Lecture 6: Machine learning

- My office hours cancelled this week
  - Can do them by appointment instead
- PS3 out tonight

Today

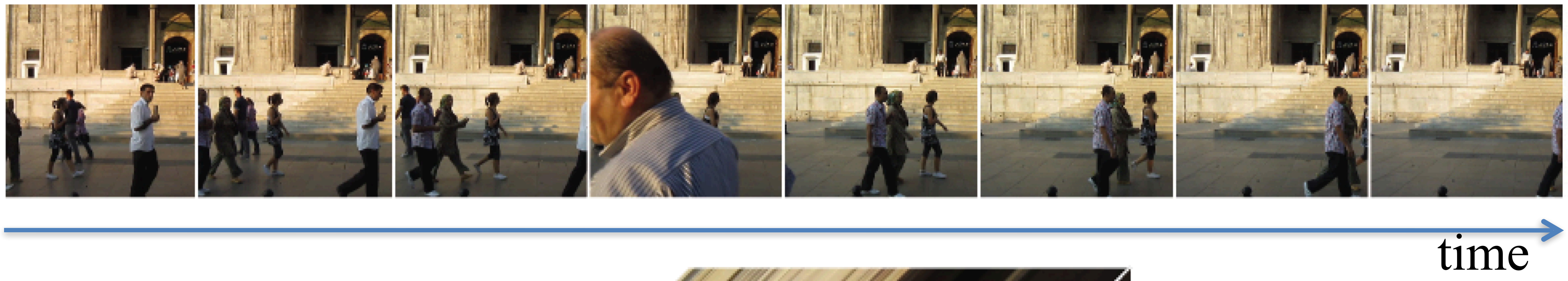
- Temporal filtering
- Introduction to machine learning

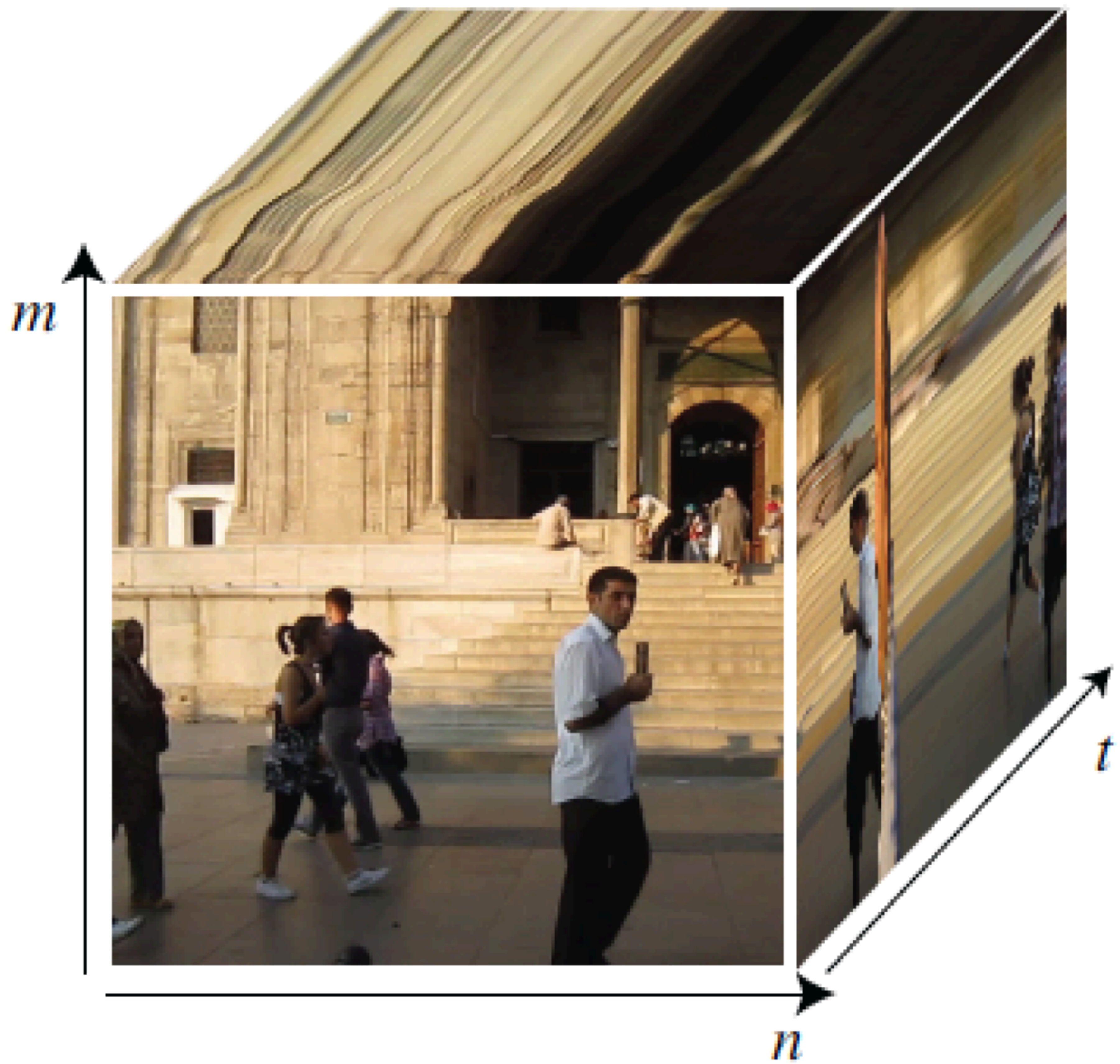
Temporal filtering

Temporal filtering

Source: Freeman, Torralba, Isola

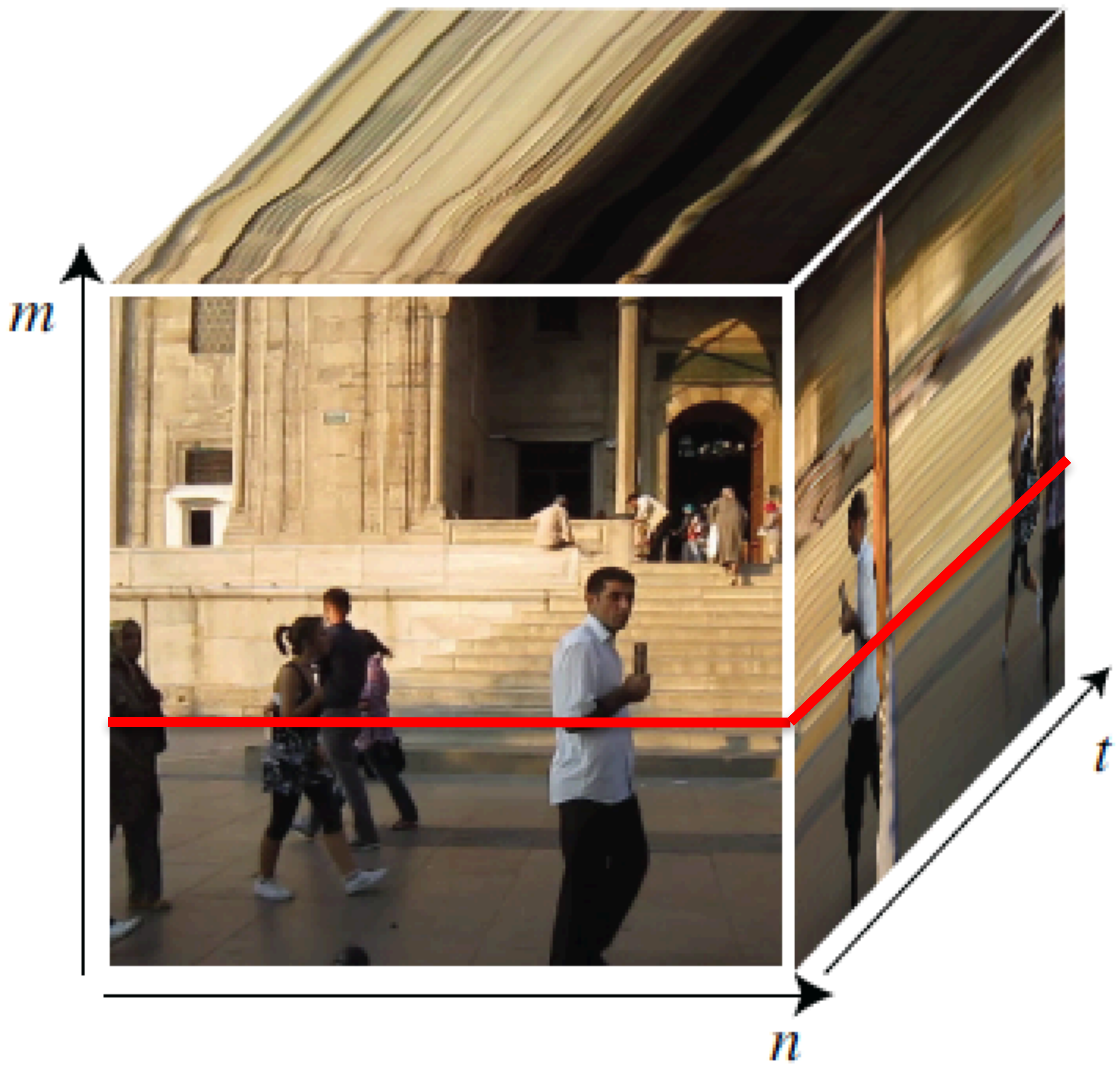
Videos



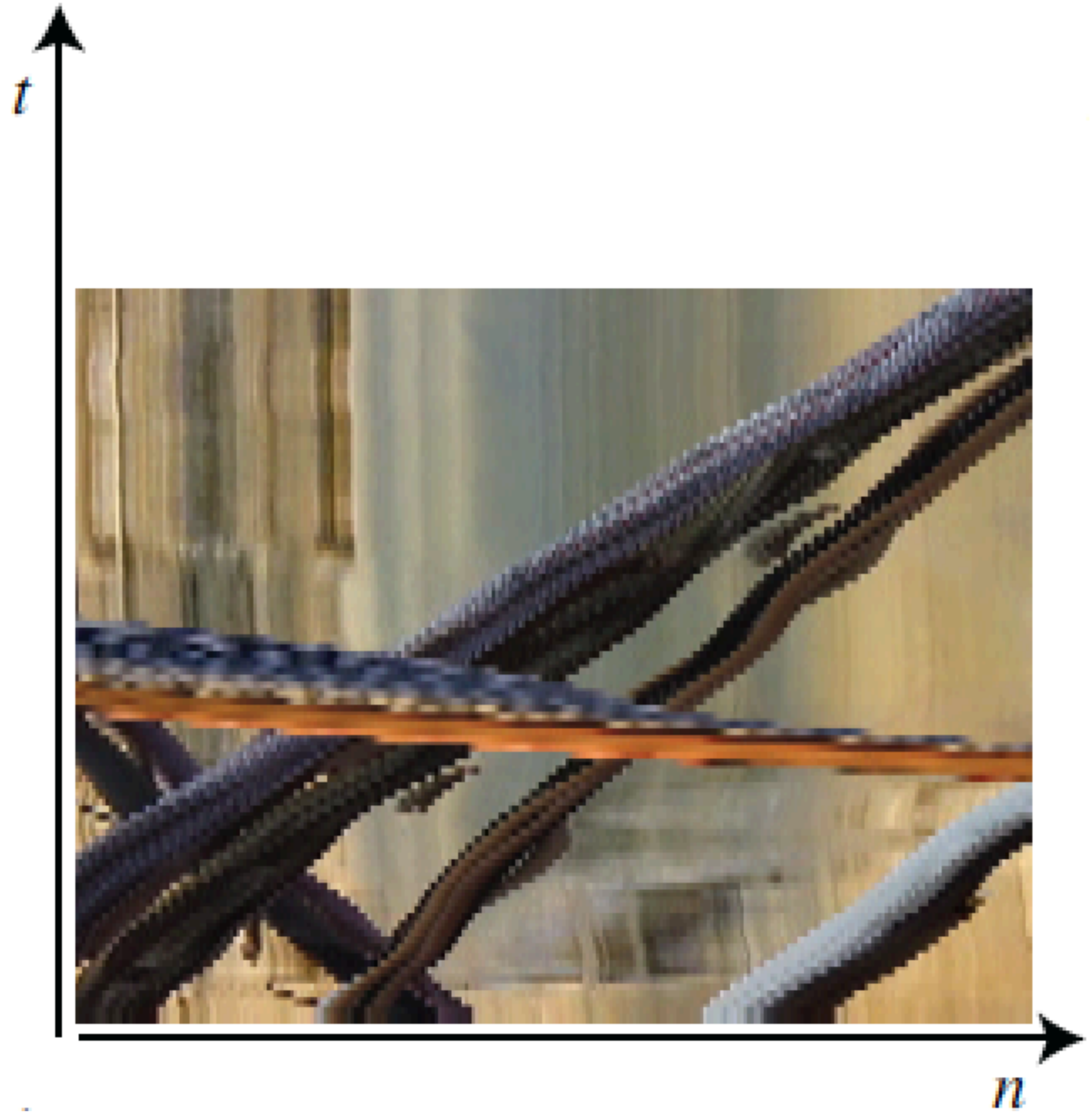


Source: Freeman, Torralba, Isola

Videos

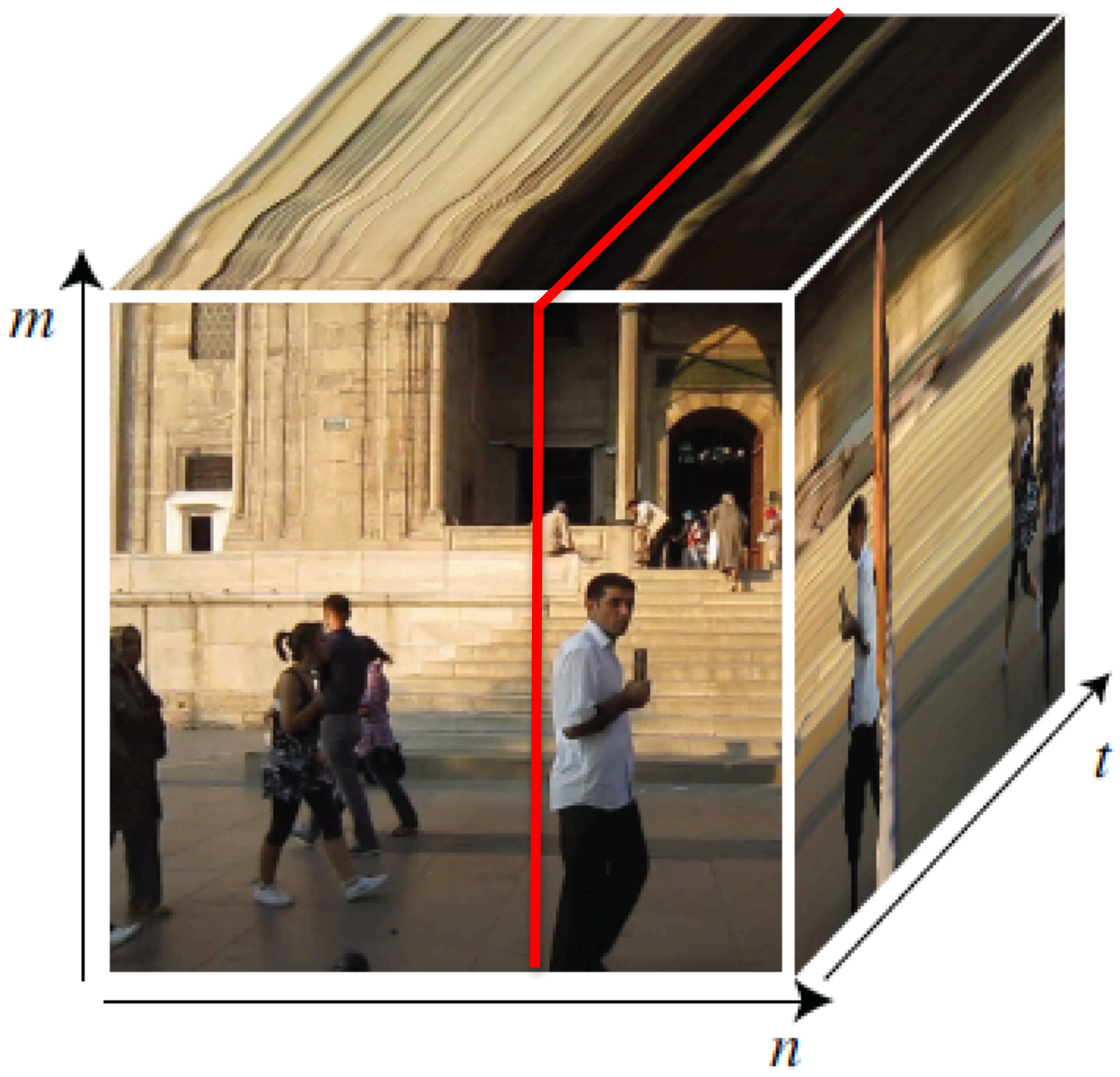


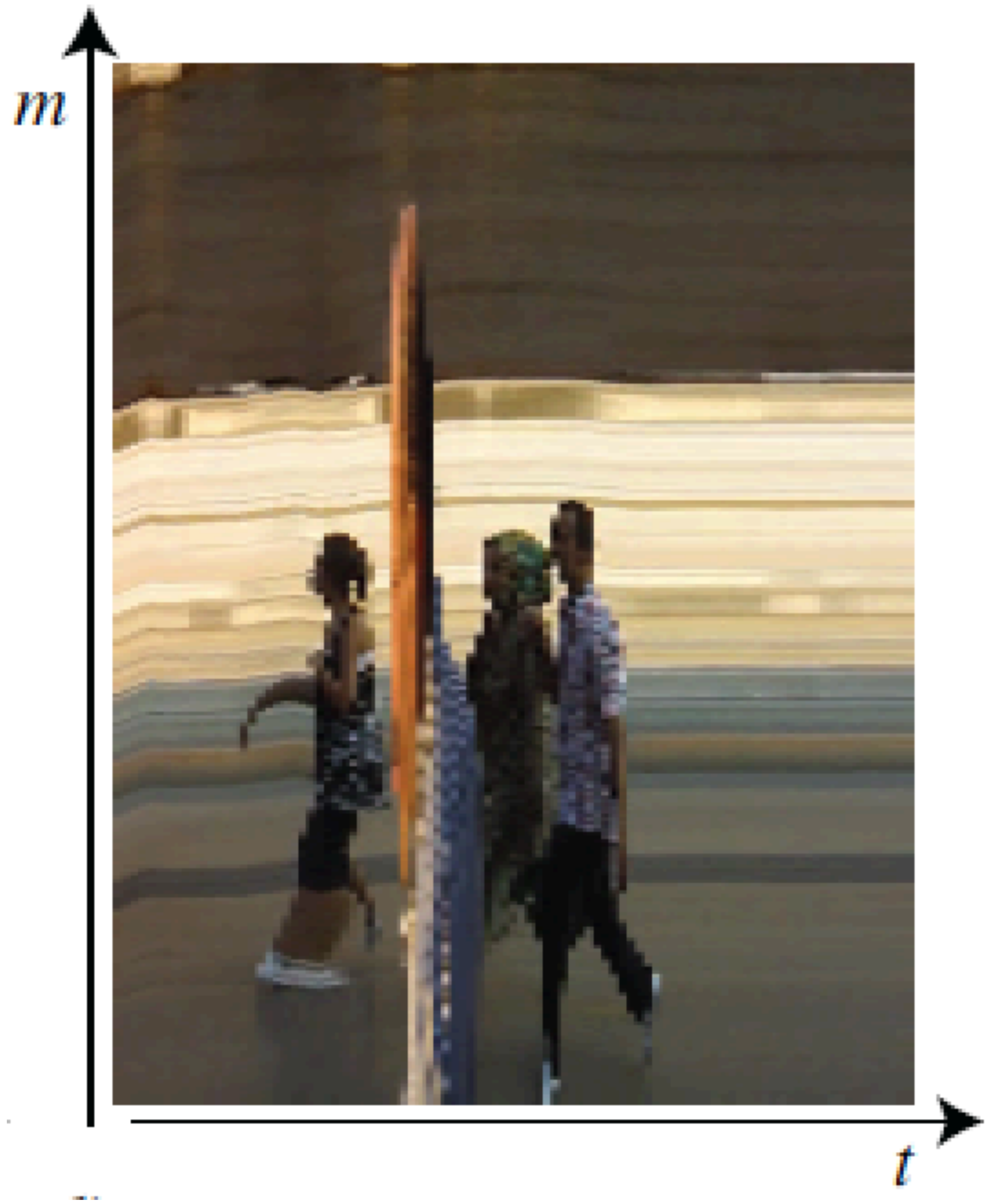
Cube size = 128x128x90



Source: Freeman, Torralba, Isola

Videos





Cube size = 128x128x90

Source: Freeman, Torralba, Isola

Color amplification

[Wadhwa et al., Phase-based Video Motion Processing. SIGGRAPH 2013, Wu et al. SIGGRAPH 2012]

Motion magnification

Original

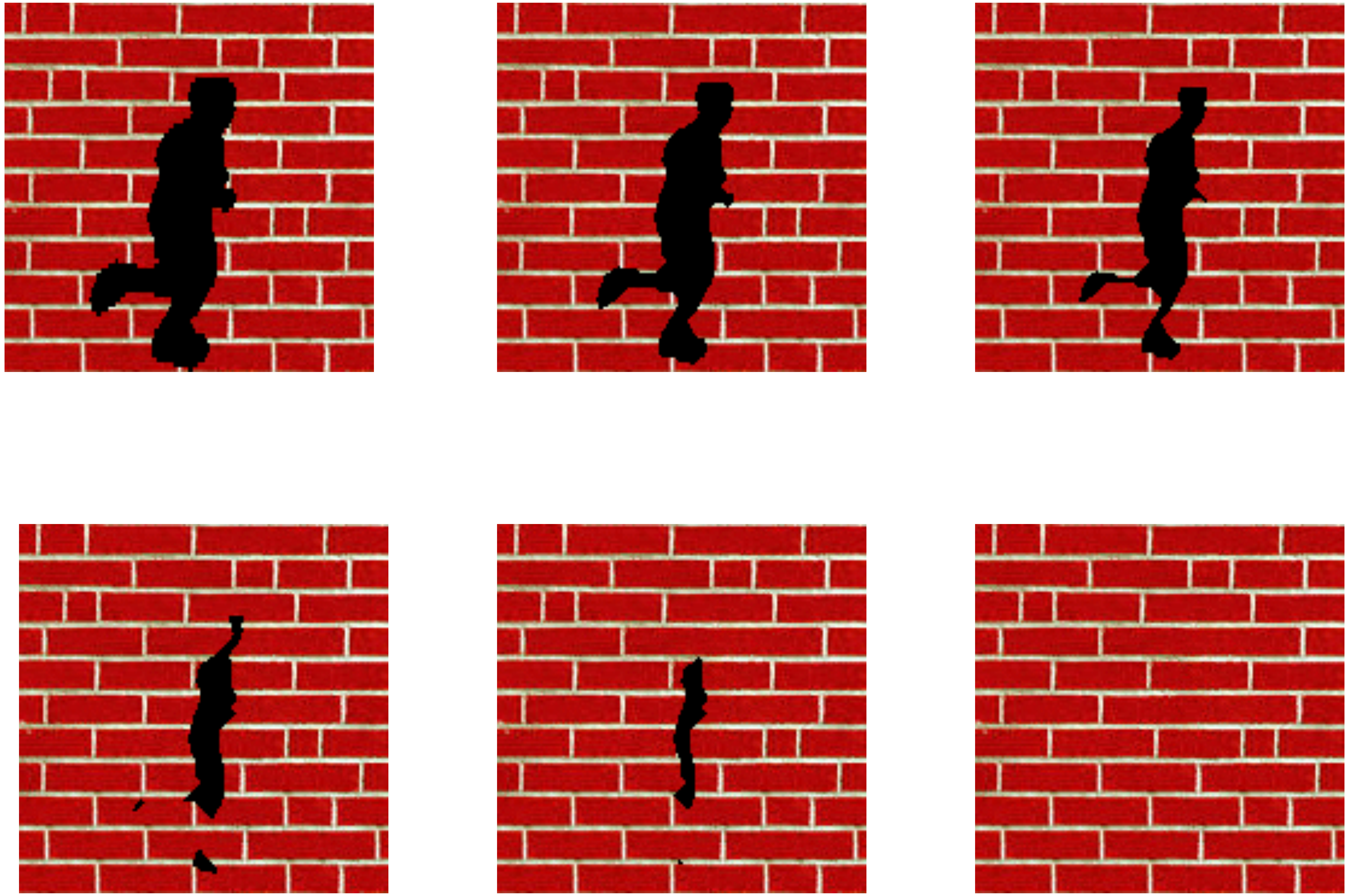
- Just spatial-temporal filtering!
- You'll do a simplified version in PS3

[Wu et al., SIGGRAPH 2012]

[Wadhwa et al., SIGGRAPH 2013]

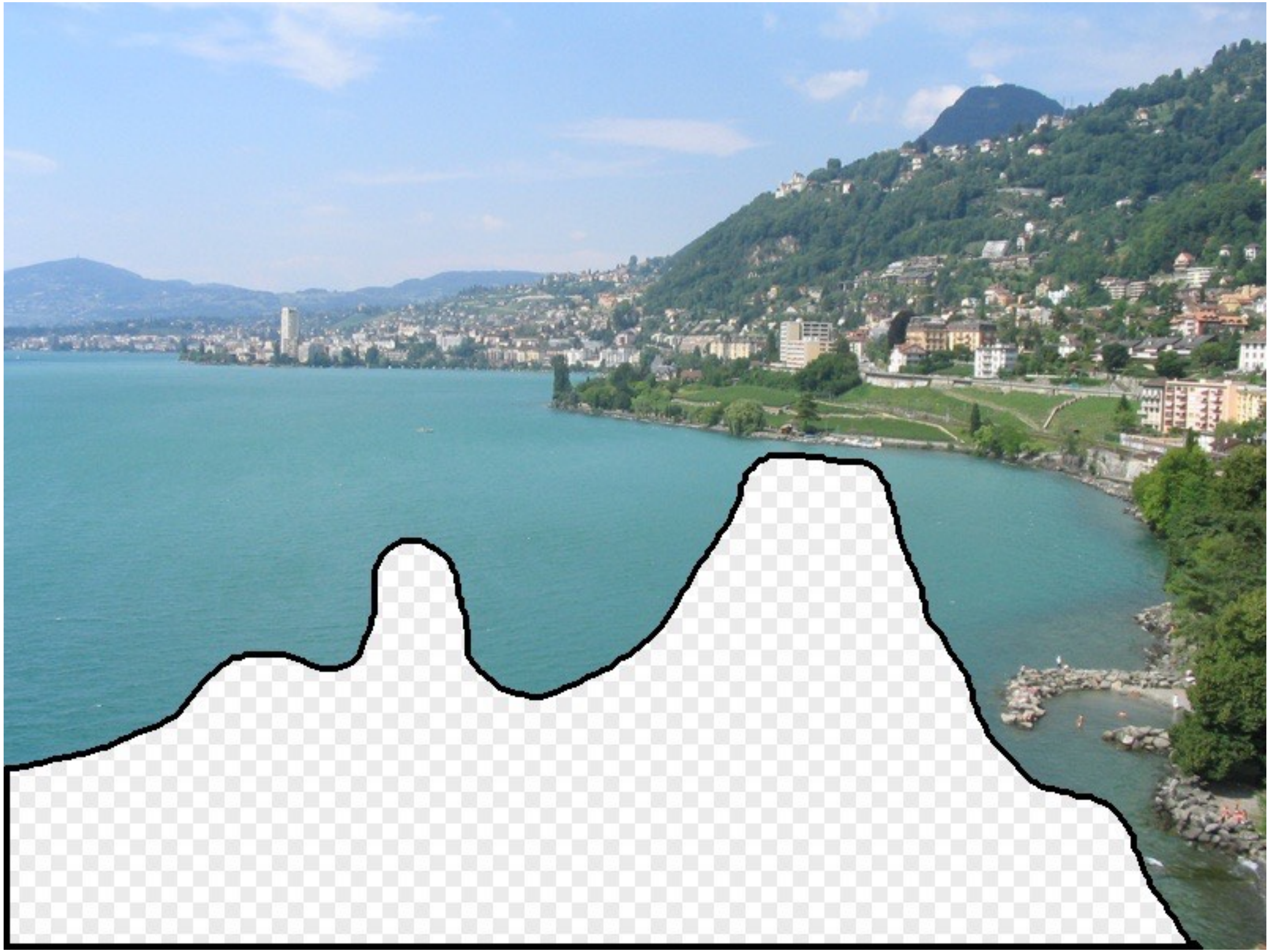
Machine learning

Last week: hole filling



Source: A. Efros

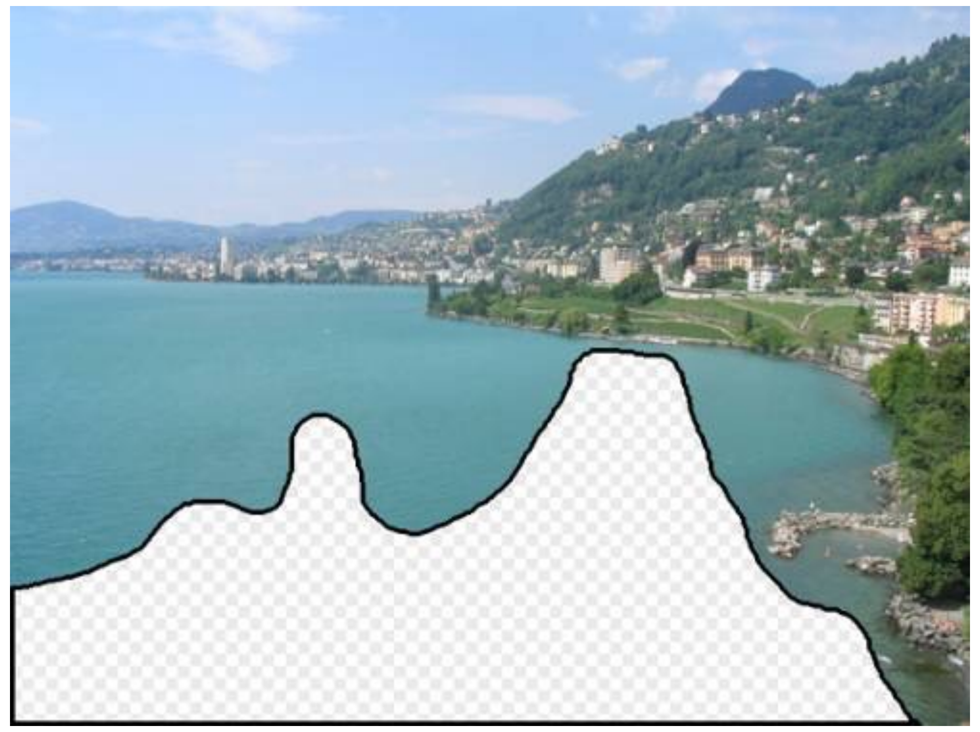
Source: A. Efros

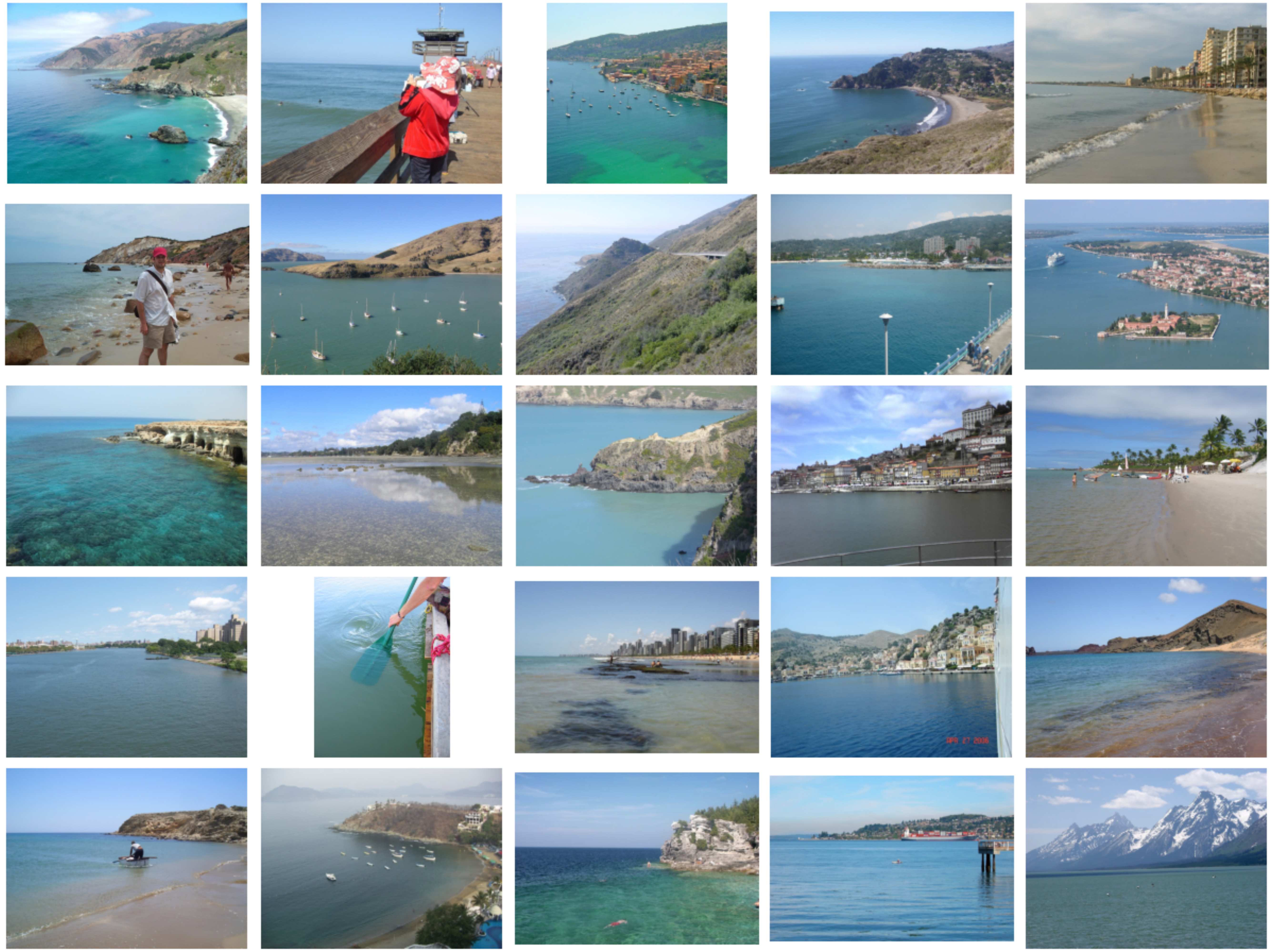


[Hays and Efros. Scene Completion Using Millions of Photographs. SIGGRAPH 2007.]

