DEEP PROBABILITY ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable probability estimation is of crucial importance in many real-world applications where there is inherent uncertainty, such as weather forecasting, medical prognosis, or collision avoidance in autonomous vehicles. Probability-estimation models are trained on observed outcomes (e.g. whether it has rained or not, or whether a patient has died or not), because the ground-truth probabilities of the events of interest are typically unknown. The problem is therefore analogous to binary classification, with the important difference that the objective is to estimate probabilities rather than predicting the specific outcome. The goal of this work is to investigate probability estimation from high-dimensional data using deep neural networks. There exist several methods to improve the probabilities generated by these models but they mostly focus on classification problems where the probabilities are related to model uncertainty. In the case of problems with inherent uncertainty, it is challenging to evaluate performance without access to ground-truth probabilities. To address this, we build a synthetic dataset to study and compare different computable metrics. We evaluate existing methods on the synthetic data as well as on three real-world probability estimation tasks, all of which involve inherent uncertainty: precipitation forecasting from radar images, predicting cancer patient survival from histopathology images, and predicting car crashes from dashcam videos. Finally, we also propose a new method for probability estimation using neural networks, which modifies the training process to promote output probabilities that are consistent with empirical probabilities computed from the data. The method outperforms existing approaches on most metrics on the simulated as well as real-world data.

1 INTRODUCTION

We consider the problem of building models that answer questions such as: *Will it rain? Will a patient survive? Will a car collide with another vehicle?* Due to the inherently-uncertain nature of these real-world phenomena, this requires performing *probability estimation*, i.e. estimating the probability of each possible outcome of the phenomenon of interest. Models for probability prediction must be trained on observed outcomes (e.g. whether it rained, a patient died, or a collision occurred), because the ground-truth probabilities are unknown. The problem is therefore analogous to binary classification, with the important difference that the objective is to estimate probabilities rather than predicting specific outcomes. In probability estimation, two identical inputs (e.g. histopathology images from cancer patients) can potentially result in two different outcomes (death vs. survival). In classification, the relationship between the input data and the class label is usually deterministic (a picture either shows a cat or does not).

The goal of this work is to investigate probability estimation from high-dimensional data using deep neural networks. Deep networks trained for classification often generate probabilities, which quantify the uncertainty of the estimate (i.e. how likely the network is to classify correctly). This quantification has been observed to be inaccurate, and several methods have been developed ot improve it (Szegedy et al., 2016; Zhang et al., 2020; Thulasidasan et al., 2020; Mukhoti et al., 2020; Guo et al., 2017; Platt, 1999). However, these works restrict their attention almost exclusively to classification problems without inherent uncertainty, perhaps because most existing benchmark datasets are focused on classification.

In order to advance deep-learning methodology for probability estimation it is crucial to build appropriate benchmark datasets. Here we take a step in this direction by building a synthetic dataset

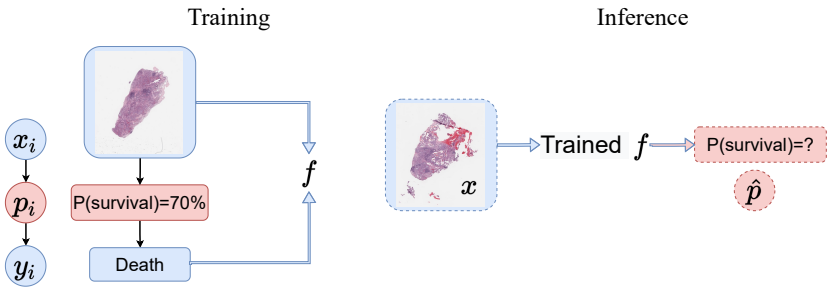


Figure 1: **The probability-estimation problem.** In probability estimation, we assume that each observed outcome y_i (e.g. death or survival in cancer patients) in the training set is randomly generated from a latent unobserved probability p_i associated to the corresponding data $\boldsymbol{x}_i$ (e.g. histopathology images). **Training** (left): Only $\boldsymbol{x}_i$ and y_i can be used for training, because p_i is not observed. **Inference** (right): Given new data $\boldsymbol{x}$, the trained network f produces a probability estimate $\hat{p} \in [0, 1]$.

and gathering three real-world datasets, which we use to systematically evaluate existing methodology for improving the probability estimates of neural networks. In addition, we propose a novel approach, which outperforms current state-of-the-art methods. In more detail, our contributions are the following:

- We introduce a new synthetic dataset for probability estimation where a population of people may have a certain disease connected to age. The task is to predict the probability that they contract the disease from a picture of their face. The data are generated based on the UTKFaces dataset (Zhang et al., 2017a), which contains age information. The dataset contains multiple versions of the synthetic labels, which are generated according to different distributions designed to mimic real-world probability-prediction datasets. The dataset serves two objectives. First, it allows us to evaluate existing methodology. Second, it enables us to evaluate different metrics in a controlled scenario where we have access to ground-truth probabilities.
- We have used publicly available data to build probability-estimation benchmark datasets for three real-world applications: (1) precipitation forecasting from radar images, (2) prediction of cancer-patient survival from histopathology images, and (3) prediction of vehicle collisions from dashcam videos. We use these datasets to systematically evaluate existing approaches, which have been previously tested mainly on classification datasets.
- We propose a novel technique called Calibrated Probability Estimation (CaPE). CaPE modifies the training process of neural networks, so that their output probabilities are consistent with empirical probabilities computed from the data. CaPE outperforms existing approaches on most metrics on the synthetic and real-world datasets.

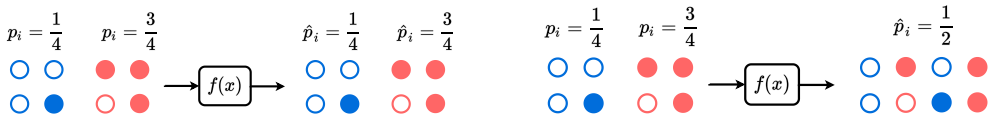
2 Problem Formulation: Probability Estimation

The goal of probability estimation is to evaluate the likelihood of a certain event of interest, based on observed data. The available training data consists of n examples $\boldsymbol{x}_i$, $1 \leq i \leq n$, each associated with a corresponding outcome y_i. In our applications of interest, the input data are high dimensional: each $\boldsymbol{x}_i$ corresponds to an image or a video. The corresponding label y_i is either 0 or 1 depending on whether or not the event in question occurred. For example, in the cancer-survival application $\boldsymbol{x}_i$ is a histopathology image of a patient, and y_i equals 1 if the patient survived for five years after $\boldsymbol{x}_i$ was collected. We assume the data have inherent uncertainty, meaning that y_i, the patient's survival, does not depend deterministically on the histopathology image. Instead, we assume that y_i equals 1 with a certain probability p_i, associated with $\boldsymbol{x}_i$, as illustrated in Figure 1.

At inference, a probability-estimation model aims to generate an estimate $\hat{p}$ of the underlying probability p, associated with a new input data point x (e.g. the probability of survival for over 5 years for new patients based on their histopathology data).

3 Evaluation Metrics

Probability estimation shares similar target labels and network outputs with binary classification. However, classification accuracy is *not* an appropriate metric for evaluating probability-estimation



(a) ✓ Accurate probability estimation
✓ Calibration

(b) ✗ Inaccurate probability estimation
✓ Calibration

Figure 2: **Calibration is not enough**. Uncolored/colored markers denote $y = 0/1$ outcomes, respectively. Blue/red stand for two classes with different associated ground-truth probabilities (1/4 and 3/4 respectively). (a) The model f retrieves the true probabilities, which requires discriminating between inputs with low and high probability. (b) The model f has no discriminative power, it just assigns the same probability to all outputs. However, the model is *perfectly calibrated* because out of all outcomes assigned 0.5 by the model, the fraction that are equal to 1 is 50%.

models due to the inherent uncertainty of the outcomes. This is illustrated by the example in Figure 2a where a perfect probability estimate would result in a classification accuracy of just 75%.

Metrics when ground-truth probabilities are available. For synthetic datasets, we have access to the ground truth probability labels and can use them to evaluate performance. Two reasonable metrics are the mean squared error or ℓ_2 distance MSE_p, and the Kullback–Leibler divergence KL_p between the estimated and ground-truth probabilities:

$$\text{MSE}_p = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - p_i)^2, \quad \text{and} \quad \text{KL}_p = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{p}_i \log\left(\frac{\hat{p}_i}{p_i}\right) + (1-\hat{p}_i)\log\left(\frac{1-\hat{p}_i}{1-p_i}\right)\right). \quad (1)$$

N is the number of data, and $p_i, \hat{p}_i$ are the ground-truth and predicted probabilities respectively.

Calibration metrics. In real-world data, ground-truth probabilities are not available. In order to evaluate the probabilities estimated by a model, we need to compare them to the observed probabilities. To this end, we aggregate the examples for which the model output equals a certain value (e.g. 0.5), and verify what fraction of them have outcomes equal to 1. If the fraction is close to the model output, then the model is said to be well calibrated.

Definition 3.1. A model f is well *calibrated* if

$$\mathbb{P}\left(y = 1 \mid f(\boldsymbol{x}) \in I(q)\right) = q, \quad \forall\, 0 \leq q \leq 1, \quad (2)$$

where y is the observed outcome, $f(\boldsymbol{x})$ is the probability predicted by model f for input $\boldsymbol{x}$, and $I(q)$ is a small interval around q.

