Artificial Intelligence II

Part 2: Lecture 9

Yalda Mohsenzadeh

Slides are adapted from Antonio Torralba and Philip Isola (MIT)

Datasets, bias, and adaptation

Garbage in, Garbage out

- A machine learning algorithm will do whatever the training data tells it to do.
- If the data is bad or biased, the learned algorithm will be too.

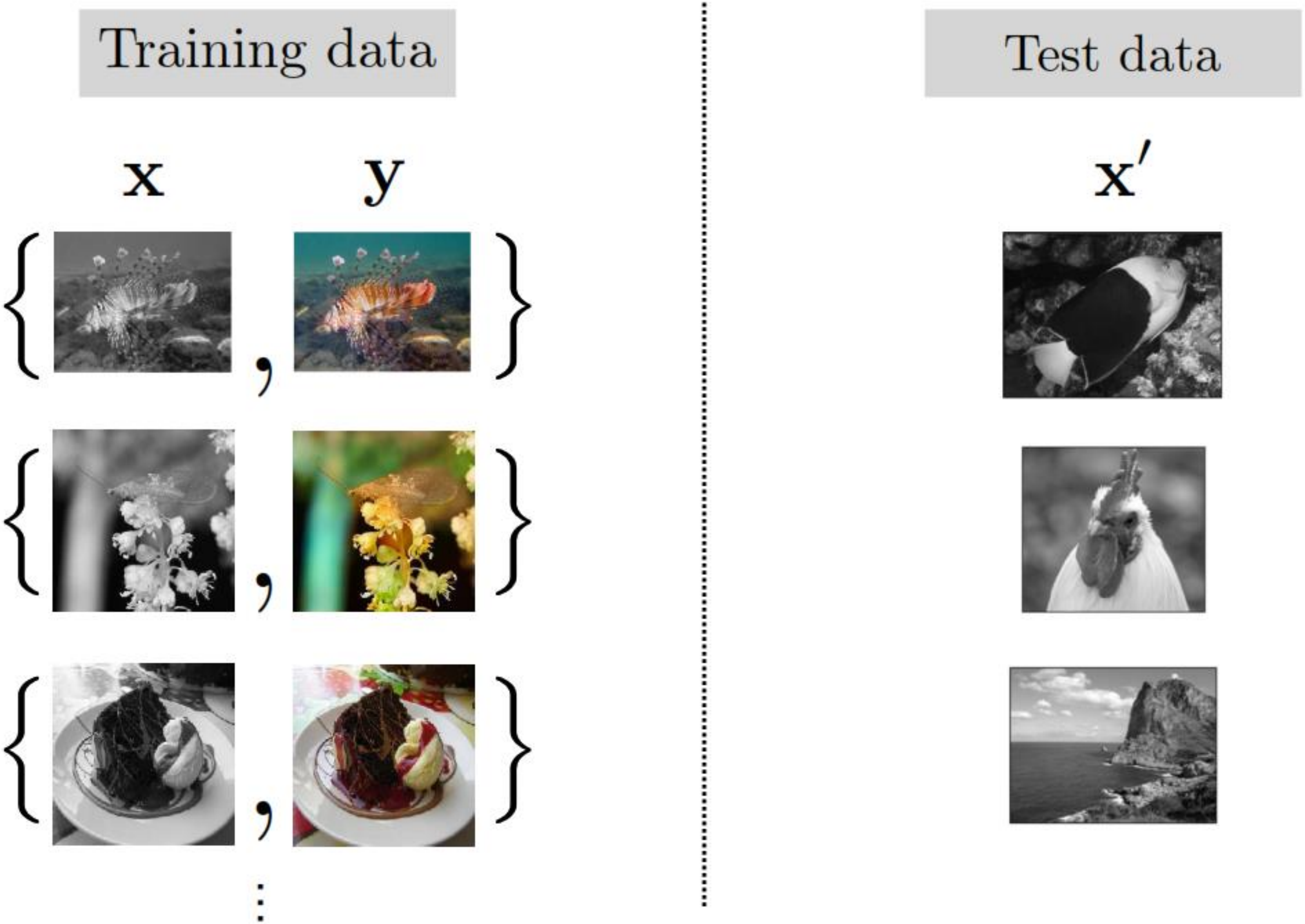
Microsoft's Tay Chatbot

- Chatbot released on twitter
- Learned from interactions with users
- Started mimicking offensive language, was shut down.

["Colorful image colorization", Zhang et al., ECCV 2016]

["Colorful image colorization", Zhang et al., ECCV 2016]

["Colorful image colorization", Zhang et al., ECCV 2016]



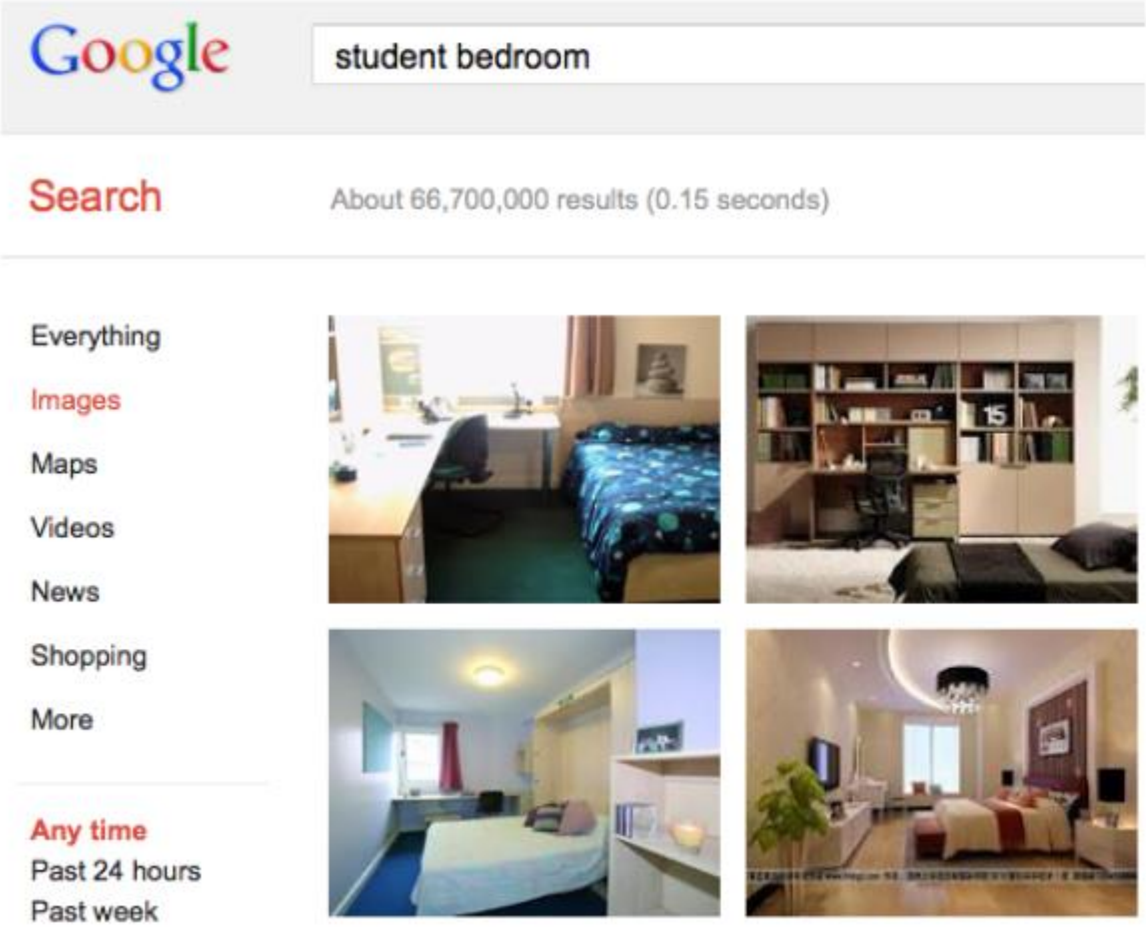
Training data
Test data
$\mathbf{x}$
$\mathbf{y}$
$\mathbf{x}'$

Training data

Test data

Training data

What Google thinks are
student bedrooms



Training data

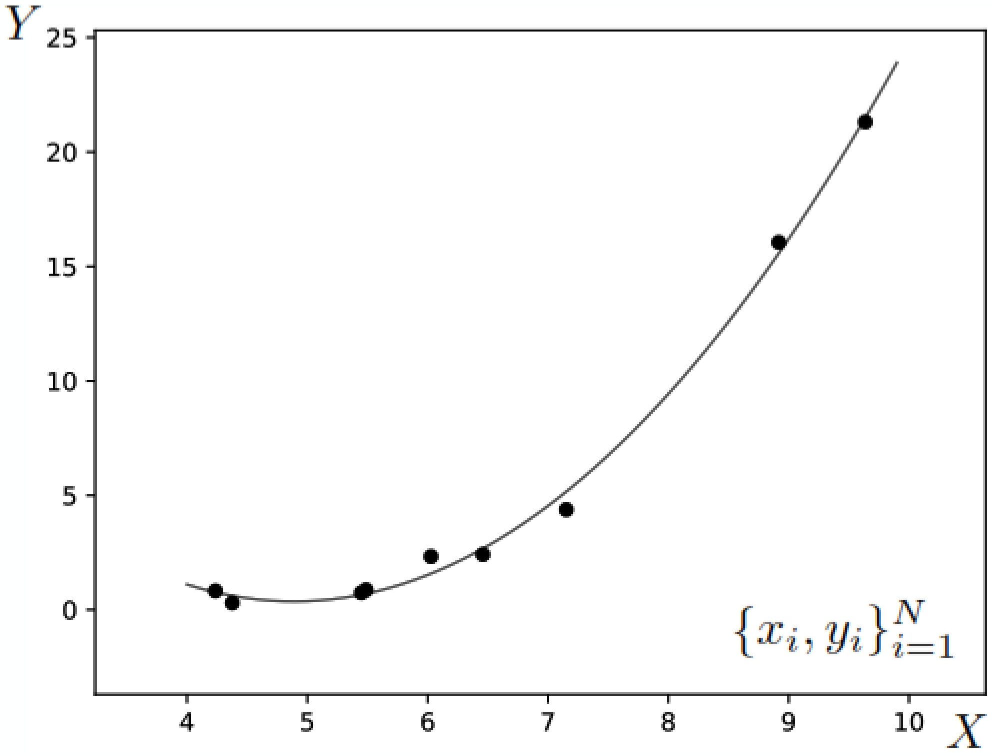
Driving simulator (GTA)