Source: A. Efros

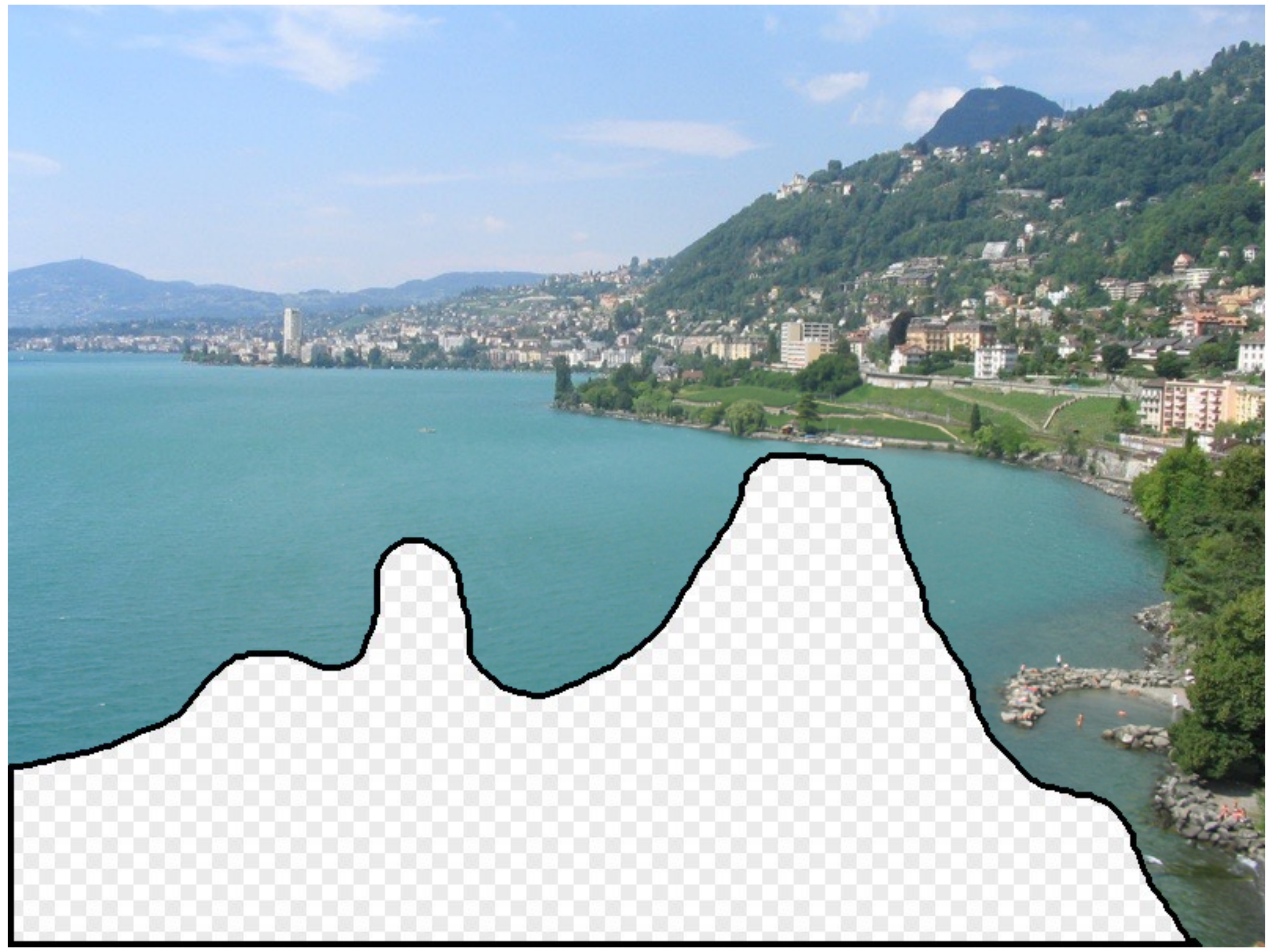
Source: A. Efros





… 200 total

Source: A. Efros



Source: A. Efros

Source: A. Efros

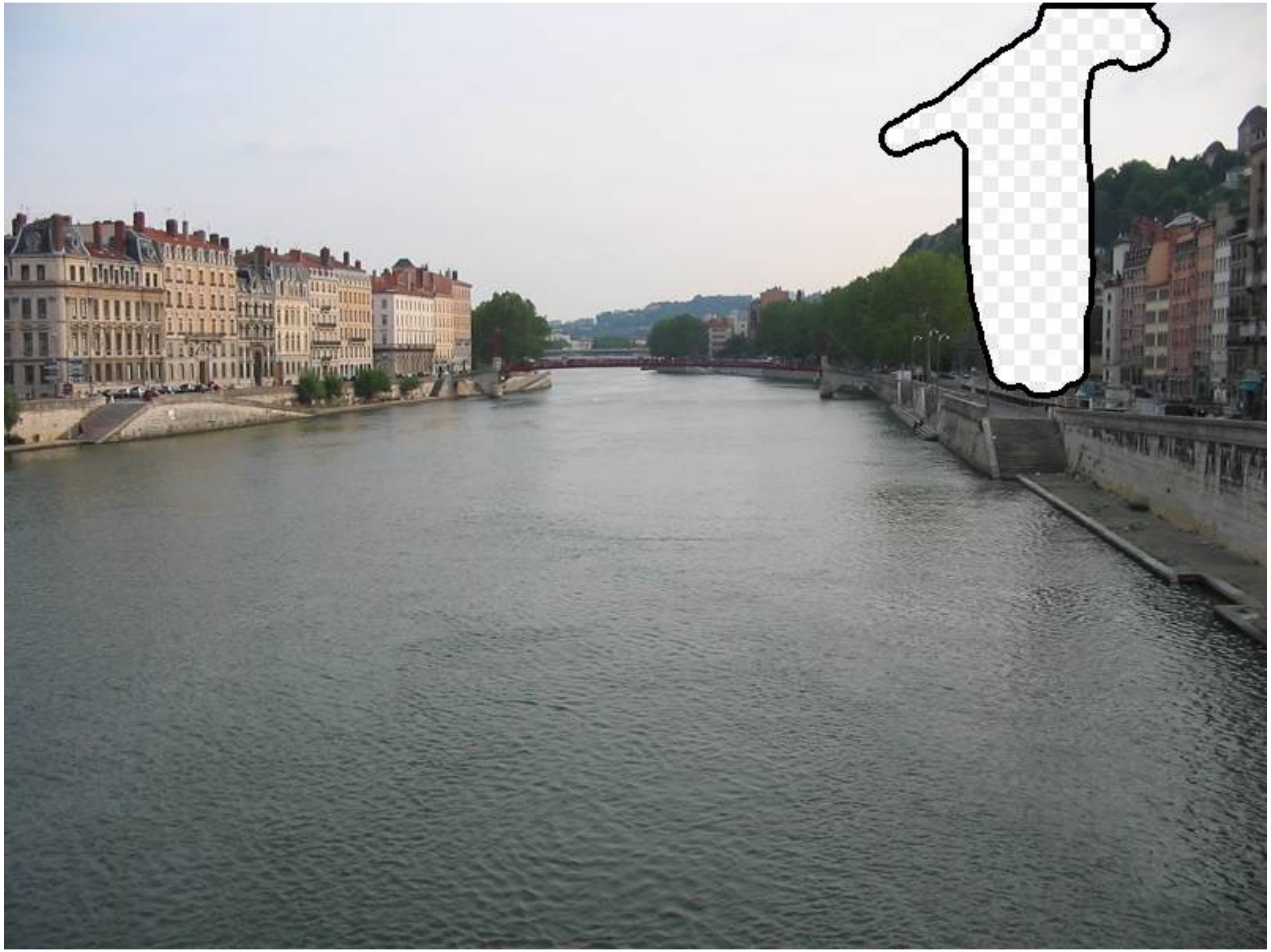
Source: A. Efros

Source: A. Efros

Source: A. Efros

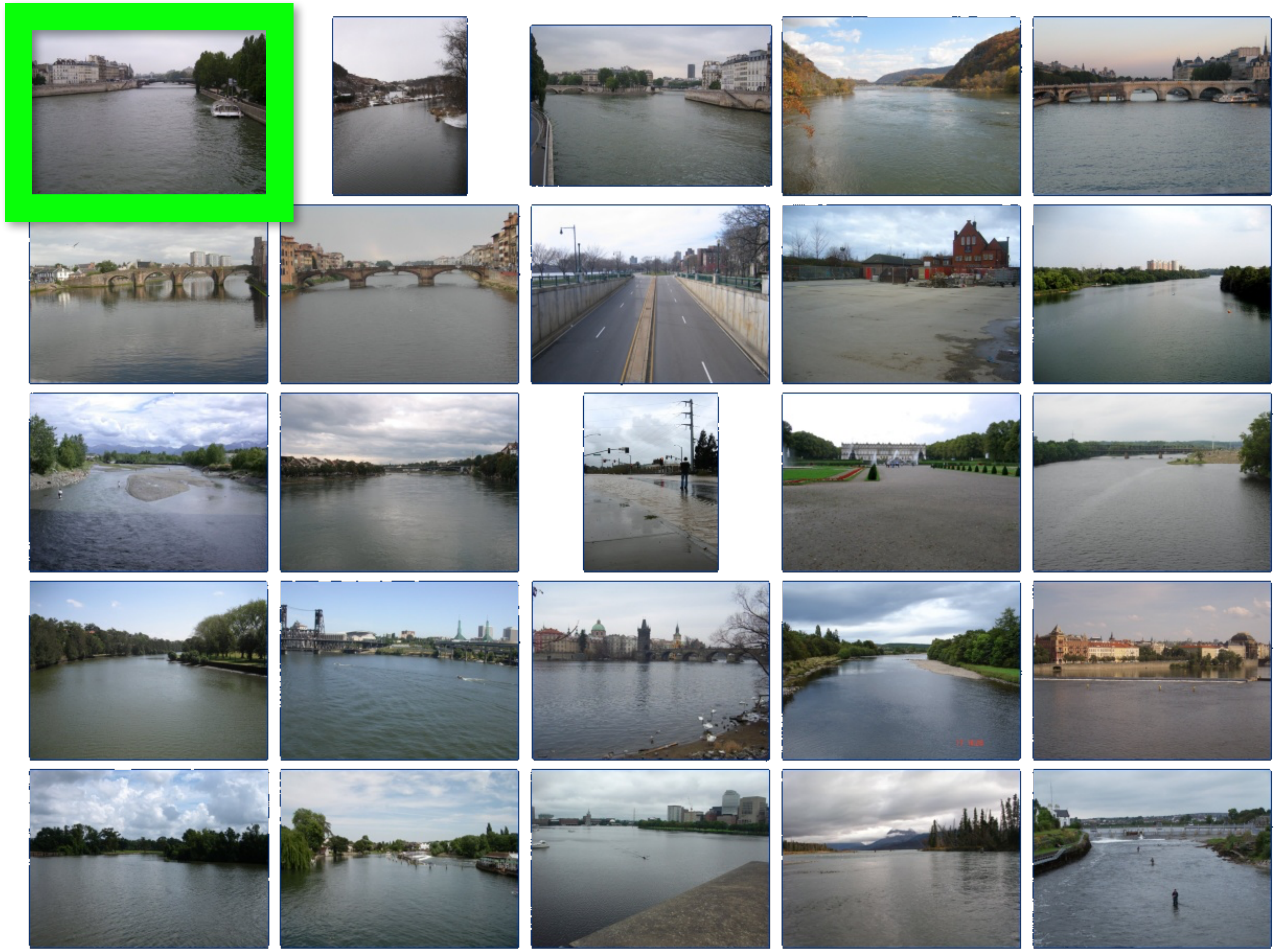
Source: A. Efros

Source: A. Efros



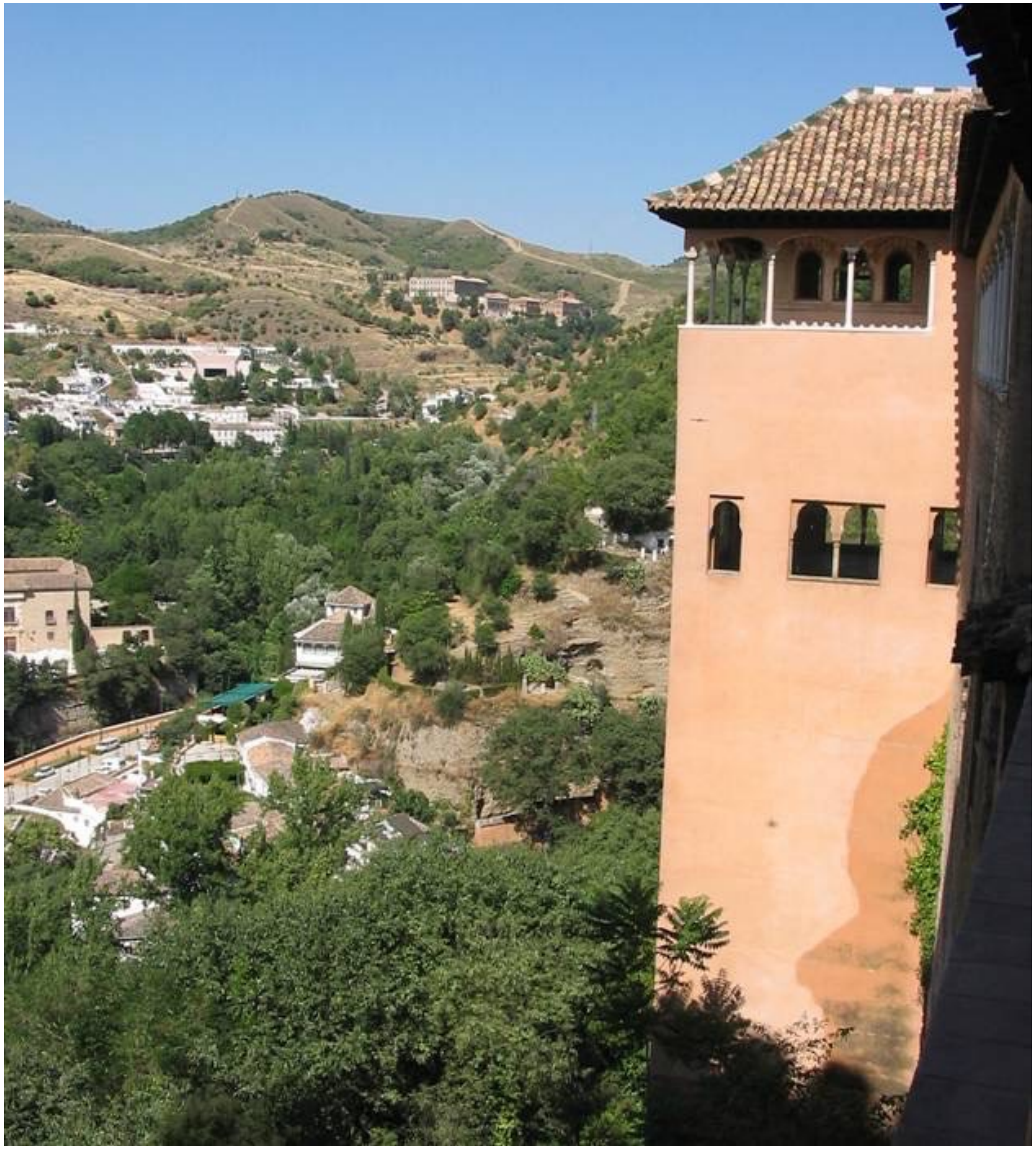
Source: A. Efros

Source: A. Efros

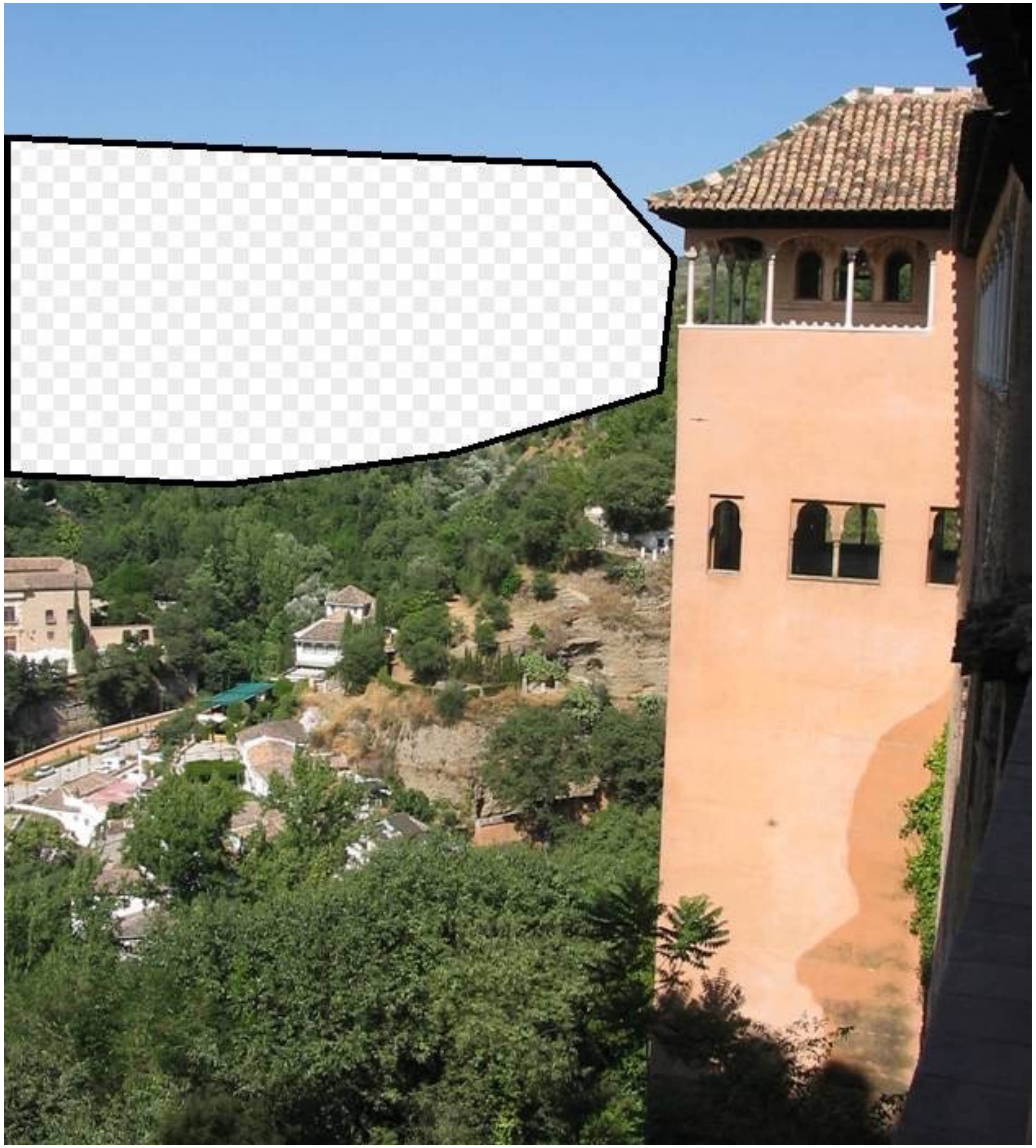


… 200 scene matches

Source: A. Efros



Source: A. Efros

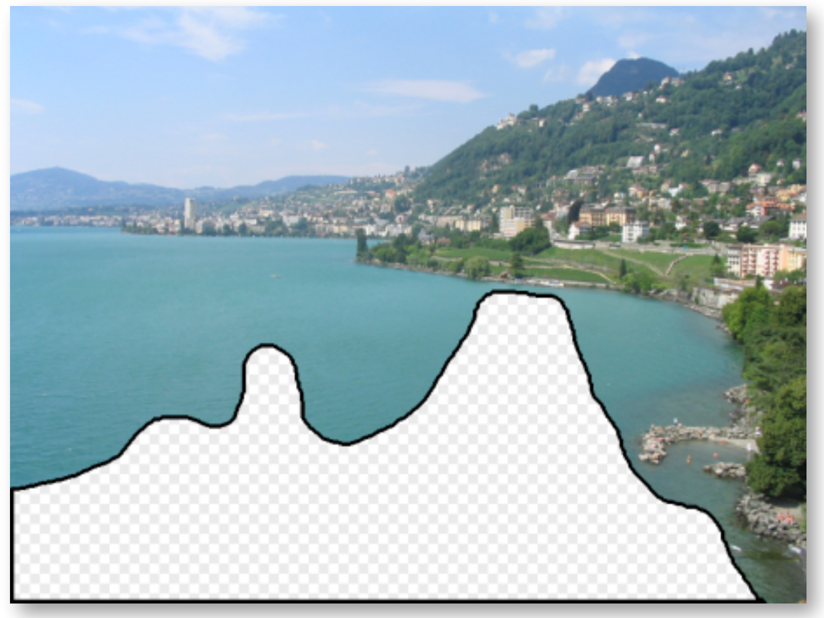


Source: A. Efros

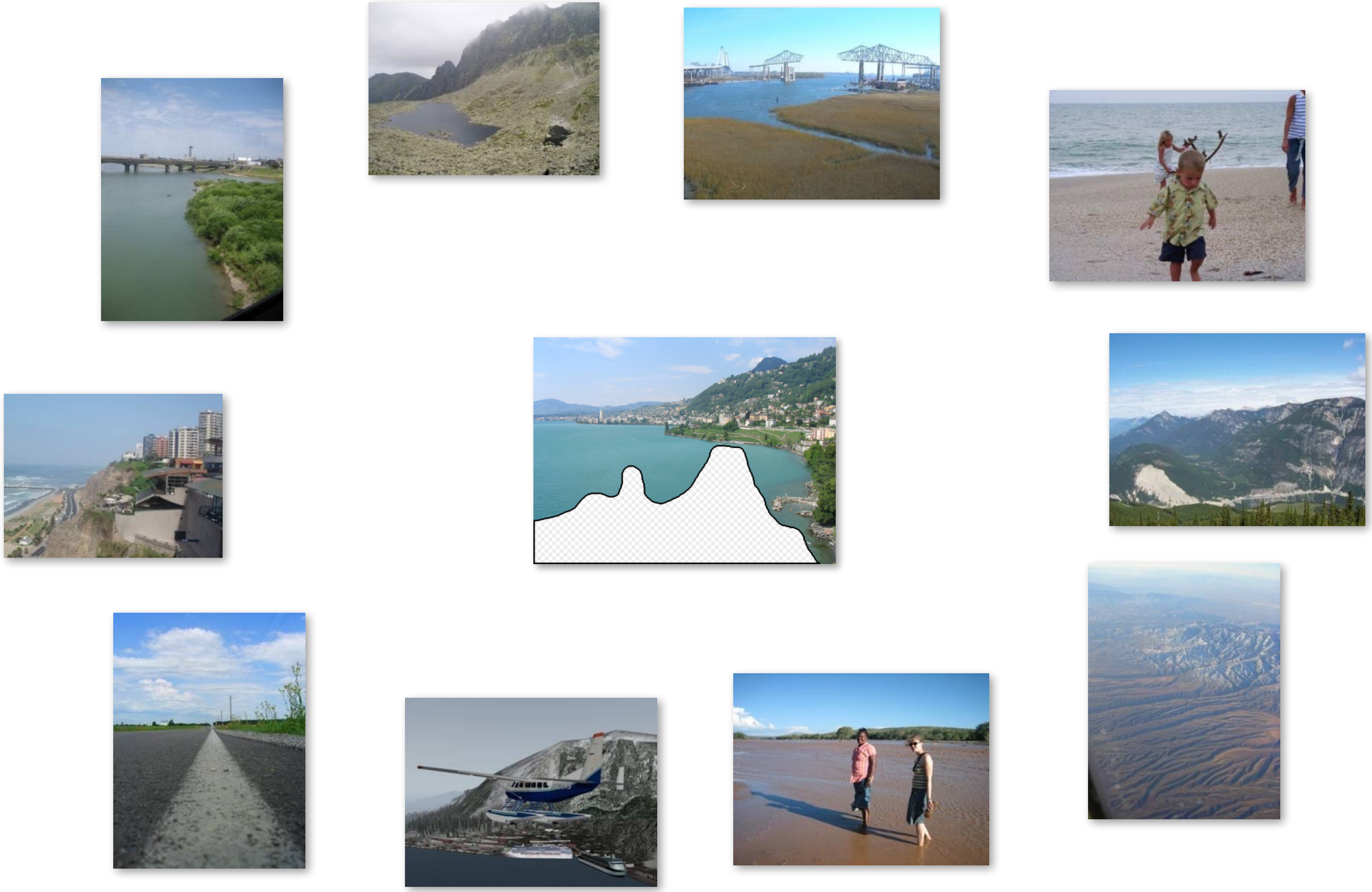
Source: A. Efros

Why does it work?

Source: A. Efros

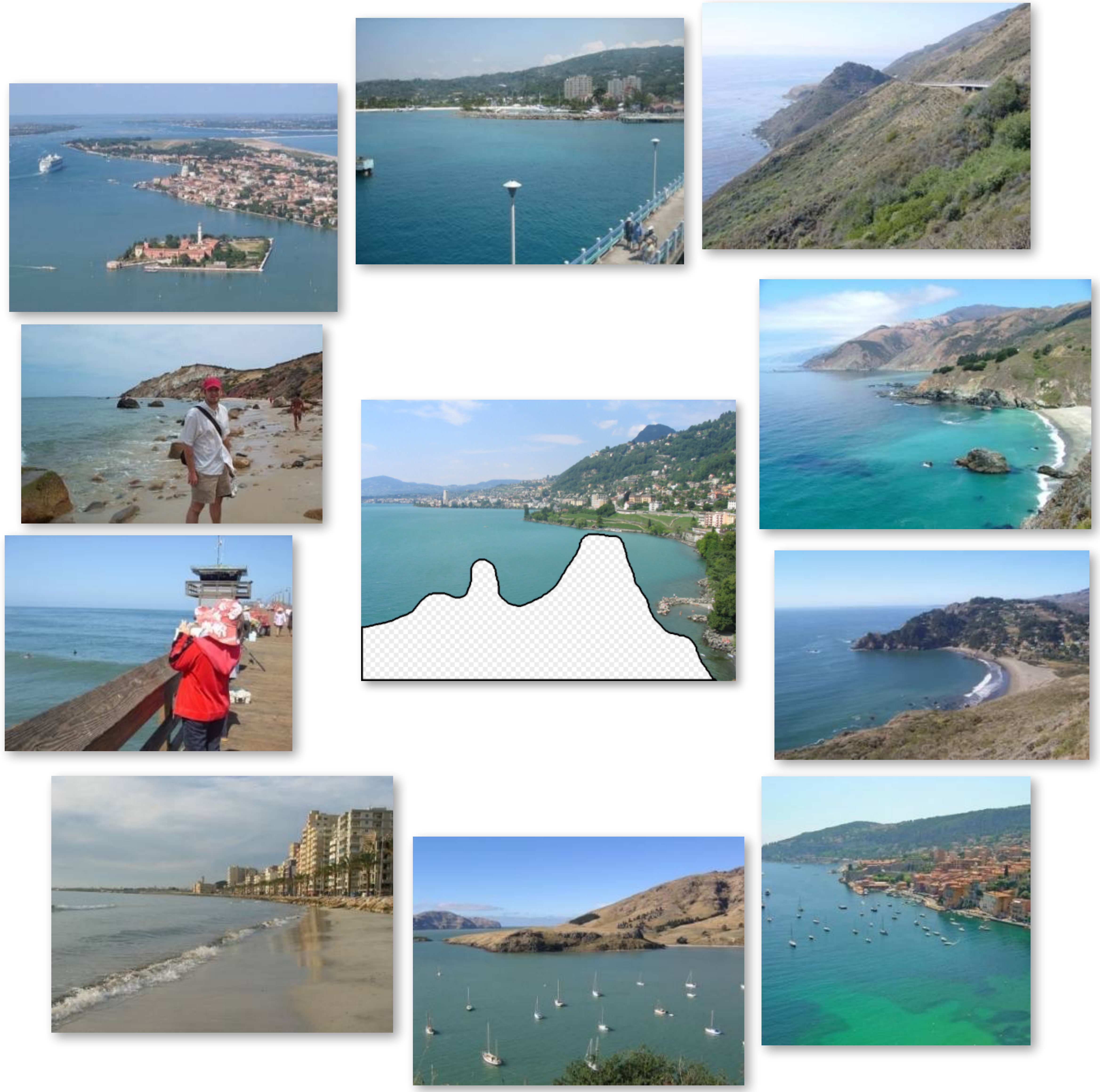


Source: A. Efros



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.

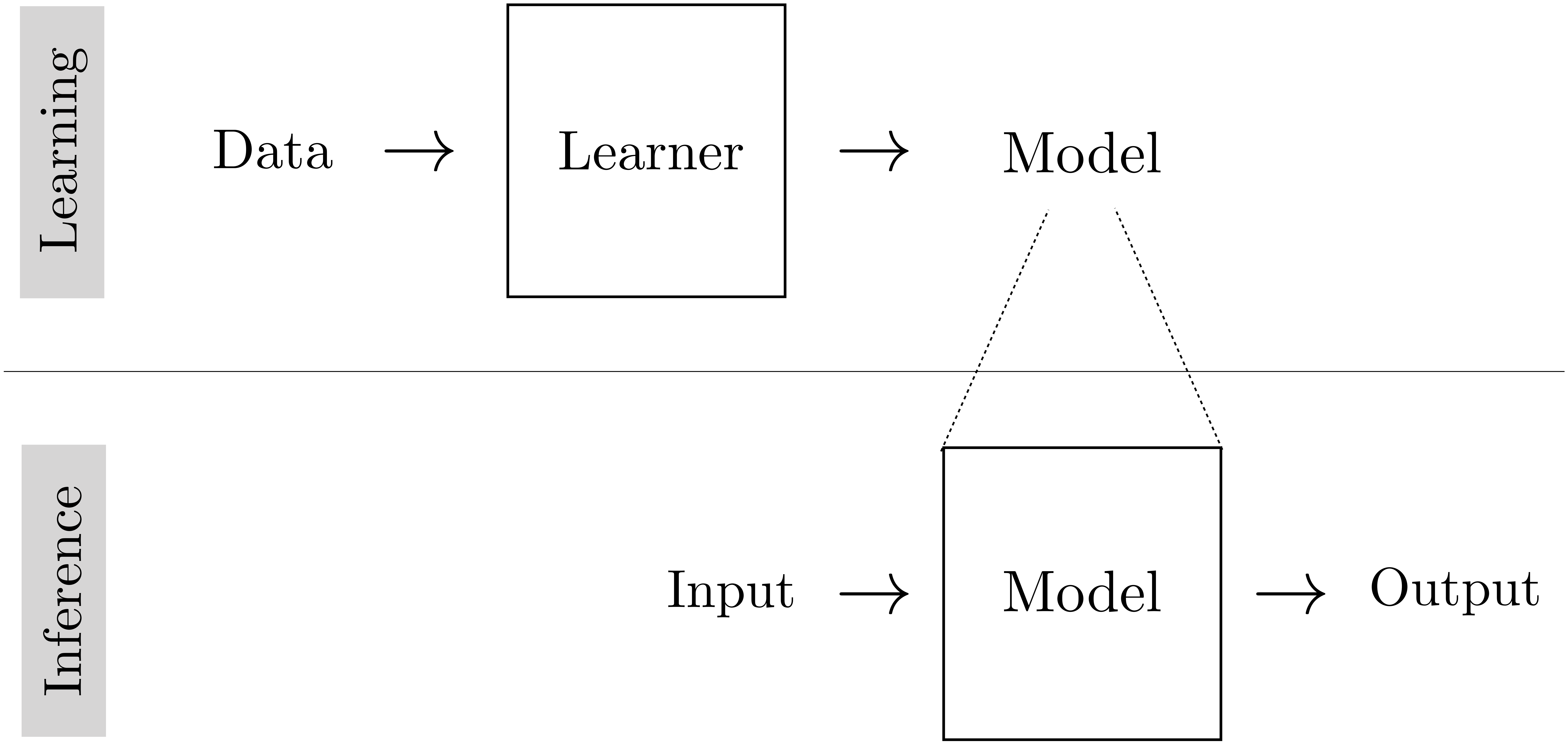
But much cannot

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



ML intro slides from: Isola, Torralba, Freeman

What does ☆ do?

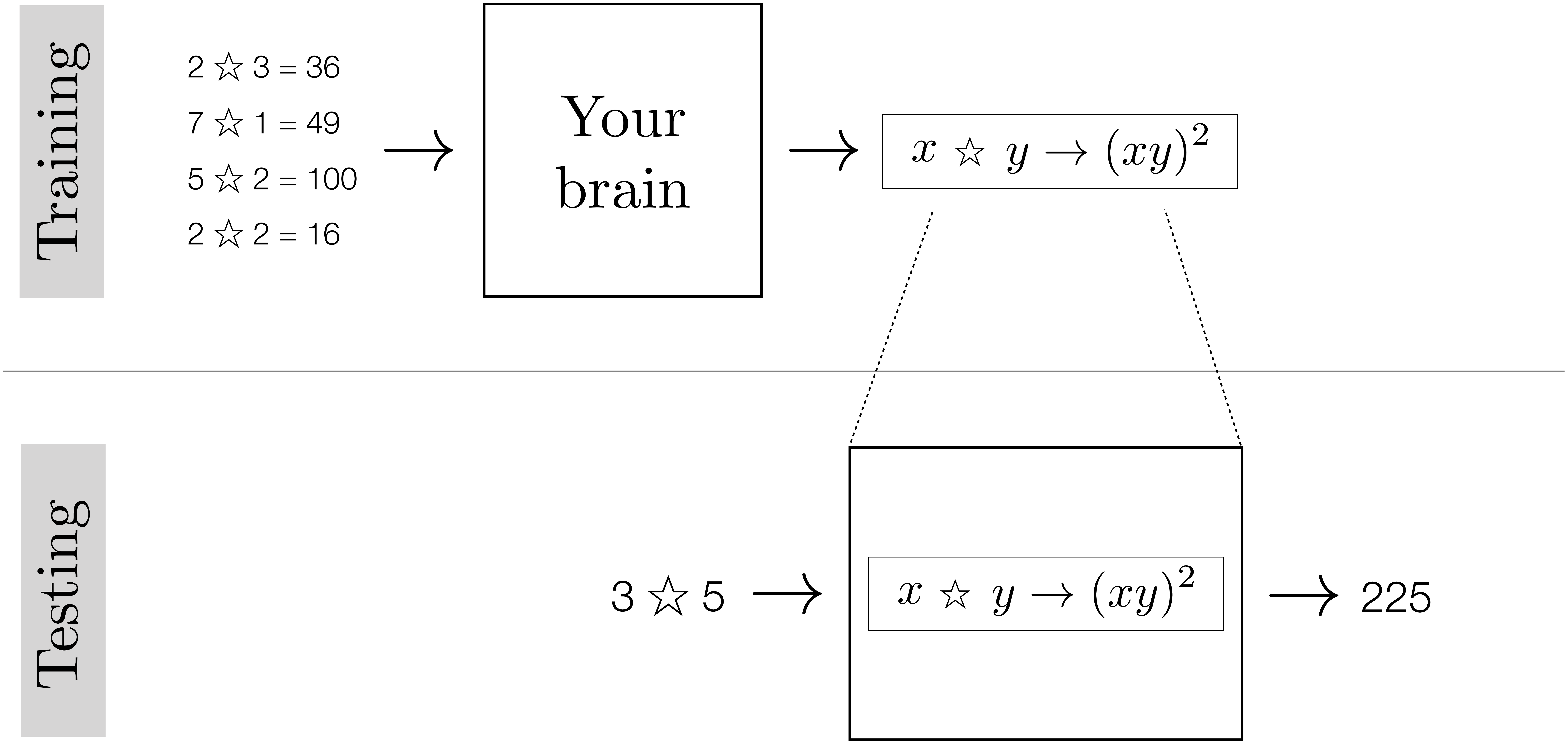
2 ☆ 3 = 36

7 ☆ 1 = 49

5 ☆ 2 = 100

2 ☆ 2 = 16

Source: Isola, Torralba, Freeman



Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

$$\{\texttt{input:}[2,3],\texttt{output:}36\}$$
$$\{\texttt{input:}[7,1],\texttt{output:}49\}$$
$$\{\texttt{input:}[5,2],\texttt{output:}100\}$$
$$\{\texttt{input:}[2,2],\texttt{output:}16\}$$

$\rightarrow$ $\boxed{\text{Learner}}$ $\rightarrow$ f

Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

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

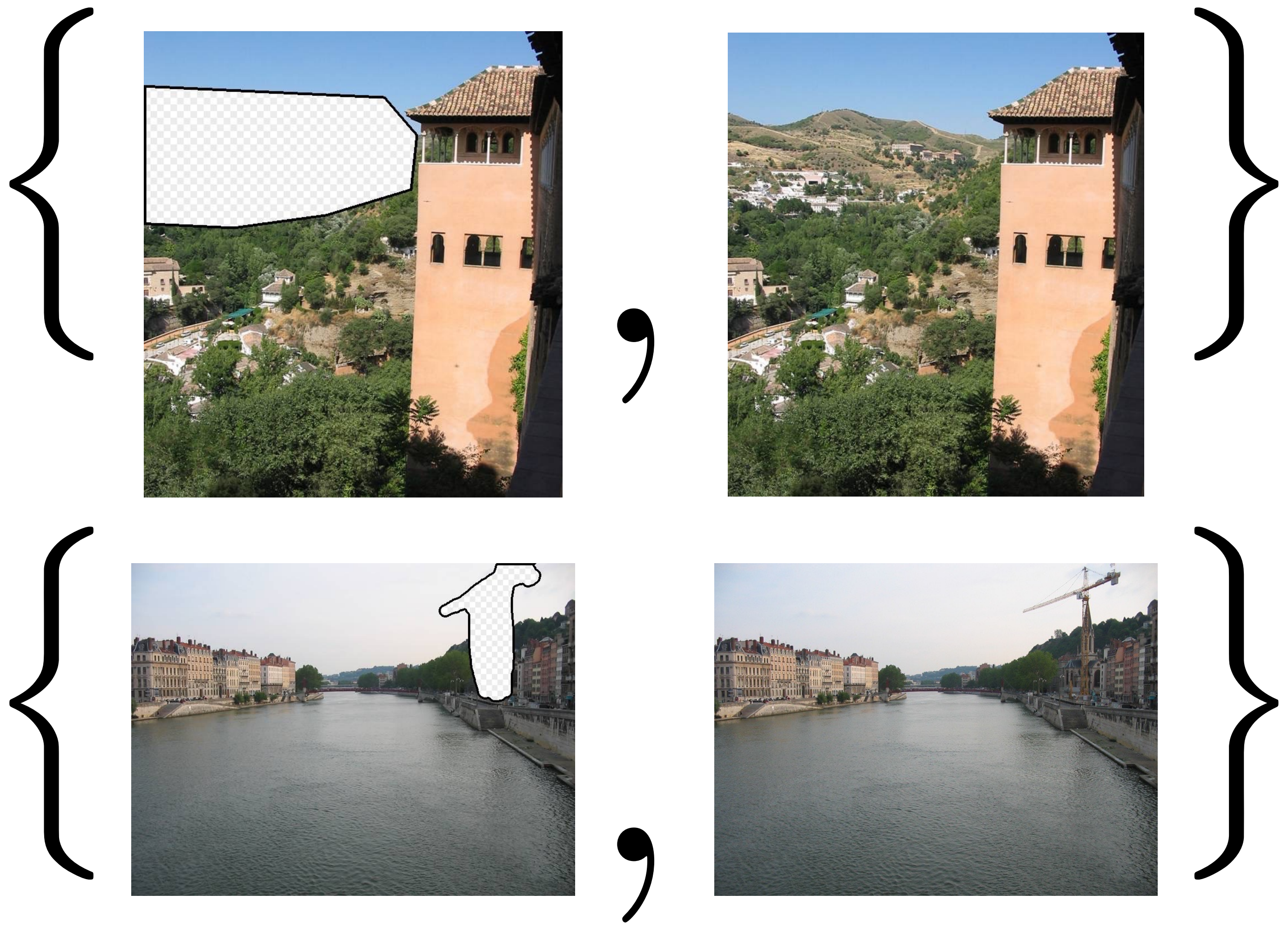
Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

X $\quad$ Y



$\vdots$

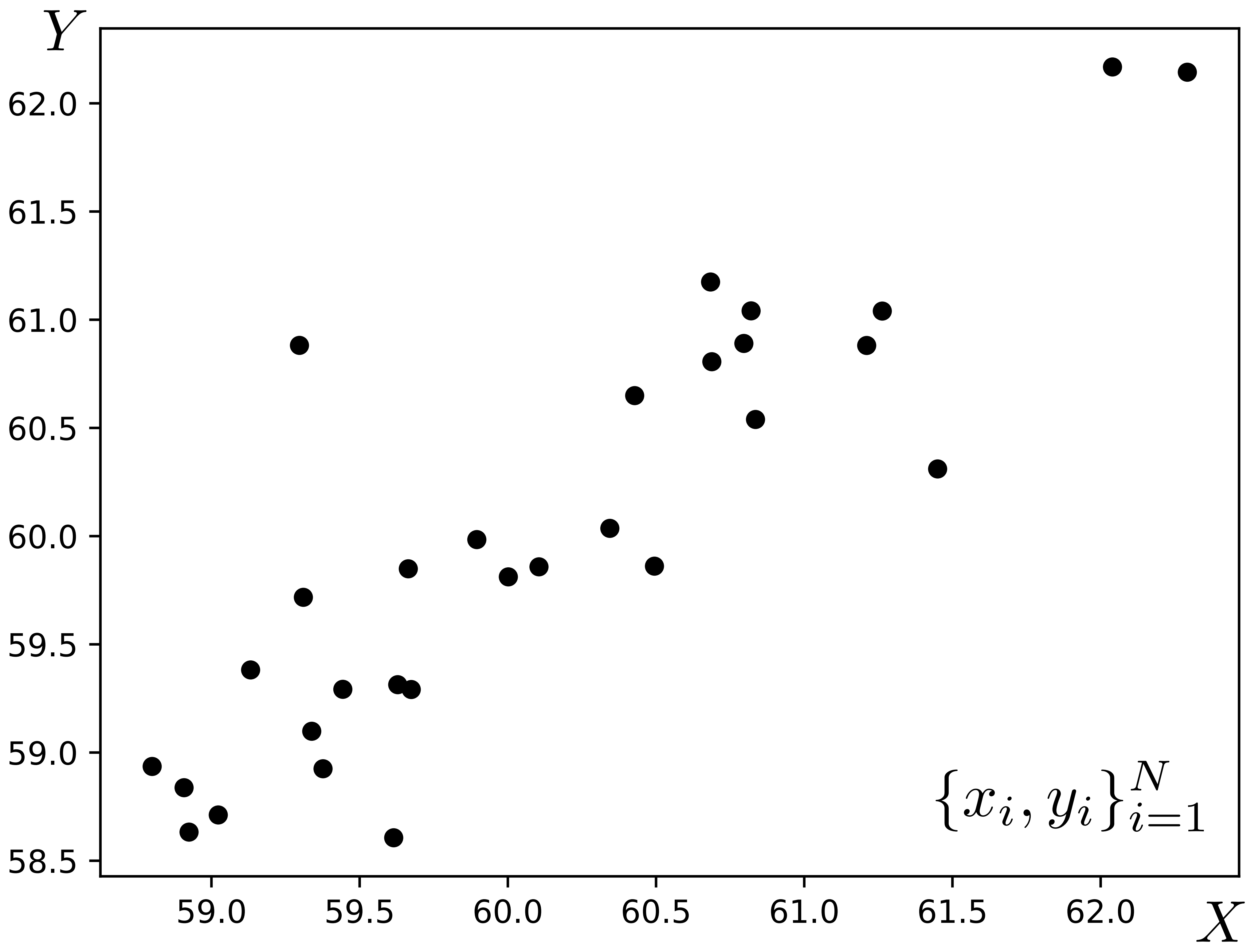
$\rightarrow$ Learner $\rightarrow$ $f : X \rightarrow Y$

