Metrics & regularization

Covered chapters: Prince Ch 8-9

Miles Cranmer

Expectations

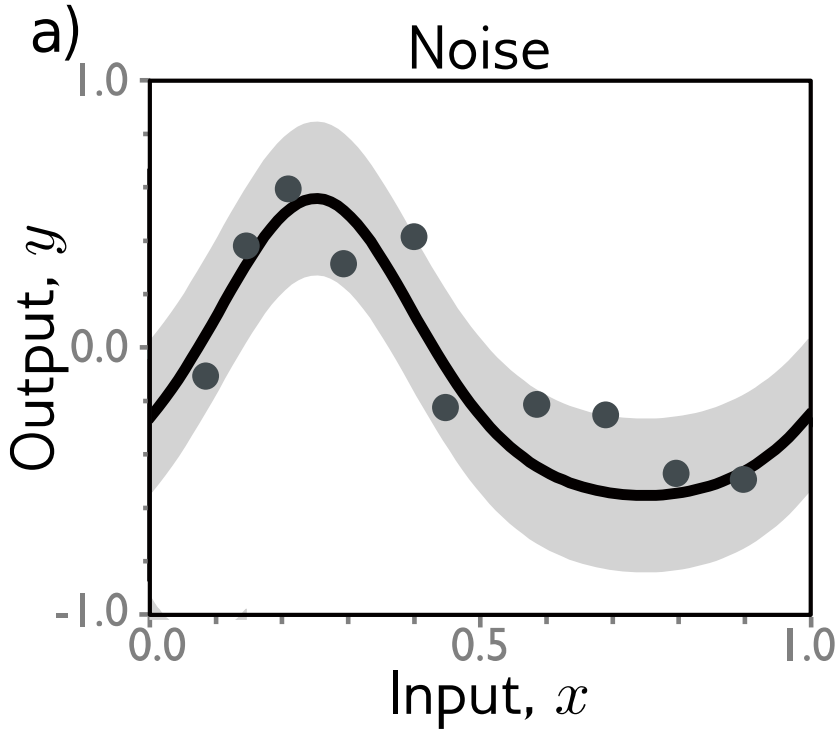
$$\mathbb{E}\Big[\mathrm{g}[x]\Big] = \int \mathrm{g}[x] Pr(x) dx,$$

Interpretation: what is the average value of g[x] when taking into account the probability of x?

Could approximate, by sampling many values of x from the distribution, calculating g[x], and taking average:

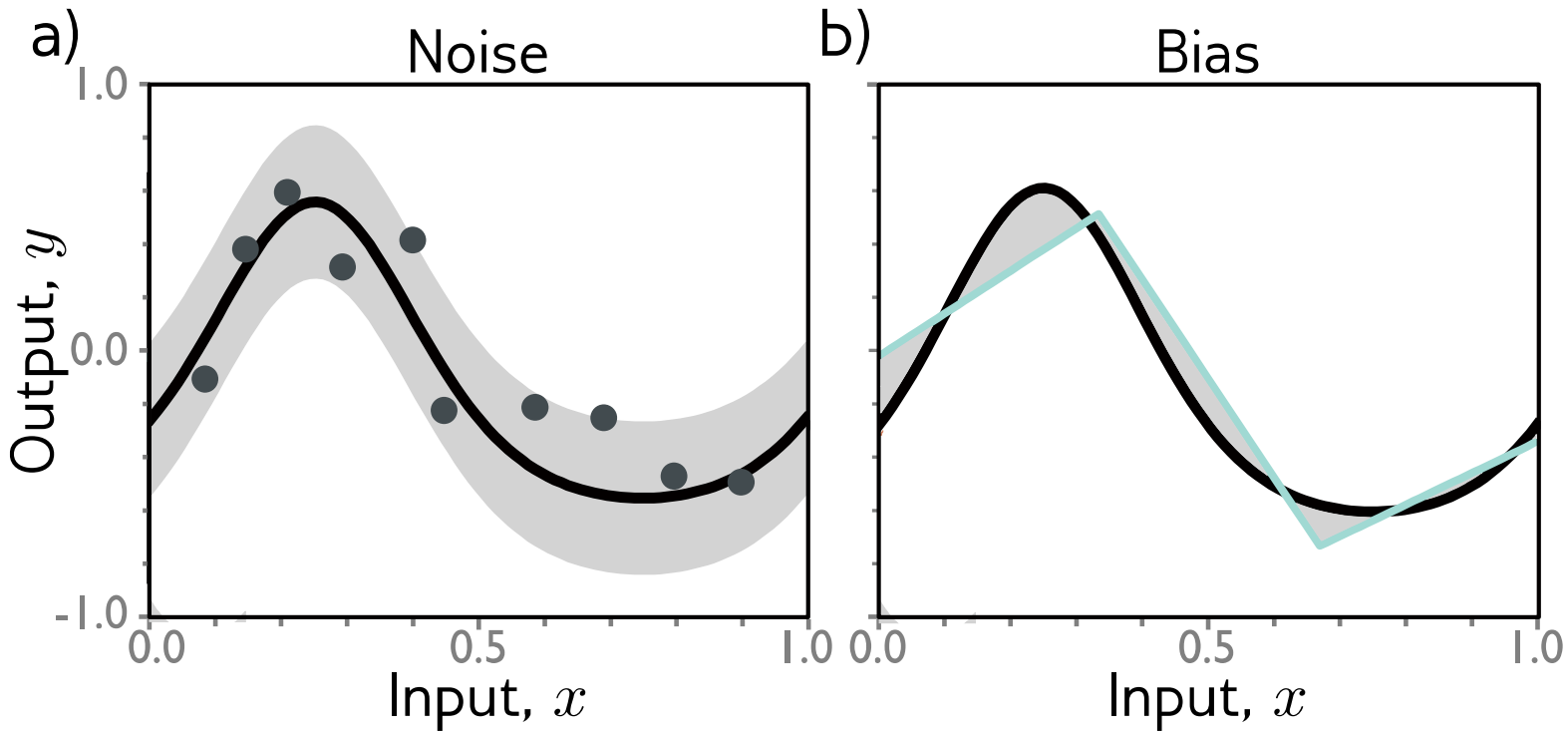
$$\mathbb{E}\Big[\mathrm{g}[x]\Big] \approx \frac{1}{N} \sum_{n=1}^{N} \mathrm{g}[x_n^*] \qquad \text{where} \qquad x_n^* \sim Pr(x)$$

Noise, bias, and variance

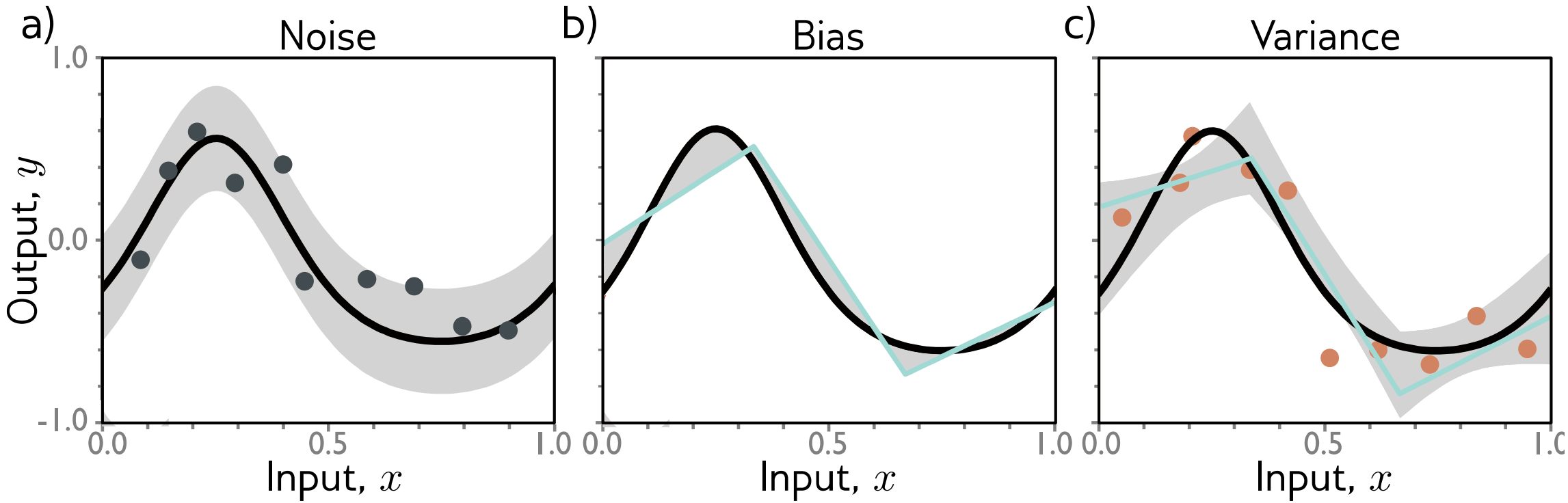


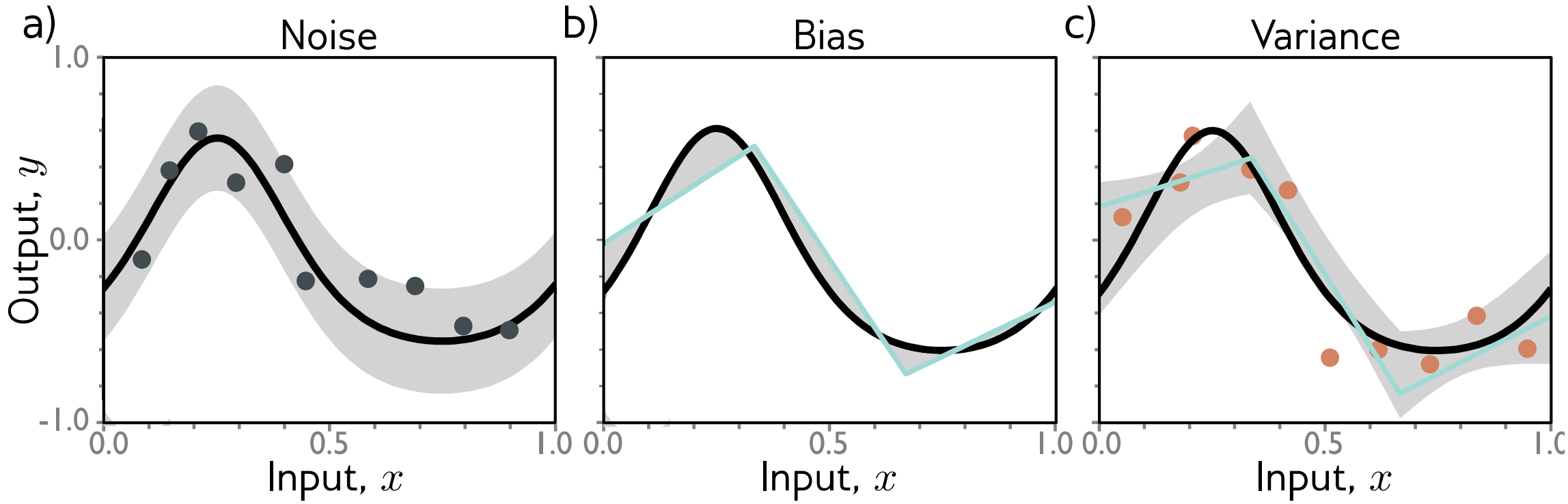
- Noise in measurements
- Some variables not observed
- Data mislabeled

Noise, bias, and variance



Noise, bias, and variance





$$\text{(Total error)} = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(\mathrm{f}[x, \boldsymbol{\phi}[\mathcal{D}]] - f_{\mu}[x]\right)^2\right]}_{\text{variance}} + \underbrace{\left(f_{\mu}[x] - \mu[x]\right)^2}_{\text{bias}} + \underbrace{\sigma^2}_{\text{noise}}$$

Noise, bias, and variance

Noise, bias, and variance

- Variance is the uncertainty in fitted model due to choice of **training set**

Noise, bias, and variance

- Variance is the uncertainty in fitted model due to choice of **training set**
- Bias is systematic deviation from the mean of the function we are modeling due to **limitations in our model**

Noise, bias, and variance

- Variance is the uncertainty in fitted model due to choice of **training set**
- Bias is systematic deviation from the mean of the function we are modeling due to **limitations in our model**
- Noise is **inherent uncertainty** in the true mapping from input to output

Least squares regression only

$$L[x] = \left(\mathrm{f}[x, \boldsymbol{\phi}] - y[x]\right)^2$$

- We can show that:

$$\mathbb{E}_{\mathcal{D}}\Big[\mathbb{E}_{y}[L[x]]\Big] = \underbrace{\mathbb{E}_{\mathcal{D}}\Big[\left(\mathrm{f}[x, \boldsymbol{\phi}[\mathcal{D}]] - f_{\mu}[x]\right)^2\Big]}_{\text{variance}} + \underbrace{\left(f_{\mu}[x] - \mu[x]\right)^2}_{\text{bias}} + \underbrace{\sigma^2}_{\text{noise}}$$

