Learning to Defer to a Population: A Meta-Learning Approach





Dharmesh Tailor Aditya Patra Rajeev Verma Putra Manggala Eric Nalisnick

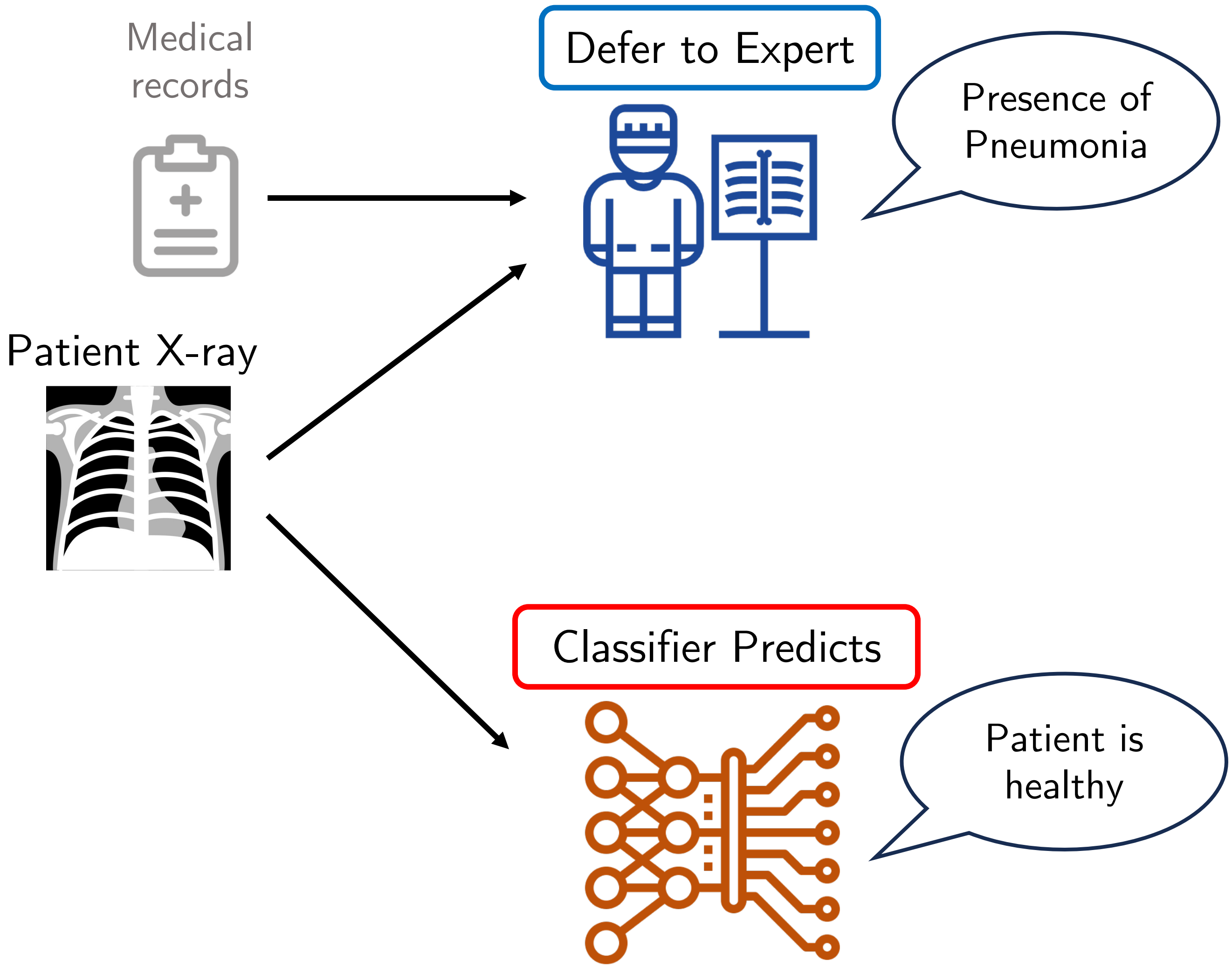
University of Amsterdam

AMLAB Amsterdam Machine Learning Lab

AISTATS

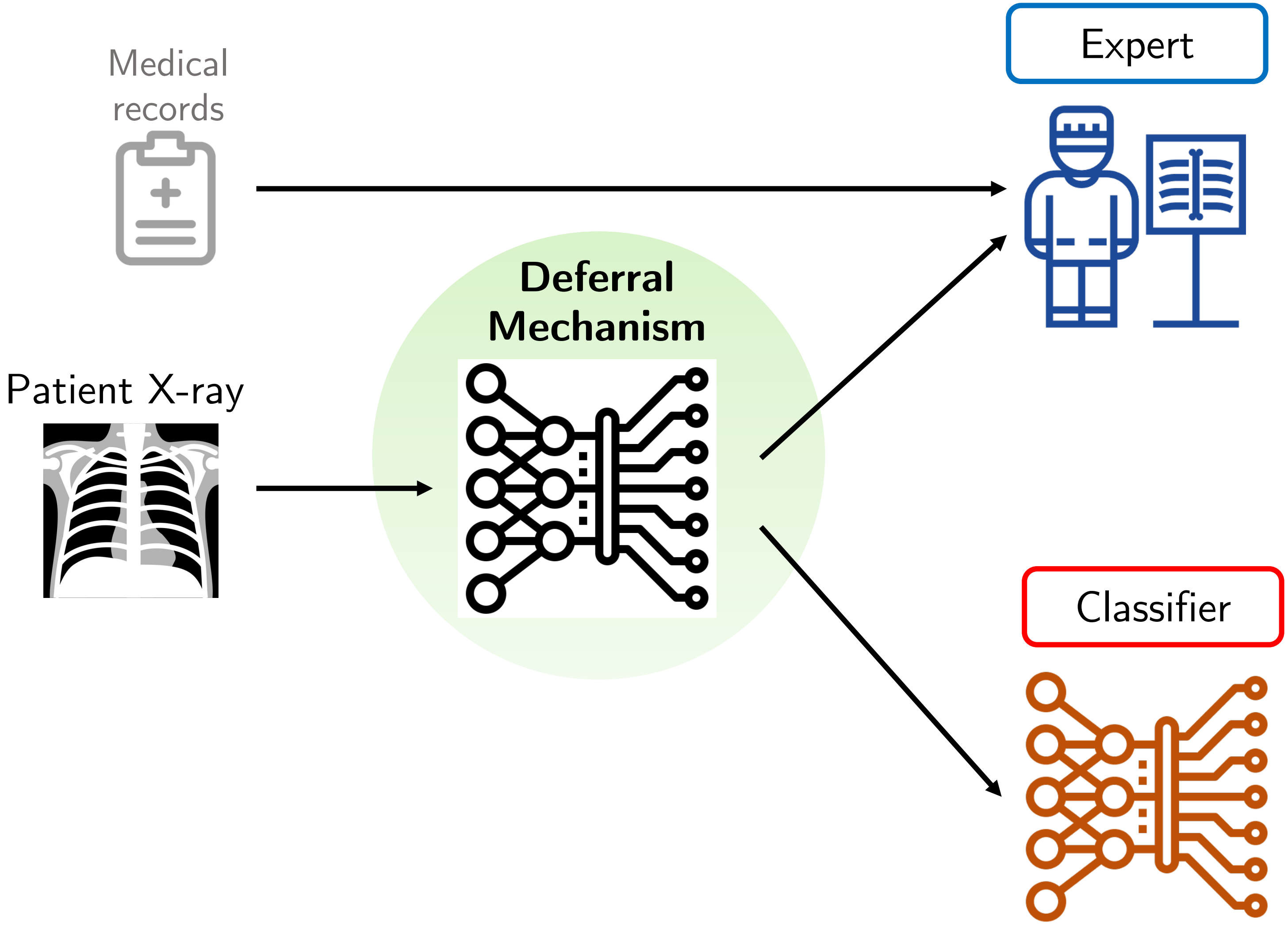
Human-AI Collaboration

Prevent critical misclassification by allocating decisions between the AI and human.



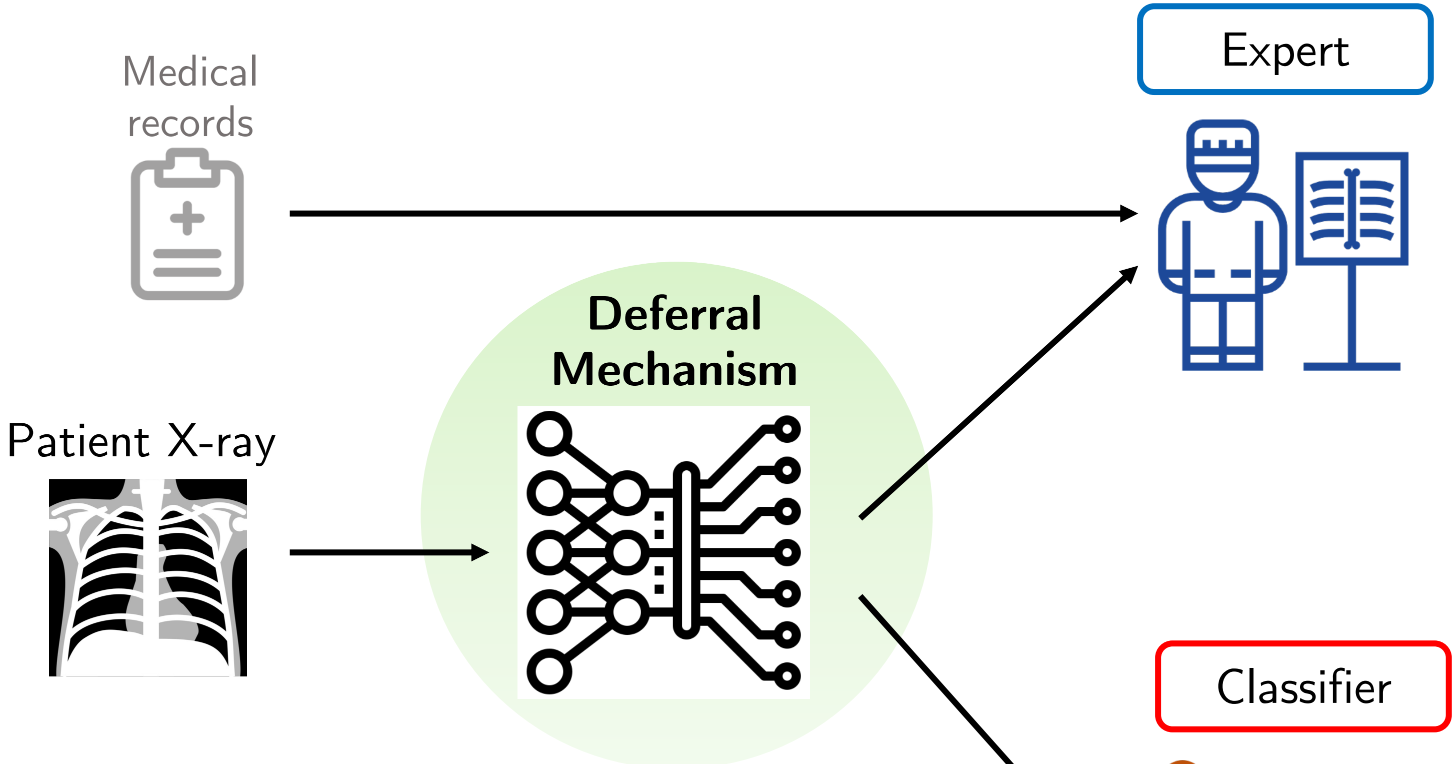
Rejection Learning

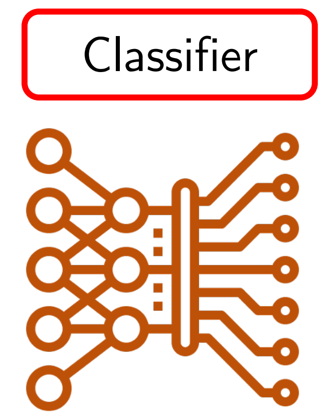
Q: How to determine which examples should be routed to the classifier or expert?



