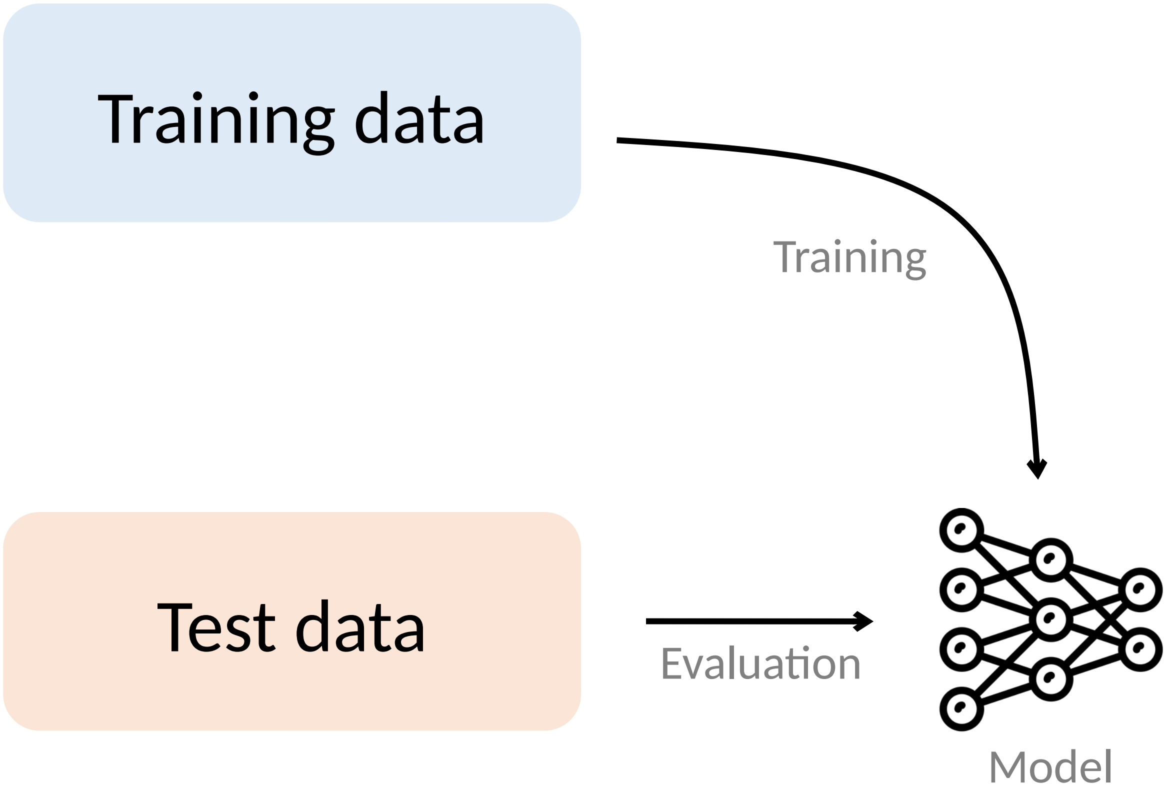
# Understanding models via their training data

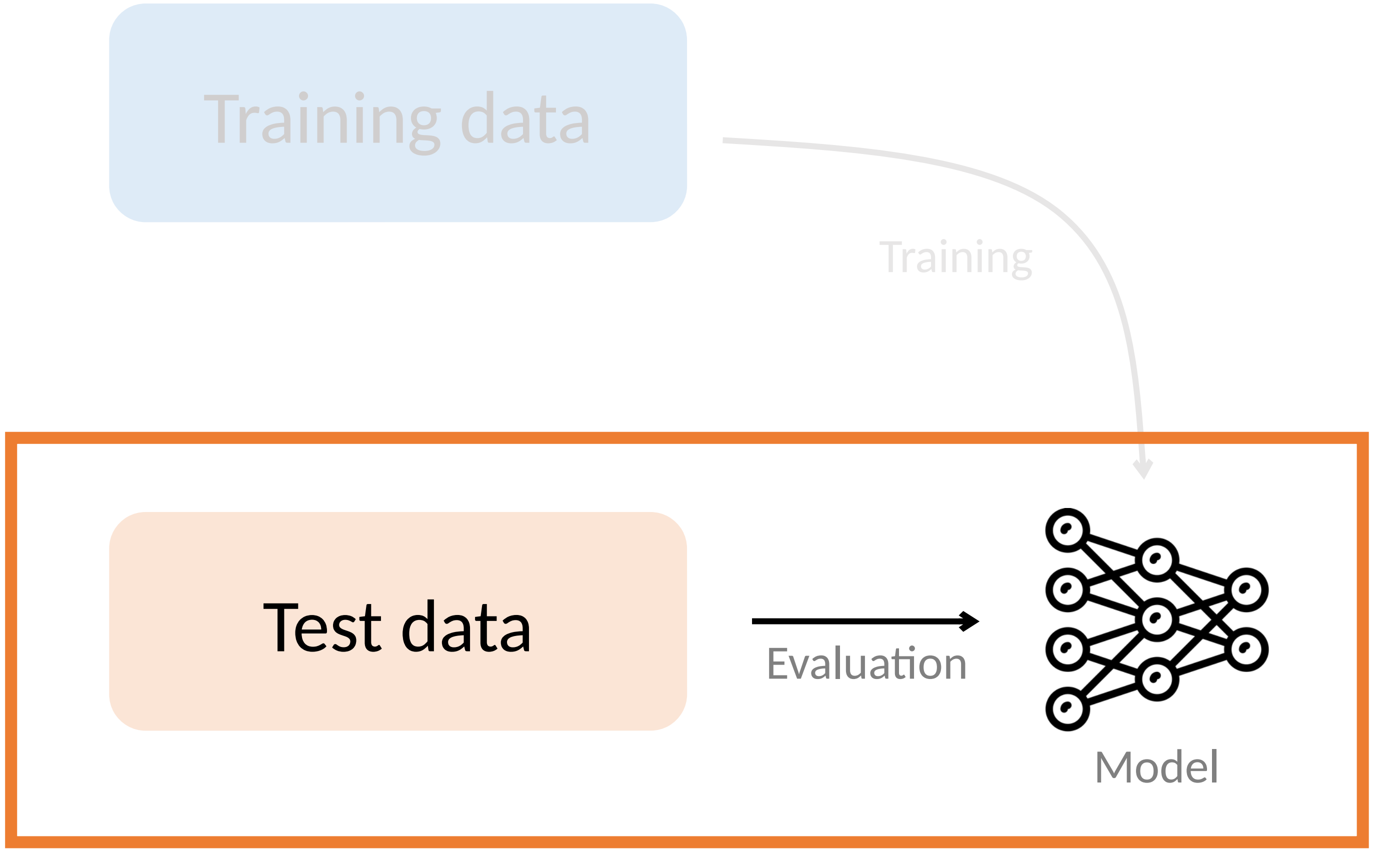
**Understanding black-box predictions via influence functions.**
Koh and Liang, ICML 2017.

