Exploiting Human-AI Dependence for Learning to Defer

Zixi Wei [1] **Yuzhou Cao** [2] **Lei Feng** [3]

Abstract

The *learning to defer* (L2D) framework allows models to defer their decisions to human experts. For L2D, the Bayes optimality is the basic requirement of theoretical guarantees for the design of *consistent* surrogate loss functions, which requires the minimizer (i.e., learned classifier) by the surrogate loss to be the Bayes optimality. However, we find that the original form of Bayes optimality fails to consider the dependence between the model and the expert, and such a dependence could be further exploited to design a better consistent loss for L2D. In this paper, we provide a new formulation for the Bayes optimality called *dependent Bayes optimality*, which reveals the dependence pattern in determining whether to defer. Based on the dependent Bayes optimality, we further present a deferral principle for L2D. Following the guidance of the deferral principle, we propose a novel consistent surrogate loss. Comprehensive experimental results on both synthetic and real-world datasets demonstrate the superiority of our proposed method.

1. Introduction

With the increasing deployment of machine learning models in risk-critical domains such as medical diagnosis (Zoabi et al., 2021), criminal justice (Chalkidis et al., 2019), and autonomous driving (Grigorescu et al., 2020), the reliability and safety issues of these models are getting crucially important. These risk-critical domains heighten the urgency to prevent critical mispredictions of models. To address this challenge, one approach is to incorporate a mechanism that allows models to defer to human experts when confronted with challenging or high-stakes decisions. *Learning to Defer* (L2D) (Madras et al., 2018; Charusaie et al., 2022; Mozannar et al., 2022; 2023; Straitouri et al., 2022; De et al., 2021; Narasimhan et al., 2022; Mao et al., 2024; Hemmer et al., 2023; Joshi et al., 2021; Liu et al., 2024; Alves et al., 2024; Lykouris & Weng, 2024) aims to avoid critical mispredictions of models by facilitating the collaboration between human experts and machine learning models. Specifically, L2D aims to defer the prediction to an expert when the expert is more likely to be correct than the model. For medical diagnosis, a CT image may involve complex anatomical structures or potential abnormalities. Such complexity could lead to a lack of confidence in the diagnostic prediction of a model. Then the model could defer the prediction and transfer the CT images to a radiologist. When faced with straightforward cases, the model can make diagnostic predictions by itself without a radiologist.

L2D can be formulated as a risk-minimization problem, where the 0-1-deferral risk (Mozannar & Sontag, 2020) needs to be minimized. The 0-1-deferral loss involves a cost of 1 when the model makes an incorrect decision without an expert or defers to a wrong decision made by an expert and involves a cost of 0 otherwise. Due to the discontinuous and non-convex properties of the 0-1-deferral loss, the risk-minimization problem is NP-hard even in simple settings (Mozannar et al., 2023). To make the optimization problem solvable, one commonly used strategy is to design a continuous surrogate loss that holds statistical consistency *w.r.t.* the 0-1-deferral loss. Concretely, we say a surrogate loss holds consistency if and only if the minimizer of the surrogate risk is the minimizer of the 0-1-deferral risk. This implies that a model trained with the surrogate loss is expected to converge to the optimal model for the 0-1-deferral loss.

Mozannar & Sontag (2020) showed the Bayes optimality for the 0-1-deferral risk (we provide a detailed description in Section 2), which states that the model should defer to an expert if the expert has a larger confidence in making the right decision than the optimal model, otherwise we should accept the decision made by the model. The Bayes optimality plays a crucial role in the design of surrogate losses, influencing the formulation of many consistent surrogate loss functions, where the consistency means that the optimal model for the surrogate risk meets the Bayes optimality. Thus a well-formulated Bayes optimality can guide the design of a superior consistent surrogate loss.

[1] College of Computer Science, Chongqing University, China [2] School of Computer Science and Engineering, Nanyang Technological University, Singapore [3] Information Systems Technology Design Pillar, Singapore University of Technology and Design, Singapore. Correspondence to: Lei Feng <feng_lei@sutd.edu.sg>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Guided by the Bayes optimality (Mozannar & Sontag, 2020), many consistent surrogate losses were properly designed to meet the requirement of the Bayes optimality. Mozannar & Sontag (2020) proposed a statistically consistent surrogate loss based on the softmax parameterization. However, the method proposed by Mozannar & Sontag (2020) would incur an unbounded confidence estimator for the expert. Verma & Nalisnick (2022) states that this problem is caused by the intrinsic property of the softmax parameterization. To alleviate this problem, the authors designed two estimators based on the One-versus-All (OvA) strategy (Zhang, 2004b) and induced a surrogate loss that holds statistical consistency. In defense of utilizing the softmax parametrization for a bounded confidence estimator, Cao et al. (2023) demonstrated that the unbounded estimator arises from the symmetric nature of surrogate losses and proposed an asymmetric softmax parameterization to obtain a bounded confidence estimator for the expert.

To summarize, previous methods determine whether to defer by first using training data to estimate confidence and then deciding whether to defer based on the estimated confidence. However, estimating the confidence of the model and the expert needs massive training data, which would cause significant difficulties in accurately estimating confidence, especially when deep neural networks are used (Guo et al., 2017; Wang et al., 2021; Wei et al., 2022). Such a complexity raises an important question: Is it possible to directly use the training data to make deferral decisions, without the confidence estimation step?

To answer this question, we propose to use the human-model dependence pattern observed in the training data to directly make the deferral decision, thereby bypassing the need for confidence estimation. However, the original form of Bayes optimality shown in Proposition 2.1 fails to consider this human-model dependence pattern. Therefore, we introduce a new formulation of the Bayes optimality, called *dependent Bayes optimality*. Unlike the original Bayes optimality that treats the confidence of the model and the expert individually, the dependent Bayes optimality can reflect the impact of the human-model dependence pattern in the deferral decision. This dependence pattern helps us to design a deferral principle for the L2D problem, which enables us to determine whether to defer only based on the dependence pattern in training data. Concretely, the deferral principle suggests the following strategy to treat each instance:

1. Accept the prediction made by the model when the expert makes a wrong prediction.

2. Defer the prediction to the expert when the expert makes the right prediction and the model makes a wrong prediction.

3. Do not determine whether to defer when both the expert and the model make the right prediction.

Following the deferral principle, we further propose a *dependent cross-entropy* (DCE) loss, which is a consistent surrogate loss for L2D.

The contributions of this paper can be summarized below:

- We provide a new Bayes optimality formulation for L2D (i.e., dependent Bayes optimality). This new formulation underscores the influence of the human-model dependence in the deferral process for the L2D problem. (Section 3.2)
- We present a deferral principle that enables us to decide whether to defer each instance solely based on the human-model dependence pattern observed in training data. (Section 3.3)
- We propose a novel loss called *dependent cross-entropy* (DCE) loss for L2D based on the dependent Bayes optimality. We show that the DCE loss is a consistent surrogate loss for L2D and it can induce a bounded confidence estimator for the expert. (Section 4)
- Experimental results on both synthetic experts and real-world experts demonstrate the superiority of our proposed method. (Section 5)

2. Preliminaries

In this section, we review the problem setting of *Learning to Defer* (L2D) and provide a succinct overview of previous studies in the field.

2.1. Poblem Setting

The goal of the L2D problem is to train an augmented classifier with a deferral option in the K-class classification scenario. In this paper, we define $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = [K]$ as the feature space and label space respectively, where $[K] = \{1, 2, \ldots, K\}$. Let $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ denote the feature vector and label respectively. Let us denote $X \times Y \times M \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ as the triplet of random variables of feature, label, and expert prediction. We use $\boldsymbol{x} \times y \times m$ to represent the realization of $X \times Y \times M \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$, which obeys an underlying joint distribution $p(\boldsymbol{x}, y, m)$. We have access to the dataset $\{(\boldsymbol{x}_i, y_i, m_i)\}_{i=1}^n$ that is collected independently and identically from the joint distribution $p(\boldsymbol{x}, y, m)$. We use $f : \mathcal{X} \to \mathcal{Y}^\perp$ to denote the classifier with a deferral option, where $\perp$ is used to denote the deferral option and $\mathcal{Y}^\perp = [K] \cup \{\perp\}$. Concretely, when $f(\boldsymbol{x}) \neq\perp$, we accept the prediction produced by the classifier, when $f(\boldsymbol{x}) =\perp$ the classifier defers the prediction to the expert, and the expert prediction is used as the prediction result.

The performance of L2D can be formulated as the 0-1-

deferral loss defined below:

$$L_{01}^{\perp}(f(\boldsymbol{x}), y, m) = \mathbb{I}_{f(\boldsymbol{x})\neq y}\mathbb{I}_{f(\boldsymbol{x})\neq\perp} + \mathbb{I}_{m\neq y}\mathbb{I}_{f(\boldsymbol{x})=\perp},$$

where $\mathbb{I}$ takes the value 1 if the statement in the subscript is true otherwise it takes the value 0. As observed, the loss function assigns a value of 1 if we accept a wrong prediction produced by the classifier or the expert provides a wrong prediction when the prediction is deferred to the expert. Otherwise, the loss function assigns a value of 0. In order to learn an effective classifier for L2D, we aim to minimize the following target risk *w.r.t.* to $L_{01}^{\perp}$:

$$R_{01}^{\perp}(f) = \mathbb{E}_{p(\boldsymbol{x},y,m)}[L_{01}^{\perp}((f(\boldsymbol{x}), y, m))],$$

We denote by f^* the minimizer (Bayes optimality) of the target risk $R_{01}^{\perp}(f)$, i.e., $f^* = \arg\min_f R_{01}^{\perp}(f)$. By further introducing $\eta_y(\boldsymbol{x}) = \mathbb{P}(Y = y|\boldsymbol{x})$ as the posterior probability, an important property of f^* was shown (Mozannar & Sontag, 2020) as follows:

Proposition 2.1 (Bayes optimality of L2D)**.** *The minimizer of $R_{01}^{\perp}(f)$ can be expressed as:*

$$f^*(\boldsymbol{x}) = \begin{cases} \perp, & \mathbb{P}(Y = M|\boldsymbol{x}) > \max\limits_{y\in\mathcal{Y}} \eta_y(\boldsymbol{x}), \\ \arg\max\limits_{y\in\mathcal{Y}} \eta_y(\boldsymbol{x}), & \text{otherwise.} \end{cases}$$

The Bayes optimality indicates that the classifier should defer the prediction to the expert if the expert has a higher confidence to predict correctly; otherwise, we take the label with the largest posterior probability as the final prediction.