Source: Isola, Torralba, Freeman

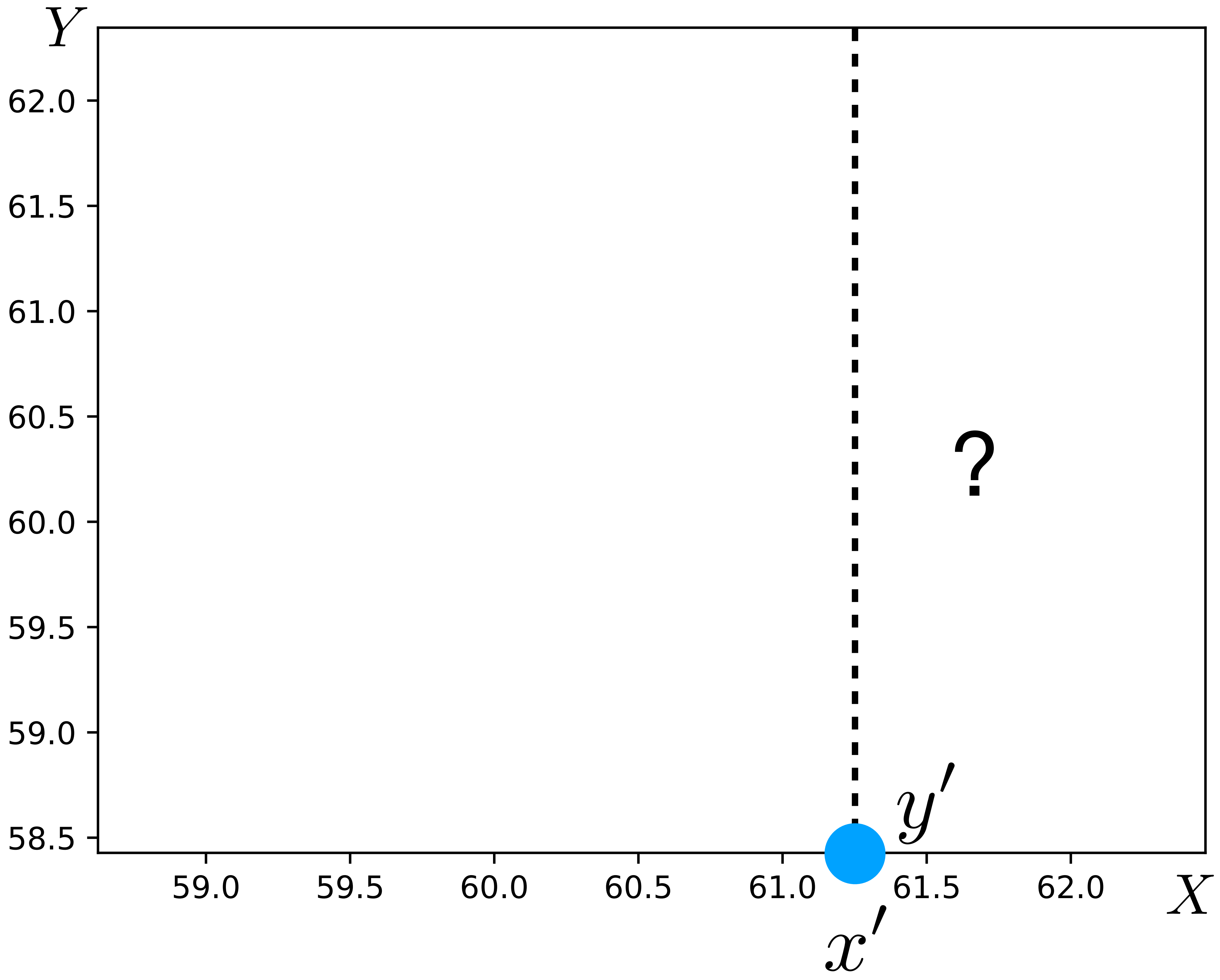
Case study #1: Linear least squares

Source: Isola, Torralba, Freeman

Training data

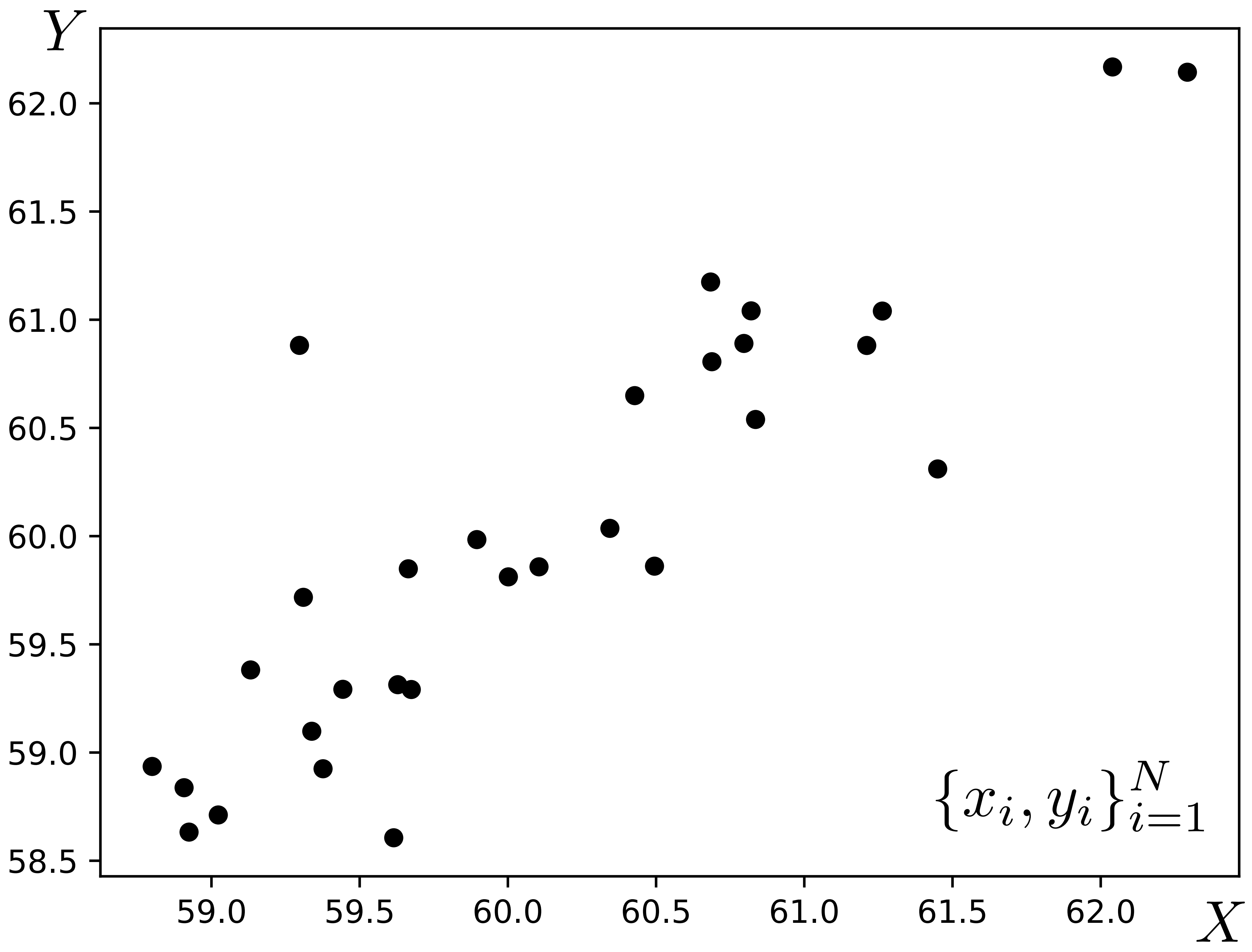


Test query



Source: Isola, Torralba, Freeman

Training data



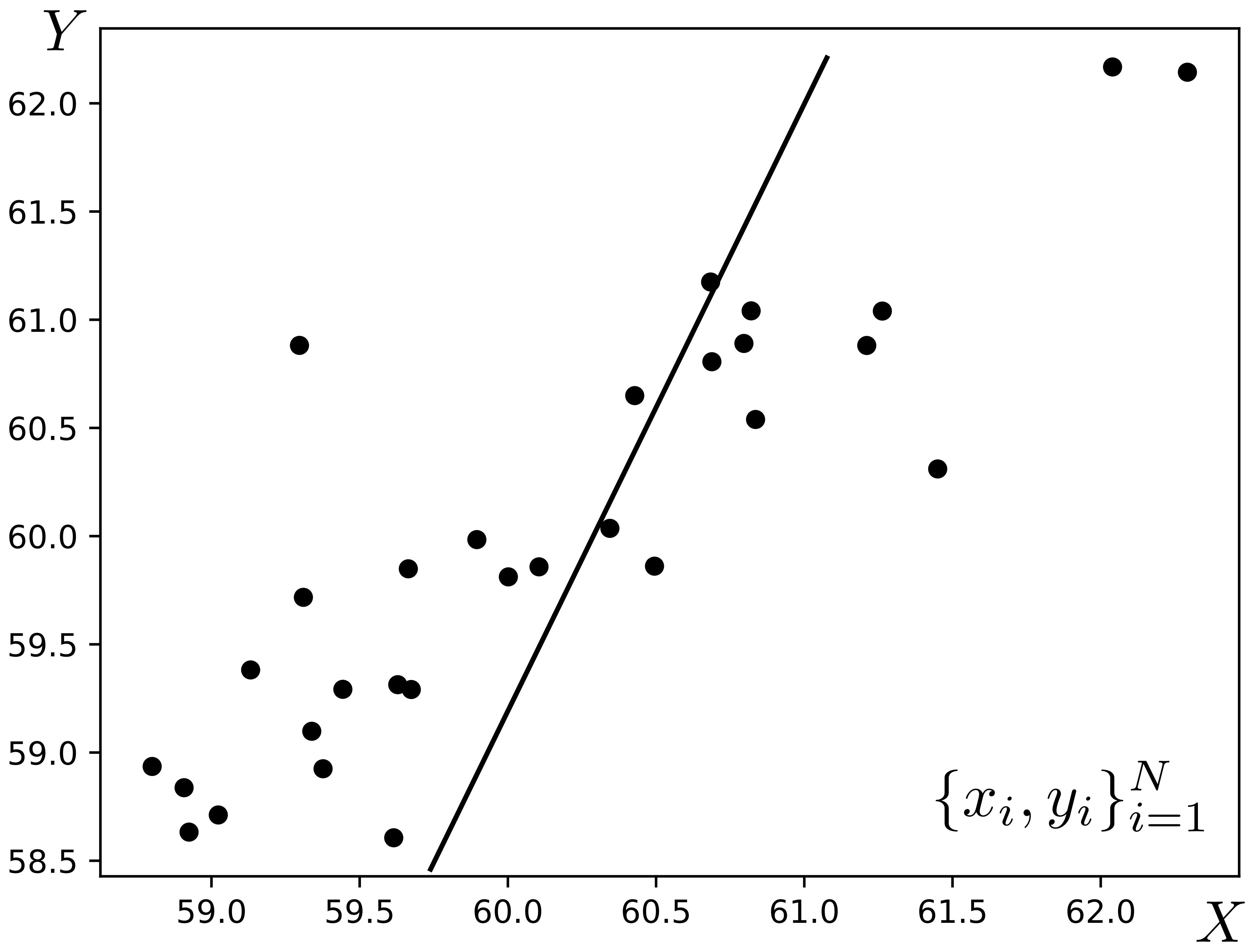
$\rightarrow$ Learner $\rightarrow$ $f_\theta(x) = \theta_1 x + \theta_0$

Hypothesis space

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

Source: Isola, Torralba, Freeman

Training data

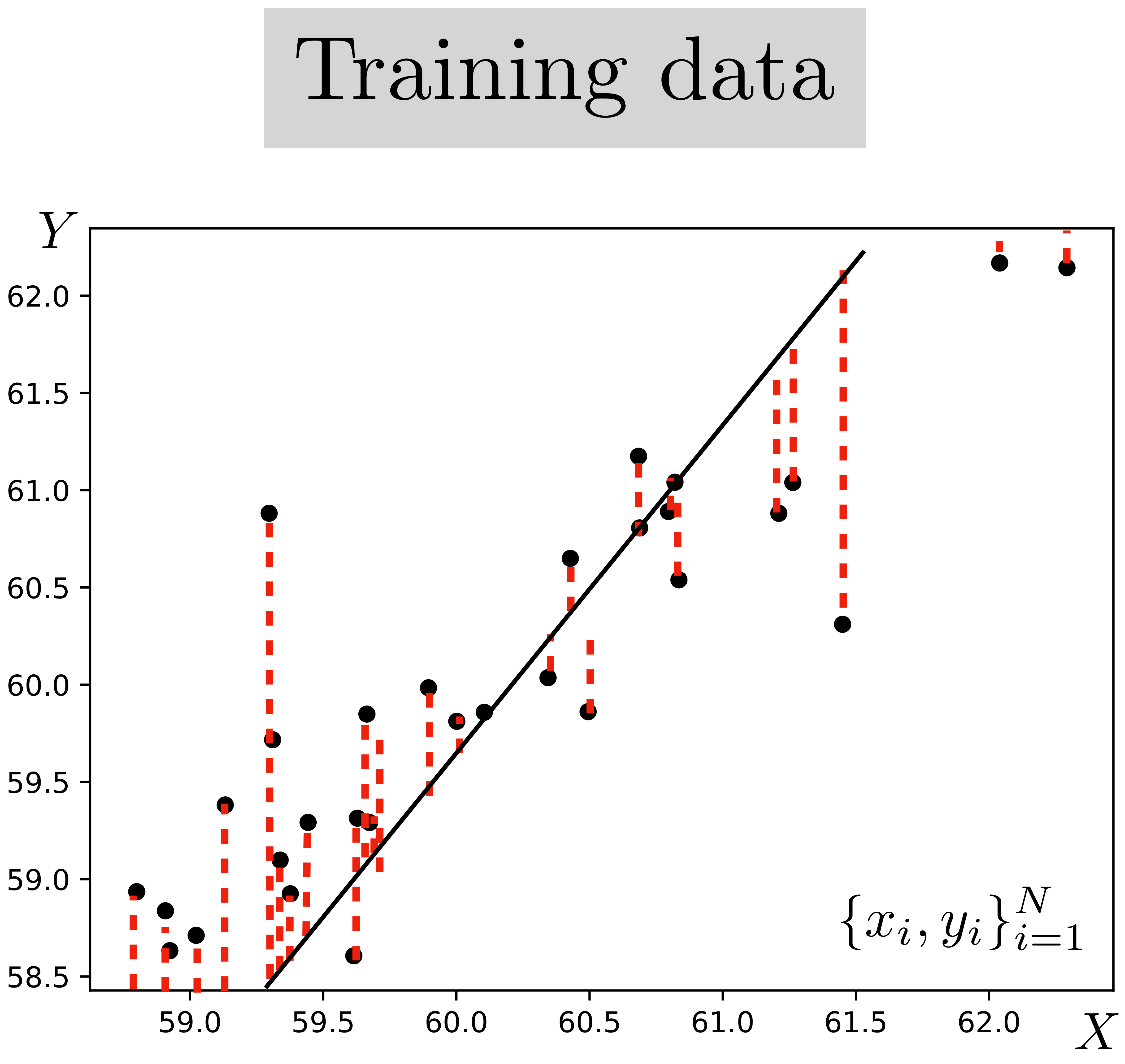


Search for the **parameters**, $\theta = \{\theta_0, \theta_1\}$, that best fit the data.

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

Best fit in what sense?

Source: Isola, Torralba, Freeman



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

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

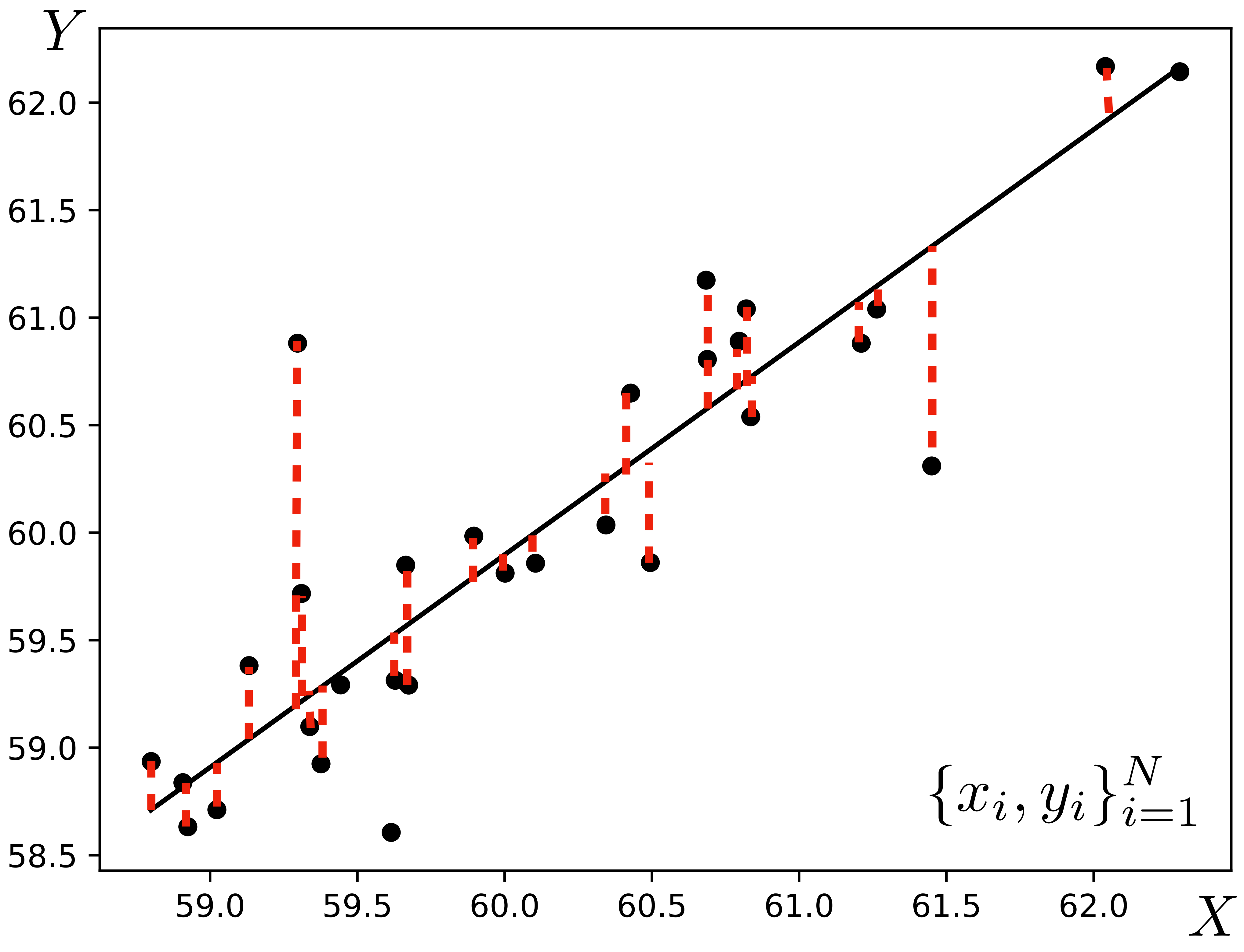
Best fit in what sense?

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.

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

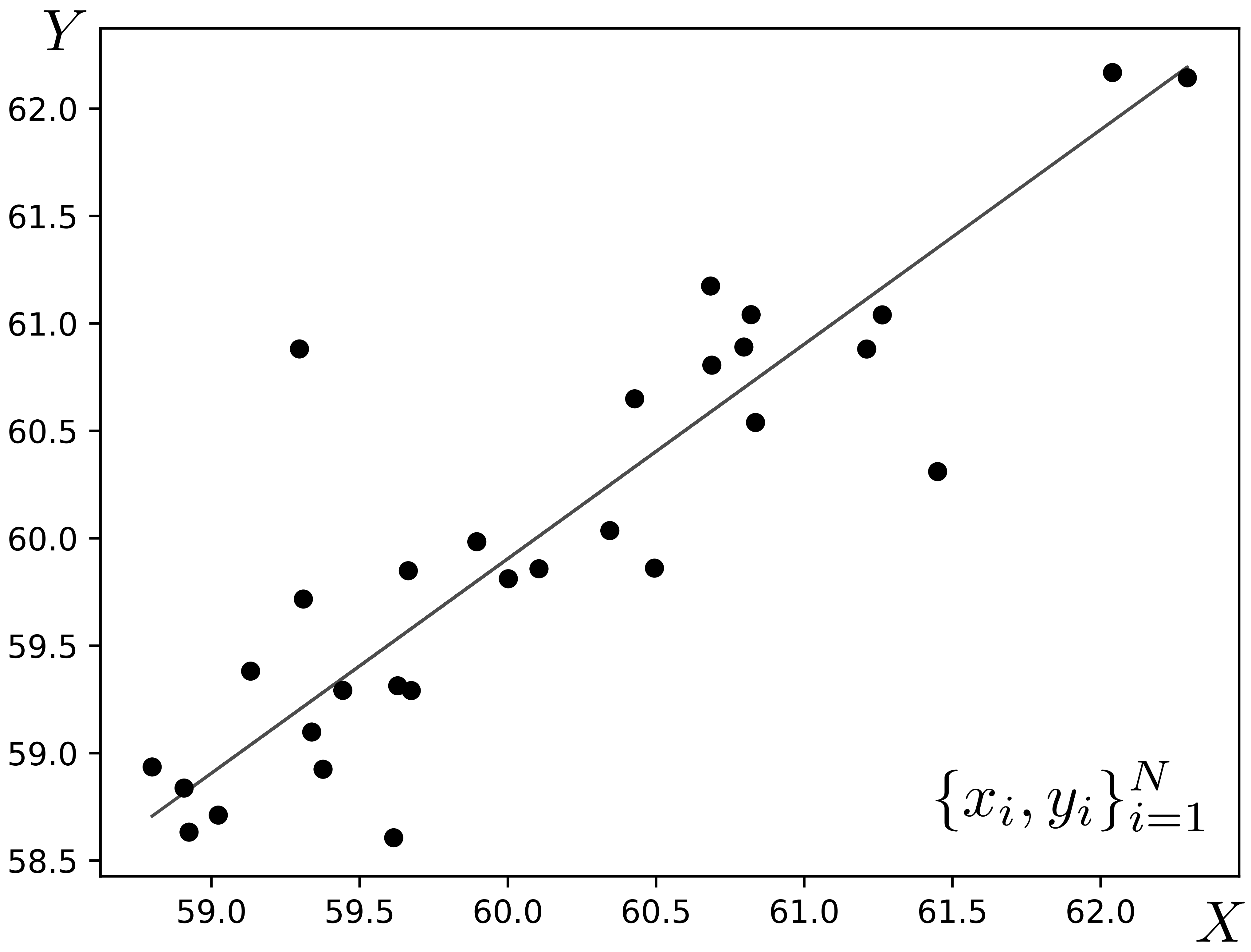
Best fit in what sense?

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



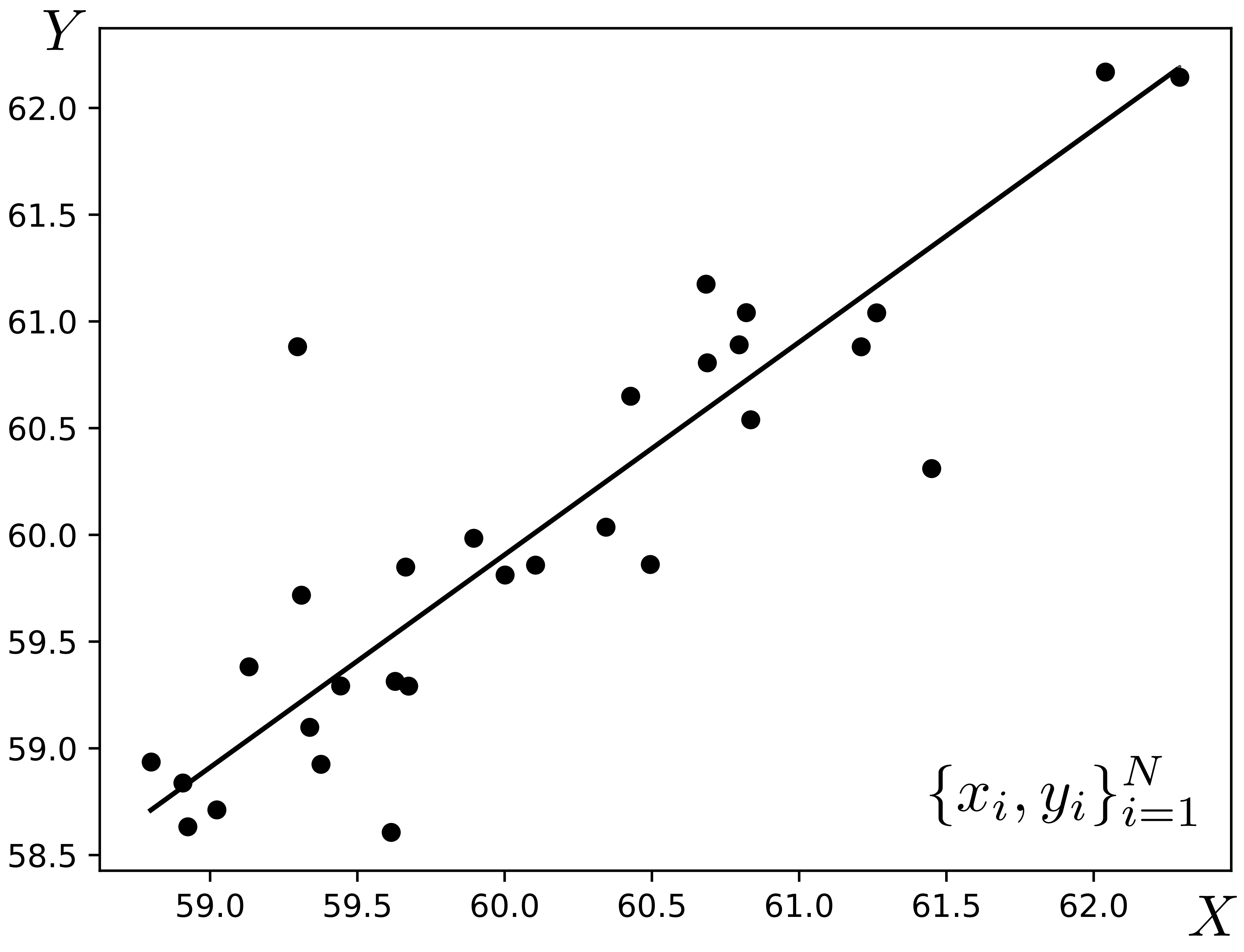
Complete learning problem:

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

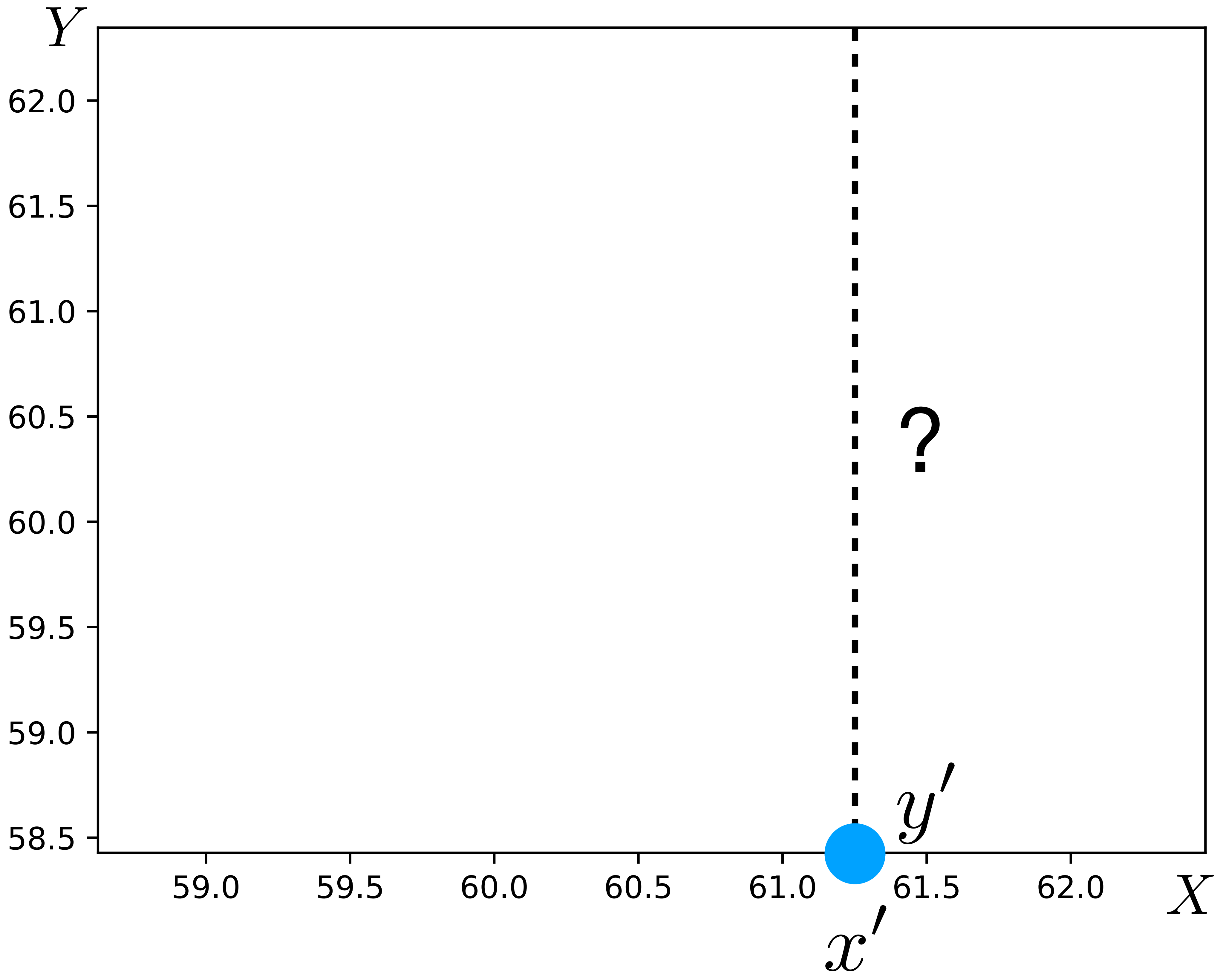
$$= \arg\min_\theta \sum_{i=1}^N (\theta_1 x_i + \theta_0 - y_i)^2$$

Source: Isola, Torralba, Freeman

Training data

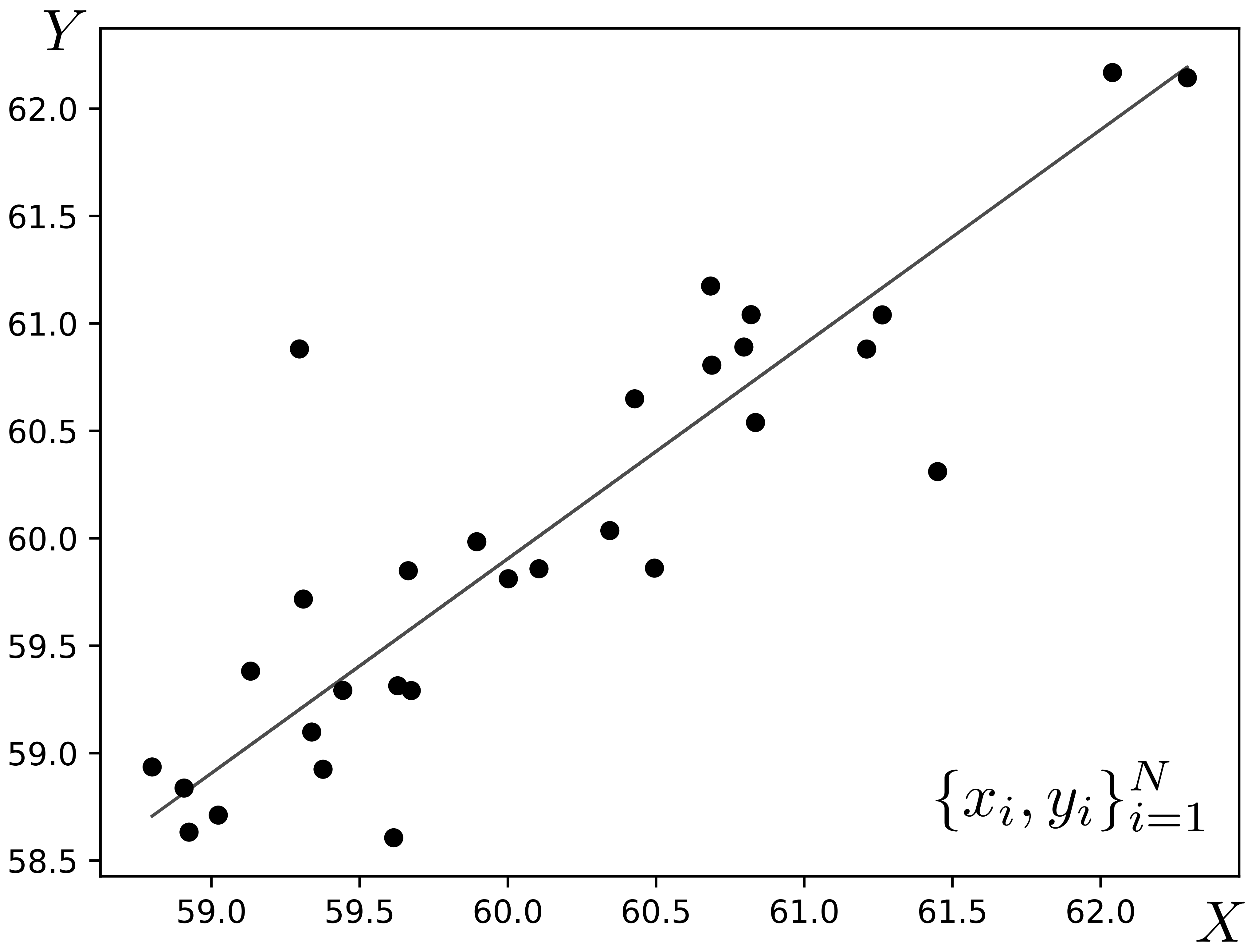


Test query

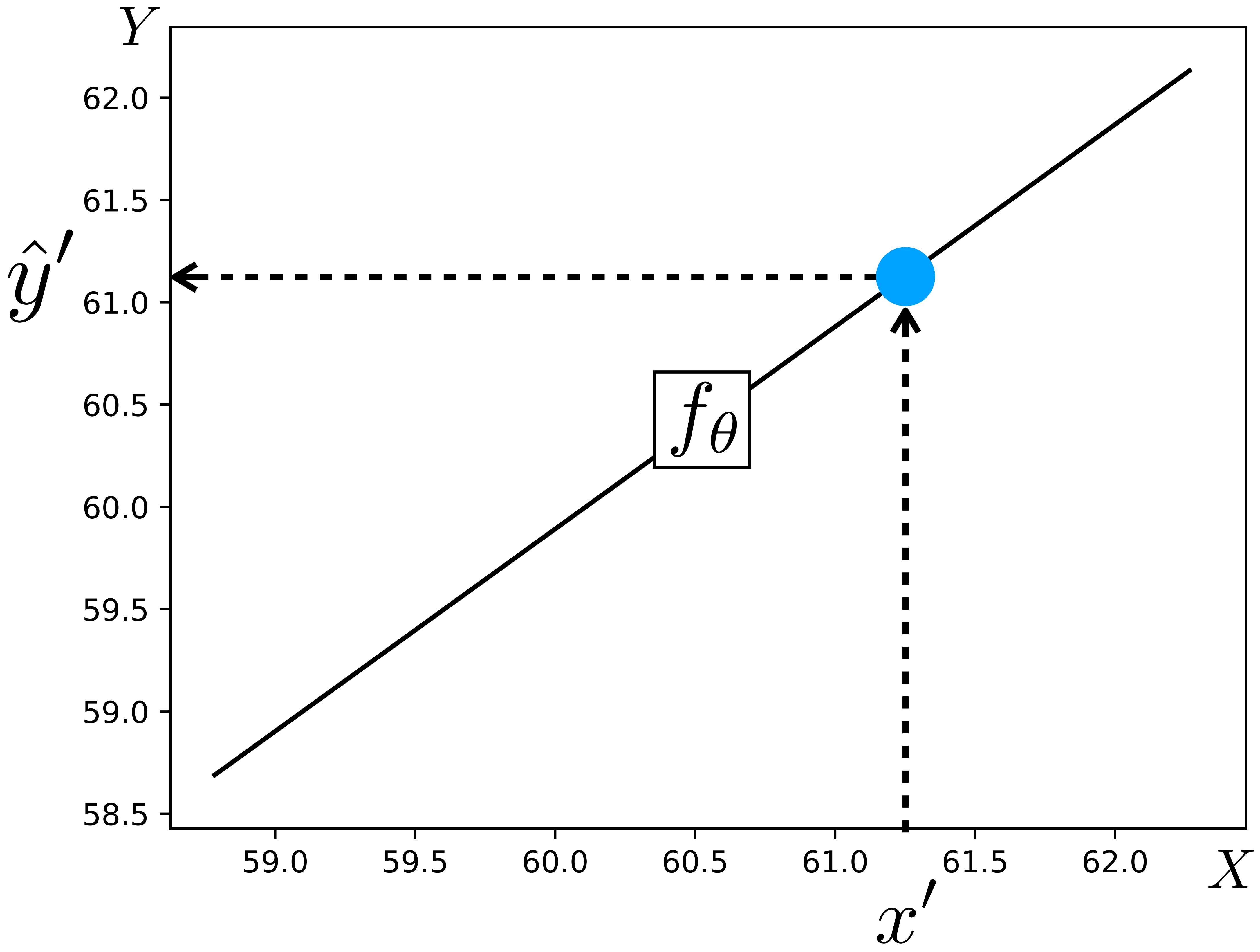


Source: Isola, Torralba, Freeman

Training data



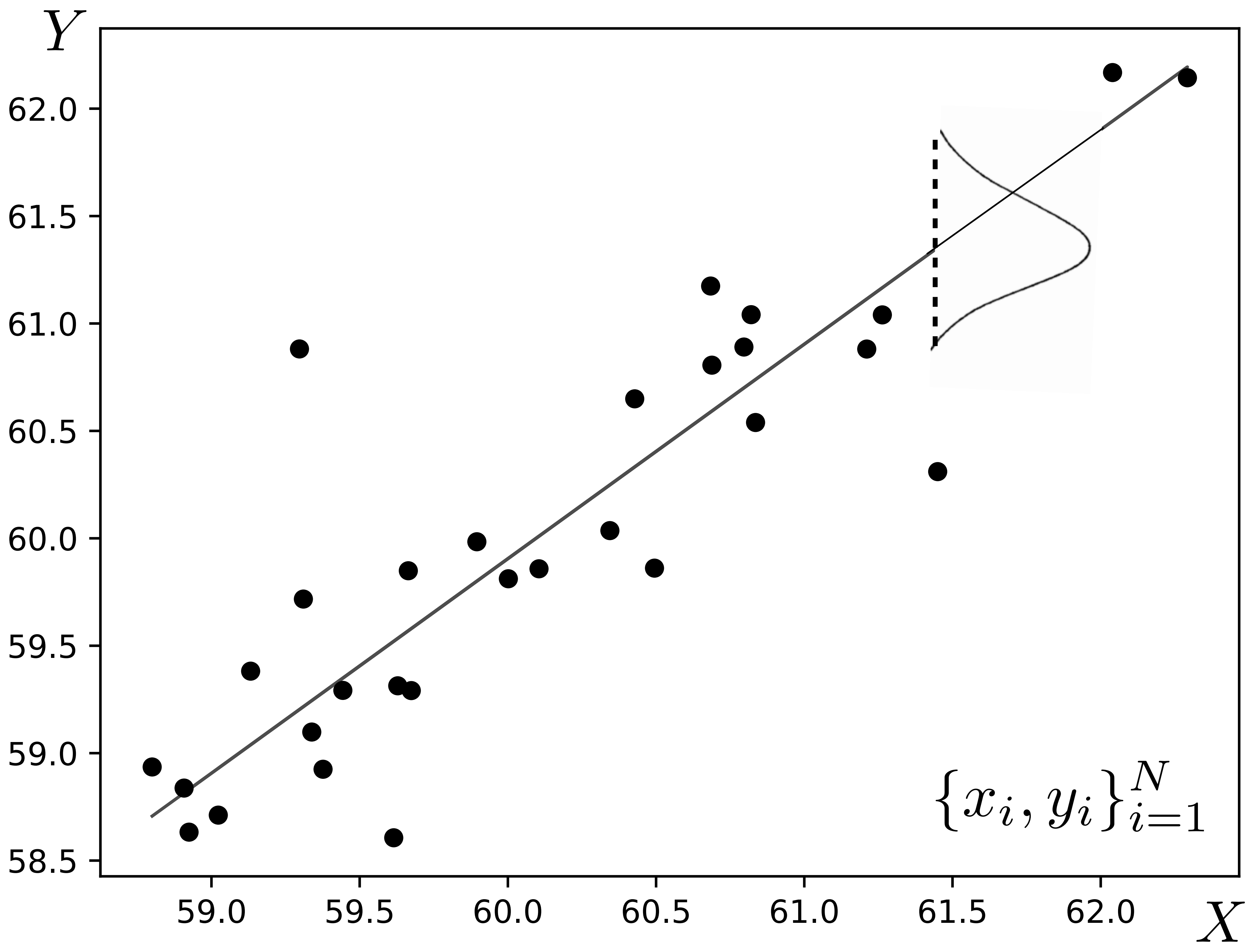
Test query



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

Source: Isola, Torralba, Freeman

Why is squared error a good objective?