Test data

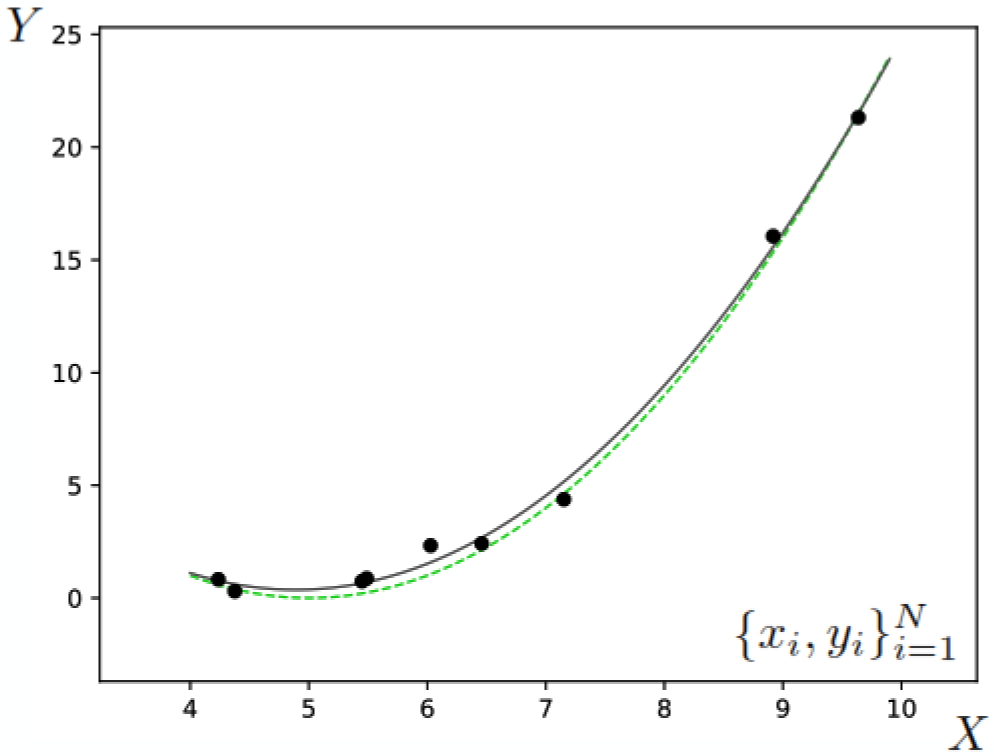
Driving in the real world

Let’s revisit the problem of generalization

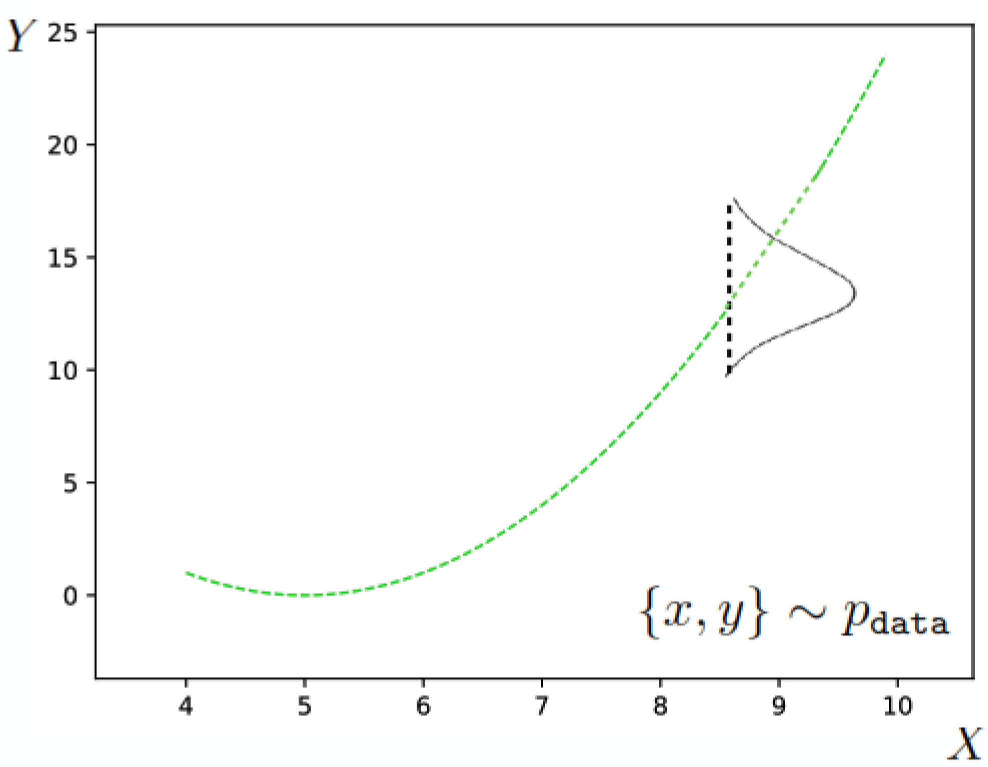
Training data



Training data



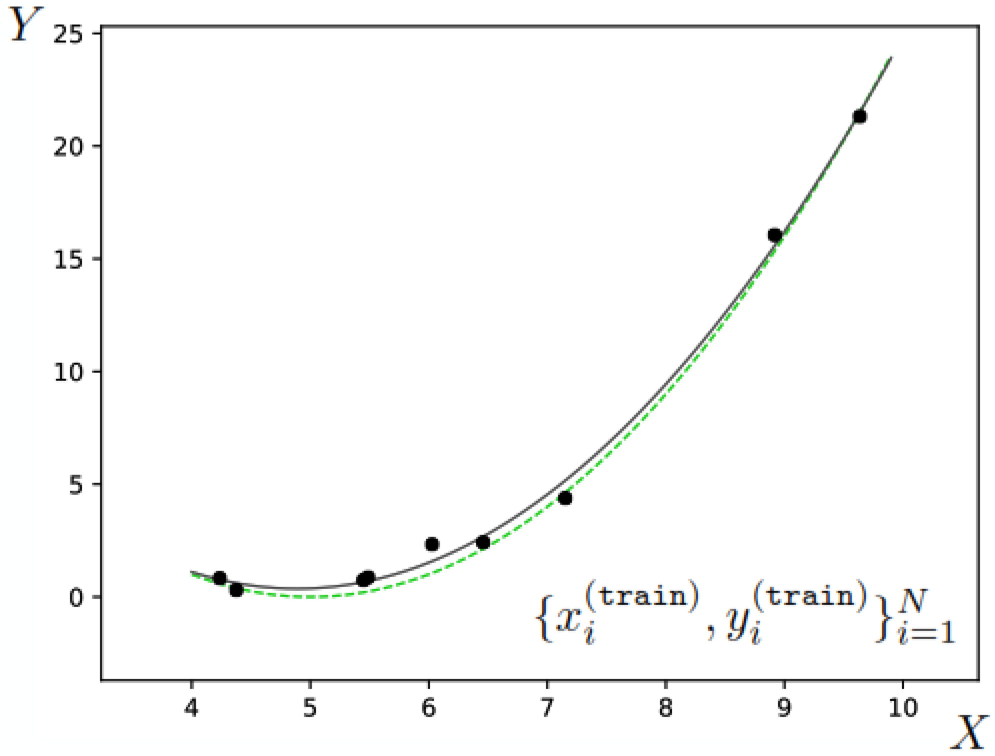
Test data



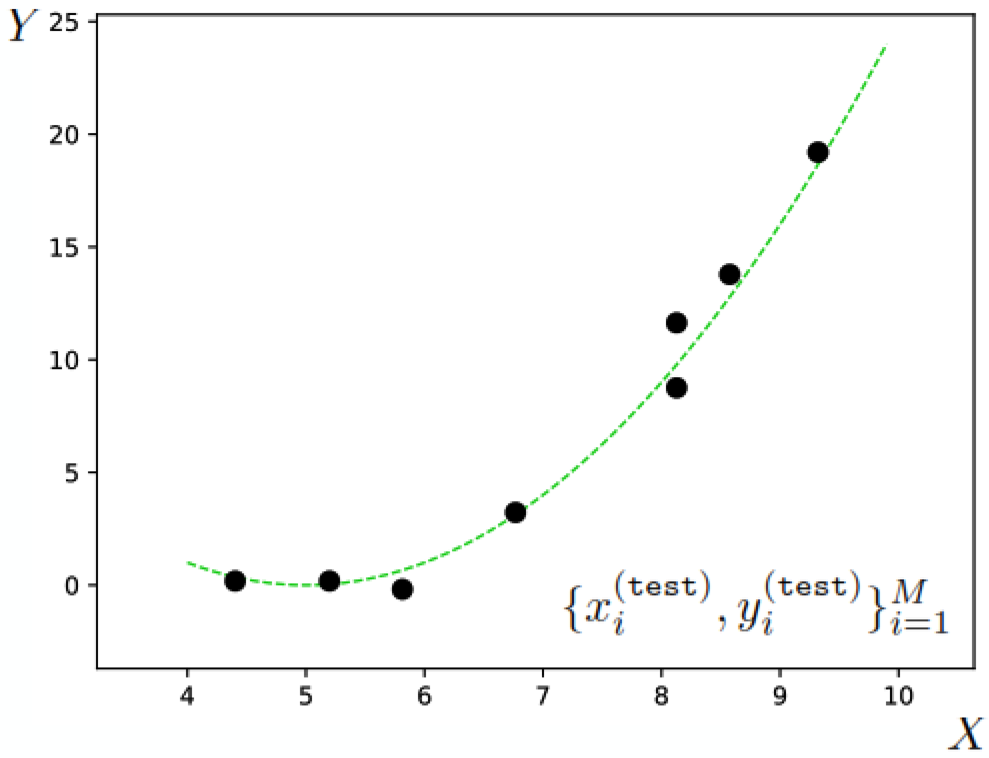
True **data-generating process**

$$p_{\texttt{data}}$$

Training data



Test data



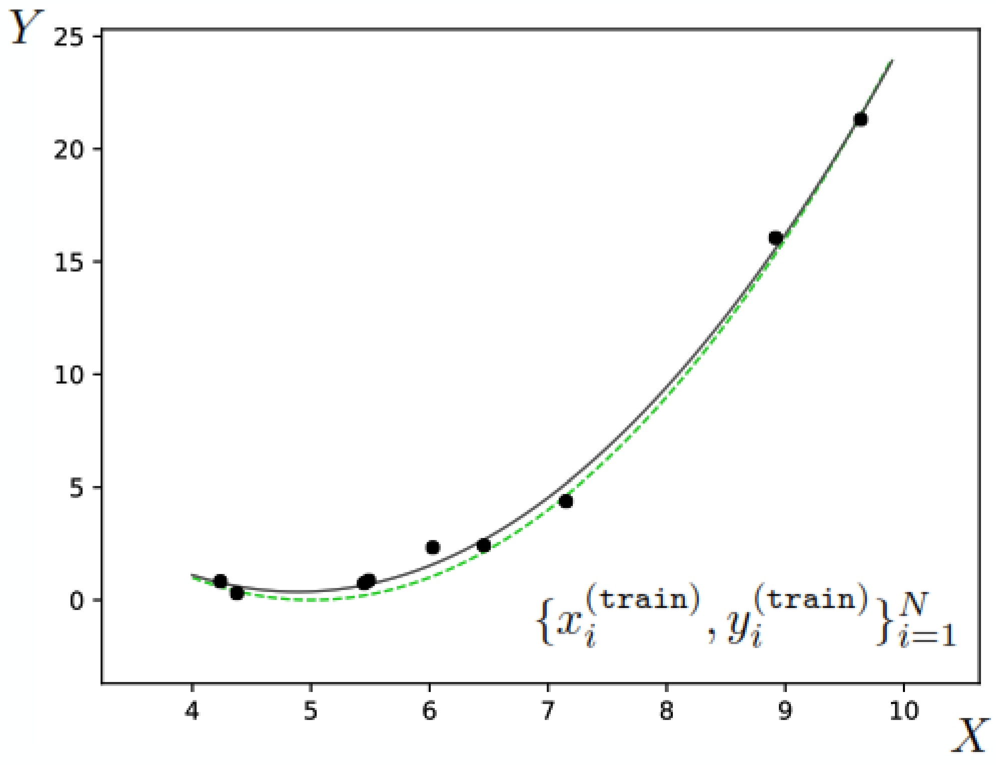
True **data-generating process**

$$p_{\mathtt{data}}$$

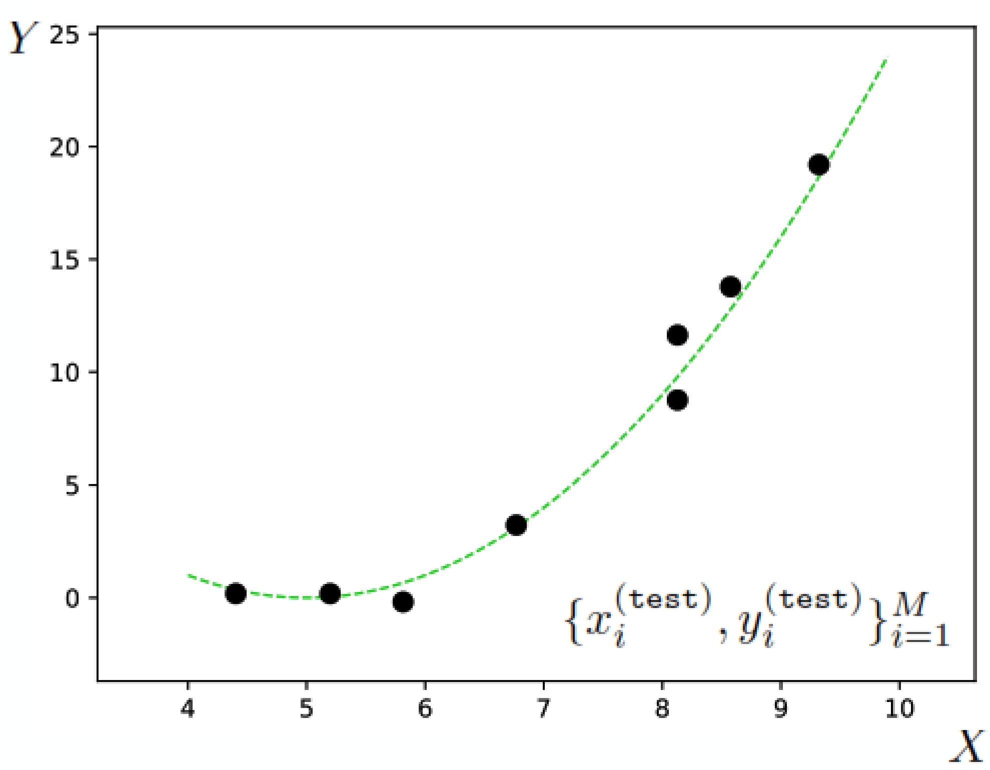
$$\{x_i^{(\mathtt{train})}, y_i^{(\mathtt{train})}\} \overset{\mathrm{iid}}{\sim} p_{\mathtt{data}}$$

$$\{x_i^{(\mathtt{test})}, y_i^{(\mathtt{test})}\} \overset{\mathrm{iid}}{\sim} p_{\mathtt{data}}$$

