Input image

magnitude

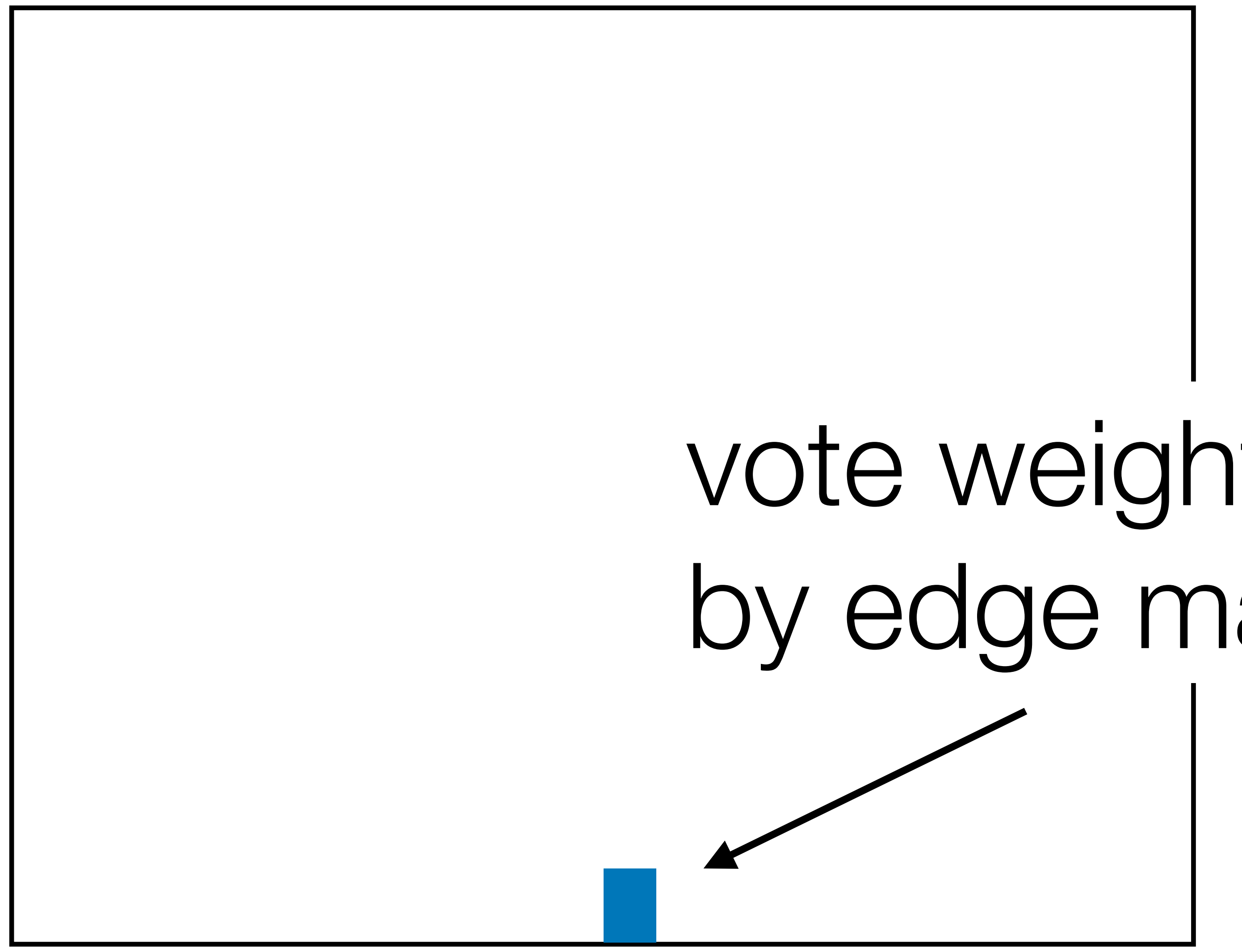


dx



dy

θ = tan^{-1}(dy/dx)



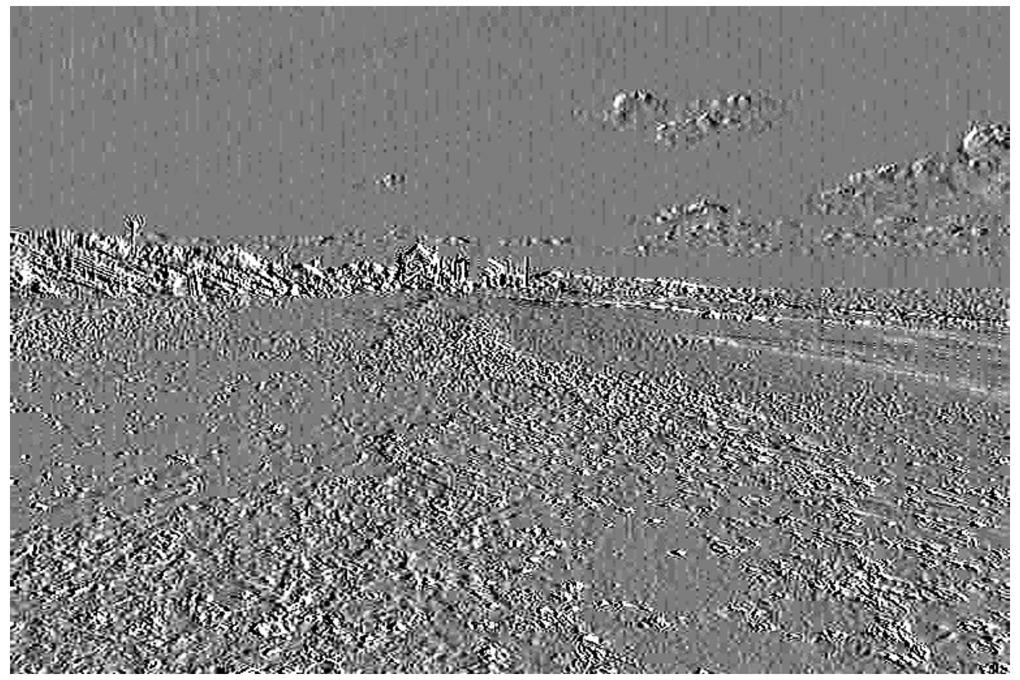
vote weighted
by edge magnitude

0 θ π

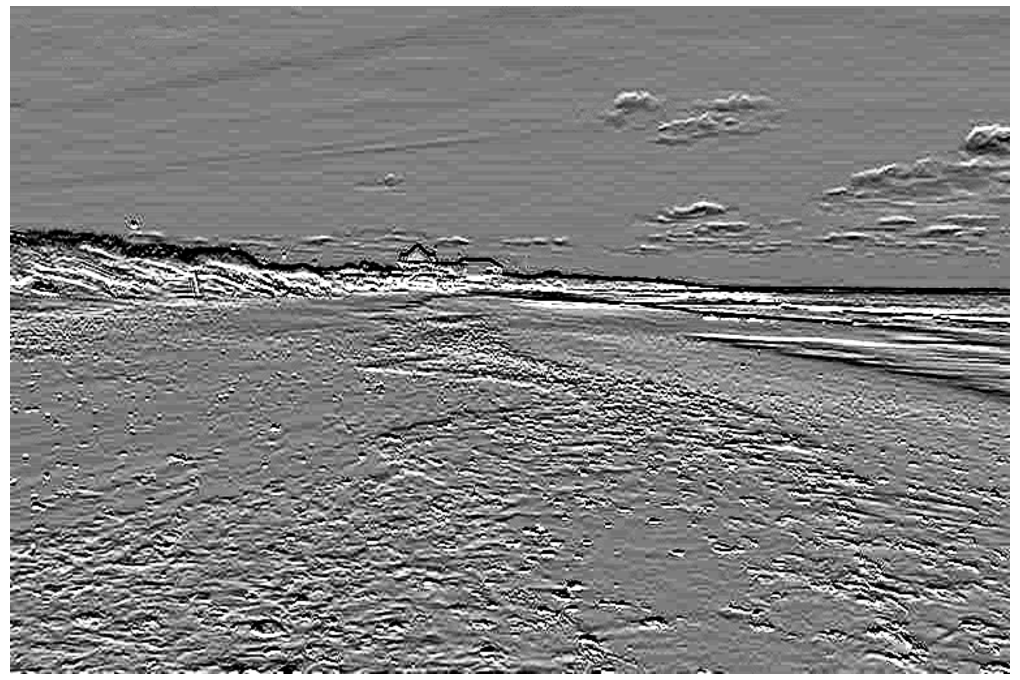
Edge orientation histograms

Input image

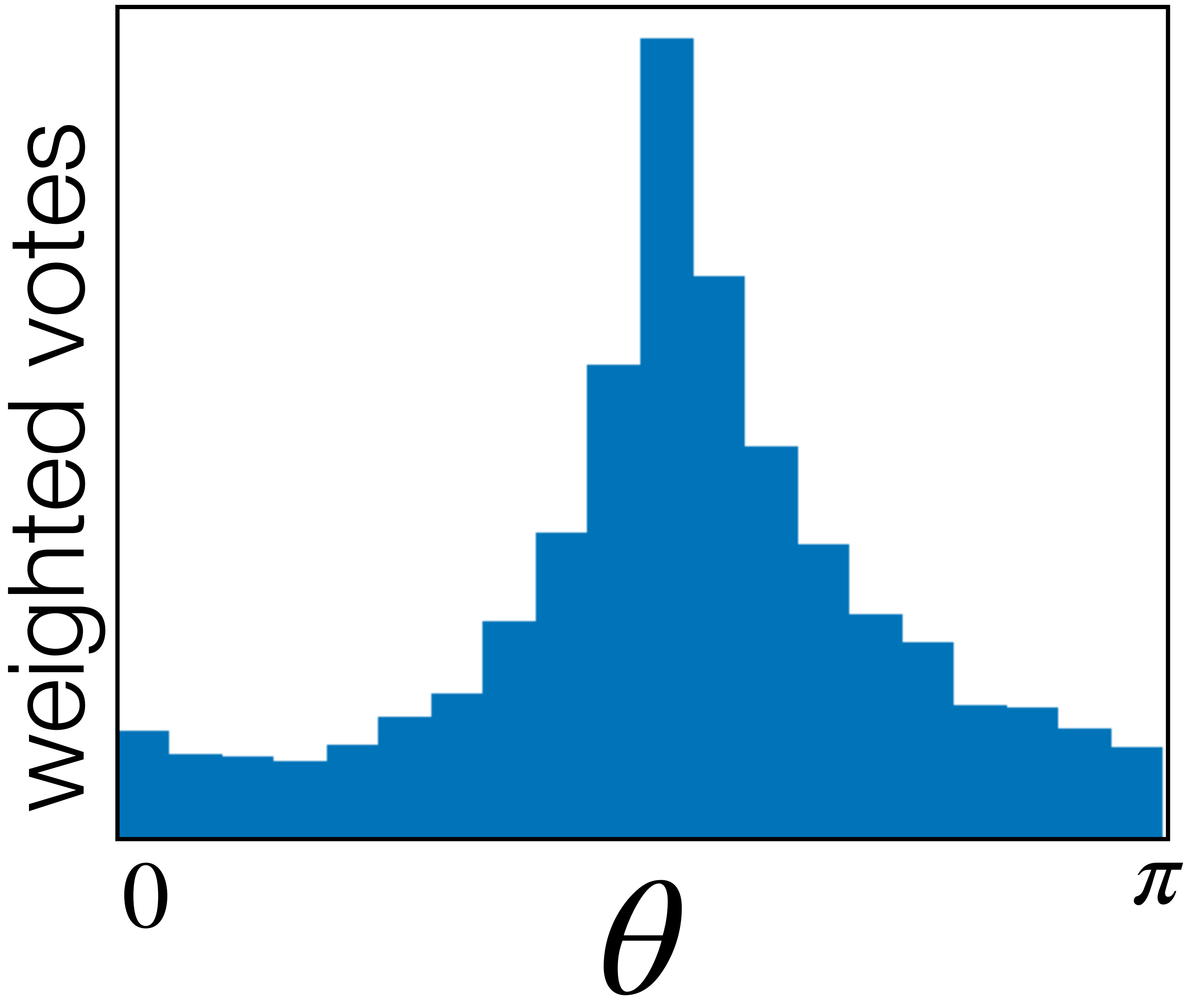
magnitude



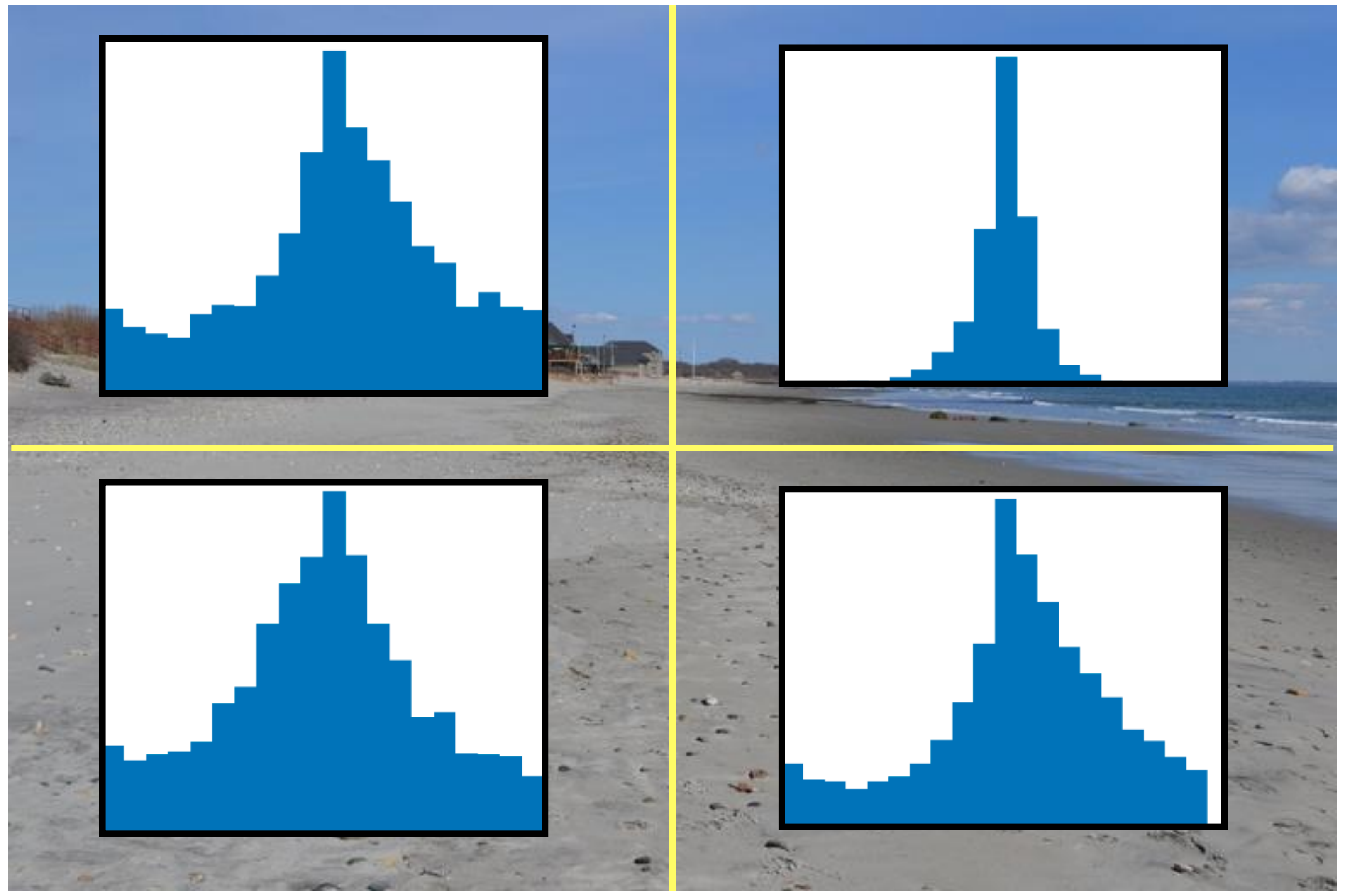
dx



dy



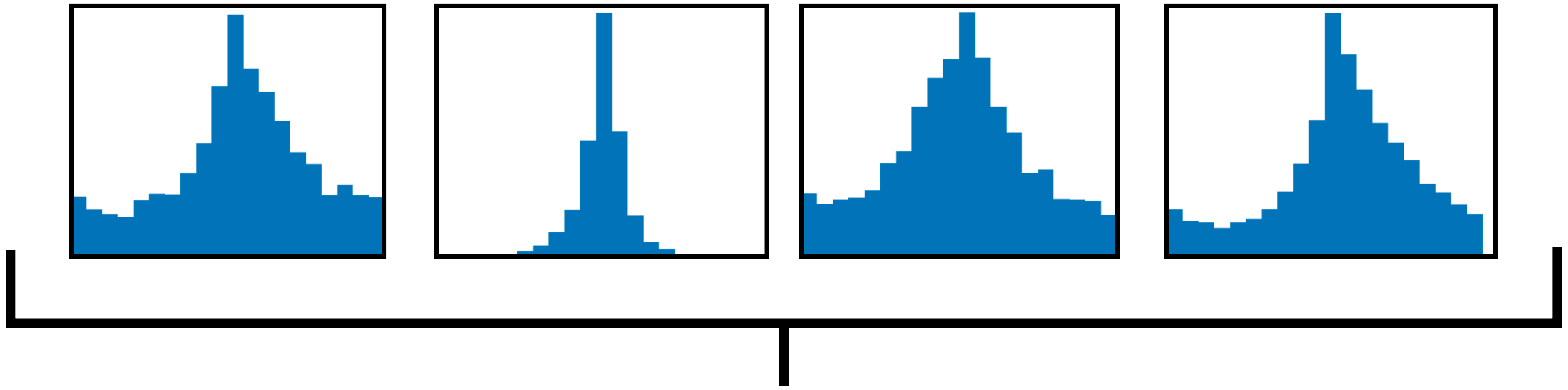
Edge orientation histograms



Input image

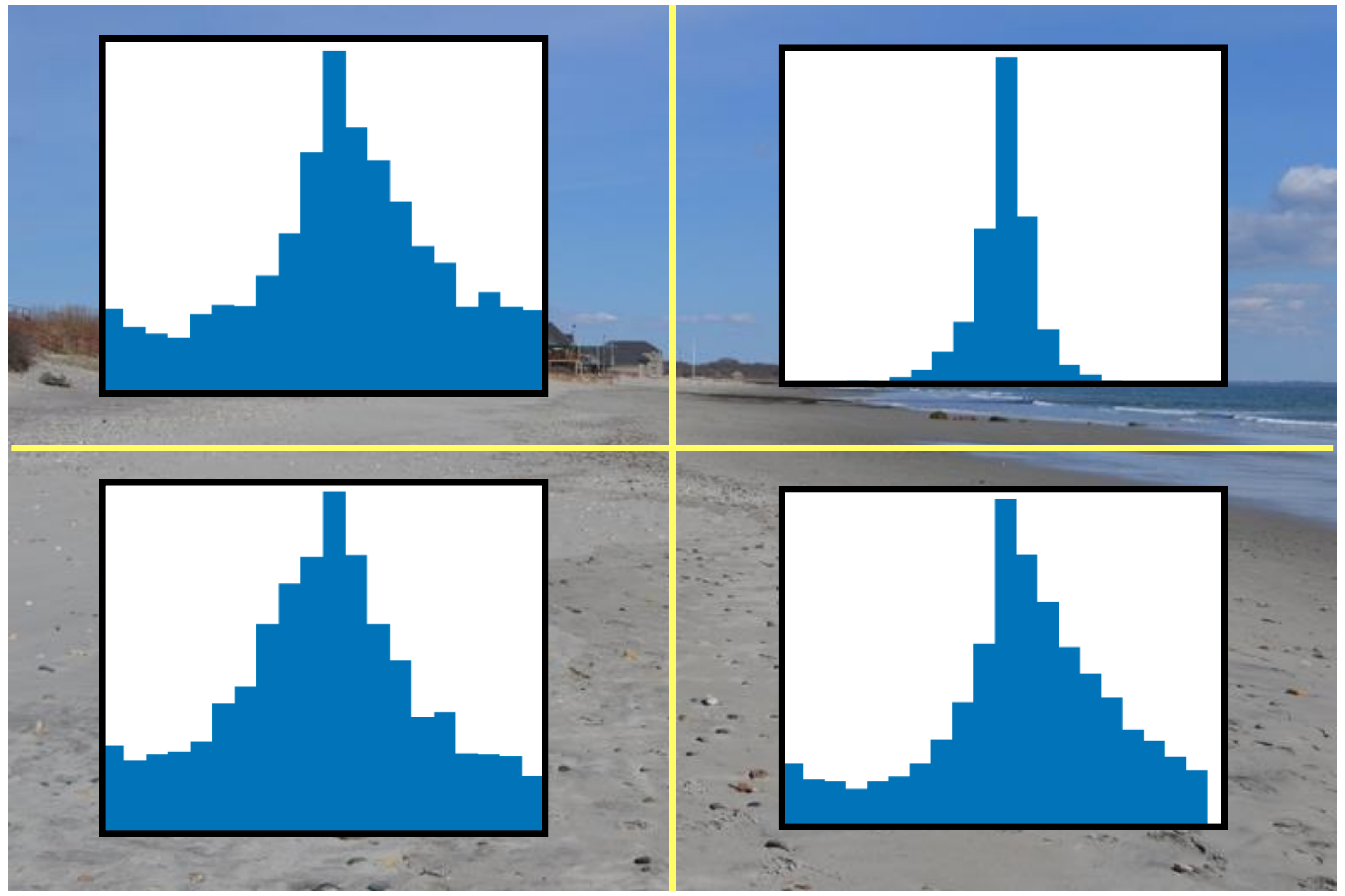
Edge orientation histograms

Input image



x

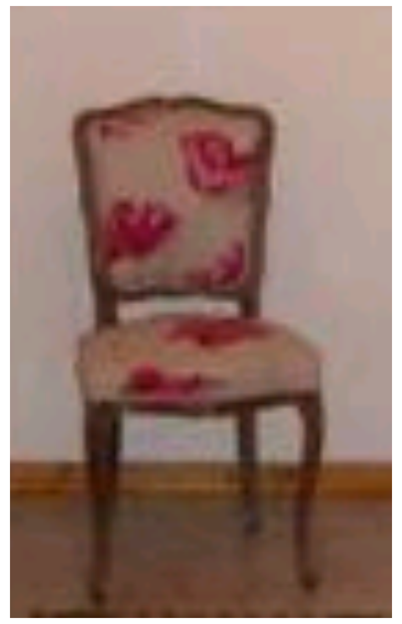
Edge orientation histograms


Input image

- Common variation is called: Histograms of Oriented Gradients (HOG) [Dalal and Triggs 2005].
- Often these use more complex normalization and weighting schemes.
- Used for image matching with SIFT features [Lowe 1999].
- We'll discuss *feature learning* methods next week. These usually perform a lot better!

