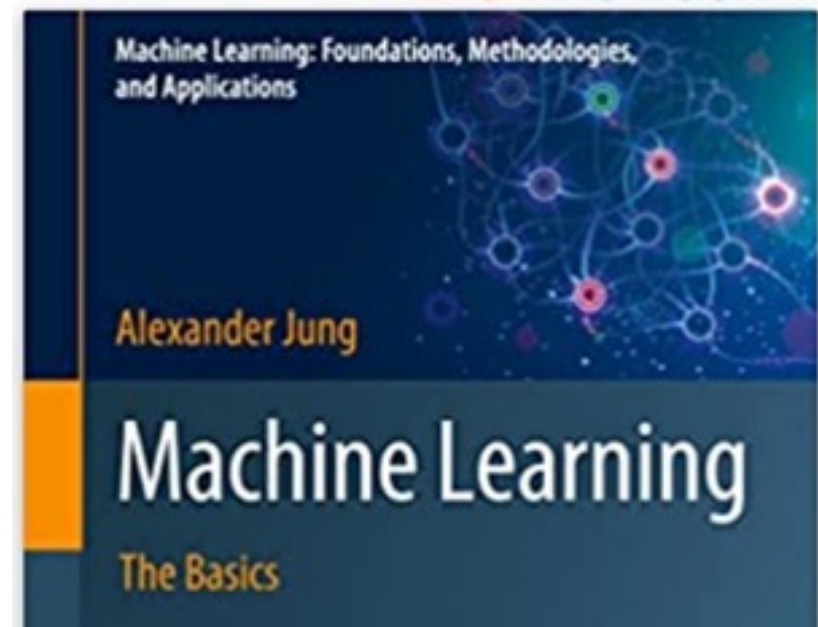# Model Validation and Selection

Alex(ander) Jung
Assistant Professor for Machine Learning
Department of Computer Science
Aalto University

Reading.

Ch. 6 of https://mlbook.cs.aalto.fi

https://scikit-learn.org/stable/model_selection.html

"Model"
=
Hypothesis Space

Learning Goals

- know train err is bad quality measure for ML method

- val.err. is more useful as quality measure for a ML model

- basic idea of k-fold CV

- hyper-parameter tuning = model selection

- Python implementations of k-fold CV / gridsearch

ML – In a Nutshell

- learn hypothesis h(.) out of model such that for any

  data point h(x)≈y

- approximation quality measured by loss L((x,y),h)

- approximate "any data point" by a training set

Model Validation
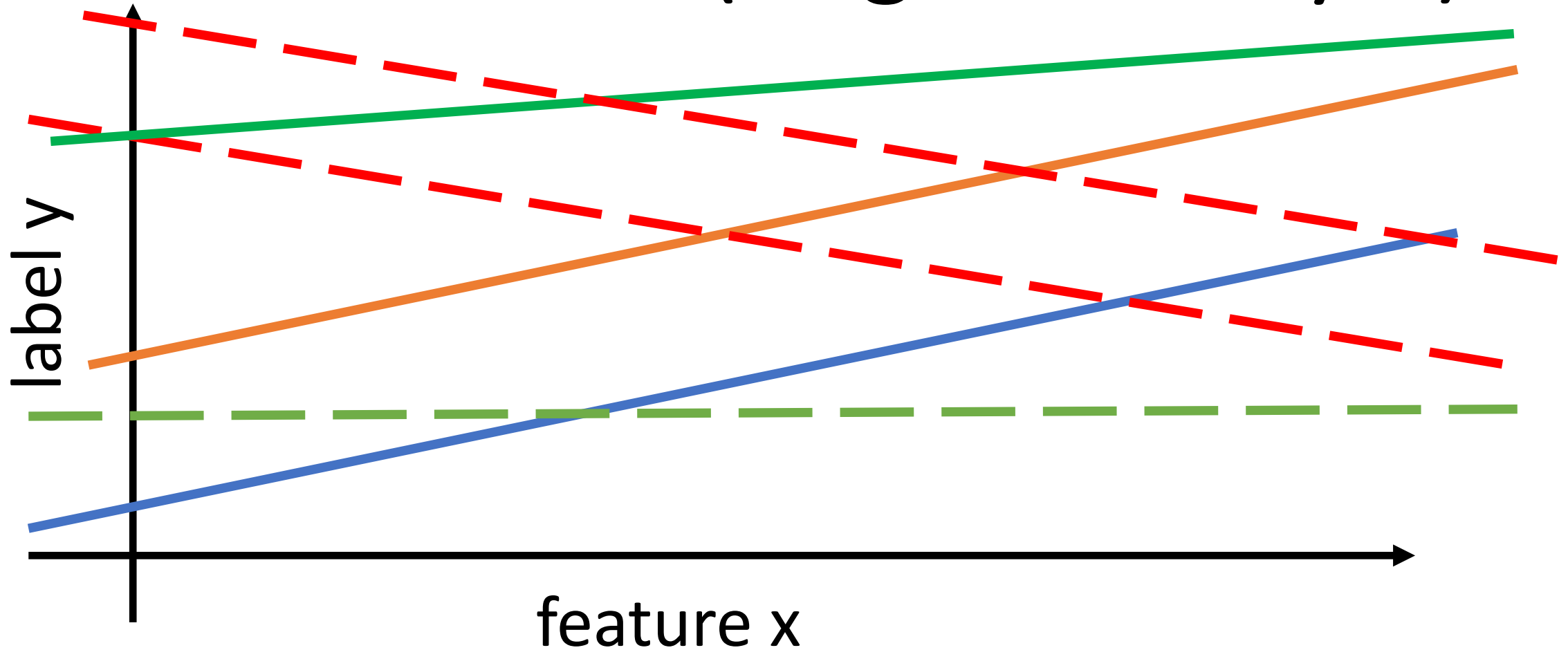
How do we know a model is any good ?

Model Selection
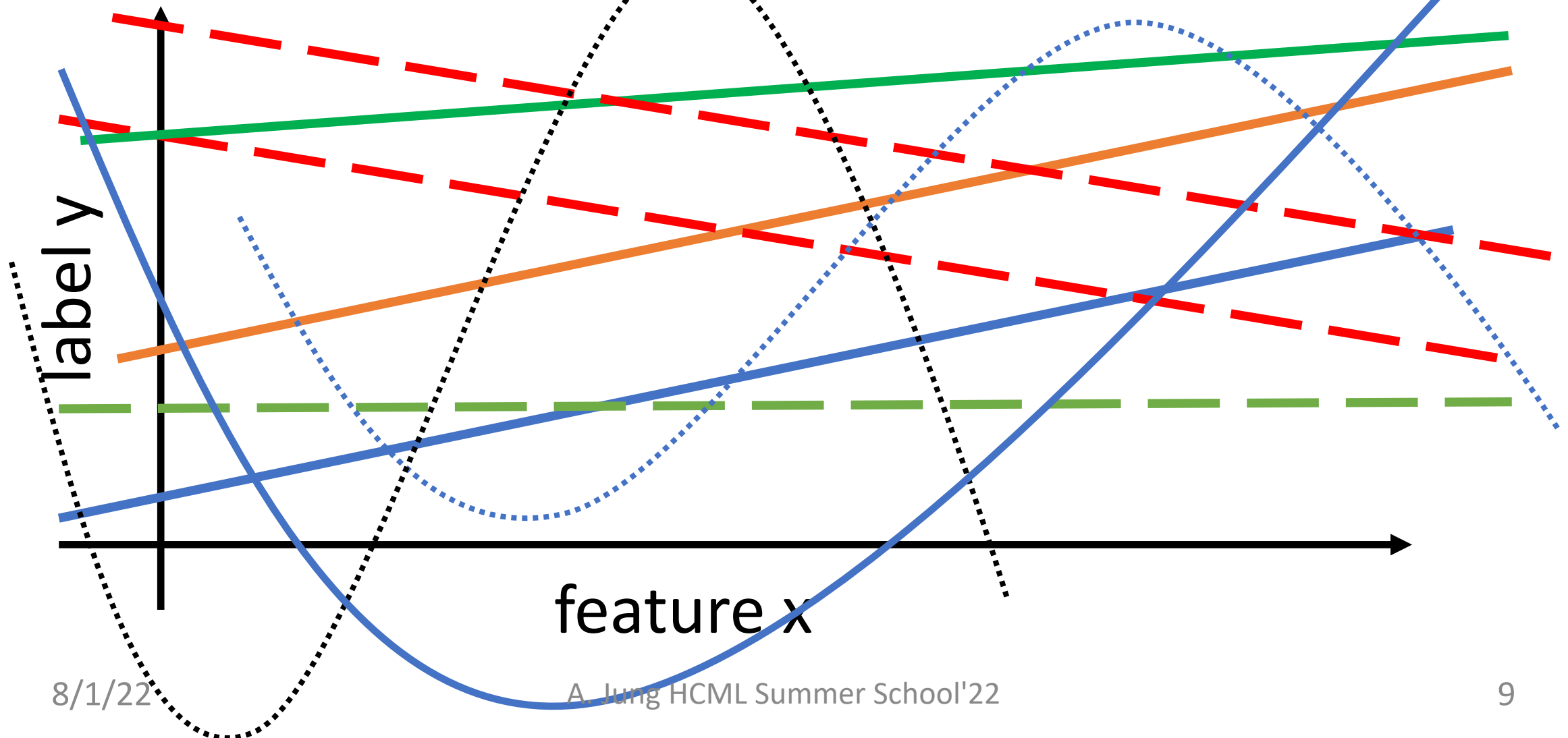
How to choose between different alternative models?
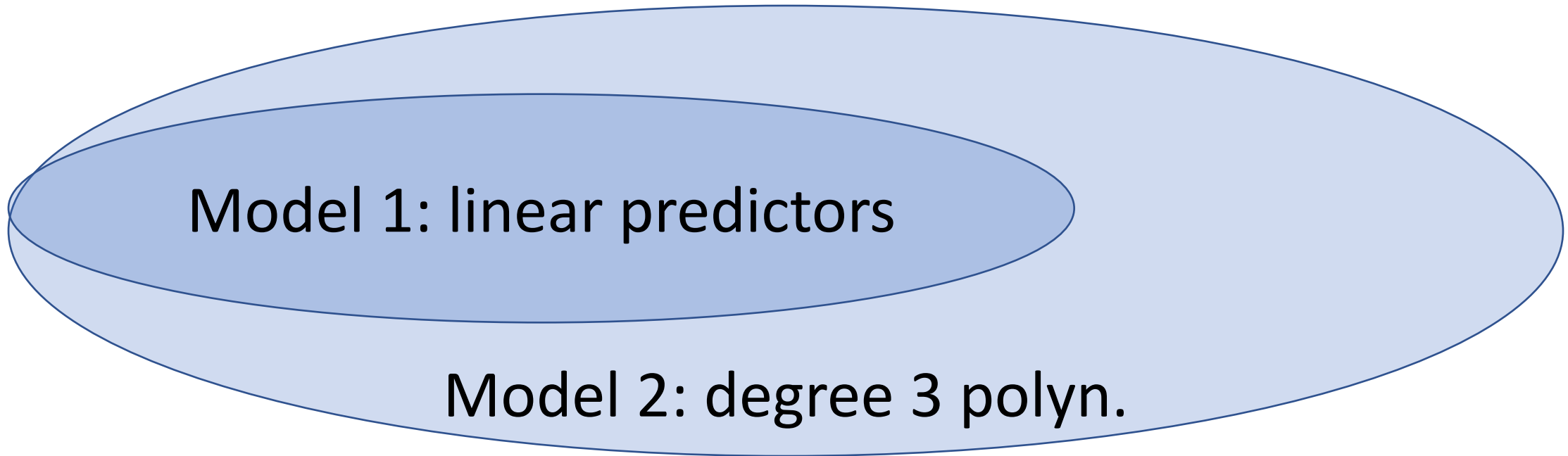
Model 1:
Linear Predictors (Degree 1 Polyn.)

label y

feature x

Model 2:
Degree 3 Polyn. Predictors

label y

feature x

Nested Models – I

Model 1: linear predictors

Model 2: degree 3 polyn.

Math Notation

$$\mathcal{H}^{(n)} = \left\{ h(x) = \sum_{l=0}^{n} w_l x^l \text{ with some } w_l \right\}$$
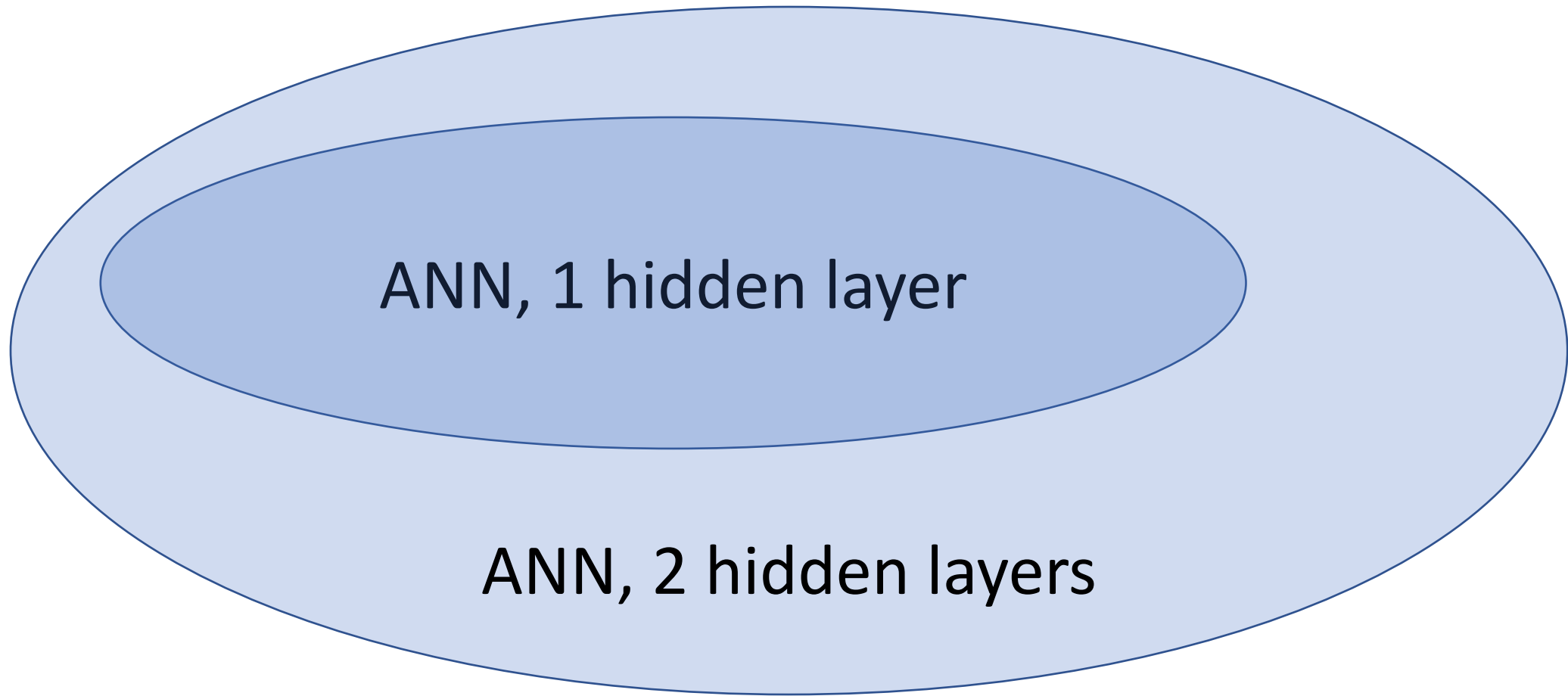
$\mathcal{H}^{(0)}$ ... constant prediction (ignores feature)
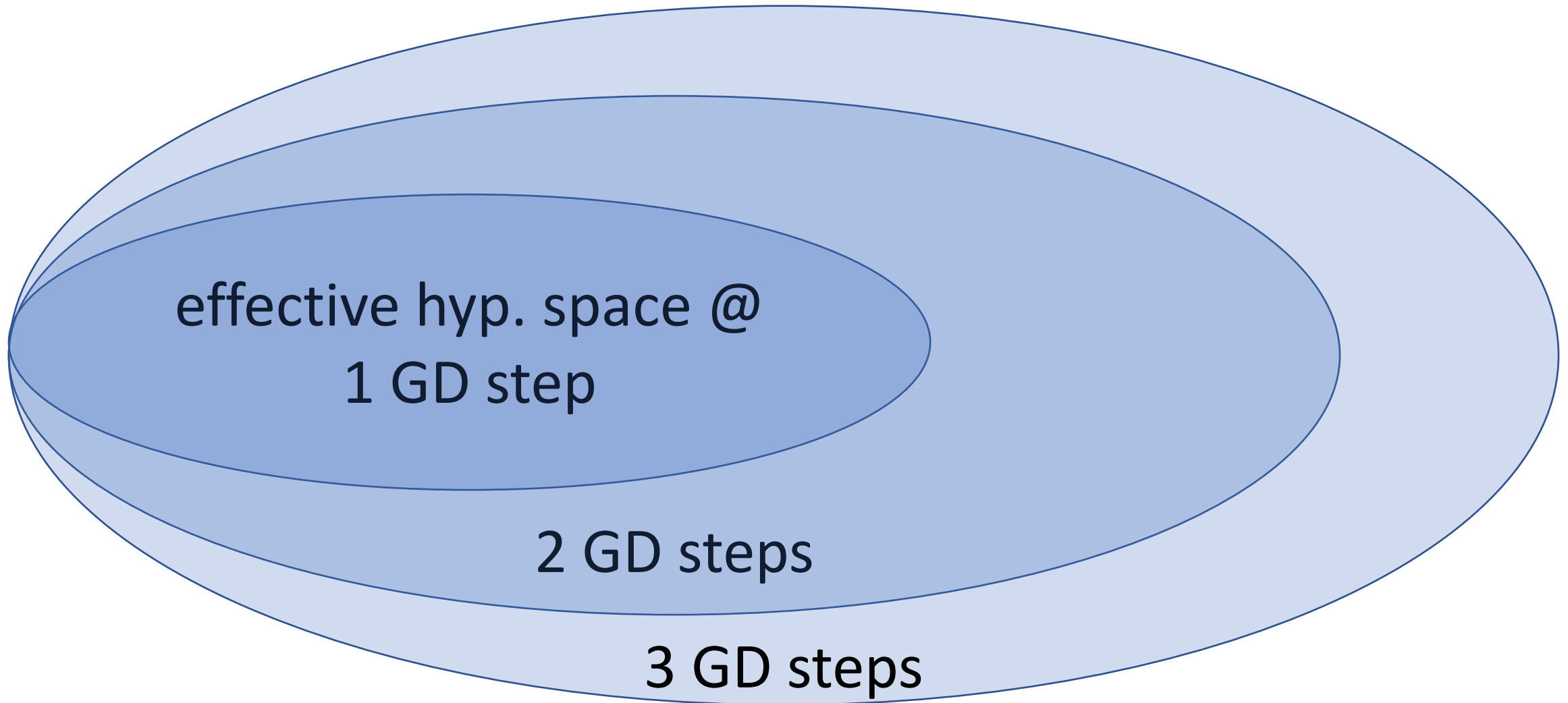
$\mathcal{H}^{(1)}$ ... linear hypotheses

$\mathcal{H}^{(3)}$ ... degree 3 polyn.