Rejection Learning

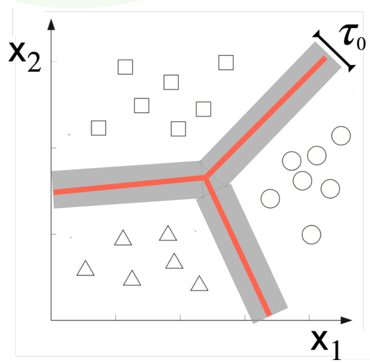
Q: How to determine which examples should be routed to the classifier or expert?





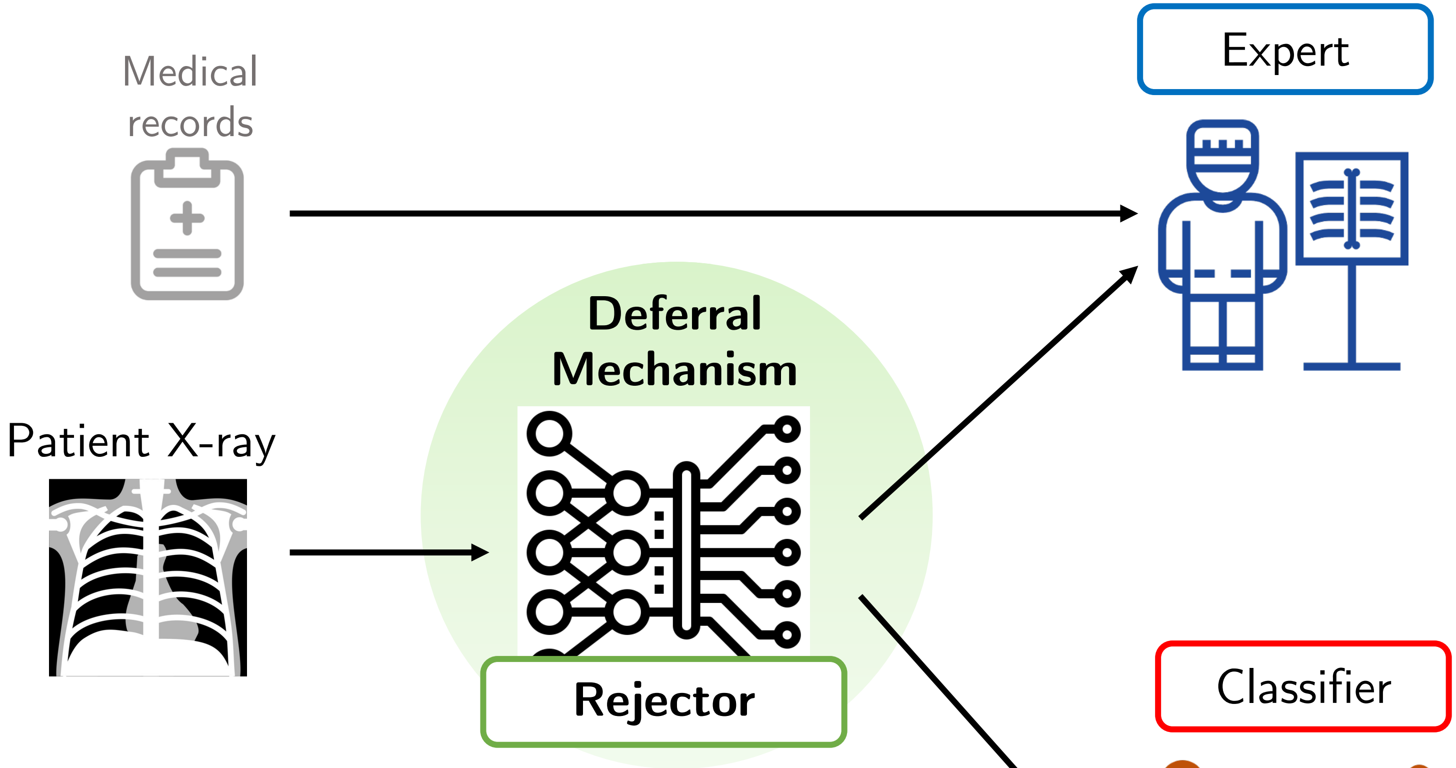
Thresholding **classifier confidence**

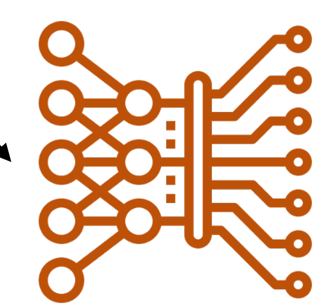
X Does not consider expert abilities



Rejection Learning

Q: How to determine which examples should be routed to the classifier or expert?





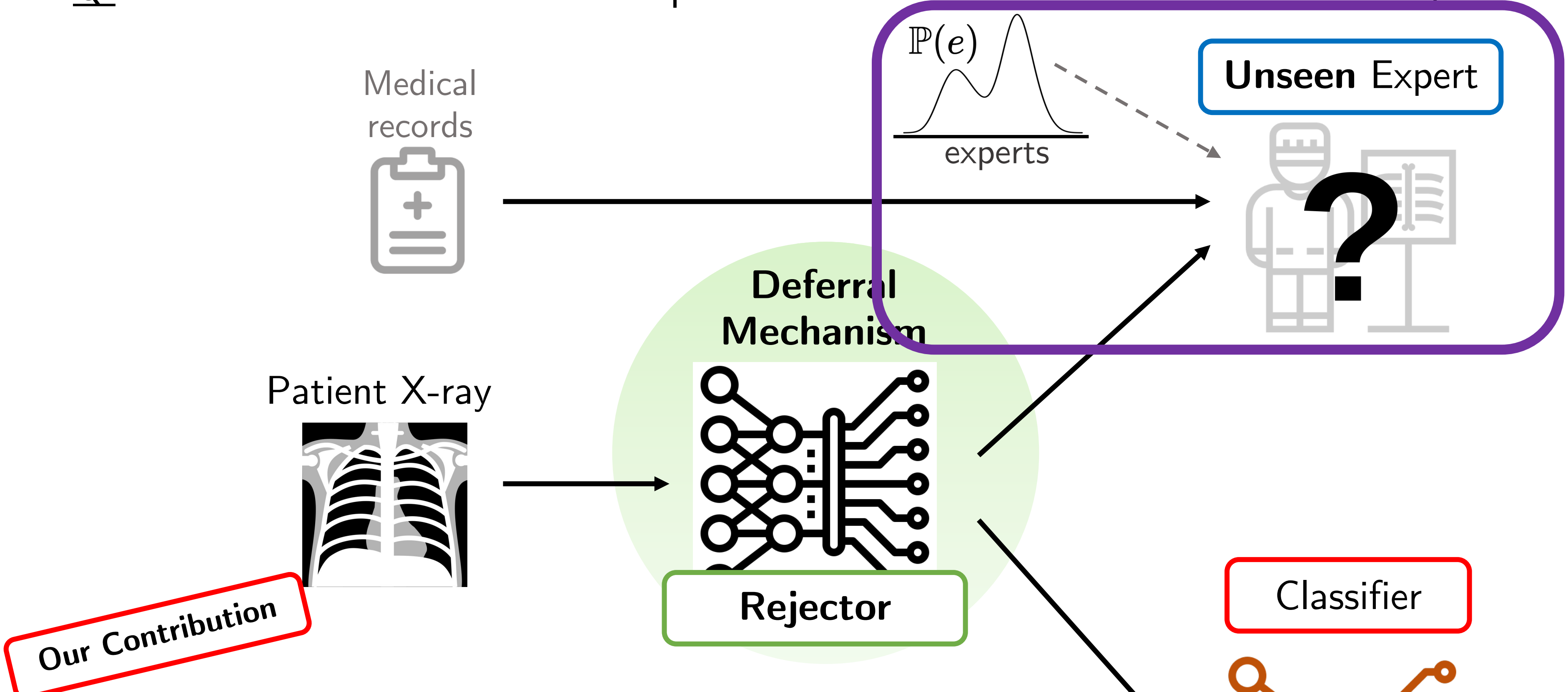
Learning to Defer: train classifier and **rejector** jointly

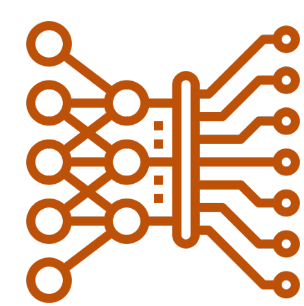
✓ Considers expert abilities & classifier adapts to expert

X Specialized for a given expert, unable to handle unseen experts

Rejection Learning

Q: How to determine which examples should be routed to the classifier or expert?





Learning to Defer to a Population

- ✓ Considers expert abilities & classifier adapts to expert
- ✓ Handles unseen experts at test-time (assuming experts drawn from a population)

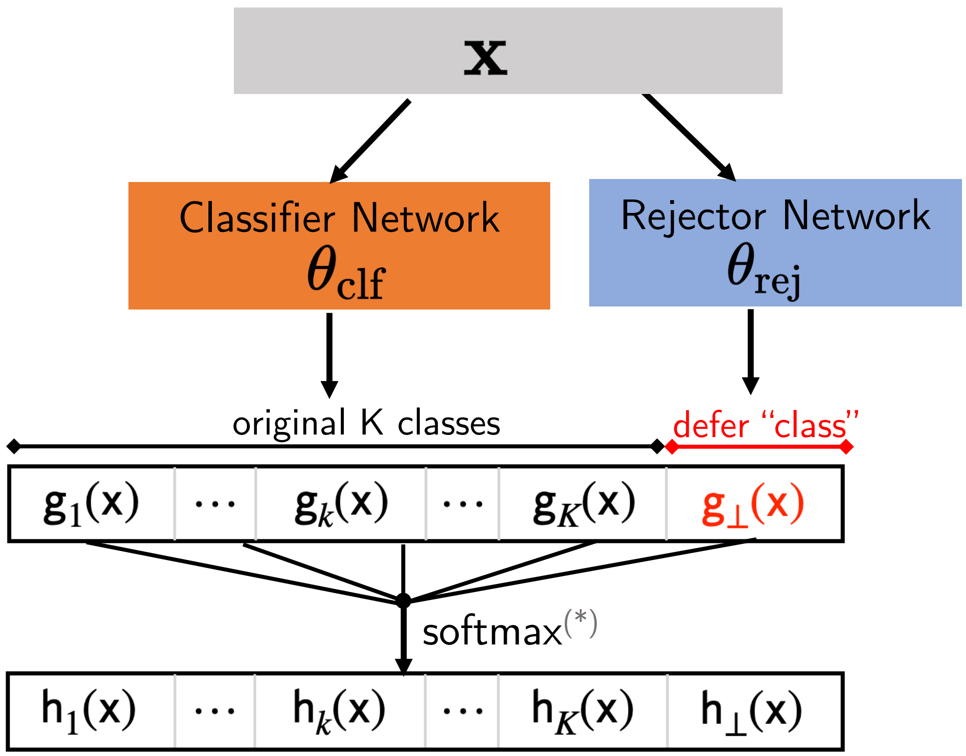
Learning to Defer (to a single expert)

expert
demonstrations

training data
$\mathcal{D} = \{\mathbf{x}_n, y_n, m_n\}_{n=1}^{N}$

Mozannar & Sontag. *Consistent estimators for learning to defer to an expert.* ICML, 2020.

