Consistent Estimators for Learning to Defer to an Expert

Hussein Mozannar [1] **David Sontag** [1]

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.

[1]CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Hussein Mozannar <mozannar@mit.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

A. Practitioner's guide to our approach

Given a dataset of tuples $S = \{(x_i, y_i, m_i)\}_{i=1}^n$ where x_i represents the covariates, y_i is the target and m_i are the expert labels, we want to construct a classifier $h : \mathcal{X} \to \mathcal{Y}$ and rejector function $r : \mathcal{X} \to \{-1, 1\}$. Our method for predicting on a new example $x \in \mathcal{X}$ given expert context $z \in \mathcal{Z}$ that only the expert can observe, a function class $\mathcal{H}$ where $h \in \mathcal{H} : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|+1}$ (an example would be the set of deep networks with $|\mathcal{Y}| + 1$ output units) , and an expert $M : \mathcal{Z} \to \mathcal{Y}$ is summarized below in Algorithm 1.

Algorithm 1 Our proposed method for prediction on a new example $x \in \mathcal{X}$ with expert input $z \in \mathcal{Z}$

Input: training data $S = \{(x_i, y_i, m_i)\}_{i=1}^n$, function class $\mathcal{H}$, example x, Expert M and expert input z
$g_1, \cdots, g_{|\mathcal{Y}|}, g_\perp \leftarrow \arg\min_{\mathbf{g} \in \mathcal{H}} \sum_{i \in S} L_{CE}(\mathbf{g}, x_i, y_i, m_i)$
prediction $= 0$
$r(x) \leftarrow \text{sign}(-\max_{y \in \mathcal{Y}} g_y(x) + g_\perp(x))$
if $r(x) = 0$ **then**
  $h(x) \leftarrow \arg\max_{y \in \mathcal{Y}} g_y(x)$
  prediction $\leftarrow h(x)$
else
  $m \leftarrow M(z)$ (expert query)
  prediction $\leftarrow m$
end
Return: prediction

Where the loss L_{CE} used in algorithm is the following:

$$L_{CE}(\mathbf{g}, r, x, y, m) = -\log\left(\frac{\exp(g_y(x))}{\sum_{y' \in \mathcal{Y} \cup \perp} \exp(g_{y'}(x))}\right) - \mathbb{I}_{m=y} \log\left(\frac{\exp(g_\perp(x))}{\sum_{y' \in \mathcal{Y} \cup \perp} \exp(g_{y'}(x))}\right)$$

Practically, integrating an expert decision maker into a machine learning model amounts to two modifications in training: increasing the output size of the function class in consideration by an additional output unit representing deferral and training with the loss L_{CE} instead of the cross entropy loss. We show how to implement L_{CE} in PyTorch below:

```
def deferral_loss_L_CE(outputs, target, expert, k_classes):
    '''
    outputs: model outputs
    target: target labels
    expert: expert agreement labels for batch
    k_classes: cardinality of target Y
    '''
    batch_size = outputs.size()[0]
    defer_position =
    outputs = torch.nn.functional.softmax(outputs, dim=1)
    loss = -expert*torch.log2(outputs[range(batch_size),k_classes])
           -torch.log2(outputs[range(batch_size), labels])
    return torch.sum(loss)/batch_size
```

B. Experimental Details and Results

All experiments were run on a Linux system with an NVIDIA Tesla K80 GPU on PyTorch 1.4.0.

B.1. CIFAR-10

Implementation Details. We employ the implementation in https://github.com/xternalz/WideResNet-pytorch for the Wide Residual Networks. To train, we run SGD with an initial learning rate of 0.1, Nesterov momentum at 0.9 and weight decay of 5e-4 with a cosine annealing learning rate schedule (Loshchilov & Hutter, 2016). We train for

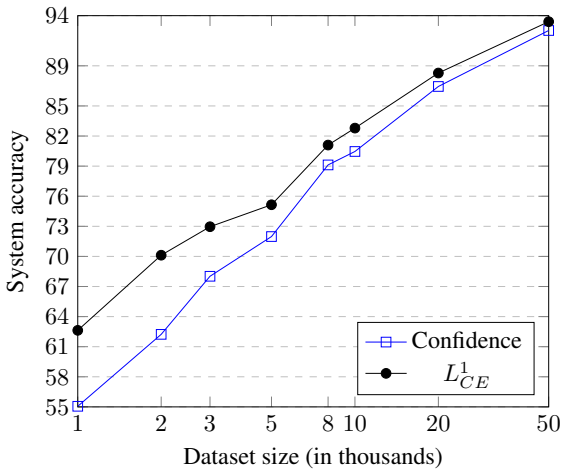


Figure 1. Varying training set size when training with expert $k = 5$ for Confidence baseline and our method L^1_{CE}.

a total of 200 epochs for all experiments, at this point the network has perfectly fit the training set, we found that early stopping based on a validation set did not make any difference and similarly training for more than 200 epochs also did not hurt test accuracy.

Expert Accuracy. In Table 3 we show the accuracy of the expert on the deferred examples versus the classes the expert can predict k. We can see that our method L^5_{CE} has higher expert accuracy than all other baselines except at $k = 1, 2$ where coverage is very high. This contrasts with Figure 2b that shows the classifier accuracy on non-deferred accuracy where L^5_{CE} had lower accuracy for each expert level compared to Confidence and L^1_{CE}. Hence there is a clear trade-off between choosing the hyper-parameter $\alpha < 1$ and $\alpha = 1$. For $\alpha < 1$, the model will prefer to always defer to the expert if it is correct, this is advantageous in this setup as the expert is perfect on a subset of the data and uniformly random on the other. However, for $\alpha = 1$, the model will compare the confidence of the expert and the model essentially performing the computation of the Bayes rejector r^B as shown by the consistency of the loss L^1_{CE}; note that for $\alpha \neq 1$ the loss L_{CE} is no longer consistent.

Table 1. Accuracy of the expert on deferred examples shown for the methods and baselines proposed with varying expert competence (k) on CIFAR-10.

METHOD / EXPERT (K)	1	2	3	4	5	6	7	8	9	10
L^1_{CE}	73.65	86.01	73.66	87.41	88.81	94.7	96.67	98.72	98.65	100
L^5_{CE}	86.44	90.96	**92.65**	**91.67**	**93.71**	**96.32**	**97.61**	98.77	**99.24**	**100**
CONFIDENCE	**87.5**	**92.74**	88.88	88.3	92.8	94.56	96.76	**98.89**	98.89	100
ORACLEREJECT	85.3	90.49	88.23	91.13	89.33	93.61	95.45	96.82	98.45	100

Increasing data size. Figure 4 plots system accuracy versus training set size when training with expert $k = 5$. We can see when data is limited our approach massively improves on the baseline, for example with 2000 training points, Confidence achieves 62.33% accuracy while our method achieves 70.12%, a 7.89 point increase. In table 4 we show the accuracy of the classifier and the coverage of the system for our method compared to the baseline Confidence for expert $k = 5$. We can see that when data is limited, our method retains high classification accuracy for the classifier versus the baseline. This is due in fact to the low coverage of our method compared to Confidence, as data size grows the coverage our method increases as now the classifier's performance improves and the system can now safely defer to it more often. On the other hand, the baseline remains at almost constant coverage, not adapting to growing data sizes.

Table 2. Accuracy of the classifier on non-deferred examples shown for our method L^1_{CE} and baseline Confidence with varying training set size for expert $k = 5$ on CIFAR-10.

METHOD / DATA SIZE (THOUSANDS)	1	2	3	5	8	10	20	50
L^1_{CE} (CLASSIFIER)	**62.84**	**71.51**	**72.63**	**75.03**	80.1	**82.11**	86.44	**95.42**
CONFIDENCE (CLASSIFIER)	50.31	59	66.3	70.12	**80.33**	78.67	**87.01**	92.45
L^1_{CE} (COVERAGE)	25.7	35.87	40.42	49.62	46.38	46.51	50	71.35
CONFIDENCE (COVERAGE)	**69.32**	**72.93**	**71.99**	**75.05**	**73.09**	**65.9**	**74.16**	**72.12**

B.2. CIFAR-10H

Class-wise Accuracy of Expert. Table 5 shows the average accuracy of the synthetic `CIFAR10H` (Peterson et al., 2019) expert on each of the 10 classes. We can see that the expert has very different accuracies for the classes which gives an opportunity for an improvement.

Results. Table 6 shows full experimental results for the CIFAR-10H results.