Nearest Neighbor

- Hard to define similarity metric

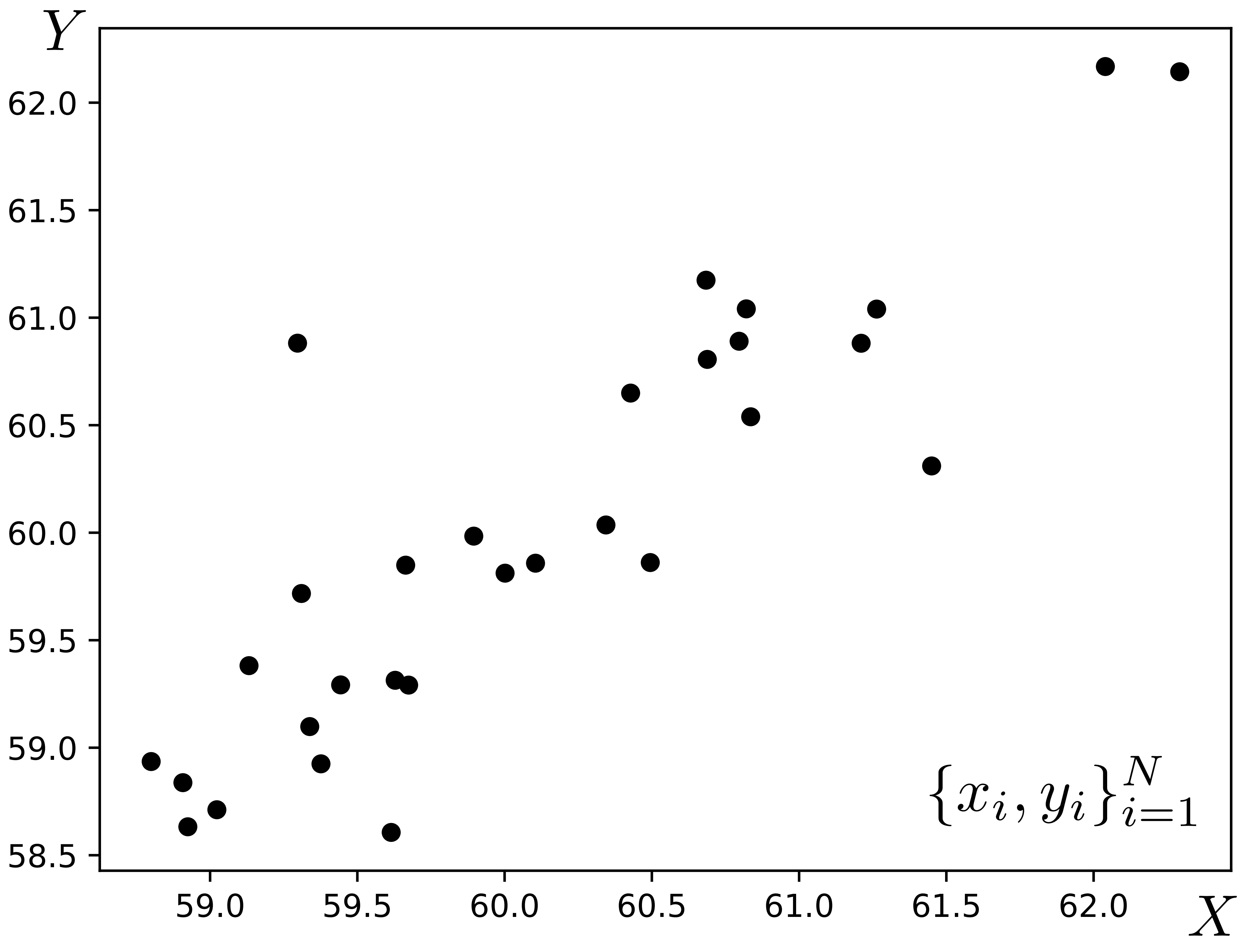
$\Big\|$  $-$  $\Big\| = \; ?$

- Pays attention to all features equally
- Can be very slow: $O(nd)$

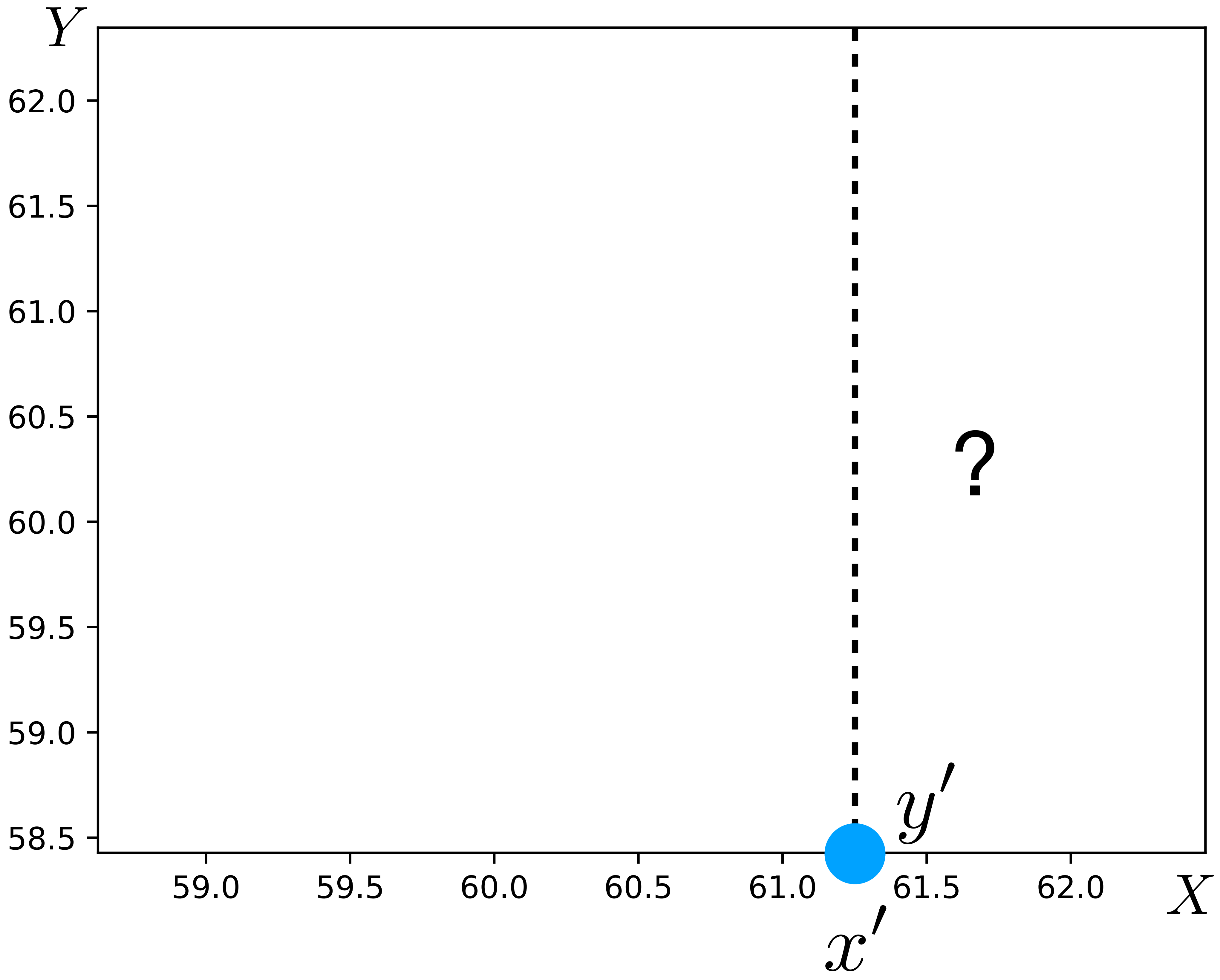
Case study #2: Linear least squares

Source: Isola, Torralba, Freeman

Training data

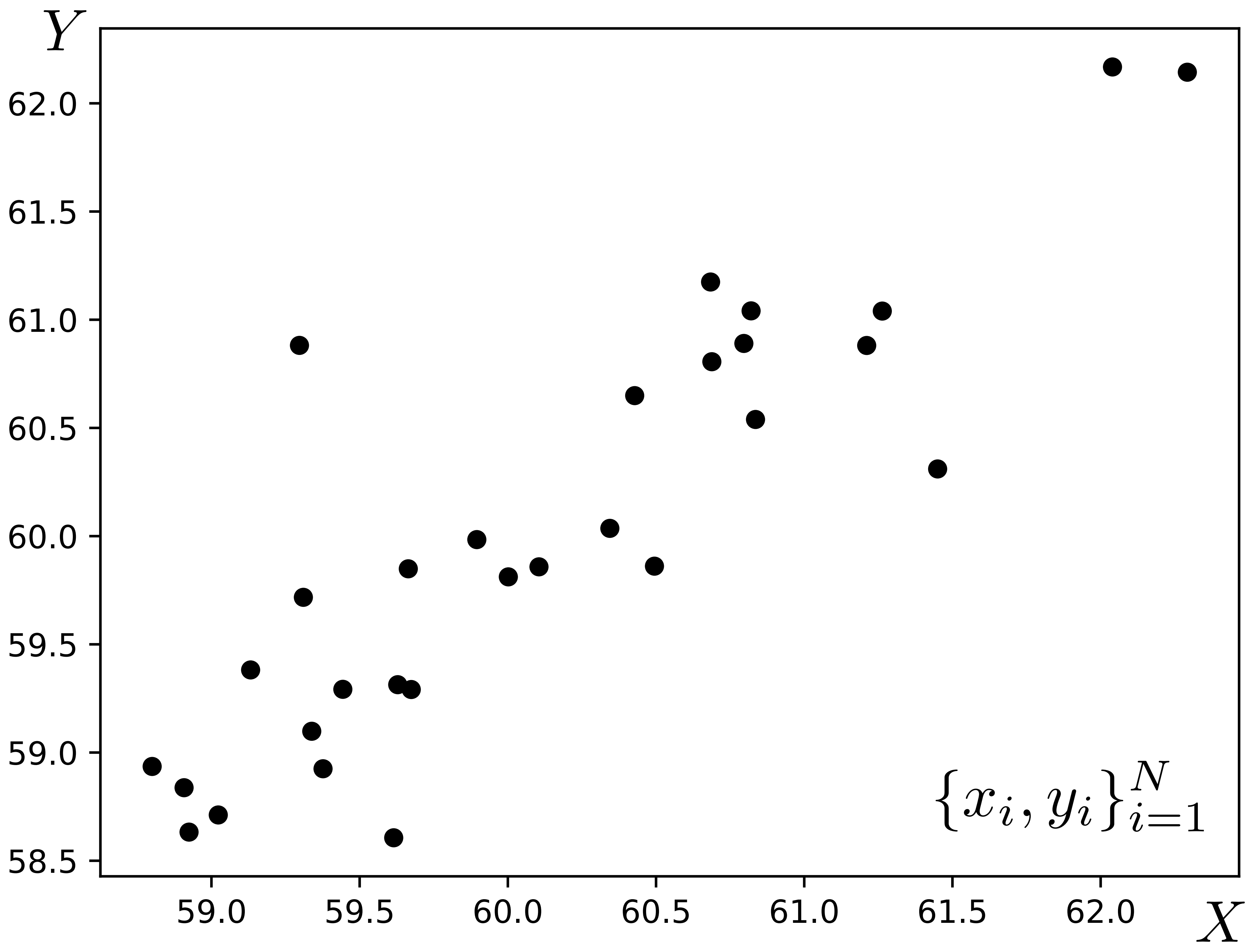


Test query



Source: Isola, Torralba, Freeman

Training data



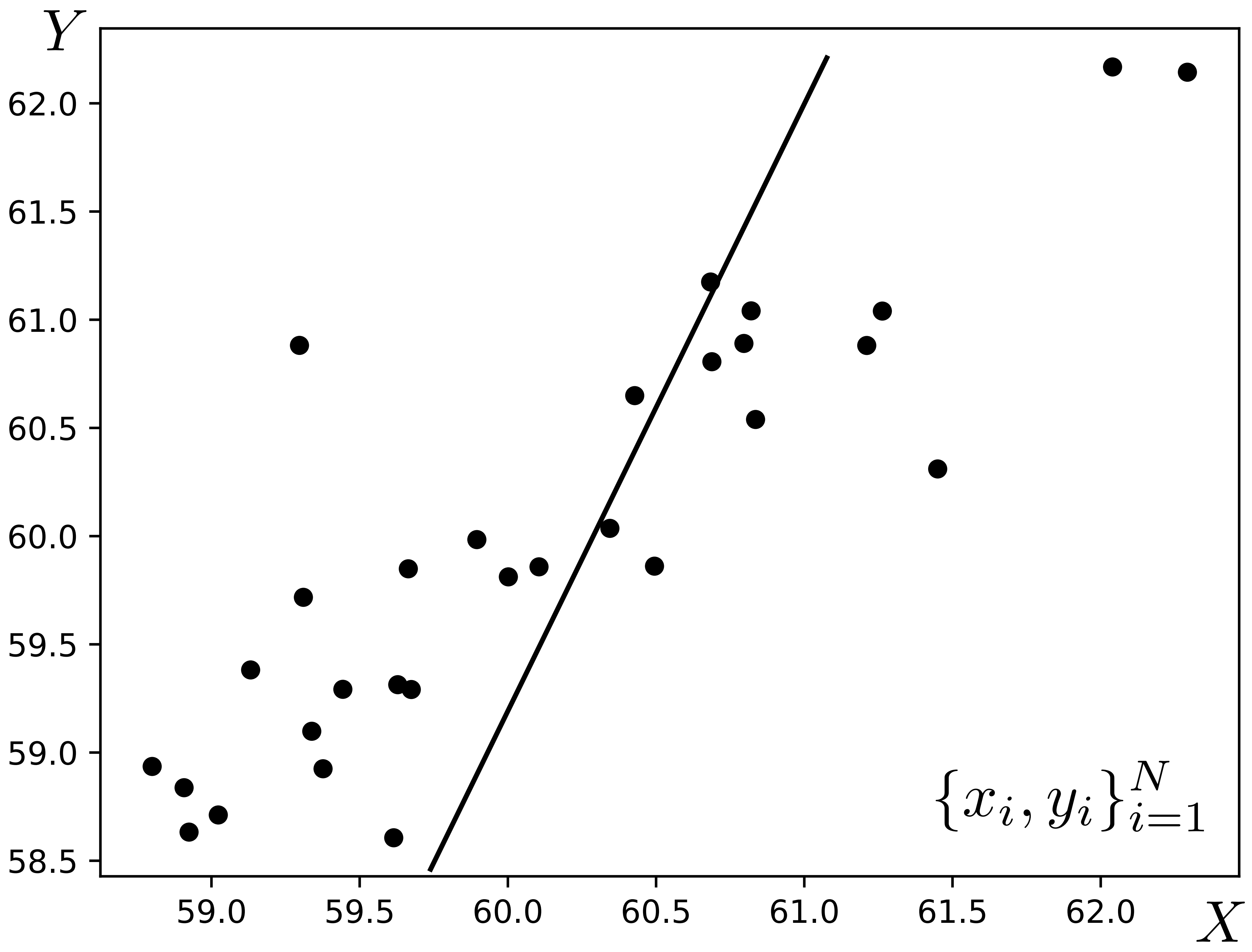
$\rightarrow$ Learner $\rightarrow$ $f_\theta(x) = \theta_1 x + \theta_0$

Hypothesis space

The relationship between X and Y is roughly linear: $y \approx \theta_1 x + \theta_0$

Source: Isola, Torralba, Freeman

Training data



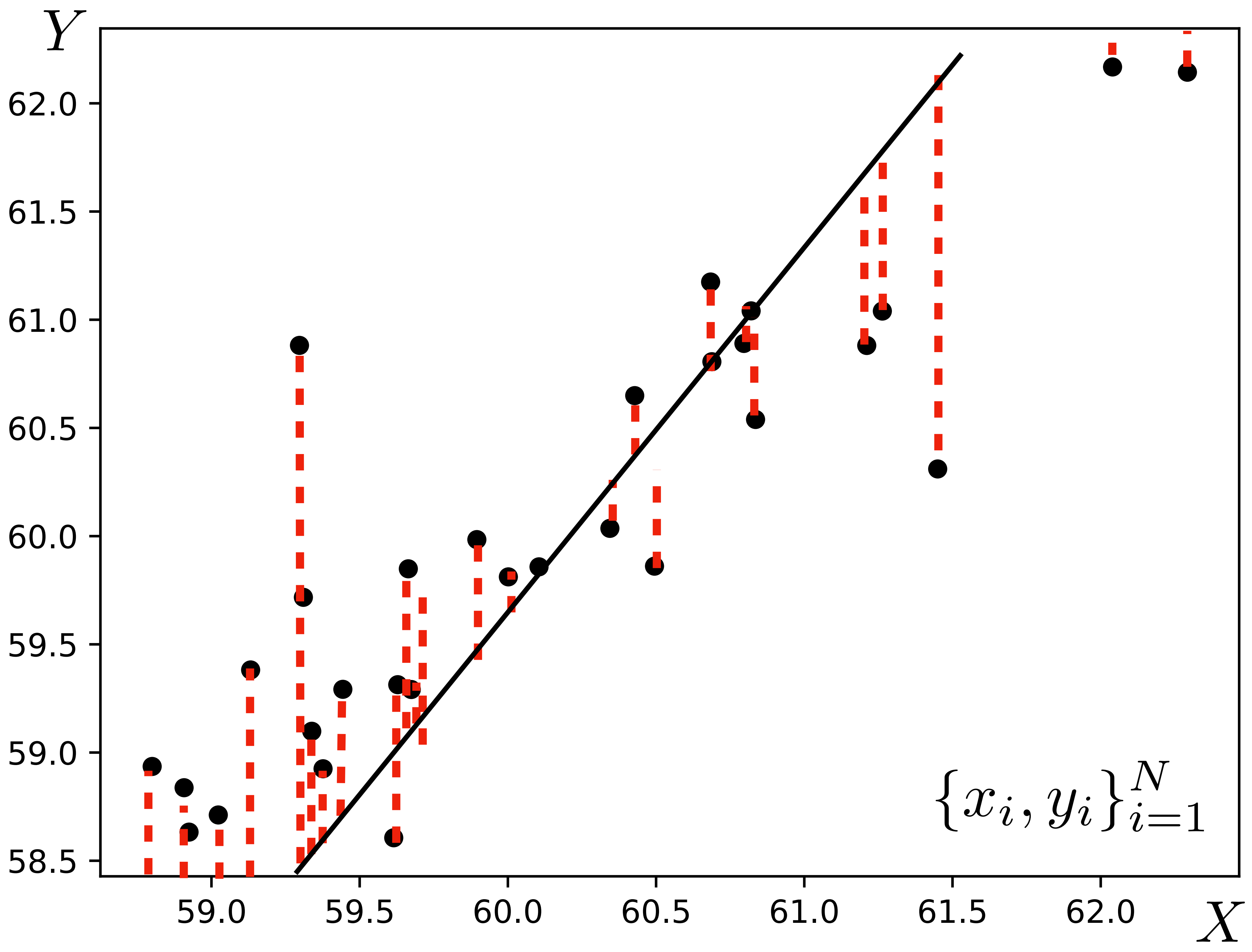
Search for the **parameters**, $\theta = \{\theta_0, \theta_1\}$, that best fit the data.

$$f_\theta(x) = \theta_1 x + \theta_0$$

Best fit in what sense?

Source: Isola, Torralba, Freeman

Training data



Search for the **parameters**, $\theta = \{\theta_0, \theta_1\}$, that best fit the data.

$$f_\theta(x) = \theta_1 x + \theta_0$$

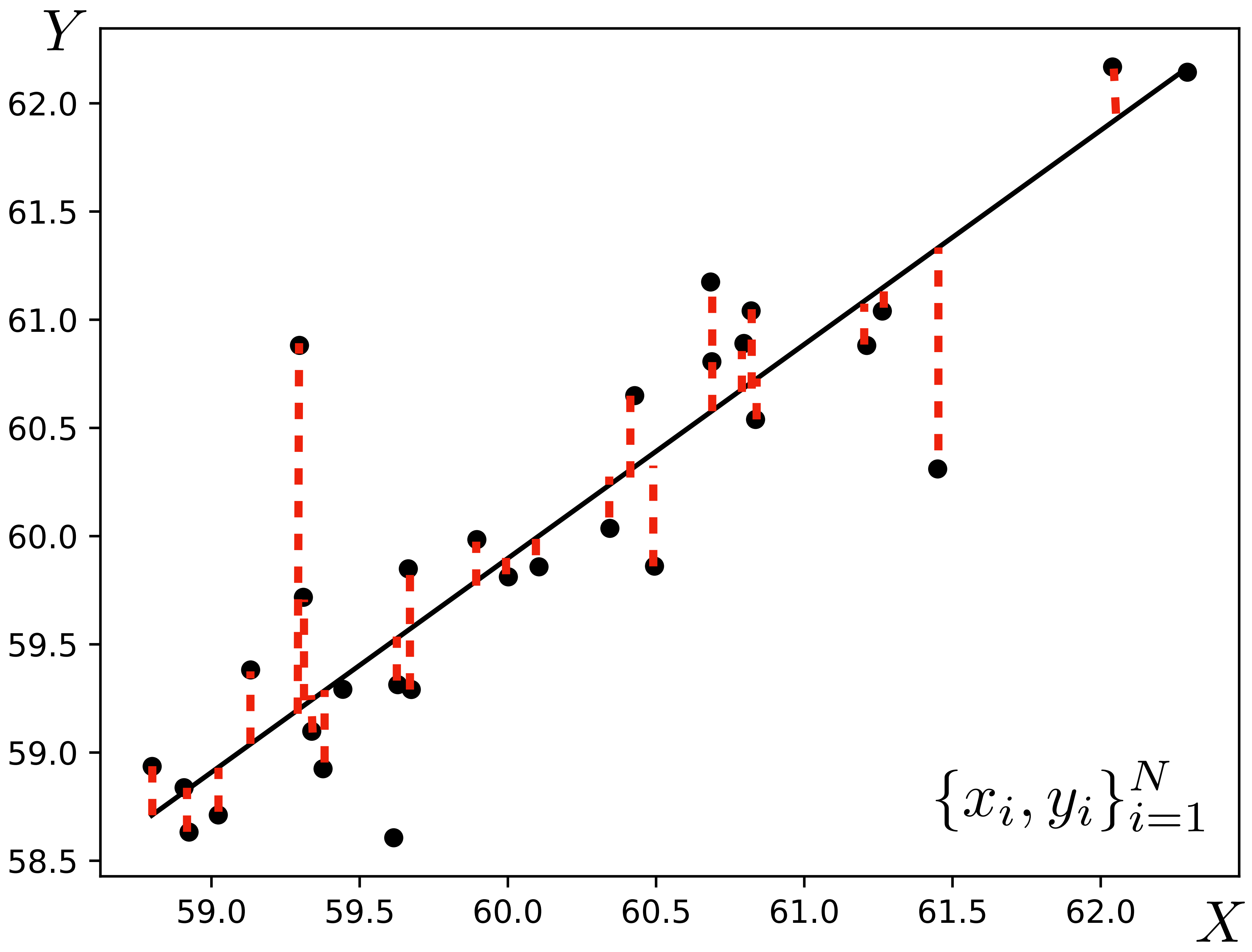
Best fit in what sense?

The least-squares **objective** (aka **loss**) says the best fit is the function that minimizes the squared error between predictions and target values:

$$\mathcal{L}(\hat{y}, y) = (\boxed{\hat{y} - y})^2 \qquad \hat{y} \equiv f_\theta(x)$$

Source: Isola, Torralba, Freeman

Training data



Search for the **parameters**, $\theta = \{\theta_0, \theta_1\}$, that best fit the data.

$$f_\theta(x) = \theta_1 x + \theta_0$$

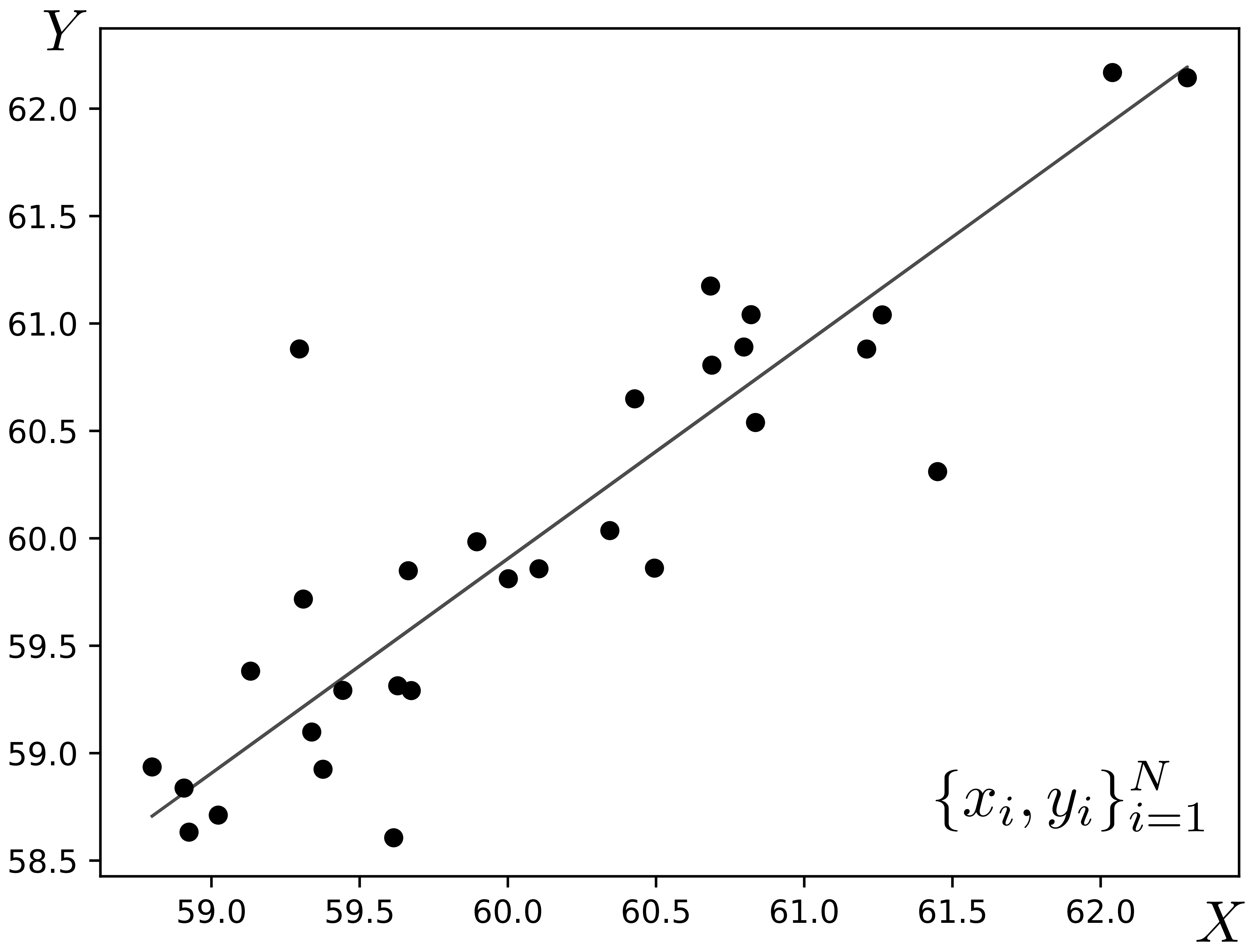
Best fit in what sense?

The least-squares **objective** (aka **loss**) says the best fit is the function that minimizes the squared error between predictions and target values:

$$\mathcal{L}(\hat{y}, y) = (\boxed{\hat{y} - y})^2 \qquad \hat{y} \equiv f_\theta(x)$$

Source: Isola, Torralba, Freeman

Training data



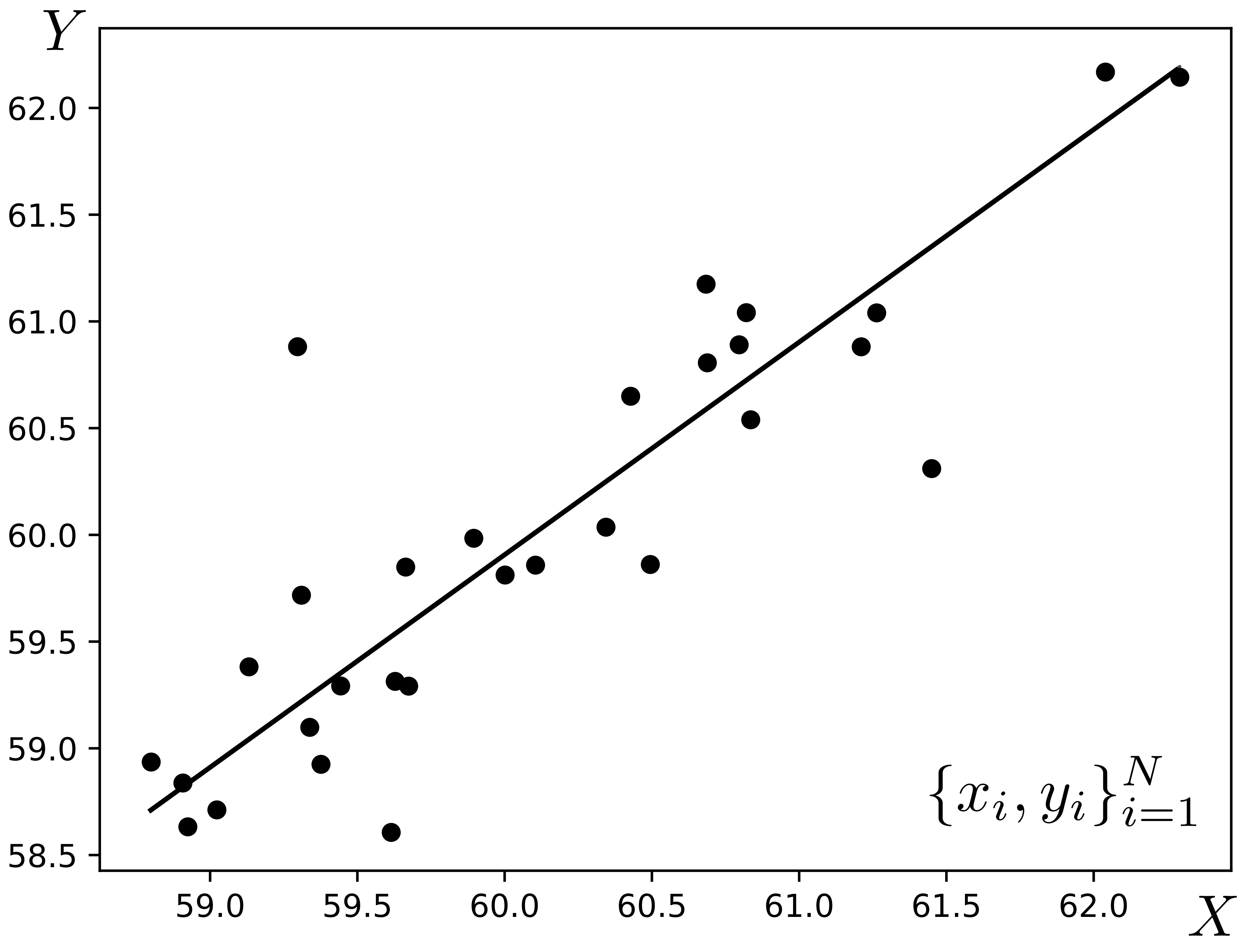
Complete learning problem:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} (f_\theta(x_i) - y_i)^2$$

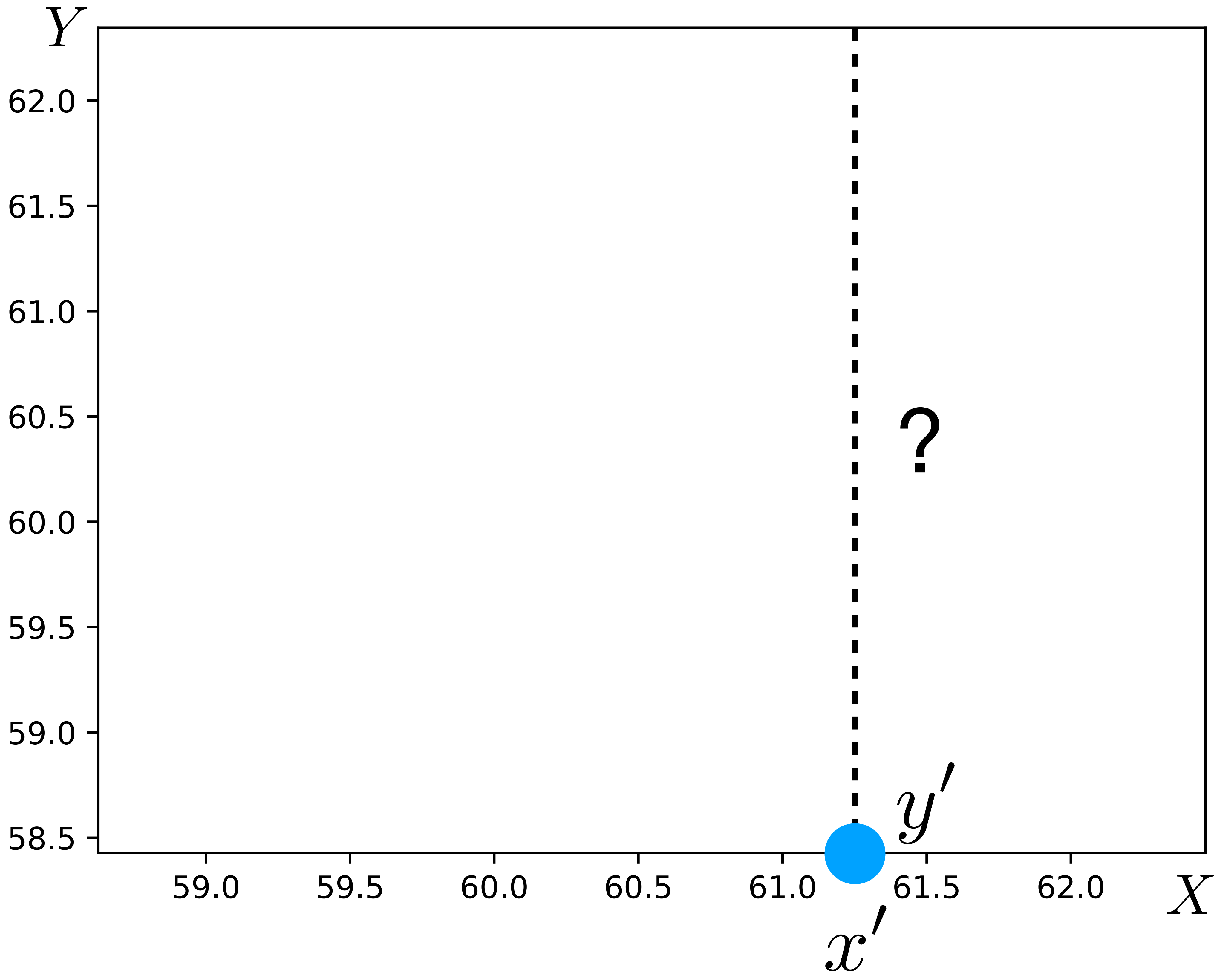
$$= \arg\min_\theta \sum_{i=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2$$