Model calibration can be evaluated using the expected calibration error (ECE) (Guo et al., 2017). Given a probability-estimation model f and a dataset of input data $\boldsymbol{x}_i$ and associated outcomes y_i, $1 \leq i \leq N$, we partition the examples into B bins, $I_1 I_2, \cdots, I_B$, according to the probabilities assigned to the examples by the model. Let $Q_1, \ldots, Q_{B-1}$ the B-quantiles of the set $\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)\}$, we have $I_b := [Q_{b-1}, Q_b] \cap \{f(\boldsymbol{x}_i)\}_{i=1}^N$ (setting $Q_0 = 0$). For each bin, we compute the mean predicted and empirical probabilities,

$$p_{\text{emp}}^{(b)} = \mathbb{E}\left(y \mid f(\boldsymbol{x}) \in I_b\right) = \frac{1}{|I_b|}\sum_{i \in \text{Index}(I_b)} y_i, \quad (3)$$

$$q^{(b)} = \frac{1}{|I_b|}\sum_{i \in \text{Index}(I_b)} f(\boldsymbol{x}_i), \quad (4)$$

where $\text{Index}(I_b) = \{i \mid f(\boldsymbol{x}_i) \in I_b\}$.

The pairs $(q^{(b)}, p_{\text{emp}}^{(b)})$ can be plotted as a **reliability diagram**, shown in the second row of Figure 4 and in Figure 6. ECE is then defined as

$$\text{ECE} = \frac{1}{B}\sum_{b=1}^{B}\left|p_{\text{emp}}^{(b)} - q^{(b)}\right|. \quad (5)$$

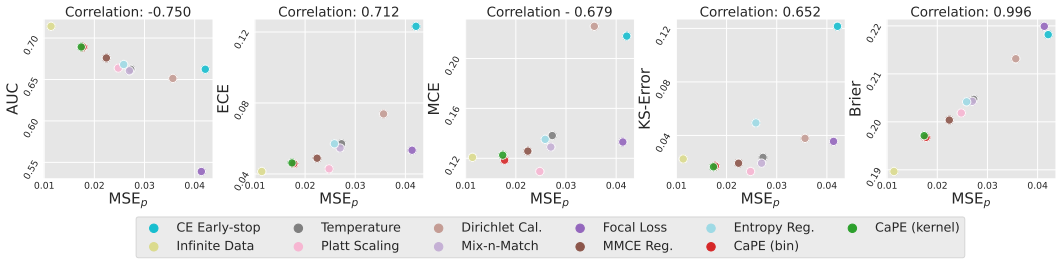


Figure 3: **Evaluating evaluation metrics.** We use synthetic data to compare different metrics to the *gold-standard* MSE_p that uses ground-truth probabilities. Brier score is highly correlated with MSE_p, in contrast to the classification metric AUC and the calibration metrics ECE, MCE and KS-Error. The graphs show the results of the proposed method CaPE, as well as the baselines described in Section 6.3 on the *Uniform* scenario (see Section 6.1. Other scenarios and a similar comparison with KL_p are included in Appendix D.)

Other metrics for calibration include the maximum calibration error (MCE) defined as

$$\text{MCE} = \max_{b=1,\ldots,B} \left| p_{\text{emp}}^{(b)} - q^{(b)} \right|,$$

and the Kolmogorov-Smirnov error (KS-error) (Gupta et al., 2021), a metric based on the cumulative distribution function, which is described in more detail in Appendix B.

Brier score. Crucially, a model without any discriminative power can be perfectly calibrated (see Figure 2b). The Brier score is a metric designed to evaluate both calibration and discriminative ability. It is the mean squared error between the predicted probability and the observed outcomes:

$$\text{Brier} = \frac{1}{N} \sum_i (\hat{p}_i - y_i)^2. \tag{6}$$

This score can be decomposed into two terms associated to calibration and discrimination ability, as shown in Appendix C. Using the synthetic data in Section 6.1, where the ground-truth probabilities are known, we show that Brier score is indeed a reliable proxy for *gold-standard* MSE metric based on ground-truth probabilities MSE_p, in contrast to calibration metrics such as ECE, MCE or KS-error, and to classification metrics such as AUC (see Figure 3 and Appendix D).

4 Proposed method: Calibrated Probability Estimation (CaPE)

Prediction models based on deep learning are typically trained by minimizing the cross entropy between the model output and the training labels (Goodfellow et al., 2016). This cost function is a *proper scoring rule*, which means that it evaluates probability estimates in a consistent manner and is therefore guaranteed to be well calibrated in an infinite-data regime (Buja et al., 2005), as illustrated by Figure 4 (first column).

Unfortunately, in practice prediction models are trained on finite data. This is crucial in the case of deep neural networks, because these models are highly overparametrized and therefore prone to overfitting (Goodfellow et al., 2016). In classification, networks have been shown to be capable of fitting arbitrary random labels (Zhang et al., 2017a). In probability estimation, we observe that neural networks indeed eventually overfit the observed outcomes completely. Moreover, the estimated probabilities collapse to 0 or 1 (Figure 4, second column), a phenomenon that has also been reported in classification (Mukhoti et al., 2020). We observe however, that calibration is preserved during the first stages of training (Figure 4, third column). This is reminiscent of the *early-learning* phenomenon observed for classification from partially corrupted labels (Yao et al., 2020; Xia et al., 2020), where neural networks learn from the correct labels before eventually overfitting the false ones (Liu et al., 2020).

Here, we propose to exploit the training dynamics of cross-entropy minimization through a method that we name *Calibrated Probability Estimation* (CaPE). Our starting point is a model obtained via early stopping using validation data on the cross-entropy loss. CaPE is designed to further

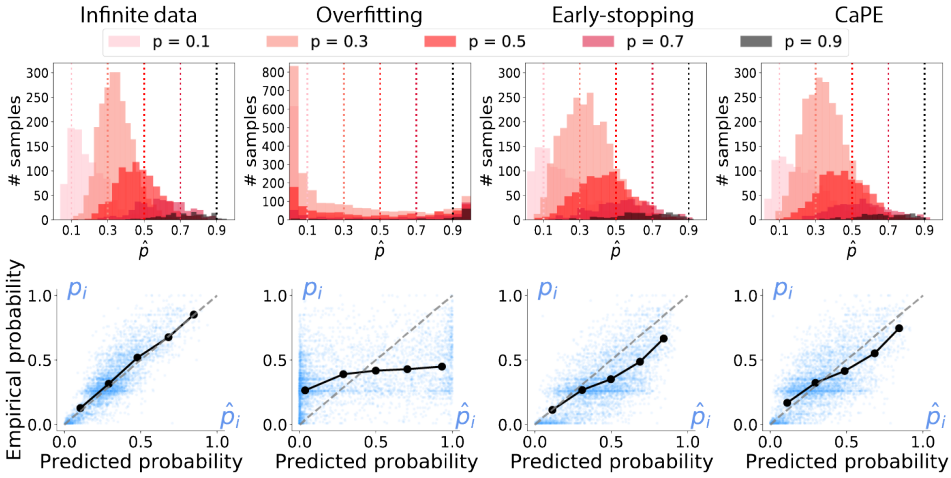


Figure 4: **Miscalibration due to overfitting and how to avoid it.** When trained on *infinite data* (i.e. resampling outcomes at each epoch according to ground-truth probabilities), neural networks trained by minimizing cross-entropy are well calibrated (first column). In the synthetic *discrete* scenario (see Section 6.1) the estimates are centered around the corresponding ground-truth probabilities (top). In the *uniform* scenario the reliability diagram shows that estimated probabilities match empirical probabilities of held-out outcomes (bottom, dashed line indicates perfect calibration). However, when trained on fixed observed outcomes, the model eventually overfits and the probabilities collapse to either 0 or 1 (second column). This is mitigated via early stopping, which yields relatively good calibration (third column). The proposed Calibration Probability Estimation (CaPE) method exploits this to further improve the model while ensuring that the output remains well calibrated. Plots for all synthetic data scenarios can be found in Appendix A.3, Figure 8.

improve the discrimination ability of the model, while ensuring that it remains well calibrated. This is achieved by alternatively minimizing the following two loss functions:

- **Discrimination loss**: Cross entropy between the model output and the observed binary outcomes,

$$\mathcal{L}_{\text{D}} = -\sum_{i=1}^{n} y_i \log(f(\boldsymbol{x}_i)).$$

- **Calibration loss**: Cross entropy between the output probability of the model and the empirical probability of the outcomes conditioned on the model's output:

$$\mathcal{L}_{\text{C}} = -\sum_{i=1}^{n} \left[p^i_{\text{emp}} \log(f(\boldsymbol{x}_i)) + (1 - p^i_{\text{emp}}) \log(1 - f(\boldsymbol{x}_i)) \right],$$

  where p^i_{emp} is an estimate of the conditional probability $\mathbb{P}[y = 1 | f(\boldsymbol{x}) \in I(f(\boldsymbol{x}_i))]$ where $I(f(\boldsymbol{x}_i))$ is a small interval centered at $f(\boldsymbol{x}_i)$. As explained in Section 3 if $f(x_i)$ is close to this value, then the model is well calibrated. We consider two approaches for estimating p^i_{emp}. (1) CaPE (bin) where we divide the training set into bins, select the bin b_i containing $f(x_i)$ and set $p^i_{\text{emp}} = p^{(b_i)}_{\text{emp}}$ in equation 3. (2) CaPE (kernel) where p^i_{emp} is estimated through a moving average with a kernel function (see Appendix E for more details). Both methods are efficiently computed by sorting the predicitions $\hat{p}_i$.

CaPE is summarized in Algorithm 1. Figure 4 shows that incorporating the calibration-loss minimization step indeed preserves calibration as training proceeds (note that this is not completely obvious because CaPE minimizes a calibration loss *on the training data*), and prevents the model from overfitting the observed outputs. This is beneficial also for the discriminative ability of the model, because it enables it to further reduce the cross-entropy loss without overfitting, as shown in Figure 5. The experiments with synthetic and real-world data reported in Section 6 suggest that this approach results in accurate probability estimates across a variety of realistic scenarios.