Table 3. Accuracy of the `CIFAR10H` (Peterson et al., 2019) expert on each of the 10 classes

CLASS	1	2	3	4	5	6	7	8	9	10
ACCURACY	95.15	97.23	94.75	91.58	90.51	94.90	96.22	97.91	97.33	96.74

Table 4. Complete results of table 2 comparing our proposed approaches and baseline.

METHOD	SYSTEM ACCURACY	COVERAGE	CLASSIFIER ACCURACY	EXPERT ACCURACY
L_{CE} IMPUTE	**96.29**±0.25	51.67±1.46	**99.2** ± 0.08	**93.18** ± 0.48
L_{CE} 2-STEP	96.03±0.21	60.81±0.87	98.11 ± 0.22	92.77 ± 0.58
CONFIDENCE	95.09±0.40	**79.48**±5.93	96.09 ± 0.42	90.94 ± 1.34

B.3. CIFAR-100

We repeat the experiments described above on the CIFAR-100 dataset (Krizhevsky et al., 2009). A 28 layer WideResNet achieves a 79.28 % test accuracy when training with data augmentation (random crops and flips). The simulated experts also operate in a similar fashion, for $k \in \{10, 20, \cdots, 100\}$, if the image is in the first k classes, the expert predicts the correct label with probability 0.94 to simulate SOTA performance on CIFAR-100 with 93.8% test accuracy (Kolesnikov et al., 2019), otherwise the expert predicts uniformly at random.

Results. In figure 5 we plot the accuracy of the combined algorithm and expert system versus k, the number of classes the expert can predict. We can see that our method dominates the baseline over all k. Compared against the confidence score baseline, the model trained with L^1_{CE} outperforms it by a 1.60 difference in test accuracy for $30 \leq k \leq 90$ on average and otherwise performs on par. This gives again gives evidence for the efficacy of our method. In table 7 we show expert, classifier and system accuracy along with coverage of both methods. Our approach L^1_{CE} obtains both better expert and classifier accuracy however gets lower coverage than Confidence.

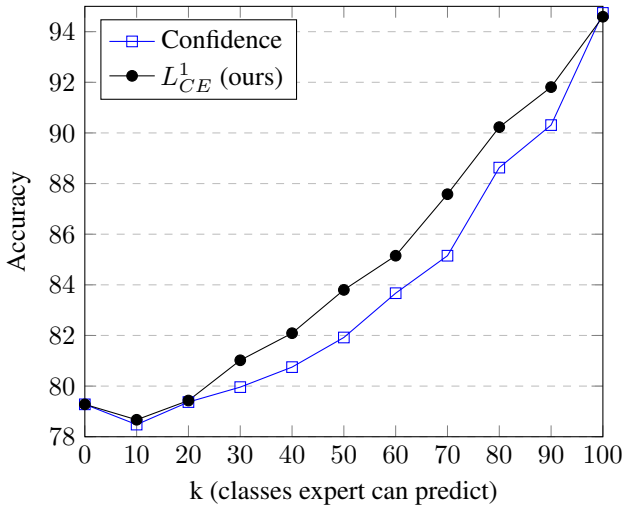


Figure 2. Comparison of the developed method L^1_{CE} on CIFAR-100 versus the confidence baseline. k is the number of classes the expert can predict

Table 5. Accuracy of the expert on deferred examples shown for the methods and baselines proposed with varying expert competence (k) on CIFAR-100.

METHOD / EXPERT (K)	10	20	30	40	50	60	70	80	90	100
L^1_{CE} (SYSTEM)	**78.67**	**79.43**	**81.02**	**82.09**	**83.8**	**85.15**	**87.58**	**90.23**	**91.81**	94.59
CONFIDENCE (SYSTEM)	78.48	79.37	79.67	80.75	81.92	83.67	85.15	88.63	90.31	**94.74**
L^1_{CE} (COVERAGE)	89.19	82.44	84.79	71.66	74.52	65.72	62.23	59.37	52.15	49.07
CONFIDENCE (COVERAGE)	**99.17**	**95.47**	**93.96**	**86.64**	**86.71**	**80.67**	**79.56**	**75.36**	**72.39**	**63.32**
L^1_{CE} (CLASSIFIER)	**82.35**	**84.03**	**84.07**	**85.29**	**86.44**	**87.78**	**90.13**	**91.89**	**92.4**	94.59
CONFIDENCE (CLASSIFIER)	78.99	80.66	81.79	84.75	84.62	87.30	88.75	90.97	92.07	**94.97**
L^1_{CE} (EXPERT)	**47.36**	**57.8**	**68.87**	**73.99**	**76.06**	**79.65**	**83.37**	**87.79**	**91.16**	**94.57**
CONFIDENCE (EXPERT)	18.07	52.09	51.49	54.79	64.4	68.55	71.13	82.11	85.70	94.30

B.4. Hate Speech experiments

Implementation details. We train all models with Adam for 15 epochs and select the best performing model on the validation set.

Results. Table 8 shows complete results of our method, baselines, expert and classifier. The performance of our method and the baselines all achieve comparable results.

Table 6. Detailed results for our method and baselines on the hate speech detection task (Davidson et al., 2017). sys: system accuracy, class: classifier accuracy, disc: system discrimination, AAE-biased: Expert 2 that has higher error rate for AAE group, non-AAE biased: Expert 3 that has higher error for non AAE tweets

METHOD/EXPERT	FAIR			AAE-BIASED		
	SYS	CLASS	DISC	SYS	CLASS	DISC
L^1_{CE} (OURS)	93.36 ± 0.16	**95.60** ± 0.44	**0.294** ±0.03	92.91 ± 0.17	**94.67** ± 0.61	**0.37** ± 0.06
CONFIDENCE	93.22 ±0.11	94.49 ± 0.12	0.45 ± 0.02	92.42 ± 0.40	94.56 ± 0.40	0.41 ± 0.02
ORACLE	**93.57** ±0.11	94.87 ±0.22	0.32 ±0.02	**93.22** ±0.11	94.49 ±0.12	0.449 ±0.024
EXPERT	89.76	–	0.031	84.28	–	0.071
CLASSIFIER	88.26	88.26	0.226	88.26	88.26	0.226

METHOD/EXPERT	NON-AAE BIASED		
	SYS	CLASS	DISC
L^1_{CE} (OURS)	90.42 ± 0.38	**94.04** ±0.81	0.231 ±0.04
CONFIDENCE	90.60 ±0.13	93.68 v0.24	0.15 ±0.03
ORACLE	**91.09** ± 0.12	92.57 ±0.15	**0.15** ±0.02
EXPERT	80.4	–	0.084
CLASSIFIER	88.26	88.26	0.226

B.5. Baseline Implementation

Description of (Madras et al., 2018) approach. A different approach to our method, is to try directly to approximate the system loss (1), this was the road taken by (Madras et al., 2018) in their differentiable model method. Let us introduce the loss used in (Madras et al., 2018):

$$L(h, r, M) = \mathbb{E}_{(x,y)\sim\mathbf{P}, m\sim M|(x,y)} \left[(1 - r(x, h(x)))l(y, h(x)) + r(x, h(x))l(y, m)\right] \tag{1}$$

where $h : \mathcal{X} \to \Delta^{|\mathcal{Y}|-1}$ (classifier), $r : \mathcal{X} \times \Delta^{|\mathcal{Y}|-1} \to \{0, 1\}$ (rejector) and the expert $M : \mathcal{Z} \to \Delta^{|\mathcal{Y}|-1}$. (Madras et al., 2018) considers only binary labels and uses the logistic loss for $l(., .)$ and thus requires the expert to produce uncertainty estimates for it's predictions instead of only a label; we can extend this to the multiclass setting by using the cross entropy loss for l. It is clear that the loss (11) is non-convex in r, hence to optimize it (Madras et al., 2018) estimates the gradient through the Concrete relaxation (Maddison et al., 2016; Jang et al., 2016). However, in the code of (Madras et al., 2018) found at `https://github.com/dmadras/predict-responsibly`, the authors replace $r(x)$ by it's estimated probability from it's model. (Madras et al., 2018) considers an additional parameter γ_{defer} found in the code, however it is not clear what effect this parameter has as we found it's description in the paper did not match the code. In detail, let $r_0, r_1 : \mathcal{X} \to \mathbb{R}$ and $r(x) = \arg\max_{i\in\{0,1\}} r_i$, the loss (Madras et al., 2018) considers is:

$$\widetilde{L}(h, r, M) = \mathbb{E}_{(x,y)\sim\mathbf{P}, m\sim M|(x,y)} \left[\frac{\exp(r_0(x))}{\exp(r_0(x)) + \exp(r_1(x))} l(y, h(x)) + \frac{\exp(r_1(x))}{\exp(r_0(x)) + \exp(r_1(x))} l(y, m)\right] \tag{2}$$

All terms in loss (12) are on the same scale which is crucial for the model to train well. We explicitly have two functions r_0 and r_1 defining r even though r is binary; this is for ease of implementation.

