UNIVERSITY OF BATH

CM20315 - Machine Learning

Prof. Simon Prince

9. Regularization

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:

$$\begin{aligned}\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\big[\mathrm{L}[\boldsymbol{\phi}]\big] \\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}\ell_i[\mathbf{x}_i,\mathbf{y}_i]\right]\end{aligned}$$

- 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:

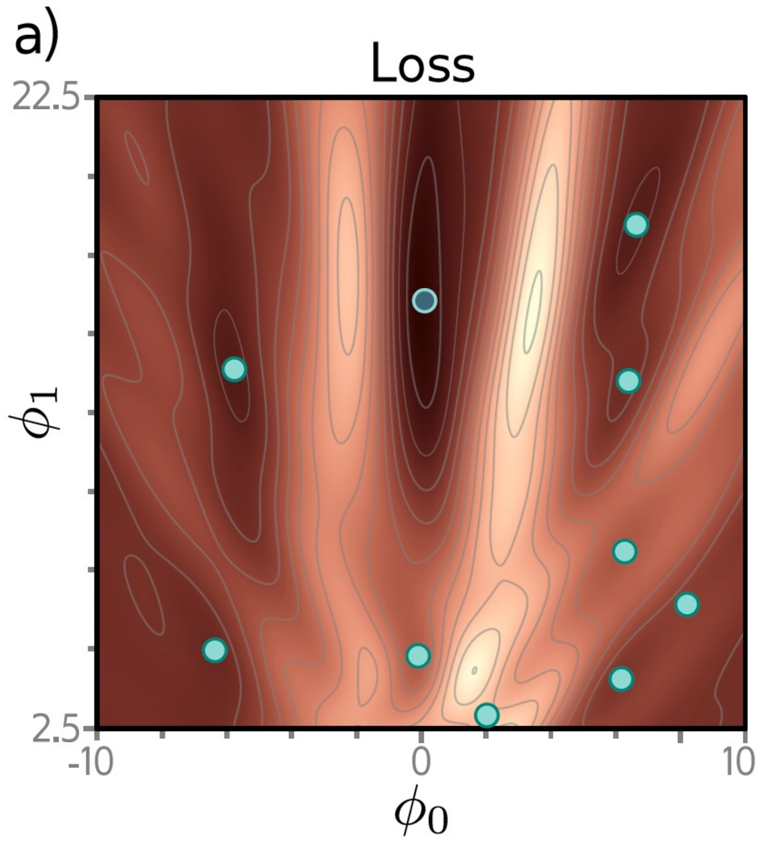
$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\big[\mathrm{L}[\boldsymbol{\phi}]\big]$$
$$= \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}\ell_i[\mathbf{x}_i, \mathbf{y}_i]\right]$$

- Regularization adds an extra term

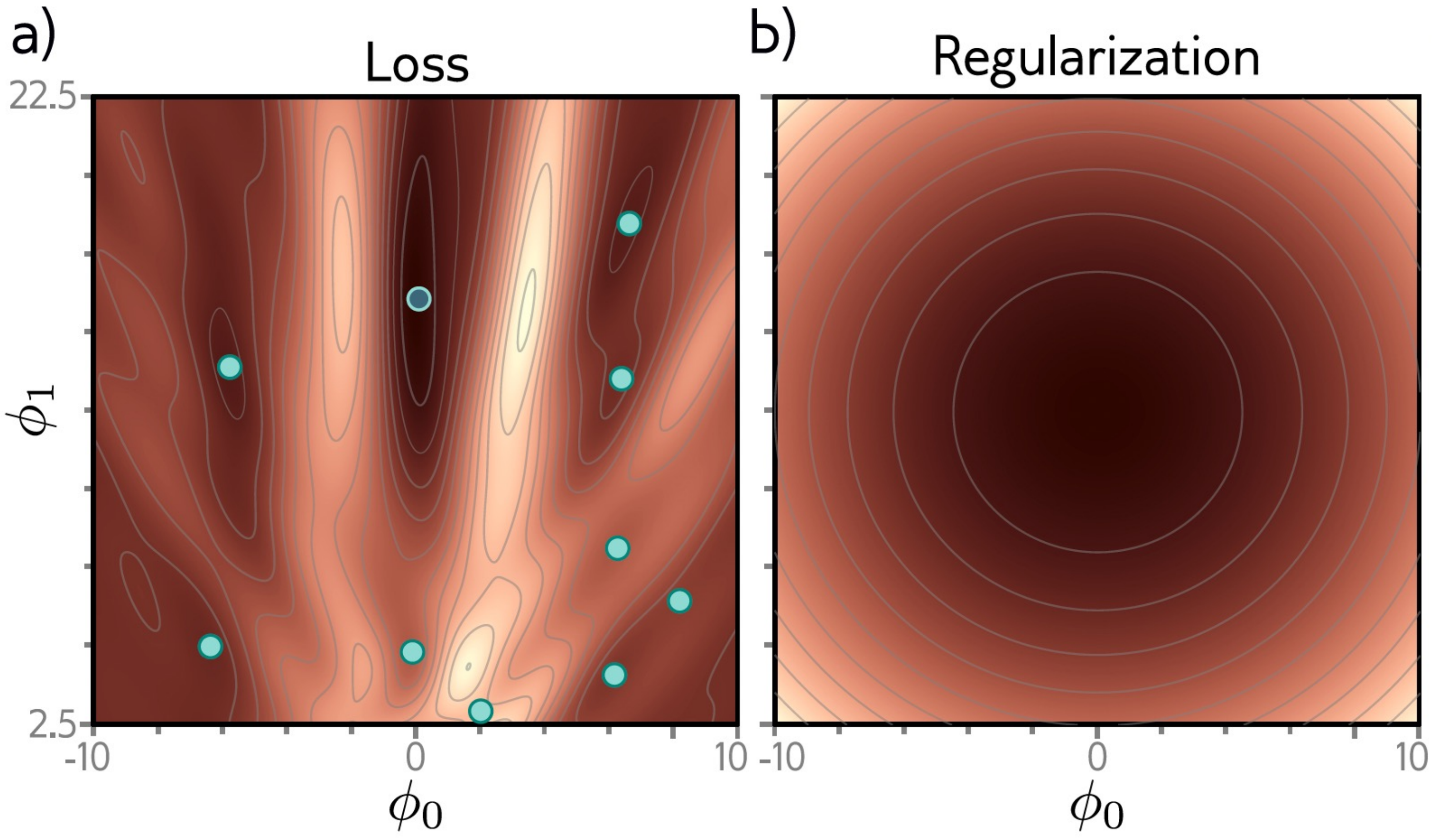
$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[\sum_{i=1}^{I}\ell_i[\mathbf{x}_i, \mathbf{y}_i] + \lambda \cdot \mathrm{g}[\boldsymbol{\phi}]\right]$$

