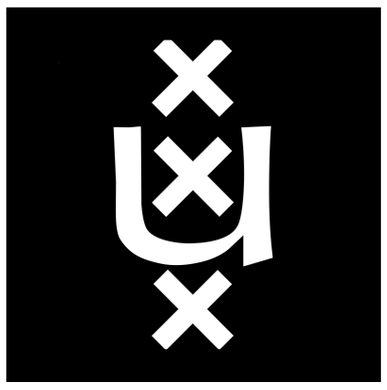
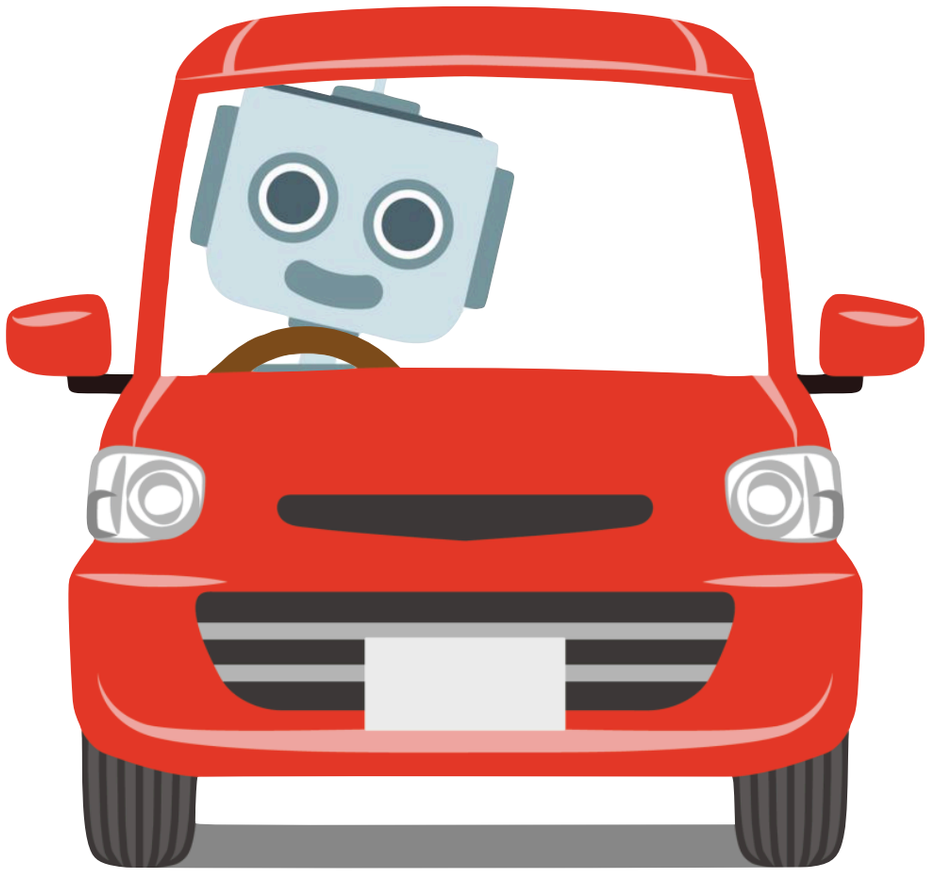
Learning to Defer to One, Multiple, or a Population of Expert(s)

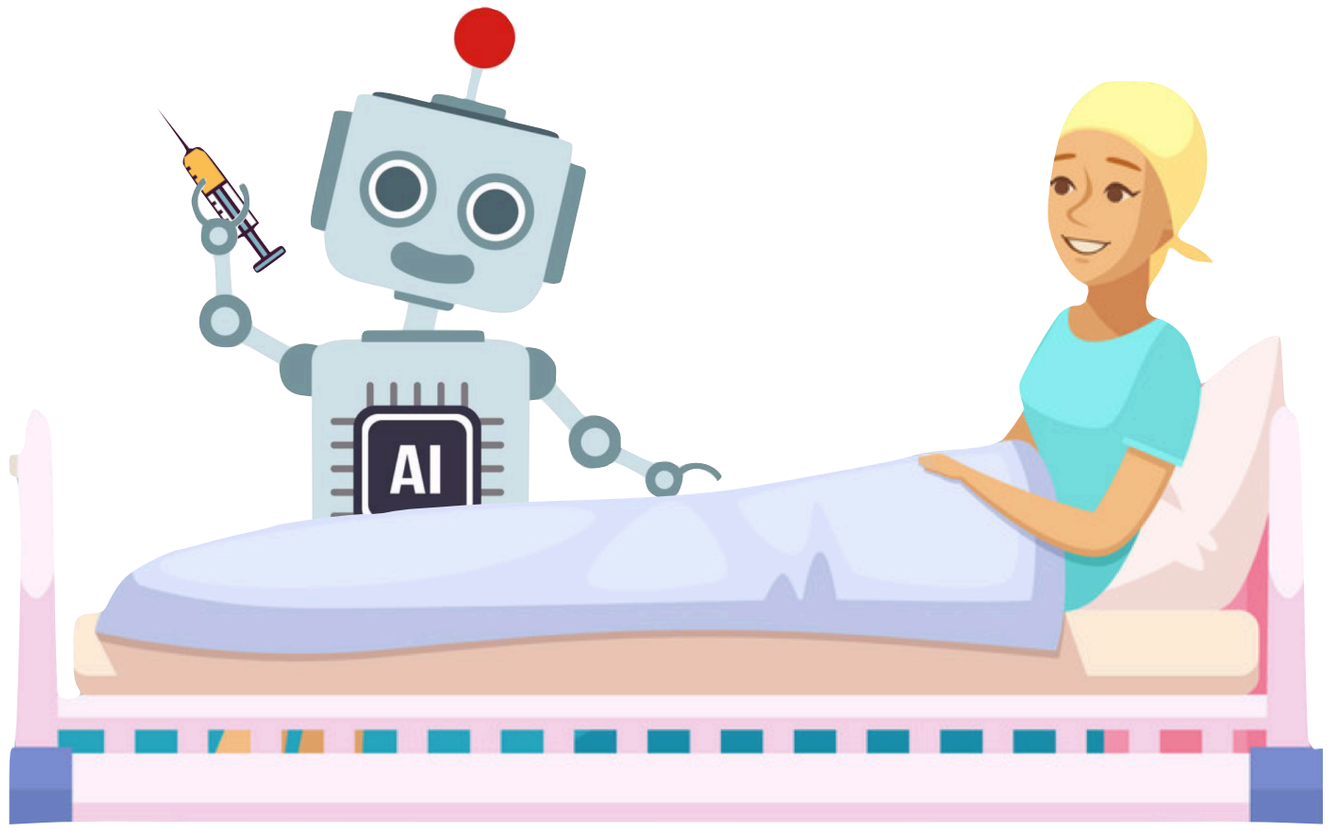
Eric Nalisnick

University of Amsterdam






Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
Happy Day 2014: International Day of Happiness today. Don't Worry Be Happy
314,268 people like this.
AI


AI

Los Angeles Times

LOG IN

A Tesla mystery: Why didn't auto-braking stop these crashes?

Search for people, places and things

Olenna Home Find Friends

Olenna Mason
Edit Profile

Welcome
News Feed
Messages
Events

Update Status Add Photos/Video

What's on your mind?

Post

TRENDING Learn More

Wheel of Fortune: Wheel Of Fortu... contestant guesses puzzle with ... letters

#happyday: Happy Day 2014: International Day of Happiness today. Don't Worry Be Happy

Christophe...

from Net Politics *and* Digital and Cyberspace Policy Program

Facebook's Content Moderation Failures in Ethiopia

Facebook has failed to moderate content in underserved countries. Facebook and other social media companies must invest more in local content moderation, instead of relying on global AI systems.

Amhara militia members ride in the back of a truck towards a fight with the Tigray People's Liberation Front. Reuters/Tiksa Negeri

Blog Post by Caroline Allen, Guest Contributor

April 19, 2022 2:36 pm (EST)

ARTIFICIAL INTELLIGENCE

Google's medical AI was super accurate in a lab. Real life was a different story.

If AI is really going to ma...
works when real humar...

By Will Douglas Heaven

Medscape Tuesday, December 13, 2022

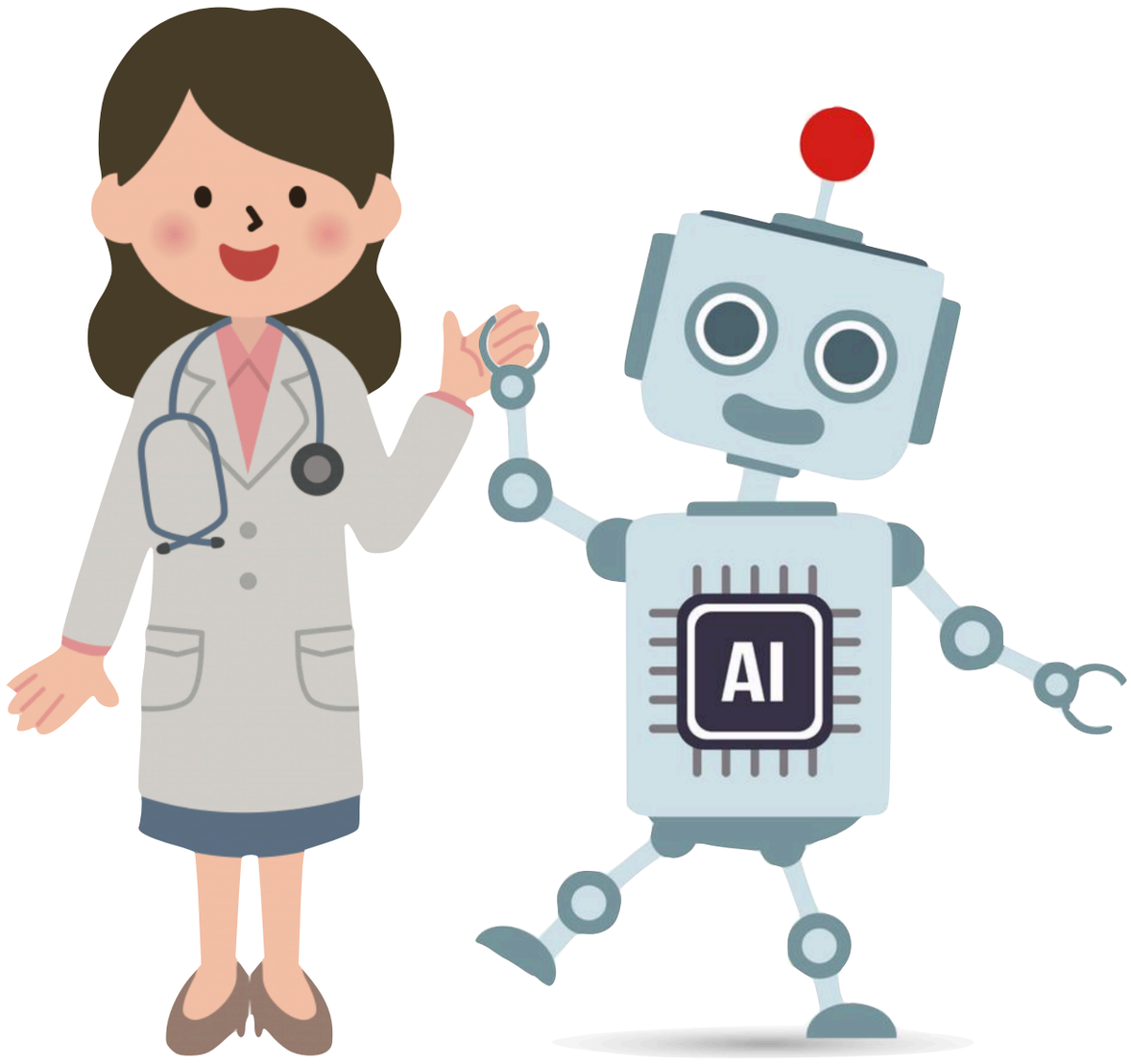
NEWS & PERSPECTIVE DRUGS & DISEASES CME & EDUCATION ACADEMY VIDEO DECISION POINT

News > Medscape Medical News > Conference News > CHEST 2022

Sepsis Predictor Tool Falls Short in Emergency Setting

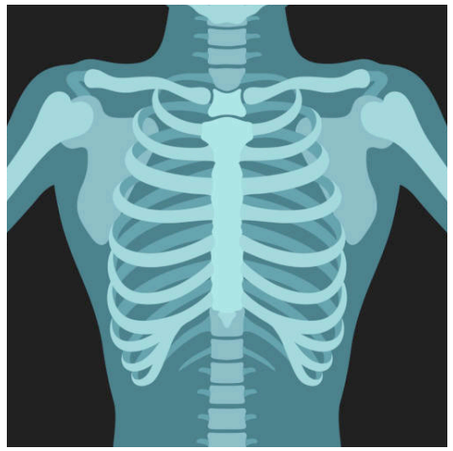
Heidi Splete

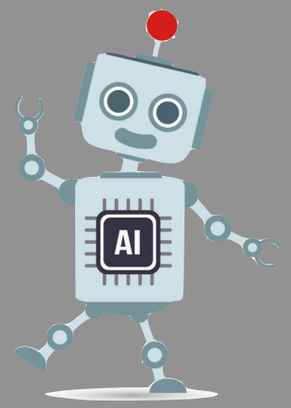
October 17, 2022



human-AI collaboration

input features



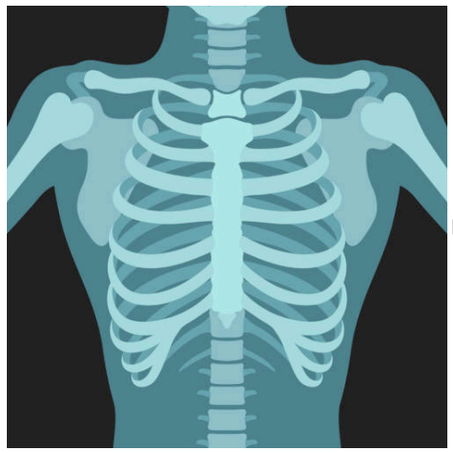


classifier

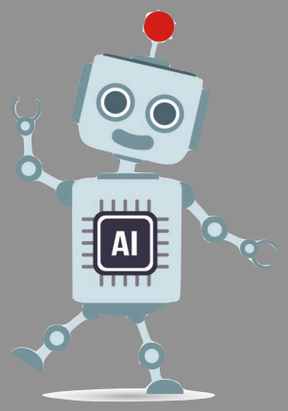
expert

learning to defer (to an expert)

input features



allocation mechanism

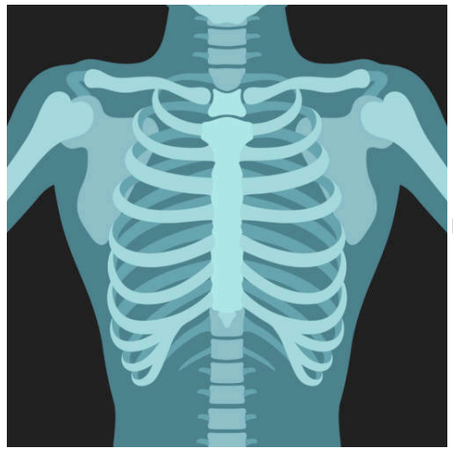


classifier

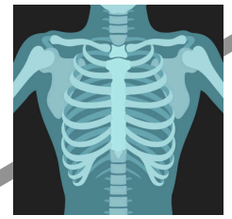
expert

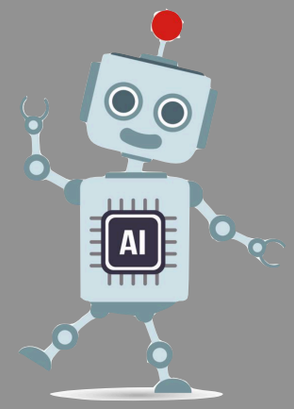
learning to defer (to an expert)

input features



allocation mechanism



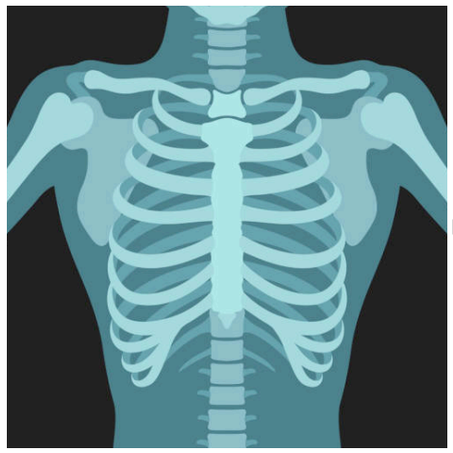


classifier

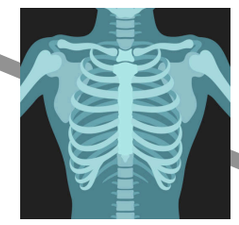
expert

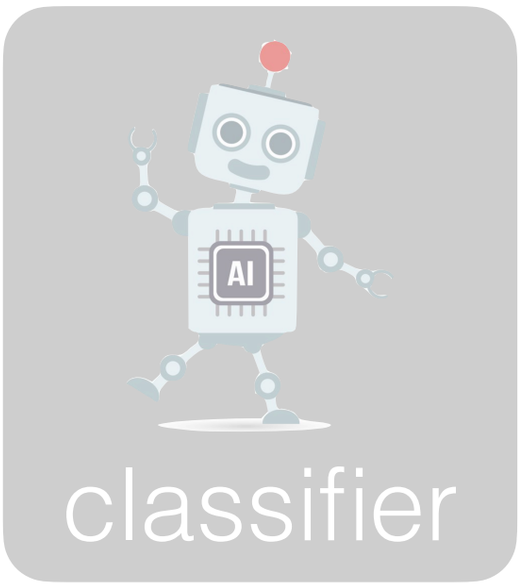
learning to defer (to an expert)

input features



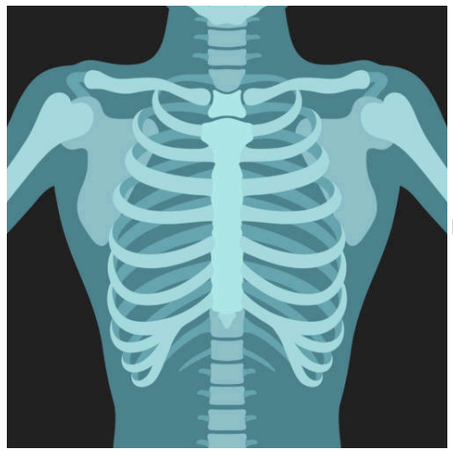
allocation mechanism



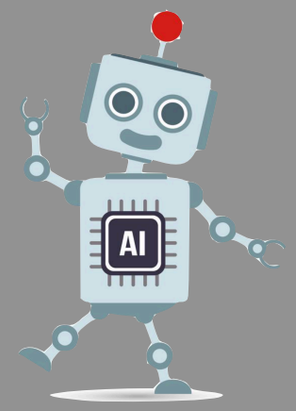


learning to defer (to an expert)

input features



allocation mechanism

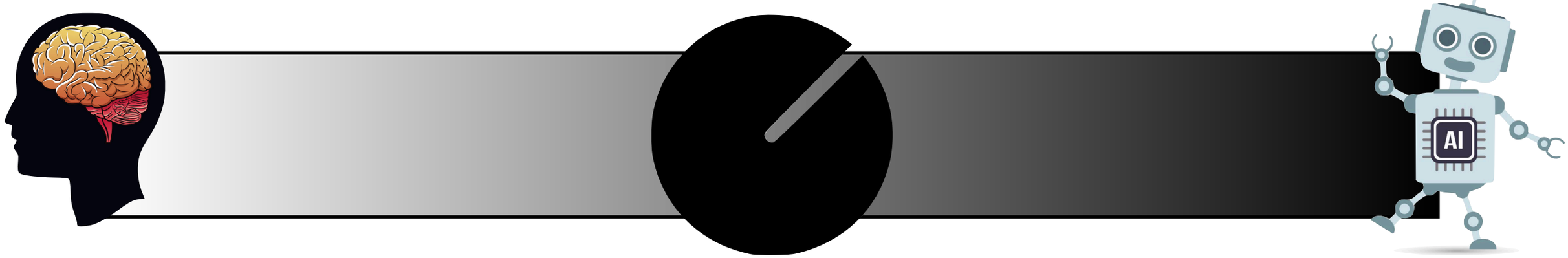


classifier

expert

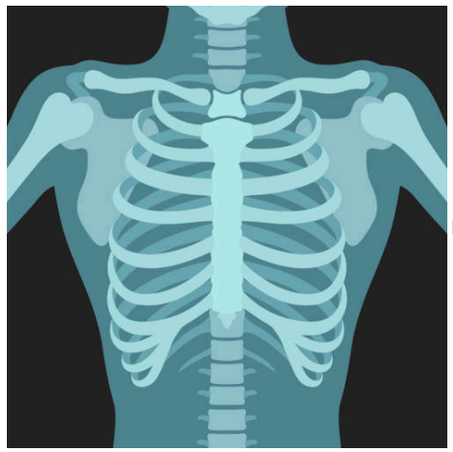
safe and robust semi-automation
via expert handling the hardest cases

safe, gradual automation

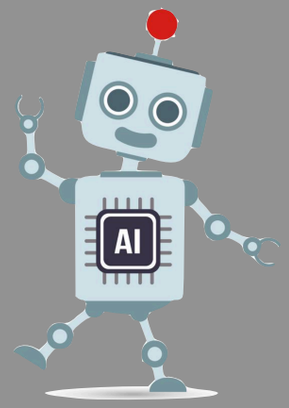


- ⊗ single expert
  - ⊗ softmax surrogate loss
  - ⊗ improving calibration via one-vs-all
- ⊗ multiple experts
  - ⊗ surrogate losses
  - ⊗ conformal sets of experts
- ⊗ population of experts
  - ⊗ surrogate losses
  - ⊗ meta-learning a rejector

input features



allocation mechanism

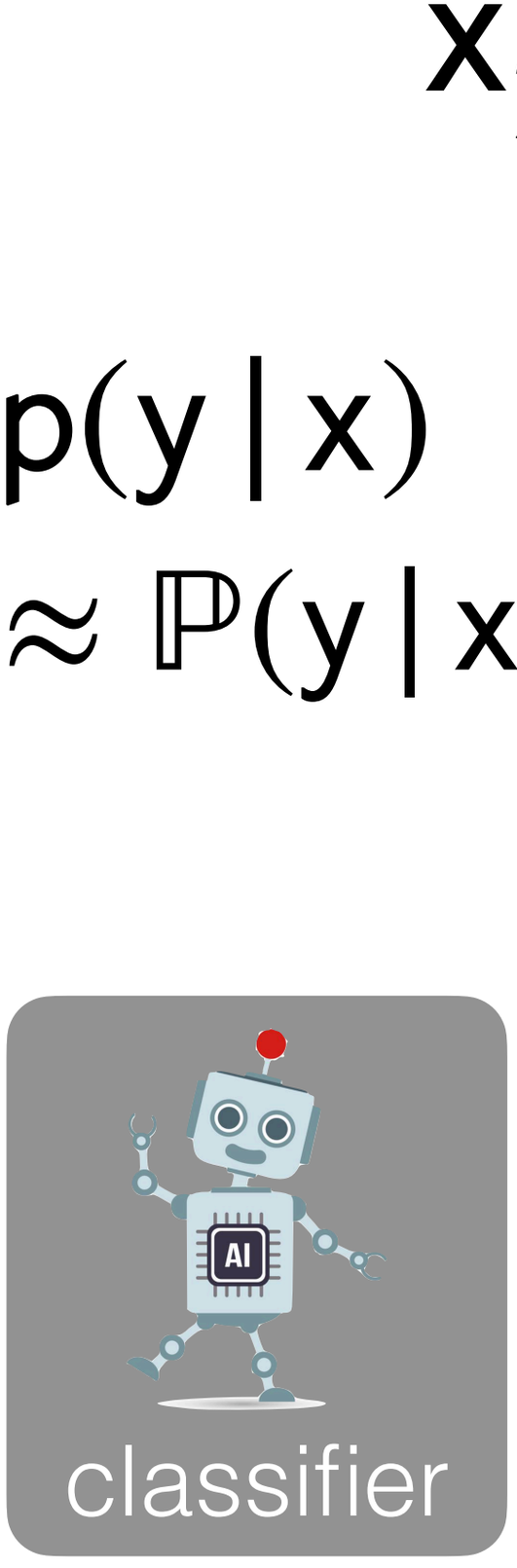


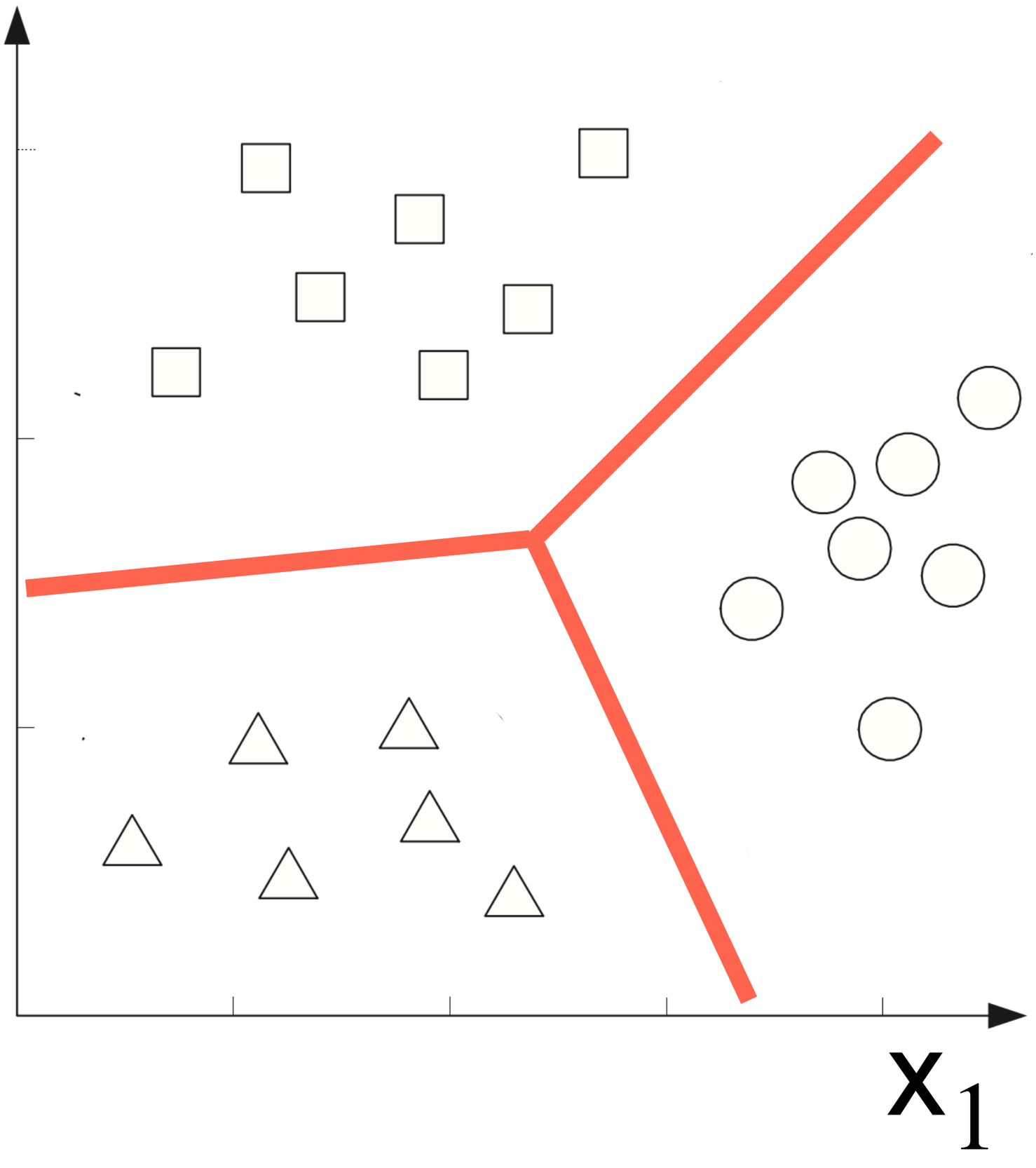
classifier

expert

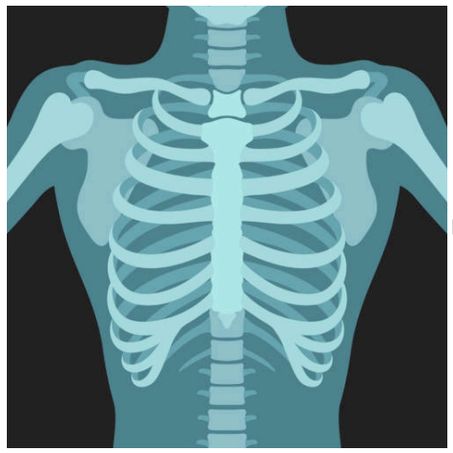
$p(y \mid x)$

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

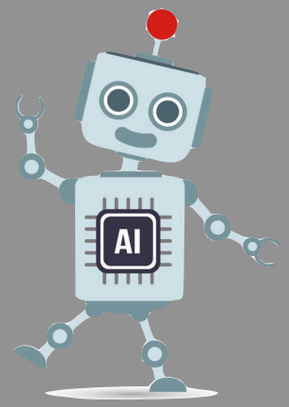




input features



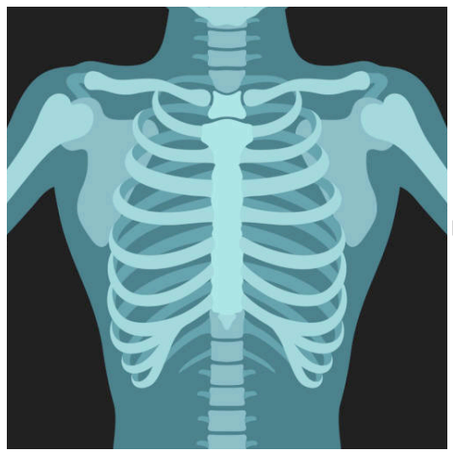
allocation mechanism



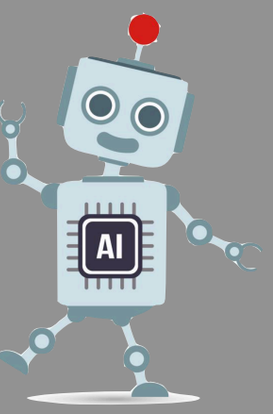
classifier

expert

input features



allocation mechanism



classifier

expert

input features

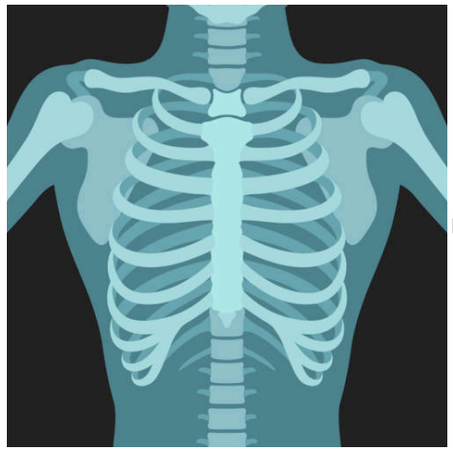
x

(black box)