Another key detail of (Madras et al., 2018) approach, is that the classifier is independently trained of the rejector by stopping the gradient from r to backpropagate through h. This no longer allows h to adapt to the expert, h is trained with the cross entropy loss on it's own concurrently with r.

CIFAR-10 details. In our CIFAR-10 setup, the dataset S contains only the final prediction m of the expert M, thus to compute $l(y, m)$ we set $l(y, m) = -\log(1 - \epsilon)$ if $y = m$ and $l(y, m) = -\log(\frac{1}{|\mathcal{Y}|})$ if $y \neq m$ (simulating a uniform prediction in accordance with our expert behavior) with $\epsilon = 10^{-12}$. One could instead train a network to model the expert's prediction, we found this approach to fail as there is a big amount of noise in the labels caused by the expert's random behavior.

Results on CIFAR-10. For expert $k < 8$, we found that the (Madras et al., 2018) baseline to almost never defer to the expert and when $k = 8, 9$ at the end of training (200 epochs) the rejector never defers but the optimal system is found in the middle of training ($\sim$100 epochs). The optimal systems achieve 46.27 and 40.22 coverage, 98.81 and 98.89 expert accuracy on deferred examples and 89.38 and 89.40 classifier accuracy on non-deferred examples respectively for $k = 8, 9$. The classifier alone for the optimal systems achieve $\sim$86 classification accuracy on all of the validation set for both experts, notice that there is not much difference between the classification accuracy on all the data and non-deferred examples, while for our method and other baselines there is a considerable increase. This indicates that the rejector is only looking at the expert loss and ignoring the classifier

What is causing this behavior is that as the classifier h trains, it's loss $l(y, h(x))$ eventually goes to 0, however the loss of the expert $l(y, m)$ is either 0 or equal to $-\log(0.1)$, hence the rejector will make the easier decision to never defer. At initial epochs, we have a non-trivial rejector as the classifier h is still learning, and the coverage progressively grows till 100% over training. Essentially, what (Madras et al., 2018) approach is trying to do is choosing between the lower cost between expert and classifier: a cost-sensitive learning problem at it's heart. Therefore, one can use the losses developed here to tackle the problem better; we leave this to future investigations. Another potential fix is to learn the classifier and rejector on two different data sets.

Table 7. System accuracy of our implementation of (Madras et al., 2018) and our method and baselines with varying expert competence (k) on CIFAR-10.

METHOD / SYSTEM ACCURACY (K)	1	2	3	4	5	6	7	8	9	10
L_{CE}^{5}	**90.92**	**91.01**	**91.94**	**92.69**	**93.66**	**96.03**	**97.11**	**98.25**	**99**	**100**
L_{CE}^{1}	90.41	91.00	91.47	92.42	93.4	95.06	96.49	97.30	97.70	100
CONFIDENCE	90.47	90.56	90.71	91.41	92.52	94.15	95.5	97.35	98.05	100
ORACLEREJECT	89.54	89.51	89.48	90.75	90.64	93.25	95.28	96.52	98.16	100
(MADRAS ET AL., 2018)	90.40	90.40	90.40	90.40	90.40	90.40	90.40	94.48	95.09	100

B.6. CheXpert Experiments

Task. CheXpert is a large chest radiograph dataset that contains over 224 thousand images of 65,240 patients automatically labeled for the presence of 14 observations using radiology reports (Irvin et al., 2019). In addition to the automatically labeled training set, (Irvin et al., 2019) make publicly accessible a validation set of 200 patients labeled by a consensus of 3 radiologists and hide a further testing set of 500 patients labeled by 8 radiologists. We focus here on the detection of only the 5 observations that make up the "competition tasks" (Irvin et al., 2019): Atelectasis, Cardiomegaly, Consolidation, Edema, and Pleural Effusion. This is a multi-task problem, we have 5 separate binary tasks, we will learn to defer on an individual task basis.

Expert. We create a simulated expert as follows: if the chest X-ray contains support devices (the presence of support devices is part of the label) then the expert is correct with probability p on all tasks independently and if the X-ray does not contain support devices, then the expert is correct with probability q. We vary $q \in \{0.5, 0.7\}$ and $p \in \{0.7, 0.8, 0.9, 1\}$ to obtain different experts, we let $p \geq q$ as one can think that a patient that has support devices might have a previous medical history that the expert is aware of and can use as side-information.

Data. We use the downsampled resolution version of CheXpert (Irvin et al., 2019) and split the training data set with an 80-10-10 split on a patient basis for training, validation and testing respectively, no patients are shared among the splits. Images are normalized and resized to be compatible with pre-trained ImageNet models, we use data augmentation in the form of random resized crops, horizontal flips and random rotations of up to 15° while training.

Baselines. We implement two baselines: a threshold confidence baseline that learns a threshold to maximize system AU-ROC on just the confidence of the classifier model to defer (ModelConfidence), this is the post-hoc thresholding method in (Madras et al., 2018), and the Confidence baseline (Raghu et al., 2019a). We use temperature scaling (Guo et al., 2017) to ensure calibration of all baselines on the validation set.

Model. Following (Irvin et al., 2019), we use the DenseNet121 architecture for our model with pre-trained weights on ImageNet, the loss for the baseline models is the average of the binary cross entropy for each of the tasks. We train the

Table 8. Average difference in AU-ROC across all coverage and difference between maximum achievable AU-ROC between our method and the Confidence baseline for each of the 5 tasks and different toy expert probabilities p and q; each entry is (average difference $\pm$ standard deviation; difference of maximums). The difference between our method and the ModelConfidence is roughly twice the values noted in table 10, only at Expert $(0.7, 0.7)$ does Confidence and ModelConfidence achieve the same performance since the expert has uniform error over the domain.

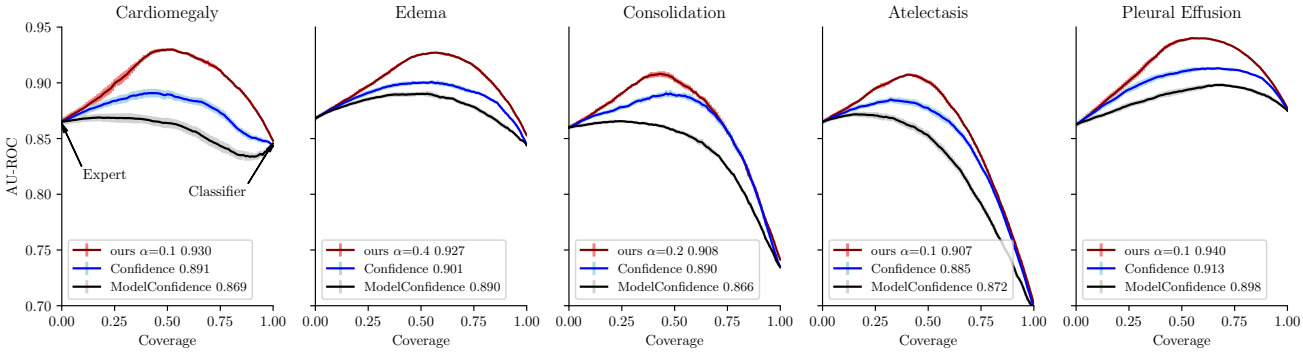
EXPERT (p, q)	CARDIOMEGALY	EDEMA	CONSOLIDATION	ATELECTASIS	PLEURAL EFFUSION
(0.5,0.7)	0.032±0.024; 0.002	0.015±0.012; 0.007	0.015±0.008; 0.007	0.017±0.009; 0.007	0.007±0.003 ;0.007
(0.5,0.9)	0.032±0.017; 0.014	0.026±0.016; 0.024	0.010±0.005; 0.015	0.016±0.008; 0.026	0.012±0.010; 0.004
(0.5,1)	0.022±0.012; 0.029	0.013±0.009; 0.019	0.007±0.008; 0.012	0.013±0.006; 0.020	0.010±0.008; 0.012
(0.7,0.7)	0.024±0.018; 0.005	0.011±0.009; 0.010	0.011±0.010; 0.009	0.006±0.006; 0.008	0.001±0.001; 0.003
(0.7,0.9)	0.032±0.020; 0.024	0.010±0.007; 0.010	0.007±0.007; 0.017	0.014±0.008; 0.017	0.010±0.006; 0.006
(0.7,1)	0.027±0.014; 0.042	0.016±0.010; 0.027	0.007±0.007; 0.019	0.013±0.007; 0.022	0.014±0.010; 0.027
(0.8,1)	0.017±0.009; 0.023	0.011±0.008; 0.012	0.001±0.004; 0.007	0.012±0.006; 0.009	0.010±0.006; 0.018

baseline models using Adam for 4 epochs. For our approach we train for 3 epochs using the cross entropy loss and then train for one epoch using L_{CE}^{α} with α chosen to maximize the area under the receiver operating characteristic curve (AU-ROC) of the combined system on the validation set for each of the 5 tasks (each task is treated separately). We also observe similar results if we train for the first three epochs with L_{CE}^{1} and then train for one epoch with a validated choice of α.