Training data



Test data

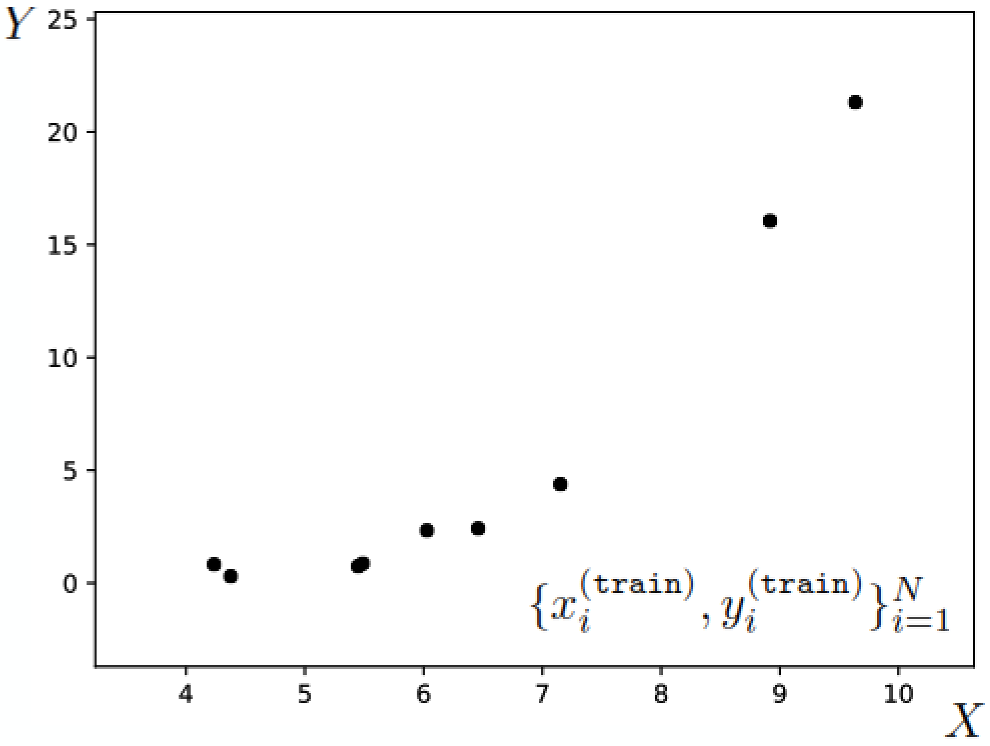


This is a huge assumption!
Almost never true in practice!

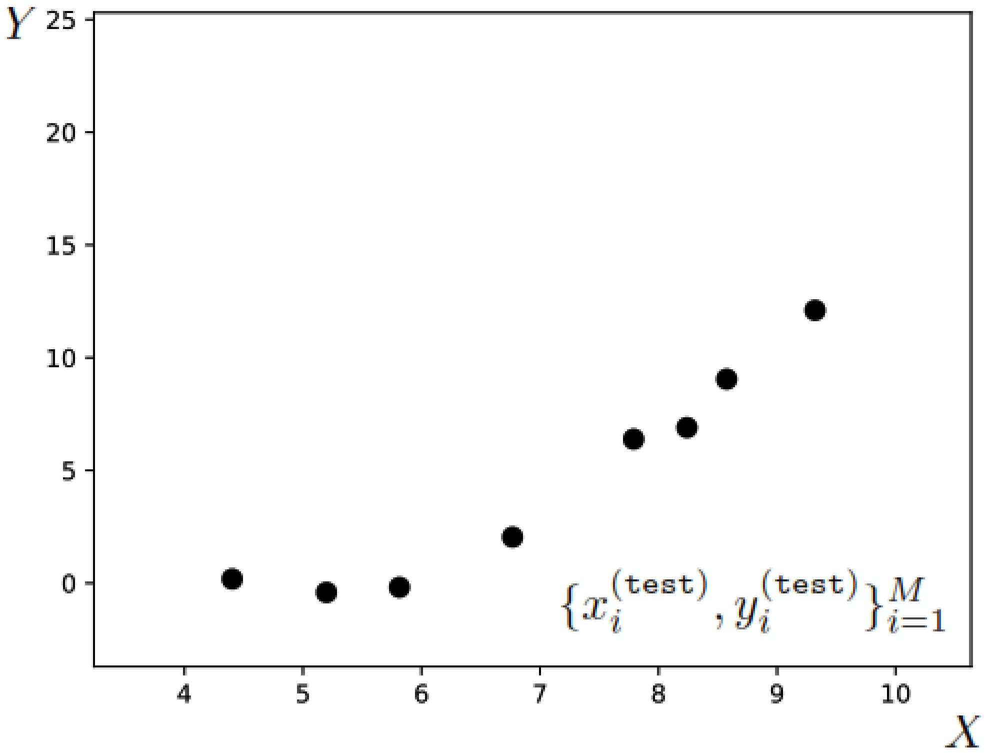
$$\{x_i^{(\texttt{train})}, y_i^{(\texttt{train})}\} \overset{\texttt{iid}}{\sim} p_{\texttt{data}}$$

$$\{x_i^{(\texttt{test})}, y_i^{(\texttt{test})}\} \overset{\texttt{iid}}{\sim} p_{\texttt{data}}$$

Training data



Test data

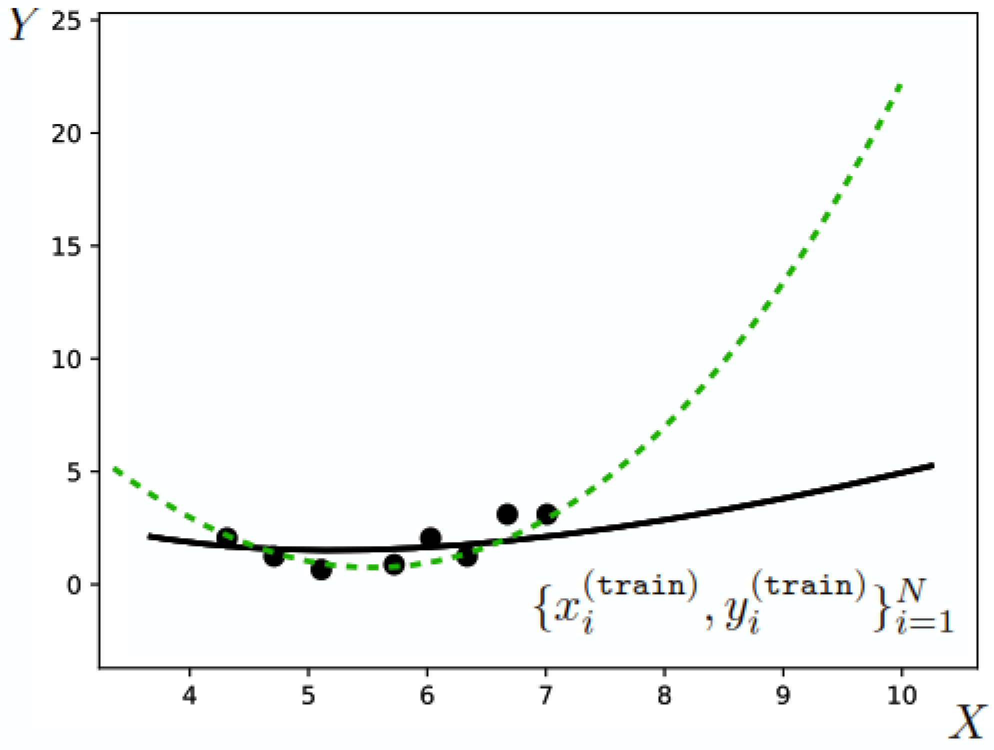


Much more commonly, we have

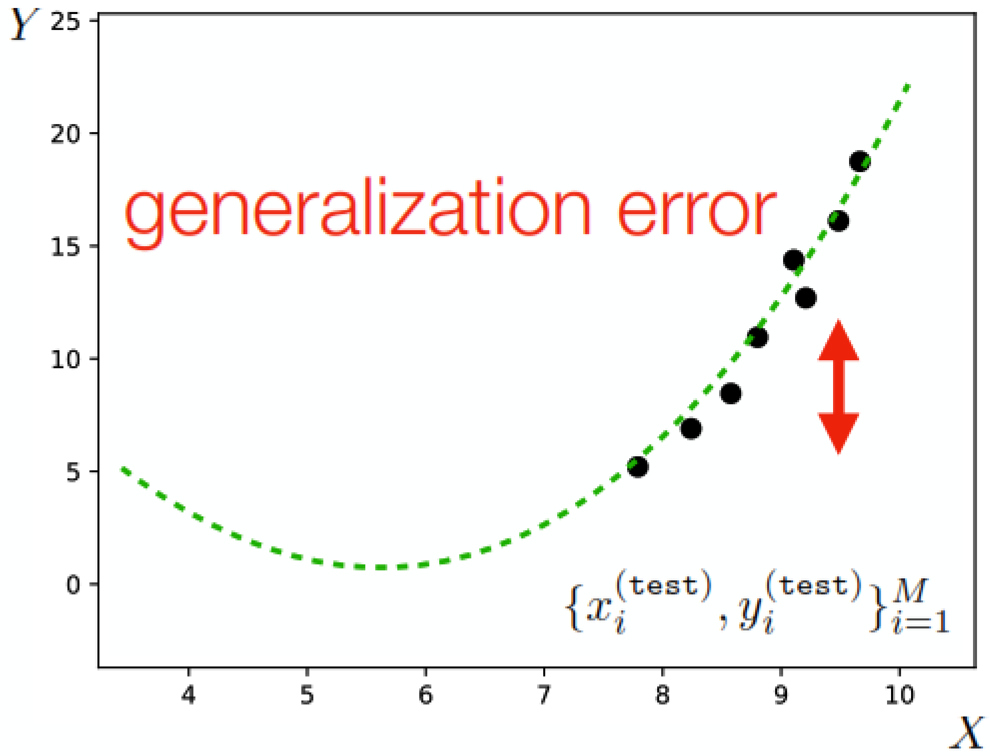
$$p_{\texttt{train}} \neq p_{\texttt{test}}$$

$$\boxed{\begin{array}{l} \{x_i^{(\texttt{train})}, y_i^{(\texttt{train})}\} \overset{\texttt{iid}}{\sim} p_{\texttt{train}} \\ \{x_i^{(\texttt{test})}, y_i^{(\texttt{test})}\} \overset{\texttt{iid}}{\sim} p_{\texttt{test}} \end{array}}$$

