CALIBRATION FOR DECISION MAKING VIA EMPIRICAL RISK MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks for classification can achieve high accuracy but their probabilistic predictions may be not well-calibrated, in particular overconfident. Different general calibration measures and methods were proposed. But how exactly does the calibration affect downstream tasks? We derive a new task-specific definition of calibration for the problem of statistical decision making with a known cost matrix. We then show that so-defined calibration can be theoretically rigorously improved by minimizing the empirical risk in the adjustment parameters like temperature. For the empirical risk minimization, which is not differentiable, we propose improvements to and analysis of the direct loss minimization approach. Our experiments indicate that task-specific calibration can perform better than a generic one. But we also carefully investigate weaknesses of the proposed tool and issues in the statistical evaluation for problems with highly unbalanced decision costs.

1 INTRODUCTION

The notion of calibration originates in forecasting, in particular in meteorology. The following example well explains the concept. Amongst all days when a forecast was made that the chance of rain is 33%, one would expect to find that about a third of them to be rainy and two thirds sunny. If this is the case for all forecasts, the predictor is said to be well-calibrated. We would like the forecaster to be accurate, however if this is not achievable, we would like at least to have it accurately reflect what it does not know, *i.e*. to be calibrated.

The most common model for classification in deep learning consists of a softmax predictive distribution for class labels atop of a deep architecture processing the observation. In view of the excessive number of parameters there is a natural concern whether such models learn accurate predictive probabilities $p(y|x)$. Indeed, neural networks are typically not well calibrated (Guo et al., 2017), in particular they may be over-confident, *i.e*., being incorrect much more often than their high confidence suggests. Like in the example above, such over-confidence is misleading for interpreting the results. It is also commonly understood that it can mislead any downstream processing relying on the predictive probabilities. There has been therefore a substantial effort to improve the calibration. Unfortunately, even measuring the basic confidence calibration accurately in practice remains challenging (Nixon et al., 2019).

In the multi-class setting a vector of class probabilities is output and all of them can be important for the downstream processing. This has led to development of more complex definitions such as distribution-calibration (Vaicenavicius et al., 2019). In this setting, despite the development of new estimators (Vaicenavicius et al., 2019; Widmann et al., 2019), it is practically infeasible to obtain a reliable estimate — there is not enough data to reject the hypothesis that the model is well calibrated. Are we lost then in the attempt to make the predictive probabilities reliable? Not necessarily. Observe that these calibration definitions, both simple and complex ones, are not considering a specific problem downstream that can be hypothetically affected by the poor probabilistic predictions. They try to address all such problems (as well as the purpose of interpretability) at once. We will argue that considering a specific downstream task allows to significantly reduce the complexity of the calibration problem, making it feasible in practice.

As a specific downstream task we will consider the Bayesian decision making with a trained NN. It is well-established to train NNs for classification by optimizing the cross-entropy loss. The model

architecture and the training pipeline are tuned by researchers to achieve the best generalization w.r.t. classification accuracy. However, this procedure does not take into account different costs of different classification mistakes. To give an example, mistakes in miss-classifying different mushrooms may have different costs for eating: some mistakes are between equally good spices and incur no cost while other mistakes lead to risks of poisoning. Given a finite set of decisions D and a cost matrix $l(y, d)$, one could try to adapt the learned NN model to form the Bayesian decisions strategy $q(x)$ achieving the smallest risk:

$$q(x) = \arg\min_d \sum_y p(y|x)l(y, d). \tag{1}$$

Such adaptation is practically desirable: it would allow one to rely on the established and tuned training approach and reuse the existing models, which may be very costly to retrain from scratch. However, because the model $p(y|x)$ is typically inaccurate in predicting all the class probabilities, this may lead to suboptimal decisions and respectively poor outcomes of such adaptation. In particular, an overconfident model will result in both: making sub-optimal decisions and underestimating their expected cost.

One could reasonably hope to learn a more accurate predictor by designing a yet better architecture and using more training data. It may nevertheless stay poorly calibrated and not suitable for the above adaptation. If the strong distribution calibration was possible to achieve as a post-processing step, the adaptation would work perfectly, however this calibration is practically not feasible. On the other hand, any weaker task-unspecific definition of calibration (e.g. class-wise calibration, Vaicenavicius et al. 2019) would not work: there would exist a decision task for which (1) would perform poorly. A formalization of calibration important for (a class of) decision making problems was proposed, only recently, by Zhao et al. (2021).

As our main theoretical result, we derive a notion of calibration important for adopting strategy (1) with a given cost matrix. The formalism and tools used for that allow also to better understand the relation between existing calibration methods, in particular empirically successful temperature scaling variants (Guo et al., 2017; Alexandari et al., 2020), and the distribution calibration. Specifically, we show that these methods are guaranteed to improve the expected miscalibration as measured by the corresponding divergence. Calibrating the model w.r.t. the new definition is shown to be equivalent to minimizing the (empirical) risk of strategy (1) in the calibration parameters. As a mean of empirical risk minimization we study the direct loss approach and relate it to margin-rescaling. Experimentally, we show that task-aware calibration, using the direct loss approach, can outperform the generic calibration. However we also observe that for tasks with extremely unbalanced losses, modeling a dangerous class, we lack reliable means to assess the quality of calibration.

2 Related Work

We consider all methods that can improve the predictive model $p(y|x)$ when provided some additional calibration data as *calibration methods*. Typically, the calibration is achieved by a post-processing of the scores or predictive probabilities such as temperature scaling, bias-corrected temperature scaling or vector scaling. We regard these as different choices of *parametrization*, *i.e.*, choice of the degrees of freedom to calibrate. Most importantly, existing calibration methods differ in the criterion they optimize.

Calibration Unaware of the Task Many calibration techniques, while motivated by the notion and a particular definition of calibration, use a generic criterion unrelated to that notion. A very practical method to calibrate a model turns out to be the likelihood maximization, *i.e.* relying on the same criterion that is commonly used for training. This is the approach taken in Guo et al. (2017); Alexandari et al. (2020); Kull et al. (2019). Methods optimizing variants of the expected calibration error (ECE), which is a measure of miscalibration, were compared by Nixon et al. (2019), however there is no performance criterion other than ECE itself. Further variants are piece-wise parametric (Kumar et al., 2019) and kernel-based (Kumar et al., 2018) confidence calibration methods.

Calibration Aware of the Task The decision-calibration of Zhao et al. (2021) takes the decision problem into the consideration. It will be discussed in detail below. Their calibration method is

designed for a given decision space, but cannot make use of a specific cost matrix was it known at the calibration time.

Empirical Risk Minimization Methods optimizing the empirical risk (Song et al., 2016; Vlastelica et al., 2019; Taskar et al., 2005) were used for training with complex objectives measuring performance in retrieval, ranking, or structured prediction. They have not been considered for calibration of classification models before. They address the difficulty of non-differentiability of the loss and have a potential to exploit the full information about the cost matrix.

