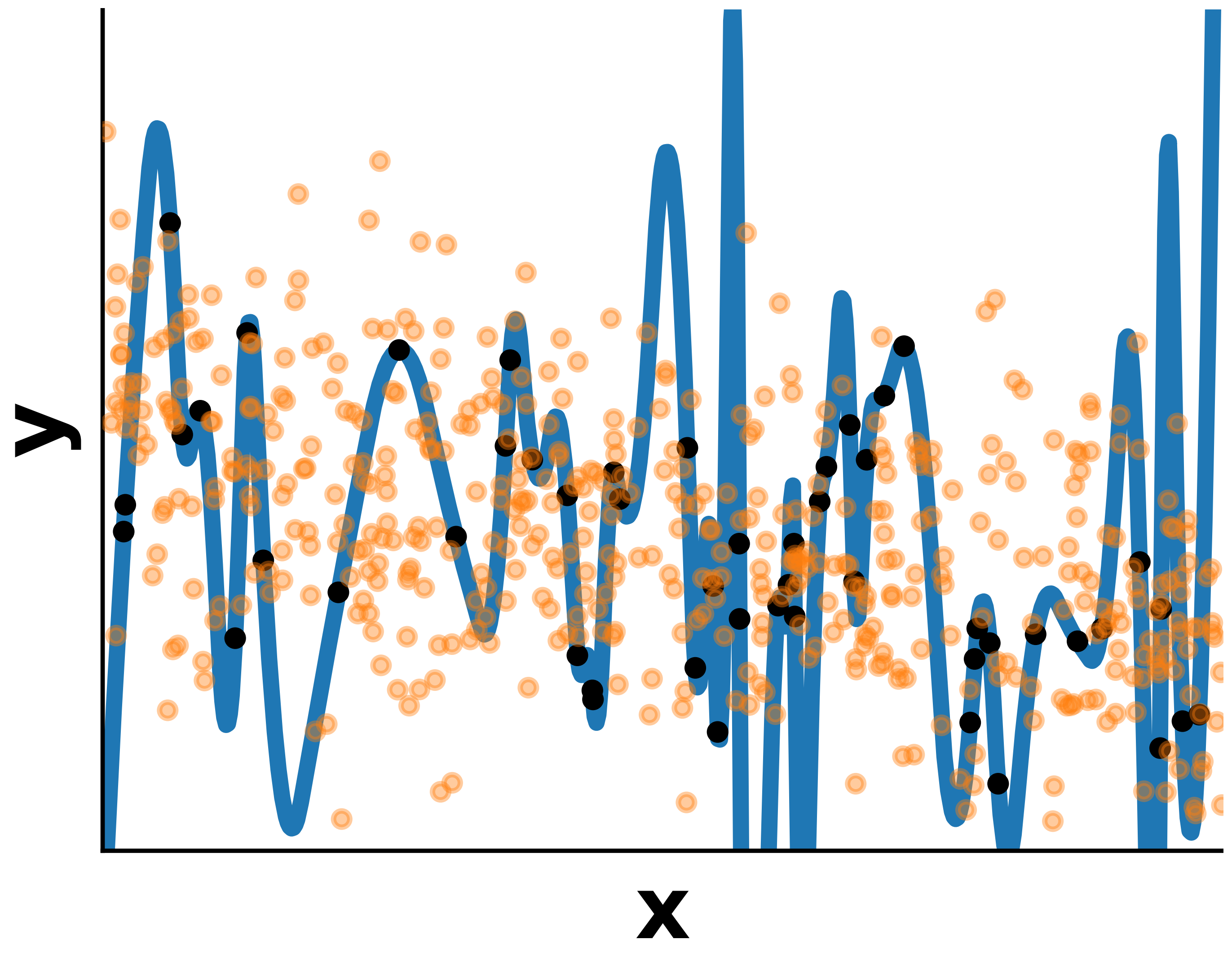
Comparing train and test errors

Varying complexity: validation curves

Varying the sample size: learning curves

Goal: understand the overfitting/underfitting trade-off

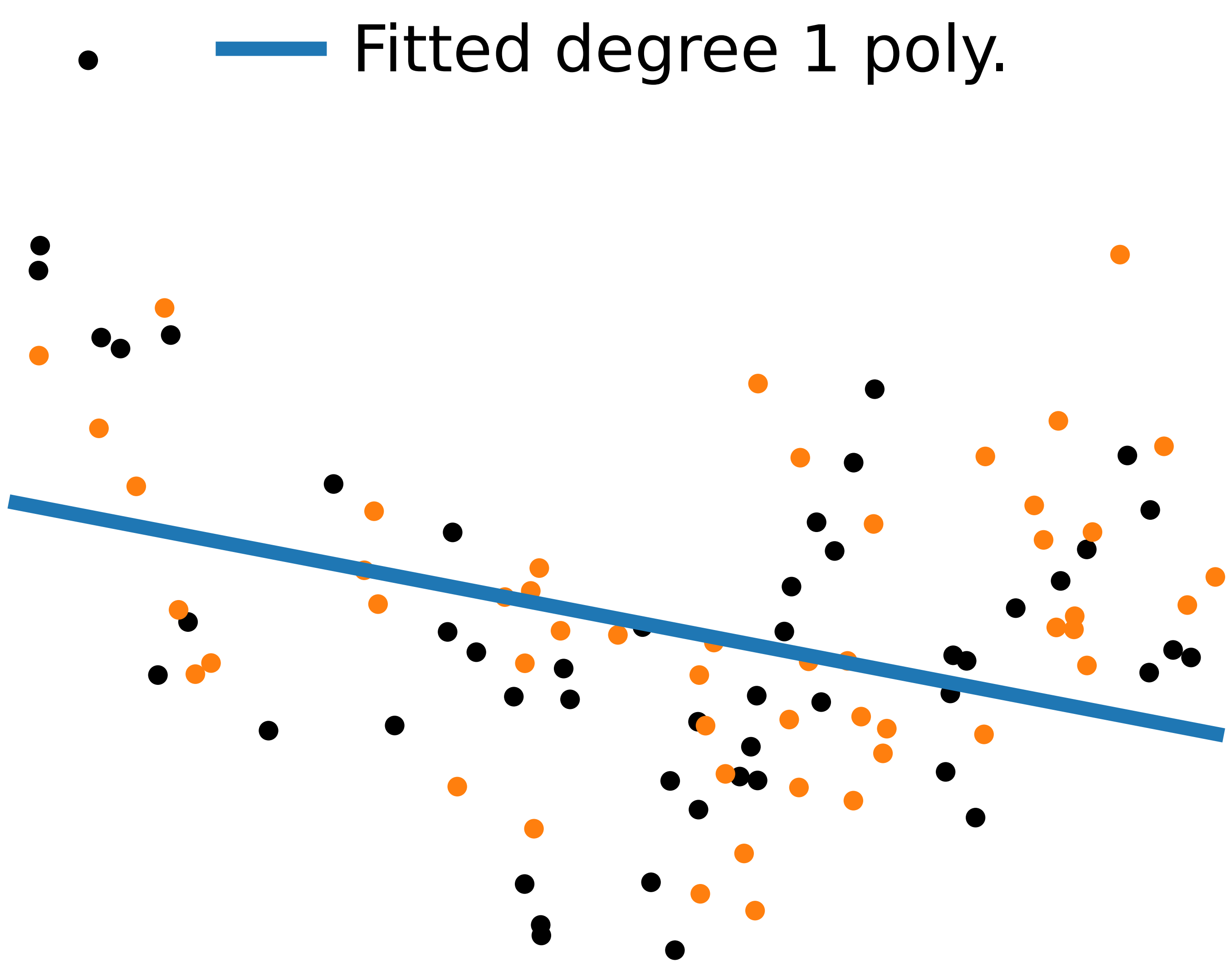
Train vs test errors

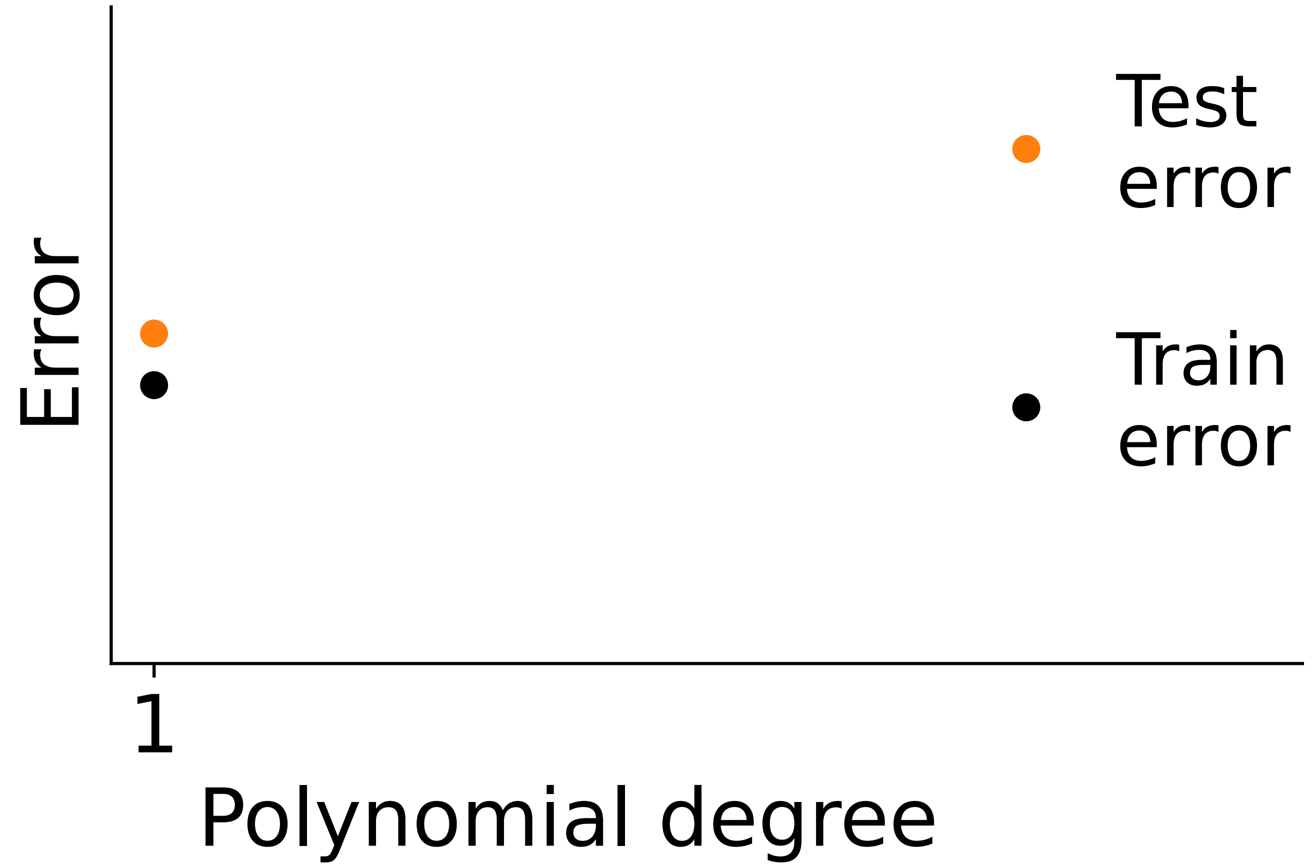


Measure:

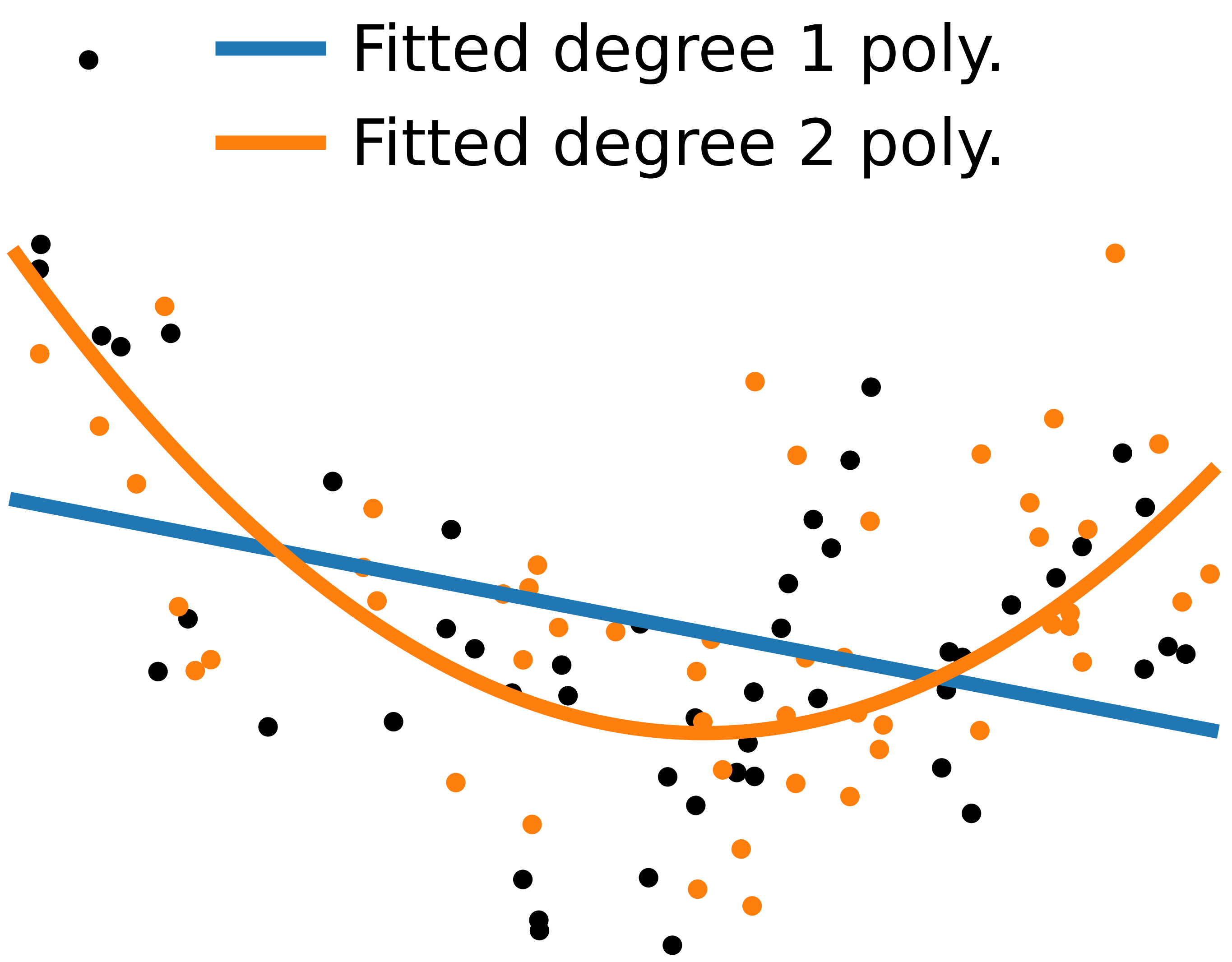
- errors on test data (generalization)
- errors on the train data

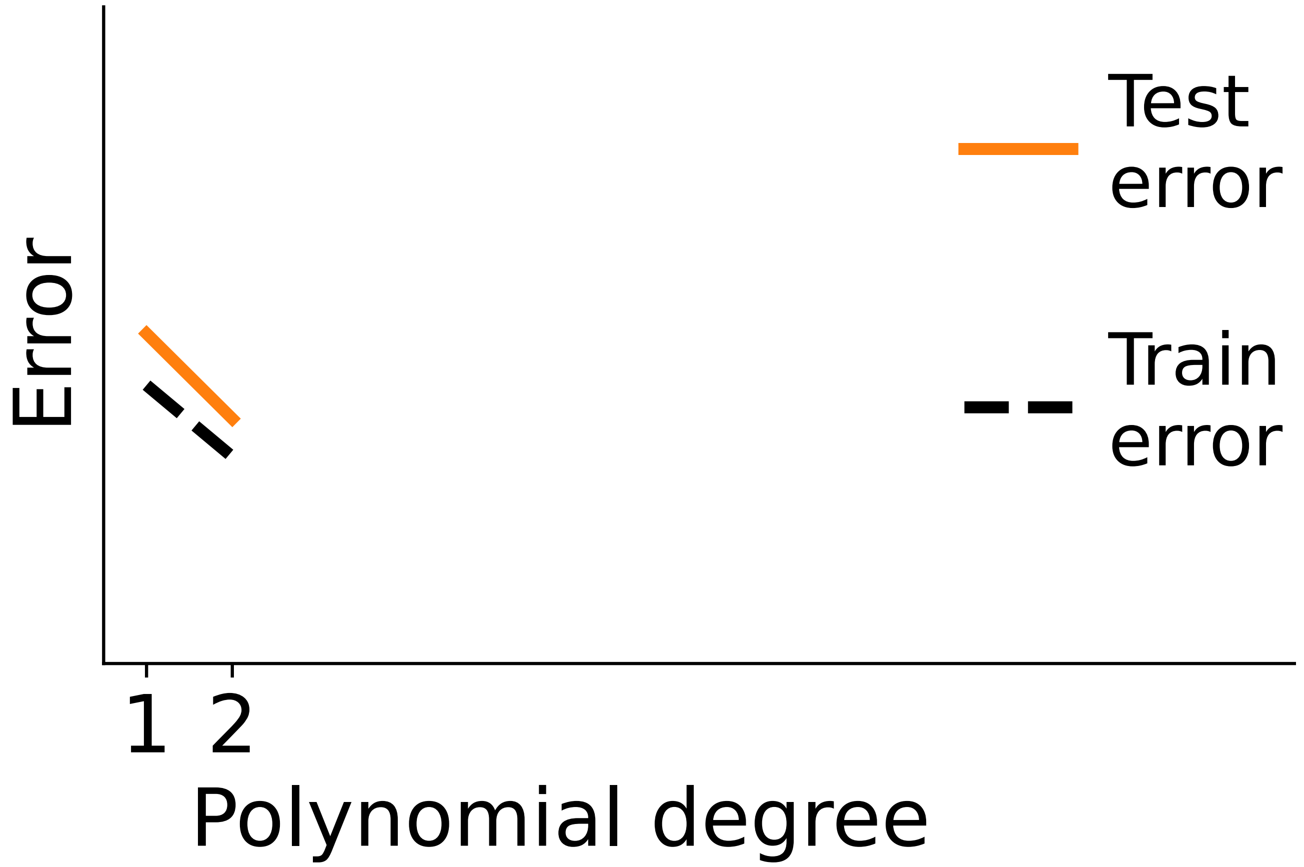
Train vs test error: increasing complexity



