Bias-Variance Decomposition

Machine Learning Course - CS-433
Oct 14, 2021
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

Today

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

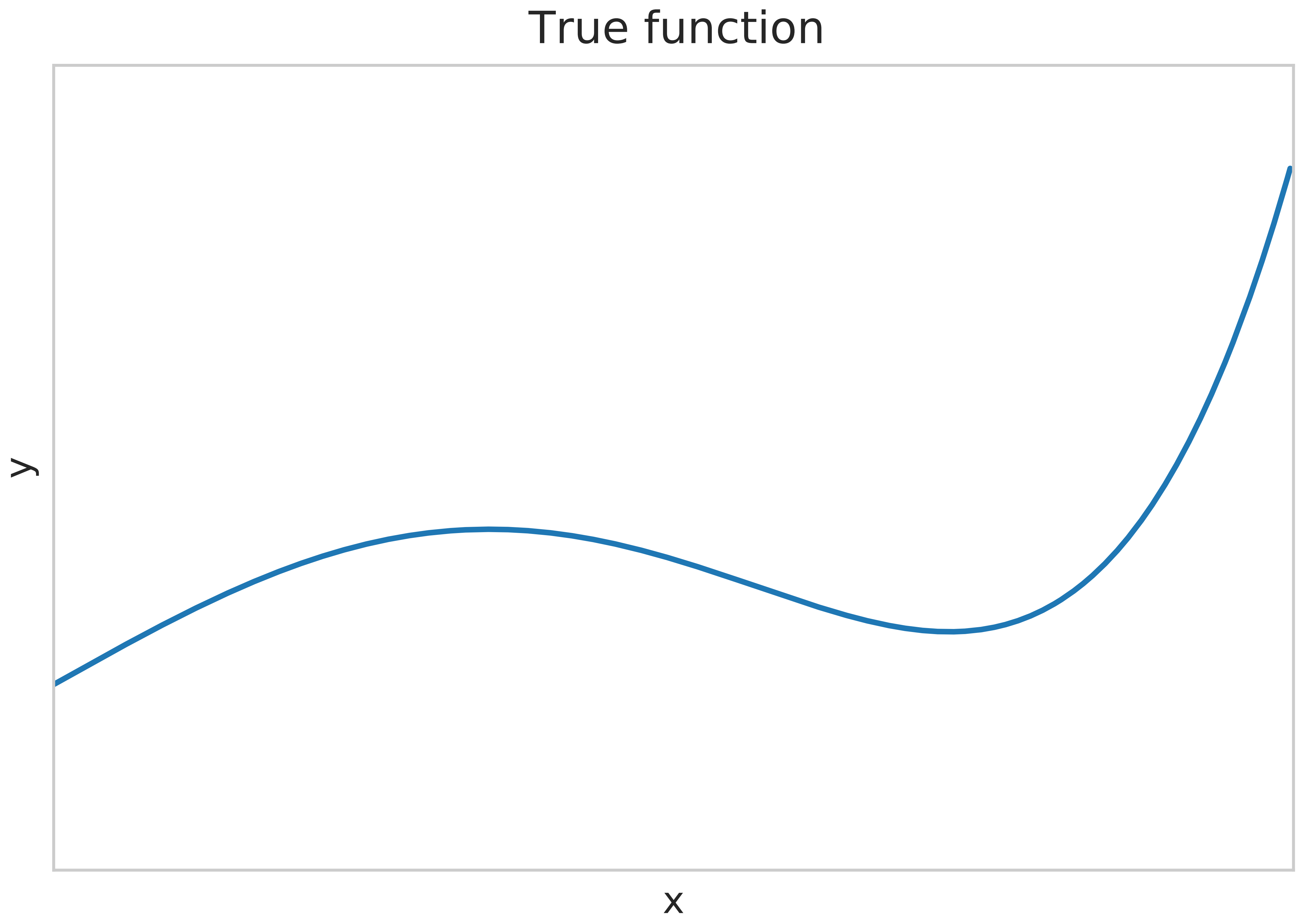
➡ ***Bias-Variance tradeoff***

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

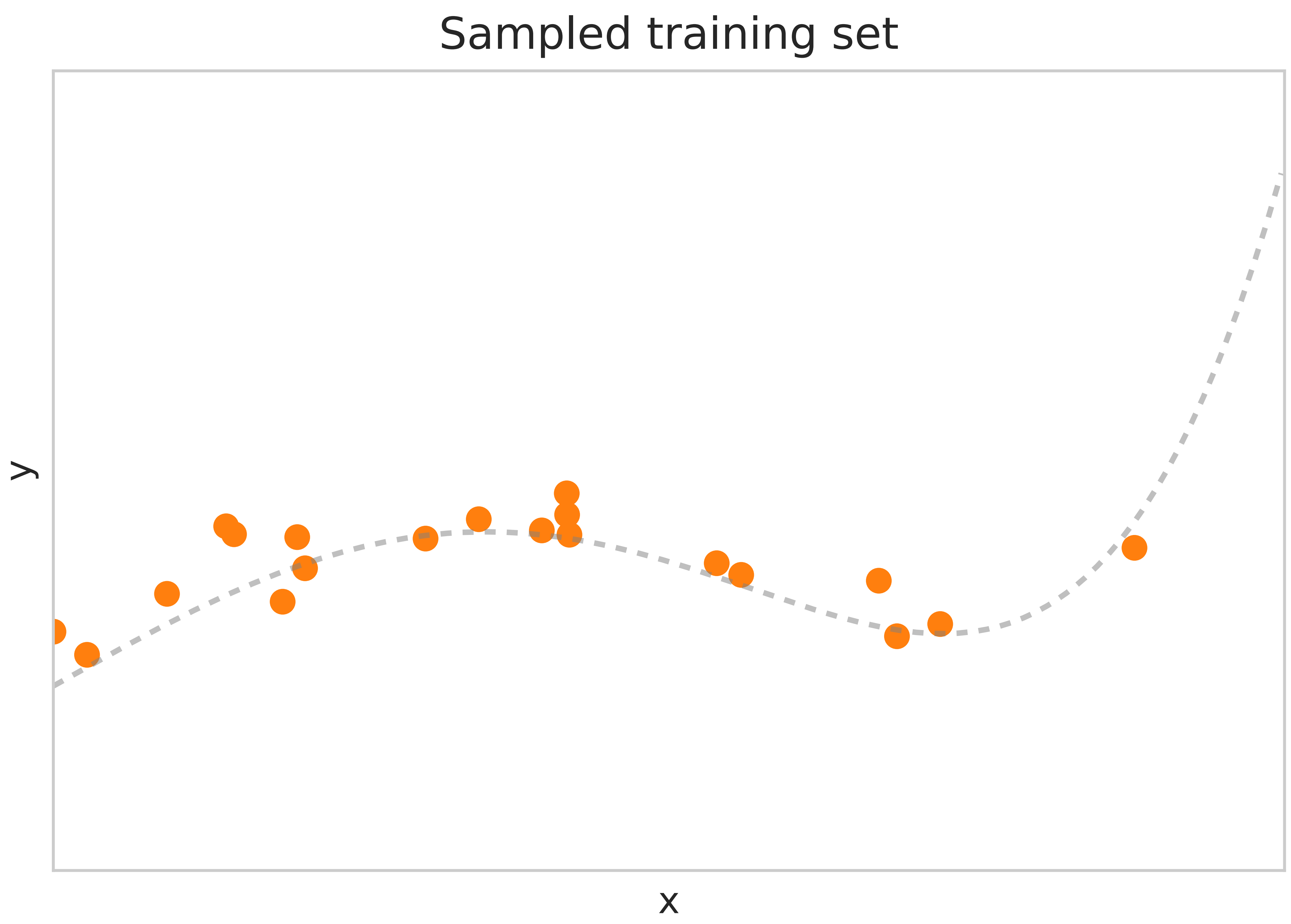
⚠ Before: quantitative

Now: ***qualitative***

A small experiment: 1D-regression



