COS324:

Introduction To Machine Learning

Prof. Yoram Singer

PRINCETON UNIVERSITY

Topic: Generalization and Regularization I

Thus Far

Definitions of learning problems

Linear and non-linear models

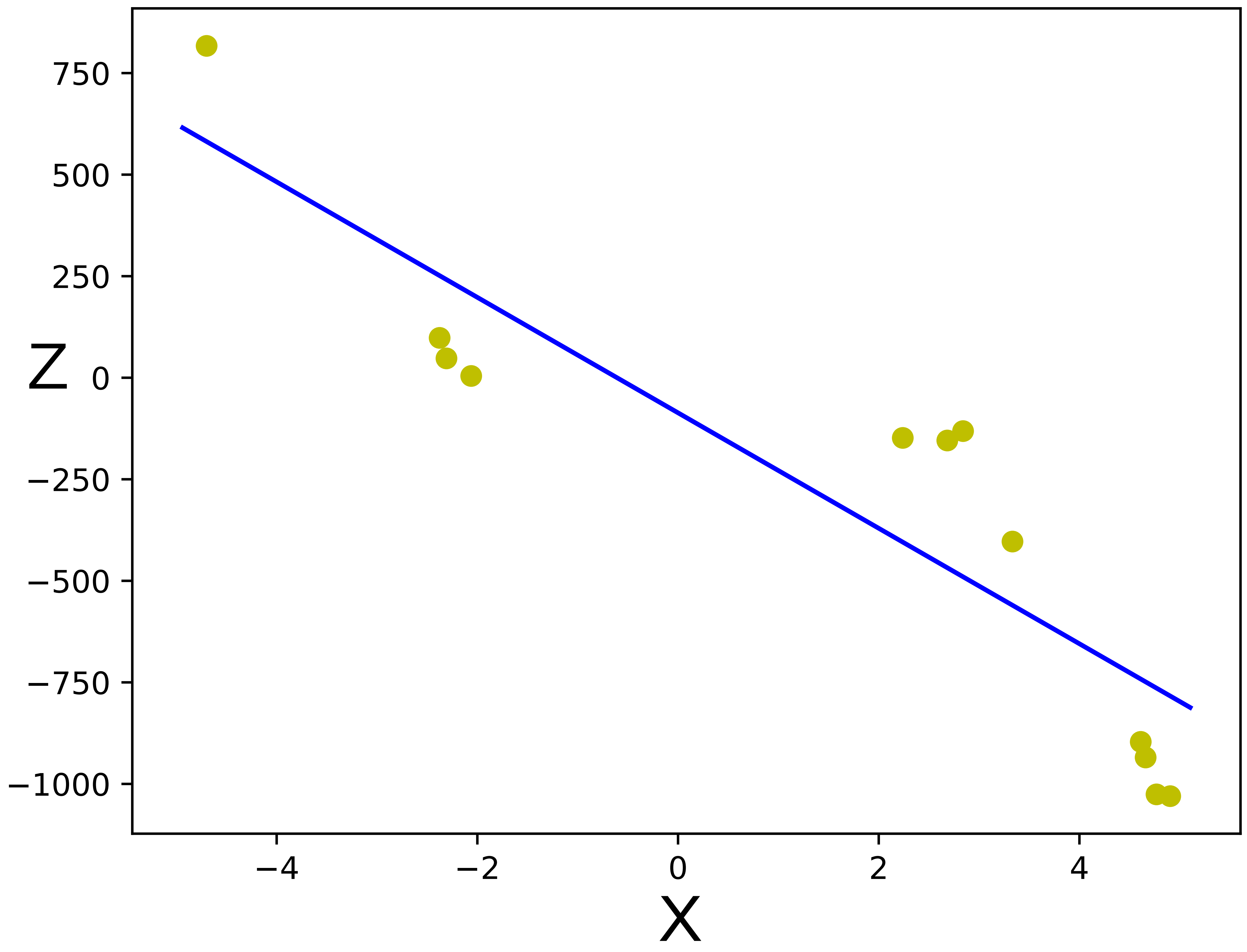
Using differentiable loss for learning

Learning algorithms

Mentioned in passing through examples **test** loss & error

Should the loss/error on unseen data resemble training loss/error ?

Dataset of examples each has two features $\left\{(x_i, z_i)\right\}_{i=1}^{20}$



Learn a function $f : \mathbf{R} \rightarrow \mathbf{R}$

Regression loss: $\left(f(x) - z\right)^2$

Choose an order $\mathbf{p}$ for a polynomial:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_p x^p$$

Learn coefficients $a_0, a_1, a_2, \ldots, a_p$

Learning Polynomials

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \cdots \\ \mathbf{x}_n \end{bmatrix}$$

Replace $x \mapsto \mathbf{x} = (1, x, x^2, x^3, \ldots, x^p)$

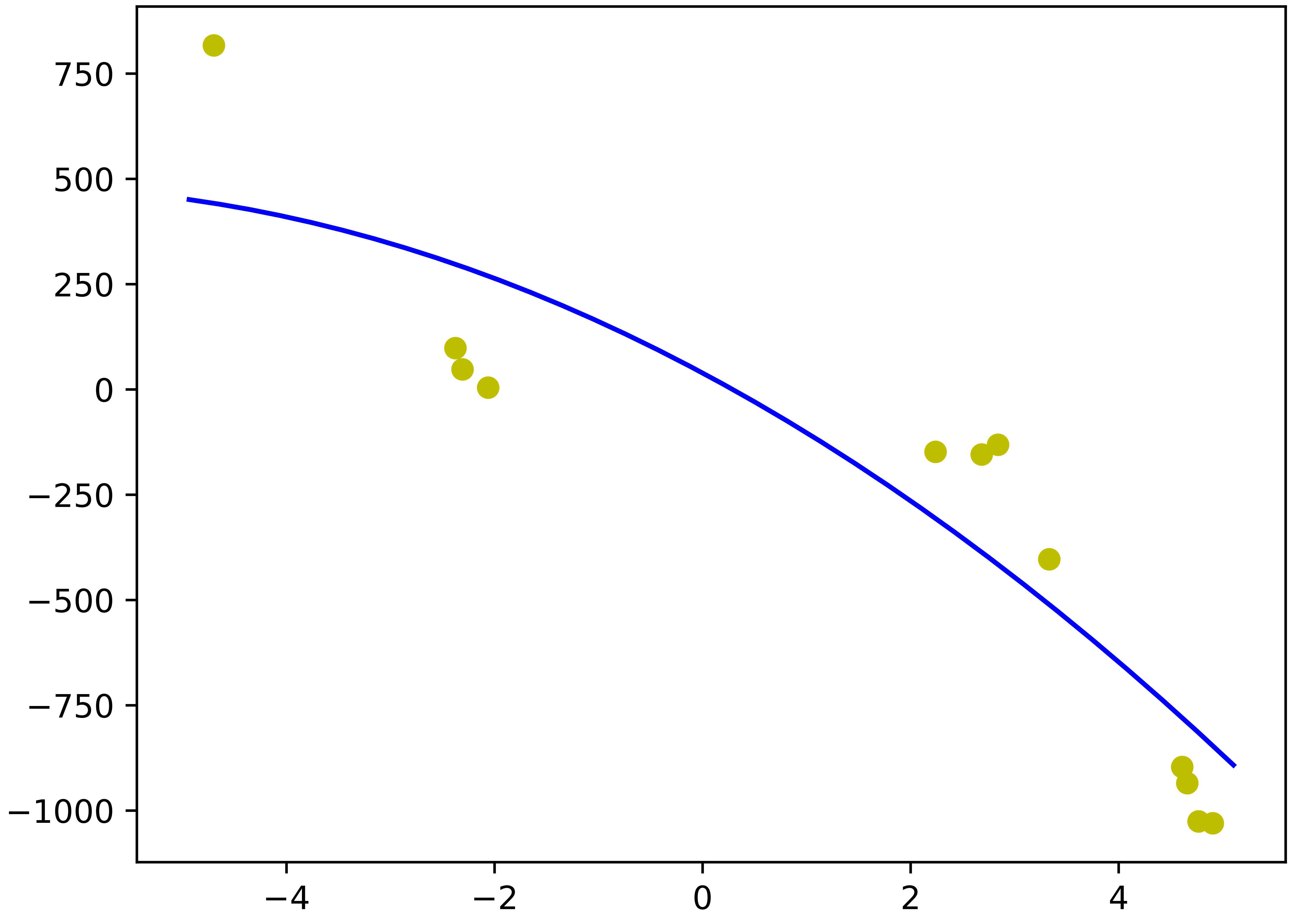
For example suppose $x_i = 3$ and $p = 5$ then $x_i \mapsto \mathbf{x}_i = (1, 3, 9, 27, 81, 243)$

1	x_1	$(x_1)^2$	…	…	$(x_1)^5$
1	x_2	$(x_2)^2$			…
1	x_3	$(x_3)^2$			…
1	x_4	$(x_4)^2$	…	…	$(x_4)^5$