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

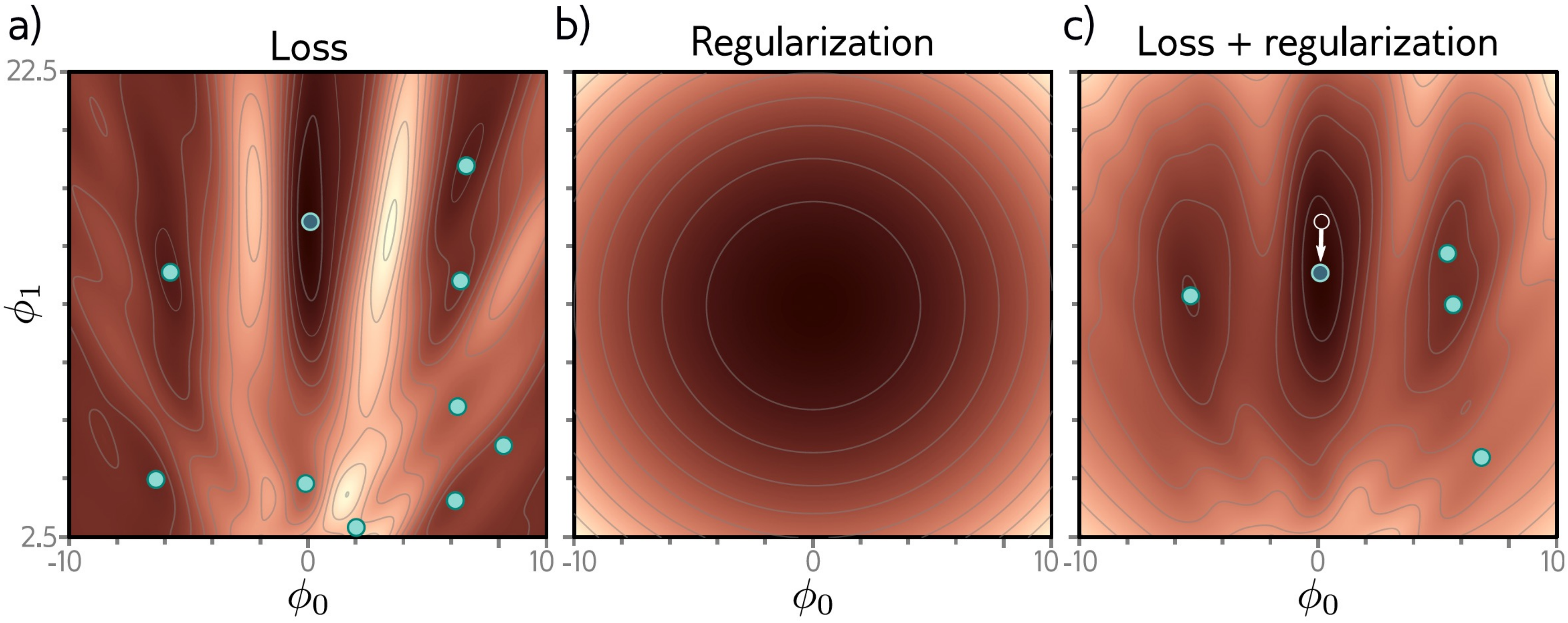
Explicit regularization



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:

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

- Probabilistic interpretation:

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

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

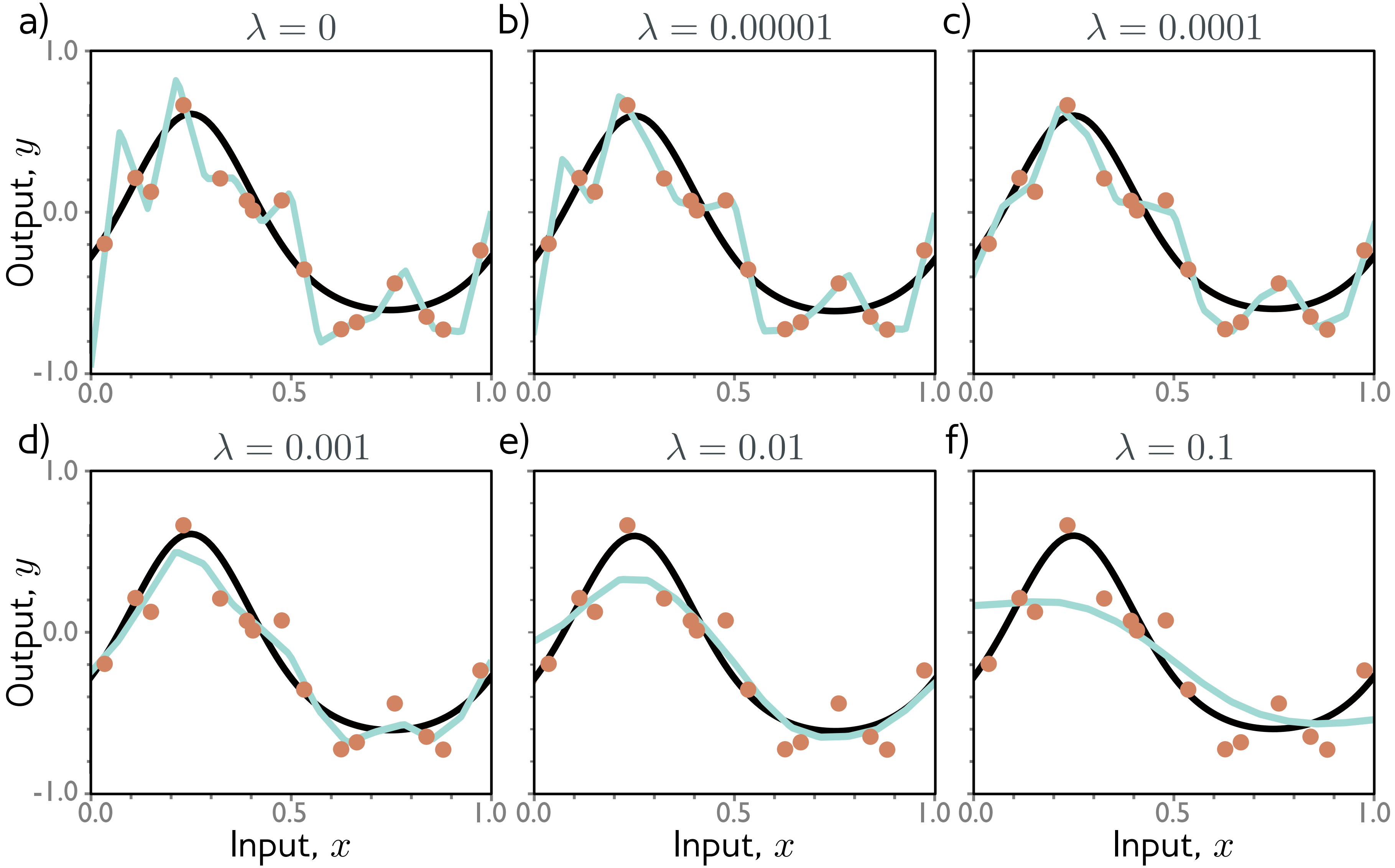
$$\hat{\phi} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[\mathrm{L}[\boldsymbol{\phi}, \{\mathbf{x}_i, \mathbf{y}_i\}] + \lambda \sum_j \phi_j^2 \right]$$

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

Why does L2 regularization help?