Learning to Defer (to a single expert)



expert demonstrations

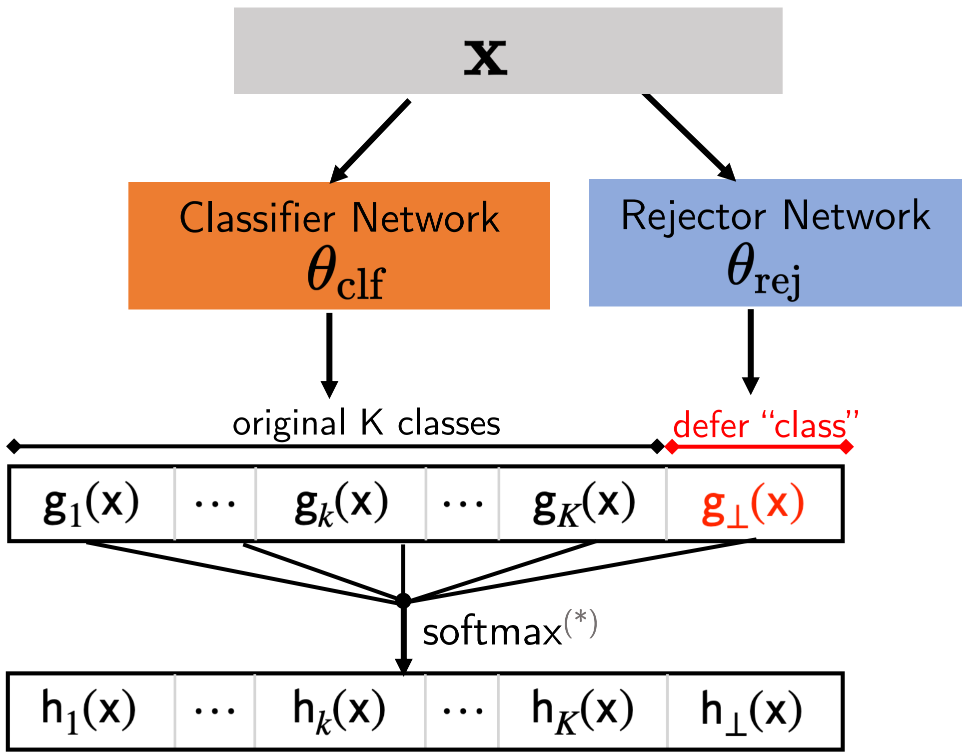
training data

$$\mathcal{D} = \{\mathbf{x}_n, y_n, m_n\}_{n=1}^{N}$$

(*) other parameterizations/surrogate losses possible

Mozannar & Sontag. *Consistent estimators for learning to defer to an expert.* ICML, 2020.

Learning to Defer (to a single expert)



expert demonstrations

training data

$$\mathcal{D} = \{\mathbf{x}_n, y_n, m_n\}_{n=1}^{N}$$

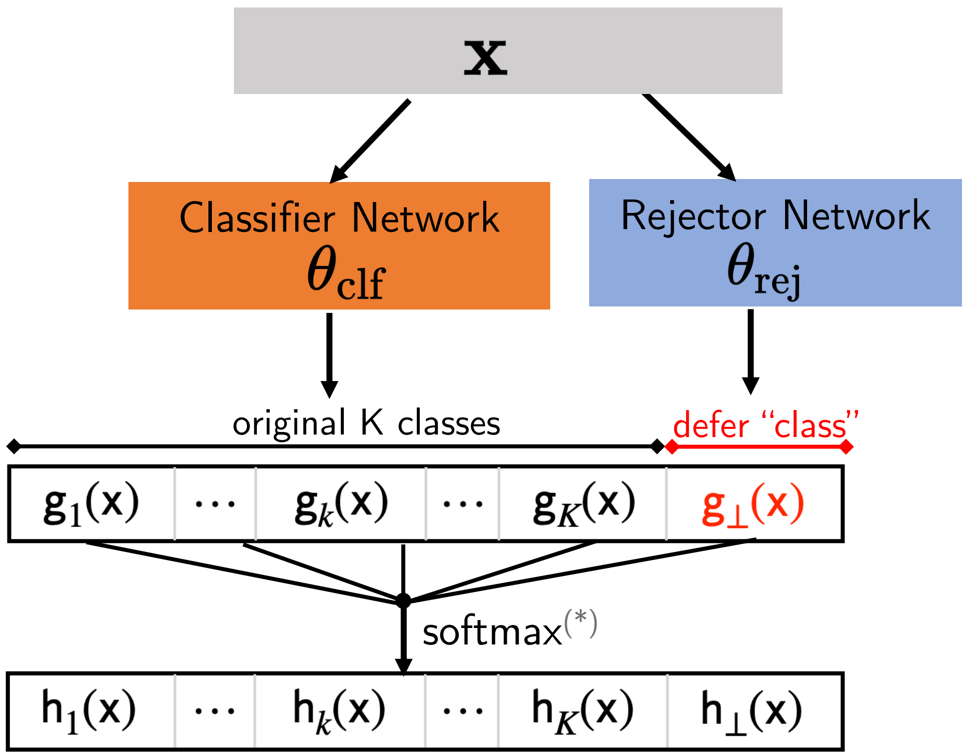
reject if

$$\mathbb{I}[g_\perp(\mathbf{x}) \geq \max_k g_k(\mathbf{x})]$$

(*) other parameterizations/surrogate losses possible

Mozannar & Sontag. *Consistent estimators for learning to defer to an expert.* ICML, 2020.

Learning to Defer (to a single expert)



expert demonstrations

training data
$$\mathcal{D} = \{\mathbf{x}_n, y_n, m_n\}_{n=1}^{N}$$

reject if
$$\mathbb{I}[g_\perp(\mathbf{x}) \geq \max_k g_k(\mathbf{x})]$$

cross-entropy loss(*)

$$\ell(\theta; \mathbf{x}, y, m) = -\log h_y(\mathbf{x}) - \underbrace{\mathbb{I}[y = m]}_{\text{incurred if expert correct}} \log h_\perp(\mathbf{x})$$

(*) other parameterizations/surrogate losses possible

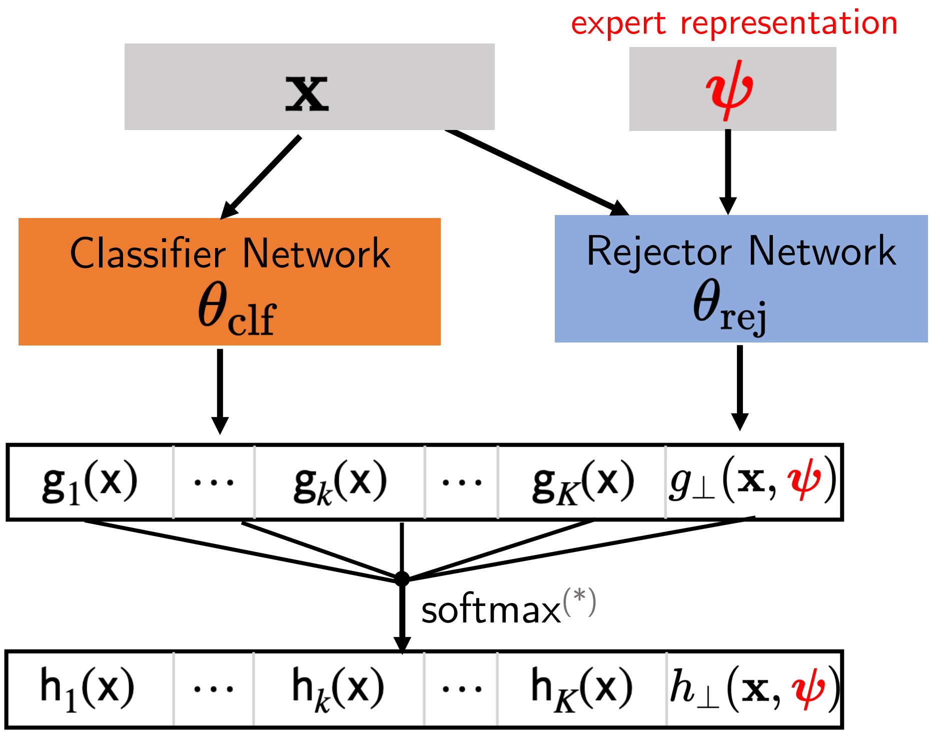
Mozannar & Sontag. *Consistent estimators for learning to defer to an expert.* ICML, 2020.

Learning to Defer to a Population

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \underbrace{\{m_{n,e}, \boldsymbol{\psi}_e\}_{e=1}^{E_n}}_{\text{multiple expert demonstrations \& representations}} \right\}_{n=1}^{N}$$

Learning to Defer to a Population



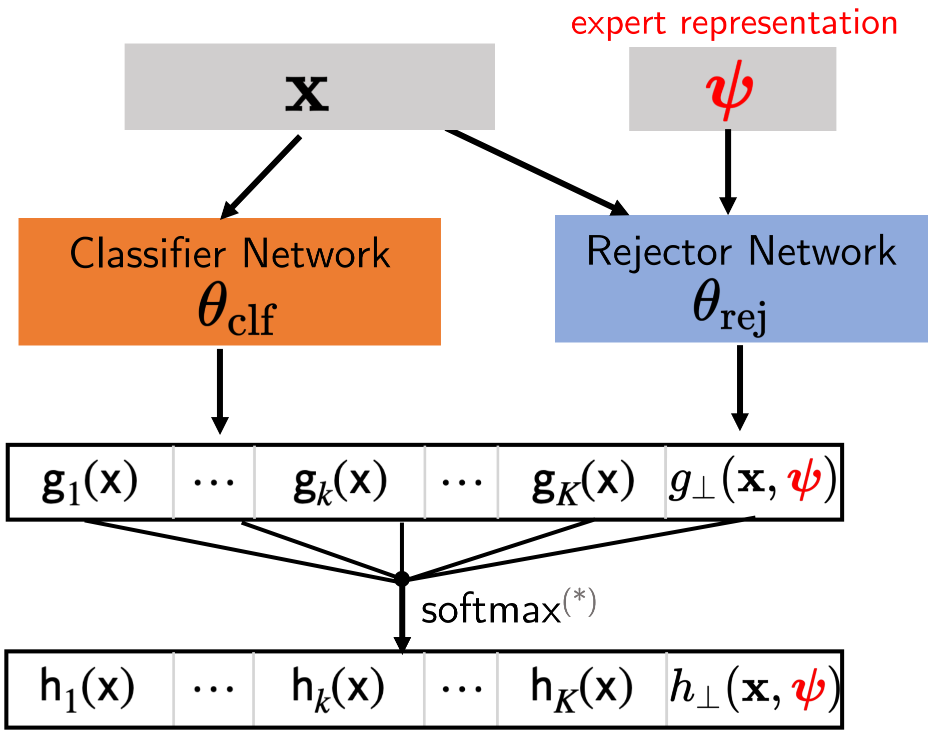
training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \underline{\{m_{n,e}, \boldsymbol{\psi}_e\}_{e=1}^{E_n}} \right\}_{n=1}^{N}$$

multiple expert demonstrations & representations

(*) other parameterizations/surrogate losses possible

Learning to Defer to a Population



training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \underbrace{\{m_{n,e}, \boldsymbol{\psi}_e\}_{e=1}^{E_n}}_{} \right\}_{n=1}^{N}$$

multiple expert demonstrations & representations

cross-entropy loss(*)

$$\ell(\theta; \mathbf{x}, y, \{m_e, \boldsymbol{\psi}_e\}_{e=1}^{E}) = \sum_{e=1}^{E} -\log h_y^{(e)}(\mathbf{x}) - \mathbb{I}[y = m_e] \log h_\perp^{(e)}(\mathbf{x}, \boldsymbol{\psi}_\mathbf{e})$$

(*) other parameterizations/surrogate losses possible

Meta-Learning to Defer

representative set of
expert demonstrations /
"context set"

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \{m_{n,e}, \mathcal{D}_e\}_{e=1}^{E_n} \right\}_{n=1}^{N}$$

