Lecture 24

Datasets, bias, and adaptation

Fall 2019

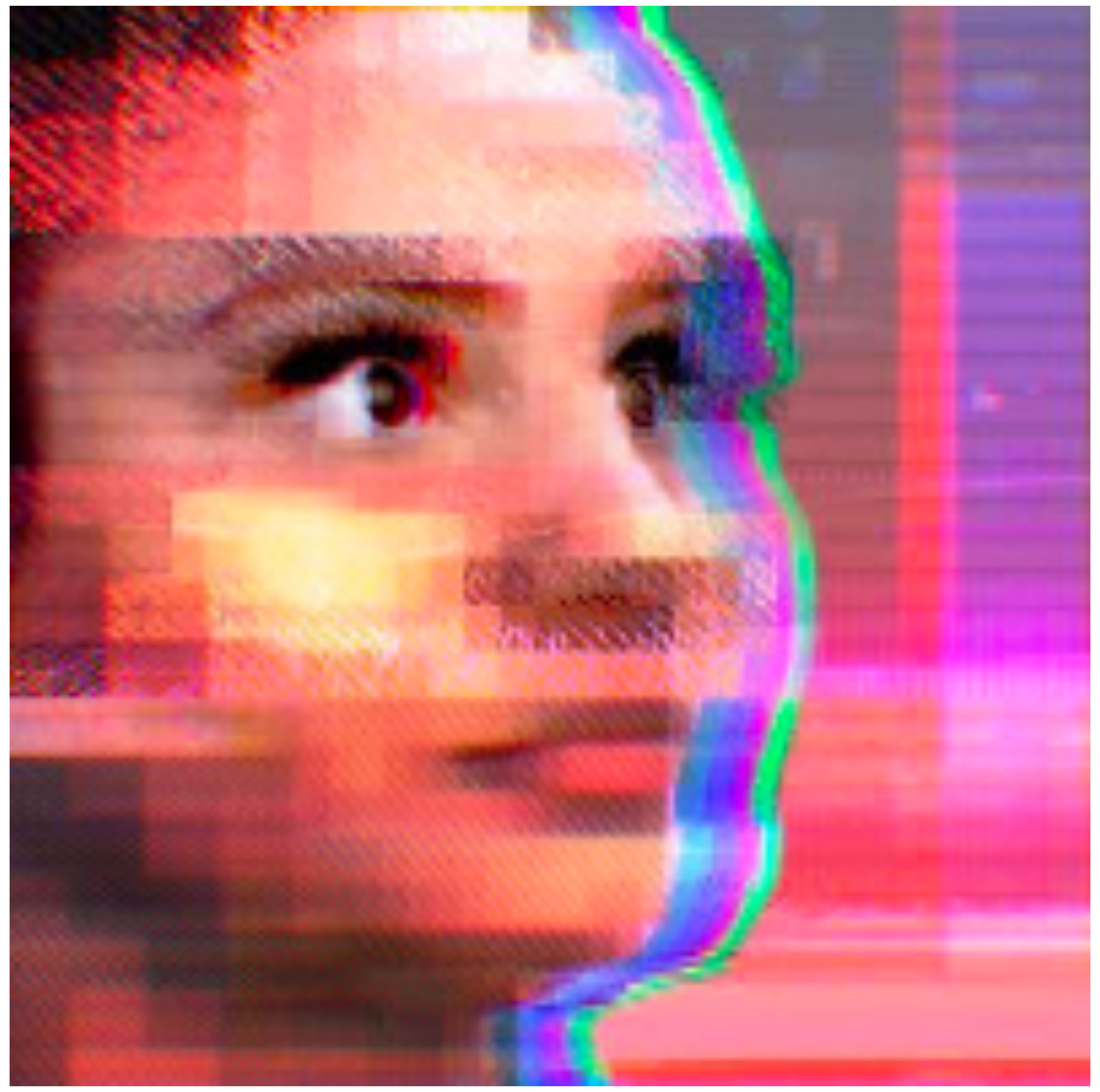
Bill Freeman, Antonio Torralba, Phillip Isola

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.

what is the yellow thing?

Submit

Predicted top-5 answers with confidence:

frisbee

79.844%

surfboard

7.319%

banana

2.844%

lemon

2.438%

surfboards

1.252%

how many trains are in the picture?

Submit

Predicted top-5 answers with confidence:

3

30.233%

5

18.270%

4

17.000%

2

11.343%

6

7.806%

Of number questions (e.g. “how many…”), 26.04% of the time, the answer is 2

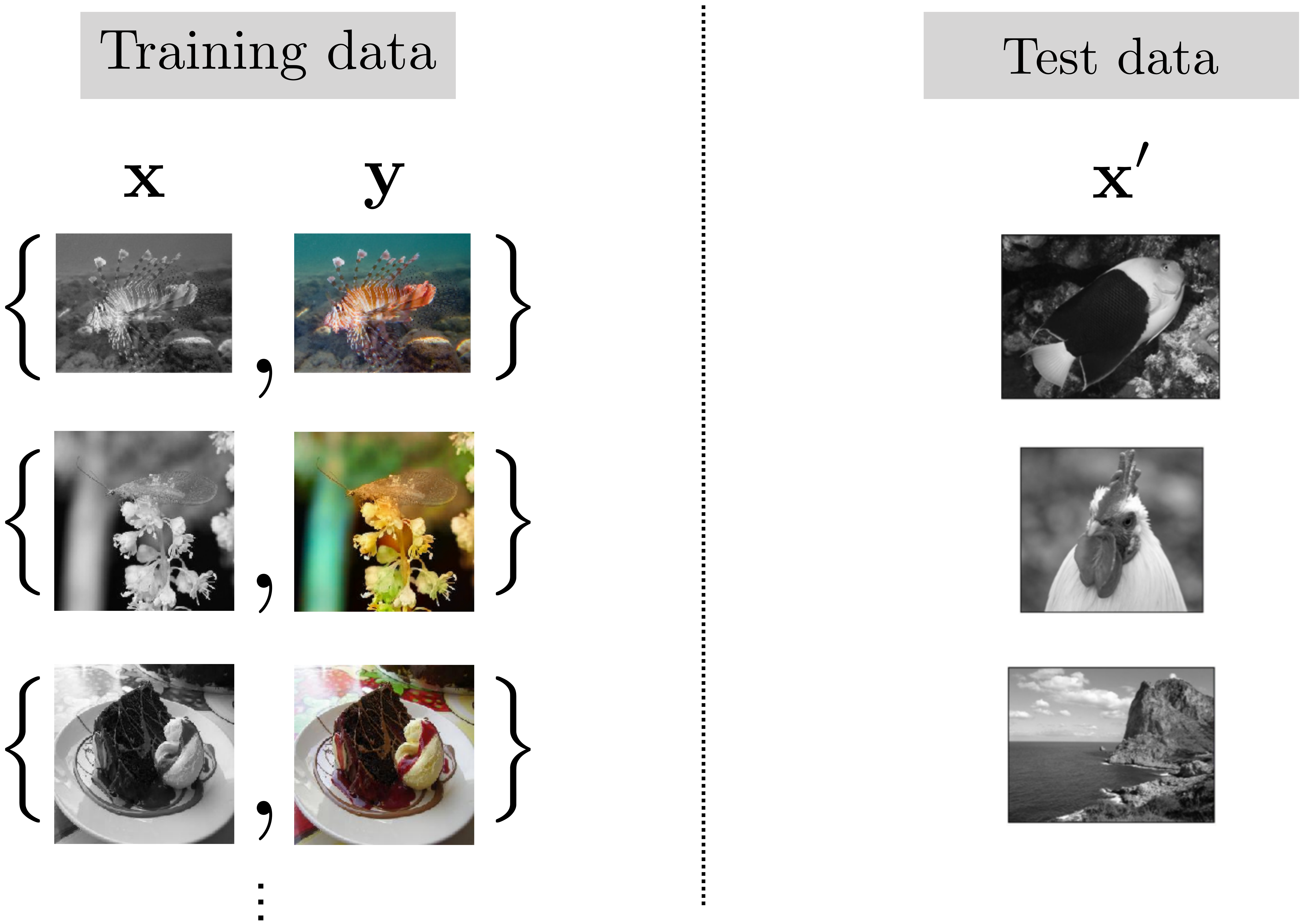
Of yes/no questions, 58.83% of the time, the answer is yes

[VQA, Agrawal, Lu, Antol et al., https://arxiv.org/pdf/1505.00468.pdf

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

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

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

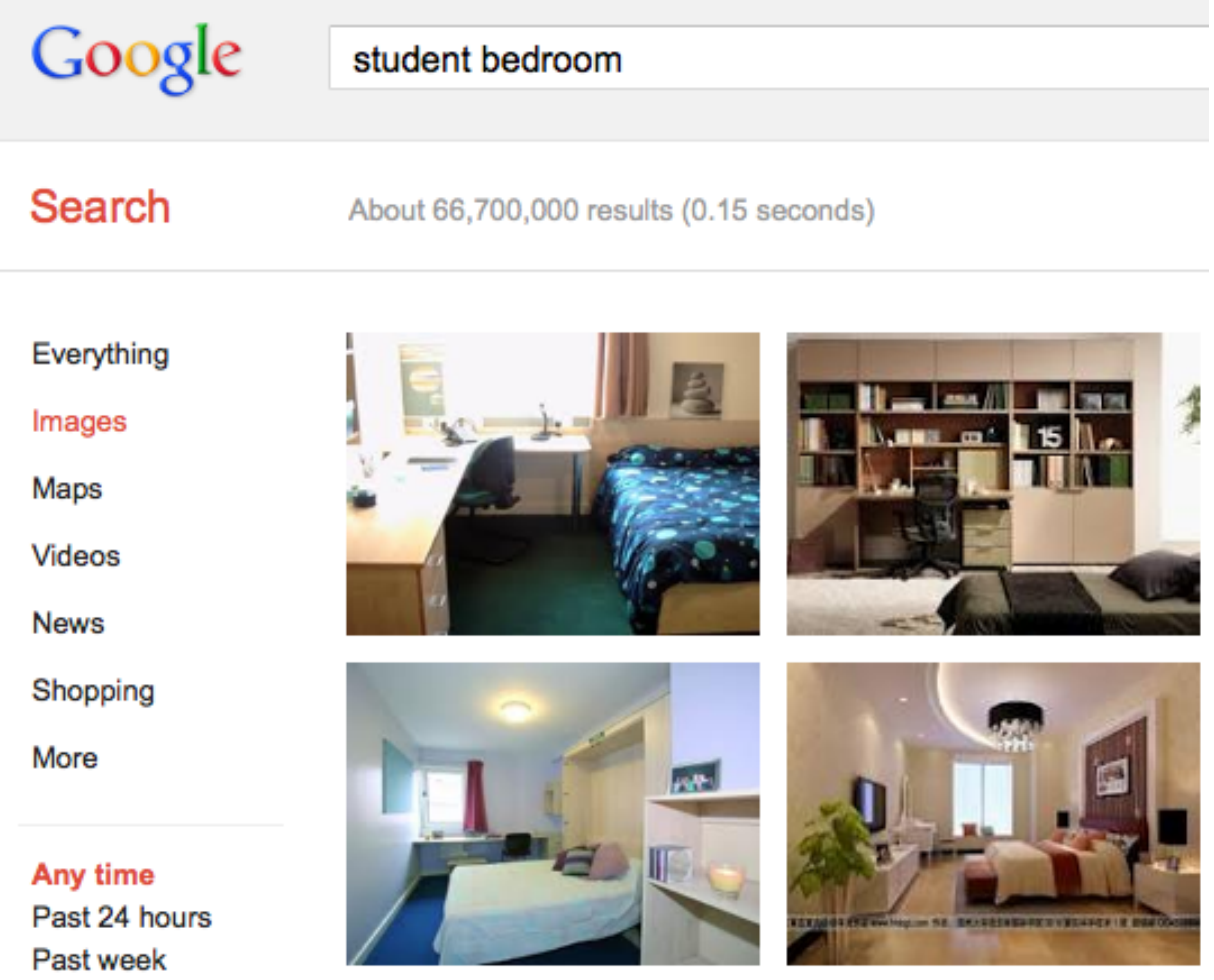

Training data
Test data
$\mathbf{x}$
$\mathbf{y}$
$\mathbf{x}'$

Training data

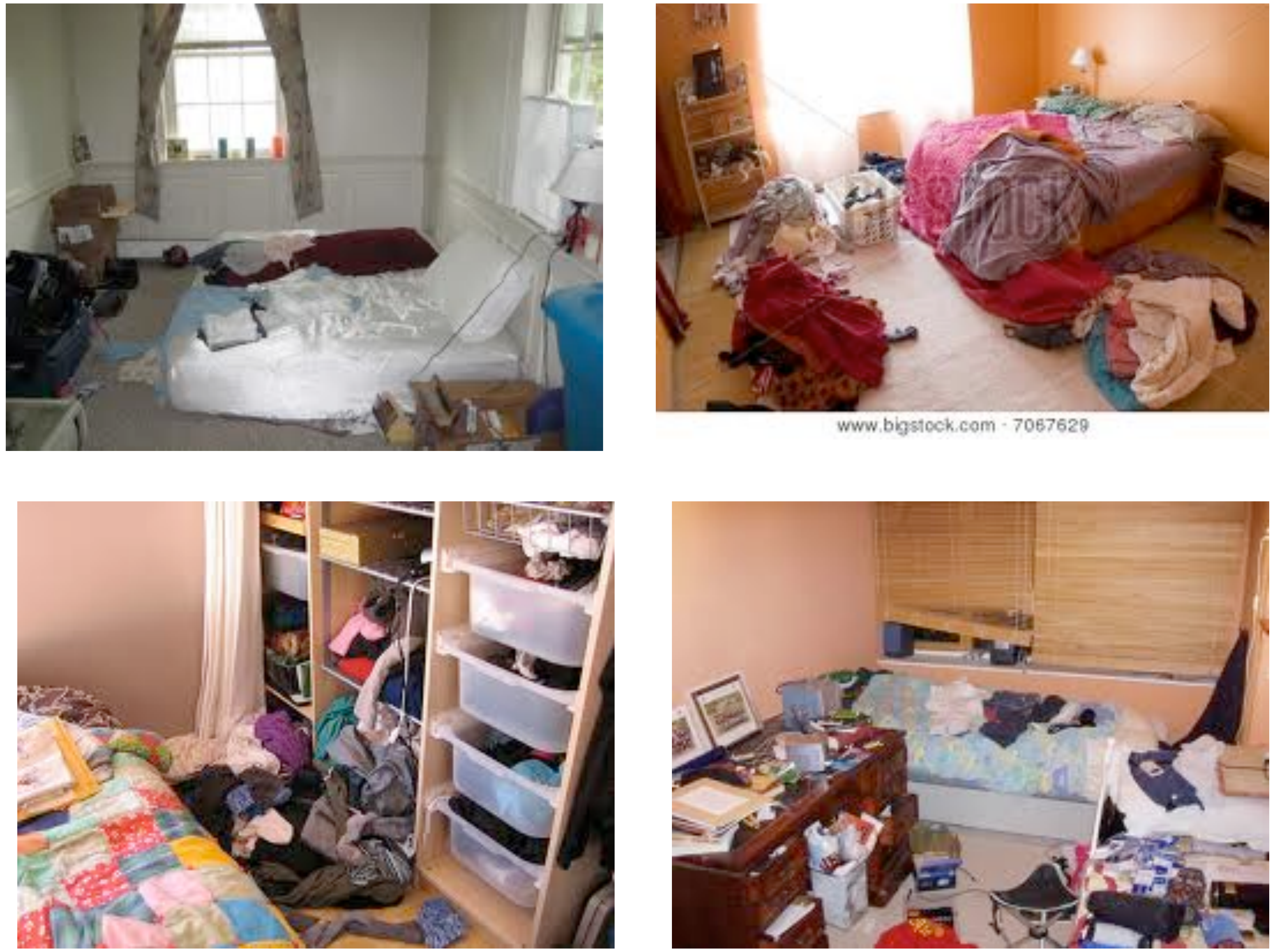
Test data

Training data

What Google thinks are student bedrooms



Test data



Training data

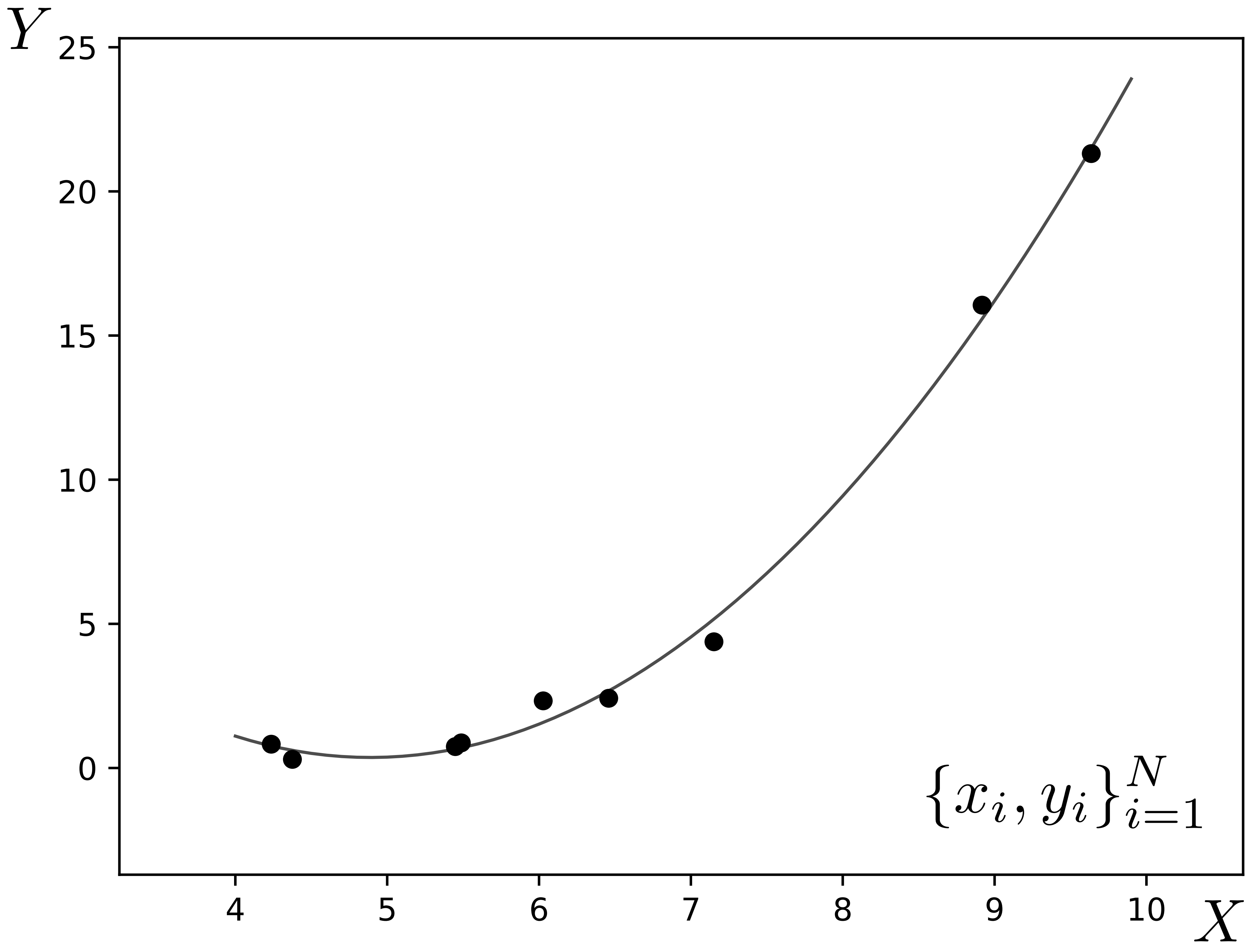
Driving simulator (GTA)

