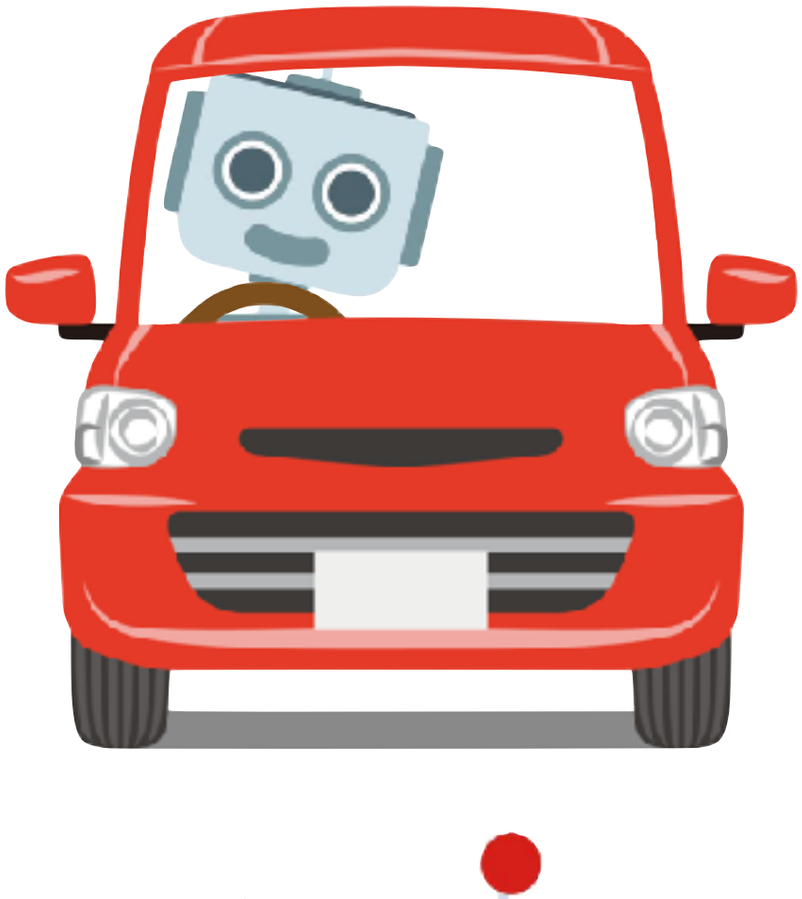
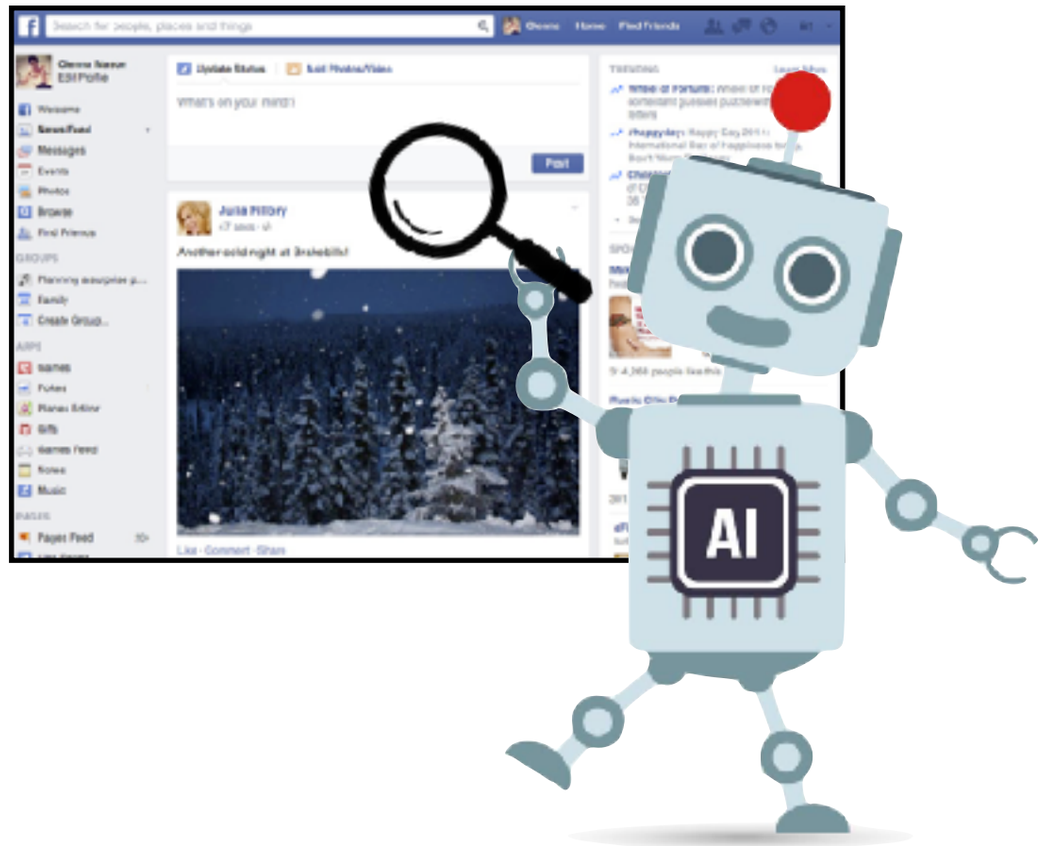
On the Calibration of Learning-to-Defer Systems

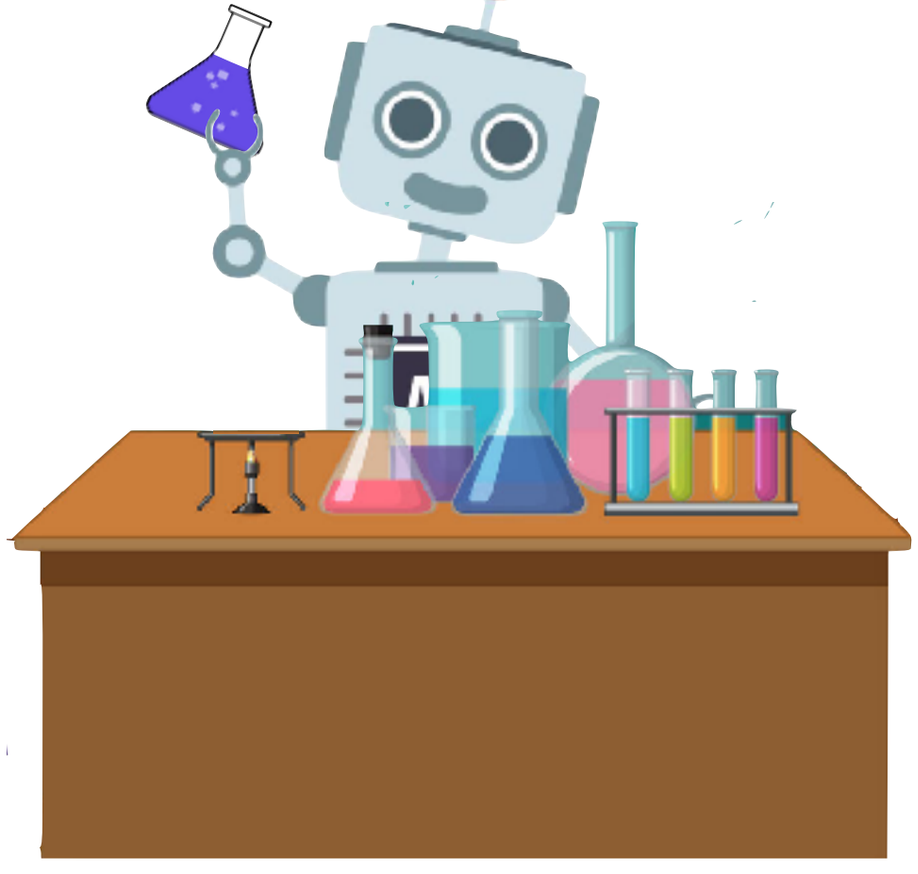
Eric Nalisnick

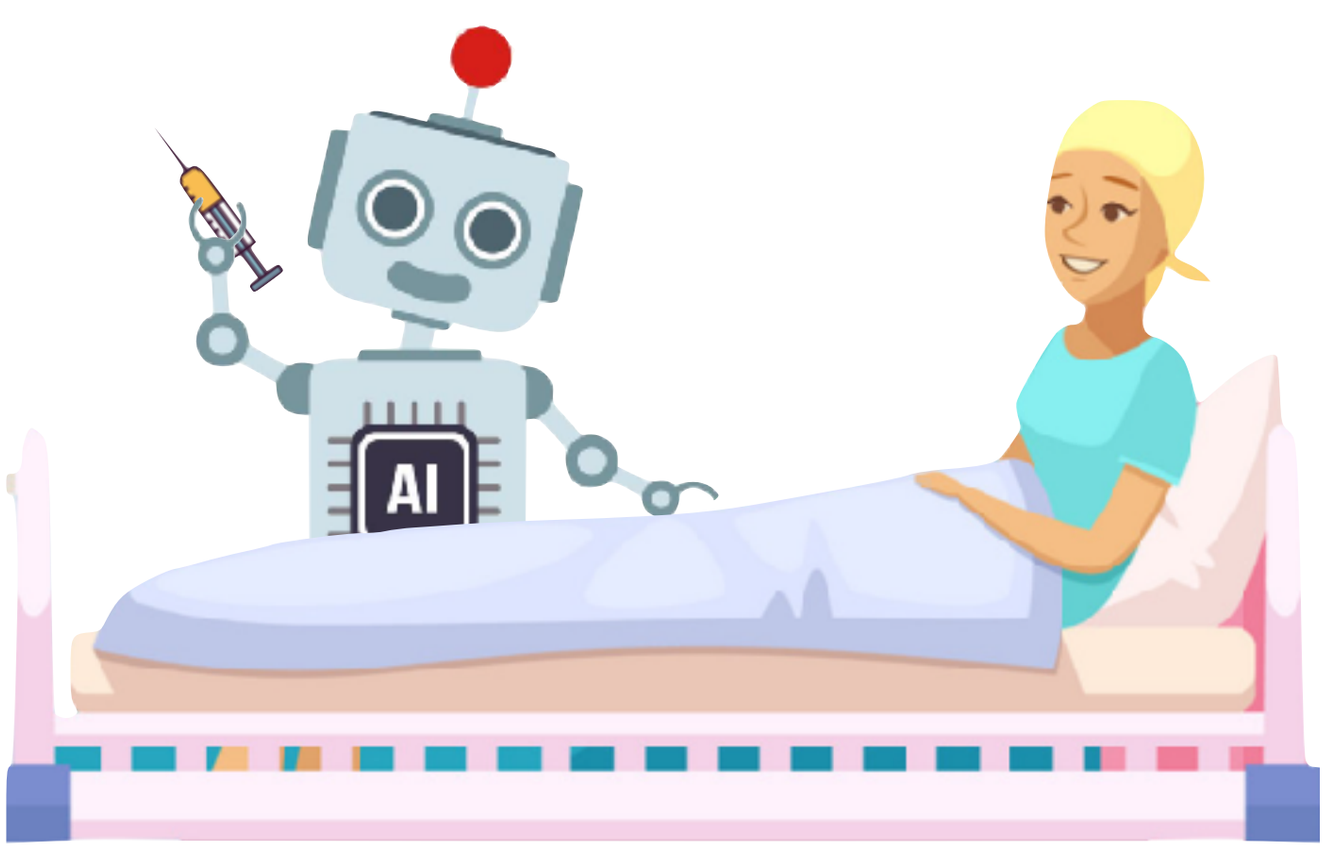
Amsterdam Machine Learning Lab

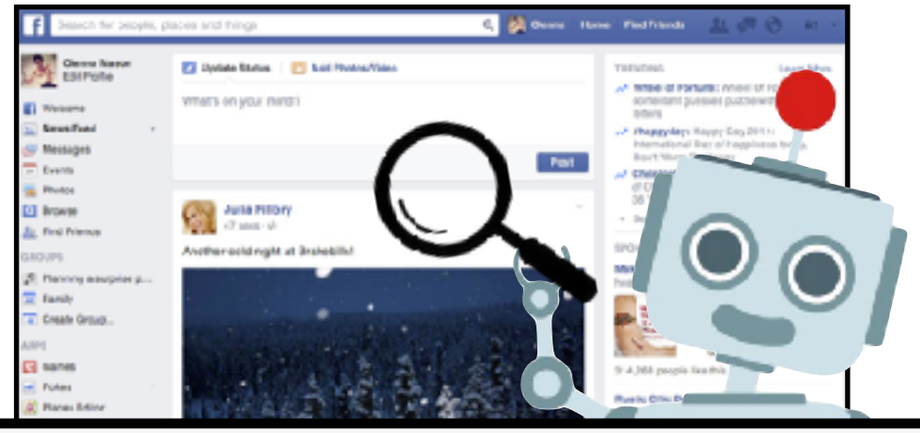
Rajeev Verma




AI

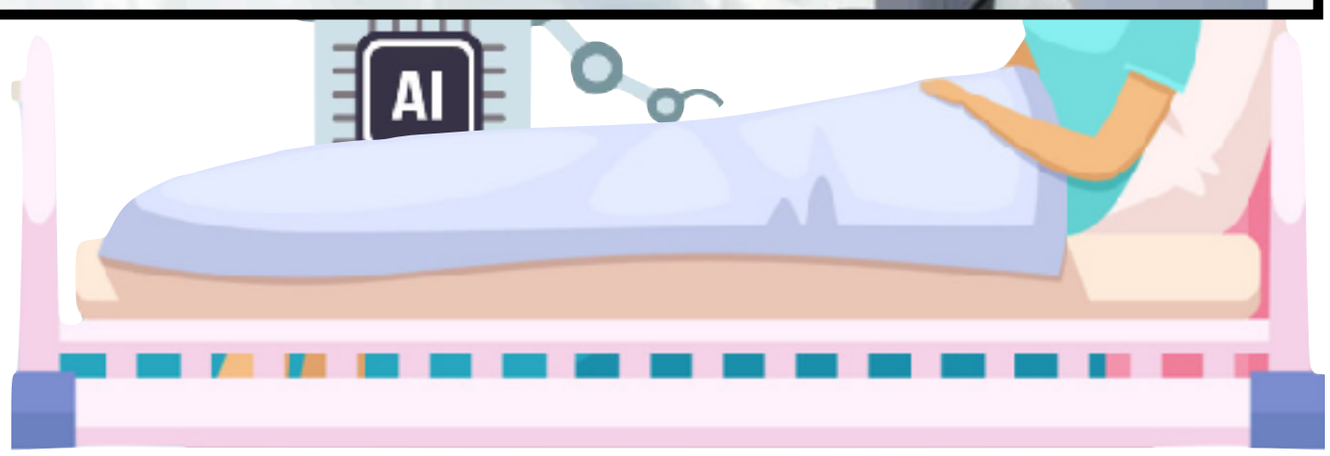


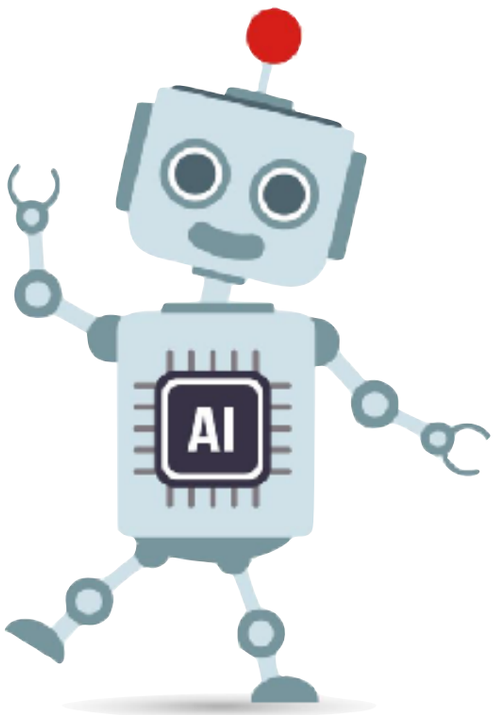

AI



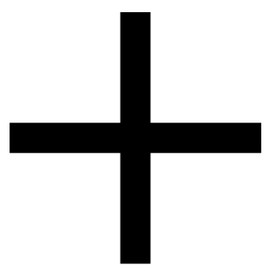
Tesla recall: ‘Full Self-Driving’ software runs stop signs

By TOM KRISHER an hour ago





Model



Human

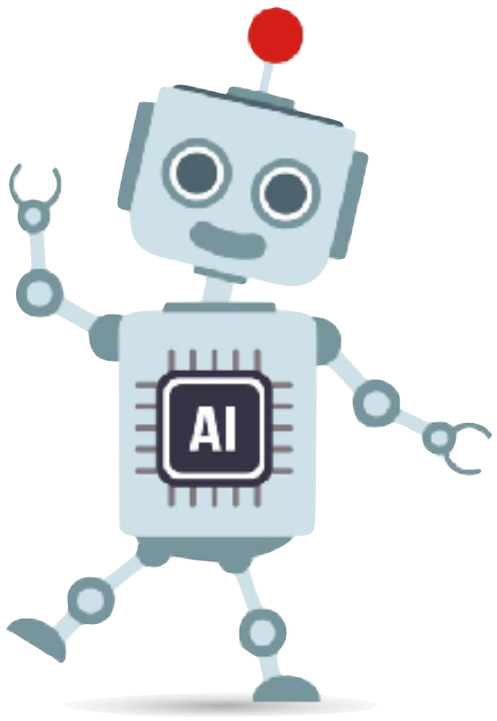
~~Artificial Intelligence~~

Hybrid Intelligence

Junior and senior radiologists benefit from AI assistance when identifying pulmonary nodules

Hannah Murphy | *January 03, 2022* | *AI*

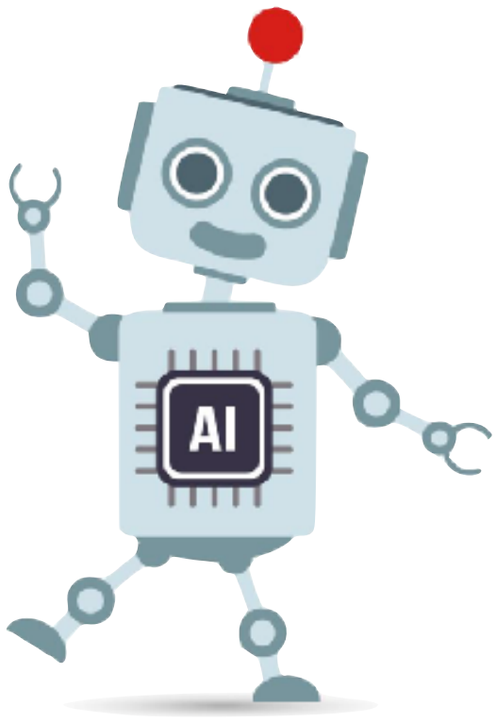
Radiologists benefit from AI-assistance when identifying pulmonary nodules on chest radiographs regardless of their experience level, according to recent research. Since most early lung cancers initially present as pulmonary nodules, these new findings could benefit patients.



Model

?

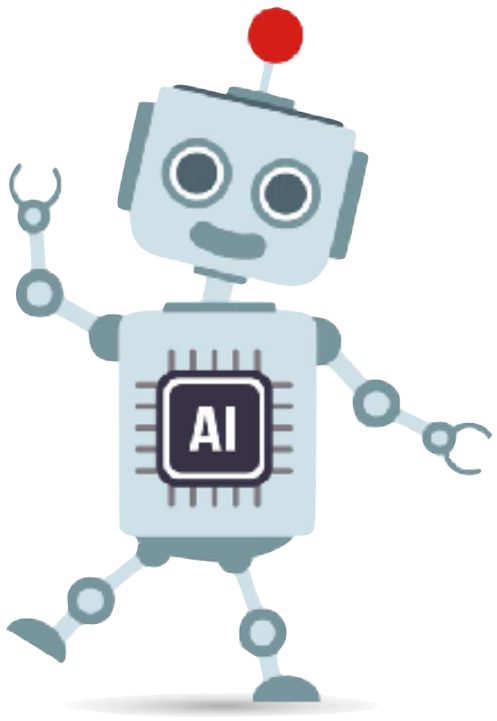
Human



Classifier

?

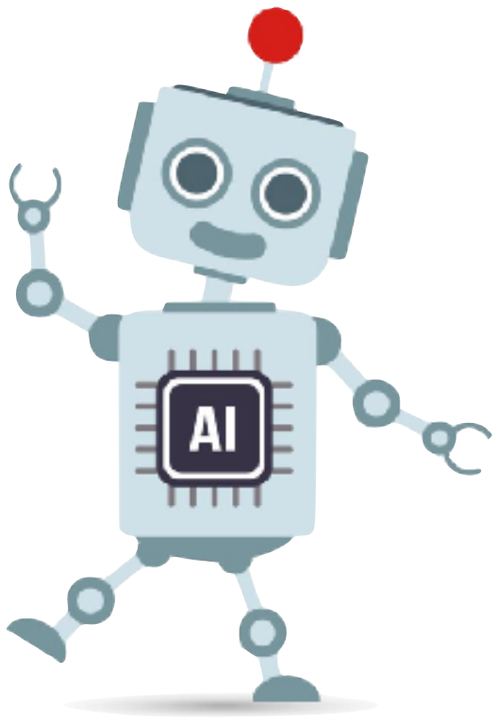
Human



Classifier

?

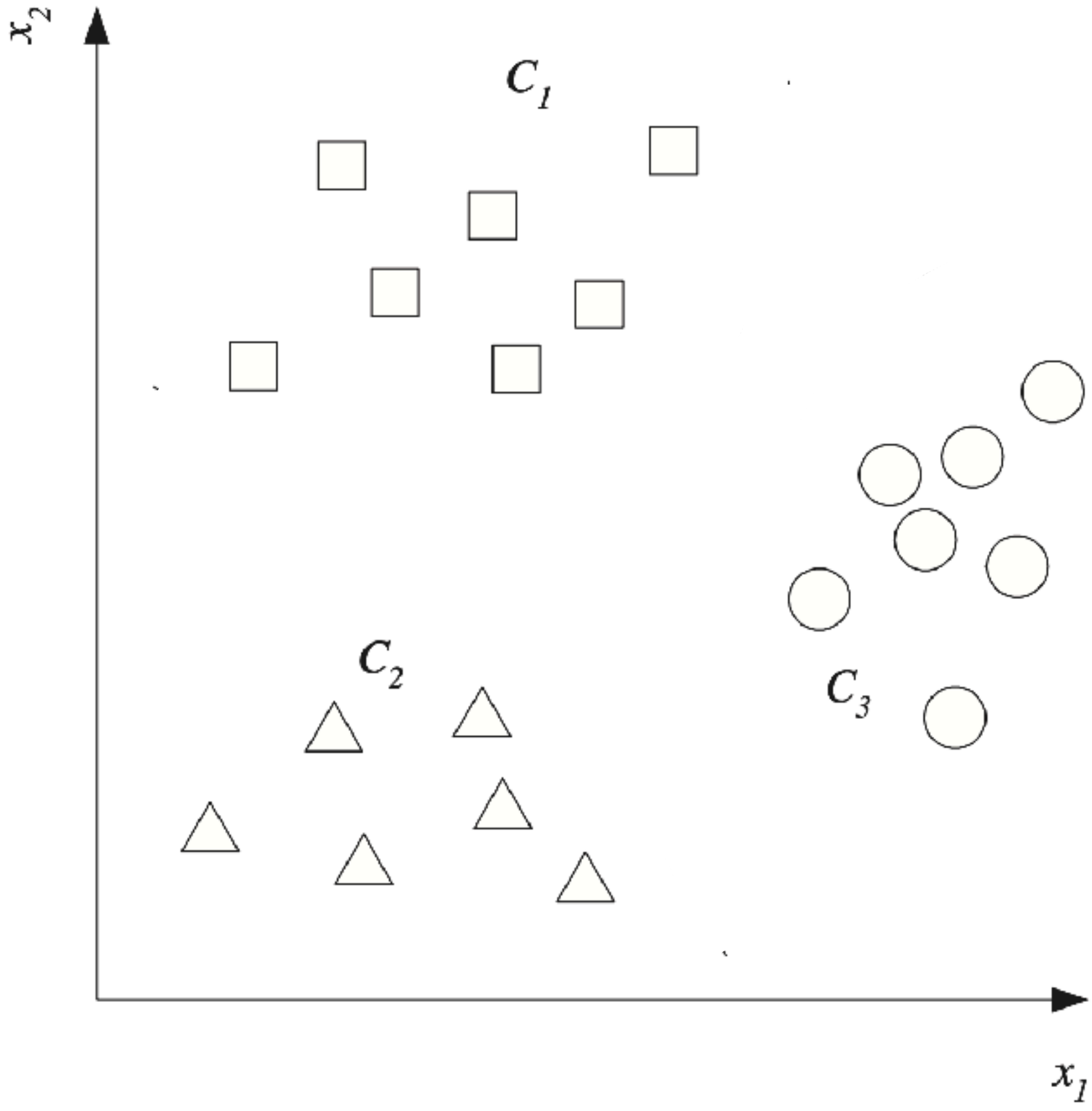
Expert

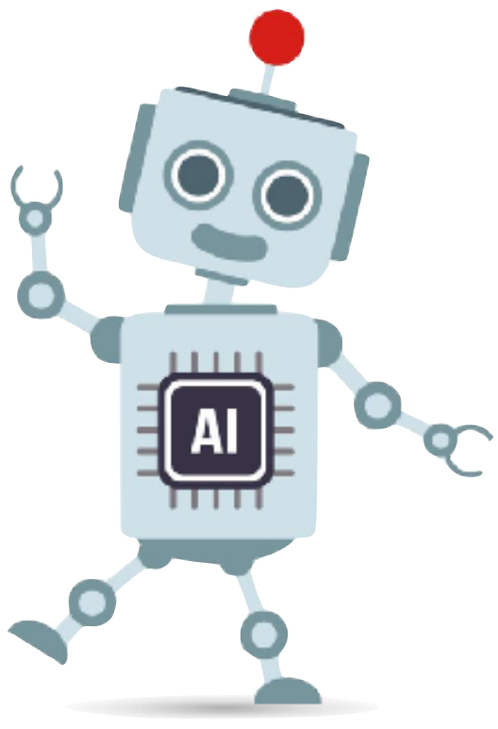


Classifier

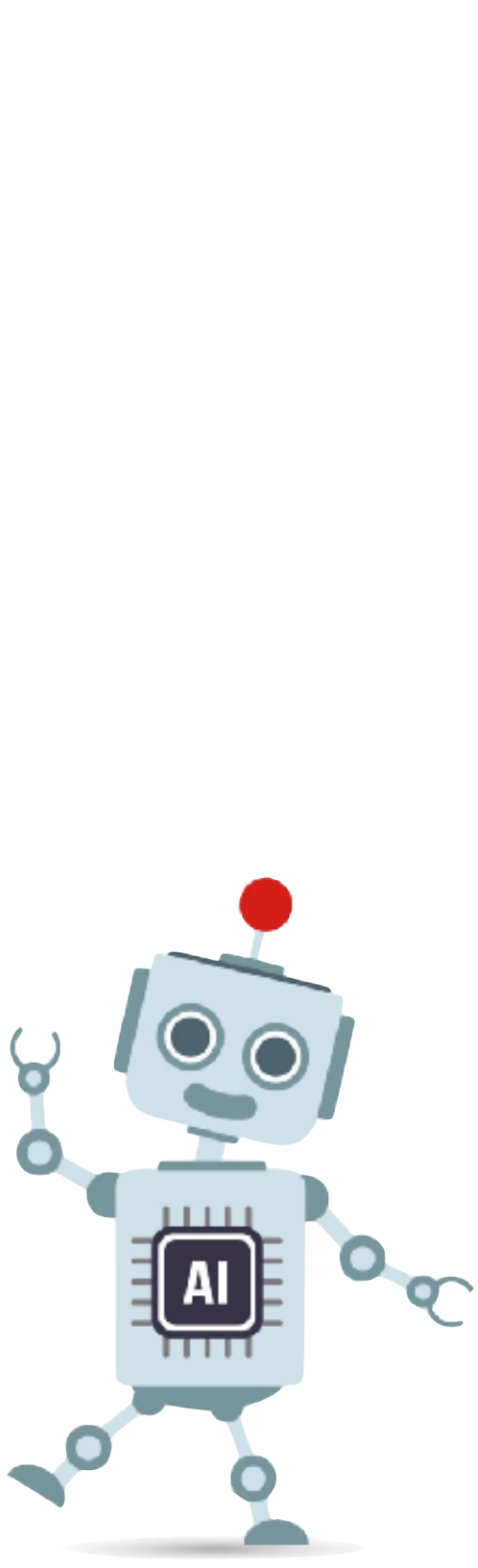
?

Expert

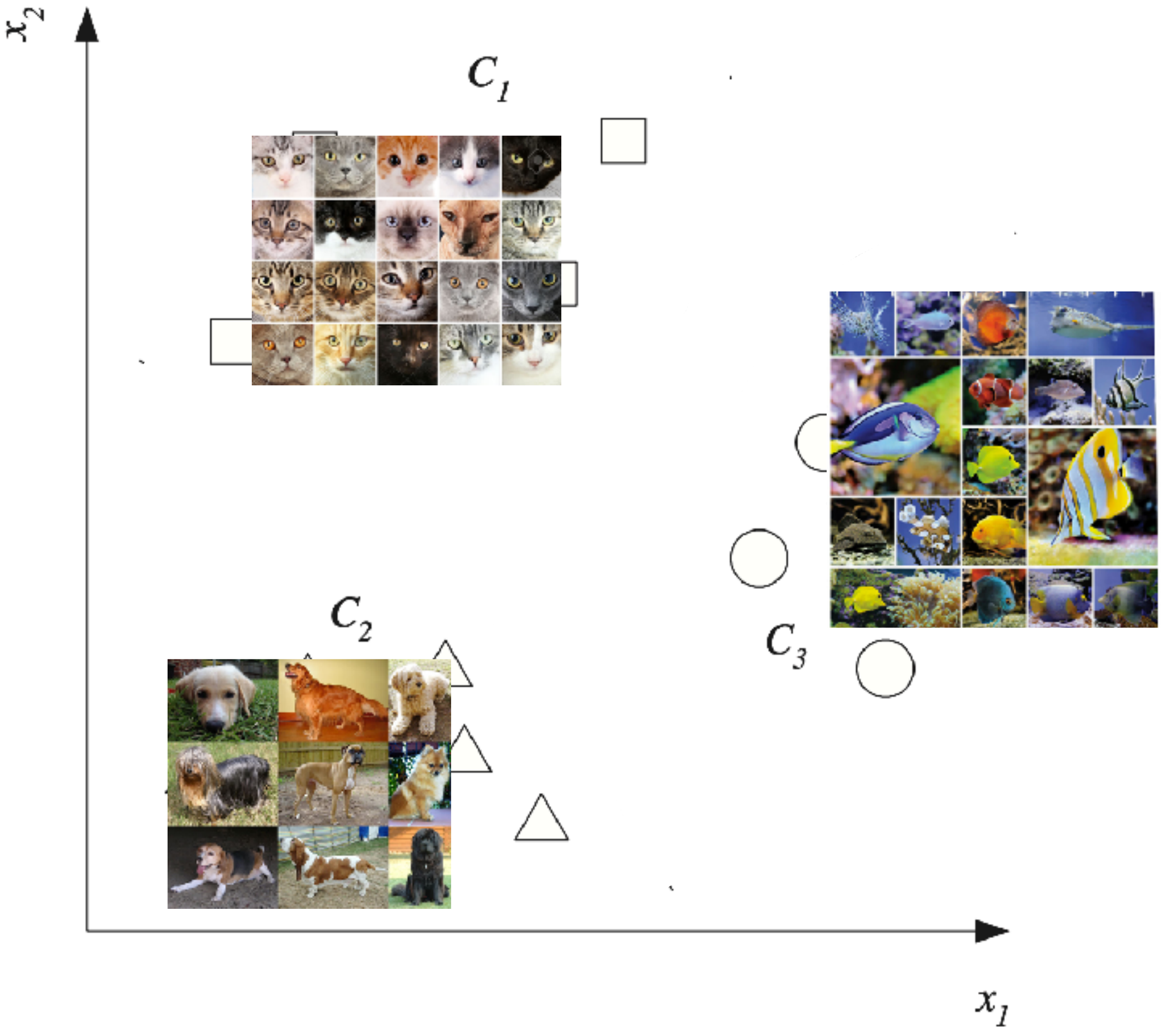


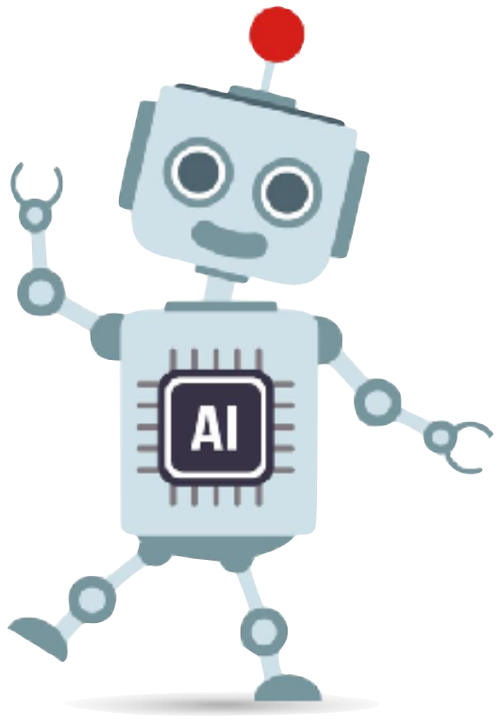


Classifier

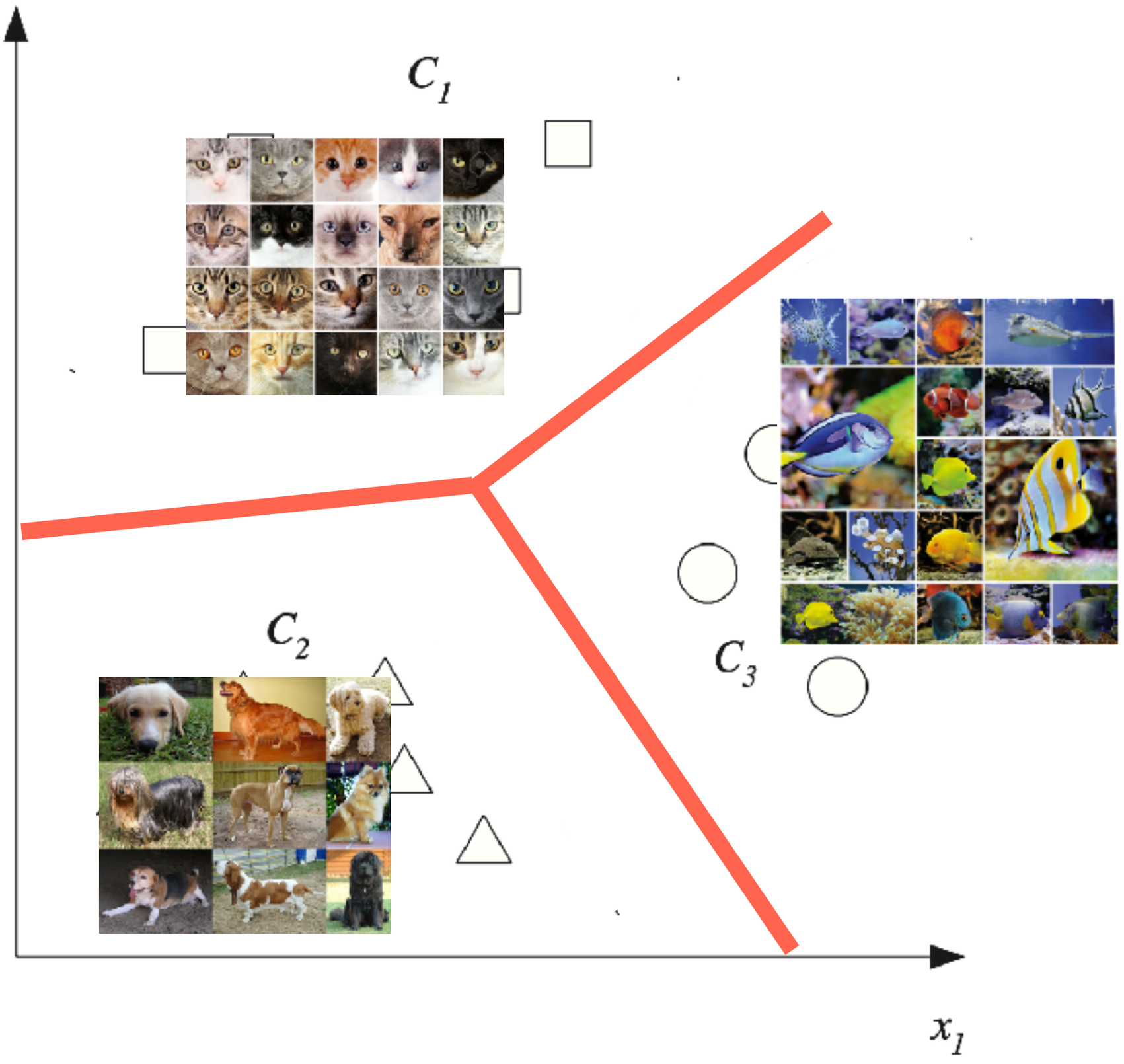


Classifier

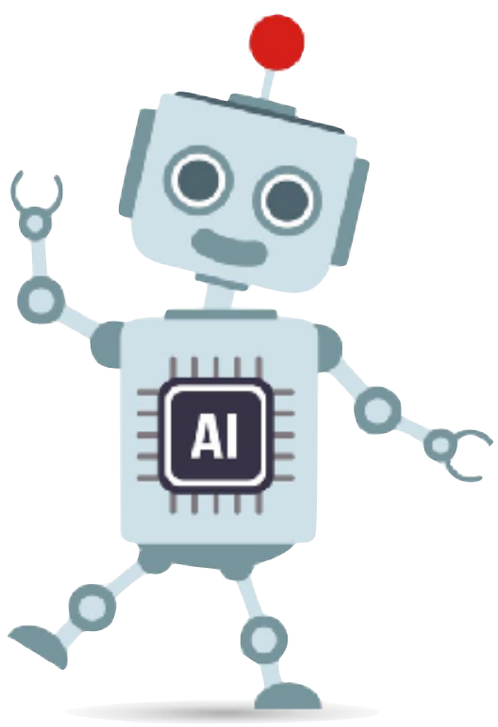




Classifier

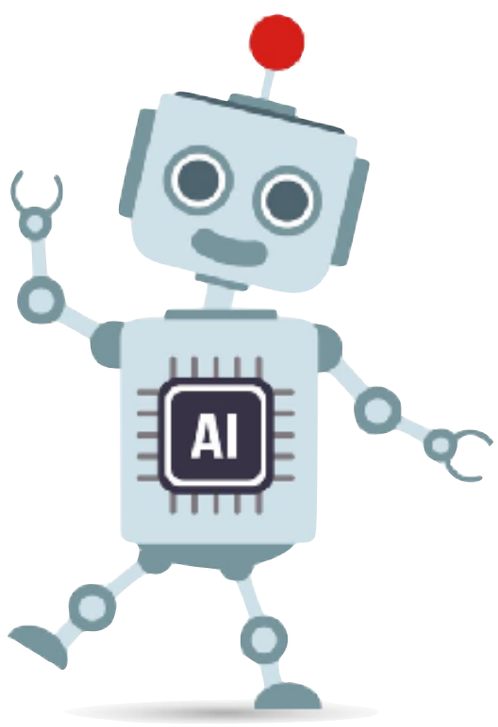


Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$



Classifier

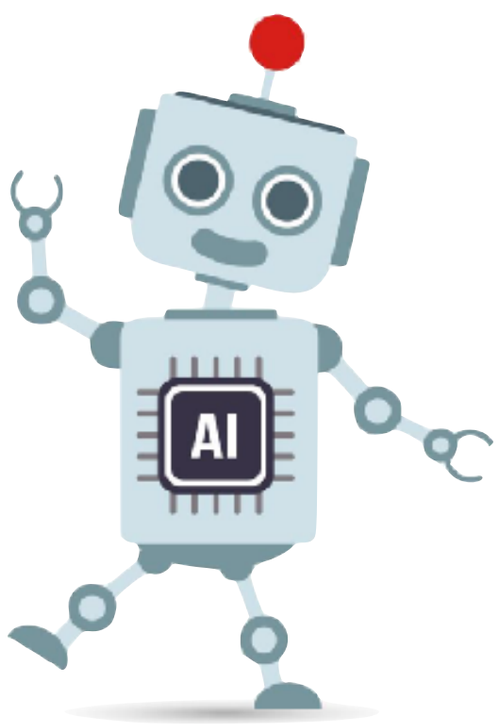
Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$



