What Does It Take to Build a Performant Selective Classifier?

Stephan Rabanser[*], **Andy Wei Liu, Nicolas Papernot**
University of Toronto & Vector Institute

Abstract

An effective approach to ensure trustworthy deployment of machine learning models is through selective classification (SC), a paradigm in which models can opt out from making predictions under high uncertainty. At its core, SC offers a tradeoff between coverage and selective accuracy on accepted points. Past works typically compute this tradeoff empirically to compare SC methods but fail to identify what constitutes the best possible tradeoff for the considered SC setup. We bridge this gap by providing tight performance guarantees for confidence-based selective classifiers. Our analysis shows that SC performance is fully determined by (i) the survival function of the confidence distribution, and (ii) the degree of model calibration. Crucially, given a particular data distribution, this decomposition allows us to identify both the expected SC tradeoff and a tight upper bound. Our experimental results demonstrate that our expected tradeoffs match closely with empirically estimated tradeoffs and that any mismatch between our bound and the empirical tradeoff is the result of model calibration failure. Finally, we show that isolating the empirical survival function of confidences enables improved evaluation of practical selective classification approaches. Notably, this evaluation identifies ensembling-based methods as the strongest selective classifiers.

1 Introduction

Selective classification (SC) [El-Yaniv et al., 2010], also known as classification with rejection [Cortes et al., 2016, Ni et al., 2019, Charoenphakdee et al., 2021, Schreuder and Chzhen, 2021], presents a compelling paradigm for managing prediction reliability. It allows classifiers to withhold predictions where predictive uncertainty exceeds a predefined threshold, enhancing overall decision-making quality. This rejection mechanism introduces an inherent tradeoff between *coverage*, the fraction of accepted data points, and *selective accuracy*, the predictive performance on accepted points. To gauge the performance of a selective classifier, it is typical to compute this accuracy-coverage tradeoff and compare it across evaluation scenarios (i.e., differing datasets or SC methods).

Despite its important role in providing reliability guarantees for both vision tasks and language modeling, much of the existing literature [Hendrycks and Gimpel, 2016, Geifman and El-Yaniv, 2017, 2019, Liu et al., 2019, Huang et al., 2020, Rabanser et al., 2022] primarily evaluates the accuracy-coverage tradeoff on an empirical basis. This methodological preference facilitates the identification of both (i) competitive techniques from a set of SC approaches; as well as (ii) datasets that are easier to selectively classify than others. However, empirical evaluation of the SC tradeoff falls short in quantifying the optimality of a proposed SC method on a particular data distribution or dataset. In particular, it remains unclear when and to what extent selective classification can improve accuracy, even when the learning setup such as the hypothesis class and data distribution are known. This naturally raises the question:

[*]Correspondence to: stephan@cs.toronto.edu

Preprint. Under review.

How do the choice of data distribution and the assumed hypothesis class influence the performance profile of an ideal selective classifier?

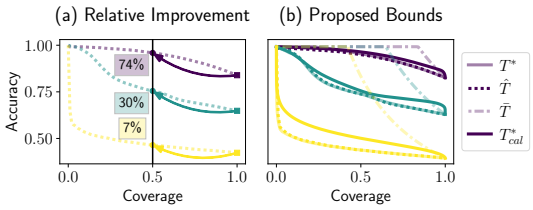

Figure 1: **Challenges in analyzing accuracy-coverage tradeoffs**. The dotted lines show 3 different empirically derived tradeoffs ($\hat{T}$). Solid and dash-dotted lines show different proposals for upper bounds. *(a)* Relative improvements over starting accuracy given a fixed coverage cost of 50% (black line) shows differing tradeoff convexity. *(b)* Previous upper bounds ($\bar{T}$) are loose at low utility. In this work we show how to derive tighter bounds on selective classification performance (T^*, T^*_{cal}).

Emphasizing the importance of this open question, consider a set of three selective classifiers trained on three distinct datasets with varying learning difficulty. We depict exemplary tradeoffs derived from these classifiers in different colors in Figure 1. It is apparent that the functional properties of the accuracy-coverage tradeoffs, most notably their convexity, differ across learning settings. For an exemplary fixed coverage cost of 50%, this disparity causes the already decently accurate model to attain a large fraction (74%) of its selective accuracy. In contrast, the least accurate model barely improves in its selective classification ability (7%) and requires a significantly higher coverage cost to attain comparable relative improvements. This raises the question of what exactly influences the shape of the accuracy-coverage tradeoff. To better understand the behavior of this tradeoff, previous works have derived upper performance bounds ($\bar{T}$ in Figure 1 (b)) [Geifman et al., 2018, Rabanser et al., 2023]. These bounds assume identifiability of the ideal acceptance ordering, i.e. a ranking of data points that accepts all correct points first and all incorrect points last. However, these bounds become loose for models with low utility [Galil et al., 2023]. The analysis presented in our work overcomes this looseness and uncovers the source of the convexity behavior, yielding a tighter bound that accurately reflects the behavior of (Bayes-)optimal classifiers (T^* and T^*_{cal} in Figure 1 (b)).

To provide intuition, we start our analysis of the accuracy-coverage tradeoff by classifying a univariate binary Gaussian mixture using a Bayes-optimal predictor. This setup is amenable to exact distributional analysis of the confidences required for selective classification. Stemming from this analysis, we show analytically that the survival function (i.e., the complementary cumulative distribution) of the confidence distribution directly yields the optimally attainable coverage. Next, informed by our distributional assumptions, we calculate the expected selective accuracy by modeling the distribution of correct acceptances. Together, these two quantities form our expected accuracy-coverage tradeoff T^* which closely models empirical tradeoffs $\hat{T}$. We also show how a tight upper bound T^*_{cal} can be obtained by assuming perfect calibration of the model's confidences. Building on this formalization, we later extend our analysis to unrestricted Gaussian mixtures and neural network feature spaces through Monte-Carlo sampling approximations. This allows us to approximate our expected tradeoffs and upper bounds even when a closed form solution of the confidence distribution is infeasible.

Our experiments across datasets show that the survival function of confidences determines the convexity properties of the accuracy-coverage tradeoff. Moreover, we find that whenever the empirical tradeoff and our upper bound overlap, selective classification is optimal. This insight gives rise to a new SC evaluation metric measuring the discrepancy between empirical and calibrated tradeoffs.

To summarize, our paper makes the following key contributions:

1. We identify the survival function of confidences and the calibration properties of the model as the primary factors influencing SC performance. To arrive at this decomposition, we rigorously analyze the accuracy-coverage tradeoff from logistic regression on a univariate Gaussian mixture.
2. We provide an algorithm which extends this formalism to arbitrary Gaussian mixtures and non-linear feature spaces extracted by neural networks. We show that, while no longer distributionally analyzable, we can employ sampling techniques to approximate the optimal performance profile.
3. We present a thorough evaluation of our tradeoff estimation using synthetic data and canonical datasets from the SC literature. We find that our tradeoffs closely match the empirical tradeoffs, and that any deviation from the upper bound correlates perfectly with miscalibration. Additionally, we show that our derived tradeoffs lead to improved evaluation of popular SC techniques, unveiling that only ensembling-based methods reliably improve SC performance.

2 Background & Related Work

Selective classification extends the standard supervised classification framework as follows.

Definition 2.1 (Selective Classifier)**.** *A selective classifier is a tuple (h, g) consisting of a classification function $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^D$ is the covariate space and $\mathcal{Y} = \{1, \ldots, K\}$ is the label space, and a selection function $g : \mathcal{X} \times (\mathcal{X} \to \mathcal{Y}) \to \mathbb{R}$. The selection function g outputs a score that is compared to a threshold $\tau \in \mathbb{R}$. The combined predictive model is defined as follows:*

$$(h, g)(\mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{if } g(\mathbf{x}, h) \geq \tau \\ \perp & \text{otherwise} \end{cases} . \tag{1}$$

A selective classifier outputs a class label if it is confident the prediction is correct; otherwise, it withholds the prediction. Abstention for a given data point $\mathbf{x}$ is determined by $g(\mathbf{x}, h)$ which assesses whether a model should make a prediction $h(\mathbf{x})$. If the value of $g(\mathbf{x}, h)$ is below a certain threshold τ, the prediction $h(\mathbf{x})$ is returned; if not, the system abstains by returning $\perp$.

Many past works have focused on deriving SC methods with competitive (h, g). The earliest method for selective prediction is the *Softmax Response* (SR) method [Hendrycks and Gimpel, 2016, Geifman and El-Yaniv, 2017]. This method uses the confidence of h as the selection score. To improve calibration and reduce estimation variance, ensembling-based methods have been proposed. *Deep Ensembles* (DE) [Lakshminarayanan et al., 2017] trains multiple models from scratch with varying initializations. Another ensembling approach is given in *Selective Classification via Training Dynamics* (SCTD) [Rabanser et al., 2022], which records intermediate models produced during training and then ensembles prediction over these models at test-time. Finally, a variety of SC methods such as *SelectiveNet* (SN) [Geifman and El-Yaniv, 2019], *Deep Gamblers* (DG) [Liu et al., 2019], and *Self-Adaptive Training* (SAT) [Huang et al., 2020] have been proposed that leverage explicit architecture and loss function adaptations. As a result, these methods directly change the training stage of the model.

The efficacy of a selective classifier is evaluated using the empirical accuracy-coverage tradeoff.

Definition 2.2 (Empirical Accuracy-Coverage Tradeoff)**.** *The empirical accuracy-coverage tradeoff for a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is a tuple $\hat{T} = (\hat{\xi}, \hat{\alpha})$. The empirical coverage $\hat{\xi} : \mathbb{R} \to [0, 1]$ represents the proportion of data points on which the classifier, given a threshold τ, decides to predict. The empirical selective accuracy $\hat{\alpha} : \mathbb{R} \to [0, 1]$ measures the correctness of predictions made on selected points. We formally define both $\hat{\xi}$ and $\hat{\alpha}$ for a selective classifier (h, g) as:*

$$\hat{\xi}_{h,g}(\tau) = \frac{|\{\mathbf{x} : g(\mathbf{x}, h) \leq \tau\}|}{|\mathcal{D}|} \qquad \hat{\alpha}_{h,g}(\tau) = \frac{|\{\mathbf{x} : h(\mathbf{x}) = y, g(\mathbf{x}, h) \leq \tau\}|}{|\{\mathbf{x} : g(\mathbf{x}, h) \leq \tau\}|} . \tag{2}$$

As a result of limited coverage, the selection function g defines a total order among data points which gives a preferred acceptance ordering: $\mathbf{x}_1$ is accepted before $\mathbf{x}_2$ iff $g(\mathbf{x}_1, h) > g(\mathbf{x}_2, h)$.

Effective selective classification strategies aim to maximize selective accuracy over the full coverage spectrum. However, since selective accuracy is improved by rejecting an increasing amount of incorrect points, boosting selective accuracy typically comes at a coverage cost (and vice-versa).