3 Background

Let $\mathcal{X}$ be the space of observations and $\mathcal{Y}$ be the set of labels. Assume there is an underlying true joint probability distribution on $\mathcal{X} \times \mathcal{Y}$, denoted p^*. Let (X, Y) be a a pair of random variables with the law p^*. All expectations and probabilities will be meant with respect to (X, Y). Let Δ denote the simplex of probabilities over $\mathcal{Y}$. Let $p(y|x)$ be a probabilistic predictor, usually a neural network with softmax output. The predictor can be considered as a mapping $\pi\colon \mathcal{X} \to \Delta\colon x \mapsto p(y|x)$.

3.1 General Calibration

First works analyzing calibration in machine learning (Guo et al., 2017) were concerned only with the *confidence* of the model, *i.e.* the model's probability of the class actually predicted. Let $\hat{y}(x) = \arg\max_k \pi_k(x)$ be the label predicted by the model and $c(X) = \max_k \pi_k(X)$ the respective predictive probability, called *confidence*.

Definition 1. The model is *confidence calibrated* if

$$\mathbb{P}\big(Y{=}\hat{y}(X) \mid c(X)\big) \overset{\text{a.s.}}{=} c(X). \tag{2}$$

It requires that amongst all data points for which the prediction has confidence c the expected occurrence of the true label to match c. The respective miscalibration can be measured *e.g.*, by the Expected Calibration Error (ECE) (Degroot & Fienberg, 1983), which is typically estimated by discretizing the probability interval into bins. Substantial efforts were put into calibrating neural networks in this sense (*e.g.*, Naeini et al. 2015; Guo et al. 2017; Nixon et al. 2019). However, Kumar et al. (2019) argue that the binning underestimates the calibration error and in fact, an accurate estimation is possible only when the predictor π outputs only a discrete set of values.

In the multi-class setting, confidence calibration may be insufficient. There may be downstream tasks which require the whole vector of predicted probabilities to be accurate. In the machine learning literature this came into attention only recently (Vaicenavicius et al., 2019). The strongest notion of calibration (Bröcker 2009, reliability Eq. 1), is as follows.

Definition 2. A predictor $\pi\colon \mathcal{X} \to \Delta$, is called *distribution calibrated* if

$$(\forall y \in \mathcal{Y}) \quad \mathbb{P}\big(Y{=}y \mid \pi(X)\big) \overset{\text{a.s.}}{=} \pi(X)_y. \tag{3}$$

In words: amongst all data points of the input space where the predicted vector of probabilities is $\pi(x) = \mu$ the true observed class labels should be distributed as μ. Respectively, the predictor

$$\phi[\pi](x)_y = \mathbb{P}\big(Y{=}y \mid \pi(X){=}\pi(x)\big) \tag{4}$$

is the (optimal) *calibration* of π: it takes the prediction $\pi(x)$ and turns it into the true distribution of labels under that initial prediction. This predictor $\phi[\pi]$ is distribution calibrated and Definition 2 can be restated as $\phi[\pi](X) \overset{\text{a.s.}}{=} \pi(X)$, *i.e.* the calibration of π is π itself.

Generalizing on ECE, the *expected miscalibration* of π w.r.t. divergence $D\colon \Delta \times \Delta \to \mathbb{R}$ is:

$$\mathbb{E}[D(\pi(X), \phi[\pi](X))], \tag{5}$$

i.e. the average divergence between the predicted distribution and its calibration. It is hard to estimate in practice, because of conditioning on a real vector $\pi(X) = \pi(x)$ in the definition of calibration $\phi[\pi]$. It becomes tricky to verify whether a model is calibrated using only a finite data sample.

Different methods have been proposed based on binning of Δ (Vaicenavicius et al., 2019) or using kernel-based divergences (Widmann et al., 2019). Unfortunately, statistical tests based on unbiased estimates (Widmann et al., 2019) were unable to reject the hypothesis that any basic neural network in a real setting, such as on MNIST data, is already calibrated. No calibration methods were proposed based on this miscalibration.

3.2 Calibration for Statistical Decision Making

Let us consider the classical statistical decision making problem for a known model p^*. Let $\mathcal{D}$ be a finite decision space. Consider the *cost matrix* $l : \mathcal{Y} \times \mathcal{D} \to \mathbb{R}_+$ and a *decision strategy* $q \colon \mathcal{X} \to \mathcal{D}$. The *risk* of the strategy q and the optimal (Bayesian) decision strategy are, respectively:

$$R^*[q] = \mathbb{E}\left[l(Y, q(X))\right], \qquad q^*(x) = \arg\min_d \sum_y p^*(y|x)\left[l(y,d)\right]. \tag{6}$$

In practice, we do not have access to the true distribution p^* to make decisions, only to the model $p(y|x)$. Let $f(d,x) = \sum_y p(y|x)l(y,d)$ denote the model-based conditional risk for observation x. The *model-based risk* and *model-based Bayesian decision strategy* are, respectively:

$$\hat{R}[q] = \mathbb{E}\left[f(d,x)\right], \qquad \hat{q}(x) = \arg\min_d f(d,x). \tag{7}$$

If the model p was distribution-calibrated, these two risks would coincide: $\hat{R}[q] = R^*[q]$ for any strategy q and any cost matrix (Zhao et al., 2021). However, as was discussed above, distribution calibration is hard even to measure in practice. This has led to the following definition.

Definition 3 (Zhao et al. 2021)**.** For a set of cost matrices $\mathcal{L}$ and a set of strategies Q, the predictor π is called $(\mathcal{L}, Q)$*-decision calibrated* if for all $l \in \mathcal{L}$ and $q \in Q$ the model risk matches the true risk: $\hat{R}[q] = R^*[q]$.

Zhao et al. (2021) show that this definition generalizes previous notions of calibration by specifying the corresponding statistical decision problems. In particular confidence calibration corresponds to recognition with the reject option with a varied cost of rejecting. The distribution calibration can be understood as $(\mathcal{L}, Q)$-decision calibration for all possible loss functions and decision strategies over all possible decision spaces $\mathcal{D}$, which is clearly far too general.

It follows from the definition that a decision-calibrated model must accurately estimate the true risk and that the model-based strategy $\hat{q}$ is the optimum of the true risk over Q. Therefore (L, Q)-decision calibration is sufficient for any statistical decision making task with $l \in L$ and $q \in Q$.

4 Method

The condition of $(\mathcal{L}, Q)$ decision calibration (Zhao et al., 2021) is still unnecessarily stringent if we have a specific fixed cost matrix l and are interested in the performance of only one particular decision strategy: the model-based Bayesian strategy $\hat{q}$, (7). Their calibration algorithm is derived under the assumption that $\mathcal{L}$ is the set of all cost matrices of bounded norm over a fixed decision space and thus cannot be chose as *e.g.* $\mathcal{L} = \{l\}$.

