Bias-Variance Decomposition

Machine Learning Course - CS-433
Oct 2, 2024
Martin Jaggi & Nicolas Flammarion

EPFL

Last time

How can we judge if a given predictor is good?
How to select the best models of a family?

➡Bound the difference between the true and empirical risks

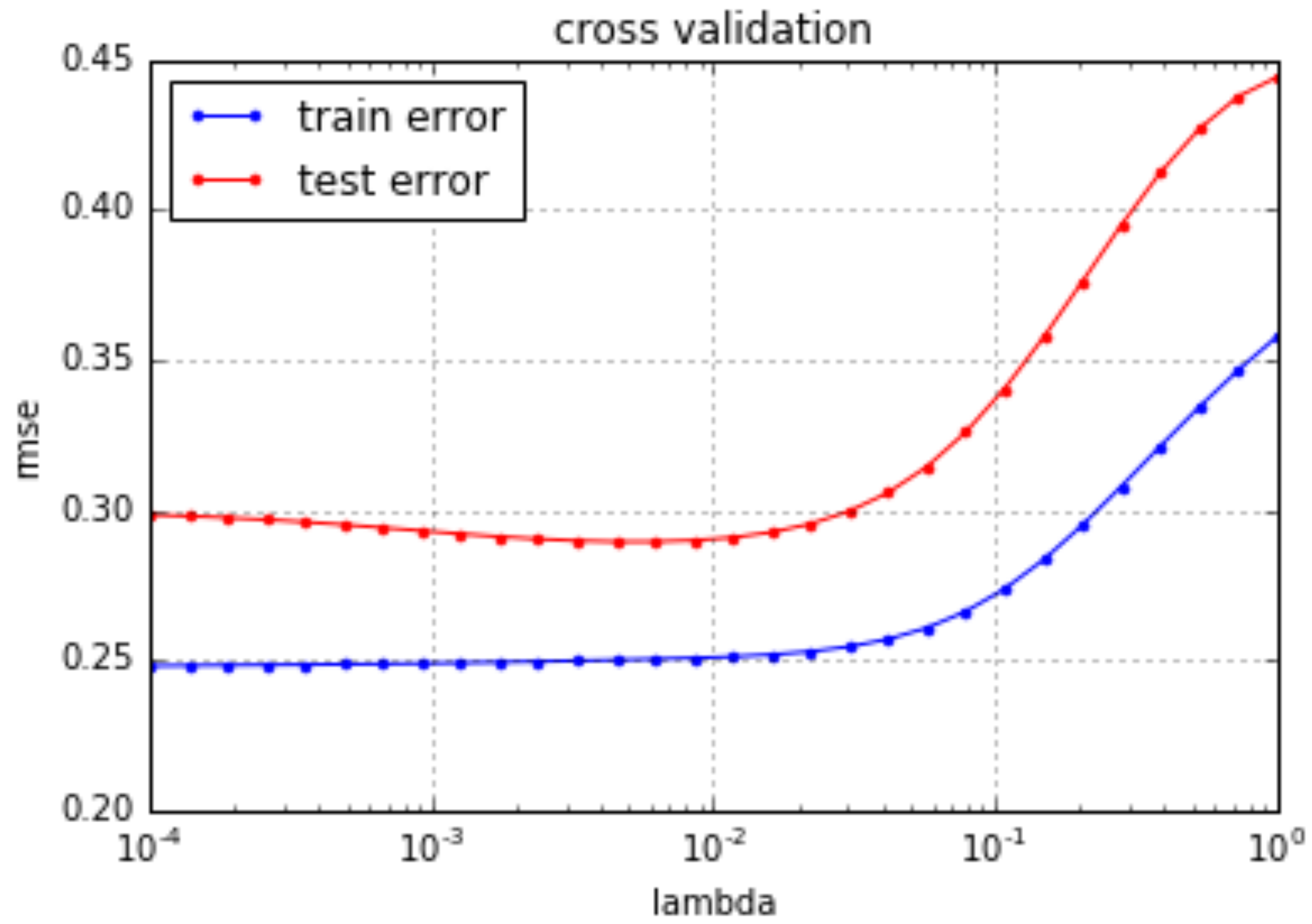
➡Split data into train and test sets (learn with the train and test on the test)

Motivation: Hyperparameters search (which often control the complexity)

But we haven’t investigated the role of the complexity of the class

Model selection curves

model complexity



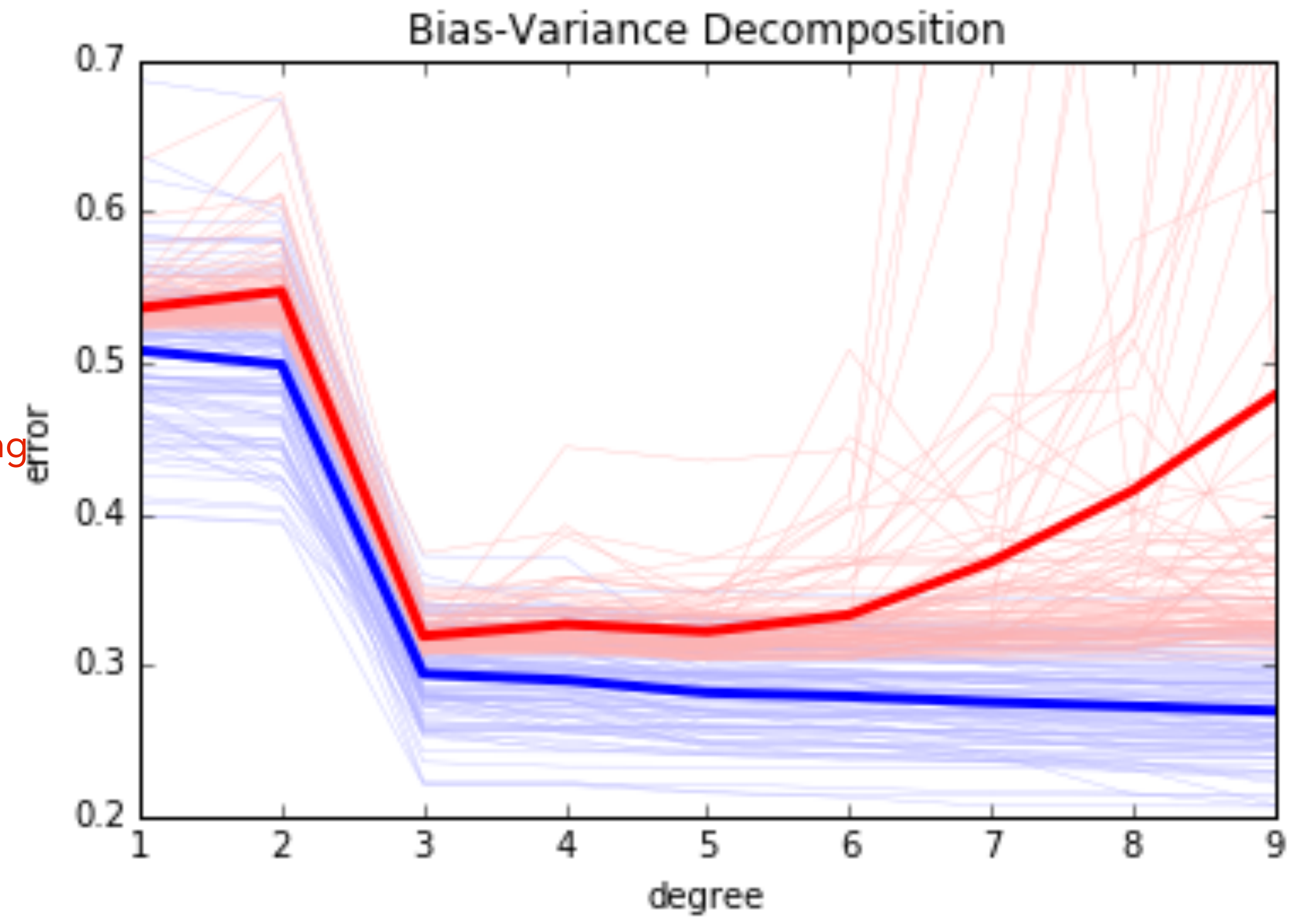
overfitting

underfitting

Ridge regression

reg. strenght

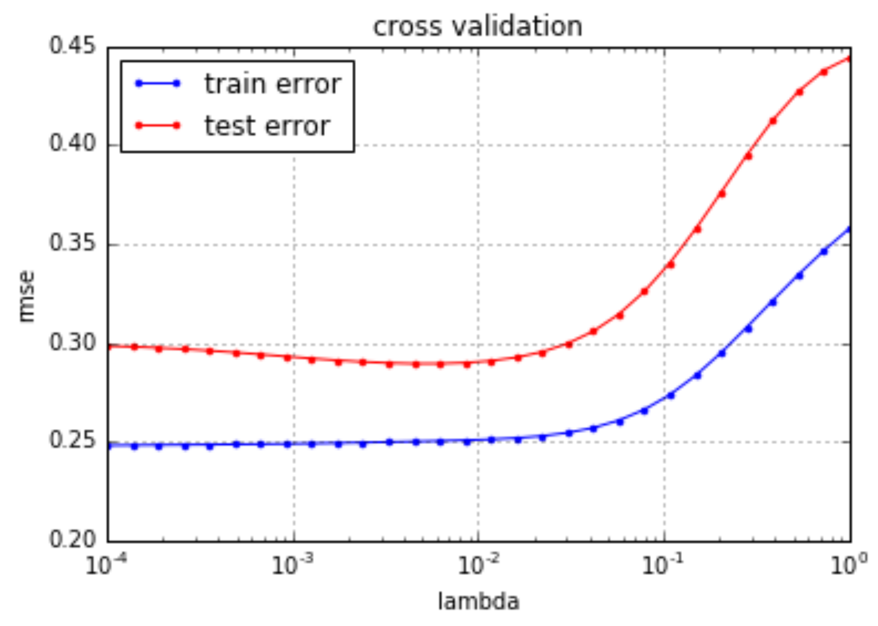
model complexity



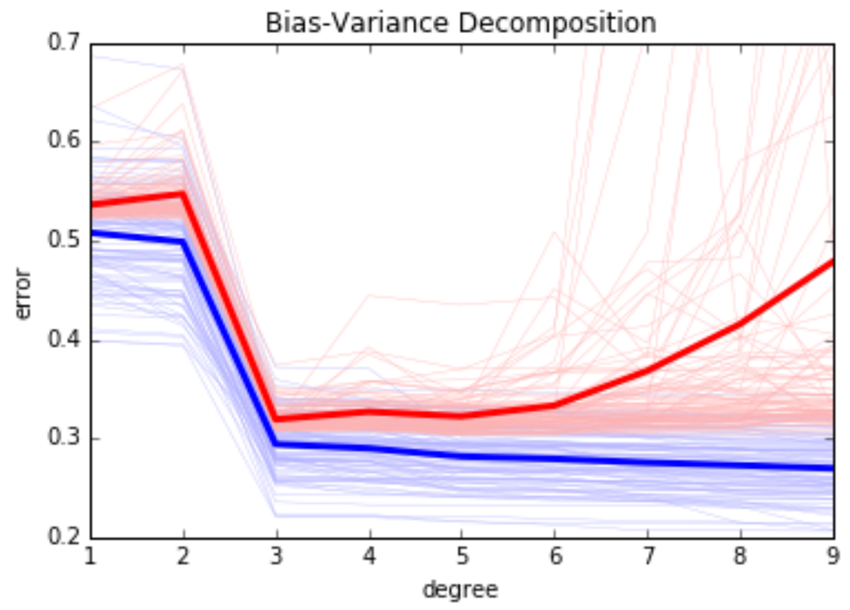
overfitting

Degree in case of a polynomial feature expansion

Today



Ridge regression



Polynomial feature expansion

How does the risk behave as a function of the complexity of the model class?