Source: Isola, Torralba, Freeman

Training data

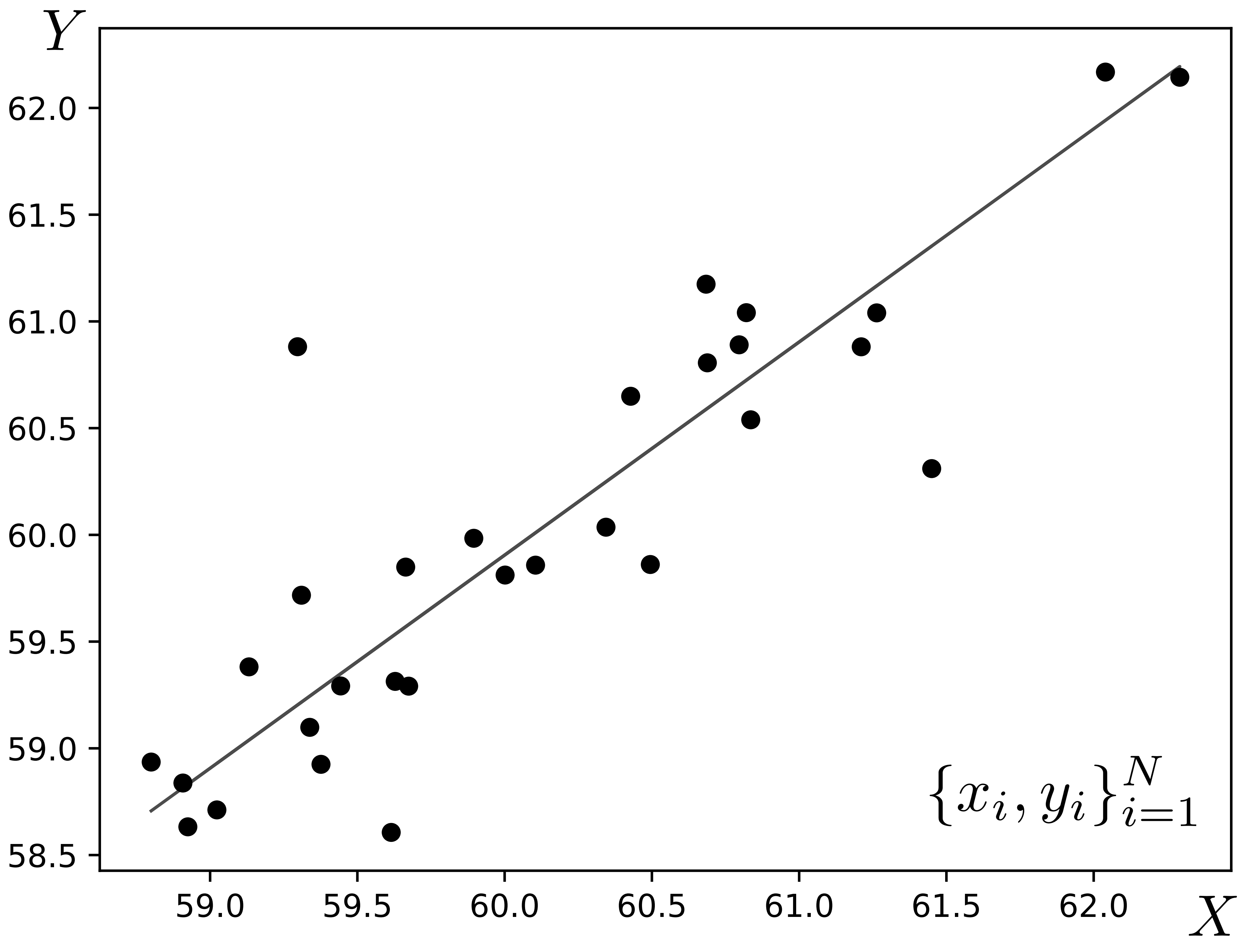


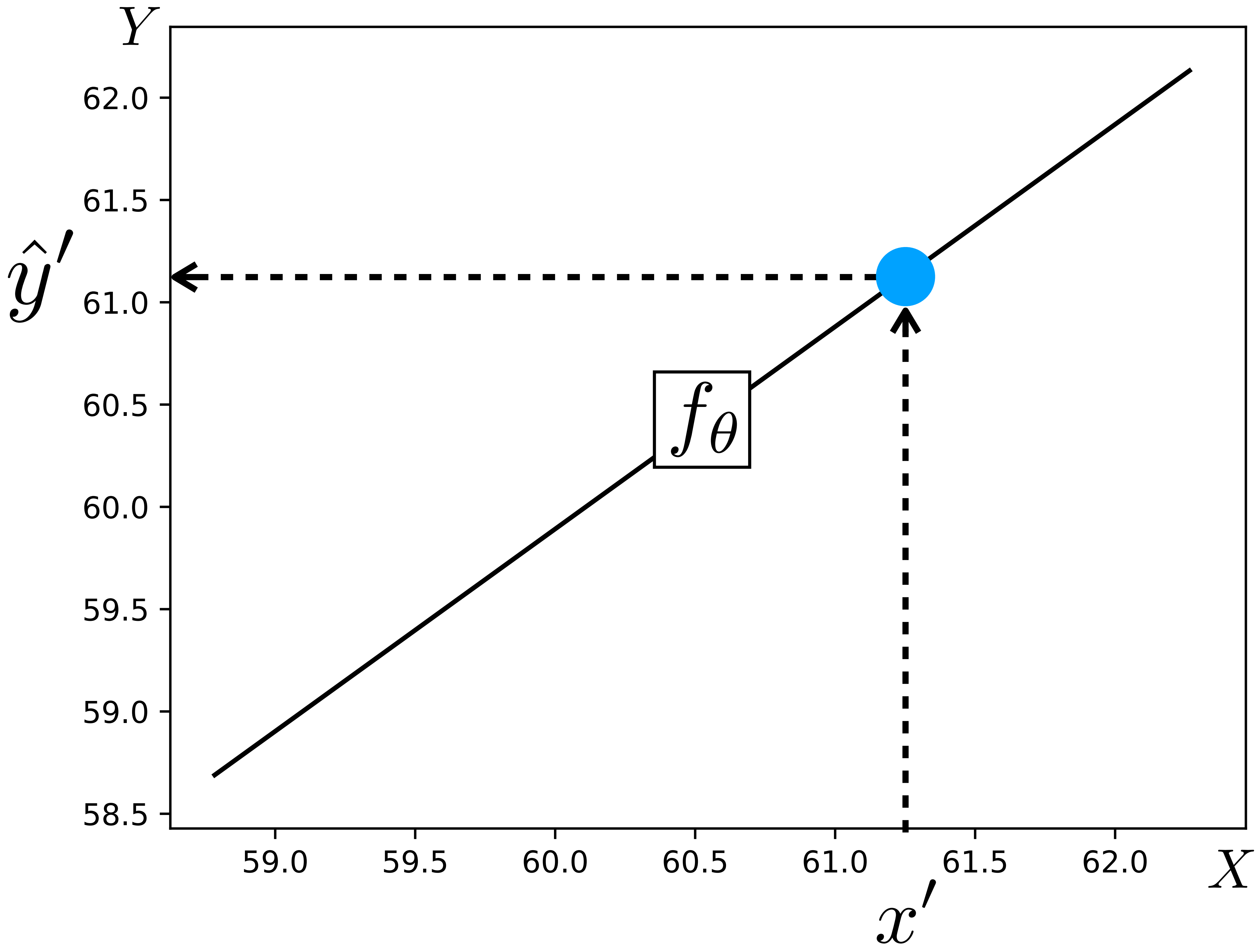
Test query



Source: Isola, Torralba, Freeman

Training data	Test query





$$x' \dashrightarrow \boxed{f_\theta} \dashrightarrow \hat{y}'$$

Source: Isola, Torralba, Freeman

How to minimize the objective w.r.t. θ?

Learning problem

$$\boxed{\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2}$$

Recall:

$$J(\theta) = \sum_{I=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2$$
$$= (\mathbf{y} - \mathbf{X}\theta)^T (\mathbf{y} - \mathbf{X}\theta)$$

$$\mathbf{X} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix} \quad \theta = \begin{pmatrix} \theta_1 & \theta_0 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

$$\theta^* = \arg\min_{\theta} J(\theta)$$

$$\frac{\partial J(\theta)}{\partial \theta} = 0$$

$$\frac{\partial J(\theta)}{\partial \theta} = 2(\mathbf{X}^T\mathbf{X}\theta - \mathbf{X}^T\mathbf{y})$$

$$2(\mathbf{X}^T\mathbf{X}\theta^* - \mathbf{X}^T\mathbf{y}) = 0$$

$$\mathbf{X}^T\mathbf{X}\theta^* = \mathbf{X}^T\mathbf{y}$$

$$\boxed{\theta^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

Solution

Source: Isola, Torralba, Freeman

Empirical Risk Minimization

(formalization of supervised learning)

Linear least squares learning problem

$$\boxed{\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2}$$

Source: Isola, Torralba, Freeman

Empirical Risk Minimization

(formalization of supervised learning)

Objective function
(loss)

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i)$$

Training data

Hypothesis space

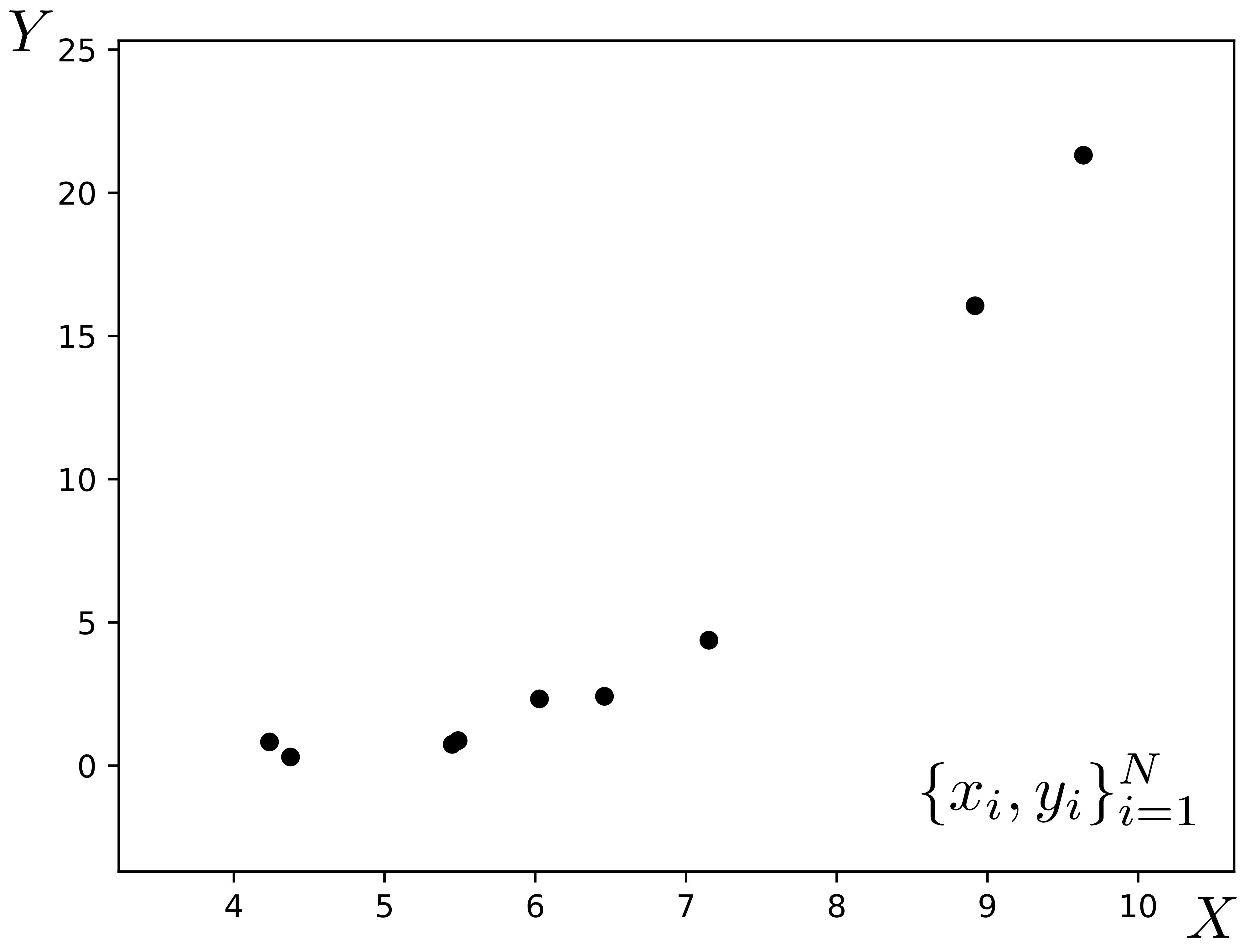
Source: Isola, Torralba, Freeman

The Problem of Generalization

Source: Isola, Torralba, Freeman

Linear regression

Training data

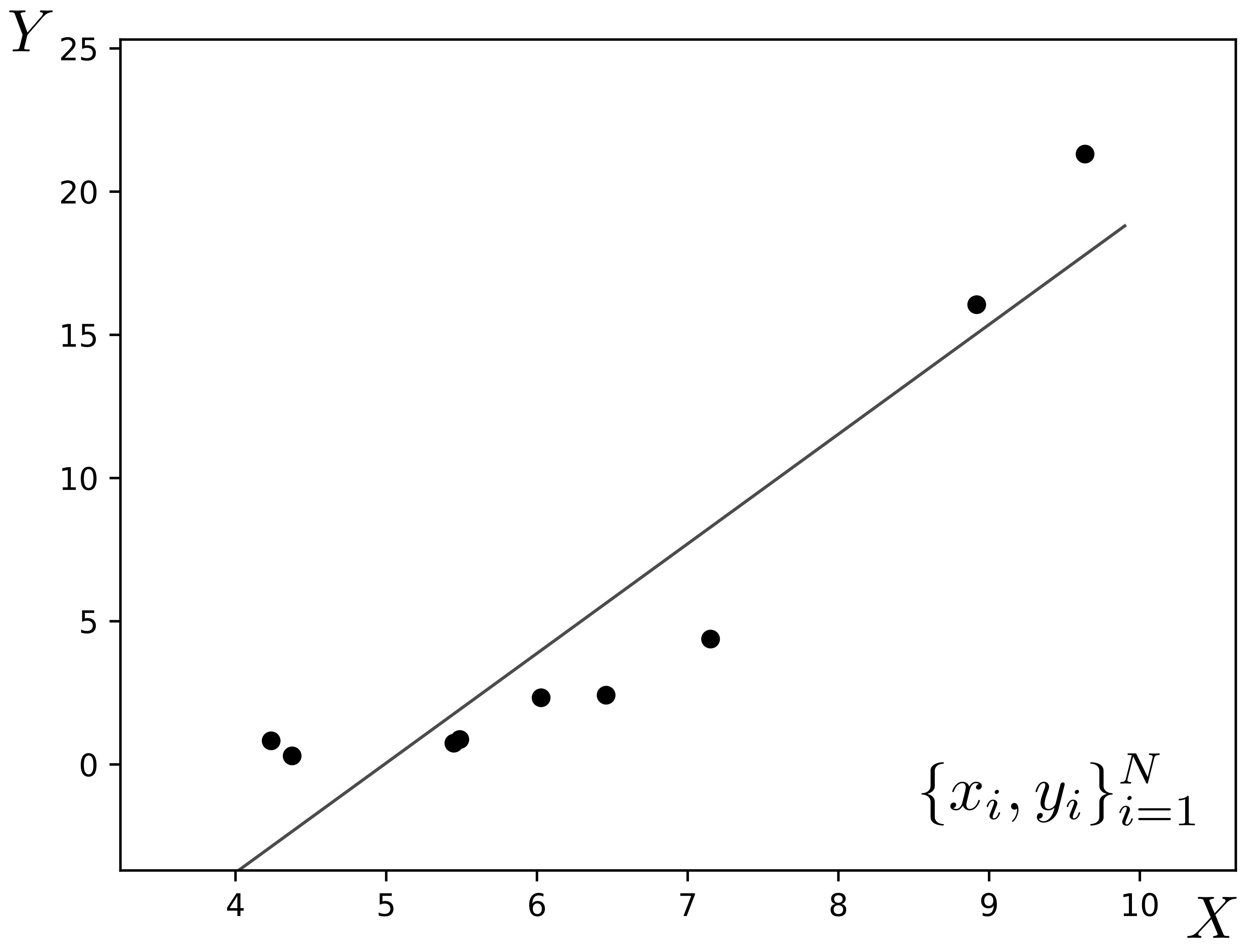


$$f_\theta(x) = \theta_0 + \theta_1 x$$

Source: Isola, Torralba, Freeman

Linear regression

Training data

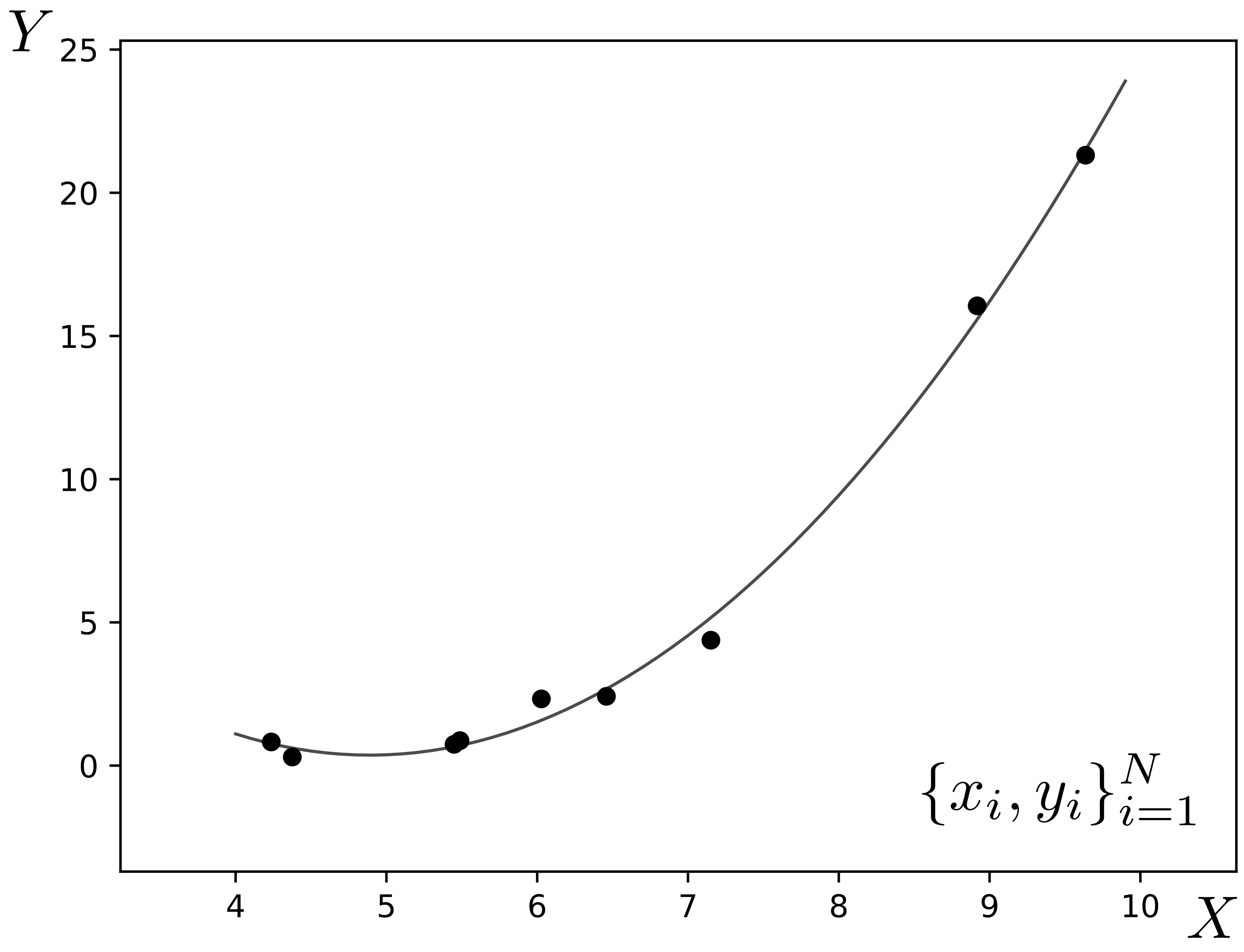


$$f_\theta(x) = \theta_0 + \theta_1 x$$

Source: Isola, Torralba, Freeman

Polynomial regression

Training data



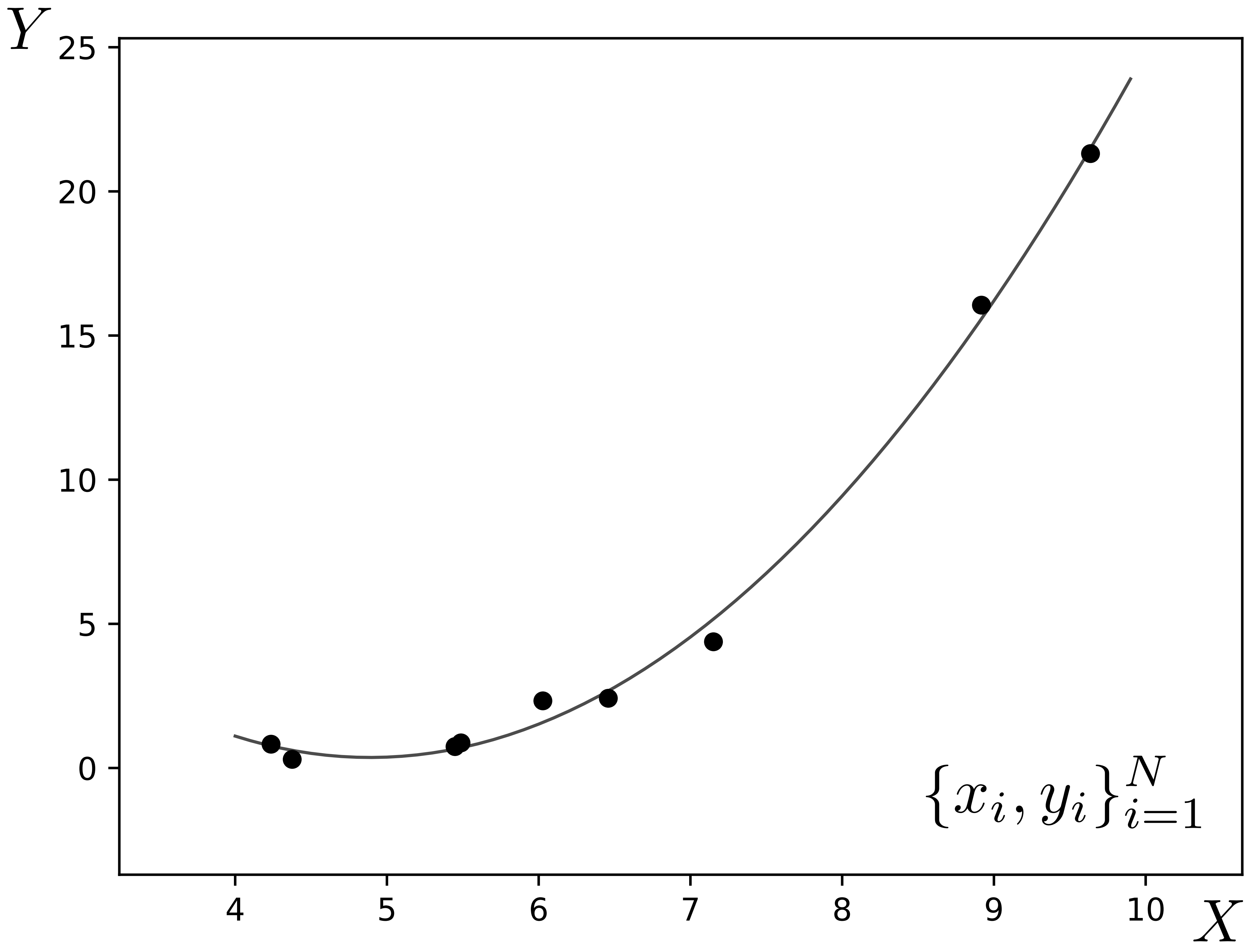
$$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

