Learning 101

Learning formulation and issues, regression, classification

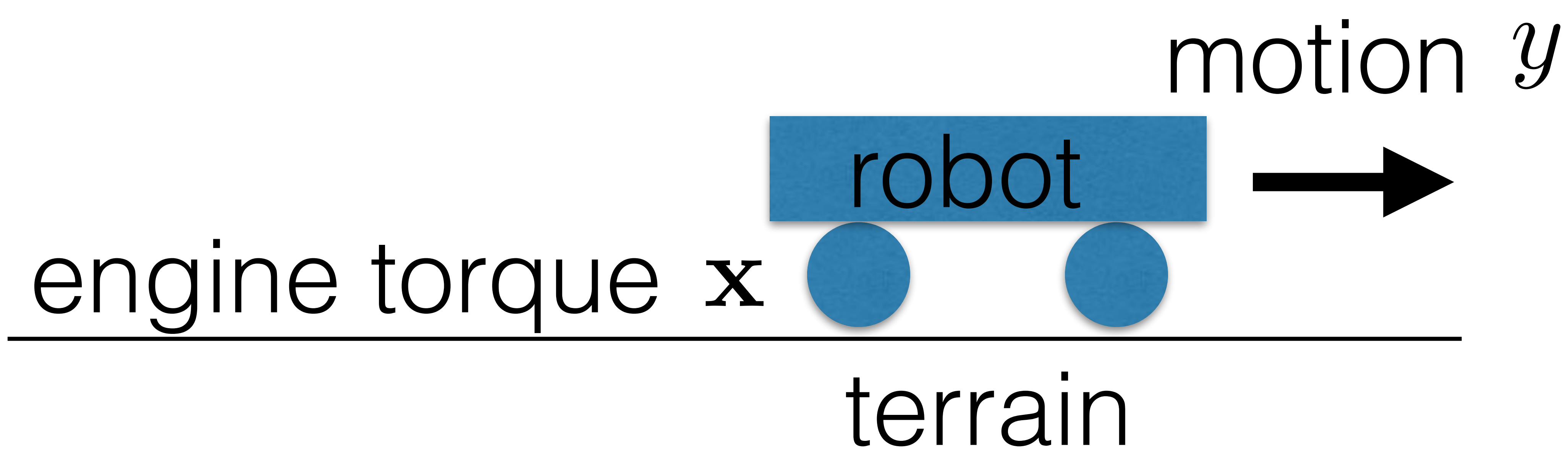
Pre-requisites:

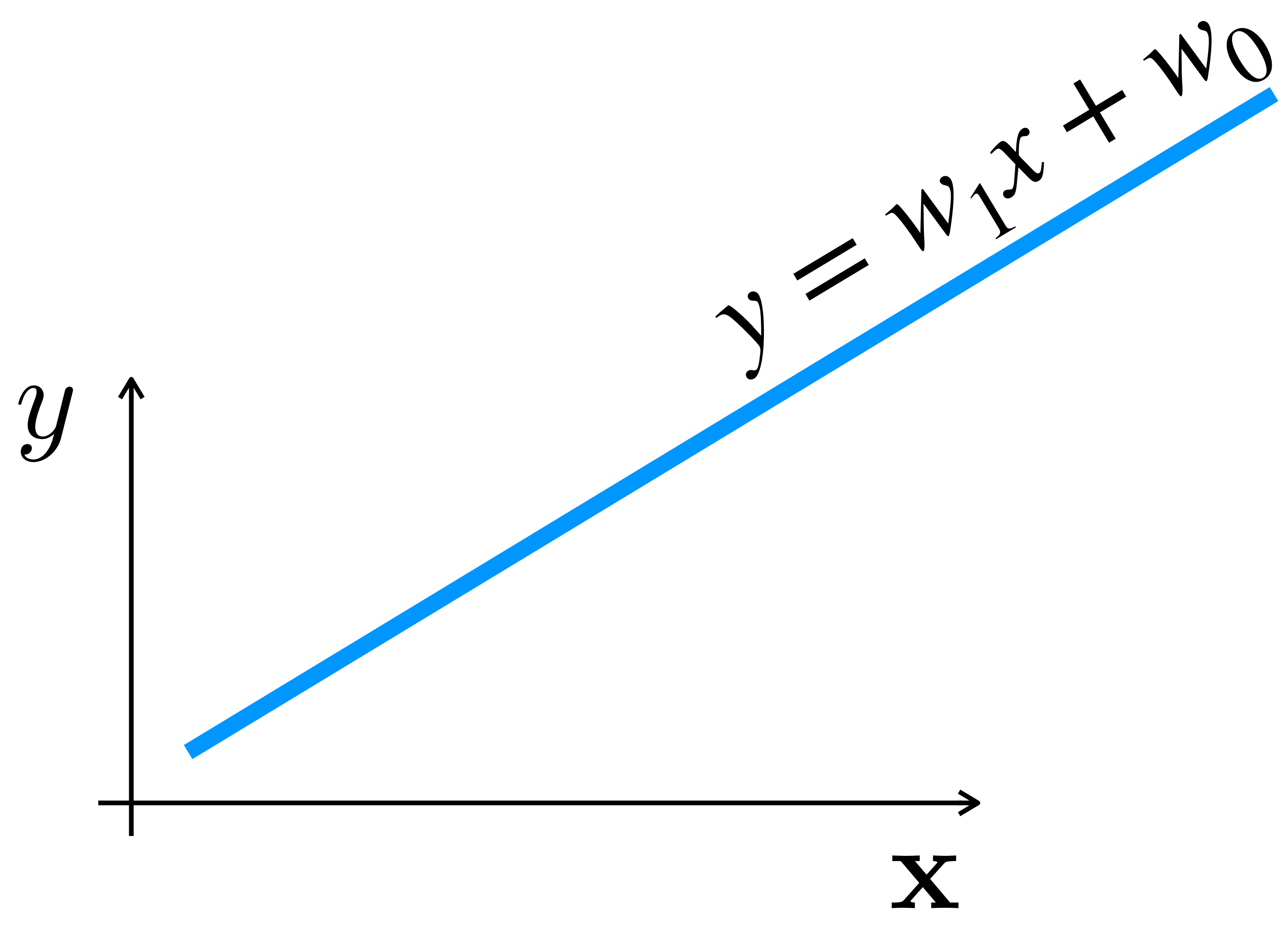
- linear algebra,

Karel Zimmermann
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

Motivation example: estimation of a motion model

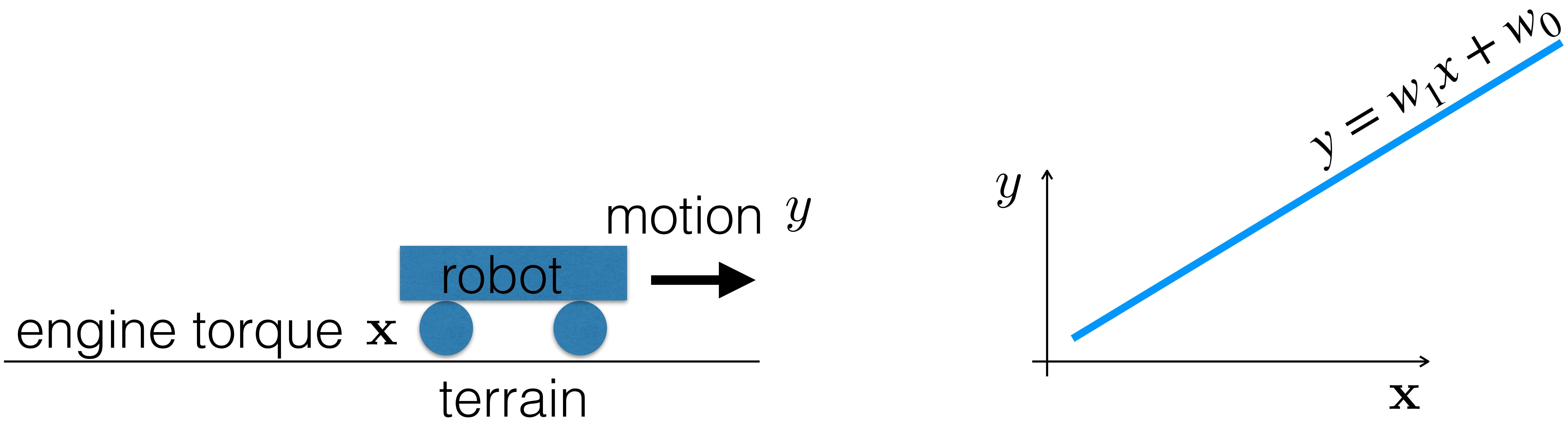
- What do I need to build a motion model?
- Algorithm that maps x on y (or prob distr of y)





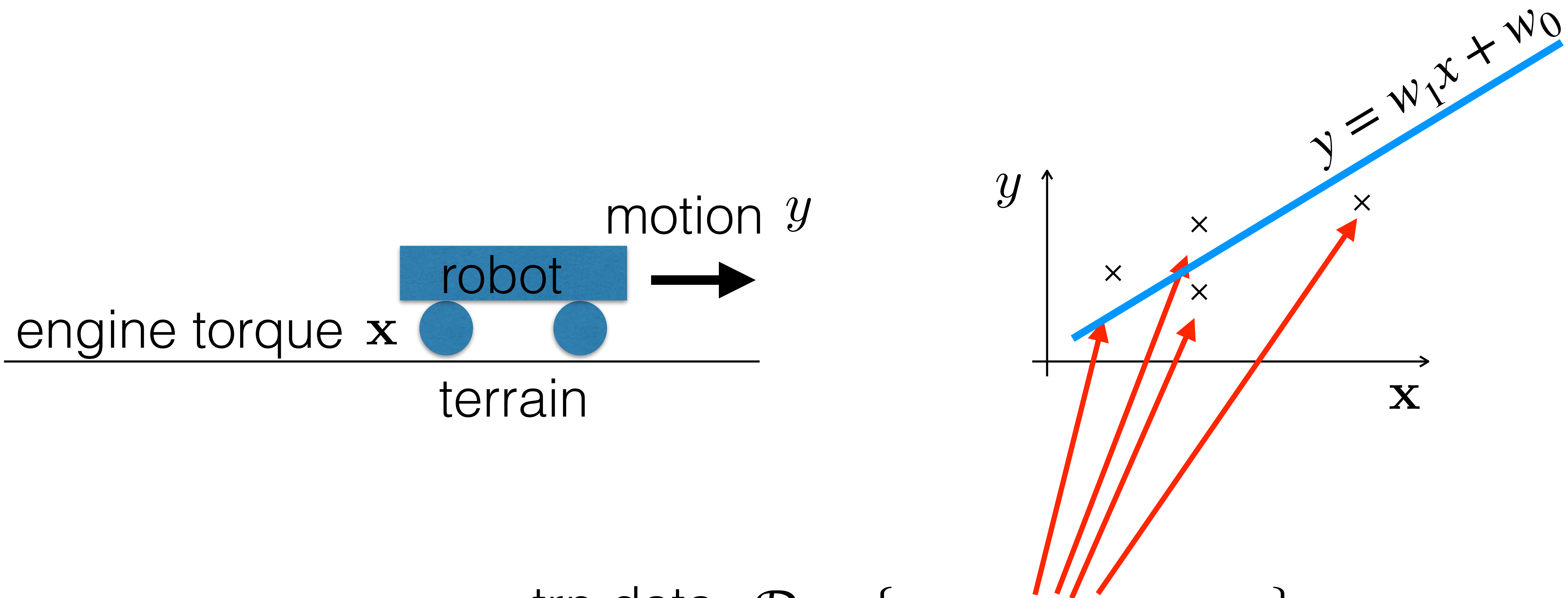
Motivation example: estimation of a motion model

- What do I need to build a motion model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? => trn data+loss+opt



Motivation example: estimation of a motion model

- What do I need to build a motion model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? =>   + loss+opt

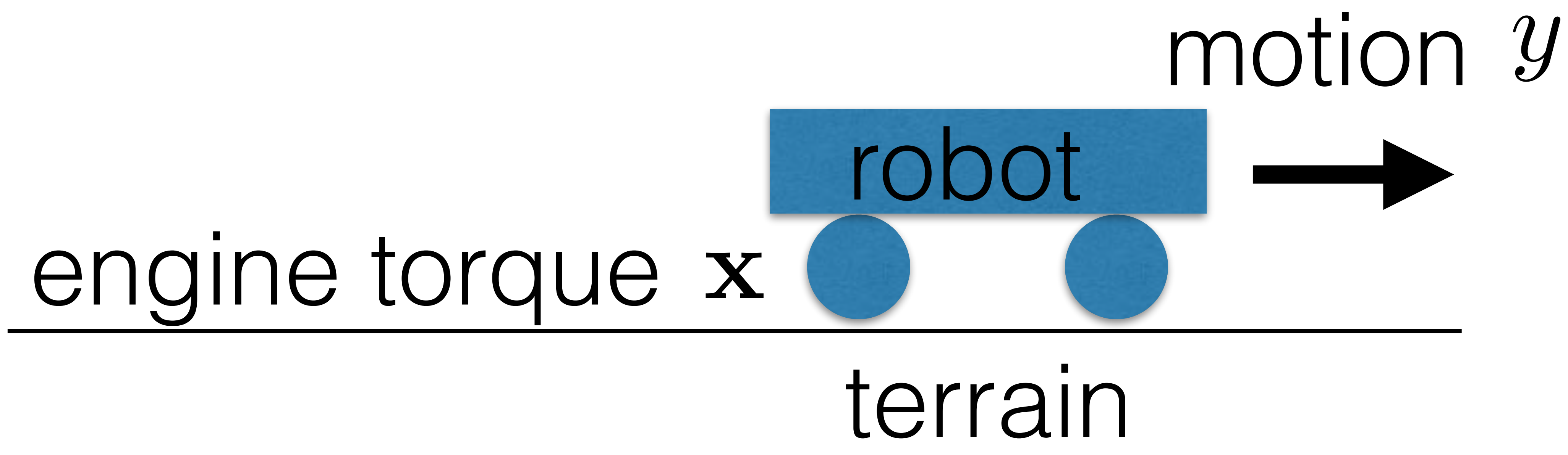


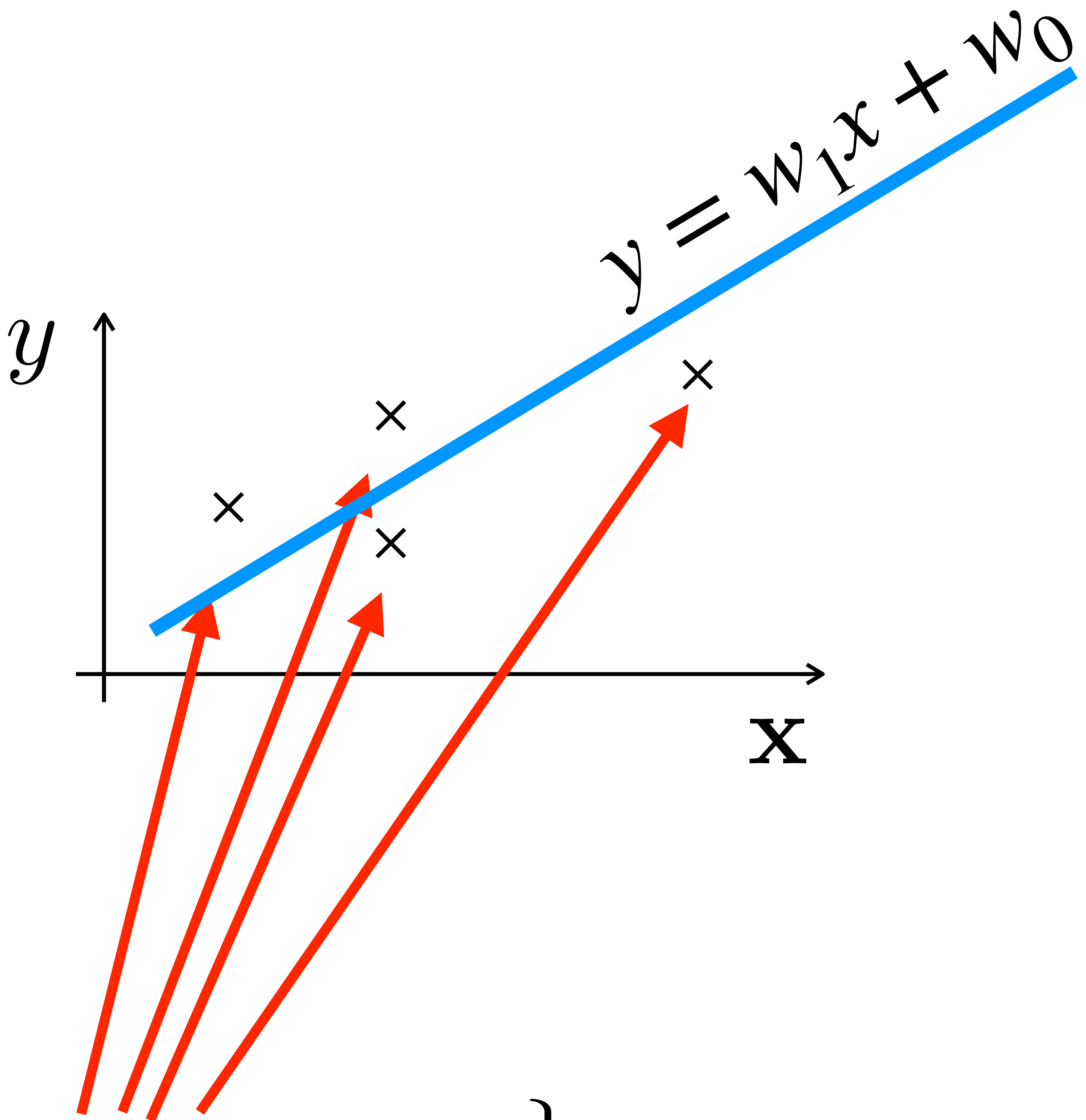
trn data $\mathcal{D} = \{\mathbf{x}_1, y_1 \dots \mathbf{x}_N, y_N\}$