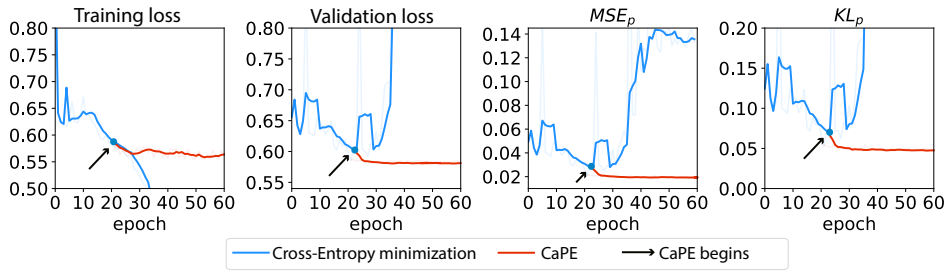


Figure 5: **Calibrated Probability Estimation prevents overfitting.** Comparison between the learning curves of cross-entropy (CE) minimization and the proposed calibrated probability estimation (CaPE), smoothed with a 5-epoch moving average. After an *early-learning* stage where both training and validation losses decrease, CE minimization overfits (first and second graph), with disastrous consequences in terms of probability estimation (third and fourth graph). In contrast, CaPE prevents overfitting, continuing to improve the model while maintaining calibration (see Figure 4).

Algorithm 1 Pseudocode for CaPE

Require: f ▷ early stopped model
Require: m ▷ freq. of training with $\mathcal{L}_C$
Require: $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ ▷ training set
Require: $K(p,q) := \exp\left[-(p-q)^2/\sigma^2\right]$ ▷ Gaussian kernel
  for $t = 1$ to `num_epochs` **do**
    if $t \mod m = 0$ **then**
      $\hat{p}_i \leftarrow f(\boldsymbol{x}_i), \forall i$
      Update $p^i_{\text{emp}}, \forall i$, with BIN or KERNEL
      $\mathcal{L} \leftarrow \mathcal{L}_C$ ▷ compute discrimination loss
    else
      $\mathcal{L} \leftarrow \mathcal{L}_D$ ▷ compute calibration loss
    end if
    $f \leftarrow$ backprop with $\mathcal{L}$ ▷ train with loss
  end for

function BIN(B) ▷ B-number of bins
  $I_1, \cdots I_B \leftarrow$ partitions by quantile of $\{\hat{p}_i\}_{i=1}^N$
  Find b such that $\hat{p}_i \in I_b$
  Index$(I_b) \leftarrow \{j | \hat{p}_j \in I_b\}$ ▷ get indices in bin b
  $p^i_{\text{emp}} \leftarrow \frac{1}{|I_b|} \sum_{i \in \text{Index}(I_b)} y_i$ ▷ empirical mean of bin b
end function

function KERNEL(r, K) ▷ r-window size; kernel
  $N_r(i) \leftarrow r$-nearest neighbor of $\hat{p}_i$
  $Z \leftarrow \sum_{\hat{p}_j \in N_r(i)} K(\hat{p}_i, \hat{p}_j)$ ▷ normalization factor
  $p^i_{\text{emp}} \leftarrow \sum_{\hat{p}_j \in N_r(i)} K(\hat{p}_i, \hat{p}_j)\, y_j / Z$ ▷ kernel smooth
end function

5 Related Work

Neural networks trained for classification often generate a probability associated with their prediction which quantifies its uncertainty. These estimates are often found to be inaccurate (Mukhoti et al., 2020; Guo et al., 2017). Techniques mitigating this issue are often described as calibration methods, and broadly fall into three categories depending on whether they: (1) postprocess the outputs of a trained model, (2) combine multiple model outputs, or (3) modify the training process.

Post-processing methods transform the output probabilities in order to improve calibration on held-out data (Zadrozny & Elkan, 2001; Gupta et al., 2020; Kull et al., 2017; 2019). For example, Platt scaling (Platt, 1999) fits a logistic function that minimizes the negative log-likelihood loss. Temperature scaling (Guo et al., 2017) does the same with a temperature parameter augmenting the softmax function. In contrast to these methods, CaPE enforces calibration *during training*, which has the advantage of enabling further improvements in the discriminative abilities of the model.

Ensembling methods combine multiple models to improve generalization. Mix-n-Match (Zhang et al., 2020) trains the network only once, and provides ensemble predictions using multiple temperature scaling transformations. In contrast, methods like (Lakshminarayanan et al., 2017; Maddox et al., 2019) train and ensemble multiple neural networks using different initializations. These approaches are compatible with the proposed method CaPE and other techniques; how to combine them effectively is an interesting direction for future research.

Modified training methods can be divided into two groups. The first group smooths the target 0/1 labels in order to prevent output estimates from collapsing to 0/1 (Mukhoti et al., 2020; Szegedy et al., 2016; Zhang et al., 2018; Thulasidasan et al., 2020). The second group, attaches additional calibration penalties to a cross entropy loss (Kumar et al., 2018; Pereyra et al., 2017; Liang et al., 2020). CaPE is most similar in spirit to the latter methods, although its data-driven calibration loss is different to the penalties used in these techniques.

Methods	*Uniform*		*Sigmoid*		*Centered*		*Skewed*		*Discrete*	
$(\times 10^{-2})$	MSE_p	KL_p	MSE_p	KL_p	MSE_p	KL_p	MSE_p	KL_p	MSE_p	KL_p
Infinite Data*	1.13	2.81	5.35	14.86	0.20	0.41	0.22	0.92	1.52	3.64
CE early-stop	4.21	10.93	6.16	17.16	0.48	0.98	0.40	1.79	2.24	5.26
Temperature	2.71	6.70	6.13	17.05	0.48	0.98	0.40	1.75	2.20	5.12
Platt Scaling	2.48	6.06	5.78	16.12	0.41	0.83	**0.39**	1.71	2.06	4.82
Dirichlet Cal.	3.56	9.06	8.71	25.33	0.46	0.94	0.48	2.31	2.74	6.52
Mix-n-match	2.69	6.70	6.13	17.10	0.48	0.98	**0.39**	1.74	2.20	5.12
Focal Loss	4.13	10.51	6.89	19.51	0.48	0.97	1.27	11.61	2.92	6.76
Entropy Reg.	2.84	7.37	7.03	21.19	0.42	0.87	1.17	10.57	2.84	6.62
MMCE Reg.	2.22	5.65	5.33	15.03	0.44	0.90	0.54	2.43	2.08	4.90
Deep Ensemble	1.90	4.55	5.85	16.43	0.44	0.89	0.55	2.58	1.97	4.61
CaPE (bin)	1.83	4.46	5.29	14.59	**0.38**	**0.78**	0.40	1.72	**1.83**	**4.31**
CaPE (kernel)	**1.81**	**4.41**	**5.22**	**14.47**	0.40	0.81	**0.39**	**1.70**	1.85	4.36

Table 1: Results on synthetic data. The full table with confidence interval of 1000 bootstrap is in Appendix A.1. * is a model trained with infinite data obtained by continuous label resampling.

Datasets for evaluation The methods discussed in this section were developed for calibration in context of classification, and were tested on datasets such as CIFAR-10/100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011), and ImageNet (Deng et al., 2009) where the relationship between labels and input data is completely deterministic. Here, we evaluate these methods on synthetic and real-world probability-estimation problems with inherent uncertainty.

6 EXPERIMENTS

6.1 SYNTHETIC DATASET: FACE-BASED RISK PREDICTION

To benchmark the performance of neural networks on the task of probability estimation, we build a synthetic dataset based on UTKFace (Zhang et al., 2017b), which contains faces images and associated ages. We use the age of the ith person z_i to assign them a risk of contracting a disease $p_i = \psi(z_i)$ for a fixed function $\psi : \mathbb{N} \to [0, 1]$. Then we simulate whether the person actually contracts the illness by assigning it a label $y_i = 1$ with probability p_i, and $y_i = 0$ otherwise. The probability-estimation task is to estimate the ground-truth probability p_i from the face image x_i, which requires learning to discriminate age and map it to the corresponding risk.

We design the risk-assignment function ψ in order to create five scenarios, inspired partly by real-world datasets (Further discussion of the probability label generation under different scenarios can be found in Appendix G):

- **Uniform**: Equally-spaced, inspired by weather forecasting: $\psi(z) = z/100$
- **Sigmoid**: Concentrated near two extremes: $\psi(z) = \sigma(25(z/100 - 0.29))$
- **Skewed**: Clustered close to zero, inspired by vehicle-collision detection: $\psi(z) = z/250$
- **Centered**: Clustered in the center, inspired by cancer-survival prediction: $\psi(z) = z/300 + 0.35$
- **Discrete**: Discretized: $\psi(z) = 0.2\left[\mathbb{1}_{\{z>20\}} + \mathbb{1}_{\{z>40\}} + \mathbb{1}_{\{z>60\}} + \mathbb{1}_{\{z>80\}}\right] + 0.1$

6.2 REAL-WORLD DATASETS

We propose to use three open-source, real-world datasets to benchmark probability-estimation approaches (see Appendix H for further details on the datasets and experiments).