A small experiment: 1D-regression

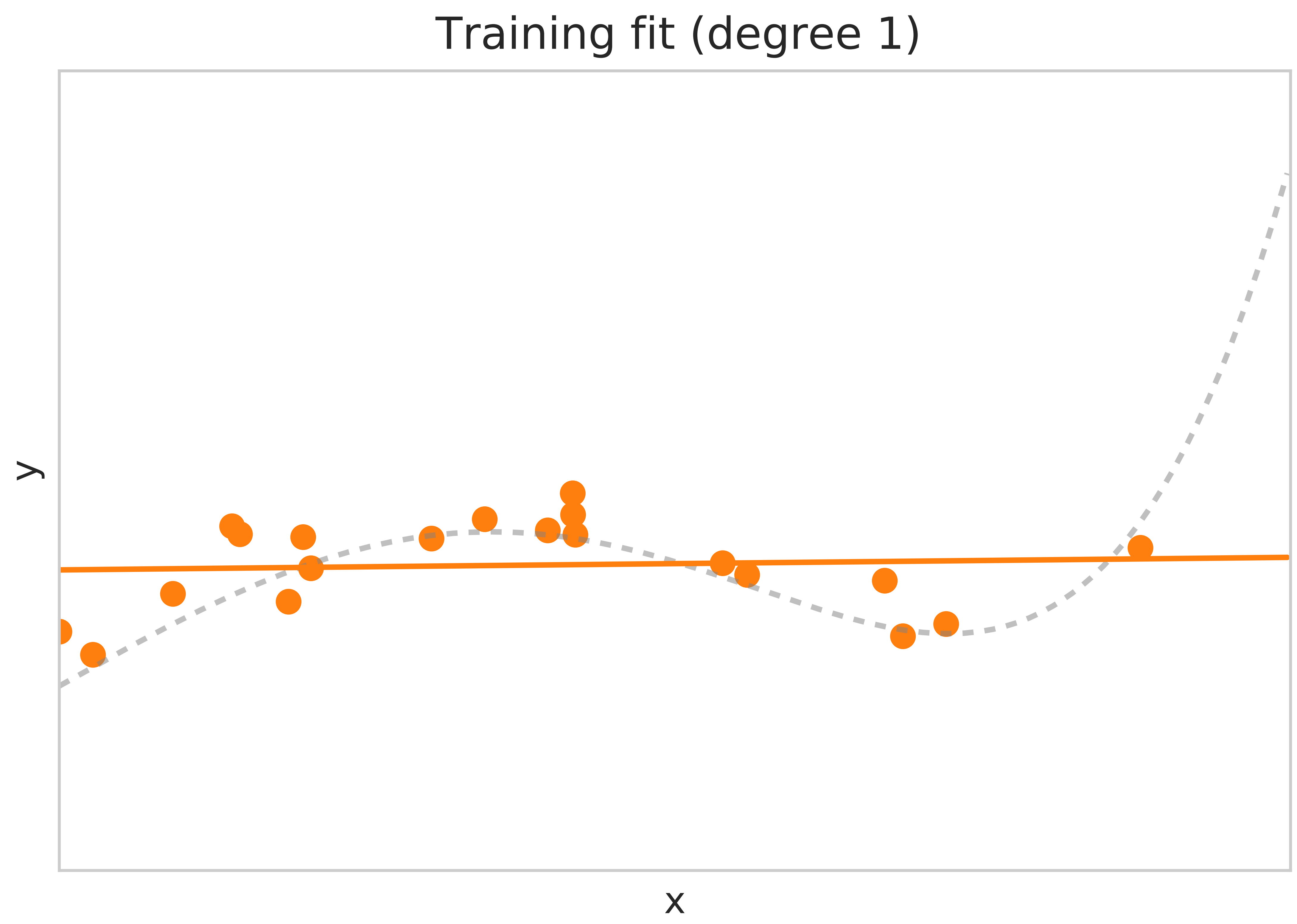


Linear regression using polynomial feature expansion $(x, x^2, x^3, \cdots, x^d)$

The maximum degree d measures the complexity of the class

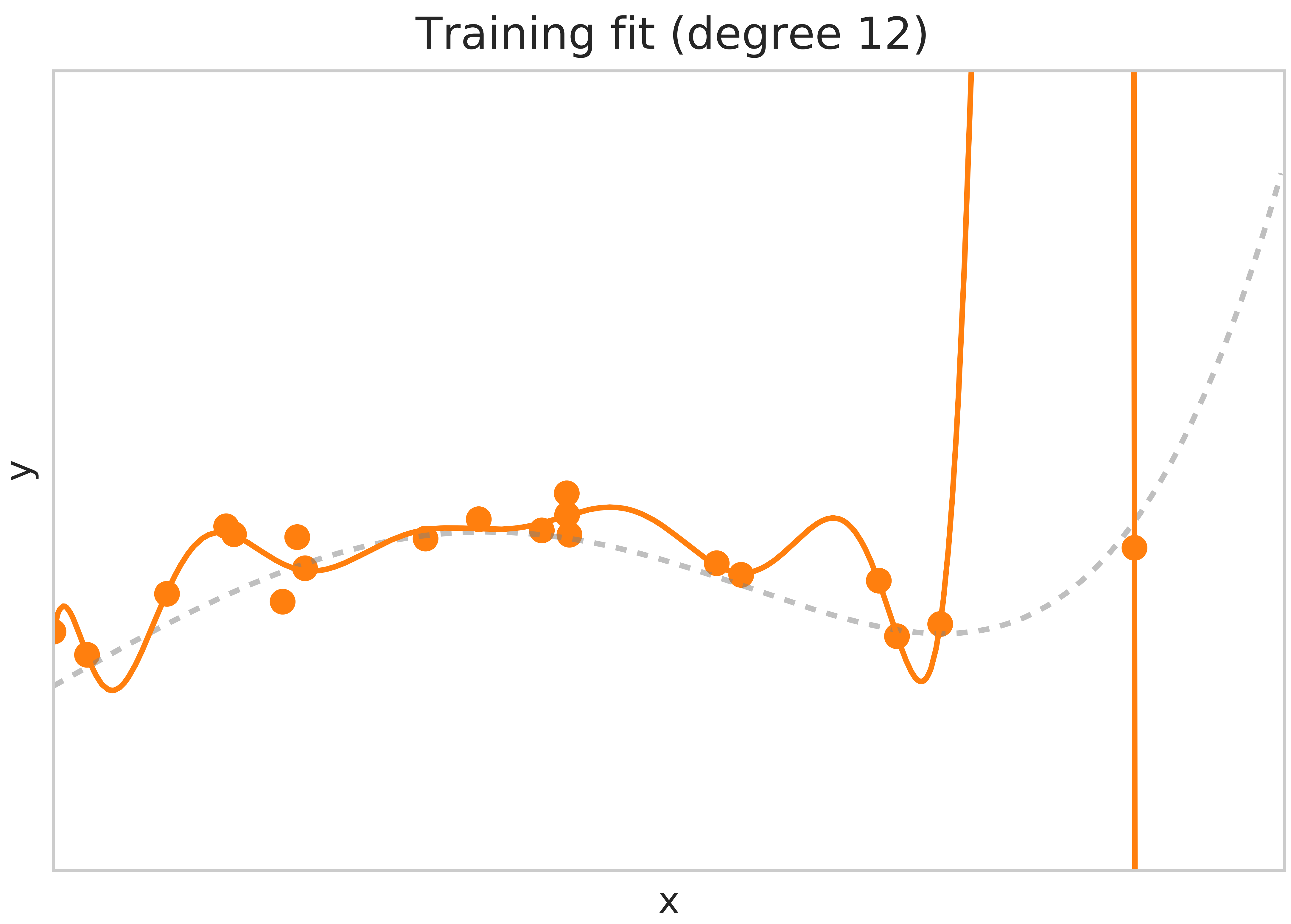
➡ How far should you go?

Simple model: bad fit



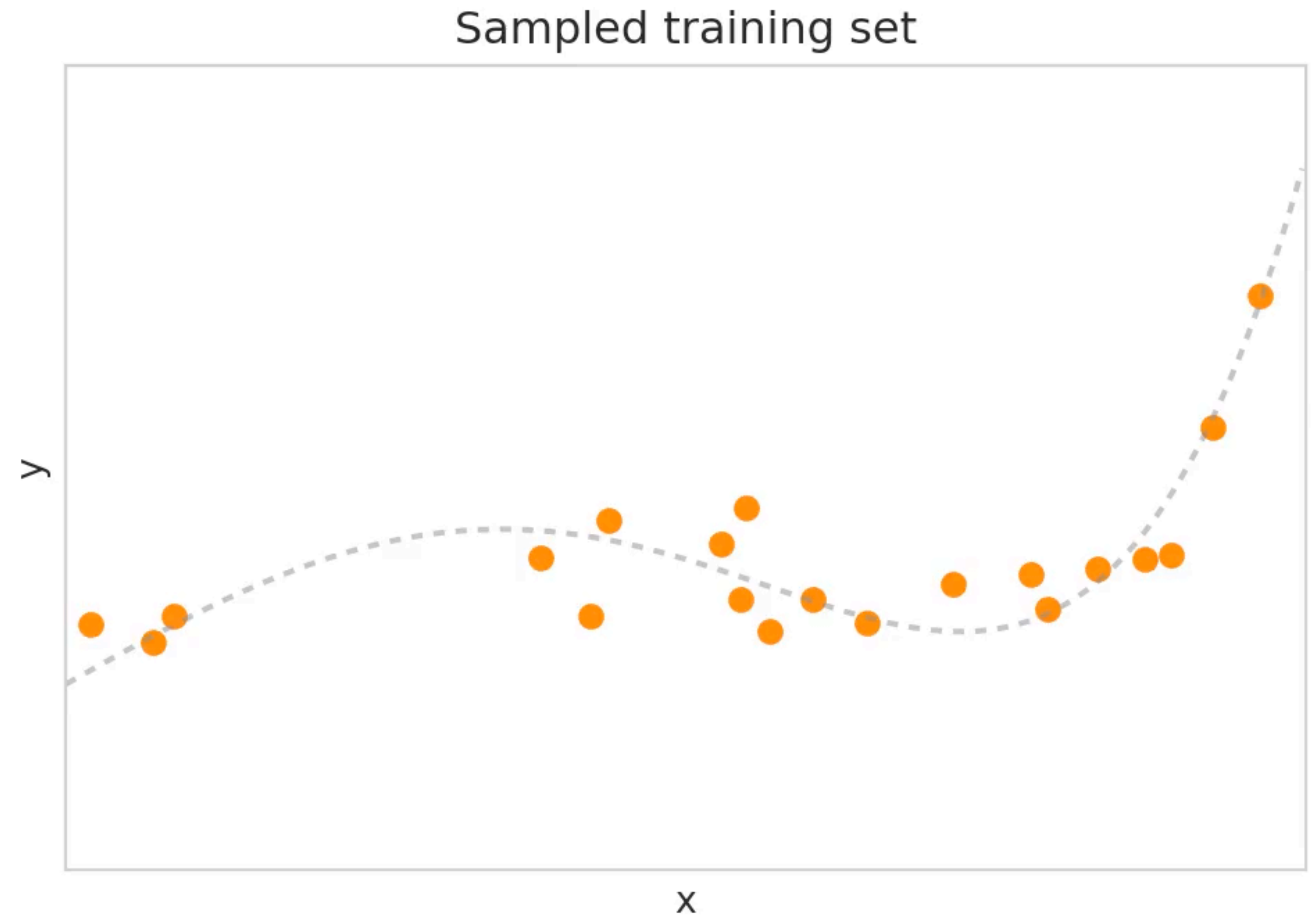
No linear function would be a good predictor. The model class is not rich enough

Complex model: good fit?



