Lecture 9: Cross-validation

Monday, Oct

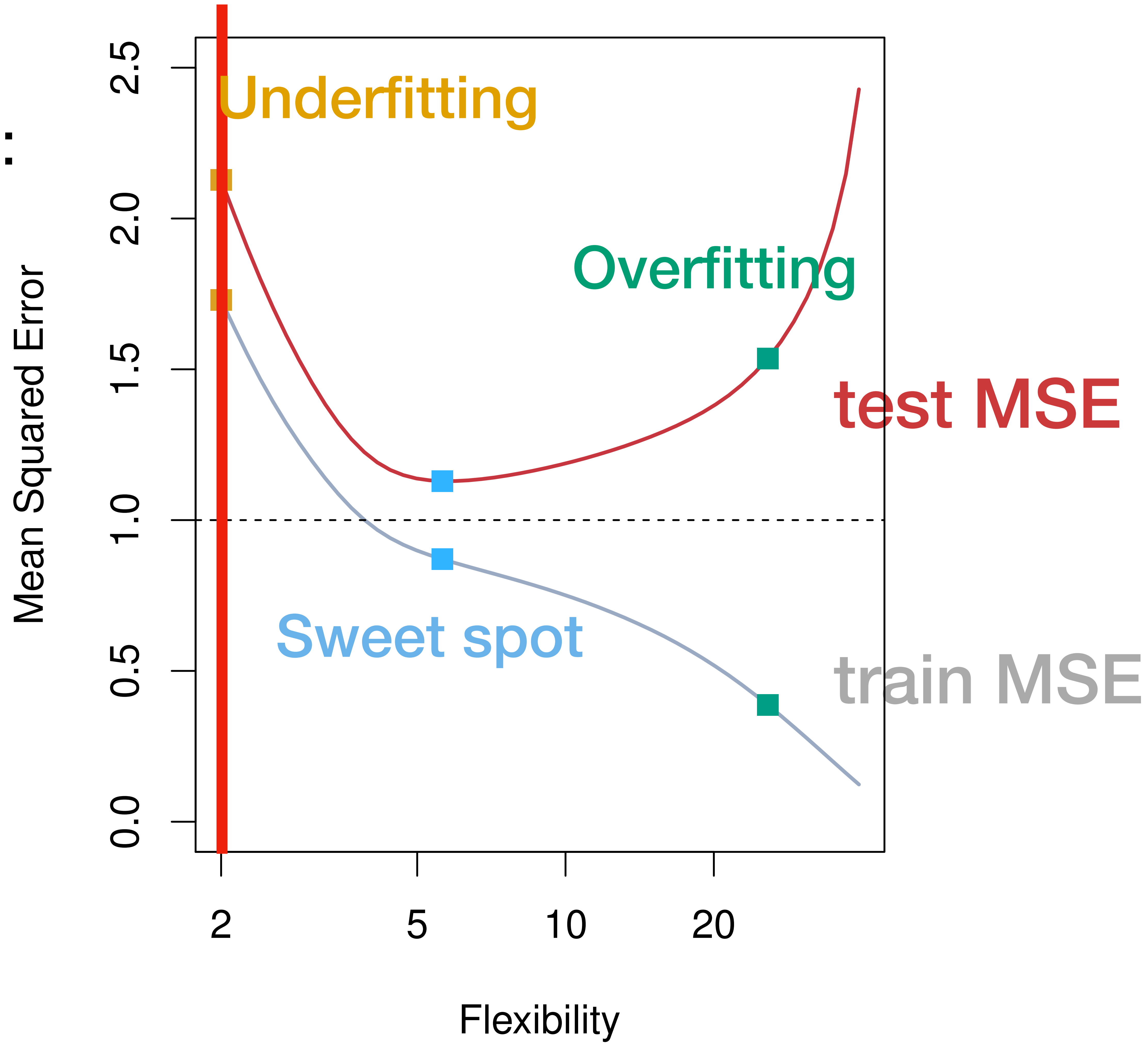
Model assessment vs. selection

Model assessment: evaluating a model's performance on the **test set** via metrics such as:

- RSE and (R)MSE (regression)
- Classification error rate (classification)
- R^2 and Adjusted R^2

Model selection / comparison: comparing the performance between **competing models**

- **Example:** selecting the proper level of **flexibility** in a model (e.g., number of parameters)



Resampling

- **Resampling:** fitting the same model multiple times using different subsets (samples) of the original dataset

- **Two methods:**
  - **Cross-validation:** resampling the data to estimate the model's test error
  - **Bootstrapping:** resampling the data to estimate the uncertainty around population statistics (e.g., standard errors and confidence intervals)