m

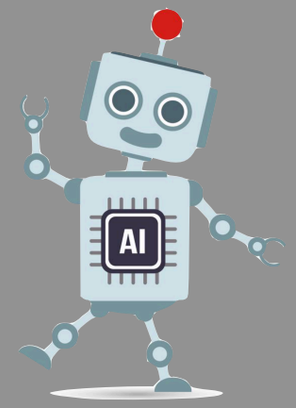
prediction

input features



allocation mechanism

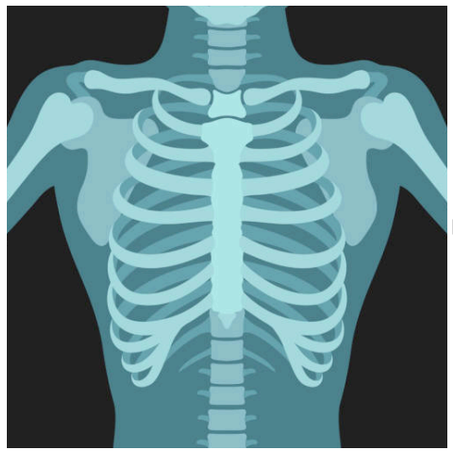
???



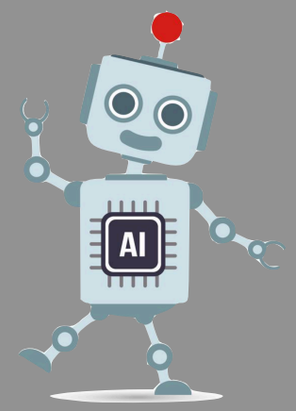
classifier

expert

input features



allocation mechanism



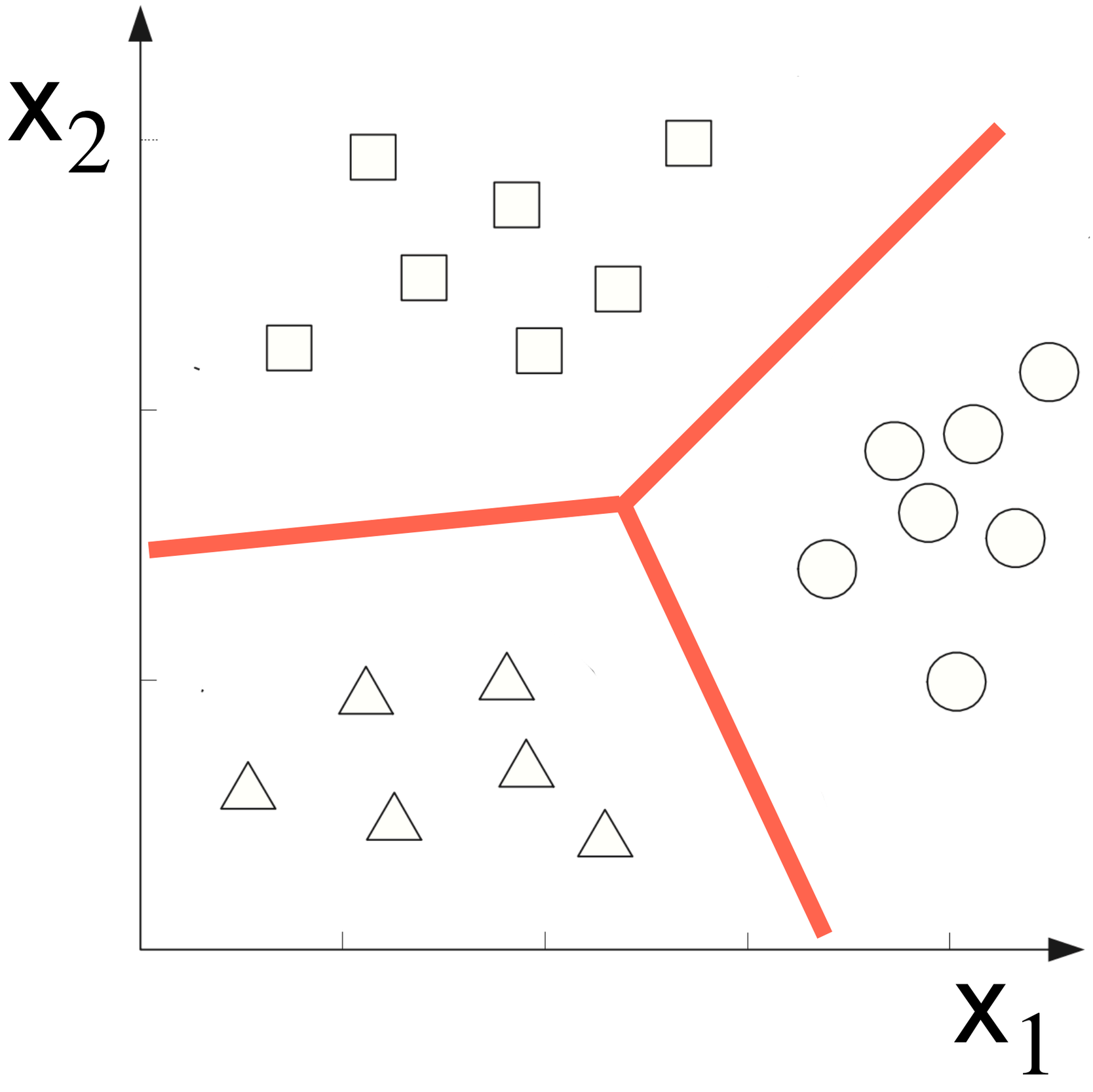
classifier

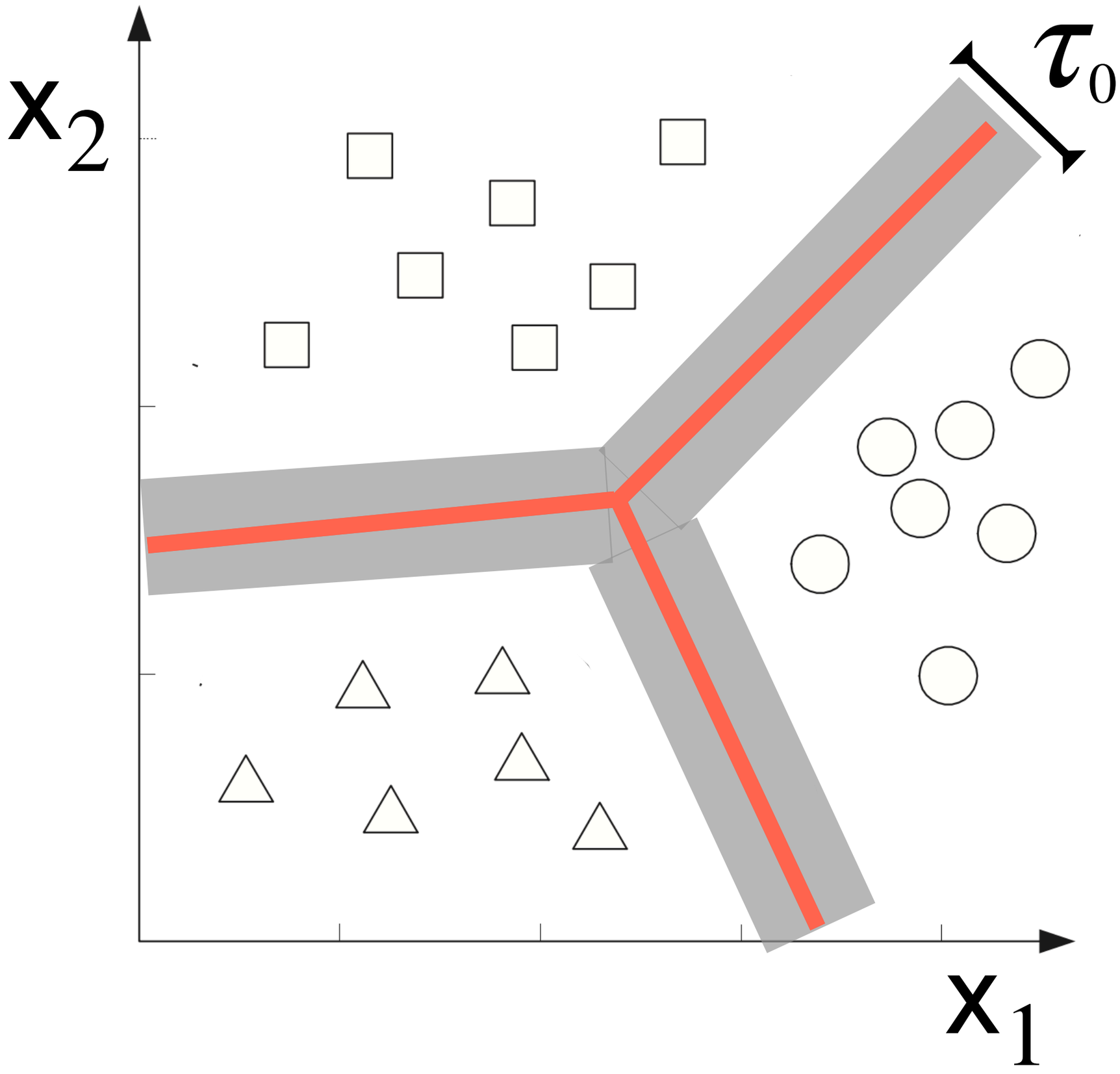
expert

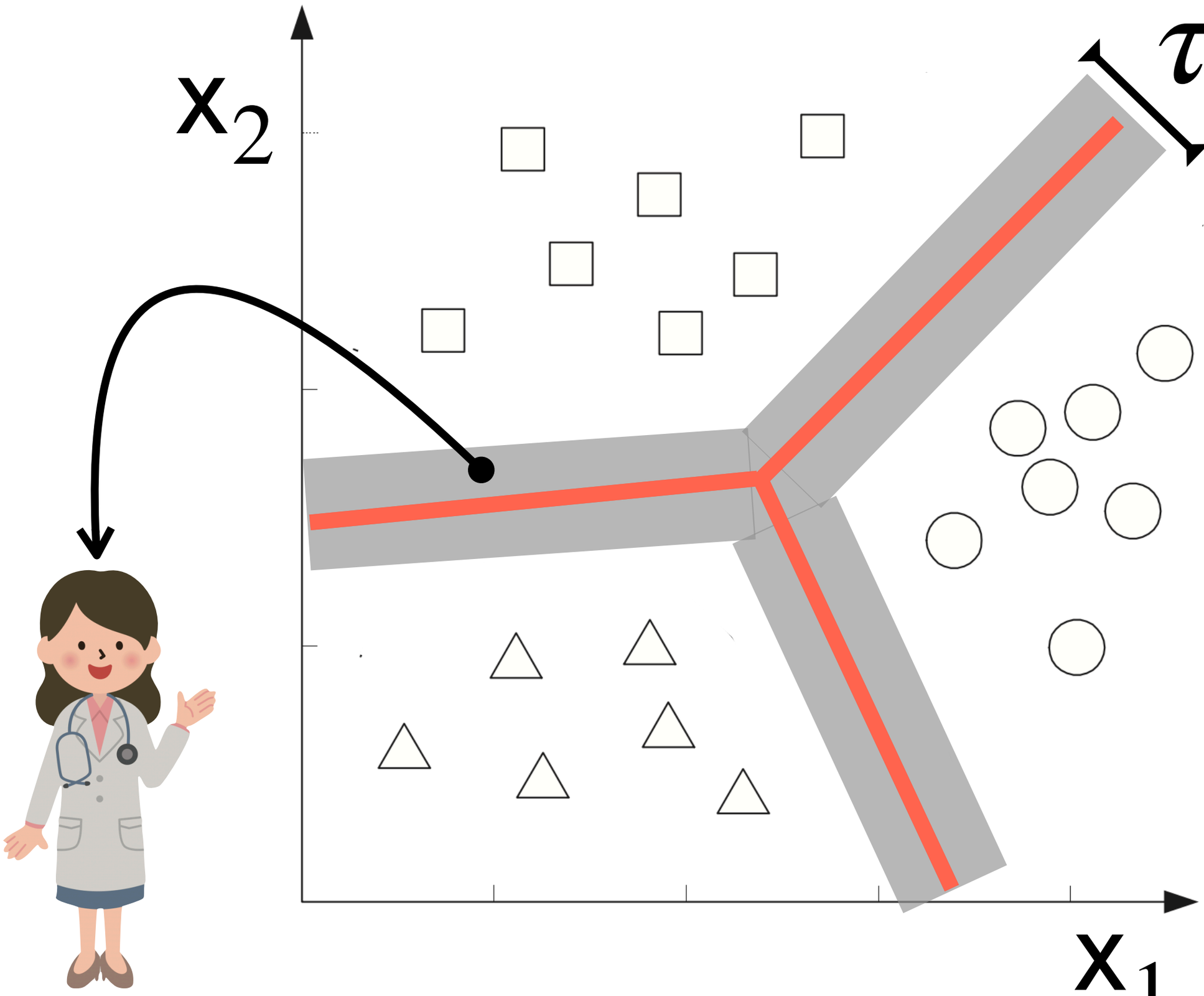
defer to expert if…

$$\max_{y} \; \mathbf{p}(y \mid \mathbf{x}) \;\leq\; \tau_0$$

(constant)

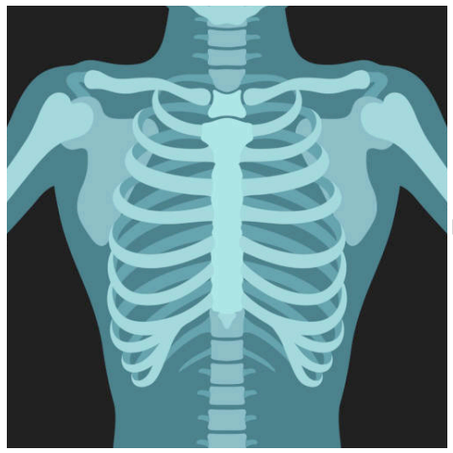

x2
x1


x_2
τ_0
x_1

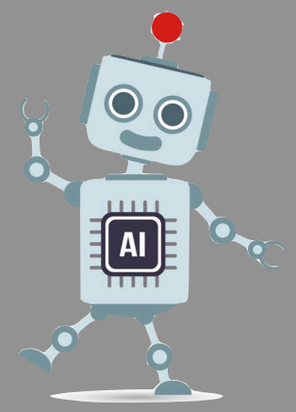


deferred to expert

input features



allocation mechanism



classifier

expert

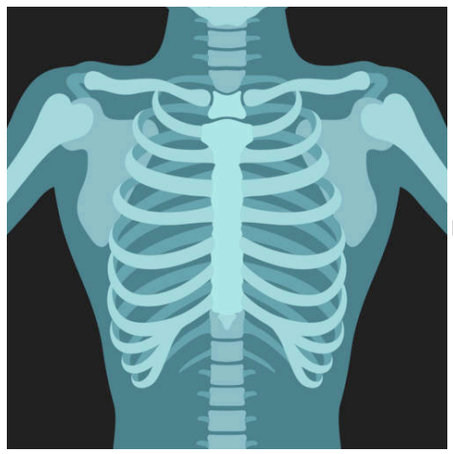
defer to expert if…

$$\max_{y} \; \mathbf{p}(y \mid \mathbf{x}) \leq \tau_0$$

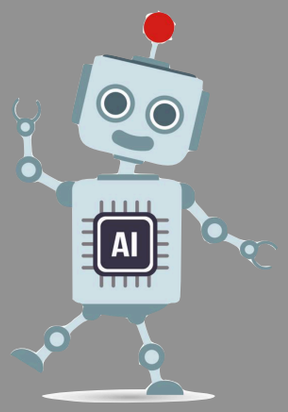
(constant)

problem?

input features



allocation mechanism



classifier

expert

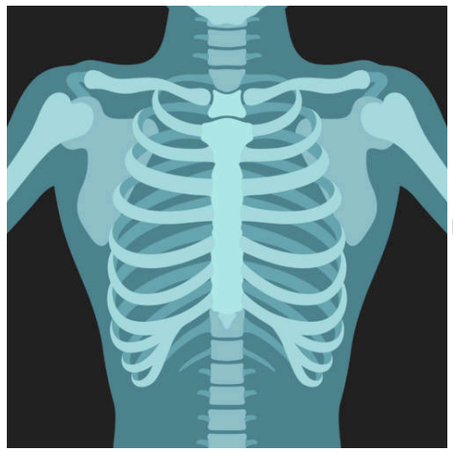
defer to expert if…

$$\max_{y} \; \mathbf{p}(y \mid \mathbf{x}) \leq \tau_0$$

(constant)

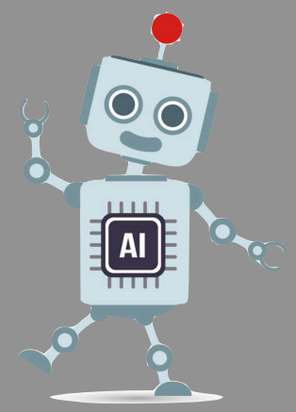
the expert's knowledge is not considered!

input features



allocation mechanism

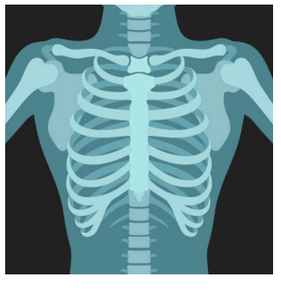
classifier



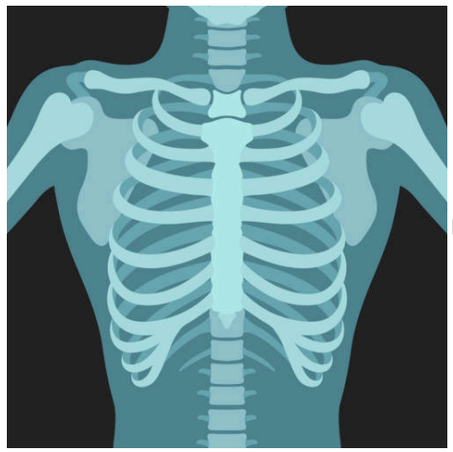
expert

defer to expert if…

$$\max_{y} \; \mathbf{p}(y \mid \mathbf{x}) \;\leq\; \tau\Big(\; , \; \Big)$$

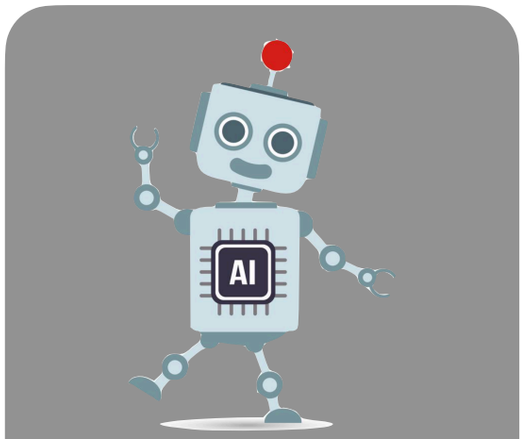


input features



allocation mechanism

L_{0-1}

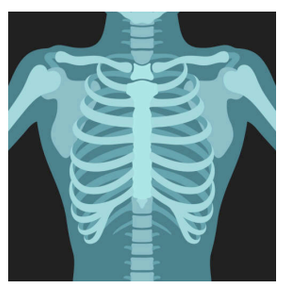


classifier

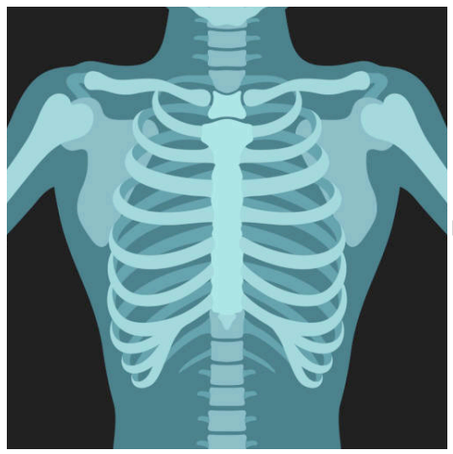
expert

defer to expert if…

$$\max_{y} \; \mathrm{p}(y \mid \mathrm{x}) \;\leq\; \tau\Big(\; , \; \Big)$$



input features



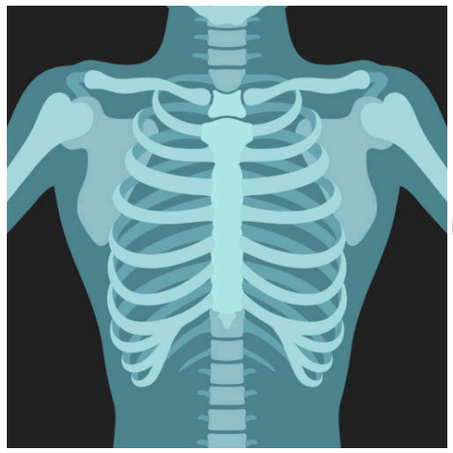
allocation mechanism

L_{0-1}

defer to expert if…

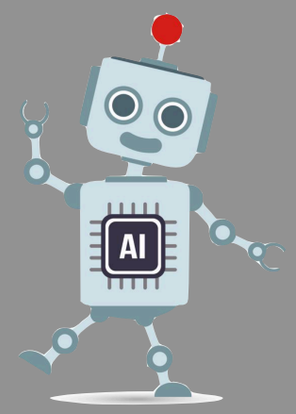
$$\max_{y} \; \mathsf{p}(y \mid \mathbf{x}) \;\leq\; \tau\Big(\; , \; \Big)$$

input features



allocation mechanism

L_{0-1}



classifier

L_{0-1}

expert

Bayes optimal deferral rule:

$$\max_{y} \ \mathbb{P}(y \mid \mathbf{x}) \ \leq \ \mathbb{P}(\mathsf{m} = \mathsf{y} \mid \mathbf{x})$$

probability that the expert is correct

[Mozannar & Sontag, 2020]

softmax implementation [Mozannar & Sontag, 2020]

training data

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

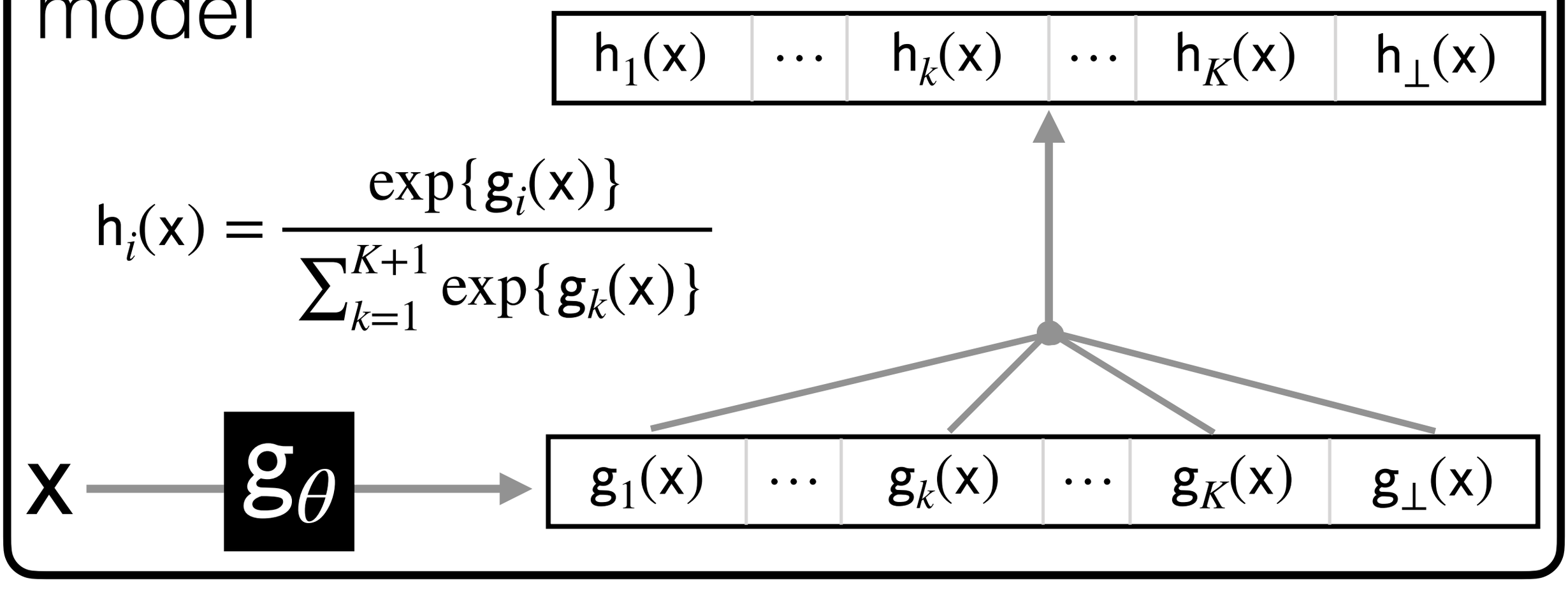
softmax implementation [Mozannar & Sontag, 2020]