➡ ***Bias-Variance tradeoff***

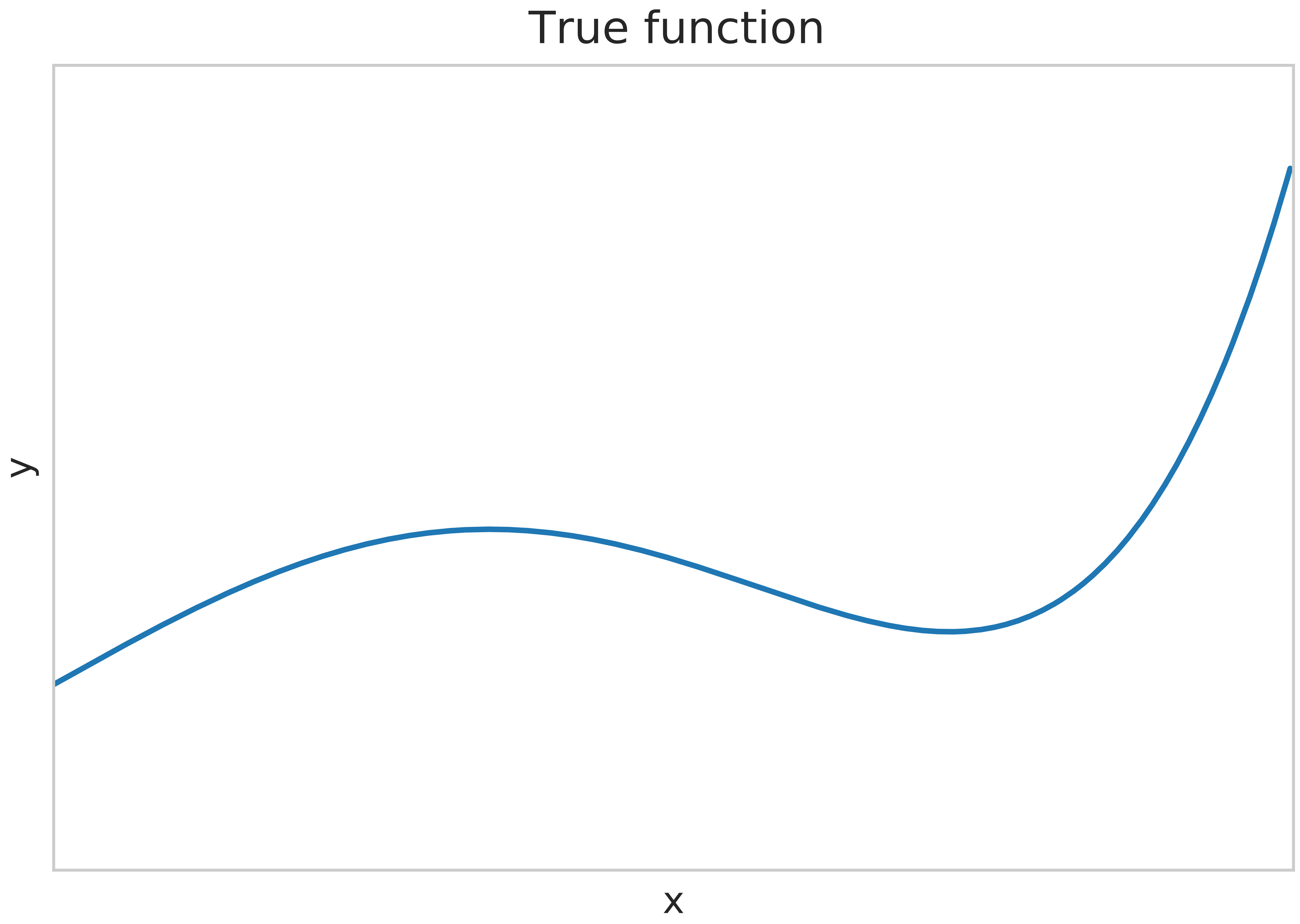
It will help us to decide how complex and rich we should make our model

Before: quantitative

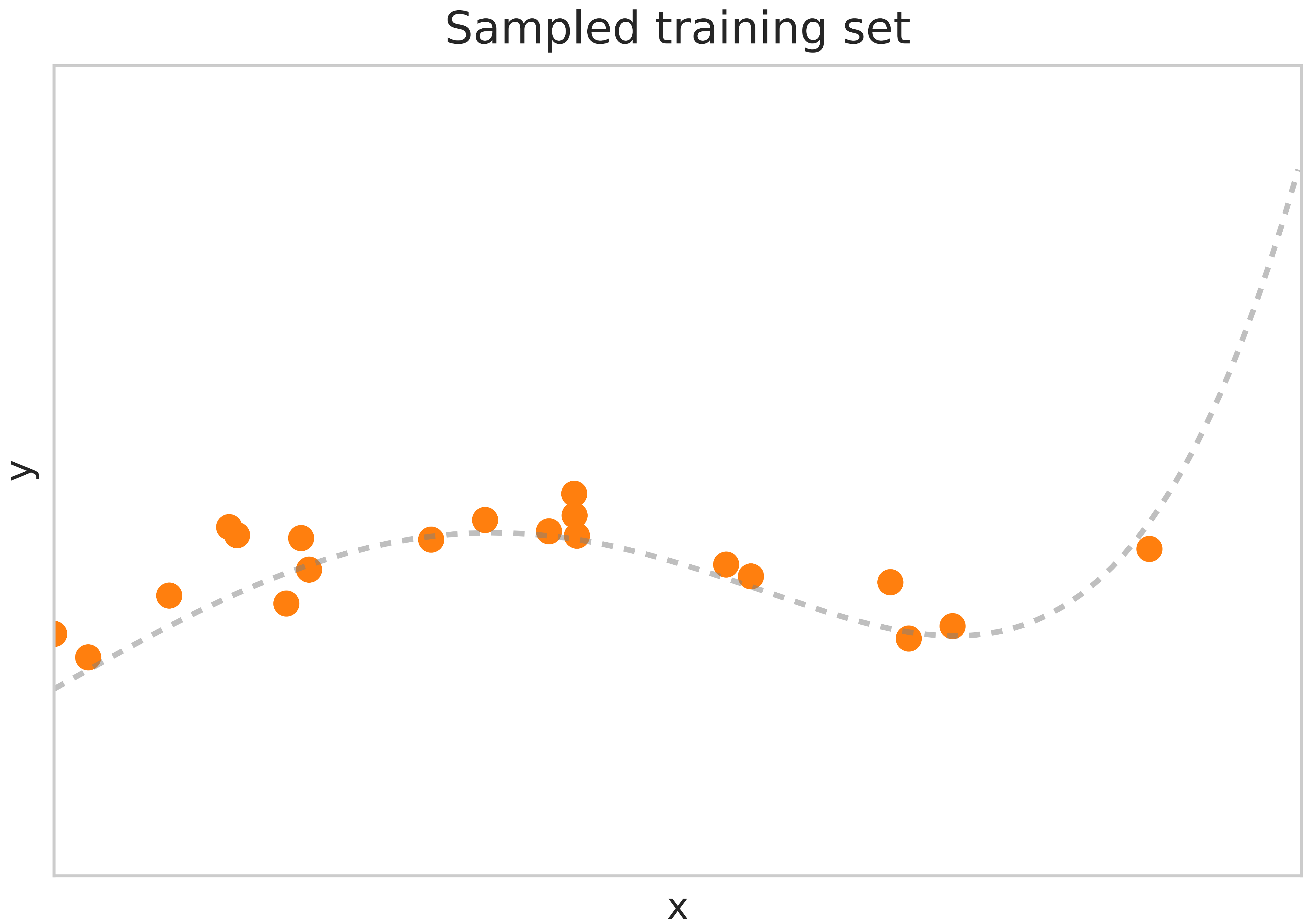
Now: ***qualitative***

how powerful should a model be

A small experiment: 1D-regression



A small experiment: 1D-regression



Linear regression using polynomial feature expansion $(x, x^2, x^3, \cdots, x^d)$

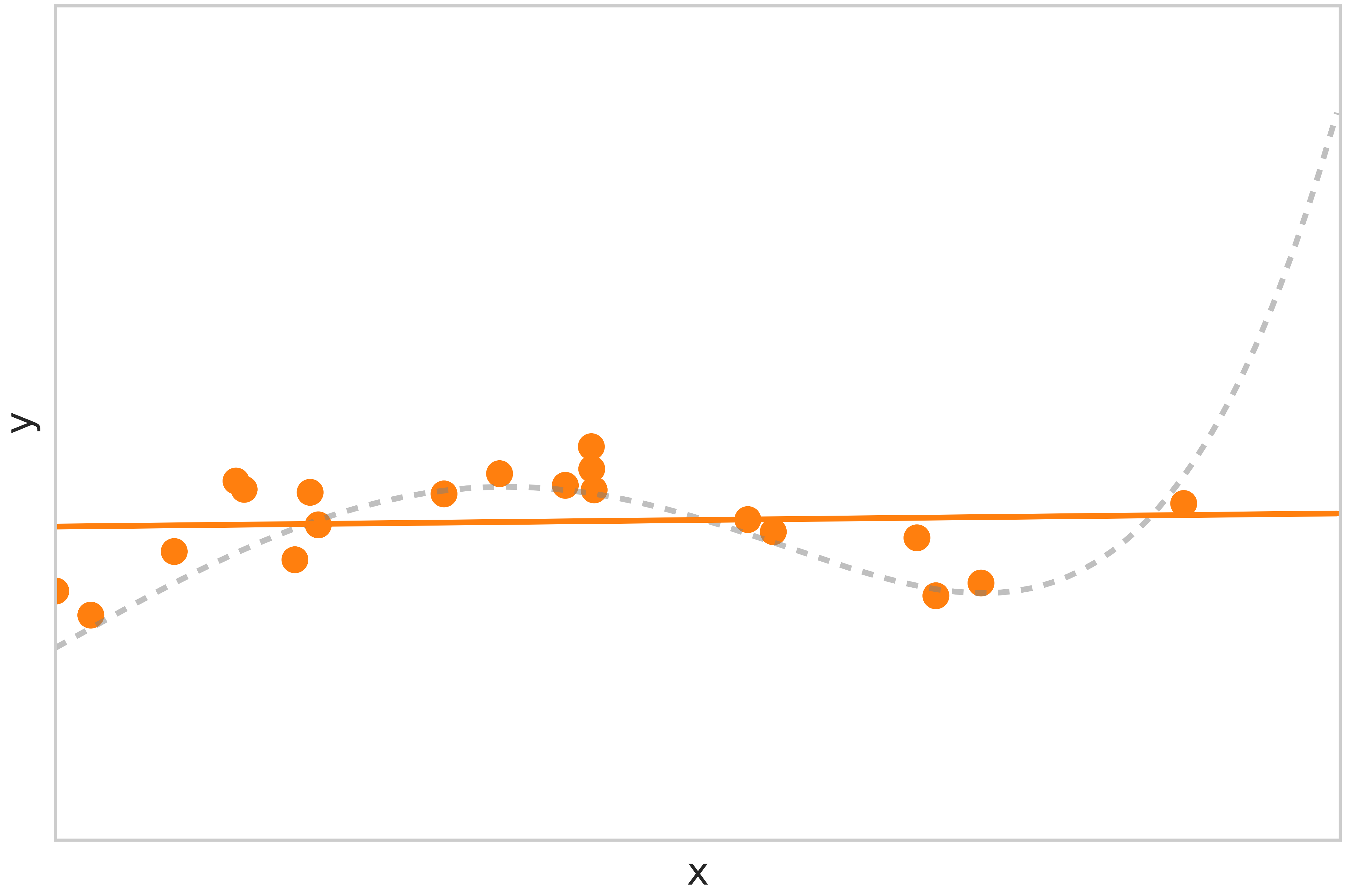
The maximum degree d measures the complexity of the class

➡ How far should you go?

Simple model: bad fit

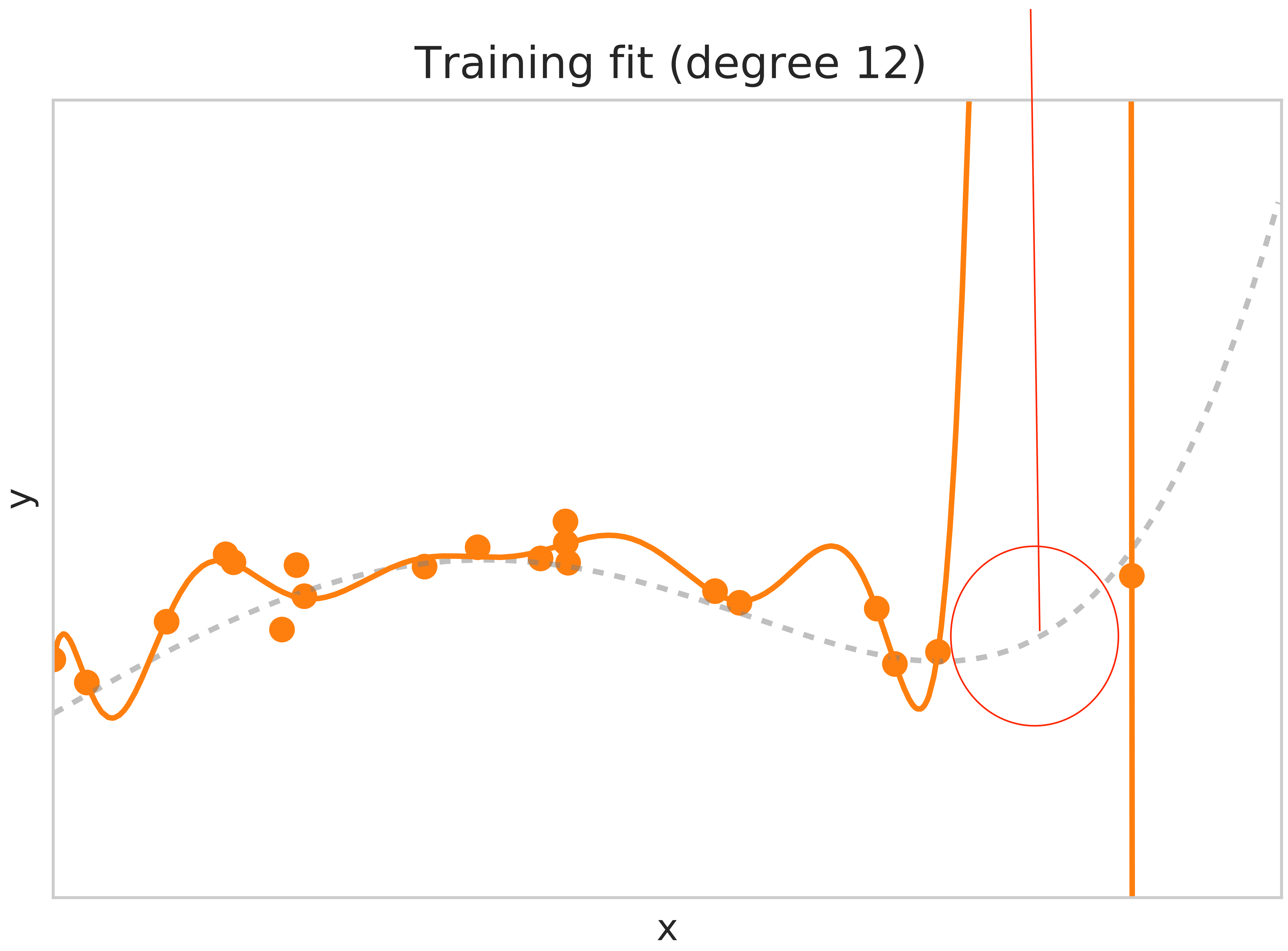
first week where model is a constant prediction