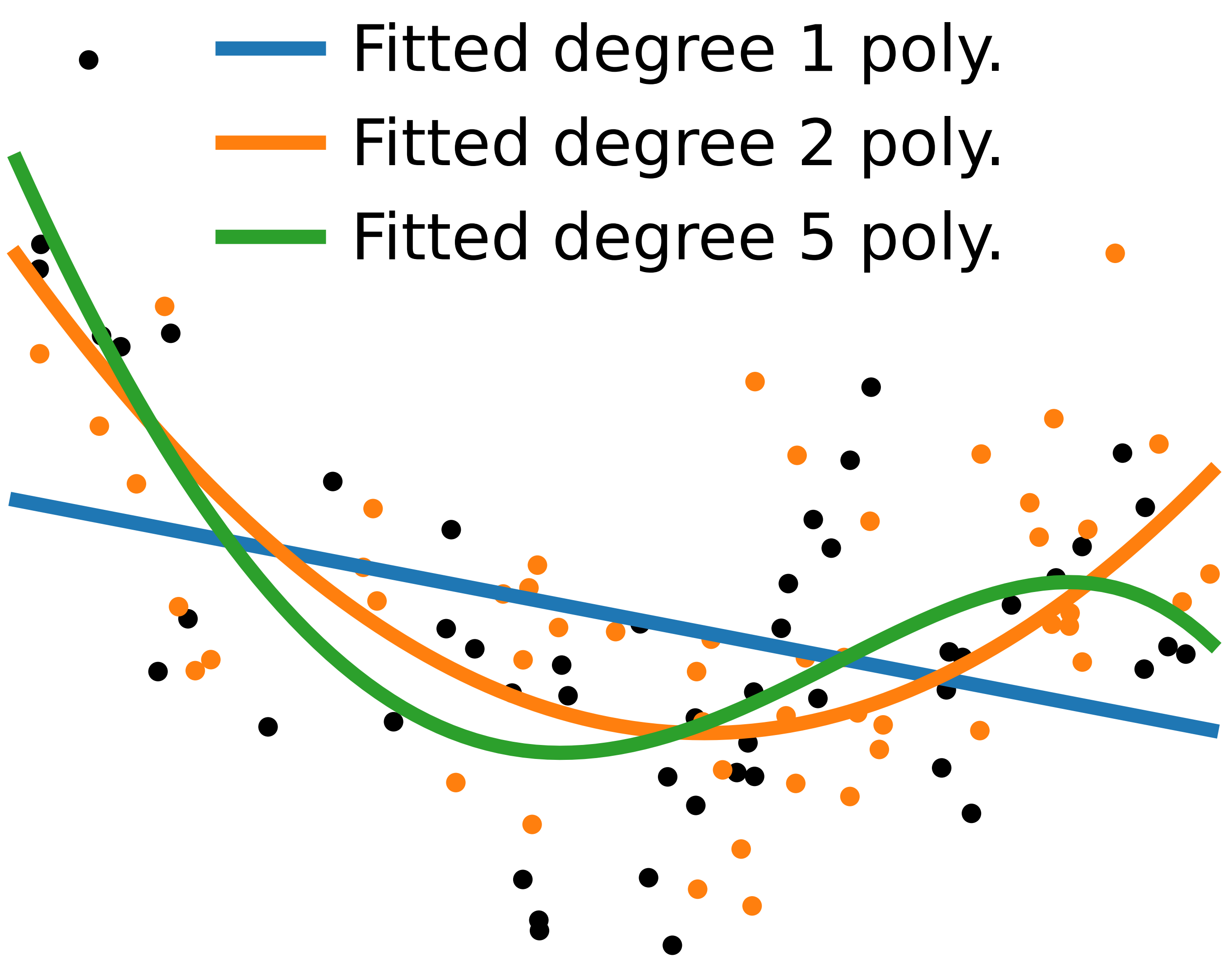


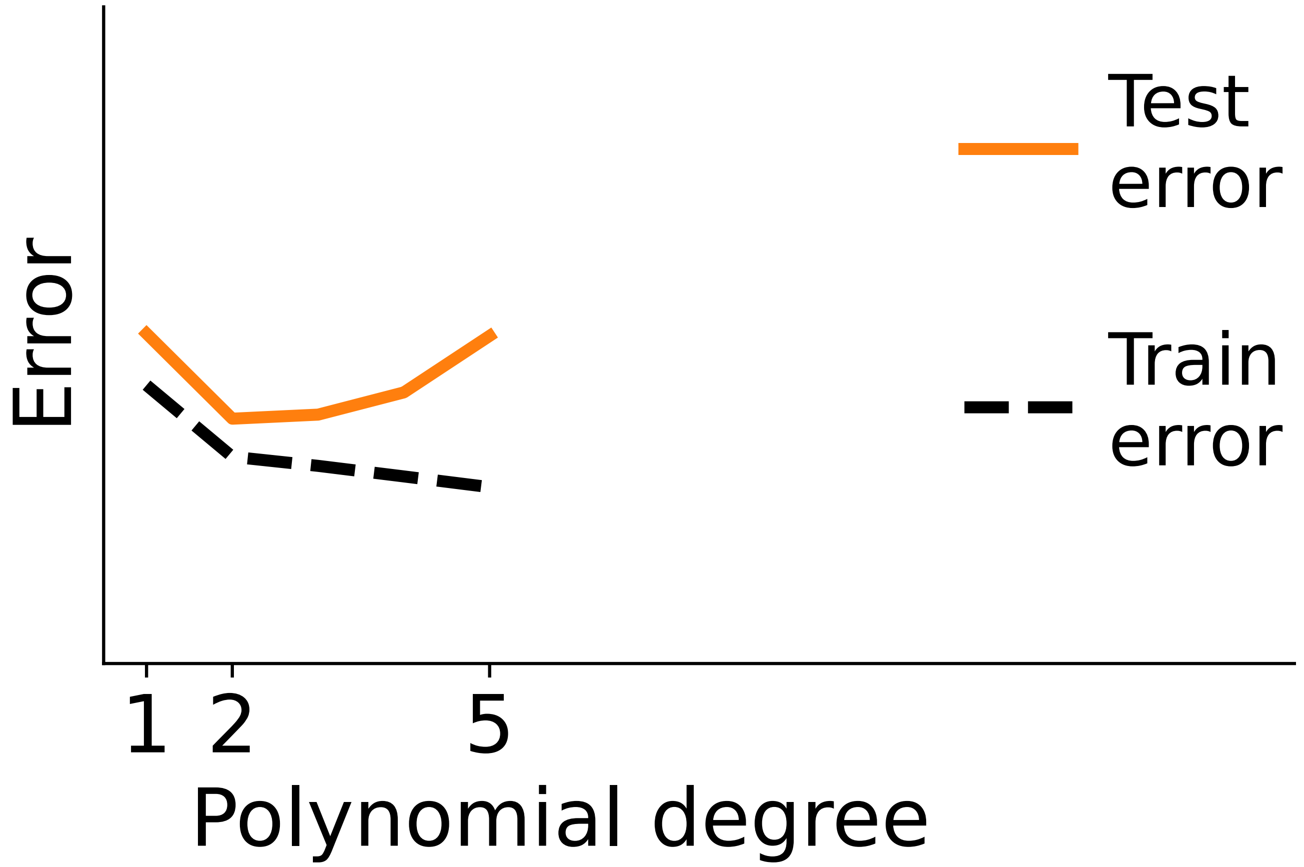
Train vs test error: increasing complexity



