

# Bias-Variance Decomposition

Machine Learning Course - CS-433

Oct 11, 2023

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***

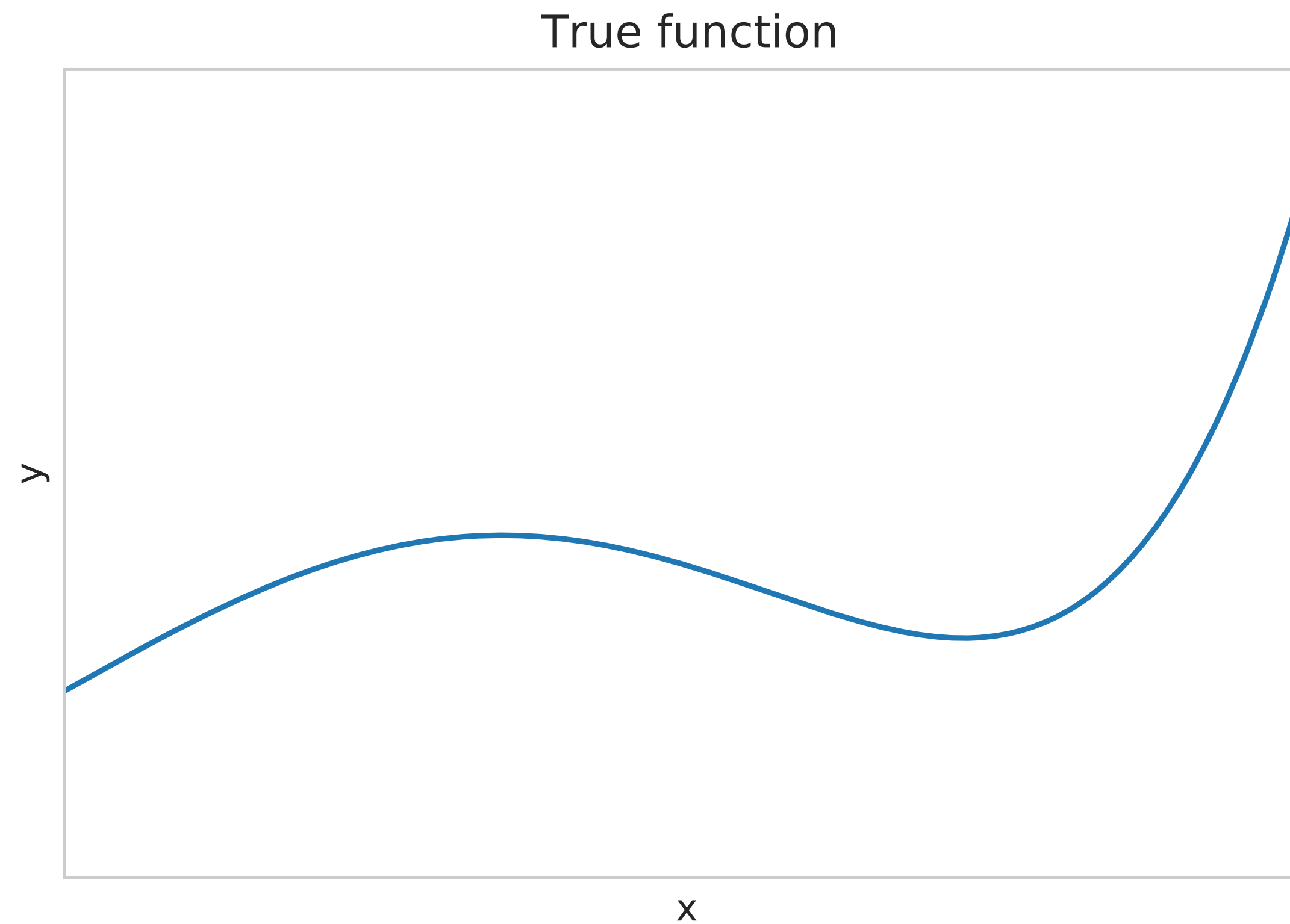
It will help us to decide how complex and rich we should make our model



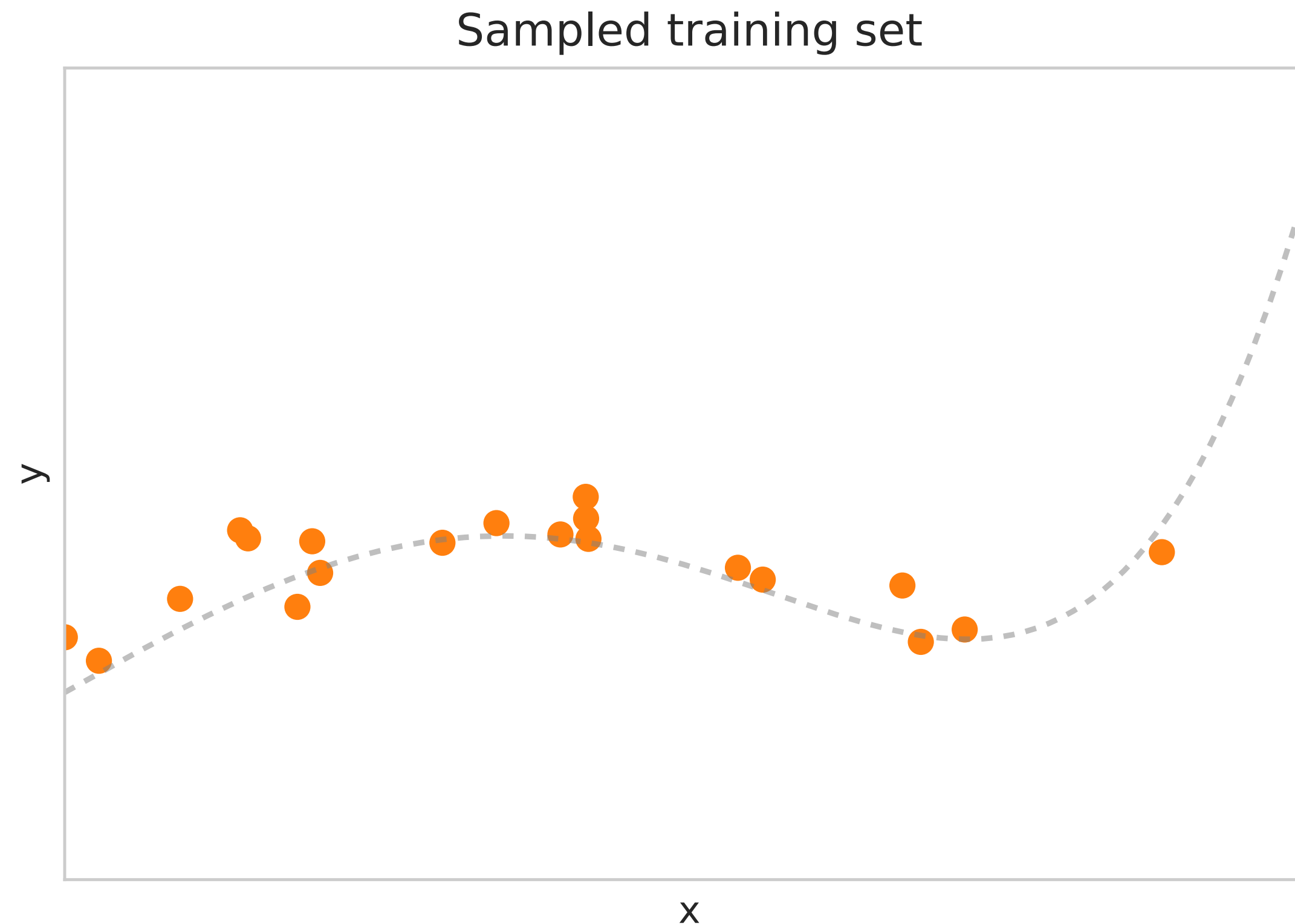
Before: quantitative

Now: ***qualitative***

# A small experiment: 1D-regression



# A small experiment: 1D-regression

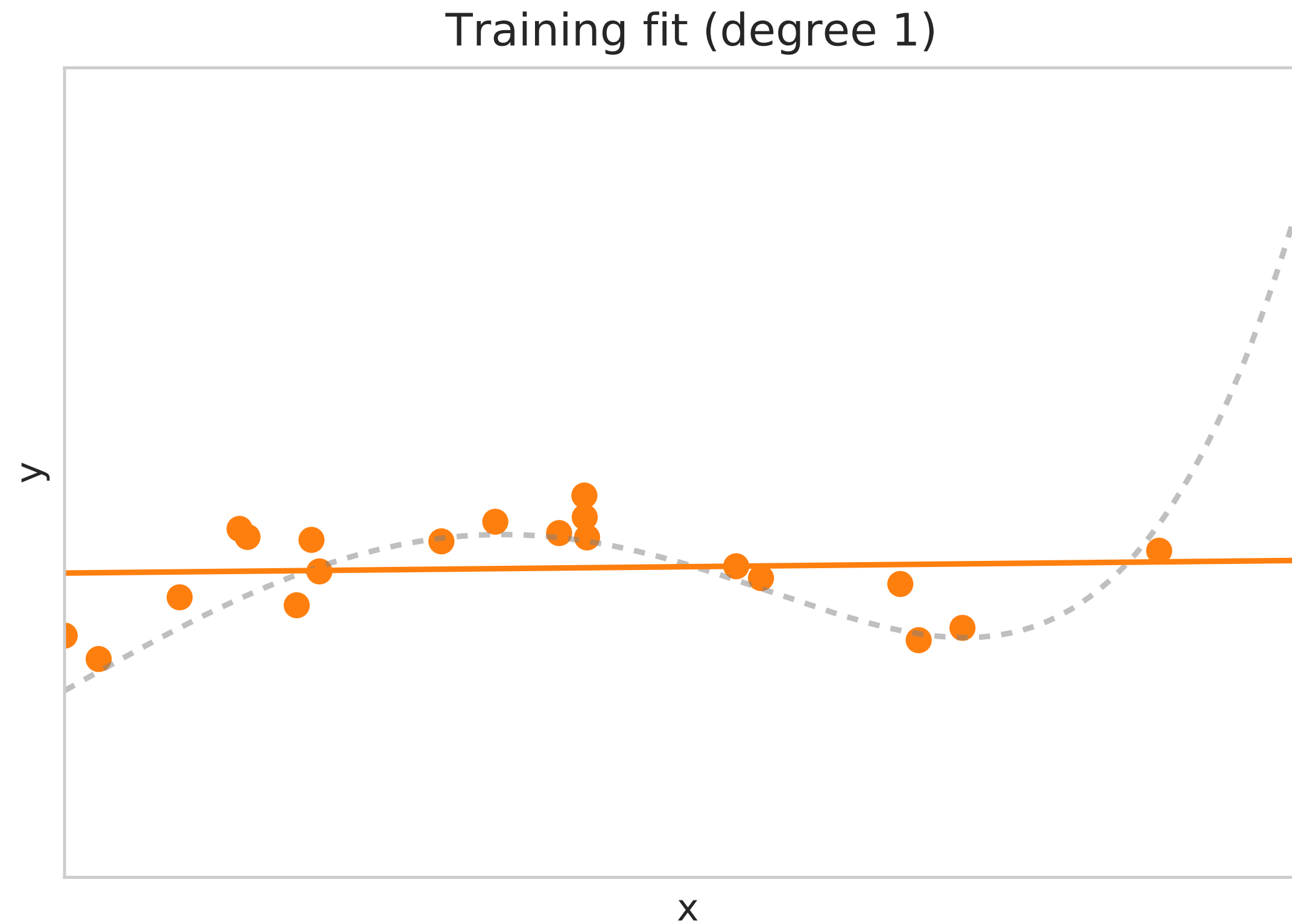


Linear regression using polynomial feature expansion  $(x, x^2, x^3, \dots, x^d)$