Expectation over noise in training data

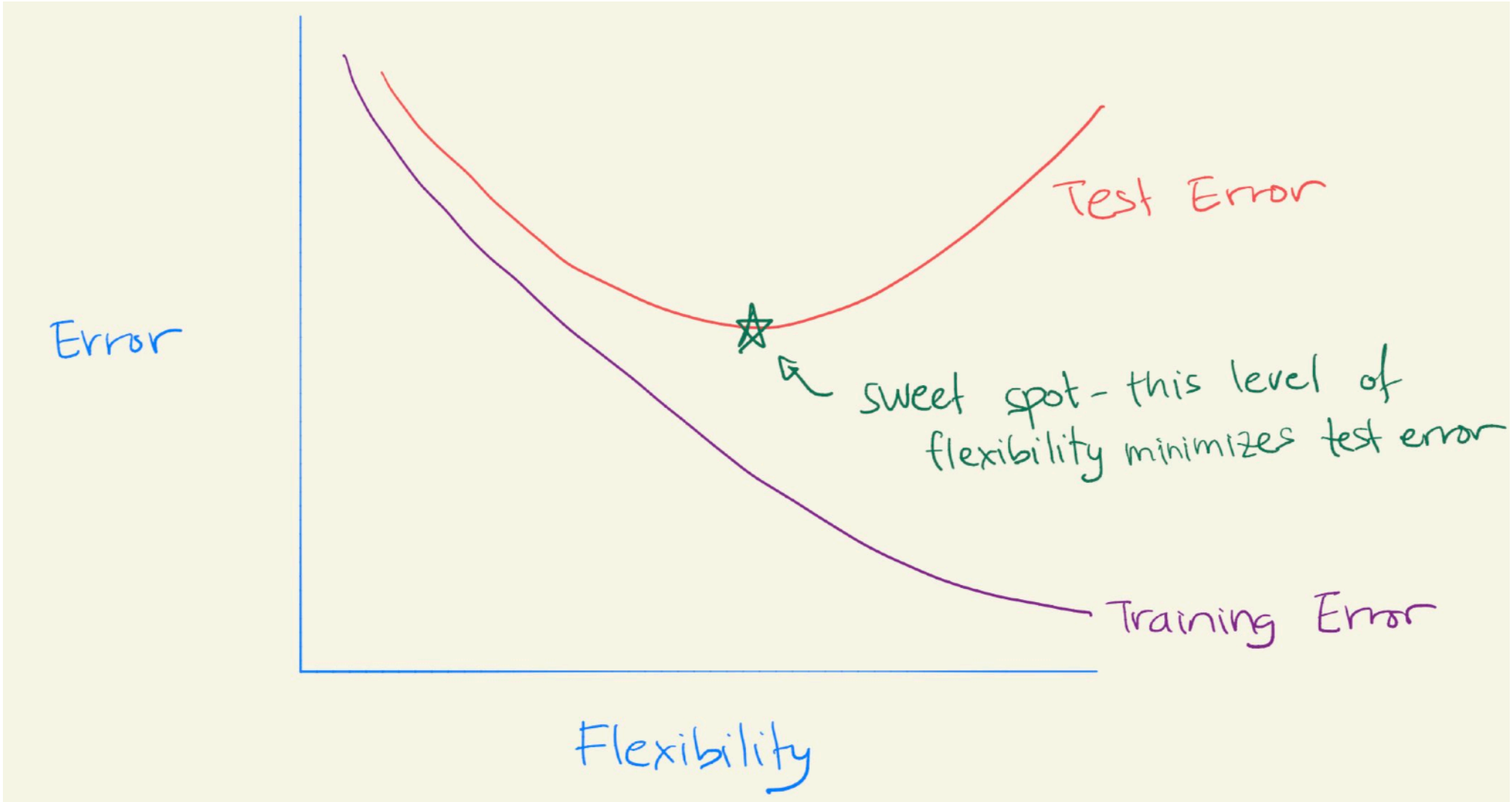
Expectation over noise in test data

Actual model

Best possible model if we had infinite data

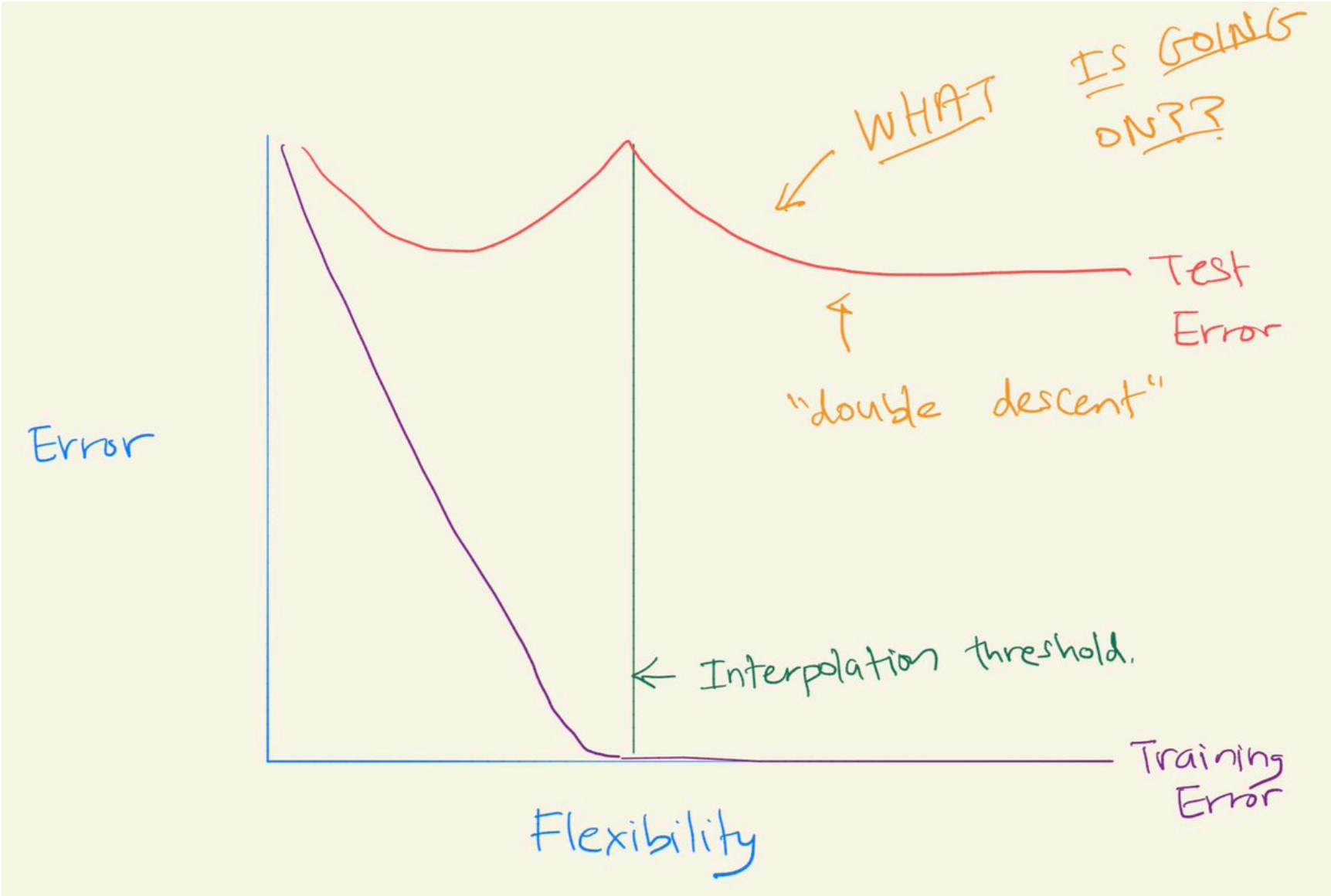
True function

Double descent

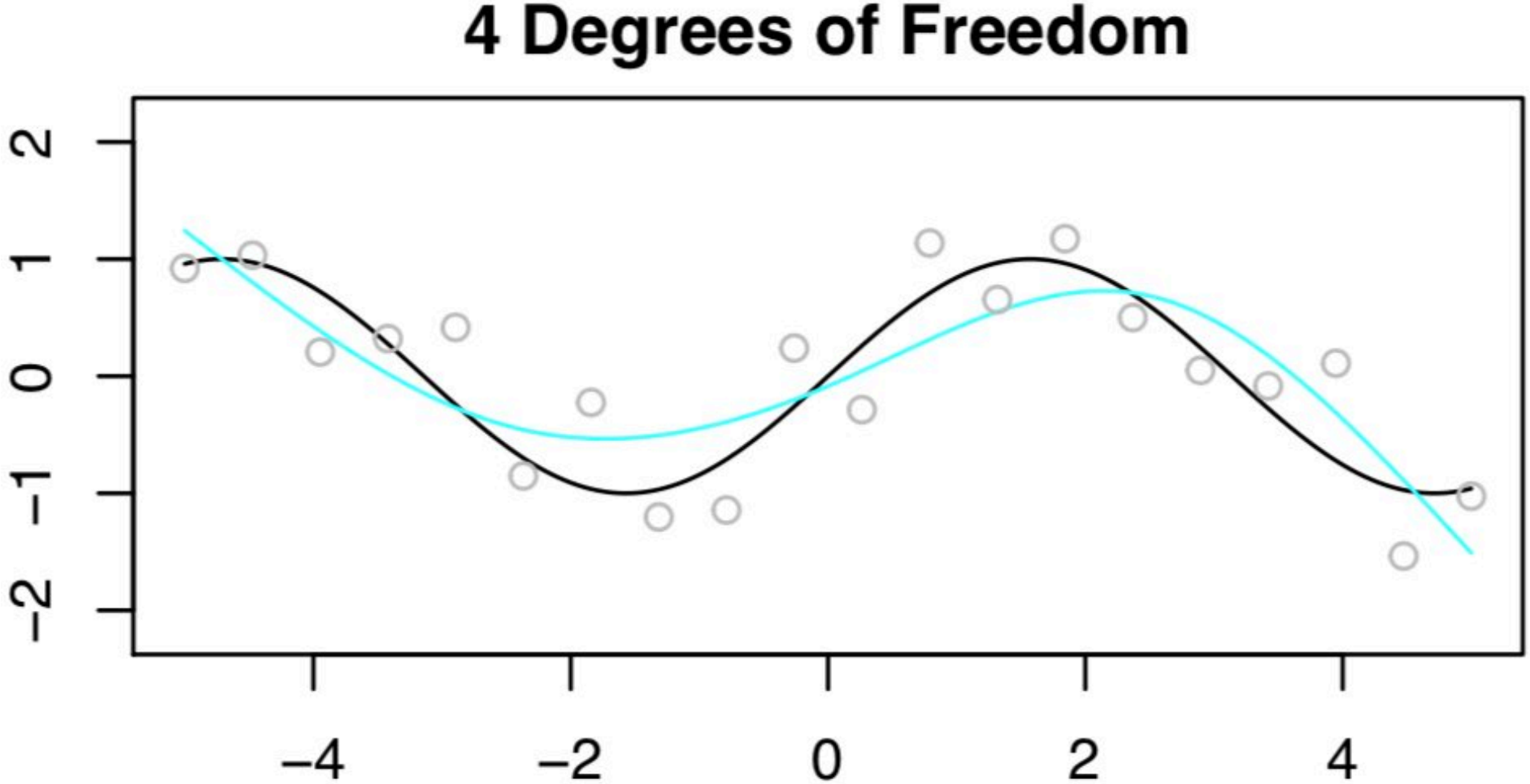


Credit: Daniela Witten

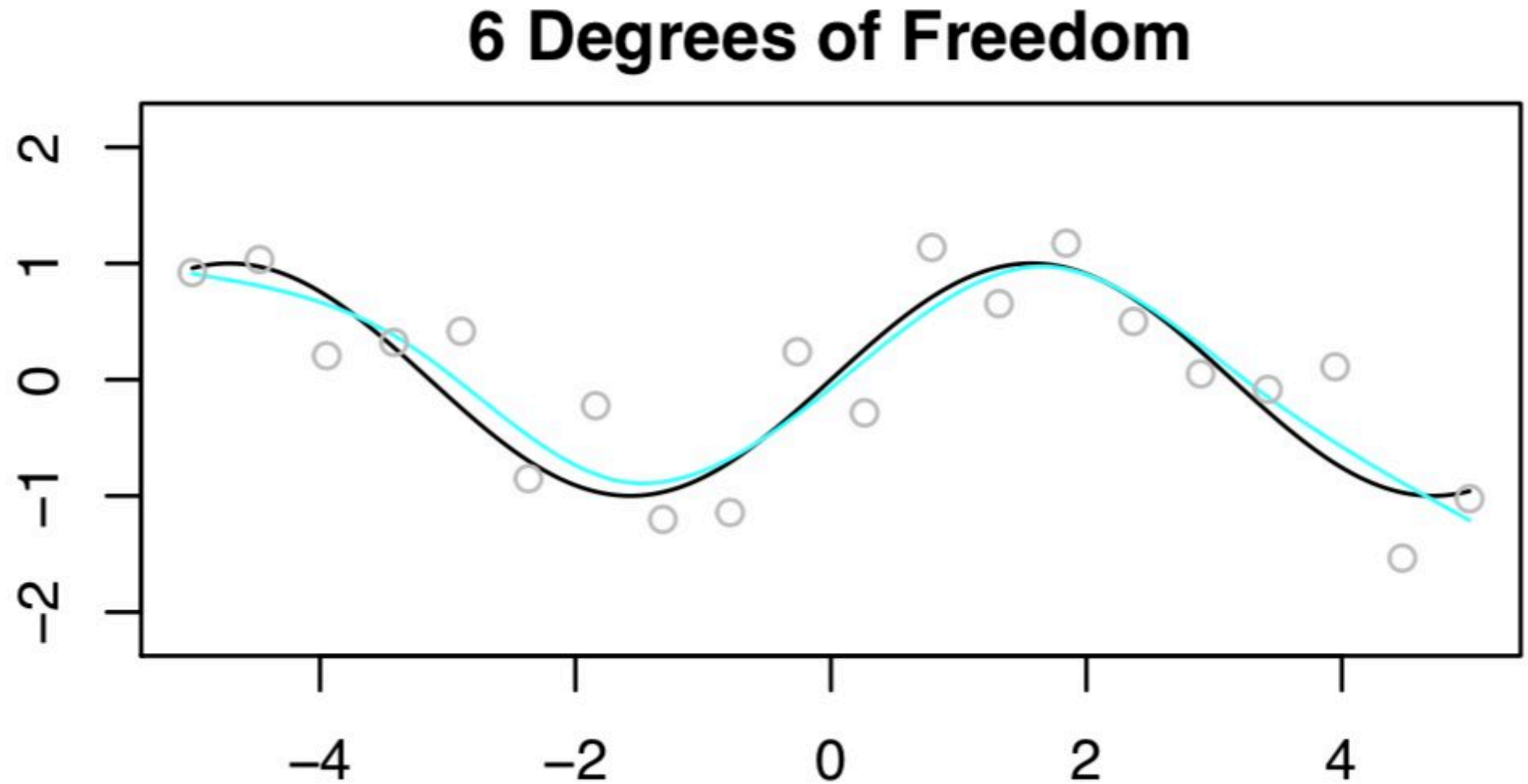
More detail – double descent



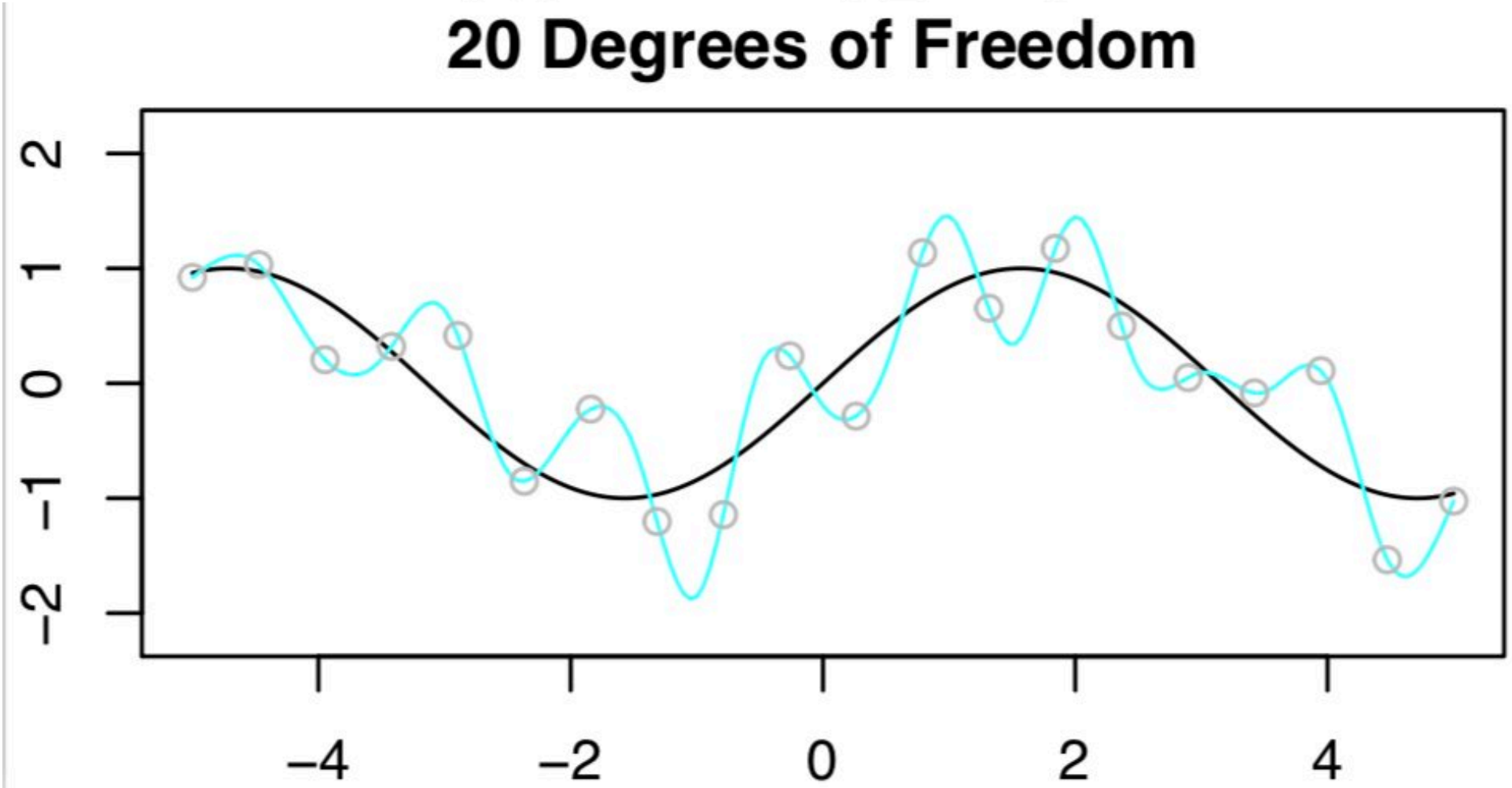
Credit: Daniela Witten



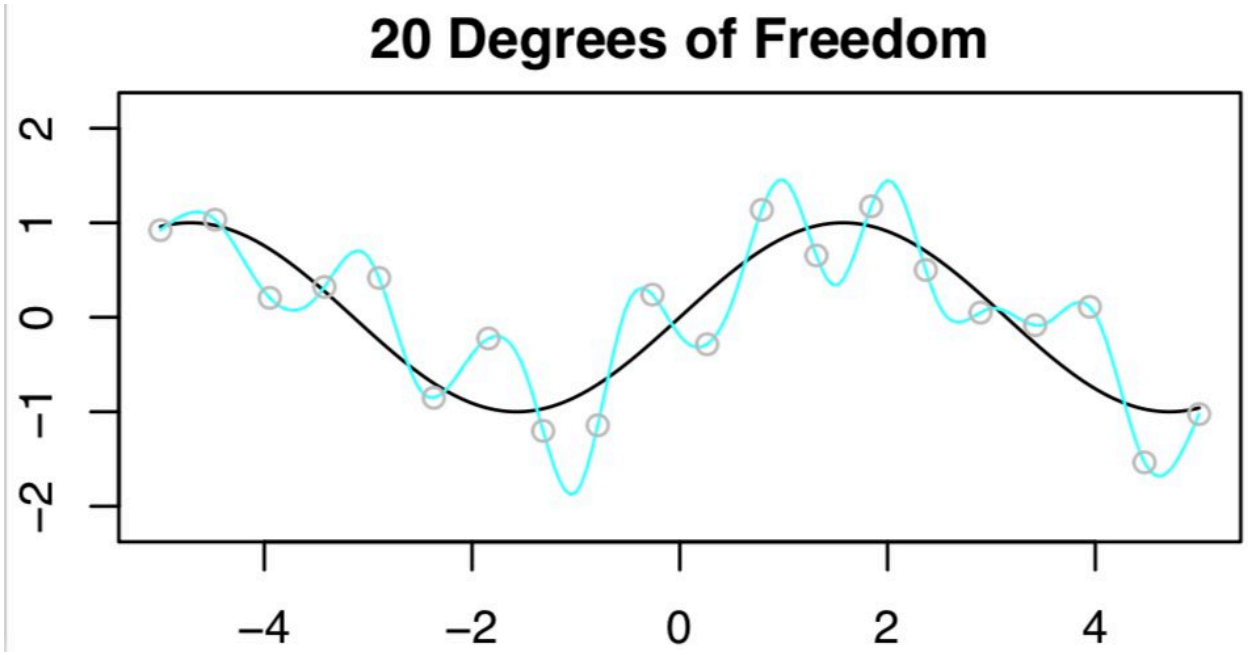
Credit: Daniela Witten

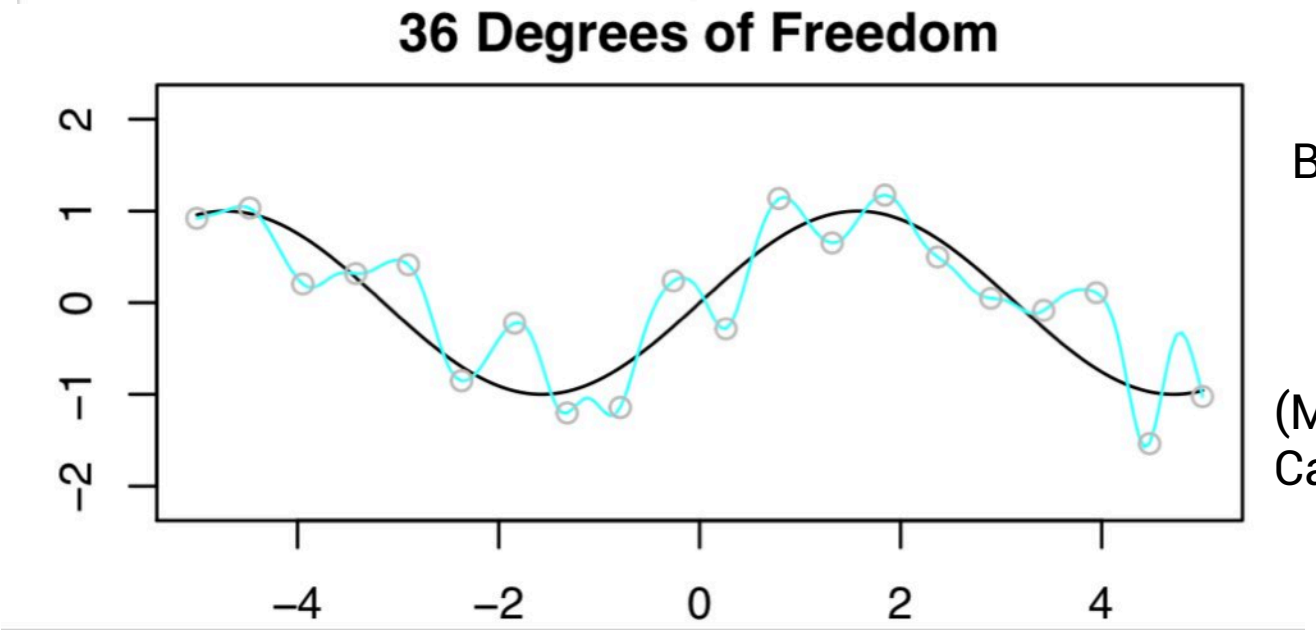


Credit: Daniela Witten



Credit: Daniela Witten





Better fit!

Important: requires that we regularize coefficients in polynomial!

(Much more extreme in Neural Networks! Can actually beat the first descent.)

Credit: Daniela Witten

Choosing hyperparameters

- Don't know bias or variance
- Don't know how much capacity to add
- How do we choose capacity in practice?
  - Or model structure
  - Or training algorithm
  - Or learning rate
- Third data set – validation set
  - Train models with different hyperparameters on training set
  - Choose best hyperparameters with validation set
  - Test once with test set

Regularization