Accuracy-coverage tradeoff evaluation. The accuracy-coverage tradeoff can be summarized into a single metric, referred to as the area under the accuracy-coverage curve (AUACC), by integrating the selective accuracy over the full coverage spectrum. Geifman et al. [2018] show that this metric overly favor models which are already highly accurate at full coverage. To mitigate this dependence, Geifman et al. [2018] and Rabanser et al. [2023] propose upper bounds on the accuracy-coverage tradeoff. However, despite being accuracy-dependent, these bounds become looser at lower utility [Galil et al., 2023]. To mitigate this preference for highly accurate models, Galil et al. [2023] and Pugnana and Ruggieri [2023] propose to measure the area under the receiver operating characteristic (AUROC) of the classifier as a better performance evaluation. However, as noted by Cattelan and Silva [2023] and supported by Ding et al. [2020], the AUROC is a not a monotonic function of AUACC. As a result, it favors models that explicitly optimize AUROC over selective classifiers that are trained using different objectives. To provide initial guarantees for selective classification, El-Yaniv et al. [2010] present a thorough characterization of the accuracy-coverage tradeoff for realizable data. In follow-up work Wiener and El-Yaniv [2011] relax the realizability assumption and consider the agnostic setting. In contrast to our work, however, both works mainly consider existence statements of optimal selective classifiers without providing an a clear explanation of their instantiation in practice.

3 Determining Optimal Selective Classification Performance

To identify the expected accuracy-coverage tradeoff as well as a tight upper bound, we derive both the coverage and accuracy functions of optimal selective classifier employing the softmax response method. Our subsequent experiments however extend beyond the softmax response method to other SC approaches. We first characterize both the expected and ideal tradeoffs analytically in Section 3.1, assuming a Bayes-optimal classifier on a binary univariate Gaussian mixture. Later, we extend our analysis to generalized Gaussian mixtures in Section 3.2 and the feature spaces of neural networks in Section 3.3 through the use of sampling approximations. To provide consistent notation throughout this section, we refer to (Bayes-)optimal quantities via $\_^*$, Monte-Carlo-sampling derived quantities via $\tilde{\_}$, and empirically estimated quantities via $\hat{\_}$.

3.1 Deriving Bayes-Optimal Accuracy-Coverage Tradeoffs

Distributional assumption. We consider a balanced mixture of two univariate Gaussians:

$$p(x) = \frac{1}{2} \sum_{y \in \{-1,1\}} p(x|y) \qquad \text{with} \qquad p(x|y) = \begin{cases} \mathcal{N}(x|a, \sigma_x^2) & \text{if } y = 1 \\ \mathcal{N}(x|-a, \sigma_x^2) & \text{if } y = -1 \end{cases} . \tag{3}$$

Model assumption. We consider a logistic regression model which consists of the the composition of a linear function $z(x) = wx + b$ and a sigmoid function $s(x) = \sigma(z(x))$. The resulting model, which produces a prediction $\hat{y}(x)$ and a confidence score $c(x)$, is given as:

$$\hat{y}(x) = \begin{cases} 1 & s(x) \geq 0.5 \\ -1 & s(x) < 0.5 \end{cases} \qquad c(x) = |s(x) - 0.5| + 0.5 \ . \tag{4}$$

Bayes-optimal decision function. To understand the behavior of a selective classifier with $h := \hat{y}(\cdot)$ and $g := c(\cdot)$, we first need to derive an optimal classifier for our setup. The Bayes-optimal classifier [Bishop, 2006] corresponds to the optimal classifier under our assumptions as it operates under perfect knowledge of the data distribution $p(x)$. For the distributional and modelling assumptions presented above, we can determine the parameters $\theta^* = (w^*, b^*)$ and the expected accuracy γ^* of the Bayes optimal decision function: $z^*(x) = w^*x + b^*$ using the following fact.

Fact 3.1 (Bayes-Optimal Predictor)**.** *Let $p(x)$ be as in Equation 3. Then the Bayes-optimal decision function $z^*(x)$ under $p(x)$ is $z^*(x) = w^*x + b^*$ with $w^* = \frac{2a}{\sigma_x^2}$ and $b^* = 0$. The Bayes-optimal predictor has accuracy $\gamma^* = 1 - \Phi(-\frac{a}{\sigma_x}) = \Phi(\frac{a}{\sigma_x})$ where $\Phi(\cdot)$ corresponds to the cumulative distribution function of the standard normal distribution. A proof is provided in Appendix C.1.*

Having obtained the parameters θ^* and the expected accuracy γ^* of the Bayes-optimal classifier, we now derive the accuracy-coverage trade-off $T^* = (\xi^*, \alpha^*)$ for the Bayes-optimal classifier. In the following, we first present a derivation of the coverage function $\xi^* : [0.5, 1] \to [0, 1]$ and subsequently discuss the corresponding accuracy function $\alpha^* : [0.5, 1] \to [0, 1]$.[2] As both ξ^* and α^* are derived from the Bayes-optimal classifier, we refer to them as the *Bayes-optimal coverage/accuracy*.

3.1.1 Deriving the Bayes-Optimal Coverage Function ξ^*

In order to arrive at a rigorous definition of ξ^*, we need to first get a better understanding of the distribution of confidence scores c^*. This is necessary since these confidences determine whether our selective classifier accepts a point x by comparing $c(x) \geq \tau$ (Equation 1). Under the distributional and modelling assumptions stated above, we are able to characterize the distribution of c^* exactly. In particular, the distribution of c^* corresponds to a folded logit-normal distribution (see Definition B.2).

Proposition 3.2 (Bayes-Optimal Confidence Distribution)**.** *Assume Proposition 3.1 holds. Then, the confidences $c^*(x)$ of the Bayes-optimal predictor are distributed according to a folded logit-normal distribution with $\mu = \frac{2a^2}{\sigma_x^2}$ and $\sigma^2 = \frac{4a^2}{\sigma_x^2}$, formally $c^*(x) \sim$ FoldedLogitNormal$(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2}) + 0.5$.*

[2] Both ξ^* and α^* are evaluated at a particular confidence level $c \in [0.5, 1]$.

Proof. We prove Proposition 3.2 is 3 steps:

1. *Linear Transform (Figure 2 (a))*: If the data is Gaussian $x \sim \mathcal{N}(x|a, \sigma_x^2)$ and the decision function is linear $z^*(x) = \frac{2a^2}{\sigma_x^2}x$, then by applying the rules of affine transformations of random variables we get that the logits are again Gaussian with $z^* \sim p(z|y=1) = \mathcal{N}(z|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})$.
2. *Sigmoid transform (Figure 2 (b))*: As the logits z^* are Gaussian distributed and the subsequent application of the sigmoid $s^* = \sigma(z^*(x))$ is invertible, we apply univariate change of variables. This results in s^* being logit-normal distributed (see Definition B.1) with the same parameters as z^*: $s^* \sim p(s|y=1) = \text{LogitNormal}(s|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})$.
3. *Confidence folding (Figure 2 (c))*: Recall from Equation 4 that turning the sigmoid scores s^* into confidence scores c^* requires calculating the absolute distance to 0.5. Distributionally, this is modeled by density folding [Tsagris et al., 2014] the score distribution s^* at 0.5 (see Definition B.2). As a result, $c^* \sim p(c) = \text{FoldedLogitNormal}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2}) + 0.5$. □

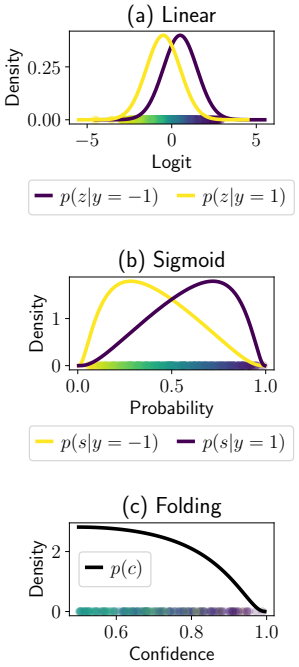

Figure 2: **Confidence distribution $p(c)$ derivation**. This derivation requires a distributional analysis of (a) the linear model; (b) the non-linear transformation to probabilities; and (c) the folding transformation from Eq. 4. We also show samples colored with the class-confidences.

Proposition 3.3 (Bayes-Optimal Coverage Function ξ^*)**.** *Assume Proposition 3.2 holds. Then the Bayes-optimal coverage function ξ^* is given by the survival function S^{FLN} (see Definition B.3) of the folded logit-normal distribution of Bayes-optimal confidences:* $\xi^*(c) := S^{FLN}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2}) = 1 - F^{FLN}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})$.

Proof. Recall from Definition 2.2 that g defines a preferred acceptance ordering. Under our modeling assumptions our selective classifier accepts points from most confident, i.e. $c^*(x) \approx 1$, to least confident, i.e. $c^*(x) \approx 0.5$ (increasing coverage from 0 to 1 with decreasing threshold τ). By Definition B.3, the rate of acceptance is then given by the survival function (i.e., the complementary CDF) of confidences. Given that the distribution of confidences is given by Proposition 3.2, the survival function is $\xi^*(c) := S^{\text{FLN}}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2}) = 1 - F^{\text{FLN}}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})$. See example in Figure 3 (a). □

3.1.2 Deriving the Bayes-Optimal Accuracy Function α^*

Having derived ξ^*, we now describe how to derive the Bayes-optimal accuracy function α^*.

Proposition 3.4 (Bayes-Optimal Accuracy Function α^*)**.** *Assume Proposition 3.3 holds and that S_+^{FLN} represents the contribution of the positive class to the folded logit-normal. Then the Bayes-optimal accuracy α^* is given by:* $\alpha^*(c) = S_+^{FLN}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})(\xi^*(c))^{-1}$.

Proof. From Proposition 3.3 we know that all acceptances are modeled via $\xi^*(c) = S^{\text{FLN}}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})$. As a result, the fraction of correct acceptances is the survival of the positive class: $S_+^{\text{FLN}}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})$. Applying Definition 2.2 gives $\alpha^*(c) = S_+^{\text{FLN}}(c|\frac{2a^2}{\sigma_x^2}, \frac{4a^2}{\sigma_x^2})(\xi^*(c))^{-1}$. See example in Figure 3 (a). □

We note that for models for which a linear increase in confidence leads to a linear increase in correct predictions, α^* is represented by a straight line. This is a desirable model property referred to as *calibration* [Guo et al., 2017]. Calibration is satisfied when the predicted probabilities of outcomes are directly proportional to the empirical frequencies of those outcomes. This property can be verified by checking if the predicted probabilities match the empirical probabilities for each class, formally

$$P(y = k \mid \hat{p}_k(\mathbf{X}) = p) = p \ , \tag{5}$$

where $\hat{p}_k(\mathbf{X})$ is the predicted probability (for class k) and p corresponds to a probability value. If Equation 5 holds, then the model and its confidence distribution is called *perfectly calibrated*.