Training fit (degree 1)



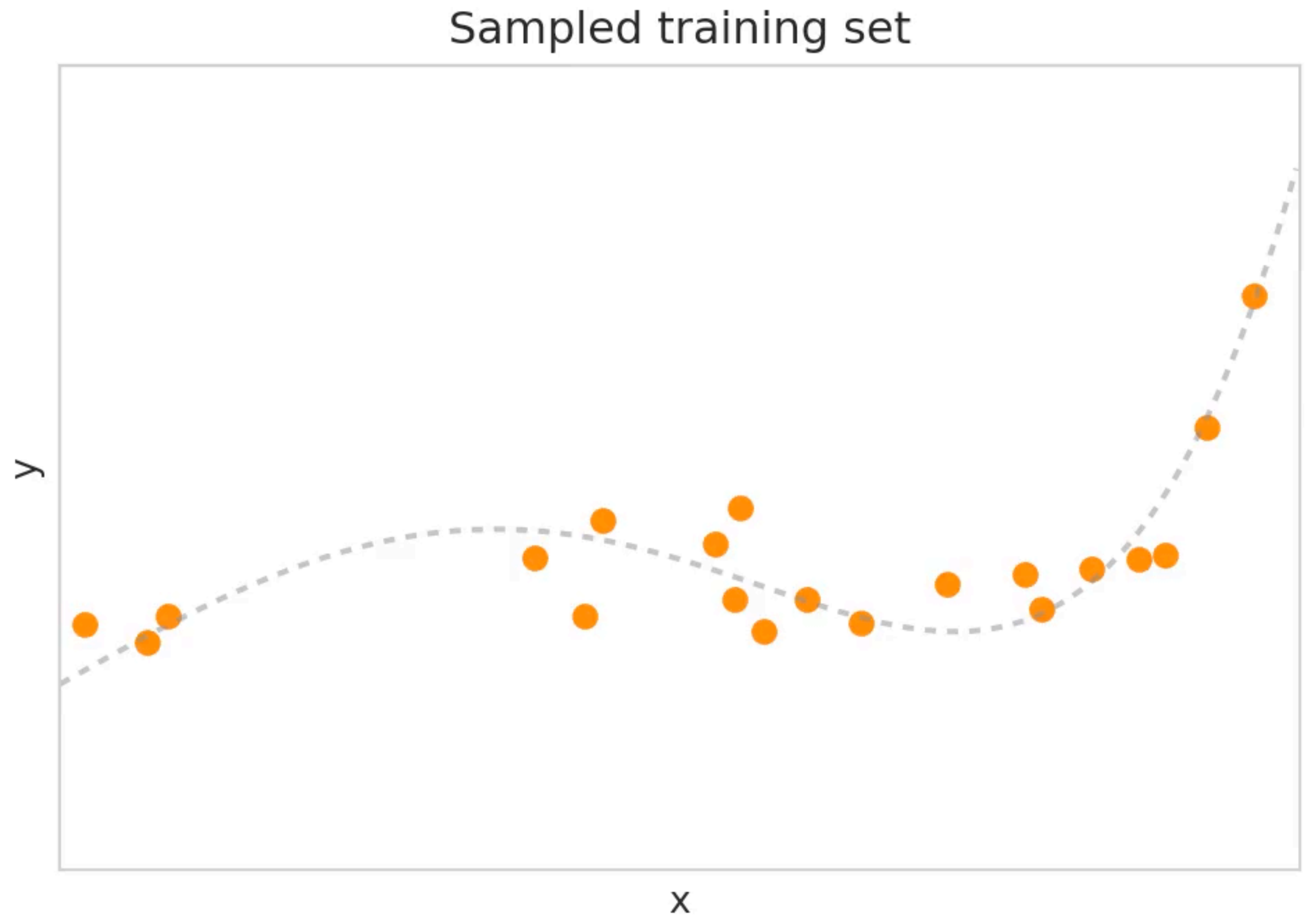
No linear function would be a good predictor. The model class is not rich enough

Complex model: good fit?



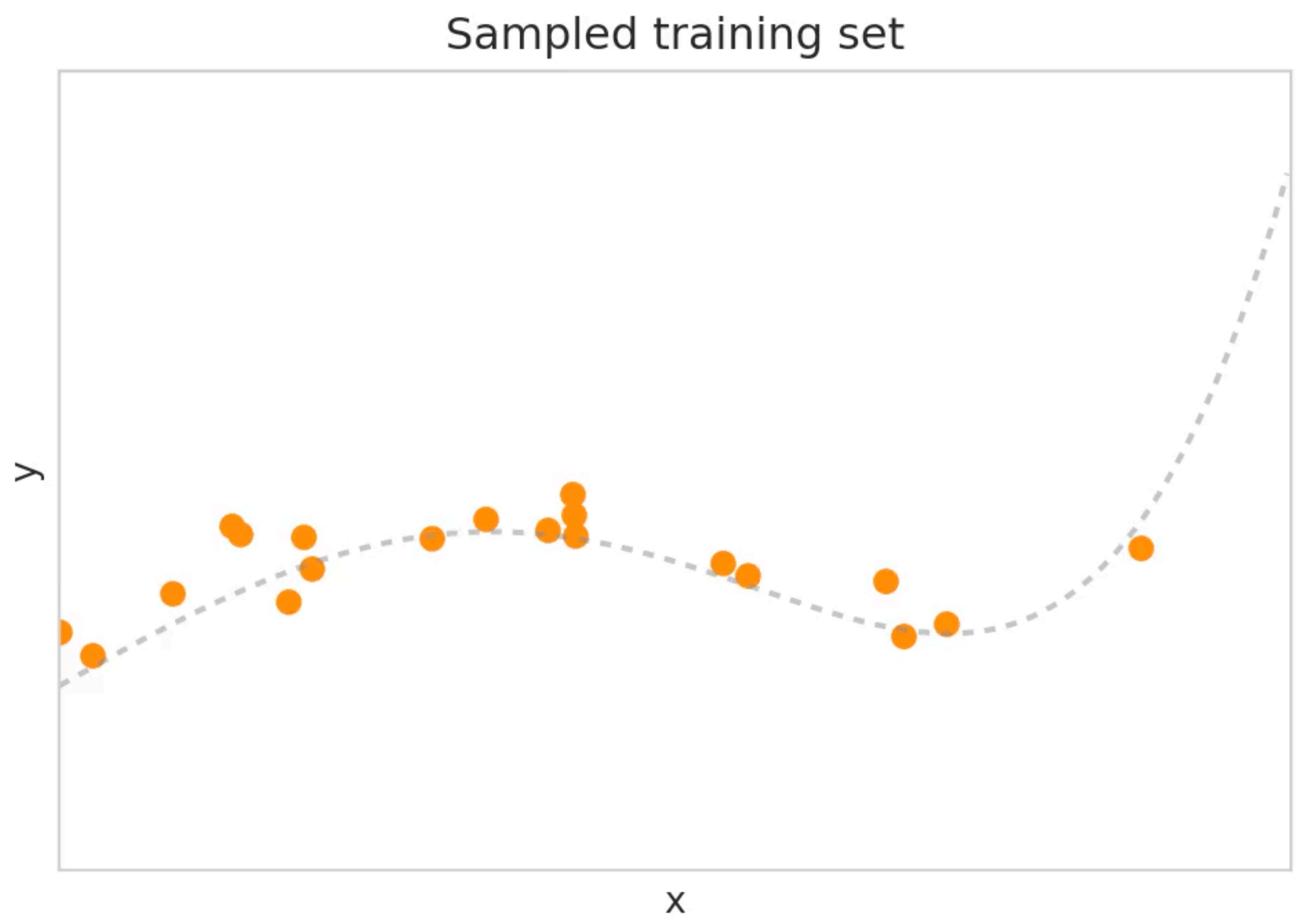
High degree polynomial will be a good fit. But?

But there is randomness in the data



We have observed one particular S_{train} but we could have observed several others!

