香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 5212
Machine Learning
Lecture 9

Generalization,
Bias-Variance Tradeoff

Junxian He
Mar 12, 2026

Training and Test Data

Training and Test Data

- Training data is the data we see and use during model development

Training and Test Data

- Training data is the data we see and use during model development
- Test data is not observed during development

not train.

Bias-Variance Tradeoff

Bias-Variance Tradeoff

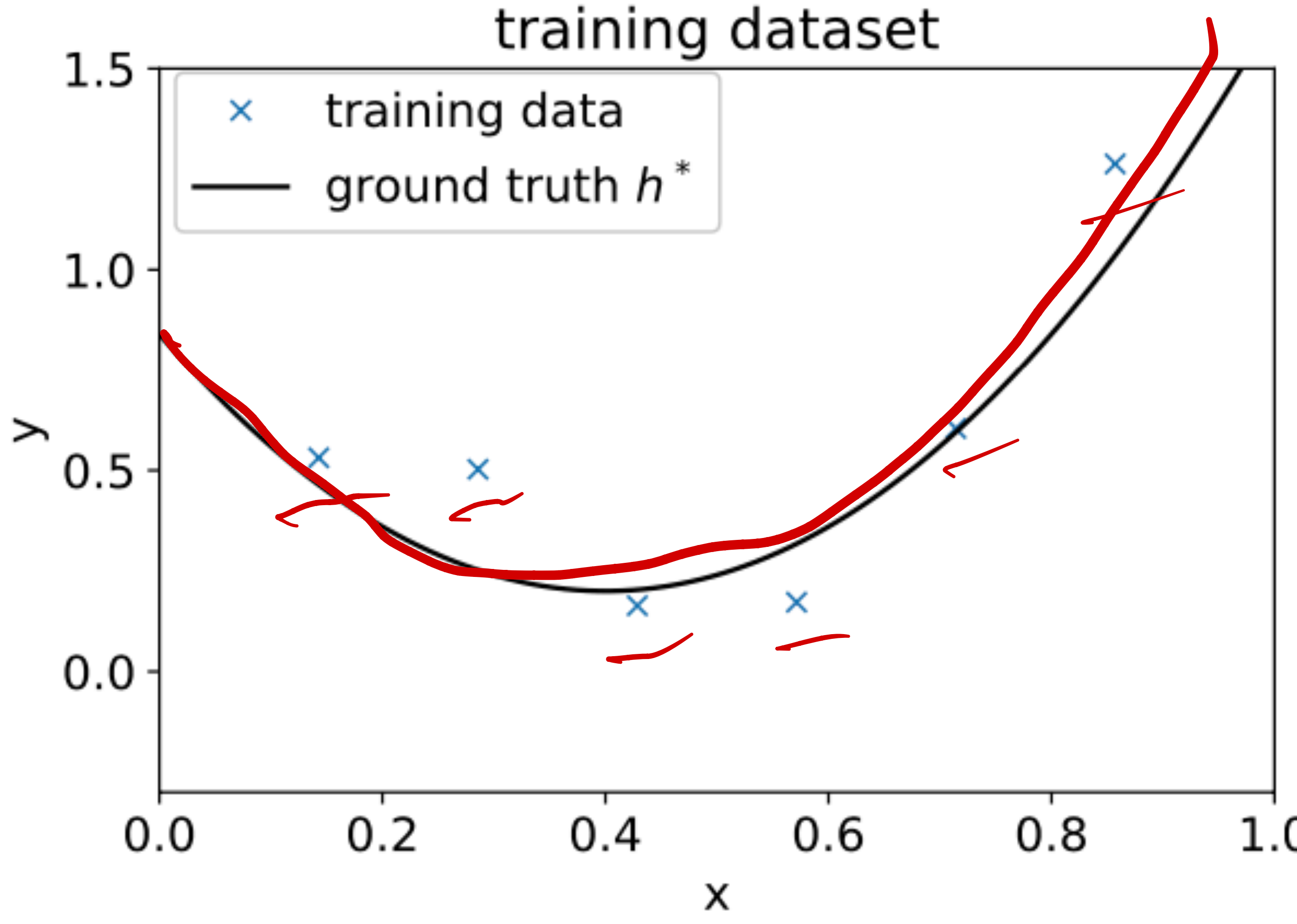
Suppose the data is generated from a quadratic function with noise

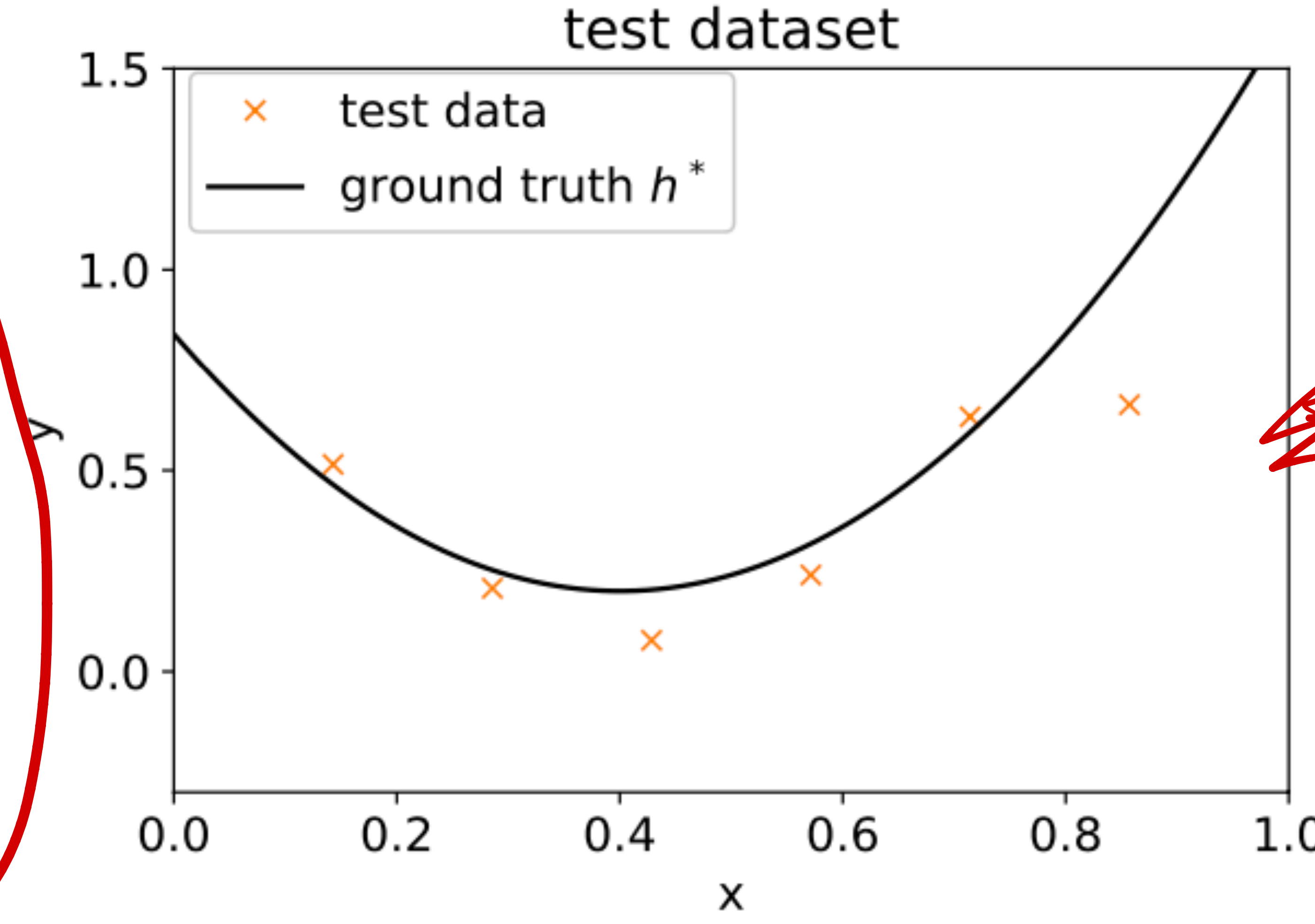
Bias-Variance Tradeoff

Suppose the data is generated from a quadratic function with noise

$$y^{(i)} = h^\star(x^{(i)}) + \xi^{(i)} \qquad \xi \sim N(0, \sigma^2)$$

Bias-Variance Tradeoff



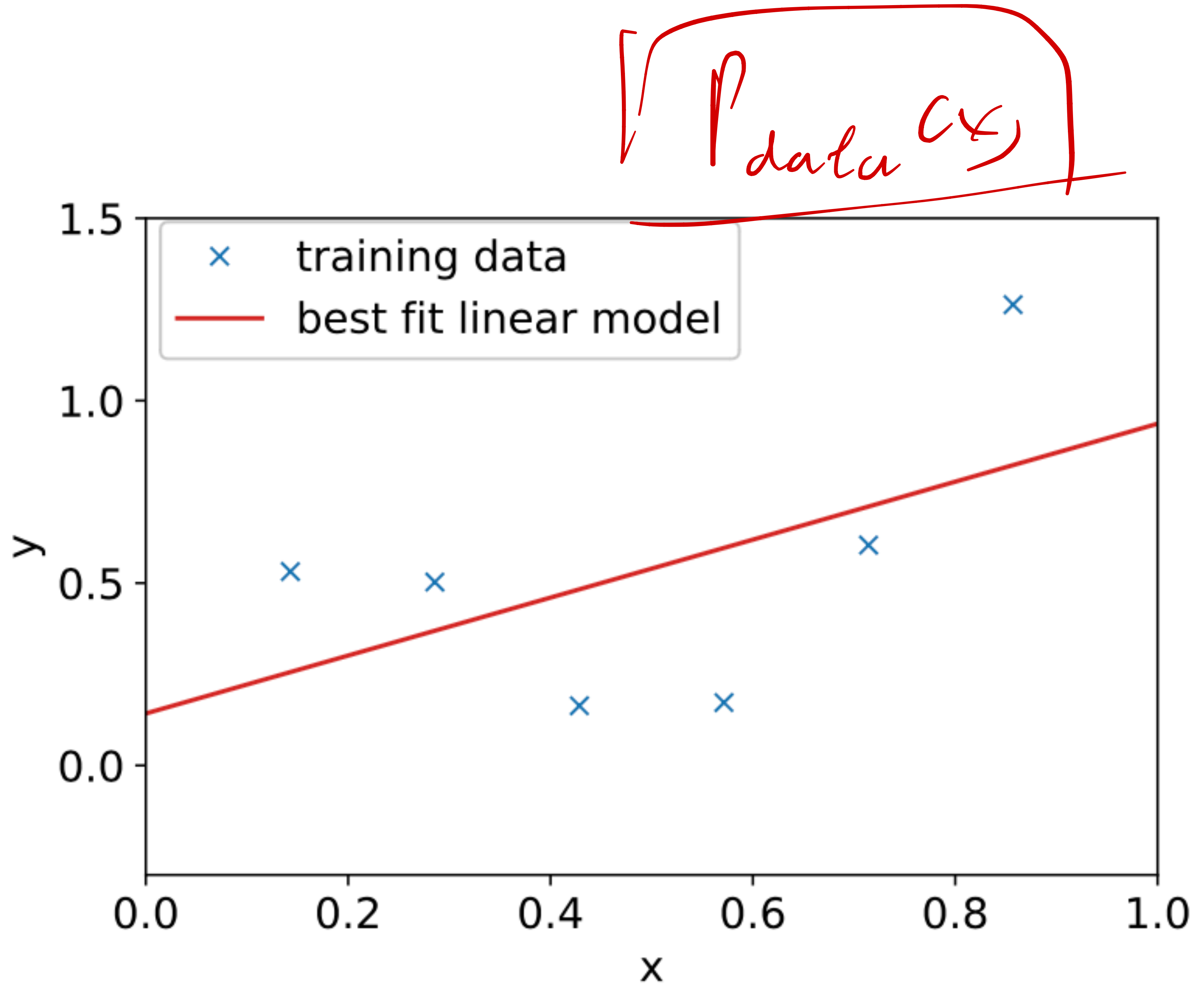


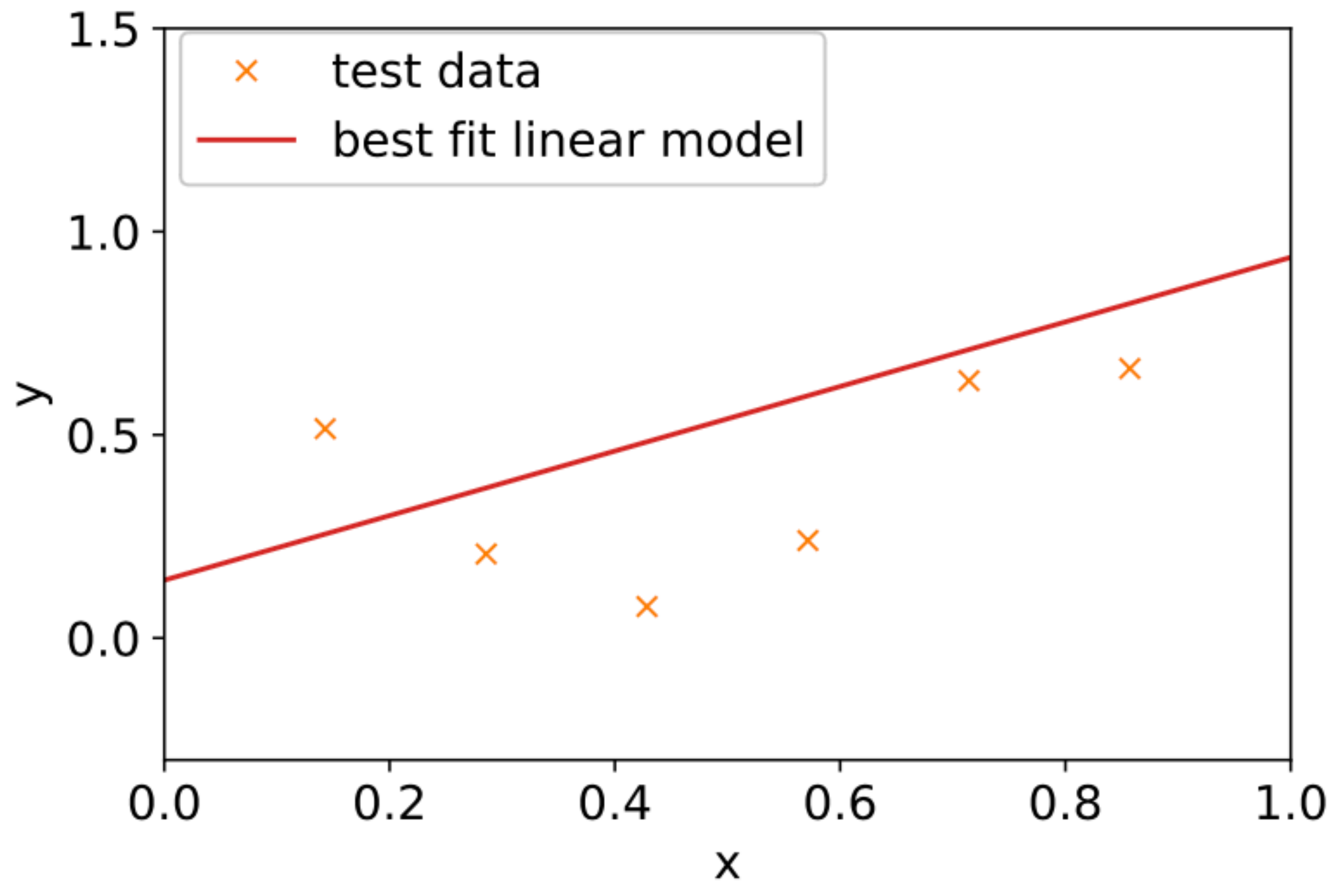
Suppose the data is generated from a quadratic function with noise