Motivation example: estimation of a motion model

- Let's implement it!   loss = **???**

opt = **???**





trn data $\mathcal{D} = \{\mathbf{x}_1, y_1 \dots \mathbf{x}_N, y_N\}$

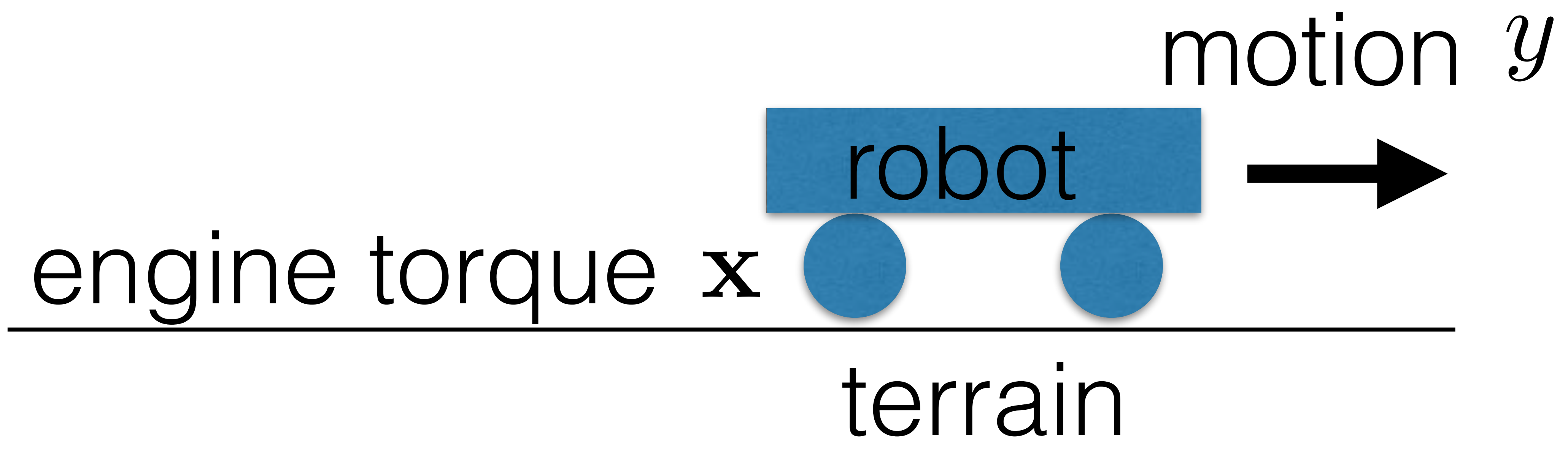
Motivation example: estimation of a motion model

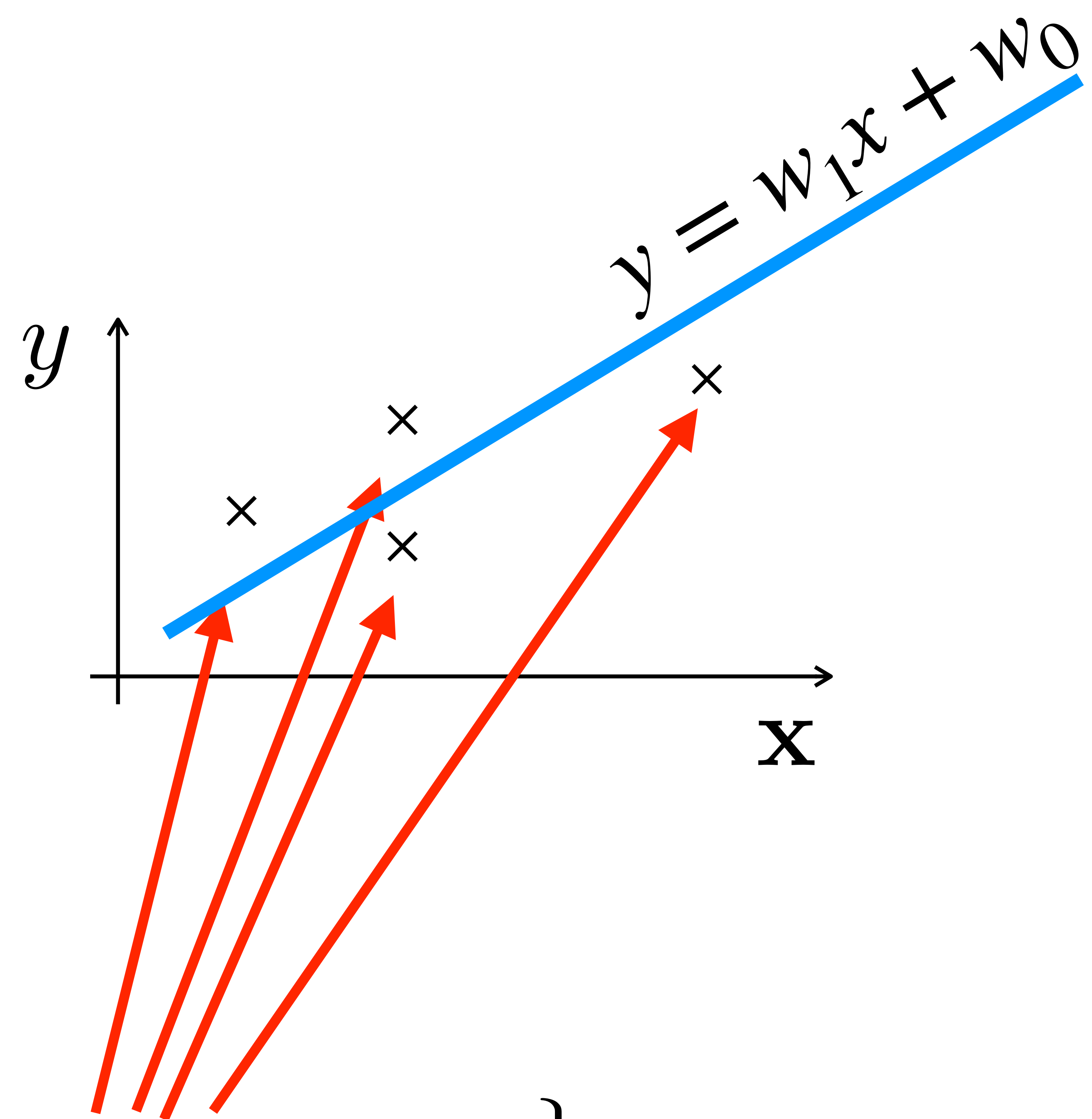
- Let's implement it!

loss: $\arg\min_{\mathbf{w}} \sum_i (\boxed{w_1 x_i + w_0} - \boxed{y_i})^2$

opt =

```
w = np.array([-2.0, 2.0])
for i in range(0, 10):
    dy = w[0] * x + w[1] - y
    loss = np.sum(dy * dy)
    w = w - 0.1 * grad(loss, w)
```





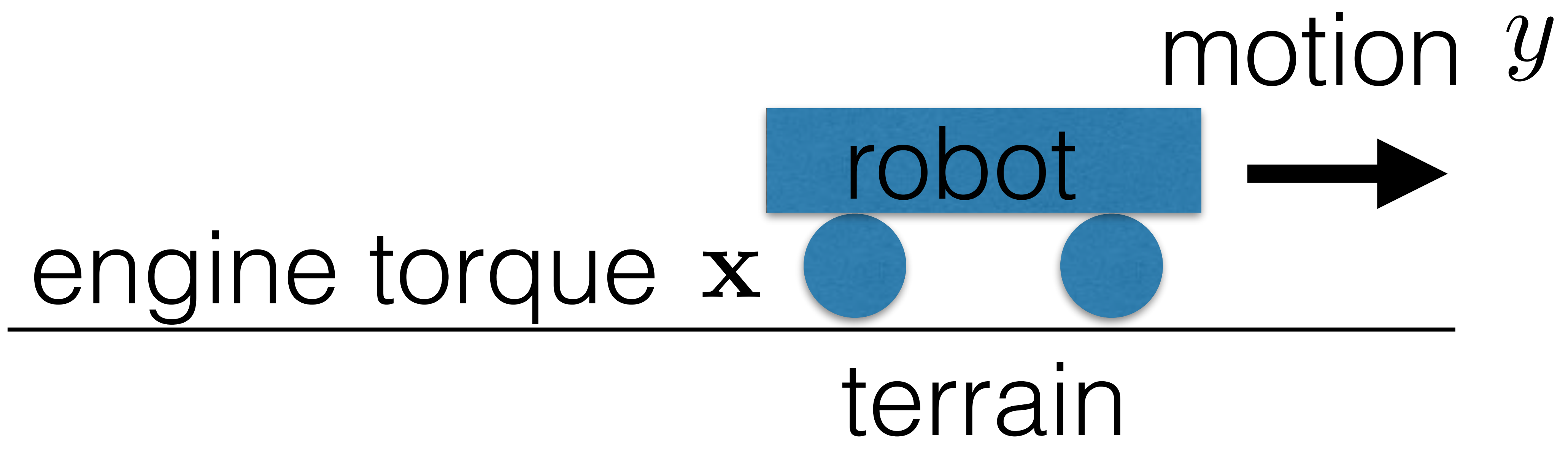
trn data $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

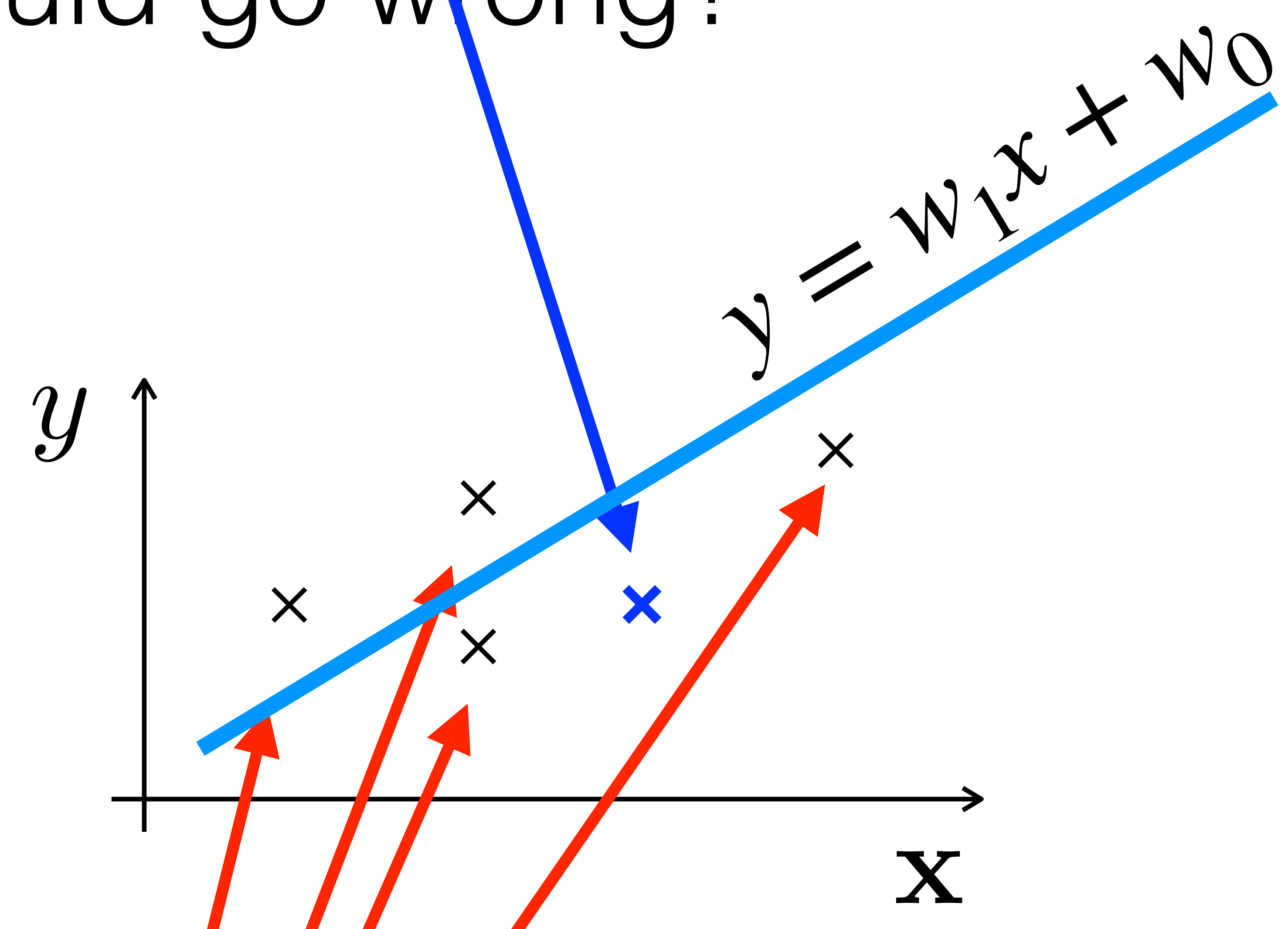
Motivation example: estimation of a motion model

- What do I need to build a motion model?
- Algorithm that maps x on y (or prob distr of y)
- This algorithm has some parameters => how to find them? => loss+trn data+opt

SOLVED

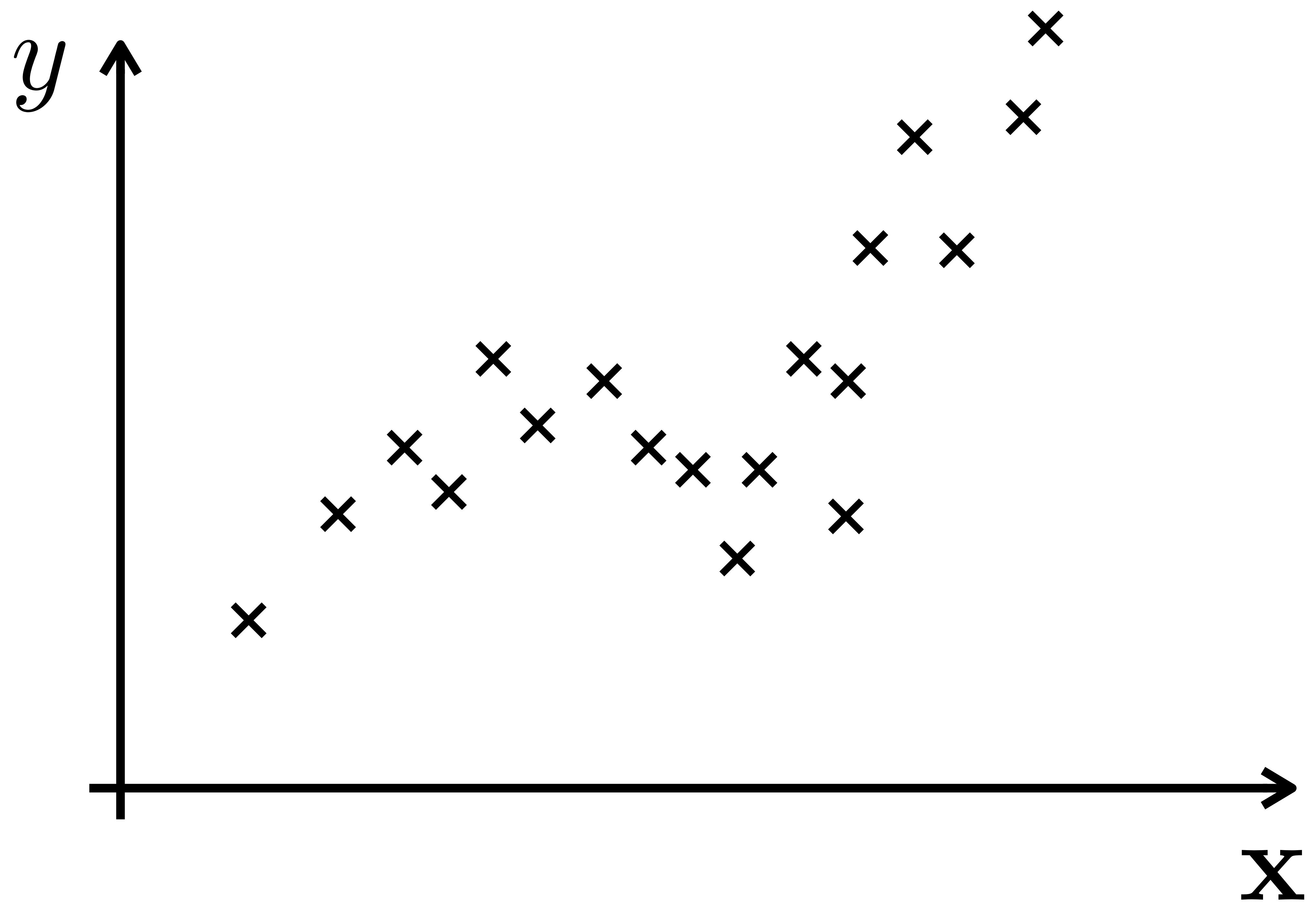
- How to decide that the algorithm works well? => tst data
- What if the algorithm does not work well? What could go wrong?





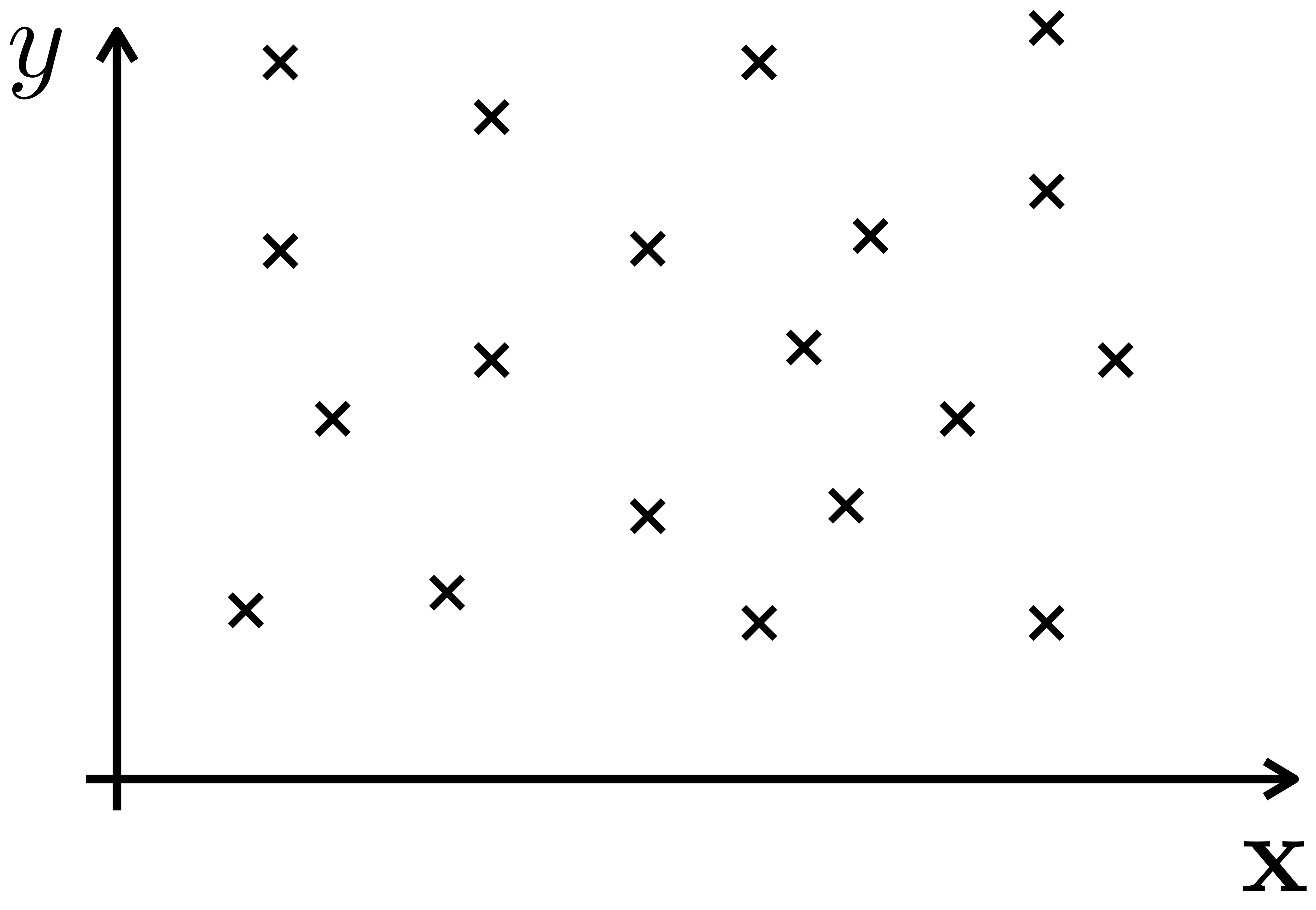
trn data $\mathcal{D} = \{\mathbf{x}_1, y_1 \dots \mathbf{x}_N, y_N\}$