We will show that the minimization of the risk of the model-based strategy, $R^*[\hat{q}]$, can improve a precise measure of calibration under a known cost matrix while also obviously not compromising on the task-specific performance metric, which is the risk $R^*[\hat{q}]$ itself.

4.1 Calibration via Loss Minimization

Bröcker (2009) showed that any loss function corresponding to a proper scoring rule satisfies a decomposition into uncertainty, resolution (sharpness) and reliability (miscalibration). A *scoring* rule S is a function $\Delta \times \mathcal{Y} \to \mathbb{R}$ and the expected score, which we call *loss* for brevity, is $\mathcal{L}[\pi] = \mathbb{E}[S(\pi(X), Y)]$. For example, the negative log likelihood loss (NLL) corresponds to the scoring rule $S(\pi, y) = -\log \pi_y$. The decomposition reads

$$\mathcal{L}[\pi] = \underbrace{H(\bar{\pi})}_{\text{uncertainty of } Y} \underbrace{- \mathbb{E}\left[D(\bar{\pi}, \phi[\pi](X))\right]}_{\text{resolution of } \pi} + \underbrace{\mathbb{E}\left[D(\pi(X), \phi[\pi](X))\right]}_{\text{reliability of } \pi}, \tag{8}$$

where $\bar{\pi}$ is the *a priori* distribution of labels: $\bar{\pi}_y = p^*(y)$, $\phi[\pi]$ is the calibration of π (4), and H and D are particular entropy and the divergence functions corresponding to the score S. In case of NLL, they are the Shannon entropy and the Kullback–Leibler divergence.

Prominently, the reliability term in this decomposition is exactly the expected miscalibration (5) w.r.t. the score-specific divergence D. If we substitute $\phi[\pi]$ as a predictor, we will find out that it has a zero expected miscalibration while the first two terms remain the same:

$$\mathcal{L}[\phi[\pi]] = H(\bar{\pi}) - \mathbb{E}\left[D(\bar{\pi}, \phi[\pi](X))\right] \leq \mathcal{L}[\pi], \tag{9}$$

where the equality uses the fact that $\phi[\phi[\pi]] = \phi[\pi]$ and the inequality is due to divergence being always non-negative. Thus $\phi[\pi]$ not only achieves distribution calibration but also is guaranteed not to decrease all losses corresponding to proper scoring rules.

This sheds some light on why optimizing NLL is good for calibration as evidenced, *e.g.*, by Guo et al. 2017; Alexandari et al. 2020, in particular improving ECE. Calibration methods often fit a parametric post-processing of a predictor π, such as temperature scaling (Guo et al., 2017). They argue about calibration but optimize NLL. We formally show why this is a perfectly correct idea.

Theorem 1. *Let $\pi\colon \mathcal{X} \to \Delta$ be a predictor and $T_\theta\colon \Delta \to \Delta$ a parametric mapping, invertible for each $\theta \in \Theta$. Finding an adjusted predictor $\pi_\theta = T_\theta \circ \pi$ minimizing the expected miscalibration is equivalent to minimizing the loss:*

$$\min_{\theta\in\Theta} \mathbb{E}\left[D(\pi_\theta, \phi[\pi_\theta])\right] = \min_{\theta\in\Theta} \mathcal{L}[\pi_\theta]. \tag{10}$$

Proof. First we show that $\phi[T \circ \pi]$ is invariant of T for any invertible T. The events $T(\pi(X)) = T(\pi(x))$ and $\pi(X) = \pi(x)$ are equal, therefore

$$\phi[T \circ \pi](x)_y = \mathbb{P}\big(Y{=}y \mid T(\pi(X)){=}T(\pi(x))\big) = \phi[\pi](x)_y. \tag{11}$$

It follows that $D(\bar{\pi}, \phi[T \circ \pi](X)) = D(\bar{\pi}, \phi[\pi](X))$. Therefore the first two terms of the decomposition stay the same for any θ. Therefore minimizing the whole loss over $\theta \in \Theta$ is equivalent to minimizing the reliability term alone. □

This allows to overcome the general difficulty of estimating the expected miscalibration by simply using the empirical estimate of the loss! In particular, no binning of the simplex Δ is involved.

4.2 Decomposition of the Risk

We observe that the true risk of the model-based strategy $R^*[\hat{q}]$ also corresponds to a proper scoring rule and thus can be decomposed according to the theory.

Proposition 1. The following *scoring rule* corresponds to the loss of the model-based decision:

$$S(\pi, y) = l(y, \arg\min_d \textstyle\sum_y \pi_y l(y, d)). \tag{12}$$

For two probability distributions π, ρ in Δ, Bröcker (2009) defines the following *scoring* function s, *divergence* D and *entropy* H:

$$s(\pi, \rho) = \sum_y S(\pi, y)\rho_y; \qquad D(\pi, \rho) = s(\pi, \rho) - s(\rho, \rho); \qquad H(\rho) = s(\rho, \rho). \tag{13}$$

In our case, the score $s(\pi(x), p^*(\cdot|x))$ is the conditional risk of the prediction $\hat{q}(x)$ and its expectation is the risk of the strategy $\hat{q}$: $\mathbb{E}[S(\pi(X), Y)] = R^*[\hat{q}]$.

Proposition 2. The score s is proper (Bröcker, 2009), *i.e.*, the "divergence" D is non-negative.

Proof. By definition,

$$s(\rho, \rho) = \sum_y S(\rho, y)\rho_y = \sum_y \rho_y l(y, \arg\min_d \sum_y \rho_y l(y, d)) = \min_d \sum_y \rho_y l(y, d). \tag{14}$$

Clearly it satisfies $s(\rho, \rho) \leq \sum_y \rho_y l(y, \hat{d})$ for any $\hat{d}$, in particular $\hat{d} = \arg\min_d \sum_y \pi_y l(y, d)$. □

Corollary 1. The decomposition (8) holds for the risk $R^*[\hat{q}]$.

The uncertainty term $H(\bar{\pi}) = \min_d \sum_y p^*(y)l(y,d)$ is just the lowest risk attainable without considering observations. Let us discuss the reliability term. In our case D is not a true divergence as it may vanish even if the two distributions are different. The reliability term is therefore more permissive. This is appropriate, indeed, if *e.g.*, the cost matrix has two identical rows, there is no need to distinguish the respective classes in the prediction and, respectively, no need to have the correct individual predictive probabilities for them. This motivates us to define the task-specific calibration accordingly:

Definition 4. Given a cost matrix l and "divergence" D_l defined by (13), a predictor $\pi(X)$ is *l-decision calibrated* if

$$D_l\big(\pi(X), \phi[\pi](X)\big) \overset{\text{a.s.}}{=} 0. \tag{15}$$

For any proper divergence, this definition would be equivalent to the distribution calibration in Definition 2. The selectivity of D_l in penalizing differences in the distribution which matter for the decision task is what makes it task-specific. It appears hard to estimate this miscalibration in general as it still involves $\phi[\pi]$. However, using Theorem 1 we can improve this task-specific calibration in parametric settings (*e.g.* temperature scaling) by simply minimizing the empirical risk of $\hat{q}$.

