Ensemble Learning

CS4780/5780 – Machine Learning
Fall 2013

Igor Labutov
Cornell University

Ensemble Learning

A class of “meta” learning algorithms

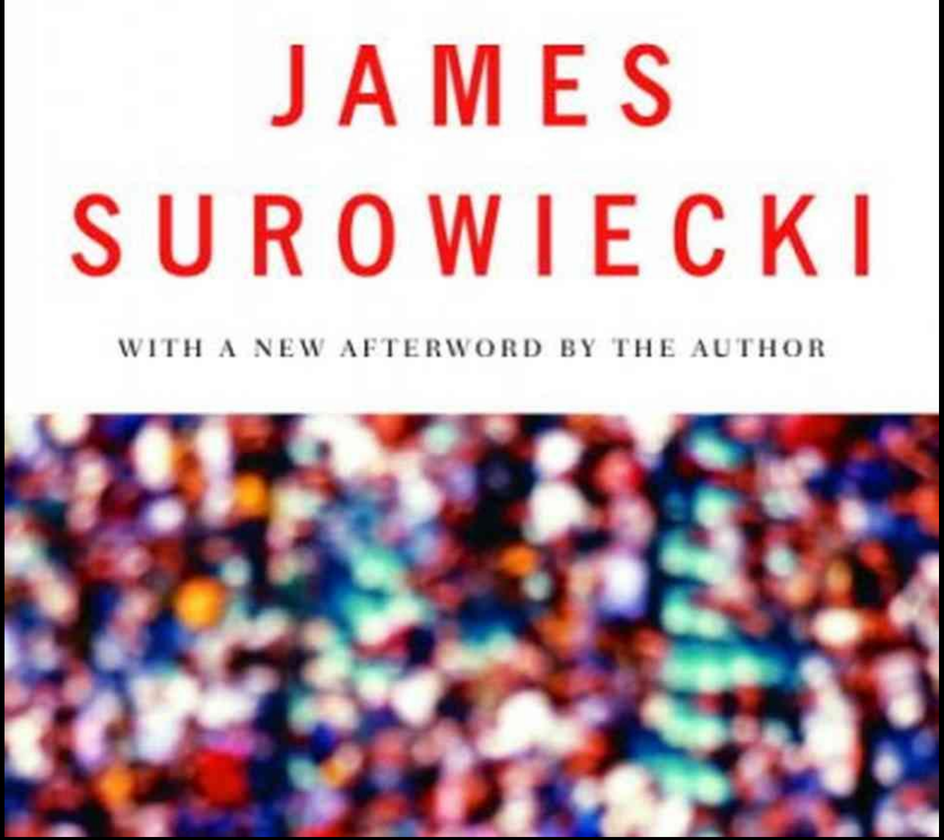
A *NEW YORK TIMES* BUSINESS BESTSELLER

"As entertaining and thought-provoking as *The Tipping Point* by Malcolm Gladwell. . . . *The Wisdom of Crowds* ranges far and wide."
—*The Boston Globe*

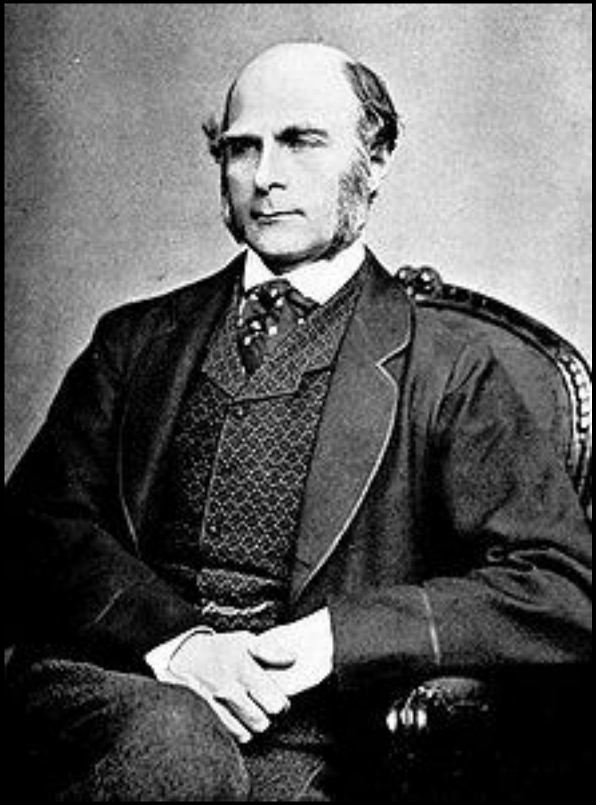
THE WISDOM OF CROWDS

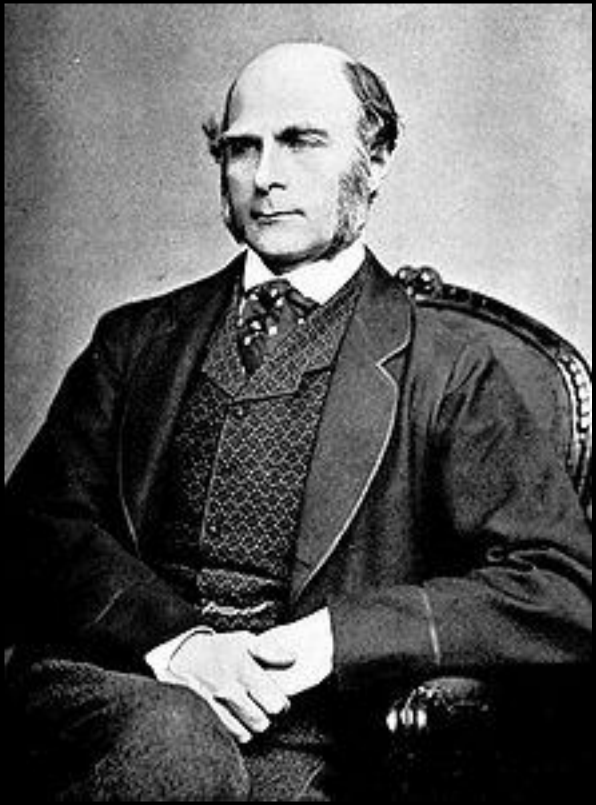
JAMES SUROWIECKI

WITH A NEW AFTERWORD BY THE AUTHOR



"The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations"





1,198 lb 1,197 lb

Criteria	Description
Diversity of opinion	*Each person should have private information even if it's just an eccentric interpretation of the known facts.*
Independence	*People's opinions aren't determined by the opinions of those around them.*
Decentralization	*People are able to specialize and draw on local knowledge.*
Aggregation	*Some mechanism exists for turning private judgments into a collective decision.*

Ensemble Learning

A class of “meta” learning algorithms

Combining multiple classifiers to increase performance

Very effective in practice

Good theoretical guarantees

Easy to implement!

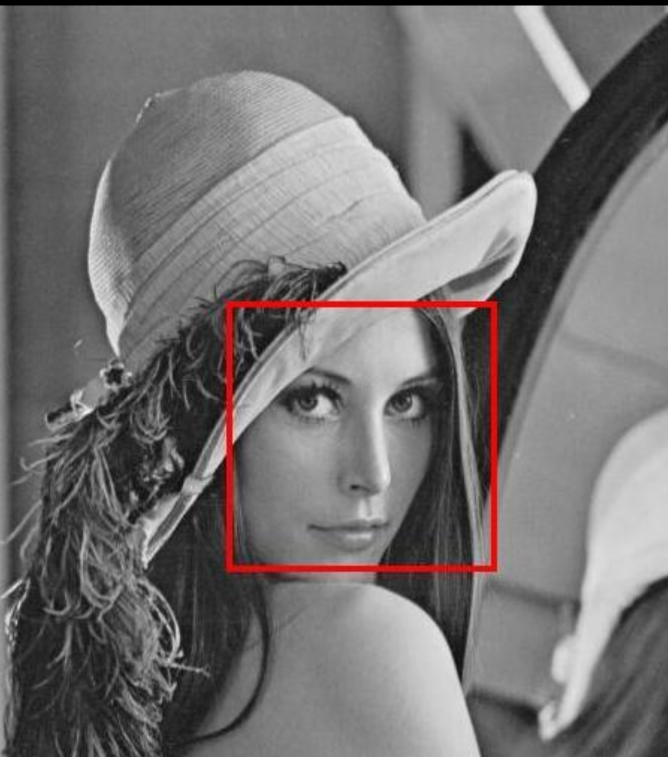
Ensemble

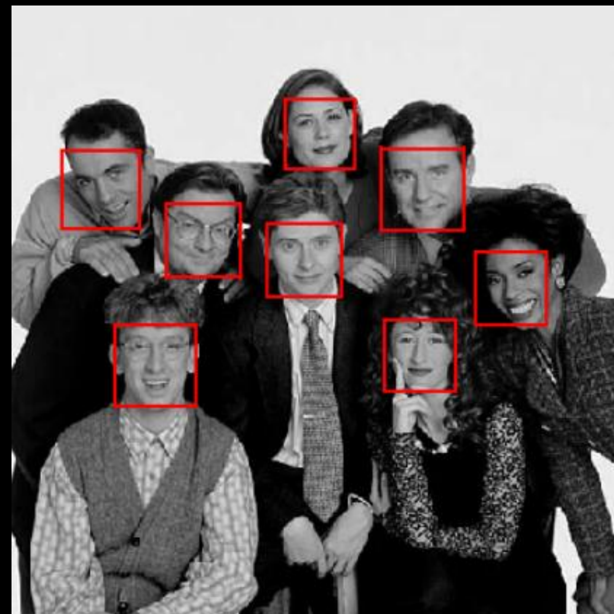
Problem : given *T* binary classification hypotheses ($h_1, \ldots, h_T$), **find** a combined classifier:

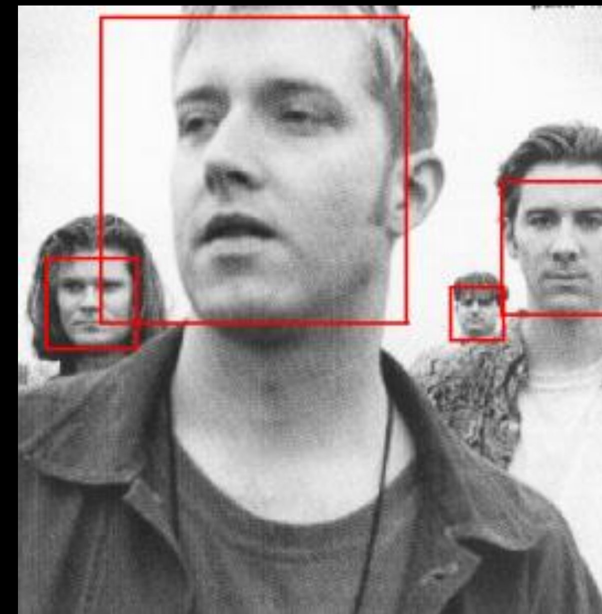
$$h_S(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

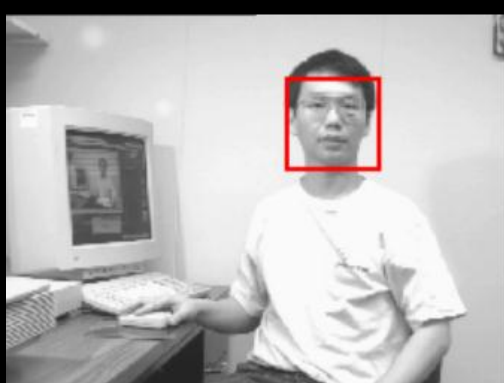
with better performance.

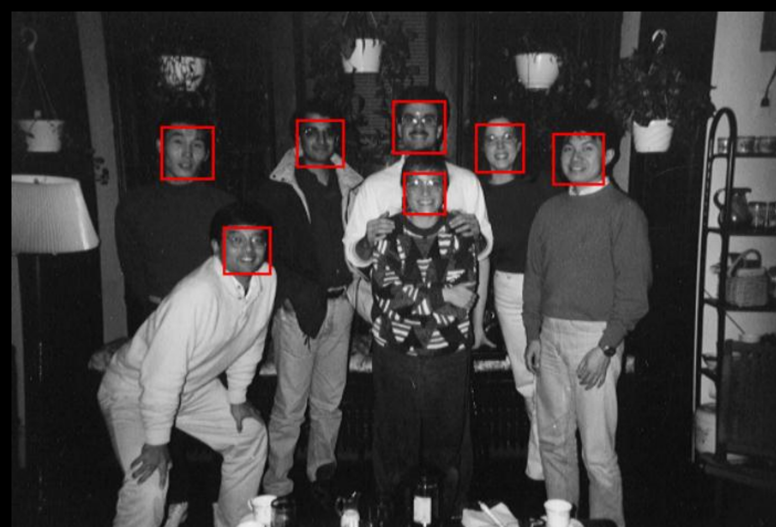
Teaser



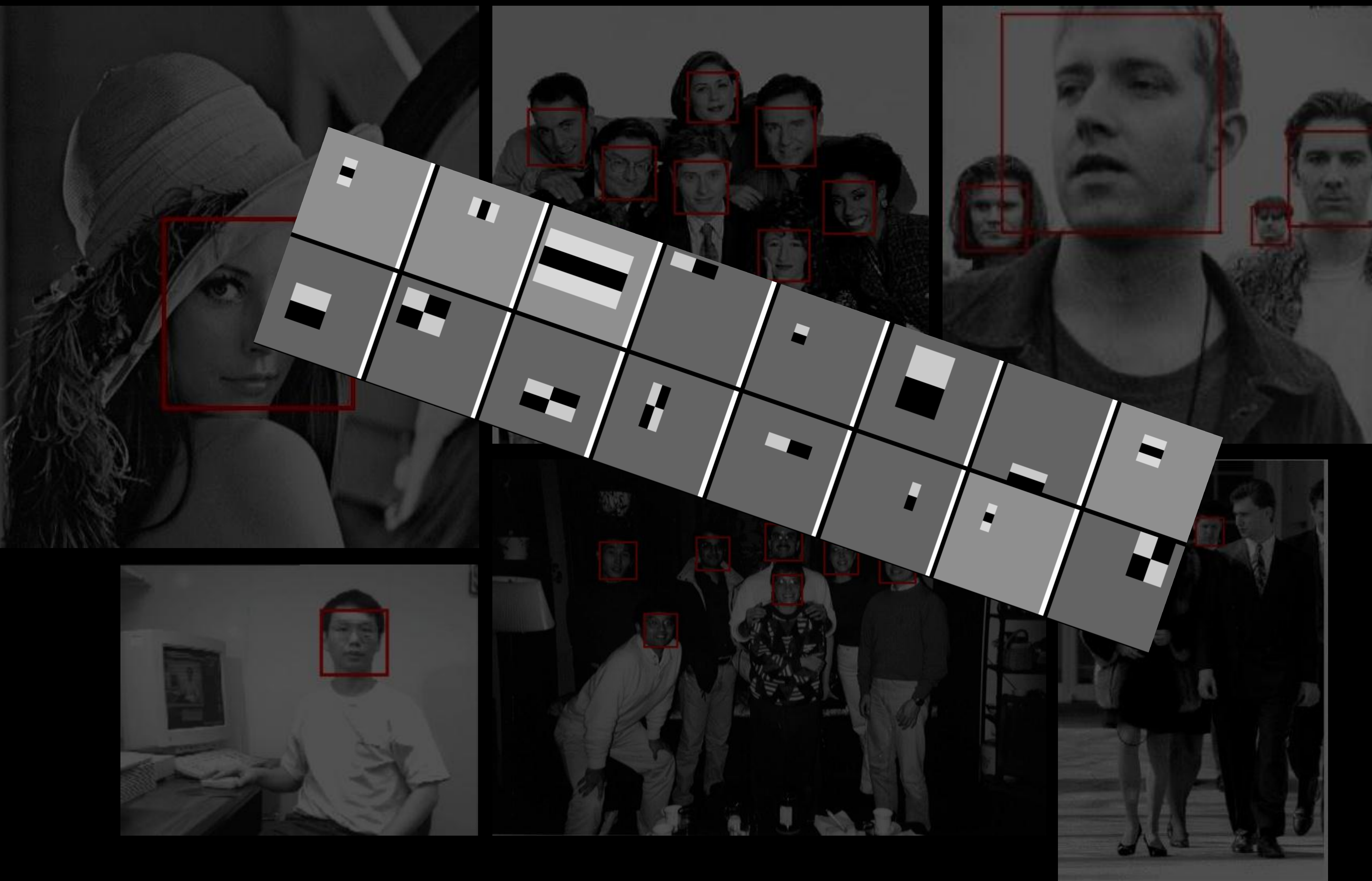








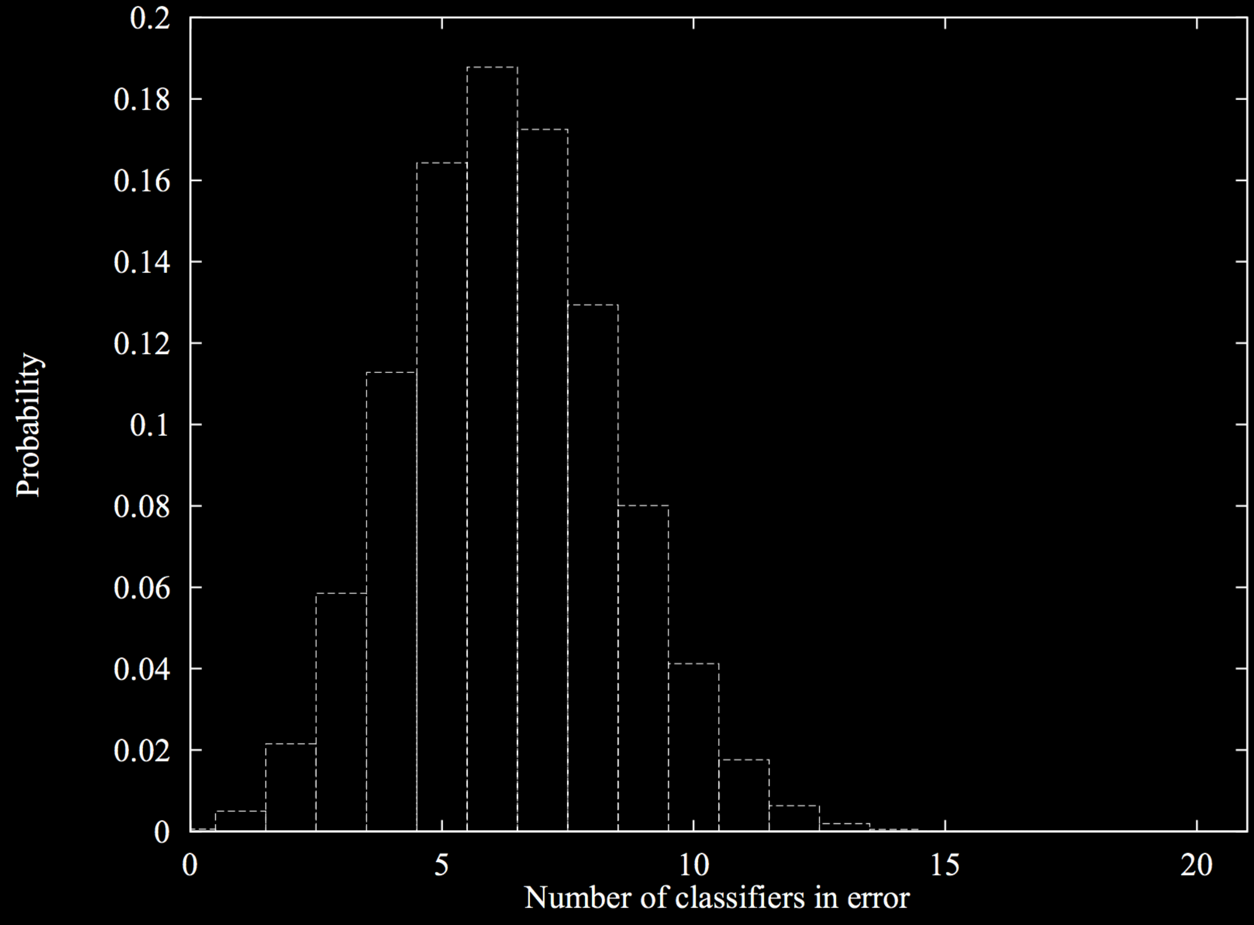
Teaser



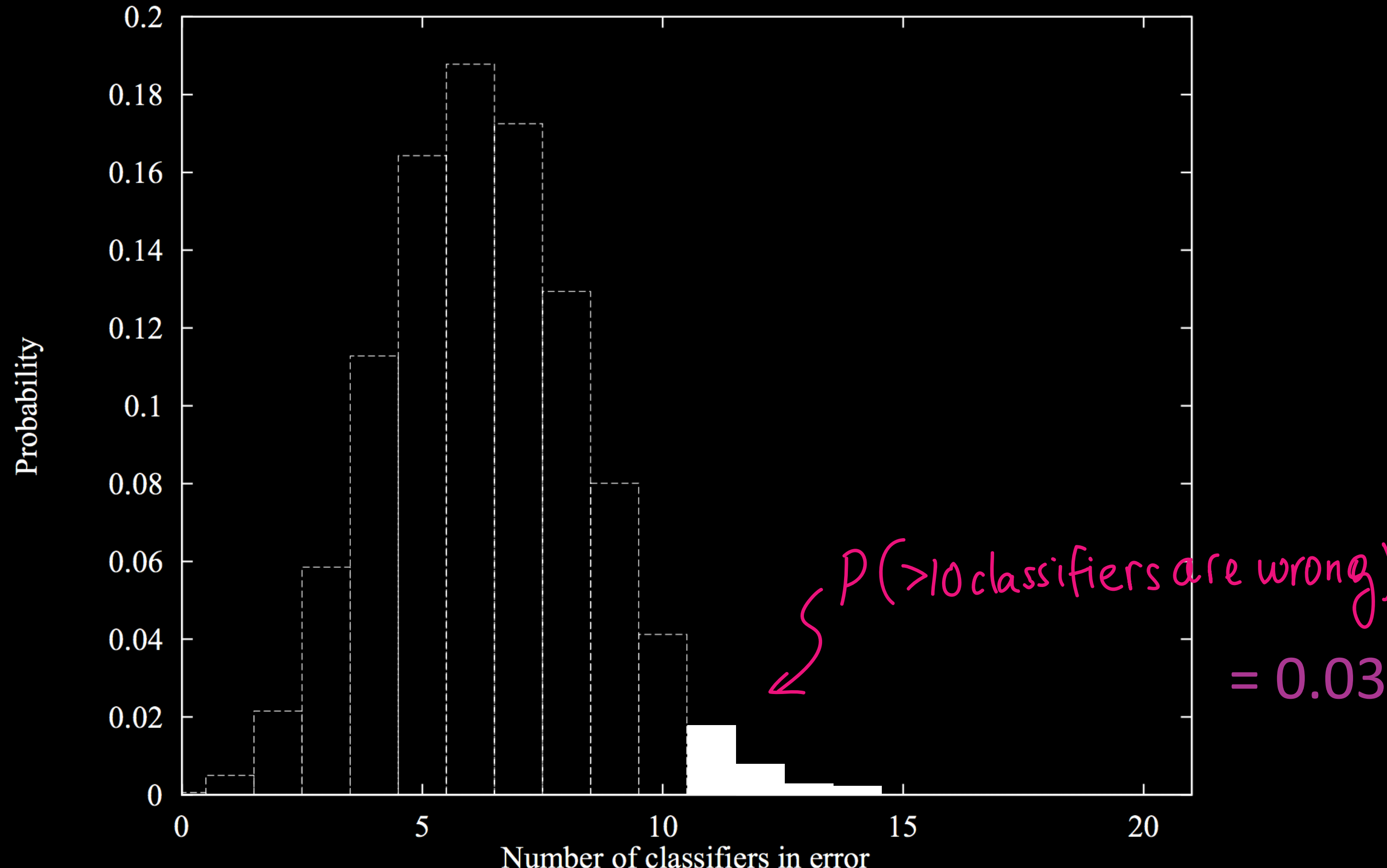
Why do Ensembles work?