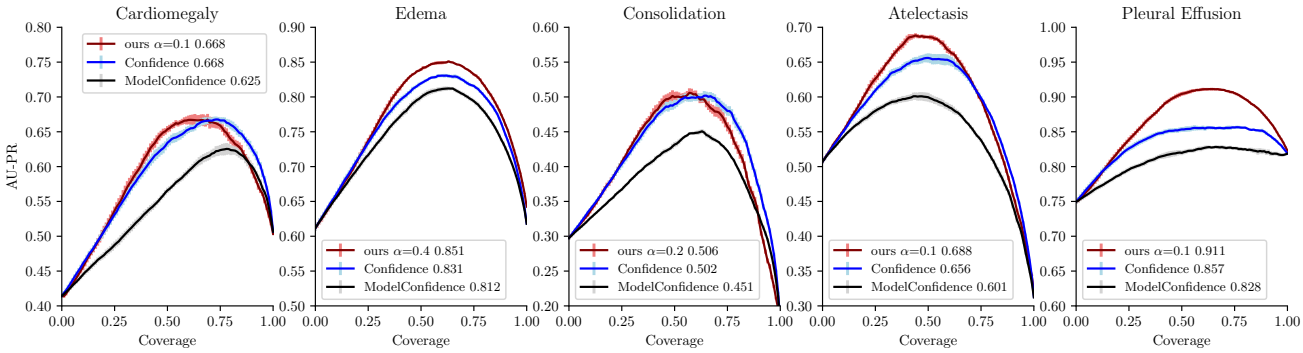
Experimental setup. In a clinical setting there might be a cost associated to querying a radiologist, this then imposes a constraint on how often we can query the radiologist i.e. our model's coverage (fraction of examples where algorithm predicts). We constrain our method and the baselines to achieve $c\%$ coverage for $c \in [100]$ to simulate the spectrum between complete automation and none.
We achieve this for our method by first sorting the test set based on $g_{\perp}(x) - \max(g_0(x), g_1(x)) := q(x)$ across all patients x in the test set, then to achieve coverage c, we define $\tau = q(x_c)$ where $q(x_c)$ is the c'th percentile of the outputs $q(x)$, then we let $r(x) = 1 \iff q(x) \geq \tau$. The definition of τ ensures that we obtain exactly $c\%$ coverage.
For ModelConfidence we achieve this by letting $q(x) = 1 - \max(g_0(x), g_1(x))$ (g is the result of a separate trained model than the one for our method), this is the natural classifier's probability of error from the softmax output, and for the Confidence we let $q(x)$ be the difference between the radiologists confidence and the classifier's confidence.

Results. In Figure 6a we plot the overall system (expert and algorithm combined) AU-ROC for each desired coverage for the methods and in Figure 6b we plot the overall system area under the precision-recall curve (AU-PR) versus the coverage; this is for the expert with $q = 0.7$ and $p = 1$. We can see that the curve for our method dominates the baselines over the entire coverage range for both AU-ROC and AU-PR, moreover the curves are concave and we can achieve higher performance by combining expert and algorithm than using both separately. Our method is able to achieve a higher maximum AU-ROC and AU-PR than both baselines: the difference between the maximum attainable AU-ROC of our method and Confidence is 0.039, 0.026, 0.018, 0.022 and 0.027 respectively for each of the five tasks. There is a clear hierarchy between the 3 compared methods: our method dominates Confidence and Confidence in turn dominates ModelConfidence, in fact ModelConfidence is a special case of the Confidence baseline, since the expert does not have uniform performance over the domain there are clear gains in modeling the expert.

This hierarchy continues to hold as we change the expert behavior as we vary the probabilities p and q, in Table 10 we show for each of the 5 tasks the difference between the average AU-ROC across all coverages (average value of the curves shown in Figure 6a) for our method and the Confidence baseline for different expert probabilities and the difference between the maximum achievable AU-ROC. A positive average difference serves to show the degree of dominance of our method over the Confidence baseline, note that the difference alone cannot imply dominance of the curves however dominance is still observed. Our method improves on the baselines as the difference between q and p increases, this difference encodes the non-uniformity of the expert behavior over the domain.



(a) AU-ROC vs coverage for expert $q = 0.7, p = 1$, maximum AU-ROC is noted.



(b) AU-PR vs coverage for expert $q = 0.7, p = 1$, maximum AU-PR is noted.

Figure 3. Plot of AU-ROC of the ROC curve (a) for each level of coverage (0 coverage means only the expert predicting and 1 coverage is only the classifier predicting) and of the area under the precision-recall curve (AU-PR) (b) for each of the 5 tasks comparing our method with the baselines on the training derived test set for the toy expert with $q = 0.7, p = 1$. We report the maximum AU-ROC and AU-PR achieved on each task, error bars are standard deviations derived from 10 runs (averaging over the expert's randomness).

C. Deferred Proofs and Derivations

C.1. Section 4

C.1.1. Binary Setting

As we eluded to in the body of the paper, we can extend the losses introduced by (Cortes et al., 2016b) to our setting for binary labels. Let $\mathcal{Y} = \{-1, +1\}$ and $r, h : \mathcal{X} \to \mathbb{R}$ where we defer if $r(x) \leq 0$, for generality we assume $l_{exp}(x, y, m) = \max(c, \mathbb{I}_{m \neq y})$ as this allows to treat rejection learning as an immediate special case. Following the derivation in (Cortes et al., 2016b), let $u \to \phi(-u)$ and $u \to \psi(-u)$ be two convex function upper bounding $\mathbb{I}_{u \leq 0}$ and let $\alpha, \beta > 0$, then:

$$
\begin{aligned}
&L_c(h, r, x, y, m) = \mathbb{I}_{h(x)y \leq 0}\mathbb{I}_{r(x)>0} + \max(c, \mathbb{I}_{m \neq y})\mathbb{I}_{r(x) \leq 0} \\
&\leq \max\left\{\mathbb{I}_{\max\{h(x)y, -r(x)\} \leq 0}, \max(c, \mathbb{I}_{m \neq y})\mathbb{I}_{r(x) \leq 0}\right\} \\
&\overset{(a)}{\leq} \max\left\{\mathbb{I}_{\frac{\alpha}{2}(h(x)y - r(x)) \leq 0}, \max(c, \mathbb{I}_{m \neq y})\mathbb{I}_{\beta r(x) \leq 0}\right\} \\
&\overset{(b)}{\leq} \max\{\phi\left(\frac{-\alpha}{2}(h(x)y - r(x))\right), \max(c, \mathbb{I}_{m \neq y})\psi\left(-\beta r(x)\right)\} \qquad (3) \\
&\leq \phi\left(\frac{-\alpha}{2}(h(x)y - r(x))\right) + \max(c, \mathbb{I}_{m \neq y})\psi\left(-\beta r(x)\right) \qquad (4)
\end{aligned}
$$

step (a) is by noting that $max(a, b) \geq \frac{a+b}{2}$, step (b) since $\phi(u)$ and $\psi(u)$ upper bound $\mathbb{I}_{u \leq 0}$. Both the right hand sides of equations (13) and (14) are convex functions of both h and r. When ϕ and ψ are both the exponential loss we obtain the following loss with $\beta(x, y, m) : \mathcal{X} \times \mathcal{Y}^2 \to \mathbb{R}^+$:

$$
L_{SH}(h, r, x, y, m) := \exp\left(\frac{\alpha}{2}(r(x) - h(x)y)\right) + (c + \mathbb{I}_{m \neq y}) \exp\left(-\beta(x, y, m)r(x)\right)
$$

we will see that it will be necessary that β is no longer constant for the loss to be consistent while in the standard case it sufficed to have β constant (Cortes et al., 2016b). The following proposition shows that for an appropriate choice of β and α we can make L_{SH} consistent.