- Discourages overcommitment to the data (overfitting)
- Encourages smoothness between datapoints

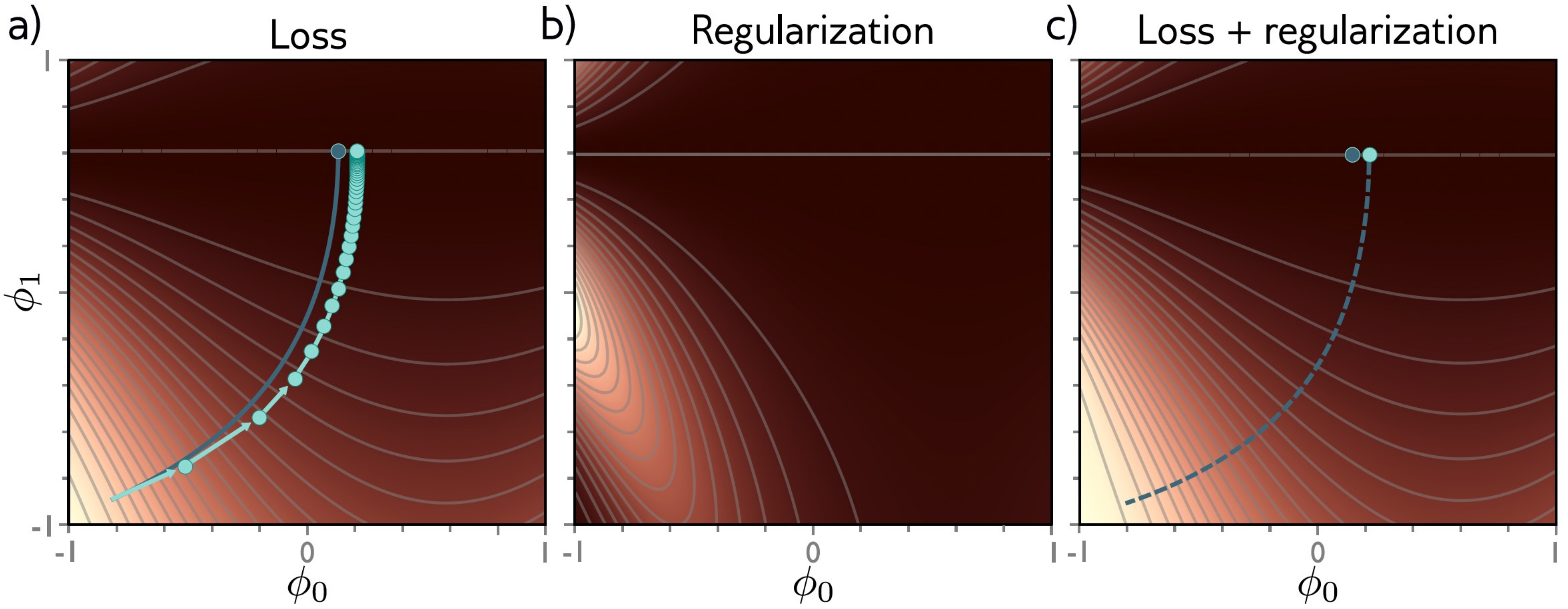
L2 regularization



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

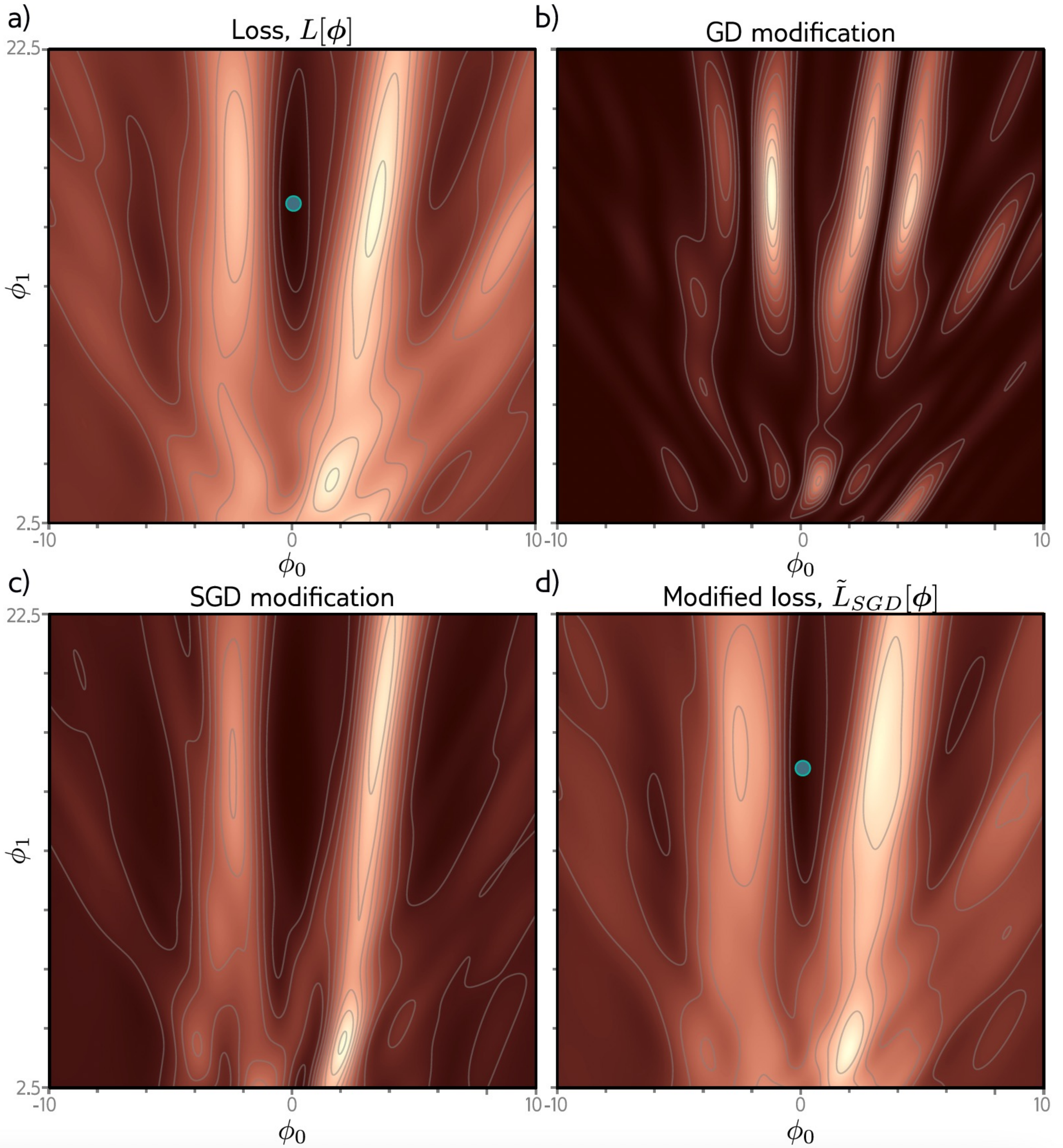
- Gradient descent disfavors areas where gradients are steep