Test data

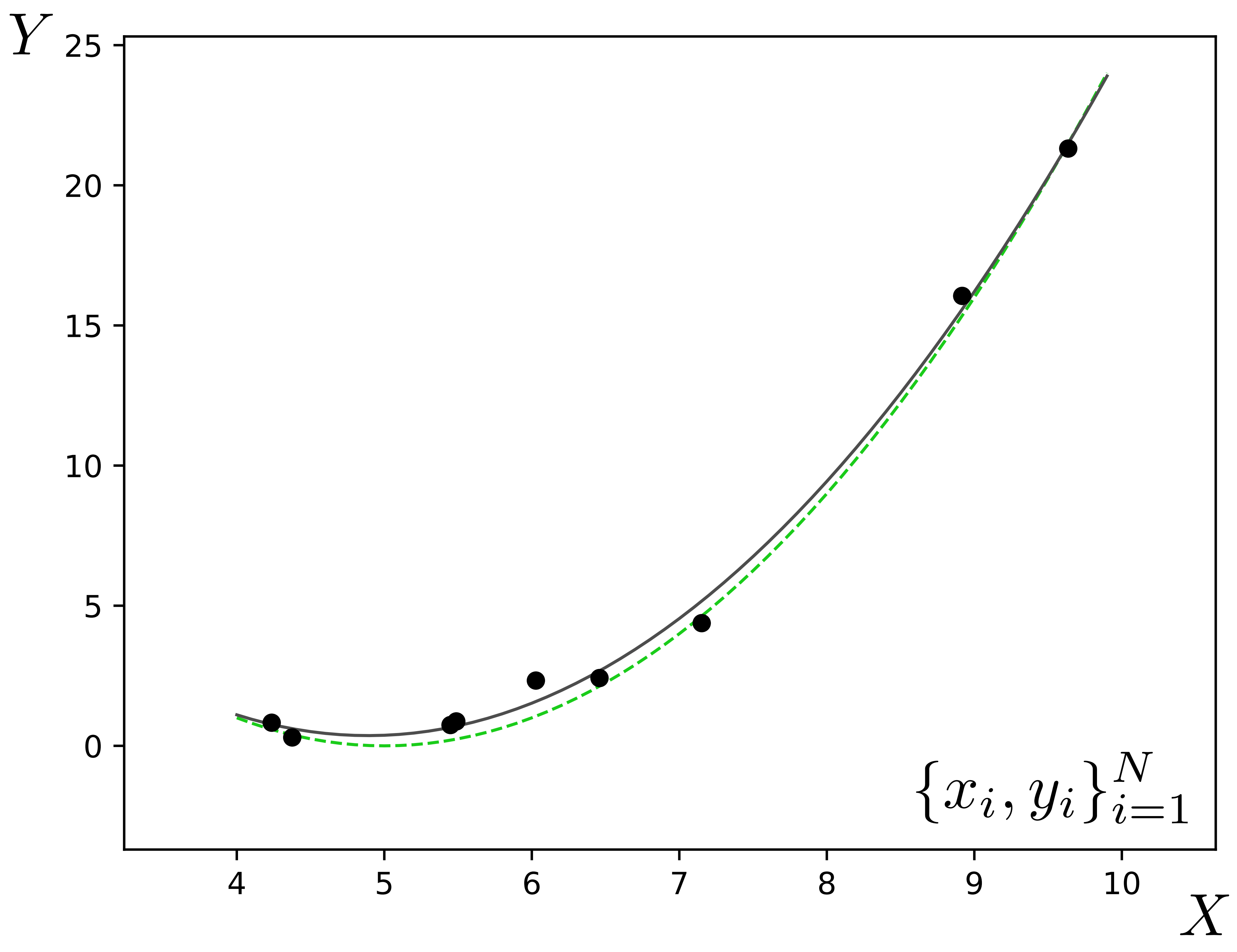
Driving in the real world

Let’s revisit the problem of generalization

Training data



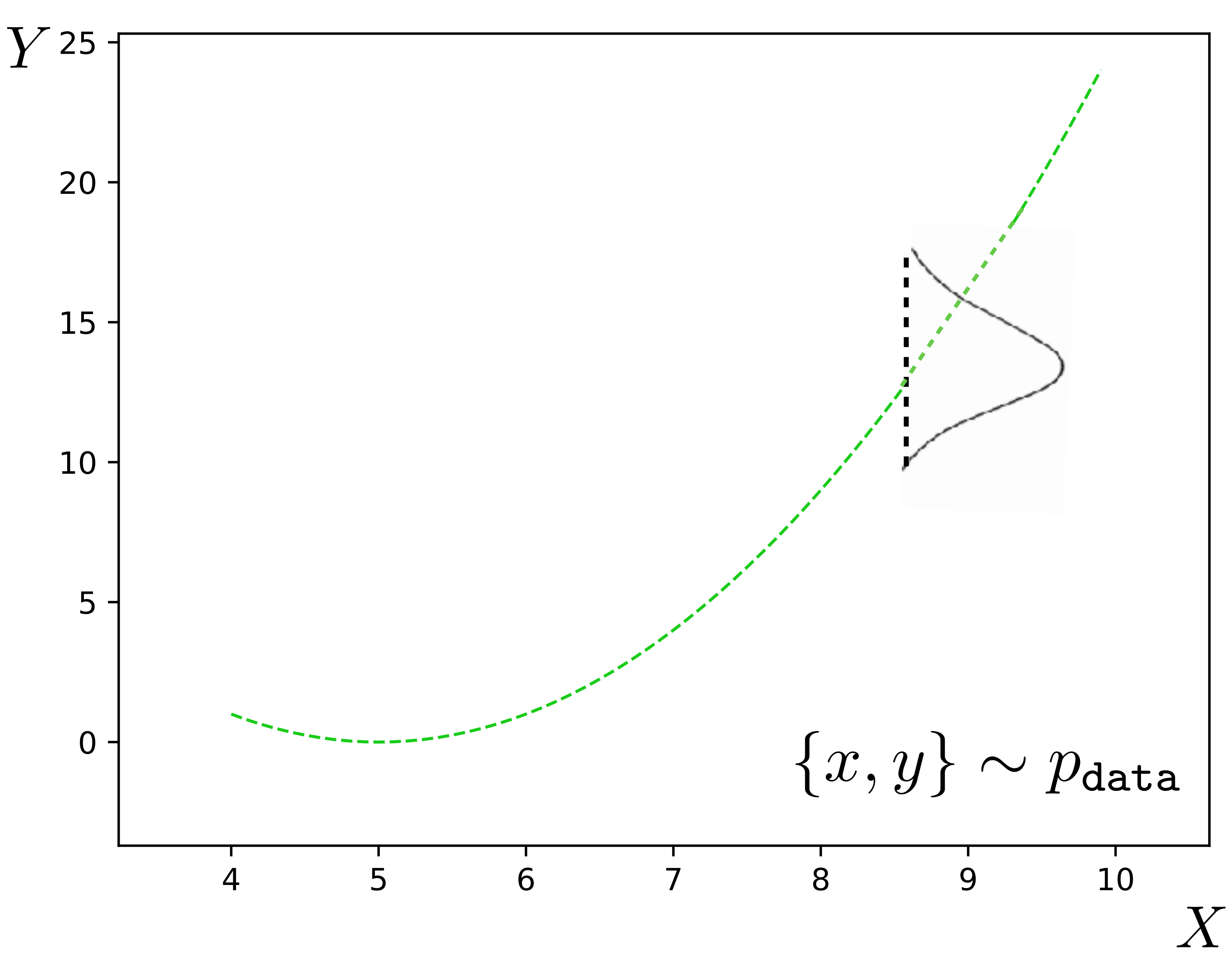
Training data



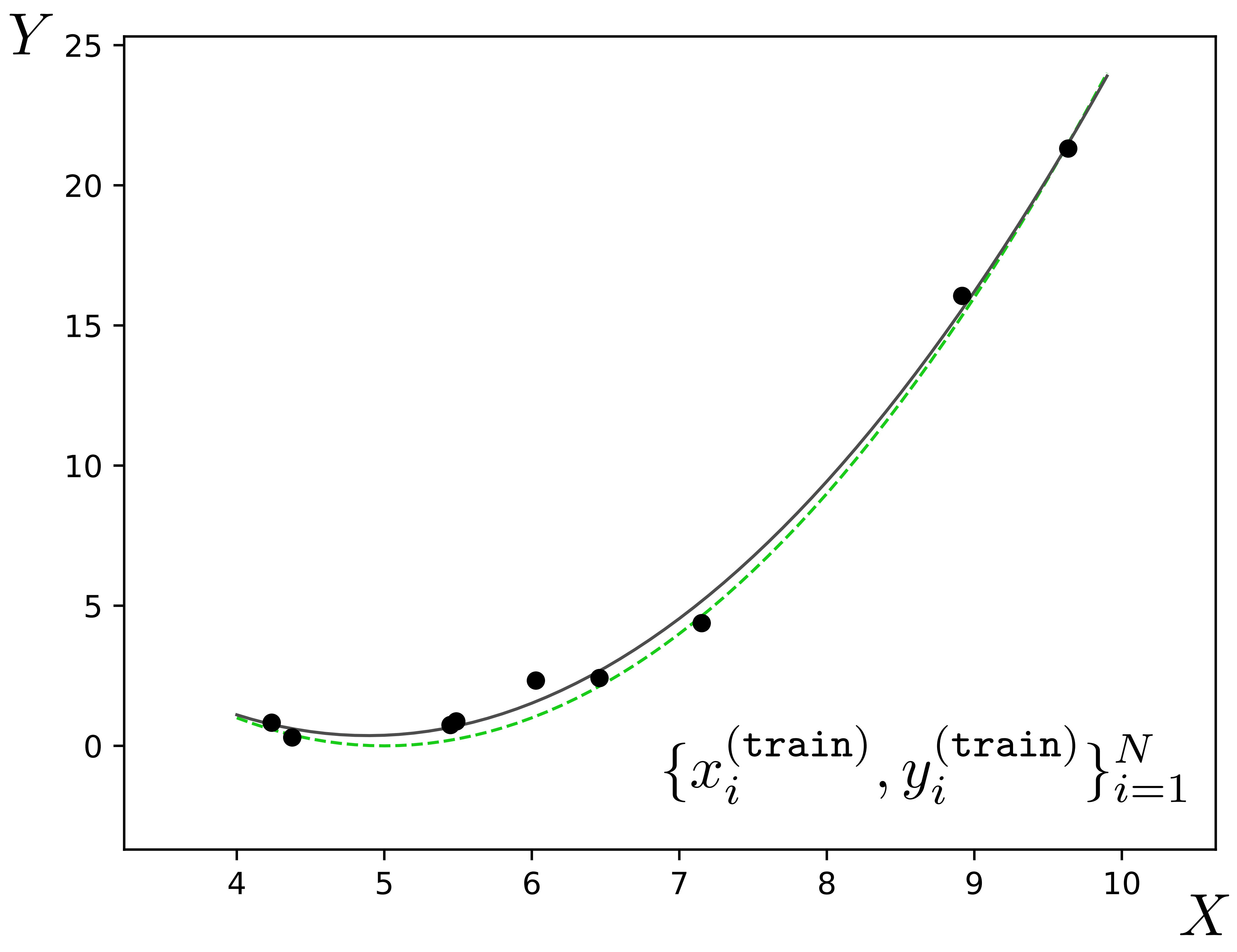
True **data-generating process**

$p_{\texttt{data}}$

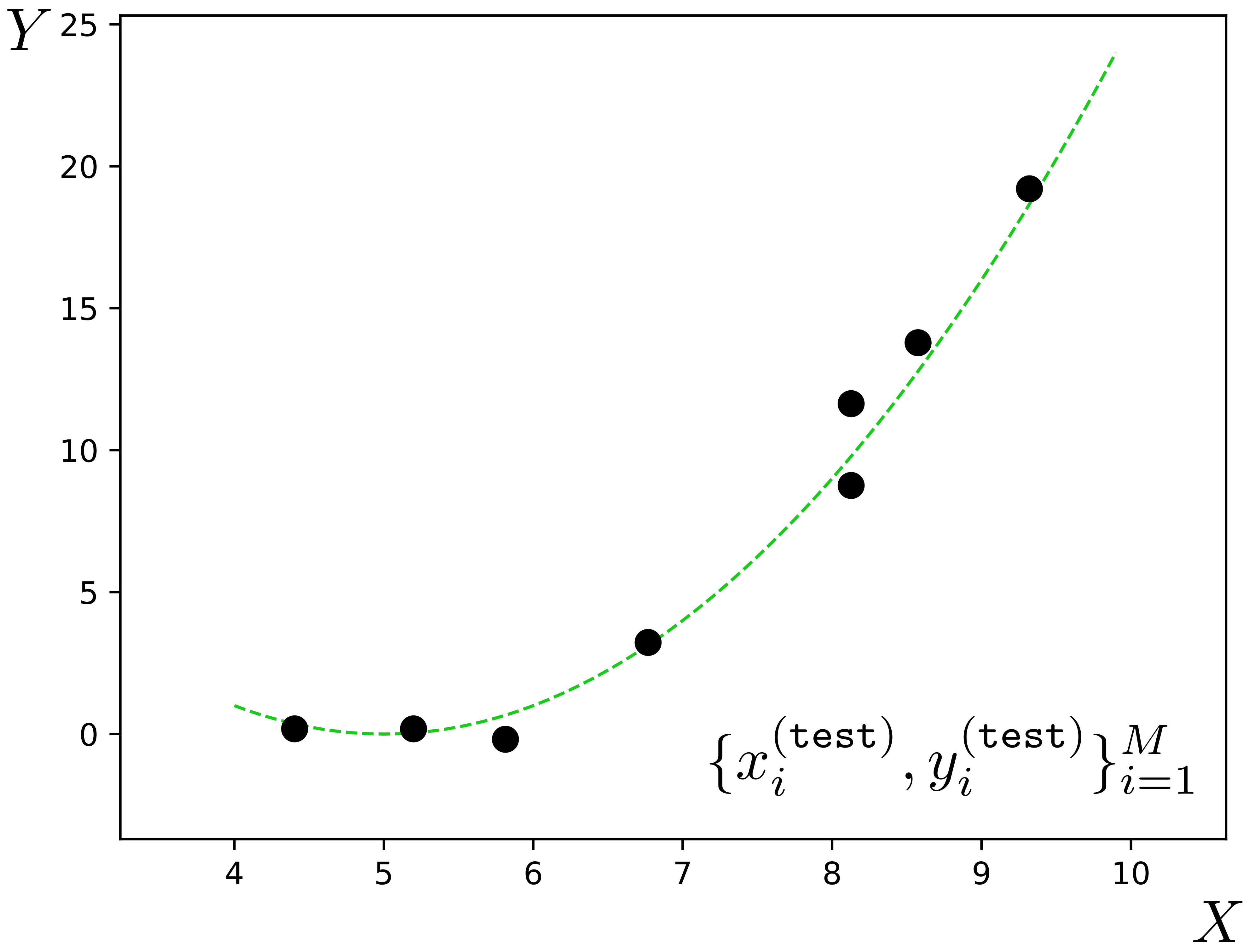
Test data



Training data



Test data



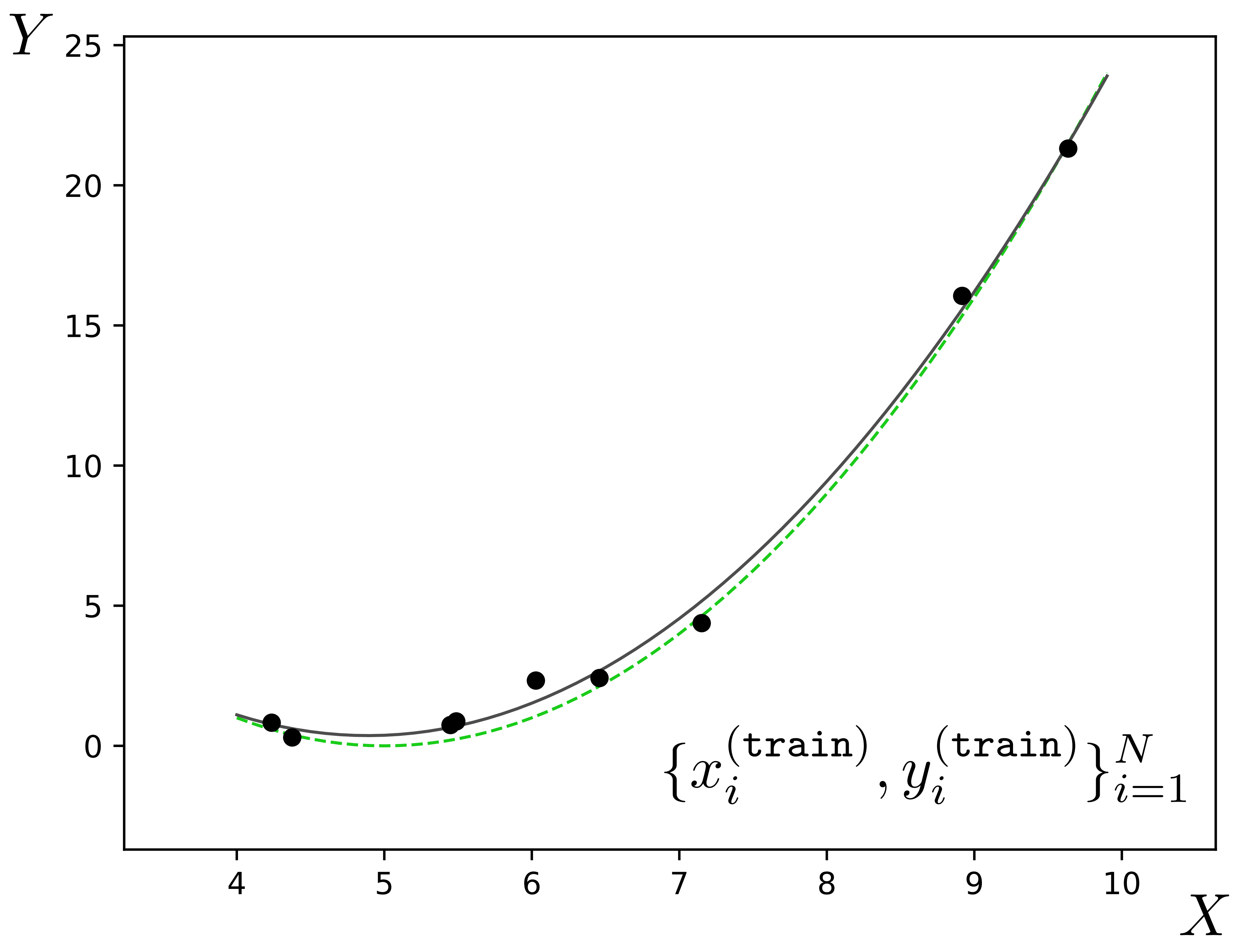
True **data-generating process**

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

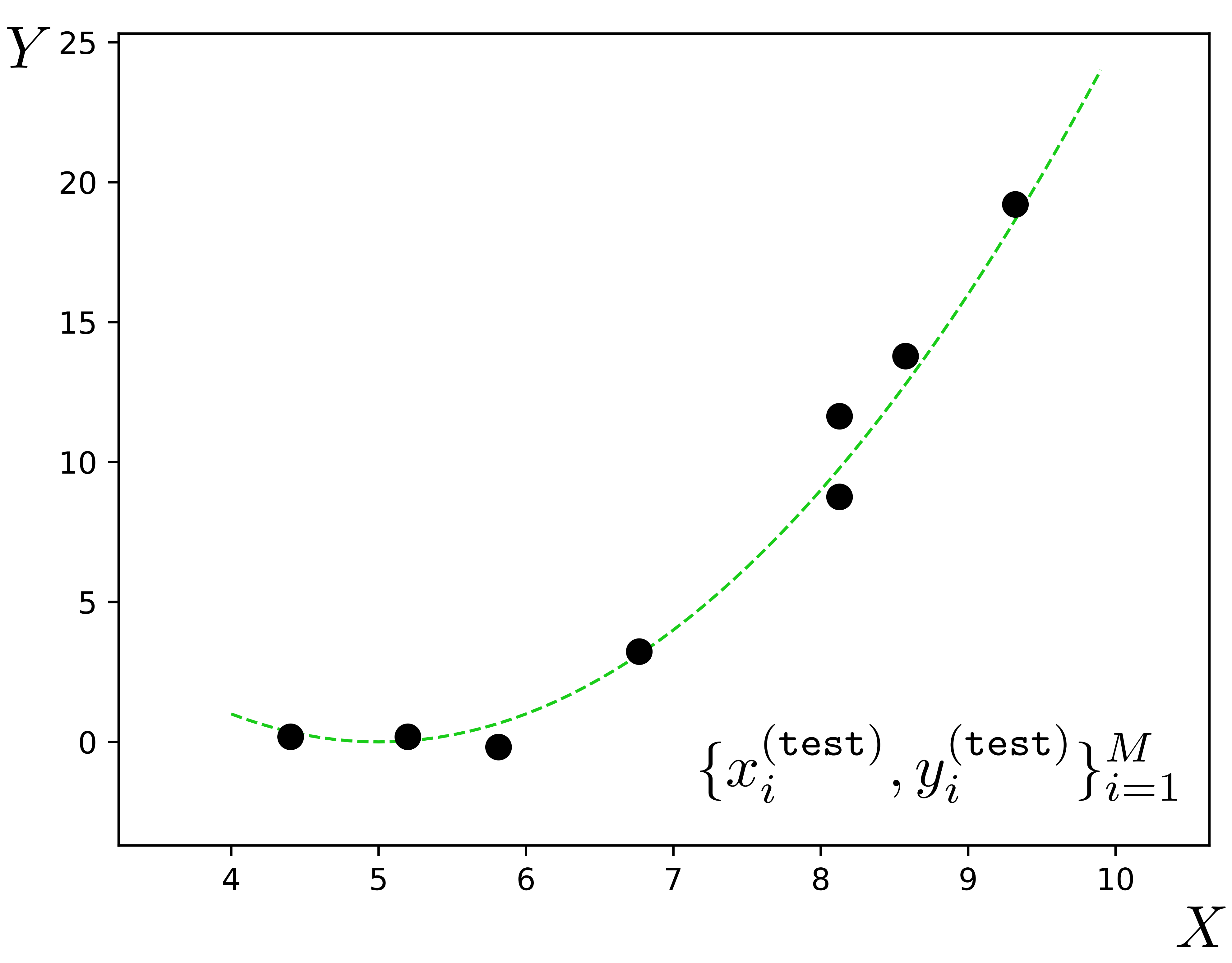
$$\{x_i^{(\mathtt{train})}, y_i^{(\mathtt{train})}\} \overset{\mathtt{iid}}{\sim} p_{\mathtt{data}}$$

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

Training data



Test data

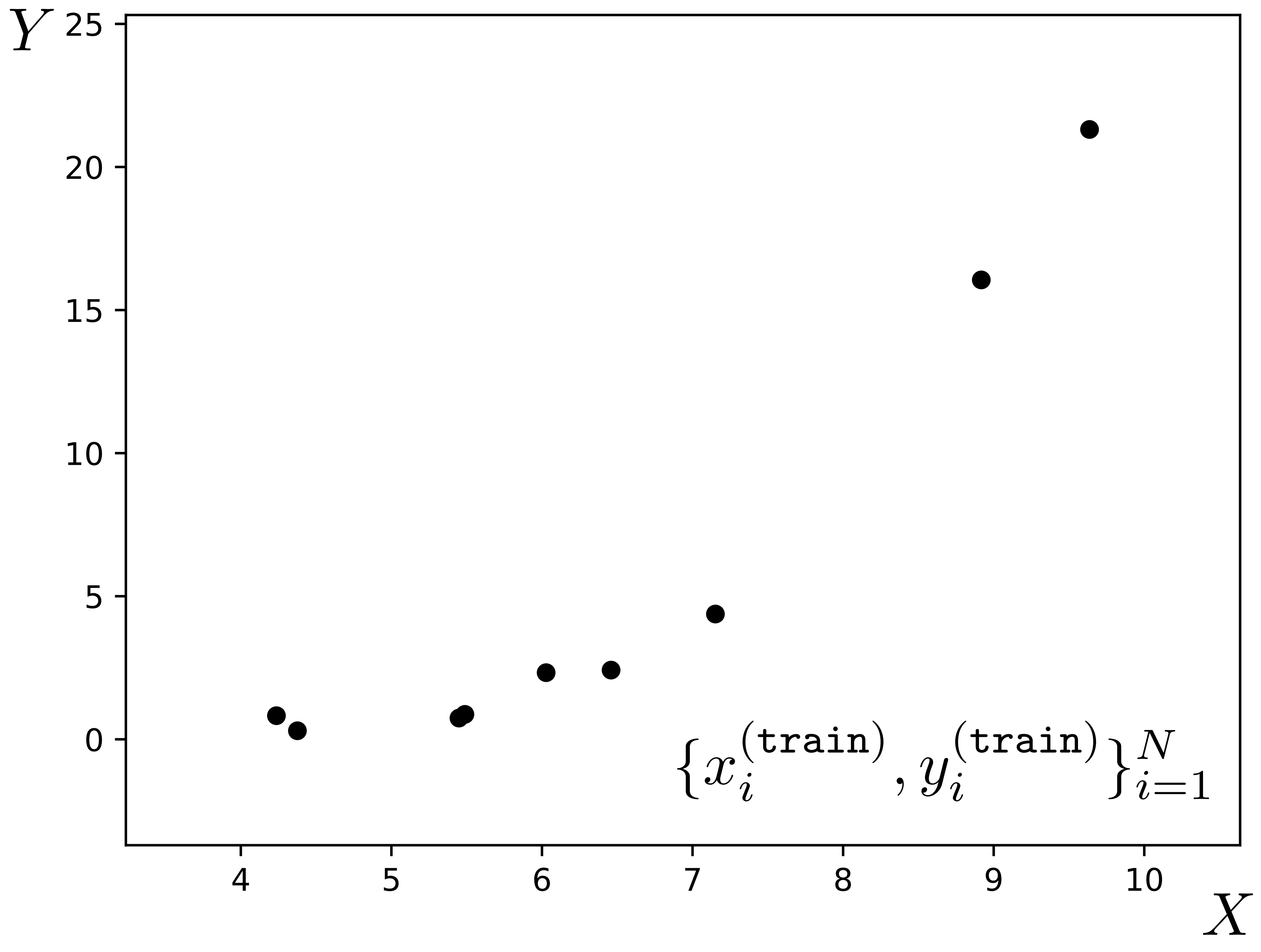


This is a huge assumption!
Almost never true in practice!

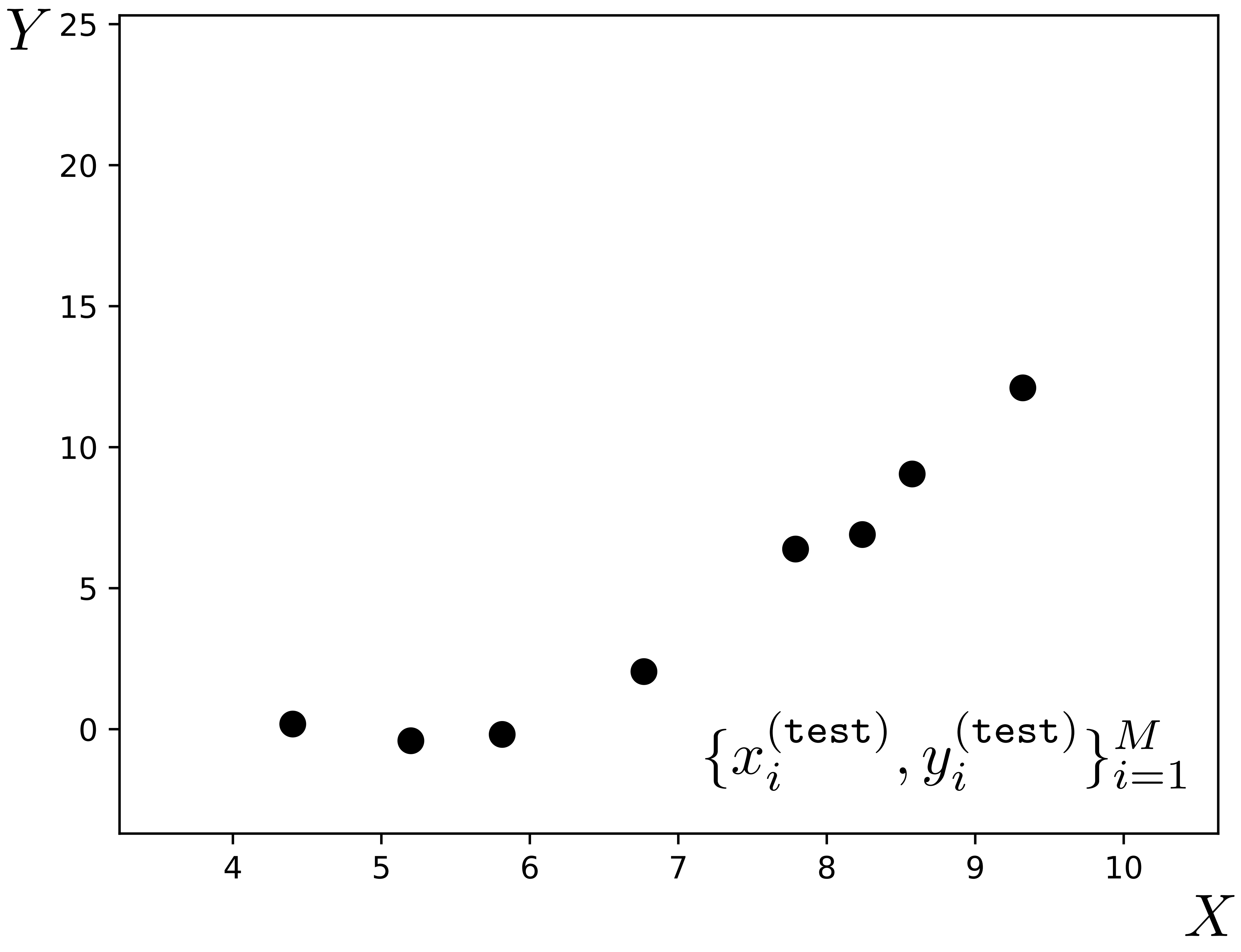
$$\{x_i^{(\mathtt{train})}, y_i^{(\mathtt{train})}\} \overset{\mathtt{iid}}{\sim} p_{\mathtt{data}}$$