training data

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

model

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



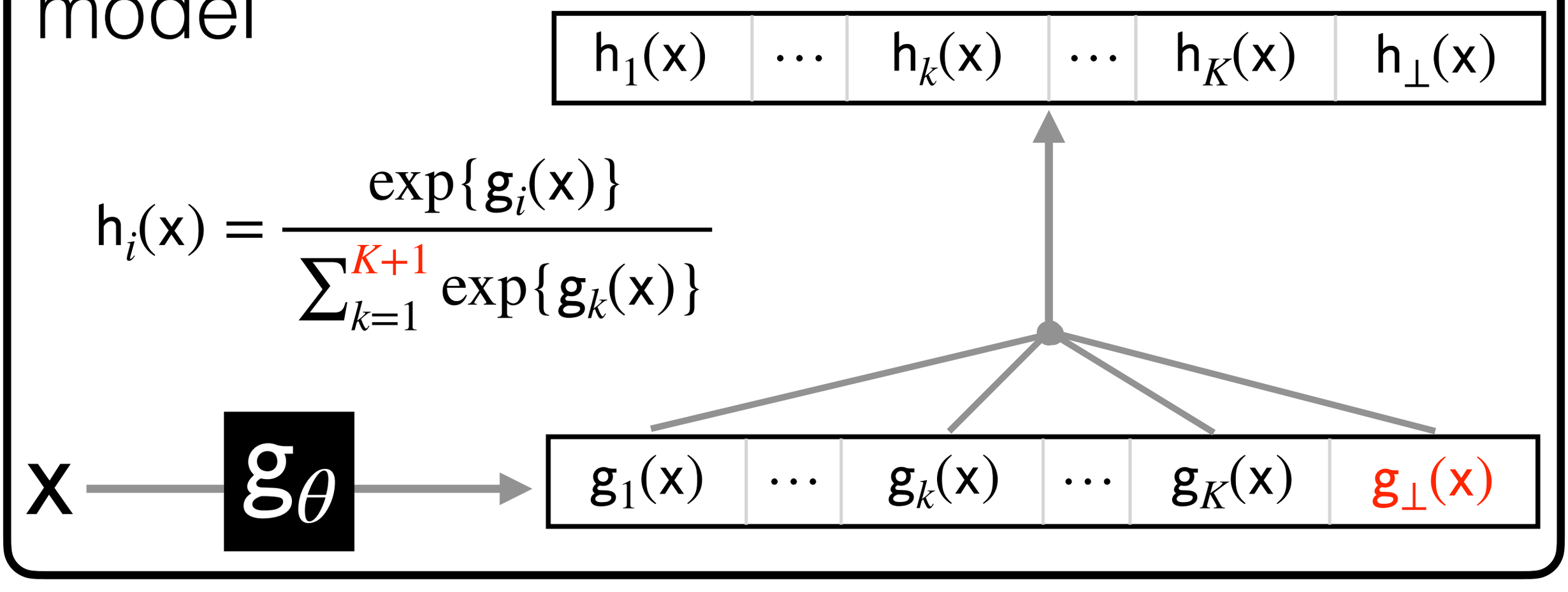
softmax implementation [Mozannar & Sontag, 2020]

training data

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

model

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



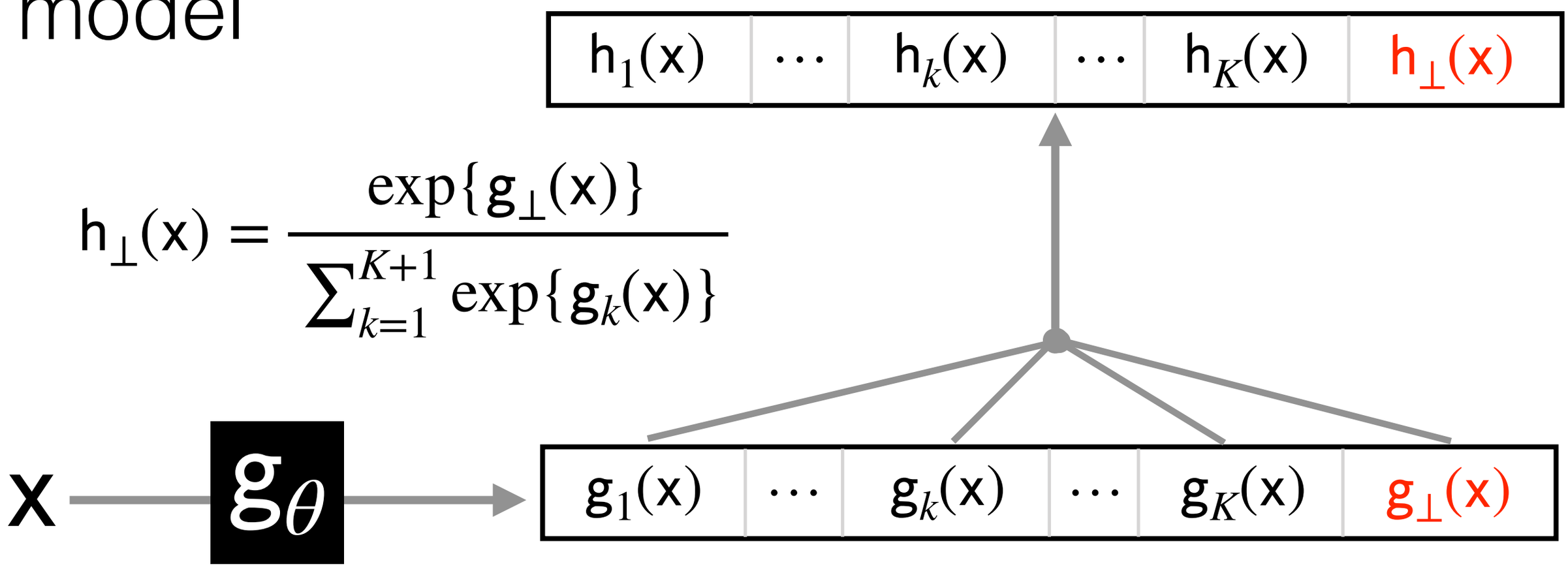
softmax implementation [Mozannar & Sontag, 2020]

training data

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

model

$$\mathsf{h}_\perp(\mathsf{x}) = \frac{\exp\{\mathsf{g}_\perp(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$



softmax implementation [Mozannar & Sontag, 2020]

training data

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

model

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

softmax implementation [Mozannar & Sontag, 2020]

training data

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

model

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

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation [Mozannar & Sontag, 2020]

training data

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

model

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

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation [Mozannar & Sontag, 2020]

training data

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

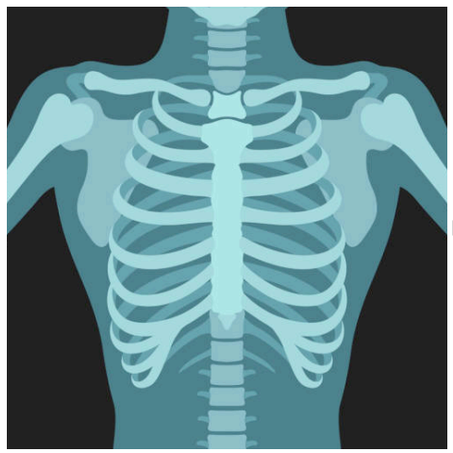
model

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

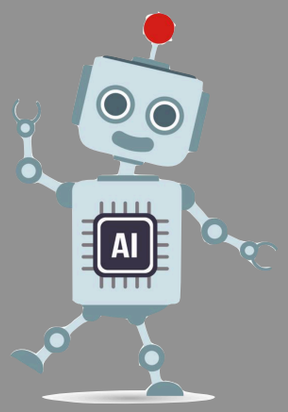
loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

input features



allocation mechanism



classifier

expert

defer to expert if…

$$\max_{y \in [1,K]} \; \mathsf{h}_y(\mathsf{x}) \;\leq\; \mathsf{h}_\perp(\mathsf{x})$$

- ⊗ single expert
  - ⊗ softmax surrogate loss
  - ⊗ improving calibration via one-vs-all
- ⊗ multiple experts
  - ⊗ surrogate losses
  - ⊗ conformal sets of experts
- ⊗ population of experts
  - ⊗ surrogate losses
  - ⊗ meta-learning a rejector

How well can the softmax-based system estimate expert correctness?

$$\hat{p}(m = y \mid x) \overset{?}{\approx} \mathbb{P}(m = y \mid x)$$

- ⊗ optimal allocation
- ⊗ transparency
- ⊗ detecting distribution shift (in the expert)

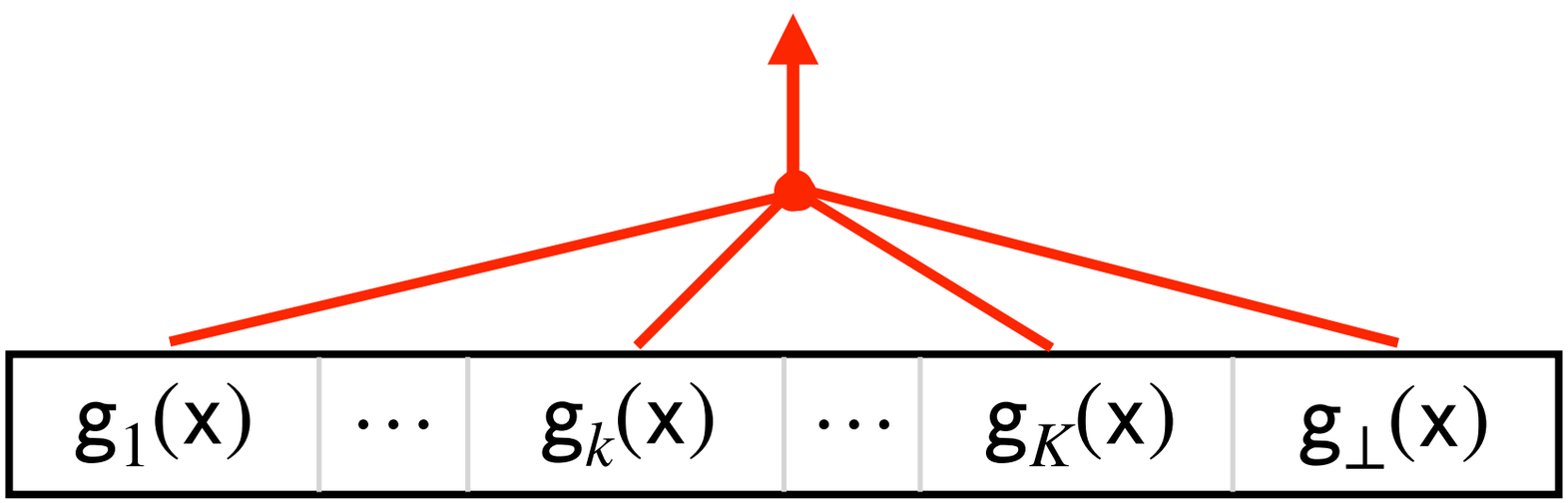
How well can the softmax-based system estimate expert correctness?

$$\hat{p}(m = y \mid x) \not\approx \mathbb{P}(m = y \mid x)$$

degenerate parameterization

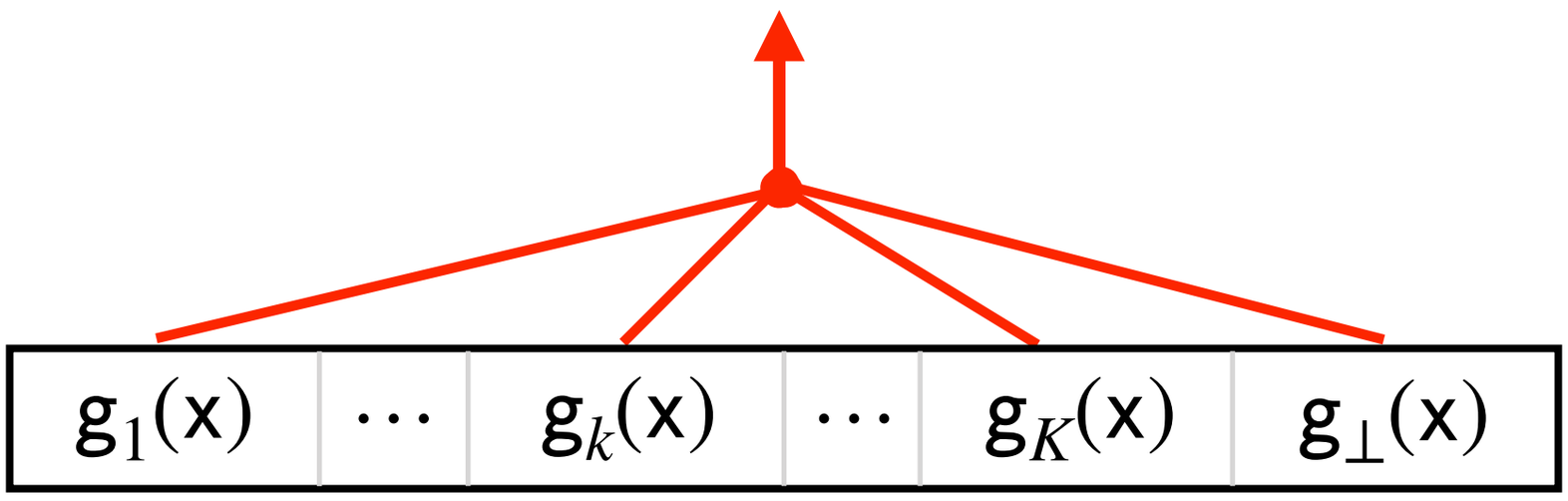
[Proposition 3.1]

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

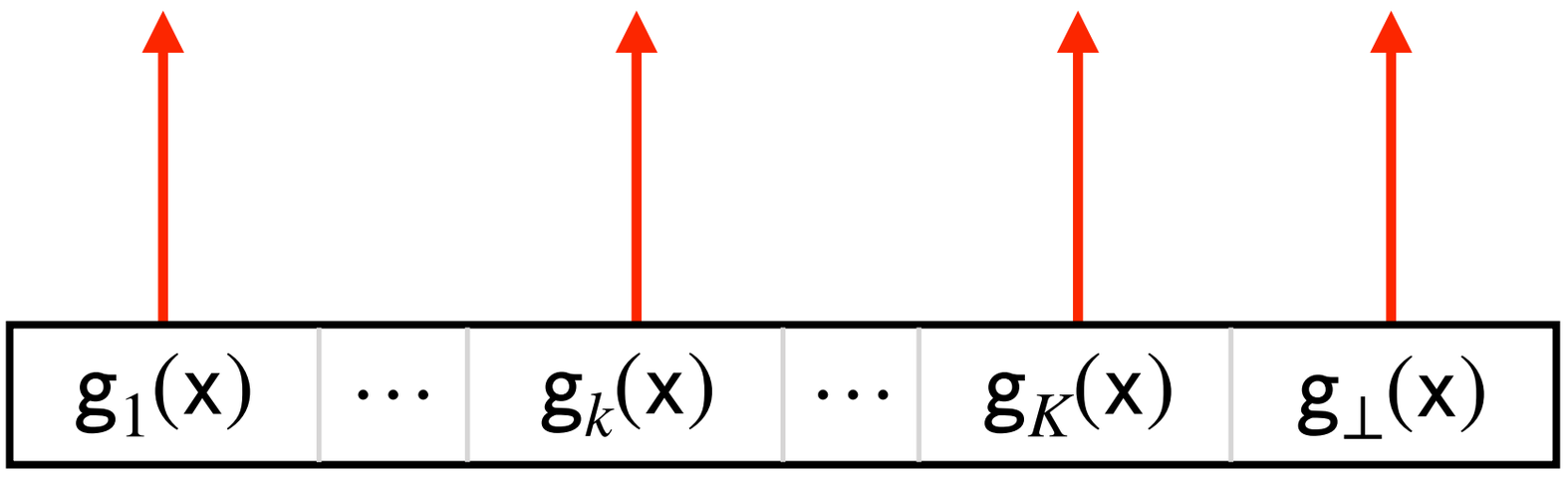


our solution: switch to a **one-vs-all** parameterization

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

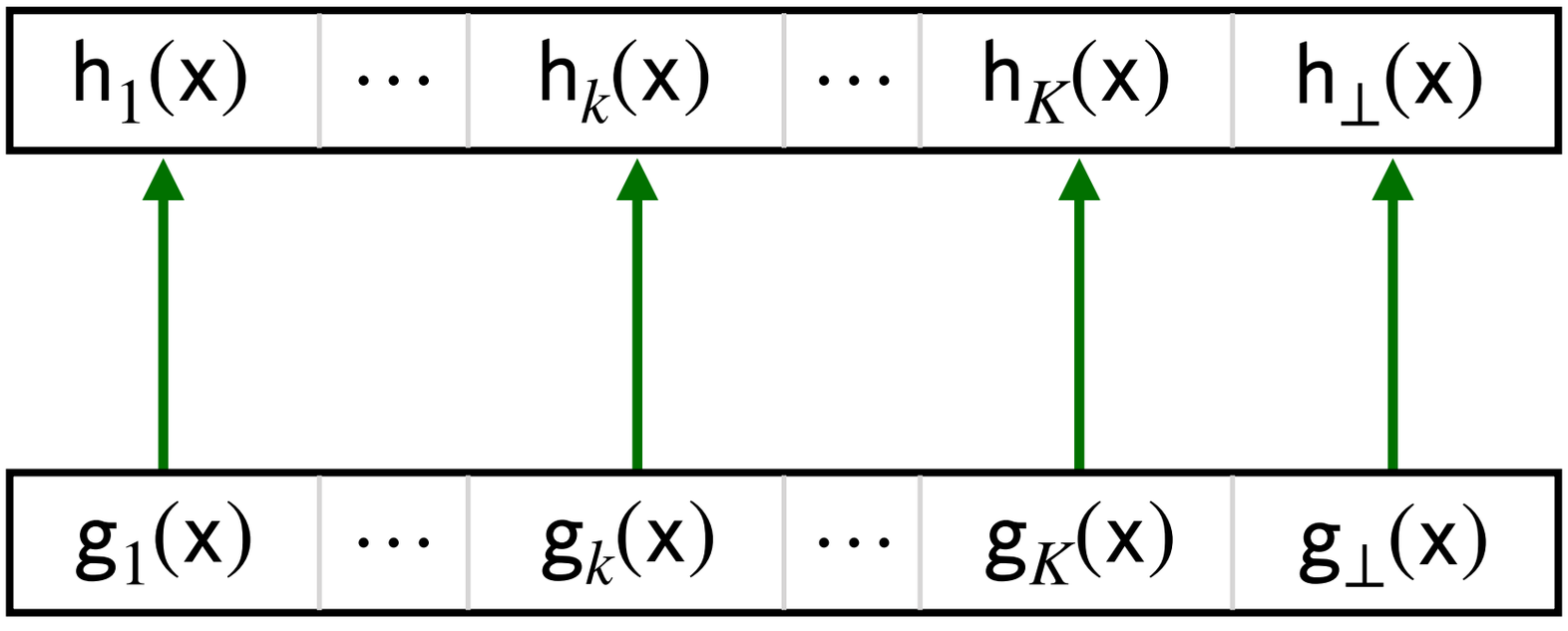


our solution: switch to a
one-vs-all parameterization



our solution: switch to a one-vs-all parameterization

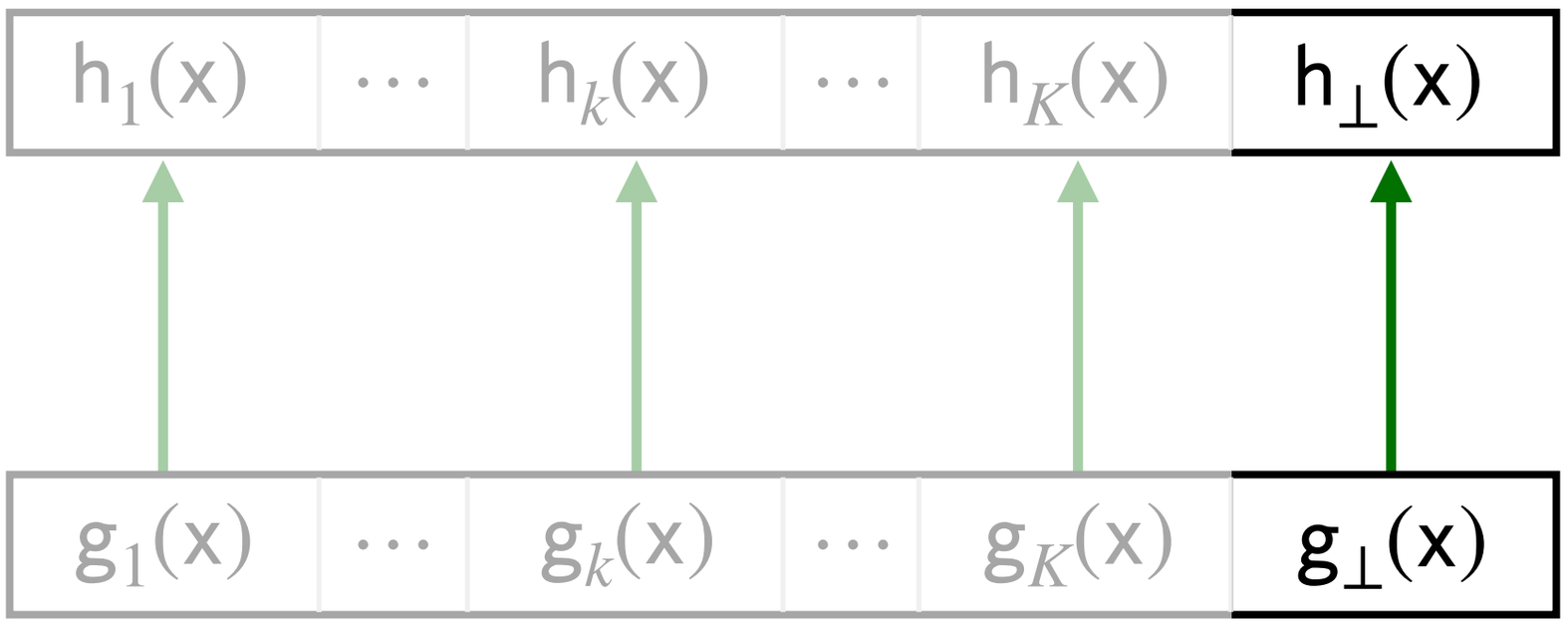
$$h_i(x) = \frac{1}{1 + \exp\{-g_i(x)\}}$$



our solution: switch to a one-vs-all parameterization

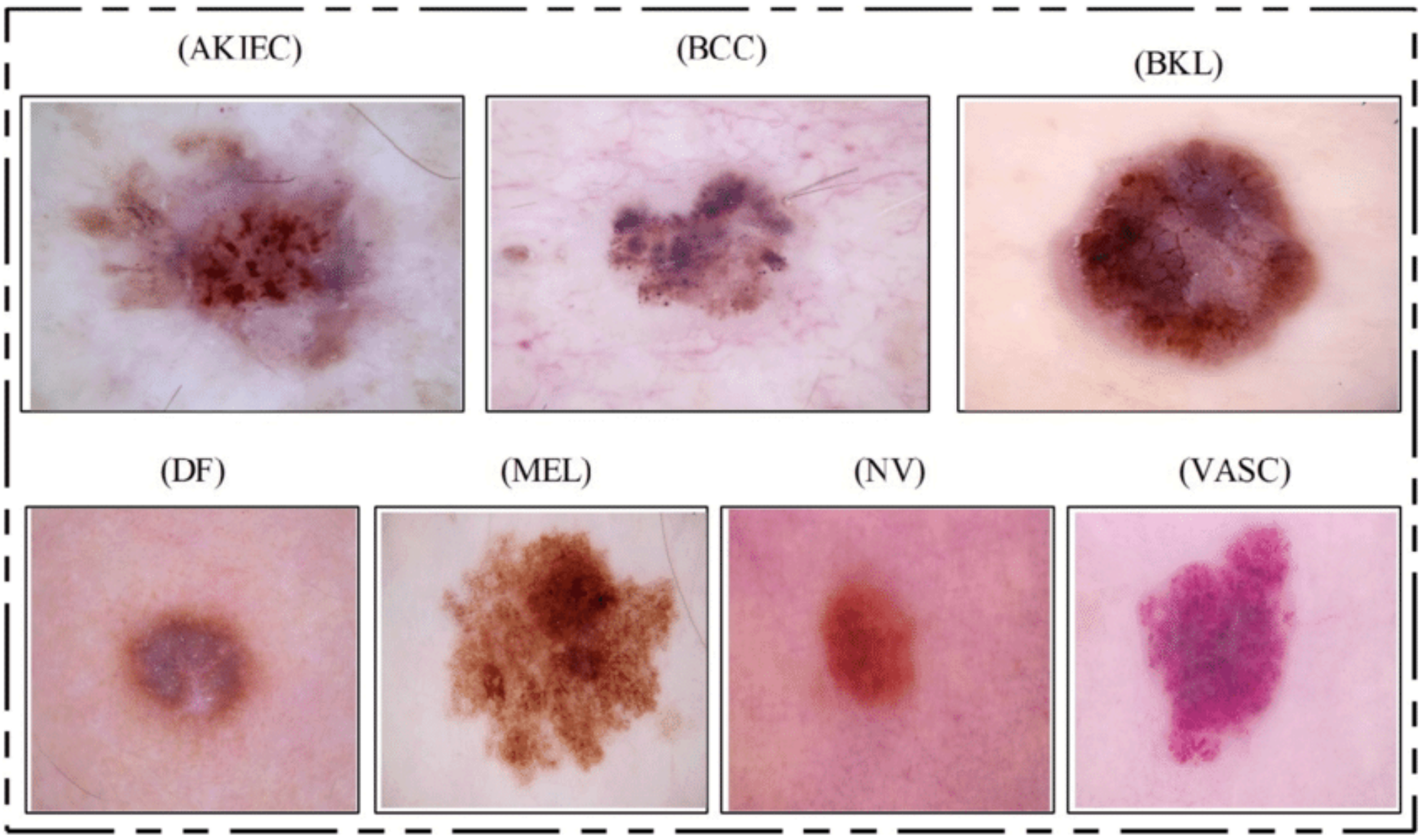
[Theorem 4.1]

$$h^*_\perp(x) = \mathbb{P}(m = y \mid x)$$



estimating expert correctness

skin lesion diagnosis



estimating expert correctness

$\hat{p}$	distance: $\hat{p}$ vs $\mathbb{P}$
softmax	26.7 ± 1.8
one-vs-all (ours)	8.0 ± 1.0 

But does one-vs-all
result in **more accurate** models?

one-vs-all (ours)

