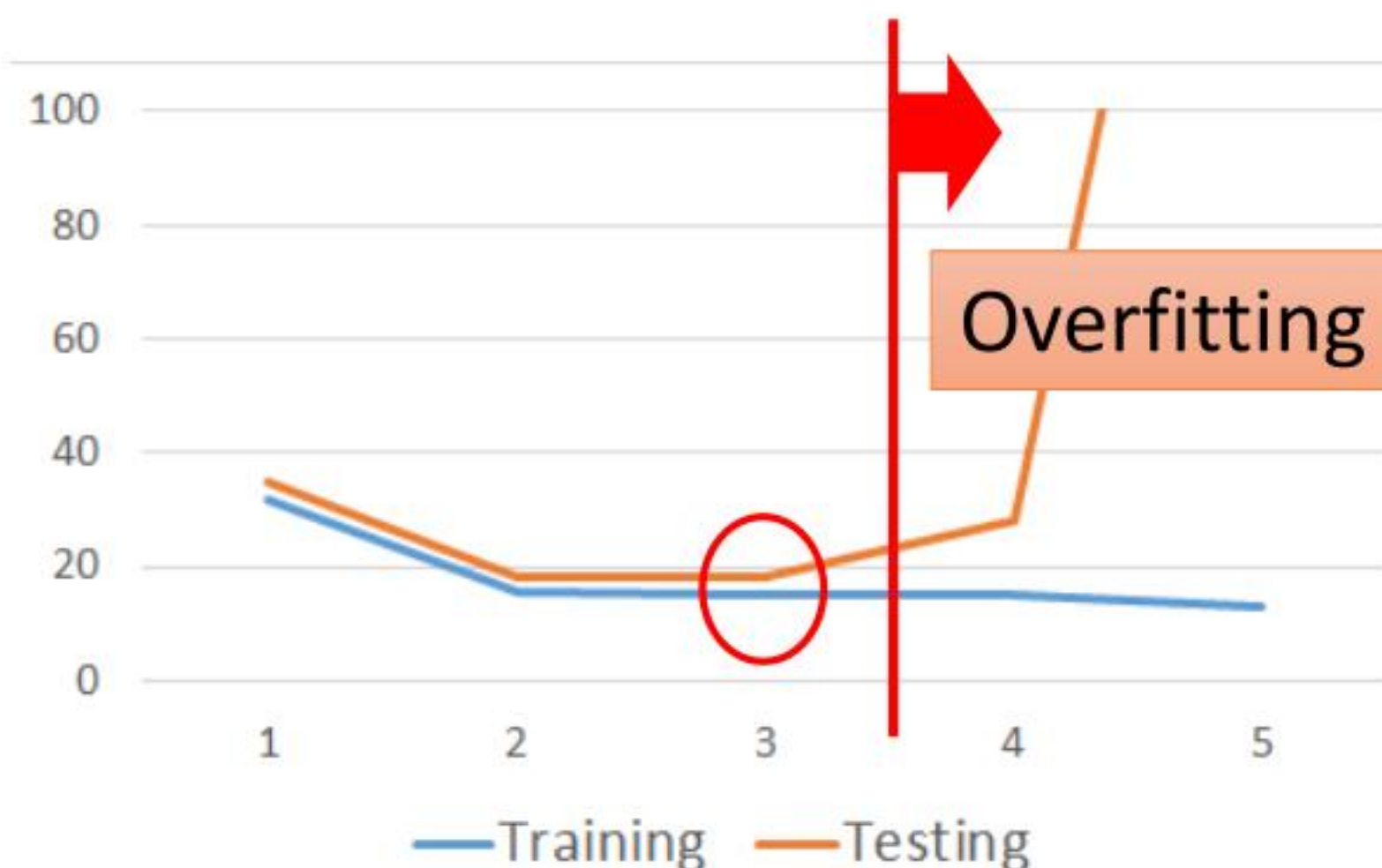




人工智能技术及应用

Artificial Intelligence and Application

Model Selection



| | Training | Testing |
|---|----------|---------|
| 1 | 31.9 | 35.0 |
| 2 | 15.4 | 18.4 |
| 3 | 15.3 | 18.1 |
| 4 | 14.9 | 28.2 |
| 5 | 12.8 | 232.1 |

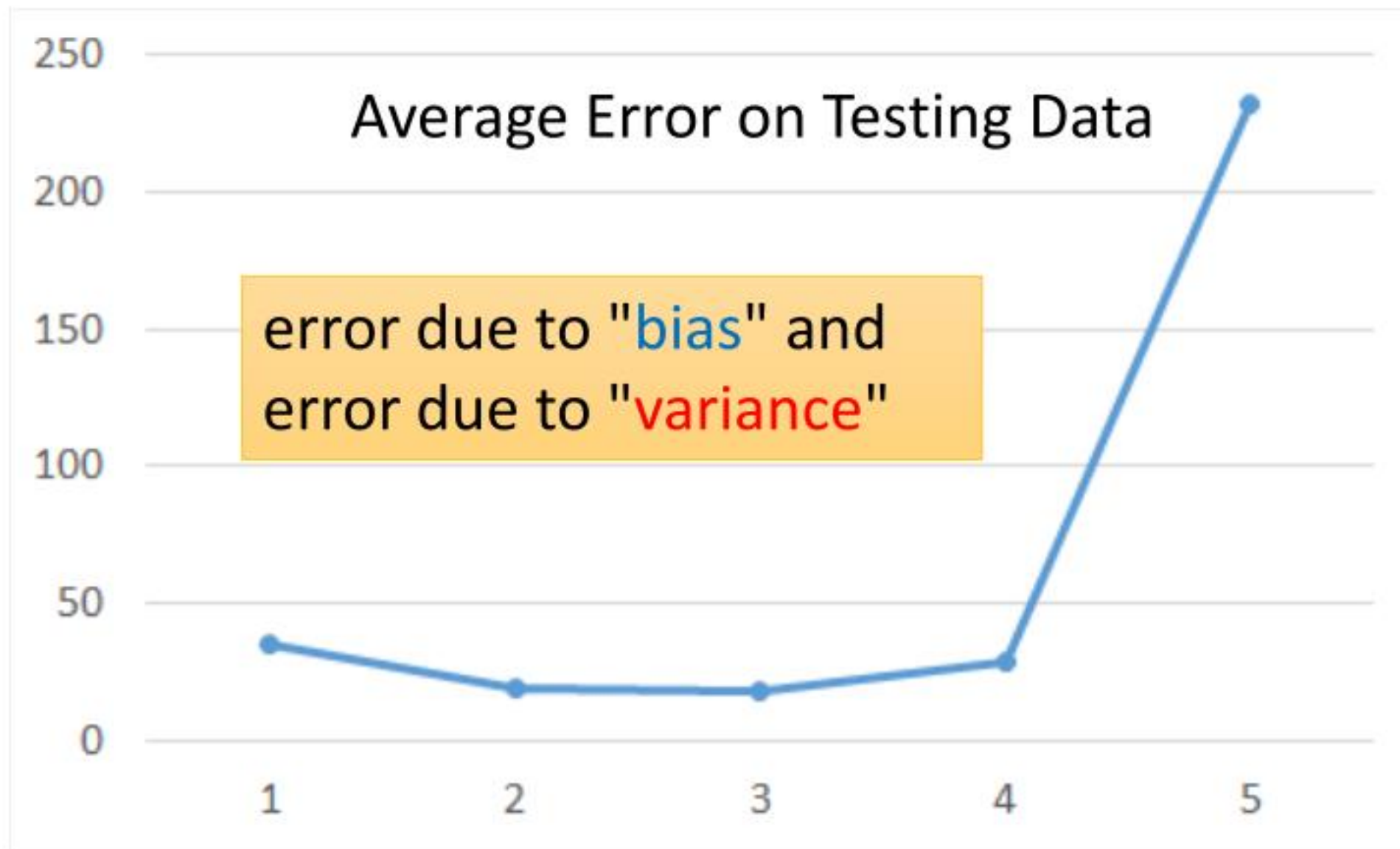
A more complex model does not always lead to better performance on testing data.

This is Overfitting.  Select suitable model

误差的来源

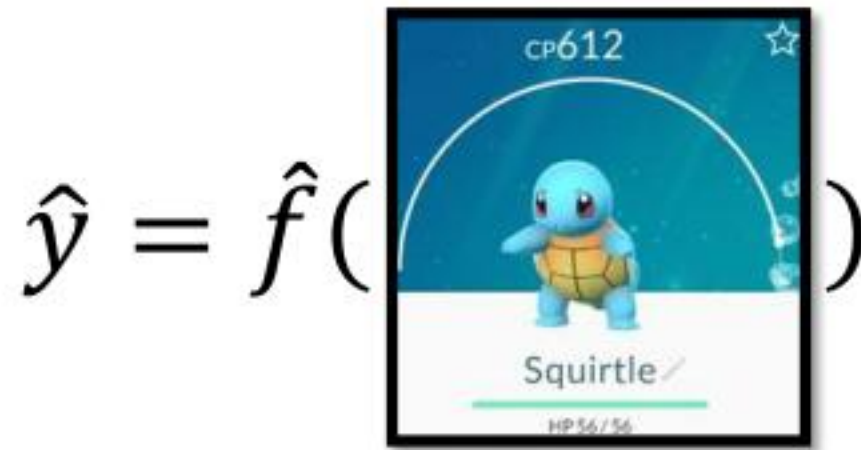


Review



A more complex model does not always lead to better performance on testing data.

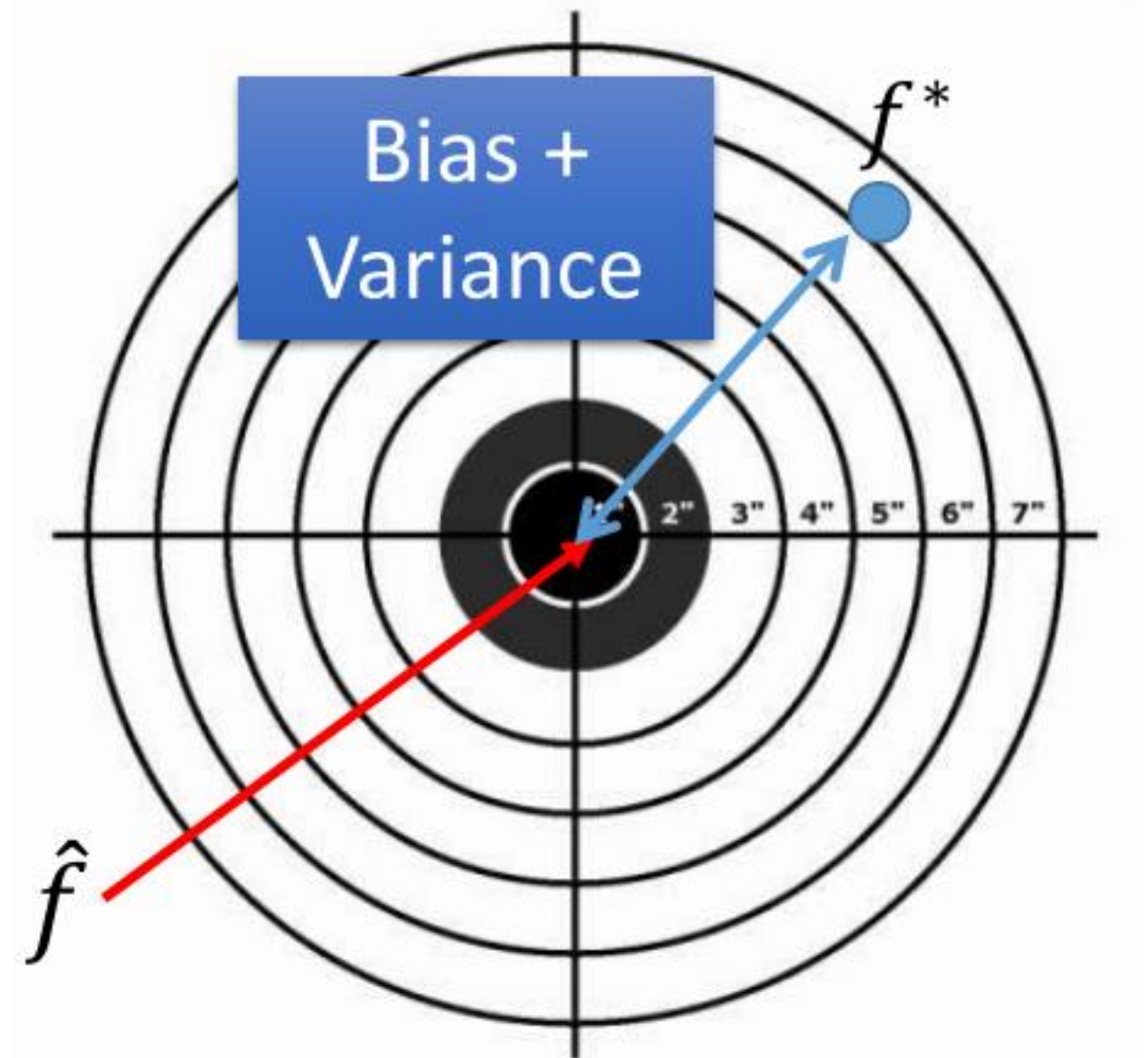
Estimator



Only Niantic knows \hat{f}

From training data,
we find f^*

f^* is an estimator of \hat{f}



Bias and Variance of Estimator

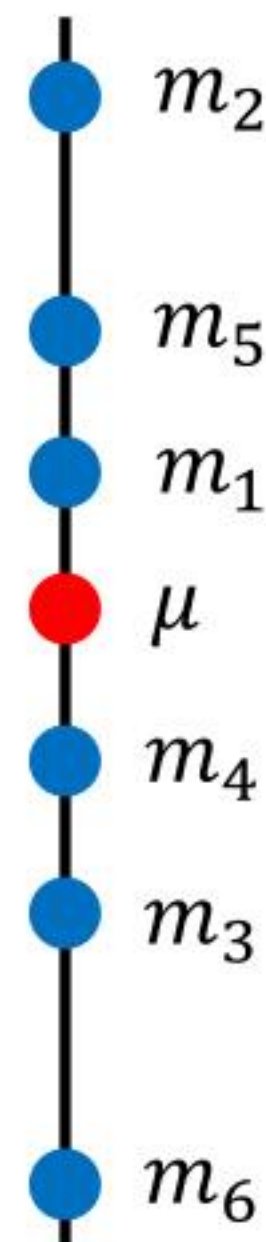
- Estimate the mean of a variable x
 - assume the mean of x is μ
 - assume the variance of x is σ^2
- Estimator of mean μ
 - Sample N points: $\{x^1, x^2, \dots, x^N\}$

$$m = \frac{1}{N} \sum_n x^n \neq \mu$$

中心极限定理

$$E[m] = E\left[\frac{1}{N} \sum_n x^n\right] = \frac{1}{N} \sum_n E[x^n] = \mu$$

unbiased



Bias and Variance of Estimator

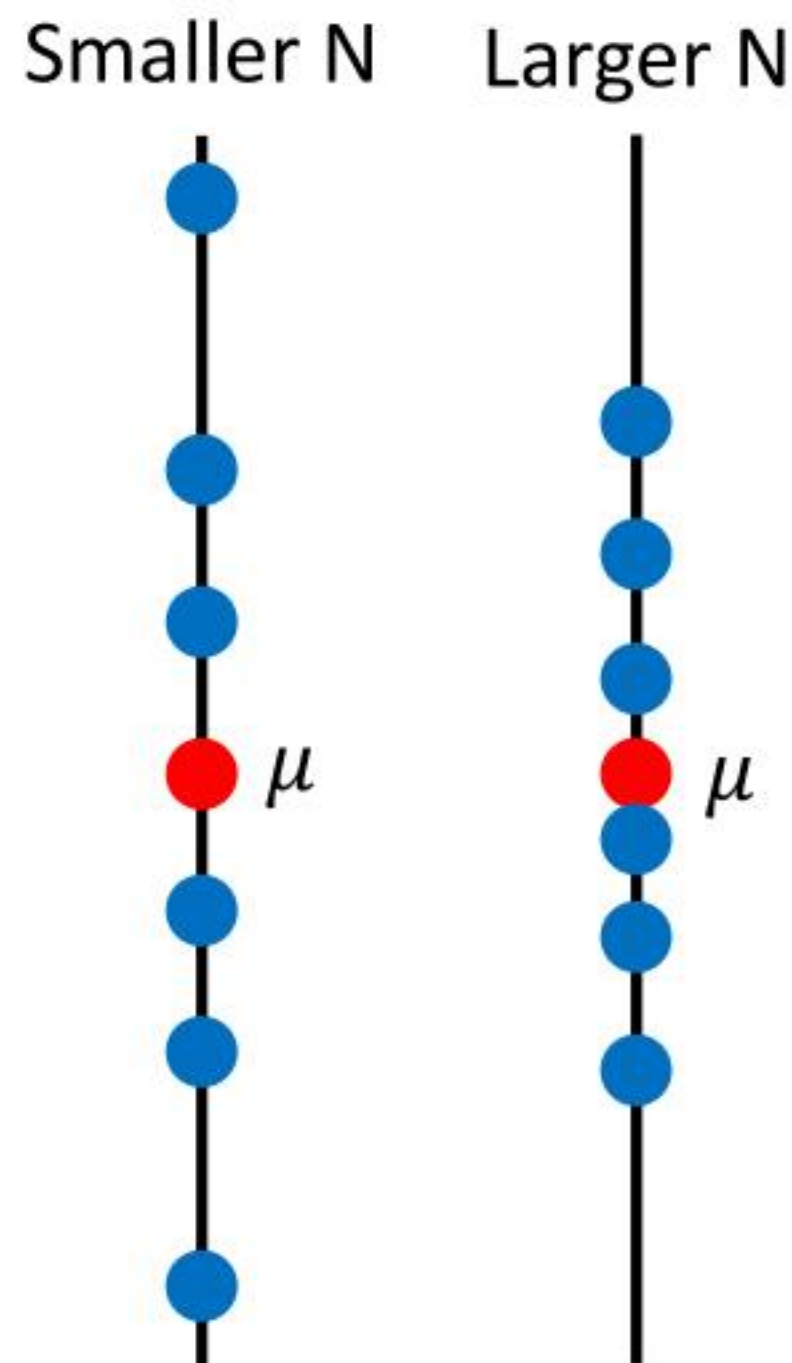
- Estimate the mean of a variable x
 - assume the mean of x is μ
 - assume the variance of x is σ^2
- Estimator of mean μ
 - Sample N points: $\{x^1, x^2, \dots, x^N\}$

$$m = \frac{1}{N} \sum_n x^n \neq \mu$$

$$\text{Var}[m] = \frac{\sigma^2}{N}$$

Variance depends
on the number of
samples

unbiased



Bias and Variance of Estimator

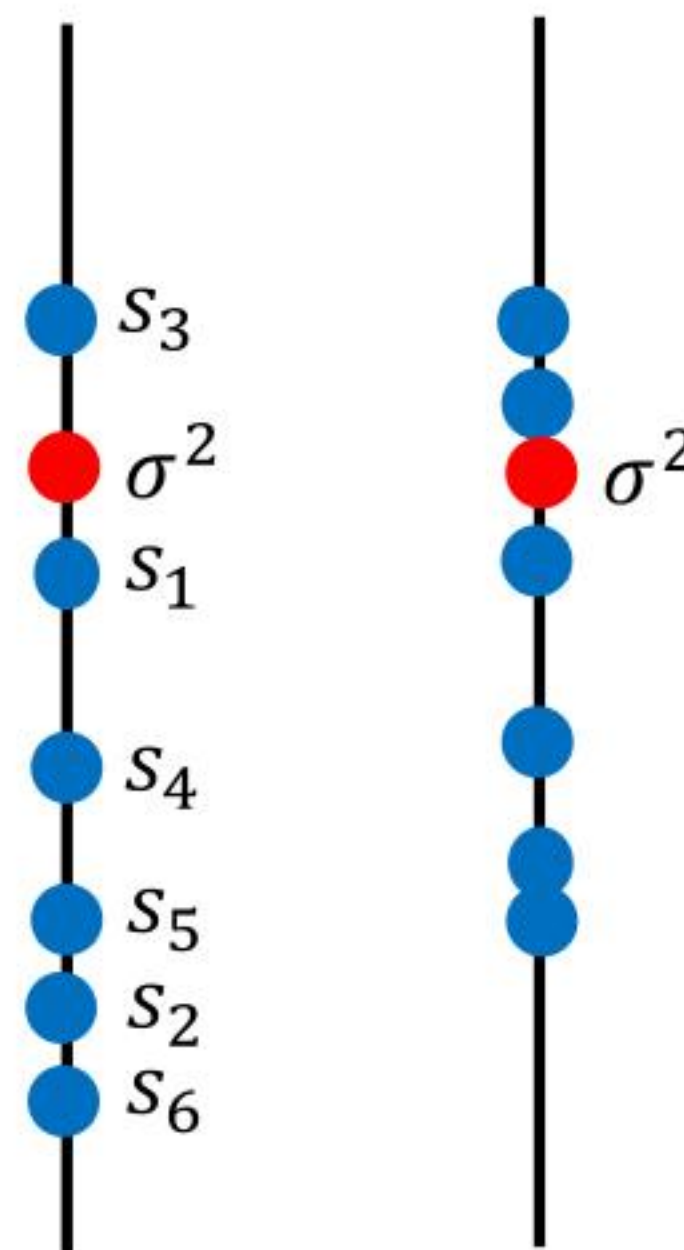
- Estimate the mean of a variable x
 - assume the mean of x is μ
 - assume the variance of x is σ^2
- Estimator of variance σ^2
 - Sample N points: $\{x^1, x^2, \dots, x^N\}$