Follows from two *modeling assumptions*

1. Observed Y's are corrupted by Gaussian noise:

$$Y \approx f_\theta(X)$$

$$\epsilon = Y - f_\theta(X) \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

$$Y = f_\theta(X) + \epsilon$$

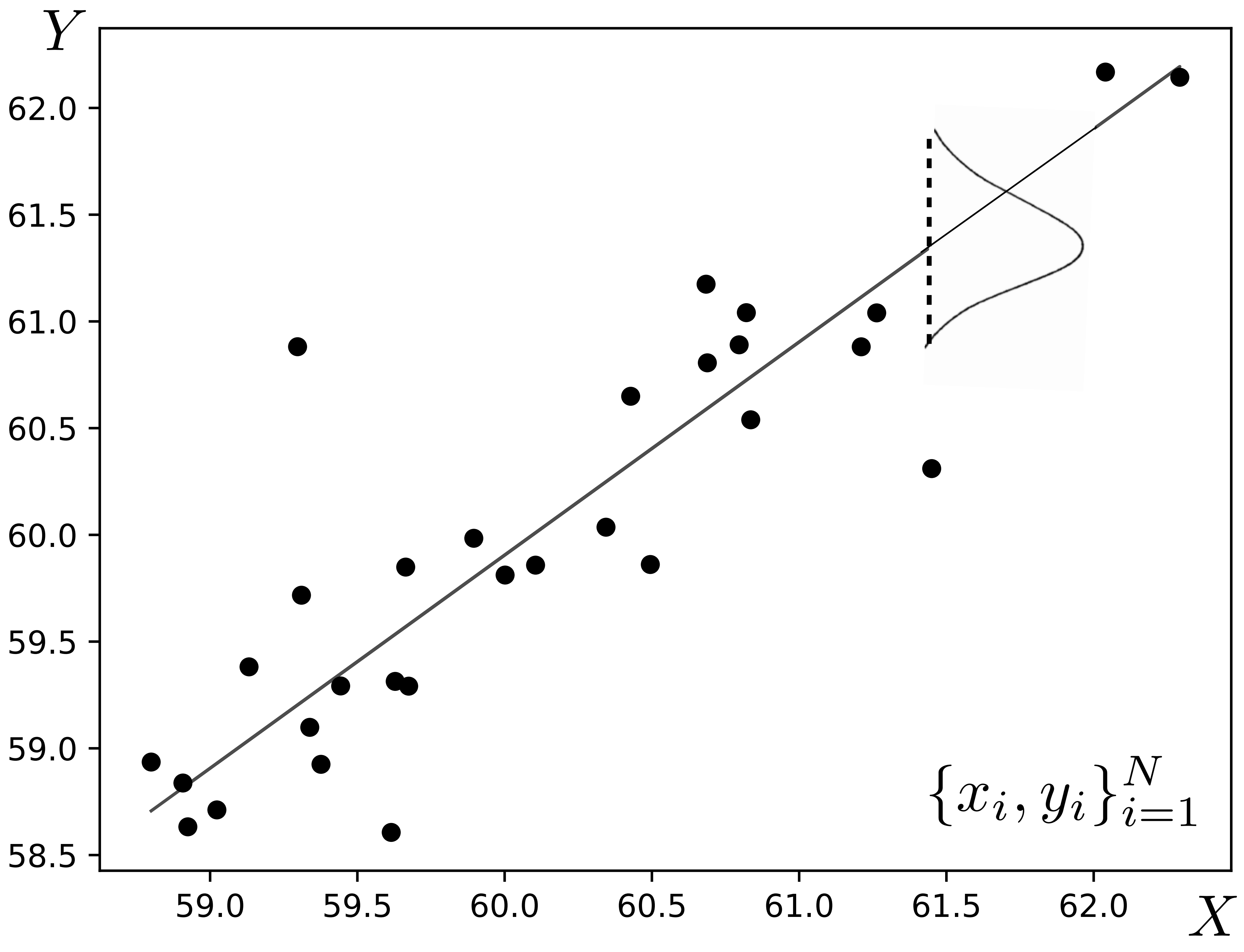
$$P_\theta(Y = y | X = x) \propto \exp \frac{-(y - f_\theta(x))^2}{2\sigma^2}$$

2. Training datapoints are independently and identically distributed (**iid**):

$$P_\theta(\{y_i\}_{i=1}^N | \{x_i\}_{i=1}^N) = \prod_{i=1}^N P_\theta(y_i | x_i)$$

Source: Isola, Torralba, Freeman

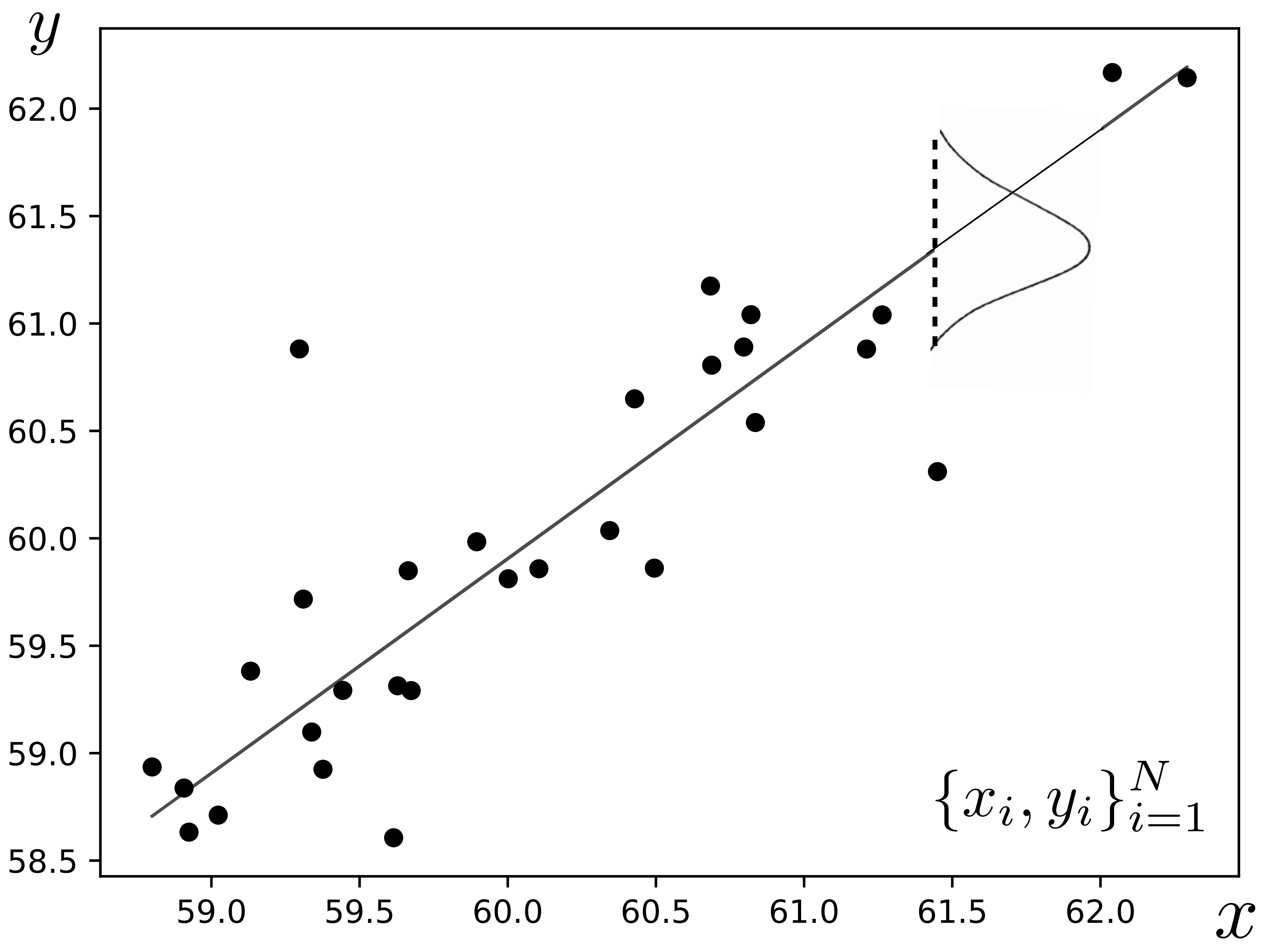
Why is squared error a good objective?



$$
\begin{aligned}
\theta^* &= \arg\max_\theta \prod_{I=1}^{N} P_\theta(y_i|x_i) \\
&= \arg\max_\theta \prod_{I=1}^{N} \exp - \frac{(y - f_\theta(x))^2}{2\sigma^2} \\
&= \arg\max_\theta \log \prod_{I=1}^{N} \exp - \frac{(y - f_\theta(x))^2}{2\sigma^2} \\
&= \arg\max_\theta \sum_{i=1}^{N} - \frac{(y - f_\theta(x))^2}{2\sigma^2} \\
&= \arg\min_\theta \sum_{i=1}^{N} \frac{(y - f_\theta(x))^2}{2\sigma^2} \\
&= \arg\min_\theta \sum_{i=1}^{N} (y - f_\theta(x))^2
\end{aligned}
$$

Source: Isola, Torralba, Freeman

Why is squared error a good objective?



Under the assumption that the data is distributed as:

$$Y = f_\theta(X) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

$$P_\theta(\{y_i\}_{i=1}^N | \{x_i\}_{i=1}^N) = \prod_{i=1}^N P_\theta(y_i|x_i)$$

The most probable estimate of θ is:

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

This is called the **maximum likelihood** estimator of θ.

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

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

We'll use methods from **optimization**.

Source: Isola, Torralba, Freeman

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

In the linear case:

Learning problem

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

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

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

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

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

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

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

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

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

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)

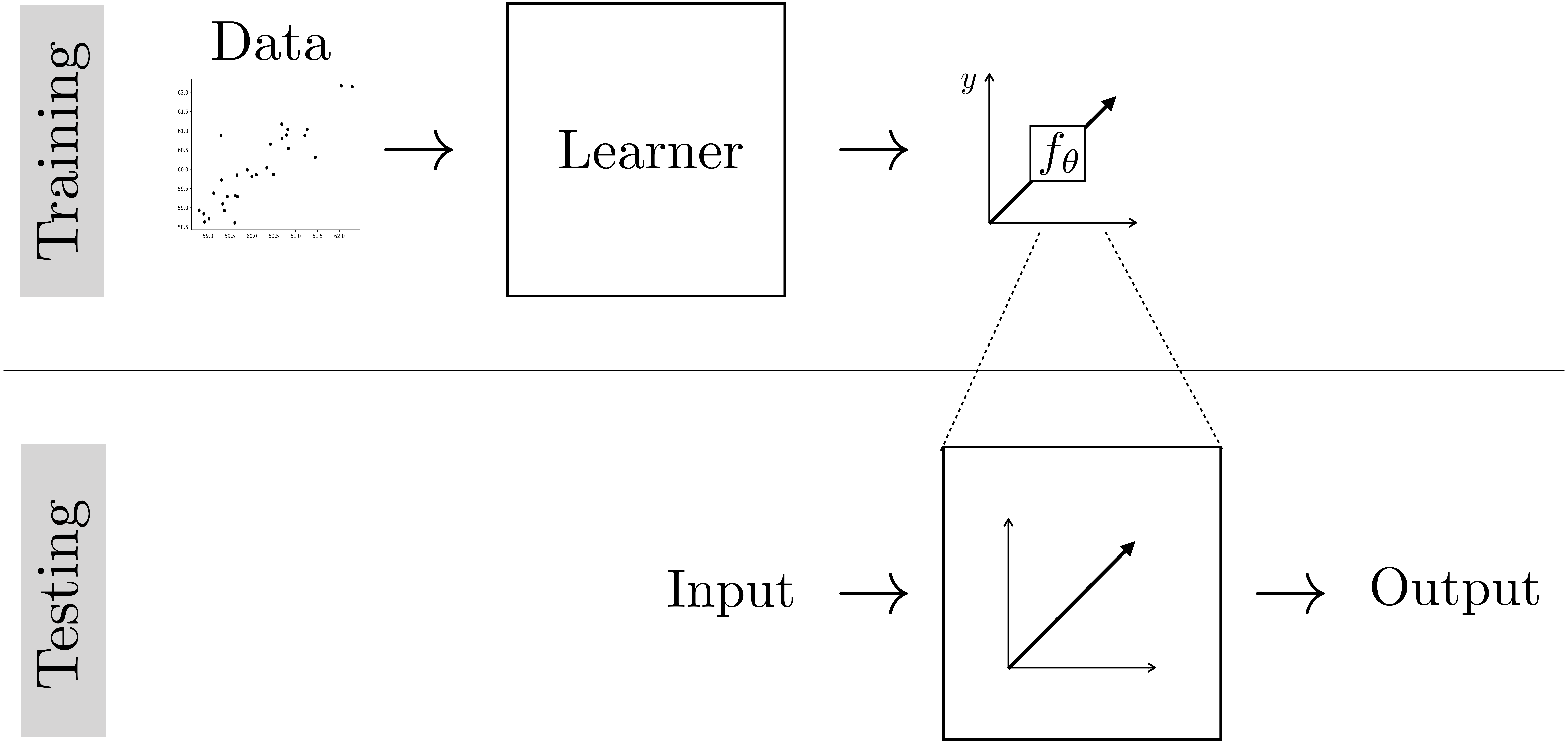
$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i)$$

Training data

Hypothesis space

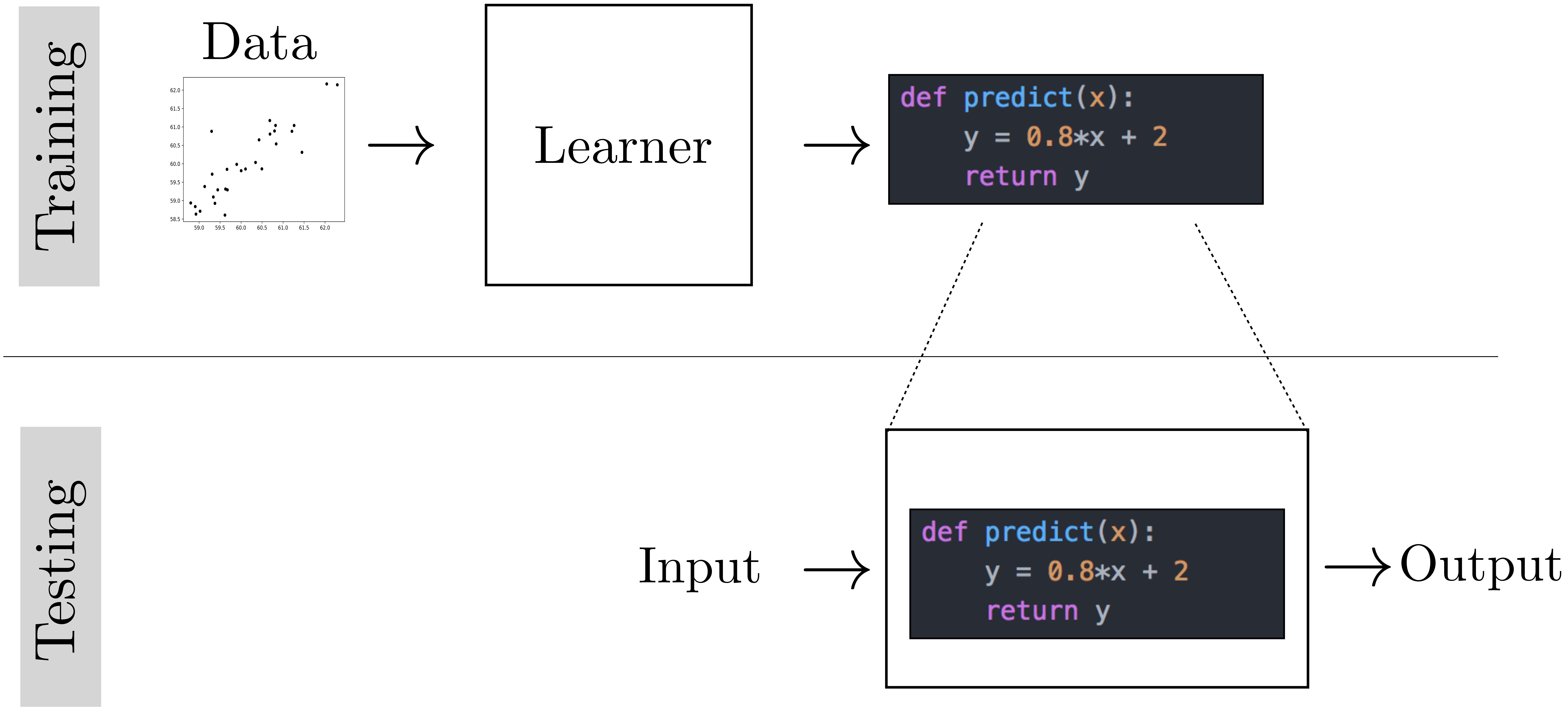
Source: Isola, Torralba, Freeman

Example 1: Linear least squares



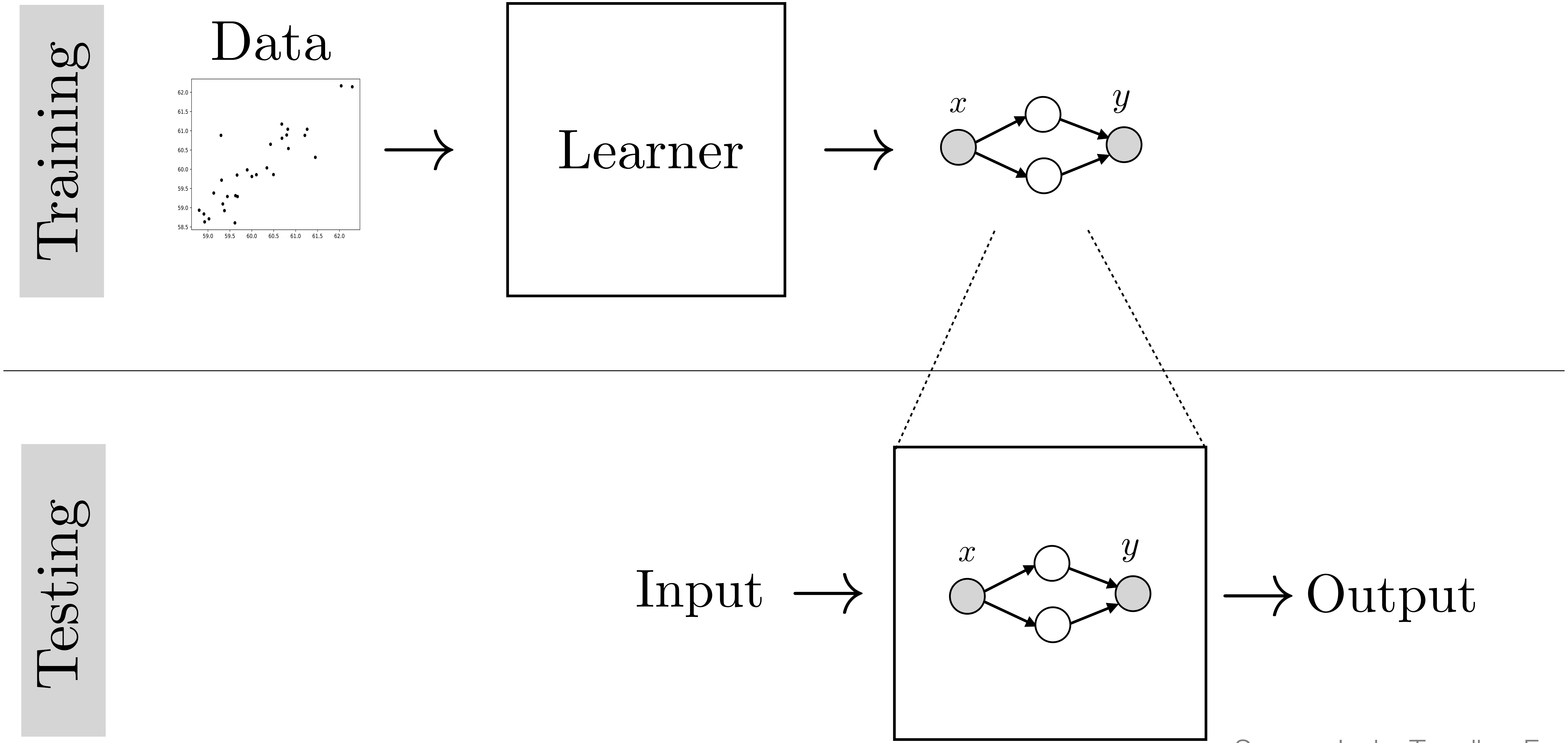
Source: Isola, Torralba, Freeman

Example 2: Program induction



Source: Isola, Torralba, Freeman

Example 3: Deep learning



Source: Isola, Torralba, Freeman

Learning for vision

Questions:

1. How do you represent the input and output?

2. What is the objective?

3. What is the hypothesis space? (e.g., linear, polynomial, neural net?)

4. How do you optimize? (e.g., gradient descent, Newton's method?)

5. What data do you train on?

Source: Isola, Torralba, Freeman

Learning for vision

Questions:

1. How do you represent the input and output?

2. What is the objective?

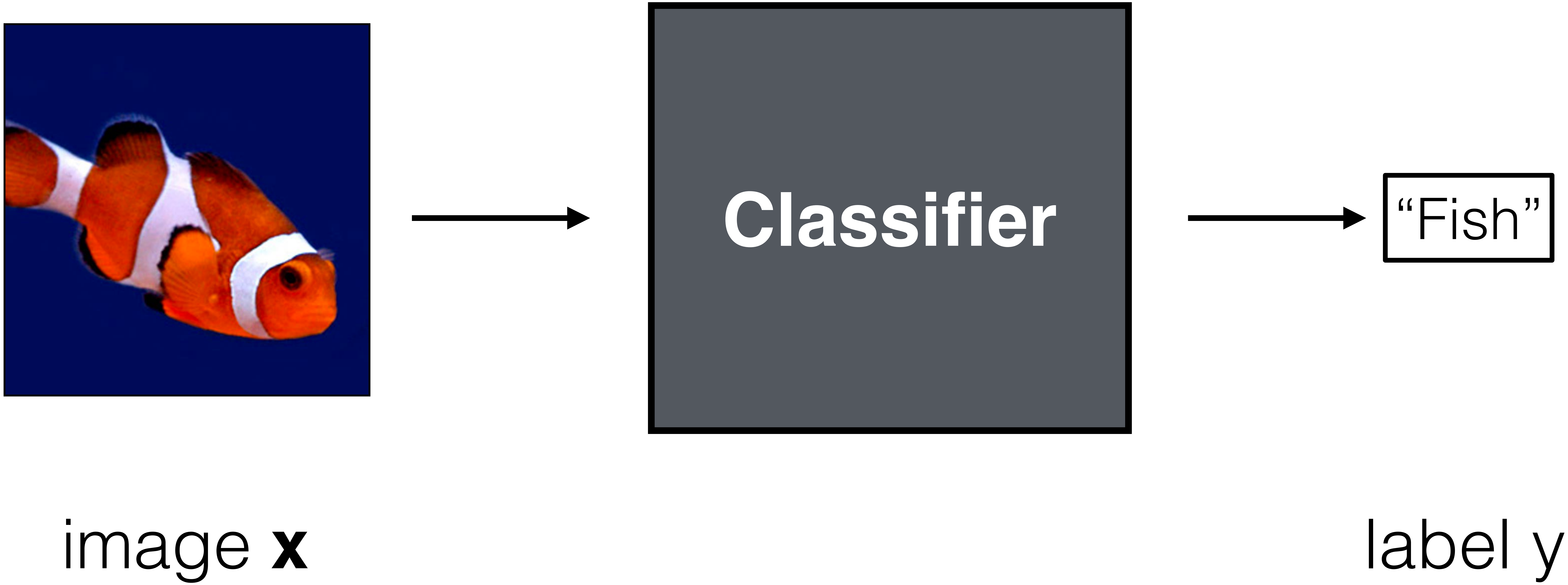
3. What is the hypothesis space? (e.g., linear, polynomial, neural net?)

4. How do you optimize? (e.g., gradient descent, Newton's method?)

5. What data do you train on?

Source: Isola, Torralba, Freeman

Image classification



Source: Isola, Torralba, Freeman

Image classification

Classifier

“Fish”

image **x**

label y

Source: Isola, Torralba, Freeman

Image classification

Classifier

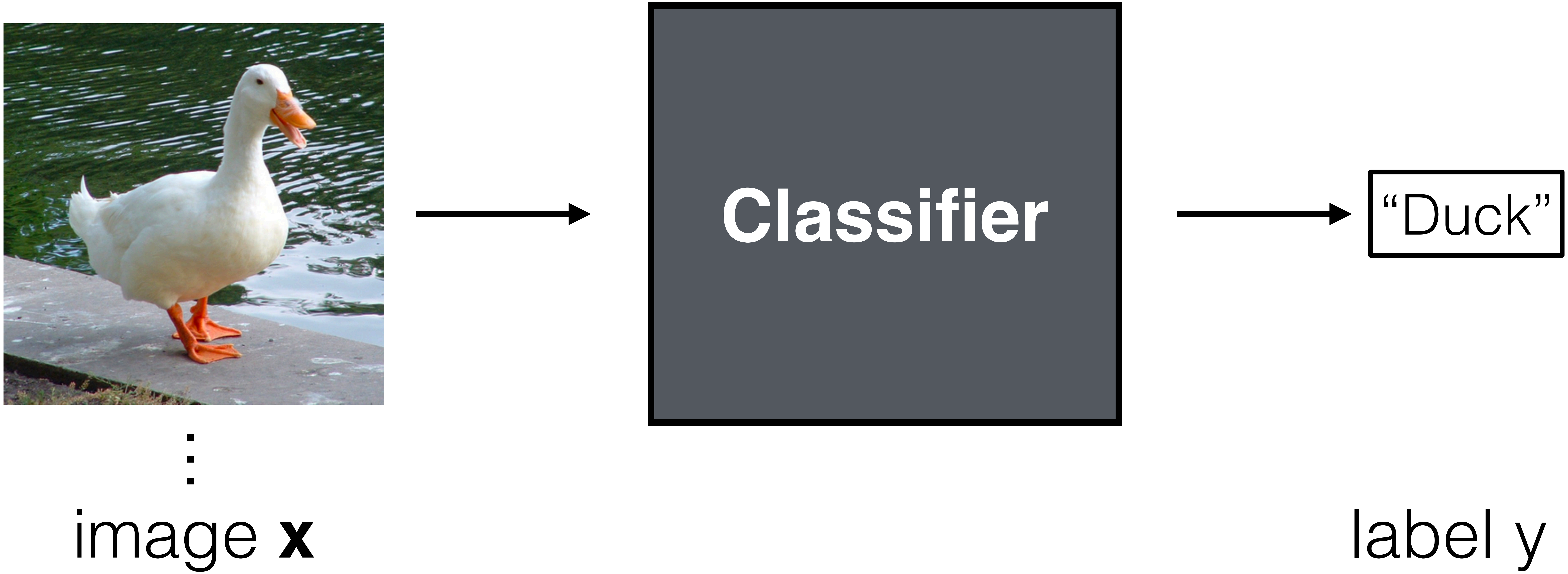
"Fish"

image **x**

label y

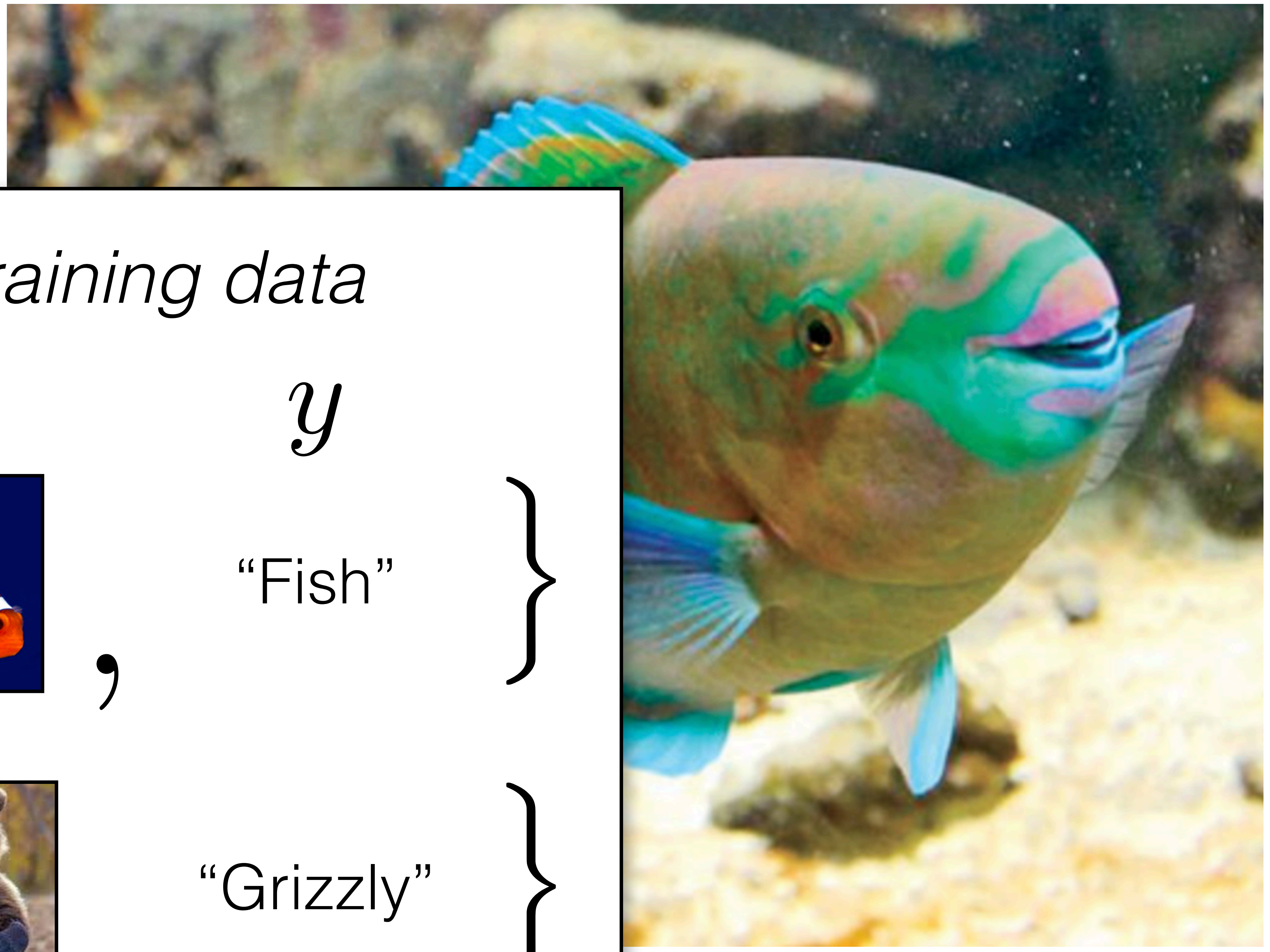
Source: Isola, Torralba, Freeman

Image classification



Source: Isola, Torralba, Freeman

$\mathbf{x}$



Training data

$\mathbf{x}$	y
{  ,	"Fish" }
{  ,	"Grizzly" }
{  ,	"Chameleon" }
⋮	

f

y

"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?

Source: Isola, Torralba, Freeman

What should the loss be?

0-1 loss (number of misclassifications)

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \mathbb{1}(\hat{\mathbf{y}} = \mathbf{y}) \longleftarrow$$ discrete; hard to optimize!

Cross entropy

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

Source: Isola, Torralba, Freeman

Ground truth label $\mathbf{y}$

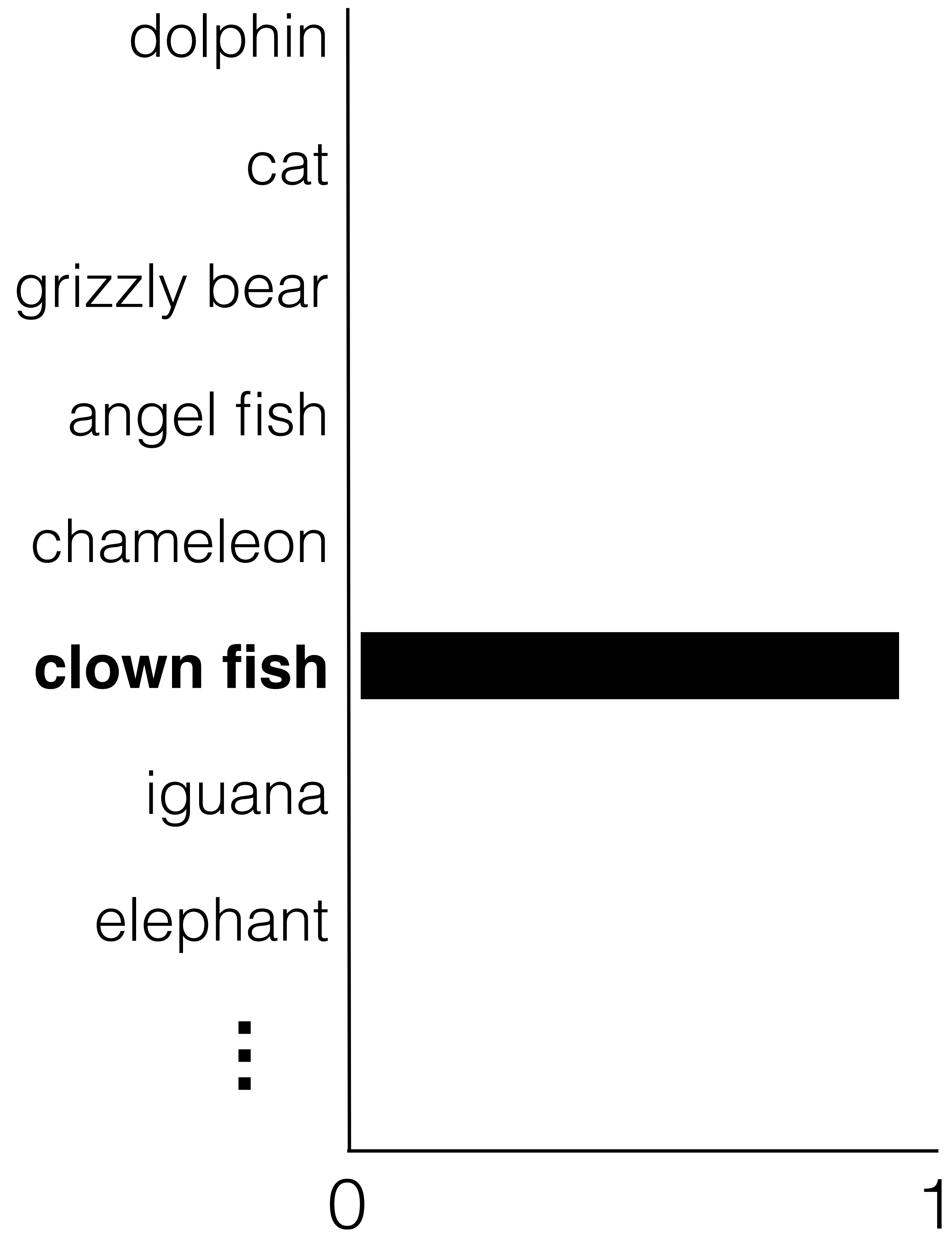
$\mathbf{x}$

[0,0,0,0,0,1,0,0,…]