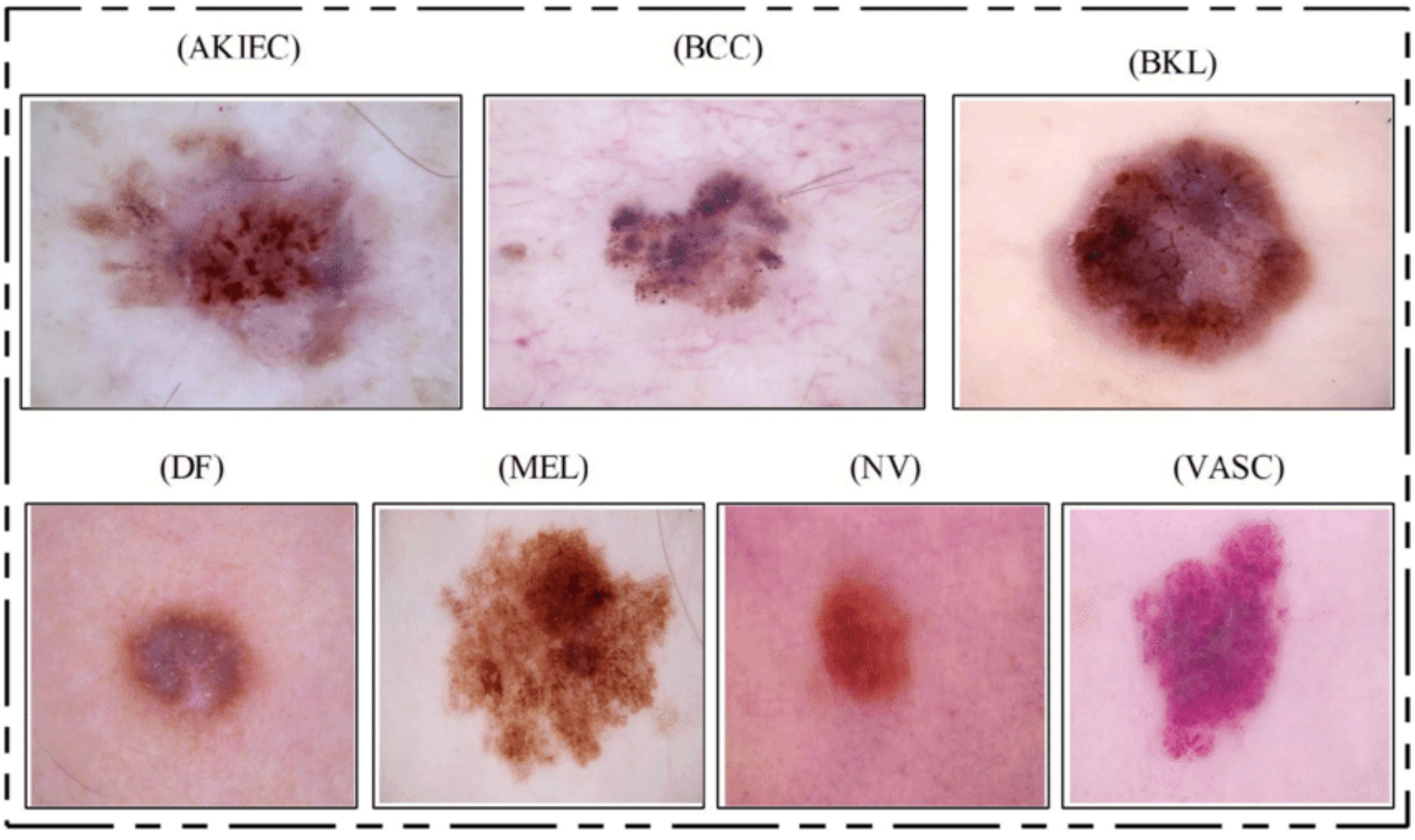
softmax

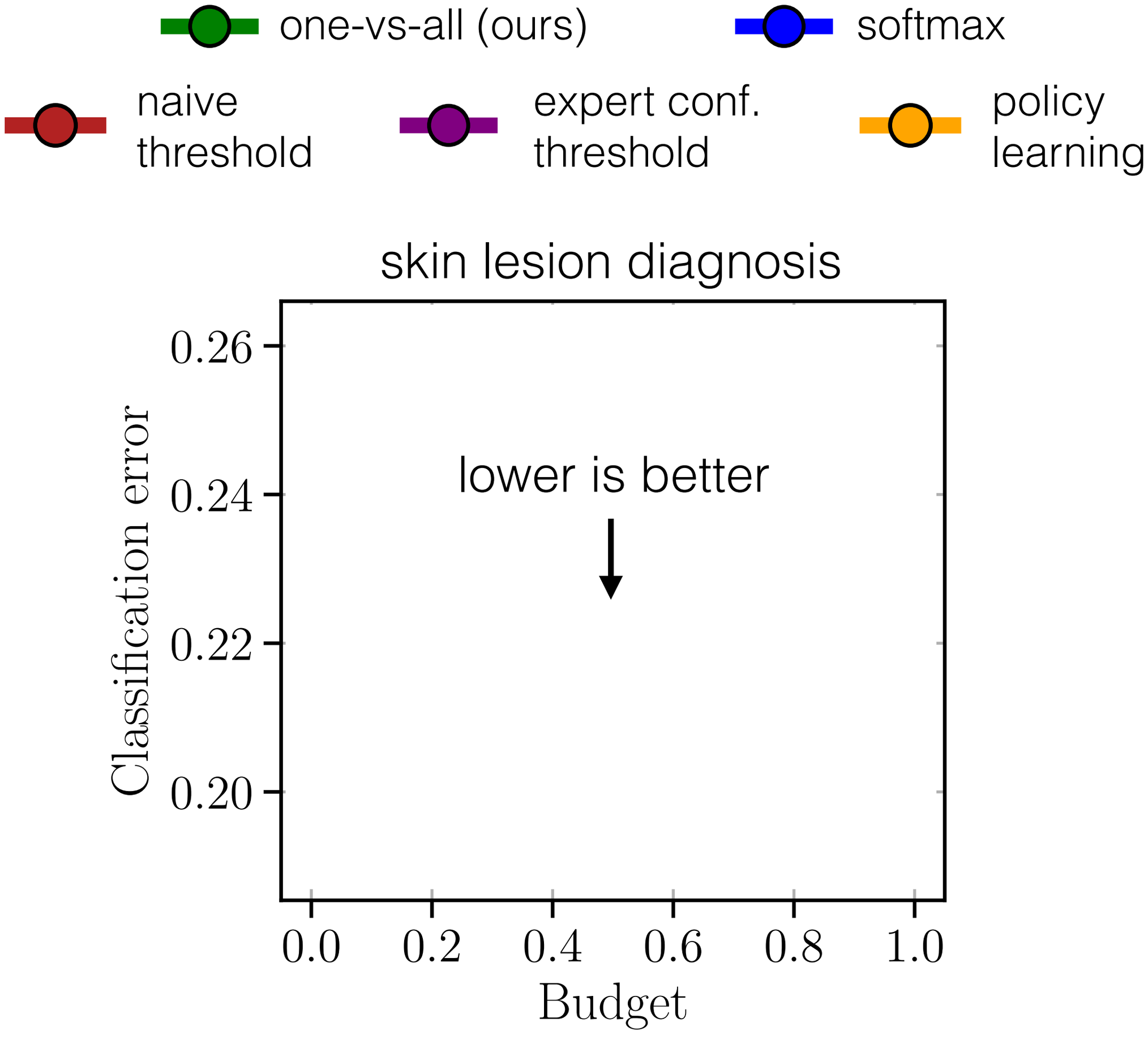
naive threshold

expert conf. threshold

policy learning

skin lesion diagnosis




one-vs-all (ours)
softmax
naive threshold
expert conf. threshold
policy learning
skin lesion diagnosis
lower is better
Classification error
0.26
0.24
0.22
0.20
0.0
0.2
0.4
0.6
0.8
1.0
Budget

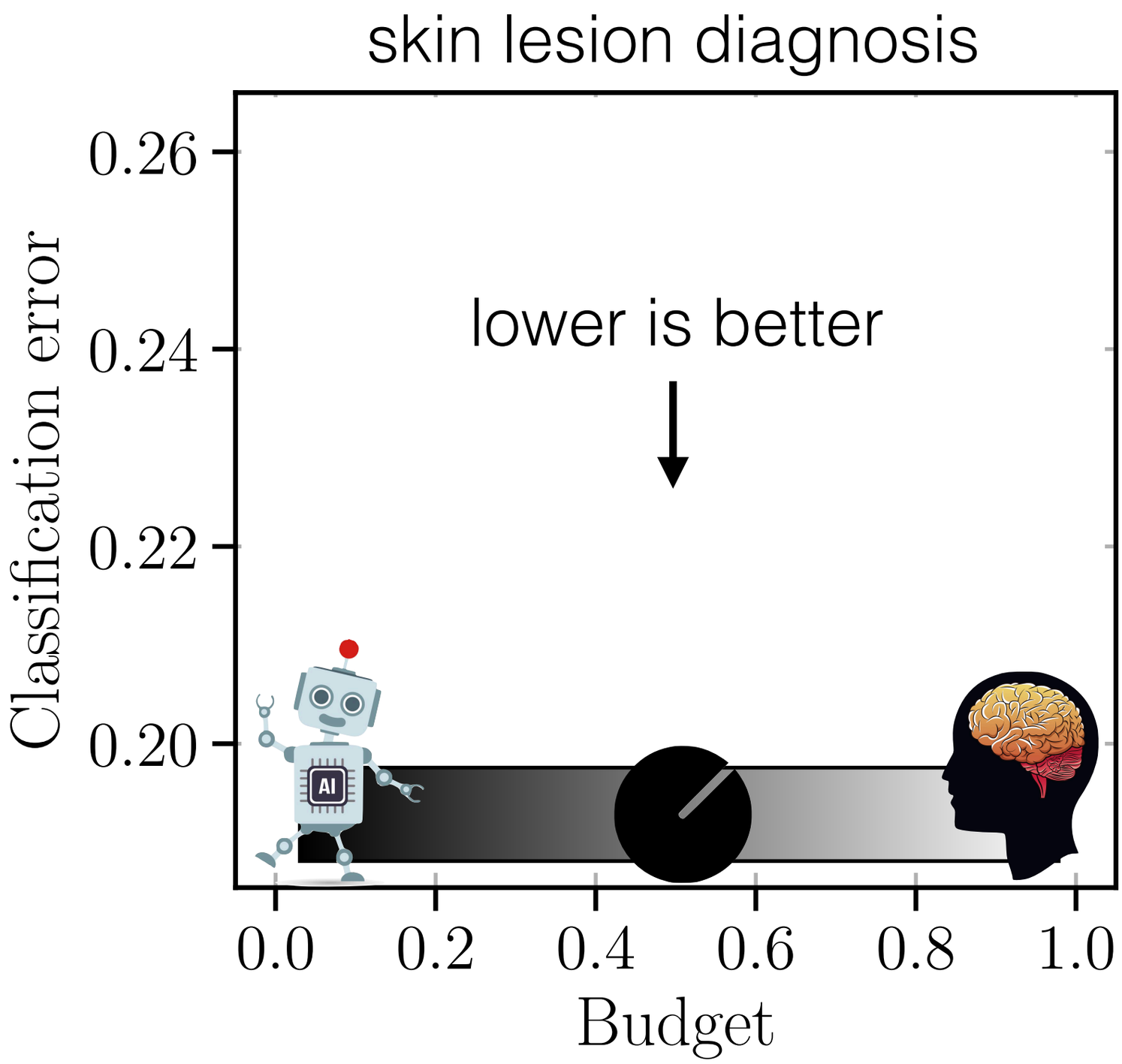
one-vs-all (ours)

softmax

naive threshold

expert conf. threshold

policy learning



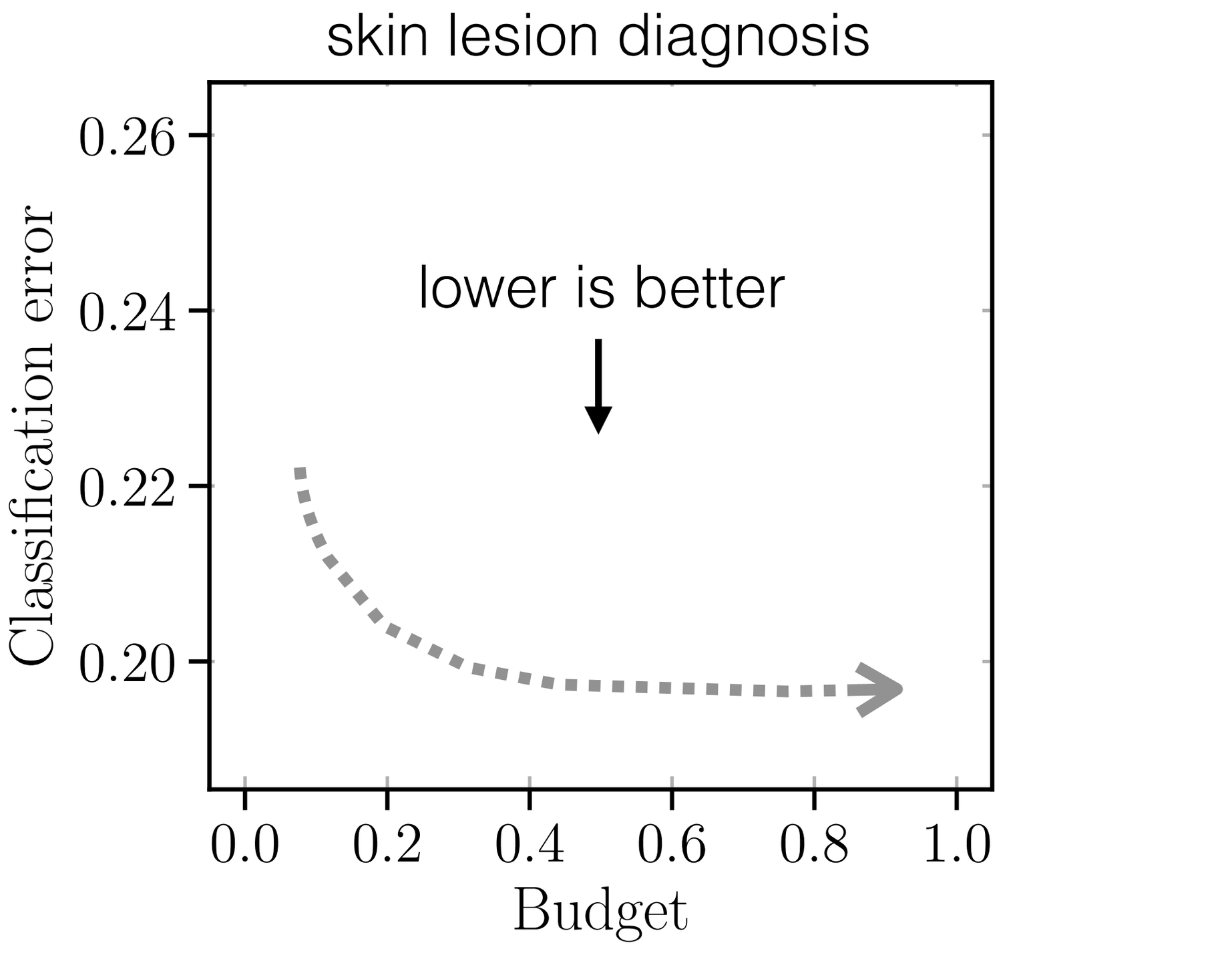
one-vs-all (ours)

softmax

naive threshold

expert conf. threshold

policy learning



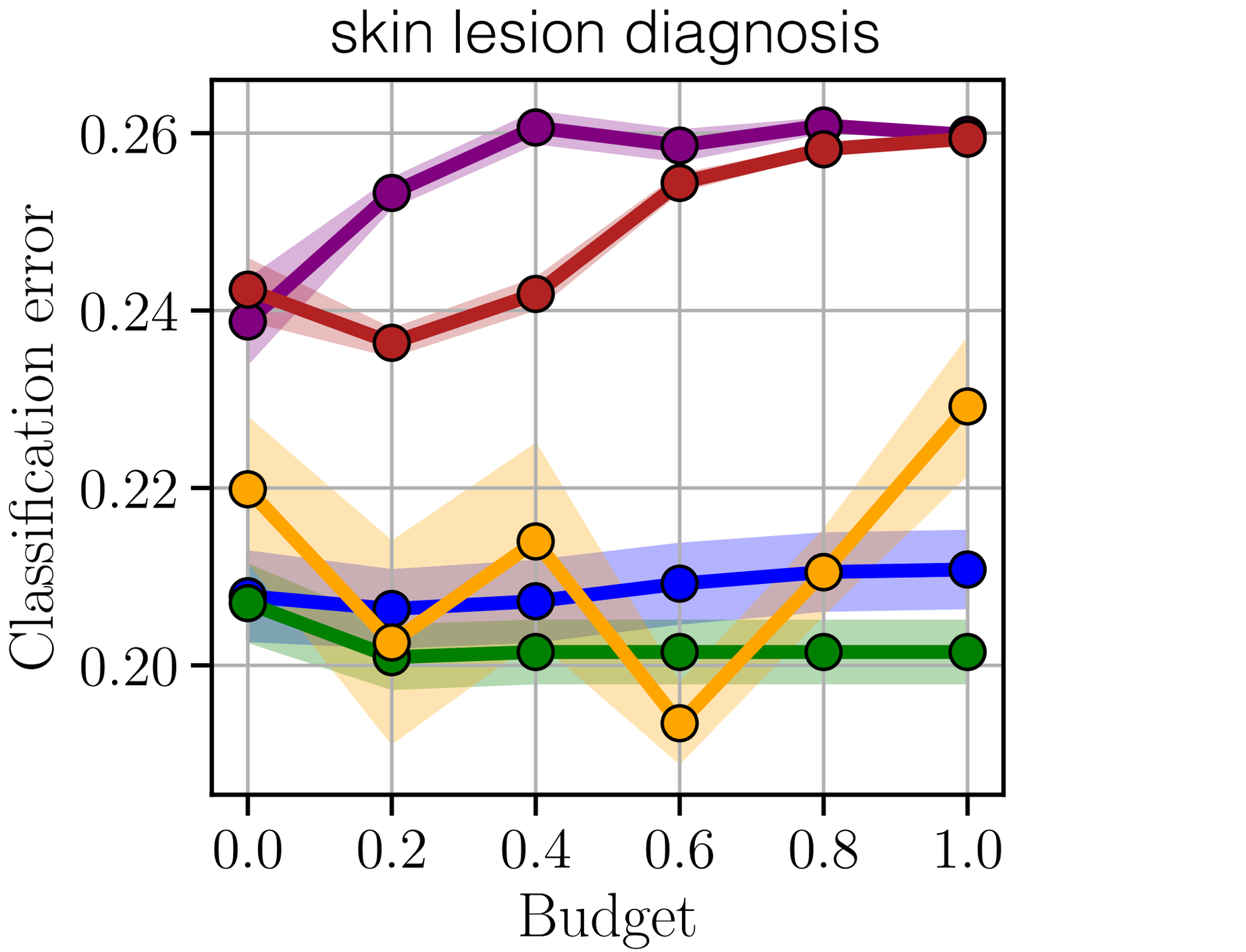
one-vs-all (ours)

softmax

naive threshold

expert conf. threshold

policy learning



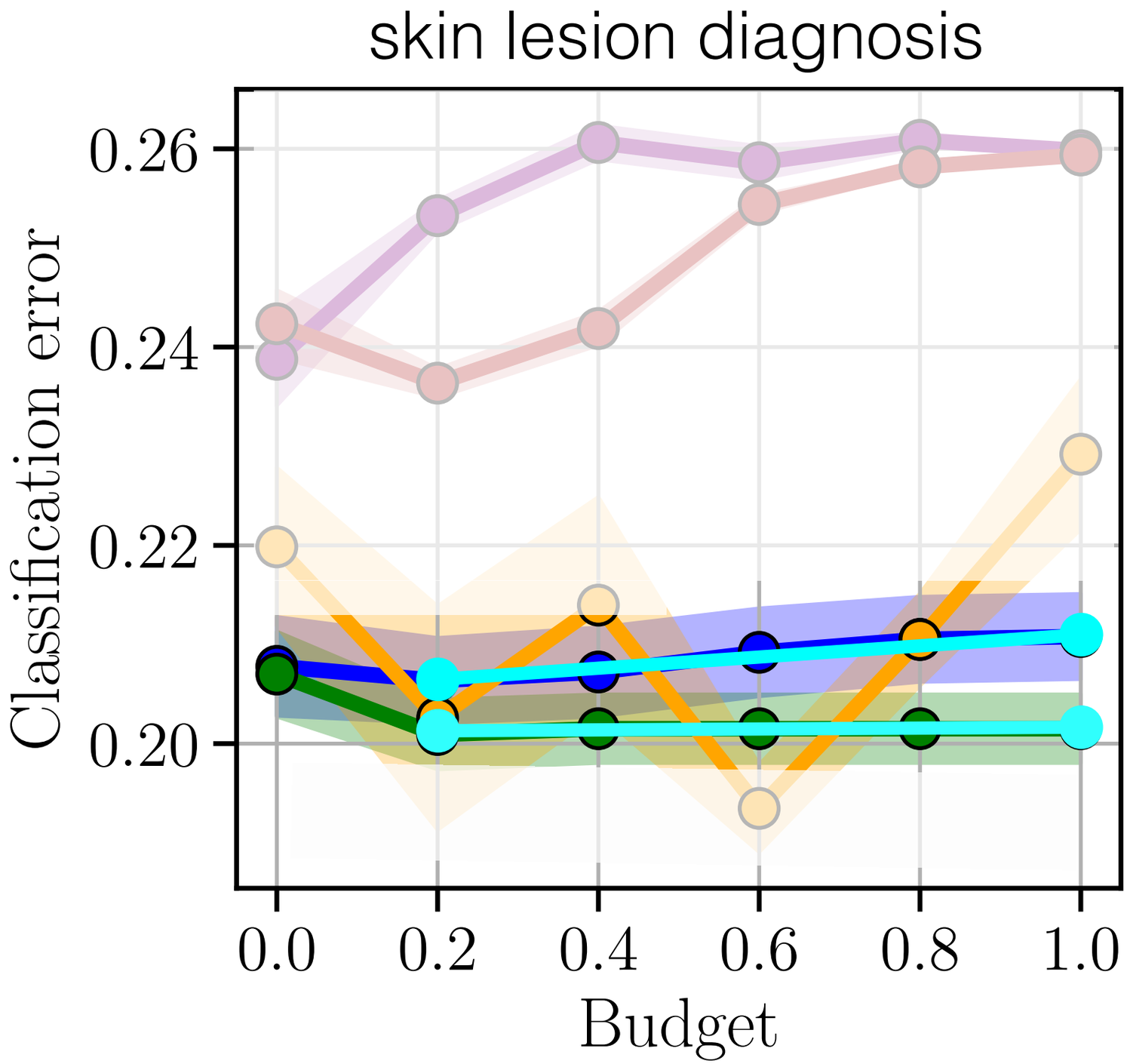
one-vs-all (ours)

softmax

naive threshold

expert conf. threshold

policy learning

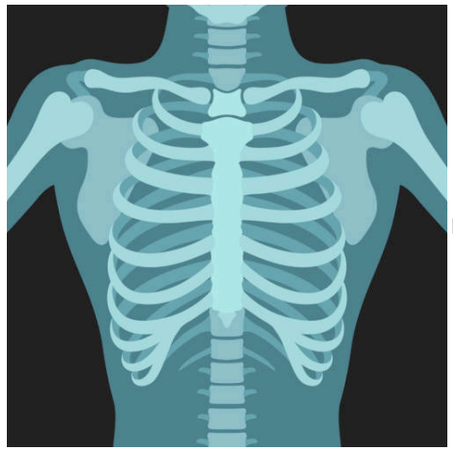


- single expert
  - softmax surrogate loss
  - improving calibration via one-vs-all

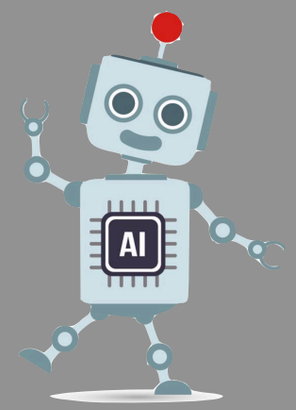
- multiple experts
  - surrogate losses
  - conformal sets of experts

- population of experts
  - surrogate losses
  - meta-learning a rejector

input features



allocation mechanism



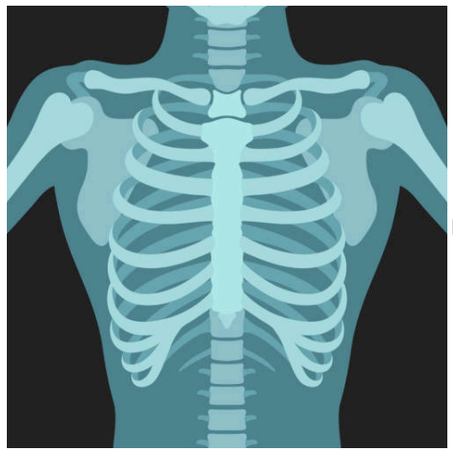
classifier

expert #1

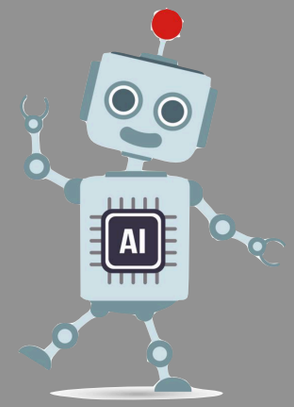
expert #3

expert #2

input features



allocation mechanism



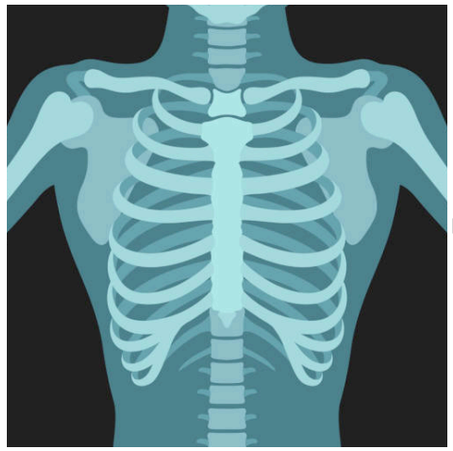
classifier

expert #1

expert #3

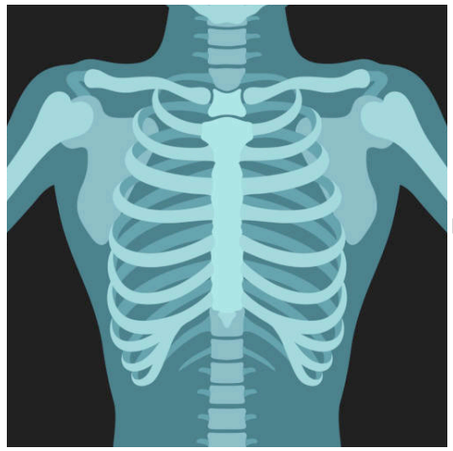
expert #2

input features



allocation mechanism

input features



allocation mechanism

?