Hypothetical Classifier with $P_{error} = 0.3$

Hypothetical Classifier with $P_{error} = 0.3$

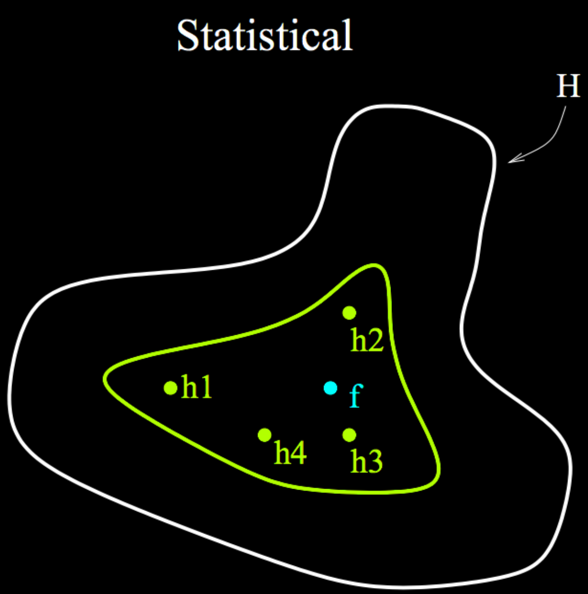


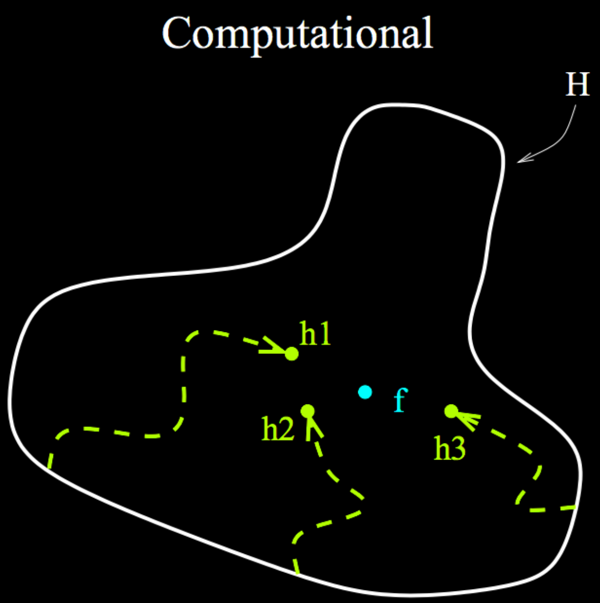
Hypothetical Classifier with $P_{error} = 0.3$

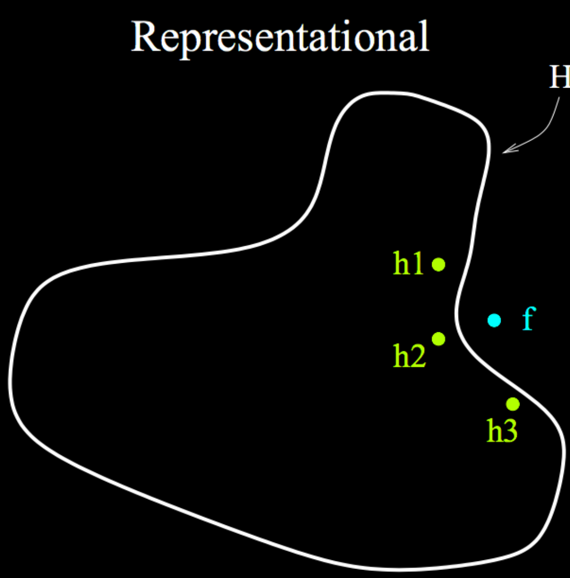


= 0.0384

Why do Ensembles work?







BAGGING

Bagging

(Breiman, 1996)

Bagging (Boostrap aggregating).

$\textsc{Bagging}(S = ((x_1, y_1), \ldots, (x_m, y_m)))$

for $t \leftarrow 1$ **to** T **do**

  $S_t \leftarrow \textsc{Bootstrap}(S) \triangleright$ i.i.d. sampling with replacement from S.

  $h_t \leftarrow \textsc{TrainClassifier}(S_t)$

return $h_S = x \mapsto \textsc{MajorityVote}((h_1(x), \ldots, h_T(x)))$

Bagging

Ensemble :

$$h_S(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Bagging : Special case where we fix:

$$\alpha_t = 1 \quad \text{and} \quad h_t = \mathbb{L}(S_t)^{*}$$

* $\mathbb{L}$ is some learning algorithm

S_t is a training set drawn from distribution $P(< x, y >)$

Bias-Variance Tradeoff

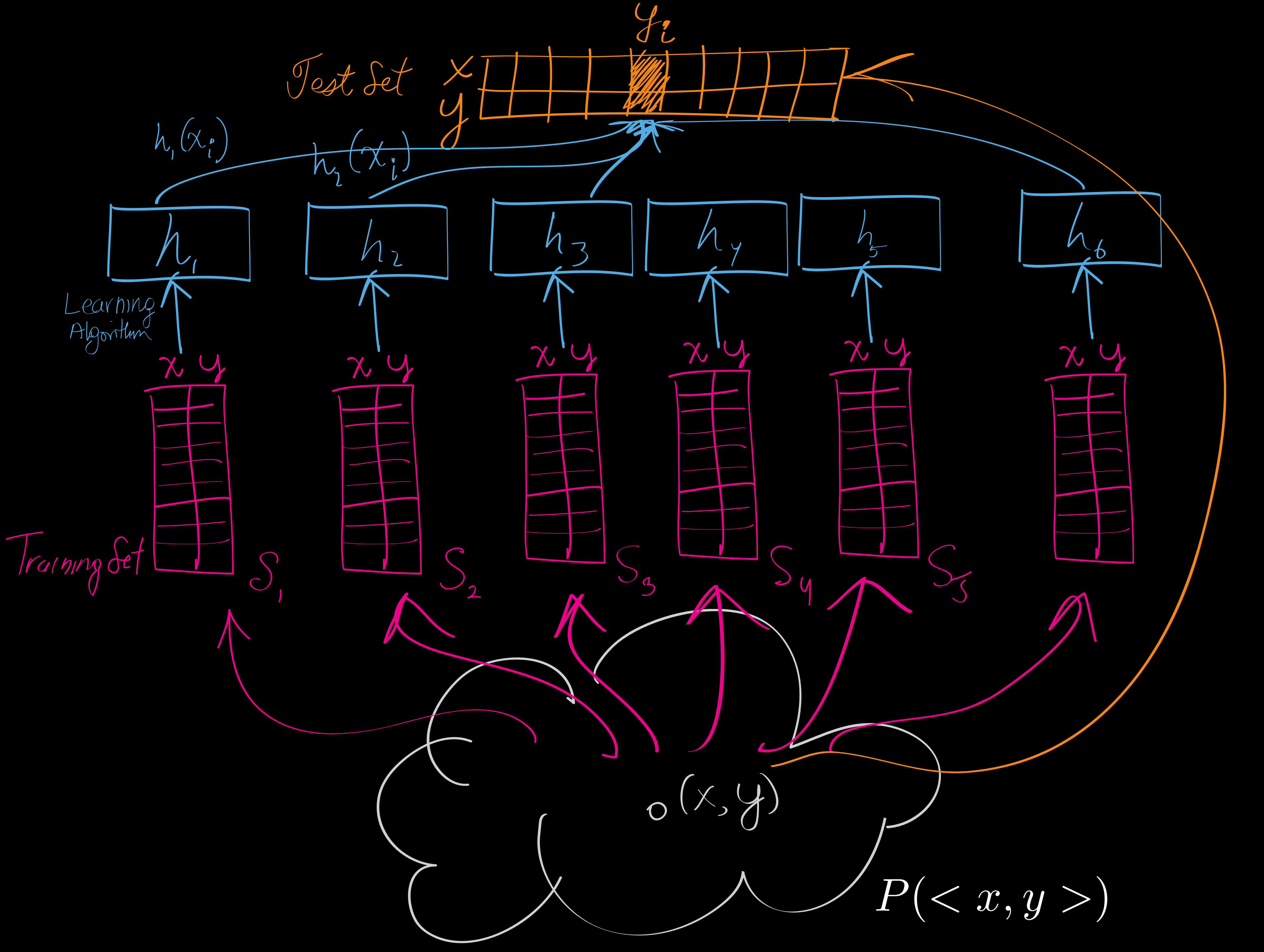
Generalization Error

Classification :

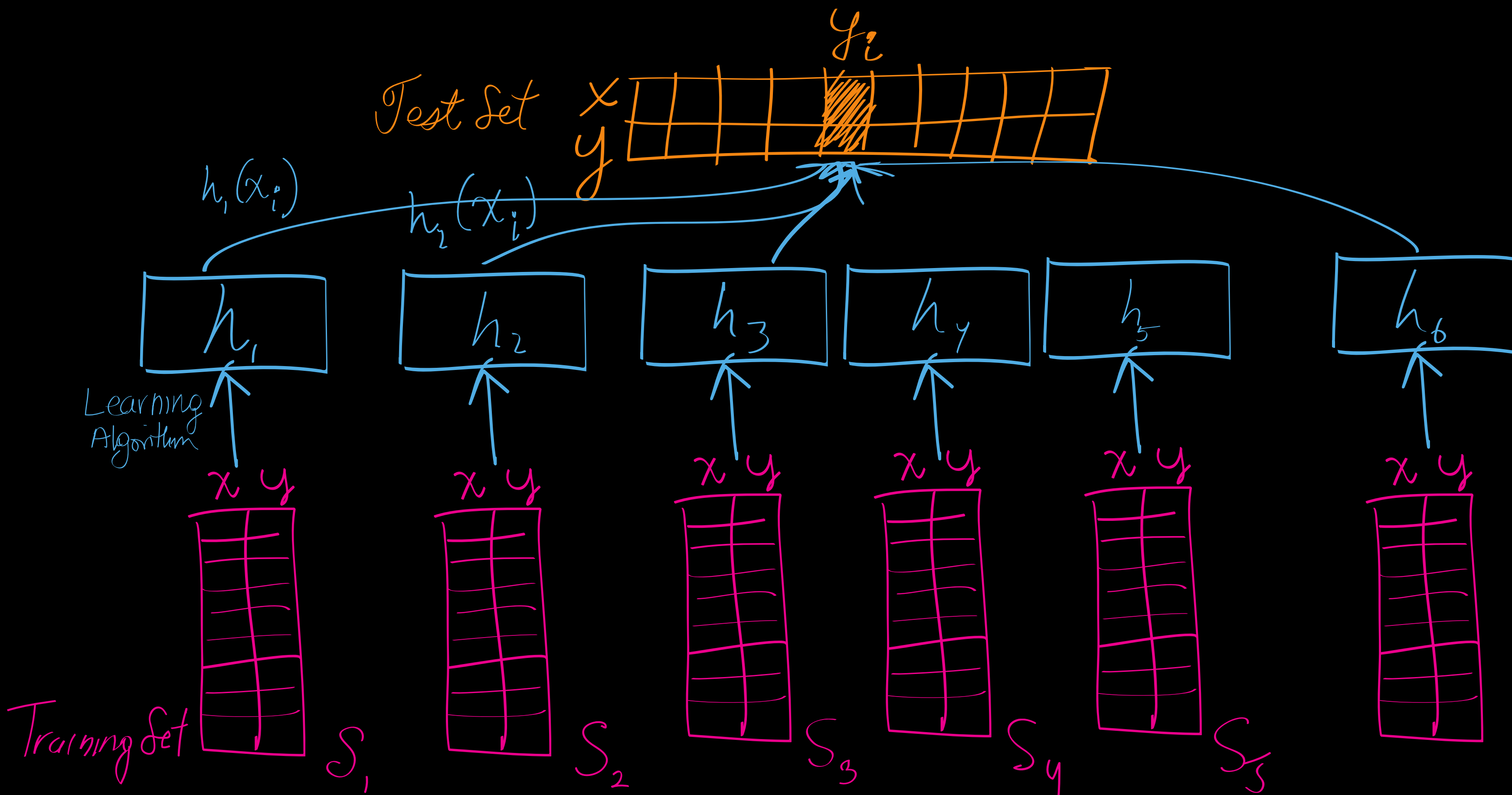
$$\epsilon_{test} = \frac{1}{n}\sum_{i}^{n} \text{Zero-One-Loss}(y_i, h(x_i))$$

Regression :

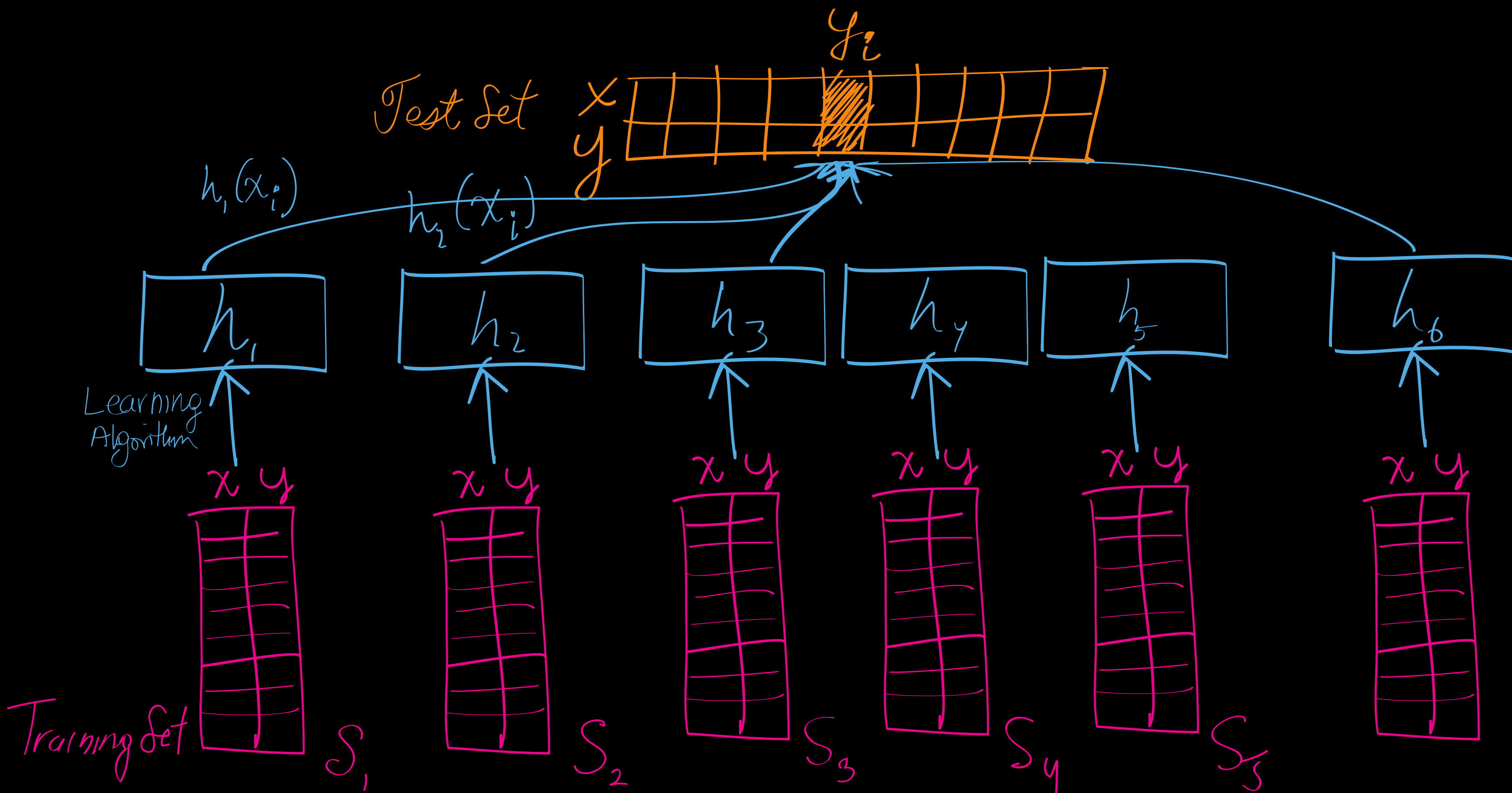
$$\epsilon_{test} = \frac{1}{n}\sum_{i}^{n} (y_i - h(x_i))^2$$



Test Set
y_i
x
y
$h_1(x_i)$
$h_2(x_i)$
h_1
h_2
h_3
h_4
h_5
h_6
Learning Algorithm
x y
Training Set
S_1
S_2
S_3
S_4
S_5
(x, y)
$P(< x, y >)$

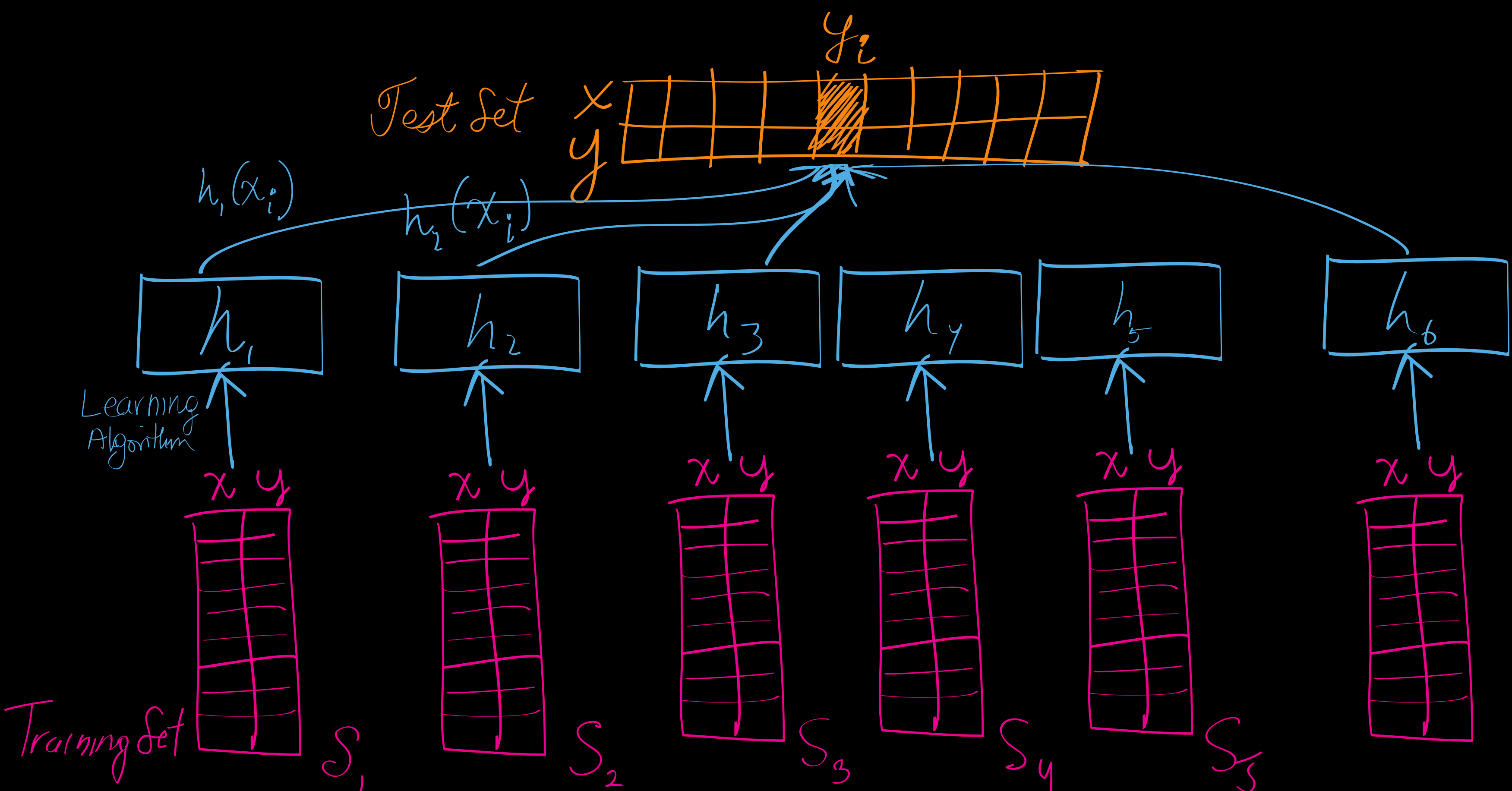


Test Set
x
y
y_i
$h_1(x_i)$
$h_2(x_i)$
h_1
h_2
h_3
h_4
h_5
h_6
Learning Algorithm
x y
Training Set
S_1
S_2
S_3
S_4
S_5



Test Set
x
y
y_i
$h_1(x_i)$
$h_2(x_i)$
h_1
h_2
h_3
h_4
h_5
h_6
Learning Algorithm
x y
Training Set
S_1
S_2
S_3
S_4
S_5

$$\bar{\epsilon}_{test}(x_i) = \frac{1}{T}\sum_{t}^{T}(y_i - h_t(x_i))^2$$



$$\bar{\epsilon}_{test}(x_i) = \frac{1}{T}\sum_{t}^{T}(y_i - h_t(x_i))^2$$

OR, as an expectation:

$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right]$$

For the entire test set:

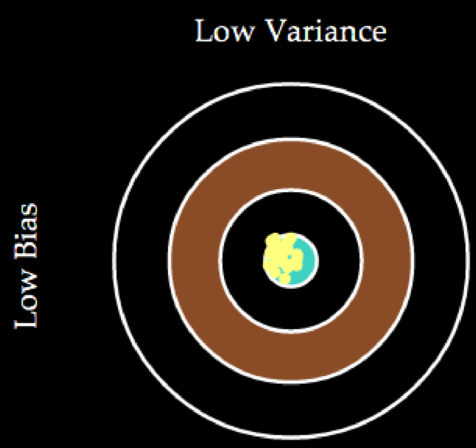
$$\mathbb{E}_{X,Y}\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right]$$

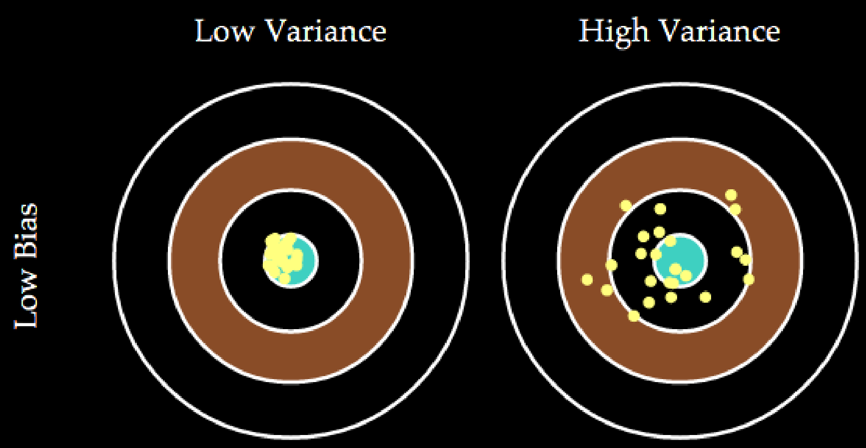
CLAIM:

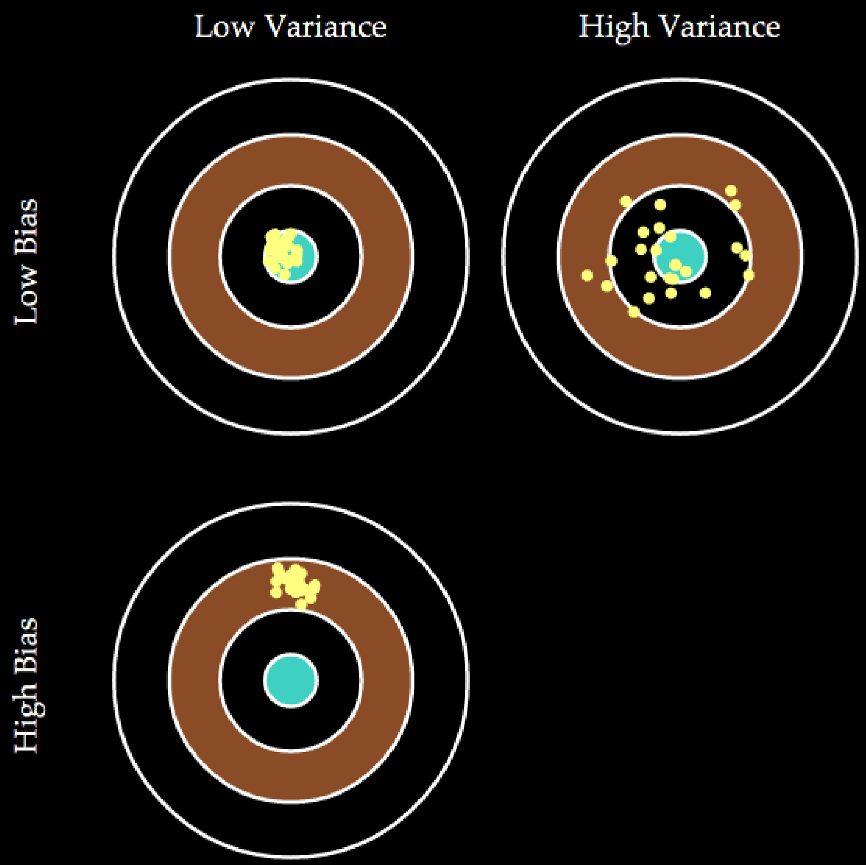
$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right] =$$

