# Calibrated Learning to Defer with One-vs-All Classifiers

**Rajeev Verma** [1] **Eric Nalisnick** [1]

## Abstract

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

## 1. Introduction

Machine learning is being deployed in ever more consequential and high-stakes tasks such as healthcare (Zoabi et al., 2021; Kadampur & Al Riyaee, 2020), criminal justice (Zhong et al., 2018; Chalkidis et al., 2019), and autonomous driving (Grigorescu et al., 2020). Thus, the trust and safety of these systems is paramount (Hendrycks & Dietterich, 2019; Nguyen et al., 2015). One near-term solution is to ensure a human is involved in the decision making process. For example, *learning with a rejection option* (Chow, 1957) allows the model to abstain from making a decision, instead passing the burden to a human. The decision to abstain or not is usually derived from the model's confidence. For a self-driving car, a winding stretch of road could make the system unconfident in its abilities. The system would then refuse to drive and forces the human to take control. When the system becomes confident again (e.g. on a straight road), it can then take back control from the human.

*Learning to defer* (L2D) (Madras et al., 2018) is another framework that supports machine-human collaboration. In L2D, the human's confidence is modeled as well as the machine's. This allows the system to compare the human's and model's expected performances. Thus, L2D systems defer when *the human is more likely than the model to take the correct action*. Returning to the example of a self-driving car, an L2D system would pass control to the human only when it expects the human to drive better than itself. In addition to safety, such behavior allows for an efficient *division of labor* between the human and machine. By knowing what the human knows, the model is free to adapt itself to complement the human. The model can concentrate on performing easy tasks well if it knows a human can be relied upon for harder tasks.

Most previous work has attempted to improve the overall accuracy of L2D systems. However, if these systems are to be used in safety-critical scenarios, then other factors such as trust, transparency, and fairness are important as well (Madras et al., 2018). Tschandl et al. (2020) found that AI systems can mislead physicians into incorrect diagnoses, even when the doctor is initially confident. To help prevent such scenarios, we want our systems to be well *calibrated*. The output probabilities should reflect the true uncertainties of the model and human. In other words, the L2D system should be a good forecaster. If the system says the expert has a 70% chance of being correct, then the expert should indeed be correct in about 70 out of 100 cases.

In this paper, we study the calibration of L2D systems. We focus on Mozannar & Sontag's (2020) formulation since it is the only consistent surrogate loss for multiclass L2D. We find that the Mozannar & Sontag (2020) loss results in models that are not well-calibrated with respect to expert correctness. The problem is intrinsic: the softmax parameterization allows the estimator to be *greater than one*. We propose an alternative loss based on one-vs-all classifiers that does not have this issue. We use the method of *error correcting*

[1] Informatics Institute, University of Amsterdam, Amsterdam, Netherlands. Correspondence to: Rajeev Verma <rajeev.ee15@gmail.com>, Eric Nalisnick <e.t.nalisnick@uva.nl>.

*Proceedings of the $39^{th}$ International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

*output codes* (Ramaswamy et al., 2018) to show the multiclass L2D problem reduces to multiple binary classification problems. In turn, our one-vs-all surrogate is a consistent loss function, thus making it a superior alternative to Mozannar & Sontag's (2020) loss. In experiments ranging from hate speech detection to galaxy classification to diagnosis of skin lesions, our model always performs comparably, if not better than, the Mozannar & Sontag (2020) formulation in addition to other L2D frameworks (e.g. Okati et al. (2021)) and common baselines (e.g. confidence thresholds).

## 2. Background: Multiclass Learning To Defer

Mozannar & Sontag (2020) proposed the only known consistent (surrogate) loss function for multiclass *learning to defer* (L2D). Hence, for much of this paper, we focus on their formulation. We discuss other related work in Section 5. We provide a technical overview of L2D in this section before moving on to our innovations in subsequent sections.

**Data** We first define the data for multiclass L2D. Let $\mathcal{X}$ denote the feature space, and let $\mathcal{Y}$ denote the output space, which we will always assume to be a categorical encoding of multiple $(K)$ classes. We assume that we have samples from the true generative process: $\mathbf{x}_n \in \mathcal{X}$ denotes a feature vector, and $\mathrm{y}_n \in \mathcal{Y}$ denotes the associated class defined by $\mathcal{Y}$ (1 of $K$). The L2D problem also assumes that we have access to (human) expert demonstrations. Denote the expert's prediction space as $\mathcal{M}$, which is usually taken to be equal to the label space: $\mathcal{M} = \mathcal{Y}$. The expert may also have access to additional information unavailable to the model. The expert demonstrations are denoted $\mathrm{m}_n \in \mathcal{M}$ for the associated features $\mathbf{x}_n$. The combined N-element training sample is $\mathcal{D} = \{\boldsymbol{x}_n, y_n, m_n\}_{n=1}^{N}$.

**Models** Turning to the models, Mozannar & Sontag's (2020) L2D framework is built from the classifier-rejector approach (Cortes et al., 2016a;b). The goal is to learn two functions: the *classifier*, $h : \mathcal{X} \to \mathcal{Y}$, and the *rejector*, $r : \mathcal{X} \to \{0, 1\}$. When $r(\mathbf{x}) = 0$, the classifier makes the decision in the typical way. When $r(\mathbf{x}) = 1$, the classifier abstains and defers the decision to a human (or other backup system). The rejector can be interpreted as a meta-classifier, determining which inputs are appropriate to pass to $h(\mathbf{x})$.

**Learning** The learning problem requires fitting both the rejector and classifier. When the classifier makes the prediction, then the system incurs a loss $\ell(h(\boldsymbol{x}), y)$. When the human makes the prediction (i.e. $r(\boldsymbol{x}) = 1$), then the system incurs a loss $\ell_{\text{exp}}(m, y)$. Using the rejector to combine these losses, we have the overall classifier-rejector loss:

$$L(h, r) = \mathbb{E}_{\mathbf{x},\mathrm{y},\mathrm{m}}\left[(1 - r(\mathbf{x}))\, \ell(h(\mathbf{x}), \mathrm{y}) \;+\; r(\mathbf{x})\, \ell_{\text{exp}}(\mathrm{m}, \mathrm{y})\right] \tag{1}$$

where the rejector is acting as an indicator function that controls which loss to use. While this formulation is valid for general losses, the canonical $0-1$ loss is of special interest for classification tasks:

$$L_{0-1}(h, r) = \mathbb{E}_{\mathbf{x},\mathrm{y},\mathrm{m}}\left[(1 - r(\mathbf{x}))\, \mathbb{I}[h(\mathbf{x}) \neq \mathrm{y}] \;+\; r(\mathbf{x})\, \mathbb{I}[\mathrm{m} \neq \mathrm{y}]\right] \tag{2}$$

where $\mathbb{I}$ denotes an indicator function that checks if the prediction and label are equal or not.

**Softmax Surrogate** The key innovation of Mozannar & Sontag (2020) is the proposal of a consistent surrogate loss for $L_{0-1}$. They accomplish this by first unifying the classifier and rejector via an augmented label space that includes the rejection option. Formally, this label space is defined as $\mathcal{Y}^{\perp} = \mathcal{Y} \cup \{\perp\}$ where $\perp$ denotes the rejection option. Secondly, Mozannar & Sontag (2020) use a reduction to cost sensitive learning that ultimately resembles the cross-entropy loss for a softmax parameterization. Let $g_k : \mathcal{X} \mapsto \mathbb{R}$ for $k \in [1, K]$ where $k$ denotes the class index, and let $g_{\perp} : \mathcal{X} \mapsto \mathbb{R}$ denote the rejection $(\perp)$ option. These $K + 1$ functions are then combined in the following softmax-parameterized surrogate loss:

$$\begin{aligned} \phi_{\text{SM}}(g_1, \ldots, g_K, g_{\perp}; \boldsymbol{x}, y, m) = \\ -\log\left(\frac{\exp\{g_y(\boldsymbol{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\boldsymbol{x})\}}\right) \\ -\mathbb{I}[m = y]\, \log\left(\frac{\exp\{g_{\perp}(\boldsymbol{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\boldsymbol{x})\}}\right). \end{aligned} \tag{3}$$

The intuition is that the first term maximizes the function $g_k$ associated with the true label. The second term then maximizes the rejection function $g_{\perp}$ but only if the expert's prediction is correct. At test time, the classifier is obtained by taking the maximum over $k \in [1, K]$: $\hat{y} = h(\boldsymbol{x}) = \arg\max_{k \in [1,K]} g_k(\boldsymbol{x})$. The rejection function is similarly formulated as $r(\boldsymbol{x}) = \mathbb{I}[g_{\perp}(\boldsymbol{x}) \geq \max_k g_k(\boldsymbol{x})]$. In practice, Mozannar & Sontag (2020) introduce a hyperparameter $\alpha \in \mathbb{R}^{+}$ that re-weights the classifier loss when the expert is correct. Using $\alpha < 1$ encourages a higher degree of division of labor between classifier and expert. Yet for all $\alpha \neq 1$, the surrogate is no longer consistent.

The function $\phi_{\text{SM}}$ is the first convex (in $g$) consistent surrogate loss proposed for L2D (Mozannar & Sontag, 2020). The minimizers $g_1^*, \ldots, g_K^*, g_{\perp}^*$ of $\phi_{\text{SM}}$ also uniquely minimize $L_{0-1}(h, r)$, the $0-1$ loss from Equation 2. The resulting optimal classifier and rejector satisfy:

$$\begin{aligned} h^*(\boldsymbol{x}) &= \arg\max_{y \in \mathcal{Y}} \; \mathbb{P}(\mathrm{y} = y | \boldsymbol{x}), \\ r^*(\boldsymbol{x}) &= \mathbb{I}\left[\mathbb{P}(\mathrm{m} = \mathrm{y} | \boldsymbol{x}) \geq \max_{y \in \mathcal{Y}} \mathbb{P}(\mathrm{y} = y | \boldsymbol{x})\right], \end{aligned} \tag{4}$$

where $\mathbb{P}(\mathrm{y}|\boldsymbol{x})$ is the probability of the label under the data generating process, and $\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})$ is the probability that the expert is correct. Recall that, by assumption, the expert likely will have additional knowledge not available to the classifier. This assumption is what allows the expert to possibly outperform the Bayes optimal classifier.

## 3. Problem with Softmax Parameterization

The minimizers of the surrogate proposed by Mozannar & Sontag (2020) should correspond to the Bayes optimal classifier and rejector. In this section, we investigate if the resulting model can correctly estimate the underlying probability that the expert is correct. We find that, unfortunately, the resulting models are not well calibrated. The problem lies in the softmax parameterization: it yields a degenerate estimate of $\mathbb{P}(\mathrm{m}=\mathrm{y}|\mathbf{x})$. Specifically, the estimator is unbounded, taking on values larger than one. We do not study the calibration of the classifier since post-hoc methods (e.g. temperature scaling (Guo et al., 2017)) can be applied to the classifier sub-components of both our method and Mozannar & Sontag's (2020).

**Probabilistic Rejector** We first introduce the probabilistic rejection function. One may be tempted to work directly with the deferral function from Equation 3:

$$p_{\perp}(\boldsymbol{x}) = \frac{\exp\{g_{\perp}(\boldsymbol{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\boldsymbol{x})\}}. \tag{5}$$

However, inspecting Mozannar & Sontag's (2020) Theorem 1, we see that $p^{*}_{\perp}(\boldsymbol{x}) = \mathbb{P}(\mathrm{m}=\mathrm{y}|\mathbf{x})/(1+\mathbb{P}(\mathrm{m}=\mathrm{y}|\mathbf{x}))$ at the Bayes optimum. Rearranging this equation gives the appropriate estimator for $\mathbb{P}(\mathrm{m}=\mathrm{y}|\mathbf{x})$:

$$p_{\mathrm{m}}(\boldsymbol{x}) = \frac{p_{\perp}(\boldsymbol{x})}{1-p_{\perp}(\boldsymbol{x})}. \tag{6}$$

The full derivation is in Appendix C.1. A crucial observation is that $p_{\mathrm{m}}(\boldsymbol{x}) \in (0, \infty)$, meaning that the function is unbounded from above. This will be of consequence when considering if it is calibrated.