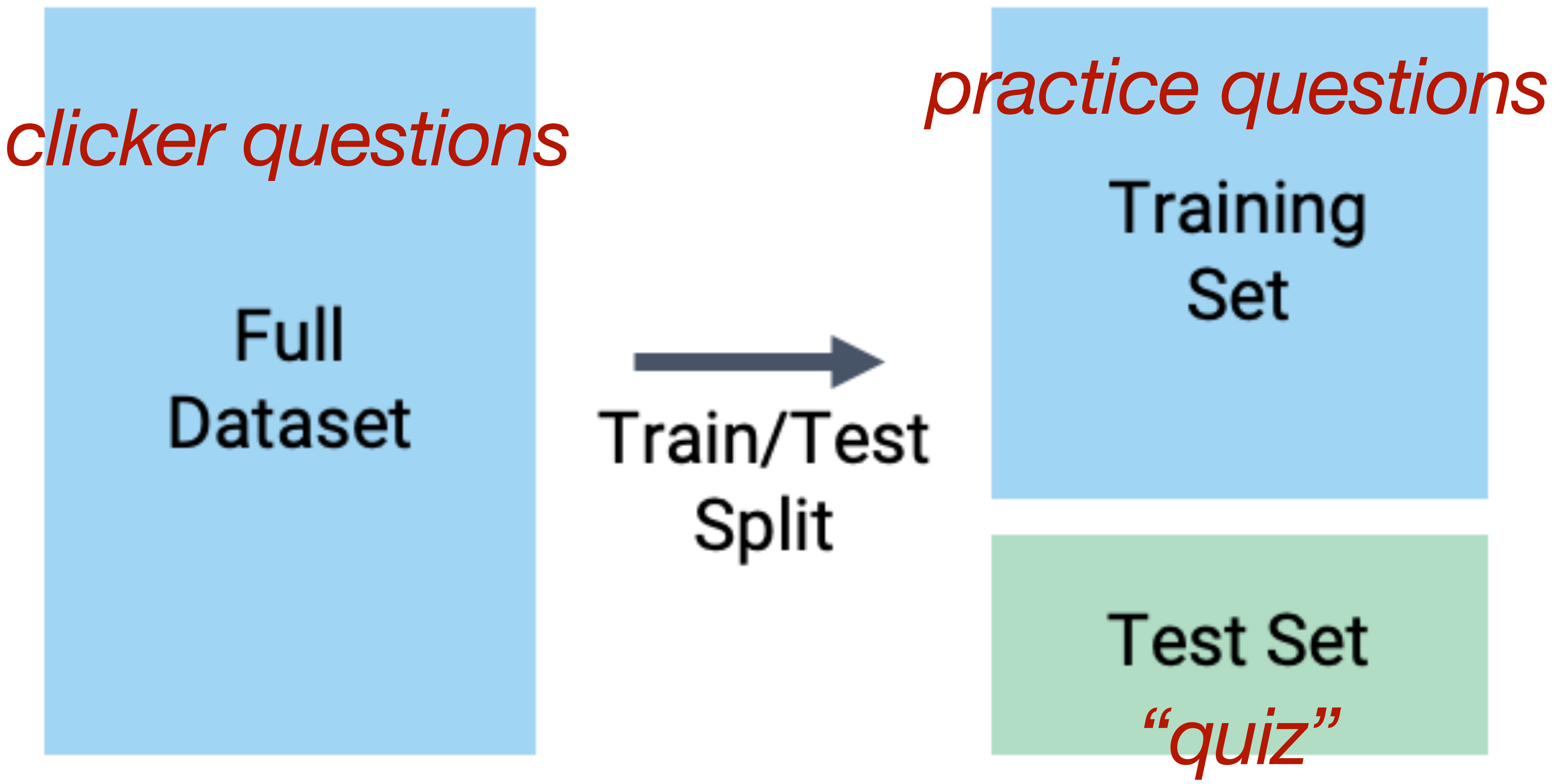
$\Rightarrow$ Both are used for model assessment (evaluating model performance) and selection (choosing the best model)

Train, test, and validation set

How do we pick the best model?

- **So far:** split data into train set (to train model) and test set (evaluate model performance)
- **However,** if we use the test set to fine tune and choose the best model, (e.g., by trying out models w/ different # parameters and comparing them) it is ***no longer a true test set.***
- **Why?** A test set is supposed to be an unbiased evaluation of a final model's fit.
  We don't want test performance to influence model choice!

(imperfect) analogy

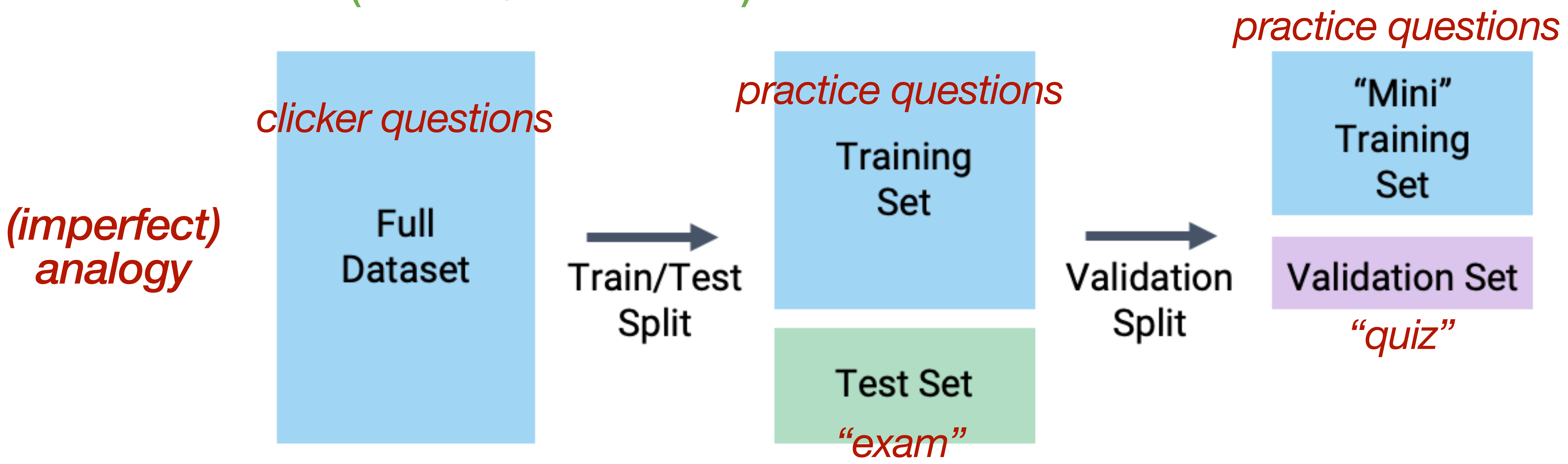


The test set must reflect ***truly unseen data*** ("exam") or else the test error becomes ***biased!***