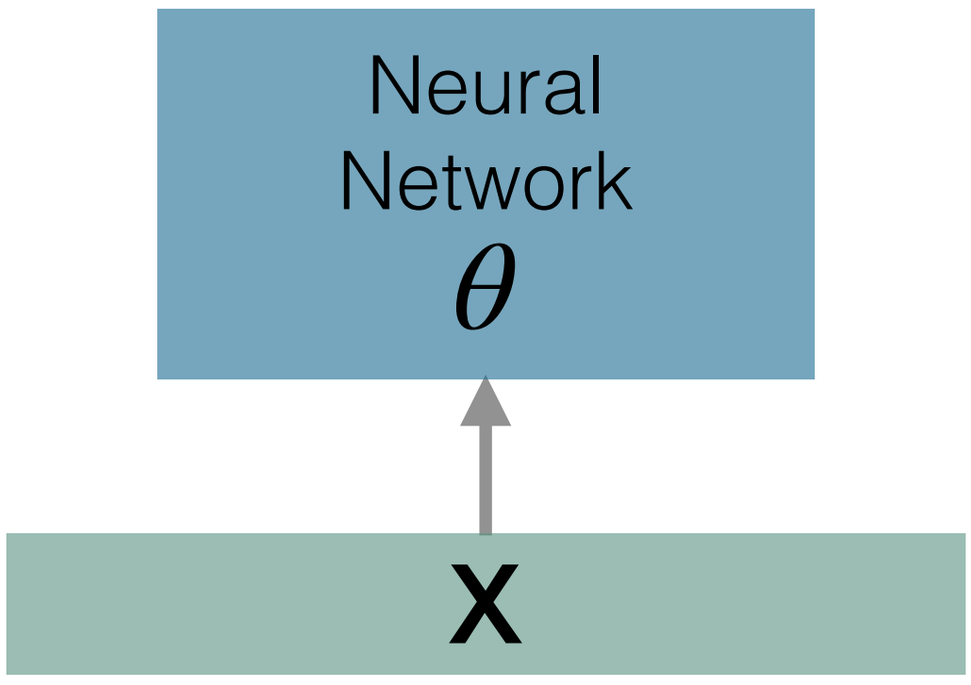
Classifier

X

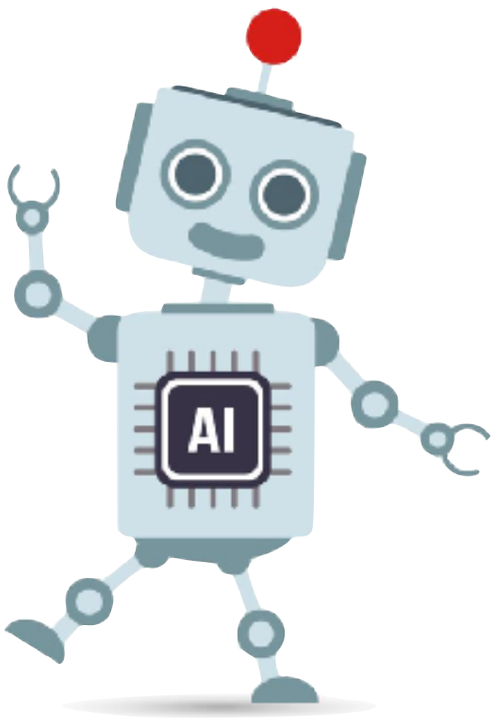
Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$



Classifier

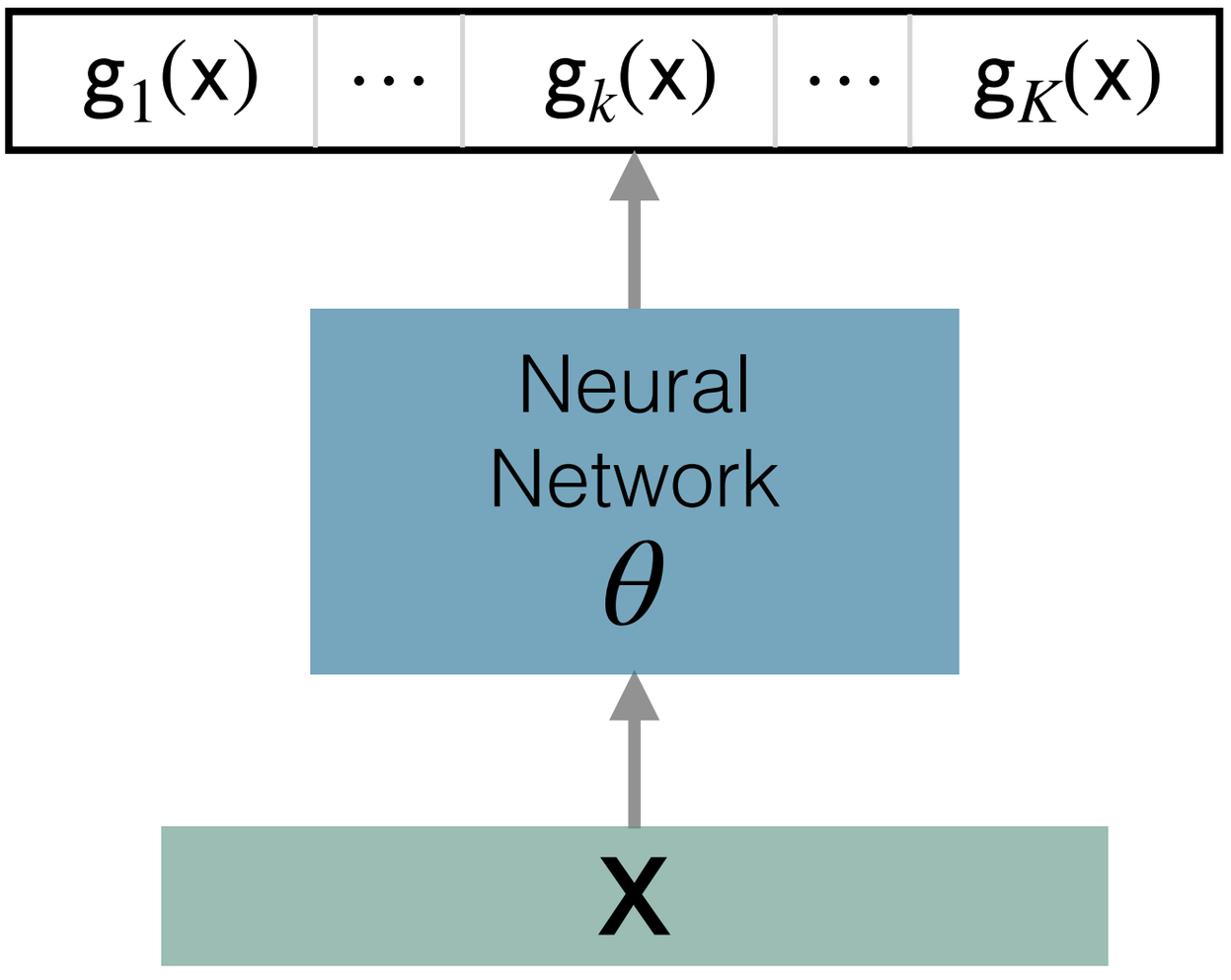


Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$



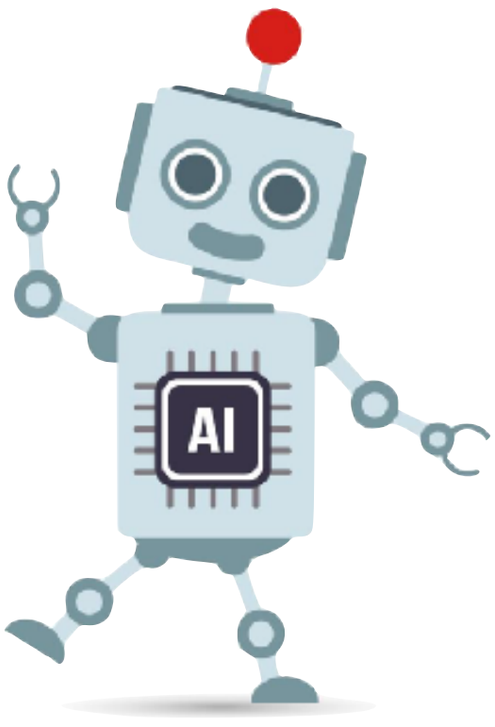
Classifier

$$\mathbf{g}_k(\mathbf{x}) \in \mathbb{R}$$

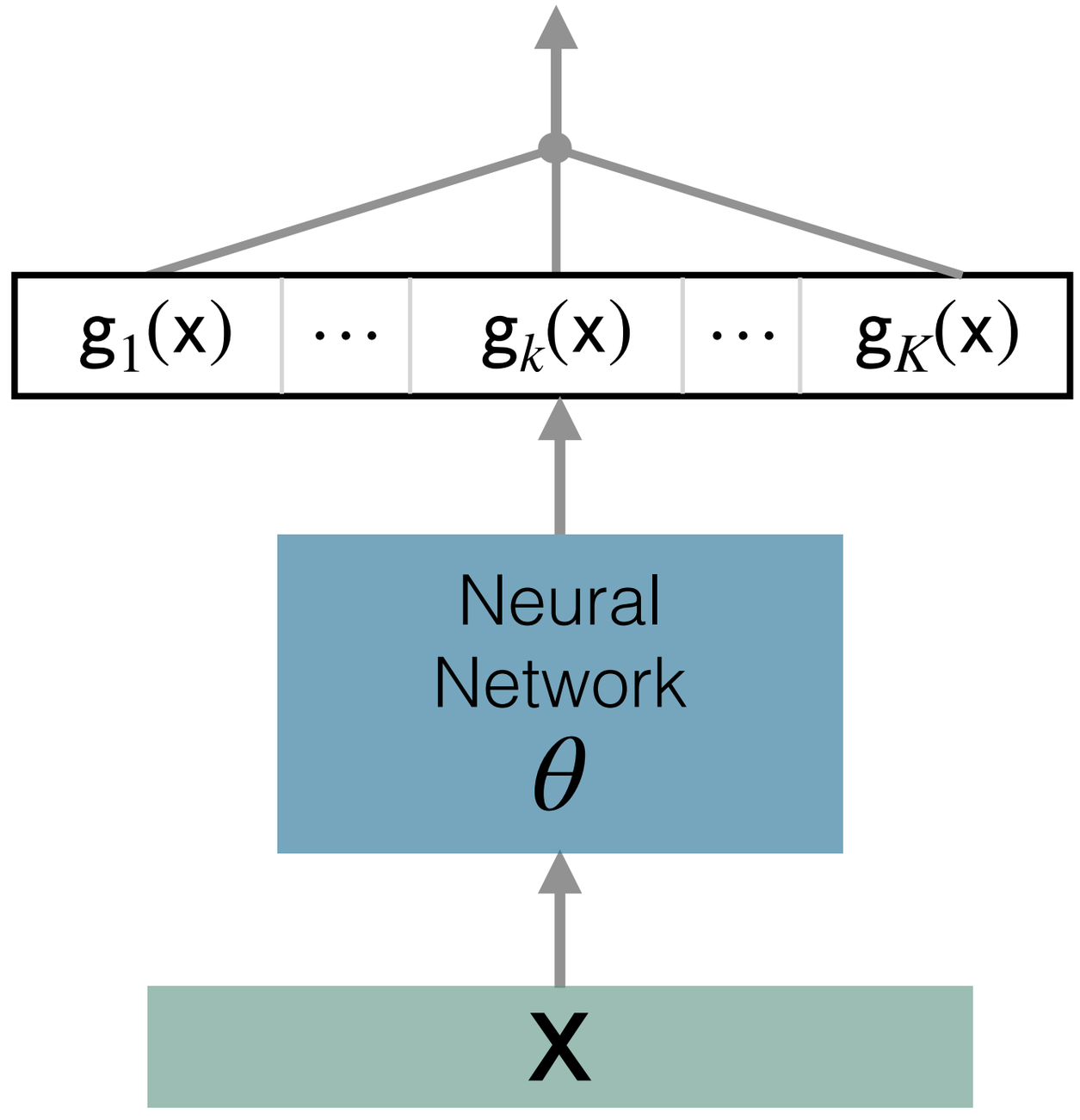


Goal: For input features **x**, predict membership in one of *K* classes: $C_1, \ldots, C_K$

$$P(C_i \mid \mathbf{x}) = \frac{\exp\{\mathbf{g}_i(\mathbf{x})\}}{\sum_{k=1}^{K} \exp\{\mathbf{g}_k(\mathbf{x})\}}$$

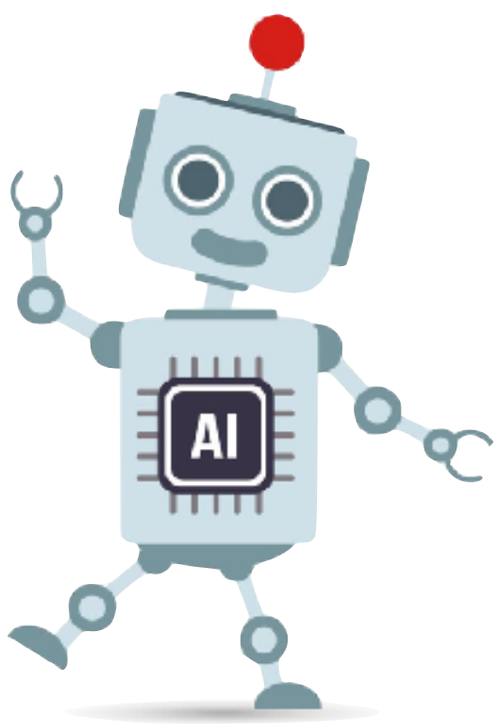


Classifier



Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$

Training: Minimize with respect to θ…



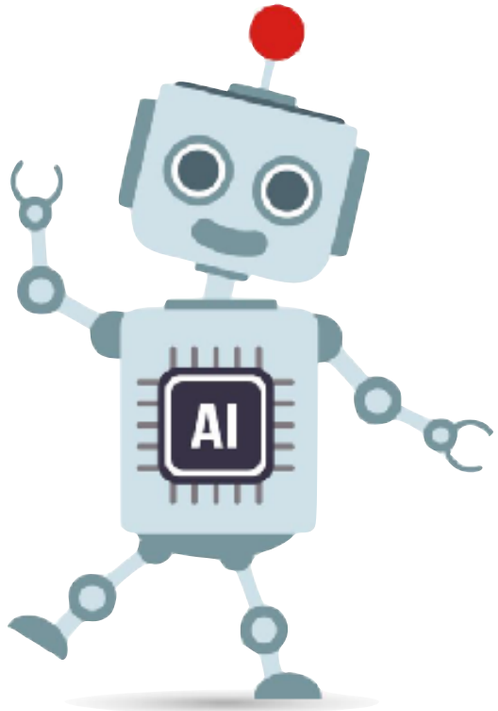
Classifier

$$\ell(\theta; \mathbf{X}, \mathbf{y}) = -\log\left\{\prod_n \mathrm{P}(\mathrm{C}_{y_n} \mid \mathrm{x}_n)\right\}$$

$$= -\sum_n \log \frac{\exp\{\mathrm{g}_{y_n}(\mathrm{x}_n)\}}{\sum_{k=1}^{K} \exp\{\mathrm{g}_k(\mathrm{x}_n)\}}$$

Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$

Training: Minimize with respect to θ…



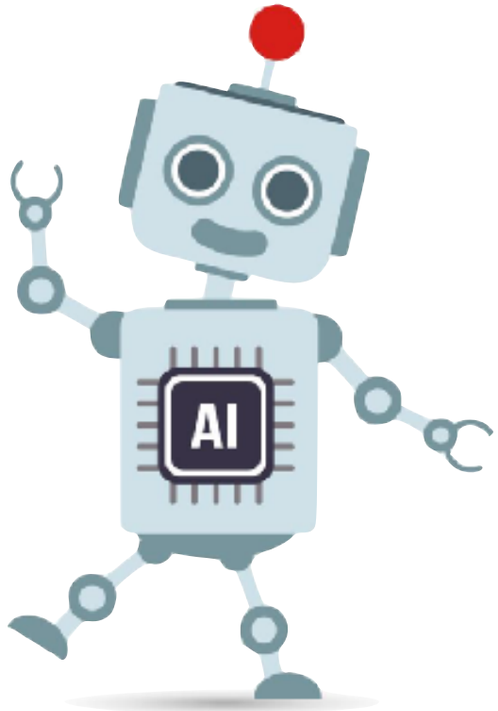
Classifier

$$\ell(\theta; \mathsf{X}, \mathsf{y}) = -\log \left\{ \prod_n \mathsf{P}(\mathsf{C}_{y_n} \mid \mathsf{x}_n) \right\}$$

$$= -\sum_n \log \frac{\exp\{\mathsf{g}_{y_n}(\mathsf{x}_n)\}}{\sum_{k=1}^{K} \exp\{\mathsf{g}_k(\mathsf{x}_n)\}}$$

Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$

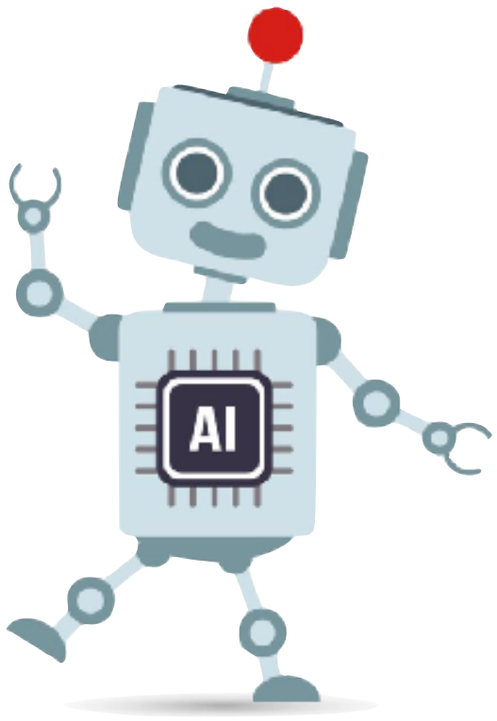
Training: Minimize with respect to θ…



Classifier

$$\ell(\theta; \mathsf{X}, \mathsf{y}) = -\log\left\{\prod_n \mathsf{P}(\mathsf{C}_{y_n} \mid \mathsf{x}_n)\right\}$$

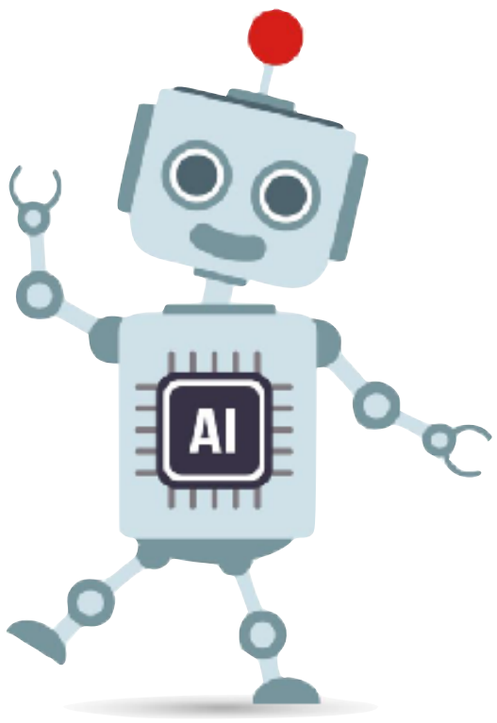
$$= -\sum_n \log \frac{\exp\{\mathbf{g}_{y_n}(\mathsf{x}_n)\}}{\sum_{k=1}^{K} \exp\{\mathbf{g}_k(\mathsf{x}_n)\}}$$



Classifier

?

Expert



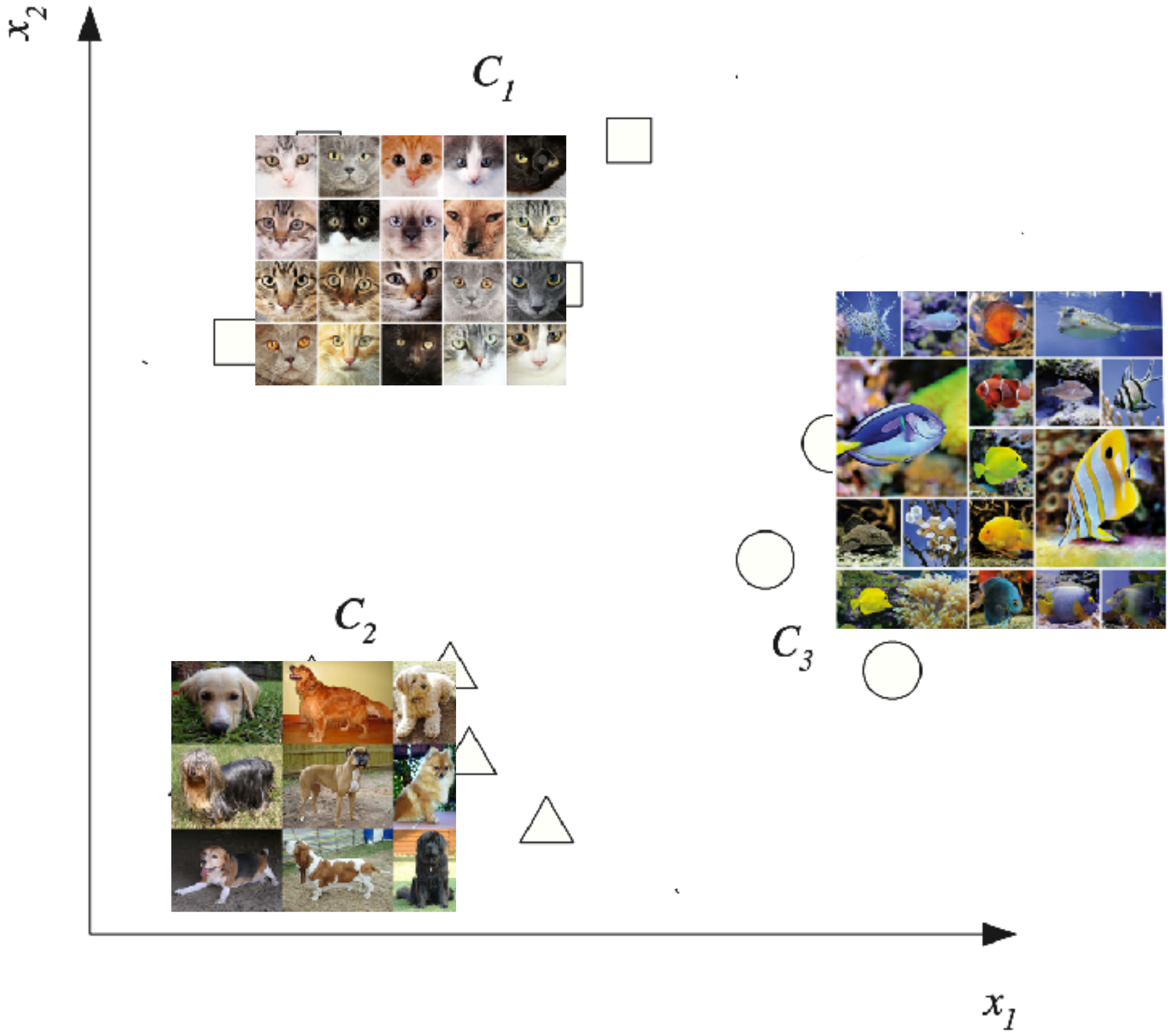
?

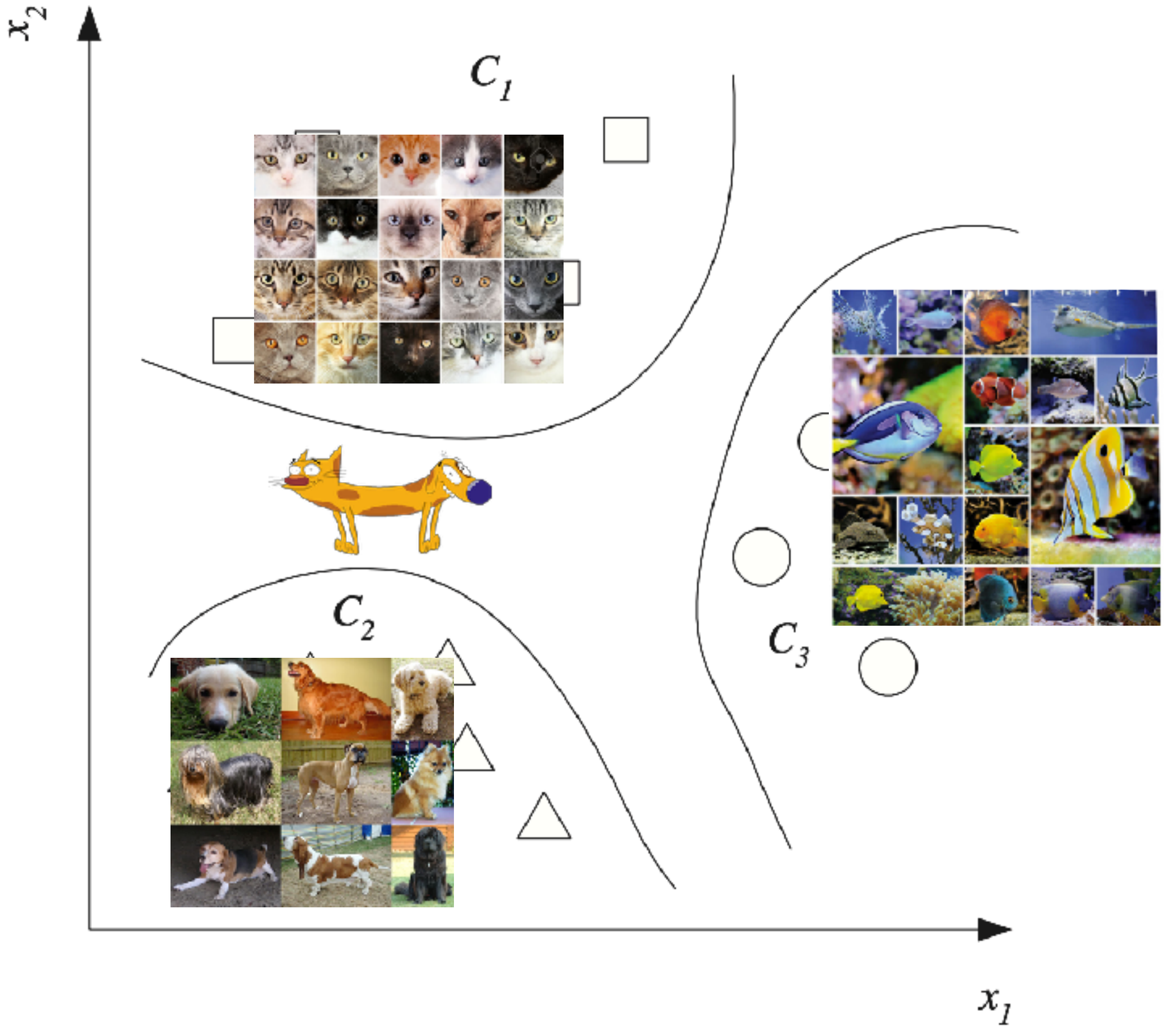
Classifier

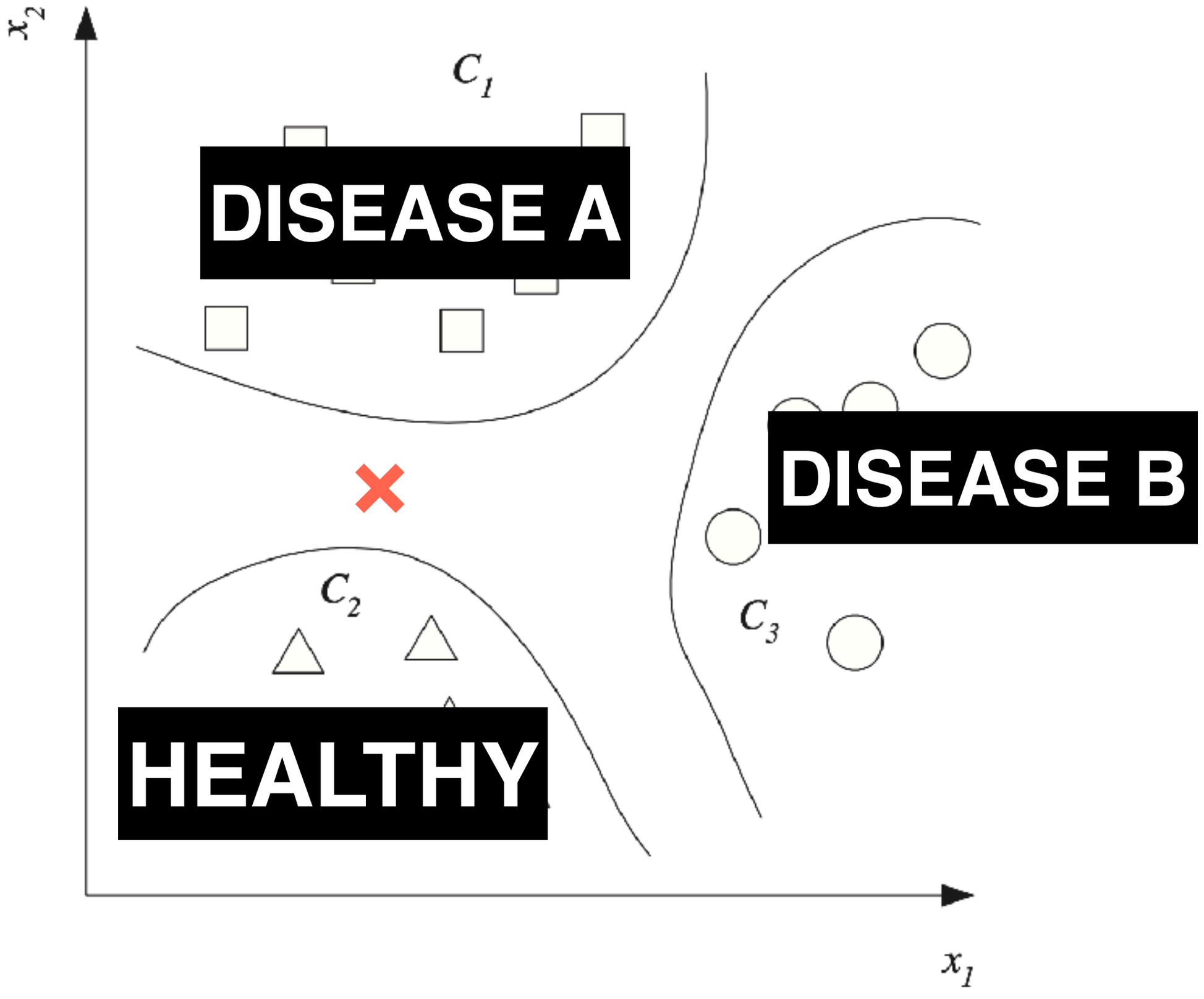
Expert

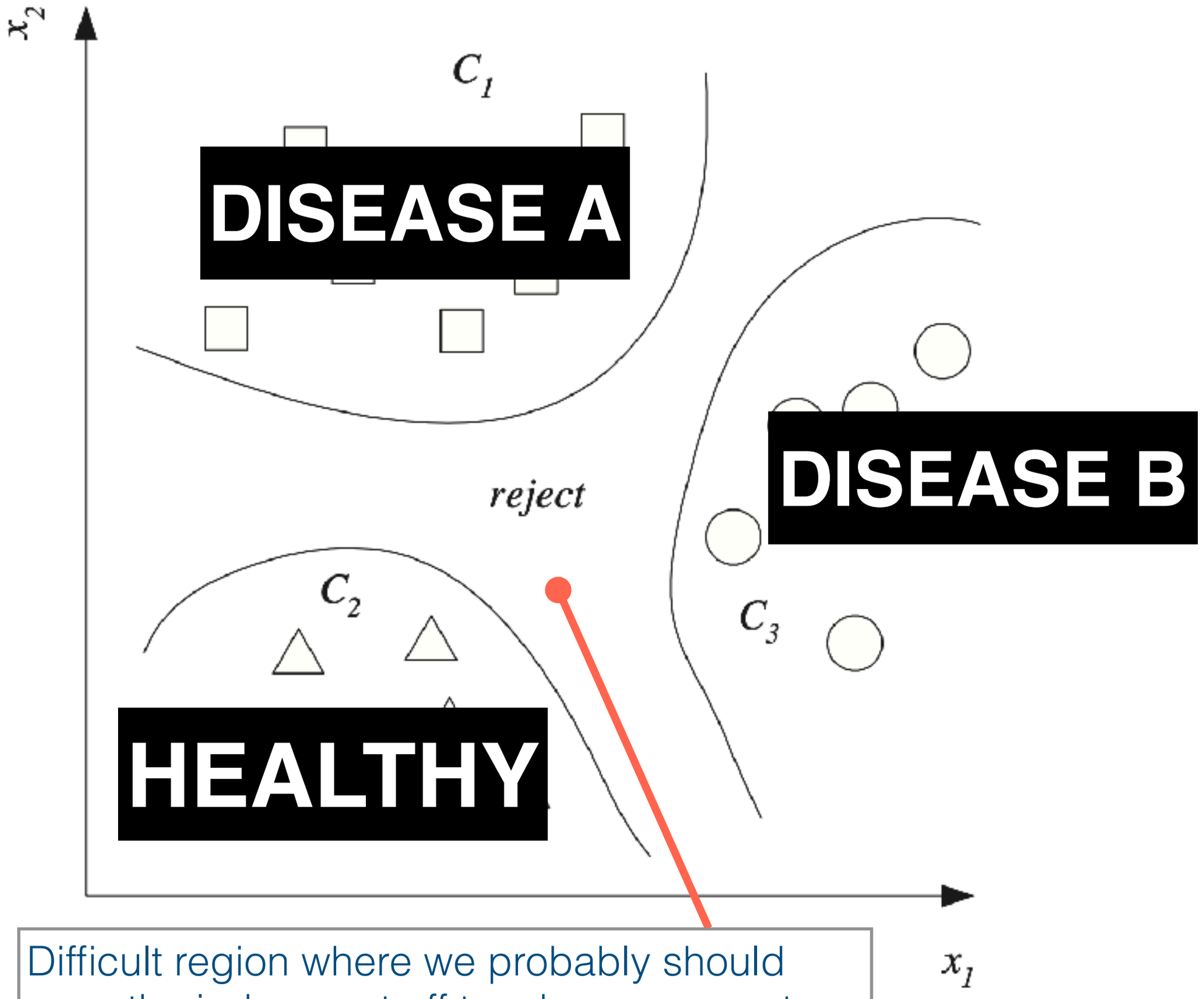
Warm Up:

Classification with a Rejection Option

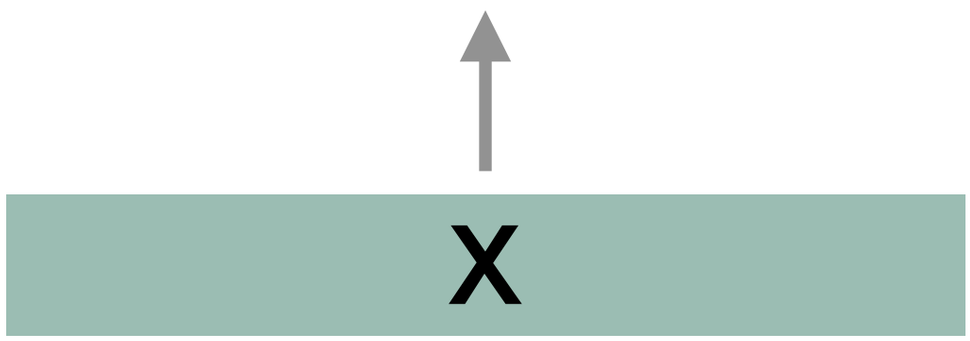

x_2
C_1
C_2
C_3
x_1

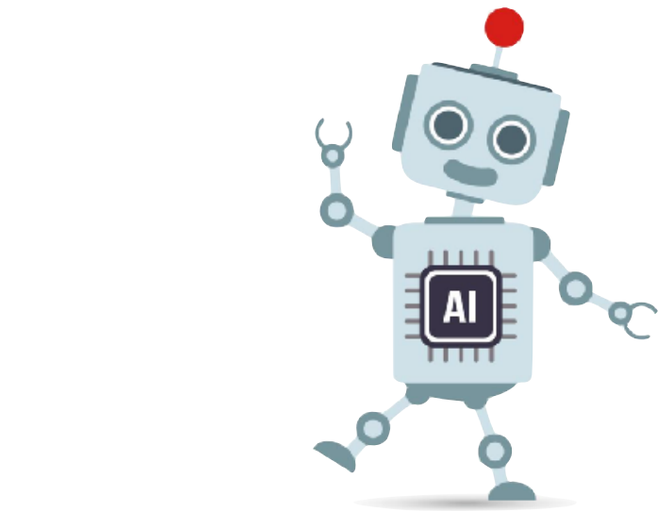

x_2
C_1
C_2
C_3
x_1


x_2
c_1
DISEASE A
DISEASE B
c_2
HEALTHY
c_3
x_1



Difficult region where we probably should pass the judgement off to a human expert

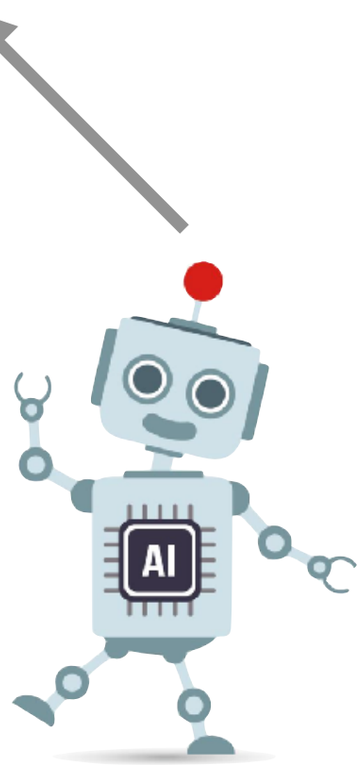

X



Classifier

X

make
prediction

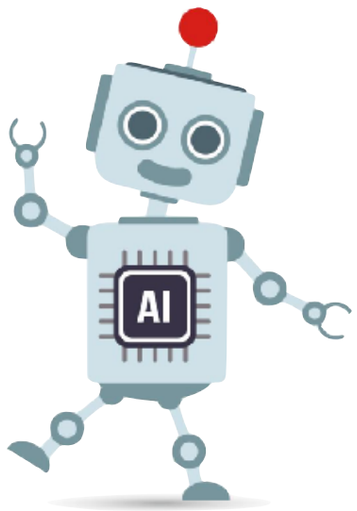


Classifier

X

make prediction

abstain

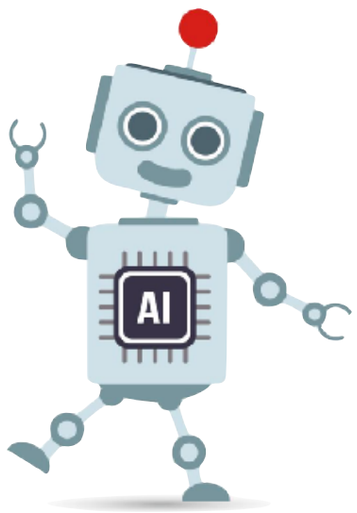


Classifier

X

make prediction

abstain

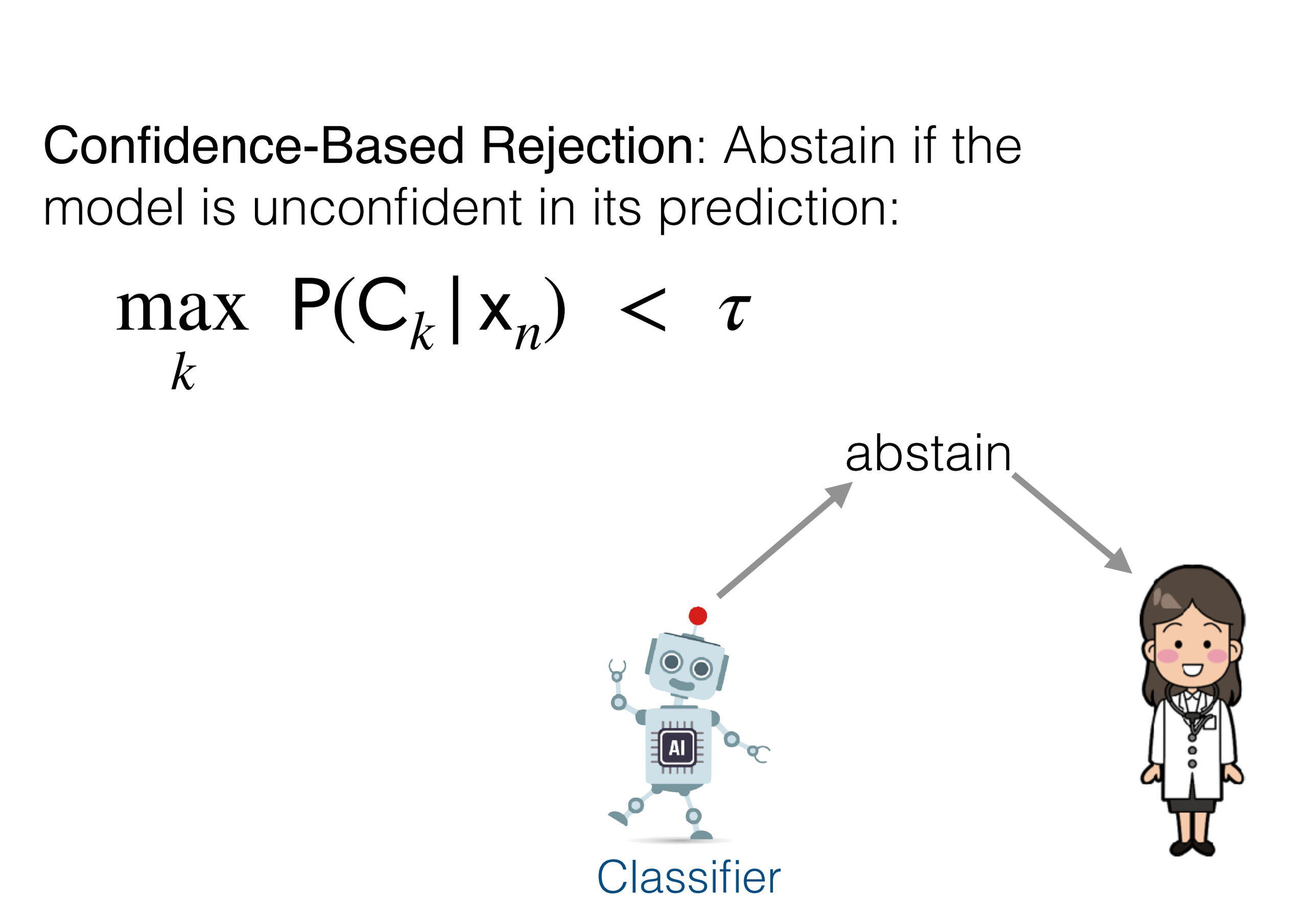


Classifier

X

Confidence-Based Rejection: Abstain if the model is unconfident in its prediction:

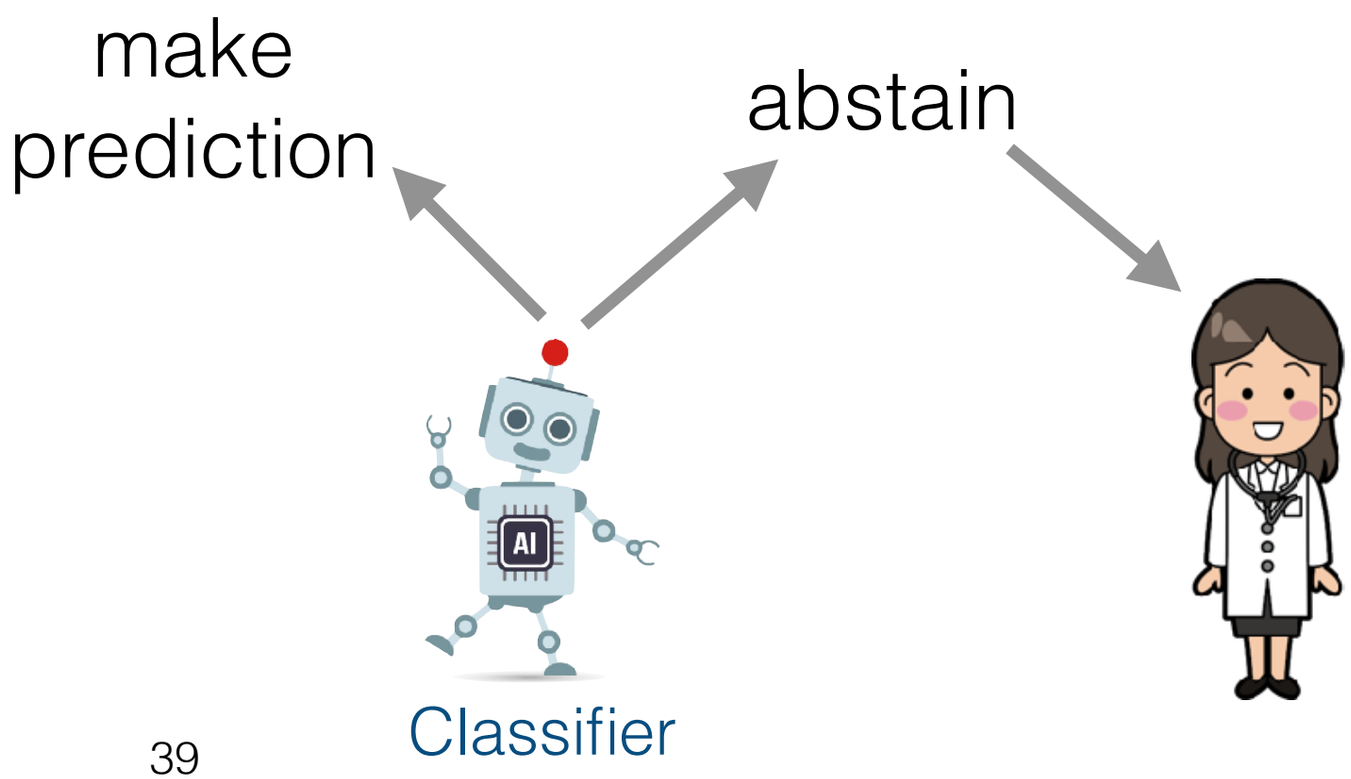
$$\max_k \; \mathrm{P}(\mathrm{C}_k \mid \mathbf{x}_n) \; < \; \tau$$



Learning to Defer to an Expert

Issue with Rejection Option

- ⊗ The human is the “backup plan”: catching the inputs that the classifier is unsure about.



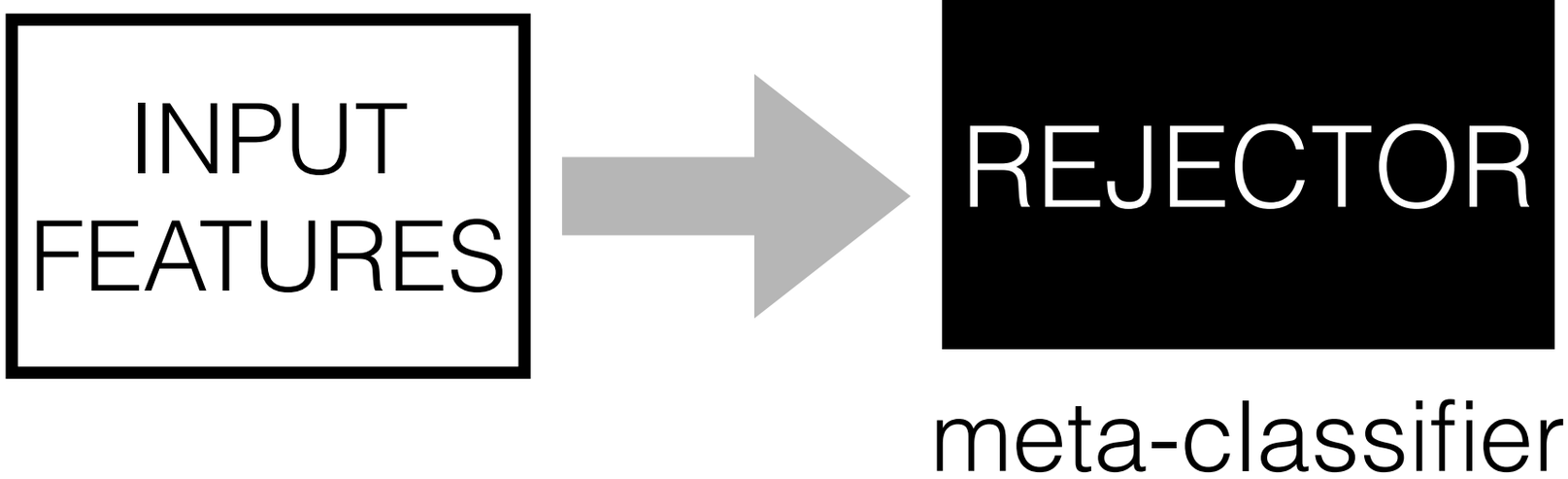
Issue with Rejection Option

- The human is the “backup plan”: catching the inputs that the classifier is unsure about.

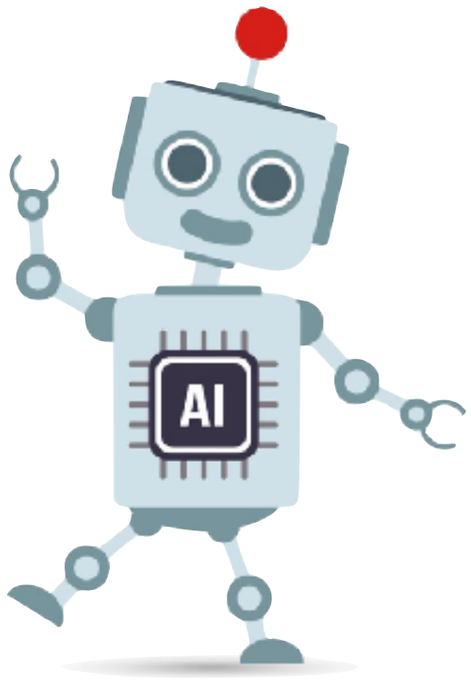
Better Formulation?

- *Model* what the human knows as well, so we can enable *collaboration*.

Learning to Defer



Learning to Defer



Classifier

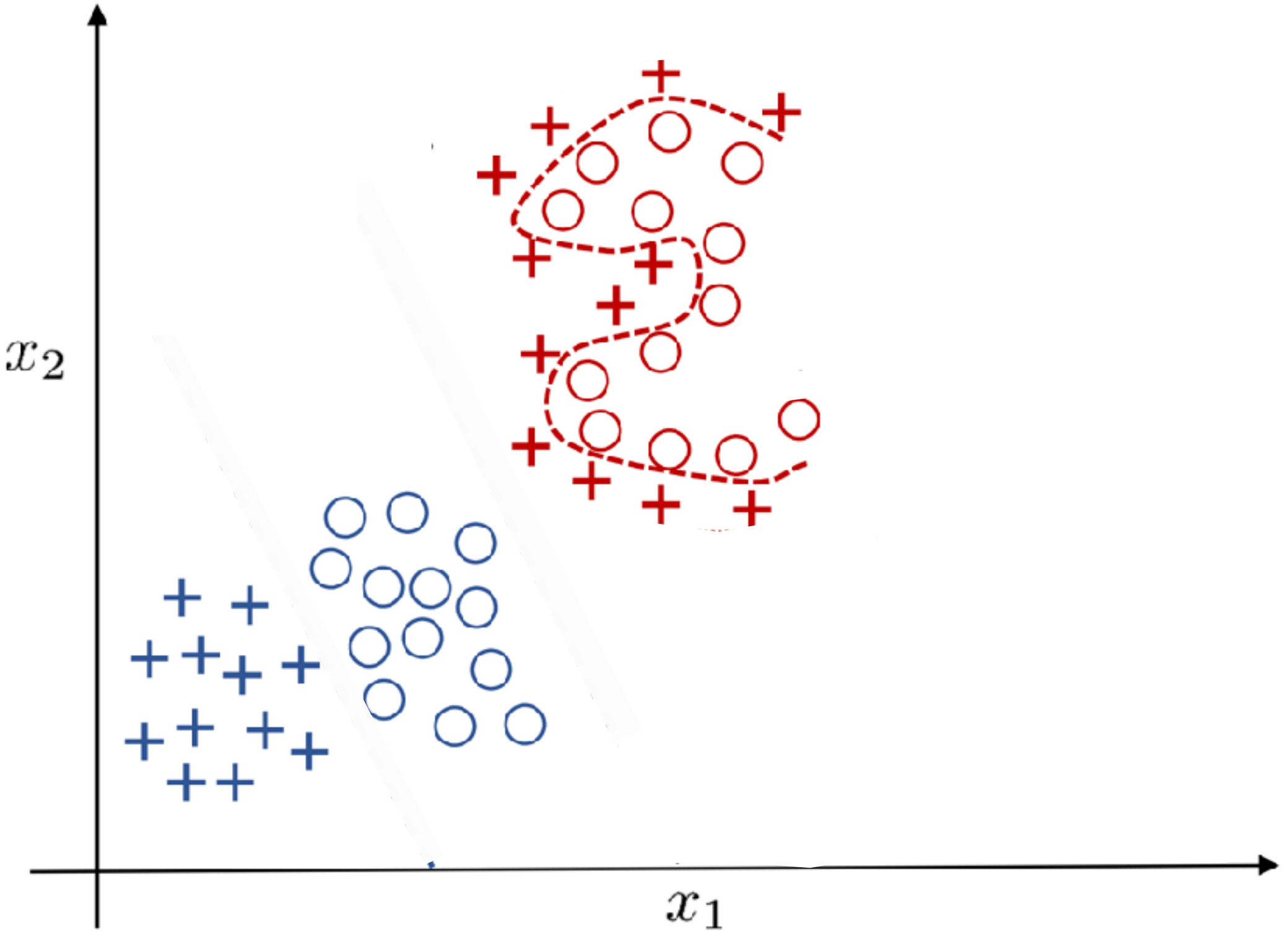
INPUT FEATURES

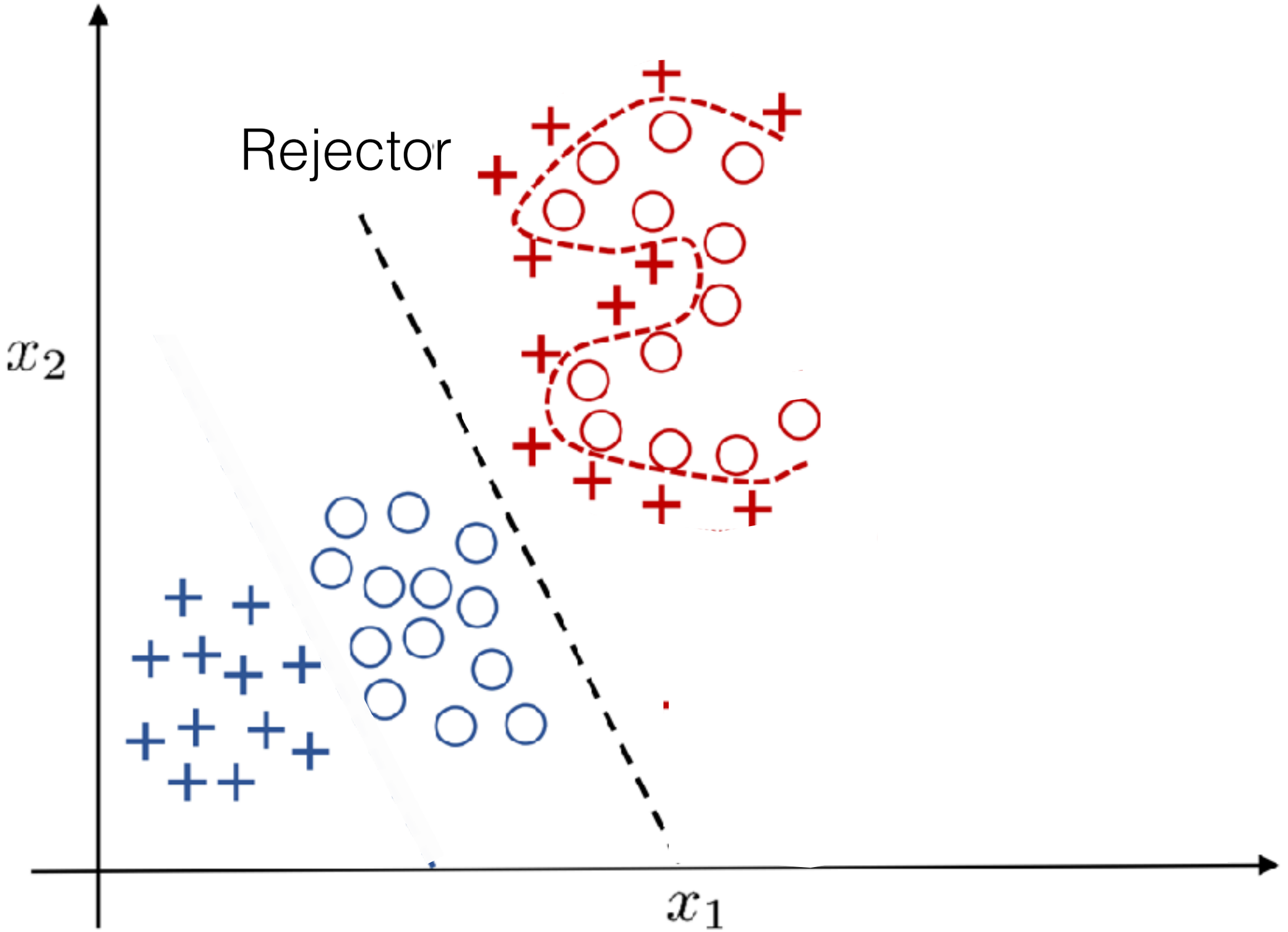
REJECTOR

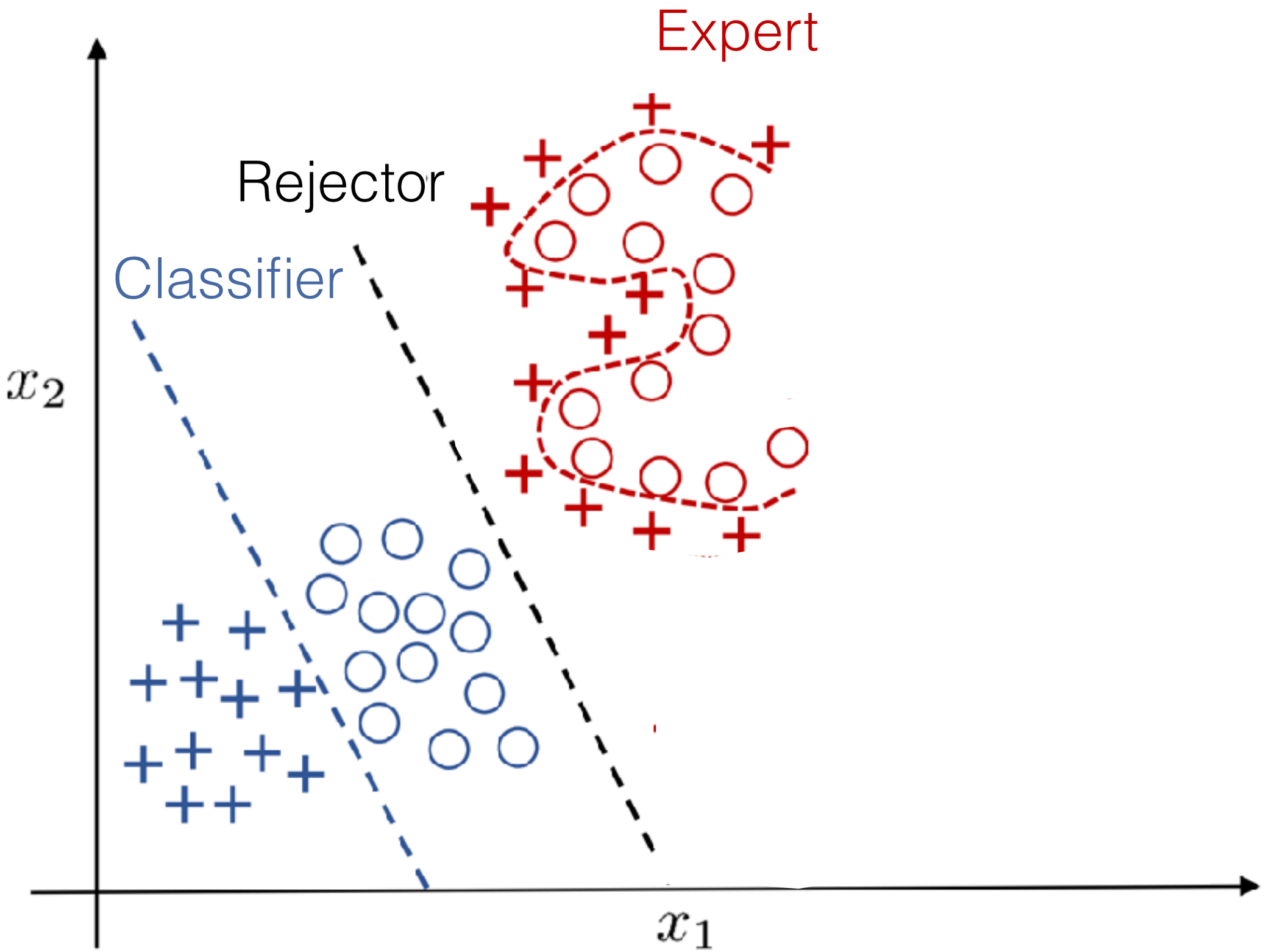
meta-classifier

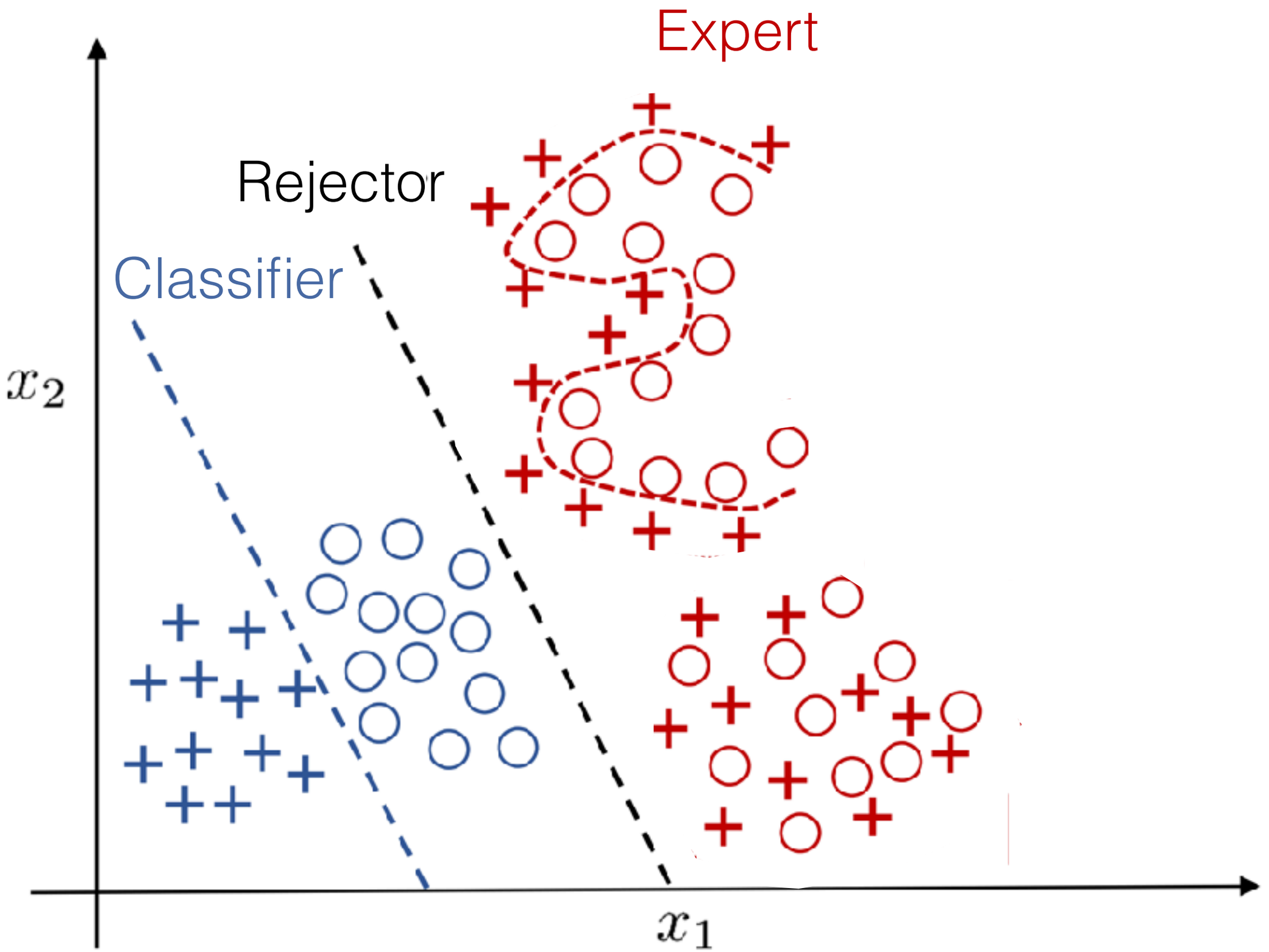
?

Expert

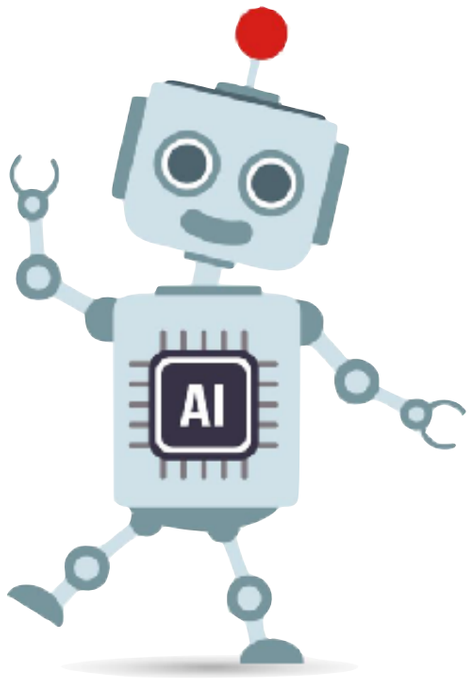

x_2
x_1


Rejector
x_2
x_1


Expert
Rejector
Classifier
x_2
x_1


Expert
Rejector
Classifier
x_2
x_1

Learning to Defer



Classifier

coupled learning

INPUT FEATURES

REJECTOR

meta-classifier

?

Expert

Learning to Defer

Data: $\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$

expert predictions

Learning to Defer

Data: $\mathfrak{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n, \mathbf{m}_n \right\}_{n=1}^{N}$

expert predictions

Models: $\mathbf{r}(\mathbf{x})$ $\mathbf{h}(\mathbf{x})$

Rejector Classifier

Learning to Defer

0-1 Loss:

$$\ell(\mathsf{r}, \mathsf{h}; \mathfrak{D}) =$$

$$\sum_n (1 - \mathsf{r}(\mathsf{x}_n))\, \mathbb{I}[\mathsf{h}(\mathsf{x}_n) \neq \mathsf{y}_n] \;+\; \mathsf{r}(\mathsf{x}_n)\, \mathbb{I}[\mathsf{m}_n \neq \mathsf{y}_n]$$

classifier loss

expert loss

Learning to Defer

0-1 Loss:

$$\ell(\mathsf{r}, \mathsf{h}; \mathfrak{D}) =$$

$$\sum_n (1 - \mathsf{r}(\mathsf{x}_n))\, \mathbb{I}[\mathsf{h}(\mathsf{x}_n) \neq \mathsf{y}_n] \;+\; \mathsf{r}(\mathsf{x}_n)\, \mathbb{I}[\mathsf{m}_n \neq \mathsf{y}_n]$$

classifier loss

expert loss

Learning to Defer

0-1 Loss:

$$\ell(\mathsf{r}, \mathsf{h}; \mathfrak{D}) =$$

$$\sum_n (1 - \mathsf{r}(\mathsf{x}_n))\ \mathbb{I}[\mathsf{h}(\mathsf{x}_n) \neq \mathsf{y}_n] \ + \ \mathsf{r}(\mathsf{x}_n)\ \mathbb{I}[\mathsf{m}_n \neq \mathsf{y}_n]$$

classifier loss

expert loss

Learning to Defer

Formerly open problem:
what's a consistent surrogate
for the 0-1 loss?