$$m = \frac{1}{N} \sum_n x^n \quad s = \frac{1}{N} \sum_n (x^n - m)^2$$

Biased estimator

$$E[s] = \frac{N-1}{N} \sigma^2 \neq \sigma^2$$

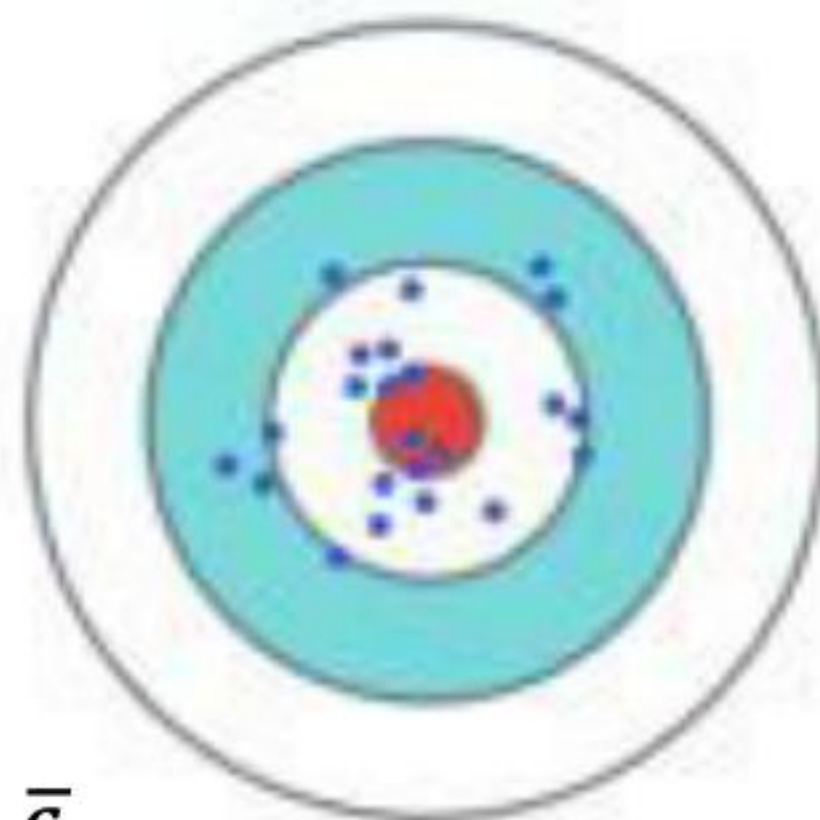
Increase N



Low Variance

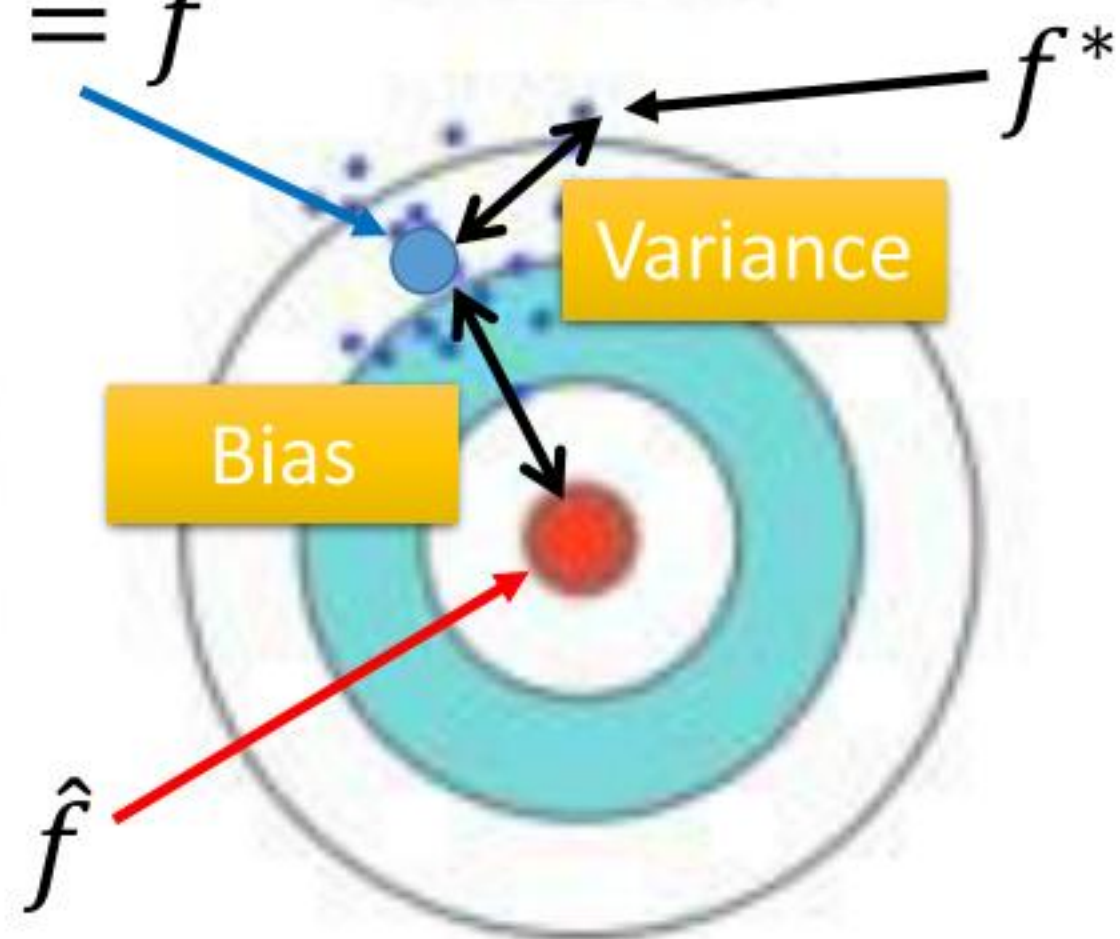
High Variance

Low Bias



$$E[f^*] = \bar{f}$$

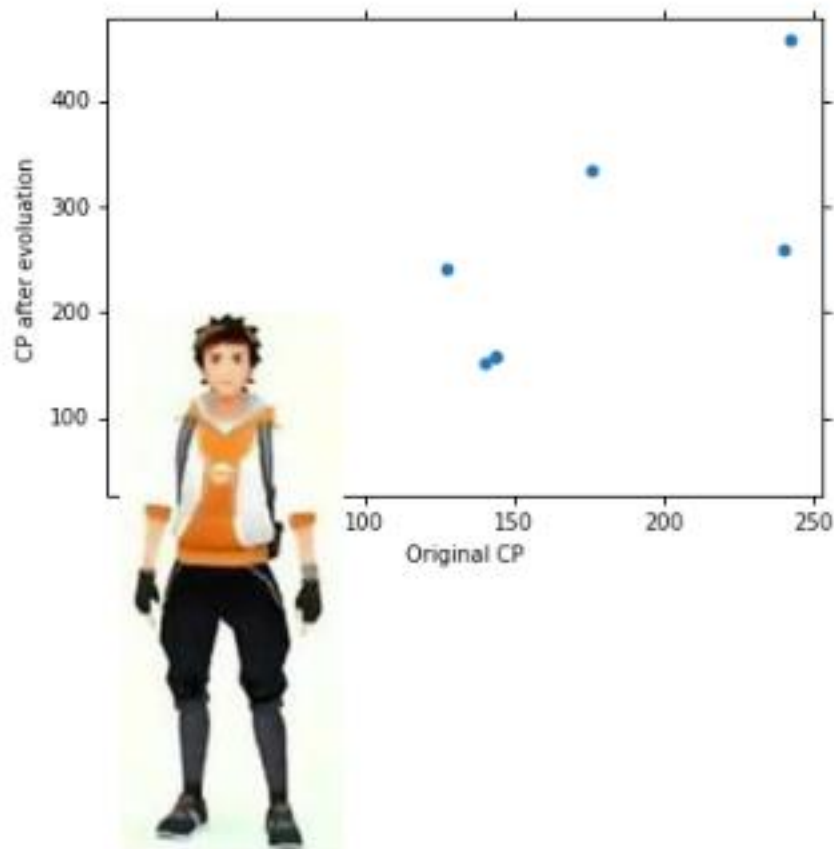
High Bias



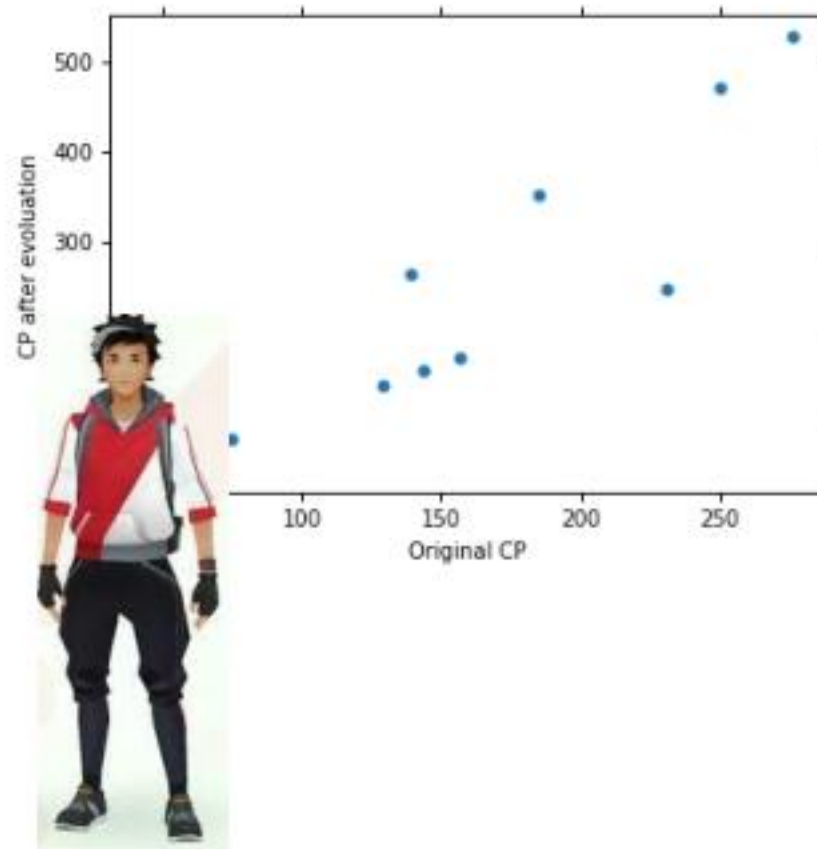
Parallel Universes

- In all the universes, we are collecting (catching) 10 Pokémon as training data to find f^*

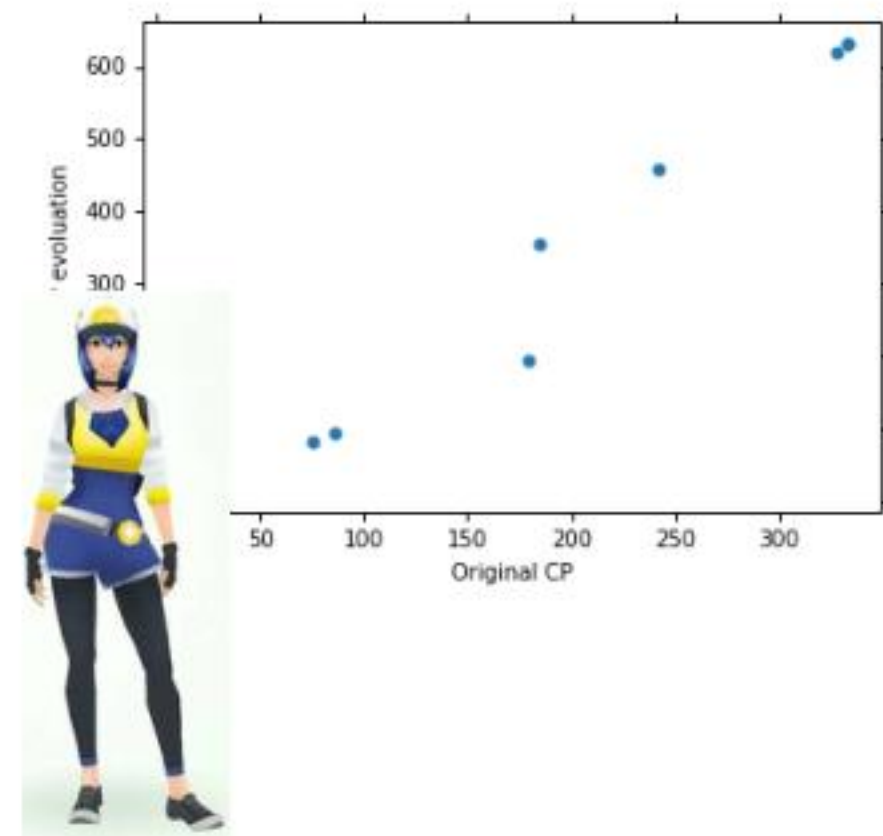
Universe 1



Universe 2



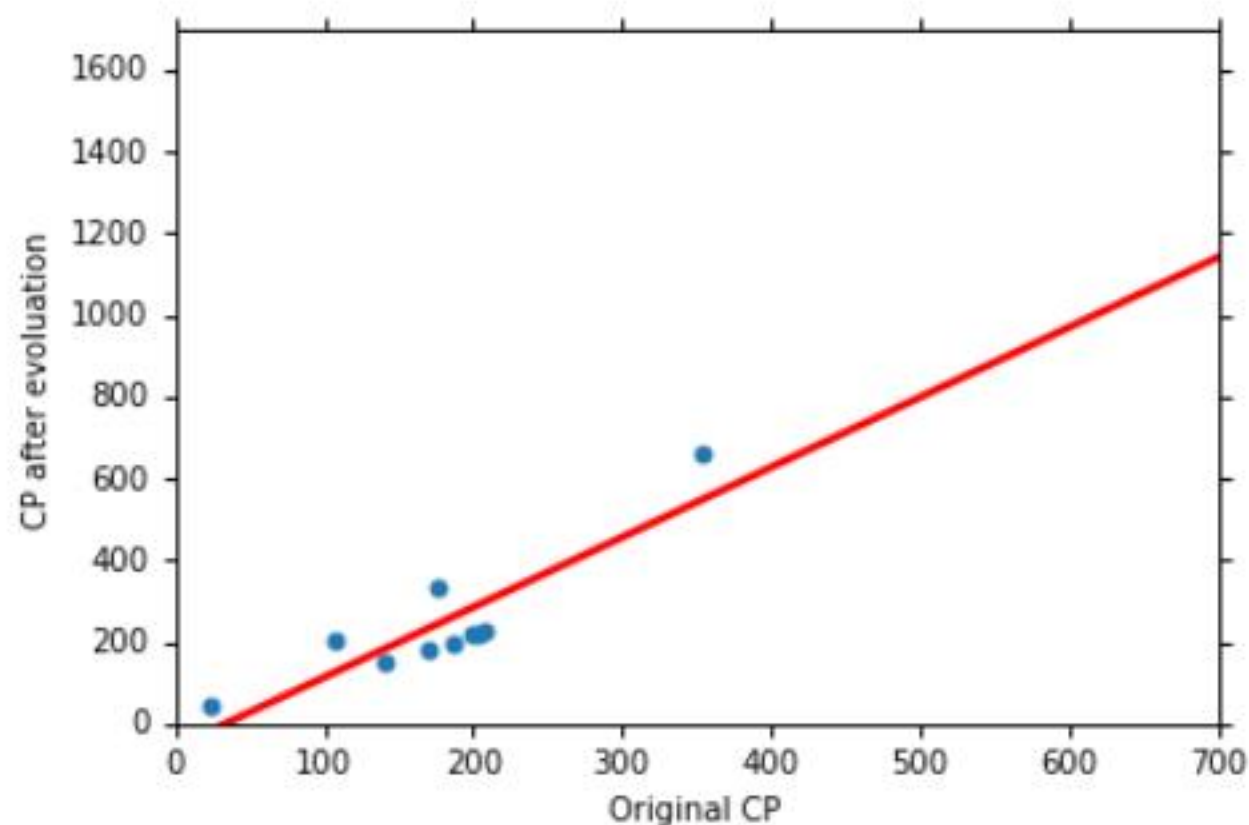
Universe 3



Parallel Universes

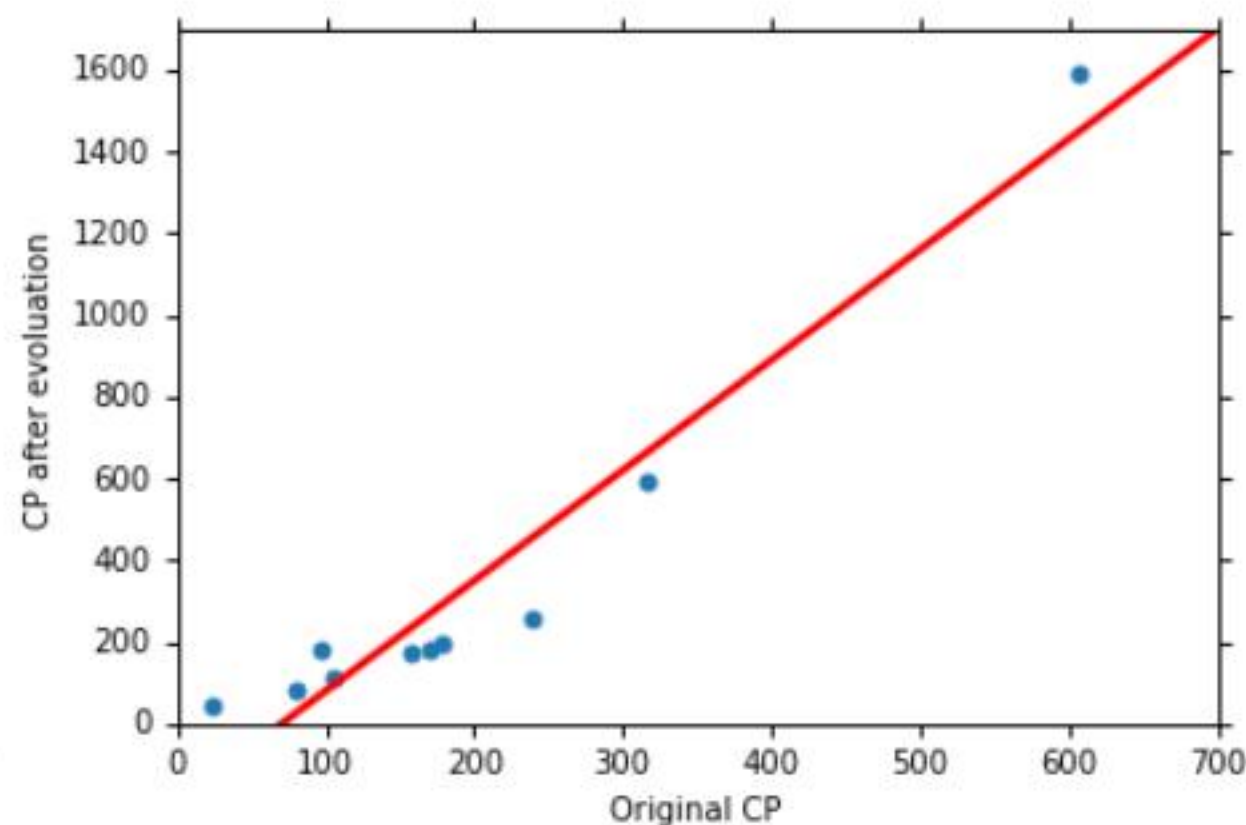
- In different universes, we use the same model, but obtain different f^*