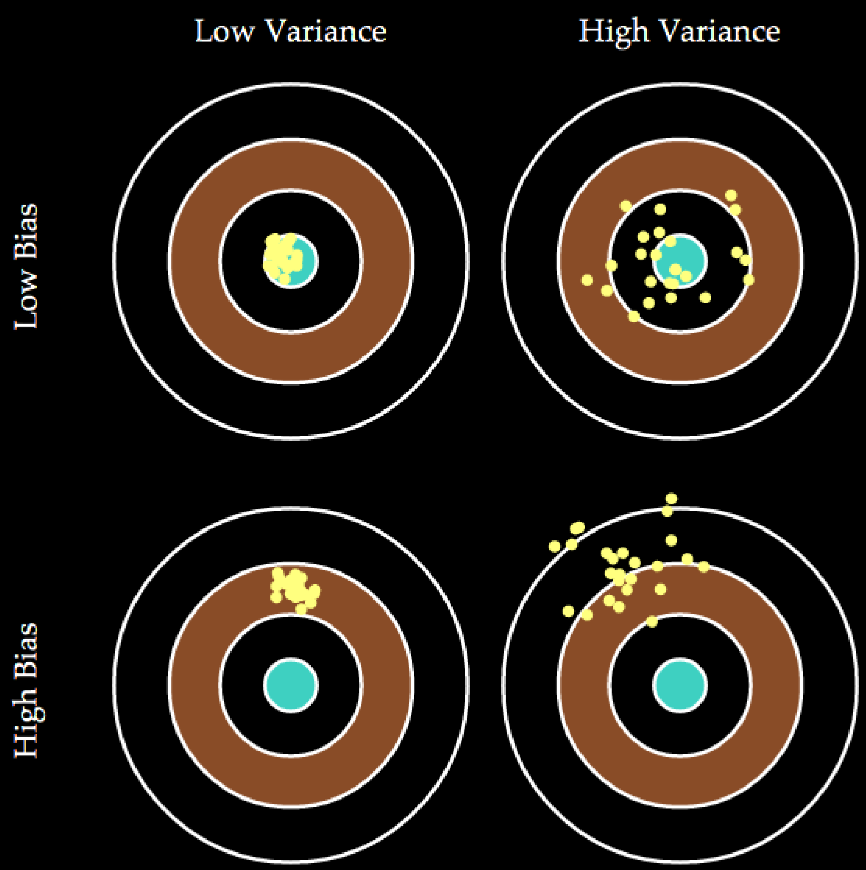
bias² $$(y_i - \mathbb{E}_S[h_S(x_i)])^2 \, +$$

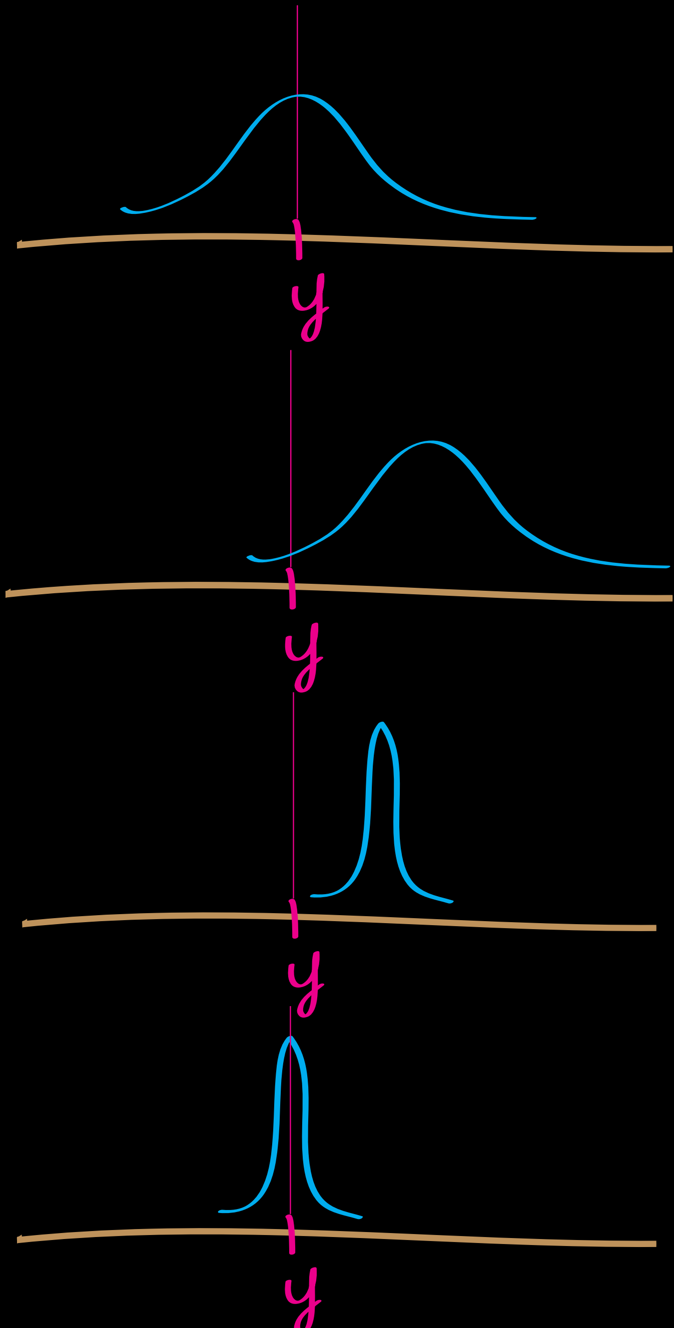
variance $$+ \, \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$$