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

Training data



Test data



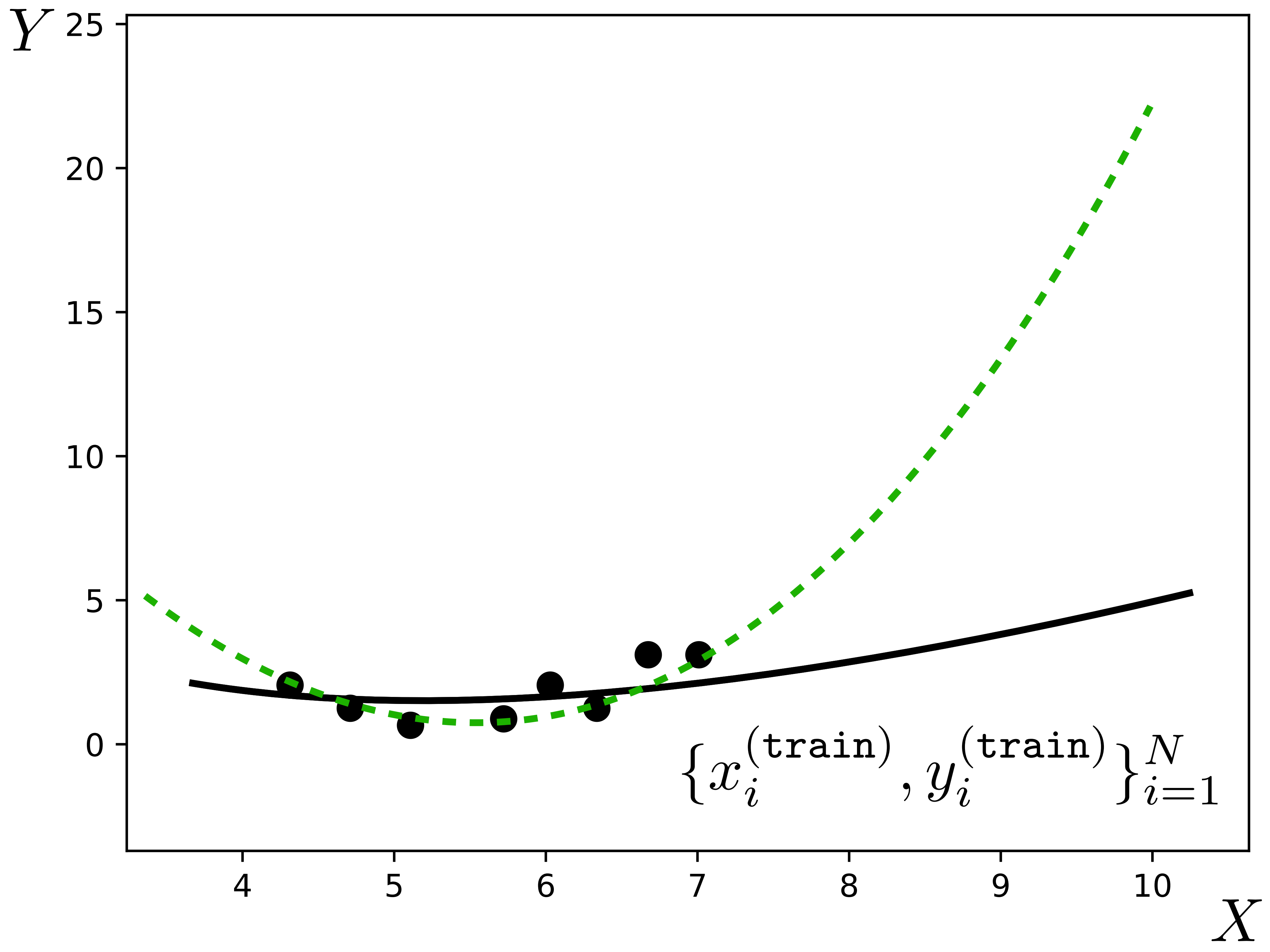
Much more commonly, we have

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

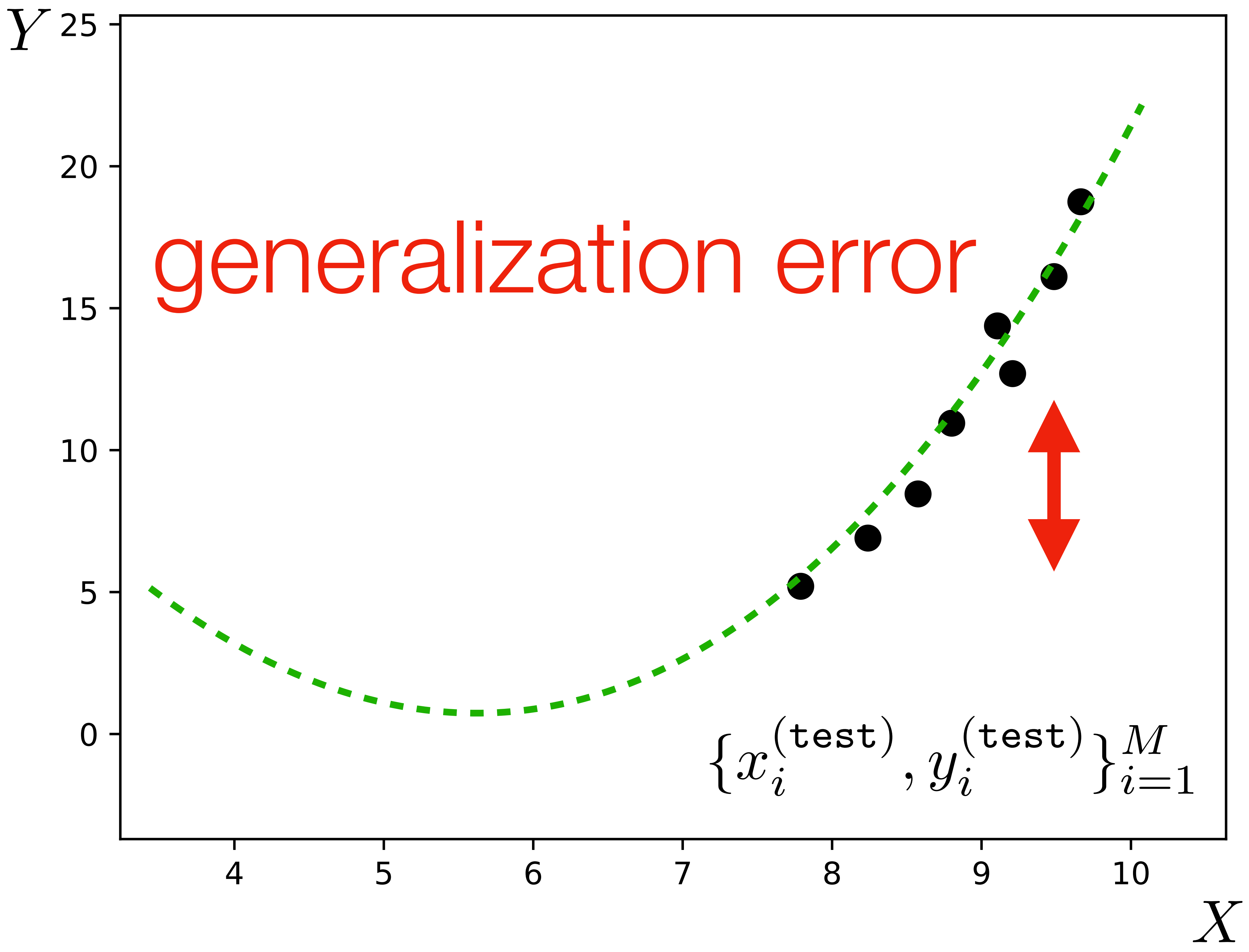
$$\{x_i^{(\mathtt{train})}, y_i^{(\mathtt{train})}\} \stackrel{\mathtt{iid}}{\sim} p_{\mathtt{train}}$$

$$\{x_i^{(\mathtt{test})}, y_i^{(\mathtt{test})}\} \stackrel{\mathtt{iid}}{\sim} p_{\mathtt{test}}$$

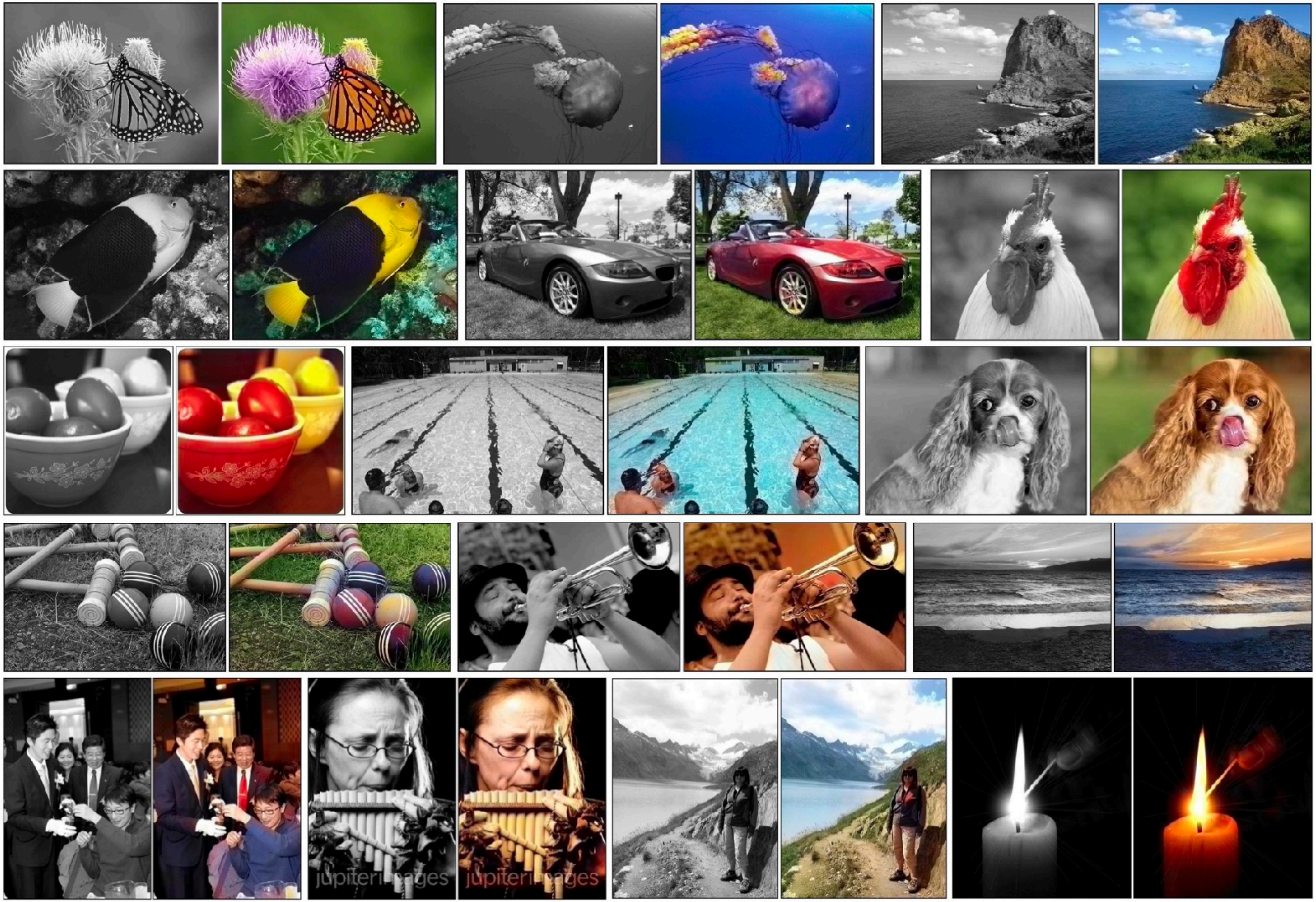
Training data



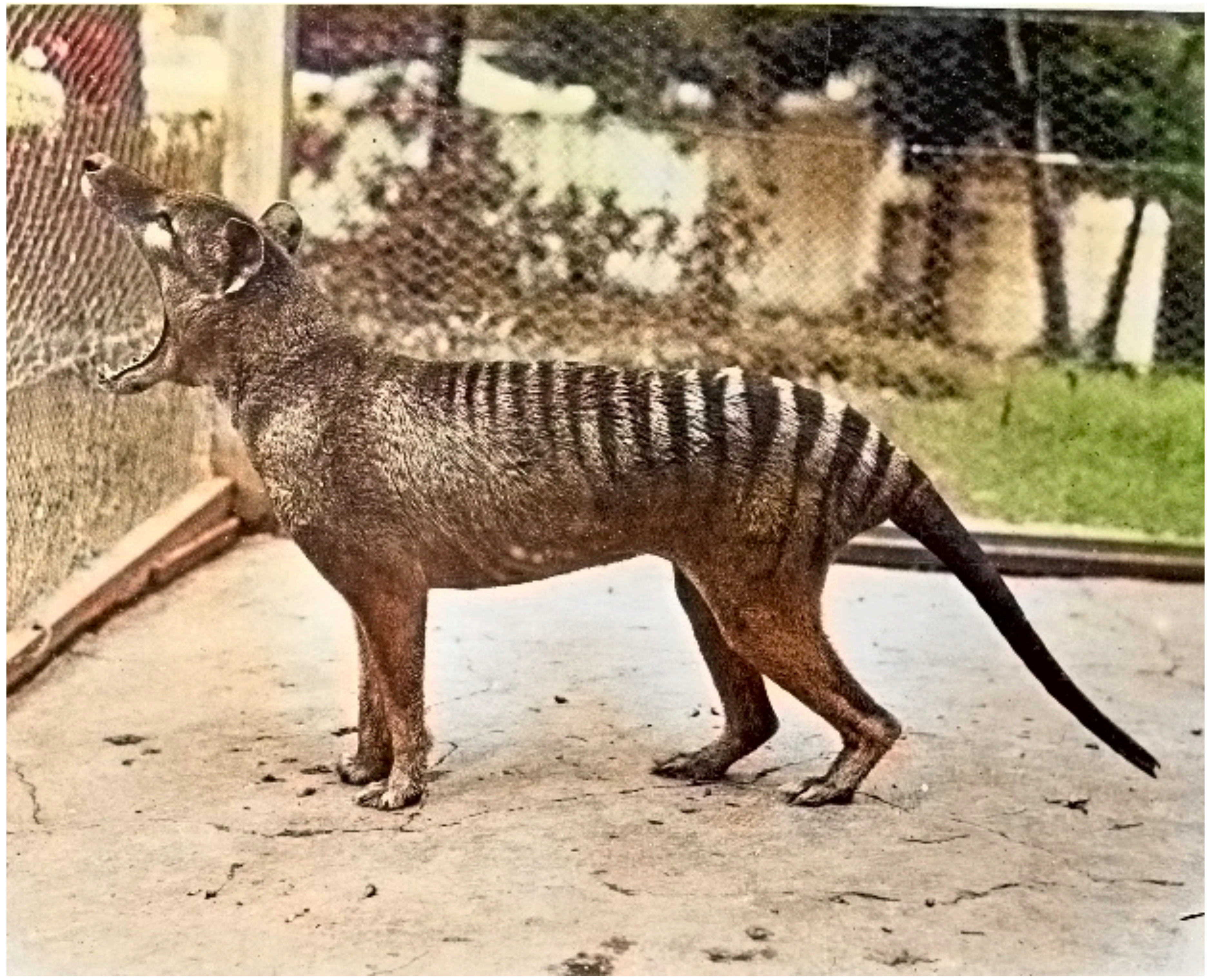
Test data



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



u/Rafael_P_S



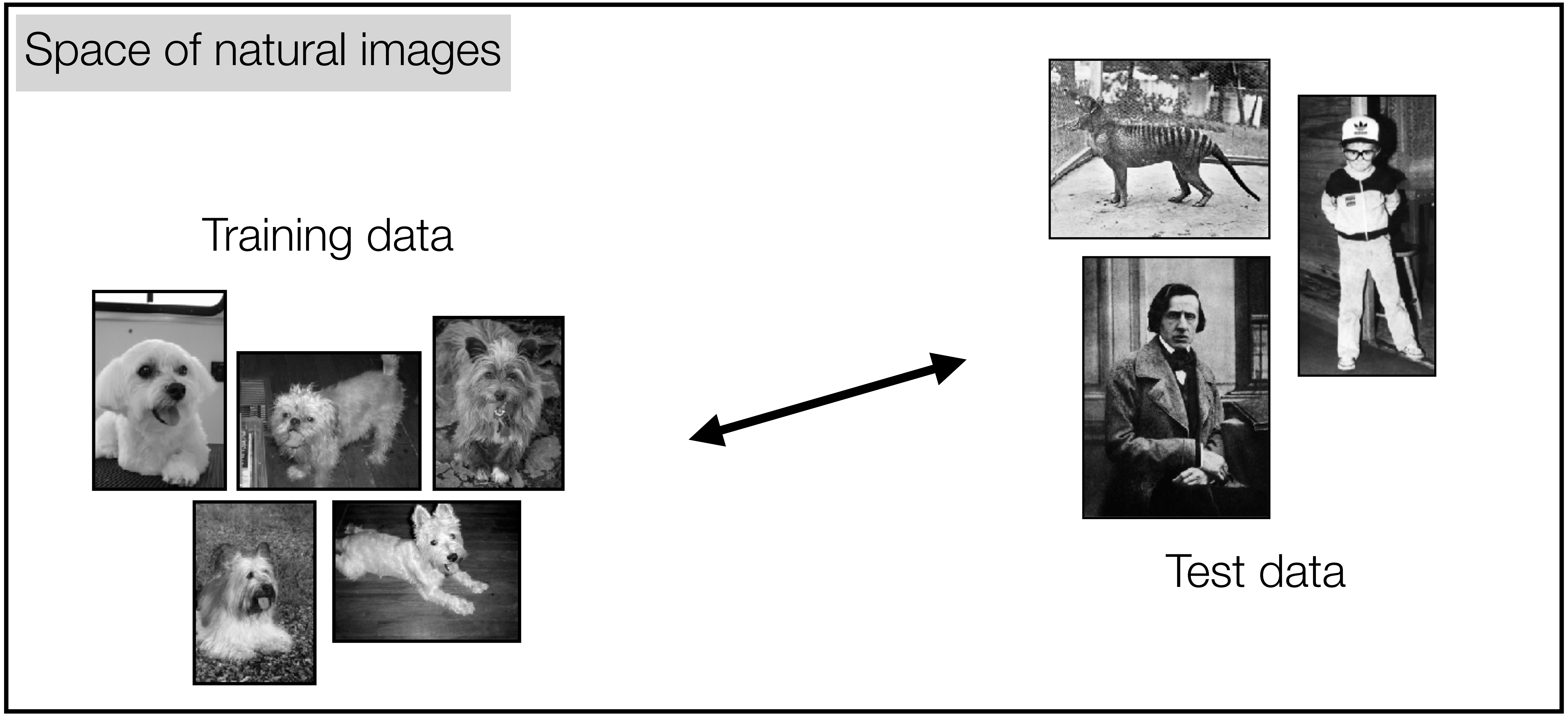
Thylacine

Chopin

training domain

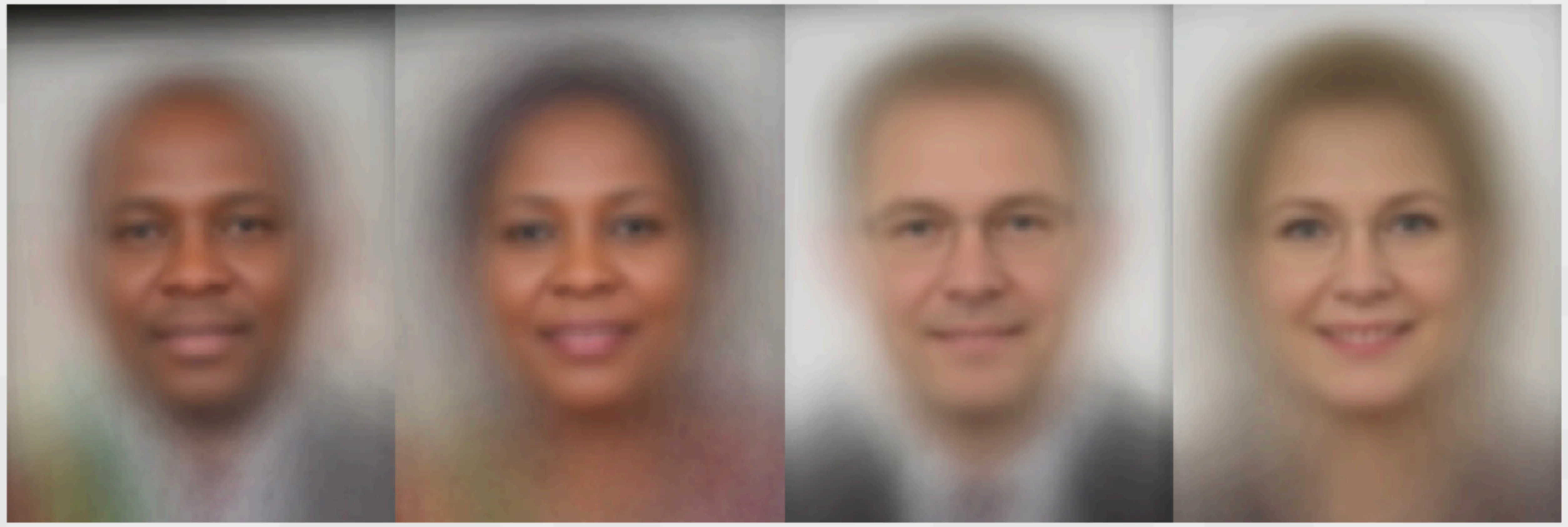
testing domain
(where we actual use our model)

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



Algorithmic Bias

Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%



http://gendershades.org/overview.html

Proceedings of Machine Learning Research 81:1–15, 2018 Conference on Fairness, Accountability, and Transparency

Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*

Joy Buolamwini JOYAB@MIT.EDU
MIT Media Lab 75 Amherst St. Cambridge, MA 02139

Timnit Gebru TIMNIT.GEBRU@MICROSOFT.COM
Microsoft Research 641 Avenue of the Americas, New York, NY 10011

Editors: Sorelle A. Friedler and Christo Wilson

Abstract

Recent studies demonstrate that machine learning algorithms can discriminate based on classes like race and gender. In this work, we present an approach to evaluate bias present in automated facial analysis algorithms and datasets with respect to phenotypic subgroups. Using the dermatologist approved Fitzpatrick Skin Type classification system, we characterize the gender and skin type distribution of two facial analysis benchmarks, IJB-A and Adience. We find that these datasets are overwhelmingly composed of lighter-skinned subjects (79.6% for IJB-A and 86.2% for Adience) and introduce a new facial analysis dataset which is balanced by gender and skin type. We evaluate 3 commercial gender classification systems using our dataset and show that darker-skinned females are the most misclassified group (with error rates of up to 34.7%). The maximum error rate for lighter-skinned males is 0.8%. The substantial disparities in the accuracy of classifying darker females, lighter females, darker males, and lighter males in gender classification systems require urgent attention if commercial companies are to build genuinely fair, transparent and accountable facial analysis algorithms.

Keywords: Computer Vision, Algorithmic Audit, Gender Classification

1. Introduction

Artificial Intelligence (AI) is rapidly infiltrating every aspect of society. From helping determine who is hired, fired, granted a loan, or how long an individual spends in prison, decisions that have traditionally been performed by humans are rapidly made by algorithms (O'Neil, 2017; Citron and Pasquale, 2014). Even AI-based technologies that are not specifically trained to perform high-stakes tasks (such as determining how long someone spends in prison) can be used in a pipeline that performs such tasks. For example, while face recognition software by itself should not be trained to determine the fate of an individual in the criminal justice system, it is very likely that such software is used to identify suspects. Thus, an error in the output of a face recognition algorithm used as input for other tasks can have serious consequences. For example, someone could be wrongfully accused of a crime based on erroneous but confident misidentification of the perpetrator from security video footage analysis.

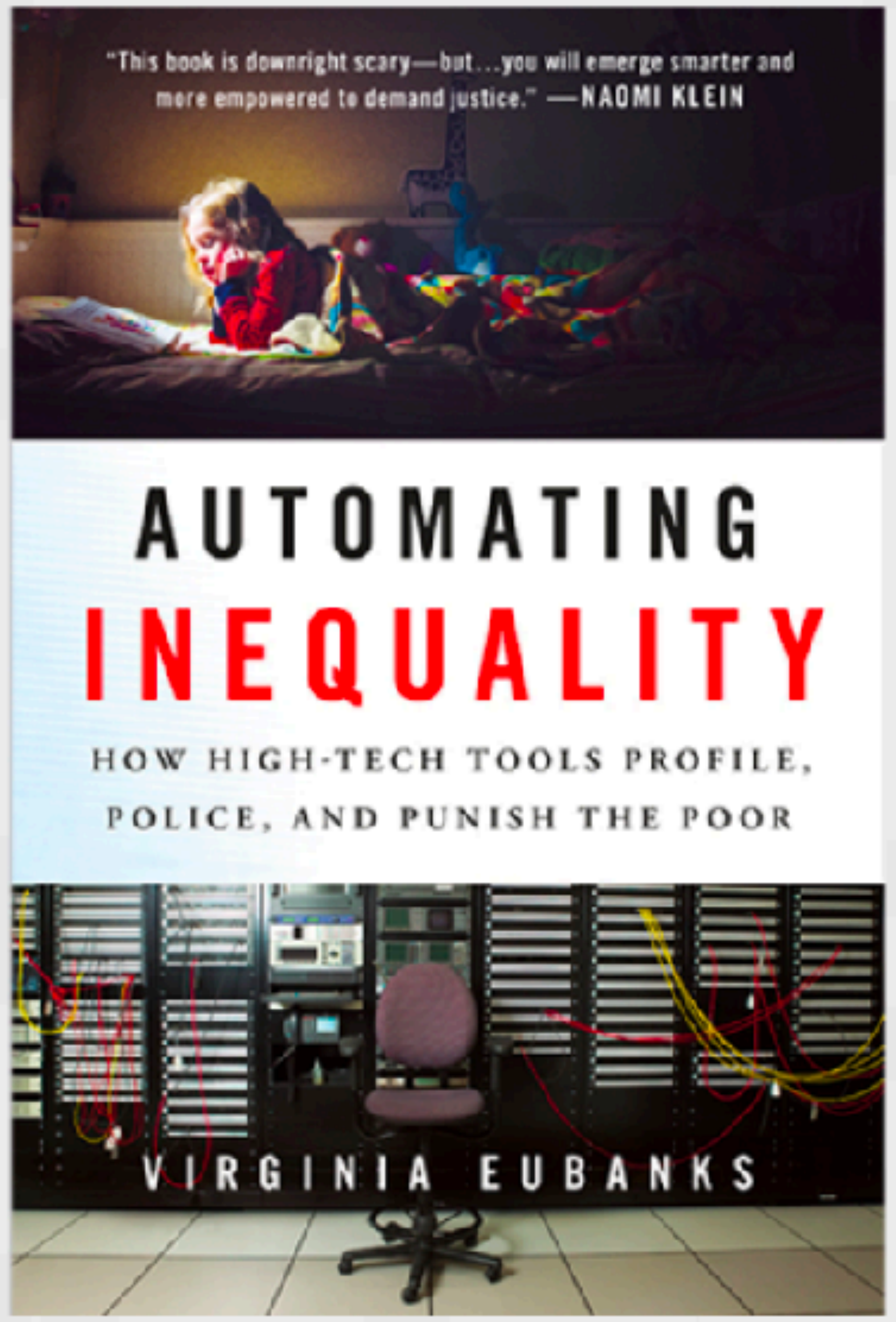
Many AI systems, e.g. face recognition tools, rely on machine learning algorithms that are trained with labeled data. It has recently been shown that algorithms trained with biased data have resulted in algorithmic discrimination (Bolukbasi et al., 2016; Caliskan et al., 2017). Bolukbasi et al. even showed that the popular word embedding space, Word2Vec, encodes societal gender biases. The authors used Word2Vec to train an analogy generator that fills in missing words in analogies. The analogy man is to computer programmer as woman is to "X" was completed with "homemaker", conforming to the stereotype that programming is associated with men and homemaking with women. The biases in Word2Vec are thus likely to be propagated throughout any system that uses this embedding.