Train, test, and validation set

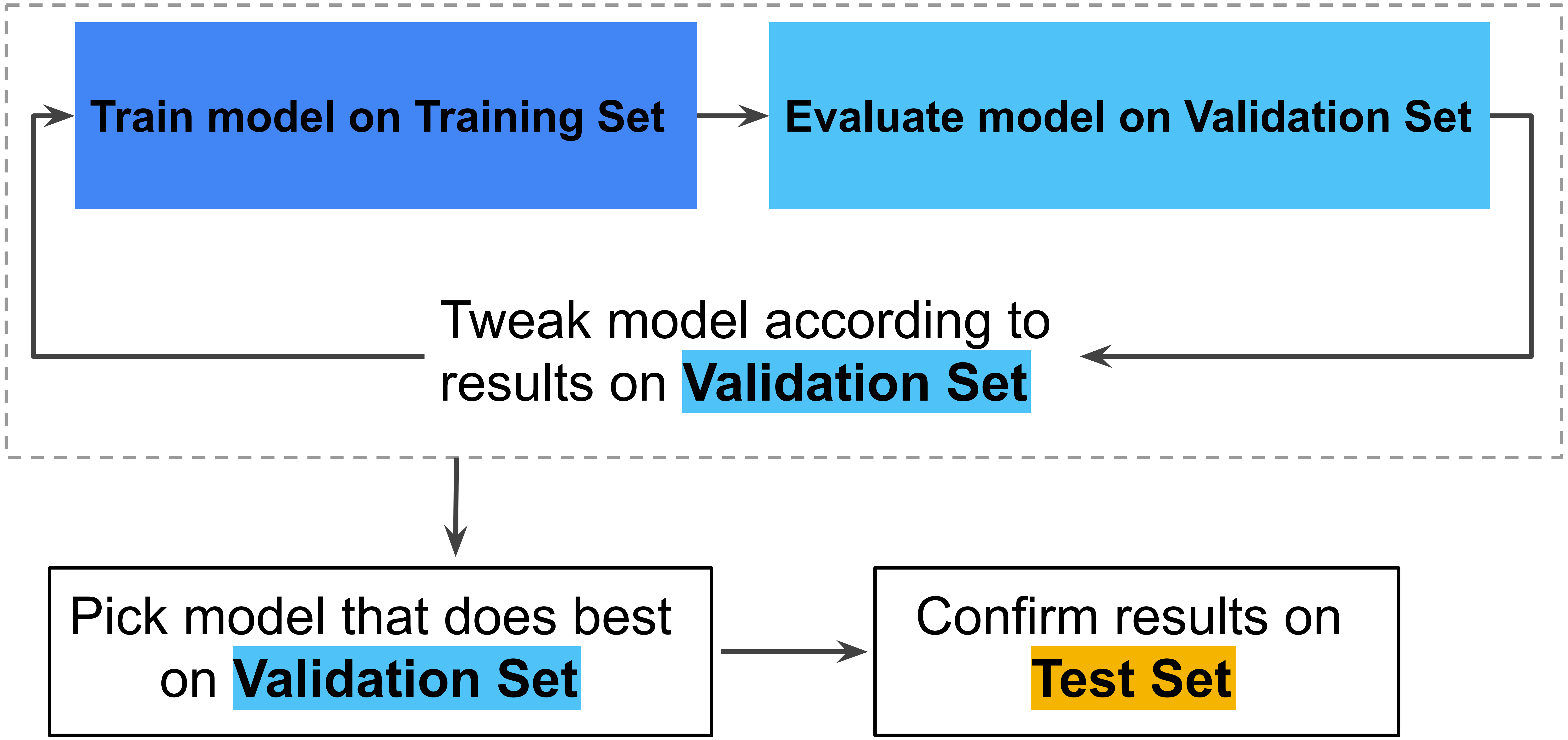
How do we pick the best model?

- **Solution:** split the dataset into:
  1. training set (60-80% of full dataset)
  2. validation set (~10-20% of full dataset) (used as an estimate of the test error)
  3. test set (~10-20% of dataset)



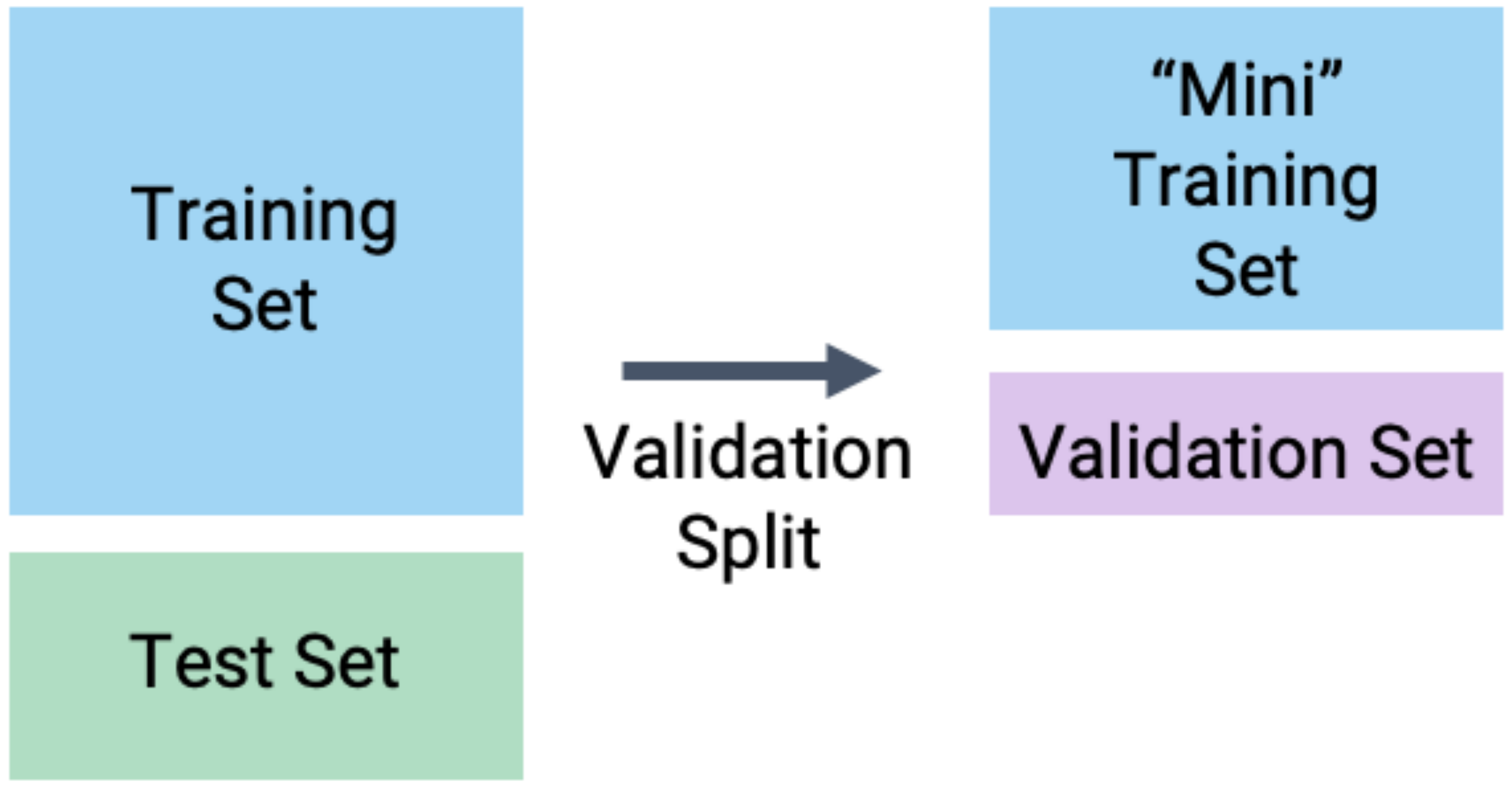
Train, test, and validation set

How do we pick the best model?