$$y^{(i)} = h^\star(x^{(i)}) + \xi^{(i)} \qquad \xi \sim N(0, \sigma^2)$$

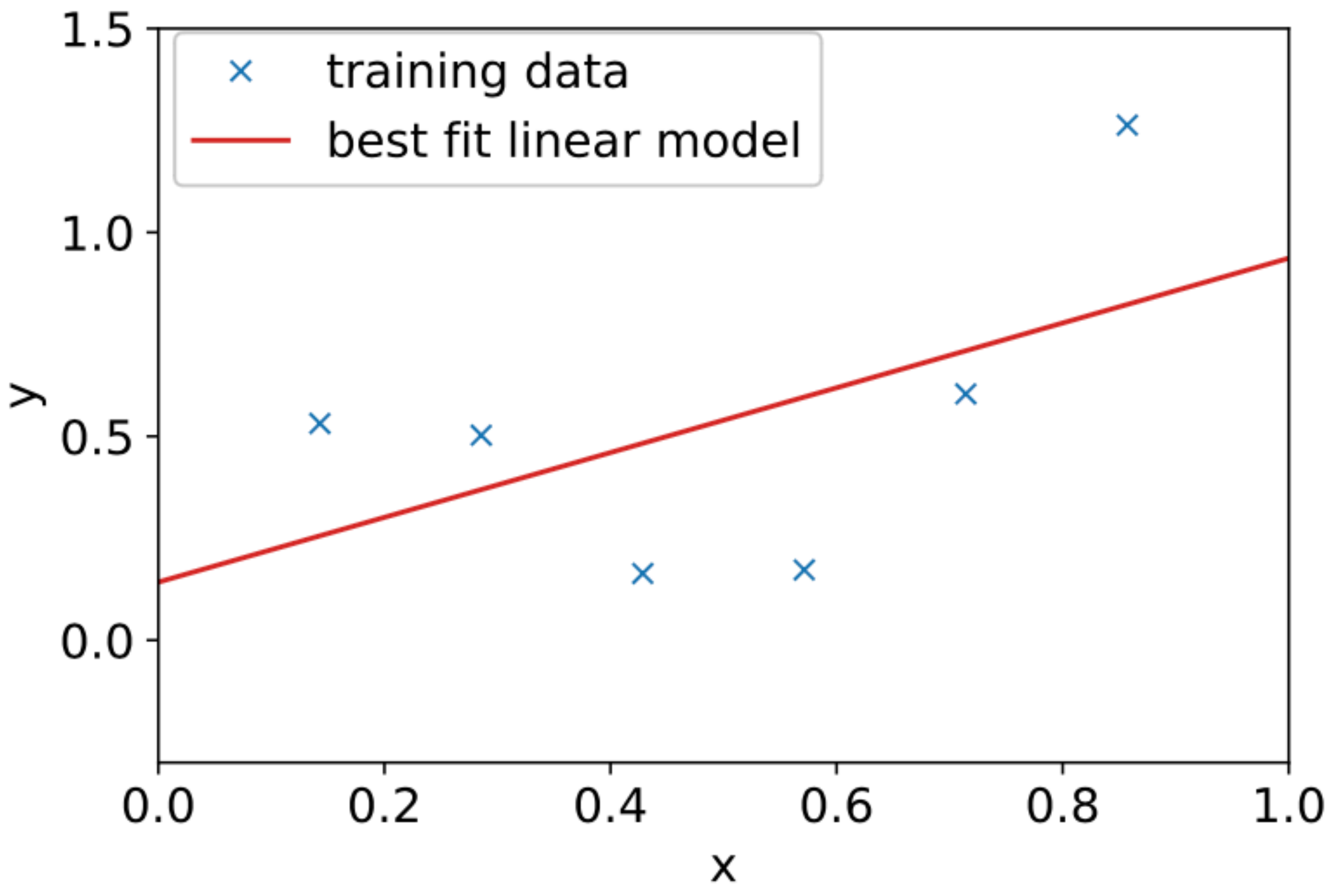
Fitting a Linear Model

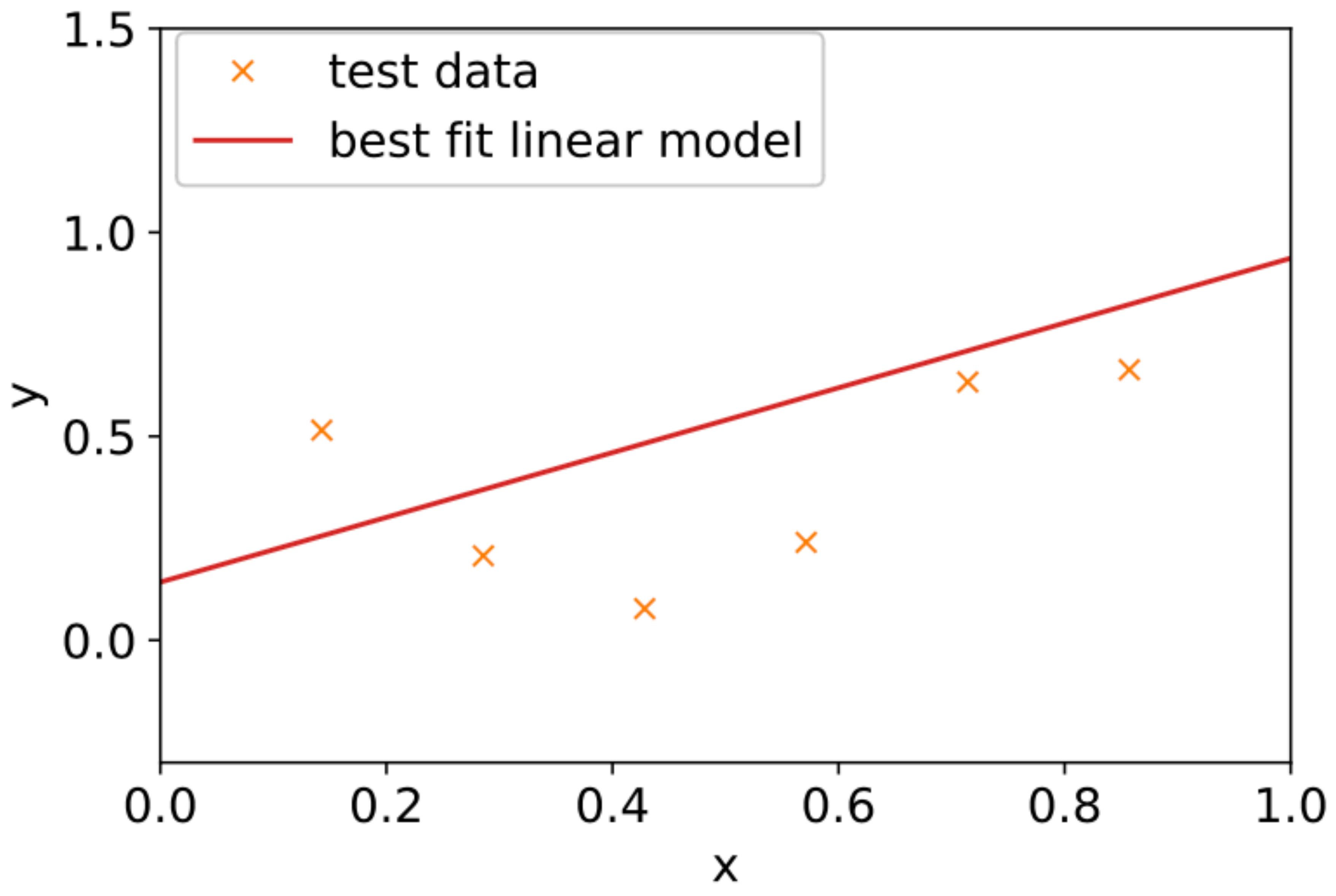
Fitting a Linear Model





Fitting a Linear Model

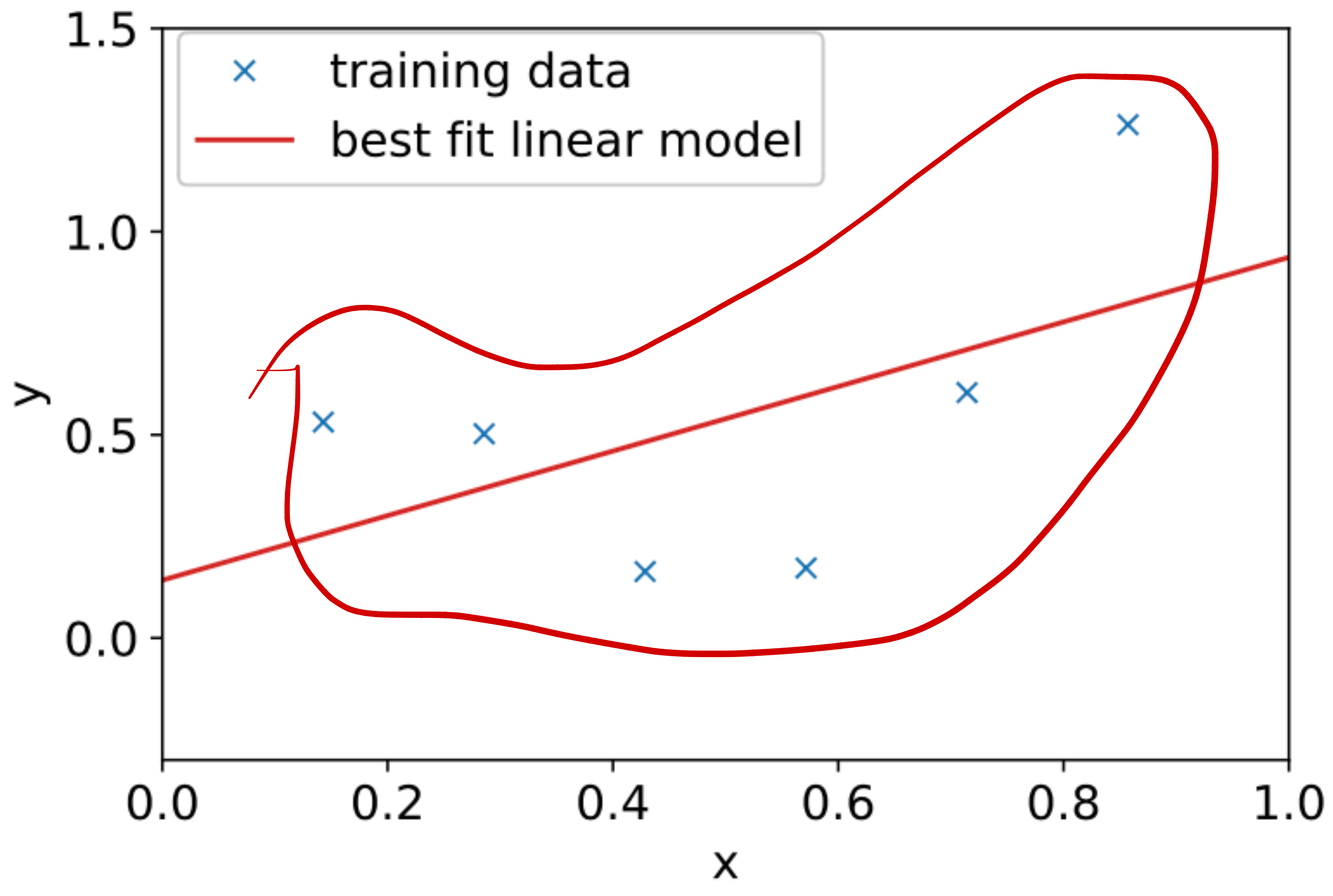


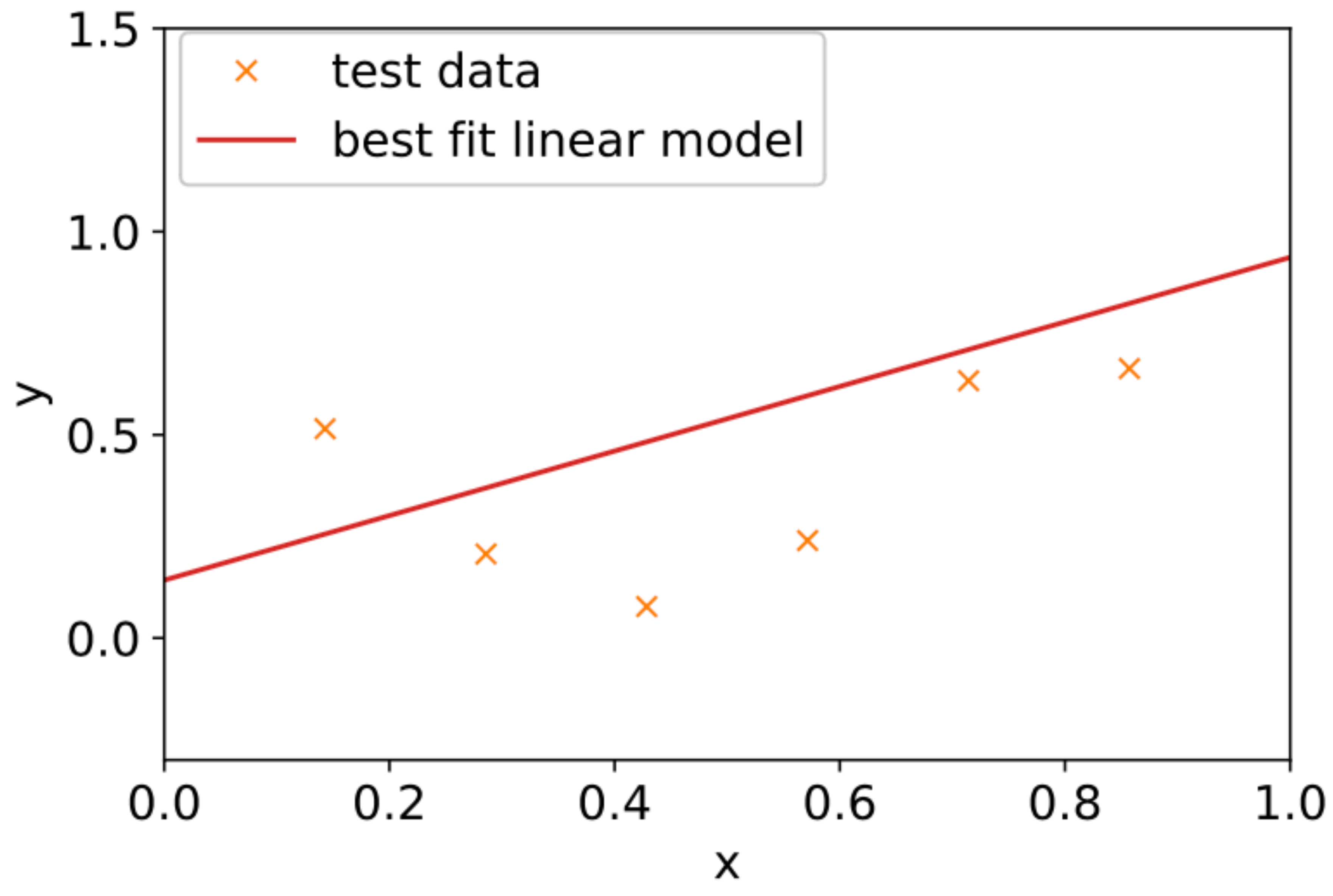


$$\text{Error} = \mathbb{E}_x[(y - h(x))^2]$$

MSE

Fitting a Linear Model



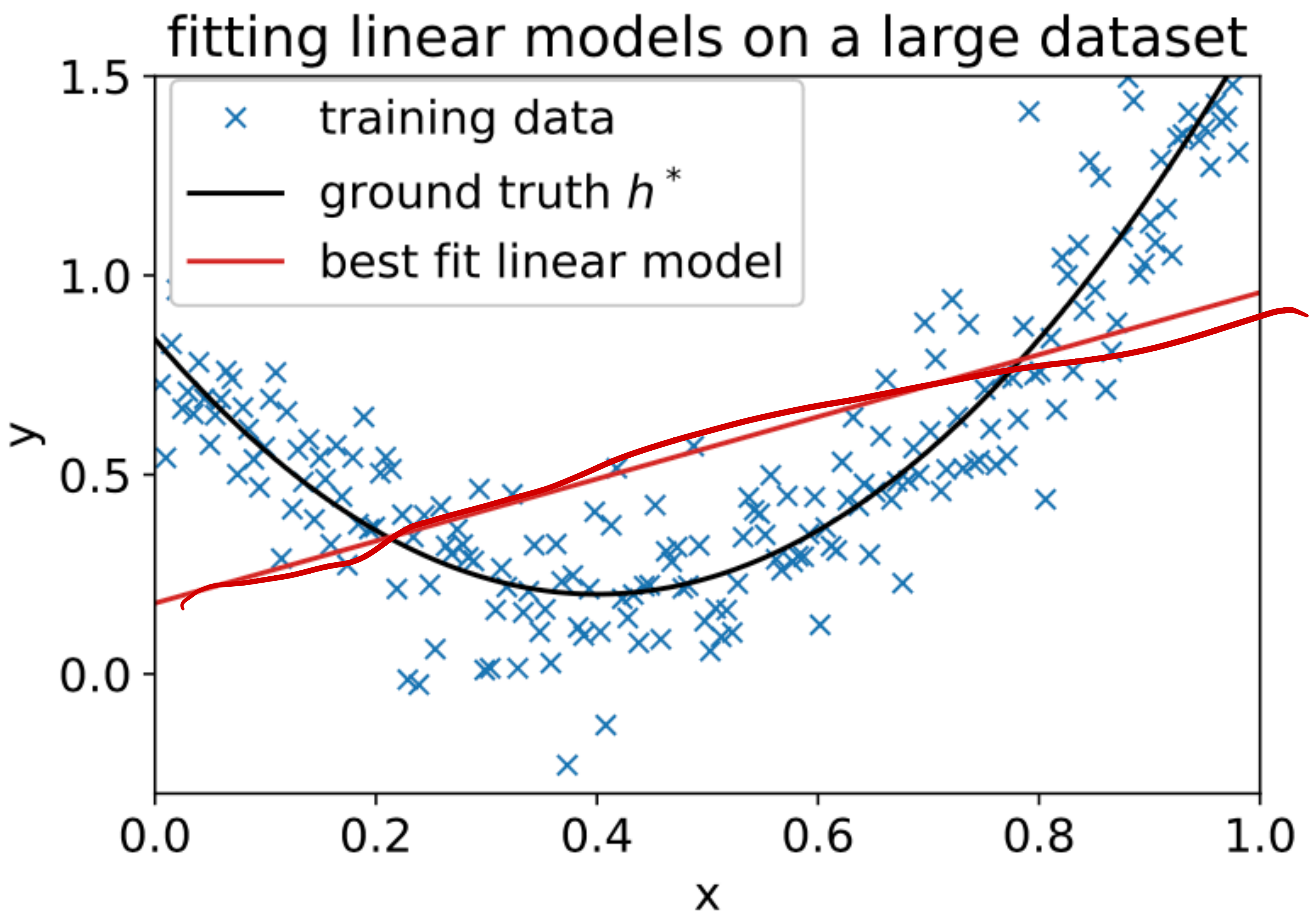


Error = $\mathbb{E}_x[(y - h(x))^2]$

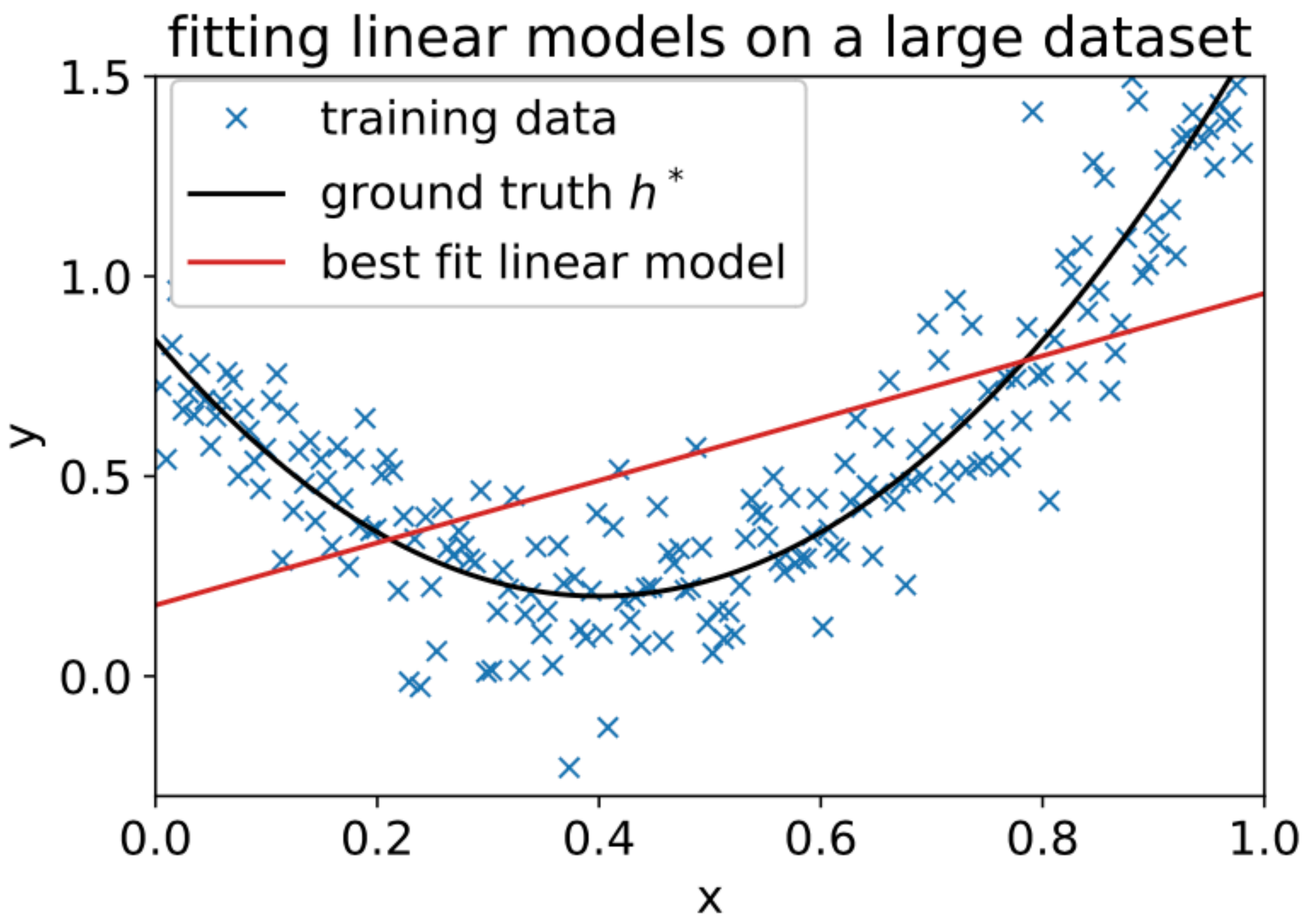
The best linear model has large training and test errors on this dataset