- Why is there a generalization gap between training and test data?
  - Overfitting (model describes statistical peculiarities)
  - Model unconstrained in areas where there are no training examples
- Regularization = methods to reduce the generalization gap
- Technically means adding terms to loss function
- But colloquially means any method (hack) to reduce gap

Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Explicit regularization

- Standard loss function:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\Big[\mathrm{L}[\boldsymbol{\phi}]\Big]$$

$$= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i]\right]$$

Explicit regularization

- Standard loss function:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\Big[\mathrm{L}[\boldsymbol{\phi}]\Big]$$

$$= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i]\right]$$

- Regularization adds an extra term

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i] + \lambda \cdot \mathrm{g}[\boldsymbol{\phi}]\right]$$

Explicit regularization

- Standard loss function:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\Big[\mathrm{L}[\boldsymbol{\phi}]\Big]$$

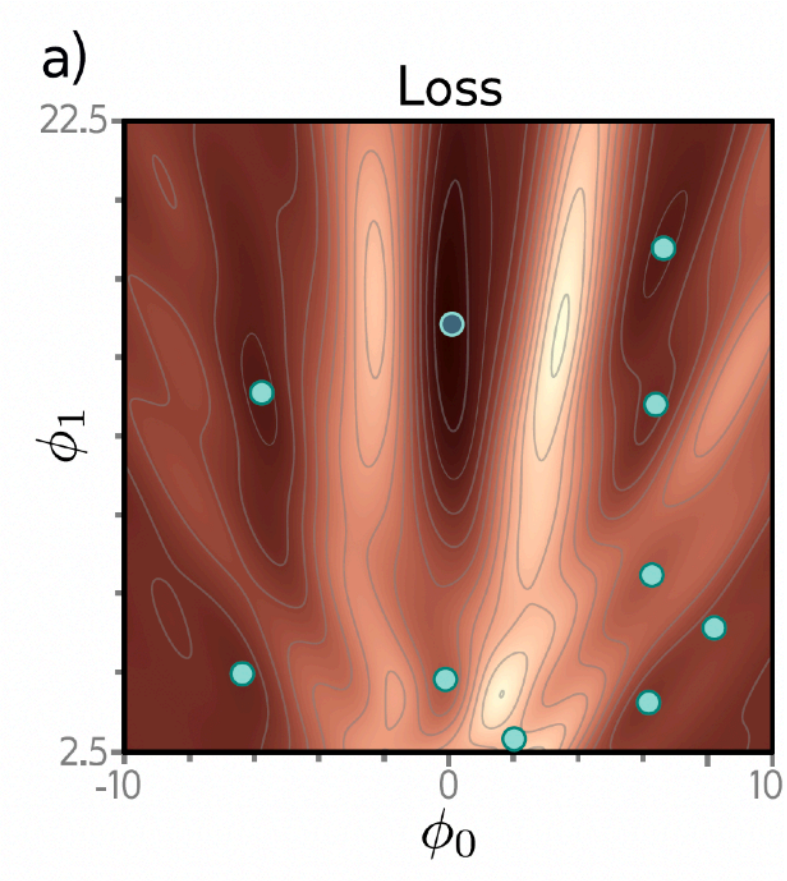
$$= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i]\right]$$