The maximum degree  $d$  measures the complexity of the class

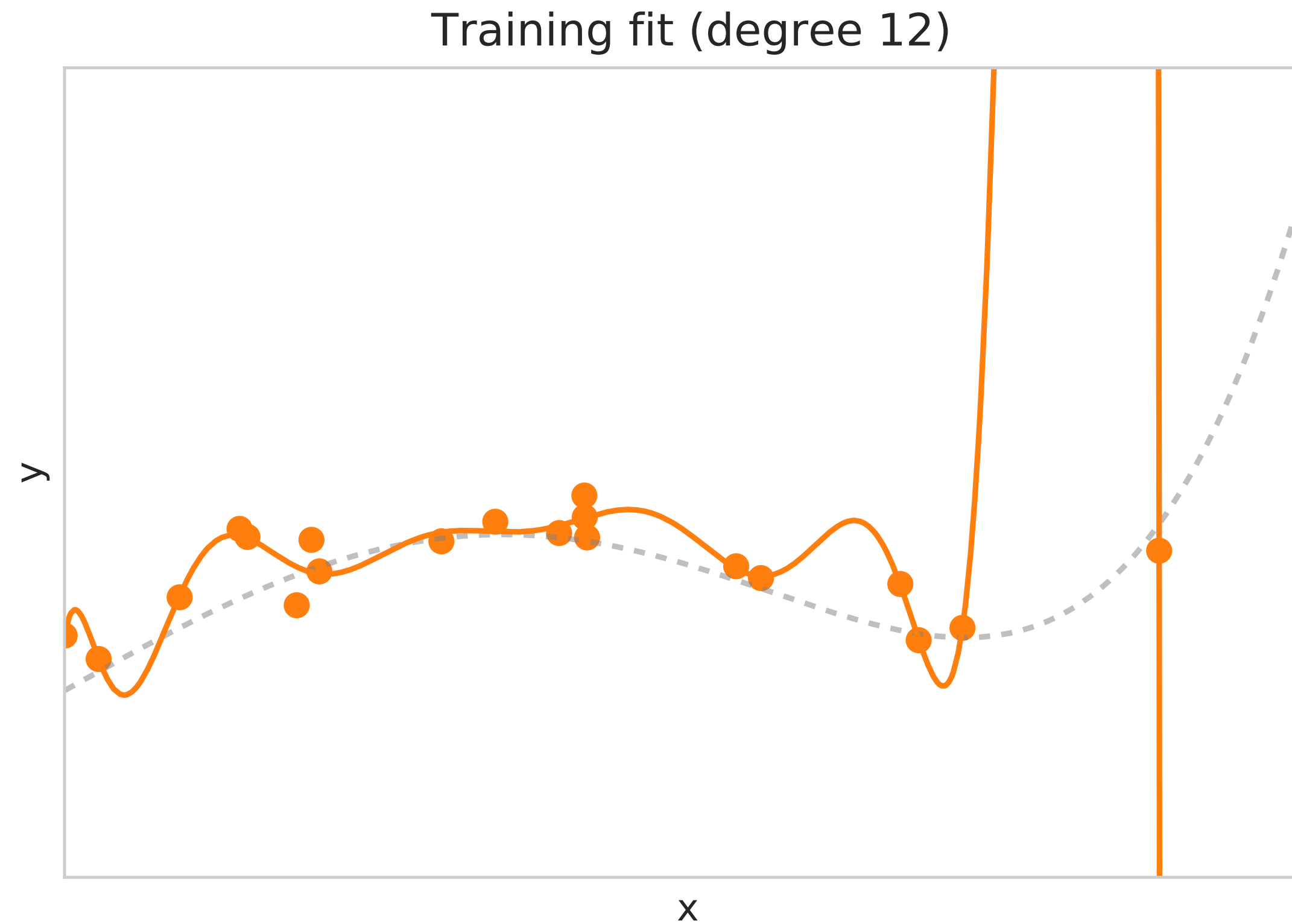
➡ How far should you go?

# Simple model: bad fit



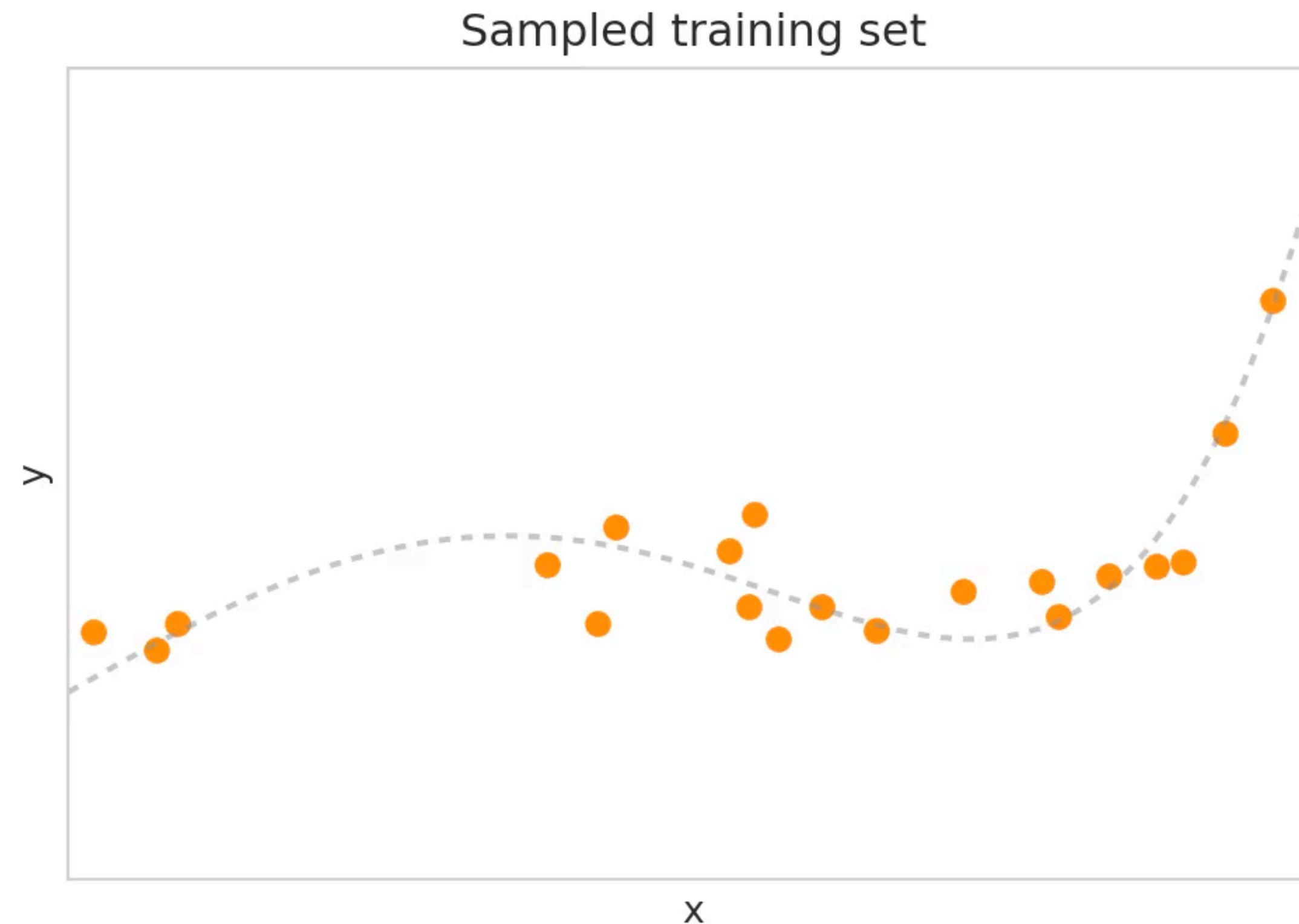
No linear function would be a good predictor. The model class is not rich enough

# Complex model: good fit?



High degree polynomial will be a good fit. But?