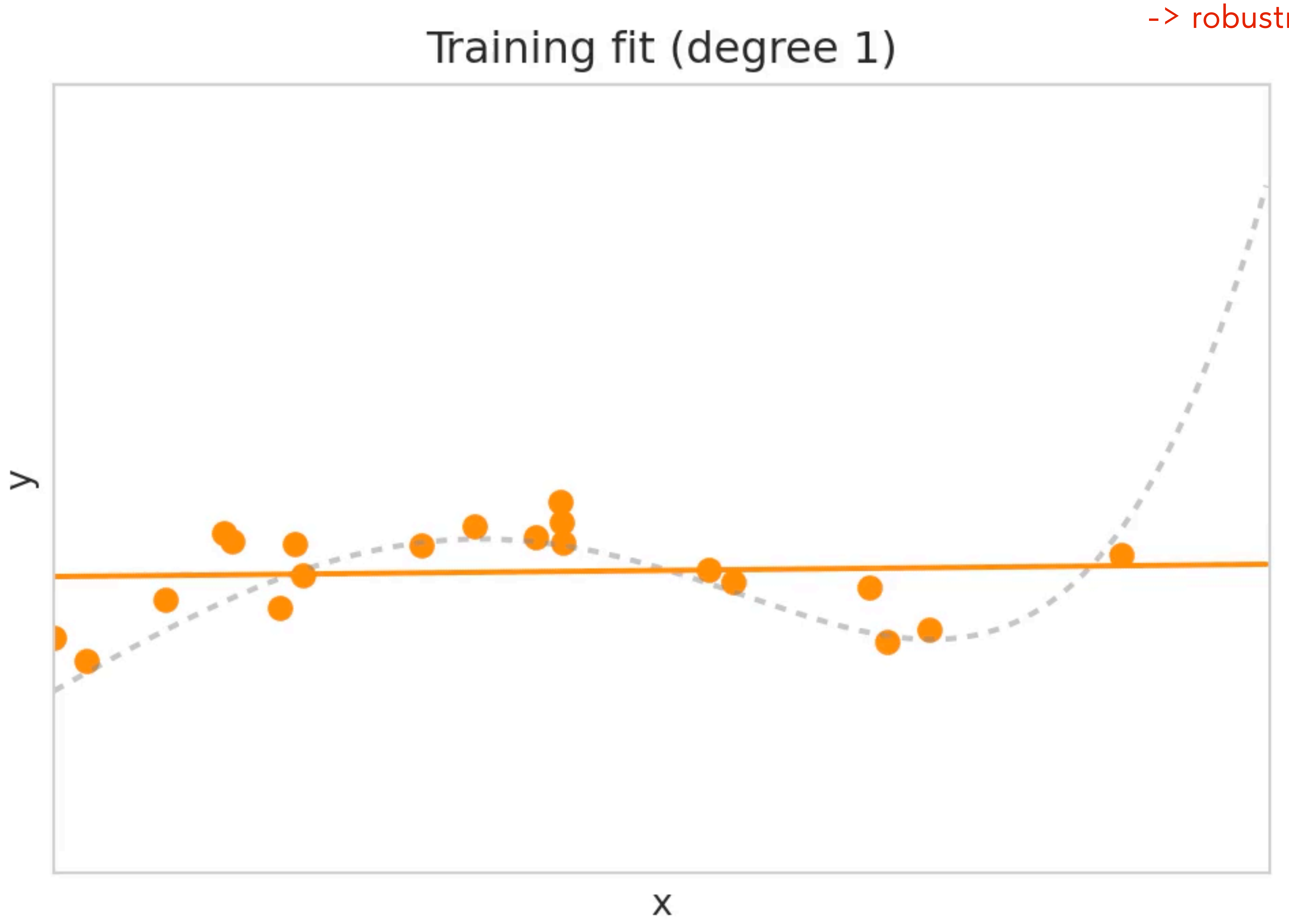
But there is randomness in the data

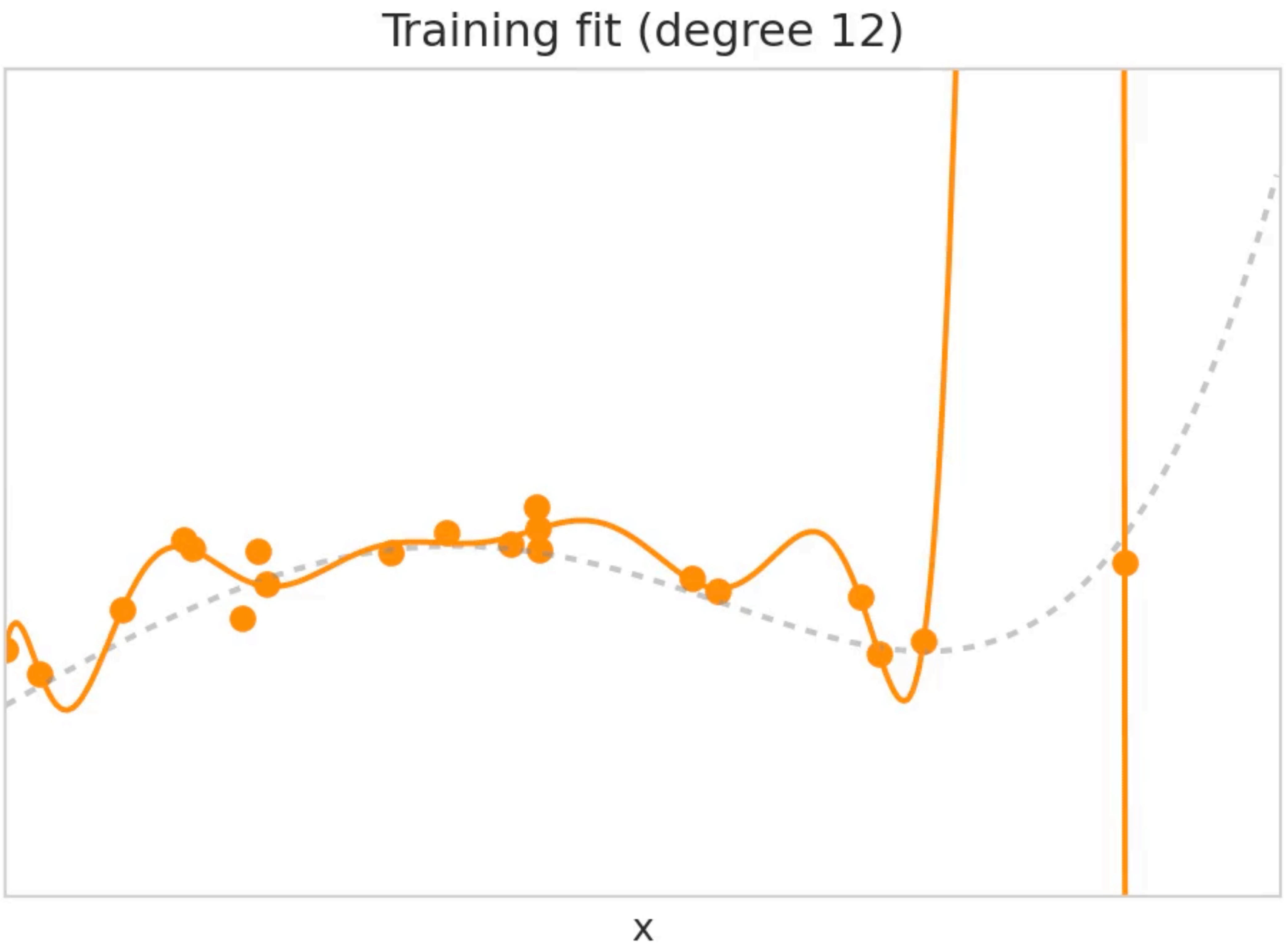


Even if we keep the same $(x_1, \cdots, x_n)$, we have variability in the observed $(y_1, \cdots, y_n)$

Thus there is randomness in the predictions

problem of sensitivity
-> robustness





Moving a single observation will cause only a small shift in the position of the line

Underfitting

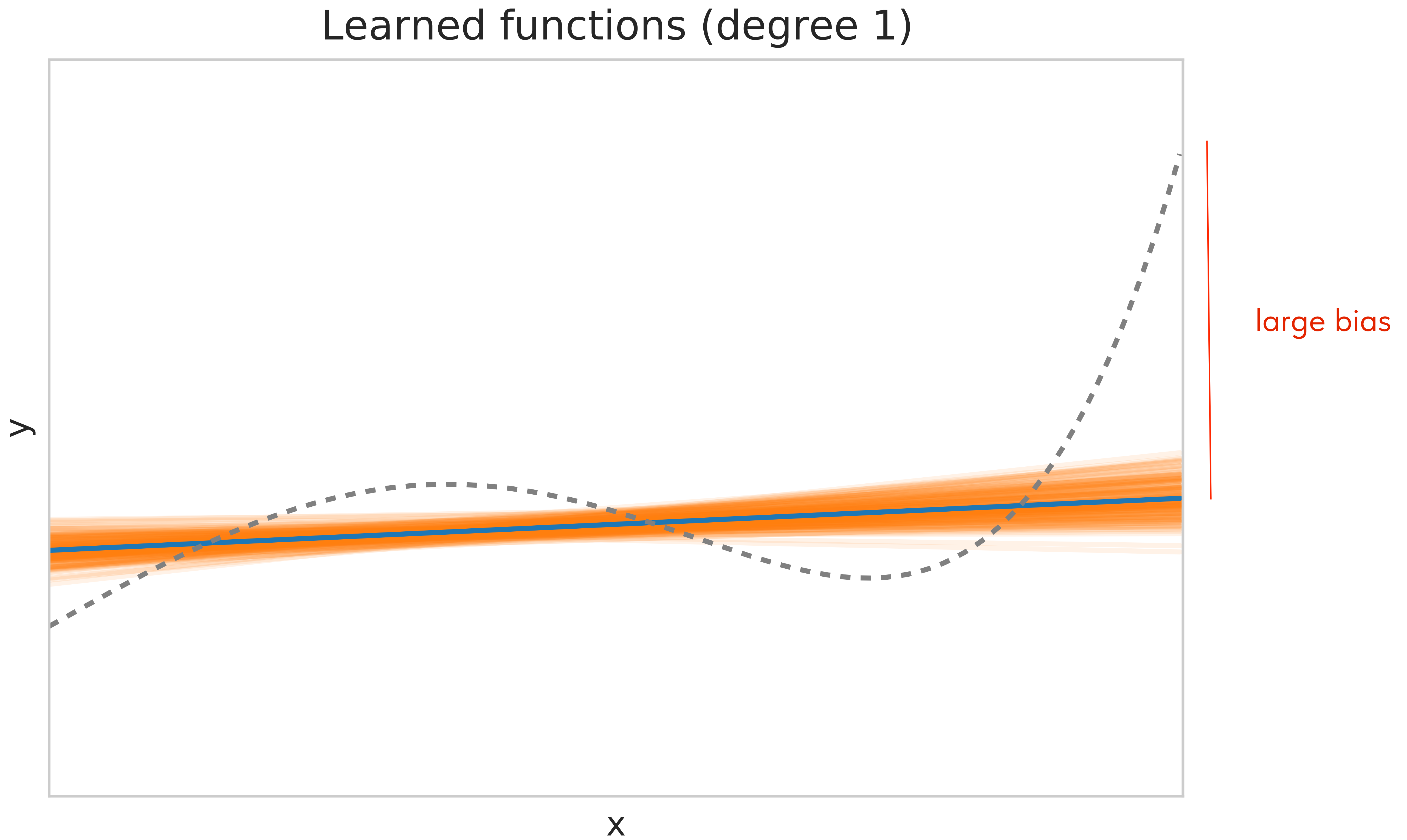
Changing one of the observations may change the prediction considerably

Overfitting

Simple models are less sensitive

Simple models have large bias but low variance

underfitting

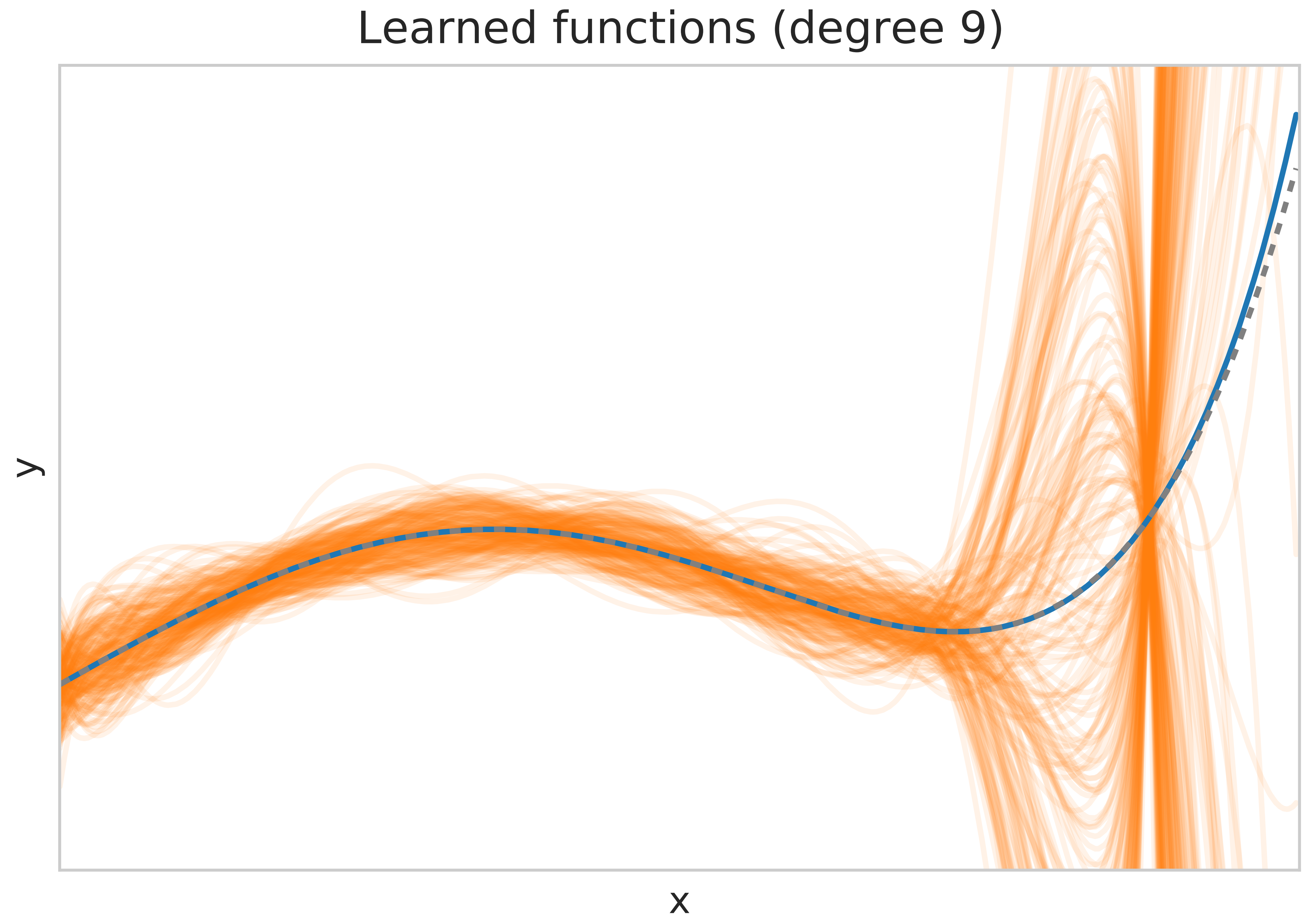


The average of the predictions f_S does not fit well the data: **large bias**

The variance of the predictions f_S as a function of S is small: **small variance**