Proposition 2.1 can be considered as a generalized version of Chow's rule (Chow, 1970) in classification with rejection (CwR) (Bartlett & Wegkamp, 2008; Yuan & Wegkamp, 2010a; Ramaswamy et al., 2018; Ni et al., 2019a), where the rejection cost $c(\boldsymbol{x})$ is replaced by a probability function $1 - \mathbb{P}(Y = M|\boldsymbol{x})$.

2.2. Consistent Surrogate Losses for L2D

Although L2D can be formulated as a risk-minimization problem *w.r.t.* the 0-1-deferral risk. However, due to the discontinuous and non-convex of properties of the 0-1-deferral loss $L_{01}^{\perp}$, the minimization of $R_{01}^{\perp}$ is an NP-hard problem (Mozannar et al., 2023). A common strategy is to design continuous surrogate losses that can induce a classifier to meet the Bayes optimality. This strategy has been widely employed in many areas including ordinary multi-class classification (Zhang, 2004a; Bartlett et al., 2006; Finocchiaro et al., 2019), multi-label classification (Gao & Zhou, 2013; Koyejo et al., 2015; Zhang et al., 2020), cost-sensitive learning (Scott, 2011; 2012), and learning to reject (Cortes et al., 2016; Yuan & Wegkamp, 2010b; Ni et al., 2019b; Charoenphakdee et al., 2020). Concretely, we say a surrogate loss function is consistent if the minimizer of the surrogate loss meets the Bayes optimality almost surely. Due to this fact, the formulation of the Bayes optimality is crucial for deriving consistent surrogate losses for the L2D problem. A well-formulated Bayes optimality could convey more useful information that can be exploited for the design of consistent surrogate losses.

In this paper, we define $\boldsymbol{s} = g(\boldsymbol{x})$ the score vector outputted by the scoring function $g : \mathcal{X} \to \mathbb{R}^{k+1}$, and we define $s_{\perp} = s_{k+1}$ the score value for deferral. The scoring function g can induce the decision function $f : \mathcal{X} \to \mathcal{Y}^{\perp}$ (i.e., the classifier with a deferral option) with the following transformation $\varphi : \mathbb{R}^{k+1} \to \mathcal{Y}^{\perp}$:

$$\varphi(g(\boldsymbol{x})) = \begin{cases} \perp, & g_{\perp}(\boldsymbol{x}) > \max\limits_{y\in\mathcal{Y}} g_y(\boldsymbol{x}) \\ \arg\max\limits_{y\in\mathcal{Y}} g_y(\boldsymbol{x}), & \text{otherwise.} \end{cases}$$

In this paper, we consider a continuous surrogate loss function $L^{\perp} : \mathbb{R} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{+}$, which takes a score vector, label, expert prediction as input and outputs a positive value.

The early consistent surrogate loss for L2D is proposed by Mozannar & Sontag (2020). The authors generalized the cross entropy loss to L2D loss based on cost-sensitive learning:

$$\begin{aligned} L_{\mathrm{CE}}(g(\boldsymbol{x}), y, m) = &-\log \psi_{\mathcal{Y}^{\perp}}^{y}(g(\boldsymbol{x})) \\ &- \mathbb{I}_{y=m} \log \psi_{\mathcal{Y}^{\perp}}^{\perp}(g(\boldsymbol{x})), \end{aligned}$$

where we define $\psi_{\mathbb{S}}^{i}(g(\boldsymbol{x})) = \frac{\exp(g_i(\boldsymbol{x}))}{\sum_{j\in\mathbb{S}} \exp(g_j(\boldsymbol{x}))}$ as the softmax transformation and $\mathbb{S} \subseteq \mathcal{Y}^{\perp}$ denotes a class set. In the context of L_{CE}, $\mathbb{S} = \mathcal{Y}^{\perp}$. However, L_{CE} would lead to an unbounded confidence estimator $\widehat{\mathbb{P}}_{\mathrm{CE}}(Y = M|\boldsymbol{x}) \in [0, +\infty]$ formulated as:

$$\widehat{\mathbb{P}}_{\mathrm{CE}}(Y = M|\boldsymbol{x}) = \psi_{\mathcal{Y}^{\perp}}^{\perp}(\boldsymbol{x})/(1 - \psi_{\mathcal{Y}^{\perp}}^{\perp}(\boldsymbol{x})).$$

When $\psi_{\mathcal{Y}^{\perp}}^{\perp}(\boldsymbol{x}) > \frac{1}{2}$, the estimator $\widehat{\mathbb{P}}_{\mathrm{CE}}(Y = M|\boldsymbol{x})$ would output an invalid estimated expert accuracy larger than 1.

Verma & Nalisnick (2022) stated that this unbounded confidence estimator is caused by the softmax parameterization. The authors addressed this problem by the one-versus-all (OvA) strategy (Zhang, 2004a) and proposed an OvA-based surrogate loss. This OvA-based loss induces a bounded confidence estimator for $\mathbb{P}(Y = M|\boldsymbol{x})$ and still holds consistency. The OvA-based surrogate loss can be formulated as:

$$\begin{aligned} &L_{\mathrm{OvA}}(g(\boldsymbol{x}), y, m) = \xi(g_y(\boldsymbol{x})) \\ &+ \sum_{y'\in\mathcal{Y}^{\perp}, y'\neq y} \xi(-g_{y'}(\boldsymbol{x})) + \mathbb{I}_{y=m}(\xi(g_{\perp}(\boldsymbol{x})) - \xi(-g_{\perp}(\boldsymbol{x}))), \end{aligned}$$

where ξ denotes a binary proper composite loss (Reid & Williamson, 2009). L_{OvA} can induce a bounded confidence estimator $\widehat{\mathbb{P}}_{\mathrm{OvA}}(Y = M|\boldsymbol{x}) \in [0, 1]$ formulated as:

$$\widehat{\mathbb{P}}_{\mathrm{OvA}}(Y = M|\boldsymbol{x}) = \psi_{\xi}(g_{\perp}(\boldsymbol{x})),$$

where $\psi_{\xi} : \mathbb{R} \rightarrow [0, 1]$ is a mapping function induced from the binary loss ξ.

In defense of the softmax parametrization for a bounded confidence estimator, Cao et al. (2023) showed that the unbounded confidence estimator is caused by the symmetric nature of the surrogate losses. The authors proposed an asymmetric softmax-based surrogate loss, which could be formulated as:

$$\begin{aligned} L_{\mathrm{A-SM}}(g(\boldsymbol{x}), y, m) = &- \log \psi_{\mathcal{Y}}^{y}(g(\boldsymbol{x})) \\ &- \mathbb{I}_{y=m} \log \psi_{\mathcal{Y}^{\perp}/q}^{\perp}(g(\boldsymbol{x})) - \mathbb{I}_{m \neq y} \log(1 - \psi_{\mathcal{Y}/q}^{\perp} g(\boldsymbol{x})), \end{aligned}$$

where $q = \arg\max_{i \in \mathcal{Y}} g_i(\boldsymbol{x})$ and $\mathcal{Y}^{\perp}/q$ is used to denote the class set excluding q from $\mathcal{Y}^{\perp}$, and we use q to represent the model's prediction in the rest of this paper. A bounded confidence estimator could be induced from $L_{\mathrm{A-SM}}$, which could be formulated as:

$$\widehat{\mathbb{P}}_{\mathrm{A-SM}}(Y = M|\boldsymbol{x}) = \psi_{\mathcal{Y}^{\perp}/q}^{\perp}.$$

3. A New Formulation of the Bayes Optimality

In this section, we first discuss the limitation of the existing Bayes optimality, and then introduce a new formulation called *dependent Bayes optimality*.

3.1. Limitation of The Existing Bayes Optimality

In this paper, we find an important limitation of the existing Bayes optimality. As demonstrated by Proposition 2.1, the Bayes optimality fails to consider the dependence between M and Y. Thus, most previous methods estimate the confidence $\mathbb{P}(Y = r|\boldsymbol{x})$ and $\mathbb{P}(Y = M|\boldsymbol{x})$ independently, treating them as uncorrelated. In practice, these methods empirically estimate $\mathbb{P}(Y = M|\boldsymbol{x})$ and $\mathbb{P}(Y = r|\boldsymbol{x})$ for each instance, and then determine whether to defer for each instance based on the estimated values.

However, using neural networks to estimate confidence could cause various issues (i.e., overconfidence (Wei et al., 2022)). This motivates us to skip the step of estimating confidence when deciding whether to defer each instance. In our paper, we give a positive answer to this question by introducing a new formulation of the Bayes Optimality and providing a new deferral principle for L2D.

3.2. Dependent Bayes Optimality

To overcome the limitation of the original Bayes optimality in Proposition 2.1, we provide a new formulation called dependent Bayes optimality as follows:

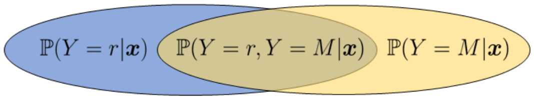
$\mathbb{P}(Y = r|\boldsymbol{x})$
$\mathbb{P}(Y = r, Y = M|\boldsymbol{x})$
$\mathbb{P}(Y = M|\boldsymbol{x})$

Figure 1. Venn diagram of $\mathbb{P}(Y = r|\boldsymbol{x})$ and $\mathbb{P}(Y = M|\boldsymbol{x})$.

Proposition 3.1 (Dependent Bayes optimality of L2D)**.** *The minimizer of $R_{01}^{\perp}(f)$ can be expressed as:*

$$f^*(\boldsymbol{x}) = \begin{cases} \perp, \mathbb{P}(Y \neq r, Y = M|\boldsymbol{x}) > \mathbb{P}(Y = r, M \neq Y|\boldsymbol{x}), \\ \arg\max_{y \in \mathcal{Y}} \eta_y(\boldsymbol{x}), \qquad \text{otherwise}, \end{cases}$$

where $r = \arg\max_{y \in \mathcal{Y}} \eta_y(\boldsymbol{x})$.