Slides by Pang Wei Koh

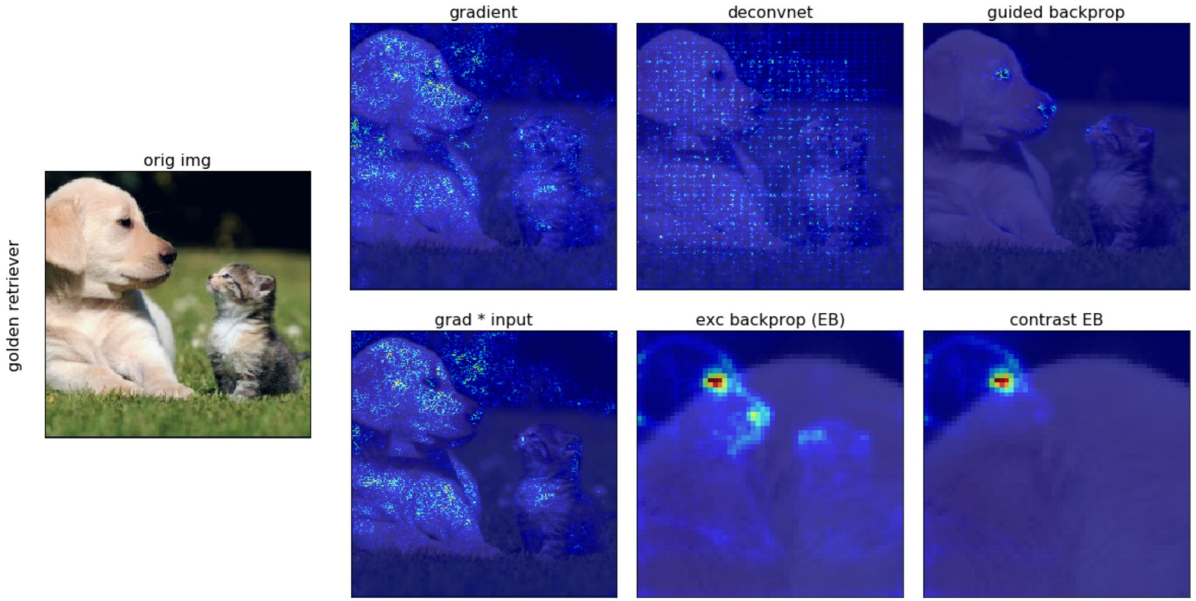
# Prior work: Focus on test data



# Prior work: Focus on test data



# Which parts of the test input most affect the model’s prediction?

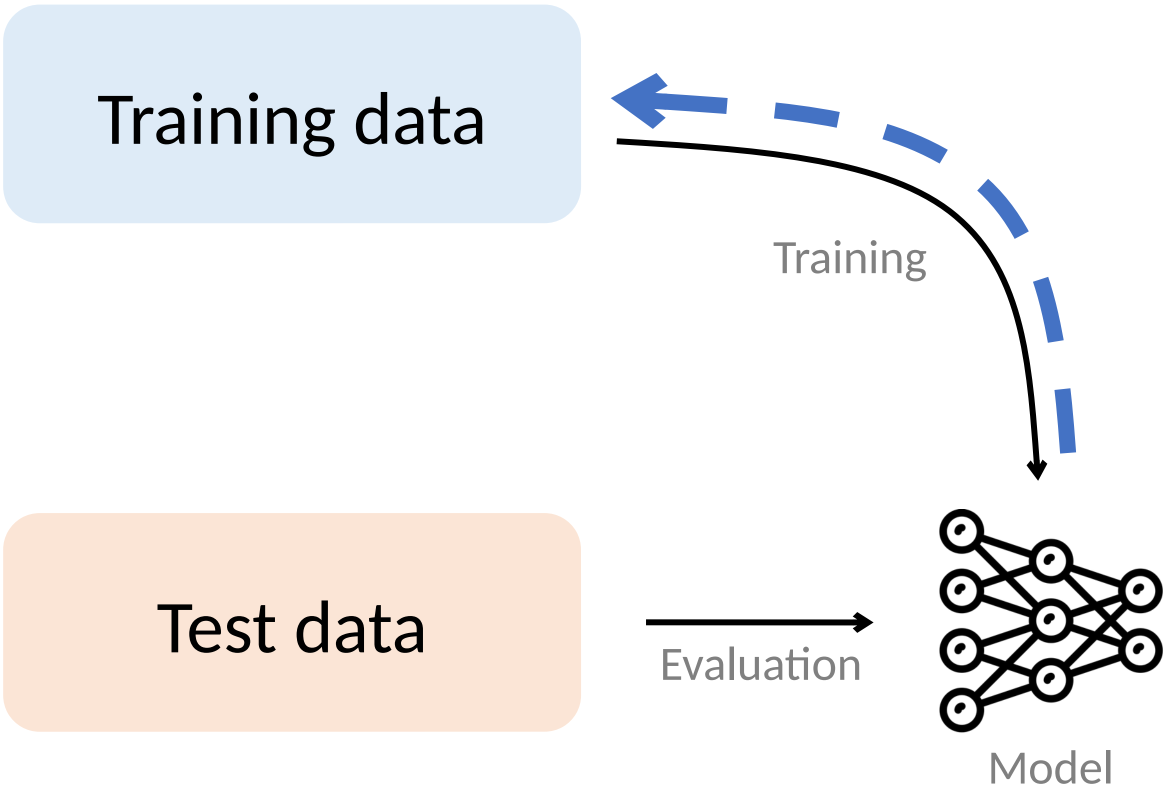



[Ruth Fong’s interpretability tutorial, 2019]

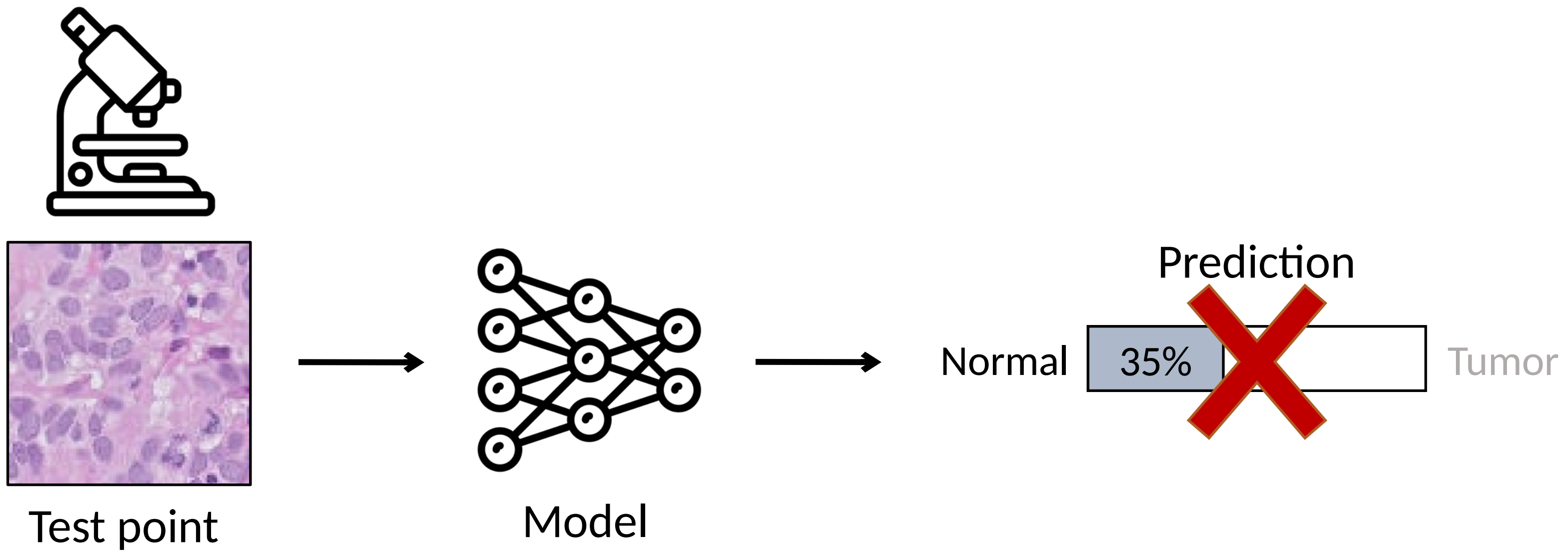
Class activation maps [Zhou et al., 2016]
Concept activation vectors [Kim et al., 2018]
DeConvNet [Zeiler & Fergus, 2014]
Deep Taylor decomposition [Montavon et al., 2017]
DeepLIFT [Shrikumar et al., 2017]
Deletion game [Fong & Vedaldi, 2017]
Generalized additive models [Caruana et al., 2015]
Gradients [Baehrens et al., 2010]
…