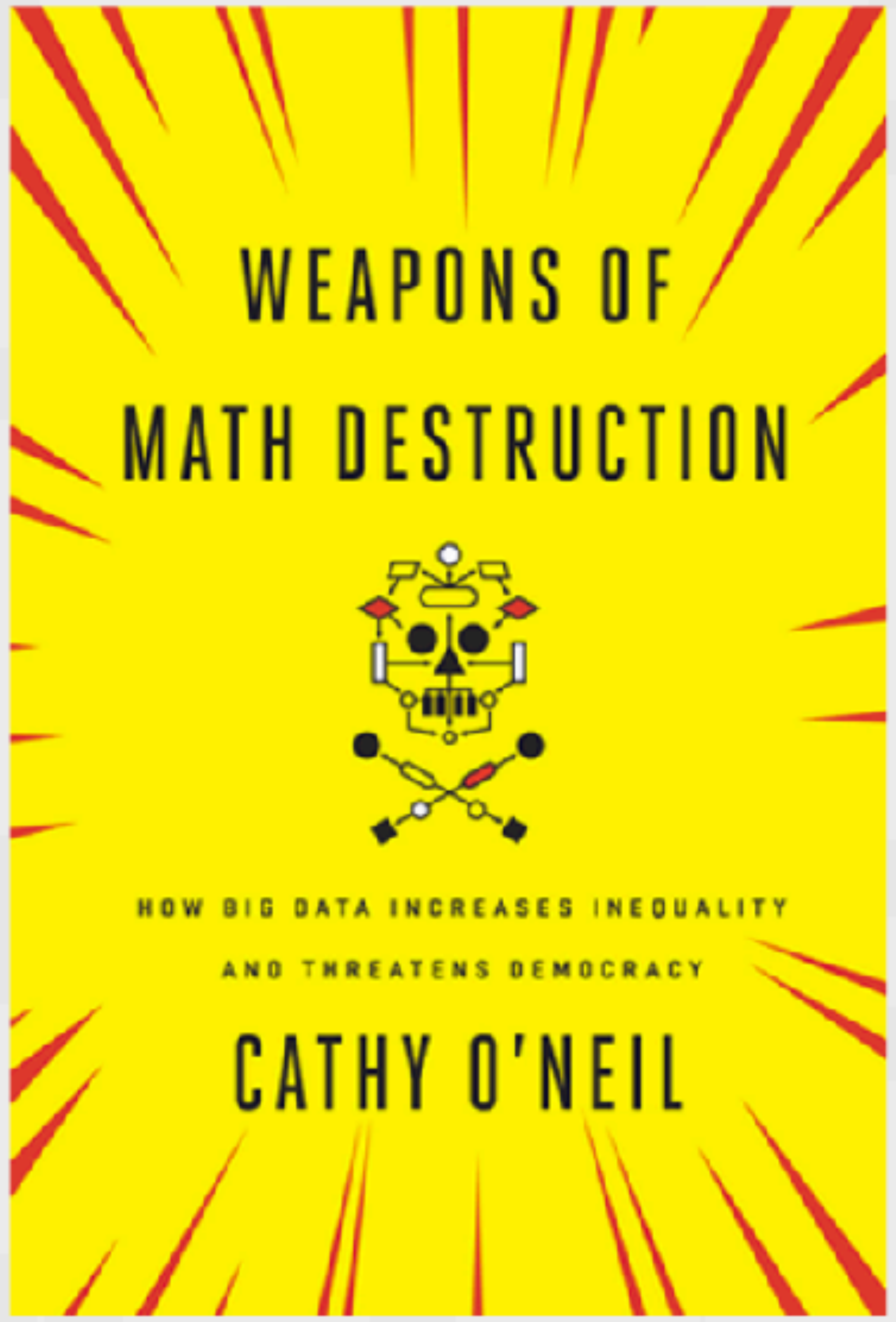
* Download our gender and skin type balanced PPB dataset at gendershades.org

http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf

While this study focused on gender classification, the machine learning techniques used to determine gender are also broadly applied to many other areas of facial analysis and automation. Face recognition technology that has not been publicly tested for demographic accuracy is increasingly used by law enforcement and at airports. AI fueled automation now helps determine who is fired, hired, promoted, granted a loan or insurance, and even how long someone spends in prison.

For interested readers, authors Cathy O'Neil and Virginia Eubanks explore the real-world impact of algorithmic bias.





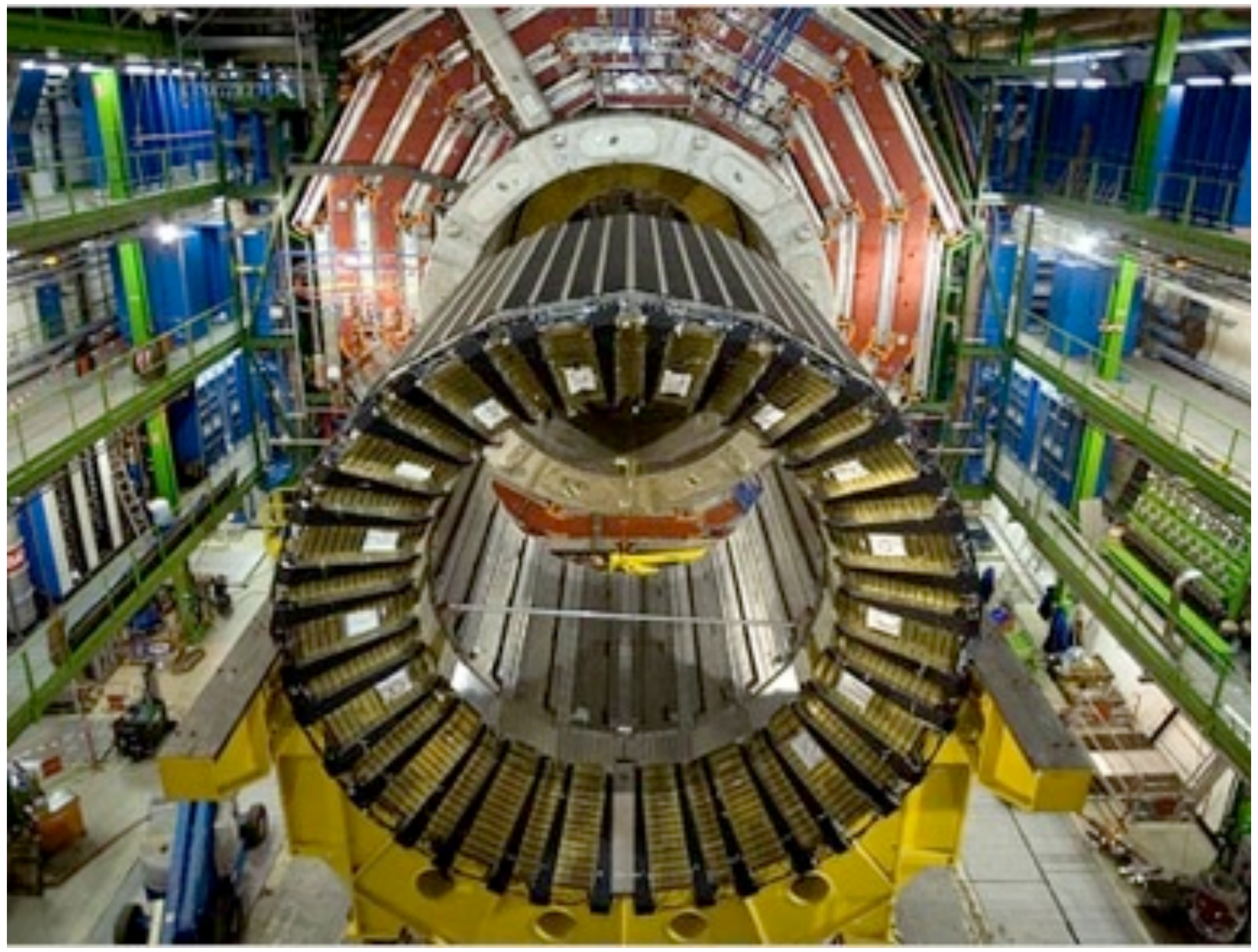
http://gendershades.org/overview.html

How can we collect good data?

+ Correctly labeled
+ Unbiased (good coverage of all relevant kinds of data)

Crowdsourcing

The value of data



The Large Hadron Collider

$ 10^{10}

Amazon Mechanical Turk

$ 10^{2} - 10^{4}

But can humans collect good data?

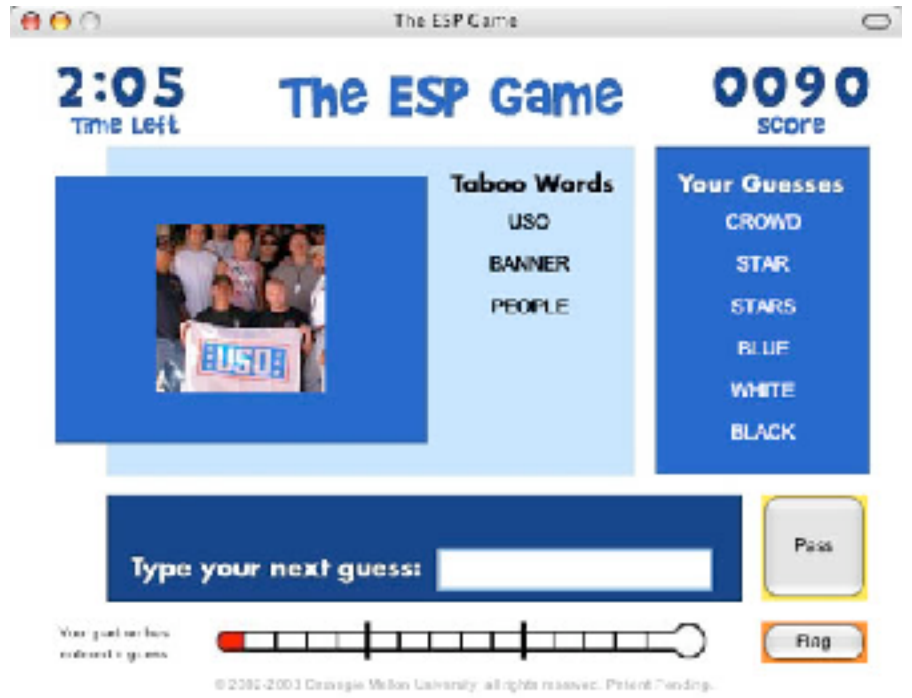

Google
mug
I OWN YOU.
Mom

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
beta
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

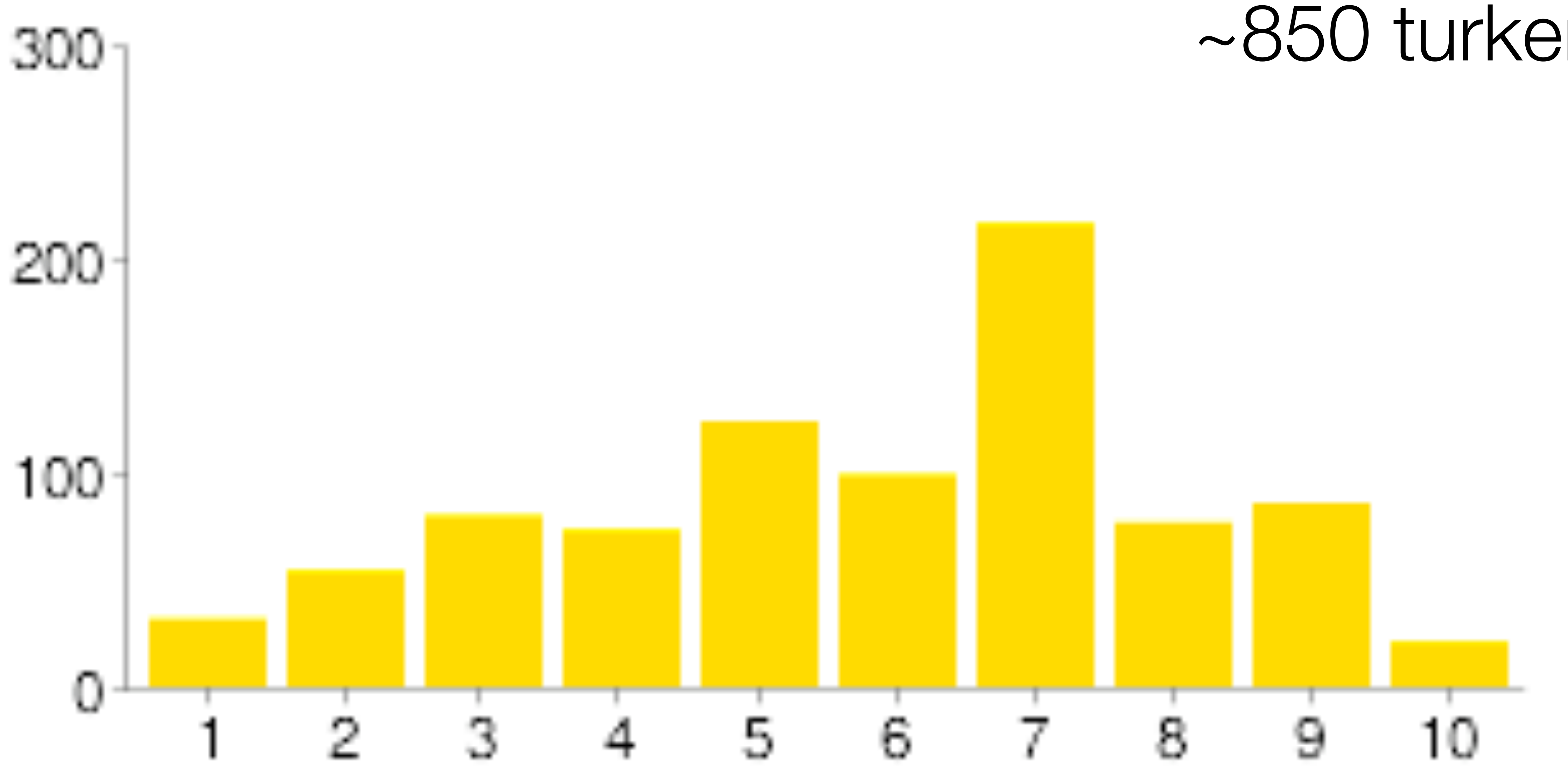
- What do you know about them?
- Will they do the work you pay for?

Let’s check a few simple experiments

People have biases…

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

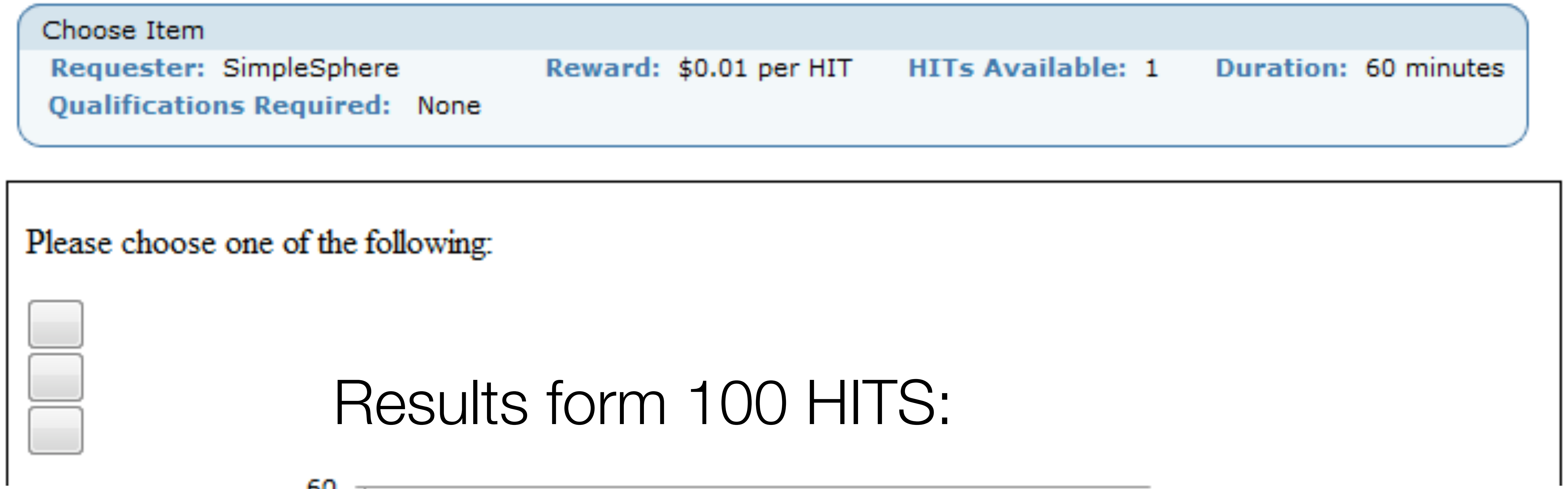
~850 turkers



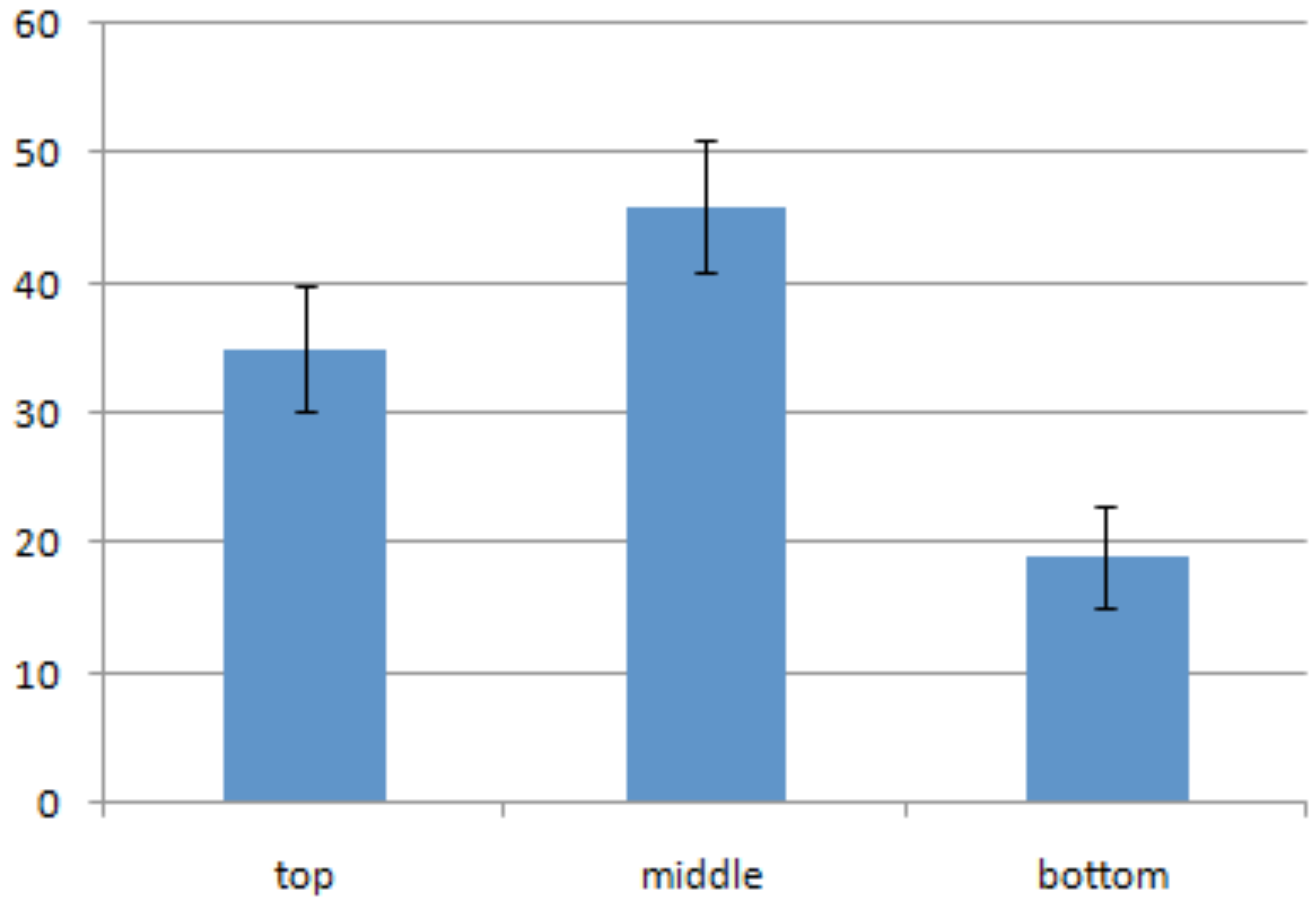
Experiment by Greg Little

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

Do humans have consistent biases?



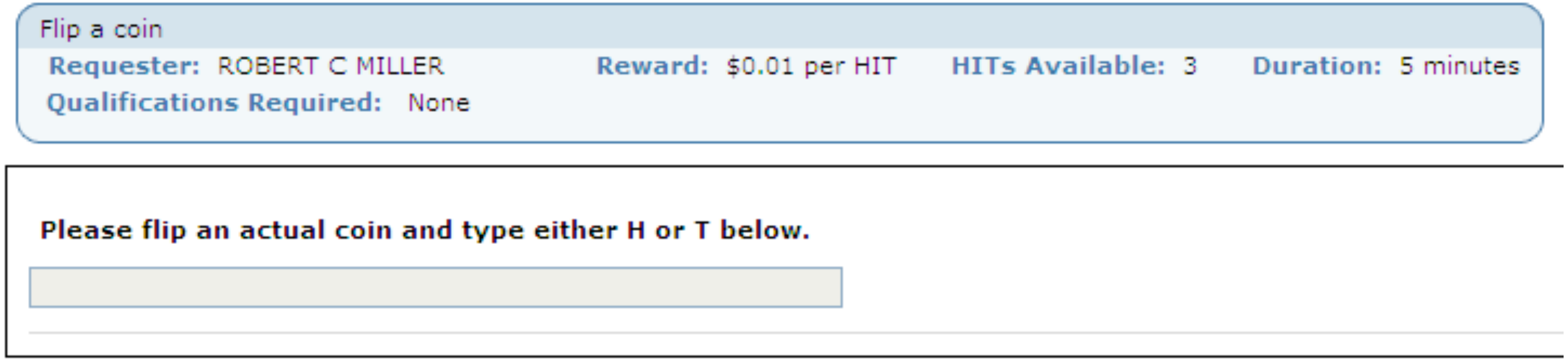
Results form 100 HITS:



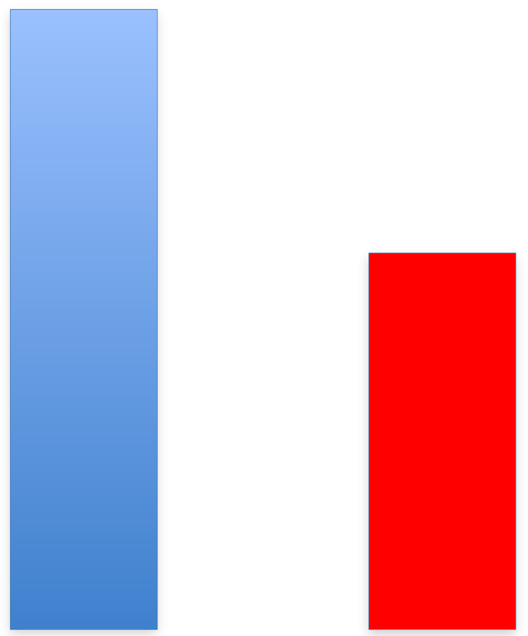
Experiment by Greg Little

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

Do humans do what you ask for?

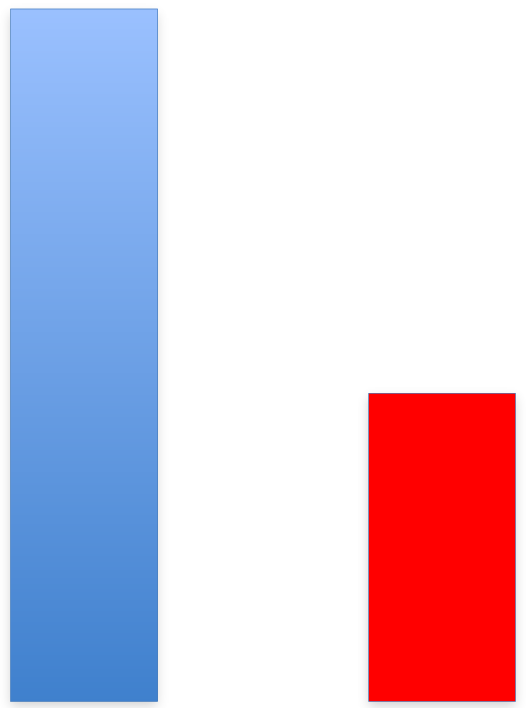


After 50 HITS:



31 heads, 19 tails

And 50 more:

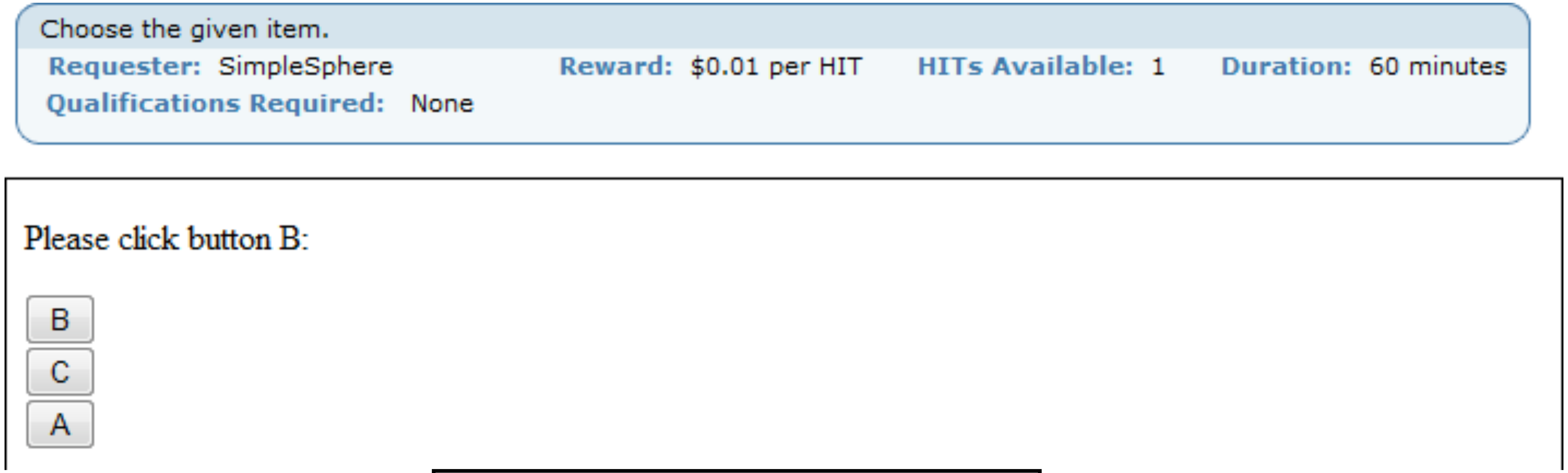


34 heads, 16 tails

Experiment by Rob Miller

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

Are humans reliable even in simple tasks?