Fitting a Linear Model

Fitting a Linear Model

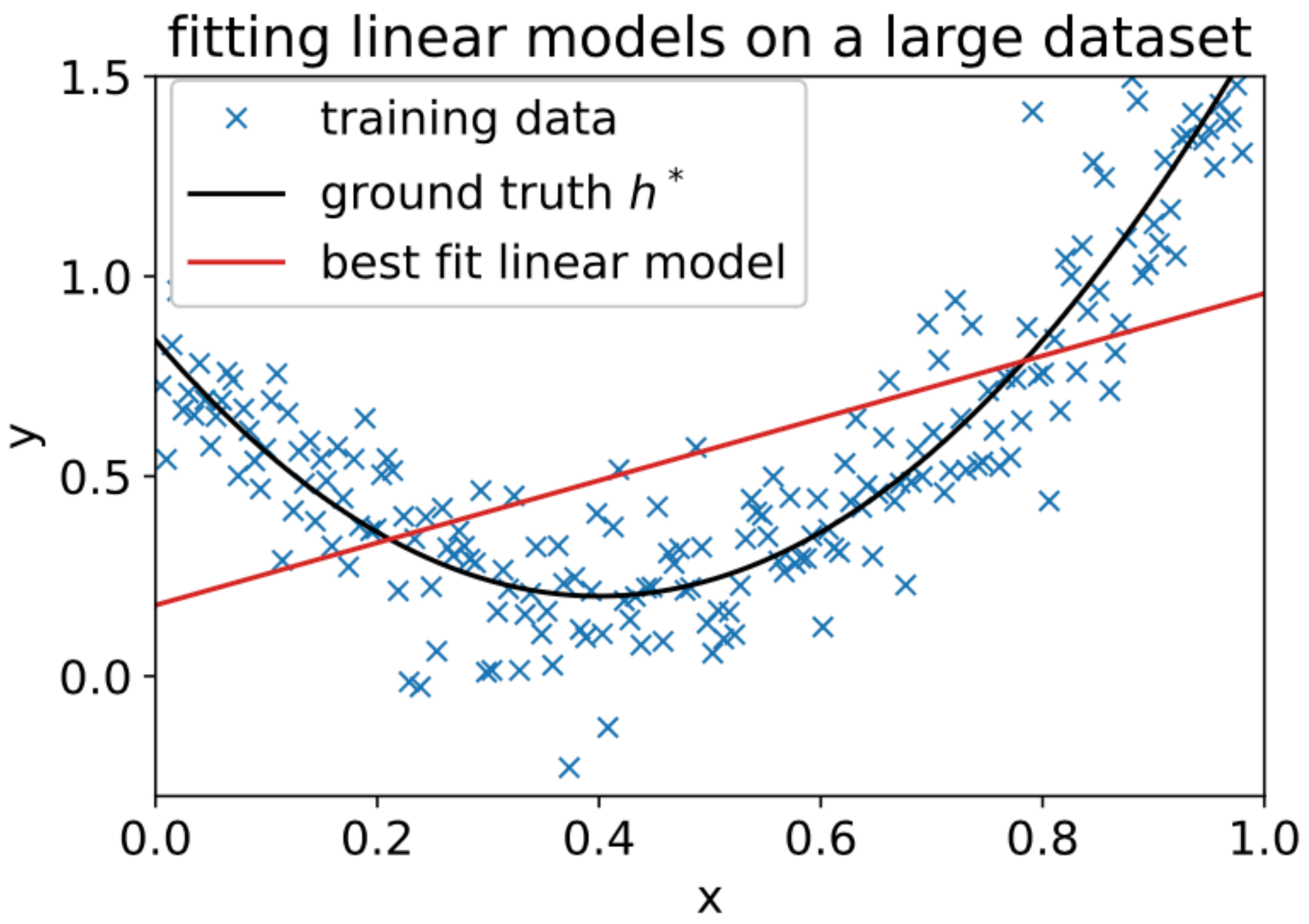


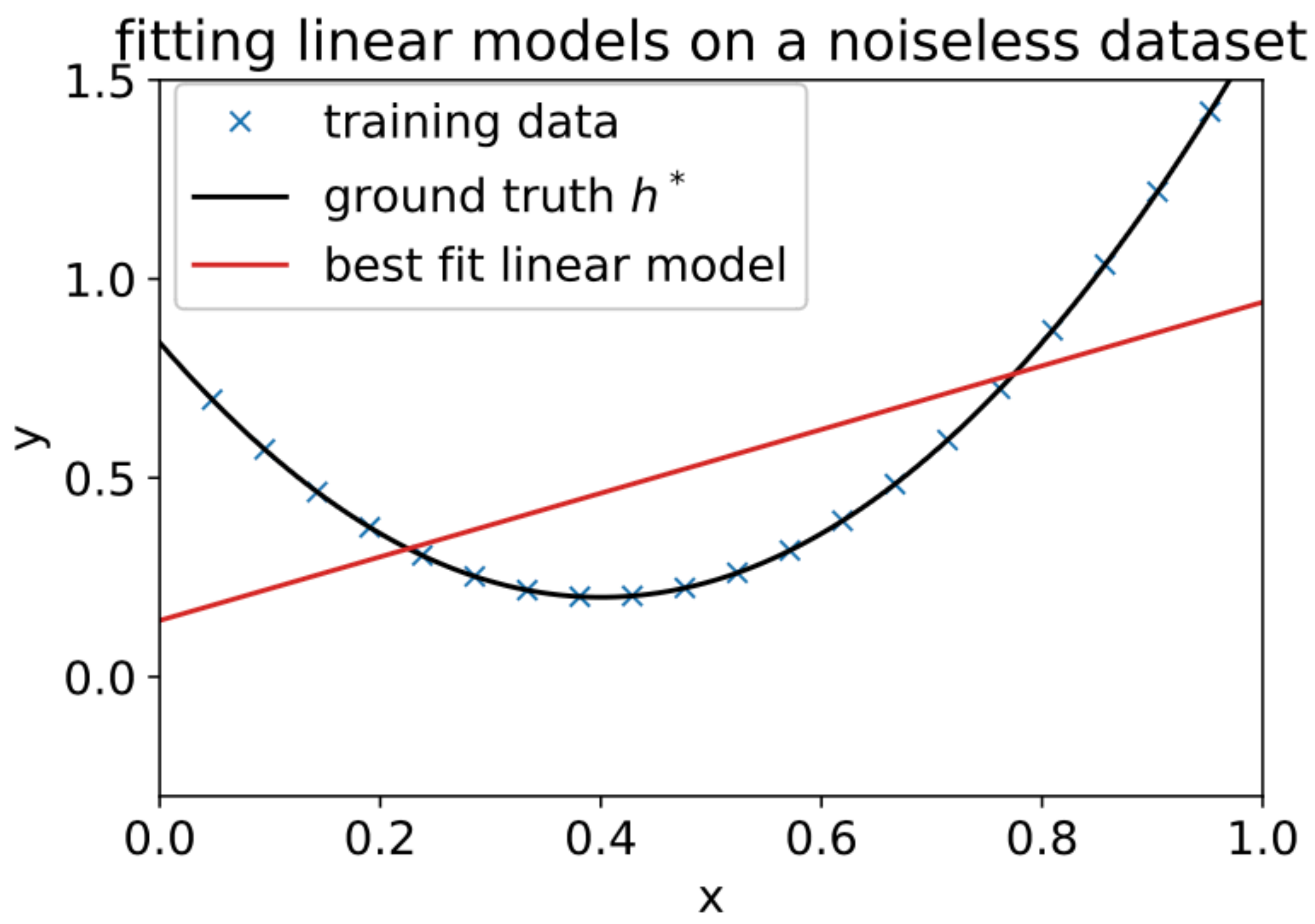
Fitting a Linear Model



Error is still large when we have many training samples

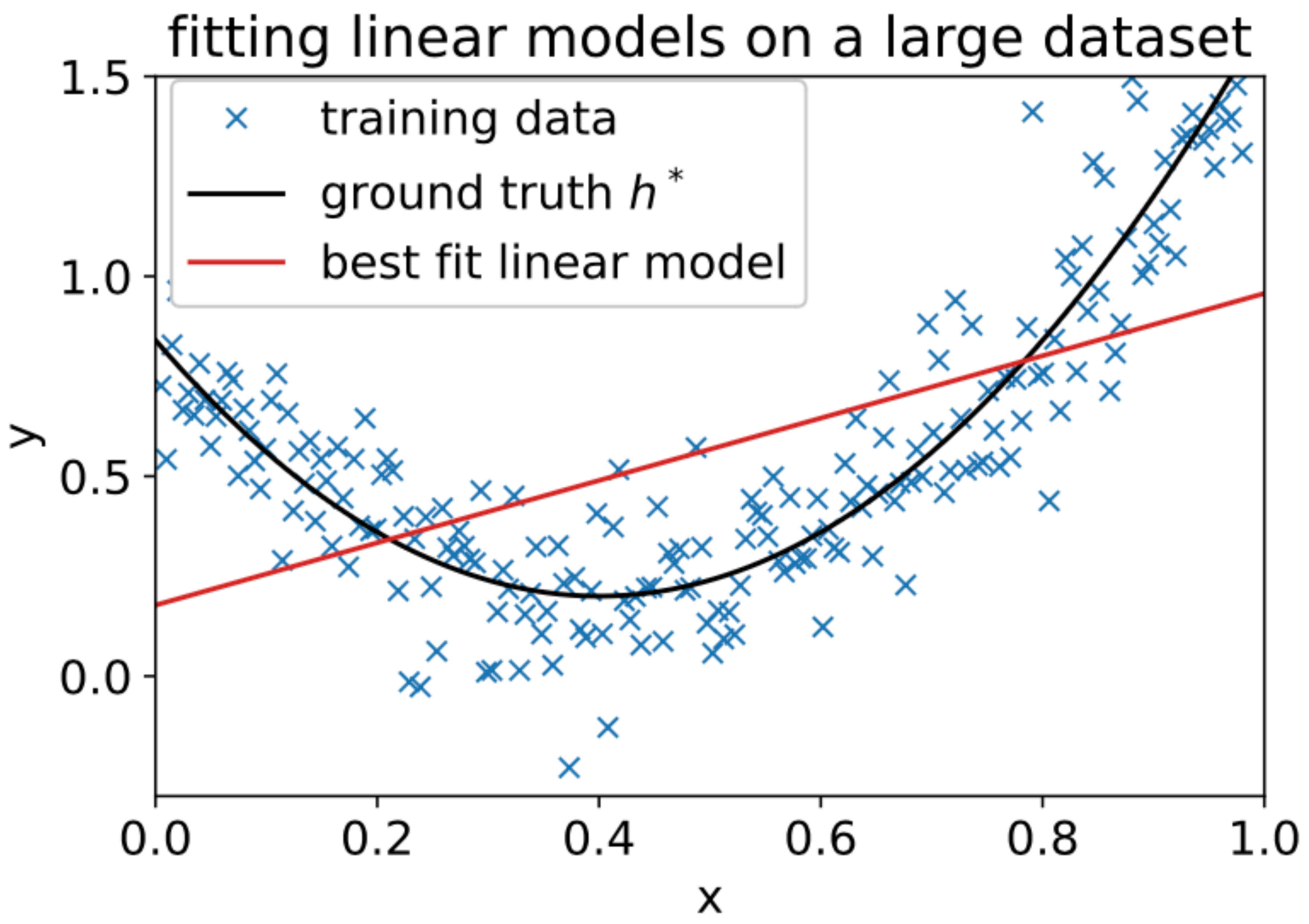
Fitting a Linear Model



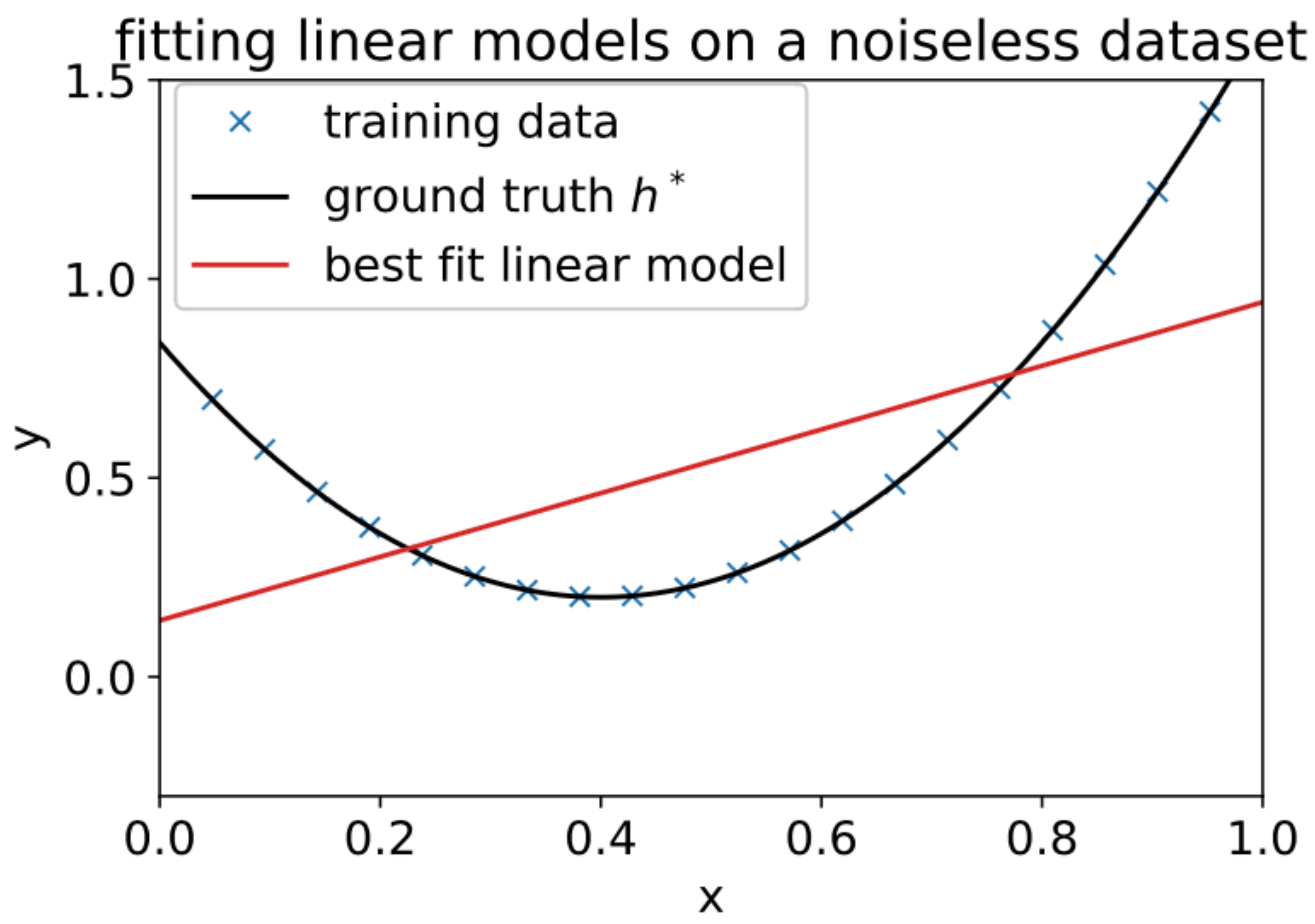


Error is still large when we have many training samples

Fitting a Linear Model

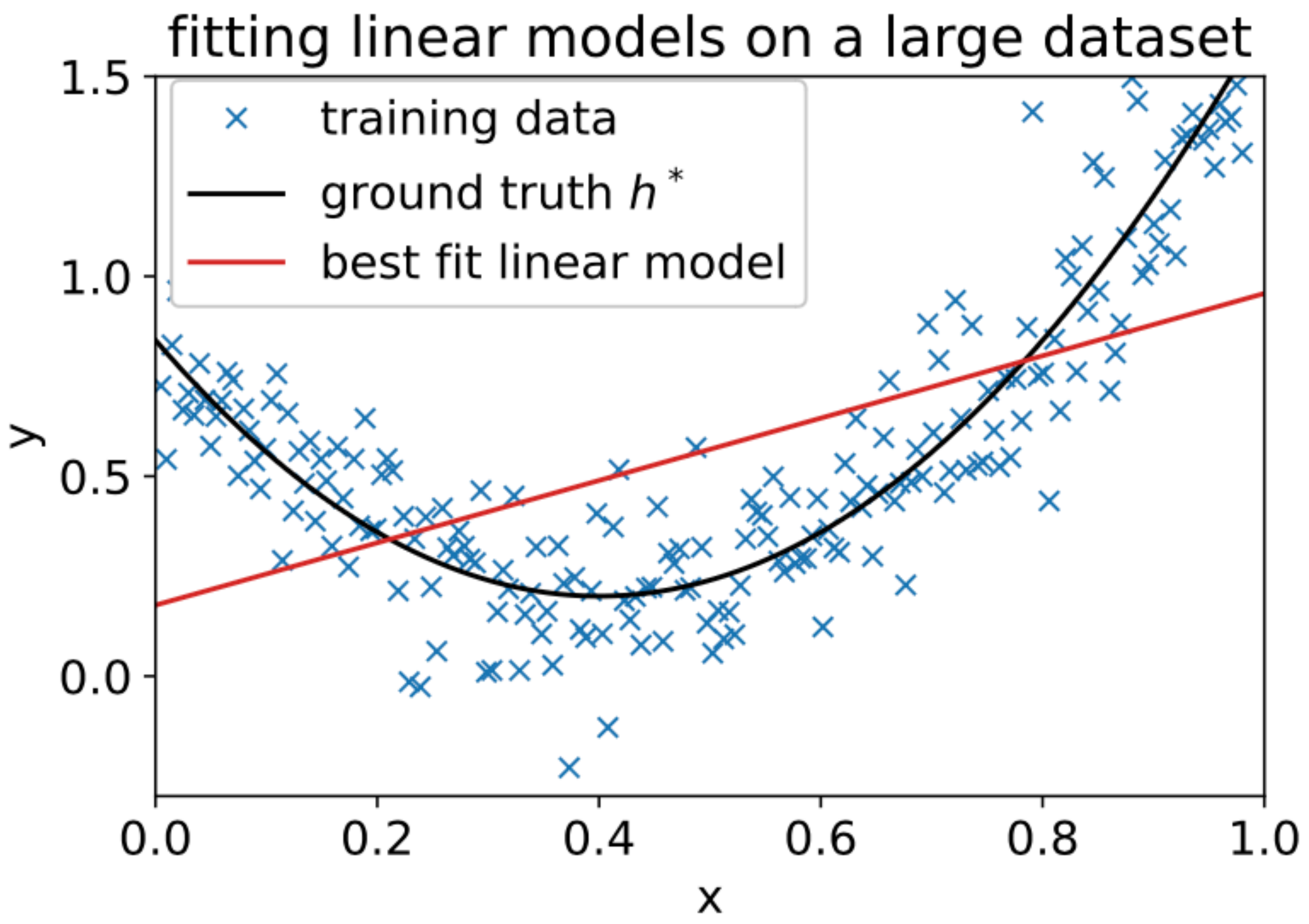


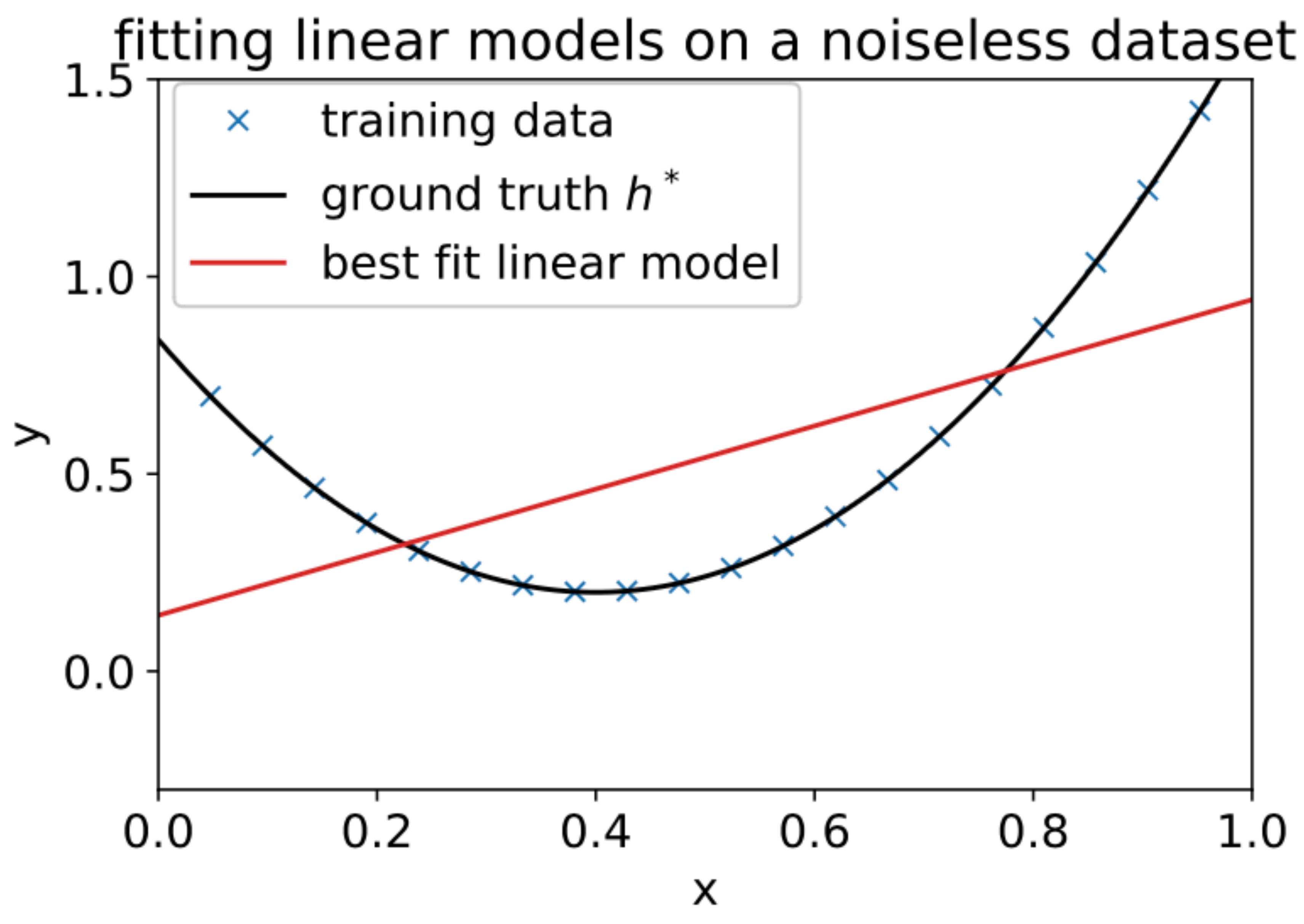
Error is still large when we have many training samples



Error is still large when we do not have noise

Fitting a Linear Model



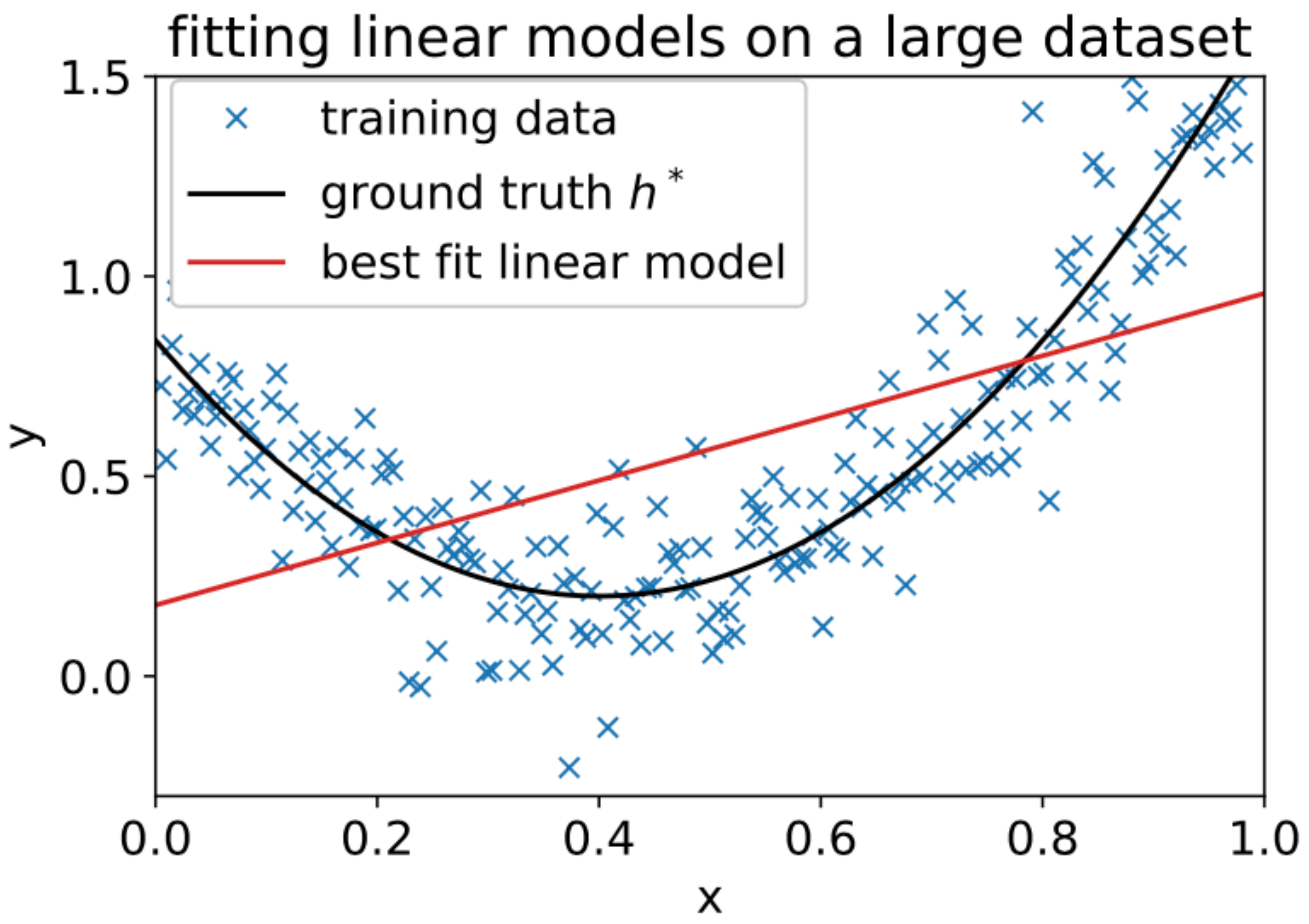


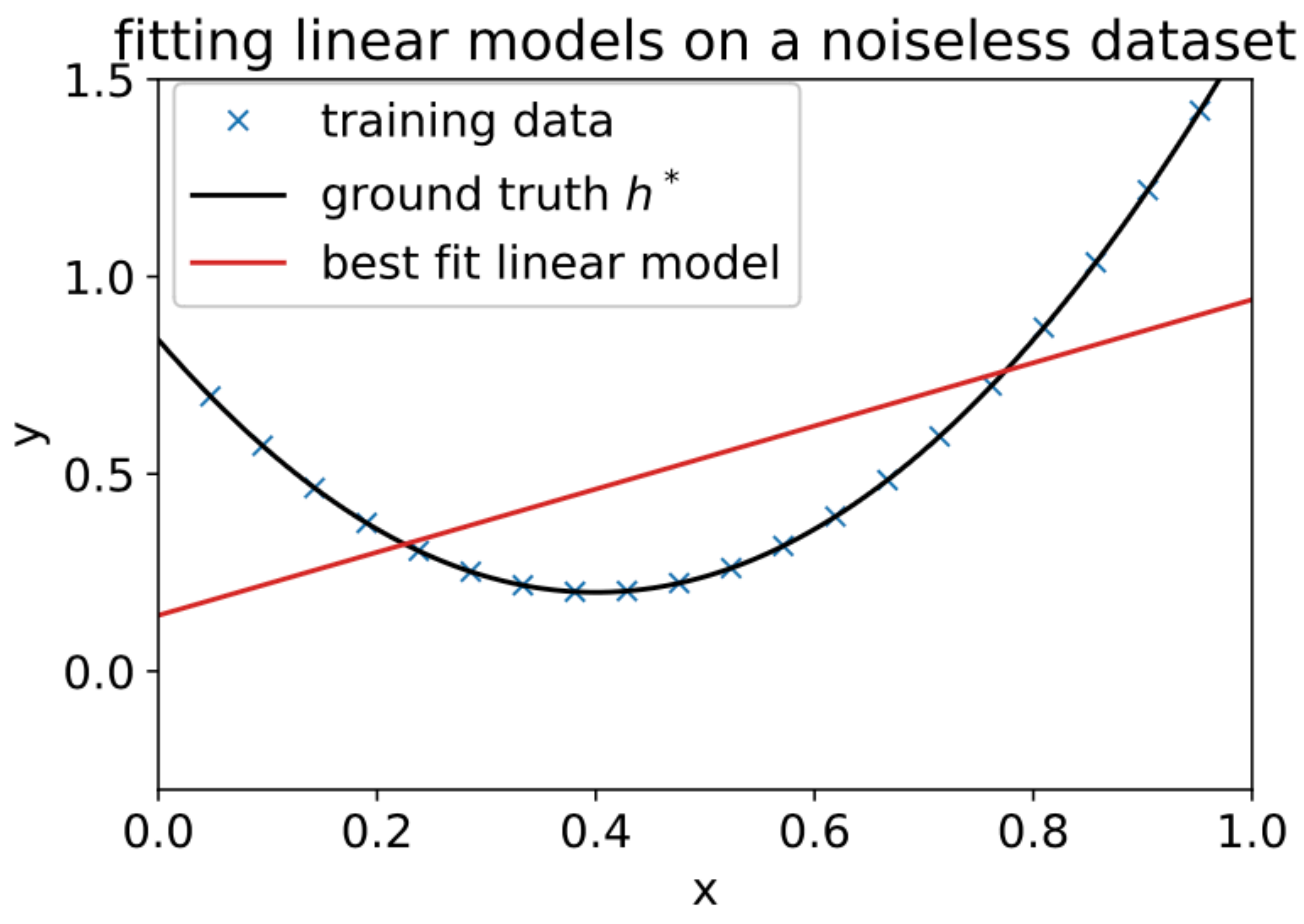
Error is still large when we have many training samples

Error is still large when we do not have noise

Inherent incapability of the linear model

Fitting a Linear Model





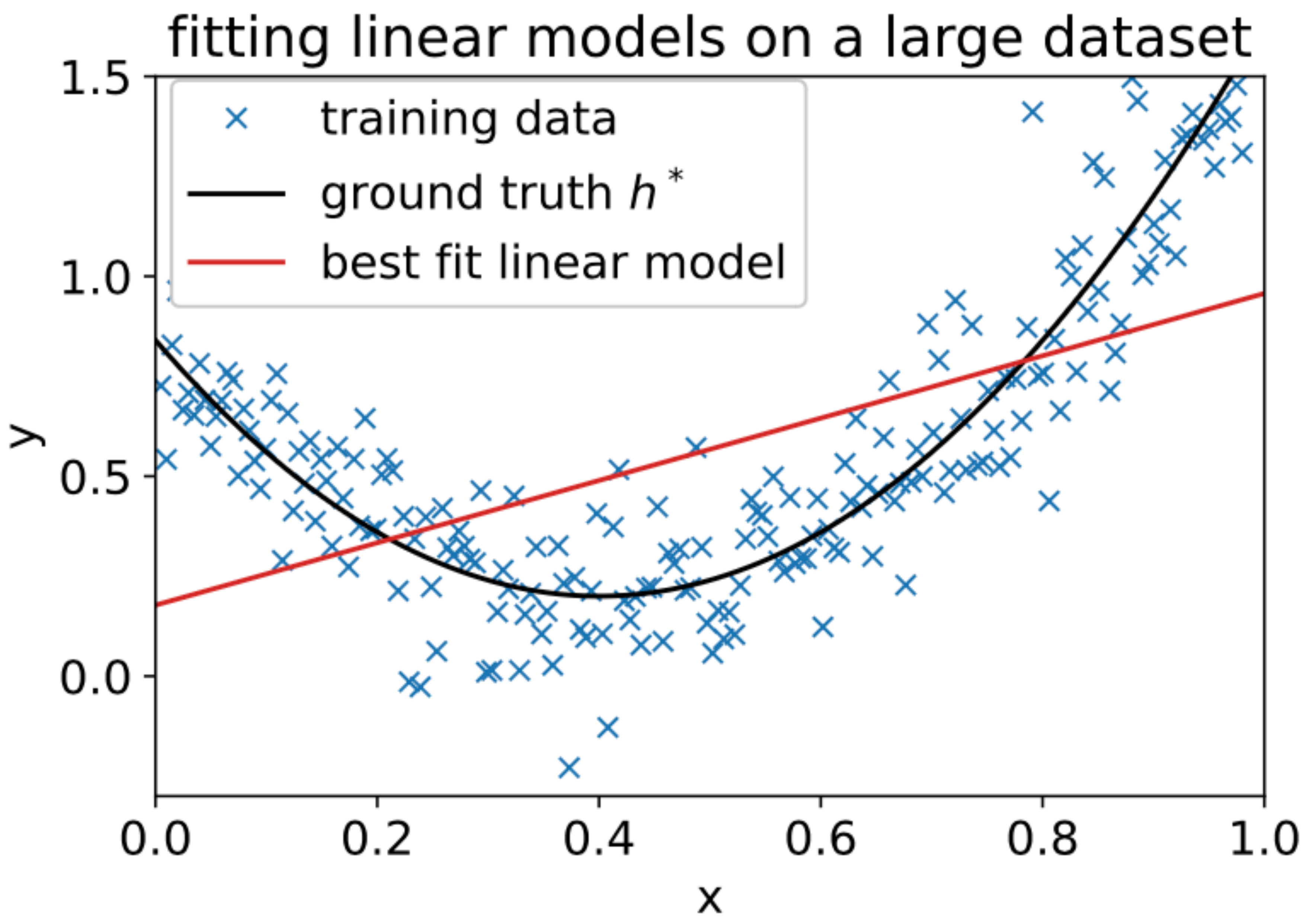
Error is still large when we have many training samples

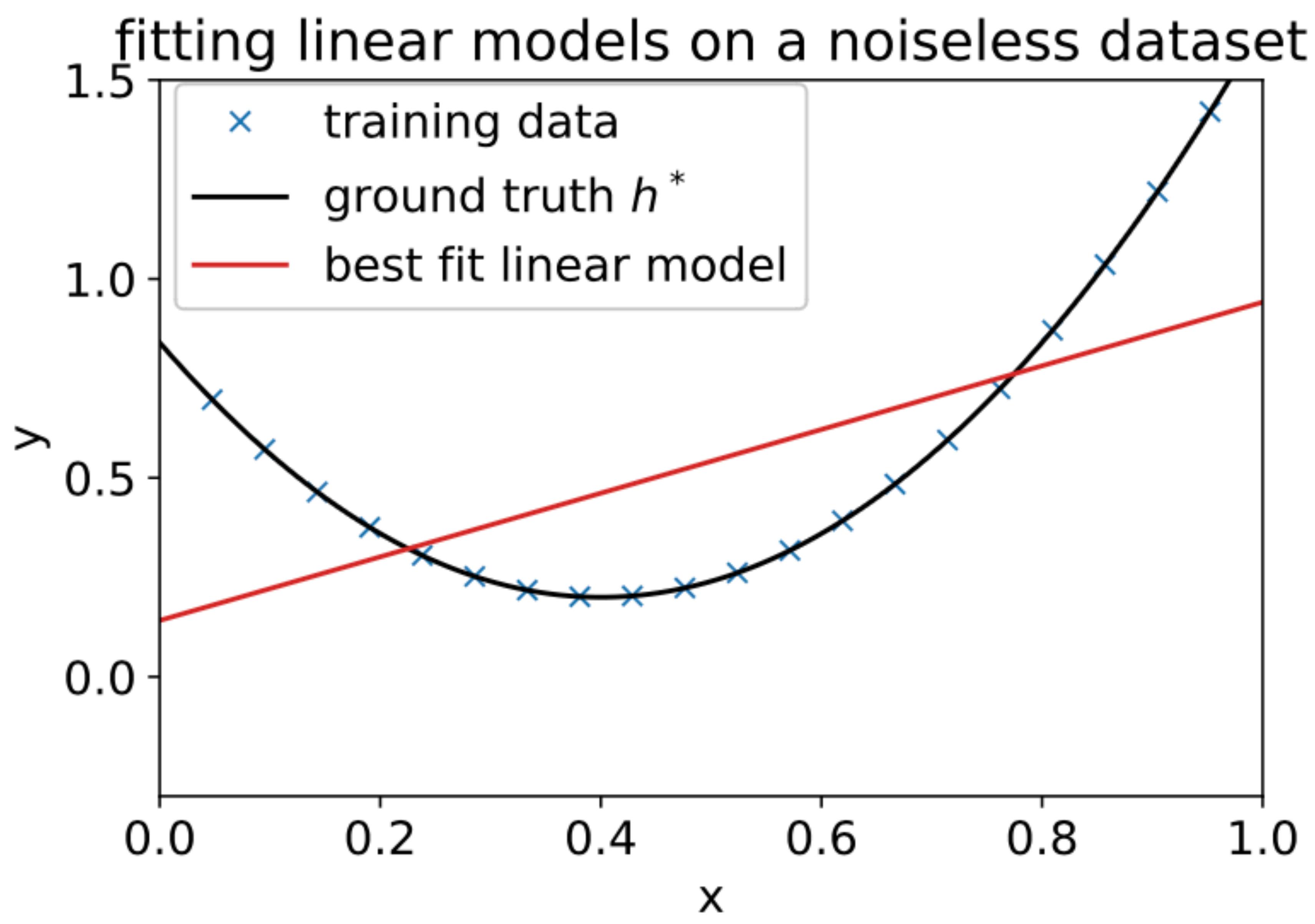
Error is still large when we do not have noise