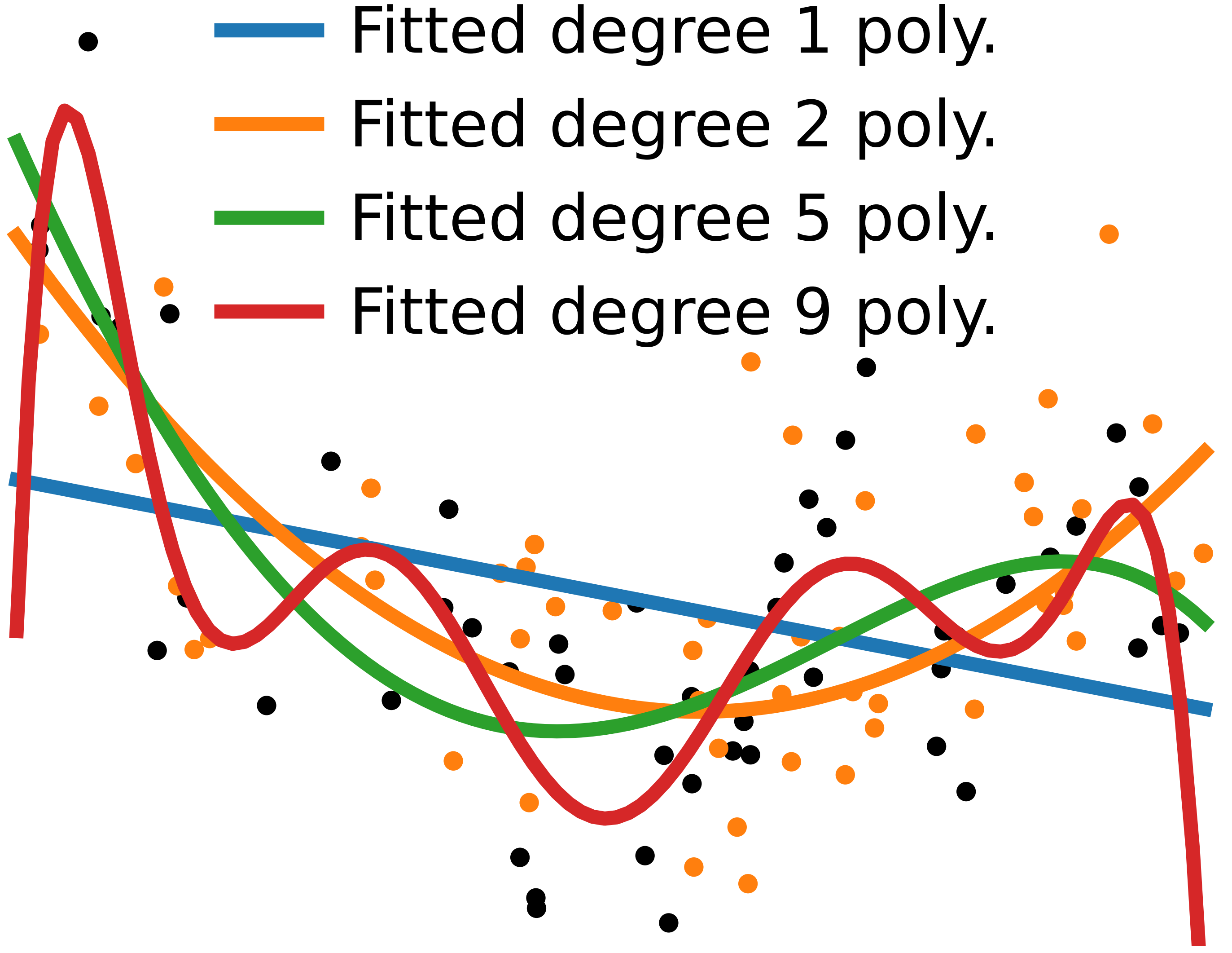


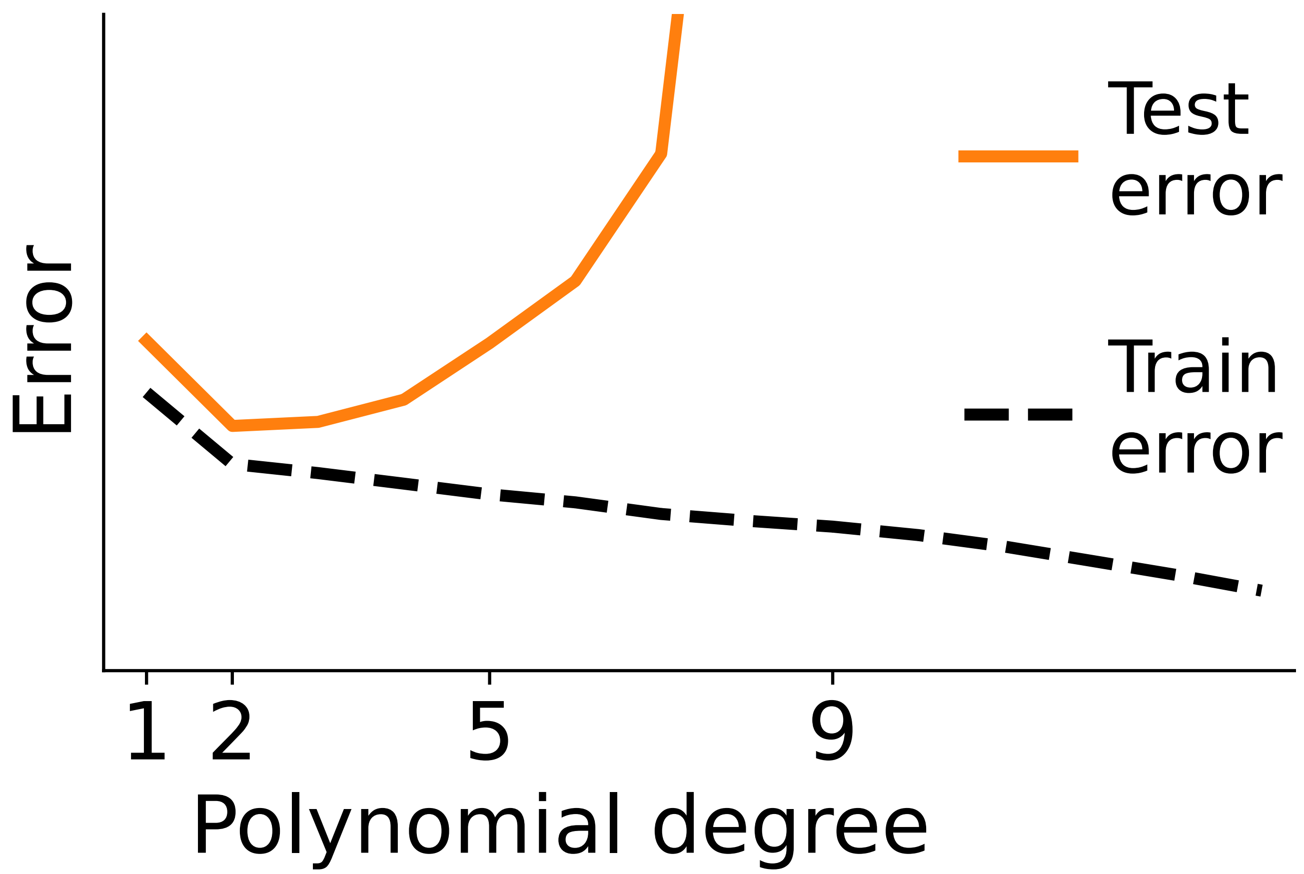
Train vs test error: increasing complexity