4.3 Empirical Risk Minimization

Consider a parametric predictor $\pi(x)_y = p(y|x;\theta)$ and let $f(d,x;\theta) = \sum_y p(y|x;\theta)l(y,d)$ as before. Given a sample $(x_i, y_i)_{i=1}^N$ from p^* the empirical risk minimization for model-based Bayesian strategy reads:

$$\min_\theta \tfrac{1}{N}\sum_i l(y_i, d_i) \quad \text{s.t.} \quad d_i = \arg\min_d f(d, x_i; \theta). \tag{16}$$

This problem is difficult because it is a so called bi-level optimization problem which has discrete decision of the inner problem and a non-linear dependence on θ. Such formulations, where the inner problem corresponds to a general predictor based on solving a combinatorial optimization problem have been studied. Two methods that have been applied to this kind of problems are large margin methods (Tsochantaridis et al., 2005; Taskar et al., 2005) and the direct loss / combinatorial black box minimization (Song et al., 2016; Vlastelica et al., 2019).

4.3.1 Direct Loss and Margin Rescaling

The empirical risk can be easily evaluated but cannot be differentiated because of $\arg\min$. This $\arg\min$ over the set of labels can be considered as a small combinatorial solver. We will specialize and analyze direct loss method (Song et al., 2016; Vlastelica et al., 2019) for this case. For simplicity, let us consider a single training sample (x^*, y^*) (with multiple training samples losses and gradients sum up). Let us denote the vector of class probabilities $\pi = p(\cdot|x^*;\theta)$. The estimate of the gradient in π according to the direct loss minimization approach is constructed as follows:

$$\hat{d} = \arg\min_d f(d, x^*); \quad \hat{d}_\lambda = \arg\min_d \big[f(d,x^*) + \lambda l(y^*, d)\big]; \quad \hat{\nabla}_\pi := \tfrac{1}{\lambda}[l(\cdot, \hat{d}_\lambda) - l(\cdot, \hat{d})]. \tag{17}$$

Appendix A.1 gives details on how this is obtained from the general method of Vlastelica et al. (2019). The gradient in θ can then be computed by the chain rule. Here $\hat{d}$ is the solution of the solver (the Bayesian decision) and $\hat{d}_\lambda$ is the decision of a perturbed problem. The strength of the perturbation is controlled by λ. Song et al. (2016) has shown that in the limit $\lambda \to 0$ the gradient of the expected loss over a continuous data distribution matches $\mathbb{E}[\nabla_\pi]$. In this limit, stochastic descent with ∇_π would directly minimize the (expectation of non-differentiable) loss, which was termed *direct* loss *minimization*. However, these arguments are not applicable to a finite training sample. In practice, λ needs to be sufficiently large for ∇_π to be non-zero for some data points, at least. In this setting we are not longer minimizing the original loss. However one can define a surrogate loss function such that (17) is its true gradient. We call it the *direct loss*, so the method can now be validly interpreted as *direct loss* minimization:

$$L_\lambda^\pm = \pm\tfrac{1}{\lambda}\Big(\min_d f(d,x^*) - \min_d \big[f(d,x^*) \mp \lambda l(y^*, d)\big]\Big), \tag{18}$$

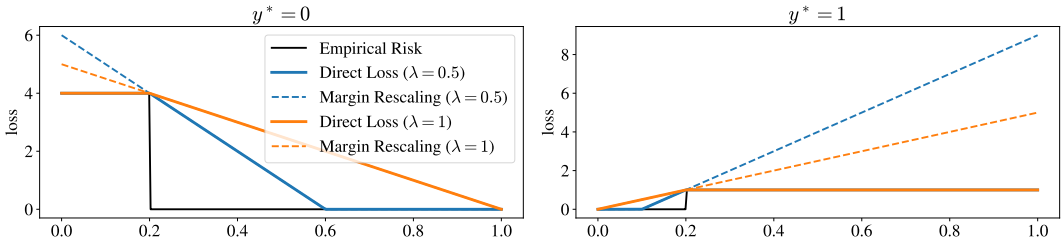


Figure 1: Comparison of margin rescaling and direct loss for binary classification with asymmetric errors: $l(0,1) = 1$, $l(1,0) = 4$. The x-axis show the model probability $\pi_0 = p(y = 0|x)$. In the case when the true label $y^* = 0$ (left), the correct decision is made when $\pi_0 > 0.2$, the cost of the error is high. If the true label $y^* = 1$ (right), the correct decision is made when $\pi_0 < 0.2$, the cost of an error is low. Observe that margin rescaling coincides with the direct loss in the region of correct classification and is an upper bound in the case of error. Because of the hinge penalty, the resulting upper bound is rather loose, more so for more imbalanced losses.

where $\mp$ is paired with $\pm$. Vlastelica et al. (2019) advocate the use of a large λ, define a similar surrogate loss to L^-_λ and show that it is a lower bound on the empirical loss for positive λ (Observation 3), where the empirical loss is $L^{\mathrm{E}} = l(y^*, \arg\min_d f(d, x^*))$. Note that there holds $L^{\pm}_{-\lambda} = L^{\mp}_{\lambda}$ and therefore we can always assume $\lambda > 0$ in order to avoid redundancy. We show the following.

Proposition 3. Direct loss L^-_λ is a lower bound on the empirical loss L^{E} and L^+_λ is an upper bound.

The proof is given in Appendix A.1. It follows that the expectation over the training data (resp. true distribution p^*) of L^+_λ is an upper bound on the empirical risk (resp. true risk).

Relation to Margin Rescaling The problem of minimizing empirical risk over discrete strategies of the form $\arg\min_y f(y;\theta)$ was also studied in structural prediction (Tsochantaridis et al., 2005; Taskar et al., 2005). One of the most common approaches is called margin re-scaling (Tsochantaridis et al., 2005) and was successfully used in combination with deep networks as well (*e.g.* Knöbelreiter et al. 2017). Like SVM, it puts a hinge loss on the violation of the classification constraints with the margin proportional to the respective loss. We can show (see Appendix A.2) that the margin re-scaling approach leads to the following surrogate loss:

$$L^{\mathrm{MR}}_\lambda = \tfrac{1}{\lambda}\Big(f(d^*, x^*) - \min_d \big[f(d, x^*) - \lambda l(y^*, d)\big]\Big), \tag{19}$$

where $d^* = \arg\min_d l(y^*, d)$ is the best decision given the true class label. Written in this form there is a striking similarity to (18). The only difference being that d^* is the best decision for a loss (knowing the true label) rather than the best decision based on the model (not knowing the true label). This leads to that margin rescaling is a less tight upper bound.

