Learning to Help in Multi-Class Settings

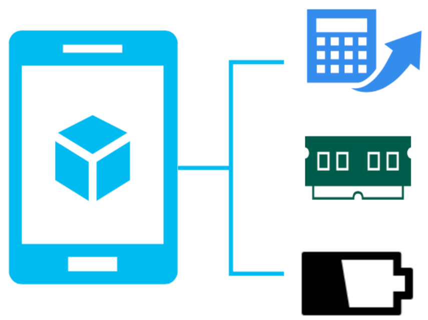
Yu Wu[§], Yansong Li[‡], Zeyu Dong[†], Nitya Sathyavageeswaran[§], Anand D. Sarwate[§]
[§]Rutgers University, [‡]University of Illinois Chicago, [†]Stony Brook University

The Thirteenth International Conference on Learning Representations (ICLR 2025)

Motivation

Onboard Machine Learning (ML) models often face significant limitations

in terms of computational resources and re-trainability.



- Local devices are typically constrained by limited processing power, memory, and battery life *(Ajani et al., 2021; Biglari & Tang, 2023)*.

- Once a local model is deployed, it may be difficult to retrain or update *(Hanzlik et al., 2021).*

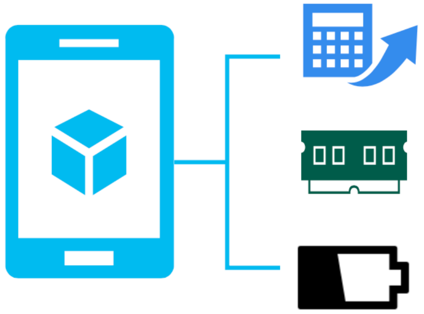
Motivation

Onboard Machine Learning (ML) models often face significant limitations

in terms of computational resources and re-trainability.

- Local devices are typically constrained by limited processing power, memory, and battery life *(Ajani et al., 2021; Biglari & Tang, 2023)*.

  Only small models can be deployed on mobile or embedded device.

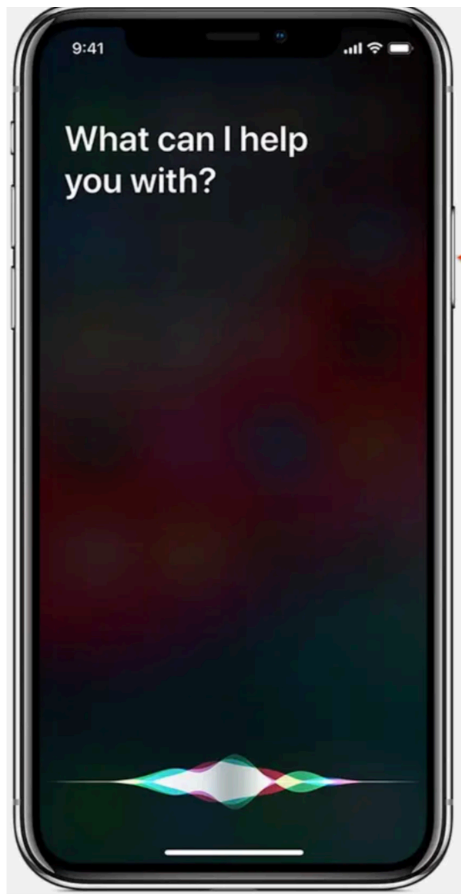


- Once a local model is deployed, it may be difficult to retrain or update *(Hanzlik et al., 2021).*

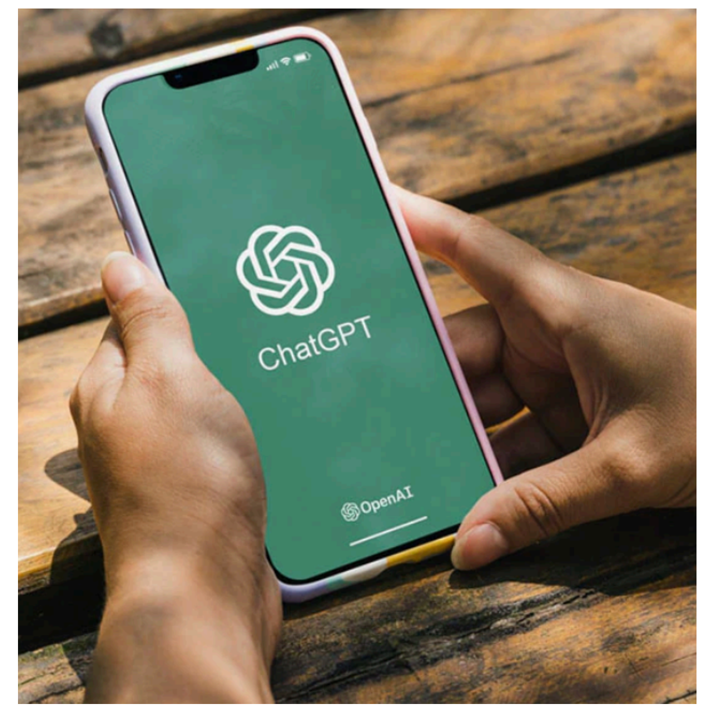
  Degradation in performance over time when data distribution drifts *(Lu et al., 2019)*.

Potential Solution

- Augment the local learning system (the "client") with an external model hosted on a remote server.