Complex models have low bias but high variance

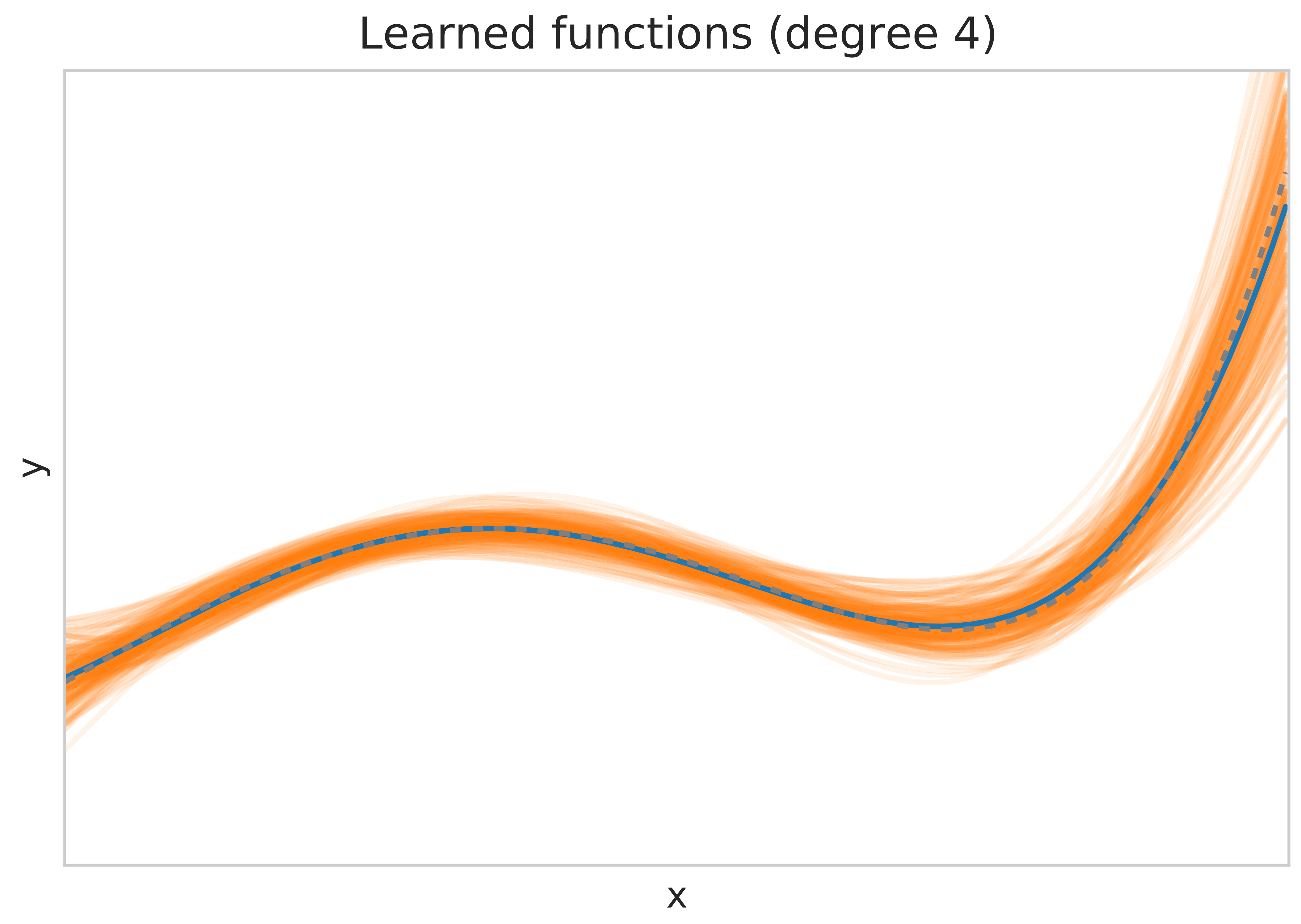
overfitting



The average of the predictions f_S fits well the data: **small bias**

The variance of the predictions f_S as a function of S is large: **large variance**

We need to balance bias & variance correctly



Data model: output perturbed by some noise

Output

True model perturbed by some noise

Joint distribution $(x, y) \sim \mathscr{D}$

$$y = \overbrace{f(x)}^{\text{true data distribution}} + \varepsilon$$

True model

f arbitrary and unknown function

The model is **not realizable**, i.e., f is not in our model class

Input

$x \sim \mathscr{D}_x$

(fixed but unknown)

Noise

$\varepsilon \sim \mathscr{D}_\varepsilon$ i.i.d., independent of x

$\mathbb{E}[\varepsilon] = 0$

We consider the square loss and will provide a decomposition of the true error

Error Decomposition

$S = \{(x_n, y_n)\}_{n=1}^{N} \sim \mathcal{D}$ i.i.d. $\mathcal{A}$ $f_S = \mathcal{A}(S)$

We are interested in how the ***expected error*** of f_S:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[(y - f_S(x))^2]$$

behaves as a ***function of the train set*** S and model class complexity

Error Decomposition

$S = \{(x_n, y_n)\}_{n=1}^{N} \sim \mathscr{D}$ i.i.d. $\quad\quad \mathscr{A} \quad\quad f_S = \mathscr{A}(S)$

The decomposition will hold true at ***every single point*** x. Therefore, to simplify, we consider the expected error of f_S for a fixed element x_0:

noise

we chose to use square error

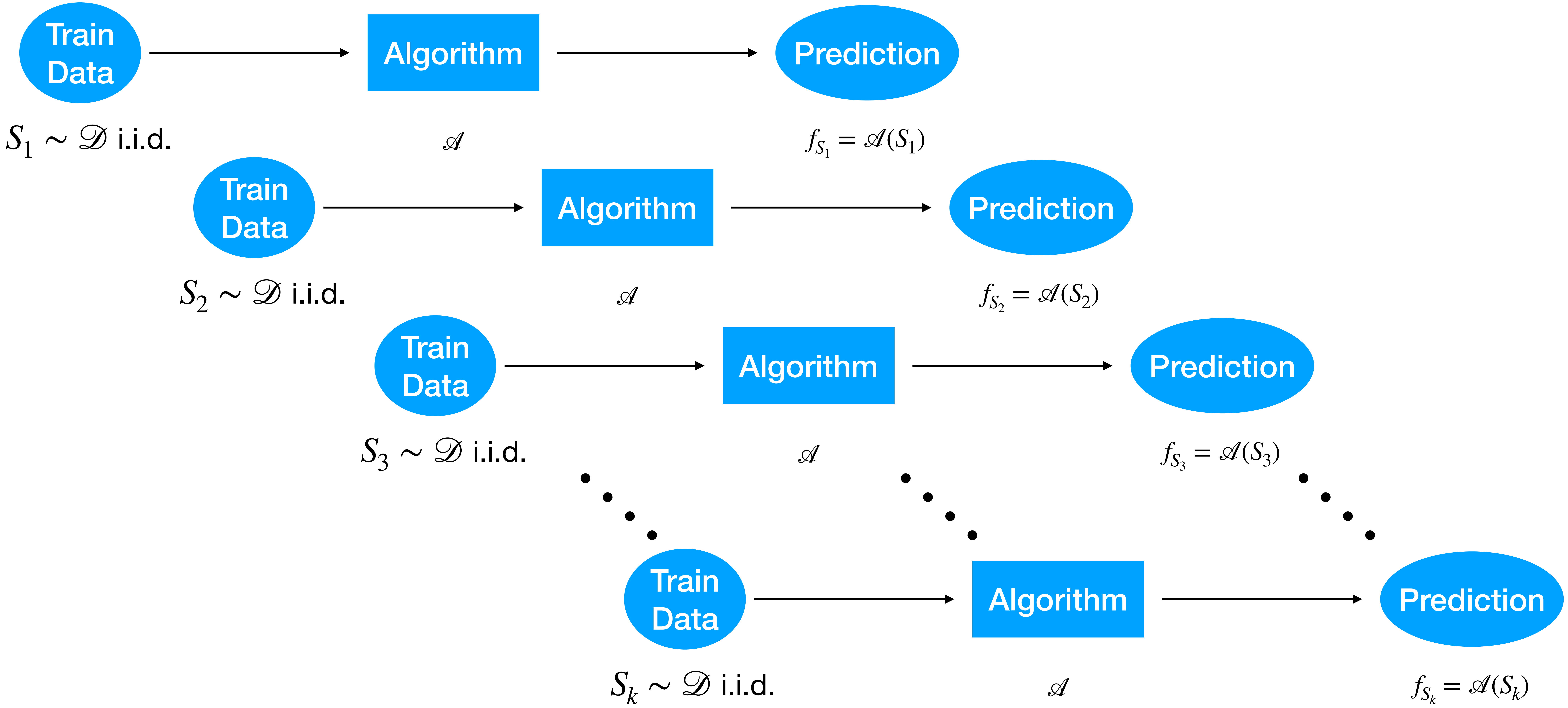
$$L(f_S) = \mathbb{E}_{\varepsilon \sim \mathscr{D}_\varepsilon}[(f(x_0) + \varepsilon - f_S(x_0))^2]$$

true y

prediction

This is a random variable. The randomness comes for the train set S

We run the experiment many times



We are interested in the ***average*** and the ***variance*** of the ***predictions*** $(f_{S_1}, \cdots, f_{S_k})$ over these multiple runs

A decomposition in three terms

We are interested in the expectation of the true risk over the training set S

$$\begin{aligned} \mathbb{E}_{S\sim\mathscr{D}}[L(f_S)] &= \mathbb{E}_{S\sim\mathscr{D}}\left[\mathbb{E}_{\varepsilon\sim\mathscr{D}_\varepsilon}[(f(x_0)+\varepsilon-f_S(x_0))^2]\right] \\ &= \mathbb{E}_{S\sim\mathscr{D},\varepsilon\sim\mathscr{D}_\varepsilon}[(f(x_0)+\varepsilon-f_S(x_0))^2] \end{aligned}$$

We will decompose this quantity in ***three non-negative terms*** and will interpret each of these terms

First we expand the square: why? bc if close term is 0

noise of test point x0

$$\mathbb{E}_{S\sim\mathscr{D},\,\varepsilon\sim\mathscr{D}_\varepsilon}[(f(x_0)+\varepsilon-f_S(x_0))^2] = \mathbb{E}_{\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon^2]$$

we can eliminate the second term

$$+2\mathbb{E}_{S\sim\mathscr{D},\,\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon(f(x_0)-f_S(x_0))]$$

$$+\mathbb{E}_{S\sim\mathscr{D}}[(f(x_0)-f_S(x_0))^2]$$

epsilon is independent so product of expectetion

Using that $\mathbb{E}_{\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon]=0$ and $\varepsilon \perp\!\!\!\perp S$:

- $\mathbb{E}_{\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon^2] = \mathrm{Var}_{\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon]$
- $\mathbb{E}_{S\sim\mathscr{D},\,\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon(f(x_0)-f_S(x_0))] = \mathbb{E}_{\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon]\times\mathbb{E}_{S\sim\mathscr{D}}[f(x_0)-f_S(x_0)]=0$

Therefore

$$\boxed{\mathbb{E}_{S\sim\mathscr{D},\,\varepsilon\sim\mathscr{D}_\varepsilon}[(f(x_0)+\varepsilon-f_S(x_0))^2] = \underbrace{\mathrm{Var}_{\varepsilon\sim\mathscr{D}_\varepsilon}[\varepsilon]}_{\text{variance of the noise}} + \underbrace{\mathbb{E}_{S\sim\mathscr{D}}[(f(x_0)-f_S(x_0))^2]}_{\text{how much does it fluctuate}}}$$

doesn't depend on S

Trick: we add and subtract the constant term $\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]$, where S' is a second training set independent from S

WE SHOULD BE ABLE TO DO IT......

$$
\begin{aligned}
\mathbb{E}_{S\sim\mathscr{D}}[(f(x_0)-f_S(x_0))^2] &= \mathbb{E}_{S\sim\mathscr{D}}\Big[(\overbrace{f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]}^{A}+\overbrace{\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-f_S(x_0)}^{B})^2\Big] \\
&= \mathbb{E}_{S\sim\mathscr{D}}\Big[(f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)])^2+(\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-f_S(x_0))^2 \\
&\quad +2(f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)])(\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-f_S(x_0))\Big]
\end{aligned}
$$

Cross-term:

A B

$$
\begin{aligned}
&\mathbb{E}_{S\sim\mathscr{D}}\Big[\big(f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]\big)\cdot\big(\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-f_S(x_0)\big)\Big] \\
&= \big(f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]\big)\cdot\mathbb{E}_{S\sim\mathscr{D}}\big[(\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-f_S(x_0))\big] \\
&= \big(f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]\big)\cdot\big(\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-\mathbb{E}_{S\sim\mathscr{D}}[f_S(x_0)]\big) = 0.
\end{aligned}
$$

$$
\boxed{\mathbb{E}_{S\sim\mathscr{D}}[(f(x_0)-f_S(x_0))^2] = \underbrace{(f(x_0)-\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)])^2}_{\text{bias}}+\underbrace{\mathbb{E}_{S\sim\mathscr{D}}[(\mathbb{E}_{S'\sim\mathscr{D}}[f_{S'}(x_0)]-f_S(x_0))^2]}_{\text{variance}}}
$$

Bias-Variance Decomposition

We obtain the following decomposition into three positive terms:

$$\mathbb{E}_{S\sim\mathcal{D},\,\varepsilon\sim\mathcal{D}_\varepsilon}[(f(x_0)+\varepsilon-f_S(x_0))^2] \quad = \quad \mathrm{Var}_{\varepsilon\sim\mathcal{D}_\varepsilon}[\varepsilon] \longleftarrow \textbf{Noise variance}$$

$$\textbf{Bias} \longrightarrow \quad + \quad (f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)])^2$$

$$\textbf{Variance} \longrightarrow \quad + \quad \mathbb{E}_{S\sim\mathcal{D}}\left[(f_S(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)])^2\right]$$