$\mathcal{H}^{(0)} \subseteq \mathcal{H}^{(1)} \subseteq \mathcal{H}^{(2)} \subseteq \mathcal{H}^{(3)} \subseteq \ldots$
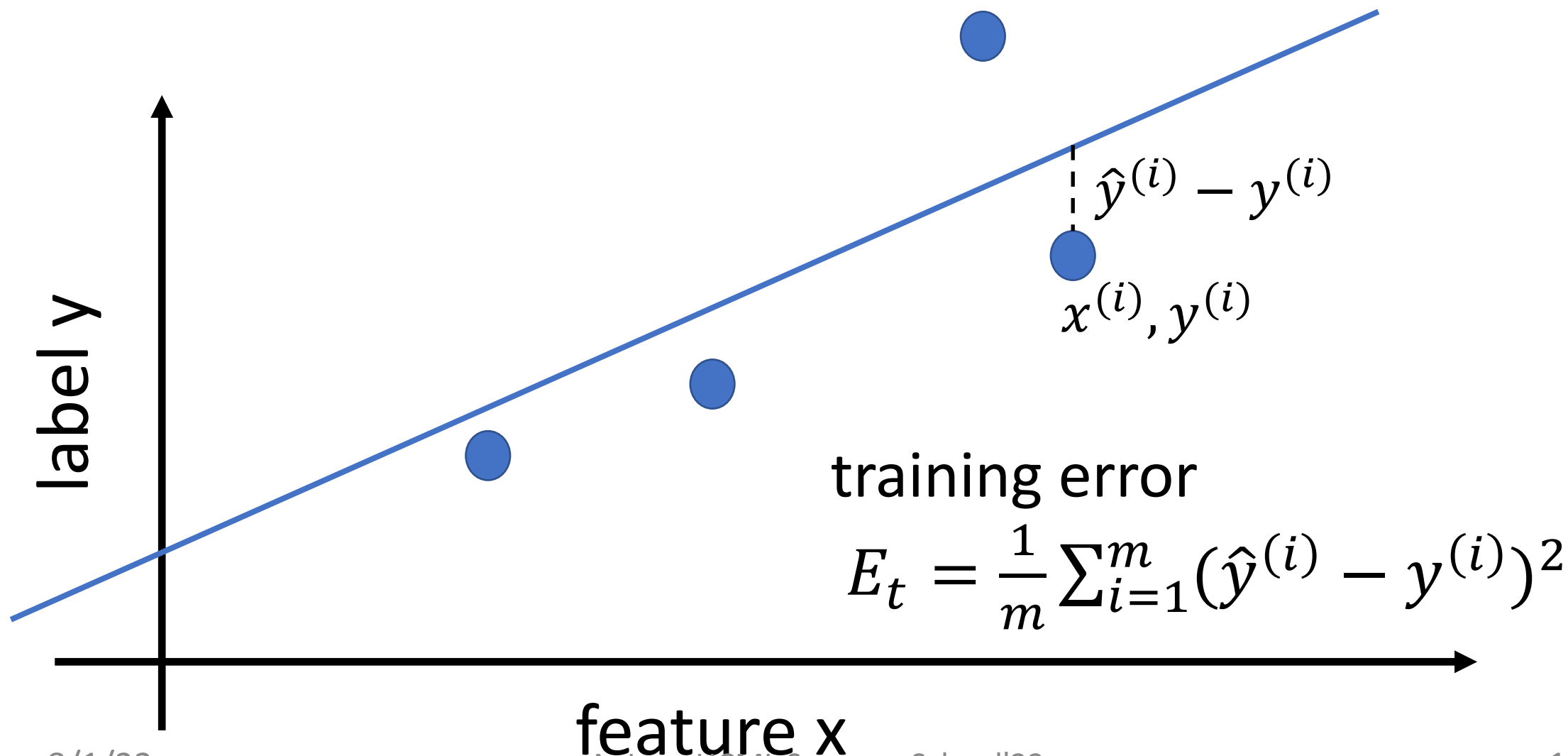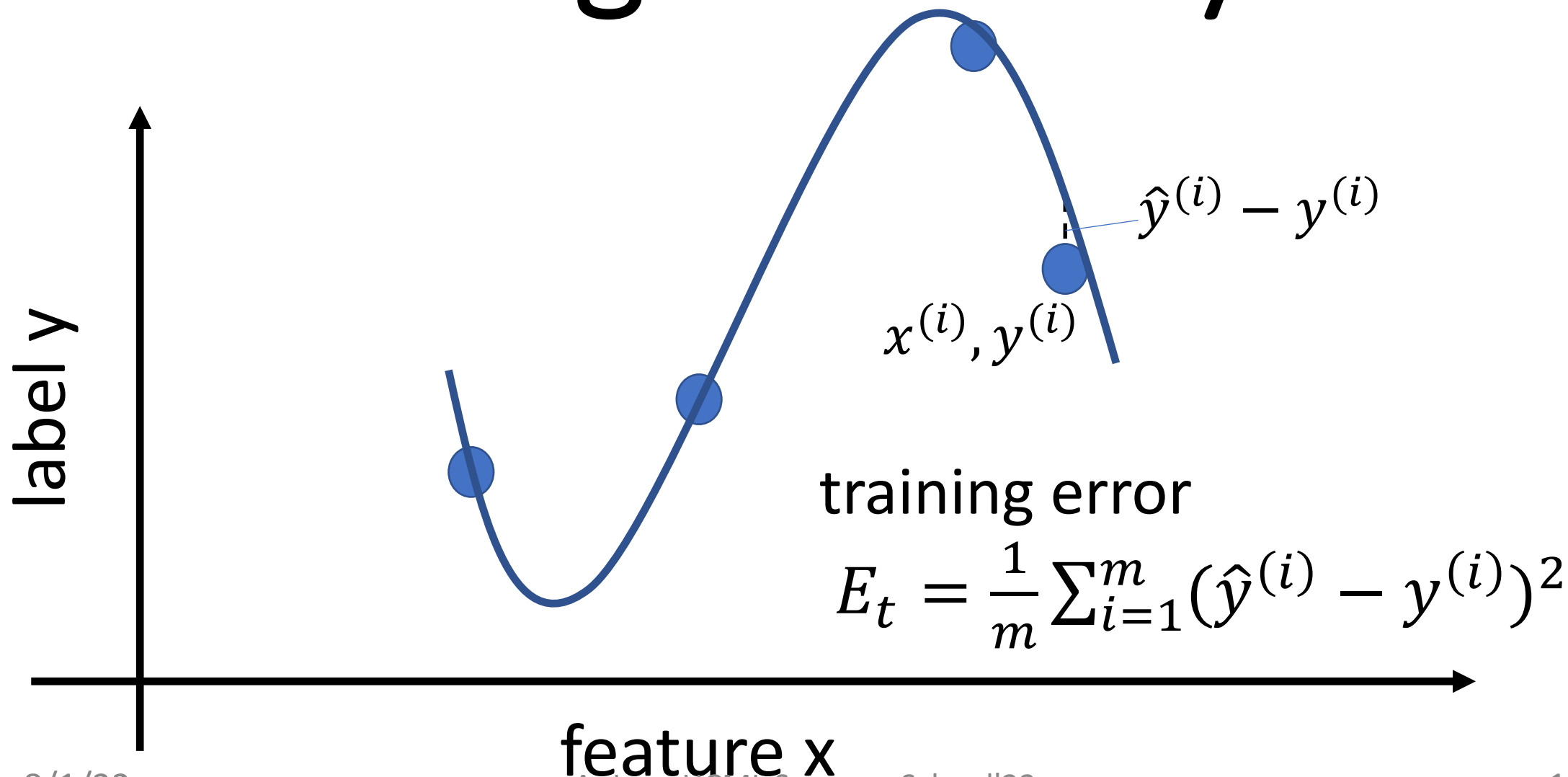
Nested Models - II

ANN, 1 hidden layer

ANN, 2 hidden layers

Nested Models - III

effective hyp. space @
1 GD step

2 GD steps

3 GD steps

Learn Linear Predictor

$\hat{y}^{(i)} - y^{(i)}$

$x^{(i)}, y^{(i)}$

label y

training error

$$E_t = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

feature x

A. Jung HCML Summer School'22

Learn Degree 3 Polyn.

$\hat{y}^{(i)} - y^{(i)}$

$x^{(i)}, y^{(i)}$

label y

training error

$$E_t = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

feature x
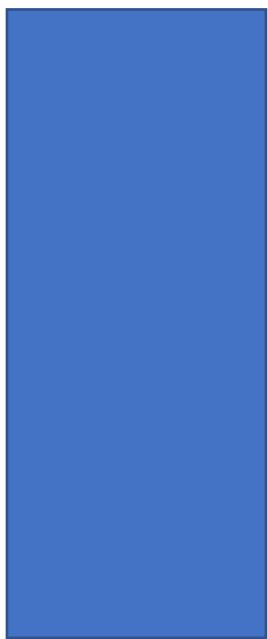
Training Errors

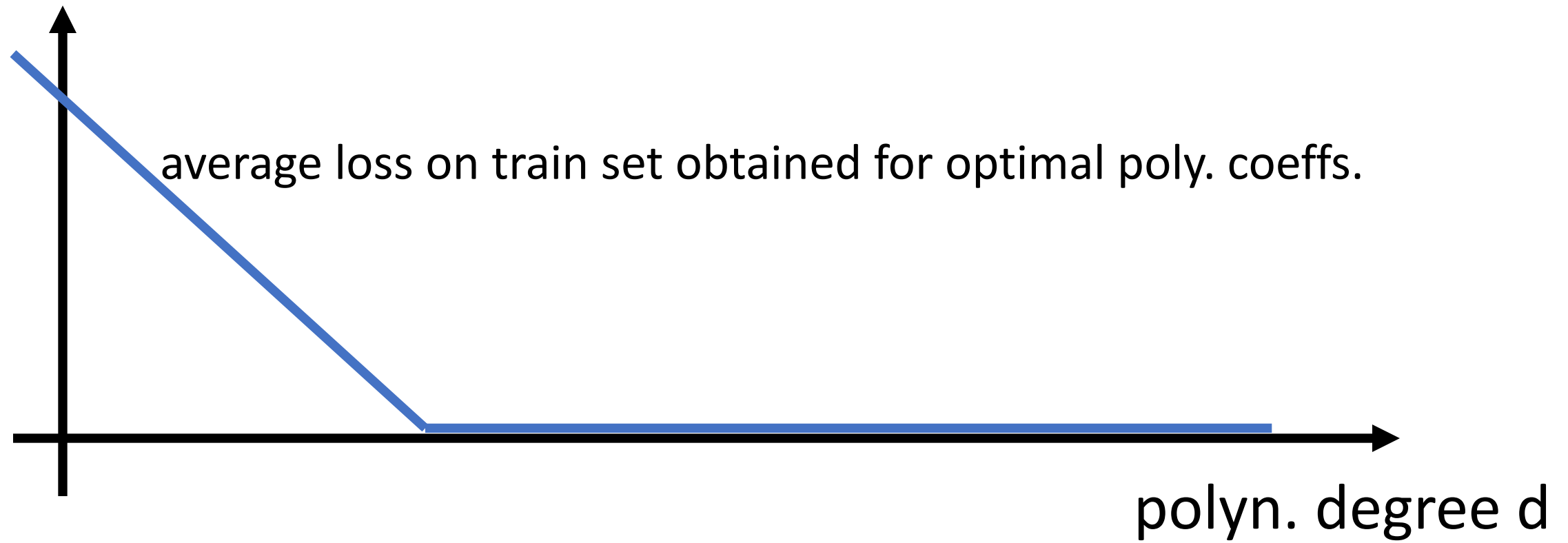model 1
linear predictors

model 2:
degree 3 polyn.

Train Error vs. Degree

average loss on train set obtained for optimal poly. coeffs.

polyn. degree d

Train Error vs. ANN Layers

average loss on train set obtained for optimal network weights
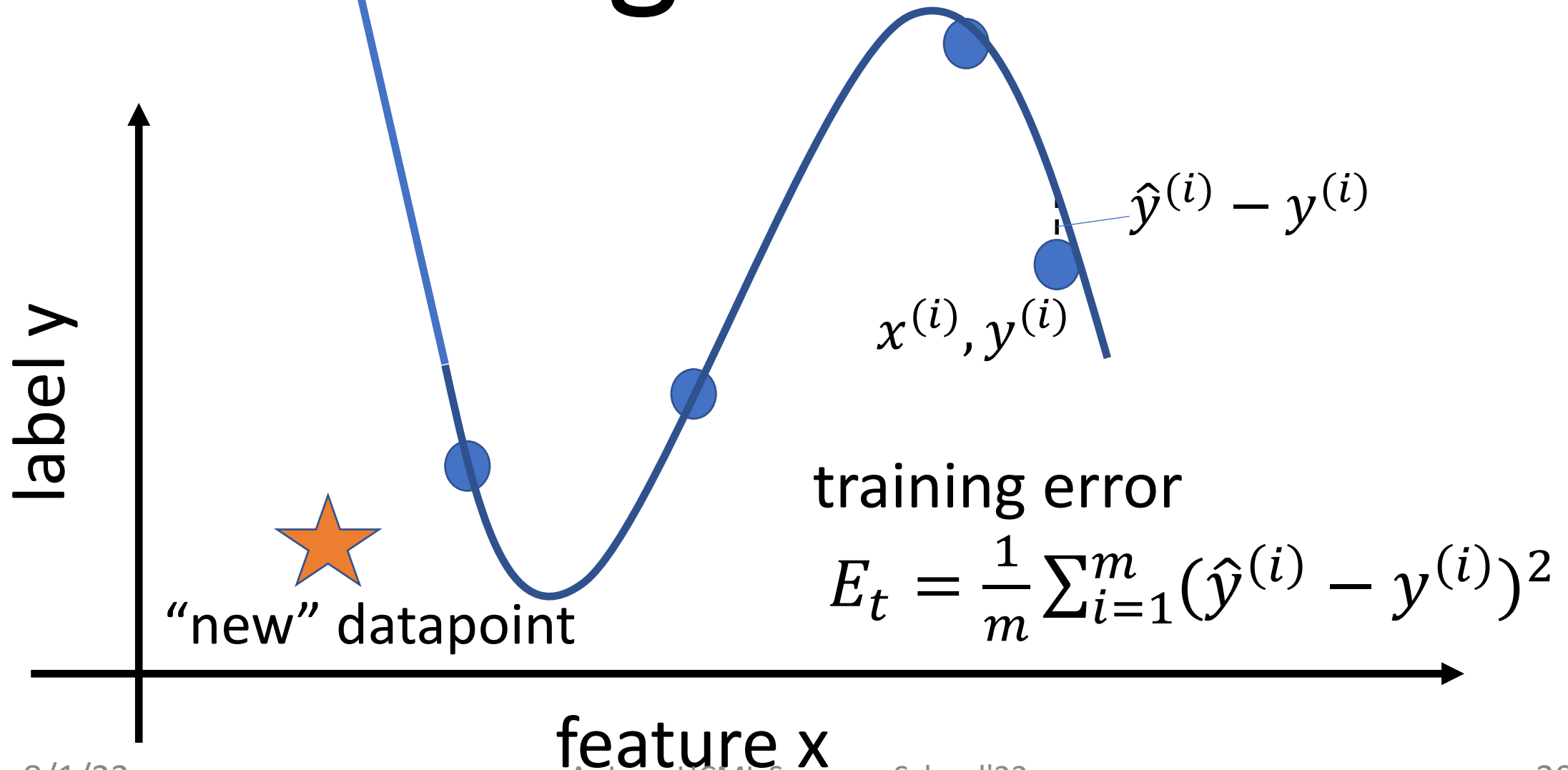
\# hidden layers
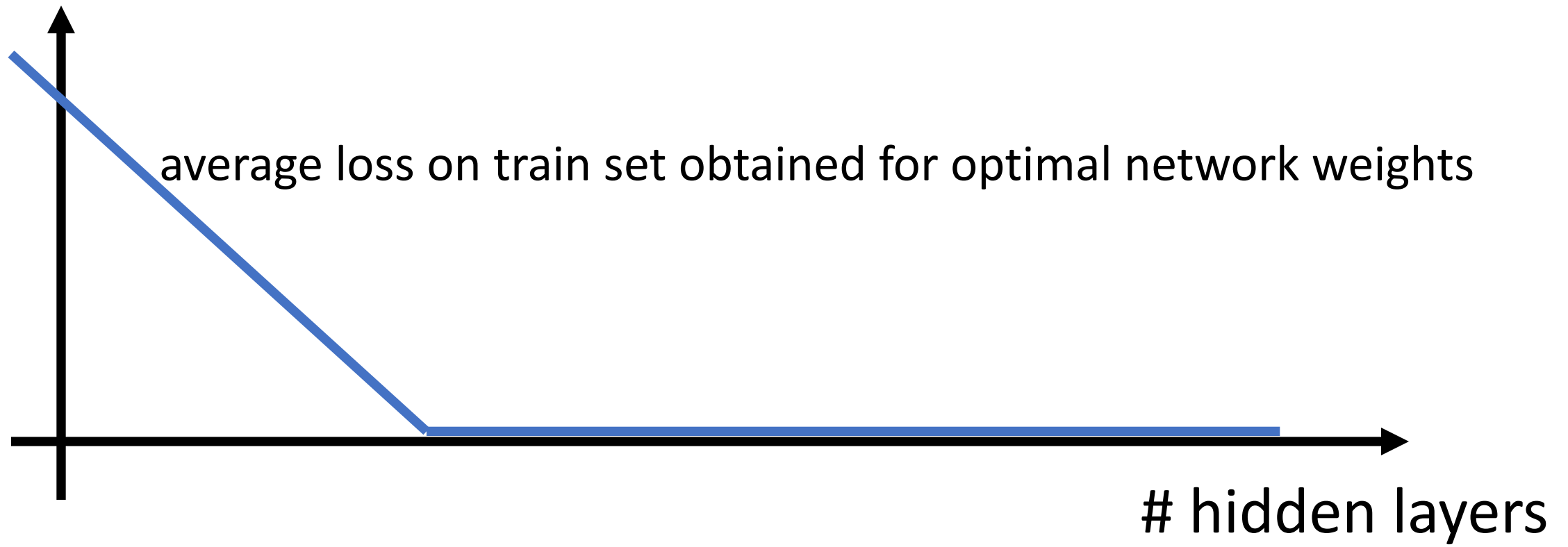
Train Error vs. Gradient Steps