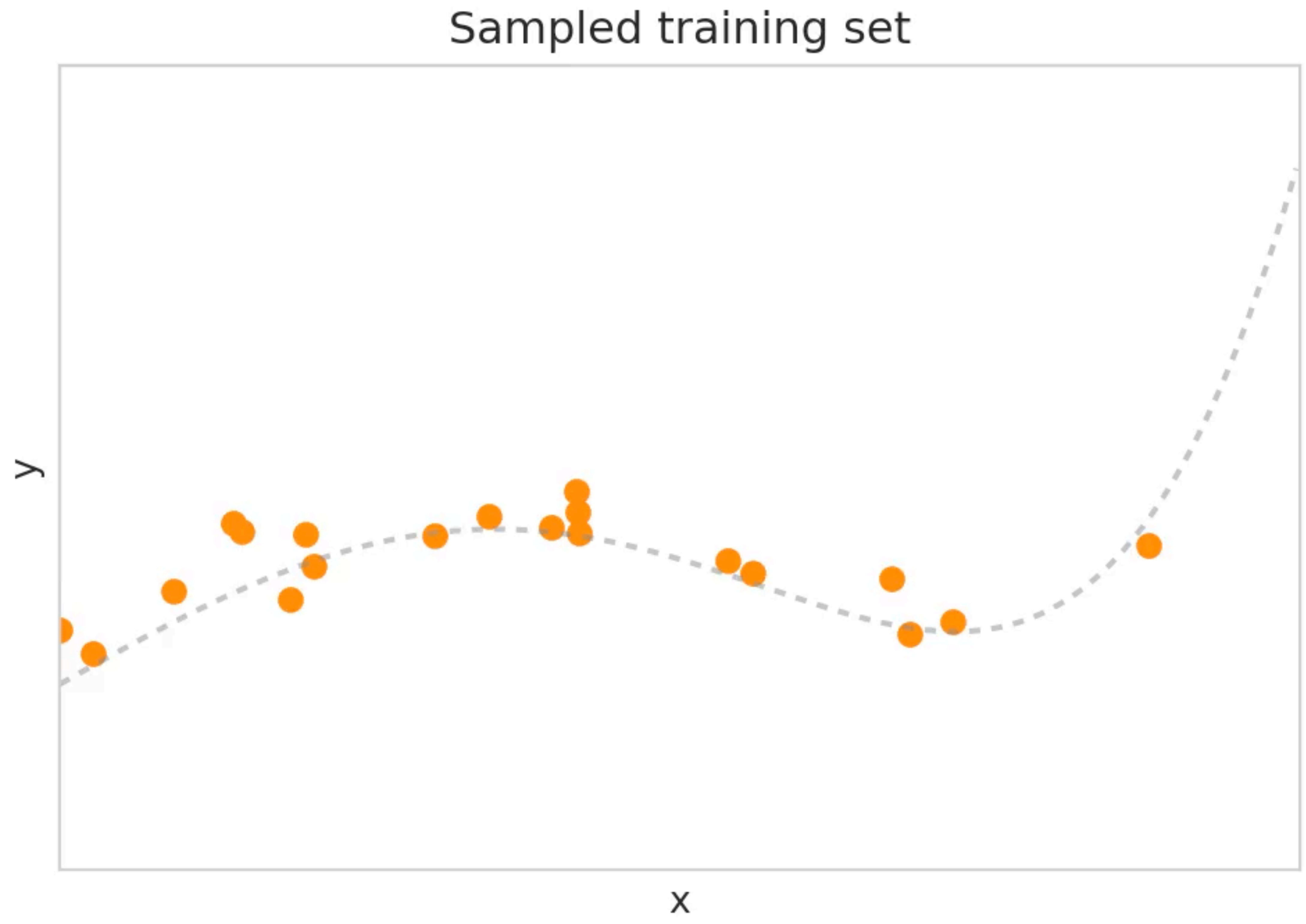
High degree polynomial will be a good fit. But?

But there is randomness in the data



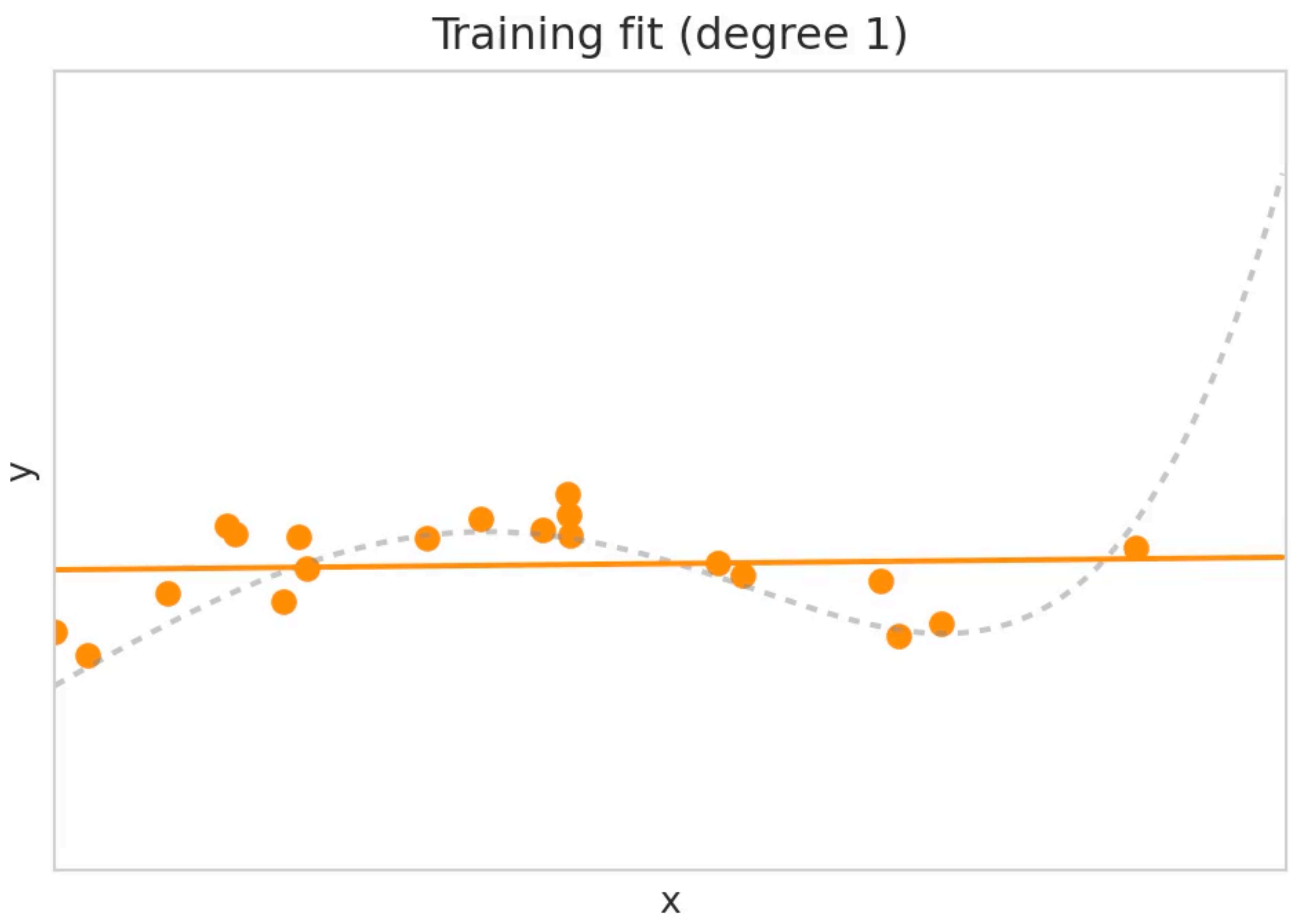
We have observed one particular S_{train} but we could have observed several others!