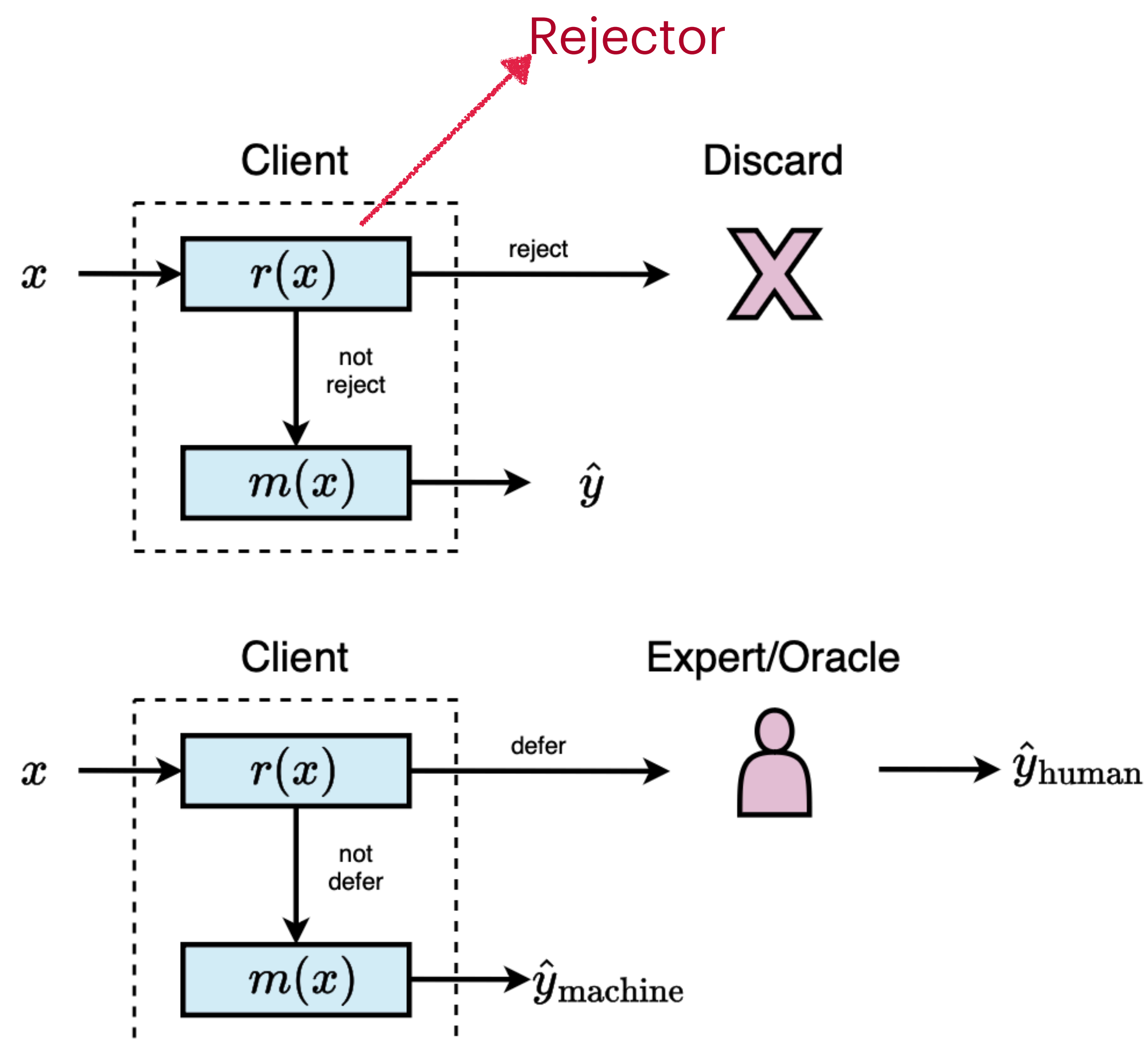


Through Internet



Apple Intelligence: A example of client-server

Existing works



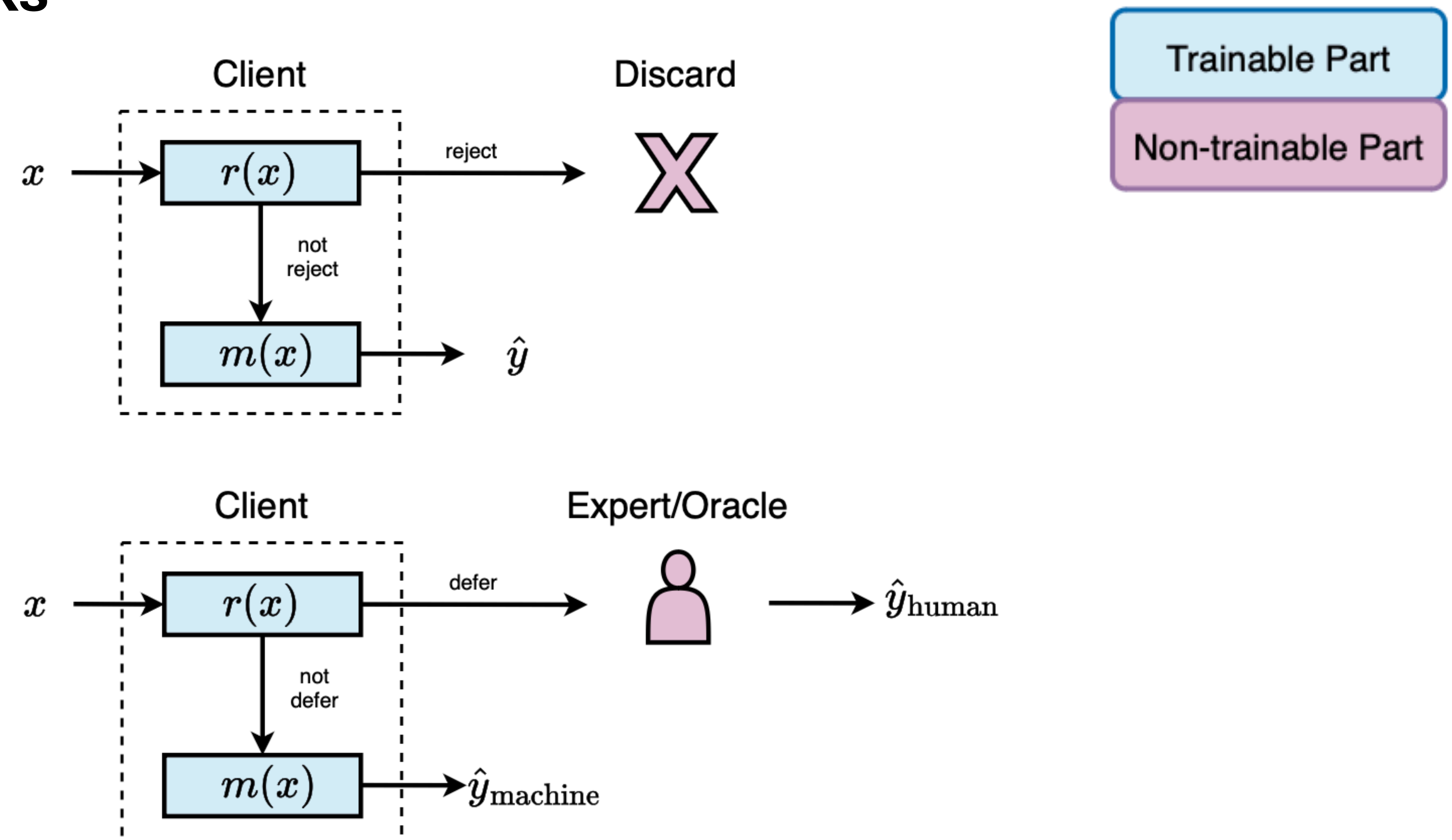
Learning with Abstention (LWA) : discard uncertain inference

Cortes, C., DeSalvo, G., & Mohri, M. (2016). Learning with rejection. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27* (pp. 67-82). Springer International Publishing.

Learning to Defer (L2D): deferring to existing human or machine experts

Madras, D., Pitassi, T., & Zemel, R. (2018). Predict responsibly: improving fairness and accuracy by learning to defer. *Advances in neural information processing systems*, *31*.