Universe 123



$$y = b + w \cdot x_{cp}$$

Universe 345



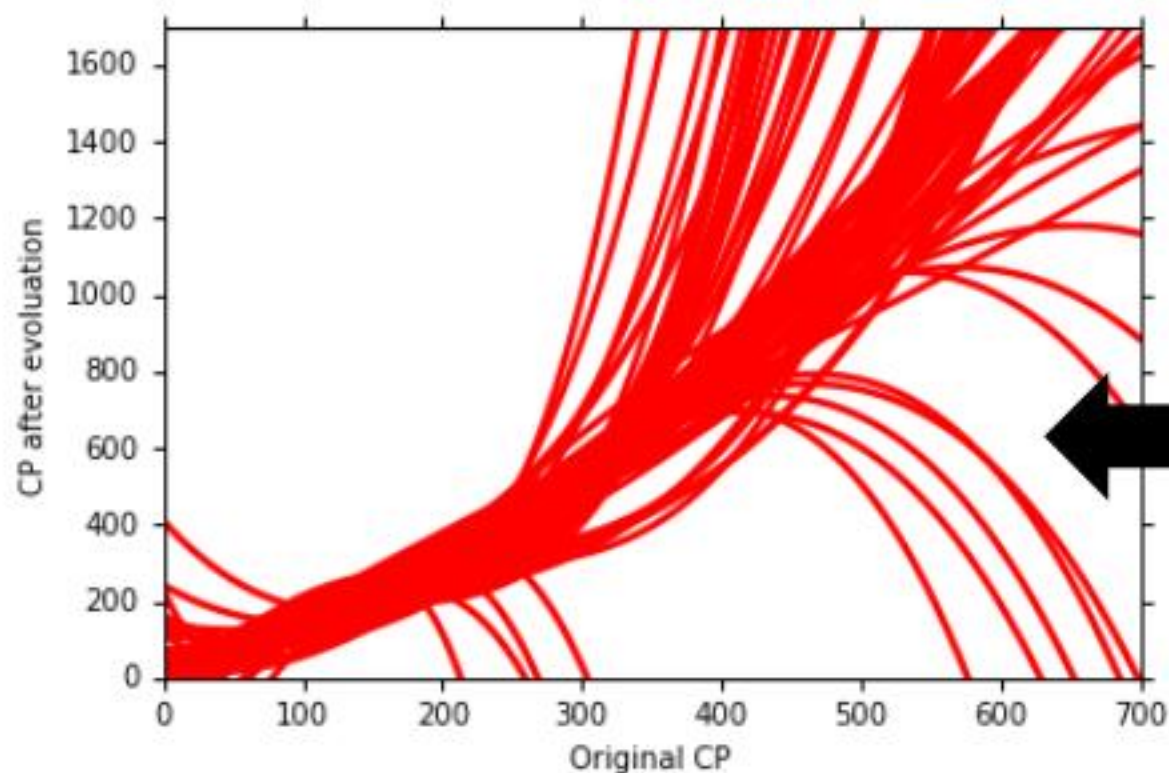
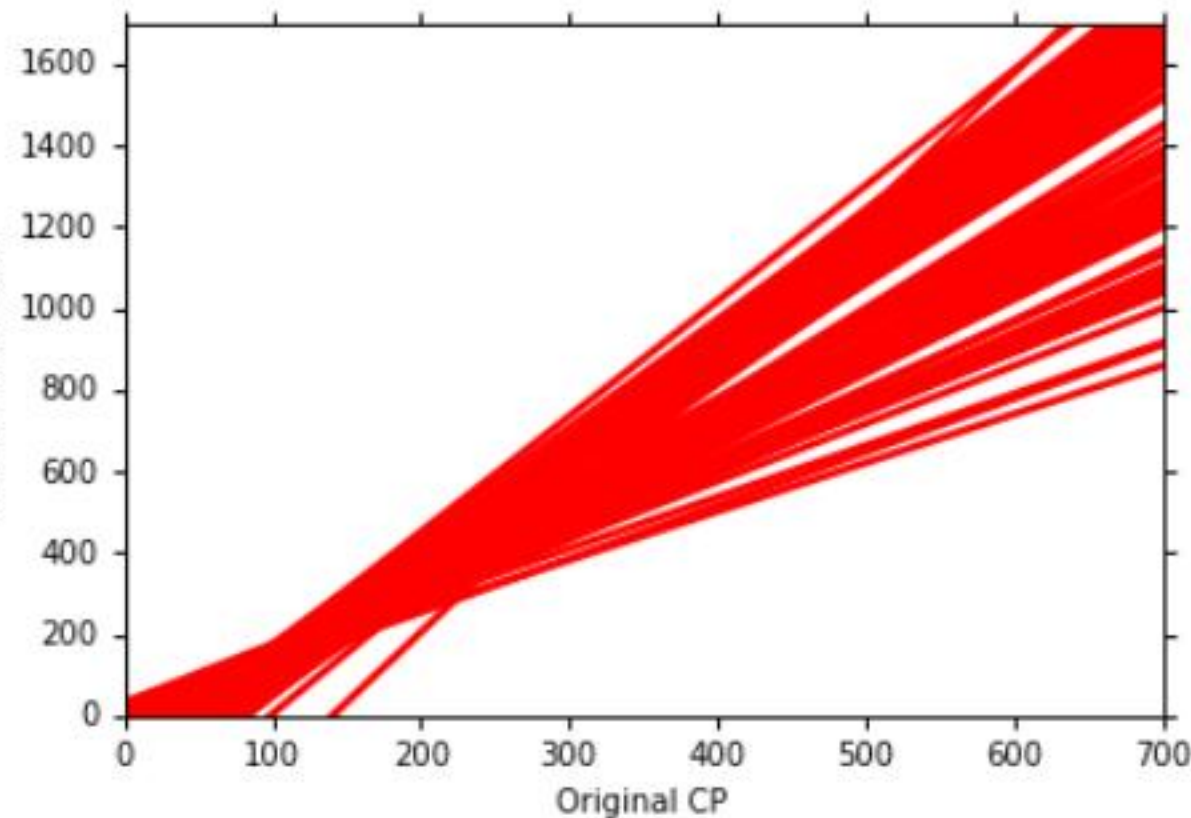
$$y = b + w \cdot x_{cp}$$

f^* in 100 Universes

$$y = b + w \cdot x_{cp}$$



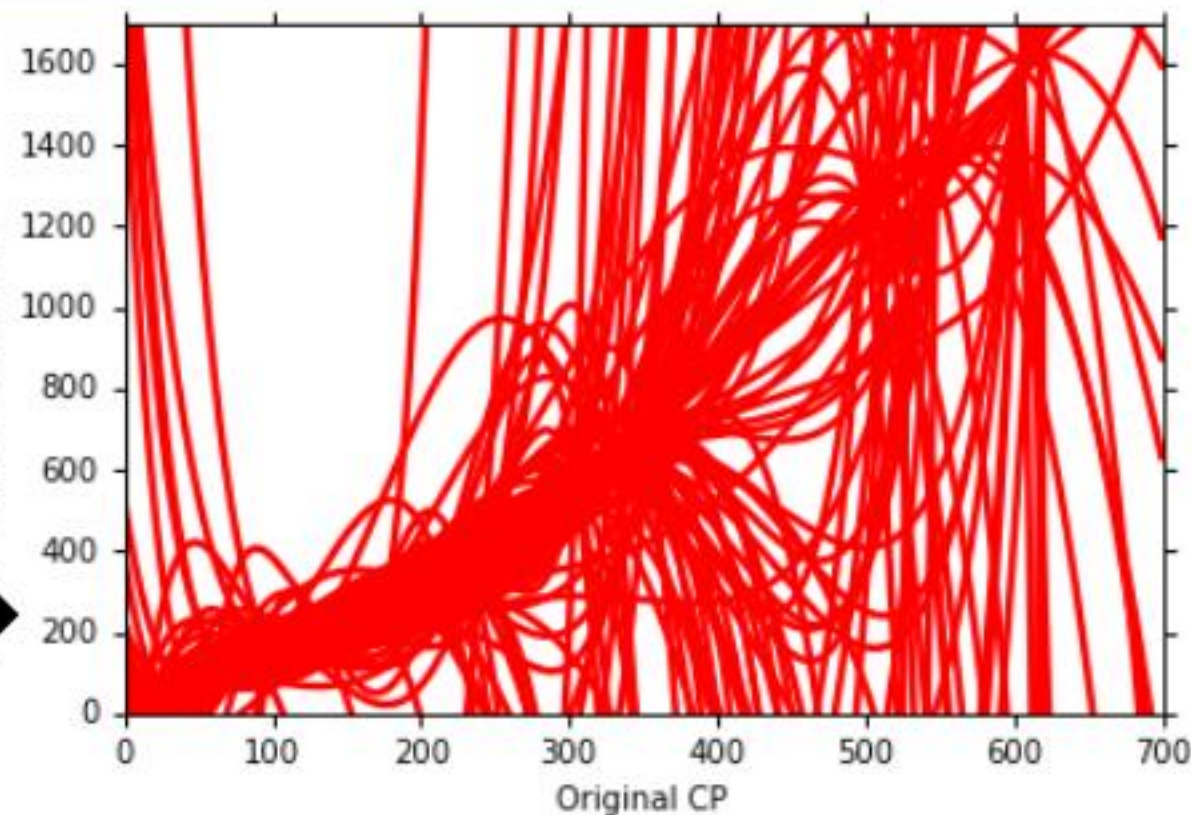
CP after evolution



$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3$$



CP after evolution

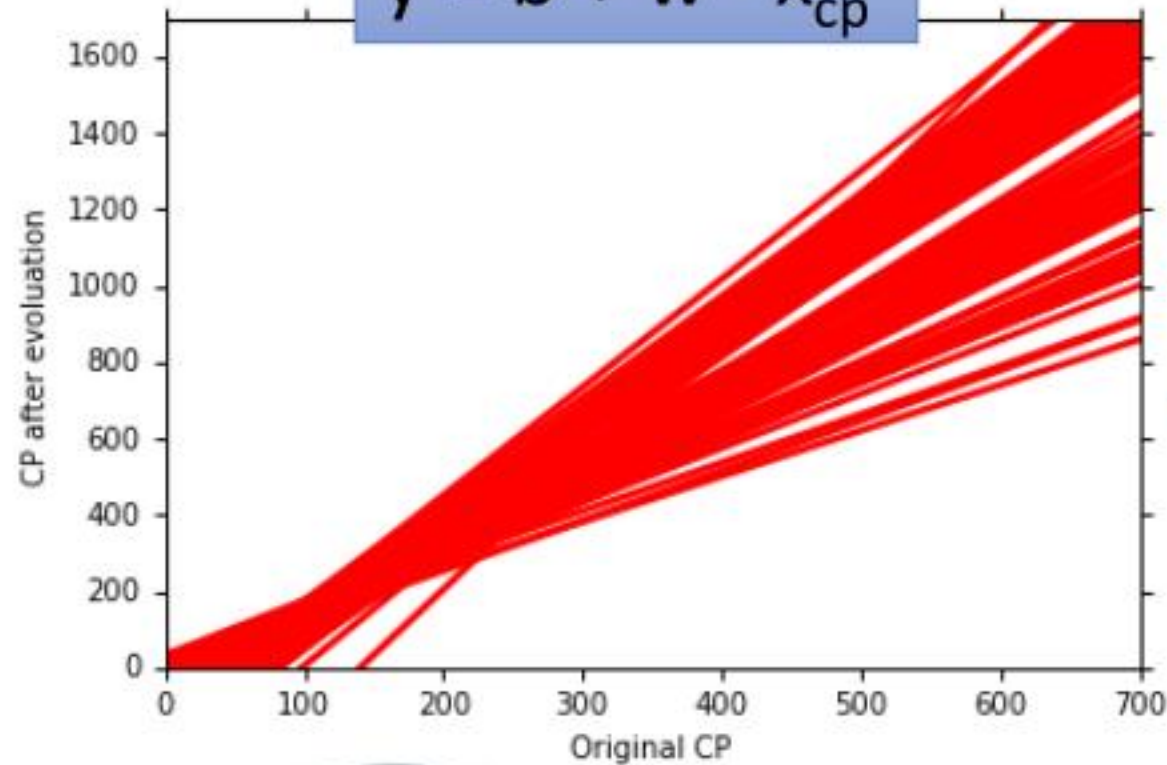


$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$$

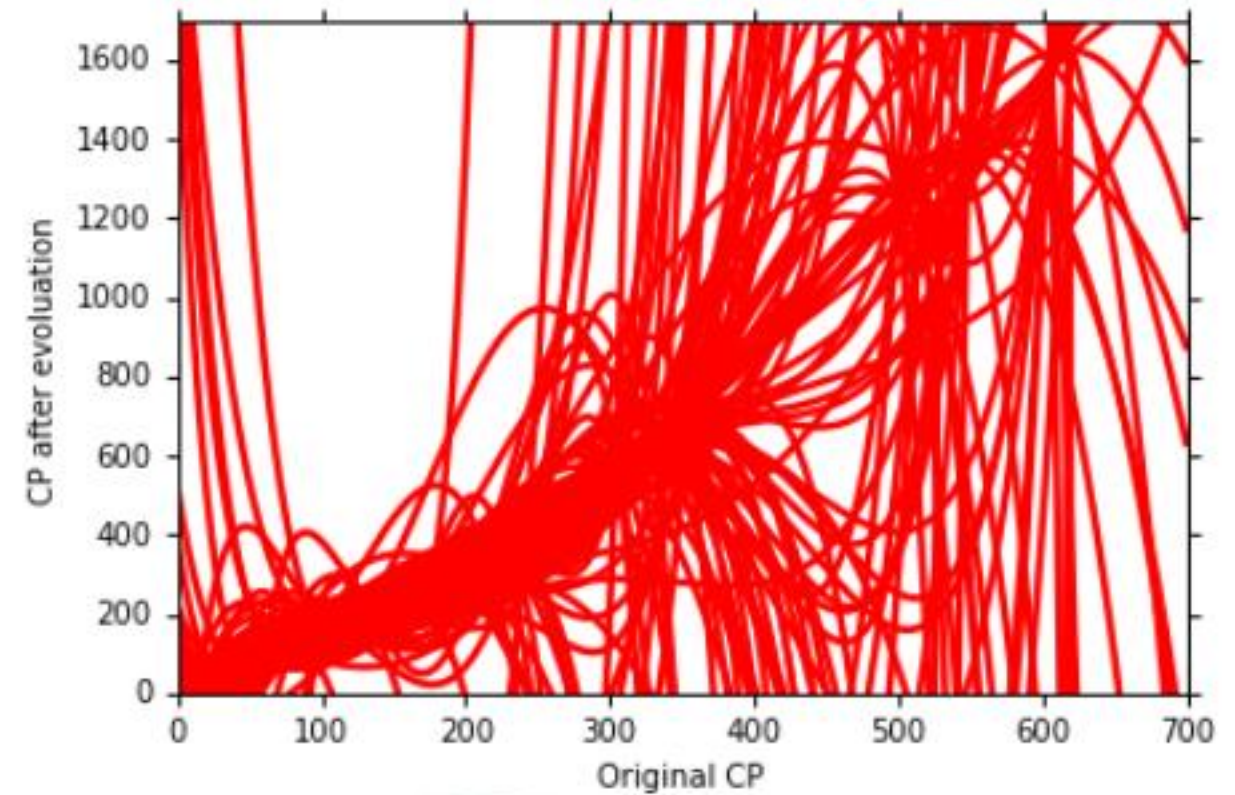


Variance

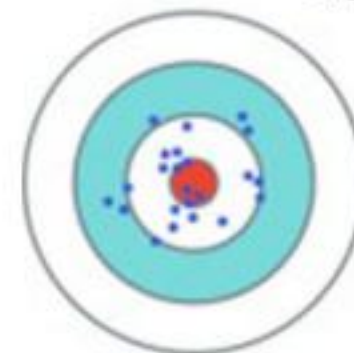
$$y = b + w \cdot x_{cp}$$



$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$$



Small
Variance



Large
Variance

Simpler model is less influenced by the sampled data

Consider the extreme case $f(x) = 5$

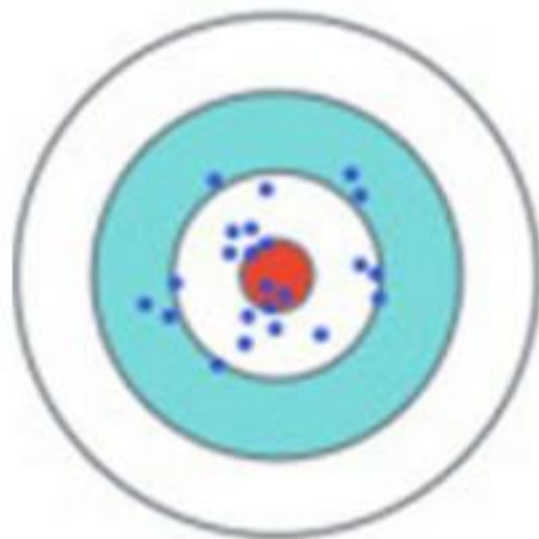
Bias

$$E[f^*] = \bar{f}$$

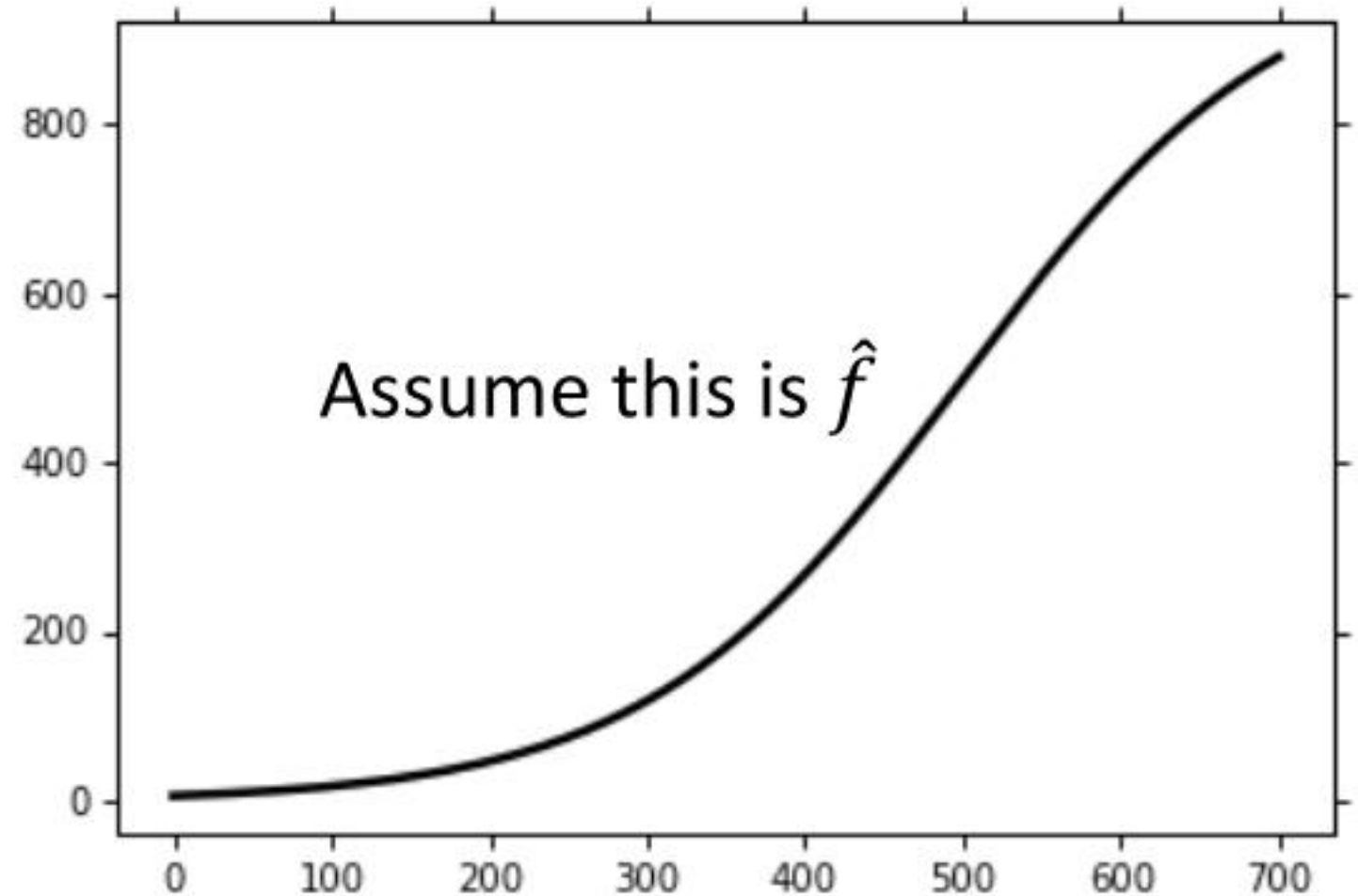
- Bias: If we average all the f^* , is it close to \hat{f} ?