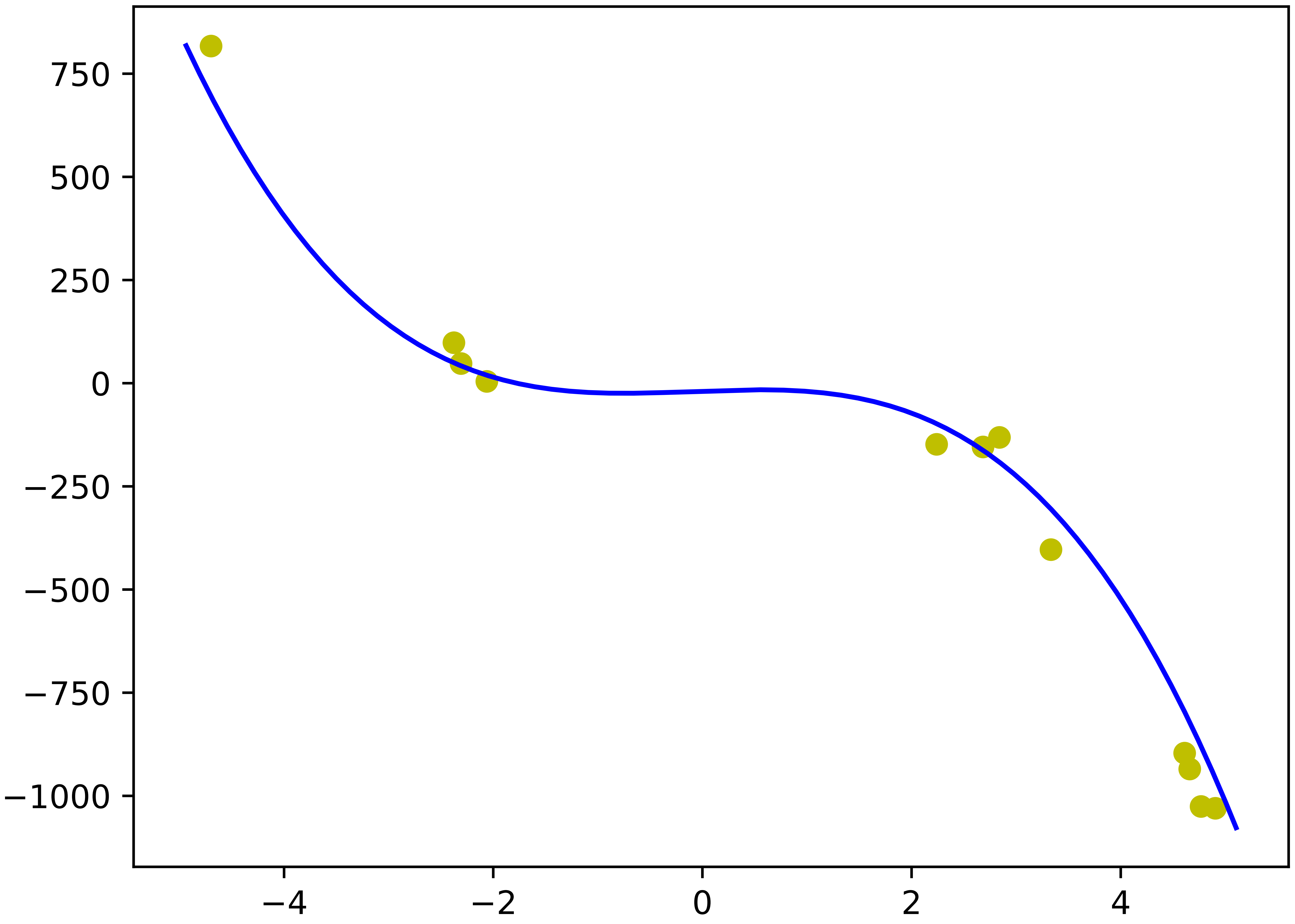
$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \approx \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