Proposition 3.1 can be derived from Proposition 2.1 by subtracting $\mathbb{P}(Y = M, Y = r|\boldsymbol{x})$ from $\mathbb{P}(Y = M|\boldsymbol{x})$ and $\mathbb{P}(Y = r|\boldsymbol{x})$[1]. As illustrated in Figure 1, by subtracting the intersection $(Y = M, Y = r)$ from $(Y = M)$ and $(Y = r)$, we obtain two contrary events $(Y = M, Y \neq r)$ and $(Y \neq M, Y = r)$ in Proposition 3.1. This subtraction is intuitive since deferring the prediction when $(Y = M = r)$ is meaningless since this deferring would not change the final prediction of an L2D system.

Mathematically, the optimality conditions presented in Propositions 2.1 and Propositions 3.1 are completely equivalent, i.e., if a classifier satisfies the conditions in one proposition, then it also satisfies the conditions in the other one. However, compared with Proposition 2.1, Proposition 3.1 characterizes the dependence pattern between M and Y based on a joint distribution in the deferring decision. The classifier is suggested to defer the prediction to the expert when the classifier is likely to make a wrong prediction and the expert is likely to predict correctly. This dependence pattern helps us to design the following deferral principle for the L2D problem.

3.3. The Deferral Principle for L2D Problem

Let $q = \arg\max_{y \in \mathcal{Y}} g_y(\boldsymbol{x})$ be the predicted class with the largest score value and $r = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(Y = y|\boldsymbol{x})$ be the class with the largest posterior probability.

Given an instance-label-expert triplet $(\boldsymbol{x}, y, m)$ sampled from $p(\boldsymbol{x}, y, m)$ and the prediction q made by the model. By considering the human-model dependence pattern, we can obtain 4 possible combinations for y and m: 1) $(y \neq q, y = m)$, 2) $(y = q, y = m)$, 3) $(y = q, y \neq m)$, 4) $(y \neq q, y \neq m)$.

For Case 1) $(y \neq q, y = m)$, the model produces a wrong prediction and the expert makes the right prediction. We

[1] $\mathbb{P}(Y = M|\boldsymbol{x}) - \mathbb{P}(Y = M, Y = r|\boldsymbol{x}) = \mathbb{P}(Y = M, Y \neq r|\boldsymbol{x})$ and $\mathbb{P}(Y = r|\boldsymbol{x}) - \mathbb{P}(Y = M, Y = r|\boldsymbol{x}) = \mathbb{P}(Y \neq M, Y = r|\boldsymbol{x})$.

can consider this case as the occurrence of the event $(Y \neq r, Y = M)$. In this case, we prefer to trust $\mathbb{P}(Y \neq r, Y = M|\boldsymbol{x}) > \mathbb{P}(Y = r, M \neq Y|\boldsymbol{x})$, and thus we prefer to defer the instance $\boldsymbol{x}$ to the expert in this case.

For Case 2) $(y = q, y = m)$, the model and the expert both produce the right prediction. We can consider this case as the occurrence of the event $(Y = r, Y = M)$. It cannot help us to determine whether $\mathbb{P}(Y \neq r, Y = M|\boldsymbol{x}) > \mathbb{P}(Y = r, M \neq Y|\boldsymbol{x})$ or not, and thus we do not need to decide whether to defer in this case.

For Cases 3) and 4), $y \neq m$ means that the expert makes a wrong prediction. In the two cases, we should accept the model prediction, since there is no need to defer to an expert with a wrong prediction.

By considering all the four cases above, our proposed dependent Bayes optimality (Proposition 3.1) suggests the following deferral principle for different combinations among y, m, and q in the training set:

1. Accept the model prediction made when $y \neq m$.

2. Defer the prediction to the expert when $y \neq q, y = m$.

3. Do not determine whether to defer when $y = q, y = m$.

This deferral principle enables us to determine whether to defer each instance only based on the human-model dependence pattern observed in the triplet (y, m, q). In the next section, we propose a novel consistent surrogate loss based on this deferral principle.

4. Proposed Dependent Surrogate Loss

In this section, we propose a consistent surrogate loss based on the deferral principle we presented in Section 3. We also demonstrate its statistical consistency. In addition, we also show that the proposed surrogate loss could induce a bounded confidence estimator for the expert.

4.1. Formulation of The Proposed Loss

Motivated by the dependent Bayes optimality in Proposition 3.1, we propose a novel **D**ependent **C**ross-**E**ntropy (DCE) Loss, $L^{\perp}_{\mathrm{DCE}}(\boldsymbol{g}(\boldsymbol{x}), y, m) : \mathbb{R}^{k+1} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{+}$, defined as follows:

$$
\begin{aligned}
L^{\perp}_{\mathrm{DCE}}(g(\boldsymbol{x}), y, m) = &-\mathbb{I}_{y \neq m} \log(\psi^{y}_{\mathcal{Y}^{\perp}}(g(\boldsymbol{x}))) \\
&- \mathbb{I}_{y=m}\big(\log(\psi^{y}_{\mathcal{Y}}(g(\boldsymbol{x}))) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/q}(g(\boldsymbol{x})))\big),
\end{aligned}
$$

where $q = \arg\max_{y \in \mathcal{Y}} g_y(\boldsymbol{x})$ and $\mathcal{Y}^{\perp}/q$ is used to represent the class set obtained by removing q from $\mathcal{Y}^{\perp}$.

Here, we explain how the DCE loss addresses the L2D problem. As is commonly understood, An ideal surrogate loss should complete two tasks: (1) Ensuring that the model makes the right prediction (i.e., $q = r$). (2) Properly establishing the magnitude ordering between $g_q(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$: a) making $g_q(\boldsymbol{x}) > g_{\perp}(\boldsymbol{x})$ when we accept the prediction made by the model; b) making $g_q(\boldsymbol{x}) < g_{\perp}(\boldsymbol{x})$ when we defer the prediction to the expert; c) not influencing the magnitude ordering between $g_q(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$ when we do not know whether to defer or not. For task (1), our proposed DCE loss ensures $q = r$ by making $g_r(\boldsymbol{x}) > g_i(\boldsymbol{x})$ for all $i \neq r, i \in \mathcal{Y}$. For task (2), our DCE loss establishes the magnitude ordering between $g_q(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$ based on the deferral principle. Now we describe how the proposed DCE loss accomplishes task (2) for different combinations of (y, m, q).

For Case 1) $(y \neq q, y = m)$, we should defer the prediction. Our proposed DCE loss would make $g_y(\boldsymbol{x}) > g_q(\boldsymbol{x})$ (the first term for $y = m$ in our DCE loss) and make $g_{\perp}(\boldsymbol{x}) > g_y(\boldsymbol{x})$ (the last term for $y = m$ in our DCE loss), which further makes $g_{\perp}(\boldsymbol{x}) > g_q(\boldsymbol{x})$, and thus the DCE loss accomplishes the task to defer the instance $\boldsymbol{x}$. It is noteworthy that the value of q may be changed during the training process. If the predicted label q becomes the true label y in the training process, then Case 1) would become Case 2), which we describe below.

For Case 2) $(y = q, y = m)$, we do not determine whether to defer. The DCE loss treats $g_y(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$ identically during training. Since the surrogate loss needs to make $g_y(\boldsymbol{x}) > g_i(\boldsymbol{x}), \forall i \neq y, i \in \mathcal{Y}$ to ensure the model makes the right prediction (the first term for $y = m$ in our DCE loss), our proposed loss also makes $g_{\perp} > g_i(x), \forall i \neq y, i \in \mathcal{Y}$ (the last term for $y = m$ in our DCE loss). Thus the DCE loss refrains from influencing the relative ordering between $g_q(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$.

For Cases 3) and 4) $(y = q, m \neq y)$ or $(y \neq q, m \neq y)$, we should accept the prediction made by the model. The DCE loss makes $g_y(\boldsymbol{x}) \geq g_i(\boldsymbol{x}), \forall i \in \mathcal{Y}^{\perp}$ (the term for $y \neq m$ in our DCE loss). Thus the DCE loss ensures $y = q$ and $g_q(\boldsymbol{x}) > g_{\perp}(\boldsymbol{x})$, thereby accomplishing the task of accepting the model prediction.

Unlike previous methods that independently estimate the confidences $\mathbb{P}(Y = r|\boldsymbol{x})$ and $\mathbb{P}(Y = M|\boldsymbol{x})$, the DCE loss determines whether to defer each instance by directly establishing the relative ordering between $g_r(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$, and thus the DCE loss could bypass the confidence estimation step. In line with the Ockham's Razor principle, this simpler method often leads to a better solution.

4.2. Consistency of The Proposed Loss

Here, we show that the proposed surrogate loss $L^{\perp}_{\mathrm{DCE}}$ is consistent. Let $R^{\perp}_{\mathrm{DCE}}(g) = \mathbb{E}_{p(\boldsymbol{x},y,m)} L^{\perp}_{\mathrm{DCE}}(g(\boldsymbol{x}), y, m)$ be the surrogate risk by taking the expectation over the joint distribution $p(\boldsymbol{x}, y, m)$, by using our proposed surrogate loss $L^{\perp}_{\mathrm{DCE}}$.

Theorem 4.1 (Consistency of the proposed surrogate loss $L^{\perp}_{\mathrm{DCE}}$)**.** *Let us denote by $g^* \in \arg\min_{g(\boldsymbol{x})} R^{\perp}_{\mathrm{DCE}}(g)$ the optimal scoring function by using our proposed surrogate loss $L^{\perp}_{\mathrm{DCE}}$. Then the classifier $\varphi(g^*(\boldsymbol{x}))$ is the (dependent) Bayes optimality in Proposition 3.1, which means that $L^{\perp}_{\mathrm{DCE}}$ is a consistent loss.*

The proof of Theorem 4.1 is provided in Appendix A. According to Theorem 4.1, our proposed DCE loss is demonstrated to be a consistent surrogate loss with respect to the 0-1-deferral loss. However, we cannot construct our consistency proof in the same manner as previous methods (Mozannar & Sontag, 2020; Verma & Nalisnick, 2022; Cao et al., 2023). The previous methods complete the proof based on two confidence estimators (i.e. one for $\mathbb{P}(Y = r|\boldsymbol{x})$ and the other for $\mathbb{P}(Y = M|\boldsymbol{x})$). Then consistency can be obtained directly by the estimated confidence estimators, which makes these methods fail to utilize the humane-model dependence pattern observed in the triplet (y, m, q).