Large
Bias



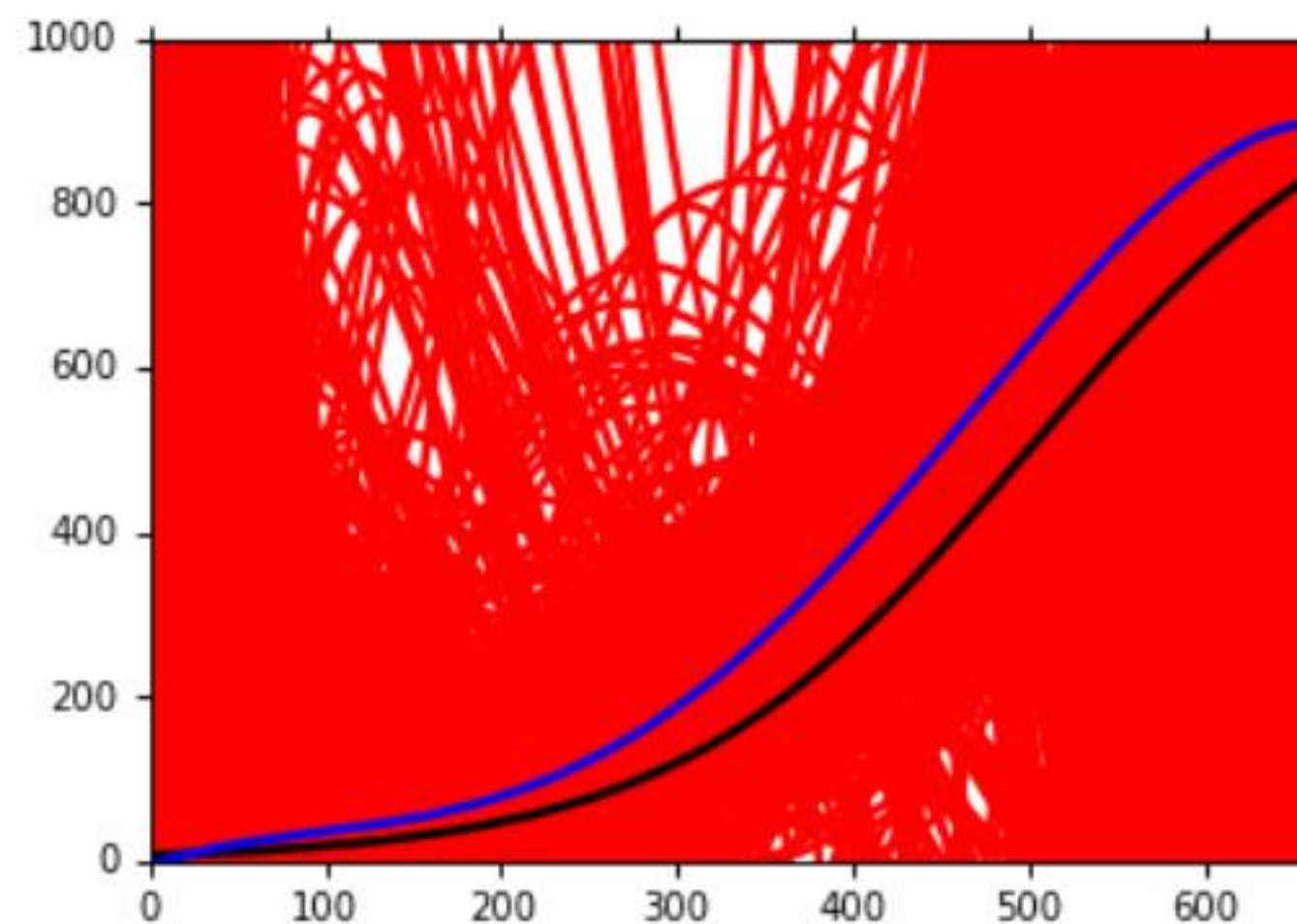
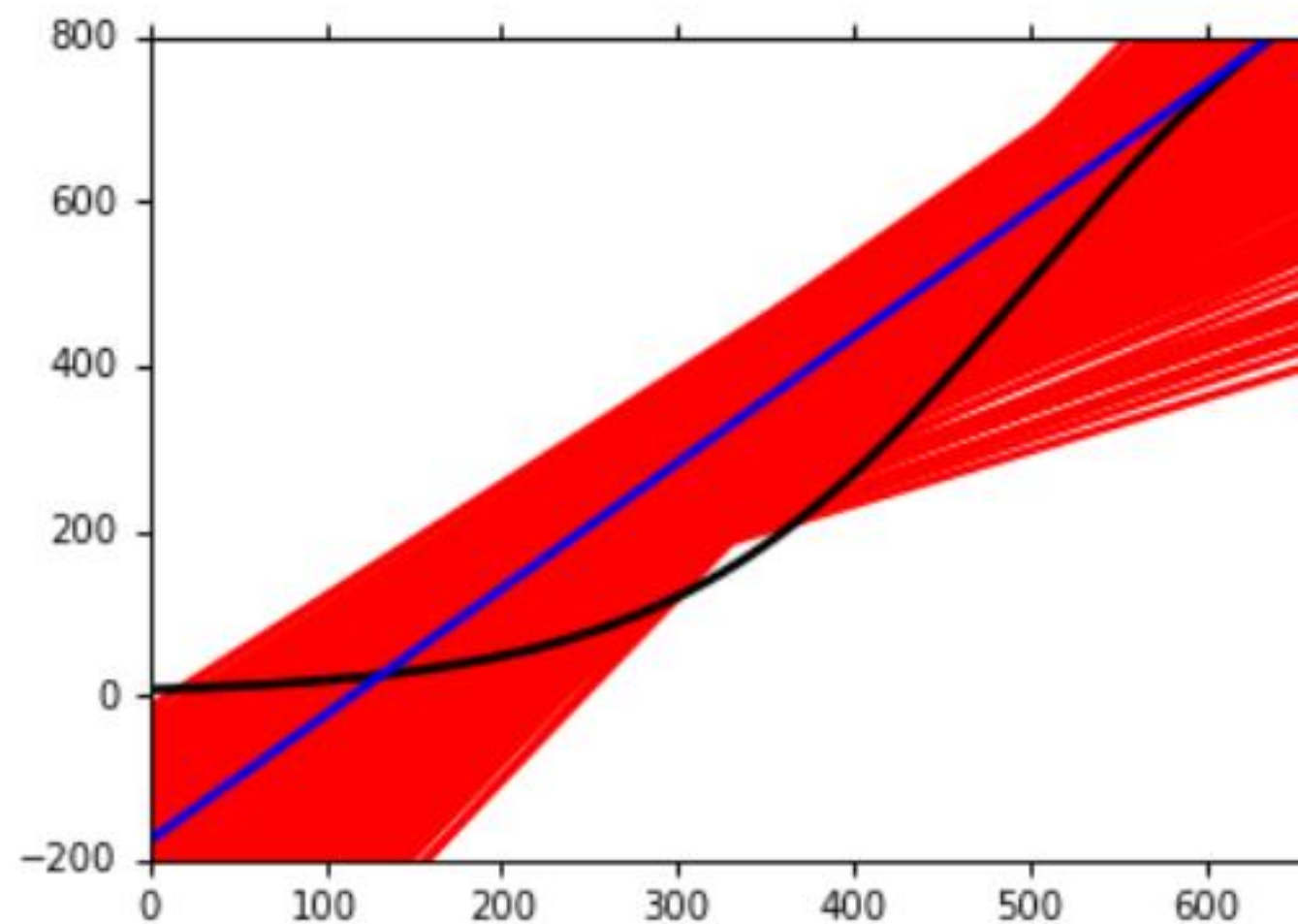
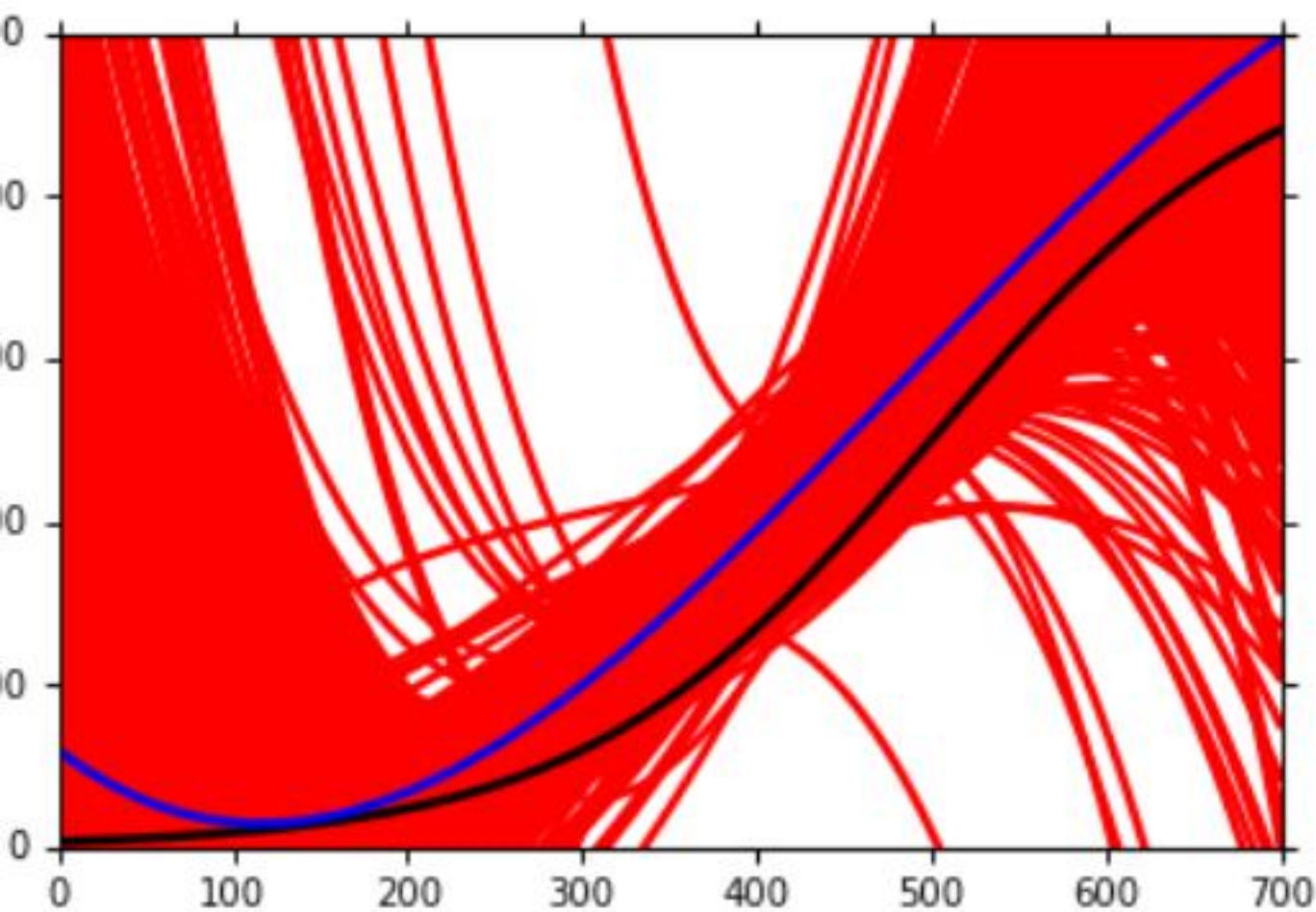
Small
Bias



Black curve: the true function \hat{f}

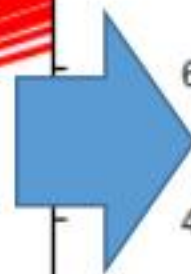
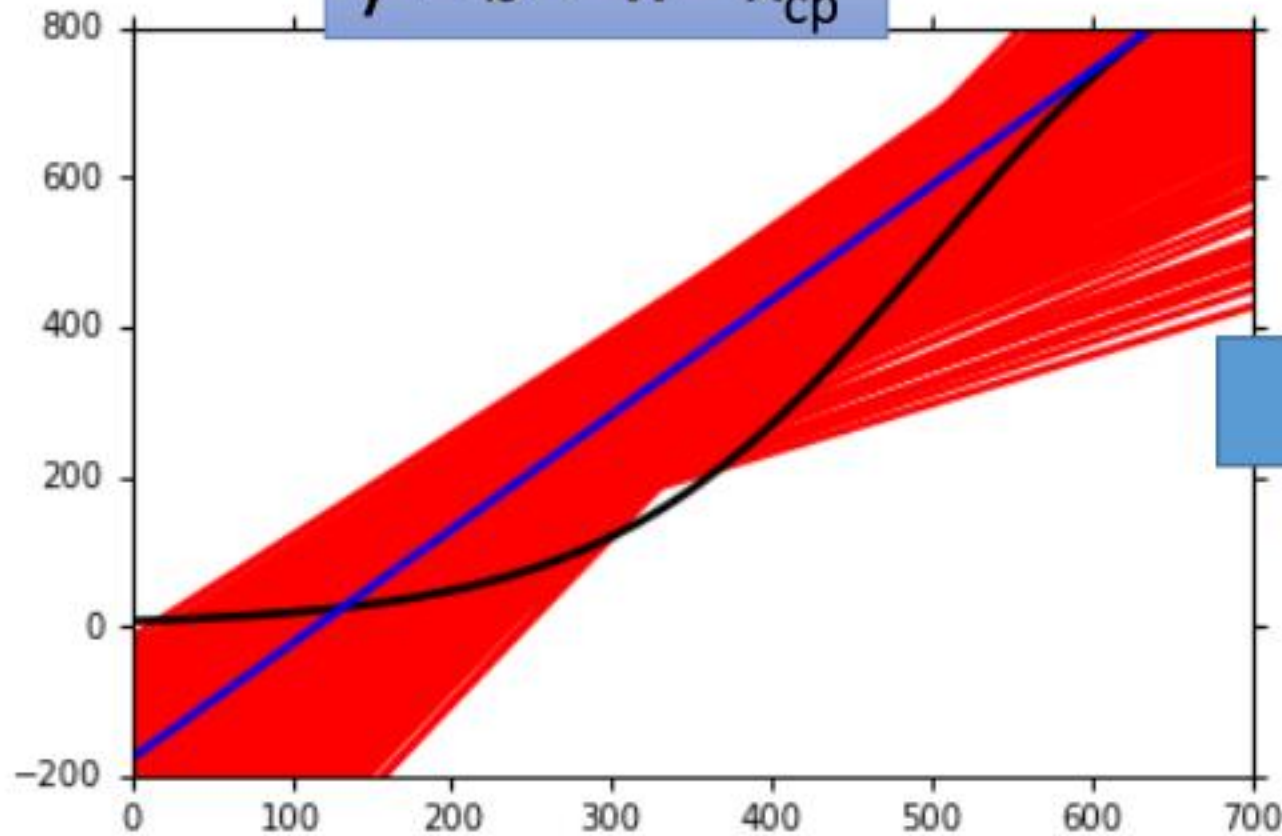
Red curves: 5000 f^*

Blue curve: the average of 5000 f^*
 $= \bar{f}$

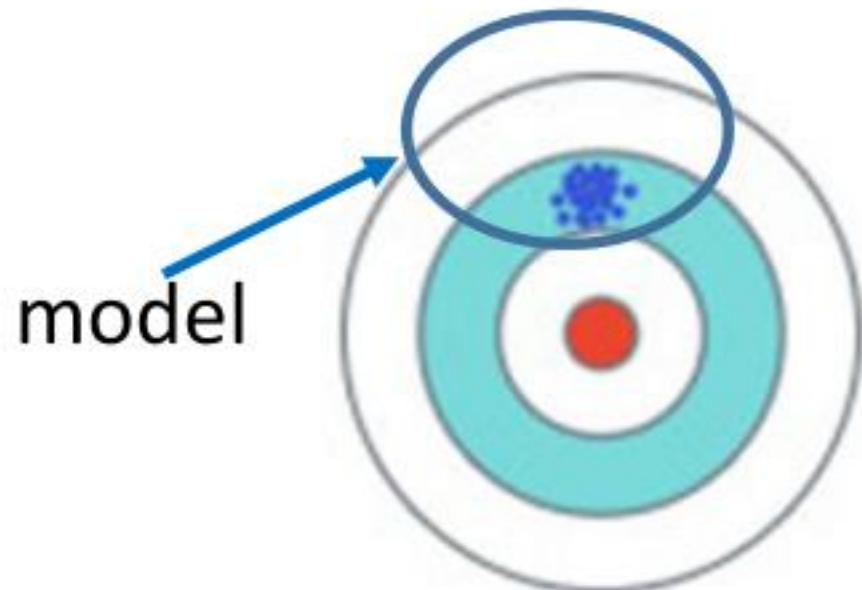
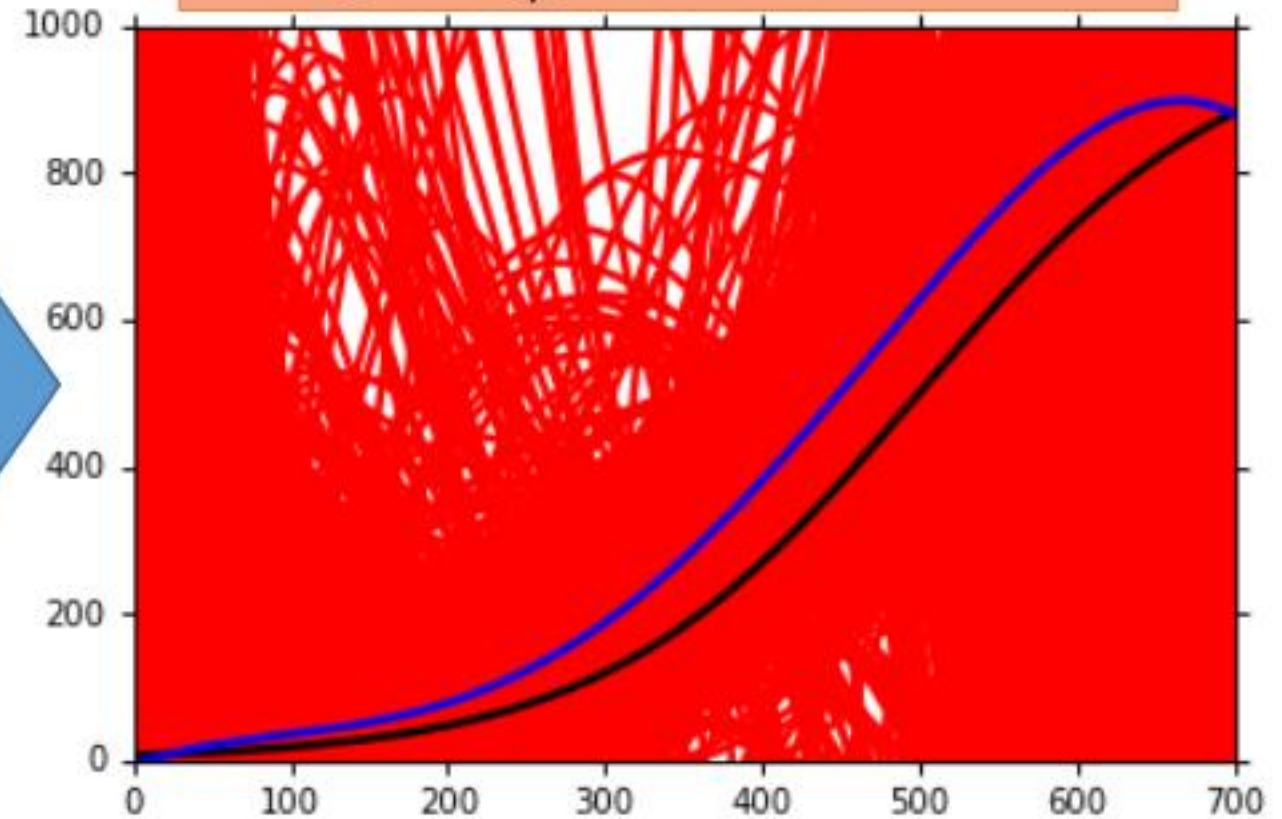