What can go wrong: **inputs x does not allow to predict y**



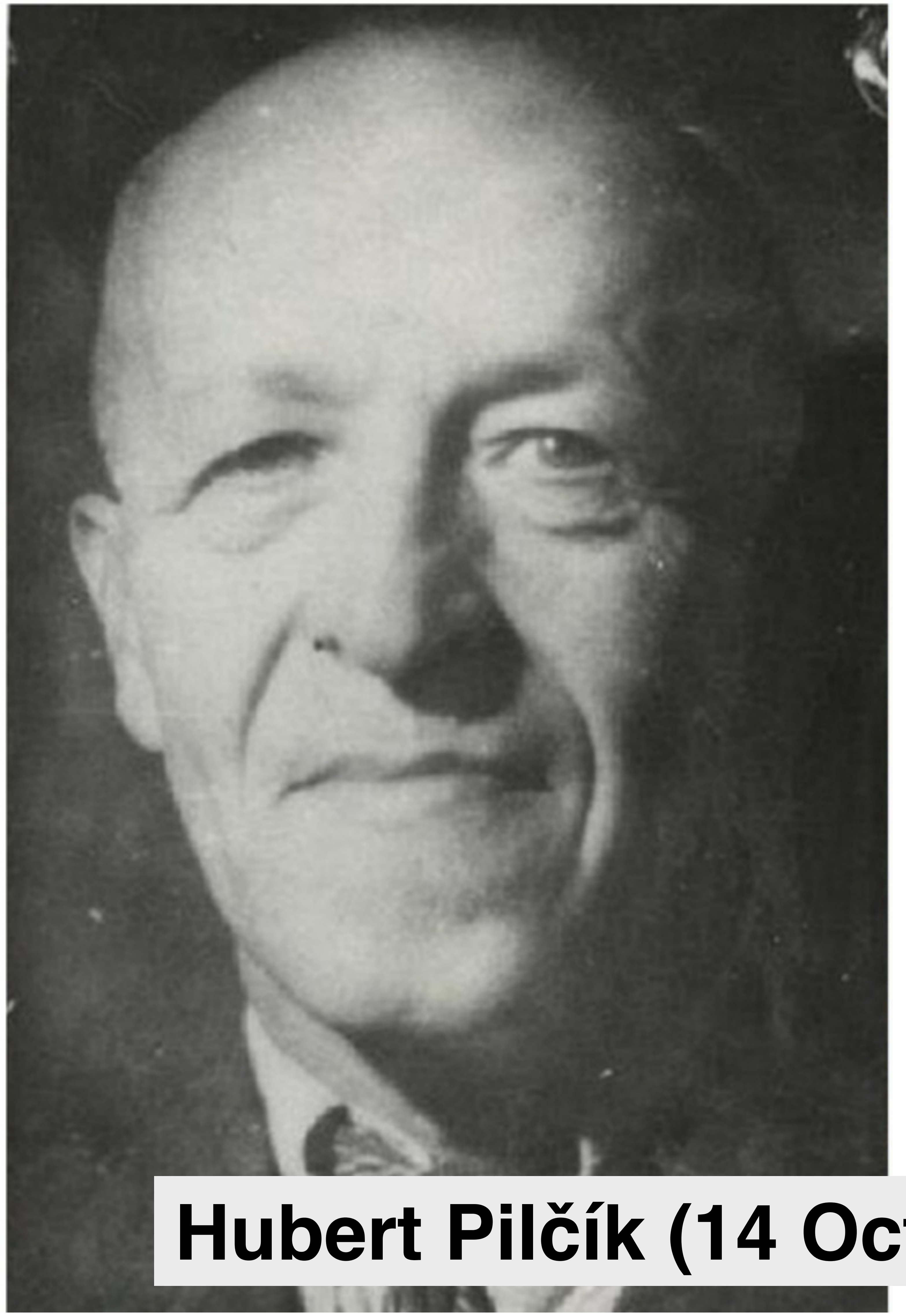
predicting person's age from face image

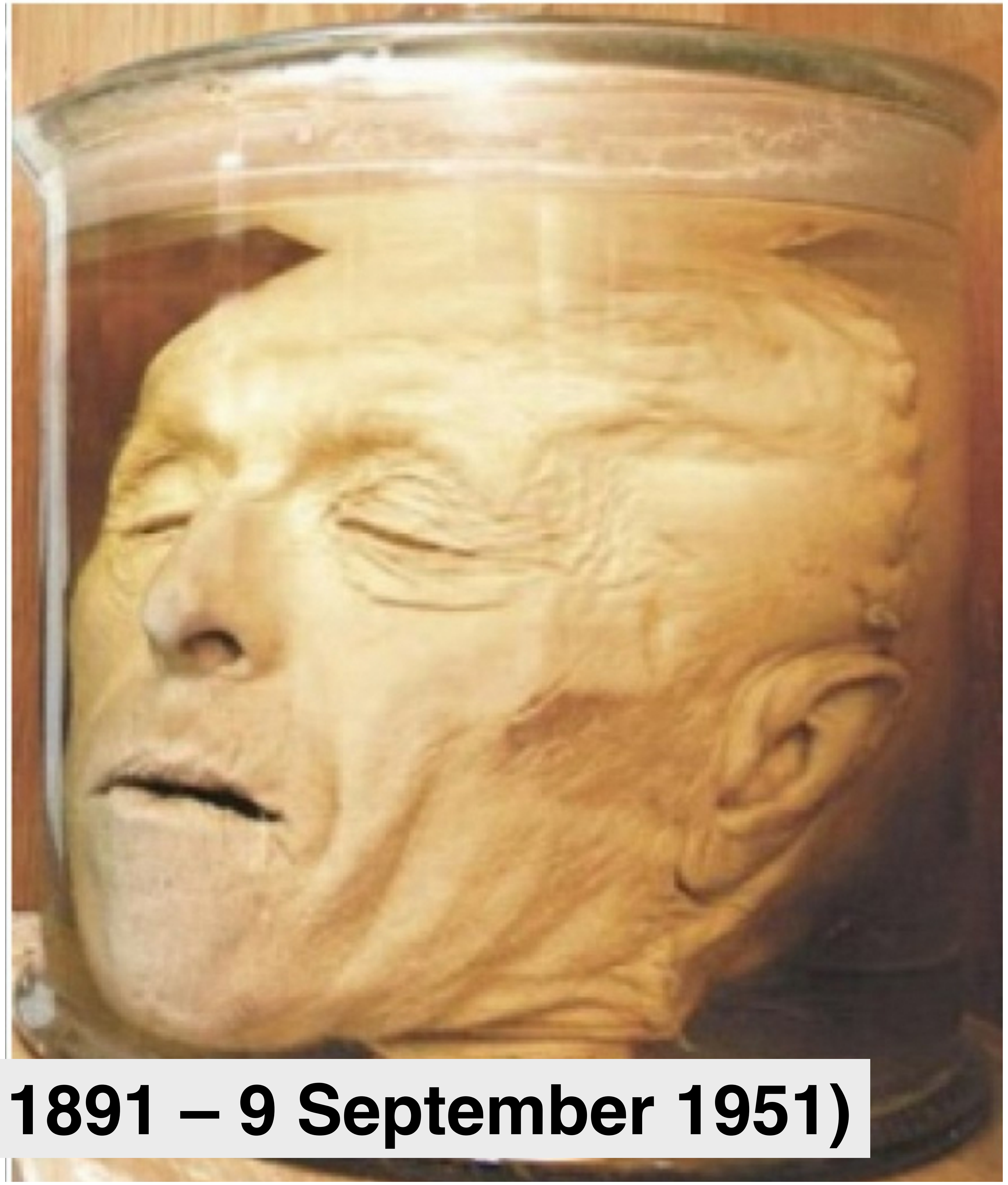
vs



predicting oil prices

What can go wrong: **inputs x does not allow to predict y**





Hubert Pilčík (14 October 1891 – 9 September 1951)

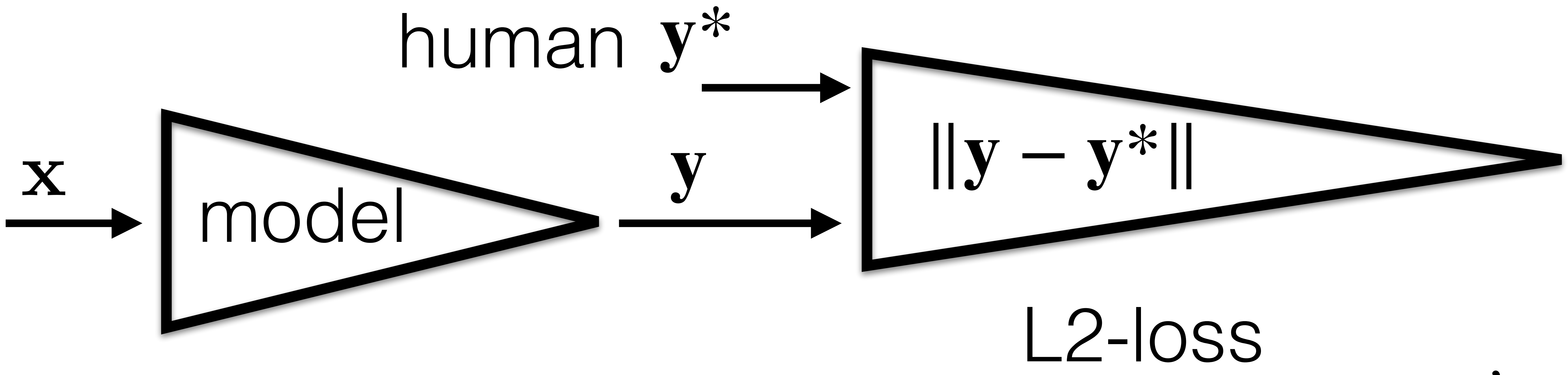
A Deep Neural Network Model to Predict Criminality Using Image Processing

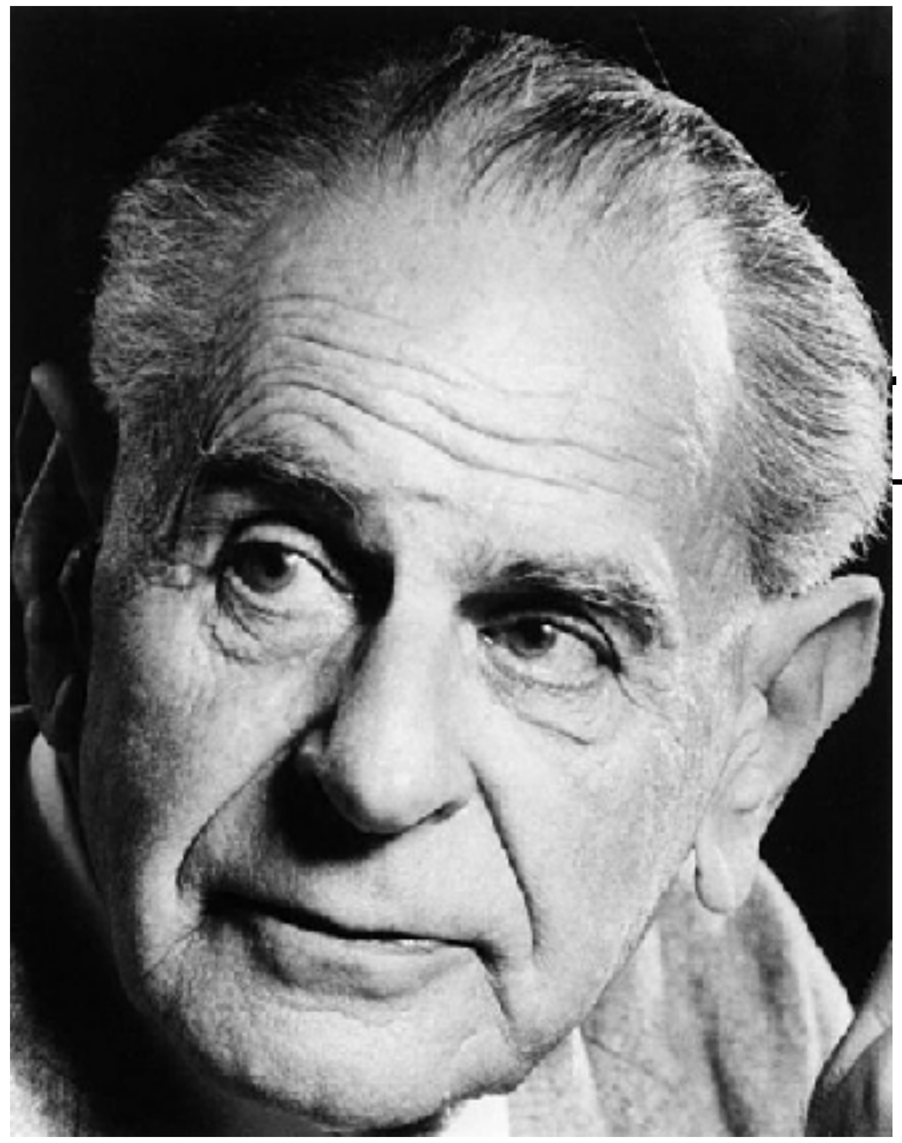
https://medium.com/@CoalitionForCriticalTechnology/abolish-the-techtoprisonpipeline-9b5b14366b16

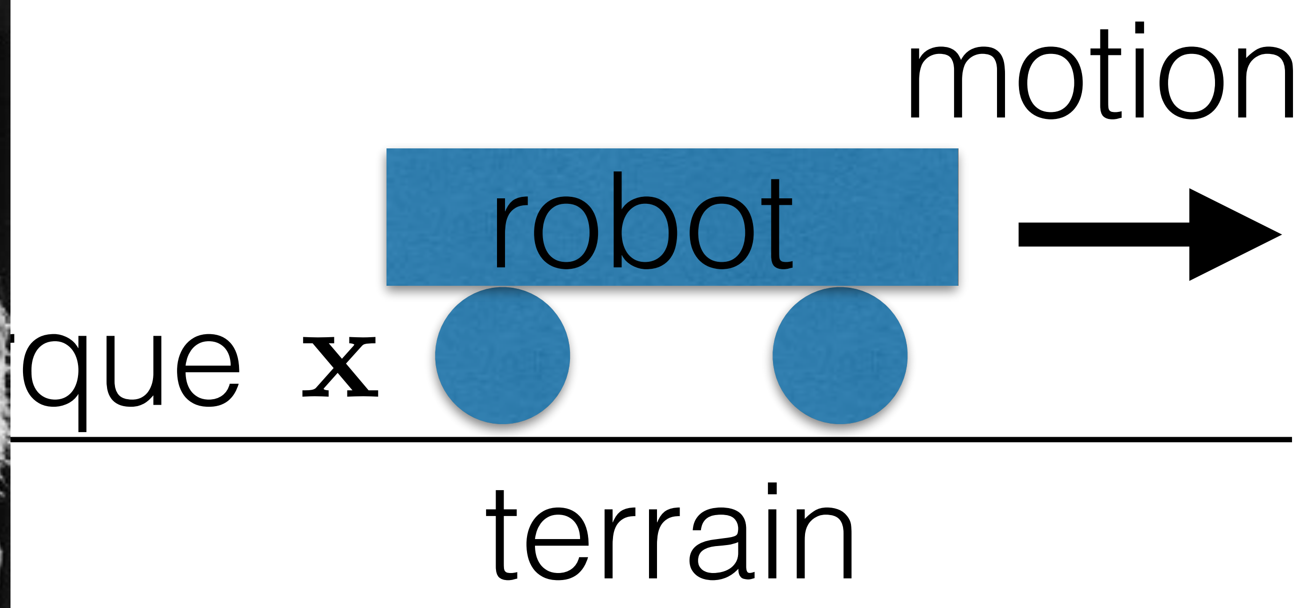
What can go wrong: **trn/tst data distribution mismatch**

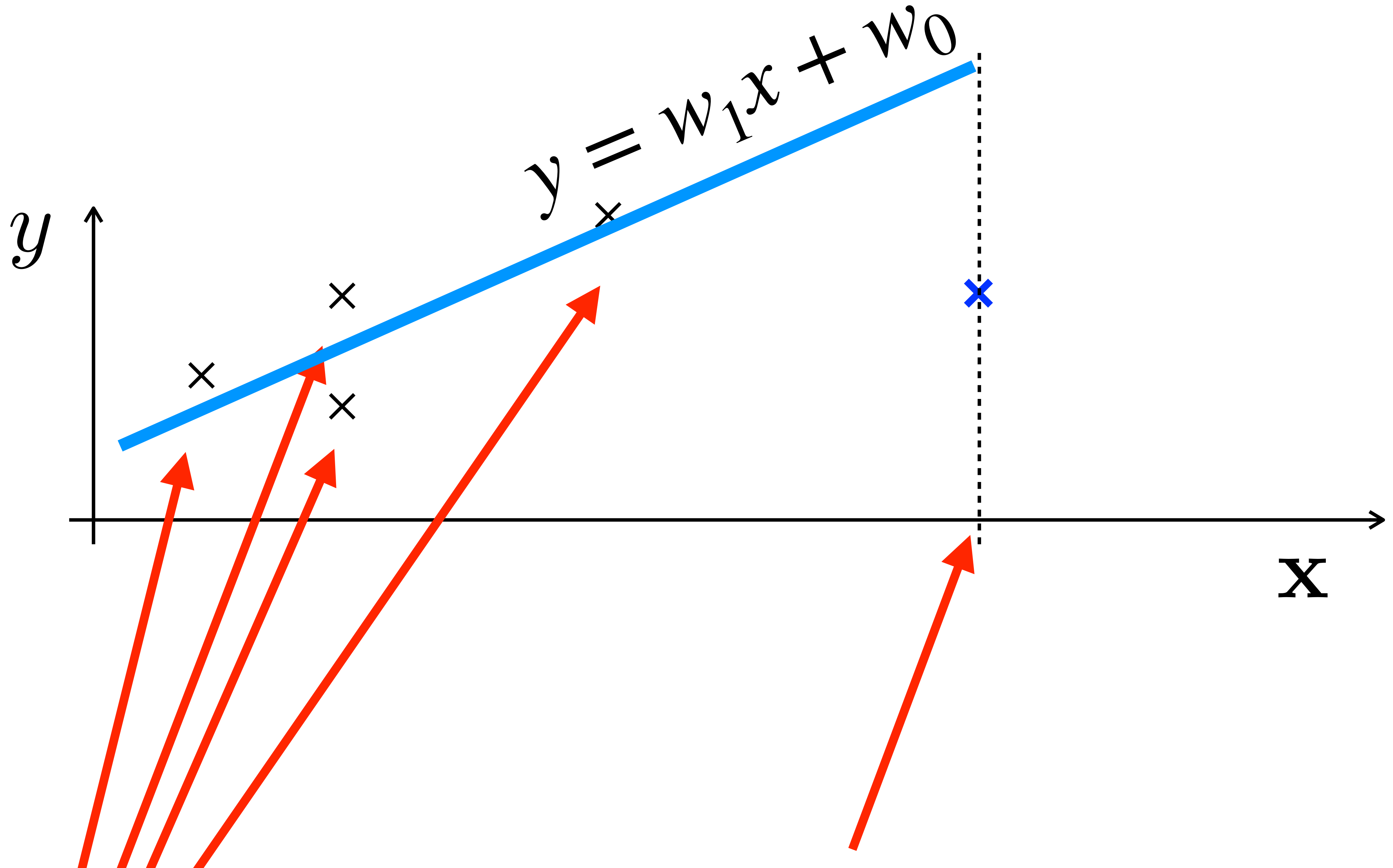
[NVidia, CVPR, 2016]