Survival of Cancer Patients. Histopathological features are useful in identification of tumor cells, cancer subtypes, and the stage and level of differentiation of cancer. Hematoxylin and Eosin (H&E)-stained slides are the most common type of histopathology data and the basis for clinical decision making. In particular, they can be used for survival prediction (Wulczyn et al., 2020), which is critical in evaluating the prognosis of patients. However, the treatments assigned to patients after diagnosis are not personalized and their impact on cancer trajectory is complex, so the survival status of a patient is not deterministic. In this work, we use the H&E slides of non-small cell lung cancers

Method	Cancer Survival			Weather forecasting			Collision Prediction		
$(\times 10^{-2})$	AUC	ECE	Brier	AUC	ECE	Brier	AUC	ECE	Brier
CE early-stop	58.88	12.25	23.96	77.64	10.91	20.57	85.68	4.36	8.59
Temperature	58.88	12.07	23.73	77.64	8.66	20.21	85.68	4.56	8.51
Platt Scaling	58.91	10.28	23.33	77.65	6.97	19.53	85.76	3.04	8.23
Dirichlet Cal.	49.89	13.83	24.08	77.51	14.29	21.89	83.36	5.78	8.78
Mix-n-match	58.88	12.16	23.67	77.64	8.65	20.21	85.68	4.40	8.52
Focal Loss	55.02	12.15	23.31	76.18	8.32	20.27	82.21	9.07	9.82
Entropy Reg.	56.29	11.73	23.62	79.01	10.53	19.77	83.15	14.54	11.10
MMCE Reg.	48.45	11.84	23.73	76.69	8.46	20.12	85.18	**2.94**	8.48
Deep Ensemble	52.46	9.99	23.47	**79.86**	7.41	18.82	85.27	3.15	8.55
CaPE (bin)	**61.44**	12.31	23.20	78.99	5.16	**18.37**	85.70	3.16	8.18
CaPE (kernel)	61.22	**9.48**	**23.18**	79.00	**5.08**	18.39	**85.95**	3.22	**8.13**

Table 2: Results on cancer-survival prediction, weather forecasting, and collision prediction. The full tables with all the metrics described in Section 3 are in Appendix A.2

from The Cancer Genome Atlas Program (TCGA)[1] to estimate the the 5-year survival probability of cancer patients. The outcome distribution is similar to the *Centered* scenario in our synthetic data.

Weather Forecasting The atmosphere is governed by nonlinear dynamics, hence weather forecast models possess inherent uncertainties (Richardson, 2007). Nowcasting is the problem of weather prediction in the near future, up to a few hours. It is of great operational significance, especially with increasing number of extreme inclement weather conditions (Agrawal et al., 2019; Ravuri et al., 2021). We use the German Weather service dataset[2], which contains quality-controlled rainfall-depth composites from 17 operational Doppler radars. We use 30 minutes of precipitation data to predict if the mean precipitation will increase or decrease after one hour. Three precipitation maps from the past 30 minutes serve as an input. The outcome distribution is similar to the *Uniform* scenario in our synthetic data.

Collision Prediction Vehicle collision is one of the leading causes of death in the world. Reliable collision prediction systems are therefore instrumental in saving many lives. These systems often use video feed from the dashcam camera, and predict potential collisions a few seconds into the future. The potential for collision is affected simultaneously by many unknown factors, and hence is not deterministic. Following Kim et al. (2019), we use about 0.3 seconds of real dash cam videos from `YouTubeCrash` dataset as input, and predict the probability of a collision happening (label 1) in the next 2 seconds. The dataset is severely imbalanced as the number of collisions is very low, so outcome distribution is similar to the *Skewed* scenario in our synthetic data.

6.3 BASELINES

We adapt existing calibration methods developed for classification as baselines for probability estimation. In particular, we compare CaPE with (1) **Best CE Model**: the network with set of parameters that achieved lowest cross entropy loss on validation set, (2) **Three post-processing methods**: Temperature Scaling (Guo et al., 2017), Platt Scaling (Platt, 1999), and Dirichlet Calibration (Kull et al., 2019) applied to the best CE model, (3) **Two Ensemble Methods**: Mix-n-Match (Zhang et al., 2020) applied to best CE model, and Deep Ensemble (Lakshminarayanan et al., 2017), and (4) **Three Modified Training methods**: focal loss (Mukhoti et al., 2020), entropy-maximizing loss (Pereyra et al., 2017), and MMCE regularization (Kumar et al., 2018). They are also chosen by best validation loss (details of these algorithms are listed in Appendix F). For our experiments on synthetic data, we also compare against a model trained on an **infinite amount of data** by repeatedly sampling new outcomes from the ground-truth probabilities at each epoch. This model provides a best-case reference of what can be achieved in each scenario.

[1] `https://www.cancer.gov/tcga`

[2] `https://opendata.dwd.de/weather/radar/`

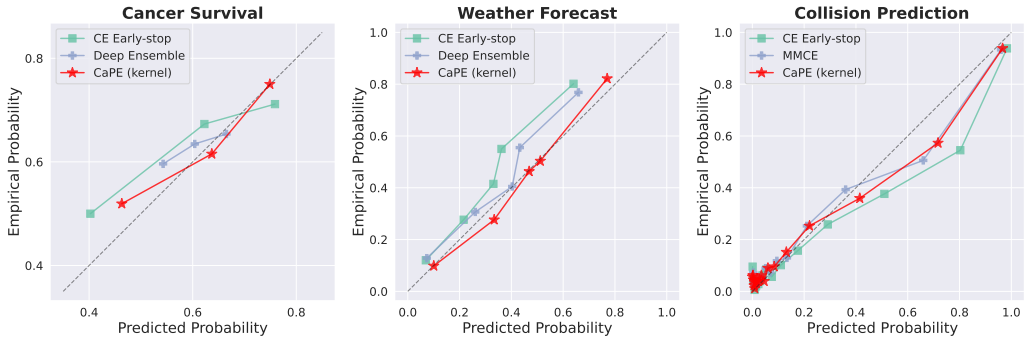


Figure 6: **Reliability diagrams for real-world data.** Reliability diagrams (Section 3), computed on held-out data for cross-entropy minimization with early stopping, the proposed method CaPE and the best baseline for each dataset. CaPE produces a better calibrated output. The remaining diagrams are included in Appendix A.3.

7 Results and Discussion

Table 1 shows that calibration methods developed for classification can be effective for probability estimation. However, the performance of some methods is not consistent across all scenarios. For instance, regularization with negative entropy, which penalizes very high/low confidence, performs worse than CE when the ground-truth probability is close to 0 or 1. In contrast, methods that do not make strong assumptions tend to generalize better to multiple scenarios (e.g. Platt scaling consistently beats CE). The proposed method CaPE outperforms other techniques in most scenarios, and even matches the performance of the *infinite data* baseline for the *Sigmoid* scenario. Finally, we observe that the *Skewed* scenario is very challenging: most methods barely improve the CE baseline.

Table 2 compares the baseline methods and CaPE on the three real-world datasets. We present AUC, ECE for 15 equally-sized bins, and Brier score, as complementary metrics since the underlying ground-truth probabilities are unobserved. As discussed in Section 3, Brier score is the metric that best captures the quality of probability estimates. CaPE has the lowest Brier score in all three datasets, while also achieving lower ECE values and higher AUC values than the other methods. This demonstrates that enforcing calibration during training also yields a more discriminative model. The reliability diagrams in Figure 6 depict the probability estimates produced by CE, CaPE and the best baseline method on the three datasets, further demonstrating that CaPE produces a well-calibrated output.

Figure 6 also shows that each real-world dataset closely aligns with a particular synthetic scenario: cancer survival with *Centered*; weather forecasting with *Uniform*; collision prediction with *Skewed*. This supports the significance of our synthetic benchmark dataset, and provides insights in the differences among baseline models. For example, model averaging with deep ensemble performs well on weather forecasting but has higher Brier scores than Platt scaling on the other two datasets (a detailed case-study in Appendix I). Accordingly, deep ensemble also underperforms in the synthetic scenarios where ground-truth probabilities are clustered closely (*Sigmoid*, *Uniform*), but is effective for *Uniform*. Finally, as in the synthetic *Skewed* scenario, all methods had similar performance on the collision prediction task. This highlights the importance of considering different scenarios when evaluation methodology for probability estimation.

8 Conclusion

In this work we evaluate existing approaches to improve the output probabilities of neural networks on probability-estimation problems. To this end, we introduce a new synthetic benchmark dataset designed to reproduce several realistic scenarios, and also gather three real-world datasets relevant to medicine, climatology, and self-driving cars. In addition, we propose a novel approach for probability-estimation via deep learning that outperforms existing approaches for most datasets. We hope that this work will motivate further advances in probability estimation.

REFERENCES

Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images. *arXiv preprint arXiv:1912.12132*, 2019.

Georgy Ayzel. Rainnet: a convolutional neural network for radar-based precipitation nowcasting. `https://github.com/hydrogo/rainnet`, 2020.

Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications," manuscript, available at www-stat.wharton.upenn.edu/ buja, 2005.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. URL `https://arxiv.org/abs/2003.04297`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

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

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. *arXiv preprint arXiv:2006.12800*, 2020.

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=eQe8DEWNN2W`.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2127–2136. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ilse18a.html`.

Hoon Kim, Kangwook Lee, Gyeongjo Hwang, and Changho Suh. Crash to not crash: Learn to identify dangerous vehicles using a simulator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 978–985, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Meelis Kull, Telmo M. Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052 – 5080, 2017. doi: 10.1214/17-EJS1338SI. URL `https://doi.org/10.1214/17-EJS1338SI`.

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. In *NeurIPS*, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. In *British Machine Vision Conference (BMVC)*, 2020.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *CoRR*, abs/1902.02476, 2019. URL `http://arxiv.org/abs/1902.02476`.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, S. Golodetz, Philip H. S. Torr, and P. Dokania. Calibrating deep neural networks using focal loss. *ArXiv*, abs/2002.09437, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011. URL `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. *CoRR*, abs/1701.06548, 2017. URL `http://arxiv.org/abs/1701.06548`.

J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3), 1999.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skillful precipitation nowcasting using deep generative models of radar. *arXiv preprint arXiv:2104.00954*, 2021.

Lewis Fry Richardson. *Weather prediction by numerical process*. Cambridge university press, 2007.

Christian Szegedy, V. Vanhoucke, S. Ioffe, Jonathon Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks, 2020.

Ellery Wulczyn, David F. Steiner, Zhaoyang Xu, Apaar Sadhwani, Hongwu Wang, I. Flament, C. Mermel, Po-Hsuan Cameron Chen, Yun Liu, and Martin C. Stumpe. Deep learning-based survival prediction for multiple cancer types using histopathology images. *PLoS ONE*, 15, 2020.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2020.

Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. Searching to exploit memorization effect in learning with noisy labels. In *International Conference on Machine Learning*, pp. 10789–10798. PMLR, 2020.

B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017a.

Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412, 2018.

Jize Zhang, B. Kailkhura, and T. Y. Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *ICML*, 2020.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017b.

A ADDITIONAL RESULTS

We present here supplementary results to the ones presented in Section 7.

A.1 FACE-BASED RISK PREDICTION

Full evaluation with confidence intervals derived using 1000 bootstrap in five simulated scenarios, *Uniform* (Table.3), *Sigmoid* (Table.4), *Centered* (Table.5), *Skewed* (Table.6), and *Discrete* (Table.7), are examined. Note that all numbers are downscaled by 10^{-2} in the tables.

Uniform	ECE	MCE	KS	Brier	MSE_p	KL_p
Infinite Data	4.14±0.81	12.07±3.29	2.24±0.88	18.97±0.33	1.14±0.04	2.82±0.11
CE early-stop	12.32±0.83	21.79±1.97	12.16±0.83	21.82±0.51	4.21±0.15	10.94±0.36
Temperature	5.7±0.74	13.82±2.71	2.36±0.74	20.47±0.37	2.73±0.11	6.75±0.25
Platt Scaling	4.29±0.77	10.94±2.55	1.3±0.45	20.18±0.36	2.48±0.09	6.07±0.22
Dirichlet Cal.	7.38±1.12	22.58±7.24	3.78±0.46	21.32±0.33	3.56±0.13	9.08±0.29
Focal Loss	5.34±0.68	13.31±2.67	3.56±0.85	21.99±0.28	4.13±0.11	10.52±0.28
Mix-n-match	5.46±0.84	12.9±2.51	1.92±0.44	20.43±0.35	2.7±0.11	6.72±0.24
Entropy Reg.	5.7±0.74	13.52±1.67	4.94±0.9	20.42±0.3	2.58±0.09	6.65±0.21
MMCE Reg.	4.89±0.74	12.57±2.27	1.92±0.46	20.04±0.38	2.24±0.08	5.68±0.2
Deep Ensemble	4.26±0.72	11.33±2.38	1.95±0.61	19.88±0.32	1.9±0.07	4.55±0.18
CaPE (bin)	4.58±0.75	11.85±2.49	1.71±0.51	19.68±0.36	1.78±0.07	4.35±0.16
CaPE (kernel)	4.62±0.62	12.25±2.31	1.65±0.38	19.71±0.34	1.74±0.07	4.3±0.17

Table 3: Performance on Face-based Risk Prediction *Uniform*.

Sigmoid	ECE	MCE	KS	Brier	MSE_p	KL_p
Infinite Data	6.4±0.71	20.63±3.44	2.74±0.45	16.28±0.44	5.34±0.2	14.82±0.51
CE early-stop	6.19±0.75	17.0±3.68	5.86±0.8	16.68±0.42	6.16±0.17	17.16±0.48
Temperature	5.57±0.71	15.32±3.09	5.02±0.83	16.58±0.34	6.13±0.17	17.09±0.43
Platt Scaling	3.45±0.68	10.32±2.79	1.3±0.43	16.33±0.34	5.78±0.19	16.15±0.47
Dirichlet Cal.	14.5±1.15	25.68±3.02	4.67±0.32	19.21±0.43	8.64±0.26	25.18±0.58
Focal Loss	4.65±0.7	11.84±2.78	2.66±0.77	16.96±0.34	6.86±0.21	19.46±0.5
Mix-n-match	5.65±0.76	15.32±3.41	5.09±0.94	16.6±0.36	6.12±0.17	17.08±0.46
Entropy Reg.	9.51±0.79	18.77±2.38	7.26±0.78	17.17±0.31	7.02±0.17	21.16±0.42
MMCE Reg.	4.67±0.76	13.63±2.59	2.5±0.53	15.9±0.51	5.35±0.18	15.06±0.49
Deep Ensemble	5.17±0.74	16.12±3.11	2.04±0.44	16.39±0.45	5.86±0.22	16.46±0.6
CaPE (bin)	3.78±0.6	11.96±2.59	2.22±0.7	15.84±0.43	5.17±0.2	14.27±0.49
CaPE (kernel)	3.9±0.75	11.73±2.79	2.05±0.54	15.85±0.41	5.16±0.2	14.34±0.49

Table 4: Performance on Face-based Risk Prediction *Sigmoid*.

Centered	ECE	MCE	KS	Brier	MSE_p	KL_p
Infinite Data	4.29±0.74	12.38±2.92	2.68±0.8	24.22±0.13	0.2±0.01	0.41±0.01
CE early-stop	5.76±0.84	15.32±3.07	4.19±1.02	24.68±0.08	0.48±0.01	0.98±0.03
Temperature	6.09±0.82	15.83±2.91	4.74±0.96	24.74±0.06	0.48±0.01	0.98±0.03
Platt Scaling	4.57±0.76	11.85±2.5	2.79±0.85	24.62±0.08	0.41±0.01	0.83±0.03
Dirichlet Cal.	4.84±1.15	13.13±7.61	2.16±0.86	24.7±0.1	0.46±0.01	0.94±0.03
Mix-n-match	6.05±0.83	15.71±2.92	4.68±0.98	24.74±0.06	0.48±0.01	0.98±0.02
Focal Loss	5.09±0.83	13.4±2.87	3.44±1.02	24.8±0.05	0.48±0.01	0.97±0.03
Entropy Reg.	5.02±0.86	12.69±3.42	3.27±0.96	24.74±0.06	0.45±0.01	0.92±0.03
MMCE Reg.	5.56±0.86	13.59±2.65	2.71±0.93	24.7±0.08	0.44±0.01	0.9±0.03
Deep Ensemble	4.84±0.78	12.39±2.52	2.64±0.71	24.69±0.07	0.44±0.01	0.89±0.03
CaPE (bin)	4.73±0.82	11.81±2.54	2.07±0.6	24.56±0.11	0.38±0.01	0.78±0.03
CaPE (kernel)	5.41±0.87	12.71±2.5	2.39±0.78	24.59±0.11	0.4±0.01	0.81±0.03

Table 5: Performance on Face-based Risk Prediction *Centered*.

Skewed	ECE	MCE	KS	Brier	MSE_p	KL_p
Infinite Data	2.7±0.46	7.64±2.14	1.05±0.39	11.0±0.51	0.22±0.01	0.92±0.03
CE early-stop	3.07±0.57	7.88±1.88	1.28±0.41	11.18±0.5	0.4±0.01	1.79±0.06
Temperature	3.14±0.49	7.92±1.84	1.12±0.33	11.22±0.47	0.4±0.02	1.76±0.06
Platt Scaling	2.99±0.53	7.73±1.59	1.07±0.37	11.1±0.54	0.39±0.01	1.72±0.06
Dirichlet Cal.	3.04±0.73	7.81±2.43	0.97±0.3	11.22±0.42	0.47±0.02	2.31±0.07
Focal Loss	8.29±0.67	14.93±1.43	6.16±0.67	12.01±0.41	1.28±0.03	1.63±0.66
Mix-n-match	2.99±0.53	7.78±1.78	1.08±0.32	11.18±0.49	0.4±0.01	1.75±0.05
Entropy Reg.	7.67±0.57	14.43±1.5	5.2±0.71	11.94±0.45	1.18±0.03	10.74±0.65
MMCE Reg.	3.68±0.59	10.94±2.76	1.47±0.31	11.14±0.44	0.54±0.02	2.44±0.08
Deep Ensemble	2.87±0.5	7.21±1.63	1.36±0.44	11.28±0.5	0.55±0.02	2.58±0.07
CaPE (bin)	3.29±0.5	8.18±1.51	1.17±0.34	11.07±0.47	0.4±0.02	1.73±0.06
CaPE (kernel)	3.16±0.5	8.14±1.58	1.09±0.33	11.17±0.53	0.39±0.01	1.69±0.06

Table 6: Performance on Face-based Risk Prediction *Skewed*.

Discrete	ECE	MCE	KS	Brier	MSE_p	KL_p
Infinite Data	4.23±0.74	11.16±2.5	1.45±0.49	20.38±0.35	1.52±0.05	3.63±0.12
CE early-stop	6.7±0.86	18.62±3.52	2.61±0.53	21.91±0.36	2.24±0.08	5.27±0.17
Temperature	6.12±0.87	16.82±3.56	3.37±0.86	21.76±0.35	2.21±0.08	5.15±0.18
Platt Scaling	4.7±0.72	11.69±2.44	1.67±0.51	21.44±0.32	2.06±0.08	4.83±0.17
Dirichlet Cal.	7.13±0.86	22.67±5.08	3.18±0.68	22.1±0.34	2.74±0.1	6.53±0.22
Focal Loss	5.7±0.75	13.68±2.32	4.62±0.91	21.77±0.28	2.92±0.09	6.77±0.21
Mix-n-match	6.27±0.76	16.83±2.95	3.47±0.93	21.77±0.33	2.21±0.08	5.14±0.18
Entropy Reg.	6.69±0.87	15.38±2.43	6.03±1.13	21.79±0.31	2.84±0.08	6.62±0.19
MMCE Reg.	3.96±0.7	10.4±2.4	1.51±0.47	21.12±0.35	2.09±0.08	4.92±0.18
Deep Ensemble	4.76±0.74	11.49±2.23	2.04±0.61	21.17±0.31	1.97±0.08	4.61±0.17
CaPE (bin)	5.41±0.74	14.45±3.15	2.24±0.59	21.33±0.36	1.81±0.08	4.28±0.18
CaPE (kernel)	4.96±0.8	12.97±2.63	2.18±0.58	21.21±0.42	1.84±0.08	4.35±0.17

Table 7: Performance on Face-based Risk Prediction *Discrete*.