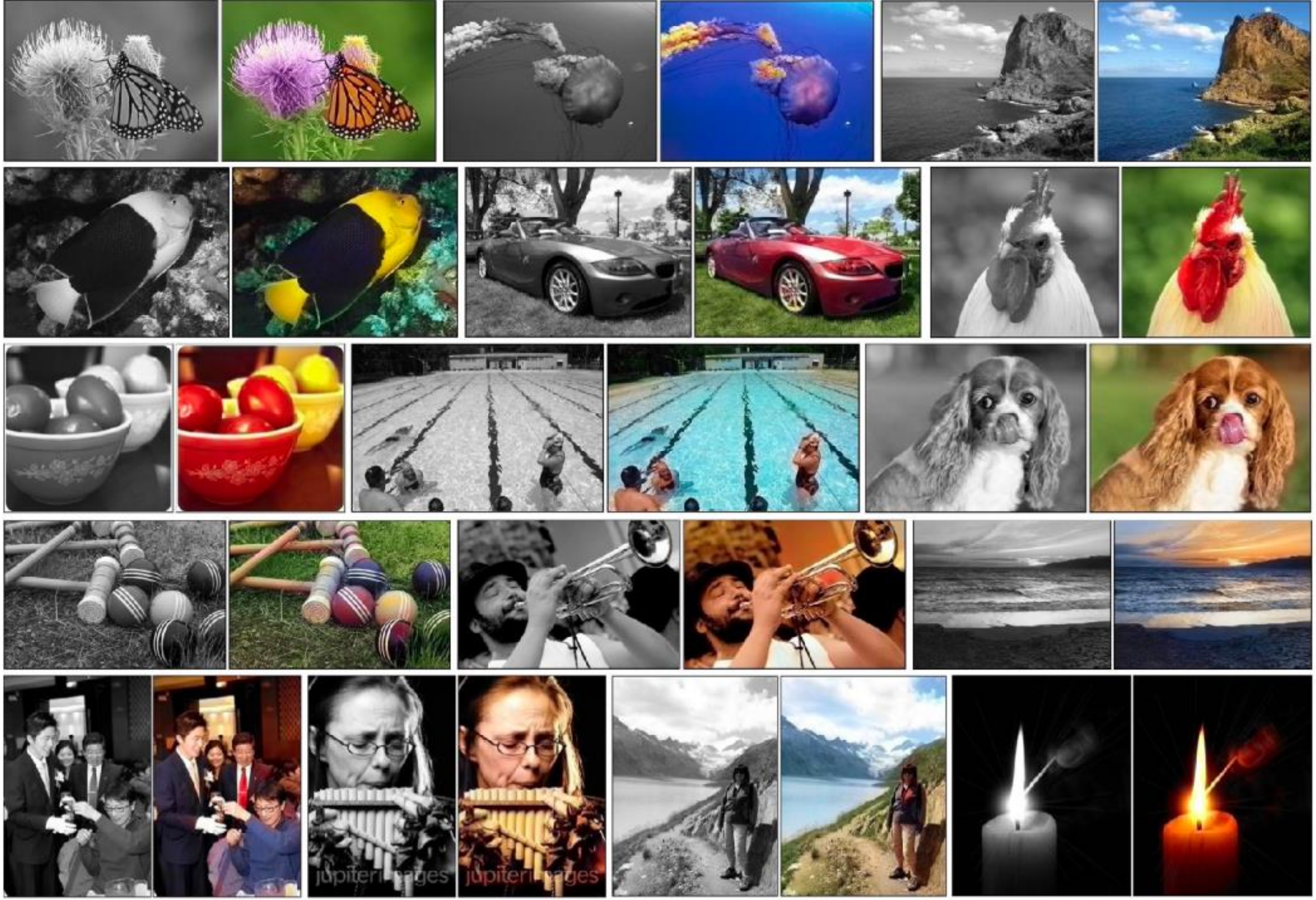
Training data

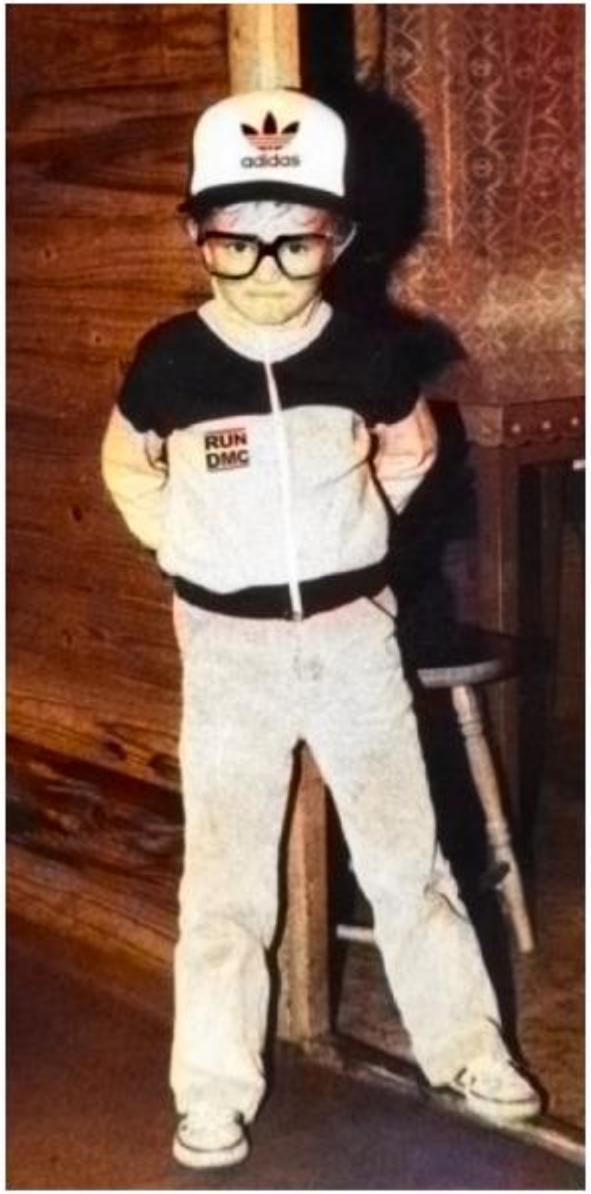


Test data



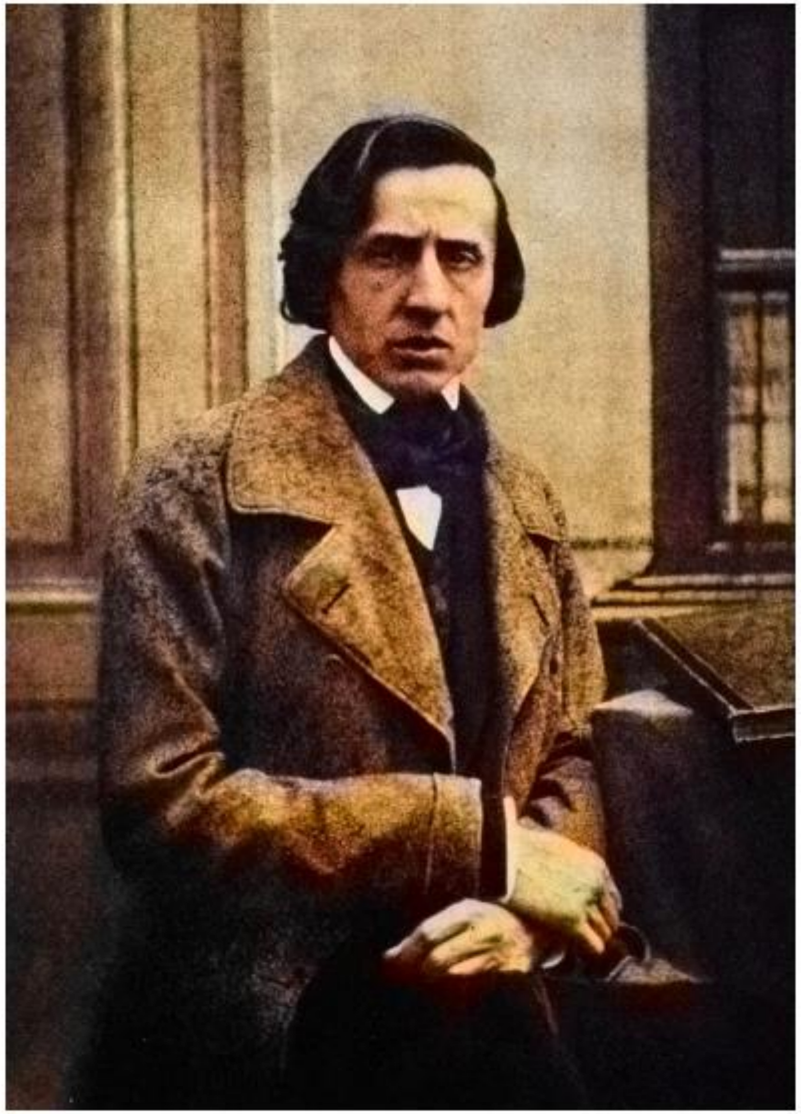
Our training data did cover the part of the distribution that was tested (**biased data**)





u/Rafael_P_S


Thylacine

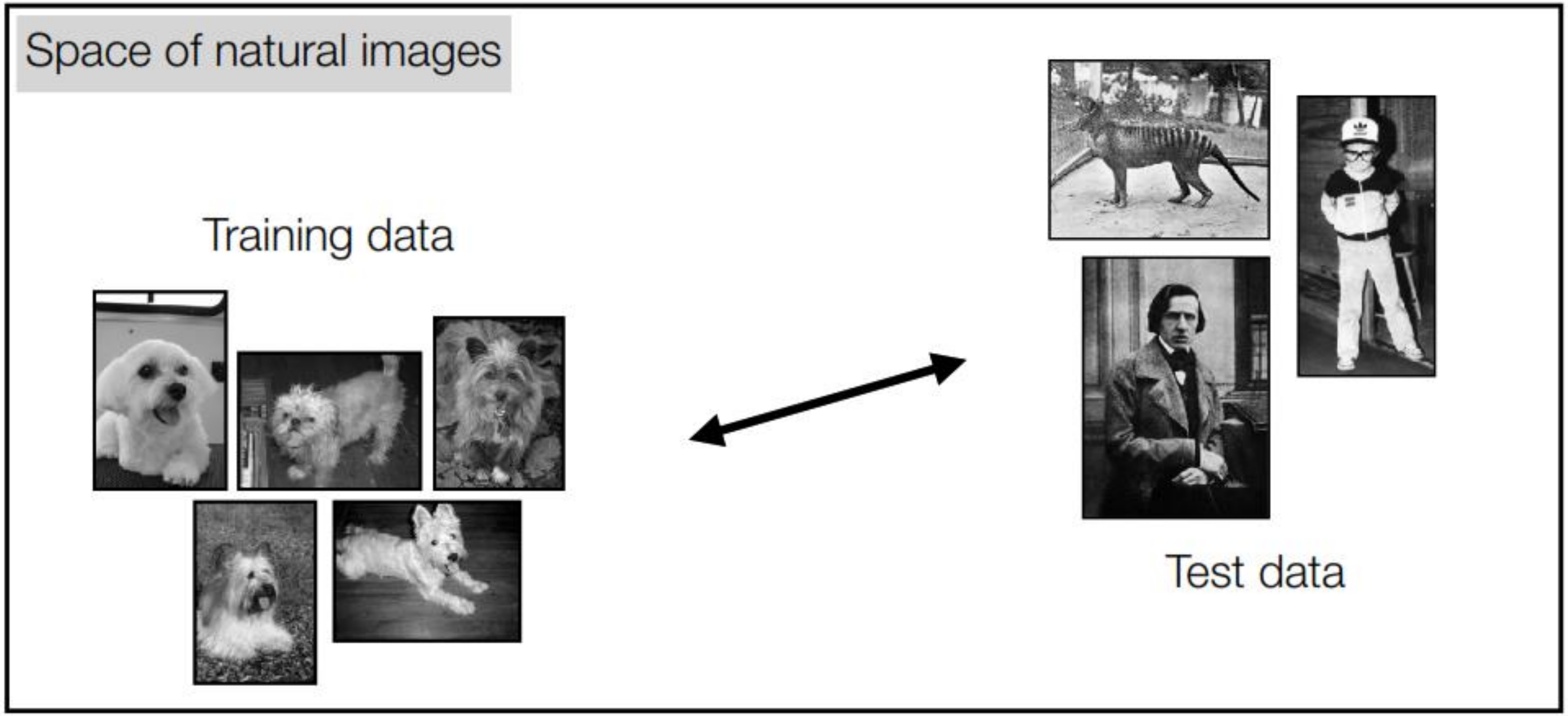

Chopin

training domain

testing domain

(where we actual use our model)

Domain gap between $p_{\texttt{train}}$ and $p_{\texttt{test}}$ will cause us to fail to generalize.