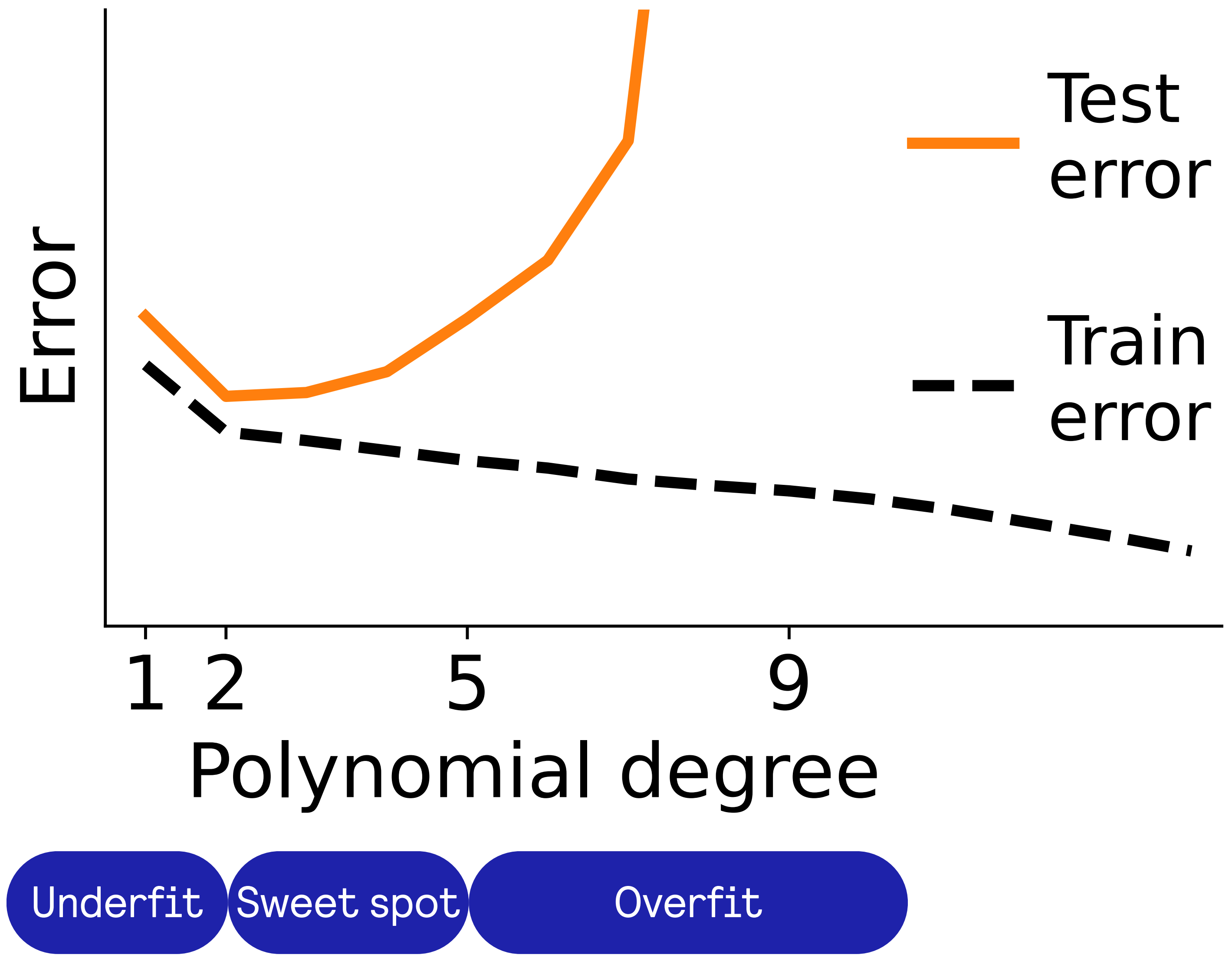


Train vs test error: increasing complexity





Train vs test error: increasing complexity

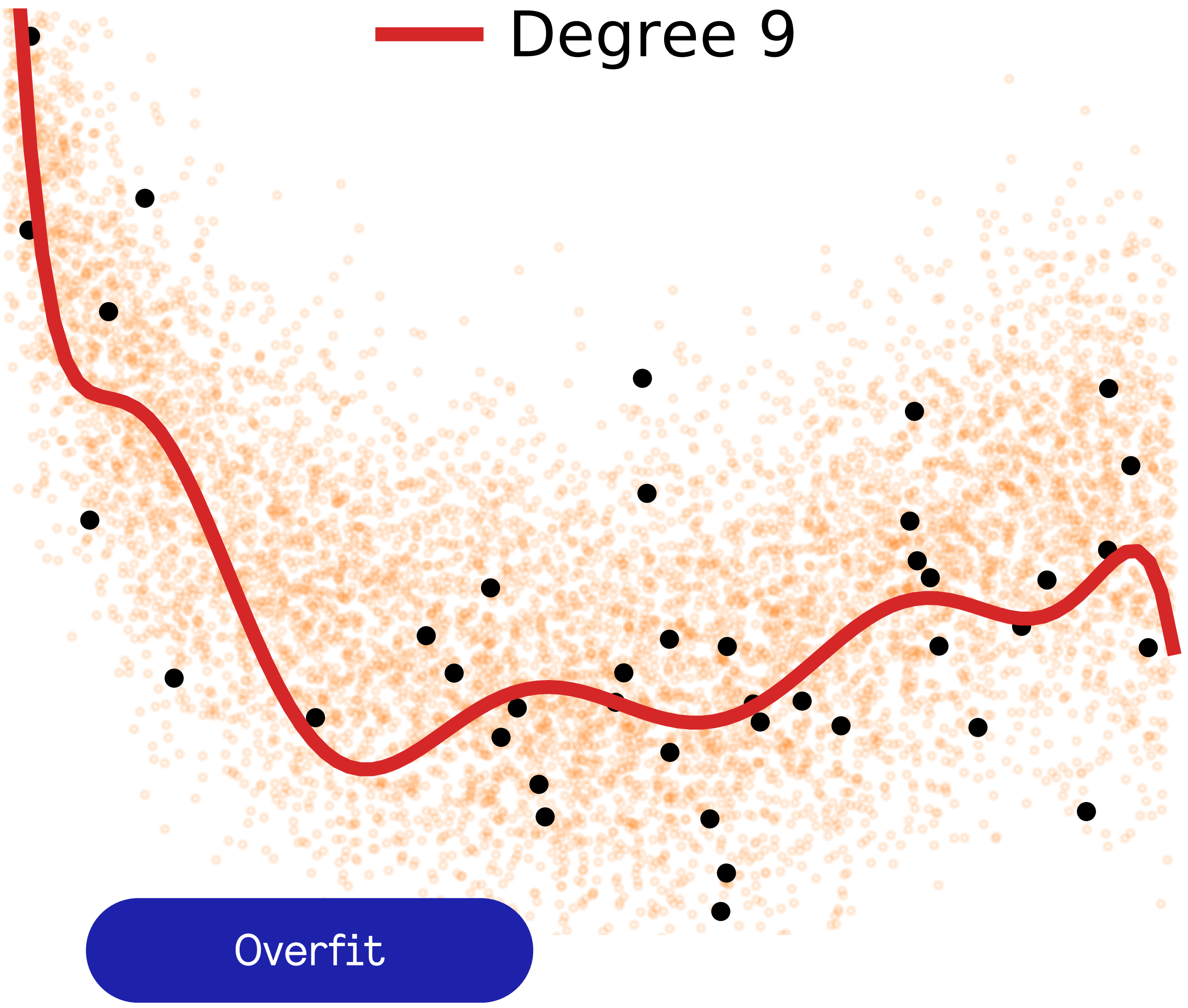


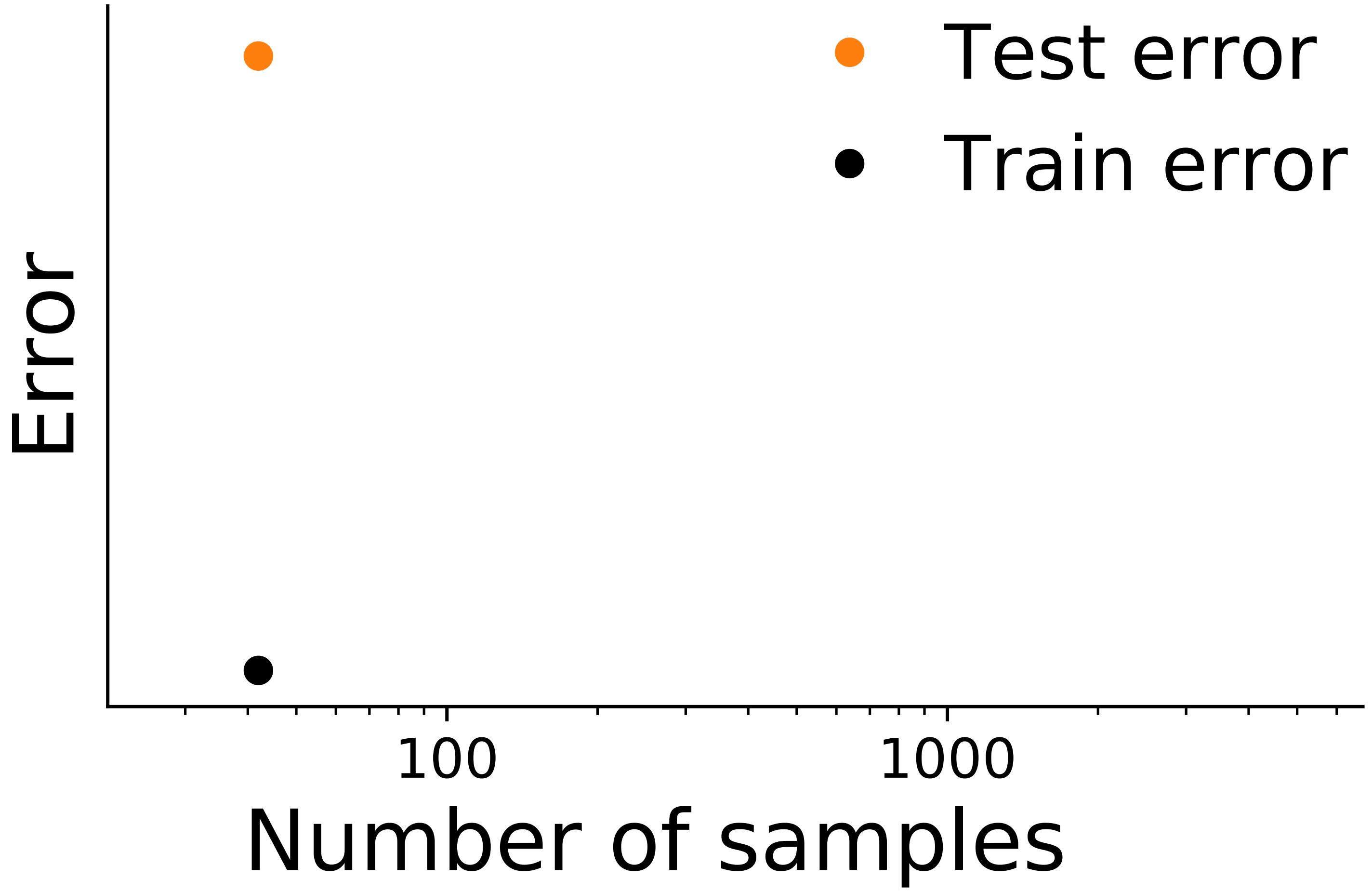
.skolar

Varying sample size