- Regularization adds an extra term

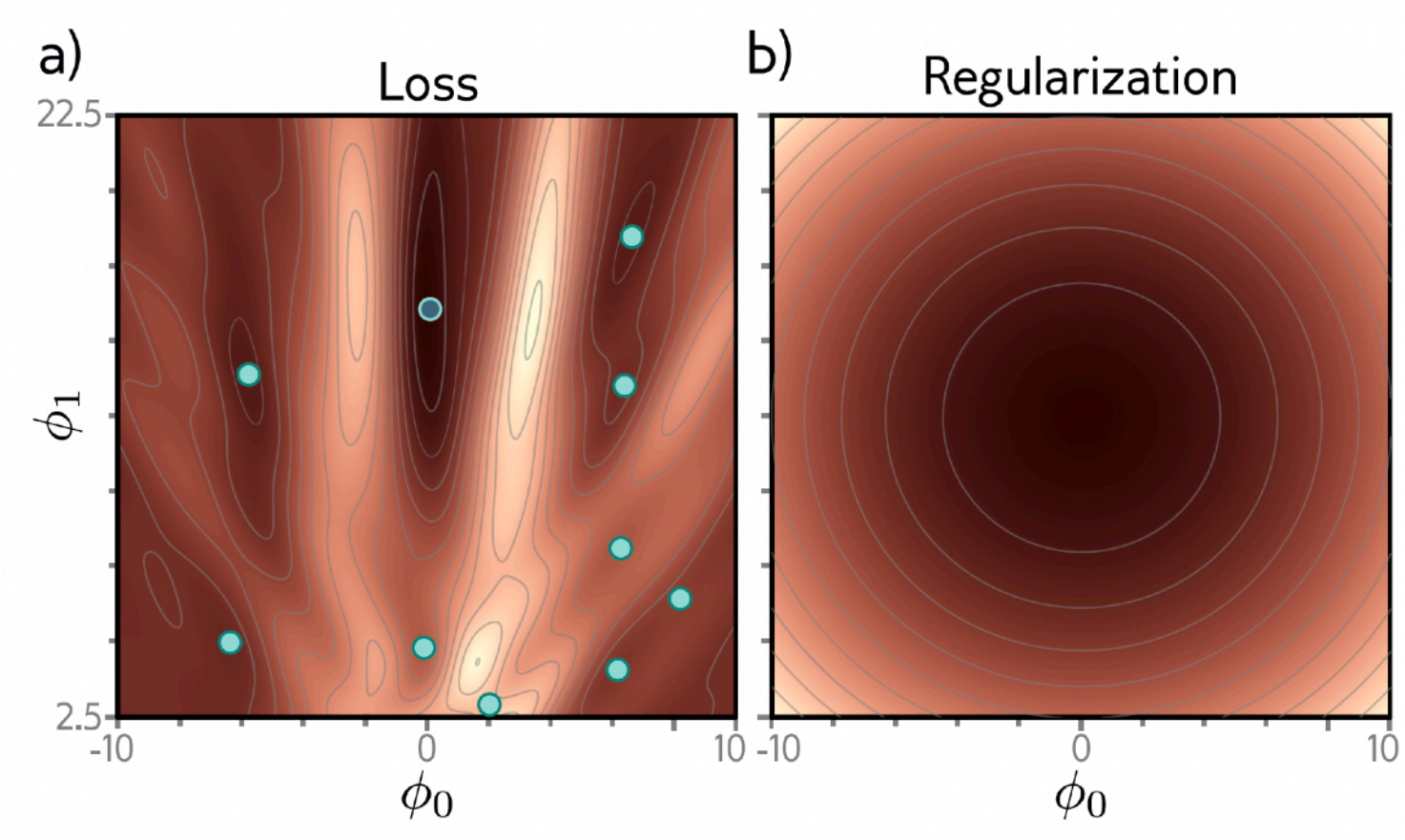
$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i] + \lambda \cdot \mathrm{g}[\boldsymbol{\phi}]\right]$$

- Favors some parameters, disfavors others.
- λ>0 controls the strength

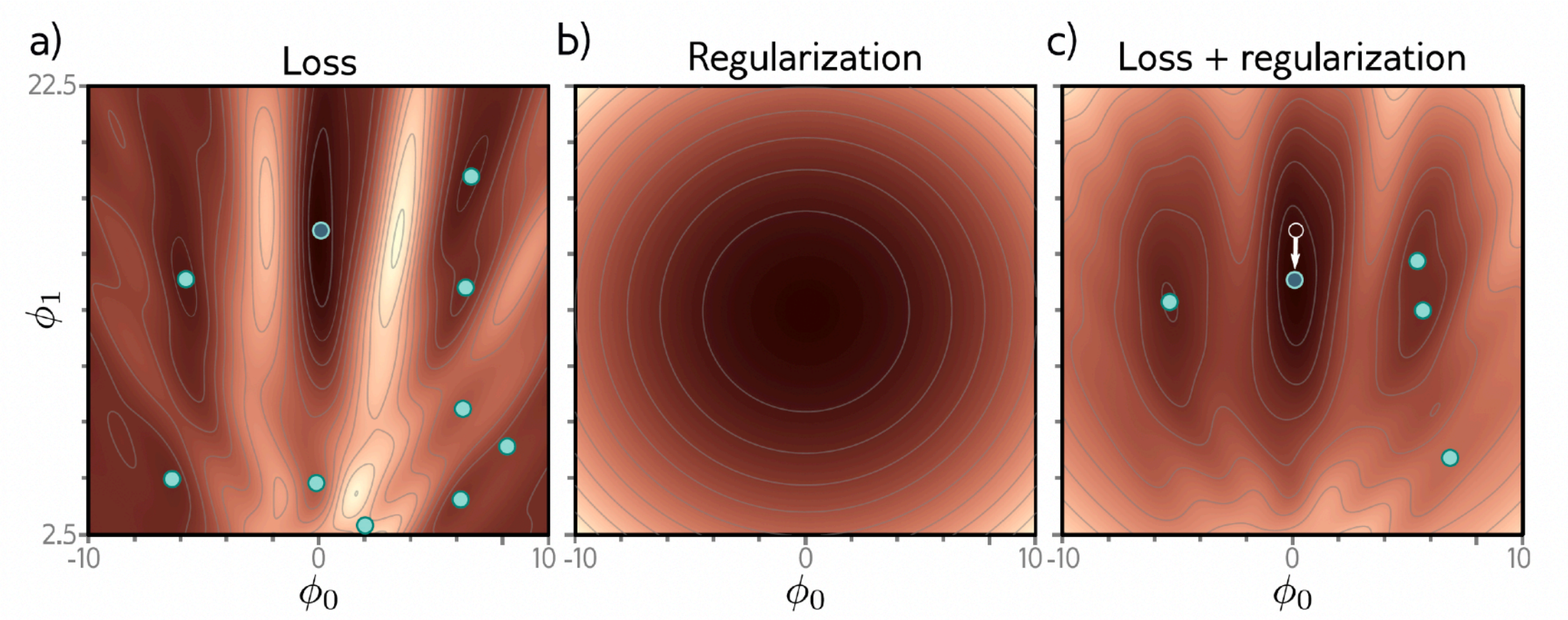
Explicit regularization



Explicit regularization



Explicit regularization



Probabilistic interpretation

- Maximum likelihood:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\phi}) \right]$$

- Regularization is equivalent to a adding a prior over parameters

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi}) \right]$$

… what you know about parameters ***before*** seeing the data

Equivalence

- Explicit regularization:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\text{argmin}} \left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i] + \lambda \cdot \text{g}[\boldsymbol{\phi}] \right]$$

- Probabilistic interpretation:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\text{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi}) \right]$$

Equivalence

- Explicit regularization:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} \ell_i[\mathbf{x}_i, \mathbf{y}_i] + \lambda \cdot \mathrm{g}[\boldsymbol{\phi}] \right]$$

- Probabilistic interpretation:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi}) \right]$$

- Mapping:

$$\lambda \cdot \mathrm{g}[\boldsymbol{\phi}] = -\log[Pr(\boldsymbol{\phi})]$$

L2 Regularization

- Can only use very general terms
- Most common is L2 regularization
- Favors smaller parameters

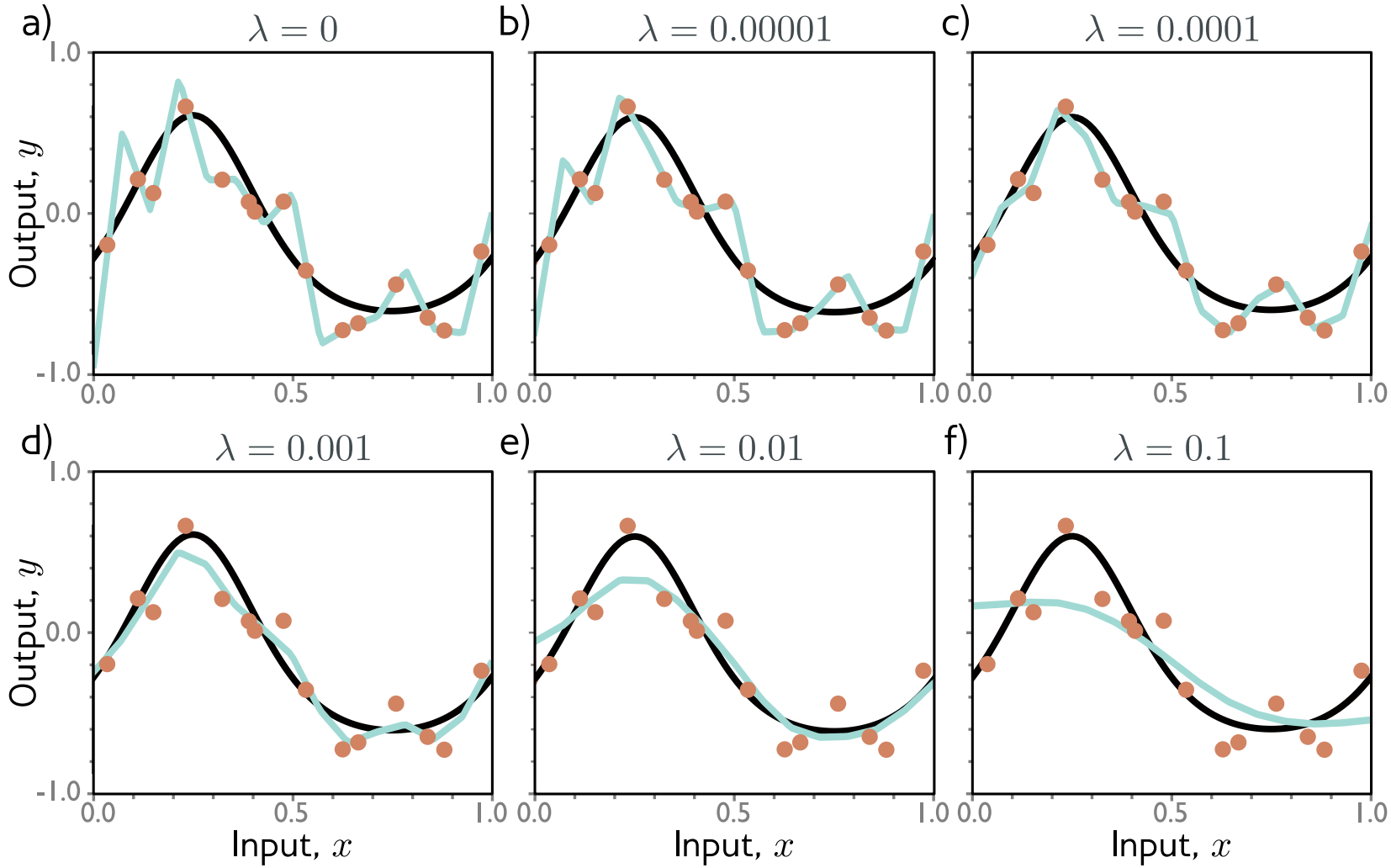
$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[\mathrm{L}[\boldsymbol{\phi}, \{\mathbf{x}_i, \mathbf{y}_i\}] + \lambda \sum_j \phi_j^2 \right]$$

- Also called Tichonov regularization, ridge regression
- In neural networks, usually just for weights and called weight decay

Why does L2 regularization help?