Bias

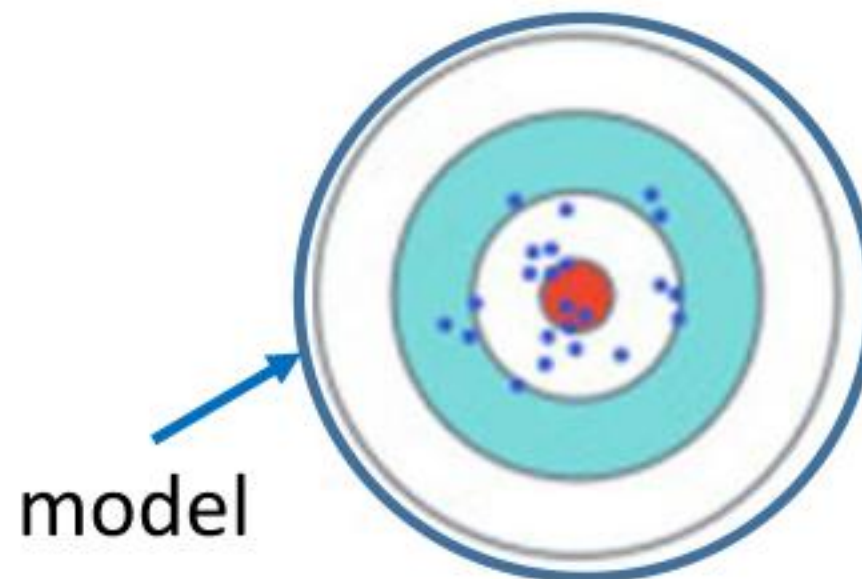
$$y = b + w \cdot x_{cp}$$



$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$$

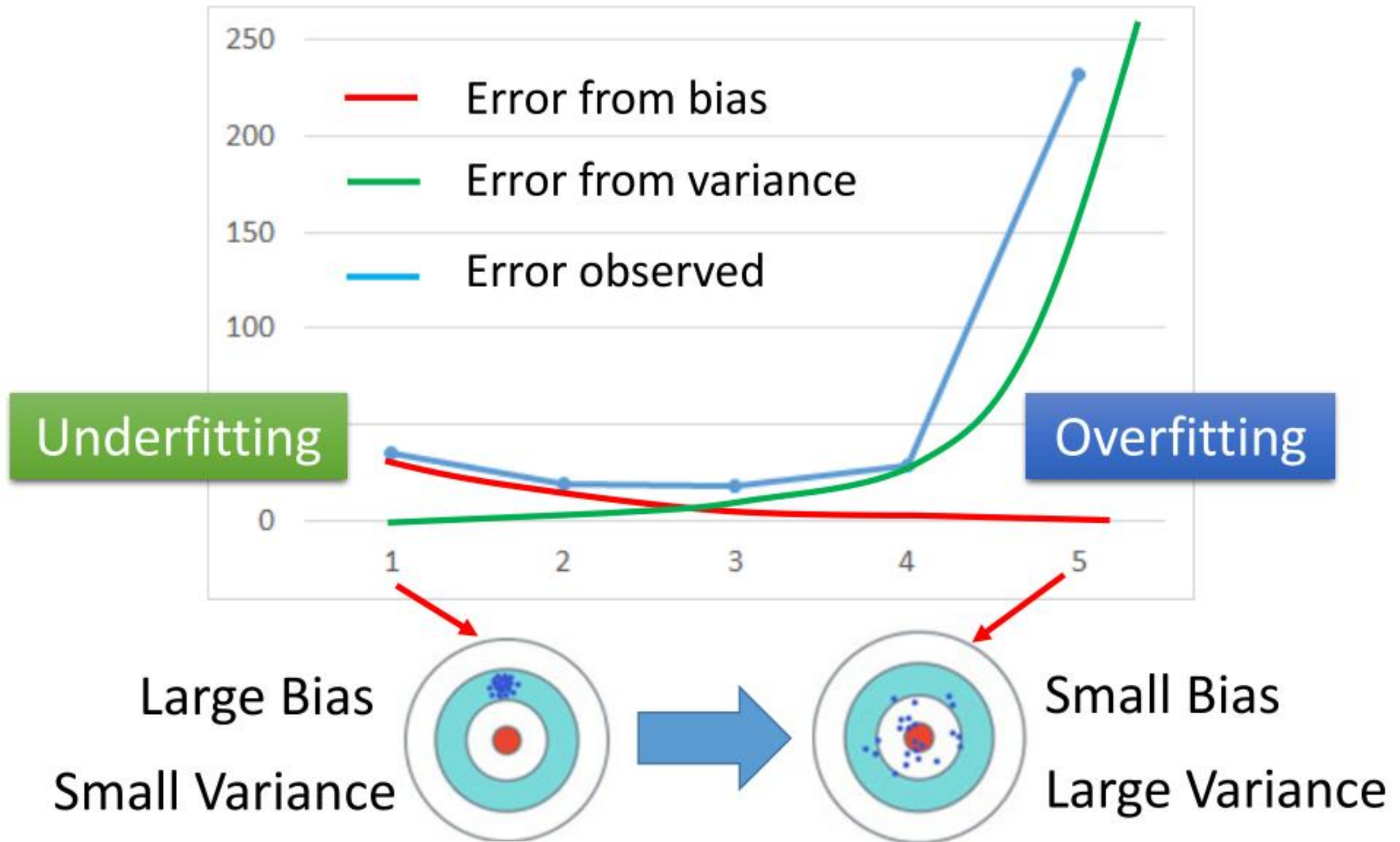


Large
Bias



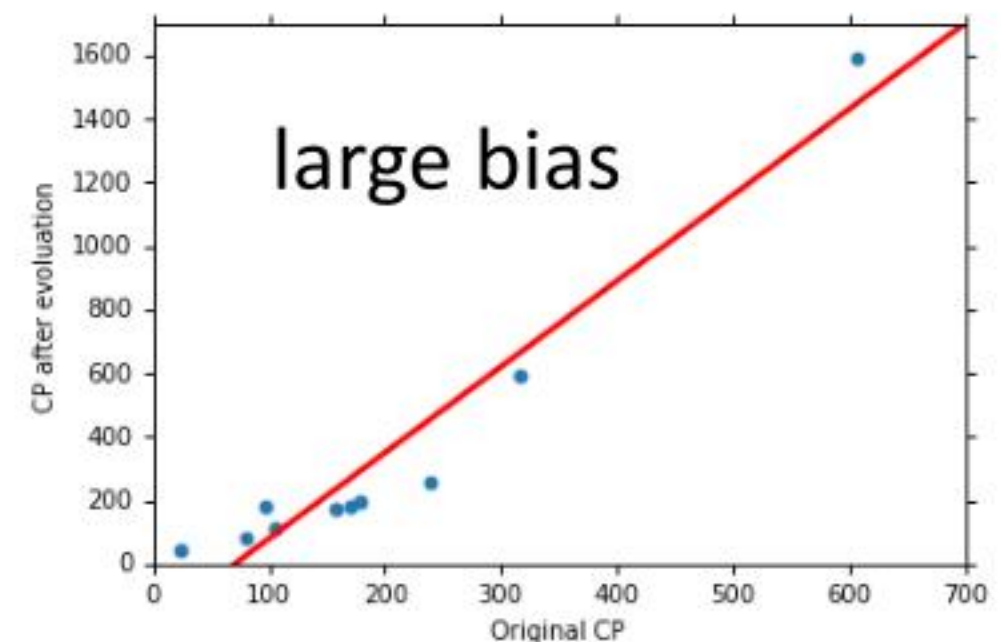
Small
Bias

Bias v.s. Variance



What to do with large bias?

- Diagnosis:
 - If your model cannot even fit the training examples, then you have large bias **Underfitting**
 - If you can fit the training data, but large error on testing data, then you probably have large variance **Overfitting**
- For bias, redesign your model:
 - Add more features as input
 - A more complex model



Overfitting

- Small loss on training data, large loss on testing data. Why?

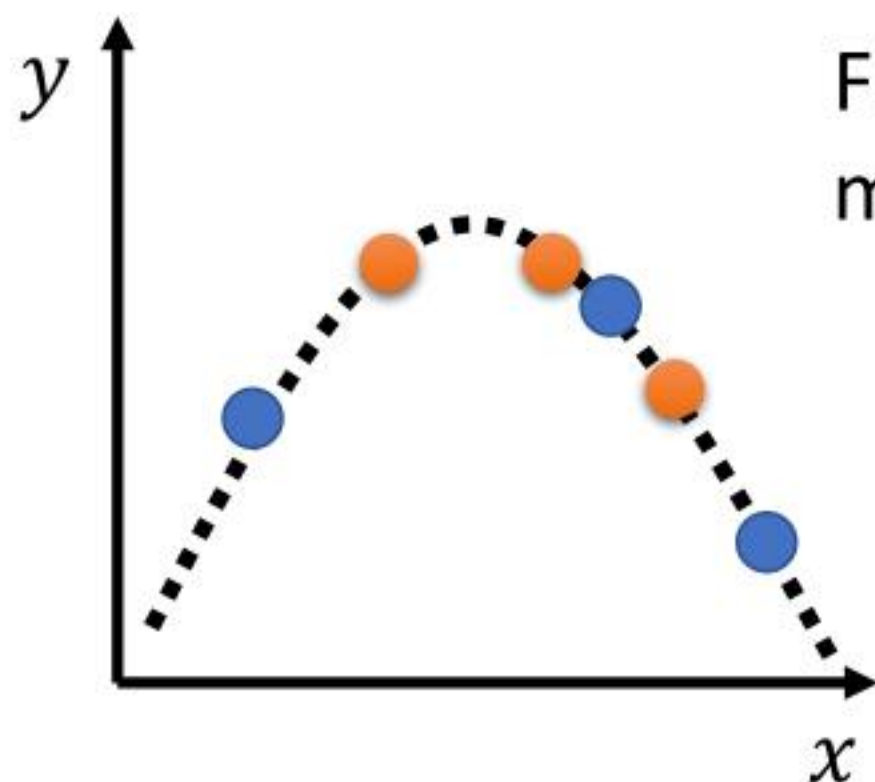
An extreme example

Training data: $\{(\mathbf{x}^1, \hat{y}^1), (\mathbf{x}^2, \hat{y}^2), \dots, (\mathbf{x}^N, \hat{y}^N)\}$

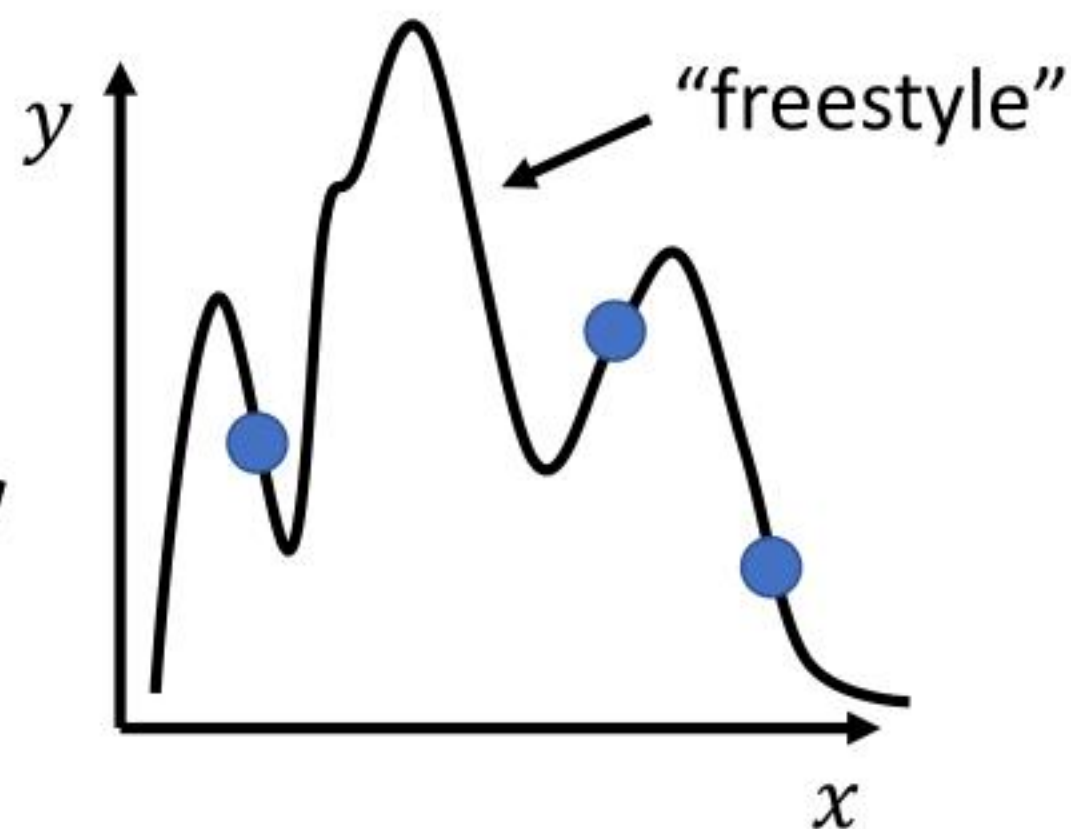
$$f(\mathbf{x}) = \begin{cases} \hat{y}^i & \exists \mathbf{x}^i = \mathbf{x} \\ random & otherwise \end{cases} \quad \text{Less than useless ...}$$

This function obtains **zero training loss**, but **large testing loss**.

Overfitting



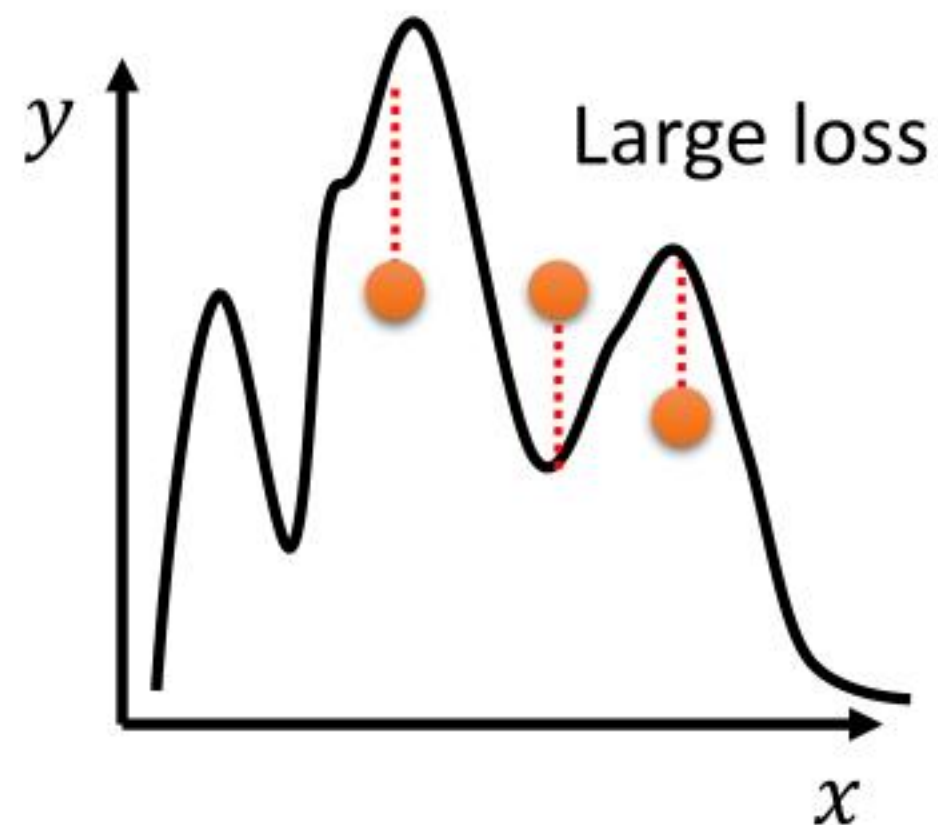
Flexible
model



■■■■ Real data distribution
(not observable)

● Training data

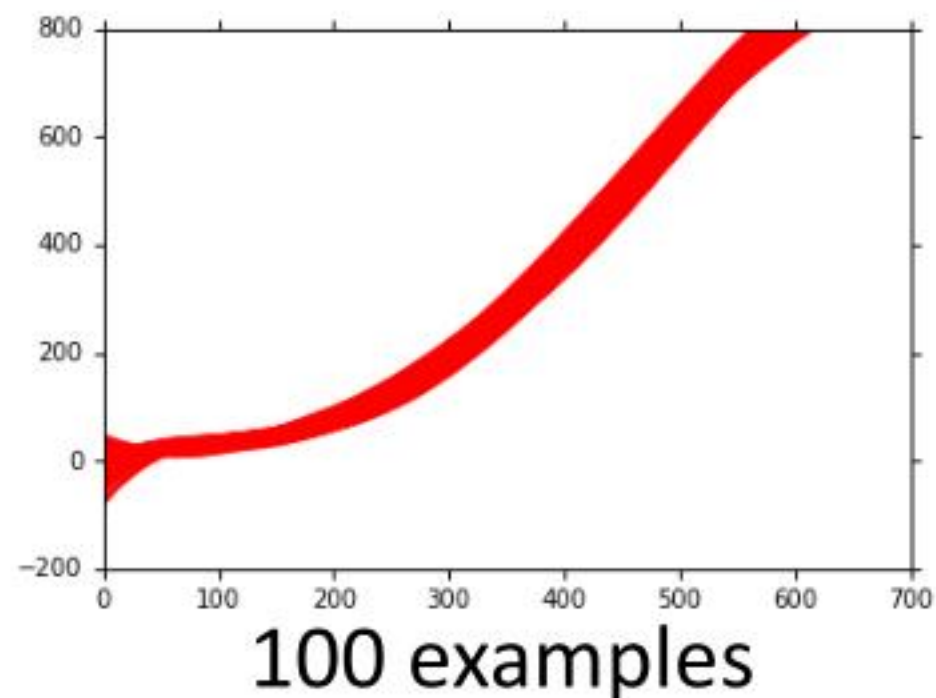
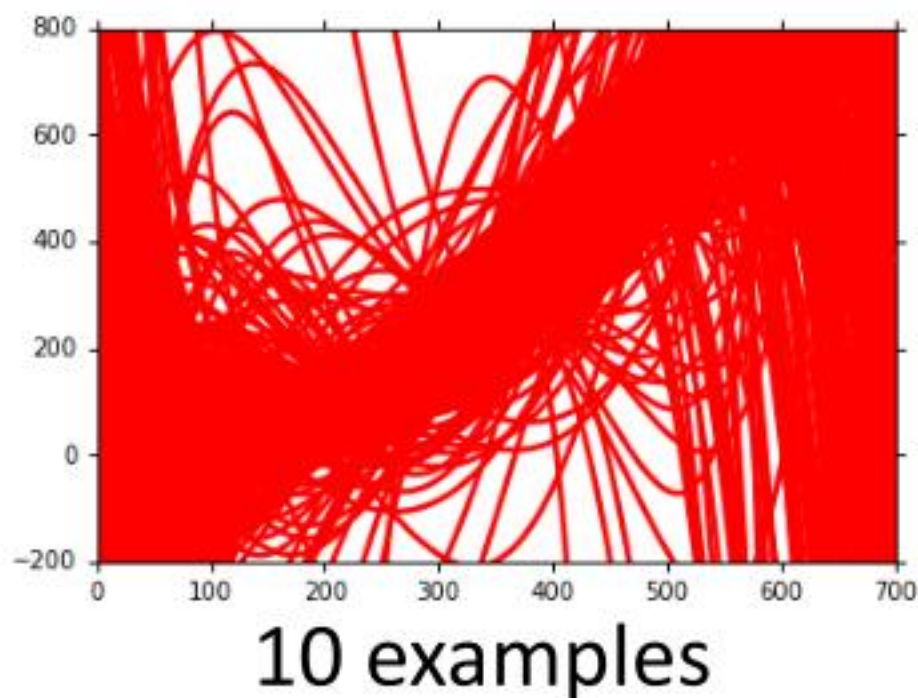
● Testing data



What to do with large variance?