HW4 Part B: Using validation sets to fine tune your model and test the best one on the held-out test set!

Validation set approach



Using the validation set approach:

1. Split data into train and test sets (50/50%)
2. Randomly split train set again into train and validation sets
3. Fit model on train and predict responses using the validation set
4. Compute the validation set error (e.g., MSE or classification error) — provides an estimate of the test error rate
5. Repeat #2-4 for many random splits of the training set **and/or** for many different models of varying flexibility

Example:

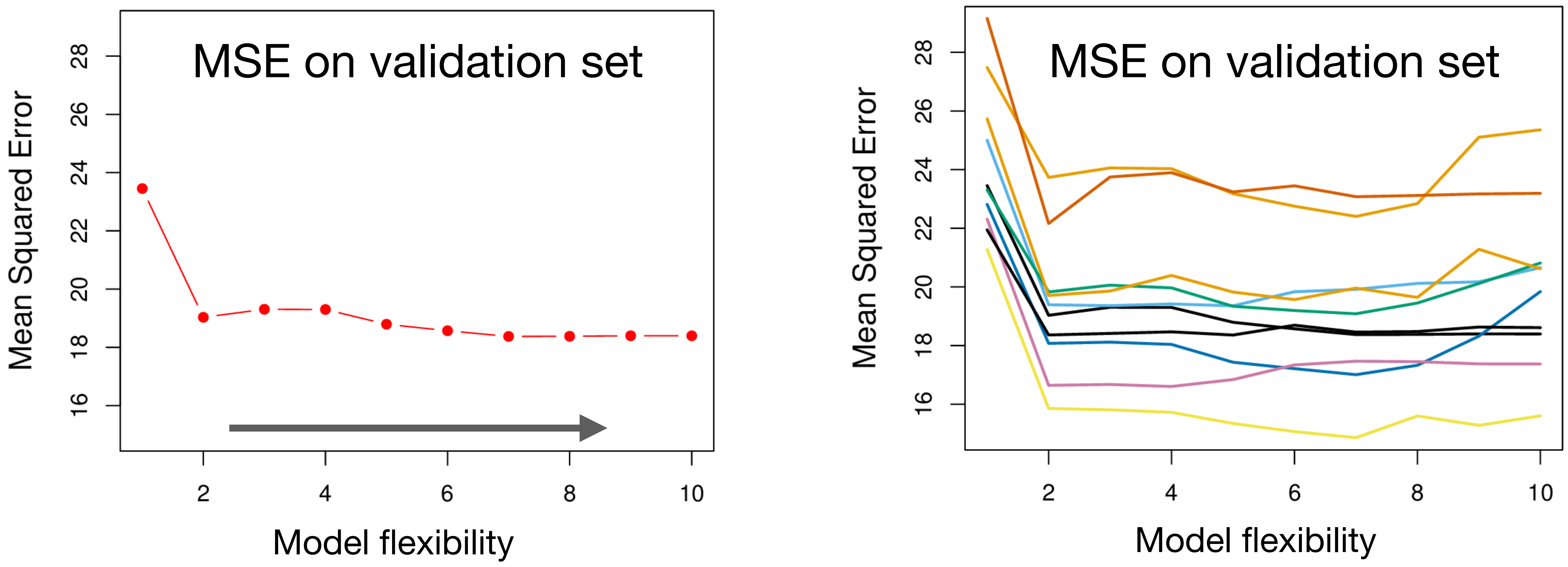
Variation 1 $\text{sales} = \beta_0 + \beta_1 \times \text{TV}$

Variation 2 $\text{sales} = \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{radio}$

Variation 3 $\text{sales} = \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{radio} + \beta_3 \times \text{newspaper}$

Increasing model flexibility

Validation set approach



Two drawbacks of the basic validation set approach:

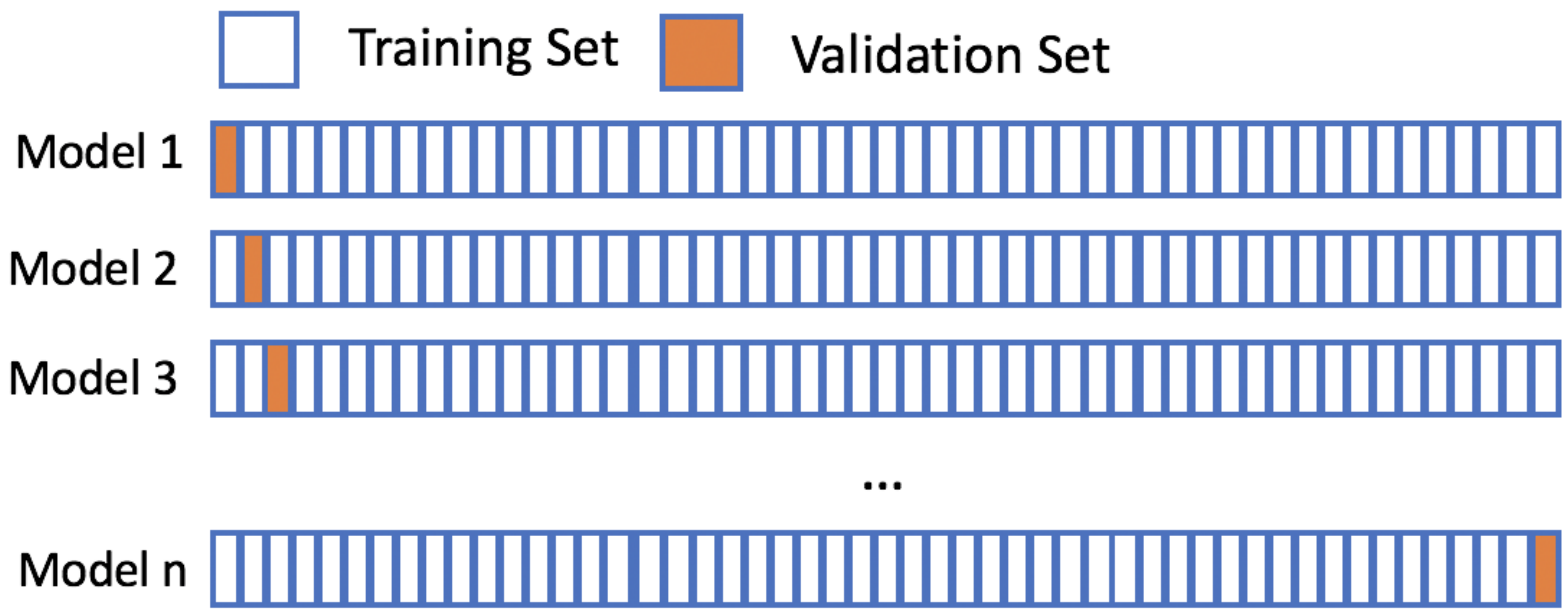
1. Validation estimate can be **highly variable** depending on how data is split
2. Only a subset of observations (train set) are used to fit model (even fewer *n* than before)