- How can we collect good data?
  - Correctly labeled
  - Unbiased (good coverage of all relevant kinds of data)

Crowdsourcing

- But Can humans collect good data?

Getting more humans in the annotation loop

Labeling to get a Ph.D.

Labeling for fun

Luis Von Ahn and Laura Dabbish 2004

Labeling for money

(Sorokin, Forsyth, 2008)

amazonmechanical turk

Artificial Artificial Intelligence

Labeling because it gives you added value

Visipedia

(Belongie, Perona, et al)

Just for labeling

LabelMe

Beware of the human in your loop

- What do you know about them?
- Will they do the work you pay for?

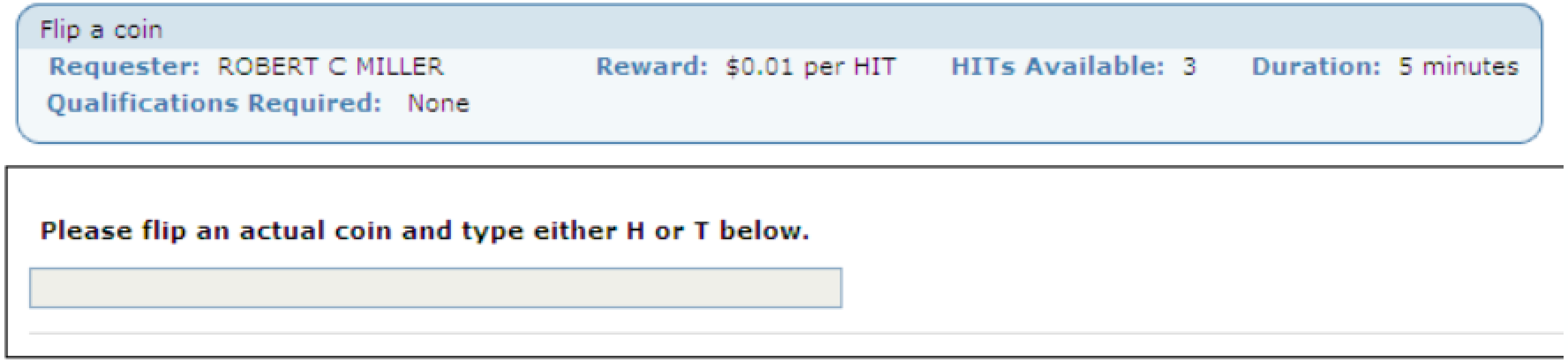
People have biases…

Turkers were offered 1 cent to pick a number from 1 to 10.

Experiment by Greg Little

From http://groups.csail.mit.edu/uid/deneme/

Do humans do what you ask for?


Flip a coin

Requester: ROBERT C MILLER   Reward: $0.01 per HIT   HITs Available: 3   Duration: 5 minutes

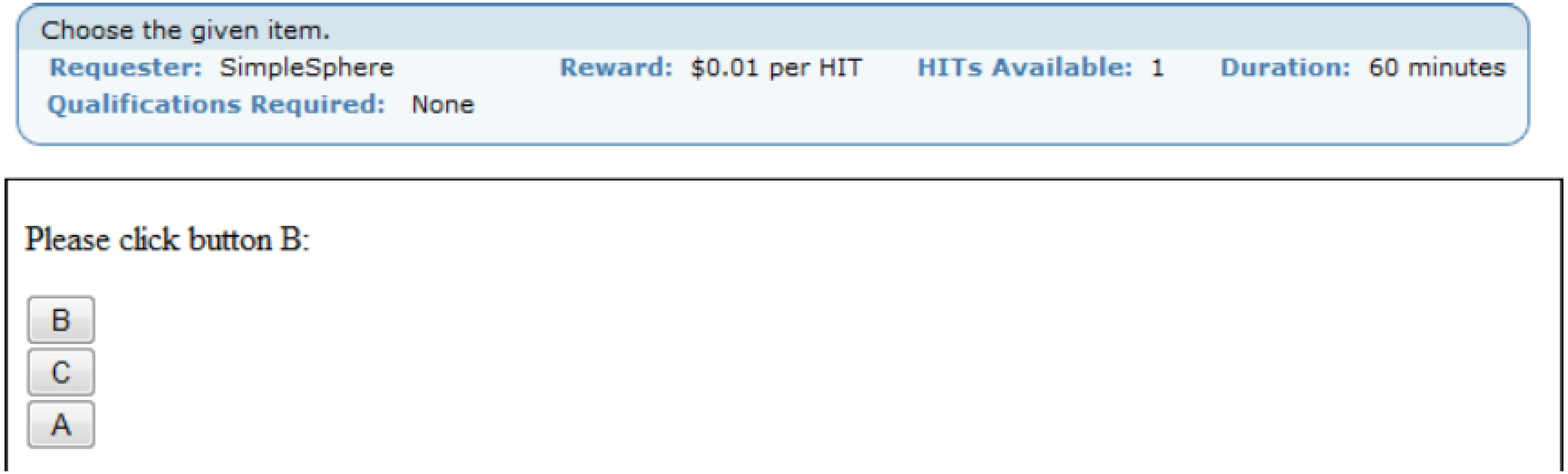
Qualifications Required: None

Please flip an actual coin and type either H or T below.

Experiment by Rob Miller

From http://groups.csail.mit.edu/uid/deneme/

Are humans reliable even in simple tasks?



Experiment by Greg Little

From http://groups.csail.mit.edu/uid/deneme/

- So we can sometimes collect good training data’

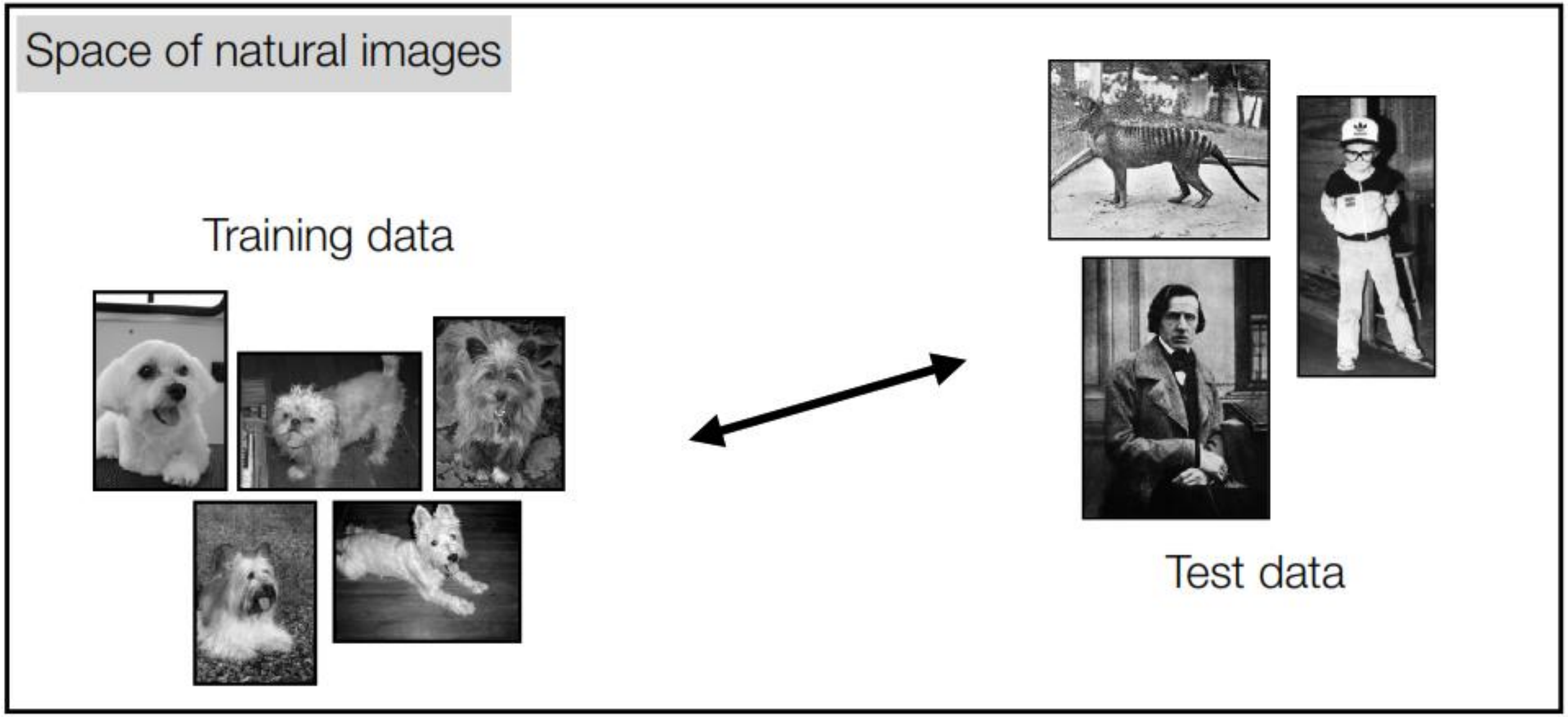
- But suppose we messed up. Our test setting does not look like the training data!

- How can we bridge the domain gap?

training domain

testing domain
(where we actual use our model)

Domain gap between $p_{\texttt{train}}$ and $p_{\texttt{test}}$ will cause us to fail to generalize.