Low Variance
Low Bias


Low Variance
High Variance
Low Bias


Low Variance
High Variance
Low Bias
High Bias


Low Variance
High Variance
Low Bias
High Bias

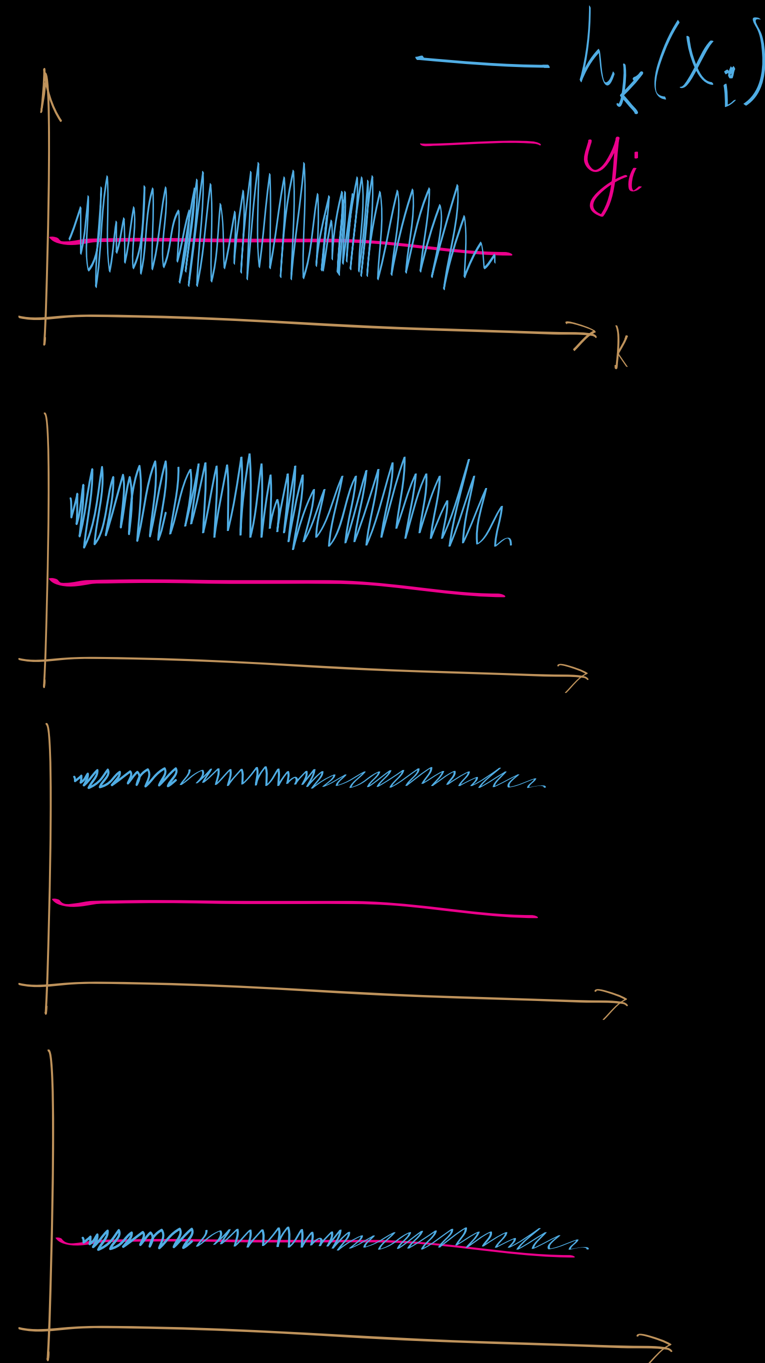


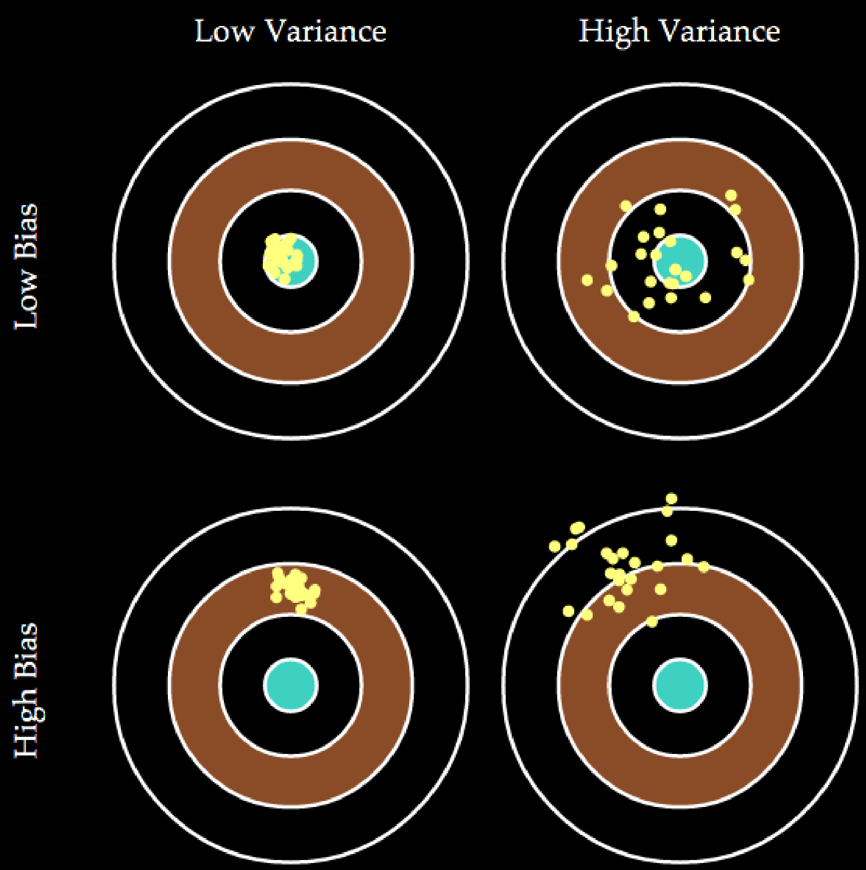
Low Bias
High Variance

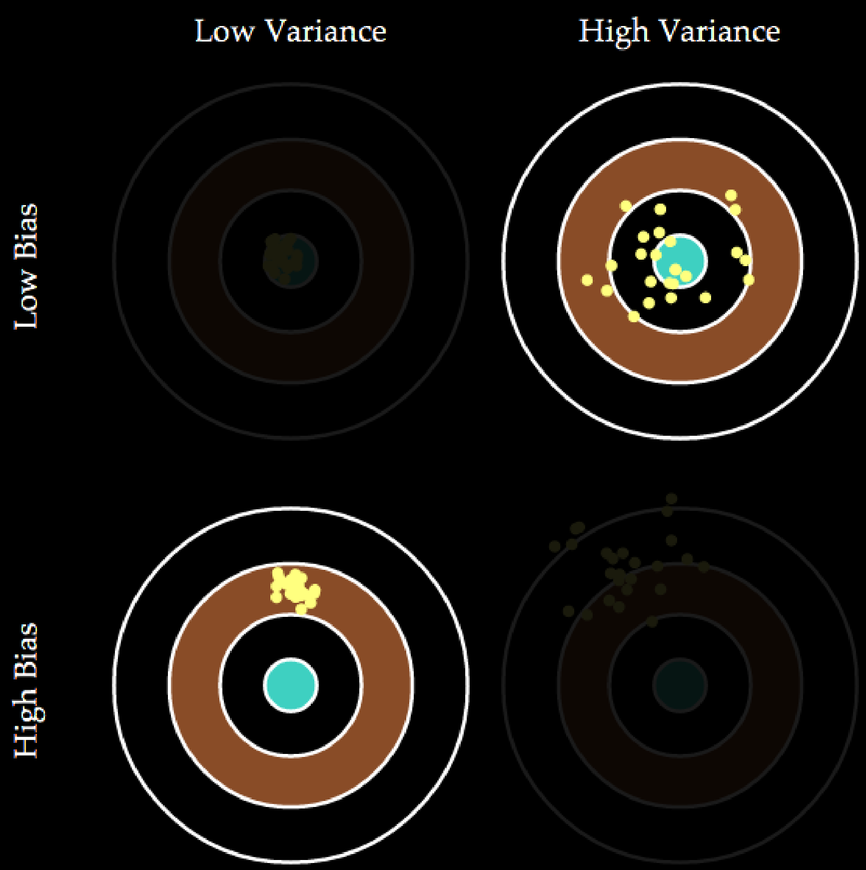
High Bias
High Variance

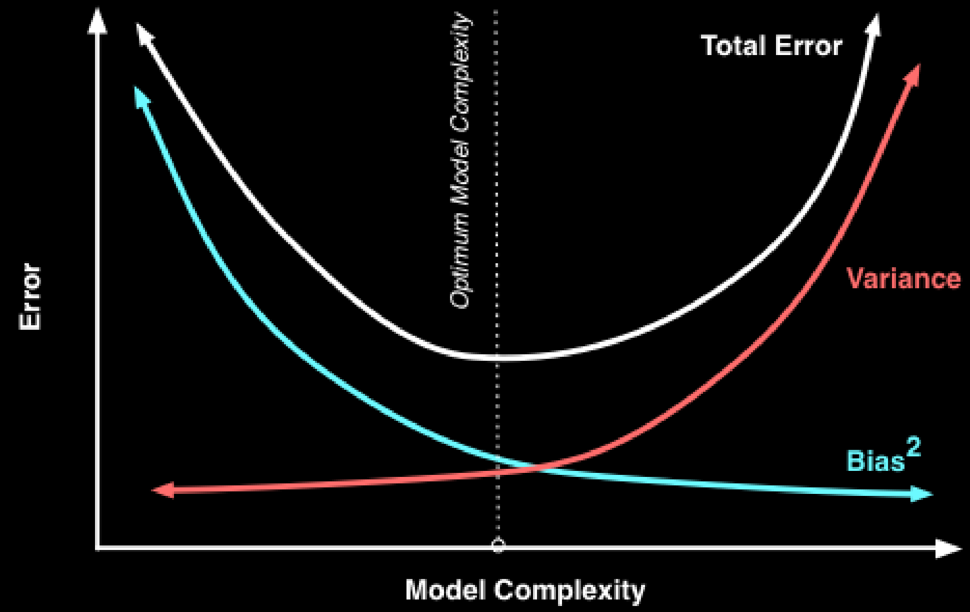
High Bias
Low Variance

Low Bias
Low Variance

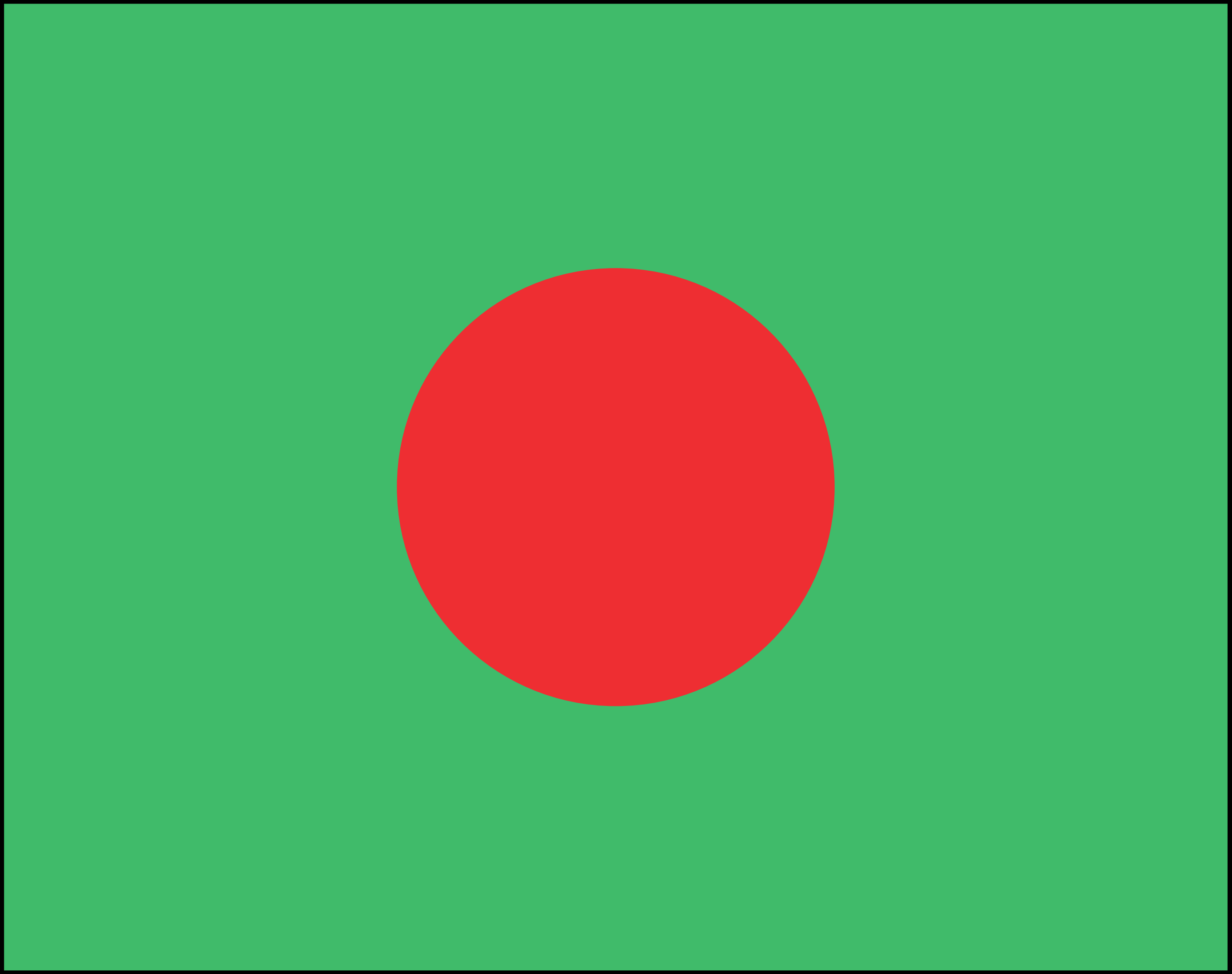



Low Variance
High Variance
Low Bias
High Bias


Low Variance
High Variance
Low Bias
High Bias

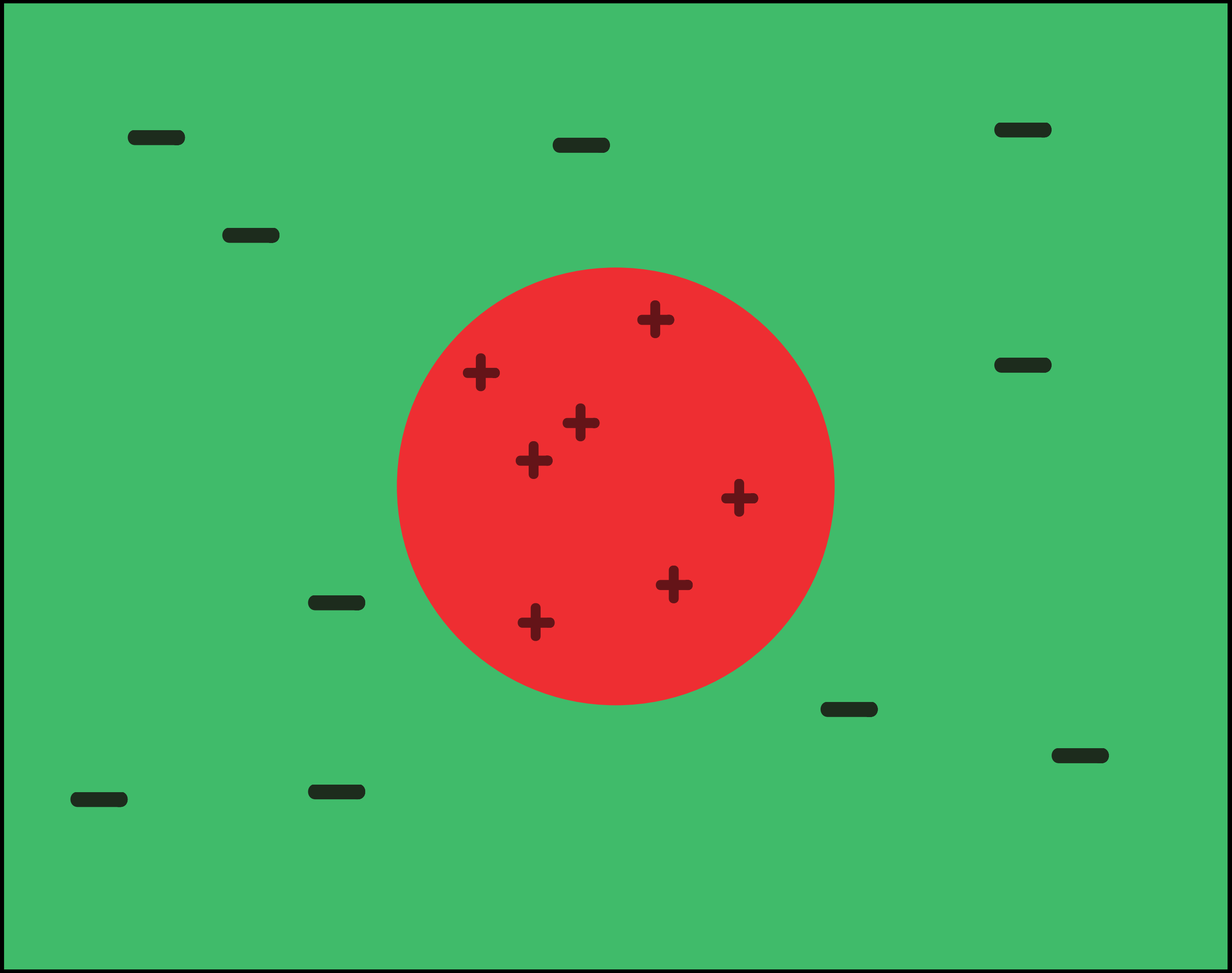

Optimum Model Complexity
Total Error
Variance
Bias2
Error
Model Complexity

$$y = f(x)$$

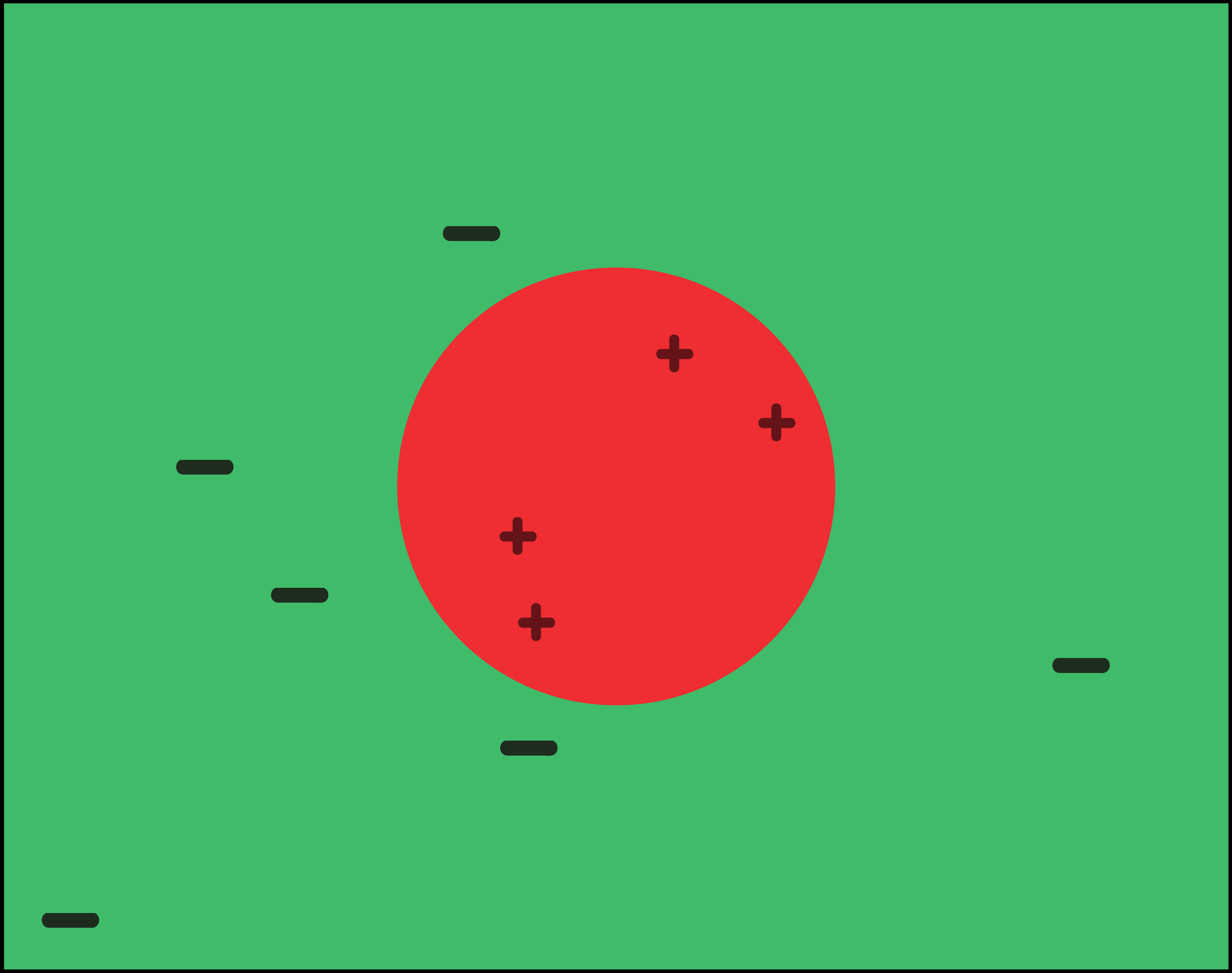


$$y \in \{+1, -1\}$$

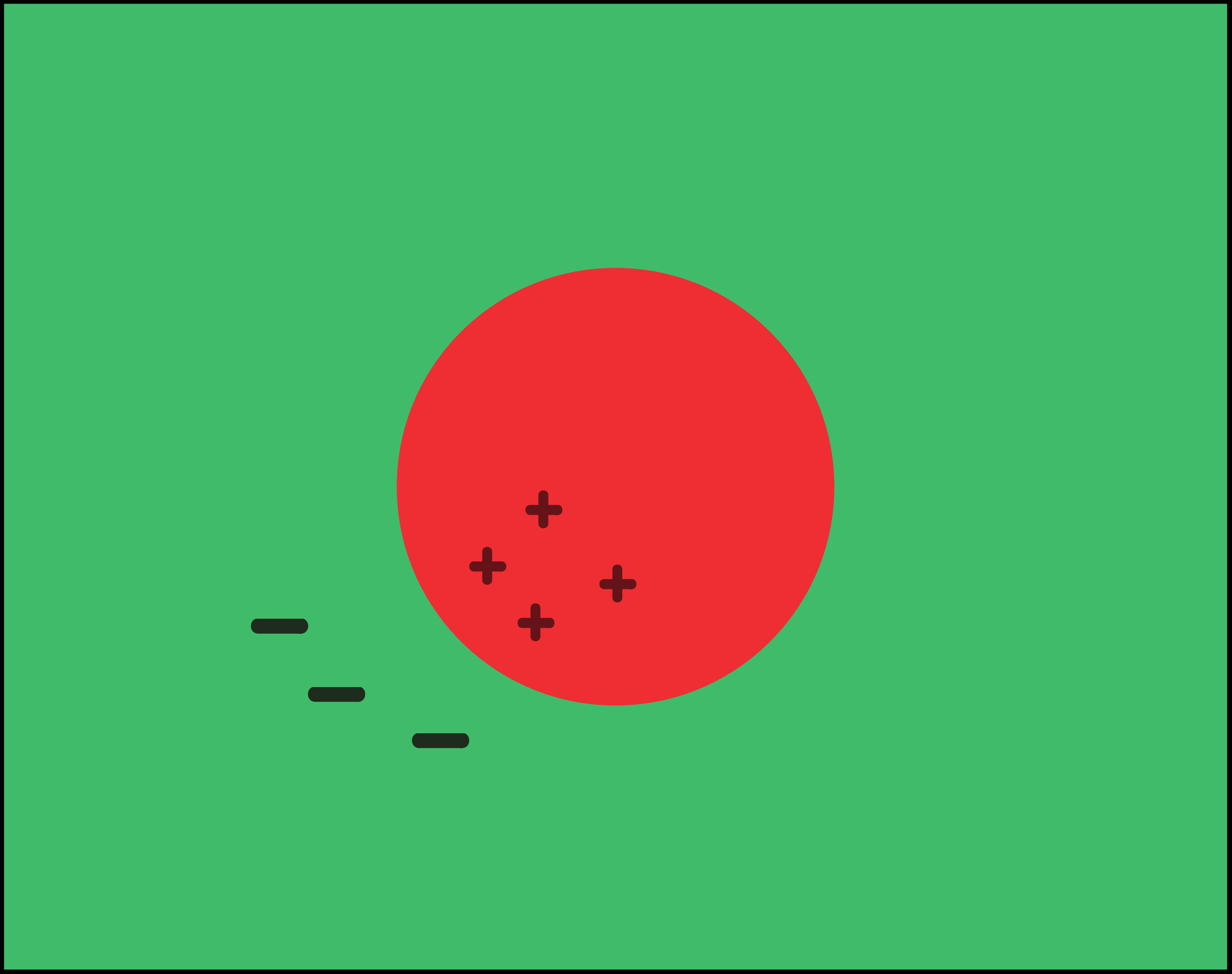
S_1



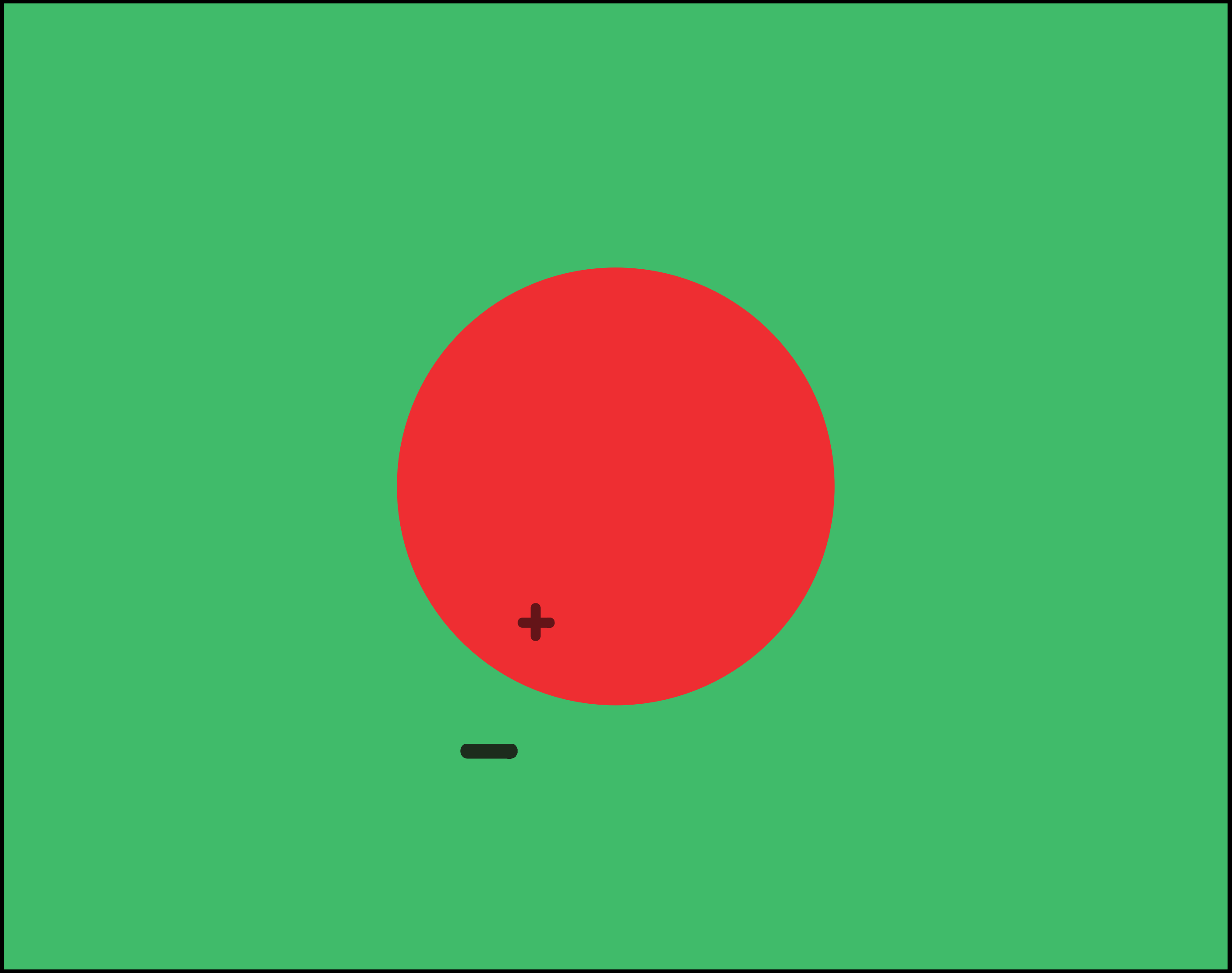
S_2



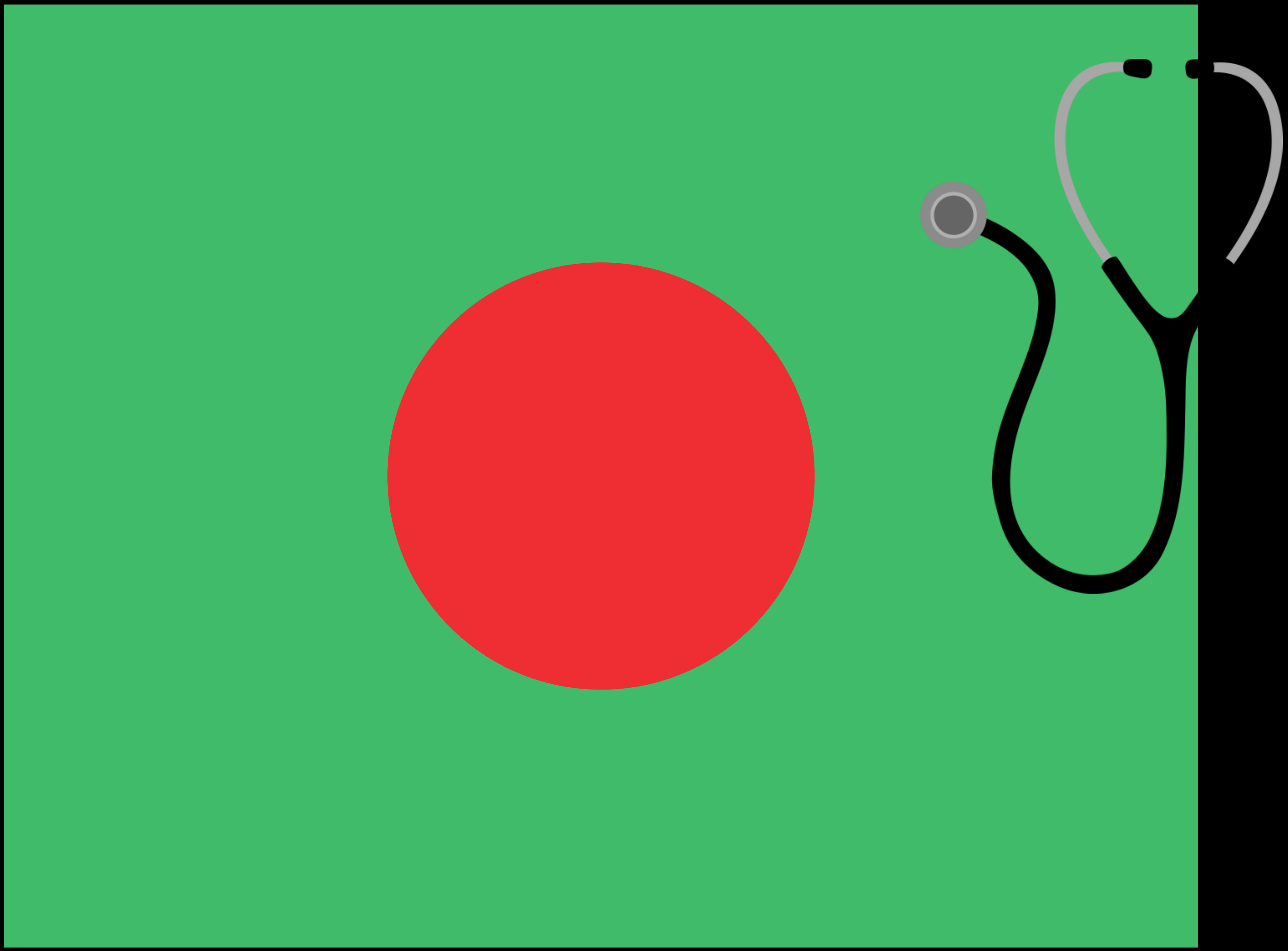
S_3



S_4

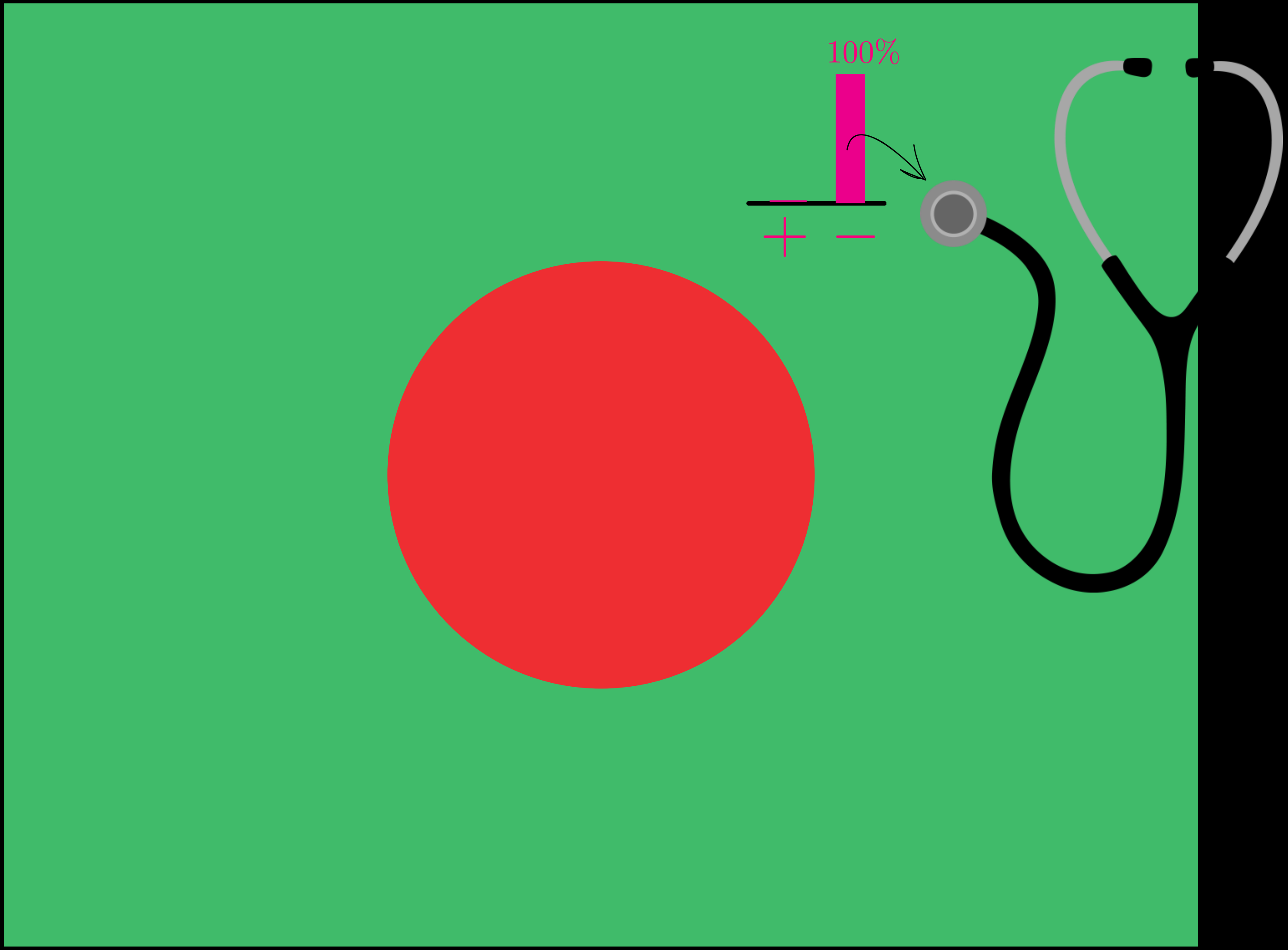