Proposition 1. *Let* $c(x) = c - c\mathbb{P}(Y \neq M|X = x) + \mathbb{P}(Y \neq M|X = x)$, *for* $\alpha = 1$ *and* $\beta = \sqrt{\frac{1-c(x)}{c(x)}}$, $\inf_{h,r} \mathbb{E}_{x,y,m}[L_{SH}(h, r, x, y, m)]$ *is attained at* (h^*_{SH}, r^*_{SH}) *such that* $sign(h^B) = sign(h^*_{SH})$ *and* $sign(r^B) = sign(r^*_{SH})$.

Proof. Denote $\eta(x) = \mathbb{P}(Y = 1|X = x)$ and $q(x, y) = \mathbb{P}(M = 1|X = x, Y = y)$, we have:

$$
\begin{aligned}
\inf_{h,r} \mathbb{E}_{x,y,m}[L_{SH}(h, r, x, y, m)] &= \inf_{h,r} \mathbb{E}_x \mathbb{E}_{y|x} \mathbb{E}_{m|x,y}[L_{SH}(h, r, x, y, m)] \\
&= \mathbb{E}_x \inf_{h(x), r(x)} \mathbb{E}_{y|x} \mathbb{E}_{m|x,y}[L_{SH}(h(x), r(x), x, y, m)]
\end{aligned}
$$

Now we will expand the inner expectation:

$$
\begin{aligned}
&\mathbb{E}_{y|x} \mathbb{E}_{m|x,y}[L_{SH}(h(x), r(x), x, y, m)] \qquad (5) \\
&= \eta(x)q(x, 1)(\exp\left(\frac{\alpha}{2}(r(x) - h(x))\right) + c \exp\left(-\beta r(x)\right)) \\
&+ (1 - \eta(x))q(x, -1)(\exp\left(\frac{\alpha}{2}(r(x) + h(x))\right) + (1) \exp\left(-\beta r(x)\right)) \\
&+ \eta(x)(1 - q(x, 1))(\exp\left(\frac{\alpha}{2}(r(x) - h(x))\right) + (1) \exp\left(-\beta r(x)\right)) \\
&+ (1 - \eta(x))(1 - q(x, -1))(\exp\left(\frac{\alpha}{2}(r(x) + h(x))\right) + c \exp\left(-\beta r(x)\right))
\end{aligned}
$$

The Bayes optimal solution for our original loss in the binary setting is:

$$
\begin{aligned}
h^B(x) &= \eta(x) - \frac{1}{2} \\
r^B(x) &= |\eta(x) - \frac{1}{2}| - (\frac{1}{2} - c - \mathbb{P}(M \neq Y|X = x))
\end{aligned}
$$

Case 1: if $\eta(x) = 0$, writing $v = r(x), u = h(x)$ then term (15) becomes:

$$q(x,-1)(\exp\left(\frac{\alpha}{2}(v+u)\right) + 1\exp\left(-\beta v\right)) + (1-q(x,-1))(\exp\left(\frac{\alpha}{2}(v+u)\right) + c\exp\left(-\beta v\right))$$

then to minimize the above it is necessary that the optimal solutions are such that $u^* < 0, v^* > 0$ which agree with the sign of the original Bayes solution.

Case 2: if $\eta(x) = 1$, then term (15) becomes:

$$q(x,1)(\exp\left(\frac{\alpha}{2}(v-u)\right) + c\exp\left(-\beta v\right)) + (1-q(x,1))(\exp\left(\frac{\alpha}{2}(v-u)\right) + (1)\exp\left(-\beta v\right))$$

then to minimize the above it is necessary that the optimal solutions are such that $u^* > 0, v^* > 0$ which agree with the sign of the original Bayes solution.

Case 3: $\eta(x) \in (0,1)$, for ease of notation denote the RHS of equation (15) as $L_\psi(u,v)$, note that $L_\psi(u,v)$ is a convex function of both u and v and therefore to find the optimal solution it suffices to take the partial derivatives with respect to each and set them to 0.

For u:

$$\begin{aligned}
&\frac{\partial_\psi(u,v)}{\partial u} = 0 \\
&\iff -\eta(x)\frac{\alpha}{2}\exp\left(\frac{\alpha}{2}(v-u^*)\right) + (1-\eta(x))\exp\left(\frac{\alpha}{2}(v+u^*)\right) = 0 \\
&\iff -\eta(x)\frac{\alpha}{2}\exp\left(\frac{-\alpha}{2}u^*\right) + (1-\eta(x))\frac{\alpha}{2}\exp\left(\frac{\alpha}{2}u^*\right) = 0 \\
&\iff u^* = \frac{1}{\alpha}\log(\frac{\eta(x)}{1-\eta(x)})
\end{aligned}$$

we note that u^* has the same sign as the minimizer of the exponential loss and hence has the same sign as $h^B(x)$.

Plugging u^* and taking the derivative with respect to v:

$$\begin{aligned}
&\frac{\partial_\psi(u^*,v)}{\partial v} = 0 \\
&\iff \eta(x)\frac{\alpha}{2}\exp\left(\frac{\alpha}{2}(v^*-u^*)\right) + (1-\eta(x))\exp\left(\frac{\alpha}{2}(v^*+u^*)\right) \\
&-\beta c(\eta(x)q(x,1) + (1-\eta(x))(1-q(x,-1))\exp(-\beta v^*) \\
&-(1-\eta(x))q(x,-1)\beta\exp(-\beta v^*) - \eta(x)(1-q(x,1))\beta\exp(-\beta v^*) = 0 \\
&\iff \eta(x)\frac{\alpha}{2}\exp\left(\frac{\alpha}{2}(v^*-u^*)\right) + (1-\eta(x))\exp\left(\frac{\alpha}{2}(v^*+u^*)\right) \\
&-\beta(c - c\mathbb{P}(M \neq Y|X=x) + \mathbb{P}(M \neq Y|X=x))\exp(-\beta v^*) = 0
\end{aligned}$$

Appealing to the proof of Theorem 1 in (Cortes et al., 2016a) we obtain that:

$$v^* = \frac{1}{\alpha/2+\beta}\log\left(\frac{c(x)\beta}{\alpha}\sqrt{\frac{1}{\eta(x)(1-\eta(x))}}\right)$$

Furthermore by the proof of Theorem 1 in (Cortes et al., 2016a), the sign of v^* matches that of $r^B(x)$ if and only if:

$$\frac{\beta}{\alpha} = \sqrt{\frac{1-c(x)}{c(x)}}$$

□

C.1.2. MULTICLASS SETTING

Proposition 1. *$\widetilde{L}_{CE}$ is convex and is a consistent loss function for $\widetilde{L}$:*

$$\textit{let } \widetilde{\boldsymbol{g}} = \arg\inf_{\mathbf{g}} \mathbb{E}\left[\widetilde{L}_{CE}(\mathbf{g}, \mathbf{c})|X = x\right], \textit{ then: } \arg\max_{i\in[K+1]} \widetilde{\boldsymbol{g}}_i = \arg\min_{i\in[K+1]} \mathbb{E}[c(i)|X = x]$$

Proof. Writing the expected loss:

$$\inf_{\mathbf{g}} \mathbb{E}_{x,\mathbf{c}}[\widetilde{L}_{CE}(\mathbf{g}, x, \mathbf{c})] = \inf_{\mathbf{g}} \mathbb{E}_x \mathbb{E}_{\mathbf{c}|x} \widetilde{L}_{CE}(\mathbf{g}, x, \mathbf{c})] = \mathbb{E}_x \inf_{\mathbf{g}(x)} \mathbb{E}_{\mathbf{c}|x} \widetilde{L}_{CE}(\mathbf{g}(x), x, \mathbf{c})]$$

Now we will expand the inner expectation:

$$\mathbb{E}_{\mathbf{c}|x}[\widetilde{L}_{CE}(\mathbf{g}(x), x, \mathbf{c})] = -\sum_{y\in[K+1]} \mathbb{E}[\max_j c(j) - c(y)|X = x] \log\left(\frac{\exp(g_y(x)}{\sum_k \exp(g_k(x))}\right)$$

The loss $\widetilde{L}_{CE}$ is convex in the predictor, so it suffices to differentiate with respect to each g_y for $y \in \mathcal{Y}^\perp$ and set to 0.