Meta-Learning to Defer

representative set of expert demonstrations / "context set"

training data

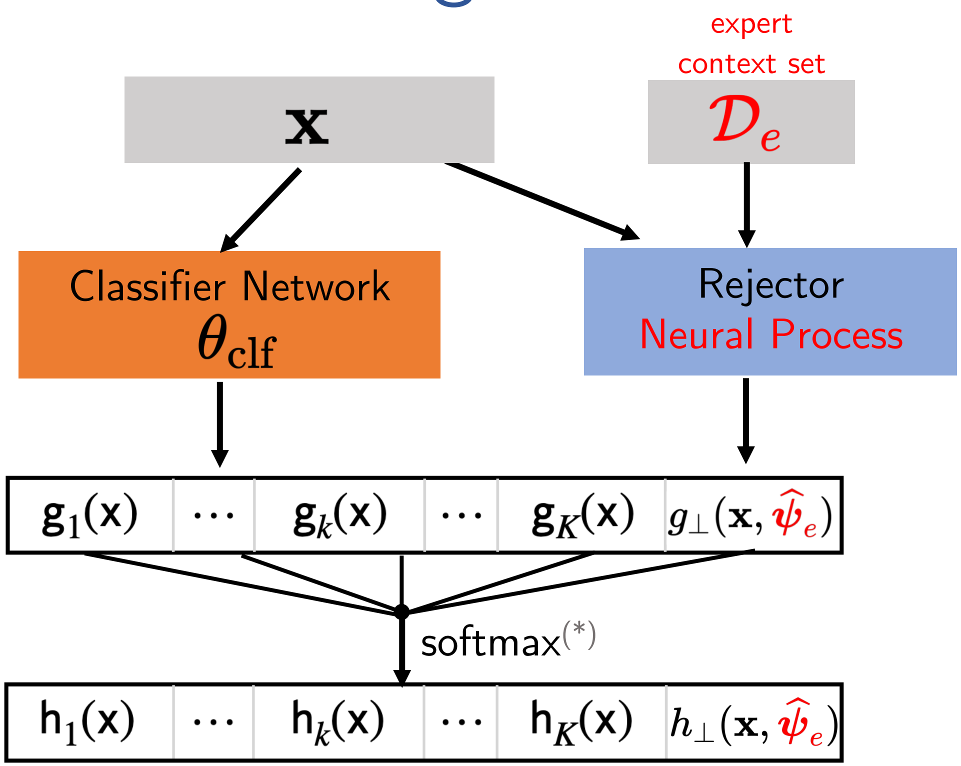
$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \{m_{n,e}, \mathcal{D}_e\}_{e=1}^{E_n} \right\}_{n=1}^{N}$$

Amortized expert representation as set embedding

$$\begin{aligned} \widehat{\boldsymbol{\psi}}_e &= \boldsymbol{\psi}(\mathcal{D}_e) \\ &= \rho\left(\sum_{b=1}^{B} f_\theta(\mathbf{x}_{e,b}, y_{e,b}, m_{e,b}) \right) \end{aligned}$$

Zaheer et. al. *Deep Sets.* NeurIPS, 2017.

Meta-Learning to Defer



representative set of
expert demonstrations /
"context set"

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \{m_{n,e}, \mathcal{D}_e\}_{e=1}^{E_n} \right\}_{n=1}^{N}$$

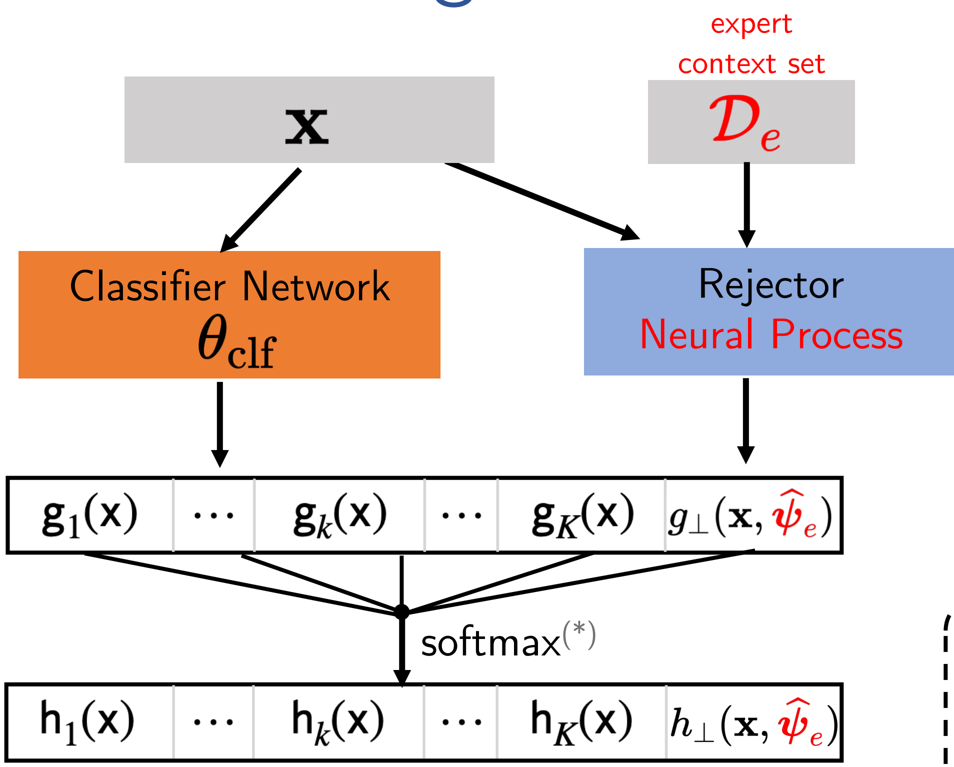
Amortized expert representation as set embedding

$$\widehat{\boldsymbol{\psi}}_e = \boldsymbol{\psi}(\mathcal{D}_e)$$

$$= \rho\left(\sum_{b=1}^{B} f_\theta(\mathbf{x}_{e,b}, y_{e,b}, m_{e,b})\right)$$

Zaheer et. al. *Deep Sets.* NeurIPS, 2017.
Garnelo et. al. *Conditional Neural Processes.* ICML, 2018.

Meta-Learning to Defer



representative set of expert demonstrations / "context set"

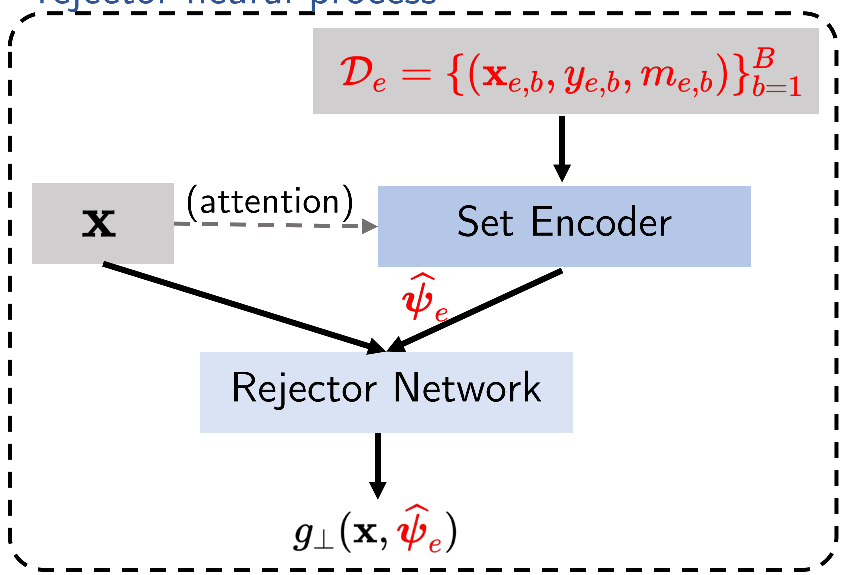
training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \{m_{n,e}, \mathcal{D}_e\}_{e=1}^{E_n} \right\}_{n=1}^{N}$$

Amortized expert representation as set embedding

$$\widehat{\psi}_e = \psi(\mathcal{D}_e) = \rho\left(\sum_{b=1}^{B} f_\theta(\mathbf{x}_{e,b}, y_{e,b}, m_{e,b}) \right)$$

rejector neural process



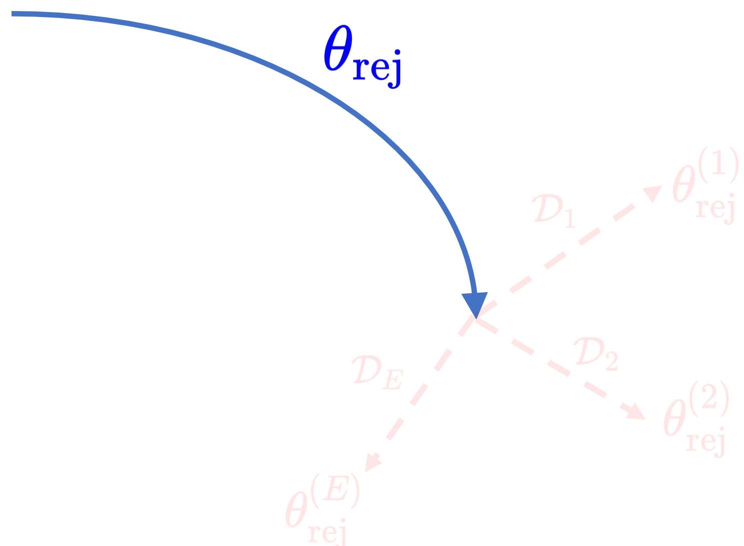
Zaheer et. al. *Deep Sets.* NeurIPS, 2017.
Garnelo et. al. *Conditional Neural Processes.* ICML, 2018.

Fine-tuning from marginal expert

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \{m_{n,e}, \mathcal{D}_e\}_{e=1}^{E_n} \right\}_{n=1}^{N}$$

representative set of expert demonstrations / "context set"



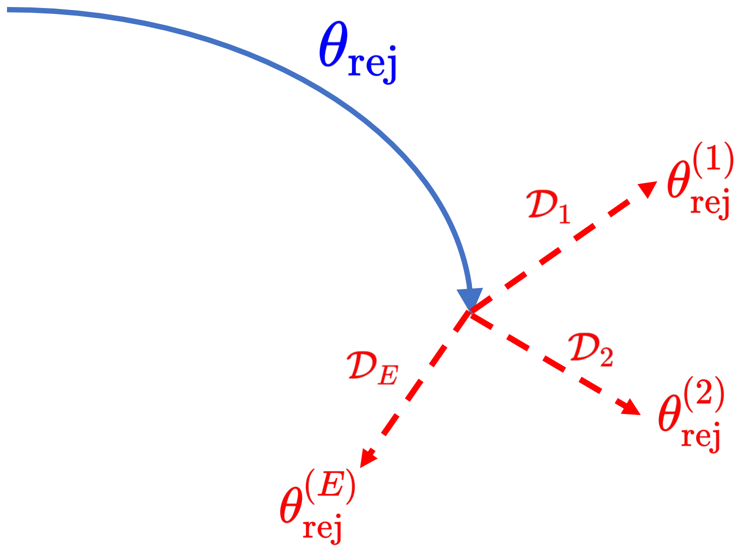
1. Model the *marginal* expert by the single-expert formulation

Fine-tuning from marginal expert

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, y_n, \{m_{n,e}, \mathcal{D}_e\}_{e=1}^{E_n} \right\}_{n=1}^{N}$$