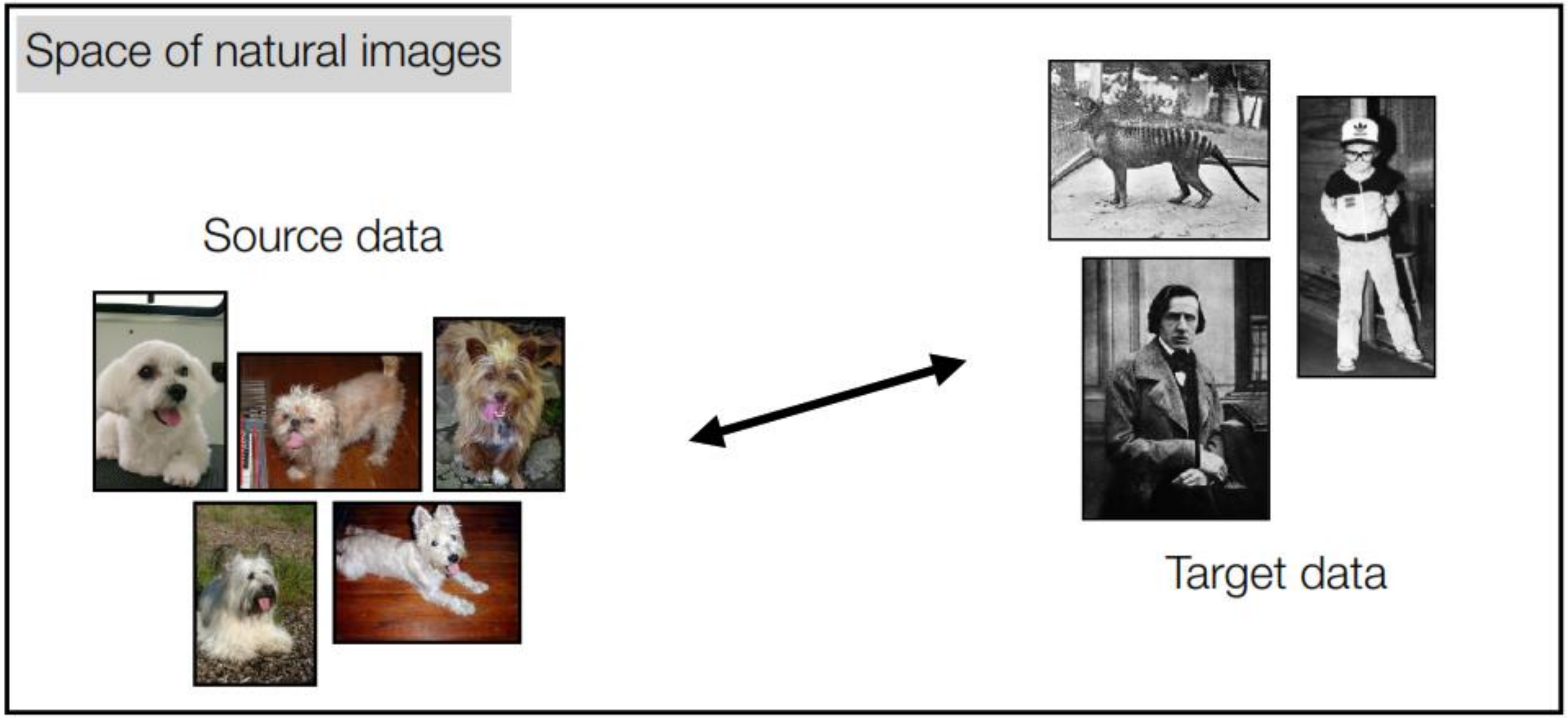


source domain

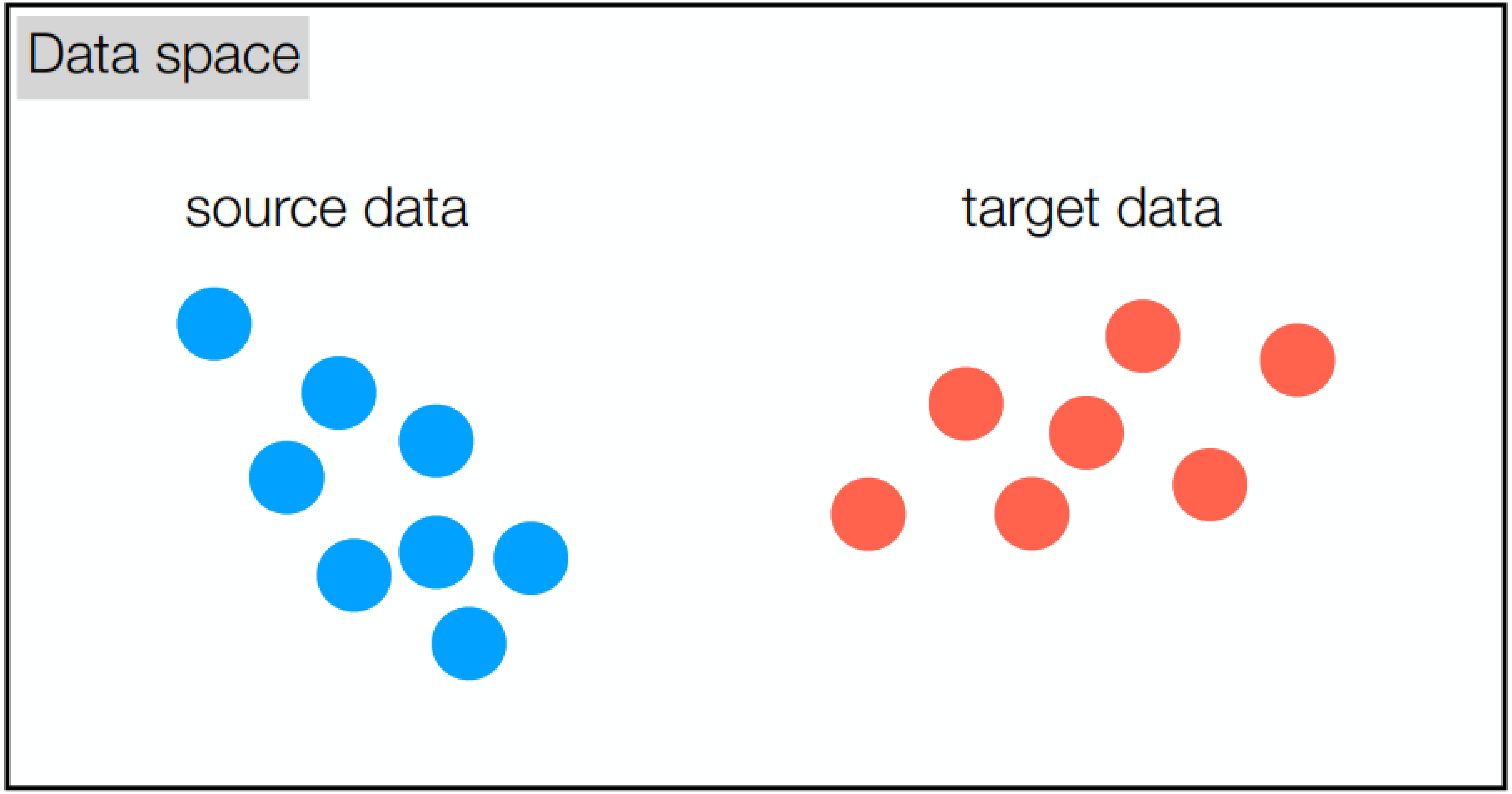
target domain

(where we actual use our model)

Domain gap between $p_{\texttt{source}}$ and $p_{\texttt{target}}$ will cause us to fail to generalize.



Idea #1: transform the target domain to look like the source domain

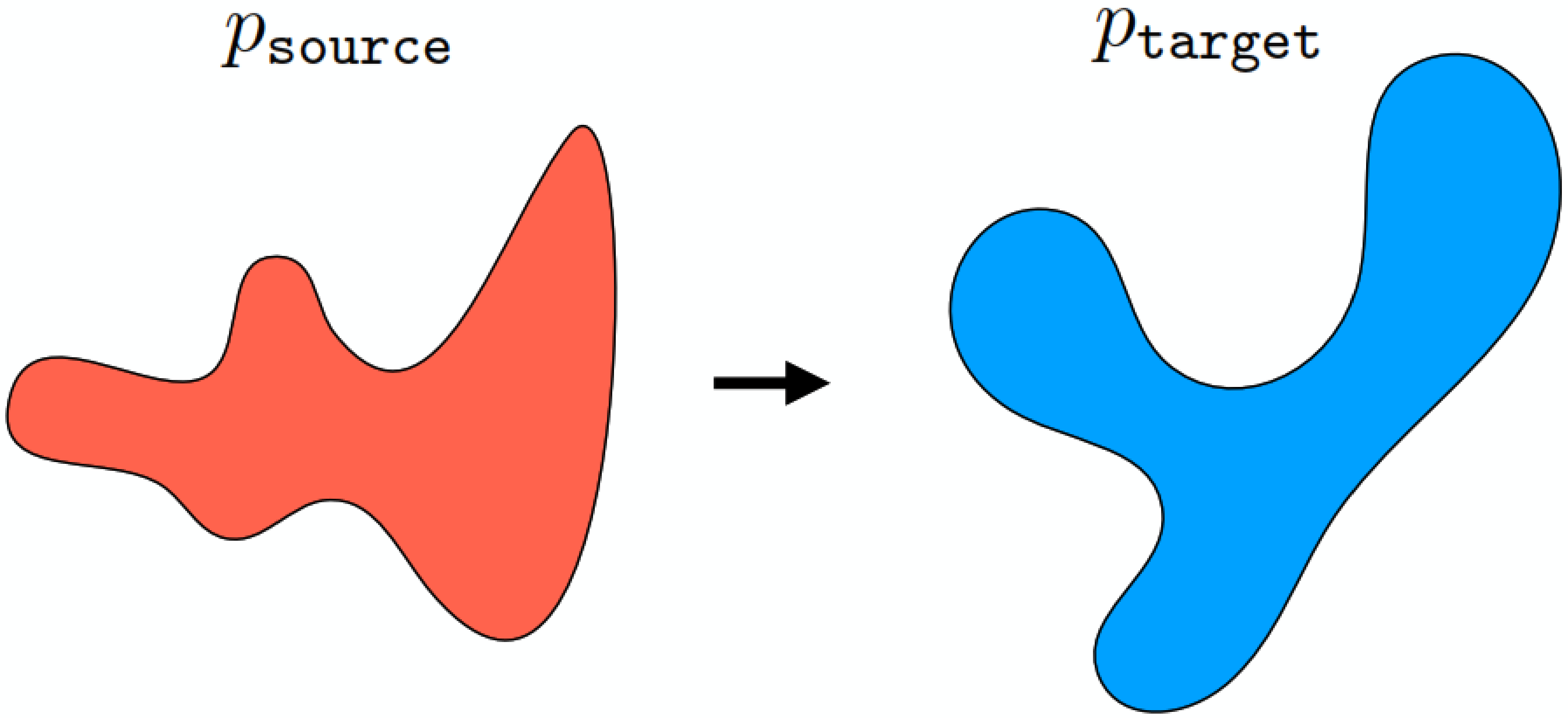


(Or vice versa)

This is called **domain adaptation**

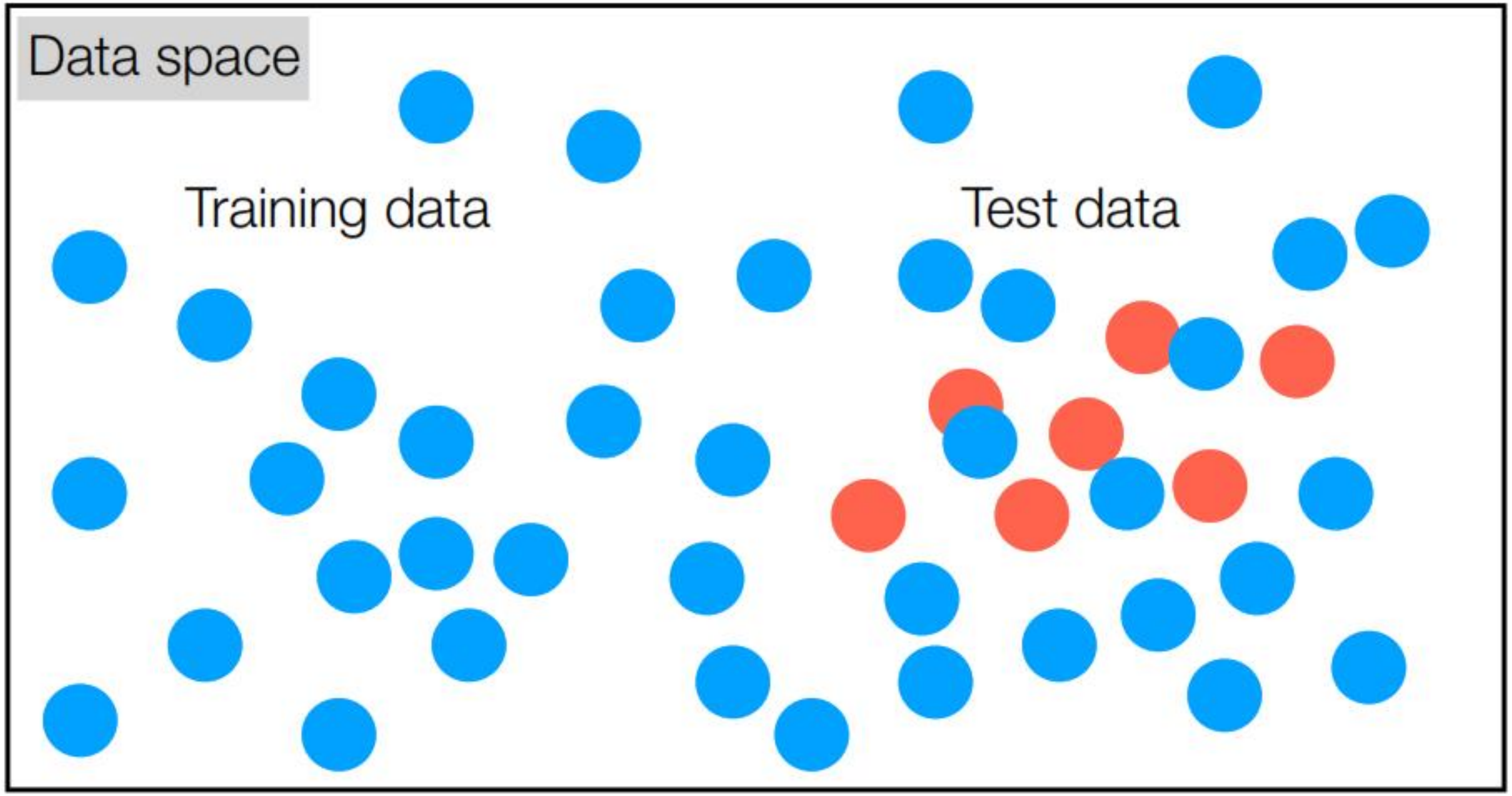
Domain adaptation

- We have source domain pairs $\{\mathbf{x}^{\mathbf{source}}, \mathbf{y}^{\mathbf{source}}\}$
- Learn a mapping F: $\mathbf{x}^{\mathbf{source}} \longrightarrow \mathbf{y}^{\mathbf{source}}$
- We want to apply F to target domain data $\mathbf{x}^{\mathbf{target}}$
- Find transformation T: $\mathbf{x}^{\mathbf{target}} \longrightarrow \mathbf{x}^{\mathbf{source}}$
- Now apply $F(T(\mathbf{x}^{\mathbf{target}}))$ to predict $\mathbf{y}^{\mathbf{target}}$