**Calibration** We next define the relevant notion of calibration. For the function $p_{\mathrm{m}}(\boldsymbol{x})$ from Equation 6, we call $p_{\mathrm{m}}$ *calibrated* if, for any confidence level $c \in (0,1)$, the actual proportion of times the expert is correct is equal to $c$:

$$\mathbb{P}(\mathrm{m}=\mathrm{y} \mid p_{\mathrm{m}}(\boldsymbol{x}) = c) = c. \tag{7}$$

This statement should hold for all possible instances $\boldsymbol{x}$ with confidence $c$. Since expert correctness is a binary classification problem, distribution calibration, confidence calibration, and classwise calibration all coincide (Vaicenavicius et al., 2019).

**Calibration of Expert Correctness** We next examine if Equation 6 is a valid estimator of the probability that the expert's prediction is correct. Unfortunately, $p_{\mathrm{m}}(\boldsymbol{x})$ is unbounded; we formalize this fact in the statement below.

**Proposition 3.1.** *If* $\exists\, \boldsymbol{x} \in \mathcal{X}$ *for which* $p_{\perp}(\boldsymbol{x}) > 1/2$*, then* $p_{\mathrm{m}}(\boldsymbol{x}) > 1$*. Hence* $p_{\mathrm{m}}(\boldsymbol{x})$ *cannot estimate* $\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})$.

This proposition is obvious from the fact that $p_{\mathrm{m}}(\boldsymbol{x})$ is the odds of $p_{\perp}(\boldsymbol{x}) \in (0,1)$. Proposition 3.1 does *not* imply a problem with the consistency of Mozannar & Sontag's (2020) surrogate loss. Rather, it means that the softmax parameterization admits many solutions that do not correspond to valid estimators for $\mathbb{P}(\mathrm{y}=\mathrm{m}|\boldsymbol{x})$. In other words, the Bayes solutions seem to be 'fragile' in the sense that they require $p_{\perp}(\boldsymbol{x}) \leq 1/2$ while its true range is $(0,1)$.

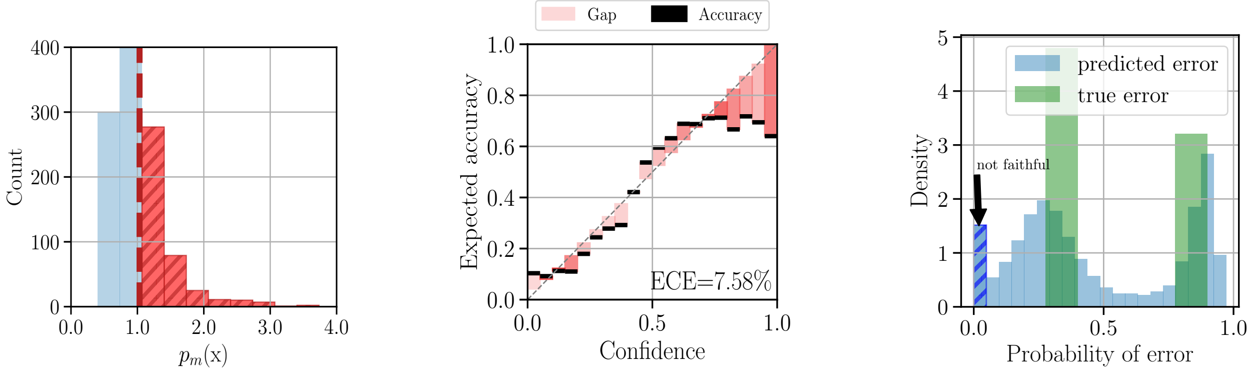
To make matters concrete, consider the case in which the expert is always correct, $\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x}) = 1$, while the class distribution is maximally entropic, $\mathbb{P}(\mathrm{y}|\boldsymbol{x}) = 1/K$. From Equation 6, a perfect expert implies that $p_{\perp}(\mathbf{x}) = 1/2$. In turn, $p_k(\mathbf{x}) = 1/(2K)$. For $K=2$, the softmax in Equation 3 would produce the vector $[1/4,\ 1/4,\ 1/2]$. While the resulting model would indeed correctly defer to the expert (since $g_{\perp} > g_k$), the output is not what we might expect for a case in which the classifier is useless and the expert is an oracle. Intuition suggests that we should see an output like $[\epsilon/2,\ \epsilon/2,\ 1-\epsilon]$ where $\epsilon$ is a small positive constant, as this seems to more accurately reflect the expert's clear superiority. In practice, perhaps optimization is finding well-performing but non-optimal solutions like this one.

**Experimental Confirmation** We now establish that $p_{\perp}(\boldsymbol{x}) > 1/2$ does occur in practice. We use a CIFAR-10 simulation that is similar to Mozannar & Sontag's (2020) CIFAR-10 experiment. The expert is assumed to have non-uniform expertise: 75% chance of being correct on the first five classes, and 20% (i.e. random) chance on the last five classes. Subfigure 1a shows a histogram of the values of $p_{\mathrm{m}}(\boldsymbol{x})$ as observed on the CIFAR-10 test set. The blue bars represent the values less than or equal to one. The red bars show the pathological cases greater than one. 39.4% of the test samples (3940 instances) resulted in $p_{\mathrm{m}}(\boldsymbol{x}) > 1$.

We also consider modifying $p_{\mathrm{m}}(\boldsymbol{x})$ so that all values greater than one are rounded down to one. In this case, since now $p_{\mathrm{m}}(\boldsymbol{x})$ is forcibly restricted to $(0,1]$, we can perform standard evaluations of calibration, such as plotting a reliability diagram and computing *expected calibration error* (ECE). In this case, the relevant ECE is defined as

$$\mathrm{ECE}(p_{\mathrm{m}}) = \mathbb{E}_{\mathbf{x}} |\mathbb{P}\left(\mathrm{m}=\mathrm{y} \mid p_{\mathrm{m}}(\mathbf{x}) = c\right) - c|.$$

Subfigure 1b shows the reliability diagram and reports the ECE for confidence calibration when $p_{\mathrm{m}}(\boldsymbol{x})$ is restricted. Unsurprisingly, we still observe that the model's estimate of



(a) Empirical Distribution of $p_{\mathrm{m}}$ Values

(b) Reliability Diagram and ECE

(c) Empirical Distribution of $1 - p_{\mathrm{m}}(\mathbf{x})$

*Figure 1. Calibration of Softmax Parameterization on CIFAR-10*: Subfigure (a) reports the observed values of $p_{\mathrm{m}}(\mathbf{x})$ on the CIFAR-10 simulation study. We find that 39.4% of test samples have $p_{\mathrm{m}}(\mathbf{x}) > 1$ (denoted in red). Subfigure (b) reports a reliability diagram and the expected calibration error (ECE) when $p_{\mathrm{m}}(\mathbf{x})$ is restricted to $(0, 1]$. The shade of the bin color represents the proportion of samples in the bin (darker shade, more samples). Subfigure (c) shows the distribution of risk estimates. Note the clear bias towards zero error.

the expert's correctness is uncalibrated, exhibiting overconfidence. The ECE is 7.58%. For comparison, our one-vs-all method has an ECE of 3.01%, as we will describe later. Subfigure 1c plots the distribution of risks: $1 - p_{\mathrm{m}}(\mathbf{x})$. Due to the probabilities being clamped to one, we see a false mode at zero error. In turn, the system is not transparent about the actual risk that decision makers would encounter.

**Proxy via Deferral Function** Returning to Equation 5, it is possible that the deferral function $p_{\perp}(\mathbf{x})$ is a useful estimator of $\mathbb{P}(\mathrm{m} = \mathrm{y}|\boldsymbol{x})$, despite that theory suggests otherwise. Here the range is no longer a problem because $p_{\perp}(\mathbf{x}) \in (0, 1)$. Moreover, as discussed in the example above, intuition suggests that $p_{\perp}(\mathbf{x})$ should correlate with the expert's degree of superiority to the classifier. In the experiments (Section 6.1), we investigate if the proxy $p_{\perp}$ is a useful estimator of $\mathbb{P}(\mathrm{m} = \mathrm{y}|\boldsymbol{x})$. We ultimately find that it is not, as it results in ECEs above 30%.

**Applicability of Post-Hoc Techniques** There are a range of post-hoc techniques designed to fix mis-calibration in classifiers, e.g. *temperature scaling* (Guo et al., 2017), *Dirichlet calibration* (Kull et al., 2019), *top-label calibration* (Gupta & Ramdas, 2022). These techniques employ a calibration map (Vaicenavicius et al., 2019): a usually simple transformation that is applied to the confidence estimates to re-calibrate them. Such a map is fitted on a held-out validation set using some goodness-of-fit measure, e.g. log-likelihood. Due to the softmax's range problem and interdependence of its $g$ functions, we do not know of a general procedure for defining and fitting a calibration map for the L2D setting.

## 4. Consistent and Calibrated L2D with a One-vs-All Surrogate Loss

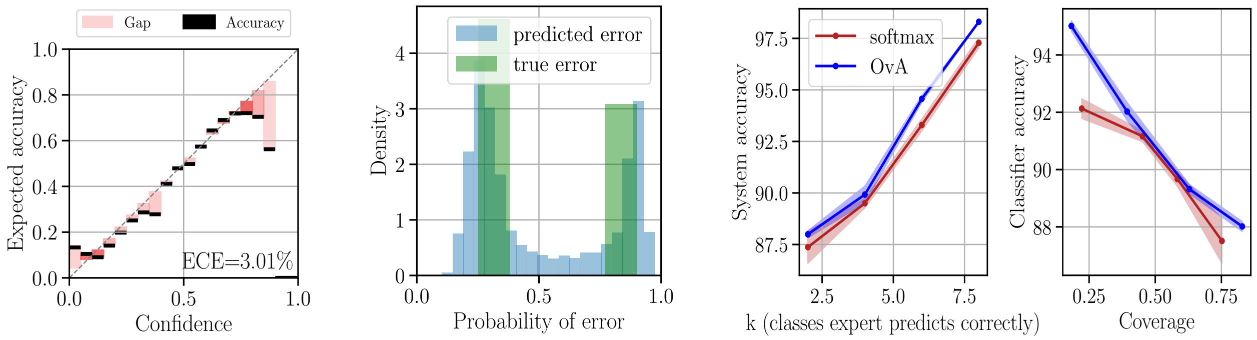
Given the difficulties in calibrating the softmax parameterization, we now consider an alternative. We propose a one-vs-all parameterization (a.k.a. one-vs-rest). We show that the accompanying loss function is calibrated as well as a consistent surrogate for the $0-1$ loss. Thus, our novel loss enjoys the same benefits as Mozannar & Sontag's (2020) formulation without its drawbacks.