Inherent incapability of the linear model

Bias of a model: the test error even if we were to fit to a very large training dataset

Fitting a Linear Model

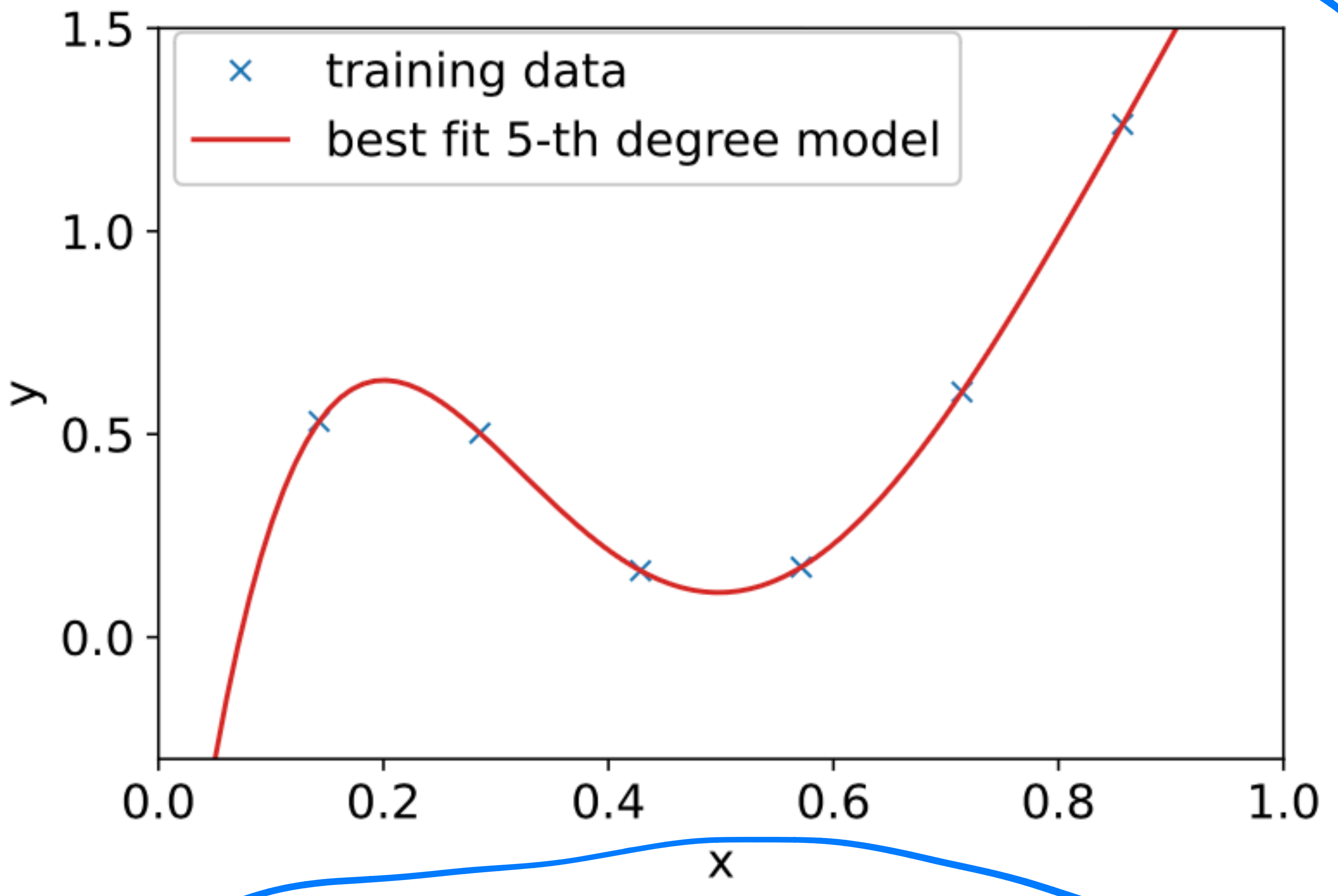




Training error is large → underfitting

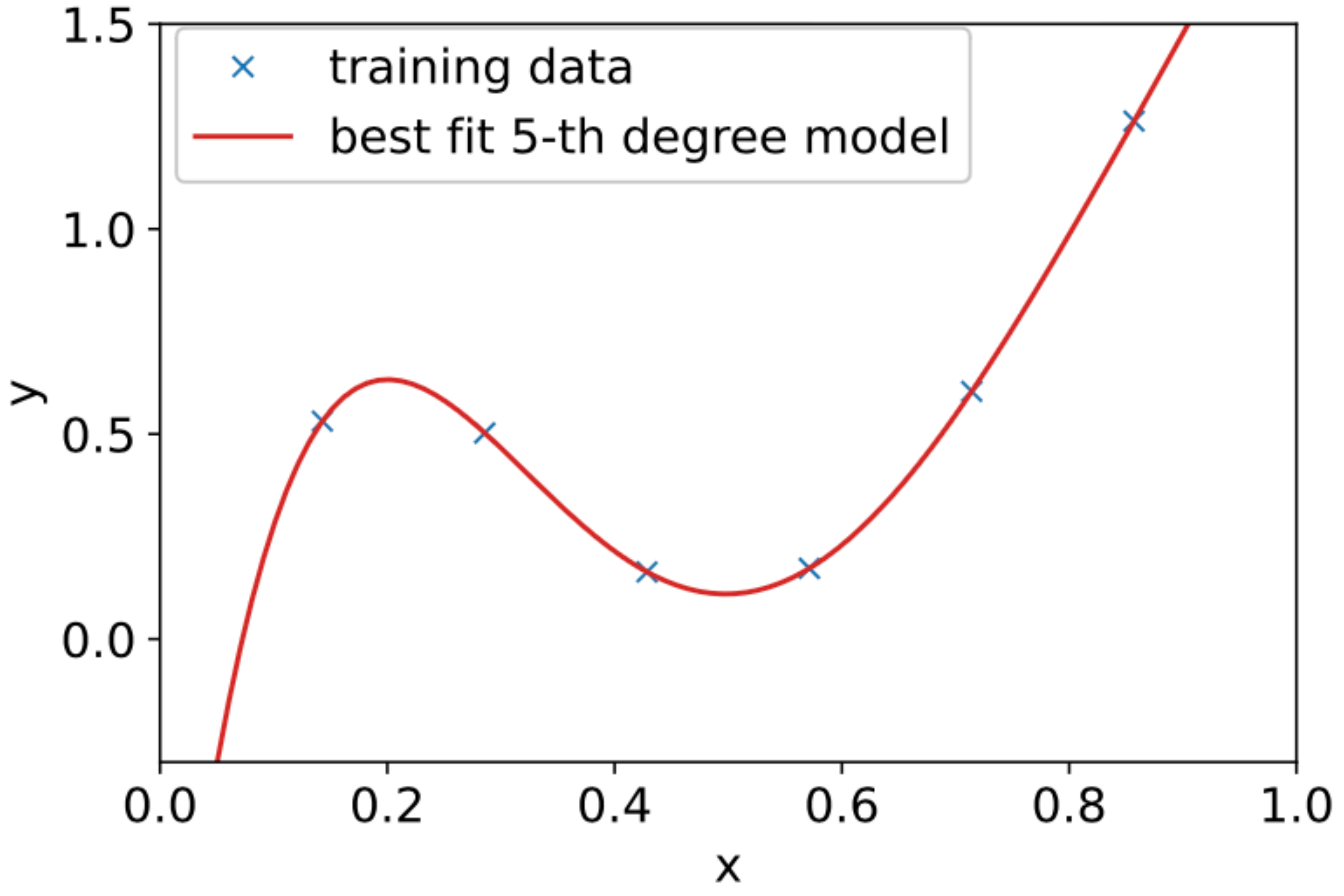
Fitting 5-th Degree Polynomials

Fitting 5-th Degree Polynomials

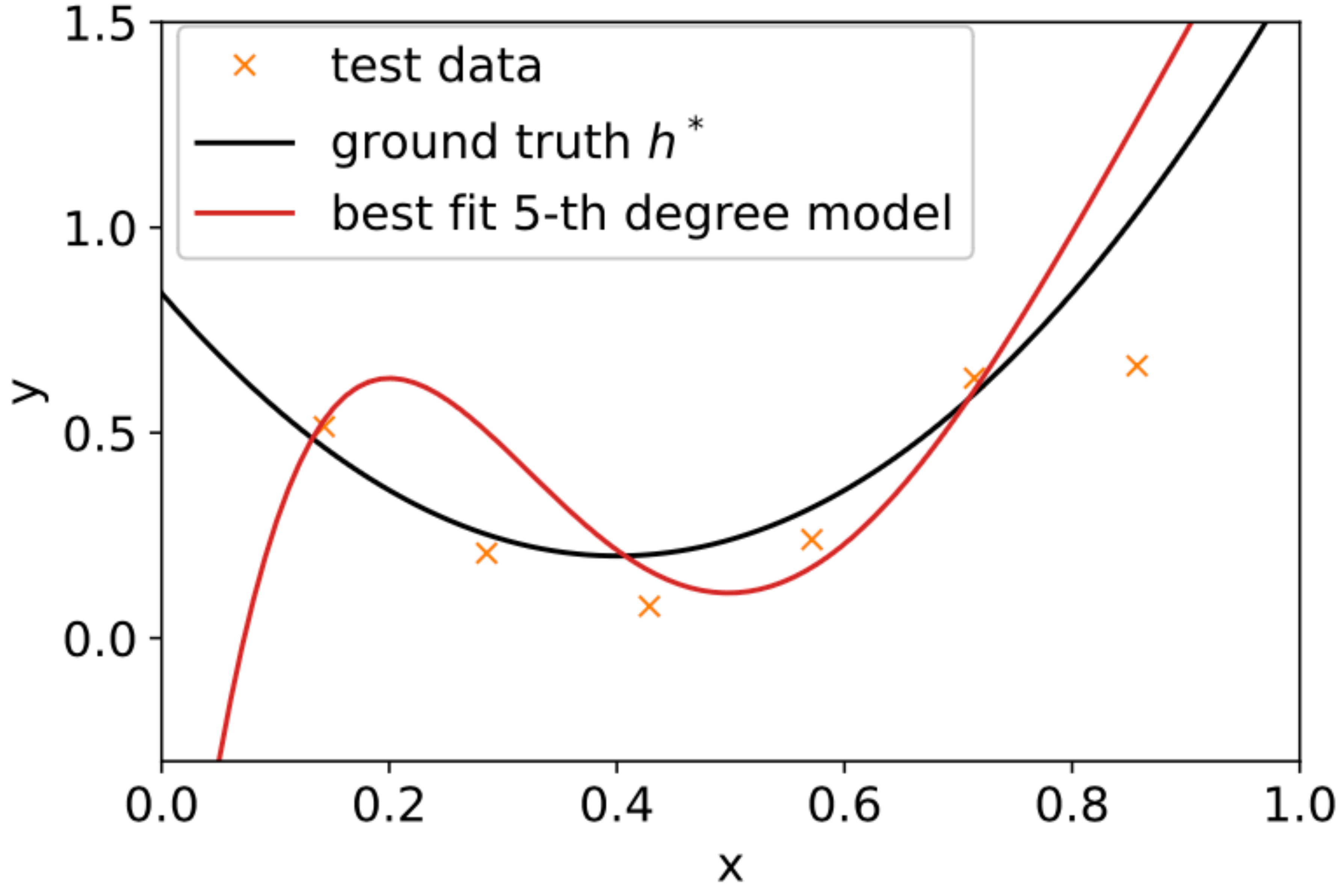


Zero training error

Fitting 5-th Degree Polynomials

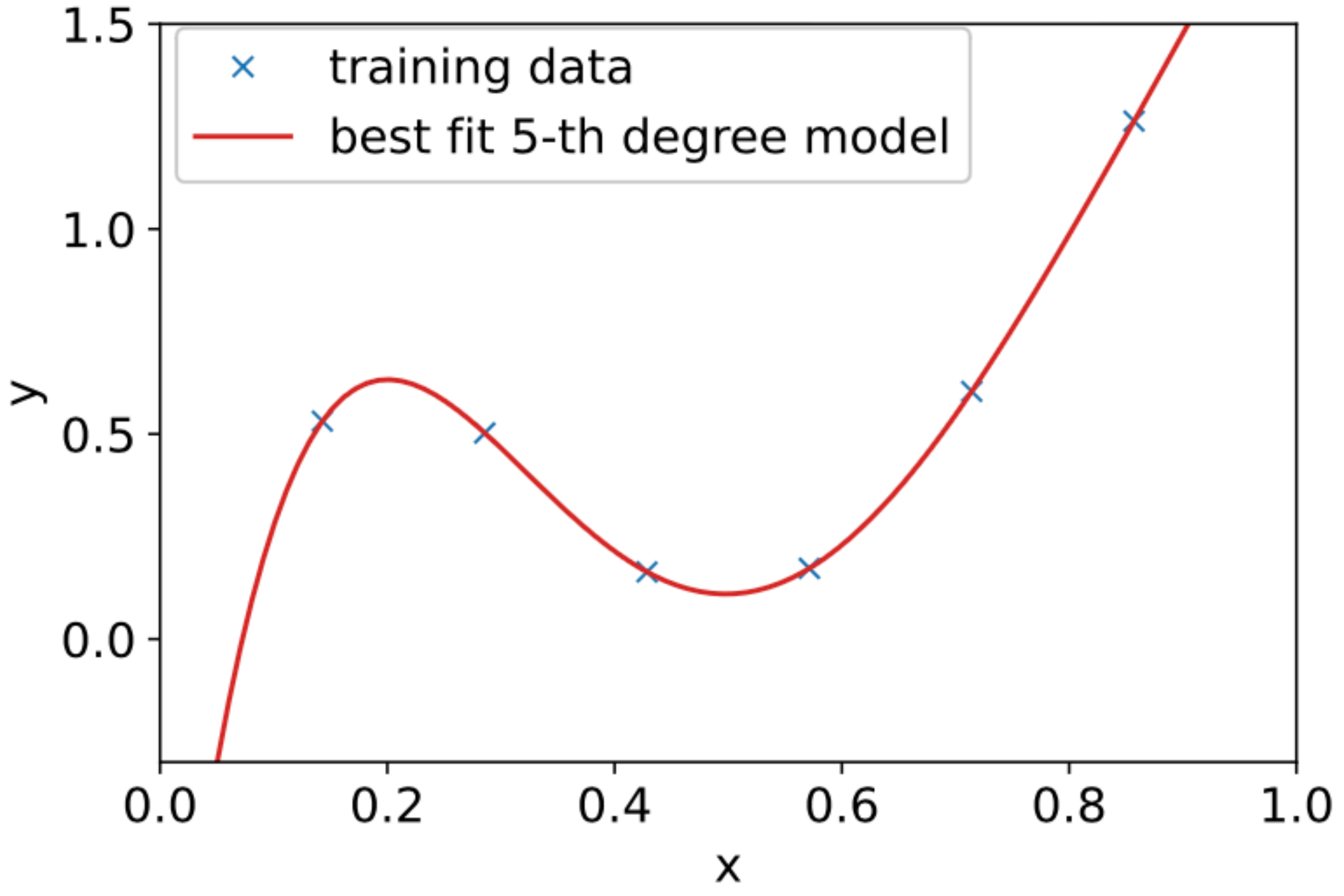


Zero training error

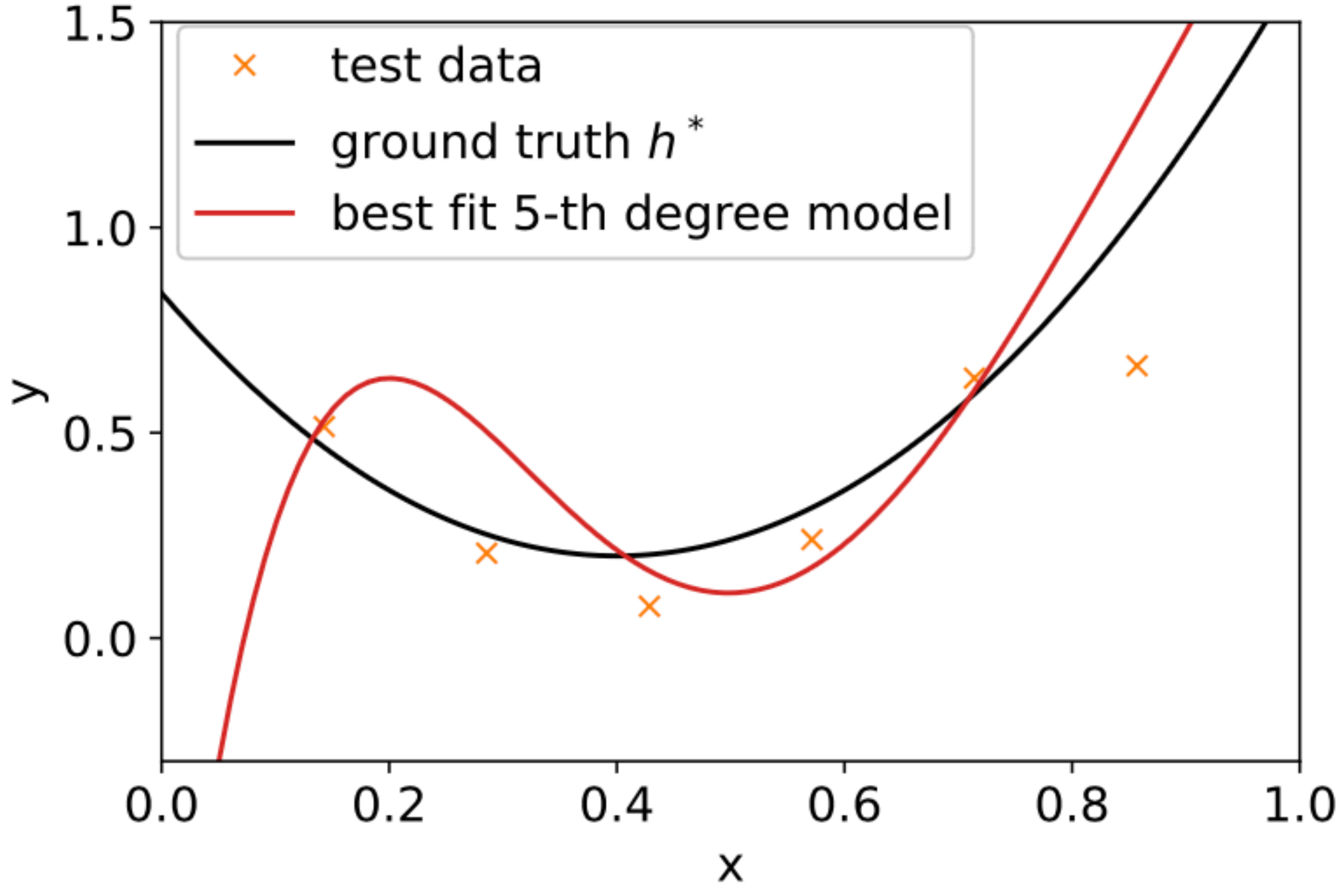


Large test error

Fitting 5-th Degree Polynomials



Zero training error

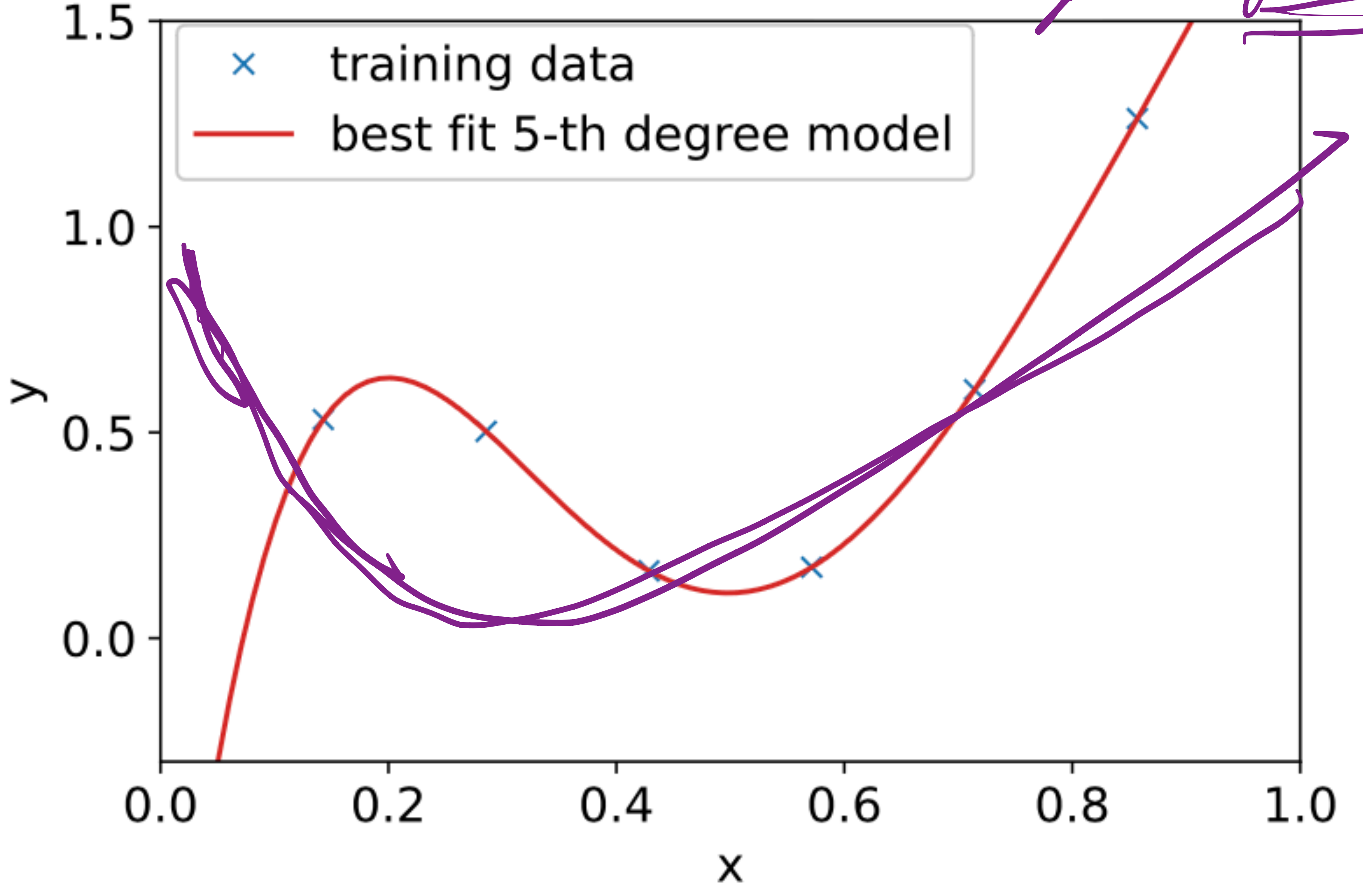


Large test error

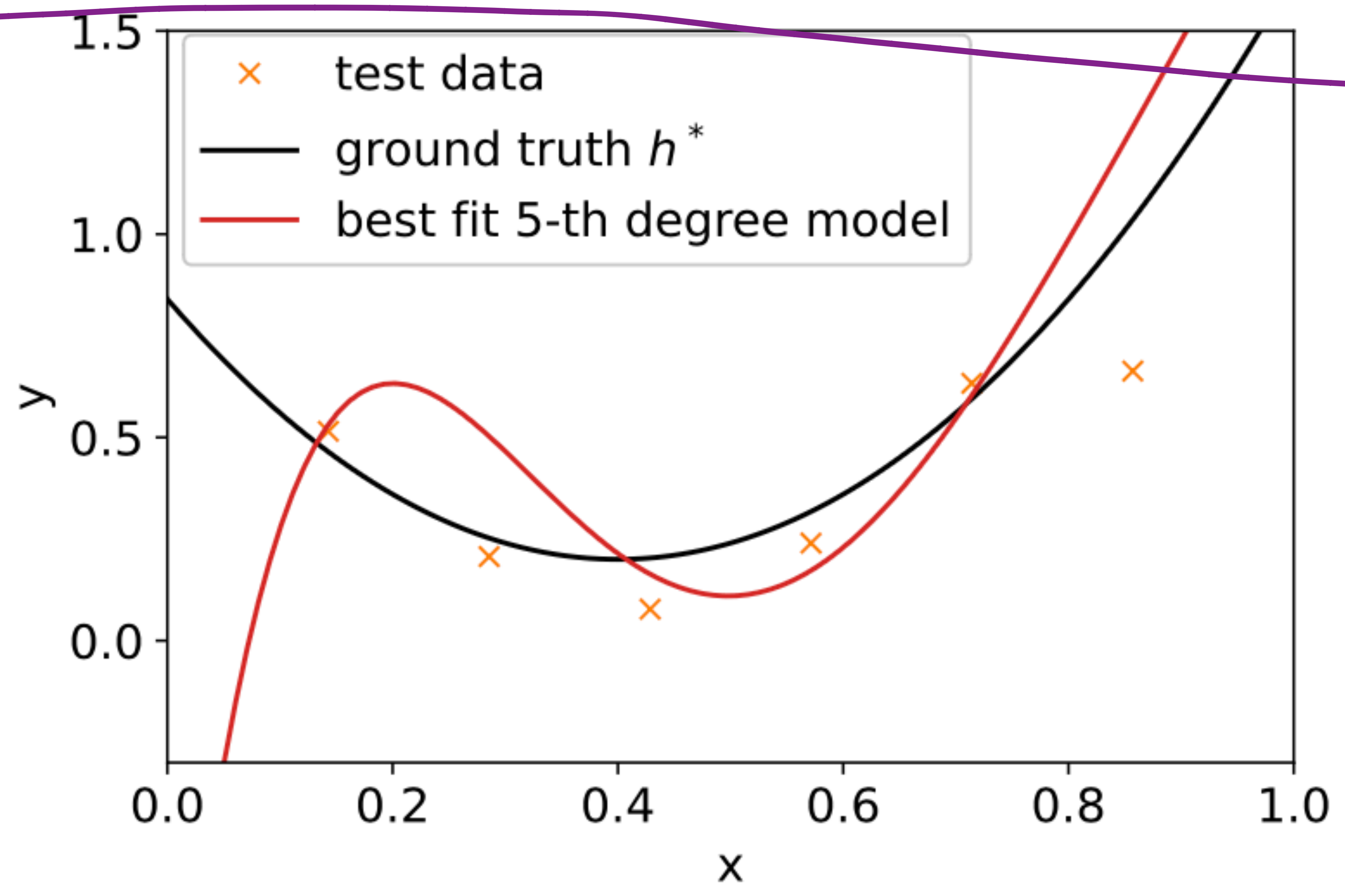
Training error is small, test error is large — the model does not *generalize*

Fitting 5-th Degree Polynomials

if we have infinite training samples ?