Under perfect calibration, the Bayes-optimal accuracy simplifies:

Proposition 3.5 (Bayes-Optimal Calibrated Accuracy Function α^*_{cal})**.** *Assume Proposition 3.1 holds and that the confidence distribution $p(c)$ is perfectly calibrated. Then the Bayes-optimal accuracy function is $\alpha^*_{cal}(c) := 2((1-\gamma^*)c + \gamma^* - 0.5)$ for accuracy γ^*. See Appendix C.2 for a proof and Figure 3 (a) for an example.*

Finally, we can combine both the Bayes-optimal coverage and accuracies into the trade-offs T^*, T^*_{cal}:

Definition 3.6 (Bayes-Optimal Accuracy-Coverage Tradeoffs T^*, T^*_{cal})**.** *Assume the Bayes-optimal coverage ξ^* and accuracy α^*, α^*_{cal} are given by Propositions 3.3, 3.4, and 3.5. Then the Bayes-optimal accuracy-coverage tradeoffs are $T^*_{cal} = (\xi^*, \alpha^*_{cal})$, $T^* = (\xi^*, \alpha^*)$.*

We remark that the calibrated trade-off T^*_{cal} effectively acts as an upper bound on the empirical trade-off $\hat{T}$ whereas the uncalibrated T^* acts as an expectation of $\hat{T}$. See Figure 3 (b) for an example.

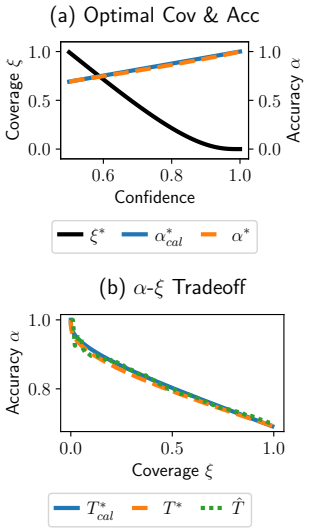

Figure 3: **(a) Bayes-optimal coverage ξ^*, accuracies α^* and α^*_{cal}. (b) The resulting accuracy (α) - coverage (ξ) trade-offs T^* and T^*_{cal}.** All quantities are derived from the same example as in Figure 2.

3.2 Approximating Optimal Tradeoffs Through Sampling

We now relax our setup to more flexible data distributions.

Setup. We assume a mixture distribution consisting of K multivariate Gaussians with mixture weights π_k with $k \in [K] = \{1, \dots, K\}$, $\pi_k \geq 0, \sum_{k\in[K]} \pi_k = 1$ and density $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|y=k)$ with $p(\mathbf{x}|y=k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Matching this assumption, we apply softmax classification $\boldsymbol{\sigma}(\mathbf{z})_k = \frac{e^{\mathbf{z}_k(\mathbf{x})}}{\sum_{j=1}^{K} e^{\mathbf{z}_j(\mathbf{x})}}$ where $\mathbf{z}_k(\mathbf{x}) = \langle \mathbf{w}_k, \mathbf{x} \rangle + b_k$ is the logit at index k and $\boldsymbol{\sigma}(\cdot)$ is the softmax function. The prediction is given by the class that gives the highest probability $\hat{y}(\mathbf{x}) = \arg\max_k \boldsymbol{\sigma}(\mathbf{z})_k$ with confidence $c(\mathbf{x}) = \max_k \boldsymbol{\sigma}(\mathbf{z})_k$.

Difficulty of distributional analysis. When attempting a similar analysis as in Section 3.1, a key step is to derive the distribution of confidences c^*. But, given the setup above, it is no longer possible to determine the ideal c^* analytically. We outline the following challenges (details in Appendix D):

1) Unknown optimal decision function: There is no known optimal linear decision function for our data and modeling assumptions. While quadratic discriminant analysis (QDA) fits our data assumption, it provides non-linear decision boundaries which we cannot analyze using our setup.

2) Non-invertibility of softmax: To study the distributional transformation from $p(\mathbf{x})$ to confidences c^* we need to apply multivariate change of variables which requires invertibility of transformations. The softmax can map distinct logit vectors to the same softmax values and is hence not invertible.

3) Unknown distribution of maximum confidence: To characterize c^* distributionally it is necessary to derive the distribution of the maximum of a multivaraite distribution (recall setup above). However, no general derivation of the maximum exists for multi-variate Gaussians with arbitrary covariance.

To mitigate these problems we approximate the optimal trade-offs $\tilde{T}$ and $\tilde{T}_{\text{cal}}$ via Monte-Carlo sampling [Kalos and Whitlock, 2009]. See Figure 8 and Algorithm 1 in the appendix for more details.

Approximating the optimal logistic regression parameters $\tilde{\theta} \approx \theta^*$. We approximate the optimal parameters $\tilde{\theta} = (\tilde{\mathbf{W}}, \tilde{\mathbf{b}})$ by sampling a large dataset $(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{y}}^{(1)})$ from the distribution $p(\mathbf{x})$ and then performing empirical risk minimization on a logistic regression model with sufficiently small learning rate. Under these conditions $\tilde{\theta} \approx \theta^*$. This also allows us to approximate the accuracy $\tilde{\gamma} \approx \gamma^*$.

Approximating the optimal coverage function $\tilde{\xi} \approx \xi^*$. We approximate the confidence $\tilde{c}$ by sampling an independent dataset $(\tilde{\mathbf{X}}^{(2)}, \tilde{\mathbf{y}}^{(2)})$ from $p(\mathbf{x})$ and computing the empirical confidence distribution across the full sample $\tilde{c} \sim \frac{1}{N} \sum_{n=1}^{N} \delta(c - c(\tilde{\mathbf{X}}^{(2)}))$ where $c(\tilde{\mathbf{X}}^{(2)}) = \max_k \boldsymbol{\sigma}(\langle \tilde{\mathbf{W}}, \tilde{\mathbf{X}}^{(2)} \rangle + \tilde{\mathbf{b}})$ and $\delta(\cdot)$ is the Dirac delta function defining an empirical distribution. This sampling procedure

side-steps the intractable distributional analysis. Finally, we derive $\tilde{\xi}$ similarly as in Proposition 3.3 by computing the sampling approximation of the survival function $\tilde{\xi}(c) = 1 - \tilde{F}_{\tilde{c}}$ where $\tilde{F}_{\tilde{c}}$ corresponds to the CDF associated with the empirical confidence distribution $\tilde{c}$.

Approximating the optimal accuracy functions $\tilde{\alpha} \approx \alpha^*$ and $\tilde{\alpha}_{\text{cal}} \approx \alpha^*_{\text{cal}}$. To estimate $\tilde{\alpha}$, we proceed in the same manner as in Proposition 3.4. We start by estimating the distribution of correct acceptance scores $\tilde{c}_+ \sim \frac{1}{N}\sum_{n=1}^{N} \delta(c - c(\tilde{\mathbf{X}}^{(2)}))\mathbb{1}[\hat{\mathbf{y}}^{(2)} = \tilde{\mathbf{y}}^{(2)}]$ and its corresponding survival function $\tilde{S}_+ = 1 - \tilde{F}_{\tilde{c}_+}$, where $\mathbb{1}[\hat{\mathbf{y}}^{(2)} = \tilde{\mathbf{y}}^{(2)}]$ corresponds to the indicator of correct classifications. To account for the expected amount of misclassifications, we rescale the survival function by the estimated model accuracy $\tilde{\gamma}$. Computing the fraction w.r.t. the approximated coverage then yields the approximation $\tilde{\alpha}(c) = \tilde{\gamma}\tilde{S}_+(c)\tilde{\xi}^{-1}(c)$. On the other hand, the optimal calibrated accuracy $\tilde{\alpha}_{\text{cal}}$ can be approximated via Proposition 3.5 by plugging in $\tilde{\gamma}$. Hence, $\tilde{\alpha}_{\text{cal}}(c) = 2((1-\tilde{\gamma})c + \tilde{\gamma} - 0.5)$.

3.3 Extension to Non-Linear Feature Spaces

We now discuss how to extend our approach to non-linear feature spaces as extracted by neural networks. A neural network $h_{\boldsymbol{\theta}}$ consists of (i) a non-linear feature extractor $\psi_{\boldsymbol{\theta}_\psi} : \mathbb{R}^D \to \mathbb{R}^E$ that maps an input $\mathbf{x}$ to an embedding vector $\mathbf{e}$ via a series of L non-linear transformations $\psi^{(1)}, \ldots, \psi^{(L)}$; followed by (ii) a linear classifier $\varphi_{\boldsymbol{\theta}_\varphi} : \mathbb{R}^E \to [0,1]^K$ applied on top of the embedding layer:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \varphi_{\boldsymbol{\theta}_\varphi}(\psi_{\boldsymbol{\theta}_\psi}(\mathbf{x})) \qquad \text{with } \psi_{\boldsymbol{\theta}_\psi}(\mathbf{x}) = \psi^{(L)} \circ \cdots \circ \psi^{(1)}(\mathbf{x}) \qquad \varphi_{\boldsymbol{\theta}_\varphi}(\mathbf{e}) = \boldsymbol{\sigma}(\langle \boldsymbol{\theta}_\varphi, \mathbf{e} \rangle) \ . \quad (6)$$

While our previous formalism does not extend to the feature extractor $\psi_{\boldsymbol{\theta}_\psi}$, it applies to the linear classification head $\varphi_{\boldsymbol{\theta}_\varphi}$ subject to a class-conditional Gaussian assumption. Hence, we propose to estimate class-wise Gaussian distributions in the feature space and then use the estimated Gaussians to conduct the same analysis as in Section 3.2. Concretely, for each class k we estimate $\mathcal{N}(\mathbf{e}|\tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k)$ with $\tilde{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{e}_i$ and $\tilde{\boldsymbol{\Sigma}}_c = \frac{1}{N_k - 1}\sum_{i=1}^{N_k}(\mathbf{e}_i - \tilde{\boldsymbol{\mu}}_k)(\mathbf{e}_i - \tilde{\boldsymbol{\mu}}_k)^\top$. As we later show in our experiments, this assumption is reasonable for many real-world datasets. In fact, multiple related works have used class-wise Gaussian approximations to perform out-of-distribution / adversarial example detection [Lee et al., 2018] as well as membership inference attacks [Carlini et al., 2022].

4 Experiments

Based on our experiments, we find that (i) our estimated tradeoffs approximate empirical tradeoffs closely; that (ii) performant selective classification necessitates model calibration; and that (iii) our calibrated upper bound allows for improved evaluation of SC methods. We publish our full code-base at the following URL: `<anonymized during submission, see attached supplement>`.

4.1 Uni- and Multivariate Gaussian Experiments

Setup. For the uni-variate case, we conduct a set of experiments with five sets of Gaussian pairs $\mathcal{N}(x|a, \sigma^2)$ and $\mathcal{N}(x|-a, \sigma^2)$ with $a \in \{2, 1, 0.5, 0.2, 0.1\}$ and $\sigma^2 = 1$. In the multivariate and multi-class setting, we conduct a set of experiments with four sets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ each consisting of 3 two-dimensional Gaussian components with varying covariances. We show results in Figure 4.