average loss on train set

gradient steps

Overfitting

label y

feature x

$\hat{y}^{(i)} - y^{(i)}$

$x^{(i)}, y^{(i)}$

training error

$$E_t = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$$
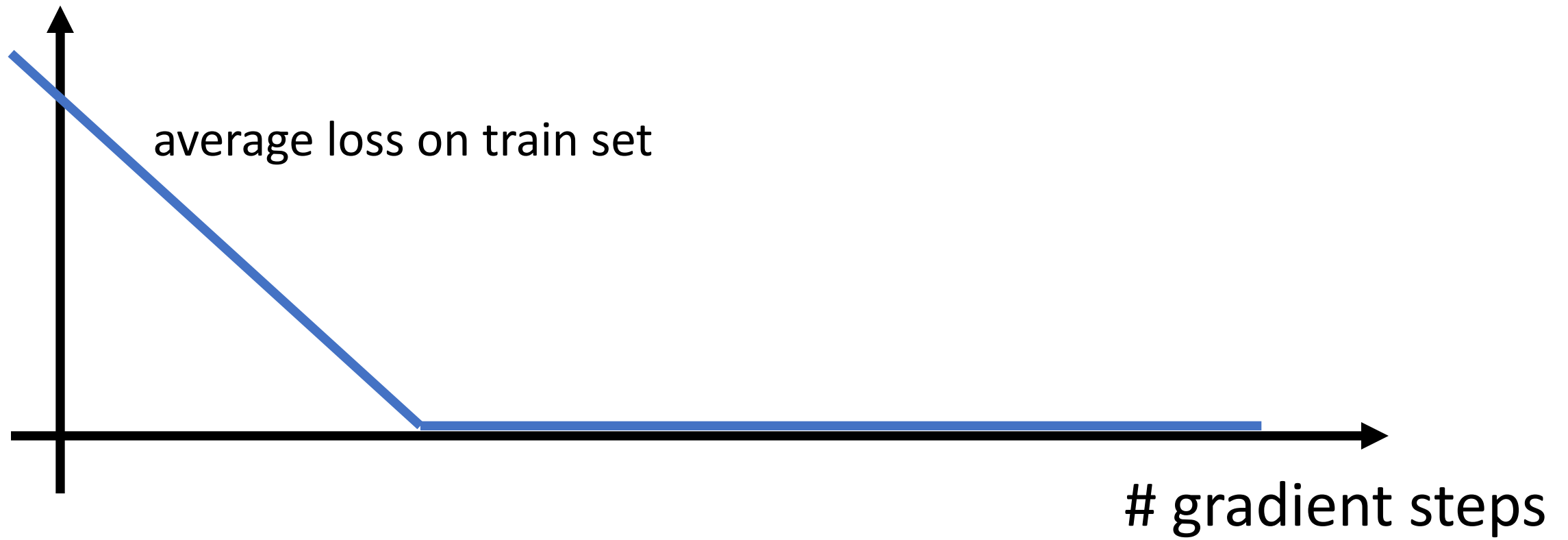
"new" datapoint

small training error does not imply good performance on new data points!

small training error merely indicates that training algorithm has been implemented correctly

A. Jung HCML Summer School'22

A Case in Point

we can perfectly fit (almost) any m data points using polynomials of degree n as soon as
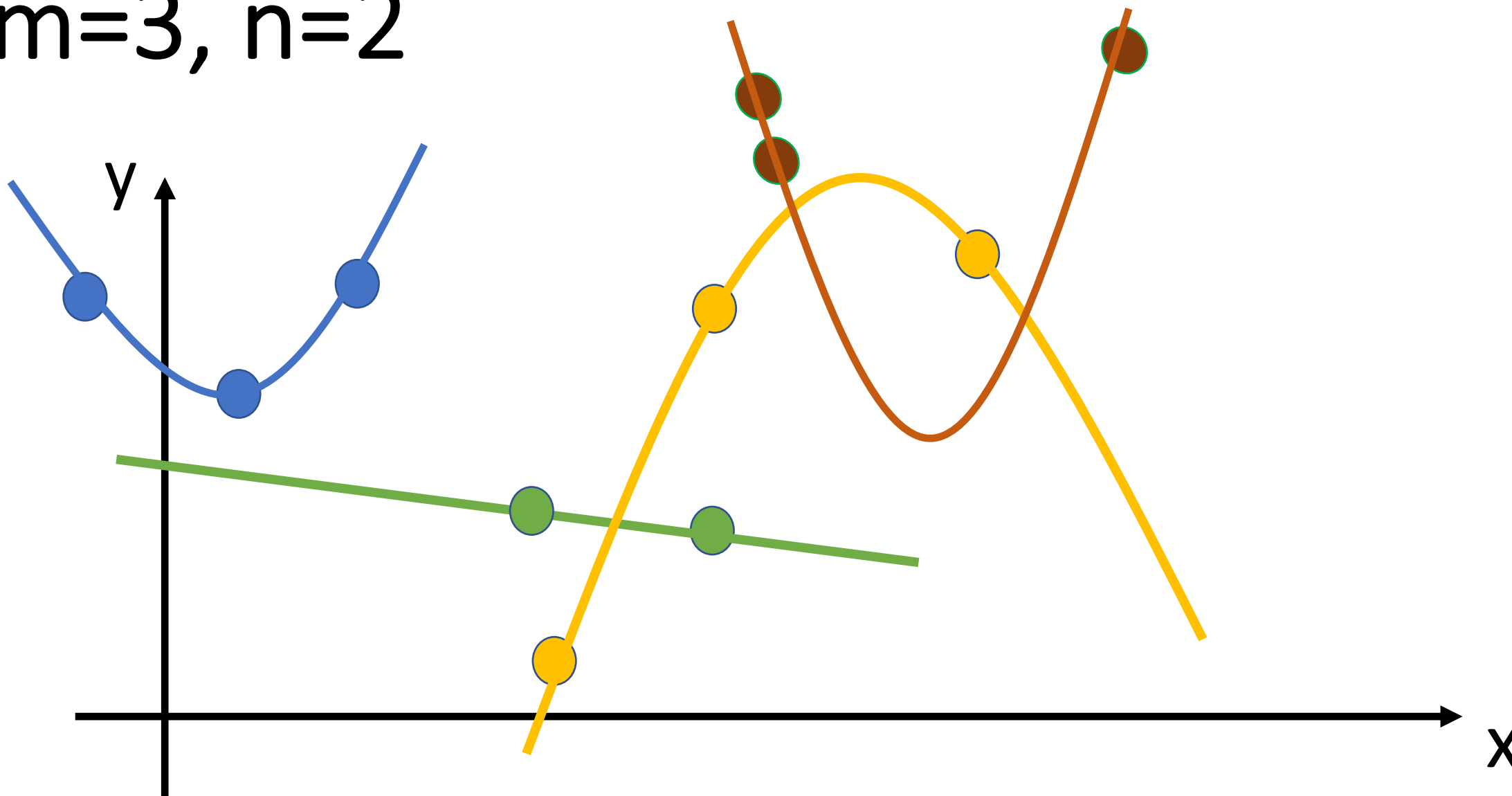
$$n \geq m-1$$

m=2, n=1

A. Jung HCML Summer School'22
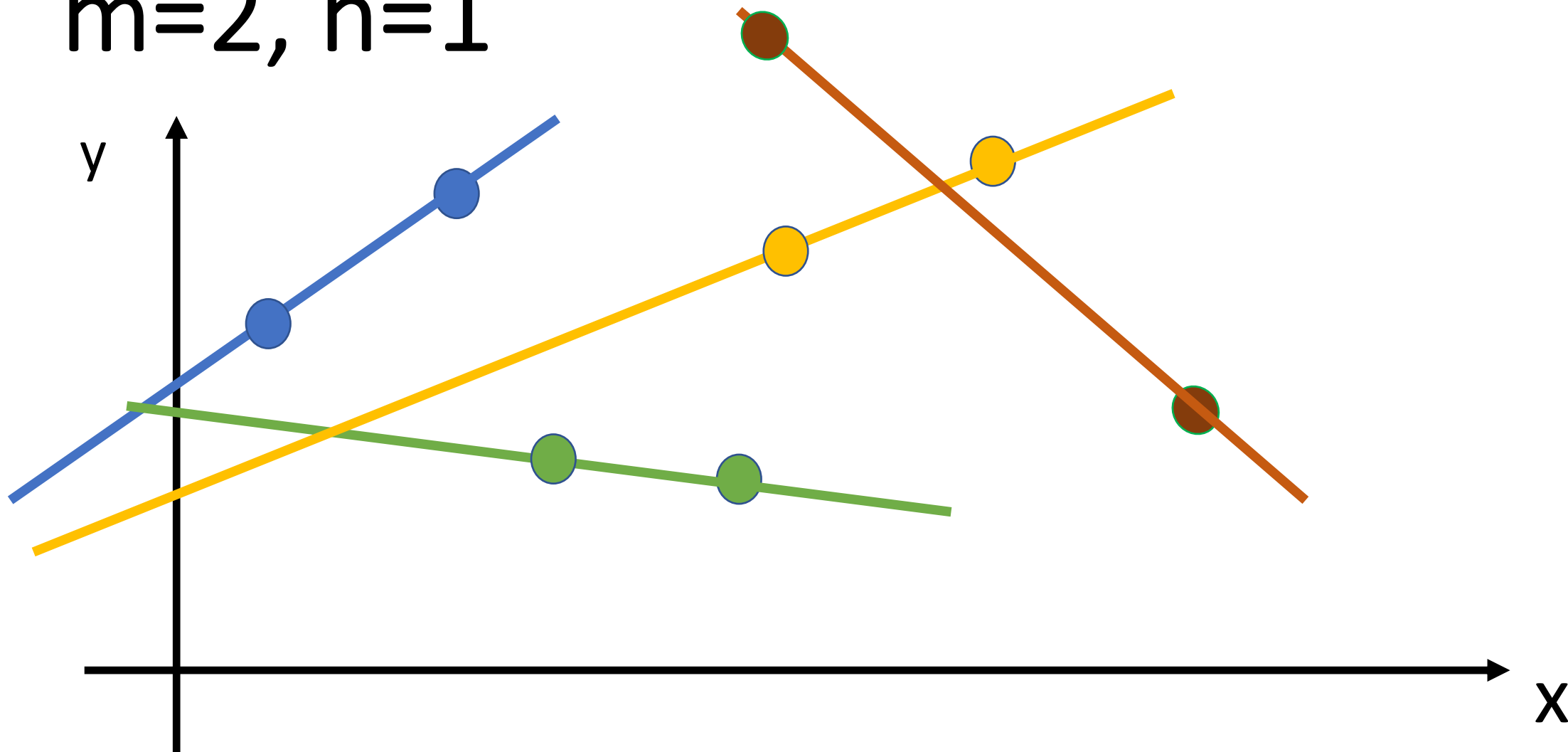
m=3, n=2

Reminder: Probabilistic Model

- data points are realizations of RVs

- joint pdf p(x,y) of features and label

- training set is a RV

- learnt hypothesis h(.) is a RV

- prediction h(x) is a RV

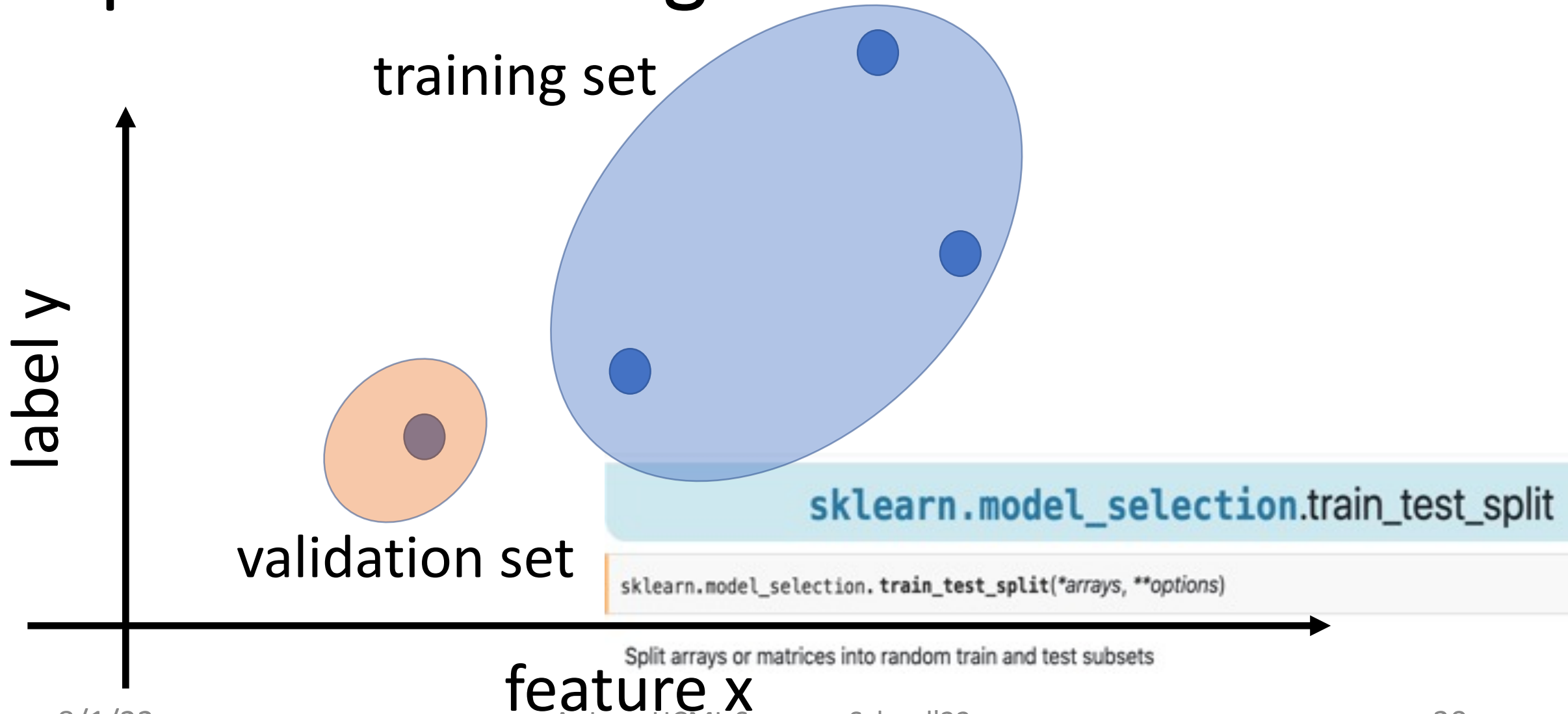# Why Can Train. Err. Mislead?