Learning to Defer

Formerly open problem:
what's a consistent surrogate
for the 0-1 loss?

$$\mathbf{h}^*(\mathbf{x}) = \operatorname*{argmax}_{y} \ \mathbb{P}(\mathbf{y} = y \mid \mathbf{x})$$

Learning to Defer

Formerly open problem:
what's a consistent surrogate for the 0-1 loss?

$$\mathsf{h}^*(\mathsf{x}) = \operatorname*{argmax}_y \ \mathbb{P}(\mathsf{y} = y \,|\, \mathsf{x})$$

$$\mathsf{r}^*(\mathsf{x}) = \mathbb{I}\left[\mathbb{P}(\mathsf{m} = \mathsf{y} \,|\, \mathsf{x}) \geq \max_y \ \mathbb{P}(\mathsf{y} = y \,|\, \mathsf{x})\right]$$

probability that the expert is correct

Consistent Estimators for Learning to Defer to an Expert

Hussein Mozannar * David Sontag †

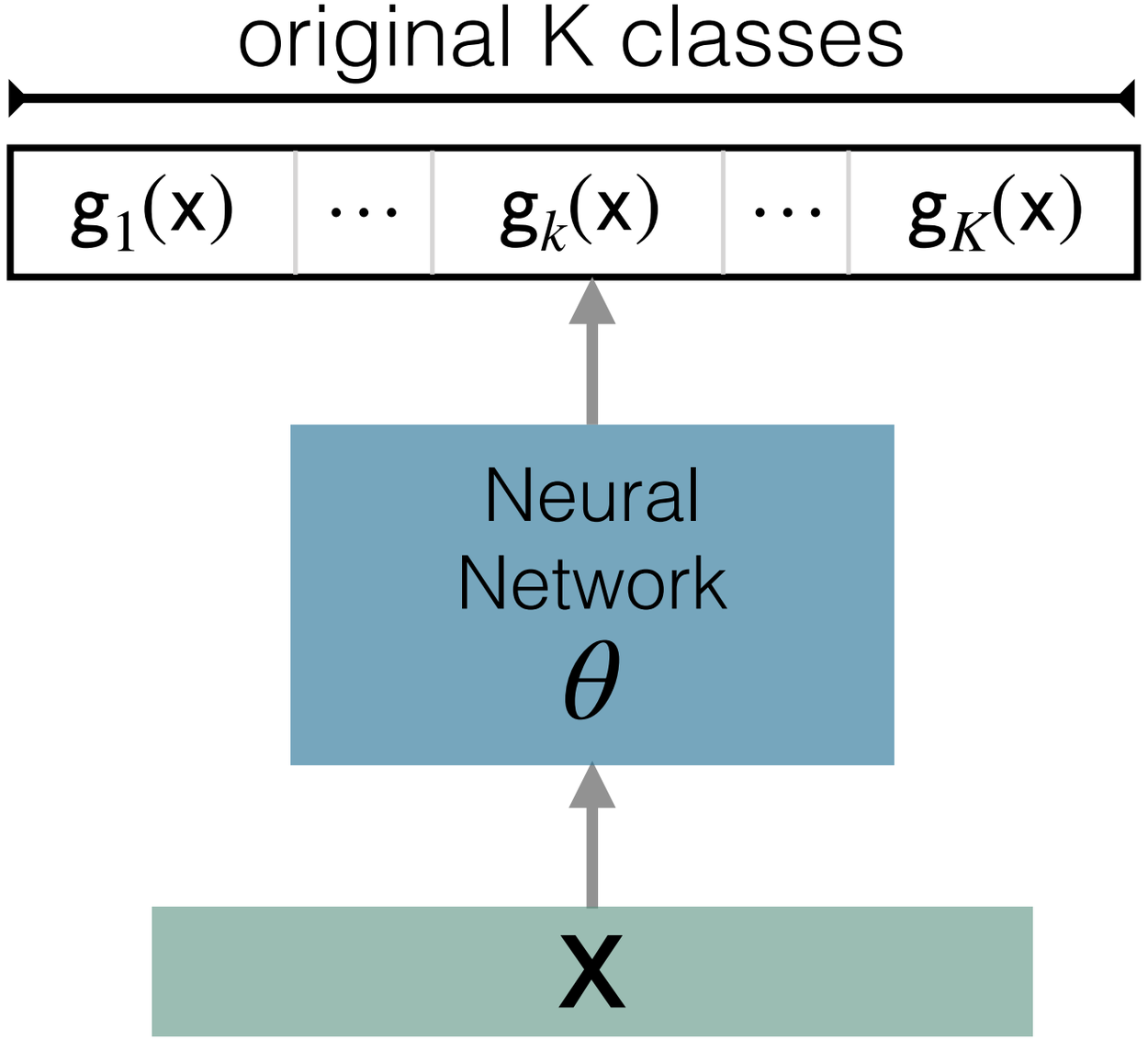
Abstract

Learning algorithms are often used in conjunction with expert decision makers in practical scenarios, however this fact is largely ignored when designing these algorithms. In this paper we explore how to learn predictors that can either predict or choose to defer the decision to a downstream expert. Given only samples of the expert's decisions, we give a procedure based on learning a classifier and a rejector and analyze it theoretically. Our approach is based on a novel reduction to cost sensitive learning where we give a consistent surrogate loss for cost sensitive learning that generalizes the cross entropy loss. We show the effectiveness of our approach on a variety of experimental tasks.

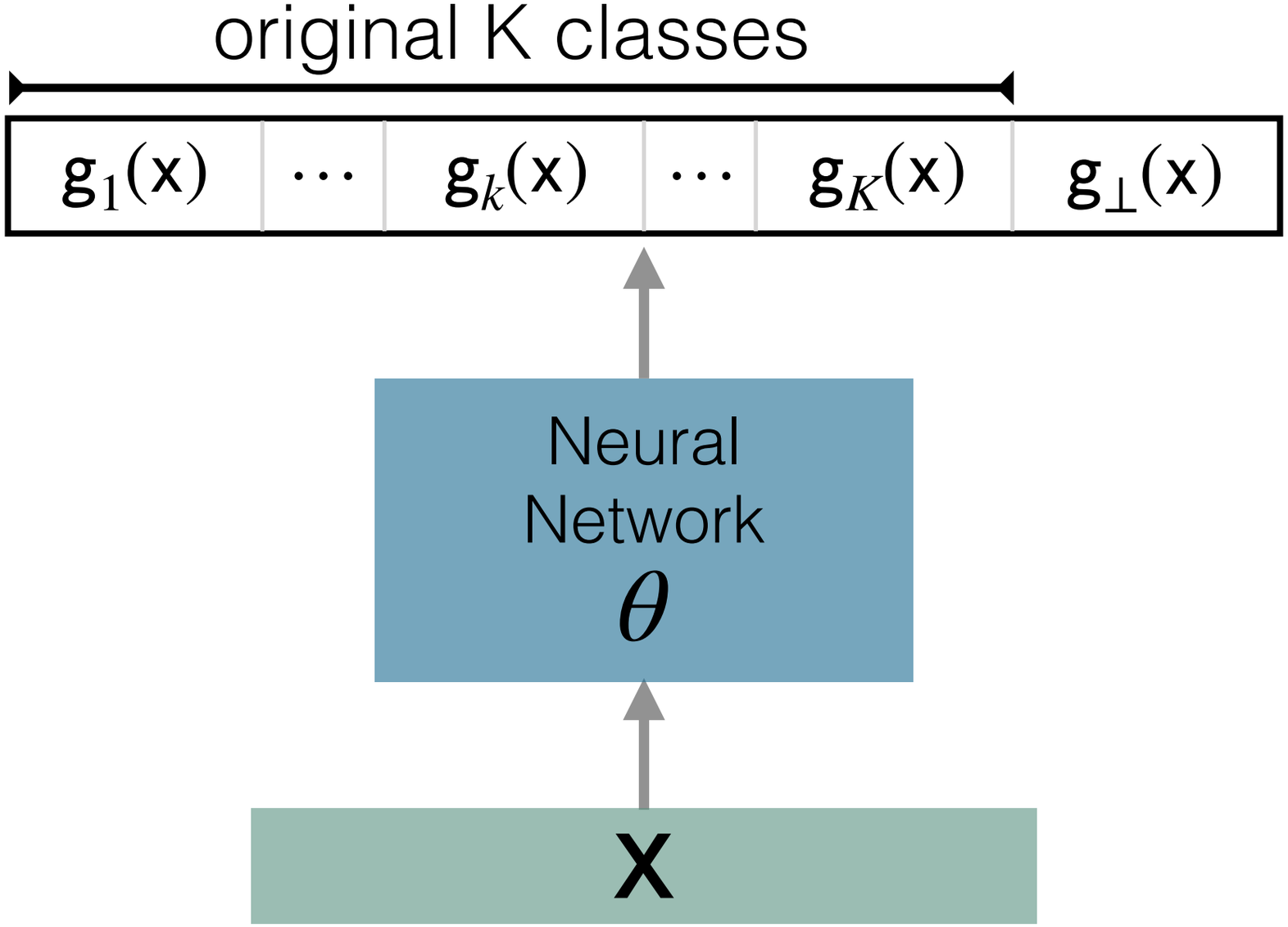
ICML 2020

Solution: Merge the rejector and classifier.

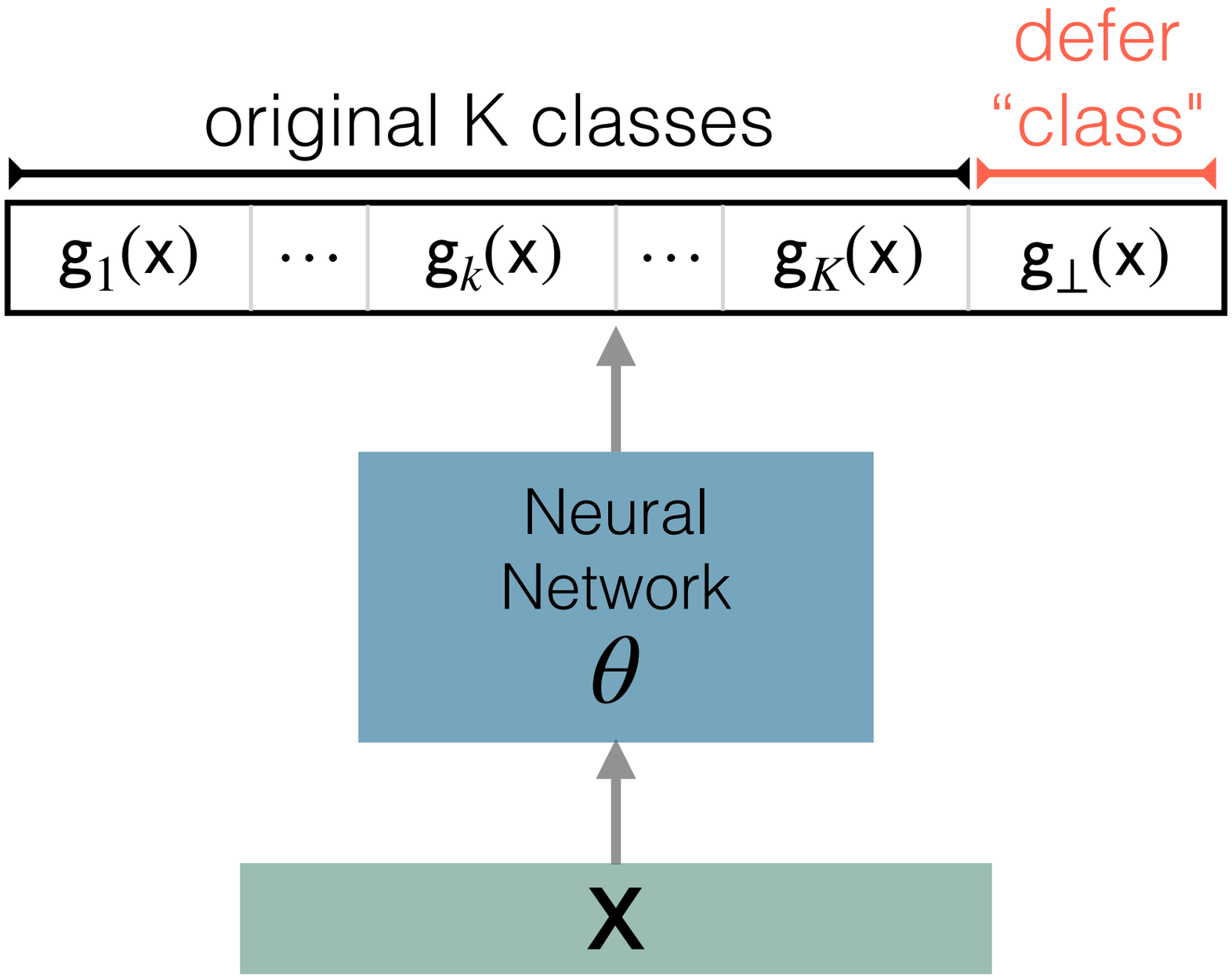
Solution: Merge the rejector and classifier.



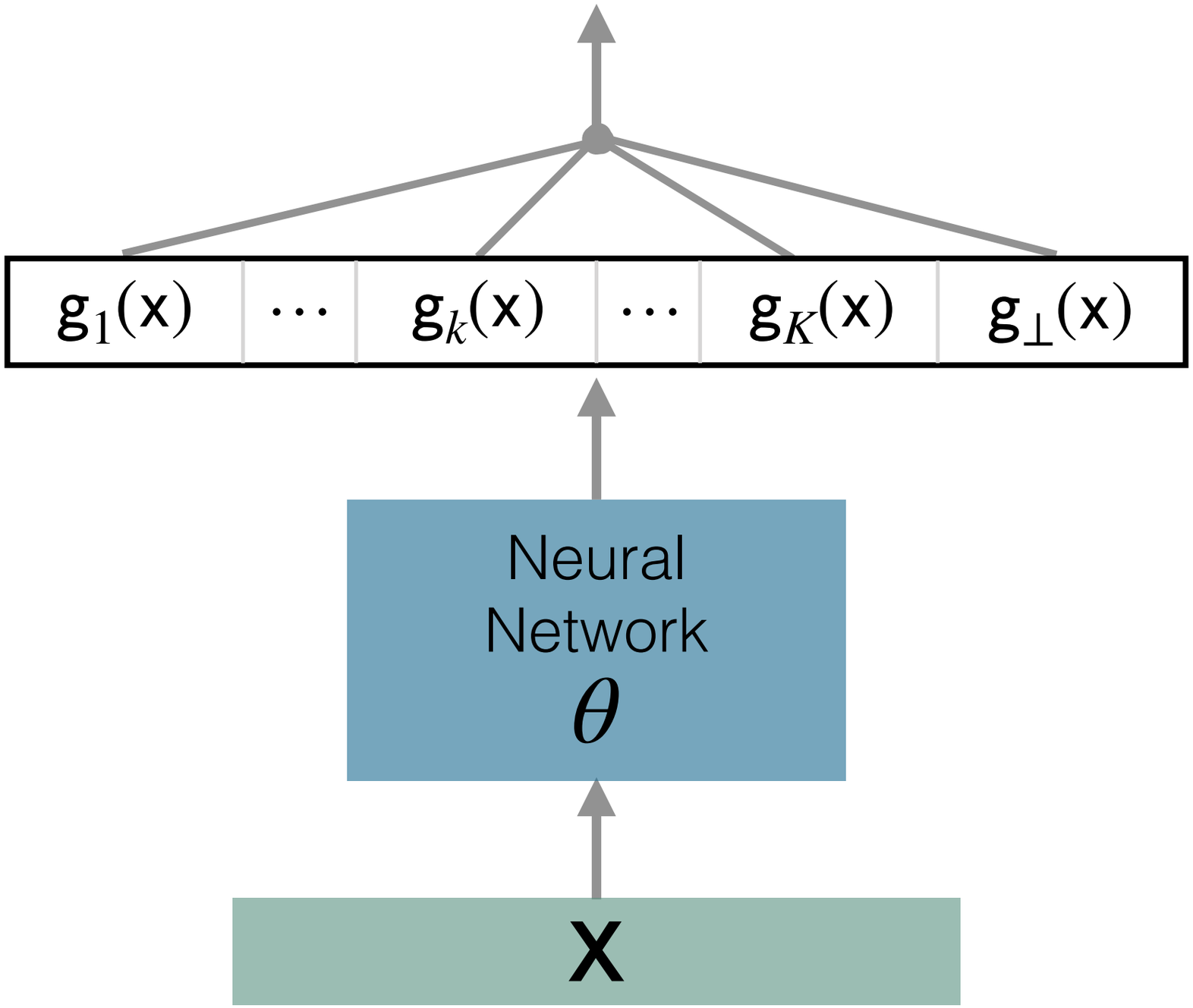
Solution: Merge the rejector and classifier.



Solution: Merge the rejector and classifier.

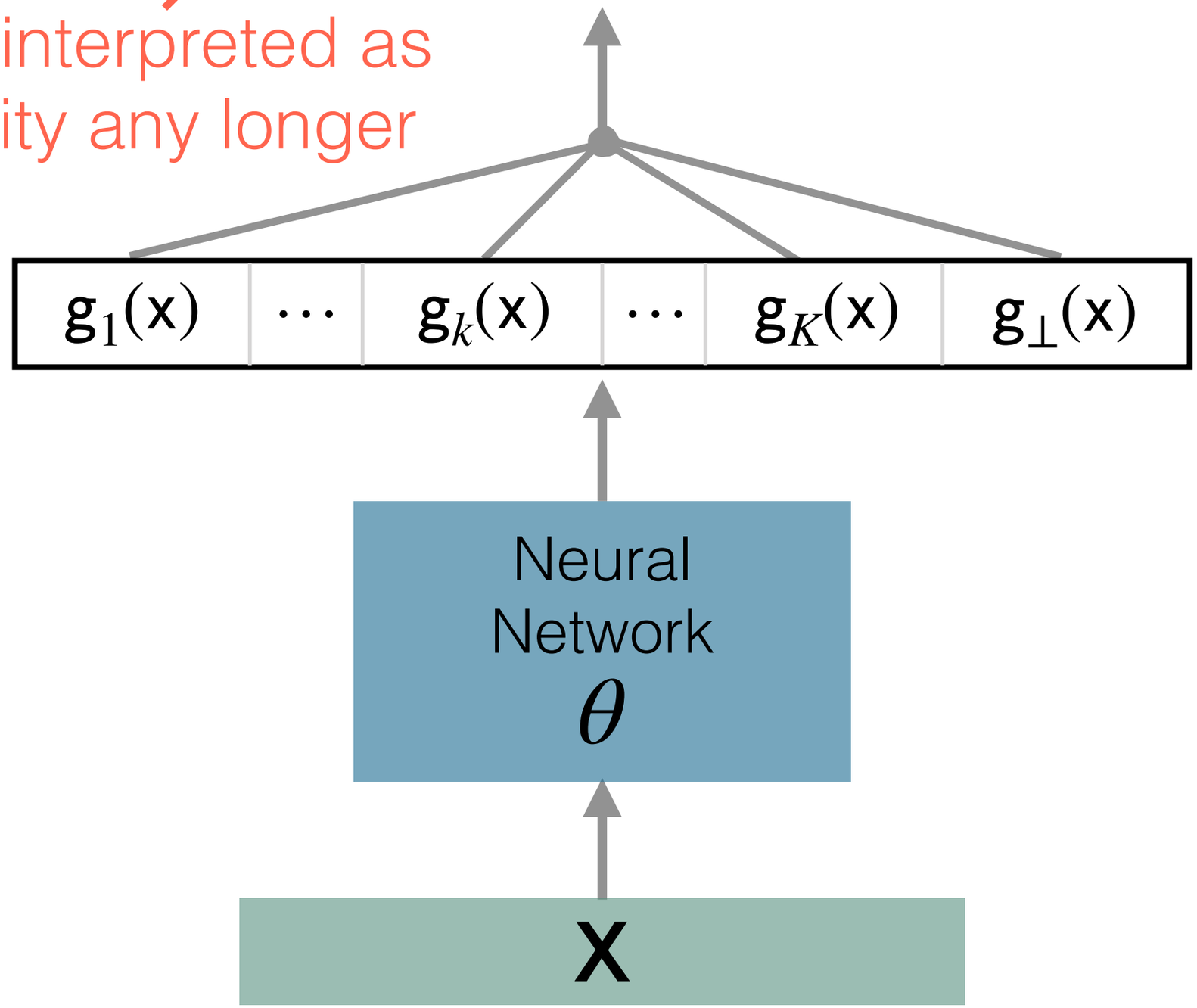


$$p_i(x) = \frac{\exp\{g_i(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$

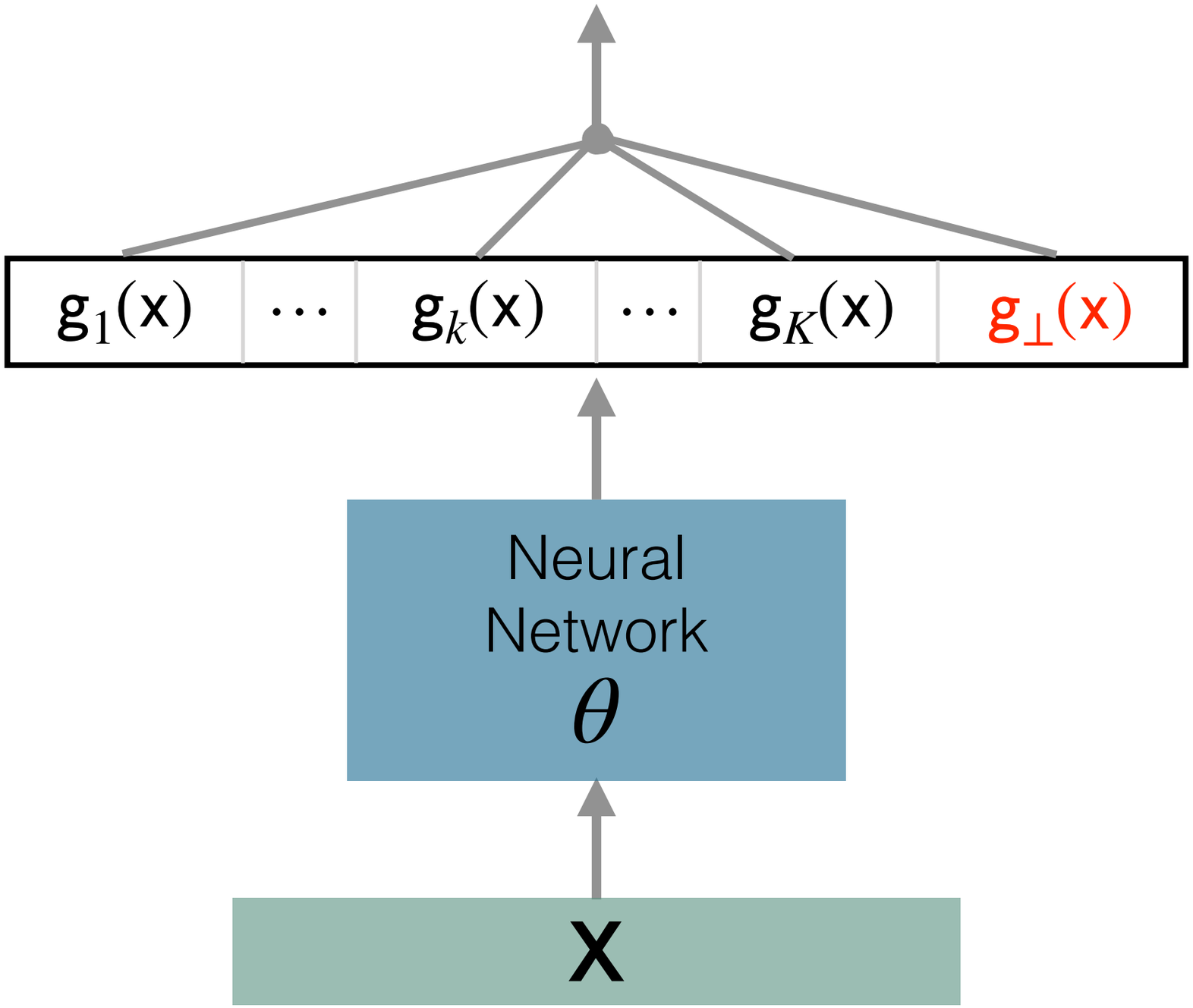


$$p_i(x) = \frac{\exp\{g_i(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$

cannot be interpreted as a probability any longer



$$p_{\perp}(x) = \frac{\exp\{g_{\perp}(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$



$$\ell(\theta; \mathfrak{D}) =$$

$$-\sum_n \left(\log \mathsf{p}_{y_n}(\mathsf{x}_n) \;+\; \mathbb{I}[\mathsf{y}_n = \mathsf{m}_n] \;\log \mathsf{p}_{\perp}(\mathsf{x}_n) \right)$$

classifier loss

rejector loss

$$\ell(\theta; \mathfrak{D}) =$$

$$-\sum_{n} \left(\log \mathsf{p}_{y_n}(\mathsf{x}_n) \; + \; \mathbb{I}[\mathsf{y}_n = \mathsf{m}_n] \; \log \mathsf{p}_{\perp}(\mathsf{x}_n) \right)$$

classifier loss

rejector loss

$$\ell(\theta; \mathfrak{D}) =$$

$$-\sum_n \left(\log \mathsf{p}_{y_n}(\mathsf{x}_n) \;+\; \mathbb{I}[\mathsf{y}_n = \mathsf{m}_n] \;\log \mathsf{p}_{\perp}(\mathsf{x}_n) \right)$$

classifier loss

rejector loss

$$\ell(\theta; \mathfrak{D}) =$$

$$-\sum_n \left(\log \mathsf{p}_{y_n}(\mathsf{x}_n) \;+\; \mathbb{I}[\mathsf{y}_n = \mathsf{m}_n] \;\log \mathsf{p}_\perp(\mathsf{x}_n) \right)$$

classifier loss

rejector loss

only if expert is correct

$$\ell(\theta; \mathfrak{D}) =$$

$$-\sum_n \left(\log \mathsf{p}_{y_n}(\mathsf{x}_n) \;+\; \mathbb{I}[\mathsf{y}_n = \mathsf{m}_n] \;\log \mathsf{p}_\perp(\mathsf{x}_n) \right)$$

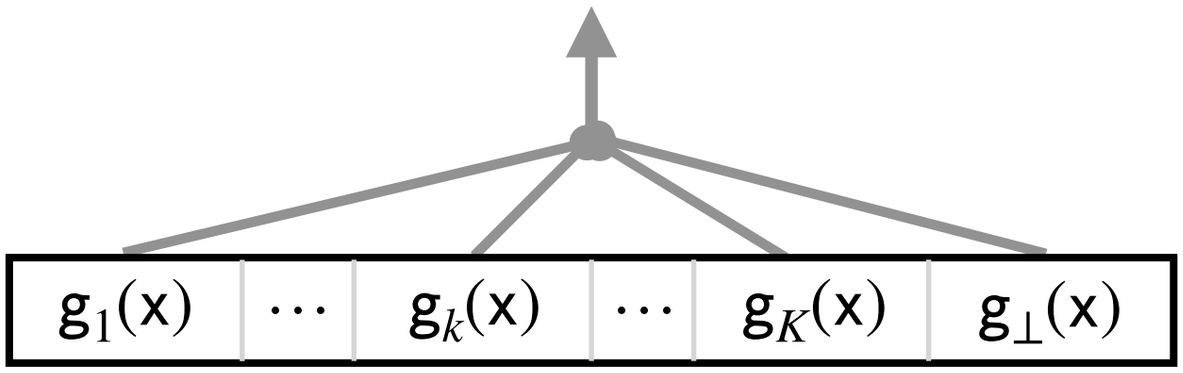
classifier loss

rejector loss

only if expert is correct

Consistency: The minimizers (w.r.t. *g*) correspond to the Bayes optimal classifier and rejector.

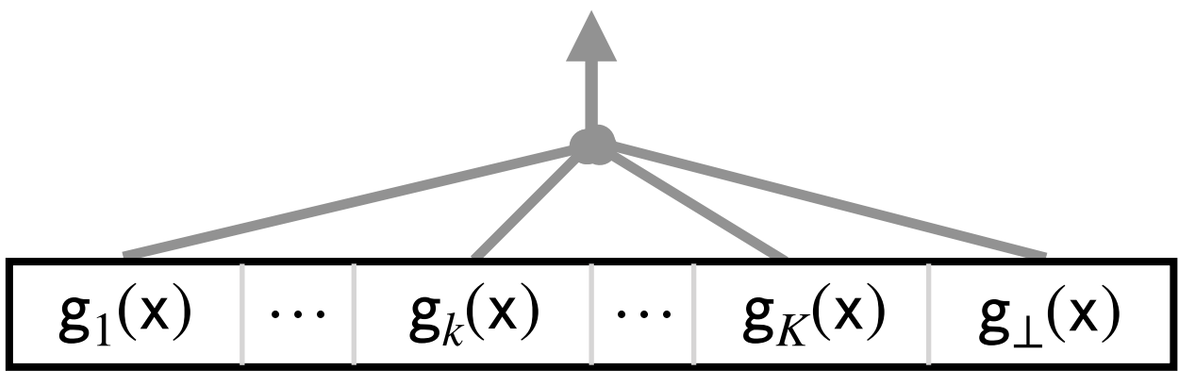
$$p_i(x) = \frac{\exp\{g_i(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$



Classifier

$$\mathbf{h}^*(\mathbf{x}) = \underset{y \in [1,K]}{\operatorname{argmax}} \; \mathbf{g}_y^*(\mathbf{x})$$

$$\mathrm{p}_i(\mathrm{x}) = \frac{\exp\{\mathrm{g}_i(\mathrm{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathrm{g}_k(\mathrm{x})\}}$$



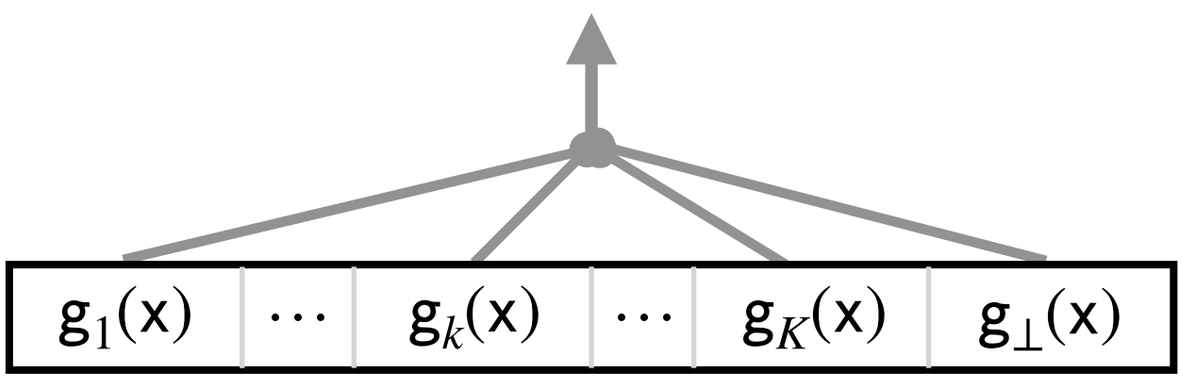
Classifier

$$\mathbf{h}^*(\mathbf{x}) = \underset{y \in [1,K]}{\operatorname{argmax}} \; \mathbf{g}_y^*(\mathbf{x})$$

Rejector

$$\mathbf{r}^*(\mathbf{x}) = \mathbb{I}\left[\mathbf{g}_\perp^*(\mathbf{x}) \geq \max_{y \in [1,K]} \mathbf{g}_y^*(\mathbf{x})\right]$$

$$\mathsf{p}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$



Our work:

Is this learning-to-defer system calibrated?

nature medicine

LETTERS

https://doi.org/10.1038/s41591-020-0942-0

Check for updates

Human–computer collaboration for skin cancer recognition

Philipp Tschandl[1,17], Christoph Rinner[2,17], Zoe Apalla[3], Giuseppe Argenziano[4], Noel Codella[5], Allan Halpern[6], Monika Janda[7], Aimilios Lallas[3], Caterina Longo[8,9], Josep Malvehy[10,11], John Paoli[12,13], Susana Puig[10,11], Cliff Rosendahl[14], H. Peter Soyer[15], Iris Zalaudek[16] and Harald Kittler[1]

The rapid increase in telemedicine coupled with recent advances in diagnostic artificial intelligence (AI) create the imperative to consider the opportunities and risks of inserting AI-based support into new paradigms of care. Here we

competitive view of AI is evolving based on studies suggesting that a more promising approach is human–AI cooperation[10–15]. The role of human–computer collaboration in health-care delivery, the appropriate settings in which it can be applied and its impact on the

"The least experienced [physicians] tended to accept AI-based support that contradicted their initial diagnosis even if they were confident."

Calibration: Is the system a good forecaster?

Calibration: Is the system a good forecaster?

$$\mathbb{P}(\mathbf{y} \mid \mathbf{x})$$

Does the classifier correctly estimate the underlying class probabilities?

Calibration: Is the system a good forecaster?

$$\mathbb{P}(y \mid x)$$

Does the classifier correctly estimate the underlying class probabilities?

$$\mathbb{P}(m = y \mid x)$$

Does the rejector correctly estimate the expert's chance of being correct?

Calibration: Is the system a good forecaster?

$$\mathbb{P}(y \mid x)$$

Does the <u>classifier</u> correctly estimate the underlying class probabilities?

$$\mathbb{P}(m = y \mid x)$$

Does the <u>rejector</u> correctly estimate the expert's chance of being correct?

Calibration: Is the system a good forecaster?

$$\mathbb{P}(y \mid x)$$

Does the classifier correctly estimate the underlying class probabilities?

$$\mathbb{P}(m = y \mid x)$$

Does the rejector correctly estimate the expert's chance of being correct?

Calibration: Is the system a good forecaster?

$$\mathbb{P}(y \mid x)$$

Does the classifier correctly estimate the underlying class probabilities?

$$\mathbb{P}(m = y \mid x)$$

Does the rejector correctly estimate the expert's chance of being correct?