Zero training error



Large test error

Training error is small, test error is large — the model does not *generalize*

The model captures **spurious** features

spurious features

email spam classification

training data

spam email: "computer", "computer", "computer"

"discount", "sale", "cheap", "price" — spam

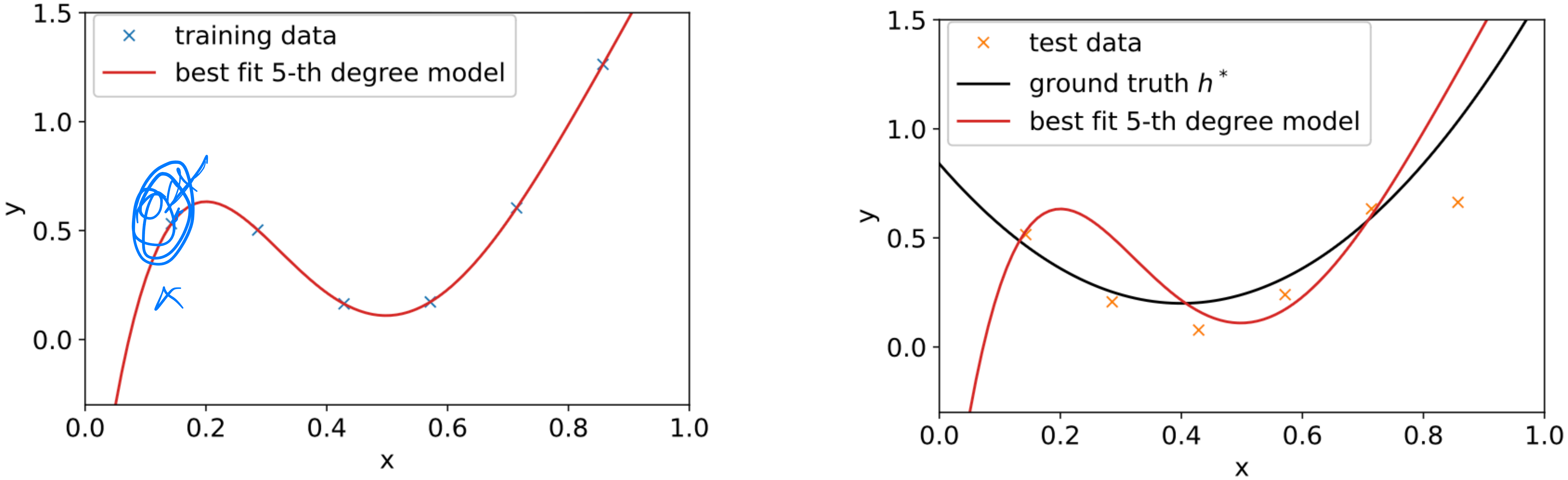
training data small

spurious feature

"computer" → spam

wrong

Fitting 5-th Degree Polynomials

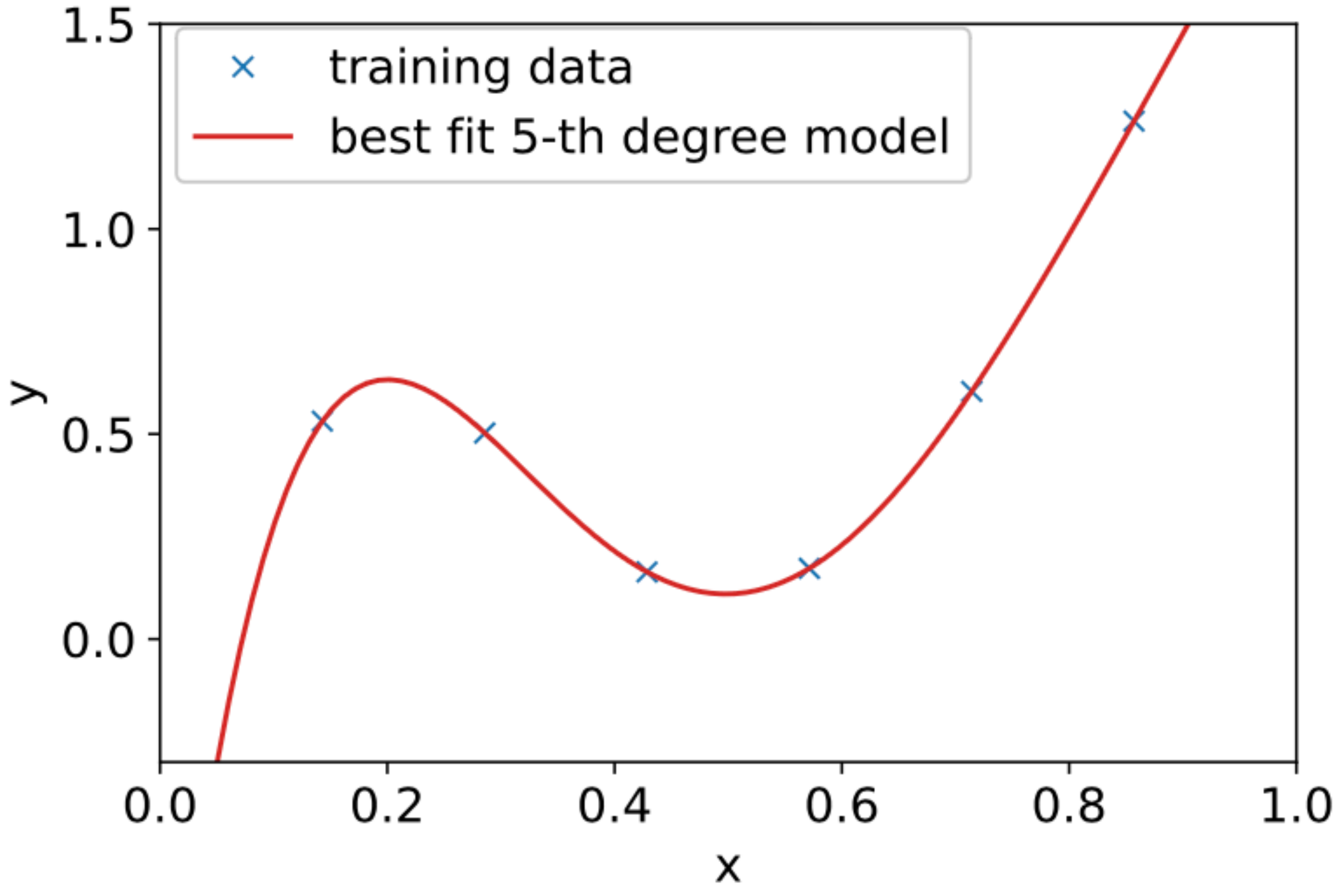


Zero training error

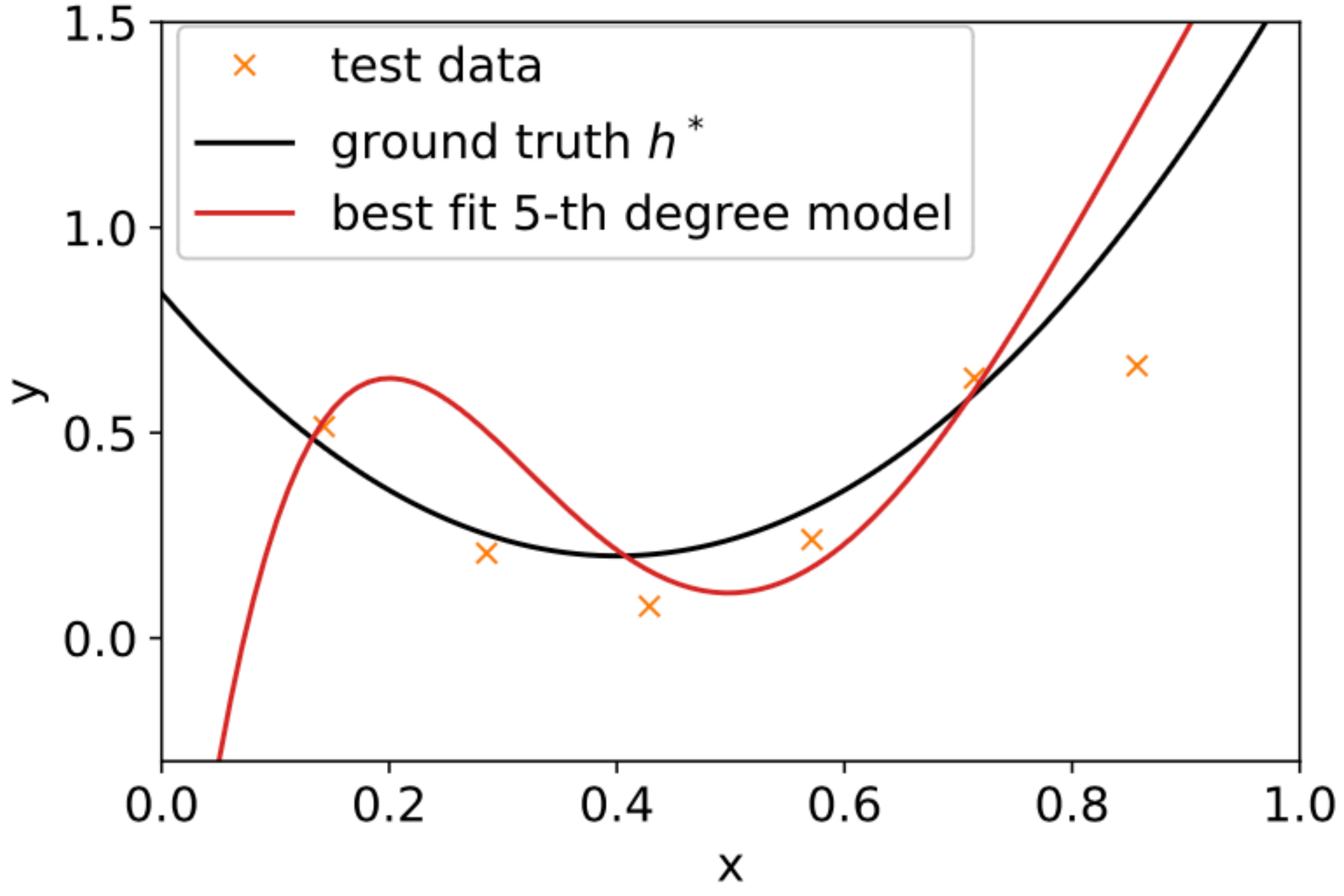
Large test error

A complex model is able to capture various patterns in the small, finite training dataset → large variance, small bias

Fitting 5-th Degree Polynomials



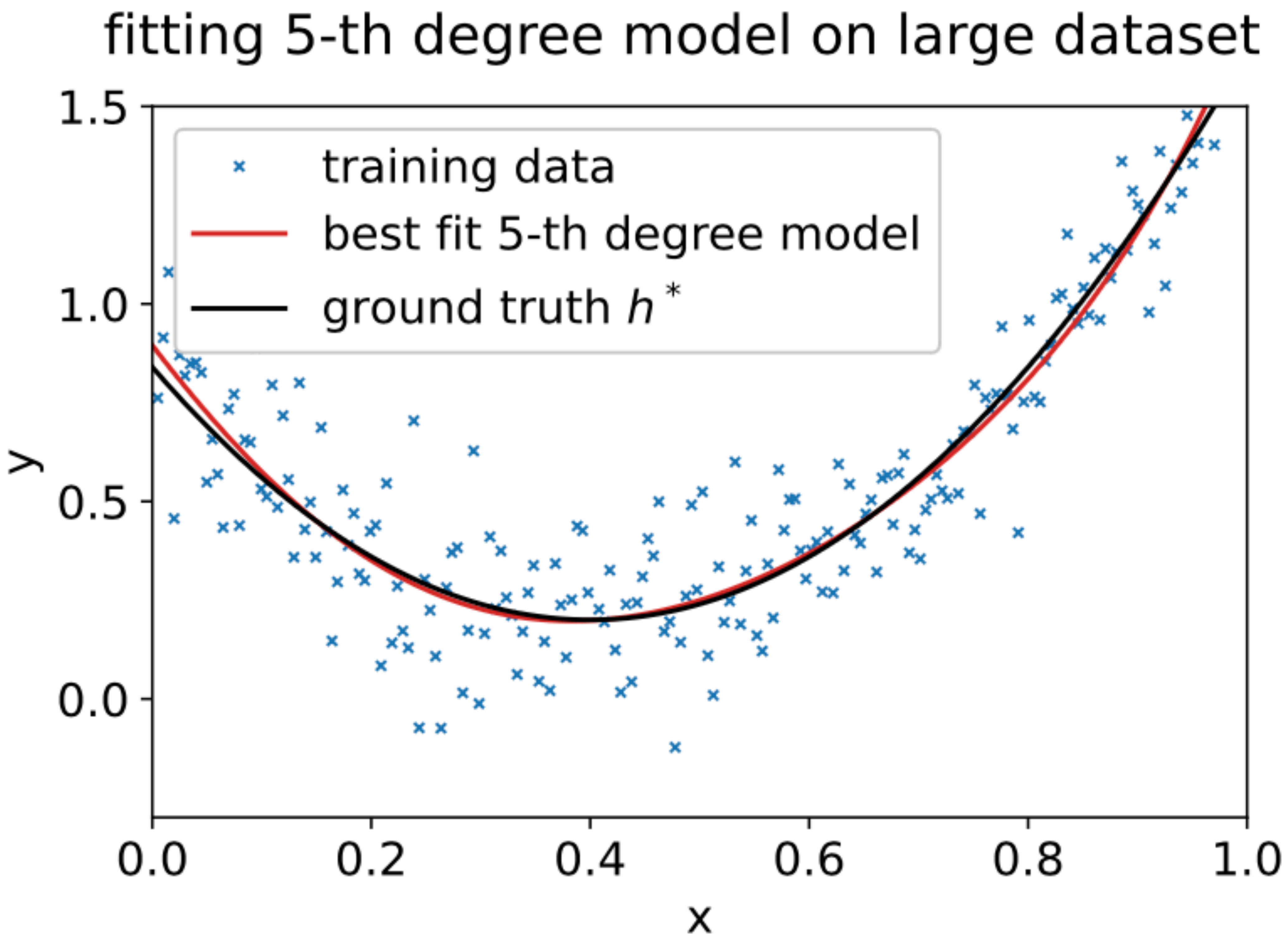
Zero training error



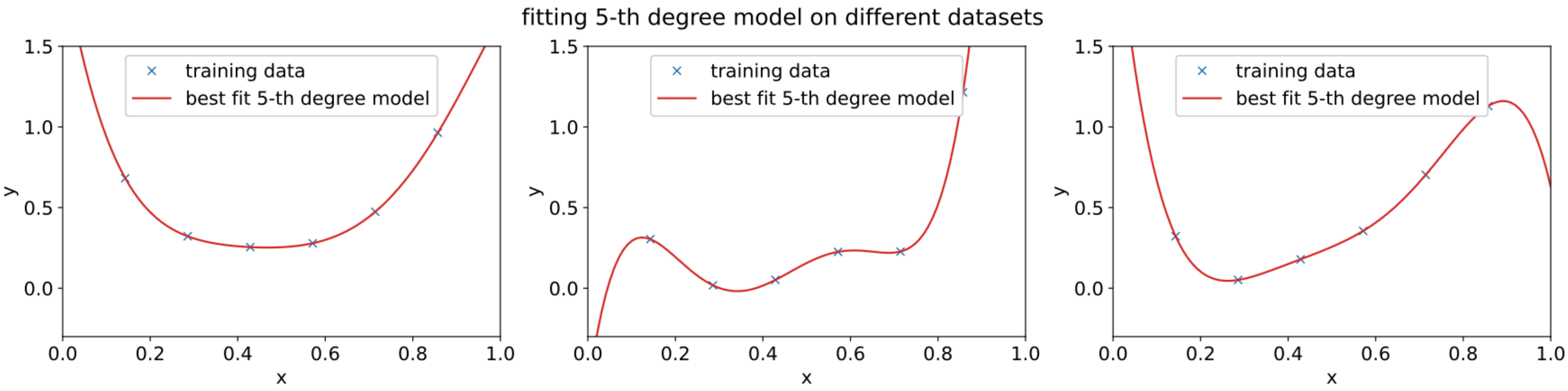
Large test error

What if we have enough training data?

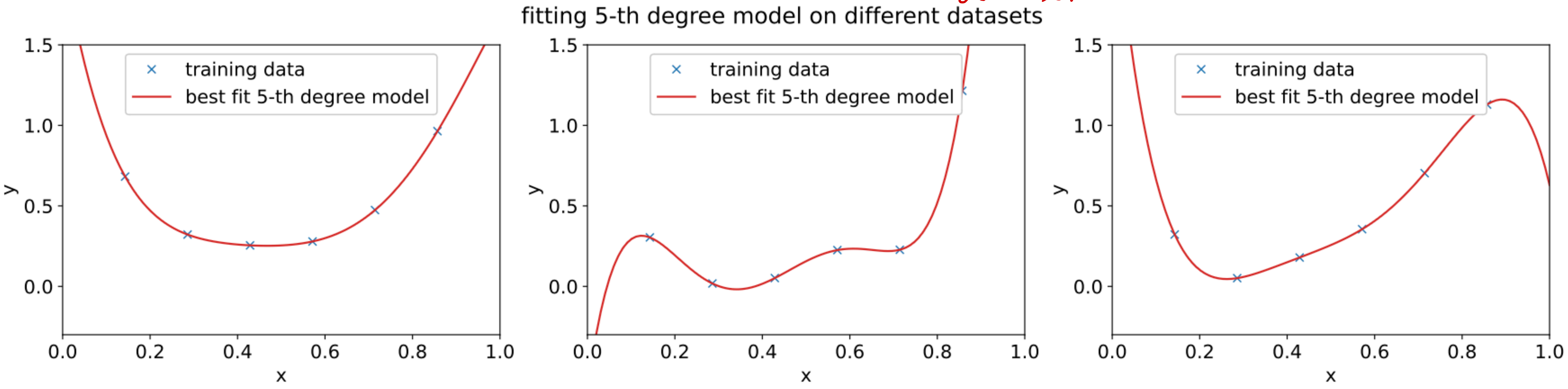
Fitting 5-th Degree Polynomials



Large Variance of 5-th Degree Model

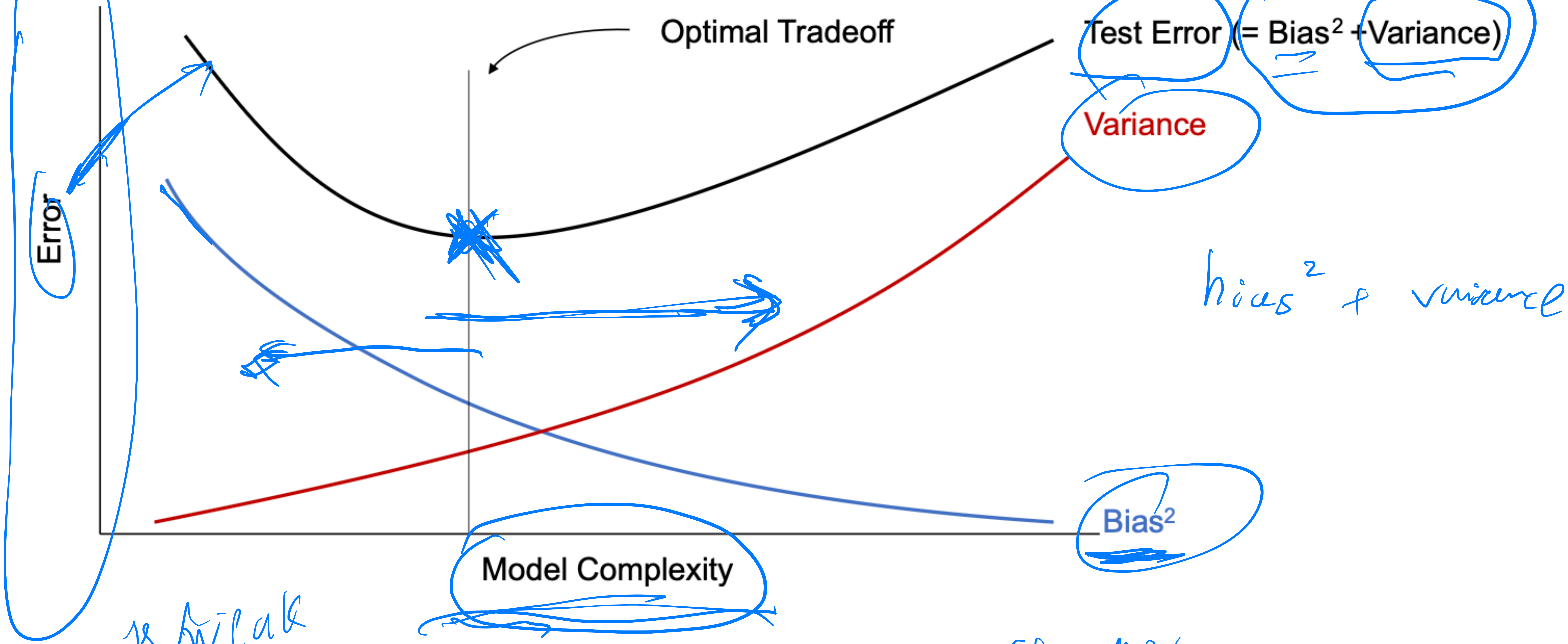


Large Variance of 5-th Degree Model

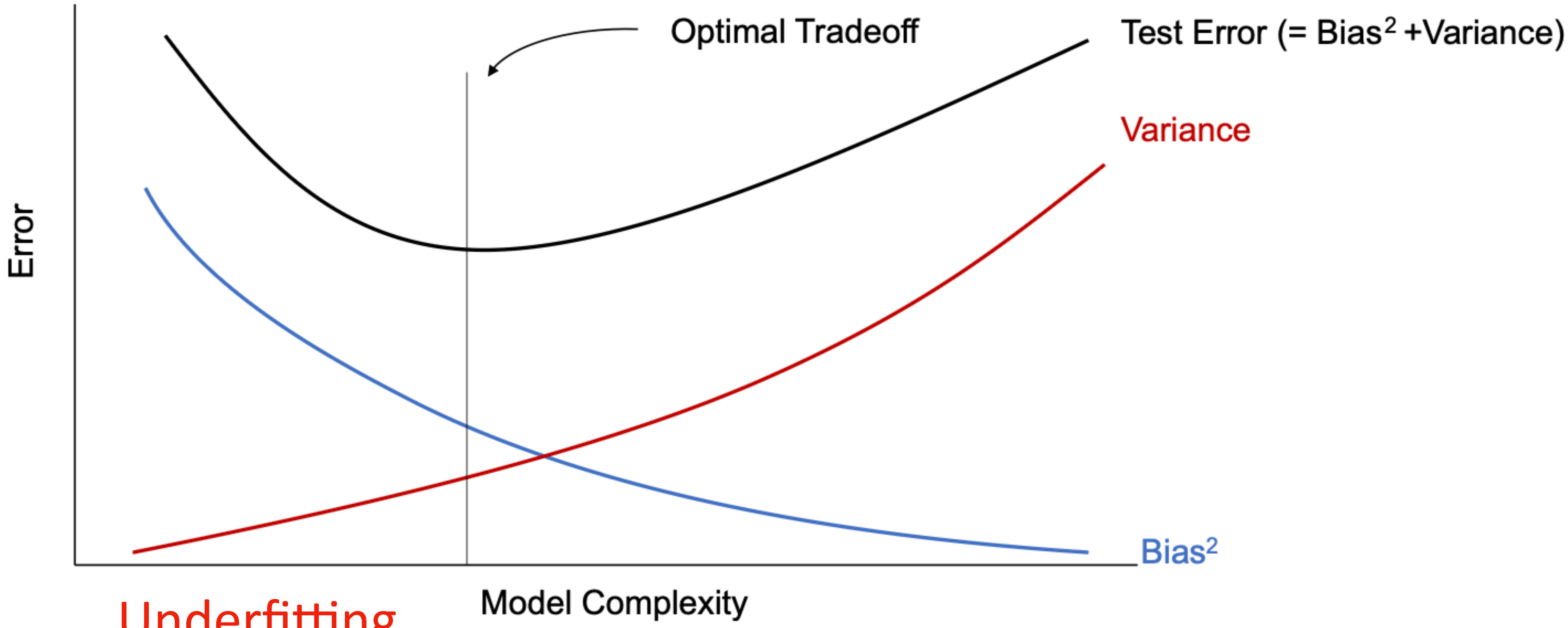


Intuitive Definition of the Variance: amount of variations across models learnt on multiple different training datasets (drawn from the same underlying distribution