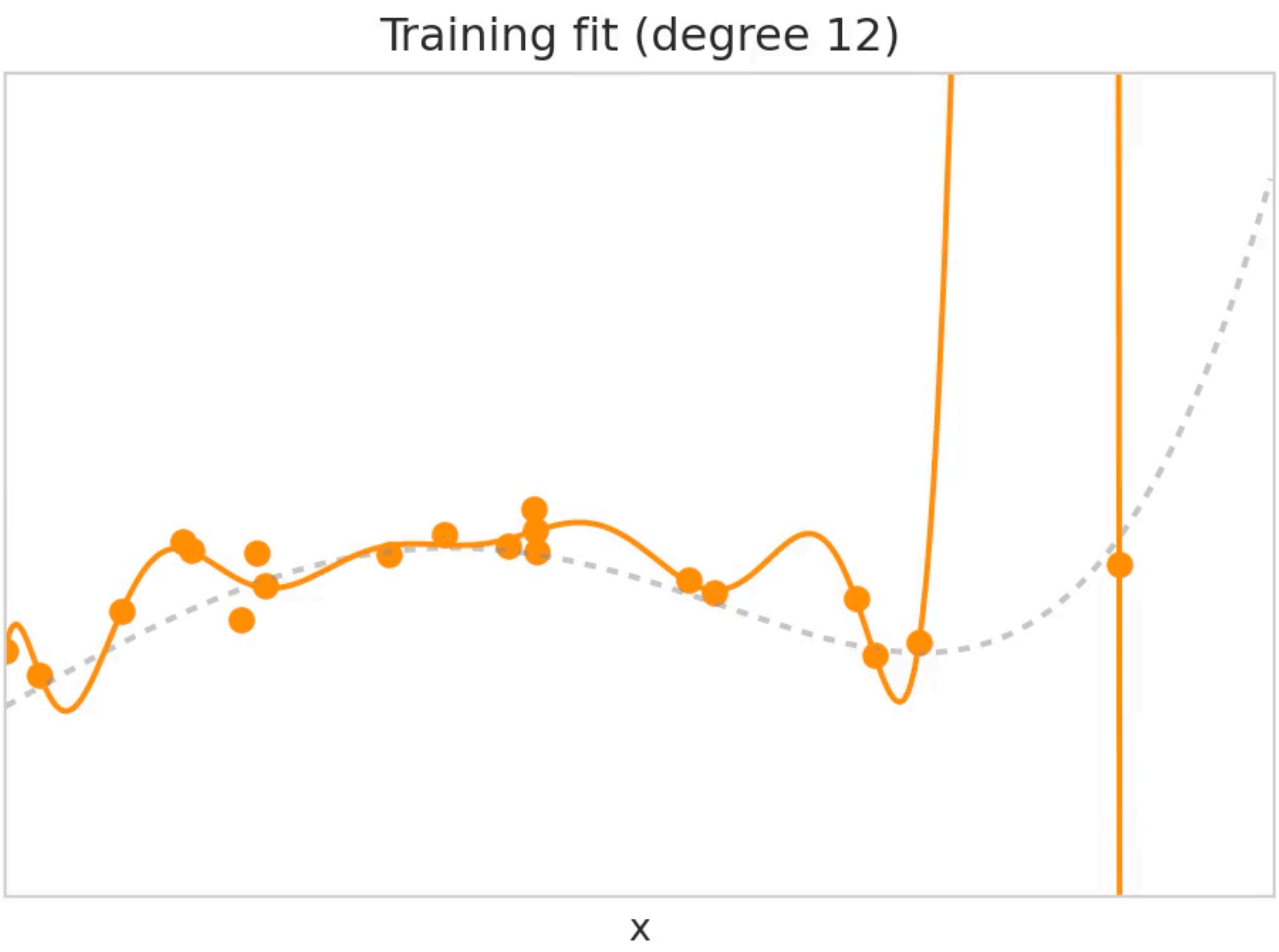
But there is randomness in the data



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

Simple models are less sensitive





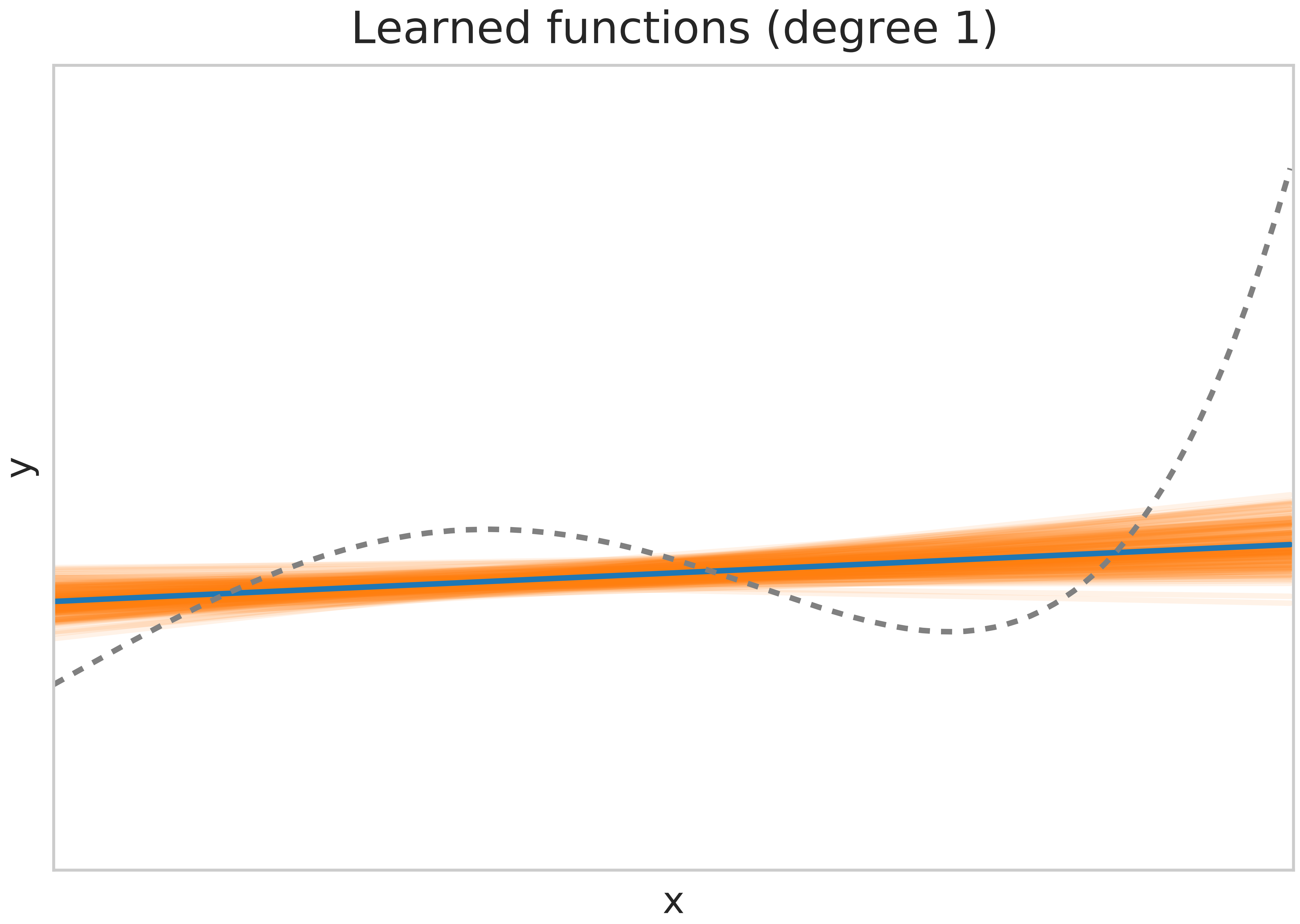
Moving a single observation will cause only a small shift in the position of the line

Changing one of the datapoint may change the prediction considerable

Underfitting

Overfitting

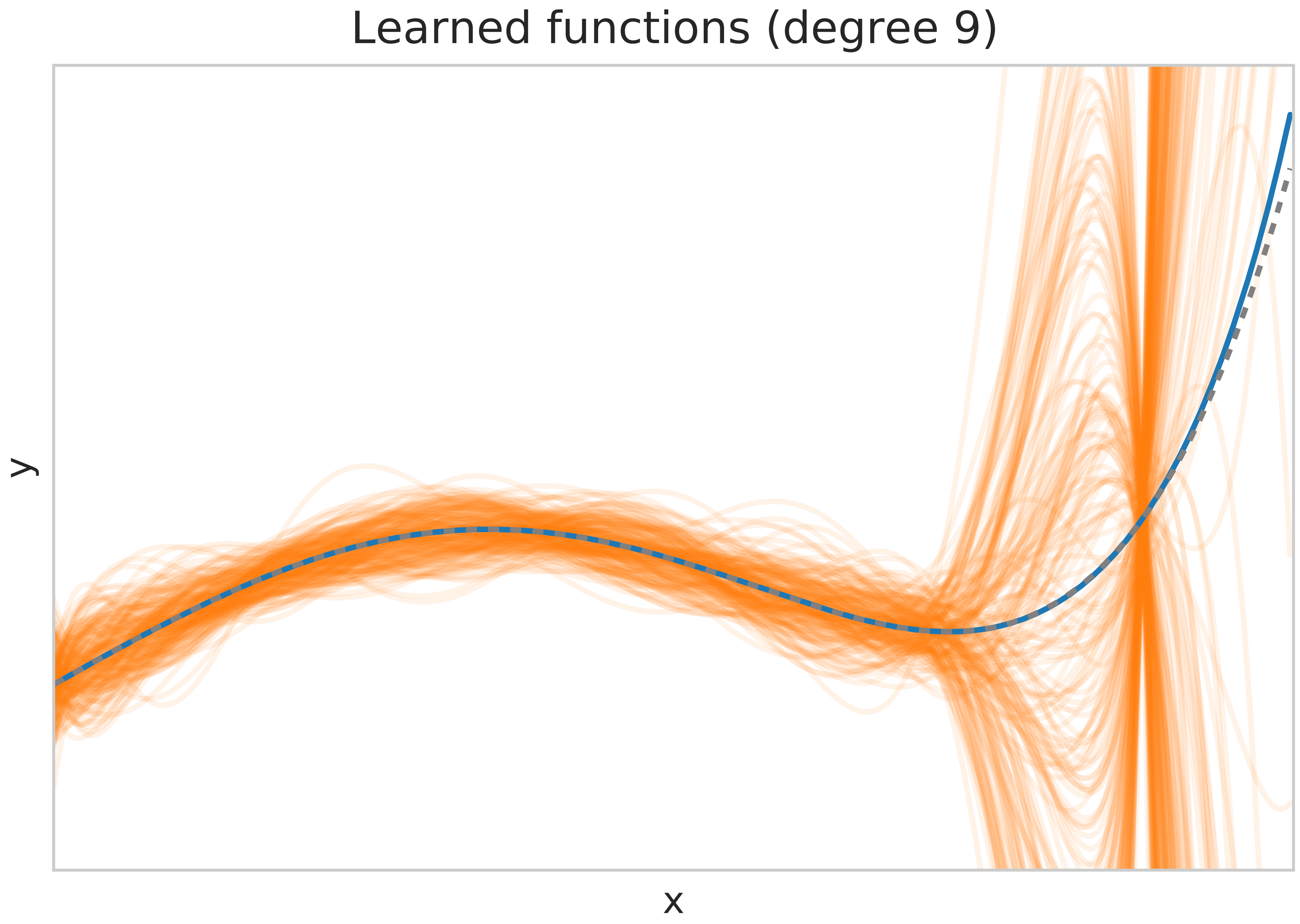
Simple models have large bias but low variance