$$\min_{\mathbf{a}} \|X\mathbf{a} - \mathbf{z}\|^2$$

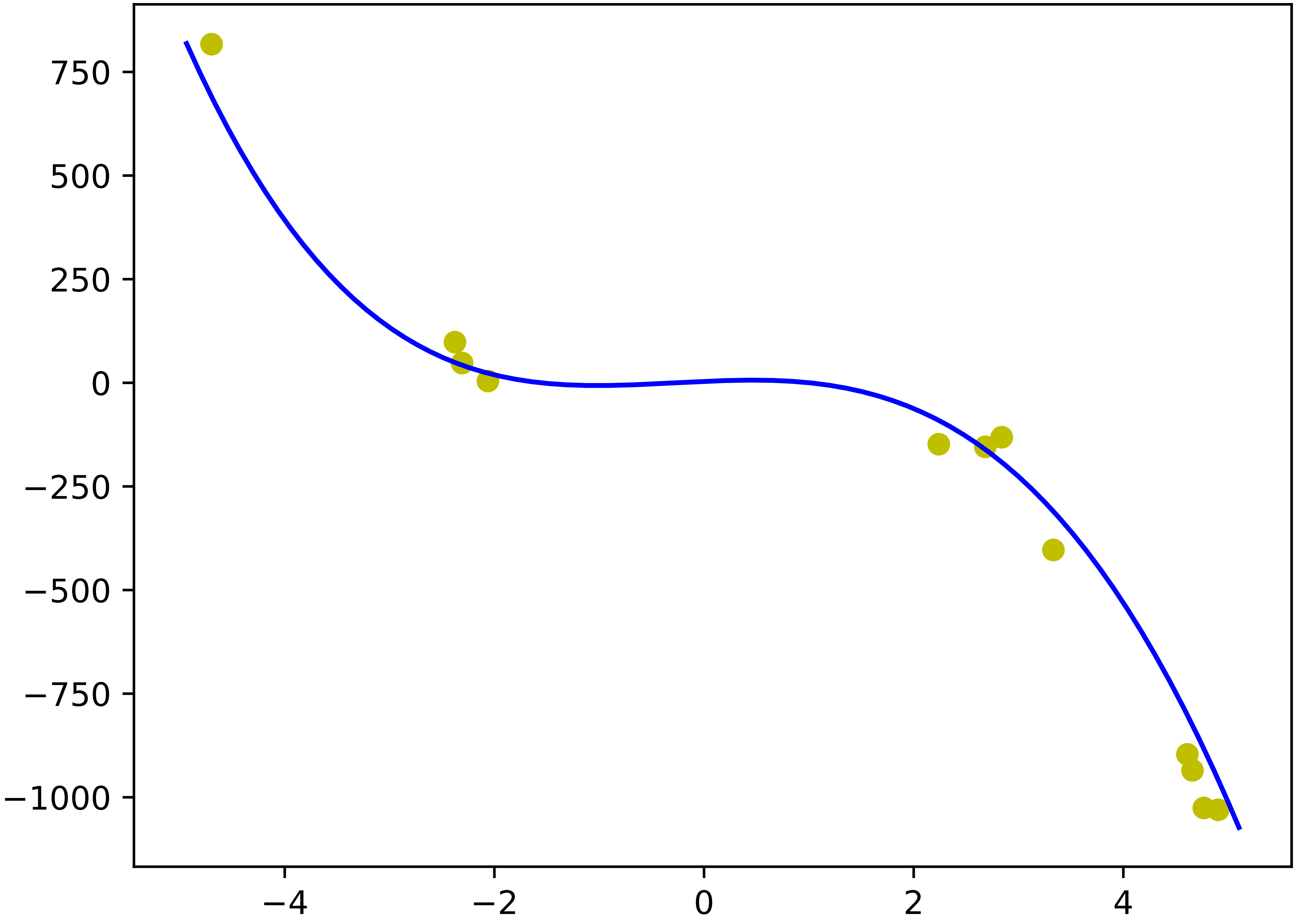
Degree 2 Fit to Training Data



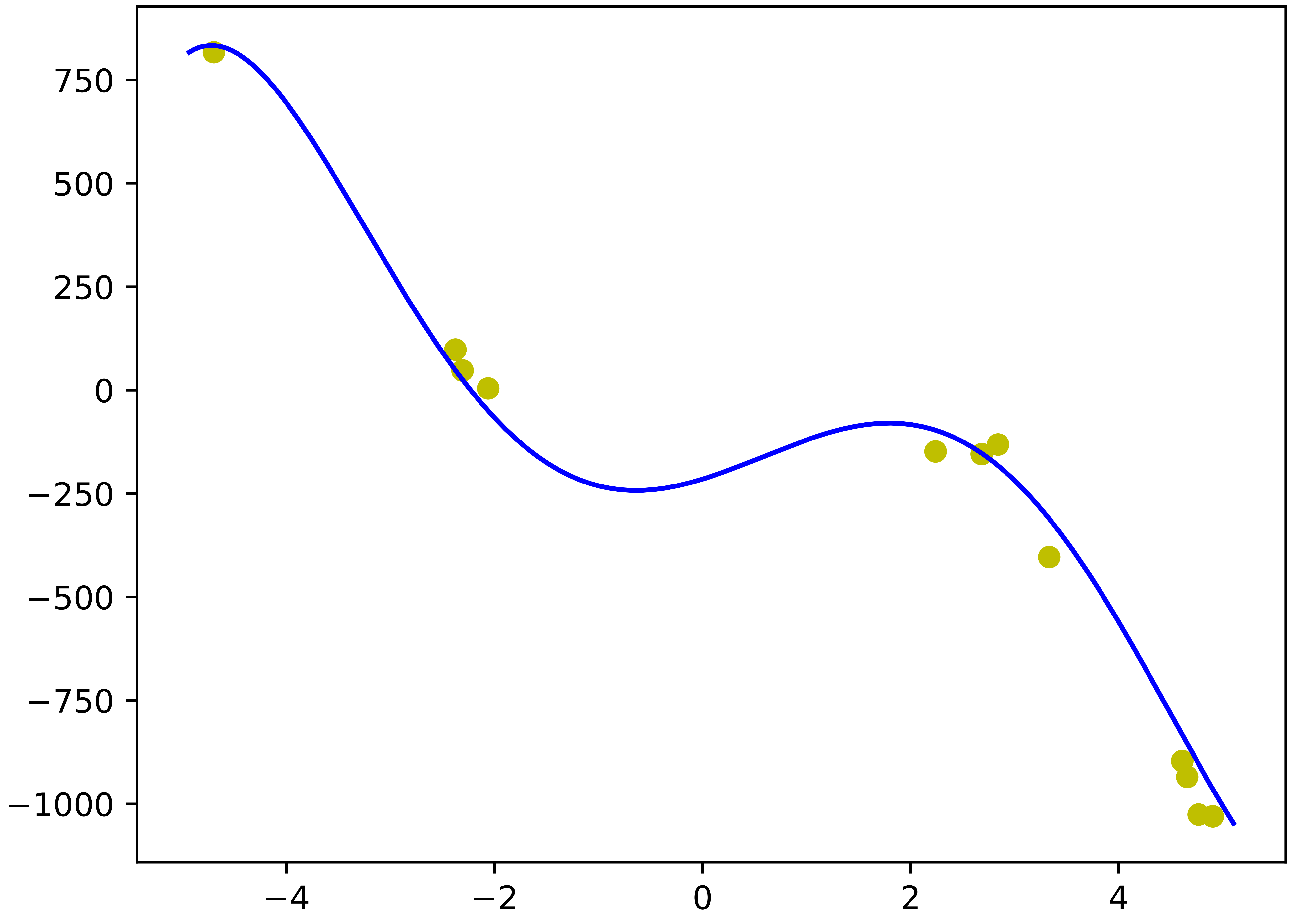
Degree 3 Fit to Training Data



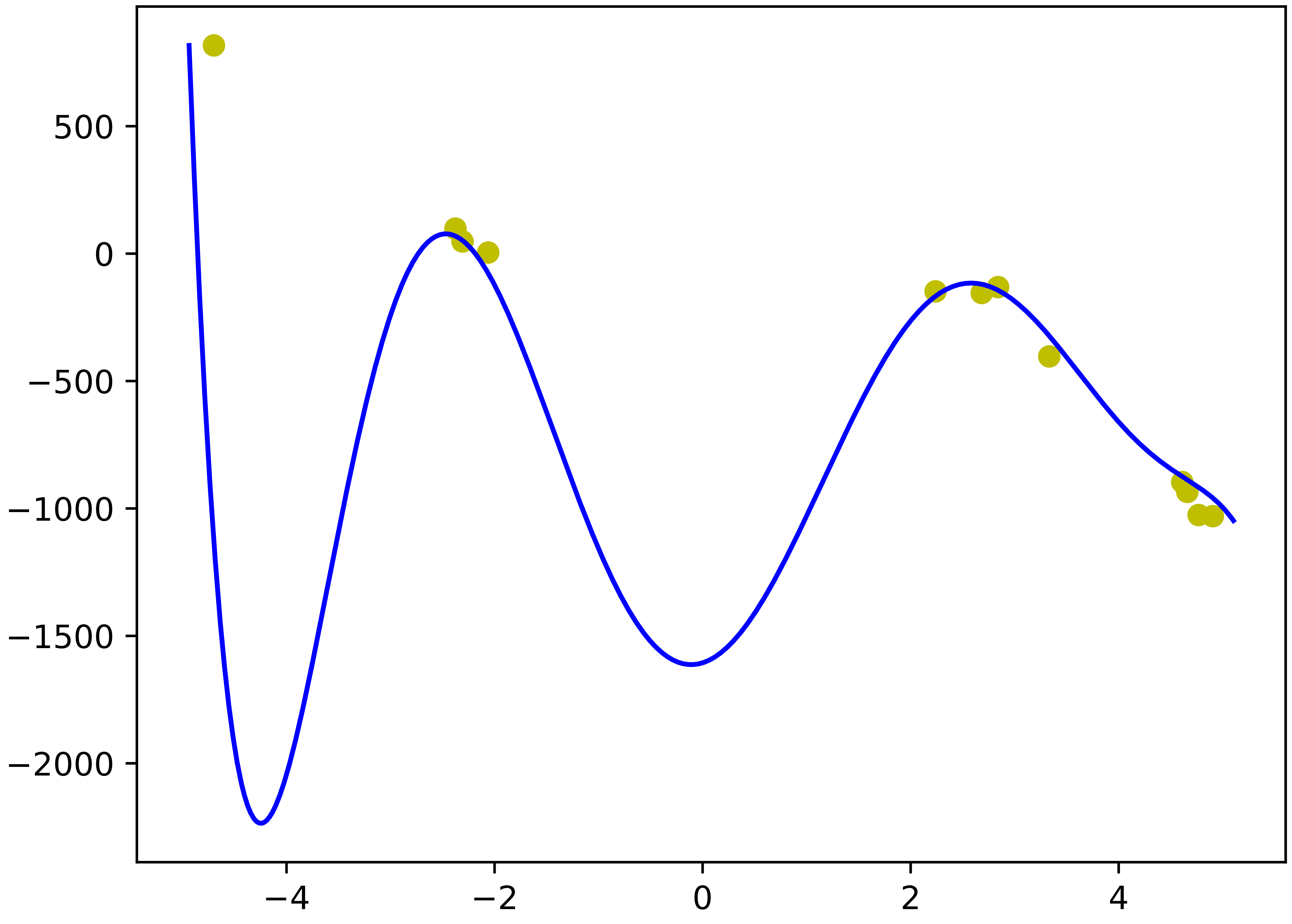
Degree 3 Fit to Training Data



Degree 5 Fit to Training Data