### 4.1. One-vs-All-Based Surrogate Loss

We propose the following one-vs-all-based surrogate for the same L2D problem described in Section 2. Again assume we have $K + 1$ functions $g_1(\mathbf{x}), \ldots, g_K(\mathbf{x}), g_{\perp}(\mathbf{x})$ such that $g : \mathcal{X} \mapsto \mathbb{R}$. And again, we observe training data of the form $\mathcal{D} = \{\boldsymbol{x}_n, y_n, m_n\}_{n=1}^{N}$. Our one-vs-all (OvA) surrogate loss takes the following point-wise form:

$$\begin{aligned}\psi_{\mathrm{OvA}}(g_1, \ldots, g_K, g_{\perp}; \boldsymbol{x}, y, m) = \\ \phi[g_y(\boldsymbol{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\boldsymbol{x})] \ + \\ \phi[-g_{\perp}(\boldsymbol{x})] + \mathbb{I}[m = y]\left(\phi[g_{\perp}(\boldsymbol{x})] - \phi[-g_{\perp}(\boldsymbol{x})]\right)\end{aligned} \tag{8}$$

where $\phi : \{\pm 1\} \times \mathbb{R} \mapsto \mathbb{R}_{+}$ is a binary surrogate loss. For instance, when $\phi$ is the logistic loss, we have $\phi[f(\boldsymbol{x})] = \log(1 + \exp\{-f(\boldsymbol{x})\})$. Our formulation is the OvA analog of Mozannar & Sontag's (2020) softmax-based loss. The $g$-functions are entirely the same; the difference is in how they are combined. Moreover, the classifier and rejector are computed exactly the same as in the softmax case: $h(\boldsymbol{x}) = \arg\max_{k \in [1,K]} g_k(\boldsymbol{x})$, $r(\boldsymbol{x}) = \mathbb{I}[g_{\perp}(\boldsymbol{x}) \geq \max_k g_k(\boldsymbol{x})]$. In the experiments, we found no need for a re-



(a) Reliability Diagram and ECE (b) Empirical Dist. of $1 - p_{\mathrm{m}}^{\mathrm{OvA}}(\boldsymbol{x})$ (c) Accuracy and Coverage

*Figure 2. Calibration and Accuracy of OvA Parameterization on CIFAR-10*: Subfigure (a) reports a reliability diagram and the expected calibration error (ECE) for $p_{\mathrm{m}}^{\mathrm{OvA}}(\boldsymbol{x})$ (Eq. 9). Darker bin shade means more samples in the bin. Subfigure (b) shows the distribution of risk estimates. Subfigure (c) reports the accuracy as a function of an expert with increasing expertise (left) and of varying coverage (right).

weighting parameter that is analogous to $\alpha$ in Mozannar & Sontag's (2020) loss. One can be introduced similarly by re-weighting the first two terms in Equation 8 when the expert is correct. Rifkin & Klautau (2004) found OvA classifiers to work just as well as other approaches for multiclass classification, but in terms of general performance, OvA might falter in cases of data scarcity or severe class imbalance.

We next turn to the probabilistic formulation of the rejector and classifier. Starting with the former, the OvA formulation directly estimates the probability that the expert is correct:

$$\mathbb{P}(\mathrm{m} = \mathrm{y}|\boldsymbol{x}) \approx p_{\mathrm{m}}^{\mathrm{OvA}}(\boldsymbol{x}) = (1 + \exp\{-g_{\perp}(\boldsymbol{x})\})^{-1}. \quad (9)$$

$p_{\mathrm{m}}^{\mathrm{OvA}}$ has the appropriate range of $(0, 1)$. Moving on to the classifier, the foremost downside of the OvA formulation is that we can no longer compute normalized probabilities for all classes. Rather, we can estimate only the probability of the most likely class:

$$\begin{aligned} \max_{k\in[1,K]} \mathbb{P}(\mathrm{y} = k|\boldsymbol{x}) &\approx \max_{k\in[1,K]} p_{k}^{\mathrm{OvA}}(\boldsymbol{x}) \\ &= \max_{k\in[1,K]} (1 + \exp\{-g_k(\boldsymbol{x})\})^{-1}. \end{aligned} \quad (10)$$

Hence, we can evaluate the confidence calibration of the OvA classifier but not its distribution calibration. This is a worthwhile trade off for having an appropriate estimator for $\mathbb{P}(\mathrm{m} = \mathrm{y}|\boldsymbol{x})$ since distribution calibration is nearly impossible to achieve anyway (Zhao et al., 2021). Multiclass-to-binary reduction has been shown to be an effective calibration strategy for traditional classifiers (Gupta & Ramdas, 2022).

### 4.2. Theoretical Analysis

We now justify the OvA loss by showing that, like Mozannar & Sontag's (2020) loss, ours is a consistent surrogate for the $0 - 1$ L2D loss (Equation 2). On one hand, this result is not surprising since our loss is the natural OvA-analog of the softmax-based loss. However, we cannot construct our consistency proof in the same direct manner as Mozannar & Sontag (2020). When we differentiate with respect to a particular $g(\boldsymbol{x})$, the other $g$'s drop from the OvA loss (but not from the softmax loss). We proceed instead by the method of *error correcting output codes* (ECOC) (Dietterich & Bakiri, 1995; Langford et al., 2005; Allwein et al., 2001; Ramaswamy et al., 2014), a general technique for reducing multiclass problems to multiple binary problems. We sketch the approach here and provide the details in Appendix C.2.

ECOC requires that we construct a coding matrix, which for our case is $\mathbf{M} \in \{-1, +1\}^{K\times(K+1)}$ with $K$ being the number of classes in the multiclass problem. Each column then corresponds to a binary problem. The entries of the matrix are determined as follows. The $K \times K$ sub-matrix $\mathbf{M}_{1:K,1:K}$ has $+1$ along its diagonal and $-1$ on the off-diagonal. The entries in the $K + 1$-th column are given by the function $m_{y,K+1}(\mathrm{m}) = (-1 + 2\mathbb{I}[y = \mathrm{m}])$. Now that we have constructed the coding matrix, we use Equation 1 from Ramaswamy et al. (2018) to derive the closed form expression of the surrogate loss in Appendix B (Equation 8). We then arrive at our final result:

**Theorem 4.1.** *For a strictly proper binary composite loss $\phi$ with a well-defined continuous inverse link function $\gamma^{-1}$, $\psi_{OvA}$ (Equation 8) is a calibrated surrogate for the $0 - 1$ learning to defer loss (Equation 2).*

The complete proof is in Appendix C.2. We also provide background information on calibration and consistency in Appendix A, which includes a discussion of proper binary composite losses. Lastly, assuming *minimizability* (Steinwart, 2007)—i.e. that our hypothesis class is sufficiently large (all measurable functions)—the calibration result from

Theorem 4.1 implies consistency.

**Corollary 4.2.** *Assume that $g \in \mathcal{F}$, where $\mathcal{F}$ is the hypothesis class of all measurable functions. Minimizability (Steinwart, 2007) is then satisfied for $\psi_{OvA}$, and it follows that $\psi_{OvA}$ is a consistent surrogate for the $0-1$ learning to defer loss (Equation 2).*

Thus, $\psi_{\mathrm{OvA}}$ is also a consistent loss function for L2D. This means that the minimizer of the proposed loss function $\psi_{\mathrm{OvA}}$ over all measurable functions agrees with the Bayes optimal classifier and rejector (Equation 4).

## 5. Related Work

Learning with a reject option (a.k.a. rejection learning) is a long-studied problem, dating back to (at least) Chow (1957)'s work on an optimal learning rule for a fixed rejection rate. This initial work then stimulated a range of follow-up approaches, which can be categorized into two types: confidence-based (Bartlett & Wegkamp, 2008; Yuan & Wegkamp, 2010; Jiang et al., 2018; Grandvalet et al., 2009; Ramaswamy et al., 2018; Ni et al., 2019) and classifier-rejector (Cortes et al., 2016a;b). The classifier-rejector approach has been well-studied for binary classification and resulted in theoretical guarantees (Cortes et al., 2016a;b). Ni et al. (2019) was the first to seriously study the multi-class formulation and found that the existing theory was hard to extend to this more general case. Most recently, Charoenphakdee et al. (2021) proposed a surrogate loss for rejection learning for general classification, taking inspiration from cost-sensitive learning.

For safety-critical applications, rejection learning is a promising paradigm. However, its learning procedure completely ignores the downstream experts who will eventually make decisions for the rejected samples. Madras et al. (2018) introduced an adaptive rejection framework termed *learning to defer* (L2D). L2D aims to directly model the interaction between the (usually human) decision makers and the autonomous system. Madras et al. (2018) propose a mixture of experts model for this end. Raghu et al. (2019) approaches the same problem by learning a classifier and comparing the expert's certainty and the classifier's certainty, deferring if the latter is lower. Wilder et al. (2020) use the same mixture of experts framework as Madras et al. (2018) and apply the same confidence-based deferral policy as Raghu et al. (2019).

In the work closest to ours, Mozannar & Sontag (2020) study the L2D classification problem with generality, finding the algorithms proposed by Madras et al. (2018) are inconsistent. They also study the limitation of confidence-based approaches (Raghu et al., 2019). Moreover, they propose the first consistent loss for multiclass L2D, establishing the importance of having a consistent surrogate. Our work, on the other hand, is the first to study the calibration of confidence estimates within L2D systems.

## 6. Experiments

We perform two types of experiments. In the first, we verify that our OvA loss results in a better calibrated model for $\mathbb{P}(\mathrm{m} = \mathrm{y}|\boldsymbol{x})$ than Mozannar & Sontag's (2020) loss. We verify this in a CIFAR-10 simulation in Section 6.1. We then show that the softmax loss's mis-calibration has consequences for safety-critical decision making. We train models for each loss on `HAM10000`, a data set for the diagnosis of skin lesions (Tschandl et al., 2018), showing in Section 6.2 that our OvA model assesses risk more accurately than its softmax counterpart.

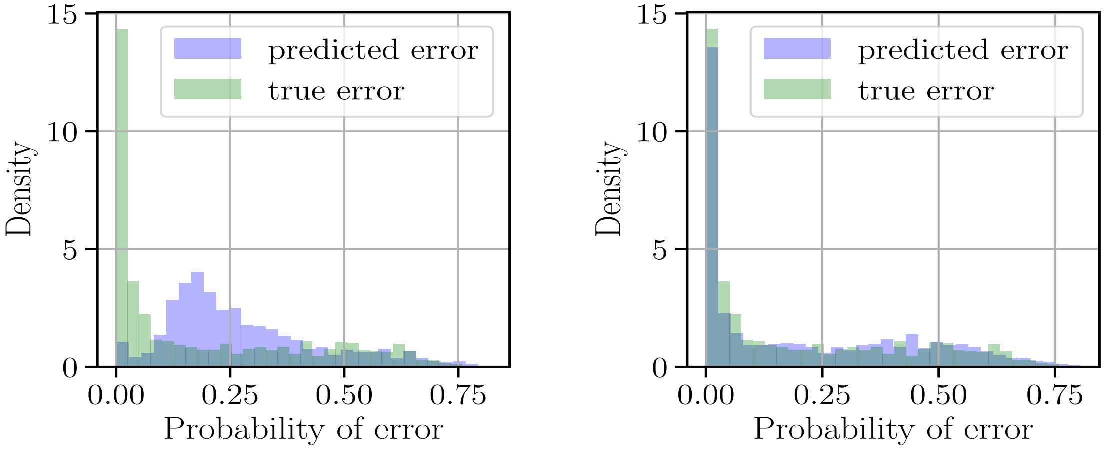
In the second type of experiment shown in Section 6.3, we assess the overall accuracy on hate speech detection, galaxy classification, and skin lesion diagnosis. We compare our OvA-based method to Mozannar & Sontag's (2020) as well as other state-of-the-art methods, such as differentiable triage (Okati et al., 2021). We find that our OvA models are at least competitive with, if not superior to, the best-performing competitor in all experiments. Thus, our OvA method enjoys the benefits of calibration without any sacrifice to predictive performance.