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**

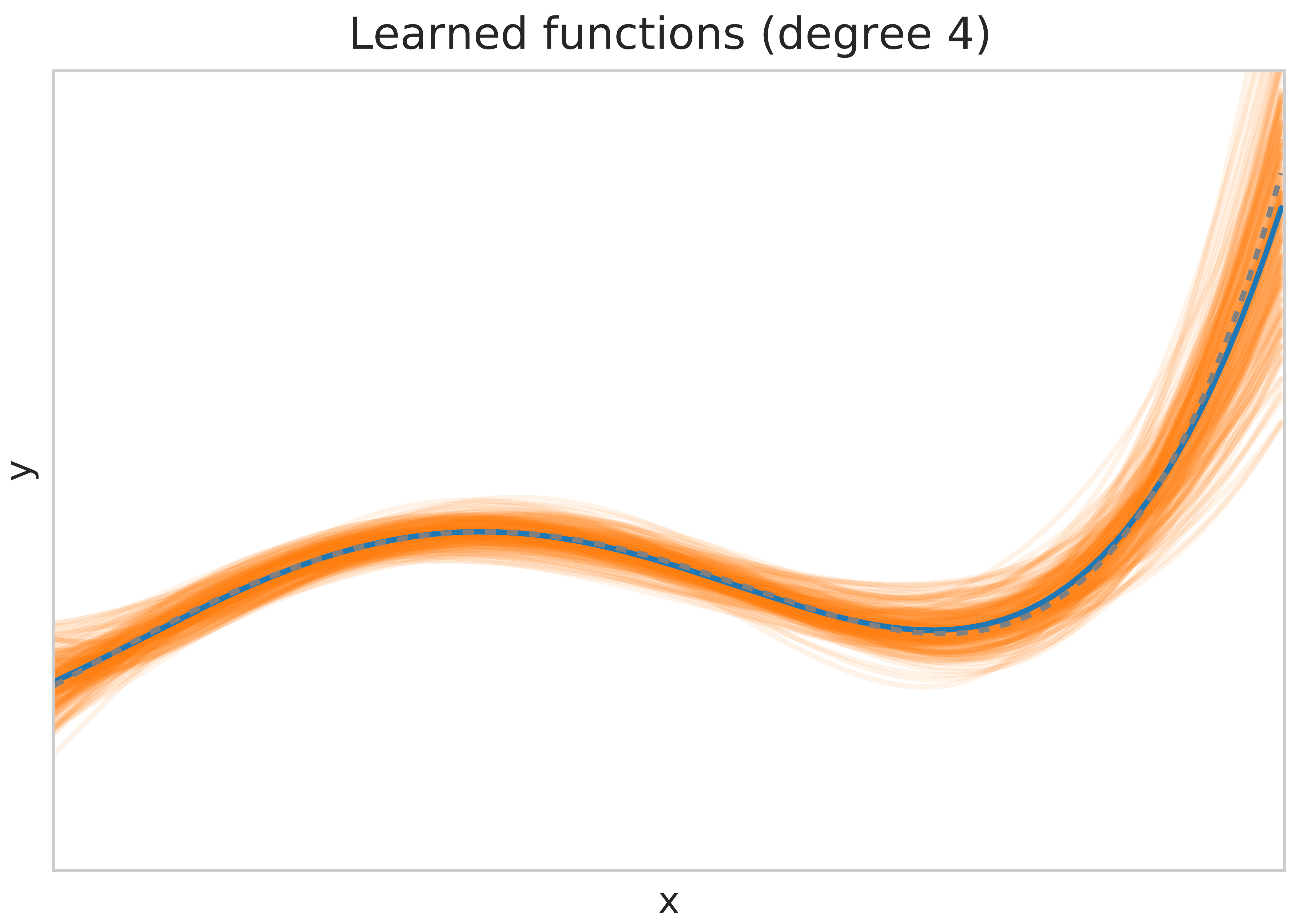
Complex models have low bias but high variance



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

$$y = f(x) + \varepsilon$$

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 for the true error

Error Decomposition

$S \sim \mathscr{D}$ i.i.d. $\qquad \mathscr{A} \qquad f_S = \mathscr{A}(S)$

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

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

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

Error Decomposition

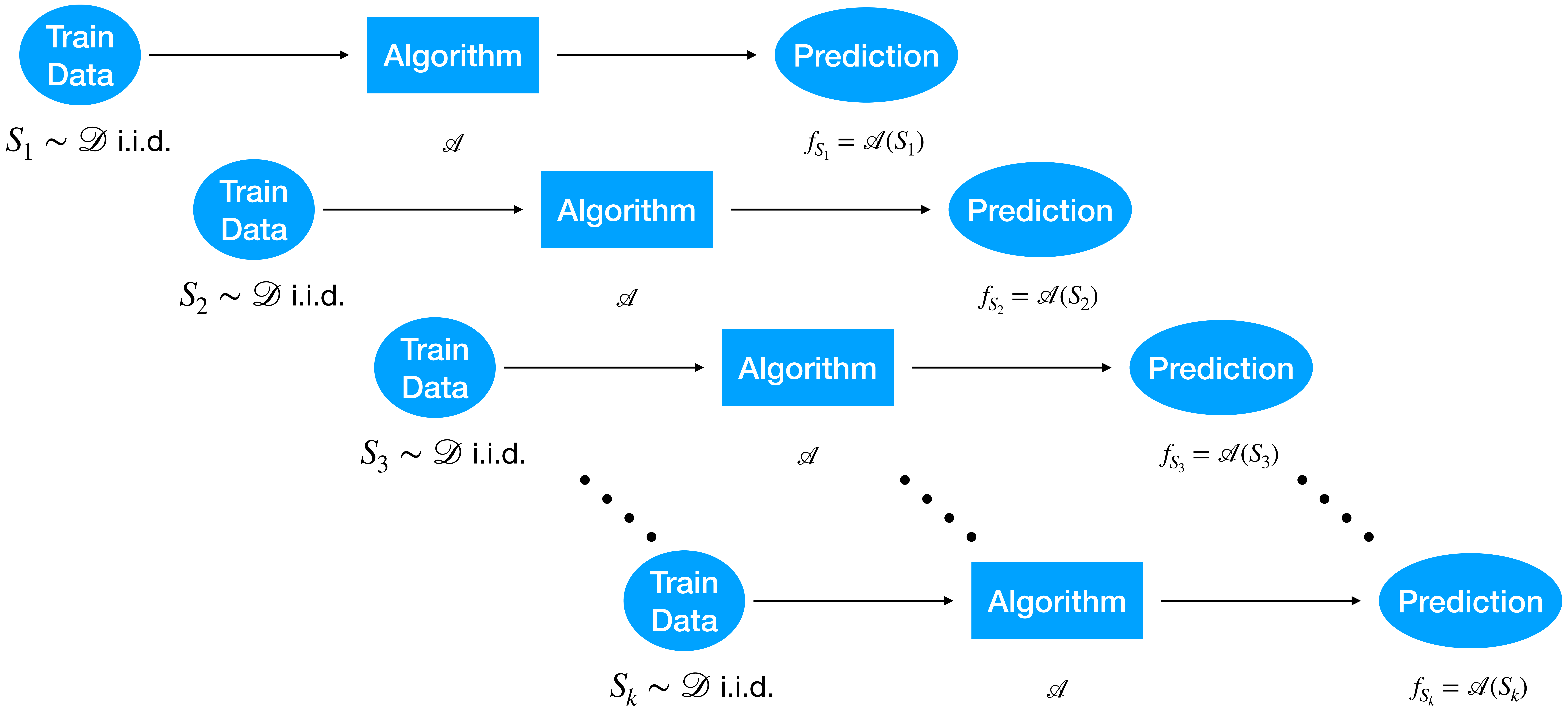
$S \sim \mathscr{D}$ i.i.d. $\qquad \mathscr{A} \qquad f_S = \mathscr{A}(S)$

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

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

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:

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

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

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

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

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

Therefore

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

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

$$\mathbb{E}_{S\sim\mathcal{D}}[(f(x_0)-f_S(x_0))^2] = \mathbb{E}_{S\sim\mathcal{D}}\left[(f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]+\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]-f_S(x_0))^2\right]$$
$$= \mathbb{E}_{S\sim\mathcal{D}}\Big[(f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)])^2+(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]-f_S(x_0))^2$$
$$+2(f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)])(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]-f_S(x_0))\Big]$$

Cross-term:

$$\mathbb{E}_{S\sim\mathcal{D}}\left[\left(f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]\right)\cdot\left(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]-f_S(x_0)\right)\right]$$
$$=\left(f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]\right)\cdot\mathbb{E}_{S\sim\mathcal{D}}\left[(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]-f_S(x_0))\right]$$
$$=\left(f(x_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]\right)\cdot\left(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)]-\mathbb{E}_{S\sim\mathcal{D}}[f_S(x_0)]\right)=0.$$

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

Bias-Variance Decomposition

We obtain the following decomposition in three positive terms:

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

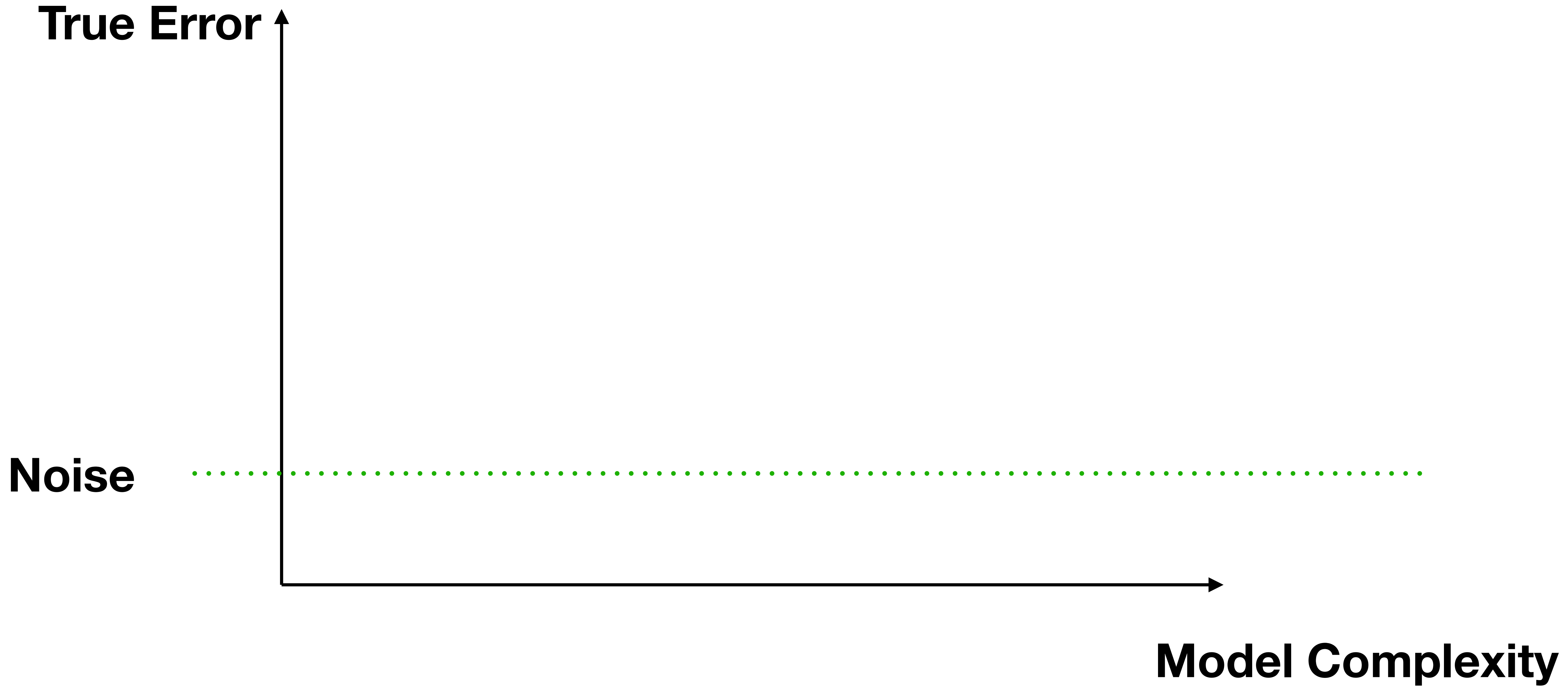
$$\textbf{Bias} \longrightarrow \quad + \quad (f(x_0) - \mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(x_0)])^2$$

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

which always lower bound the true error

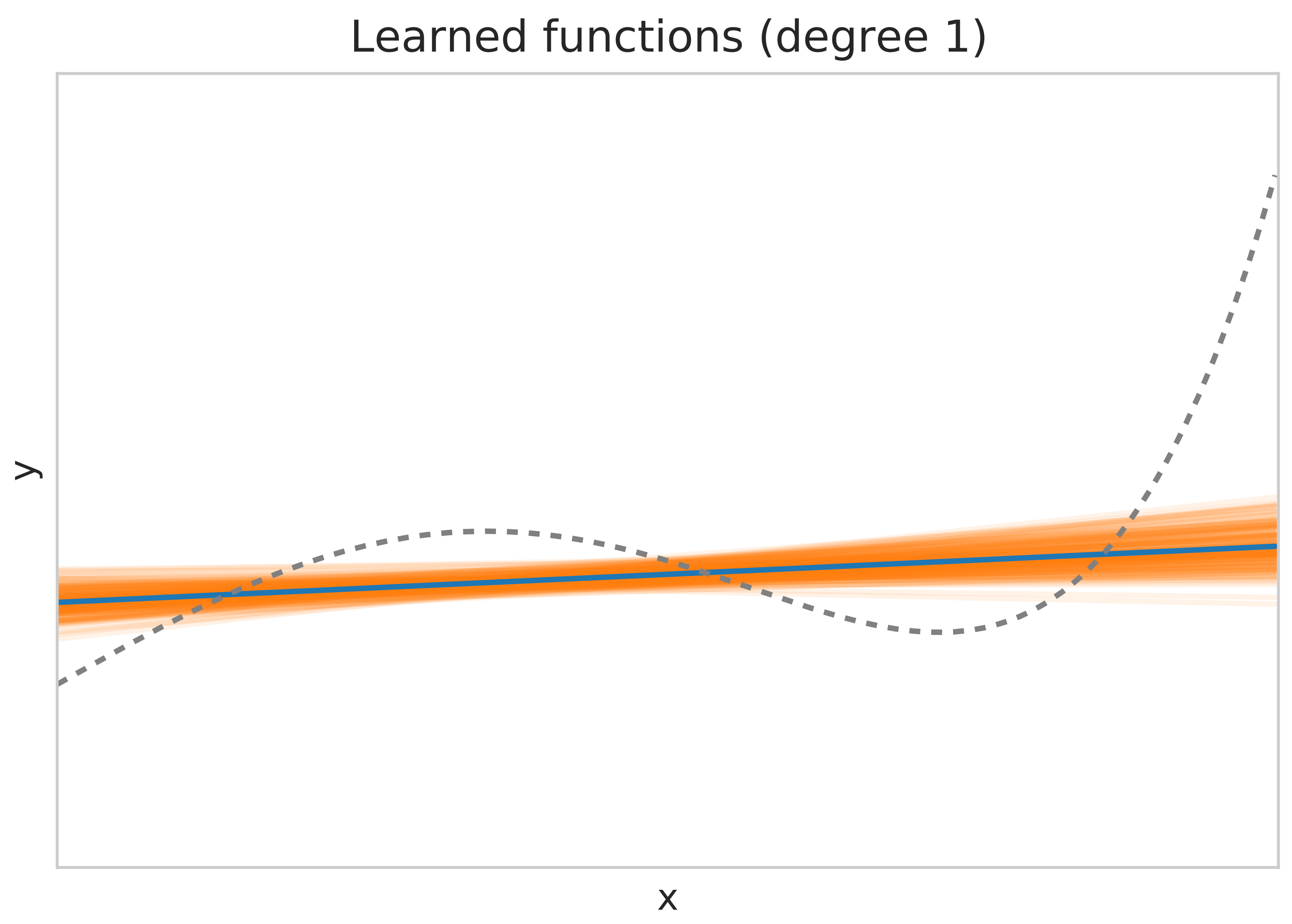
➡ In order to minimize the true error, we need to select a method that ***simultaneously achieves low bias and low variance***

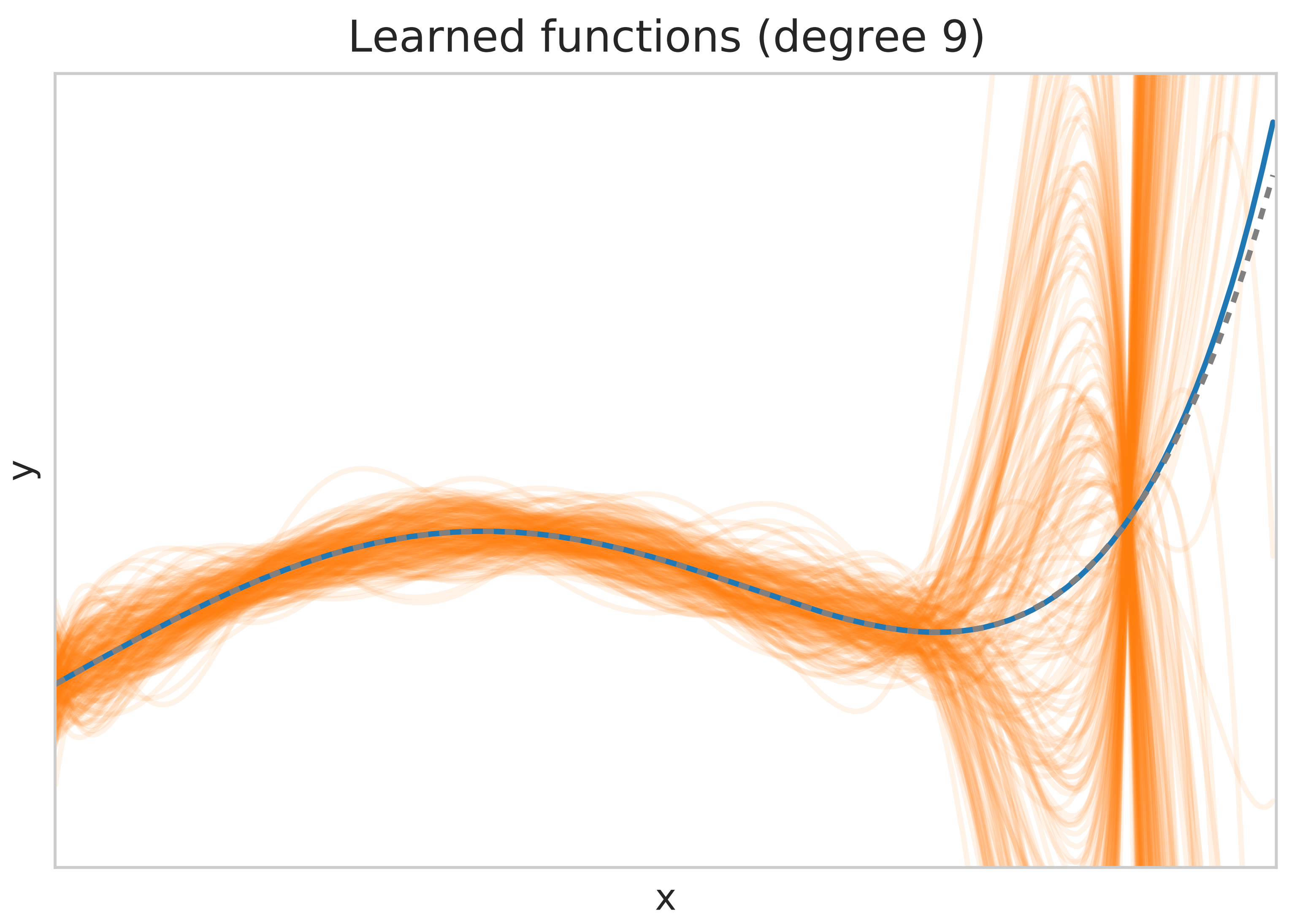
Noise: a strict lower bound on what error we can achieve.