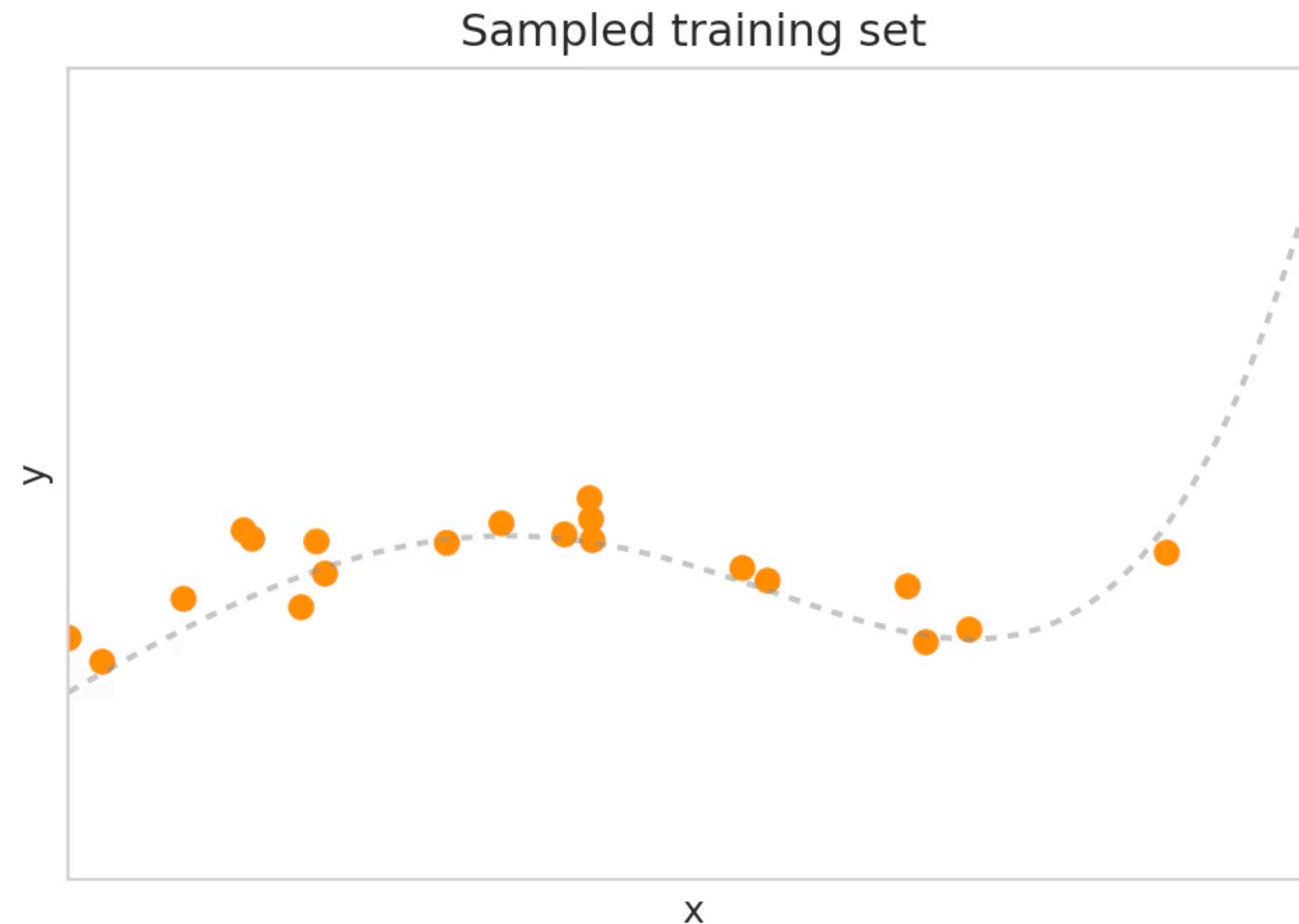
# But there is randomness in the data



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

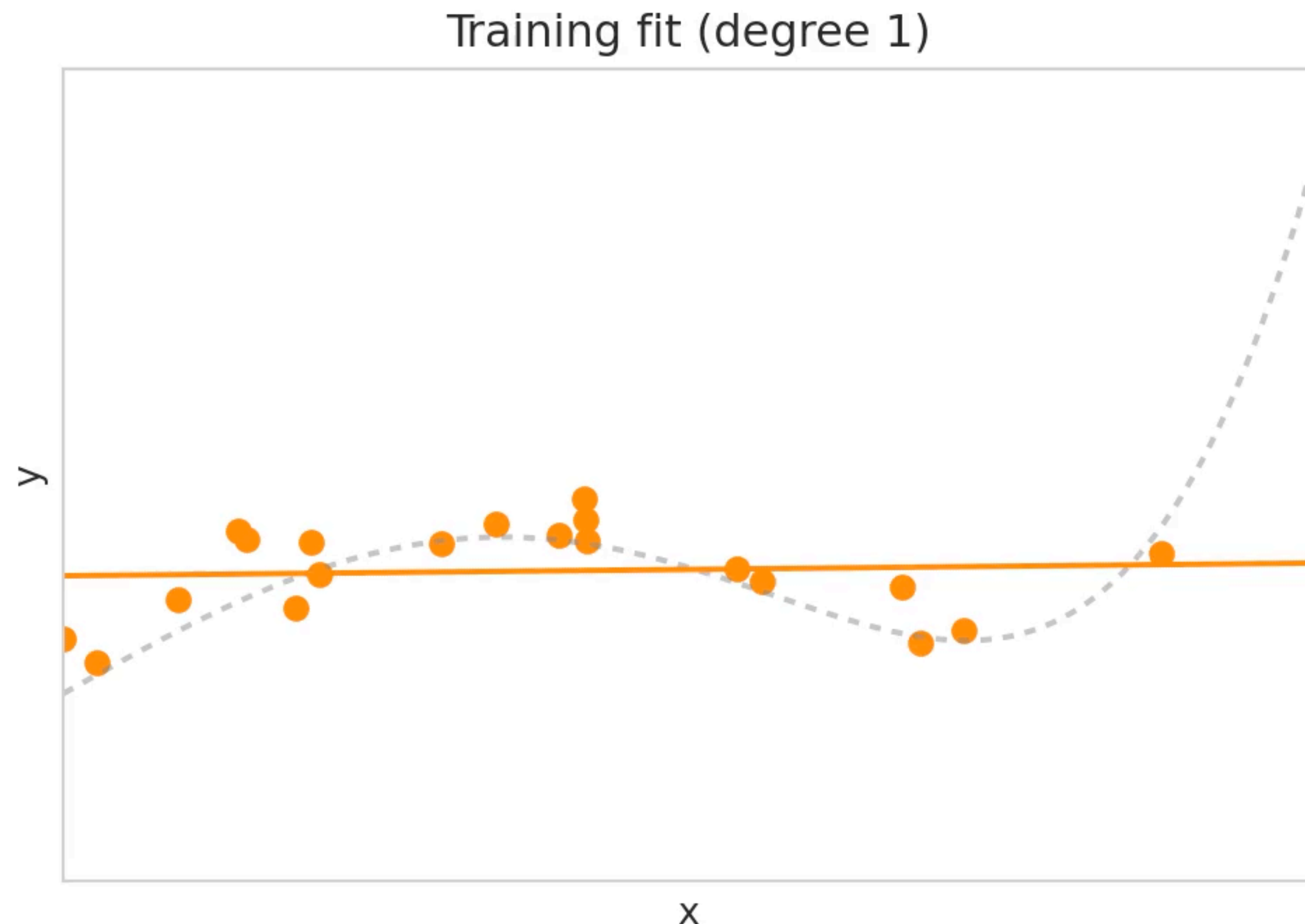


# But there is randomness in the data



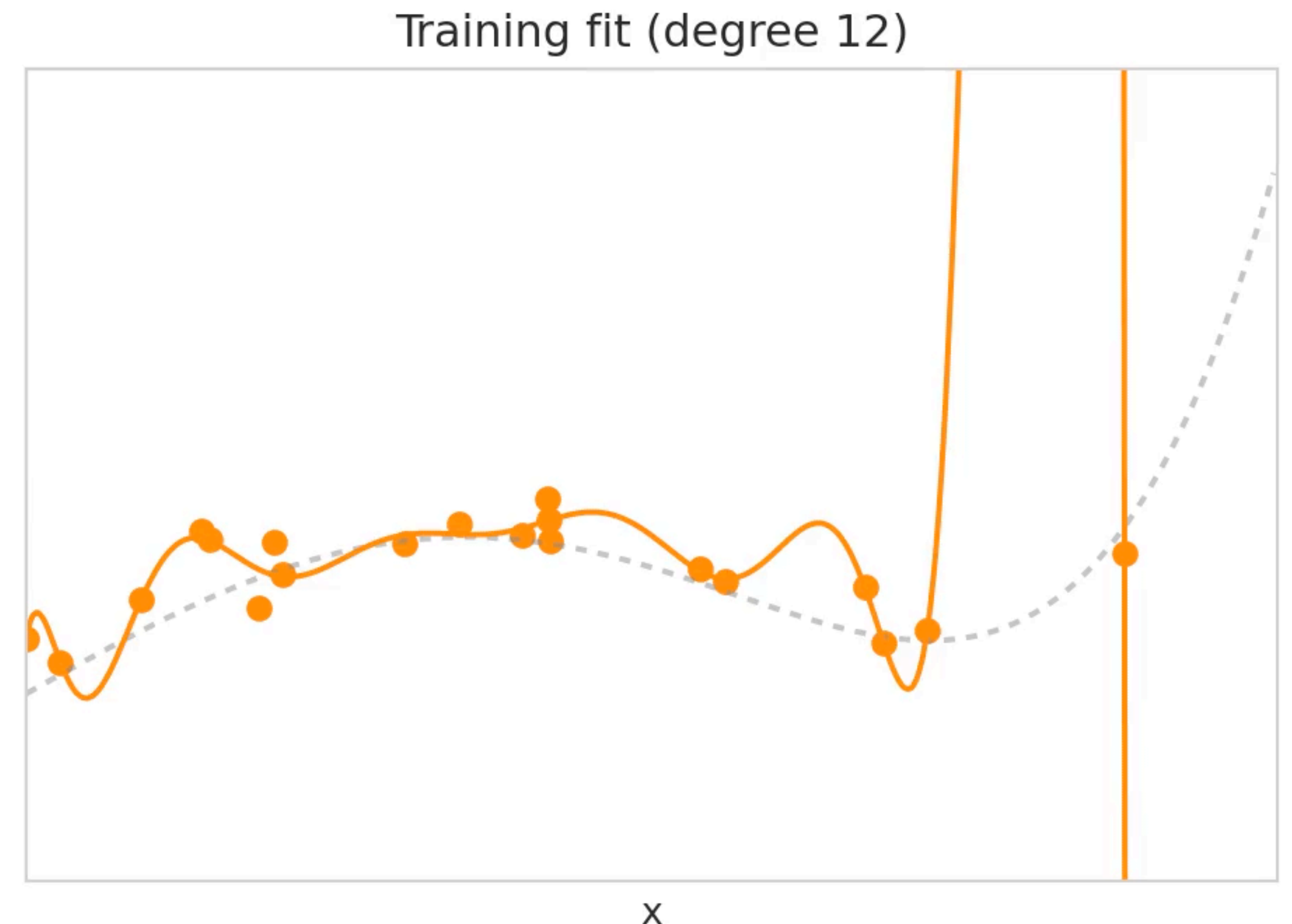
Even if we keep the same  $(x_1, \dots, x_n)$ , we have variability in the observed  $(y_1, \dots, y_n)$

# Thus there is randomness in the predictions



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

**Underfitting**

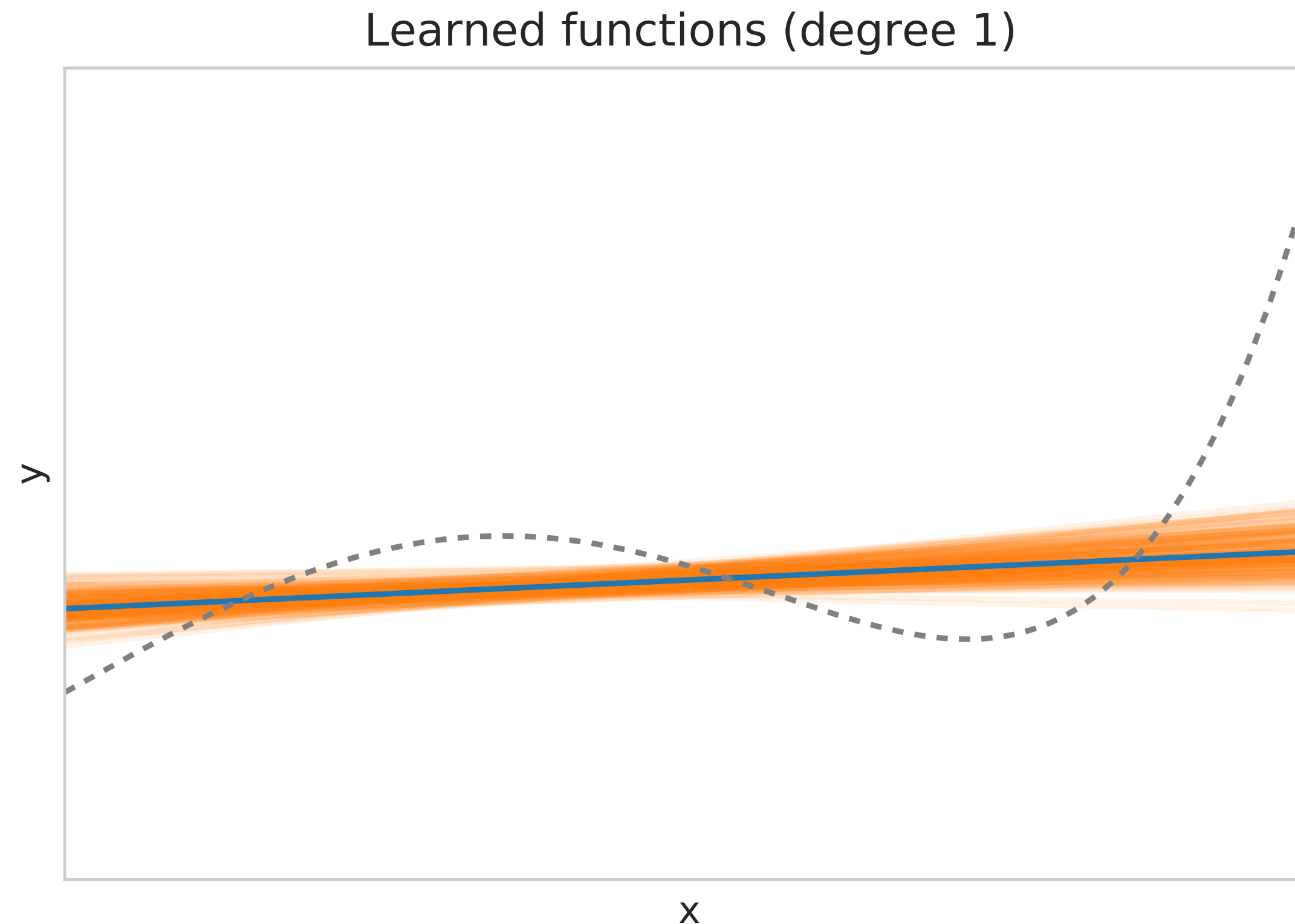


Changing one of the observations may change the prediction considerably

**Overfitting**

**Simple models are less sensitive**

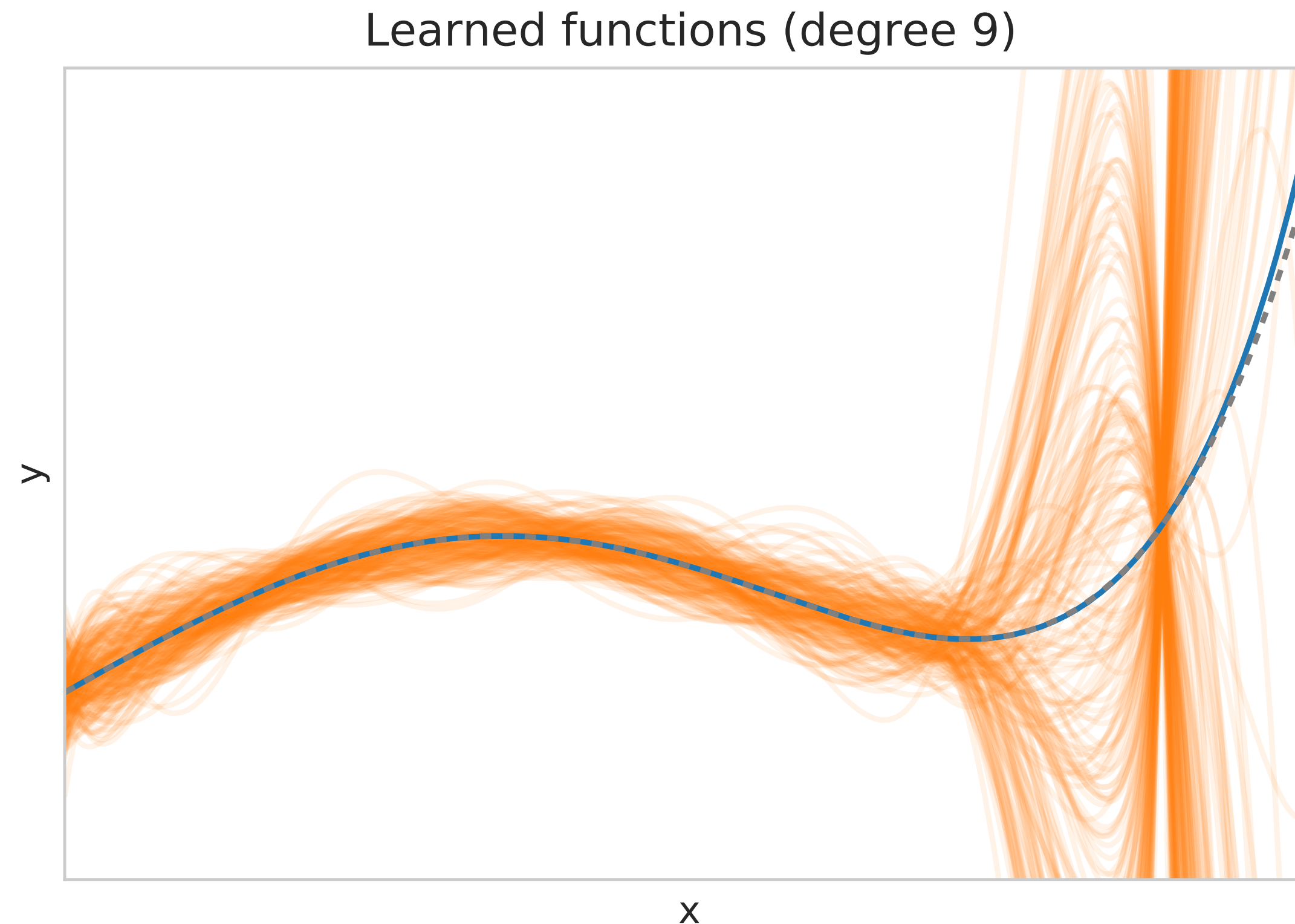
# Simple models have large bias but low variance