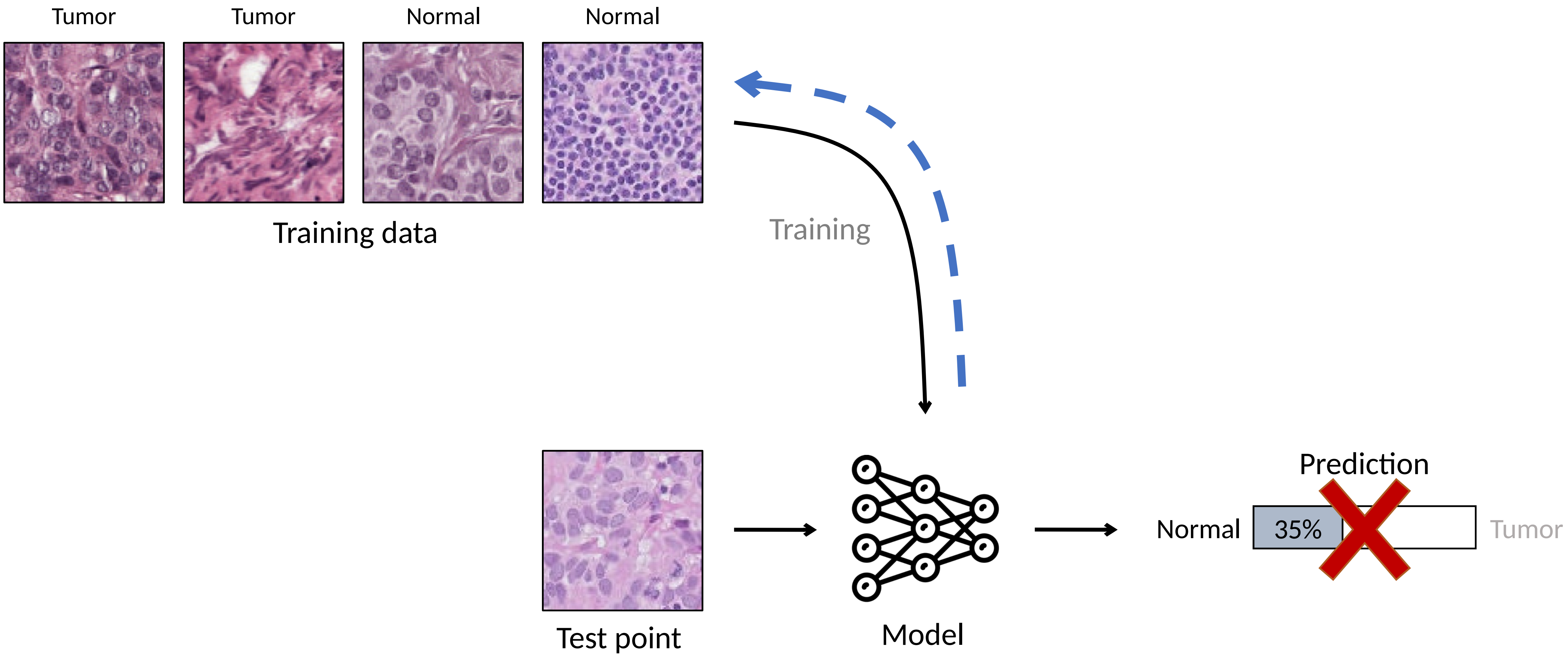
Guided BackProp [Springenberg et al., 2015]
Grad-CAM [Selvaraju et al., 2016]
Integrated Gradients [Sundararajan et al., 2017]
Layer-wise relevance propagation [Bach et al., 2015]
LIME [Ribeiro et al., 2016]
Saliency maps [Simonyan et al., 2014]
SHAP [Lundberg et al., 2017]
SmoothGrad [Smilkov et al., 2017]
…