Here we provide a nutshell of our proof. We prove the consistency by contradiction. We show that if $g^*_{\perp}(\boldsymbol{x}) > g^*_r(\boldsymbol{x})$ when $\mathbb{P}(Y = r, Y \neq M|\boldsymbol{x}) > \mathbb{P}(Y \neq r, Y = M|\boldsymbol{x})$. There must exist another scoring function g' with lower $R^{\perp}_{\mathrm{DCE}}(g')$ compared with $R^{\perp}_{\mathrm{DCE}}(g^*)$, where $g'_{\perp}(\boldsymbol{x}) < g'_r(\boldsymbol{x})$. And the proof $g^*_{\perp}(\boldsymbol{x}) < g^*_r(\boldsymbol{x})$ when $\mathbb{P}(Y = r, Y \neq M|\boldsymbol{x}) < \mathbb{P}(Y \neq r, Y = M|\boldsymbol{x})$ can be accomplished in a similarly manner. Therefore, the DCE loss can directly establish the relative ordering between $g_r(\boldsymbol{x})$ and $g_{\perp}(\boldsymbol{x})$ during the training process based on the comparison between $\mathbb{P}(Y = r, Y \neq M|\boldsymbol{x})$ and $\mathbb{P}(Y \neq r, Y = M|\boldsymbol{x})$, which utilize the human-model dependence pattern between $Y = r$ and $Y = M$.

4.3. Confidence Estimation via Our Proposed Loss

In some safety-critical scenarios such as Medical Diagnosis Systems, the accuracy of the L2D system is not the only concern. We are also interested in the uncertainties of the classifier and the expert since the classifier may mislead the expert to provide incorrect predictions even when the expert can make correct predictions initially (Madras et al., 2018; Tschandl et al., 2020). To prevent such a misleading issue, we hope that an L2D system could be a good forecaster, which means that we expect the L2D system to have the ability to recover the confidence about the degree that the classifier and the expert are right. This confidence information can subsequently offer the expert more insight into whether to trust the model predictions or not in critical decision-making scenarios. Therefore, a question naturally arises here: can we still induce a confidence estimator from our proposed DCE loss? We give an affirmative answer to this question. We show that we can induce confidence estimators for the model and the expert based on our proposed DCE loss by the following proposition:

Proposition 4.2. *Let $\lambda(\boldsymbol{x}) = \sum_{i \in \mathcal{Y}} \exp(g_i(\boldsymbol{x}))$, $q = \arg\max_{i \in \mathcal{Y}} g_i(\boldsymbol{x})$ and $\mu(\boldsymbol{x}) = \lambda(\boldsymbol{x}) - \exp(g_q(\boldsymbol{x}))$. Let us define $\rho(\boldsymbol{x}) = \frac{\mu(\boldsymbol{x})(\lambda(\boldsymbol{x}) + \exp(g_{\perp}(\boldsymbol{x})))}{\exp(g_{\perp}(\boldsymbol{x}))(\mu(\boldsymbol{x}) + \exp(g_{\perp}(\boldsymbol{x})))}$. Then confidence estimators induced from L_{DCE} can be written as:*

$$\widehat{\mathbb{P}}(Y = y|\boldsymbol{x}) = \exp(g_y(\boldsymbol{x}))/\lambda(\boldsymbol{x})$$
$$\widehat{\mathbb{P}}(Y = M|\boldsymbol{x}) = 1/(1 + \rho(\boldsymbol{x})),$$

Where we use $\widehat{\mathbb{P}}$ to denote the confidence function estimated by the scoring function.

The proof of Proposition 4.2 is provided in Appendix B. We could observe that the estimator for $\widehat{\mathbb{P}}(Y = y|\boldsymbol{x})$ takes a standard softmax formulation. The expert's estimator $\widehat{\mathbb{P}}(Y = M|\boldsymbol{x})$ shares a similar formulation with the sigmoid which is a commonly employed estimator in binary classification scenarios. $g_{\perp}(\boldsymbol{x})$ shows a positive correlation with $\widehat{\mathbb{P}}(Y = M|\boldsymbol{x})$, when $g_{\perp}(\boldsymbol{x})$ is sufficiently large, $\widehat{\mathbb{P}}(Y = M|\boldsymbol{x})$ approaches 1. Conversely, when $g_{\perp}(\boldsymbol{x})$ is sufficiently small, $\widehat{\mathbb{P}}(Y = M|\boldsymbol{x})$ approaches 0. Since $\rho(\boldsymbol{x})$ is larger than 0, we could obtain that $1/(1 + \rho(\boldsymbol{x})) \in (0, 1)$ directly, which demonstrates that the DCE loss could induce a bounded confidence estimator.

5. Experiments

In this section, we perform comprehensive experiments on both synthetic experts and real-world experts to empirically demonstrate the effectiveness of the proposed DCE loss and verify the importance of dependence in the L2D problem. In addition to validating the accuracy of L2D systems trained by each method, we also verify the performance of each method under deferral budget requirements. Furthermore, we report the Expected Calibration Error (ECE) and coverage of the L2D system trained by each method, providing a comprehensive evaluation of the L2D system.

Datasets and Models. We conduct experiments for the proposed loss and baselines on widely used benchmark datasets with both synthetic and real-world experts. For synthetic experts, we perform experiments using CIFAR-100 datasets (Krizhevsky, 2009) with the standard train-test split. Similar to the previous studies, the synthetic expert has a probability of p to predict the label correctly in the first $k \in \{20, 40, 60\}$ classes, and the synthetic expert would predict the label randomly otherwise. To simulate experts with high accuracy and medium accuracy, we conduct experiments on $p = 94\%$ following Mozannar & Sontag (2020) for high accuracy and $p = 75\%$ following Verma & Nalisnick (2022) for medium accuracy. For experiments involving real-world experts, we leverage the CIFAR-10N and CIFAR-100N datasets introduced by (Wei et al., 2021), where the expert prediction is obtained from human annotations on Amazon

Table 1. Test performance of each method on CIFAR-100 for 5 trials with $p = 94\%$. The mean(%)(standard error(%)) of related metrics are reported. The best method for the misclassification error and budgeted errors are highlighted in boldface.

Method	Expert	Error	Budgeted Error			Coverage
			10%	20%	30%	ECE

CE	20	22.31(0.54)	27.64(1.08)	22.44(0.62)	22.31(0.54)	79.21(1.13)
						5.45(0.40)

	40	20.65(0.98)	36.35(2.24)	29.27(2.34)	22.00(1.55)	66.93(2.83)
						9.61(1.28)

	60	16.22(0.19)	49.57(1.87)	42.17(1.78)	34.31(1.93)	48.21(2.00)
						11.09(0.32)

OvA	20	24.33(2.01)	24.33(2.01)	24.33(2.01)	24.33(2.01)	93.09(0.48)
						4.53(0.37)

	40	25.82(2.23)	29.00(3.10)	25.82(2.24)	25.82(2.23)	82.90(2.35)
						8.36(0.95)

	60	19.33(1.86)	28.78(2.87)	21.85(2.63)	19.33(1.86)	74.81(1.78)
						7.64(0.60)

A-SM	20	21.94(0.24)	21.94(0.24)	21.94(0.24)	21.94(0.24)	98.35(0.11)
						4.17(0.21)

	40	21.22(0.77)	**21.34(0.90)**	21.22(0.77)	21.22(0.77)	90.64(0.99)
						5.57(0.50)

	60	18.40(0.74)	**22.20(1.31)**	18.40(0.74)	18.40(0.74)	83.95(0.91)
						5.11(0.23)

DCE
(Proposed)
	20	**21.21(0.23)**	**21.44(0.22)**	**21.21(0.23)**	**21.21(0.23)**	88.15(0.21)
						1.68(0.27)

	40	**19.09(0.37)**	22.71(0.40)	**19.10(0.38)**	**19.09(0.37)**	80.95(0.86)
						4.59(0.35)

	60	**15.81(0.31)**	24.89(0.62)	**18.20(0.57)**	**15.81(0.31)**	74.59(0.70)
						5.57(0.23)

Mechanical Turk. We randomly partition the data into 80% training data and 20% test data for each trial.

We also conduct experiments on ImageNet-16H (Kerrigan et al., 2021) with real-world experts for noise type "110" and "125", using an 80-20 train-test split for each trial. The experimental results are reported in Table 6 in Appendix C.

Following the previous works (Mozannar & Sontag, 2020; Verma & Nalisnick, 2022; Cao et al., 2023), we employ a wide residual network (Zagoruyko & Komodakis, 2016) to parameterize the scoring function $g(\boldsymbol{x})$ and SGD is used for optimization. We train the model on each dataset for 400 epochs on 8 NVIDIA GeForce 3090 GPUs. The learning rate is chosen from $\{3e-1, 1e-1, 3e-2, 1e-2\}$ and the batch size is chosen from $\{512, 1024\}$, i.e., $\{64, 128\}$ on each GPU. The weight decay is set as $5e-4$.

Metric. To better evaluate the performance of the L2D system, we recorded metrics across multiple dimensions during our experiments. We report the misclassification error for the L2D system and the error with deferral budget for each method on each dataset, where the misclassification error denotes the misclassification error rate for the L2D systems, i.e., the average 0-1-deferral loss on the test data. The budget represents the maximum allowable proportion of instances that the L2D system can defer to the expert. Concretely, if we use $b\%$ to denote the budget. We only defer the instance with the top $b\%$ $\widehat{\mathbb{P}}(Y = M|\boldsymbol{x})$ to the expert when the coverage is below $1 - b\%$. For the other instances, we accept the prediction made by the model.

We also report the coverage, and expected calibration error (ECE) in experimental results, and coverage stands for the proportion of instances the L2D system has *not* deferred. The ECE in L2D could be defined as:

$$\mathrm{ECE}(\widehat{\mathbb{P}}(Y = M|\cdot)) = \mathbb{E}_{p(\boldsymbol{x})}[|\mathbb{P}(Y = M|\widehat{\mathbb{P}}(Y = M|\boldsymbol{x}) = c) - c|].$$

Concretely, this ECE could measure how well the estimated confidence aligns with the true likelihood of the experts making the right prediction, providing insights into the model's reliability and accuracy of its uncertainty estimates.