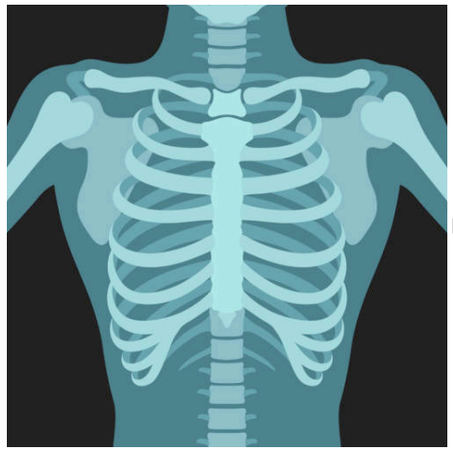
classifier

expert #1

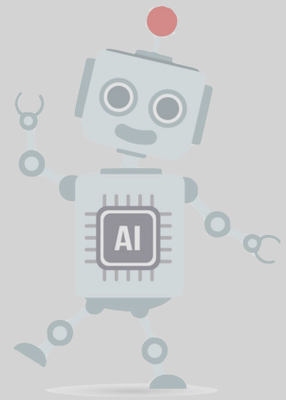
expert #3

expert #2

input features



allocation mechanism



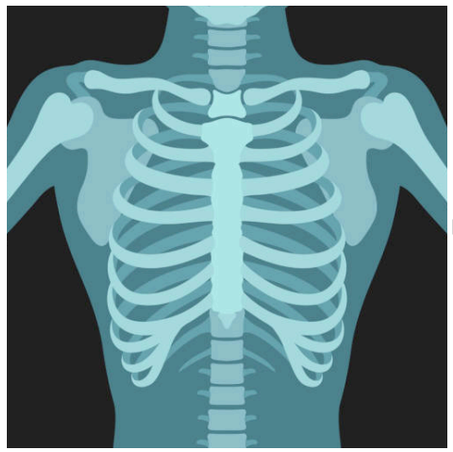
classifier

expert #1

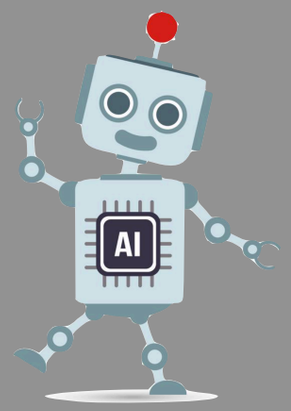
expert #3

expert #2

input features



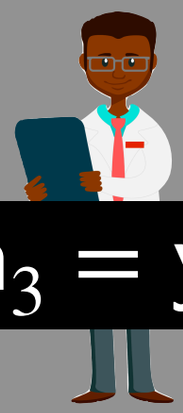
allocation mechanism



classifier

$\mathbb{P}(m_1 = y \mid x)$

expert #1



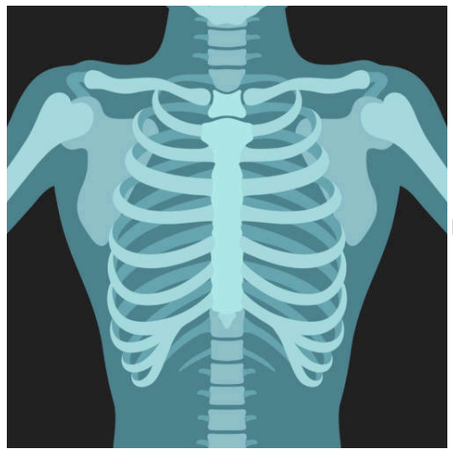
$\mathbb{P}(m_3 = y \mid x)$

expert #3

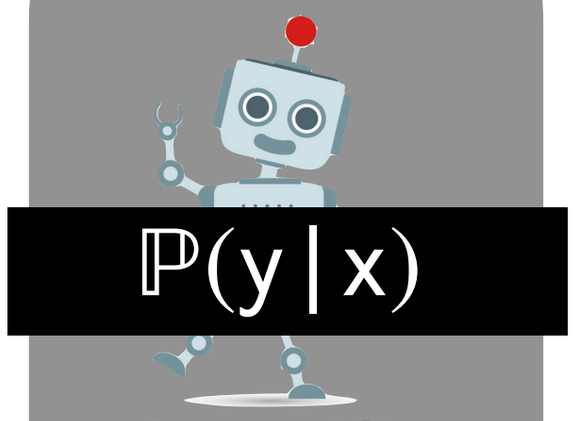
$\mathbb{P}(m_2 = y \mid x)$

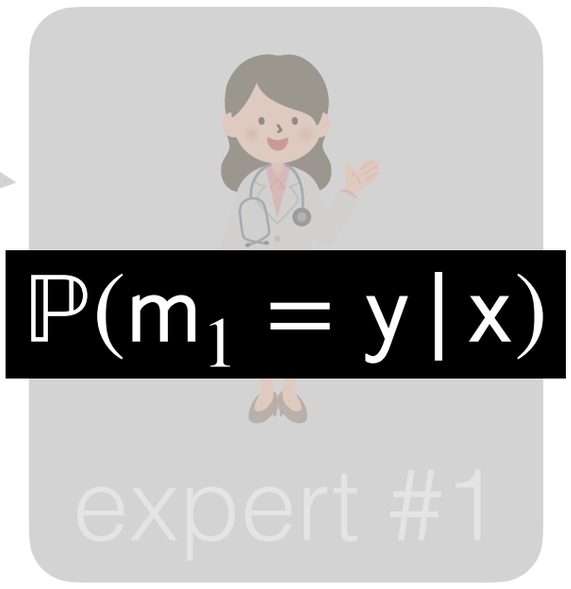
expert #2

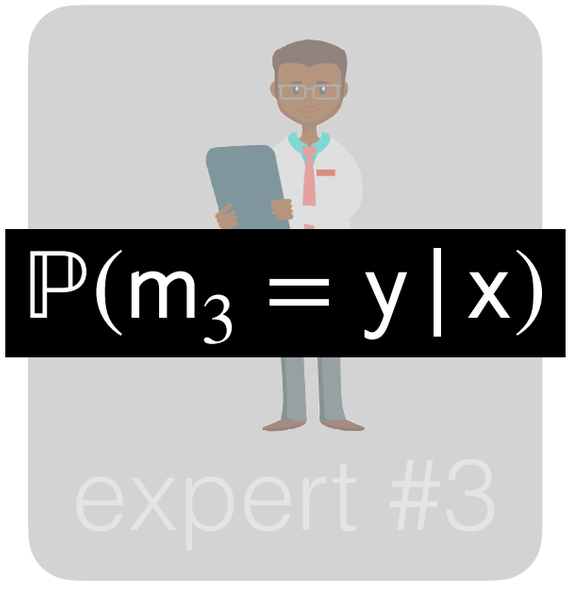
input features

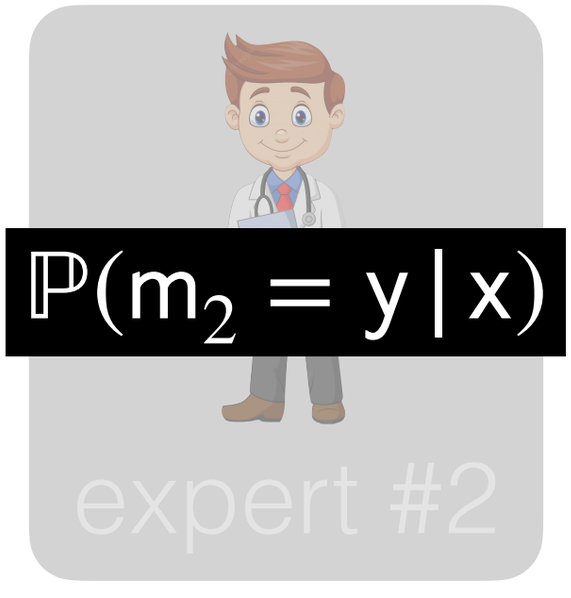


allocation mechanism





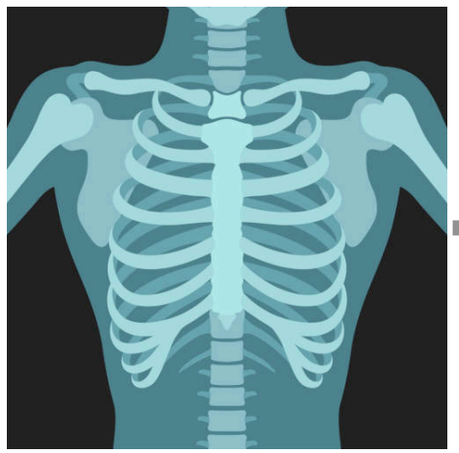




use classifier if…

$$\max_y \ \mathbb{P}(y \mid x) > \mathbb{P}(m_j = y \mid x), \ \forall j$$

input features



allocation mechanism

classifier

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

expert #1

$\mathbb{P}(m_1 = y \mid x)$

expert #3

$\mathbb{P}(m_3 = y \mid x)$

expert #2

$\mathbb{P}(m_2 = y \mid x)$

else, pick best expert:

$$\arg\max_j \ \mathbb{P}\left(m_j = y \mid x\right)$$

multi-expert implementation

training data

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

multi-expert implementation

training data

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

model

$\mathsf{h}_1(\mathsf{x})$	$\cdots$	$\mathsf{h}_k(\mathsf{x})$	$\cdots$	$\mathsf{h}_K(\mathsf{x})$	$\mathsf{h}_{\perp,1}(\mathsf{x})$	$\cdots$	$\mathsf{h}_{\perp,J}(\mathsf{x})$

K classes

J experts

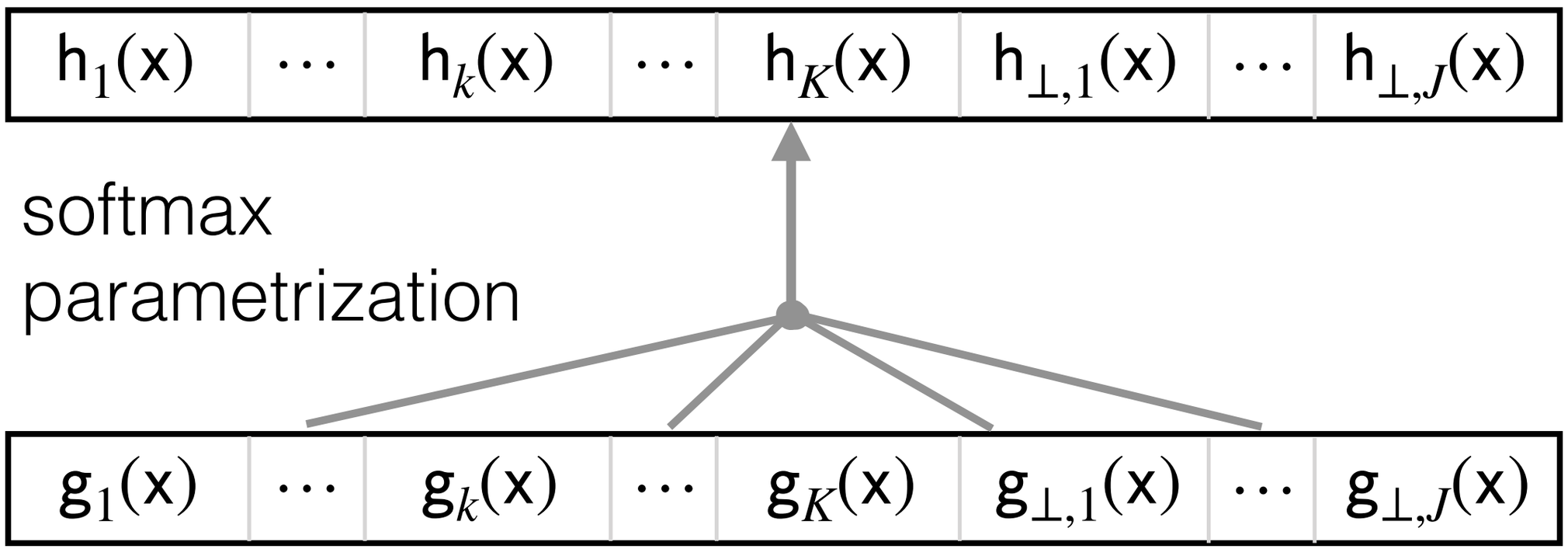
$\mathsf{g}_1(\mathsf{x})$	$\cdots$	$\mathsf{g}_k(\mathsf{x})$	$\cdots$	$\mathsf{g}_K(\mathsf{x})$	$\mathsf{g}_{\perp,1}(\mathsf{x})$	$\cdots$	$\mathsf{g}_{\perp,J}(\mathsf{x})$

multi-expert implementation

training data

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

model

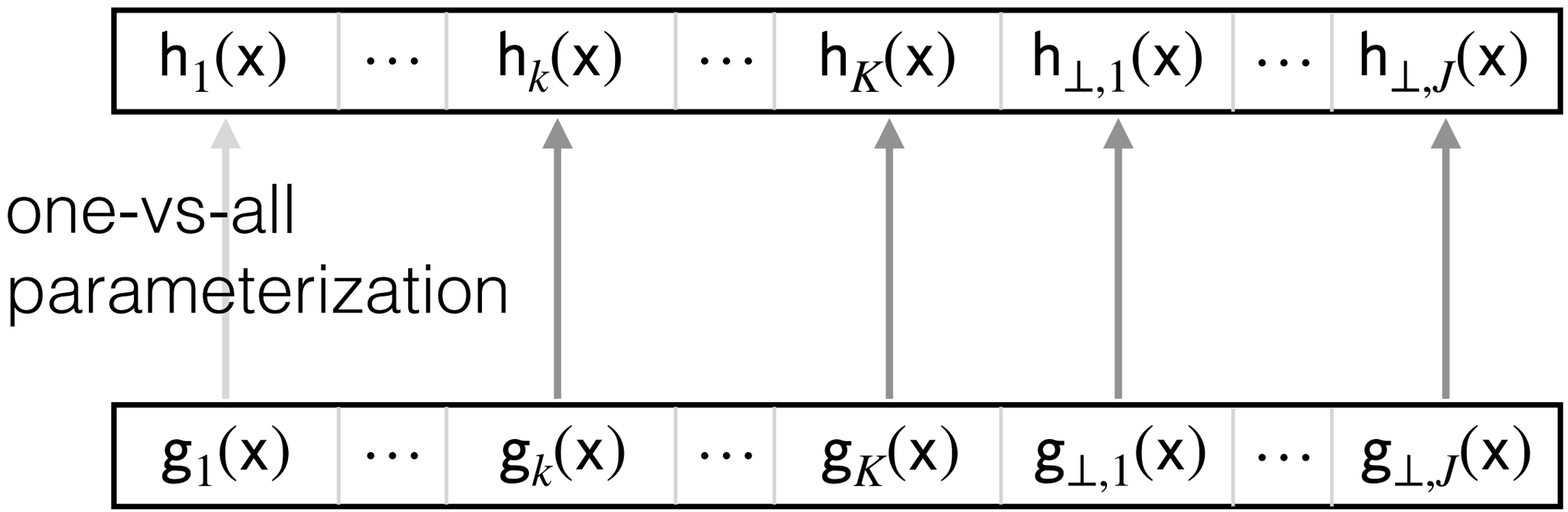


multi-expert implementation

training data

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

model



multi-expert implementation

training data

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

model

- ⊗ softmax and one-vs-all variants
- ⊗ both consistent w.r.t. 0-1 loss

multi-expert implementation

training data

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

model

- ⊗ softmax and one-vs-all variants
- ⊗ both consistent w.r.t. 0-1 loss

softmax loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) - \sum_j \mathbb{I}\left[\mathsf{y} = \mathsf{m}_j\right] \cdot \log \mathsf{h}_{\perp,j}(\mathsf{x})$$

multi-expert implementation

training data

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

model

- ⊗ softmax and one-vs-all variants
- ⊗ both consistent w.r.t. 0-1 loss

softmax loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) - \sum_j \mathbb{I}\left[\mathsf{y} = \mathsf{m}_j\right] \cdot \log \mathsf{h}_{\perp,j}(\mathsf{x})$$

multi-expert implementation

training data

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

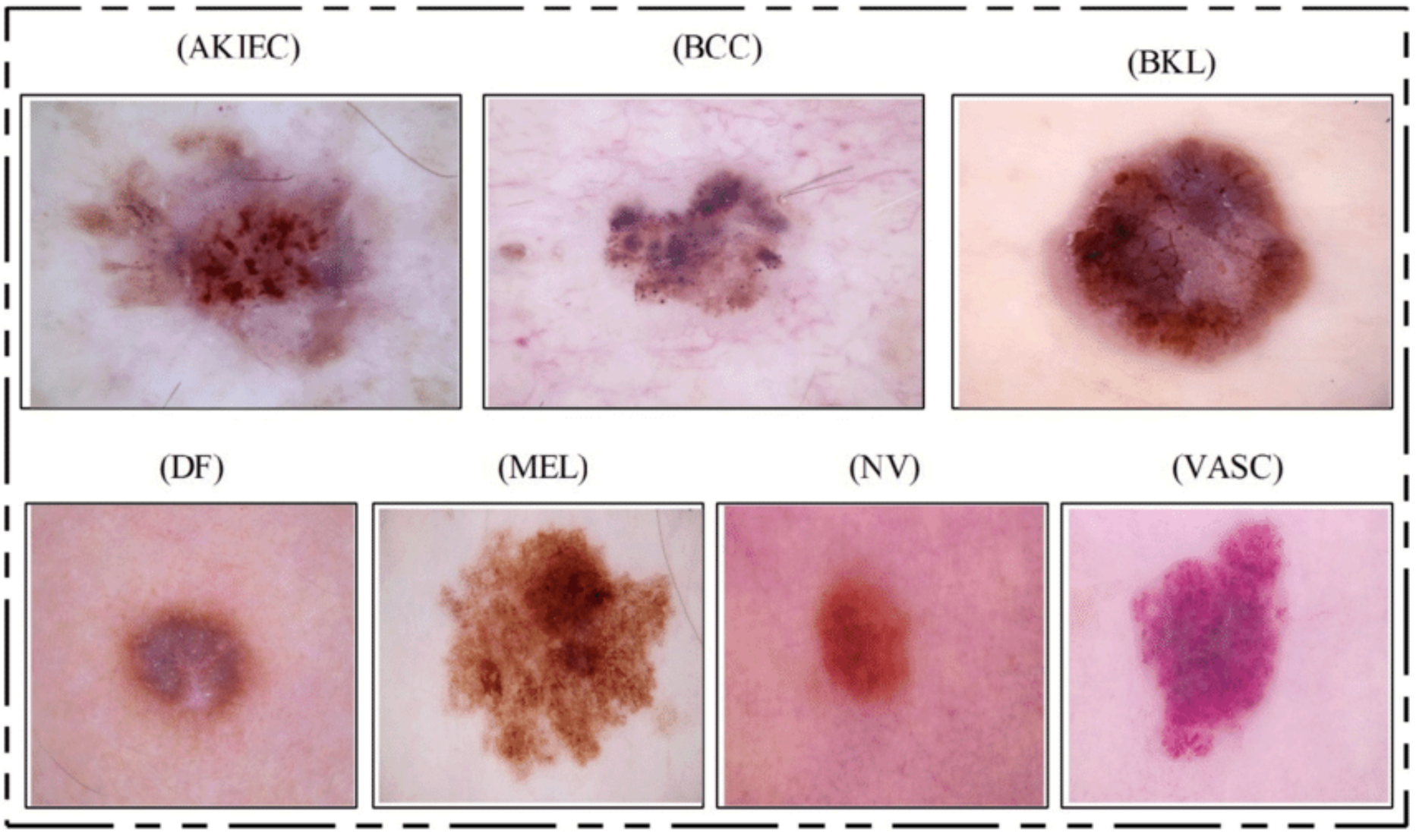
model

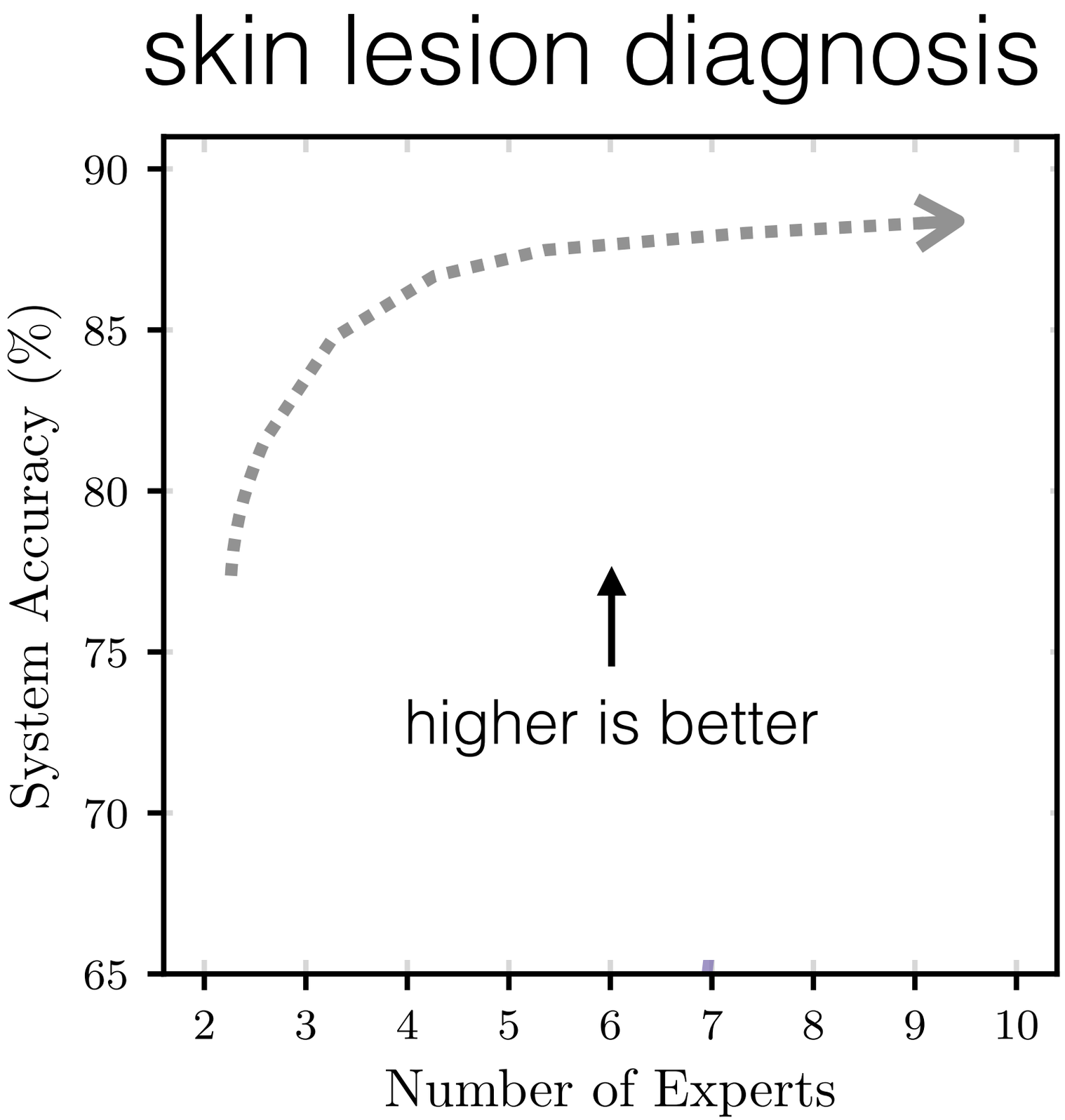
- ⊗ softmax and one-vs-all variants
- ⊗ both consistent w.r.t. 0-1 loss

softmax loss function

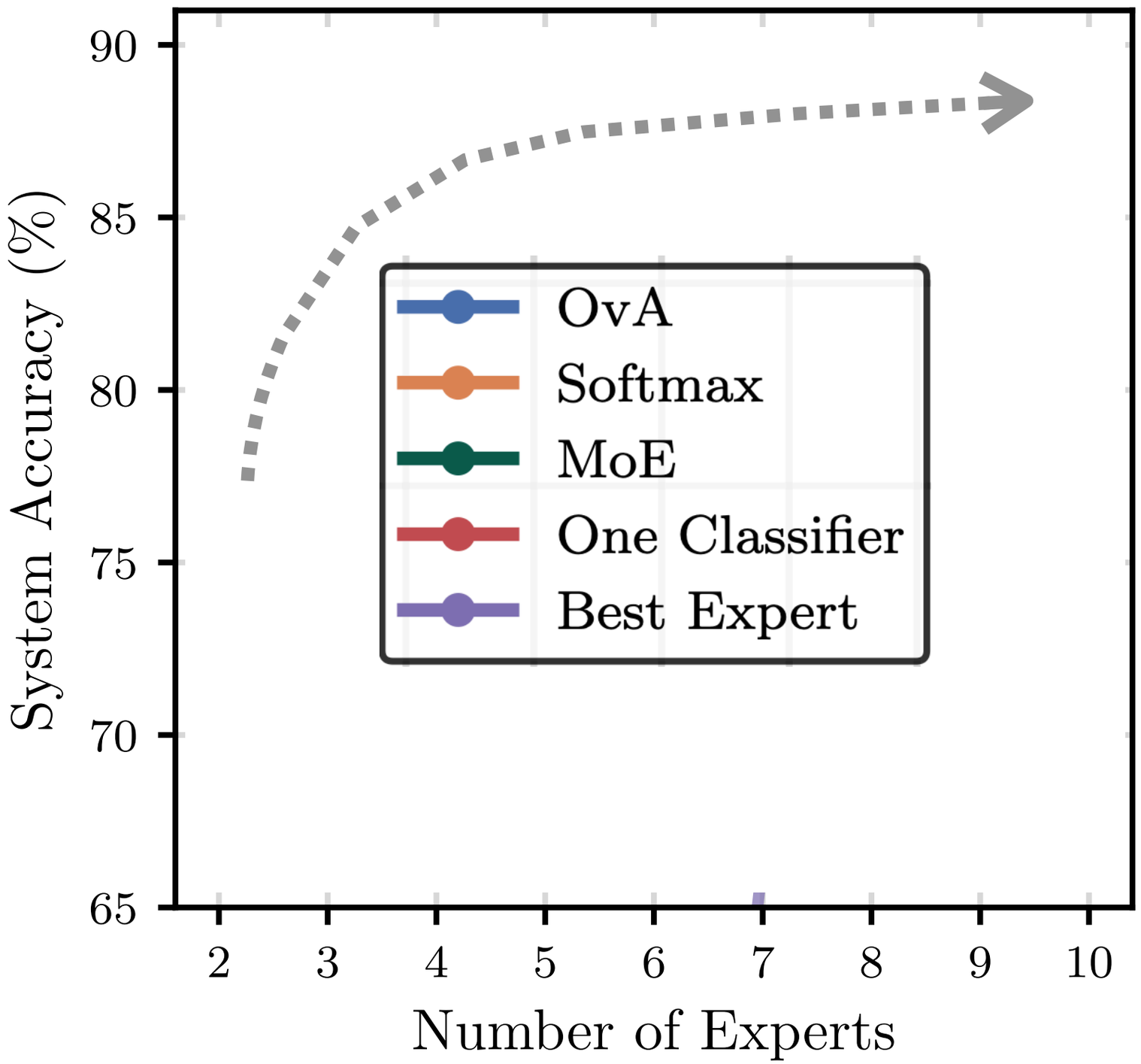
$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) - \sum_j \mathbb{I}\left[\mathsf{y} = \mathsf{m}_j\right] \cdot \log \mathsf{h}_{\perp,j}(\mathsf{x})$$