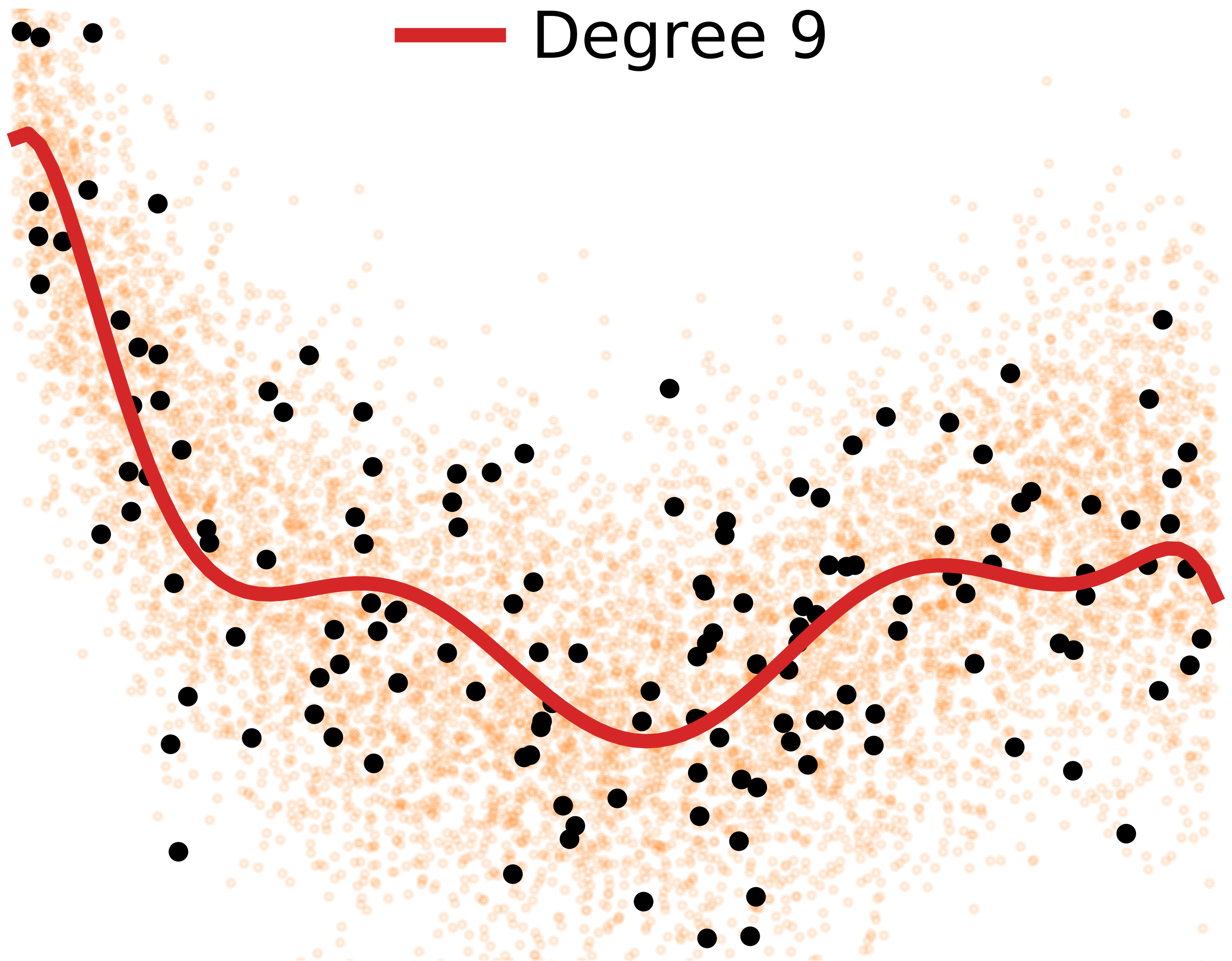
Learning curves

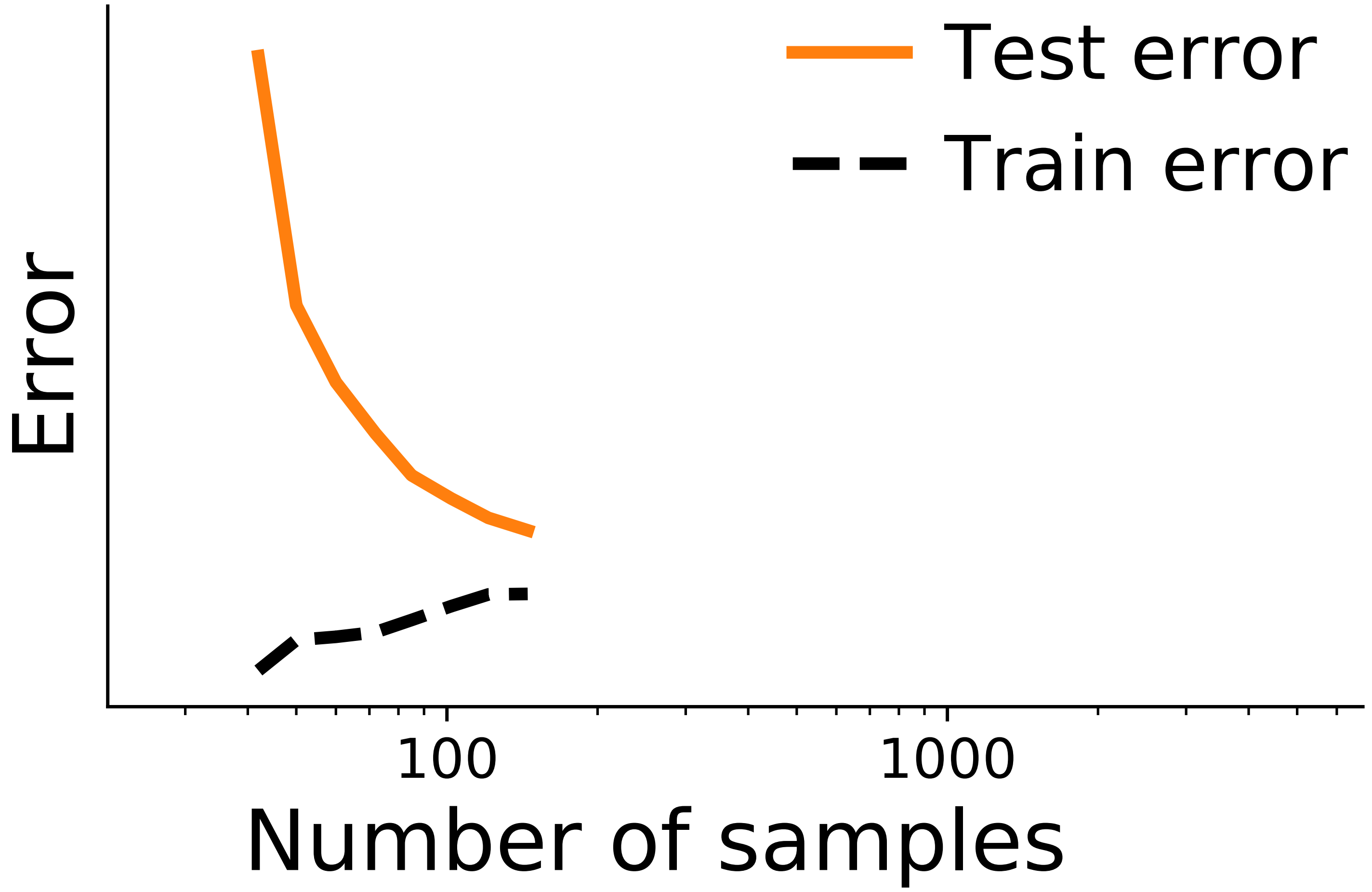
Varying sample size



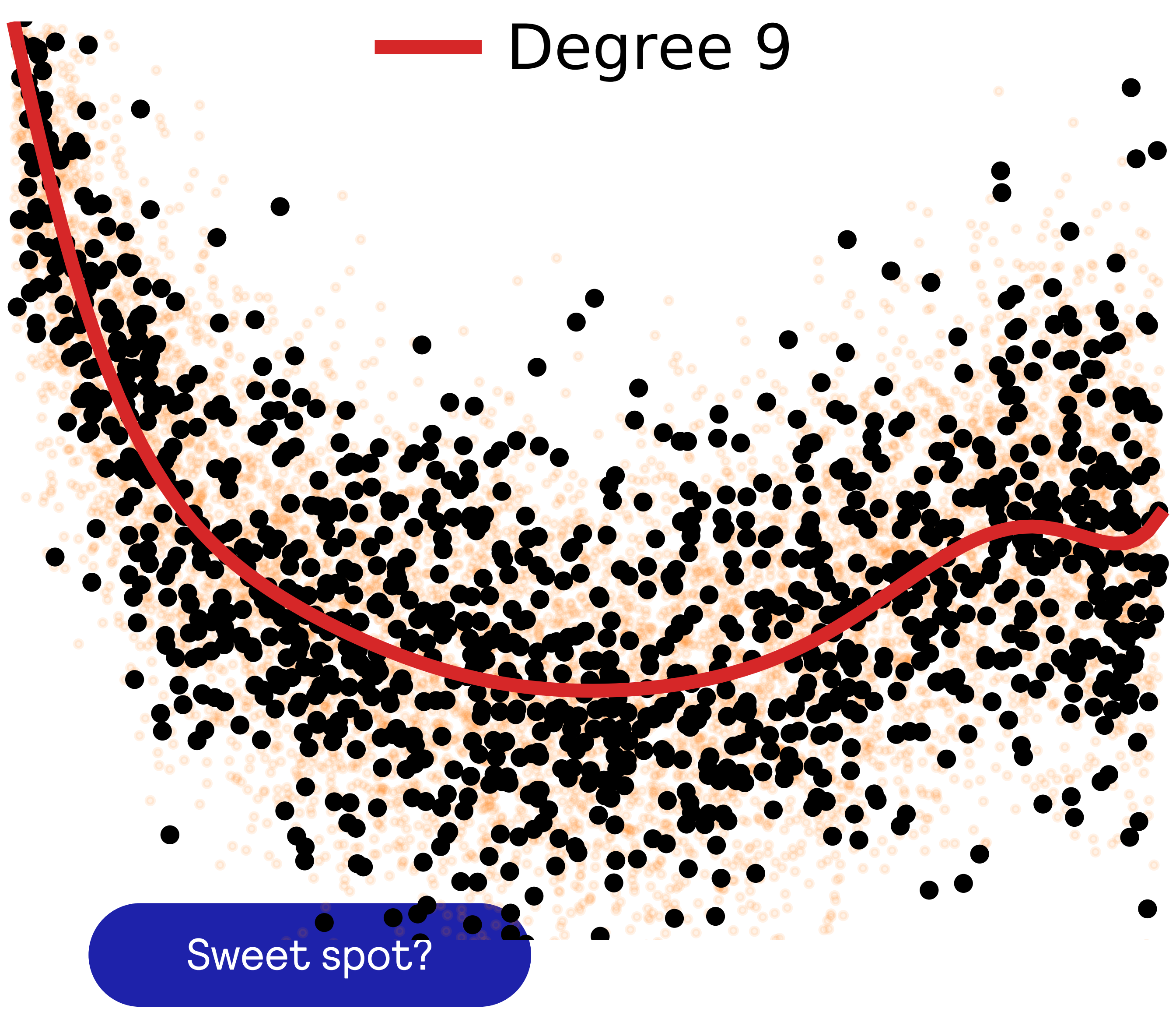


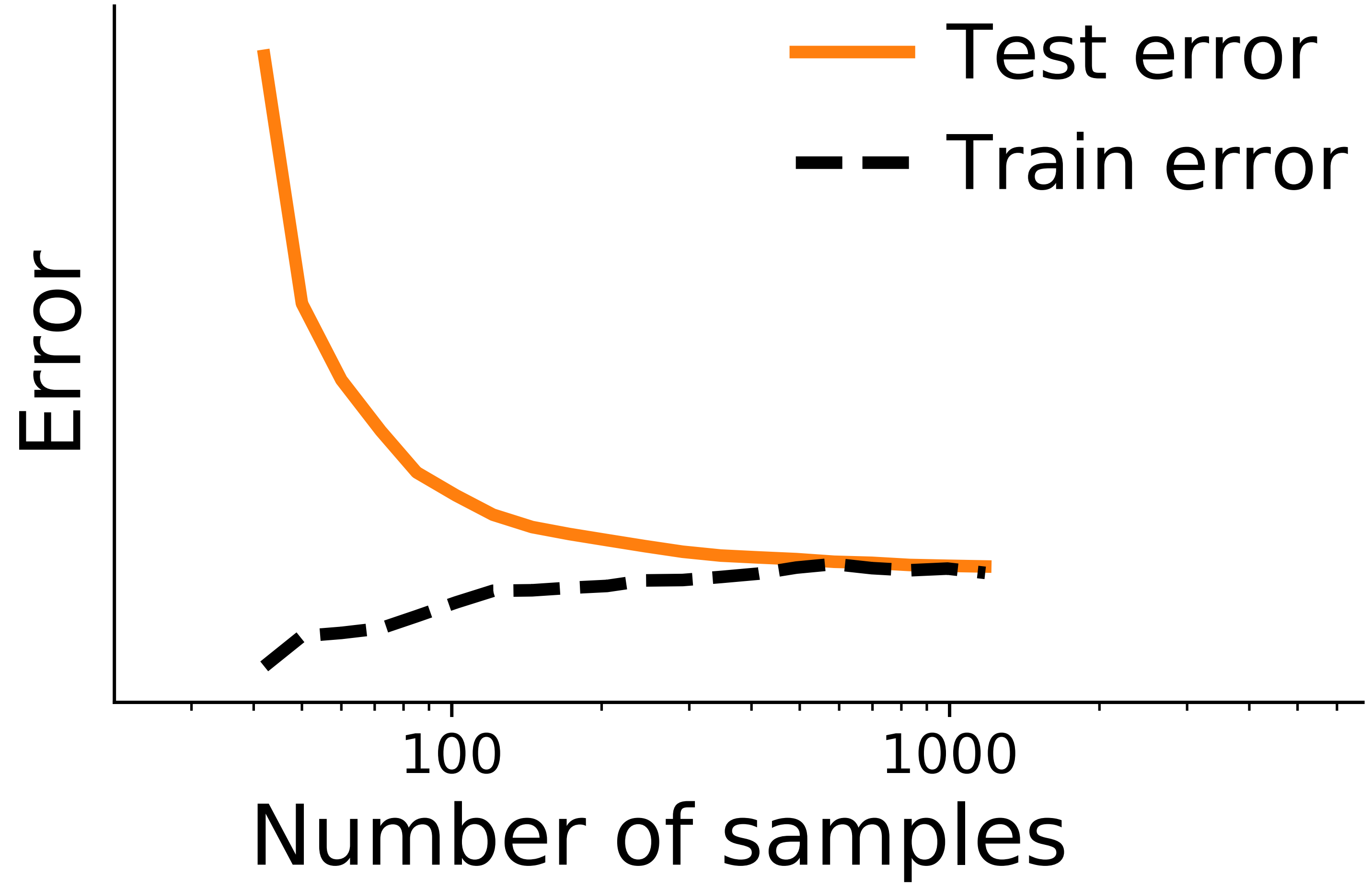
Varying sample size