$\Rightarrow$ Validation set error tends to **overestimate** test error rate

Leave-one-out-cross-validation (LOOCV)

Using the LOOCV approach ("look-v"):

1. Split <u>train set</u> into <u>train (n-1 data points)</u> and <u>validation (1 data point)</u> set
2. Fit model on <u>n-1 data points</u> and predict the response of the single, held-out <u>validation</u> data point
3. Compute the **validation error** (e.g., MSE or classification error) of the single prediction
4. Repeat #2-4, each time **holding one data point out** as the validation set *(leave-one-out)*



$$\text{CV error} = \frac{1}{n}\sum_{i=1}^{n} \text{Err}_i$$

where

$\text{Err}_i = (y_i - \hat{y}_i)^2$ *(regression)*

$\text{Err}_i = I(y_i \neq \hat{y}_i)$ *(classification)*

Leave-one-out-cross-validation (LOOCV)

LOOCV advantages:

- **Always yields the same CV error** because no random splits of data (in contrast to validation set approach)
- **Less bias** and therefore **does not overestimate test error** (because we are using almost all of the train data to fit the model, instead of a fraction)

LOOCV drawbacks:

- In most cases, it is computationally expensive (must fit the model *n* times)

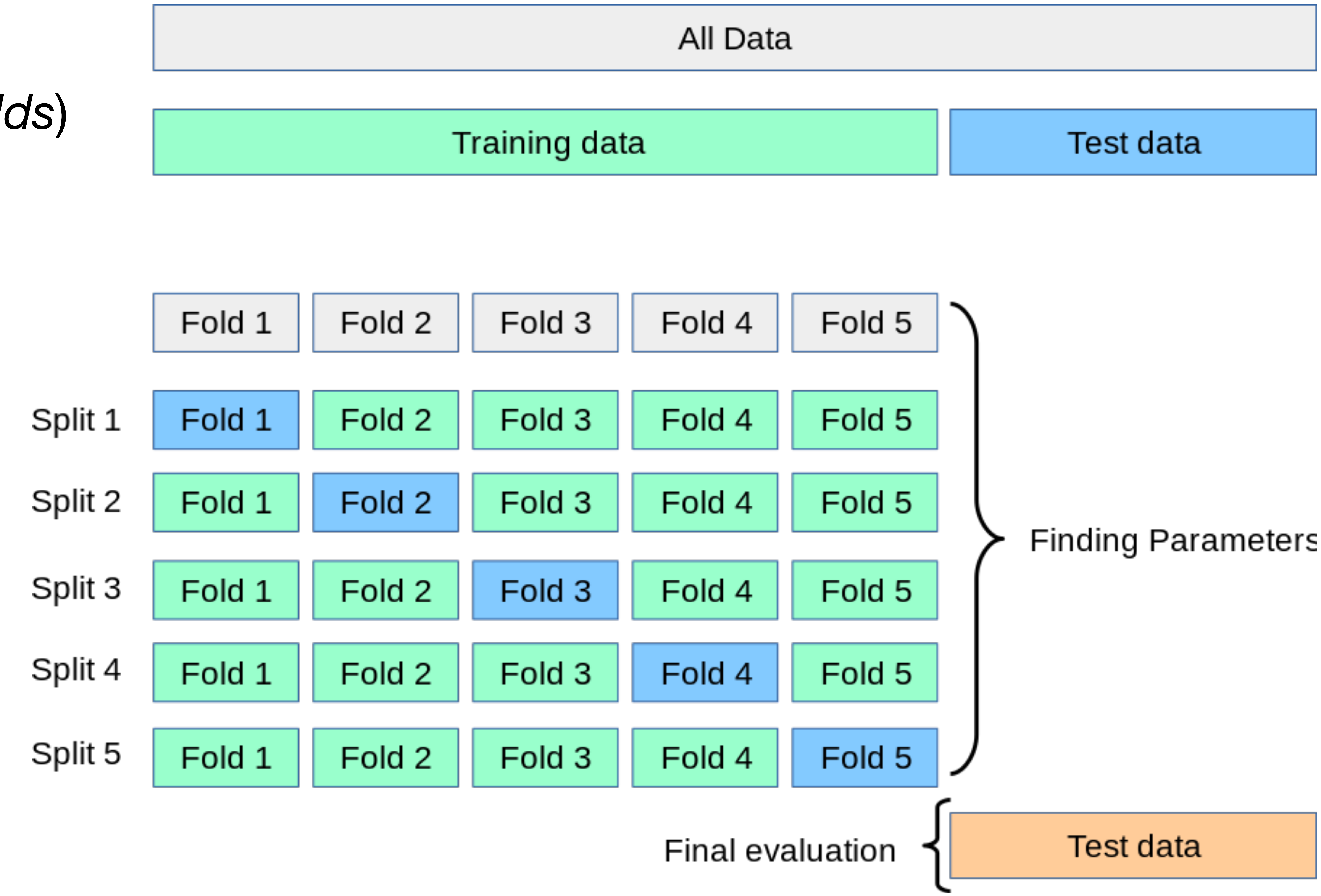
⇒ **Workaround:** *k-fold* cross validation (generalization of LOOCV)