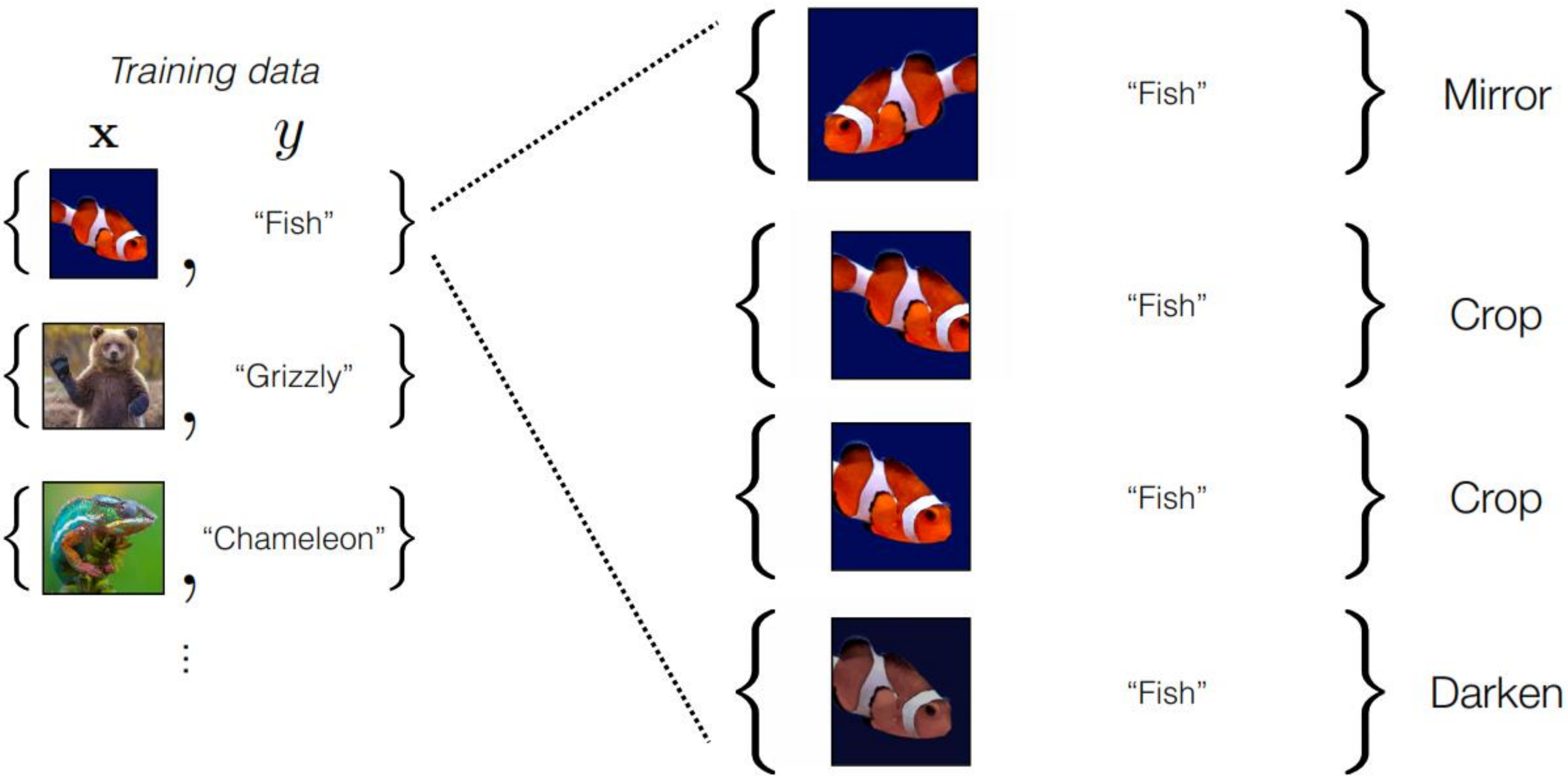
It’s a just another distribution mapping problem!

Idea #2: train on randomly perturbed data, so that test set just looks like another random perturbation



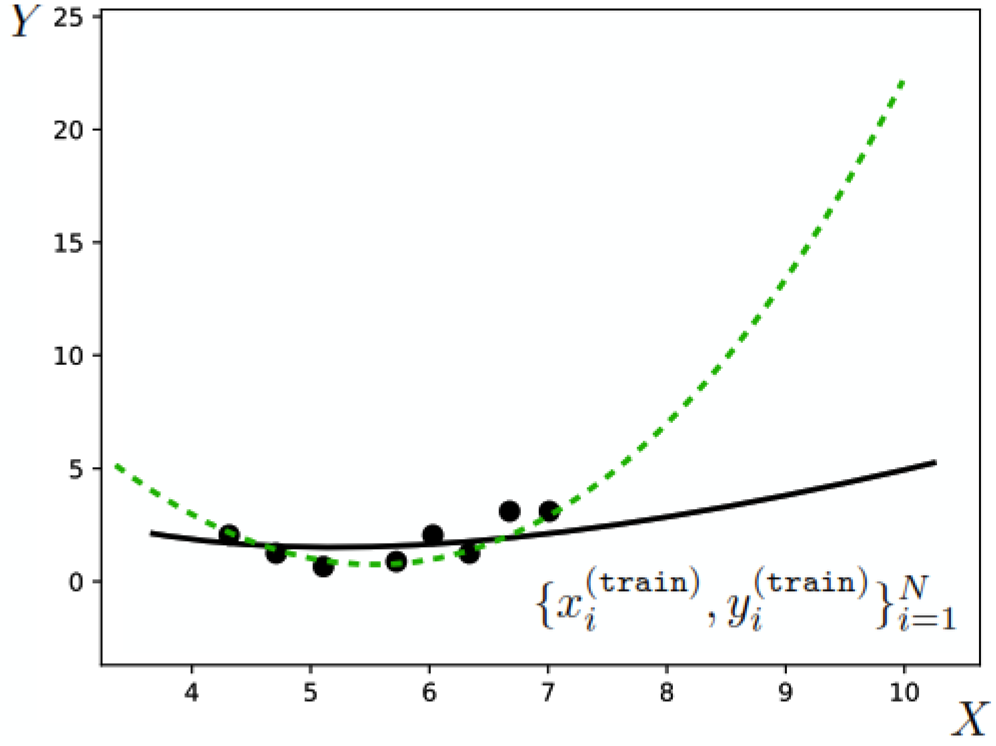
This is called **domain randomization** or **data augmentation**

Data augmentation

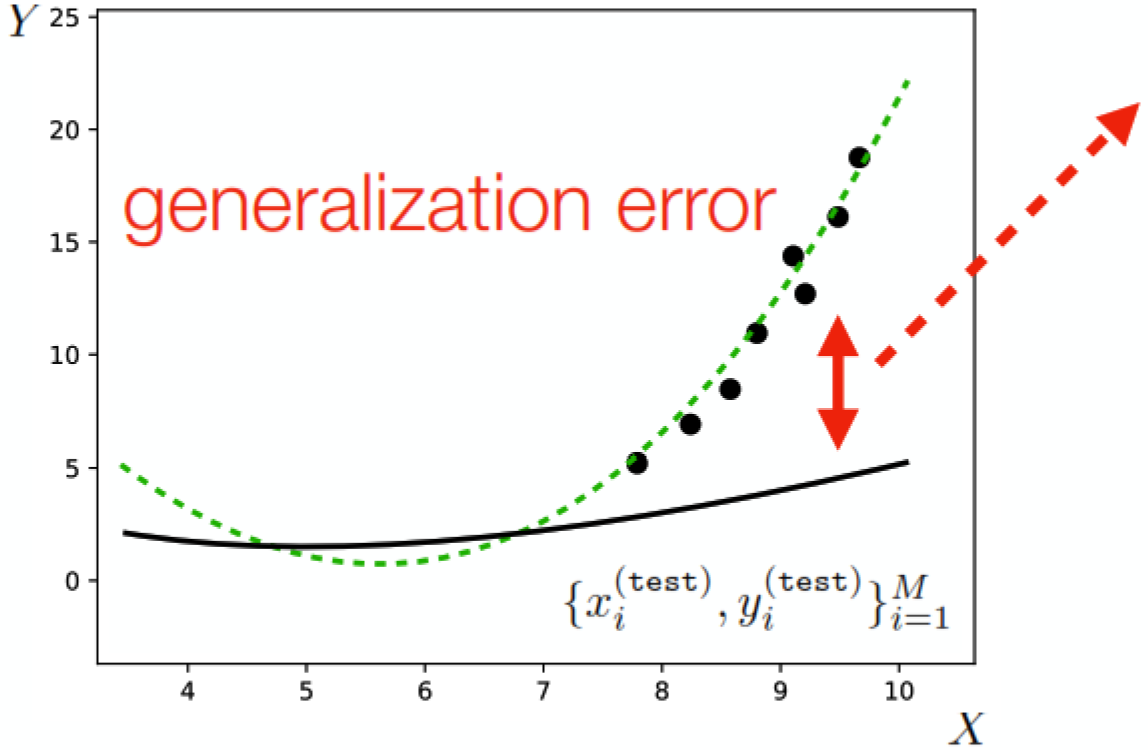


What if we go waaaay outside of the training distribution?