Statistical consistency issues





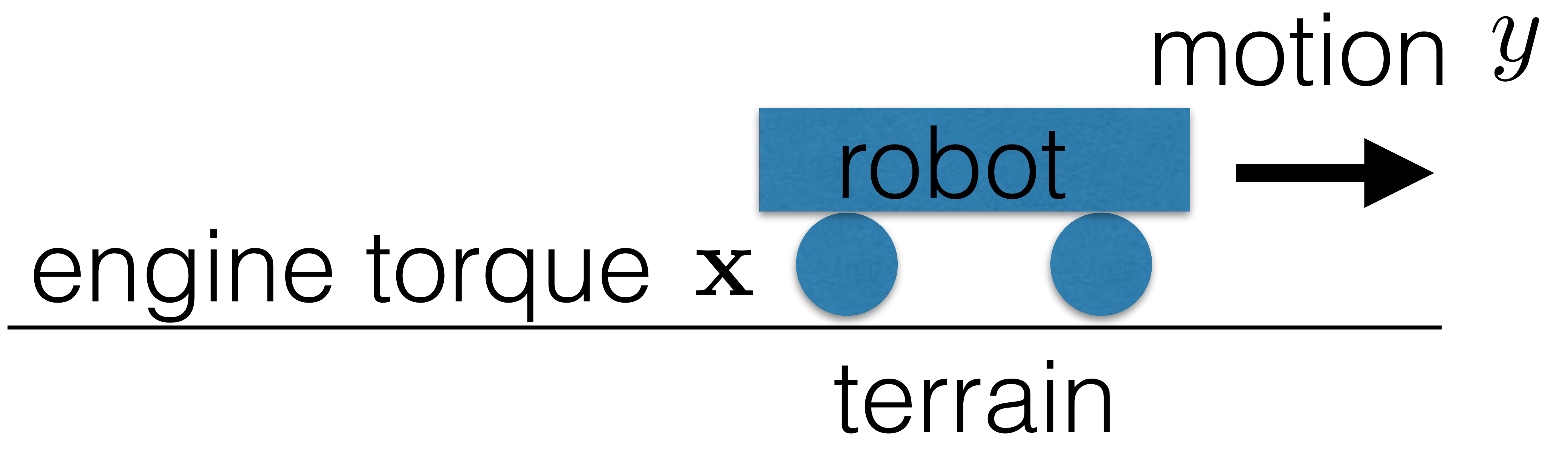




trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

tst data:

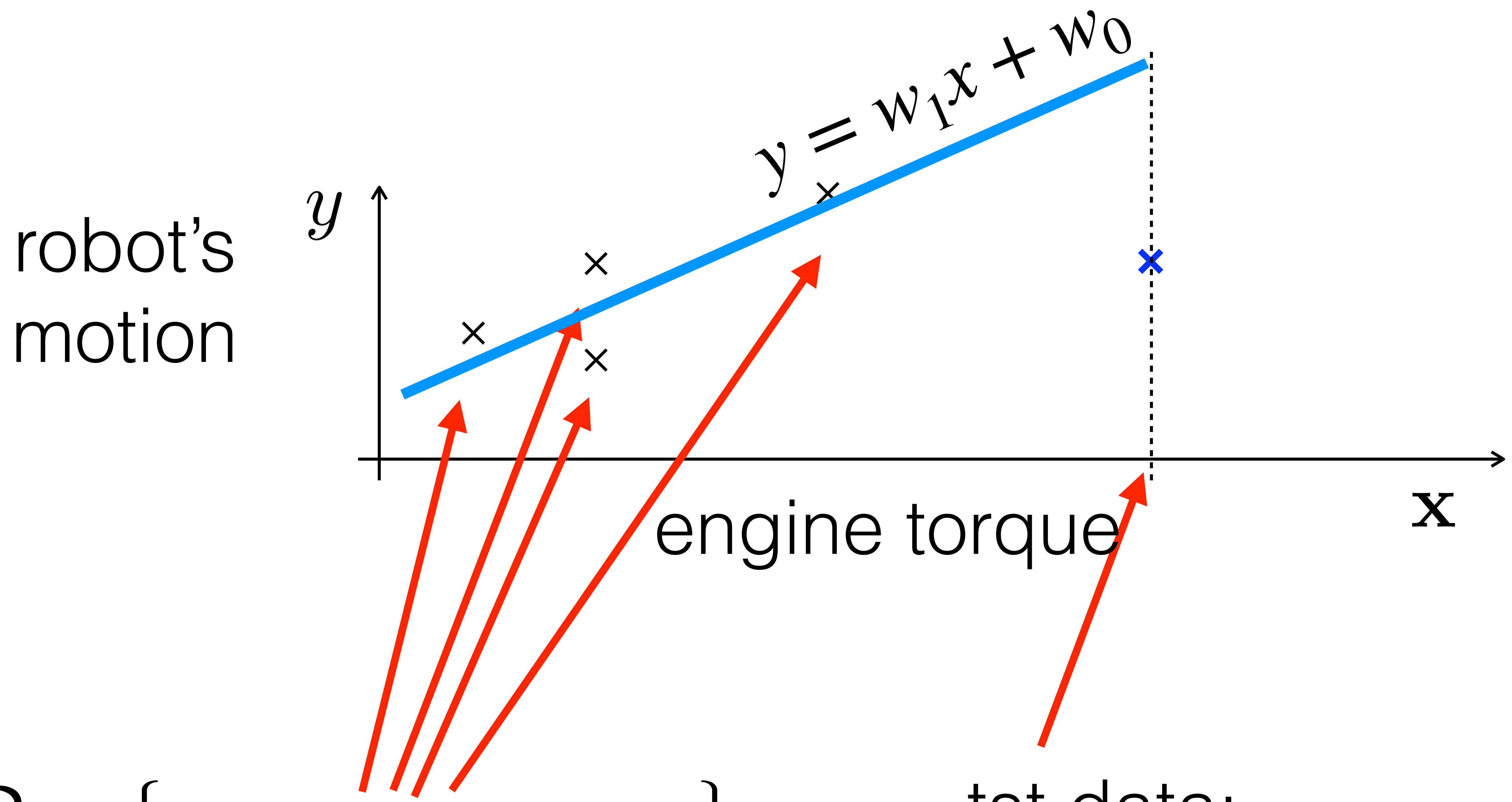
What can go wrong: **inappropriate model**



Underfitting happens due to:

- oversimplified models
- inability to measure important features

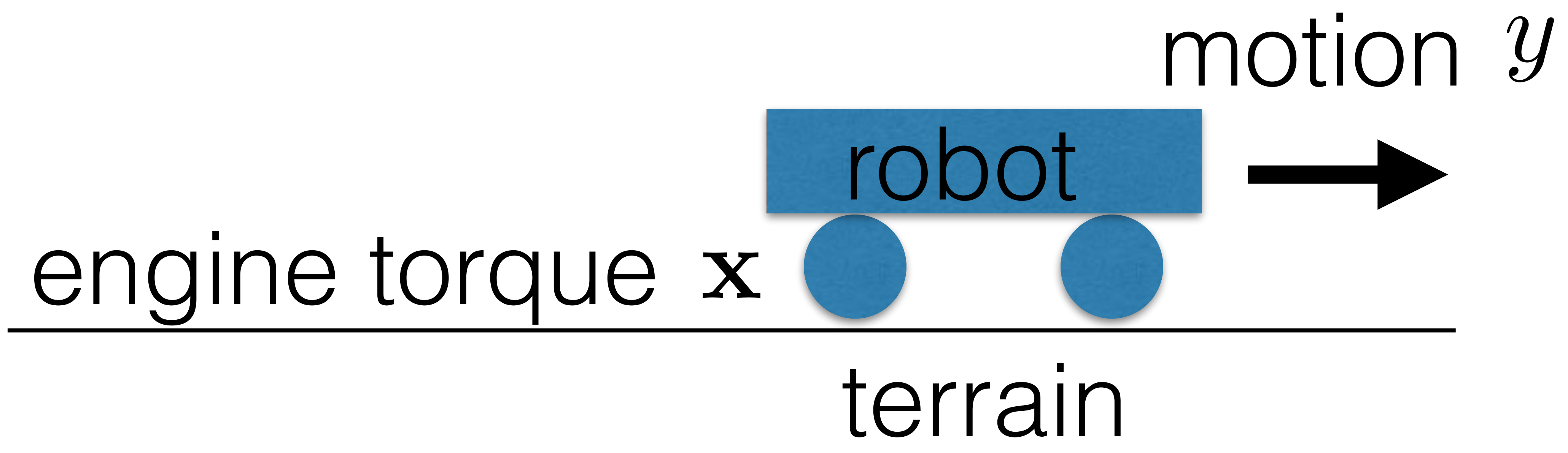
linear function => underfitting



trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

tst data:

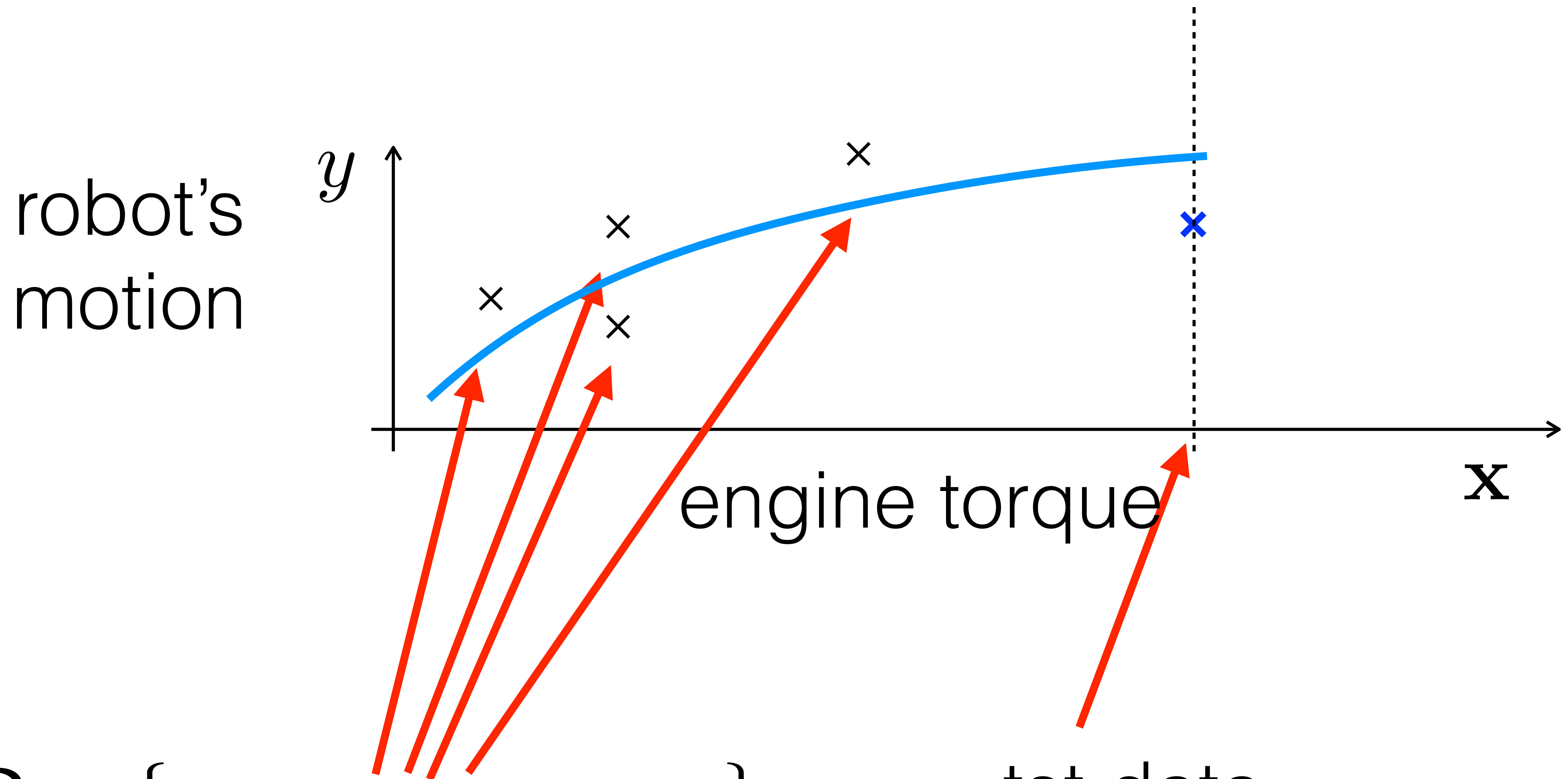
What can go wrong: **inappropriate model**



log function => good fit

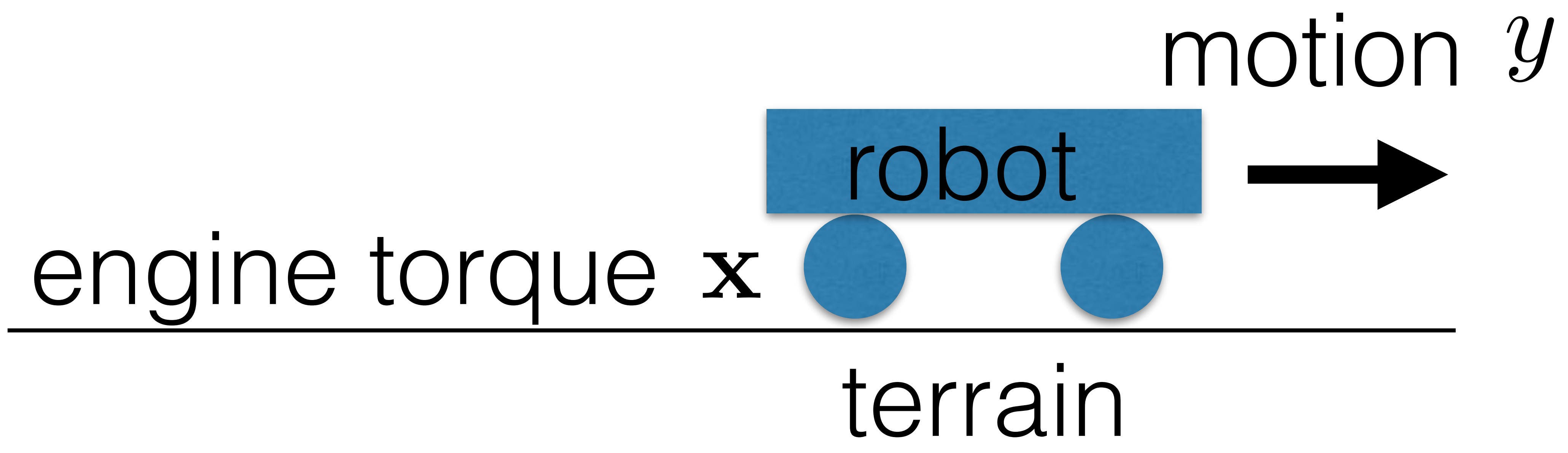
Good model provides:

- good generalization
  (less sensitive to trn/tst mismatch)



trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$ tst data:

What can go wrong: **inappropriate model**

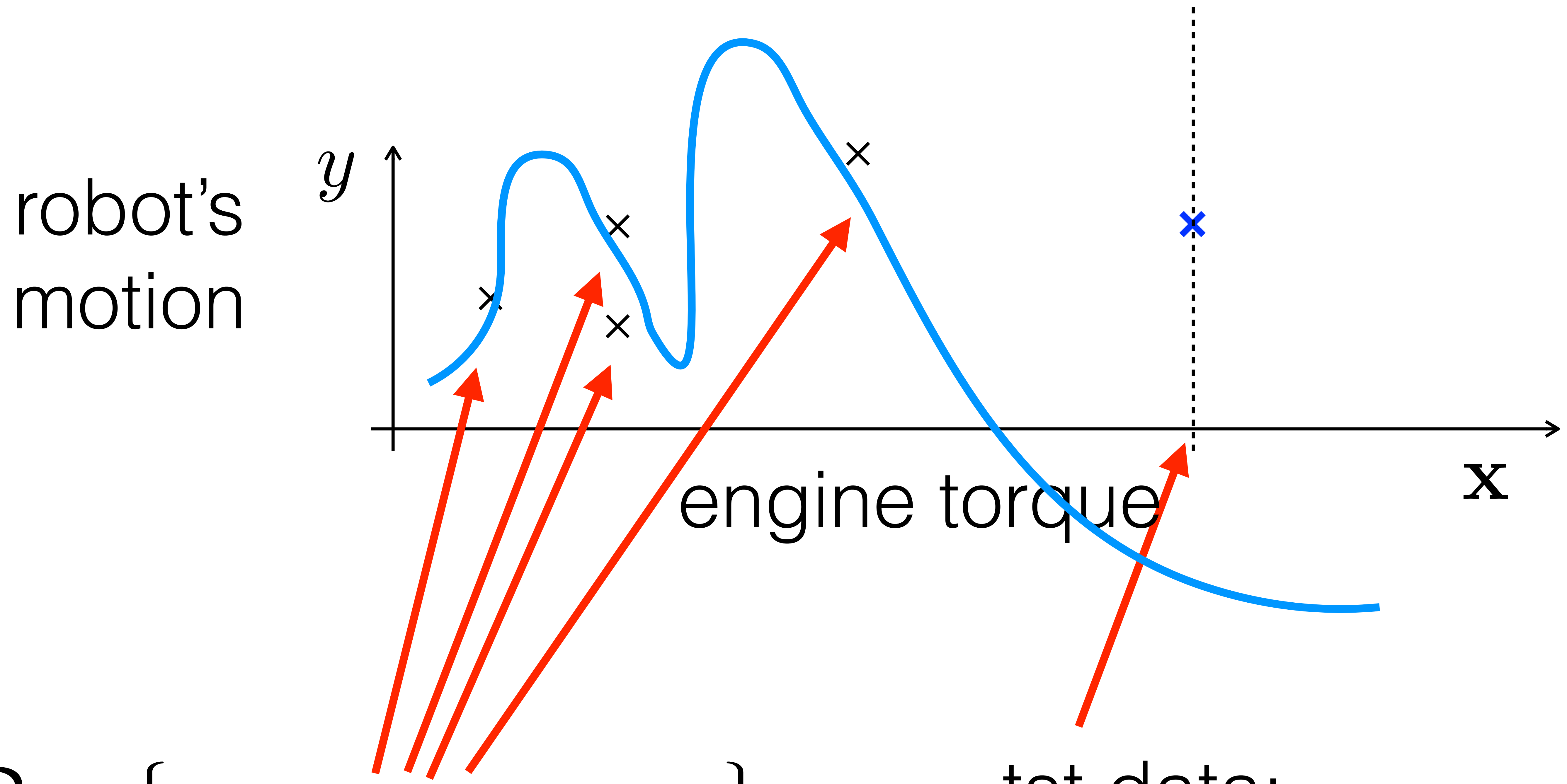


Overfitting happens due to:

- model complexity
- effort to interpret noise (in features that has no connection with the problem)

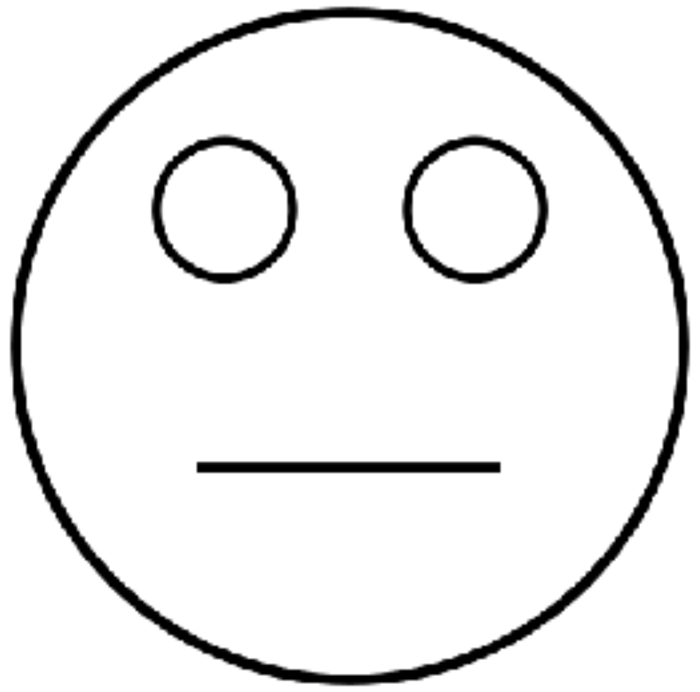
Do humans overfit?

complicated function=>overfitting



trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$ tst data:

What can go wrong: **inappropriate model**

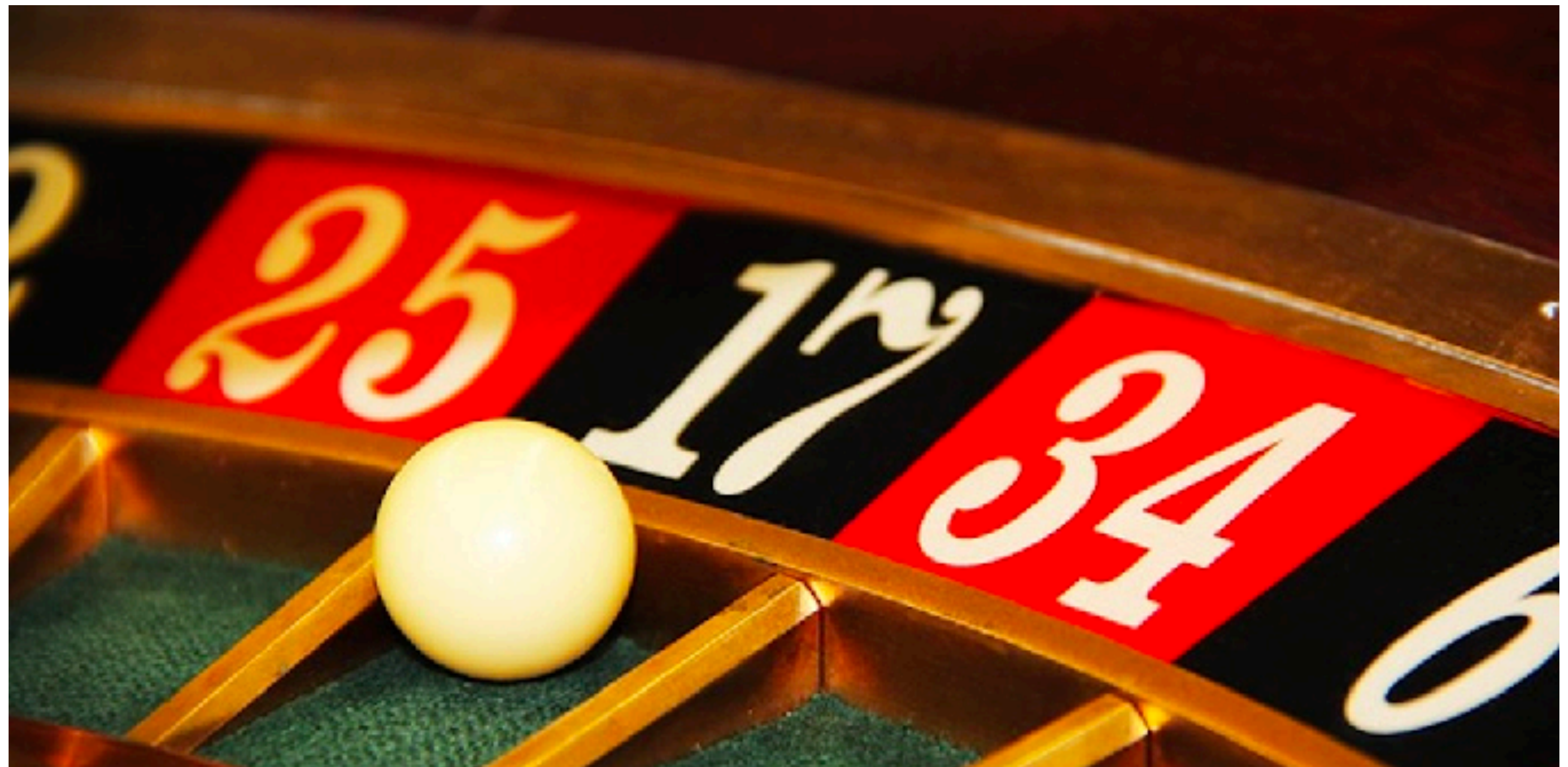


Overfitting happens due to:

- model complexity
- bad features

Do humans overfit?

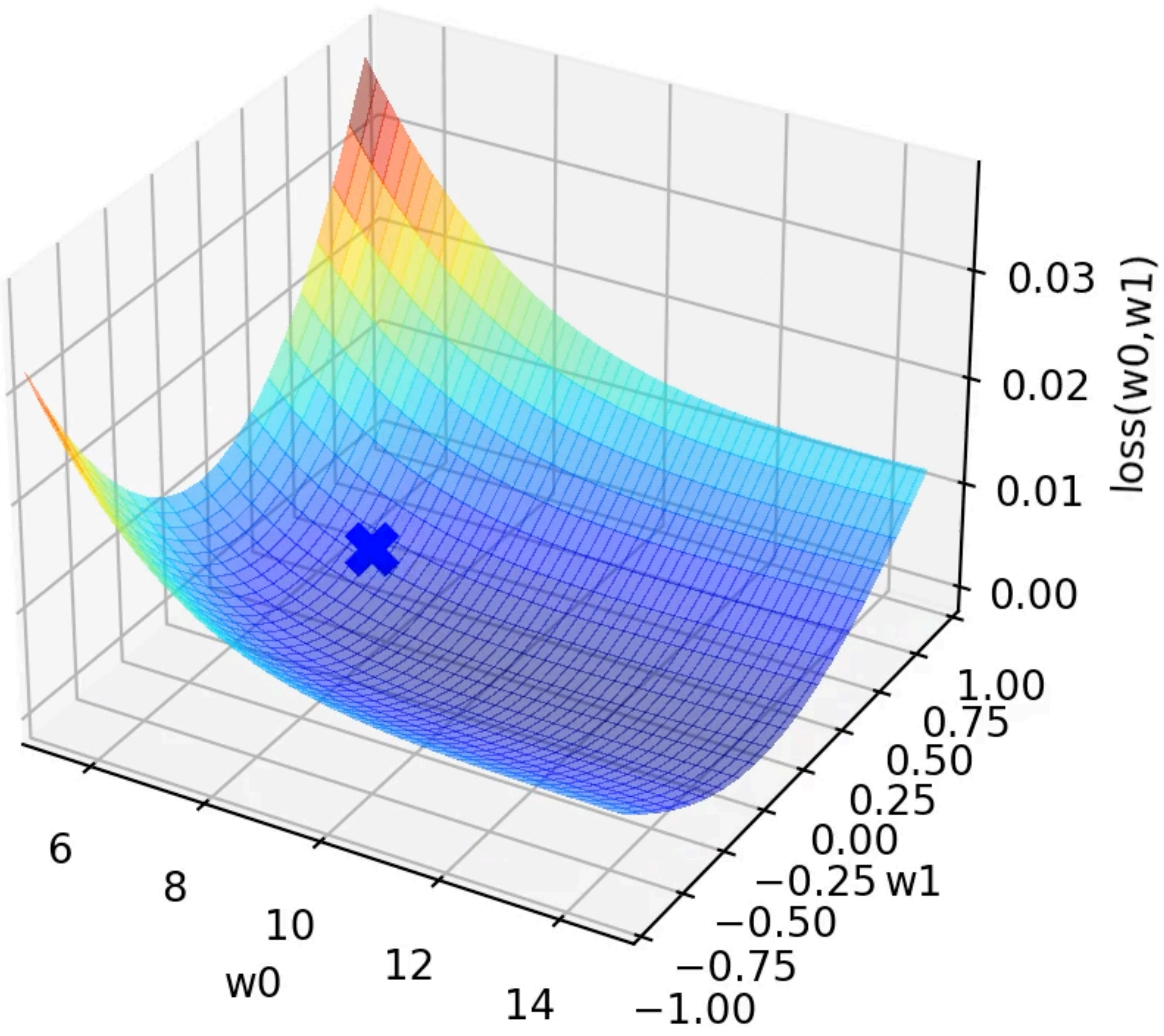
Apofenia=human overfitting



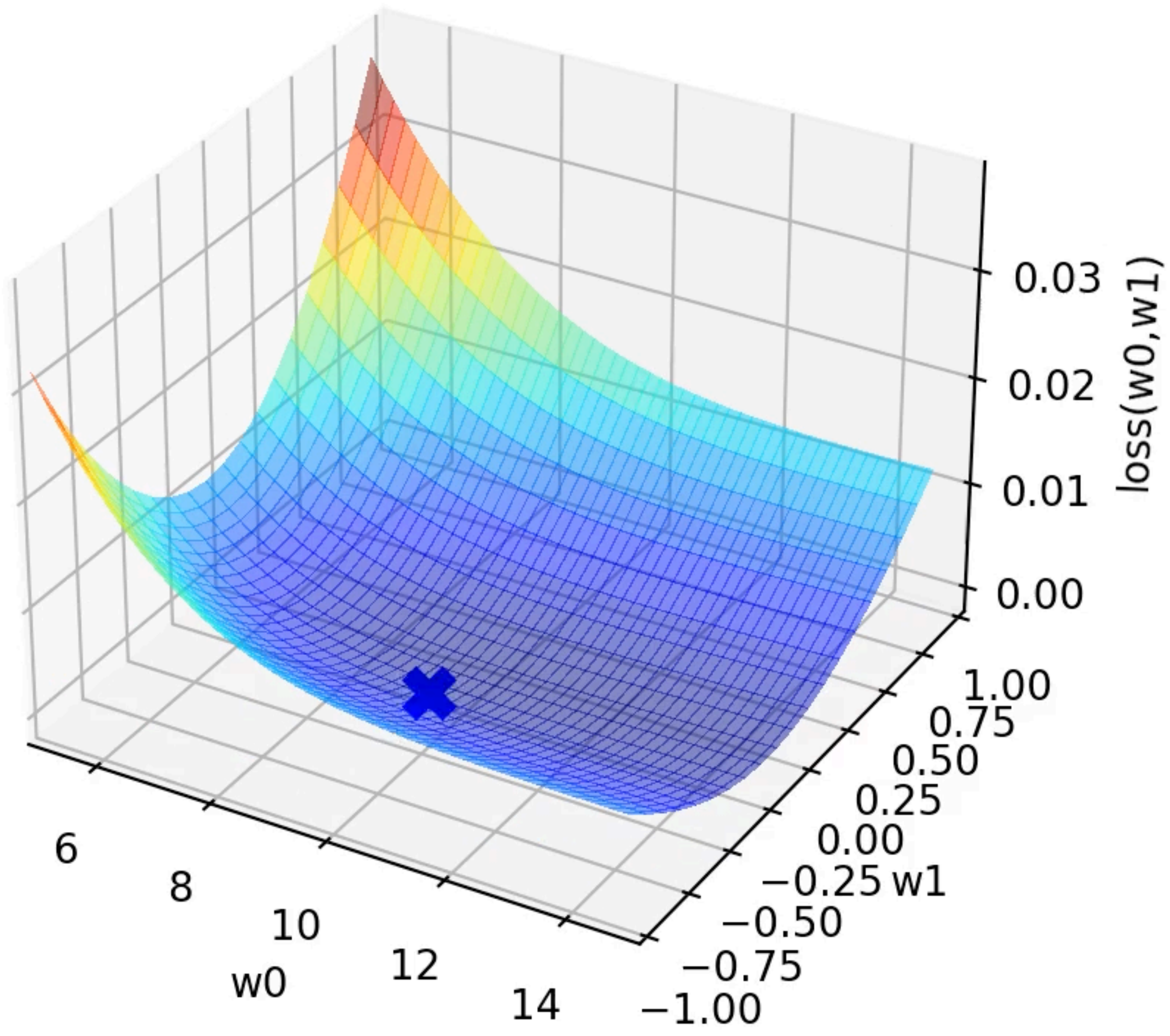
What can go wrong: **learning fails to find good model parameters**