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**

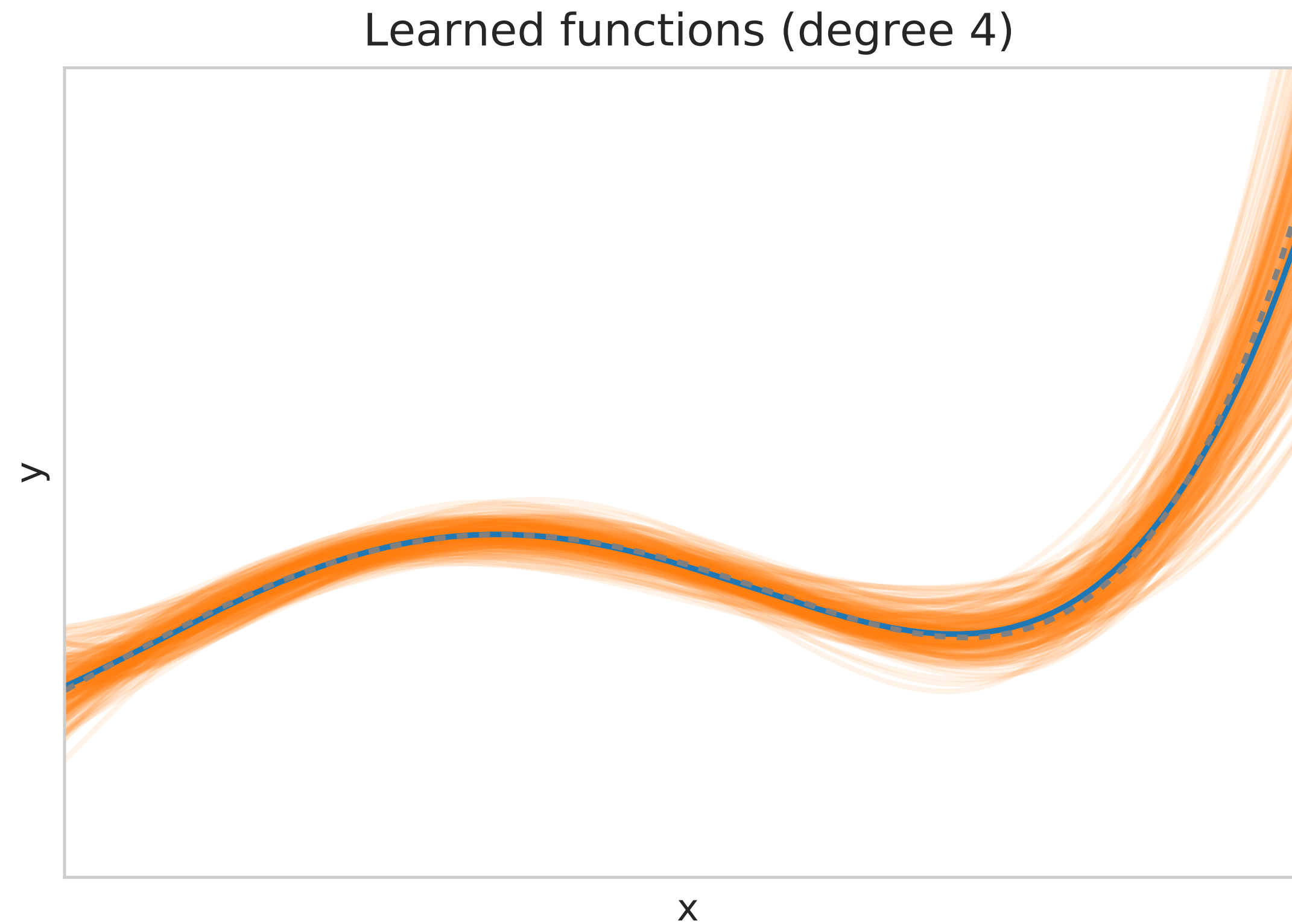
# Complex models have low bias but high variance



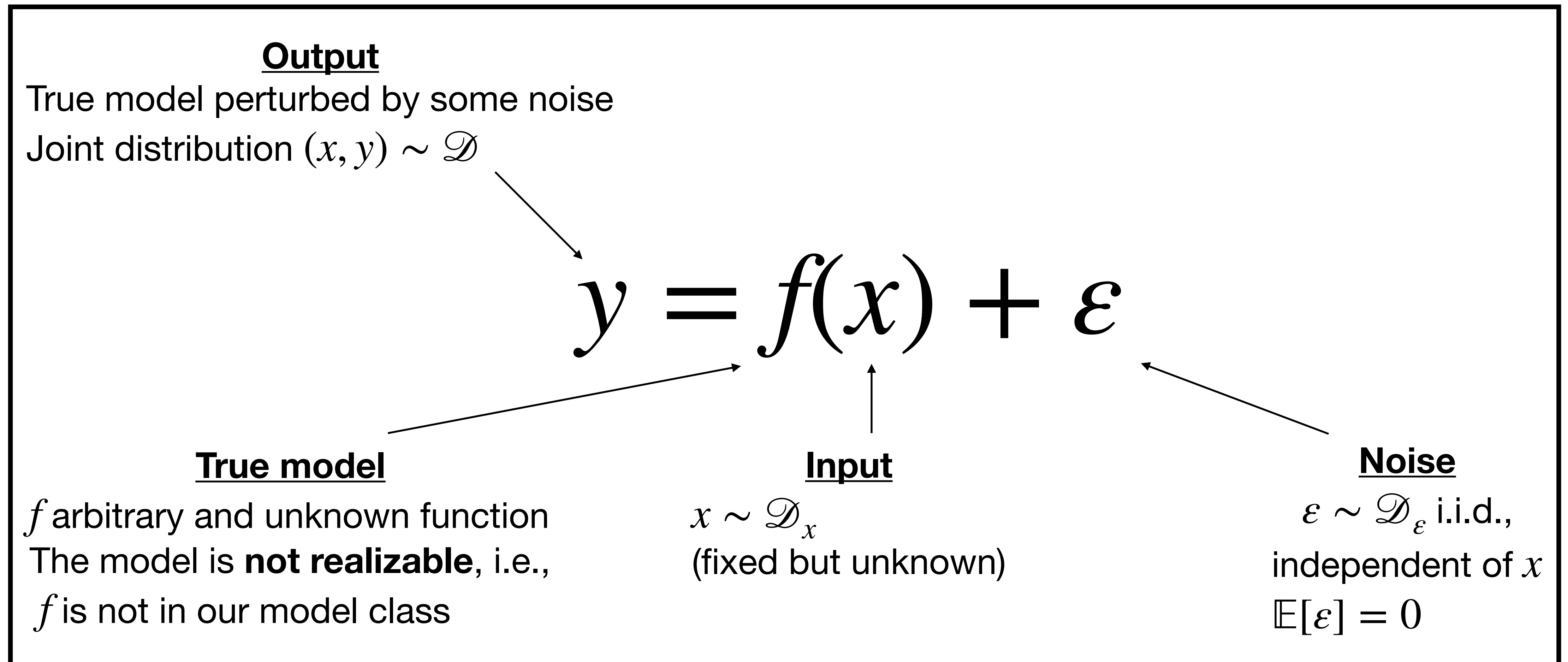
The average of the predictions  $f_S$  fits well the data: **small bias**

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

# We need to balance bias & variance correctly

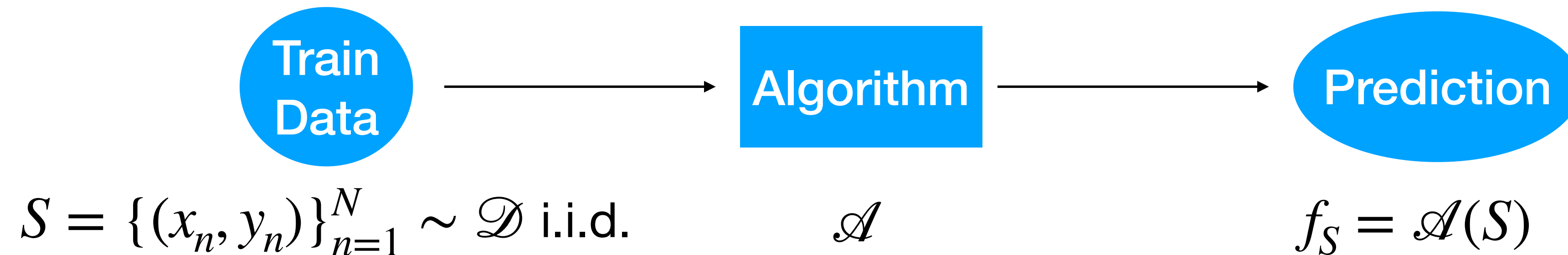


# Data model: output perturbed by some noise



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

# Error Decomposition



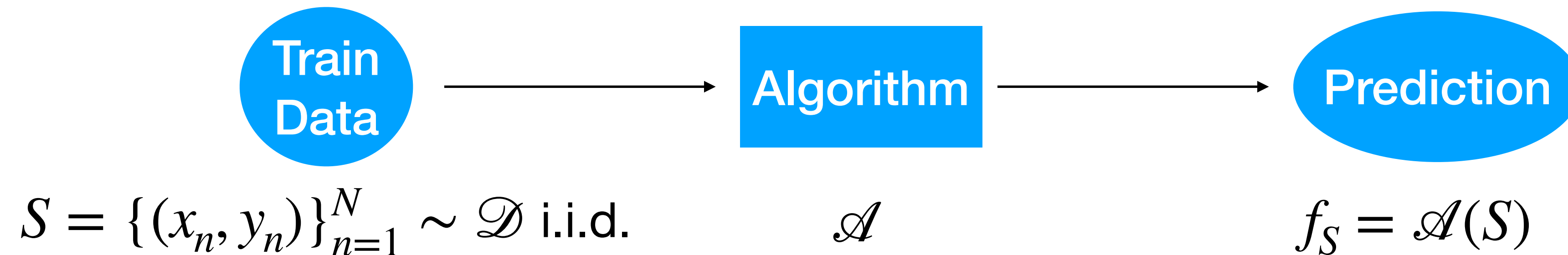
We are interested in how the **expected error** of  $f_S$ :

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

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



# Error Decomposition



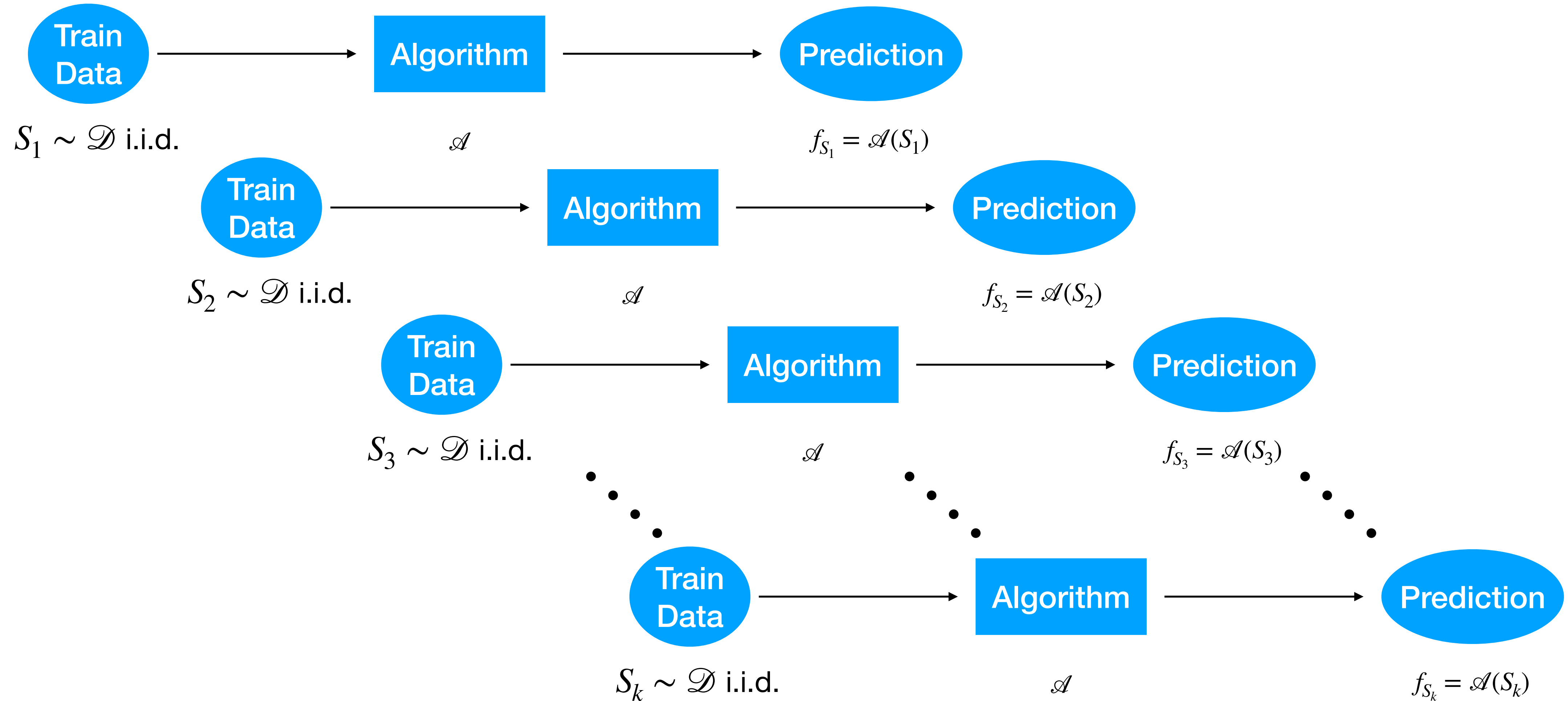
The decomposition will hold true at ***every single point***  $x$ . Therefore, to simplify, we consider the expected error of  $f_S$  for a fixed element  $x_0$ :