- consider expected loss of hypothesis

- estimate expectation using sample average
- this only works if hypothesis does not depends on data points used in average

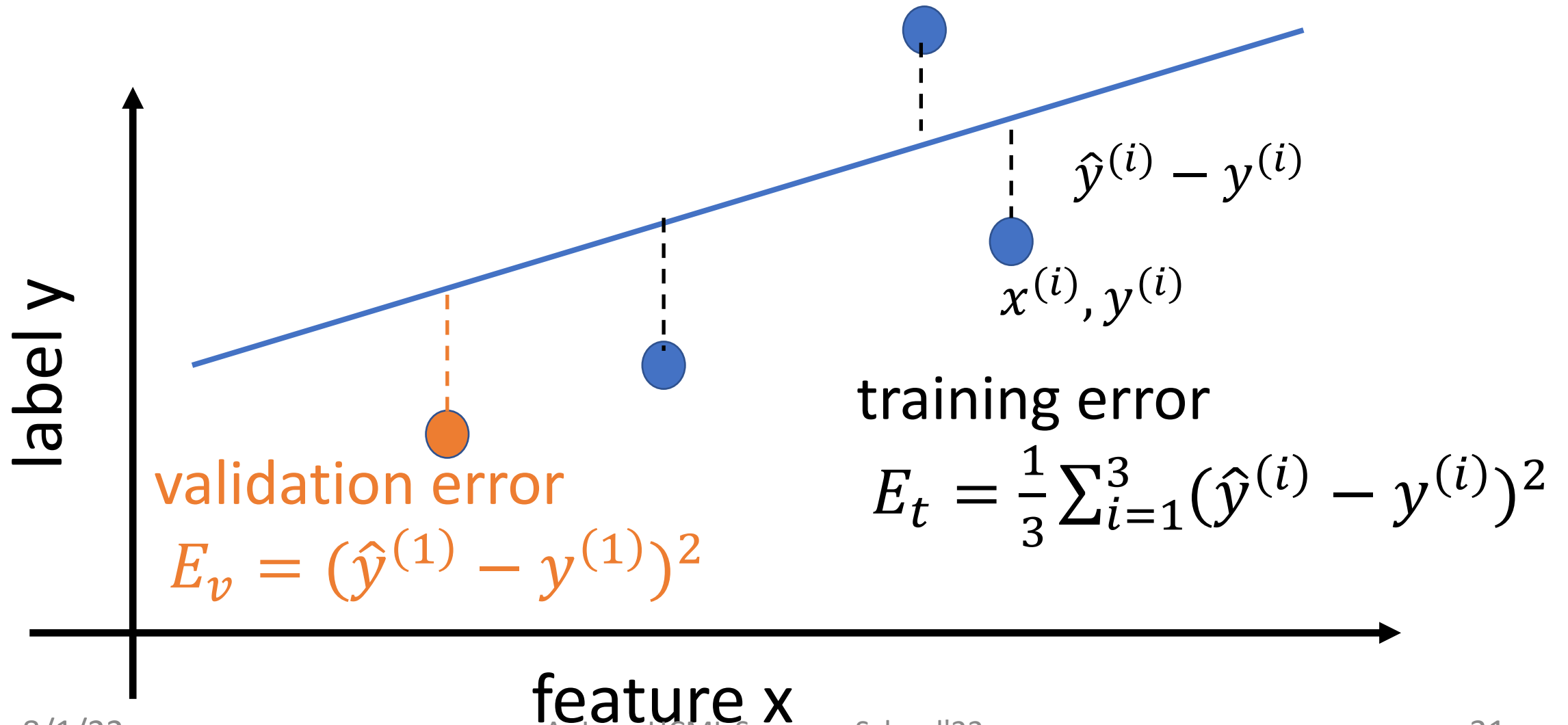- does not hold for training error

Model Validation

Basic Idea of Validation

- divide data points into two subsets

- use training set to learn predictor

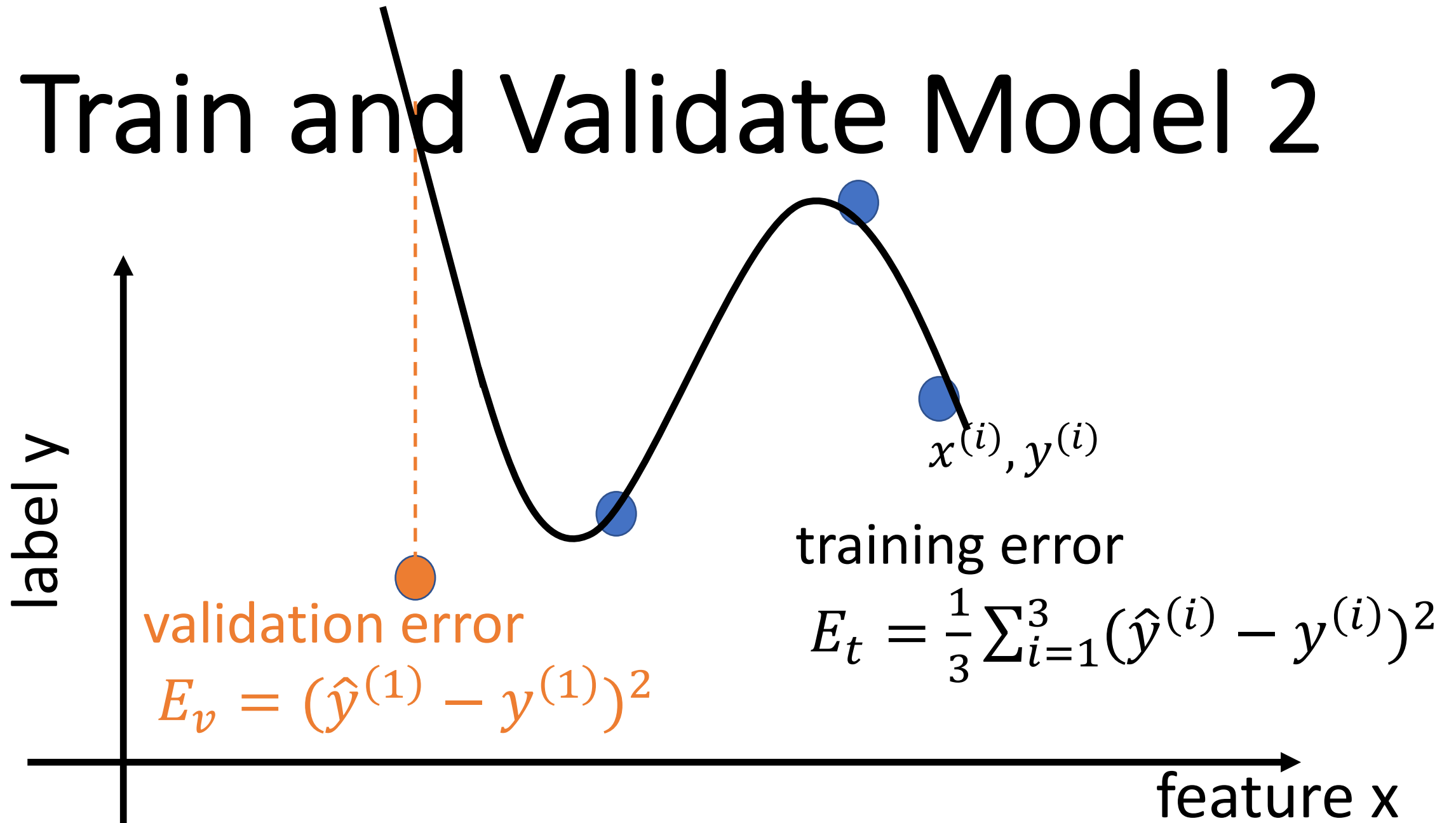- use validation set to estimate loss

# Split into Training and Validation Set

training set

validation set

label y

feature x

sklearn.model_selection.train_test_split

sklearn.model_selection. **train_test_split**(*arrays, **options)

Split arrays or matrices into random train and test subsets

Train and Validate Model 1

$$\hat{y}^{(i)} - y^{(i)}$$

$$x^{(i)}, y^{(i)}$$

training error

$$E_t = \frac{1}{3}\sum_{i=1}^{3}(\hat{y}^{(i)} - y^{(i)})^2$$

validation error

$$E_v = (\hat{y}^{(1)} - y^{(1)})^2$$

label y

feature x

Train and Validate Model 2

$x^{(i)}, y^{(i)}$

training error

$E_t = \frac{1}{3}\sum_{i=1}^{3}(\hat{y}^{(i)} - y^{(i)})^2$

validation error

$E_v = (\hat{y}^{(1)} - y^{(1)})^2$

label y

feature x

Basic Idea of Model Selection
choose model via validation error

training
error

validation
error

model 1: linear maps

model 2: degree 3 polyn.
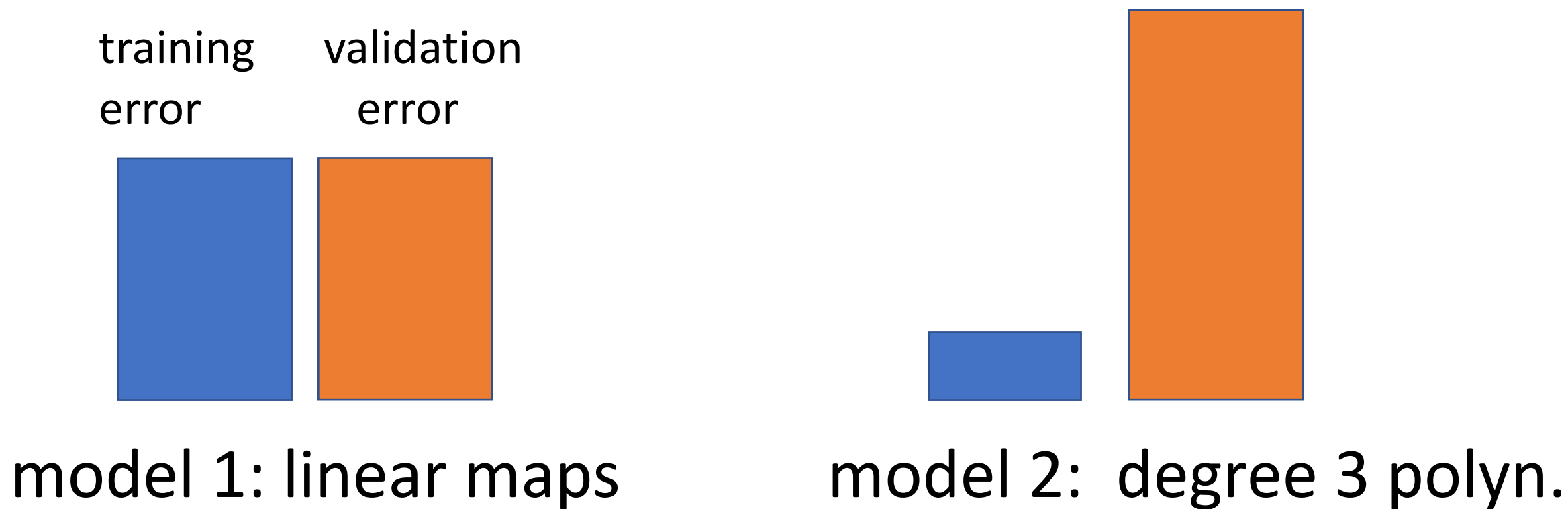
Train/Val Error vs Model Complexity

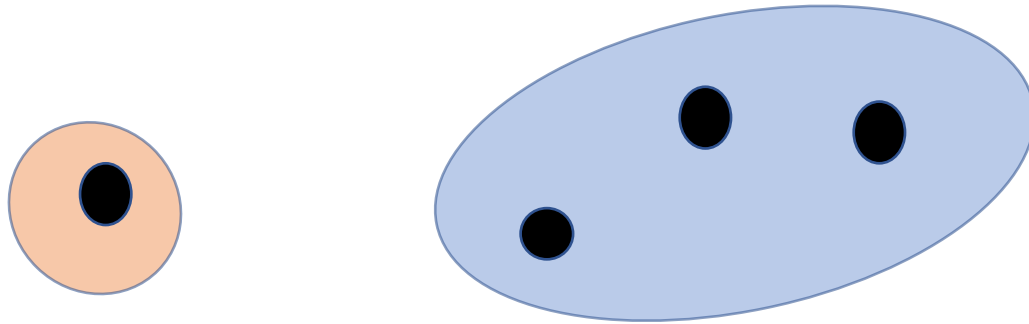$$\mathcal{H}^{(n)} = \left\{ h(x) = \sum_{l=0}^{n-1} w_l x^l \text{ with } \textcolor{red}{\text{weights } w_l} \right\}$$

model dimension/complexity n
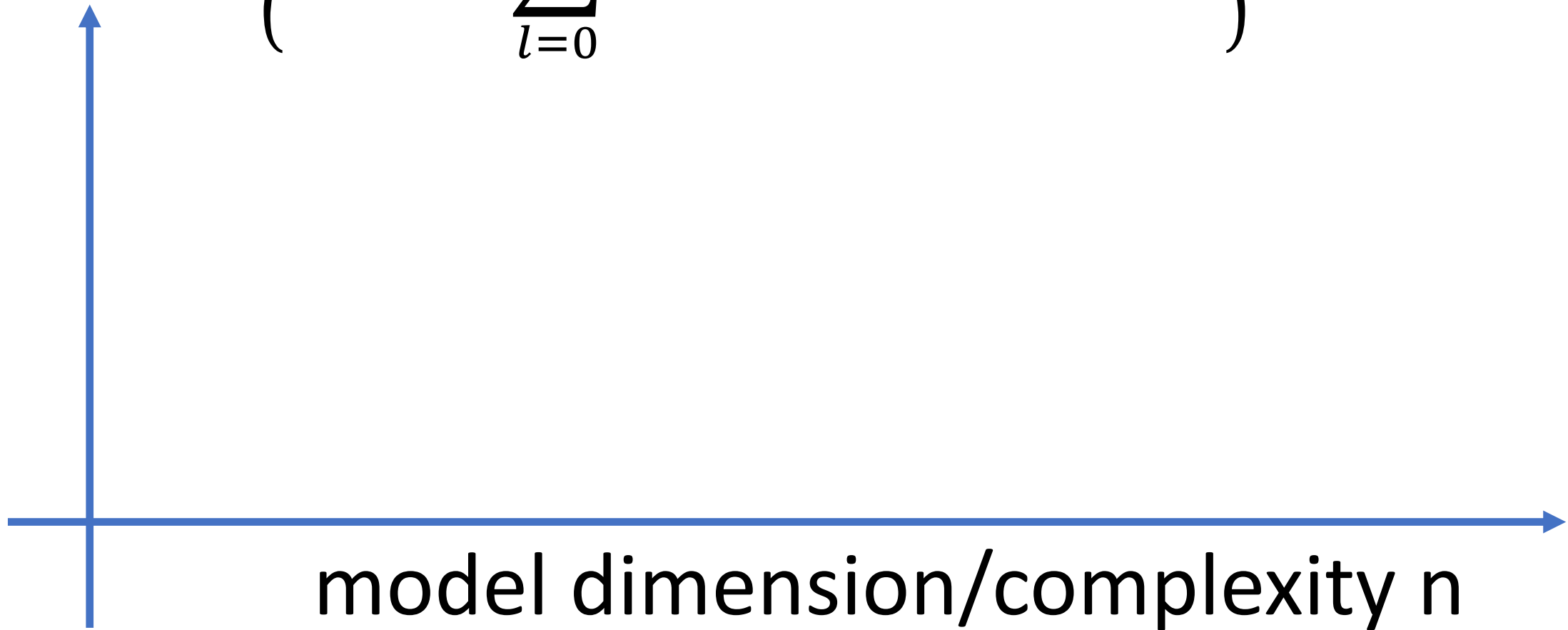
Unlucky Train/Val Split
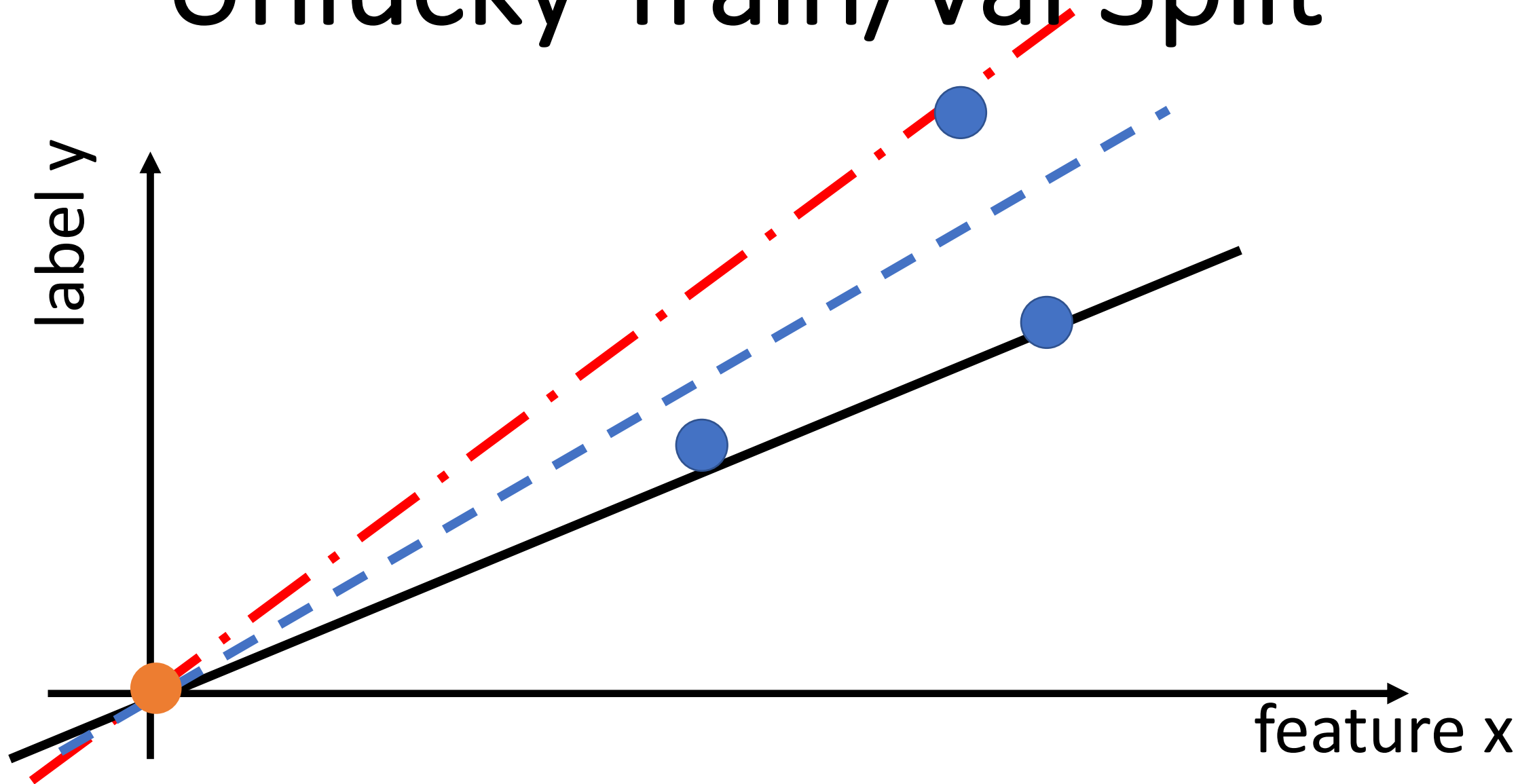
k-Fold Cross Validation