In all our implementations of Mozannar & Sontag's (2020) loss, we set the re-weighting parameter as $\alpha = 1$. Although Mozannar & Sontag (2020) observe better performance when tuning $\alpha$, $\alpha = 1$ is the *only* value for which their surrogate is provably consistent. The same is true for our loss and so our OvA surrogate does not include re-weighting either. Comparing these losses in their 'purest' forms is appropriate since our primary experimental concern is validating calibration. For all OvA results, we use the logistic loss as the surrogate loss for binary classification. Results are averaged over re-runs with six different random seeds. Our software implementations are publicly available.[1]

### 6.1. Comparison to the Softmax Loss on `CIFAR-10`

**Data, Model, and Training** We use the standard train-test splits of `CIFAR-10` (Krizhevsky, 2009). We further partition the training split by $90\% - 10\%$ to form training and validation sets, respectively. We simulate the expert demonstrations from the training labels, as is described in detail below. We use the same neural network and training settings for both the OvA and softmax methods. Following Mozannar & Sontag (2020), we use a wide residual networks (Zagoruyko & Komodakis, 2016) to parameterize the $g(\boldsymbol{x})$ functions. We train a 28-layer network using stochastic gradient descent (SGD) with momentum and a cosine annealing schedule for the learning rate. We employ early

[1] https://github.com/rajevv/OvA-L2D



(a) Risk for Model Trained with Softmax Surrogate

(b) Risk for Model Trained with OvA Surrogate (Ours)

*Figure 3. Risk for Softmax vs OvA models on HAM10000*: Subfigure (a) reports the distribution of risks for the softmax method: $1 - p_{\mathrm{m}}(\boldsymbol{x})$. Subfigure (b) reports the distribution of risks for the OvA method: $1 - p_{\mathrm{m}}^{\mathrm{OvA}}(\boldsymbol{x})$. We observe markedly more overlap for the latter. The Wasserstein distance between the empirical and true error distributions is $8.02 \pm 1.37$ for OvA and $26.72 \pm 1.77$ for softmax.

stopping, terminating training if the validation loss does not improve for 20 epochs. Additional experimental details can be found in Appendix F.

**OvA Method's Calibration** We now test our OvA method's calibration in the same experimental setting used to test the softmax method in Section 3. To reiterate, the expert has a 75% chance of being correct on the first five classes and random chance on the last five. Figure 2a reports a reliability diagram and the ECE. Comparing to the softmax results in Figure 1b, our OvA loss produces a model that has an over fifty percent reduction in ECE: 7.58% for softmax, 3.01% for OvA. Figure 2b reports the empirical distribution of error estimates: $1 - p_{\mathrm{m}}^{\mathrm{OvA}}(\boldsymbol{x})$. Unlike the corresponding softmax results in Figure 1c, the OvA method produces sharper modes nearer to the true error values. Moreover, OvA does not have a false mode at zero.

**Comparing Calibration Across Estimators** We next test OvA's calibration against not only the softmax but also the proxy function $p_{\perp}$ from Equation 5. We consider two types of experts: a useful one and a random one. The useful one is an *oracle* (i.e. always correct) for the first seven classes and predicts randomly for the last three classes. The random expert predicts uniformly over all classes. Moreover, we consider when the data is useful, i.e. the original CIFAR-10 training split, and when it is random, i.e. training labels are uniformly random.

ECE results for the OvA (Eq. 9), softmax (Eq. 6), and proxy (Eq. 5) methods are reported in Table 1. OvA has the best ECE in all but one case—the one in which both expert and data are random. Yet the $p_{\perp}$ proxy is clearly not a viable estimator since it has an egregious ECE of 37.15% when both data and expert are useful. Furthermore, its ECE is an even worse 39.06% when the expert is useful and data is random. In general, the softmax's true estimator $p_{\mathrm{m}}$ is competent but still consistently worse than the OvA estimator. We compare the ECE values for the classifier for OvA and softmax in Table 2 in Appendix D.1.

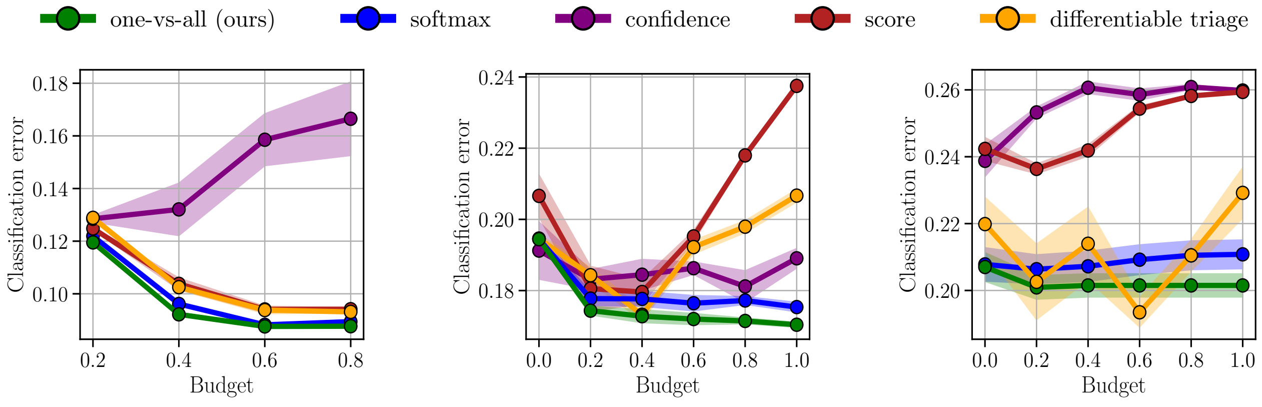
Expected Calibration Error (%) on CIFAR-10

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

*Table 1. ECE (%) on CIFAR-10 Simulation.* We compare calibration across the three parameterizations considered: OvA (Eq. 9), softmax (Eq. 6), and proxy (Eq. 5).

**System Accuracy and Coverage** For the final `CIFAR-10` experiment, we compare the OvA system's accuracy to the softmax's. The expert in this case has a 70% chance of being correct if the image belongs to the classes $[1, k]$ and random chance if it belongs to classes $[k, 10]$. We then vary $k$ from $k = 2$ to $k = 8$. The left plot in Figure 2c shows accuracy vs $k$. Our OvA model (blue) has a modest but consistent advantage over the softmax model (red).

The right plot in Figure 2c reports the accuracy vs coverage, where coverage is the proportion of samples that the system



(a) HateSpeech

(b) Galaxy-Zoo

(c) HAM10000

*Figure 4. Accuracy on HateSpeech, Galaxy-Zoo, and HAM10000*: The subfigures report the classification error of OvA method, softmax method, and baselines for three data sets. OvA (green) is competitive in all cases and is superior for `HateSpeech` and `Galaxy-Zoo`.

has *not* deferred. *Classifier accuracy* is the accuracy on the non-deferred samples. An L2D system ideally should have high coverage and high accuracy. Again, the results show the OvA method's (blue) advantage at most coverage levels. Note the OvA's significant superiority at low coverage ($0.2 - 0.3$). Here the rejector must carefully choose which instances to pass to the classifier. We conjecture that OvA's success is likely due to OvA's superior calibration in estimating when to defer.

### 6.2. Risk Assessment on `HAM10000`

**Data, Model, and Expert** We again study risk assessment but this time for a high-stakes medical task. `HAM10000` (Tschandl et al., 2018) is a data set of 10,015 dermatoscopic images containing seven categories of human skin lesions. We partition the data into 60% training, 20% validation, and 20% test splits. Each image includes metadata such as age, gender, and diagnosis type of the lesion. For our simulated expert model, we train an 8-layer MLPMixer (Tolstikhin et al., 2021). To simulate the expert having extra information, we input the image metadata into to the final feedforward layer. This model has a classification accuracy of 74% (see Table 3). For the classifier, we fine-tune a 34-layer residual network (ResNet34) (He et al., 2016), following Tschandl et al. (2020). We use data augmentations such as random cropping, reflection, and horizontal flipping.

**Results** Figure 3 visualizes the expert's predicted error and the expert's true error on the `HAM10000` test set. Subfigure (a) shows results for the softmax method and (b) for our OvA method. We restrict $p_{\mathrm{m}}(\boldsymbol{x}) \in (0, 1]$ for the softmax surrogate. The gap between the predicted and true error is substantially reduced for OvA. We confirm this quantitatively by computing the Wasserstein distance between the true and predicted error. The distance is $8.02 \pm 1.37$ for OvA and $26.72 \pm 1.77$ for softmax. OvA provides clearly superior estimates of the expert's error.

### 6.3. Overall Accuracy

**Data** Lastly, we examine the OvA method's classification error on three real-world tasks: `HAM10000` (Tschandl et al., 2018) for diagnosing skin lesions, `Galaxy-Zoo` (Bamford et al., 2009) for scientific discovery, and `HateSpeech` (Davidson et al., 2017) for detecting offensive language. Following Okati et al. (2021), we use a random sample of $10,000$ images for `Galaxy-Zoo`. We use 60% train, 20% validation, and 20% test splits for `HAM10000` and `HateSpeech`.

**Baselines** We compare the OvA- and softmax-based surrogates to three baselines. The first is *differentiable triage* (Okati et al., 2021), a policy-learning method. The other two baselines are confidence-based methods that do not enjoy theoretical guarantees. The two are the *score* baseline (Raghu et al., 2019) and the *confidence* baseline (Bansal et al., 2021). We give more details about the baselines and their implementation in Appendix E.

**Models and Experts** We closely follow the setup of Okati et al. (2021) for these experiments. Our base model is a 50-layer residual network (ResNet50) for `Galaxy-Zoo`. For `HateSpeech`, we first embed the tweet's text into a 100-dimensional feature vector using *fasttext* (Joulin et al., 2016). Our base model for `HateSpeech` is the text classification CNN developed by Kim (2014). For the surrogate loss methods, we sample the expert demonstrations from the expert model's predictive distribution. For training the surrogate models, we early stop if the validation loss does not improve for 20 epochs. We train the models using Adam (Kingma & Ba, 2015), a cosine-annealed learning rate, and a warm-up period of 5 epochs. For other baselines, we use the same experimental setup as Okati et al. (2021).

**Results** Figure 4 reports the classification accuracy for each data set and each baseline as a function of the *budget*. The budget is the upper limit on the proportion of samples the system can defer to the expert. The OvA surrogate is competitive among all baselines for the range of budgets considered. This shows that the OvA does not sacrifice accuracy for improved calibration. Rather, our model enjoys the benefits of both predictive performance and uncertainty quantification. OvA's performance is also quite stable across random seeds.

## 7. Conclusions

In human-AI collaboration, it is vital that the system be reliable and trusted by the human. Having a well-calibrated system—one that is a good forecaster—can help engender this trust. Our work investigates confidence calibration for the softmax surrogate loss, which previously was the only consistent surrogate for multiclass L2D. We find that the softmax parameterization suffers from degenerate estimates of expert correctness. We solve this issue by deriving an alternative loss function that is also consistent. We experimentally show that our one-vs-all loss results in models better calibrated than those trained with the softmax-based surrogate. In future work, we plan to investigate calibration in non-surrogate-based L2D systems, such as differentiable triage (Okati et al., 2021).

## Acknowledgements

We thank Hussein Mozannar for helpful discussions and Daniel Barrejón for helping create the figures. This publication is part of the project *Continual Learning under Human Guidance* (VI.Veni.212.203), which is financed by the Dutch Research Council (NWO). This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

## References

Allwein, E. L., Schapire, R. E., and Singer, Y. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1: 113–141, 2001.

Bamford, S. P., Nichol, R. C., Baldry, I. K., Land, K., Lintott, C. J., Schawinski, K., Slosar, A., Szalay, A. S., Thomas, D., Torki, M., Andreescu, D., Edmondson, E. M., Miller, C. J., Murray, P., Raddick, M. J., and Vandenberg, J. Galaxy Zoo: the dependence of morphology and colour on environment*. *Monthly Notices of the Royal Astronomical Society*, 393(4):1324–1352, 02 2009. ISSN 0035-8711. doi: 10.1111/j.1365-2966.2008.14252.x.

Bansal, G., Nushi, B., Kamar, E., Horvitz, E., and Weld, D. S. Is the most accurate ai the best teammate? optimizing ai for teamwork. In *AAAI Conference on Artificial Intelligence*, 2021.

Bartlett, P. L. and Wegkamp, M. H. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, jun 2008. ISSN 1532-4435.

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.