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

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



# We run the experiment many times



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

# A decomposition in three terms

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

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

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

First we expand the square:

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

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

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

Therefore

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

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  $S$

$$\begin{aligned}\mathbb{E}_{S \sim \mathcal{D}}[(f(x_0) - f_S(x_0))^2] &= \mathbb{E}_{S \sim \mathcal{D}}[(f(x_0) - \mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)] + \mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)] - f_S(x_0))^2] \\ &= \mathbb{E}_{S \sim \mathcal{D}}\left[(f(x_0) - \mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)])^2 + (\mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)] - f_S(x_0))^2\right. \\ &\quad \left.+ 2(f(x_0) - \mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)])(\mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)] - f_S(x_0))\right]\end{aligned}$$

Cross-term:

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

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

# Bias-Variance Decomposition

We obtain the following decomposition into three positive terms:

$$\begin{aligned} \mathbb{E}_{S \sim \mathcal{D}, \varepsilon \sim \mathcal{D}_\varepsilon}[(f(x_0) + \varepsilon - f_S(x_0))^2] &= \text{Var}_{\varepsilon \sim \mathcal{D}_\varepsilon}[\varepsilon] \longleftarrow \text{Noise variance} \\ &\quad \text{Bias} \longrightarrow + (f(x_0) - \mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)])^2 \\ &\quad \text{Variance} \longrightarrow + \mathbb{E}_{S \sim \mathcal{D}}[(f_S(x_0) - \mathbb{E}_{S' \sim \mathcal{D}}[f_{S'}(x_0)])^2] \end{aligned}$$

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

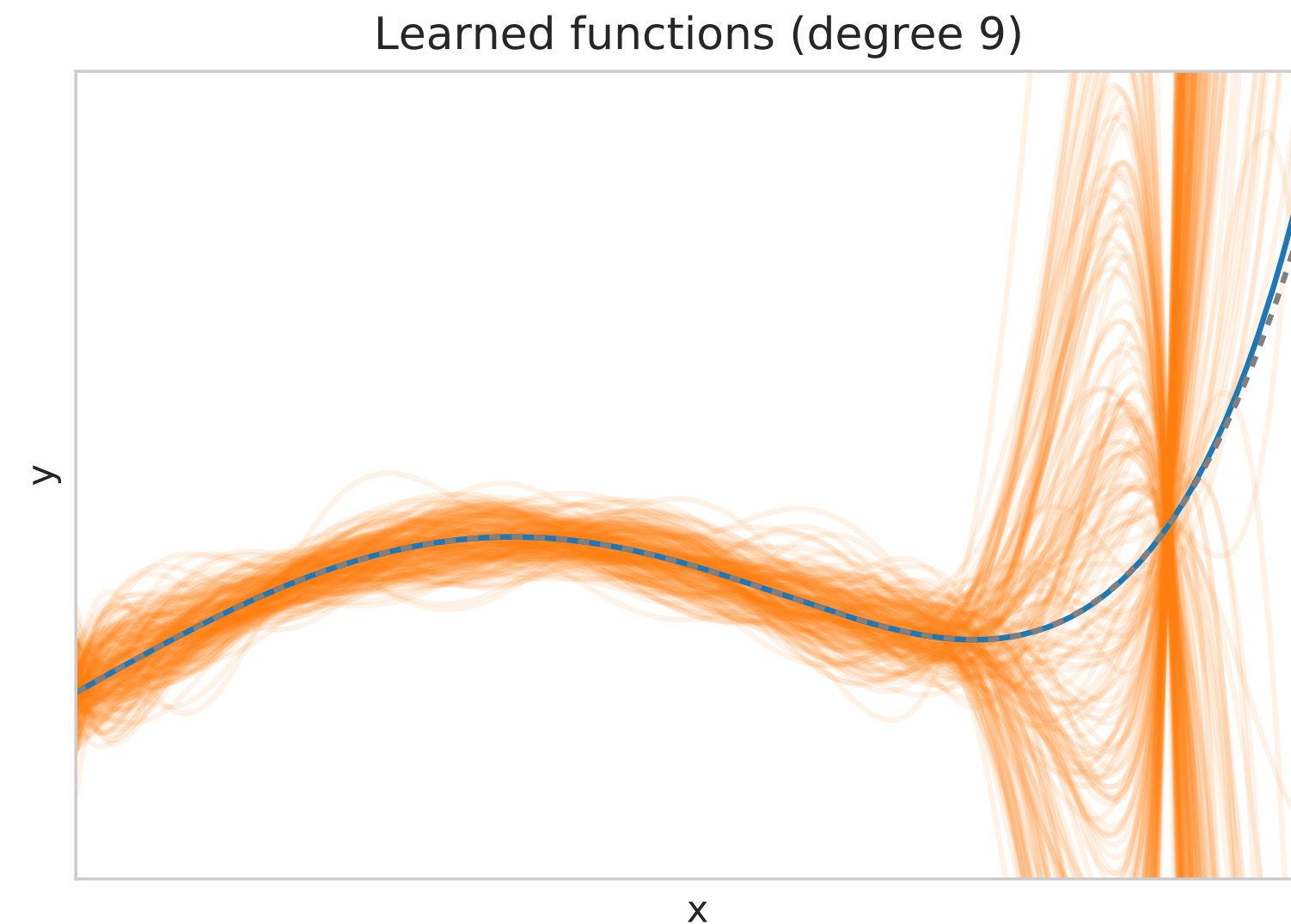
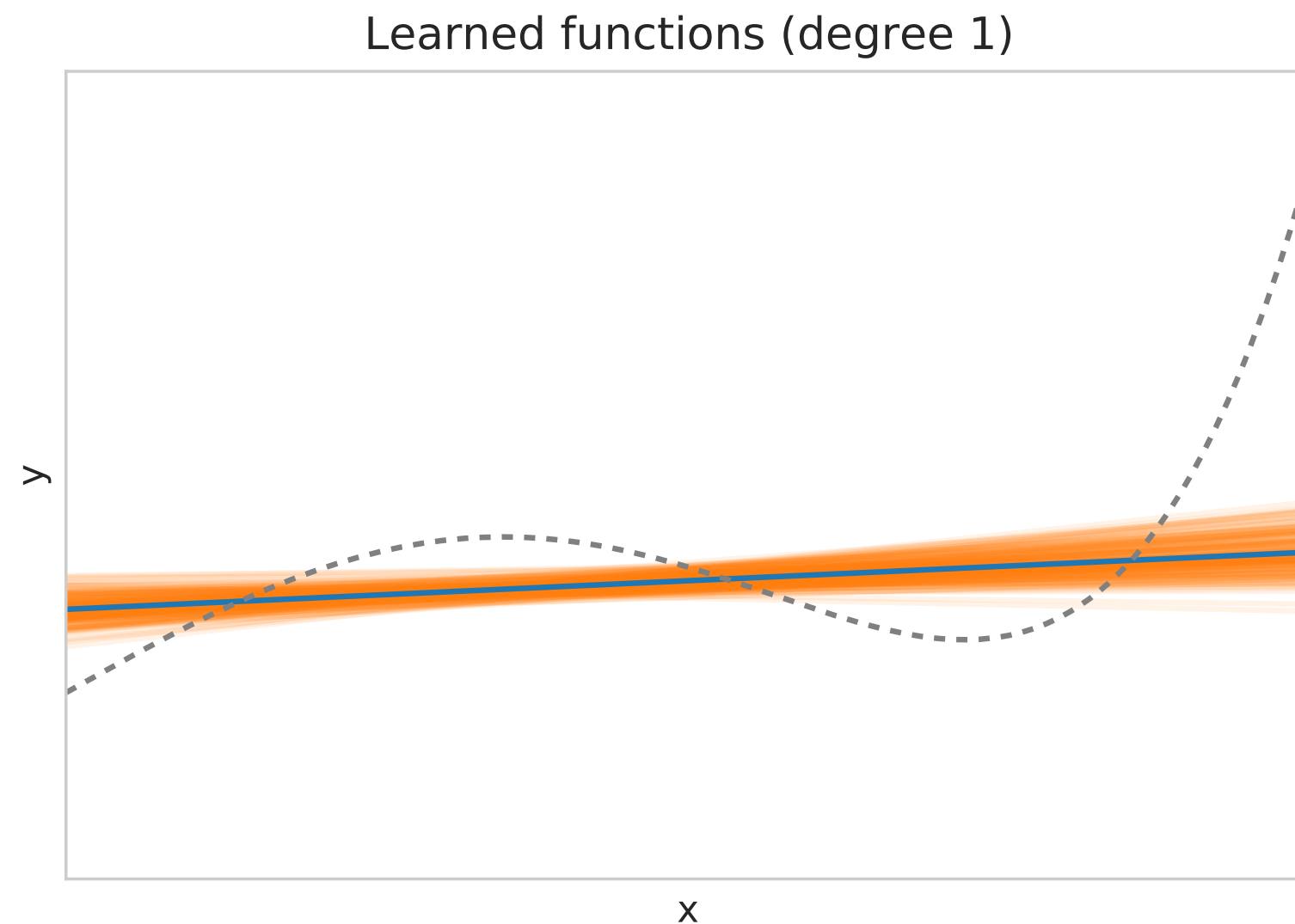
➡ To minimize the true error, we must choose a method that achieves **low bias and low variance** simultaneously

# Noise: a strict lower bound on the achievable error



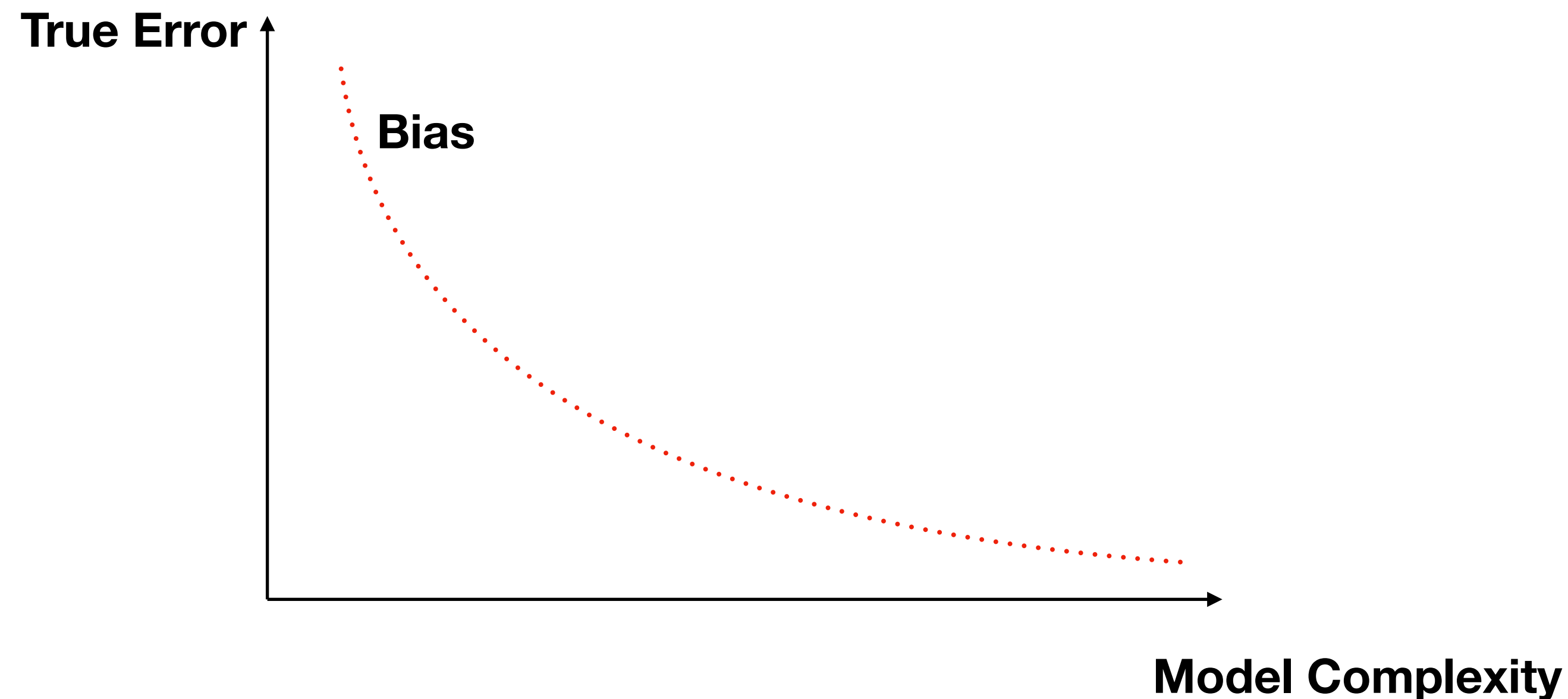
- It is not possible to go below the noise level
- Even if we know the true model  $f$ , we still suffer from the noise:  $L(f) = \mathbb{E}[\varepsilon^2]$
- It is not possible to predict the noise from the data since they are independent