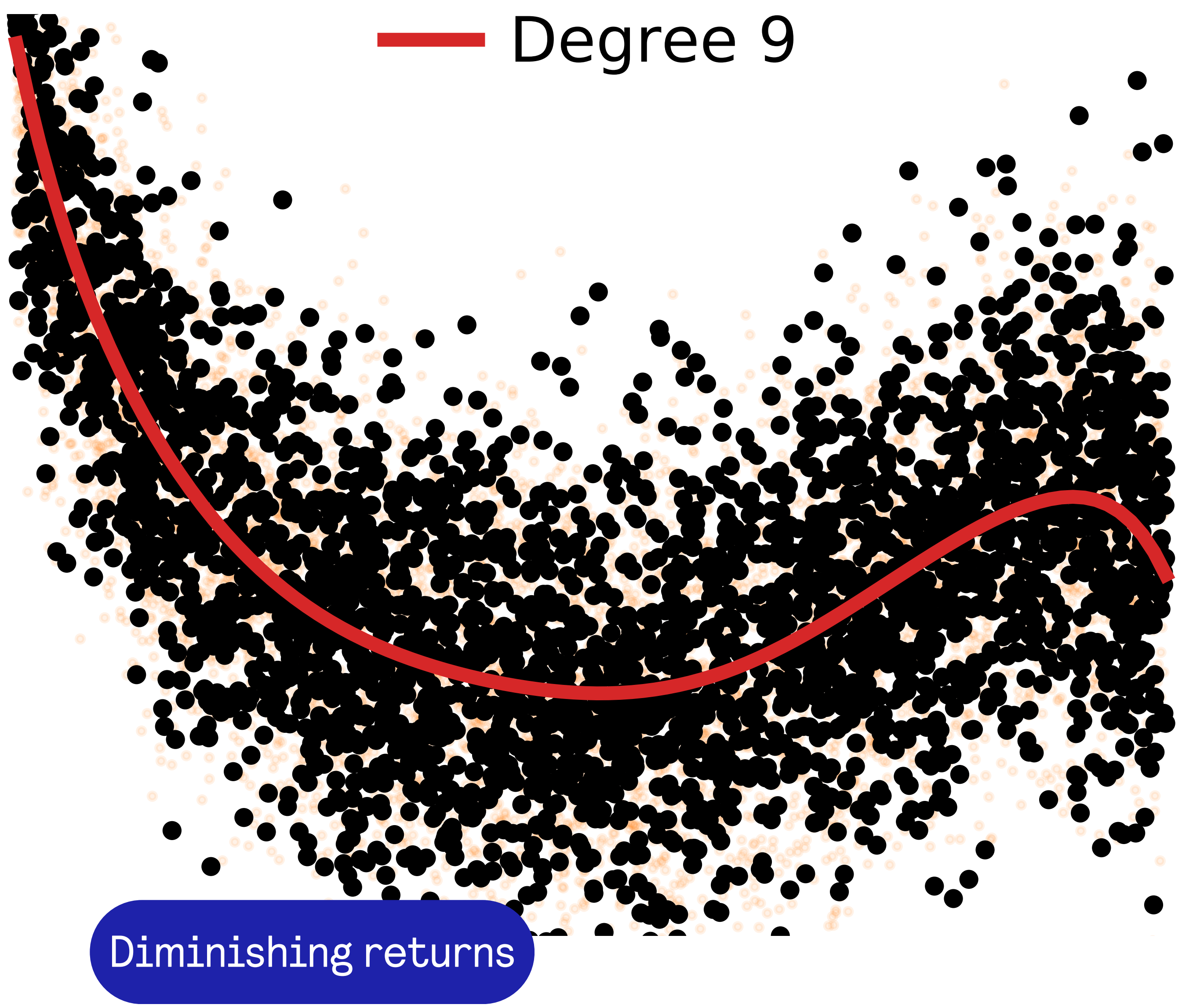


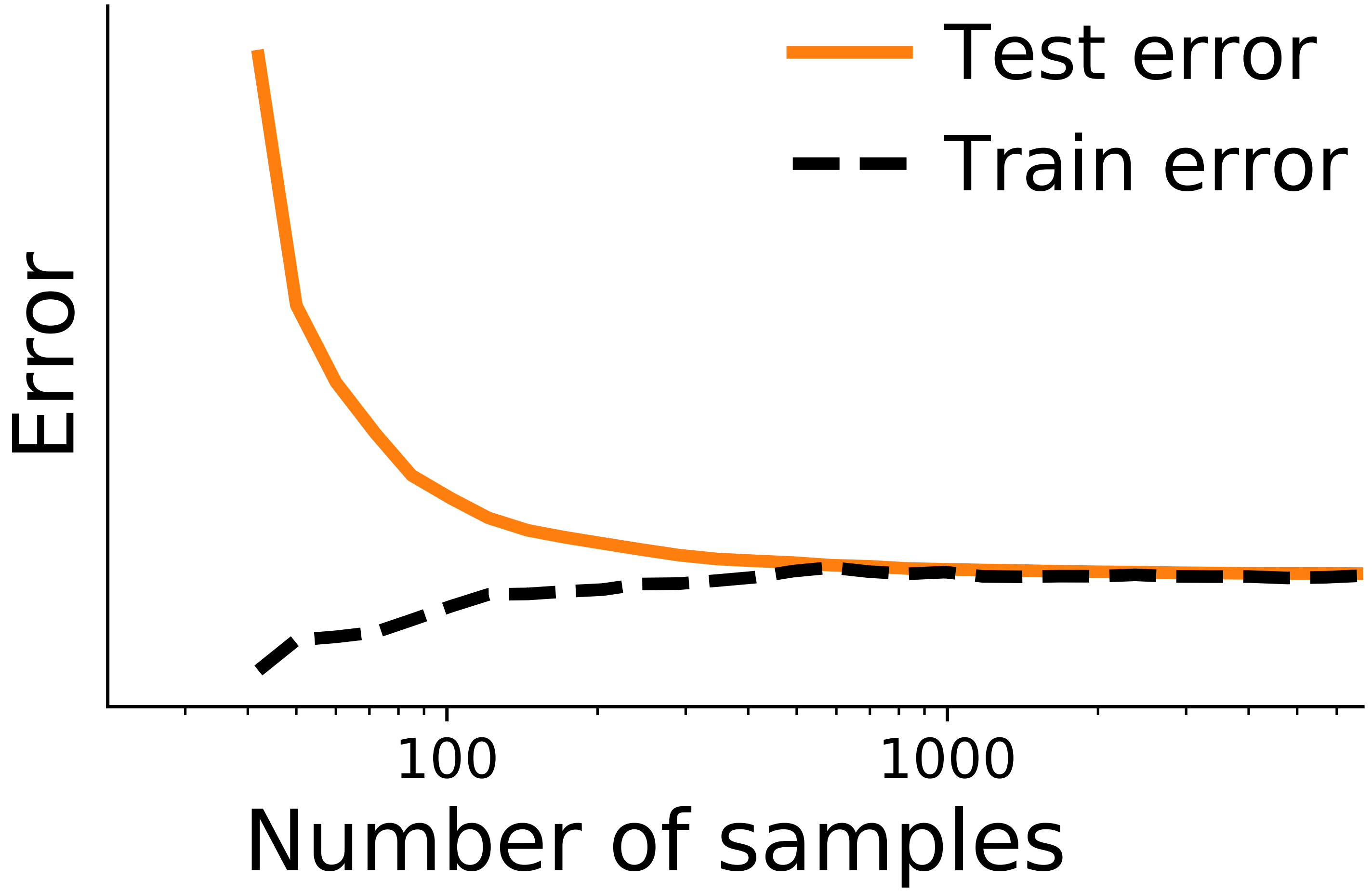
Varying sample size



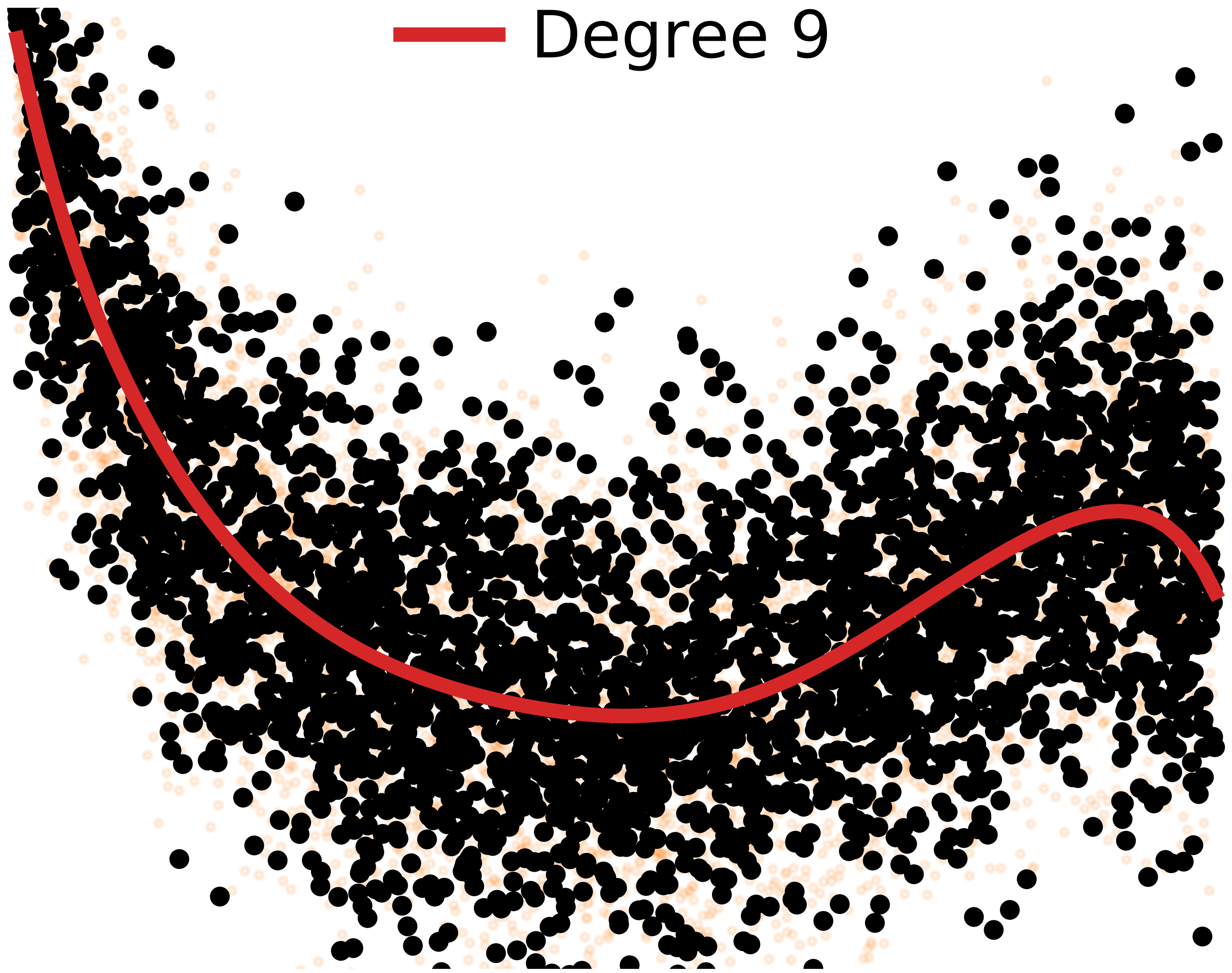


Varying sample size





Bayes error rate



The error of the best model trained on unlimited data.