- Encourages smoothness between datapoints

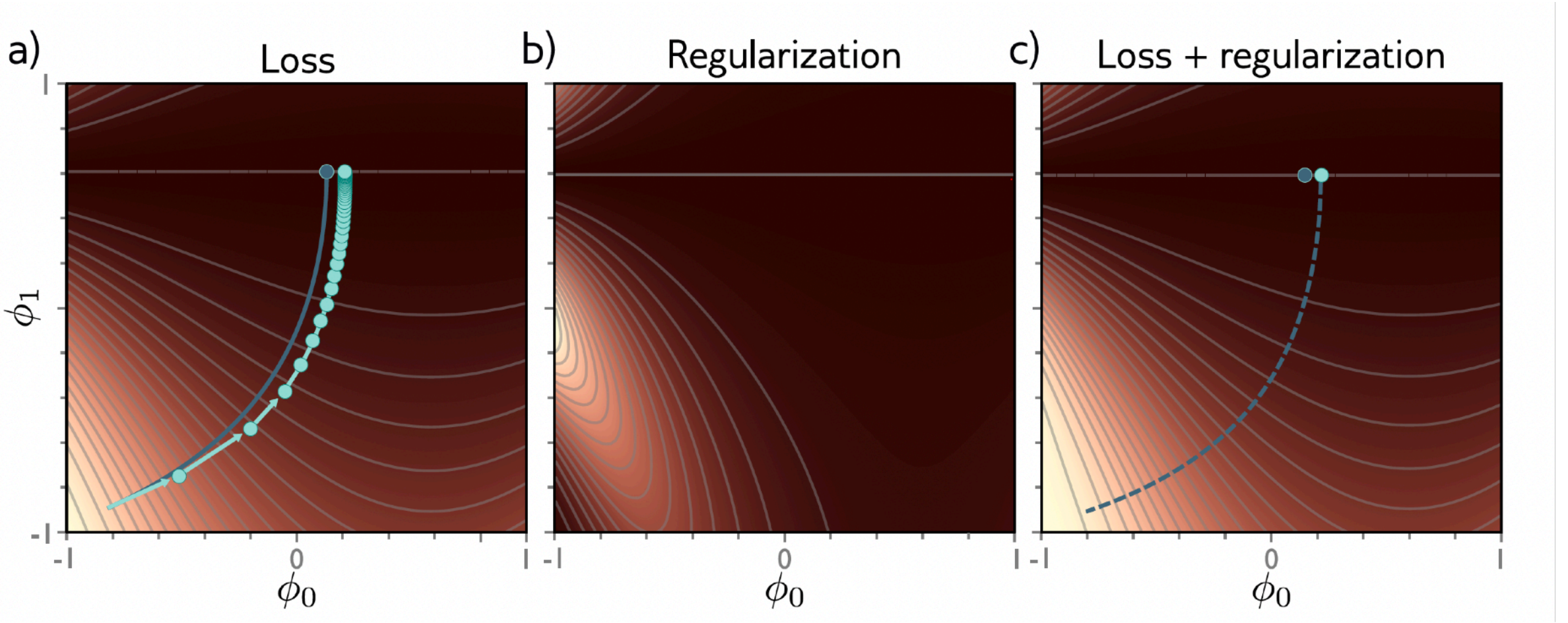
L2 regularization



Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Implicit regularization



Gradient descent approximates a differential equation (infinitesimal step size)

Finite step size equivalent to regularization

Add in that regularization and differential equation converges to same place

Implicit regularization

- Gradient descent disfavors areas where gradients are steep

$$\tilde{\mathrm{L}}_{GD}[\boldsymbol{\phi}] = \mathrm{L}[\boldsymbol{\phi}] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2$$

Implicit regularization

- Gradient descent disfavors areas where gradients are steep

$$\tilde{\mathrm{L}}_{GD}[\boldsymbol{\phi}] = \mathrm{L}[\boldsymbol{\phi}] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2$$

- SGD likes all batches to have similar gradients

$$\begin{aligned}\tilde{\mathrm{L}}_{SGD}[\boldsymbol{\phi}] &= \tilde{\mathrm{L}}_{GD}[\boldsymbol{\phi}] + \frac{\alpha}{4B}\sum_{b=1}^{B}\left\|\frac{\partial L_b}{\partial \boldsymbol{\phi}} - \frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2 \\ &= \mathrm{L}[\boldsymbol{\phi}] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2 + \frac{\alpha}{4B}\sum_{b=1}^{B}\left\|\frac{\partial L_b}{\partial \boldsymbol{\phi}} - \frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2\end{aligned}$$

Implicit regularization

- Gradient descent disfavors areas where gradients are steep

$$\tilde{\mathrm{L}}_{GD}[\boldsymbol{\phi}] = \mathrm{L}[\boldsymbol{\phi}] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2$$

- SGD likes all batches to have similar gradients

$$\begin{aligned}\tilde{\mathrm{L}}_{SGD}[\boldsymbol{\phi}] &= \tilde{\mathrm{L}}_{GD}[\boldsymbol{\phi}] + \frac{\alpha}{4B}\sum_{b=1}^{B}\left\|\frac{\partial L_b}{\partial \boldsymbol{\phi}} - \frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2 \\ &= \mathrm{L}[\boldsymbol{\phi}] + \frac{\alpha}{4}\left\|\frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2 + \frac{\alpha}{4B}\sum_{b=1}^{B}\left\|\frac{\partial L_b}{\partial \boldsymbol{\phi}} - \frac{\partial L}{\partial \boldsymbol{\phi}}\right\|^2\end{aligned}$$

- Depends on learning rate – perhaps why larger learning rates generalize better.

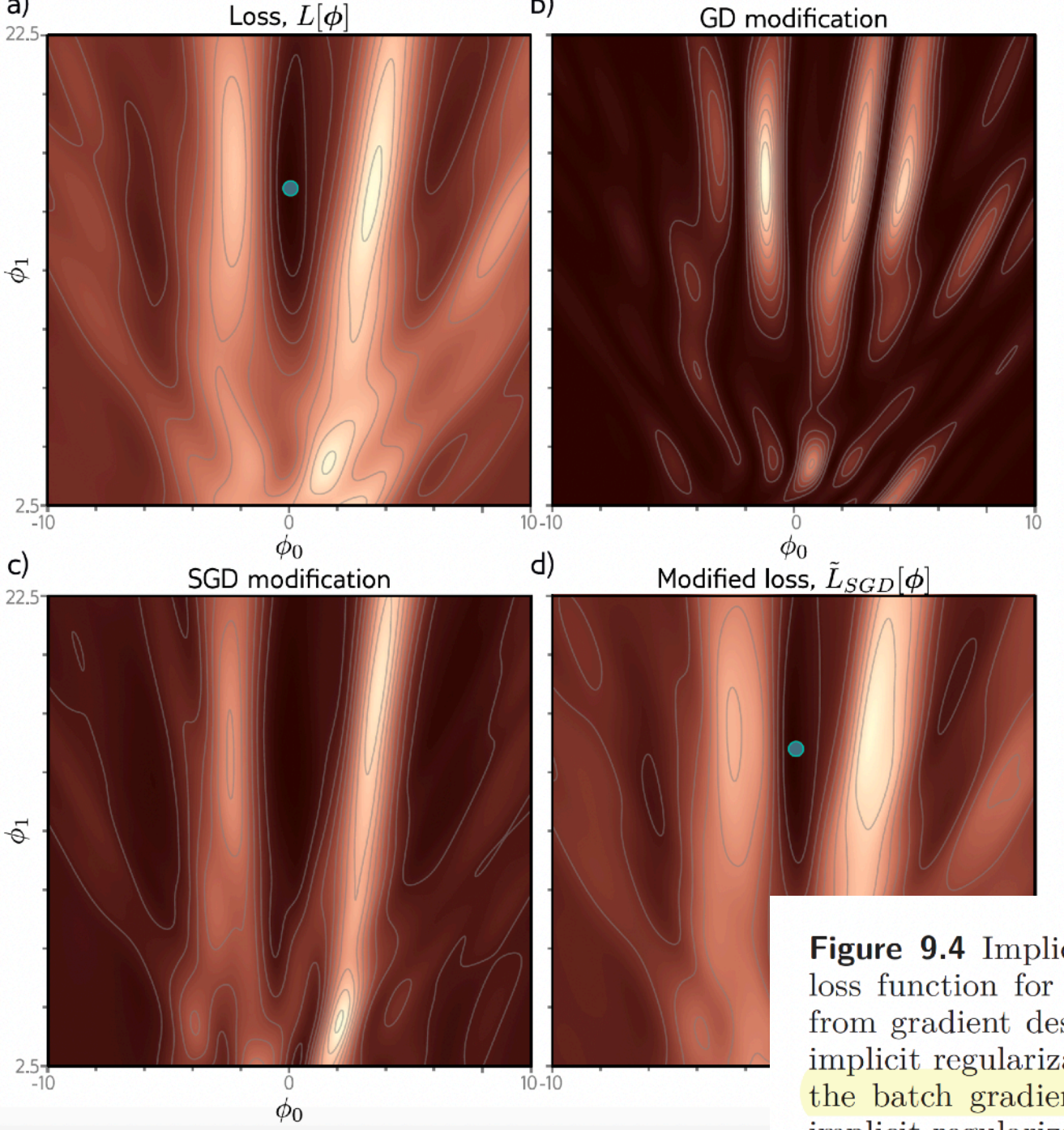
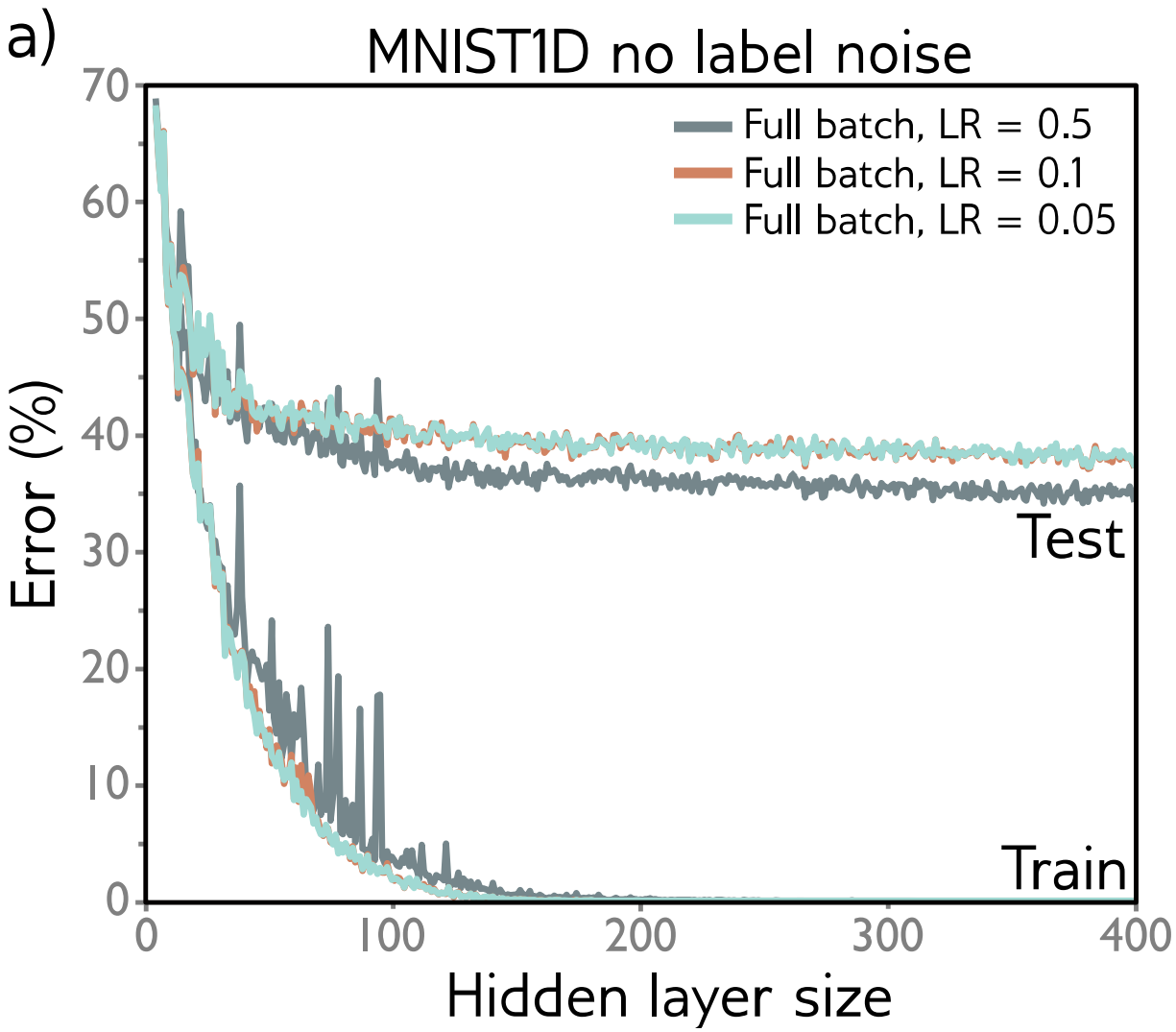
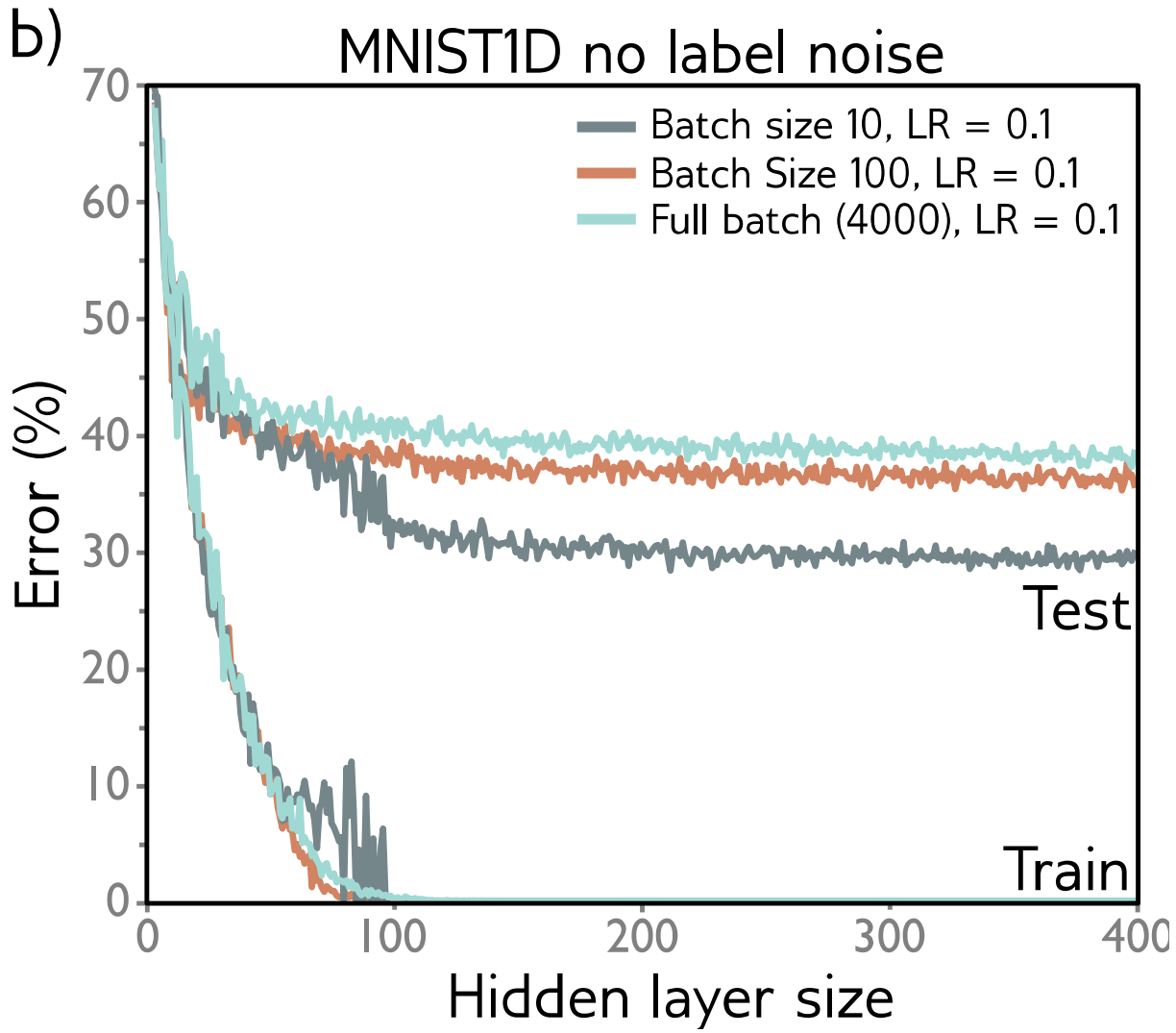