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



- 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 model **complexity** is **low**, **bias** is typically **high**
- If model **complexity** is **high**, **bias** is typically **low**

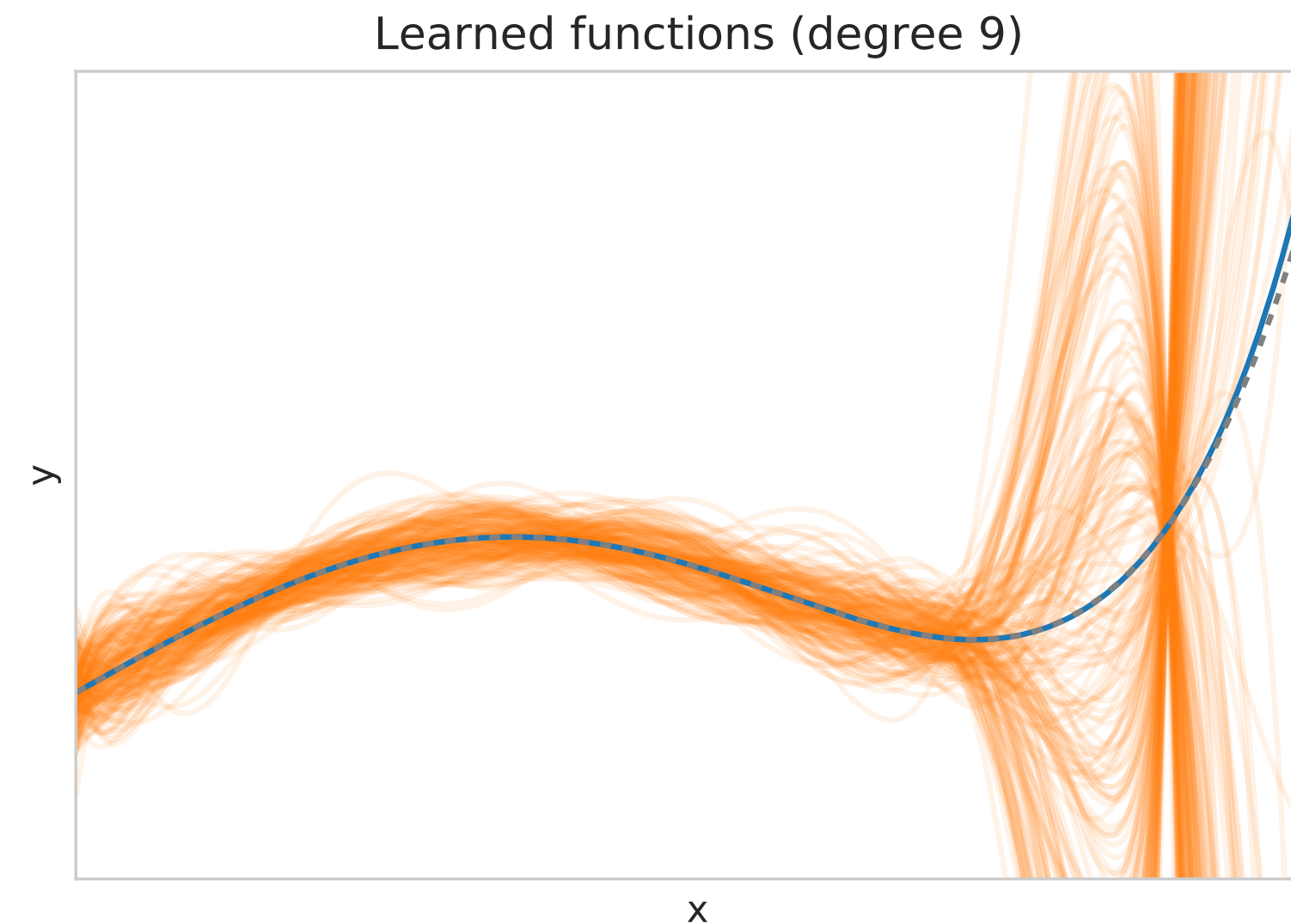
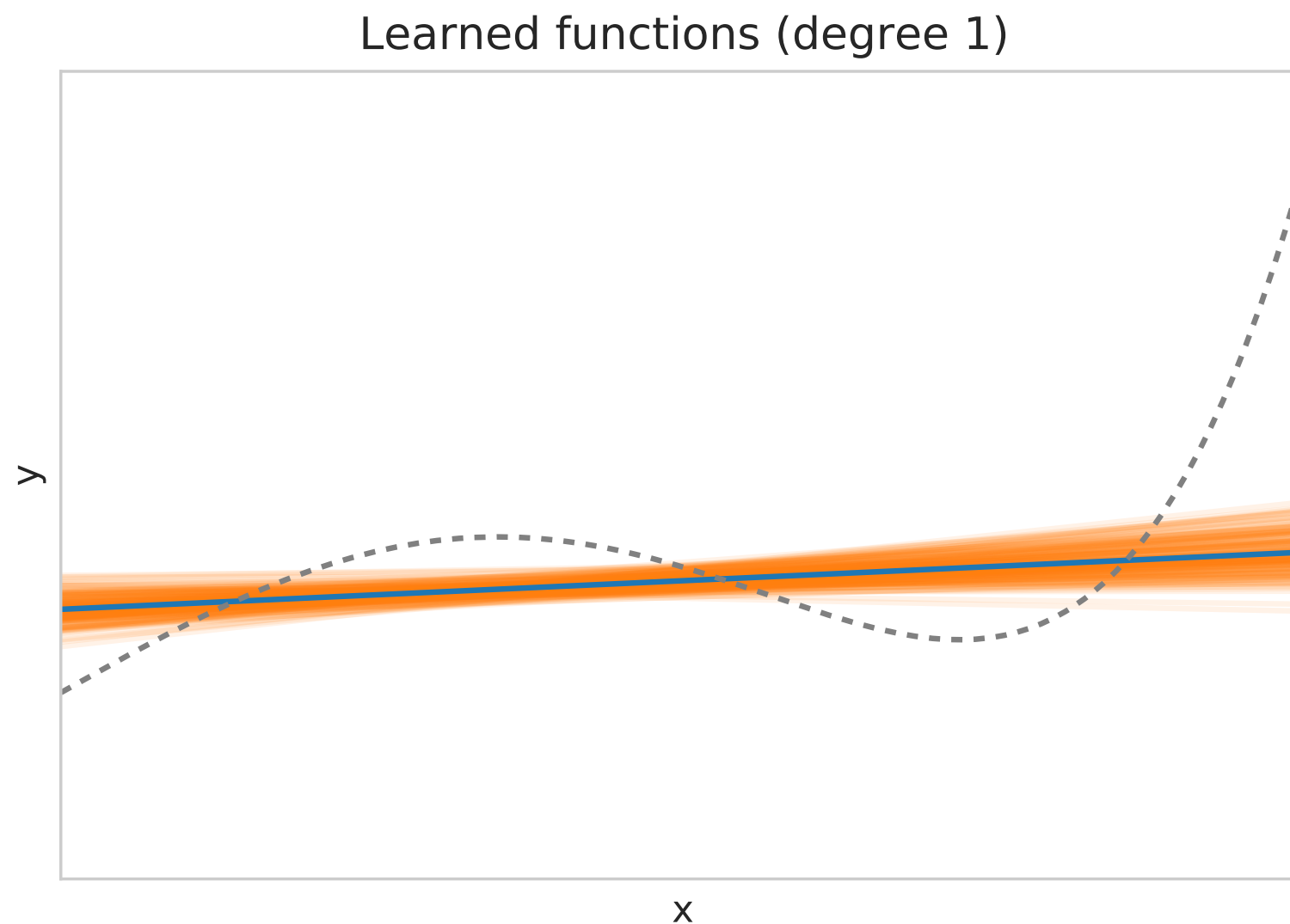
# Bias: $(f(x_0) - \mathbb{E}_{S \sim \mathcal{D}}[f_S(x_0)])^2$



- 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 model **complexity** is **low**, **bias** is typically **high**
- If model **complexity** is **high**, **bias** is typically **low**