k-fold cross-validation

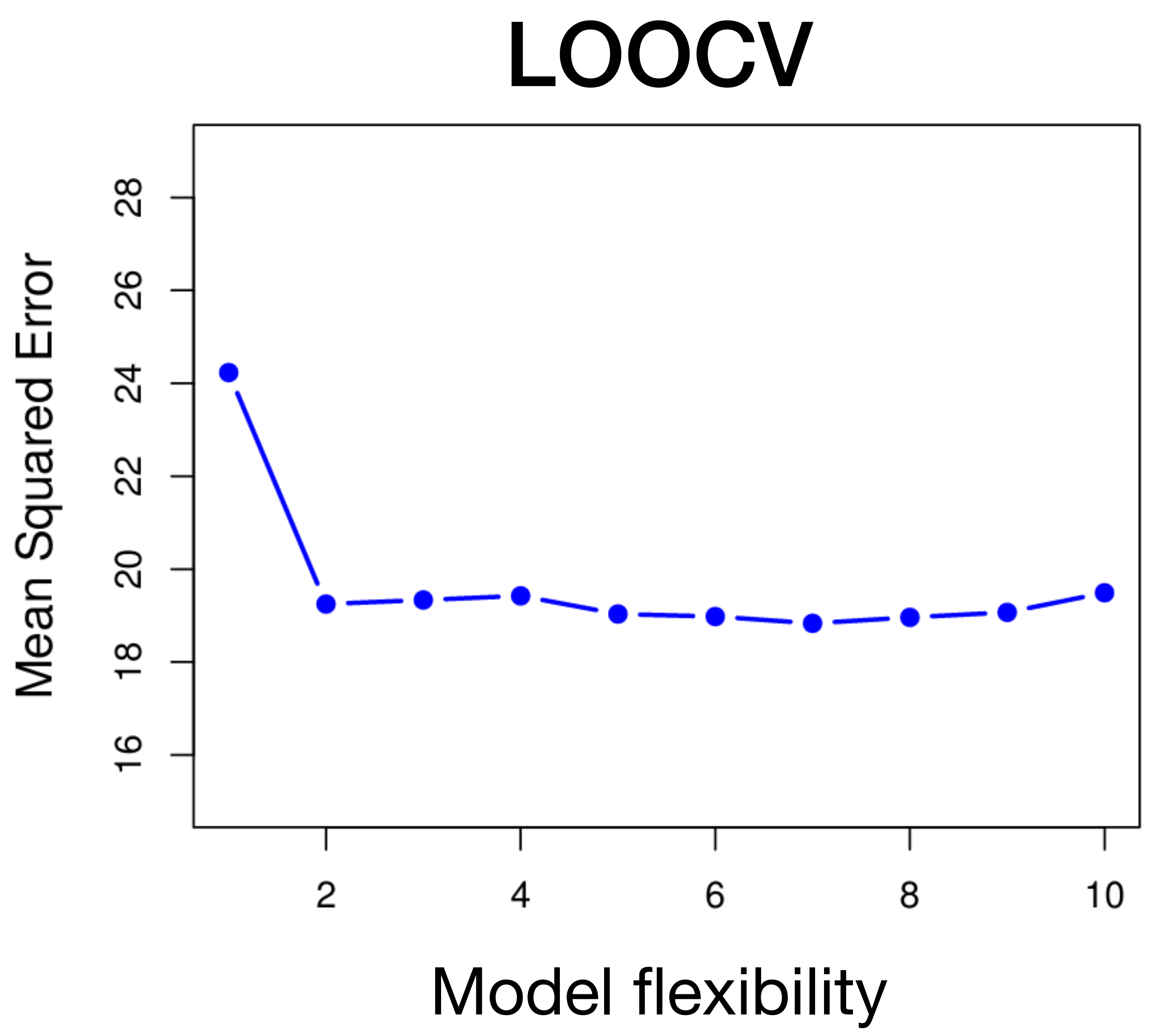
Using the k-fold CV approach:

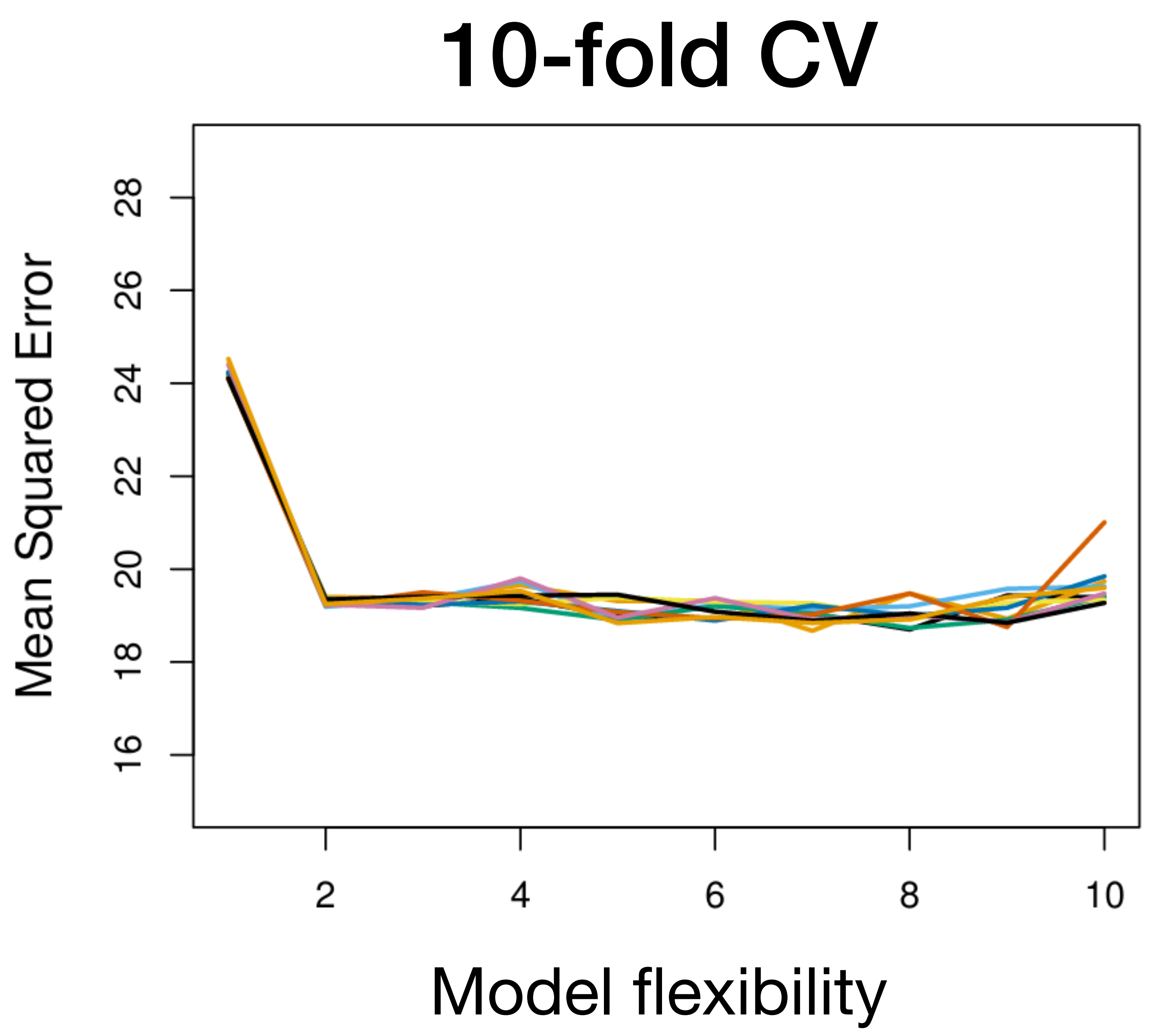
1. Split train set into k groups (*folds*) of equal size (n/k)
2. One fold is treated as the validation set
3. Fit model on train and predict responses on validation set
4. Compute the **validation error**
5. Repeat #2-4, using the next fold as the validation set