Training vs. Test Error

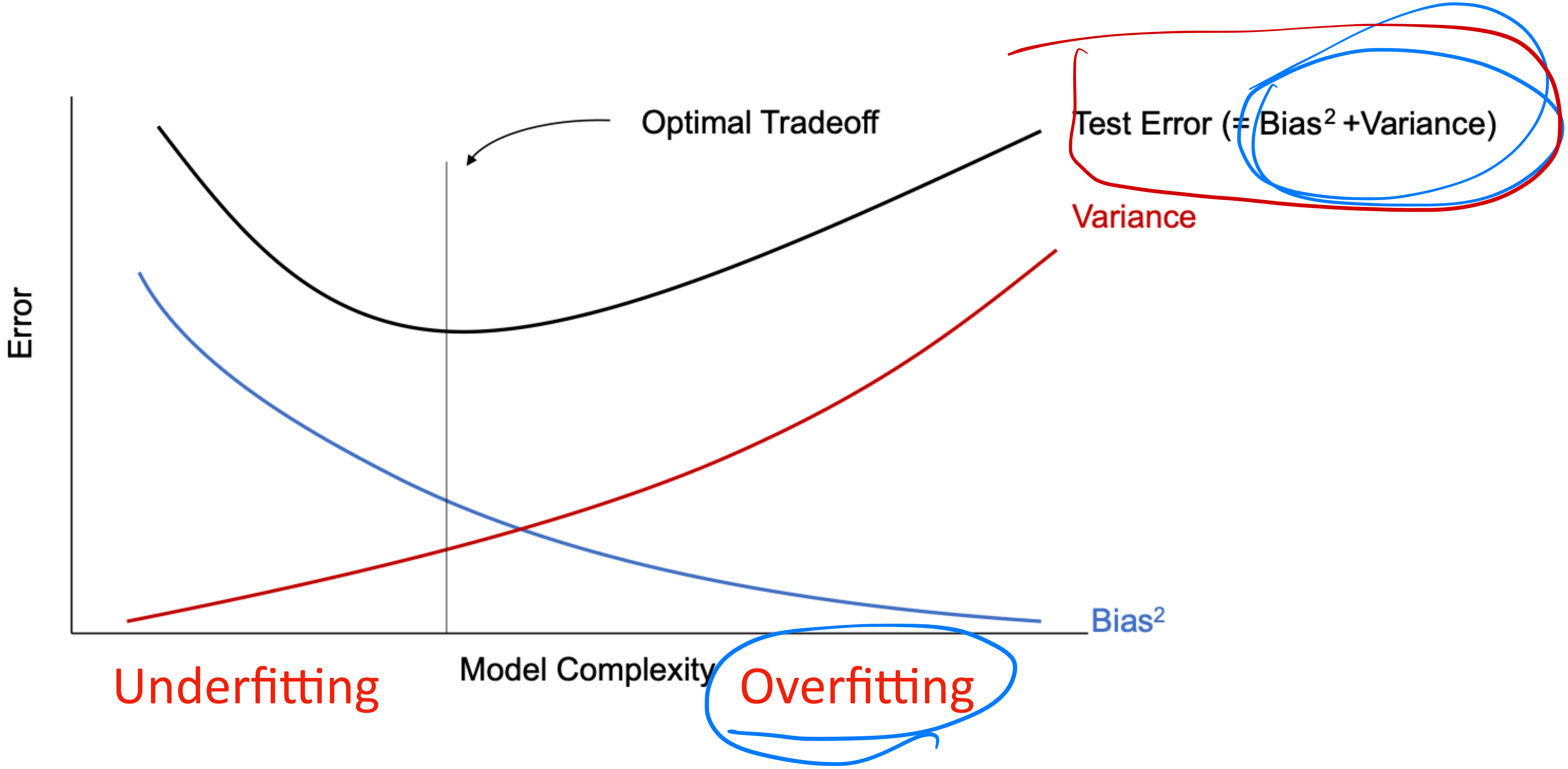


Training vs. Test Error



Underfitting

Training vs. Test Error



An Example of Bias-Variance Tradeoff in Regression

An Example of Bias-Variance Tradeoff in Regression

- Draw a training dataset $S = \{x^{(i)}, y^{(i)}\}_{i=1}^n$ such that $y^{(i)} = h^\star(x^{(i)}) + \xi^{(i)}$ where $\xi^{(i)} \in N(0, \sigma^2)$.

An Example of Bias-Variance Tradeoff in Regression

- Draw a training dataset $S = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ such that $y^{(i)} = h^\star(x^{(i)}) + \xi^{(i)}$ where $\xi^{(i)} \in N(0, \sigma^2)$.

- Train a model on the dataset S, denoted by $\hat{h}_S$.

An Example of Bias-Variance Tradeoff in Regression

- Draw a training dataset $S = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ such that $y^{(i)} = h^{\star}(x^{(i)}) + \xi^{(i)}$ where $\xi^{(i)} \in N(0, \sigma^2)$.

- Train a model on the dataset S, denoted by $\hat{h}_S$.

- Take a test example (x, y) such that $y = h^{\star}(x) + \xi$ where $\xi \sim N(0, \sigma^2)$

An Example of Bias-Variance Tradeoff in Regression

- Draw a training dataset $S = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ such that $y^{(i)} = h^\star(x^{(i)}) + \xi^{(i)}$ where $\xi^{(i)} \in N(0, \sigma^2)$.

- Train a model on the dataset S, denoted by $\hat{h}_S$.

- Take a test example (x, y) such that $y = h^\star(x) + \xi$ where $\xi \sim N(0, \sigma^2)$

$$\text{MSE}(x) = \mathbb{E}_{S,\xi}[(y - h_S(x))^2]$$

Mean square error on the test set

An Example of Bias-Variance Tradeoff in Regression

$$\text{MSE}(x) = \mathbb{E}_{S,\xi}[(y - h_S(x))^2]$$

$$\text{MSE}(x) = \underbrace{\sigma^2}_{\text{unavoidable}} + \underbrace{(h^\star(x) - h_{\text{avg}}(x))^2}_{\triangleq \text{ bias}^2} + \underbrace{\text{var}(h_S(x))}_{\triangleq \text{ variance}}$$

$h_{avg}(x) = E_S[h_S(x)]$

$$MSE = E_{S,\varepsilon}[(y - h_S(x))^2]$$

$$y = h^*(x) + \varepsilon$$

$$MSE = E_{S,\varepsilon}[(h^*(x) + \varepsilon - h_S(x))^2]$$

$$= E_{S,\varepsilon}[(h^*(x) - h_S(x))^2 + 2\varepsilon(h^*(x) - h_S(x)) + \varepsilon^2]$$

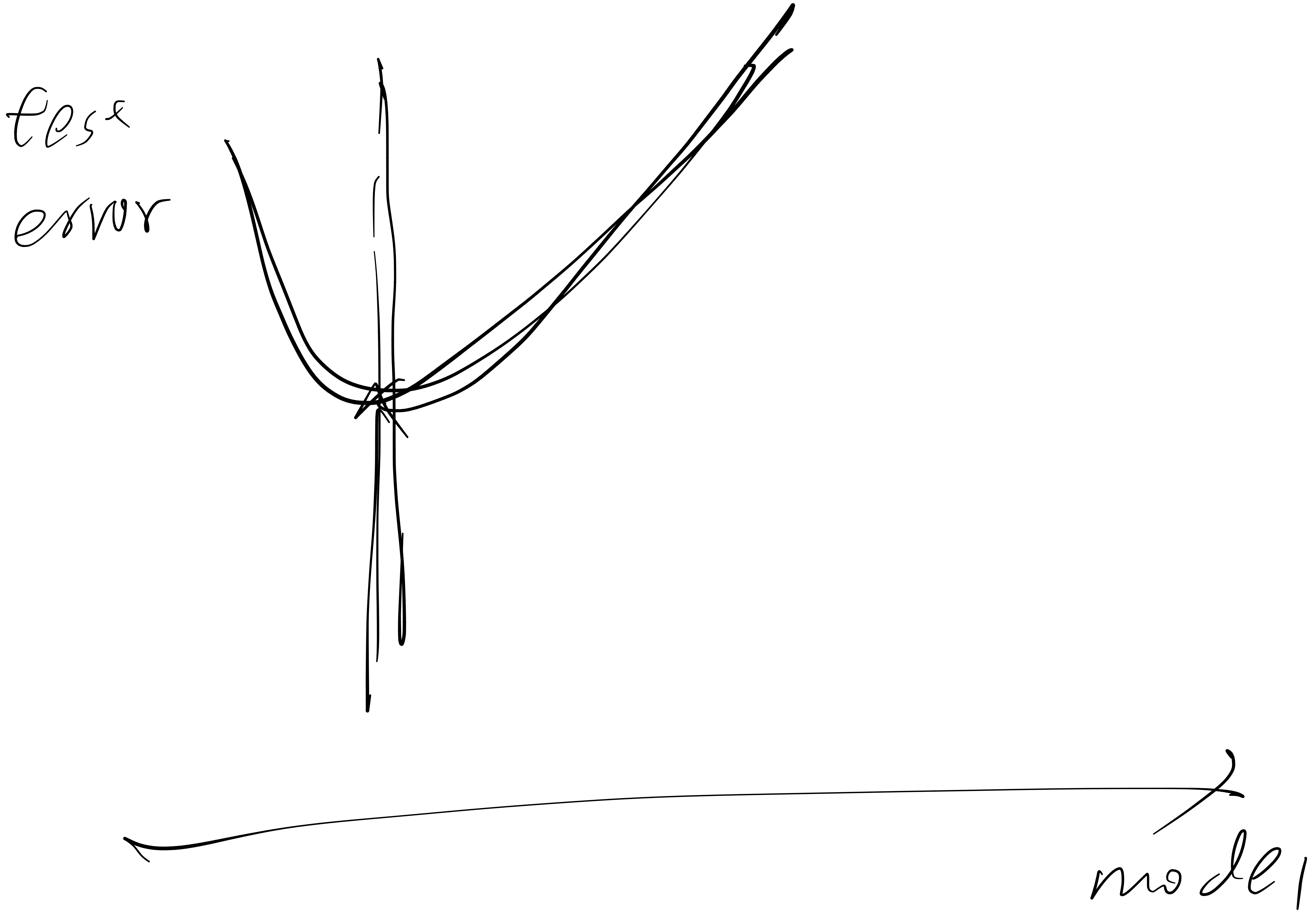
0 ; $\varepsilon^2 \to \sigma^2$; σ^2

$$E(\varepsilon) \cdot E[h^*(x) - h_S(x)]$$

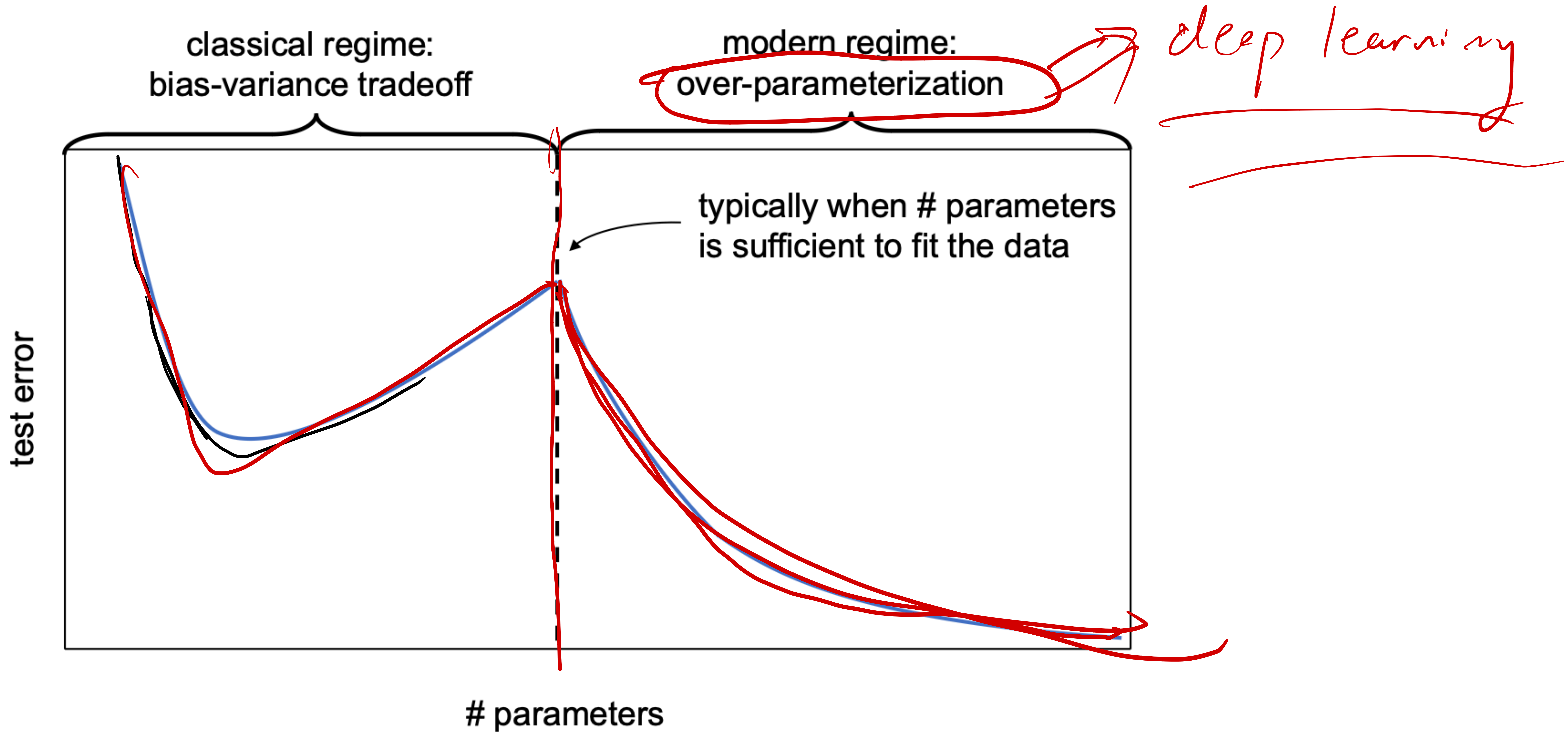
$E(\varepsilon) = 0$; $= 0$

$$= \sigma^2 + E_{S,\varepsilon}[(h^*(x) - h_S(x))^2]$$

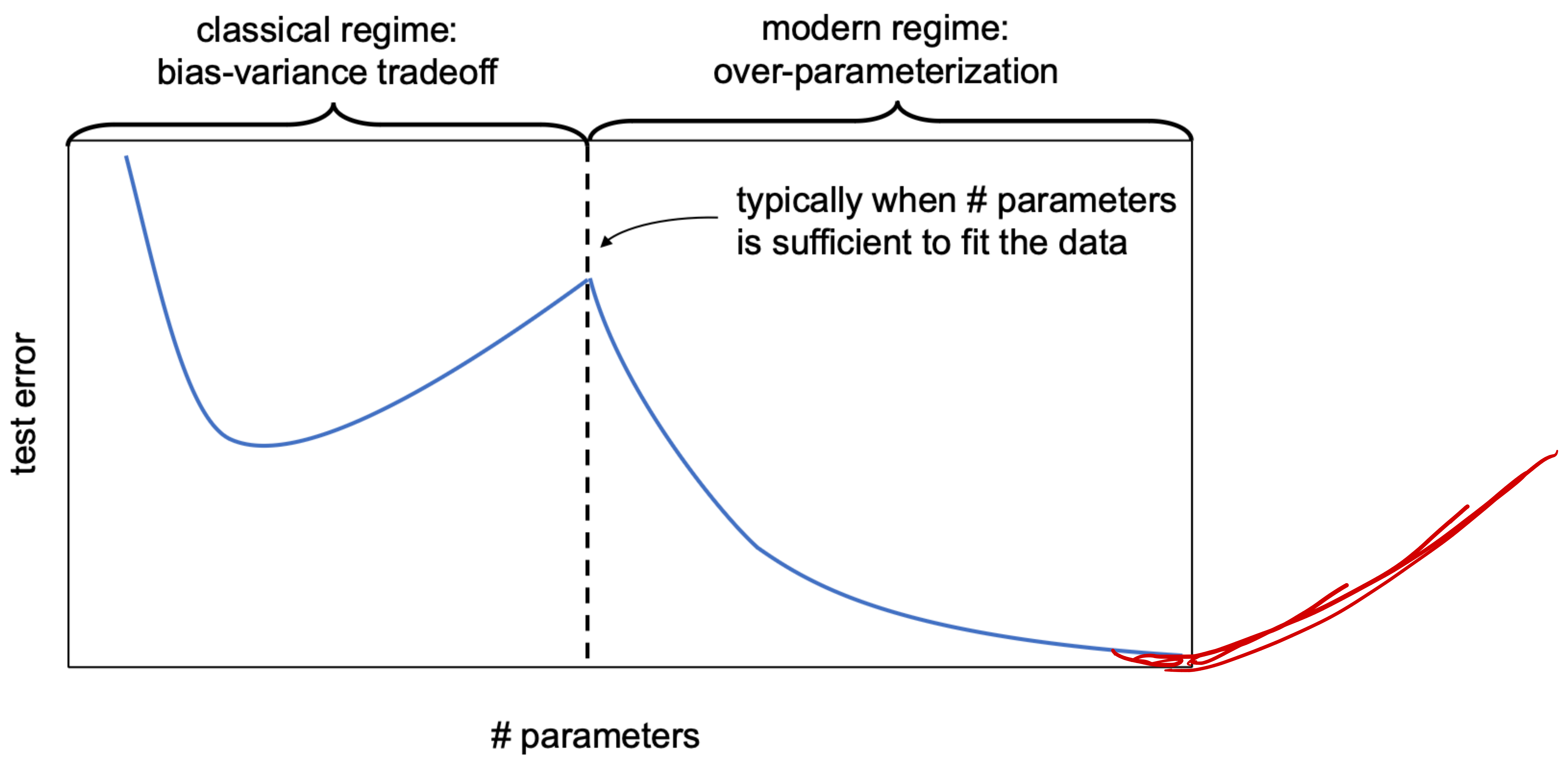
The Double-Descent Phenomenon



The Double-Descent Phenomenon

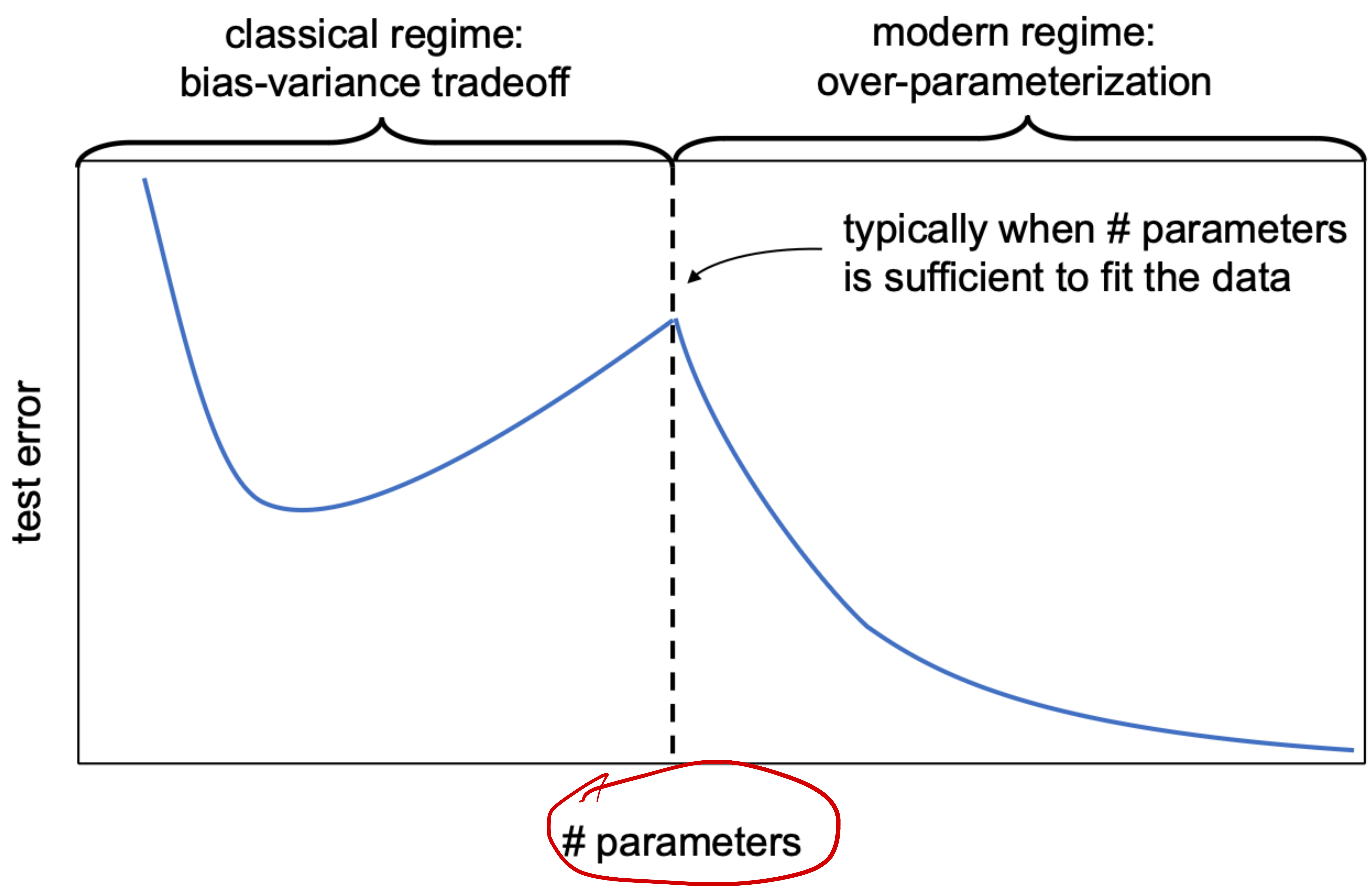


The Double-Descent Phenomenon



Overparameterization is very successful in deep learning, but is still mysterious