Existing works

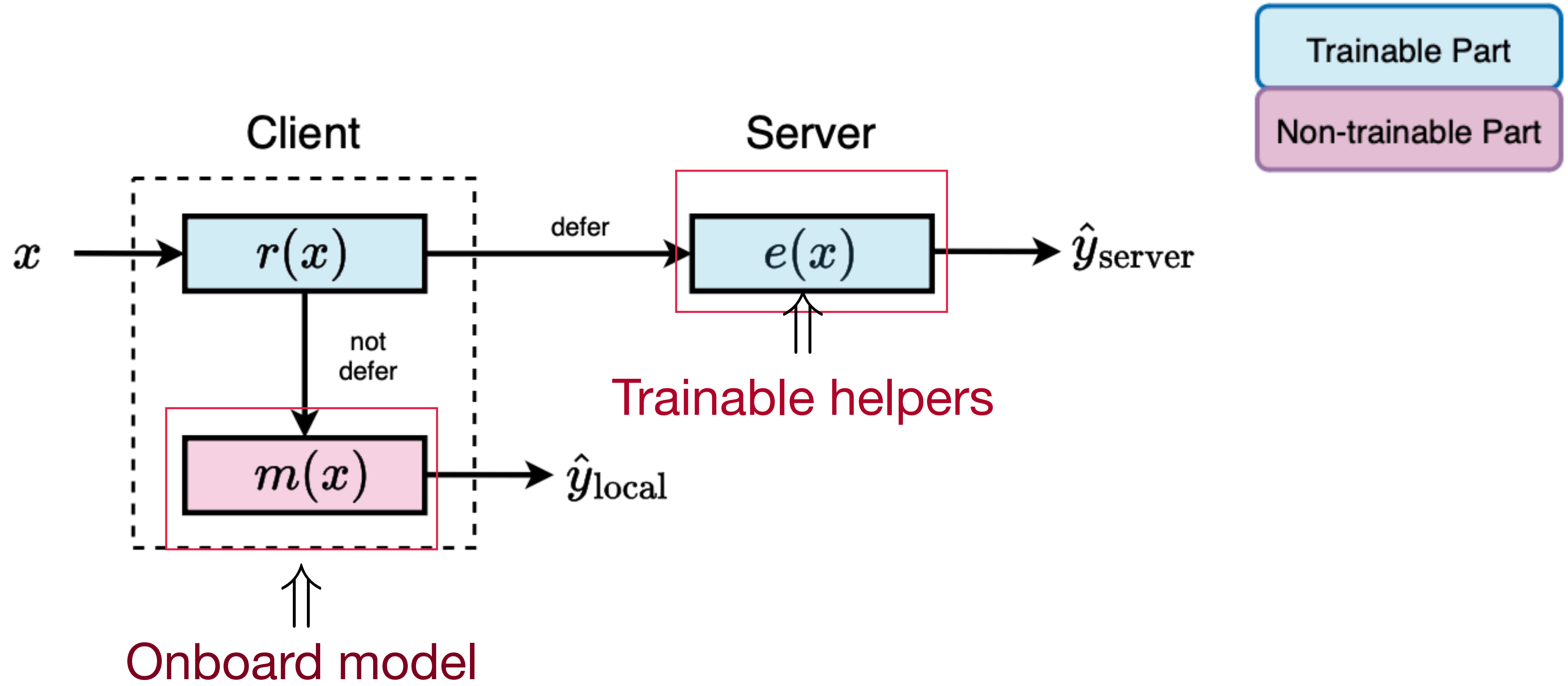


Learning with Abstention (LWA) : discard uncertain inference
Abstention doesn't solve the problem

Learning to Defer (L2D): deferring to existing human or machine experts

1. legacy local model can't be re-trained
2. asking existing experts for help is not efficient (for human expert) and not adaptable

Learning to Help (L2H) framework with helpers



Compare to LWA:

Uncertain tasks will be sent to larger model on server

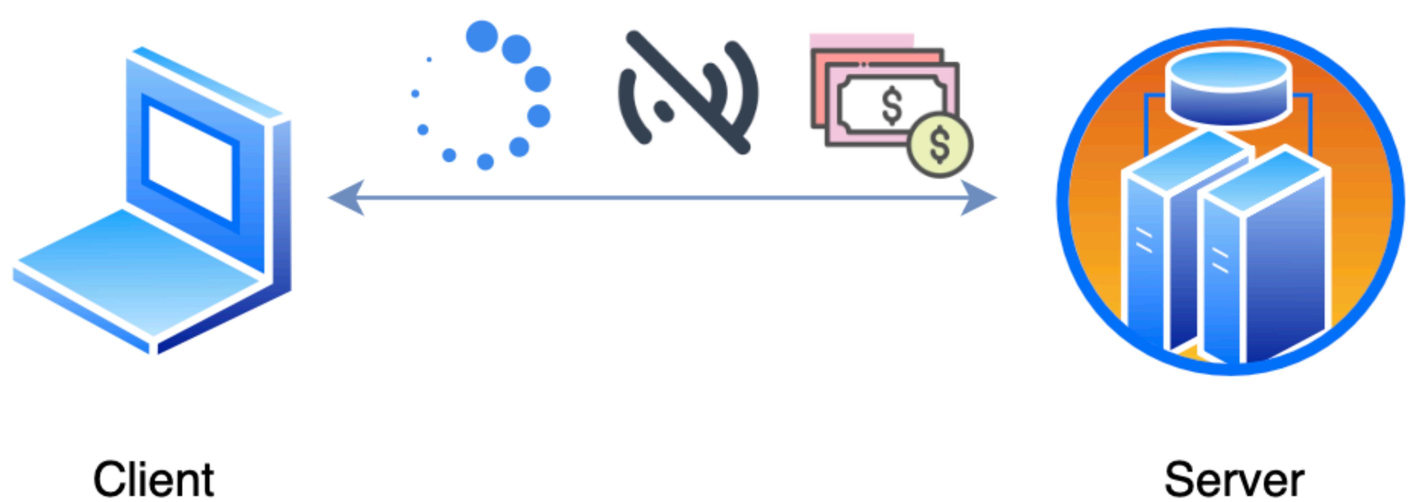
Compare to L2D:

1. Local models are fixed!
2. Non-human helpers and server model can be adaptively trained.

Learning to Help (L2H): Key Challenges

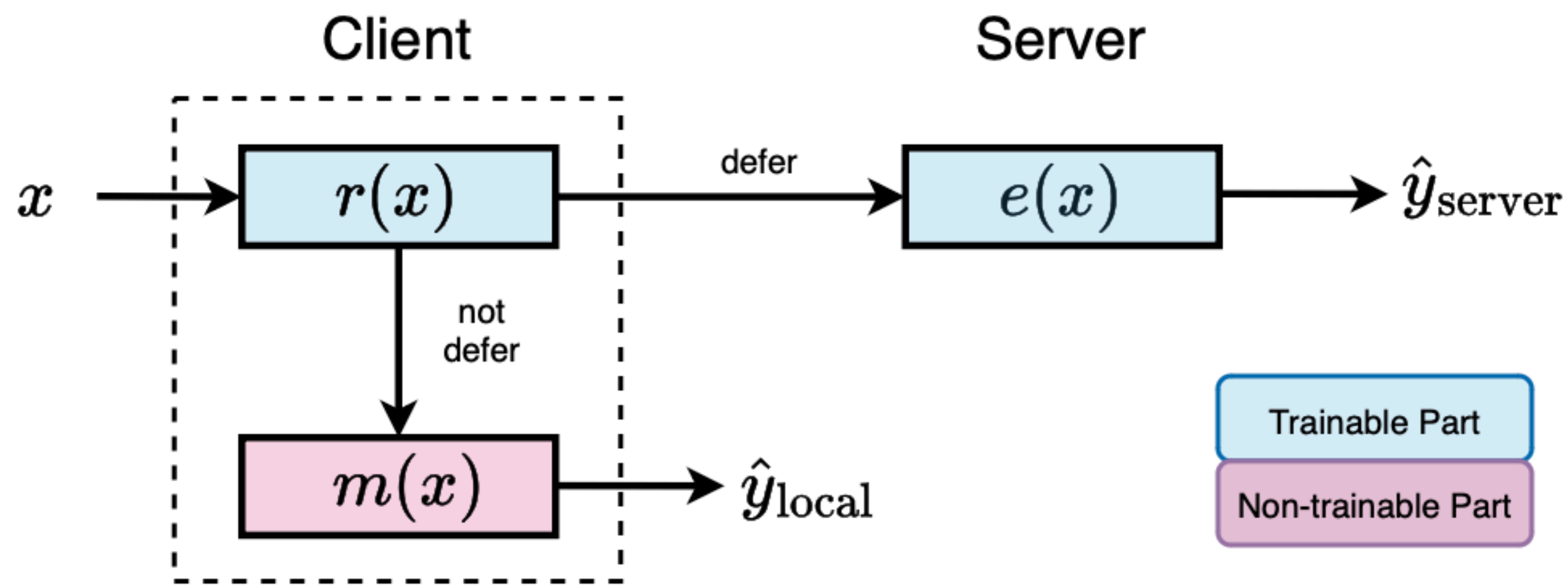
Asking for help is not free

Call server-side models can be costly: **transfer latency, instability connection, and service fees**.