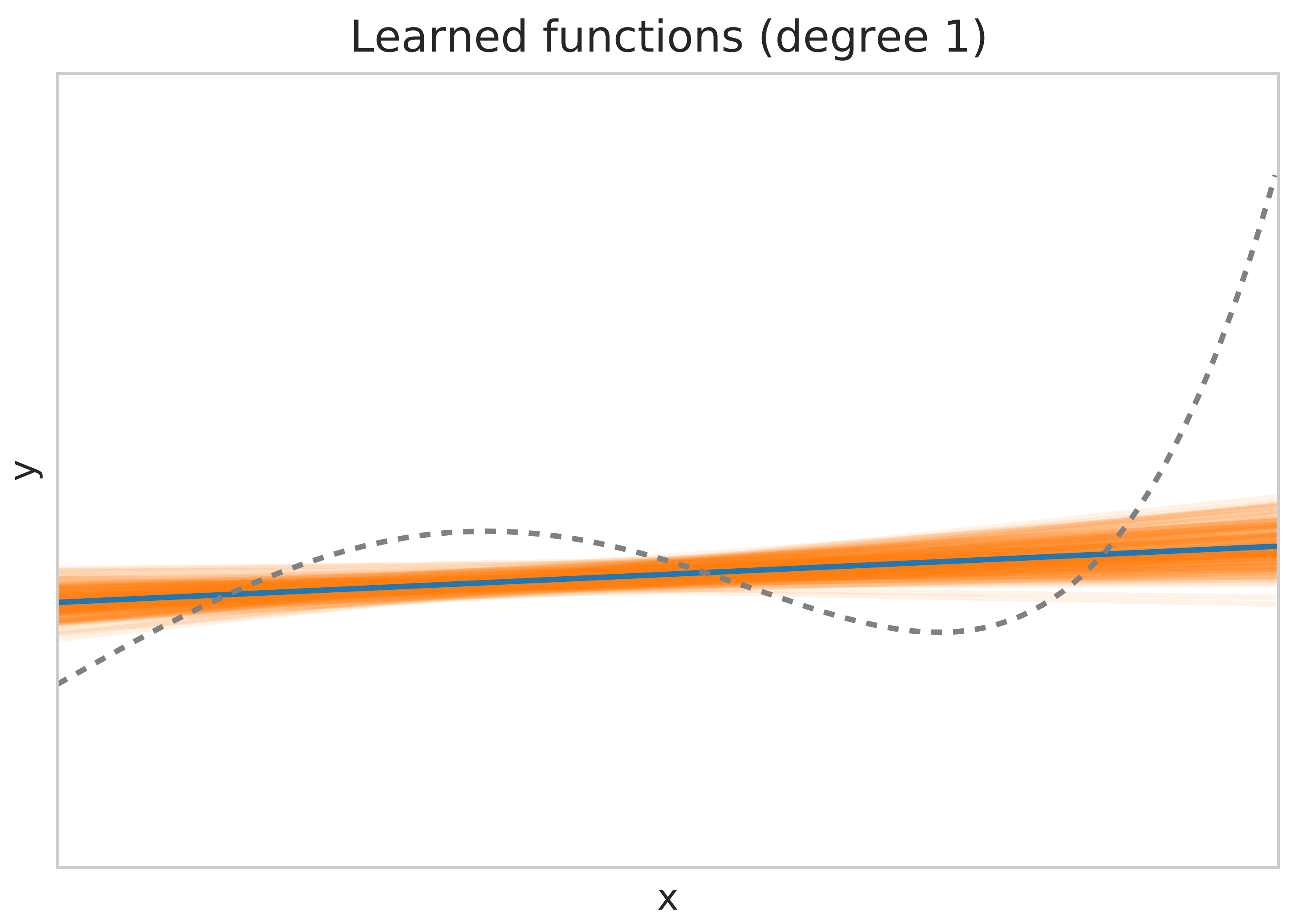
each of which always provides a lower bound of the true error

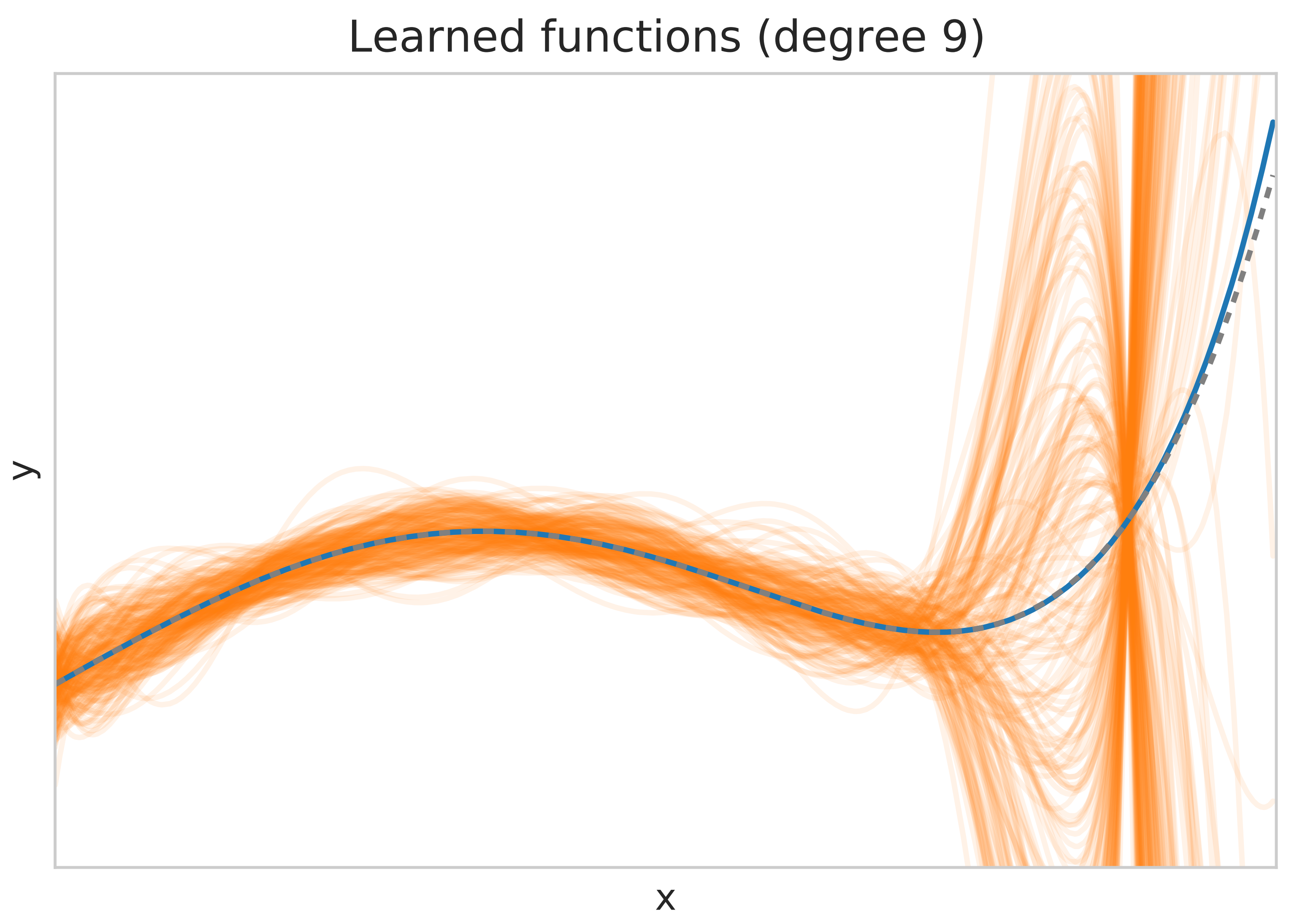
➡ To minimize the true error, we must choose a method that achieves **low bias and low variance** simultaneously

Noise: a strict lower bound on the achievable error

- It is not possible to go below the noise level
- Even if we know the true model f, we still suffer from the noise: $L(f) = \mathbb{E}[\varepsilon^2]$
- It is not possible to predict the noise from the data since they are independent

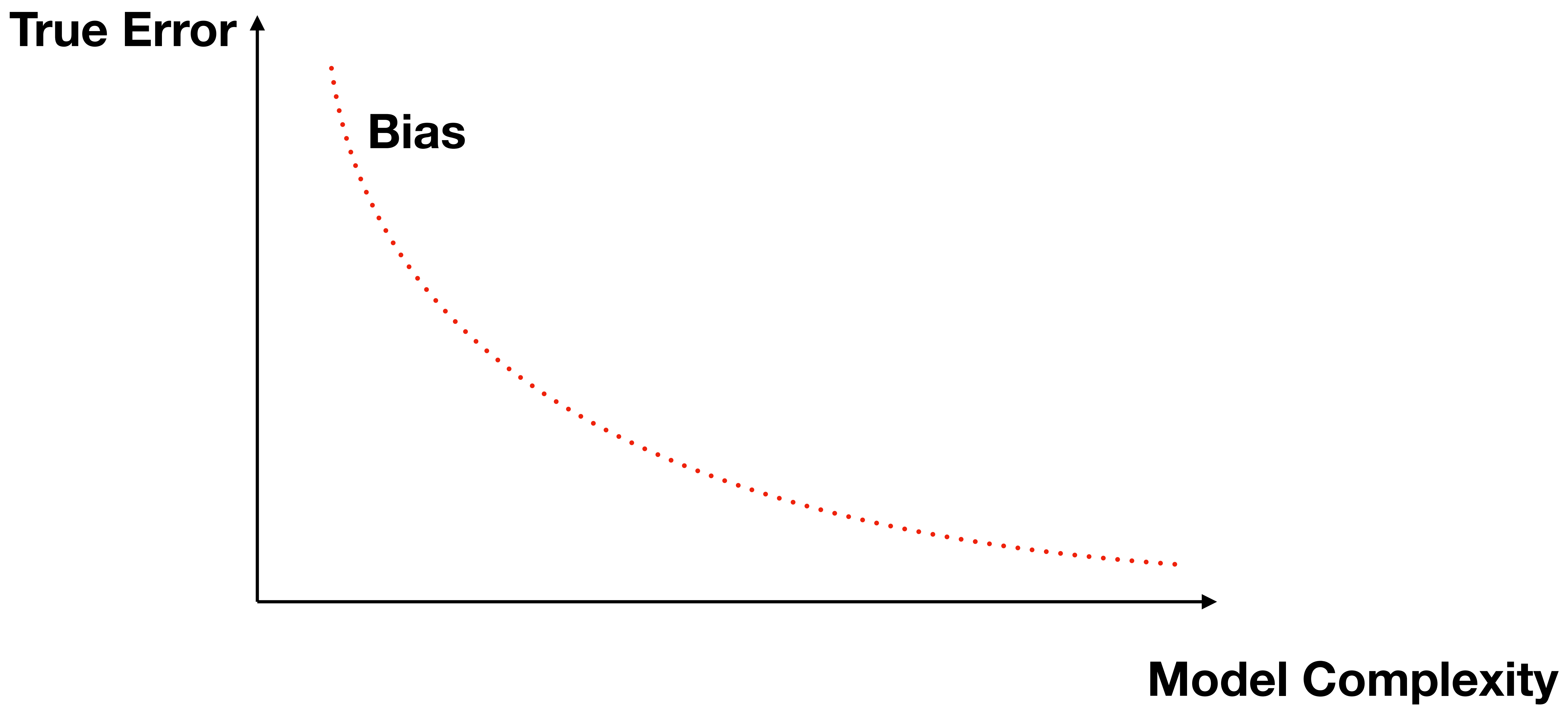
Bias: $(f(x_0) - \mathbb{E}_{S\sim\mathscr{D}}[f_S(x_0)])^2$





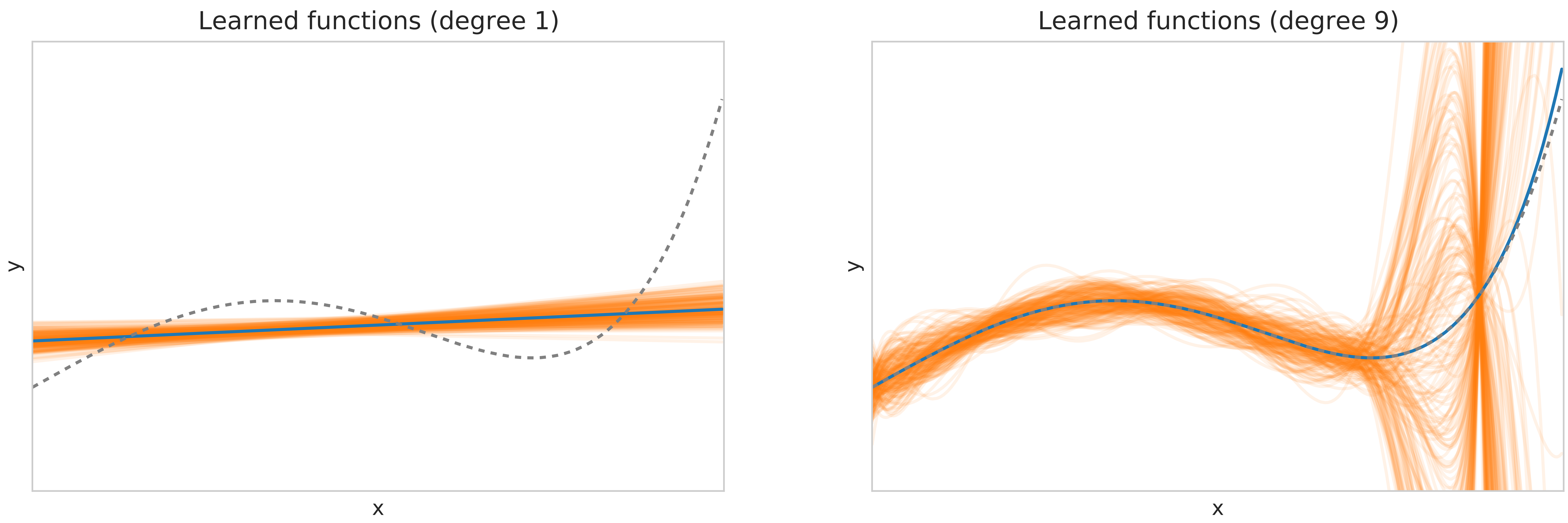
- Squared of the difference between the actual value $f(x_0)$ and the expected prediction
- It measures how far off in general the models' predictions are from the correct value
- If model **complexity** is **low**, **bias** is typically **high**
- If model **complexity** is **high**, **bias** Is typically **low**