Recall…

$$p_{\perp}(x) = \frac{\exp\{g_{\perp}(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$

Recall…

$$p_{\perp}(x) = \frac{\exp\{g_{\perp}(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$

From Mozannar's & Sontag's Theorem #1…

$$\frac{\mathbb{P}(m = y \mid x)}{1 + \mathbb{P}(m = y \mid x)} = p^*_{\perp}(x)$$

Recall…

$$p_{\perp}(x) = \frac{\exp\{g_{\perp}(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$

From Mozannar's & Sontag's Theorem #1…

$$\frac{\mathbb{P}(m = y \mid x)}{1 + \mathbb{P}(m = y \mid x)} = p^*_{\perp}(x)$$

Rearranging…

$$\mathbb{P}(m = y \mid x) \;=\; \frac{p^*_{\perp}(x)}{1 - p^*_{\perp}(x)}$$

Recall…

$$p_{\perp}(x) = \frac{\exp\{g_{\perp}(x)\}}{\sum_{k=1}^{K+1} \exp\{g_k(x)\}}$$

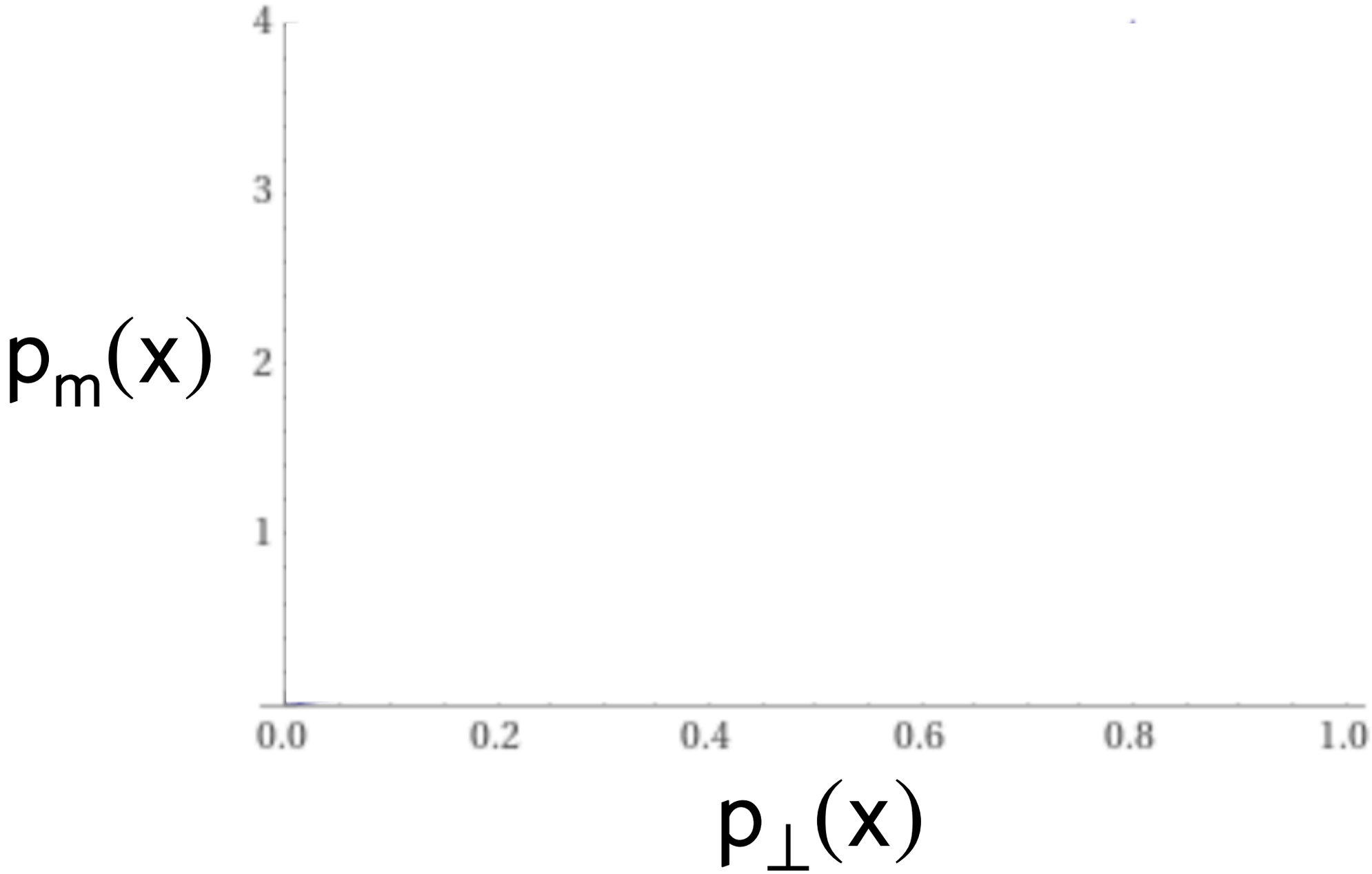
From Mozannar's & Sontag's Theorem #1…

$$\frac{\mathbb{P}(m = y \mid x)}{1 + \mathbb{P}(m = y \mid x)} = p_{\perp}^{*}(x)$$

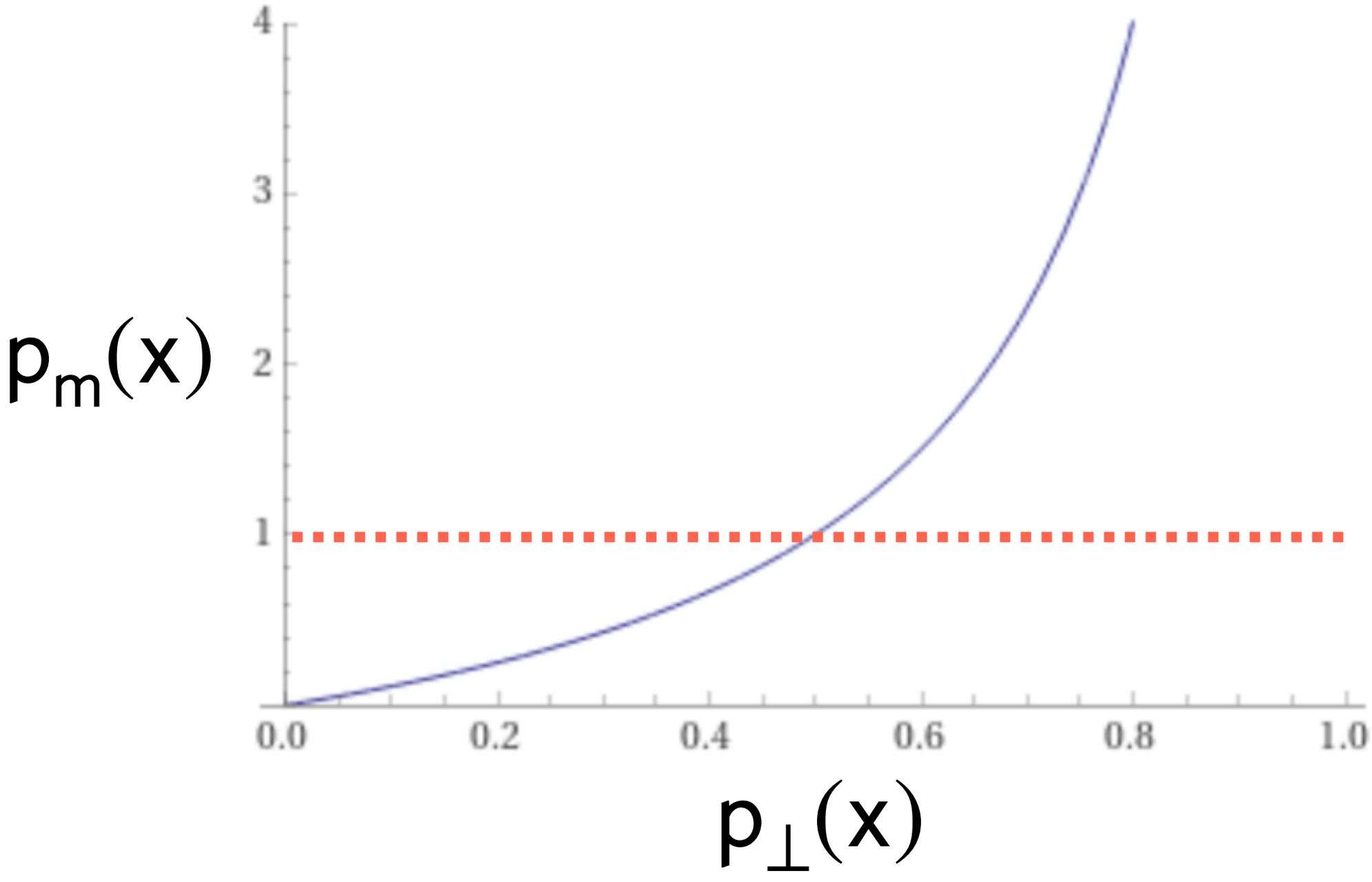
Rearranging…

$$\mathbb{P}(m = y \mid x) \;=\; \frac{p_{\perp}^{*}(x)}{1 - p_{\perp}^{*}(x)} \;\triangleq\; p_m(x)$$

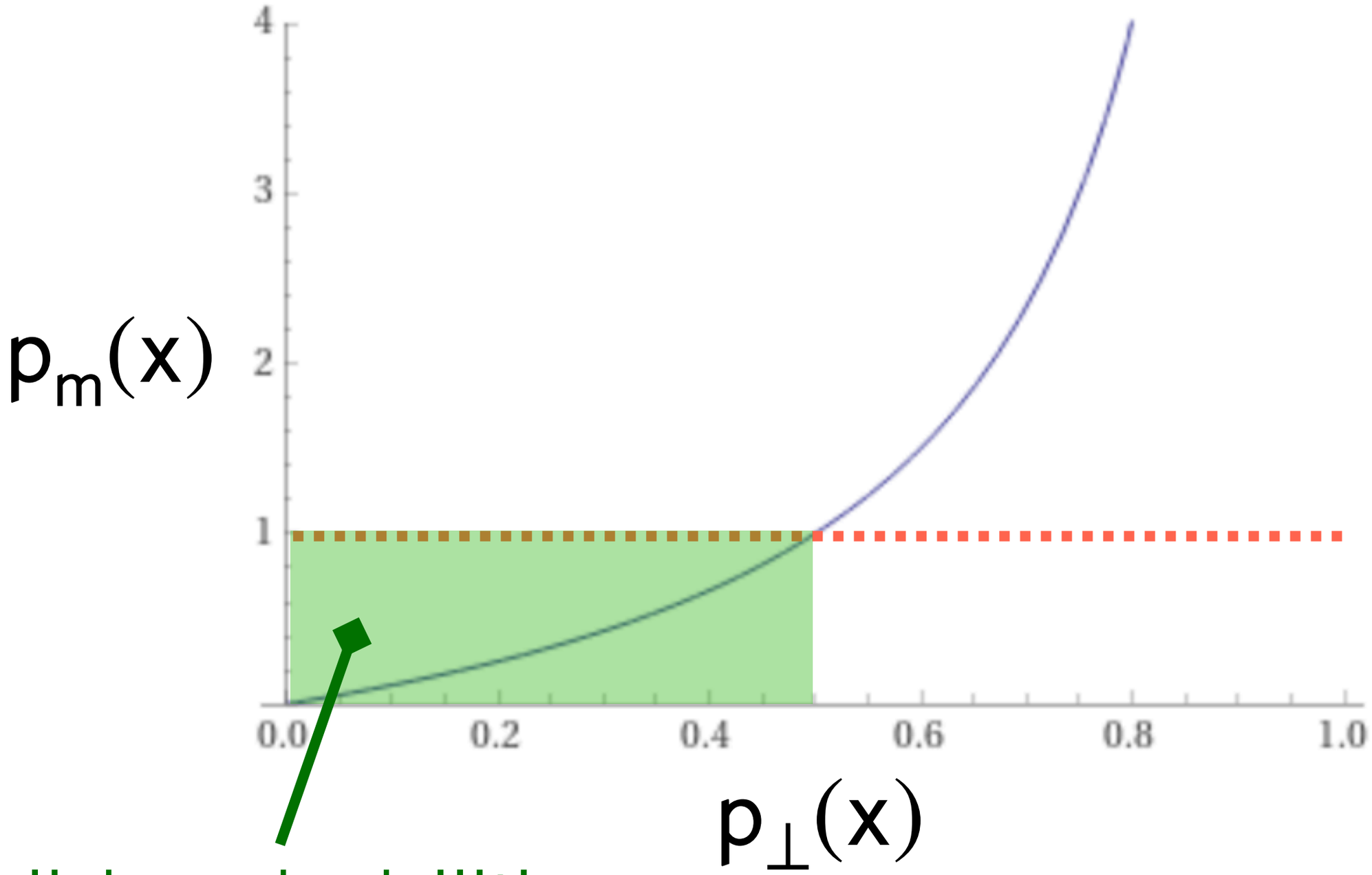
Estimating $\mathbb{P}(m = y \mid x)$



Estimating $\mathbb{P}(m = y \mid x)$

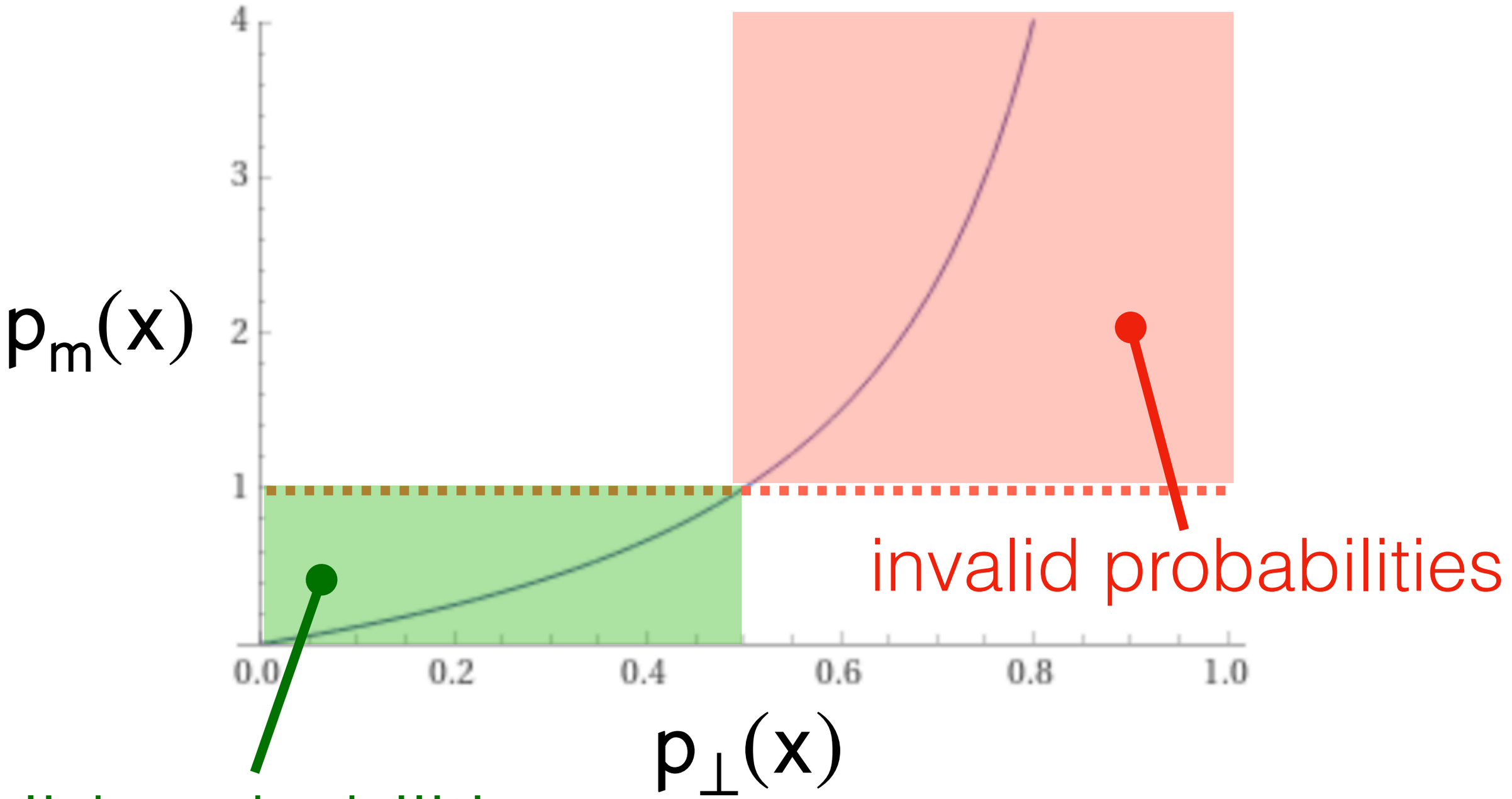


Estimating $\mathbb{P}(m = y \mid x)$



valid probabilities

Estimating $\mathbb{P}(m = y \mid x)$

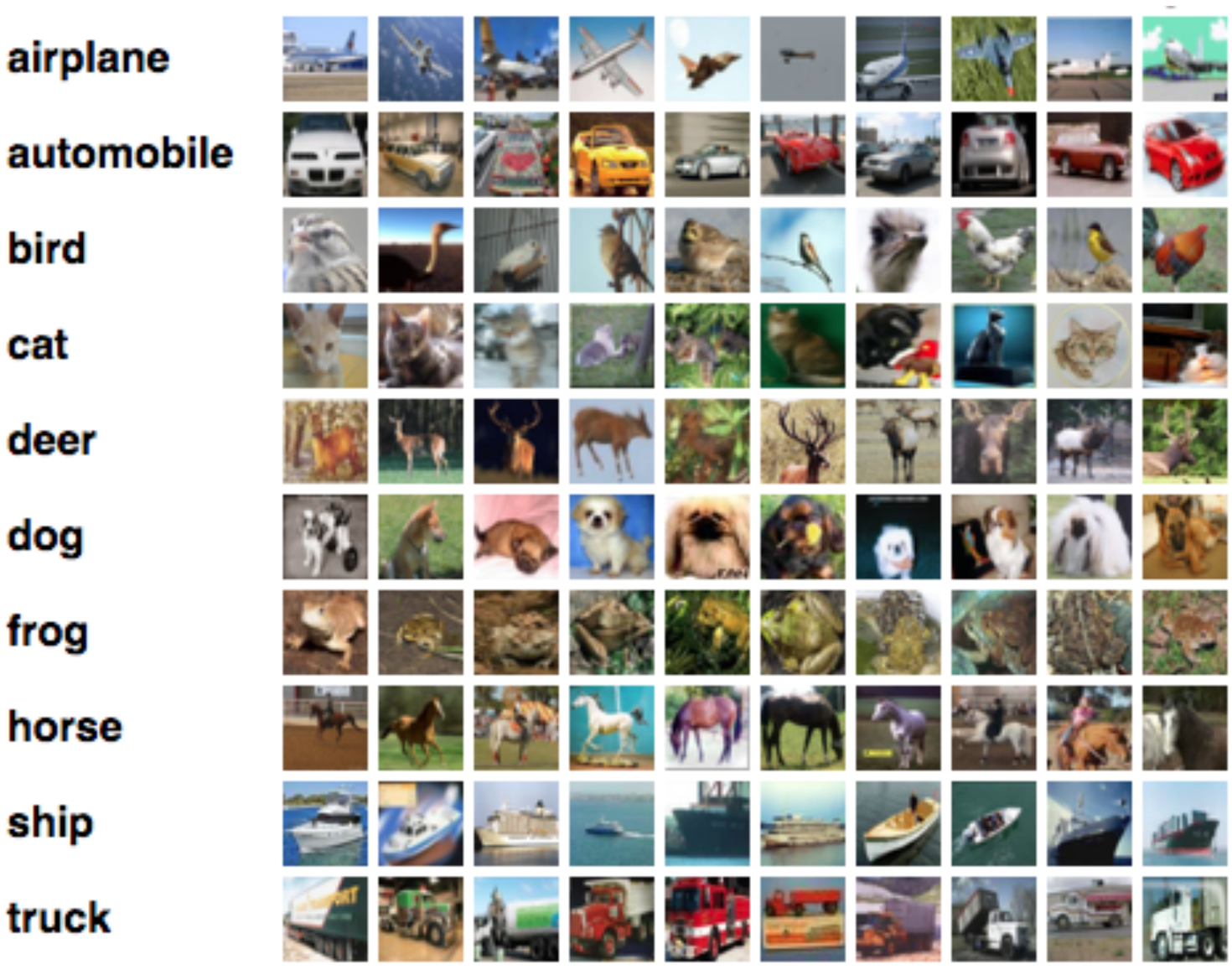


What happens in practice?

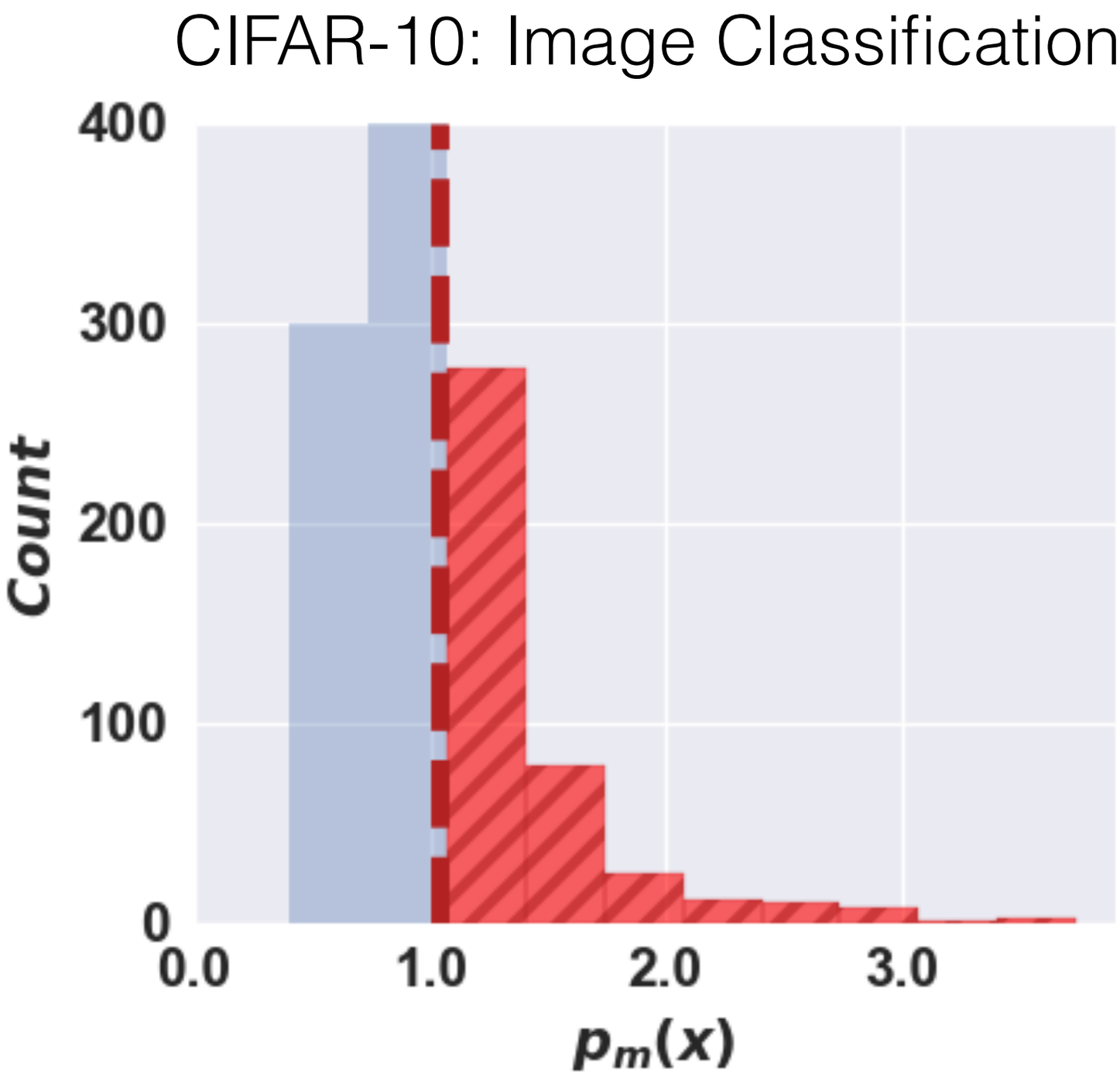
CIFAR-10: Image Classification

What happens in practice?

CIFAR-10: Image Classification



What happens in practice?

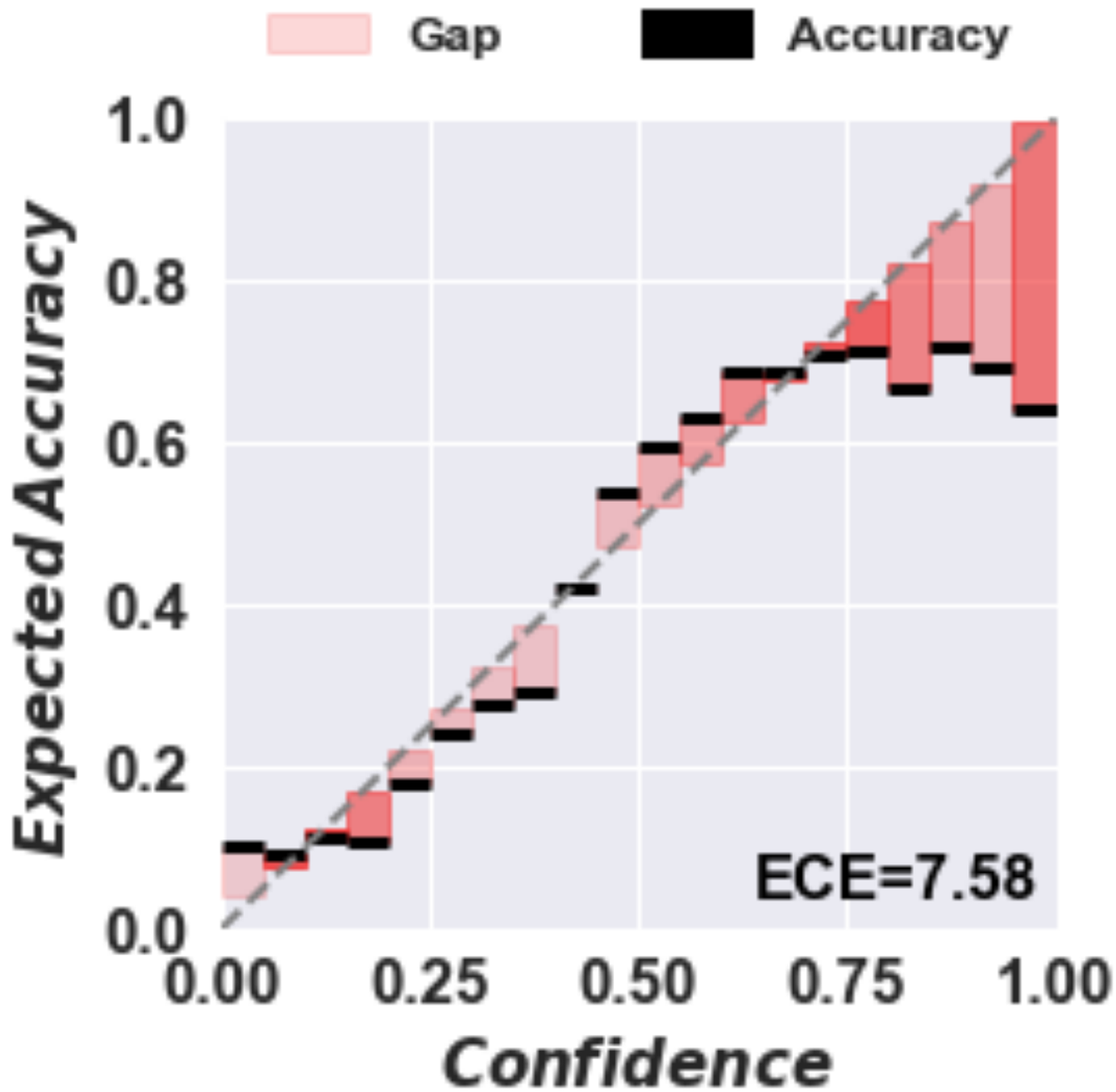


Truncation: $p_m(x) \in (0,1]$

CIFAR-10: Image Classification

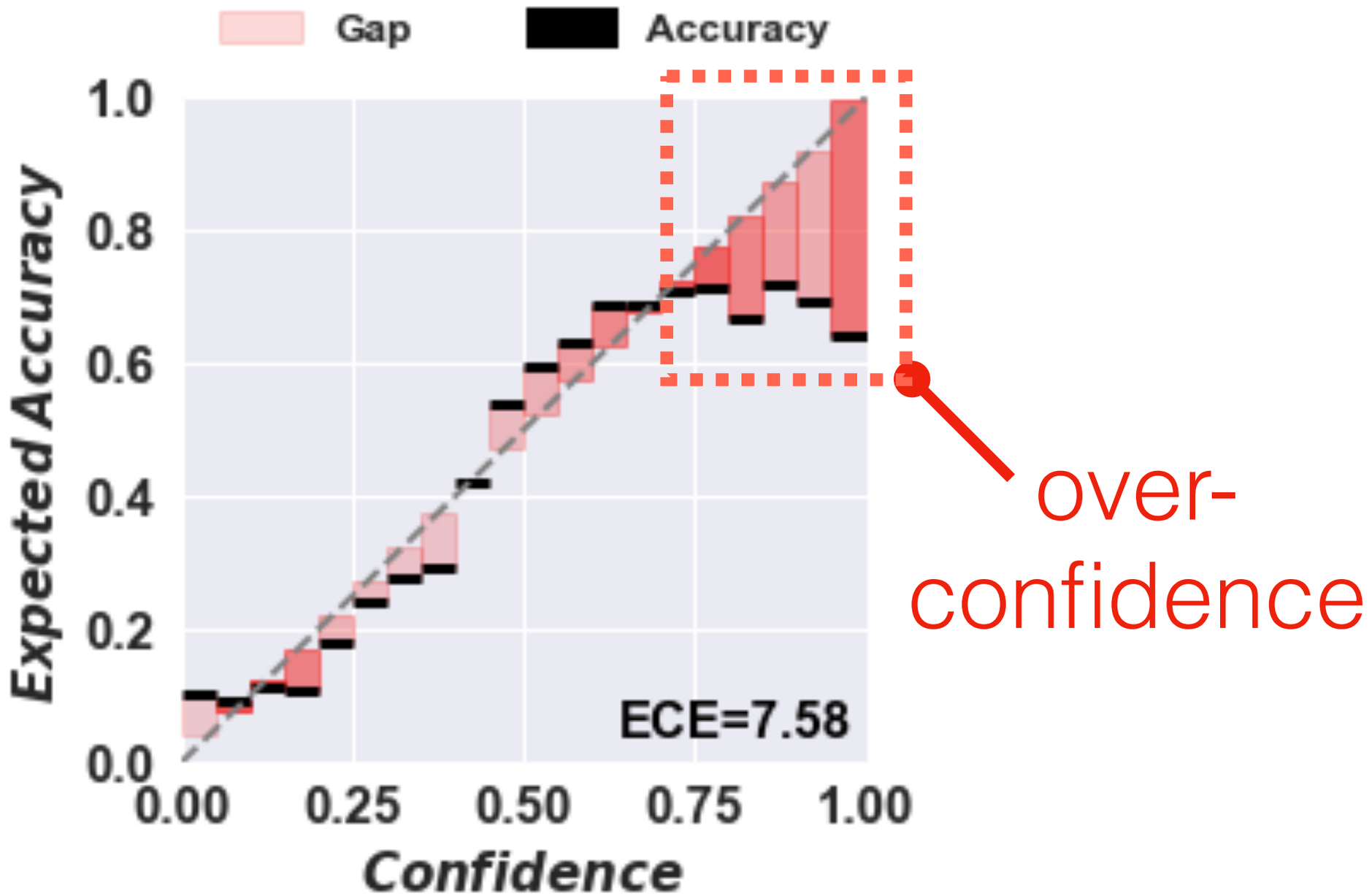
Truncation: $p_m(x) \in (0,1]$

CIFAR-10: Image Classification



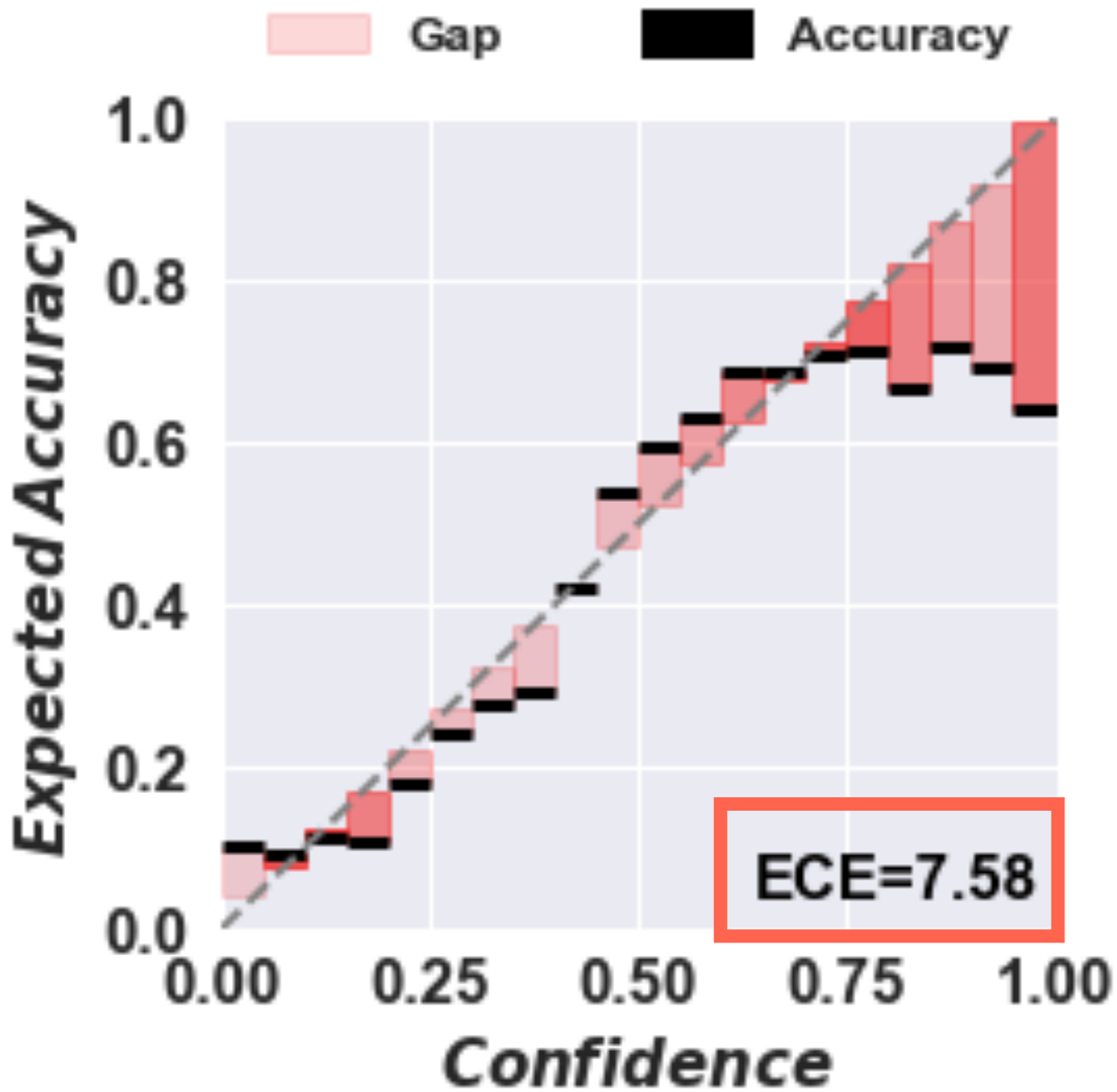
Truncation: $p_m(x) \in (0,1]$

CIFAR-10: Image Classification



Truncation: $p_m(x) \in (0,1]$

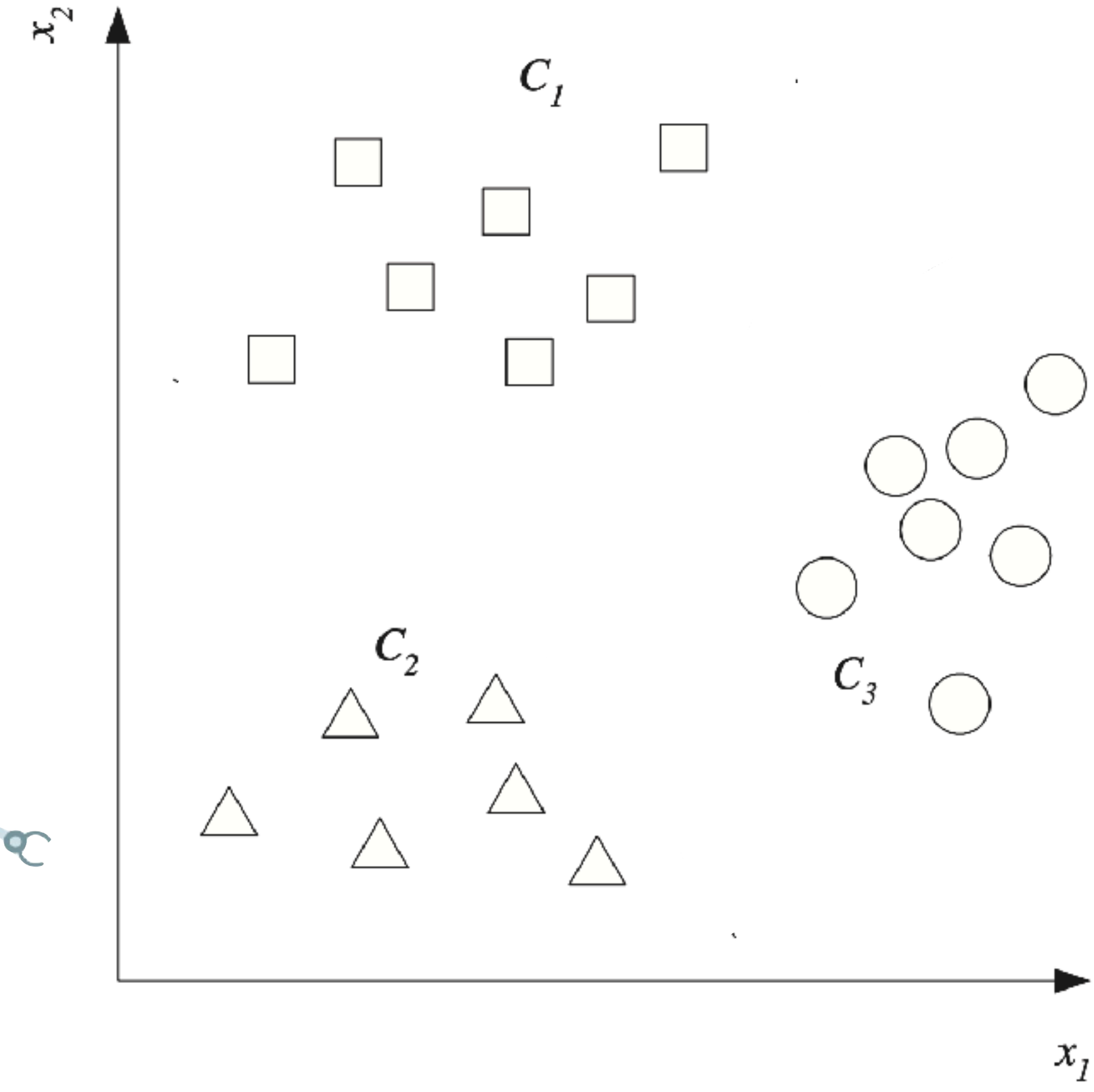
CIFAR-10: Image Classification

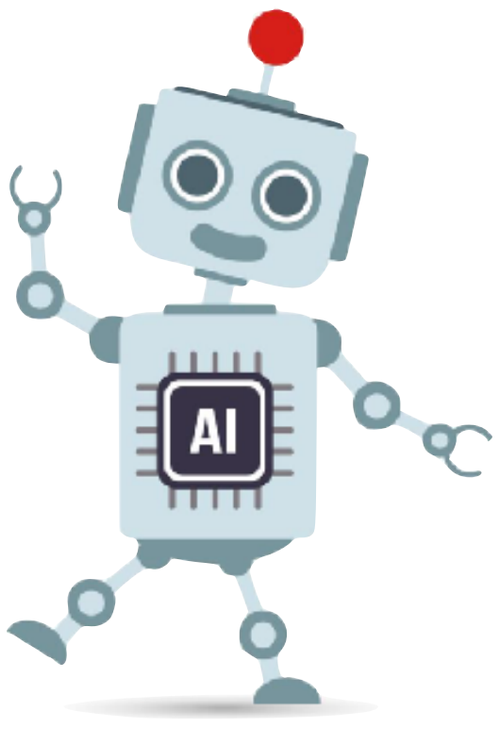


ECE = expected calibration error

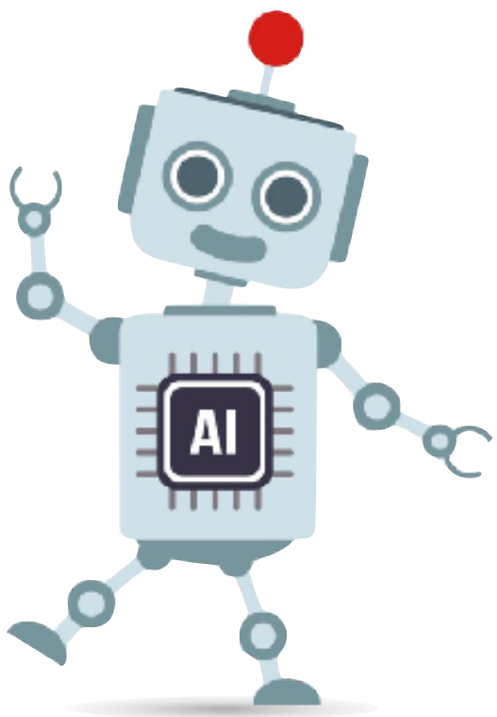
Solution: Learning to Defer with a One-vs-All Classifier