$$\begin{aligned}
&\frac{\partial L_{CE}}{\partial g^*_y} = 0 \\
&\iff \mathbb{E}[\max_j c(j) - c(y)|X = x] - \frac{\exp(g^*_y(x)}{\sum_k \exp(g_k(x))} \sum_{i\in[K+1]} \mathbb{E}[\max_j c(j) - c(i)|X = x] = 0 \\
&\iff \frac{\exp(g^*_y(x)}{\sum_k \exp(g_k(x))} = \frac{\mathbb{E}[\max_j c(j) - c(y)|X = x]}{\sum_{i\in[K+1]} \mathbb{E}[\max_j c(j) - c(i)|X = x]}
\end{aligned}$$

From this we can deduce:

$$\begin{aligned}
h(x) &= \arg\max_{y\in[K+1]} g^*_y(x) = \arg\max_{y\in[K+1]} \frac{\exp(g^*_y(x))}{\sum_{y\in[K+1]} \exp(g^*_y(x))} \\
&= \arg\max_{y\in[K+1]} \frac{\mathbb{E}[\max_j c(j)|X = x] - \mathbb{E}[c(y)|X = x]}{\sum_{i\in[K+1]} \mathbb{E}[\max_j c(j) - c(i)|X = x]} \\
&= \arg\min_{y\in[K+1]} \mathbb{E}[c(y)|X = x] = \widetilde{h}^B(x)
\end{aligned}$$

□

Proposition 2. *The minimizers of the loss L_{0-1} (3) are defined point-wise for all $x \in \mathcal{X}$ as:*

$$\begin{aligned}
h^B(x) &= \arg\max_{y\in\mathcal{Y}} \eta_y(x) \\
r^B(x) &= \mathbb{I}_{\max_{y\in\mathcal{Y}} \eta_y(x) \leq \mathbb{P}(Y=M|X=x)}
\end{aligned} \tag{6}$$

Proof. When we don't defer, the loss incurred by the model is the misclassification loss in the standard multiclass setting and hence by standard arguments (Friedman et al., 2001) we can define h^B point-wise regardless of r:

$$h^B(x) = \arg\inf_h \mathbb{E}_y[\mathbb{I}_{h\neq y}] = \arg\max_{y\in\mathcal{Y}} \eta_y(x)$$

Now for the rejector, we should only defer if the expected loss of having the expert predict is less than the error of the classifier h^B defined above, define $r^B : \mathcal{X} \to \{0, +1\}$ as:

$$\begin{aligned}
r^B(x) &= \mathbb{I}_{\mathbb{E}[\mathbb{I}_{M\neq Y}|X=x] \leq \mathbb{E}[\mathbb{I}_{h^B(x)\neq Y}|X=x]} \\
&= \mathbb{I}_{\mathbb{P}(Y\neq M) \leq (1-\max_{y\in\mathcal{Y}} \eta_y(x))} \\
&= \mathbb{I}_{\mathbb{P}(Y=M) \geq \max_{y\in\mathcal{Y}} \eta_y(x)}
\end{aligned}$$

□

Theorem 2. *The loss L_{CE} is a convex upper bound of L_{0-1} and is consistent:*
$\inf_{h,r} \mathbb{E}_{x,y,m}[L_{CE}(h,r,x,y,m)]$ *is attained at* (h^*_{CE}, r^*_{CE}) *such that* $h^B(x) = h^*_{CE}(x)$ *and* $r^B(x) = r^*_{CE}(x)$ *for all* $x \in \mathcal{X}$.

Proof. The fact that L_{CE} is convex is immediate as $\mathbb{I}_{m=y} \geq 0$ and the cross entropy loss is convex.

Now we show that L_{CE} is an upper bound of L_{0-1}:

$$\begin{aligned} &L_{0-1}(h,r,x,y,m) = \mathbb{I}_{h(x)\neq y}\mathbb{I}_{r(x)=0} + \mathbb{I}_{m\neq y}\mathbb{I}_{r(x)=1} \\ &\overset{(a)}{\leq} -\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) - \mathbb{I}_{m=y}\log\left(\frac{\exp(g_\perp(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \end{aligned} \tag{7}$$

To justify inequality (a), consider first if $r(x) = 0$, then if $\mathbb{I}_{h(x)\neq y} = 1$ we know that $\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))} \leq \frac{1}{2}$ giving $-\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \geq 1$, moreover all the terms in the RHS of (a) are always positive.

On the other hand if $r(x) = 1$, then again $\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))} \leq \frac{1}{2}$ as we decided to reject and since also giving $-\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \geq 1$. Finally note that $L_{0-1}(h,r,x,y,m) \leq 1$.

We will now show that the optimal rejector minimizing the upper bound (17) is in fact consistent.

Denote $q_m(x,y) = \mathbb{P}(M = m|X = x, Y = y)$ and $\eta_y(x) = \mathbb{P}(Y = y|X = x)$, we have:

$$\begin{aligned} \inf_{h,r}\mathbb{E}_{x,y,m}[L_{CE}(h,r,x,y,m)] &= \inf_{h,r}\mathbb{E}_x\mathbb{E}_{y|x}\mathbb{E}_{m|x,y}[L_{CE}(h,r,x,y,m)] \\ &= \mathbb{E}_x \inf_{h(x),r(x)}\mathbb{E}_{y|x}\mathbb{E}_{m|x,y}[L_{CE}(h(x),r(x),x,y,m)] \end{aligned}$$

Let us expand the inner expectation:

$$\begin{aligned} &\mathbb{E}_{y|x}\mathbb{E}_{m|x,y}[L_{CE}(h(x),r(x),x,y,m)] \\ &= \mathbb{E}_{y|x}\left[-\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) - \sum_{m\in\mathcal{Y}}\mathbb{I}_{m=y}\log\left(\frac{\exp(g_\perp(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right)\right] \\ &= -\sum_{y\in\mathcal{Y}}\eta_y(x)\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \\ &- \sum_{y\in\mathcal{Y}}\eta_y(x)\sum_{m\in\mathcal{Y}}q_m(x,y)\mathbb{I}_{m=y}\log\left(\frac{\exp(g_\perp(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \\ &\overset{(a)}{=} -\sum_{y\in\mathcal{Y}}\eta_y(x)\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) - \sum_{y\in\mathcal{Y}}\eta_y(x)q_y(m,y)\log\left(\frac{\exp(g_\perp(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \\ &\overset{(b)}{=} -\sum_{y\in\mathcal{Y}}\eta_y(x)\log\left(\frac{\exp(g_y(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \\ &- \mathbb{P}(Y = M|X = x)\log\left(\frac{\exp(g_\perp(x))}{\sum_{y'\in\mathcal{Y}\cup\perp}\exp(g_{y'}(x))}\right) \end{aligned} \tag{8}$$

In step (a) all terms that differed on y and m disappear, in step (b) we have:

$$\sum_{y\in\mathcal{Y}}\eta_y(x)q_y(m,y) = \sum_{y\in\mathcal{Y}}\mathbb{P}(M = y, Y = y|X = x) = \mathbb{P}(Y = M|X = x)$$

For ease of notation denote the RHS of equation (18) as $L_{CE}(g_1,\cdots,g_{|\mathcal{Y}|},g_\perp)$, note that it is a a convex function, hence we will take the partial derivatives with respect to each argument and set them to 0.

For any $g_\perp$, and for $i \in \mathcal{Y}$ we have :

$$\frac{\partial L_{CE}(g_1^*, \cdots, g_{|\mathcal{Y}|^*}, g_\perp)}{\partial g_i^*} = 0$$
$$\iff \frac{\exp(g_i^*(x))}{\sum_{y' \in \widetilde{\mathcal{Y}}} \exp(g_{y'}^*(x))} = \frac{\eta_i(x)}{1 + \mathbb{P}(Y = M|X = x)} \tag{9}$$

The optimal h^* for any $g_\perp$ should satisfy equation (19) for every $i \in \mathcal{Y}$, however since exponential is an increasing function we get that the optimal h^* in fact agrees with the Bayes solution as:

$$\begin{aligned}\arg\max_{y \in \mathcal{Y}} g_y^*(X) &= \arg\max_{y \in \mathcal{Y}} \frac{\exp(g_y^*(x))}{\sum_{y \in \mathcal{Y}} \exp(g_y^*(x)) + \exp(g_\perp(x))} \\ &= \arg\max_{y \in \mathcal{Y}} \frac{\eta_y(x)}{1 + \mathbb{P}(Y = M|X = x)} = h^B(x)\end{aligned}$$

Plugging h^* and taking the derivative with respect to the optimal $g_\perp^*$:

$$\frac{\partial L_{CE}(g_1^*, \cdots, g_{|\mathcal{Y}|}^*, g_\perp^*)}{\partial g_\perp^*} = 0$$
$$\iff \frac{\exp(g_\perp^*(x))}{\sum_{y' \in \mathcal{Y}} \exp(g_{y'}^*(x))} = \frac{\mathbb{P}(Y = M|X = x)}{1 + \mathbb{P}(Y = M|X = x)}$$