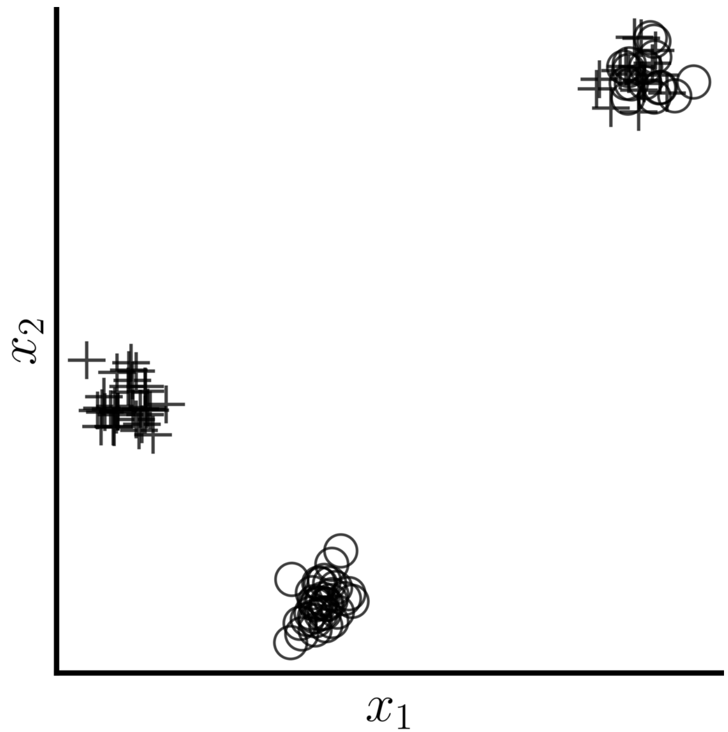
representative set of expert demonstrations / "context set"