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

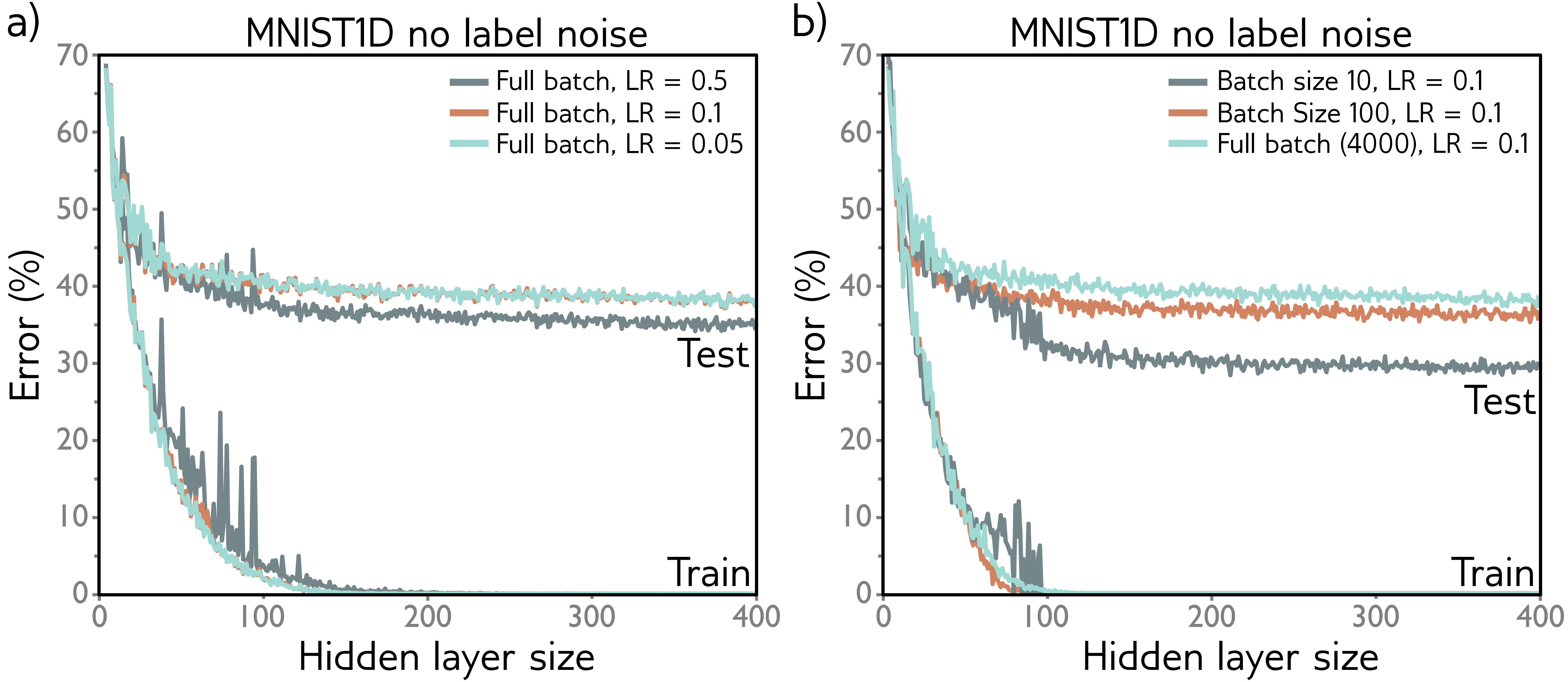
- SGD likes all batches to have similar gradients

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

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



a)
Loss, $L[\phi]$
b)
GD modification
c)
SGD modification
d)
Modified loss, $\tilde{L}_{SGD}[\phi]$
22.5
2.5
-10
0
10
ϕ_0
ϕ_1



Generally performance

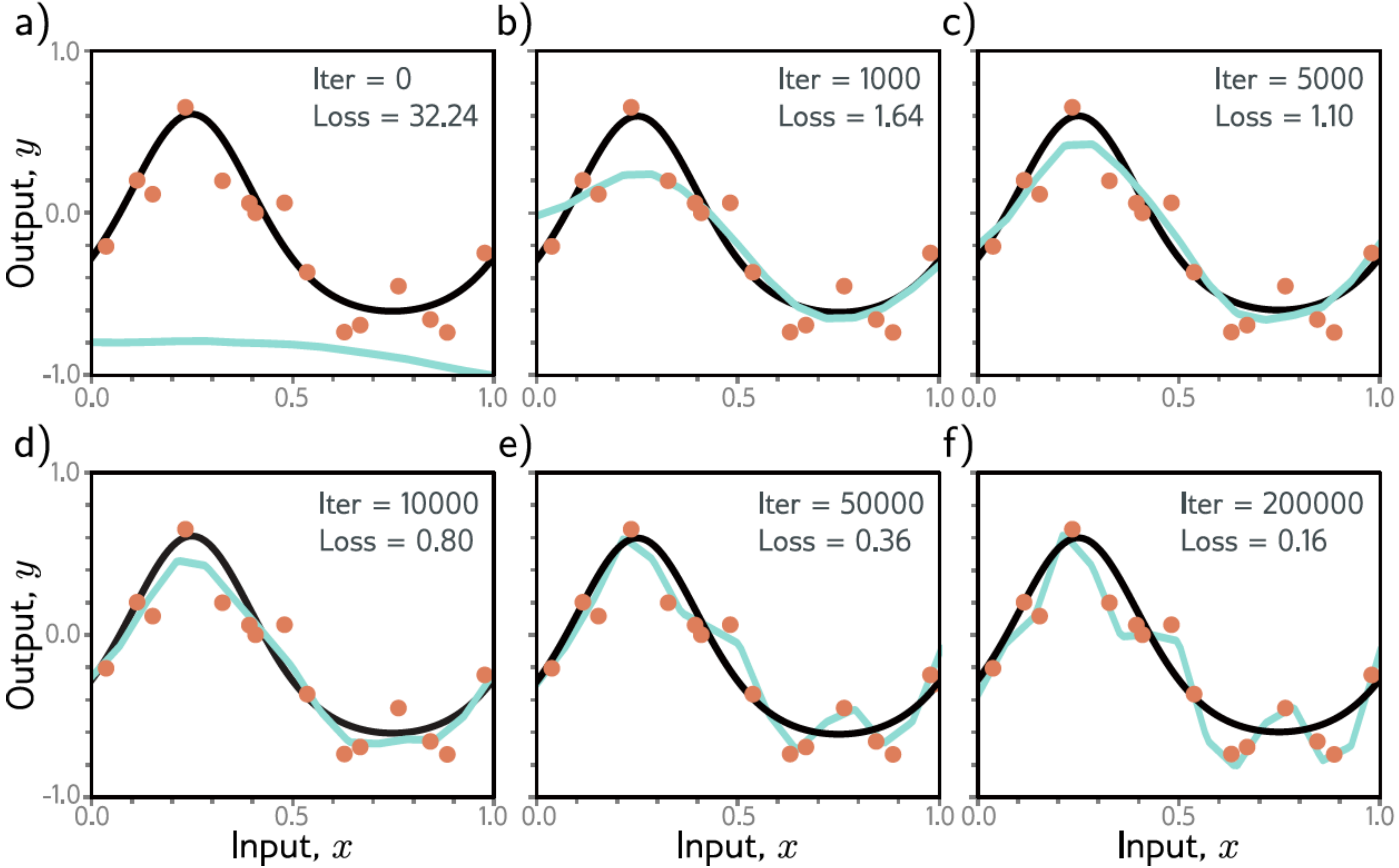
- best for larger learning rates
- best with smaller batches