A.2 SUPPLEMENTARY METRICS ON REAL-WORLD DATASET

We present here additional metrics on the real world data. Cancer Survival (Table 8), Climate Forecasting (Table 9), and Collision Prediction (Table 10).

Methods $(\times 10^{-2})$	AUC	ECE	MCE	NLL	Brier	KS
CE Early-stop	58.88	12.25	25.35	67.92	23.96	6.44
Temperature	58.88	12.07	24.65	67.11	23.73	6.92
Platt Scaling	58.91	10.28	27.69	66.11	23.33	4.91
Dirichlet Cal.	49.89	13.83	35.52	67.57	24.08	6.00
Mix-n-match	58.88	12.16	**24.52**	66.89	23.67	7.18
Focal loss	55.02	12.15	26.34	65.92	23.31	6.38
Entropy Reg.	56.29	11.73	30.81	66.49	23.62	6.83
MMCE Reg.	48.45	11.84	37.36	66.83	23.73	3.64
Deep Ensemble	52.26	9.99	28.30	66.22	23.47	5.02
CaPE (bin)	**61.44**	12.31	25.27	65.75	23.20	**2.59**
CaPE (kernel)	61.22	**9.48**	32.40	**65.70**	**23.18**	3.70

Table 8: Baselines with full metrics for cancer survival

Methods $(\times 10^{-2})$	AUC	ECE	MCE	NLL	Brier	KS
CE Early-stop	77.64	10.91	25.50	59.97	20.57	11.03
Temperature	77.64	8.66	23.56	58.77	20.21	7.41
Platt Scaling	77.65	6.97	16.47	57.38	19.53	3.26
Dirichlet Cal.	77.51	14.29	30.09	62.83	21.89	5.21
Mix-n-match	77.64	8.65	23.58	58.77	20.21	7.39
Focal Loss	76.18	8.32	21.25	59.01	20.27	4.45
Entropy Reg	79.01	10.53	20.72	57.83	19.77	5.00
MMCE Reg	76.69	8.46	19.73	59.25	20.12	7.31
Deep Ensemble	**79.86**	7.41	18.24	55.28	18.82	7.57
CaPE (bin)	78.99	5.16	15.09	**79.00**	**18.37**	**2.34**
CaPE (kernel)	79.00	**5.08**	**13.28**	54.32	18.39	**2.34**

Table 9: Baselines with full metrics for weather prediction

Methods $(\times 10^{-2})$	AUC	ECE	MCE	NLL	Brier	KS
CE Early-stop	85.68	4.36	19.87	31.67	8.59	1.54
Temperature	85.68	4.56	16.79	30.36	8.52	2.9
Platt Scaling	85.76	3.04	12.39	**29.42**	8.23	**1.52**
Dirichlet Cal.	83.36	5.78	18.13	30.90	8.77	1.60
Mix-n-match	85.68	4.40	17.41	30.25	8.52	2.60
Focal Loss	82.21	9.07	19.85	34.41	9.82	8.72
Entropy Reg	83.15	14.54	21.27	38.74	11.10	13.44
MMCE Reg.	85.18	**2.94**	**8.95**	30.65	8.48	2.44
Deep Ensemble	85.27	3.15	16.53	30.20	8.54	2.01
CaPE (bin)	8.57	3.16	12.21	30.61	8.18	2.13
CaPE (kernel)	**85.95**	3.22	13.32	30.44	**8.13**	2.10

Table 10: Baselines with full metrics for collision prediction

A.3 ADDITIONAL RELIABILITY DIAGRAM

We present here additional reliability curves to the ones illustrated in Figure 6

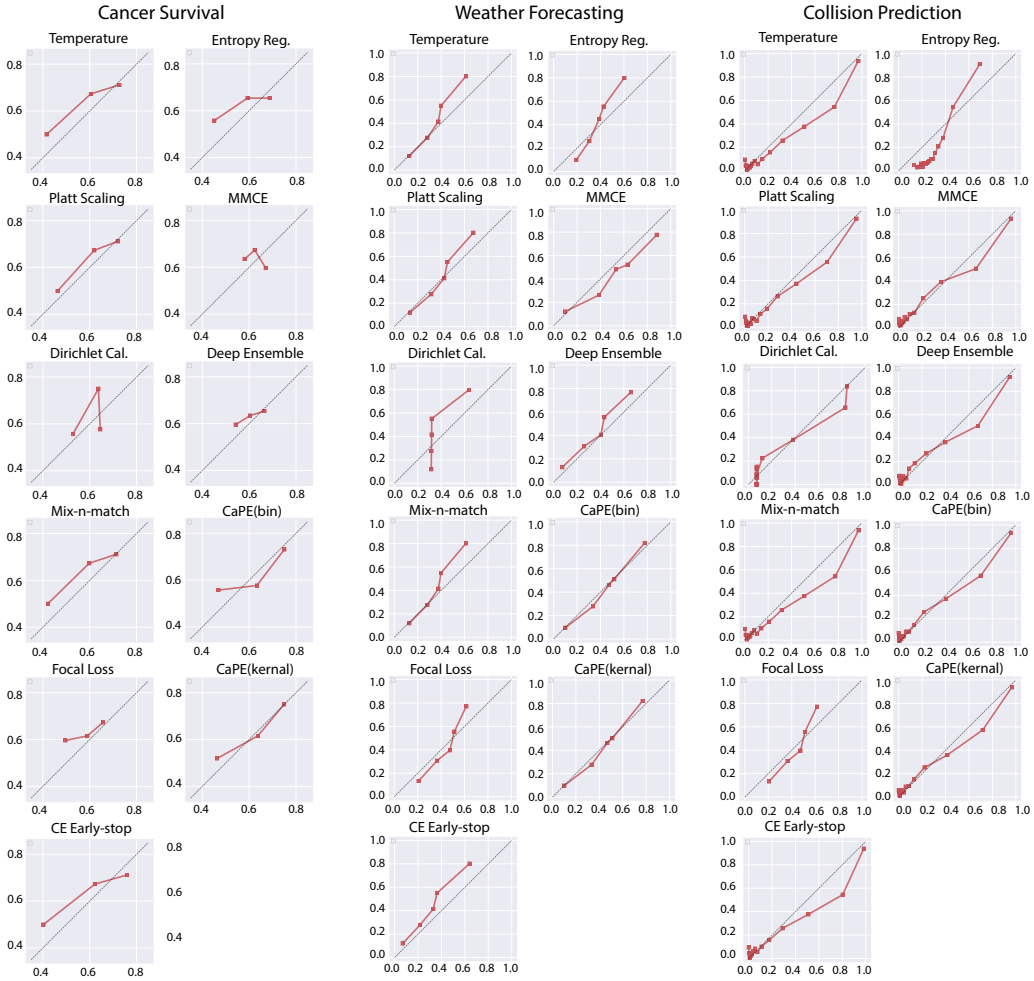


Figure 7: The reliability diagrams of all the baselines on real-world datasets. We implement all baseline methods to each of the task and plot the empirical probability(y-axis) against predicted probability(x-axis). The axis labels are removed due to space constraint.

We present here additional reliability curves for the different synthetic data scenarios, mentioned in Figure 4.

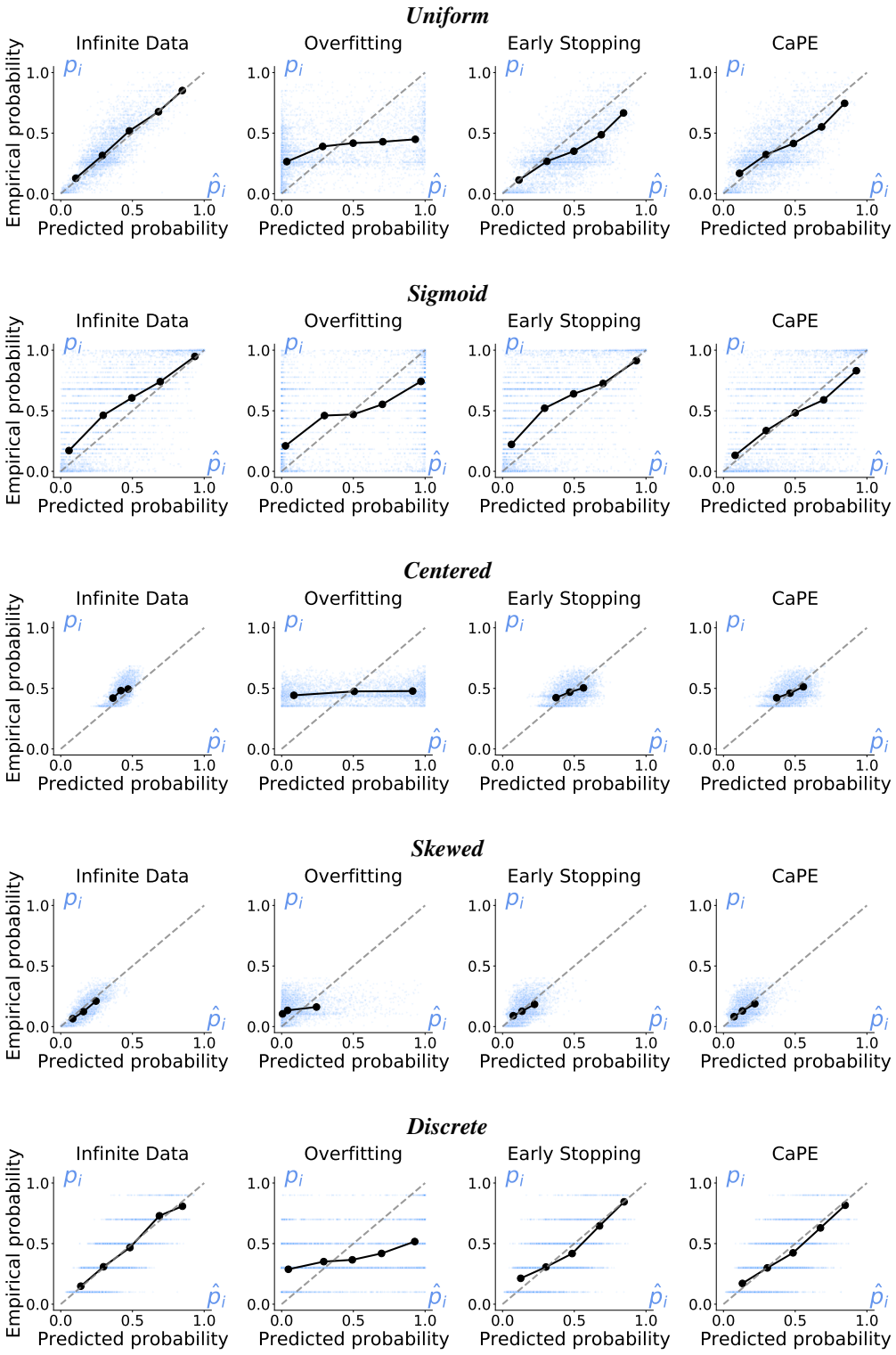


Figure 8: Miscalibration due to overfitting and how to avoid it in five scenarios from the synthetic data).