Source: Isola, Torralba, Freeman

Ground truth label $\mathbf{y}$

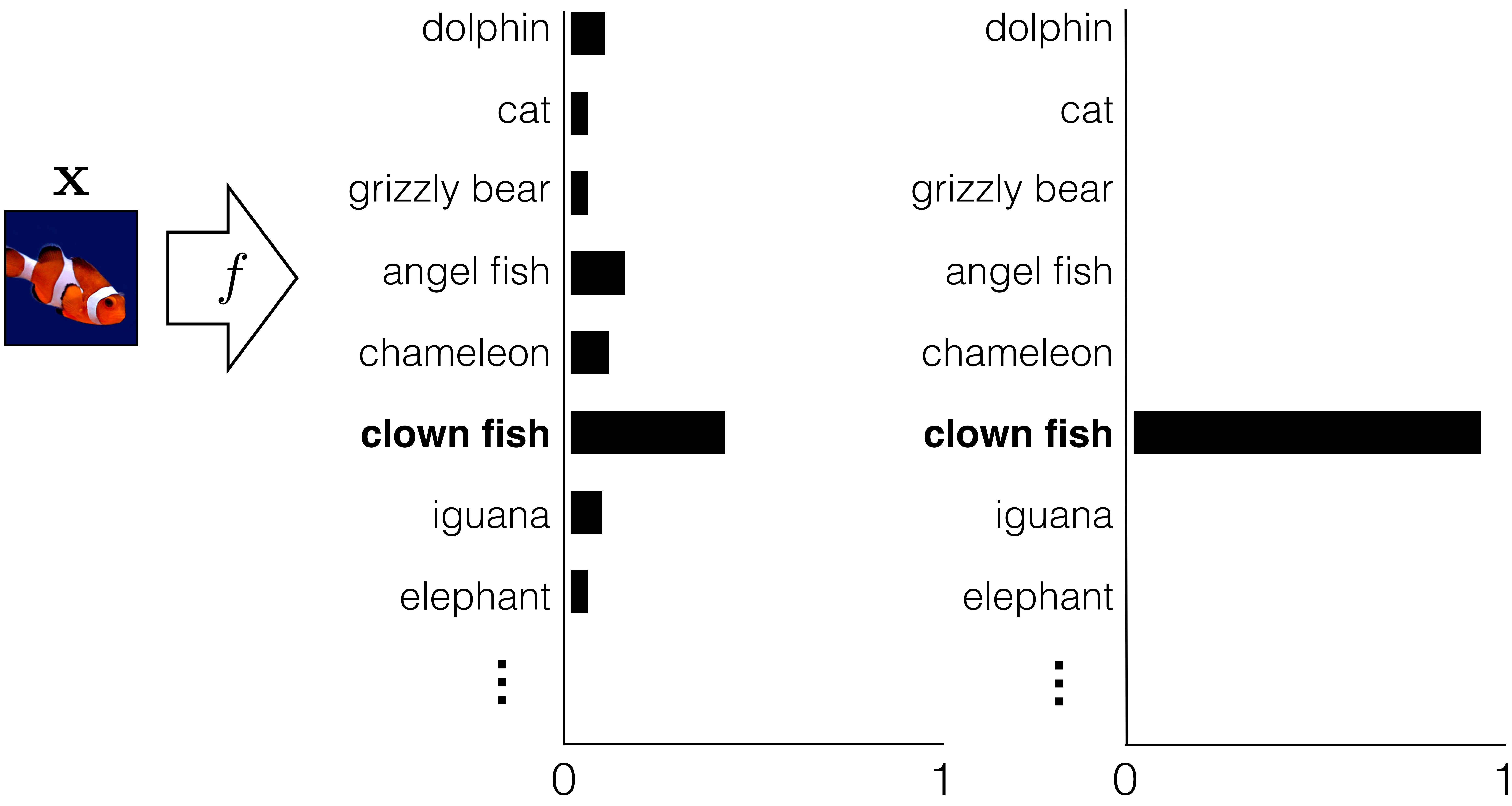
$\mathbf{x}$



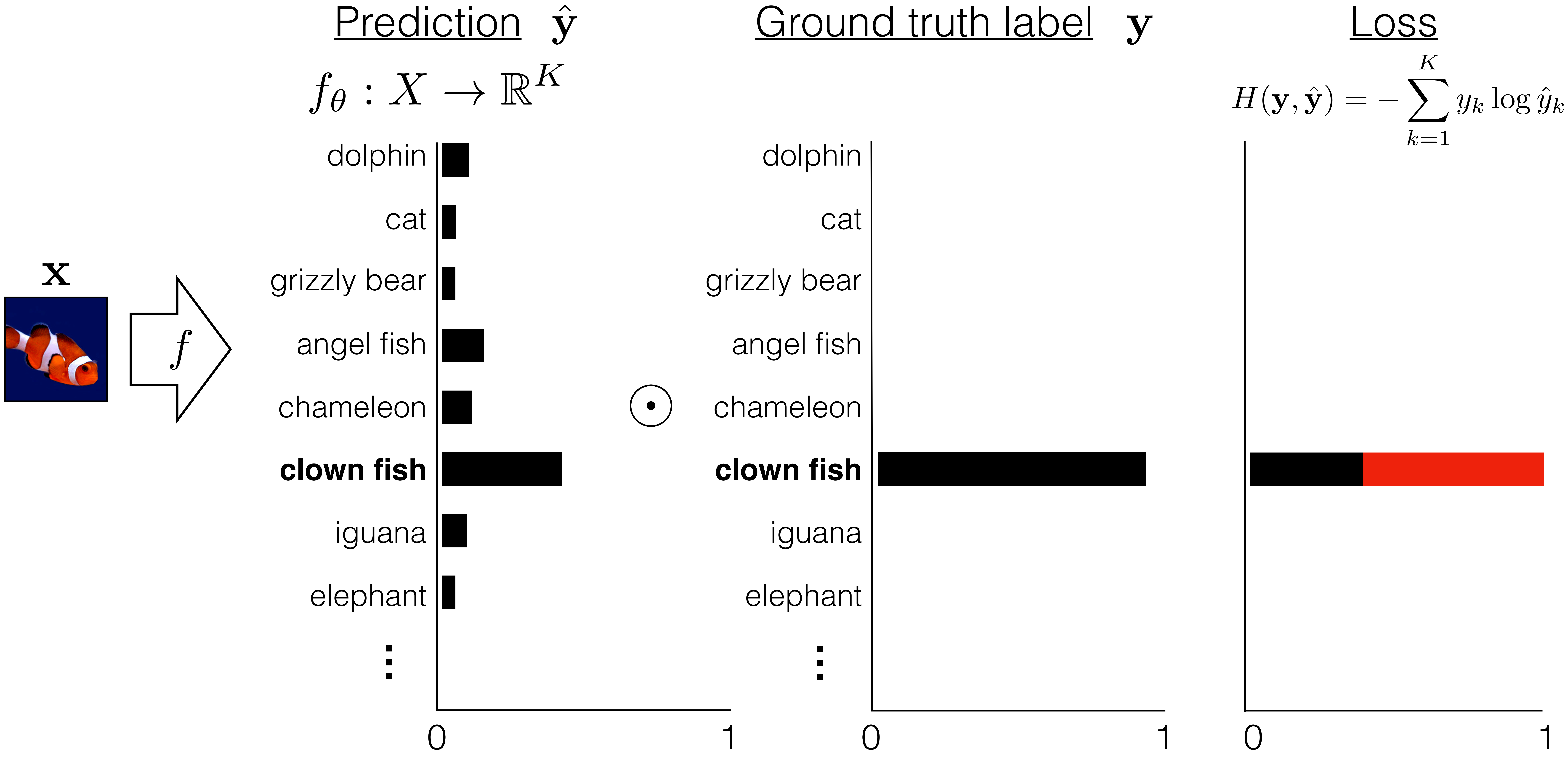
Source: Isola, Torralba, Freeman

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

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



Source: Isola, Torralba, Freeman



Source: Isola, Torralba, Freeman

Source: Isola, Torralba, Freeman

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

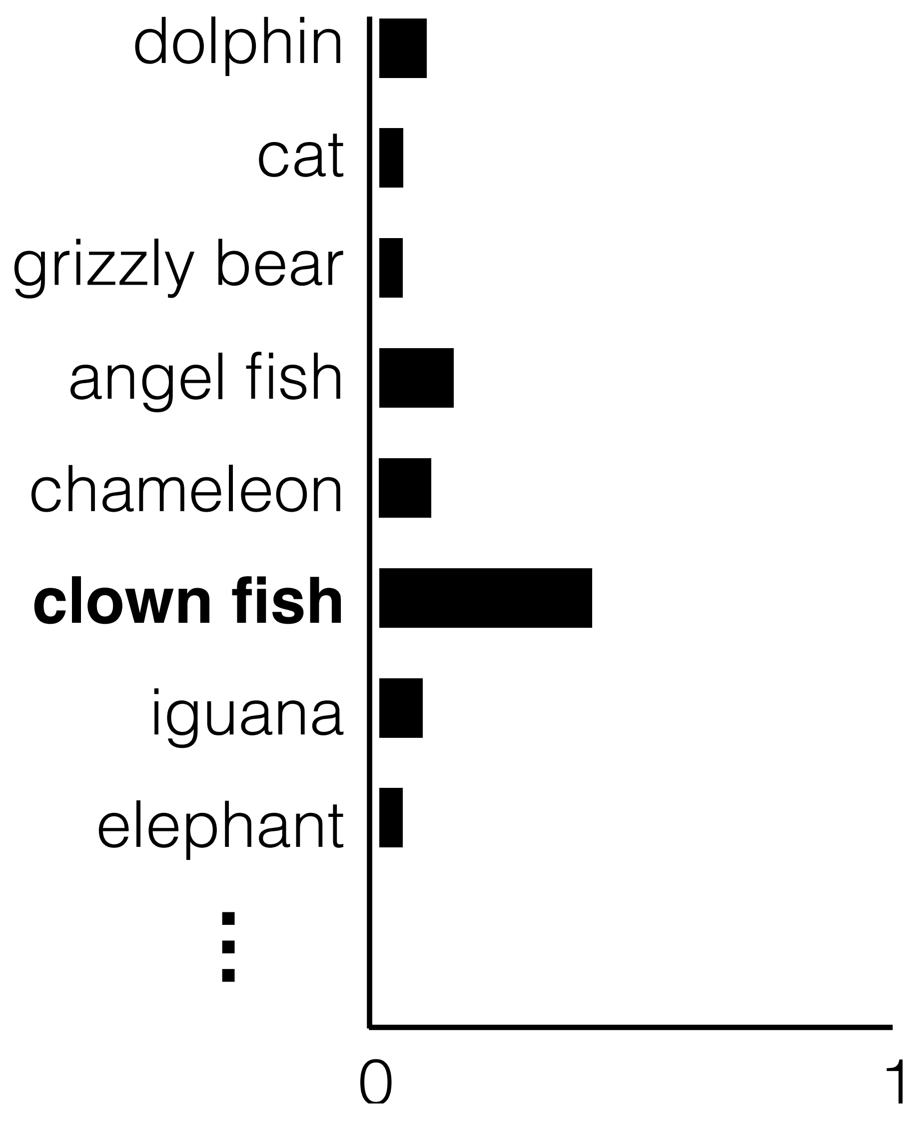
$$f_\theta : X \to \mathbb{R}^K$$

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

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

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

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



Source: Isola, Torralba, Freeman

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

Probabilistic interpretation:

$$\hat{\mathbf{y}} \equiv [P_\theta(Y = 1|X = \mathbf{x}), \ldots, P_\theta(Y = K|X = \mathbf{x})]$$ ⟵ predicted probability of each class given input **x**

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$ ⟵ picks out the -log likelihood of the ground truth class $\mathbf{y}$ under the model prediction $\hat{\mathbf{y}}$

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} H(\mathbf{y}_i, \hat{\mathbf{y}}_i)$$ ⟵ maximum likelihood

Source: Isola, Torralba, Freeman

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

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

$$\mathbf{z} = f_\theta(\mathbf{x})$$

$$\hat{\mathbf{y}} = \texttt{softmax}(\mathbf{z})$$

Data

$\{x_i, y_i\}_{i=1}^N \rightarrow$

Learner

Objective

$$\mathcal{L}(\mathbf{y}, f_\theta(\mathbf{x})) = H(\mathbf{y}, \texttt{softmax}(f_\theta(\mathbf{x})))$$

$\rightarrow f$

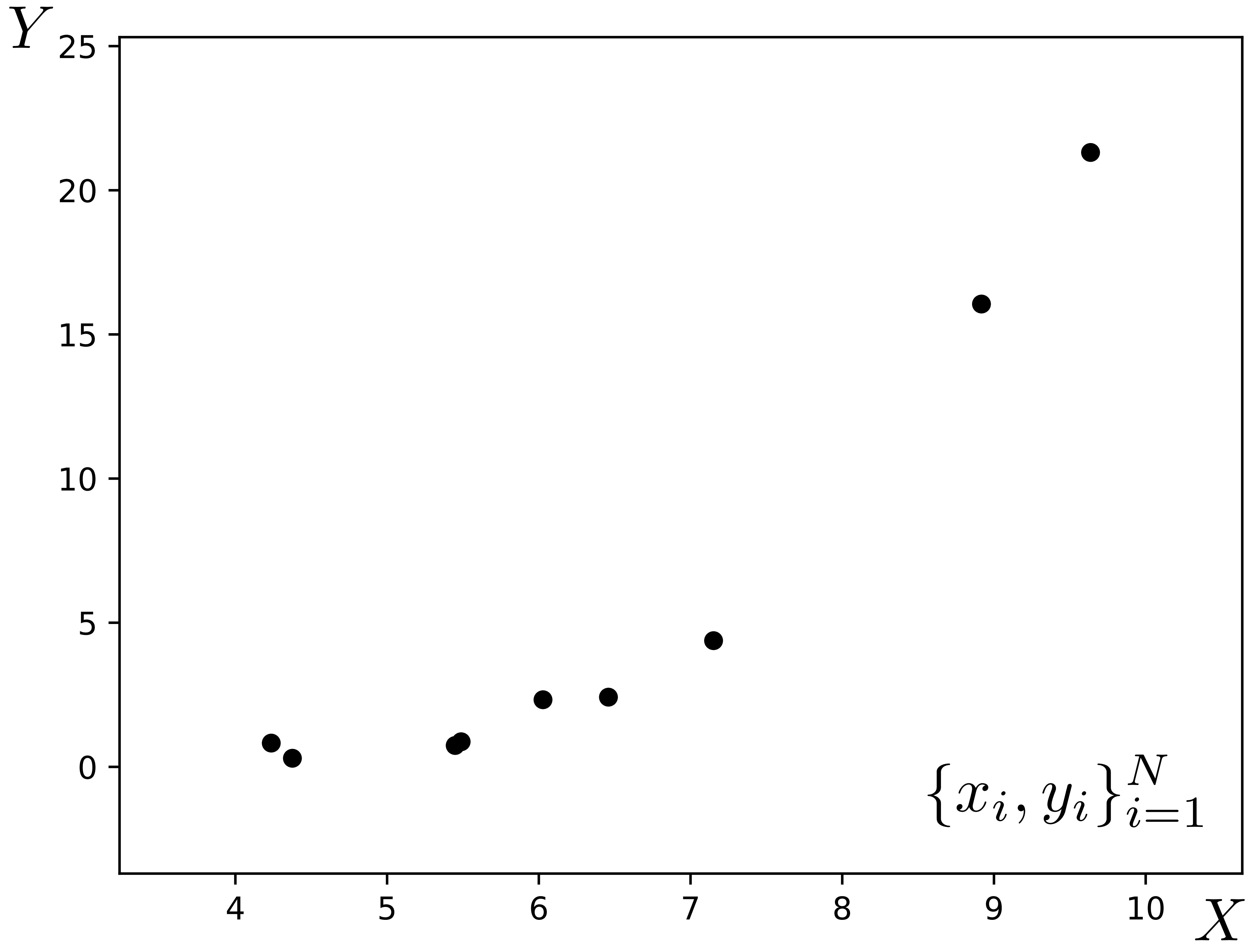
Source: Isola, Torralba, Freeman

The Problem of Generalization

Source: Isola, Torralba, Freeman

Linear regression

Training data

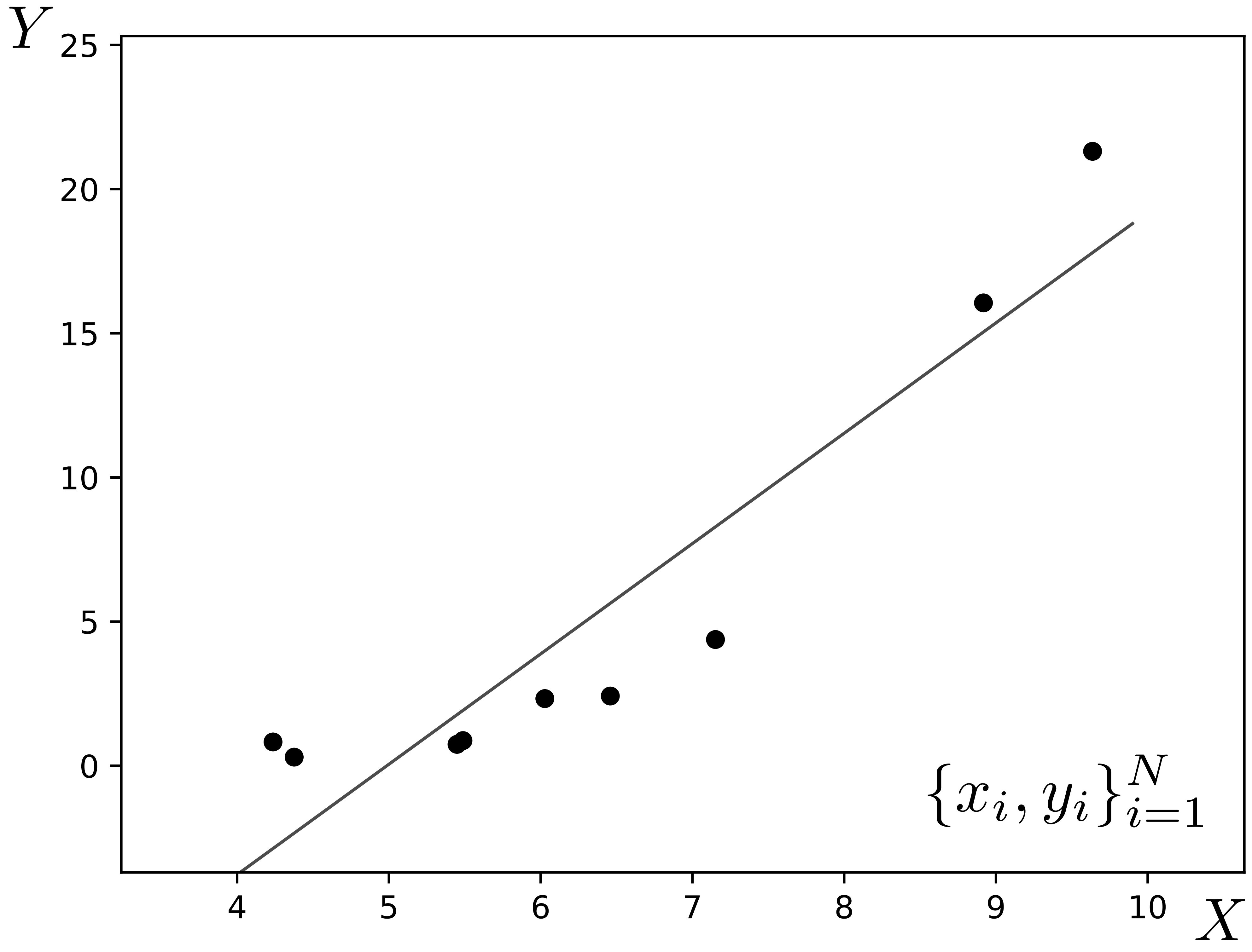


$$f_\theta(x) = \theta_0 + \theta_1 x$$

Source: Isola, Torralba, Freeman

Linear regression

Training data

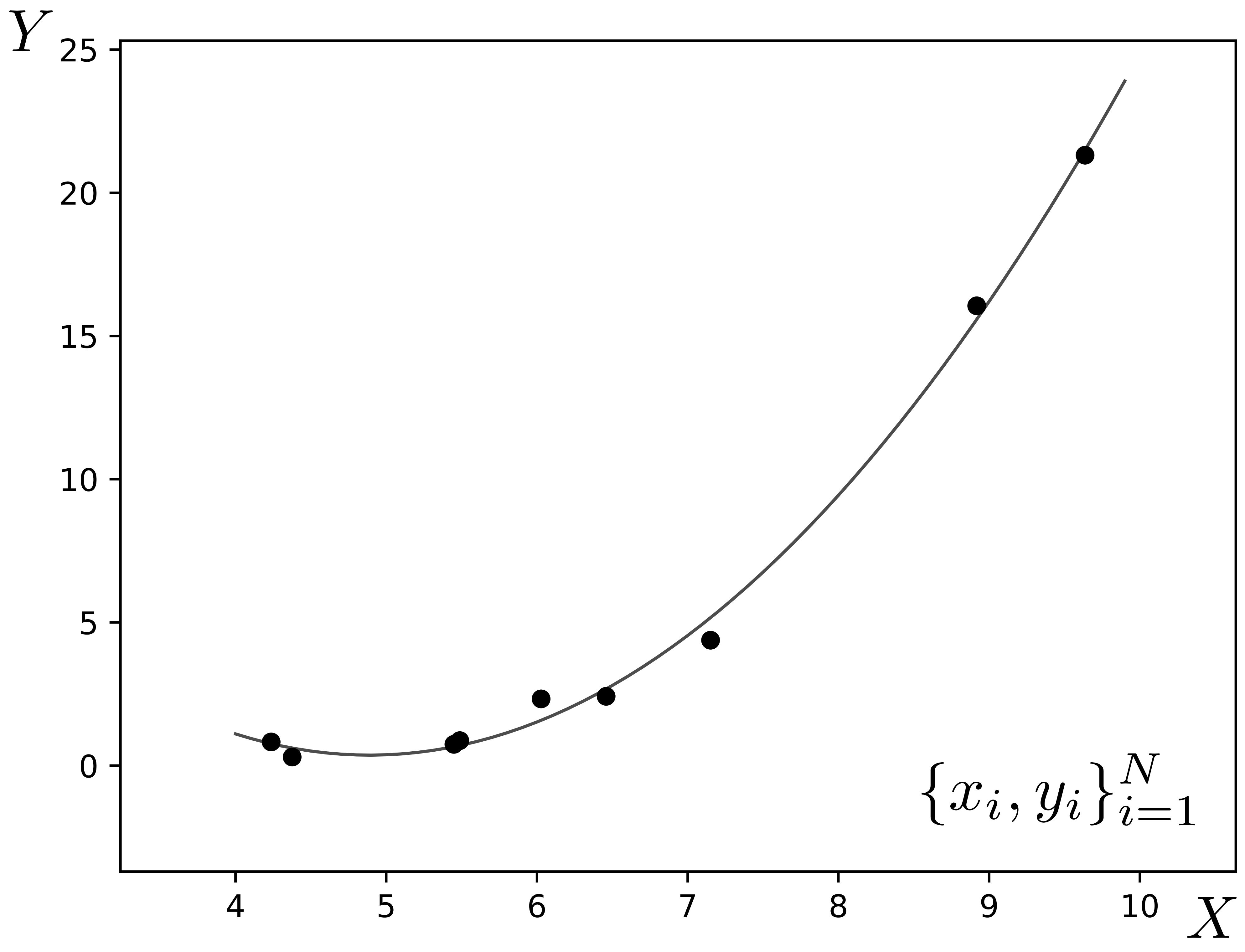


$$f_\theta(x) = \theta_0 + \theta_1 x$$

Source: Isola, Torralba, Freeman

Polynomial regression

Training data



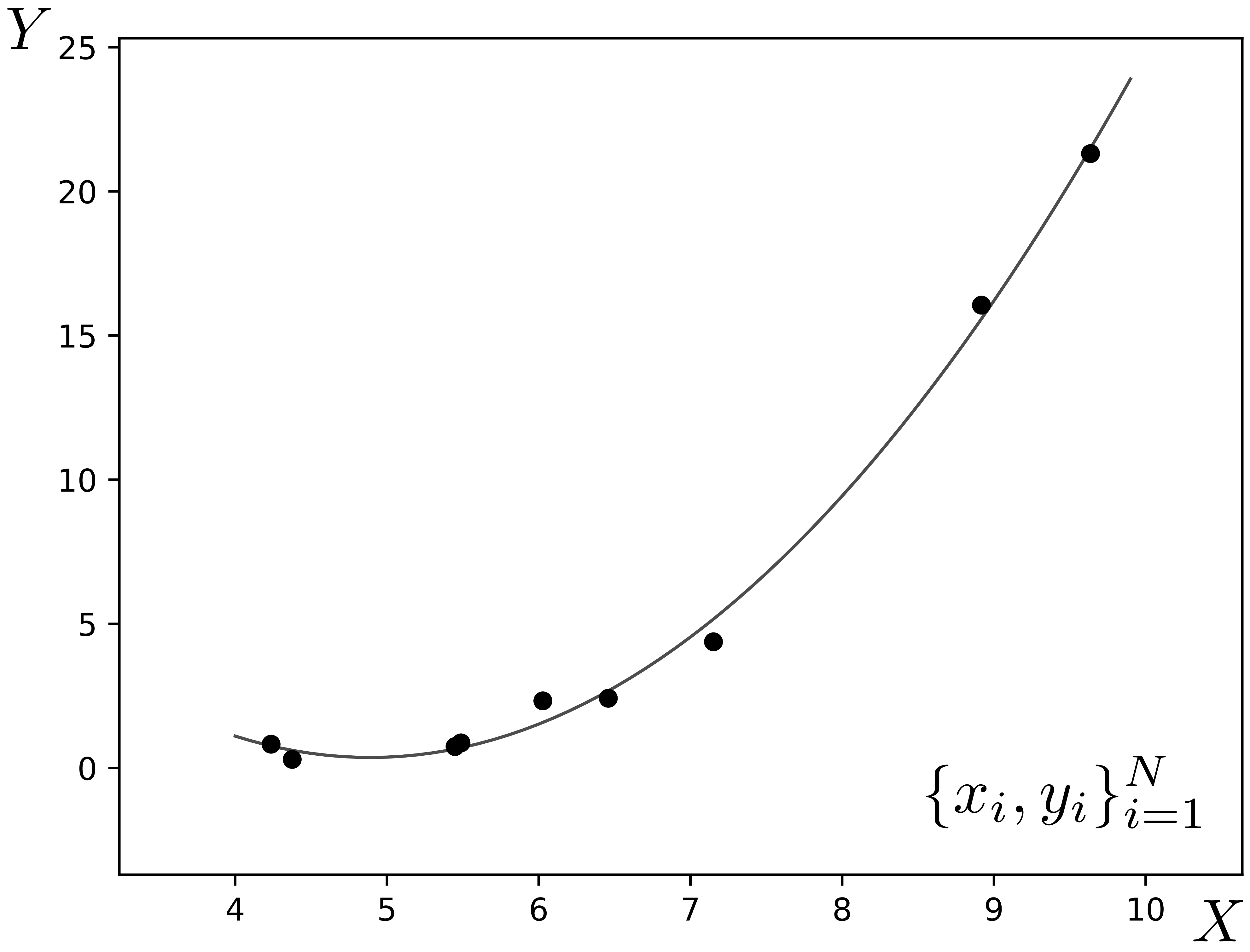
$$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

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

K-th degree polynomial regression

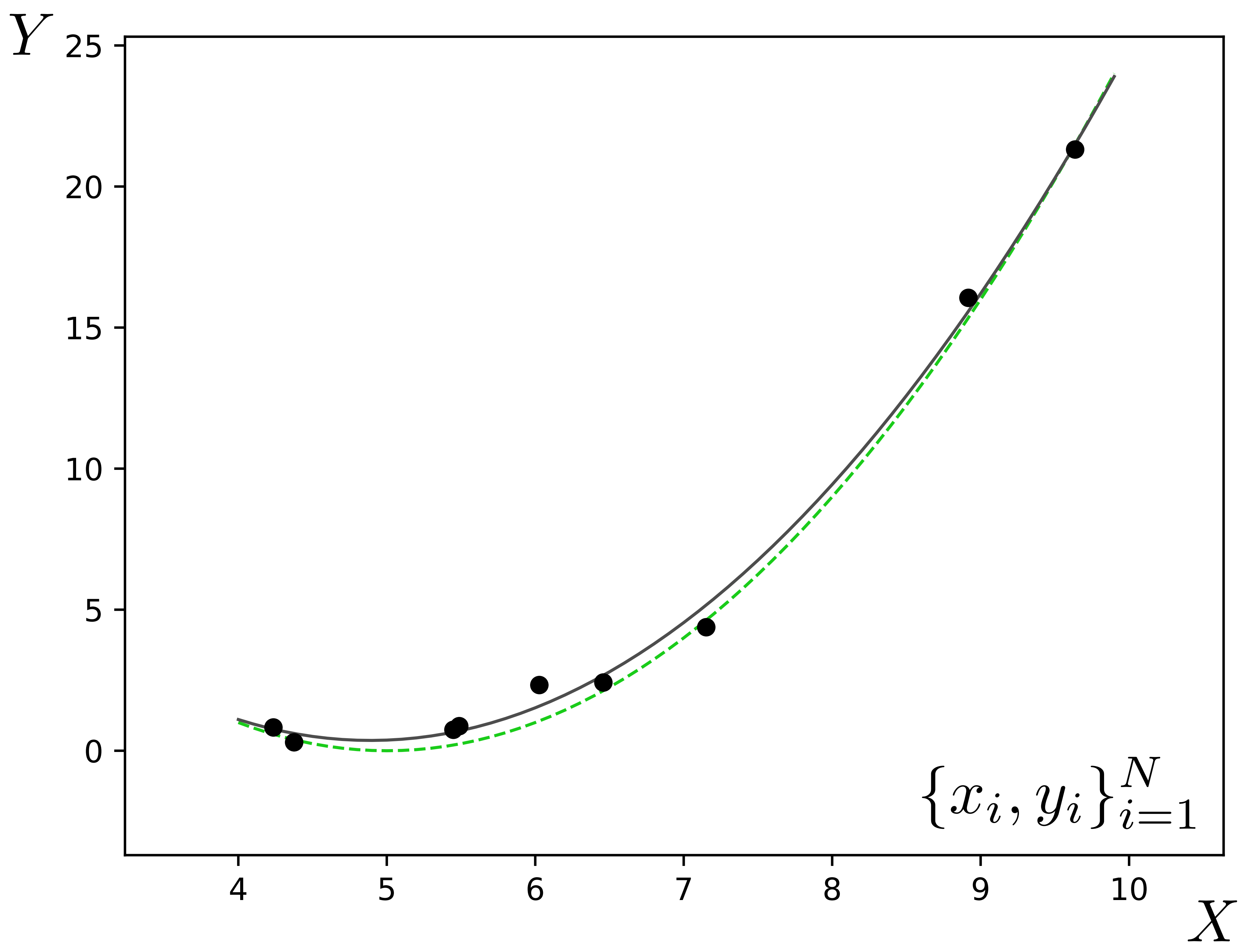
Source: Isola, Torralba, Freeman

Training data

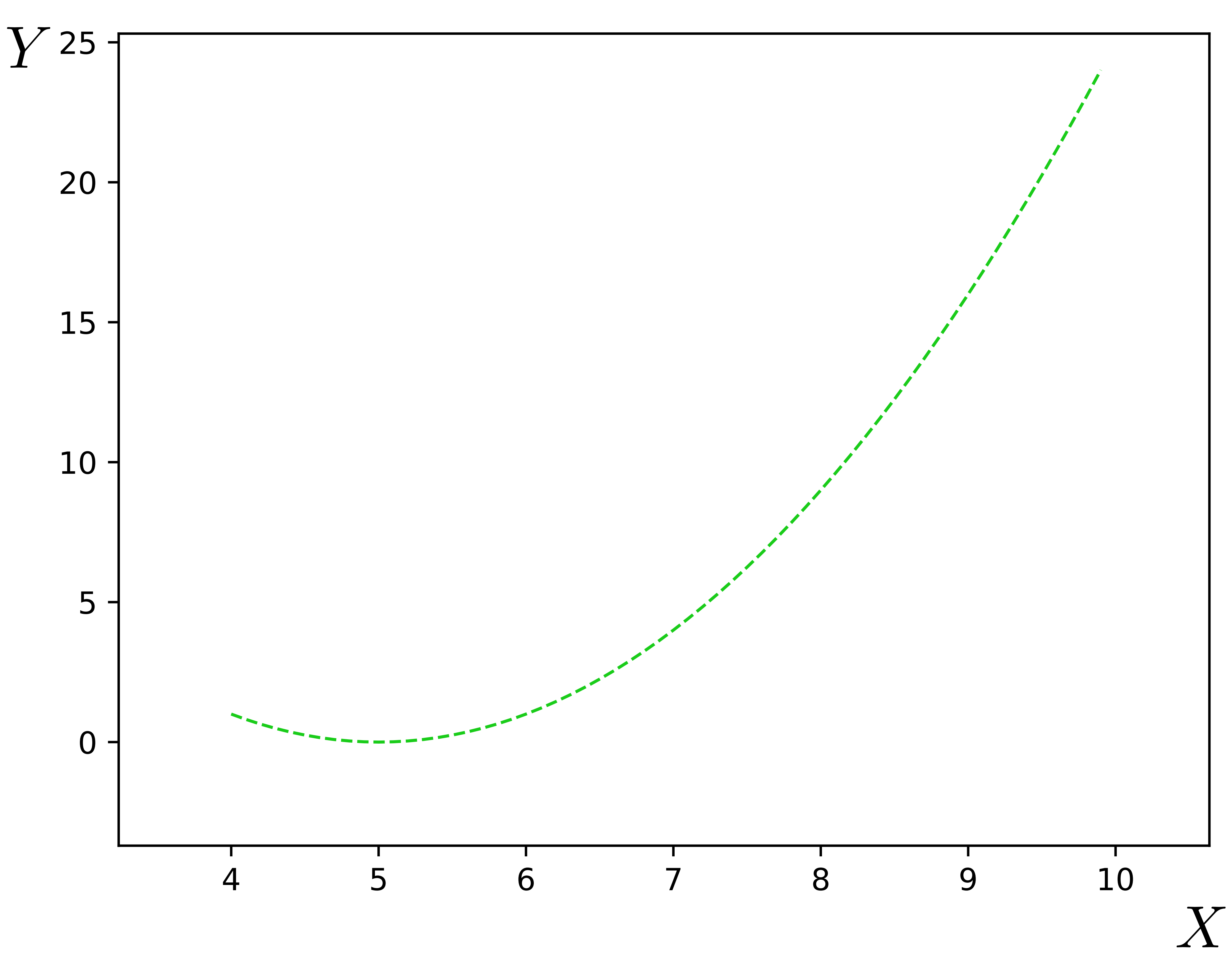


Source: Isola, Torralba, Freeman

Training data



Test data

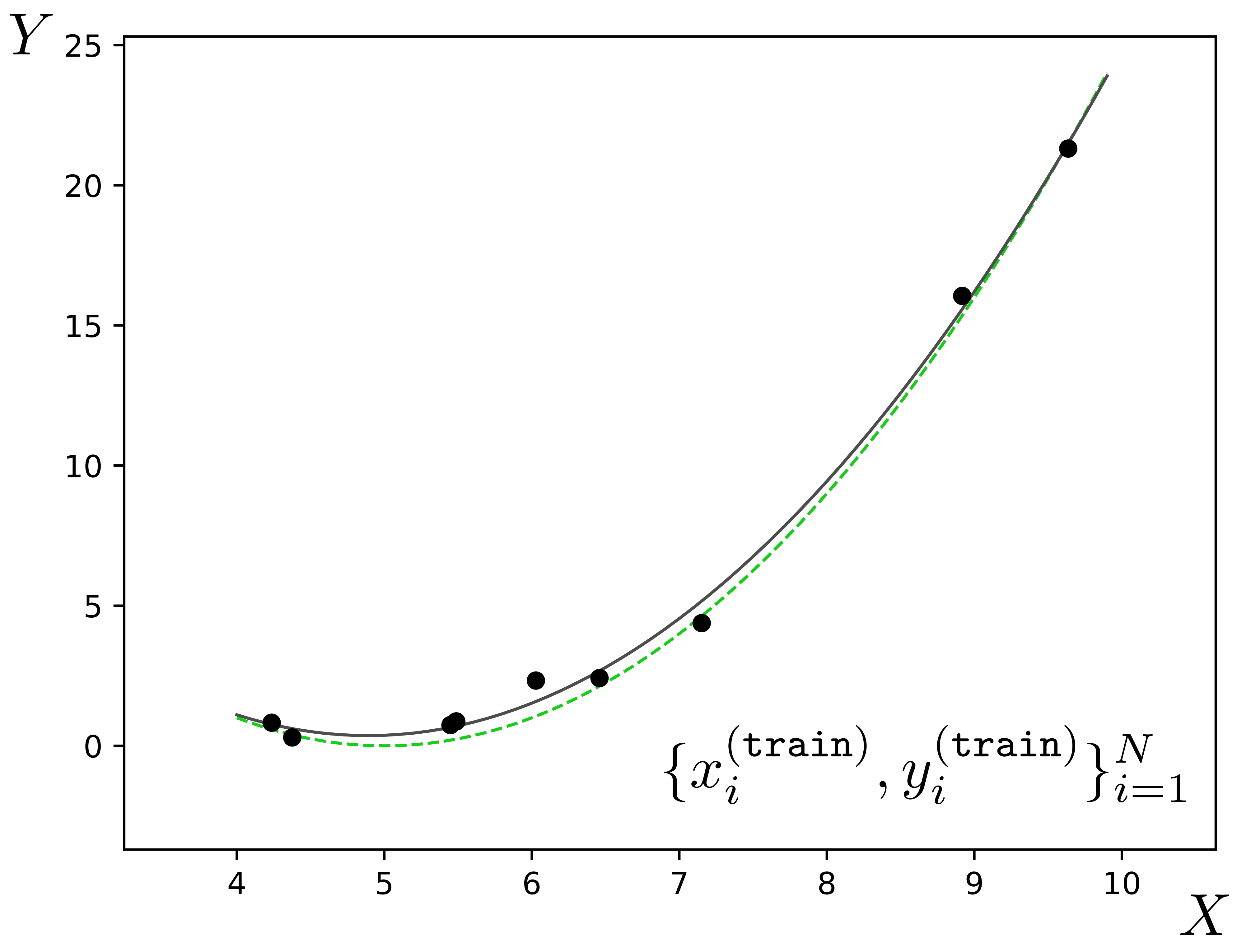


True **data-generating process**

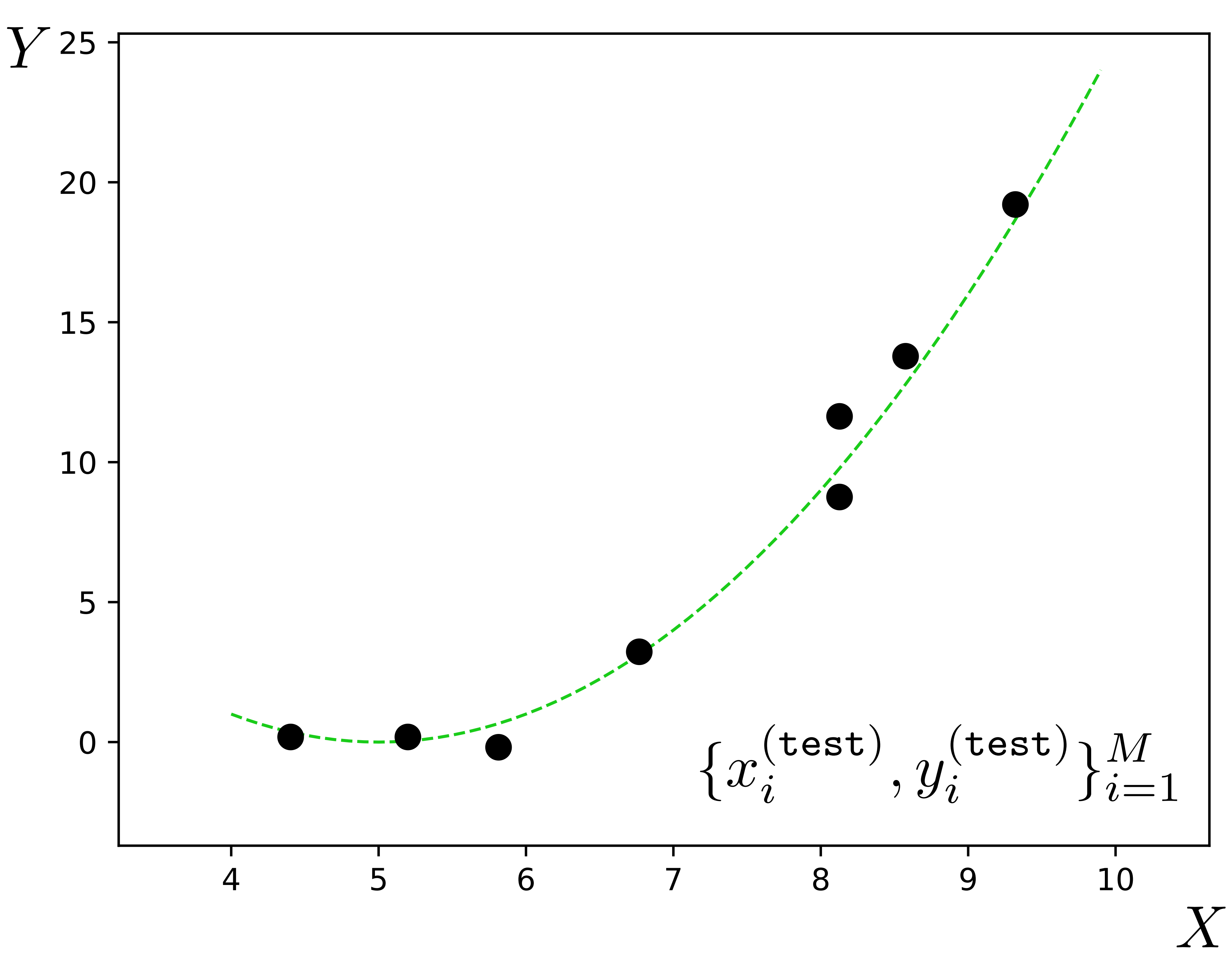
$p_{\texttt{data}}$

Source: Isola, Torralba, Freeman

Training data



Test data



True **data-generating process**

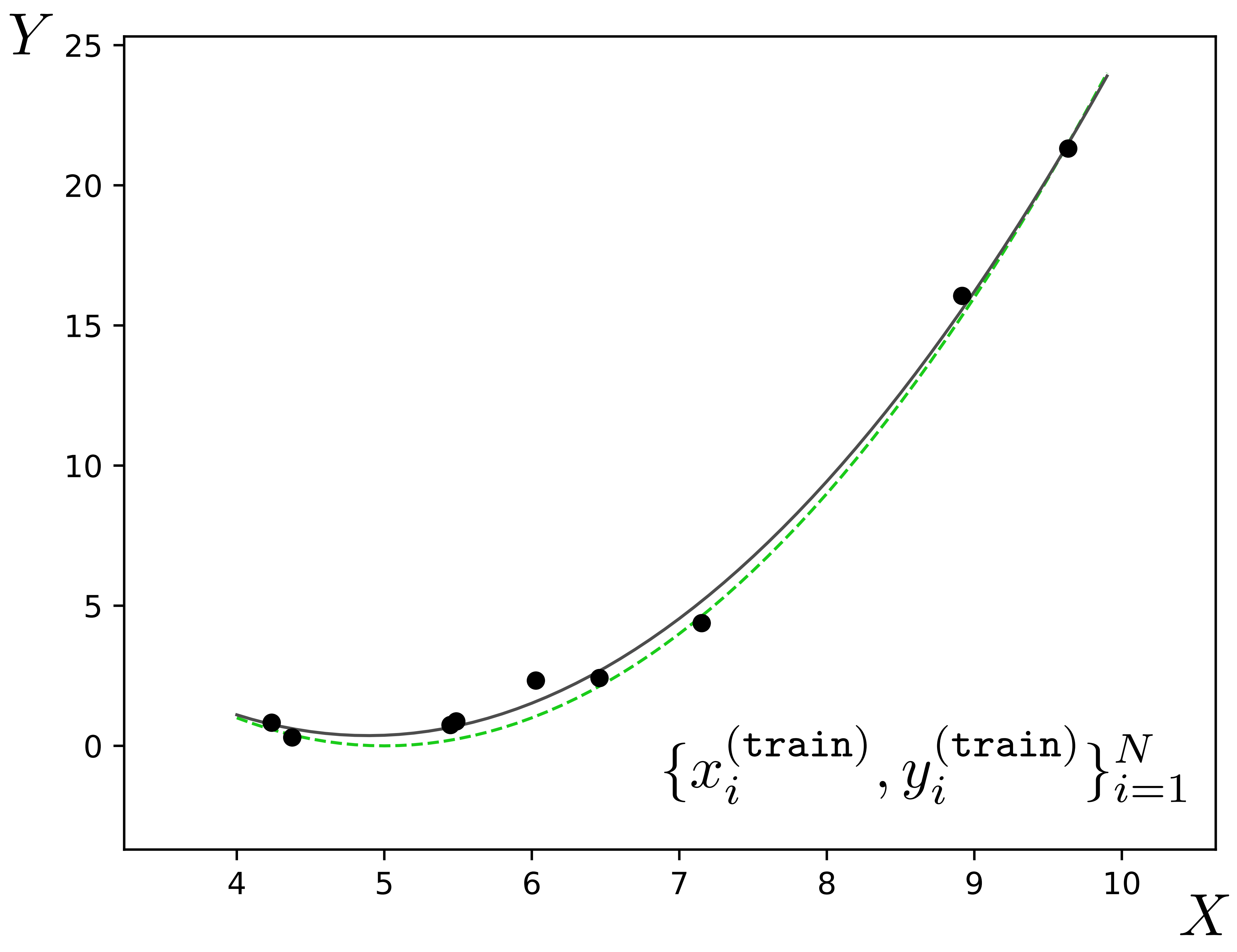
$$p_{\mathtt{data}}$$

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

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

Source: Isola, Torralba, Freeman

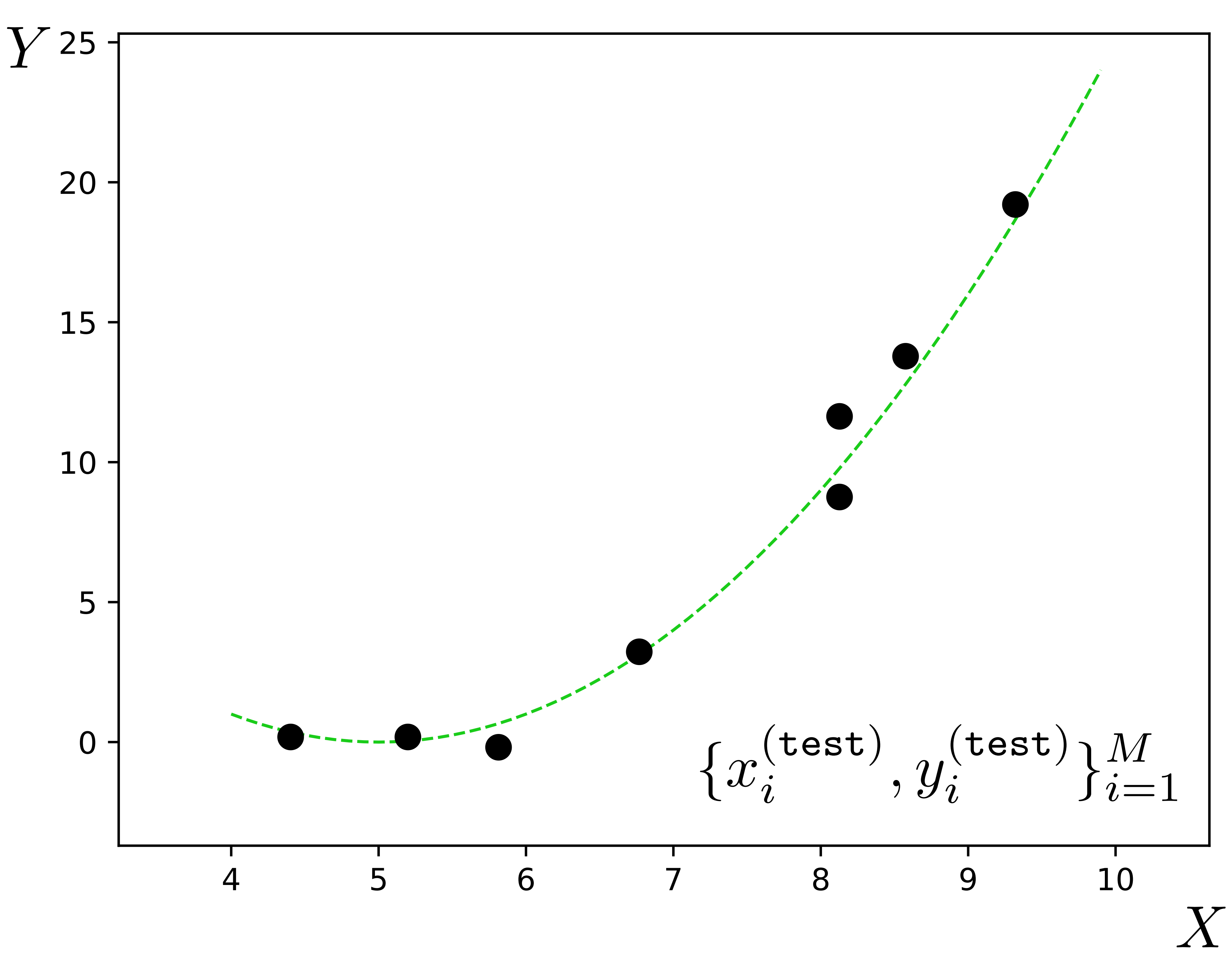
Training data



Training objective:

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

Test data

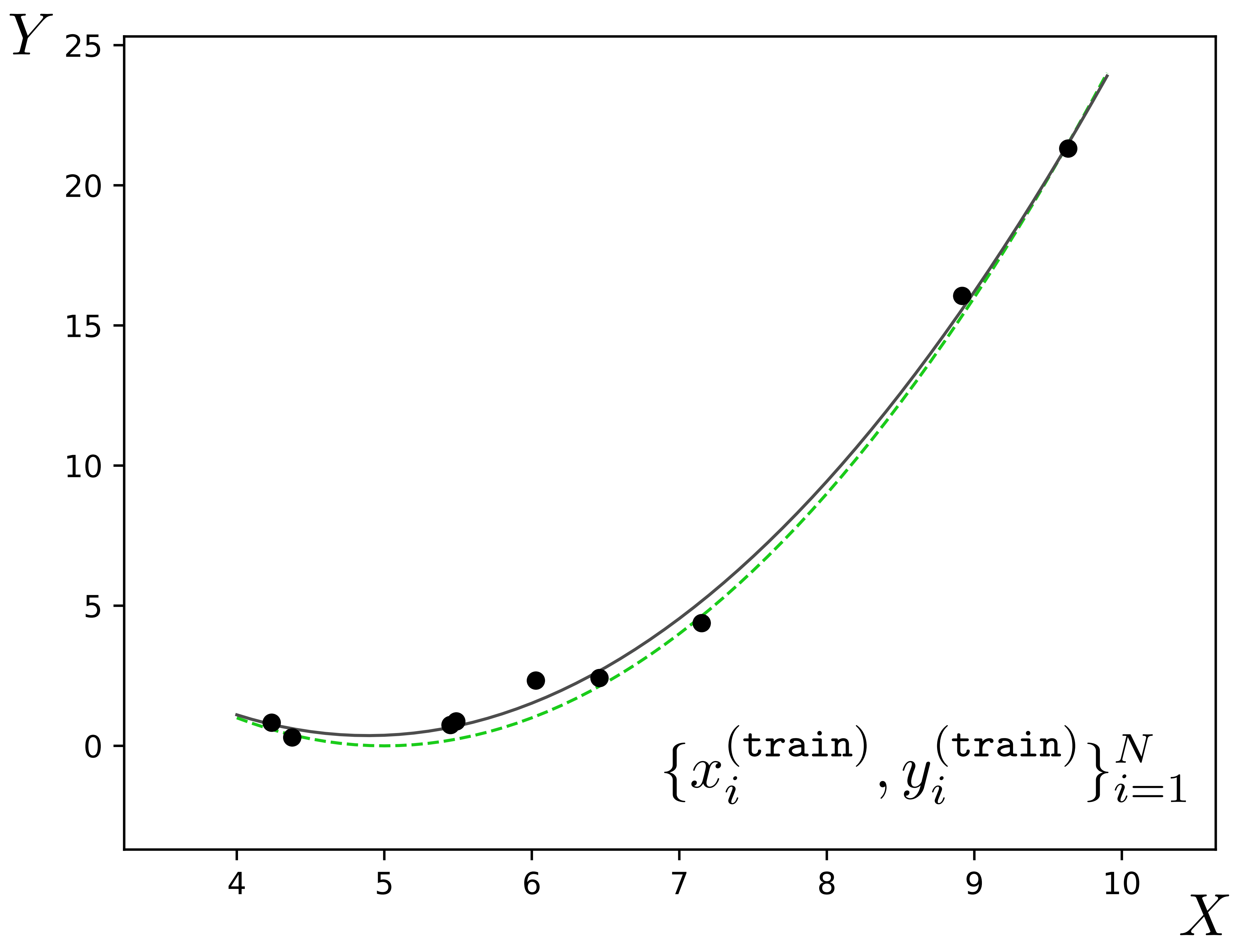


Test time evaluation:

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

Source: Isola, Torralba, Freeman

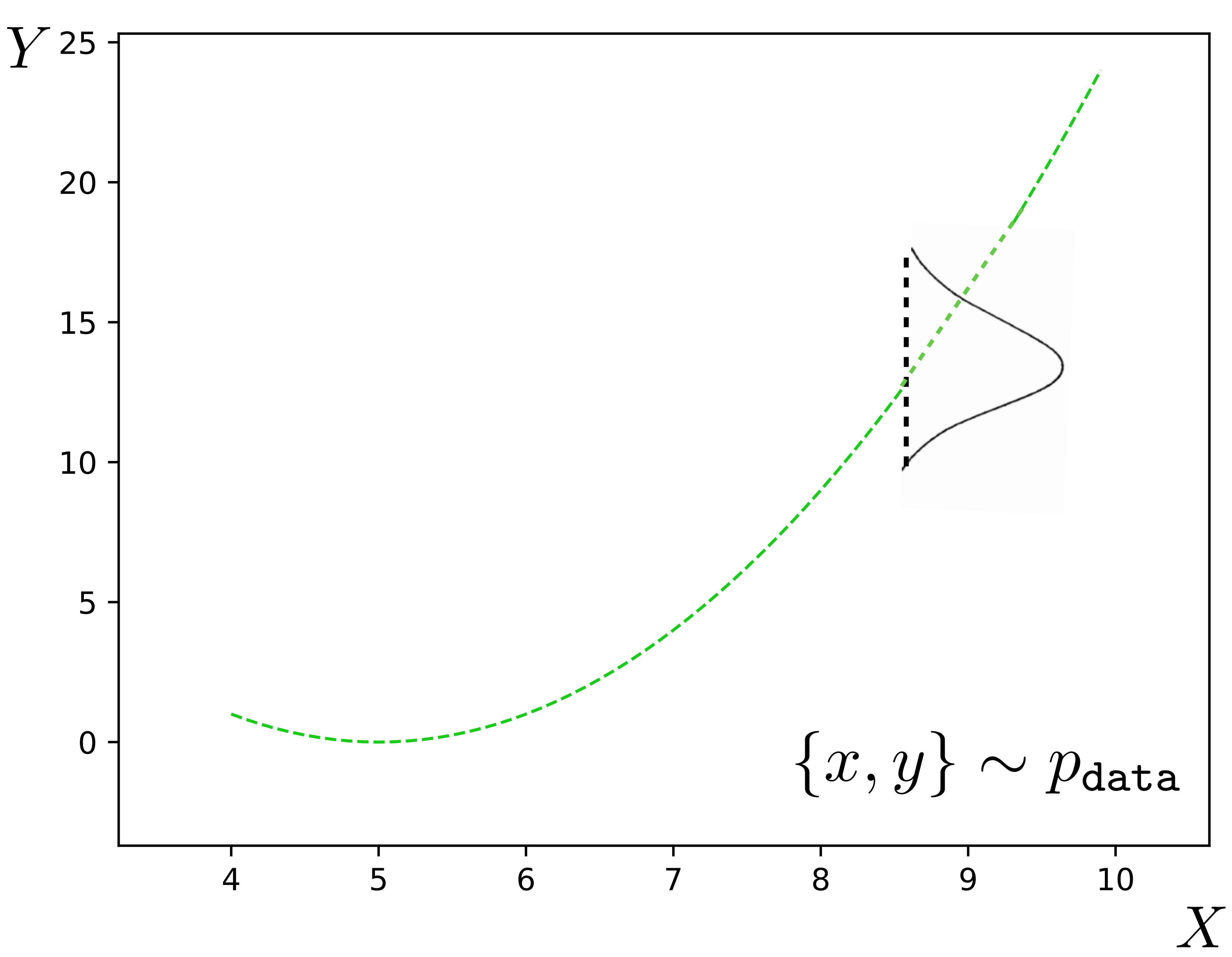
Training data



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

Generalization

"The central challenge in machine learning is that our algorithm must perform well on new, previously unseen inputs—not just those on which our model was trained. The ability to perform well on previously unobserved inputs is called **generalization**.

… [this is what] separates machine learning from optimization."

— Deep Learning textbook (Goodfellow et al.)

What does ☆ do?