A cost c_e is incurred!

Learning to Help (L2H): General loss function



Client made decisions:

$$\text{Client correct: } \hat{y}_{\mathrm{local}} = y \implies c_{\mathrm{cc}} = 0$$

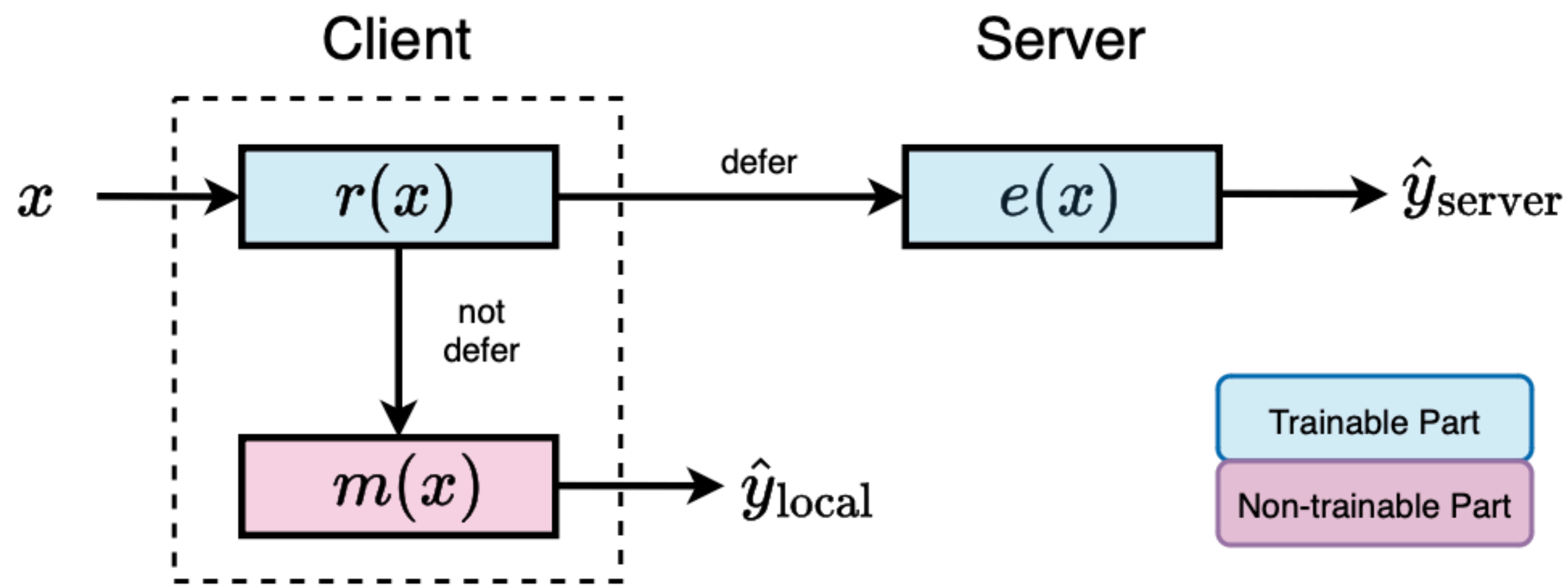
$$\text{Client error: } \hat{y}_{\mathrm{local}} \neq y \implies c_{\mathrm{ce}} = 1$$

Server made decisions:

$$\text{Server correct: } \hat{y}_{\mathrm{server}} = y \implies c_{\mathrm{sc}} + c_{\mathrm{e}} = c_{\mathrm{e}}$$

$$\text{Server error: } \hat{y}_{\mathrm{server}} \neq y \implies c_{\mathrm{se}} + c_{\mathrm{e}} = c_1 + c_{\mathrm{e}}$$

Learning to Help (L2H): General loss function



Client made decisions:

Client correct: $\hat{y}_{\text{local}} = y \implies c_{\text{cc}} = 0$

Client error: $\hat{y}_{\text{local}} \neq y \implies c_{\text{ce}} = 1$

Server made decisions:

Server correct: $\hat{y}_{\text{server}} = y \implies c_{\text{sc}} + c_{\text{e}} = c_{\text{e}}$

Server error: $\hat{y}_{\text{server}} \neq y \implies c_{\text{se}} + c_{\text{e}} = c_1 + c_{\text{e}}$

$$
\begin{aligned}
L_{\text{general}}(r, e, x, y; m) = \\
& 0 \cdot \mathbf{1}_{m(x)=y}\mathbf{1}_{r(x)=\text{LOCAL}} \\
& + \mathbf{1}_{m(x)\neq y}\mathbf{1}_{r(x)=\text{LOCAL}} \\
& + c_{\text{e}}\mathbf{1}_{e(x)=y}\mathbf{1}_{r(x)=\text{REMOTE}} \\
& + (c_{\text{e}} + c_1)\mathbf{1}_{e(x)\neq y}\mathbf{1}_{r(x)=\text{REMOTE}}.
\end{aligned}
$$

L2H objective: Bayes Classifiers

Definition: Generalized 0-1 loss function

The generalized 0-1 loss for multi-classification for L2H is defined as

$$\begin{aligned} L_{\text{general}}(r, e, x, y; m) = 0 \cdot \mathbf{1}_{m(x)=y}\mathbf{1}_{r(x)=\text{LOCAL}} \\ + \mathbf{1}_{m(x)\neq y}\mathbf{1}_{r(x)=\text{LOCAL}} \\ + c_{\text{e}}\mathbf{1}_{e(x)=y}\mathbf{1}_{r(x)=\text{REMOTE}} \\ + (c_{\text{e}} + c_1)\mathbf{1}_{e(x)\neq y}\mathbf{1}_{r(x)=\text{REMOTE}}. \end{aligned}$$

Definition: Bayes Classifiers

The *Bayes Classifiers* is defined as:

$$r^B, e^B \in \arg\min_{r,e} \mathbf{E}_{(X,Y)\sim\mathcal{D}}[L_{\text{general}}(r, e, x, y; m)]. \tag{1}$$

Theorem: Analytical Solution to Bayes Classifiers

Theorem: Bayes Classifiers

The solutions of Bayes classifiers are:

$$e^B = \arg\max_i \eta_i(x), \tag{1}$$

where $\eta_i(x) = P(Y = i|X = x)$ and

$$r^B = \mathbf{1}[\eta_{j^*(x)}(x) > (1 - c_e - c_1) + c_1 \max_i \eta_i(x)] \cdot 2 - 1, \tag{2}$$

where $j^*(x) \triangleq \arg\max_j m_j(x)$.

However, we cannot directly get the Bayes classifiers (e^B, r^B) in real-world tasks.

- The distribution $\mathcal{D}$ of the data set is unknown.
- The generalized 0-1 loss L_{general} is not differentiable, so gradient-based methods fail.