K-th degree polynomial regression

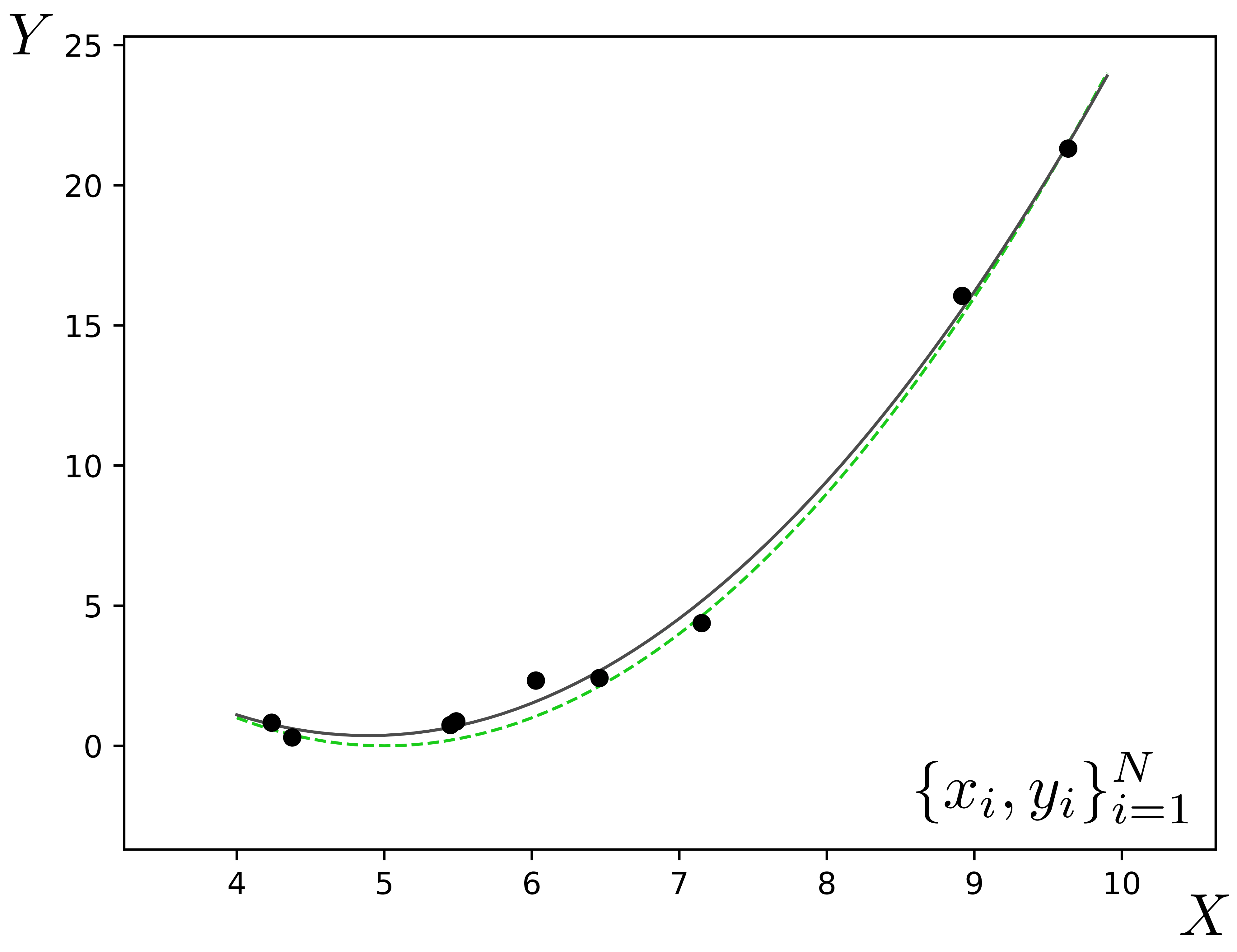
Source: Isola, Torralba, Freeman

Training data

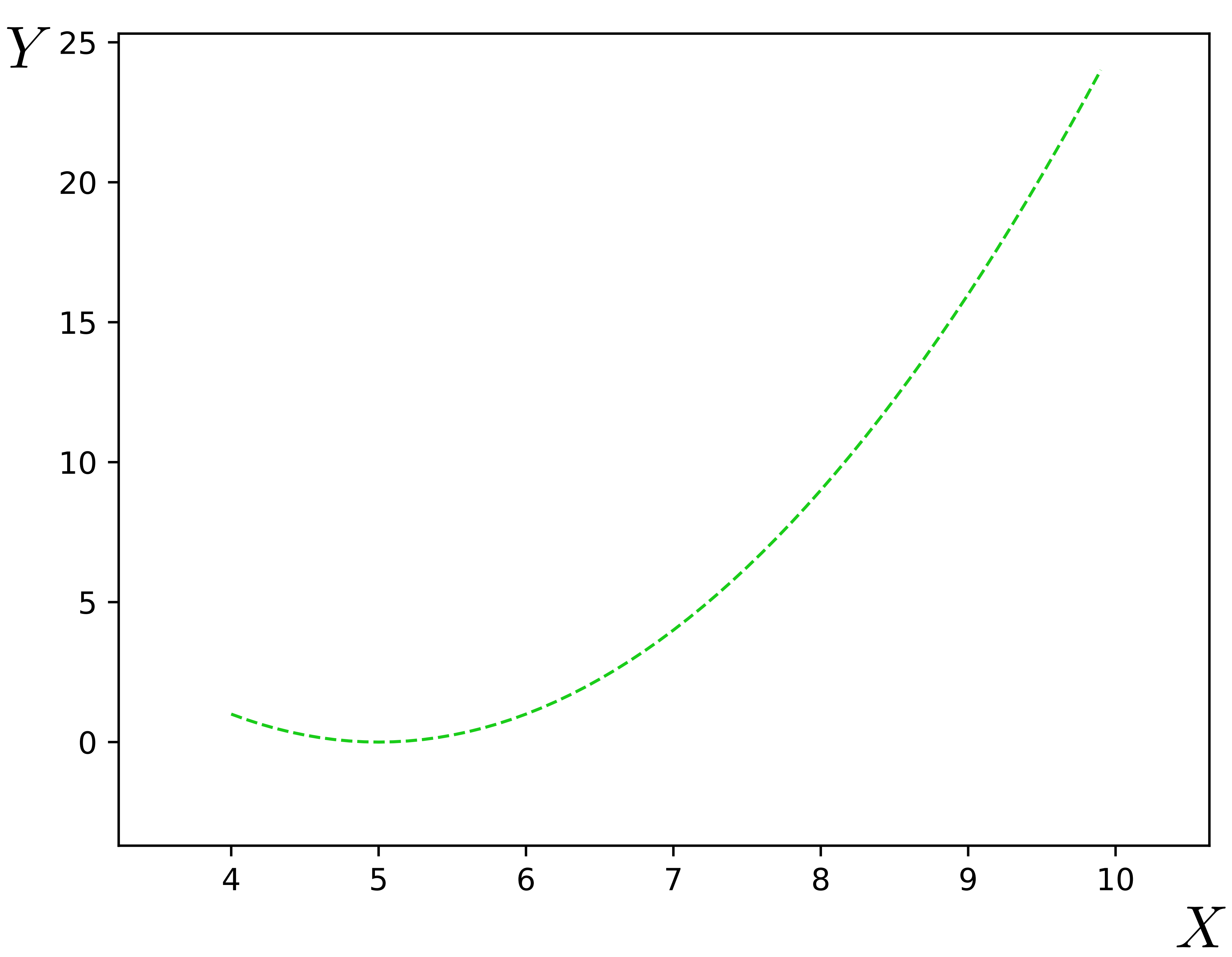


Source: Isola, Torralba, Freeman

Training data



Test data

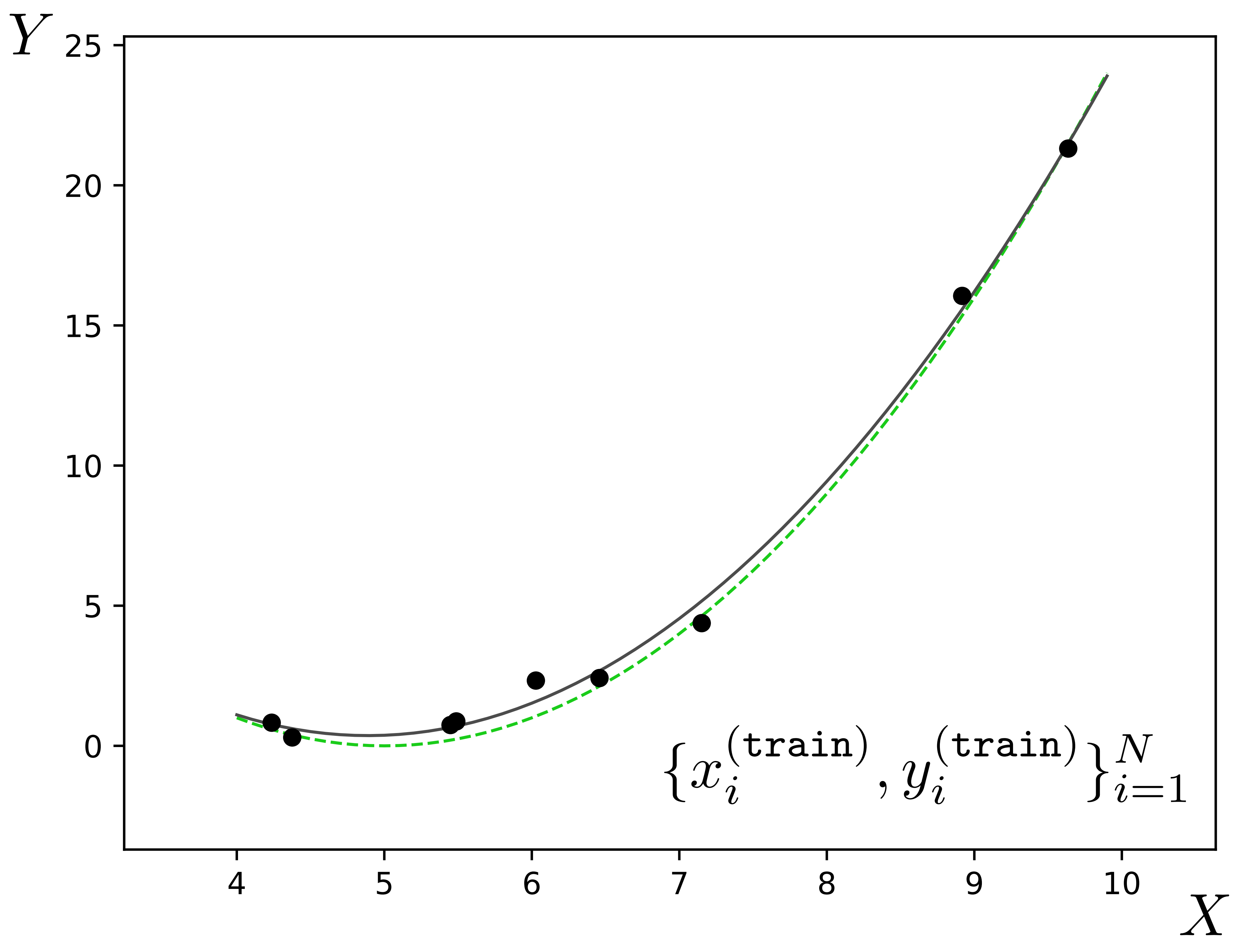


True **data-generating process**

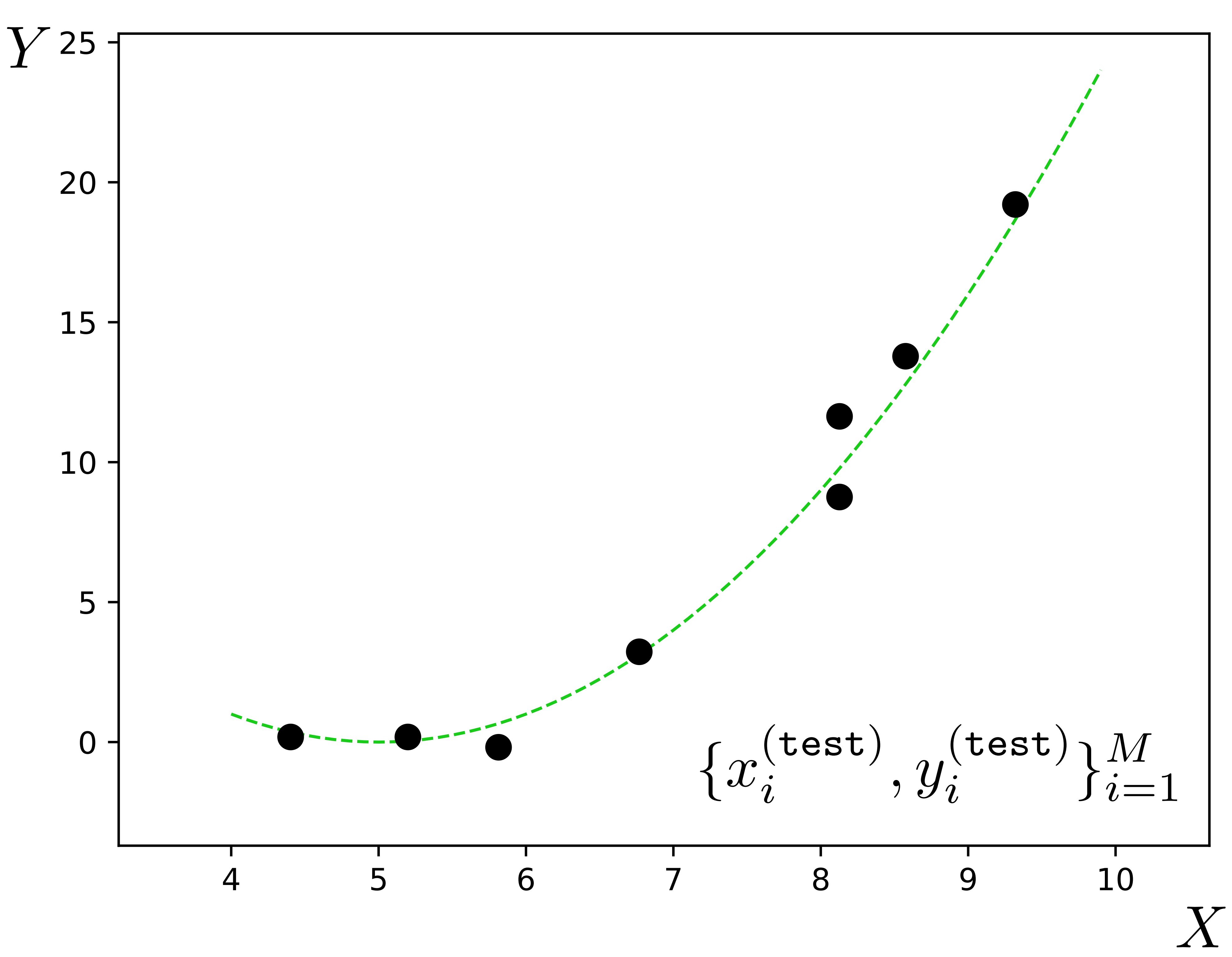
$p_{\texttt{data}}$

Source: Isola, Torralba, Freeman

Training data



Test data



True **data-generating process**

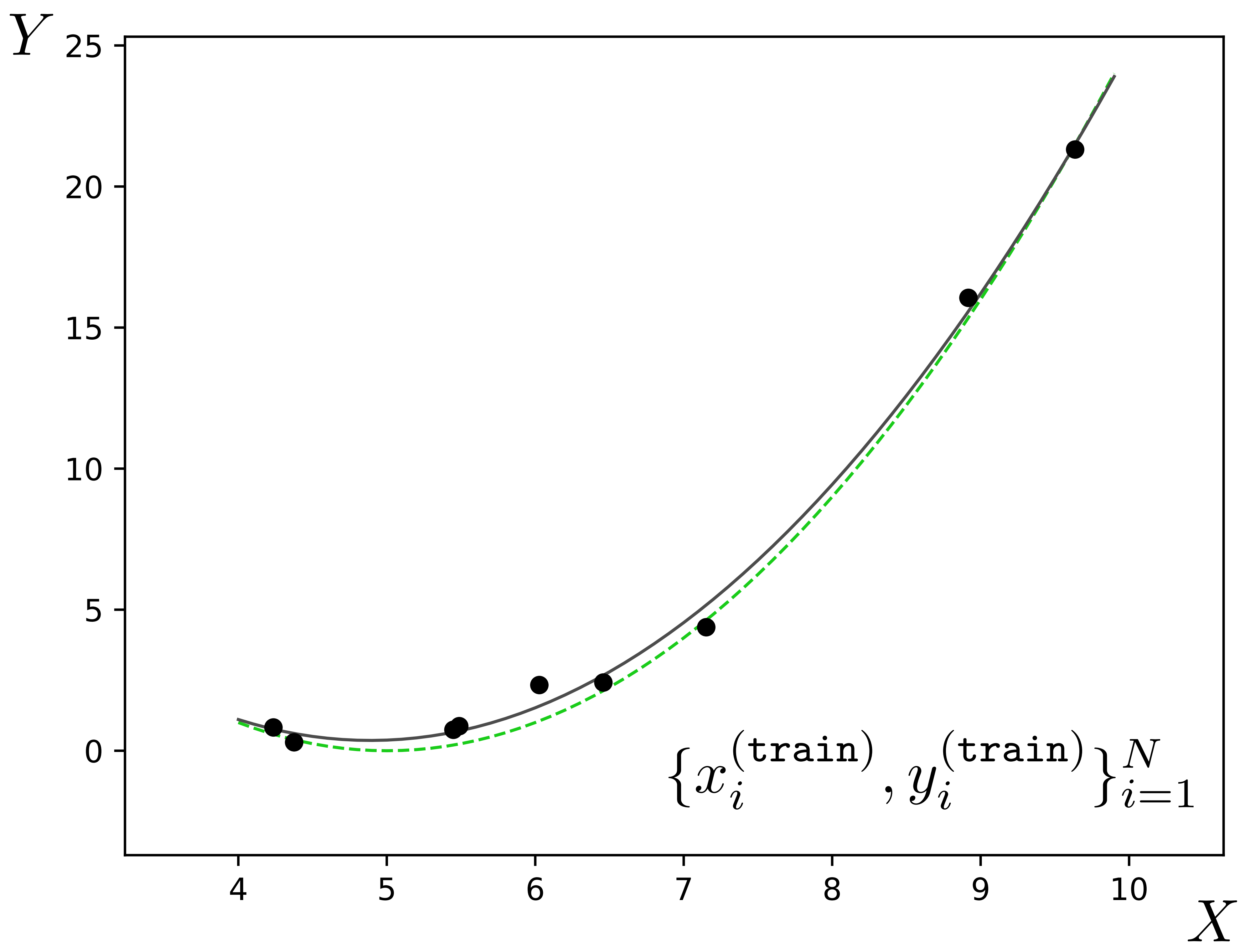
$$p_{\texttt{data}}$$

$$\{x_i^{(\texttt{train})}, y_i^{(\texttt{train})}\} \overset{\texttt{iid}}{\sim} p_{\texttt{data}}$$

$$\{x_i^{(\texttt{test})}, y_i^{(\texttt{test})}\} \overset{\texttt{iid}}{\sim} p_{\texttt{data}}$$

Source: Isola, Torralba, Freeman

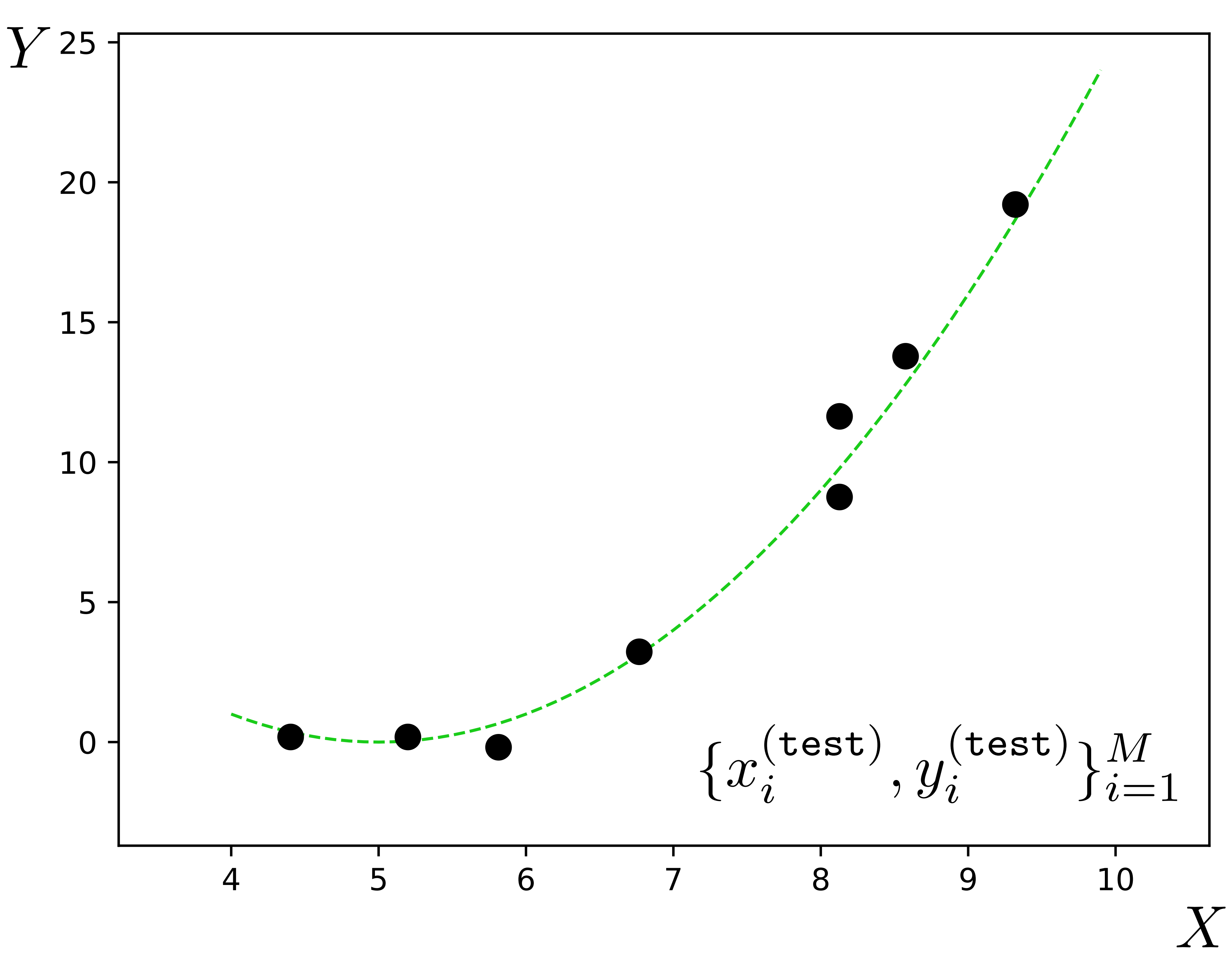
Training data



Training objective:

$$\sum_{i=1}^{N}(f_\theta(x_i^{\mathtt{train}}) - y_i^{\mathtt{train}})^2$$

Test data

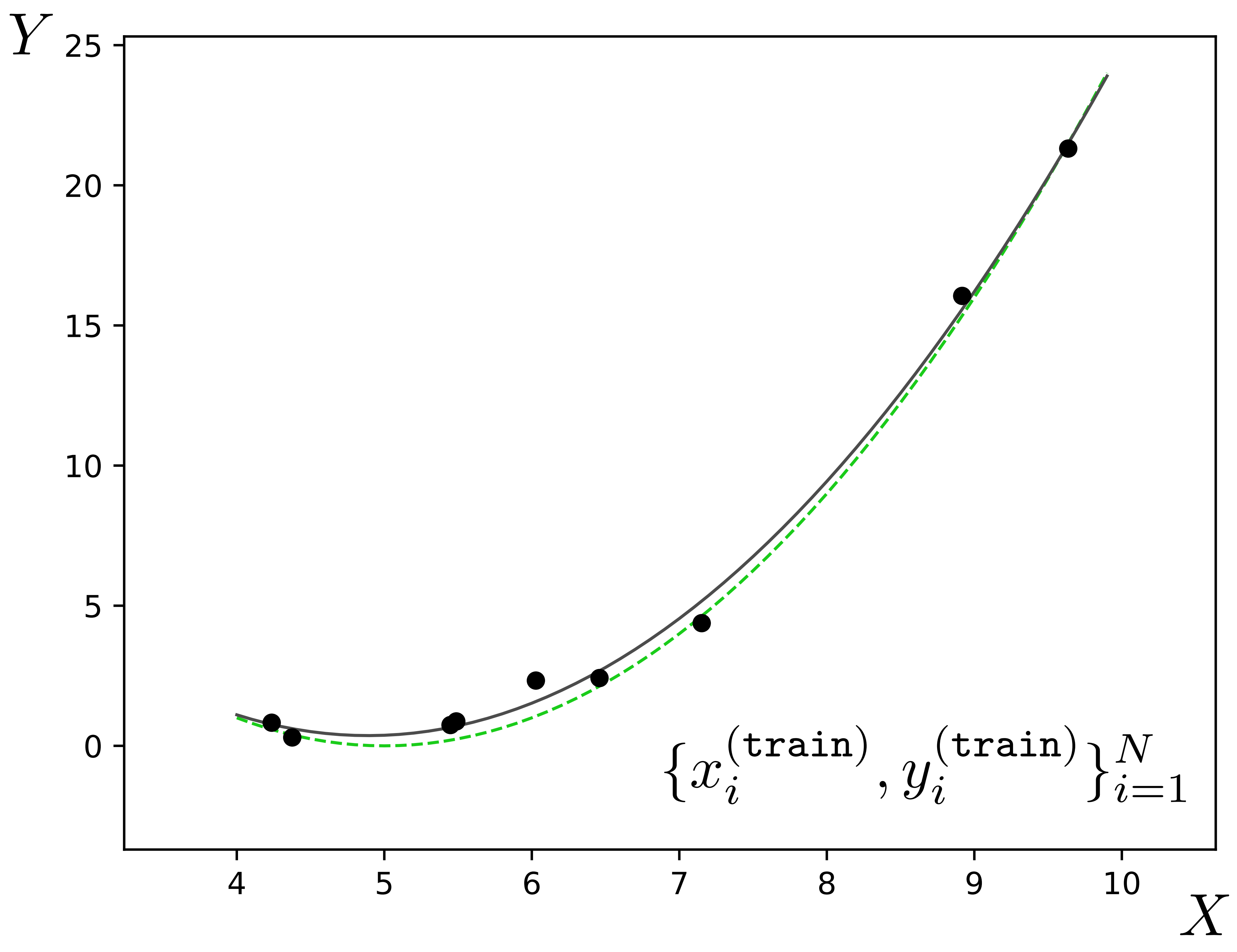


Test time evaluation:

$$\sum_{i=1}^{M}(f_\theta(x_i^{\mathtt{test}}) - y_i^{\mathtt{test}})^2$$

Source: Isola, Torralba, Freeman

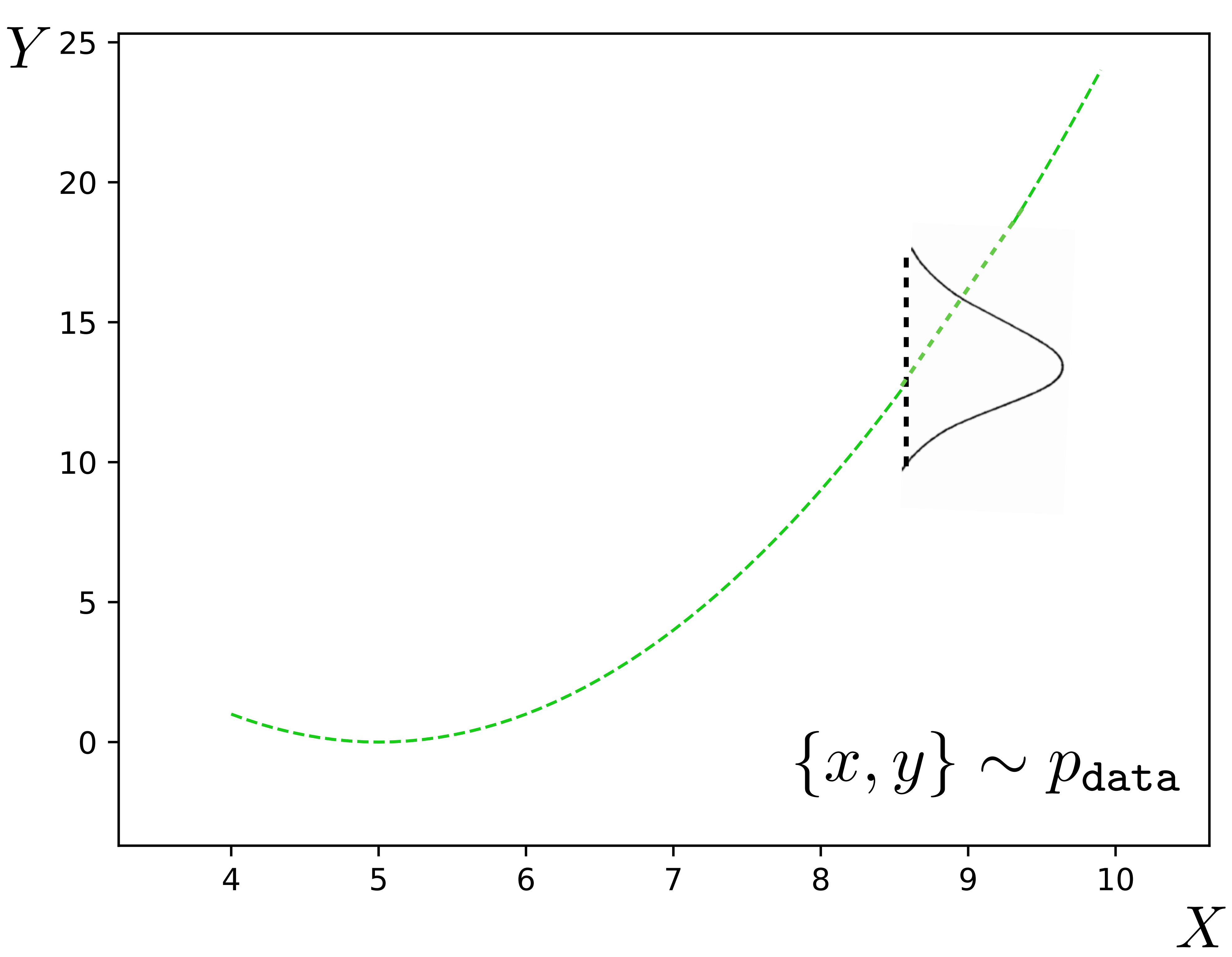
Training data



Training objective:

$$\sum_{i=1}^{N}(f_\theta(x_i^{\texttt{train}}) - y_i^{\texttt{train}})^2$$

Test data



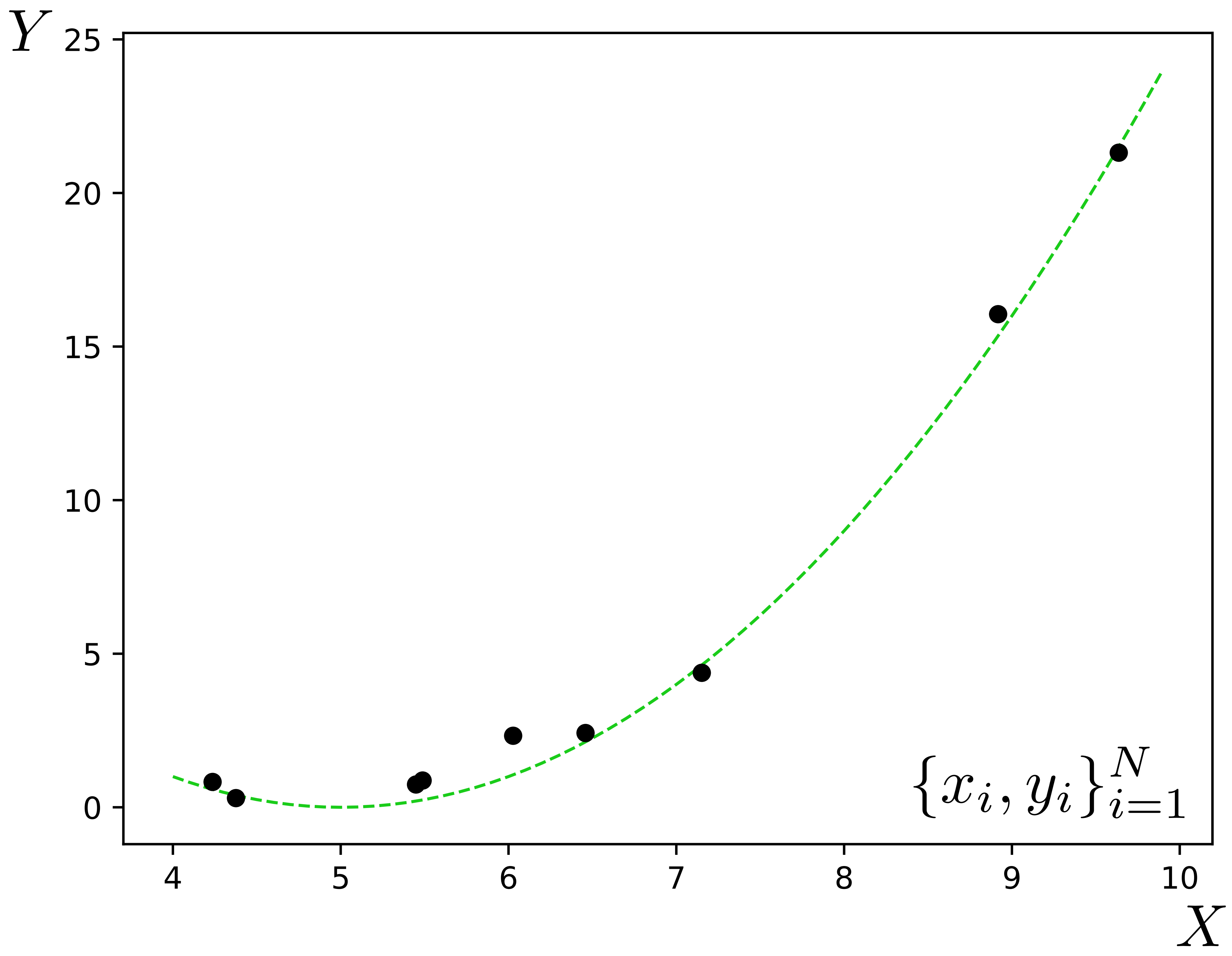
True objective:

$$\mathbb{E}_{\{x,y\}\sim p_{\texttt{data}}}[(f_\theta(x) - y)^2]$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