Baselines. We compared our proposed Dependent Cross-Entropy Loss with the previous surrogate loss including Cross-Entropy (CE) based loss (Mozannar & Sontag, 2020), One-versus-All (OvA) based loss (Verma & Nalisnick, 2022) and Asymmetric SoftMax (A-SM) parametrization based method (Cao et al., 2023). Since the confidence estimator induced from CE is unbounded, we clip the estimates to fall within the range of [0, 1] to ensure the validity.

Experimental Results. We run 5 trials on each dataset for each method. The best results in terms of the misclassifica-

Table 2. Test performance of each method on CIFAR-100 for 5 trials with $p = 75\%$. The mean(%)(standard error(%)) of related metrics are reported. The best method for the misclassification error and budgeted errors are highlighted in boldface.

Method	Expert	Error	Budgeted Error			Coverage
			10%	20%	30%	ECE

CE	20	23.57(0.38)	24.50(0.40)	23.57(0.38)	23.57(0.38)	85.07(0.38)
						6.12(0.13)

	40	23.21(0.80)	29.47(1.48)	24.39(1.16)	23.21(0.80)	76.58(3.28)
						11.88(0.59)

	60	21.18(0.18)	32.08(0.43)	26.25(0.30)	21.19(0.19)	70.47(0.42)
						15.97(0.12)

OvA	20	23.63(0.35)	24.54(0.35)	23.63(0.35)	23.63(0.35)	85.05(0.32)
						5.91(0.38)

	40	22.73(0.42)	28.86(1.05)	23.78(0.79)	22.73(0.42)	76.38(1.53)
						12.00(0.77)

	60	21.18(0.27)	32.40(0.48)	26.53(0.56)	21.24(0.28)	70.09(0.66)
						16.06(0.34)

A-SM	20	22.21(0.16)	22.21(0.16)	22.21(0.16)	22.21(0.16)	99.50(0.07)
						3.85(0.32)

	40	22.49(0.47)	**22.49(0.47)**	22.49(0.47)	22.49(0.47)	96.42(0.37)
						5.06(0.10)

	60	21.06(0.54)	**21.06(0.54)**	21.06(0.54)	21.06(0.54)	92.54(0.22)
						5.20(0.45)

DCE
(Proposed)
	20	**22.12(0.27)**	**22.13(0.26)**	**22.12(0.27)**	**22.12(0.27)**	90.08(0.29)
						1.70(0.15)

	40	**21.28(0.48)**	22.91(0.61)	**21.28(0.48)**	**21.28(0.48)**	84.19(0.47)
						6.88(0.65)

	60	**19.81(0.85)**	24.08(1.14)	**19.84(0.89)**	**19.81(0.85)**	80.27(0.59)
						11.43(1.81)

Table 3. Test performance of each method on CIFAR-100N for 5 trials. The mean(%)(standard error(%)) of related metrics are reported. The best method for the misclassification error and budgeted errors are highlighted in boldface.

Method	Error	Budgeted Error			Coverage
		10%	20%	30%	ECE

CE	25.61(0.52)	31.97(0.60)	25.94(0.54)	25.61(0.52)	79.69(1.13)
					27.29(0.50)

OvA	32.07(0.53)	54.32(1.31)	46.97(1.33)	40.24(1.17)	55.35(2.07)
					32.23(0.25)

A-SM	28.50(0.34)	49.06(1.12)	41.66(1.06)	34.91(0.99)	58.38(1.53)
					31.73(0.37)

DCE	**21.34(0.34)**	**26.94(0.41)**	**21.88(0.48)**	**21.34(0.34)**	78.74(0.63)
					21.94(0.42)

tion error and the budgeted errors are highlighted in boldface. The experimental results on synthetic experts using CIFAR-100 with $p = 94\%$ and $p = 75\%$ are meticulously presented in Tables 1 and 2 respectively. As illustrated by these 2 tables, our proposed DCE loss consistently outperforms all baselines across different expert levels of misclassification error, empirically validating the superiority of our proposed DCE loss.

All methods demonstrate lower error rates as k or p increases, with more instances being deferred to the expert. This demonstrates the effective utilization of human expert capabilities by existing methods.

Furthermore, we observe that as k increases, the DCE loss outperforms baseline models by larger gaps in misclassification error. This improvement indicates that with more classes accurately predicted by the expert, the L2D system trained with the DCE loss can collaborate more effectively with the expert, achieving better performance.

Almost all methods demonstrated superior performance in terms of ECE when $p = 94\%$. This suggests that higher expert accuracy (i.e., closer to 1) simplifies confidence estimation, as neural networks are often overconfident in their confidence estimates. Additionally, the confidence estimator developed with the DCE loss shows excellent performance at $k = 20$. This reveals that when the expert focuses on a

Table 4. Test performance of each method on CIFAR-10N with "Worse Label" for 5 trials. The mean(%)(standard error(%)) of related metrics are reported in Table. The best method for the misclassification error and budgeted errors are highlighted in boldface.

Method	Error	Budgeted Error			Coverage
		10%	20%	30%	ECE

CE	17.83(0.28)	25.62(1.15)	19.75(1.00)	17.83(0.28)	76.67(1.48)
					36.03(1.00)

OvA	19.96(0.97)	29.63(1.99)	23.30(2.00)	20.03(1.08)	73.43(2.33)
					37.86(0.84)

A-SM	21.20(0.82)	30.70(1.82)	24.82(1.77)	21.21(0.82)	73.11(2.55)
					36.92(0.96)

DCE	**10.69(0.43)**	**10.69(0.43)**	**10.69(0.43)**	**10.69(0.43)**	95.25(0.23)
					8.92(1.13)

Table 5. Test performance of each method on CIFAR-10N with "Aggregate Label" for 5 trials. The mean(%)(standard error(%)) of related metrics are reported. The best method for the misclassification error and budgeted errors are highlighted in boldface.

Method	Error	Budgeted Error			Coverage
		10%	20%	30%	ECE

CE	8.20(1.43)	33.84(7.42)	24.85(7.36)	16.52(6.73)	61.36(8.76)
					6.74(0.45)

OvA	8.52(0.23)	64.15(3.06)	54.59(3.00)	45.23(2.96)	28.55(3.29)
					9.83(0.52)

A-SM	8.82(0.31)	64.40(1.68)	54.95(1.67)	45.68(1.67)	28.43(1.85)
					9.76(0.45)

DCE	**6.34(0.21)**	**20.82(2.44)**	**12.32(2.33)**	**6.38(0.26)**	72.62(2.78)
					6.81(0.45)

limited number of classes, the L2D system trained with the DCE loss can more precisely identify these classes.

Table 3, Table 4 and Table 5 present the experimental results on real-world experts using CIFAR-100N and CIFAR-10N, respectively. For CIFAR-10N, we conducted experiments on 2 types of experts, i.e., "Worse" and "Aggregate". Compared with synthetic experts, the human-model dependence pattern between Y and M is more complicated in real-world experts. Despite this complexity, the DCE loss still significantly outperforms all baselines on real-world experts.

For misclassification errors with a deferral budget, the DCE loss achieves the best performance across all budgets and datasets on real-world experts, thus the DCE loss can handle the scenarios with budget requirements. We can observe that the coverage of our method is always higher than the baselines with real-world experts, which shows that the DCE loss can induce an ideal model. Moreover, the DCE loss demonstrates a larger performance gap compared with baselines in real-world scenarios than in synthetic ones, highlighting its superior ability to handle complex situations.

6. Conclusion

In the *Learning to Defer* (L2D) problem, the Bayes optimality serves as the foundational criterion for designing a consistent surrogate loss, playing a crucial role in the formulation of the consistent surrogate loss. However, the existing Bayes optimality shown by Mozannar & Sontag (2020) fails to consider the dependence pattern between human and model, while this dependence pattern appears commonly in real-world scenarios since human experts may achieve better performance on specific classes. To address this issue, we provided a new formulation of the Bayes optimality called *dependent Bayes optimality*, and presented a deferral principle based on the dependent Bayes optimality. This deferral principle enables us to determine whether to defer based on the dependence pattern observed in training data. Following the deferral principle, we further proposed a *dependent cross-entropy* (DCE) loss for the L2D problem. The DCE loss is a consistent surrogate loss and can induce a bounded confidence estimator for the expert. To empirically validate our proposed surrogate loss, we conducted extensive experiments on various benchmark datasets, involving both synthetic and real-world experts. The comprehensive experimental results demonstrate that our proposed DCE loss consistently outperforms the baselines, demonstrating the superiority of our method. In future work, we will further explore whether there exist other interesting patterns that can help design a better consistent surrogate loss, and investigate the design of more efficient surrogate losses based on the deferral principle.

Acknowledgement

This work is supported by the Chongqing Overseas Chinese Entrepreneurship and Innovation Support Program.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alves, J. V., Leitão, D., Jesus, S., Sampaio, M. O., Liébana, J., Saleiro, P., Figueiredo, M. A., and Bizarro, P. Cost-sensitive learning to defer to multiple experts with workload constraints. *arXiv preprint arXiv:2403.06906*, 2024.

Bartlett, P. L. and Wegkamp, M. H. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008.

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

Cao, Y., Mozannar, H., Feng, L., Wei, H., and An, B. In defense of softmax parametrization for calibrated and consistent learning to defer. In *NeurIPS*, 2023.

Chalkidis, I., Androutsopoulos, I., and Aletras, N. Neural legal judgment prediction in English. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *ACL*, pp. 4317–4323, Florence, Italy, July 2019. Association for Computational Linguistics.

Charoenphakdee, N., Cui, Z., Zhang, Y., and Sugiyama, M. Classification with rejection based on cost-sensitive classification. In *ICML*, 2020.

Charusaie, M.-A., Mozannar, H., Sontag, D., and Samadi, S. Sample efficient learning of predictors that complement humans. In *ICML*, pp. 2972–3005. PMLR, 2022.

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. Learning with rejection. In Ortner, R., Simon, H. U., and Zilles, S. (eds.), *Algorithmic Learning Theory*, pp. 67–82, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46379-7.

De, A., Okati, N., Zarezade, A., and Rodriguez, M. G. Classification under human assistance. In *AAAI*, pp. 5905–5913, 2021.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. *Advances in neural information processing systems*, 33:10948–10960, 2020.

Finocchiaro, J., Frongillo, R. M., and Waggoner, B. An embedding framework for consistent polyhedral surrogates. In *NeurIPS*, 2019.