2 ☆ 3 = 36

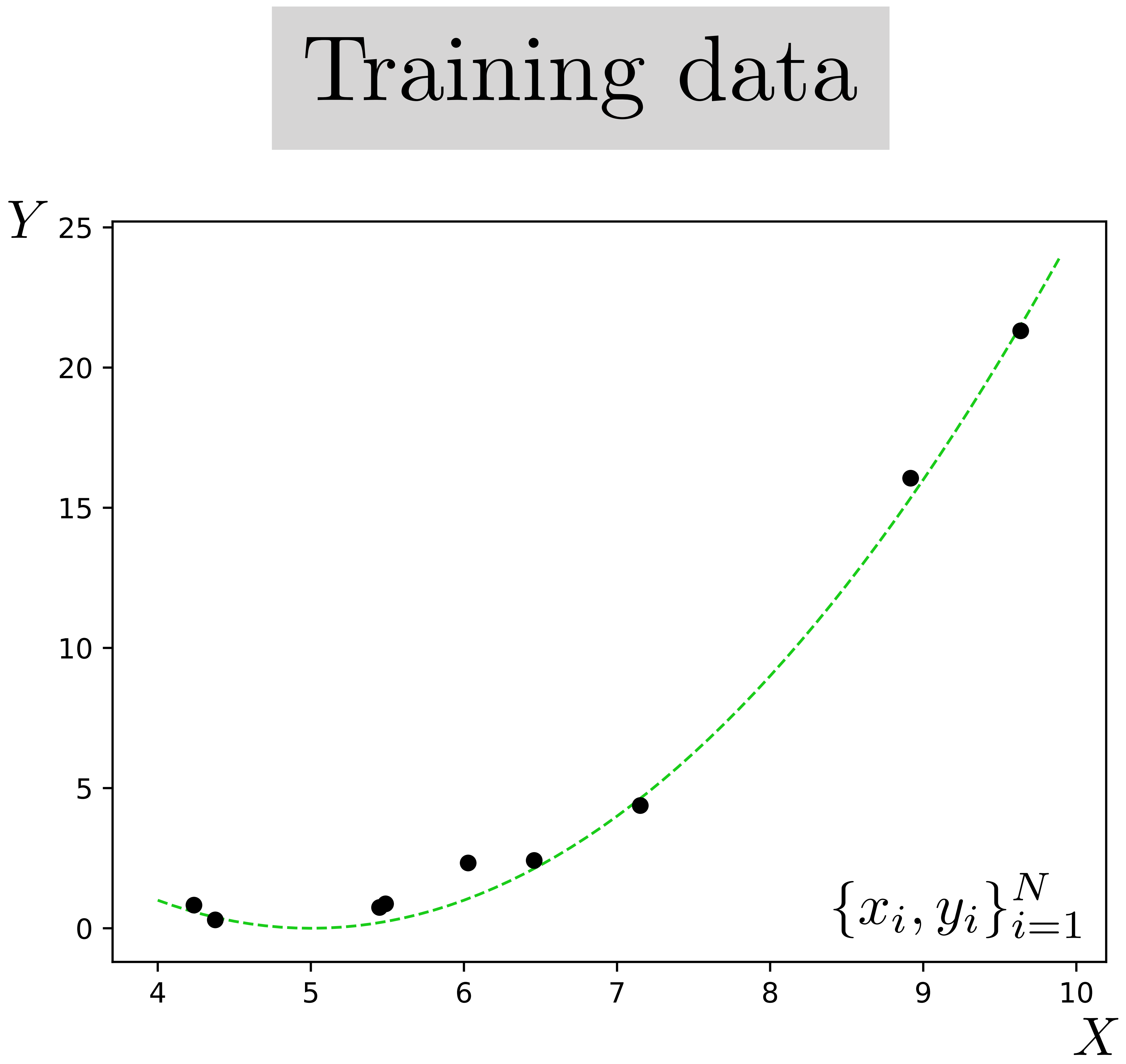
7 ☆ 1 = 49

5 ☆ 2 = 100

2 ☆ 2 = 16

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

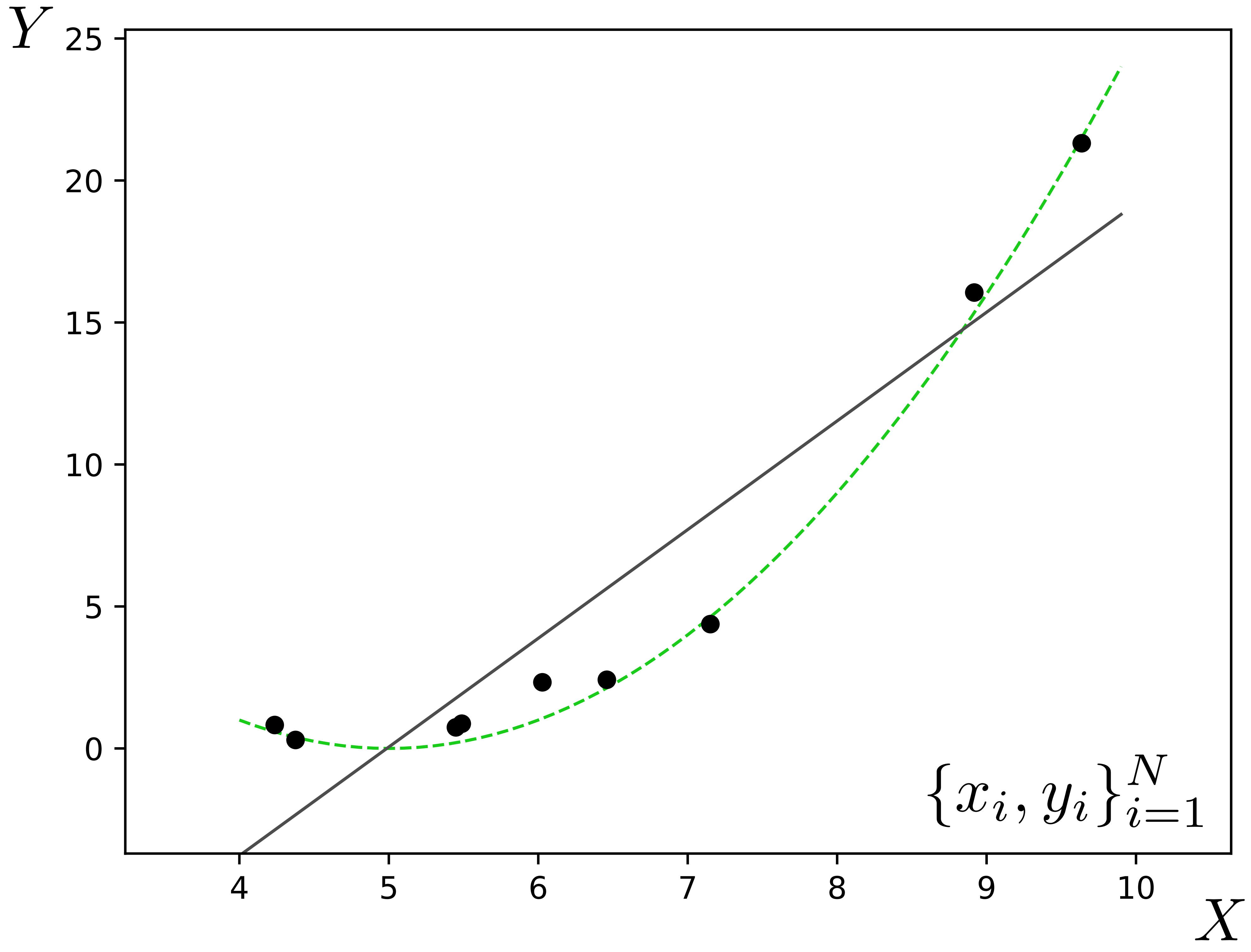


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 1

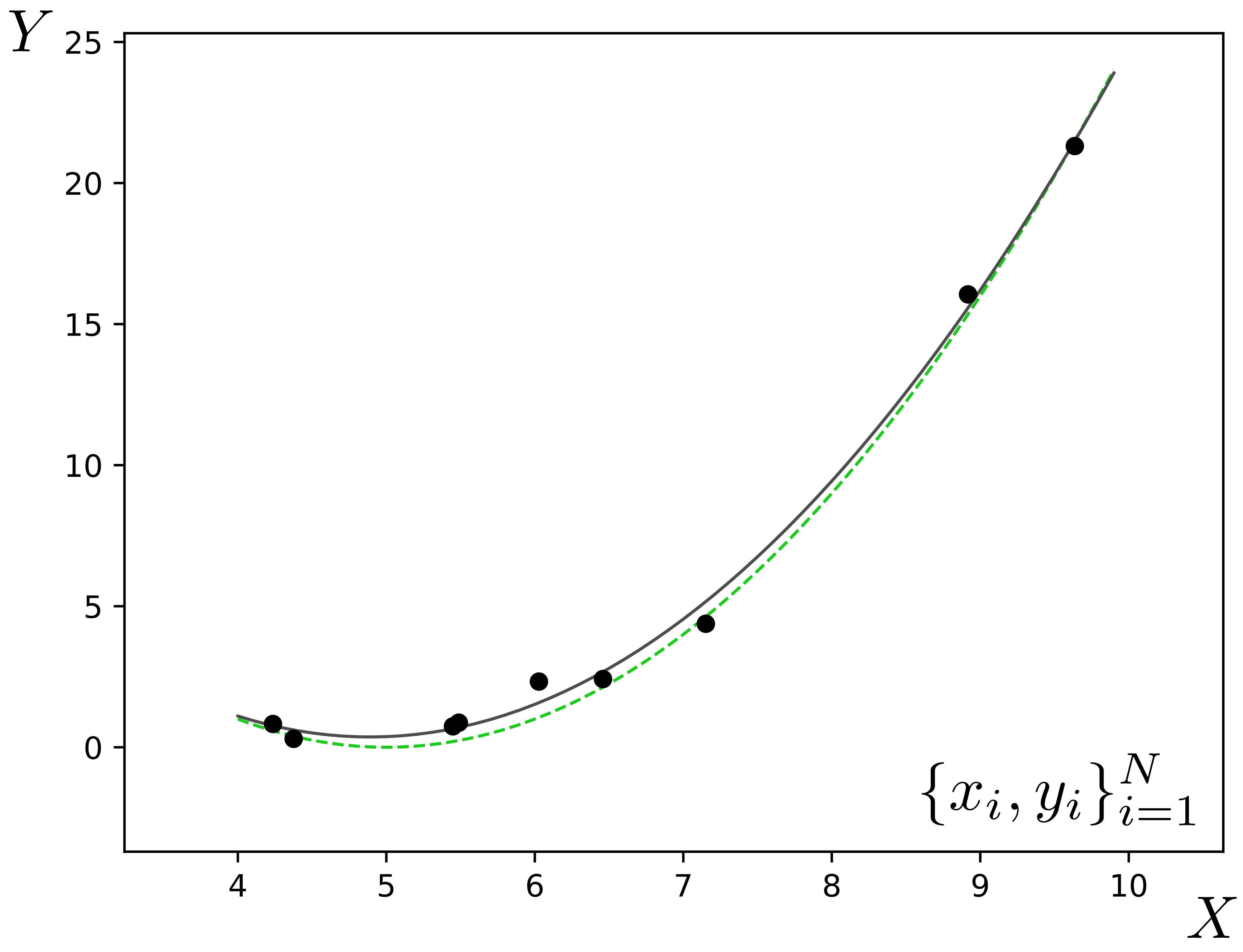


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 2

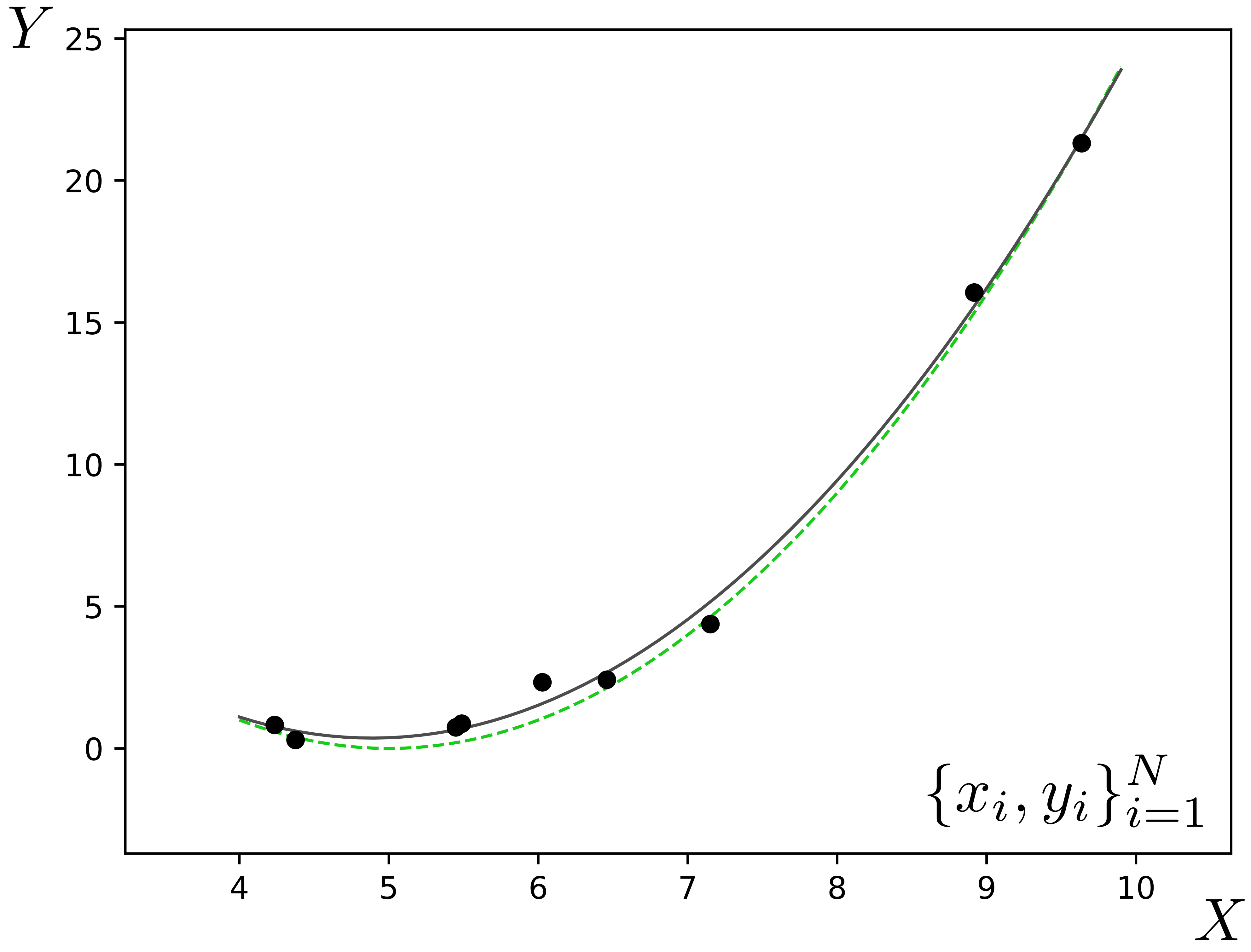


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 3

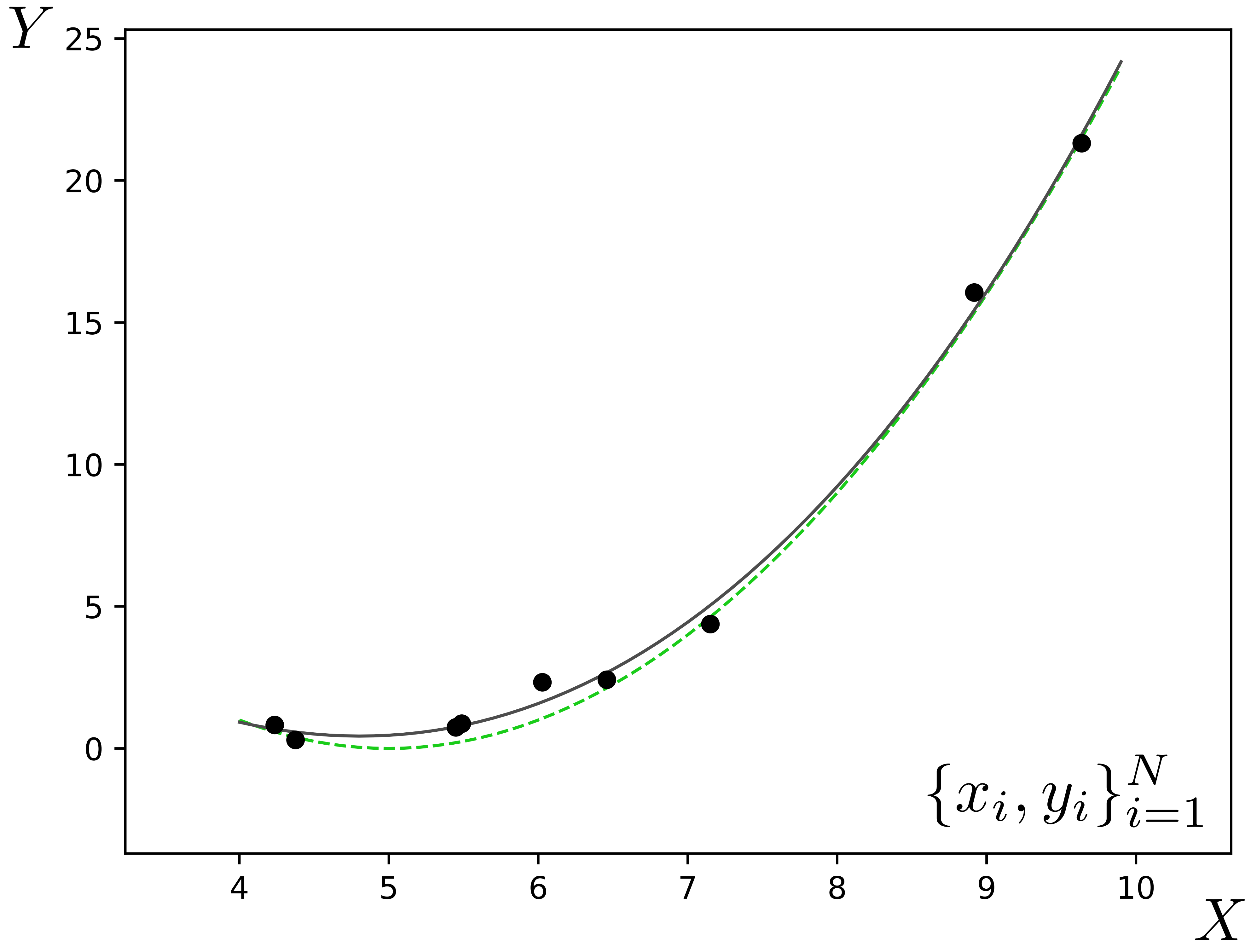


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 4

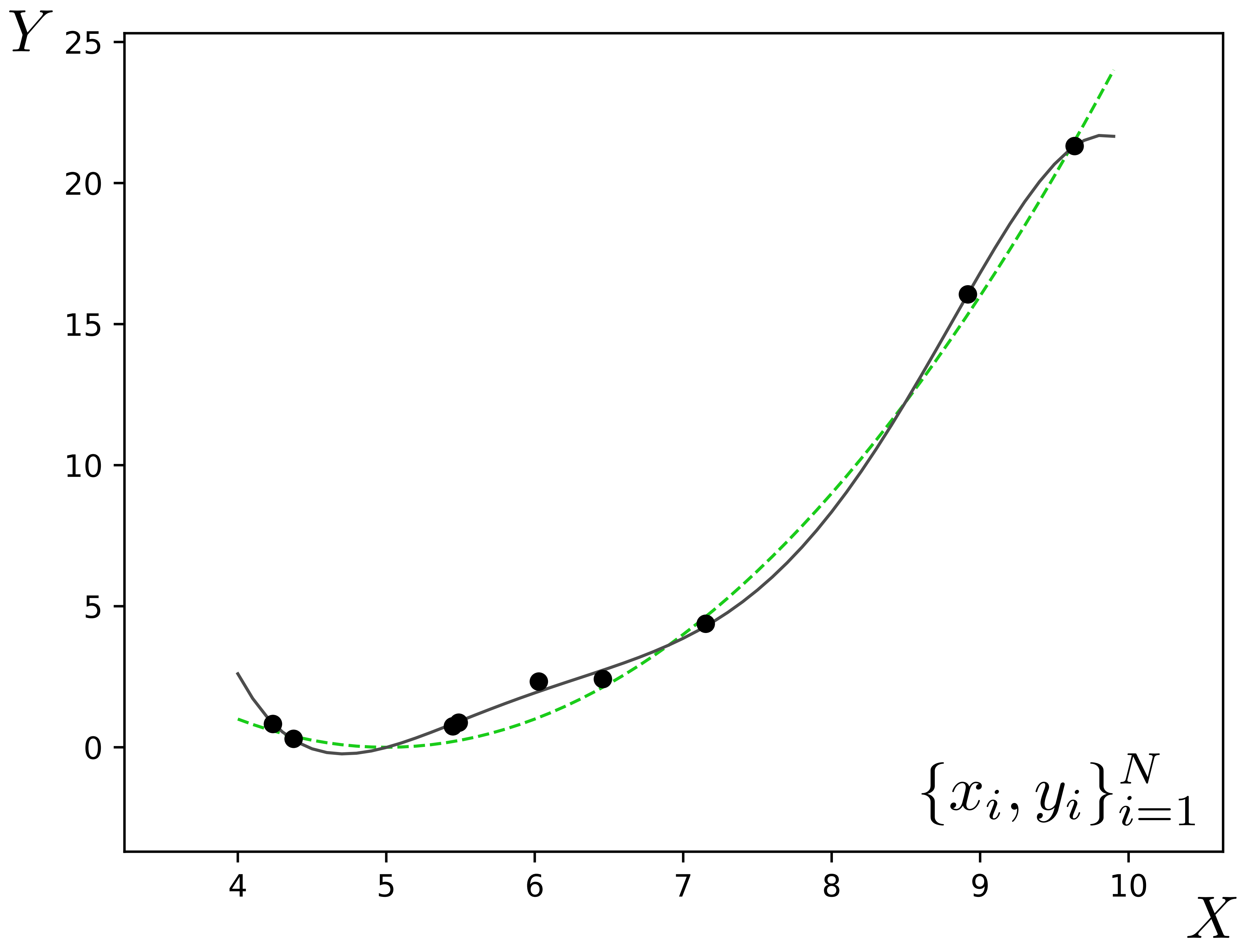


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 5

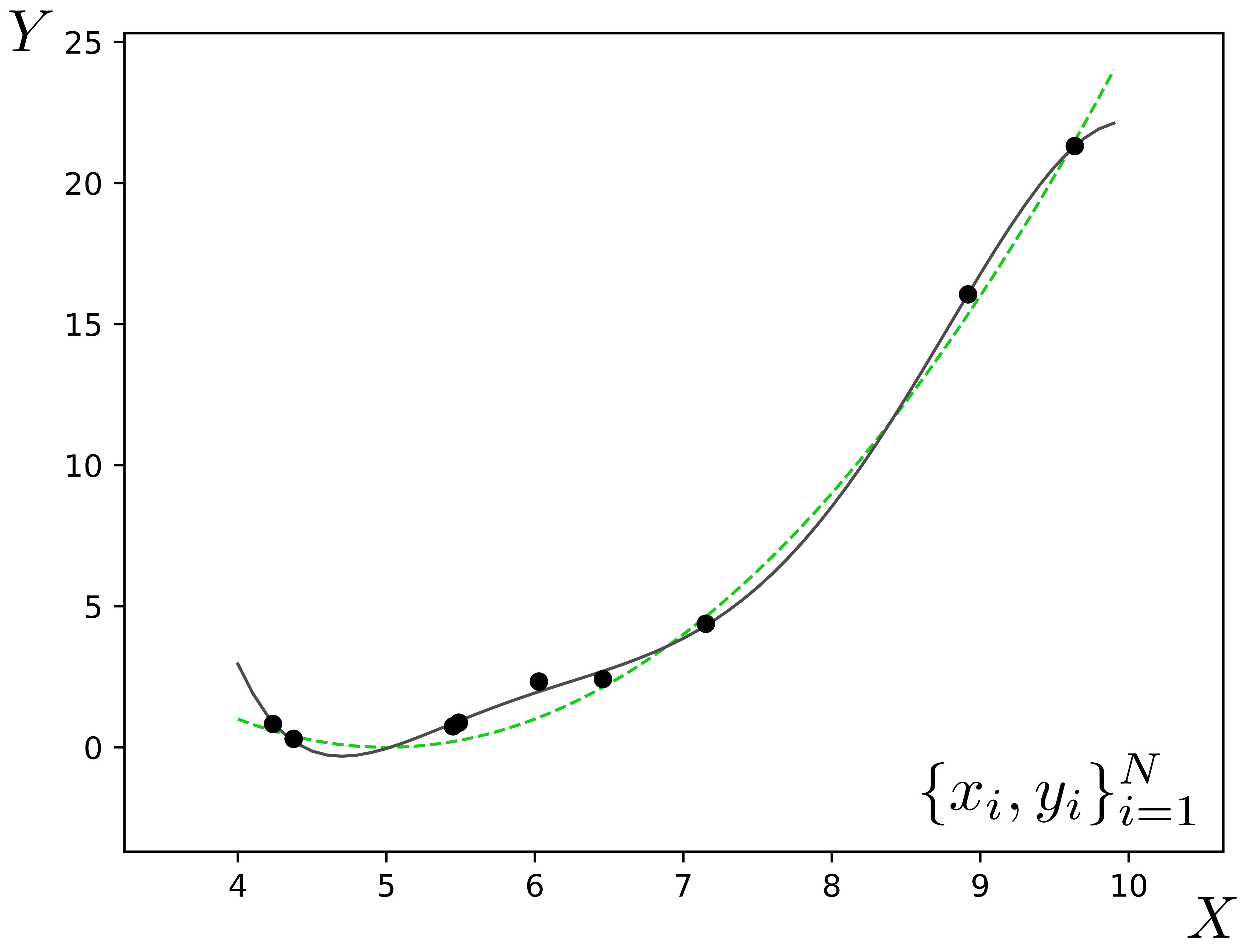


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 6

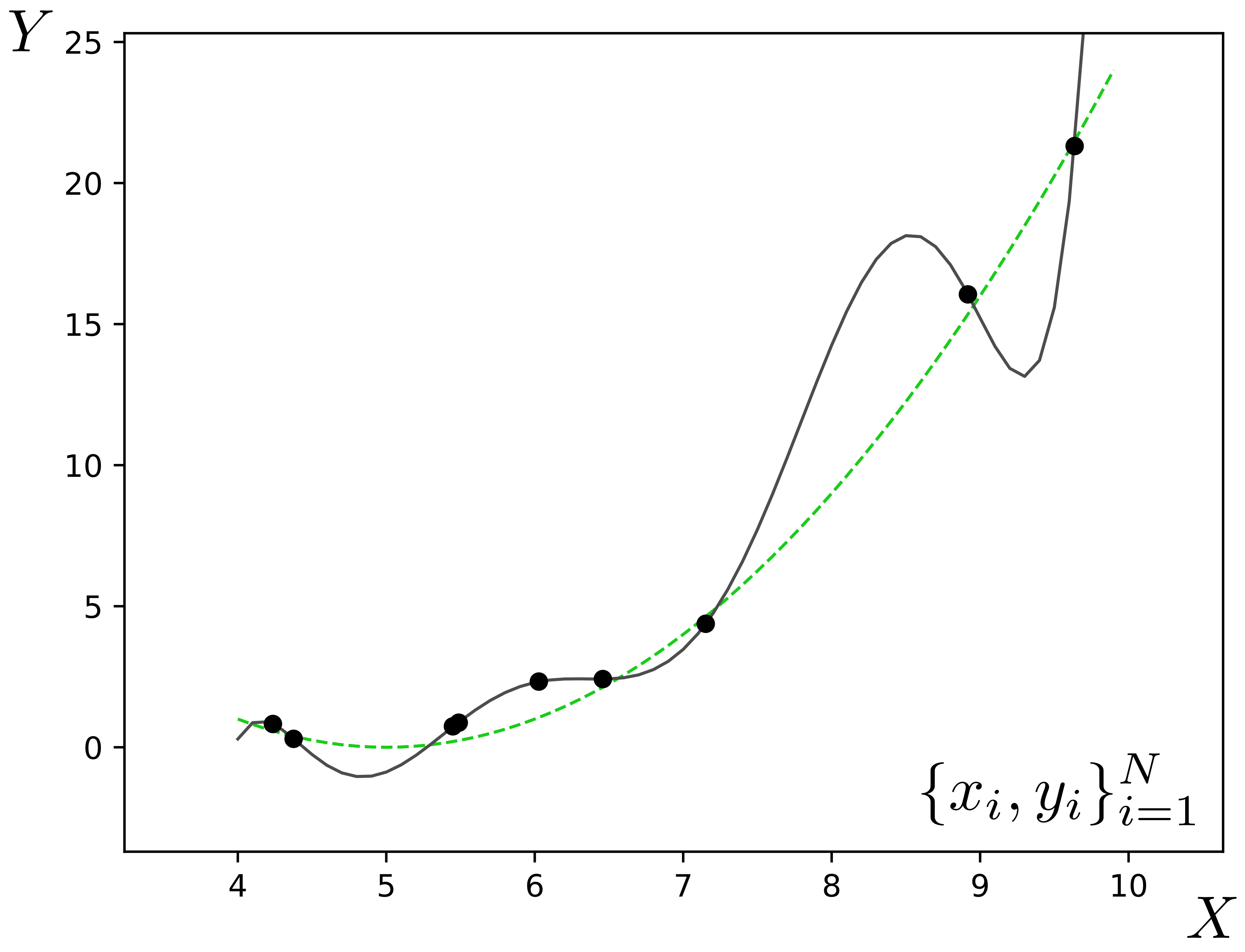


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 7

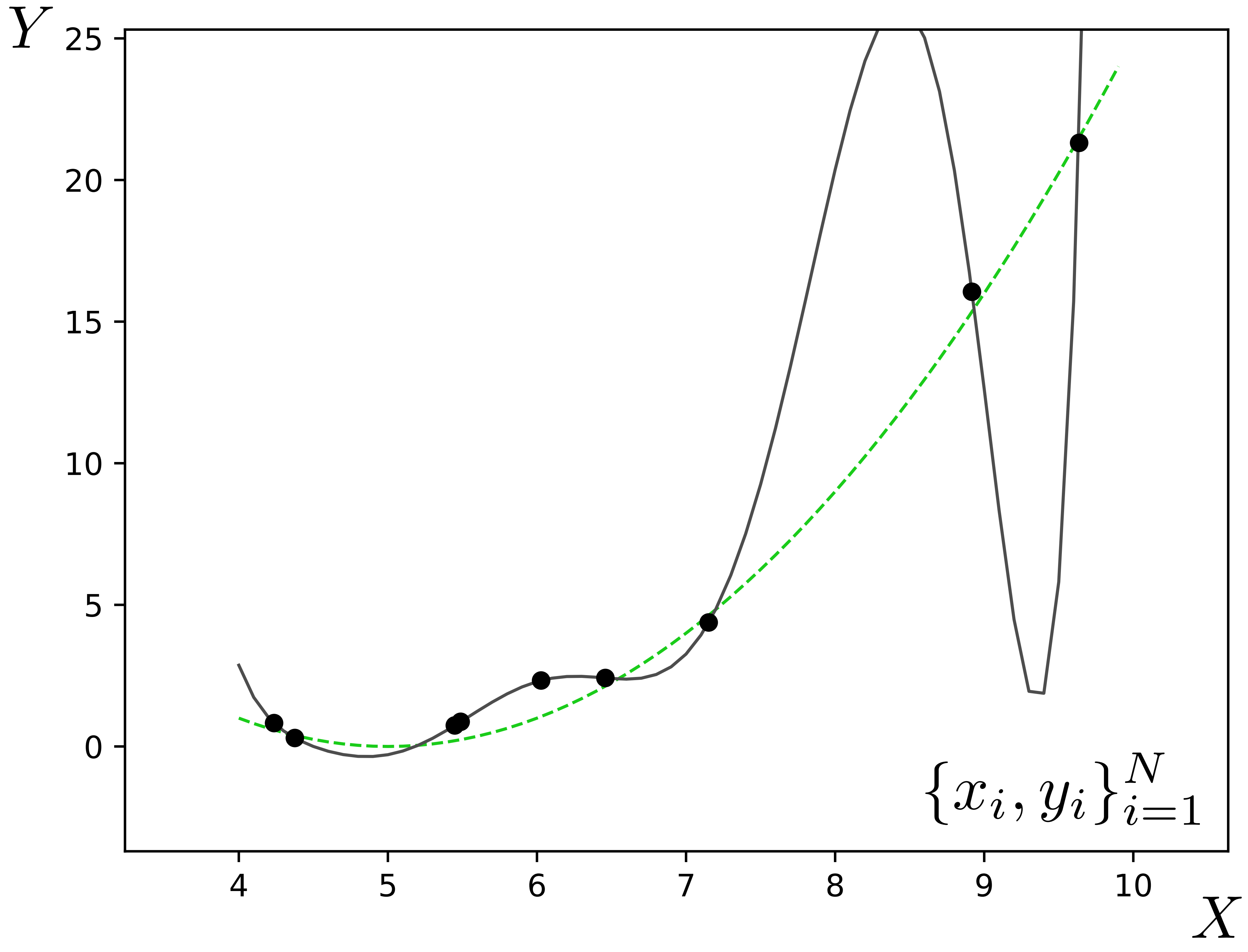


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

$\boxed{\mathbf{K = 8}}$

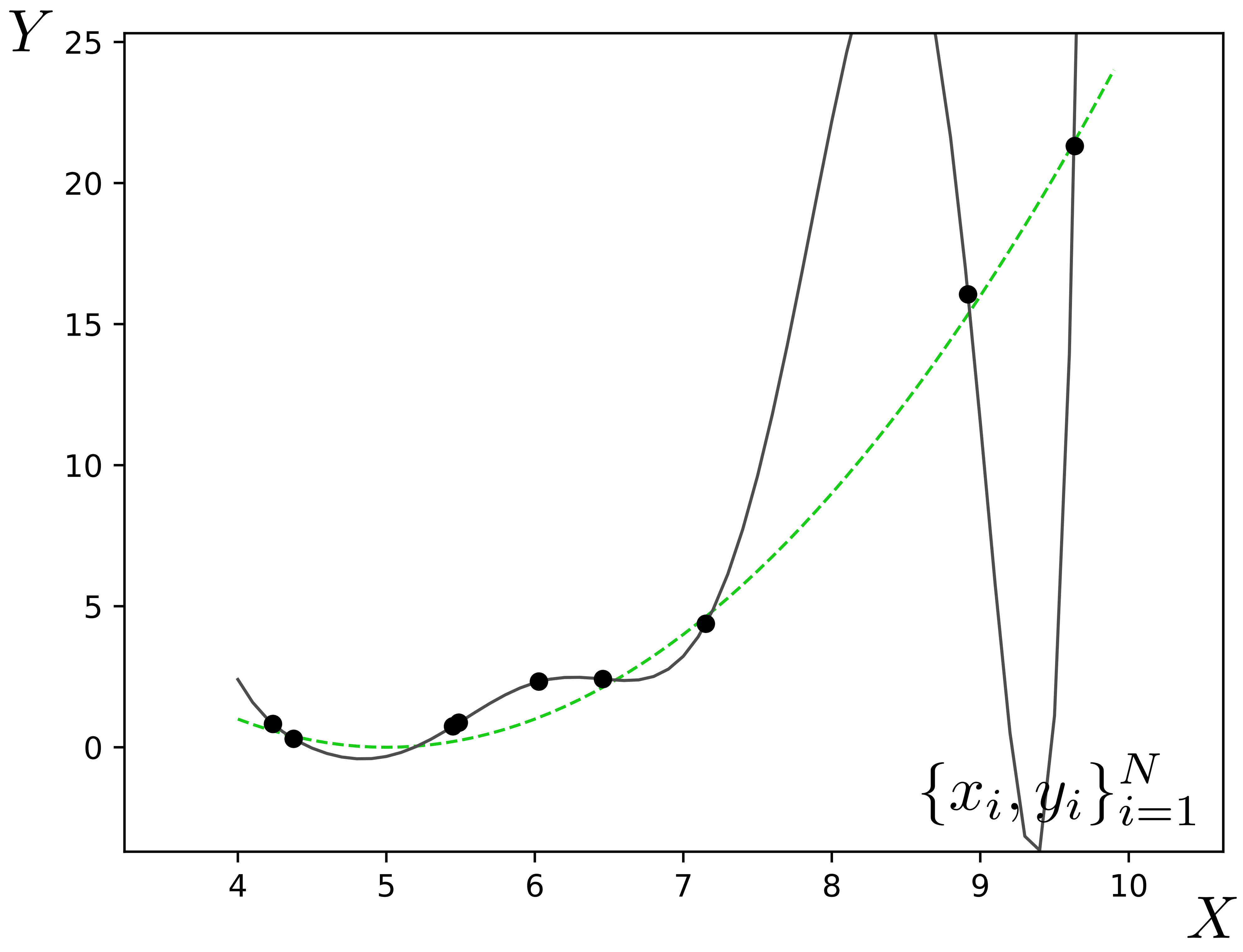


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

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

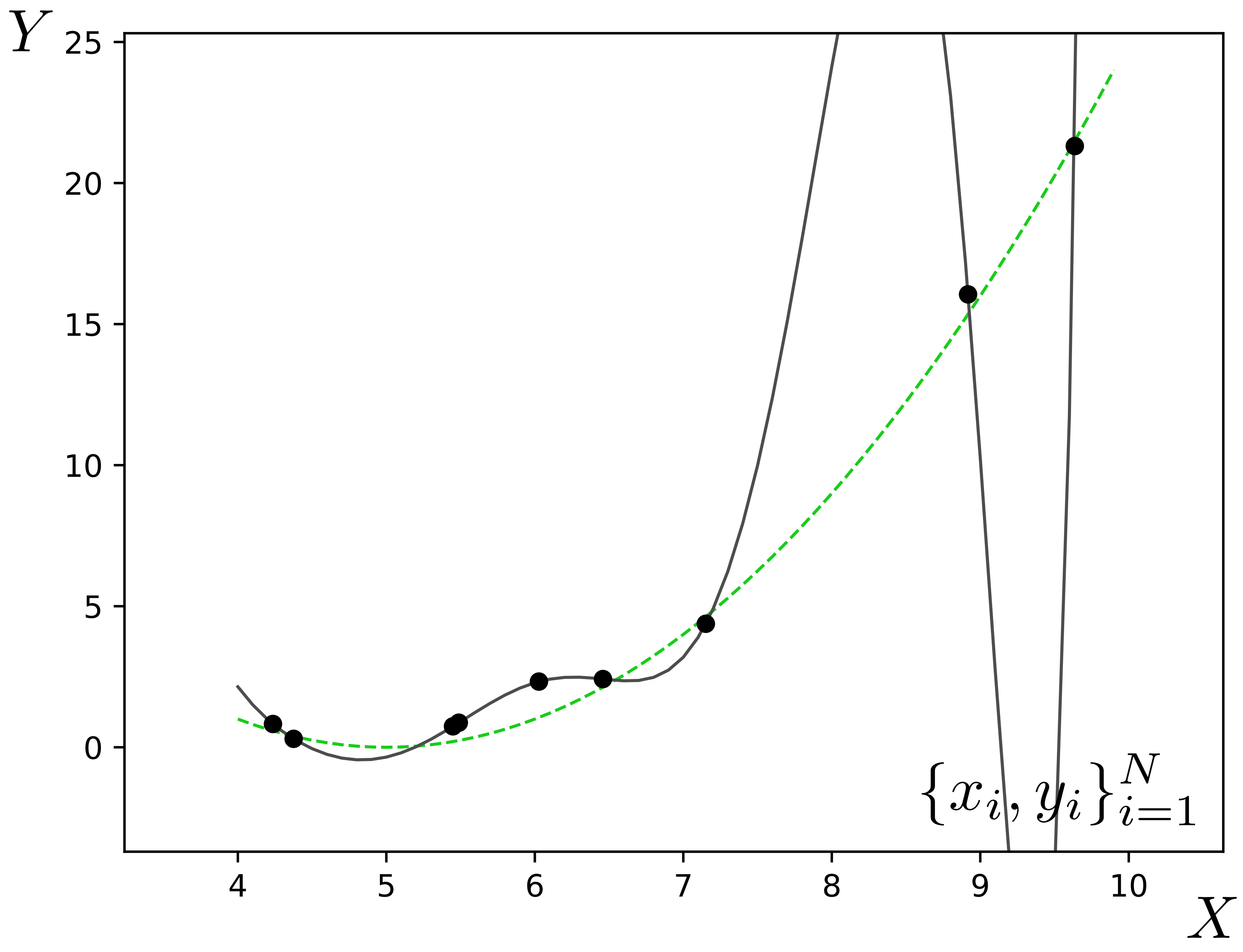


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman

What happens as we add more basis functions?

K = 10

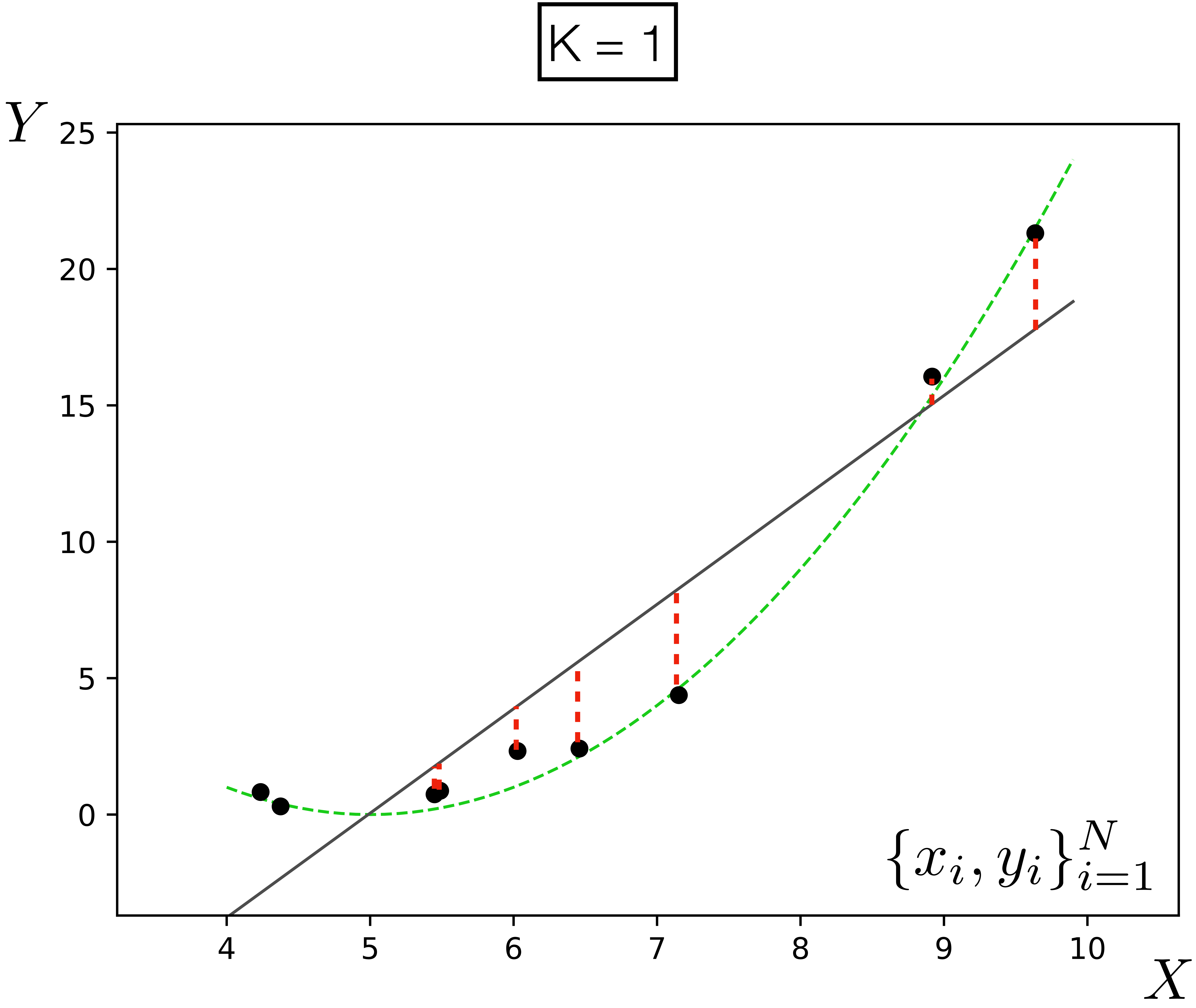


$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

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

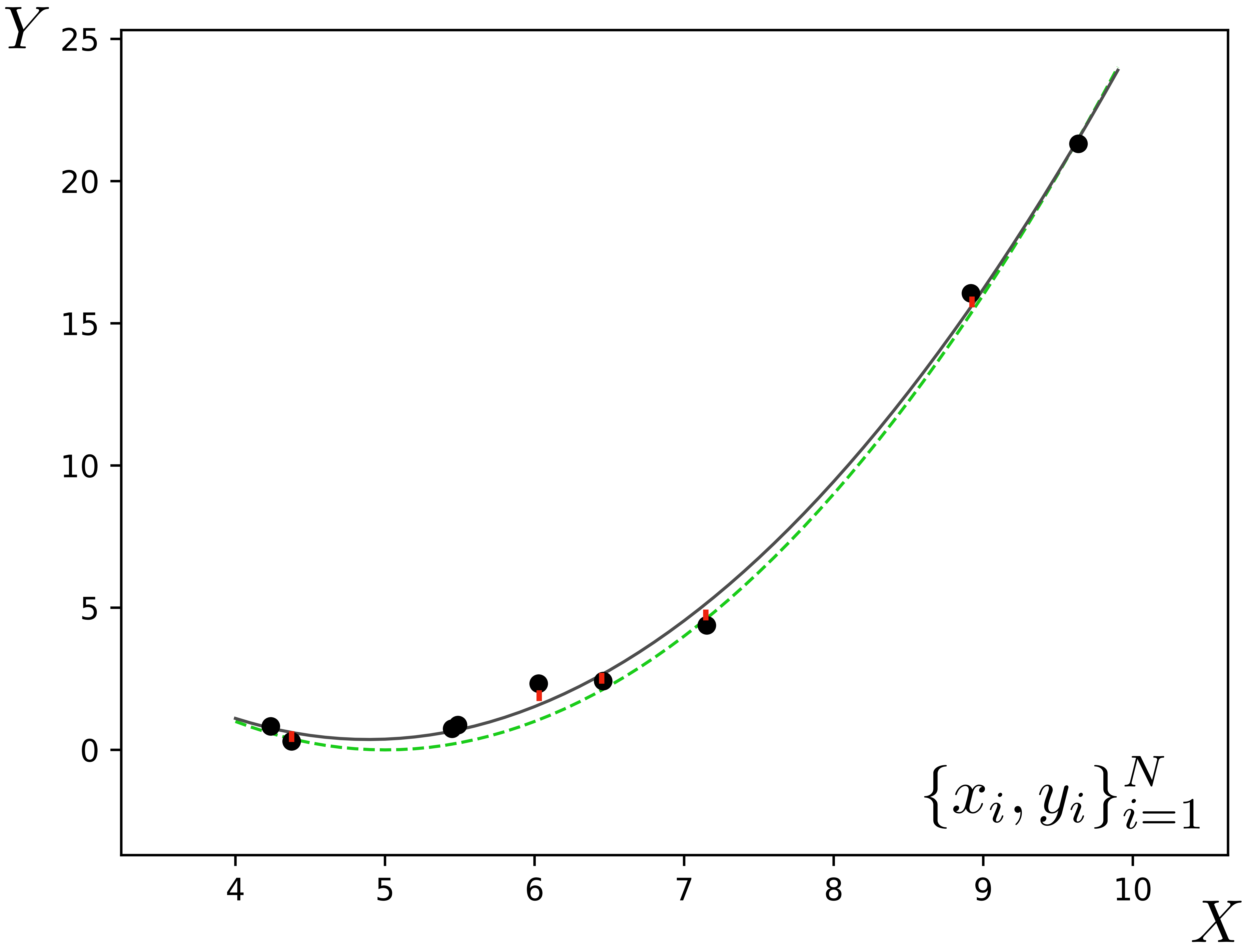


When the model does not have the capacity to capture the true function, we call this **underfitting**.

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

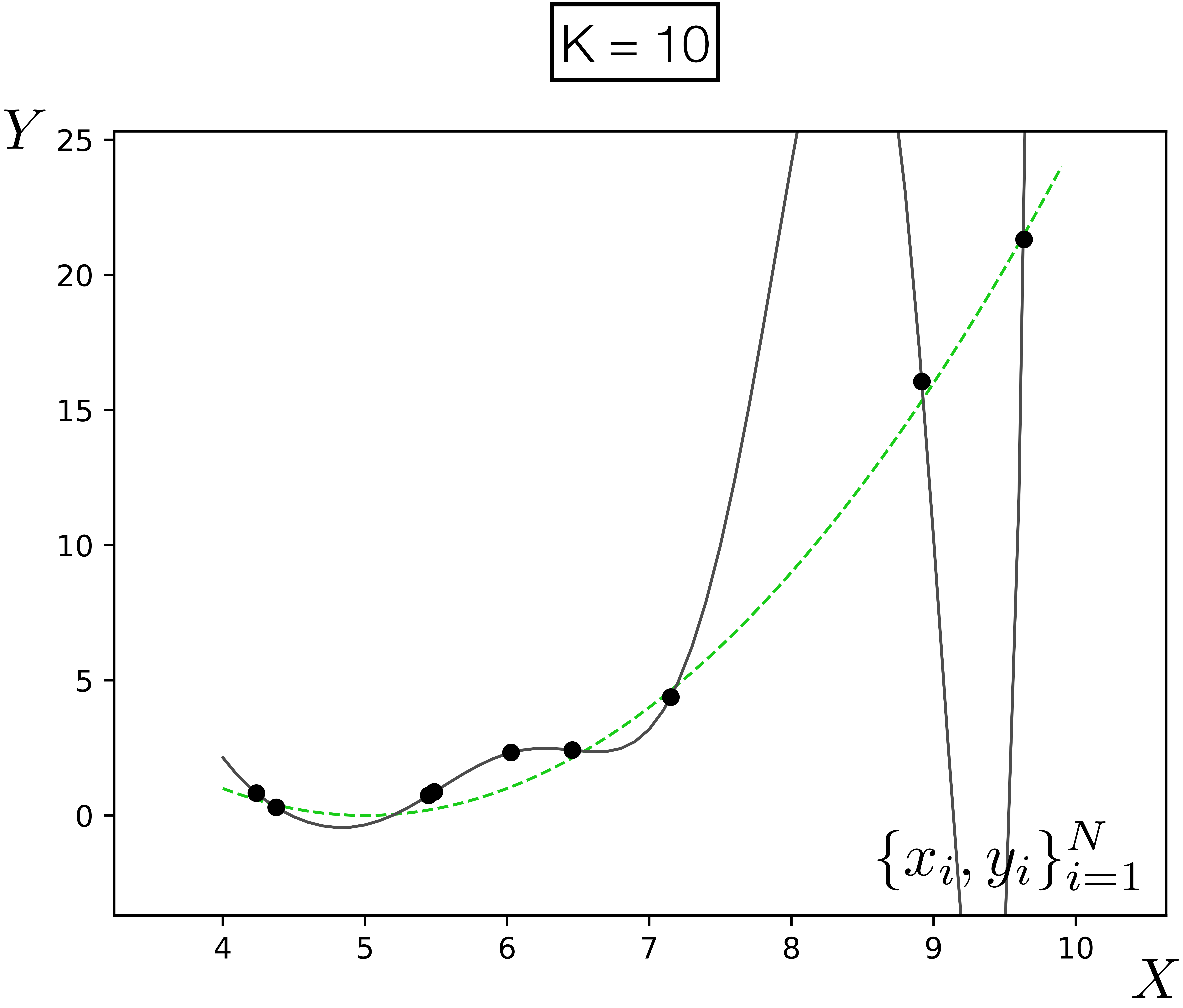
Source: Isola, Torralba, Freeman

$\boxed{K = 2}$



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

Source: Isola, Torralba, Freeman



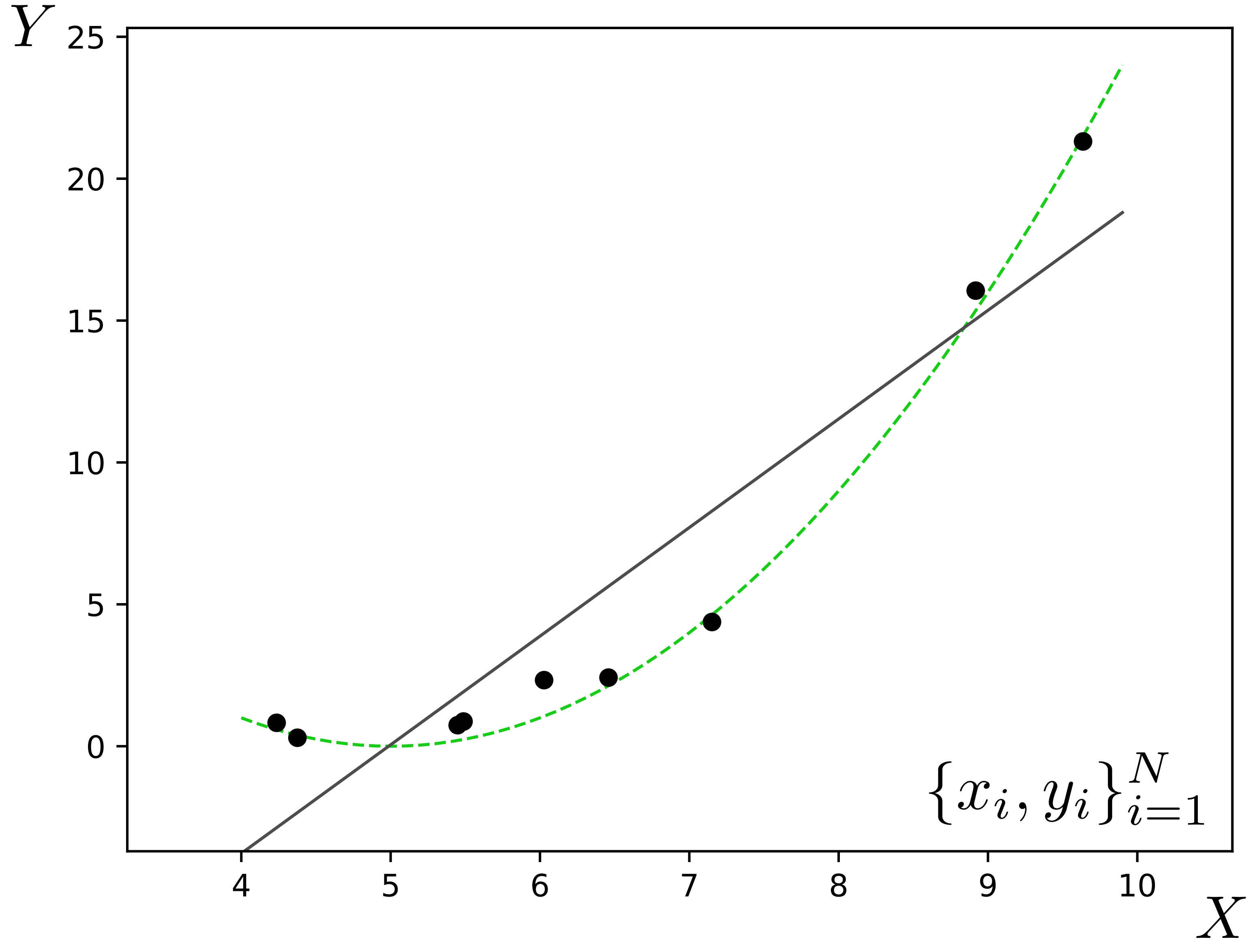
Now we have zero approximation error — the curve passes exactly through each training point.