Buja, A., Stuetzle, W., and Shen, Y. Loss functions for binary class probability estimation and classification: Structure and applications. *Technical Report*, 2005.

Chalkidis, I., Androutsopoulos, I., and Aletras, N. Neural legal judgment prediction in English. In *Meeting of the Association for Computational Linguistics*, 2019.

Charoenphakdee, N., Cui, Z., Zhang, Y., and Sugiyama, M. Classification with rejection based on cost-sensitive classification. In *International Conference on Machine Learning*, 2021.

Chow, C. K. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, 6:247–254, 1957.

Cortes, C., DeSalvo, G., and Mohri, M. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, 2016a.

Cortes, C., DeSalvo, G., and Mohri, M. Boosting with abstention. In *Advances in Neural Information Processing Systems*, 2016b.

Davidson, T., Warmsley, D., Macy, M. W., and Weber, I. Automated hate speech detection and the problem of offensive language. In *International AAAI Conference on Web and Social Media*, 2017.

Dietterich, T. G. and Bakiri, G. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(1):263–286, jan 1995. ISSN 1076-9757.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., and Canu, S. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems*, 2009.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. doi: https://doi.org/10.1002/rob.21918.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Gupta, C. and Ramdas, A. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Jiang, H., Kim, B., Guan, M. Y., and Gupta, M. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, 2018.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

Kadampur, M. A. and Al Riyaee, S. Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images. *Informatics in Medicine Unlocked*, 18:100282, 2020.

Kim, Y. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Kull, M., Perello-Nieto, M., Kängsepp, M., Filho, T. S., Song, H., and Flach, P. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, 2019.

Langford, J., Tti-Chicago, Net, J., and Beygelzimer, A. Sensitive error correcting output codes. In *Conference on Learning Theory*, 2005.

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

Mozannar, H. and Sontag, D. A. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, 2020.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Conference on Computer Vision and Pattern Recognition*, 2015.

Ni, C., Charoenphakdee, N., Honda, J., and Sugiyama, M. On the calibration of multiclass classification with rejection. In *Advances in Neural Information Processing Systems*, 2019.

Okati, N., De, A., and Gomez-Rodriguez, M. Differentiable learning under triage. In *Advances in Neural Information Processing Systems*, 2021.

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*, 2019.

Ramaswamy, H. G., Srinivasan Babu, B., Agarwal, S., and Williamson, R. C. On the consistency of output code based learning algorithms for multiclass learning problems. In *Conference on Learning Theory*, 2014.

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

Reid, M. D. and Williamson, R. C. Composite binary losses. *Journal of Machine Learning Research*, 11(83): 2387–2422, 2010.

Rifkin, R. and Klautau, A. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5: 101–141, 2004.

Steinwart, I. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., and Dosovitskiy, A. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*, 2021.

Tschandl, P., Rosendahl, C., and Kittler, H. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

Tschandl, P., Rinner, C., Apalla, Z., Argenziano, G., Codella, N. C. F., Halpern, A. C., Janda, M., Lallas, A., Longo, C., Malvehy, J., Paoli, J., Puig, S., Rosendahl, C., Soyer, H. P., Zalaudek, I., and Kittler, H. Human–computer collaboration for skin cancer recognition. *Nature Medicine*, pp. 1–6, 2020.

Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., and Schön, T. Evaluating model calibration in classification. In *Conference on Artificial Intelligence and Statistics*, 2019.

Wilder, B., Horvitz, E., and Kamar, E. Learning to complement humans. In *International Joint Conference on Artificial Intelligence*, 2020.

Yuan, M. and Wegkamp, M. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11(5):111–130, 2010.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *British Machine Vision Conference*, 2016.

Zhao, S., Kim, M., Sahoo, R., Ma, T., and Ermon, S. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 2021.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., and Sun, M. Legal judgment prediction via topological learning. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

Zoabi, Y., Deri-Rozov, S., and Shomron, N. Machine learning-based prediction of covid-19 diagnosis based on symptoms. *npj Digital Medicine*, 4(1):1–5, 2021.

# A. A Primer on Calibration and Consistency for Classification

## A.1. A General Classification Problem and Surrogate Losses

Given $\mathcal{X} \subseteq \mathbb{R}^n$ as the input space, and $\mathcal{Y} = [n]$ as the output label space, we have an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. The output prediction label space is $\hat{\mathcal{Y}} = [k]$, and in general classification problem $k$ and $n$ can be different. The goal of the classification problem then is to learn a mapping $h : \mathcal{X} \to \hat{\mathcal{Y}}$. We assess the performance of the prediction function $h$ via a loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to \mathbb{R}_+$, and we aim to find $h$ with small $\ell$ *-risk* which is defined as follows:

$$\mathcal{R}^{\ell}_{\mathcal{D}}[h] = \mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}} \left[\ell\left(y, h\left(\boldsymbol{x}\right)\right)\right] \tag{11}$$

We define the *Bayes $\ell$-risk* $\mathcal{R}^{\ell,*}_{\mathcal{D}}$ as the minimal *$\ell$-risk* one can hope to achieve for the distribution $\mathcal{D}$, i.e $\mathcal{R}^{\ell,*}_{\mathcal{D}} := \inf_{h:\mathcal{X}\to\hat{\mathcal{Y}}} \mathcal{R}^{\ell}_{\mathcal{D}}[h]$. In practical settings , the classification learning problem assumes access to the training sample $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ drawn independently and identically distributed from $\mathcal{D}$, and the learning algorithm seeks to learn $h$ by minimizing an empirical version of $\ell$ *-risk* $\hat{\mathcal{R}}^{\ell}_{\mathcal{D}}[h]$. For $h \in \mathcal{H}$, $\hat{\mathcal{R}}^{\ell}_{\mathcal{D}}[h]$ is defined as

$$\hat{\mathcal{R}}^{\ell}_{\mathcal{D}}[h] = \frac{1}{N}\sum_{i=1}^{N} \ell\left(y_i, h\left(\boldsymbol{x}_i\right)\right) \tag{12}$$

An important notion of success for such a learning algorithm is the convergence of $\mathcal{R}^{\ell}_{\mathcal{D}}[h_S] \to \mathcal{R}^{\ell,*}_{\mathcal{D}}$, i.e. when the learning algorithm receives increasingly large sample $S \sim \mathcal{D}^N$, the *$\ell$-risk* of the function $h_S$ returned by the learning algorithm converges in probability to the *Bayes $\ell$-risk*, written formally as

$$\forall \epsilon > 0 \; P_{S\sim\mathcal{D}^N} \left(\mathcal{R}^{\ell}_{\mathcal{D}}[h_S] > \mathcal{R}^{\ell,*}_{\mathcal{D}} + \epsilon\right) \to 0 \text{ as } N \to \infty \tag{13}$$

However, minimizing the *$\ell$-risk* (similarly, empirical *$\ell$-risk*) is computationally difficult for some classes of loss functions. For instance, for the misclassification loss $\ell_{0-1} : (y, \hat{y}) \mapsto \mathbb{I}(y \neq \hat{y})$, computationally minimizing $\ell - risk$ is NP-hard. Thus, a surrogate loss $\psi(\cdot)$ over a surrogate prediction space $\mathcal{C} \subseteq \mathbb{R}^k$ is generally employed as a replacement for the target loss $\ell(\cdot)$.

For a surrogate prediction space $\mathcal{C} \subseteq \mathbb{R}^k$, a surrogate loss $\psi : \mathcal{Y} \times \mathcal{C} \to \mathbb{R}_+$, the goal is to learn a function $f : \mathcal{X} \to \mathcal{C}$ over some suitable class of functions $\mathcal{F}$, and a suitable decoding function $g : \mathcal{C} \to \hat{\mathcal{Y}}$. We then have the usual notions of $\mathcal{R}^{\psi}_{\mathcal{D}}[f]$ and $\mathcal{R}^{\psi,*}_{\mathcal{D}}$. An important question in such a setting is whether the convergence $\mathcal{R}^{\psi}_{\mathcal{D}}[f_S] \to \mathcal{R}^{\psi,*}_{\mathcal{D}}$ implies the convergence $\mathcal{R}^{\ell}_{\mathcal{D}}[g \circ f] \to \mathcal{R}^{\ell,*}_{\mathcal{D}}$. A positive answer to this question is necessary for the success of the classification problem learned by minimizing a surrogate loss $\psi(\cdot)$, and it is formally known as the *consistency* of the surrogate loss $\psi(\cdot)$ w.r.t. the target loss $\ell(\cdot)$ as defined below:

**Definition A.1.** ($\mathcal{F}$-Consistency). A surrogate loss function $\psi(\cdot)$ is said to be $\mathcal{F}$-consistent with respect to the loss function $\ell(\cdot)$ if for any sequence of functions $f_n \in \mathcal{F}$

$$\mathcal{R}^{\psi}_{\mathcal{D}}[f_n] \to \mathcal{R}^{\psi,*}_{\mathcal{D}} \implies \mathcal{R}^{\ell}_{\mathcal{D}}[g \circ f_n] \to \mathcal{R}^{\ell,*}_{\mathcal{D}} \tag{14}$$

for all distributions $\mathcal{D}$.

Define $\eta_y\left(\boldsymbol{x}\right) = \mathbb{P}(\mathrm{y} = y|\mathbf{x} = \boldsymbol{x})$ for each $y \in \mathcal{Y}$. $\boldsymbol{x}$ and $y$ are the realizations of the random variables $\mathbf{x}$ and y respectively over $\mathcal{X} \times \mathcal{Y}$. Then, we can write $\mathcal{R}^{\ell}_{\mathcal{D}}[h]$ as

$$\mathcal{R}^{\ell}_{\mathcal{D}}[h] = \mathbb{E}_{\boldsymbol{x}\sim\mathbf{x}} \left[\sum_{y=1}^{n} \eta_y\left(\boldsymbol{x}\right) \ell\left(y, h\left(\boldsymbol{x}\right)\right)\right] = \mathbb{E}_{\boldsymbol{x}\sim\mathbf{x}} \left[\boldsymbol{\eta}\left(\boldsymbol{x}\right)^T \boldsymbol{\ell}\left(h\left(\boldsymbol{x}\right)\right)\right] \tag{15}$$

where $\boldsymbol{\eta}(\boldsymbol{x}) = [\eta_1(\boldsymbol{x}), \eta_2(\boldsymbol{x}), \ldots, \eta_n(\boldsymbol{x})]^T$, and $\boldsymbol{\ell}(h(\boldsymbol{x})) = [\ell\left(\mathrm{y} = 1, h(\boldsymbol{x})\right), \ell\left(\mathrm{y}, h(\boldsymbol{x})\right), \ldots, \ell\left(\mathrm{y} = n, h(\boldsymbol{x})\right)]^T$. The quantity $\boldsymbol{\eta}(\boldsymbol{x})^T\boldsymbol{\ell}(h(\boldsymbol{x}))$ is known as the *inner $\ell - risk$* denoted as $\mathcal{C}^{\ell}_{\boldsymbol{\eta}(\boldsymbol{x}),\boldsymbol{x}}[h]$. More generally, $\forall \boldsymbol{x} \in \mathcal{X}$, $\forall \boldsymbol{\eta} \in [0,1]^n$, $\mathcal{C}^{\ell}_{\boldsymbol{\eta},\boldsymbol{x}}[h] := \boldsymbol{\eta}^T\boldsymbol{\ell}(\mathbf{h}(\boldsymbol{x}))$ is known as the *inner $\ell$-risk*. We also define *Bayes inner $\ell$-risk* $\mathcal{C}^{\ell,*}_{\boldsymbol{\eta},\boldsymbol{x}} := \inf_{h:\mathcal{X}\to\hat{\mathcal{Y}}} \mathcal{C}^{\ell}_{\boldsymbol{\eta},\boldsymbol{x}}[h]$. We can also define these quantities for the surrogate loss $\psi(\cdot)$. A property called *Calibration* of the *inner $\psi$-risk* of the surrogate loss $\psi(\cdot)$ w.r.t. *inner $\ell$-risk* is then the necessary condition for the *consistency* of the surrogate loss $\psi(\cdot)$ w.r.t $\ell(\cdot)$, and is usually a powerful tool for establishing and studying *consistency* for surrogate losses. It is formally defined as follows:

**Definition A.2** (Steinwart (2007))**.** ($\mathcal{F}$-Calibration). A surrogate loss function $\psi(\cdot)$ is said to be $\mathcal{F}$-calibrated with respect to the loss function $\ell(\cdot)$ if, for all $\epsilon > 0$, $\boldsymbol{\eta} \in [0,1]^n$, and $\boldsymbol{x} \in \mathcal{X}$, there exists $\delta > 0$ such that for any function $f \in \mathcal{F}$

$$\mathcal{C}^{\psi}_{\boldsymbol{\eta},\boldsymbol{x}}[f] < \mathcal{C}^{\psi,*}_{\boldsymbol{\eta},\boldsymbol{x}} + \delta \implies \mathcal{C}^{\ell}_{\boldsymbol{\eta},\boldsymbol{x}}[g \circ f] < \mathcal{C}^{\ell,*}_{\boldsymbol{\eta},\boldsymbol{x}} + \epsilon. \tag{16}$$

As stated before, $\mathcal{F}$-calibration is a necessary condition for $\mathcal{F}$-consistency. However, with the satisfaction of an additional condition called *minimizability*(Steinwart, 2007), $\mathcal{F}$-calibration also implies $\mathcal{F}$-consistency. We note that when $\mathcal{F} = \mathcal{F}_{\text{all}}$, i.e. when the hypothesis class consists of all measurable functions, *minimizability* condition is satisfied. Thus, in such a case, it is enough to verify the calibration property of the surrogate loss to ensure the consistency of the surrogate loss w.r.t. the target loss. Intuitively, a surrogate loss $\psi(\cdot)$ is said to be calibrated with respect to the target loss $\ell(\cdot)$ if minimizing $\psi(\cdot)$ results in a classifier $f$ with suitable decoding function $g$ whose *inner* $\ell$*-risk* is close to the *Bayes inner* $\ell$*-risk* for each $\boldsymbol{x} \in \mathcal{X}$. Moreover, with an additional condition of *minimizability*, calibration theoretically guarantees that for each $\boldsymbol{x} \in \mathcal{X}$, the optimal solution of the *inner* $\psi$*-risk* minimization problem agrees with the optimal solution function of the $\ell$*-risk* minimization problem evaluated at $\boldsymbol{x}$. We state some important results for Binary Classification in the next section, and refer the reader to Steinwart (2007) for more details.

### A.2. Calibration of Binary Surrogate Losses

Following the notation in the previous section, we have $\mathcal{Y} = \{-1, 1\}$. Here $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1 | \mathbf{x} = \boldsymbol{x})$. We define the *inner* $\ell$*-risk* as $\mathcal{C}^{\ell}_{\eta,\boldsymbol{x}}[h] = \eta\ell\left(1, h\left(\boldsymbol{x}\right)\right) + \left(1-\eta\right)\ell\left(-1, h\left(\boldsymbol{x}\right)\right)$. Similarly, we define *inner* $\psi$*-risk* for a surrogate loss $\psi : \mathcal{Y} \times \mathcal{C} \to \mathbb{R}_+$ acting on a surrogate prediction space $\mathcal{C} \subseteq \mathbb{R}$. The calibration of binary surrogate losses (especially margin-based losses) with respect to the misclassification loss $\ell_{0-1}$ has been widely studied in the literature (Bartlett et al., 2006). In this section, we state some of the results in this direction.

**Definition A.3** (Bartlett et al. (2006))**.** For a surrogate prediction space $\mathcal{C} \subseteq \mathbb{R}$, we say a binary classification surrogate loss $\psi : \mathcal{Y} \times \mathcal{C} \to \mathbb{R}_+$ is classification-calibrated if, for any $\eta \neq \frac{1}{2}$, we have

$$\inf_{f(\boldsymbol{x})\left(\eta - \frac{1}{2}\right)} \mathcal{C}^{\psi}_{\eta,\boldsymbol{x}}[f] > \inf_{f(\boldsymbol{x})} \mathcal{C}^{\psi}_{\eta,\boldsymbol{x}}[f] \tag{17}$$

The above definition states that minimizing a calibrated surrogate loss $\psi(\cdot)$ can give us the Bayes optimal binary classifier. It is well known that a convex $\psi(\cdot)$ is classification calibrated iff $\psi$ is differentiable in second argument at 0, and $\psi^{'}\left(\cdot, 0\right) < 0$ (Bartlett et al., 2006).

### A.3. Binary Proper Losses and Proper Composite Surrogate Losses

In this section, we briefly review binary proper losses and proper composite surrogate losses. Recall the definition of $\mathcal{C}^{\psi}_{\eta,\boldsymbol{x}}[f]$ from Section A.2 for some loss $\psi : \{-1, 1\} \times \mathcal{C} \to \mathbb{R}_+$. To simplify the notation, we rewrite $\mathcal{C}^{\psi}_{\eta,\boldsymbol{x}}[f] = \mathcal{C}^{\psi}\left(\eta, f\left(\boldsymbol{x}\right)\right) = \mathcal{C}^{\psi}\left(\eta, u\right)$ where $u = f\left(\boldsymbol{x}\right)$. Next, we define *proper composite losses*.

**Definition A.4** (Ramaswamy et al. (2014))**.** For $\mathcal{C} \subseteq \mathbb{R}$, a surrogate loss function $\psi : \{-1, 1\} \times \mathcal{C} \to \mathbb{R}_+$ is called proper composite loss if there exists a strictly increasing link function $\gamma : [0, 1] \to \mathcal{C}$ such that:

$$\gamma\left(p\right) \in \arg\min_{u \in \mathcal{C}} \mathcal{C}^{\psi}\left(p, u\right), \forall\, p \in [0, 1]$$

If the above minimizer is unique for all $p \in [0, 1]$, then we call the surrogate loss strictly proper composite loss.

An important property of strictly proper composite losses is that their minimization leads to Fisher consistent class probability estimates (Buja et al., 2005). Logistic Loss ($\psi\left(y, u\right) = \log\left(1 + \exp\left(-yu\right)\right)$ is a common example of a strictly composite proper composite loss with the inverse link function $\gamma^{-1} = \frac{1}{1+\exp(-u)}$. Thus, if $f : \mathcal{X} \to \mathbb{R}$ is learnt by minimizing the logistic loss, then $\hat{p}_{\boldsymbol{x}} = \frac{1}{1+\exp(-f(\boldsymbol{x}))}$ acts as a class probability estimate. Furthermore, strictly proper composite losses are classification calibrated (Reid & Williamson, 2010). Thus, based on the definition of calibration of the binary surrogate losses(Section A.2), we can write the final predictor learnt by minimizing the logistic loss as:

$$h(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x})) = \text{sign}(\hat{p}(\boldsymbol{x}) - \frac{1}{2}) \tag{18}$$

### A.4. Code Based Surrogates for Multiclass Classification

Code Based methods are a class of classification techniques where some code matrix is used to decompose the multiclass classification problem into multiple binary classification problems. Mention could be made of error-correcting coding mechanism (Dietterich & Bakiri, 1995; Langford et al., 2005; Allwein et al., 2001). We briefly describe the setup here, and refer the reader to Ramaswamy et al. (2014) for full details. The goal of such a code based mechanism it to use a code matrix $\mathbf{M} = \{\pm 1, 0\}^{n\times k}$ to decompose a $n$-class classification problem into $k$ binary classification problems. Following the notation from Section A.1, we use $\mathbf{M}$ to split the training sample $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ into $k$-training samples $\tilde{S}_j$ for each $j \in [k]$ such that $\tilde{S}_j = \{(\boldsymbol{x}_i, M_{y_i,j})\,; i \in [1, N], M_{y_i,j} \neq 0\}$. Thus, each $\tilde{S}_j$ is a subset from the original $S$ with output (binary)labels replaced provided by the $\mathbf{M}$. For $\mathcal{C} \subseteq \mathbb{R}$, we use these $\tilde{S}_j$ to learn a $k$-binary classifiers $f_j : \mathcal{X} \to \mathcal{C}$. Thus, for each $\boldsymbol{x} \in \mathcal{X}$, we get a prediction $f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})] \in \mathbb{R}^k$. We then a use a suitable decoding function to map $f(\boldsymbol{x})$ to the original prediction space $\hat{\mathcal{Y}}$. If we use some suitable surrogate loss $\ell : \{-1, 1\} \times \mathcal{C} \to \mathbb{R}_+$, then intuitively, the whole code matrix based mechanism can be viewed as learning a function $f : \mathcal{X} \to \mathcal{C}^k$ by minimizing a surrogate multiclass classification loss $\psi : \mathcal{Y} \times \mathcal{C}^k \to \mathbb{R}_+$ given as

$$\psi(\mathbf{y}, \mathbf{u}) = \sum_{j=1}^{k} \left(\mathbb{I}(M_{yj} = 1)\,\ell(1, u_j) + \mathbb{I}(M_{yj} = -1)\,\ell(-1, u_j)\right) \tag{19}$$

Obviously, we care about the consistency of such a surrogate loss $\psi(\cdot)$ for a successful classification algorithm. Ramaswamy et al. (2014) analyze the conditions related to consistency and calibration of such a surrogate loss for general losses.

## B. One-vs-All surrogate Loss for L2D

We derive the closed-form expression for surrogate loss $\psi_{\mathrm{OvA}}$ using the procedure described in Appendix A.4 for the code matrix $\mathbf{M}$ defined Section 4. Following the notation from Appendix A.4, we have $n = K$ and $k = K + 1$ for our L2D problem. For the surrogate prediction space $\mathbb{R}$, and $g_y : \mathcal{X} \to \mathbb{R}, y \in \mathcal{Y}$ and $g_\perp : \mathcal{X} \to \mathbb{R}$ and $\boldsymbol{g}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_\perp(\mathbf{x})]$, we can use $\mathbf{M}$ to derive the closed form expression for the surrogate loss $\psi : \mathcal{Y} \times \mathbb{R}^{n+1} \to \mathbb{R}$ as follows:

1. **Case 1:** $\psi(\boldsymbol{g}; \boldsymbol{x}, y, m)$ for $y$ such that $\mathbb{I}[y \neq \mathrm{m}] = 1$
   In this case, we can follow the definition of $\mathbf{M}$ to gather that $m_{yj} = 1$ only if $j = y$. Thus, we can follow Eqn. 19, and get

$$\psi(\boldsymbol{g}; \boldsymbol{x}, y, m) = \phi[g_y(\boldsymbol{x})] + \sum_{\substack{y' \in \mathcal{Y} \cup \{\perp\} \\ y' \neq y}} \phi\left[-g_{y'}(\boldsymbol{x})\right]$$

2. **Case 2:** $\psi(\boldsymbol{g}; \boldsymbol{x}, y, m)$ for $y$ such that $\mathbb{I}[y = m] = 1$
   In this case, we have $m_{yy} = 1$ as well as $m_{y\perp} = 1$ where $\perp$ denotes the index $(K + 1)$. Thus,

$$\psi(\boldsymbol{g}; \boldsymbol{x}, y, m) = \phi[g_y(\boldsymbol{x})] + \phi[g_\perp(\boldsymbol{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi\left[-g_{y'}(\boldsymbol{x})\right]$$

Finally, we can combine both the cases to get

$$\psi(\boldsymbol{g}; \boldsymbol{x}, y, m) = \phi[g_y(\boldsymbol{x})] + \phi[-g_\perp(\boldsymbol{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi\left[-g_{y'}(\boldsymbol{x})\right] + \mathbb{I}[m = y]\left(\phi[g_\perp(\boldsymbol{x})] - \phi[-g_\perp(\boldsymbol{x})]\right)$$

where $\phi : \{\pm 1\} \times \mathbb{R} \to \mathbb{R}_+$ is a binary classification surrogate loss, and $\phi[g_y(\boldsymbol{x})] = \phi(1, g_y(\boldsymbol{x}))$. Similarly, $\phi[-g_y(\boldsymbol{x})] = \phi(-1, g_y(\boldsymbol{x}))$.