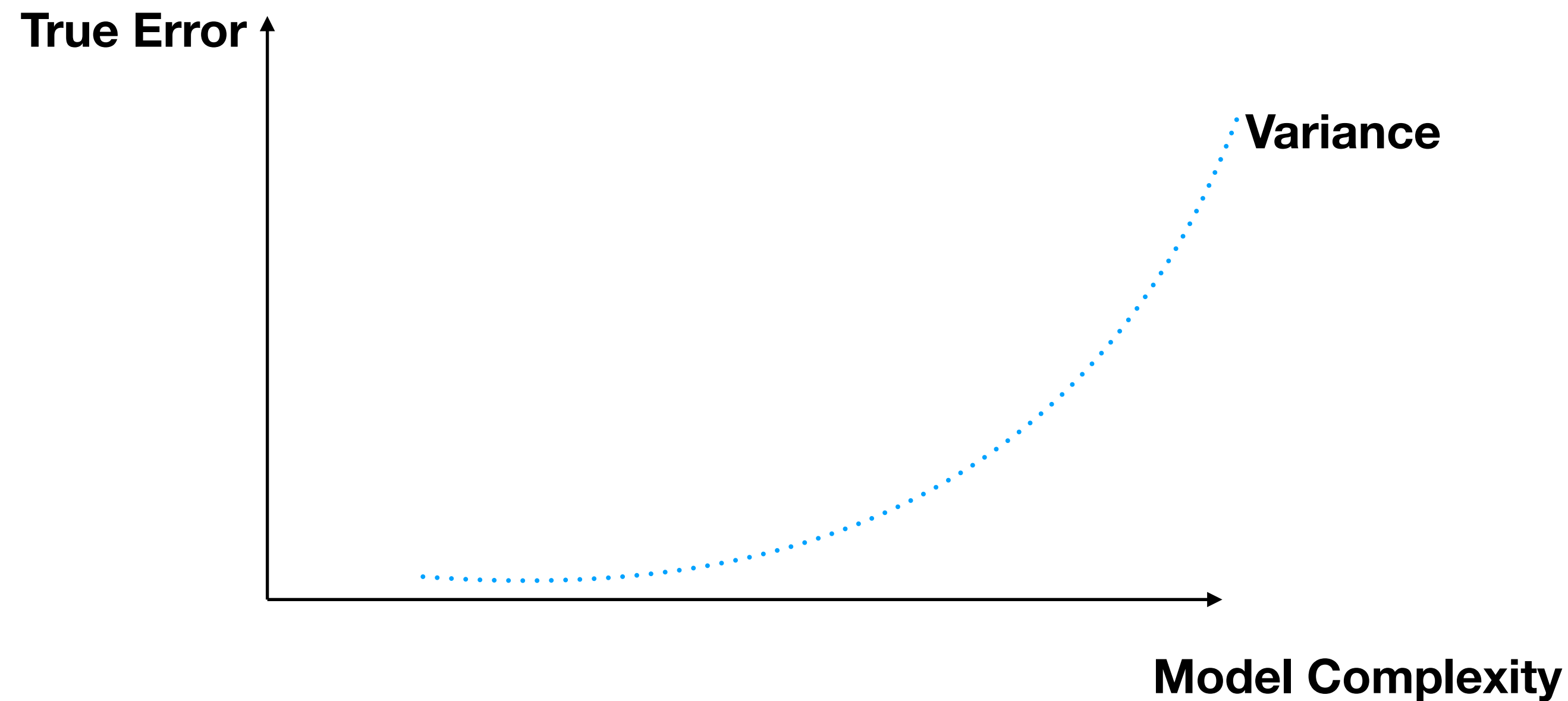


# Variance: $\mathbb{E}_{s \sim \mathcal{D}} [(f_s(x_0) - \mathbb{E}_{s \sim \mathcal{D}}[f_s(x_0)])^2]$



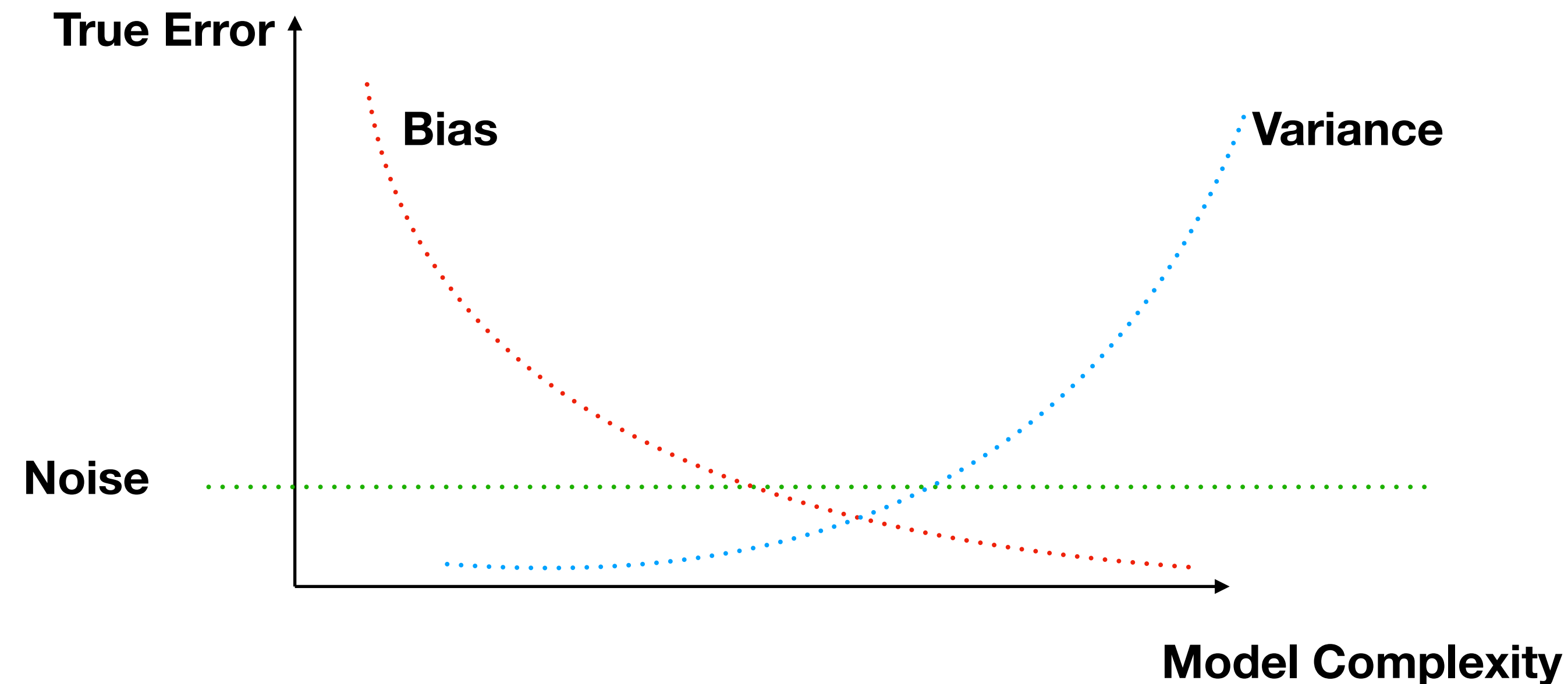
- Variance of the prediction function
- It measures the variability of predictions at a given point across different training set realizations
- If we consider complex models, small variations in the training set can lead to significant changes in the predictions

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



- 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

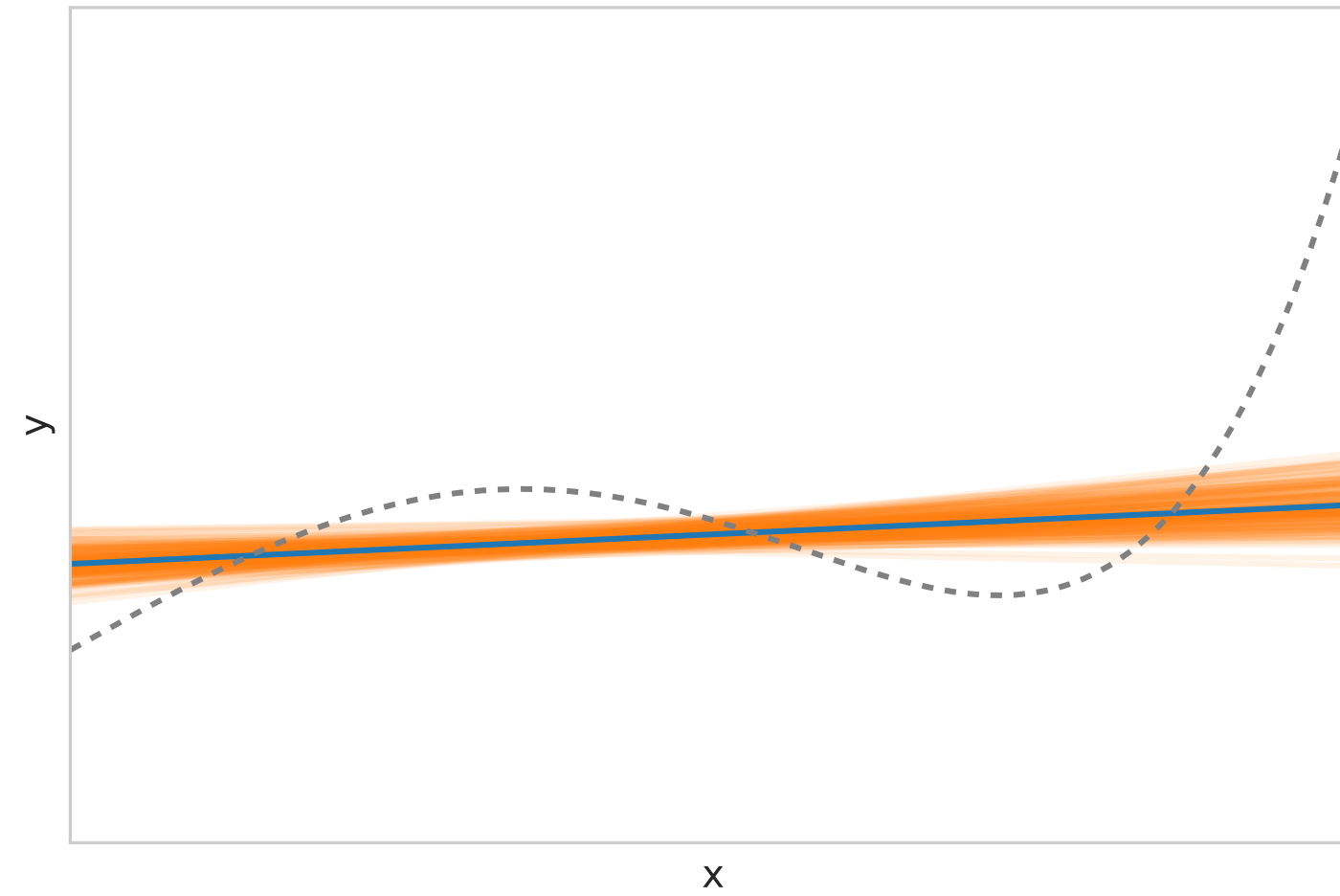
# Bias Variance tradeoff and U-shape curve



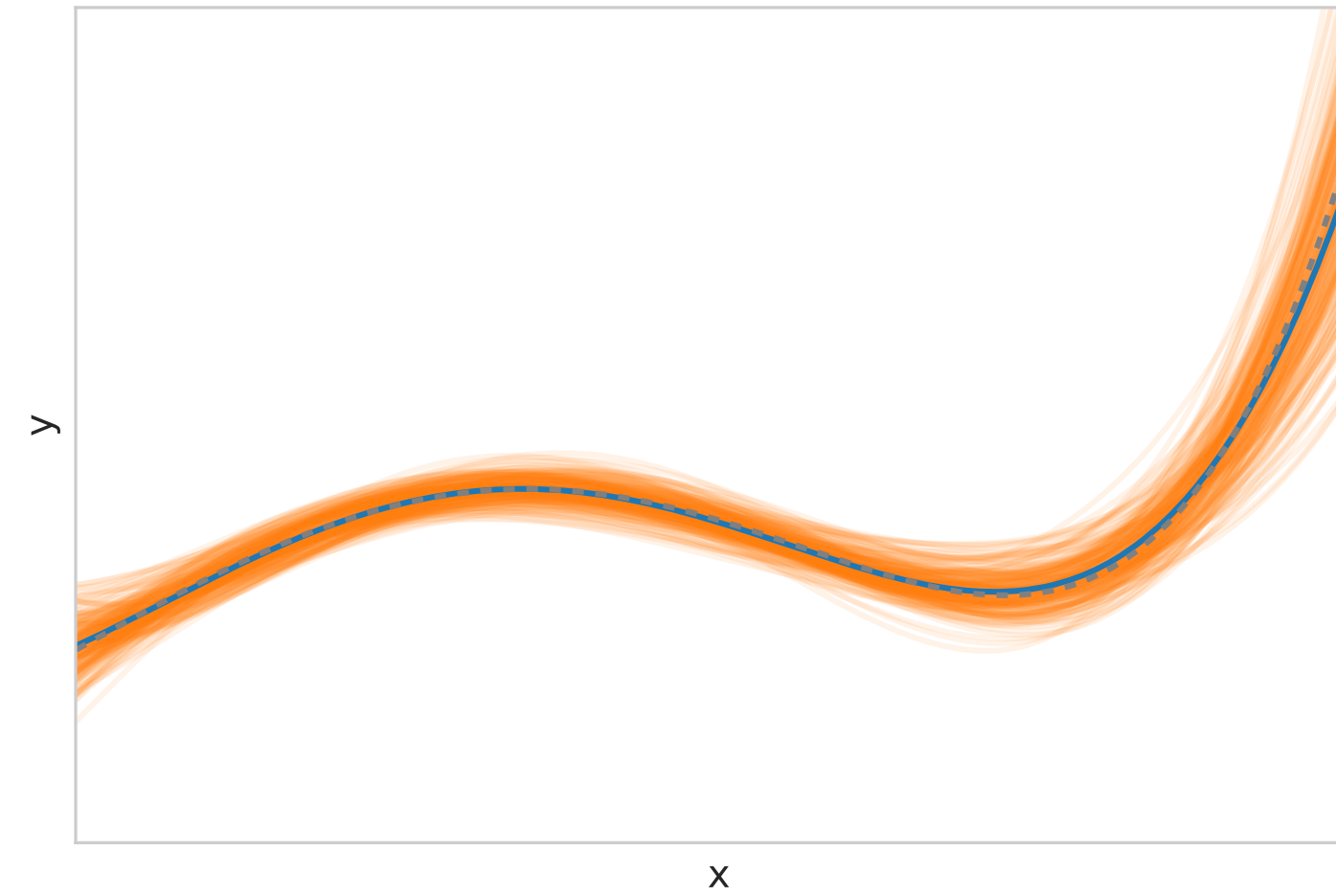
- If model complexity is too low, approximation will be poor (underfitting)
  - If model complexity is too high, it may cause issues with variance (overfitting)
- ➡ This phenomenon is known as the bias-variance tradeoff