1. Model the *marginal* expert by the single-expert formulation
2. Then finetune on expert context set at test-time

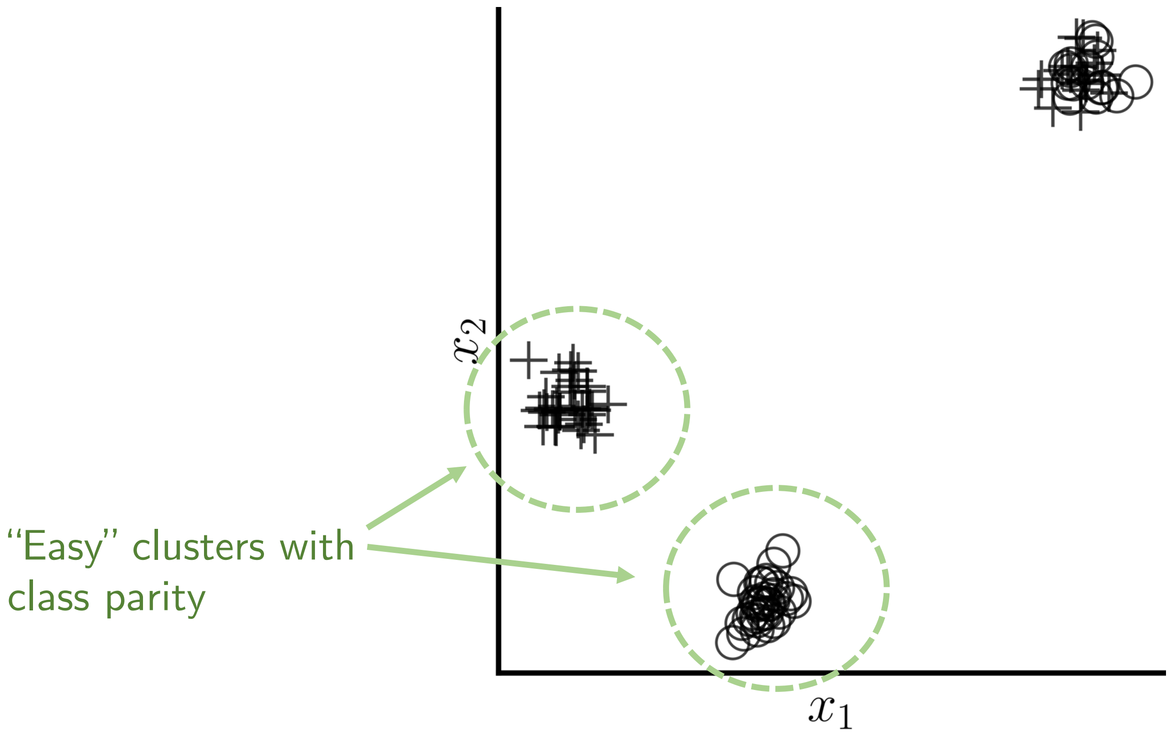
Experiment: Synthetic data

+ : class 0 O : class 1

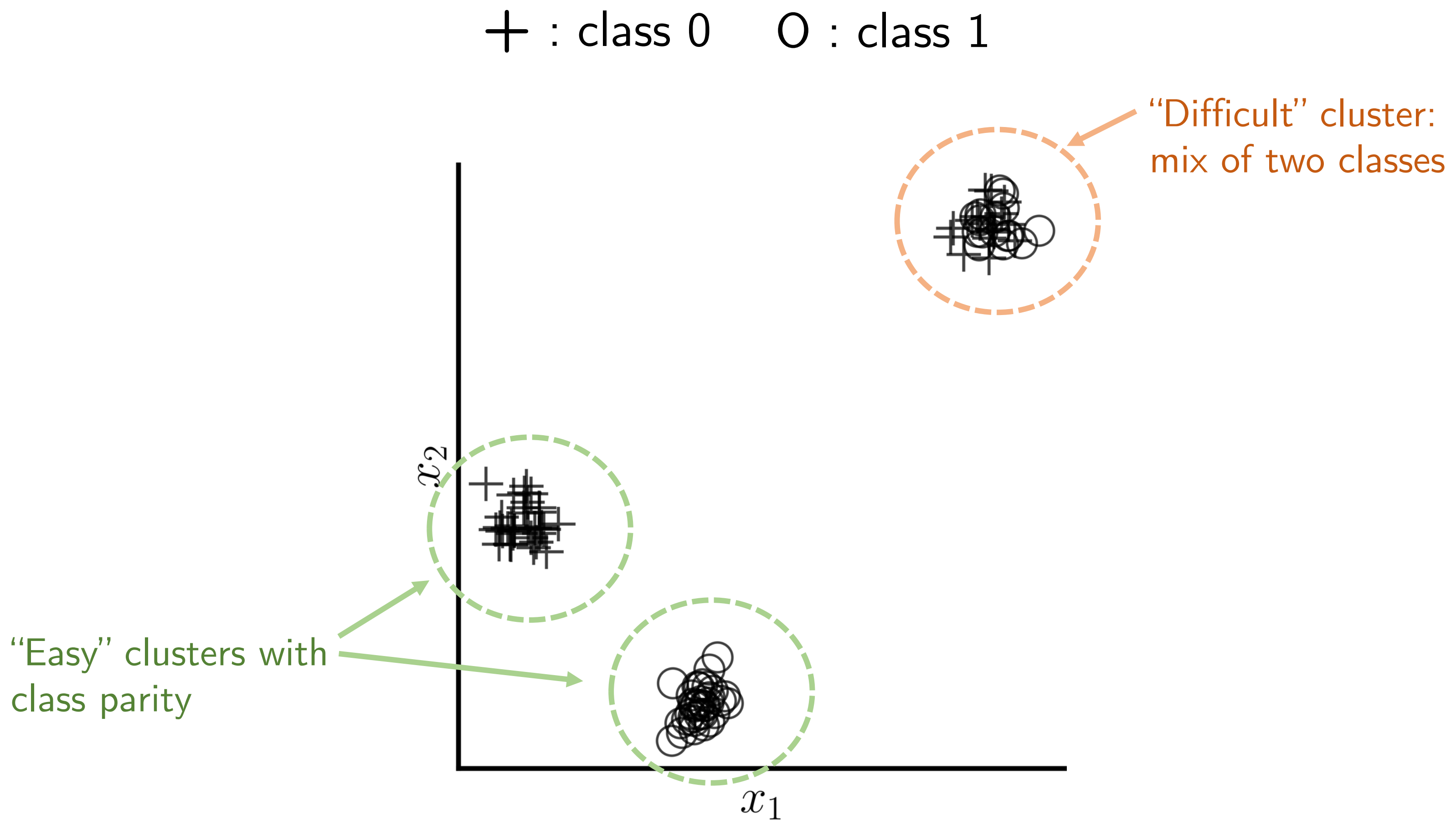


Experiment: Synthetic data

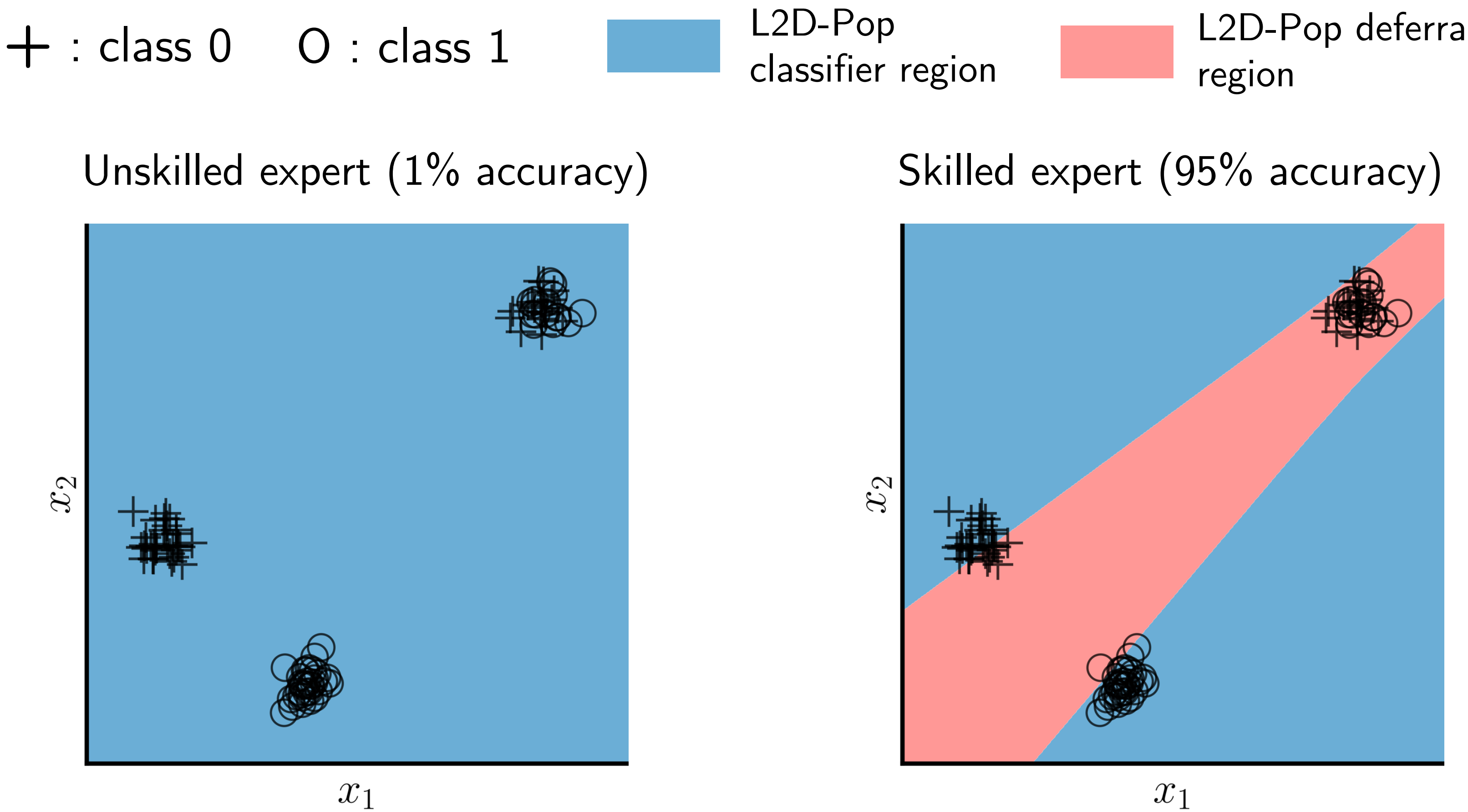
+ : class 0 O : class 1



Experiment: Synthetic data

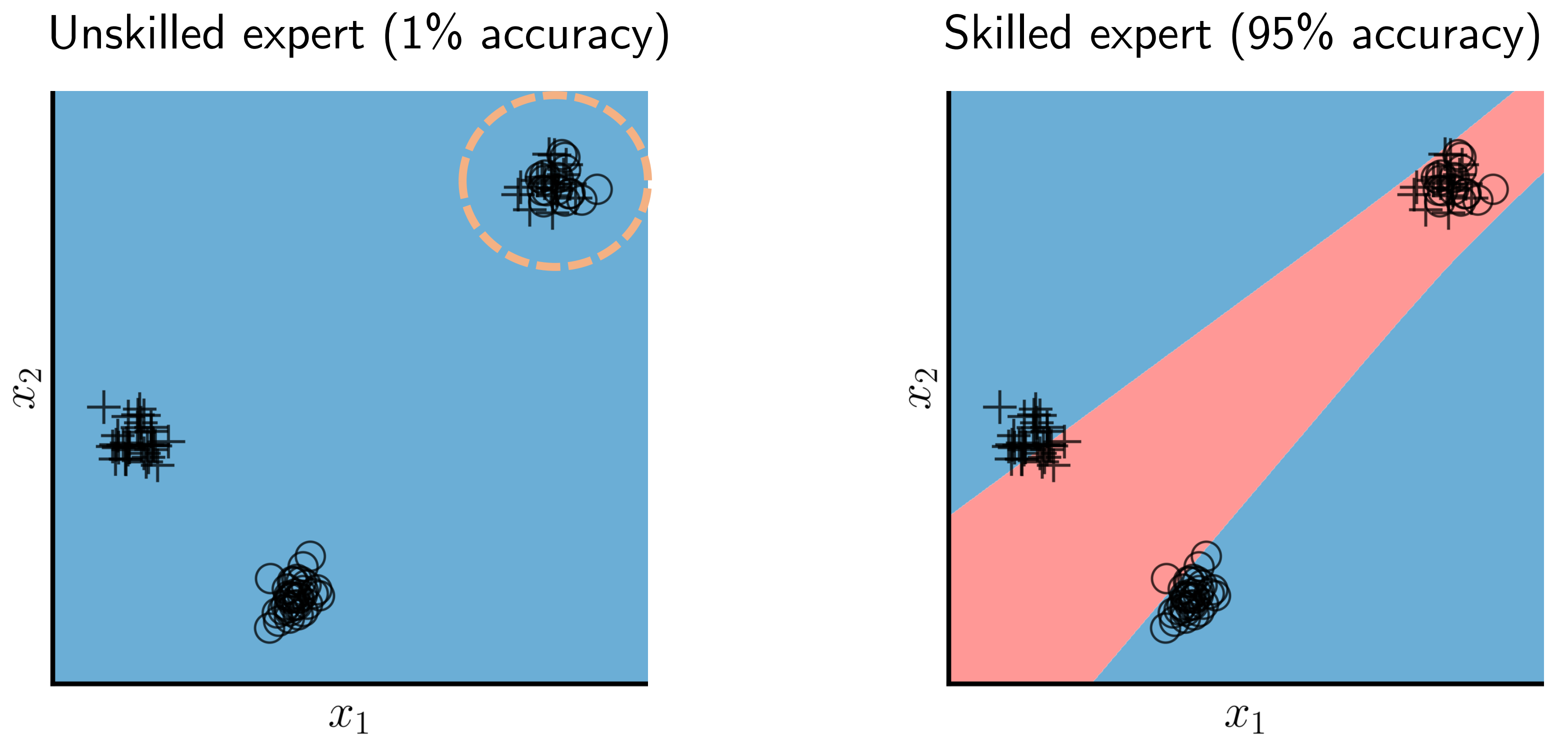


Experiment: Synthetic data



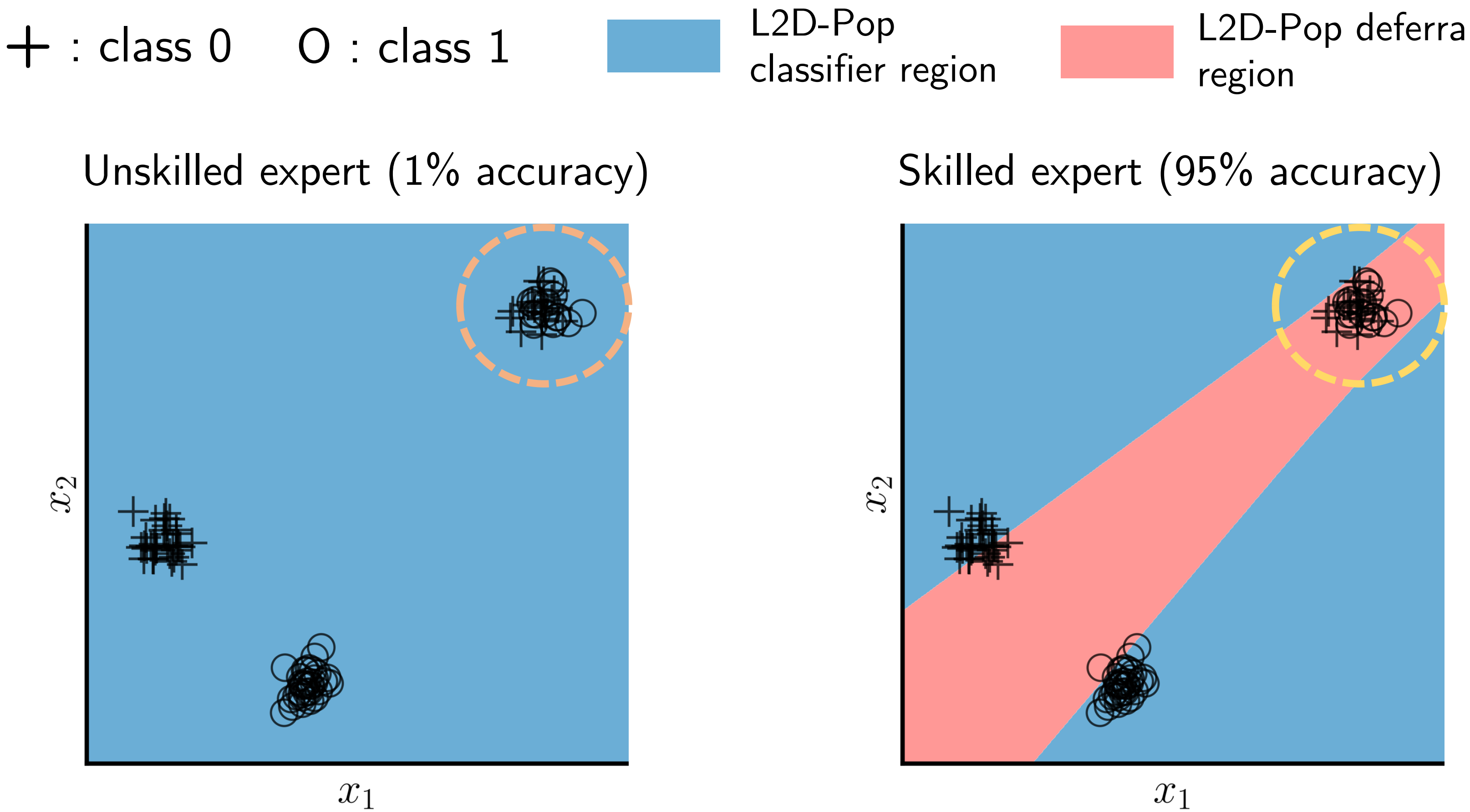
Experiment: Synthetic data

+ : class 0   O : class 1   L2D-Pop classifier region   L2D-Pop deferral region



L2D-Pop (adaptive) ✓ Doesn't defer when the expert is poor

Experiment: Synthetic data



L2D-Pop (adaptive)

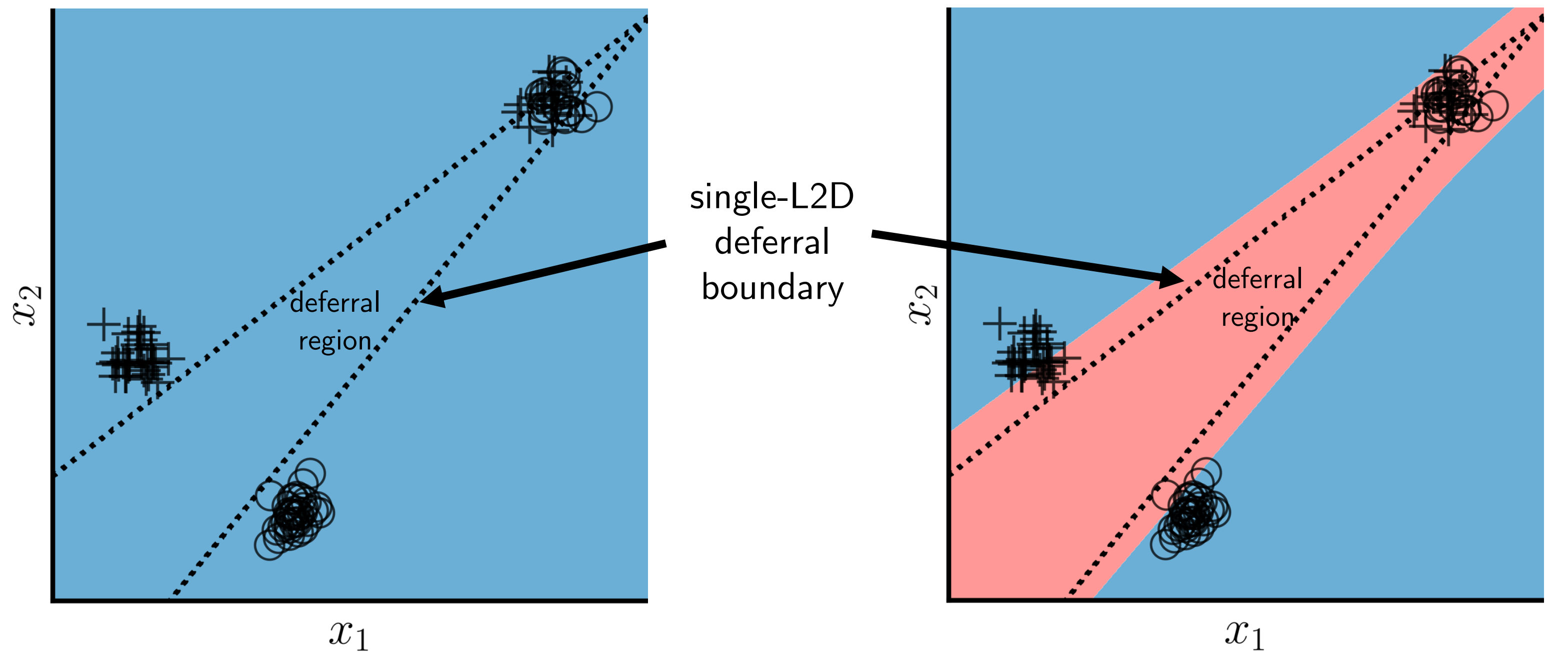
✓ Doesn't defer when the expert is poor

✓ Defers whole of difficult cluster when expert is good

Experiment: Synthetic data

+ : class 0   O : class 1   L2D-Pop classifier region   L2D-Pop deferral region

Unskilled expert (1% accuracy)   Skilled expert (95% accuracy)