Results of 100 HITS:
A: 2
B: 96
C: 2

Experiment by Greg Little

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

So we can sometimes collect good training data.

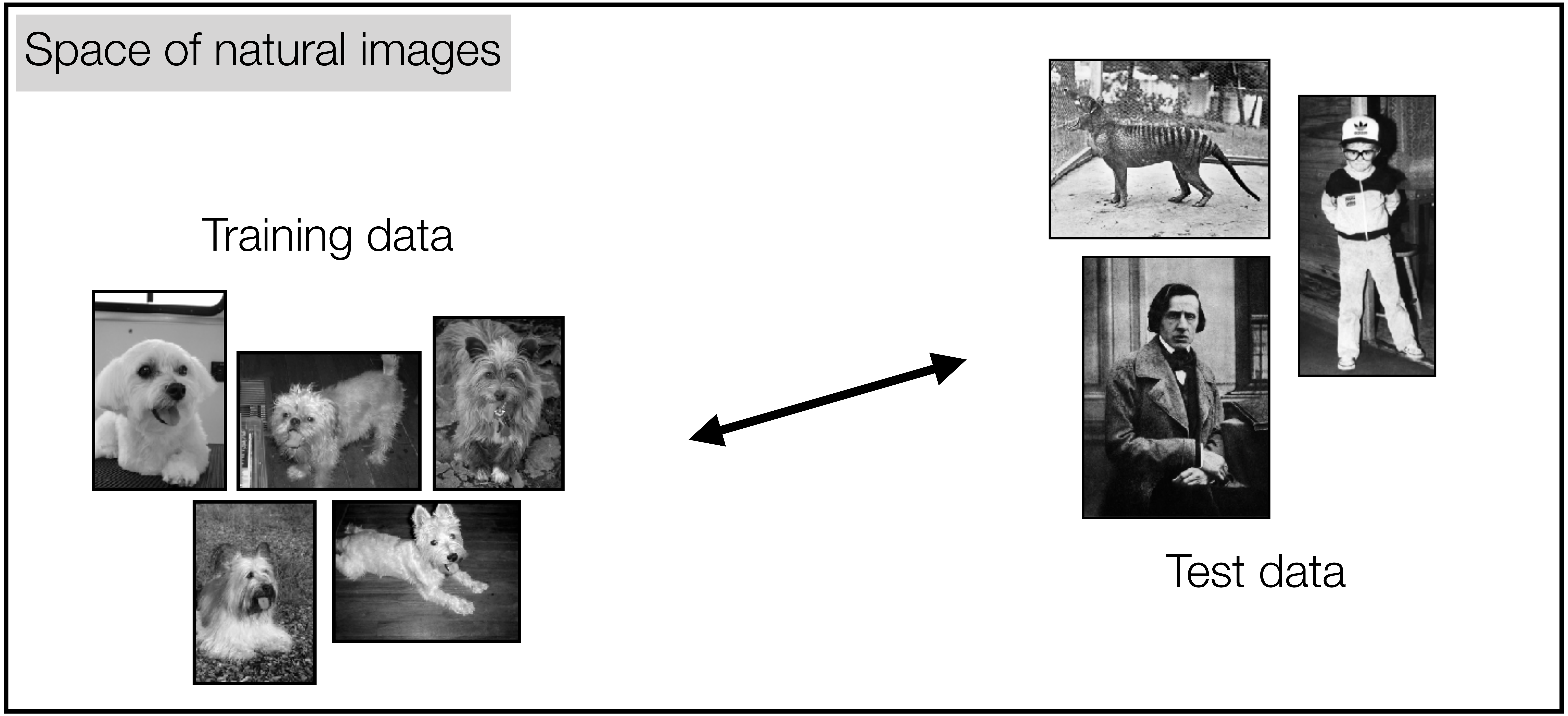
But suppose we messed up. Our test setting doesn’t look like the training data!

How can we bridge the domain gap?

training domain

testing domain
(where we actual use our model)

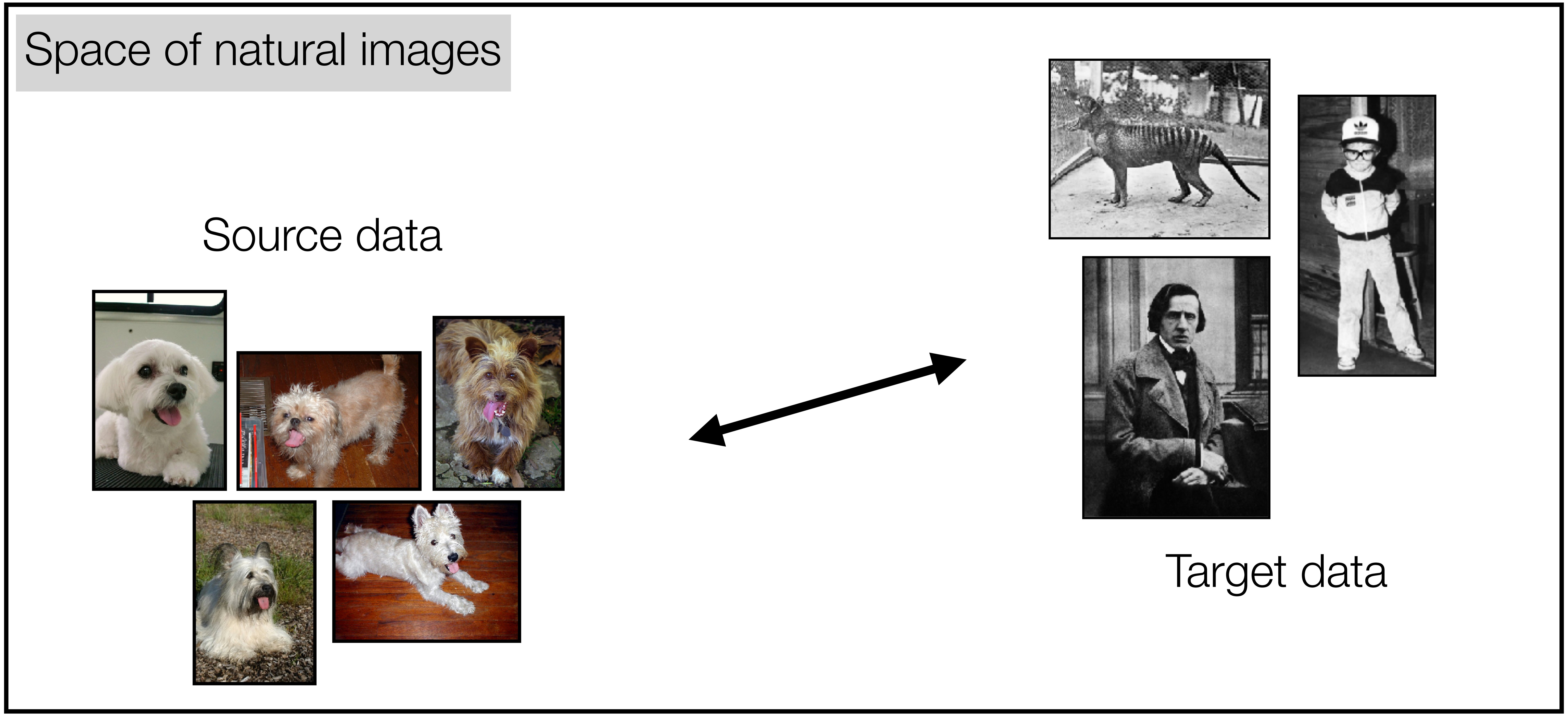
Domain gap between $p_{\texttt{train}}$ and $p_{\texttt{test}}$ will cause us to fail to generalize.



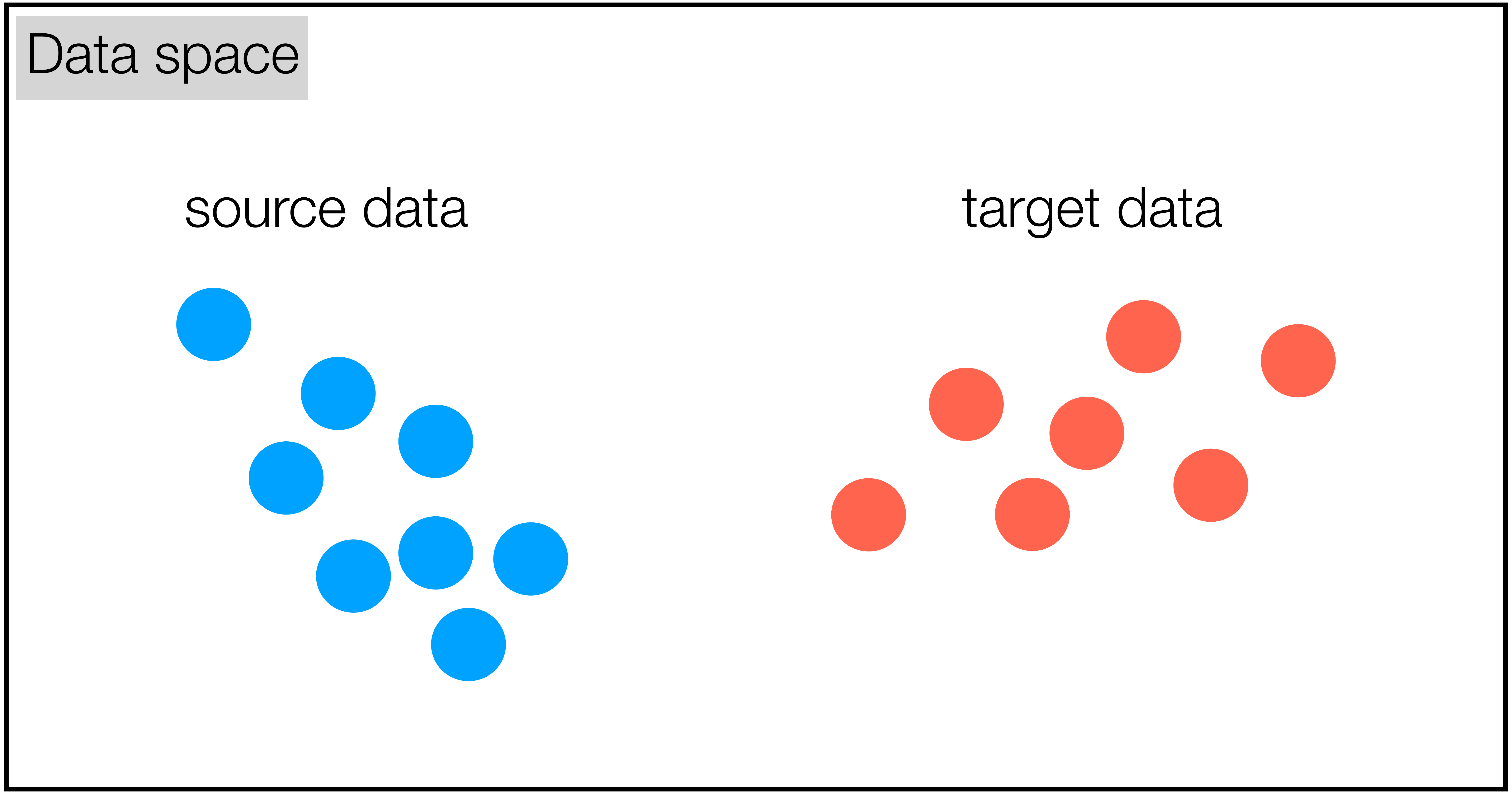
source domain

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

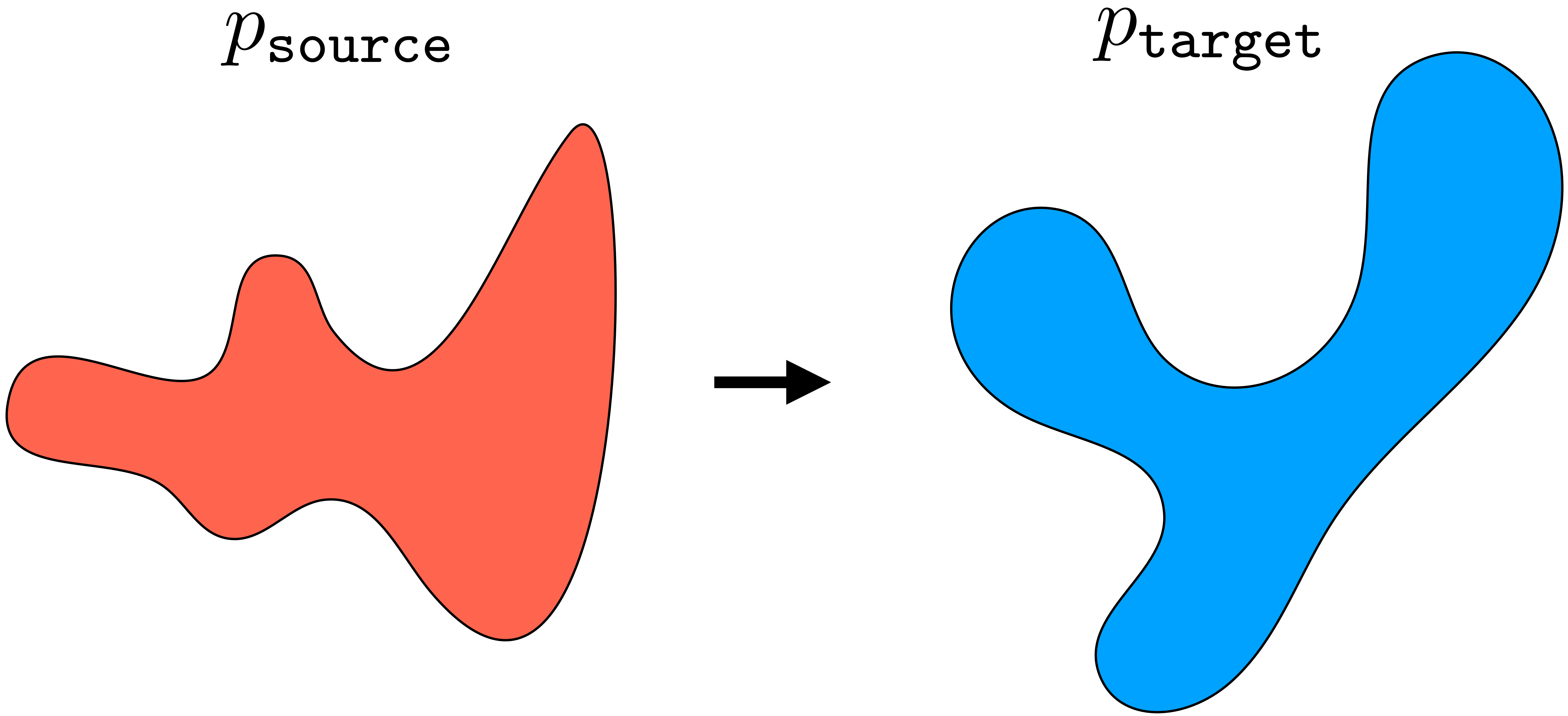


(Or vice versa)

This is called **domain adaptation**

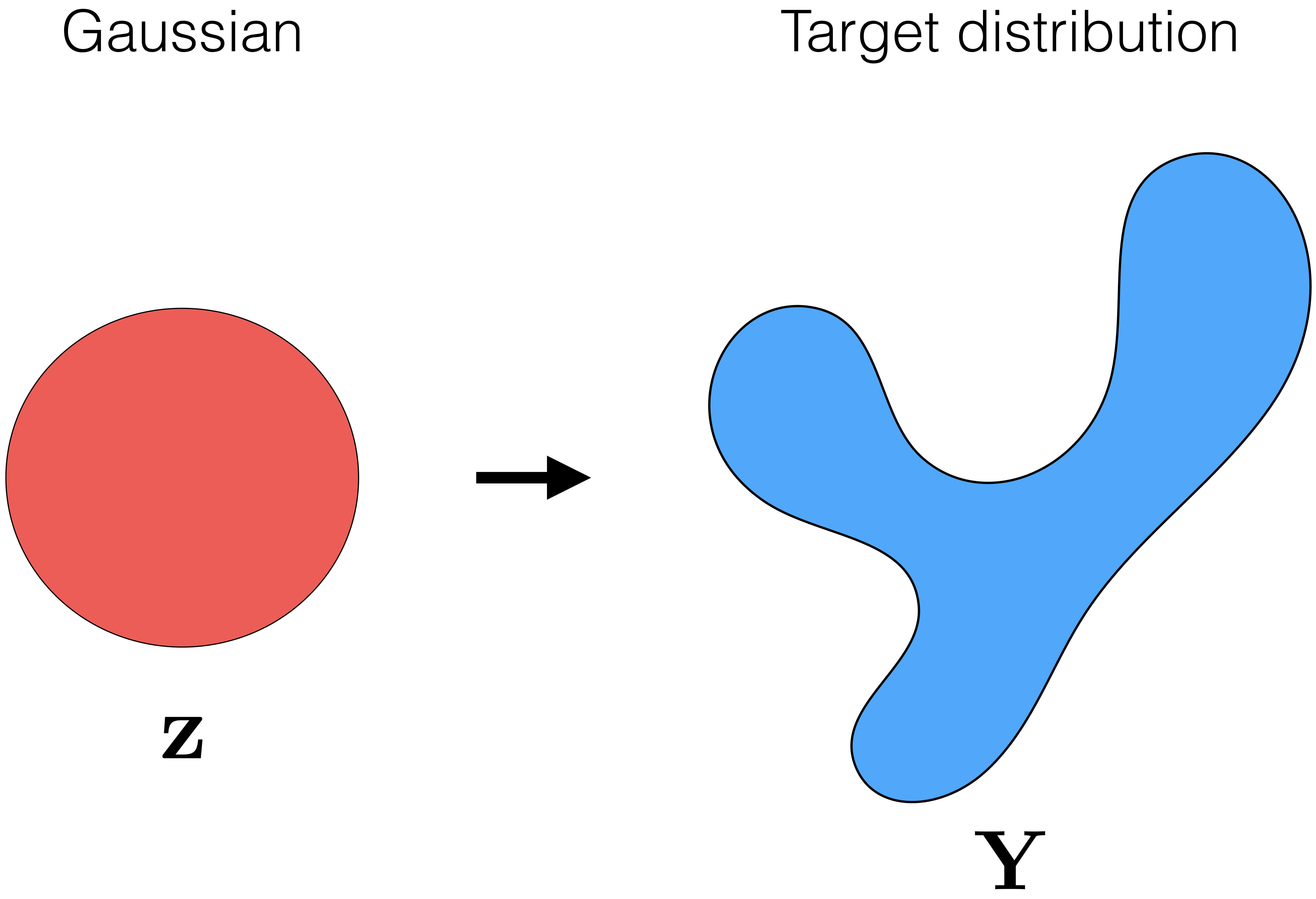
Domain adaptation

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

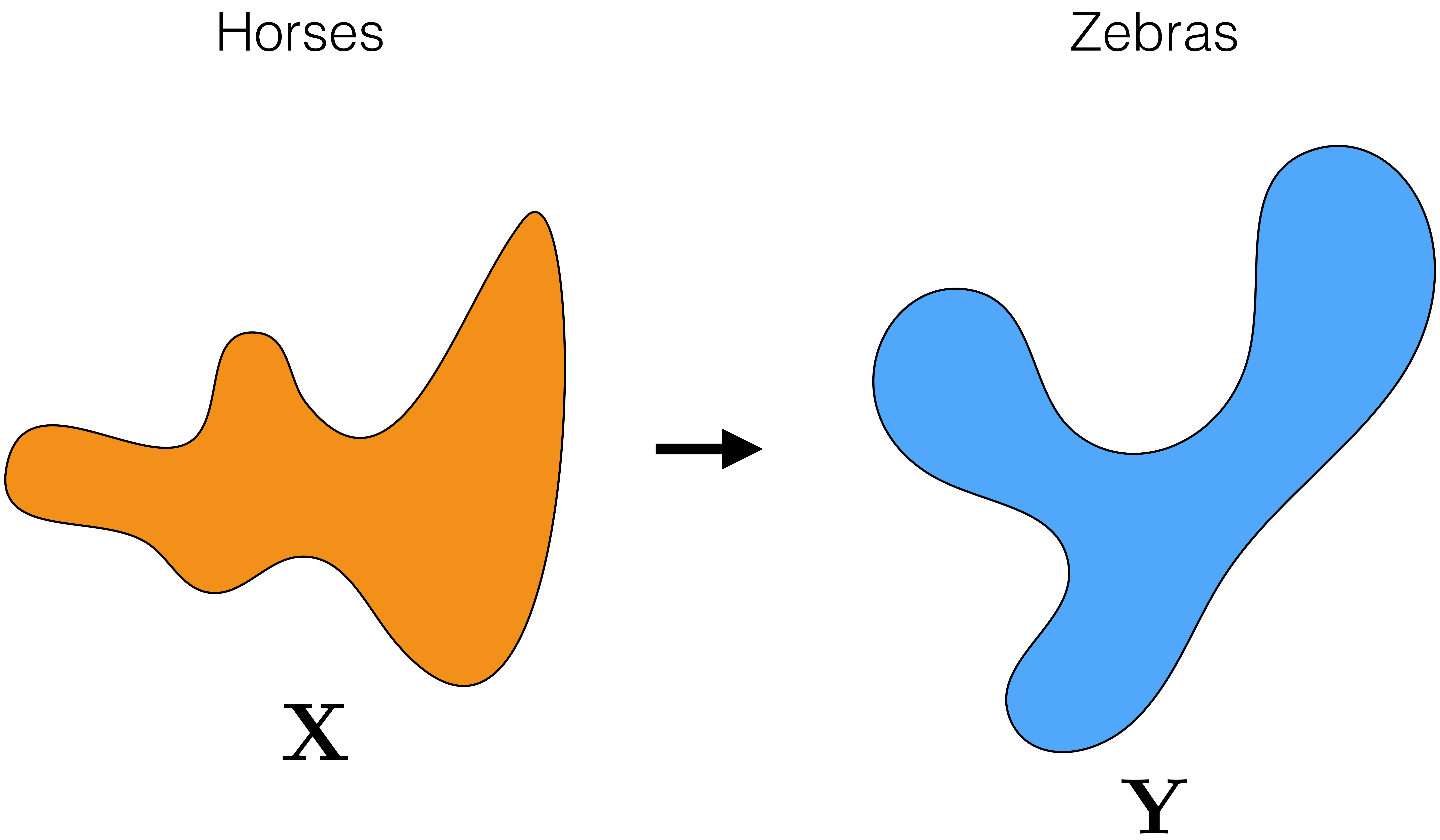


It's a just another distribution mapping problem!