Proposition 4. Margin re-scaling L^{MR}_λ coincides with the direct loss L^+_λ in the region where the classifier makes correct decisions. Furthermore $L^E \leq L^+_\lambda \leq L^{\mathrm{MR}}_+$.

The proof is given in Appendix A.2. We believe this connection has not been known before. The two surrogate losses are illustrated in Fig. 1.

For both approaches, if λ is small, the size of the margin is small and there is a flat region with zero gradient. As a simple remedy we propose to smooth the minimum in (18) using the smooth minimum function $\min^\beta(x) = -\frac{1}{\beta}\log\sum_k e^{-\beta x_k}$, where the smoothing degree is controlled by β.

5 Experiments

In the experiments, we compare different calibration criteria for the same choice of a parametric family. Assuming that networks outputs scores s (or elsewise let $s_y = \log p(y|x)$), we consider the following common choices to parametrize the corrected predictor π. **TS**: Temperature Scaling (Guo et al., 2017): $\pi = \mathrm{softmax}(s/T)$, where T is a (non-negative) scalar temperature to calibrate. **BCTS**: Bias-Corrected Temperature Scaling (Alexandari et al., 2020): $\pi = \mathrm{softmax}((s + b)/T)$,

Results

Parametrization	Calibration Criterion	Test Empirical Risk
No calibration	—	2117 ± 886
TS	NLL	944 ± 32
	ECE	944 ± 40
	Direct Loss	$\mathbf{758 \pm 35}$
BCTS	NLL	897 ± 31
	ECE	823 ± 34
	Direct Loss	$\mathbf{730 \pm 31}$
VS	NLL	1416 ± 649
	ECE	1095 ± 70
	Direct Loss	$\mathbf{779 \pm 68}$

Cost matrix

	Accept	Reject
deadly_poisonous	10000	0
poisonous	1000	0
inedible	100	0
edible_bad	40	0
edible	0	10
edible_good	0	20

Data Examples:

Figure 2: Fungi Experiment. In results report mean and standard deviation of all calibration methods with respect to 15 validation-test splits.

where additionally b is a vector of per-class biases to calibrate. **VS**: Vector Scaling (Alexandari et al., 2020): $\pi = \mathrm{softmax}(s \odot w + b)$, where w is a vector of scaling factors, b is a vector of biases and $\odot$ is the coordinate-wise product.

We optimize each criterion in the above parameters using Adam optimizer. In order to find hyperparameters (learning rate, λ, β) we use the nested cross-validation procedure detailed in Appendix B.

5.1 Fungi Edibility (Danish Fungi 2020)

In this experiment we consider a decision problem of whether to cook (eat) a mushroom given its predicted edibility category, based on the Danish fungi dataset (Picek et al., 2022). In order to compare calibration methods, we create 15 folds of the data that was not used during training into calibration and test parts. Full details can be found in Appendix B. The cost matrix and obtained results are shown in Fig. 2. Calibration with the DirectLoss criterion achieved a lower average test risk in all parametrizations, notably performing well also in the VS parametrization where other criteria performed the worst. The improvement over other methods can be considered statistically significant if one trusts the estimates of the mean and the variance (see below).

5.2 Skin Cancer Lesion Treatment (HAM10000)

In this experiment, we consider a decision problem of whether to assign a treatment given the lesion classification using skin lesion dataset (Tschandl, 2018). The training is performed on 75% of the data for 100 epochs. From the remaining data we create 100 random splits into calibration (15%) and test (10%). Fig. 3(a) shows that the network significantly underestimates the true risk. After calibration (DirectLoss BCTS), the risk decreases, but the risk gap increases for some data splits. Indeed, with our calibration and optimization criterion being the empirical risk, there is no requirement that this gap should be made small or even decrease. Nevertheless, such increase in the gap is unexpected of a calibration method and might indicate overfitting. Fig. 3 (a,b) show statistics of the differences between pairs: No calibration - DirectLoss and NLL - DirectLoss, confirming that calibration is helpful, but unable to tell whether NLL or DirectLoss is a better calibration objective. Comparisons for TS and VS parametrizations are shown in Figs. B.2 and B.3.

5.3 Rare Expensive Mistakes

We present a failure mode of calibration on the example of CIRAF10 dataset with the trucks class considered as dangerous (cost of mistake 10000) and other mistakes cost 1. In this setting the decision boundary of $\hat{q}$ significantly shifts towards classifying nearly all observations as trucks. Instances of trucks for which the model can nevertheless make a mistake become very rare. Depending on whether such an instance falls into the calibration set or into the test set, it may lead to a high cost at the test time. In Fig. 4 for many splits, DirectLoss may be better than NLL in calibration, but in one split it makes an expensive single mistake. Only by chance such case was not observed for

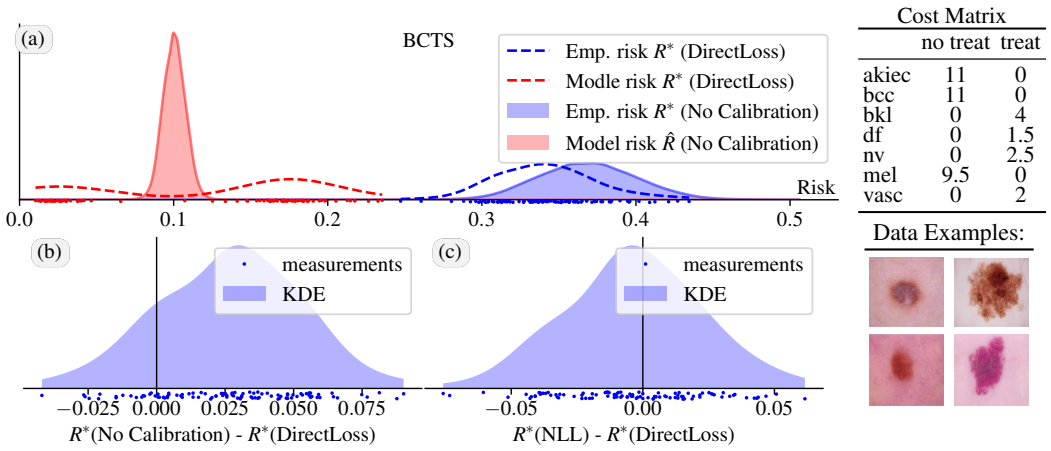
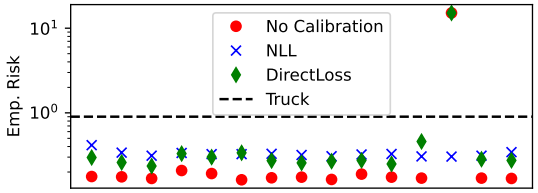


Figure 3: HAM10000 Experiment. (a): Model's self-assessed risk $\hat{R}(\hat{q})$ substantially underestimates the true risk $R^*(\hat{q})$. After calibration the true risk improves but the model-assessed risk gets further away for some data splits, indicating overfitting. (b): Statistical comparison of improvement due to calibration (positive means improvement). (c): Statistical comparison of NLL with DirectLoss calibrations (positive means DirectLoss was better). BCTS parametrization is used.