- might be unlucky with train/val split

- problematic for small datasets

- IDEA: randomly split several times

- "average out" unlucky splits

# K-Fold Cross Validation

fold 1
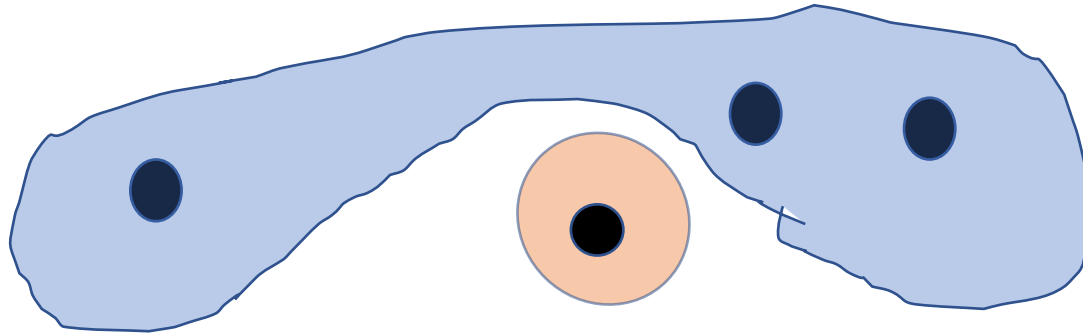
fold 2
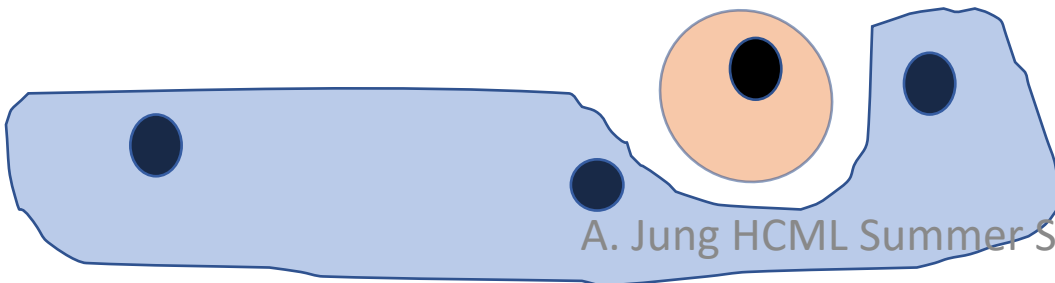
fold 3

k-Fold Cross Validation

how to choose nr of folds (the "k" in k-fold CV) ?

- train fold should be sufficiently large (avoid overfitting)

- val folds should sufficiently large (to get reliable estimate of generalization)

CAUTION!

• k-fold CV requires a method to split into folds

• most basic method: evenly divide into k folds

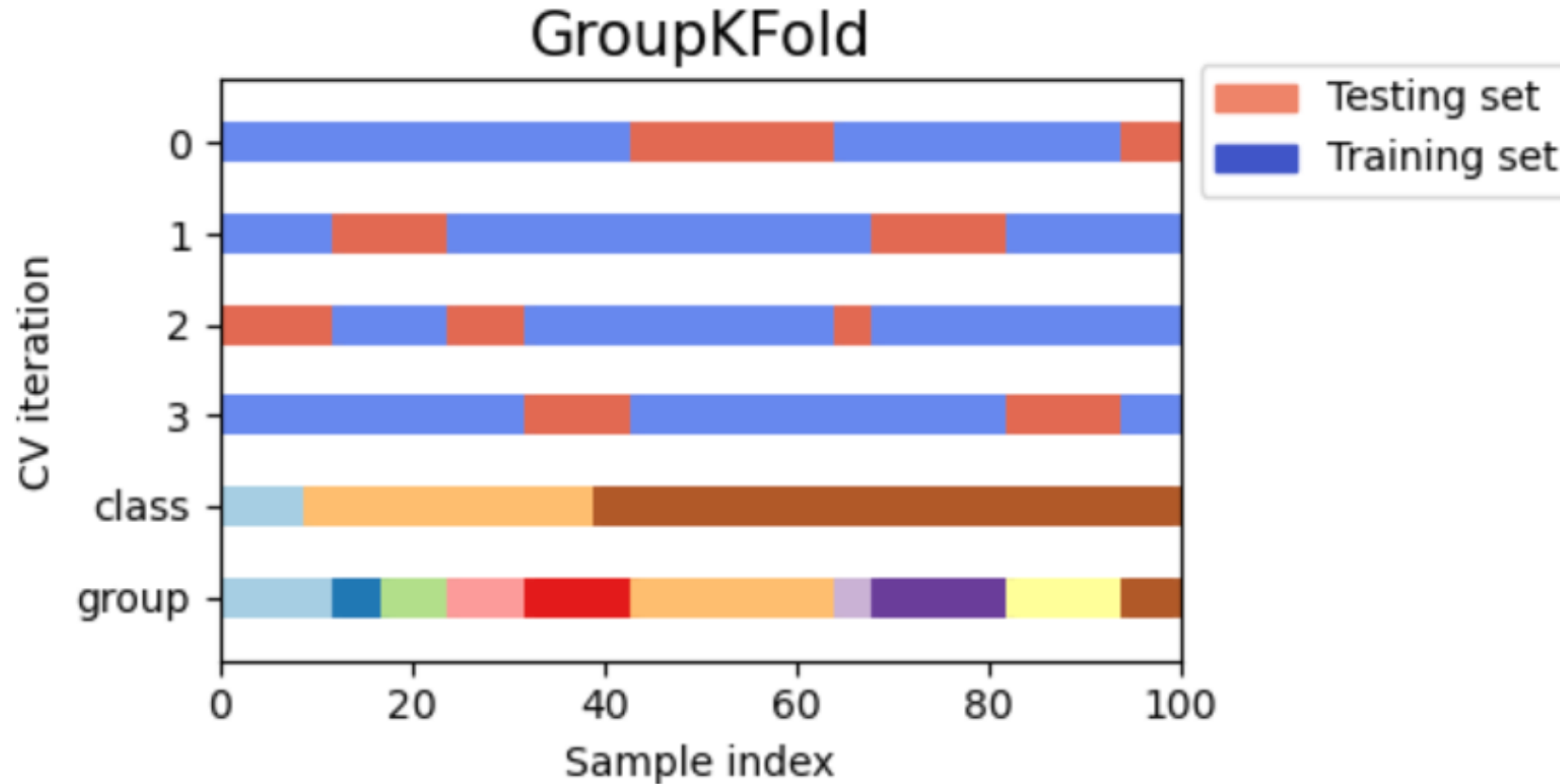• works if data is i.i.d. ("order of data points is arbitrary")

• fails if data points are grouped or ordered

Imbalanced Classes and Group Structure
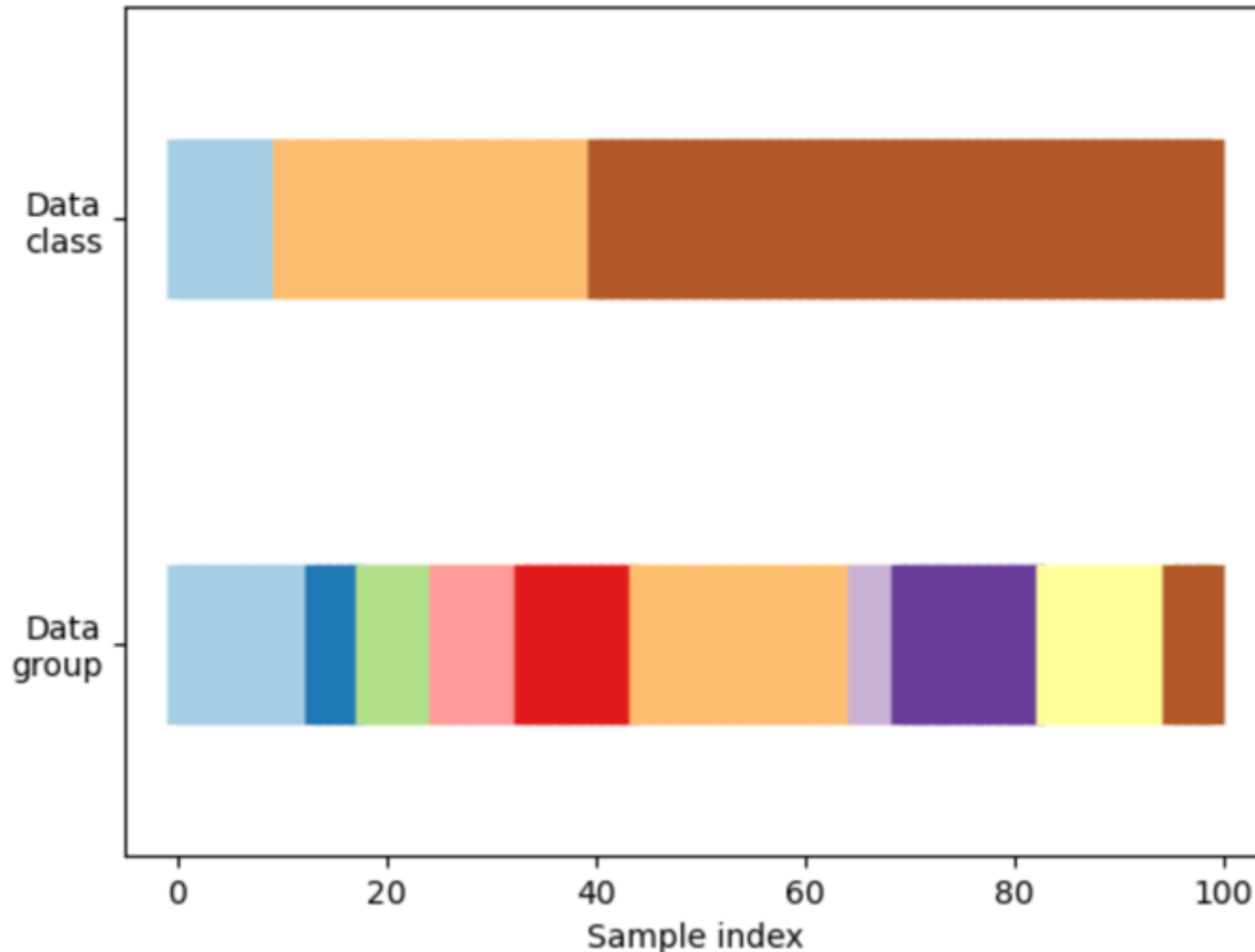
- e.g. data points with same label are contiguous blocks

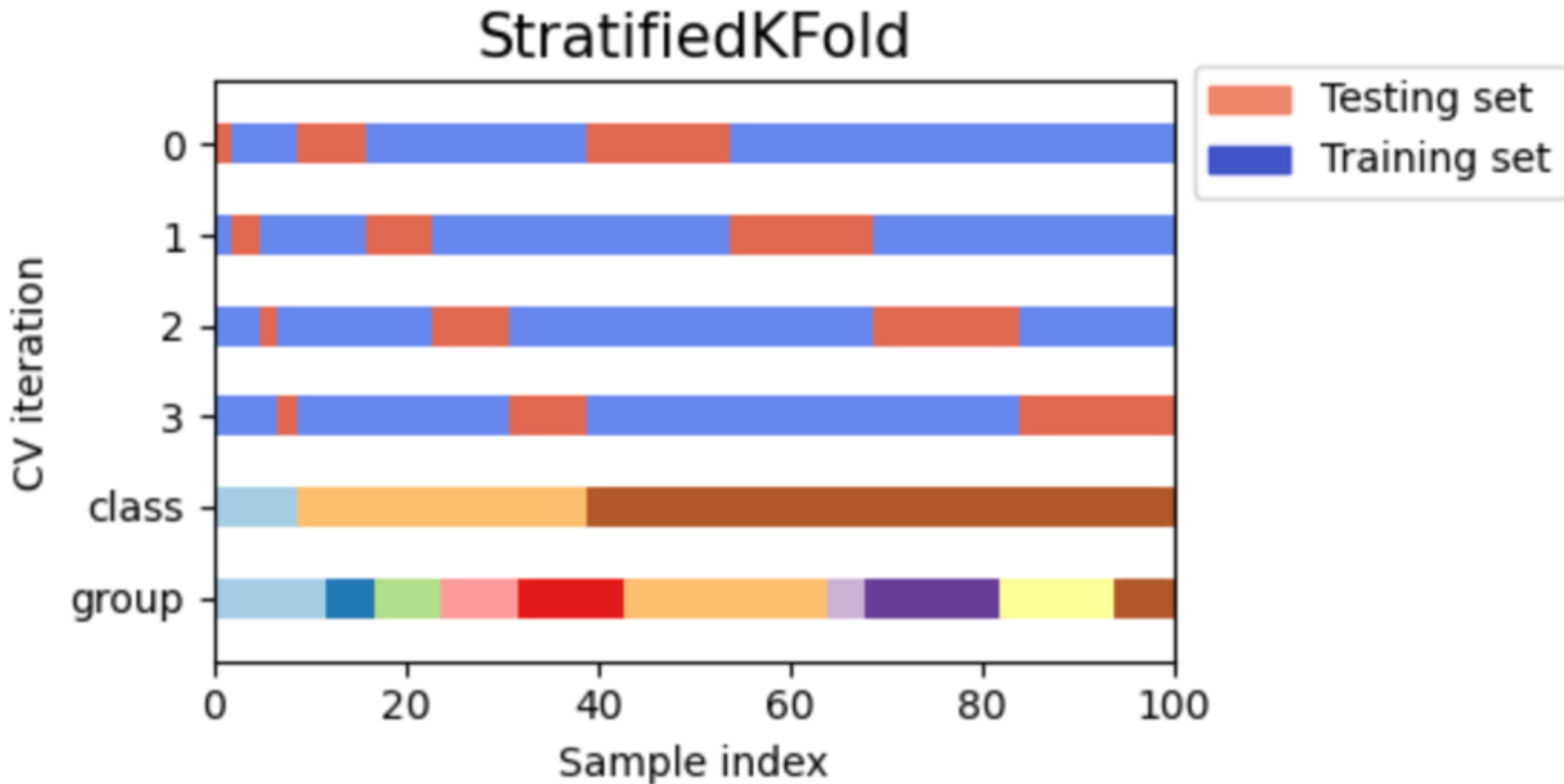- or data points are obtained at consecutive time instants ($\rightarrow$ correlations)
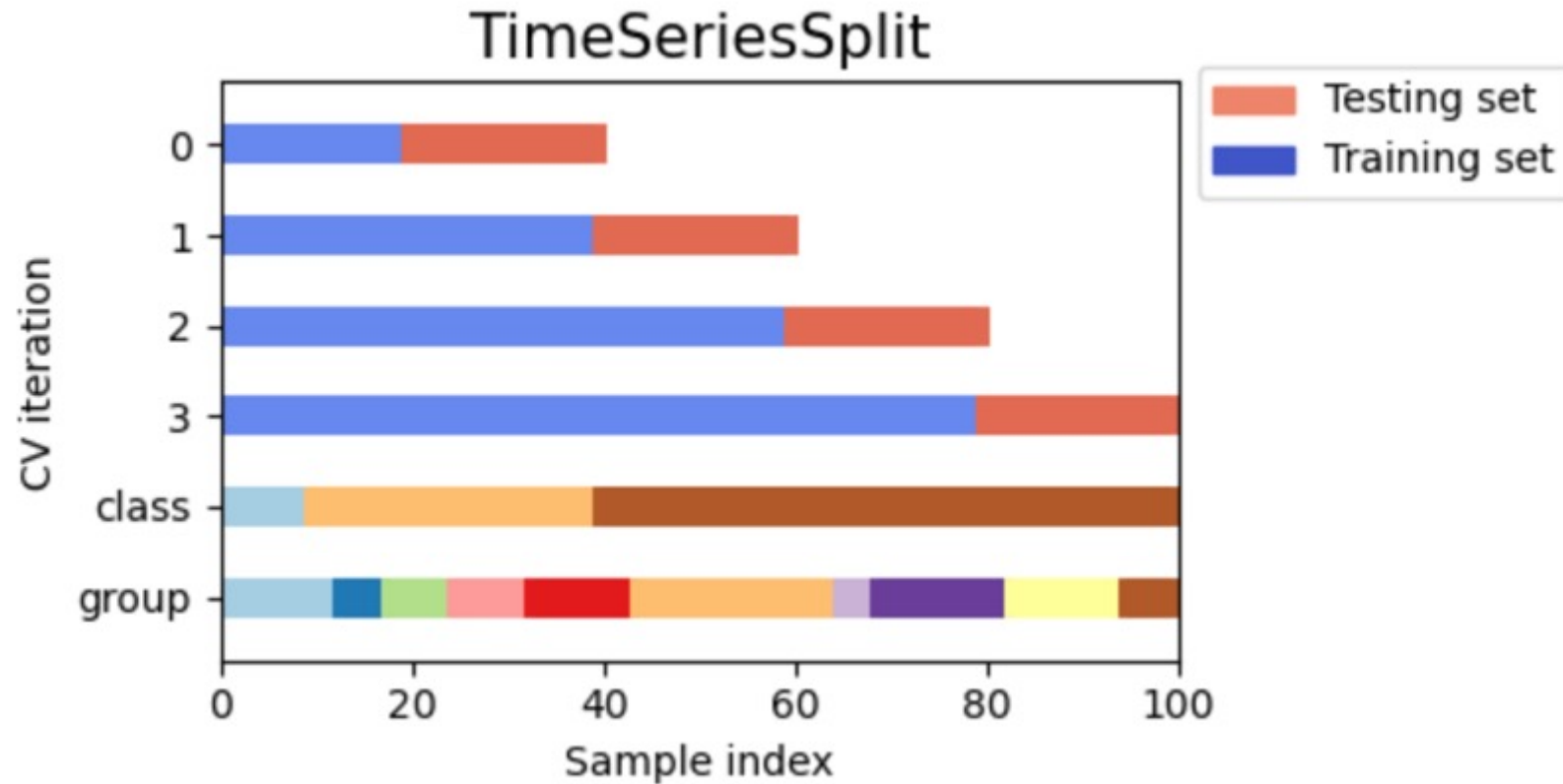
Group-Preserving Splitting