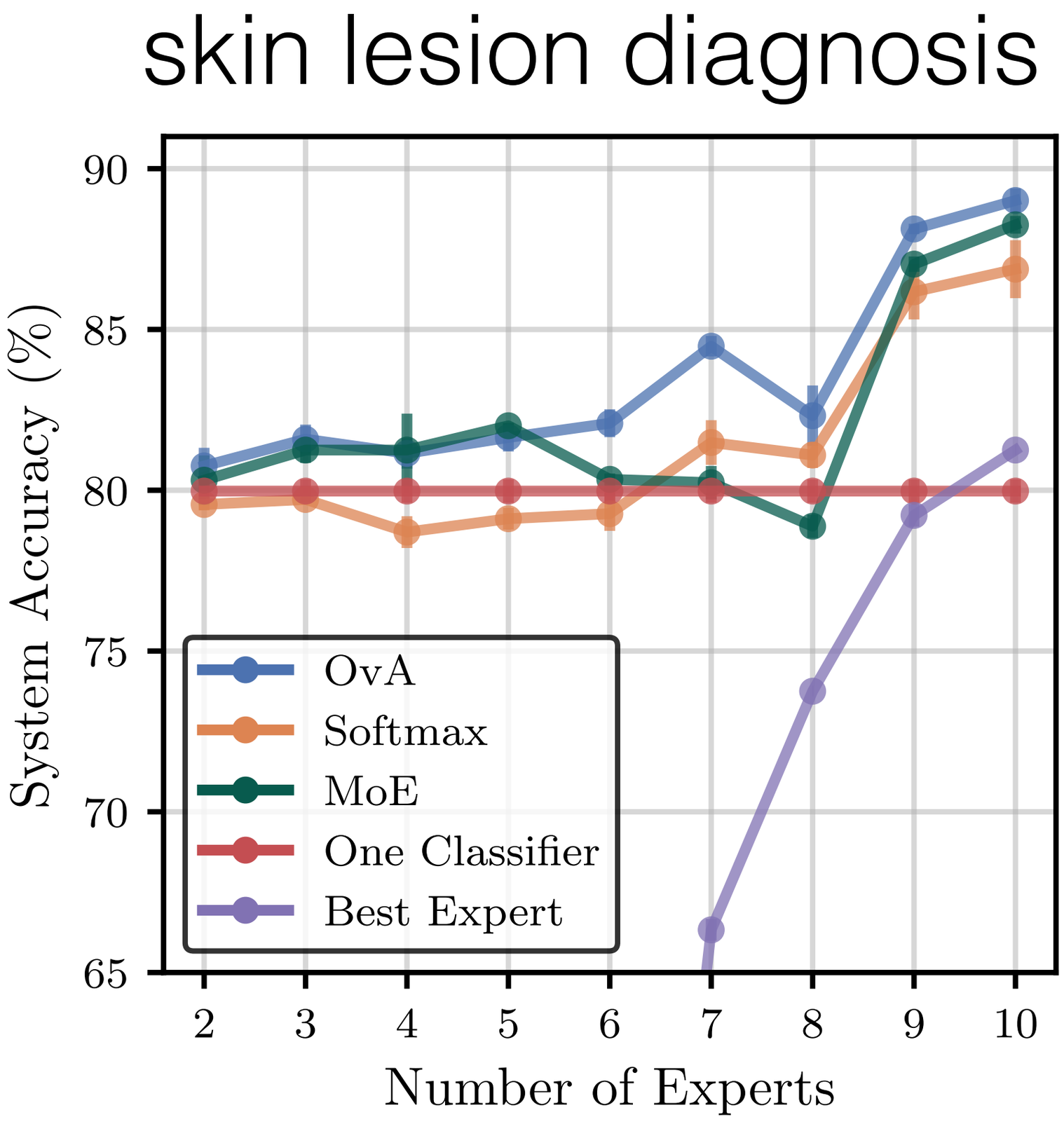
skin lesion diagnosis



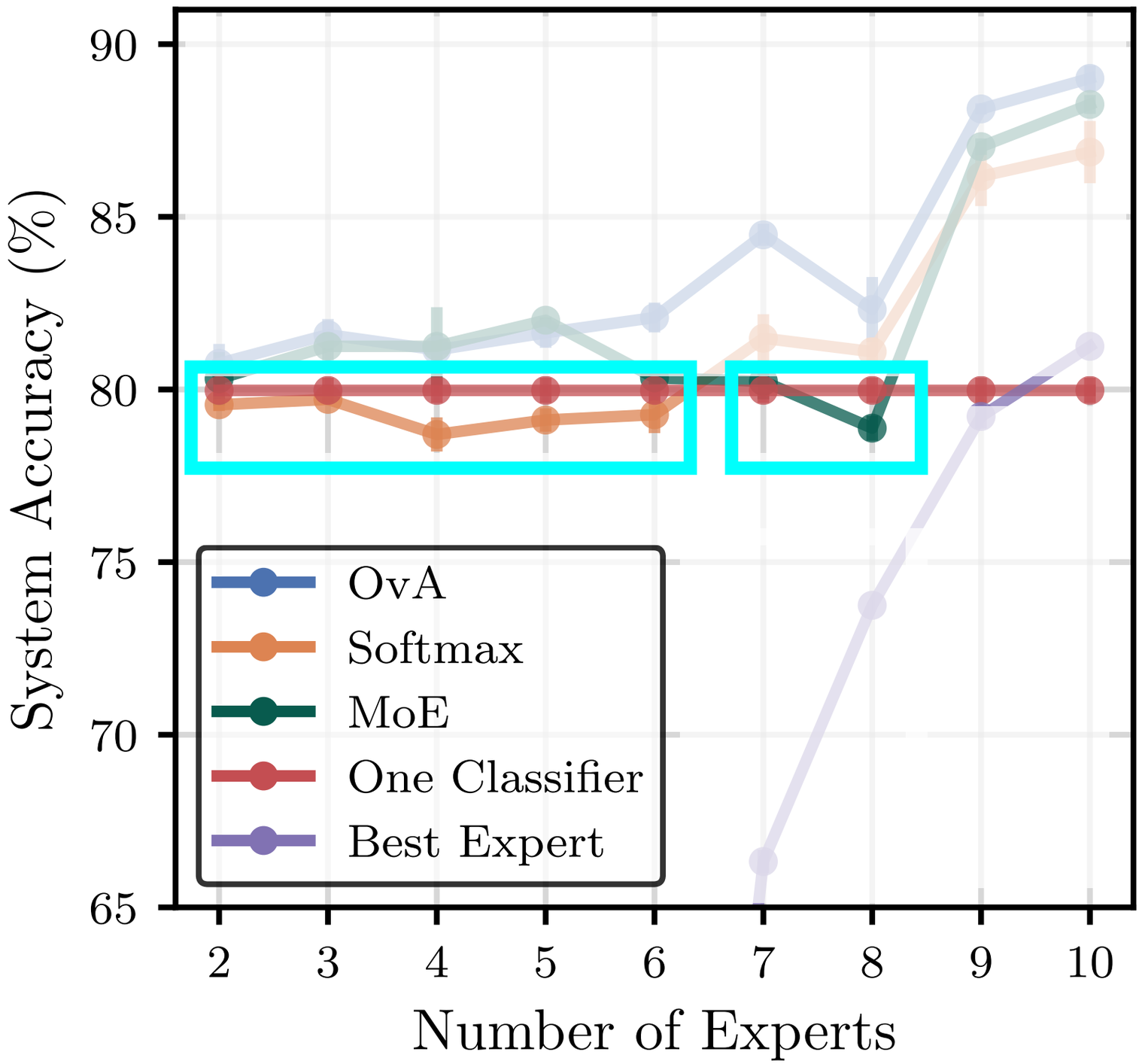

skin lesion diagnosis
System Accuracy (%)
90
85
80
75
70
65
2
3
4
5
6
7
8
9
10
Number of Experts
higher is better

skin lesion diagnosis

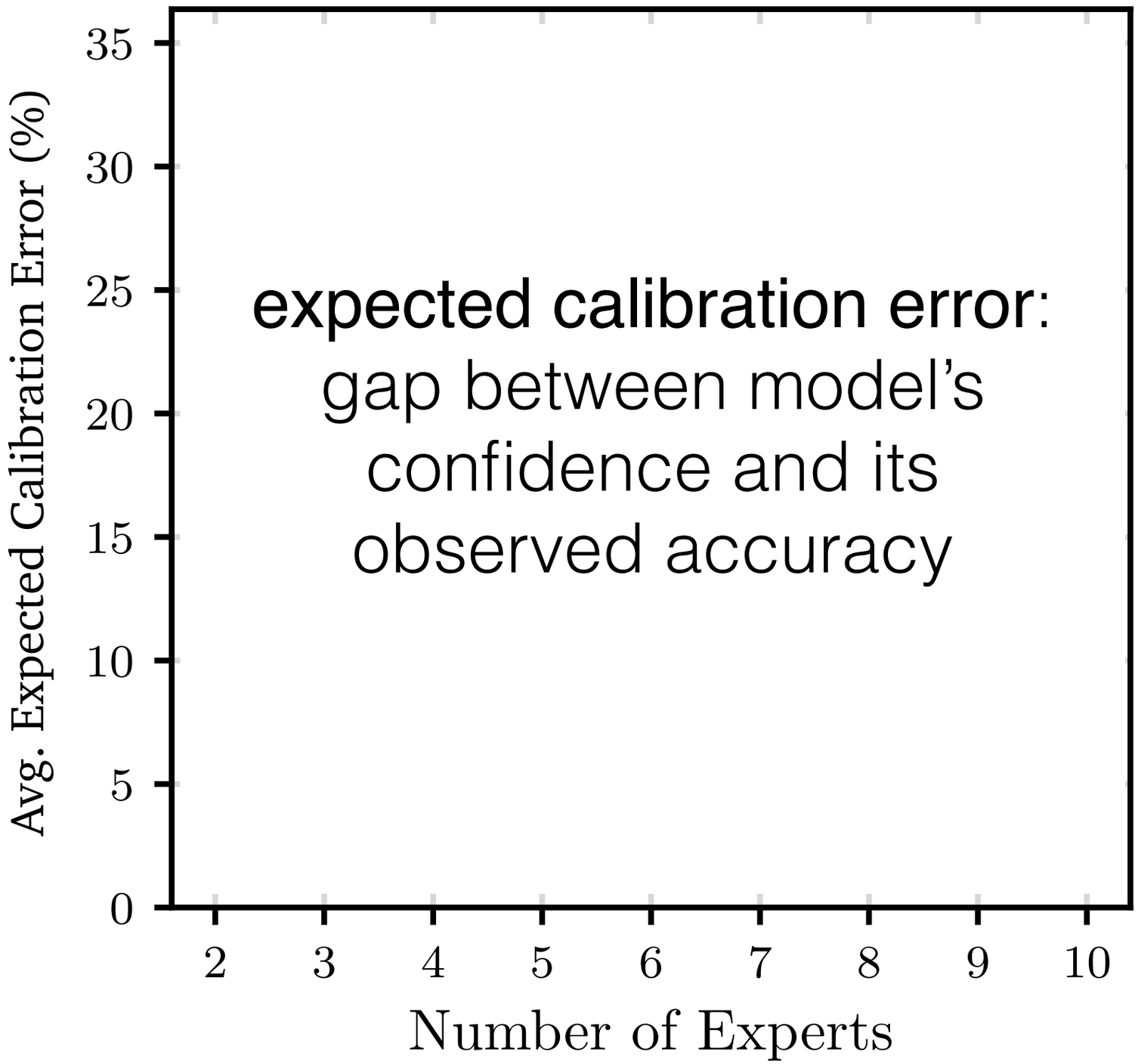


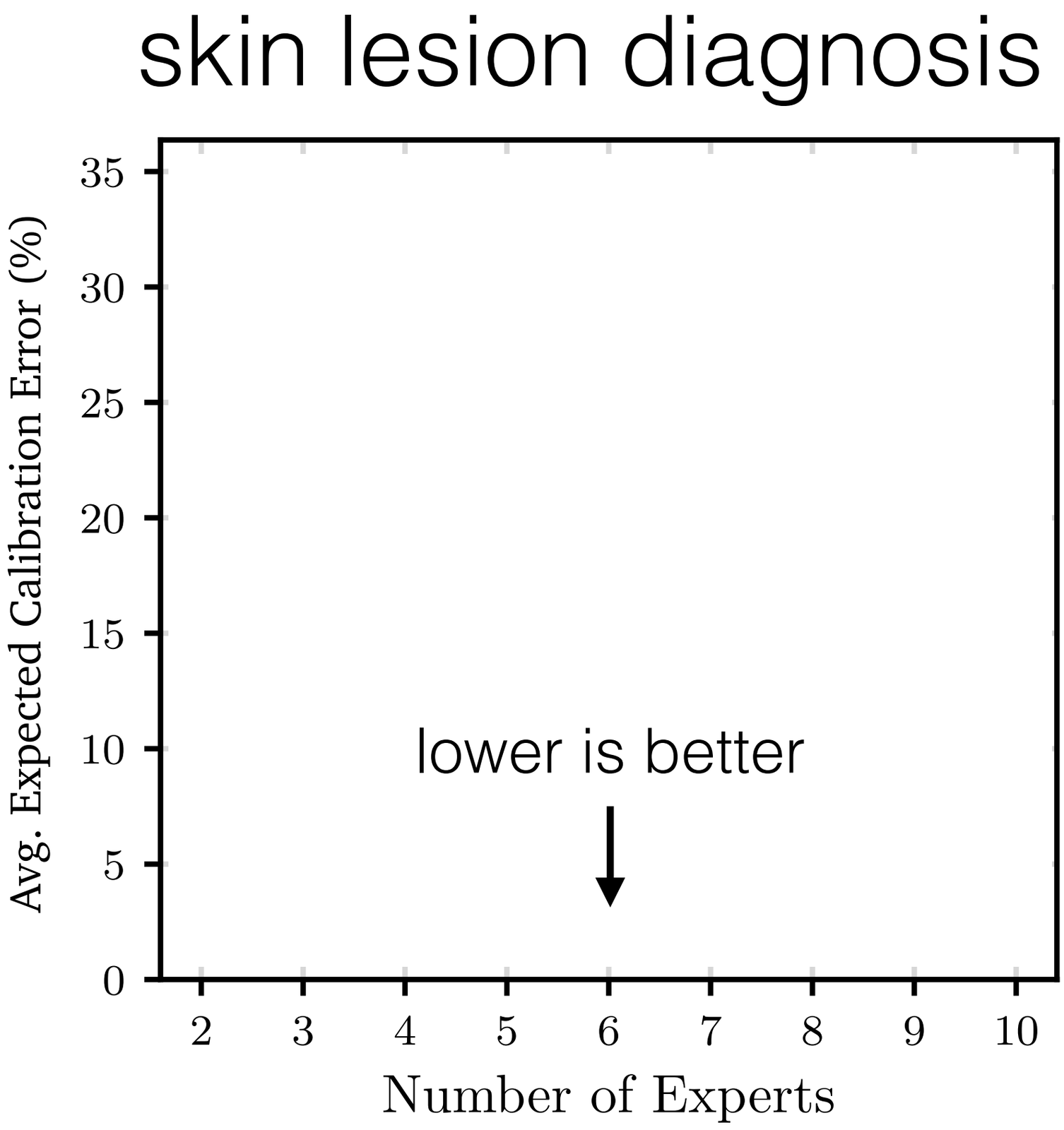

skin lesion diagnosis
System Accuracy (%)
90
85
80
75
70
65
2
3
4
5
6
7
8
9
10
Number of Experts
OvA
Softmax
MoE
One Classifier
Best Expert

skin lesion diagnosis

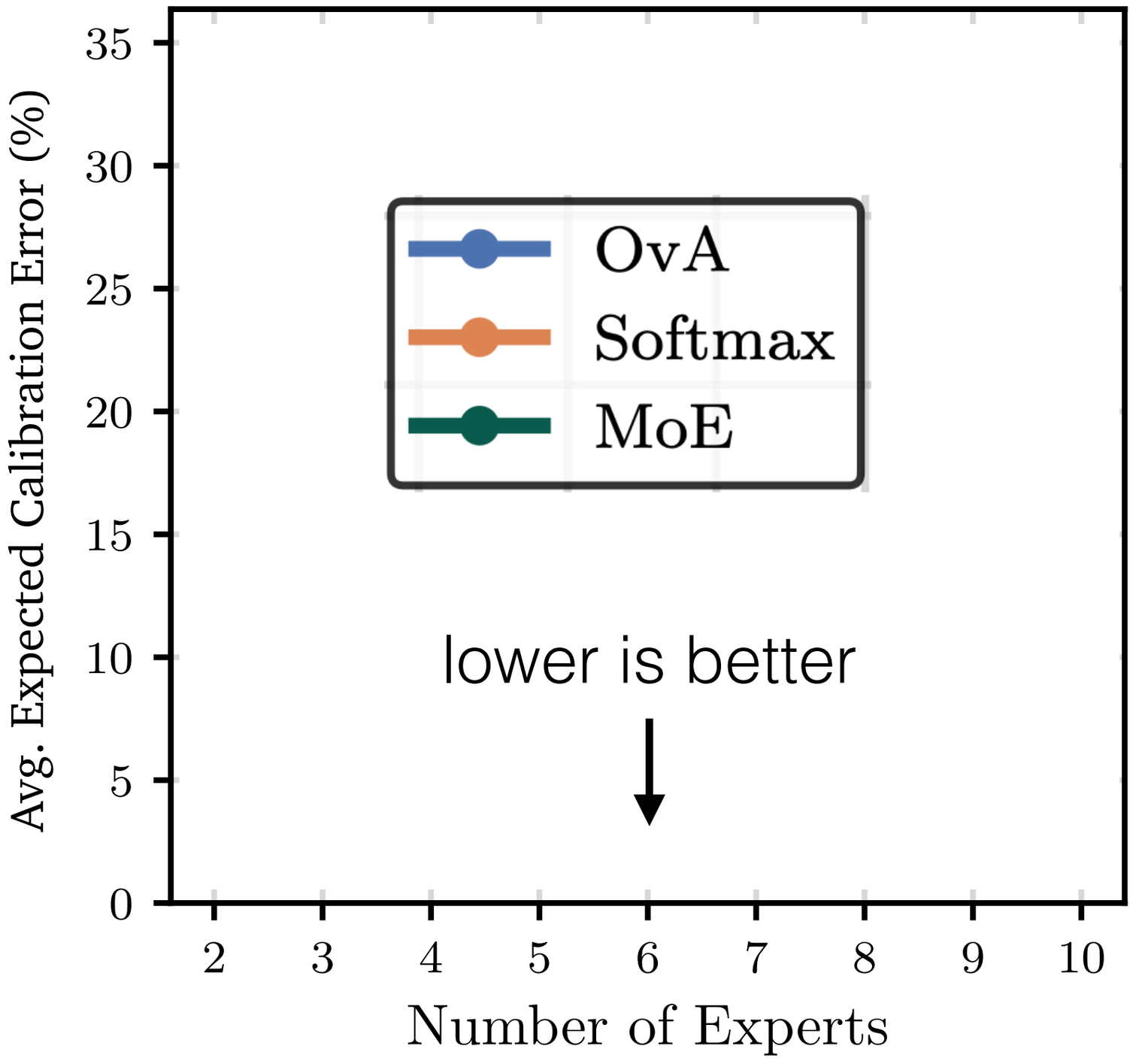


skin lesion diagnosis

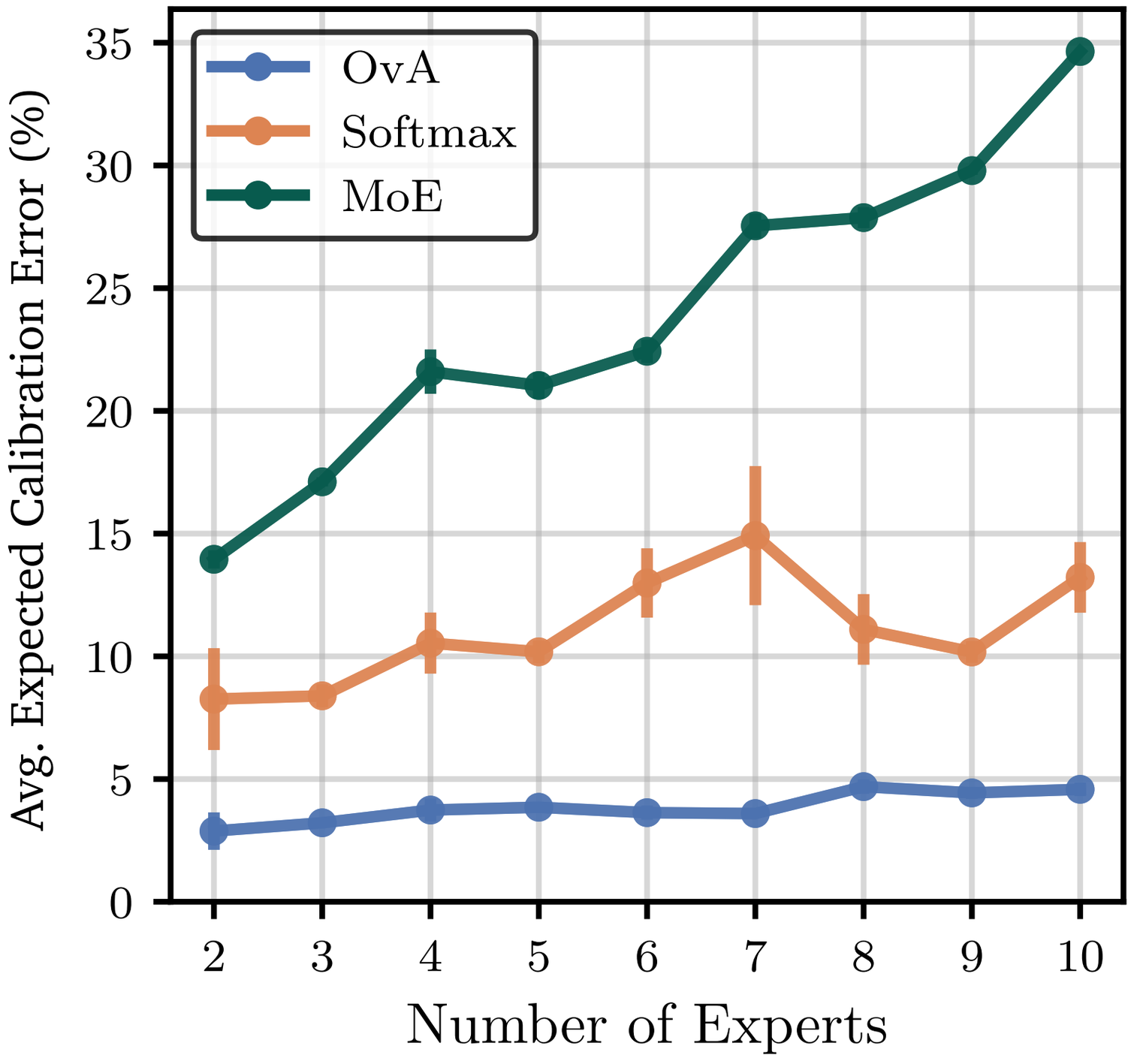



skin lesion diagnosis
Avg. Expected Calibration Error (%)
35
30
25
20
15
10
5
0
lower is better
2
3
4
5
6
7
8
9
10
Number of Experts

skin lesion diagnosis



skin lesion diagnosis

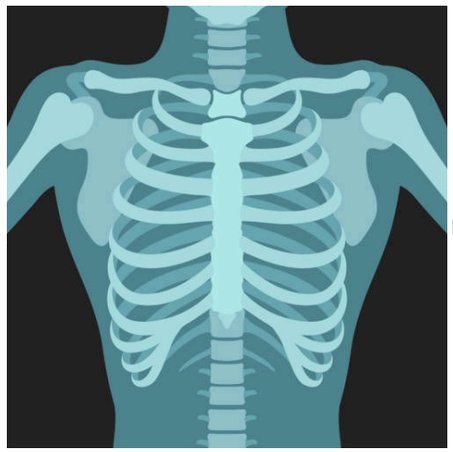


- single expert
  - softmax surrogate loss
  - improving calibration via one-vs-all

- multiple experts
  - surrogate losses
  - conformal sets of experts

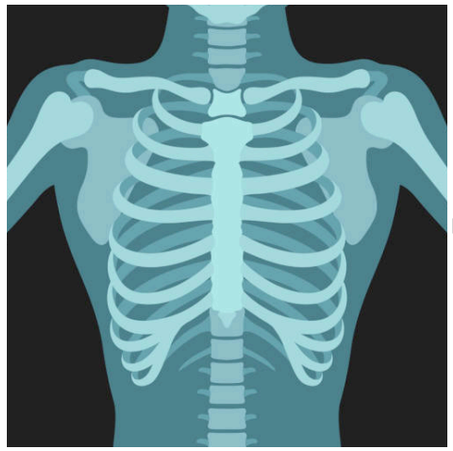
- population of experts
  - surrogate losses
  - meta-learning a rejector

input features



allocation mechanism

input features



allocation mechanism

?

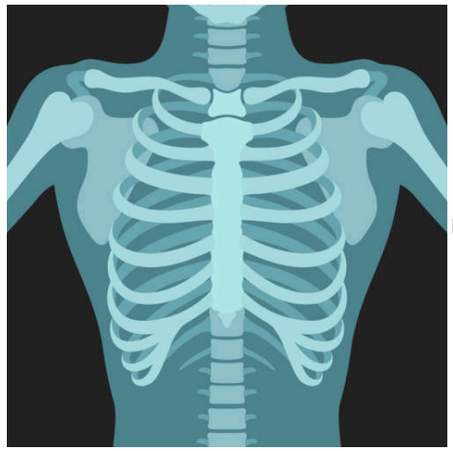
classifier

expert #1

expert #3

expert #2

input features



allocation mechanism

classifier

expert #1

expert #3

expert #2

conformal inference

assume there's a best expert, j*:

$$\mathbb{P}\left(\mathsf{m}_{j*} = \mathsf{y} \,|\, \mathsf{x}\right) \;>\; \mathbb{P}\left(\mathsf{m}_{e} = \mathsf{y} \,|\, \mathsf{x}\right), \;\; \forall e \neq j*$$

construct a confidence set of experts:

$$\mathbb{P}\left(\mathsf{j}* \in \mathsf{C}(\mathsf{x})\right) \;\geq\; 1 - \alpha$$