Gao, W. and Zhou, Z.-H. On the consistency of multi-label learning. *Artif. Intell.*, 199–200(1):22–44, jun 2013. ISSN 0004-3702.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, pp. 362–386, Apr 2020.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *ICML*, pp. 1321–1330. PMLR, 2017.

Hemmer, P., Thede, L., Vössing, M., Jakubik, J., and Kühl, N. Learning to defer with limited expert predictions. In *AAAI*, pp. 6002–6011, Jun. 2023.

Joshi, S., Parbhoo, S., and Doshi-Velez, F. Pre-emptive learning-to-defer for sequential medical decision-making under uncertainty. *arXiv preprint arXiv:2109.06312*, 2021.

Kerrigan, G., Smyth, P., and Steyvers, M. Combining human predictions with model probabilities via confusion matrices and calibration. In *NeurIPS*, 2021.

Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I. S. Consistent multilabel classification. In *NeurIPS*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Liu, S., Cao, Y., Zhang, Q., Feng, L., and An, B. Mitigating underfitting in learning to defer with consistent losses. In *AISTATS*, pp. 4816–4824. PMLR, 2024.

Lykouris, T. and Weng, W. Learning to defer in content moderation: The human-ai interplay. *arXiv preprint arXiv:2402.12237*, 2024.

Madras, D., Pitassi, T., and Zemel, R. Predict responsibly: improving fairness and accuracy by learning to defer. In *NeurIPS*, volume 31, 2018.

Mao, A., Mohri, C., Mohri, M., and Zhong, Y. Two-stage learning to defer with multiple experts. In *NeurIPS*, 2024.

Mozannar, H. and Sontag, D. Consistent estimators for learning to defer to an expert. In III, H. D. and Singh, A. (eds.), *ICML*, volume 119 of *Proceedings of Machine*

Learning Research, pp. 7076–7087. PMLR, 13–18 Jul 2020.

Mozannar, H., Satyanarayan, A., and Sontag, D. Teaching humans when to defer to a classifier via exemplars. In *AAAI*, pp. 5323–5331, 2022.

Mozannar, H., Lang, H., Wei, D., Sattigeri, P., Das, S., and Sontag, D. Who should predict? exact algorithms for learning to defer to humans. In *AISTATS*, pp. 10520–10545. PMLR, 2023.

Narasimhan, H., Jitkrittum, W., Menon, A. K., Rawat, A. S., and Kumar, S. Post-hoc estimators for learning to defer to an expert. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *NeurIPS*, 2022.

Ni, C., Charoenphakdee, N., Honda, J., and Sugiyama, M. On the calibration of multiclass classification with rejection. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *NeurIPS*, volume 32. Curran Associates, Inc., 2019a.

Ni, C., Charoenphakdee, N., Honda, J., and Sugiyama, M. On the calibration of multiclass classification with rejection. In *NeurIPS*, 2019b.

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

Reid, M. D. and Williamson, R. C. Composite binary losses. *J. Mach. Learn. Res.*, 11:2387–2422, 2009.

Scott, C. D. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *ICML*, 2011.

Scott, C. D. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012.

Straitouri, E., Wang, L., Okati, N., and Rodriguez, M. G. Improving expert predictions with prediction sets. *arXiv preprint arXiv:2201.12006*, 2022.

Tschandl, P., Rinner, C., Apalla, Z., Argenziano, G., Codella, N., Halpern, A., Janda, M., Lallas, A., Longo, C., Malvehy, J., et al. Human–computer collaboration for skin cancer recognition. *Nature Medicine*, 26(8):1229–1234, 2020.

Verma, R. and Nalisnick, E. T. Calibrated learning to defer with one-vs-all classifiers. In *ICML*, 2022.

Wang, D.-B., Feng, L., and Zhang, M.-L. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. In *NeurIPS*, volume 34, pp. 11809–11820, 2021.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. Mitigating neural network overconfidence with logit normalization. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23631–23644. PMLR, 17–23 Jul 2022.

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. In *ICLR*, 2021.

Yuan, M. and Wegkamp, M. Classification methods with reject option based on convex risk minimization. *J. Mach. Learn. Res.*, 11:111–130, mar 2010a. ISSN 1532-4435.

Yuan, M. and Wegkamp, M. H. Classification methods with reject option based on convex risk minimization. *J. Mach. Learn. Res.*, 11:111–130, 2010b.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*. British Machine Vision Association, 2016.

Zhang, M., Ramaswamy, H. G., and Agarwal, S. Convex calibrated surrogates for the multi-label f-measure. *Proceedings of machine learning research*, 119:11246–11255, 2020.

Zhang, T. Statistical analysis of some multi-category large margin classification methods. *J. Mach. Learn. Res.*, 5: 1225–1251, 2004a.

Zhang, T. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5(Oct):1225–1251, 2004b.

Zoabi, Y., Deri-Rozov, S., and Shomron, N. Machine learning-based prediction of covid-19 diagnosis based on symptoms. *npj Digital Medicine*, Jan 2021.

A. Proof of Theorem 4.1

To begin with proof, let us define

$$
\begin{aligned}
C^{\perp}_{\mathrm{DCE}}(g(\boldsymbol{x})) = \sum_{y \in \mathcal{Y}} \Big[& -\mathbb{P}(Y = y, M = y|\boldsymbol{x})\big(\log(\psi^{y}_{\mathcal{Y}}(g(\boldsymbol{x}))) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/q}(g(\boldsymbol{x})))\big) \\
& - \mathbb{P}(Y = y, M \neq y|\boldsymbol{x})\psi_{\mathcal{Y}} \log(\psi^{y}_{\mathcal{Y}}(g(\boldsymbol{x})))\Big]
\end{aligned}
$$

The conditional surrogate risk *w.r.t.* $\boldsymbol{x}$. We first proof that $\underset{\tilde{y} \in \mathcal{Y}}{\arg\max}\, g^*_{\tilde{y}}(\boldsymbol{x}) = \arg\max_{\tilde{y} \in \mathcal{Y}} \eta_{\tilde{y}}(\boldsymbol{x})$ by contradiction. Let us use $r = \arg\max_{\tilde{y} \in \mathcal{Y}} \eta_{\tilde{y}}(\boldsymbol{x})$, $q = \underset{\tilde{y} \in \mathcal{Y}}{\arg\max}\, g^*_{\tilde{y}}(\boldsymbol{x})$ to denote the index of maximum dimension in posterior distribution and scoring function respectively. For simplicity in notation, we represent the score vector outputted by the scoring function as $\boldsymbol{s} = g(\boldsymbol{x})$, and $\boldsymbol{s}^* = g^*(\boldsymbol{x})$

Suppose $r \neq q$, i.e. $\boldsymbol{s}^*_q > \boldsymbol{s}^*_r$. Then we show that we could obtain a lower value of conditional surrogate risk by switching the value between $\boldsymbol{s}^*_q$ and $\boldsymbol{s}^*_r$. Let us define $\tilde{\boldsymbol{s}}$ the score vector obtained by switching the value between $\boldsymbol{s}^*_q$ and $\boldsymbol{s}^*_r$, i.e. $\tilde{\boldsymbol{s}}_r = s^*_q$, $\tilde{\boldsymbol{s}}_q = s^*_r$ and $\tilde{\boldsymbol{s}}_i = s^*_i$ for $i \in \mathcal{Y}^{\perp}, i \neq r, q$. Thus $\underset{\tilde{y} \in \mathcal{Y}}{\arg\max}\, \tilde{\boldsymbol{s}} = r$ Then $C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) - C^{\perp}_{\mathrm{DCE}}(\tilde{\boldsymbol{s}})$ could be expressed as:

$$
\begin{aligned}
& C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) - C^{\perp}_{\mathrm{DCE}}(\tilde{\boldsymbol{s}}) \\
& = \mathbb{P}(Y = q, M = q|\boldsymbol{x})\big(\log(\psi^{q}_{\mathcal{Y}}(\tilde{\boldsymbol{s}})) - \log(\psi^{q}_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\tilde{\boldsymbol{s}})) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/q}(\boldsymbol{s}^*))\big) \\
& + \mathbb{P}(Y = q, M \neq q|\boldsymbol{x})\big(\log(\psi^{q}_{\mathcal{Y}^{\perp}}(\tilde{\boldsymbol{s}})) - \log(\psi^{q}_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& + \mathbb{P}(Y = r, M = r|\boldsymbol{x})\big(\log(\psi^{r}_{\mathcal{Y}}(\tilde{\boldsymbol{s}})) - \log(\psi^{r}_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/}(\tilde{\boldsymbol{s}})) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/q}(\boldsymbol{s}^*))\big) \\
& + \mathbb{P}(Y = r, M \neq r|\boldsymbol{x})\big(\log(\psi^{r}_{\mathcal{Y}^{\perp}}(\tilde{\boldsymbol{s}})) - \log(\psi^{r}_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& \overset{(a)}{=} \mathbb{P}(Y = q, M = q|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_r)}{\exp(\boldsymbol{s}^*_q)}) + \log(\frac{\exp(\boldsymbol{s}^*_r) + \sum_{i \neq r,q} \exp(\boldsymbol{s}^*_i)}{\exp(\boldsymbol{s}^*_r) + \sum_{i \neq r,q} \exp(\boldsymbol{s}^*_i))}\big) \\
& + \mathbb{P}(Y = q, M \neq q|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_r)}{\exp(\boldsymbol{s}^*_q)})\big) \\
& + \mathbb{P}(Y = r, M = r|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_q)}{\exp(\boldsymbol{s}^*_r)}) + \log(\frac{\exp(\boldsymbol{s}^*_r) + \sum_{i \neq r,q} \exp(\boldsymbol{s}^*_i)}{\exp(\boldsymbol{s}^*_r) + \sum_{i \neq r,q} \exp(\boldsymbol{s}^*_i)}\big) \\
& + \mathbb{P}(Y = r, M \neq r|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_q)}{\exp(\boldsymbol{s}^*_r)})\big) \\
& = (\mathbb{P}(Y = r|\boldsymbol{x}) - \mathbb{P}(Y = q|\boldsymbol{x}))\log(\frac{\exp(\boldsymbol{s}^*_q)}{\exp(\boldsymbol{s}^*_r)}) \\
& > 0
\end{aligned}
$$

The equation (a) holds since $\sum_{i \in \mathcal{Y}} \exp(\boldsymbol{s}^*_i) = \sum_{i \in \mathcal{Y}} \exp(\tilde{\boldsymbol{s}}_i)$ and $\sum_{i \in \mathcal{Y}^{\perp}} \exp(\boldsymbol{s}^*_i) = \sum_{i \in \mathcal{Y}^{\perp}} \exp(\tilde{\boldsymbol{s}}_i)$. Because $\mathbb{P}(Y = r|\boldsymbol{x}) > \mathbb{P}(Y = q|\boldsymbol{x})$ and $\boldsymbol{s}^*_q > \boldsymbol{s}^*_r$, then $(\mathbb{P}(Y = r|\boldsymbol{x}) - \mathbb{P}(Y = q|\boldsymbol{x}))\log(\frac{\exp(\boldsymbol{s}^*_q)}{\exp(\boldsymbol{s}^*_r)}) > 0$,$C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) > C^{\perp}_{\mathrm{DCE}}(\tilde{\boldsymbol{s}})$. This contradicts to $\boldsymbol{s}^*$ is the minimizer of $C^{\perp}_{\mathrm{DCE}}$, thus $r = q$. Then we can conclude the proof of $r = q$.