## C. Proofs

### C.1. Derivation of $p_{\mathrm{m}}(\boldsymbol{x})$ and $p_k(\boldsymbol{x})$ for the Softmax Surrogate Loss

Let $\mathcal{Y}^\perp = \mathcal{Y} \cup \{\perp\}$. From the proof of Theorem 1 of Mozannar & Sontag (2020), we have that for the (Bayes) optimal $g_1^*, \ldots, g_K^*, g_\perp^*$:

$$\begin{aligned} \frac{\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})}{1+\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})} &= \frac{\exp g_\perp^*(\boldsymbol{x})}{\sum_{y'\in\mathcal{Y}^\perp}\exp g_{y'}^*(\boldsymbol{x})} \\ &= p_\perp^*(\boldsymbol{x}) \end{aligned} \tag{20}$$

where $p_\perp^*(\boldsymbol{x})$ is the function we define in Equation 5 evaluated at the Bayes optimal $g$'s. Rearranging, we then have:

$$p_\perp^*(\boldsymbol{x}) = \frac{\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})}{1+\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})} = \frac{1}{\mathbb{P}^{-1}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})+1}. \tag{21}$$

Solving for $\mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x})$, we have:

$$\begin{aligned} \mathbb{P}(\mathrm{m}=\mathrm{y}|\boldsymbol{x}) &= \frac{1}{(p_\perp^*(\boldsymbol{x}))^{-1}-1} \\ &= \frac{p_\perp^*(\boldsymbol{x})}{1-p_\perp^*(\boldsymbol{x})} \end{aligned} \tag{22}$$

Similarly, from the proof of Theorem 1 of Mozannar & Sontag (2020), we have for the Bayes Optimal $g_k^*$, $k \in [K]$:

$$\frac{\mathbb{P}\left(\mathrm{y}=k|\boldsymbol{x}\right)}{1+\mathbb{P}\left(\mathrm{m}=\mathrm{y}|\boldsymbol{x}\right)} = \frac{\exp g_k^*(\boldsymbol{x})}{\sum_{y'\in\mathcal{Y}^\perp}\exp g_{y'}^*(\boldsymbol{x})} \tag{23}$$

$$\implies p_k(\boldsymbol{x}) = \mathbb{P}(\mathrm{y}=k|\boldsymbol{x}) = \frac{1}{1-p_\perp^*(\boldsymbol{x})}\frac{\exp g_k^*(\boldsymbol{x})}{\sum_{y'\in\mathcal{Y}^\perp}\exp g_{y'}^*(\boldsymbol{x})} \tag{24}$$

### C.2. Proof of Theorem 4.1

For $K+1$ surrogate prediction function $g_1(\mathbf{x}), \ldots, g_K(\mathbf{x}), g_\perp(\mathbf{x})$, and the binary classification surrogate $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$, the proposed one-vs-all (OvA) surrogate is has the following point-wise form:

$$\begin{aligned} &\psi_{\mathrm{OvA}}(g_1,\ldots,g_K,g_\perp;\boldsymbol{x},y,m) = \\ &\quad \phi[g_y(\boldsymbol{x})] \;+\; \phi[-g_\perp(\boldsymbol{x})] \;+ \sum_{y'\in\mathcal{Y},y'\neq y} \phi[-g_{y'}(\boldsymbol{x})]\mathbb{I}[m=y]\left(\phi[g_\perp(\boldsymbol{x})]-\phi[-g_\perp(\boldsymbol{x})]\right) \end{aligned} \tag{25}$$

We consider the point-wise *inner* $\psi$*-risk* for some $\mathbf{x} = \boldsymbol{x}$ written as follows:

$$\mathbb{E}_{\mathrm{y}|\mathbf{x}=\boldsymbol{x}}\mathbb{E}_{\mathrm{m}|\mathbf{x}=\boldsymbol{x},y}\psi_{\mathrm{OvA}}(g_1,\ldots,g_K,g_\perp;\boldsymbol{x},y,m) \tag{26}$$

We simplify the *inner* $\psi$*-risk* by expanding both the expectations below:

$$\begin{aligned} &\mathbb{E}_{\mathrm{y}|\mathbf{x}=\boldsymbol{x}}\mathbb{E}_{\mathrm{m}|\mathbf{x}=\boldsymbol{x},y}\psi_{\mathrm{OvA}}(g_1,\ldots,g_K,g_\perp;\boldsymbol{x},y,m) = \\ &\quad \mathbb{E}_{\mathrm{y}|\mathbf{x}=\boldsymbol{x}}\Bigg[\phi(g_y(\boldsymbol{x}))+\phi(-g_\perp(\boldsymbol{x}))+\sum_{y'\in\mathcal{Y},y'\neq y}\phi(-g_{y'}(\boldsymbol{x})) \\ &\quad +\sum_{m\in\mathcal{Y}}\mathbb{P}(\mathrm{m}=m|\mathbf{x}=\boldsymbol{x},\mathrm{y}=y)\mathbb{I}\left[m=y\right]\left[\phi(g_\perp(\boldsymbol{x}))-\phi(-g_\perp(\boldsymbol{x}))\right]\Bigg] \end{aligned} \tag{27}$$

Expanding the outer expectation, and $\eta_y\left(\boldsymbol{x}\right) = p\left(\mathrm{y} = y|\mathbf{x} = \boldsymbol{x}\right)$

$$
\begin{aligned}
&\mathbb{E}_{\mathrm{y}|\mathbf{x}=\boldsymbol{x}}\mathbb{E}_{\mathrm{m}|\mathbf{x}=\boldsymbol{x},y}\psi_{\mathrm{OvA}}(g_1,\ldots,g_K,g_\perp;\boldsymbol{x},y,m) = \\
&\quad \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right)\left[\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \sum_{y'\in\mathcal{Y},y'\neq y} \phi\left(-g_{y'}\left(\boldsymbol{x}\right)\right)\right] + \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right) \\
&\quad + \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right) \sum_{m\in\mathcal{Y}} \mathbb{P}\left(\mathrm{m} = m|\mathbf{x} = \boldsymbol{x}, \mathrm{y} = y\right)\mathbb{I}\left[m = y\right]\left[\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) - \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right)\right] \\
&\quad = \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right)\left[\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \sum_{y'\in\mathcal{Y},y'\neq y} \phi\left(-g_{y'}\left(\boldsymbol{x}\right)\right)\right] + \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right) \\
&\quad + \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right) \sum_{m\in\mathcal{Y}} \mathbb{P}\left(\mathrm{m} = y|\mathbf{x} = \boldsymbol{x}, \mathrm{y} = y\right)\left[\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) - \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right)\right] \\
&\quad = \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right)\left[\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \sum_{y'\in\mathcal{Y},y'\neq y} \phi\left(-g_{y'}\left(\boldsymbol{x}\right)\right)\right] + \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right) \\
&\quad + \underbrace{\sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right) \sum_{m\in\mathcal{Y}} \mathbb{P}\left(\mathrm{m} = y|\mathbf{x} = \boldsymbol{x}, \mathrm{y} = y\right)}_{\mathbb{P}(\mathrm{y}=\mathrm{m}|\mathbf{x}=\boldsymbol{x})}\left[\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) - \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right)\right] \\
&\quad = \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right)\left[\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \sum_{y'\in\mathcal{Y},y'\neq y} \phi\left(-g_{y'}\left(\boldsymbol{x}\right)\right)\right] + \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right) \\
&\quad + \mathbb{P}(\mathrm{y} = \mathrm{m}|\mathbf{x} = \boldsymbol{x})\left[\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) - \phi\left(-g_\perp\left(\boldsymbol{x}\right)\right)\right] \\
&\quad = \sum_{y\in\mathcal{Y}} \eta_y\left(\boldsymbol{x}\right)\left[\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \sum_{y'\in\mathcal{Y},y'\neq y} \phi\left(-g_{y'}\left(\boldsymbol{x}\right)\right)\right] + \mathbb{P}(\mathrm{y} = \mathrm{m}|\mathbf{x} = \boldsymbol{x})\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) \\
&\quad + \left(1 - \mathbb{P}\left(\mathrm{y} = \mathrm{m}|\mathbf{x} = \boldsymbol{x}\right)\right)\phi\left(-g_\perp\left(\boldsymbol{x}\right)\right)
\end{aligned}
$$

Using the usual notation $p_{\mathrm{m}}(\boldsymbol{x}) = p(\mathrm{y} = \mathrm{m}|\mathbf{x} = \boldsymbol{x})$, we can further rewrite the above equation in the following form,

$$
\begin{aligned}
&\mathbb{E}_{\mathrm{y}|\mathbf{x}=\boldsymbol{x}}\mathbb{E}_{\mathrm{m}|\mathbf{x}=\boldsymbol{x},y}\psi_{\mathrm{OvA}}(g_1,\ldots,g_K,g_\perp;\boldsymbol{x},y,m) = \\
&\quad \sum_{y\in\mathcal{Y}}\left[\eta_y\left(\boldsymbol{x}\right)\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \left(1 - \eta_y\left(\boldsymbol{x}\right)\right)\phi\left(-g_y\left(\boldsymbol{x}\right)\right)\right] + p_{\mathrm{m}}\left(\boldsymbol{x}\right)\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) + \left(1 - p_{\mathrm{m}}\right)\phi\left(-g_\perp\left(\boldsymbol{x}\right)\right)
\end{aligned}
\tag{28}
$$

The above expression says that we have $K + 1$ binary classification problems where the *inner* $\phi$*-risk* for the $i^{th}$ binary classification problem is given as $\eta_y\left(\boldsymbol{x}\right)\phi\left(g_y\left(\boldsymbol{x}\right)\right) + \left(1 - \eta_y\left(\boldsymbol{x}\right)\right)\phi\left(-g_y\left(\boldsymbol{x}\right)\right)$ when $i \in [K]$ and $p_{\mathrm{m}}\left(\boldsymbol{x}\right)\phi\left(g_\perp\left(\boldsymbol{x}\right)\right) + \left(1 - p_{\mathrm{m}}\left(\boldsymbol{x}\right)\right)\phi\left(-g_\perp\left(x\right)\right)$ when $i \in \{K+1\}$. This means that the point-wise minimizer of the inner $\psi$*-risk* can be analyzed in terms of the point-wise minimizer of the *inner* $\phi$*-risk* for each of the $K + 1$ binary classification problems we have. Denote the minimizer of point-wise *inner* $\psi_{\mathrm{OvA}}$*-risk* as $\boldsymbol{g}^*$, then the above decomposition means $g_i^*$ corresponds to the minimizer of the *inner* $\phi$*-risk* for the $i^{th}$ binary classification problem.

We know that the Bayes solution for the binary classification problem is $\mathrm{sign}\left(\eta(\boldsymbol{x}) - \frac{1}{2}\right)$ where $\eta(\boldsymbol{x})$ denotes $p(\mathrm{y} = 1|\mathbf{x} = \boldsymbol{x})$. Now when the binary surrogate loss $\phi$ is a strictly proper composite loss for binary classification, by the property of strictly proper composite losses, we have $\mathrm{sign}(g_y^*(\boldsymbol{x}))$ would agree with the Bayes solution of the Binary classification (refer Eqn. 18), i.e. $g_y^*(\boldsymbol{x}) > 0$ if $\eta_y\left(\boldsymbol{x}\right) > \frac{1}{2}$. And similarly $g_\perp^*\left(\boldsymbol{x}\right) \geq 0$ if $p_{\mathrm{m}}\left(\boldsymbol{x}\right) > \frac{1}{2}$. Furthermore, we have the existence of a continuous and increasing inverse link function $\gamma^{-1}$ for the binary surrogate $\phi$ with the property that $\gamma^{-1}\left(g_y^*\left(\boldsymbol{x}\right)\right)$ would converge to $\eta_y\left(\boldsymbol{x}\right)$. Similarly, $\gamma^{-1}\left(g_\perp^*\left(\boldsymbol{x}\right)\right)$ would converge to $p_{\mathrm{m}}\left(\boldsymbol{x}\right)$.