Bias: $(f(x_0) - \mathbb{E}_{S\sim\mathscr{D}}[f_S(x_0)])^2$



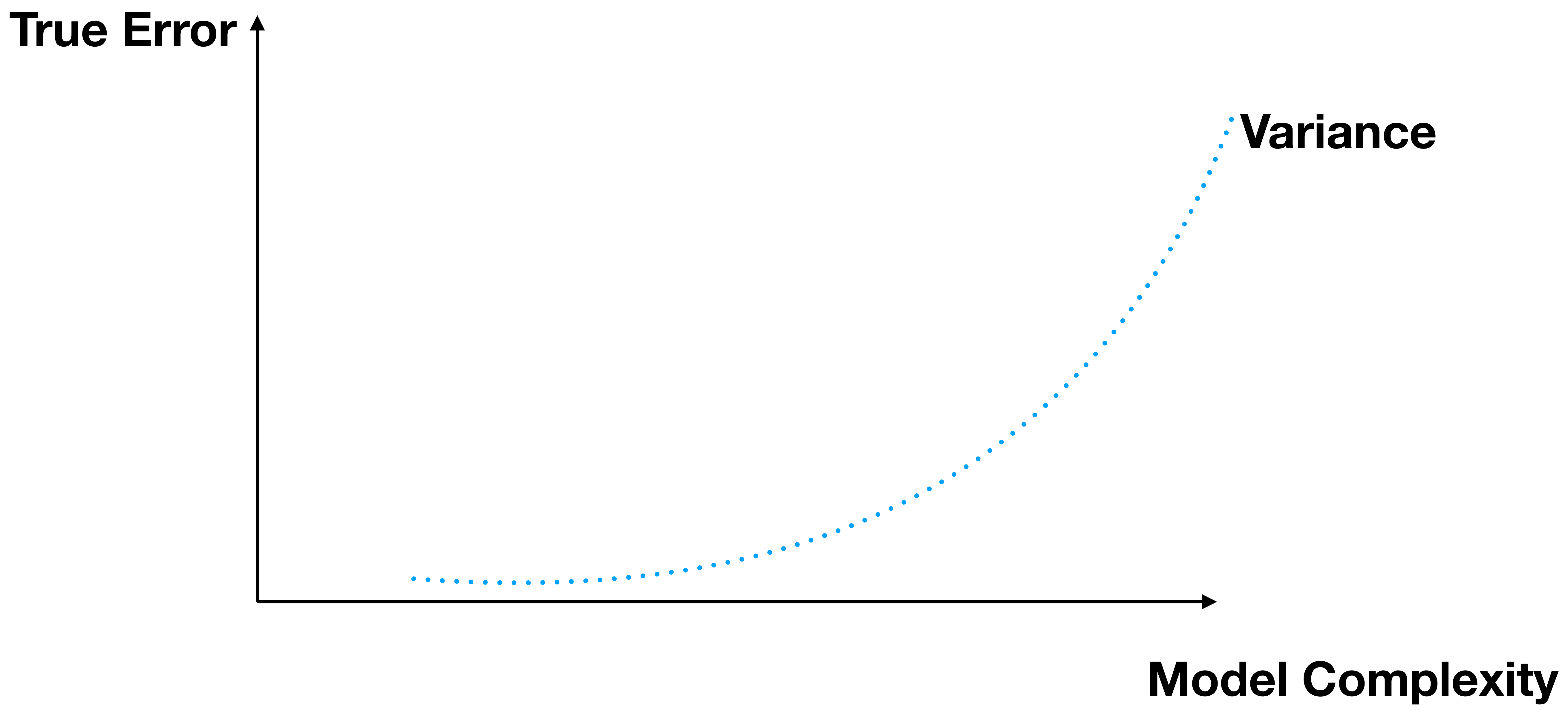
- Squared of the difference between the actual value $f(x_0)$ and the expected prediction
- It measures how far off in general the models' predictions are from the correct value
- If model **complexity** is **low**, **bias** is typically **high**
- If model **complexity** is **high**, **bias** is typically **low**

Variance: $\mathbb{E}_{S\sim\mathcal{D}}\left[\left(f_S(x_0) - \mathbb{E}_{S\sim\mathcal{D}}[f_S(x_0)]\right)^2\right]$



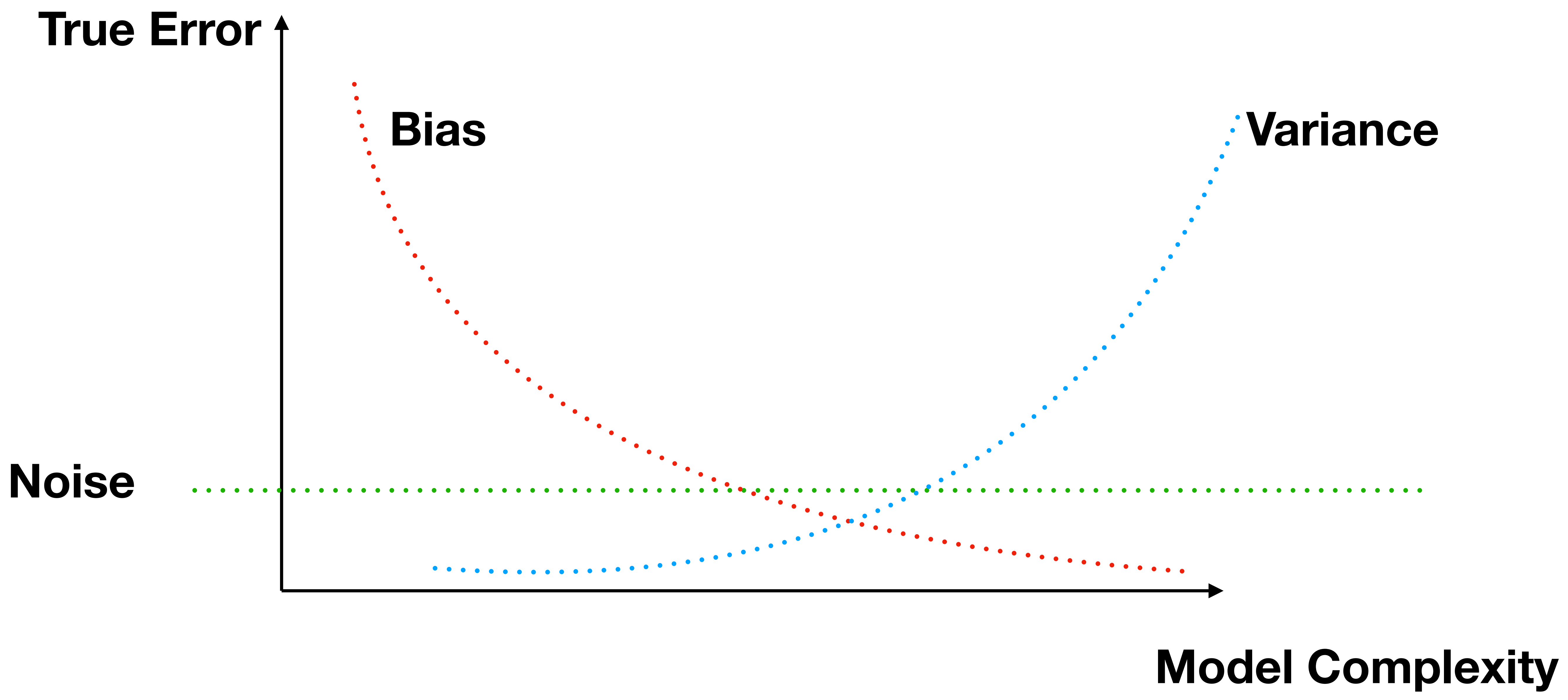
- Variance of the prediction function
- It measures the variability of predictions at a given point across different training set realizations
- If we consider complex models, small variations in the training set can lead to significant changes in the predictions

Variance: $\mathbb{E}_{S\sim\mathcal{D}}\left[(f_S(x_0) - \mathbb{E}_{S\sim\mathcal{D}}[f_S(x_0)])^2\right]$



- Variance of the prediction function
- It measures the variability of predictions at a given point across different training set realizations
- If we consider complex models, small variations in the training set can lead to significant changes in the predictions

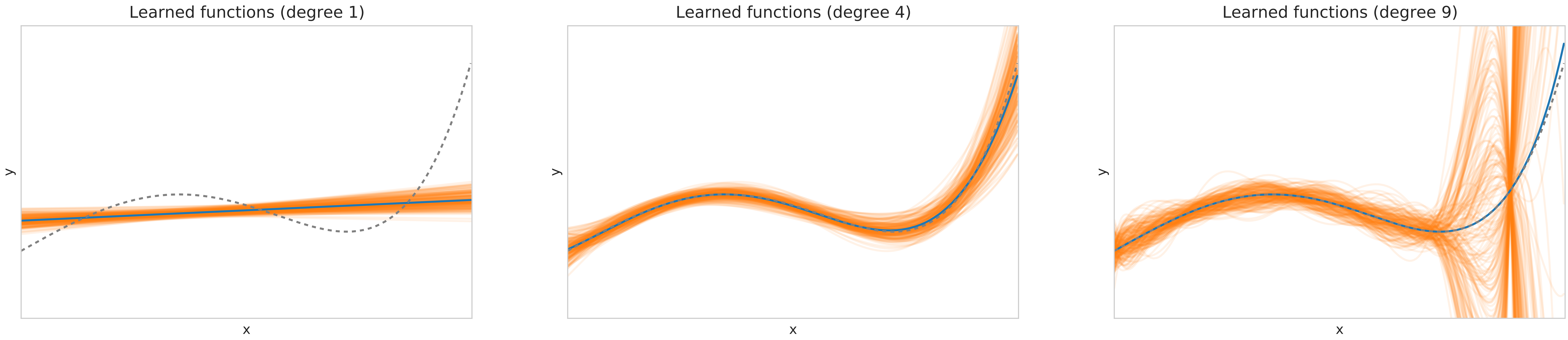
Bias Variance tradeoff and U-shape curve



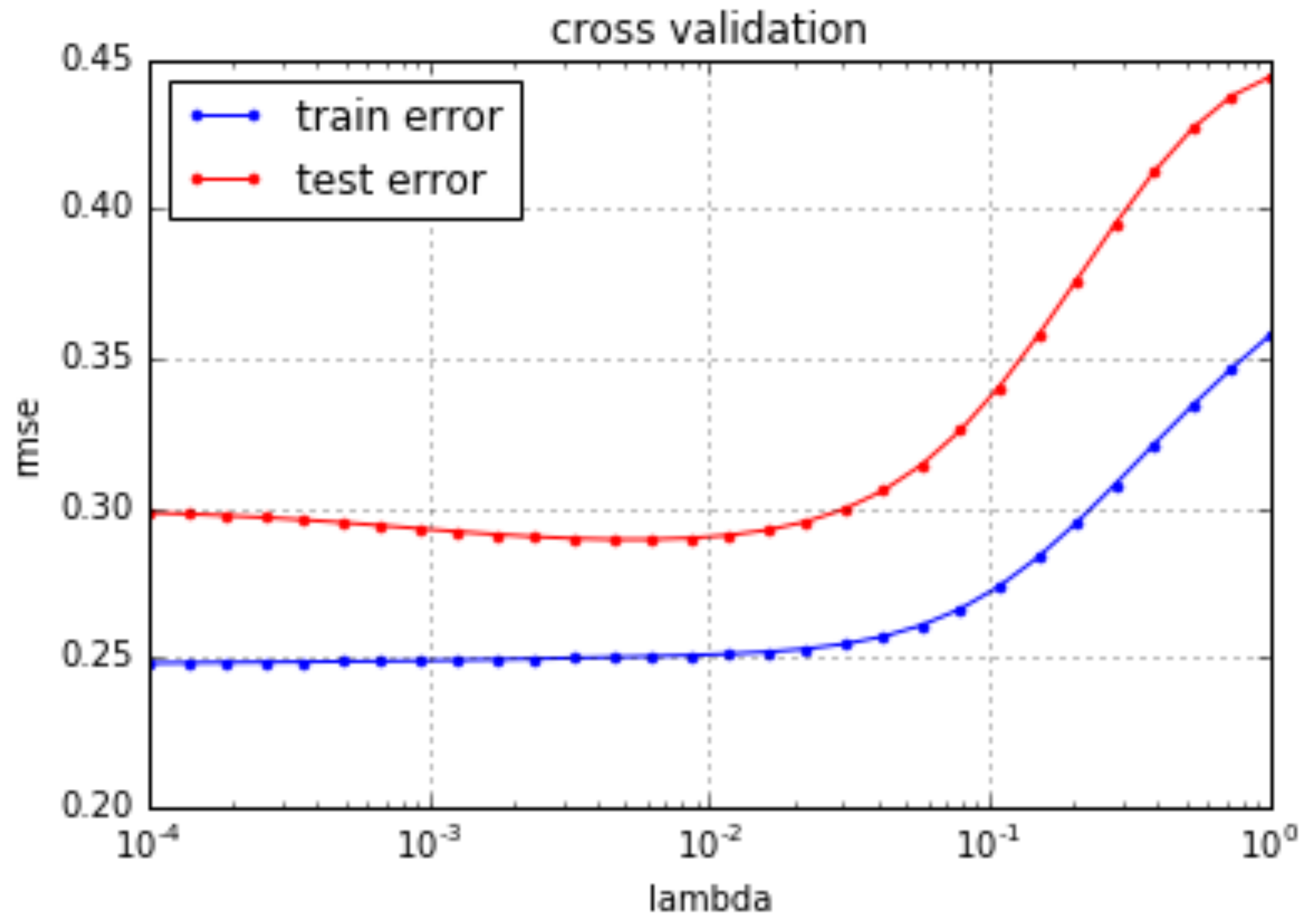
- If model complexity is too low, approximation will be poor (underfitting)
- If model complexity is too high, it may cause issues with variance (overfitting)