Training data

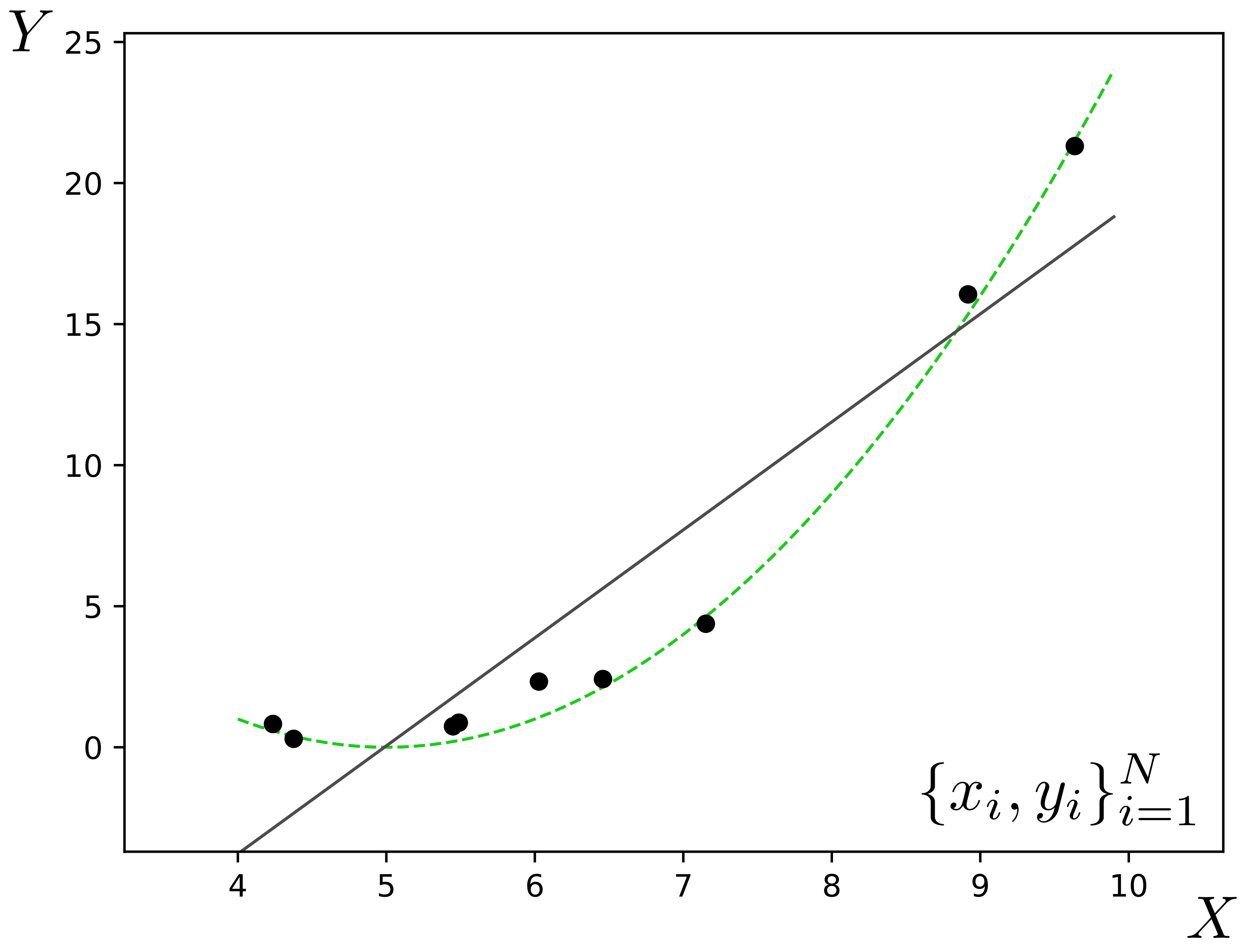


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 1

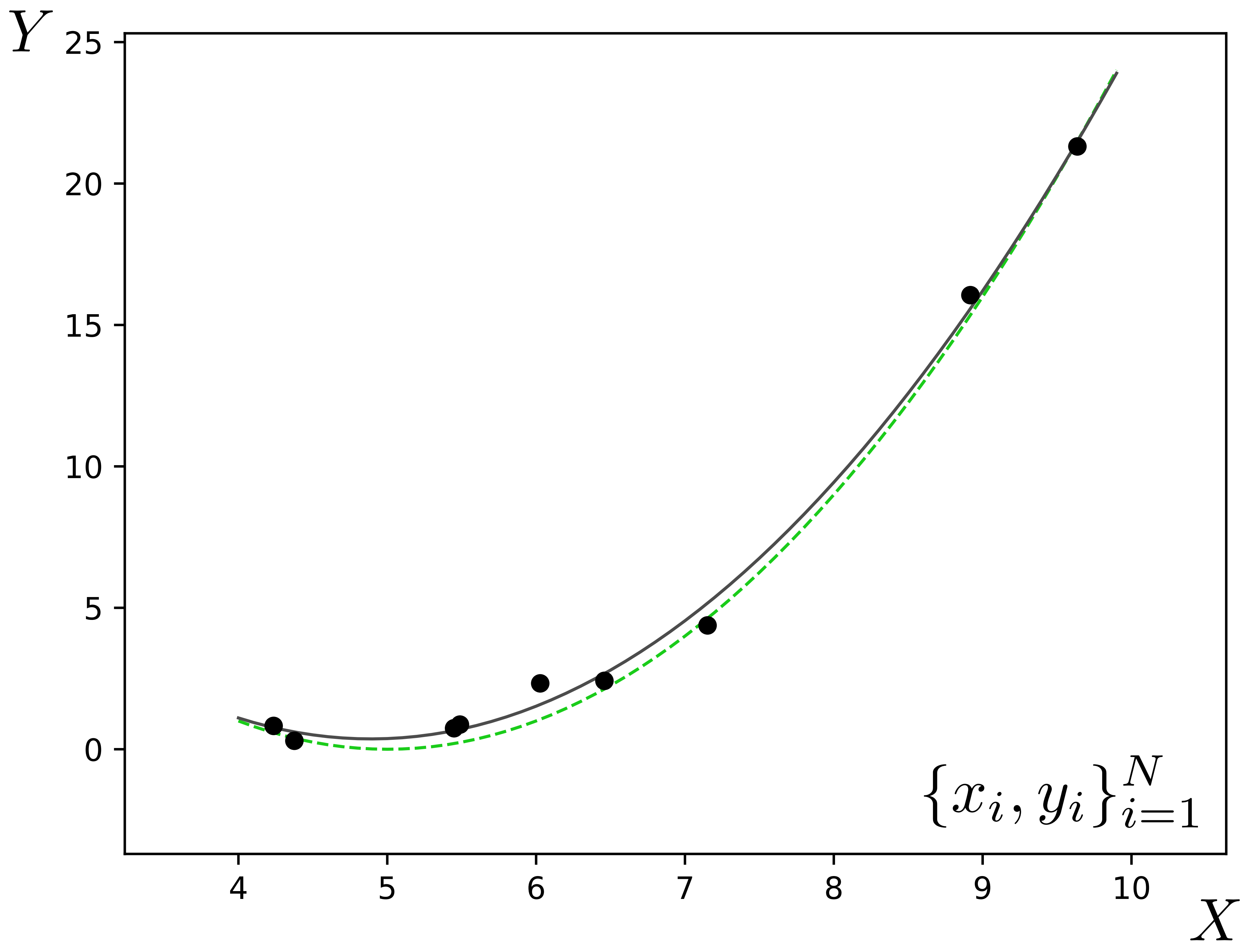


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 2

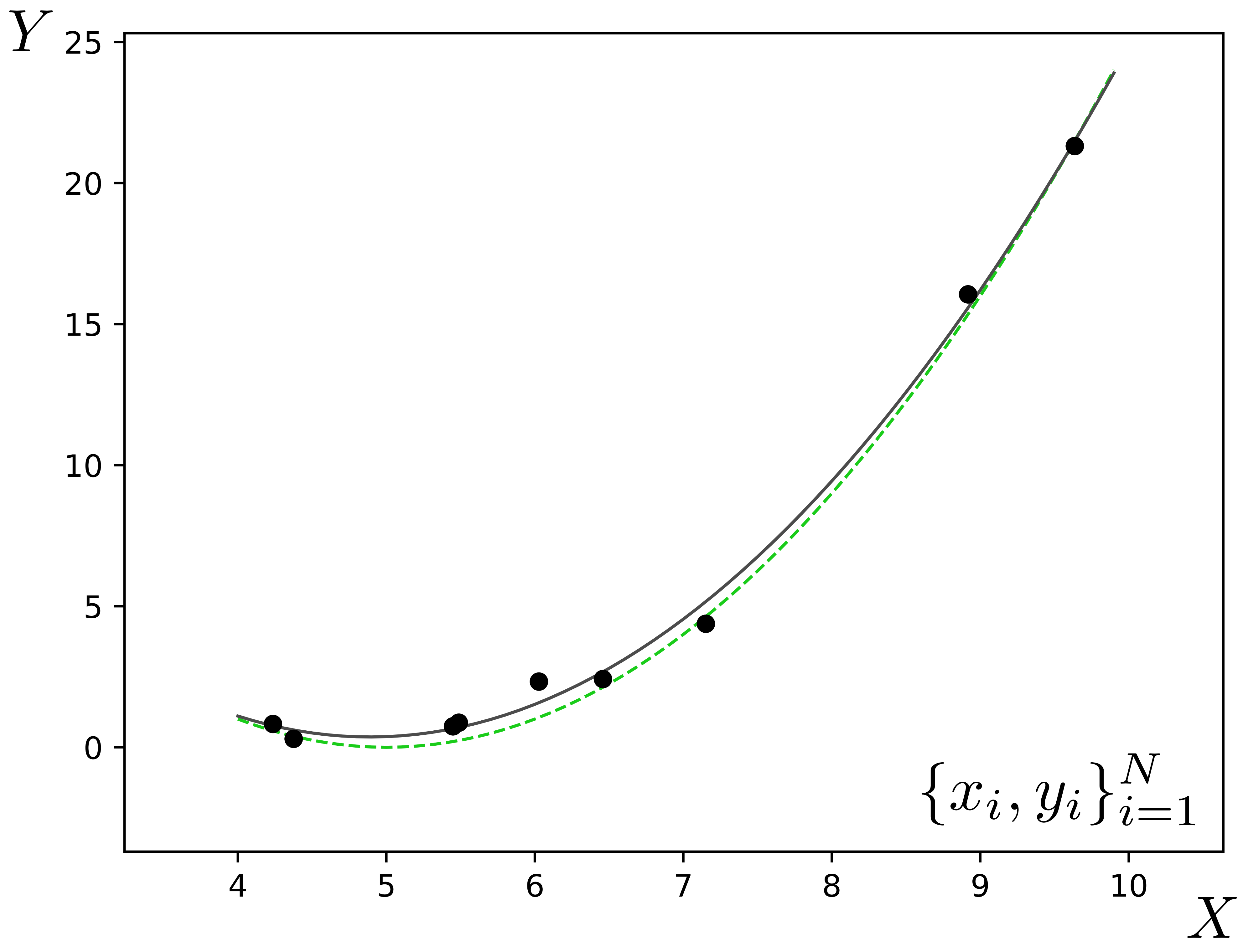


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 3

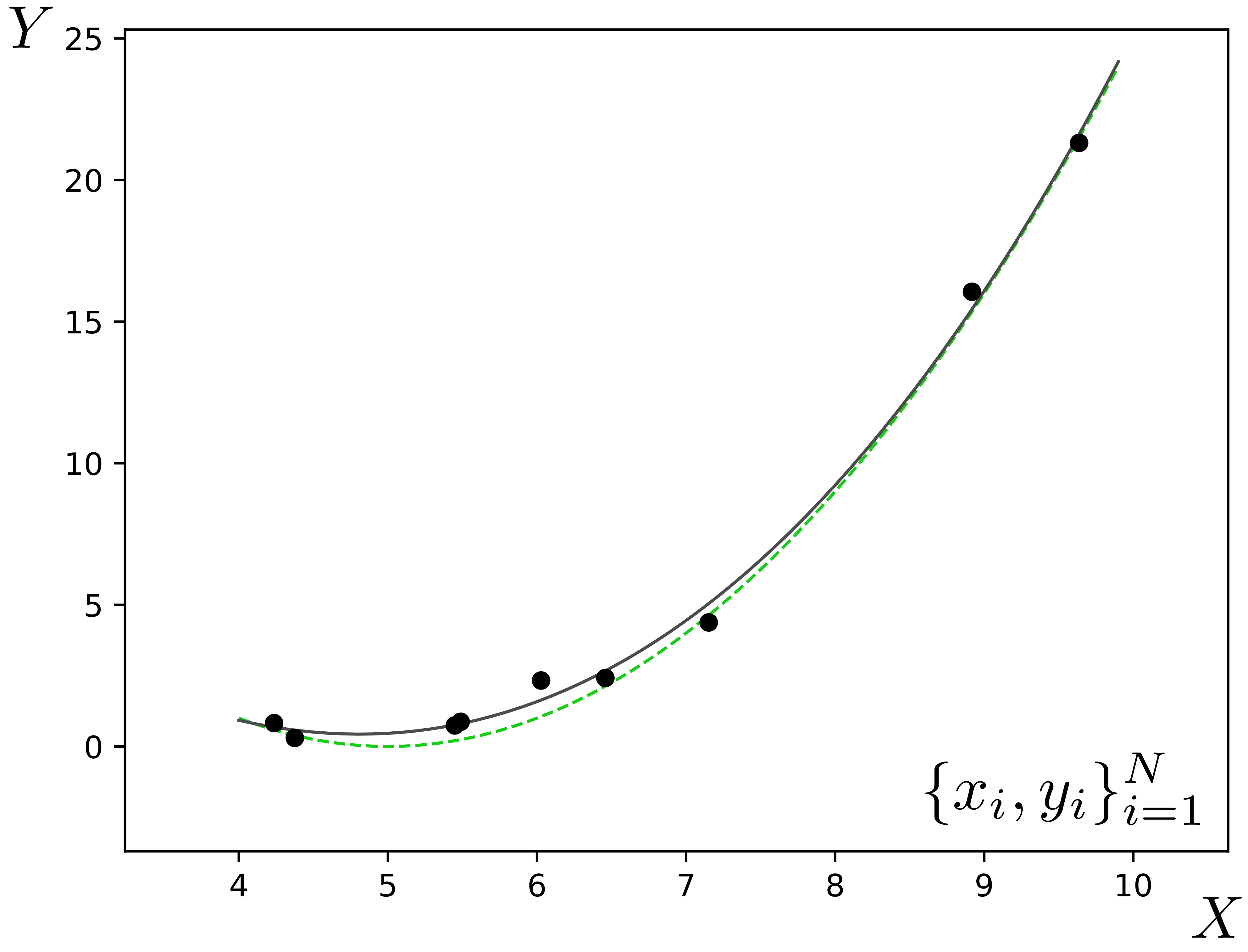


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 4

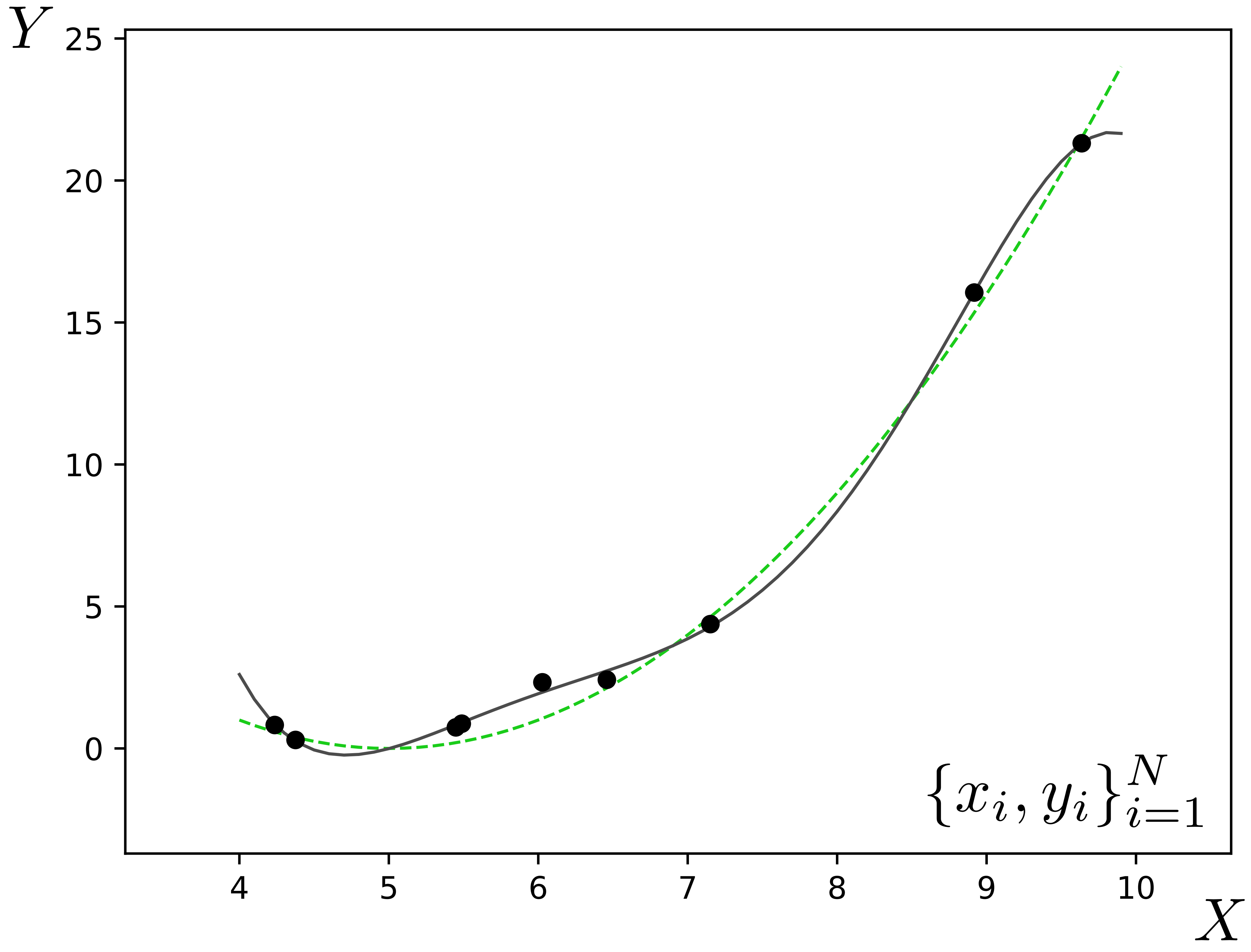


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 5

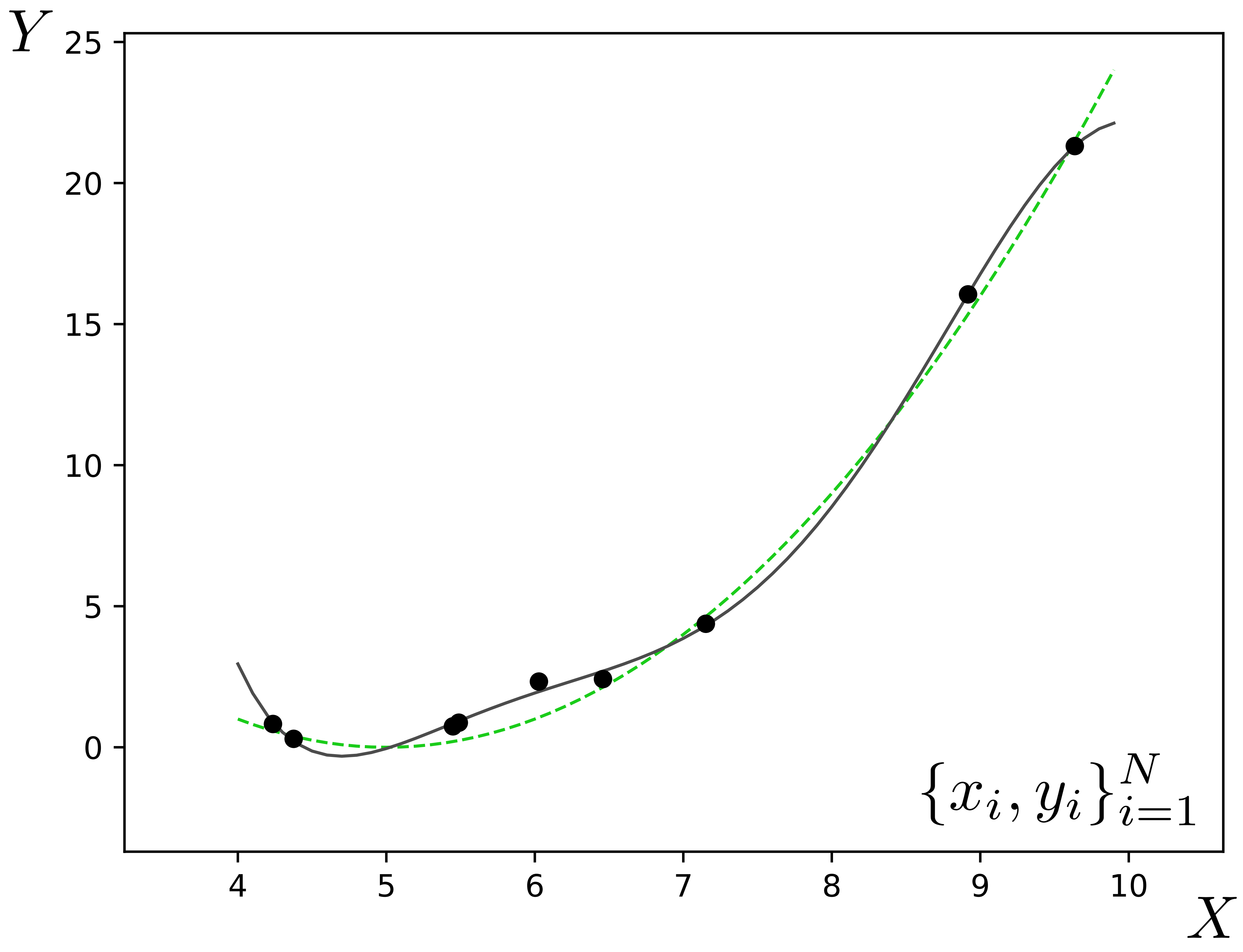


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 6

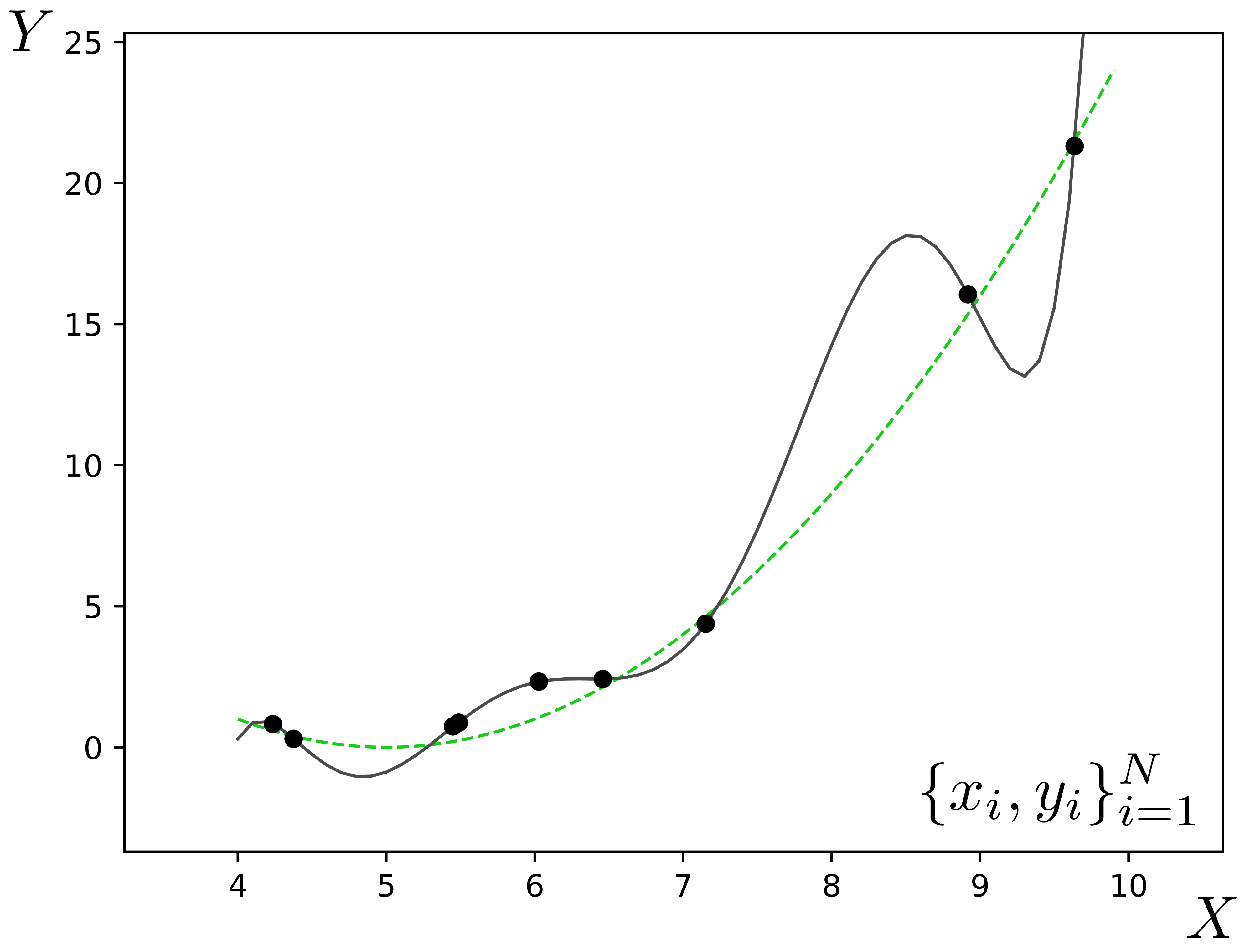


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

$\boxed{\textbf{K = 7}}$

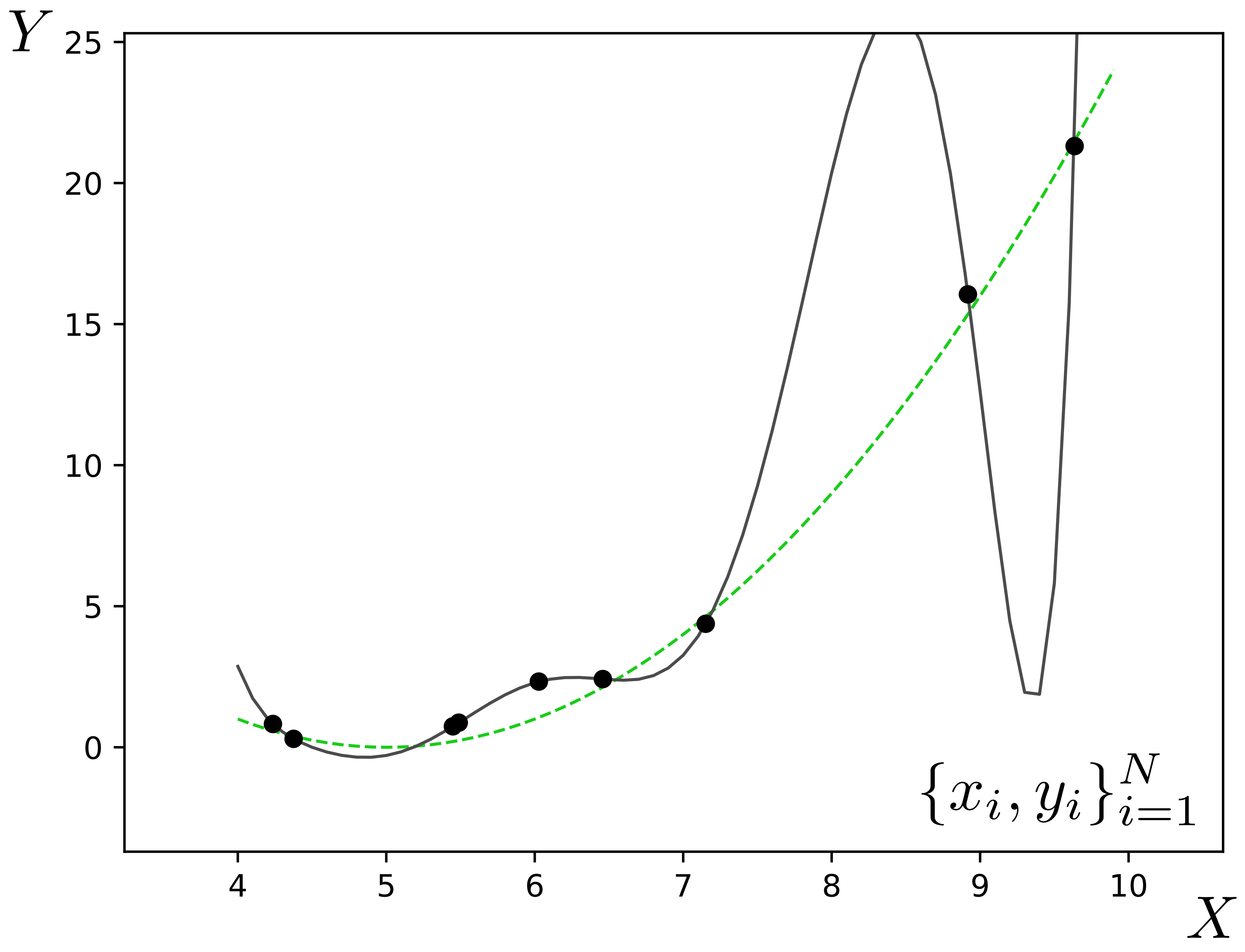


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 8

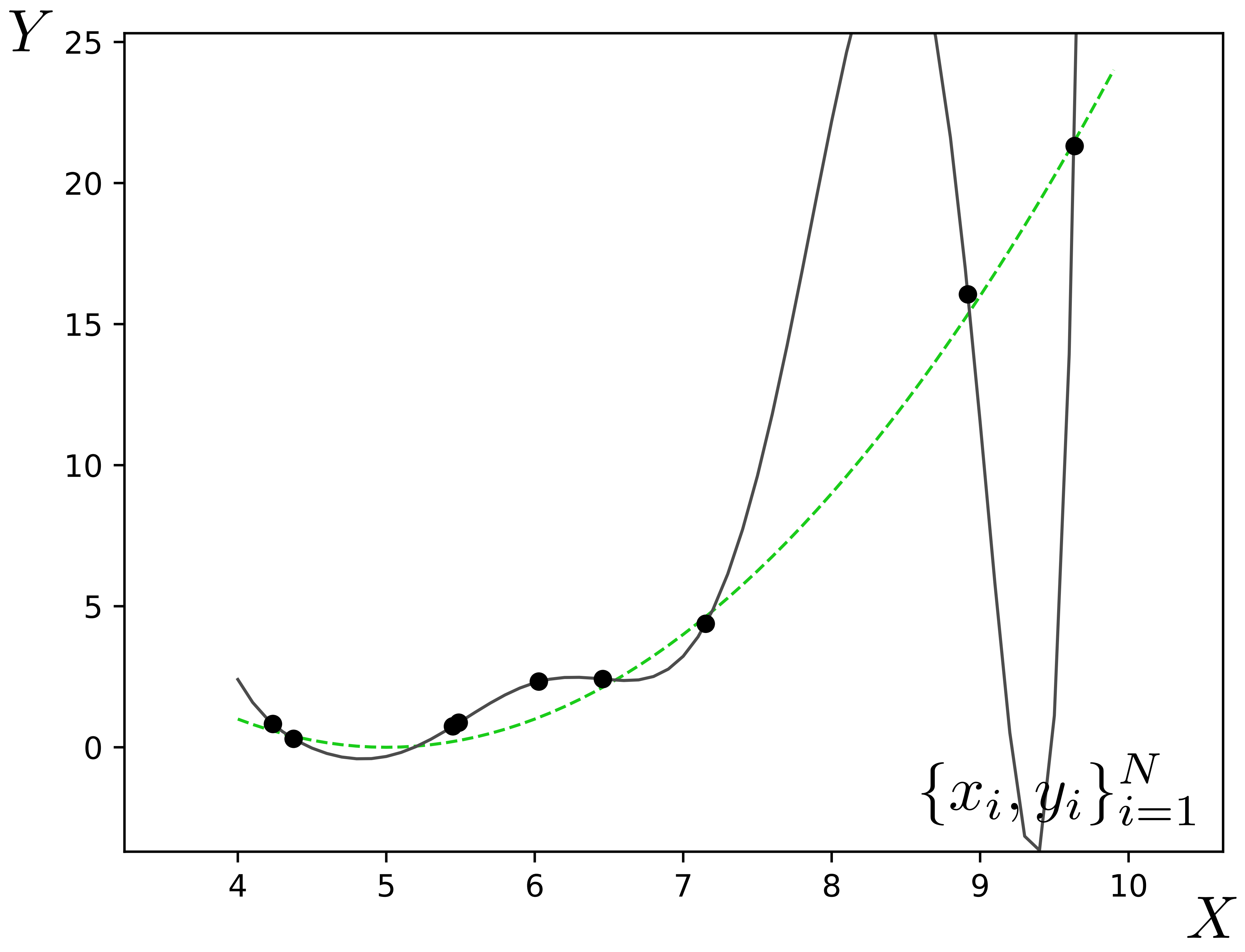


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

$\boxed{\mathbf{K = 9}}$

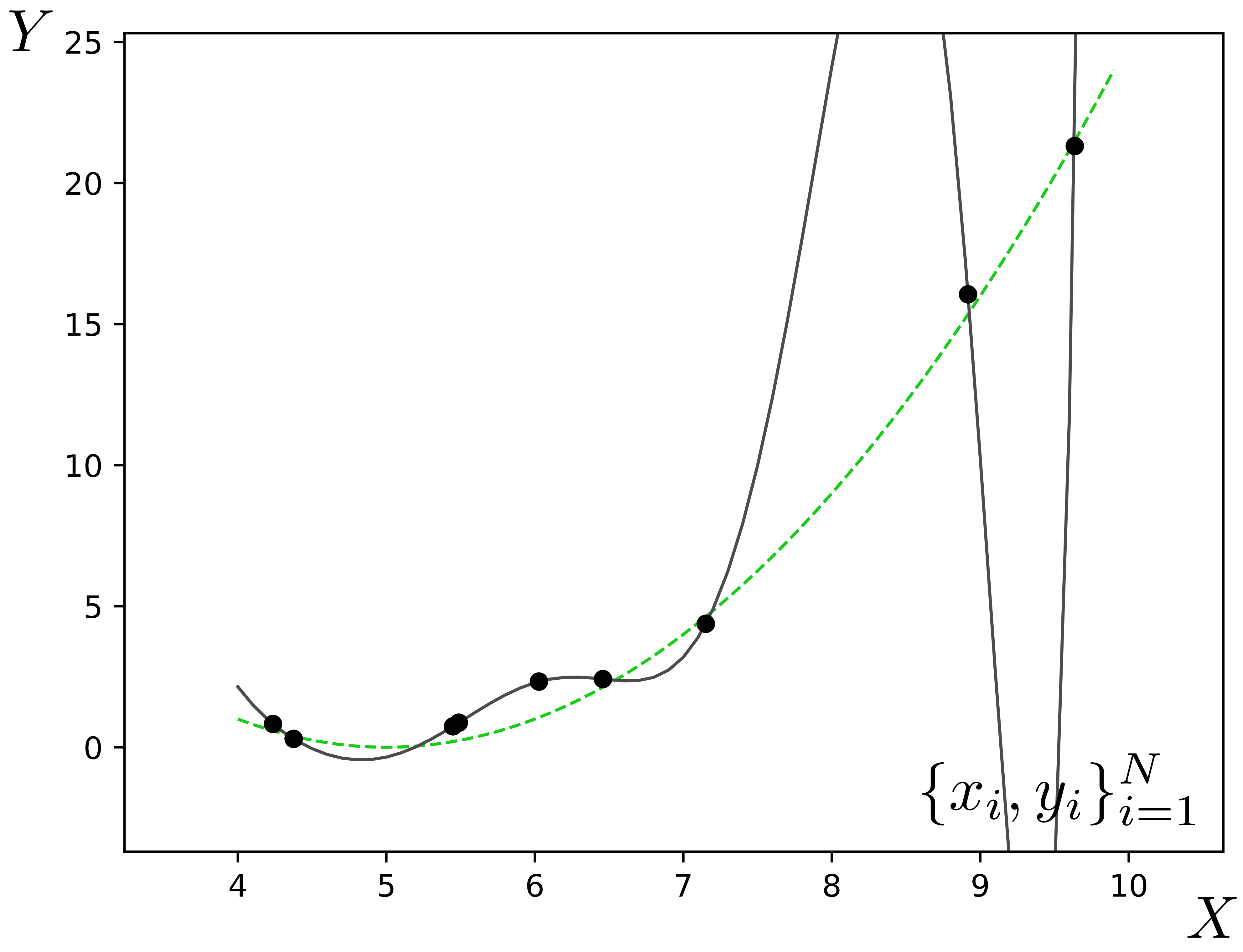


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 10



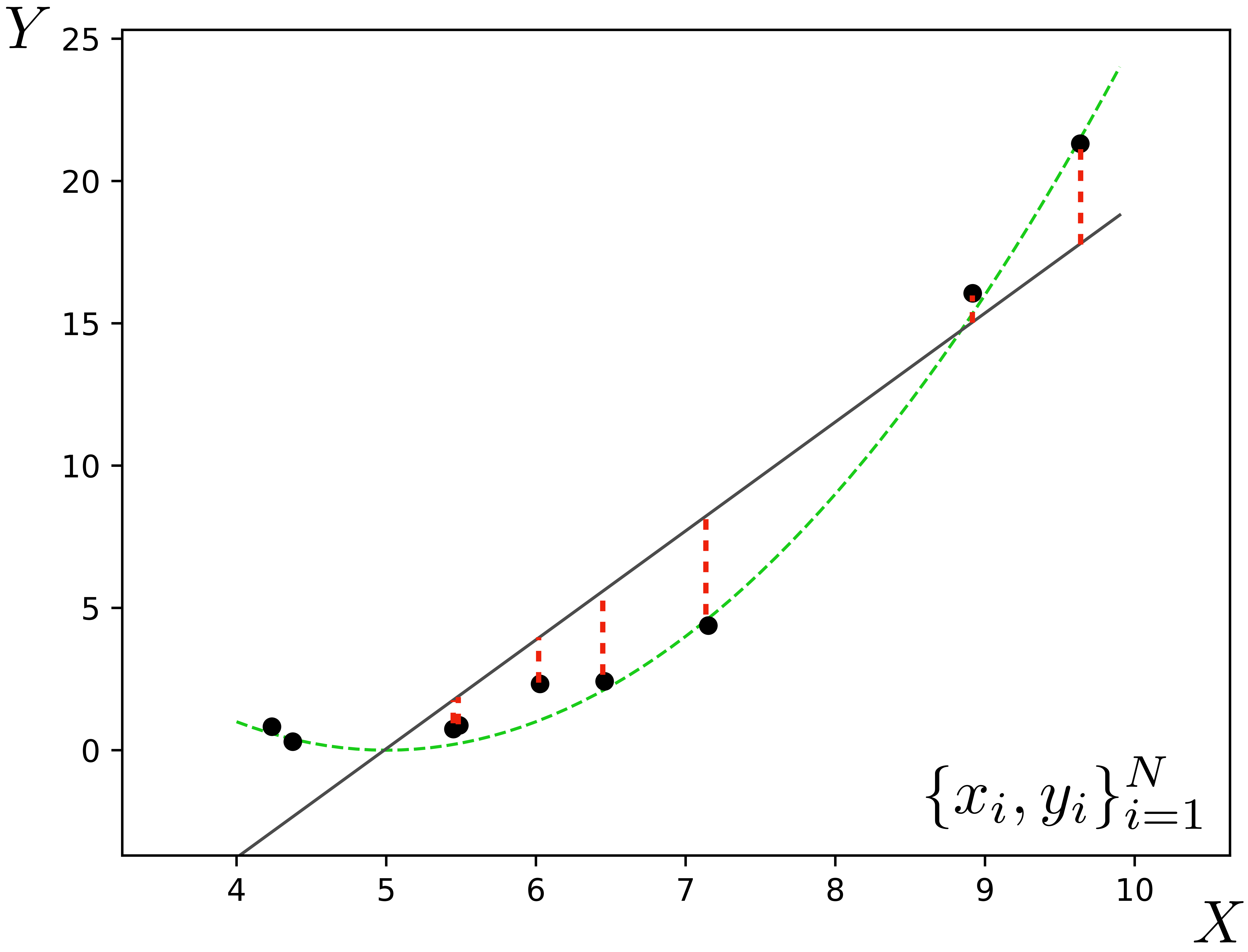
$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

This phenomenon is called **overfitting**.

It occurs when we have too high **capacity** a model, e.g., too many free parameters, too few data points to pin these parameters down.