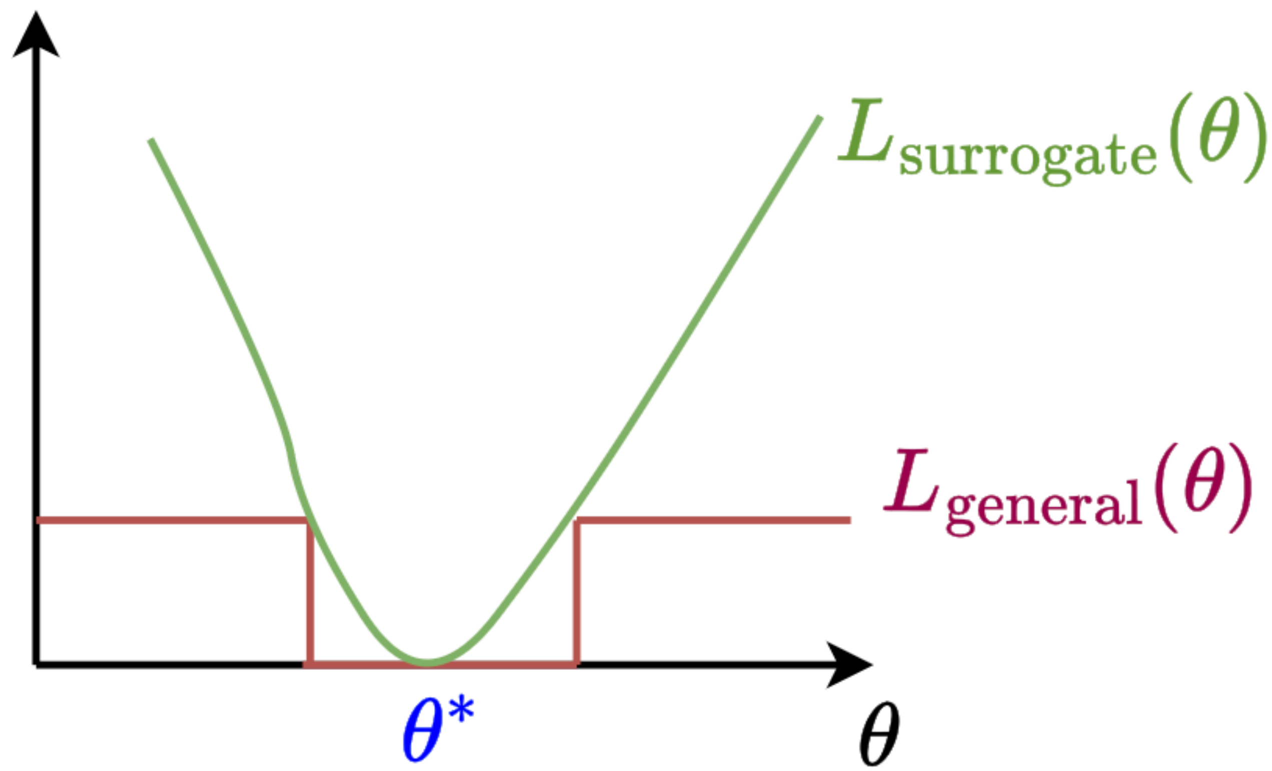
Definition: Generalized 0-1 loss function

The generalized 0-1 loss for multi-classification for L2H is defined as

$$\begin{aligned} L_{\text{general}}(r, e, x, y; m) = {} & 0 \cdot \mathbf{1}_{m(x)=y}\mathbf{1}_{r(x)=\text{LOCAL}} \\ & + \mathbf{1}_{m(x)\neq y}\mathbf{1}_{r(x)=\text{LOCAL}} \\ & + c_{\text{e}}\mathbf{1}_{e(x)=y}\mathbf{1}_{r(x)=\text{REMOTE}} \\ & + (c_{\text{e}} + c_1)\mathbf{1}_{e(x)\neq y}\mathbf{1}_{r(x)=\text{REMOTE}}. \end{aligned}$$

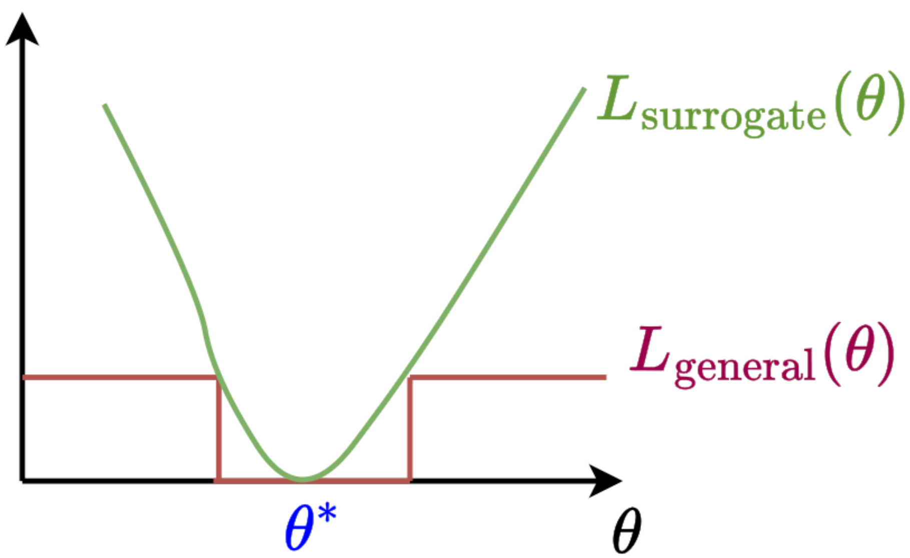
Surrogate loss function: Motivation

The desired surrogate loss function must be:



1. Differentiable
2. Convex
3. Consistent to Bayes classifiers

Surrogate loss function: Definition



1. Differentiable
2. Convex
3. Consistent to Bayes classifiers

Definition: Stage-switching surrogate loss function

Based on the definitions stated above, we propose a *stage-switching* surrogate loss function, which is differentiable and can be used in both synchronous and asynchronous settings. The surrogate loss function is defined as:

$$L_{\text{S}}(r, e, x, y; m) = L_1(e, x, y) + L_2(r, e, x, y; m) \tag{1}$$

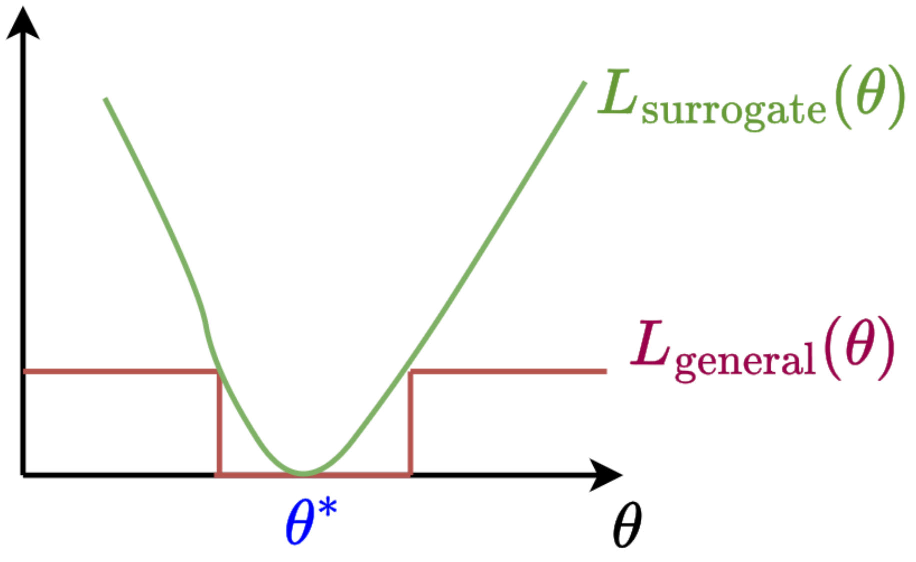
where

$$L_1(e, x, y) = -\ln \frac{\exp(e_y(x))}{\sum_{j=1}^{K} \exp(e_j(x))}, \tag{2}$$

and

$$\begin{aligned} L_2(r, e, x, y; m) = &-(1 - c_e - c_1 + c_1 \mathbf{1}_{e(x)=y}) \ln \frac{\exp(r_2(x))}{\exp(r_2(x)) + \exp(r_1(x))} \\ &- \mathbf{1}_{m(x)=y} \ln \frac{\exp(r_1(x))}{\exp(r_2(x)) + \exp(r_1(x))}. \end{aligned}$$