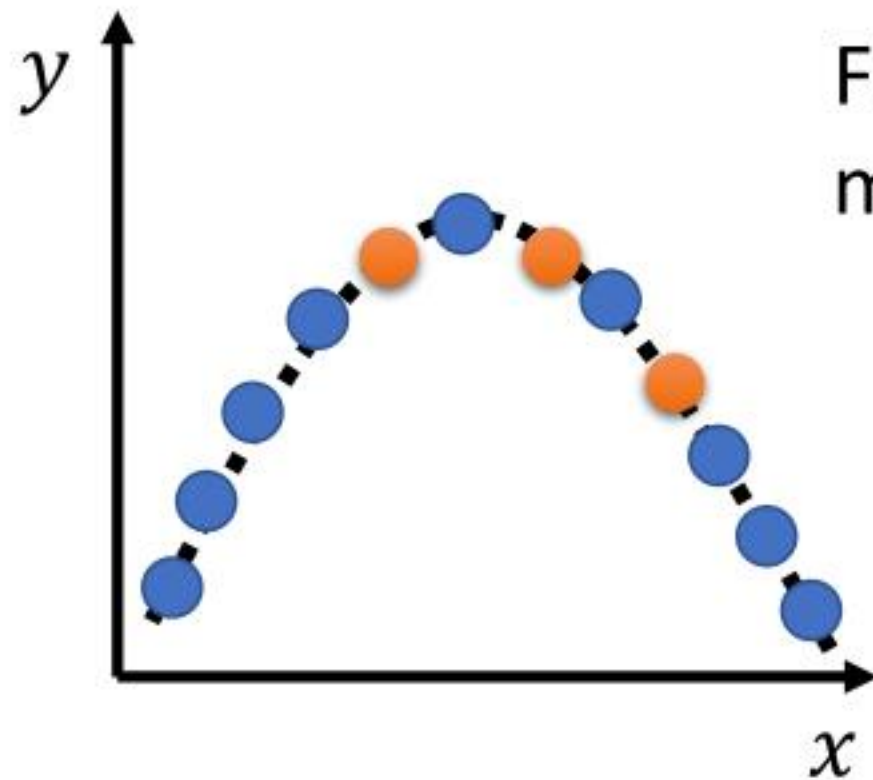
- More data

Very effective,
but not always
practical

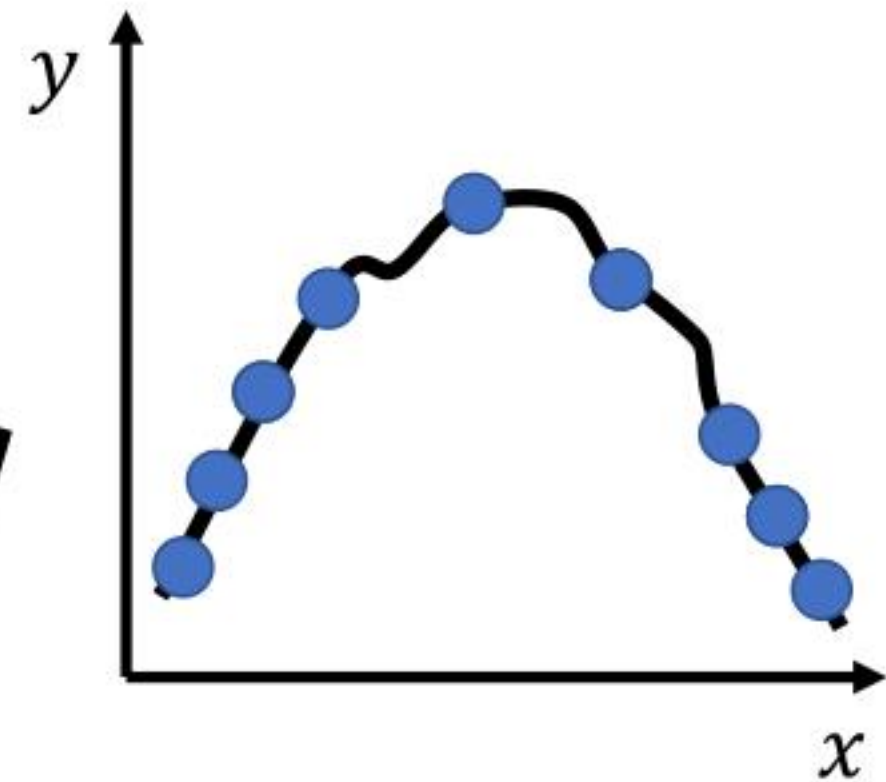


- Add constraints

Overfitting



Flexible
model

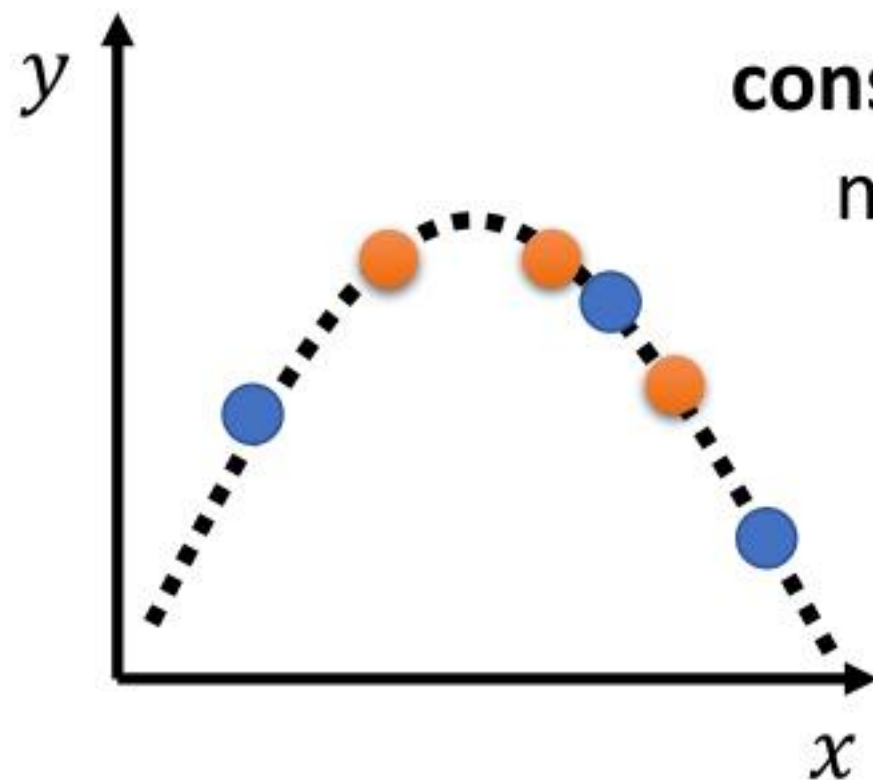


More training data
(cannot do it in HWs)

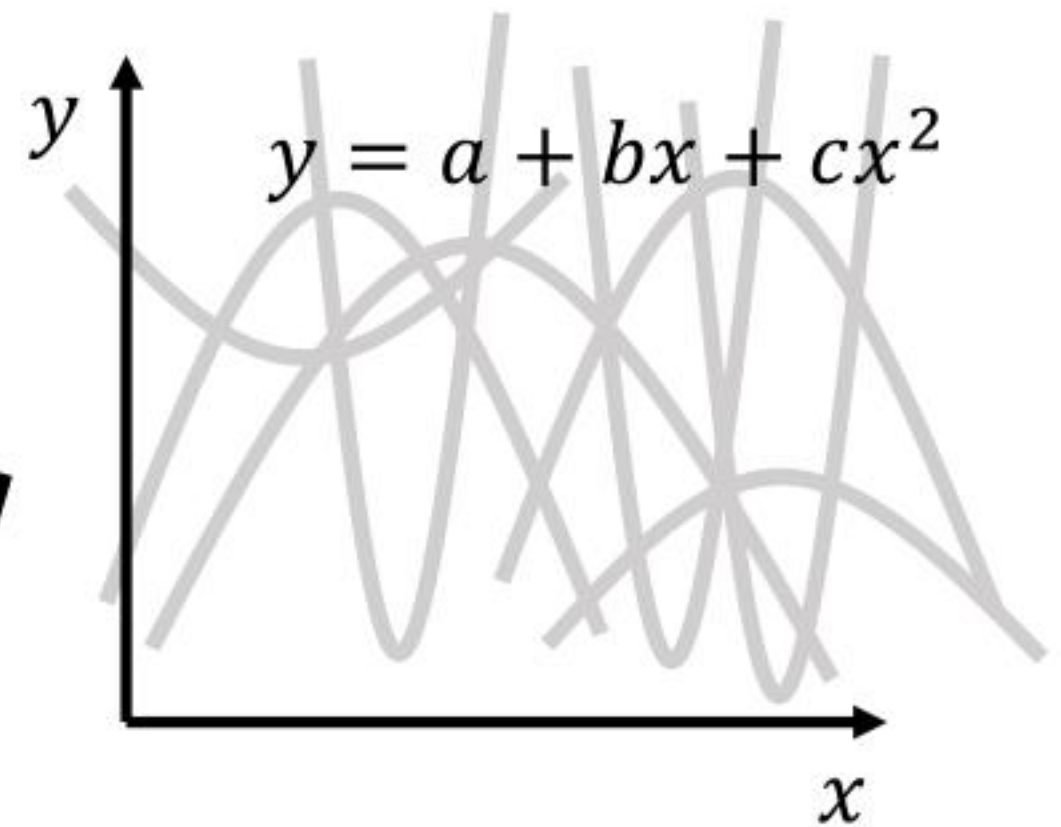
Data augmentation (you can do that in HWs)



Overfitting



**constrained
model**

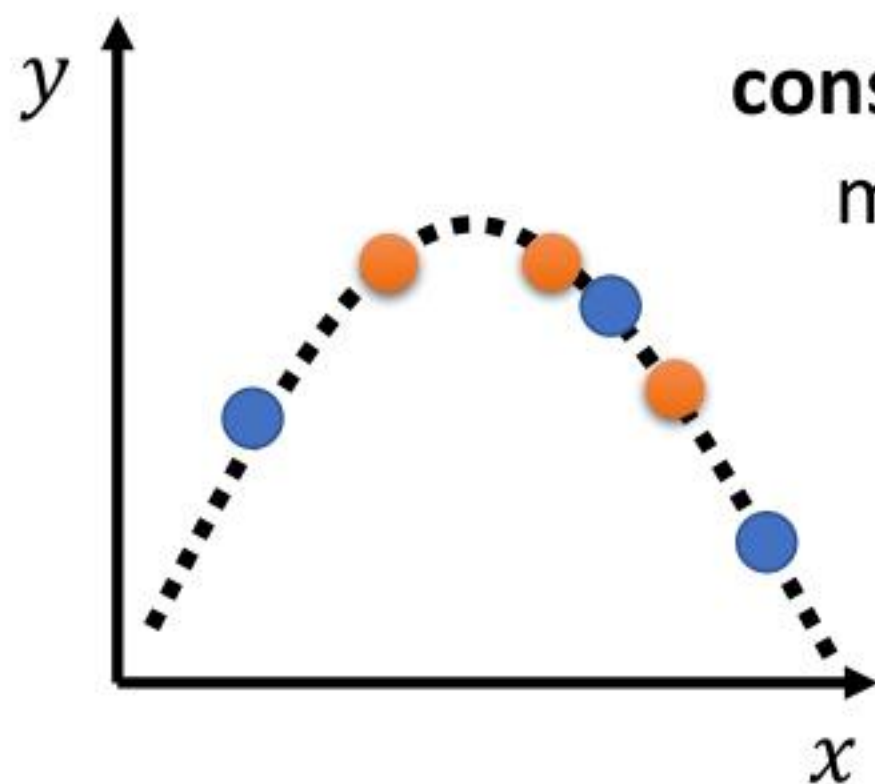


■■■■ Real data distribution
(not observable)

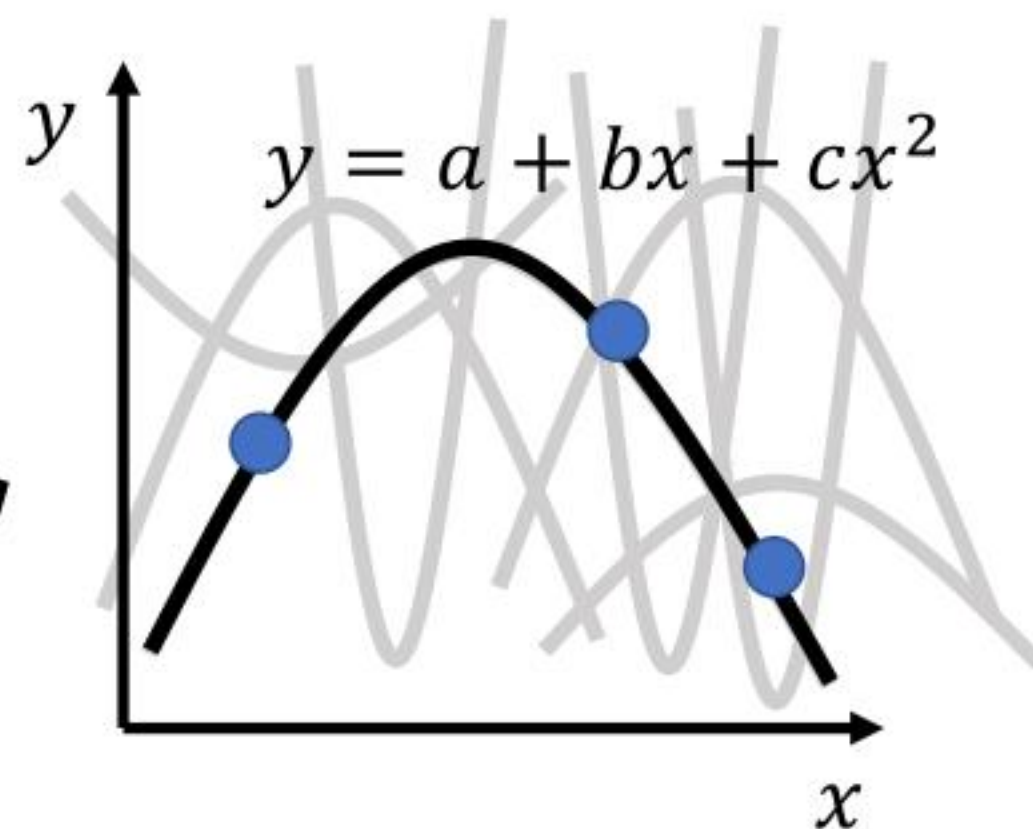
● Training data

● Testing data

Overfitting



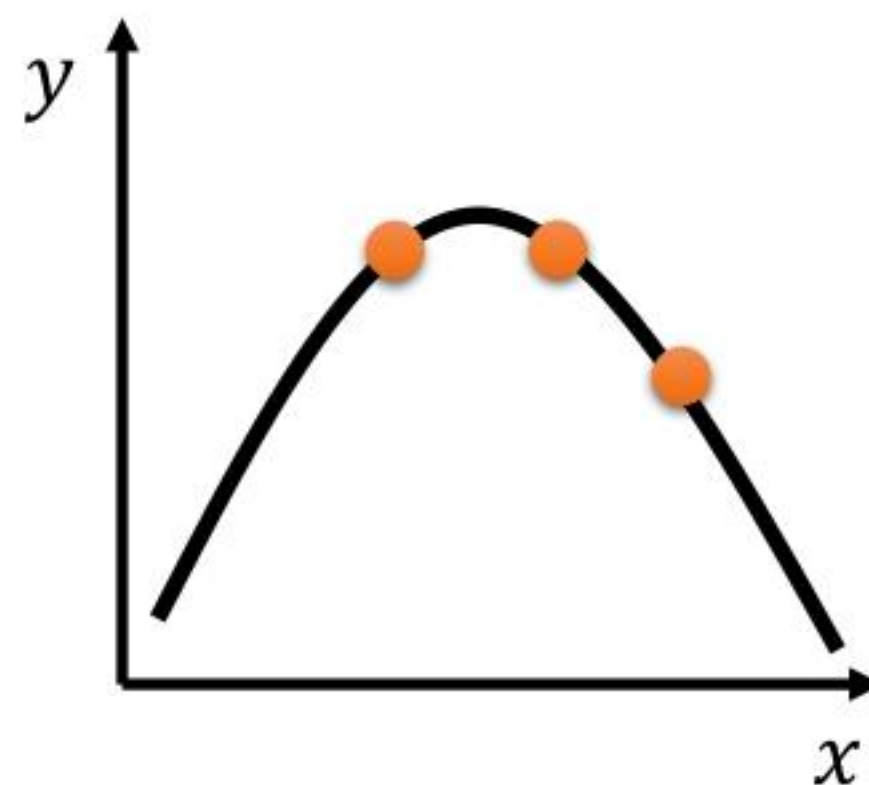
constrained
model



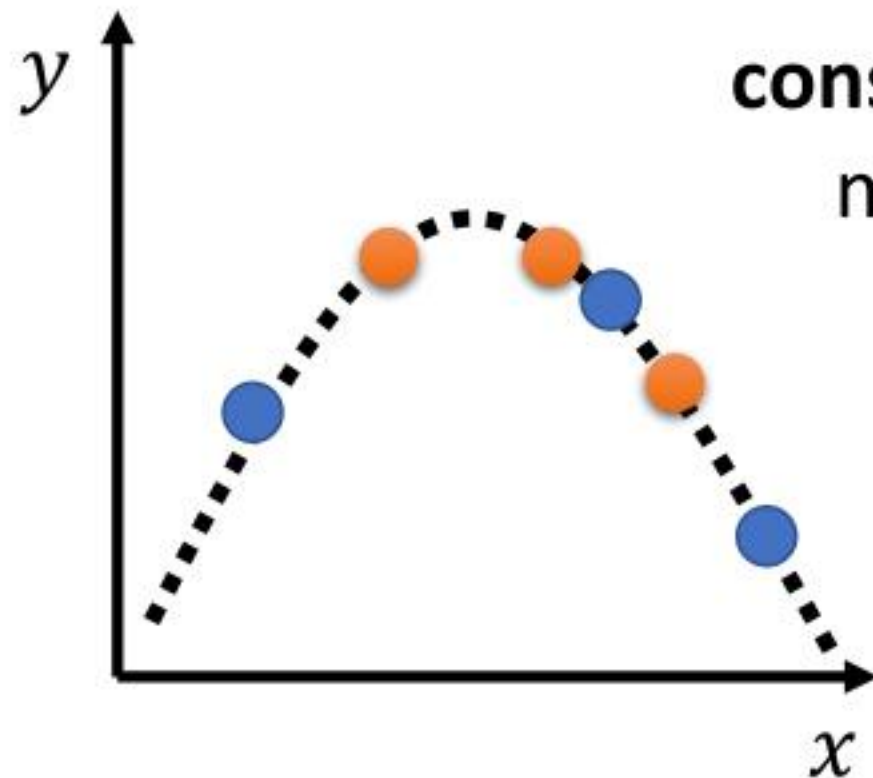
■■■■ Real data distribution
(not observable)