The Double-Descent Phenomenon

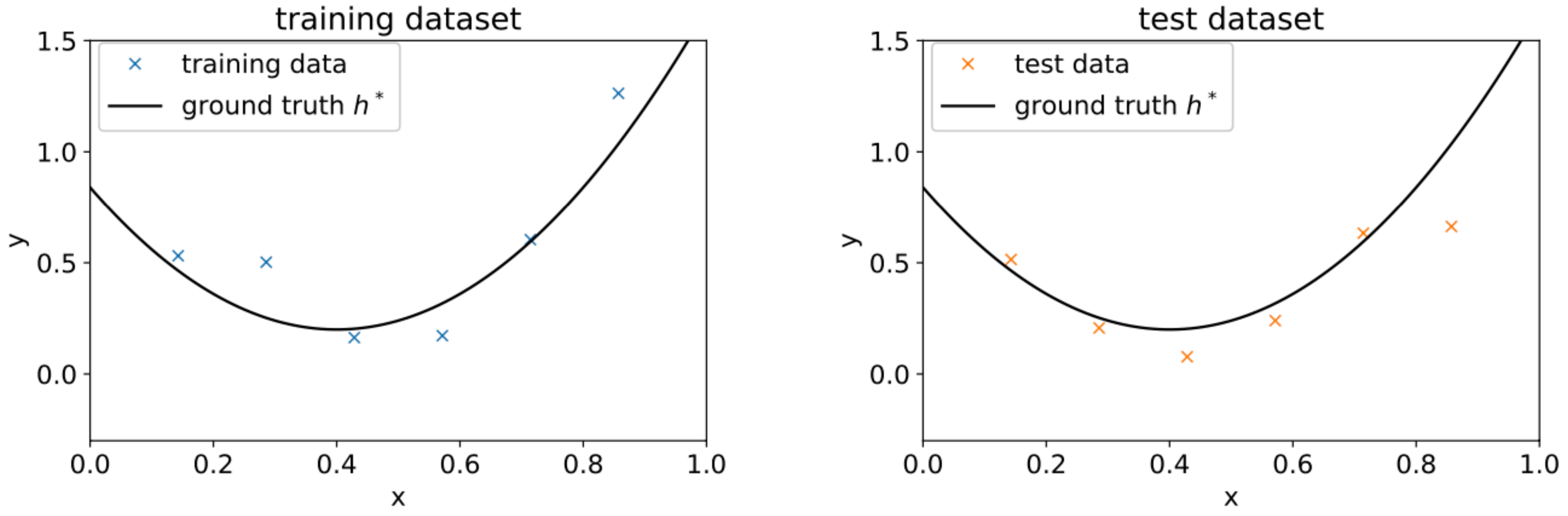


1000 parameters → attention < 500 → feedforward

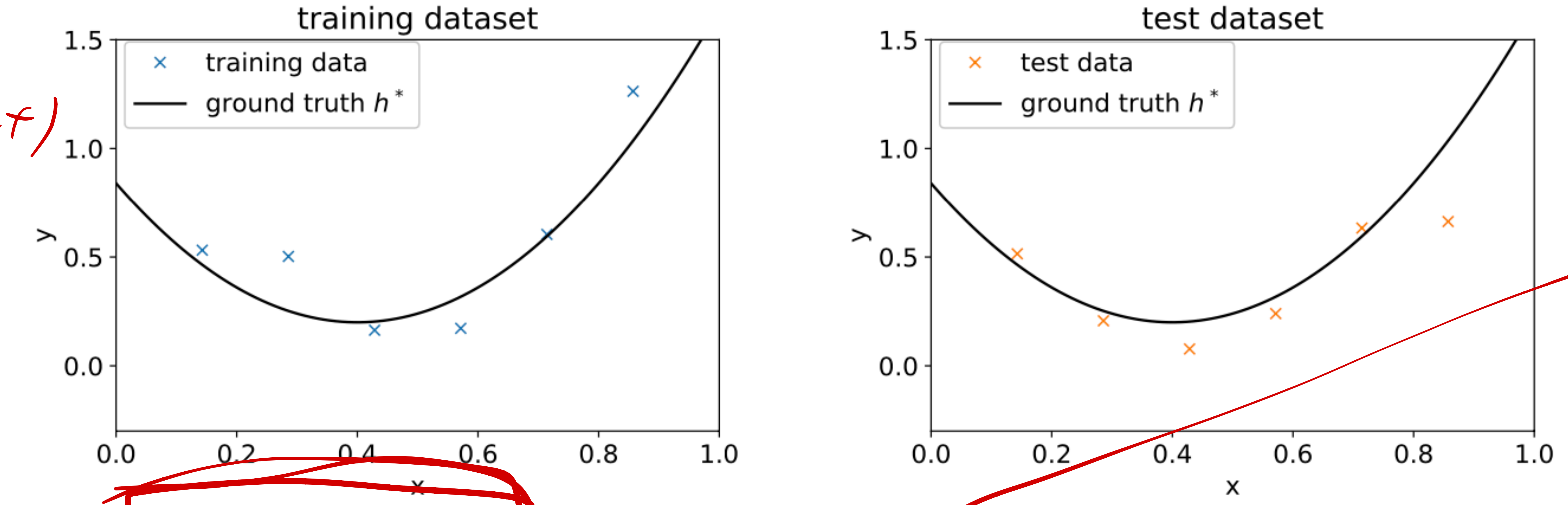
This figure uses # parameters to represent model complexity, is this the best measure?

Overparameterization is very successful in deep learning, but is still mysterious

Revisit the Train-Test Mismatch

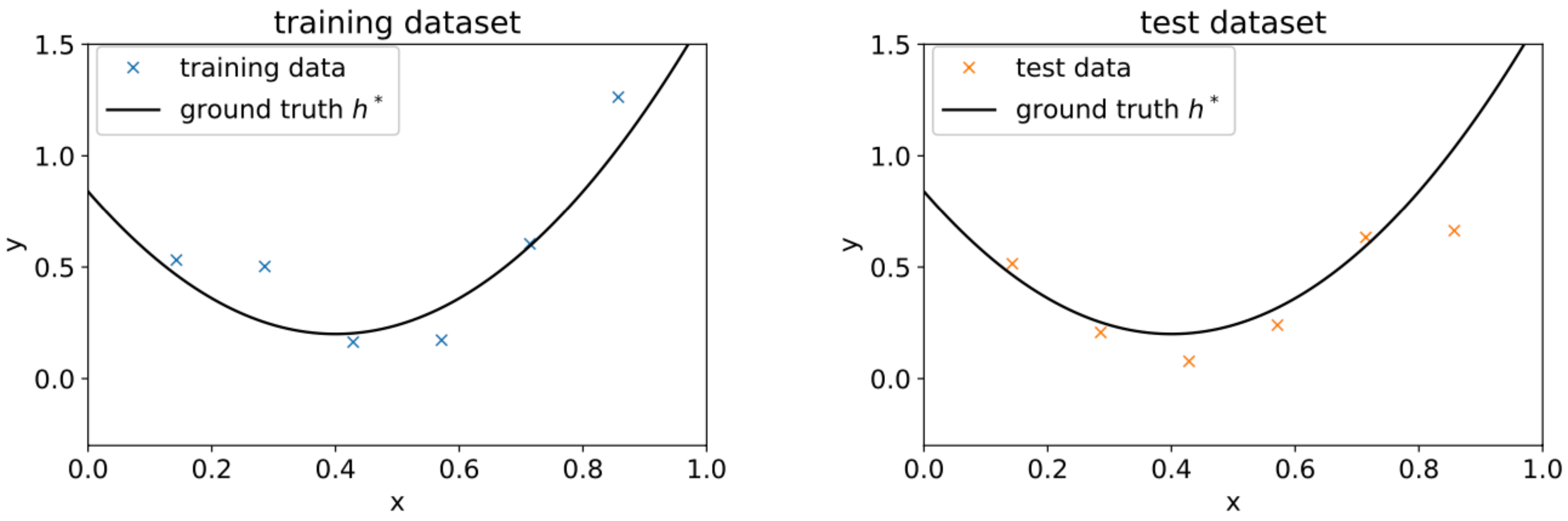


Revisit the Train-Test Mismatch



- The training / test empirical distributions are different with finite samples, even though their ground-truth distributions are the same

Revisit the Train-Test Mismatch



- The training / test empirical distributions are different with finite samples, even though their ground-truth distributions are the same
- In practice, the ground-truth distributions may be different

$D_{train} \sim P_{data}(x)$

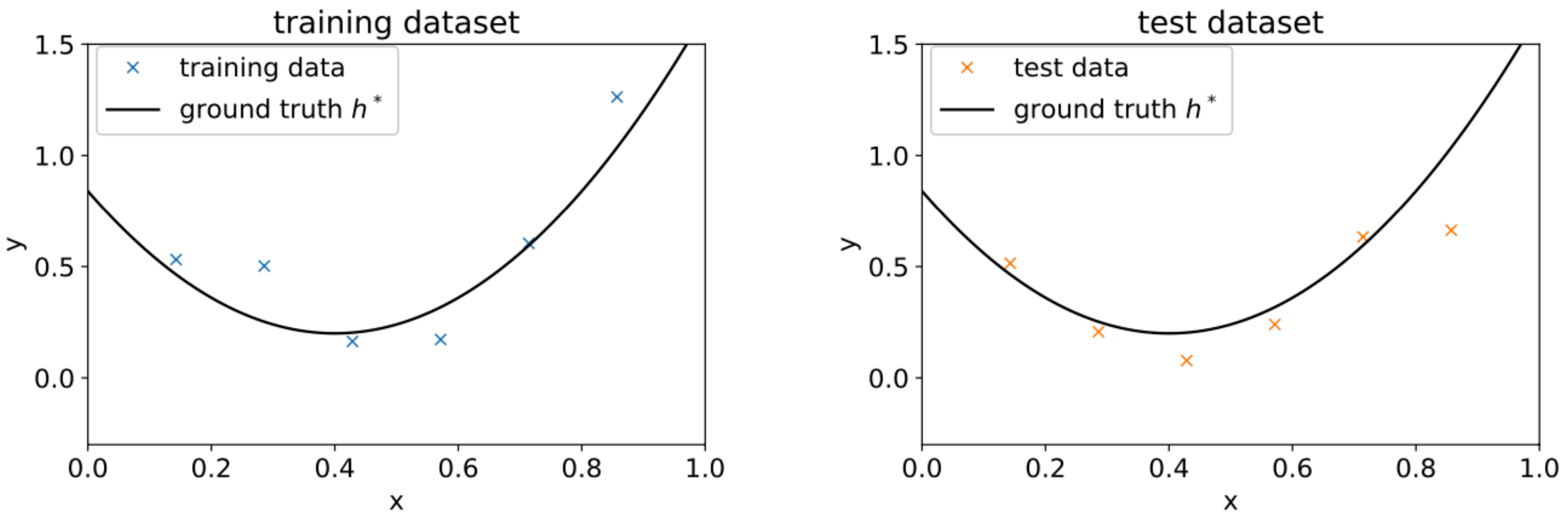
$D_{test} \sim P'_{data}(x)$

email spam classification

training: IT topic

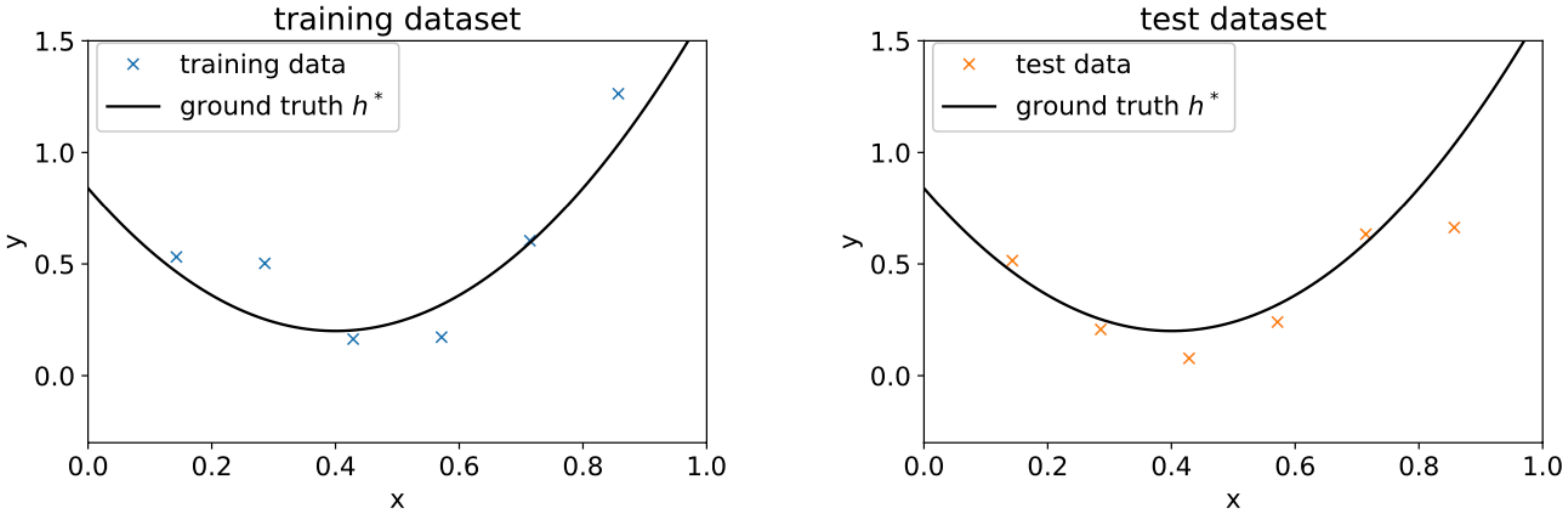
test: entertainment topic

Revisit the Train-Test Mismatch



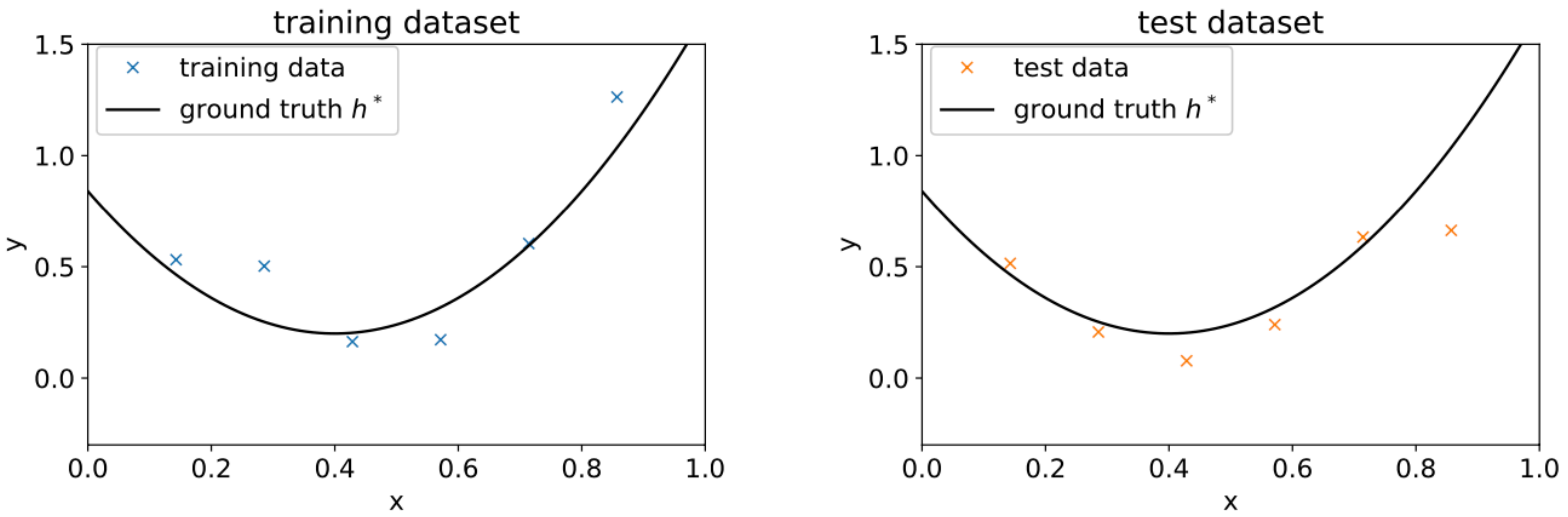
- The training / test empirical distributions are different with finite samples, even though their ground-truth distributions are the same
- In practice, the ground-truth distributions may be different Transfer Learning

Revisit the Train-Test Mismatch



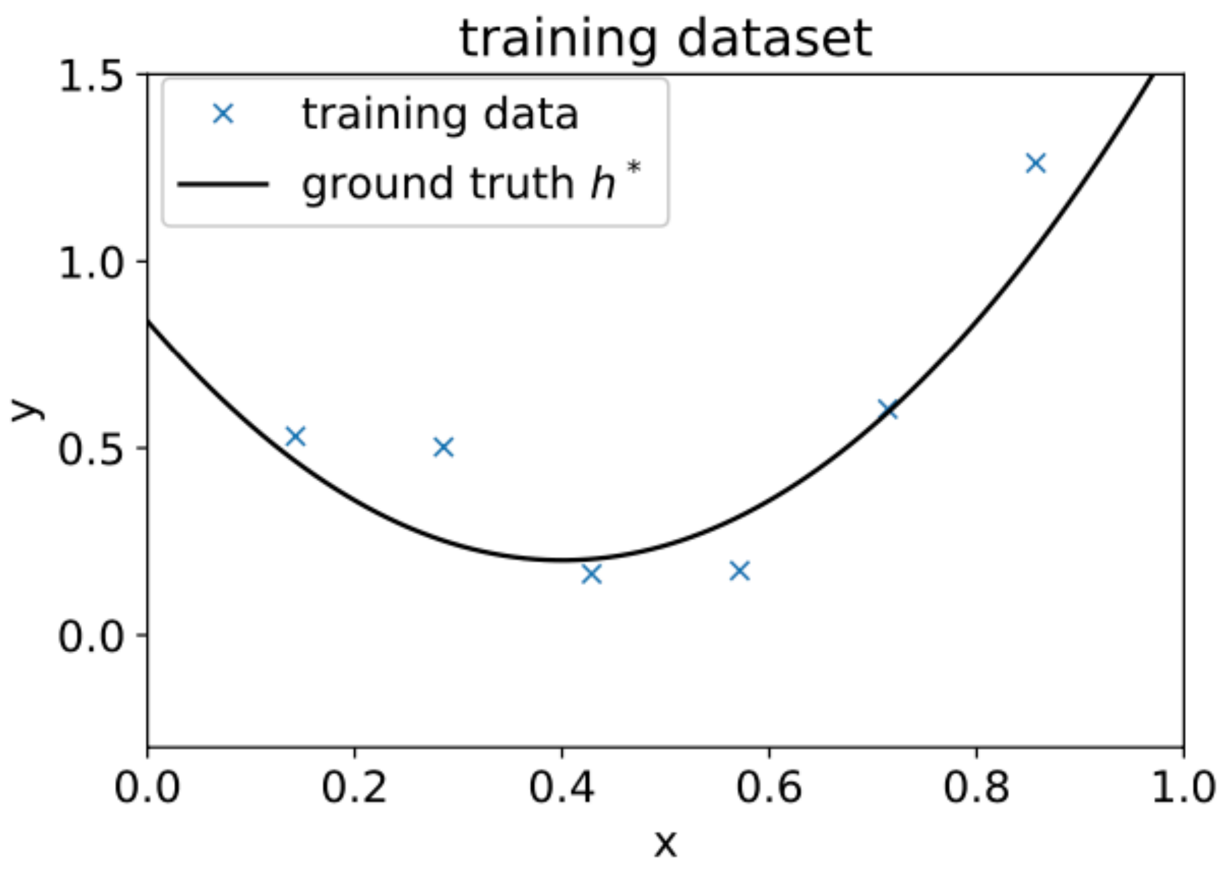
- The training / test empirical distributions are different with finite samples, even though their ground-truth distributions are the same
- In practice, the ground-truth distributions may be different Transfer Learning
- We always want a model that performs well on unseen data (test data)

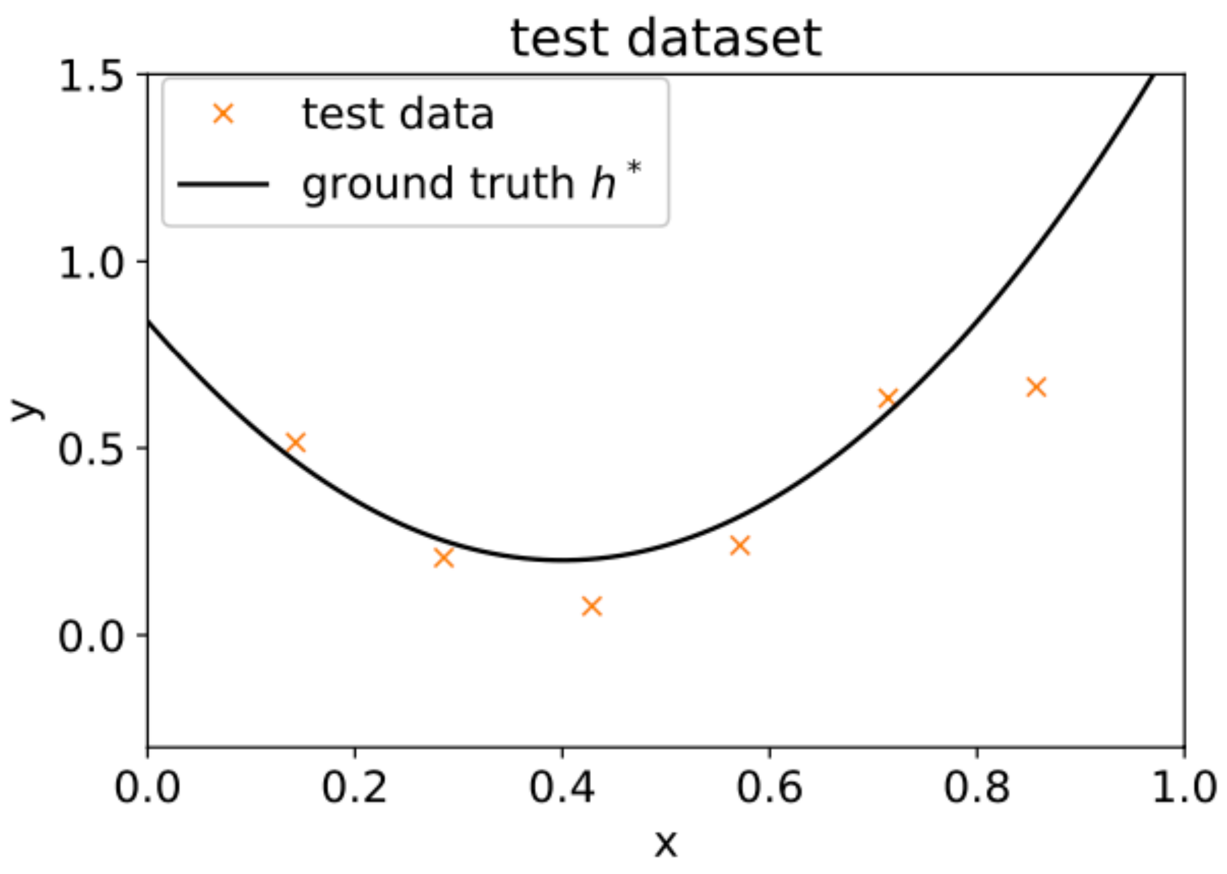
Revisit the Train-Test Mismatch



- The training / test empirical distributions are different with finite samples, even though their ground-truth distributions are the same
- In practice, the ground-truth distributions may be different Transfer Learning
- We always want a model that performs well on unseen data (test data)
- When a model performs well on THE unseen data, we say it generalizes to the data (but not any unseen data)

Revisit the Train-Test Mismatch





- The training / test empirical distributions are different with finite samples, even though their ground-truth distributions are the same
- In practice, the ground-truth distributions may be different Transfer Learning
- We always want a model that performs well on unseen data (test data)
- When a model performs well on THE unseen data, we say it generalizes to the data (but not any unseen data)
- When a model generalizes well to many unseen distributions, we say it is robust

GPT-2

Tom goes everywhere with Catherine Green, a 54-year-old secretary. He moves around her office at work and goes shopping with her. ”Most people don’t seem to mind Tom,” says Catherine, who thinks he is wonderful. ”He’s my fourth child,” she says. She may think of him and treat him that way as her son. He moves around buying his food, paying his health bills and his taxes, but in fact Tom is a dog.

Catherine and Tom live in Sweden, a country where everyone is expected to lead an orderly life according to rules laid down by the government, which also provides a high level of care for its people. This level of care costs money.

People in Sweden pay taxes on everything, so aren’t surprised to find that owning a dog means more taxes. Some people are paying as much as 500 Swedish kronor in taxes a year for the right to keep their dog, which is spent by the government on dog hospitals and sometimes medical treatment for a dog that falls ill. However, most such treatment is expensive, so owners often decide to offer health and even life _ for their dog.

In Sweden dog owners must pay for any damage their dog does. A Swedish Kennel Club official explains what this means: if your dog runs out on the road and gets hit by a passing car, you, as the owner, have to pay for any damage done to the car, even if your dog has been killed in the accident.