Then we prove that $\boldsymbol{s}^*_r > \boldsymbol{s}^*_{\perp}$ when $\mathbb{P}(Y = r, Y \neq M|\boldsymbol{x}) > \mathbb{P}(Y \neq r, Y = M|\boldsymbol{x})$. Suppose $\boldsymbol{s}^*_{\perp} > \boldsymbol{s}^*_r$ when $\mathcal{Y}(Y = r, M \neq r)(\boldsymbol{x}) > \mathbb{P}(Y \neq r, Y = M|\boldsymbol{x})$ and we use $\boldsymbol{s}'$ to represent the score vector obtained by switching the value between $\boldsymbol{s}^*_r$ and $\boldsymbol{s}^*_{\perp}$, i.e. $\boldsymbol{s}'_r = \boldsymbol{s}^*_{\perp}$, $\boldsymbol{s}'_{\perp} = \boldsymbol{s}^*_r$ and $\boldsymbol{s}'_i = \boldsymbol{s}^*_i$ for all $i \in \mathcal{Y}, i \neq r$. We can directly derive that $r = \underset{\tilde{y} \in \mathcal{Y}}{\arg\max}\, \boldsymbol{s}'$. Then

$C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) - C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}')$ could be expressed as:

$$
\begin{aligned}
& C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) - C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}') \\
& = \mathbb{P}(Y=r, M=r|\boldsymbol{x})\big(\log(\psi^r_{\mathcal{Y}}(\boldsymbol{s}')) - \log(\psi^r_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}')) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}^*))\big) \\
& + \mathbb{P}(Y=r, M\neq r|\boldsymbol{x})\big(\log(\psi^r_{\mathcal{Y}^{\perp}}(\boldsymbol{s}')) - \log(\psi^r_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y=i, M=i|\boldsymbol{x})\big(\log(\psi^i_{\mathcal{Y}}(\boldsymbol{s}')) - \log(\psi^i_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}')) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}^*))\big) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y=i, M\neq i|\boldsymbol{x})\big(\log(\psi^i_{\mathcal{Y}^{\perp}}(\boldsymbol{s}')) - \log(\psi^i_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& = \mathbb{P}(Y=r, M\neq r|\boldsymbol{x})\big(\log(\psi^r_{\mathcal{Y}^{\perp}}(\boldsymbol{s}')) - \log(\psi^r_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y=i, M=i|\boldsymbol{x})\big(\log(\psi^i_{\mathcal{Y}}(\boldsymbol{s}')) - \log(\psi^i_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}')) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}^*))\big) \\
& = \mathbb{P}(Y=r, M\neq r|\boldsymbol{x})\log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\exp(\boldsymbol{s}^*_r)}) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y=i, M=i|\boldsymbol{x})\big(\log(\frac{\sum_{j\in\mathcal{Y}, j\neq r}\exp(\boldsymbol{s}^*_j) + \exp(\boldsymbol{s}^*_r)}{\sum_{j\in\mathcal{Y}, j\neq r}\exp(\boldsymbol{s}^*_j) + \exp(\boldsymbol{s}^*_{\perp})}) + \log(\frac{\exp(\boldsymbol{s}^*_r)}{\exp(\boldsymbol{s}^*_{\perp})}\frac{\sum_{j\in\mathcal{Y}, j\neq r}\exp(\boldsymbol{s}^*_j) + \exp(\boldsymbol{s}^*_{\perp})}{\sum_{j\in\mathcal{Y}, j\neq r}\exp(\boldsymbol{s}^*_j) + \exp(\boldsymbol{s}^*_r)}))) \\
& = \mathbb{P}(Y=r, M\neq r|\boldsymbol{x})\log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\exp(\boldsymbol{s}^*_r)}) \\
& - \mathbb{P}(Y\neq r, Y=M|\boldsymbol{x})\log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\exp(\boldsymbol{s}^*_r)}) \\
& = \big(\mathbb{P}(Y=r, M\neq r|\boldsymbol{x}) - \mathbb{P}(Y\neq r, Y=M|\boldsymbol{x})\big)\log(\frac{\exp(\boldsymbol{s}^*_r)}{\exp(\boldsymbol{s}^*_{\perp})}) > 0
\end{aligned}
$$

Thus $C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) > C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}')$, which is contradictory to $\boldsymbol{s}^*$ is a minimizer of $C^{\perp}_{\mathrm{DCE}}$, then we conclude the proof that $\boldsymbol{s}^*_r > \boldsymbol{s}^*_{\perp}$ if $\mathbb{P}(Y=r, Y\neq M|\boldsymbol{x}) > \mathbb{P}(Y\neq r, Y=M|\boldsymbol{x})$.

Lastly, we prove that $\boldsymbol{s}^*_{\perp} > \boldsymbol{s}^*_r$ when $\mathbb{P}(Y\neq r, Y=M|\boldsymbol{x}) > \mathbb{P}(Y=r, Y\neq M|\boldsymbol{x})$. Suppose $\boldsymbol{s}^*_r > \boldsymbol{s}^*_{\perp}$ when $\mathbb{P}(Y\neq r, Y=M|\boldsymbol{x}) > \mathbb{P}(Y=r, Y\neq M|\boldsymbol{x})$ and we still use $\boldsymbol{s}'$ to represent the score vector obtained by switching the value between $\boldsymbol{s}^*_r$ and $\boldsymbol{s}^*_{\perp}$, i.e. $\boldsymbol{s}'_r = \boldsymbol{s}^*_{\perp}$, $\boldsymbol{s}'_{\perp} = \boldsymbol{s}^*_r$ and $\boldsymbol{s}'_i = \boldsymbol{s}^*_i$ for all $i\in\mathcal{Y}, i\neq r$. Let $t = \underset{\tilde{y}\in\mathcal{Y}}{\arg\max}\, \boldsymbol{s}'_y$. We define $\epsilon = \sum_{i\in\mathcal{Y}, i\neq r}\exp(\boldsymbol{s}^*)$ and $\epsilon' = \sum_{i\in\mathcal{Y}, i\neq r}\exp(\boldsymbol{s}')$. We could derive that $\epsilon \geq \epsilon'$. Then $C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) - C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}')$ could be expressed as:

$$
\begin{aligned}
& C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}^*) - C^{\perp}_{\mathrm{DCE}}(\boldsymbol{s}') \\
& = \mathbb{P}(Y = r, M = r|\boldsymbol{x})\big(\log(\psi^r_{\mathcal{Y}}(\boldsymbol{s}')) - \log(\psi^r_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/t}(\boldsymbol{s}')) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}^*))\big) \\
& + \mathbb{P}(Y = r, M \neq r|\boldsymbol{x})\big(\log(\psi^r_{\mathcal{Y}^{\perp}}(\boldsymbol{s}')) - \log(\psi^r_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y = i, M = i|\boldsymbol{x})\big(\log(\psi^i_{\mathcal{Y}}(\boldsymbol{s}')) - \log(\psi^i_{\mathcal{Y}}(\boldsymbol{s}^*)) + \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/t}(\boldsymbol{s}')) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/r}(\boldsymbol{s}^*))\big) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y = i, M \neq i|\boldsymbol{x})\big(\log(\psi^i_{\mathcal{Y}^{\perp}}(\boldsymbol{s}')) - \log(\psi^i_{\mathcal{Y}^{\perp}}(\boldsymbol{s}^*))\big) \\
& = \mathbb{P}(Y = r, M = r|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\epsilon + \exp(\boldsymbol{s}^*_{\perp})}) - \log(\frac{\exp(\boldsymbol{s}^*_{r})}{\epsilon + \exp(\boldsymbol{s}^*_{r})}) + \log(\frac{\exp(\boldsymbol{s}^*_{r})}{\epsilon' + \exp(\boldsymbol{s}^*_{r})}) - \log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\epsilon + \exp(\boldsymbol{s}^*_{\perp})})\big) \\
& + \mathbb{P}(Y = r, M \neq r|\boldsymbol{x}) \log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\exp(\boldsymbol{s}^*_{r})}) \\
& + \sum_{i\in\mathcal{Y}, i\neq r} \mathbb{P}(Y = i, M = i|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_{i})}{\epsilon + \exp(\boldsymbol{s}^*_{\perp})}) - \log(\frac{\exp(\boldsymbol{s}^*_{i})}{\epsilon + \exp(\boldsymbol{s}^*_{r})}) + \log(\frac{\exp(\boldsymbol{s}^*_{r})}{\epsilon' + \exp(\boldsymbol{s}^*_{r})}) - \log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\epsilon + \exp(\boldsymbol{s}^*_{\perp})})\big) \\
& = \mathbb{P}(Y = r, M = r|\boldsymbol{x})\big(\log(\frac{\epsilon + \exp(\boldsymbol{s}^*_{r})}{\epsilon' + \exp(\boldsymbol{s}^*_{r})})\big) \\
& + \mathbb{P}(Y = r, M \neq r|\boldsymbol{x}) \log(\frac{\exp(\boldsymbol{s}^*_{\perp})}{\exp(\boldsymbol{s}^*_{r})}) \\
& + \mathbb{P}(Y \neq r, Y = M|\boldsymbol{x})\big(\log(\frac{\exp(\boldsymbol{s}^*_{r})}{\exp(\boldsymbol{s}^*_{\perp})}) + \log(\frac{\epsilon + \exp(\boldsymbol{s}^*_{r})}{\epsilon' + \exp(\boldsymbol{s}^*_{r})})\big) \\
& = \mathbb{P}(Y = r, M = r|\boldsymbol{x})\big(\log(\frac{\epsilon + \exp(\boldsymbol{s}^*_{r})}{\epsilon' + \exp(\boldsymbol{s}^*_{r})})\big) \\
& + \big(\mathbb{P}(Y \neq r, Y = M|\boldsymbol{x}) - \mathbb{P}(Y = r, M \neq r|\boldsymbol{x})\big) \log(\frac{\exp(\boldsymbol{s}^*_{r})}{\exp(\boldsymbol{s}^*_{\perp})}) \\
& + \mathbb{P}(Y = r, M \neq r|\boldsymbol{x}) \log(\frac{\epsilon + \exp(\boldsymbol{s}^*_{r})}{\epsilon' + \exp(\boldsymbol{s}^*_{r})}) > 0
\end{aligned}
$$

Which conclude the proof $\boldsymbol{s}^*_{\perp} > \boldsymbol{s}^*_{r}$ when $\mathbb{P}(Y \neq r, Y = M|\boldsymbol{x}) > \mathbb{P}(Y = r, Y \neq M|\boldsymbol{x})$ by contradiction.