team of experts: adaptive in size and membership

conformal inference: ensembling

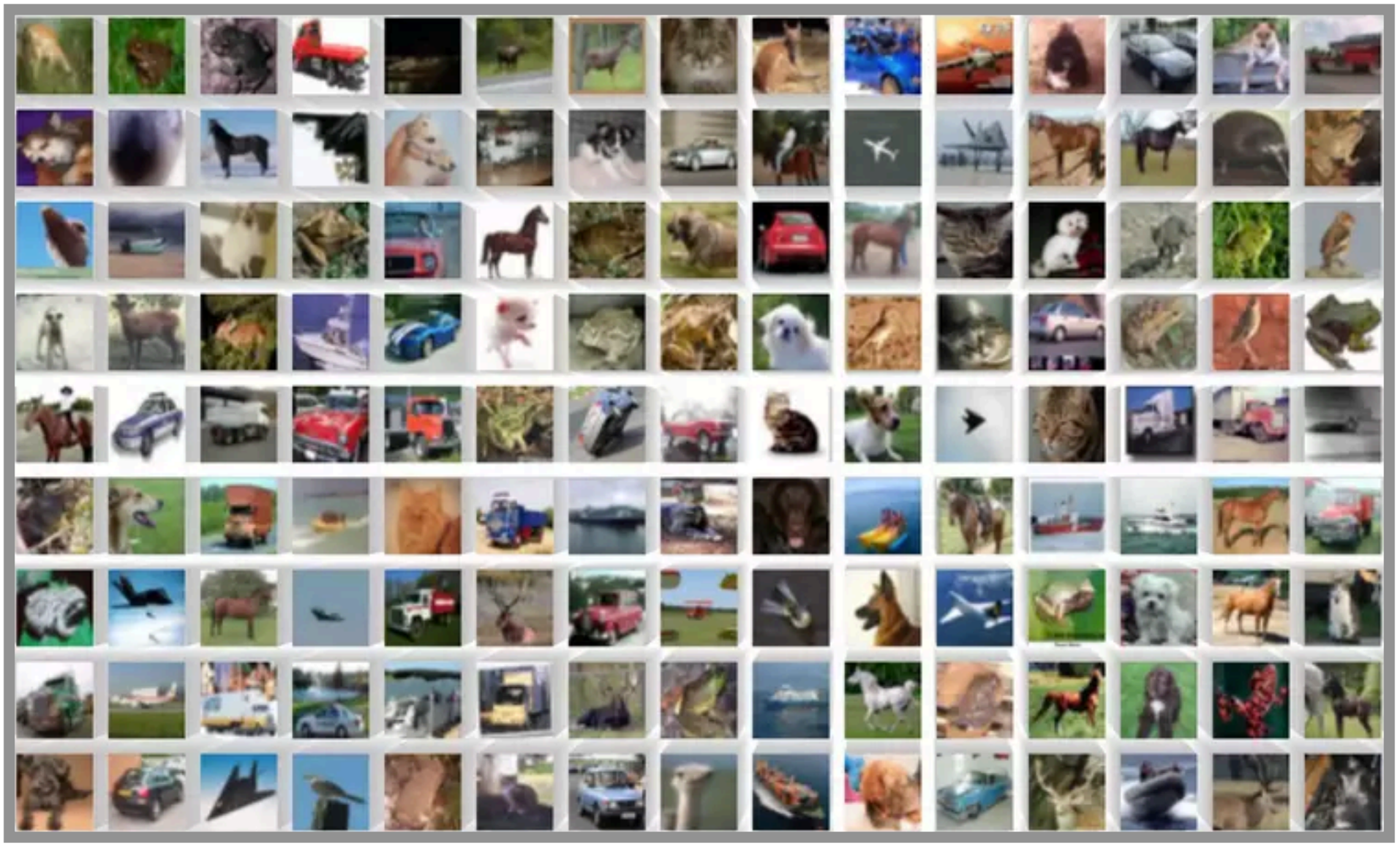
$$C(x) = \{ \quad , \quad \}$$

predictions: m_3 m_1

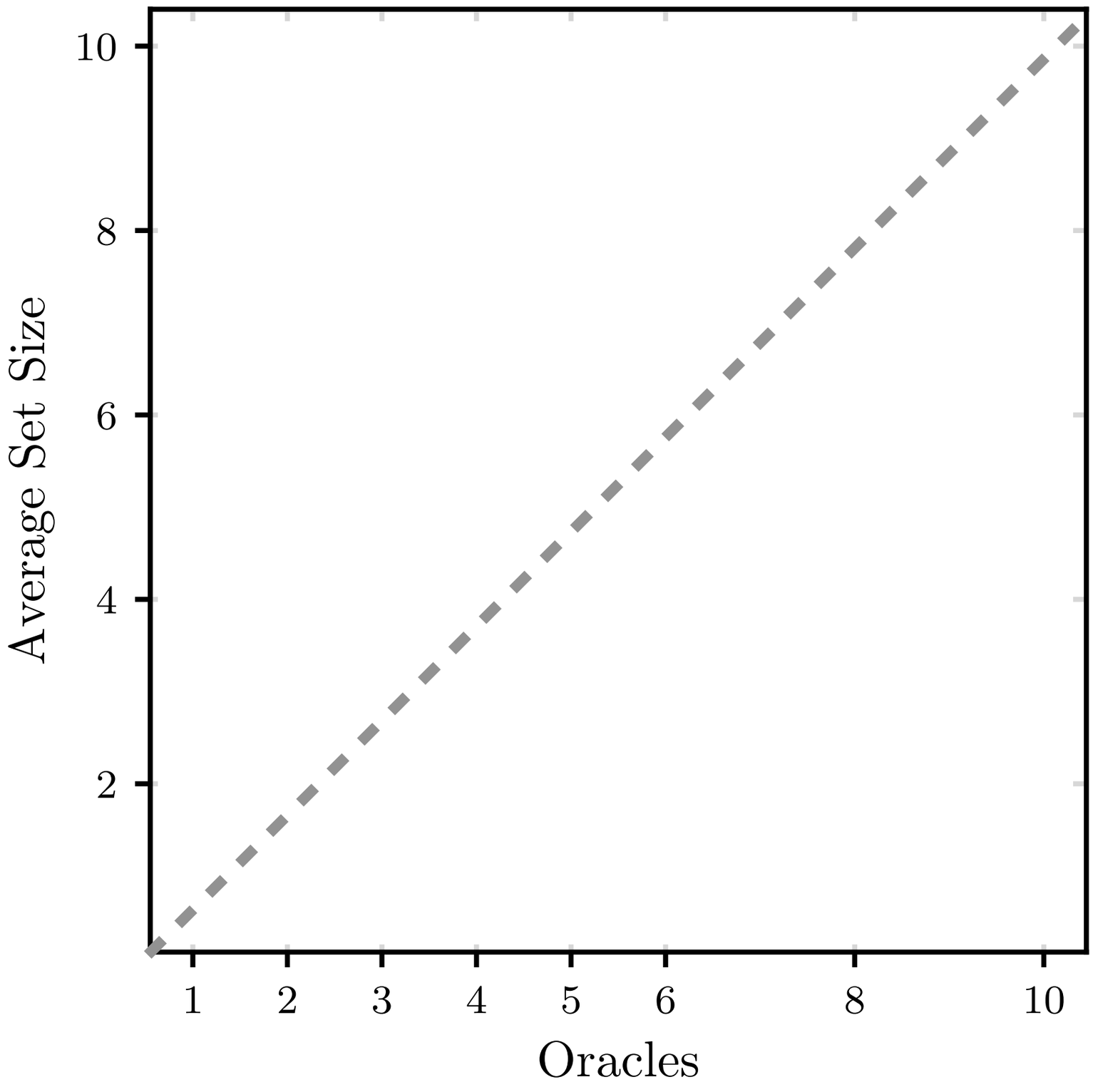
aggregated prediction: $\bar{m}$
(e.g. by voting)

expert selection

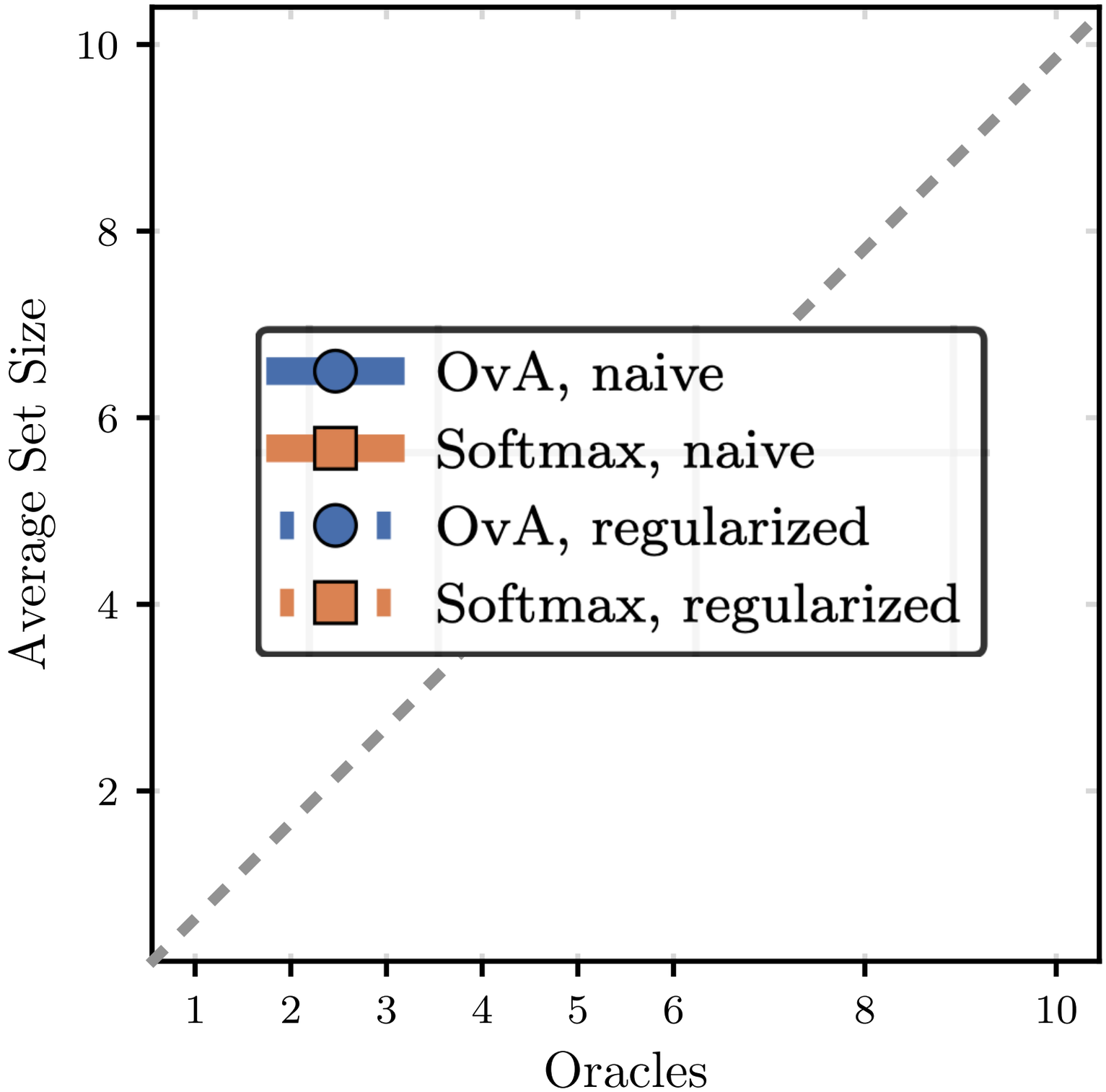
CIFAR-10



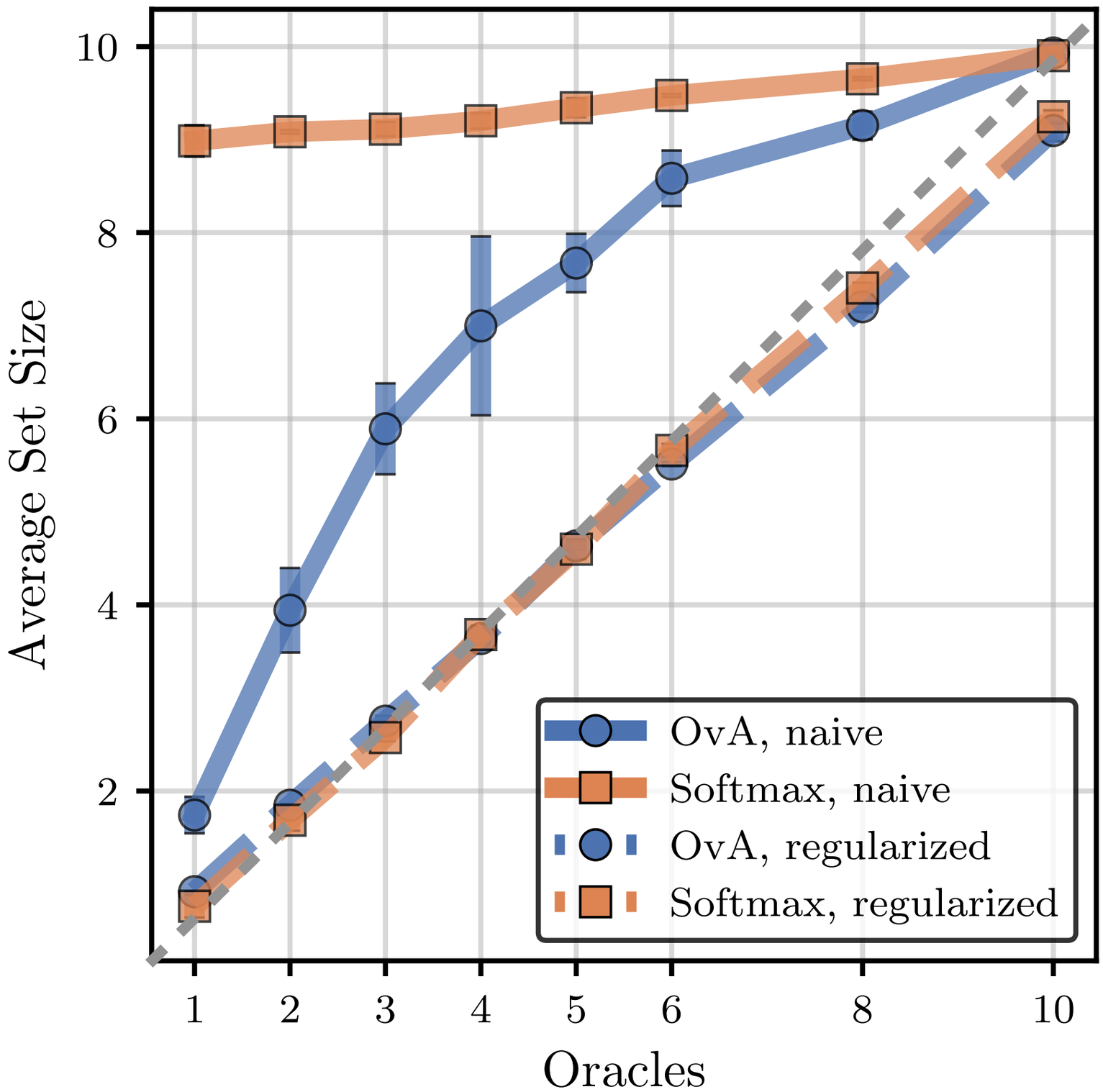
expert selection



expert selection

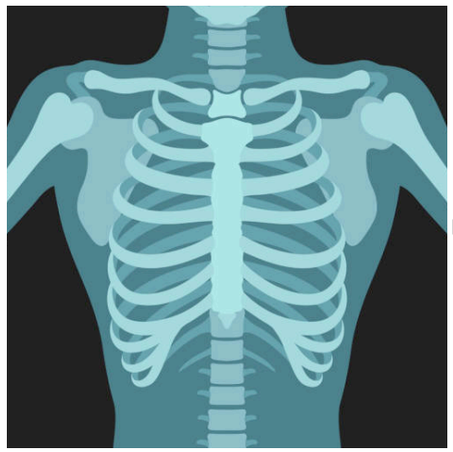


expert selection



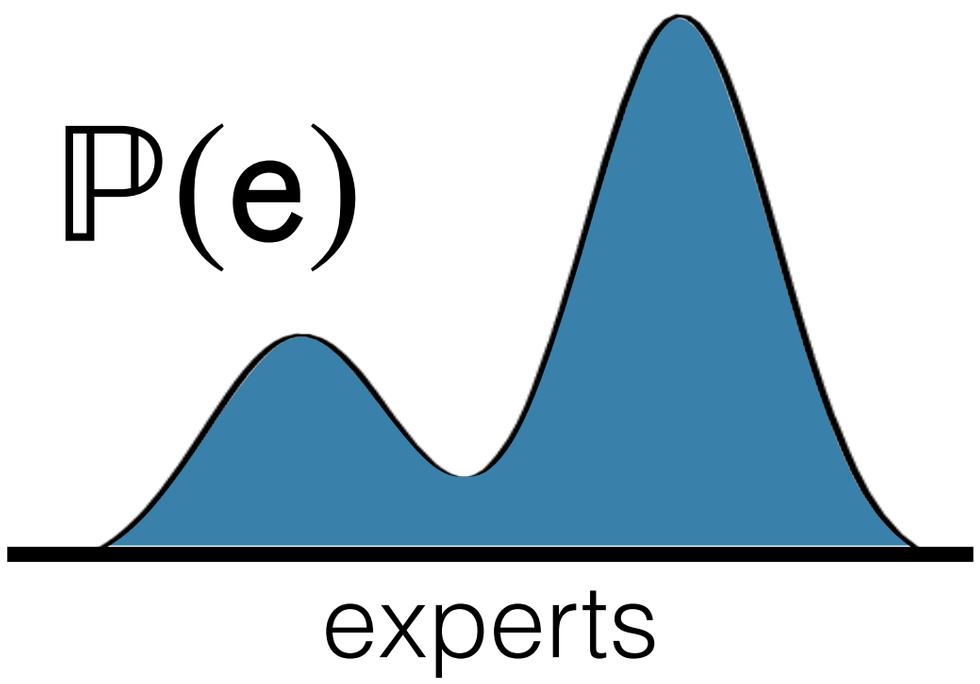
- single expert
  - softmax surrogate loss
  - improving calibration via one-vs-all
- multiple experts
  - surrogate losses
  - conformal sets of experts
- **population of experts**
  - surrogate losses
  - meta-learning a rejector

input features



allocation mechanism

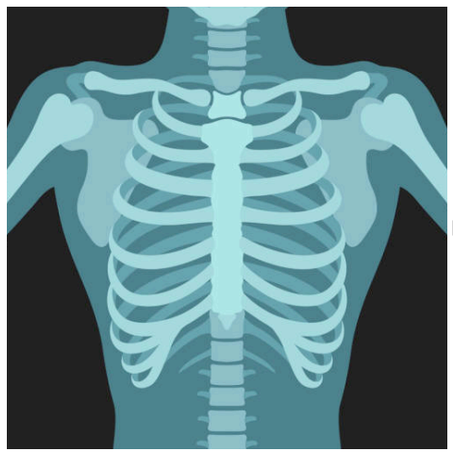
classifier



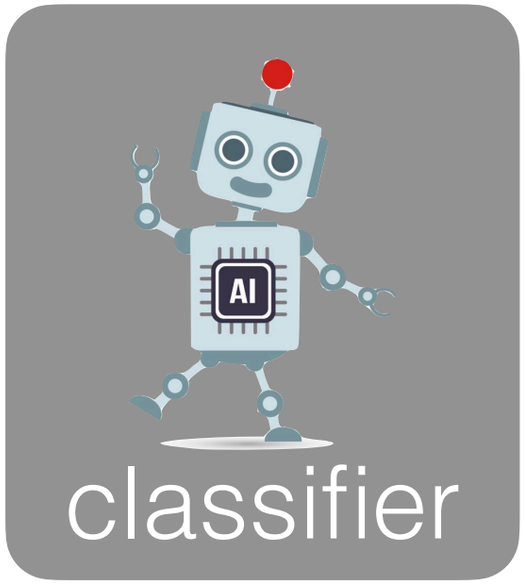
?

expert

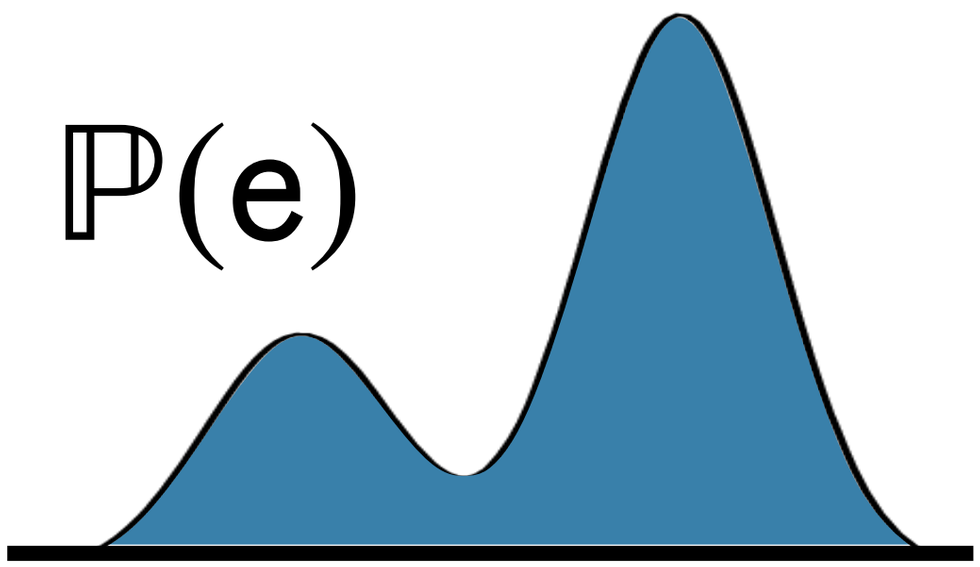
input features



allocation mechanism



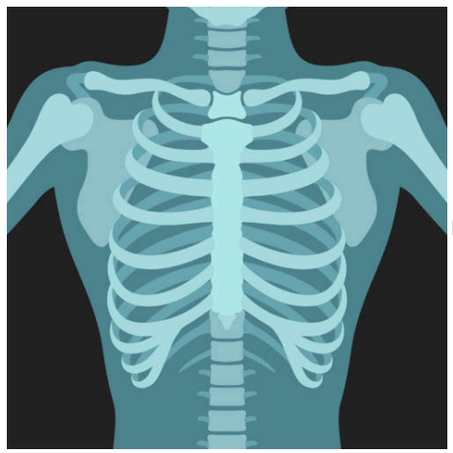
$\mathbb{P}(\mathbf{e})$



experts

$\mathbf{e} \sim \mathbb{P}(\mathbf{e})$

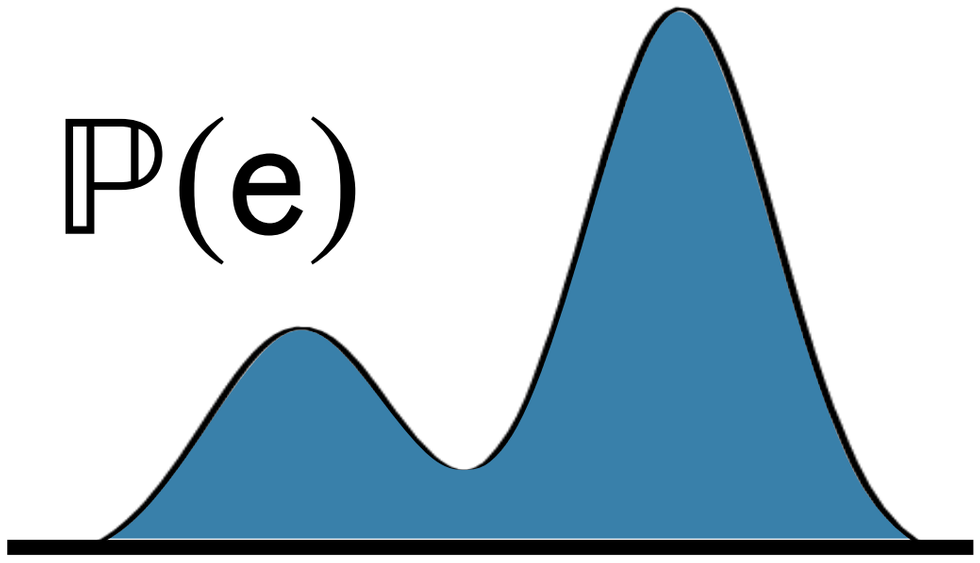
input features



allocation mechanism

classifier

$\mathbb{P}(\mathbf{e})$

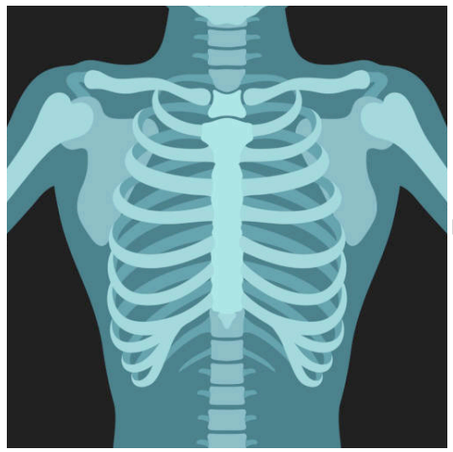


experts

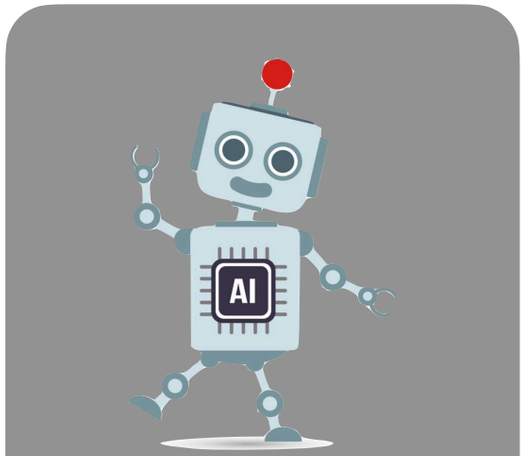
$\mathbf{e} \sim \mathbb{P}(\mathbf{e})$

expert

input features



allocation mechanism



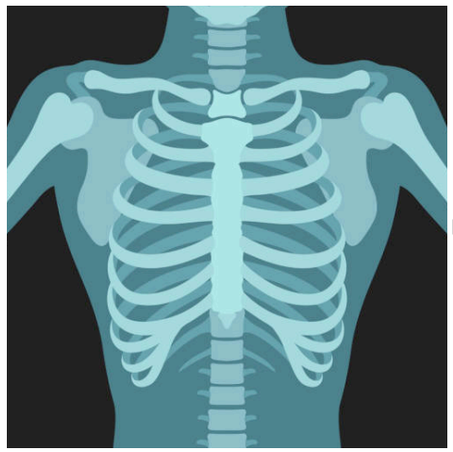
classifier

expert

$$e \sim \mathbb{P}(e)$$

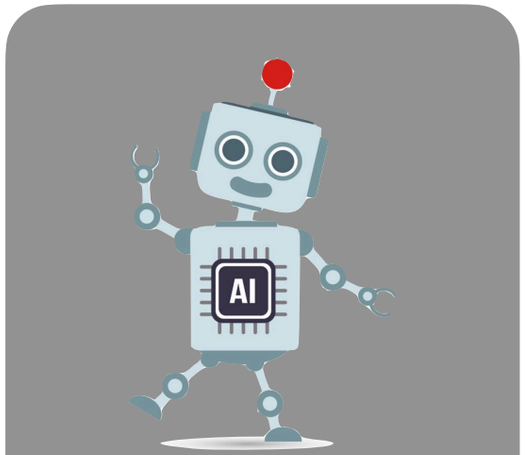
$$m \sim \mathbb{P}(m \mid e)$$

input features



allocation mechanism

L_{0-1}



classifier

L_{0-1}

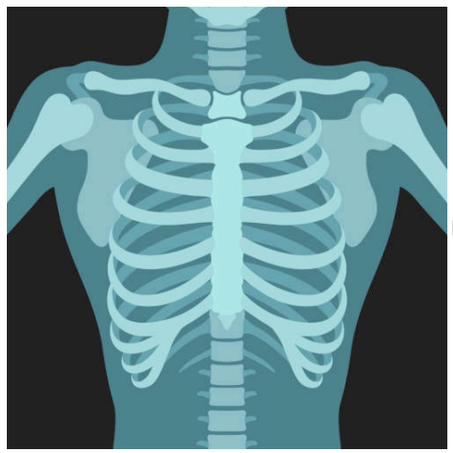
?

expert

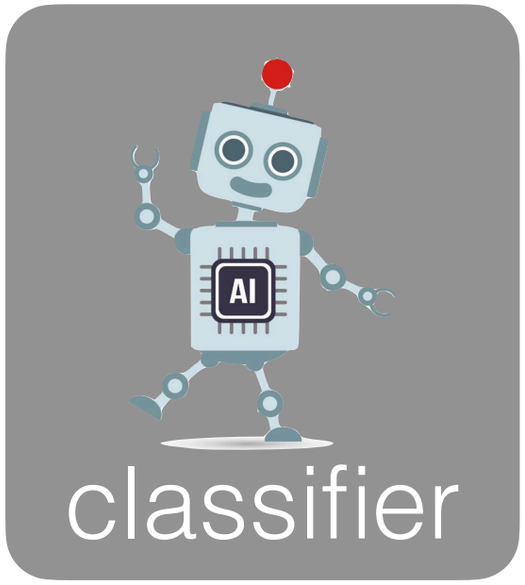
Bayes optimal deferral rule:

$$\max_{y} \ \mathbb{P}(y \mid \mathbf{x}) \ \leq \ \mathbb{P}(\mathsf{m} = \mathsf{y} \mid \mathbf{x}, \mathsf{e})$$

input features



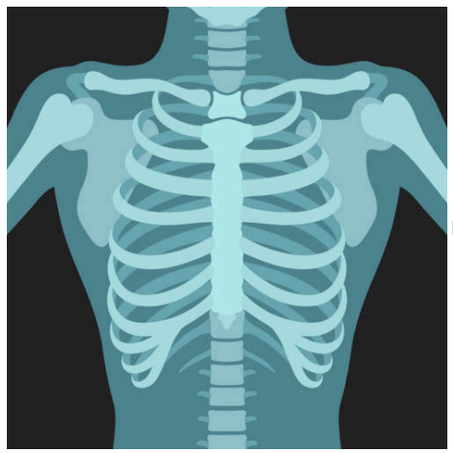
allocation mechanism



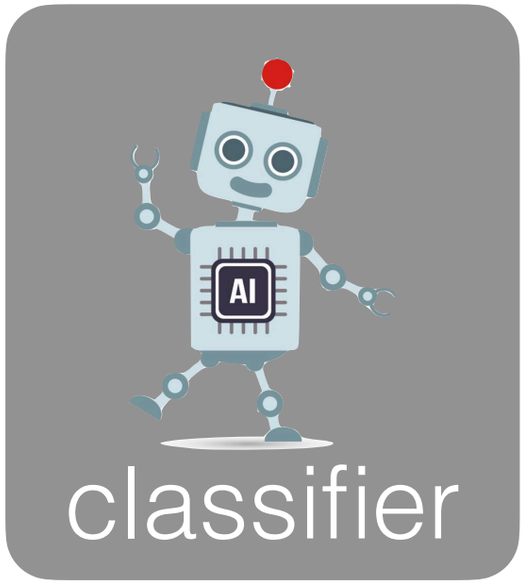
defer to expert if…

$$\max_{y \in [1,K]} \; \mathsf{h}_y(\mathsf{x}) \;\leq\; \mathsf{h}_\perp(\mathsf{x}, \mathsf{e})$$

input features



allocation mechanism



defer to expert if…

?