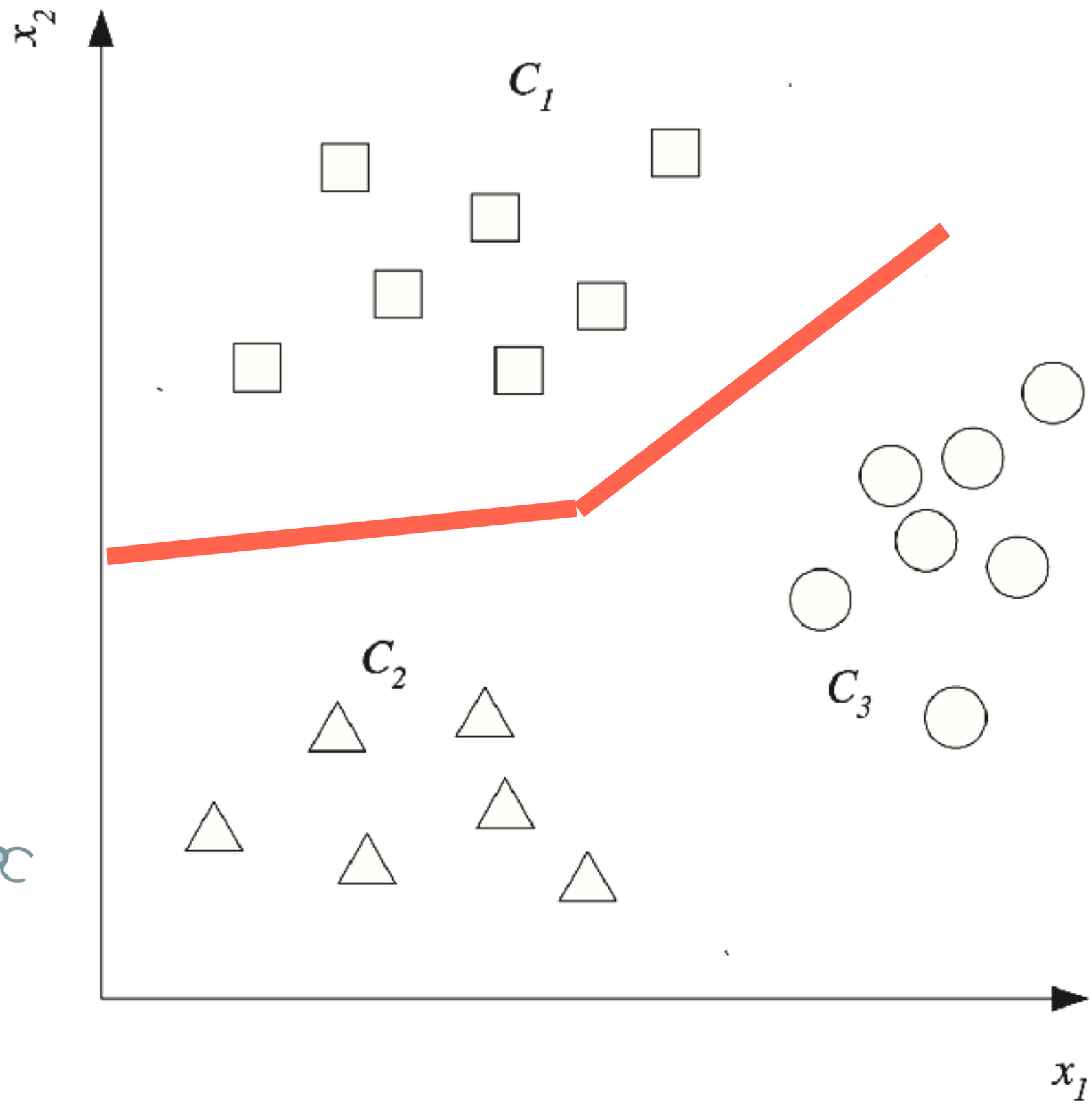
Solution: Learning to Defer with a One-vs-All Classifier



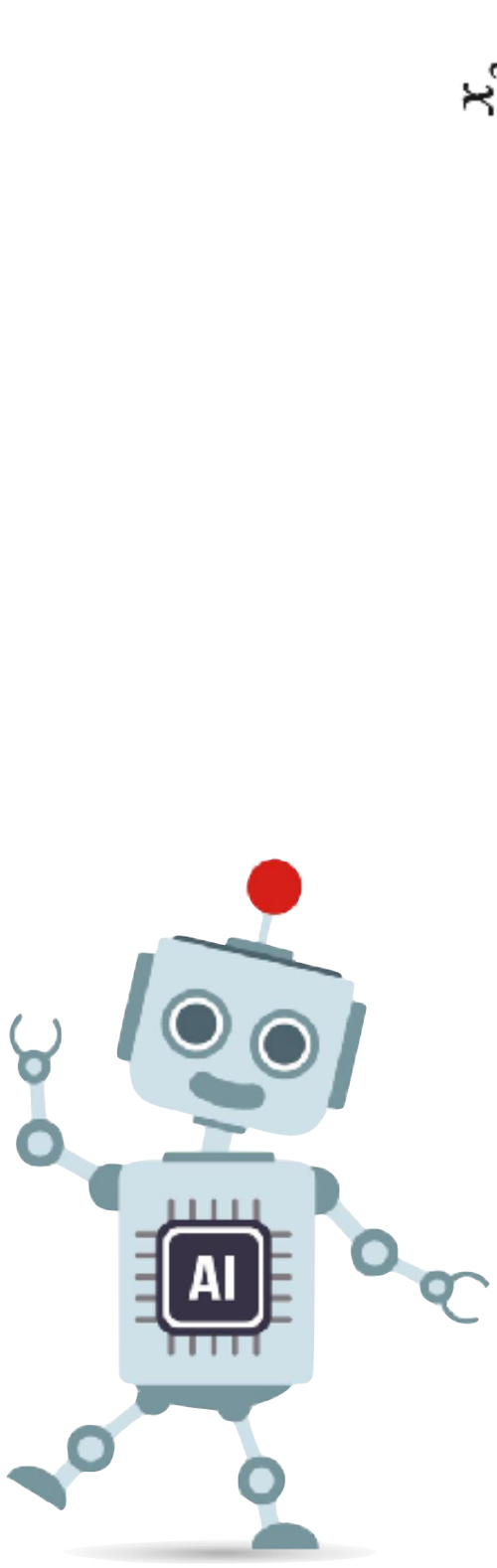


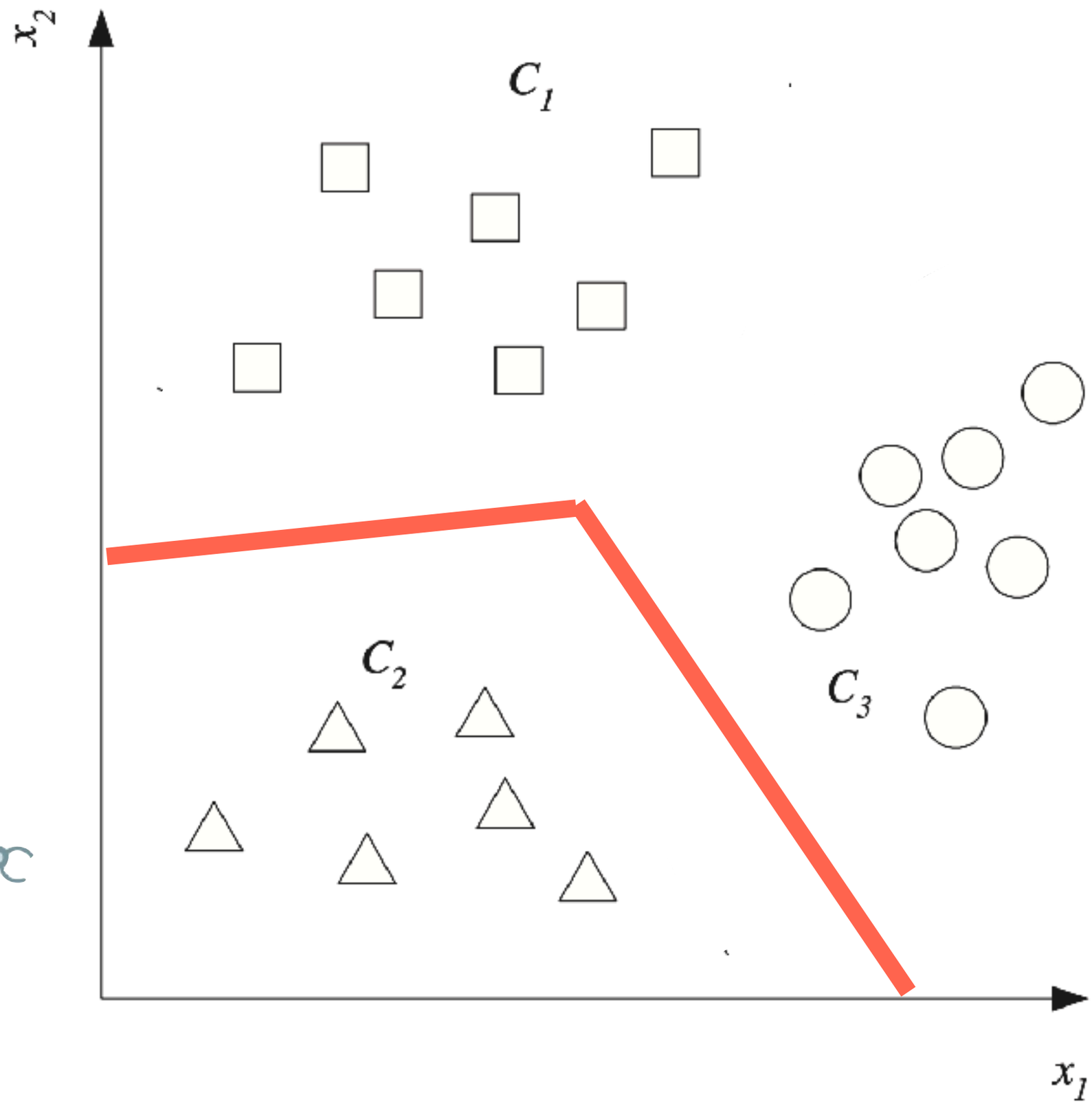
One-vs-All Classifier



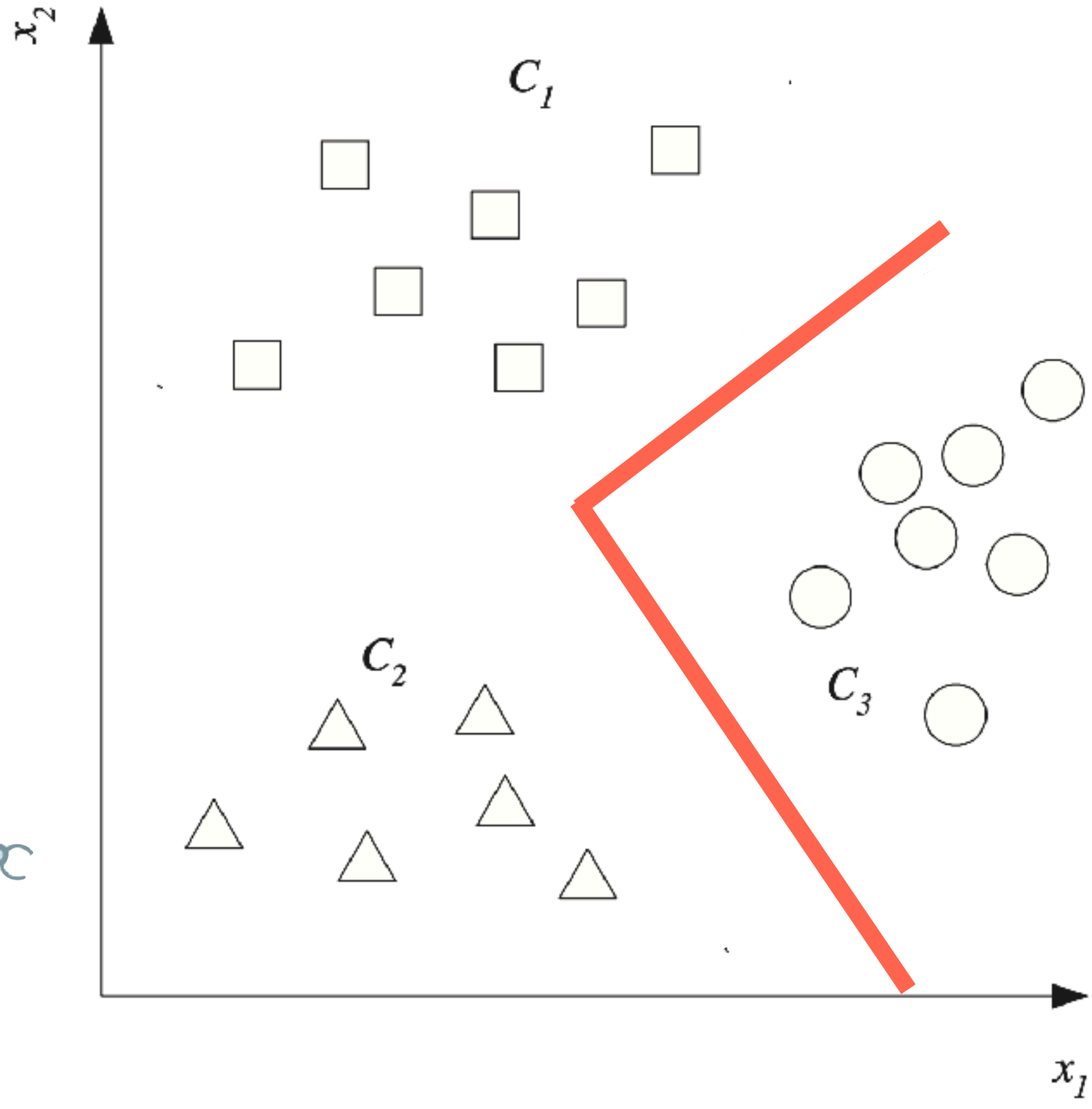


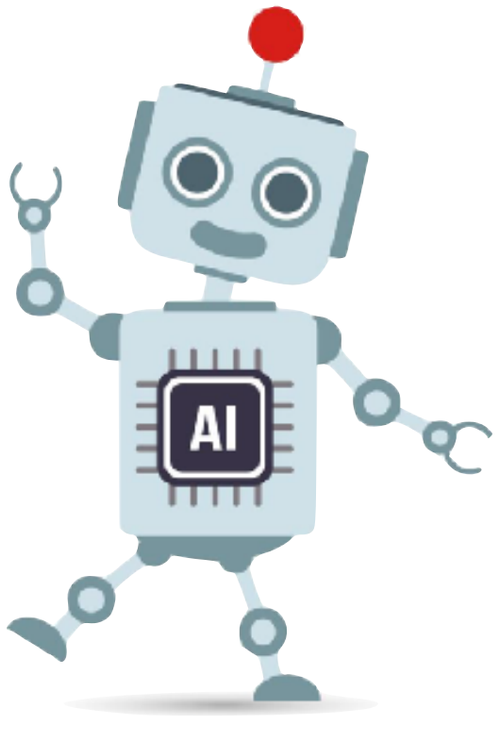
One-vs-All Classifier





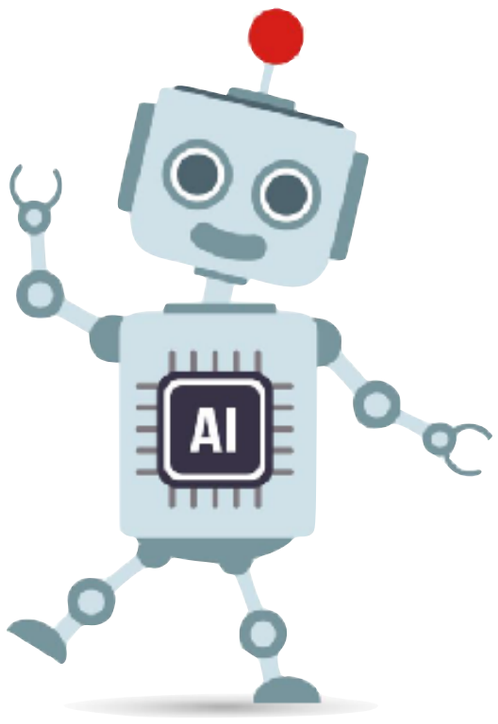
One-vs-All Classifier



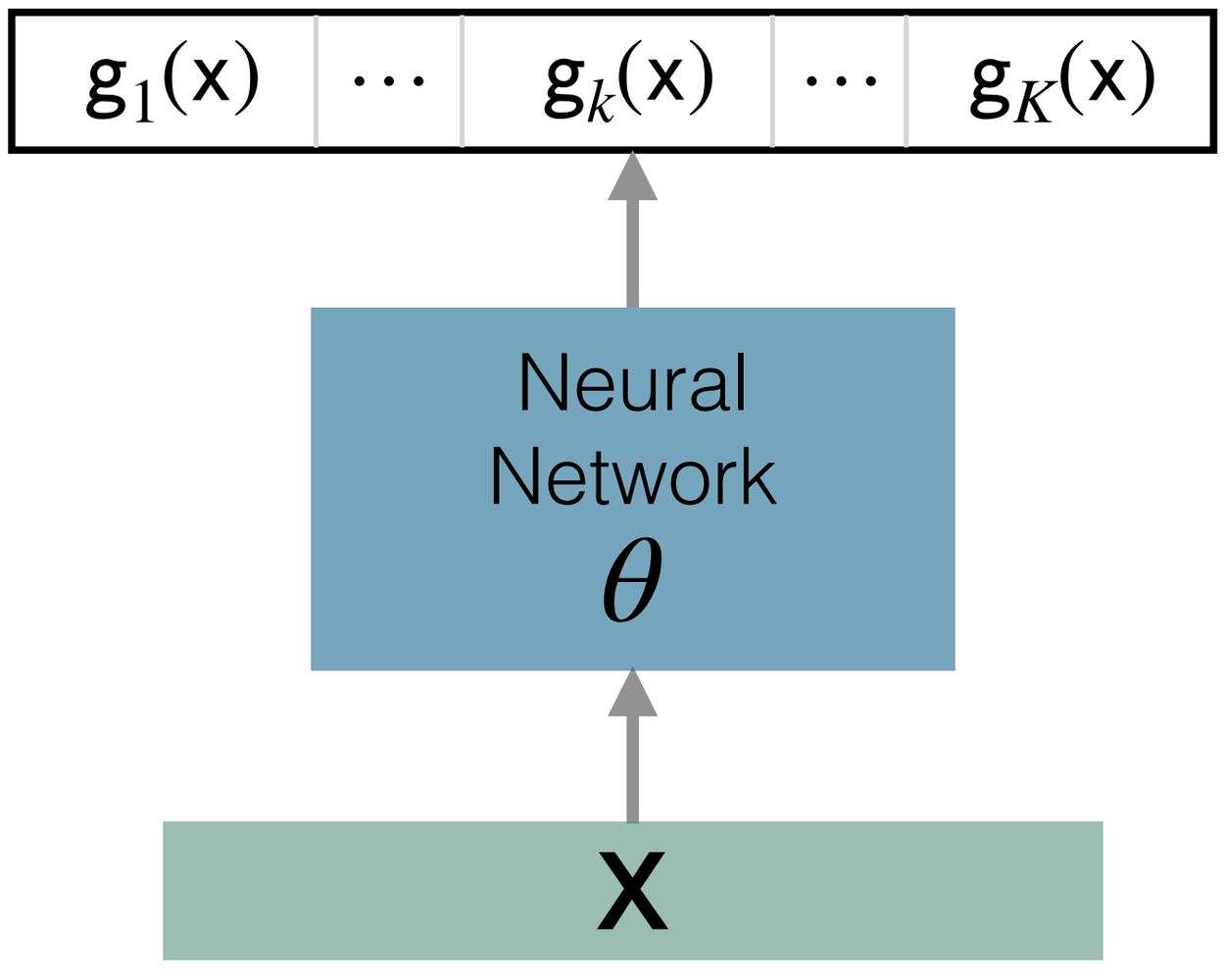


One-vs-All Classifier

Goal: For input features **x**, predict membership in one of K classes: $C_1, \ldots, C_K$

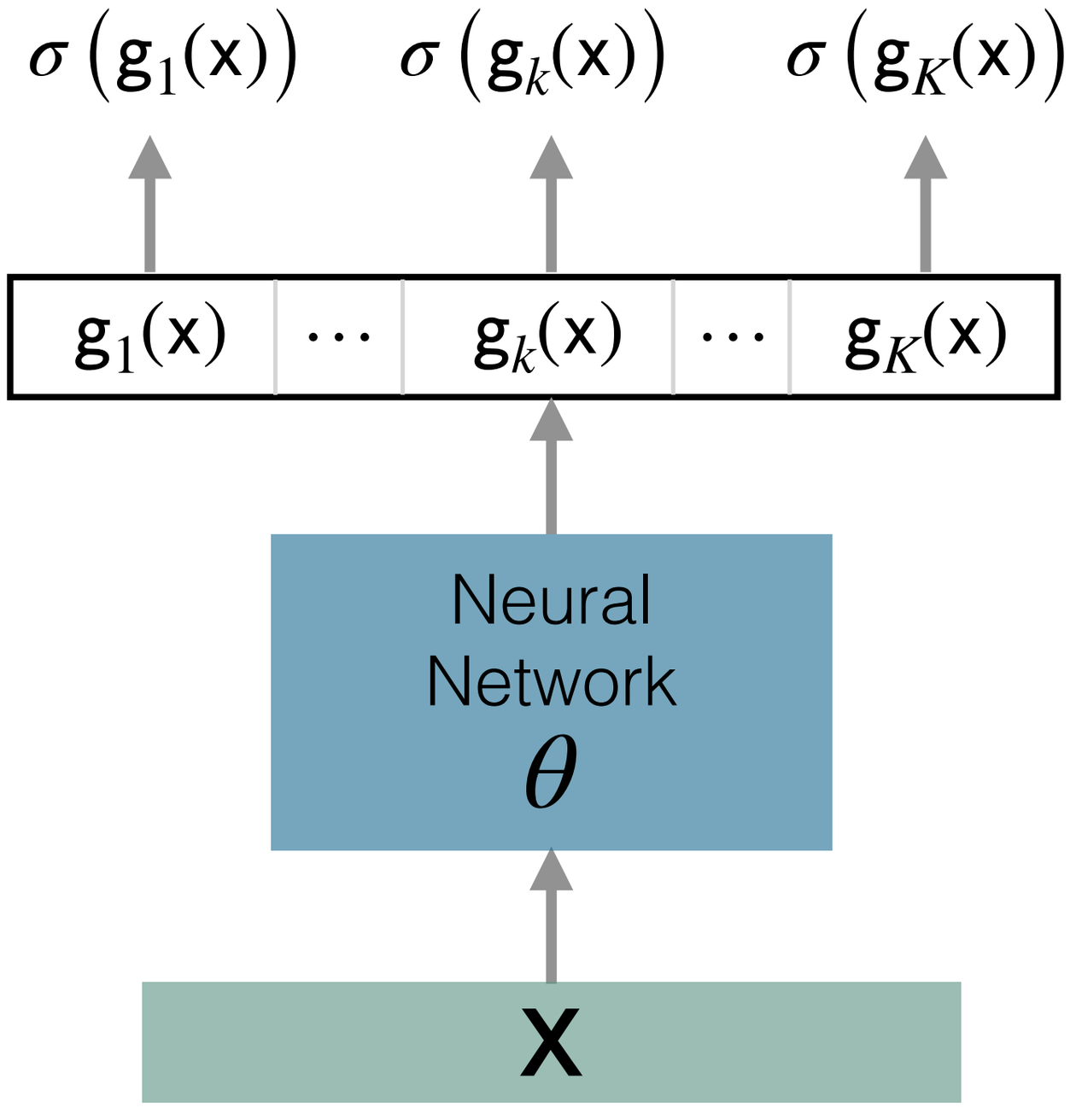


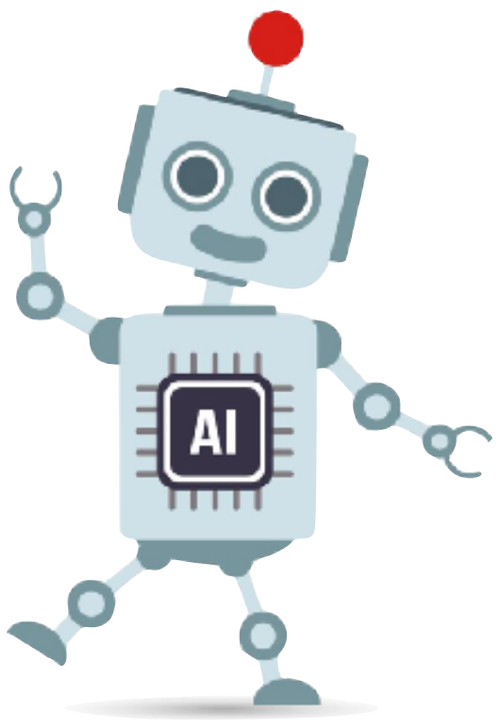
One-vs-All Classifier



Goal: For input features **x**, predict membership in one of *K* classes: $C_1, \ldots, C_K$

$$\sigma(z) = \frac{1}{1 + \exp\{-z\}}$$

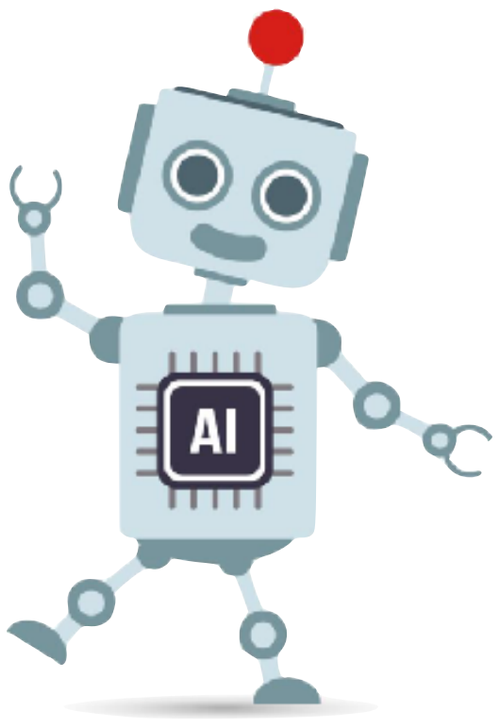


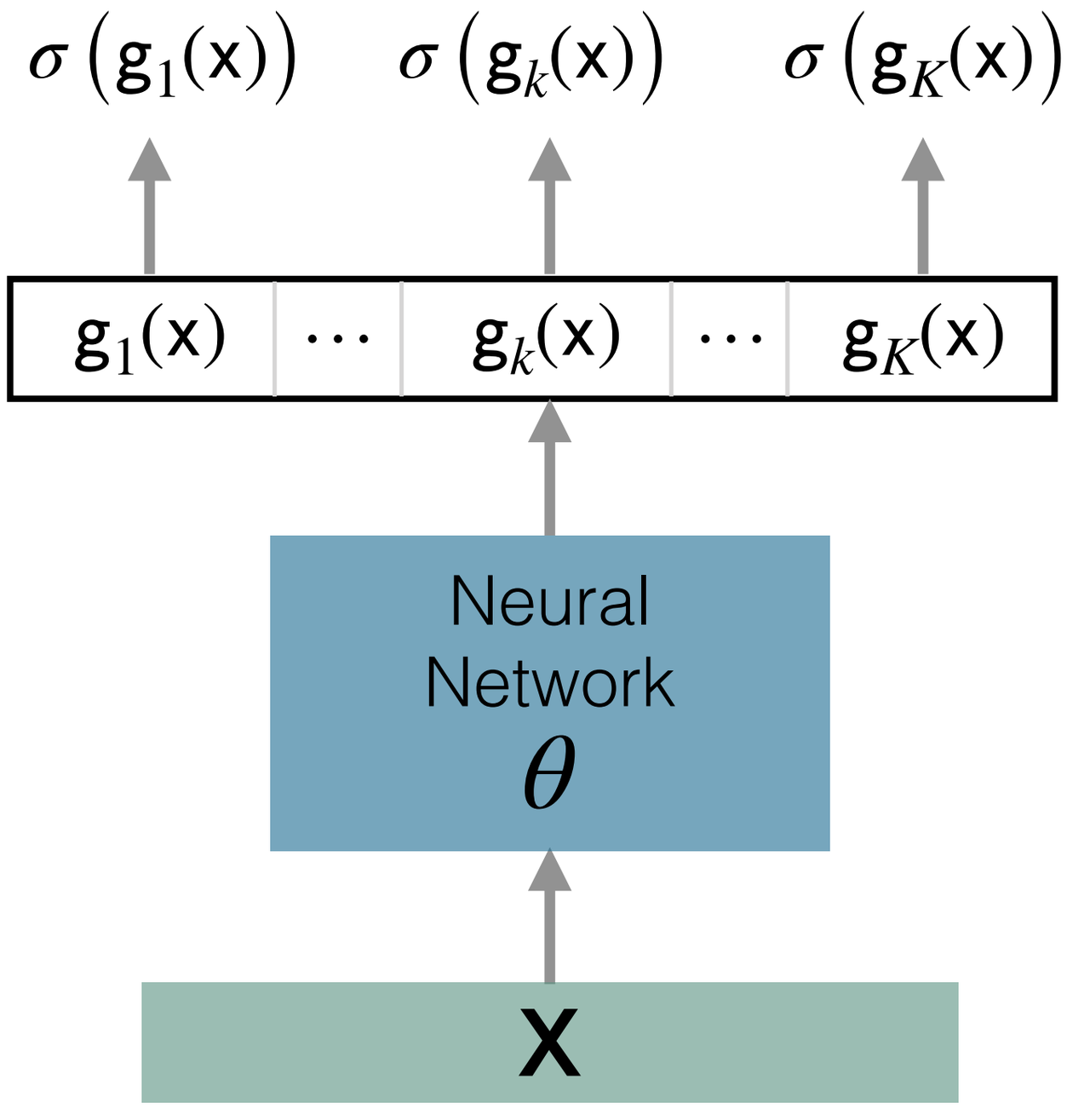


One-vs-All Classifier

Goal: For input features **x**, predict membership in one of *K* classes: $C_1, \ldots, C_K$