due to hyper-parameters, local optima, bad initialization …

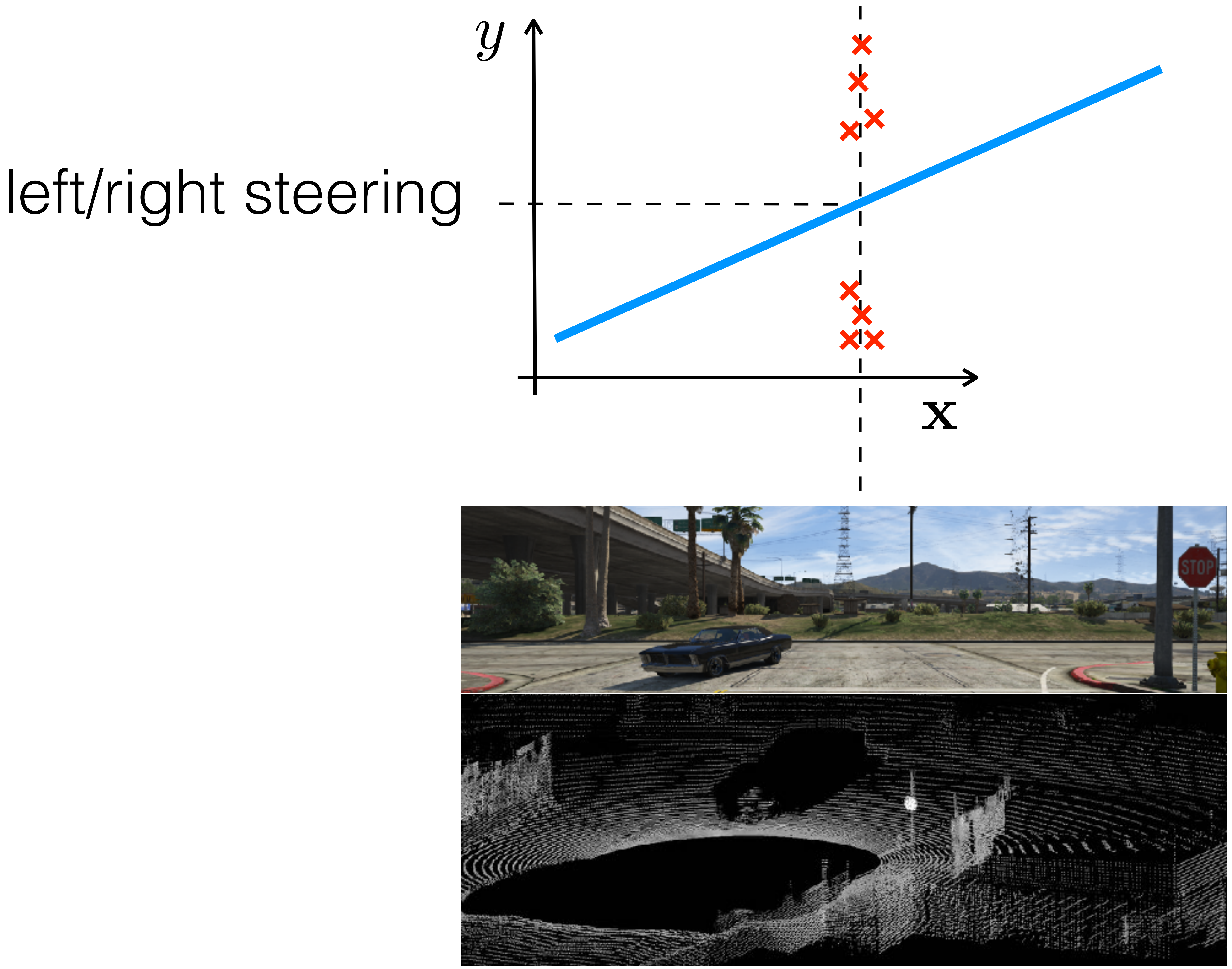
reasonable learning rate

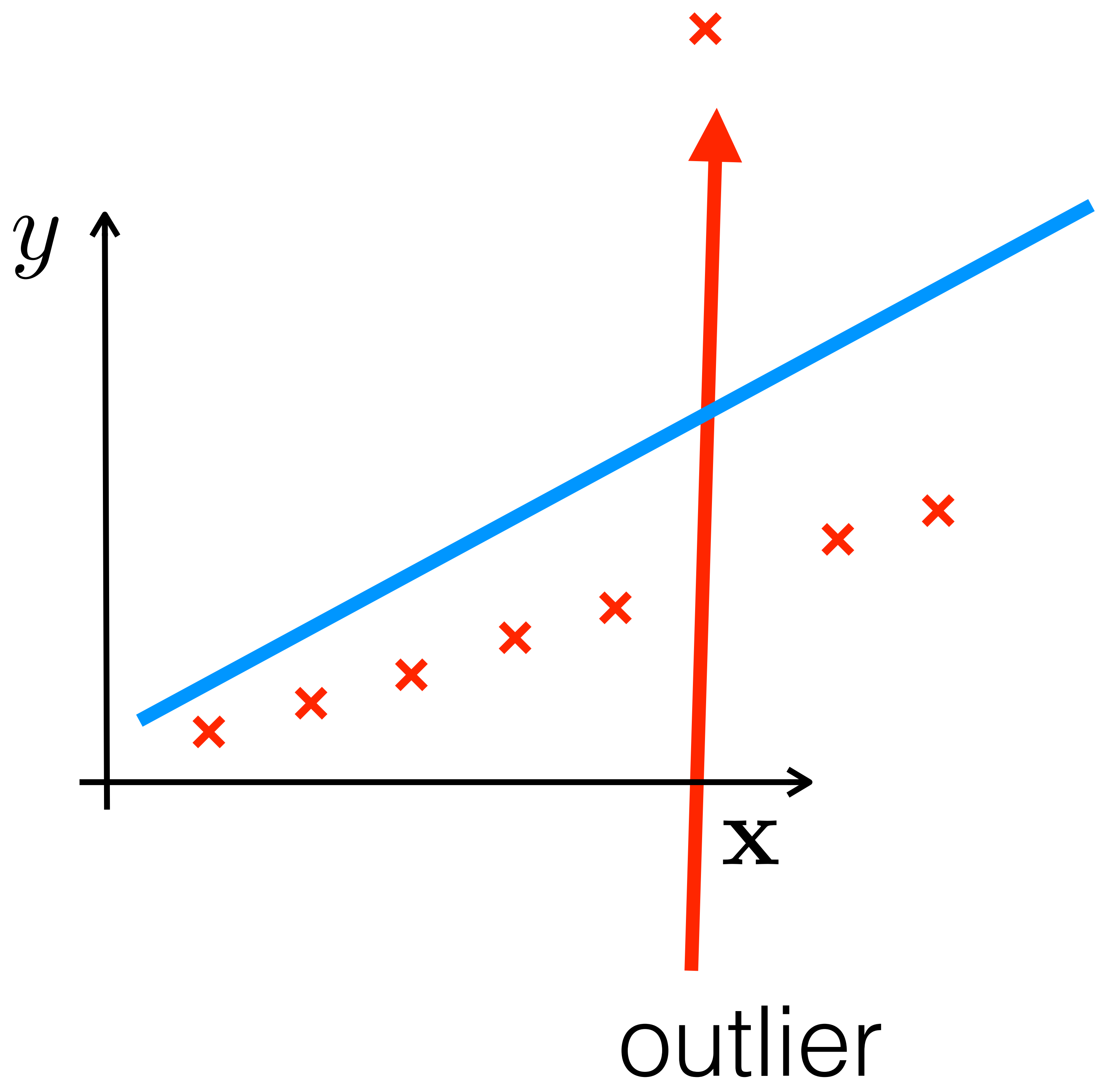


too big learning rate

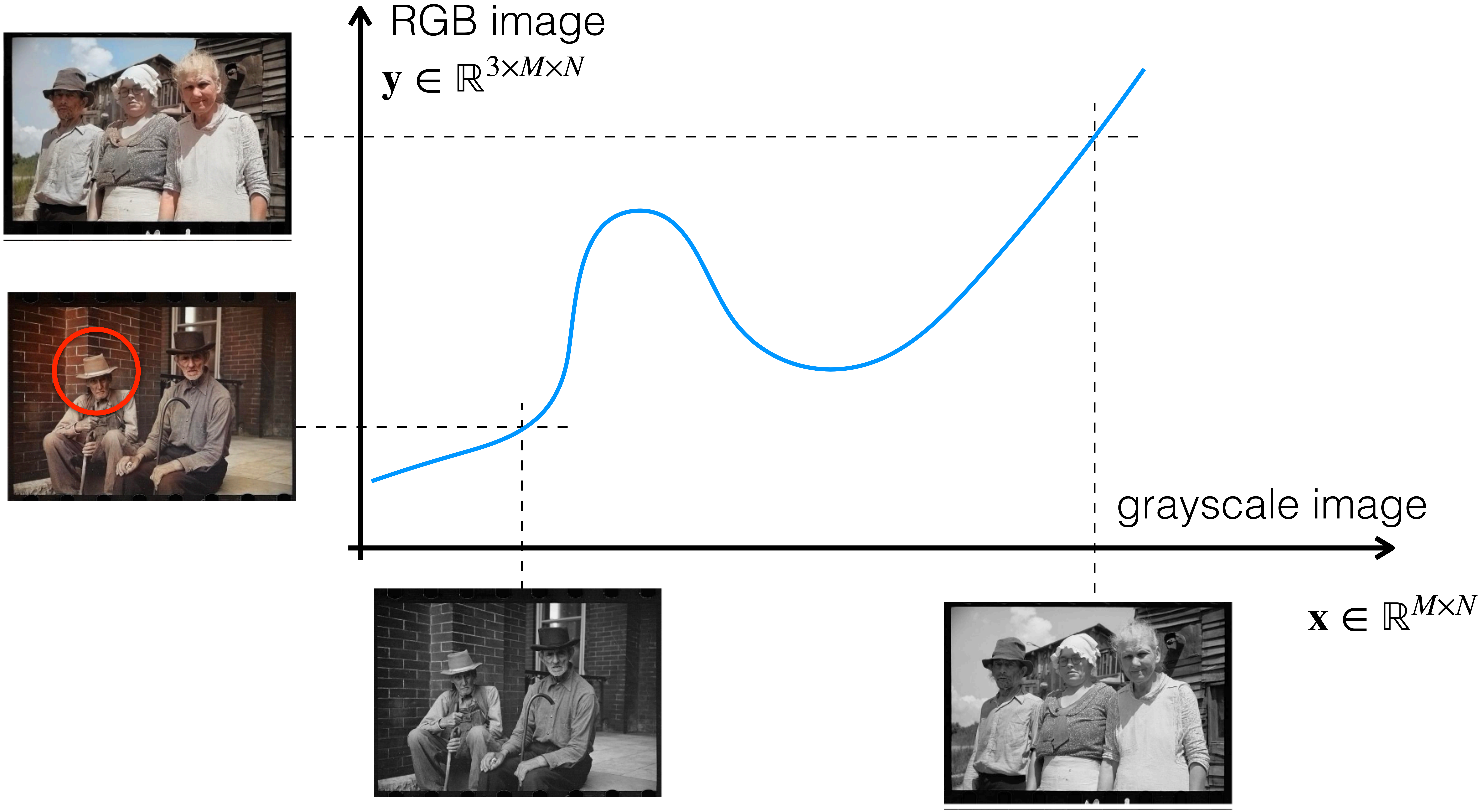


What can go wrong: **inappropriate choice of loss function**

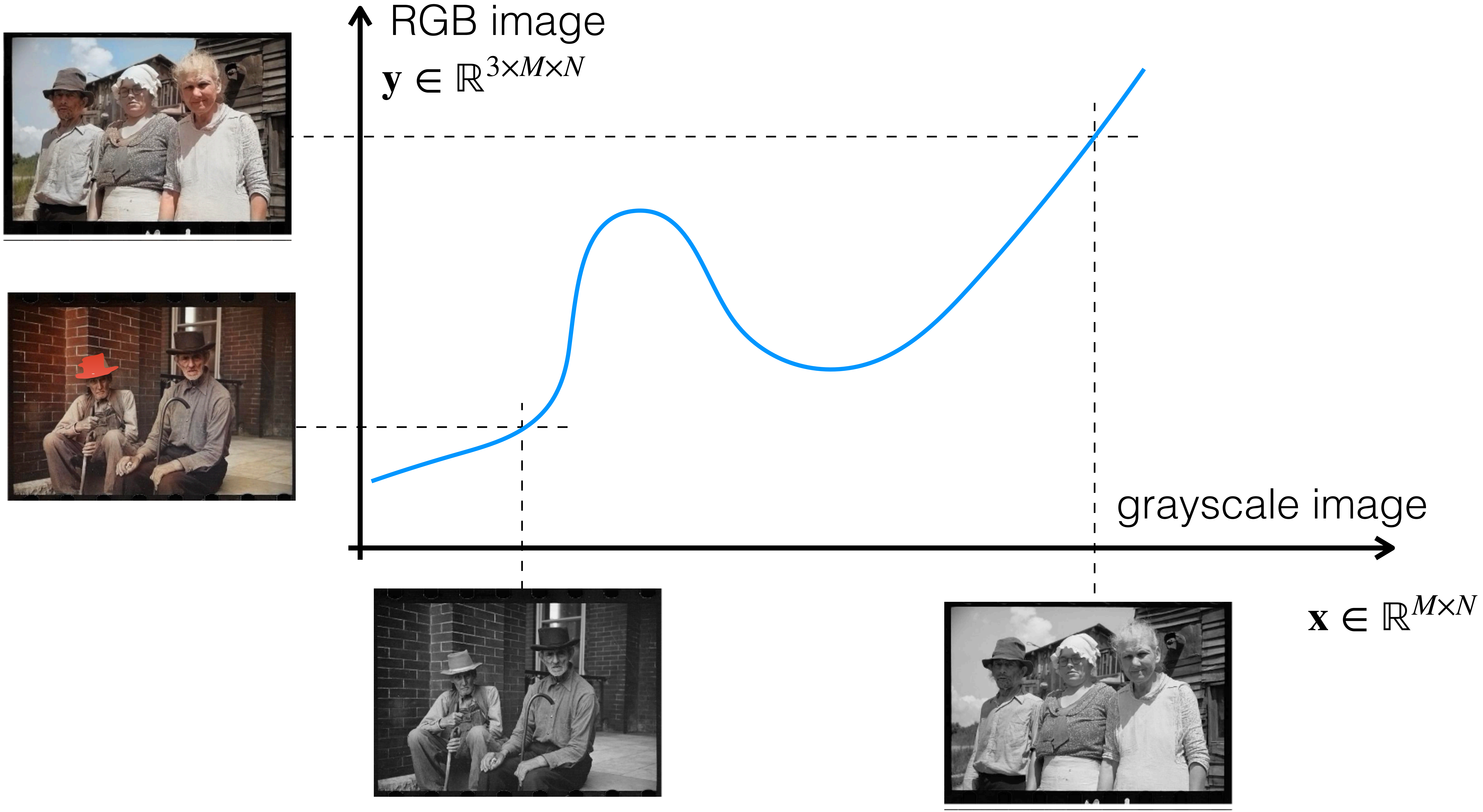




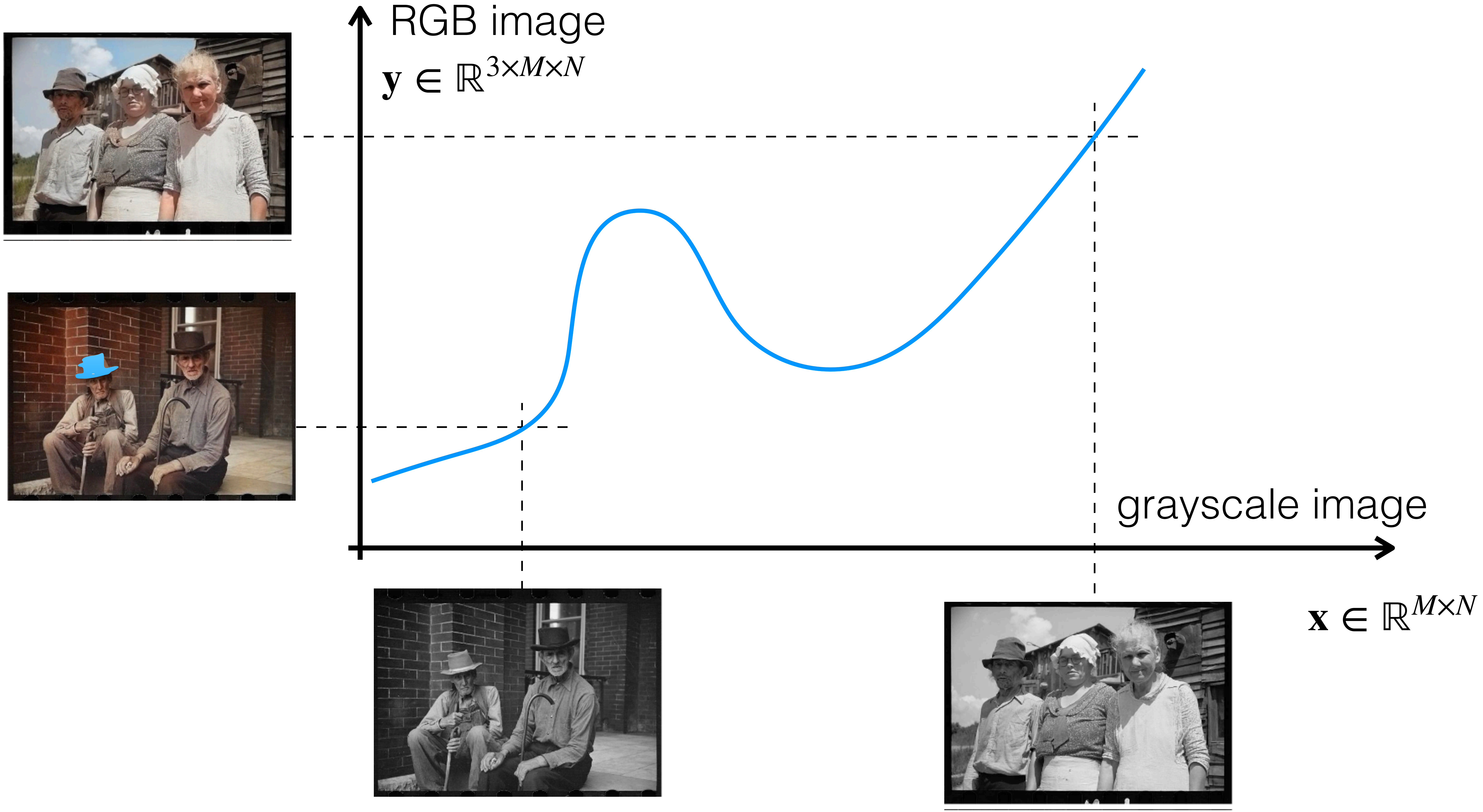
What can go wrong: **inappropriate choice of loss function**



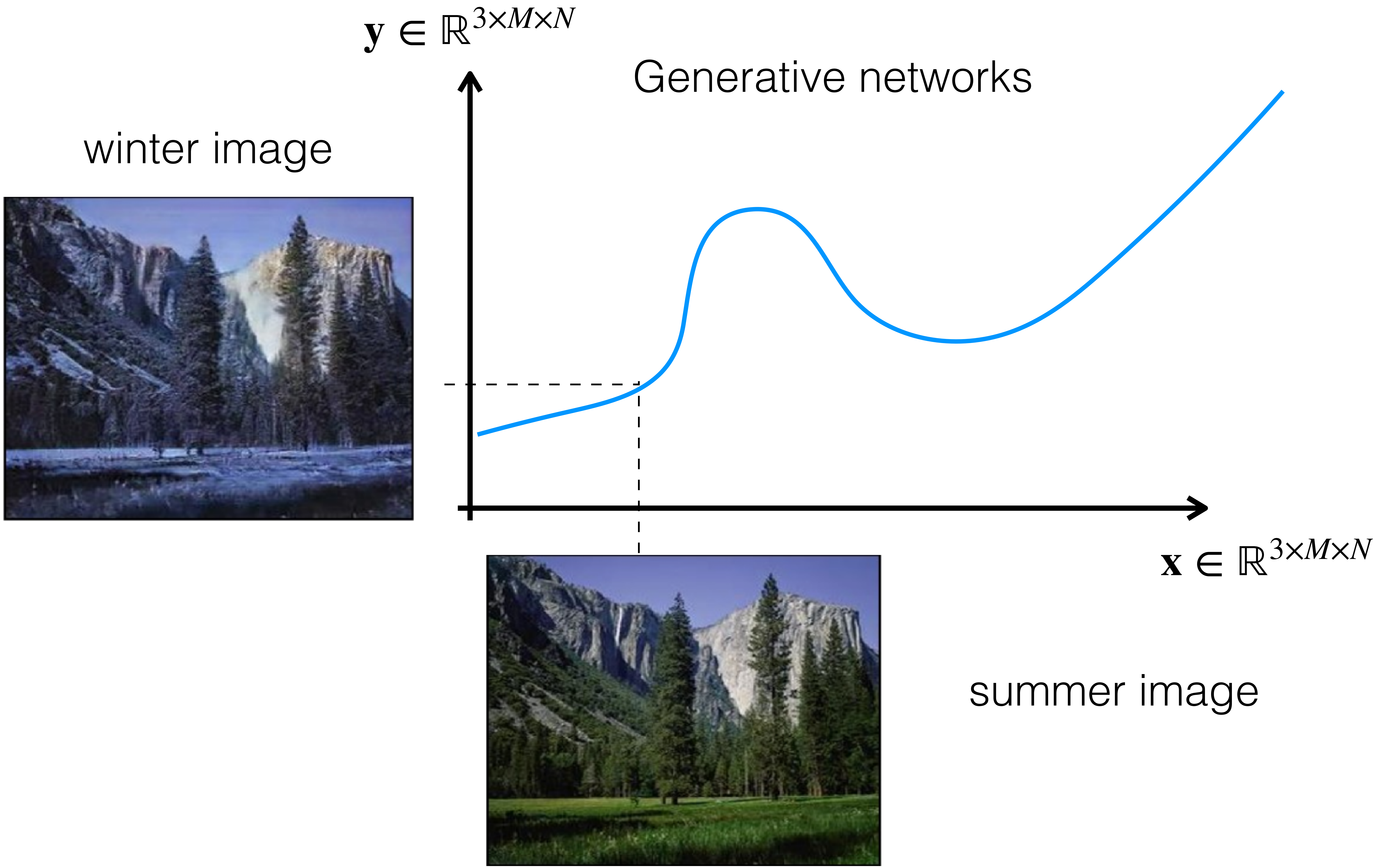
What can go wrong: **inappropriate choice of loss function**