B. Proof of Proposition 4.2

The formulation of DCE could be rewritten as:

$$
L^{\perp}_{\mathrm{DCE}}(g(\boldsymbol{x}), y, m) = \begin{cases} -\log(\psi^y_{\mathcal{Y}}(g(\boldsymbol{x}))) - \log(\psi^{\perp}_{\mathcal{Y}^{\perp}/q}(g(\boldsymbol{x}))) \\ \qquad (y = m) \\ -\log(\psi^y_{\mathcal{Y}}(g(\boldsymbol{x}))) - \log(\frac{\lambda(\boldsymbol{x})}{\lambda(\boldsymbol{x}) + \exp(g_{\perp}(\boldsymbol{x}))}) \\ \qquad (y \neq m) \end{cases}
$$

Thus $L^{\perp}_{\mathrm{DCE}}$ could be expressed as:

$$
L^{\perp}_{\mathrm{DCE}}(g(\boldsymbol{x}), y, m) = -\log(\frac{\exp(g_y(\boldsymbol{x}))}{\lambda(x)}) - \mathbb{I}_{y=m} \log(\frac{\exp(g_{\perp}(\boldsymbol{x}))}{\mu(\boldsymbol{x}) + \exp(g_{\perp}(\boldsymbol{x}))}) - \mathbb{I}_{m\neq y} \log(\frac{\lambda(\boldsymbol{x})}{\lambda(\boldsymbol{x}) + \exp(g_{\perp}(\boldsymbol{x}))})
$$

Then the conditional surrogate risk can be expressed as:

$$
\begin{aligned}
& C^{\perp}_{\mathrm{DCE}}(g(\boldsymbol{x})) \\
& = -\sum_{i \in \mathcal{Y}} \mathbb{P}(Y = i|\boldsymbol{x}) \log(\frac{\exp(g_i(\boldsymbol{x}))}{\lambda(\boldsymbol{x})}) \\
& - \mathbb{P}(Y = M|\boldsymbol{x}) \log(\frac{\exp(g_\perp(\boldsymbol{x})}{\mu(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x}))}) - \mathbb{P}(M \neq Y|\boldsymbol{x}) \log(\frac{\lambda(\boldsymbol{x})}{\lambda(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x}))})
\end{aligned}
$$

To minimize $-\sum_{i \in \mathcal{Y}} \mathbb{P}(Y = i|\boldsymbol{x}) \log(\frac{\exp(g_i(\boldsymbol{x}))}{\lambda(\boldsymbol{x})})$, we could directly obtained that $\mathbb{P}(Y = \tilde{y}|\boldsymbol{x}) = \psi^{\tilde{y}}_{\mathcal{Y}}(g^*(\boldsymbol{x}))$ due to the calibration of Cross-Entropy (Lemma 2 in Feng et al. (2020)).

Now we focus on deriving the minimizer *w.r.t.* $-\mathbb{P}(Y = M|\boldsymbol{x}) \log(\frac{\exp(g_\perp(\boldsymbol{x})}{\mu(\boldsymbol{x})+\exp(g_\perp(\boldsymbol{x}))}) - \mathbb{P}(M \neq Y|\boldsymbol{x}) \log(\frac{\lambda(\boldsymbol{x})}{\lambda(\boldsymbol{x})+\exp(g_\perp(\boldsymbol{x}))})$.

The derivative of $-\mathbb{P}(Y = M|\boldsymbol{x}) \log(\frac{\exp(g_\perp(\boldsymbol{x})}{\mu(\boldsymbol{x})+\exp(g_\perp(\boldsymbol{x}))}) - \mathbb{P}(M \neq Y|\boldsymbol{x}) \log(\frac{\lambda(\boldsymbol{x})}{\lambda(\boldsymbol{x})+\exp(g_\perp(\boldsymbol{x}))})$ *w.r.t.* $\exp(g_\perp(\boldsymbol{x}))$ is:

$$
\begin{aligned}
& \frac{\partial - \mathbb{P}(Y = M|\boldsymbol{x}) \log(\frac{\exp(g_\perp(\boldsymbol{x})}{\mu(\boldsymbol{x})+\exp(g_\perp(\boldsymbol{x}))}) - \mathbb{P}(M \neq Y|\boldsymbol{x}) \log(\frac{\lambda(\boldsymbol{x})}{\lambda(\boldsymbol{x})+\exp(g_\perp(\boldsymbol{x}))})}{\partial \exp(g_\perp \boldsymbol{x})} \\
& = \mathbb{P}(Y = M|\boldsymbol{x}) \frac{\mu(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x}))}{\exp(g_\perp(\boldsymbol{x}))} \cdot \frac{\mu(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x})) - \exp(g_\perp(\boldsymbol{x}))}{(\mu(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x})))^2} \\
& + \mathbb{P}(Y \neq M|\boldsymbol{x}) \frac{\lambda(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x}))}{\lambda(\boldsymbol{x})} \cdot \frac{\lambda(\boldsymbol{x})}{(\lambda(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x})))^2} \\
& = \mathbb{P}(Y = M|\boldsymbol{x}) \frac{\mu(\boldsymbol{x})}{\exp(g_\perp(\boldsymbol{x})) \cdot (\mu(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x})))} \\
& + \mathbb{P}(Y \neq M|\boldsymbol{x}) \frac{1}{\lambda(\boldsymbol{x}) + \exp(g_\perp(\boldsymbol{x}))}
\end{aligned}
$$

By setting this derivative to 0. We obtain that:

$$
\begin{aligned}
\mathbb{P}(Y = M|\boldsymbol{x}) \frac{\mu^*(\boldsymbol{x})}{\exp(g^*_\perp(\boldsymbol{x})) \cdot (\mu^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x})))} &= \mathbb{P}(Y \neq M|\boldsymbol{x}) \frac{1}{\lambda^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))} \\
\mathbb{P}(Y = M|\boldsymbol{x}) \frac{\mu^*(\boldsymbol{x})}{\mu^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))} &= \mathbb{P}(Y \neq M|\boldsymbol{x}) \frac{\exp(g^*_\perp(\boldsymbol{x}))}{\lambda^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))} \\
\mathbb{P}(Y = M|\boldsymbol{x}) \frac{\mu^*(\boldsymbol{x})}{\mu^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))} &= (1 - \mathbb{P}(Y = M|\boldsymbol{x})) \frac{\exp(g^*_\perp(\boldsymbol{x}))}{\lambda^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))} \\
\mathbb{P}(Y = M|\boldsymbol{x}) &= \frac{\exp(g^*_\perp(\boldsymbol{x})) \cdot (\mu^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x})))}{\exp(g^*_\perp(\boldsymbol{x})) \cdot (\mu^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))) + \mu^*(\boldsymbol{x}) \cdot (\lambda^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x})))} \\
\mathbb{P}(Y = M|\boldsymbol{x}) &= \frac{1}{1 + \frac{\mu^*(\boldsymbol{x}) \cdot (\lambda^*(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x})))}{\exp(g^*_\perp(\boldsymbol{x})) \cdot (\mu^(\boldsymbol{x}) + \exp(g^*_\perp(\boldsymbol{x}))}}
\end{aligned}
$$

Where we use $\mu^*(\boldsymbol{x})$ and $\lambda(\boldsymbol{x})$ to represent the summation *w.r.t.* the optimal scoring function g^*. Then we conclude the proof.

C. Additional Experiments Results

Table 6. Test performance of each method on ImageNet-16H with for 5 trials. The mean(%)(standard error(%)) of related metrics are reported in Table. The best and comparable methods for the misclassification error are highlighted in boldface.

Method	Error	Budgeted Error			Coverage
		10%	20%	30%	ECE
		Image Noise Type="110"			

CE	22.59(0.49)	68.08(3.66)	60.58(3.00)	52.50(2.84)	30.42(5.12)
					30.28(1.78)

OvA	22.67(2.02)	59.83(4.84)	51.42(4.72)	43.67(4.32)	38.92(4.62)
					19.72(1.73)

A-SM	22.42(1.59)	**50.83(4.51)**	**43.00(4.66)**	**35.58(3.94)**	49.83(3.40)
					10.99(2.07)

DCE	**21.33(1.50)**	60.58(3.98)	52.67(3.55)	45.17(3.30)	36.33(3.97)
					20.96(1.71)

		Image Noise Type="125"			

CE	33.00(3.20)	59.25(2.41)	53.25(2.88)	46.83(2.07)	44.50(2.29)
					33.54(3.18)

OvA	33.00(2.53)	50.17(2.64)	43.75(2.50)	38.92(1.91)	54.17(2.25)
					21.39(3.06)

A-SM	**31.92(2.02)**	**39.33(0.68)**	**34.58(0.79)**	**31.92(2.02)**	73.75(2.06)
					15.36(2.02)

DCE	32.58(2.26)	51.92(1.84)	46.25(1.37)	41.08(1.25)	50.92(3.00)
					22.88(3.47)