Consider: $h(x) = f(x)$



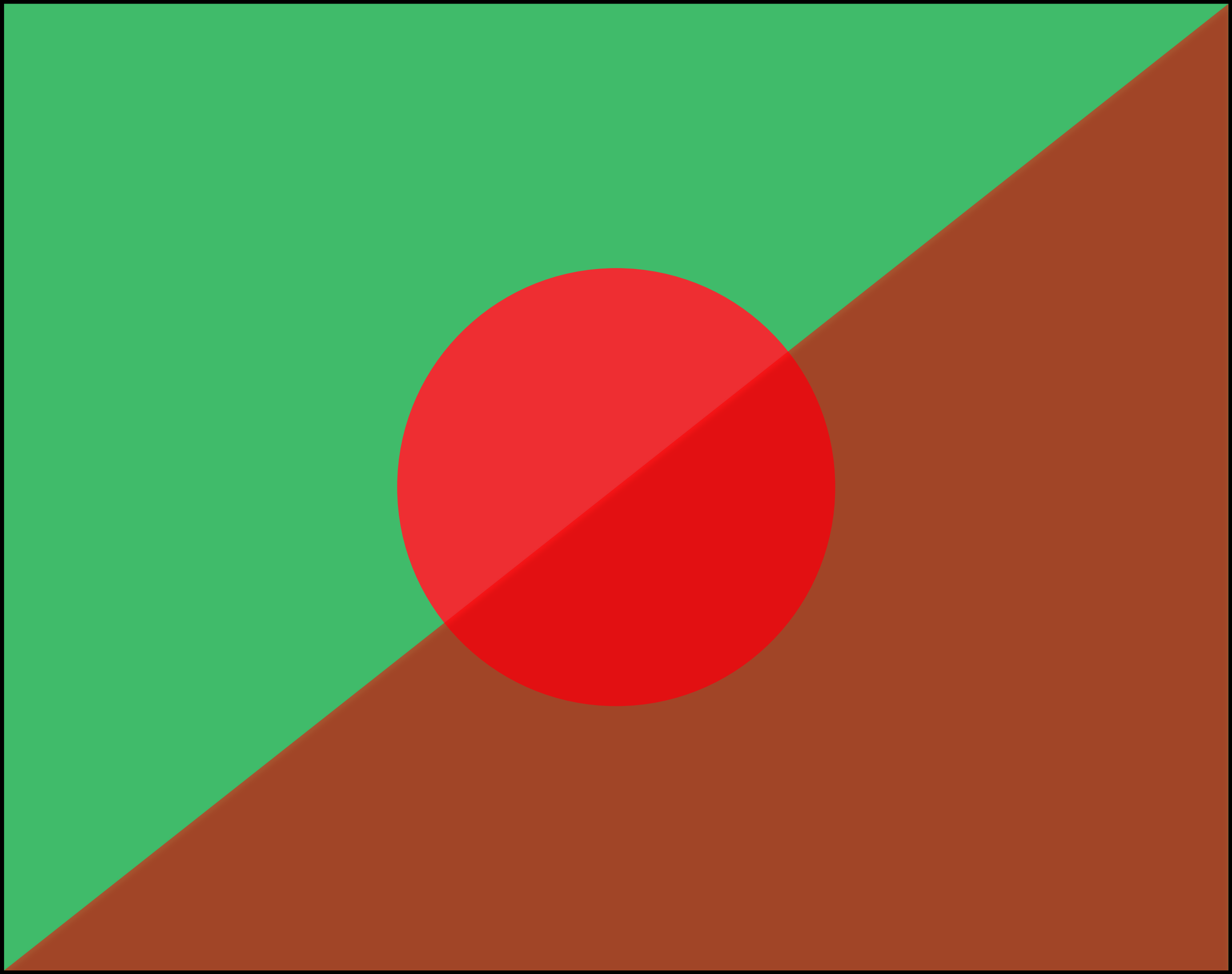
$y \in \{+1, -1\}$

Consider: $h(x) = f(x)$



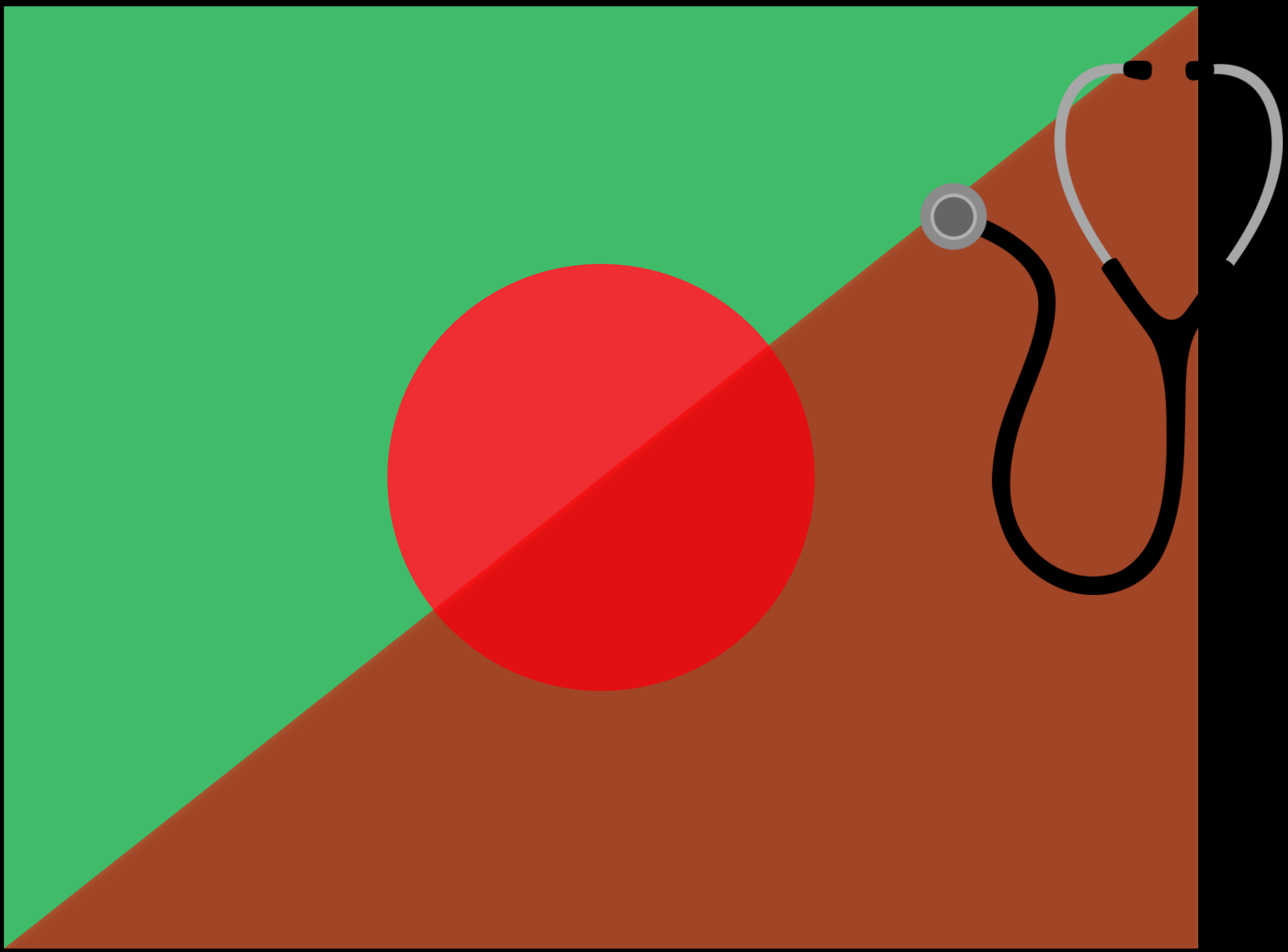
$y \in \{+1, -1\}$

Consider: $h(x) = \mathbf{w}x + b$



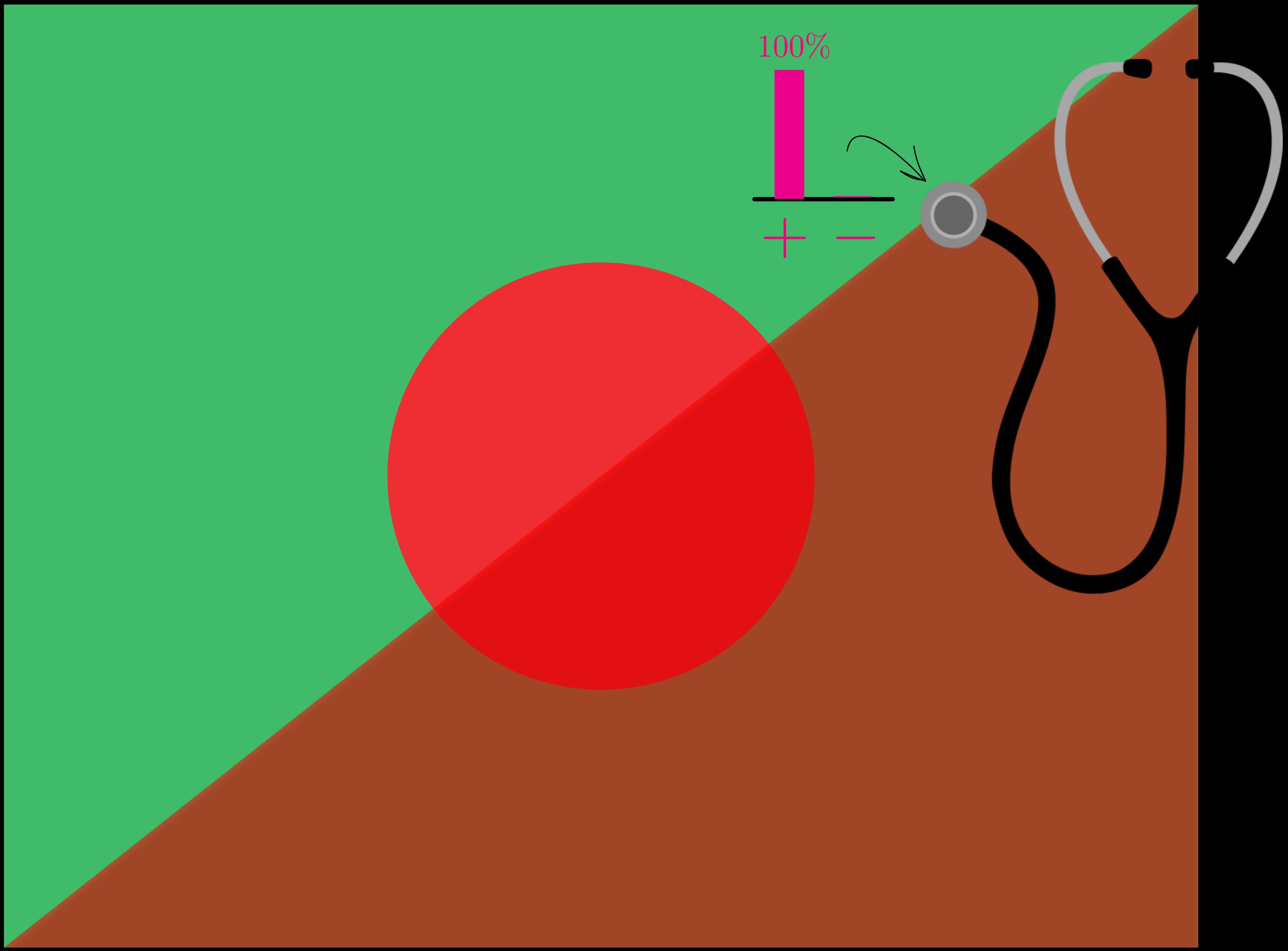
$y \in \{+1, -1\}$

Consider: $h(x) = \mathbf{w}x + b$



$y \in \{+1, -1\}$

Consider: $h(x) = \mathbf{w}x + b$



$y \in \{+1, -1\}$

$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right] =$$

$$(y_i - \mathbb{E}_S[h_S(x_i)])^2 \; +$$

$$+\; \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$$

Label Noise

Noise-free:

$$y_i = f(x_i)$$

Regression:

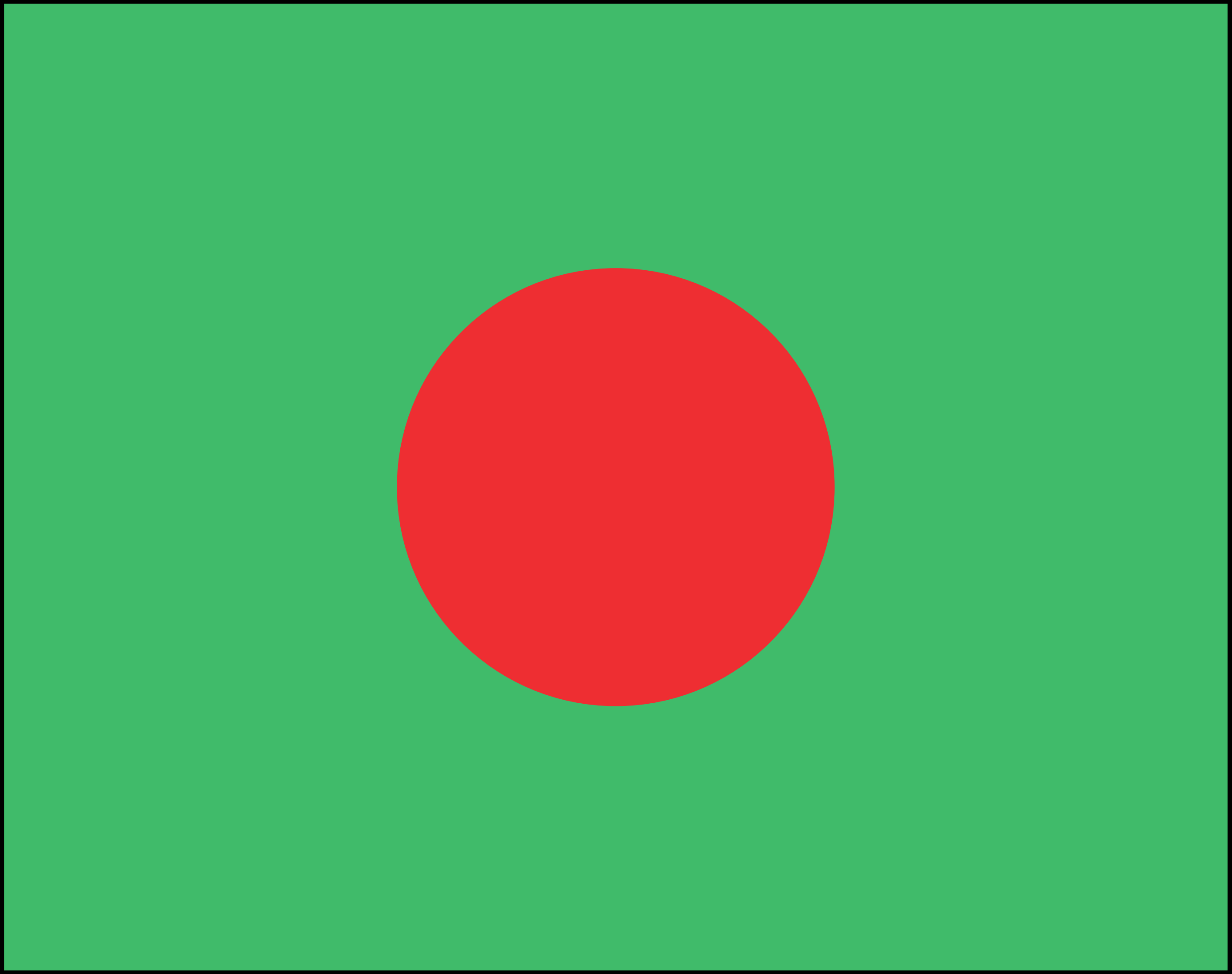
$$y_i = f(x_i) + noise$$
$$y_i = f(x_i) + \mathcal{N}(0, \sigma^2)$$

Classification:

$$y_i = noisy(f(x_i))$$

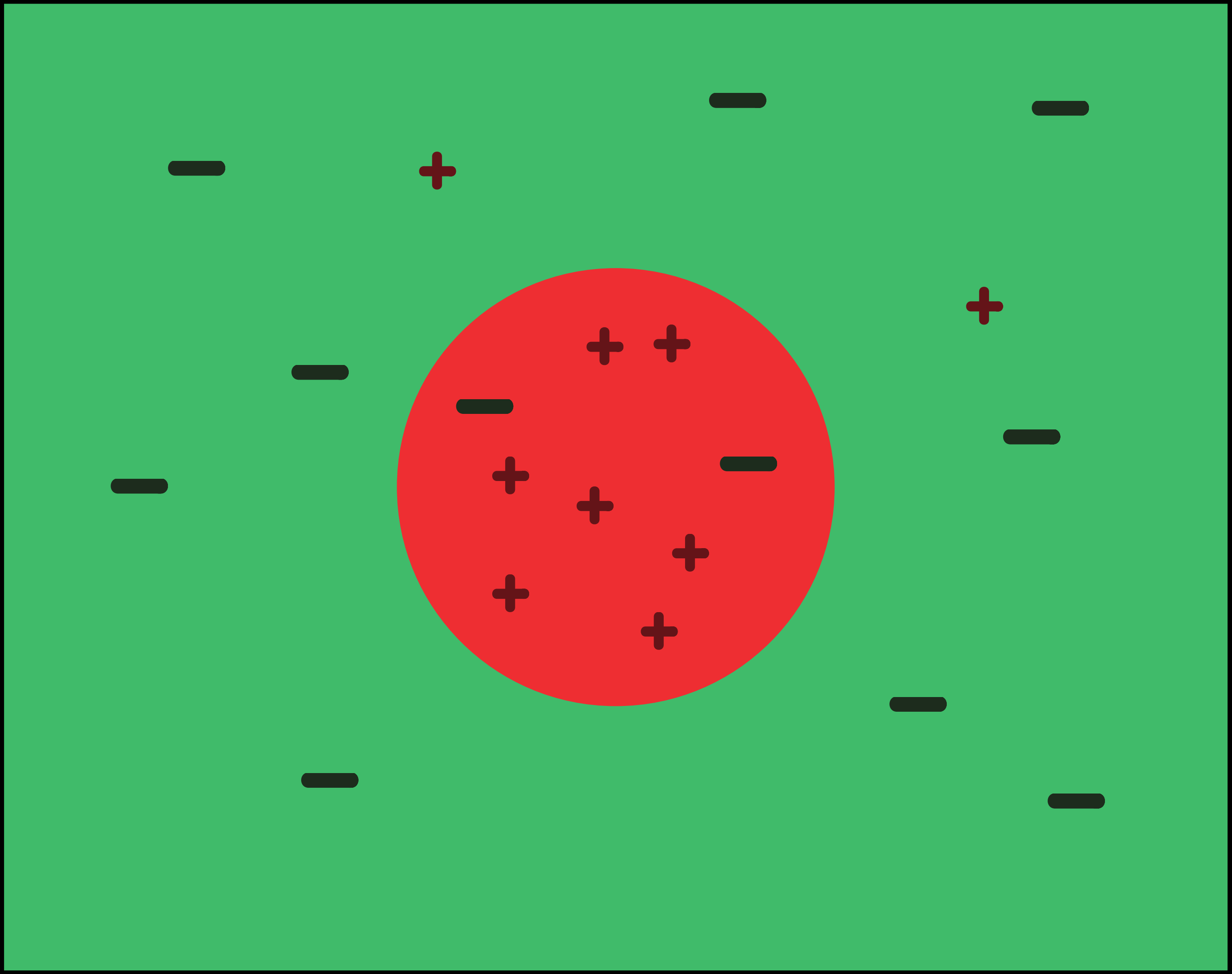
(*noisy()* switches label with probability *p*)

$$y = \text{noisy}(f(x))$$ (flip sign with probability 0.25)