$$\text{CV error} = \frac{1}{k}\sum_{i=1}^{k} \text{Err}_i$$



LOOCV vs. 10-fold CV



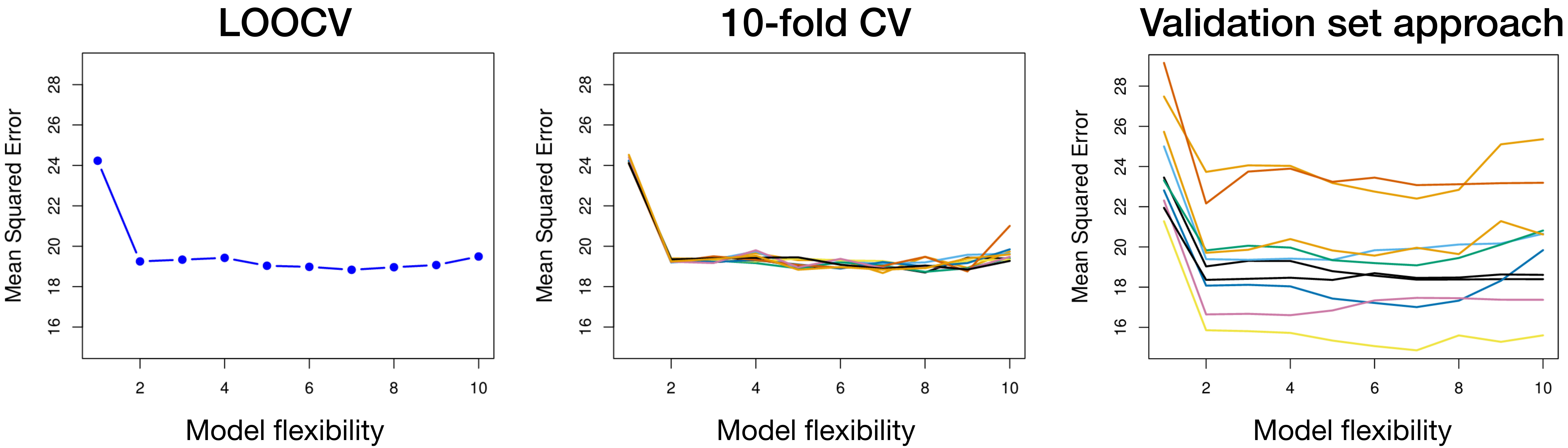


Both k-fold CV and LOOCV can be used to assess *any* model, no matter how complex!

k-fold advantages:

- Computationally cheaper than LOOCV (only need to fit the model *k* times)
- Gives more accurate estimates of test error rate…**Why?** ⇒ *bias-variance trade-off!*