Surrogate loss function: convexity and monotonicity



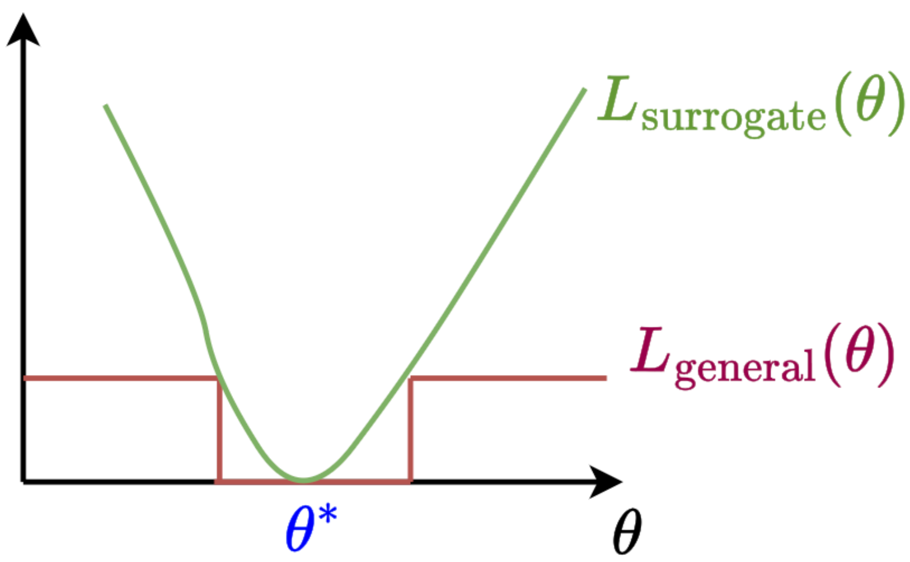
1. Differentiable
2. Convex
3. Consistent to Bayes classifiers

Proposition: Convexity and monotonicity of surrogate loss

For each given (x, y), the loss function L_1 is convex over $e_i(x)$, for any $i \in [K]$; and the loss function L_2 is:

- convex over $r_1(x)$ and $r_2(x)$, when $1 - c_e - c_1 + c_1 \mathbf{1}_{e=y} > 0$;
- monotonically decreasing over r_1 and monotonically increasing over r_2 when $1 - c_e - c_1 + c_1 \mathbf{1}_{e=y} \leq 0$.

Surrogate loss function: Consistency



1. Differentiable
2. Convex
3. Consistent to Bayes classifiers

Theorem: Consistency of surrogate loss function

Under the space of all measurable functions, the surrogate loss function is consistent with the generalized 0-1 loss function, that is, the minimizer of the risk of surrogate loss function also minimizes the risk of original loss function:

$$r^*, e^* \in \arg\min_{r,e} R_{\text{general}}(r, e; m), \tag{1}$$

for all $r^*, e^* \in \arg\min_{r,e} R_{\text{S}}(r, e; m)$.

Surrogate loss function: Flexibility

- Pay-Per-Request(PPR): the device must pay a cost each time the rejector defers to the server
- Intermittent Availability(IA): connection between client and server is not stable during training
- Bounded Reject Rate(BRR): the rate of rejections/ deferrals per unit time may not exceed a predefined upper limit (with post-training calibration)

Learning to Help (L2H): Experiments

Incorporating a server classifier indeed helps increase the overall accuracy, while both the cost of rejection and inaccuracy on the server will balance the usage of the client classifier and server classifier.

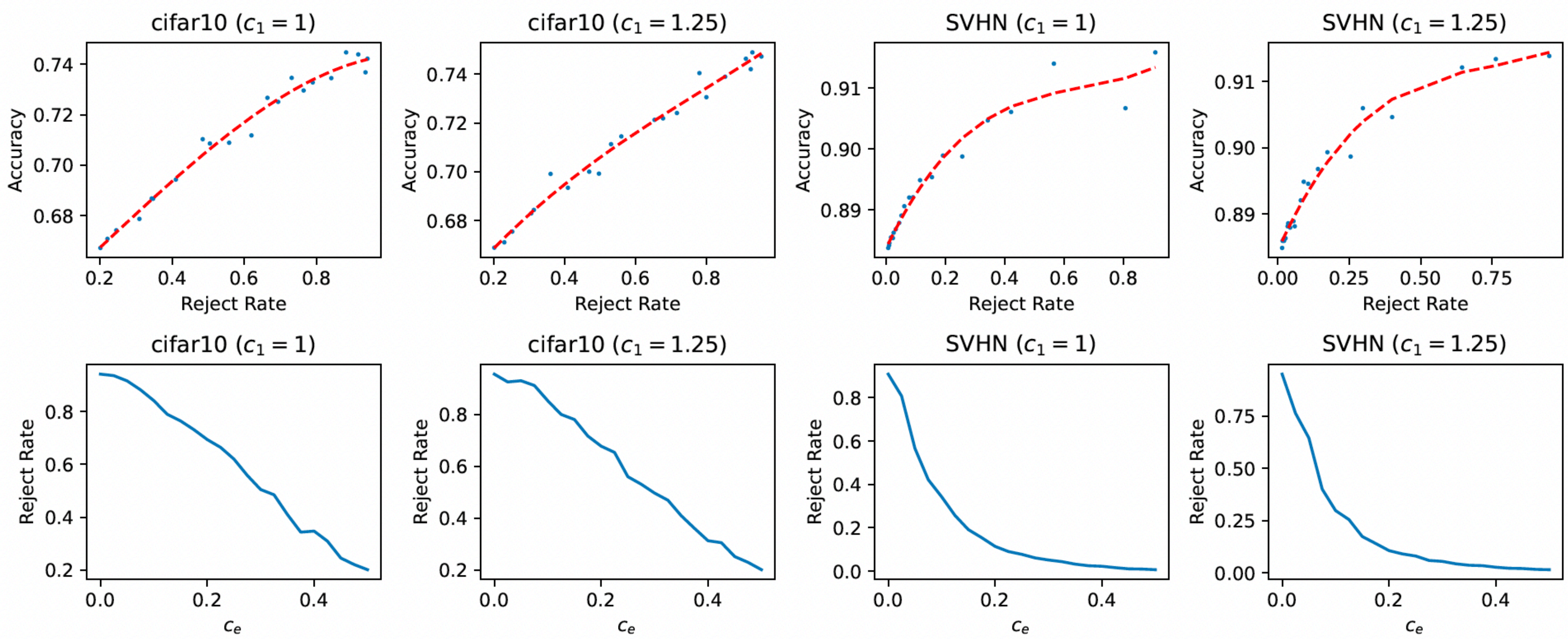


Figure 2: Impact of c_e nad c_1 on accuracy and reject rate. First row: change of accuracy as reject rate changes; Second row: change of reject rate as reject cost c_e changes.