Note note that $r^*(x) = 1$ only if $\mathbb{P}(Y = M|X = x) \geq \max_{y \in \mathcal{Y}} \eta_y(x)$ which agrees with $r^B(x)$ □

C.2. Section 5

Proposition 3. *L_{mix} is not a consistent surrogate loss function for L (3).*

Proof. Looking at the Bayes solution of L_{mix}, denote $q_m(x, y) = \mathbb{P}(M = m|X = x, Y = y)$, we have:

$$\inf_{h,r} \mathbb{E}_{x,y,m}[L_{mix}(h, r, x, y, m)] = \mathbb{E}_x \inf_{h(x), r(x)} \mathbb{E}_{y|x} \mathbb{E}_{m|x,y}[L_{mix}(h(x), r(x), x, y, m)]$$

Let us expand the inner expectation:

$$\mathbb{E}_{y|x} \mathbb{E}_{m|x,y}[L_{mix}(h(x), r(x), x, y, m)] = \tag{10}$$
$$-\sum_{y \in \mathcal{Y}} \eta_y(x) \log\left(\frac{\exp(g_y(x))}{\sum_{y' \in \mathcal{Y}} \exp(g_{y'}(x))}\right) \frac{\exp(r_0(x))}{\sum_{i \in \{0,1\}} \exp(r_i(x))} + \mathbb{P}(Y \neq M|X = x) \frac{\exp(r_1(x))}{\sum_{i \in \{0,1\}} \exp(r_i(x))}$$

Denote the RHS of (20) by $L_{mix}(g_1, \cdots, g_{|\mathcal{Y}|}, r_0, r_1)$, it is a convex function in g_i for all $i \in \mathcal{Y}$, consider any r_0, r_1, we have :

$$\frac{\partial L_{mix}(g_1^*, \cdots, g_{|\mathcal{Y}|^*}, r_0, r_1)}{\partial g_i^*} = 0 \iff \frac{\exp(g_i^*(x))}{\sum_{y' \in \mathcal{Y}} \exp(g_{y'}^*(x))} = \eta_i(x) \tag{11}$$

Since the optimal h^* for any r_0, r_1 *does not depend* on the form of r_0 and r_1 we conclude that (21) gives the optimal choice of h. We now need to find the optimal choice of $r_0(x)$ and $r_1(x)$ to minimize $L_{mix}(g_1^*, \cdots, g_{|\mathcal{Y}|^*}, r_0, r_1)$ which takes the following form:

$$L_{mix}(g_1^*, \cdots, g_{|\mathcal{Y}|^*}, r_0, r_1) = \mathrm{H}(h^B(x)) \frac{\exp(r_0(x))}{\sum_{i \in \{0,1\}} \exp(r_i(x))} + \mathbb{P}(Y \neq M|X = x) \frac{\exp(r_1(x))}{\sum_{i \in \{0,1\}} \exp(r_i(x))}$$

where $\mathrm{H}(X)$ is the Shannon entropy of the random variable X, here by $\mathrm{H}(h^B(x))$ we refer to the entropy of the probabilistic form of $h^B(x)$ according to (21) . Clearly the optimal r_0^* and r_1^* have the following behavior for a given $x \in \mathcal{X}$:

$$\begin{cases} r_0(x) = \infty, r_1(x) = -\infty & if\ \mathrm{H}(h^B(x)) < \mathbb{P}(Y \neq M | X = x) \\ r_0(x) = -\infty, r_1(x) = \infty & if\ \mathrm{H}(h^B(x)) \geq \mathbb{P}(Y \neq M | X = x) \end{cases}$$

This does not have the form of $r^B(x)$, as this rejector compares the entropy of $h^B(x)$ instead of it's confidence to the probability of error of the expert which will not always be in accordance. □

Theorem 2. *For any expert M and data distribution $\mathbf{P}$ over $\mathcal{X} \times \mathcal{Y}$, let $0 < \delta < \frac{1}{2}$, then with probability at least $1 - \delta$, the following holds for the empirical minimizers $(\hat{h}^*, \hat{r}^*)$:*

$$L_{0-1}(\hat{h}^*, \hat{r}^*) \leq L_{0-1}(h^*, r^*) + \mathfrak{R}_n(\mathcal{H}) + \mathfrak{R}_n(\mathcal{R}) + \mathfrak{R}_{n\mathbb{P}(M \neq Y)/2}(\mathcal{R}) + 2\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2n}} + \frac{\mathbb{P}(M \neq Y)}{2} \exp\left(-\frac{n\mathbb{P}(M \neq Y)}{8}\right)$$

Proof. Let $\mathcal{L}_{\mathcal{H},\mathcal{R}}$ be the family of functions defined as $\mathcal{L}_{\mathcal{H},\mathcal{R}} = \{(x, y, m) \to L(h, r, x, y, m); h \in \mathcal{H}, r \in \mathcal{R}\}$ with $L(h, r, x, y, m) := \mathbb{I}_{h(x) \neq y}\mathbb{I}_{r(x)=-1} + \mathbb{I}_{m \neq y}\mathbb{I}_{r(x)=1}$. Let $\mathfrak{R}_n(\mathcal{L}_{\mathcal{H},\mathcal{R}})$ be the Rademacher complexity of $\mathcal{L}_{\mathcal{H},\mathcal{R}}$, then since $L(h, r, x, y, m) \in [0, 1]$, by the standard Rademacher complexity bound (Theorem 3.3 in (Mohri et al., 2018)), with probability at least $1 - \delta/2$ we have:

$$L_{0-1}(\hat{h}^*, \hat{r}^*) \leq L_{0-1}^S(\hat{h}^*, \hat{r}^*) + 2\mathfrak{R}_n(\mathcal{L}_{\mathcal{H},\mathcal{R}}) + \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2n}}$$

We will now relate the complexity of $\mathcal{L}_{\mathcal{H},\mathcal{R}}$ to the individual classes:

$$\begin{aligned}
&\mathfrak{R}_n(\mathcal{L}_{\mathcal{H},\mathcal{R}}) = \mathbb{E}_{\boldsymbol{\epsilon}}[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{h(x_i)\neq y_i}\mathbb{I}_{r(x_i)=-1} + \epsilon_i \mathbb{I}_{m_i \neq y_i}\mathbb{I}_{r(x_i)=1}] \\
&\overset{(a)}{\leq} \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{h(x_i)\neq y_i}\mathbb{I}_{r(x_i)=-1}\right] \\
&+ \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{m_i \neq y_i}\mathbb{I}_{r(x_i)=1}\right] \\
&\overset{(b)}{\leq} \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{h(x_i)\neq y_i}\right] + \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{r(x_i)=-1}\right] \\
&+ \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{m_i \neq y_i}\mathbb{I}_{r(x_i)=1}\right] \\
&\leq \frac{1}{2}\mathfrak{R}_n(\mathcal{H}) + \frac{1}{2}\mathfrak{R}_n(\mathcal{R}) + \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{m_i \neq y_i}\mathbb{I}_{r(x_i)=1}\right]
\end{aligned} \tag{12}$$

step (a) follows as the supremum is a subadditive function , step (b) is the application of Lemma 2 in (DeSalvo et al., 2015) to $\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{I}_{h(x_i)\neq y_i}\mathbb{I}_{r(x_i)=-1}\right]$ which says that the Rademacher complexity of a product of two indicators functions is upper bounded by the sum of the complexities of each class, now we will take a closer look at the last term in the RHS of inequality (22). Denote $n_m^S = \sum_{j\in S} \mathbb{I}_{y_j \neq m_j}$ and define the random variable $S_m = \{i : y_i \neq m_i\}$, we

have that $n_m^S \sim \text{Binomial}(n, \mathbb{P}(M \neq Y))$ and $\mathbb{E}[n_m^S | S_m] = n\mathbb{P}(M \neq Y)$, hence:

$$
\begin{aligned}
&\mathbb{E}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i \mathbb{I}_{m_i \neq y_i} \mathbb{I}_{r(x_i)=1}\right] \\
&= \mathbb{E}\left[\sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{m} \sum_{i=1 \ s.t. \ y_i \neq m_i}^{m} \epsilon_i \mathbb{I}_{r(x_i)=1}\right] \\
&= \mathbb{E}\left[\frac{n_m^S}{m} \sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{n_m^S}\sum_{i=1}^{n_m^S} \epsilon_i \mathbb{I}_{r(x_i)=1}\right] \text{ (by relabeling)} \\
&\overset{(a)}{=} \mathbb{E}\left[\mathbb{E}_{\boldsymbol{\epsilon}}\left[\frac{n_m^S}{m} \sup_{(h,r)\in\mathcal{H}\times\mathcal{R}} \frac{1}{n_m^S}\sum_{i=1}^{n_m^S} \epsilon_i \mathbb{I}_{r(x_i)=1} | S_m\right]\right] \\
&\overset{(b)}{=} \mathbb{E}\left[\frac{n_m^S}{m}\hat{\mathfrak{R}}_{S_m}(\mathcal{R})\right] \\
&\overset{(c)}{=} \mathbb{P}(n_m^S < \frac{n\mathbb{P}(A)}{2})\mathbb{E}\left[\frac{n_m^S}{m}\hat{\mathfrak{R}}_{S_m}(\mathcal{R}) | n_m^S < \frac{n\mathbb{P}(A)}{2}\right] + \mathbb{P}(n_m^S \geq \frac{n\mathbb{P}(A)}{2})\mathbb{E}\left[\frac{n_m^S}{m}\hat{\mathfrak{R}}_{S_m}(\mathcal{R}) | n_m^S \geq \frac{n\mathbb{P}(A)}{2}\right] \\
&\overset{(d)}{\leq} \frac{\mathbb{P}(M \neq Y)}{2}\exp\left(-\frac{n\mathbb{P}(M \neq Y)}{8}\right) + \mathfrak{R}_{n\mathbb{P}(M\neq Y)/2}(\mathcal{R})
\end{aligned}
$$

In step (a) we conditioned on the dataset S_m, in step (b) we used the definition of the empirical Rademacher complexity $\hat{\mathfrak{R}}_{S_m}(\mathcal{R})$ on S_m, step (c) we introduce the event $A = \{M \neq Y\}$, step (d) follows from a Chernoff bound on n_m^S and since the Rademacher complexity is bounded by 1 and is non-increasing with respect to sample size.

We can now proceed with inequality (22):

$$
\mathfrak{R}_n(\mathcal{L}_{\mathcal{H},\mathcal{R}}) \overset{(a)}{\leq} \frac{1}{2}\mathfrak{R}_n(\mathcal{H}) + \frac{1}{2}\mathfrak{R}_n(\mathcal{R}) + \frac{\mathbb{P}(M \neq Y)}{2}\exp\left(-\frac{n\mathbb{P}(M \neq Y)}{8}\right) + \mathfrak{R}_{n\mathbb{P}(M\neq Y)/2}(\mathcal{R})
$$

step (a) follows as the Rademacher complexity of indicator functions based on a certain class is equal to half the Rademacher complexity of the class (Mohri et al., 2018).

The final step is to note by Hoeffding's inequality we have with probability at least $1 - \delta/2$:

$$
L^S(h^*, r^*) \leq L(h^*, r^*) + \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2n}}
$$

Now since $(\hat{h}^*, \hat{h}^*)$ are the empirical minimizers we have that $L^S(\hat{h}^*, \hat{r}^*) \leq L^S(h^*, r^*)$, collecting all the inequalities we obtain the following generalization bound with probability at least $1 - \delta$:

$$
\begin{aligned}
L(\hat{h}^*, \hat{r}^*) \leq\ & L(h^*, r^*) + \mathfrak{R}_n(\mathcal{H}) + \mathfrak{R}_n(\mathcal{R}) + 2\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2n}} \\
&+ \frac{\mathbb{P}(M \neq Y)}{2}\exp\left(-\frac{n\mathbb{P}(M \neq Y)}{8}\right) + \mathfrak{R}_{n\mathbb{P}(M\neq Y)/2}(\mathcal{R})
\end{aligned}
$$

□

References

Bansal, G., Nushi, B., Kamar, E., Weld, D. S., Lasecki, W. S., and Horvitz, E. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2429–2437, 2019.

Bansal, G., Nushi, B., Kamar, E., Horvitz, E., and Weld, D. S. Optimizing ai for teamwork. *arXiv preprint arXiv:2004.13102*, 2020.

Bartlett, P. L. and Wegkamp, M. H. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Blodgett, S. L., Green, L., and O'Connor, B. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1119–1130, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1120. URL https://www.aclweb.org/anthology/D16-1120.

Canetti, R., Cohen, A., Dikkala, N., Ramnarayan, G., Scheffler, S., and Smith, A. From soft classifiers to hard decisions: How fair can we be? In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 309–318. ACM, 2019.

Chen, M., Gummadi, R., Harris, C., and Schuurmans, D. Surrogate objectives for batch policy optimization in one-step decision making. In *Advances in Neural Information Processing Systems*, pp. 8825–8835, 2019.

Chow, C. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

Cortes, C., DeSalvo, G., and Mohri, M. Boosting with abstention. In *Advances in Neural Information Processing Systems*, pp. 1660–1668, 2016a.

Cortes, C., DeSalvo, G., and Mohri, M. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pp. 67–82. Springer, 2016b.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, 2017.

De, A., Koley, P., Ganguly, N., and Gomez-Rodriguez, M. Regression under human assistance. *arXiv preprint arXiv:1909.02963*, 2019.

DeSalvo, G., Mohri, M., and Syed, U. Learning with deep cascades. In *International Conference on Algorithmic Learning Theory*, pp. 254–269. Springer, 2015.

Dwork, C. and Ilvento, C. Fairness under composition. *arXiv preprint arXiv:1806.06122*, 2018.

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(May):1605–1641, 2010.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Futoma, J., Hariharan, S., Heller, K., Sendak, M., Brajer, N., Clement, M., Bedoya, A., and O'Brien, C. An improved multi-output gaussian process rnn with real-time validation for early sepsis detection. In *Machine Learning for Healthcare Conference*, pp. 243–254, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pp. 4878–4887, 2017.

Geifman, Y. and El-Yaniv, R. Selectivenet: A deep neural network with an integrated reject option. *arXiv preprint arXiv:1901.09192*, 2019.

Green, B. and Chen, Y. Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 90–99. ACM, 2019a.

Green, B. and Chen, Y. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019b.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

Hamid, K., Asif, A., Abbasi, W., Sabih, D., et al. Machine learning with abstention for automated liver disease diagnosis. In *2017 International Conference on Frontiers of Information Technology (FIT)*, pp. 356–361. IEEE, 2017.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jhaver, S., Birman, I., Gilbert, E., and Bruckman, A. Human-machine collaboration for content regulation: The case of reddit automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 26(5):1–35, 2019.

Jiang, H., Kim, B., Guan, M., and Gupta, M. To trust or not to trust a classifier. In *Advances in neural information processing systems*, pp. 5541–5552, 2018.

Jordan, M. I. and Jacobs, R. A. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Large scale learning of general visual representations for transfer. *arXiv preprint arXiv:1912.11370*, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Link, D., Hellingrath, B., and Ling, J. A human-is-the-loop approach for semi-automated content moderation. In *ISCRAM*, 2016.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Madras, D., Pitassi, T., and Zemel, R. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, pp. 6150–6160, 2018.

Meresht, V. B., De, A., Singla, A., and Gomez-Rodriguez, M. Learning to switch between machines and humans. *arXiv preprint arXiv:2002.04258*, 2020.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Ni, C., Charoenphakdee, N., Honda, J., and Sugiyama, M. On possibility and impossibility of multiclass classification with rejection. *arXiv preprint arXiv:1901.10655*, 2019.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. Human uncertainty makes classification more robust. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9617–9626, 2019.

Raghu, M., Blumer, K., Corrado, G., Kleinberg, J., Obermeyer, Z., and Mullainathan, S. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019a.

Raghu, M., Blumer, K., Sayres, R., Obermeyer, Z., Kleinberg, B., Mullainathan, S., and Kleinberg, J. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*, pp. 5281–5290, 2019b.

Ramaswamy, H. G., Tewari, A., Agarwal, S., et al. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12(1):530–554, 2018.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Vincent, J. Ai won't relieve the misery of facebook's human moderators. https://www.theverge.com/2019/2/27/18242724/facebook-moderation-ai-artificial-intelligence-platforms, February 2019.

Wilder, B., Horvitz, E., and Kamar, E. Learning to complement humans. *arXiv preprint arXiv:2005.00582*, 2020.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, C. and Chaudhuri, K. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pp. 703–711, 2015.

Ziyin, L., Wang, Z., Liang, P. P., Salakhutdinov, R., Morency, L.-P., and Ueda, M. Deep gamblers: Learning to abstain with portfolio theory. *arXiv preprint arXiv:1907.00208*, 2019.