Degree 7 Fit to Training Data

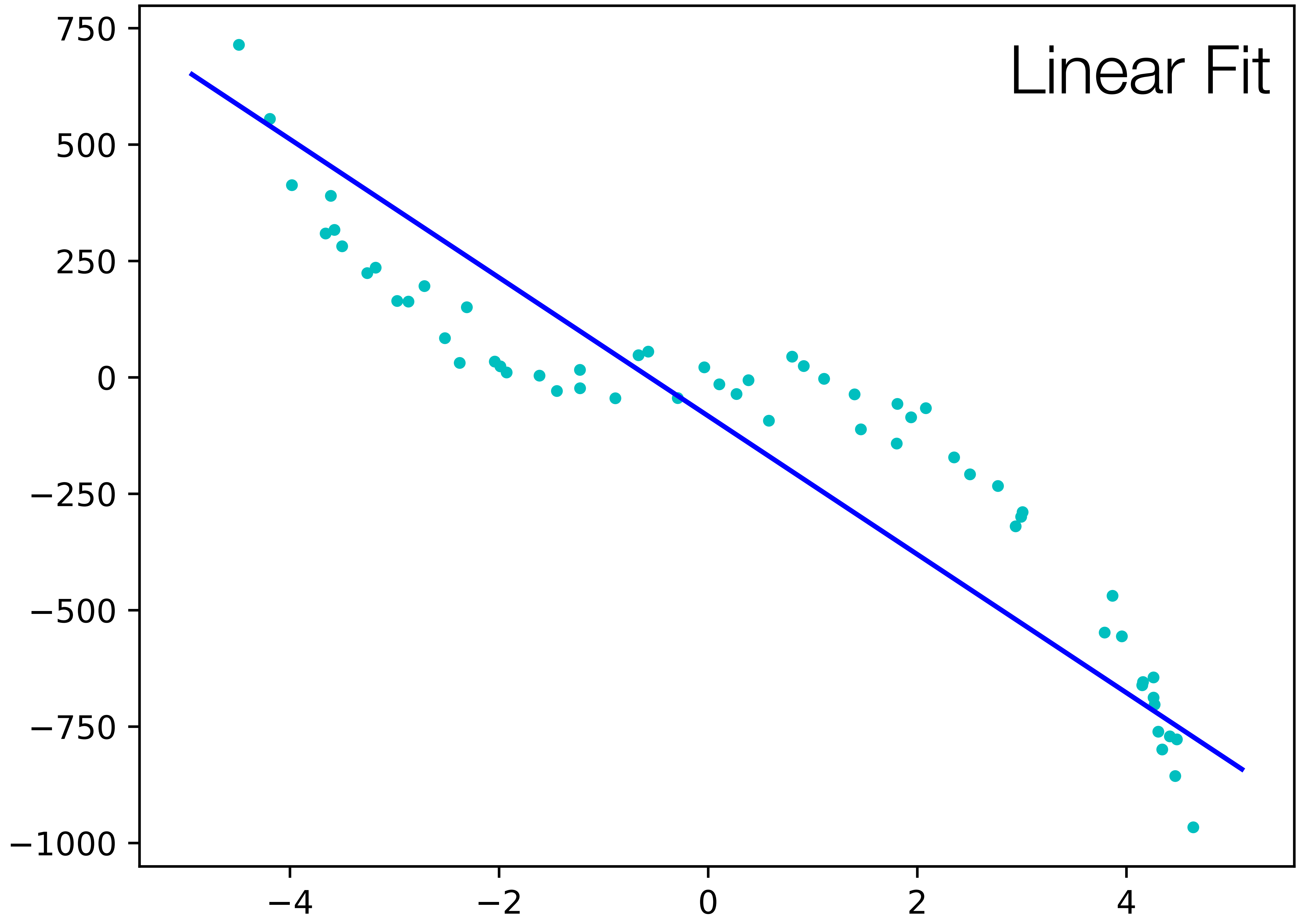


Test Data

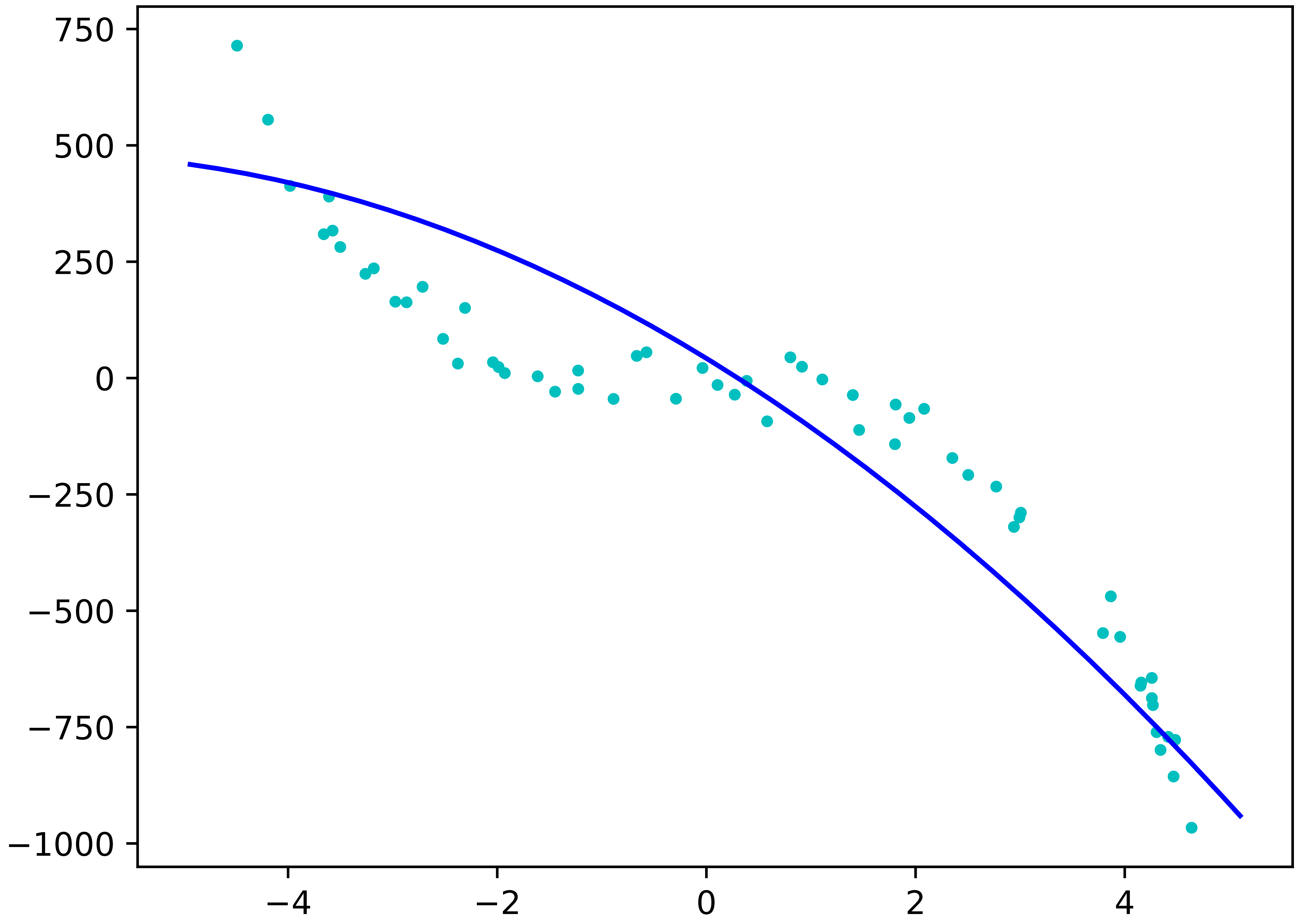
Received many more examples

$$\left\{(x_i, z_i)\right\}_{i=1}^{200}$$

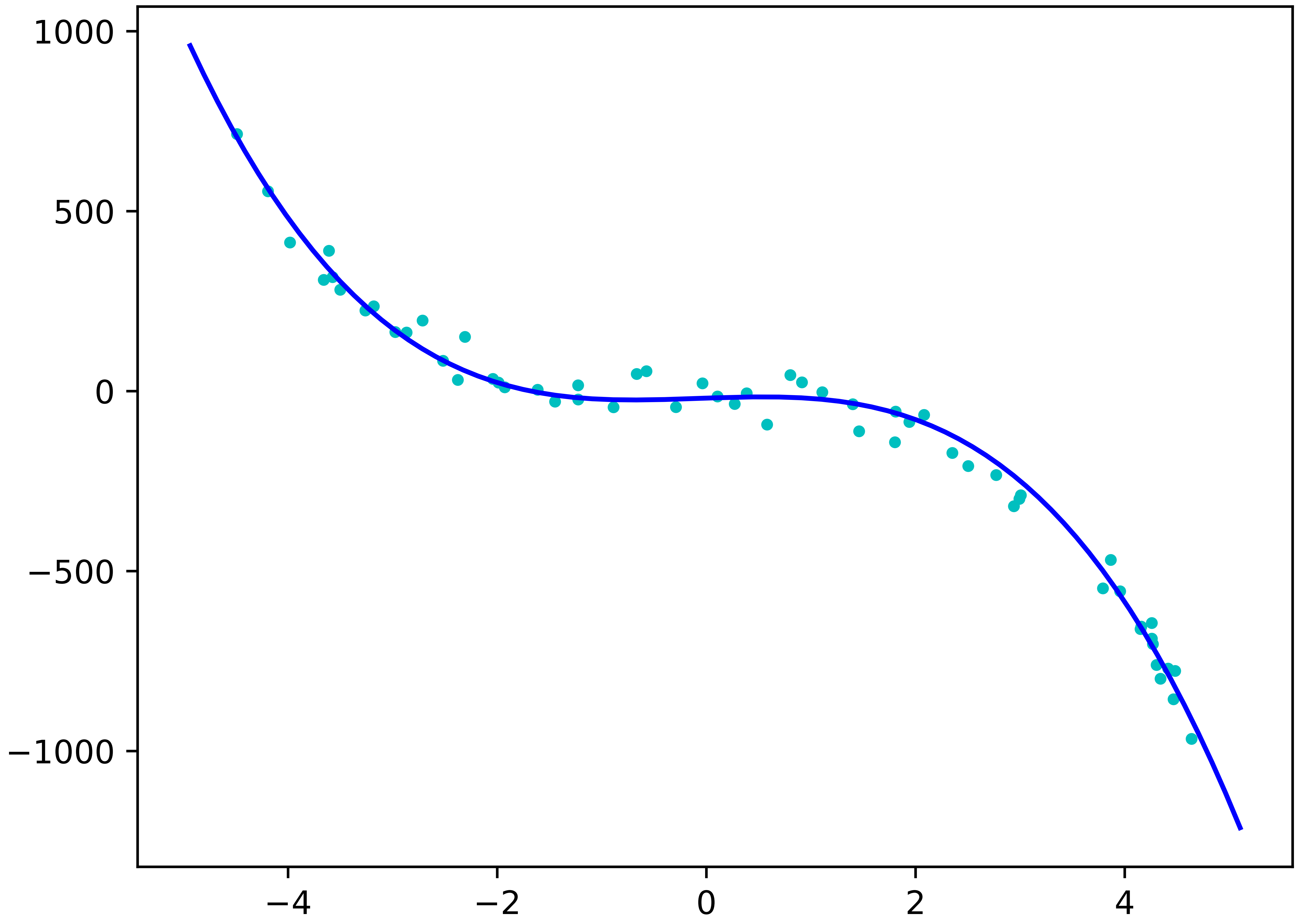
Tested fit on unseen (during training) examples