➡ This phenomenon is known as the bias-variance tradeoff

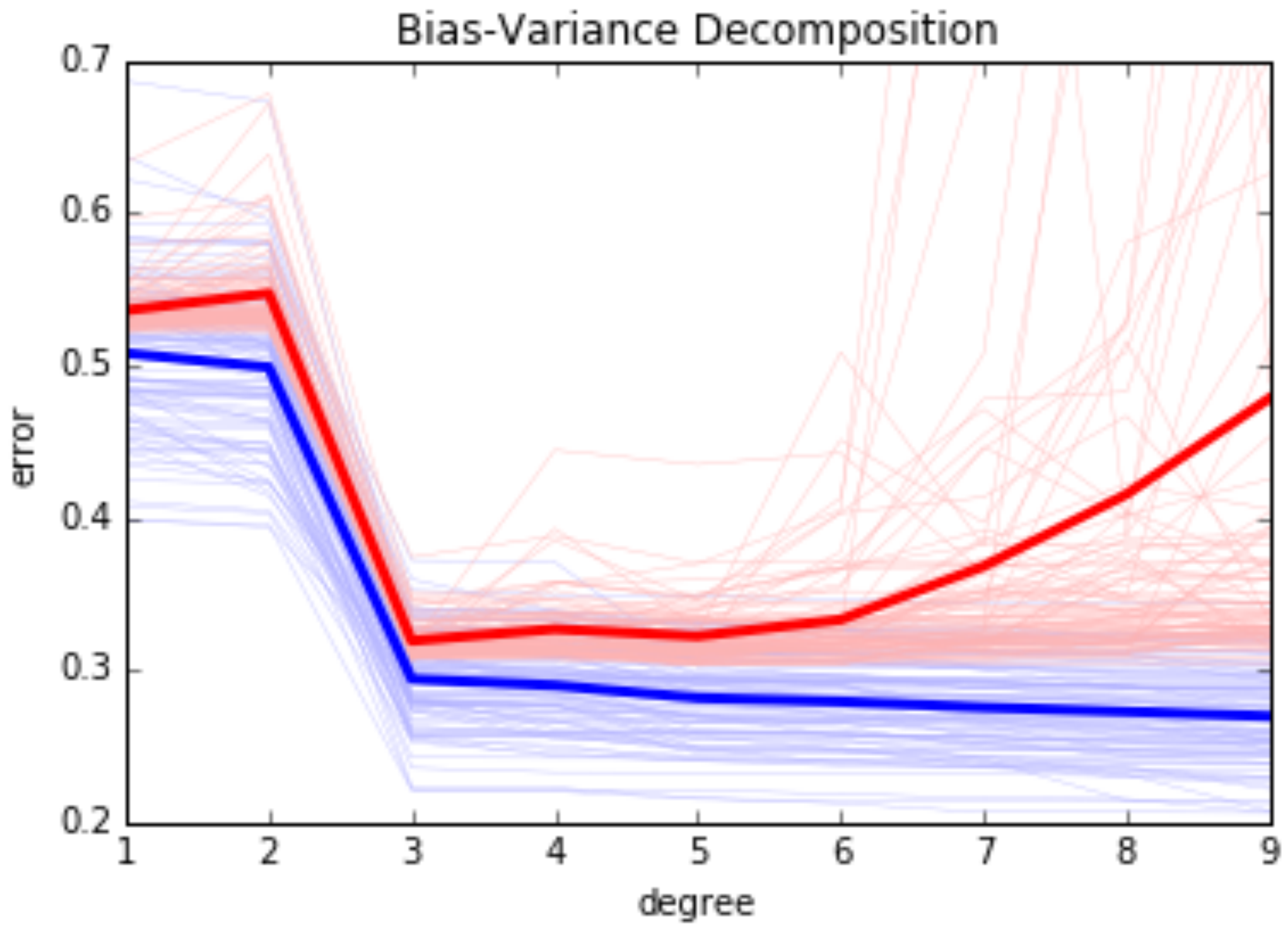
Challenge: Identify a method that ensures both low variance and low bias



Model selection curves

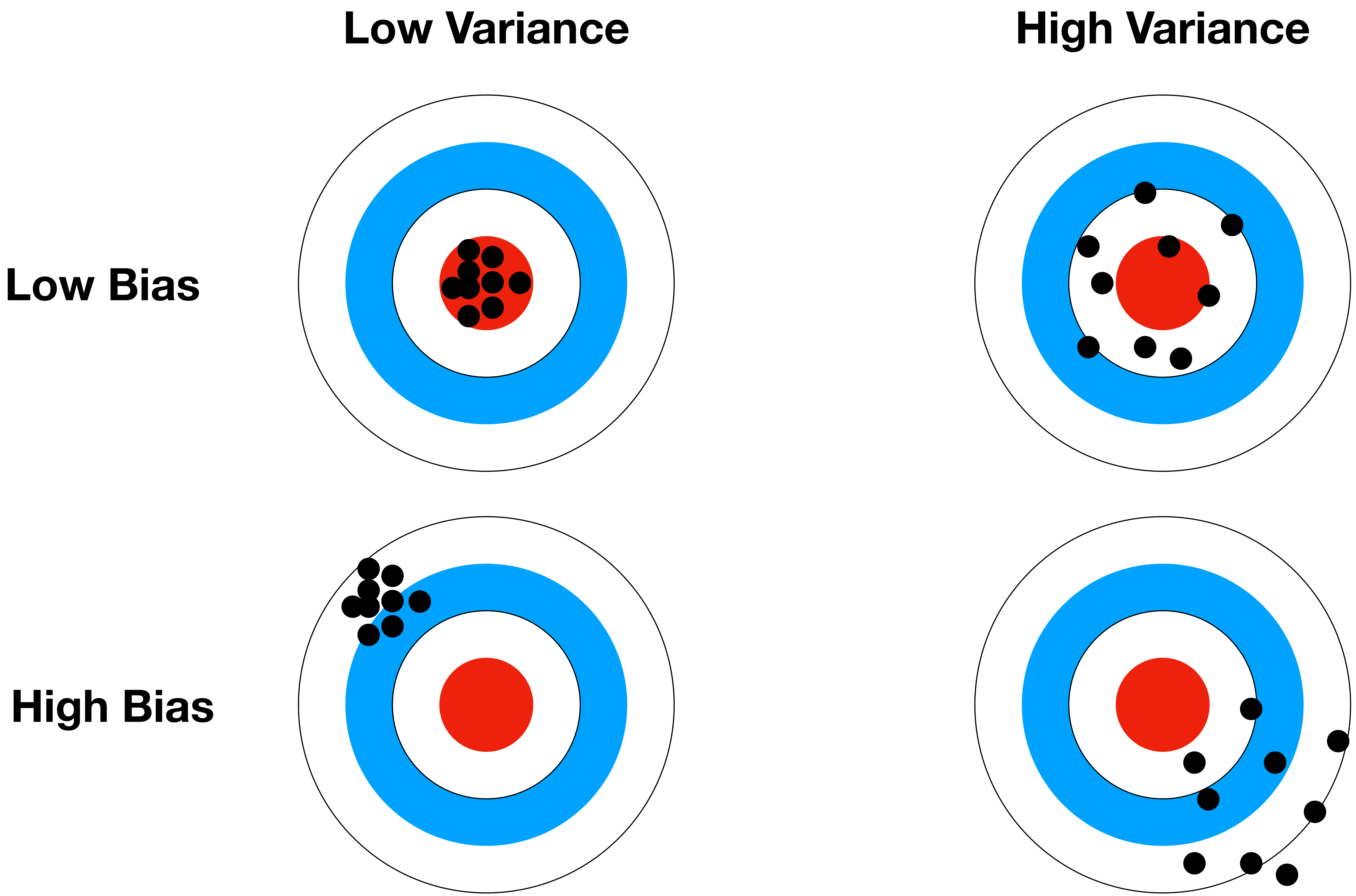


Ridge regression



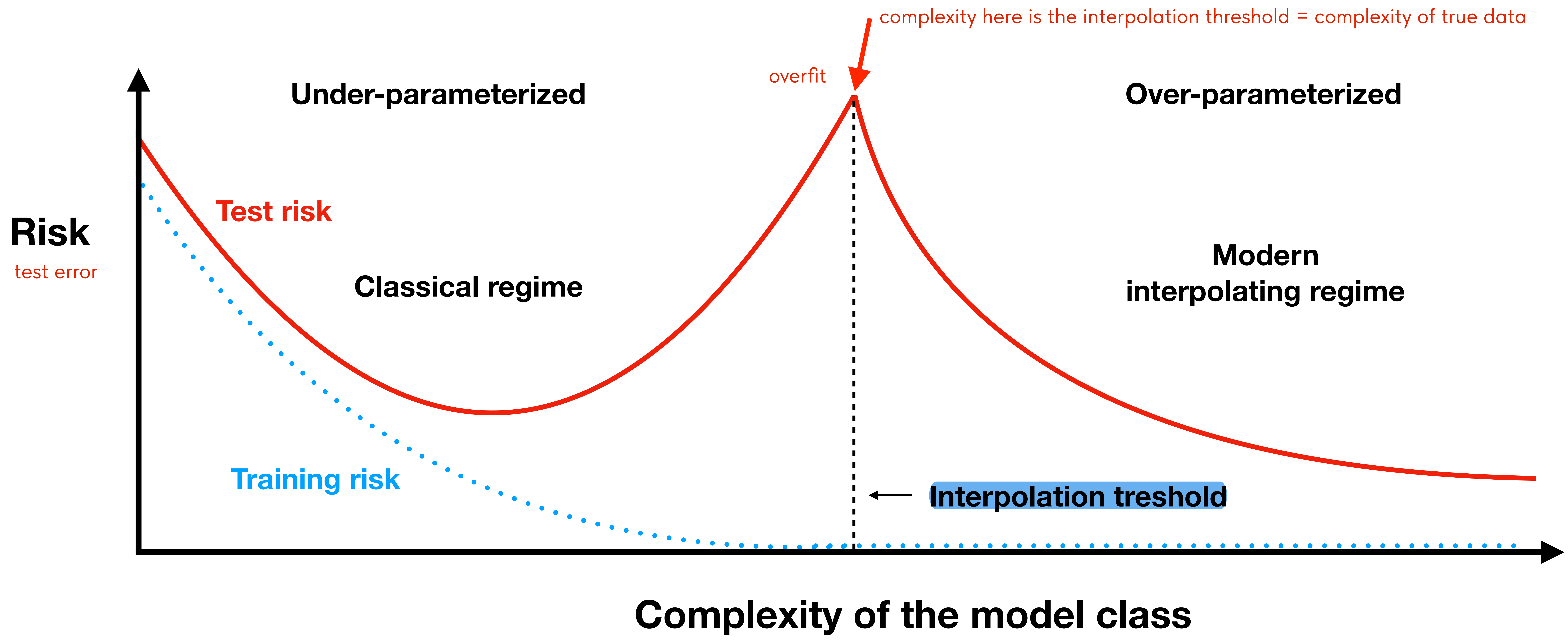
Degree in case of a polynomial feature expansion

Conclusion



But this depends on the algorithm!

Double descent curve



Reconciling modern machine-learning practice and the classical bias–variance trade-off

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal, PNAS, 2019