$$y \in \{+1, -1\}$$

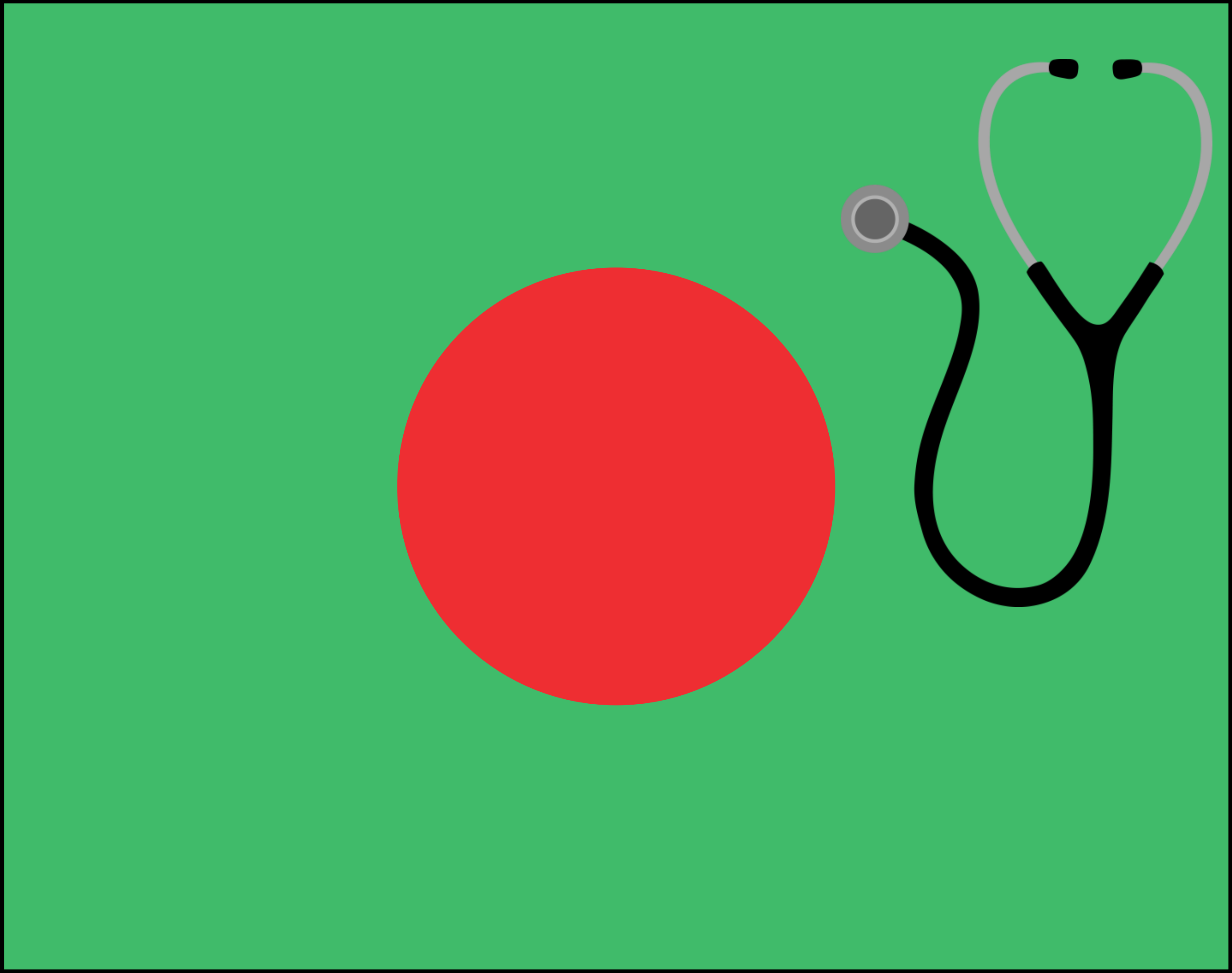
S_1



$y \in \{+1, -1\}$

Consider: $h(x) = f(x)$

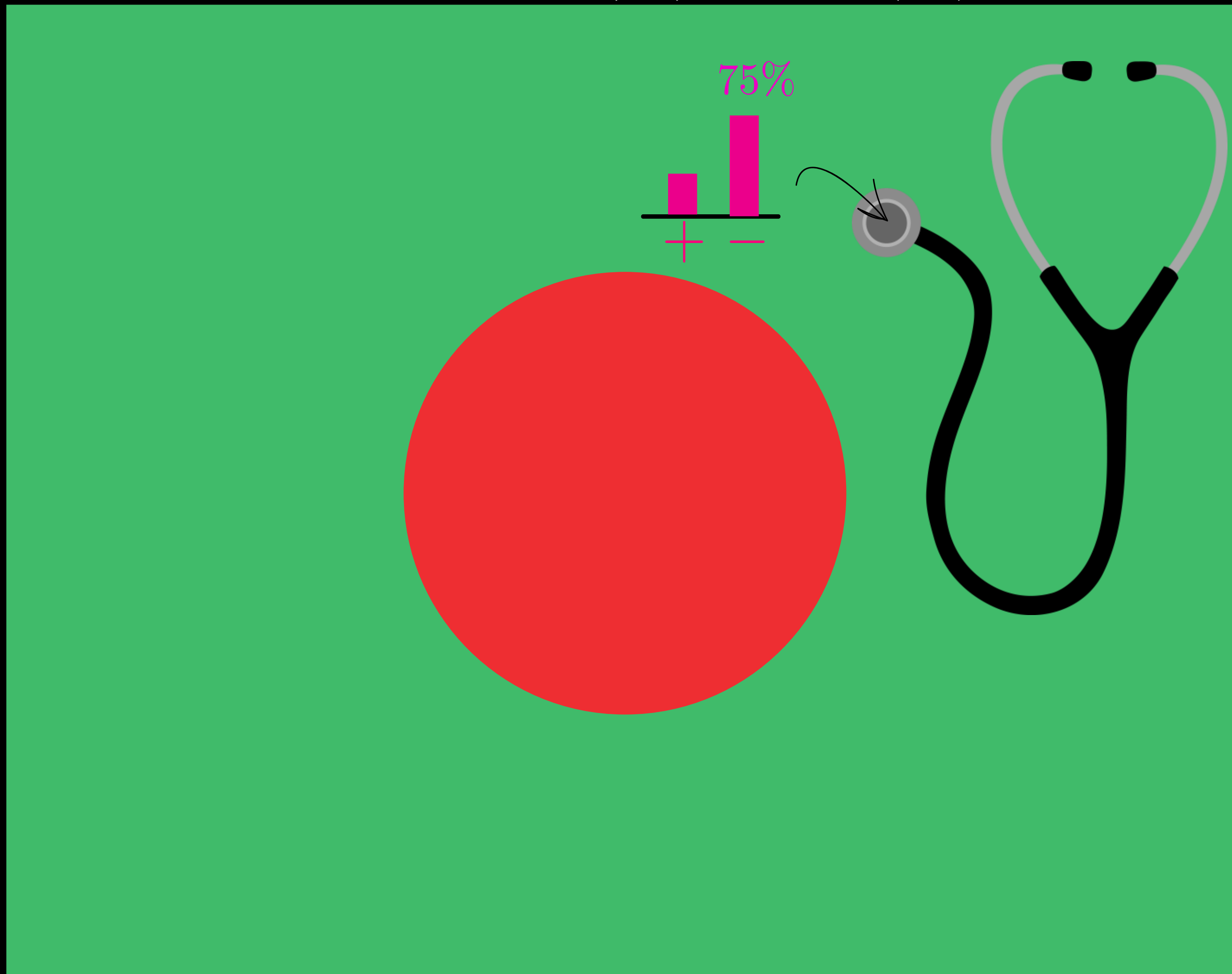
$y = \text{noisy}(f(x))$



$y \in \{+1, -1\}$

Consider: $h(x) = f(x)$

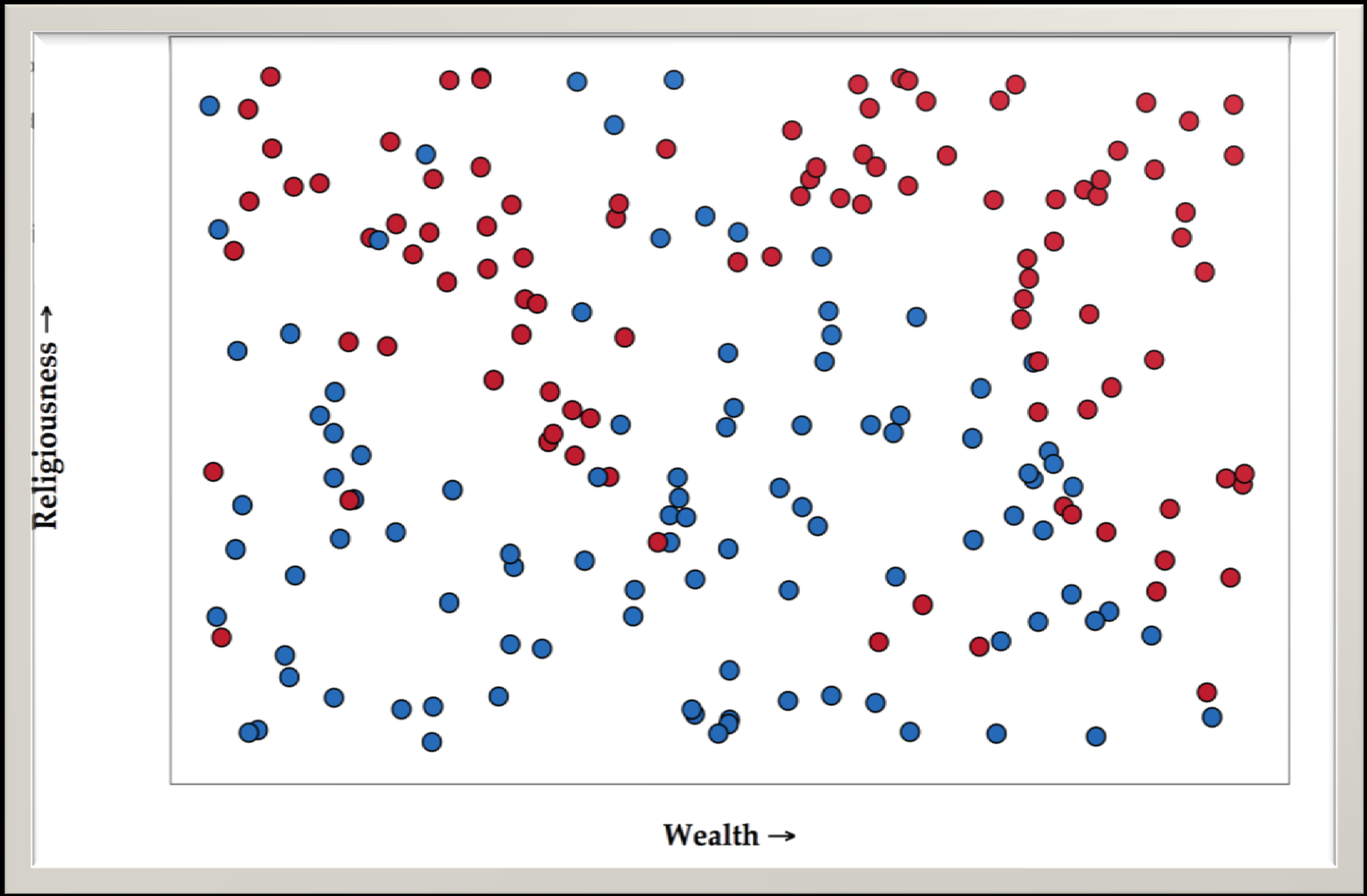
$y = \text{noisy}(f(x))$



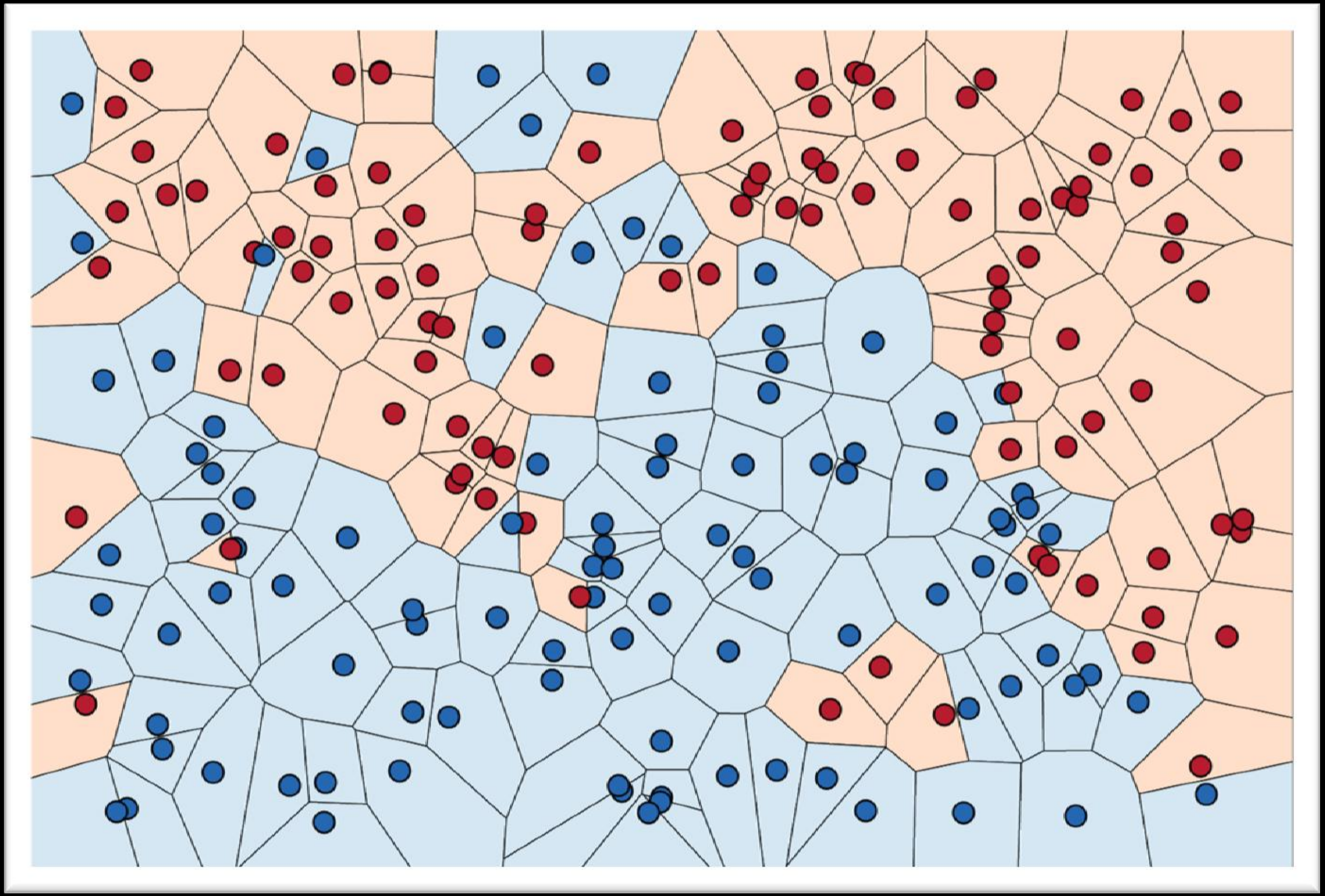
$y \in \{+1, -1\}$

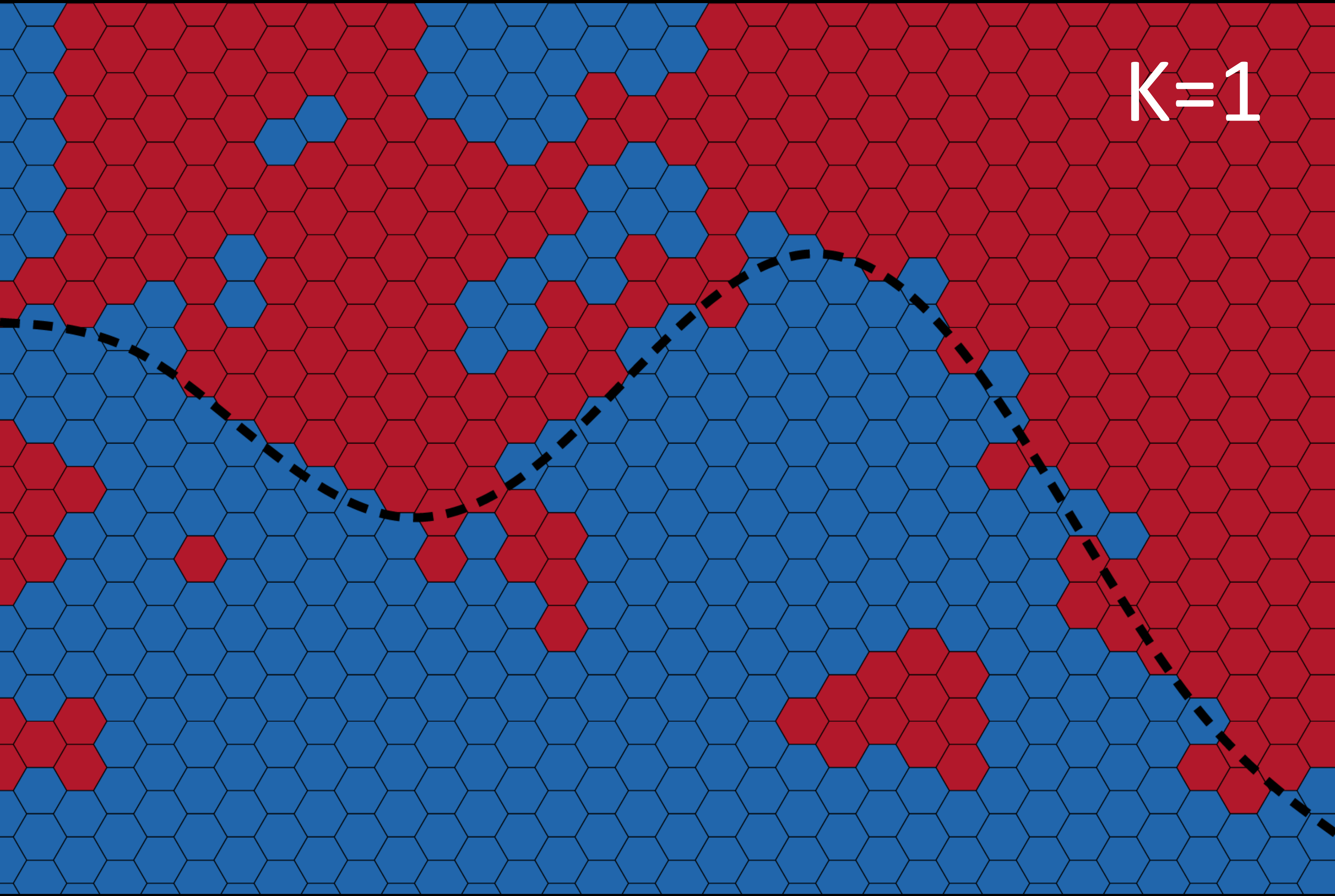
Example
(kNN)

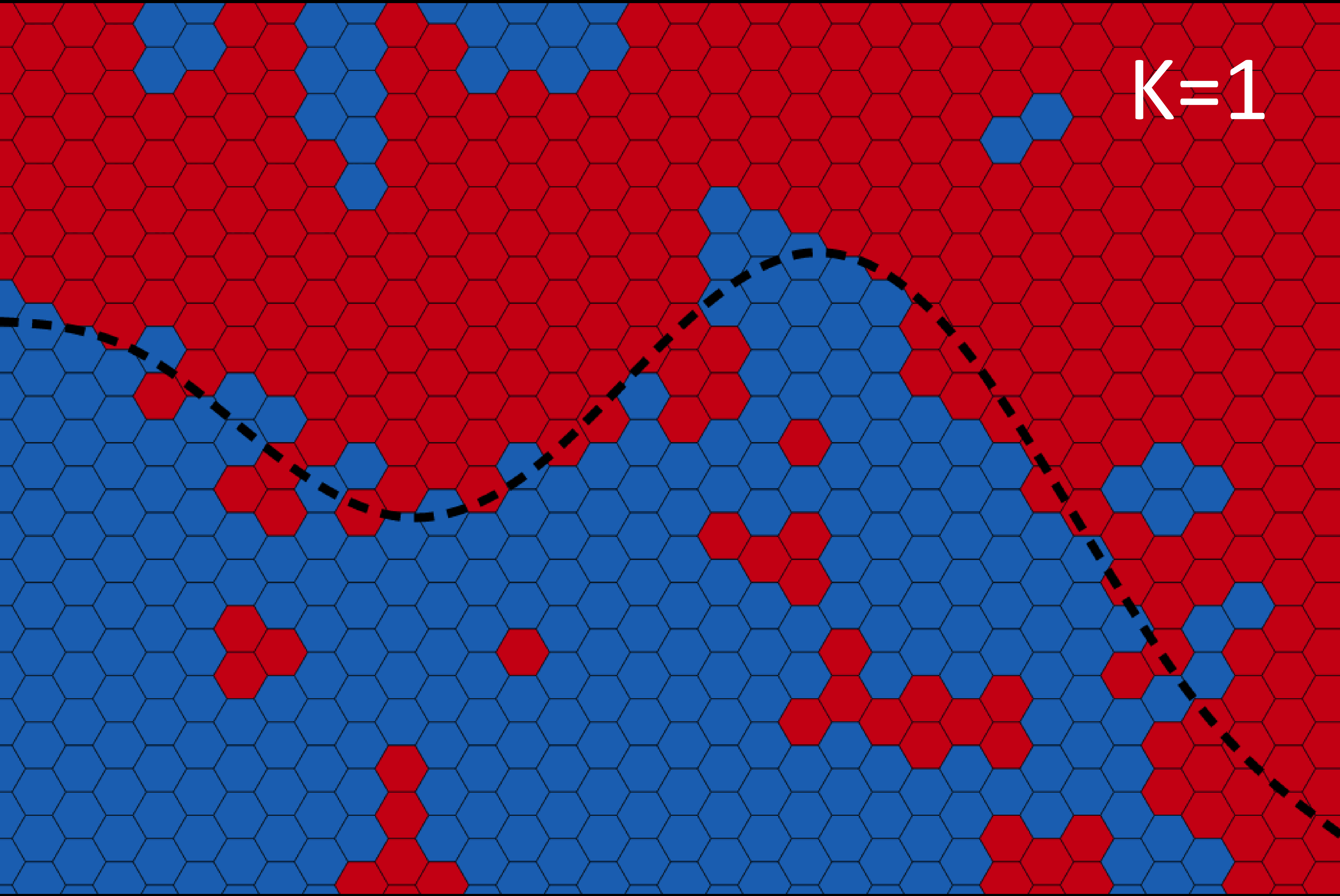
Democrat vs Republican party association