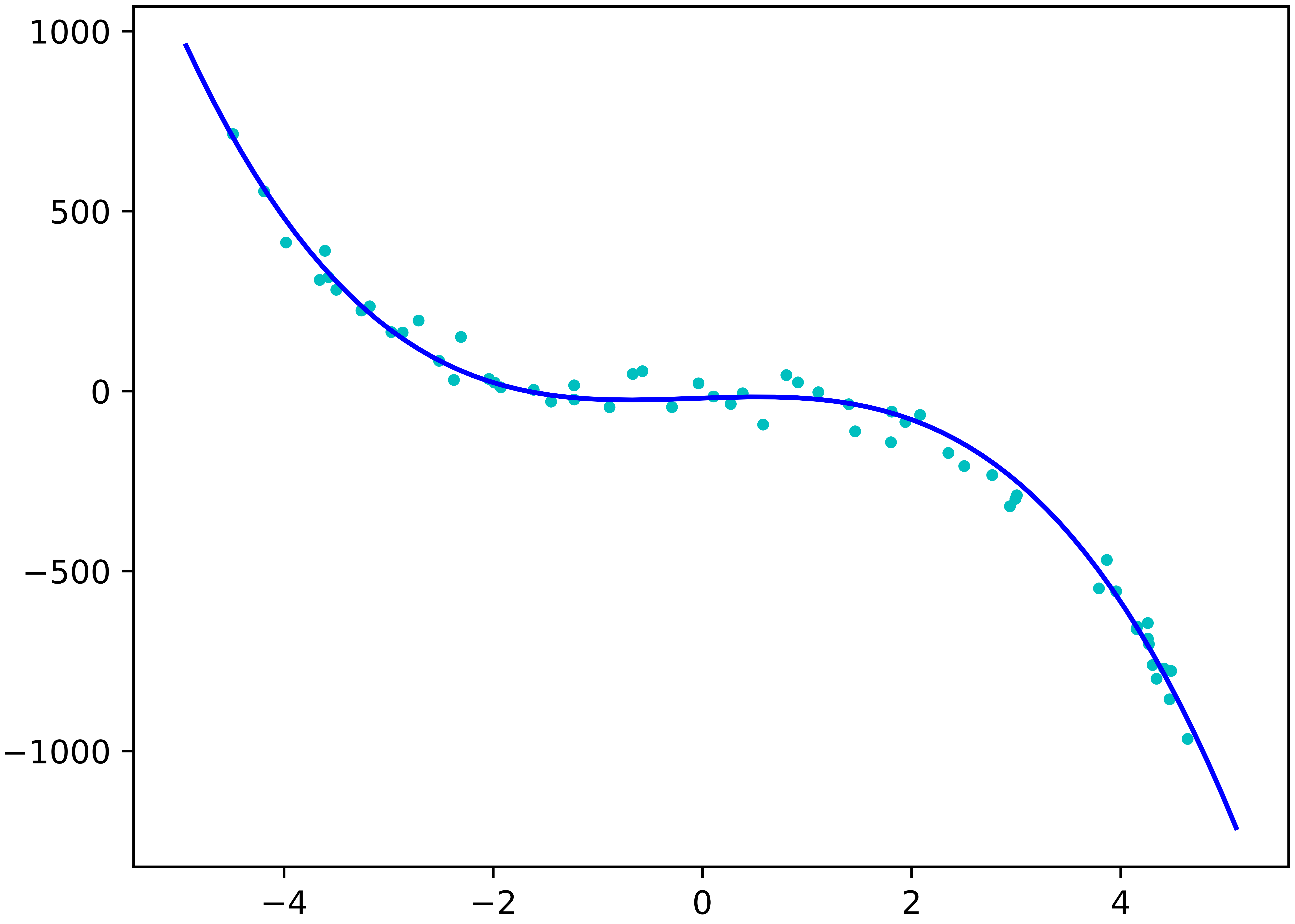
Degree 2 Fit to Test Data



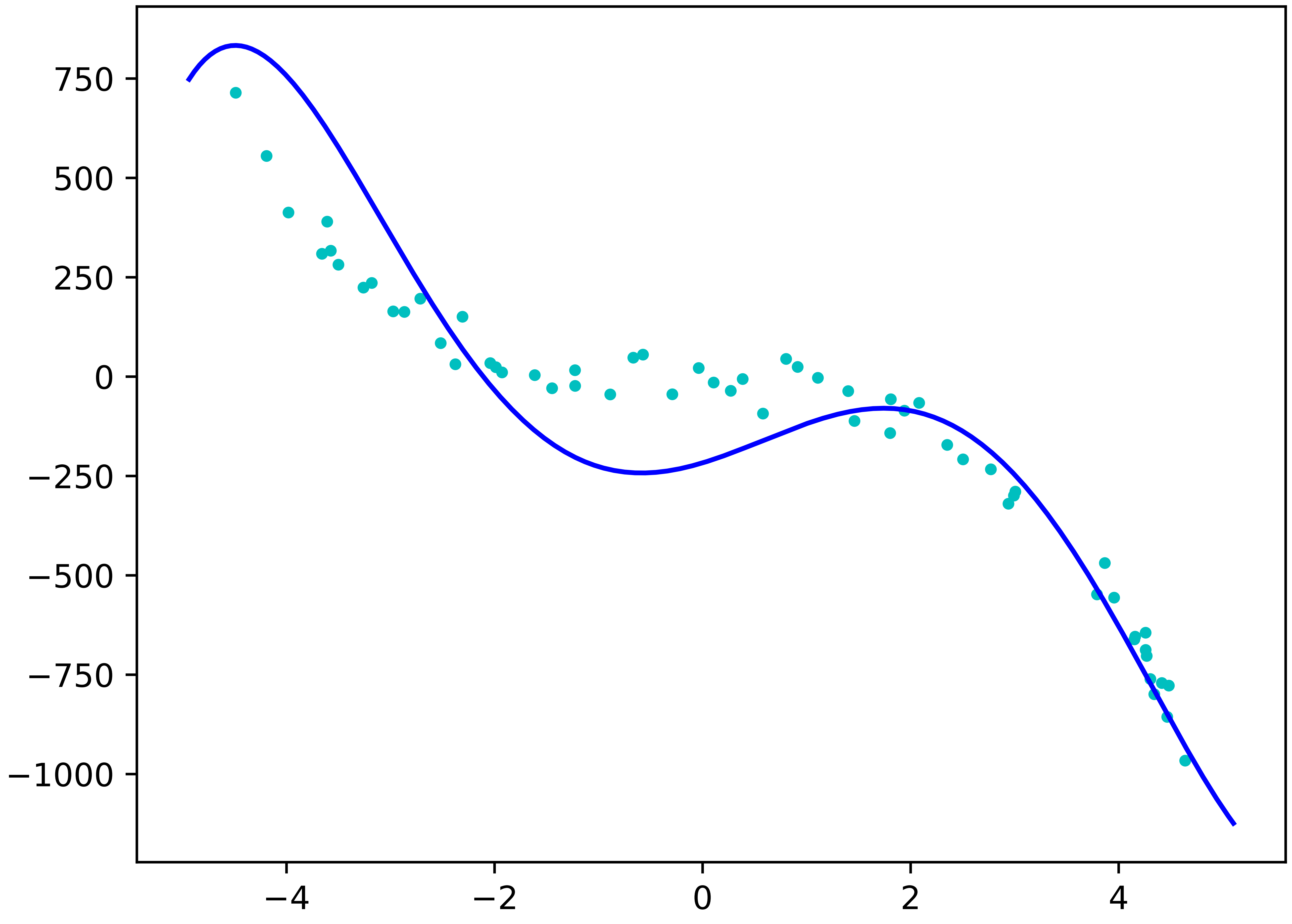
Degree 3 Fit to Test Data



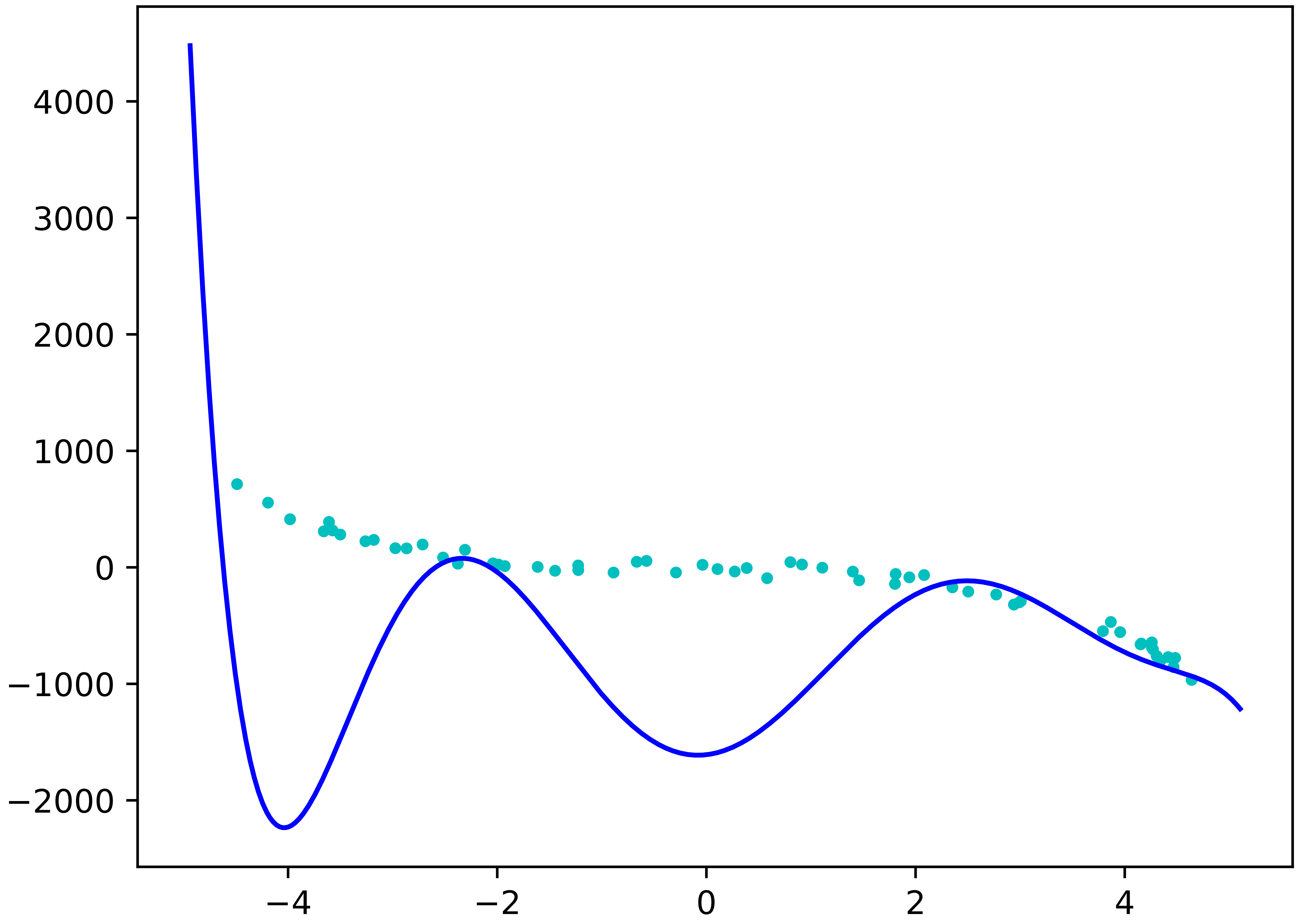
Degree 3 Fit to Test Data

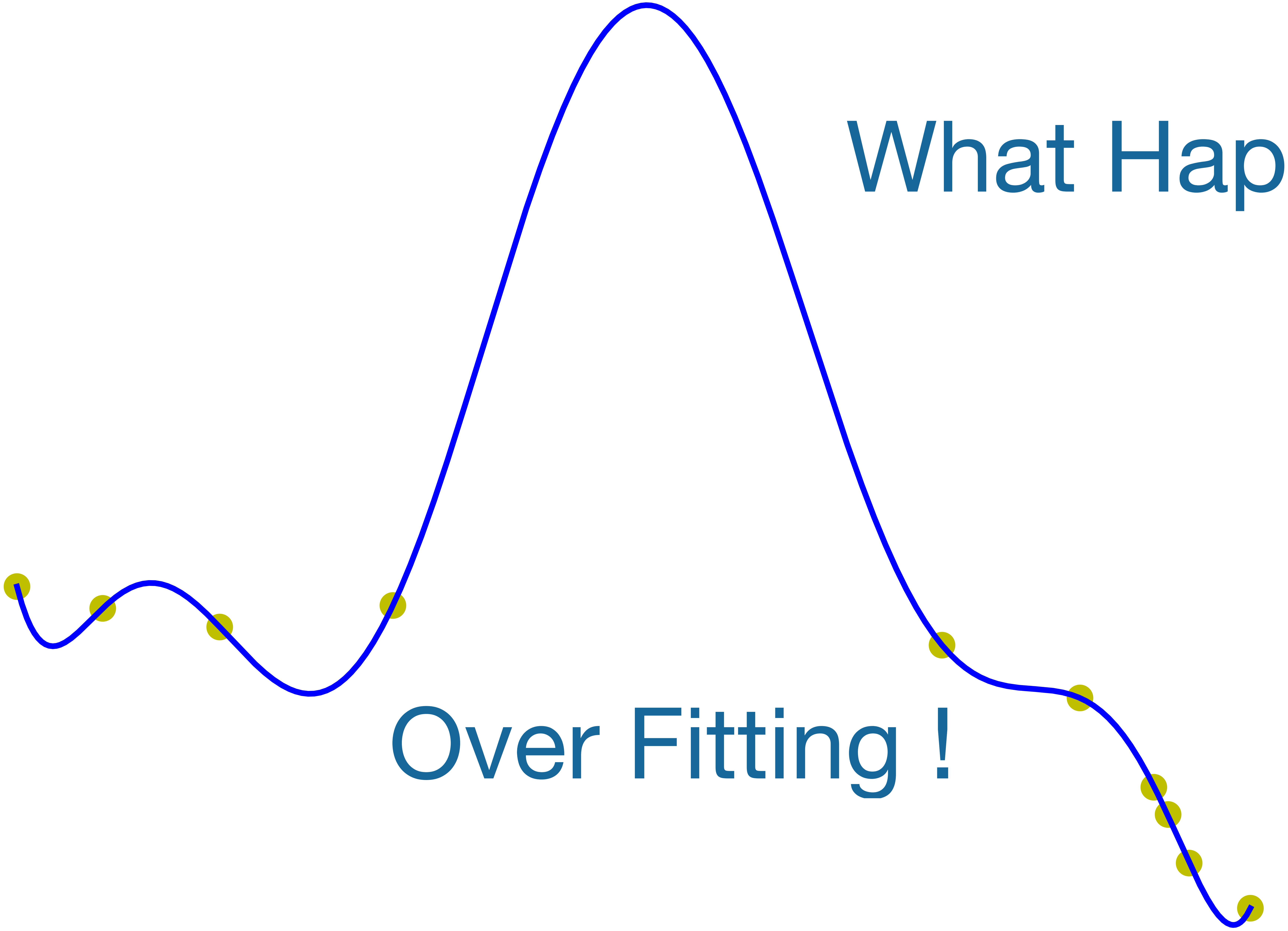


Degree 4 Fit to Test Data



Degree 7 Fit to Test Data