Accuracy-coverage tradeoff & calibration computation. For each dataset we compute the empirical tradeoff $\hat{T}$ by sampling points from each Gaussian mixture, fitting a logistic regression model, and computing the confidences based on Equation 4 on an i.i.d. validation set. For each dataset we then compute both the Bayes-optimal accuracy coverage tradeoffs $T^* = (\xi^*, \alpha^*)$, $T^*_{\text{cal}} = (\xi^*, \alpha^*_{\text{cal}})$ or an approximation to the optimal accuracy coverage tradeoff $\tilde{T} = (\tilde{\xi}, \tilde{\alpha})$, $\tilde{T}_{\text{cal}} = (\tilde{\xi}, \tilde{\alpha}_{\text{cal}})$. We also include tradeoff bounds proposed in Geifman et al. [2018], Rabanser et al. [2023] as $\bar{T}$. Alongside the tradeoffs, we also compute the *Brier score* (see Definition B.6) as a measure of model calibration.

Our synthetic evaluation on Gaussians yield the following interesting observations:

The functional shape of the Bayes-optimal accuracy-coverage tradeoff is determined by the survival function. Depending on the Bayes error across mixture components, the accuracy-coverage tradeoff exhibits different convexity properties. While models with high utility feature a mostly concave

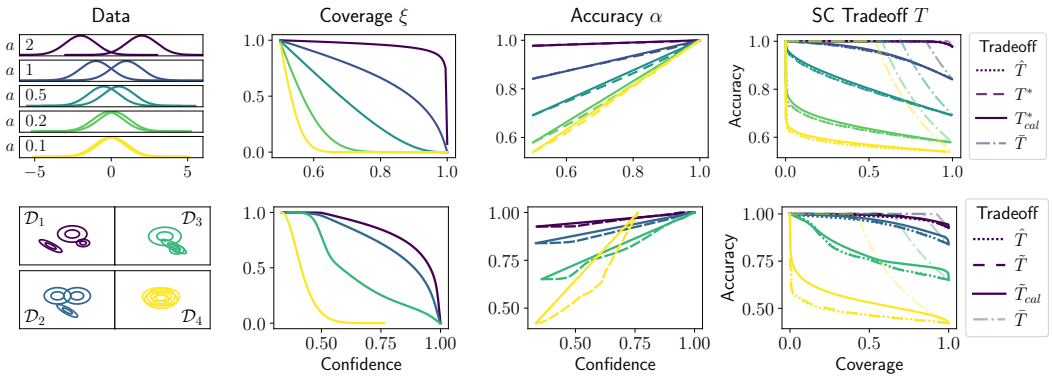

Figure 4: **Synthetic 1D (top) and 2D (bottom) Gaussian experiments**. We observe that the survival function of confidences (i.e., coverage) ξ determines the shape of the tradeoffs while the accuracy α determines the closeness to the empirical tradeoff. Previous bounds $\bar{T}$ are loose at low utility.

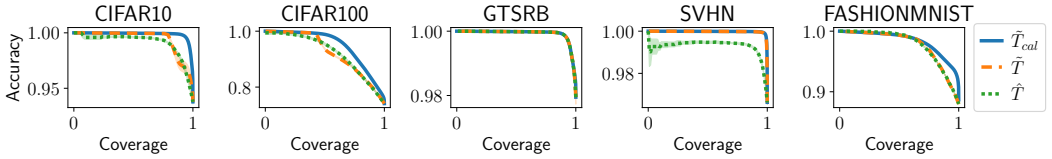

Figure 5: **Estimating SC performance based on neural network feature spaces**. We observe that $\tilde{T}_{\text{cal}}$ consistently provides an upper bound on $\hat{T}$ across experiments. However, the quality of the expected tradeoff $\bar{T}$ depends on the extent to which the Gaussian assumption holds (Appendix E.4).

tradeoff and models with low utility feature a mostly convex tradeoff, medium-utility models interpolate between the two settings. In contrast to previous bounds, our derivation captures this change in convexity for the first time via the survival functions ξ^* in our 1D and $\tilde{\xi}$ in our 2D experiments. As a result, we see that high-utility models can attain high selective accuracy with little coverage cost. On the other hand, low-utility models need a disproportionately high coverage cost to boost accuracy.

Performant selective classification requires model calibration. Across both 1D and 2D settings, we see that the calibrated tradeoffs, T^*_{cal} and $\tilde{T}_{\text{cal}}$, form upper bounds on the expected tradeoffs informed by the distribution of correct acceptances, T^* and $\tilde{T}$. In the 1D setting, both tradeoffs are close and any differences between the two can be attributed to calibration failures of the underlying model (see Figure 6). For our 2D experiments we observe stronger deviations as a result of overconfidence. For example, on $\mathcal{D}_4$ the confidence distribution should be more concentrated at $\frac{1}{3}$ to reflect high aleatoric uncertainty. Moreover, since we are able to derive the discrepancy between calibrated and uncalibrated tradeoffs, our analysis allows us to pinpoint calibration failures at specific coverage levels [Fisch et al., 2022]. Hence, mis-calibration is translated into sub-par SC performance in particular coverage regions.

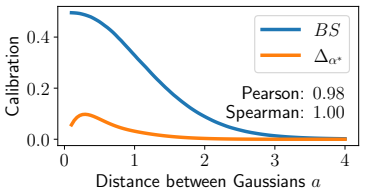

Figure 6: **Correlation between Brier score (BS) and the area between accuracies α^* and α^*_{cal}.** We observe a perfect ranking correlation, verifying that the gap $\Delta_{\alpha^*} = \int_c |\alpha^*_{\text{cal}} - \alpha^*| dc$ is indeed a result of insufficient model calibration.

4.2 Application to Neural Network Feature Spaces

Next, we apply our tradeoff estimation to the feature spaces of neural networks as described in Section 3.3. We train 5 randomly initialized ResNet-18 models [He et al., 2016] on the FashionMNIST [Xiao et al., 2017], CIFAR-10/CIFAR-100 [Krizhevsky et al., 2009], GTSRB [Houben et al., 2013], and SVHN [Netzer et al., 2011] datasets. From a validation set, we then compute the mixture coefficients, mean embeddings, and covariances $\{(\tilde{\pi}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K$ estimated for each class in the

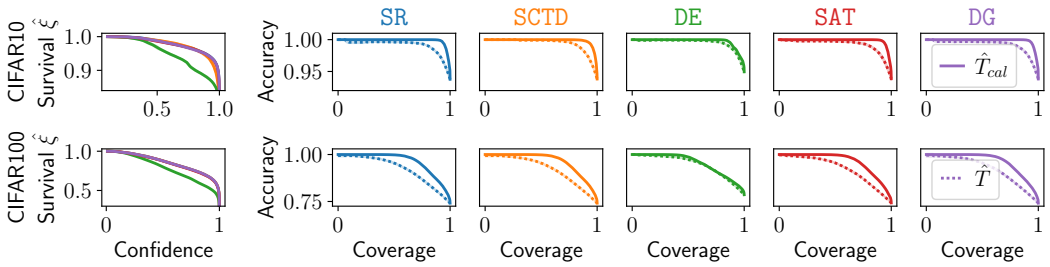

Figure 7: **Evaluation of SC methods**: The leftmost column shows the empirical survival curves $\hat{\xi}$ across datasets, other columns show both the empirical tradeoffs $\hat{T}$ and the calibrated upper bound $\hat{T}_{\text{cal}}$ for different SC methods. We observe that DE most closely matches its own upper bound.

feature space $\mathbb{R}^{512}$ to parameterize a Gaussian for each class. We then sample from this mixture to generate a new dataset which we use to conduct our sampling-based analysis from Section 3.2.

Results. We show results in Figure 5 and notice that our take-aways from the 2D Gaussian experiments generalize well to neural network feature spaces if the Gaussian assumption holds. If this assumption is violated (noticeably in SVHN, more details in Appendix E.4), then the calibrated tradeoff $\tilde{T}_{\text{cal}}$ still provides a valid SC performance ceiling. However, the distribution of correct acceptances is not guaranteed to be faithful and $\tilde{T}$ therefore overestimates the empirical tradeoff $\hat{T}$.

4.3 Analyzing Performance of Selective Classification Techniques

We can further use our insights to analyze the performance profile of other popular selective classification techniques without relying on sampling approximations. Instead, we derive the empirical survival function of confidences $\hat{\xi}$, as well as the calibrated accuracy function $\hat{\alpha}_{\text{cal}}$, directly from a validation sample with accuracy $\hat{\gamma}$. This allows us to construct a matching performance ceiling $\hat{T}_{\text{cal}}$ (details in Algorithm 2). Again, we train 5 ResNet-18 models across the same datasets as in Section 4.2 but now apply the following SC methods: Softmax Response (SR), Self-Adaptive Training (SAT), Deep Gamblers (DG), Deep Ensembles (DE), and Selective Classification Training Dynamics (SCTD).

Results. We present a subset of our results in Figure 7 (extended in Appendix E.5 and Figure 11). Most notably we find that ensembling approaches, in particular Deep Ensembles, consistently stay close to their calibrated upper bound $\hat{T}_{\text{cal}}$. Based on the survival function of confidences $\hat{\xi}$, it is clear that the confidences for DE are more spread out. This helps the model to mitigate overconfidence. On the other hand, all other approaches fail to close the gap to their individual idealized tradeoffs.

5 Conclusion

In this work we have derived tight guarantees for the accuracy-coverage tradeoff, the central performance metric in selective classification. To that end, we have conducted a rigorous characterization of the Bayes-optimal accuracy-coverage trade-off for 1D Gaussians. Through the use of Monte-Carlo approximations we have further demonstrated how these insights can be applied to arbitrary Gaussian mixtures and neural network feature spaces. Across a wide range of experiments, our analysis has identified the critical role of the survival function of confidences as well as the calibration properties of the employed model. Together, they have lead us to a better understanding of what performance gains we can expect from a selective classifier. Finally, we have analyzed popular SC methods by isolating their empirical coverage and constructing an idealized tradeoff assuming perfect calibration. This factorization allows precise identification of the benefits provided by a particular SC method.

Limitations & Future Work. Our analysis focuses on optimal classifiers that maximize utility. However, other notions of optimality exist under fairness or privacy constraints. Future work should explore the optimal tradeoffs under these guarantees. Moreover, the class-conditional Gaussian assumption can be restrictive; more flexible density estimators like normalizing flows could relax this assumption. Finally, our empirical analysis on neural networks is restricted to residual networks. Extensions of this work should investigate a broader class of model architectures.

References

J. Atchison and S. M. Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67 (2):261–272, 1980.

C. M. Bishop. Pattern recognition and machine learning. *Springer*, 2:1122–1128, 2006.

N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

L. F. P. Cattelan and D. Silva. How to fix a broken confidence estimator: Evaluating post-hoc methods for selective classification with deep neural networks. 2023.

N. Charoenphakdee, Z. Cui, Y. Zhang, and M. Sugiyama. Classification with rejection based on cost-sensitive classification. In *International Conference on Machine Learning*, pages 1507–1517. PMLR, 2021.

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

Y. Ding, J. Liu, J. Xiong, and Y. Shi. Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 4–5, 2020.

R. El-Yaniv et al. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010.

L. Feng, M. O. Ahmed, H. Hajimirsadeghi, and A. H. Abdi. Towards better selective classification. In *The Eleventh International Conference on Learning Representations*, 2023.