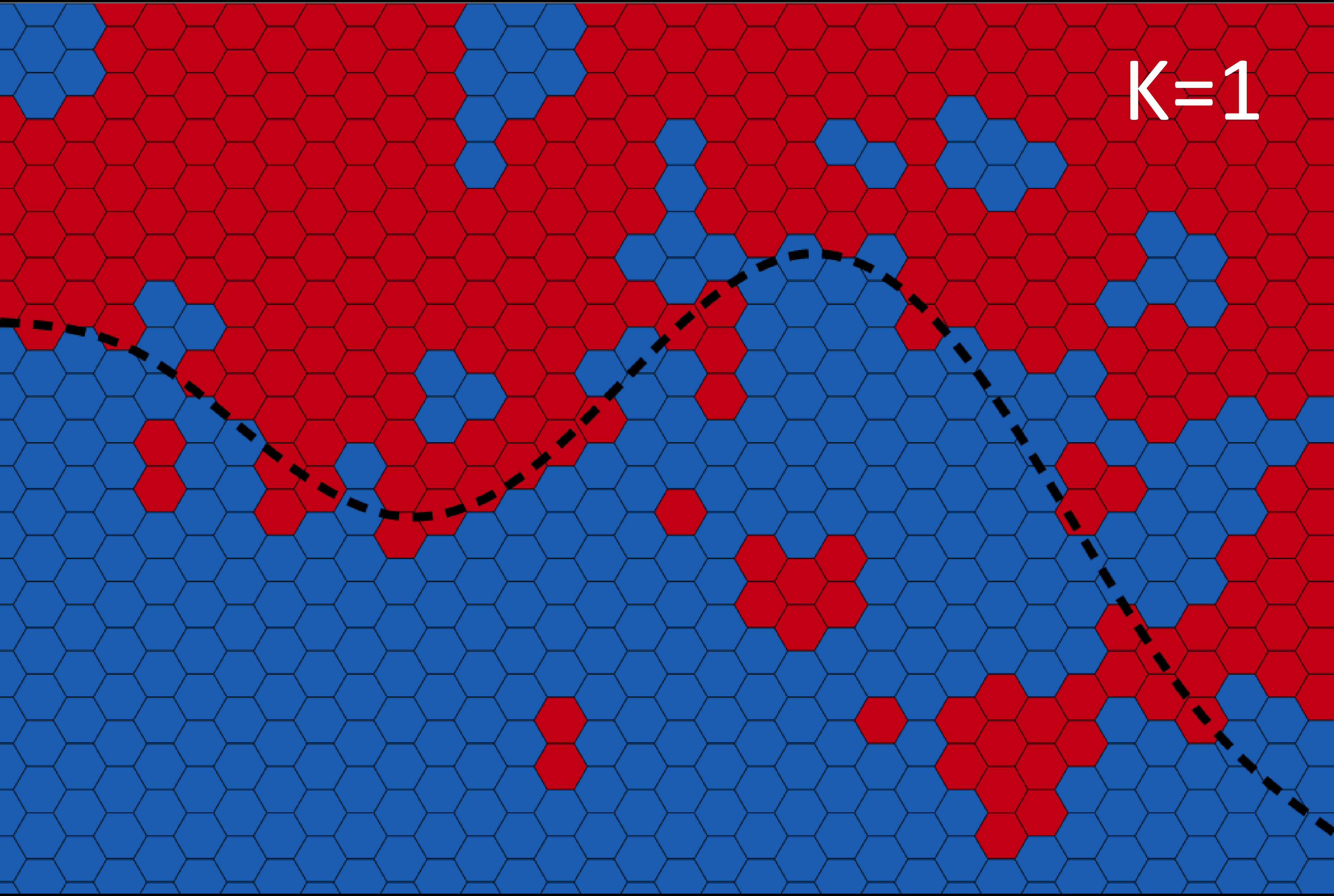


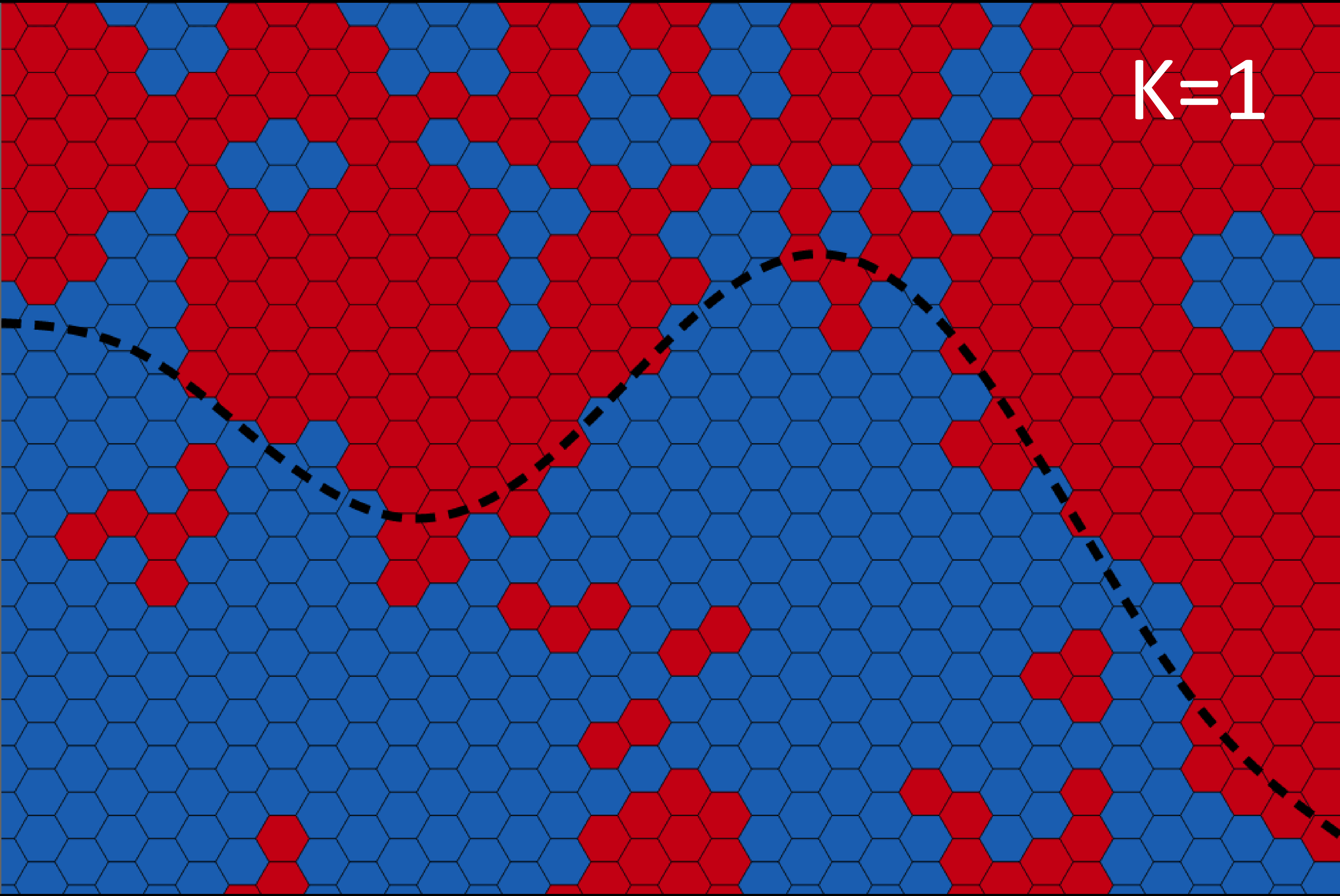
K=1

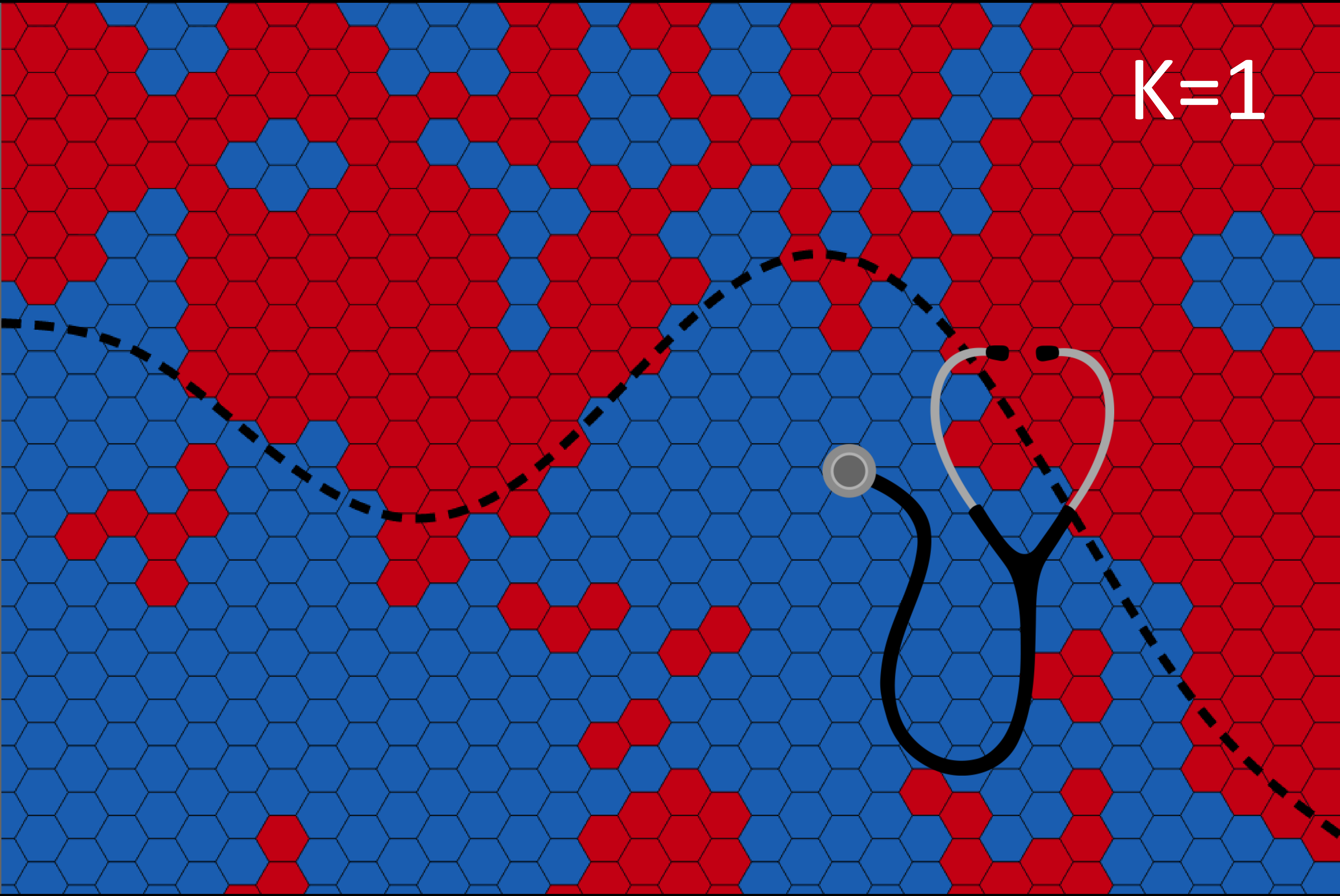


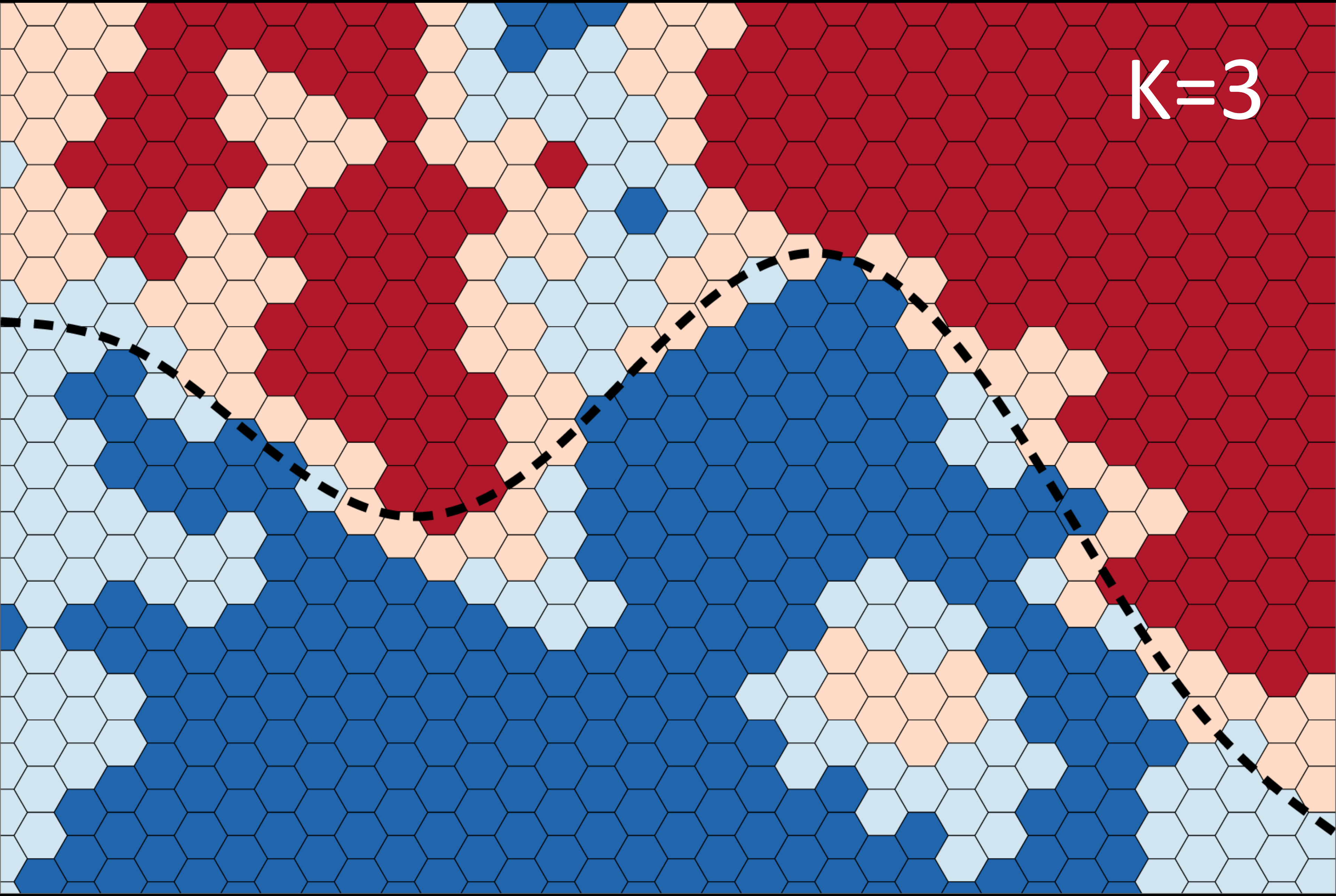

K=1

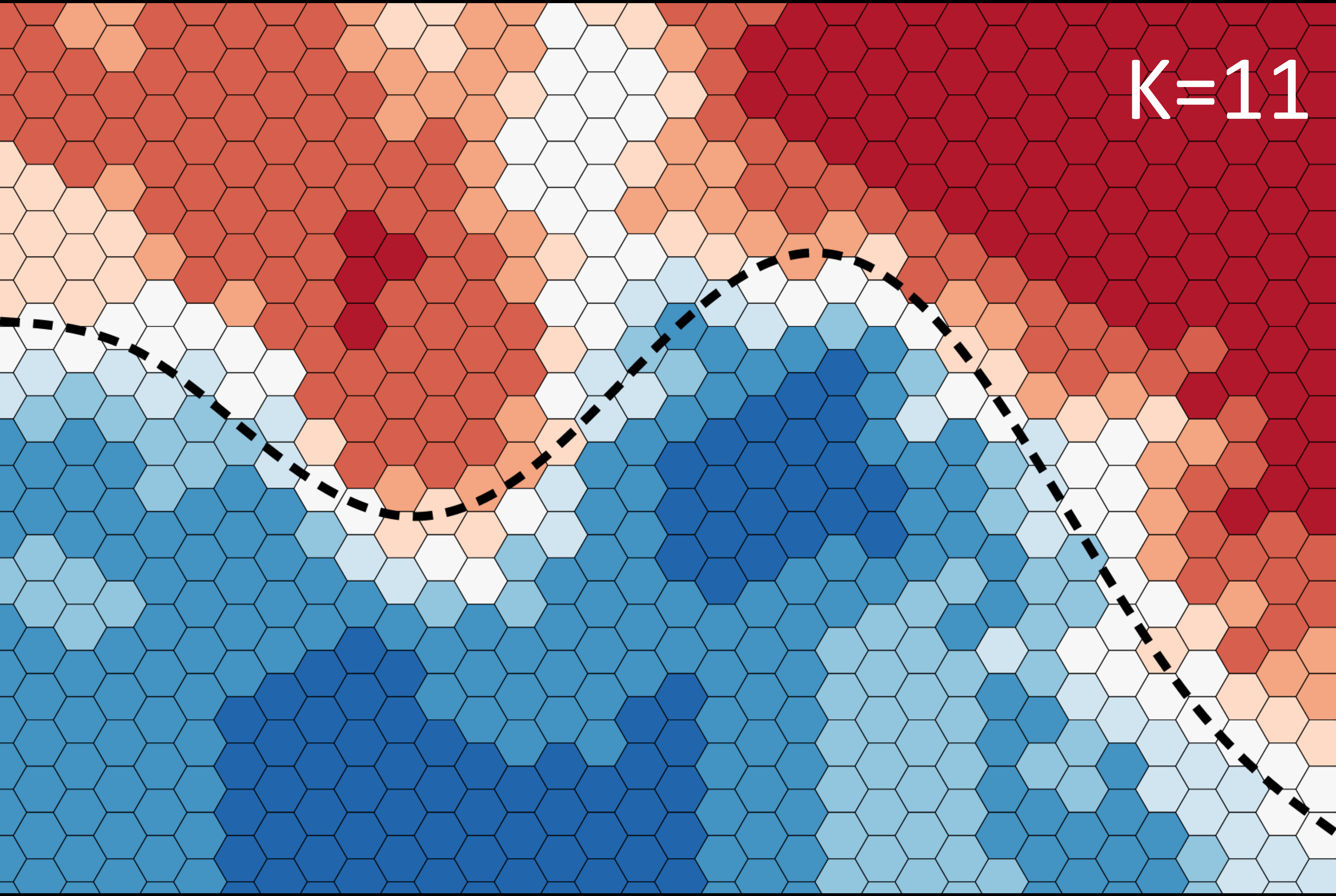

K=1

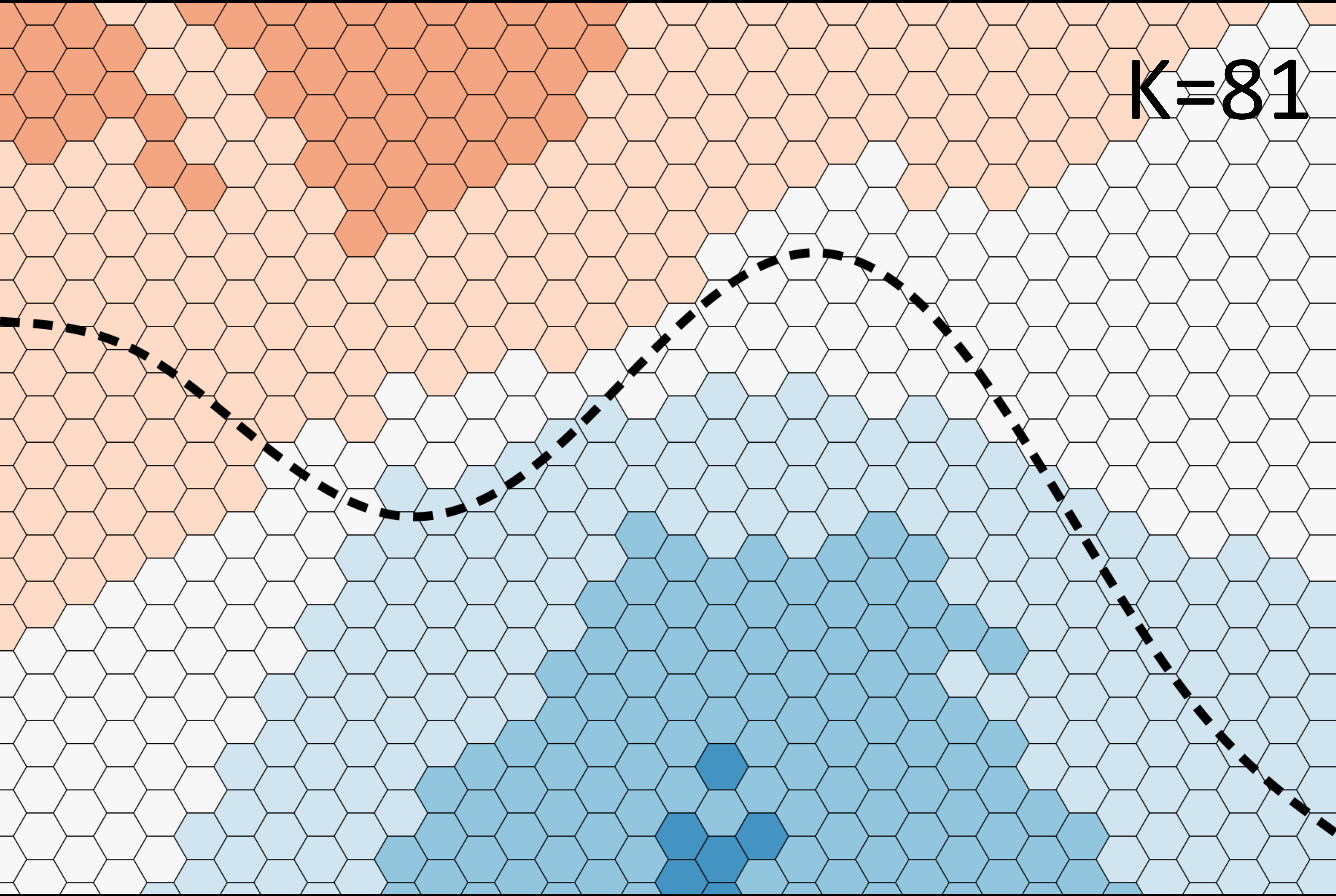

K=1

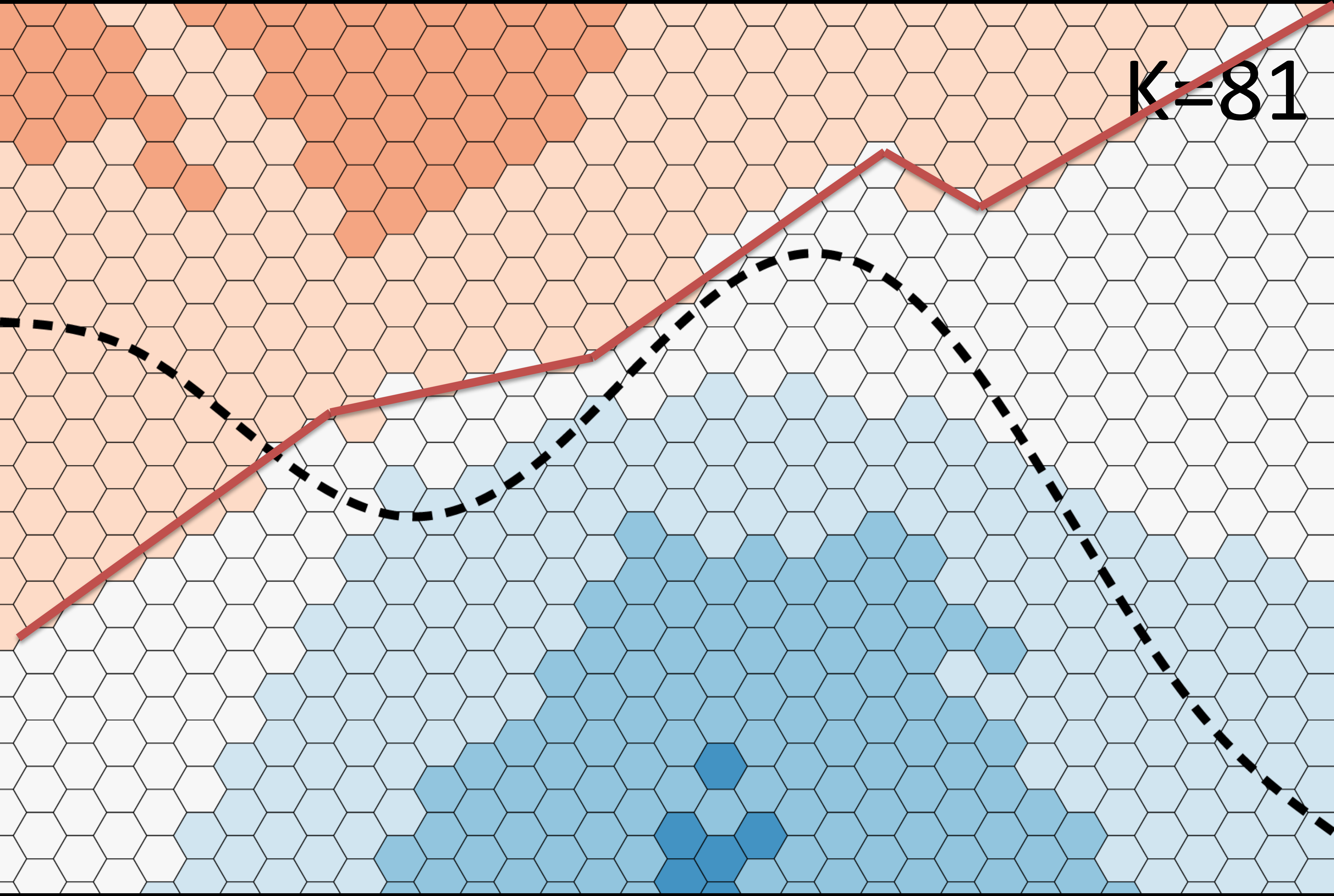

K=1

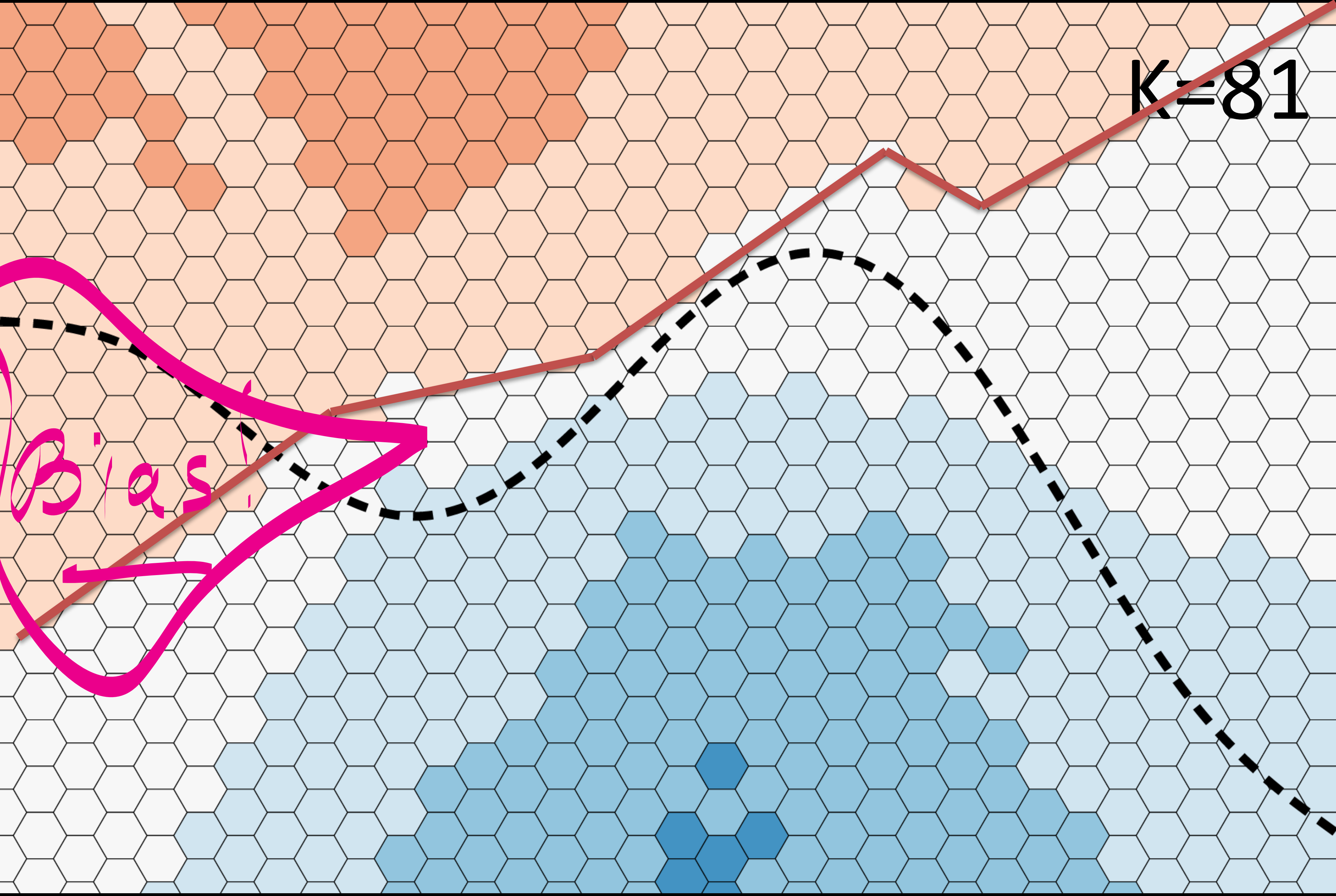

K=1

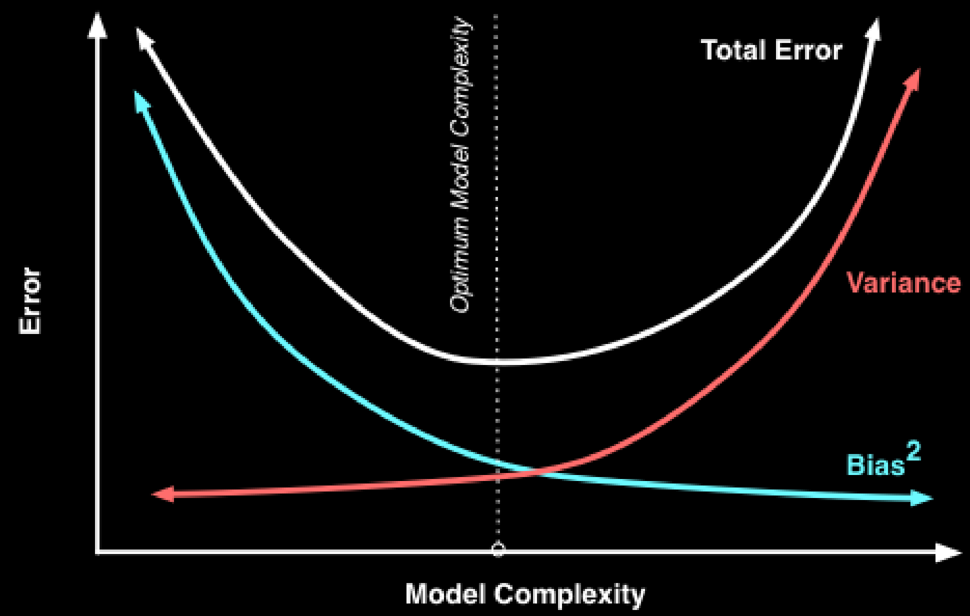

K=3

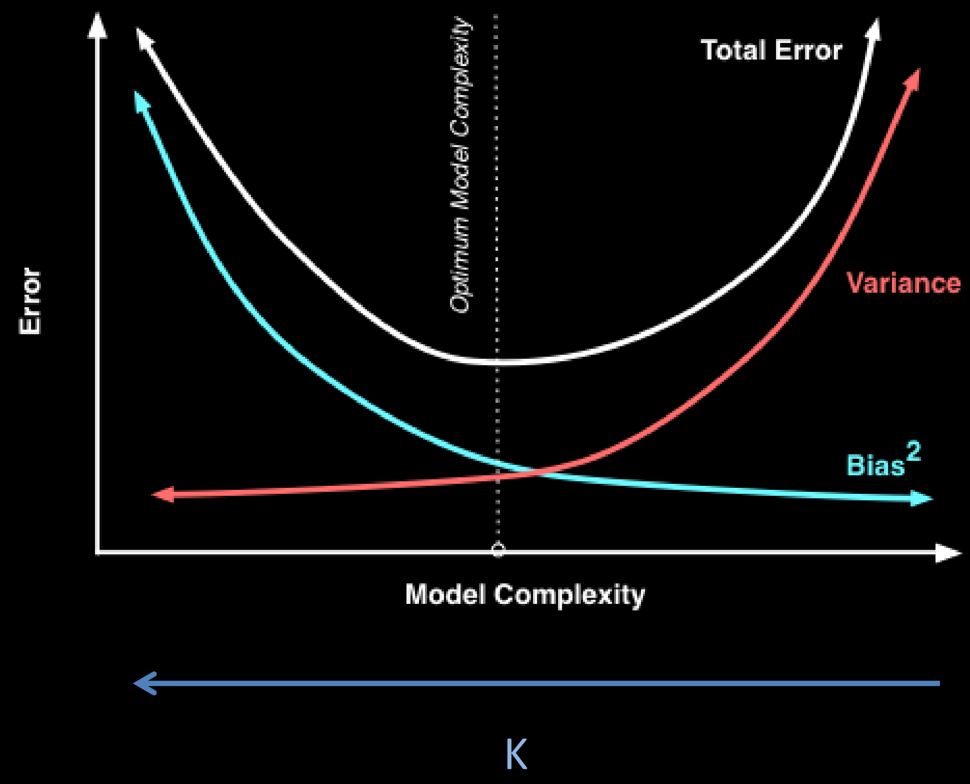

K=11

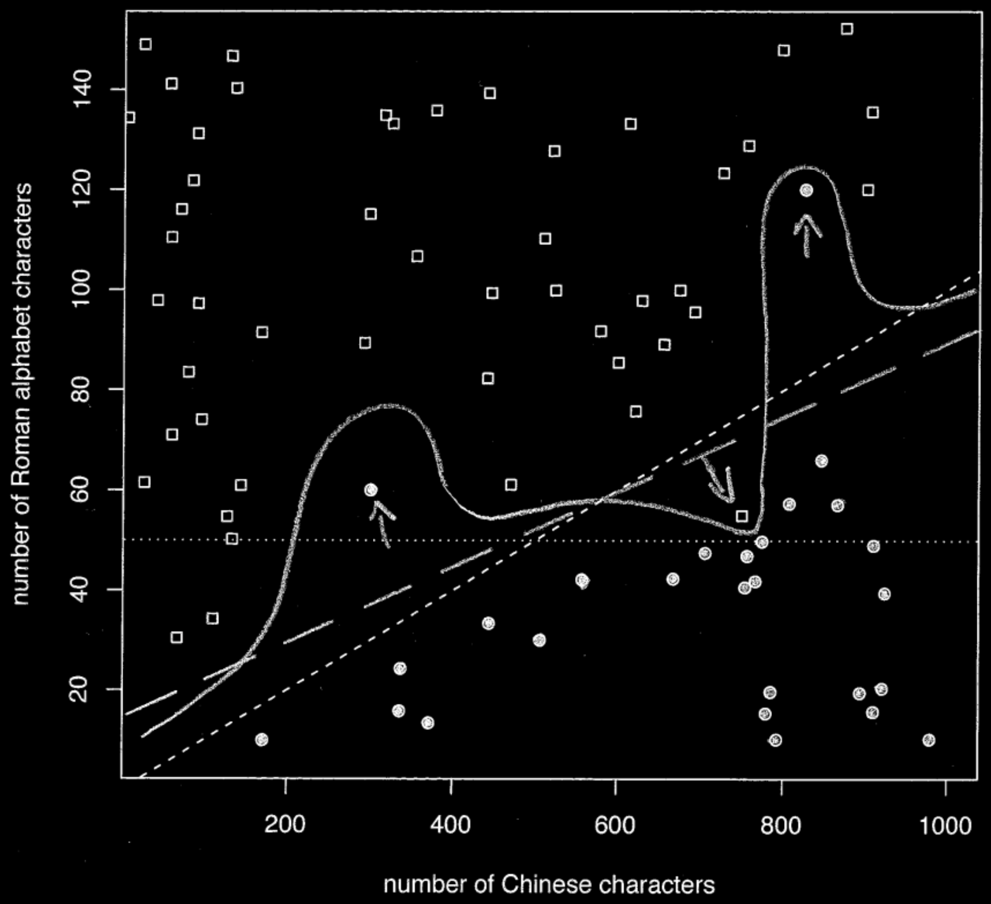

K=81


K=81


K=81
Bias!