Learning to Help (L2H): Experiments

Table 1: Contrastive Evaluation Results with $c_1 = 1.25$ and $c_e = 0.25$

	cifar10 (%)				SVHN (%)			
	ratio	m	e	differ.	ratio	m	e	differ.
data with $r(x) = \text{LOCAL}$	44.11	73.9	81.9	**8.0**	91.71	90.6	93.3	**2.7**
data with $r(x) = \text{REMOTE}$	55.9	54.5	67.7	**13.2**	8.29	61.2	72.8	**11.6**

The rejector mostly only sends the samples that are predicted inaccurately on $m(x)$ while predicted more accurately on $e(x)$ to the server end.

Learning to Help (L2H): Experiments

Samples predicted locally

Samples sent to remote model

Learning to Help (L2H): Experiments

Table 6: Accuracy of classifiers on SVHN when client classifier is pre-trained without "9" class

	"9" (%)	other classes (%)	all classes(%)
only local classifier	0	92.2	83.0
only remote classifier	94.5	89.5	90.0
jointly work	90.0	88.9	89.0
rejected rate under jointly work	93.5	12.0	17.8

Learning to Help (L2H): Experiments on Intermittent Availability (IA)

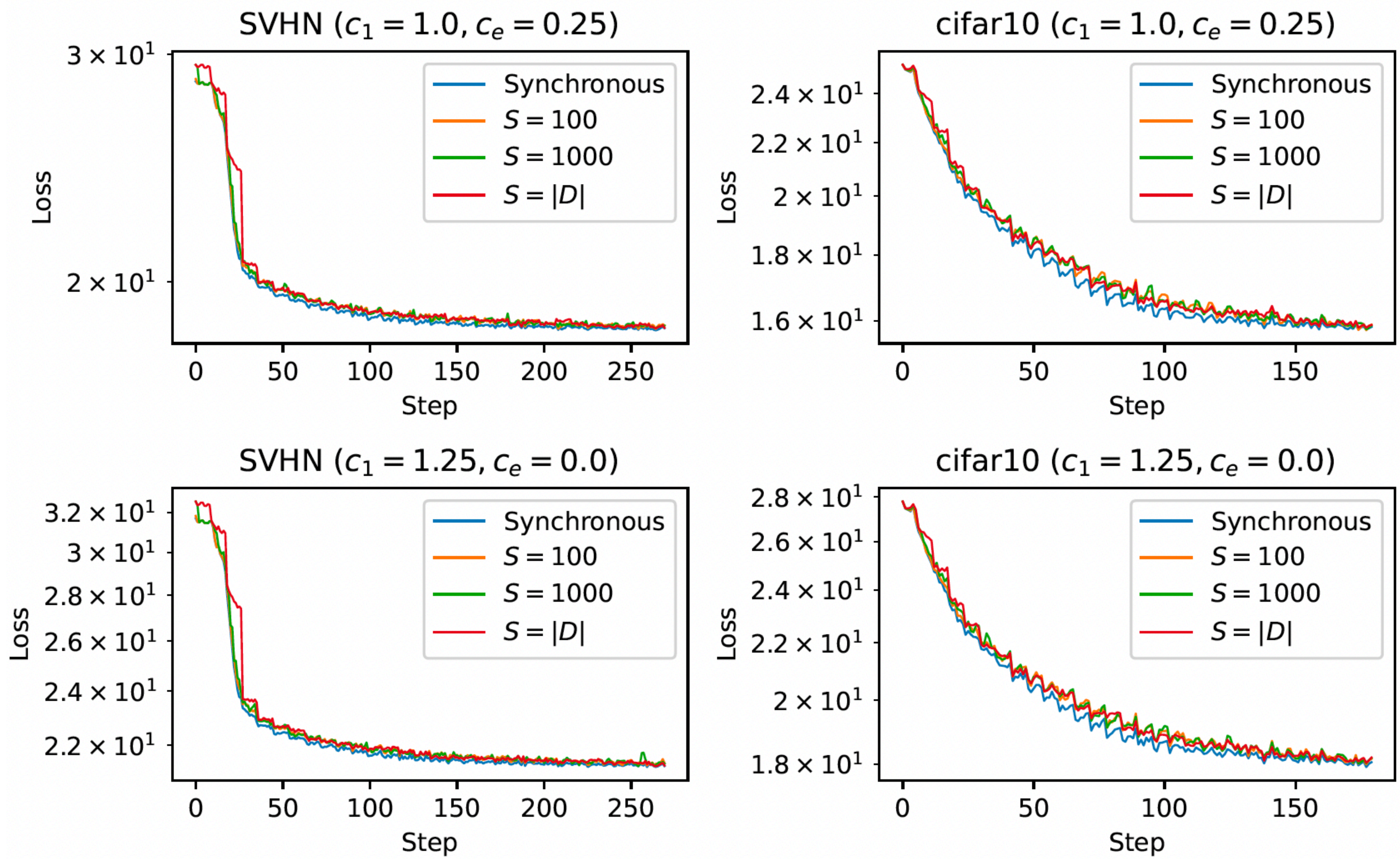


Figure 4: Comparison of synchronization and synchronization with different parameters

Learning to Help (L2H): Experiments on Bounded Reject Rate (BRR)