A. Fisch, T. Jaakkola, and R. Barzilay. Calibrated selective classification. *arXiv preprint arXiv:2208.12084*, 2022.

I. Galil, M. Dabbah, and R. El-Yaniv. What can we learn from the selective prediction and uncertainty estimation performance of 523 imagenet classifiers. *arXiv preprint arXiv:2302.11874*, 2023.

Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

Y. Geifman and R. El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pages 2151–2159. PMLR, 2019.

Y. Geifman, G. Uziel, and R. El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. *arXiv preprint arXiv:1805.08206*, 2018.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

L. Huang, C. Zhang, and H. Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in neural information processing systems*, 33:19365–19376, 2020.

E. Jones, S. Sagawa, P. W. Koh, A. Kumar, and P. Liang. Selective classification can magnify disparities across groups. *arXiv preprint arXiv:2010.14134*, 2020.

M. H. Kalos and P. A. Whitlock. *Monte carlo methods*. John Wiley & Sons, 2009.

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

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

J. K. Lee, Y. Bu, D. Rajan, P. Sattigeri, R. Panda, S. Das, and G. W. Wornell. Fair selective classification via sufficiency. In *International conference on machine learning*, pages 6076–6086. PMLR, 2021.

K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Z. Liu, Z. Wang, P. P. Liang, R. R. Salakhutdinov, L.-P. Morency, and M. Ueda. Deep gamblers: Learning to abstain with portfolio theory. *Advances in Neural Information Processing Systems*, 32, 2019.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

C. Ni, N. Charoenphakdee, J. Honda, and M. Sugiyama. On the calibration of multiclass classification with rejection. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Pugnana and S. Ruggieri. Auc-based selective classification. In *International Conference on Artificial Intelligence and Statistics*, pages 2494–2514. PMLR, 2023.

S. Rabanser, A. Thudi, K. Hamidieh, A. Dziedzic, and N. Papernot. Selective classification via neural network training dynamics. *arXiv preprint arXiv:2205.13532*, 2022.

S. Rabanser, A. Thudi, A. Thakurta, K. Dvijotham, and N. Papernot. Training private models that know what they don't know. In *Advances in Neural Information Processing Systems*, 2023.

N. Schreuder and E. Chzhen. Classification with abstention but without disparities. In *Uncertainty in Artificial Intelligence*, pages 1227–1236. PMLR, 2021.

M. Tsagris, C. Beneki, and H. Hassani. On the folded normal distribution. *Mathematics*, 2(1):12–28, 2014.

Y. Wiener and R. El-Yaniv. Agnostic selective classification. *Advances in neural information processing systems*, 24, 2011.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A Broader Impact

Past work has identified that modern neural networks are often overconfident in their decision-making [Guo et al., 2017]. This issue, along with other limitations, hampers their use in critical scenarios. Although modern selective classification methods reduce overconfidence, recent studies have shown that these algorithms tend to disproportionately reject samples from minority groups [Jones et al., 2020, Lee et al., 2021]. This introduces a trade-off between improved coverage and fairness, highlighting the need for further exploration of the relationship between fairness and sample rejection. While our work does not address this current limitation of selective classification, it provides tighter performance guarantees. These can be used to identify how effective a selective classifier can be with a given SC method and dataset.

B Definitions

Definition B.1 (Logit-Normal Distribution [Atchison and Shen, 1980])**.** *The logit-normal distribution is a probability distribution for a random variable X whose logit (i.e., the logarithm of the odds) is normally distributed. Specifically, logit$(X) = \log(\frac{X}{1-X})$ follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$. The probability density function of X, defined on the open interval $(0, 1)$, is given by:*

$$f_X^{LN}(x|\mu, \sigma^2) = \frac{1}{x(1-x)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log(\frac{x}{1-x}) - \mu)^2}{2\sigma^2}\right) \quad . \tag{7}$$

We denote the PDF via LogitNormal$(x|\mu, \sigma^2) := f_X(x|\mu, \sigma^2)$. The cumulative distribution function of X is given by:

$$F_X^{LN}(x|\mu, \sigma^2) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\log(\frac{x}{1-x}) - \mu}{\sqrt{2\sigma^2}}\right)\right] \quad . \tag{8}$$

Definition B.2 (Folded Logit-Normal Distribution)**.** *The folded logit-normal distribution (folded at 0.5) is a probability distribution for a random variable Y derived by folding a logit-normal distributed variable X at 0.5, such that $Y = |X - 0.5|$. This transformation creates a new variable that measures the absolute deviation from the midpoint 0.5 of the original X whose logit is normally distributed. The probability density function (PDF) of Y, defined on the interval $[0, 0.5]$, is given by:*

$$f_Y^{FLN}(y|\mu, \sigma^2) = \frac{2}{(0.5+y)(0.5-y)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log(\frac{0.5+y}{0.5-y}) - \mu)^2}{2\sigma^2}\right) \quad . \tag{9}$$

We denote the PDF via FoldedLogitNormal$(y|\mu, \sigma^2) := f_Y(y|\mu, \sigma^2)$. The cumulative distribution function (CDF) of Y is given by:

$$F_Y^{FLN}(y|\mu, \sigma^2) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{\log(\frac{0.5+y}{0.5-y}) - \mu}{\sqrt{2\sigma^2}}\right) - \operatorname{erf}\left(\frac{\log(\frac{0.5-y}{0.5+y}) - \mu}{\sqrt{2\sigma^2}}\right)\right] + \frac{1}{2} \quad . \tag{10}$$

Definition B.3 (Survival Function)**.** *Let X be a continuous random variable with cumulative distribution function (CDF) $F_X(x) = P(X \leq x)$. The survival function $S_X(x)$ is defined as:*

$$S_X(x) = P(X > x) = 1 - F_X(x) \tag{11}$$

Definition B.4 (Multivariate Change of Variables)**.** *Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a vector of random variables, and $\mathbf{Y} = \mathbf{g}(\mathbf{X})$ be a transformation of $\mathbf{X}$, where $\mathbf{g}$ is a vector-valued function $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$. If $\mathbf{g}$ is differentiable and bijective, the probability density function (PDF) of $\mathbf{Y}$ can be obtained using multivariate change of variables*

$$f_{\mathbf{Y}}(\mathbf{y}) = f_{\mathbf{X}}(\mathbf{g}^{-1}(\mathbf{y})) \cdot \left|det\left(\mathbf{J}_{\mathbf{g}^{-1}}(\mathbf{y})\right)\right| \quad , \tag{12}$$

where $\mathbf{J}_{\mathbf{g}^{-1}}$ is the Jacobian matrix of derivatives of $\mathbf{g}^{-1}$.

Definition B.5 (Logistic-Normal Distribution)**.** *The logistic-normal distribution is a probability distribution for a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ where each component X_i is constrained to the open interval $(0, 1)$. The logit-transform of the vector, defined as logit$(\mathbf{X}) =$*

$\left(\log\left(\frac{X_1}{1-X_1}\right), \ldots, \log\left(\frac{X_k}{1-X_k}\right)\right)$*, follows a multivariate normal distribution* $\mathcal{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$*. The probability density function (PDF) of* $\mathbf{X}$*, defined on the open interval* $(0,1)^k$*, is given by:*

$$f_{\mathbf{X}}^{LN}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\prod_{i=1}^{k} \frac{1}{x_i(1-x_i)}\right) \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathit{logit}(\mathbf{x}) - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathit{logit}(\mathbf{x}) - \boldsymbol{\mu})\right) \quad . \tag{13}$$

Definition B.6 (Brier Score)**.** *For a set of* N *predictions across* K *classes, the multi-class Brier score is defined as*

$$BS = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} (\hat{p}_{ik} - o_{ik})^2 \ , \tag{14}$$

where $\hat{p}_{ik}$ *is the predicted probability of instance* i *belonging to class* k*, and* o_{ik} *is the actual outcome, which is* 1 *if* i *belongs to class* k *and* 0 *otherwise. The score ranges from* 0 *(perfect calibration) to* 1 *(poorest calibration).*

Definition B.7 (Perfect Ordering Upper Bound [Rabanser et al., 2023])**.** The upper bound on the selective classification performance for a fixed full-coverage accuracy $a_{\text{full}} \in [0,1]$ and a variable coverage level $c \in [0,1]$ is given by

$$\overline{\text{acc}}(a_{\text{full}}, c) = \begin{cases} 1 & 0 < c \leq a_{\text{full}} \\ \frac{a_{\text{full}}}{c} & a_{\text{full}} < c < 1 \end{cases} . \tag{15}$$

Definition B.8 (Accuracy-Normalized Selective Classification Score (`ANSC`) [Rabanser et al., 2023])**.** The accuracy-normalized selective classification score $s_{a_{\text{full}}}(f, g)$ for a selective classifier (f, g) with full-coverage accuracy a_{full} is given by

$$s_{a_{\text{full}}}(f, g) = \int_0^1 (\overline{\text{acc}}(a_{\text{full}}, c) - \text{acc}_c(f, g)) dc \approx \sum_c (\overline{\text{acc}}(a_{\text{full}}, c) - \text{acc}_c(f, g)). \tag{16}$$

C Additional Proofs and Derivations

C.1 Bayes Optimal Predictor & Accuracy for 1D Gaussians

Fact C.1 (Bayes-Optimal Predictor)**.** *Let* $p(x)$ *be as in Equation 3. Then the Bayes-optimal decision function* $z^*(x)$ *under* $p(x)$ *is* $z^*(x) = w^* x + b^*$ *with* $w^* = \frac{2a}{\sigma_x^2}$ *and* $b^* = 0$*. The Bayes-optimal predictor has accuracy* $\gamma^* = 1 - \Phi(-\frac{a}{\sigma_x}) = \Phi(\frac{a}{\sigma_x})$ *where* $\Phi(\cdot)$ *corresponds to the cumulative distribution function of the standard normal distribution. A proof is provided in Appendix C.1.*

Proof. Recall that the mixture is given by:

$$p(x) = \frac{1}{2}\left(\frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x-a)^2}{2\sigma_x^2}} + \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x+a)^2}{2\sigma_x^2}}\right) \tag{17}$$

The Bayes-optimal decision function $z^*(x)$ minimizes the probability of error under $p(x)$. It is given by the sign of the posterior difference:

$$z^*(x) = \text{sign}(p(y=1|x) - p(y=-1|x)) \tag{18}$$

Using Bayes' theorem $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$ and the fact that our mixture is balanced $p(y=1) = p(y=-1) = \frac{1}{2}$, we get:

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} = \frac{\frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x-a)^2}{2\sigma_x^2}} \cdot \frac{1}{2}}{p(x)} \tag{19}$$

$$p(y=-1|x) = \frac{p(x|y=-1)p(y=-1)}{p(x)} = \frac{\frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x+a)^2}{2\sigma_x^2}} \cdot \frac{1}{2}}{p(x)} \tag{20}$$