Total Error
Optimum Model Complexity
Variance
Error
Bias^2
Model Complexity


Total Error
Optimum Model Complexity
Variance
Bias2
Error
Model Complexity
K



number of Roman alphabet characters
20
40
60
80
100
120
140
200
400
600
800
1000
number of Chinese characters

CLAIM:

$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right] =$$

bias² $$(y_i - \mathbb{E}_S[h_S(x_i)])^2 \; +$$

variance $$+\; \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$$

USEFUL LEMMA:

$$\mathbb{E}[(\alpha - \mathbb{E}[\alpha])^2] = \mathbb{E}[\alpha^2] + \mathbb{E}[\alpha]^2$$

$$y_i = f(x_i) + \mathcal{N}(0, \sigma^2)$$

$$\mathbb{E}_S \left[(y_i - h_S(x_i))^2 \right] =$$

bias² $\quad (y_i - \mathbb{E}_S[h_S(x_i)])^2 \; +$

variance $\quad + \, \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$

$$y_i = f(x_i) + \mathcal{N}(0, \sigma^2)$$

$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right] =$$

bias² $$(f(x_i) - \mathbb{E}_S[h_S(x_i)])^2 \; +$$

variance $$+ \, \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$$

noise $$+ \, \mathbb{E}_S[(f(x_i) - y_i)^2]$$

$$y_i = f(x_i) + \mathcal{N}(0, \sigma^2)$$

$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right] =$$

bias² $$(f(x_i) - \mathbb{E}_S[h_S(x_i)])^2 \; +$$

variance $$+ \, \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$$

noise $$+ \, \sigma^2$$

$$\mathbb{E}_S\left[(y_i - h_S(x_i))^2\right] =$$

bias² $$(y_i - \mathbb{E}_S[h_S(x_i)])^2 \; +$$

variance $$+\; \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$$

BAGGING

revisited

Bagging

(Breiman, 1996)

Bagging (Boostrap aggregating).

$\textsc{Bagging}(S = ((x_1, y_1), \ldots, (x_m, y_m)))$

for $t \leftarrow 1$ **to** T **do**

  $S_t \leftarrow \textsc{Bootstrap}(S)$ $\triangleright$ i.i.d. sampling with replacement from S.

  $h_t \leftarrow \textsc{TrainClassifier}(S_t)$

return $h_S = x \mapsto \textsc{MajorityVote}((h_1(x), \ldots, h_T(x)))$

Why does it work?

Bagging

Ensemble :

$$h_S(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Bagging : Special case where we fix:

$$\alpha_t = 1 \quad \text{and} \quad h_t = \mathbb{L}(S_t)^*$$

* $\mathbb{L}$ is some learning algorithm

S_t is a training set drawn from distribution $P(< x, y >)$

Bagging

Bagging Ensemble :

$$h_S(x) = \text{sign}\left(\sum_{t=1}^{T} h_t(x)\right)$$

What happens to *bias* and *variance?*

Bagging

Bagging Ensemble (regression) :

$$h_S(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$$

bias^2 $\quad (y_i - \mathbb{E}_S[h_S(x_i)])^2$

variance $\quad \mathbb{E}_S[(h_s(x_i) - \mathbb{E}_S[(h_S(x_i))])^2]$

Bagging

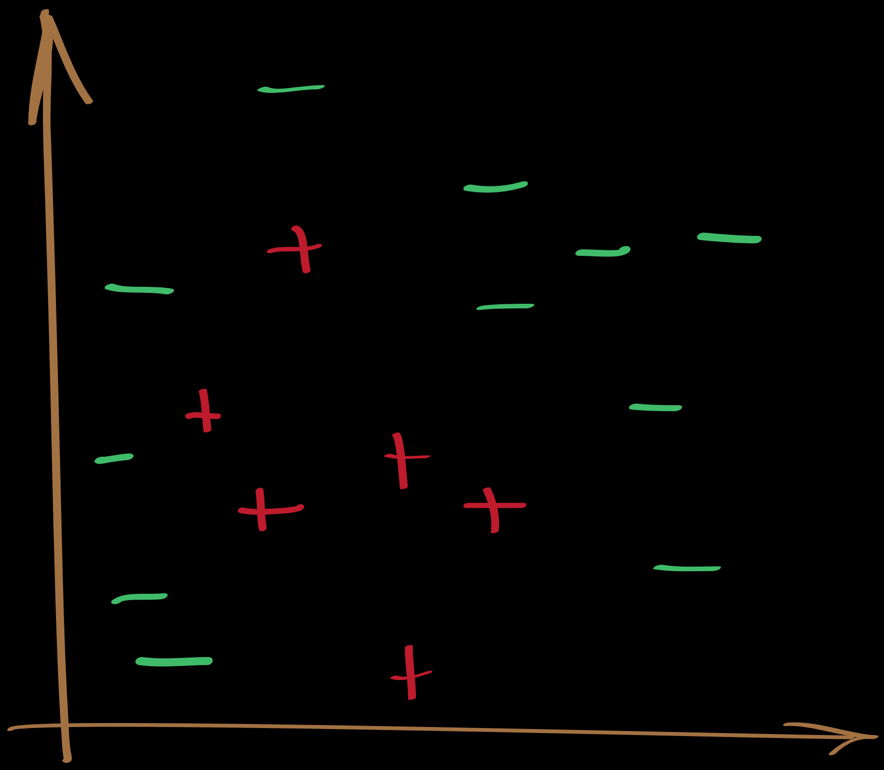
What happens to *bias* and *variance?*

$$\text{Bias}(h_s, x_i) =$$

$$\text{Var}(h_s, x_i) \approx$$

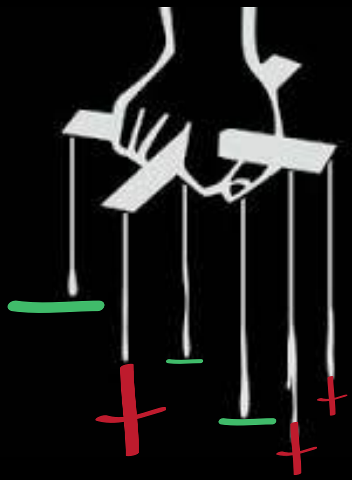
Bagging has approximately the same bias, but reduces variance of individual classifiers!

Bagging

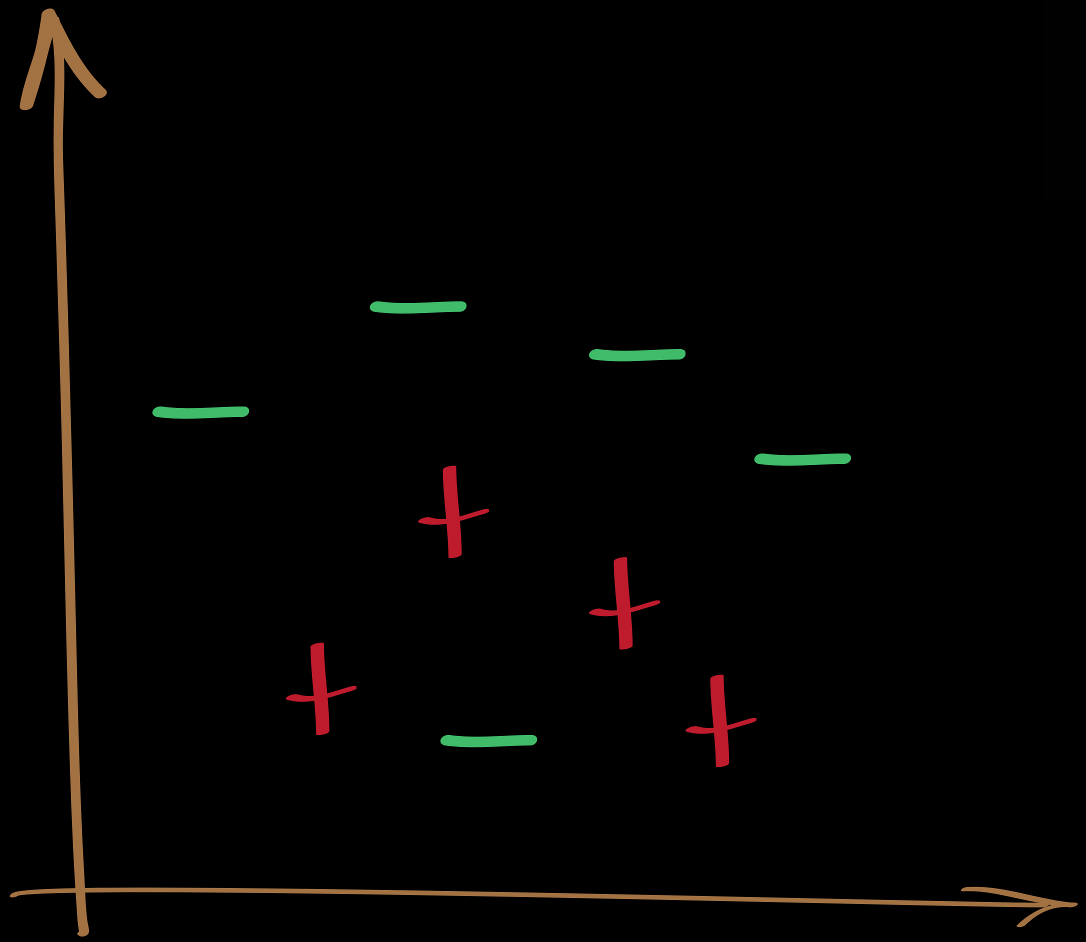


Bagging as a “Training set manipulator”

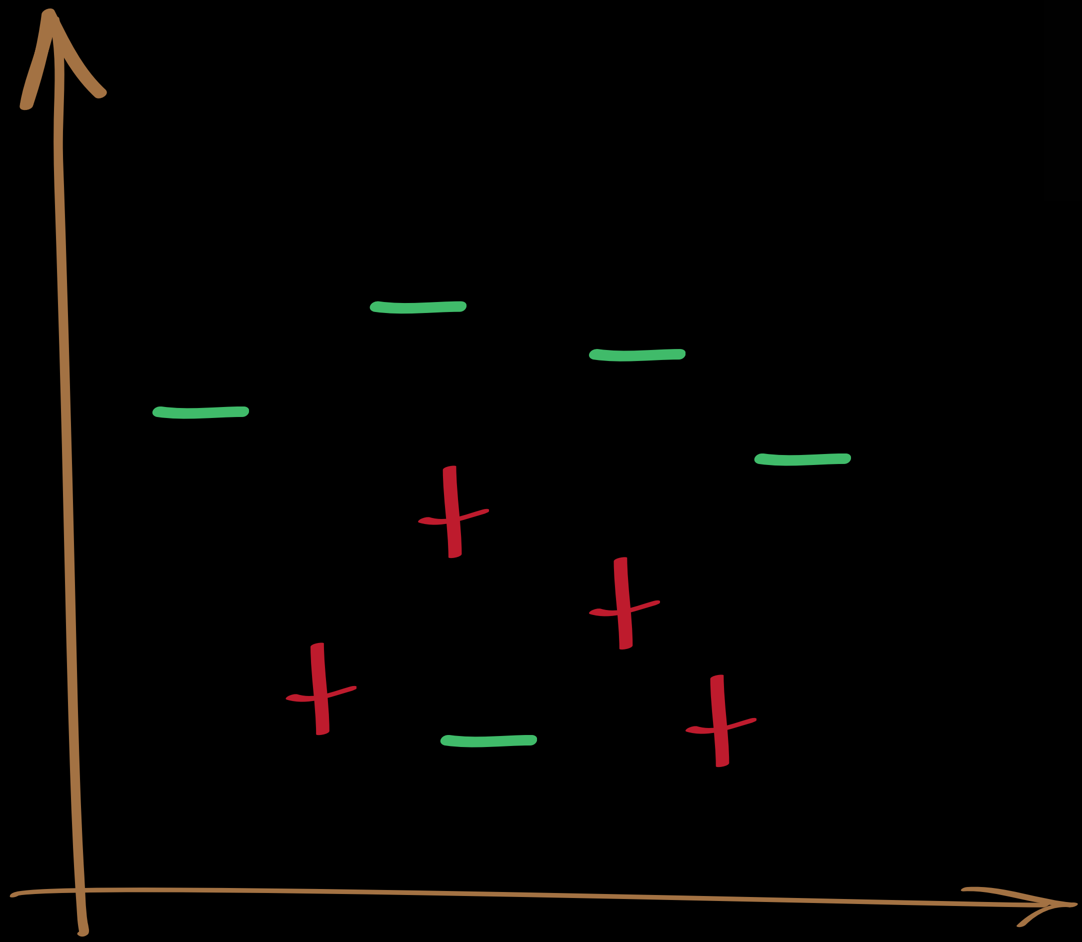
Bagging as a "Training set manipulator"



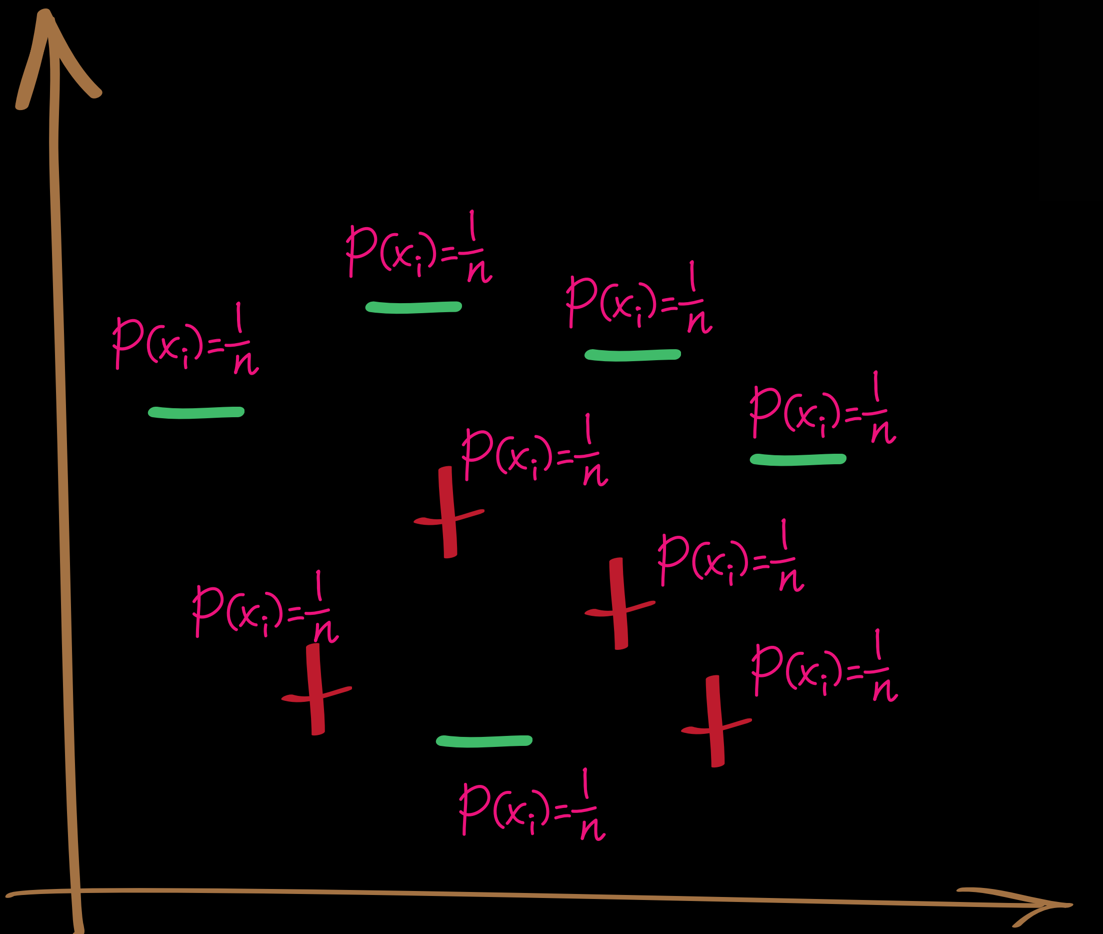
Bagging as a “Training set manipulator”



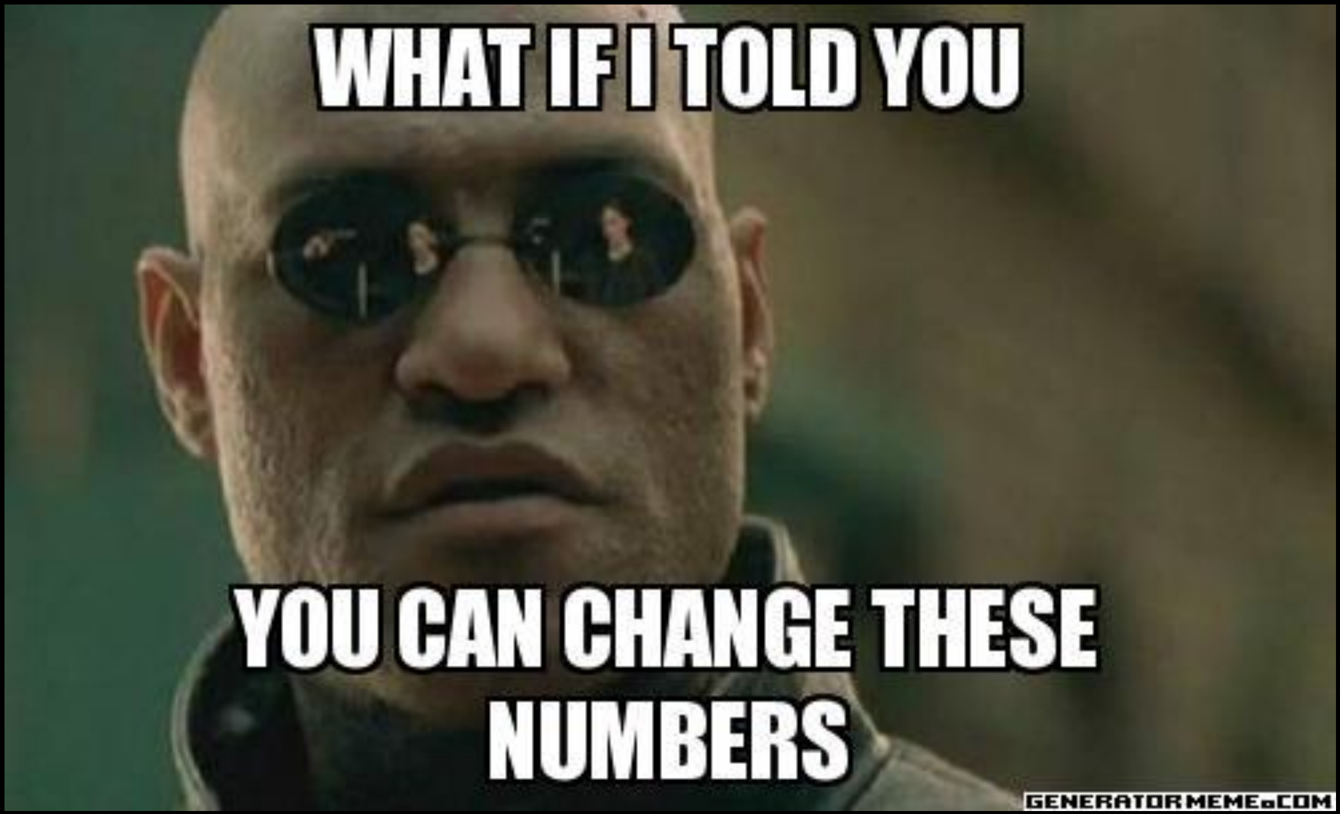
Bagging as a "Training set manipulator"



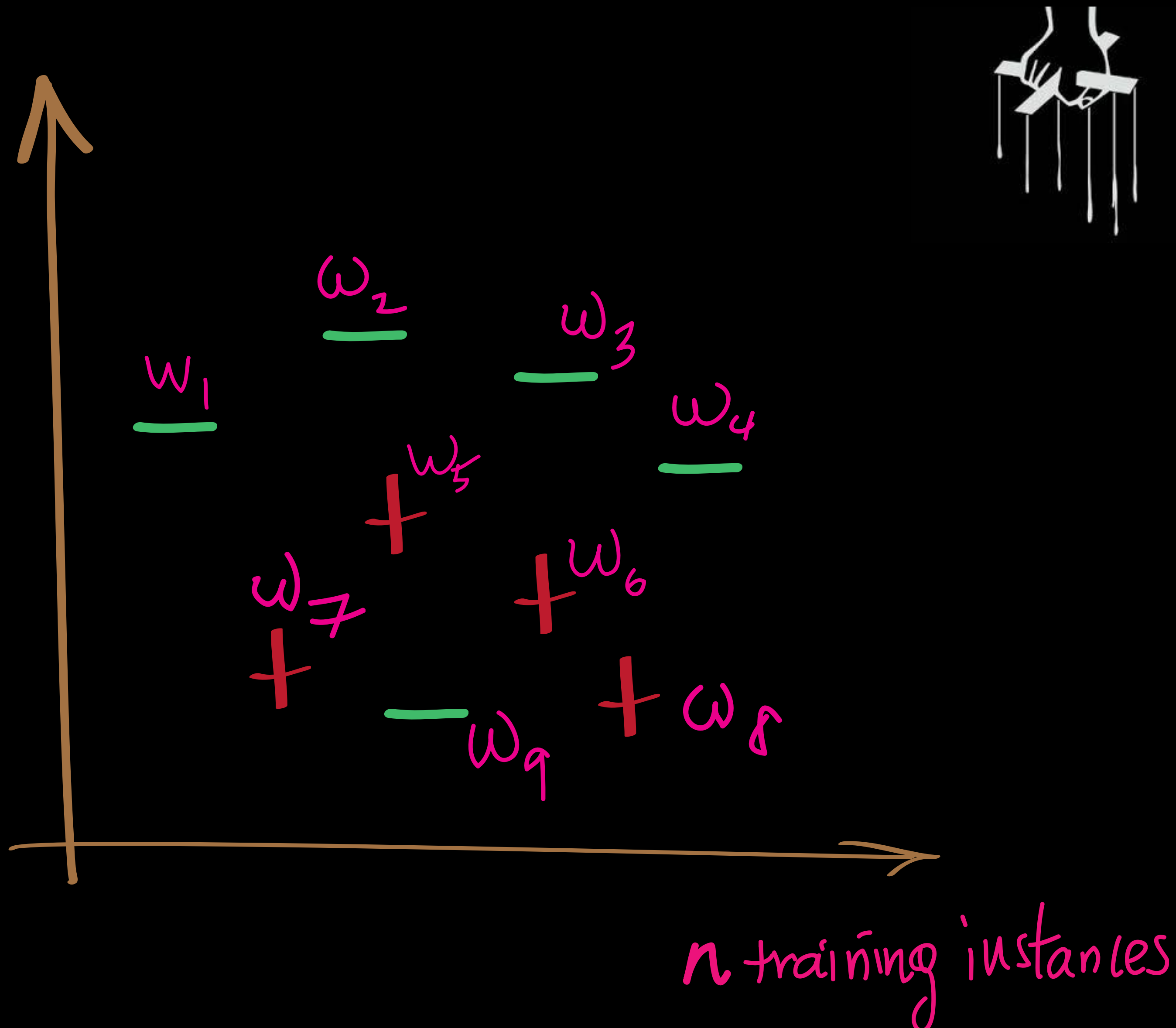
Bagging as a "Training set manipulator"



Bagging as a "Training set manipulator"



Bagging as a "Training set manipulator"



Ensemble

Problem : given T binary classification hypotheses (h_1,…, h_T), **find** a combined classifier:

$$h_S(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

with better performance.

Teaser