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html

Class-Ratio Preserving Splitting
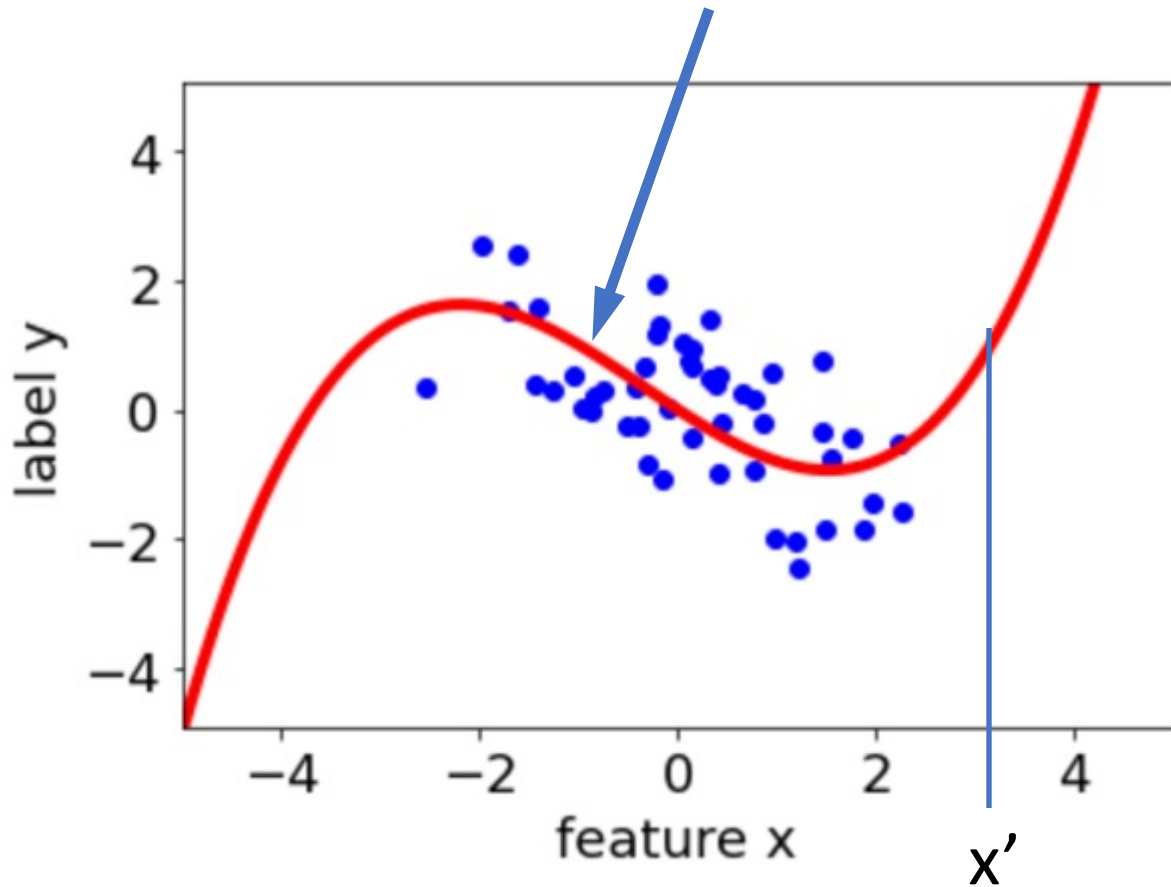

StratifiedKFold

Temporal Successive Splitting

source: https://scikit-learn.org/stable/

Bias and Variance Decomposition

A. Jung HCML Summer School'22

"Bias" error component due to model being too small

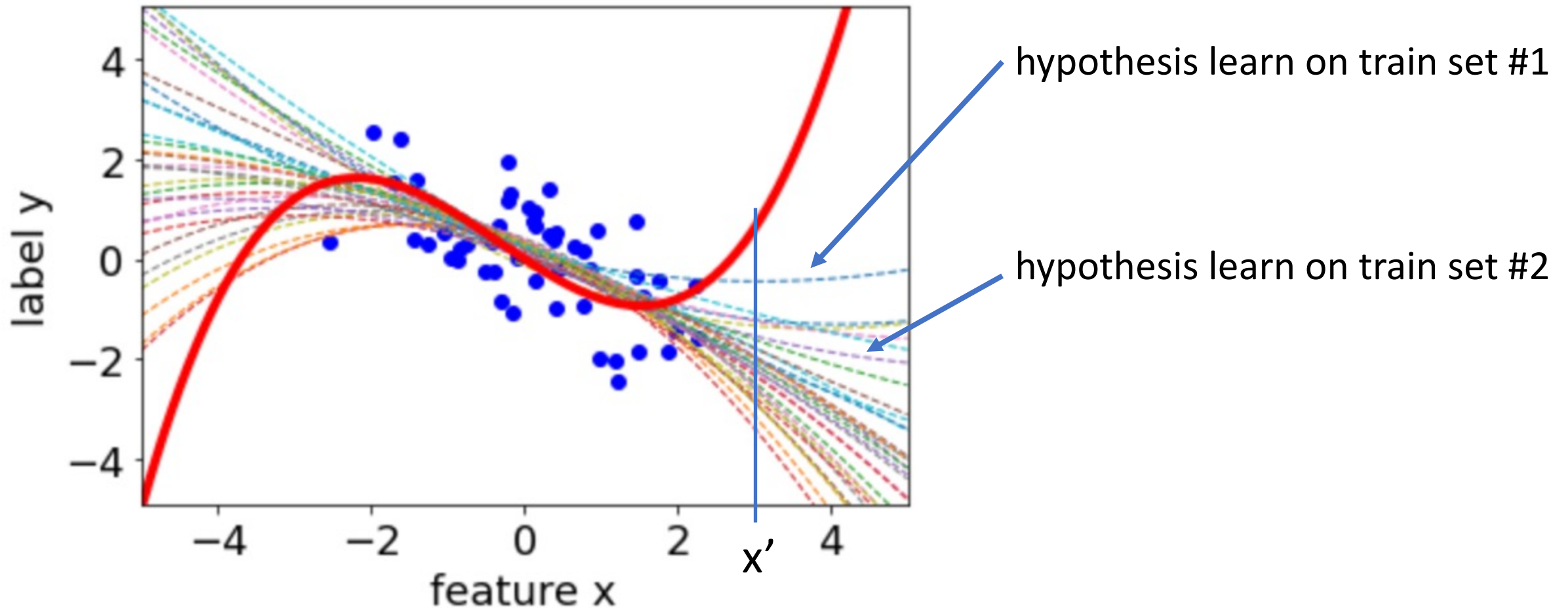# "Variance" reflects error due to dataset being too small

Toy Data

$$y = g(x) + \text{"noise"}$$

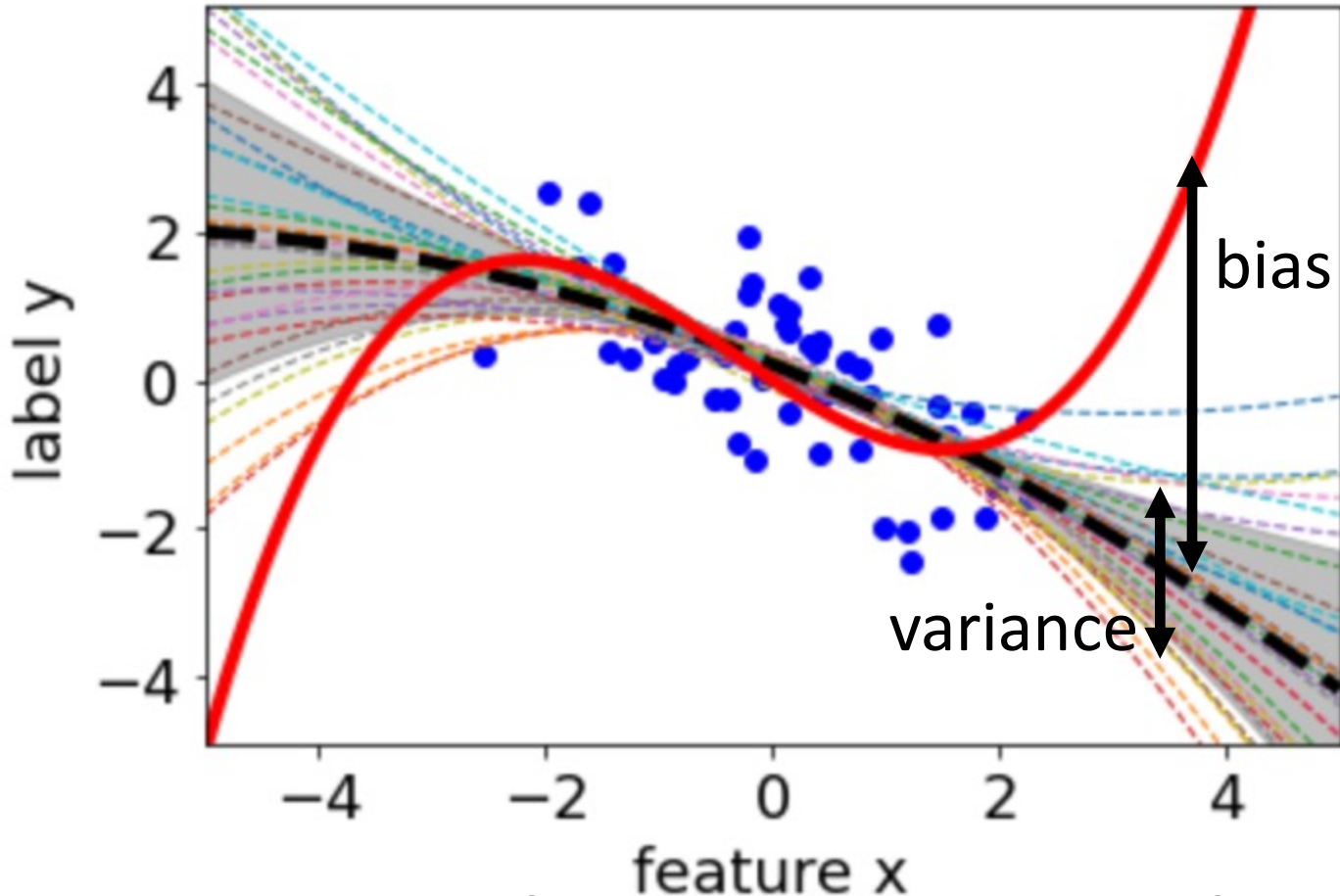learn hypothesis h(.) using a randomly selected training set

compute prediction h(x') for a fixed feature value x'
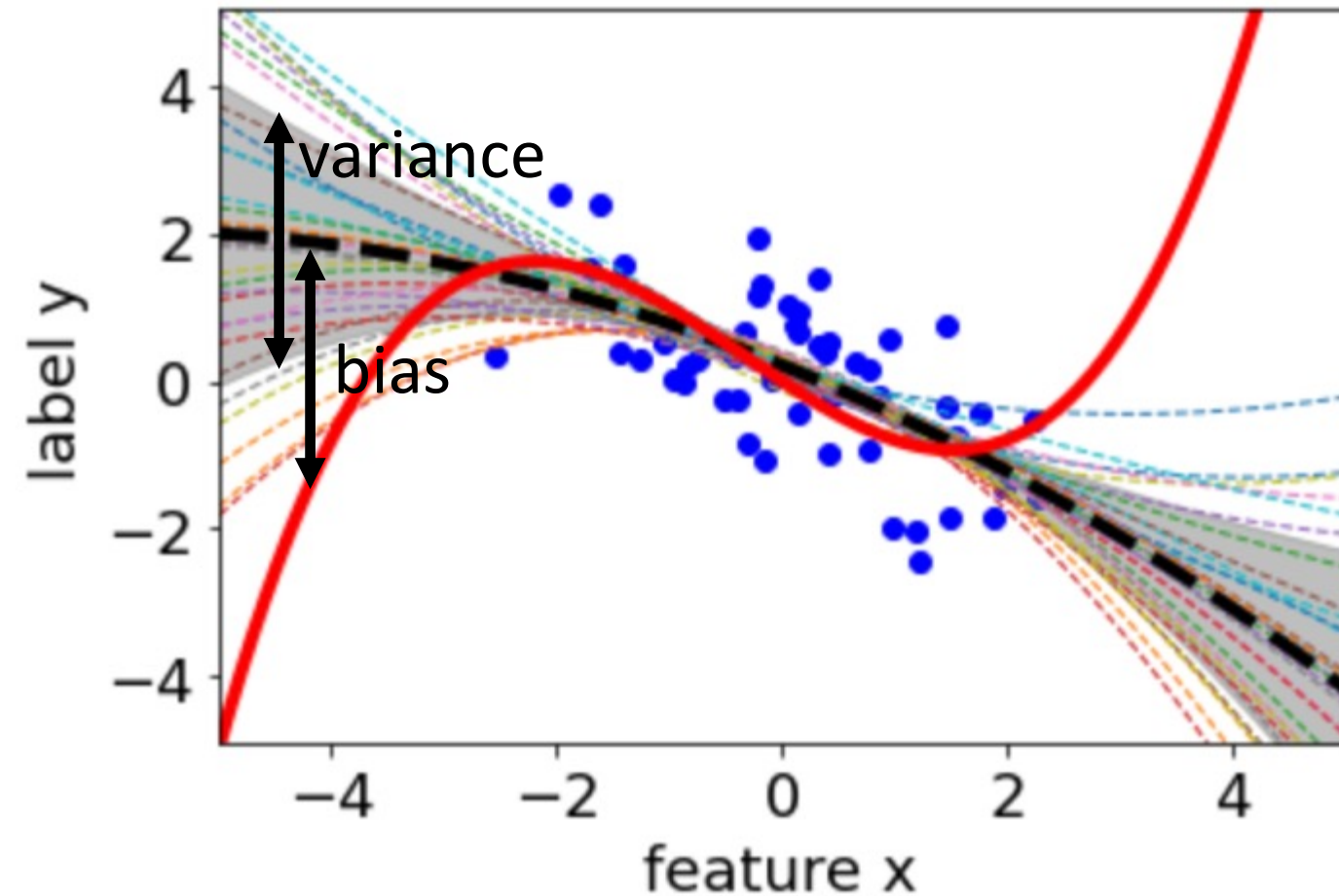
Ensemble of Learnt Hypotheses

hypothesis learn on train set #1

hypothesis learn on train set #2

Bias and Variance

$$\hat{y} = h(x')$$

RV since obtained from a randomly selected training set

$$\mathsf{E}\{(\hat{y} - y)^2\} = (\mathsf{E}\{\hat{y}\} - y)^2 + \mathsf{E}\{(\hat{y} - \mathsf{E}\{\hat{y}\})^2\}$$
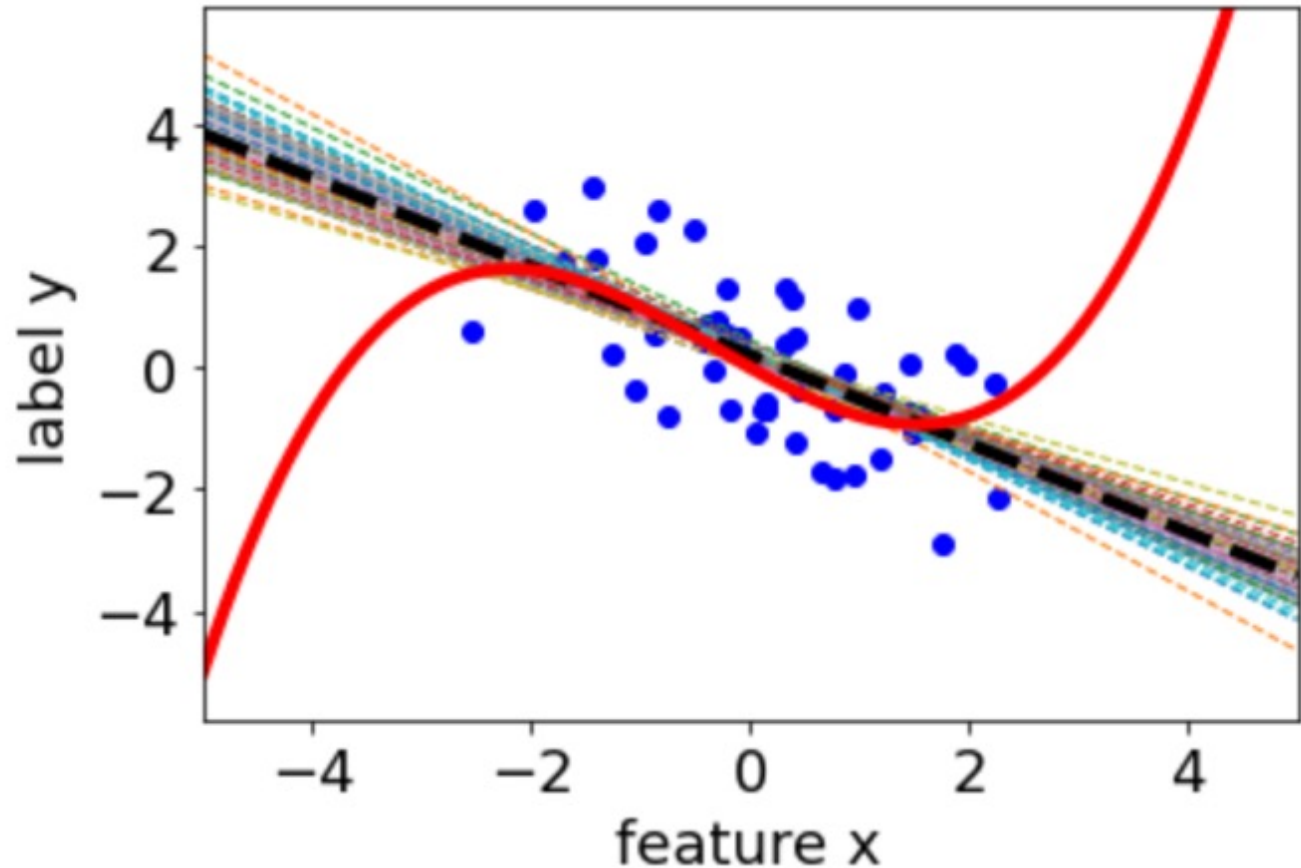
Bias and Variance Tradeoff