The decision function becomes:

$$z^*(x) = \text{sign}\left(\frac{\frac{1}{\sqrt{2\pi\sigma_x^2}}e^{-\frac{(x-a)^2}{2\sigma_x^2}}}{2p(x)} - \frac{\frac{1}{\sqrt{2\pi\sigma_x^2}}e^{-\frac{(x+a)^2}{2\sigma_x^2}}}{2p(x)}\right) \tag{21}$$

$$z^*(x) = \text{sign}\left(e^{-\frac{(x-a)^2}{2\sigma_x^2}} - e^{-\frac{(x+a)^2}{2\sigma_x^2}}\right) \tag{22}$$

Since the exponentials are the primary terms, the argument of the exponentials determines the sign:

$$z^*(x) = \text{sign}\left(-\frac{(x-a)^2}{2\sigma_x^2} + \frac{(x+a)^2}{2\sigma_x^2}\right) \tag{23}$$

We further simplify the expression inside the sign function:

$$-\frac{(x-a)^2}{2\sigma_x^2} + \frac{(x+a)^2}{2\sigma_x^2} = -\frac{x^2-2ax+a^2}{2\sigma_x^2} + \frac{x^2+2ax+a^2}{2\sigma_x^2} = \frac{4ax}{2\sigma_x^2} = \frac{2ax}{\sigma_x^2} \tag{24}$$

Thus, the decision function reduces to:

$$z^*(x) = \text{sign}\left(\frac{2ax}{\sigma_x^2}\right) = \text{sign}\,(x) \quad \text{since } \frac{2a}{\sigma_x^2} > 0 \tag{25}$$

From the decision function $z^*(x) = \text{sign}(w^*x + b^*)$ we identify:

$$w^* = \frac{2a}{\sigma_x^2}, \quad b^* = 0 \tag{26}$$

The accuracy γ^* is the probability that the decision function correctly classifies x, i.e. $\gamma^* = P(z^*(x) = y)$. Given $y = 1$ implies $x \sim \mathcal{N}(a, \sigma_x^2)$ and $y = -1$ implies $x \sim \mathcal{N}(-a, \sigma_x^2)$, the probability of correct classification is:

$$\gamma^* = P(x > 0|y = 1) \cdot P(y = 1) + P(x < 0|y = -1) \cdot P(y = -1) \tag{27}$$

Since $P(y = 1) = P(y = -1) = \frac{1}{2}$:

$$\gamma^* = \frac{1}{2}P(x > 0|x \sim \mathcal{N}(a, \sigma_x^2)) + \frac{1}{2}P(x < 0|x \sim \mathcal{N}(-a, \sigma_x^2)) \tag{28}$$

For $x \sim \mathcal{N}(a, \sigma_x^2)$:

$$P(x > 0) = 1 - \Phi\left(\frac{0-a}{\sigma_x}\right) = 1 - \Phi\left(-\frac{a}{\sigma_x}\right) = \Phi\left(\frac{a}{\sigma_x}\right) \tag{29}$$

For $x \sim \mathcal{N}(-a, \sigma_x^2)$:

$$P(x < 0) = \Phi\left(\frac{0+a}{\sigma_x}\right) = \Phi\left(\frac{a}{\sigma_x}\right) \tag{30}$$

Therefore:

$$\gamma^* = \frac{1}{2}\Phi\left(\frac{a}{\sigma_x}\right) + \frac{1}{2}\Phi\left(\frac{a}{\sigma_x}\right) = \Phi\left(\frac{a}{\sigma_x}\right) \tag{31}$$

□

C.2 Bayes-Optimal Calibrated Accuracy

Proposition C.2 (Bayes-Optimal Calibrated Accuracy Function α^*_{cal})**.** *Assume Proposition 3.1 holds and that the confidence distribution $p(c)$ is perfectly calibrated. Then the Bayes-optimal accuracy function is $\alpha^*_{cal}(c) := 2((1-\gamma^*)c + \gamma^* - 0.5)$ for accuracy γ^*. See Appendix C.2 for a proof and Figure 3 (a) for an example.*

Algorithm 1: Approximation of accuracy-coverage tradeoff with sampling.

Require: Gaussian mixture estimated in input or feature space $\{(\tilde{\pi}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K$, number of samples used for approximation N, confidence level c.

$\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{y}}^{(1)} \leftarrow$ `sample_from_gaussian_mixture`$(\{(\tilde{\pi}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K, N)$

$\tilde{\mathbf{X}}^{(2)}, \tilde{\mathbf{y}}^{(2)} \leftarrow$ `sample_from_gaussian_mixture`$(\{(\tilde{\pi}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K, N)$

$(\tilde{\mathbf{W}}, \tilde{\mathbf{b}}), \tilde{\gamma} \leftarrow$ `fit_logistic_regression_model`$(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{y}}^{(1)})$

$c(\tilde{\mathbf{X}}^{(2)}) \leftarrow \max_k \boldsymbol{\sigma}(\langle \tilde{\mathbf{W}}, \tilde{\mathbf{X}}^{(2)} \rangle + \tilde{\mathbf{b}})$ {Compute maximum confidences}

$\hat{\mathbf{y}} \leftarrow \arg\max_k \boldsymbol{\sigma}(\langle \tilde{\mathbf{W}}, \tilde{\mathbf{X}}^{(2)} \rangle + \tilde{\mathbf{b}})$ {Compute predictions}

$\tilde{c} \sim \frac{1}{N} \sum_{n=1}^N \delta(c - c(\tilde{\mathbf{X}}^{(2)}))$ {Compute distribution of all max confidences}

$\tilde{c}_+ \sim \frac{1}{N} \sum_{n=1}^N \delta(c - c(\tilde{\mathbf{X}}^{(2)})) \mathbb{1}[\tilde{\mathbf{y}} = \hat{\mathbf{y}}]$ {Compute distribution of correct max confidences}

$\tilde{\xi}(c) \leftarrow 1 - \hat{F}_{\tilde{c}}(c)$ {Compute overall survival}

$\hat{S}_+(c) \leftarrow 1 - \hat{F}_{\tilde{c}_+}(c)$ {Compute correct survival}

$\tilde{\alpha}(c) \leftarrow \tilde{\gamma} \hat{S}_+(c) \tilde{\xi}^{-1}(c)$ {Compute accuracy}

$\tilde{\alpha}_{\text{cal}}(c) \leftarrow 2((1 - \tilde{\gamma})c + \tilde{\gamma} - 0.5)$ {Compute calibrated accuracy}

return $\tilde{T} \leftarrow (\tilde{\xi}, \tilde{\alpha}),\ \tilde{T}_{\text{cal}} \leftarrow (\tilde{\xi}, \tilde{\alpha}_{\text{cal}})$

Algorithm 2: Calibrated accuracy-coverage tradeoff from validation sample.

Require: Train sample $(\hat{\mathbf{X}}^{(\text{train})}, \hat{\mathbf{y}}^{(\text{train})})$, Validation sample $(\hat{\mathbf{X}}^{(\text{val})}, \hat{\mathbf{y}}^{(\text{val})})$, SC method $m \in \{\texttt{SR}, \texttt{DG}, \texttt{SAT}, \texttt{DE}, \texttt{SCTD}\}$, confidence level c.

$(h, g) \leftarrow$ `fit_sc_method`$(\hat{\mathbf{X}}^{(\text{train})}, \hat{\mathbf{y}}^{(\text{train})}, m)$

$\hat{\gamma} \leftarrow \mathbb{1}[h(\hat{\mathbf{X}}^{(\text{val})}) = \hat{\mathbf{y}}^{(\text{val})}]$ {Get accuracy on validation set}

$c(\hat{\mathbf{X}}^{(\text{val})}) \leftarrow g(\tilde{\mathbf{X}}^{(\text{val})}, h)$ {Get scores from SC method}

$\hat{c} \sim \frac{1}{N} \sum_{n=1}^N \delta(c - c(\tilde{\mathbf{X}}^{(\text{val})}))$ {Compute distribution of confidences}

$\hat{\xi}(c) \leftarrow 1 - \hat{F}_{\hat{c}}(c)$ {Compute empirical survival}

$\hat{\alpha}_{\text{cal}}(c) \leftarrow 2((1 - \hat{\gamma})c + \hat{\gamma} - 0.5)$ {Compute calibrated accuracy}

return $\hat{T}_{\text{cal}} \leftarrow (\hat{\xi}, \hat{\alpha}_{\text{cal}})$

Proof. We approach this proof by first stating two known points of the Bayes-optimal accuracy function $\alpha^*_{\text{cal}}(c)$ and then linearly interpolating between these points. Recall that the Bayes-optimal accuracy α is the accuracy over the full data distribution, i.e. at full coverage. Hence, $\alpha^*_{\text{cal}}(c)$ needs to pass through $(0.5, \alpha)$ where 0.5 confidence corresponds to the full coverage setting. Moreover, an optimal selective classifier with no coverage is per definition perfectly accurate. Therefore, as the confidence $c \to 1$, the coverage goes to 0 but the selective accuracy goes to 1. This gives our second known point $(1, 1)$. Under the assumption of perfect calibration (Equation 5), i.e. a linear increase in the confidence corresponds to a linear increase in accuracy, $\alpha^*_{\text{cal}}(c)$ needs to interpolate linearly between these endpoints. Finding the linear function that interpolates between $(0.5, \alpha)$ and $(1, 1)$ yields the optimal accuracy function under perfect calibration: $\alpha^*_{\text{cal}}(c) = 2((1 - \alpha)c + \alpha - 0.5)$. □

D Difficulty of Distributional Analysis of Generalized Gaussian Mixture

Extended Model Assumption. We can generalize binary logistic regression via two distinct approaches: In (i) *One-vs-Rest (OvR) Classification* we train multiple binary logistic regression models, each predicting the probability of one class versus all others; while in (ii) *Multinomial/Softmax Classification* a single model predicts probabilities directly for all classes using the softmax function. Both methods predict the probability of a particular class k as follows:

$$p_{\text{OvR}}(y = k \mid \mathbf{z}(\mathbf{x})) = \sigma(\mathbf{z}_k(\mathbf{x})) = \frac{1}{1 + e^{-\mathbf{z}_k(\mathbf{x})}} \qquad p_{\text{Softmax}}(y = k \mid \mathbf{z}(\mathbf{x})) = \boldsymbol{\sigma}(\mathbf{z})_k = \frac{e^{\mathbf{z}_k(\mathbf{x})}}{\sum_{j=1}^K e^{\mathbf{z}_j(\mathbf{x})}} \quad (32)$$

where $\mathbf{z}_k(\mathbf{x}) = \langle \mathbf{w}_k, \mathbf{x} \rangle + b_k$ corresponds to the logit at index k and $\sigma(\cdot)$ and $\boldsymbol{\sigma}(\cdot)$ correspond to the sigmoid and softmax functions, respectively. Across both methods, the final class assignment is

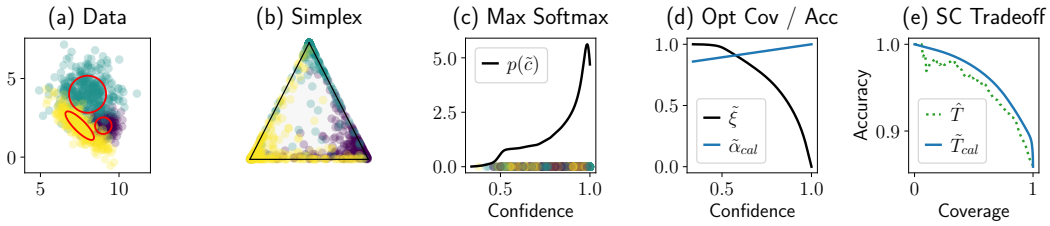

Figure 8: **Tradeoff estimation for 2D Gaussians**. Similar as Figures 2 and 3.

based on the class that gives the highest probability $\hat{y}(\mathbf{x})$ with confidence $c(\mathbf{x})$ defined as follows:

$$\hat{y}(\mathbf{x}) = \arg\max_k p(y = k \mid \mathbf{z}(\mathbf{x})) \qquad c(\mathbf{x}) = \max_k p(y = k \mid \mathbf{z}(\mathbf{x})) \tag{33}$$

In a similar fashion as with Proposition 3.2, the first step in our analysis is to derive the distribution of the optimal confidence function $c^*(\cdot)$. Unfortunately, given the distributional and modeling assumptions above, it is no longer possible to determine the ideal $c^*(\cdot)$ analytically.