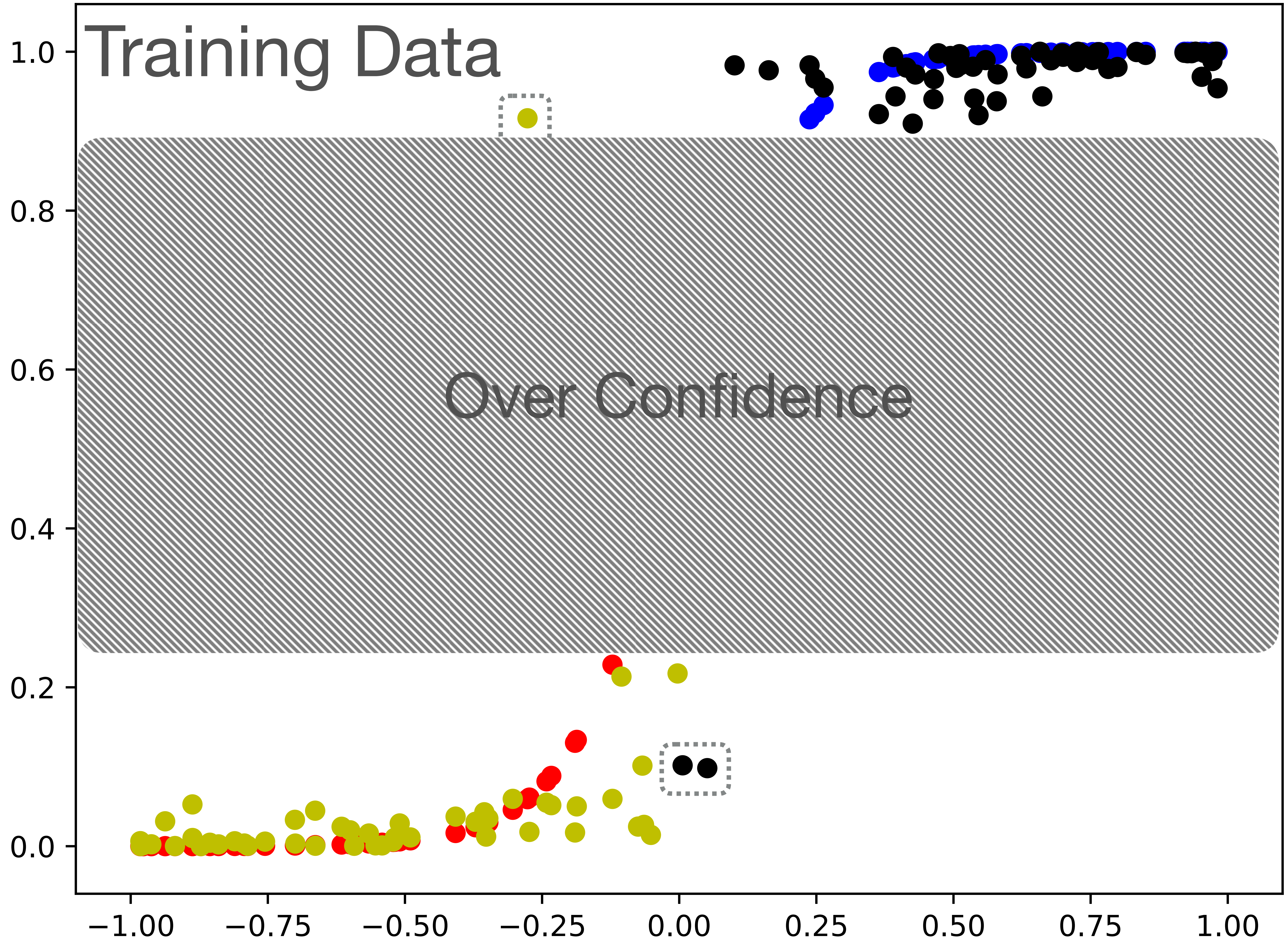




What Happened ?

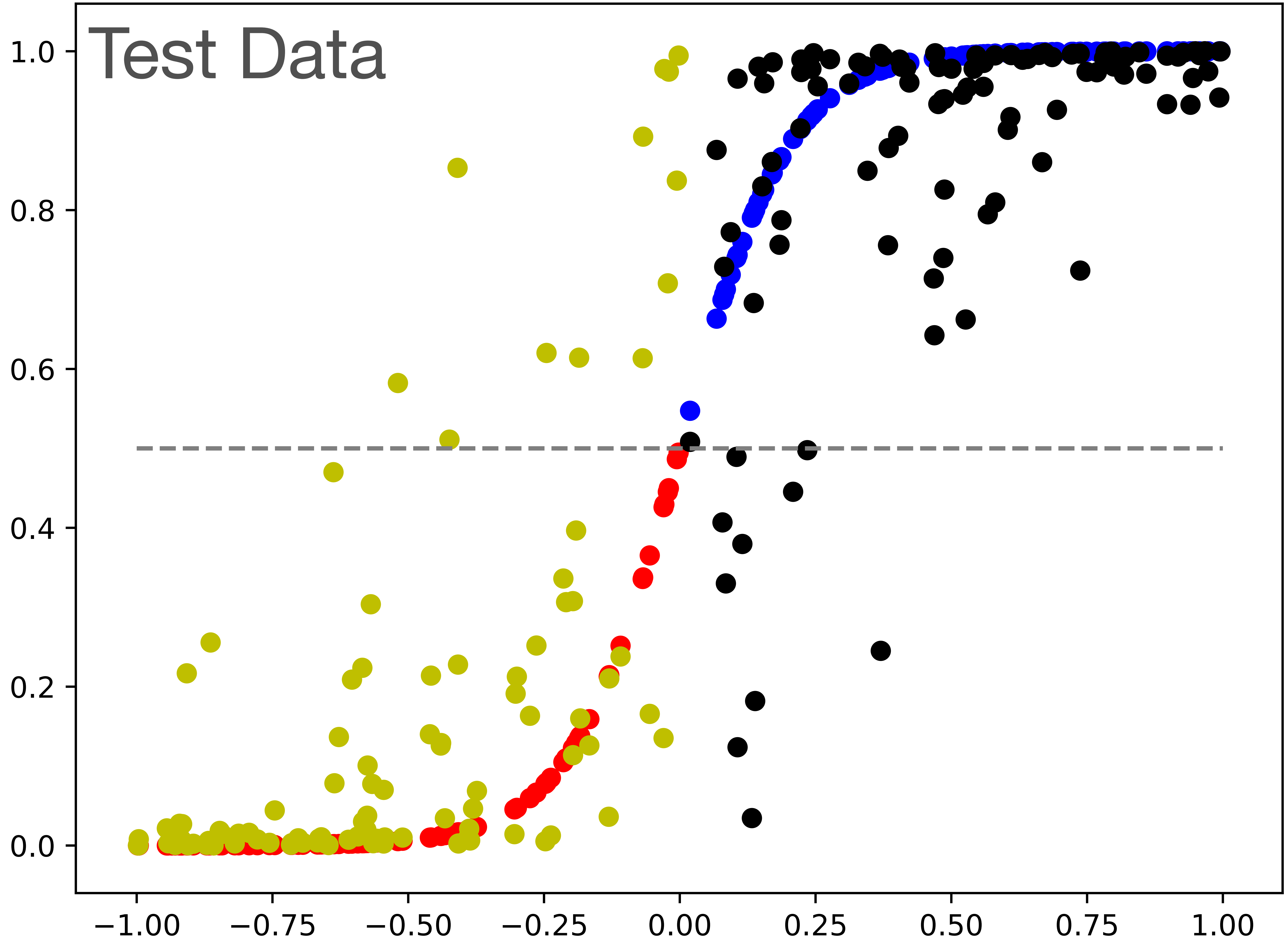
Over Fitting !

Overfitting in Logistic Regression



mislabelled examples

Overfitting in Logistic Regression



Overfitting in Classification

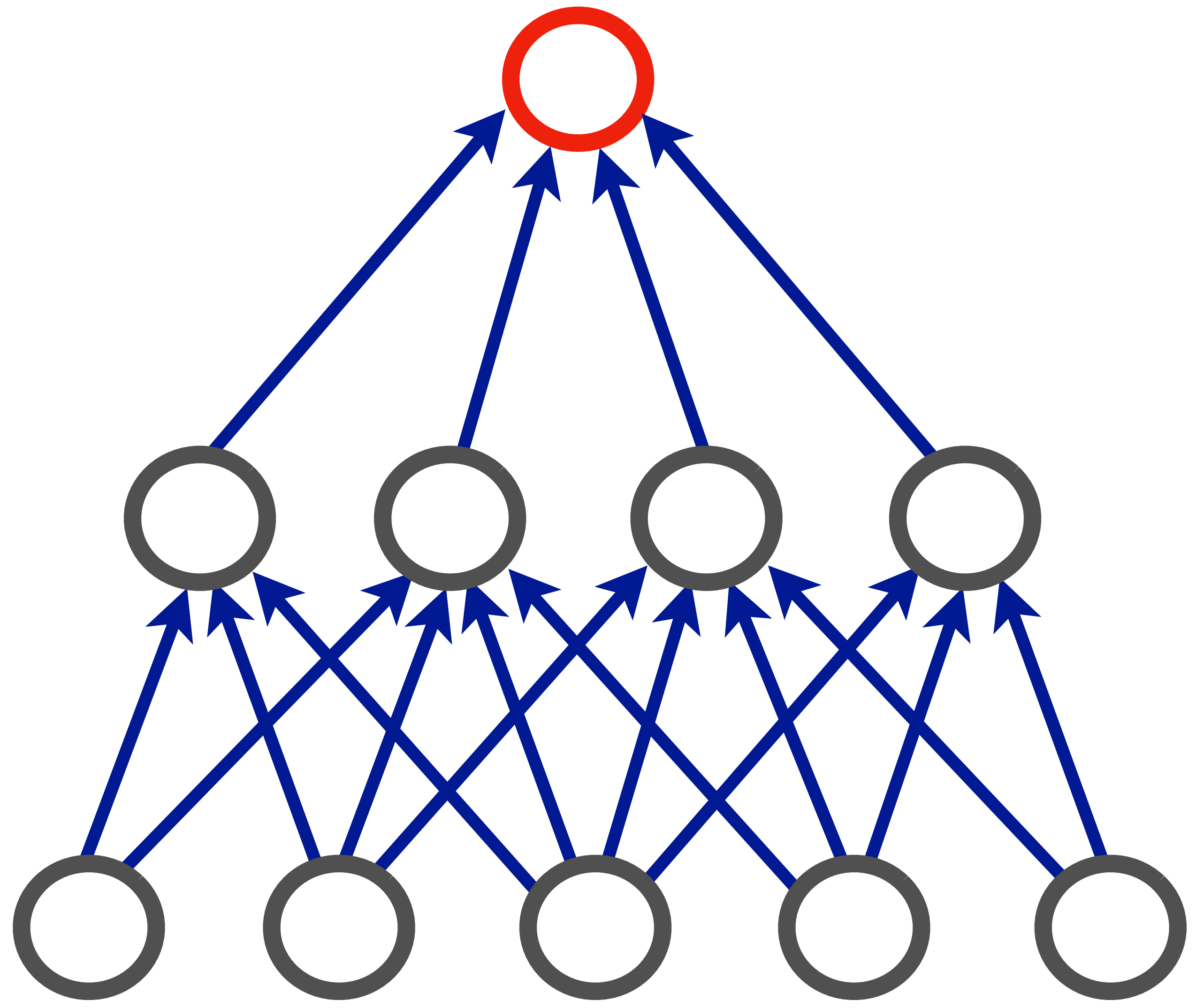
Trained a two-later NN on binary image classification