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

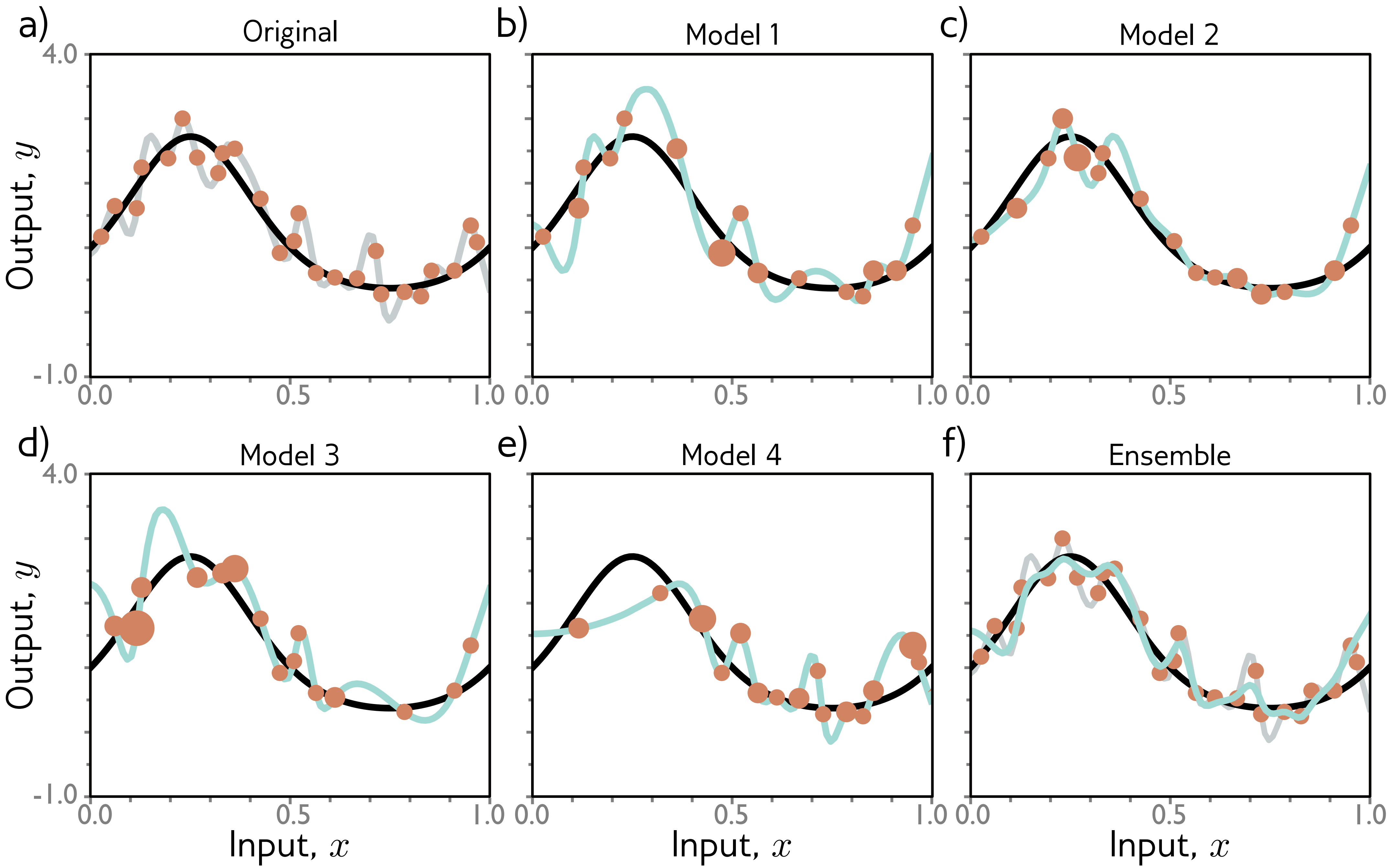

a)
Iter = 0
Loss = 32.24
b)
Iter = 1000
Loss = 1.64
c)
Iter = 5000
Loss = 1.10
d)
Iter = 10000
Loss = 0.80
e)
Iter = 50000
Loss = 0.36
f)
Iter = 200000
Loss = 0.16
Output, y
Input, x
1.0
0.0
-1.0
0.0
0.5
1.0

Regularization

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

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

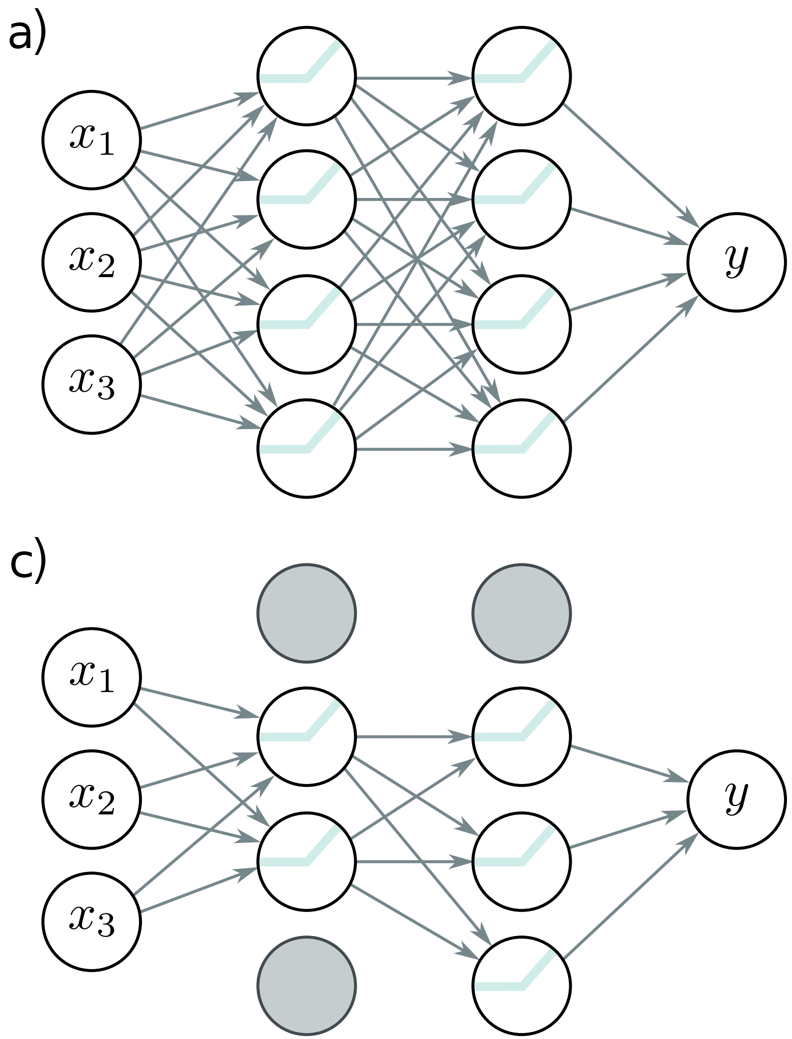


a)
Original
b)
Model 1
c)
Model 2
d)
Model 3
e)
Model 4
f)
Ensemble
4.0
-1.0
Output, y
0.0
0.5
1.0
Input, x