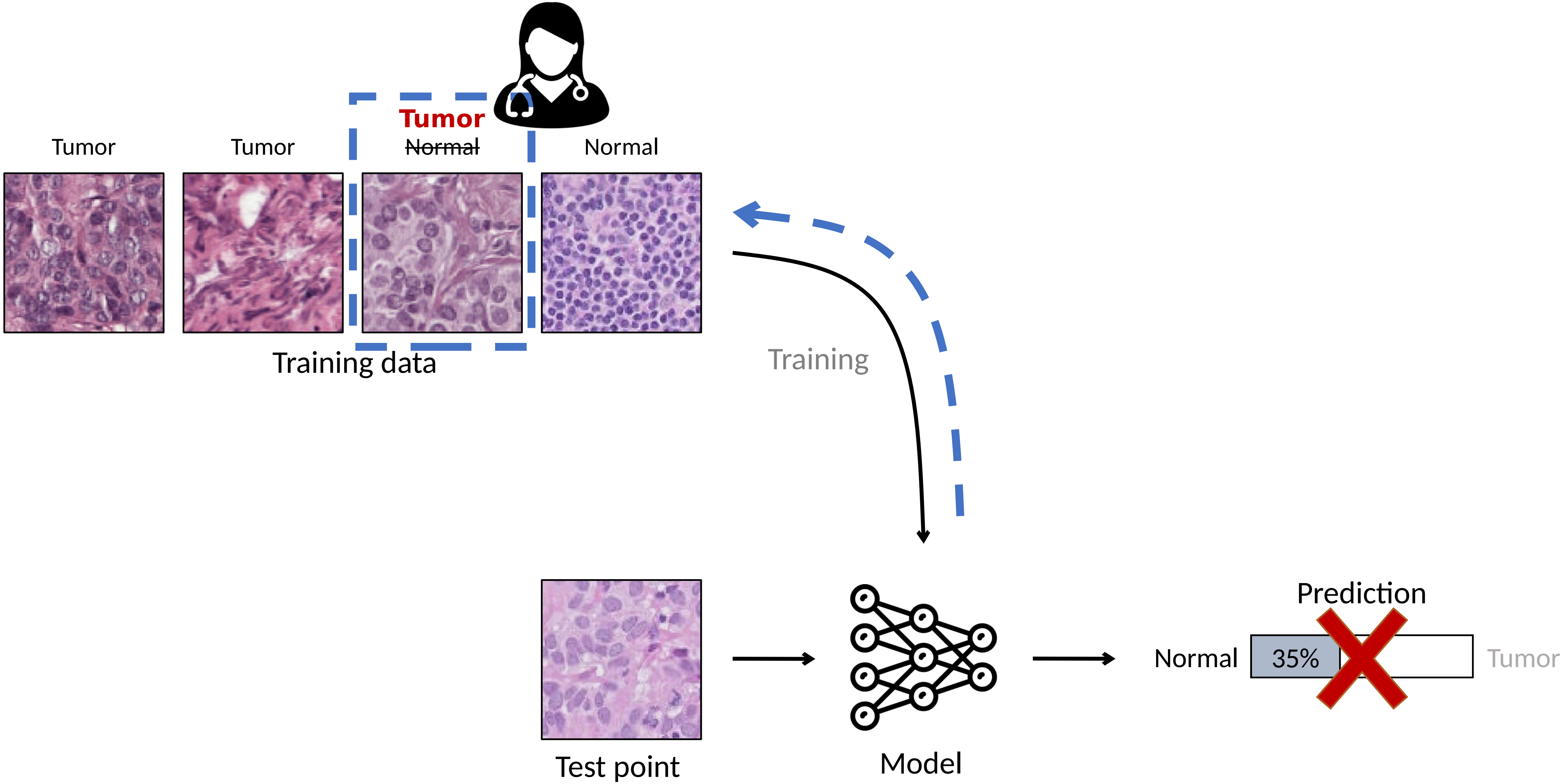
# Our work: Link model to training data



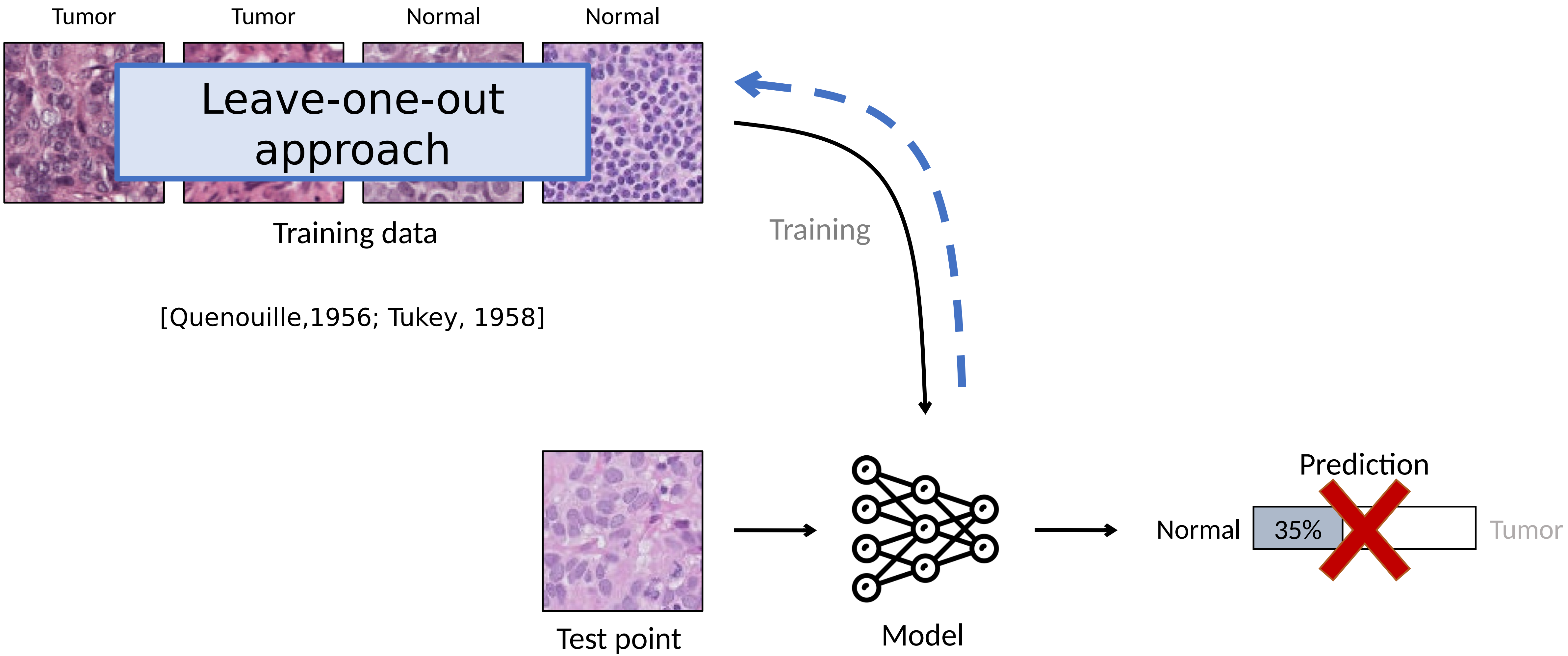
# Example: Dataset debugging



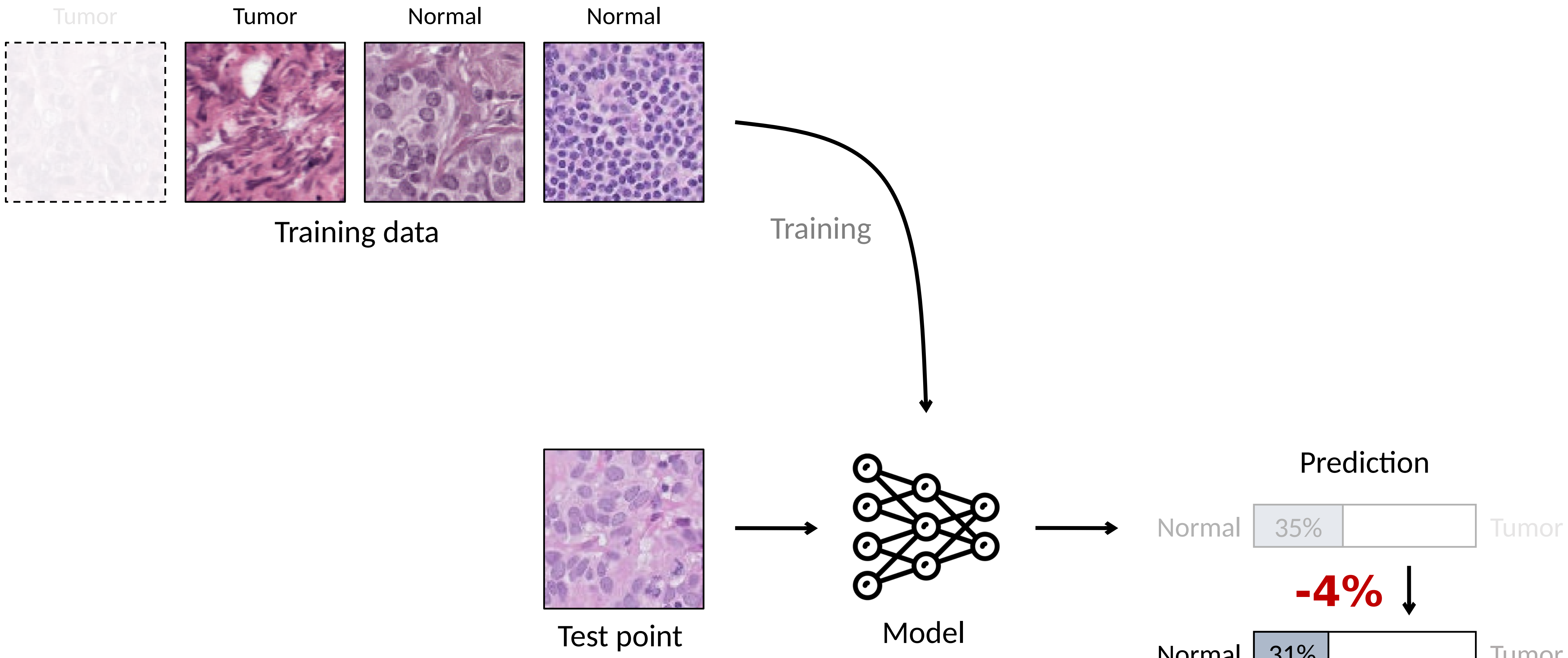

Tumor
Tumor
Normal
Normal
Training data
Training
Test point
Model
Prediction
Normal
35%
Tumor


Tumor
Tumor
Tumor
~~Normal~~
Normal
Training data
Training
Test point
Model
Prediction
Normal
35%
Tumor