Cost Matrix: $l(k, d) =$

$$\begin{cases} 0, & \text{if } d = k; \\ 1, & \text{if } d \neq k \text{ and } k \neq \texttt{truck}; \\ 10000, & \text{if } d \neq k \text{ and } k = \texttt{truck}. \end{cases}$$

Figure 4: Empirical Test Risk in different validation-test splits without and with calibration (TS). The dashed line shows a trivial baseline: the constant strategy that classifies any input a truck.

NLL. Aslo, this was not observed in the Fungi experiment above (which also has extreme costs) presumably because deadly poisonous mushrooms are rather rare in the dataset.

Empirical risk is theoretically backed up by the generalization guarantees such as Hoeffding inequality: $\mathbb{P}(|R^*(q) - R^*_{\text{emp}}(q)| > \varepsilon) < 2e^{-2N\varepsilon^2/\Delta l^2}$, where N is the number of samples and Δl is the difference between the maximum and minimum cost. This means that in order to achieve the same confidence we used to have for 0-1 cost, we need to use 10^8 times more samples. We therefore would like to warn the community from relying on basic statistical evaluation like in our Fungi experiment and would be happy to receive feedback on how to approach the problems associated with high costs, in particular when evaluating calibration methods.

6 Conclusion

We have given a so-far-missing theoretical justification for post-processing recalibration methods optimizing generic criteria, in particular NLL, showing how they are related to notions of calibration. We then developed a decomposition of the risk of model-based Bayesian decision strategy and derived the respective definition of calibration from it. This approach gives a constructive way to obtain new task-specific definitions of calibration. We then improved the understanding of direct loss and margin rescaling methods for ERM. We believe these results generalize beyond our calibration setup. In the experiments we observed that calibration was important to improve the test risk and that the task-specific calibration, represented by the DirectLoss, can be more efficient (Fungi experiment, high costs). The calibration was also helpful in the lesions experiment (moderate costs), however the increase in the risk gap indicates an overfitting with DirectLoss. Finally, we demonstrated a failure case of DirectLoss and a flaw in the comparison under high costs.

ETHICS STATEMENT

Please be aware that neural networks can make unpredictable mistakes and produce overconfident estimates. Calibration methods, in particular the proposed one, are not guaranteed to fix these issues. They can improve statistical performance and measures of miscalibration. However, the statistics are random quantities and have to be considered very carefully, especially in the case of high costs, as we show in Section 5.3. The experiments conducted on decision making with fungi or lesion datasets should be considered only as proof of concept.

REPRODUCIBILITY STATEMENT

Appendix A contains proofs not included in the main paper. Appendix B contains description of datasets and details of training, calibration and testing procedures. Details of implementation can be provided to reviewers confidentially through OpenReview upon request.

REFERENCES

Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *ICML: International Conference on Machine Learning*, pp. 222–232, 2020.

Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, 2009.

Morris H. Degroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1983.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML: International Conference on Machine Learning*, pp. 1321–1330, 2017.

Patrick Knöbelreiter, Christian Reinbacher, Alexander Shekhovtsov, and Thomas Pock. End-to-end training of hybrid CNN-CRF models for stereo. In *CVPR: Computer Vision and Pattern Recognition Conference*, 2017.

Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. In *NeurIPS: Advances in Neural Information Processing Systems*, 2019.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *NeurIPS: Advances in Neural Information Processing Systems*, volume 32, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML: International Conference on Machine Learning*, volume 80, pp. 2805–2814, 10–15 Jul 2018.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI Conference on Artificial Intelligence*, pp. 2901–2907, 2015.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, volume 2, 2019.

Lukáš Picek, Milan Šulc, Jiří Matas, Thomas S Jeppesen, Jacob Heilmann-Clausen, Thomas Læssøe, and Tobias Frøslev. Danish fungi 2020-not just another image recognition dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1525–1535, 2022.

Yang Song, Alexander Schwing, Raquel Urtasun, et al. Training deep neural networks via direct loss minimization. In *ICML: International Conference on Machine Learning*, pp. 2169–2177, 2016.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *ICML: International Conference on Machine Learning*, pp. 896–903, 2005.

Philipp Tschandl. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, 2018.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(50):1453–1484, 2005.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *AISTATS: International Conference on Artificial Intelligence and Statistics*, pp. 3459–3467, 2019.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. *arXiv preprint arXiv:1912.02175*, 2019.

Bin Wang and Xiaofeng Wang. Bandwidth selection for weighted kernel density estimation. *arXiv: Methodology*, 2007.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In *NeurIPS: Advances in Neural Information Processing Systems*. Curran Associates Inc., 2019.

Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. In *NeurIPS: Advances in Neural Information Processing Systems*, volume 34, pp. 22313–22324, 2021.

Appendix

A PROOFS

A.1 DIFFERENTIATION OF BLACKBOX COMBINATORIAL SOLVERS (DIRECT LOSS)

We first detail how the general method of Vlastelica et al. (2019) is instantiated for our problem and verify that it is the gradient of the function L_λ^- we define in (18).

A general linear combinatorial solver is formalized in Vlastelica et al. (2019) as:

$$\text{Solver}(w) = \arg\min_d w^T \phi(d), \tag{20}$$

where ϕ represents discrete choice d as a vector of the same dimension as w. And the direct loss method (Vlastelica et al., 2019, Alg.1) is given by

$$\hat{d} := \text{Solver}(w); \tag{21a}$$

$$w' := w + \lambda \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\phi}(\hat{d}); \tag{21b}$$

$$\hat{d}_\lambda := \text{Solver}(w'); \tag{21c}$$

$$\nabla_w := -\frac{1}{\lambda}\Big[\phi(\hat{d}) - \phi(\hat{d}_\lambda)\Big]. \tag{21d}$$

In our case the solver needs to be

$$\hat{d} = \arg\min_d f(d, x^*) = \arg\min_d \sum_y p(y|x^*;\theta) l(y,d). \tag{22}$$

Let $\pi = p(\cdot|x^*;\theta)$. Two choices for ϕ qualify:

1. Let $\phi(d) = \text{one\_hot}(d)$ and $w \in \mathbb{R}^D$ with $w_d = \sum_y \pi_y l(y,d)$;
2. Let $\phi(d)_y = l(y,d)$ and $w \in \mathbb{R}^K$ with $w_k = \pi_k$.