● Training data

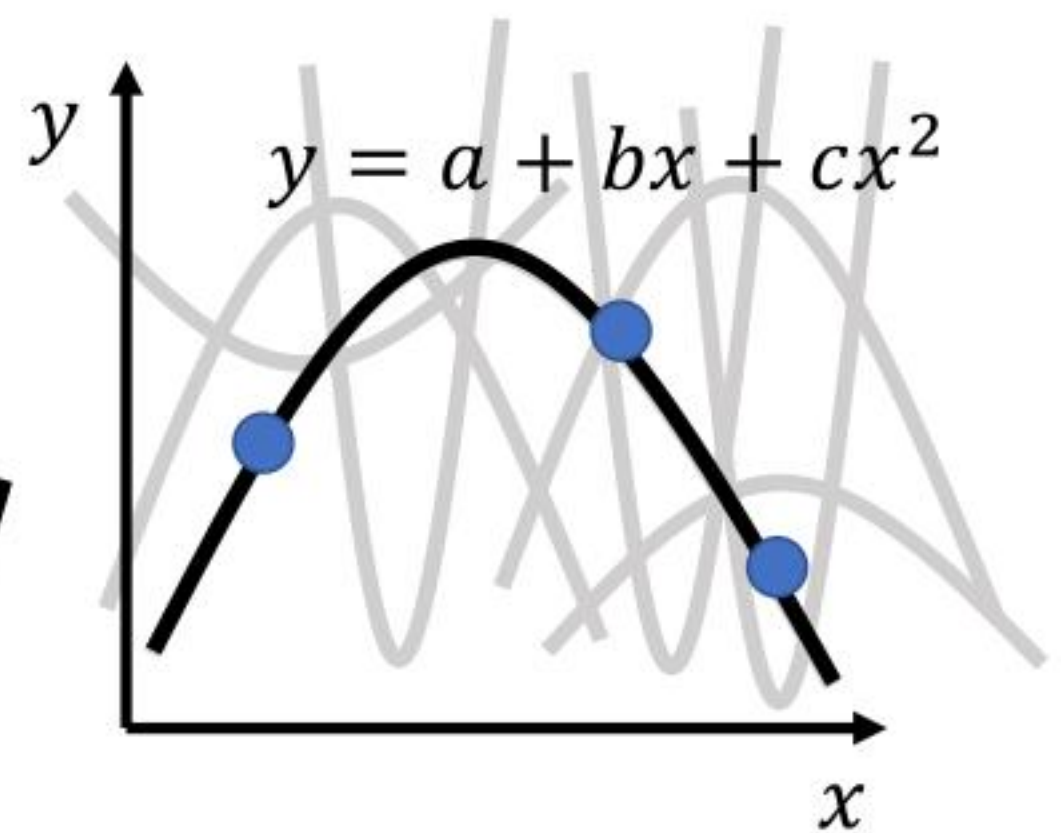
● Testing data



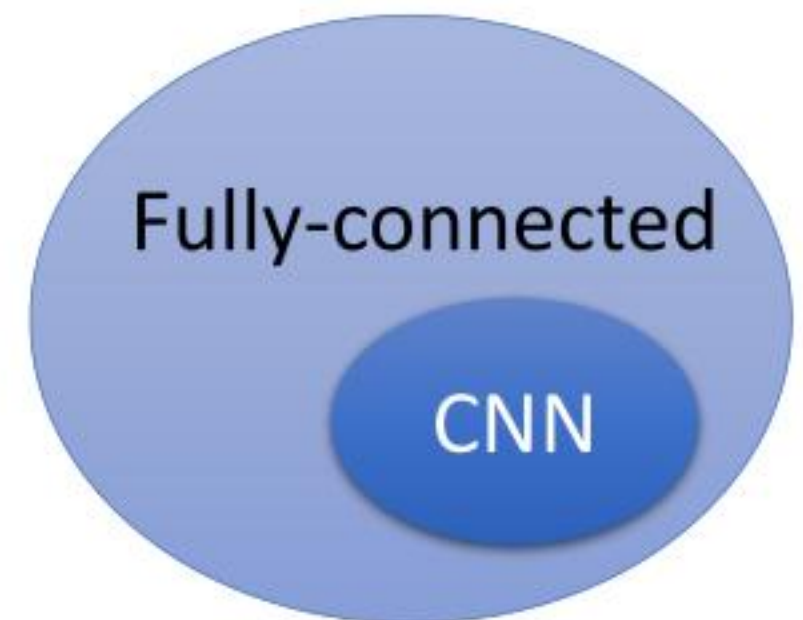
Overfitting



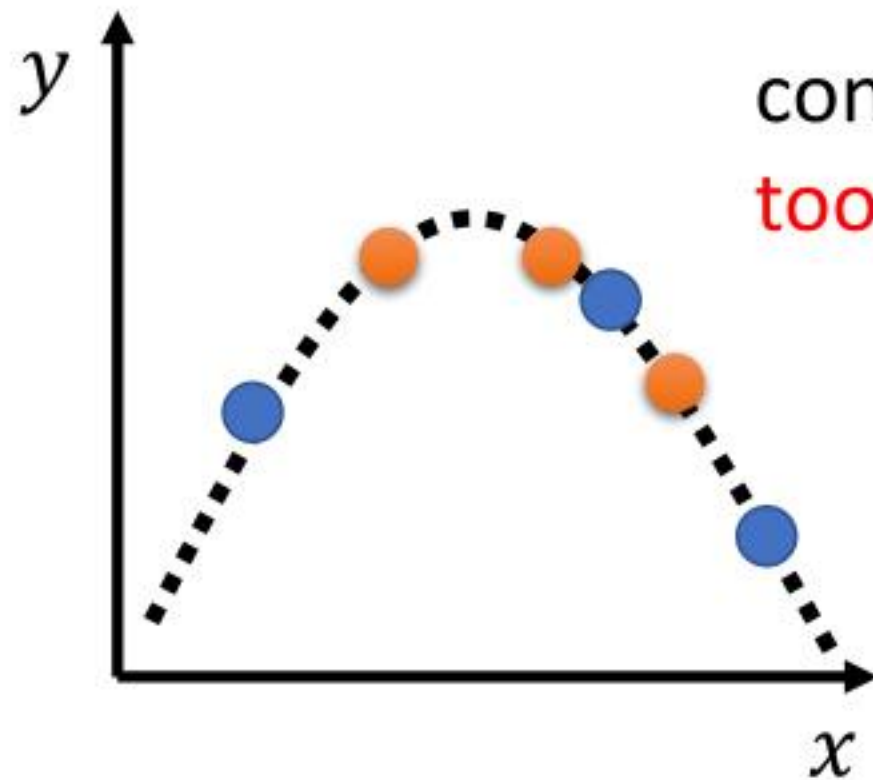
constrained
model



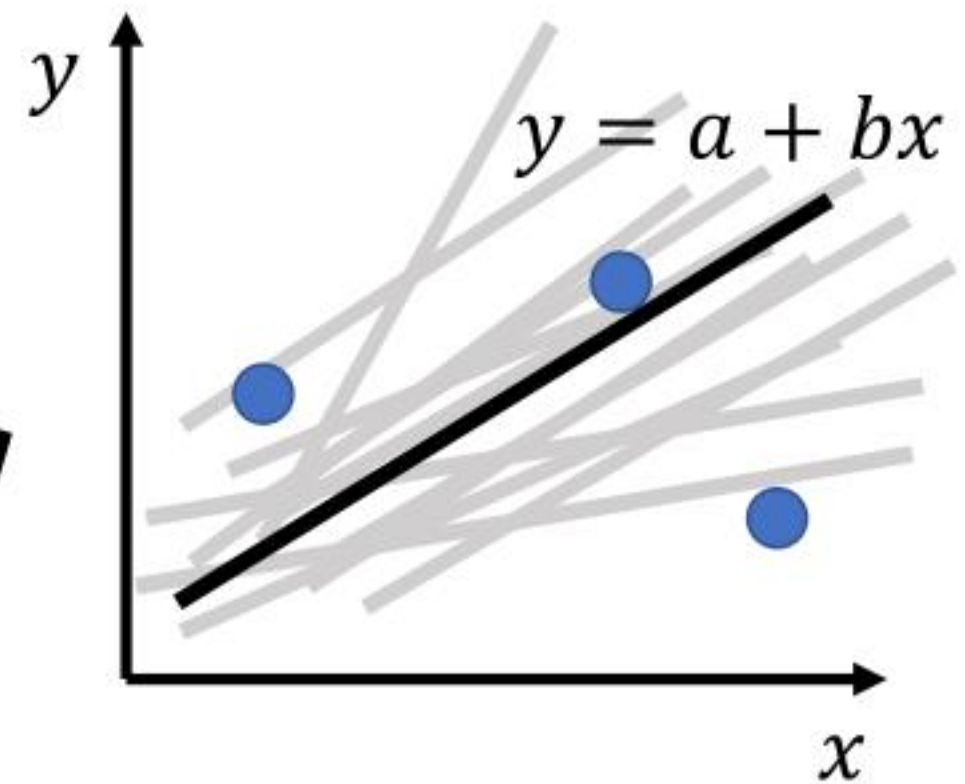
- Less parameters, sharing parameters
- Less features
- Early stopping
- Regularization
- Dropout



Overfitting



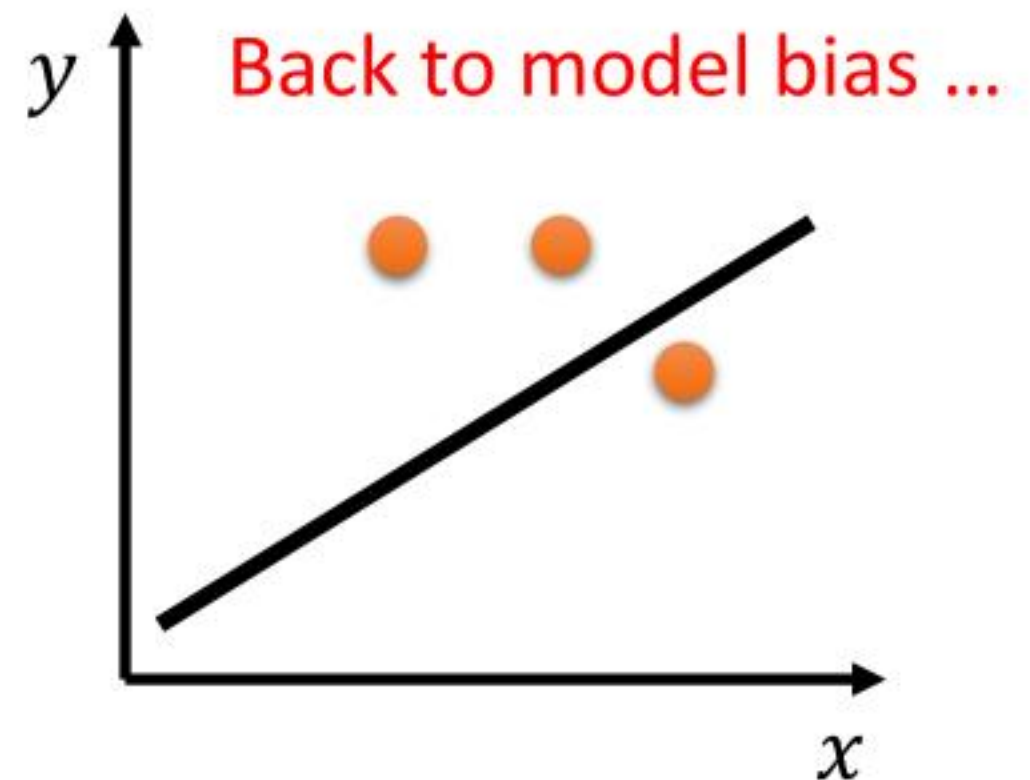
constrain
too much



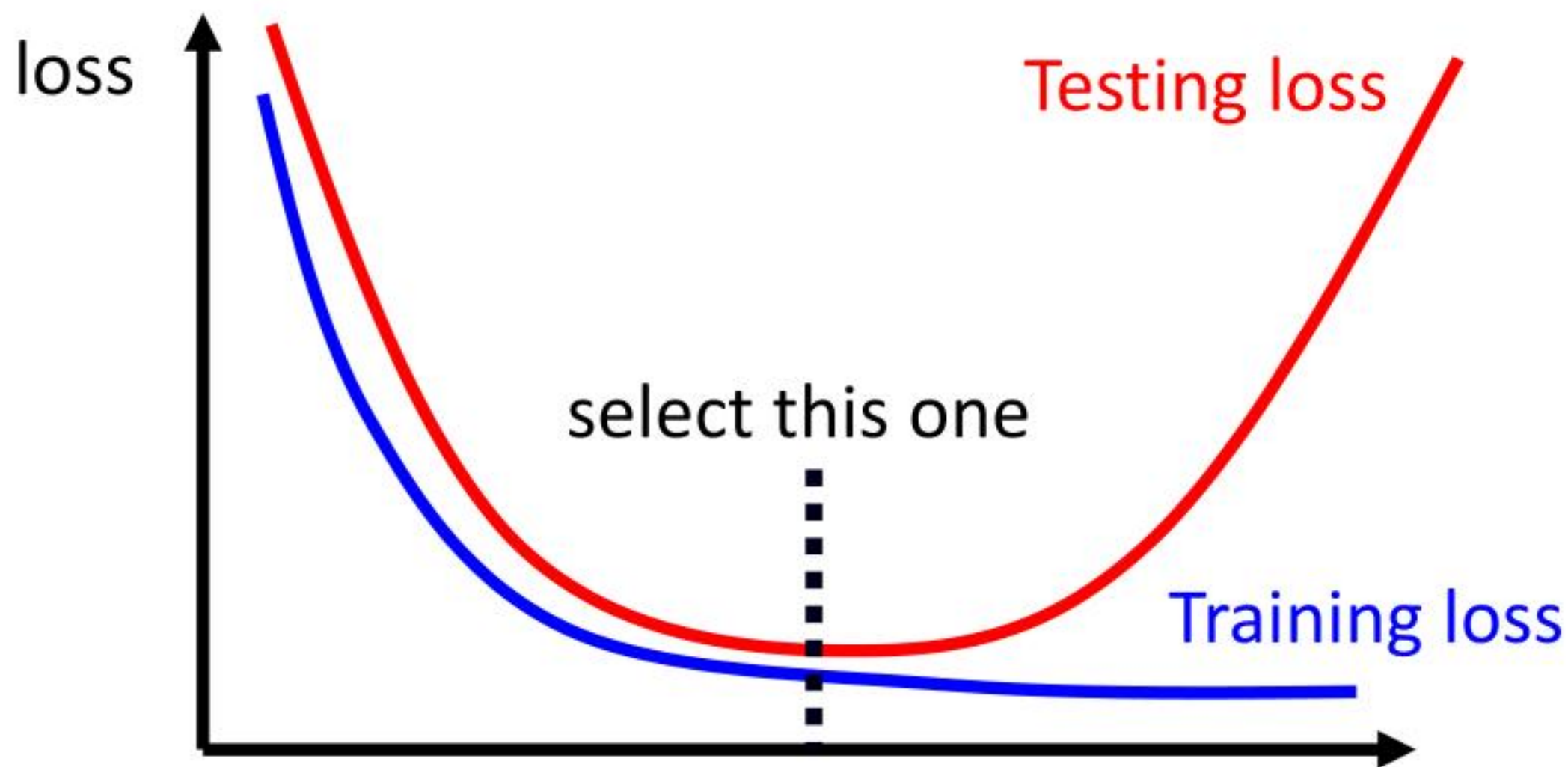
■■■■ Real data distribution
(not observable)

● Training data

● Testing data



Bias-Complexity Trade-off



Model becomes complex
(e.g. more features, more parameters)

Back to step 2: Regularization

$$y = b + \sum w_i x_i$$

$$L = \sum_n \left(\hat{y}^n - \left(b + \sum w_i x_i \right) \right)^2$$

The functions with smaller w_i are better

$$+ \lambda \sum (w_i)^2$$

➤ Smaller w_i means ...

smoother

$$y = b + \sum w_i x_i$$

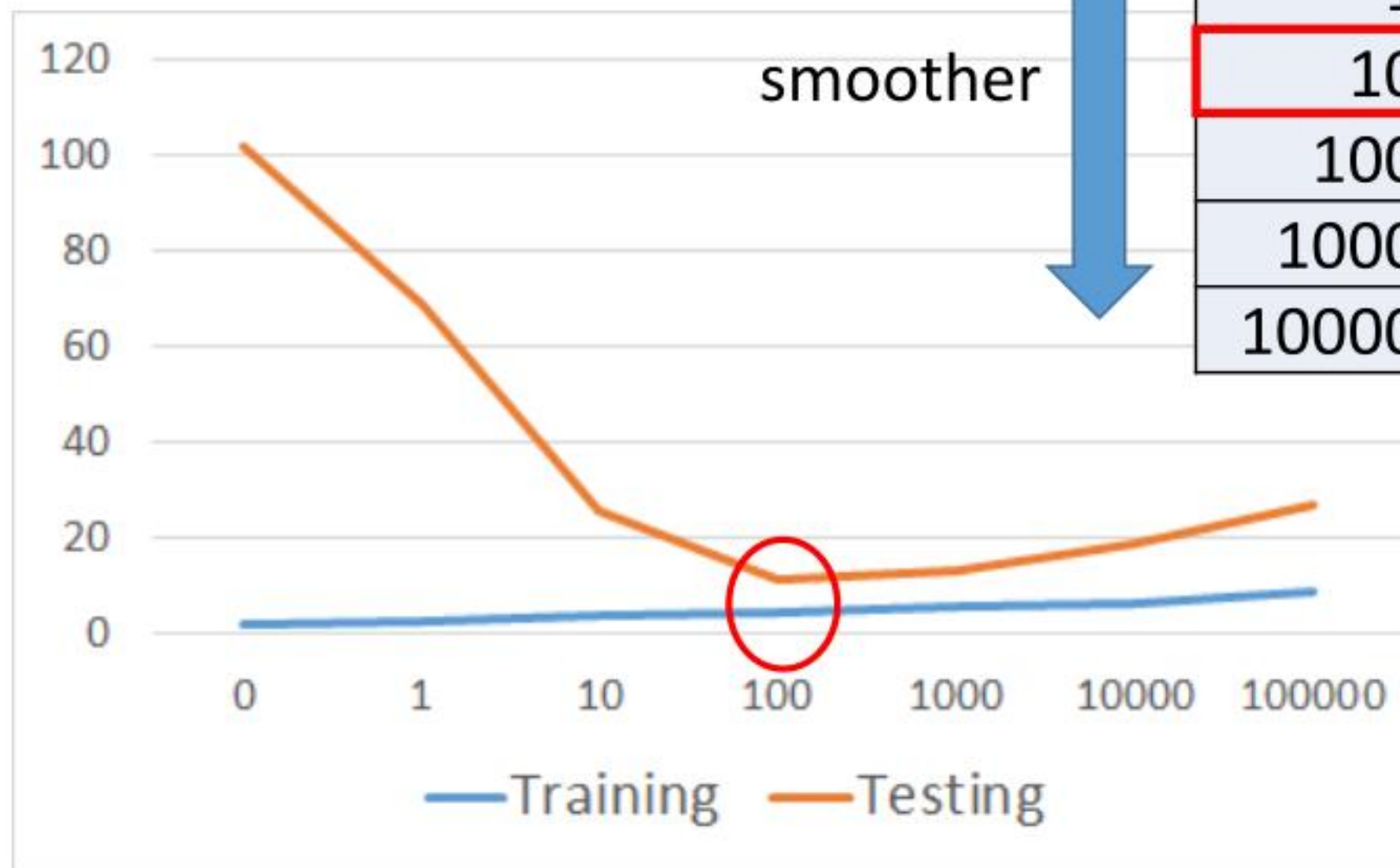
奥卡姆剃刀

$$y + \sum w_i \Delta x_i = b + \sum w_i (x_i + \Delta x_i)$$

➤ We believe smoother function is more likely to be correct

Do you have to apply regularization on bias?

Regularization



| λ | Training | Testing |
|-----------|----------|---------|
| 0 | 1.9 | 102.3 |
| 1 | 2.3 | 68.7 |
| 10 | 3.5 | 25.7 |
| 100 | 4.1 | 11.1 |
| 1000 | 5.6 | 12.8 |
| 10000 | 6.3 | 18.7 |
| 100000 | 8.5 | 26.8 |

How smooth?

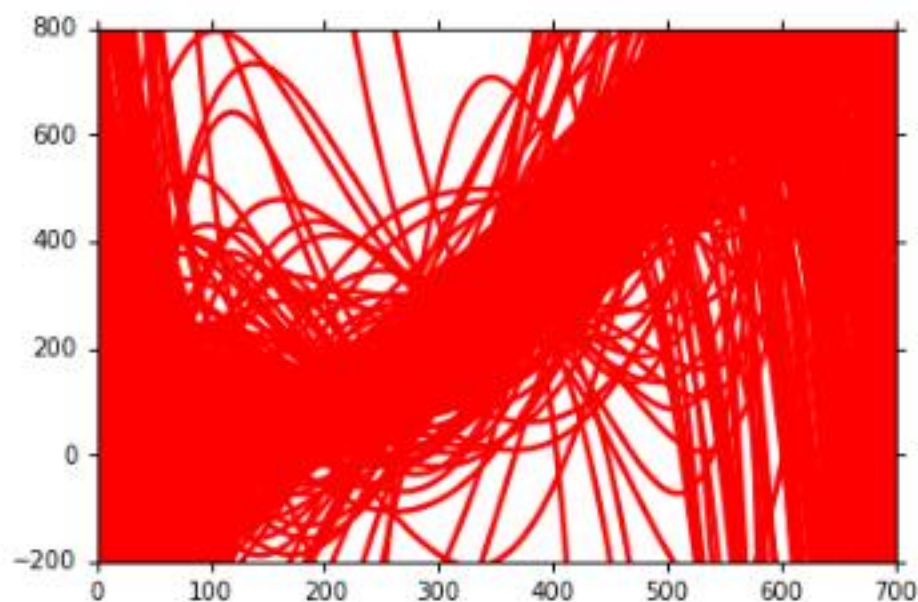
Select λ obtaining the best model

- Training error: larger λ , considering the training error less
- We prefer smooth function, but don't be too smooth.

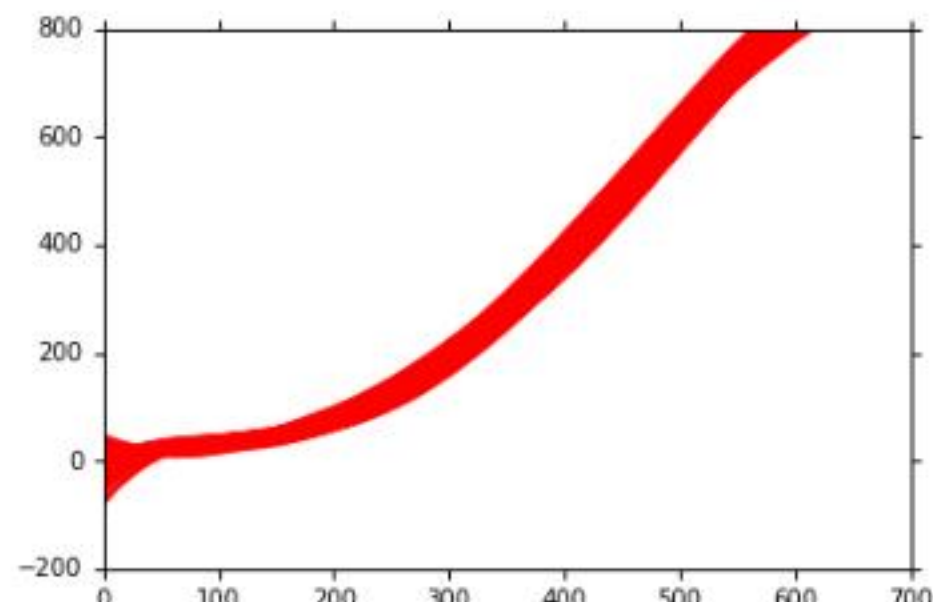
What to do with large variance?