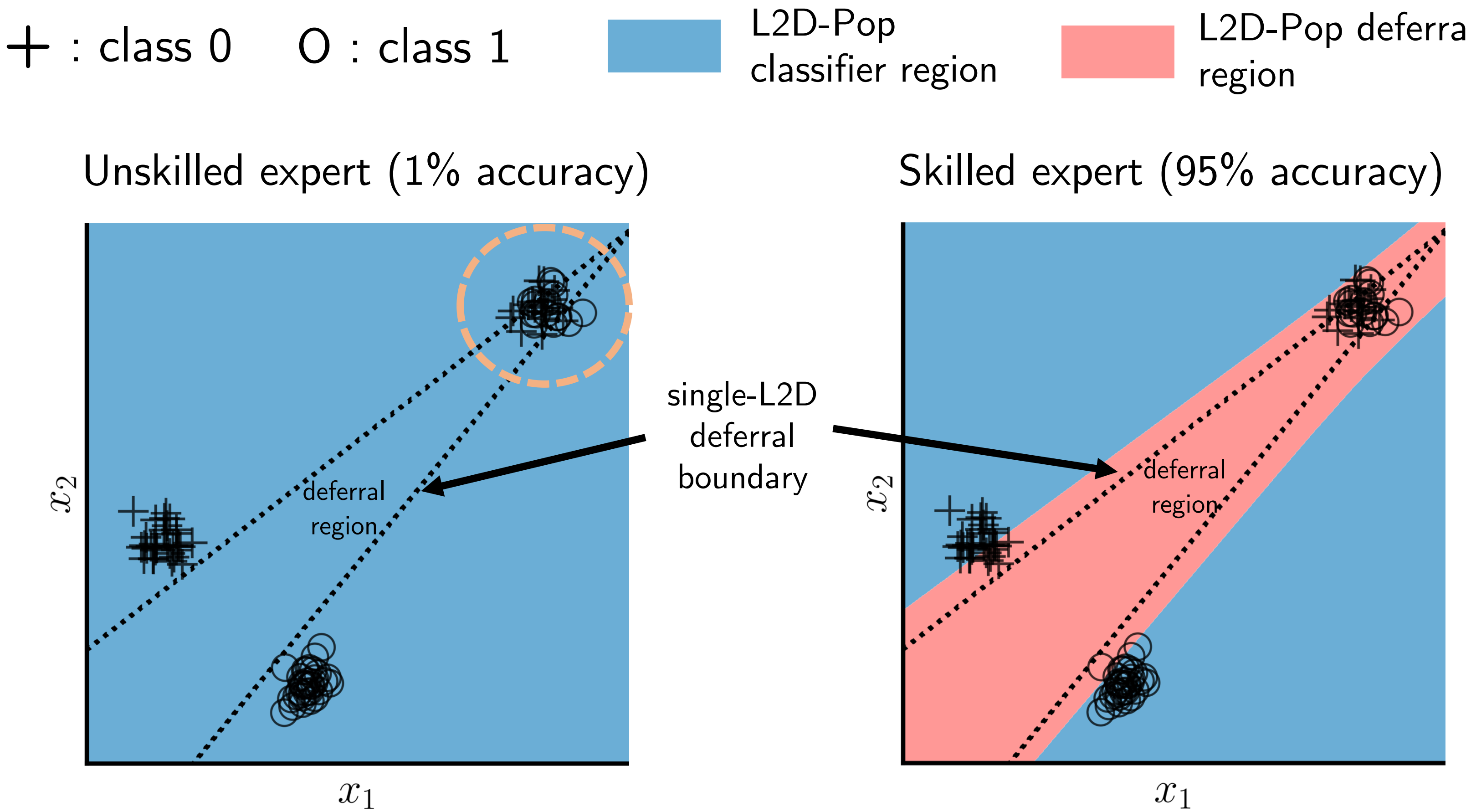


L2D-Pop (adaptive)

✓ Doesn't defer when the expert is poor

✓ Defers whole of difficult cluster when expert is good

Experiment: Synthetic data

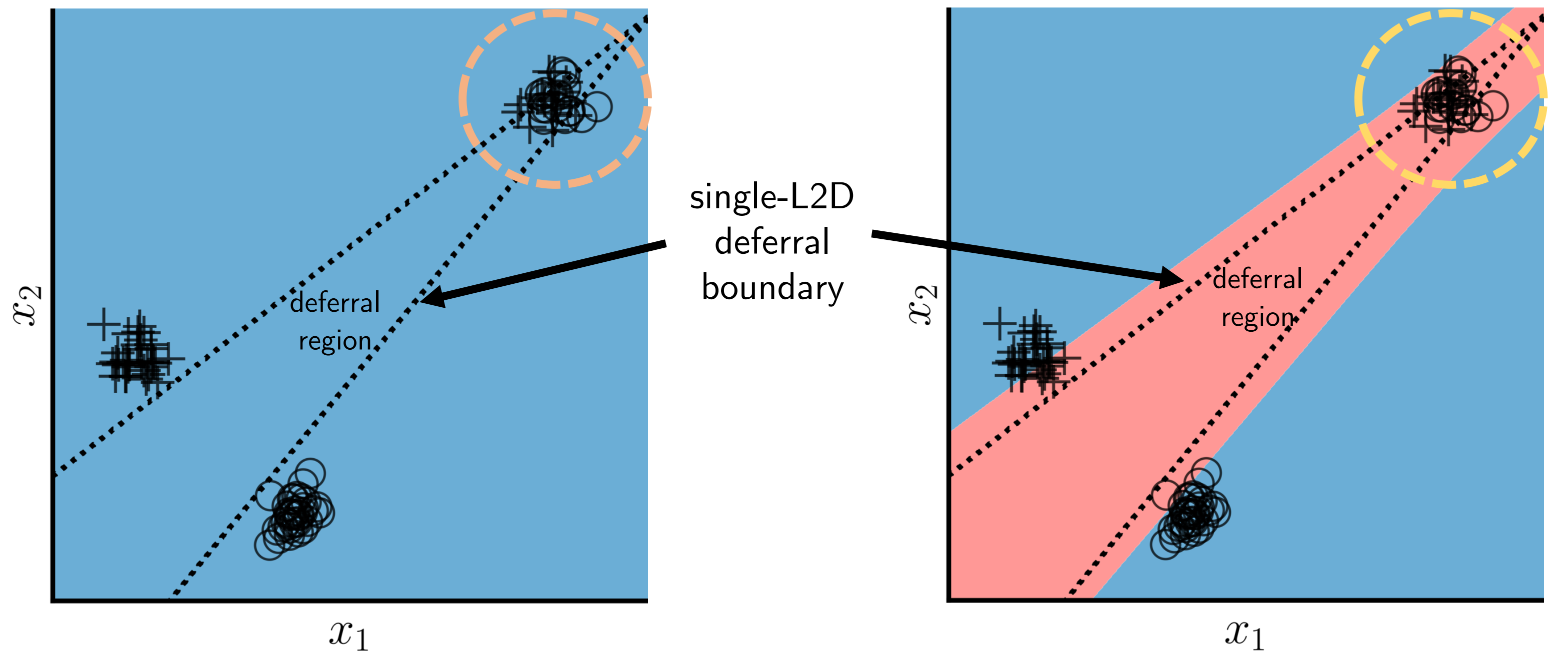


L2D-Pop (adaptive)	✓ Doesn't defer when the expert is poor	✓ Defers whole of difficult cluster when expert is good
single-L2D (constant)	X Over-defers as expert does worse than random on difficult cluster	

Experiment: Synthetic data

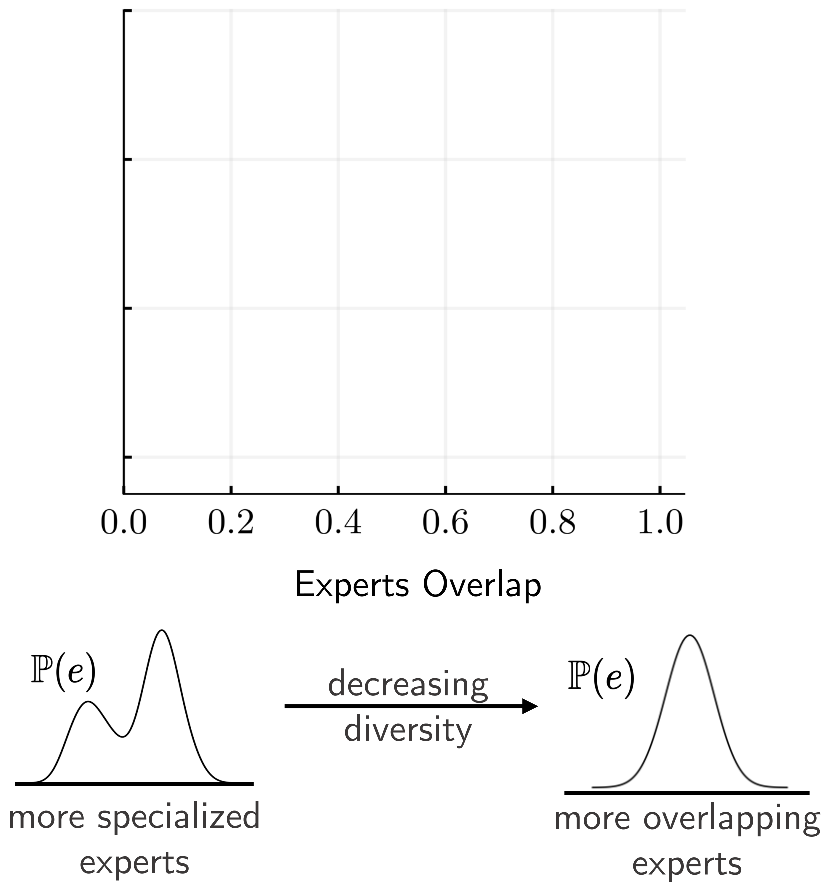
+ : class 0   O : class 1   L2D-Pop classifier region   L2D-Pop deferral region

Unskilled expert (1% accuracy)   Skilled expert (95% accuracy)



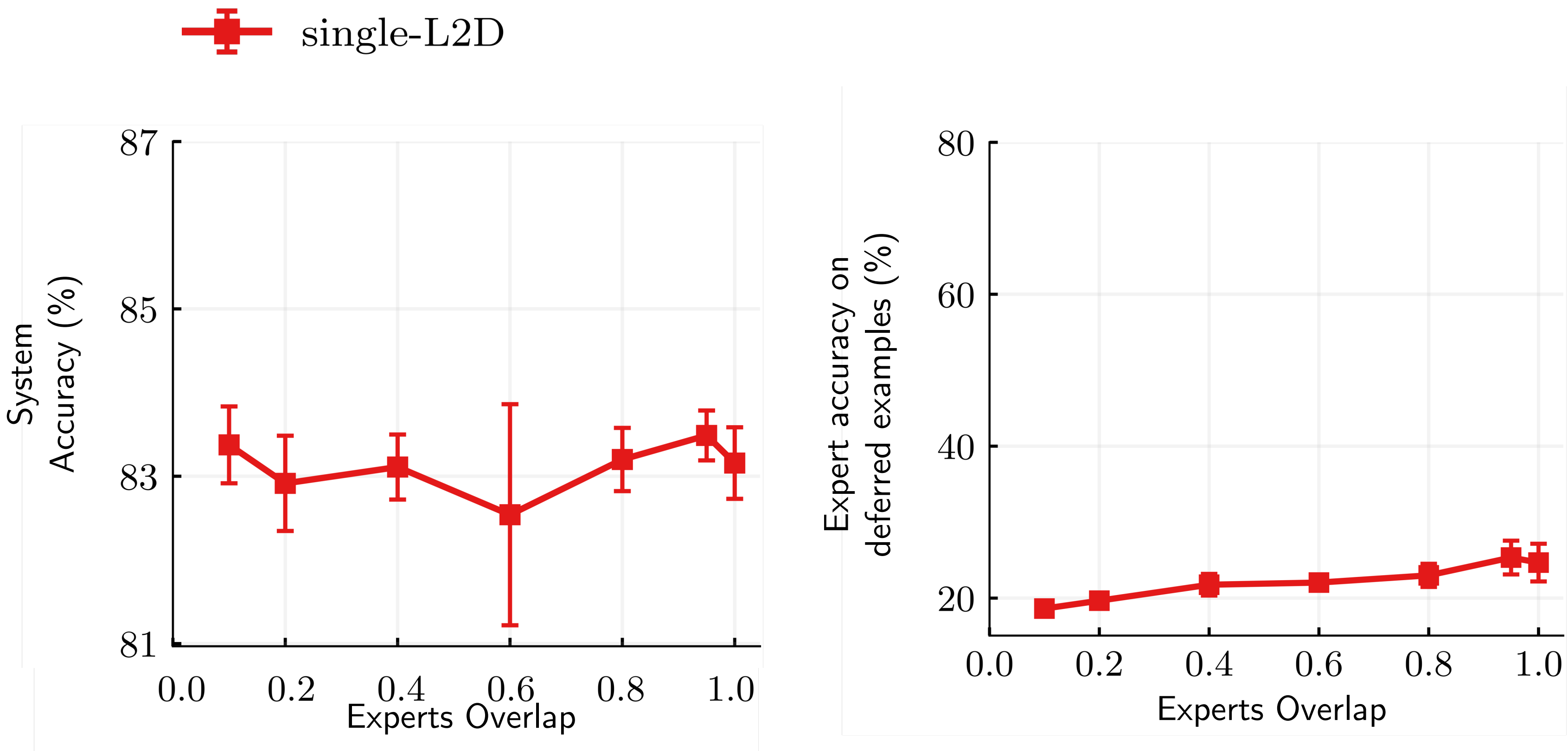
	Unskilled expert	Skilled expert
L2D-Pop (adaptive)	✓ Doesn't defer when the expert is poor	✓ Defers whole of difficult cluster when expert is good
single-L2D (constant)	X Over-defers as expert does worse than random on difficult cluster	X Under-defers as classifier only has random chance of being correct on difficult cluster