Figure 9.4 Implicit regularization for stochastic gradient descent. a) Original loss function for Gabor model (section 6.1.2). b) Implicit regularization term from gradient descent penalizes the squared gradient magnitude. c) Additional implicit regularization from stochastic gradient descent penalizes the variance of the batch gradients. d) Modified loss function (sum of original loss plus two implicit regularization components).


a)
MNIST1D no label noise
Full batch, LR = 0.5
Full batch, LR = 0.1
Full batch, LR = 0.05
Test
Train
Error (%)
Hidden layer size
0
10
20
30
40
50
60
70
0
100
200
300
400

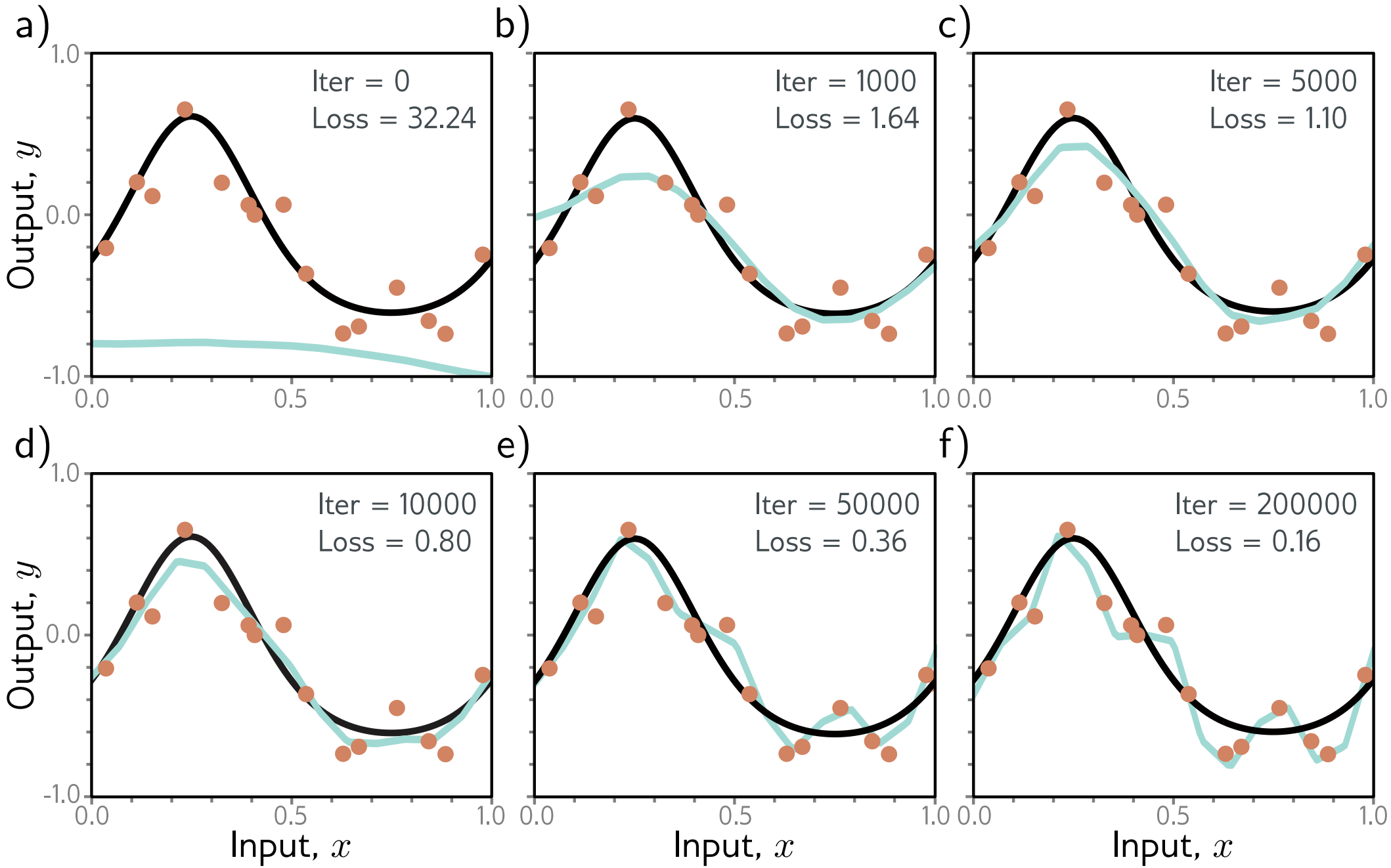

b)
MNIST1D no label noise
Batch size 10, LR = 0.1
Batch Size 100, LR = 0.1
Full batch (4000), LR = 0.1
Test
Train
Error (%)
Hidden layer size
0
10
20
30
40
50
60
70
0
100
200
300
400

Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Early stopping

- If we stop training early, weights don't have time to overfit to noise
- Weights start small, don't have time to get large
- Reduces effective model complexity
- Known as early stopping
- Don't have to re-train

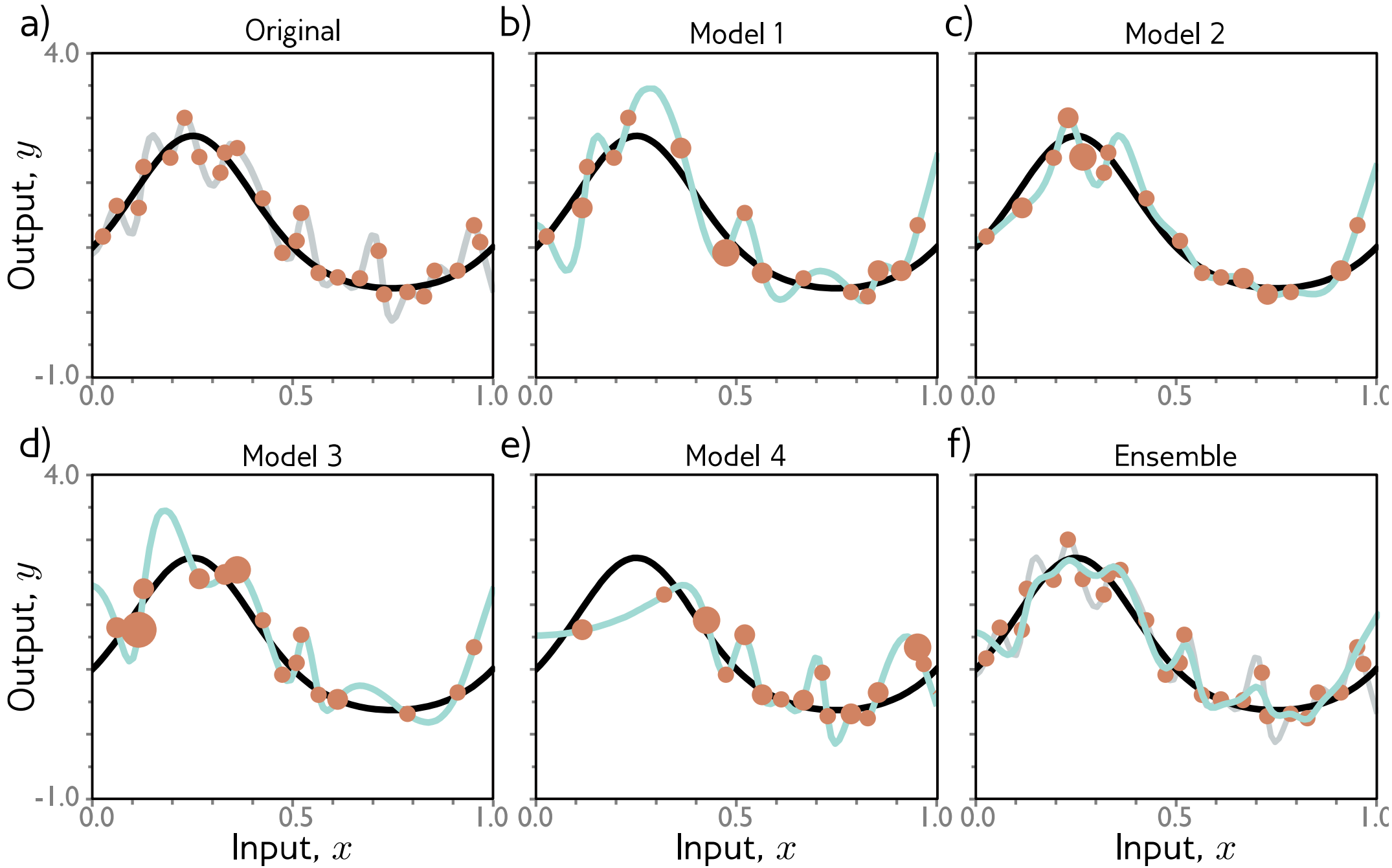

a)
b)
c)
d)
e)
f)
Iter = 0
Loss = 32.24
Iter = 1000
Loss = 1.64
Iter = 5000
Loss = 1.10
Iter = 10000
Loss = 0.80
Iter = 50000
Loss = 0.36
Iter = 200000
Loss = 0.16
Output, y
Input, x
1.0
0.0
-1.0
0.5

Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Ensembling

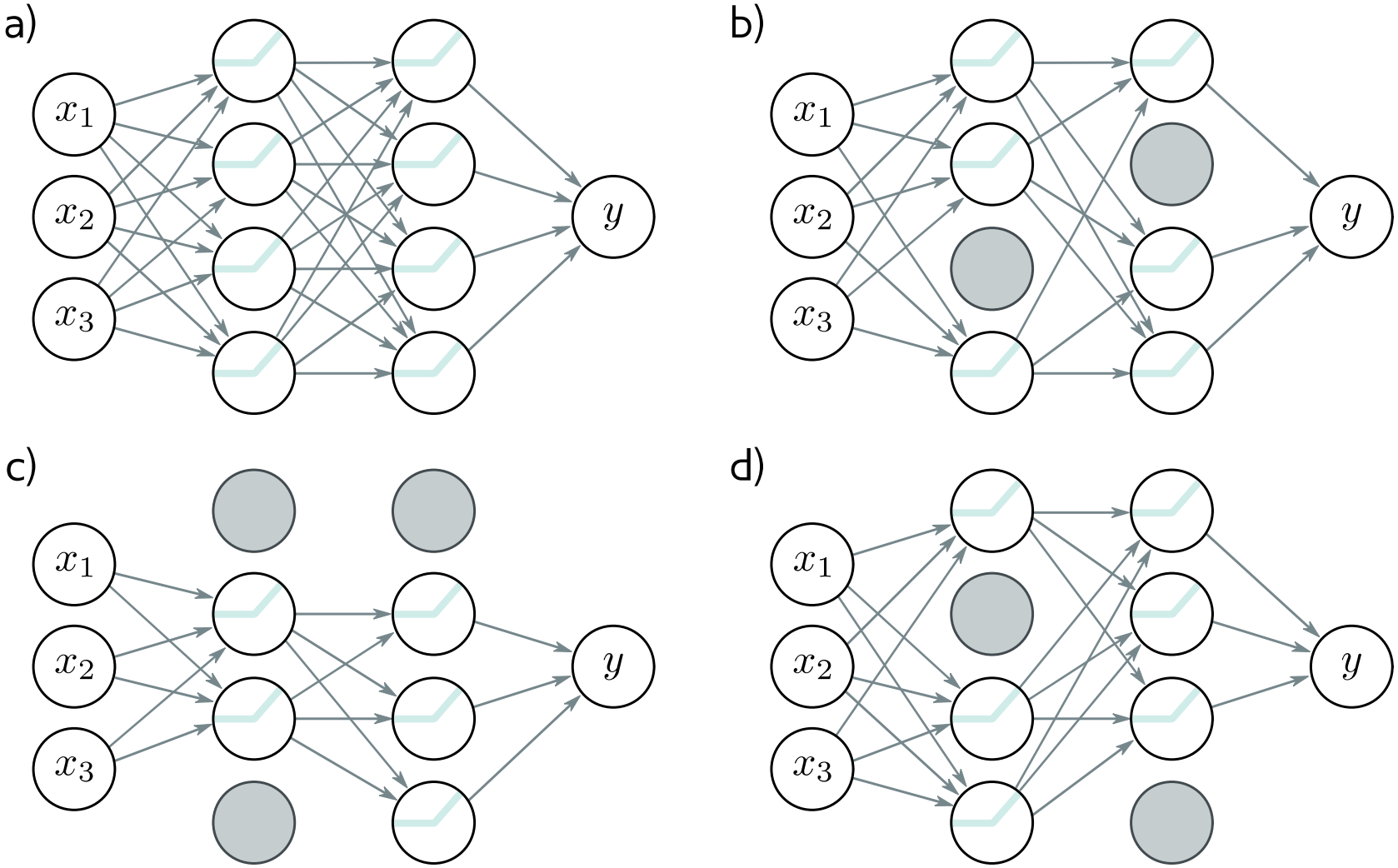
- Average together several models – an ensemble
- Can take mean or median
- Different initializations / different models
- Different subsets of the data resampled with replacements -- bagging


a)
Original
b)
Model 1
c)
Model 2
d)
Model 3
e)
Model 4
f)
Ensemble
Output, y
Input, x
4.0
-1.0
0.0
0.5
1.0

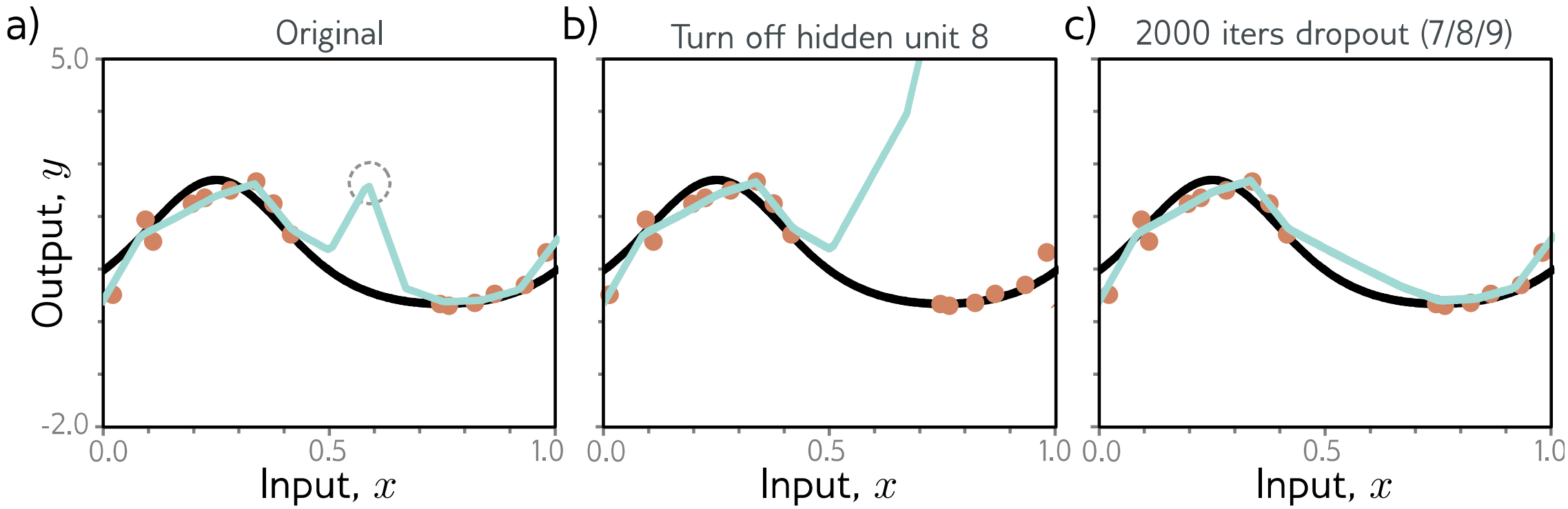
Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Dropout



Dropout

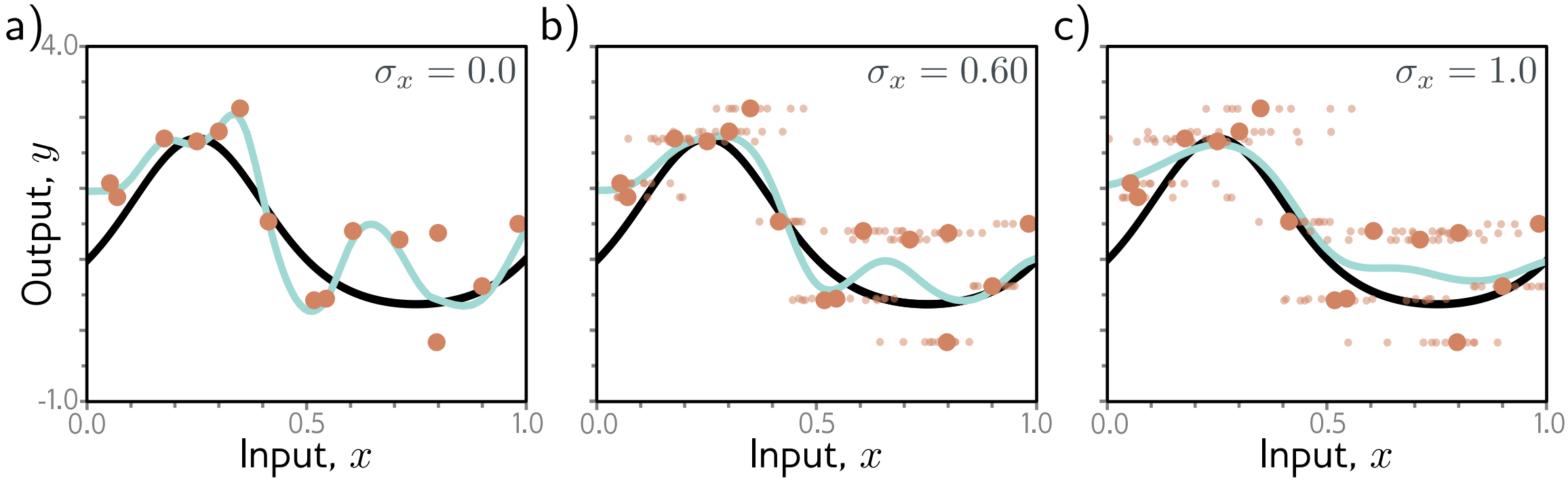


Can eliminate kinks in function that are far from data and don't contribute to training loss

Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Adding noise



- to inputs
- to weights
- to outputs (labels)

Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Bayesian approaches

- There are many parameters compatible with the data
- Can find a probability distribution over them

Prior info about parameters

$$Pr(\boldsymbol{\phi}|\{\mathbf{x}_i, \mathbf{y}_i\}) = \frac{\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi})}{\int \prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi}) d\boldsymbol{\phi}}$$

Bayesian approaches

- There are many parameters compatible with the data
- Can find a probability distribution over them

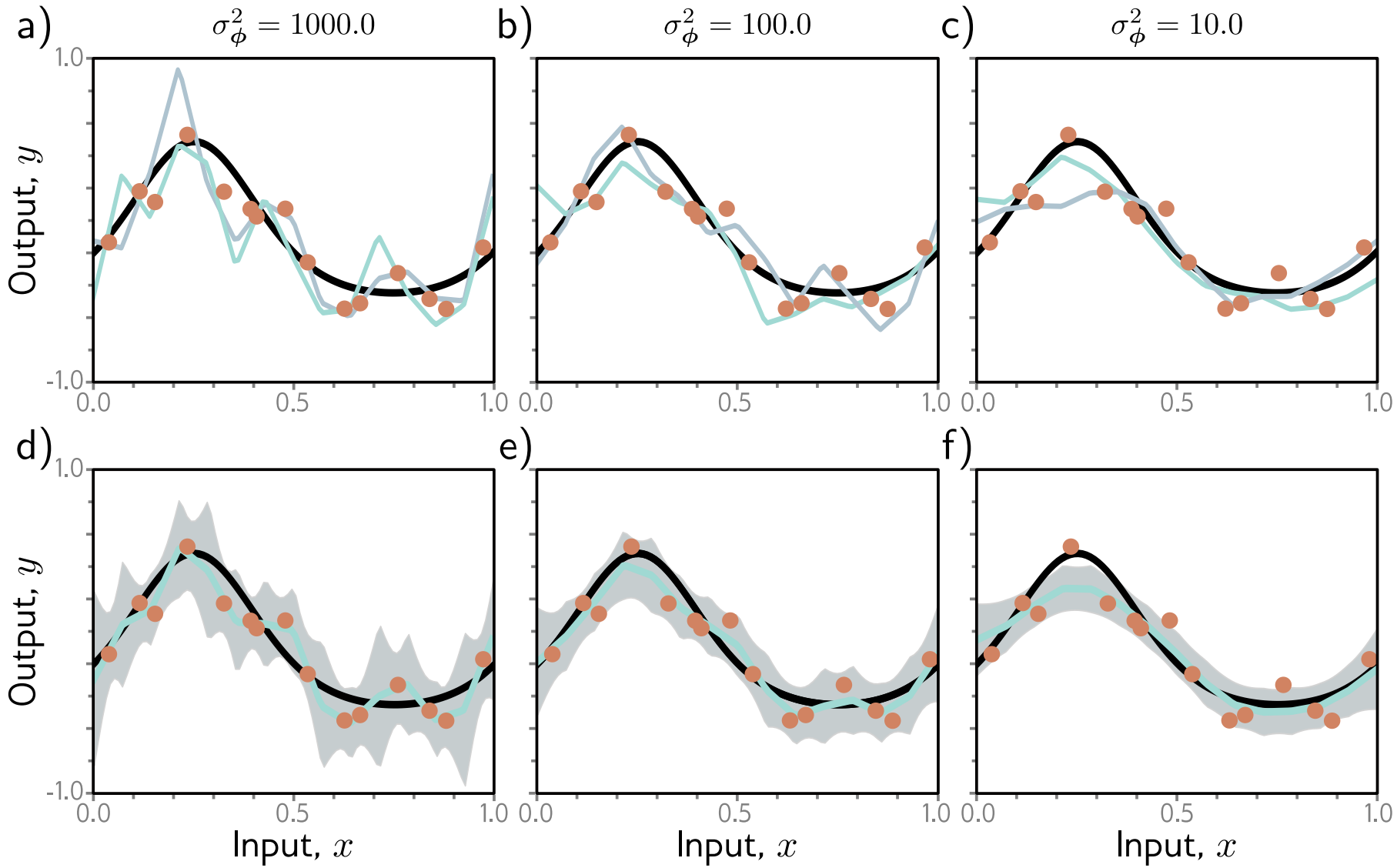
Prior info about parameters

$$Pr(\boldsymbol{\phi}|\{\mathbf{x}_i, \mathbf{y}_i\}) = \frac{\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi})}{\int \prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\phi}) Pr(\boldsymbol{\phi}) d\boldsymbol{\phi}}$$

- Take all possible parameters into account when make prediction

$$Pr(\mathbf{y}|\mathbf{x}, \{\mathbf{x}_i, \mathbf{y}_i\}) = \int Pr(\mathbf{y}|\mathbf{x}, \boldsymbol{\phi}) Pr(\boldsymbol{\phi}|\{\mathbf{x}_i, \mathbf{y}_i\}) d\boldsymbol{\phi}$$