- More data

Very effective,
but not always
practical



10 examples

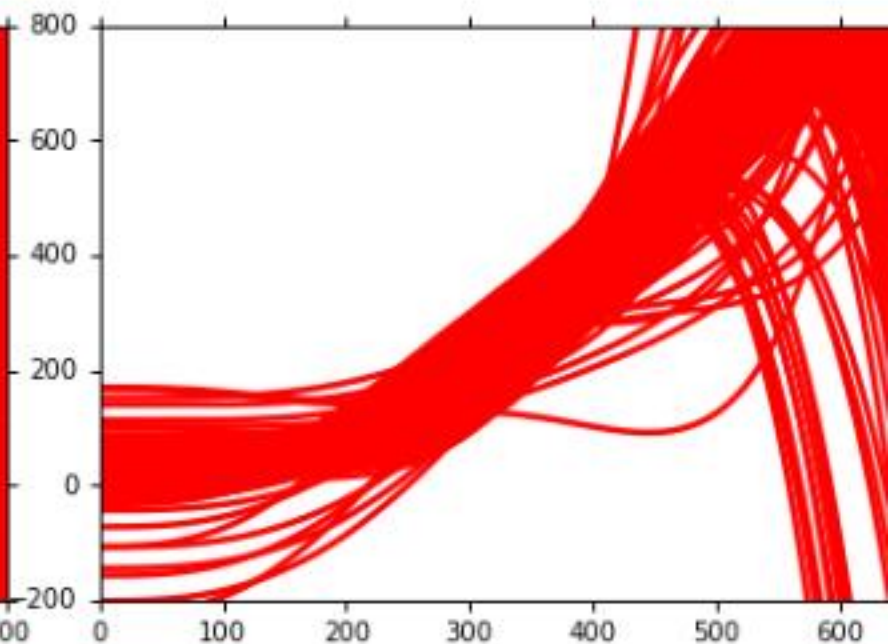
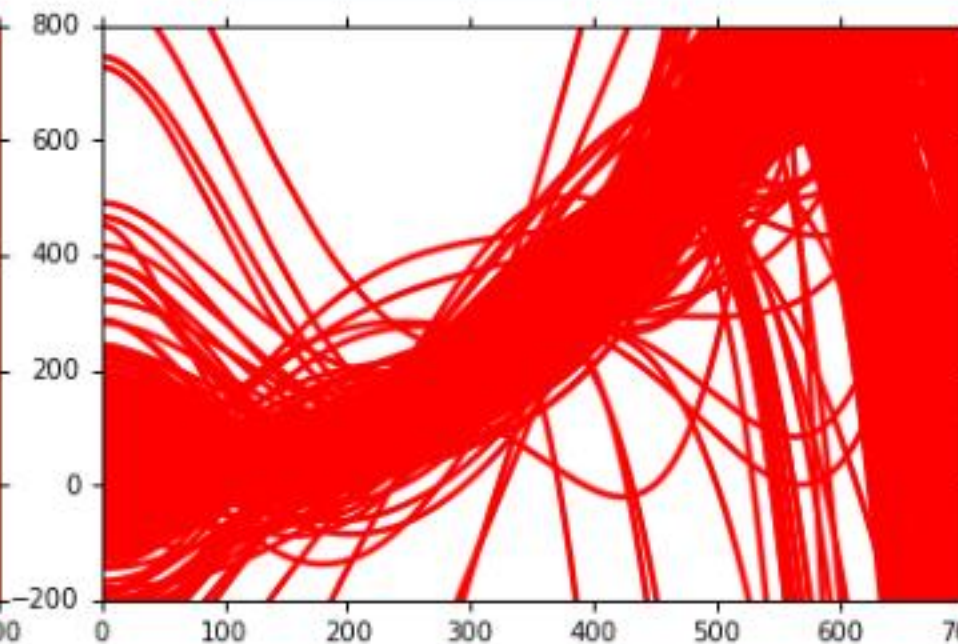
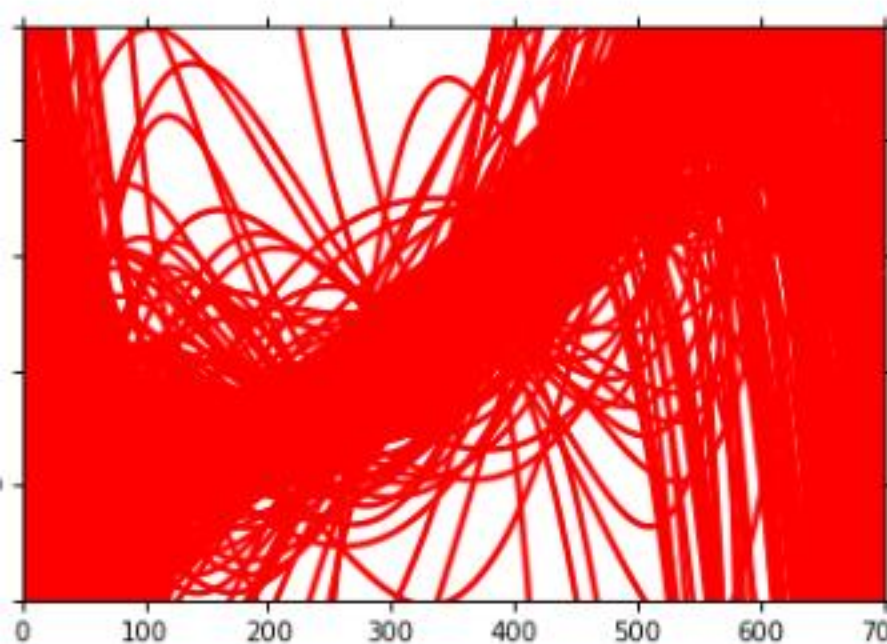


100 examples

- Regularization

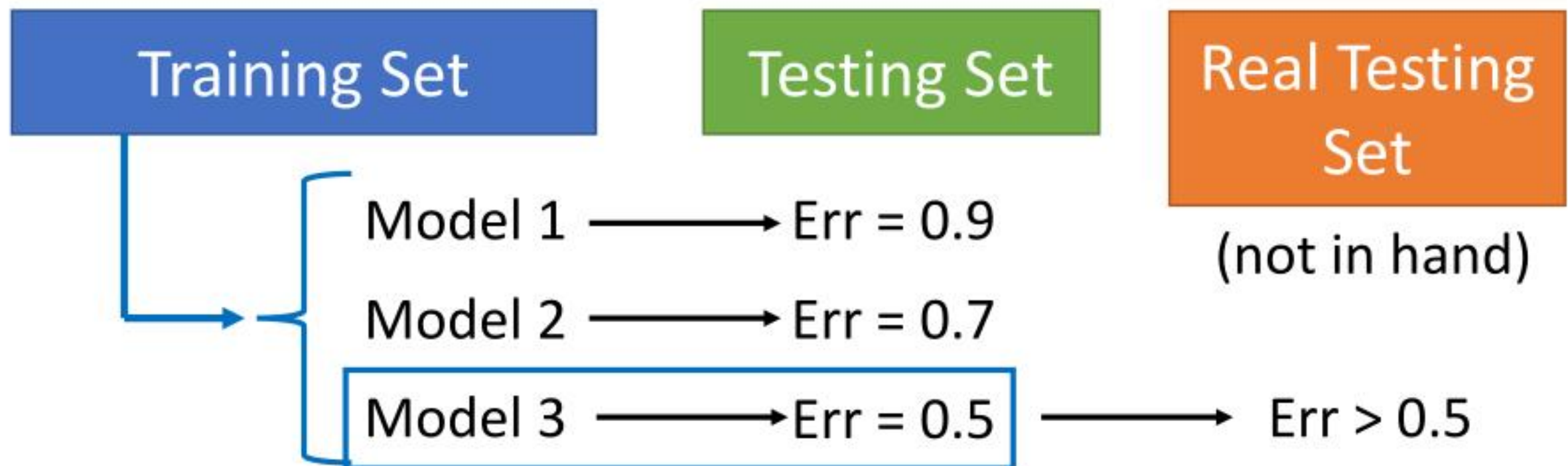


May increase bias

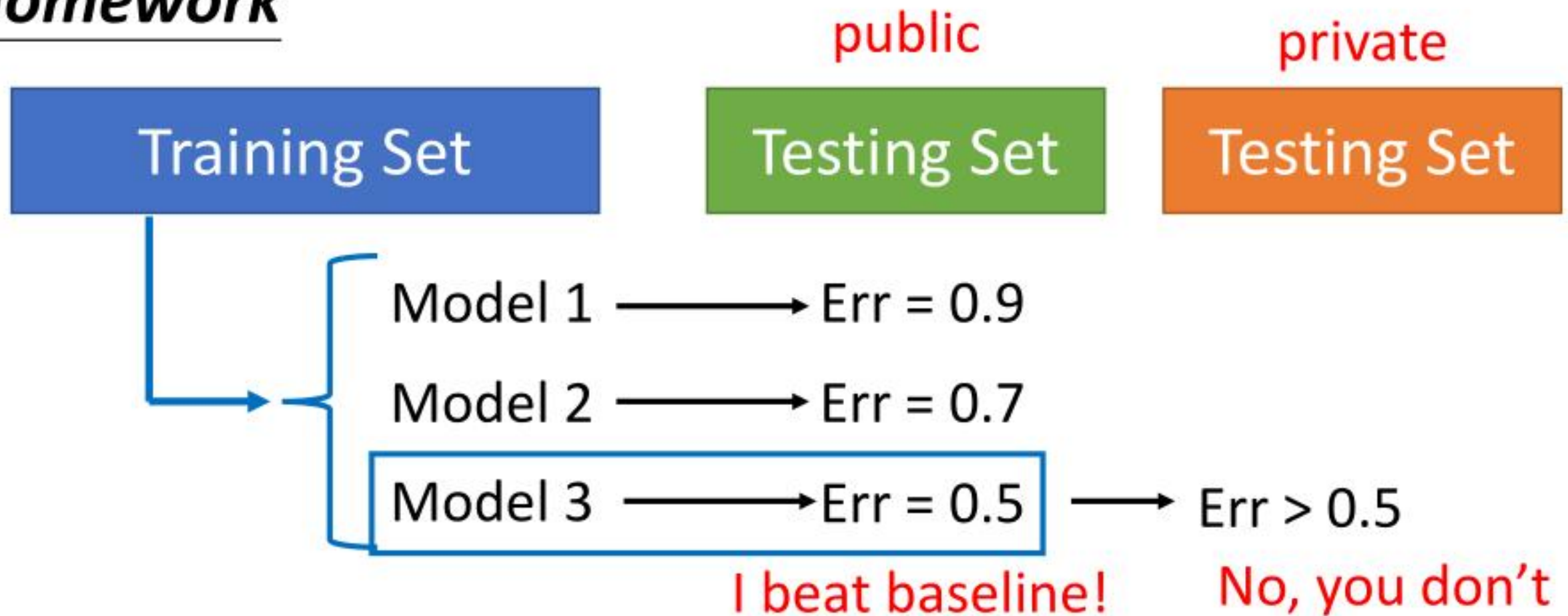


Model Selection

- There is usually a trade-off between bias and variance.
- Select a model that balances two kinds of error to minimize total error
- What you should NOT do:



Homework

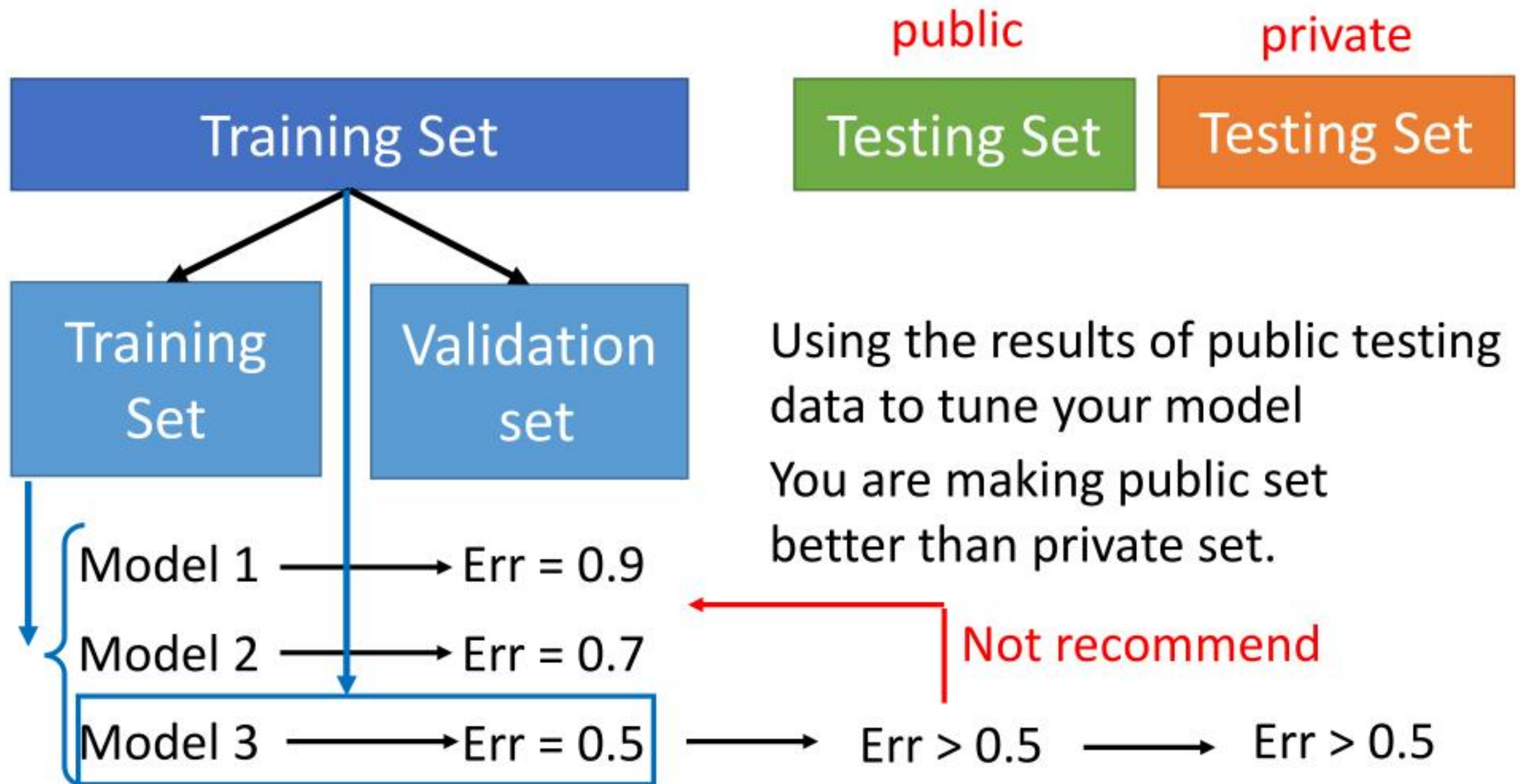


What will happen?

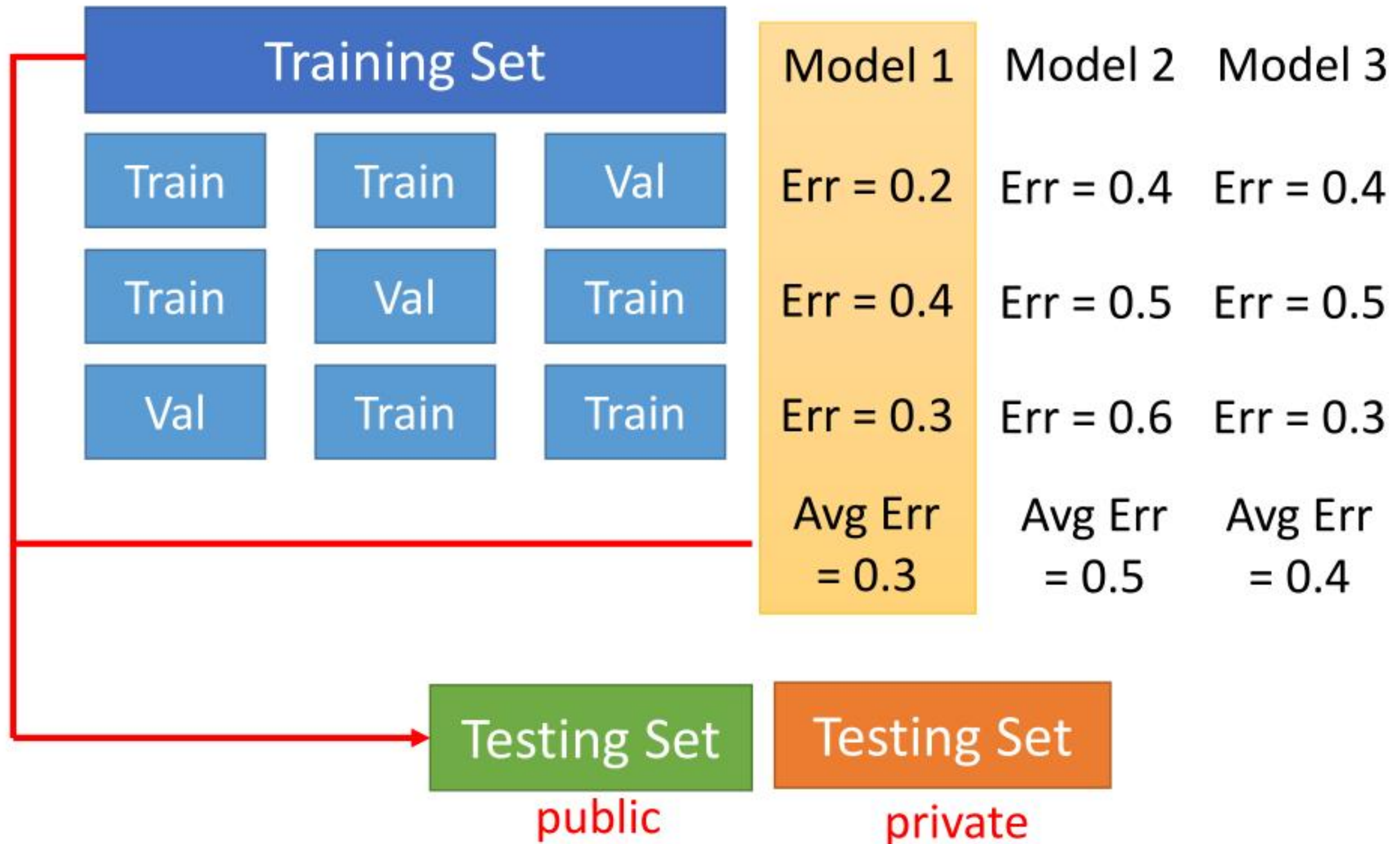
<http://www.chioka.in/how-to-select-your-final-models-in-a-kaggle-competitio/>



Cross Validation



N-fold Cross Validation



More about Validation Set

I used a validation set,
but my model still overfitted?

Validation Set

Training Set \mathcal{D}_{train}

Model \mathcal{H}_1 $h_1^* = \arg \min_{h \in \mathcal{H}_1} L(h, \mathcal{D}_{train})$

Model \mathcal{H}_2 $h_2^* = \arg \min_{h \in \mathcal{H}_2} L(h, \mathcal{D}_{train})$

Model \mathcal{H}_3 $h_3^* = \arg \min_{h \in \mathcal{H}_3} L(h, \mathcal{D}_{train})$

Validation Set \mathcal{D}_{val}

$$L(h_1^*, \mathcal{D}_{val}) = 0.9$$

$$L(h_2^*, \mathcal{D}_{val}) = 0.7$$

$$L(h_3^*, \mathcal{D}_{val}) = 0.5$$



Testing Set \mathcal{D}_{test}

Approximation of \mathcal{D}_{all}

Training Set \mathcal{D}_{train}

Model \mathcal{H}_1 $h_1^* = \arg \min_{h \in \mathcal{H}_1} L(h, \mathcal{D}_{train})$

Model \mathcal{H}_2 $h_2^* = \arg \min_{h \in \mathcal{H}_2} L(h, \mathcal{D}_{train})$

Model \mathcal{H}_3 $h_3^* = \arg \min_{h \in \mathcal{H}_3} L(h, \mathcal{D}_{train})$

Validation Set \mathcal{D}_{val}

$$L(h_1^*, \mathcal{D}_{val}) = 0.9$$

$$L(h_2^*, \mathcal{D}_{val}) = 0.7$$

$$L(h_3^*, \mathcal{D}_{val}) = 0.5$$

$$\mathcal{H}_{val} = \{h_1^*, h_2^*, h_3^*\} \quad h^* = \arg \min_{h \in \mathcal{H}_{val}} L(h, \mathcal{D}_{val})$$

Using validation set to select model =
considered as “*training*” by \mathcal{D}_{val}

Your model is $\mathcal{H}_{val} = \{h_1^*, h_2^*, h_3^*\}$

Using validation set to select model =

considered as “*training*” by \mathcal{D}_{val}

Your model is $\mathcal{H}_{val} = \{h_1^*, h_2^*, h_3^*\}$

$$L(h^{train}, \mathcal{D}_{all}) - L(h^{all}, \mathcal{D}_{all}) \leq \delta$$

$$P(\mathcal{D}_{train} \text{ is } \mathbf{bad}) \leq |\mathcal{H}| \cdot 2\exp(-2N\varepsilon^2)$$

$$L(h^{val}, \mathcal{D}_{all}) - L(h^{all}, \mathcal{D}_{all}) \leq \delta$$

$$P(\mathcal{D}_{val} \text{ is } \mathbf{bad}) \leq |\mathcal{H}_{val}| \cdot 2\exp(-2N_{val}\varepsilon^2)$$



It is small.

Hopefully 😊