Experiments: Varying Population Diversity



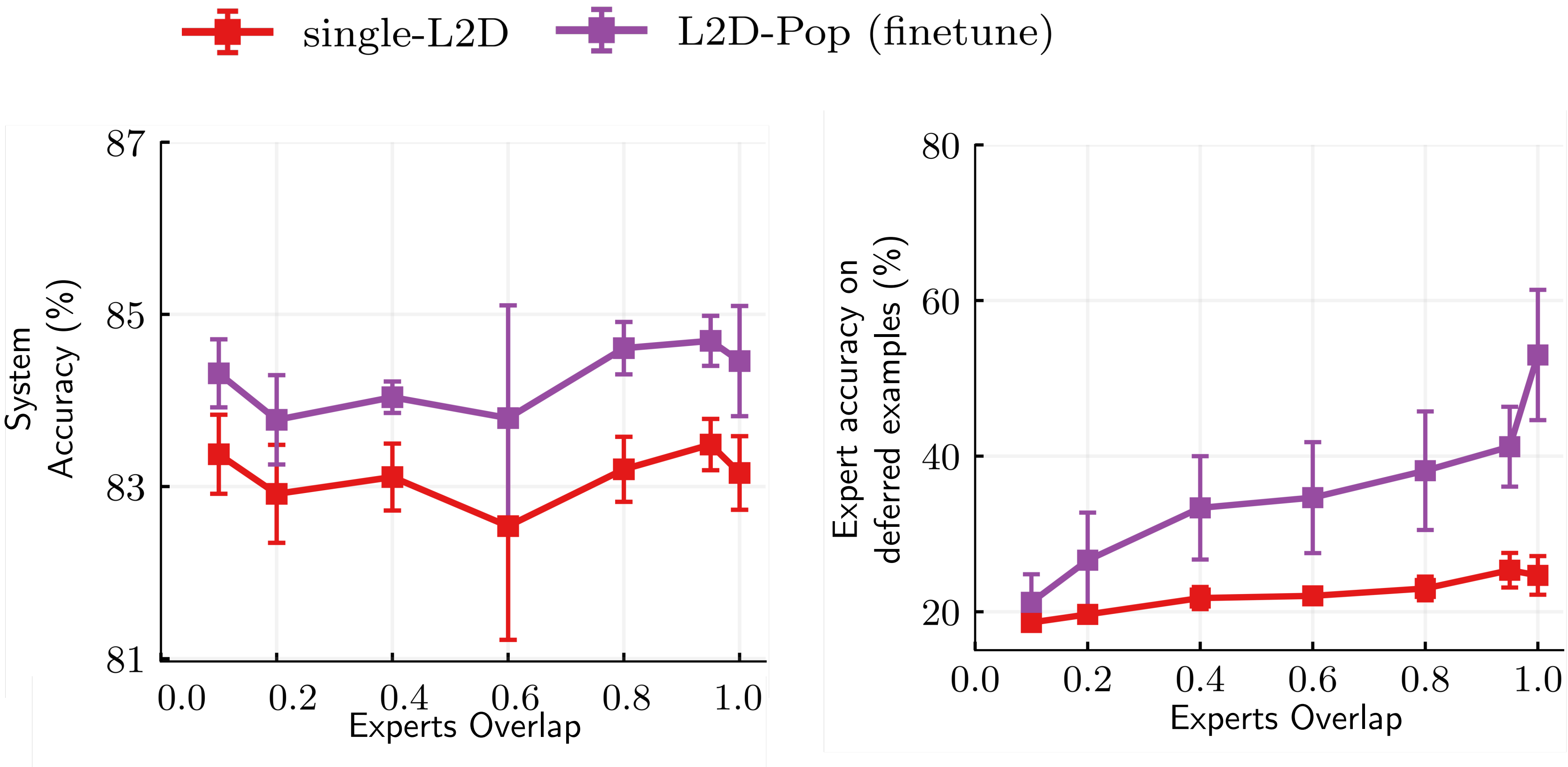
Experiments: Varying Population Diversity

CIFAR-20 results



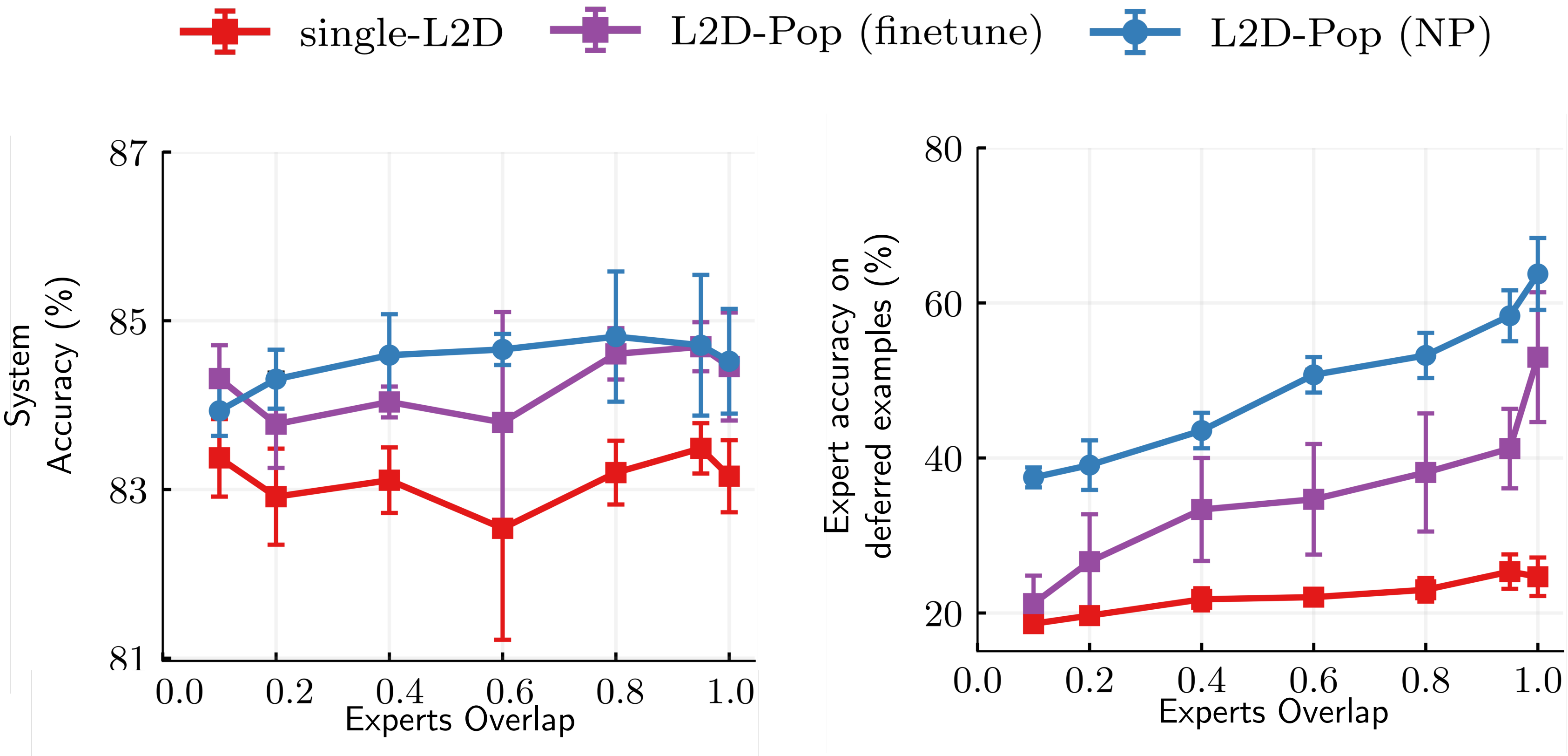
Experiments: Varying Population Diversity

CIFAR-20 results



Experiments: Varying Population Diversity

CIFAR-20 results



L2D-Pop is superior at deferring as shown by the expert accuracy on deferred examples (right) leading to a boost in system accuracy (left). The improvement is greater for the Neural Process implementation.

Further Experiments and Results

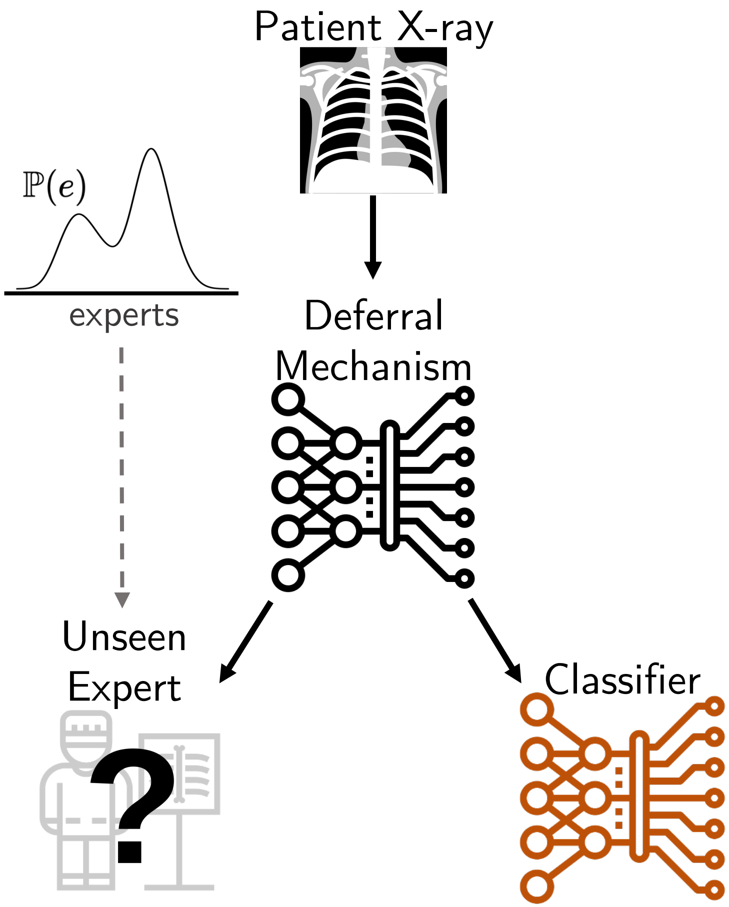
- **Further experiments in paper**
  - Additional benchmark problems: traffic sign detection and skin lesion diagnosis
  - Using OvA surrogate

- **Consistency of softmax and OvA surrogate loss functions** for L2D-Pop

- **Attentive neural process** implementation of L2D-Pop

- **Model-agnostic meta-learning (MAML)** implementation of L2D-Pop

Learning to Defer to a Population: A Meta-Learning Approach



Thank you!

Amsterdam
Machine Learning Lab