Problem 1: Unknown optimal decision function The Bayes-optimal classifier for a set of Gaussians with arbitrary means, covariances and mixture weights can be derived using Quadratic Discriminant Analysis (QDA) which yields non-linear decision boundaries between classes. However, our logistic regression assumption, which we introduced due to its widespread applicability (including on top of neural network feature spaces), requires a linear decision boundary. Although a linear boundary would be yielded by Linear Discriminant Analysis (LDA), this approach assumes shared covariances or an approximation via a weighted average of the covariances (pooling). Independent of whether an approximation is used, LDA and logistic regression generally do not yield the same solution.

Problem 2: Non-invertibility of softmax While the softmax classification setup is more widely used, only the OvR setting can be effectively analyzed from a distributional standpoint. To see why this is the case, we need to revisit multivariate change of variables.

While the sigmoid from the OvR setting is invertible via the logit function $\mathbf{z}_k = \log(\frac{\sigma(\mathbf{z}_k)}{1-\sigma(\mathbf{z}_k)})$, the softmax function is not invertible since we cannot isolate $\mathbf{z}$:

$$\boldsymbol{\sigma}(\mathbf{z})_k = \frac{e^{\mathbf{z}_k}}{\sum_{j=1}^K e^{\mathbf{z}_j}} \implies \mathbf{z}_k = \log(\boldsymbol{\sigma}(\mathbf{z})_k) + \log\left(\sum_{j=1}^K e^{\mathbf{z}_j}\right) = \log(\boldsymbol{\sigma}(\mathbf{z})_k) + \text{const}(\mathbf{z}) \tag{34}$$

This seems intuitive as the softmax function defines a surjective mapping where multiple distinct logit vectors $\mathbf{z}^{(i)}, \mathbf{z}^{(j)}$ can map to the same softmax vector $\boldsymbol{\sigma}(\mathbf{z}^{(i)}) = \boldsymbol{\sigma}(\mathbf{z}^{(j)})$.

Problem 3: Unknown Distribution of Maximum over Logistic Normal While the distribution after element-wise sigmoidal transformation is known, the distribution of the maximum within each logistic-normal is not (no closed formulation exists for an arbitrary number of dimensions) and by extension the maximum over the full mixture cannot be determined. In fact, no closed solution even exists for the maximum of a multivariate Gaussian.

E Extension of Empirical Results

E.1 Hyper-parameters

General hyper-parameters. Training on real-life datasets was done using stochastic gradient descent with a batch size 128, momentum 0.9 and weight decay 0.0005. Learning rates were were adaptively chosen from range $[0.001, 0.01]$. A multi-step learning rate scheduler was used with a schedule interval of 25 epochs and a $\gamma = 0.75$.

Table 1: **Hyper-parameters used for all selective classification methods.**

Dataset	SC Algorithm	Hyper-Parameters
CIFAR-10	Softmax Response (SR)	N/A
	Self-Adaptive Training (SAT)	$P = 100$
	Deep Gamblers (DG)	$P = 100$
	Deep Ensembles (DE)	$E = 5$
	Selective Classification Training Dynamics (SCTD)	$T = 1600,\ k = 3$
CIFAR-100	Softmax Response (SR)	N/A
	Self-Adaptive Training (SAT)	$P = 100$
	Deep Gamblers (DG)	$P = 100$
	Deep Ensembles (DE)	$E = 5$
	Selective Classification Training Dynamics (SCTD)	$T = 1600,\ k = 3$
GTSRB	Softmax Response (SR)	N/A
	Self-Adaptive Training (SAT)	$P = 100$
	Deep Gamblers (DG)	$P = 100$
	Deep Ensembles (DE)	$E = 5$
	Selective Classification Training Dynamics (SCTD)	$T = 800,\ k = 3$
SVHN	Softmax Response (SR)	N/A
	Self-Adaptive Training (SAT)	$P = 100$
	Deep Gamblers (DG)	$P = 100$
	Deep Ensembles (DE)	$E = 5$
	Selective Classification Training Dynamics (SCTD)	$T = 2200,\ k = 3$

2D Gaussian experiments. Our experiments use the following datasets:

1. $\mathcal{D}_1$:
$$\left\{\left(\frac{1}{3},\ \begin{bmatrix}9\\2.5\end{bmatrix},\ 0.2\mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}8\\4\end{bmatrix},\ \mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}6\\2\end{bmatrix},\ \begin{bmatrix}0.6 & -0.5\\-0.5 & 0.6\end{bmatrix}\right)\right\} \tag{35}$$
2. $\mathcal{D}_2$:
$$\left\{\left(\frac{1}{3},\ \begin{bmatrix}-1\\2\end{bmatrix},\ 0.2\mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}1\\2\end{bmatrix},\ \mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}0\\-0.5\end{bmatrix},\ \begin{bmatrix}0.6 & -0.5\\-0.5 & 0.6\end{bmatrix}\right)\right\} \tag{36}$$
3. $\mathcal{D}_3$:
$$\left\{\left(\frac{1}{3},\ \begin{bmatrix}9\\2\end{bmatrix},\ 0.2\mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}8\\4\end{bmatrix},\ \mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}8.75\\2.5\end{bmatrix},\ \begin{bmatrix}0.6 & -0.5\\-0.5 & 0.6\end{bmatrix}\right)\right\} \tag{37}$$
4. $\mathcal{D}_4$: For $a = 0.2$:
$$\left\{\left(\frac{1}{3},\ \begin{bmatrix}-a\\a\end{bmatrix},\ \mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}a\\a\end{bmatrix},\ \mathbf{I}_2\right),\ \left(\frac{1}{3},\ \begin{bmatrix}0\\-a\end{bmatrix},\ \mathbf{I}_2\right)\right\} \tag{38}$$

Monte Carlo Sampling. We use a sampling resolution of $N = 10000$ across all experiments.

Other selective classification methods. We document full hyper-parameter settings for all selective classification methods in Table 1. Based on recent insights from Feng et al. [2023], we train SAT and DG with additional entropy regularization with $\beta = 0.01$.

E.2 Compute Resources

Synthetic experiments were conducted on an Apple MacBook Pro with a M1 Pro chip and 32GB of RAM. Individual experiments lasted < 1 minute. Non-synthetic experiments were conducted on a mix of 2 types of machines: (i) Machine Type I: CPU Intel Xeon Silver 4210 with 128GB RAM and GPU NVIDIA RTX 2080Ti (11GB VRAM); or (ii) Machine Type II: CPU AMD EPYC 7643 with 512GB RAM and GPU NVIDIA A100 (80GB VRAM). Individual Experiments lasted < 20 minutes.

E.3 Confidence Modeling Using Beta Distributions

To show the dependence of the accuracy-coverage tradeoff on the employed confidence distribution, we model the confidence distribution using a Beta distribution. We present results in Figure 10 where we confirm that the confidence distribution and its corresponding survival function determines the shape of the tradeoff.

E.4 Verifying the Gaussian Assumption

To verify whether the class-conditional Gaussian assumption from Section 3.2 is reasonable, we perform principal components analysis (PCA) to project the data from the 512-dimensional feature space of the ResNet down to 2 dimensions for visualization. We show the result of this projection in Figure 12. It is apparent that the assumption is reasonable across almost all datasets we consider. SVHN is a notable exception as it shows a lot of pronounced outliers. These outliers can be explained by the fact that SVHN digits often show more than one digit even though the ground truth label is just a single label. This causes the embedding space to show less disentaglement across classes.

E.5 Evaluation Metrics for Selective Classification

We provide an extended evaluation on various evaluation metrics for selective classification.

- The area under the accuracy-coverage curve (`AUACC`) as discussed in Geifman et al. [2018].
- The area under the receiver operating characteristic (`AUROC`) as suggested by Galil et al. [2023].
- The accuracy normalized selective classification score (`ANSC`) from Geifman et al. [2018] and Rabanser et al. [2023].
- The area between the calibrated accuracy $\hat{\alpha}_{\text{cal}}$ and the empirical accuracy $\hat{\alpha}$ as suggested in this work: $\Delta_{\hat{\alpha}} = \int_c |\hat{\alpha}_{\text{cal}} - \hat{\alpha}| dc$

We document results in Table 2. Our proposed evaluation metric allows us to quantify the optimality of any given selective classification method. If $\hat{\alpha}_{\text{cal}}$ is 0 (such as in GTSRB), then no improvement is possible and the selective classifier is optimal. We note that `DE` in particular often comes close to delivering optimal performance on other data sets as well.

Table 2: **Evaluation of SC approaches using various evaluation metrics**.