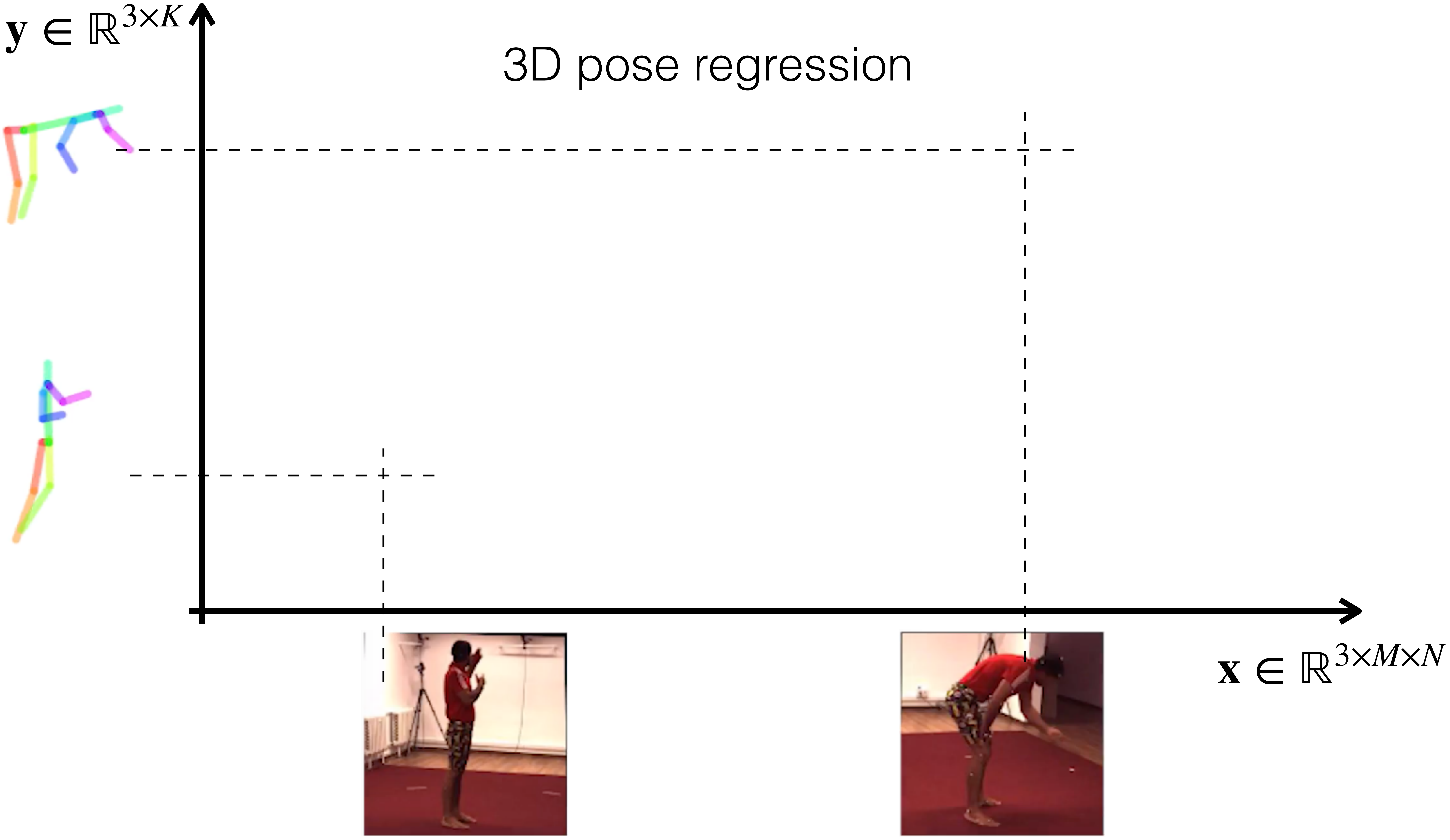
What can go wrong: **inappropriate choice of loss function**



What can go wrong: **inappropriate choice of loss function**



What can go wrong: **inappropriate choice of loss function**

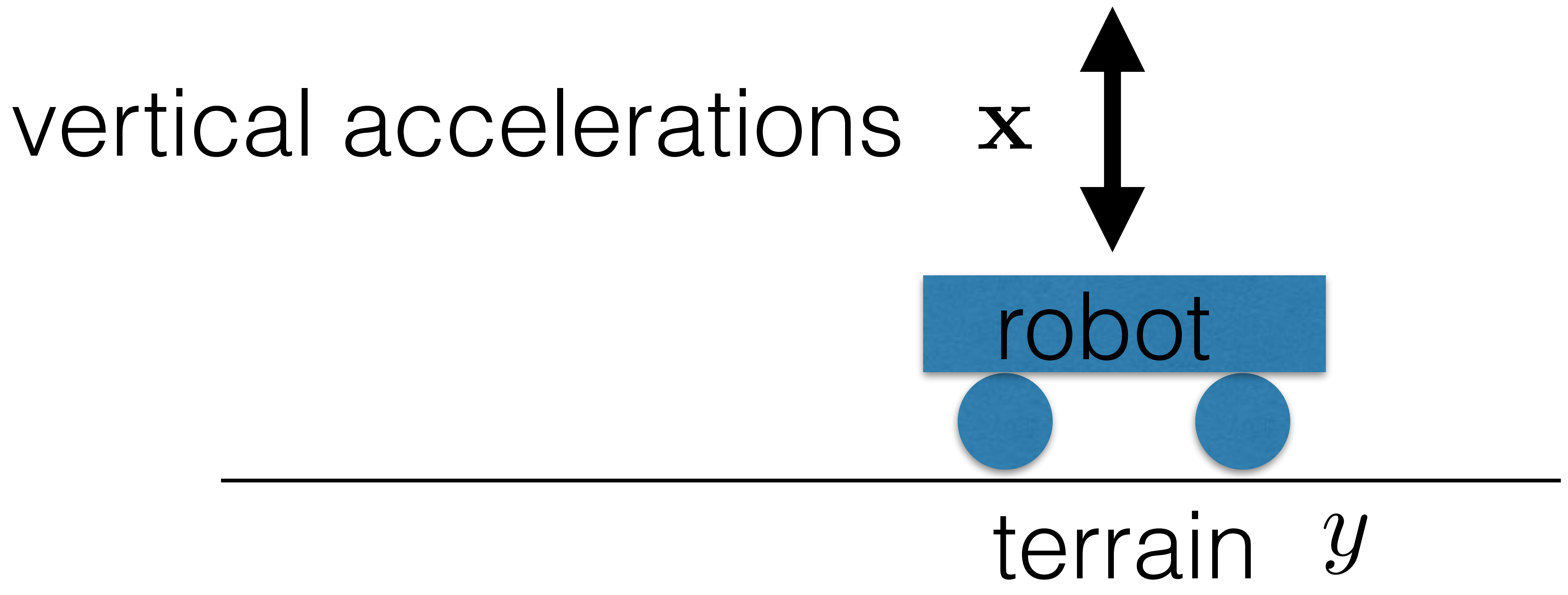


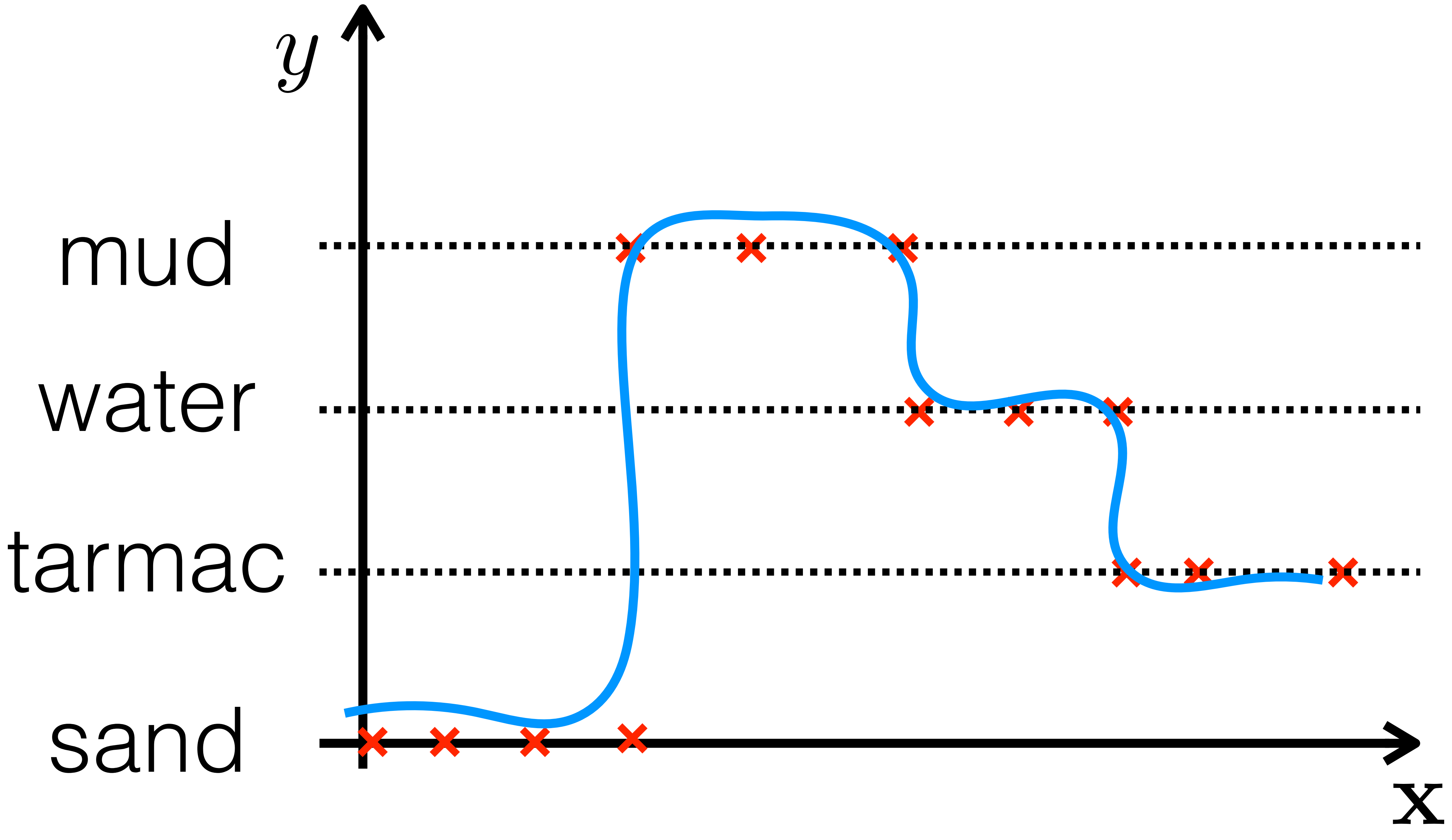
What can go wrong: **inappropriate choice of loss function**
[Heess 2017] https://arxiv.org/abs/1707.02286

What can go wrong: **inappropriate choice of loss function**

- Can I treat problem as regression?
- Suffers from:
  - bad optimization,
  - enforced ordering (loss for misclasifying mud-to-tarmac << mud-to-sand)
  - cannot model: “mud or sand but definitely not tarmac”.

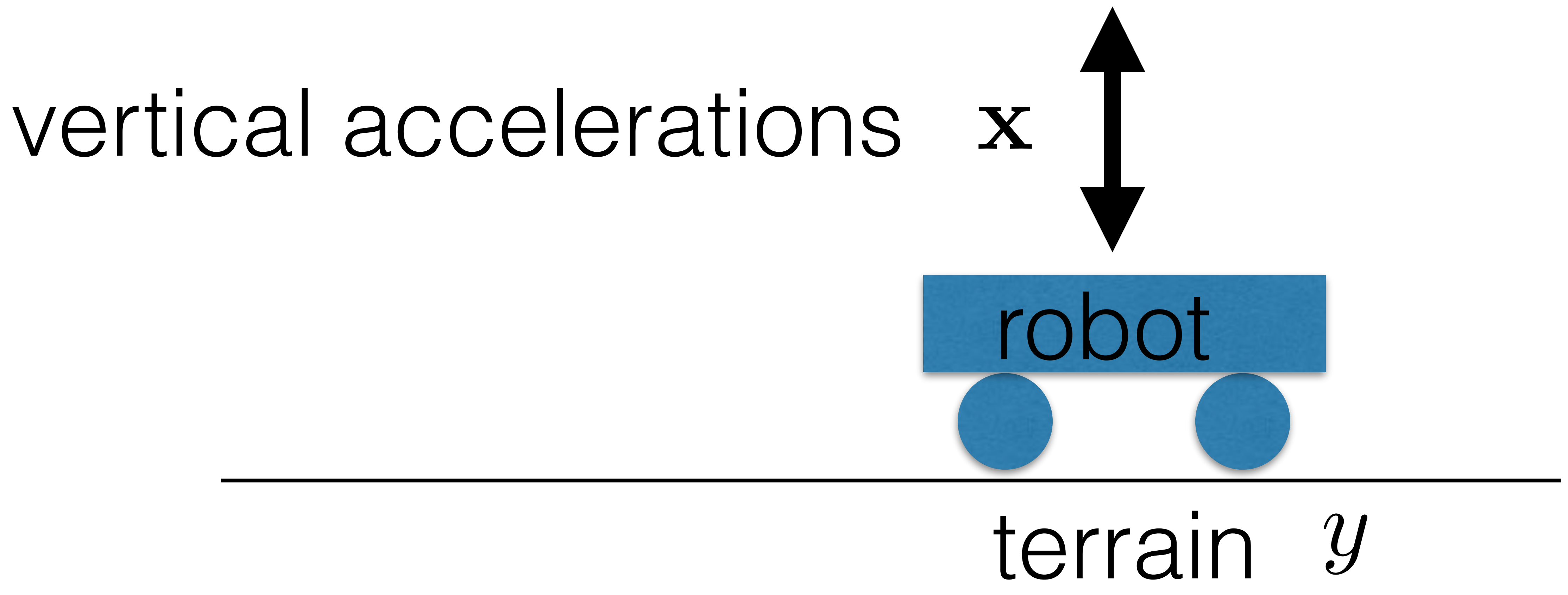
Motivation example: classification

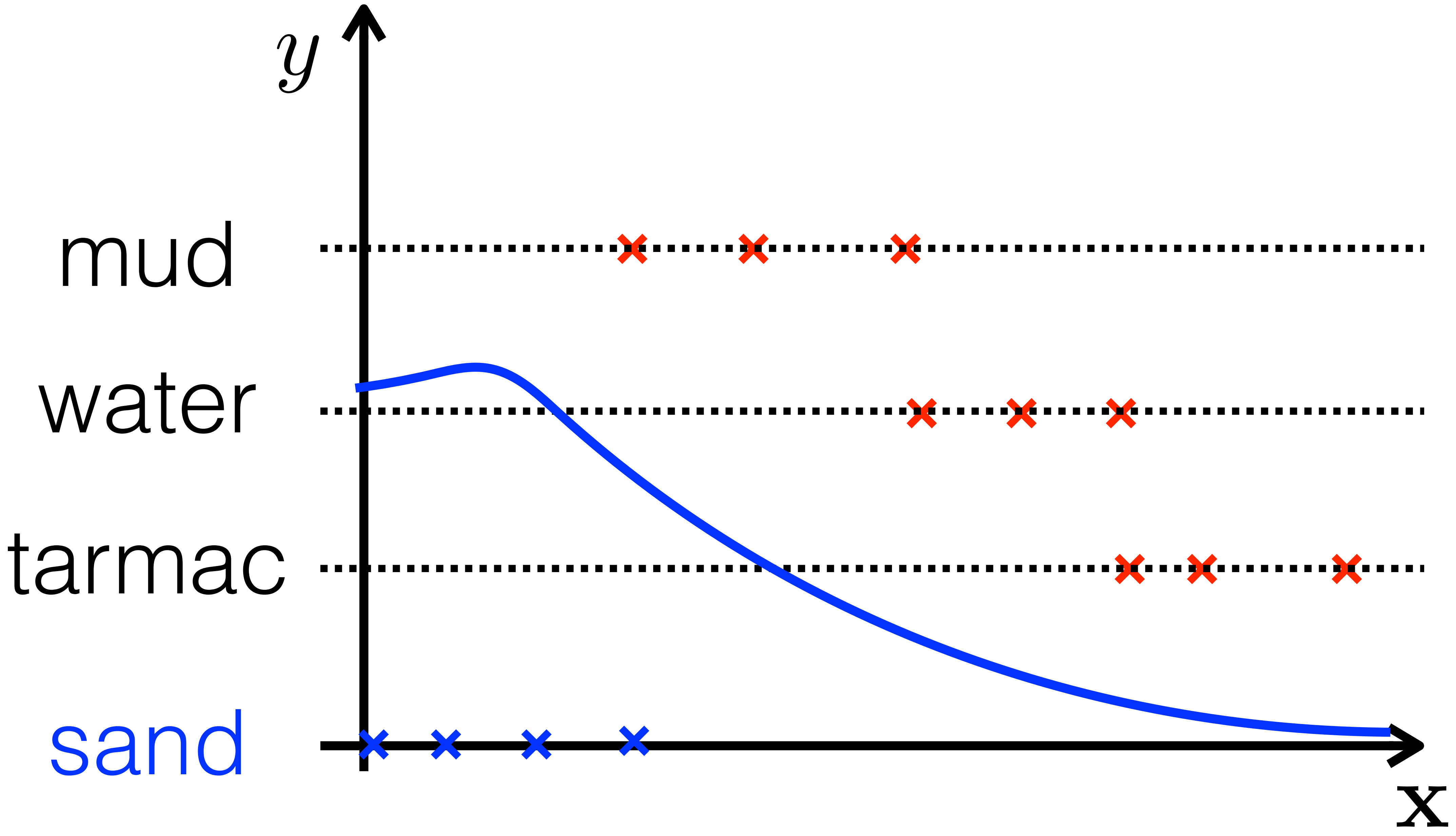




trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \dots \mathbf{x}_N, y_N\}$