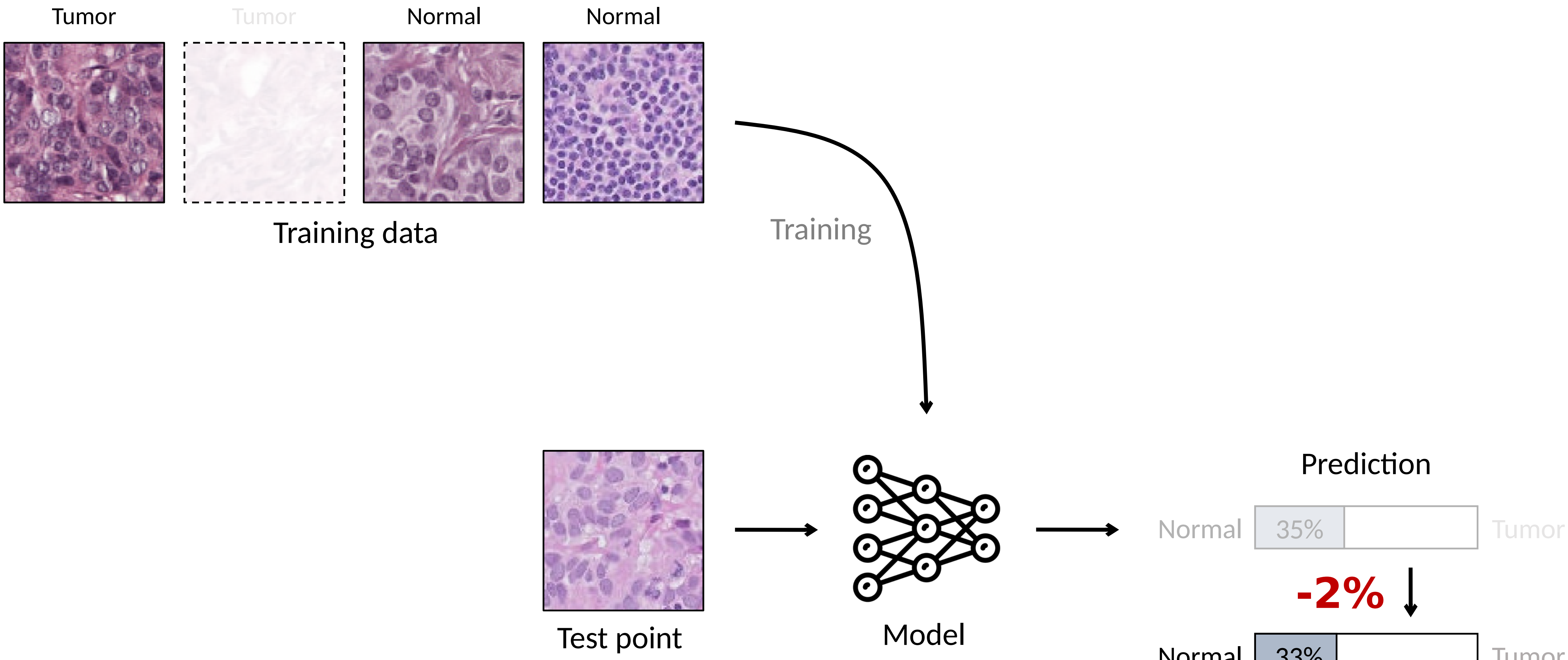
# Leave-one-out approach



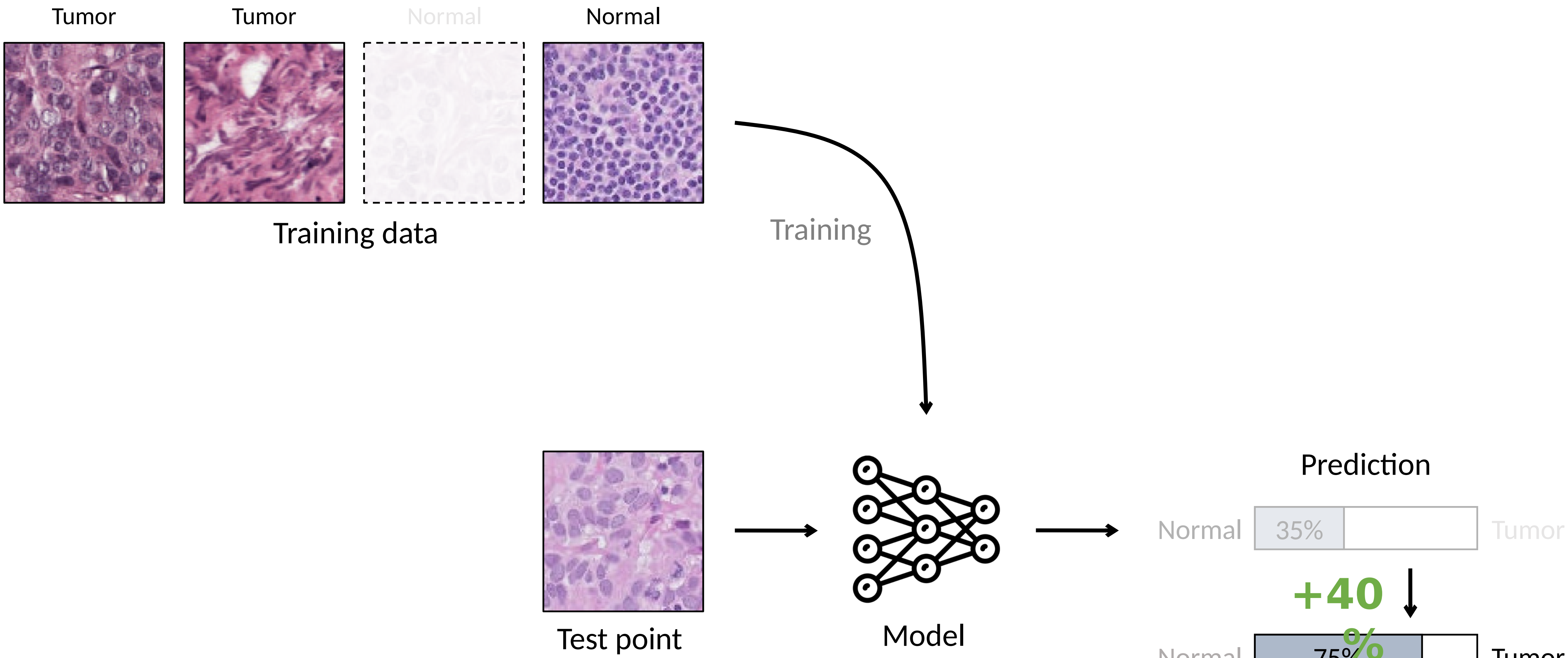

[Quenouille,1956; Tukey, 1958]

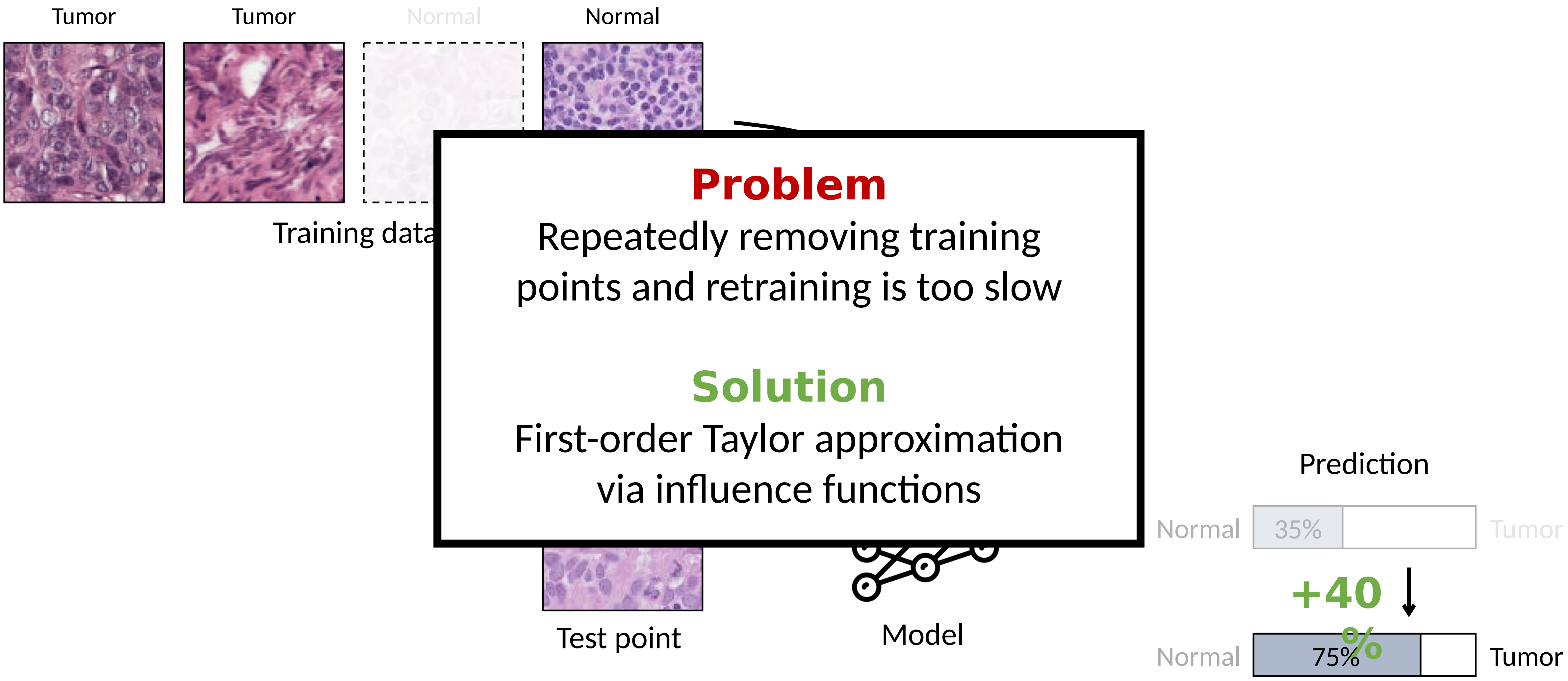


Tumor
Tumor
Normal
Normal
Training data
Training
Test point
Model
Prediction
Normal
35%
Tumor
-4%
Normal
31%
Tumor