Dataset	Method	$1 -$ `AUACC`	`ANSC`	`AUROC`	$\Delta_{\hat{\alpha}}$
CIFAR10	`SR`	0.053 ± 0.002	0.007 ± 0.000	0.918 ± 0.002	0.007 ± 0.000
	`SCTD`	0.056 ± 0.001	0.004 ± 0.000	0.938 ± 0.002	0.005 ± 0.000
	`DE`	0.046 ± 0.002	0.004 ± 0.000	0.939 ± 0.003	0.004 ± 0.000
	`SAT`	0.054 ± 0.002	0.006 ± 0.000	0.924 ± 0.005	0.006 ± 0.000
	`DG`	0.054 ± 0.001	0.006 ± 0.000	0.922 ± 0.005	0.007 ± 0.000
CIFAR100	`SR`	0.181 ± 0.001	0.041 ± 0.001	0.865 ± 0.003	0.026 ± 0.000
	`SCTD`	0.184 ± 0.002	0.037 ± 0.000	0.872 ± 0.002	0.022 ± 0.000
	`DE`	0.159 ± 0.001	0.030 ± 0.001	0.880 ± 0.003	0.012 ± 0.000
	`SAT`	0.180 ± 0.001	0.041 ± 0.001	0.866 ± 0.003	0.023 ± 0.000
	`DG`	0.182 ± 0.001	0.041 ± 0.001	0.867 ± 0.002	0.026 ± 0.000
GTSRB	`SR`	0.020 ± 0.002	0.001 ± 0.000	0.986 ± 0.003	0.000 ± 0.000
	`SCTD`	0.019 ± 0.002	0.001 ± 0.000	0.986 ± 0.005	0.000 ± 0.000
	`DE`	0.015 ± 0.001	0.001 ± 0.000	0.986 ± 0.002	0.000 ± 0.000
	`SAT`	0.027 ± 0.001	0.001 ± 0.000	0.984 ± 0.002	0.000 ± 0.000
	`DG`	0.019 ± 0.003	0.001 ± 0.000	0.986 ± 0.002	0.000 ± 0.000
SVHN	`SR`	0.027 ± 0.000	0.006 ± 0.001	0.895 ± 0.004	0.007 ± 0.000
	`SCTD`	0.029 ± 0.003	0.003 ± 0.001	0.932 ± 0.005	0.004 ± 0.000
	`DE`	0.021 ± 0.001	0.005 ± 0.000	0.912 ± 0.003	0.005 ± 0.000
	`SAT`	0.028 ± 0.001	0.006 ± 0.000	0.895 ± 0.002	0.007 ± 0.000
	`DG`	0.026 ± 0.001	0.007 ± 0.000	0.896 ± 0.006	0.007 ± 0.000

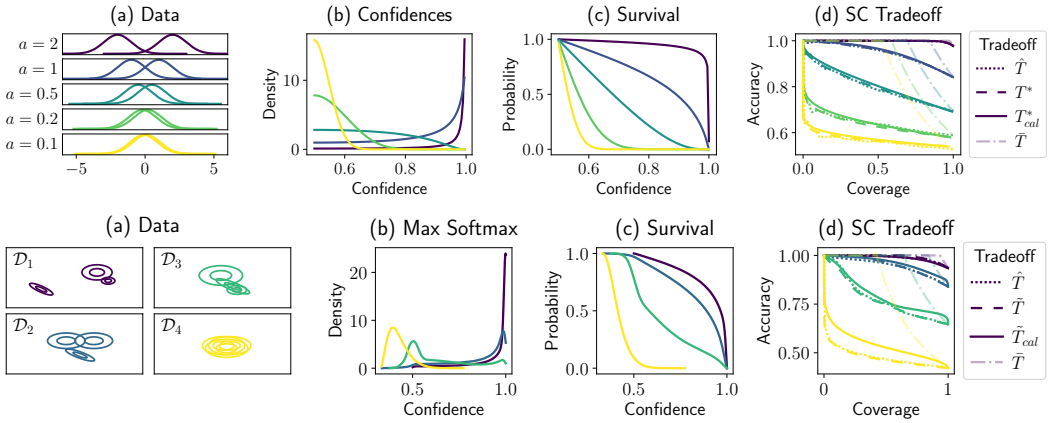

Figure 9: **Synthetic Gaussian Experiments**: Similar as Figure 4 but we show the confidences leading to the coverage function.

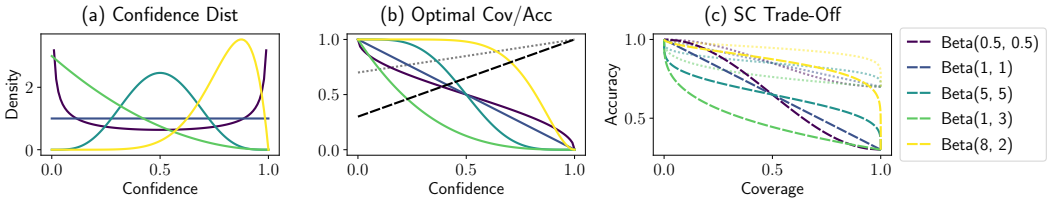

Figure 10: **Varying accuracy-coverage tradeoffs stemming from different confidence distributions under perfect calibration**. The shape of the tradeoff is fully determined by coverage.

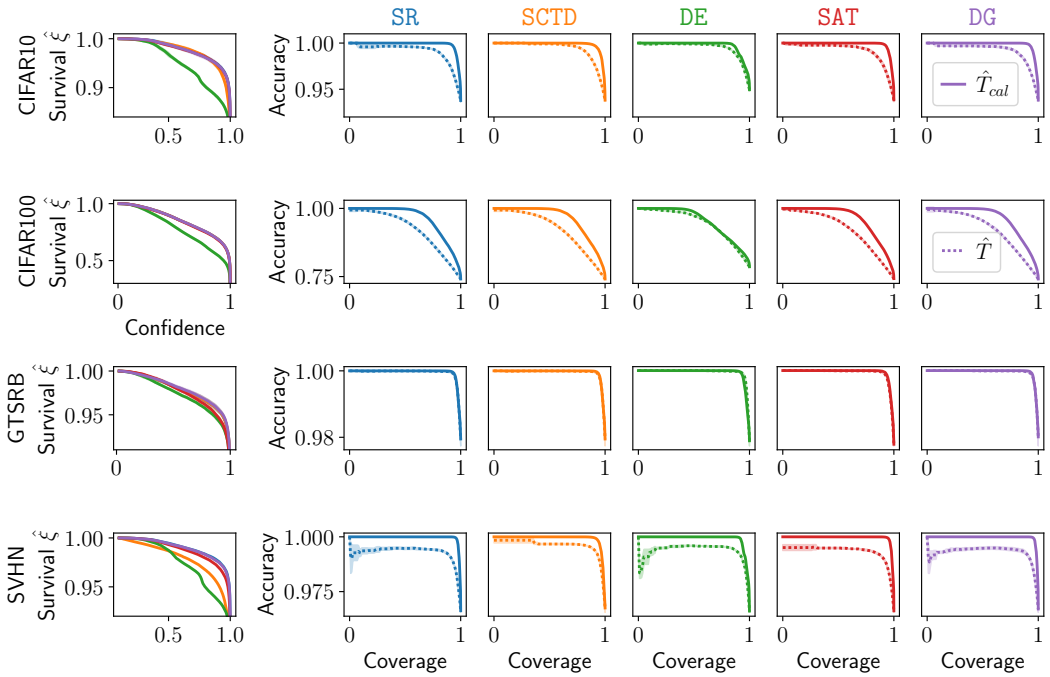

Figure 11: **Extended results from Figure 7**.

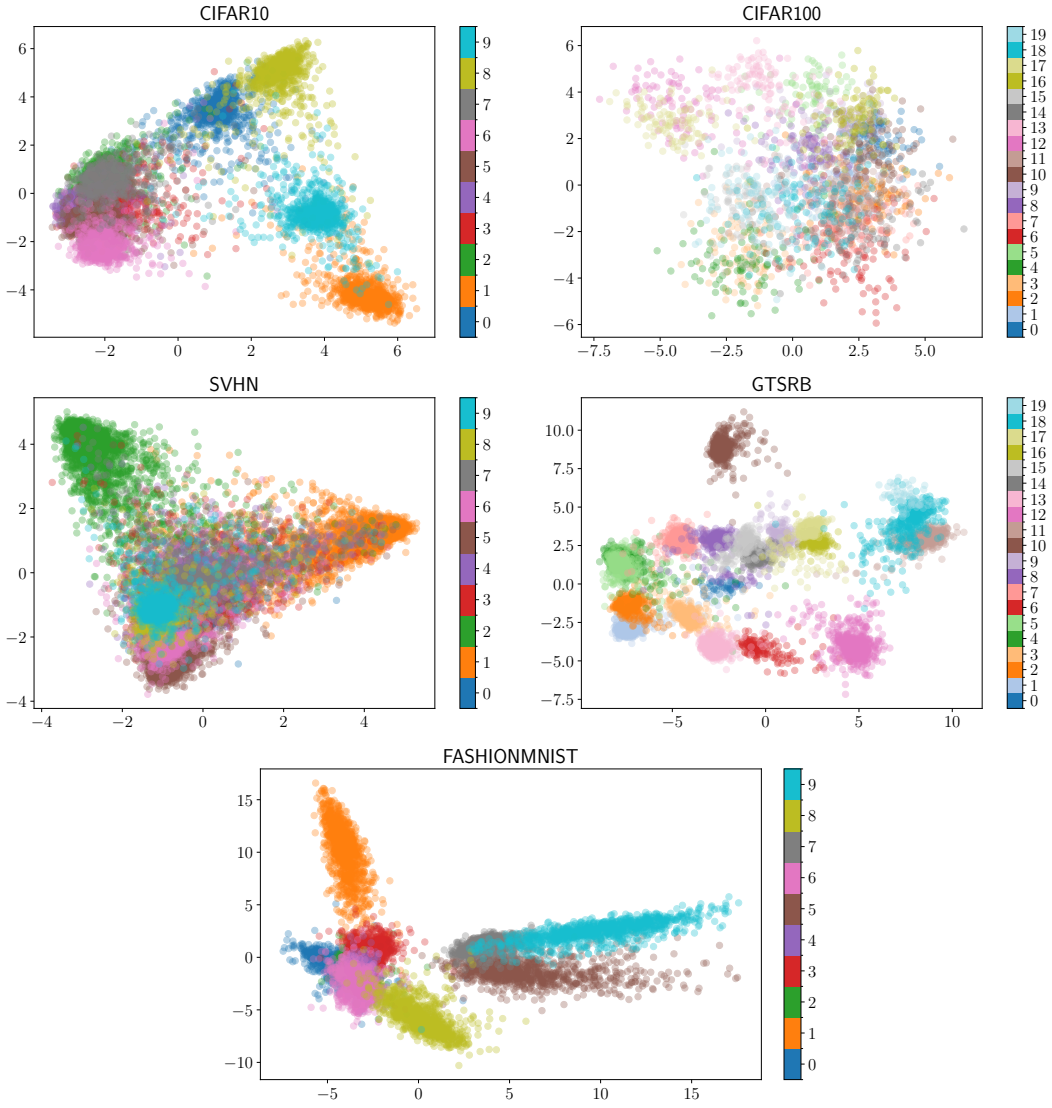

Figure 12: **2-dimensional PCA of feature spaces across datasets**: For GTSRB and CIFAR-100 we only plot the first 20 classes for visibility. We see that the class-Gaussian assumption is reasonable across datasets. SVHN is a noticeable exception due to digit overlap present in the dataset.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Contributions are clearly stated in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a limitations paragraph in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We include proofs in the main body and in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We document hyper-parameters in Appendix E.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code included in supplementary material and will be published to GitHub.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We document hyper-parameters in Appendix E.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide 1-σ error bars whenever appropriate. Our results are based on 5 random runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We document our compute resources in Appendix E.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the code of conduct and are confident that the research outlined in this paper conforms to the code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release such models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our evaluation is either based on synthetic data or standard datasets from the selective classification literature. While we do not include license information, we do cite the corresponding papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: See attached supplementary material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.