Here, the data-generating process is a degree-9 polynomial

We cannot do better

Predictions limited by noise

Model families

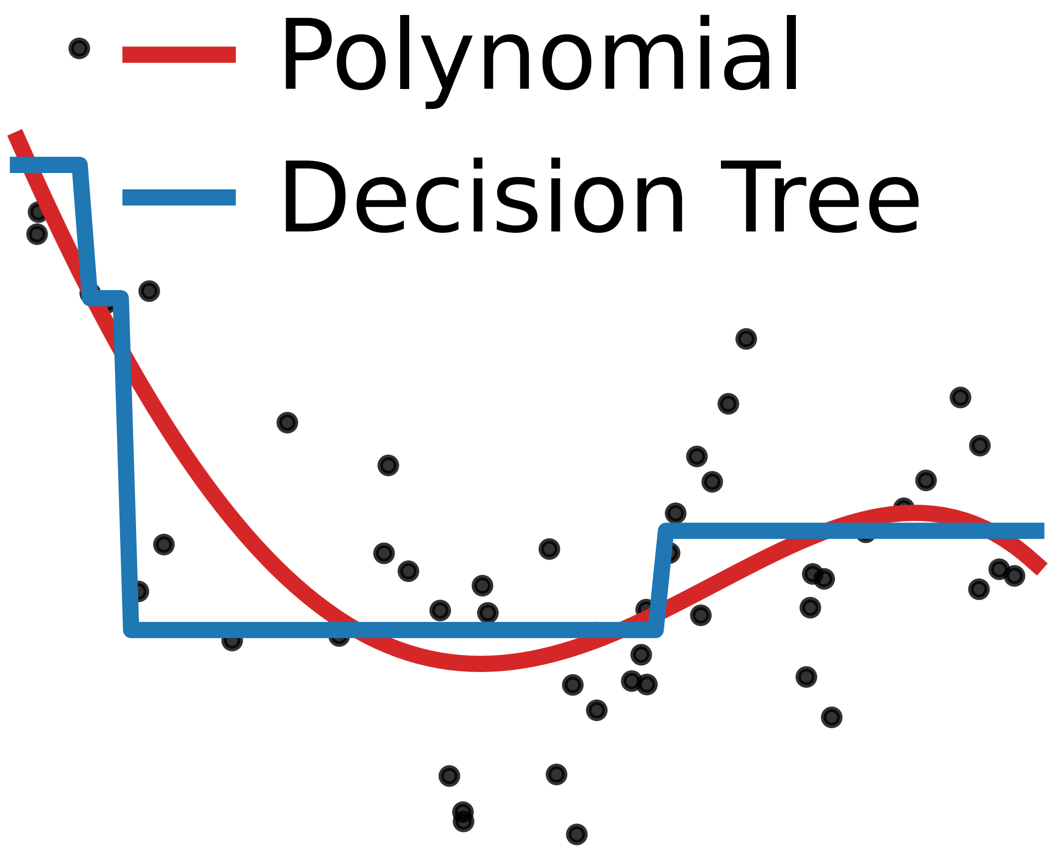
Crucial to match:

- statistical model
- data-generating process

So far: polynomials for both

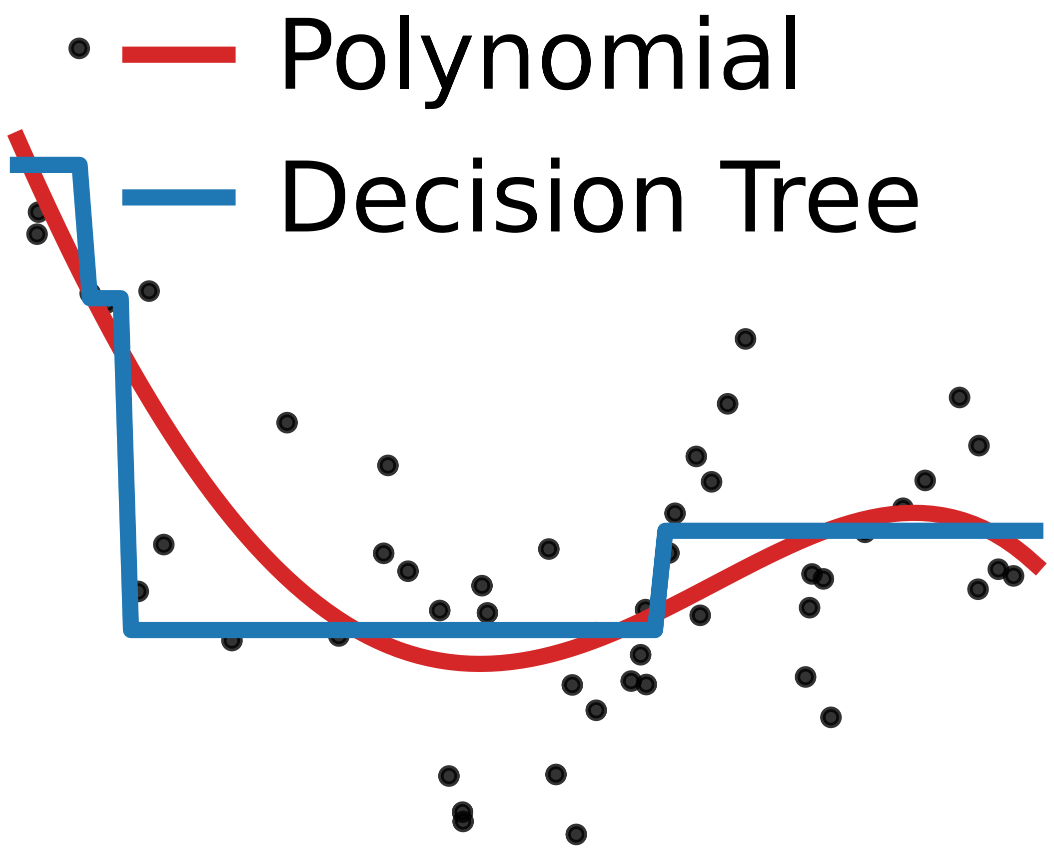
Some family names: linear models, decision trees, random forests, kernel machines, multi-layer perceptrons

Different model families

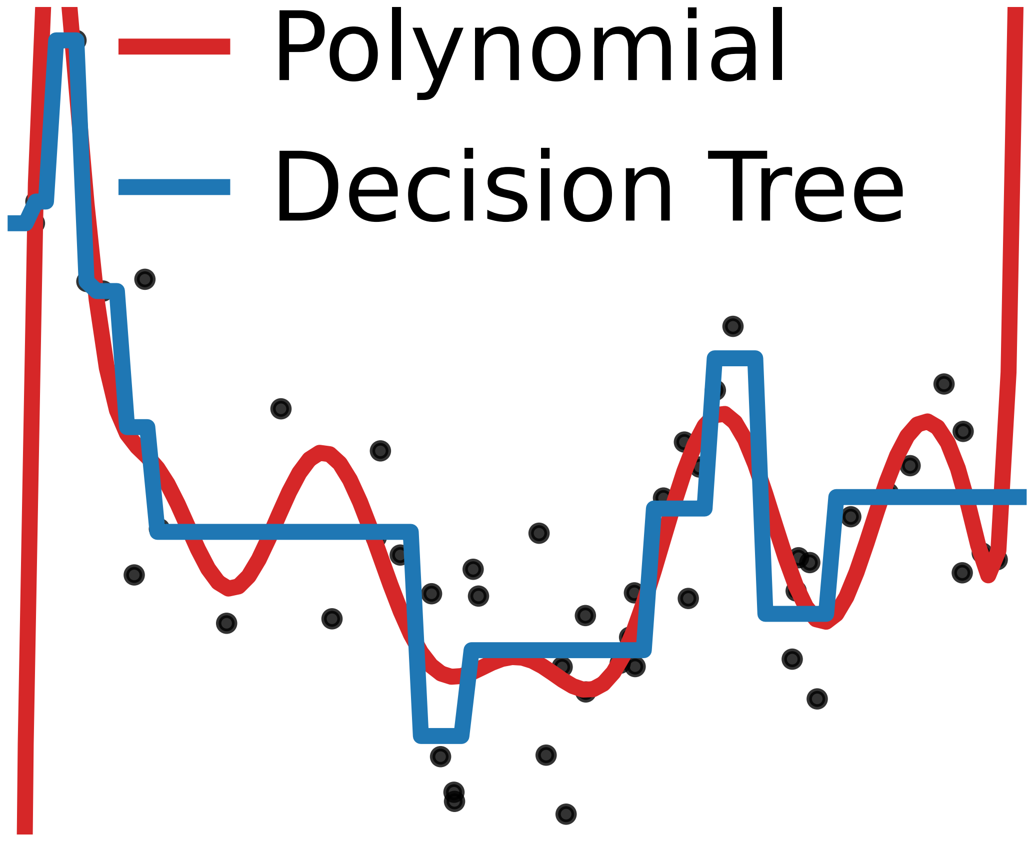


- Different "inductive bias"
- Different notion of "complexity"

Different model families



Simple variant



Complex variant

← regularization ←

Main takeaways

Models overfit:

- number of examples in the training set is too small
- testing error is much bigger than training error

Models underfit:

- models fail to capture the shape of the training set
- even the training error is large

Different model families = different complexity & inductive bias