Q: How old is Catherine?
A: 54

Q: where does she live?
A:

Radford et al. 2018. Language Models are Unsupervised Multitask Learners

LM

a

We are taking ___ A ? ___ A ?

GPT-2

Tom goes everywhere with Catherine Green, a 54-year-old secretary. He moves around her office at work and goes shopping with her. ”Most people don’t seem to mind Tom,” says Catherine, who thinks he is wonderful. ”He’s my fourth child,” she says. She may think of him and treat him that way as her son. He moves around buying his food, paying his health bills and his taxes, but in fact Tom is a dog.

Catherine and Tom live in Sweden, a country where everyone is expected to lead an orderly life according to rules laid down by the government, which also provides a high level of care for its people. This level of care costs money.

People in Sweden pay taxes on everything, so aren’t surprised to find that owning a dog means more taxes. Some people are paying as much as 500 Swedish kronor in taxes a year for the right to keep their dog, which is spent by the government on dog hospitals and sometimes medical treatment for a dog that falls ill. However, most such treatment is expensive, so owners often decide to offer health and even life _ for their dog.

In Sweden dog owners must pay for any damage their dog does. A Swedish Kennel Club official explains what this means: if your dog runs out on the road and gets hit by a passing car, you, as the owner, have to pay for any damage done to the car, even if your dog has been killed in the accident.

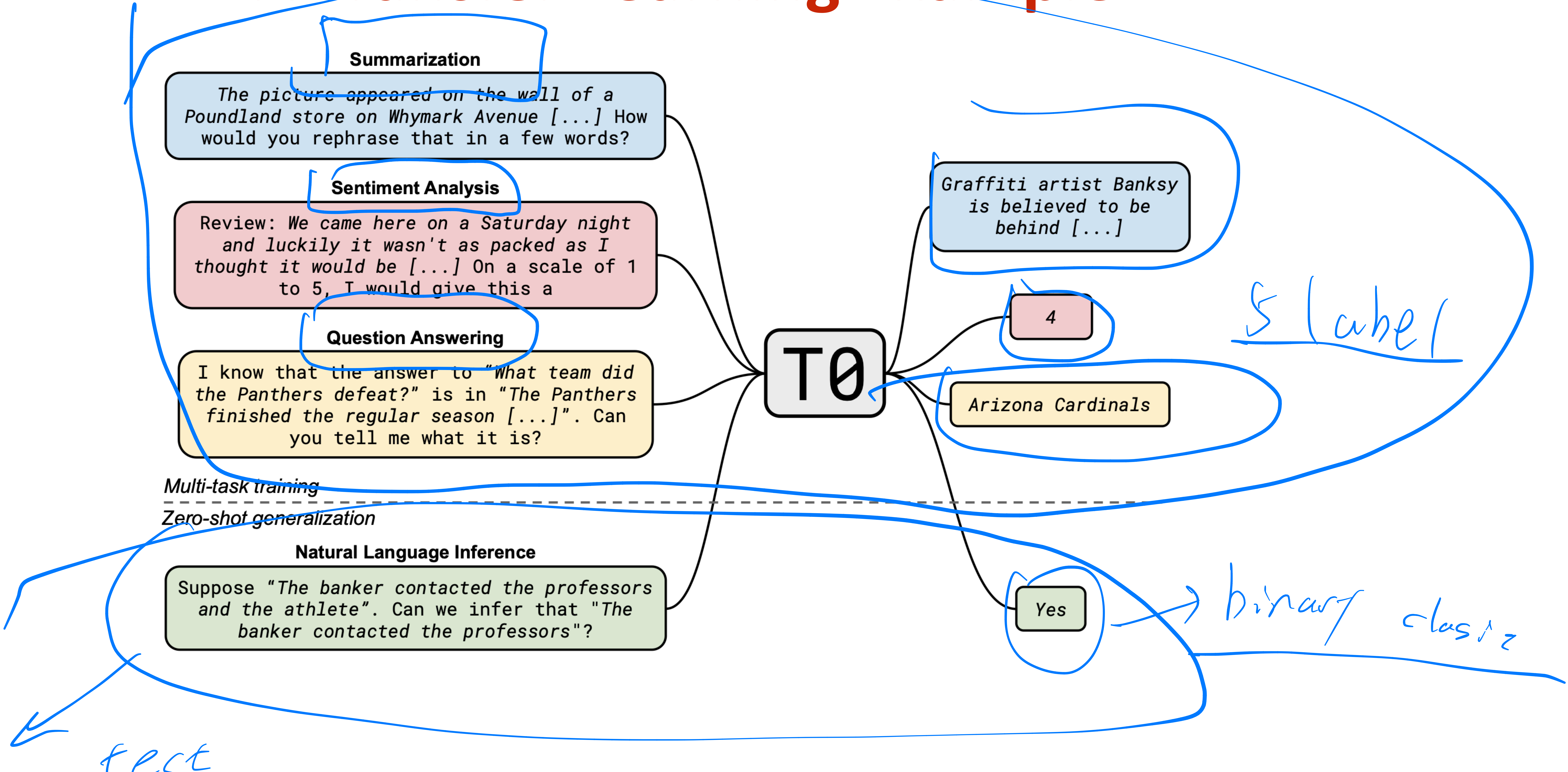
Q: How old is Catherine?
A: 54

Q: where does she live?
A:

When everything is in training, there is no out-of-distribution data

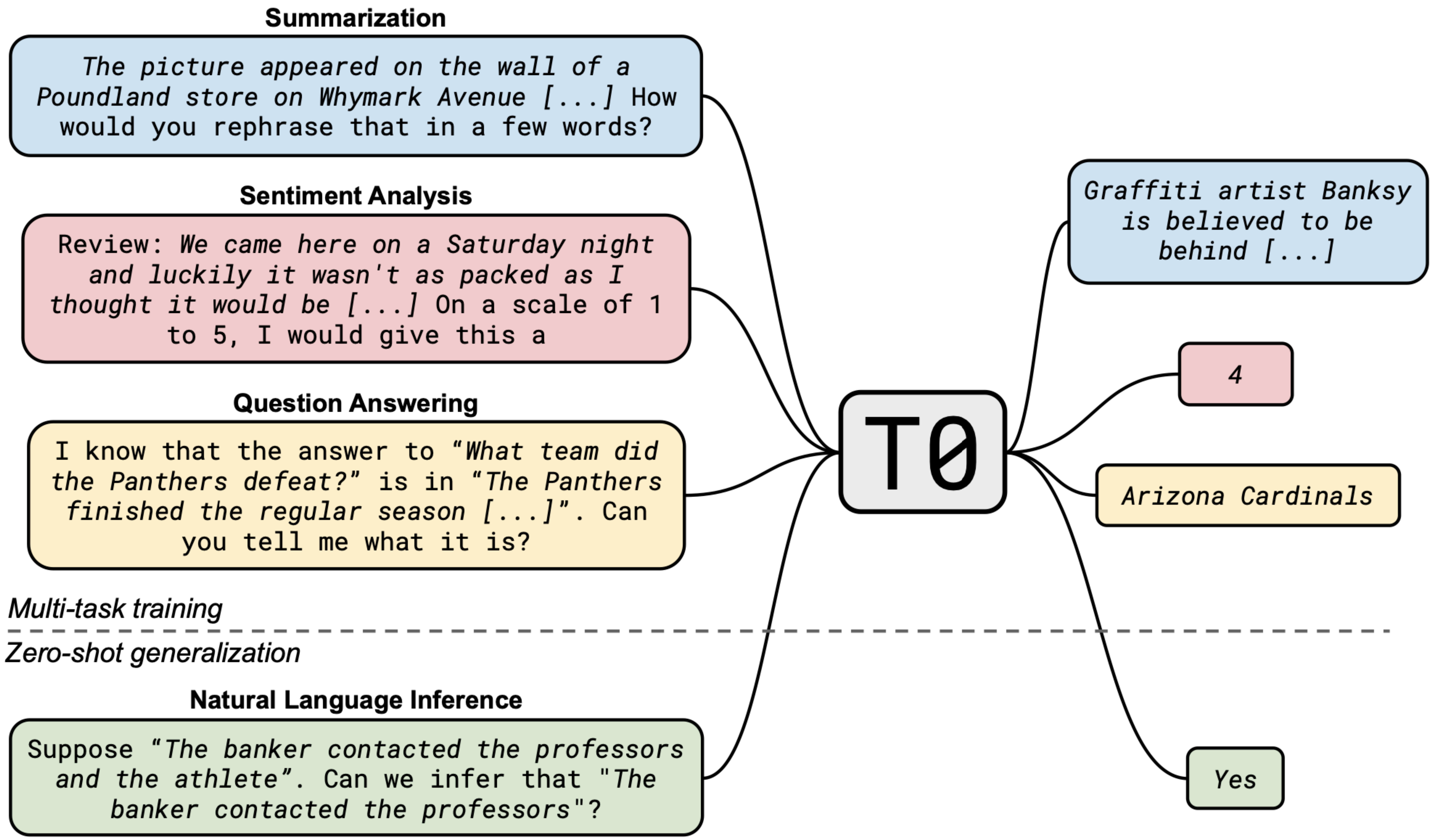
Radford et al. 2018. Language Models are Unsupervised Multitask Learners

A Transfer Learning Example



Sanh et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization

A Transfer Learning Example



Prompts break the task boundary, enabling better transfer

Sanh et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization

How Do We Know Generalization in Practice

- We don’t have test data, cannot compute test error

How Do We Know Generalization in Practice

- We don't have test data, cannot compute test error

Hold-out or Cross-validation

Hold-out method

Hold-out method

<u>Hold - out procedure:</u>

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^{n}$

Hold-out method

spam no spam

1 : 2

1 : 2

Hold - out procedure:

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^{n}$

1) Split into two sets (randomly and preserving label proportion):

Training dataset $D_T = \{X_i, Y_i\}_{i=1}^{m}$

Validation/Hold-out dataset $D_V = \{X_i, Y_i\}_{i=m+1}^{n}$

Hold-out method

Hold – out procedure:

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^{n}$

1) Split into two sets (randomly and preserving label proportion):

Training dataset | Validation/Hold-out dataset

$$D_T = \{X_i, Y_i\}_{i=1}^{m} \qquad D_V = \{X_i, Y_i\}_{i=m+1}^{n}$$

2) Train classifier on D_T. Report error on validation dataset D_V.
Overfitting if validation error is much larger than training error

Hold-out method

Hold – out procedure:

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^{n}$

1) Split into two sets (randomly and preserving label proportion):

Training dataset | Validation/Hold-out dataset

$D_T = \{X_i, Y_i\}_{i=1}^{m}$ $\quad$ $D_V = \{X_i, Y_i\}_{i=m+1}^{n}$

2) Train classifier on D_T. Report error on validation dataset D_V.
Overfitting if validation error is much larger than training error

Validation Error

Hold-out method

Hold – out procedure:

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^n$

1) Split into two sets (randomly and preserving label proportion):

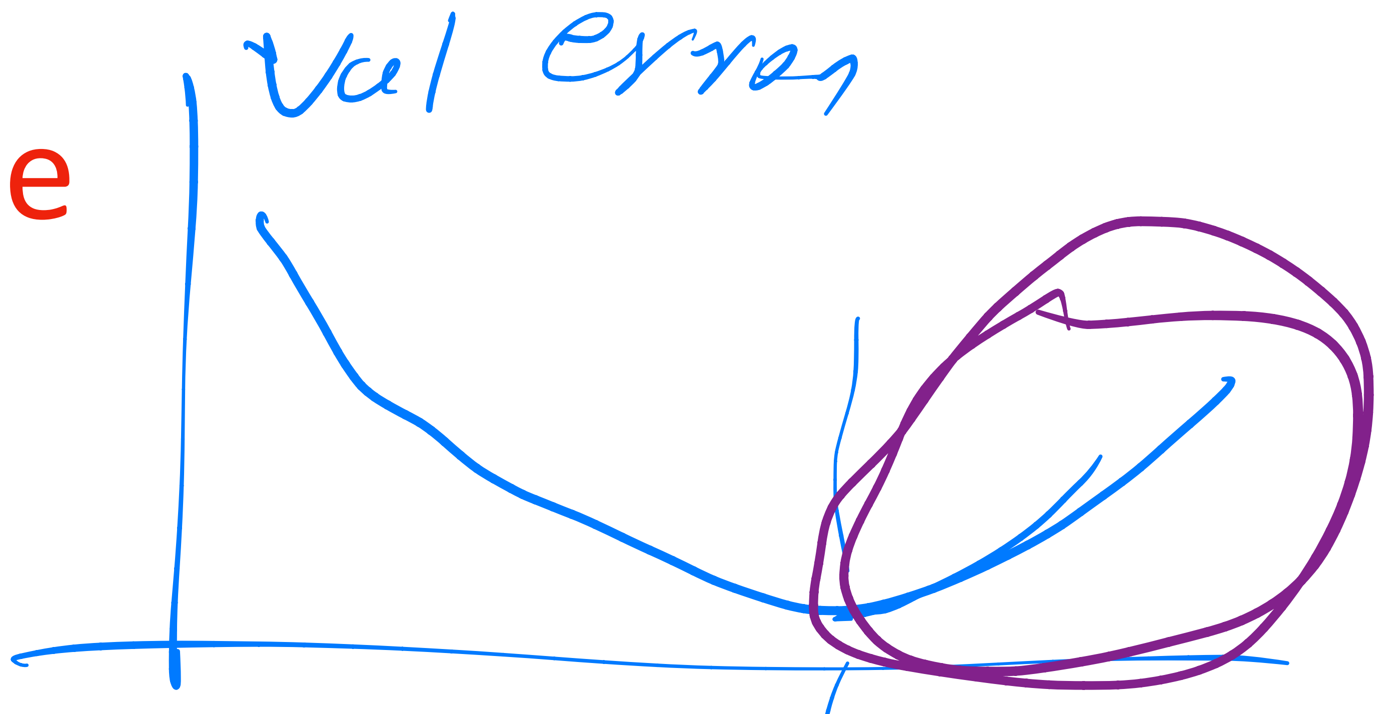
Training dataset $D_T = \{X_i, Y_i\}_{i=1}^m$

Validation/Hold-out dataset $D_V = \{X_i, Y_i\}_{i=m+1}^n$

2) Train classifier on D_T. Report error on validation dataset D_V.
Overfitting if validation error is much larger than training error

In case of gradient descent, we can observe whether the validation error increases

Validation Error



Hold-out method

Hold – out procedure:

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^n$

Use the validation dataset to mimic the test case

1) Split into two sets (randomly and preserving label proportion):

Training dataset

$D_T = \{X_i, Y_i\}_{i=1}^m$

Validation/Hold-out dataset

$D_V = \{X_i, Y_i\}_{i=m+1}^n$

2) Train classifier on D_T. Report error on validation dataset D_V.
Overfitting if validation error is much larger than training error

Validation Error

In case of gradient descent, we can observe whether the validation error increases

Drawback of Hold-Out Method

- Validation error may be misleading if we get an “unfortunate” split

Validation is essentially mimicking the test

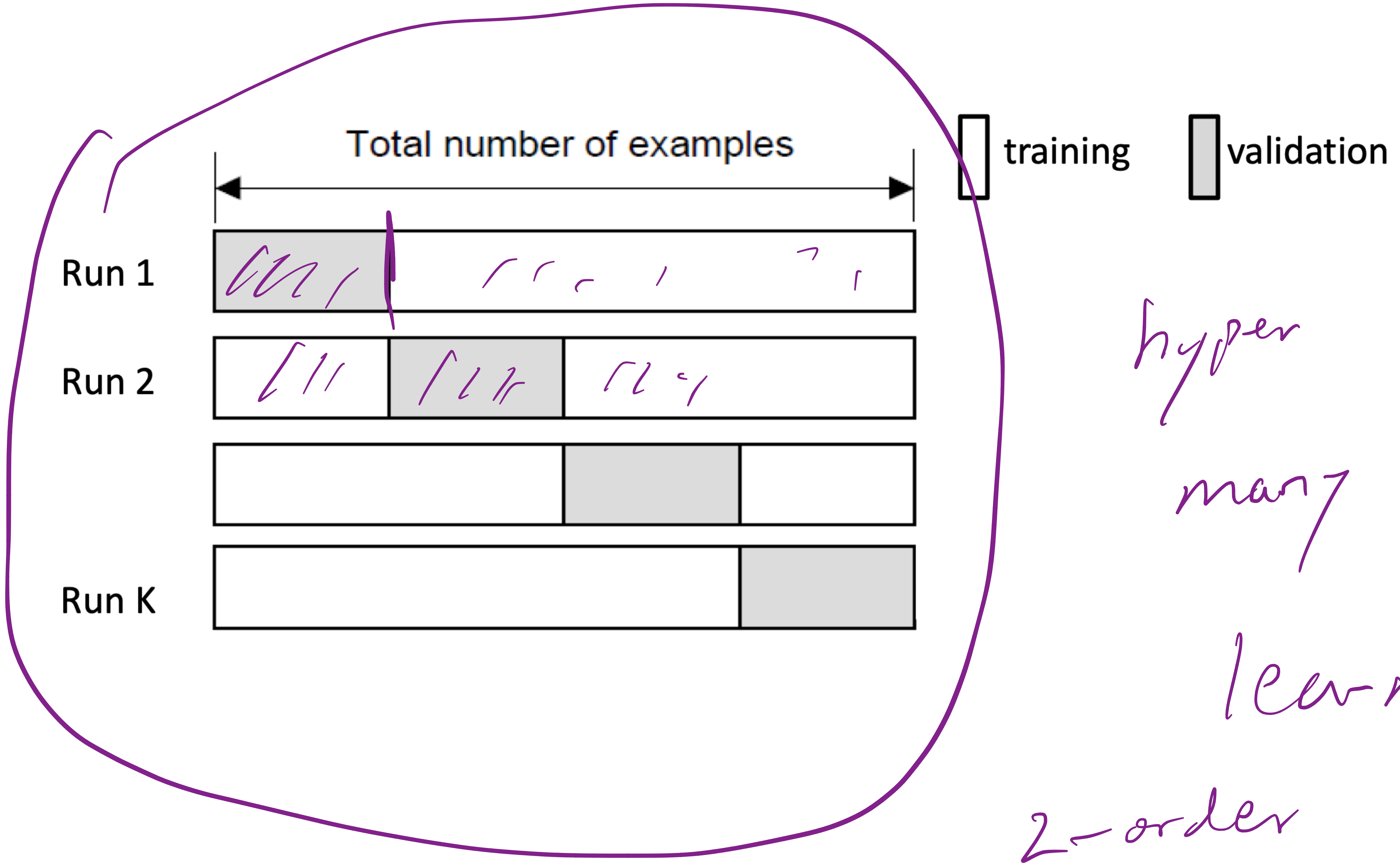
Cross-Validation

K-fold cross-validation

Create K-fold partition of the dataset.

Do K runs: train using K-1 partitions and calculate validation error on remaining partition (rotating validation partition on each run).

Report average validation error



hyper

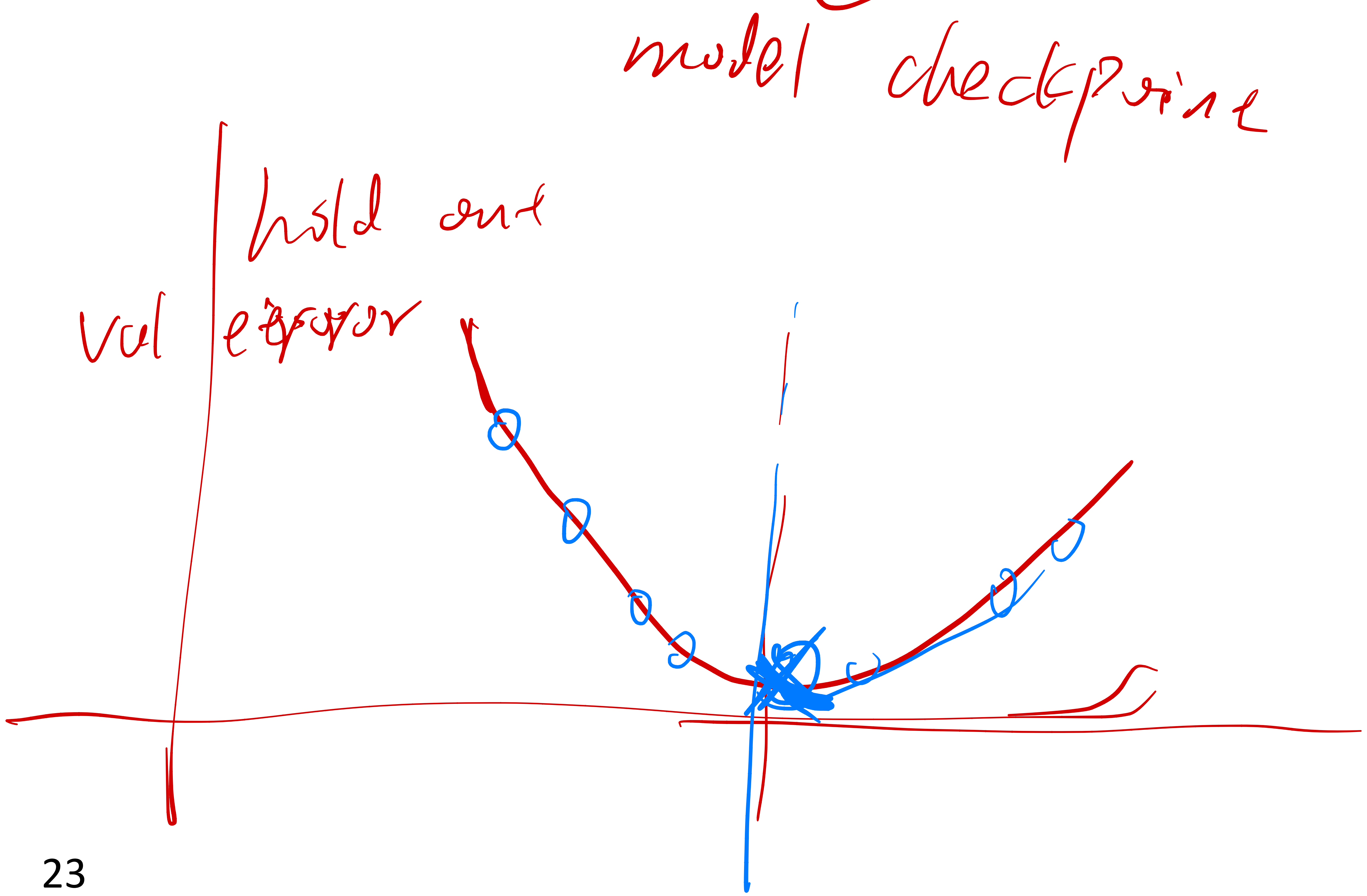
many step

learning rule

2-order 3 order 5 orde

Drawback of Cross-Validation

- Cannot be used to select a specific model, more often used to select method design, hyperparameters, etc.
- Expensive



Drawback of Cross-Validation

- Cannot be used to select a specific model, more often used to select method design, hyperparameters, etc.

- Expensive

Hold-out is more commonly used nowadays, and the validation dataset is provided in advance

Hold-Out Method

Validation is essentially mimicking the test, always try to pick validation data that may align with test data, unnecessarily to hold out training data for validation

Train, Validation, Test

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Train, Validation, Test

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Test dataset is another set of pairs $\{(\tilde{x}^{(1)}, \tilde{y}^{(1)}), \cdots, (\tilde{x}^{(L)}, \tilde{y}^{(L)})\}$

Does not overlap with training and validation dataset

Train, Validation, Test

Validation dataset is another set of pairs $\{(\hat{x}^{(1)}, \hat{y}^{(1)}), \cdots, (\hat{x}^{(m)}, \hat{y}^{(m)})\}$

Does not overlap with training dataset

Test dataset is another set of pairs $\{(\tilde{x}^{(1)}, \tilde{y}^{(1)}), \cdots, (\tilde{x}^{(L)}, \tilde{y}^{(L)})\}$

Does not overlap with training and validation dataset

Completely unseen before deployment

Realistic setting

Validation is Very Important

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)
- Decide when to stop training

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)
- Decide when to stop training

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)
- Decide when to stop training
- Select hyperparameters

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)
- Decide when to stop training
- Select hyperparameters

Hyperparameter tuning

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)
- Decide when to stop training
- Select hyperparameters

Hyperparameter tuning

When you tune hyperparameters harder, it is more likely the validation error would mismatch the test error, because you are overfitting on the validation

Validation is Very Important

- Track underfitting/overfitting (in case of iterative training)
- Decide when to stop training
- Select hyperparameters

Hyperparameter tuning

When you tune hyperparameters harder, it is more likely the validation error would mismatch the test error, because you are overfitting on the validation

Hyperparameter tuning is a form of training

Good ML Practice

Good ML Practice

- Do not look at or evaluate on the test dataset

Good ML Practice

- Do not look at or evaluate on the test dataset
- Always track the training and validation metrics/errors/losses

Good ML Practice

- Do not look at or evaluate on the test dataset

  Many people are implicitly using test dataset as validation

- Always track the training and validation metrics/errors/losses

Thank You!
Q & A