$$P(C_i \mid \mathbf{x}) = \sigma\left(g_i(\mathbf{x})\right)$$

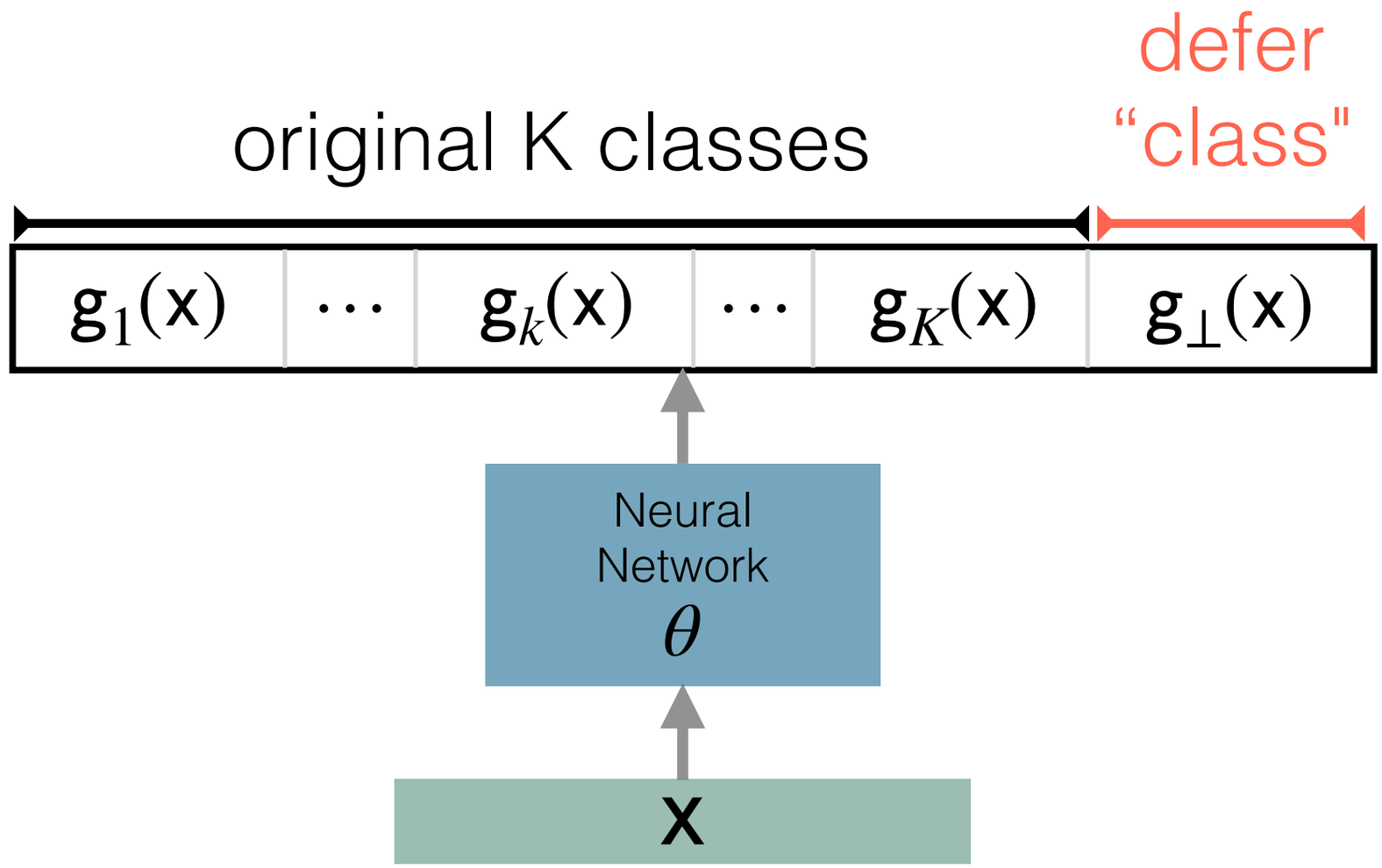




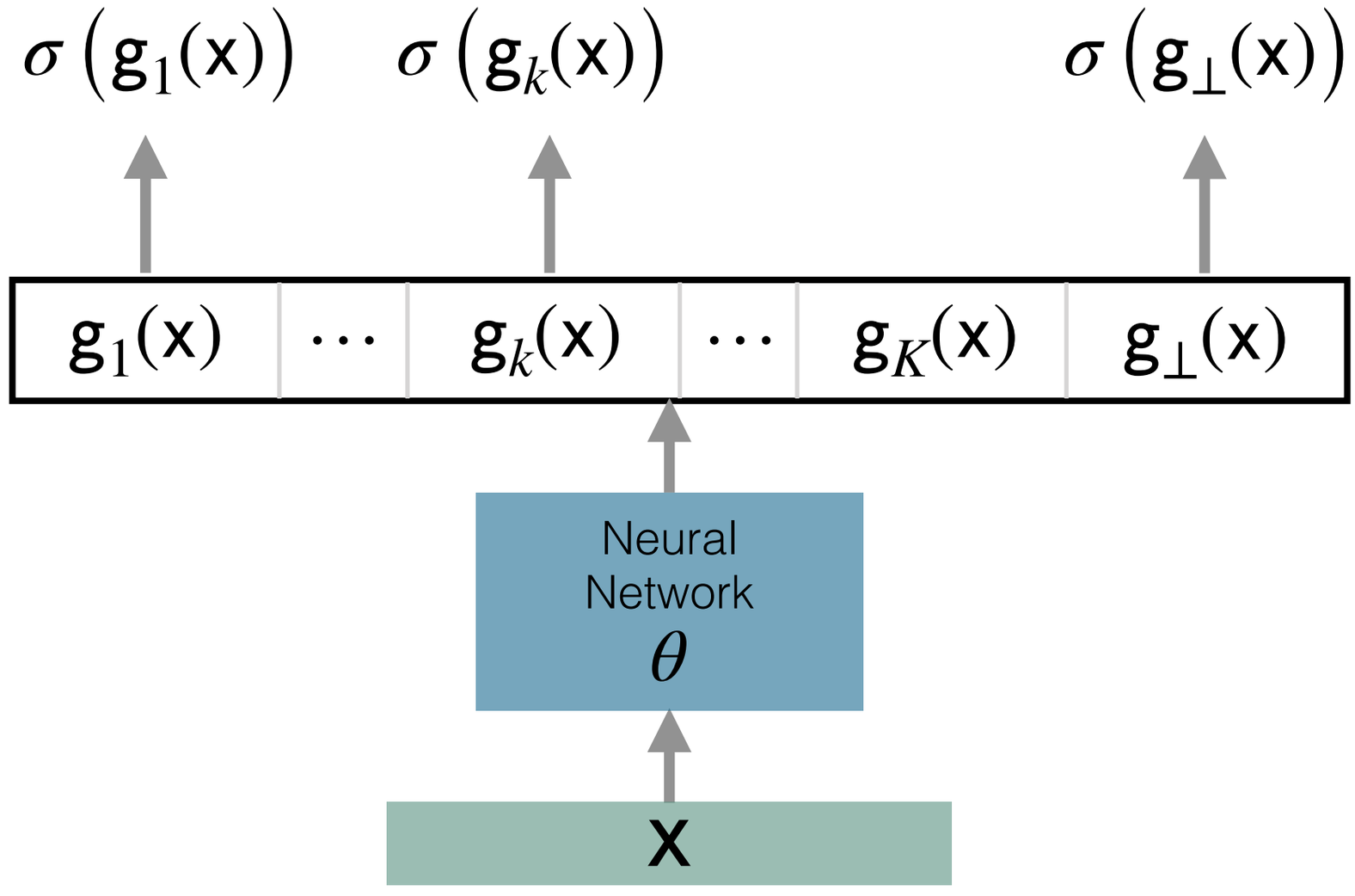
One-vs-All Classifier

Learning-to-Defer: One-vs-All Parameterization

Learning-to-Defer: One-vs-All Parameterization



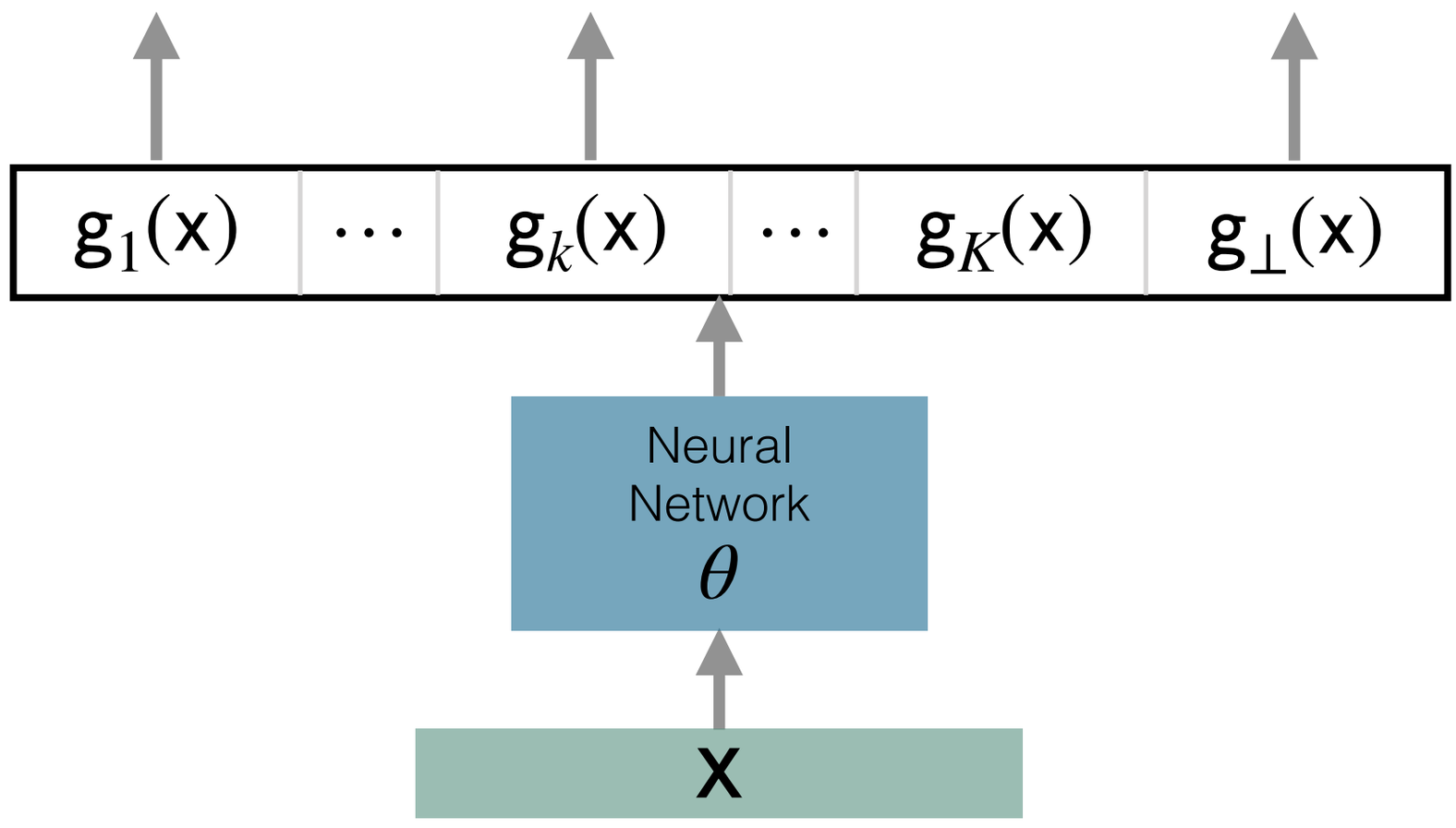
Learning-to-Defer: One-vs-All Parameterization



Learning-to-Defer: One-vs-All Parameterization

$$\mathbb{P}(\mathrm{y} = i \,|\, \mathrm{x}) \approx \sigma\left(\mathrm{g}_i(\mathrm{x})\right)$$

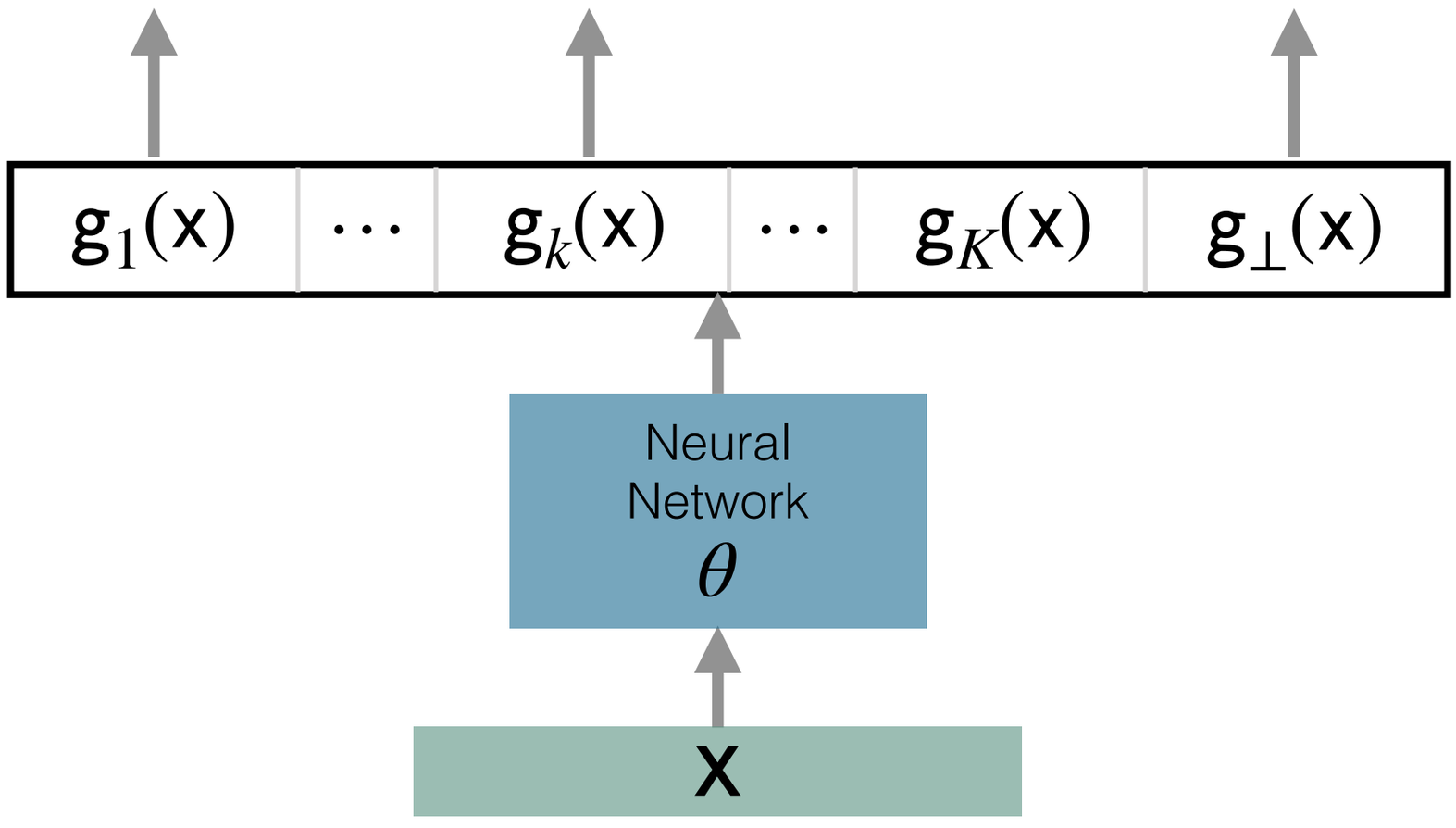
$\sigma\left(\mathrm{g}_1(\mathrm{x})\right)$ $\sigma\left(\mathrm{g}_k(\mathrm{x})\right)$ $\sigma\left(\mathrm{g}_\perp(\mathrm{x})\right)$



Learning-to-Defer: One-vs-All Parameterization

$$\mathbb{P}(\mathsf{m} = \mathsf{y} \,|\, \mathsf{x}) \approx \sigma\left(\mathsf{g}_{\perp}(\mathsf{x})\right)$$

$\sigma\left(\mathsf{g}_1(\mathsf{x})\right)$ $\sigma\left(\mathsf{g}_k(\mathsf{x})\right)$ $\sigma\left(\mathsf{g}_{\perp}(\mathsf{x})\right)$



Learning-to-Defer: One-vs-All Parameterization

Theorem 4.1 + Corollary 4.2 [Verma & Nalisnick, 2022]**:** The one-vs-all loss is a *consistent* surrogate for the 0-1 learning-to-defer loss.

Learning-to-Defer: One-vs-All Parameterization

Theorem 4.1 + Corollary 4.2 [Verma & Nalisnick, 2022]**:** The one-vs-all loss is a *consistent* surrogate for the 0-1 learning-to-defer loss.

Proof: Via method of *error correcting output codes* [Dietterich & Bakiri, 1995], by applying Ramaswamy et al.'s [2018] Equation 1.

Learning-to-Defer: One-vs-All Parameterization

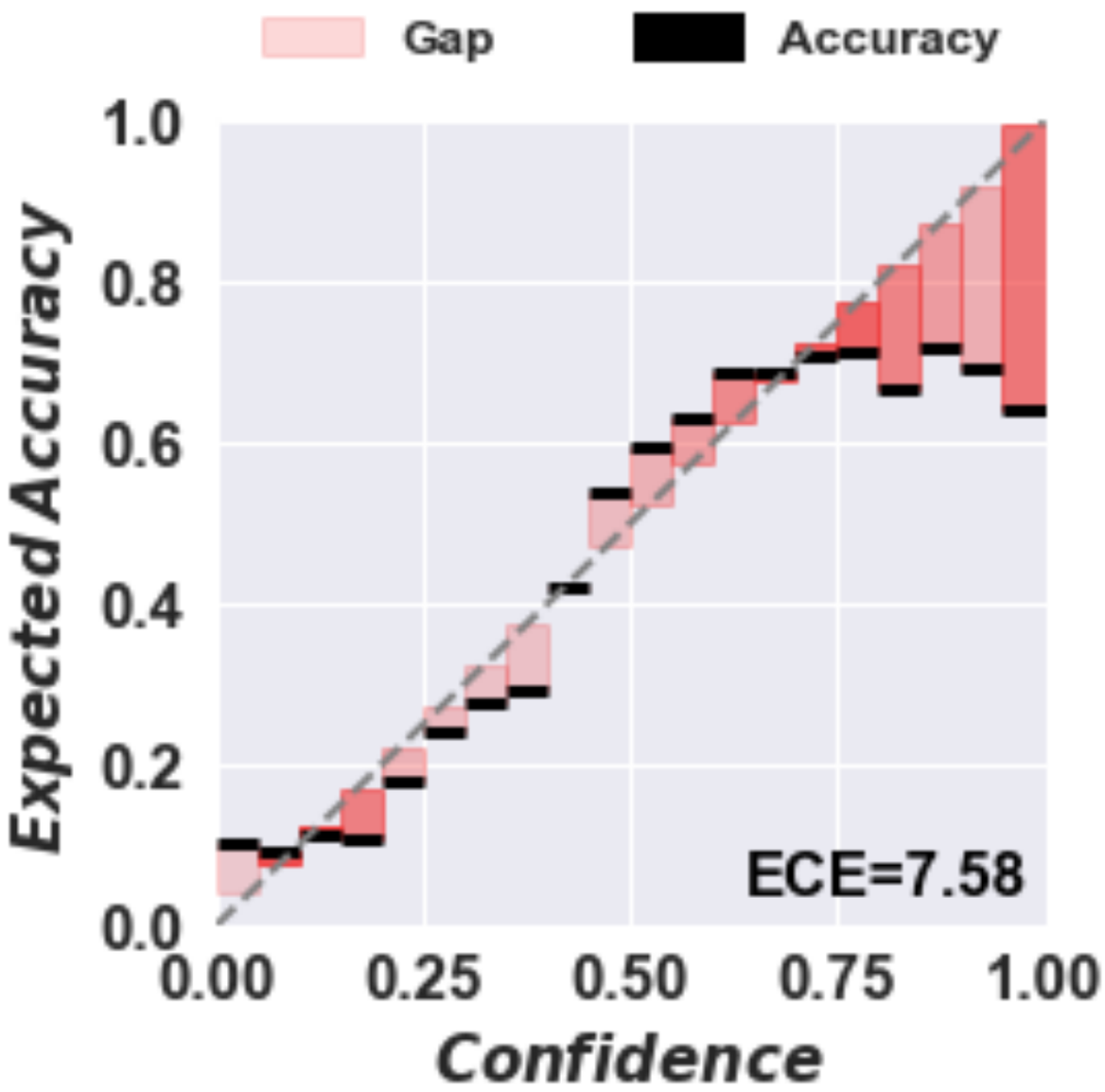
Theorem 4.1 + Corollary 4.2 [Verma & Nalisnick, 2022]**:** The one-vs-all loss is a *consistent* surrogate for the 0-1 learning-to-defer loss.

Proof: Via method of *error correcting output codes* [Dietterich & Bakiri, 1995], by applying Ramaswamy et al.'s [2018] Equation 1.

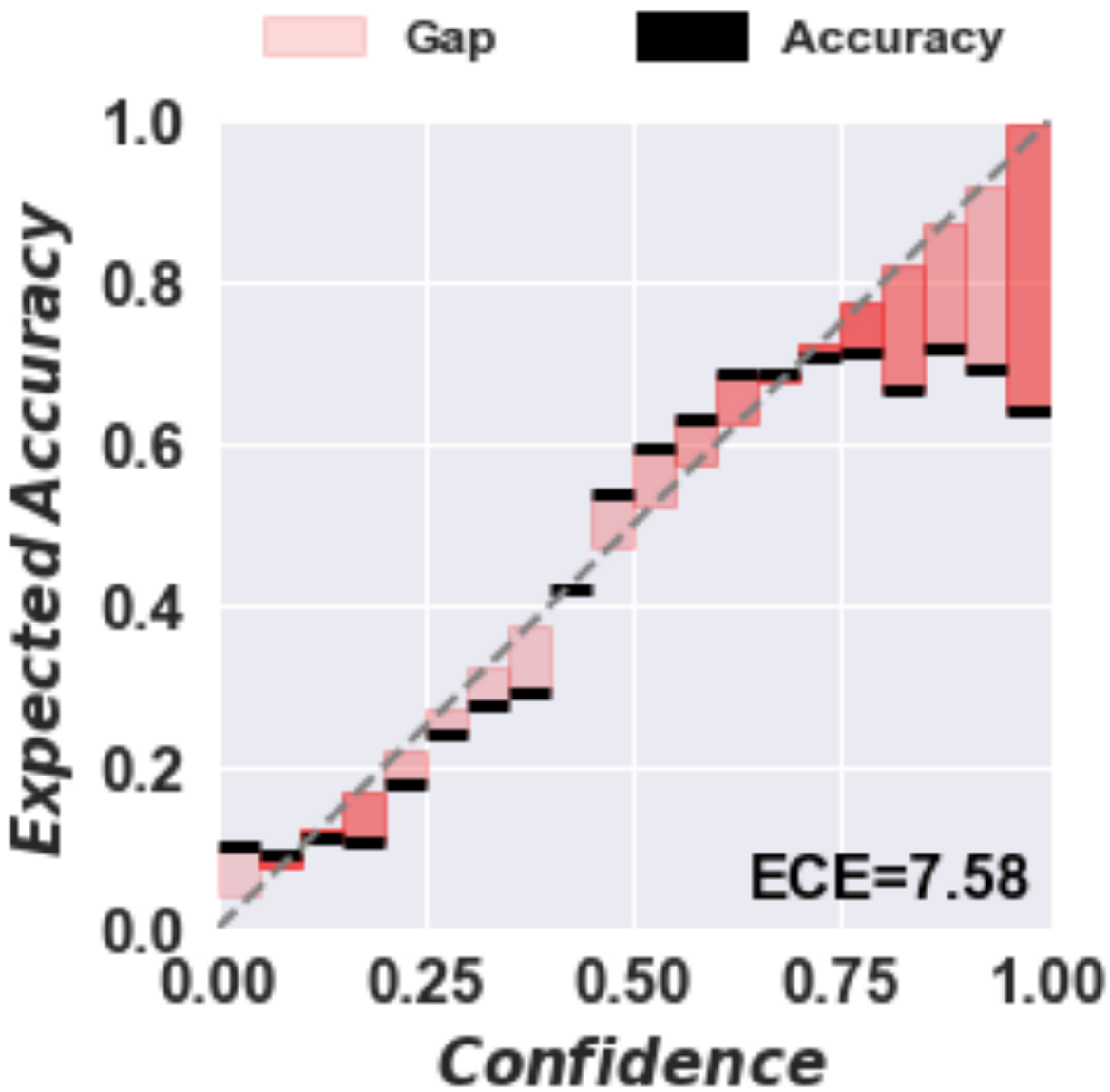
This means that our one-vs-all loss is as theoretically well-justified as Mozannar & Sontag's softmax-based surrogate.

Does the one-vs-all loss result in **better calibrated** models in practice?

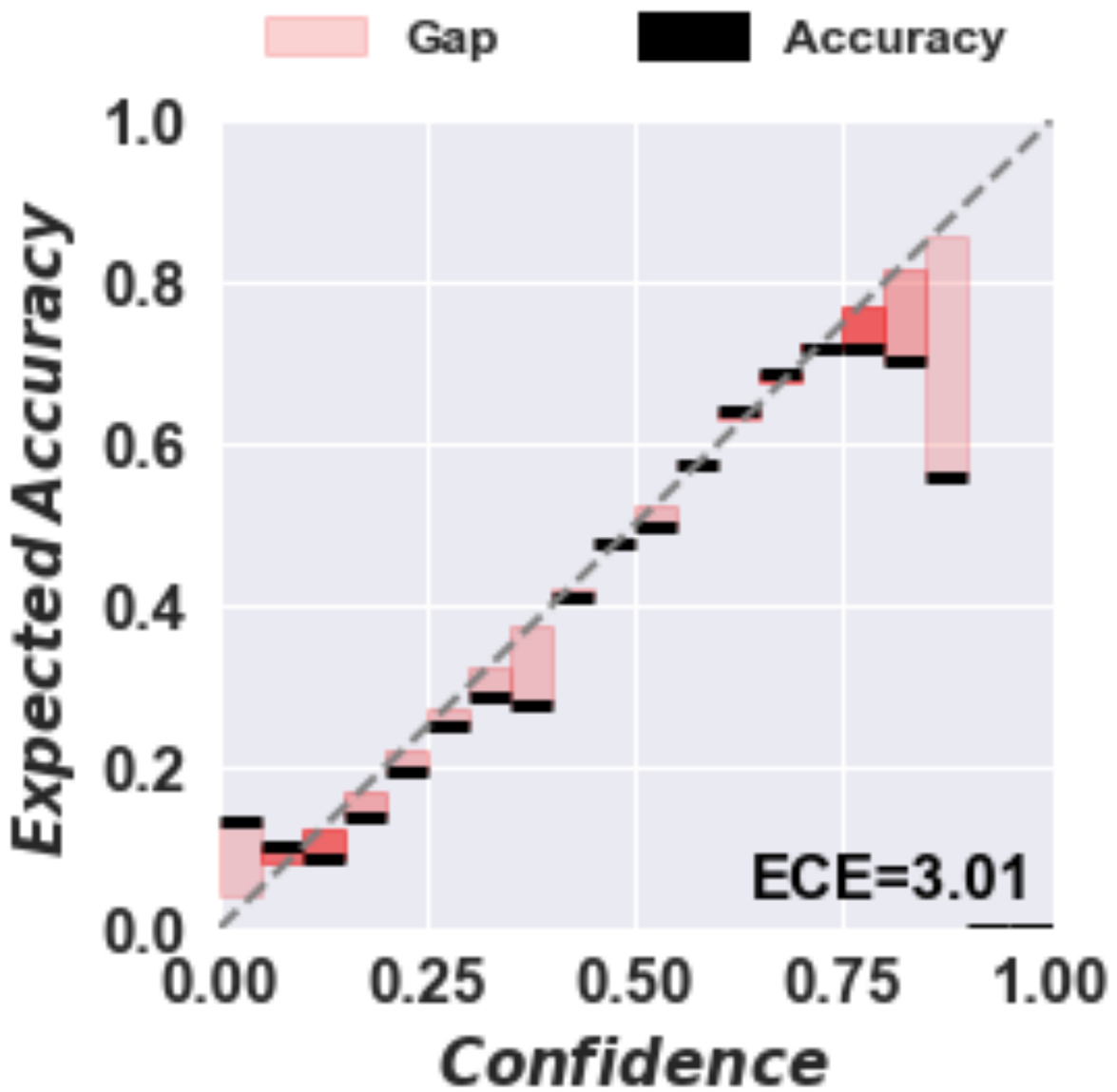
CIFAR-10: Softmax



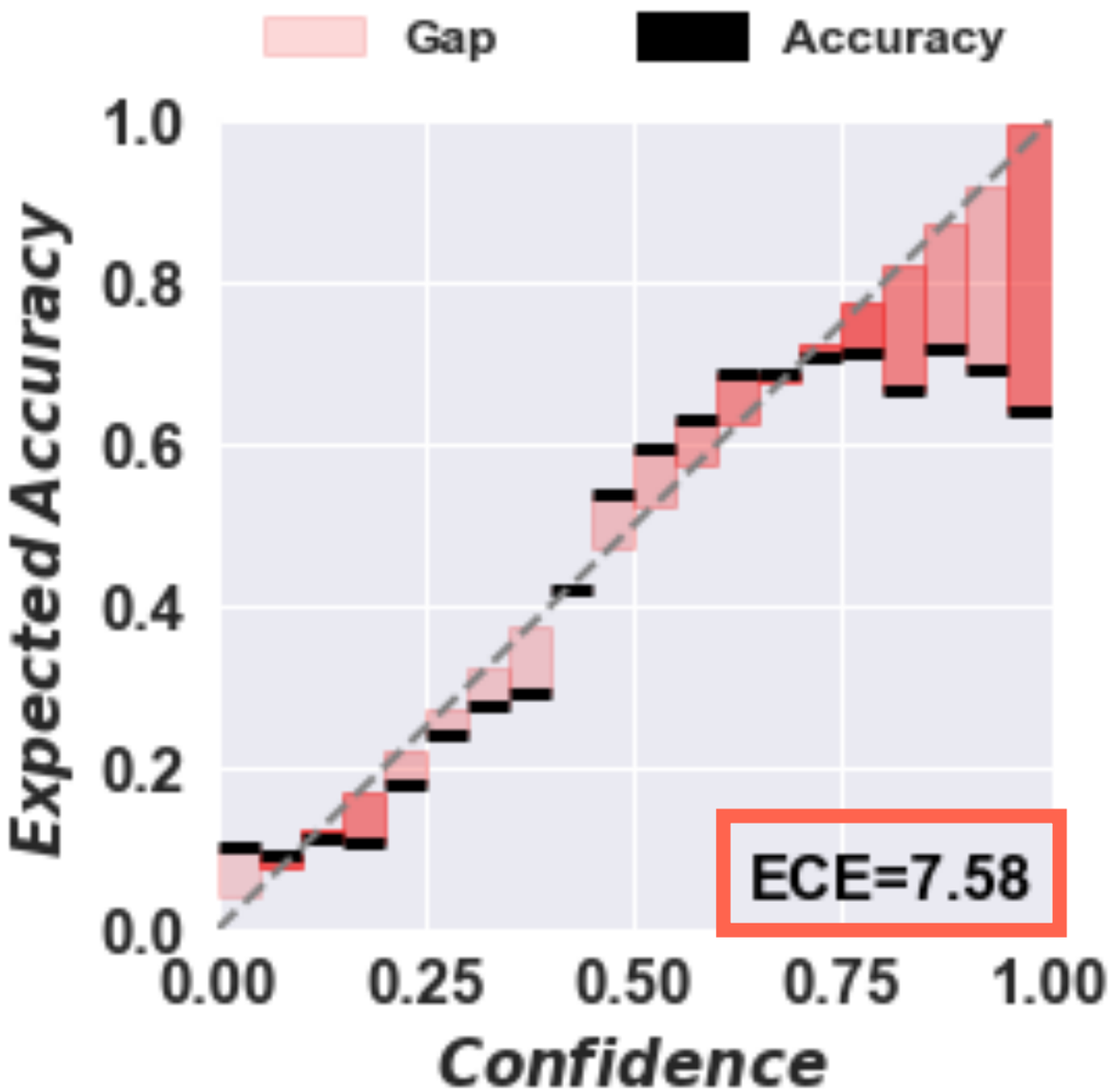
CIFAR-10: Softmax



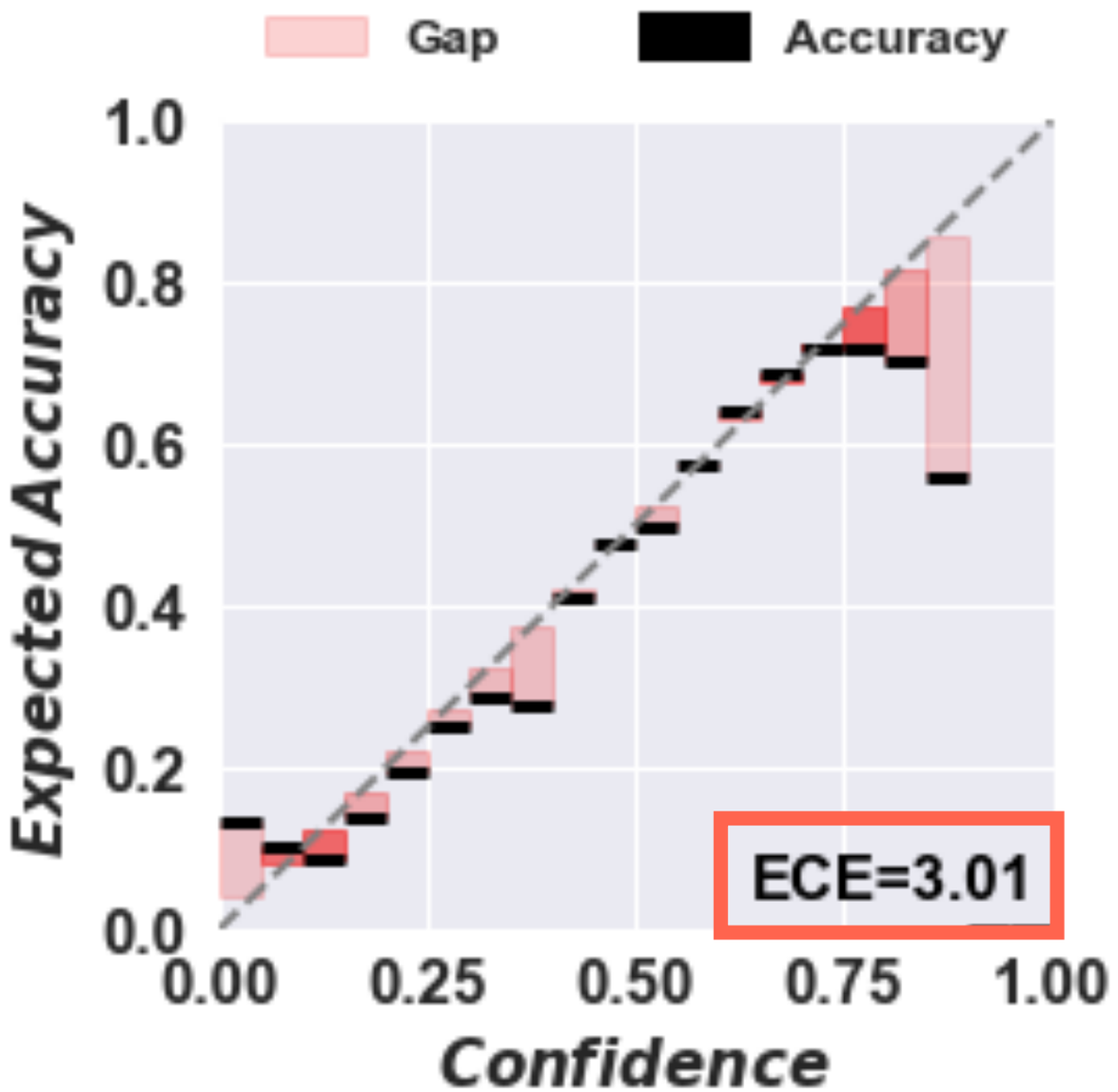
CIFAR-10: One-vs-All



CIFAR-10: Softmax



CIFAR-10: One-vs-All



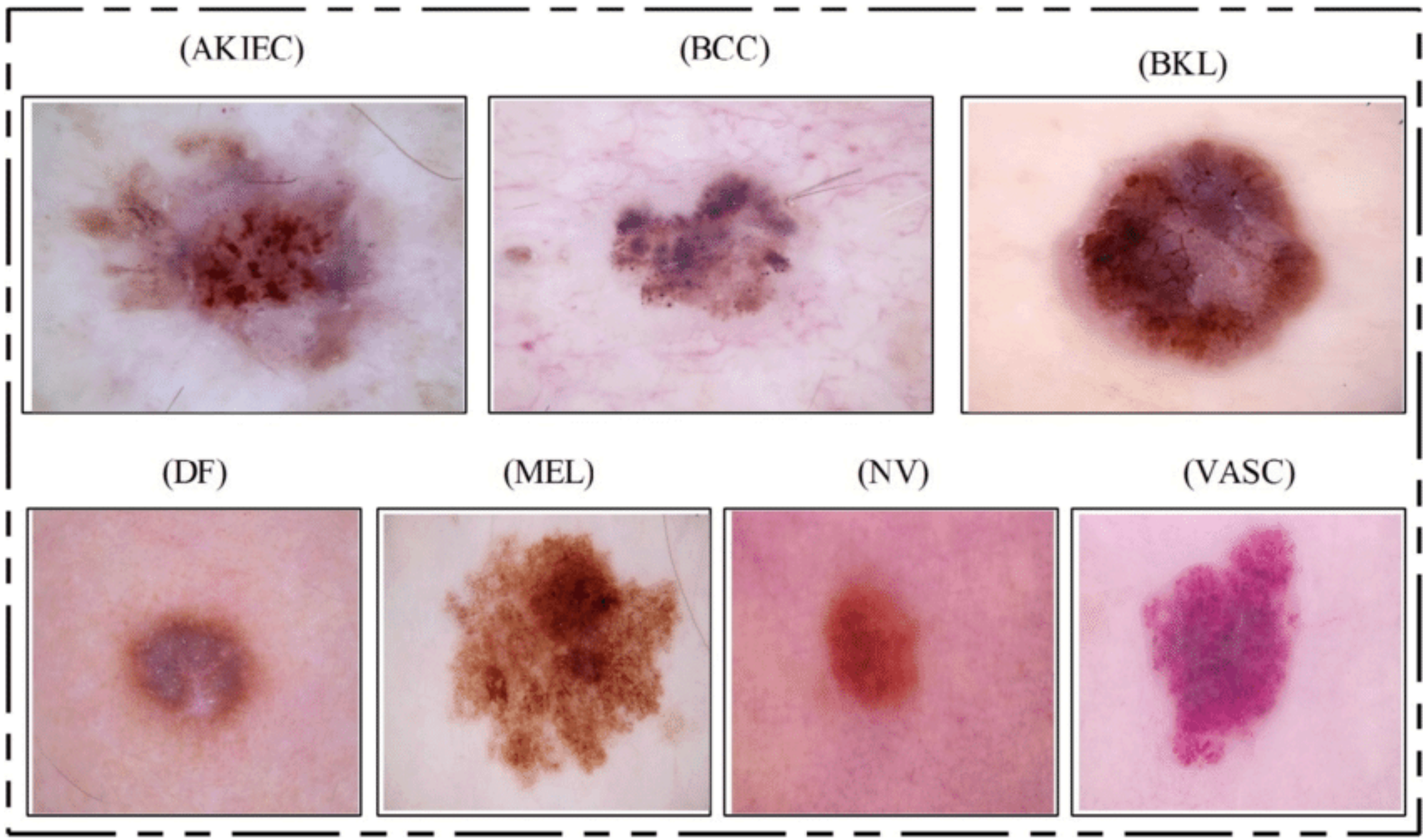
CIFAR-10: ECE Across Parameterizations

	OvA	Softmax	Proxy
Both Random			
Random Expert	$\sigma\left(g_{\perp}(x)\right)$	$\frac{p_{\perp}(x)}{1-p_{\perp}(x)}$	$p_{\perp}(x)$
Random Data			
Both Useful			