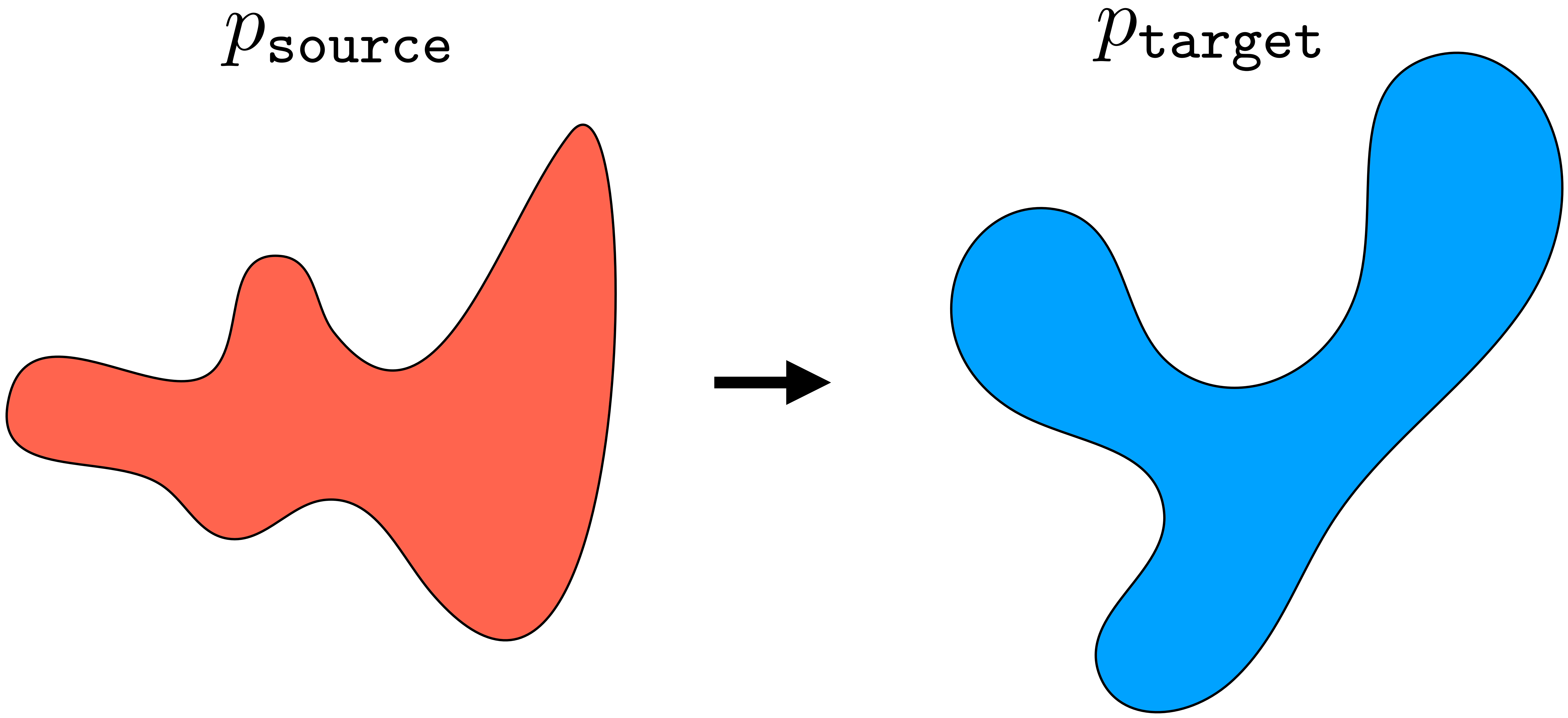
GANs



CycleGAN



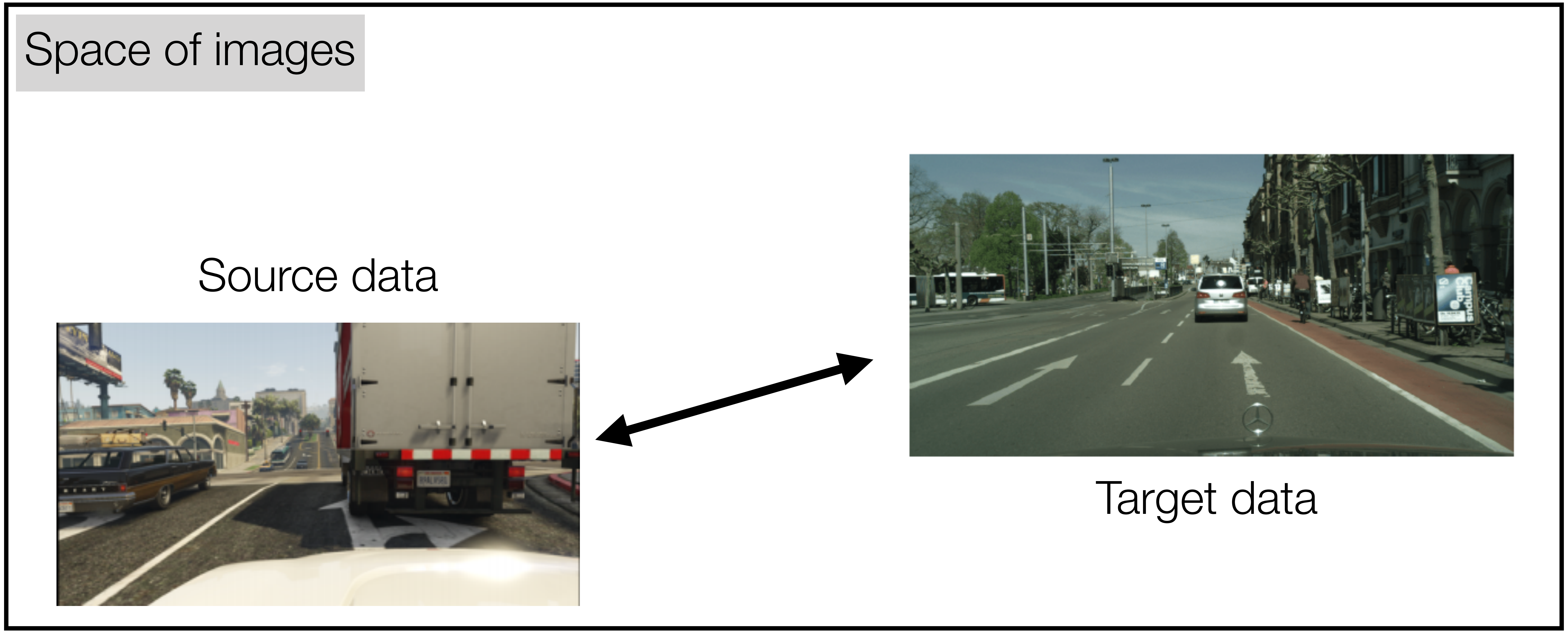
Domain adaptation



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.



CyCADA: Cycle-Consistent Adversarial Domain Adaptation

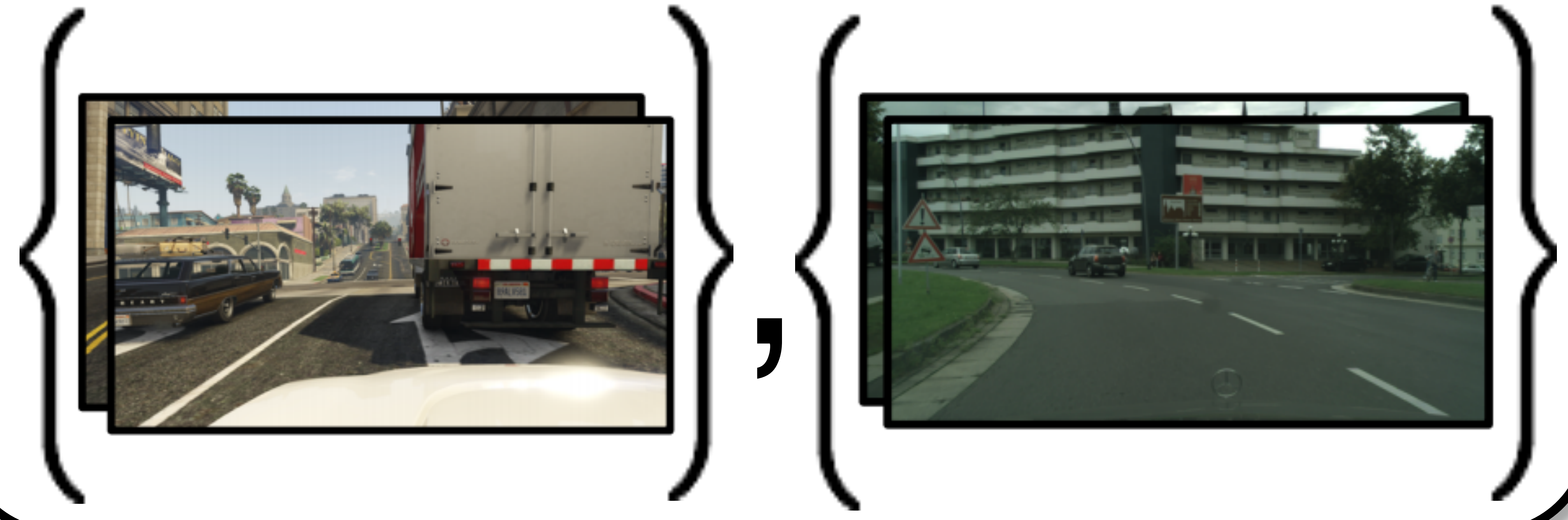
Source domain

Target domain

[Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Darrell, Efros, arXiv 2017]

CycleGAN

Training data

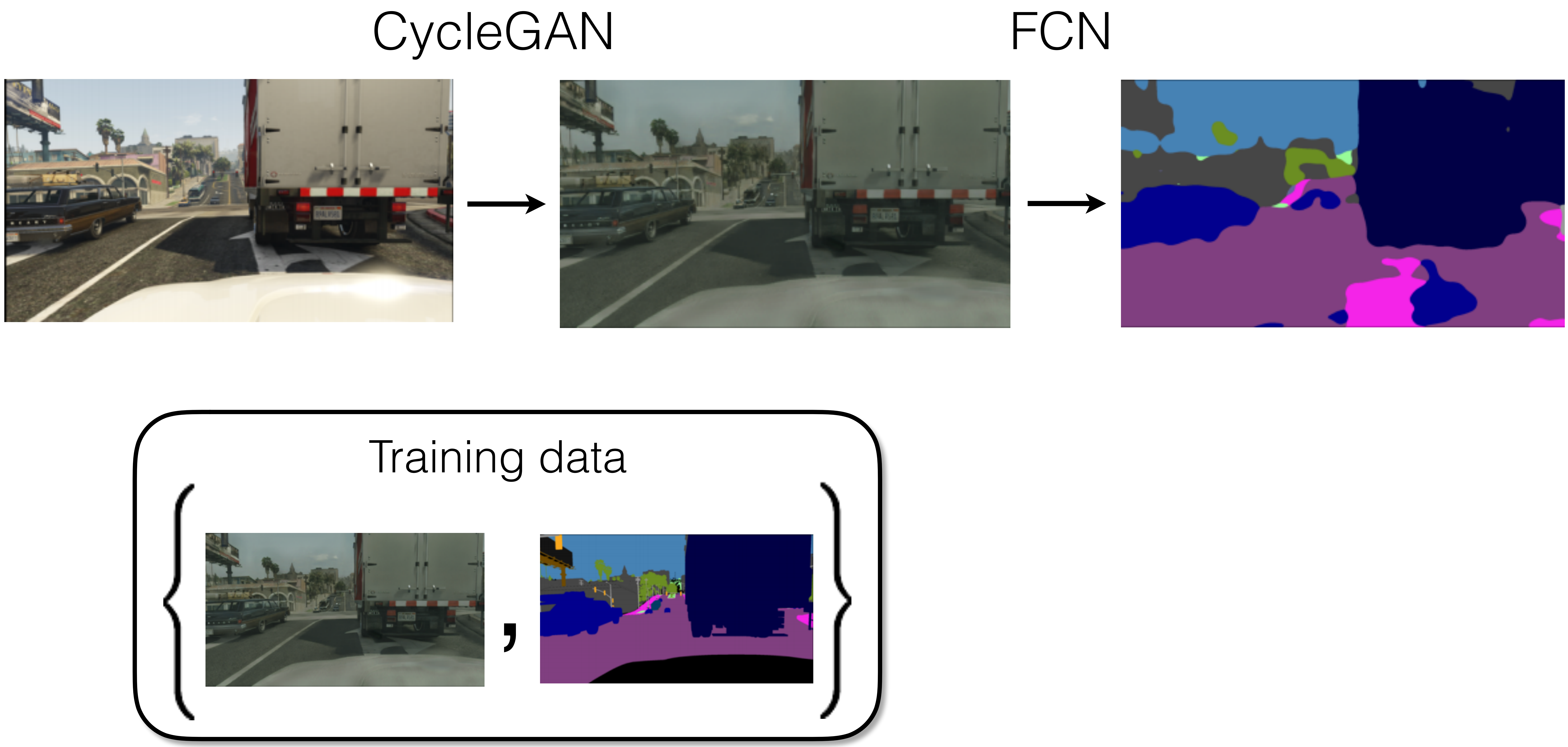


CycleGAN

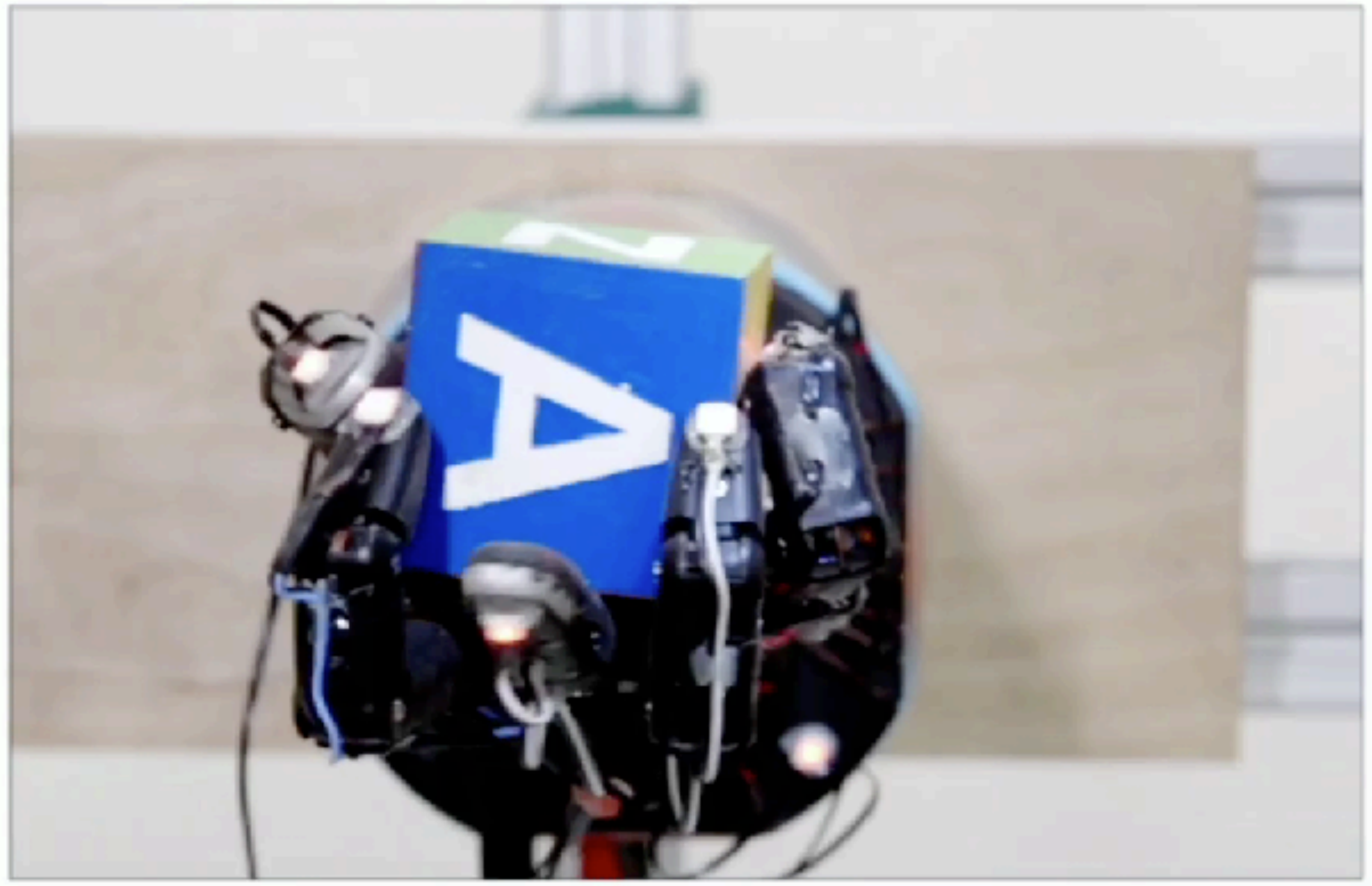
 → 

Training data

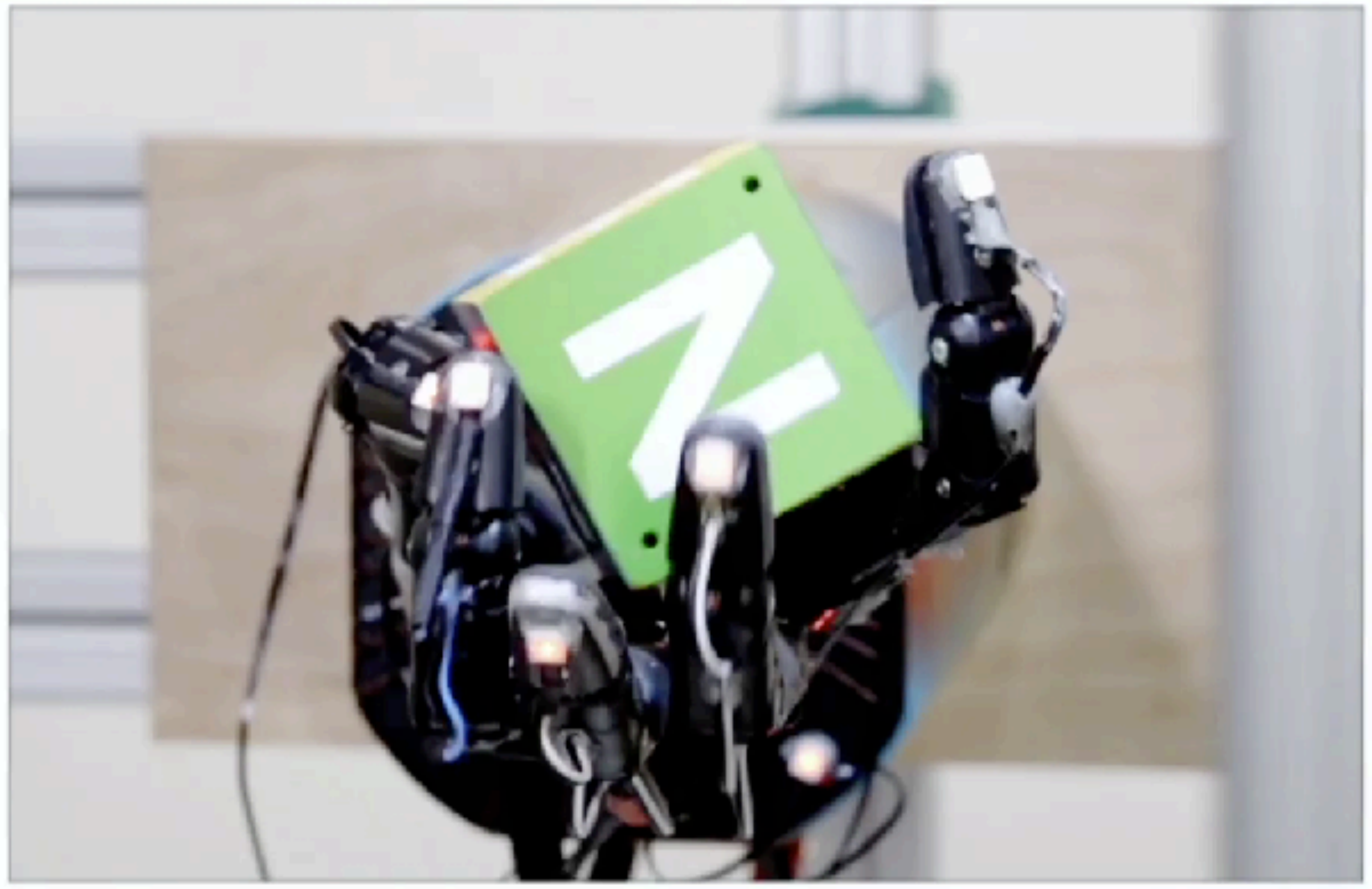
{  ,  }


CycleGAN
FCN
Training data

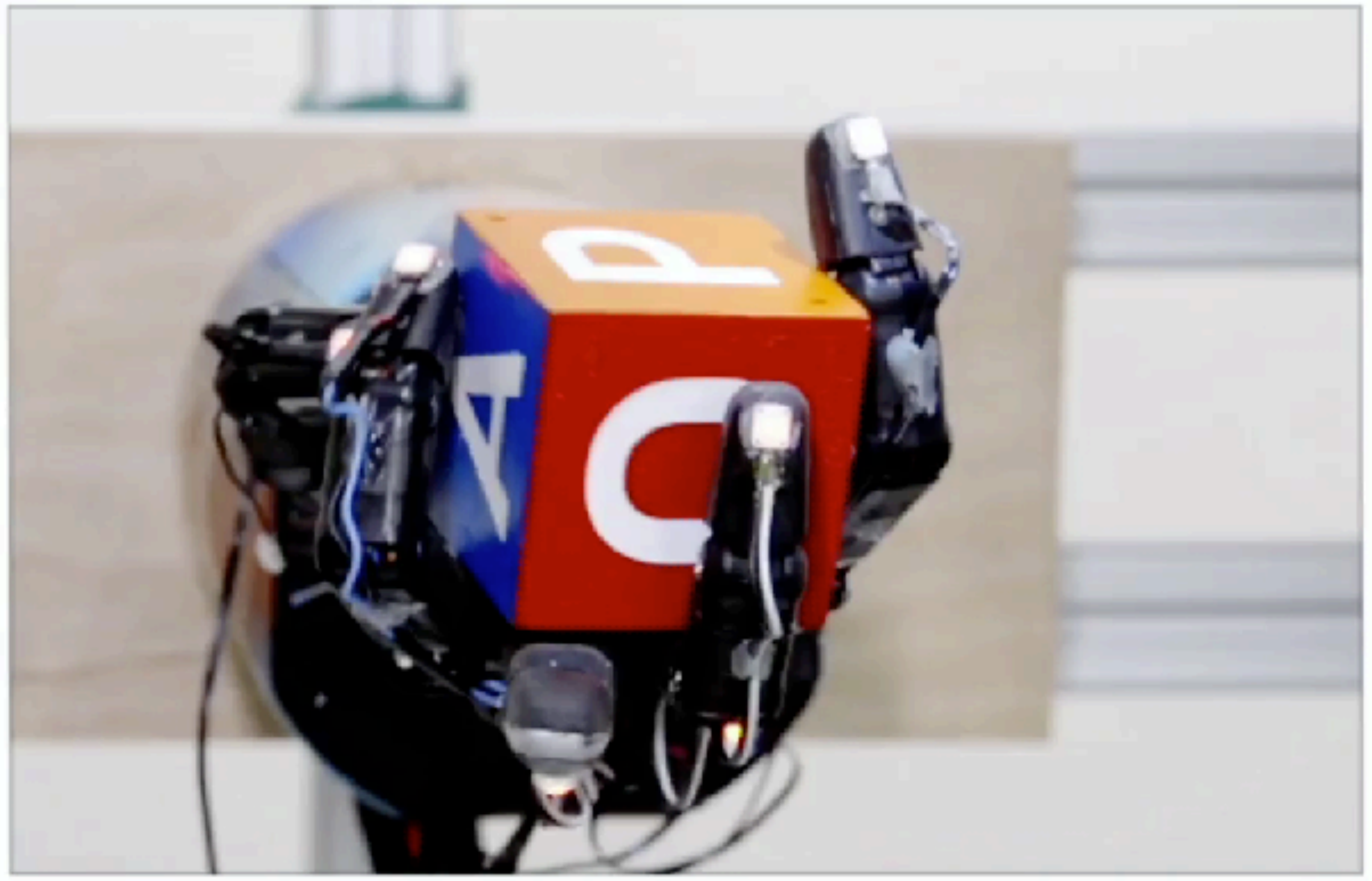
OpenAI Dactyl



FINGER PIVOTING



SLIDING

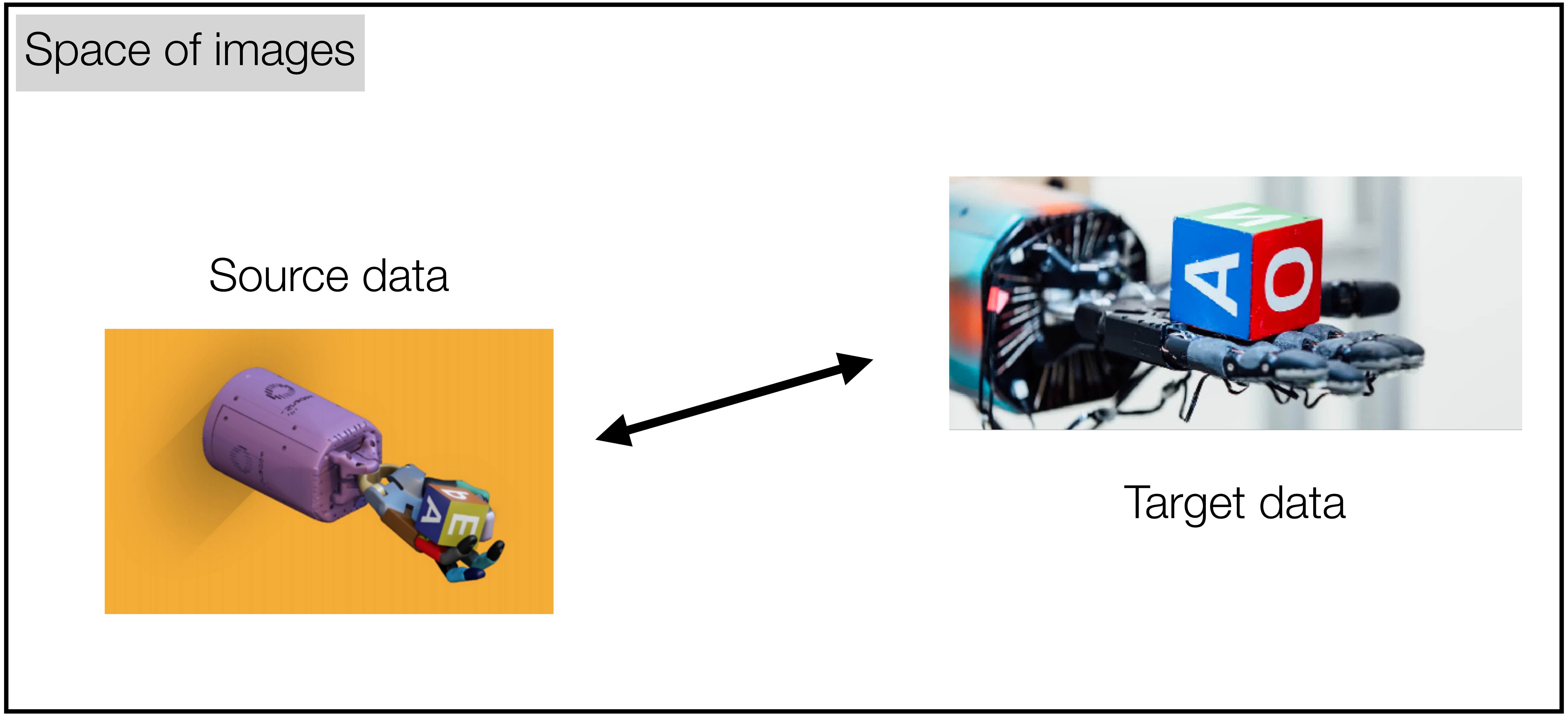


FINGER GAITING

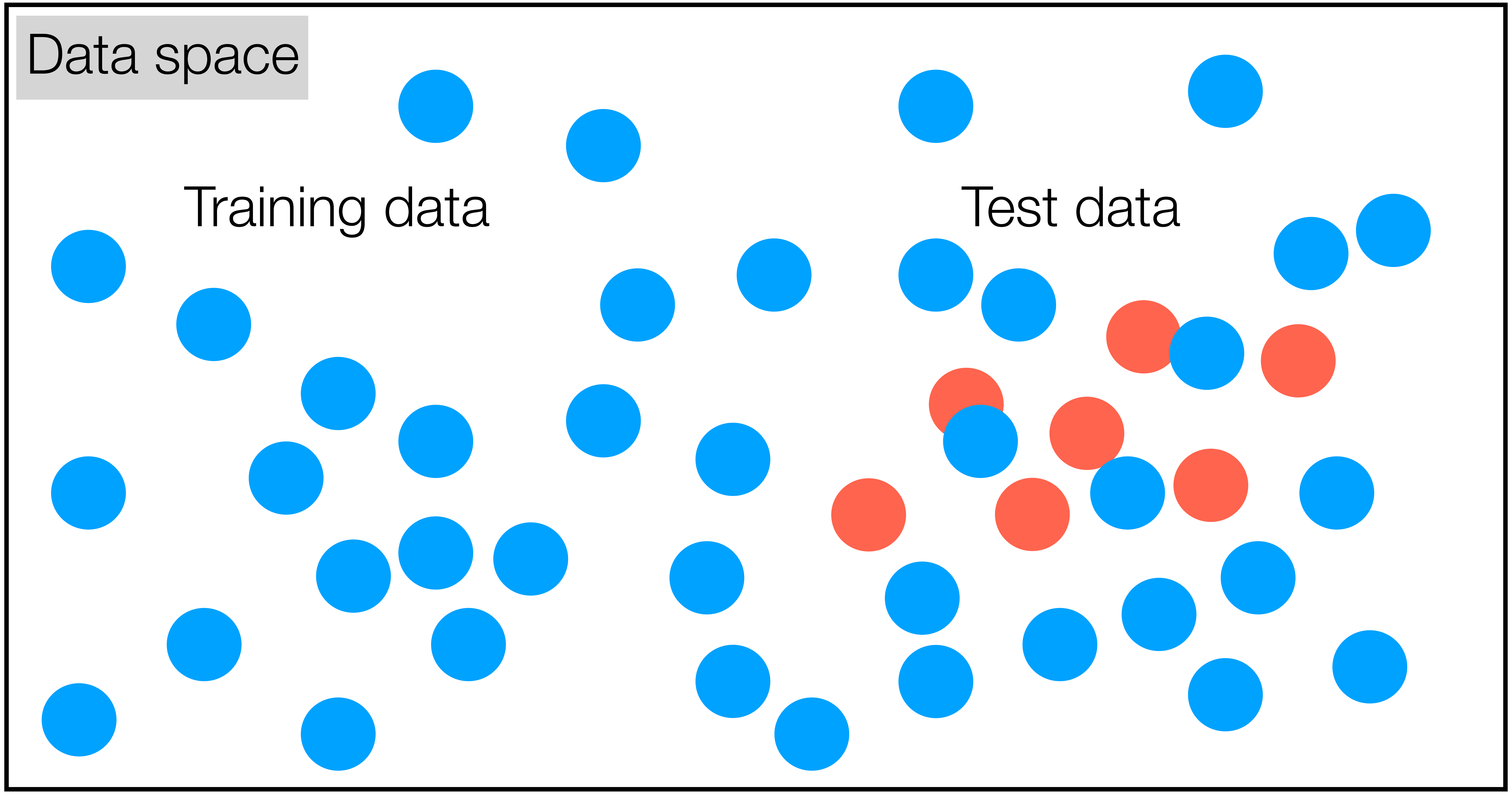
source domain

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.