B KOLMOGOROV-SMIRNOV ERROR

We derive the KS-error, mentioned in Section3.

For a calibrated estimator

$$\mathbb{P}[y = 1 | f(\boldsymbol{x}) \in I(q)] = q, \;\; \forall 0 \leq q \leq 1,$$

for some small interval $I(q)$ around q. Hence

$$\mathbb{P}[y = 1, f(\boldsymbol{x}) \in I(q)] = \mathbb{P}[f(\boldsymbol{x}) \in I(q)]\, q, \;\; \forall 0 \leq q \leq 1$$

Similarly to Kolmogorov-Smirnov (KS) test for distribution functions, we can recast this property in integral form

$$\phi_1(\sigma) = \int_0^\sigma \mathbb{P}[y = 1, f(\boldsymbol{x}) \in I(q)] dq, \quad \phi_2(\sigma) = \int_0^\sigma \mathbb{P}[f(\boldsymbol{x}) \in I(q)]\, q dq$$

We can evaluate ϕ_1, ϕ_2 from a finite sample $(\boldsymbol{x}_i, y_i), i = 1 \ldots n$,

$$\phi_1(\sigma) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i = 1, f(\boldsymbol{x}_i) \leq \sigma), \quad \phi_2(\sigma) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(f(\boldsymbol{x}_i) \leq \sigma) f(\boldsymbol{x}_i)$$

The KS error is defined as

$$\text{KS} = \max_{1 \leq \sigma \leq 1} |\phi_1(\sigma) - \phi_2(\sigma)|$$

ϕ_1, ϕ_2 can be efficiently computed by sorting the data points with respect to their confidence scores $f(\boldsymbol{x}_i)$. The KS error has the advantage of being independent of binning configurations, unlike ECE and MCE.

C BRIER SCORE DECOMPOSITION

We present here a decomposition the Brier score into two components, discussed in Section 3.

The Brier score can be interpreted as a sum of two terms, calibration and refinement. Assume the network outputs K distinct possible predictions, i.e., $\hat{p} \in \{\hat{q}_1, \ldots, \hat{q}_K\}$.

Denote S_k, the set of all inputs with output p_k and $\bar{p}_k$ the empirical probability, i.e.,

$$S_k = \{\boldsymbol{x} | f(\boldsymbol{x}) = \hat{q}_k\}, \quad |S_k| = n_k, \quad \bar{q}_k = \frac{1}{n_k} \sum_{\boldsymbol{x}_i \in S_k} y_i$$

Then we can write

$$\text{Brier} = \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - y_i)^2 = \frac{1}{N} \sum_{k=1}^{K} n_k (\hat{q}_k - \bar{q}_k)^2 + \frac{1}{N} \sum_{k=1}^{K} n_k \bar{q}_k (1 - \bar{q}_k),$$

The first term on the RHS, calibration, is similar to MSE_p, with the true labels substituted by the empirical probabilities $\bar{p}_k$. The second term, refinement, is an estimate of the confidence in determining $\bar{p}_k$. It is related to the area under curve (AUC), which measures to the achievable accuracy of the network as a classifier. The term is smaller as the prediction classes f_i tend towards 0 or 1. Thus, this term penalizes empirically calibrated predictors, with low discriminative power, as in Figure 2b.

D METRIC COMPARISON

Various metrics, as well as KL-divergence are compared among all five scenarios in our Face-based Risk Prediction dataset, referred to in Section 3.

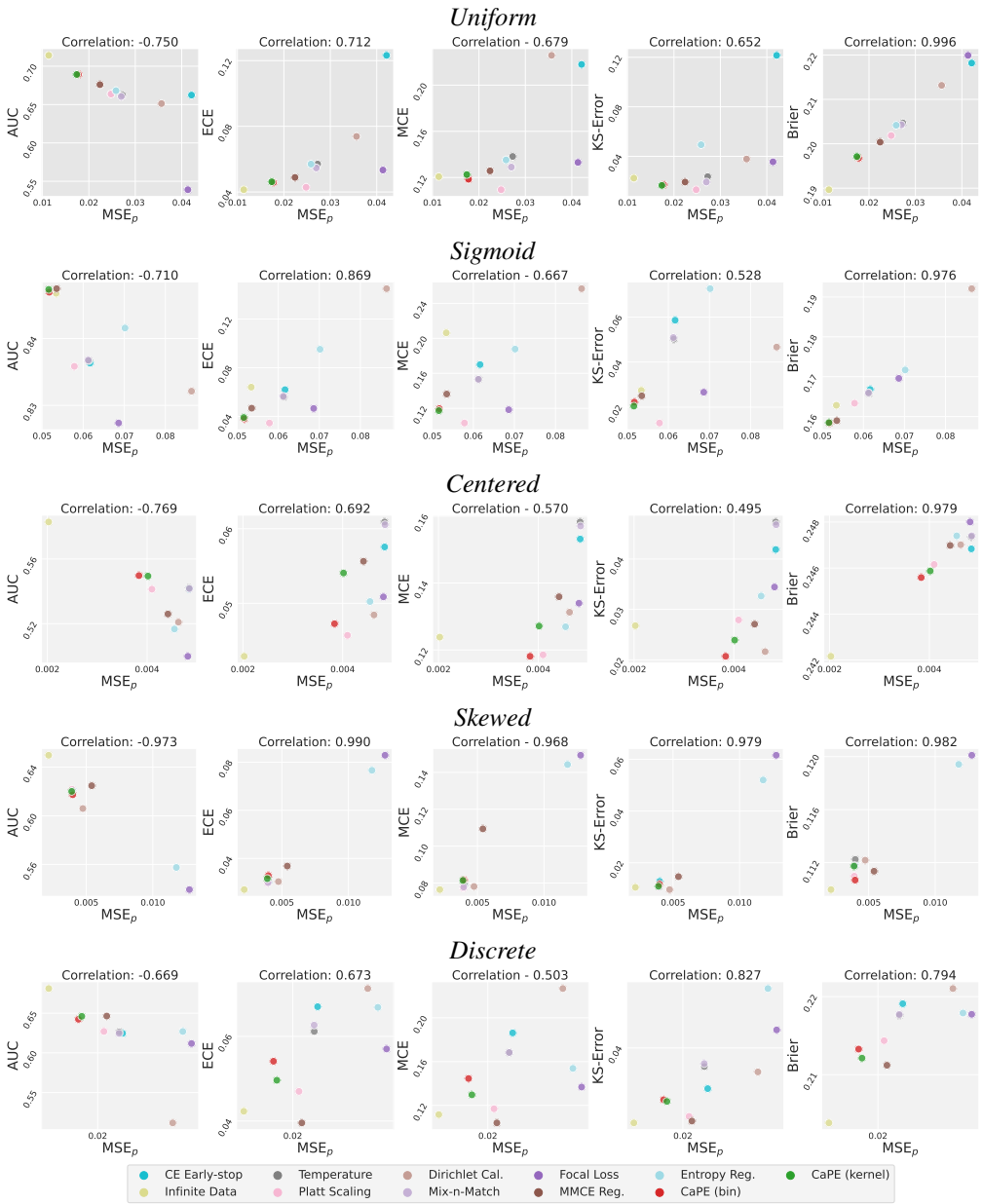


Figure 9: The comparison between MSE$_p$ and other metrics on synthetic data

Figure 9: Comparison between KL_p and other metrics on synthetic data

E ESTIMATION OF EMPIRICAL PROBABILITY IN CAPE

We describe in further detail the two ways to estimate the conditional probability $\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)]$, introduced in Section 4.

We wish to estimate the probability the conditional probability of an output y given a network prediction $f(\boldsymbol{x})$, $\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)]$ We can approximate the probability by averaging over points $\hat{p} \in I(q)$,

$$\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)] \approx \frac{1}{|I(q)|} \sum_{\hat{p} \in I(q)} \mathbb{P}[y = 1|f(\boldsymbol{x}) = p] \tag{7}$$

An empirical estimate of $\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)]$ would be

$$\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)] \approx \frac{1}{|\text{Index}(I(q))|} \sum_{f(\boldsymbol{x}_i) \in I(q)} y_i, \tag{8}$$

where $\text{Index}(I_q) = \{i|f(\boldsymbol{x}_i) \in I(q)\}$.