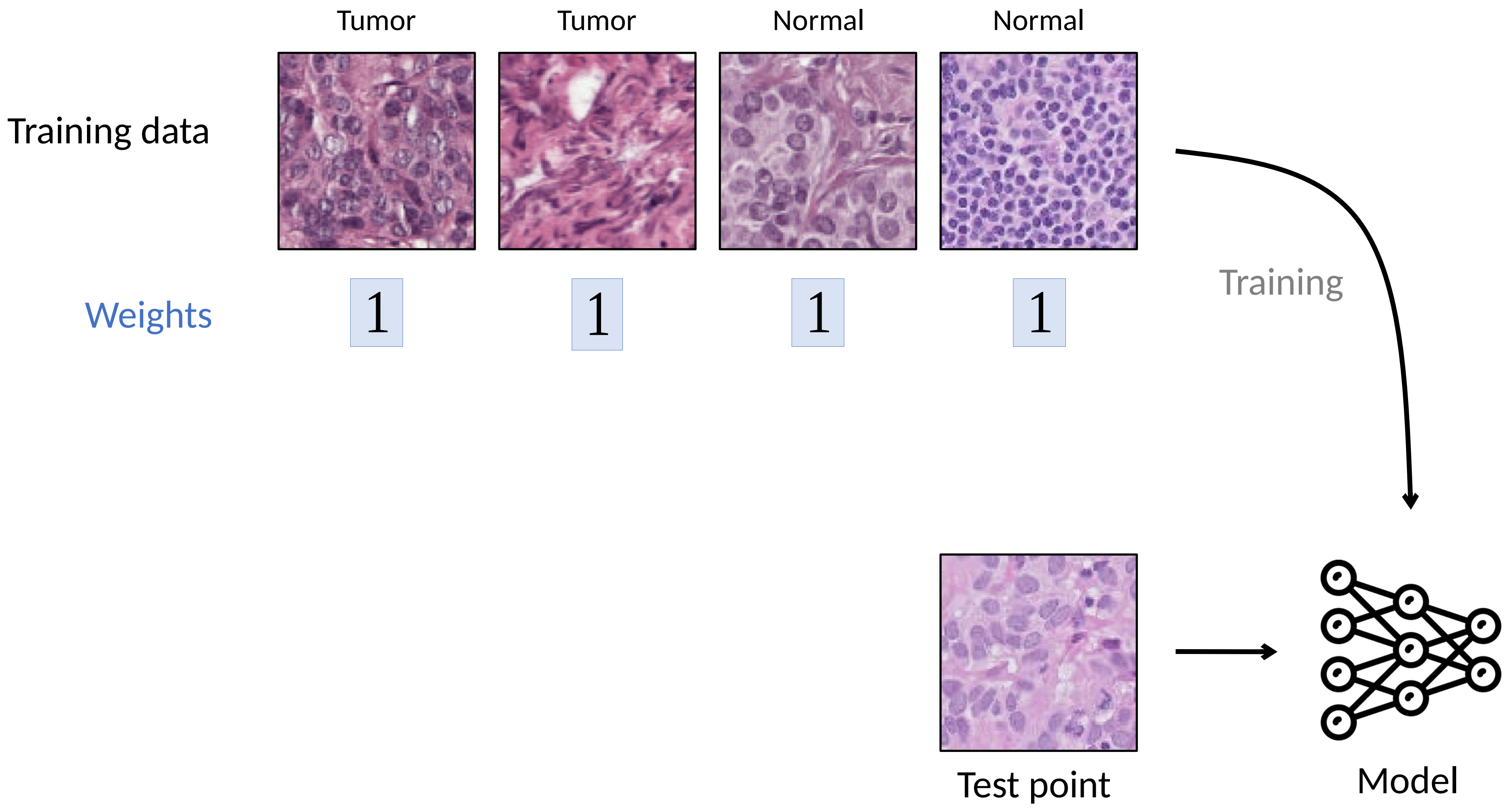


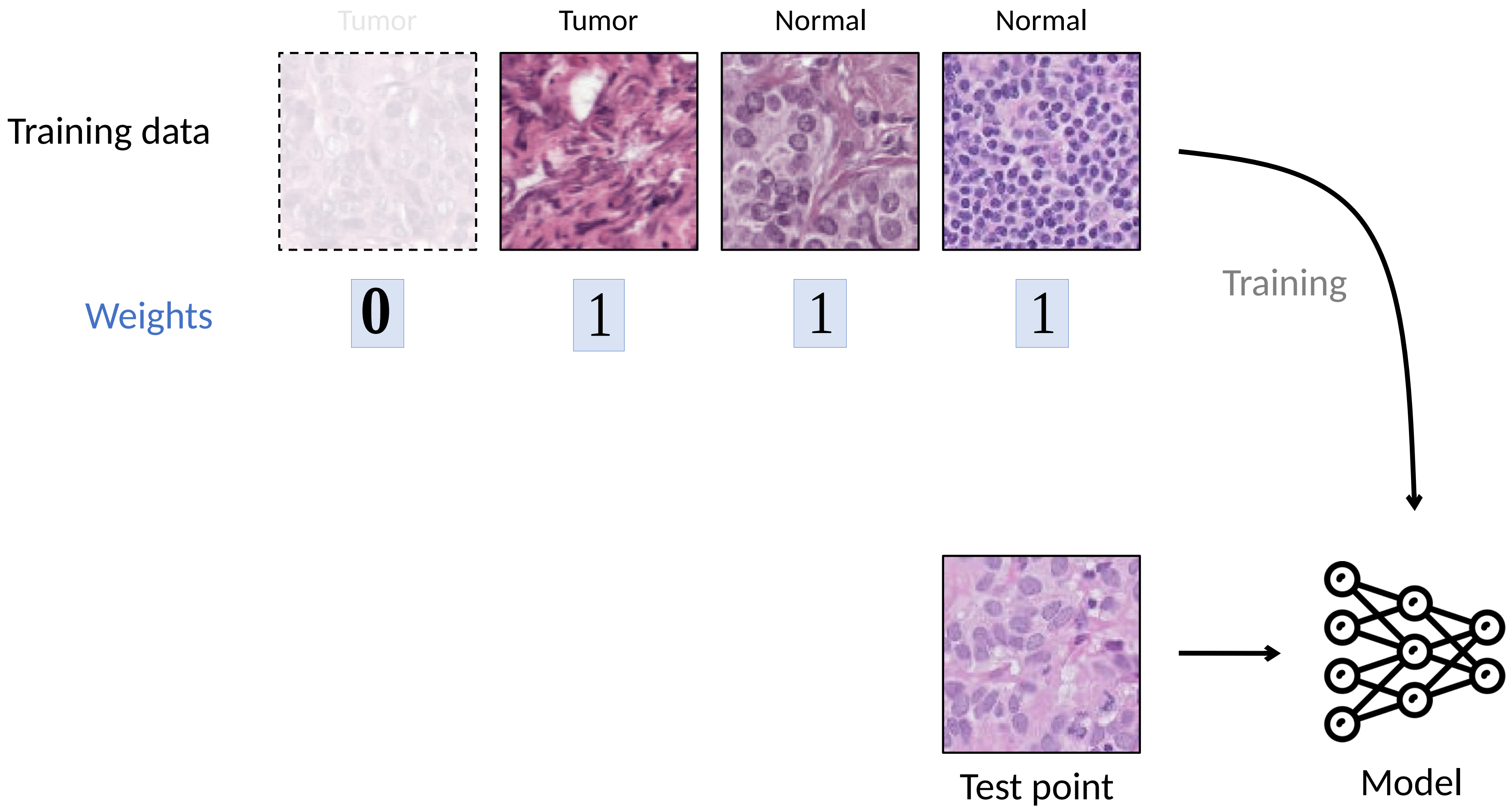
Tumor
Tumor
Normal
Normal
Training data
Training
Test point
Model
Prediction
Normal
35%
Tumor
-2%
Normal
33%
Tumor