Both choices lead to equivalent algorithms. We proceed with the second one for convenience as it will define the gradient in π. Our loss is $\mathcal{L}(d) = l(y^*, d)$, therefore $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\phi_y}(\hat{d}) = [\![y{=}y^*]\!]$. The direct loss method specializes as follows:

$$\hat{d} := \arg\min_d \sum_y \pi_y l(y,d); \tag{23a}$$

$$\pi' := \pi + \lambda\, \text{one\_hot}(y^*); \tag{23b}$$

$$\hat{d}_\lambda := \arg\min_d \sum_y \pi'_y l(y,d) = \arg\min_d \Big(\sum_y \pi_y l(y,d) + \lambda l(y^*,d)\Big); \tag{23c}$$

$$\nabla_\pi := -\frac{1}{\lambda}[l(\cdot,\hat{d}) - l(\cdot,\hat{d}_\lambda)]. \tag{23d}$$

This is the form we present in (17). Finally, observe that the gradient ∇_π in (23) matches the gradient of L_λ^- as defined in (18). Therefore minimizing L_λ^- is equivalent to the method of Vlastelica et al. (2019).

Next we give a very simple proof of the upper / lower bound property of $L_\lambda^\pm$ (it is extendible to the general combinatorial solver case as well).

Proposition 3. Direct loss L_λ^- is a lower bound on the empirical loss L^{E} and L_λ^+ is an upper bound.

Proof. We will assume that all losses are non-negative (wlog) and will show the bound property for a given training sample (x^*, y^*).

Let $f(d) = \sum_y p(y|x^*) l(y,d)$ and let $\hat{d} = \arg\min_d f(d)$. Using the inequality

$$\min_d \big[\sum_y p(y|x^*) l(y,d) + \lambda l(y^*,d)\big] \geq \frac{1}{\lambda} f(\hat{d}) + \lambda l(y^*, \hat{d}), \tag{24}$$

in L_- all terms cancel except of $l(y^*, \hat{d})$.

Similarly, using the inequality

$$-\min_d \big[\sum_y p(y|x^*)l(y,d) - \lambda l(y^*,d)\big] \geq -f(\hat{d}) + \lambda l(y^*,\hat{d}) \tag{25}$$

in L_+ all terms cancel except of $l(y^*, \hat{d})$. □

A.2 Derivation of Margin Rescaling

The derivation of margin rescaling approach in Tsochantaridis et al. (2005) is somewhat obscure. The reasonable starting point could be given by the SVM-like objective with slacks (but without the quadratic penalty on the weights):

$$\begin{aligned} &\frac{1}{\lambda}\min_{\xi,\theta}\sum_i \xi \\ &\text{s.t.}\quad (\forall d)\;\; f_i(d_i^*) \leq f_i(d) - \lambda(l(y_i,d) - l(y_i,d^*)) + \xi_i, \end{aligned} \tag{26}$$

where $f_i(d) = \sum_y p(y|x_i;\theta)l(y,d)$, (x_i, y_i) is the i'th training example and $d_i^* = \arg\min l(y_i, d)$ is the optimal decision for the training example i. The constraint in this formulation requires that the model loss of the best decision $f_i(d_i^*)$ must be strictly less that the loss of any other decision $f_i(d)$ with a *margin* $\lambda(l(y_i,d) - l(y_i,d^*))$, proportional to the loss excess of the respective decision. A violation of this constraint is penalized by a slack ξ_i and the goal is to minimize the total slack. Notice that the constraint ensures that the slack is non-negative because for $d = d_i^*$ all terms except ξ_i vanish.

Solving for optimal ξ_i in each summand, we obtain that the summand i can be expressed as

$$L_\lambda^{\mathrm{MR}} = \frac{1}{\lambda}\max_d(f_i(d_i^*) - f_i(d) + \lambda(l(y_i,d) - l(y_i,d^*))) \tag{27}$$

$$= \frac{1}{\lambda}\Big(f_i(d_i^*) - \min_d(f_i(d) - \lambda(l(y_i,d) - l(y_i,d^*)))\Big). \tag{28}$$

Finally, under the assumption that costs l are non-negative and that $l(y_i, d^*) = 0$ (which can be made without loss of generality), we obtain the formulation (19).

Proposition 4. Margin re-scaling L_λ^{MR} coincides with the direct loss L_λ^+ in the region where the classifier makes correct decisions. Furthermore $L^E \leq L_\lambda^+ \leq L_+^{\mathrm{MR}}$.

Proof. The inequality $L_\lambda^+ \geq 0$ is already shown in Proposition 3. The proof is simple, once the two approaches are written in the respective forms that we have shown:

$$L_\lambda^+ = \frac{1}{\lambda}\Big(\min_d f(d,x^*) - \min_d\big[f(d,x^*) - \lambda l(y^*,d)\big]\Big), \tag{29a}$$

$$L_\lambda^{\mathrm{MR}} = \frac{1}{\lambda}\Big(f(d^*,x^*) - \min_d\big[f(d,x^*) - \lambda l(y^*,d)\big]\Big). \tag{29b}$$

Let us verify that $L_\lambda^{\mathrm{MR}} \geq L_\lambda^+$. Since the summand $-\min_d\big[f(d,x^*) - \lambda l(y^*,d)\big]$ is common in both, the inequality follows trivially from

$$f(d^*,x^*) \geq \min_d f(d,x^*). \tag{30}$$

The remaining claim of the proposition is also trivial. If the decision made by classifier is correct, *i.e.*, the optimal one, then (30) holds with equality. □

B Experiment Details

B.1 Cross-Validation Procedure

Given a subset of data available for calibration (in the current calibration-test split), we create 10 folds for the internal cross-validation. We used stratified folds to maintain the class balance. In each

Hyperparameter	Searched Values
learning rate	1, 0.1, 0.01, 0.001
λ	-50, -10, -5, -1
β	None, 1, 5, 10, 20, 30, 40

Table B.1: Search grid for the cross-validation procedure.

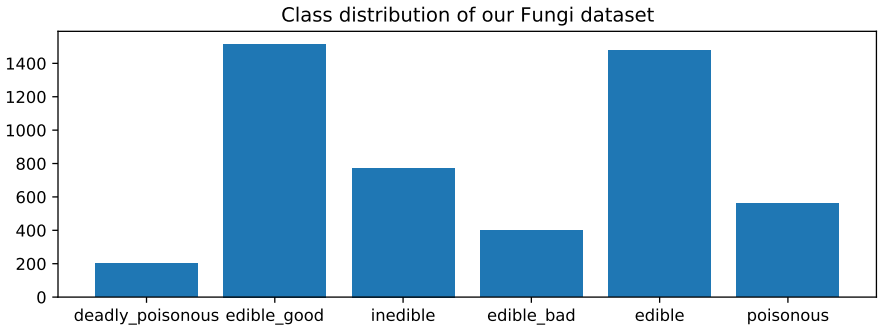


Figure B.1: The distribution of edibility classes in the remapped Fungi dataset.

fold we have $9/10$ for optimization of calibration parameters and $1/10$ for validation of hyperparameters. Hyperparameters corresponding to the best average risk over the 10 folds are selected. We perform selection of the following hyper-parameters: learning rate α for all methods; λ and β for Direct Loss with smooth minimum. The chosen lambda values are then multiplied by $1/\kappa$, where κ is the maximum value of the loss function. This is to normalize the loss function to be invariant to the scale of lambda. The search grids for different methods are shown in Table B.1.