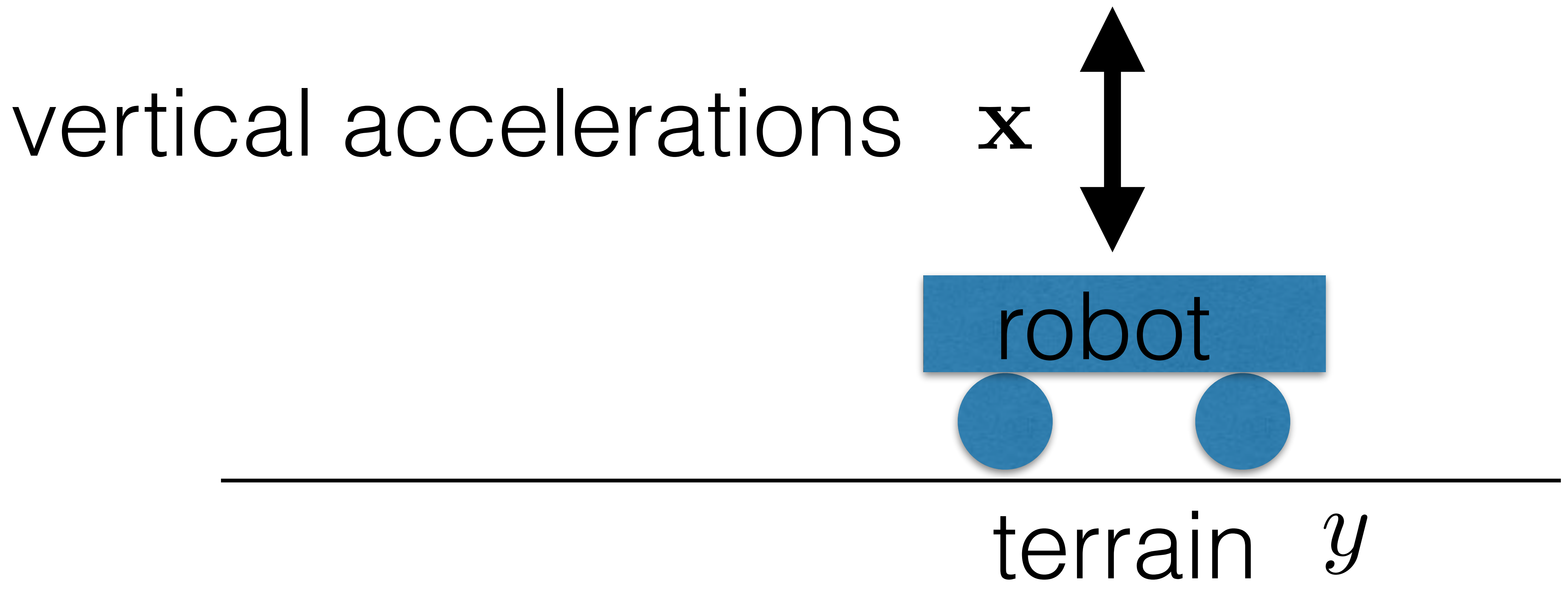
Motivation example: classification

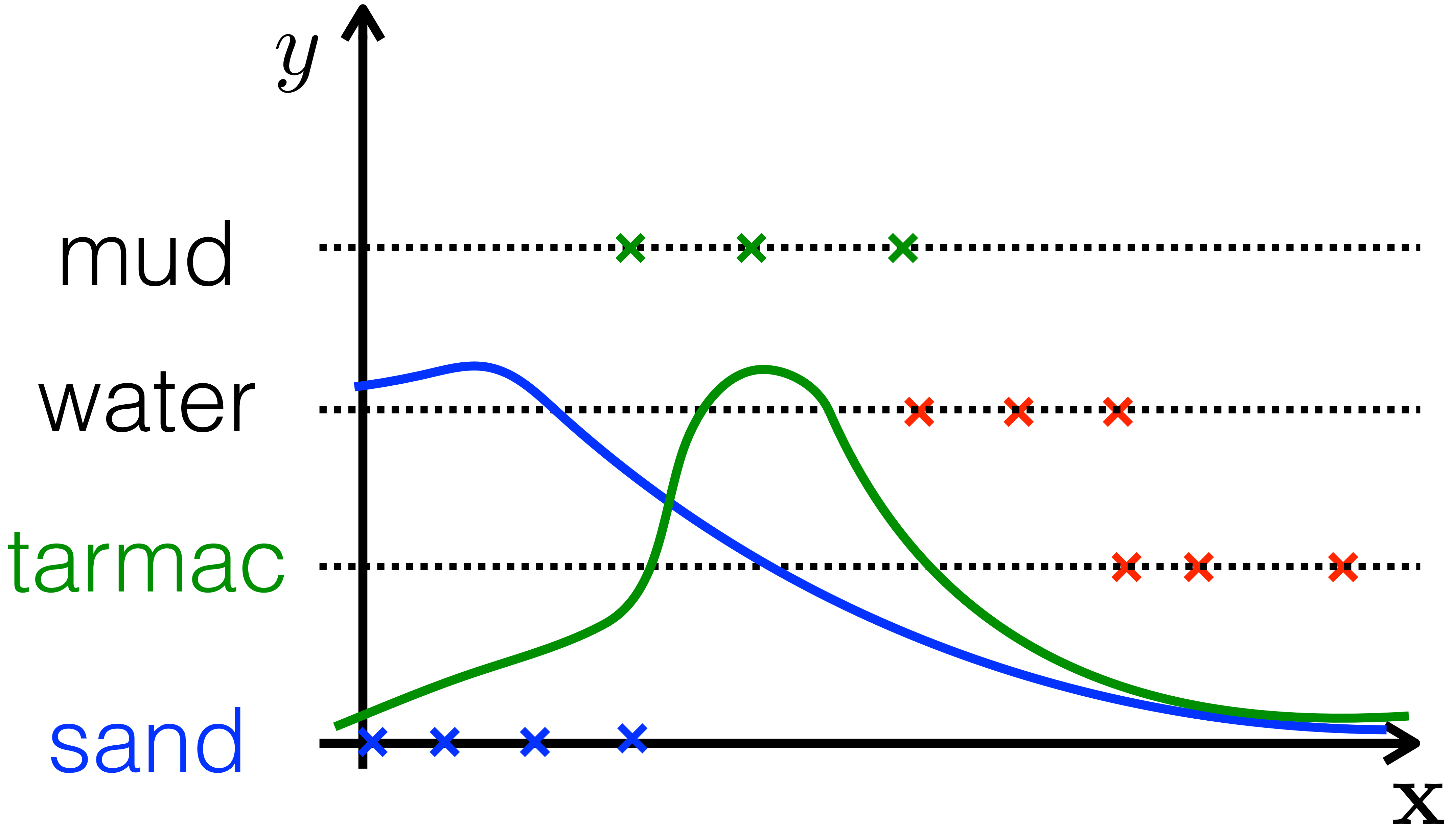




trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

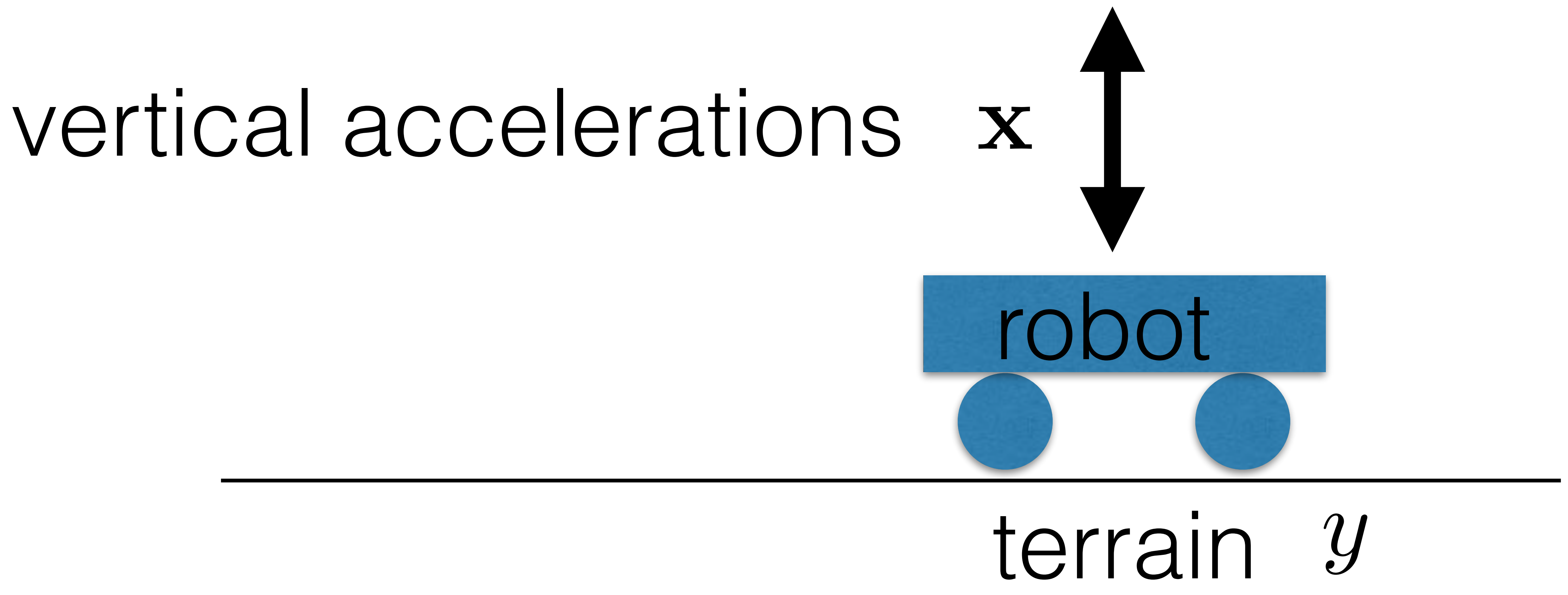


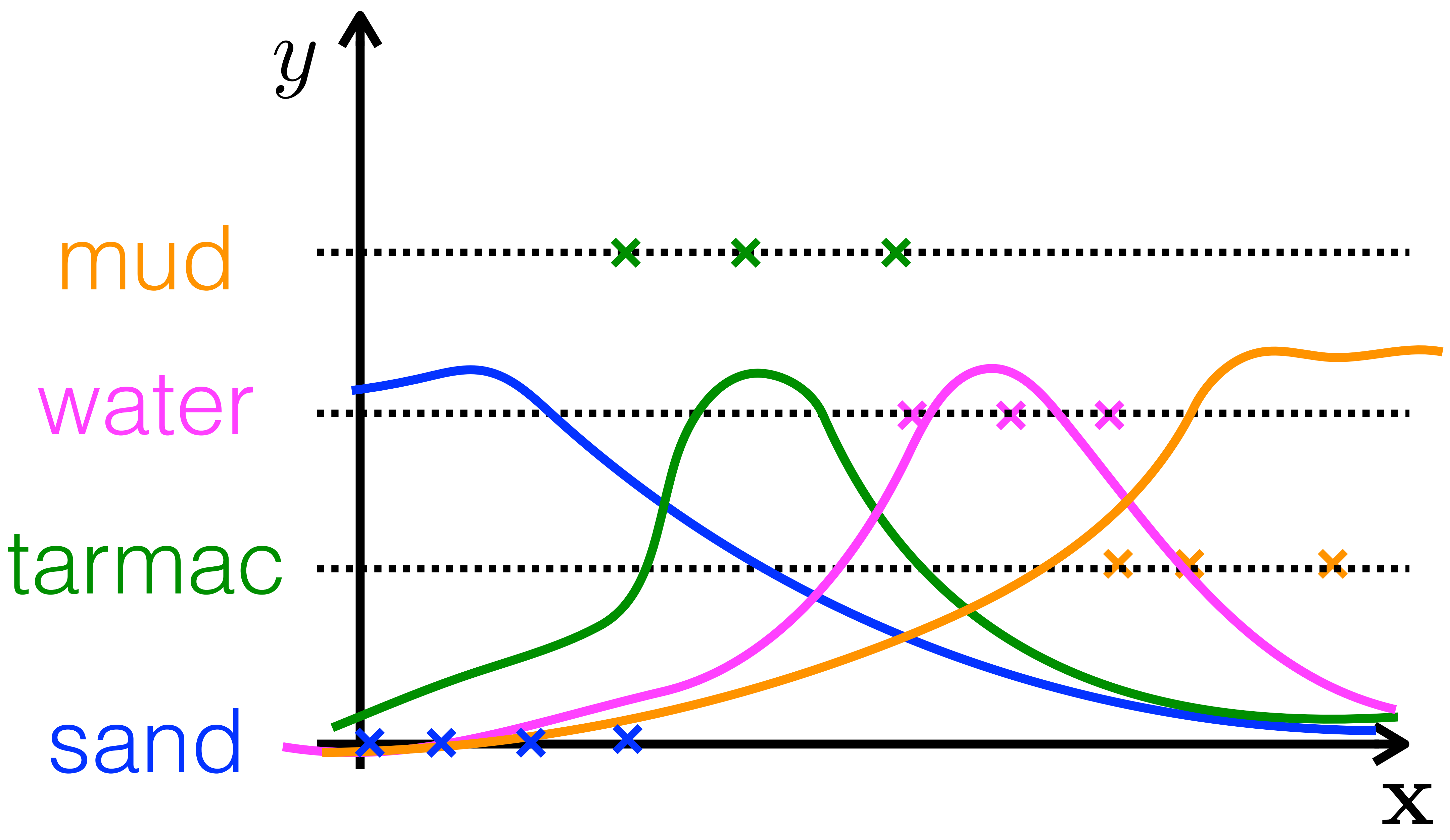


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

- 4 functions predicting class probabilities $p(y_1 | x, w_1),\ p(y_2 | x, w_2),\ \ldots$

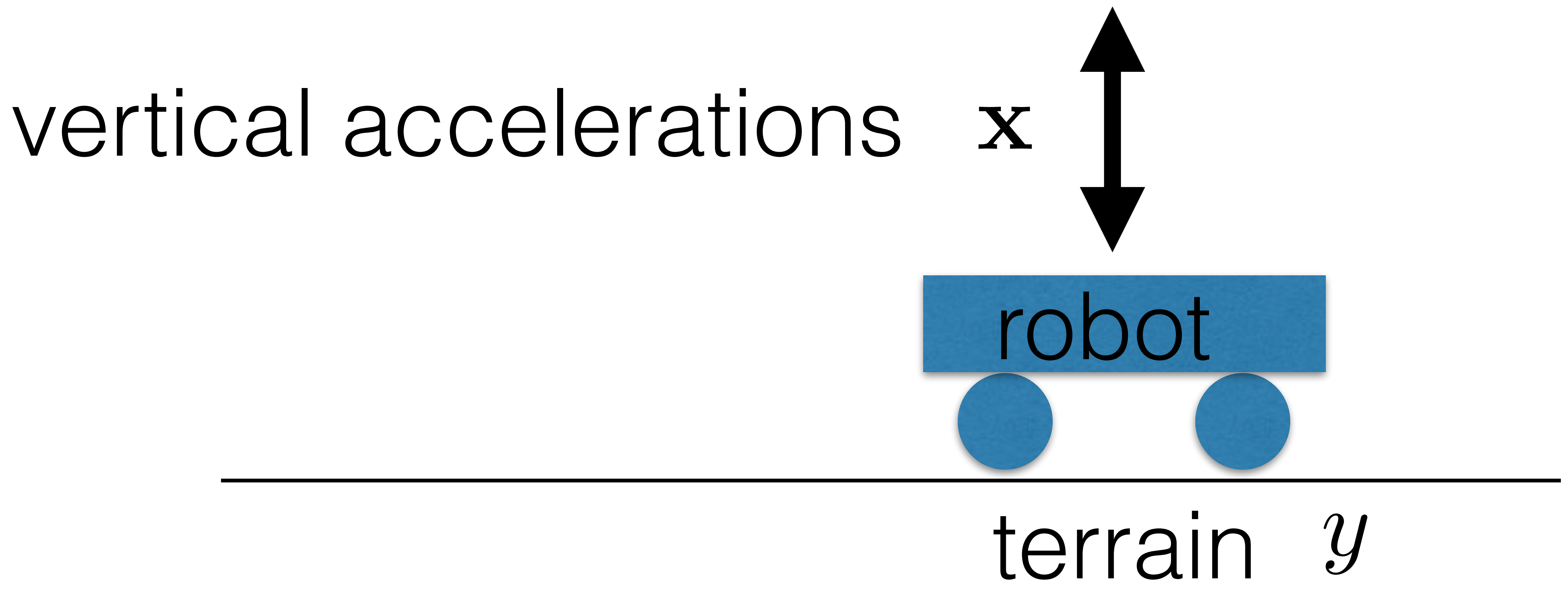


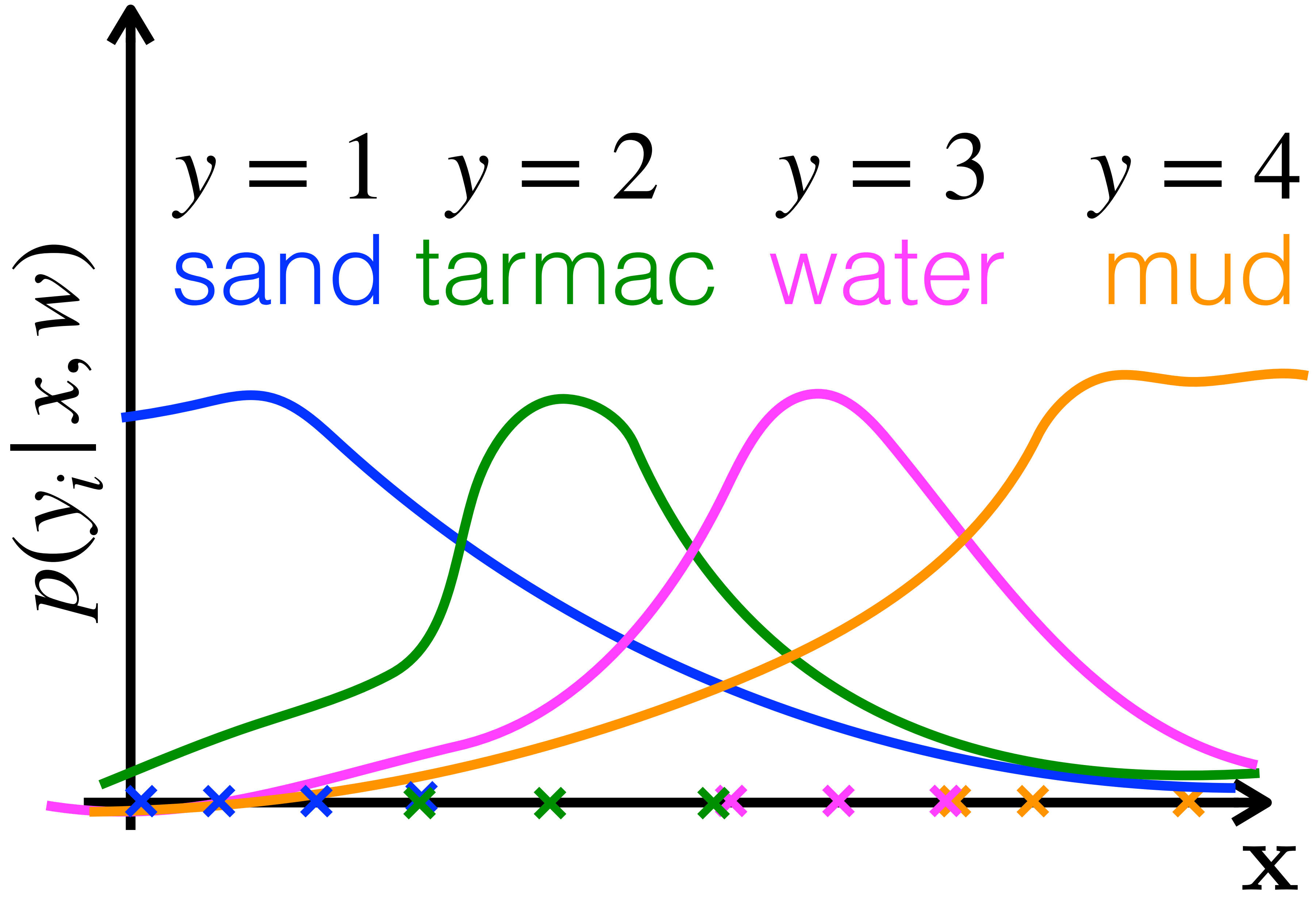


trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

- 4 functions predicting class probabilities $p(y_1 | x, w_1),\ p(y_2 | x, w_2),\ \ldots$
- that sum up to one over classes for given x $\sum_i p(y_i | x, w) = 1$
- Loss pull blue function up for blue points, etc… what is a suitable shape?





trn data: $\mathcal{D} = \{\mathbf{x}_1, y_1 \ldots \mathbf{x}_N, y_N\}$

Motivation example: classification

Competencies required for the test T1

- Model (or Architecture/Program) with parameters => learning
- Learning = loss + trn data + optimization procedure
- Evaluation = measuring performance (not necessary loss) on tst data
- What could go wrong?
  - inputs x does not allow to predict y
  - trn/tst data distribution mismatch
  - model does not generalize well
  - learning fails to find good parameters
  - inappropriate choice of loss function
- Regression vs Classification
- **Next lecture:** Linear classification of RGB images