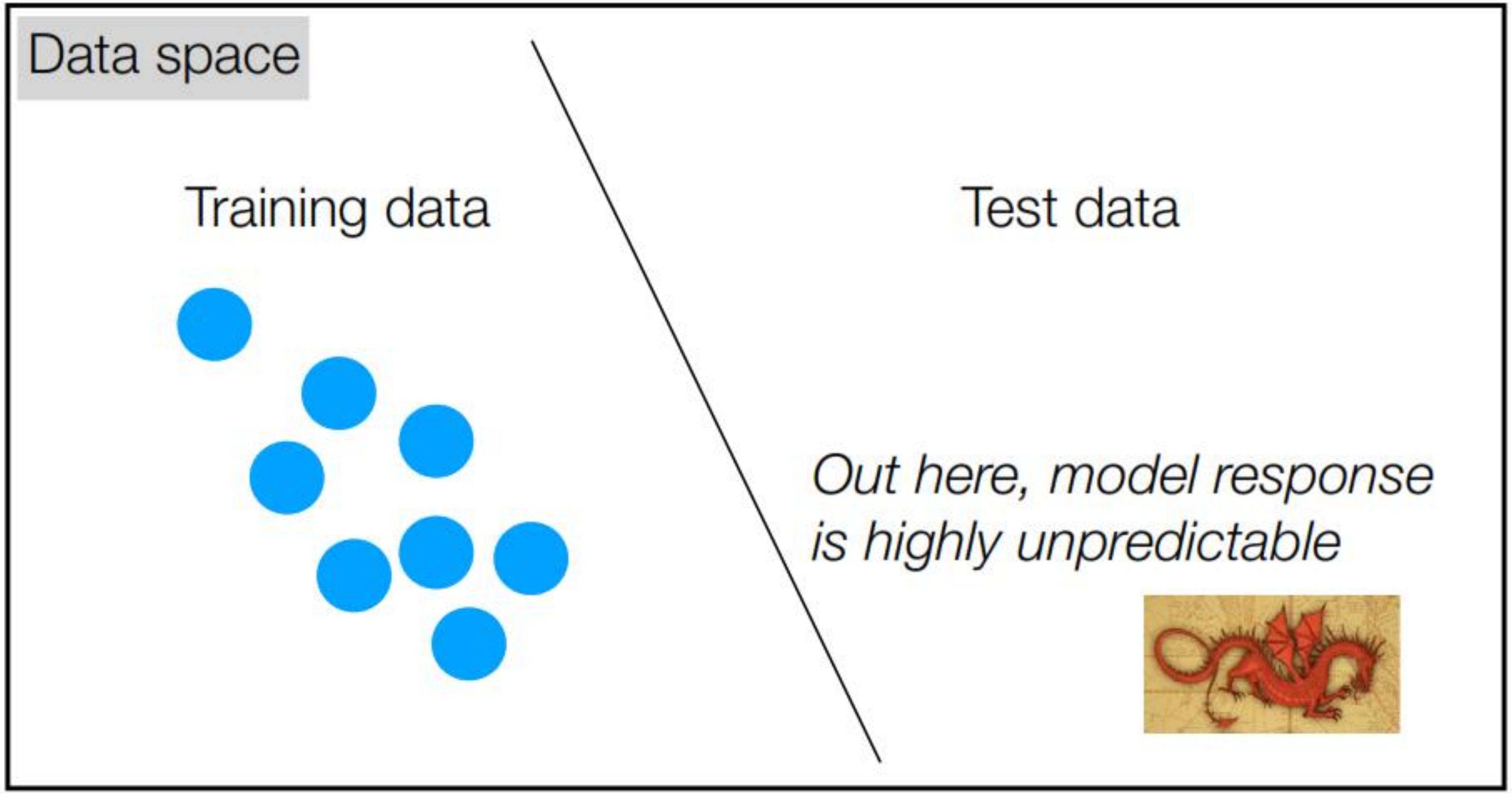
Training data



Test data

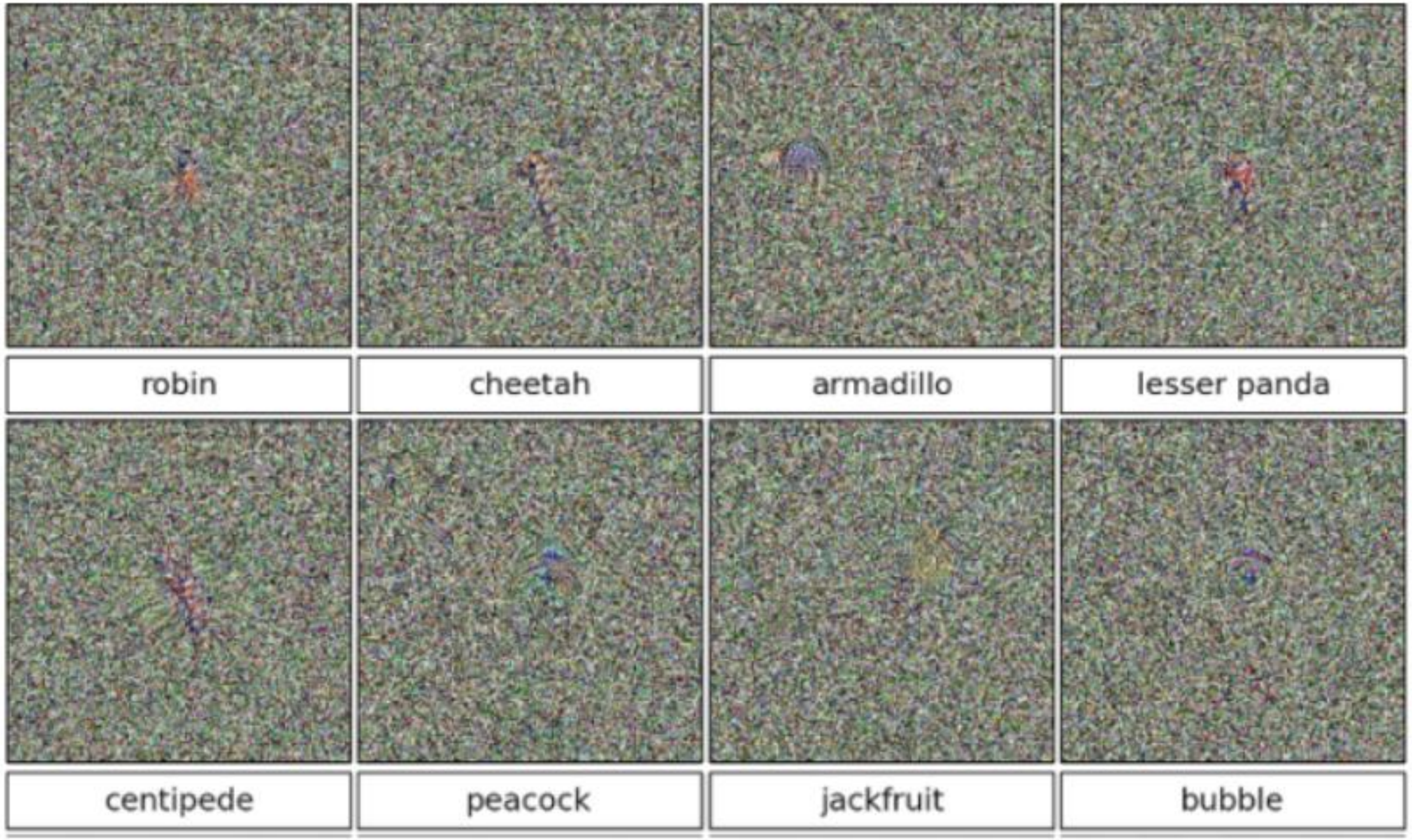


Our training data did not cover the part of the distribution that was tested (**biased data**)

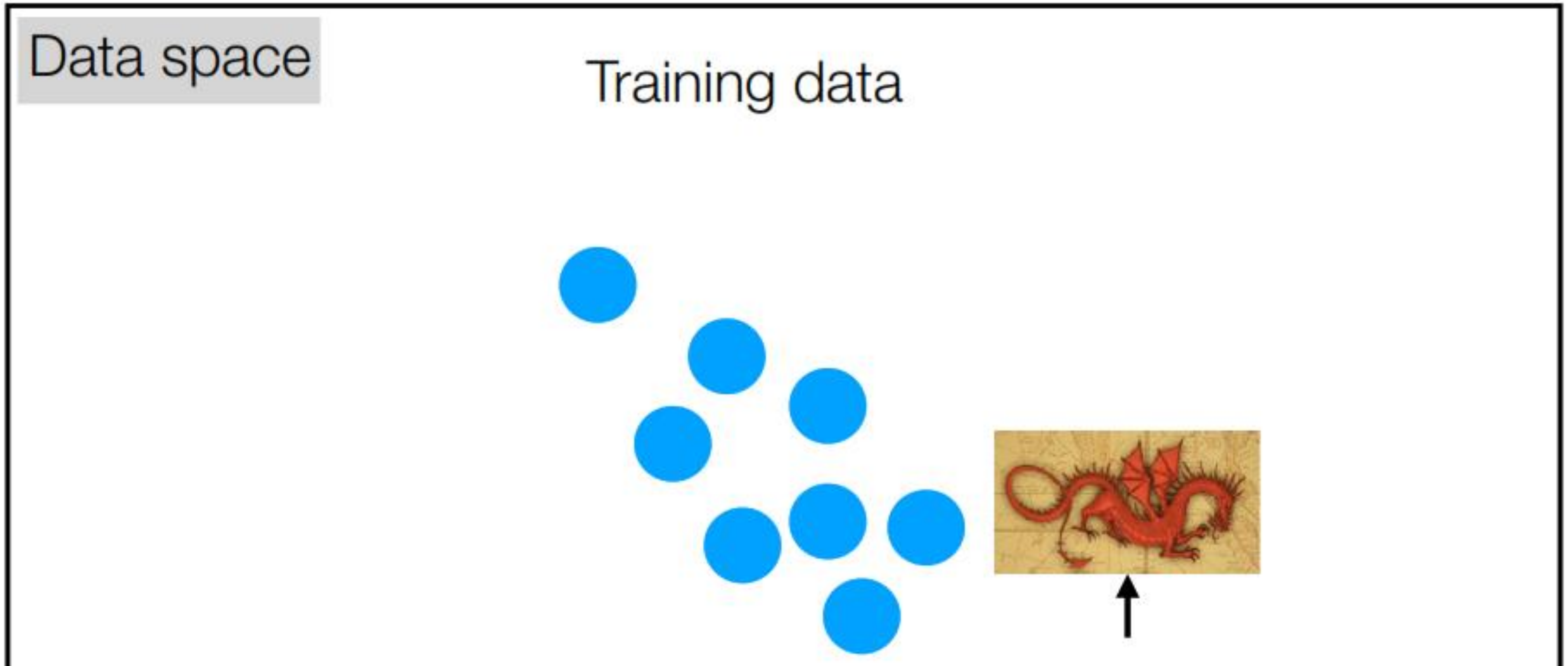

Data space
Training data
Test data
Out here, model response
is highly unpredictable

“Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images”

[Nguyen, Yosinski, and Clune, CVPR 2015]



Weirdness of high-dimensional space:

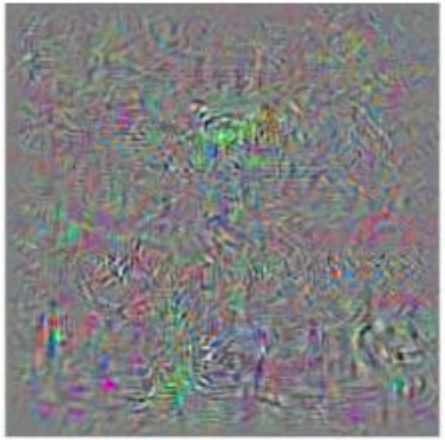


Usually, there are *blind spots* where the model has not fit the distribution well, and behaves unpredictably

Adversarial noise

$\mathbf{x}$ + $\mathbf{r}$ = $\mathbf{x} + \mathbf{r}$

+



=

$\Downarrow f$ $\quad$ $\Downarrow f$

y “School bus” “Ostrich”

$$\arg\max_{\mathbf{r}} p(y = \texttt{ostrich} | \mathbf{x} + \mathbf{r}) \quad \text{subject to} \quad \|\mathbf{r}\| < \epsilon$$

[“Intriguing properties of neural networks”, Szegedy et al. 2014]

Anything to worry about?

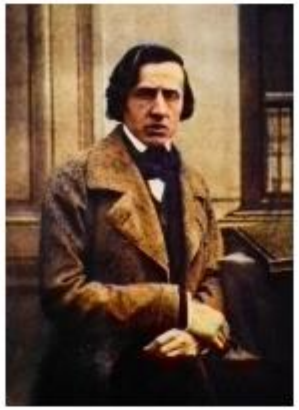
- Current deep models have bad **worst-case performance**
- Can be exploited by an adversary
- Few guarantees, can’t fully trust what the model’s output

Anything else to worry about?

- Our datasets are often poorly labeled

- And usually biased (overrepresent certain categories)

- ML method perform beautifully on laboratory data, but often generalize poorly to real-world data



- Can have negative social consequences