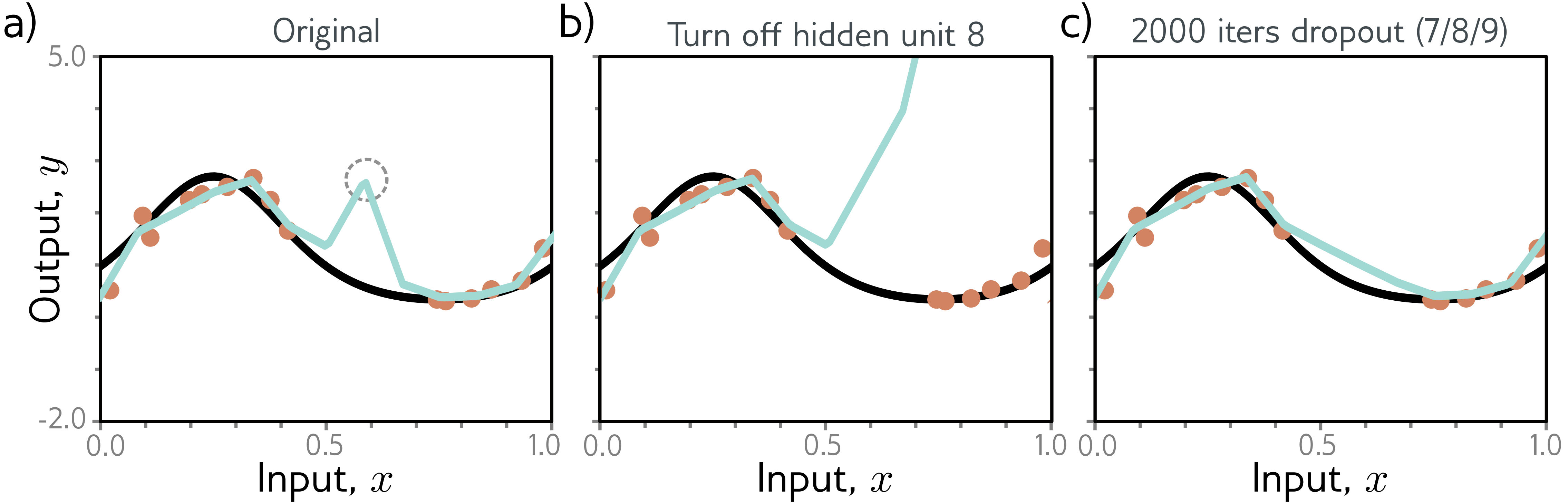
Regularization

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

Dropout



Dropout

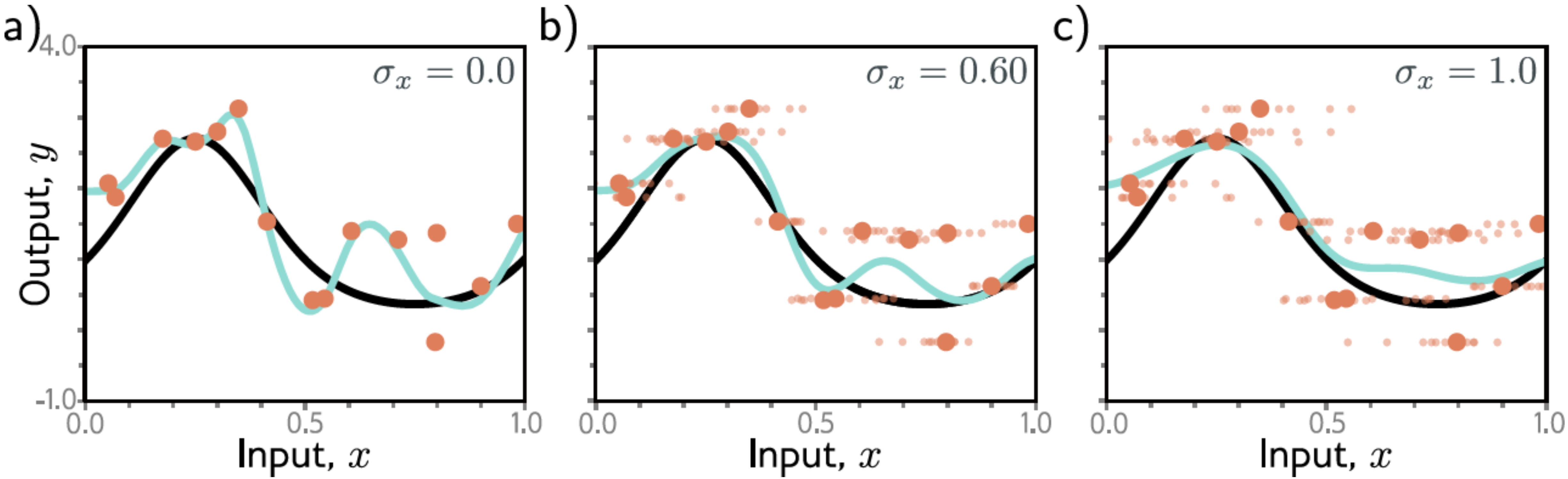


Can eliminate kinks in function that are far from data and don't contribute to training loss