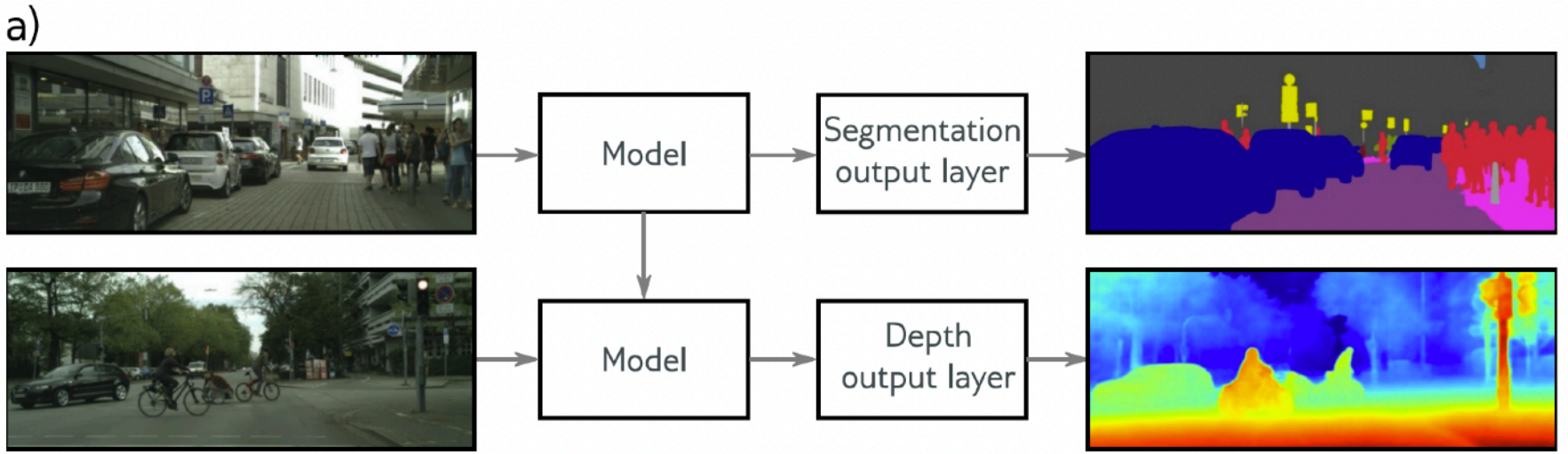
Bayesian approaches



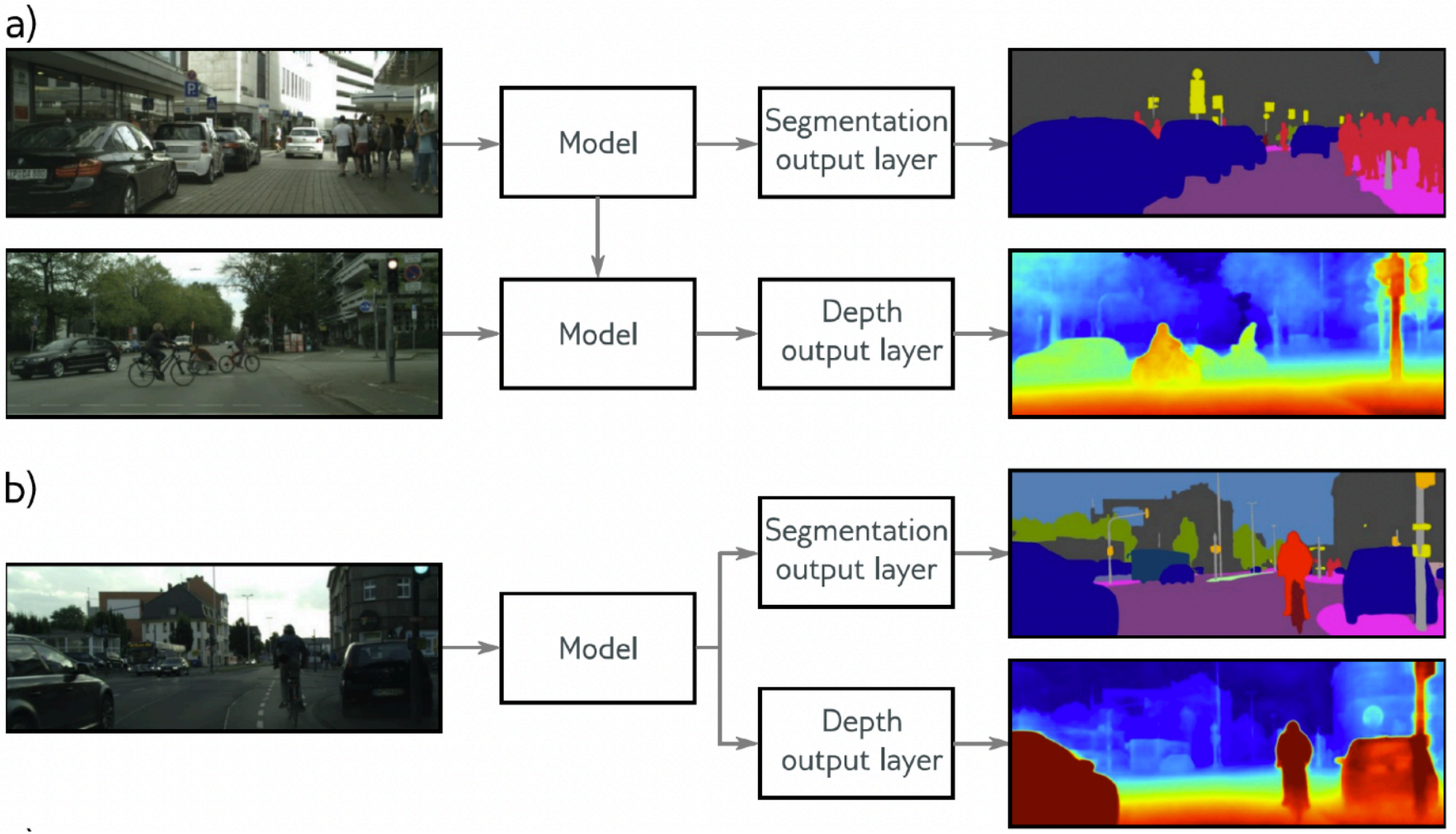
Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

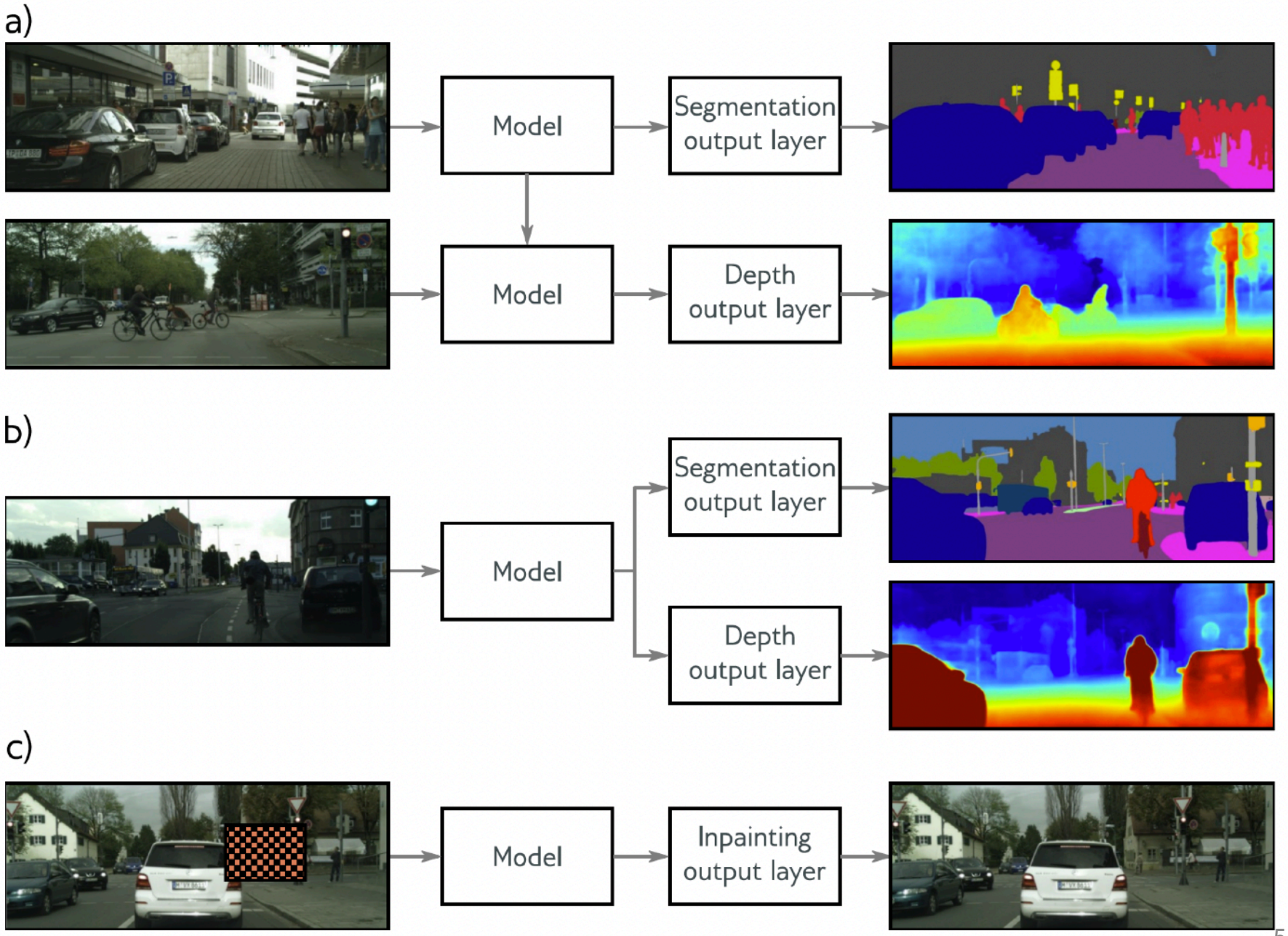
- Transfer learning



- Transfer learning
- Multi-task learning



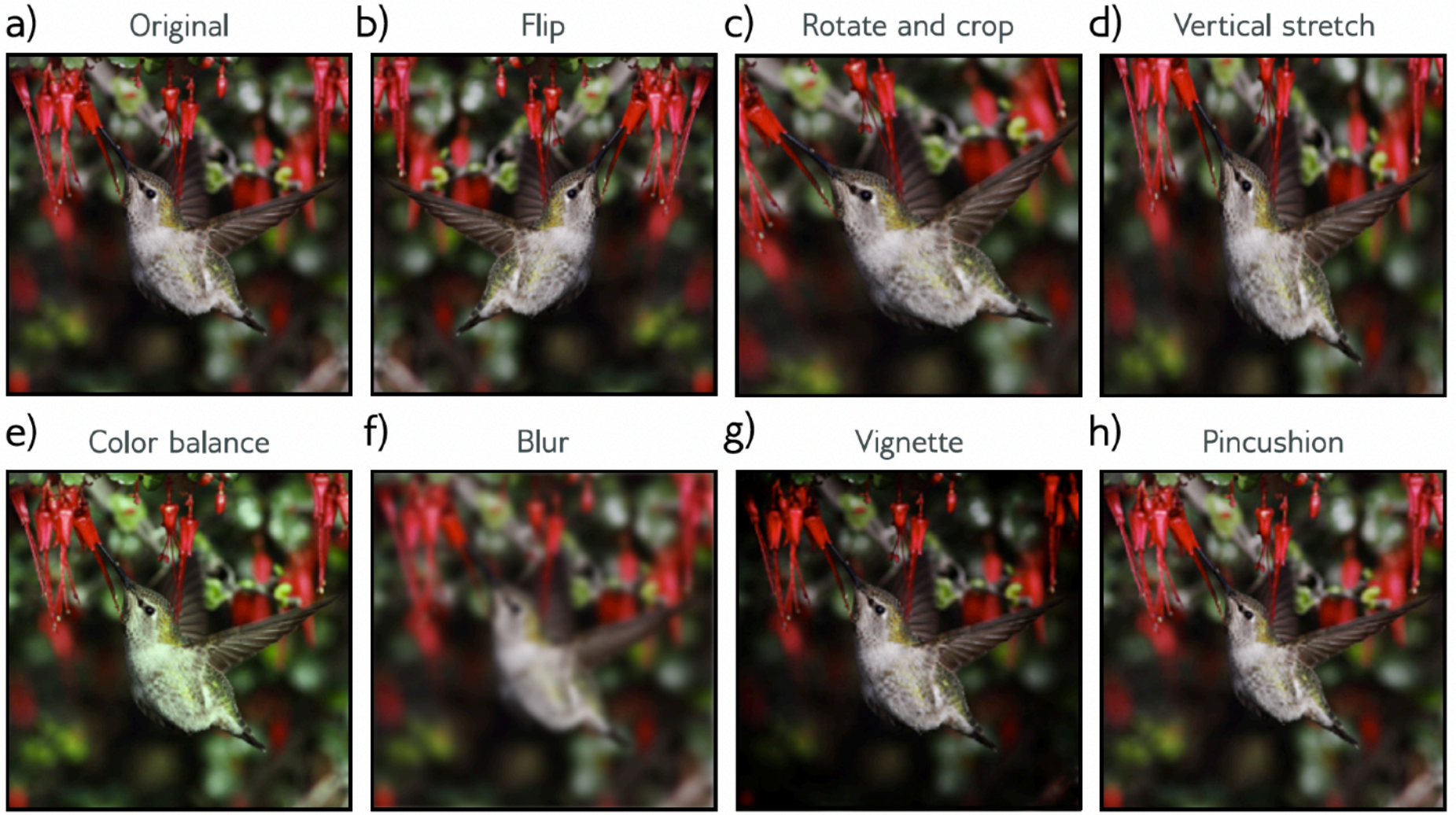
- Transfer learning
- Multi-task learning
- Self-supervised learning



Regularization

- Explicit regularization
- Implicit regularization
- Early stopping
- Ensembling
- Dropout
- Adding noise
- Bayesian approaches
- Transfer learning, multi-task learning, self-supervised learning
- Data augmentation

Data augmentation



Regularization overview