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

$\boxed{K = 1}$

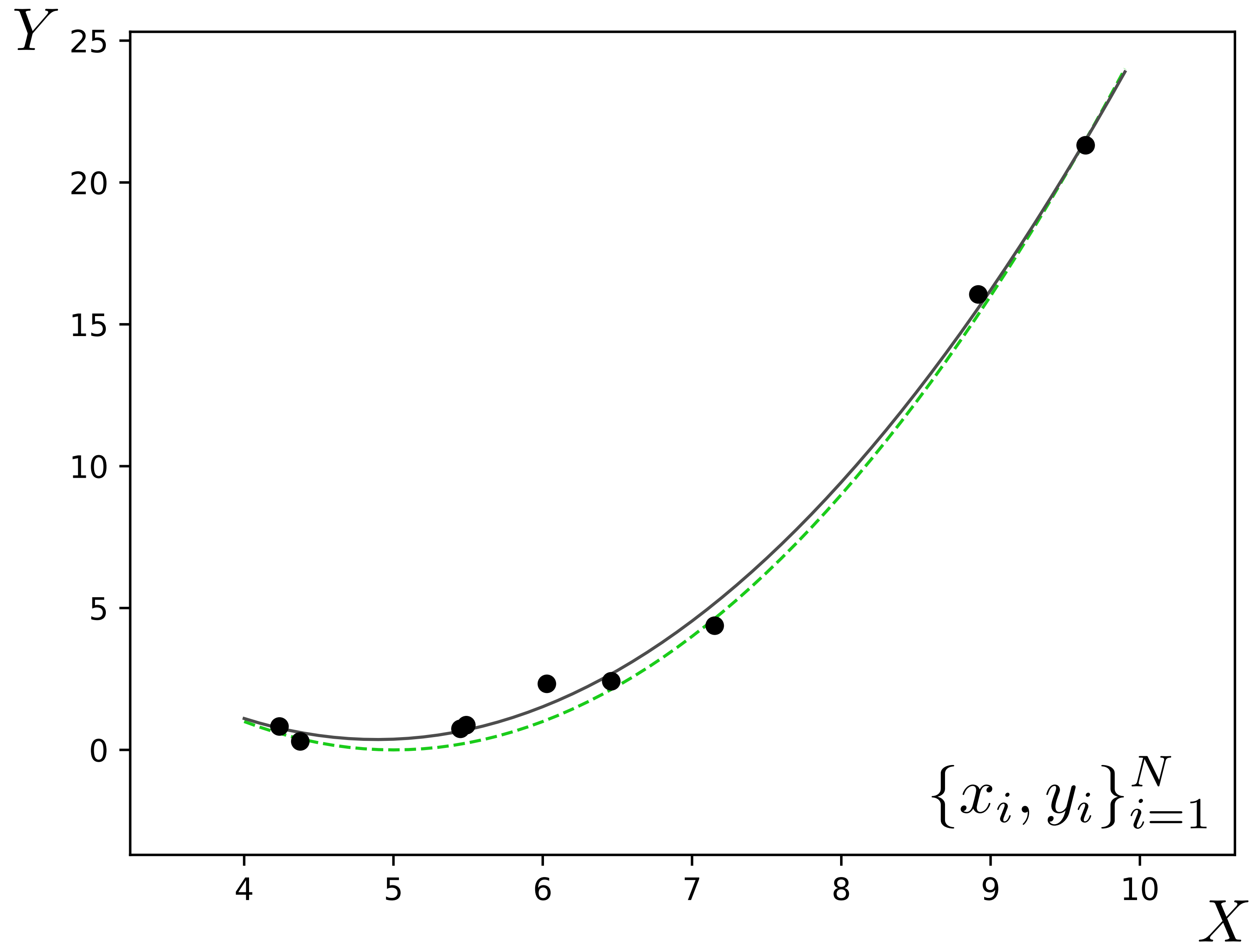


High error on train set

High error on test set

Appropriate model

$\boxed{K = 2}$

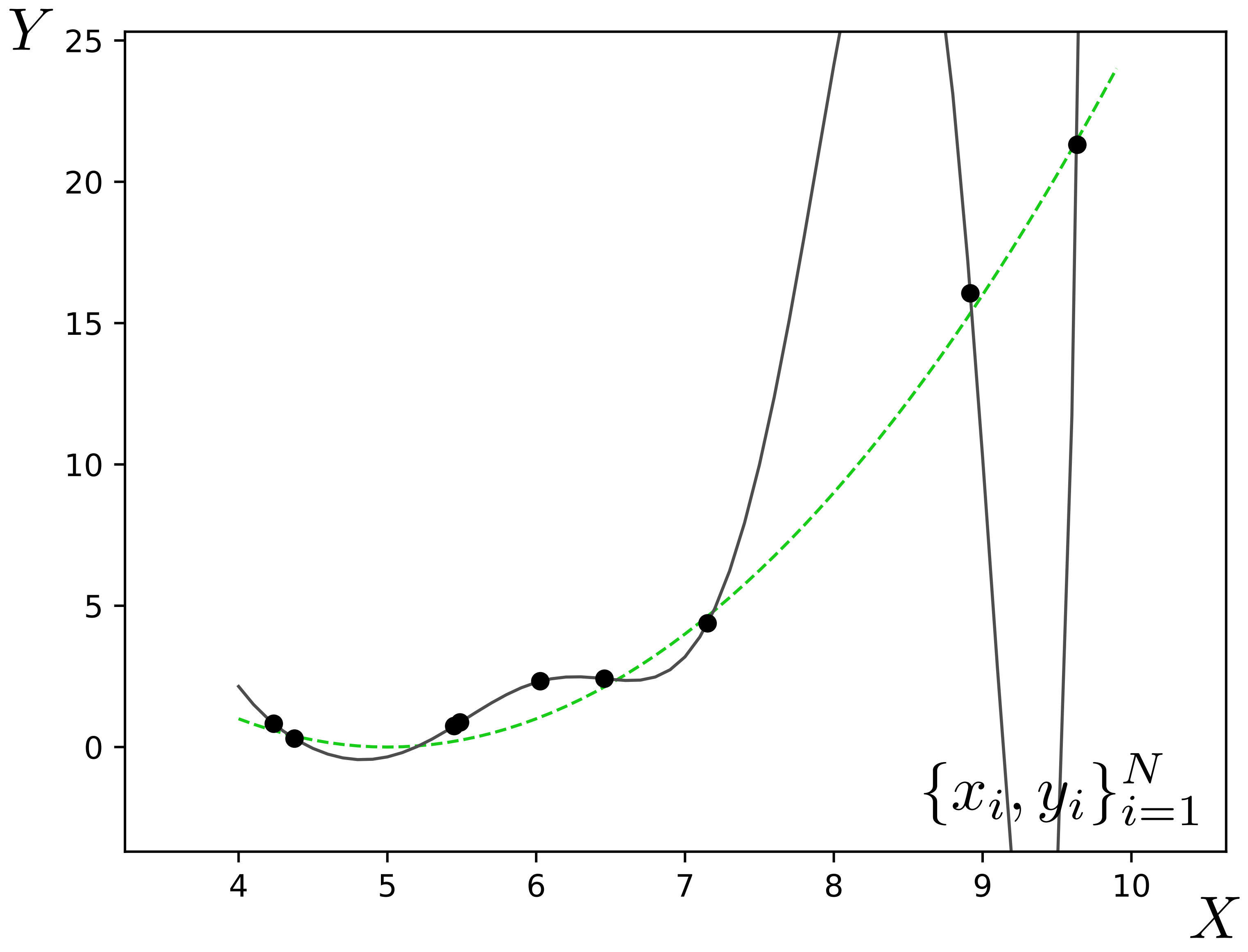


Low error on train set

Low error on test set

Overfitting

$\boxed{K = 10}$



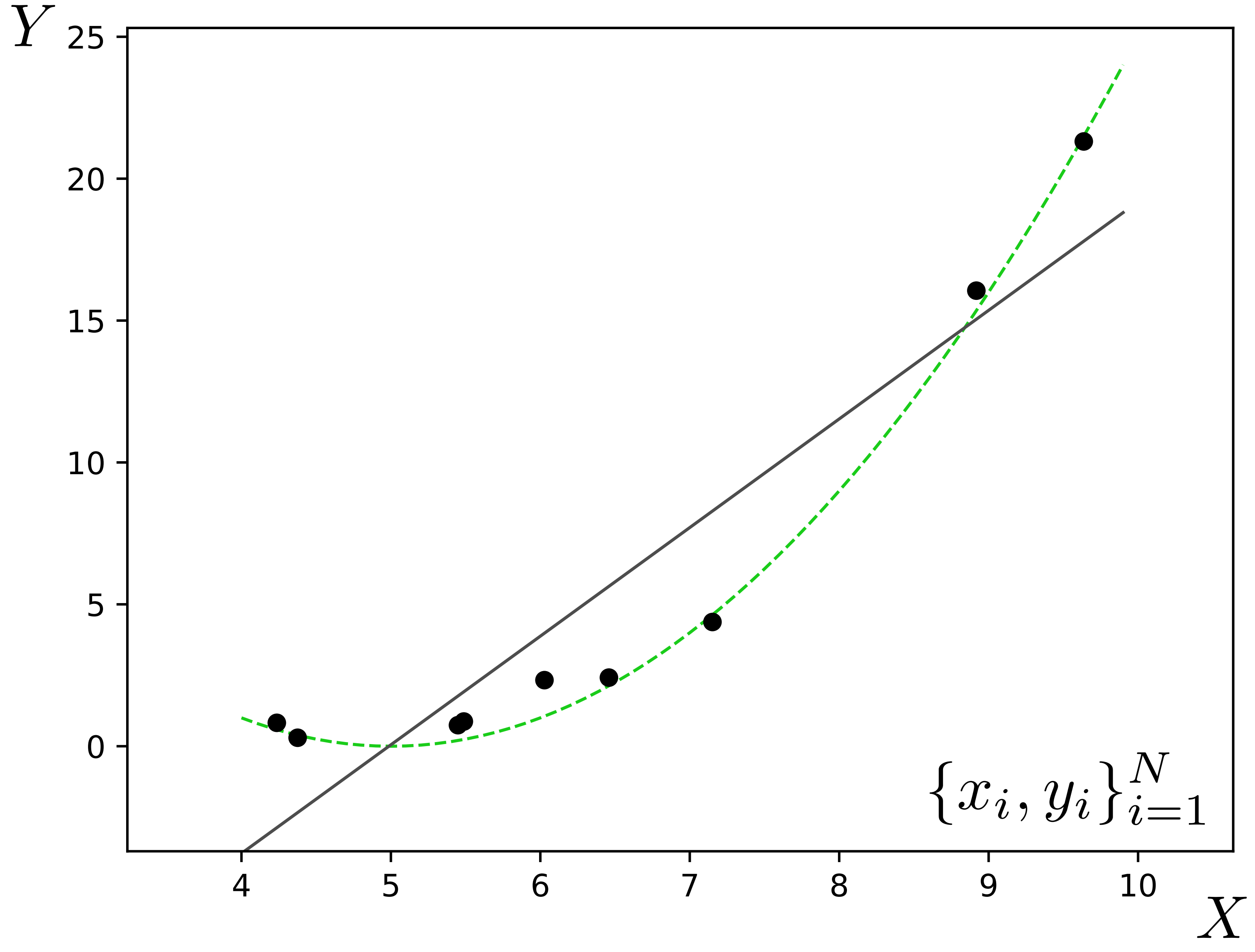
Lowest error on train set

High error on test set

Source: Isola, Torralba, Freeman

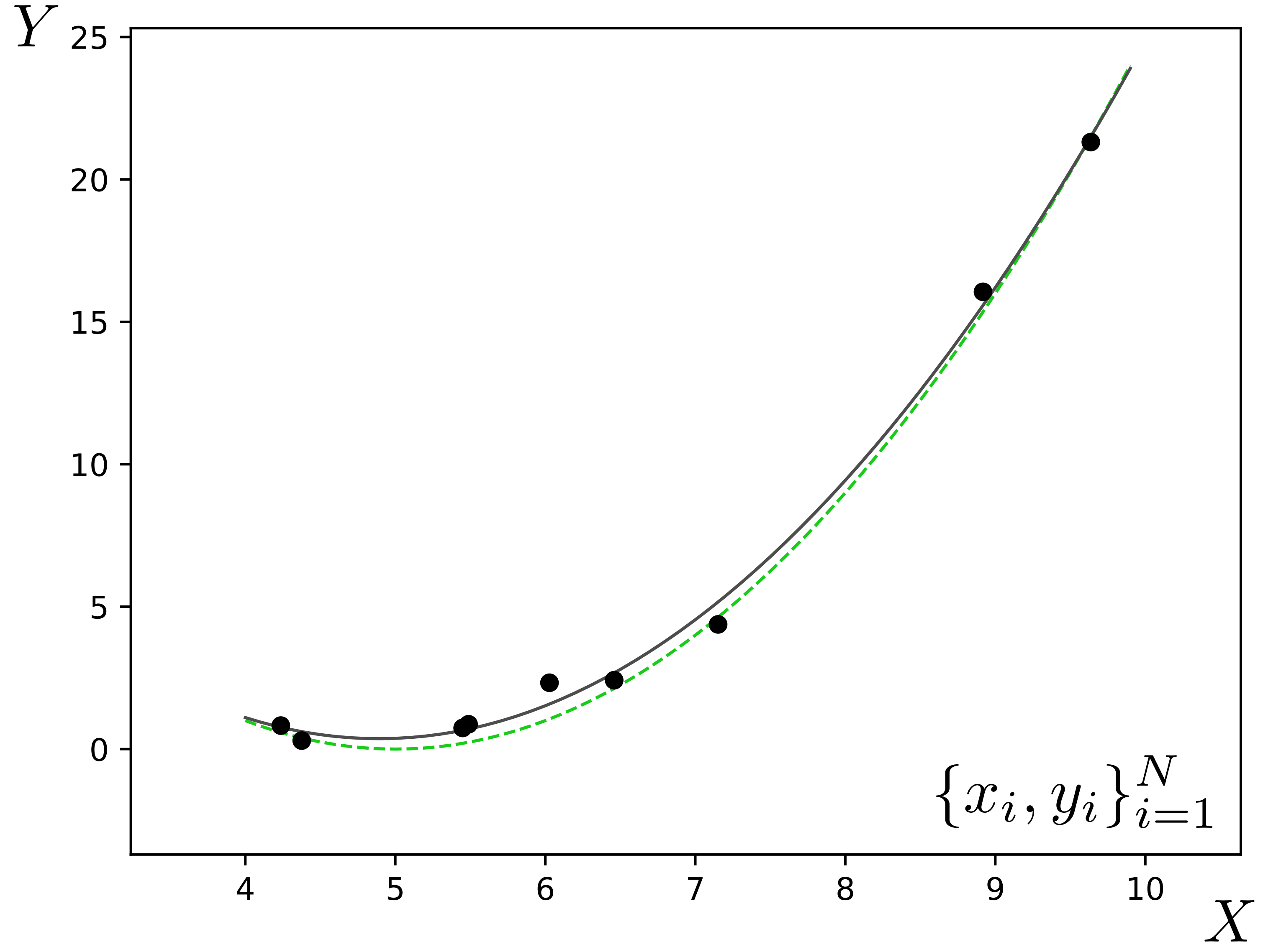
Underfitting

$\boxed{K = 1}$



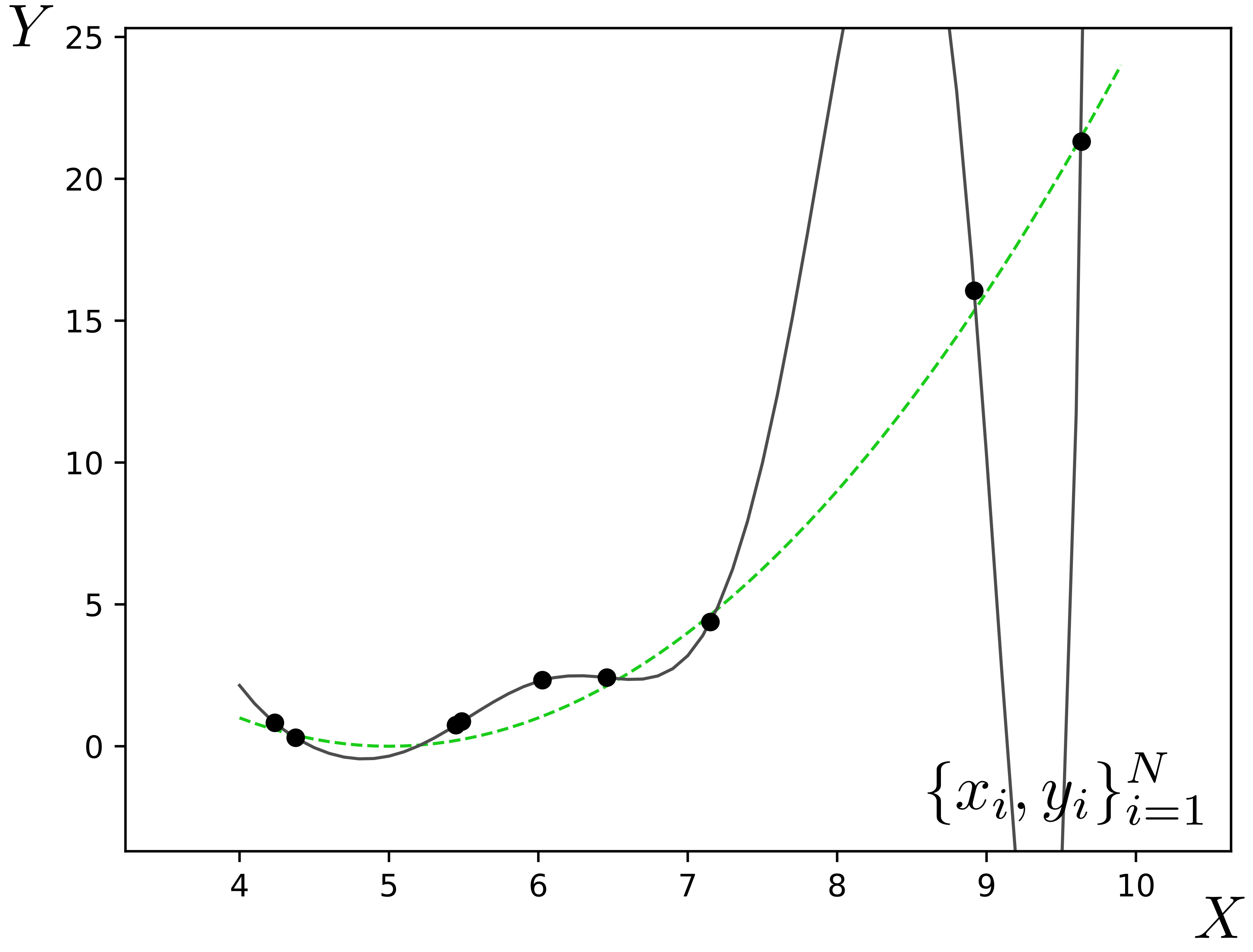
Appropriate model

$\boxed{K = 2}$



Overfitting

$\boxed{K = 10}$



Source: Isola, Torralba, Freeman

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

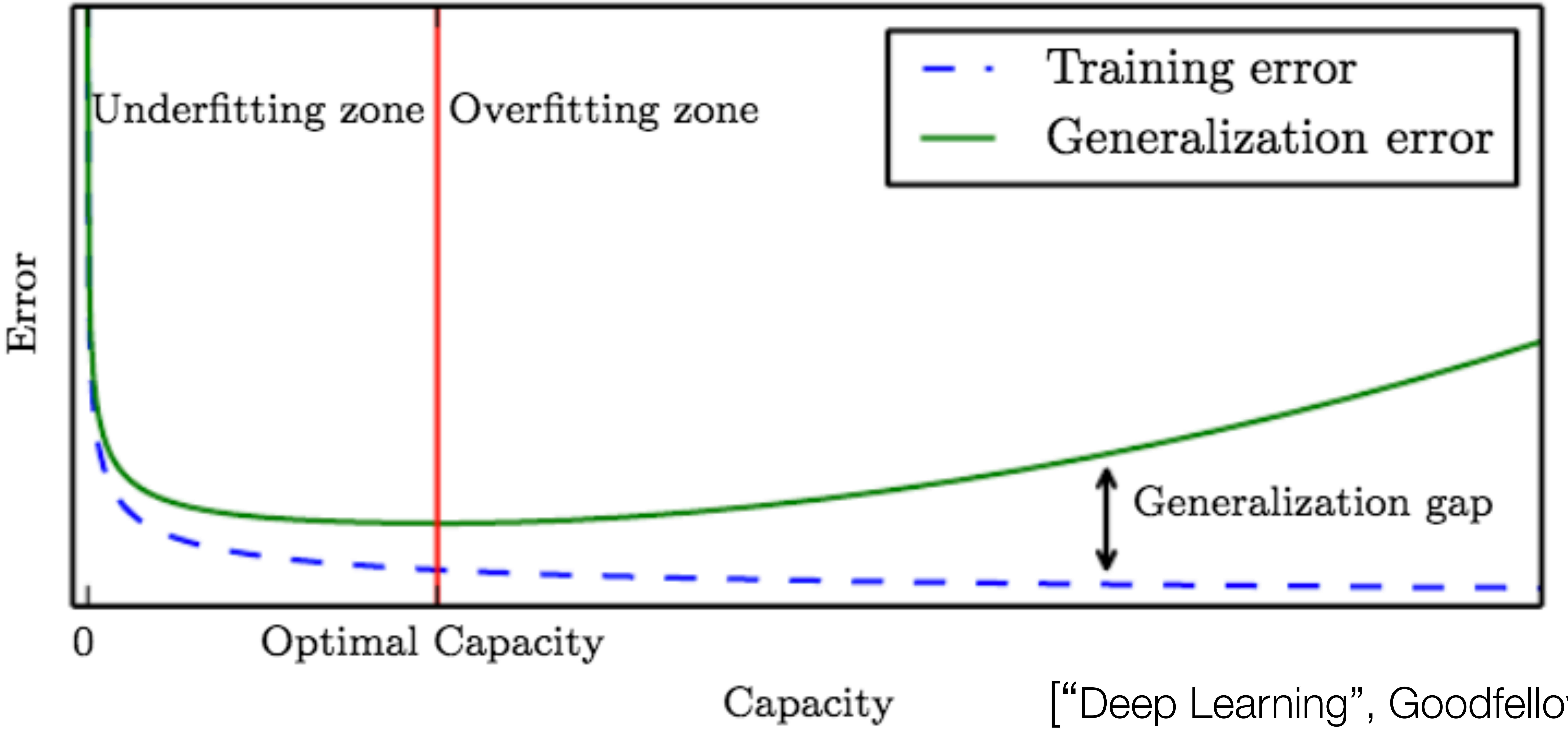
The capacity may be defined as the number of hypotheses under consideration in the hypothesis space.

Complex models with many free parameters have high capacity.

Simple models have low capacity.

Source: Isola, Torralba, Freeman

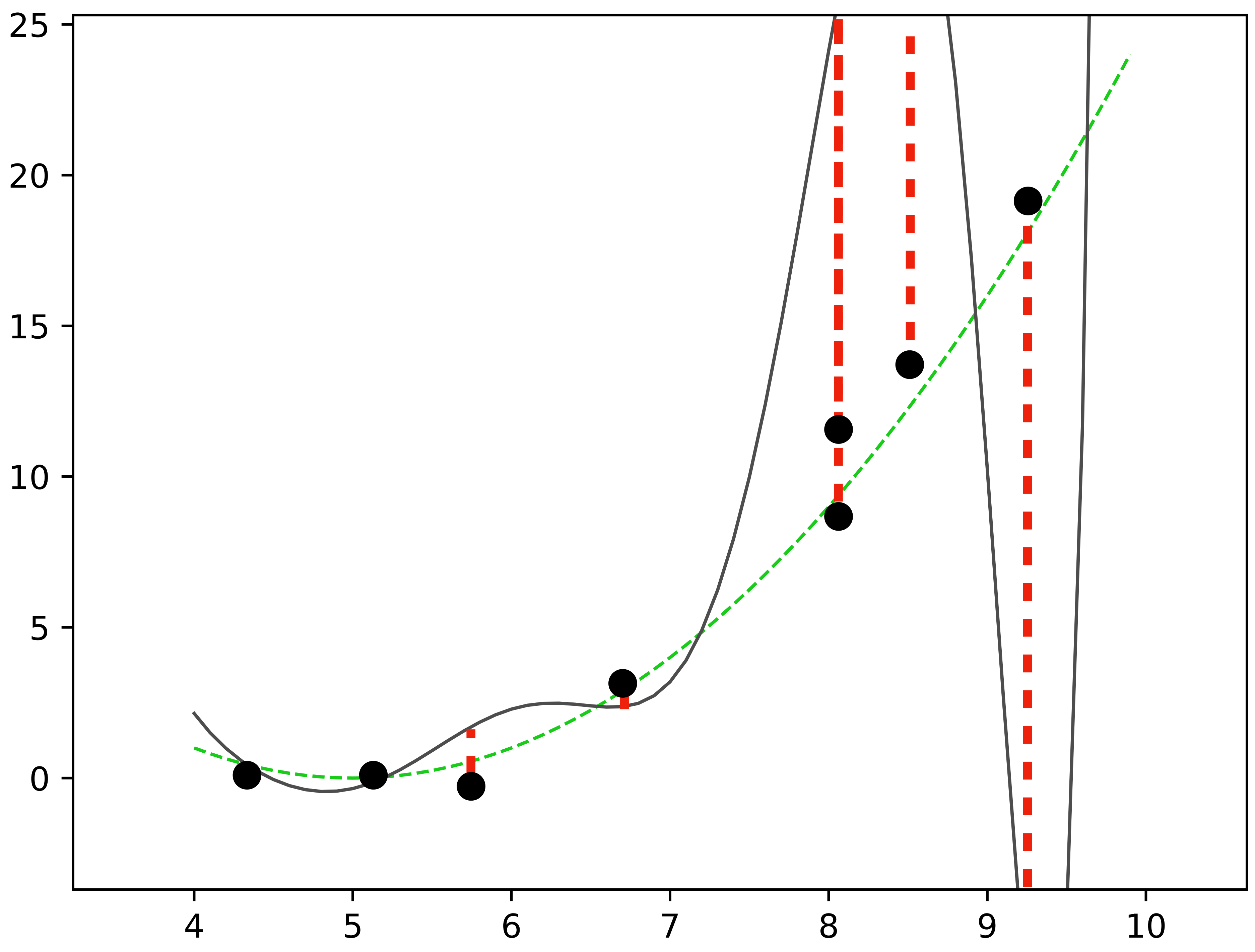
Training error versus generalization error

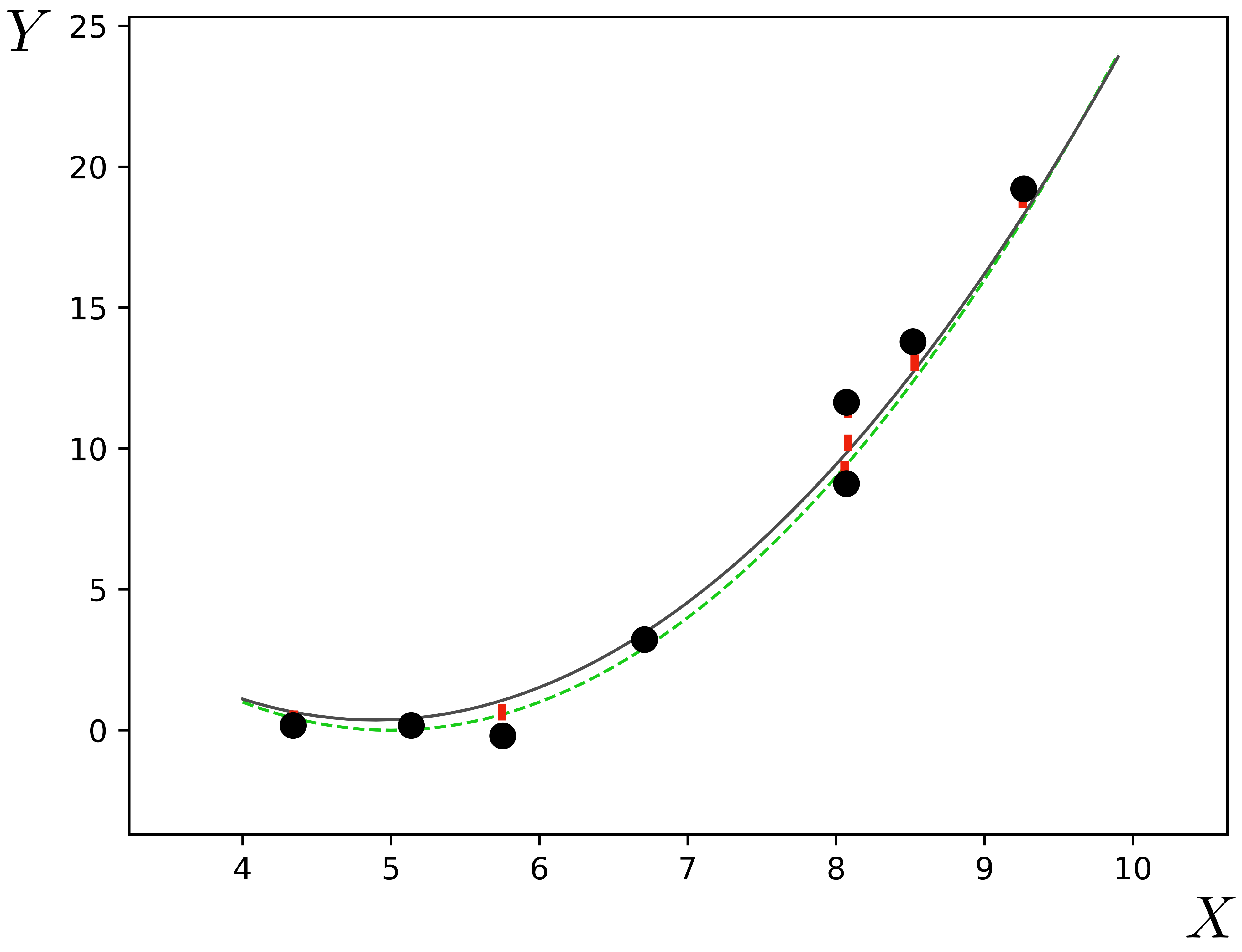


["Deep Learning", Goodfellow et al.]

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 data

Source: Isola, Torralba, Freeman

Fitting just right

Underfitting?

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

Overfitting?

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

Selecting a *hypothesis space* of functions with just the right capacity is known as **model selection**

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

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

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

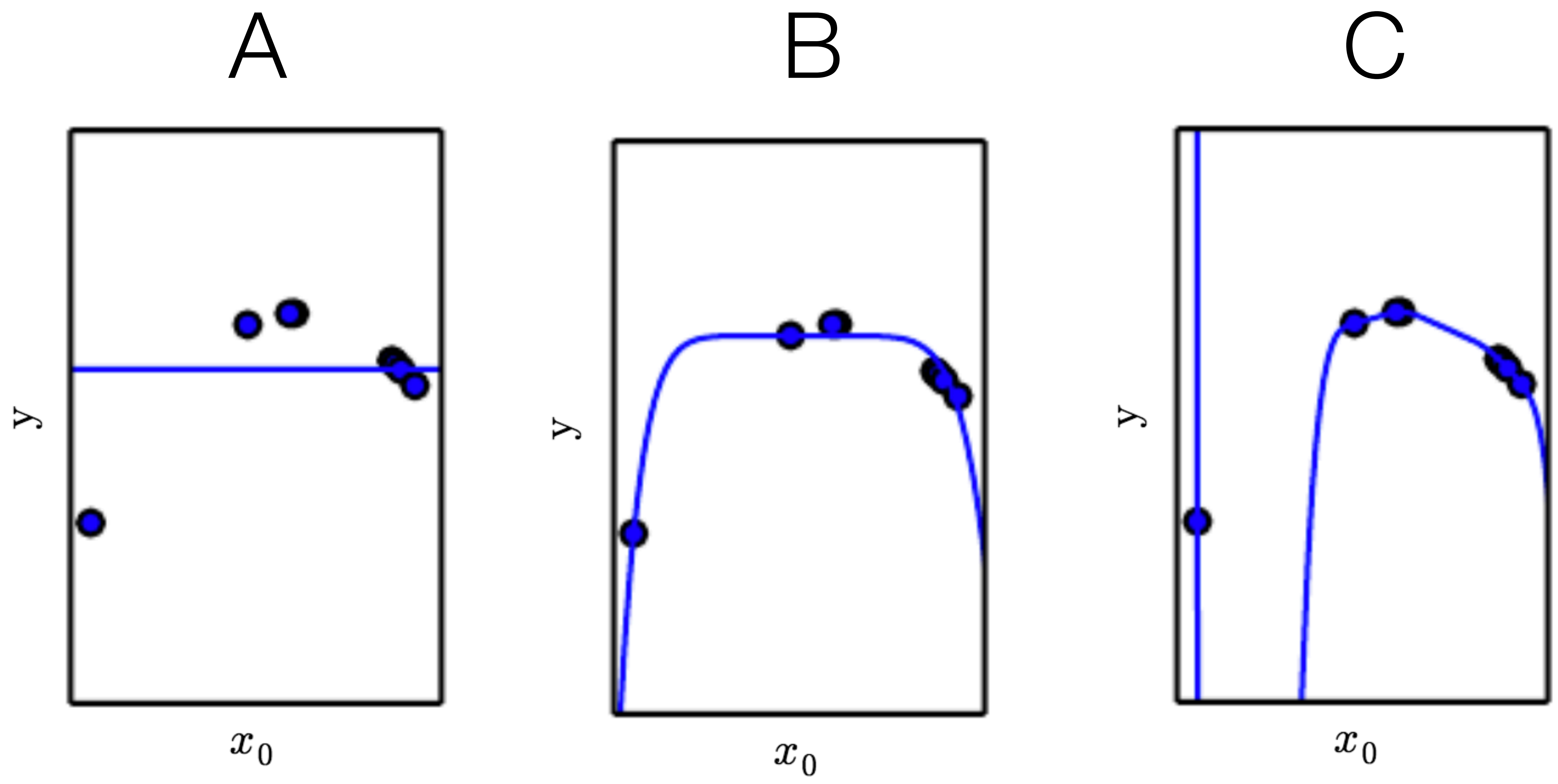
Only use polynomial terms if you really need them! Most terms should be zero

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

(Probabilistic interpretation: R is a Gaussian **prior** over values of the parameters.)

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 — ?

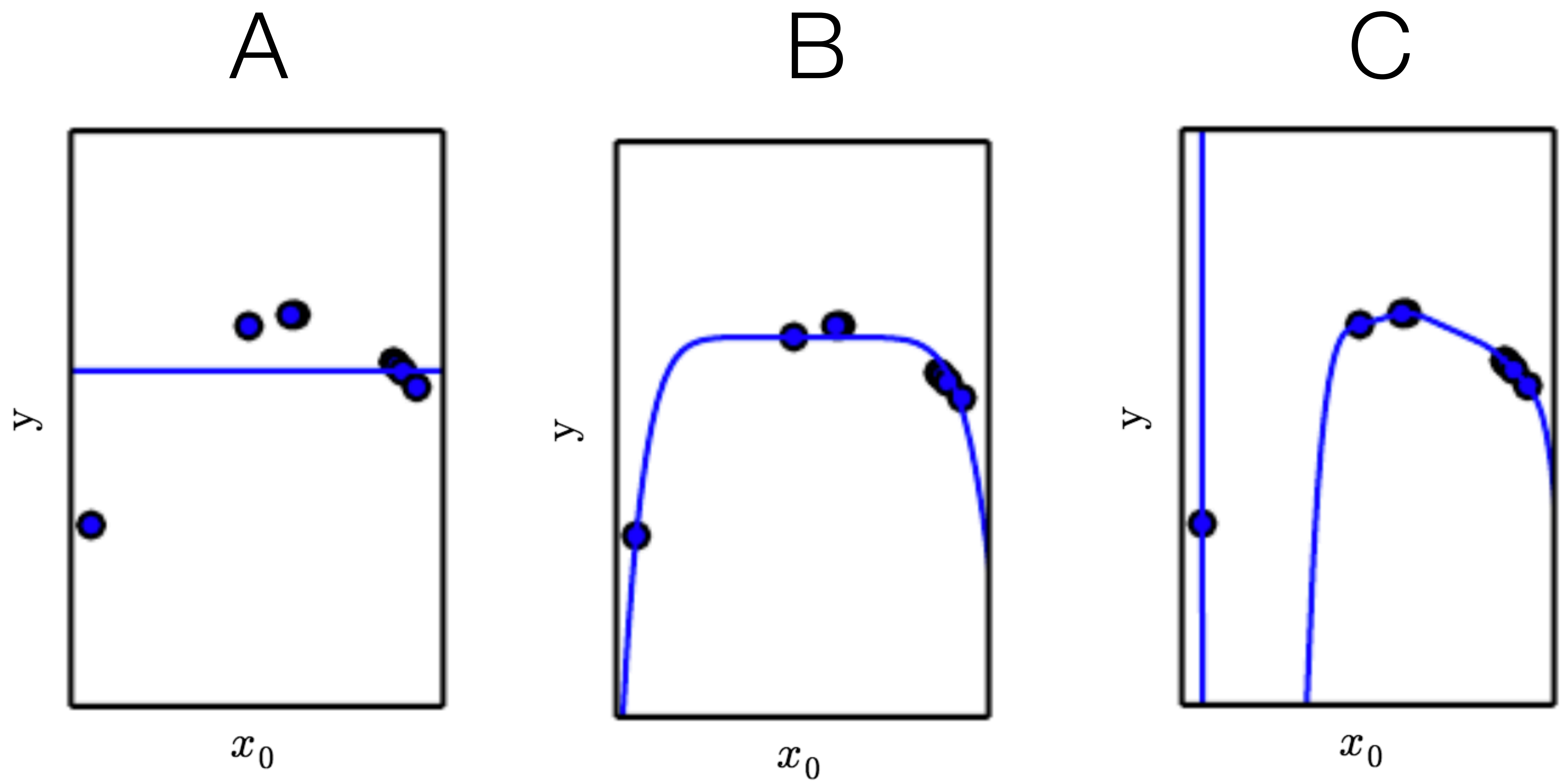
Medium λ — ?

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.]

Regularized polynomial least squares regression

Data

$\{x_i, y_i\}_{i=1}^N \quad \rightarrow$

Learner

Objective

$$\sum_{i=1}^{N} (f_\theta(x_i) - y_i)^2 + \lambda \|\theta\|_2^2$$

Hypothesis space

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

Optimizer

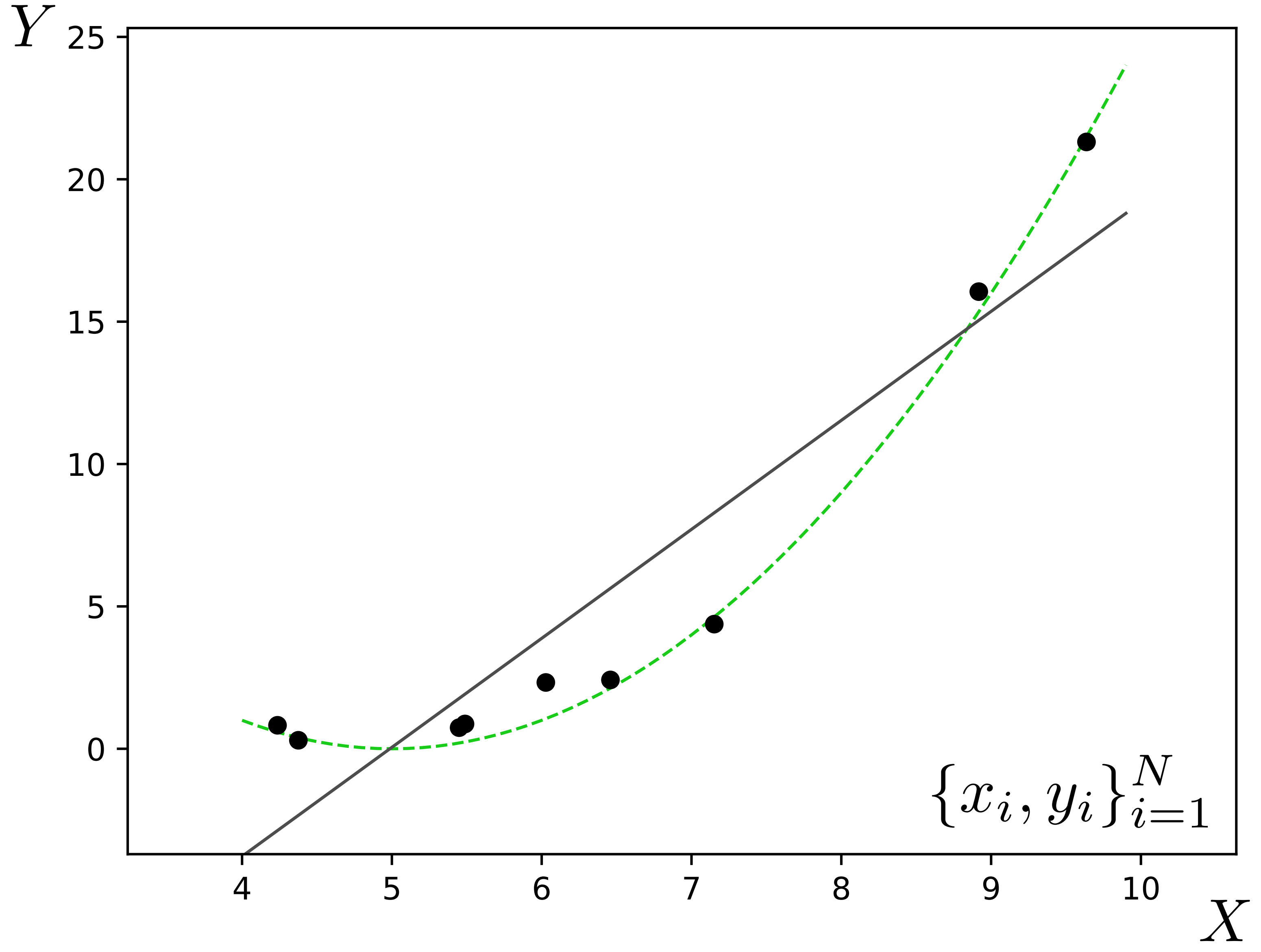
$$\theta^* = (\mathbf{\Phi}^T \mathbf{\Phi} + \lambda I)^{-1} \mathbf{\Phi}^T \mathbf{y}$$

$\rightarrow f$

Source: Isola, Torralba, Freeman

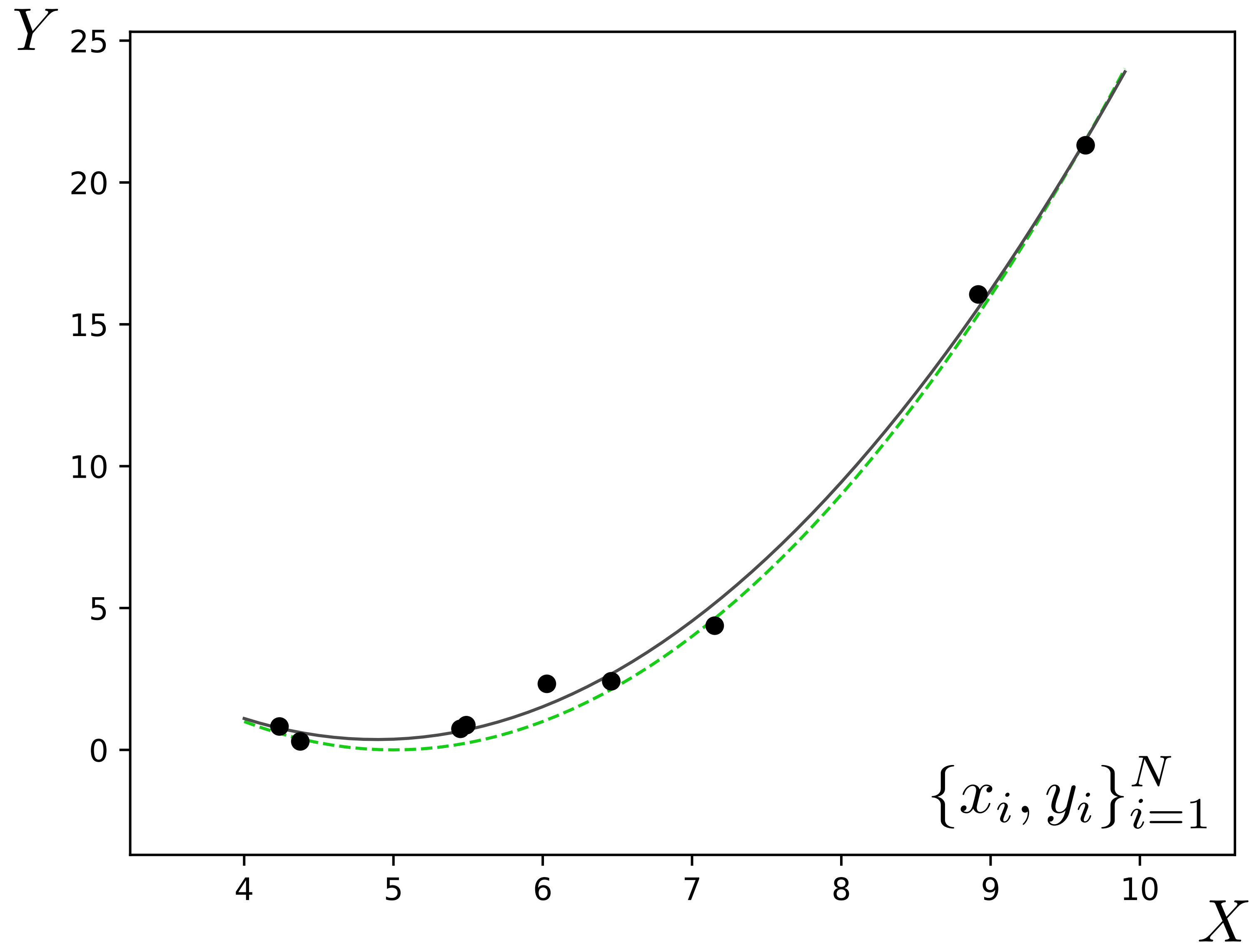
Underfitting

$\boxed{K = 1}$



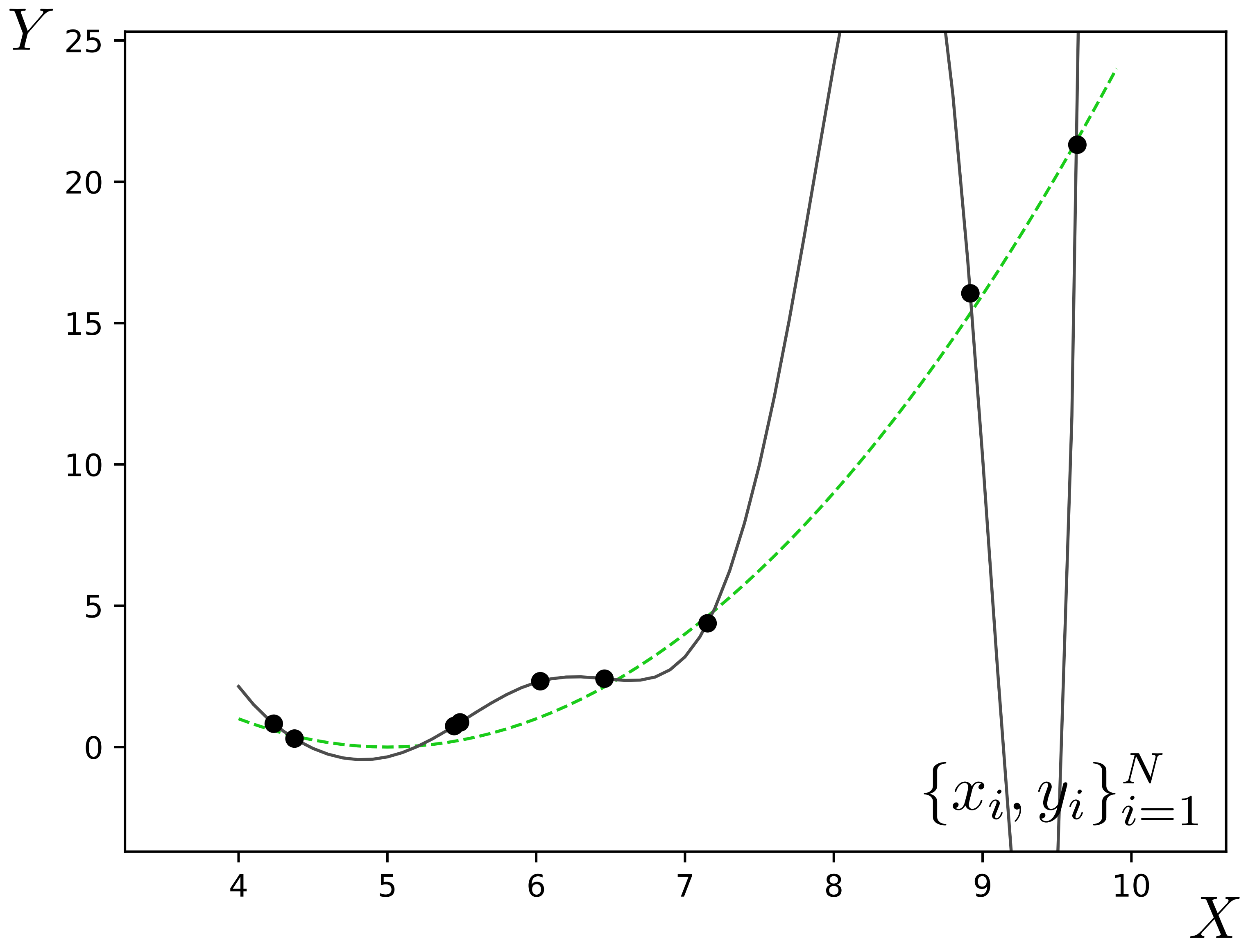
Appropriate model

$\boxed{K = 2}$



Overfitting

$\boxed{K = 10}$



Simple model

Doesn't fit the training data

Simple model

Fits the training data

Complex model

Fits the training data

Source: Isola, Torralba, Freeman

Next lecture: neural networks