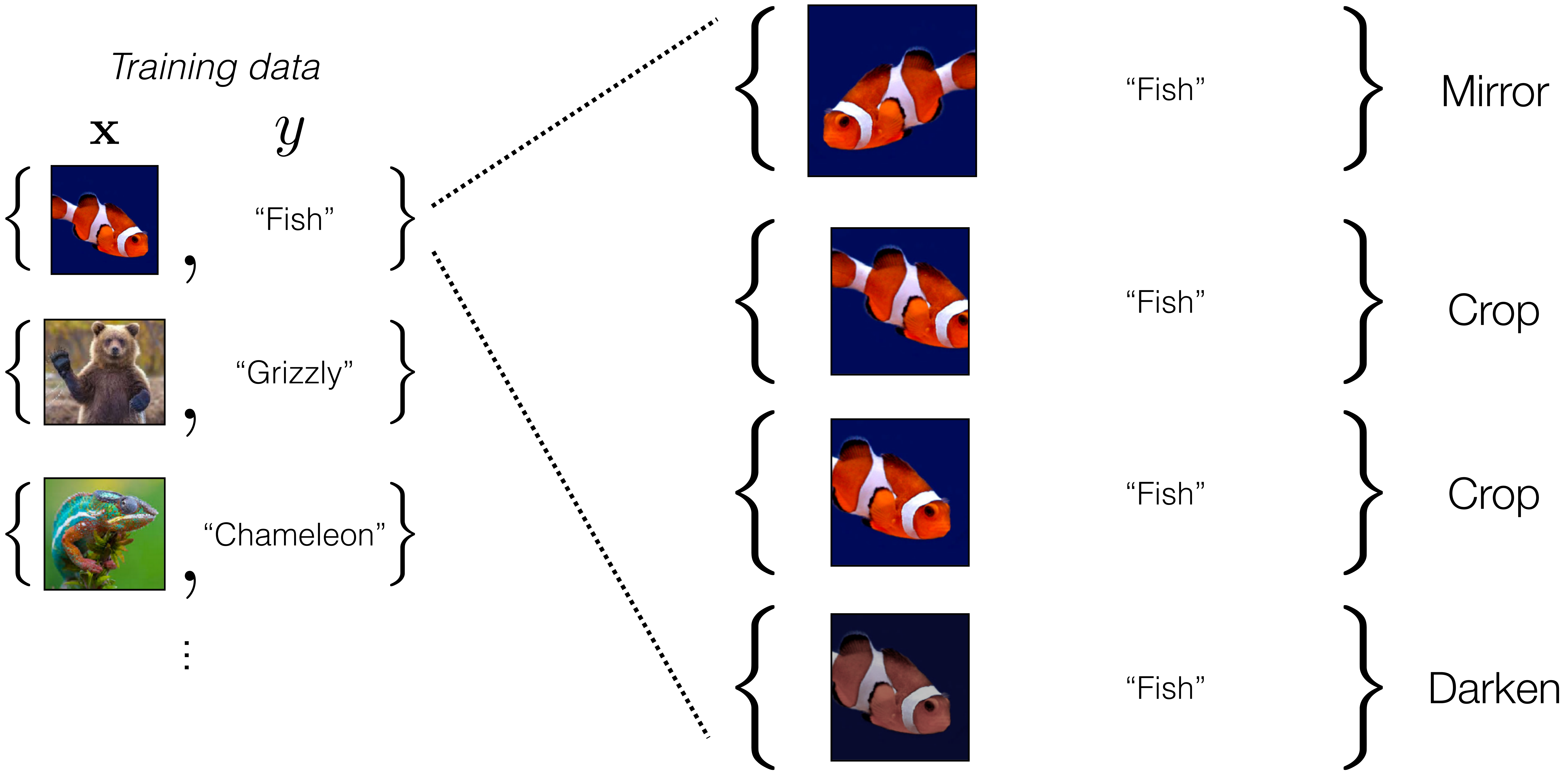


Idea #2: train on randomly perturbed data, so that test set just looks like another random perturbation



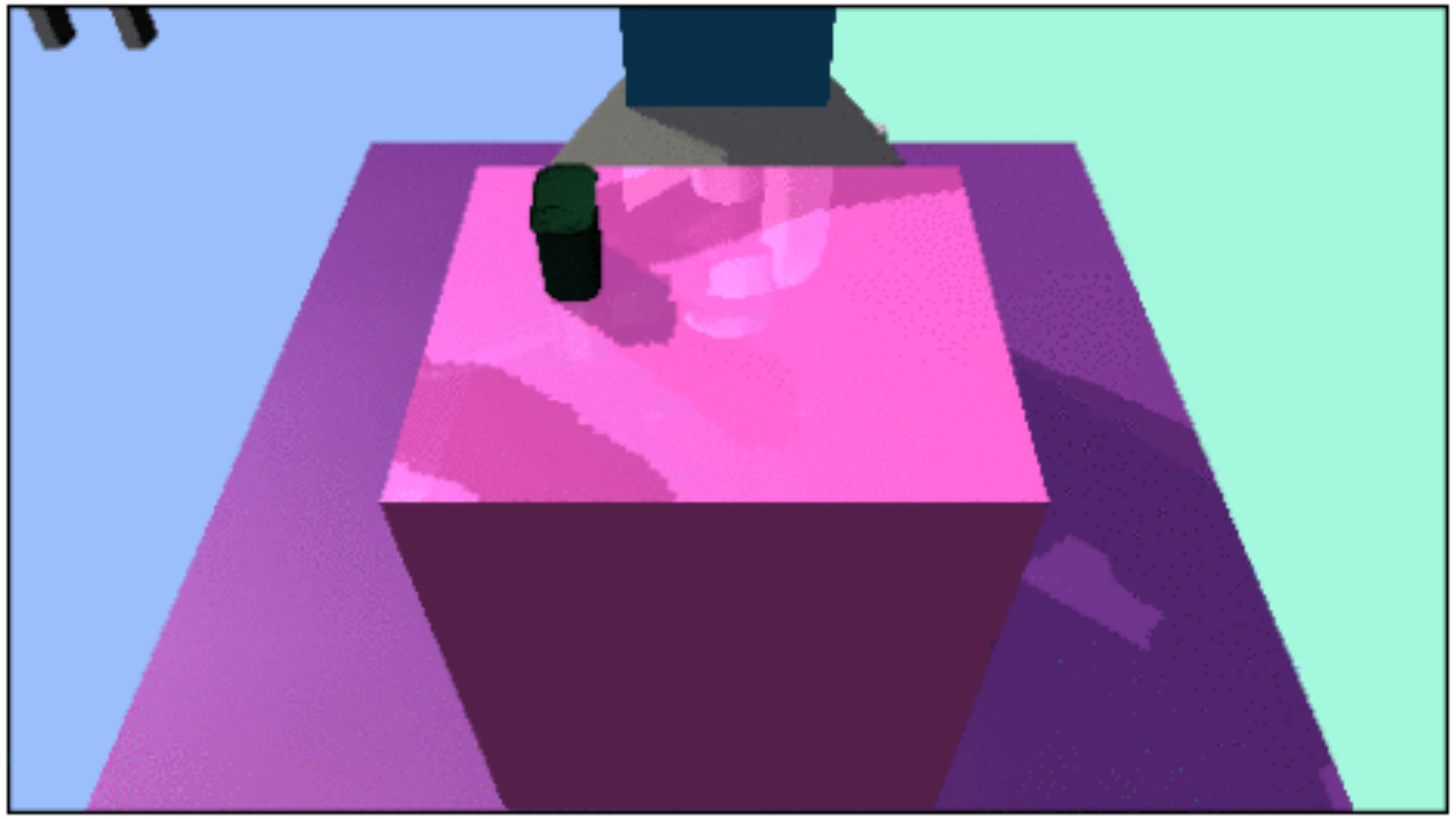
This is called **domain randomization** or **data augmentation**

Data augmentation

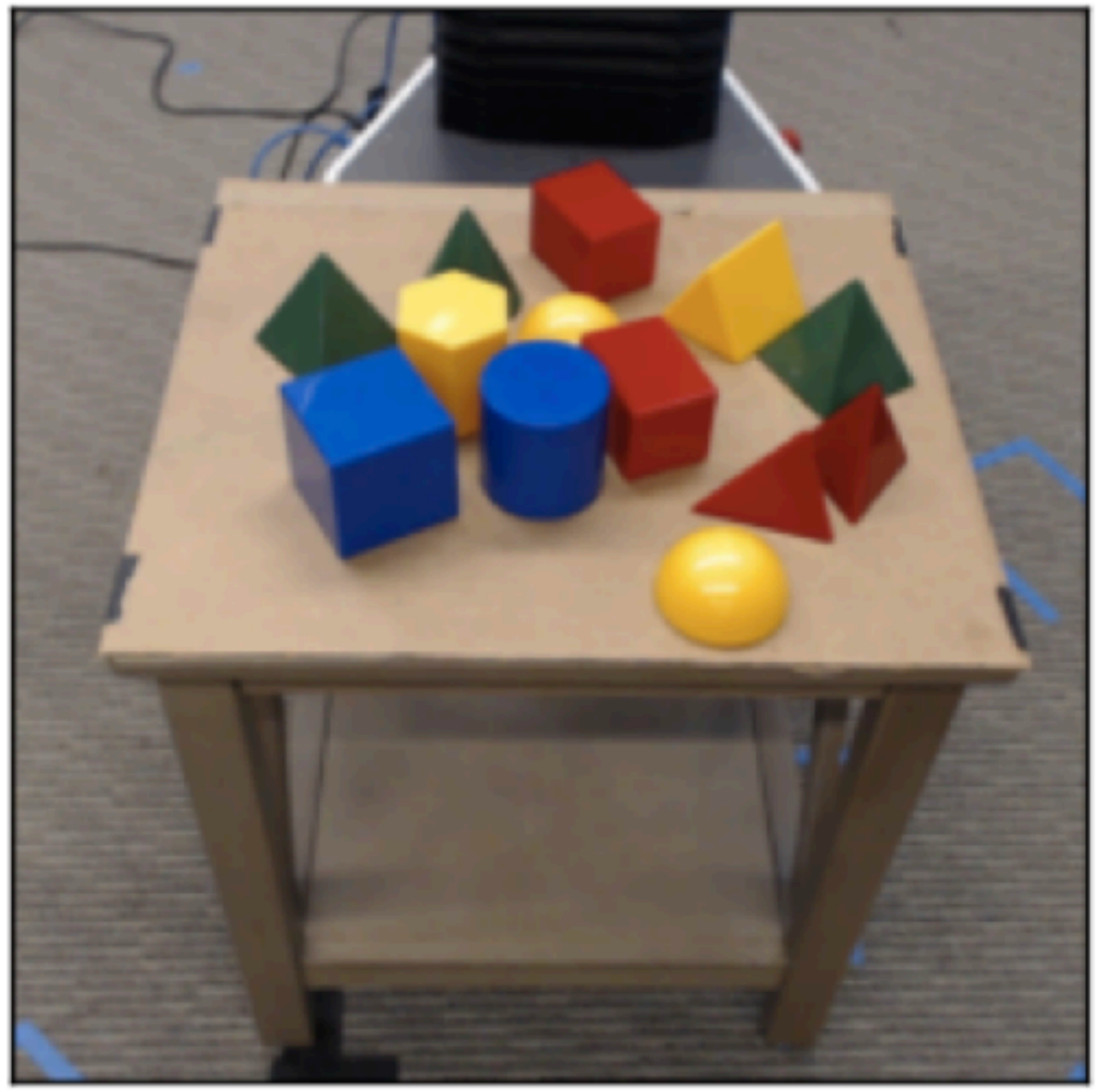


Domain randomization

Training data



Test data

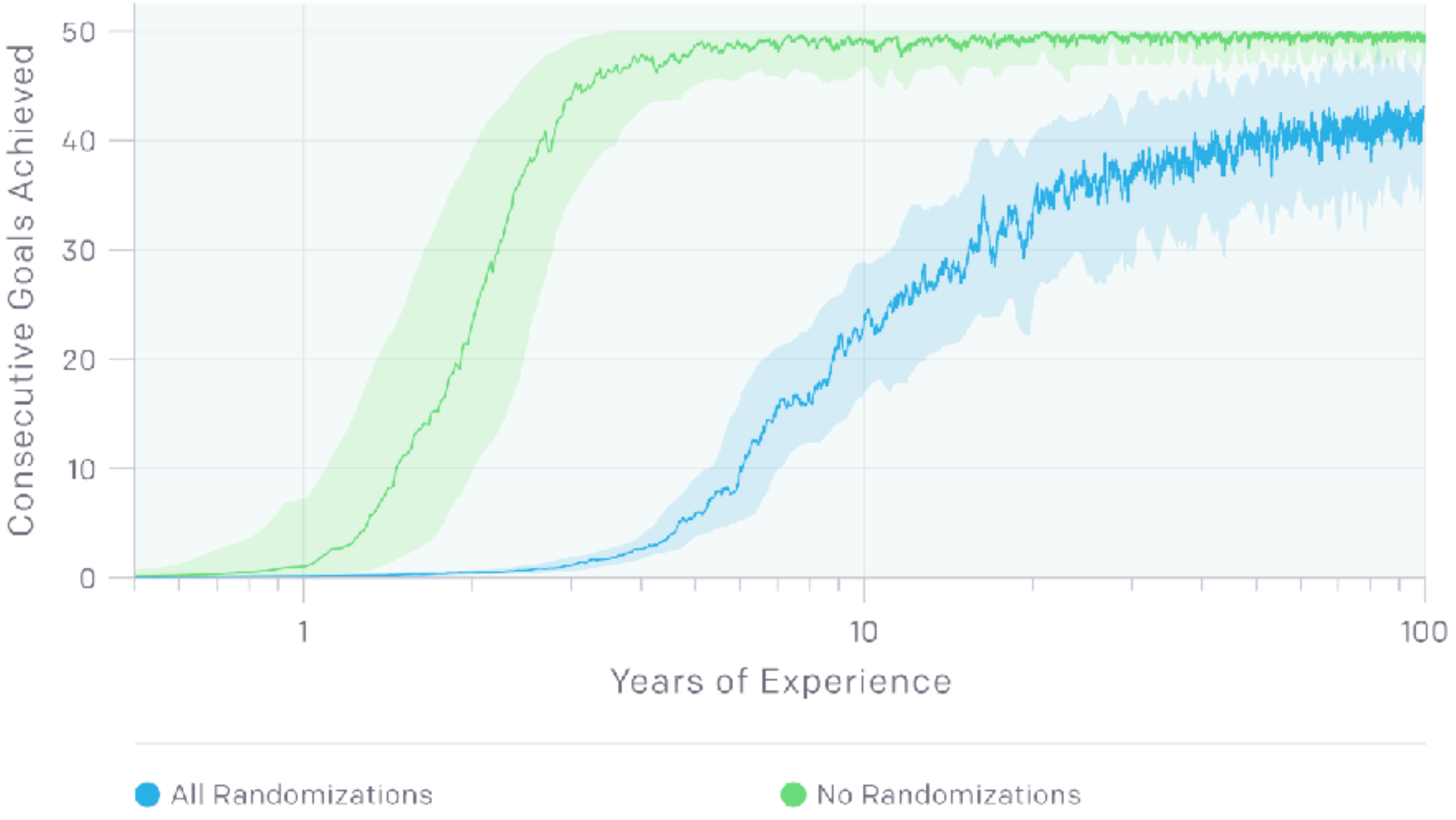


[Sadeghi & Levine 2016]

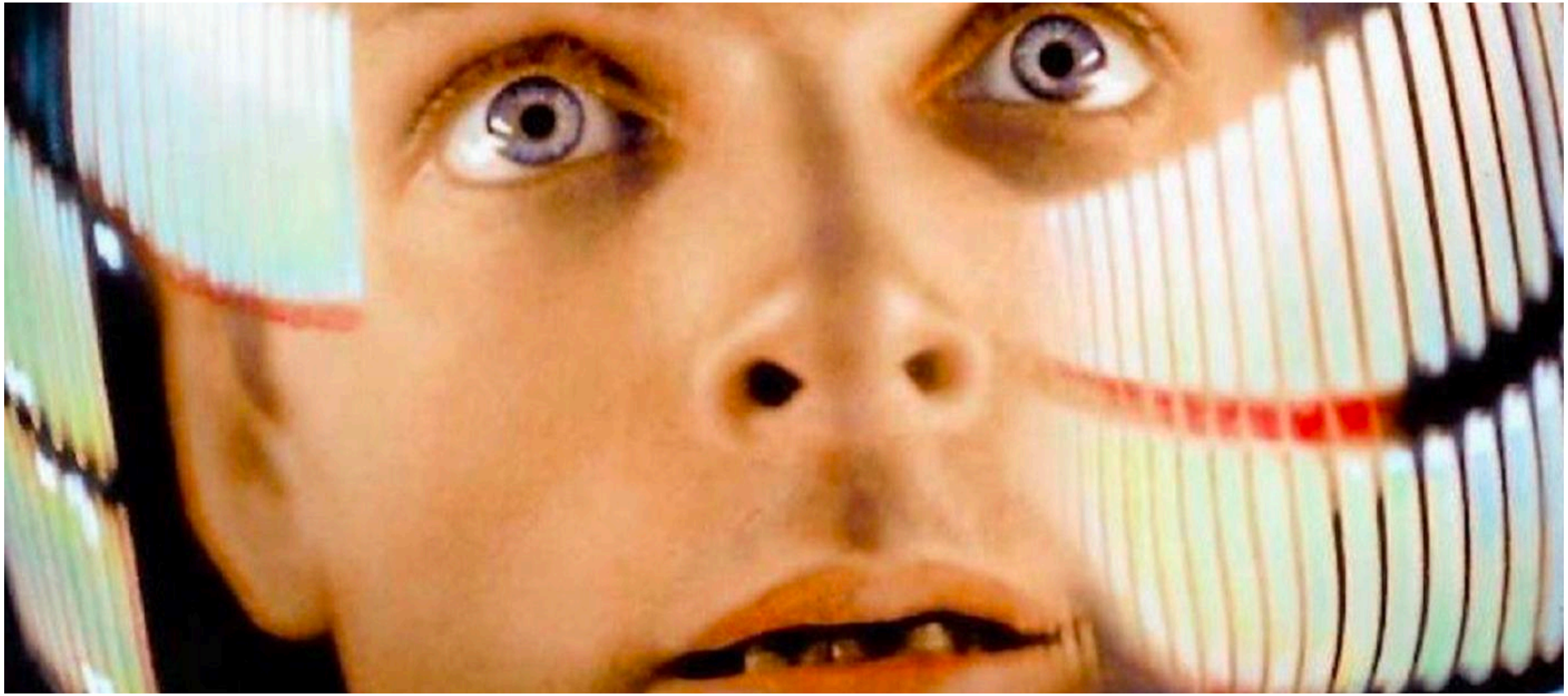
Above example is from [Tobin et al. 2017]

Table 1: Ranges of physics parameter randomizations.