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

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

Bayesian approaches

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

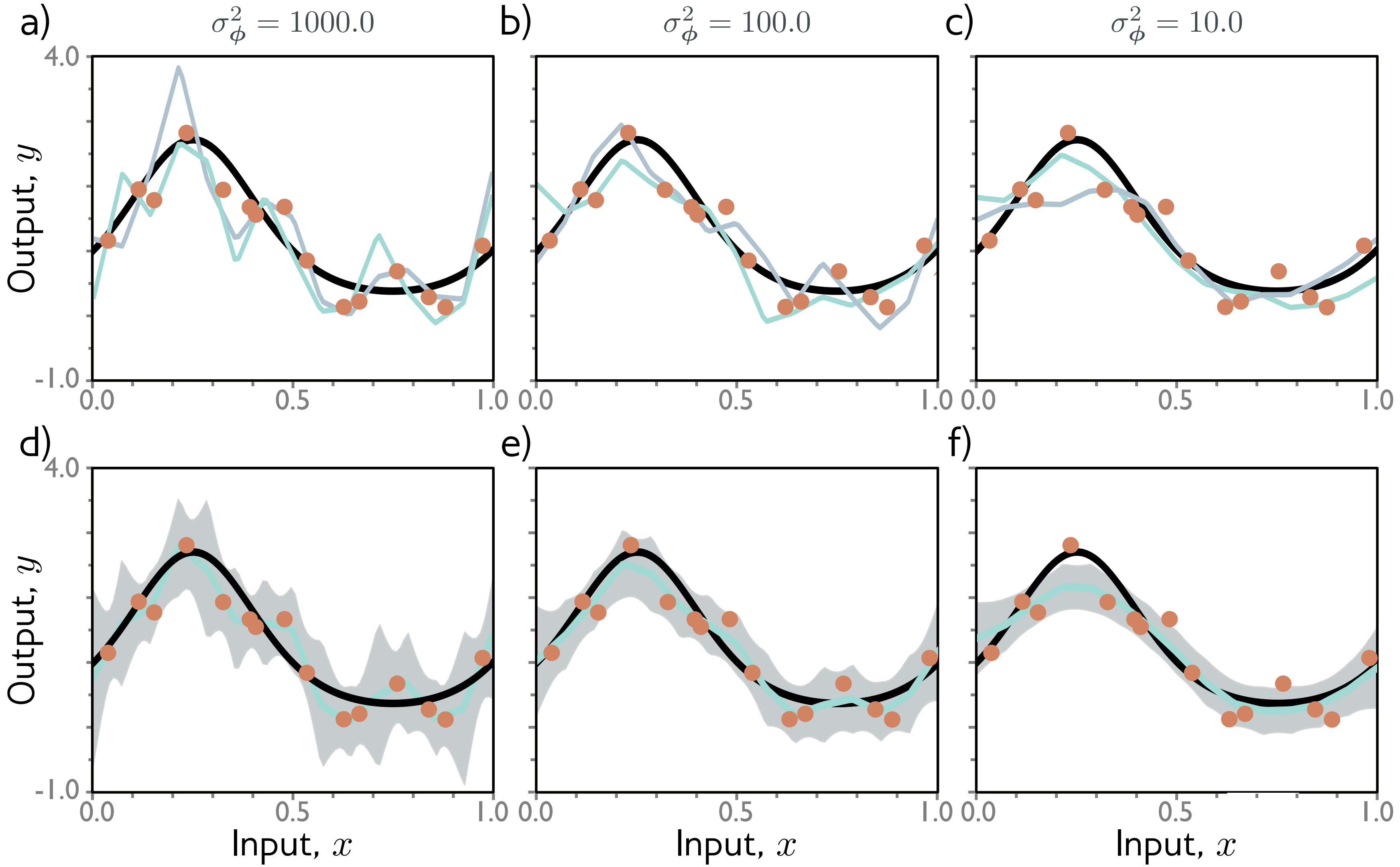
Prior info about parameters

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

- Take all possible parameters into account when make prediction

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