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

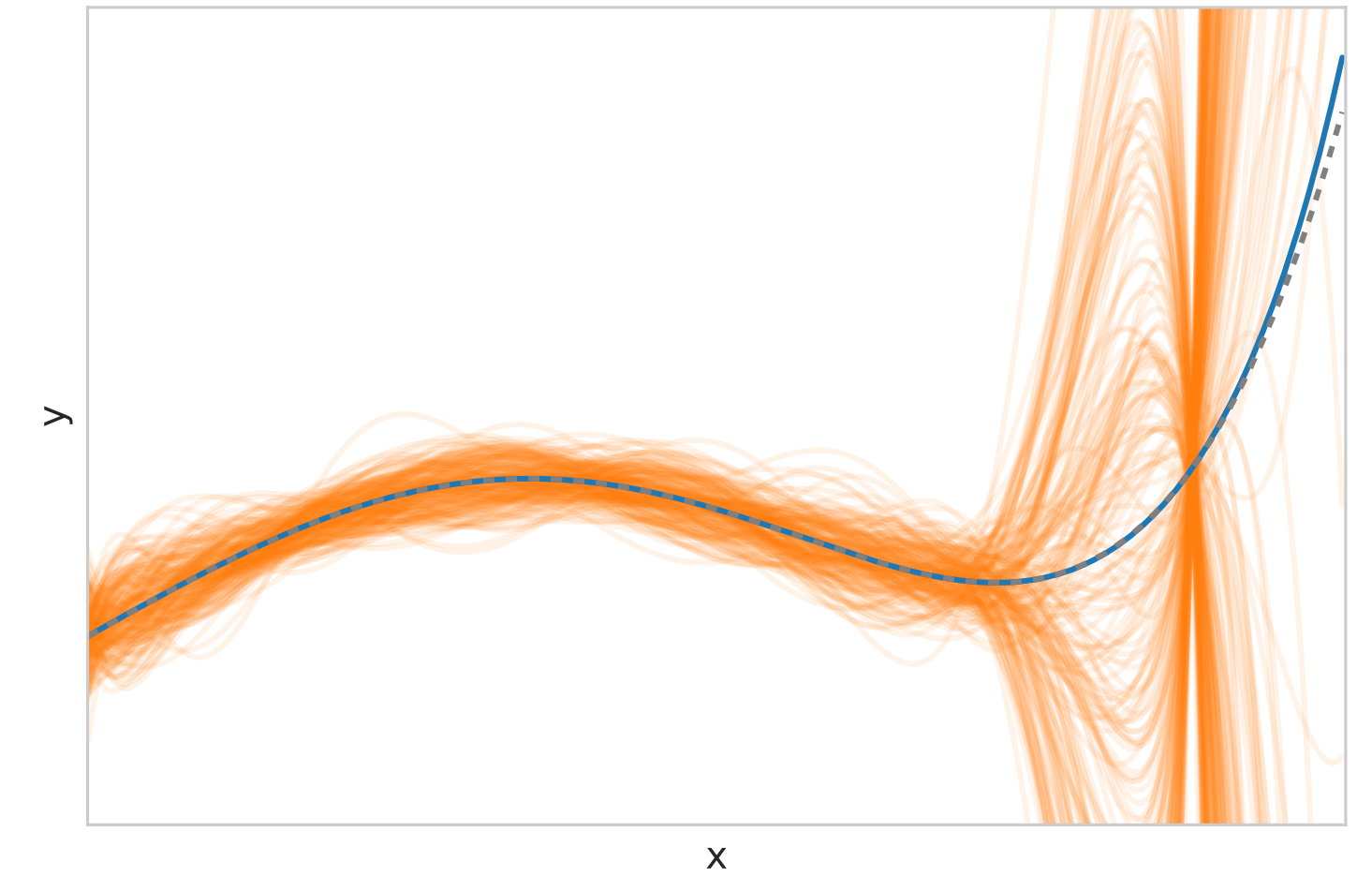
Learned functions (degree 1)



Learned functions (degree 4)



Learned functions (degree 9)

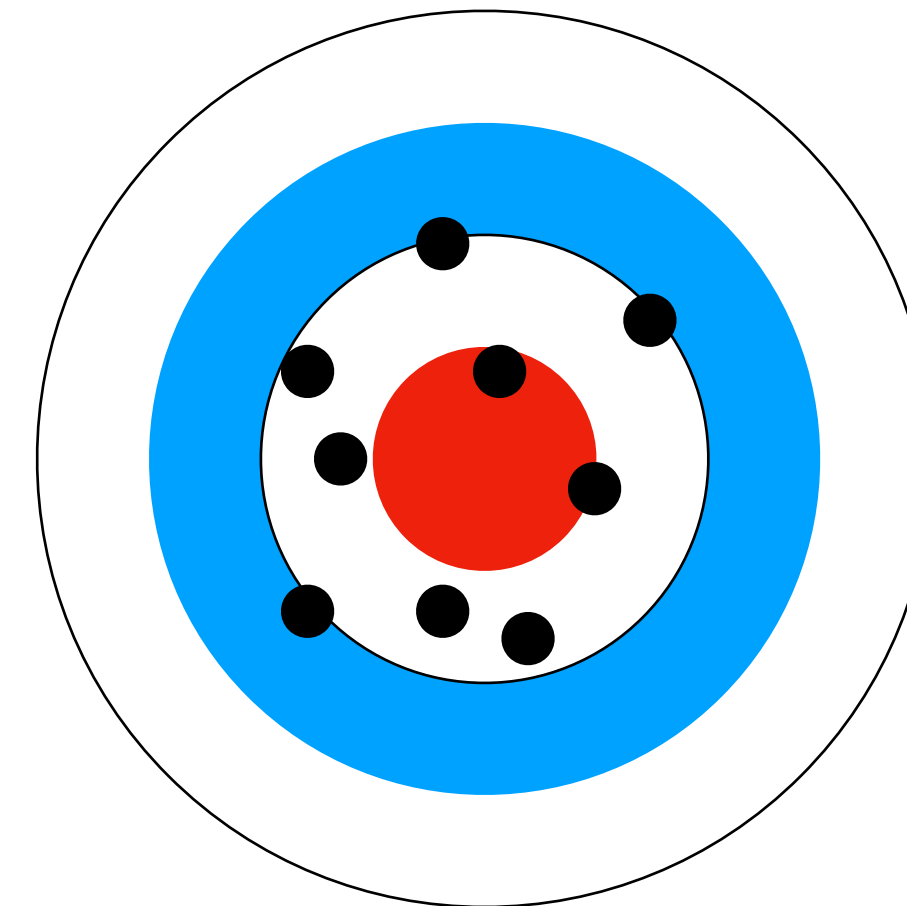
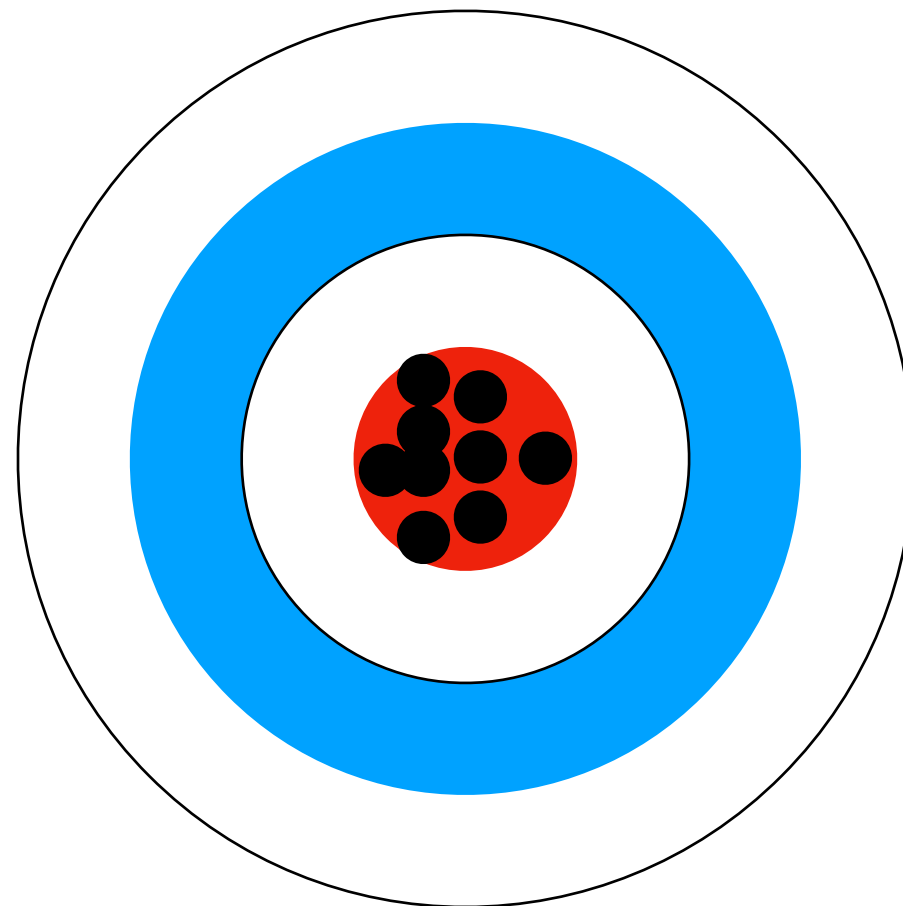


# Conclusion

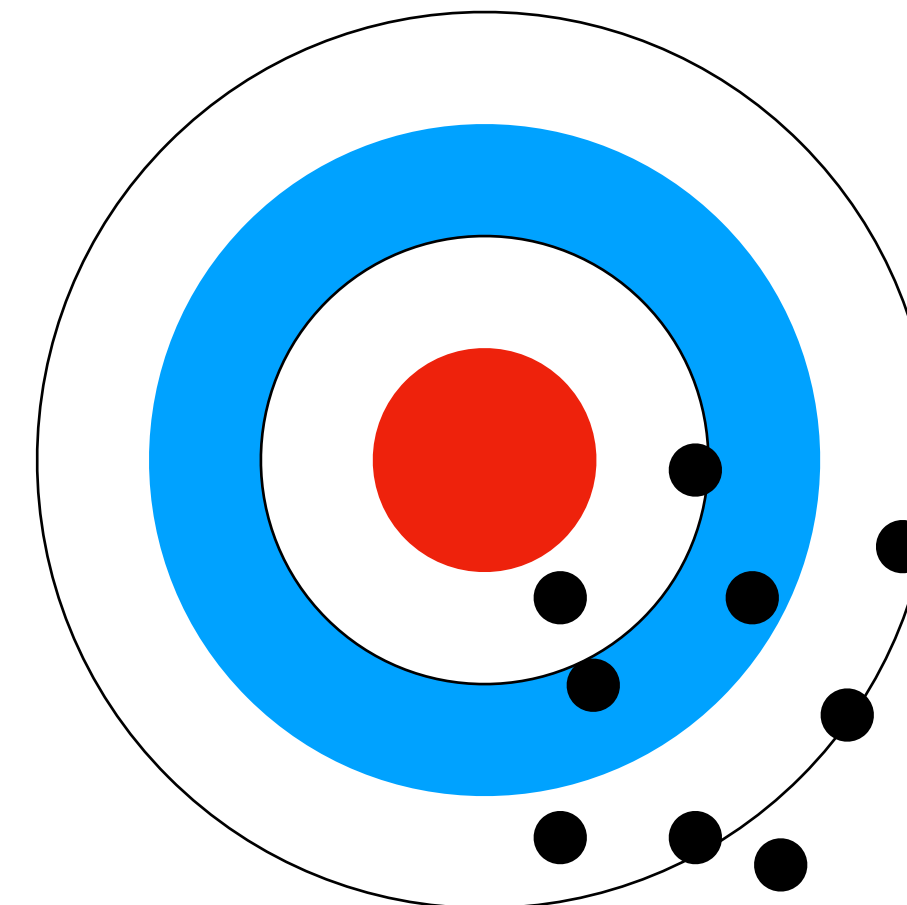
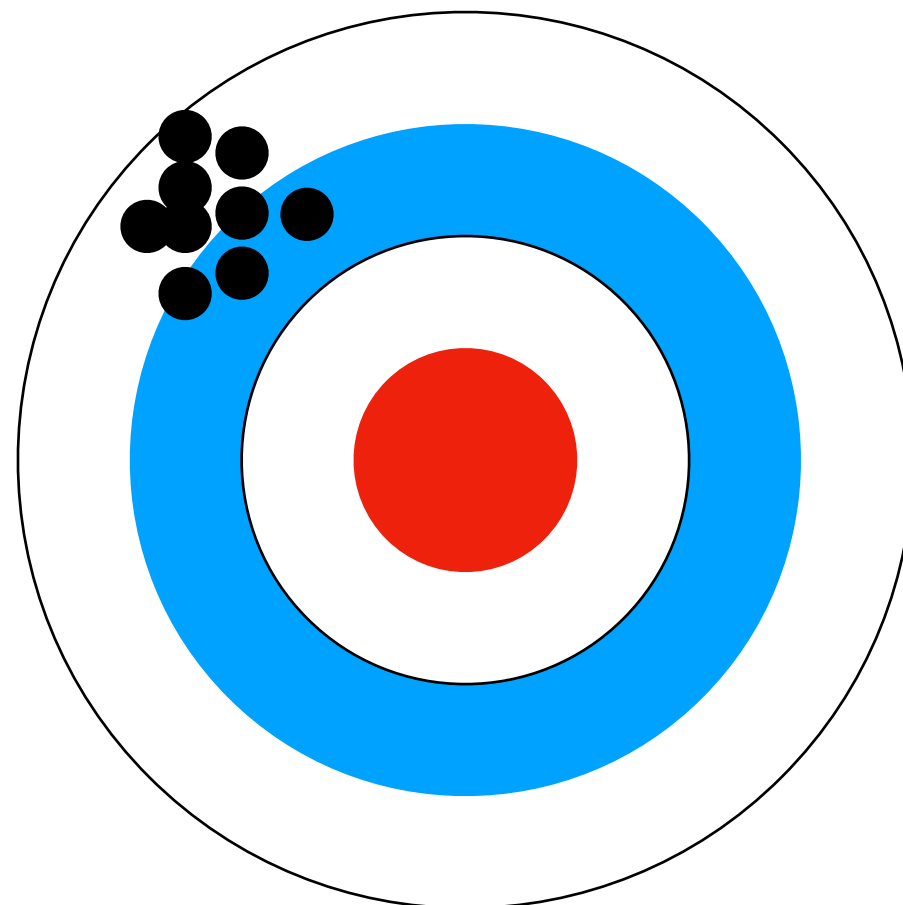
**Low Variance**

**High Variance**

**Low Bias**



**High Bias**



**But this depends on the  
algorithm!**

# Double descent curve