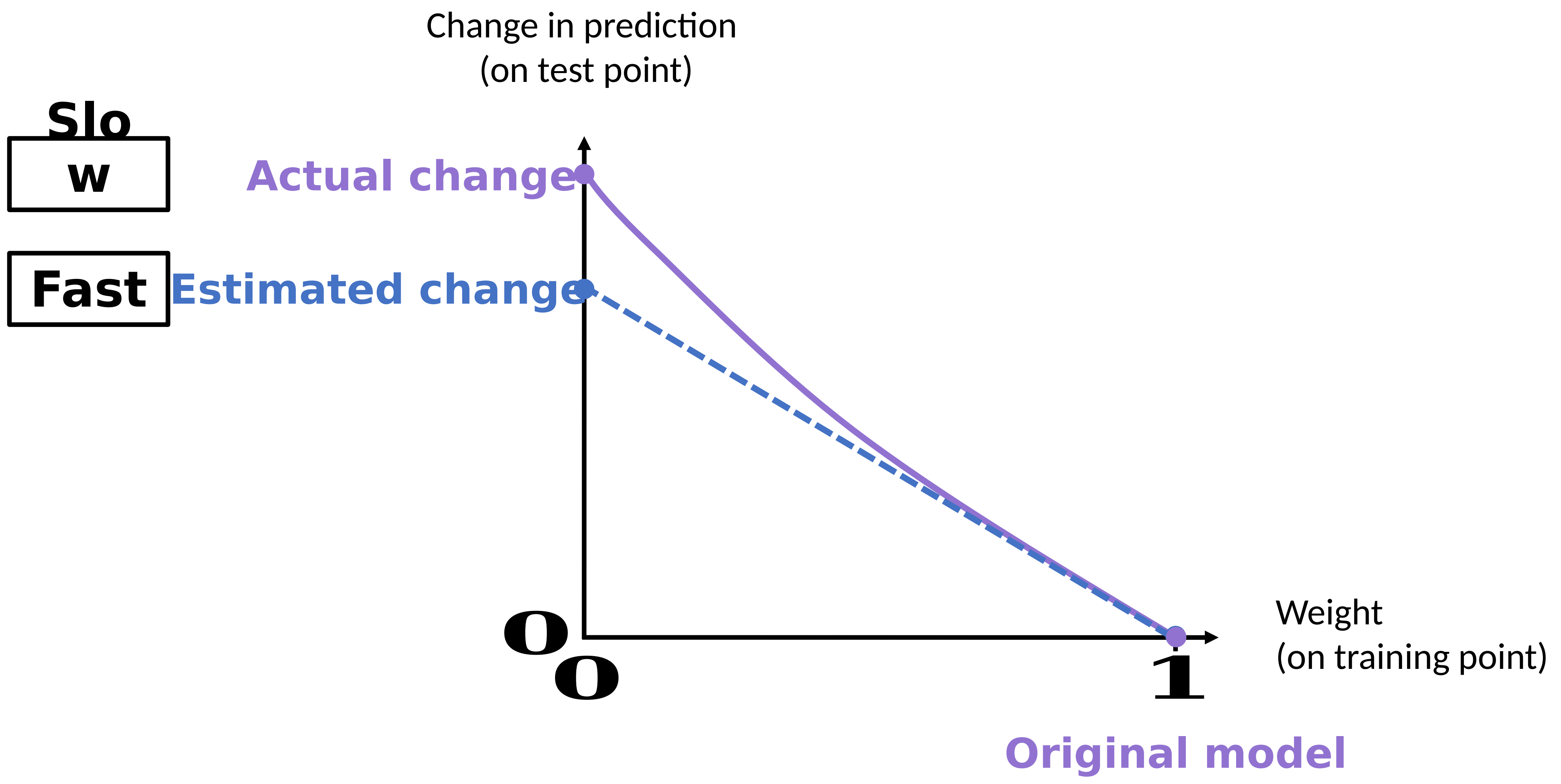

Tumor
Tumor
Normal
Normal
Training data
Training
Test point
Model
Prediction
Normal
35%
Tumor
+40%
Normal
75%
Tumor



Tumor
Tumor
Normal
Normal
Training data
Problem
Repeatedly removing training points and retraining is too slow
Solution
First-order Taylor approximation via influence functions
Prediction
Normal
35%
Tumor
+40%
Normal
75%
Tumor
Test point
Model


Tumor
Tumor
Normal
Normal
Training data
Weights
1
1
1
1
Training
Test point
Model


Tumor
Tumor
Normal
Normal
Training data
Weights
0
1
1
1
Training
Test point
Model


Slo
w
Fast
Change in prediction
(on test point)
Actual change
Estimated change
0
0
1
Weight
(on training point)
Original model

# The influence function approximation

$z_{test}, \hat{\theta}\Big) \approx \nabla$

Loss on
(original)

# The influence function approximation

$$\underbrace{\ell\left(z_{test}, \hat{\theta}_{-z_{train}}\right) - \ell\left(z_{test}, \hat{\theta}\right)}_{\text{Change in loss on  after removing}} \approx \nabla$$

Loss on
(after removing )

Loss on
(original)

# The influence function approximation

$$\overbrace{\ell\left(z_{test}, \hat{\theta}_{-z_{train}}\right) - \ell\left(z_{test}, \hat{\theta}\right)}^{\text{Change in loss on  after removing}} \approx \nabla$$

# The influence function approximation

$$\overbrace{\ell\left(z_{test}, \hat{\theta}_{-z_{train}}\right) - \ell\left(z_{test}, \hat{\theta}\right)}^{\text{Change in loss on after removing}} \approx \nabla_{\theta} \ell\left(z_{test}, \hat{\theta}\right)^{T} H_{\hat{\theta}}^{-1} \nabla_{\theta} \ell\left(z_{train}, \hat{\theta}\right)$$

Gradient of loss on

Gradient of loss on

Inverse of the Hessian

# The influence function approximation

Doesn't require retraining!

Change in loss on  after removing

$$\underbrace{\ell\left(z_{test}, \hat{\theta}_{-z_{train}}\right) - \ell\left(z_{test}, \hat{\theta}\right)} \approx \nabla_\theta \ell\left(z_{test}, \hat{\theta}\right)^T H_{\hat{\theta}}^{-1} \nabla_\theta \ell\left(z_{train}, \hat{\theta}\right)$$

Gradient of loss on

Gradient of loss on

Inverse of the Hessian

# The influence function approximation

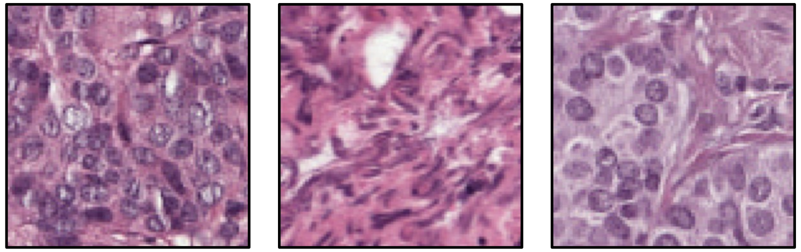
$$\overbrace{\ell\left(z_{test},\hat{\theta}_{-z_{train}}\right)-\ell\left(z_{test},\hat{\theta}\right)}^{\text{Change in loss on after removing}}\approx\nabla_{\theta}\overbrace{\underbrace{\ell\left(z_{test},\hat{\theta}\right)^{T}}_{\text{Gradient of loss on}}}^{\text{Model's representation of}}\overbrace{\underbrace{H_{\hat{\theta}}^{-1}}_{\text{Inverse of the Hessian}}}^{\text{Effect of other training points}}\overbrace{\underbrace{\nabla_{\theta}\,\ell\left(z_{train},\hat{\theta}\right)}_{\text{Gradient of loss on}}}^{\text{Model's representation of}}$$

# A little bit more technical details …

Training data $\longrightarrow \hat{\Theta} \longrightarrow \ell\left(z_{test}, \hat{\theta}\right)$

Tumor | Tumor | Normal

Training data



Weights: $+\epsilon$ | $1$ | $1$

$Z$

$$\hat{\theta}_{\epsilon,z} \stackrel{\text{def}}{=} \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

# A little bit more technical details …

Training data $\longrightarrow \widehat{\Theta} \longrightarrow \ell\left(z_{test}, \widehat{\theta}\right)$

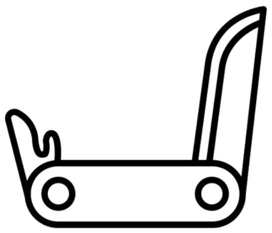
$$\left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0}$$

$$\left.\frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\right|_{\epsilon=0}$$

$$= \nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top \left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0}$$

$$\hat{\theta}_{\epsilon,z} \stackrel{\text{def}}{=} \arg\min_{\theta\in\Theta} \tfrac{1}{n}\textstyle\sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

# From classical to modern settings



Jaeckel, 1972. The infinitesimal jackknife.

Hampel, 1974. The influence curve and its role in robust estimation.

Cook, 1977. Detection of influential observations in linear regression.

...

# From classical to modern settings

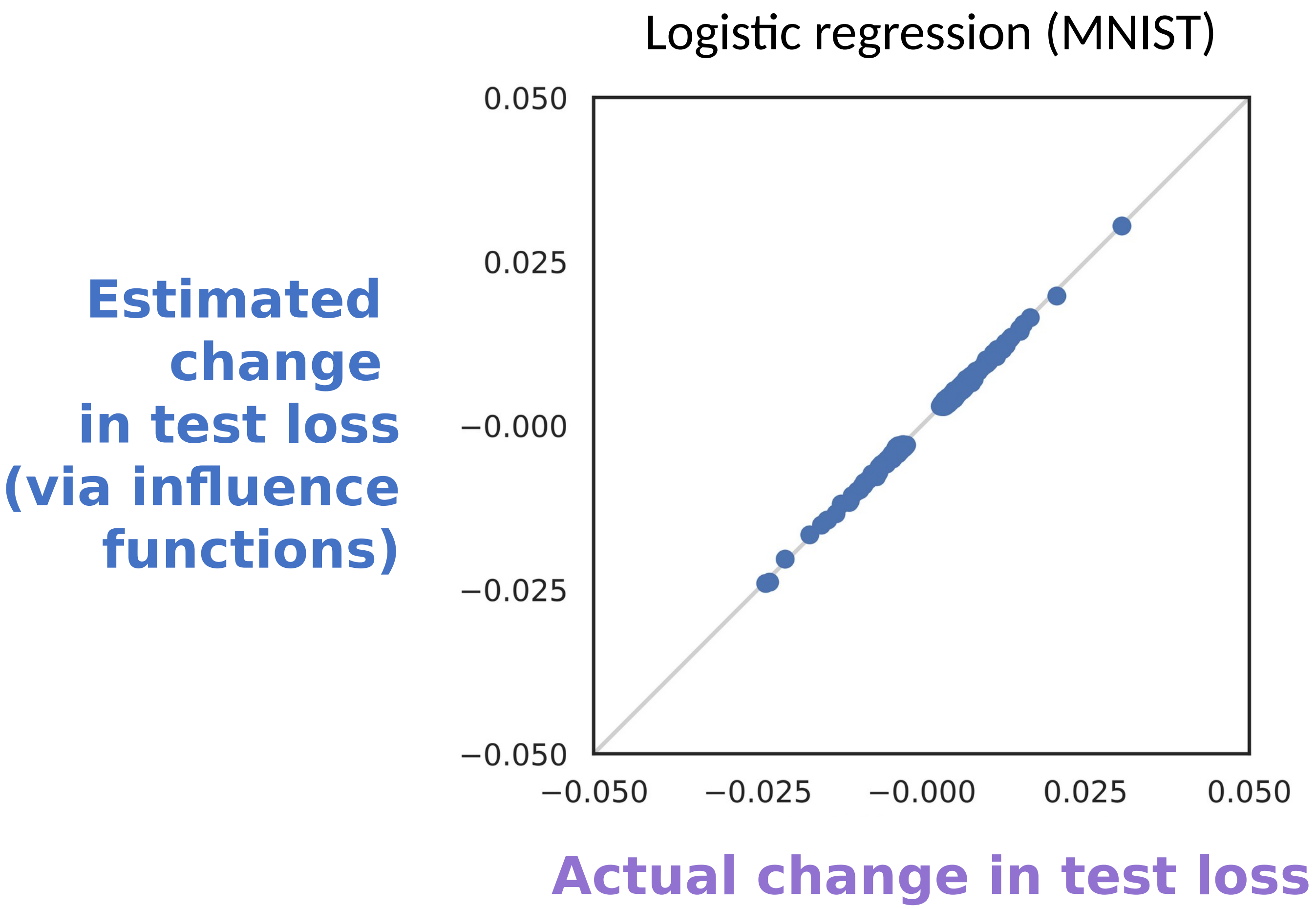
Small datasets → Large datasets
Low-dimensional → High-dimensional