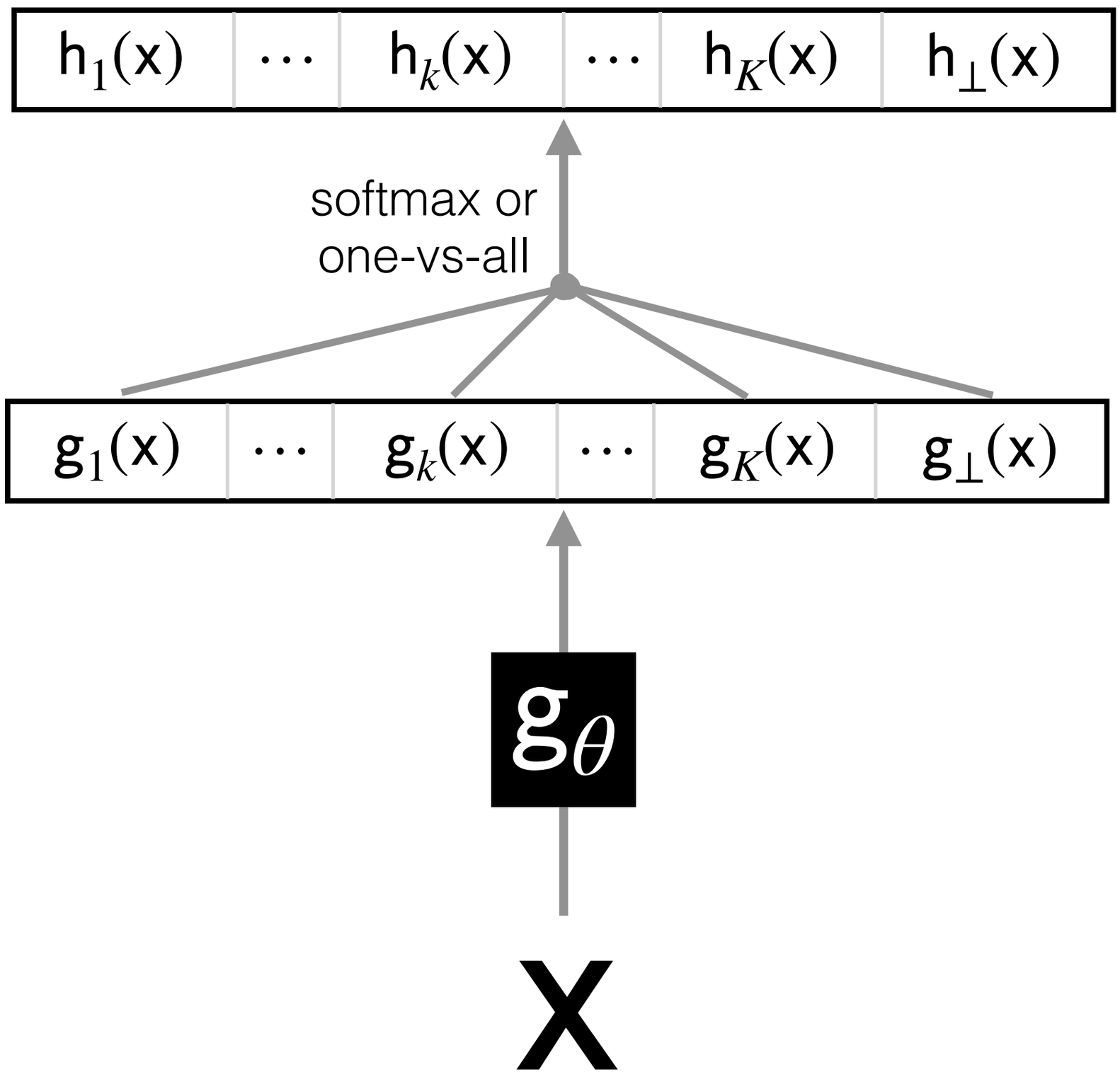
$$\max_{y \in [1,K]} \; h_y(x) \;\leq\; h_\perp(x, e)$$

- single expert
  - softmax surrogate loss
  - improving calibration via one-vs-all

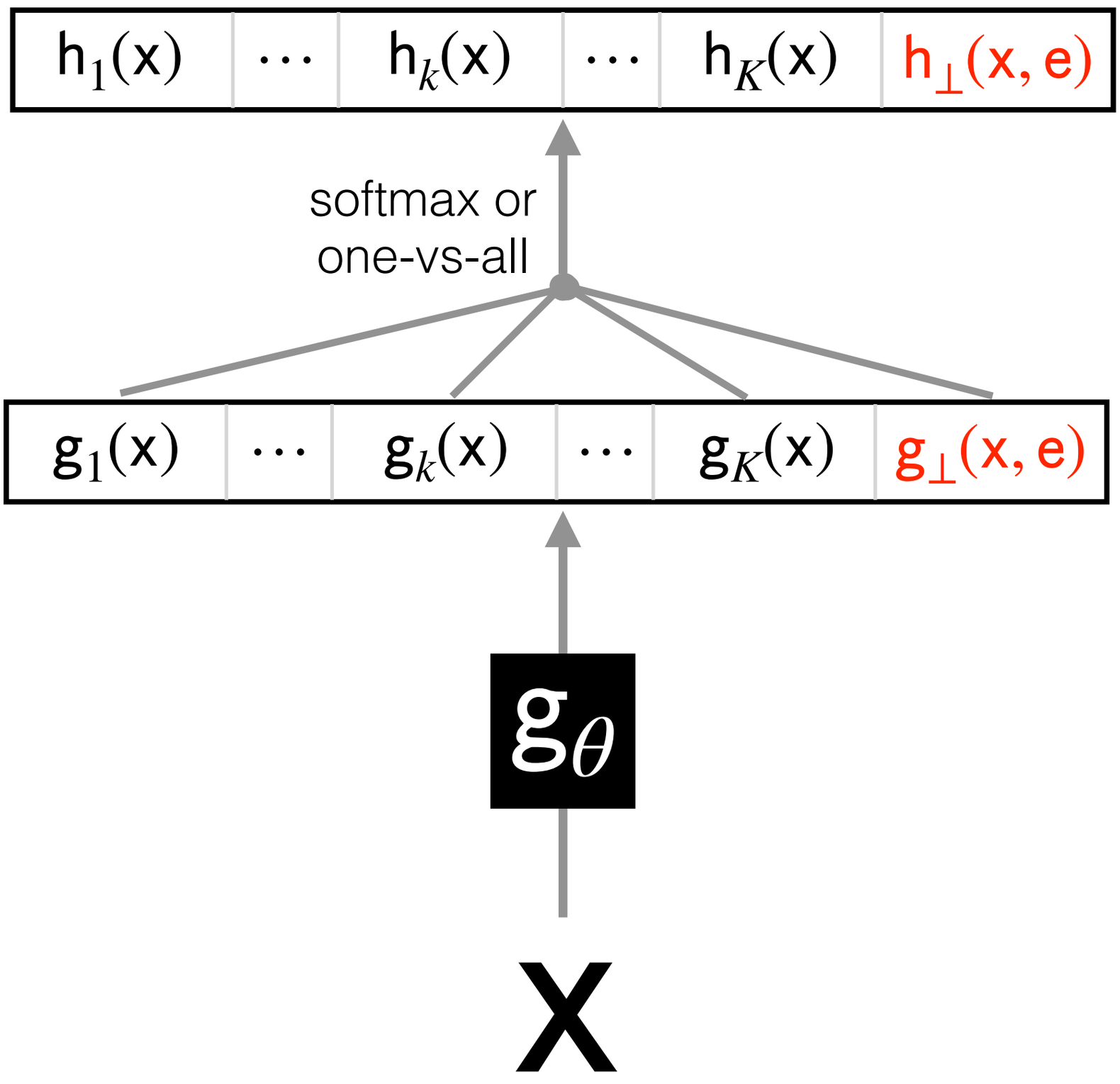
- multiple experts
  - surrogate losses
  - conformal sets of experts

- **population of experts**
  - surrogate losses
  - meta-learning a rejector

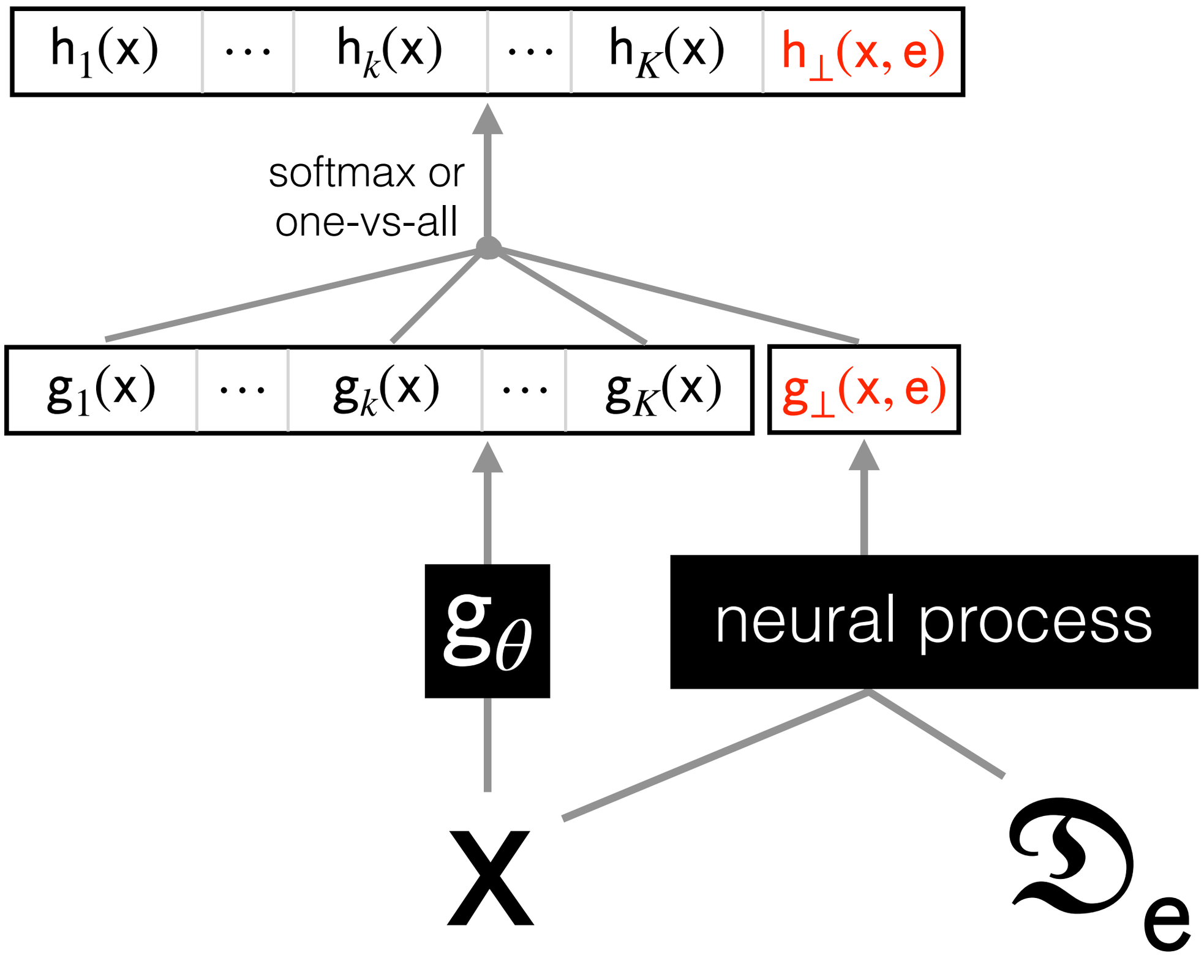
meta-learning implementation



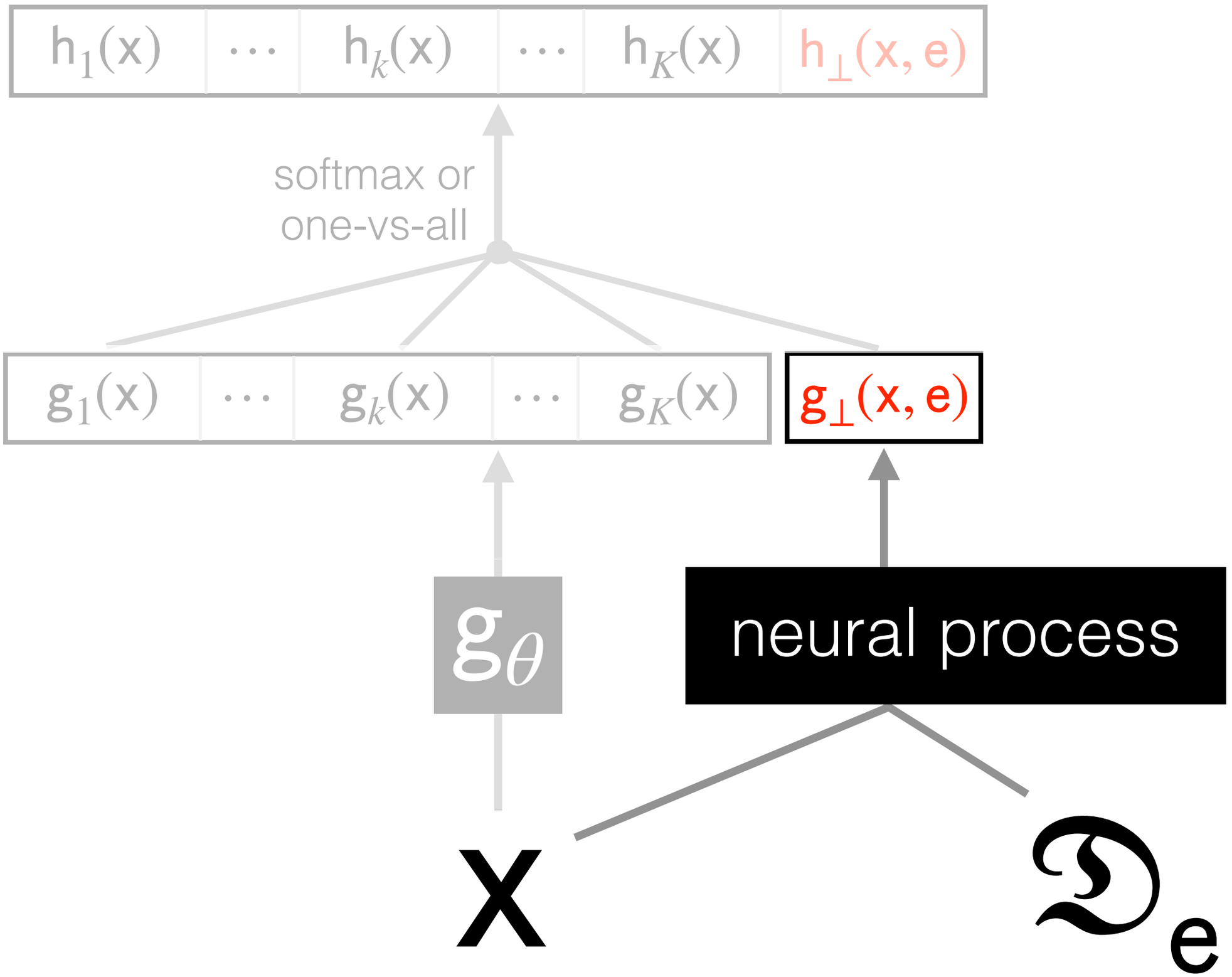
meta-learning implementation



meta-learning implementation



meta-learning implementation



neural process rejector

$$g_\perp(x, e)$$

feedforward neural network

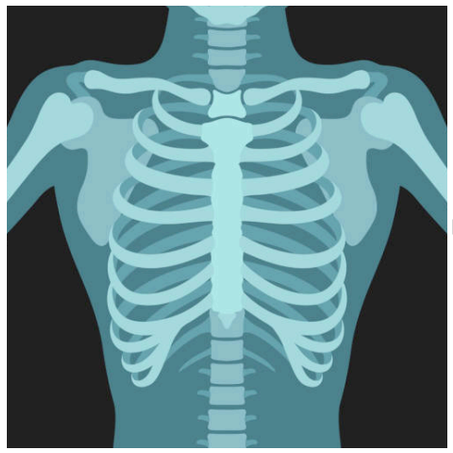
x

set encoder (permutation-invariant)

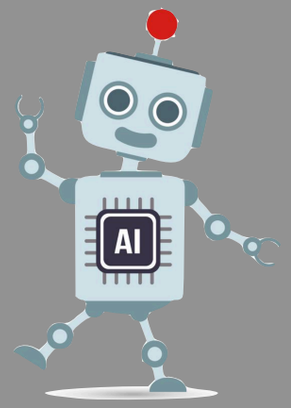
$$\mathfrak{D}_e = \{(x_n, y_n, m_{e,n})\}_{n=1}^{N}$$

- ⊗ single expert
  - ⊗ softmax surrogate loss
  - ⊗ improving calibration via one-vs-all
- ⊗ multiple experts
  - ⊗ surrogate losses
  - ⊗ conformal sets of experts
- ⊗ population of experts
  - ⊗ surrogate losses
  - ⊗ meta-learning a rejector

input features



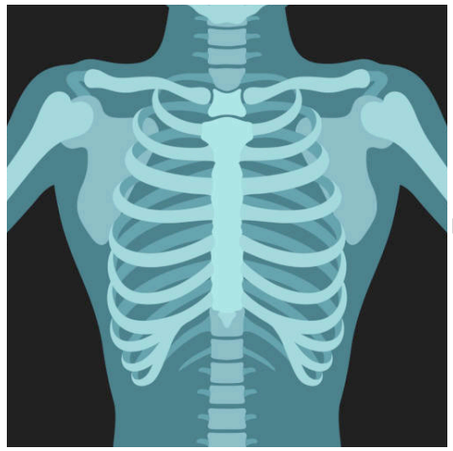
allocation mechanism



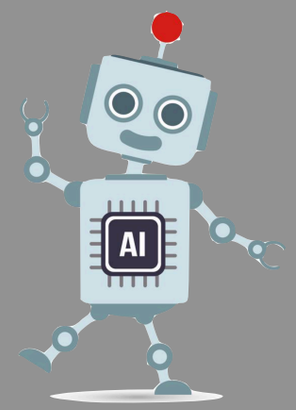
classifier

expert

input features



allocation mechanism



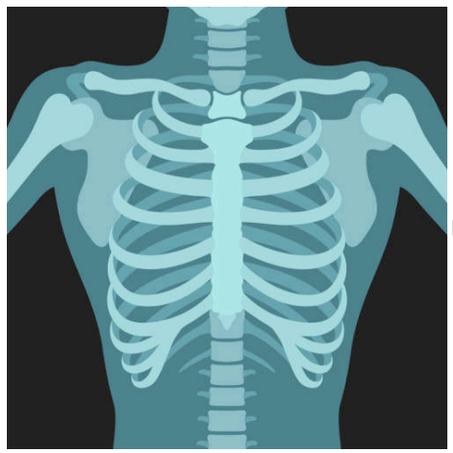
classifier

expert #1

expert #3

expert #2

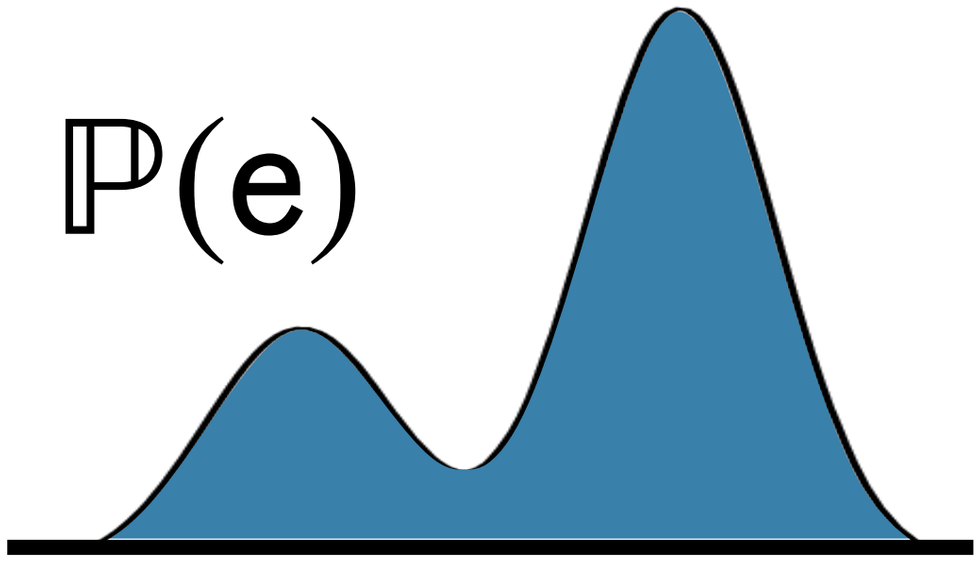
input features



allocation mechanism

classifier

$\mathbb{P}(\mathbf{e})$

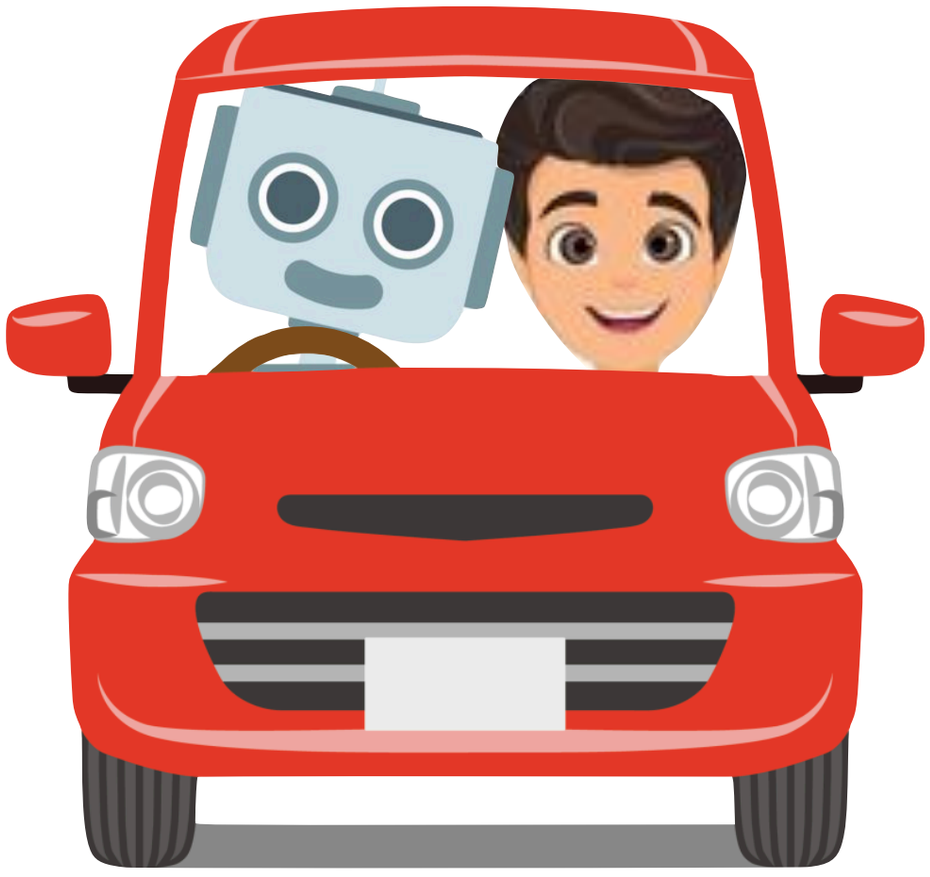


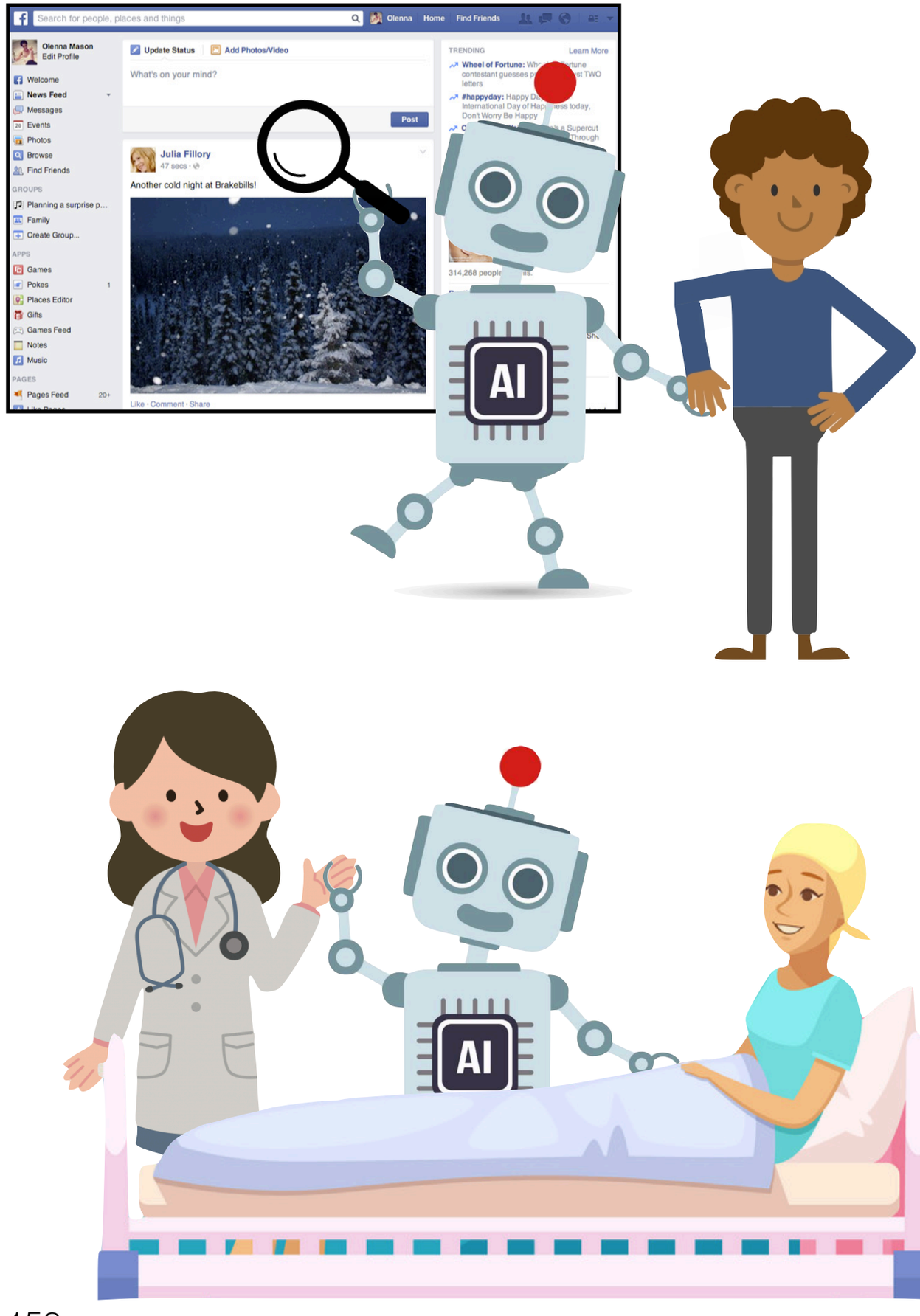
experts

$\mathbf{e} \sim \mathbb{P}(\mathbf{e})$

?

expert




Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
AI

AI

papers & code

funding provided by

Netherlands Organisation for Scientific Research

co-authors

Rajeev Verma

Daniel Barrejón

Dharmesh Tailor

Putra Manggala

Aditya Patra