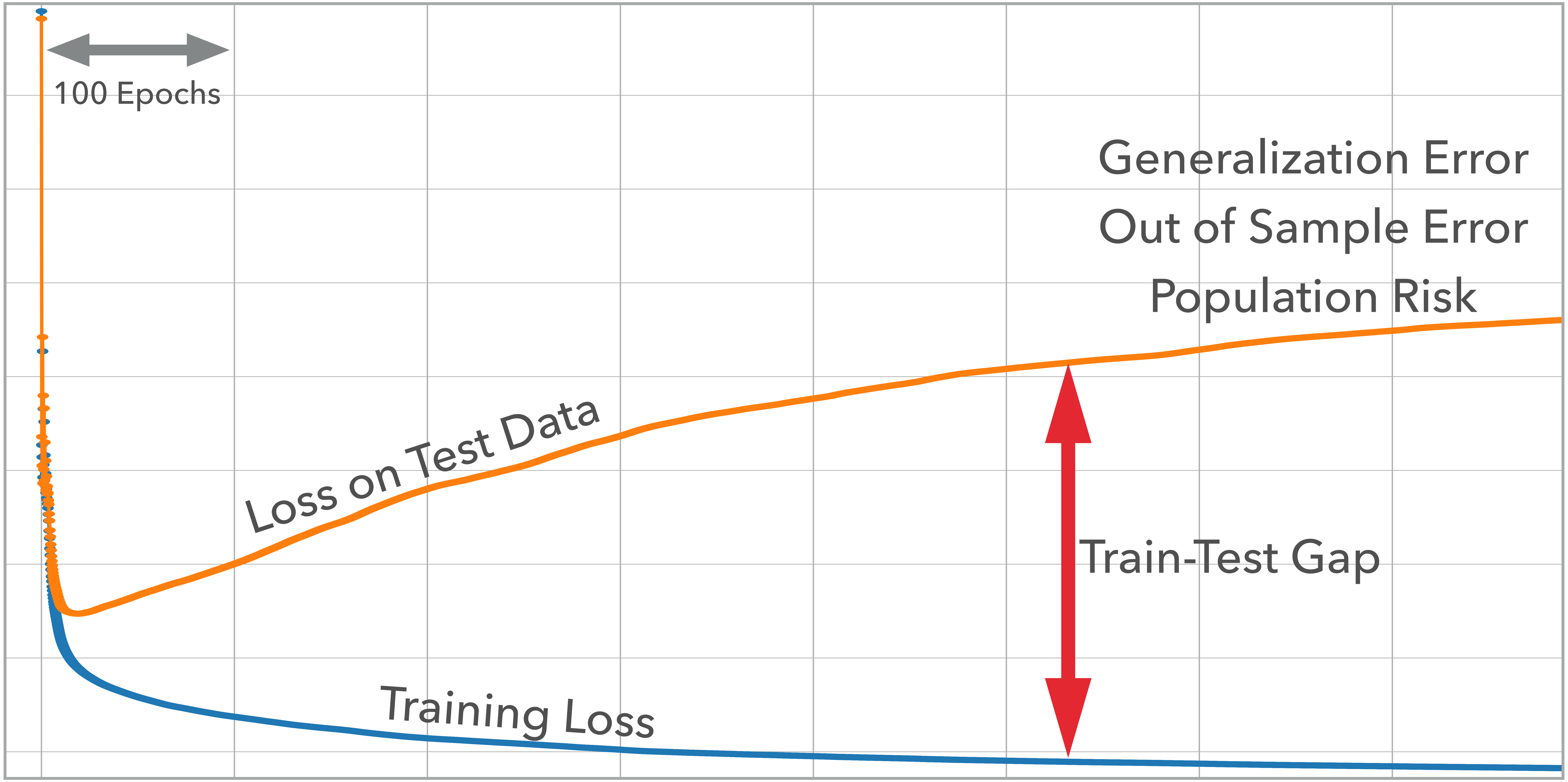
Thumbnail images: 10x10 (input dimension 100)

Dataset size 10,000

Hidden layer size: 20

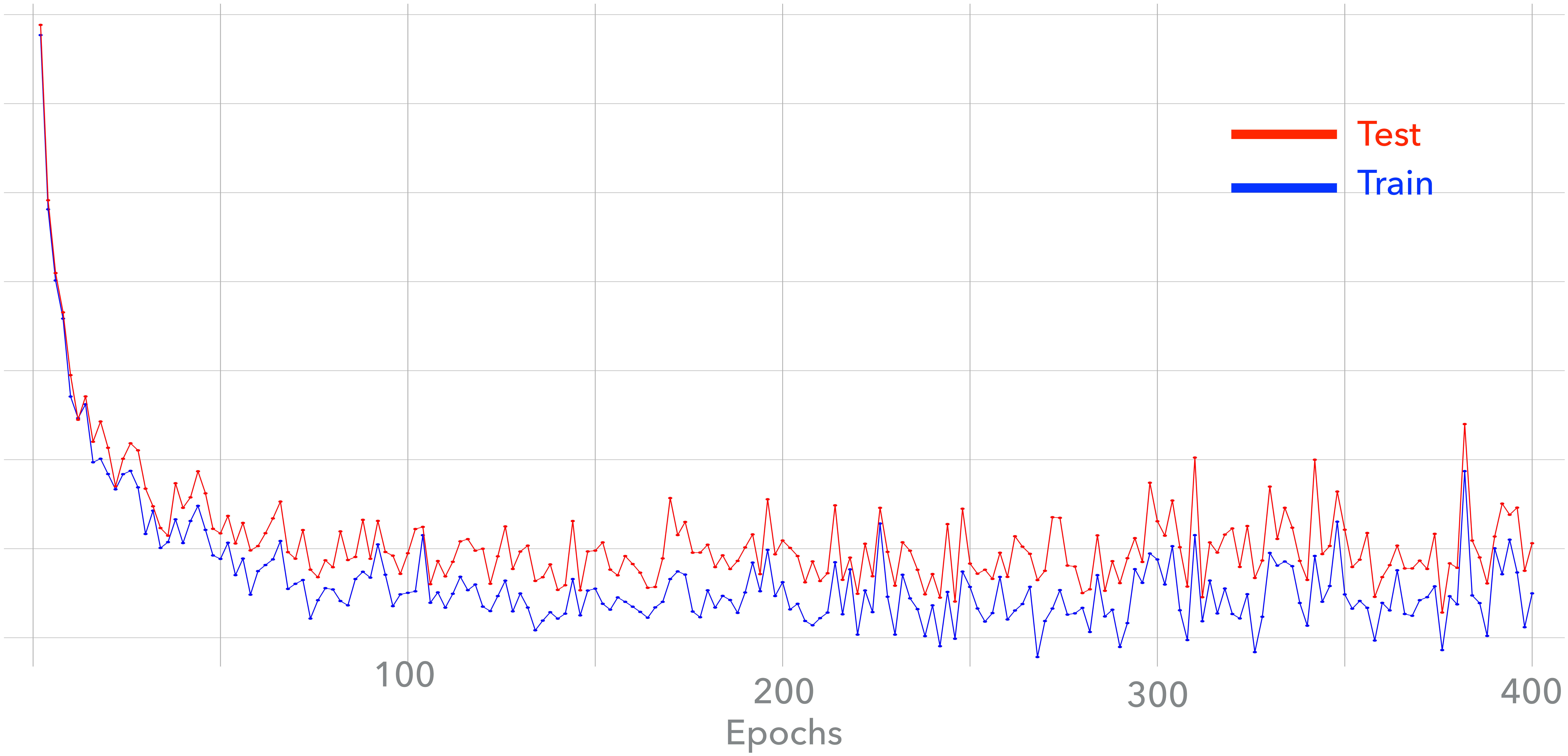
Tuned SGD well and ran for many iterations