Source: Isola, Torralba, Freeman

$\boxed{K = 1}$

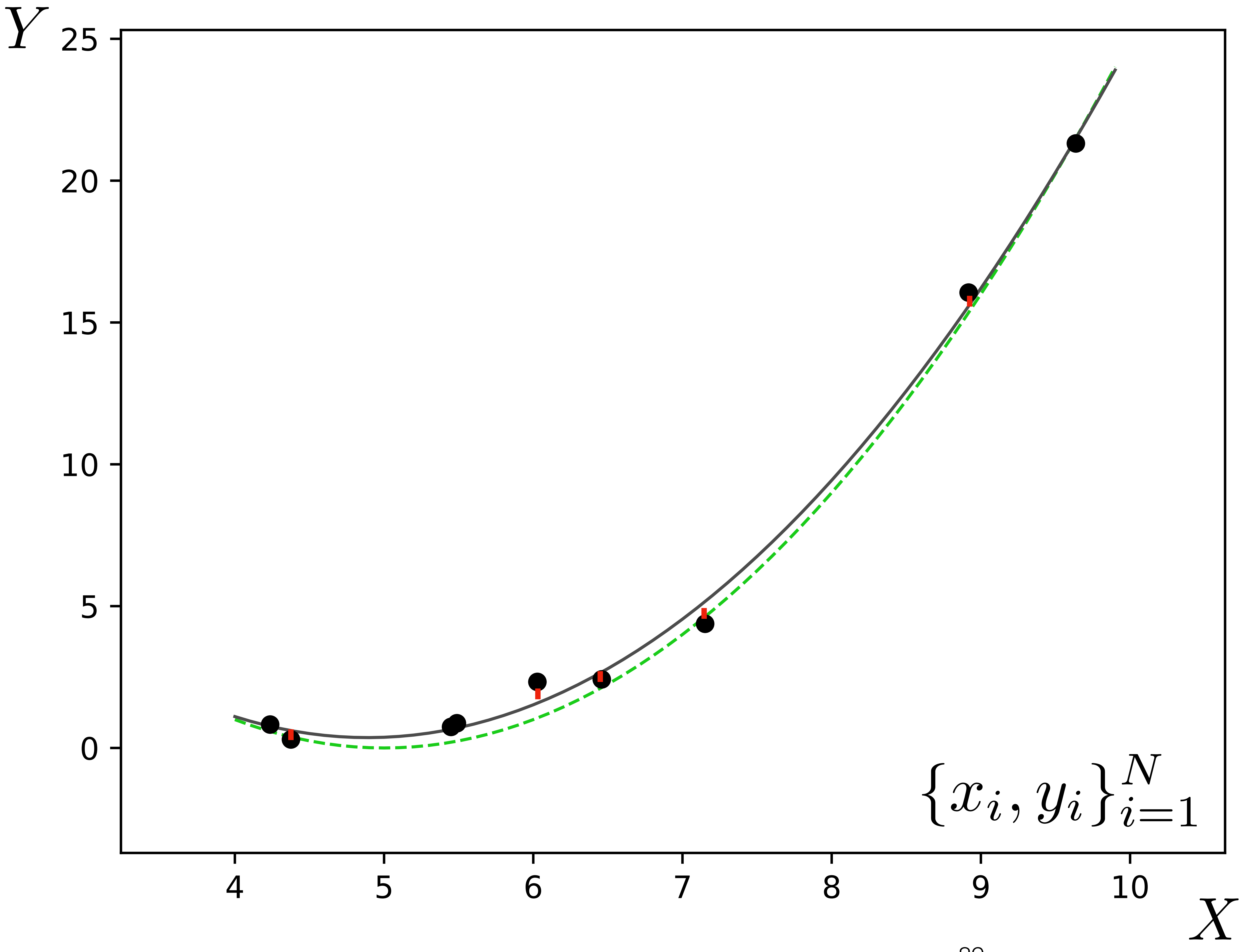


When the model does not have the capacity to capture the true function, we call this **underfitting**.

An underfit model will have high error on the training points. This error is known as **approximation error**.

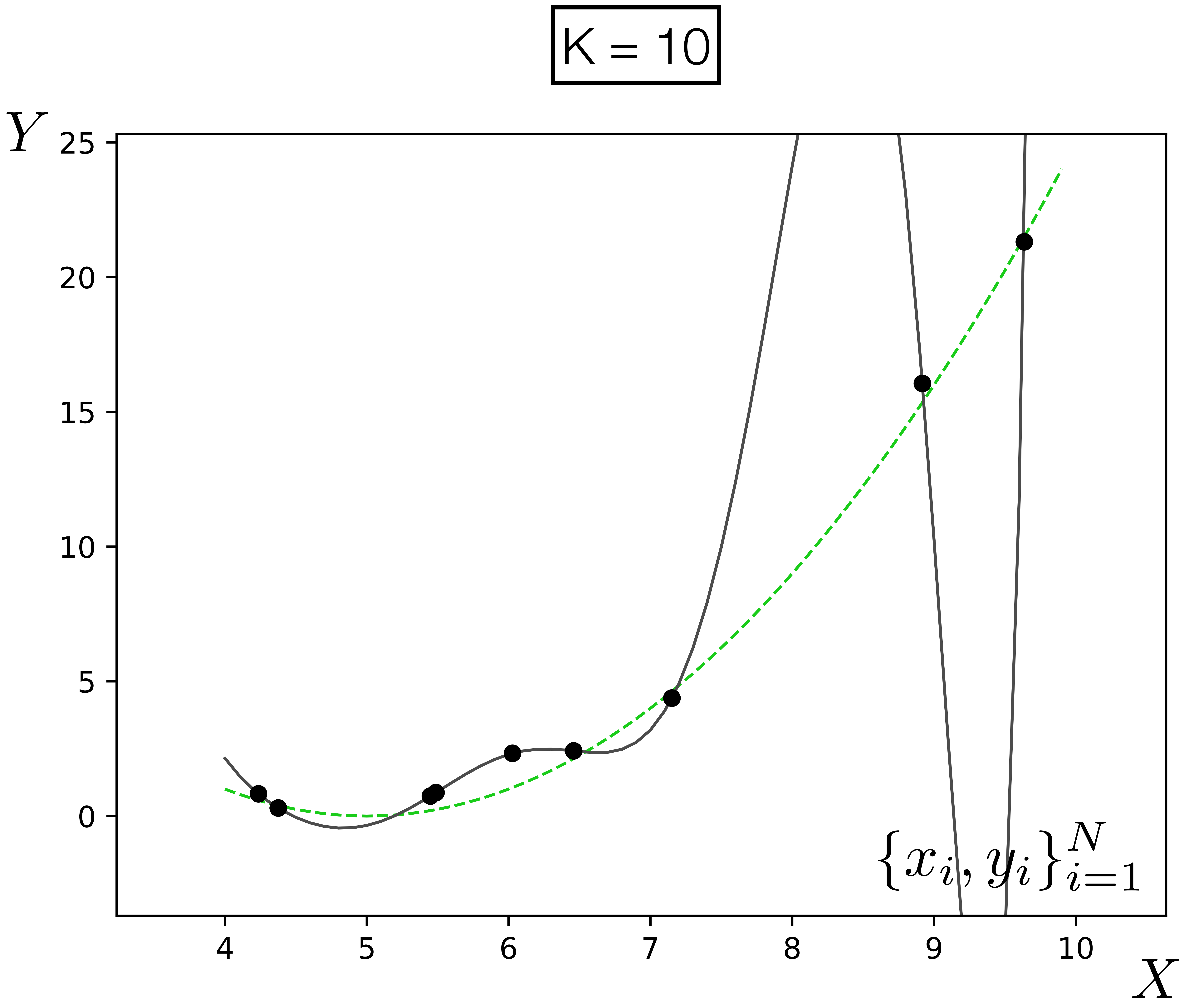
Source: Isola, Torralba, Freeman

$\boxed{K = 2}$



The true function is a quadratic, so a quadratic model (K=2) fits well.

Source: Isola, Torralba, Freeman



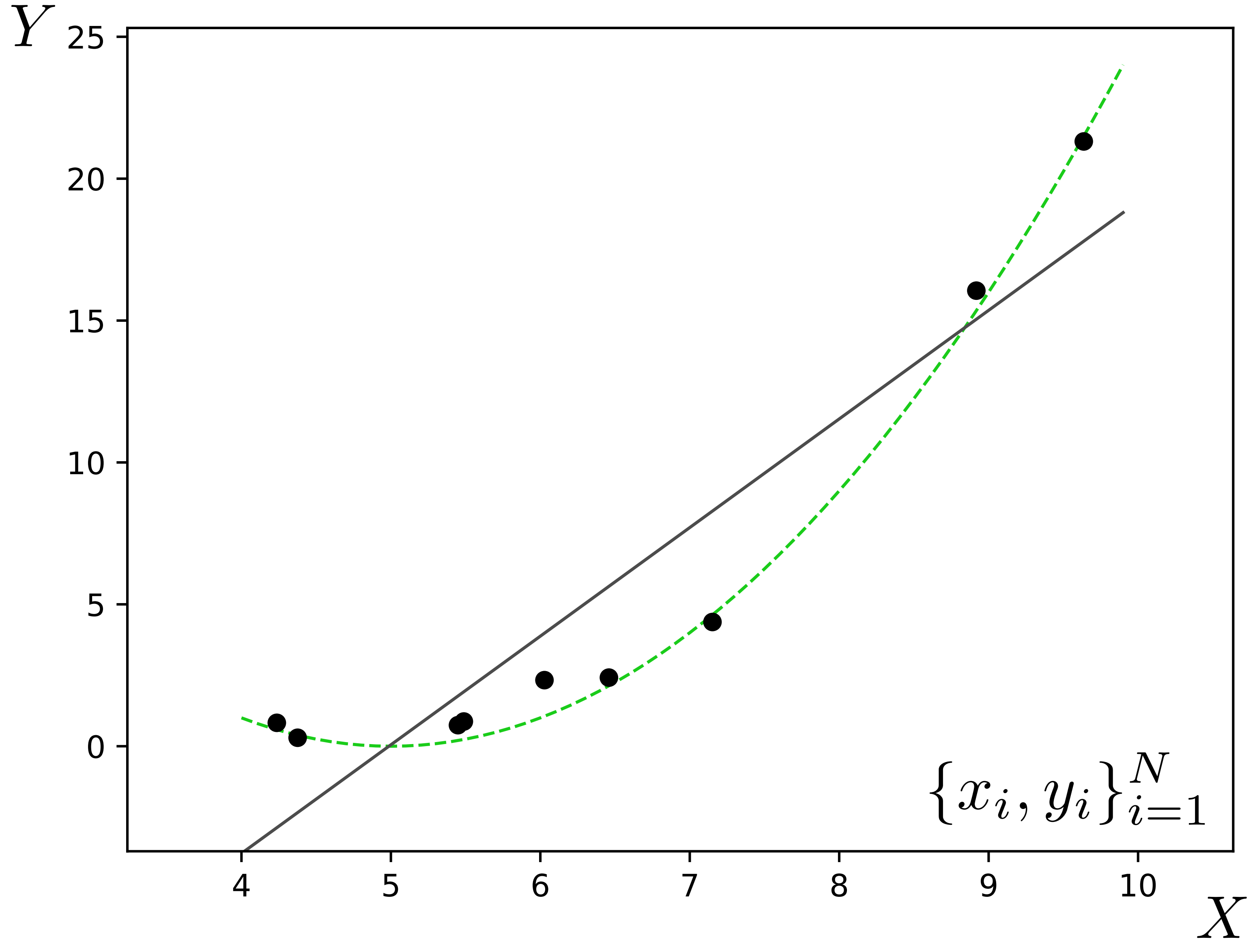
Now we have zero approximation error — the curve passes exactly through each training point.

But we have high **generalization error**, reflected in the gap between the true function and the fit line. We want to do well on *novel* queries, which will be sampled from the green curve (plus noise).

Source: Isola, Torralba, Freeman

Underfitting

$\boxed{K = 1}$

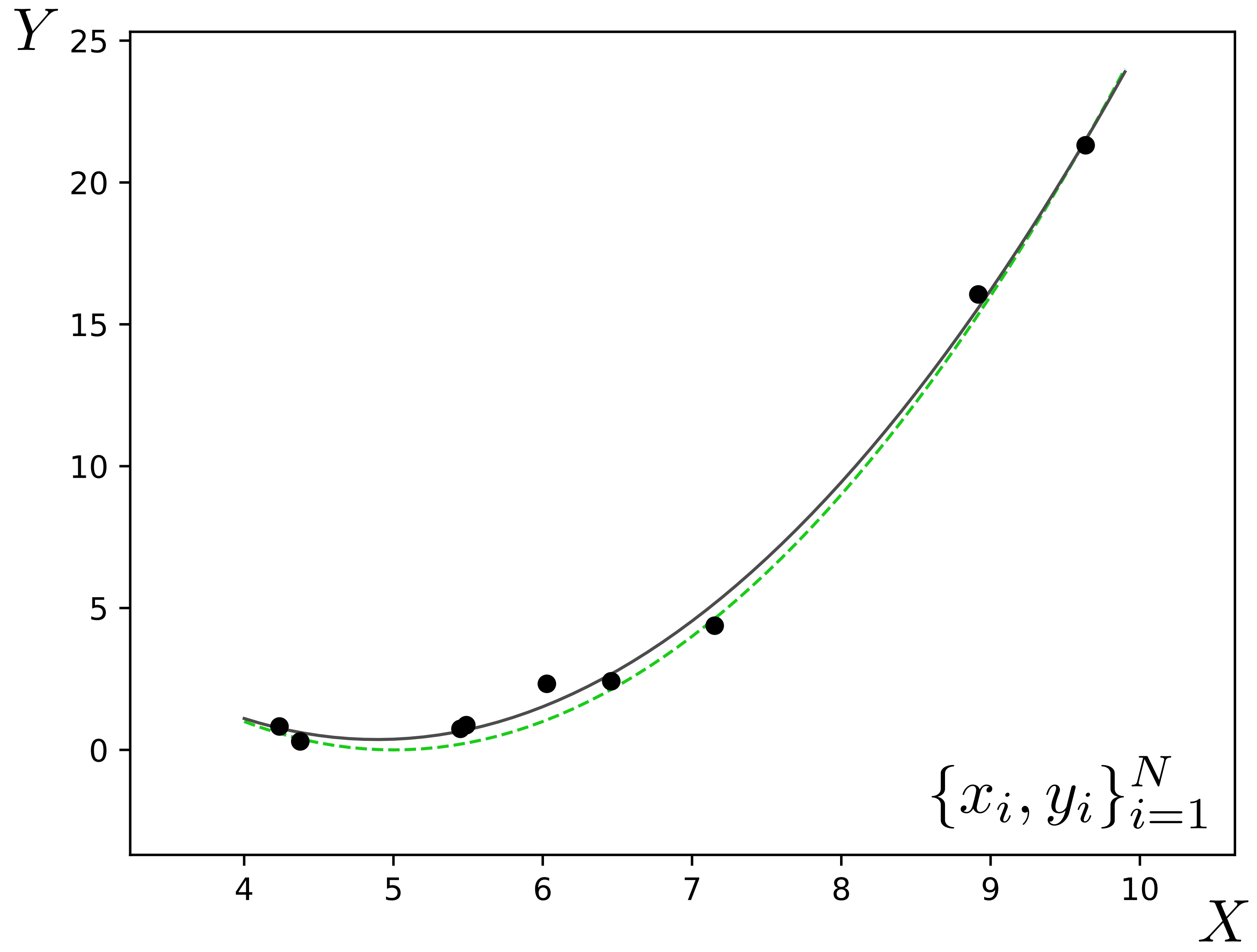


High error on train set

High error on test set

Appropriate model

$\boxed{K = 2}$

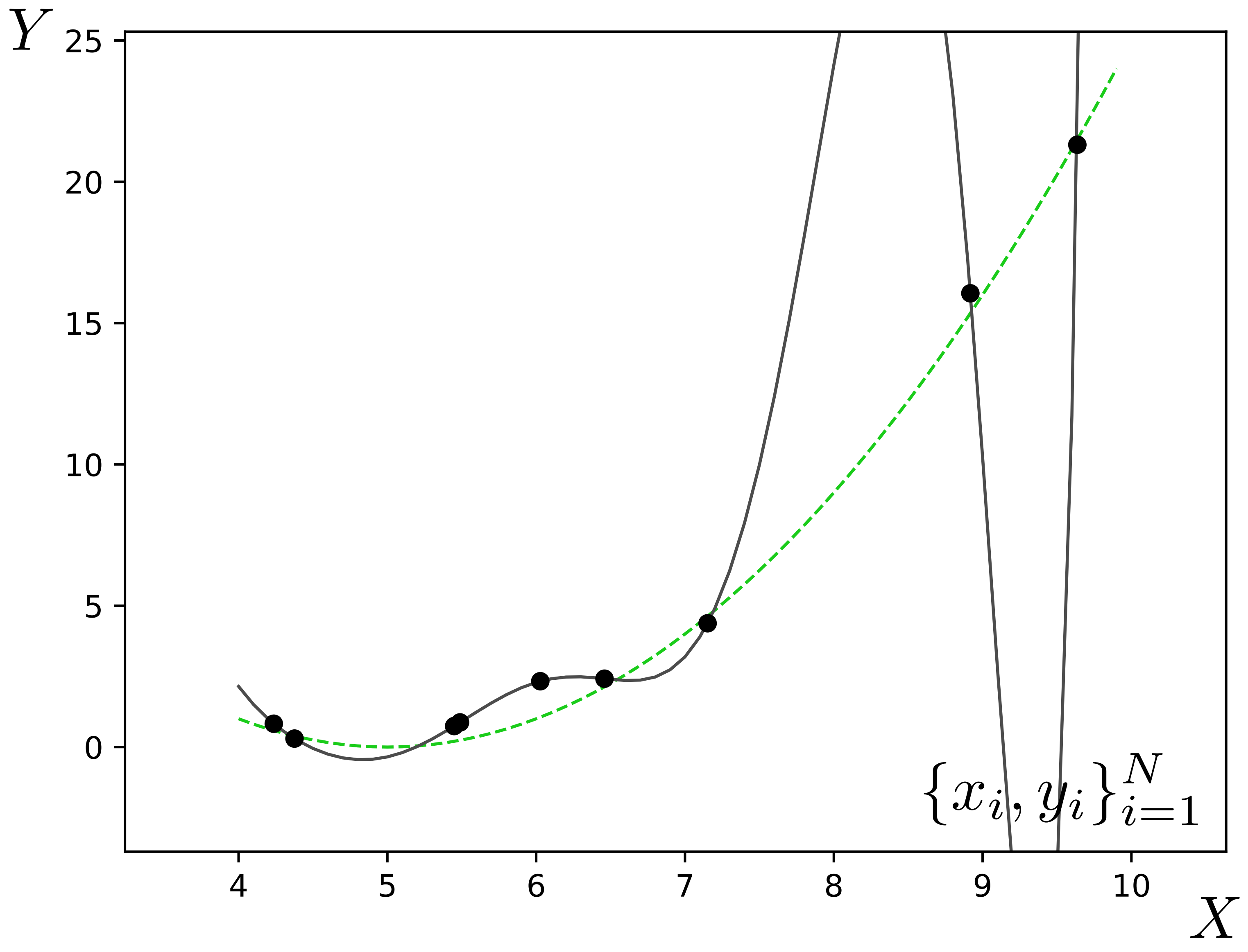


Low error on train set

Low error on test set

Overfitting

$\boxed{K = 10}$



Lowest error on train set

High error on test set

Source: Isola, Torralba, Freeman

We need to control the **capacity** of the model (e.g., use the appropriate number of free parameters).