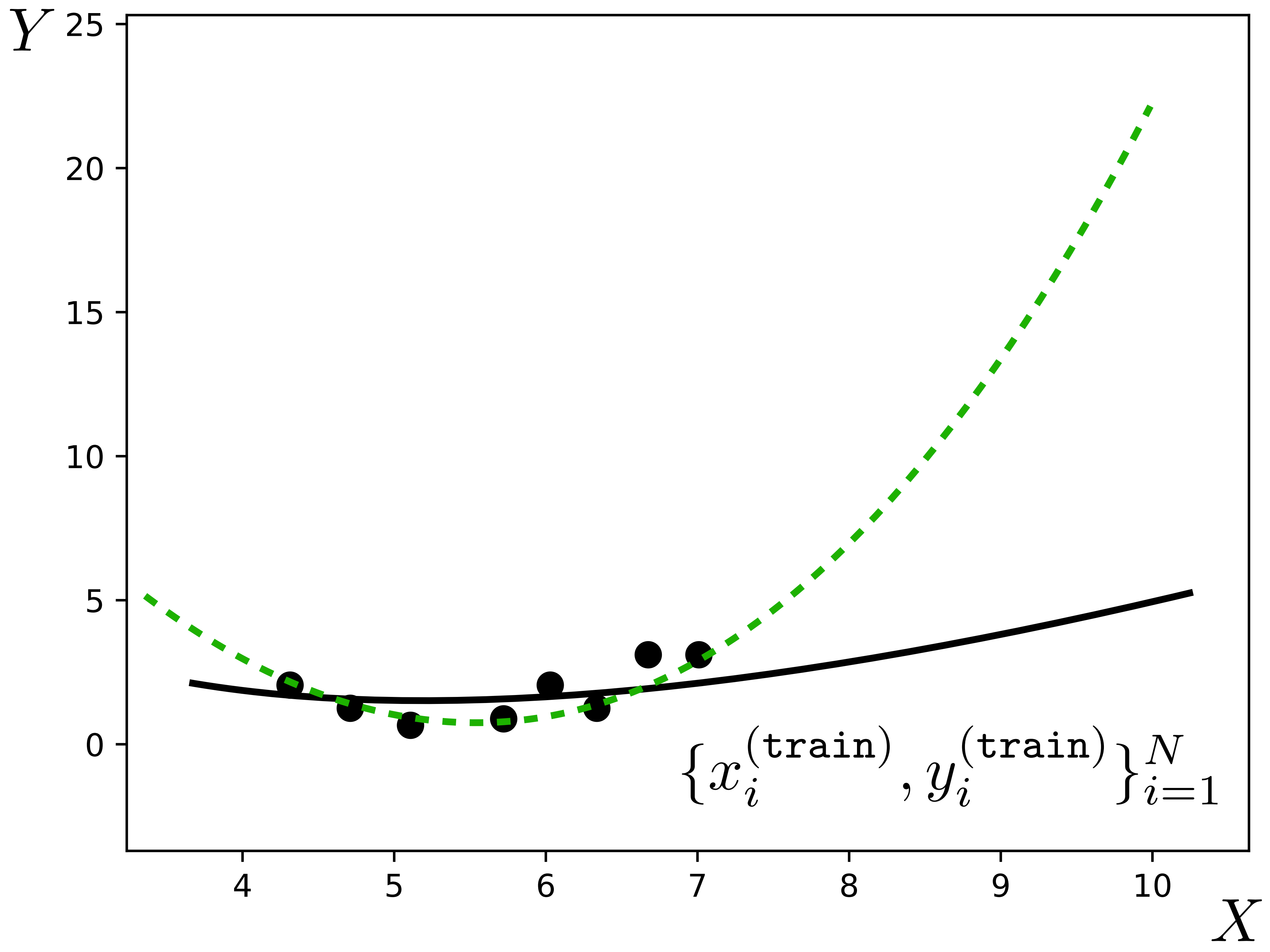
Parameter	Scaling factor range	Additive term range
object dimensions	uniform([0.95, 1.05])	
object and robot link masses	uniform([0.5, 1.5])	
surface friction coefficients	uniform([0.7, 1.3])	
robot joint damping coefficients	loguniform([0.3, 3.0])	
actuator force gains (P term)	loguniform([0.75, 1.5])	
joint limits		$\mathcal{N}(0, 0.15)$ rad
gravity vector (each coordinate)		$\mathcal{N}(0, 0.4)$ m/s^2



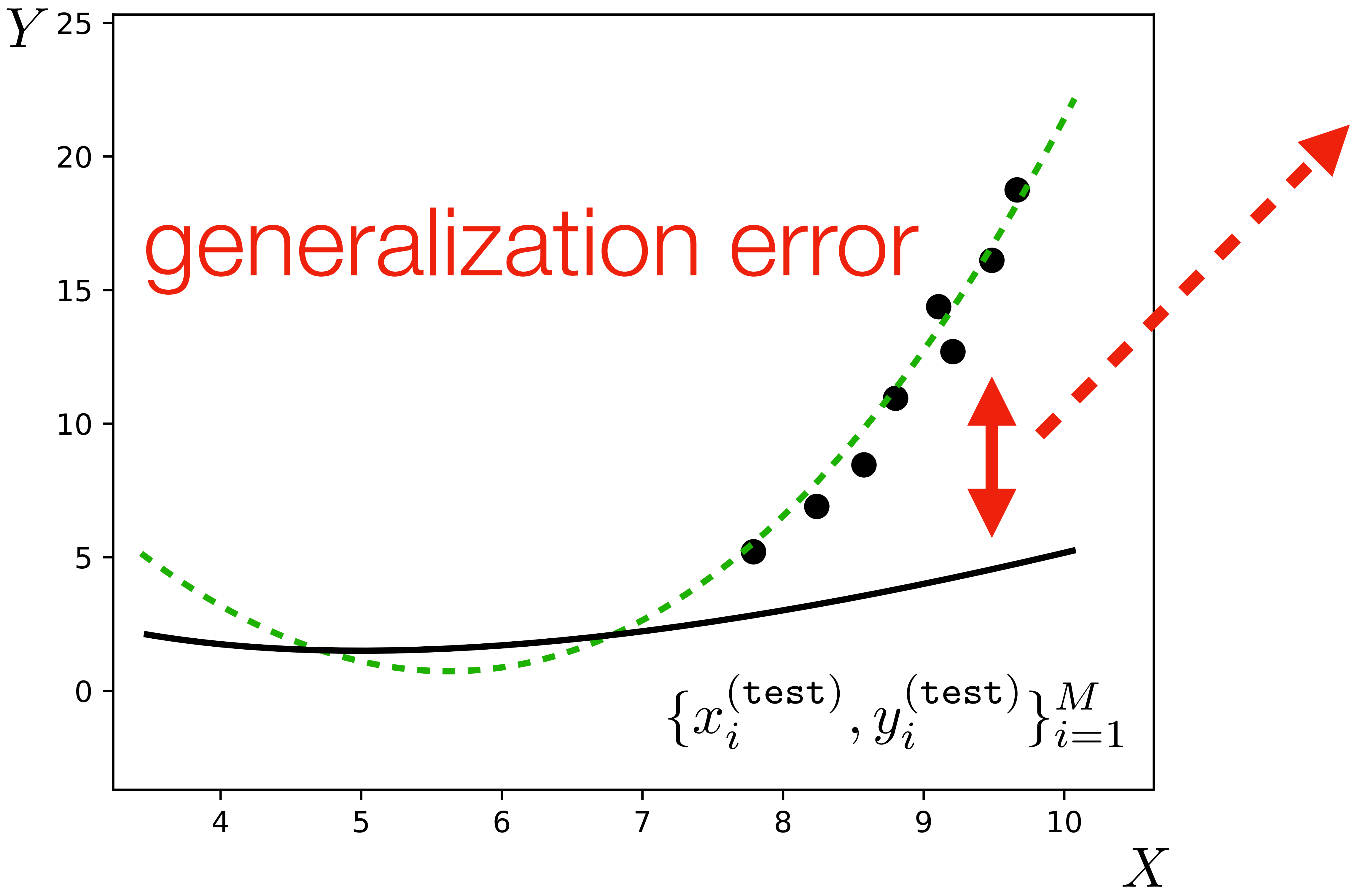
What if we go waaaay outside of the training distribution?



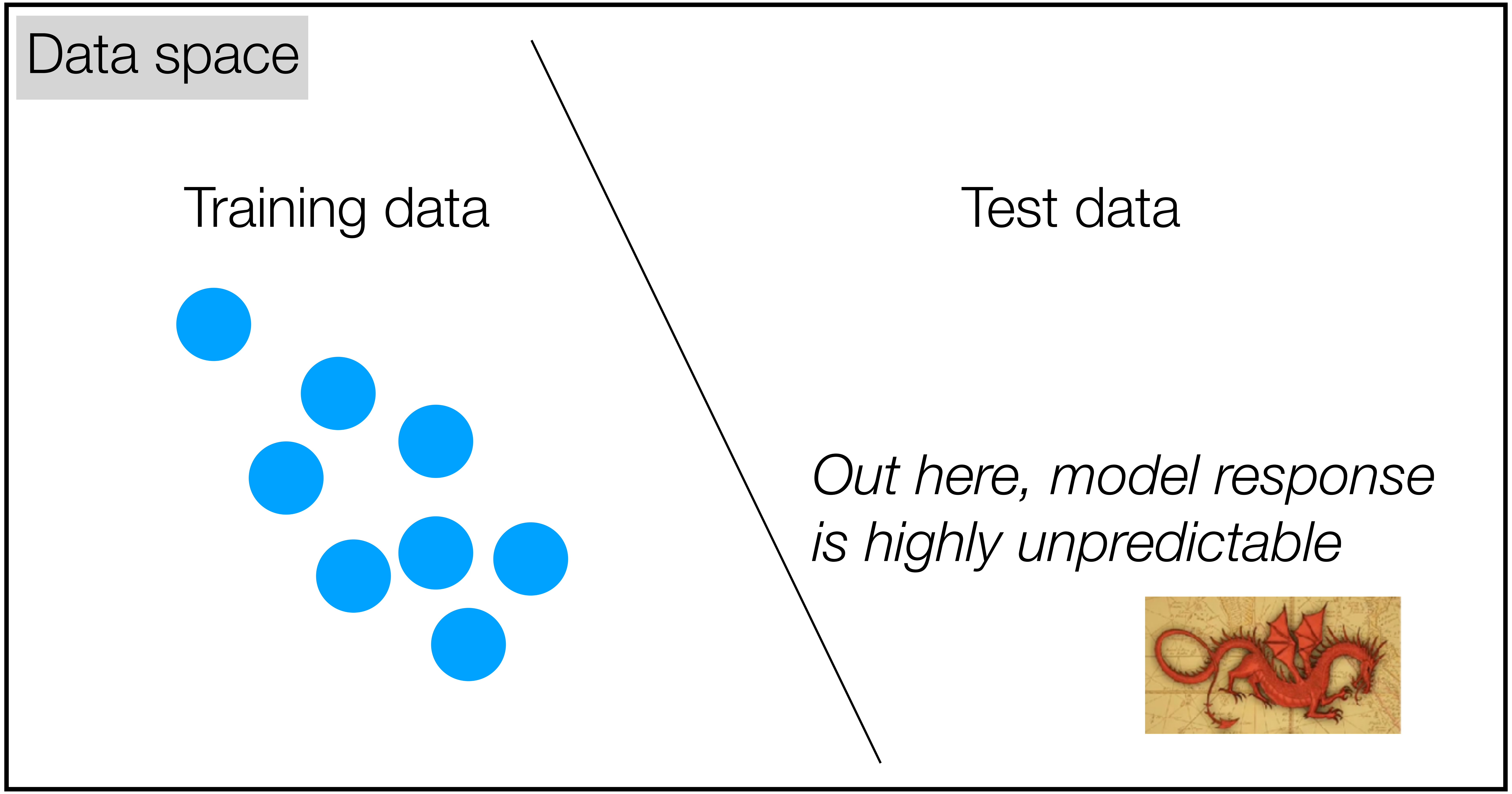
Training data



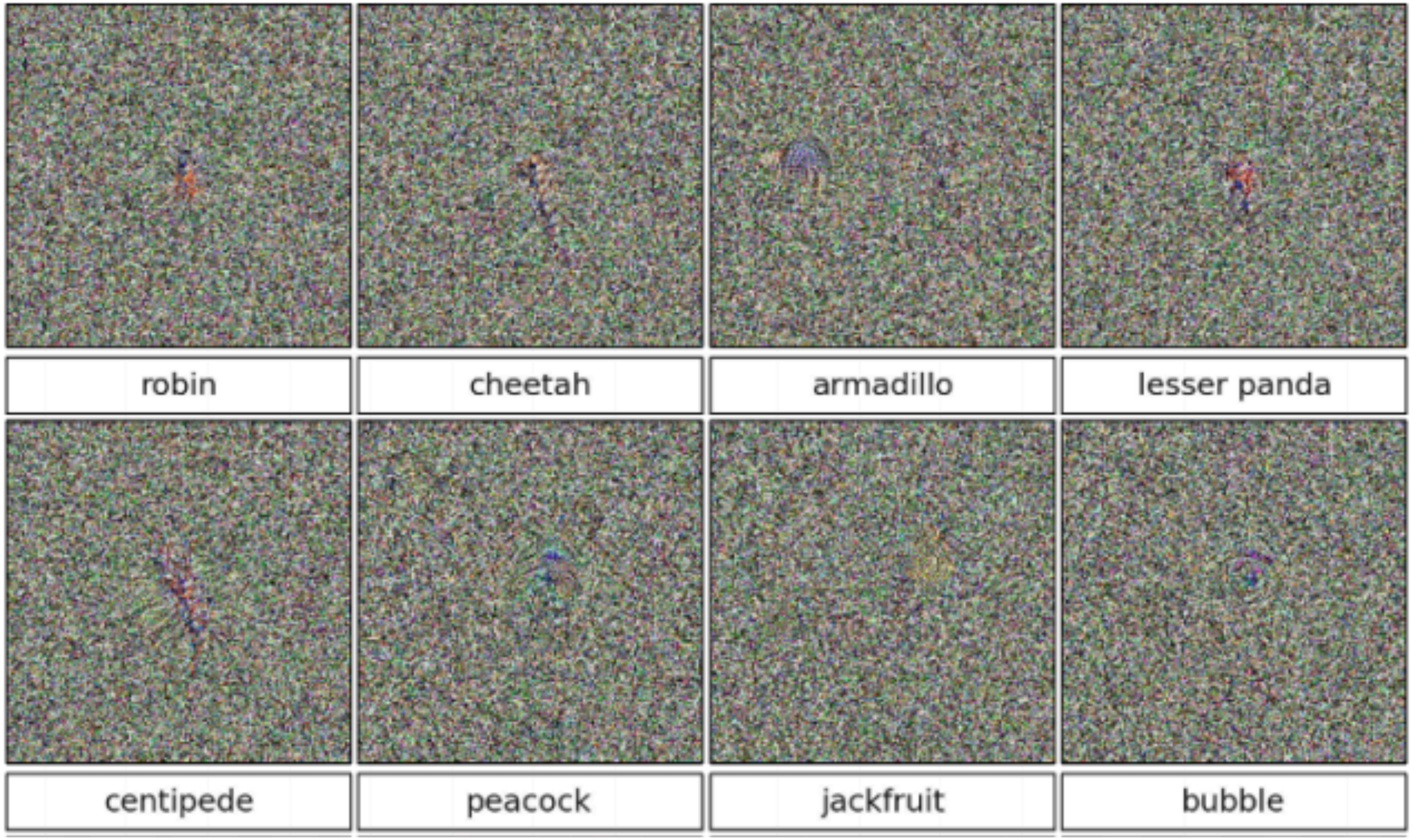
Test data



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

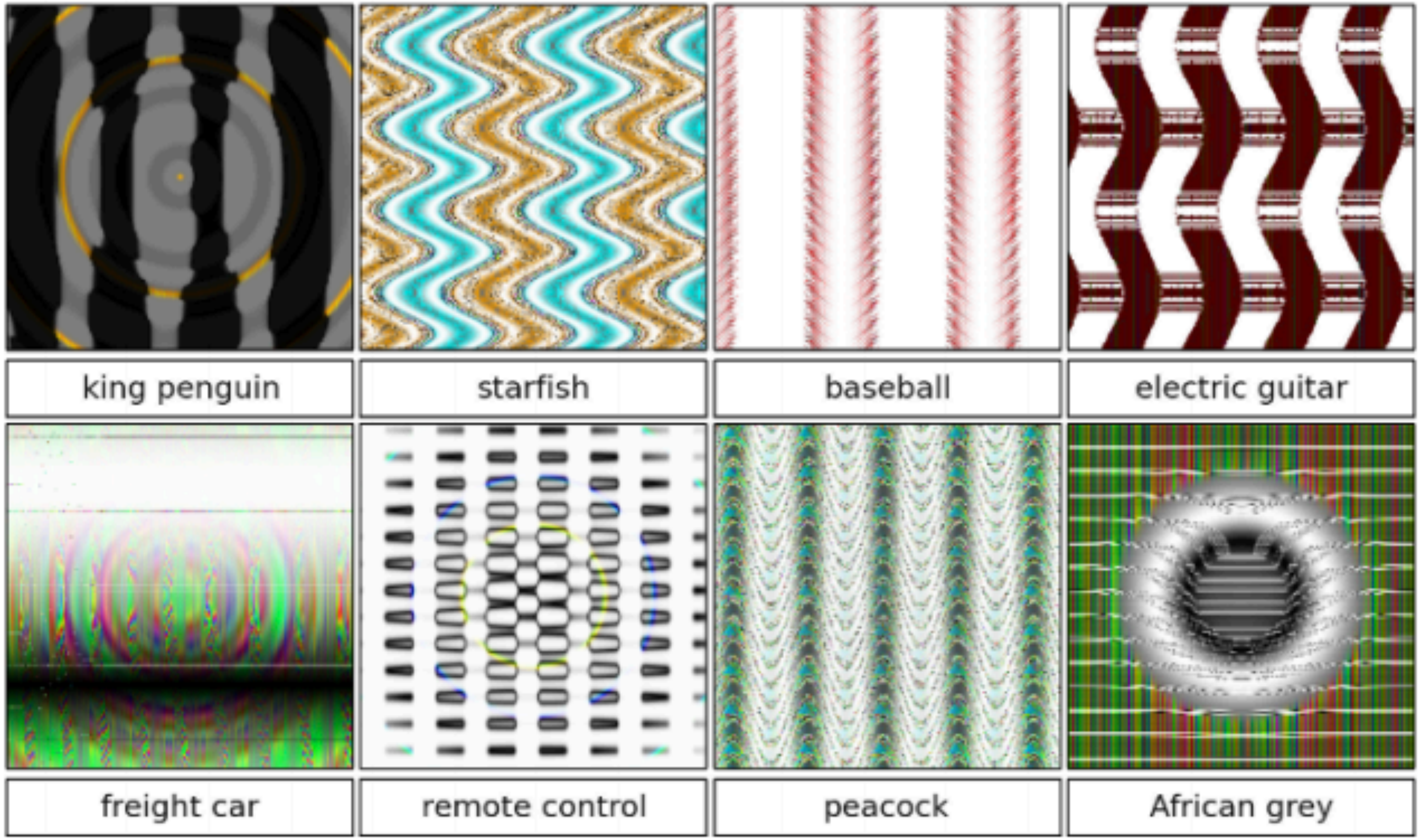

Data space
Training data
Test data
Out here, model response
is highly unpredictable

“Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images”
[Nguyen, Yosinski, and Clune, CVPR 2015]

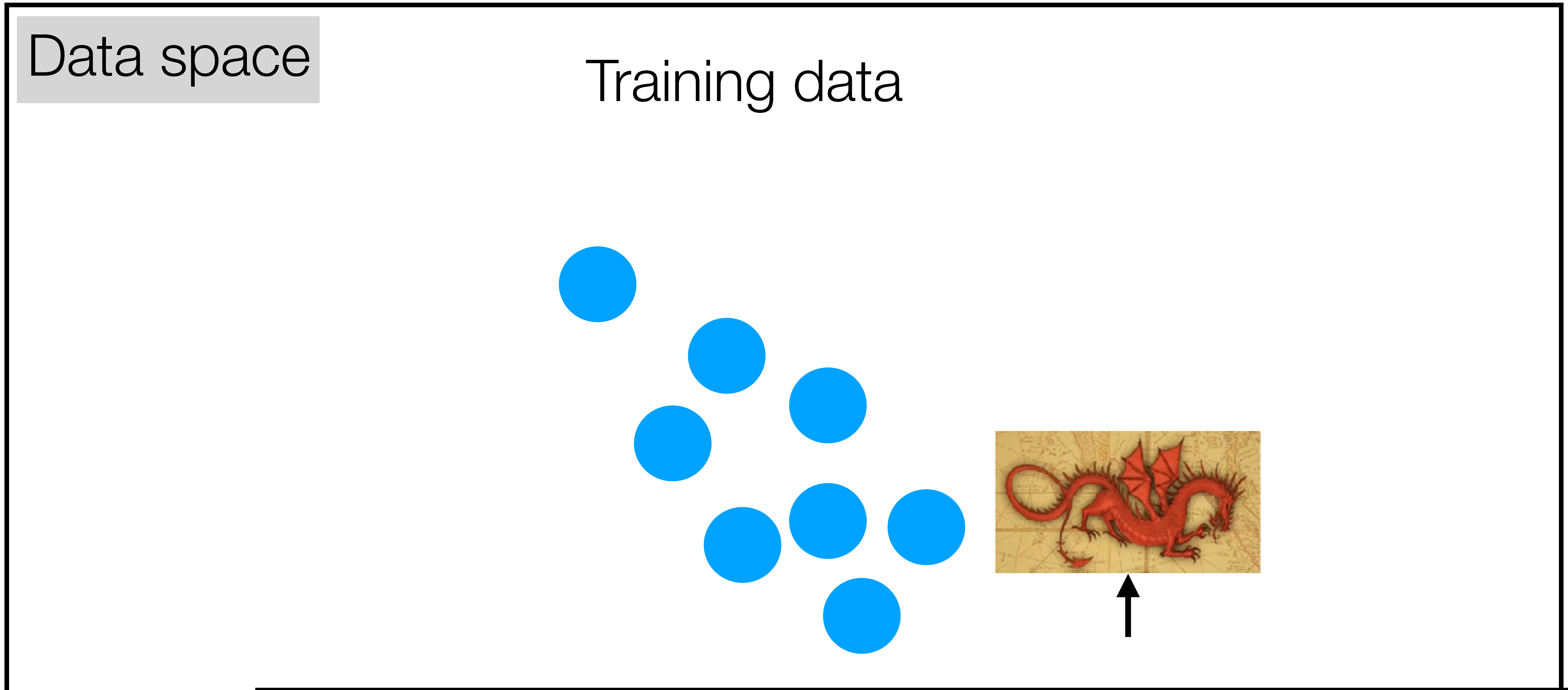


“Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images”

[Nguyen, Yosinski, and Clune, CVPR 2015]



Weirdness of high-dimensional space:

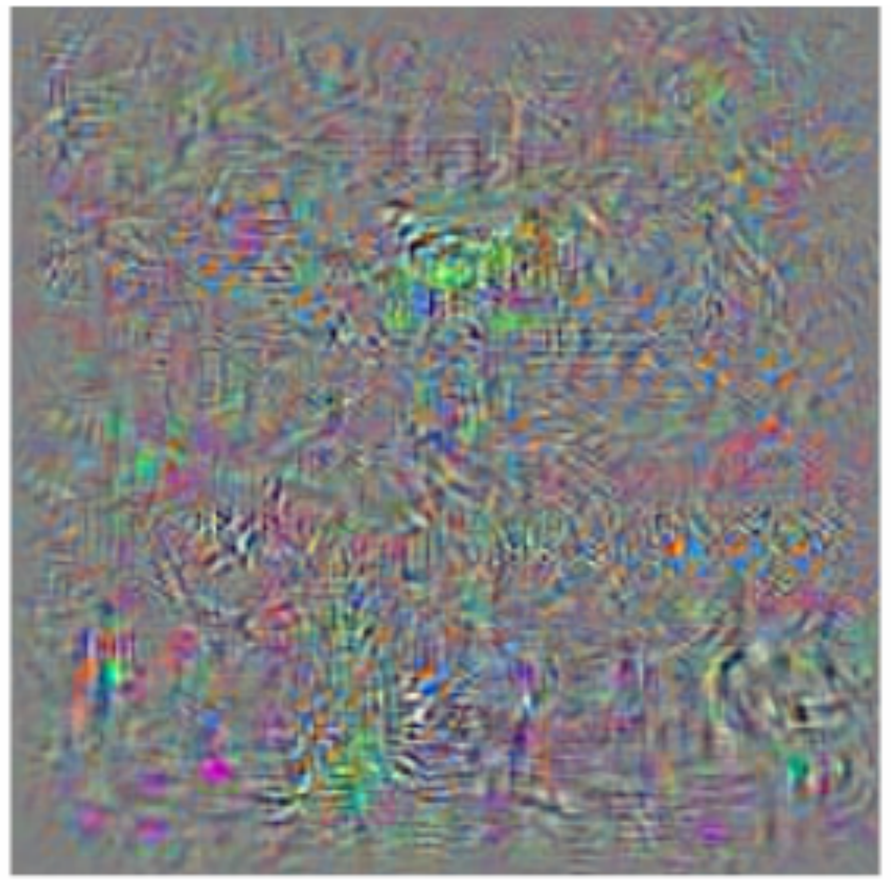


Usually, there are *blind spots* where the model has not fit the distribution well, and behaves unpredictably

Adversarial noise

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

+



=

f $\quad$ f

y "School bus" $\quad$ "Ostrich"

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

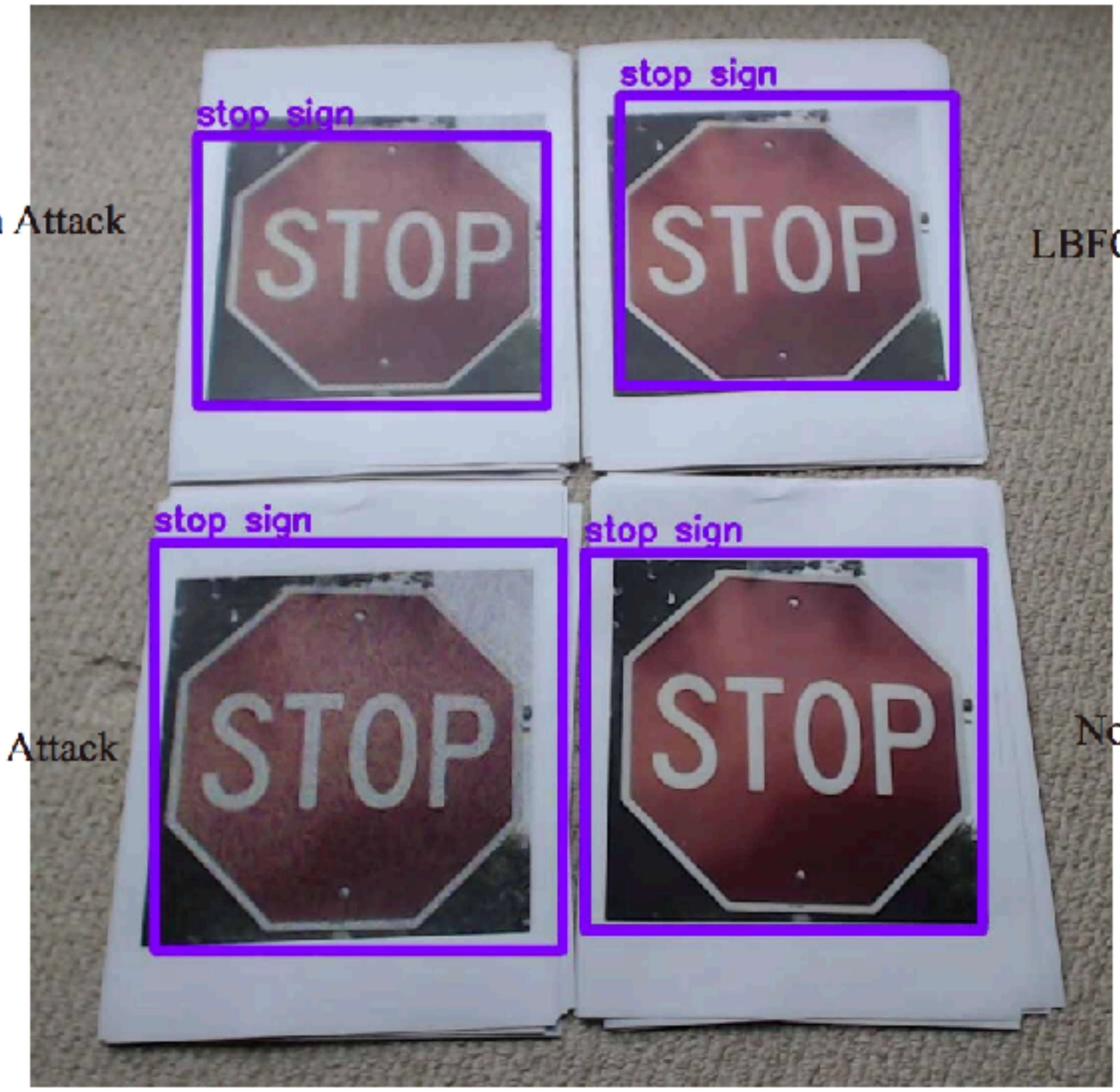
["Intriguing properties of neural networks", Szegedy et al. 2014]

Anything to worry about?

“NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles”, Lu et al. 2017

FastSign Attack

LBFGS Attack



Iterative Attack

No Attack

(Early) 2017’s attacks fail on physical objects, since they are optimized to attack a single view!

Anything to worry about?

Later in 2017…

“Synthesizing Robust Adversarial Examples”, Athalye, Engstrom, Ilyas, Kwok, 2017

3D-printed **turtle** model classified as **rifle** from most viewpoints

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