100 Epochs
Generalization Error
Out of Sample Error
Population Risk
Loss on Test Data
Train-Test Gap
Training Loss

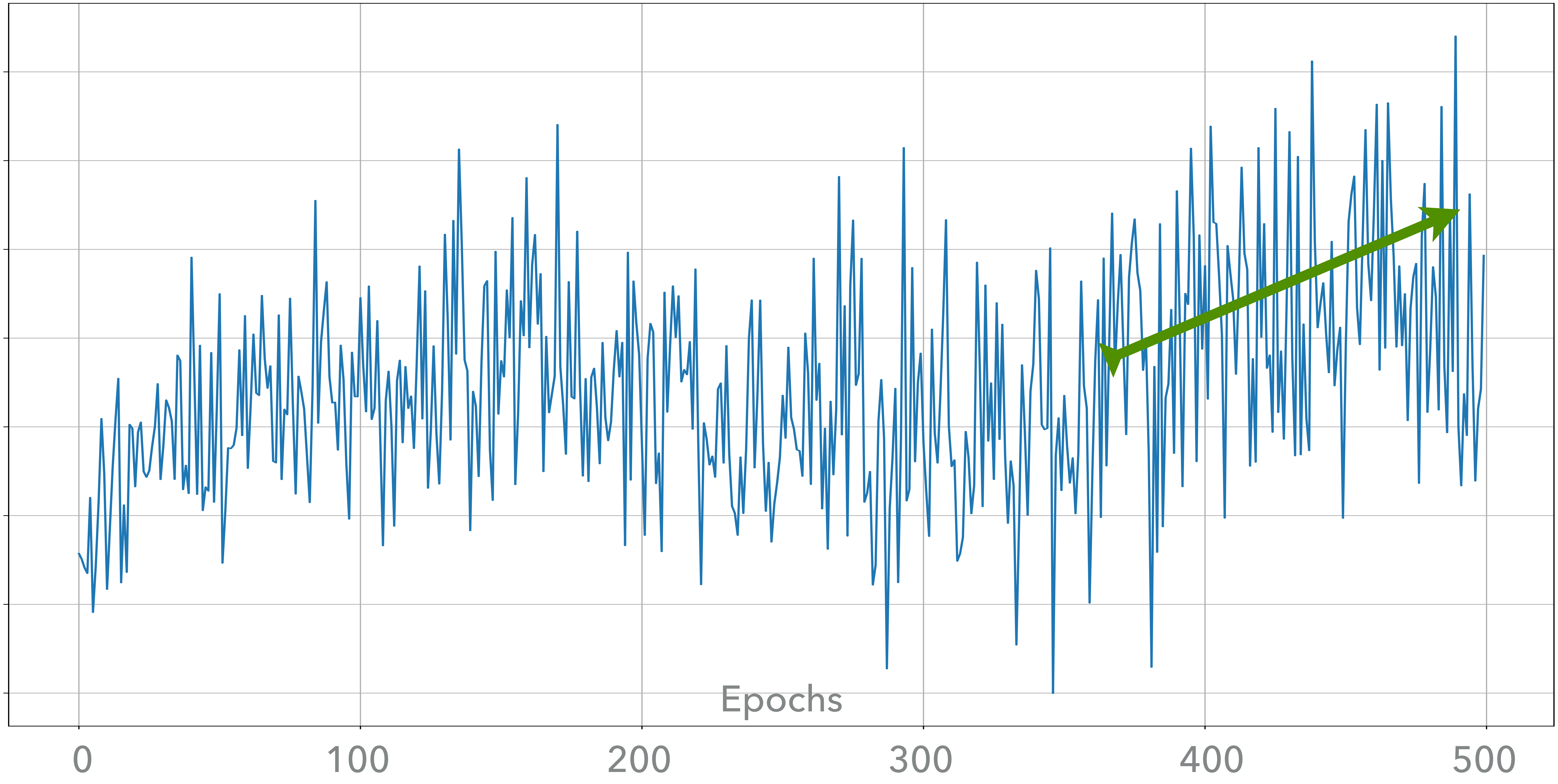
Early Stopping

- Use a validation set which is not used for training
- Check every **k** updates/epochs performance on validation set
- Once test-train gap is growing stop training
- Works well in practice when scheme is feasible
  - Requires three sets of examples: Train, Validation, Test
  - Loss of stochastic methods not monotone & gap not easy to monitor

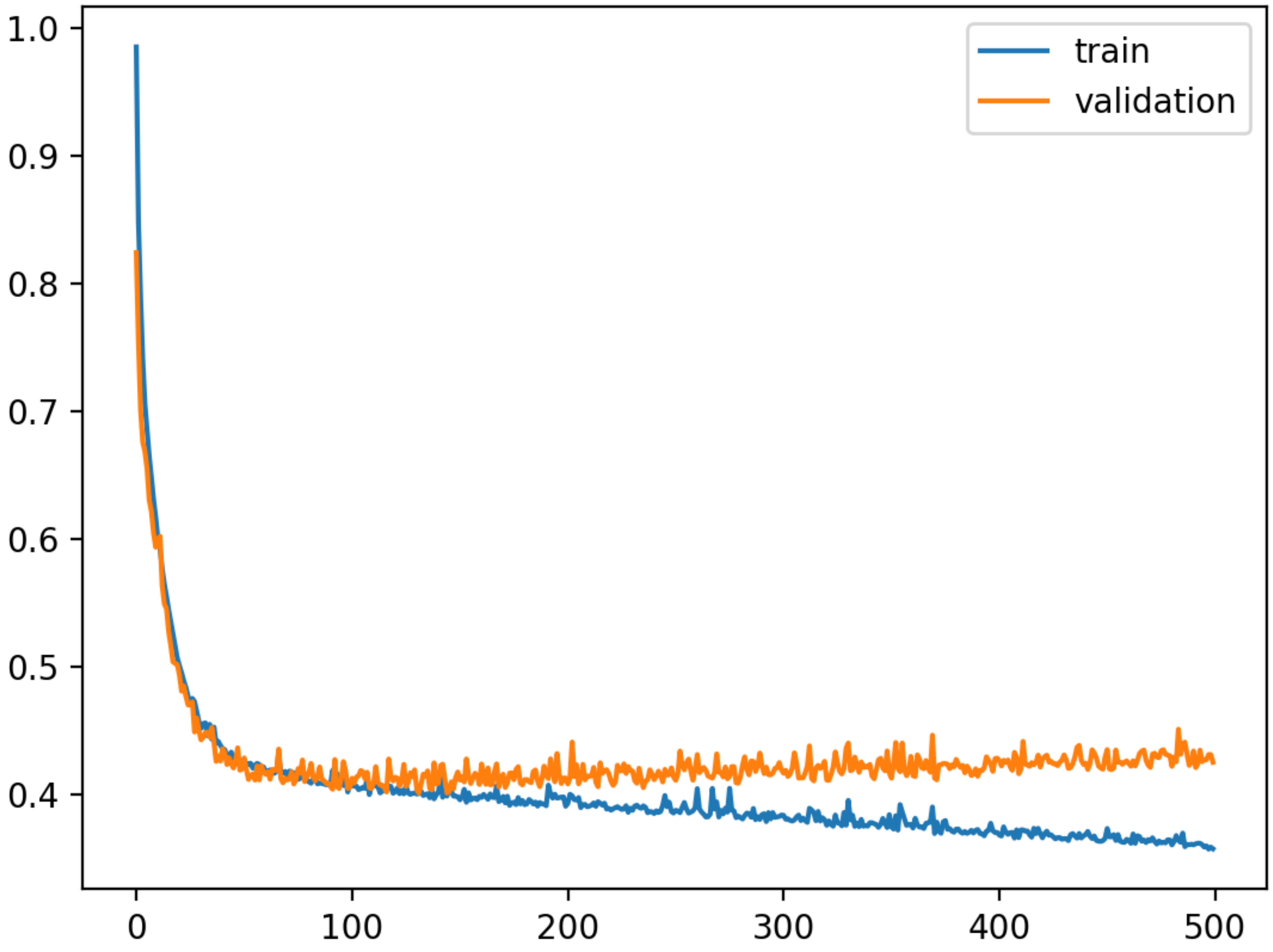


Test
Train
100
200
300
400
Epochs

Test-Train Gap



Train-Test Gap (large mini-batch)



Finite Case

Suppose we have only **k** predictors — no weight learning: $f_1, \ldots, f_k$

One has zero generalization error, rest have generalization error $\geq \epsilon$:

$$\exists j : \forall (\mathbf{x}, y) : \; f_j(\mathbf{x}) = f^\star(\mathbf{x}) = y \quad ; \quad \forall i \neq j : \mathbf{P}[f(\mathbf{x}) \neq y] \geq \epsilon$$

Received training set S with only **n** examples sampled independently

Evaluate errors on S: $\epsilon_i = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\left[f_i(\mathbf{x}) \neq y_i\right]$

Choose any f_i for which $\epsilon_j = 0$

Generalization: Finite Case

Probability that $\epsilon_i = 0$ is at most $(1-\epsilon)^n \leq e^{-\epsilon n}$ [independence of sample]

Probability α that $\exists i \neq j \ s.t. \ \epsilon_i = 0$ is at most $\alpha = (k-1)\, e^{-\epsilon n}$

If $\alpha \leq \dfrac{1}{k}$ it is unlikely we do not find correct predictor

This means that we need about $O\left(\dfrac{\log(d)}{\epsilon}\right)$ samples

I.I.D

- I.I.D: Identically Independently Distributed
- Generalization analysis typically assumes $\exists D$:

  unknown distribution $D(\mathbf{x}, y)$

- W.L.O.G assume $\mathbf{x} \in \{0,1\}^d \quad y \in \{-1,1\}$
- Identically [no dependence on i]:

$$\forall i \in S : \quad D((\mathbf{x}_i, y_i) = (\mathbf{a}, b)) \text{ is } D(\mathbf{a}, b)$$

- Independence:

$$D\Big((\mathbf{x}_i, y_i) = (\mathbf{a}, b) \,\wedge\, D(\mathbf{x}_j, y_j) = (\mathbf{a}', b')\Big) = D(\mathbf{a}, b)\, D(\mathbf{a}', b')$$

x_0	x_1	y	D(x,y)
0	0	-1	0.07
0	0	1	0.01
0	1	-1	0.03
…	…	…	…
…	…	…	…
1	1	1	0.005

"Continuous Case"

Find best model with weights $\mathbf{w} \in \mathbf{R}^{d}$

For bfloat16: each entry of $\mathbf{w}$ can take 2^{16} different values

Different vectors that can be represented $2^{16 \times d}$

Denote each weight vector as a predictor: $f_1, \ldots, f_{2^{16d}}$

If $\exists \mathbf{w}^\star$ where $f_{\mathbf{w}^\star}(\mathbf{x}) = y$ for all $\mathbf{x}, y$ with $D(\mathbf{x}, y) > 0$ (perfect generalization):

it would take only $\tilde{O}(d)$ examples to find it !

Caveats?

- $\tilde{O}(d)$ hides pretty bad constants
- Time of finding $\mathbf{w}^\star$ is exponential in **d**