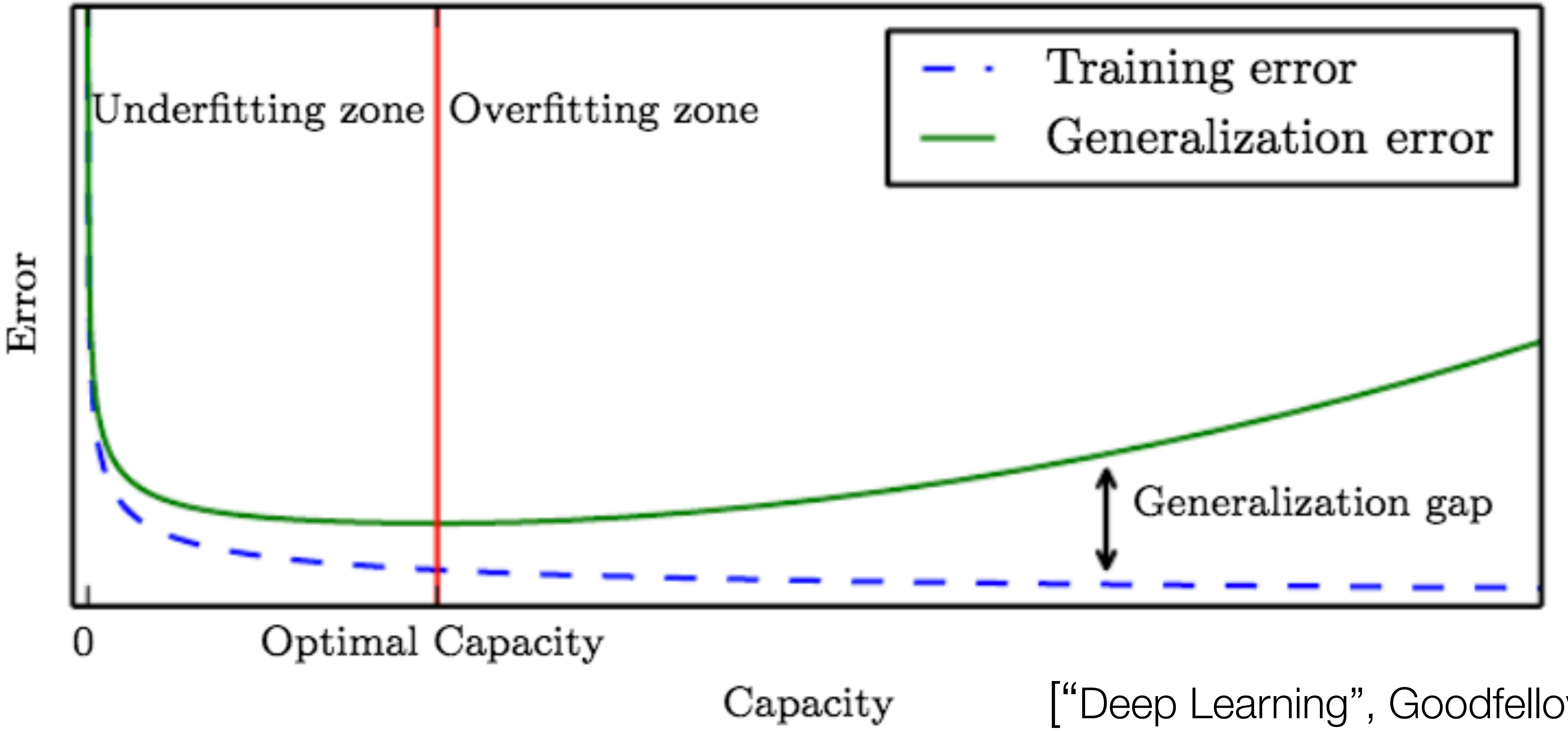
The capacity may be defined as the number of hypotheses under consideration in the hypothesis space.

Complex models with many free parameters have high capacity.

Simple models have low capacity.

Source: Isola, Torralba, Freeman

Training error versus generalization error

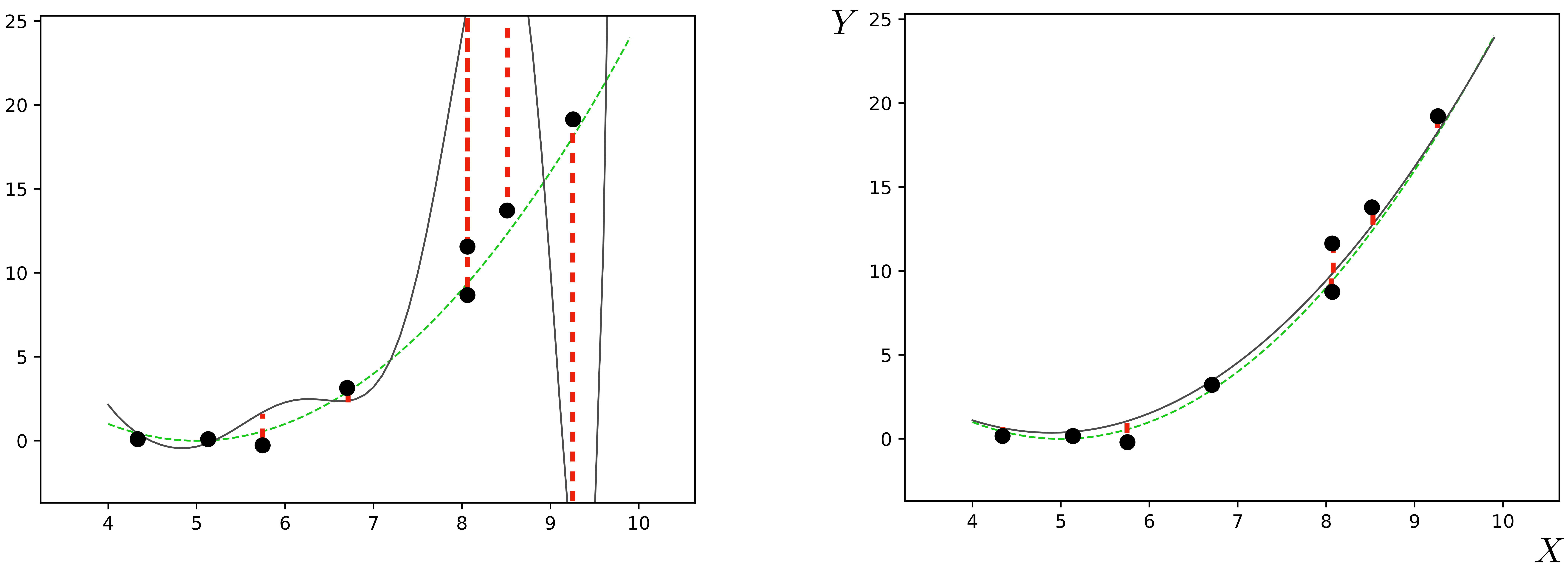


["Deep Learning", Goodfellow et al.]

Source: Isola, Torralba, Freeman

How do we know if we are underfitting or overfitting?

Validation data $\{x_i^{(\texttt{val})}, y_i^{(\texttt{val})}\}$



Cross validation: measure prediction error on validation set

Source: Isola, Torralba, Freeman

Fitting a model

Underfitting?

1. add more parameters (more features, more layers, etc.)

Overfitting?

1. remove parameters
2. add **regularizers**

Source: Isola, Torralba, Freeman

Regularization

Empirical risk minimization:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), \mathbf{y}_i) + R(\theta)$$

Source: Isola, Torralba, Freeman

Regularized least squares

$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

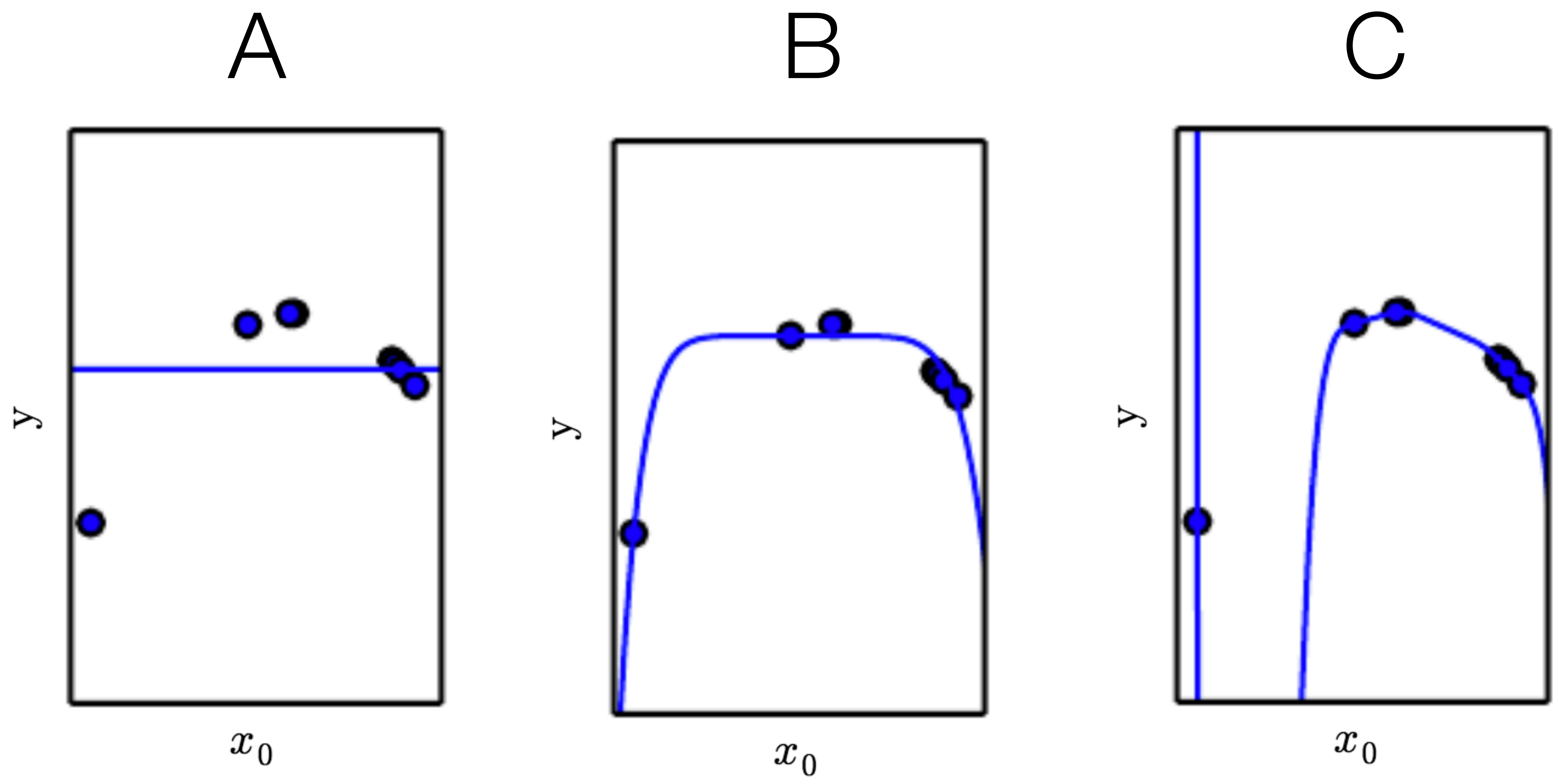
$$R(\theta) = \lambda \left\|\theta\right\|_2^2 \longleftarrow$$

Only use polynomial terms if you really need them! Most terms should be zero

ridge regression, a.k.a., **Tikhonov regularization**

Source: Isola, Torralba, Freeman

$$\theta^* = \arg\min_{\theta} \sum_{I=1}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), \mathbf{y}_i) + \lambda \|\theta\|_2^2$$



Low λ — ?

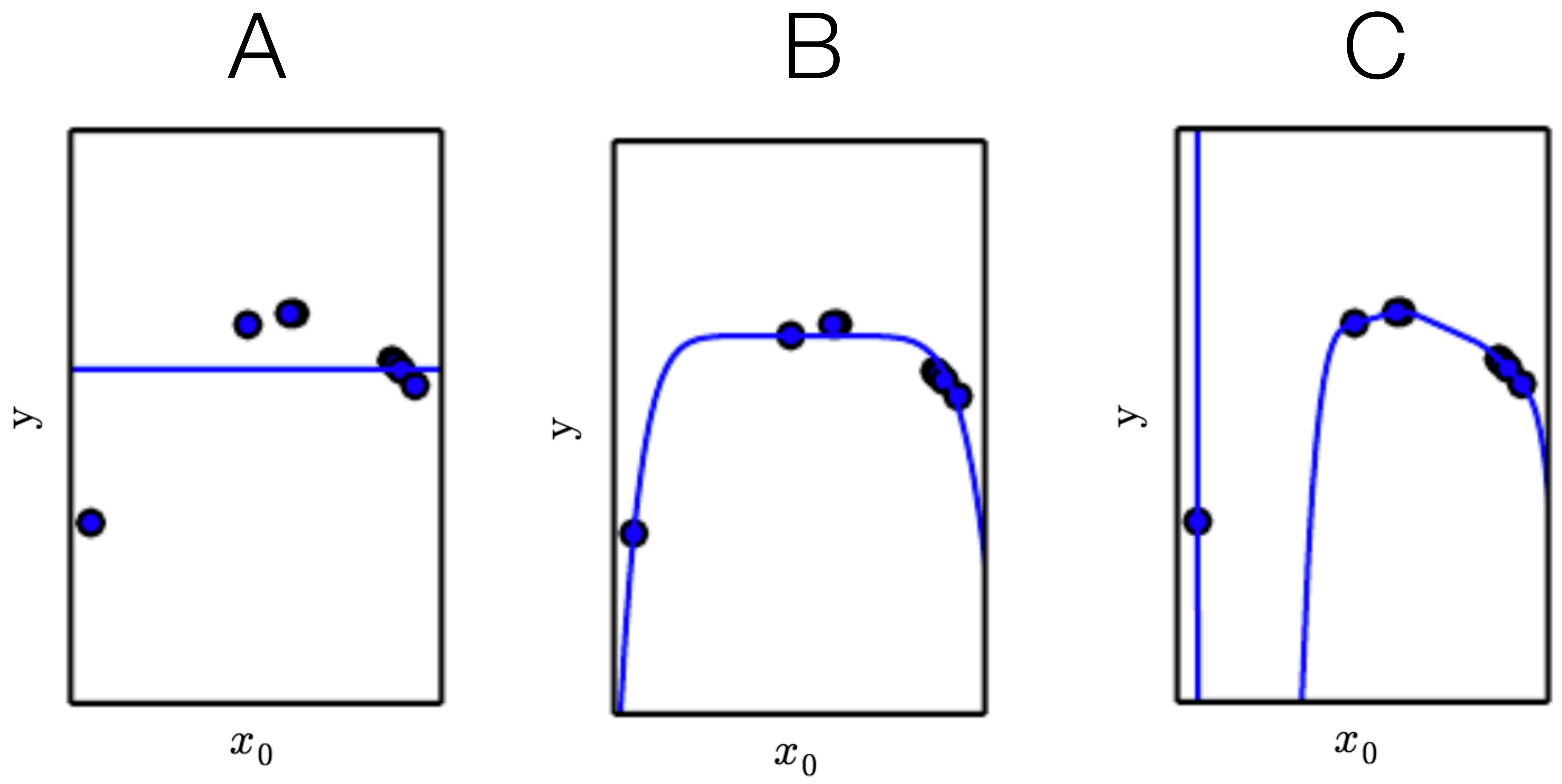
Medium λ — ?

High λ — ?

[Adapted from "Deep Learning", Goodfellow et al.]

Source: Isola, Torralba, Freeman

$$\theta^* = \arg\min_{\theta} \sum_{I=1}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), \mathbf{y}_i) + \lambda \|\theta\|_2^2$$



Low λ — C

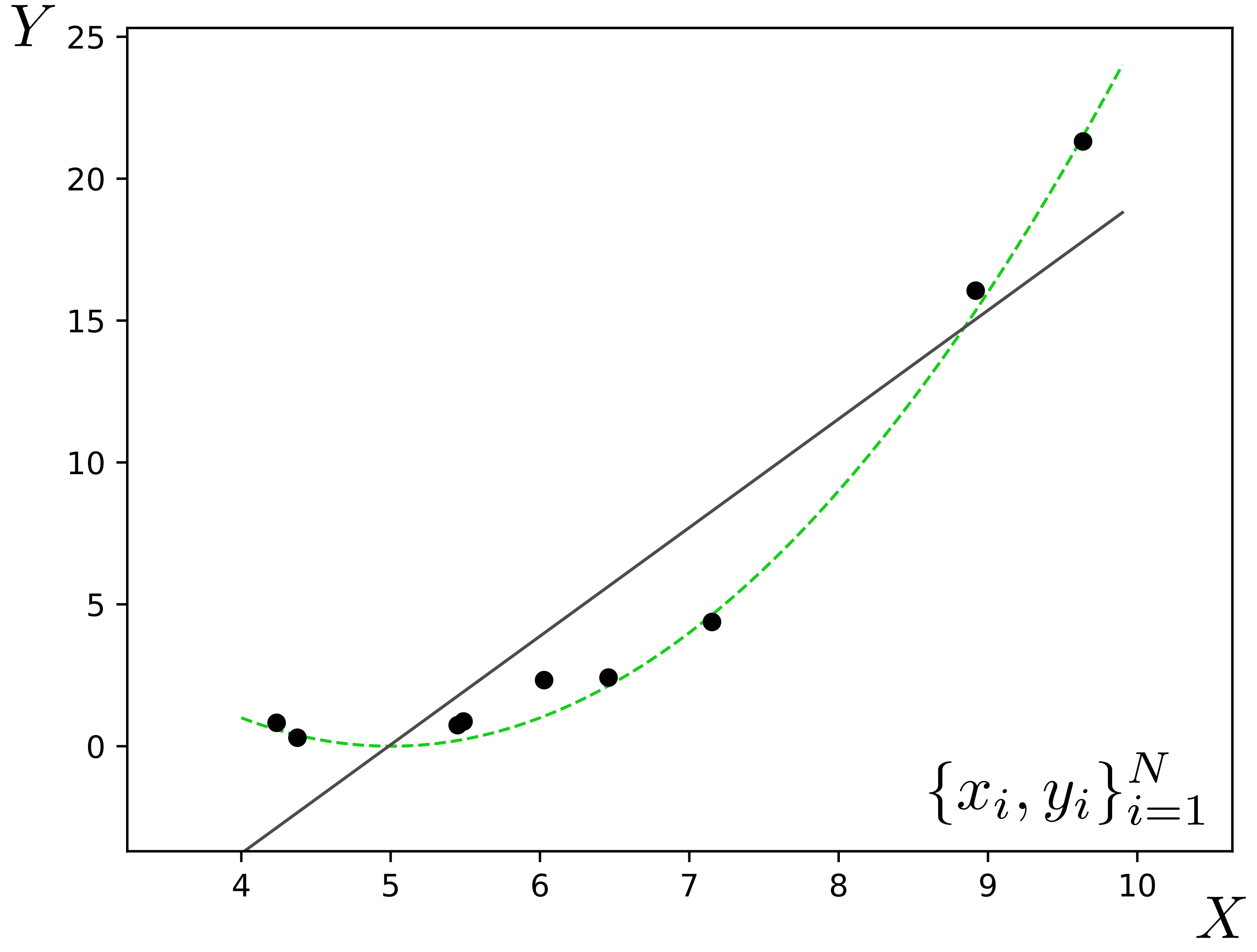
Medium λ — B

High λ — A

[Adapted from “Deep Learning”, Goodfellow et al.]