B.2 Fungi Experiment

The trained neural network for mushroom classification (Picek et al., 2022) is adapted to our decision problem (to decide the edibility of the mushrooms) as follows. There is 1604 species in the dataset, out of which we found and annotated the edibility information (6 categories) for 203 species. After this procedure the distribution of species becomes uneven, as shown in Fig. B.1. In particular deadly poisonous mushrooms are relatively rare.

We adopted the ResNet-50 network from (Picek et al., 2022) as follows. From the probability vector over spices produced by the model we compute the probability vector over edibility states by marginalization. The accuracy of the model in classifying these 6 states was at 91%. Then we consider a decision problem with 6 states and 2 decisions (accept or not for cooking). We designed a realistic loss function, shown in Fig. 2 top-right. The calibration-test splits were created by using 15 stratified folds of the test set and adding the validation set of the training to the calibration set. For this decision task, we are not longer interested in the accuracy of the classification, but in the expected loss, *i.e.* the risk shown in Fig. 2 left.

B.3 HAM10000 Experiment

We tried to follow the setup of Zhao et al. (2021) in order to allow for an indirect comparison[1]. In particular we used the same data split and network and tried to evaluate also the gap between the model-estimated (emperical) risk and the true empirical risk. We trained resnet121 model for 100 epochs on 75% of the data. All lesions having multiple views in the dataset were used for training. The remaining 25% consisted of independent instances, each with 1 view only. The training achieved validation accuracy of 90% (the validation set was not used for choosing hyperparameters, only to report this number). The 25% of the data not used for training we split randomly into 15% for calibration and 10% for test. All splits were stratified (preserving class balance). This results in

[1]A direct comparison is not feasible at the moment: we evaluate only parametric calibration methods; the code and some details of their method are not available to us

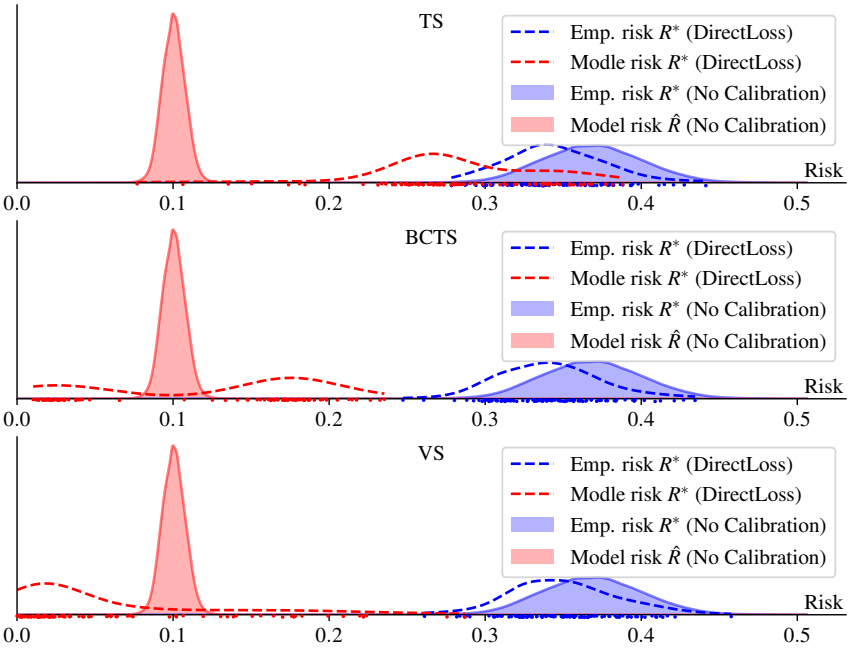


Figure B.2: HAM10000 Experiment: Risk gap before and after calibration for TS, BCTS and VS parameterizations. The trend is that with more parameters the model self-assessed risk is less accurate, indicating an overfitting in terms of calibrating reliable probabilities.

the test set size of 1015 data points (in each split). Fig. 3 is showing the statistical analysis over 40 splits. As each split requires a calibration (with the nested cross-validation procedure), collecting more statistics is difficult. In our cost matrix we tried to closely replicate the values depicted in Zhao et al. (2021, Fig.1) (motivated by medical domain knowledge) by matching the colors in the image and the color bar. We added a constant in each row to make all losses non-negative. This affects neither the Bayesian decision strategy nor the differences between any two risks.

Pairwise comparisons for TS and VS parametrizations, complementing Fig. 3 are shown in Fig. B.3. All kernel density estimates shown are computed with awkde[2] (Wang & Wang, 2007) using the default `silverman` adaptive method. The calibration has a positive effect in these cases as well, however the advantage for VS parametrization appears to be on the side of NLL.

B.4 CIFAR-10 Experiment

In this experiment we used CIFAR-10 dataset. The data splitting and calibration protocol were the same as in the fungi experiment. We trained EfficientNetB0 that achieved validation accuracy 94.7%.

[2]Adaptive Width KDE with Gaussian Kernels https://github.com/mennthor/awkde

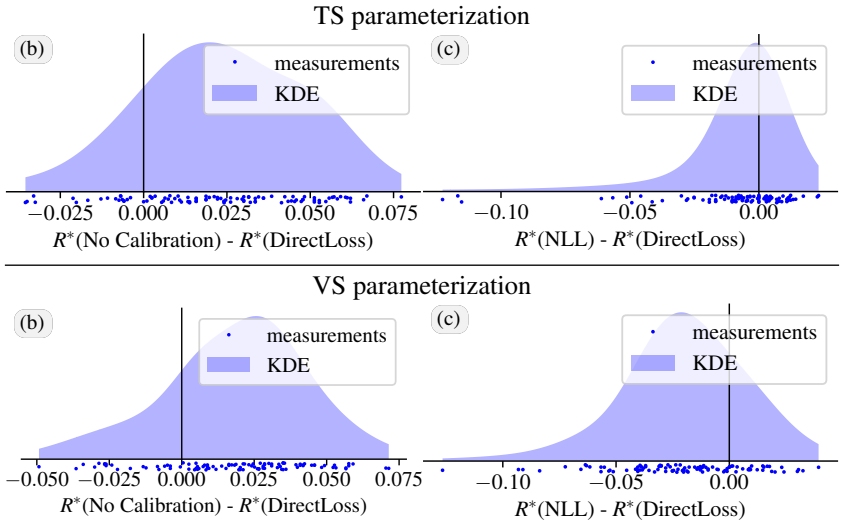


Figure B.3: HAM10000 Experiment: Pairwise comparisons for TS and VS parametrizations.