CIFAR-10: ECE Across Parameterizations

	OvA	Softmax	Proxy
Both Random	0.53	0.97	**0.04**
Random Expert	**0.68**	3.72	2.83
Random Data	**2.05**	2.07	39.06
Both Useful	**1.68**	3.32	37.15

HAM10000: Skin Lesion Classification



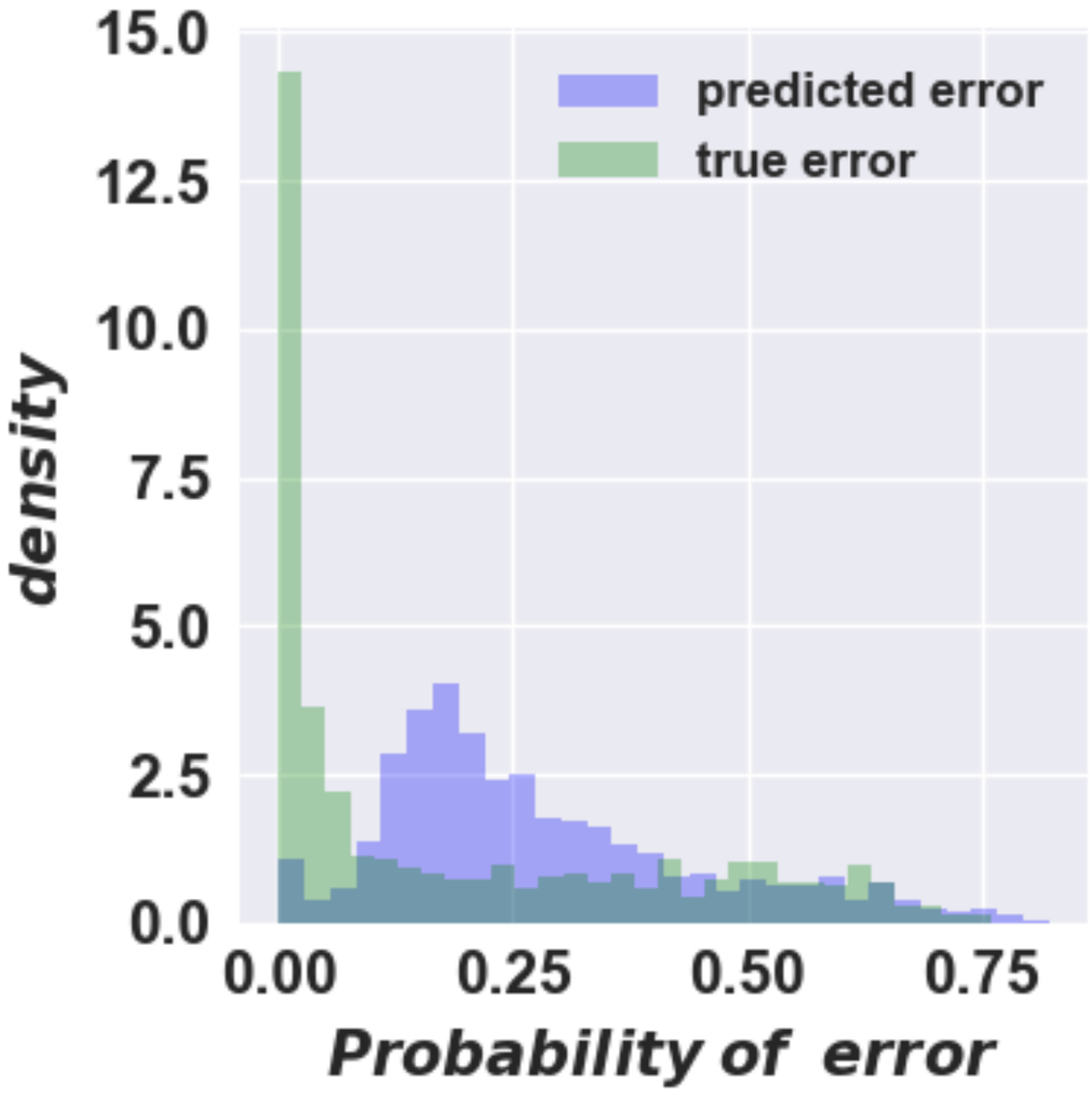
HAM10000: Skin Lesion Classification

Softmax

One-vs-All

HAM10000: Skin Lesion Classification

Softmax

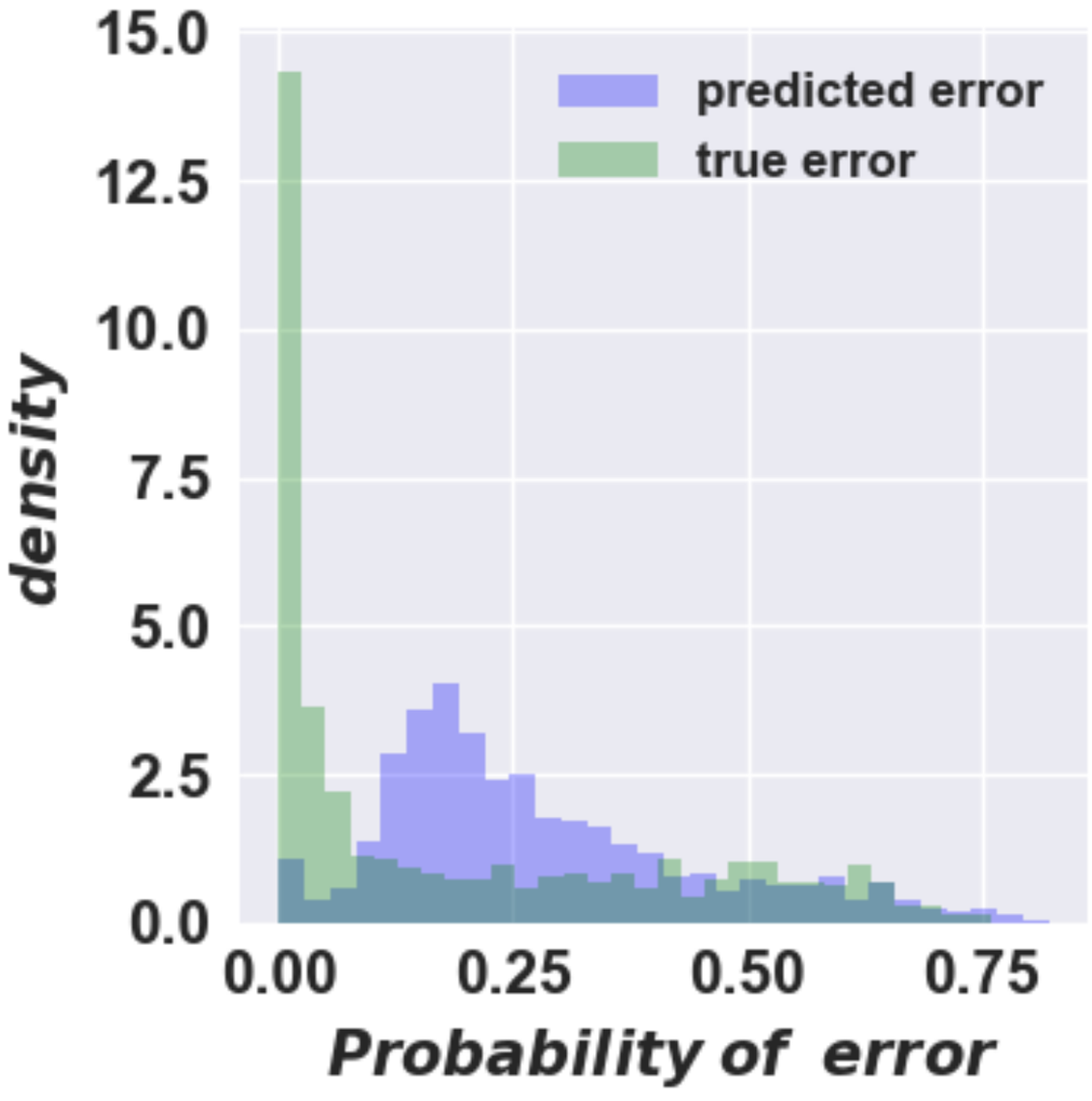


$1 - p_m(x)$

One-vs-All

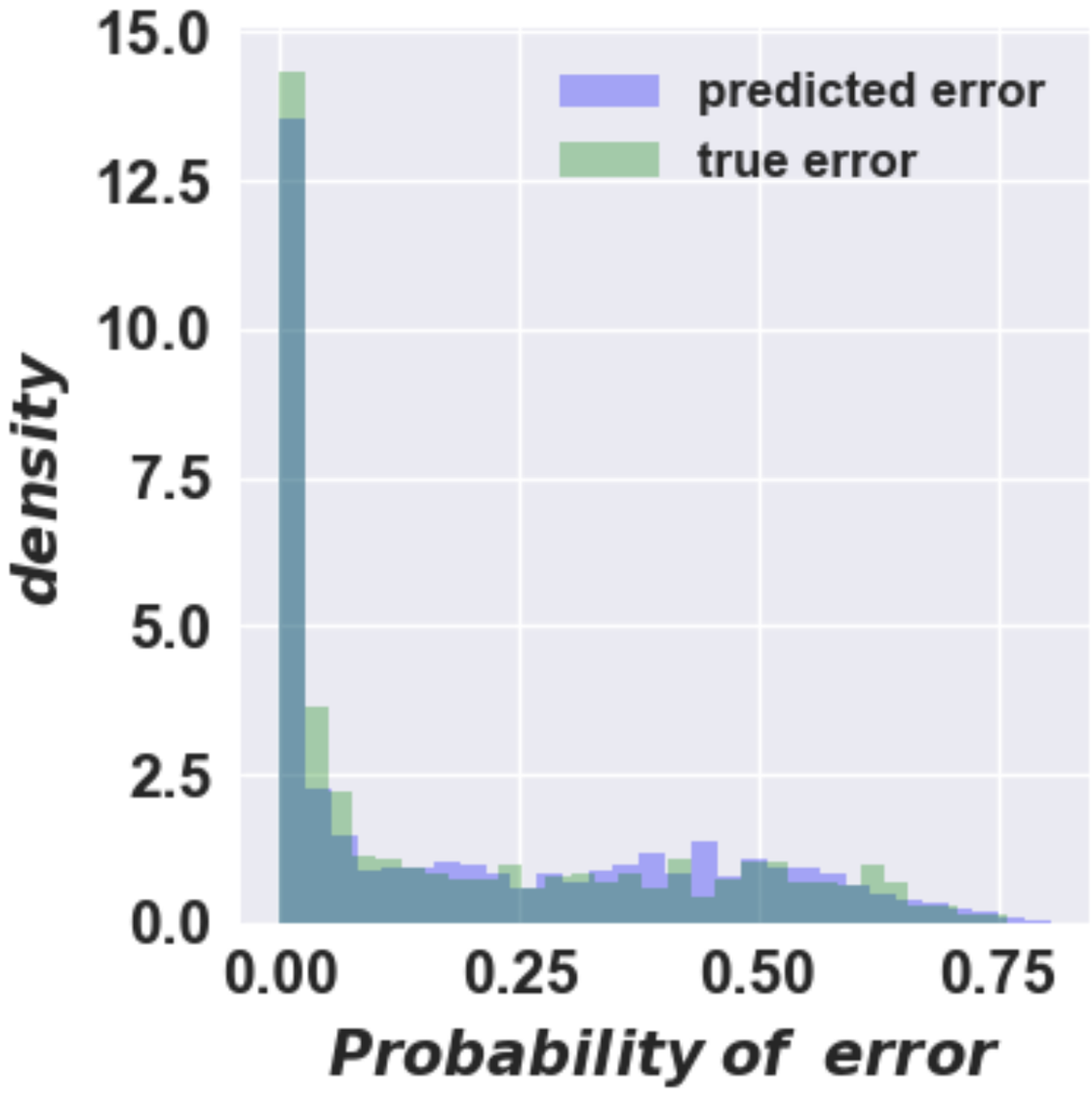
HAM10000: Skin Lesion Classification

Softmax



$1 - p_m(x)$

One-vs-All

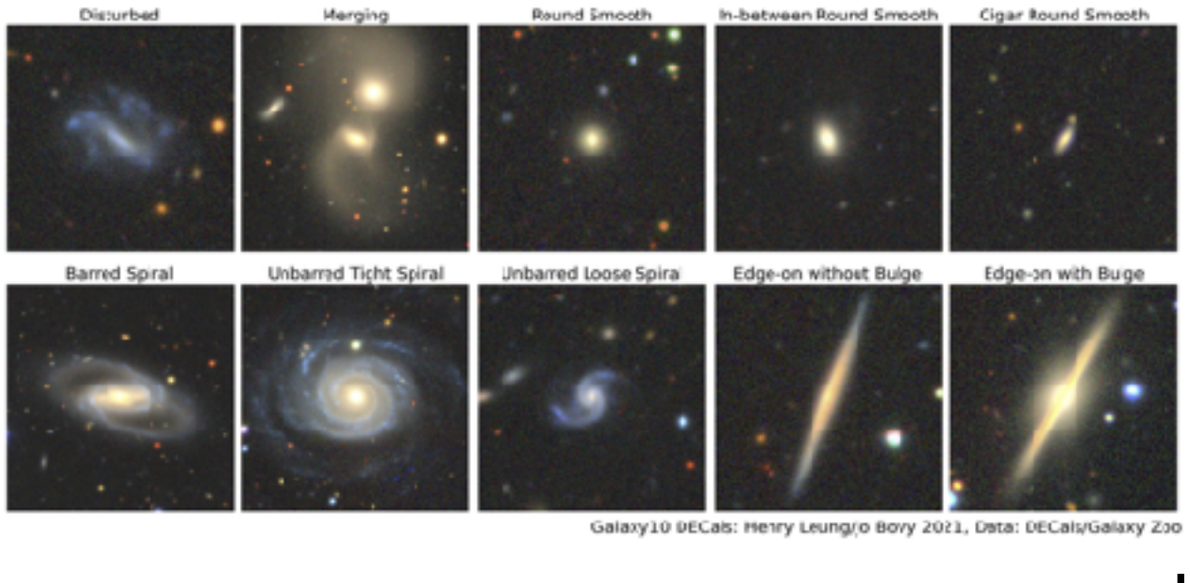


$1 - p_m^{OvA}(x)$

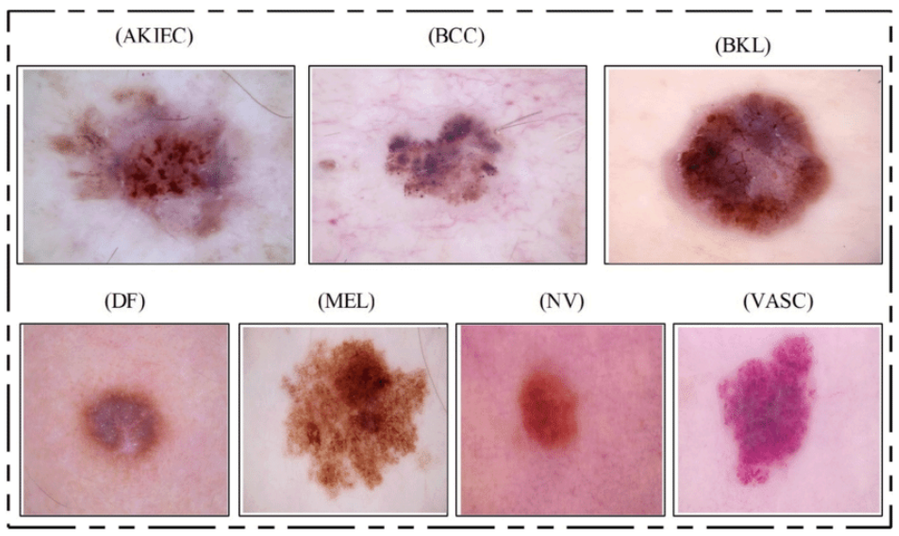
Does the one-vs-all loss result in **more accurate** models in practice?

one-vs-all (ours) softmax confidence score differentiable triage

Galaxy Zoo



HAM10000



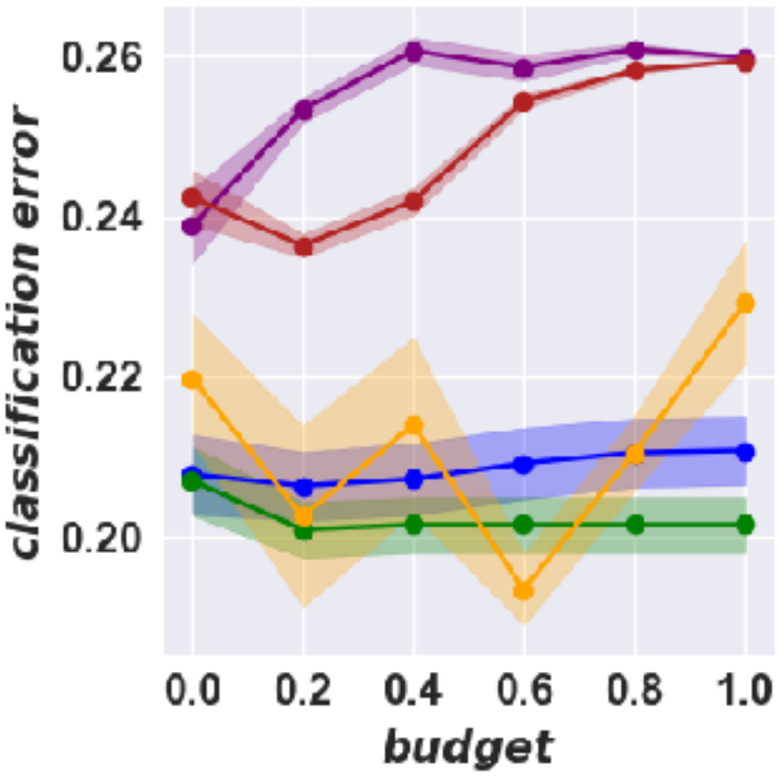
Hate Speech

[Davidson et al., ICWSM 2017]

one-vs-all (ours) softmax confidence score differentiable triage

Galaxy Zoo

HAM10000

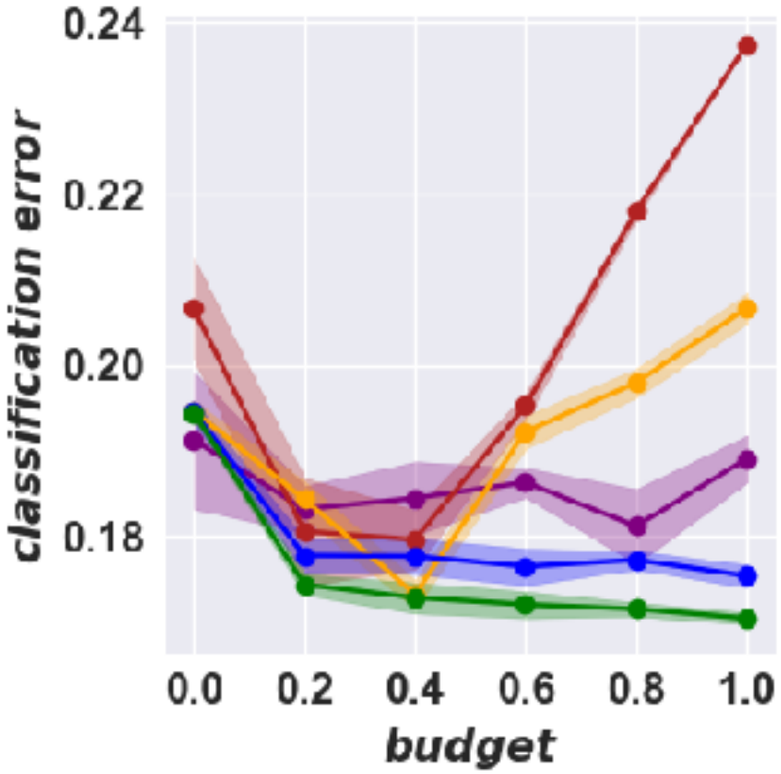


Hate Speech

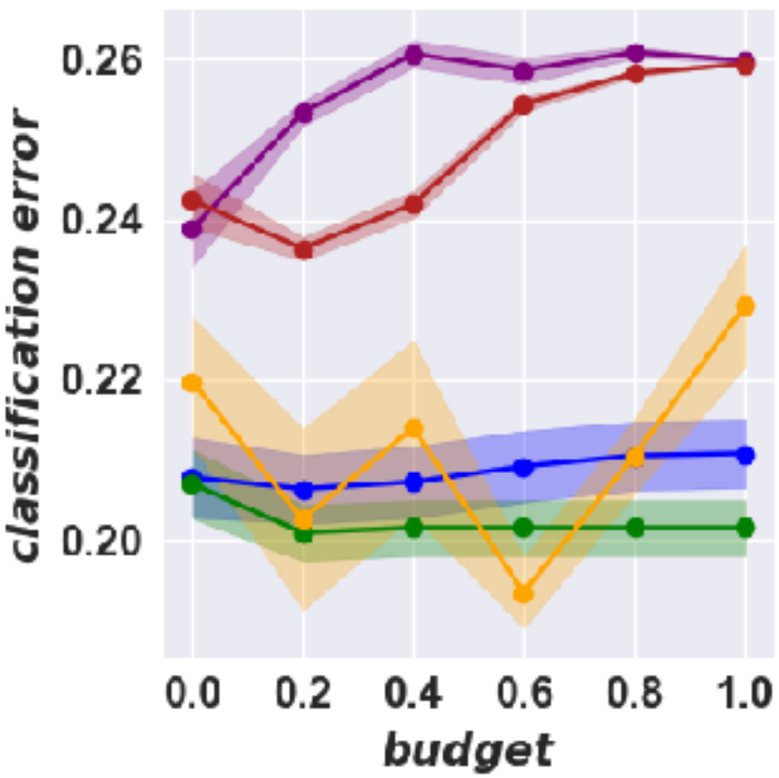
[Davidson et al., ICWSM 2017]

one-vs-all (ours) softmax confidence score differentiable triage

Galaxy Zoo



HAM10000

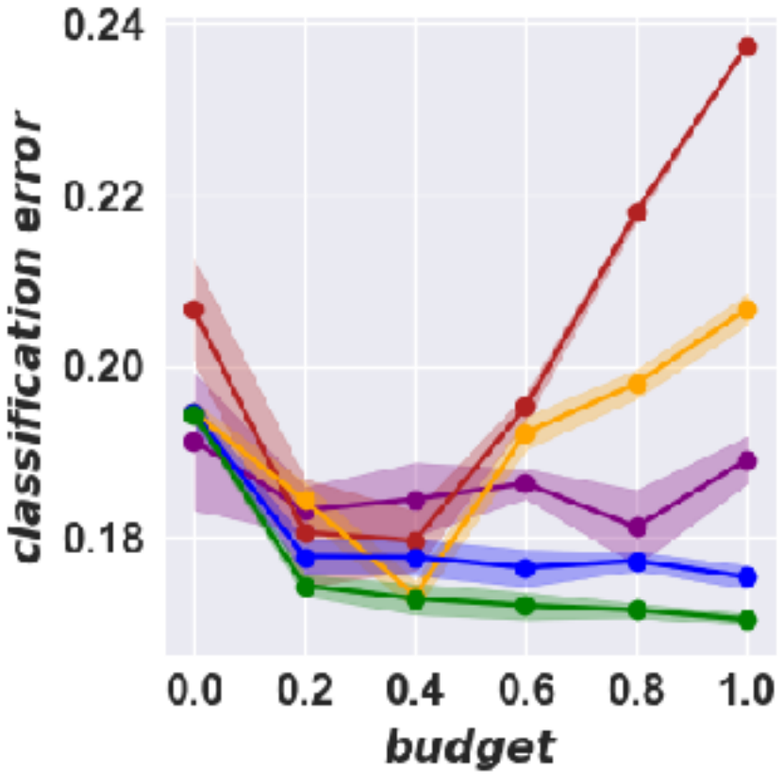


Hate Speech

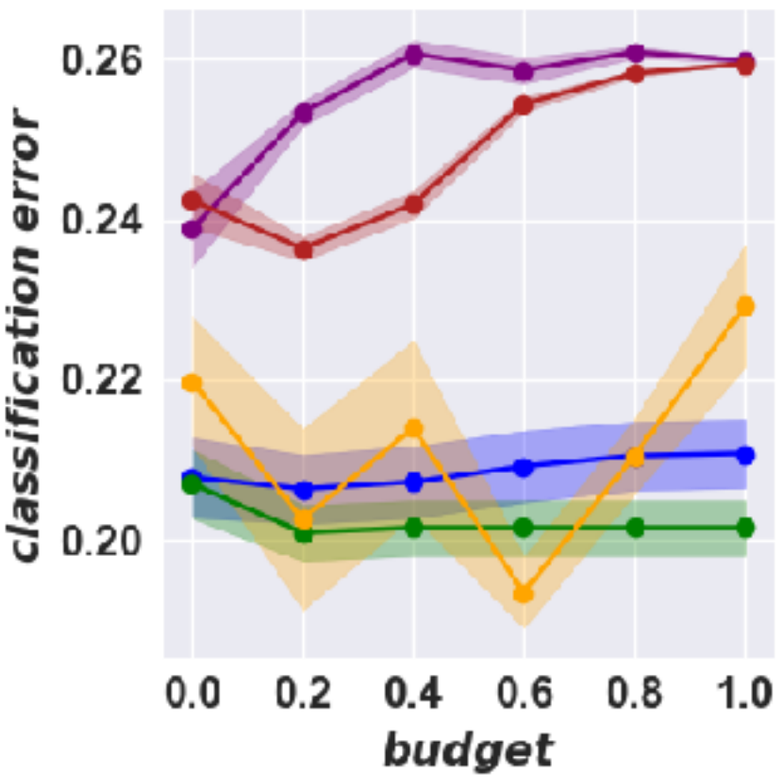
[Davidson et al., ICWSM 2017]

one-vs-all (ours) softmax confidence score differentiable triage

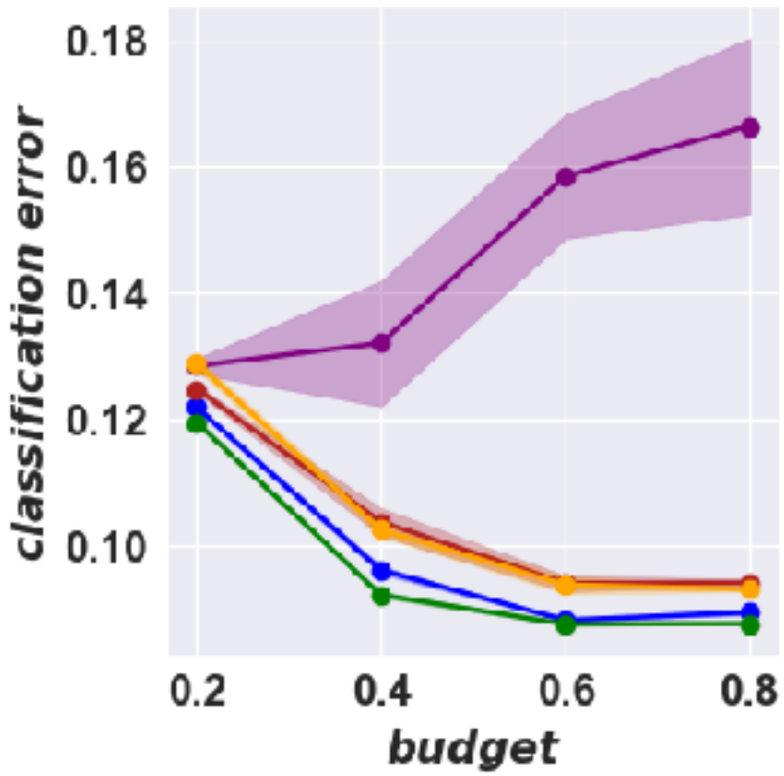
Galaxy Zoo



HAM10000

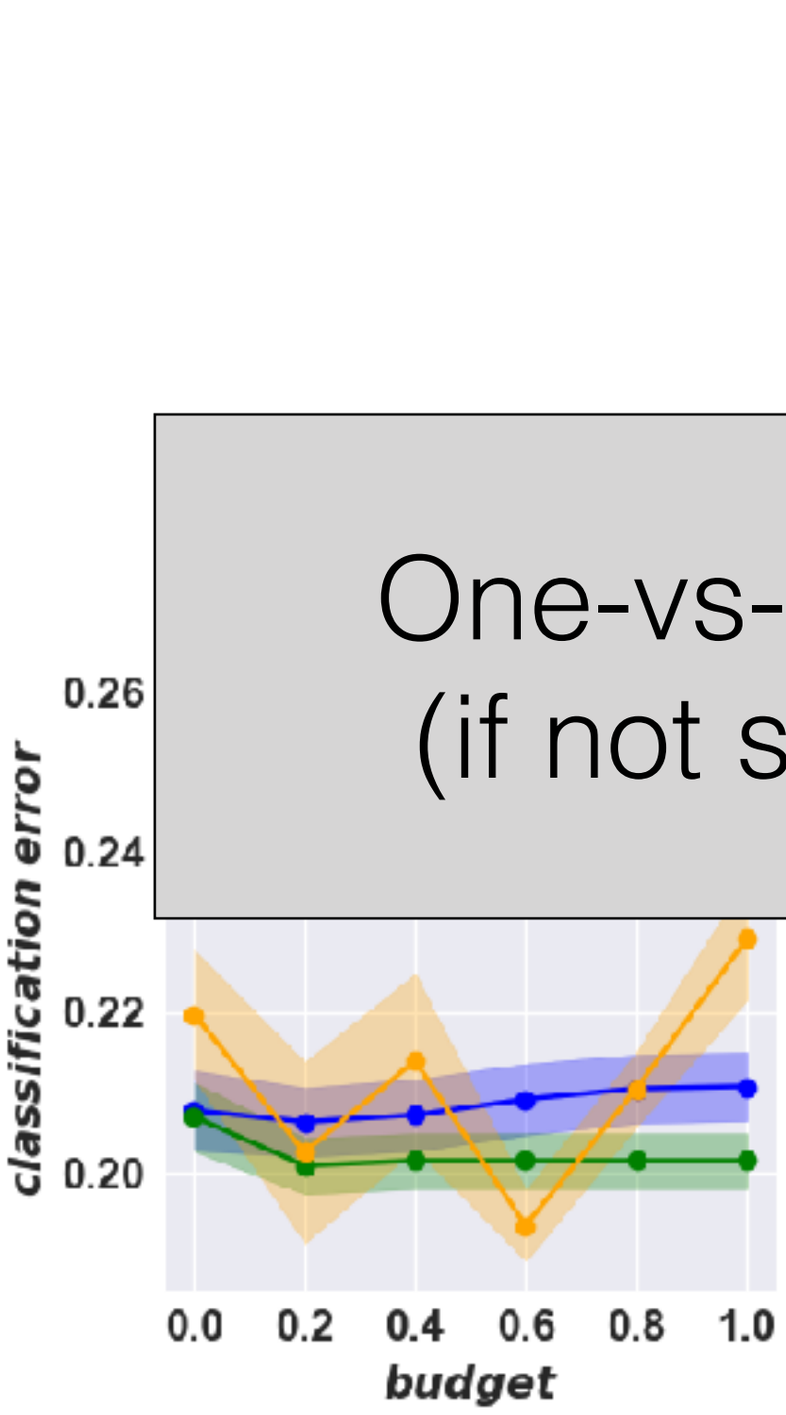


Hate Speech

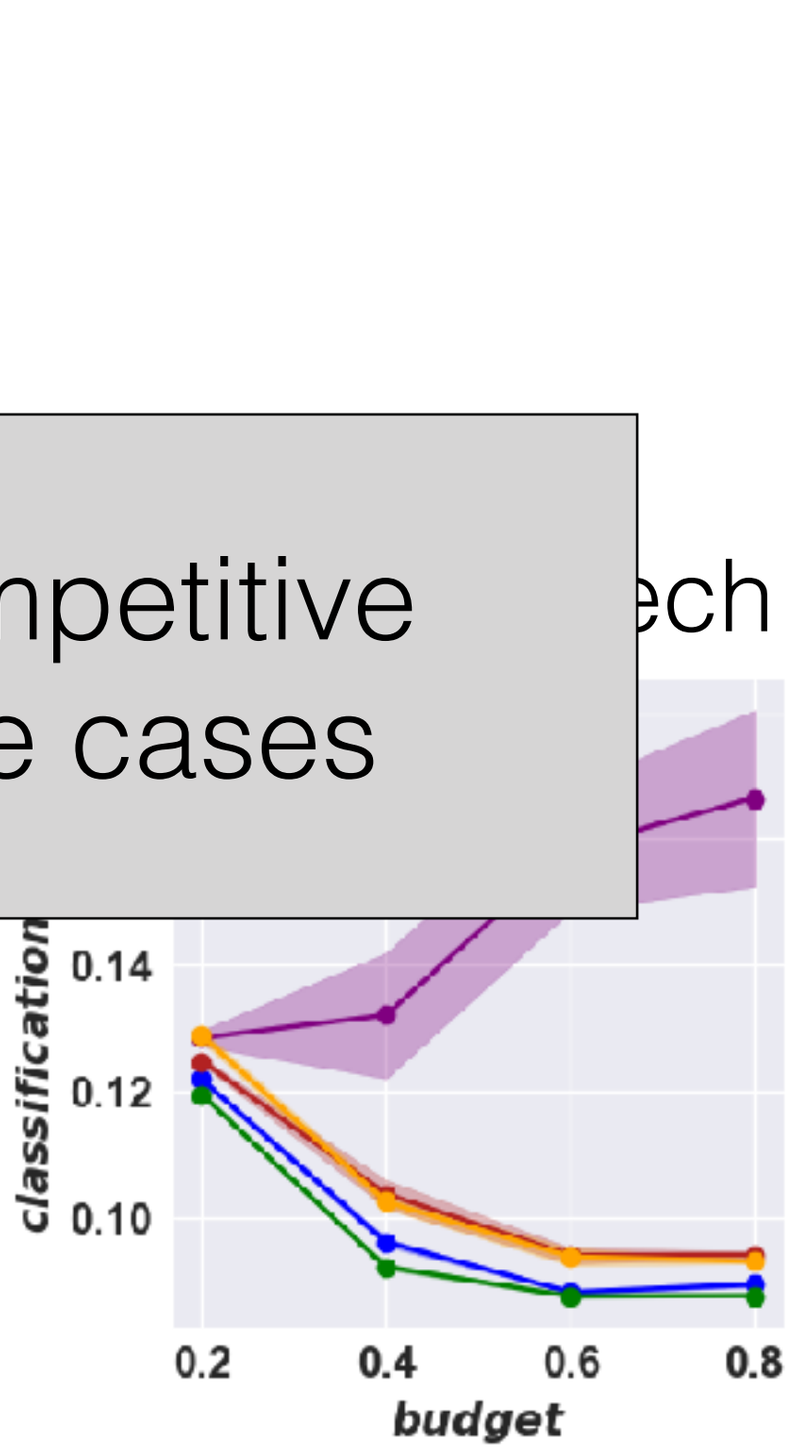


one-vs-all (ours) softmax confidence score differentiable triage

Galaxy Zoo



One-vs-all models are competitive (if not superior) in all three cases



Summary

- ⊗ We show that the softmax-based loss for learning-to-defer can produce invalid estimators for expert correctness.
- ⊗ Our one-vs-all formulation is retains theoretical and practical benefits while also producing calibrated estimates of expert correctness.

Future Work

⊗ Learning-to-defer is still under-studied!

⊗ One problem: heavy dependence on supervised data.

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

[Submitted on 8 Feb 2022]

Calibrated Learning to Defer with One-vs-All Classifiers

Rajeev Verma, Eric Nalisnick

The learning to defer (L2D) framework has the potential to make AI systems safer. For a given input, the system can defer the decision to a human if the human is more likely than the model to take the correct action. We study the calibration of L2D systems, investigating if the probabilities they output are sound. We find that Mozannar & Sontag's (2020) multiclass framework is not calibrated with respect to expert correctness. Moreover, it is not even guaranteed to produce valid probabilities due to its parameterization being degenerate for this purpose. We propose an L2D system based on one-vs-all classifiers that is able to produce calibrated probabilities of expert correctness. Furthermore, our loss function is also a consistent surrogate for multiclass L2D, like Mozannar & Sontag's (2020). Our experiments verify that not only is our system calibrated, but this benefit comes at no cost to accuracy. Our model's accuracy is always comparable (and often superior) to Mozannar & Sontag's (2020) model's in tasks ranging from hate speech detection to galaxy classification to diagnosis of skin lesions.

Subjects: **Machine Learning (cs.LG)**; Machine Learning (stat.ML)
Cite as: arXiv:2202.03673 [**cs.LG**]
(or arXiv:2202.03673v1 [**cs.LG**] for this version)

Thank you. Questions?