Underfitting

$\boxed{K = 1}$

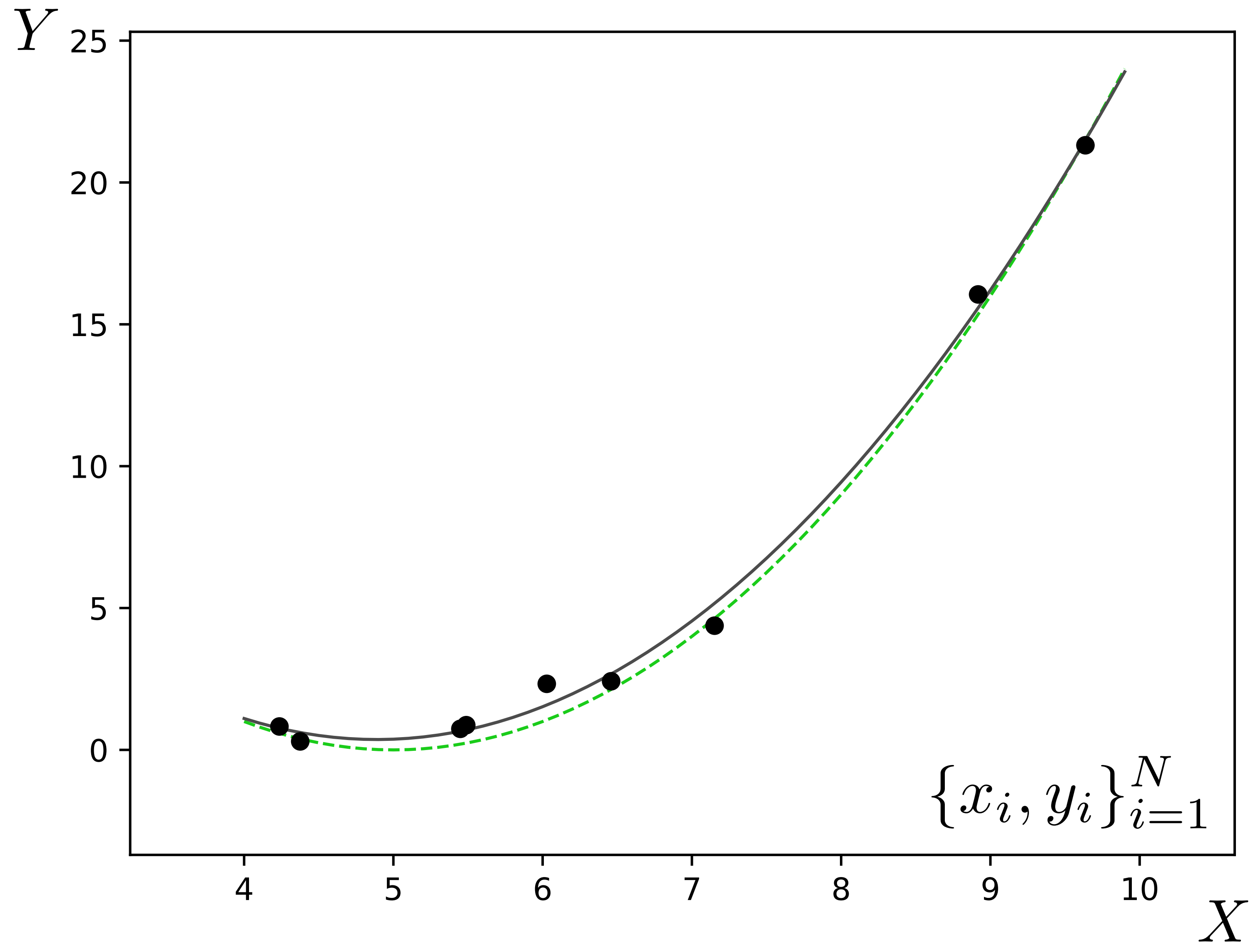


Simple model

Doesn’t fit the training data

Appropriate model

$\boxed{K = 2}$

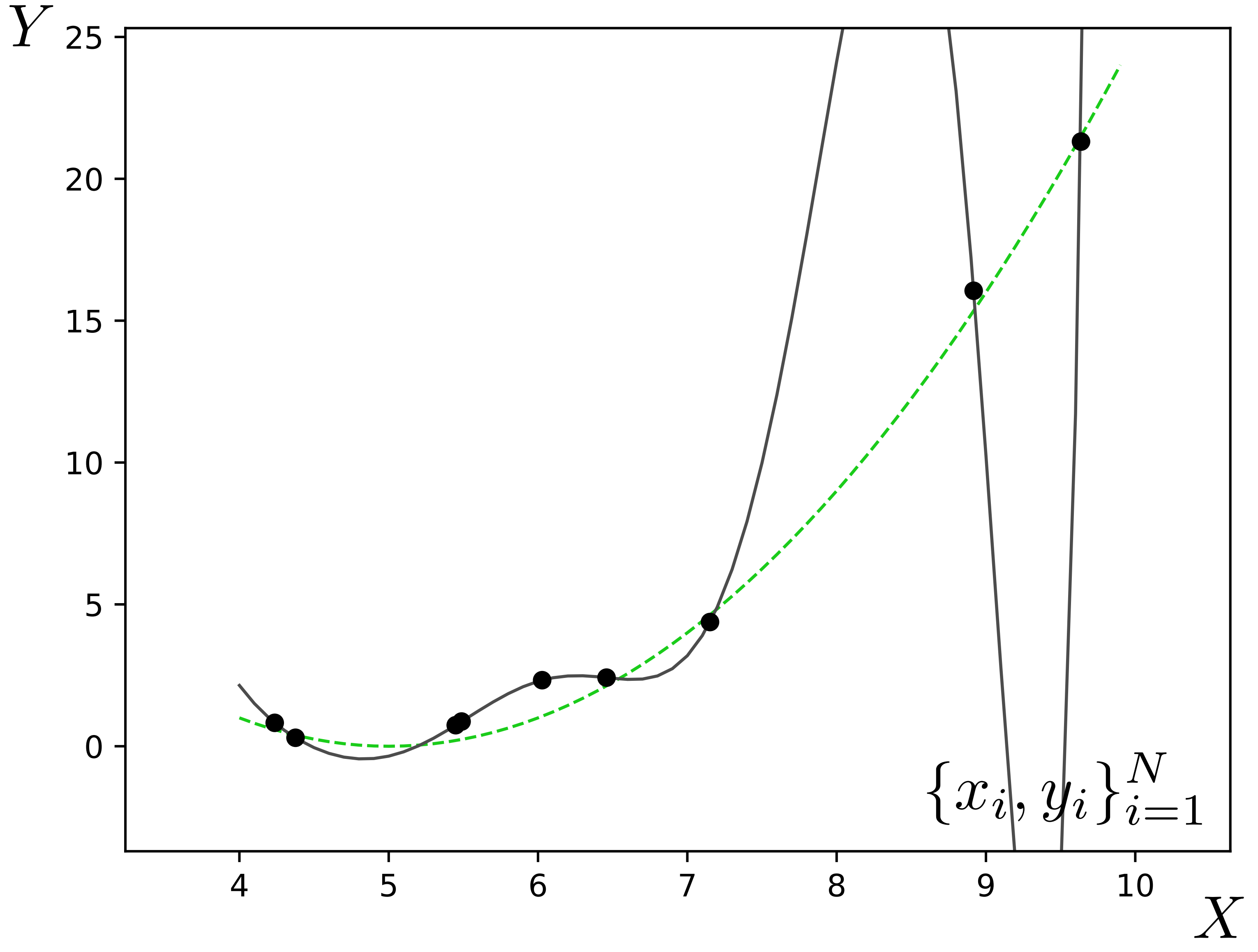


Simple model

Fits the training data

Overfitting

$\boxed{K = 10}$

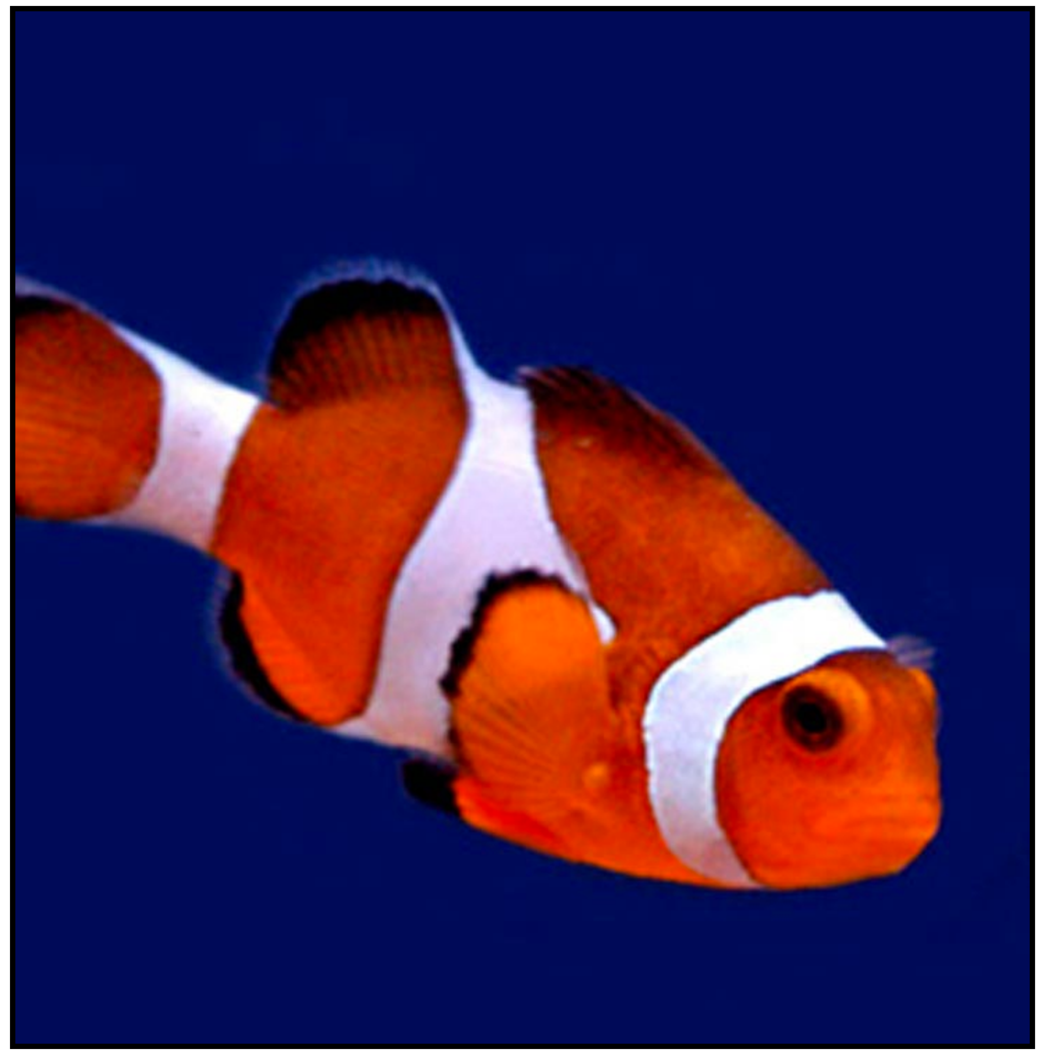


Complex model

Fits the training data

Source: Isola, Torralba, Freeman

Image classification



→ **Classifier** → “Fish”

image **x**

label y

Source: Isola, Torralba, Freeman

Image classification

→ **Classifier** → "Fish"

image **x**

label y

Source: Isola, Torralba, Freeman

Image classification

Classifier

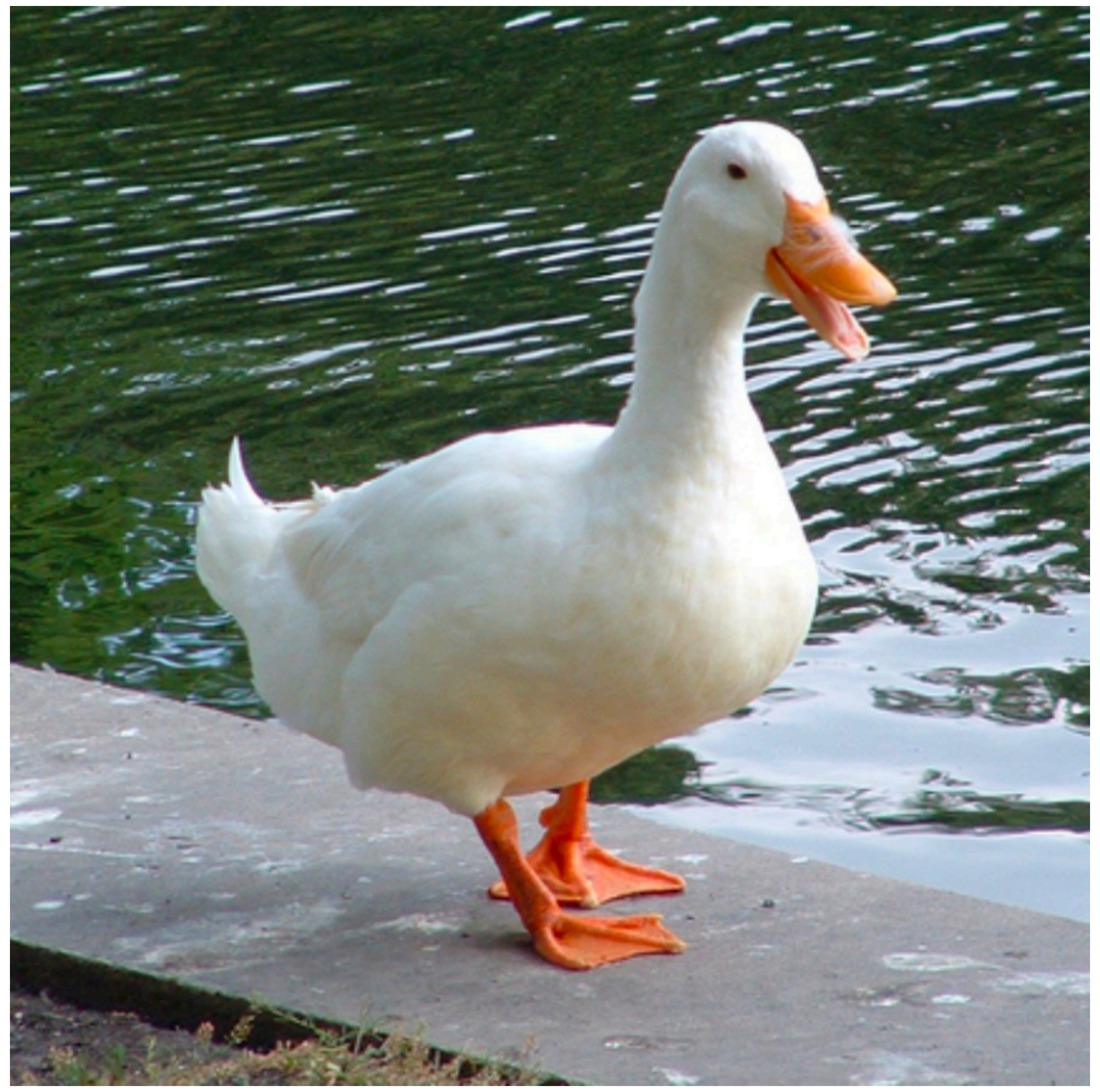
"Fish"

image **x**

label y

Source: Isola, Torralba, Freeman

Image classification



Classifier

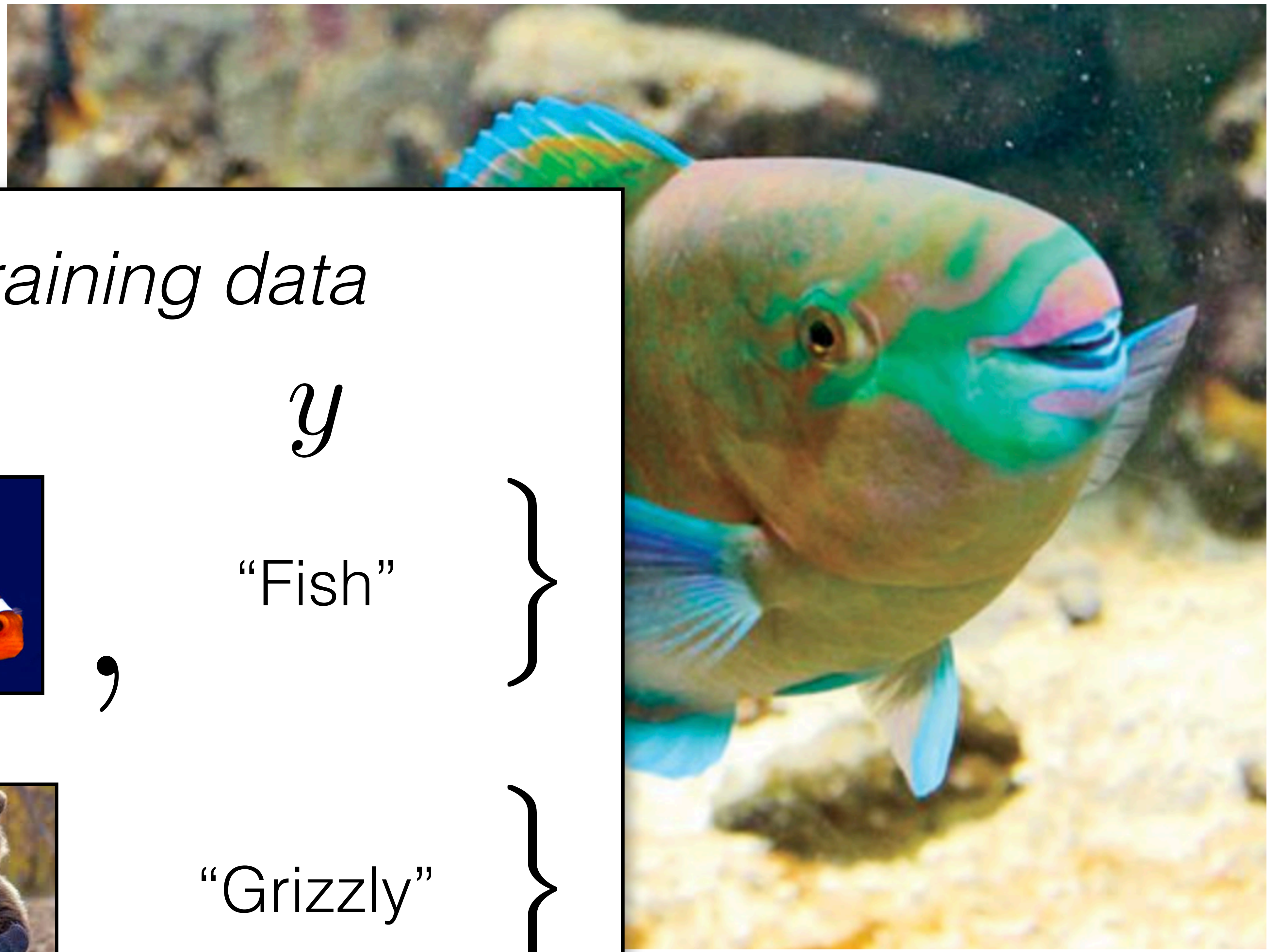
"Duck"

image **x**

label y

Source: Isola, Torralba, Freeman

$\mathbf{x}$



Training data

$\mathbf{x}$ y

$\{$  , "Fish" $\}$

$\{$  , "Grizzly" $\}$

$\{$  , "Chameleon" $\}$

$\vdots$

$\xrightarrow{f}$ y $\boxed{\text{“Fish”}}$

$$\arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), y_i)$$

Source: Isola, Torralba, Freeman

How to represent class labels?

Training data

$\mathbf{x}$ y

{  , "Fish" }

{  , "Grizzly" }

{  , "Chameleon" }

⋮

→

Training data

$\mathbf{x}$ y

{  , 1 }

{  , 2 }

{  , 3 }

⋮

→

One-hot vector

Training data

$\mathbf{x}$ $\mathbf{y}$

{  , [0,0,1] }

{  , [0,1,0] }

{  , [1,0,0] }

⋮

Source: Isola, Torralba, Freeman

Recall: loss function

0-1 loss (number of misclassifications)

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = 1 - \mathbb{1}(\hat{\mathbf{y}}, \mathbf{y}) \longleftarrow$$ number of misclassifications

Least squares (predict 1 for true class, 0 for others)

$$\mathcal{L}(\hat{y}, y) = \sum_{k=1}^{K} (y_k - \hat{y}_k)^2$$

Cross entropy: a good surrogate that we'll be able to optimize:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

Adapted from P. Isola

Ground truth label $\mathbf{y}$

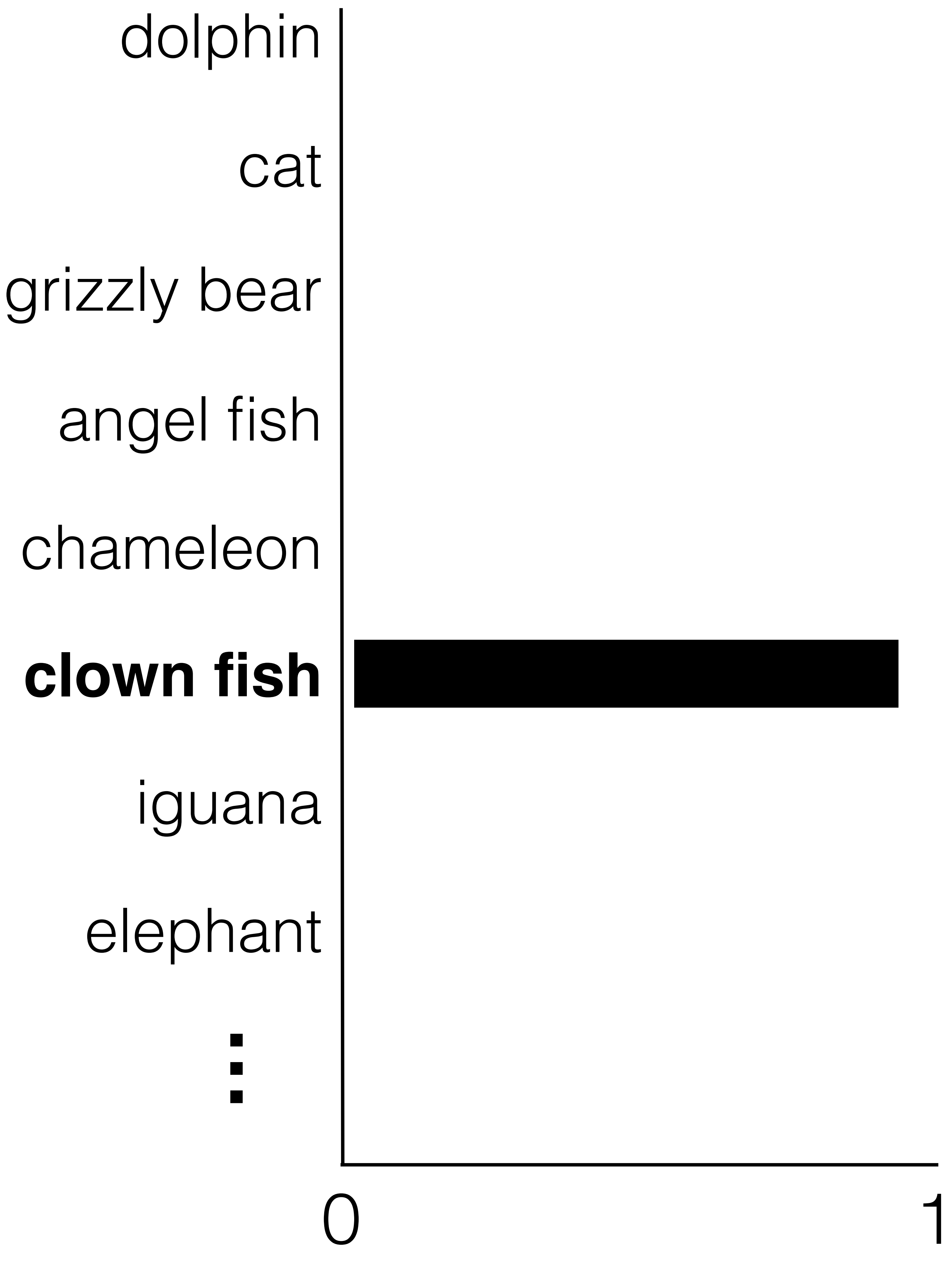
$\mathbf{x}$

[0,0,0,0,0,1,0,0,…]

Source: Isola, Torralba, Freeman

Ground truth label $\mathbf{y}$

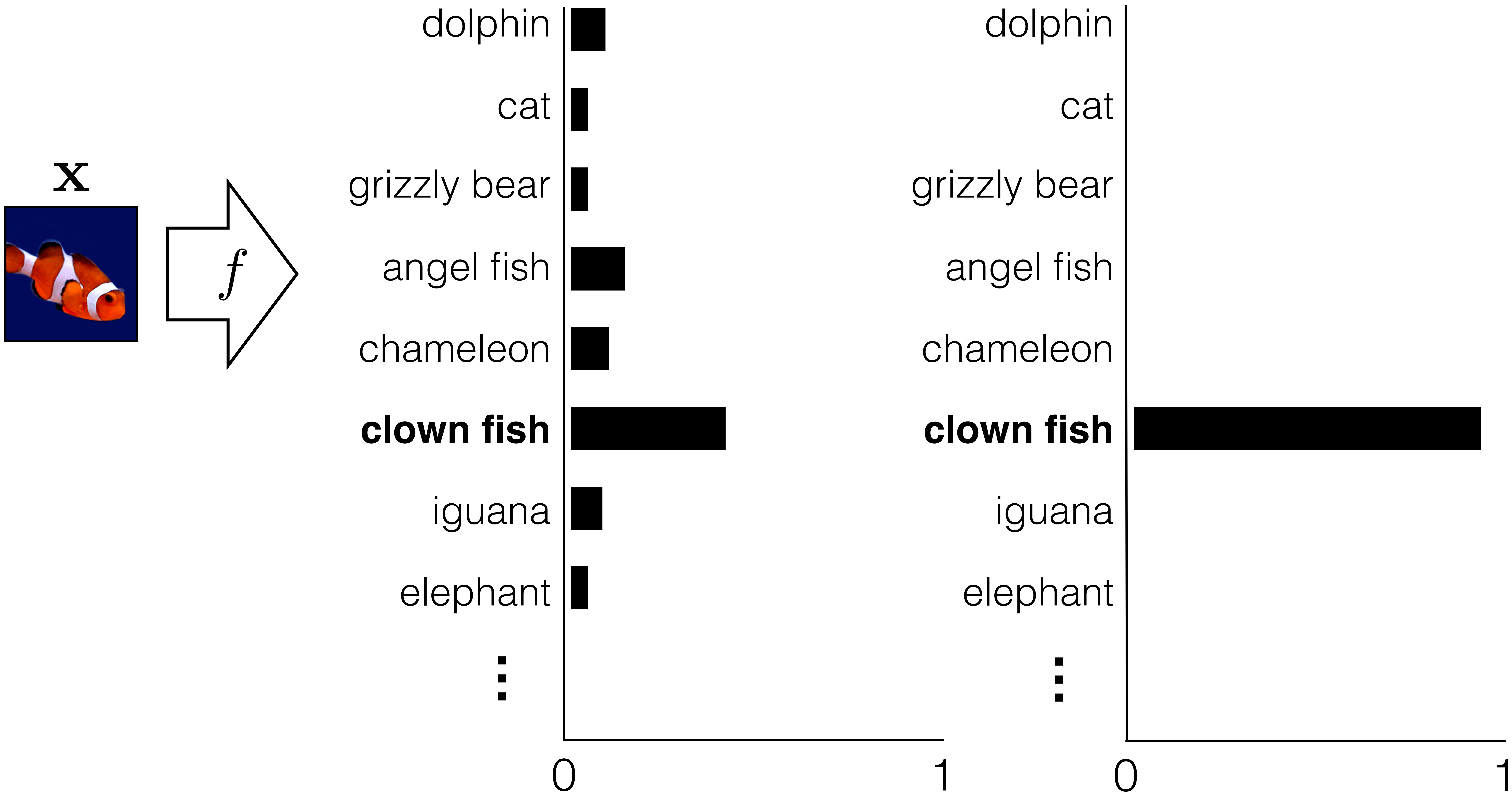
$\mathbf{x}$



Source: Isola, Torralba, Freeman

Prediction $\hat{\mathbf{y}}$ Ground truth label $\mathbf{y}$

$$f_\theta : X \to \mathbb{R}^K$$



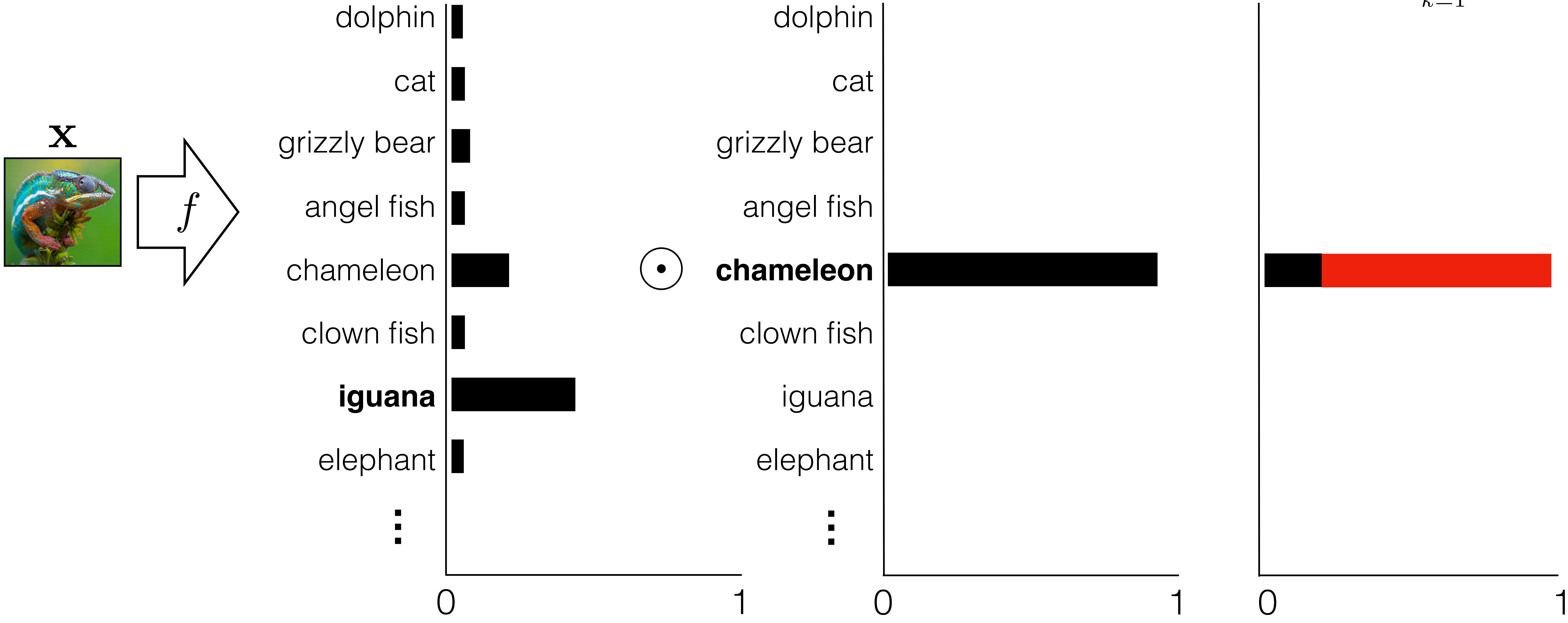
Source: Isola, Torralba, Freeman

Source: Isola, Torralba, Freeman

Prediction $\hat{\mathbf{y}}$ | Ground truth label $\mathbf{y}$ | Loss

$$f_\theta : X \to \mathbb{R}^K$$

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$



Source: Isola, Torralba, Freeman

Softmax regression (a.k.a. multinomial logistic regression)

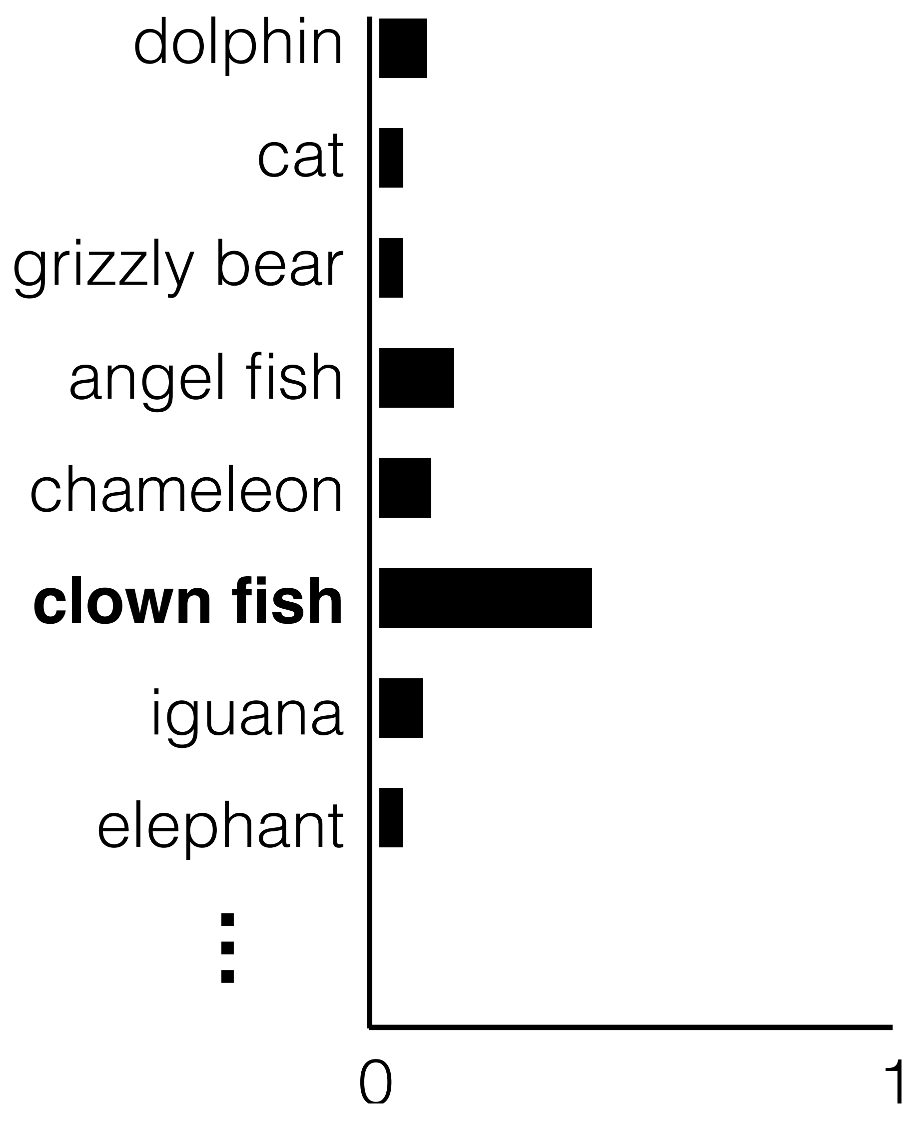
$$f_\theta : X \to \mathbb{R}^K$$

$$\mathbf{z} = f_\theta(\mathbf{x})$$ ⟵ **logits**: vector of K scores, one for each class

$$\hat{\mathbf{y}} = \mathtt{softmax}(\mathbf{z})$$ ⟵ squash into a non-negative vector that sums to 1 — i.e. **a probability density function**

$$\hat{y}_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_j}}$$

$\hat{\mathbf{y}} =$



Source: Isola, Torralba, Freeman

Next lecture: more linear models