"Prediction Error = Bias + Variance"

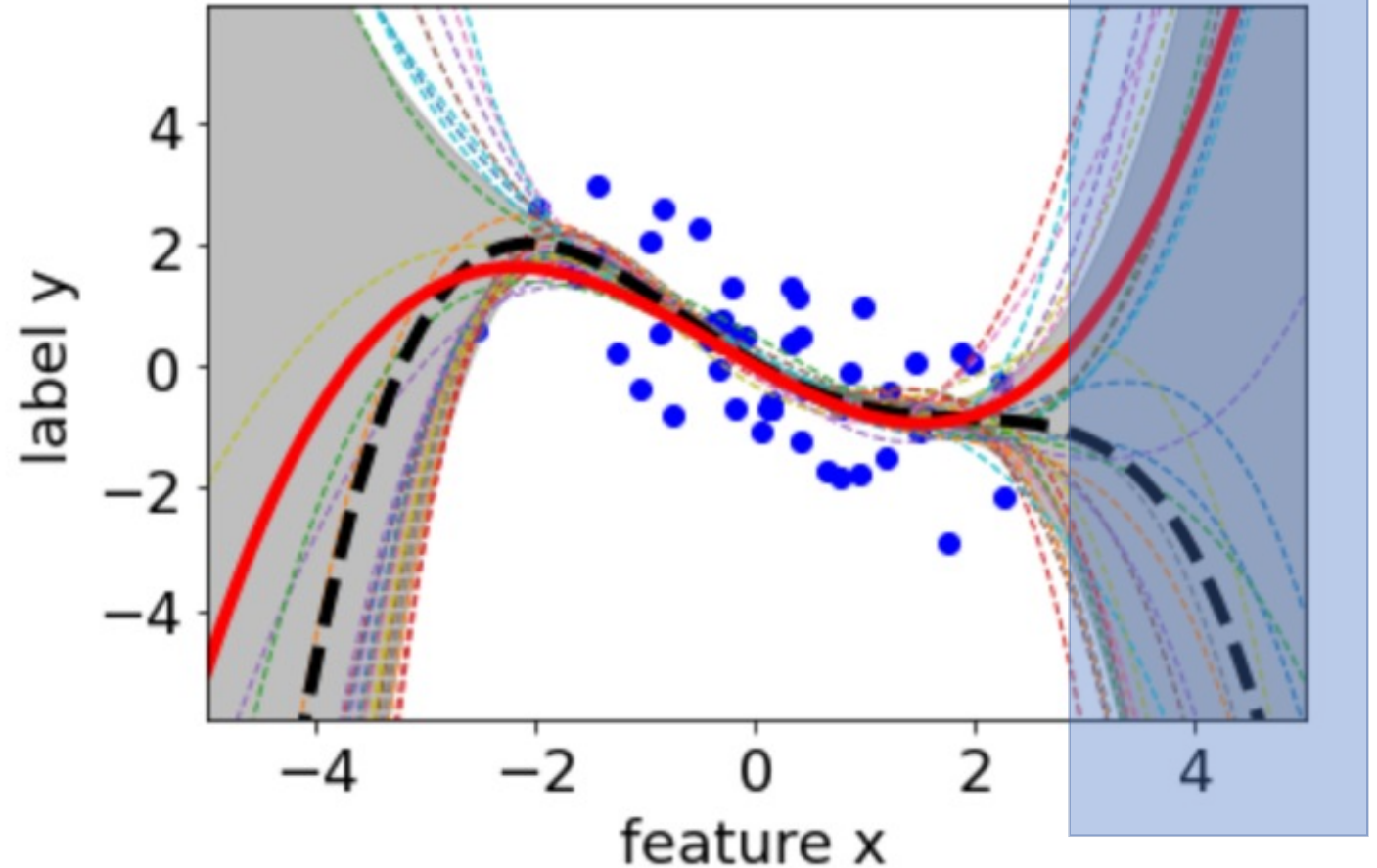bias reduction typically incurs variance increase and vice versa

Smaller Model (Poly.Degree)
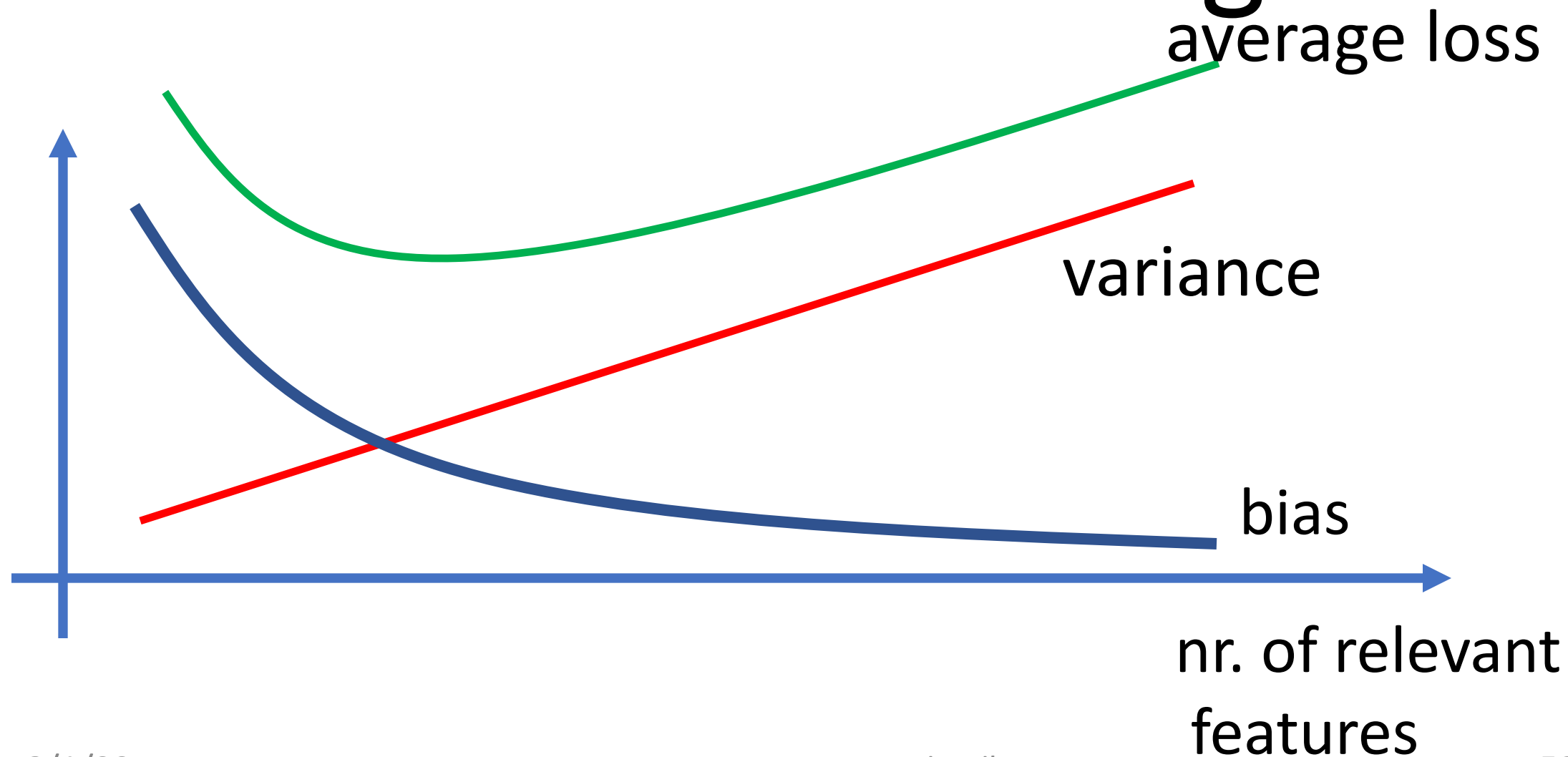
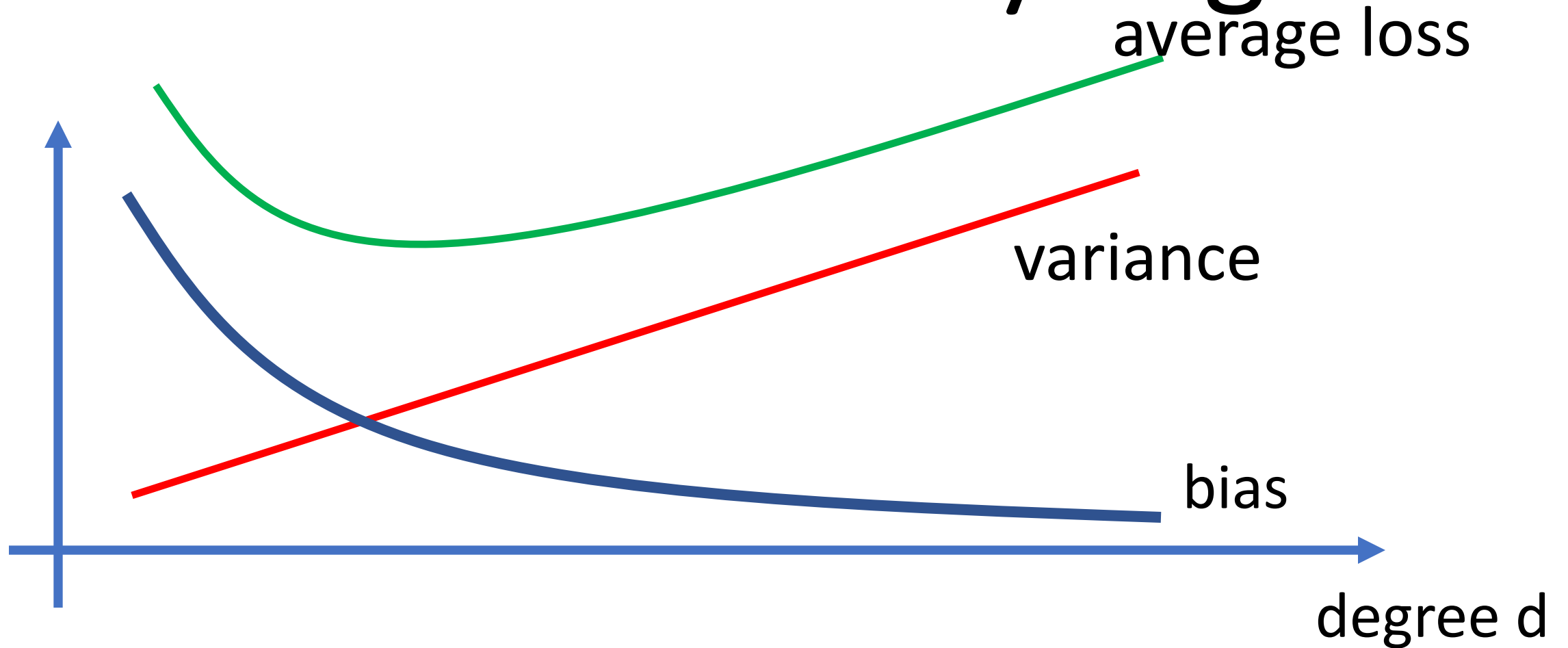- small variance

- large bias

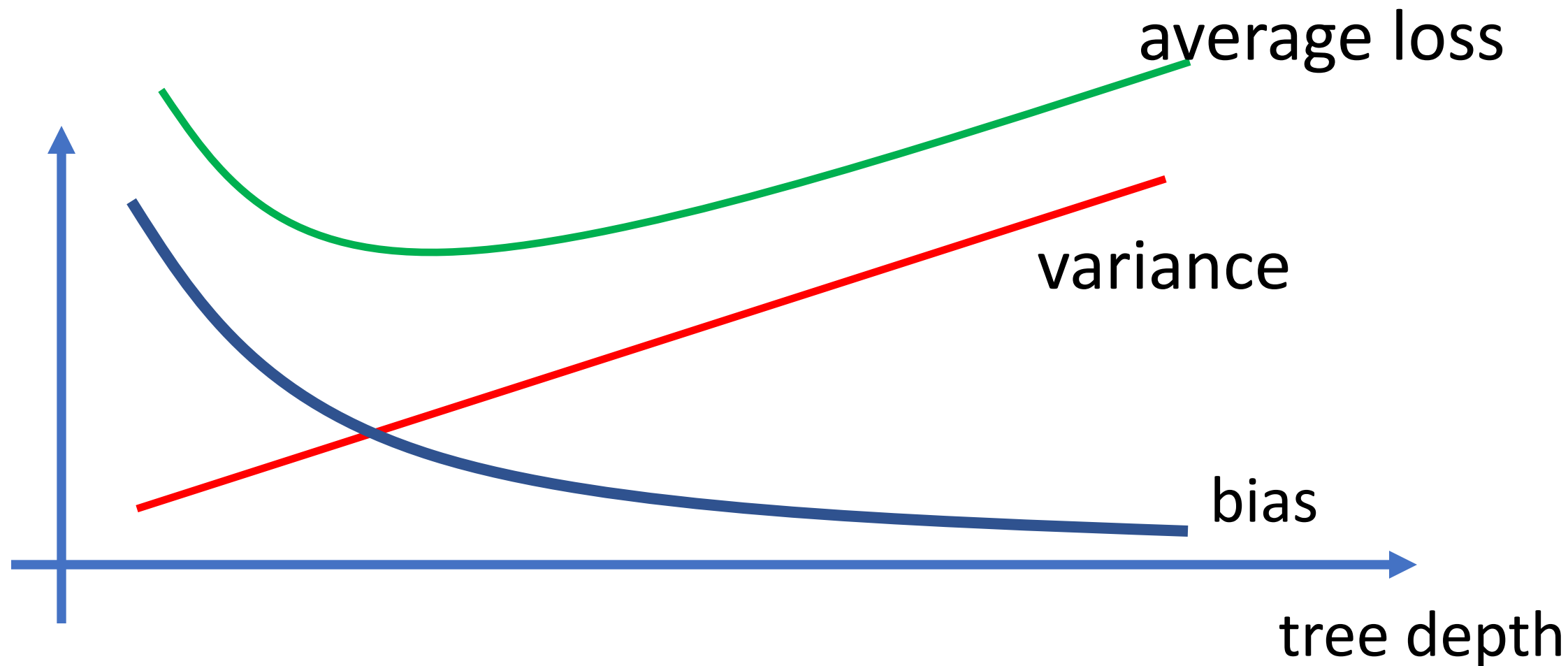# Larger Model (Poly. Degree)

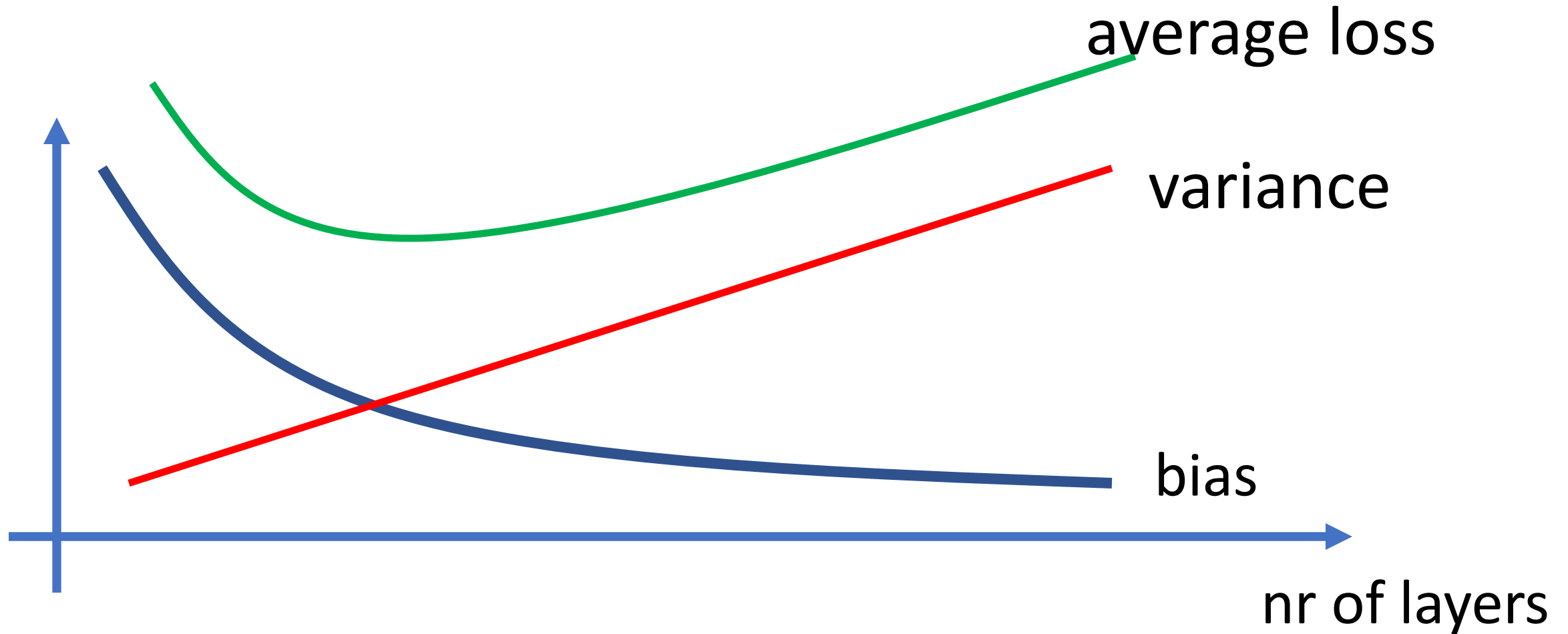- large variance

- small bias

Bias vs. Variance Lin.Reg.