Difficult to compute

$$\nabla_\theta \ell\left(z_{test}, \hat{\theta}\right)^T \boldsymbol{H}_{\hat{\boldsymbol{\theta}}}^{-1} \nabla_\theta \ell\left(z_{train}, \hat{\theta}\right)$$

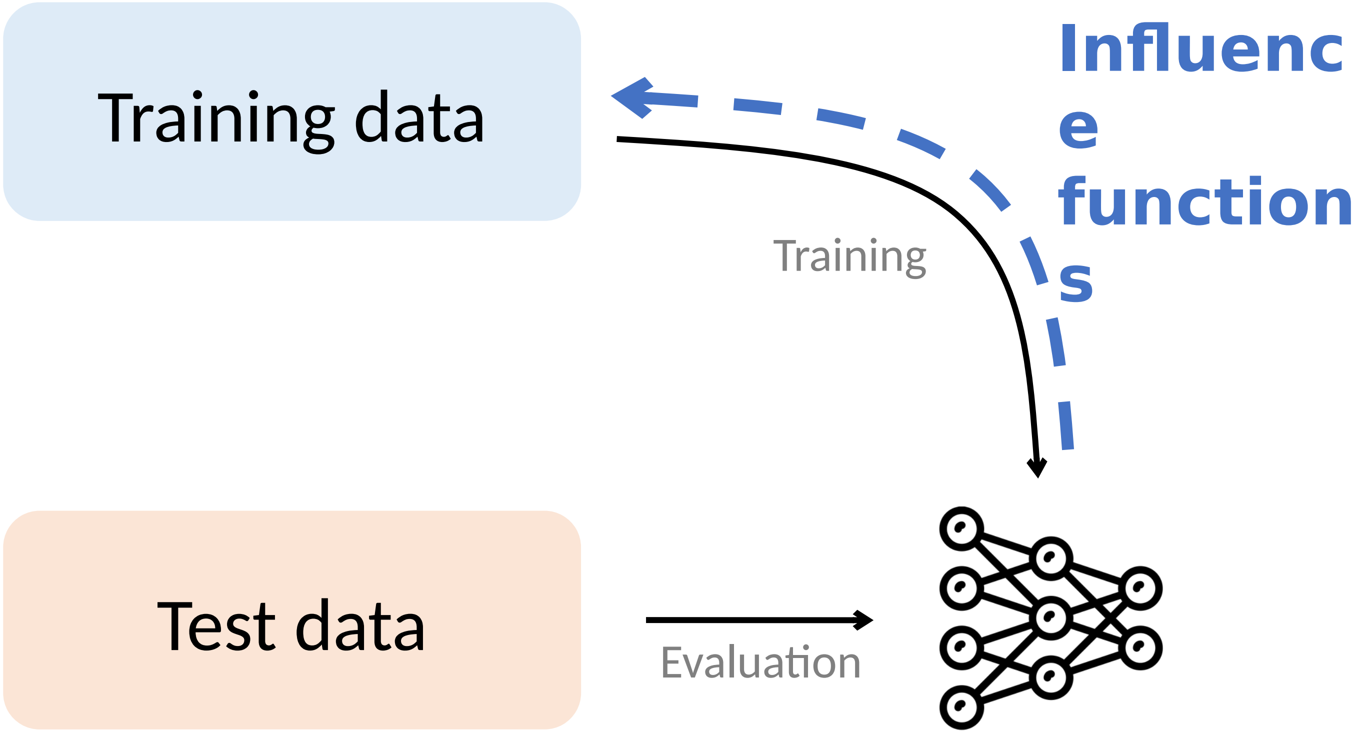
We use tools from
2nd-order optimization & stochastic estimation
[Pearlmutter, 1994; Martens, 2010, Agarwal et al., 2017]

# Removing single points




Each point represents removing one training example

# Understanding Models via Their Training Data




**Where can you apply influence functions?**

# Impact

## Robustness

Training set biases
[Ren et al., 2018]

Inference reliability
[Broderick et al., 2021]

Cross-validation
[Stephenson et al., 2020]

Memorization
[Feldman, 2019]

## Applications

Data distillation
[Wang et al., 2020]

Data valuation
[Jia et al., 2019]

Active learning
[Gudovskiy et al., 2020]

Data debugging
[Guo et al., 2021]

## Fairness & security

Algorithmic bias
[Verma et al., 2021]

Data labor
[Arrieta-Ibarra, 2018]

Privacy
[Shokri et al., 2021]

Data poisoning
[Chen et al., 2017]

# Limitations

- What assumptions on the models are needed to calculate influence functions?
- Can you calculate influence functions for a neural network with the recipe described earlier?

$$\left.\frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\right|_{\epsilon=0}$$

Training data $\longrightarrow \widehat{\Theta} \longrightarrow \ell\left(z_{test}, \widehat{\Theta}\right)$

$$\hat{\theta}_{\epsilon,z} \stackrel{\text{def}}{=} \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} L(z_i,\theta) + \epsilon L(z,\theta)$$

# Limitations

- What assumptions on the models are needed to calculate influence functions?
- Can you calculate influence functions for a neural network with the recipe described earlier?

If Influence Functions are the Answer, Then What is the Question?

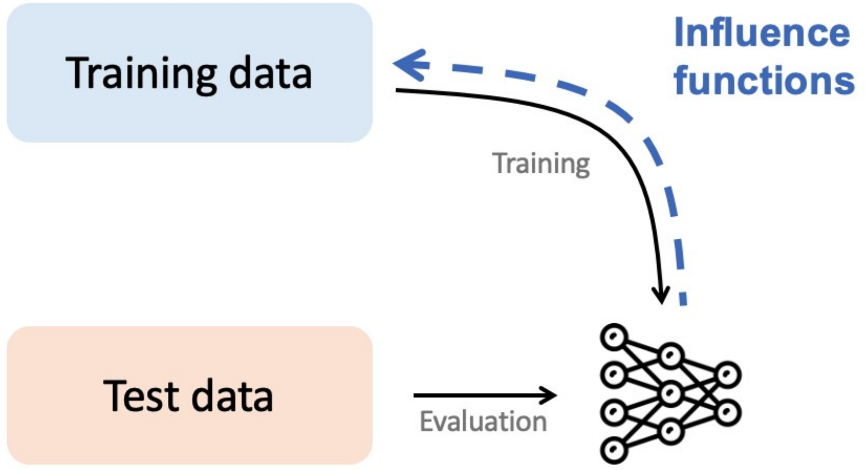
Juhan Bae[*†], Nathan Ng[†‡], Alston Lo[†], Marzyeh Ghassemi[‡], Roger Grosse[†]

Non-convexity Non-convergence

$$\hat{\theta}_{\epsilon,z} \stackrel{\text{def}}{=} \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} L(z_i,\theta) + \epsilon L(z,\theta)$$

# Summary

- Link model behavior to training data
- Efficient to calculate
- Many applications




- Limitations when applying to complex models