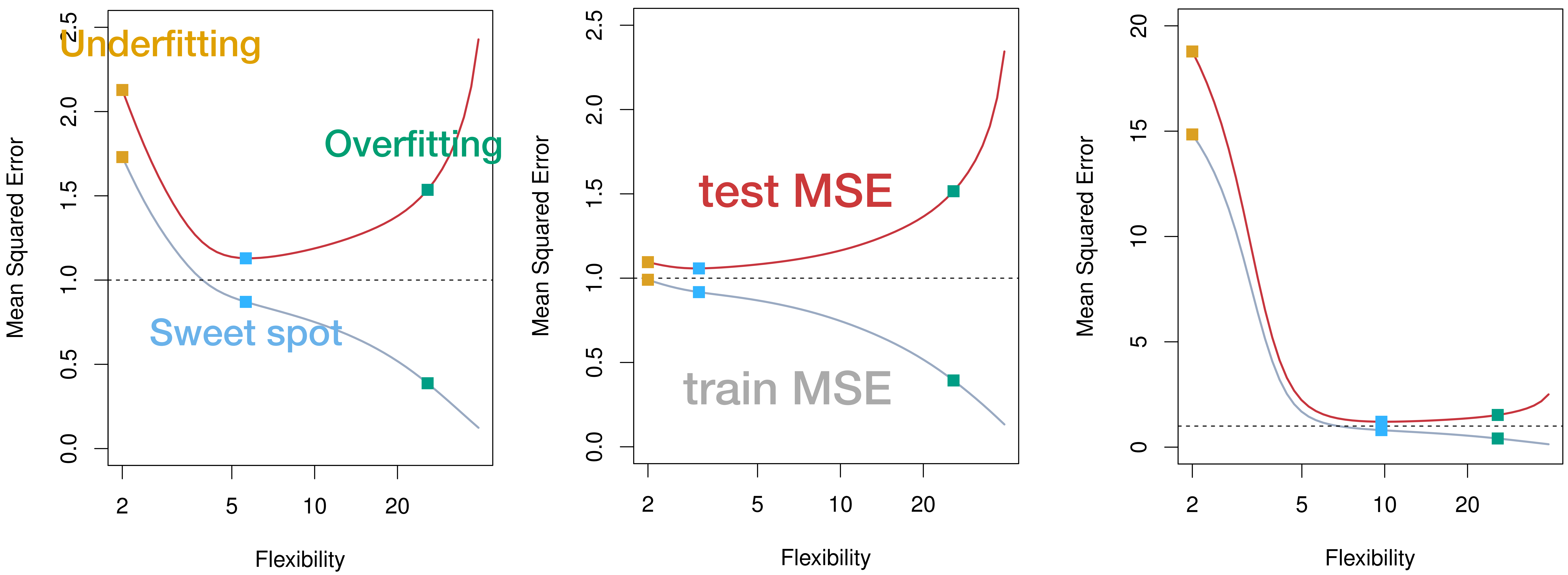
Comparing cross-validation methods



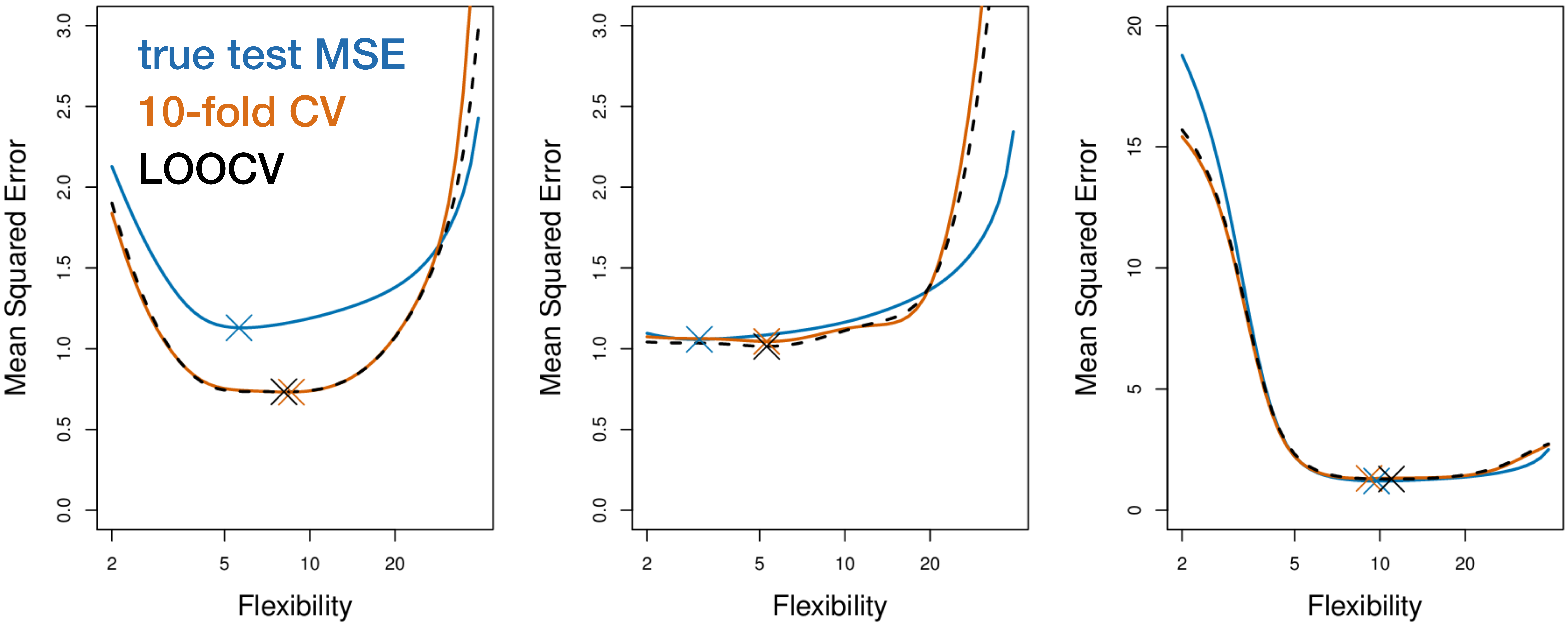
Bias and variance in test error estimates:

- In terms of ***bias*** (accuracy) in test error estimate: LOOCV < *k*-fold CV < validation set
- In terms of ***variance*** (noise) in test error estimate: *k*-fold CV < LOOCV < validation set
- **We typically use *k*=5 or *k*=10-fold CV** because it hits the sweet spot (not too biased, nor too much variance)

Comparing cross-validation methods



Comparing cross-validation methods



- Sometimes we care more about the **accuracy** of the test error estimate (e.g., MSE)
- Sometimes we only care about finding the **optimal model flexibility (×)**

Predictive modeling workflow (so far)

Using logistic regression as an example:

Fit a multiple logistic regression model to predict a patient's status (1 = alive, 0 = deceased) from the following predictors:

- **`sex:`** Factor with levels "Female" and "Male"
- **`diagnosis:`** Factor with levels "Meningioma", "LG glioma", "HG glioma", and "Other".
- **`loc:`** Location factor with levels "Infratentorial" and "Supratentorial".
- **`ki:`** Karnofsky index (0-100, assess a patient's functional ability and prognosis
- **`gtv:`** Gross tumor volume, in cubic centimeters.

$$\text{Let } p = P(\text{status} = 1 \mid \text{sex}, \text{diagnosis}, \text{loc}, \text{ki}, \text{gtv})$$

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1\text{ki} + \beta_2\text{gtv} + \beta_3\text{Male[Yes]} + \beta_4\text{LGglioma[Yes]} + \ldots \\ \beta_5\text{Meningioma[Yes]} + \beta_6\text{OtherDiag[Yes]} + \beta_7\text{Supratentorial[Yes]}$$

Predictive modeling workflow (so far)

Using multiple logistic regression as an example:

Fit a multiple logistic regression model to predict a patient's status (1 = alive, 0 = deceased) from the following predictors: `sex`, `diagnosis`, `loc`, `ki`, `gtv`.

1. Split data set into **test** and **train** set
2. Split train set further into a train and validation set using each of the following approaches:
    1. Validation set approach (random 50/50 split)
    2. LOOCV
    3. 10-fold CV
3. Fit the **multiple logistic regression** model on the training set and predict held-out outcomes on validation set
4. Compute the validation error using `sklearn.model_selection.cross_validate`
5. **Model selection:** Repeat steps 3-4 for several variations of the logistic regression model (use different subsets of the predictors)
6. Plot average classification error vs. number of predictors for each approach
7. Report the **best model** using your chosen cross-validation method

Upcoming + Reminders

Assignments:

- Quiz 3 (DUE: TODAY @ 11:59pm)
- Group Project Checkpoint 1 (DUE: TODAY @ 11:59pm)
  - We will provide written feedback on each video update via Canvas by **Friday**

Wednesday's topic: *Bootstrapping*

- **Read:** ISLP Ch. 5.2