A. Jung HCML Summer School'22

Bias vs. Variance Poly.Reg.

average loss

variance

bias

degree d

Bias vs. Variance Dec. Tree.

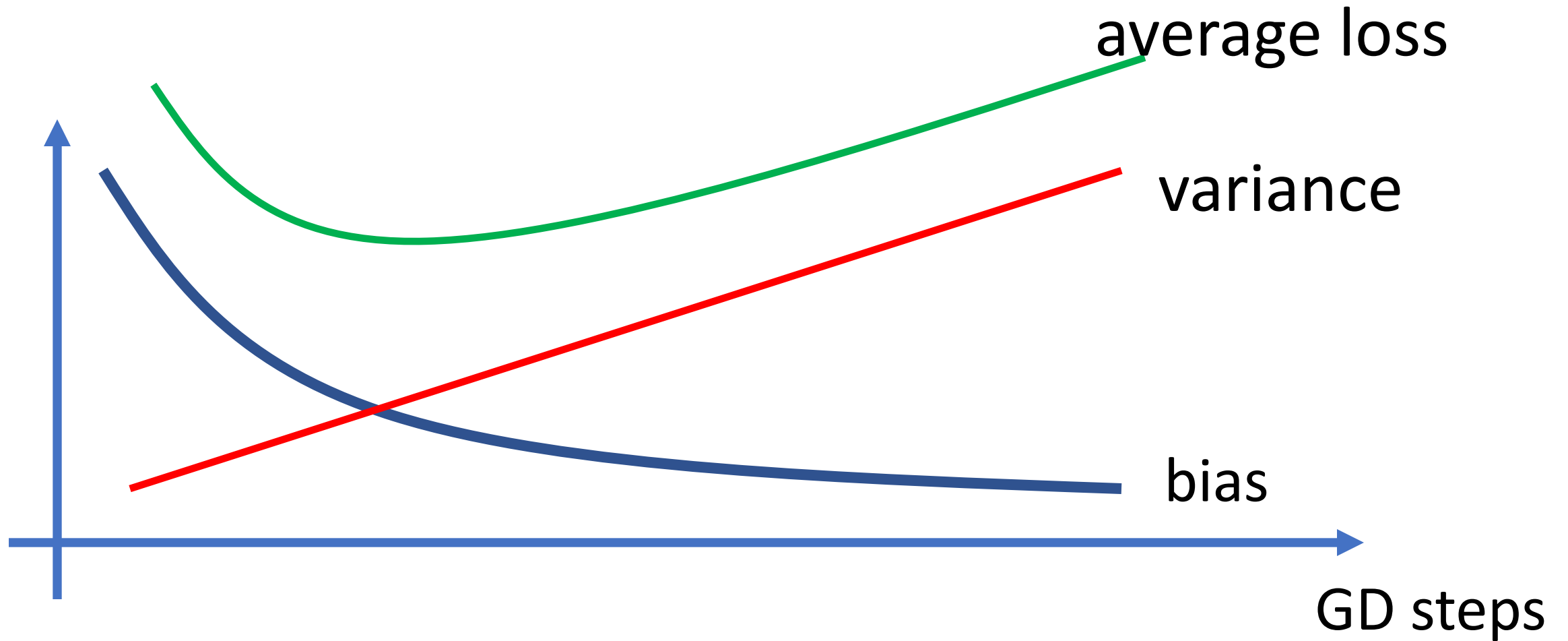average loss

variance

bias

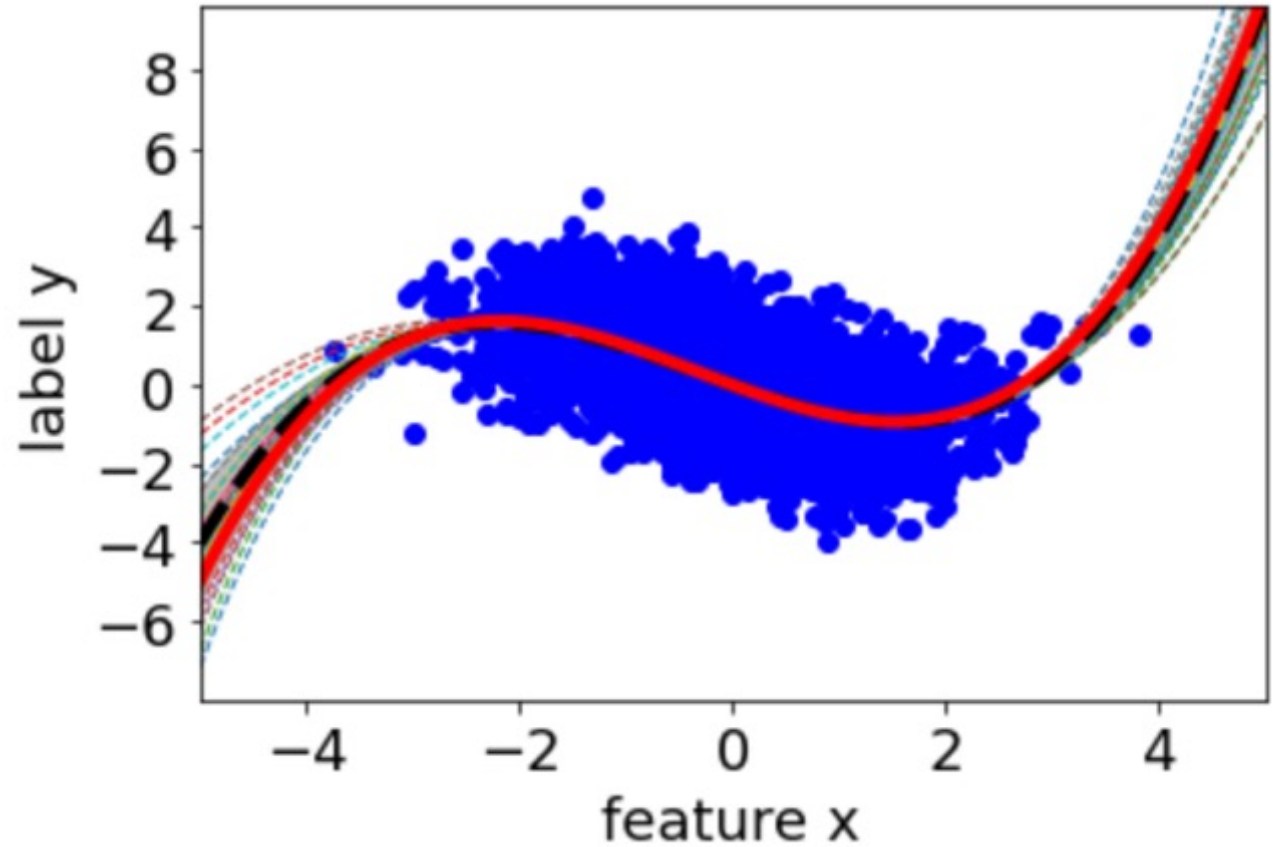tree depth

Bias vs. Variance Deep Learning
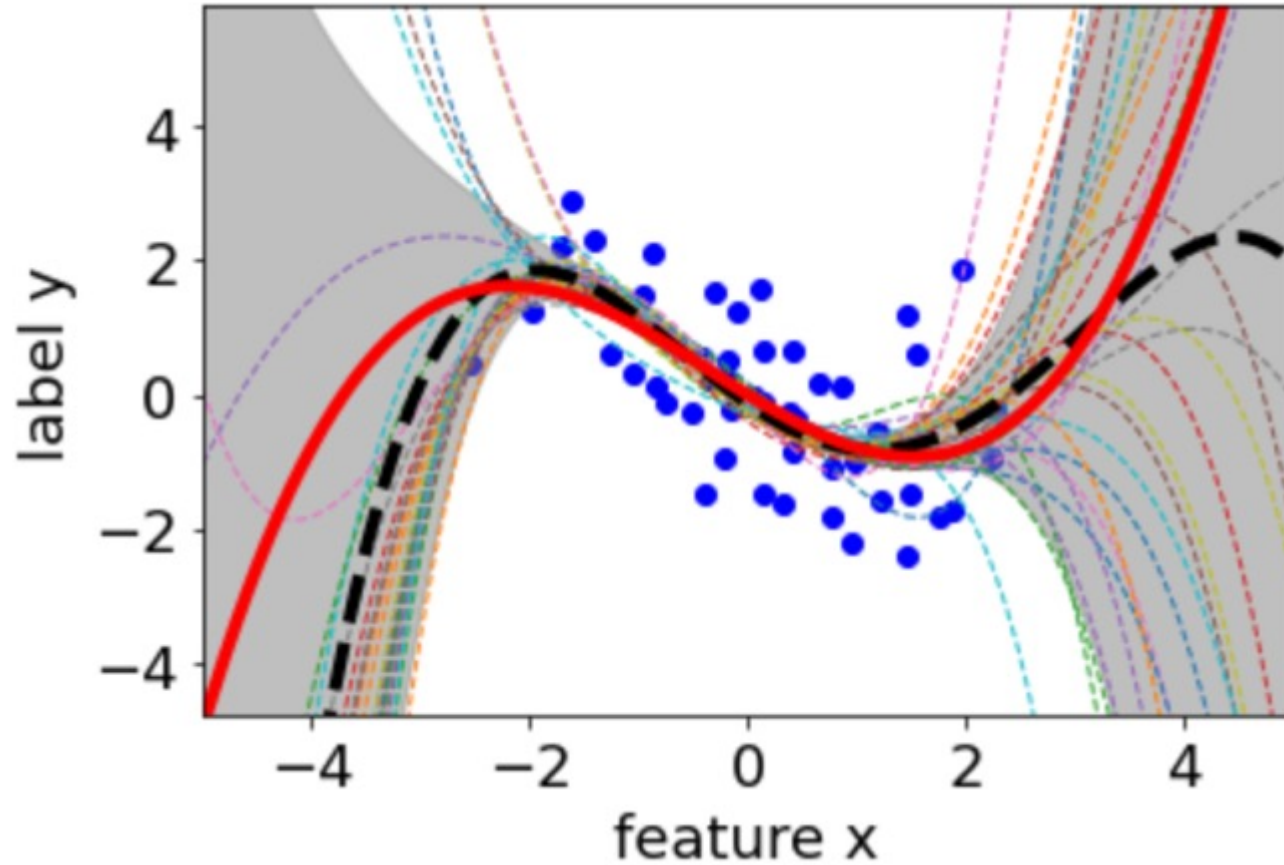
Bias vs. Variance Grad. Desc.
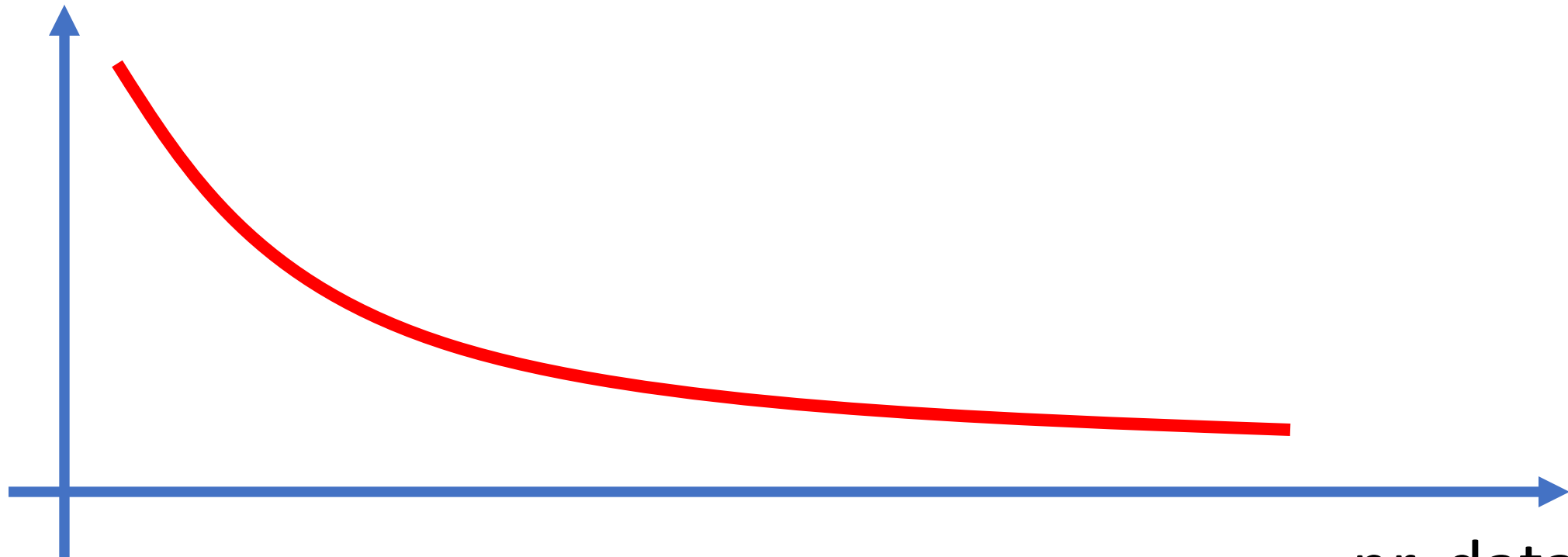
More Data

-> smaller variance

Less Data

-> larger variance

Learning Curve

variance

nr. data points

Alex' Rule of Thumb

effective number of training data points

\>

10 * nr. tunable effective model parameters

stretch the term "effective" as much as possible !

ML Diagnosis

A. Jung HCML Summer School'22

Simple Recipe

- consider ML method with some hypothesis space

- learn hypothesis by min. average loss on train.set

- training error = average loss of learnt hypothesis

- compute validation error

- compare val err, train err with a baseline

Benchmark/Baseline

could be obtained from

- probabilistic models

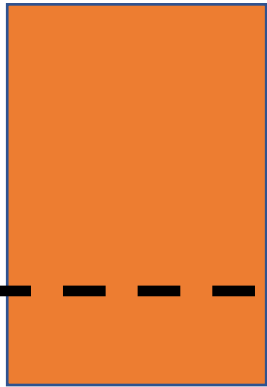- domain expertise

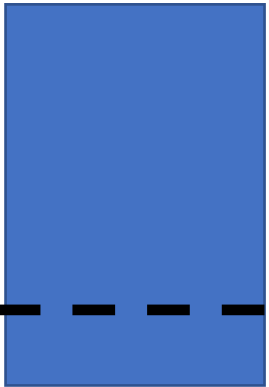- existing ML methods

- human performance

- …

training error   validation error

- small train error -> hypothesis space is large
- large val err -> overfitting

- Workaround ?
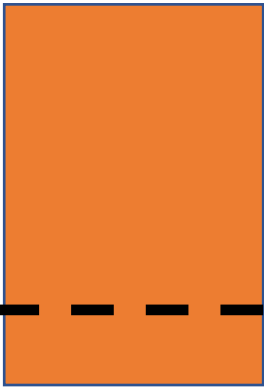
benchmark

training error validation error

- large train error -> no good hypothesis found
- Workaround ?

training
error

validation
error

• Case Solved !

Take Home Messages

- large models (e.g. deep nets) often overfit

- small training error does not mean much!

- diagnosis by comparing train/val err

- bias/variance analysis can guide model improvement

Thank You !