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

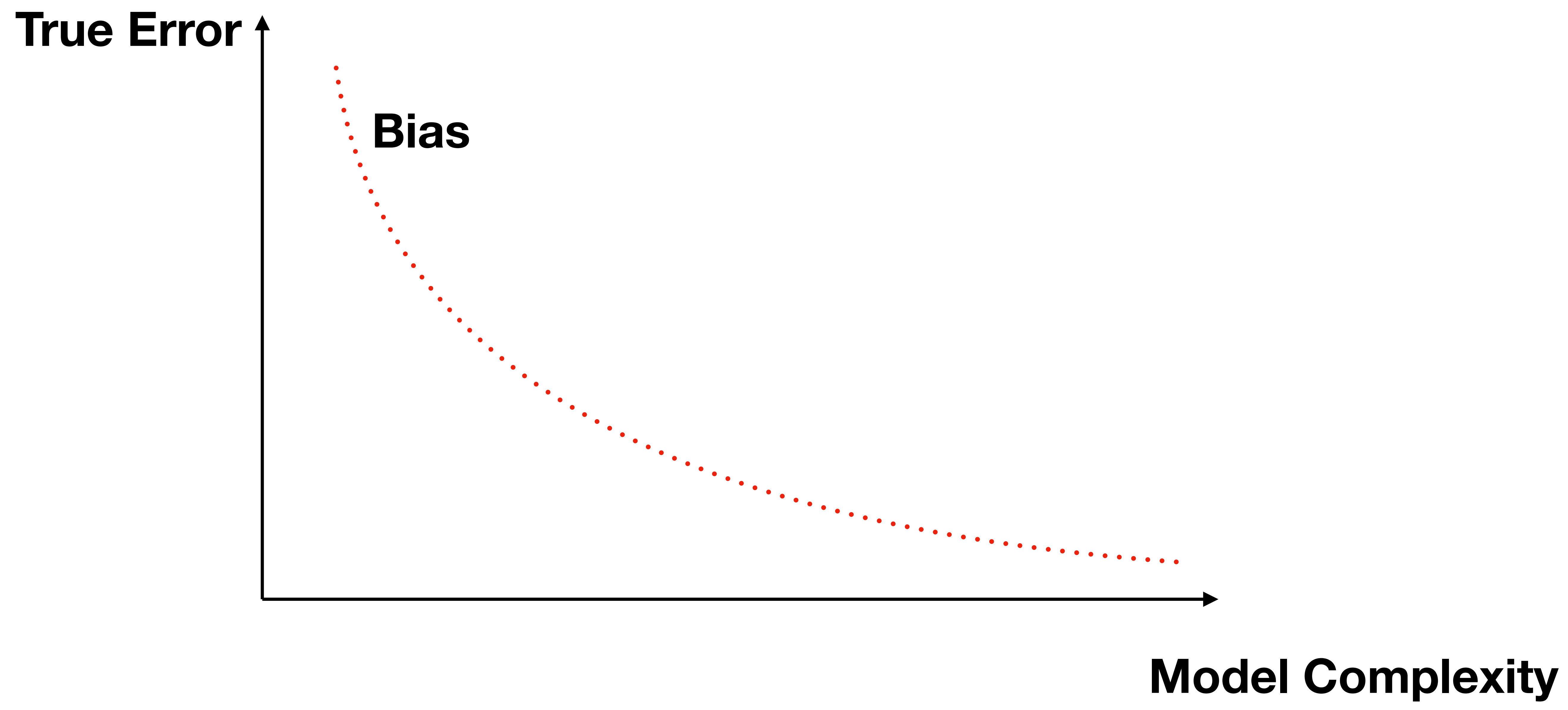
Bias: $(f(x_0) - \mathbb{E}_{S\sim\mathcal{D}}[f_S(x_0)])^2$





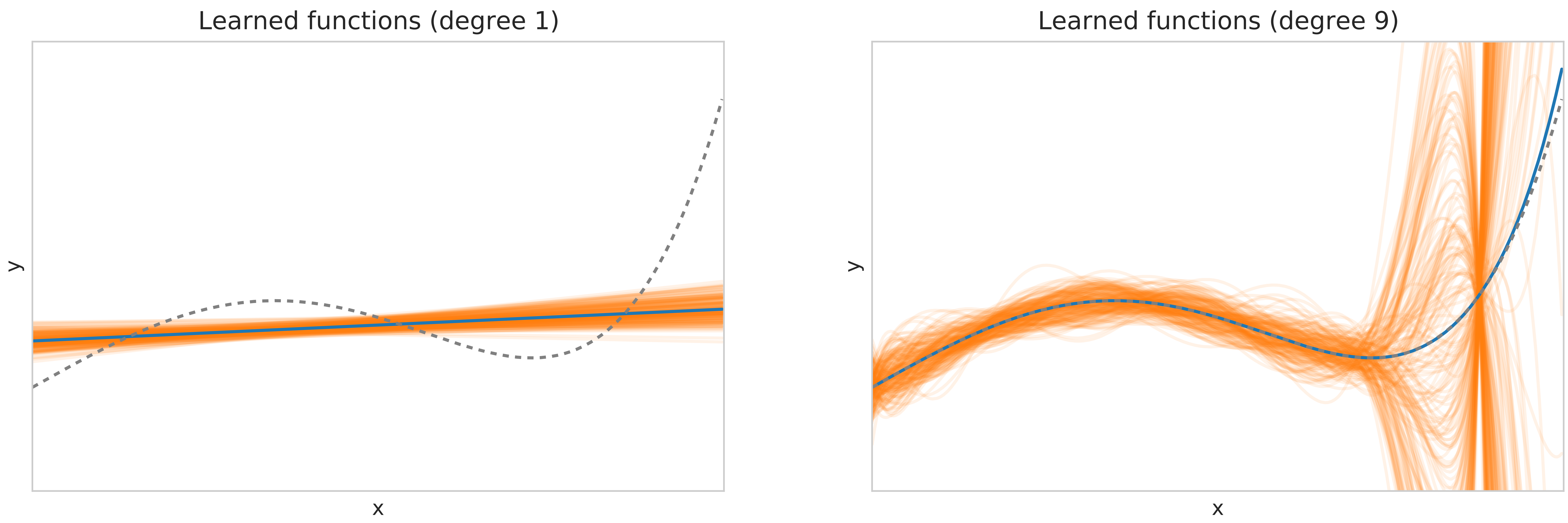
- Squared of the difference between the actual value $f(x_0)$ and the expected prediction
- It measures how far off in general the models' predictions are from the correct value
- If complexity is small then high bias
- If complexity is high then low bias

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



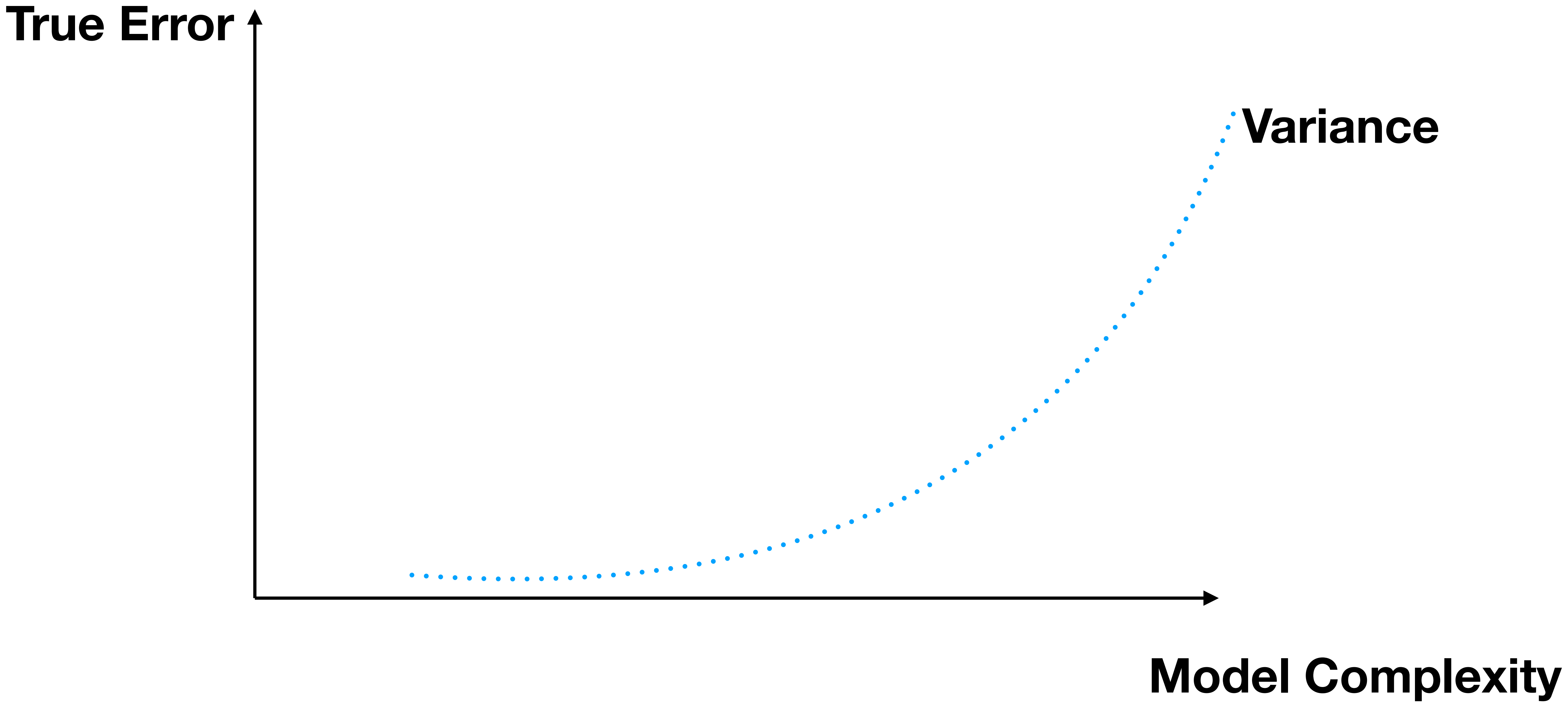
- Squared of the difference between the actual value $f(x_0)$ and the expected prediction
- It measures how far off in general the models' predictions are from the correct value
- If complexity is small then high bias
- If complexity is high then low bias