Using the above, we can establish the Bayes optimal decision for this minimizer $\boldsymbol{g}^*$ using following cases.

**Case 1:** If we have $g_y^*(\boldsymbol{x}) > 0$ and $g_\perp^*(\boldsymbol{x}) > 0$ for some $y \in \mathcal{Y}$. Note that we cannot have $y \neq y^{'}$ both belonging to $[K]$ such that $g_y^*(\boldsymbol{x}) > 0$ and $g_{y'}^*(\boldsymbol{x}) > 0$. Because this would imply $\eta_y(\boldsymbol{x}) > \frac{1}{2}$ and $\eta_{y'}(\boldsymbol{x}) > \frac{1}{2}$ which contradicts the rules of probabilities. Thus, theoretically, only one such $y \in \mathcal{Y}$ is possible such that $g_y^*(\boldsymbol{x}) > 0$. And if we take the prediction for our L2D problem as $\arg\max_{k \in [K+1]} g_k^*(\boldsymbol{x})$, our prediction would correspond to the Bayes Optimal decision, i.e. if

$$
\begin{aligned}
& g_y^*(\boldsymbol{x}) < g_\perp^*(\boldsymbol{x}) \quad \forall y \in \mathcal{Y} \\
\implies & \gamma^{-1}\left(g_y^*(\boldsymbol{x})\right) < \gamma^{-1}\left(g_\perp^*(\boldsymbol{x})\right) \quad \forall y \in \mathcal{Y} \\
\implies & \eta_y(\boldsymbol{x}) < p_{\mathrm{m}}(\boldsymbol{x}) \quad \forall y \in \mathcal{Y}
\end{aligned}
$$

Thus, such if $g_\perp^*(\boldsymbol{x}) > g_y^*(\boldsymbol{x})$ such that $g_\perp^*(\boldsymbol{x}) > 0, g_y^*(\boldsymbol{x}) > 0$, then the prediction following the decision rule $\arg\max_{k \in [K+1]} g_k^*(\boldsymbol{x})$ would correspond with the Bayes optimal rule

$$
r(\boldsymbol{x}) = \mathbb{I}\left[\max_{\eta_y \in \mathcal{Y}} \eta_y(\boldsymbol{x}) < p_{\mathrm{m}}(\boldsymbol{x})\right]
$$

**Case 2:** In this case, if $\nexists y \in \mathcal{Y}$ s.t. $g_y^*(\boldsymbol{x}) > 0$, but $g_\perp^*(\boldsymbol{x}) > 0$, then the same argument as above implies the decision with the Bayes optimal rule.

**Case 3:** if $\exists y \in \mathcal{Y}$ s.t. $g_y^*(\boldsymbol{x}) > 0$, but $g_\perp^*(\boldsymbol{x}) < 0$, then the same argument as above implies the decision with the Bayes optimal rule. In this case, we will have $r(\boldsymbol{x}) = 0$, and the classifier's prediction would correspond with the regular Bayes Optimal Classifier, i.e. $\arg\max_{y \in \mathcal{Y}} \eta_y(\boldsymbol{x})$.

**Case 4:** In this case, if $\nexists y \in \mathcal{Y}$ s.t. $g_y^*(\boldsymbol{x}) > 0$, and also $g_\perp^*(\boldsymbol{x}) < 0$. This situation invokes the common "None of the above" classification rule for One-vs-All classifiers.

Thus, the cases above imply that the minimizer of the point-wise *inner* $\psi$*-risk* gives the Bayes Optimal Classifier and Rejection prediction for $\mathbf{x} = \boldsymbol{x}$. Thus, the surrogate loss $\phi$ is calibrated for 0-1 L2D.

## D. Additional Results

### D.1. ECE values with respect to the classifier correctness

| Expected Calibration Error (%) on CIFAR-10 | | |
|---|---|---|
| | **OvA** | Softmax |
| Both Random | 0.51 | **0.34** |
| Random Expert | **6.47** | 7.22 |
| Random Data | **1.94** | 2.36 |
| Both Useful | **6.92** | 7.92 |

*Table 2. ECE for Classifier on CIFAR-10 Simulation.* We compare calibration across the two parameterizations: OvA (Eq. 10) and softmax (Eq. 24).

### D.2. Effect of Calibration on System's Accuracy

In this section, we verify calibration's role in the overall system's accuracy. For the trained one-vs-all model from Figure 2c, we apply a post-processing calibration technique called *temperature scaling* (Guo et al., 2017) to further calibrate the rejector. In Figure 5, we see that this additional calibration step marginally improves the system's accuracy. This result shows that calibration does positively correlate with accuracy.

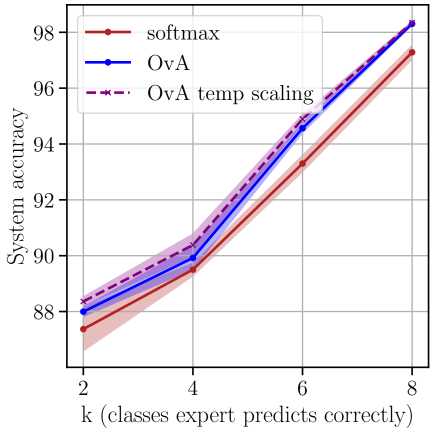


*Figure 5. Effect of post-processing calibration for One-Vs-All rejector.* We can see that post-processing calibration of $p_{\mathrm{m}}(\boldsymbol{x})$ further improves the system accuracy. This shows the effect of calibration for the overall system's accuracy for L2D.

### D.3. Class-wise performance of the simulated MLPMixer expert for HAM10000

| | Classes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| metric | bkl | df | mel | nv | vasc | akiec | bcc | weighted avg |
| precision | 0.52 | 0.33 | 0.51 | 0.82 | 0.27 | 0.44 | 0.47 | 0.71 |
| recall | 0.37 | 0.06 | 0.21 | 0.95 | 0.48 | 0.39 | 0.45 | 0.74 |
| f1-score | 0.43 | 0.10 | 0.30 | 0.88 | 0.34 | 0.41 | 0.46 | 0.71 |

*Table 3. Performance of simulate MLPMixer Expert on HAM10000.* We can see that the trained model has non-uniform performance across different classes. The resulting model is still a valid simulation of real world expert who might be expert for some classes(class nv for example).

## E. Additional Information about the Methods

In this section, we provide additional implementation details for our comparison systems. We first note that the differentiable-triage algorithm (Okati et al., 2021) considers the triage level(or *budget*) in the training of the algorithm. None of the other baselines have this aspect. Thus, to fairly compare all the other methods with the differentiable-triage algorithm, we use the same methodology employed by Okati et al. (2021) in their paper (We refer the reader to Appendix C of their paper for more details). For each of the method, we also provide the details below:

1. Softmax Surrogate (Mozannar & Sontag, 2020): for a *budget* $b$ and the samples size $\mathcal{D}$, it sorts the samples in increasing order of $\max_{k\in[K]} p_k(\boldsymbol{x}) - p_\perp(\boldsymbol{x})$, and then defers the $\min\left(\lfloor b|\mathcal{D}|\rfloor, n_c\right)$ where $n_c$ is the number of samples for which $p_\perp(\boldsymbol{x}) \geq \max_{k\in[K]} p_k(\boldsymbol{x})$.

2. One-Vs-All Surrogate: we use the same procedure as the softmax surrogate.

3. Score Baseline (Raghu et al., 2019): this method first trains a classifier model, and uses the classifier's predictive uncertainty to defer to the expert. Note that this classifier is trained in a regular way, i.e. it doesn't employ any additional procedure for deferral. During test time, it first sorts the dataset of size $|\mathcal{D}|$ in the increasing order of $\max_{k\in[K]} p_k(\boldsymbol{x})$, and defers to the expert first $\lfloor b|\mathcal{D}|\rfloor$ for the *budget* $b$. The performance of this method depends on the reliability of the uncertainty estimates the classifier provides. We, therefore, use a post-processing calibration technique called Temperature Scaling (Guo et al., 2017) to calibrate the classifier using the validation dataset split.

4. Confidence Baseline (Bansal et al., 2021): this method first estimates $p(y = m)$, the probability of the expert being correct. However, this estimate is independent of the input sample $\boldsymbol{x}$, i.e. $p(y = m|\boldsymbol{x}) = p(y = m)$. Having obtained this estimate, it trains the system sequentially where at each iteration, it uses only $\min\left(\lfloor b\mathcal{D}\rfloor, n_c\right)$ samples with the

lowest value of $p(y = m) - \max_{k \in [K]} p_k(\boldsymbol{x})$ in the corresponding mini-batch for training. Here, $n_c$ is the number of samples where $p(y = m) > \max_{k \in [K]} p_k(\boldsymbol{x})$. During test time for the *budget* $b$, it first sorts the dataset of size $|\mathcal{D}|$ in the increasing order of $\max_{k \in [K]} p_k(\boldsymbol{x})$, and defer the first $\min(\lfloor b|\mathcal{D}| \rfloor, n_c)$ samples to the expert, where $n_c$ denotes the same quantity as before except this time for the test set samples.

5. Differentiable Triage (Okati et al., 2021): this is a sequential learning algorithm that first estimates the predictive model for a given *budget* $b$, and then having learned the model, it approximates the optimal triage policy for the learned model and $b$. The optimal triage policy is to compare the model's prediction loss and the expert's prediction loss, and defer to the expert if the latter is smaller than the former. Therefore, the training algorithm assumes access to the expert's predictive loss as opposed to just the expert's predictions for the surrogate loss methods. Following the original authors, we use the Negative Log-Likelihood loss as the expert's loss. At test time, it use the learned approximation of the optimal triage policy to defer to the expert.

## F. Additional Experimental Details

Below we provide more details on our experimental set-up.

**CIFAR-10** For the experiments on CIFAR-10, we use 28-layer Wide Residual Networks (Zagoruyko & Komodakis, 2016) without using any data augmentation techniques following Mozannar & Sontag (2020). We use SGD with a momentum of $0.9$, weight decay $5e-4$, and initial learning rate of $0.1$. We further use cosine annealing learning rate schedule. We monitor validation loss, and employ early stopping to terminate the training if the loss doesn't improve for 20 epochs. The datasets are standardized to have 0 mean and unit variance. We train the models with a batch size of 1024. These experimental settings apply to both the Softmax Surrogate and the One-vs-All surrogate loss.

**HAM10000** To simulate the expert, we train an 8-layer MLPMixer model (Tolstikhin et al., 2021). We make use of the publicly available code [2] for MLPMixer model. We resize the HAM10000 images to $224 \times 224$ for our experiments. The 8-layer model has patch size of 16, expansion factor 2, and the dimensionality of the features to be 128. We train this model with Adam optimization algorithm with a learning rate of $0.001$, weight decay of $5e-4$. We further use cosine annealing learning rate schedule with a warm-up period of $5$ epochs. The model is trained with a batch size of $1024$, again with early stopping with a patience of 20 epochs. Since our goal was to simulate the real-world expert, we did not do extensive hyperparameter search for the expert model. For our main model on HAM10000, we finetune ResNet34 model. The training settings are same for the surrogate loss methods for CIFAR-10 experiments.

For our other baselines, we use the code made available by the respective authors.

[2] https://github.com/jaketae/mlp-mixer/