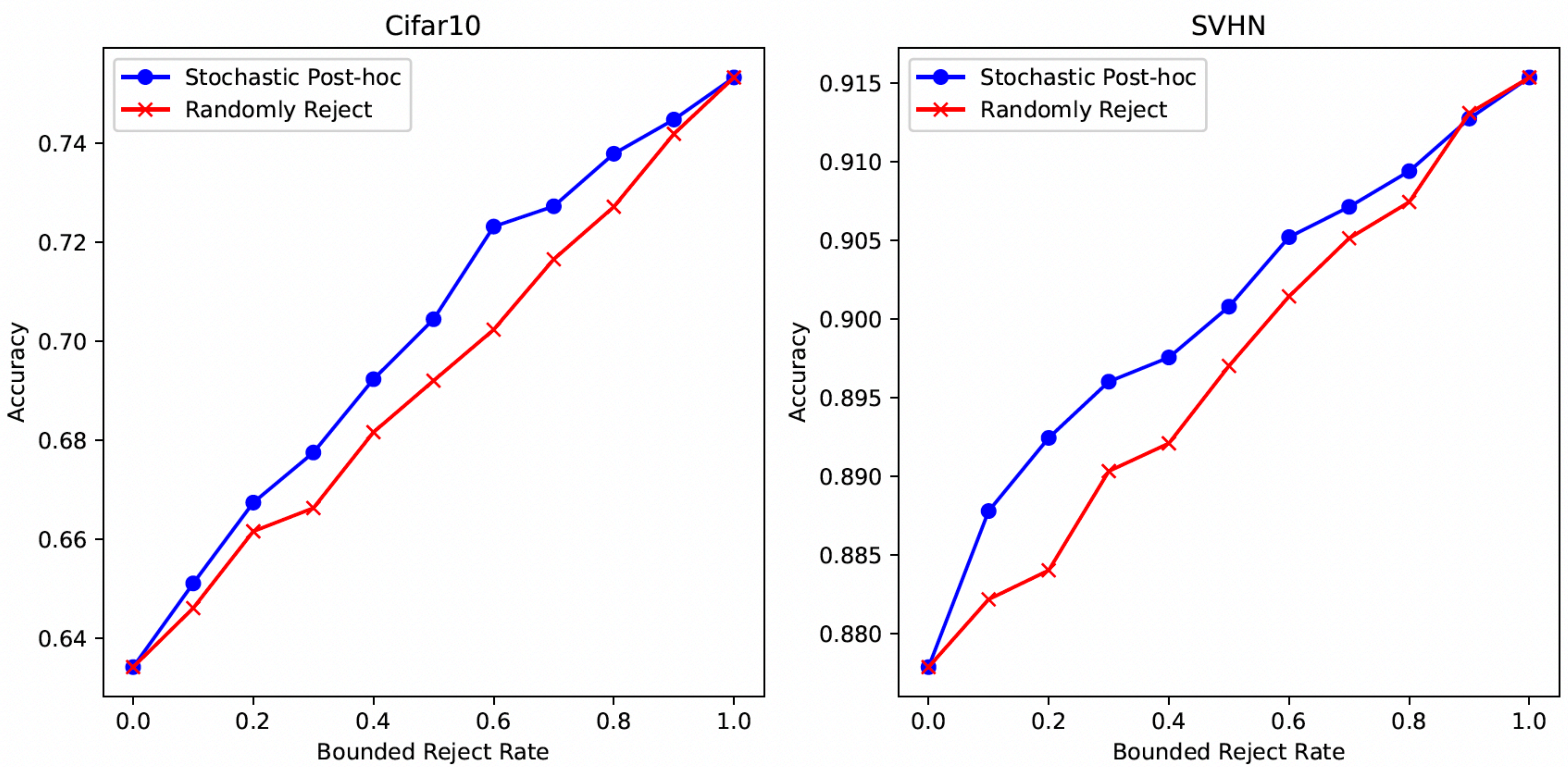


Figure 7: Comparison with randomly reject after Stochastic Post-hoc Algorithm when $c_e = 0.25$ and $c_1 = 1.12$

Learning to Help (L2H): Scalability and Generality

Flexible to scale

AlexNet (61M) → ViT (632M) $\xrightarrow{?}$ Large ViT (22B) and More

(Theoretically and empirically proved)

CIFAR-10 → CIFAR-100 $\xrightarrow{?}$ ImageNet

Common Crawl

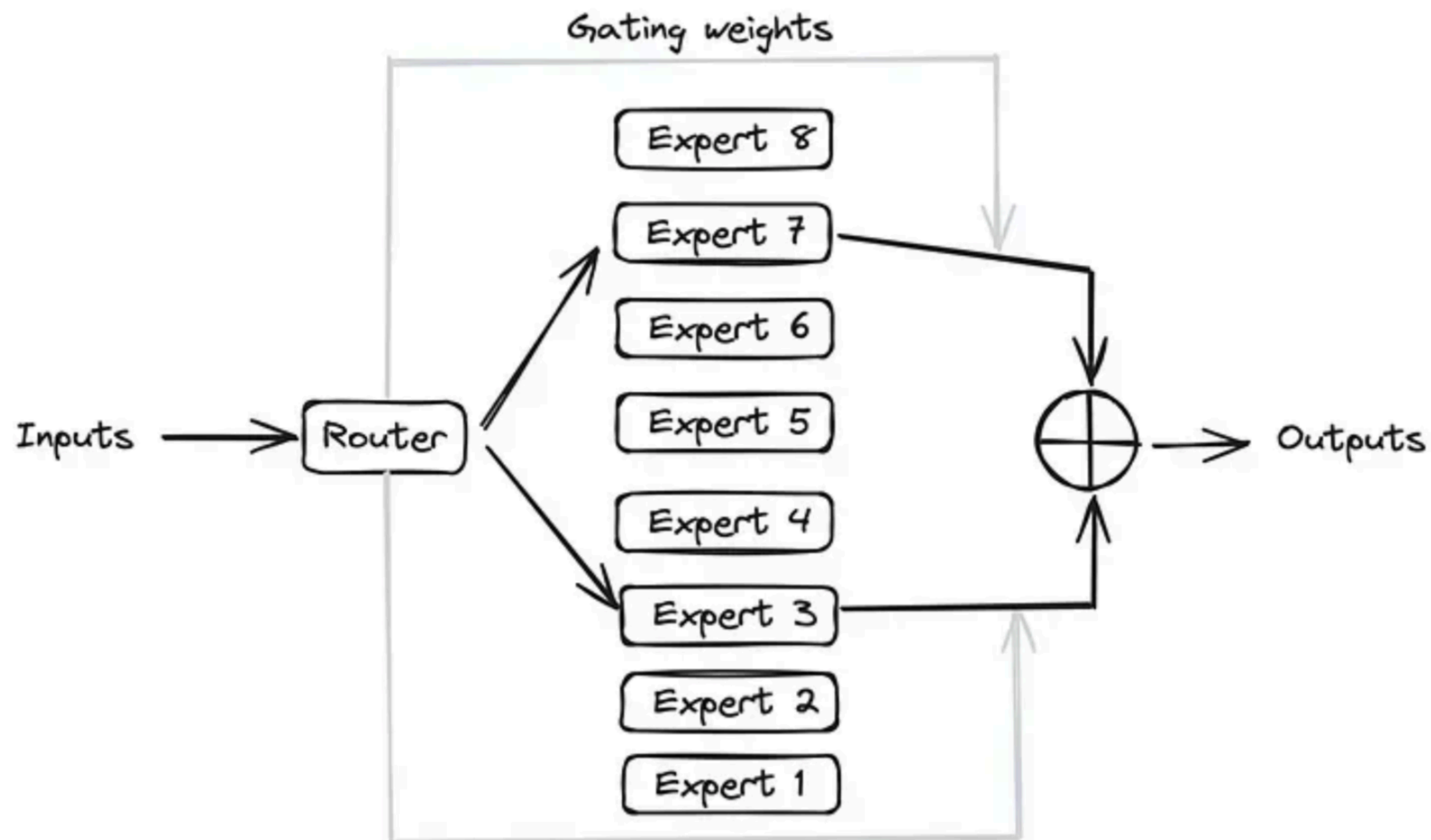
Flexible to tasks

CV $\xrightarrow{?}$ NLP

One client - One server $\xrightarrow{?}$ Multi-client - One server

One client - Multi-server

Route within Model: Mix of Experts (MoE)



MoE: routers and multiple experts in parallel

Thank you!

Reference

- Taiwo Samuel Ajani, Agbotiname Lucky Imoize, and Aderemi A. Atayero. An overview of machine learning within embedded and mobile devices–optimizations and applications. Sensors, 21 (13), 2021. ISSN 1424-8220. doi: 10.3390/s21134412. URL https://www.mdpi.com/1424-8220/21/13/4412.

- Amin Biglari and Wei Tang. A review of embedded machine learning based on hardware, application, and sensing scheme. Sensors, 23(4), 2023. ISSN 1424-8220. doi: 10.3390/s23042131. URL https://www.mdpi.com/1424-8220/23/4/2131.

- Lucjan Hanzlik, Yang Zhang, Kathrin Grosse, Ahmed Salem, Maximilian Augustin, MichaelBackes, and Mario Fritz. Mlcapsule: Guarded offline deployment of machine learning as a service.In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops(CVPRW), pp. 3295–3304, 2021. doi: 10.1109/CVPRW53098.2021.00368.

- Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857. URL https://doi.org/10.1109/TKDE.2018.2876857.

- Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles (eds.), Algorithmic Learning Theory, pp. 67–82, Cham, 2016. Springer International Publishing. ISBN 978-3-319- 46379-7. doi: 10.1007/978-3-319-46379-7_5. URL https://doi.org/10.1007/978-3-319-46379-7{_}5.

- David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In S. Bengio, H.Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/09d37c08f7b129e96277388757530c72-Paper.pdf.