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



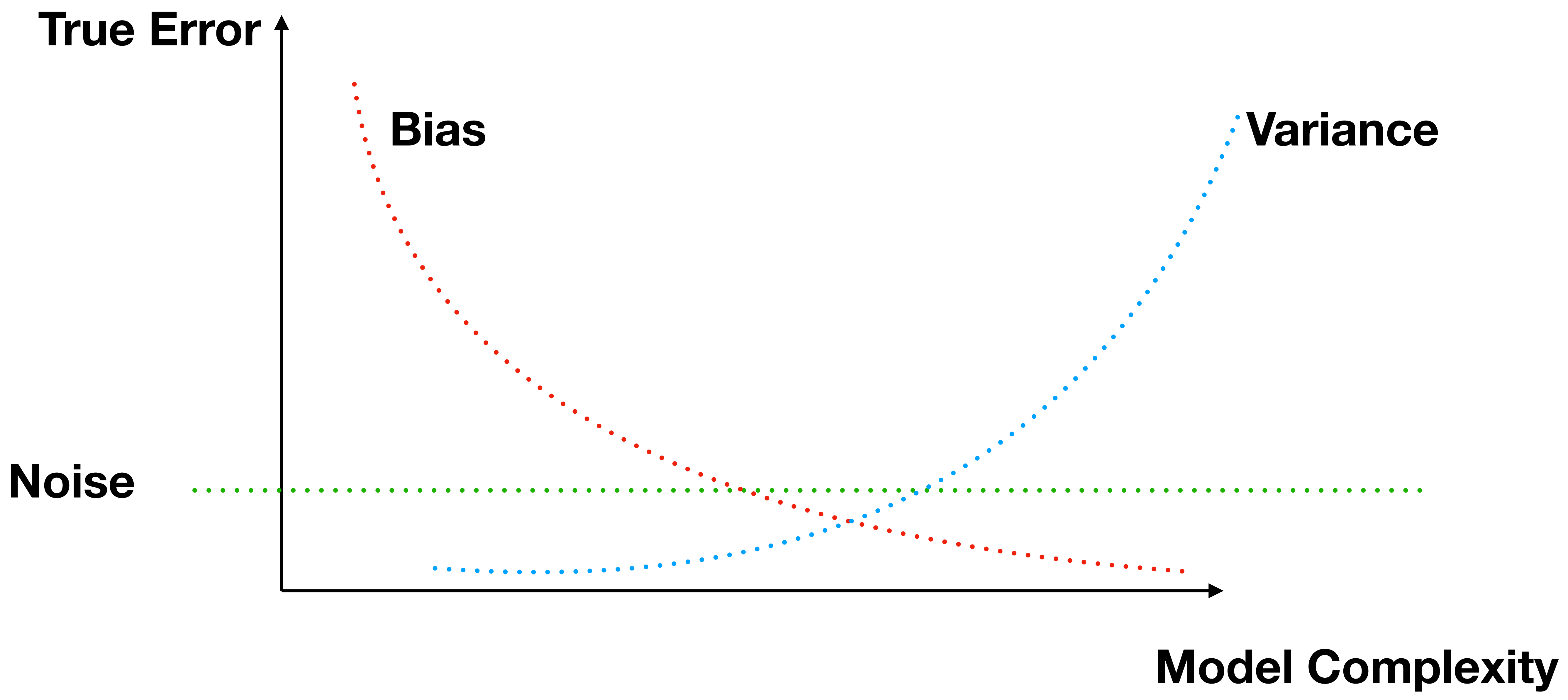
- Variance of the prediction function.
- It is how much the predictions for a given point vary between different realizations of the training set
- If we consider complicated models then small variations in the training set can result in large changes in the prediction

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



- Variance of the prediction function.
- It is how much the predictions for a given point vary between different realizations of the training set
- If we consider complicated models then small variations in the training set can result in large changes in the prediction

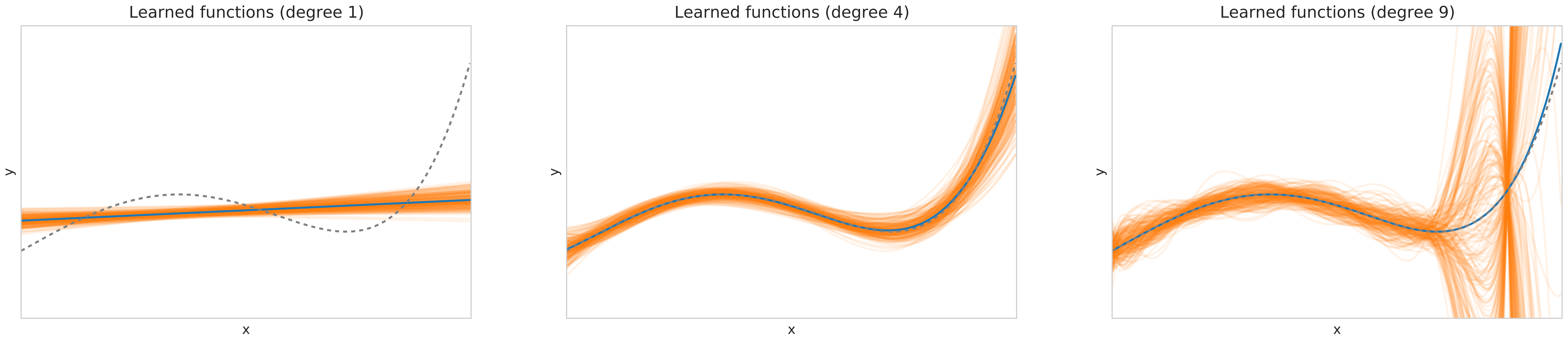
Bias Variance tradeoff and U-shape curve



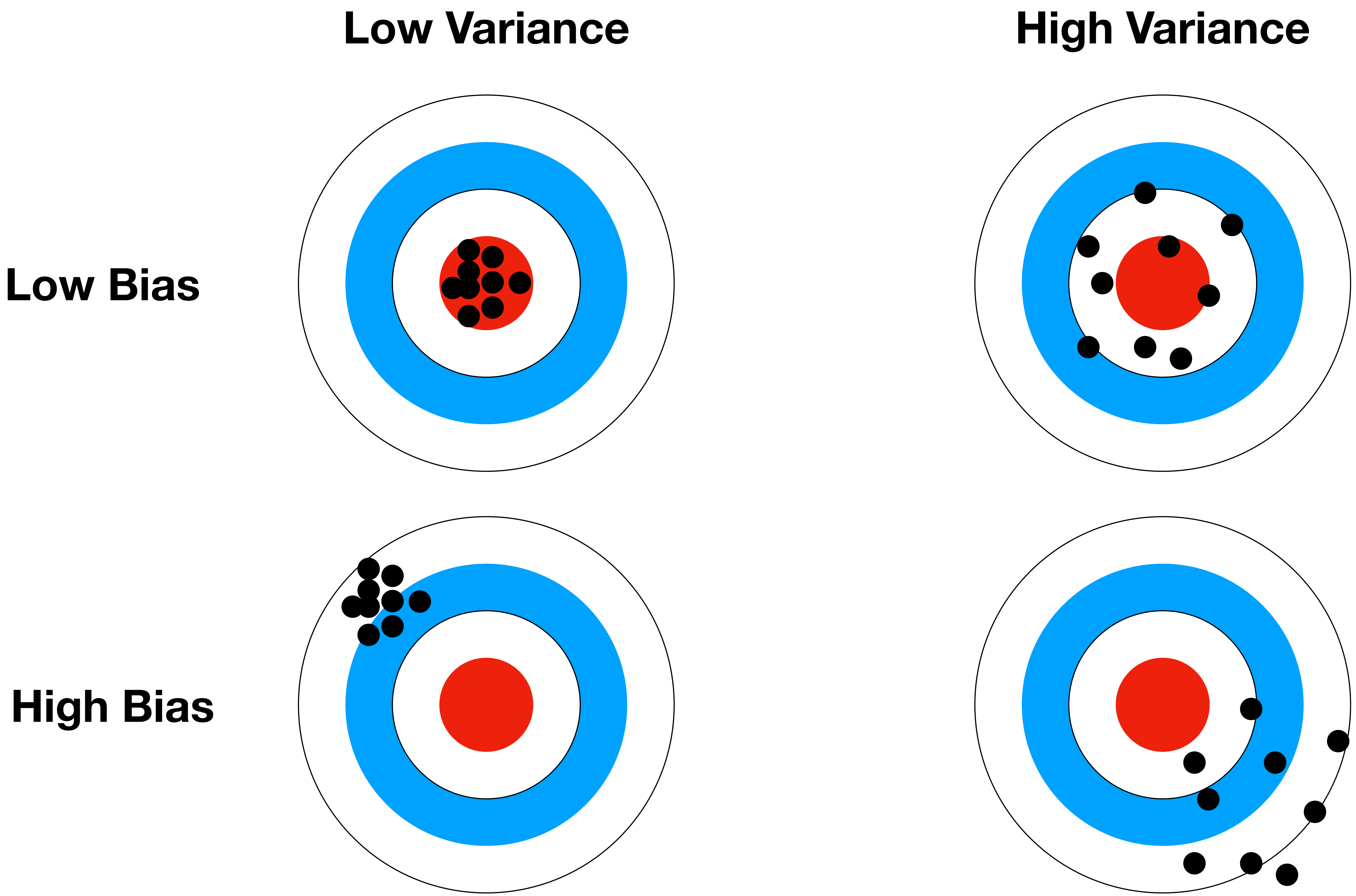
- If the complexity is too low, you cannot approximate well (underfitting)
- If the complexity is too large, you have a problem with the variance (overfitting)

➡ This is referred to as the bias-variance tradeoff

Challenge: find a method for which both the variance and the bias are low



Conclusion



But this depends on the algorithm!

Double descent curve