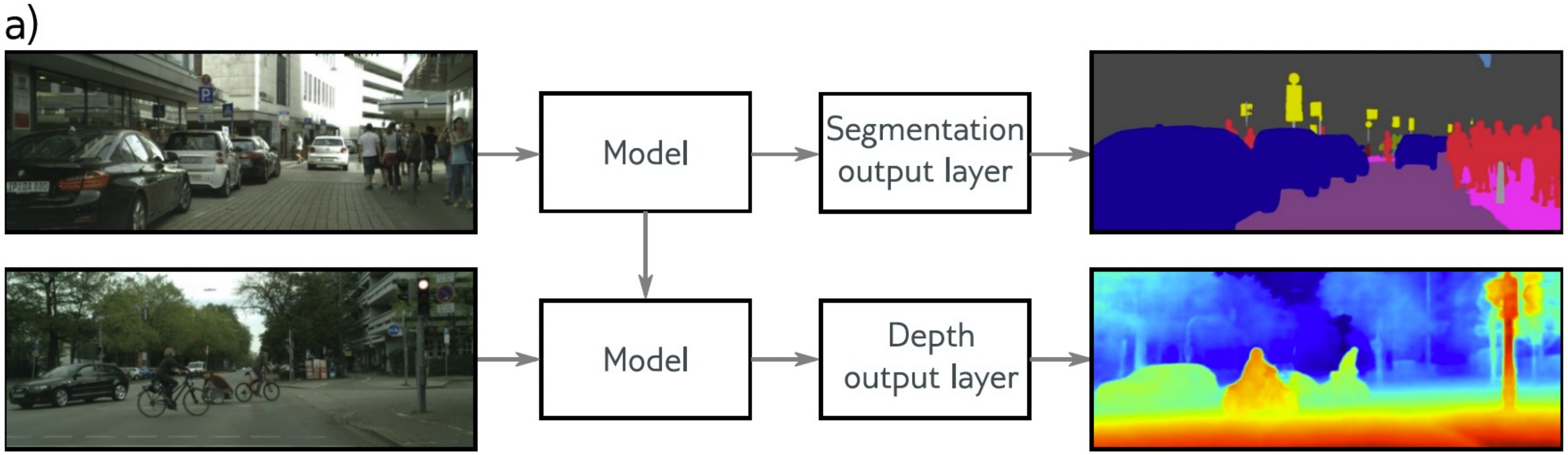
Bayesian approaches



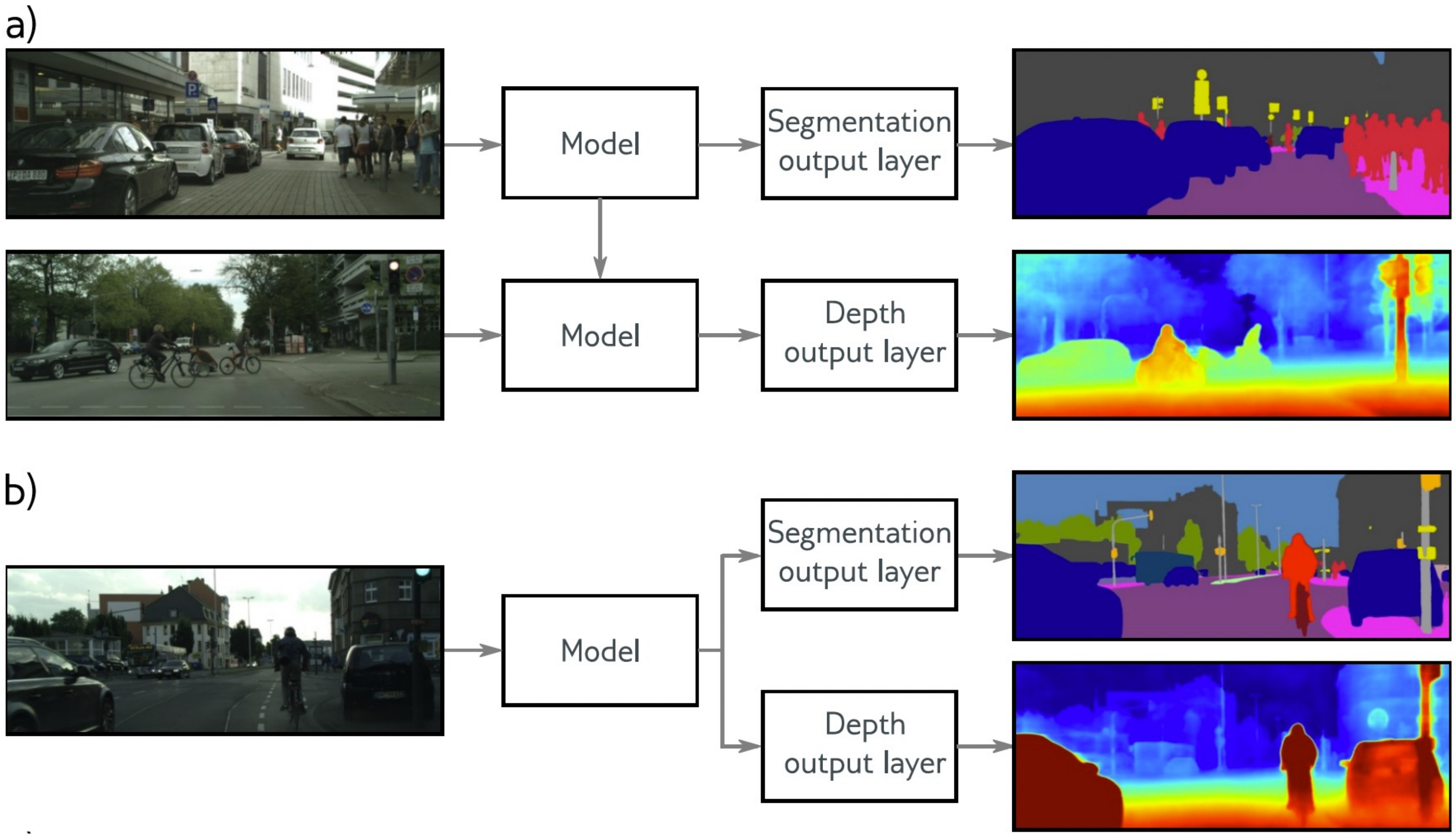
Regularization

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

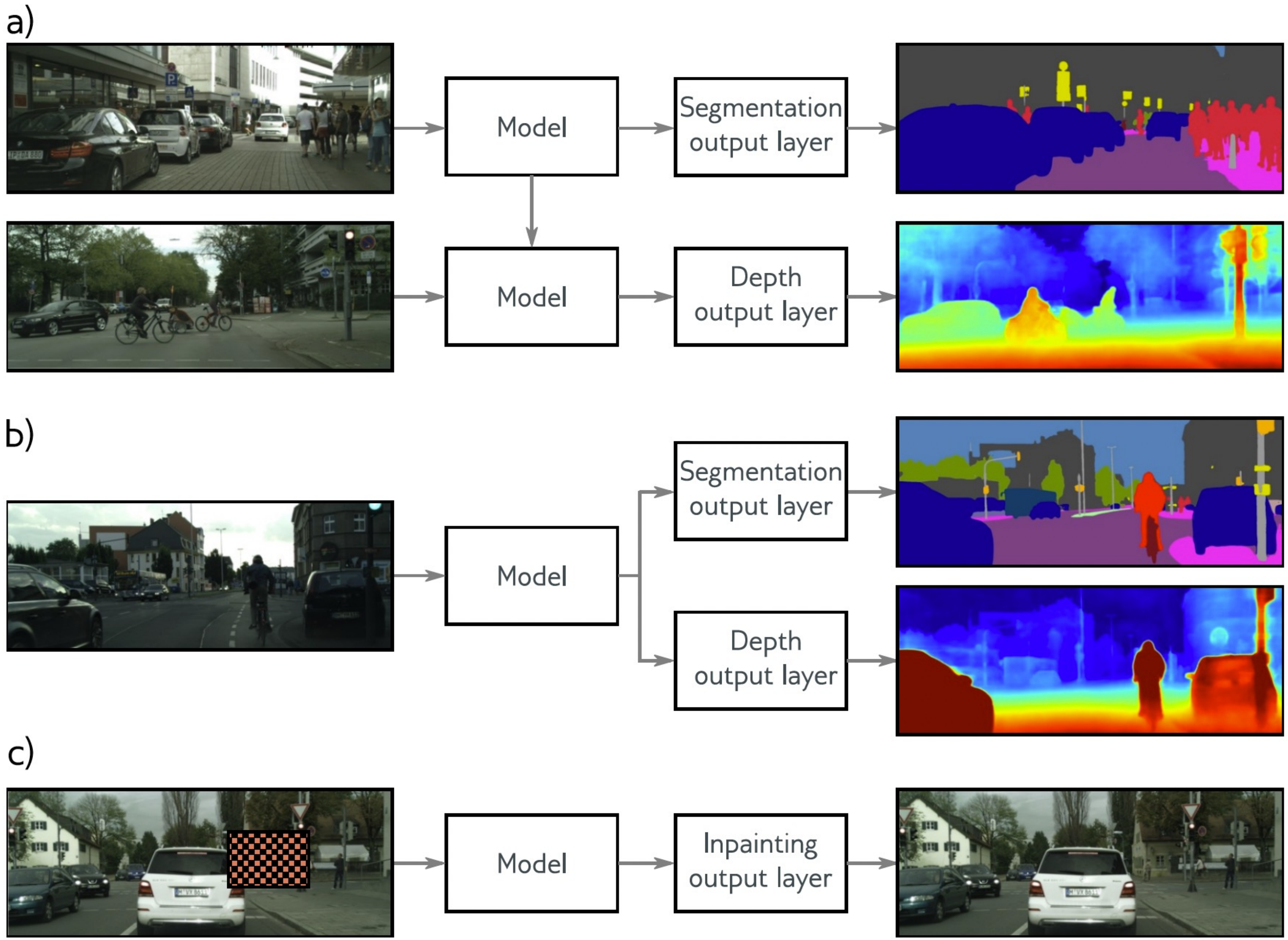
- Transfer learning



- Transfer learning
- Multi-task learning



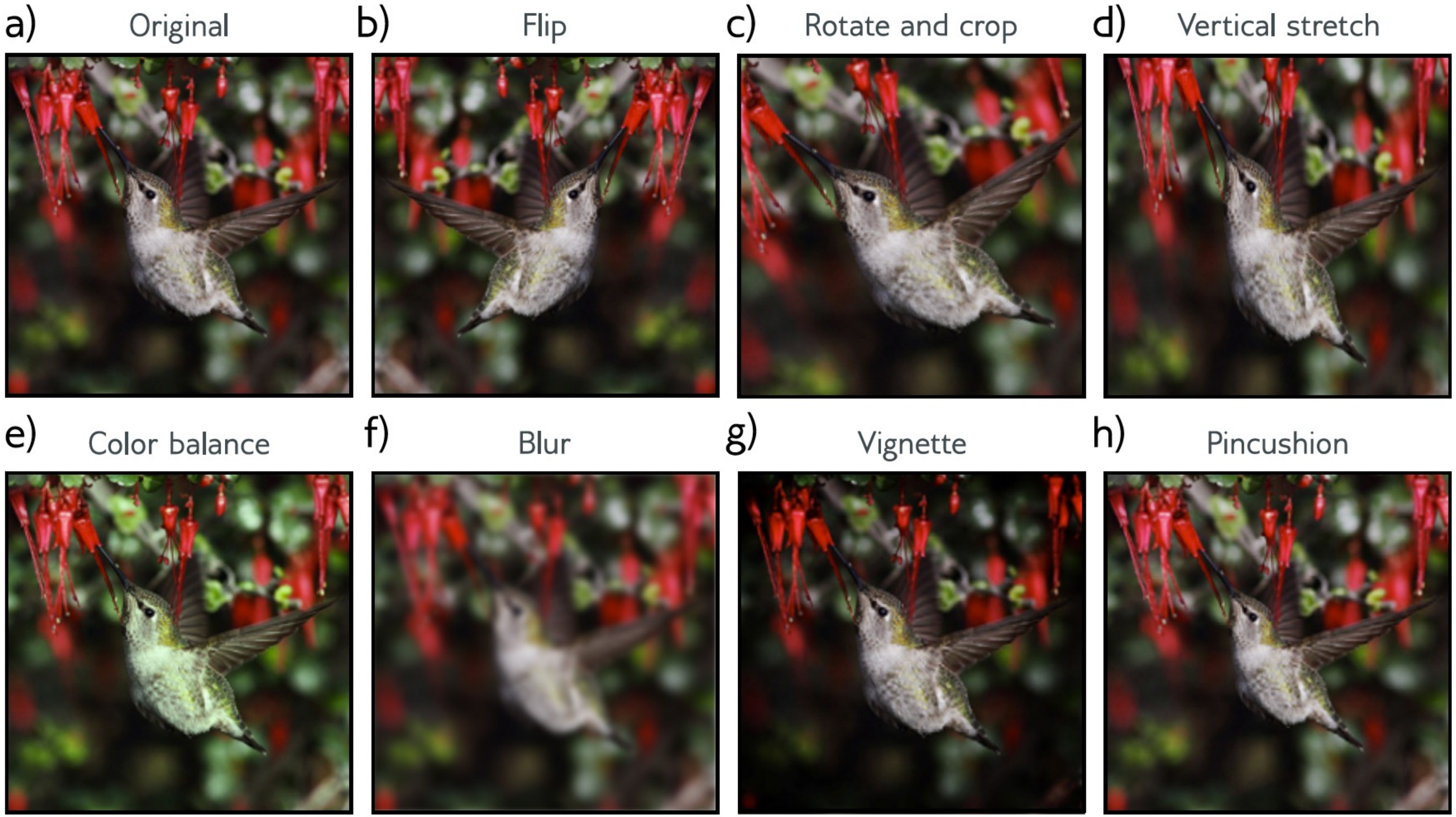
- Transfer learning
- Multi-task learning
- Self-supervised learning



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