Alternatively, we can use a kernel density estimation:

$$\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)] \approx \frac{1}{Z} \sum_{\hat{p} \in I(q)} \mathbb{P}[y = 1|f(\boldsymbol{x}) = \hat{p}] \cdot \exp\left(-\frac{(p-q)^2}{\sigma^2}\right), \tag{9}$$

where $Z = \sum_{p \in I(q)} \exp\left(-\frac{(p-q)^2}{\sigma^2}\right)$ is the normalization factor. An empirical estimate of the conditional probability would then be

$$\mathbb{P}[y = 1|f(\boldsymbol{x}) \in I(q)] \approx \frac{1}{Z} \sum_{f(\boldsymbol{x}_i) \in I(q)} y_i \exp\left(-\frac{(f(\boldsymbol{x}_i) - q)^2}{\sigma^2}\right). \tag{10}$$

Based on these two approximation methods, we can design an algorithm to estimate p^i_{emp}.

Bin We divide our data into B bins of equal size. $Q_1, \ldots, Q_B$ are the data B-quantiles. We wish to estimate $\mathbb{P}[y = 1|f(\boldsymbol{x}) \in [Q_{b-1}, Q_b]]$, $b = 1, \ldots, B, Q_0 = 0$. Denote $I_b := [Q_{b-1}, Q_b] \cap \{f(\boldsymbol{x}_i)\}_{i=1}^N$, set of all predictions in $[Q_{b-1}, Q_b]$, and $\text{Index}(I_b) = \{i|f(\boldsymbol{x}_i) \in I_b\}$. We have,

$$\mathbb{P}[y = 1|f(\boldsymbol{x}) \in [Q_{b-1}, Q_b]] \approx p^{(b)}_{\text{emp}} = \frac{1}{|I_b|} \sum_{i \in \text{Index}(I_b)} y_i$$

We assign $p^{(b)}_{\text{emp}}$ to all data points i in the b-th quantile

$$p^i_{\text{emp}} = p^{(b)}_{\text{emp}} \quad \forall i \in \text{Index}(I_b)$$

Kernel In this case we use the kernel density estimation:

$$p^i_{\text{emp}} = \frac{\sum_{k \in \text{NN}(i,r)} K(i,k)\, y_k}{\sum_{k \in \text{NN}(i,r)} K(i,k)}. \tag{11}$$

$\text{NN}(i, r)$ defines r data points whose predictions are nearest to $\hat{p}_i = f(\boldsymbol{x}_i)$ $K(i, j)$ is the Gaussian kernel

$$K(i,j) = \exp\left(-\frac{(\hat{p}_i - \hat{p}_j)^2}{\sigma^2}\right),$$

with hyperparameter σ.

F CALIBRATION BASELINES

This section includes a review of the baseline methods, discussed in Section 6

Post-processing Postprocessing for calibration is to find a function $f_1 : \mathbb{R} \to [0, 1], f_0 = 1 - f_1$ referring to validation set that maps outputs of neural networks to probability space such that $f_j(\hat{p}_i) = P(y_i = j \mid \hat{p}_i)$ where $\hat{p}_i$ is the output of the network.

- Platt scaling (Platt, 1999) optimizes f on validation set within the following family,
$$f_1(\hat{p}_i) = \sigma\left(W^T \hat{p}_i + b\right) \tag{12}$$
where $W \in \mathbb{R}^2, b \in \mathbb{R}$ and σ is the Sigmoid function. The non-probabilistic predictions of a classifier are used as features for a logistic regression model, which is trained on the validation set to return probabilities.
- Temperature scaling (Guo et al., 2017) is a single parameter variant of Platt Scaling where we only change the temperature of the softmax to obtain the calibrated probabilities.
$$f(\hat{p}_i) = \text{Softmax}\left(\hat{p}_i / T\right) \tag{13}$$
where $T \in \mathbb{R}$ minimizes the negative log-likelihood of validation set.
- Beta/Dirchlet calibration (Dir-ODIR) (Kull et al., 2017; 2019) assumes that the probabilities, falls in $[0, 1]$, can be parametrized by a Beta/Dirchlet distribution i.e.
$$f_j \sim \text{Beta}(\alpha^{(j)}, \beta^{(j)}) \tag{14}$$
Assume the prior to be $p(y = j) = \pi_j, \pi_j \in \mathbb{R}$, we have $P(y|f_j) \propto \pi_j f_j$, and then $\alpha^{(j)}, \beta^{(j)}$ are computed by maximizing the posterior.

Ensembling These methods of calibration train the neural networks from end to end with variant ways of training.

- Mix-n-Match (Zhang et al., 2020) improves calibration by ensembling parametric and non-parametric calibrators. Denote the temperature scaling function with $g(\hat{y}_i, T)$. Then Mix-n-Match ensembles different temperatures
$$f_j(\hat{p}_i) = w_1 g_j(\hat{p}_i, T) + w_2 g_j(\hat{p}_i, 0) + w_3 g_j(\hat{p}_i, \infty) \tag{15}$$
After ensembling the parametric temperature scaling, Mix-n-Match applies non-parametric method Isotonic regression.
- Deep ensemble (Lakshminarayanan et al., 2017) Suppose we train M neural networks with different initialization, the probability estimation is the average of single model estimations.
$$p(y_i \mid x_i) = \frac{1}{M} \sum_j^M p_{\theta_j}(y_i \mid x_i) \tag{16}$$

Modified training These methods of calibration train the neural networks from end to end with variant ways of training.

- Confidence penalty (Pereyra et al., 2017): Penalizing low entropy output distributions (confidence penalty) and label smoothing improve state-of-the-art models across benchmarks.
$$\mathcal{L}(\theta) = -\sum_i \log p_\theta(y_i|x_i) - \beta H(p_\theta(y_i|x_i)) \tag{17}$$
- Focal loss (Mukhoti et al., 2020): Focal loss implicitly maximises entropy while minimising the KL divergence between the predicted and the target distributions. It also regularizes the weights of neural network and avoids overfitting.
$$\mathcal{L}(\theta) = -\sum_i (1 - p_\theta(y_i|x_i))^\gamma \log p_\theta(y_i|x_i) \tag{18}$$
where $\gamma \in \mathbb{R}$.
- Kernel MMCE (Kumar et al., 2018) is a RKHS kernel based measure of calibration that is efficiently trainable alongside the negative likelihood loss. Given data samples $\mathcal{D} = \{(c_i, r_i)\}_{i=0}^m$, where $c_i = \chi_{\{\hat{y}_i = y_i\}}$ and $r_i = \mathbb{P}(c_i = 1|\hat{y}_i)$, MMCE is computed on samples $\mathcal{D}$ as following,
$$\text{MMCE}^2(\mathcal{D}) = \sum_{i,j} \frac{(c_i - r_i)(c_j - r_j)k(r_i, r_j)}{m^2} \tag{19}$$
where (r_i, r_j) is a kernel function. MMCE is optimized together with the cross entropy loss as a regularization. The strength of calibration can be adjusted by scale $\lambda \in \mathbb{R}$.
$$\mathcal{L}(\theta) = -\sum_i \log p_\theta(y_i|x_i) + \lambda \left(\text{MMCE}^2(D)\right)^{\frac{1}{2}} \tag{20}$$

Intuitively, by minimizing MMCE, the distances among the confidence level of different samples are enlarged. The samples with higher confidence are regularized more.

G SYNTHETIC DATA GENERATION

We describe the generation of synthetic probability labels under different scenarios, discussed in Section 6.

The synthetic data is split into training, validation, and test sets with 16641, 4738, and 2329 samples, respectively. The training and validation sets contain only images x_i and 0-1 labels y_i for training and tuning the model. The held-out test set contains the ground truth probabilities p_i, in addition to x_i and y_i, in order to evaluate the performance of the model for probability estimation. Note that we do not use ground-truth p_i values during training or inference - we only use it compare the performance of different models.

Ground Truth Probability Generation The ground truth probability is simulated by $p_i = \psi(z_i)$ where z_i is age of the person.

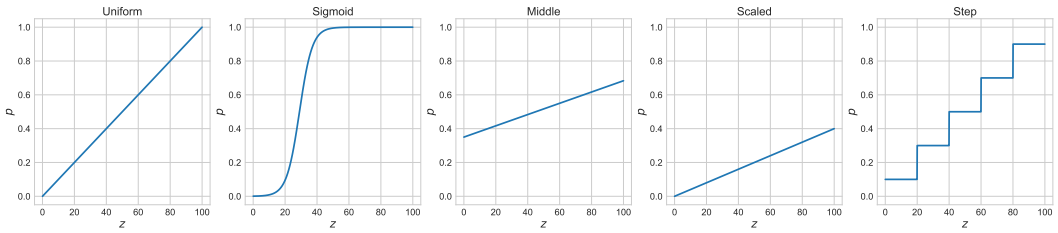


Figure 10: Illustration of $\psi(z)$ for synthetic data simulations

Label distribution After determining the probability p_i using $\psi(z)$, the label y_i is sampled from a Bernoulli distribution parametrized by p_i, so that it takes the value 1 with probability p_i. The Distributions of y_i under five different scenarios are illustrated in Fig.11.

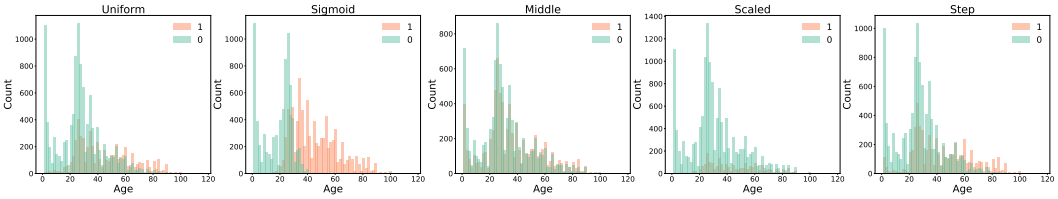


Figure 11: The histograms of y_i's for simulation in synthetic data

H REAL-WORLD DATA AND EXPERIMENT DETAILS

We present here supplementary information for the real-world datasets, introduced in Section 6.

Cancer Survival Histopathological features are useful in identification of tumor cells, cancer subtypes, and the stage and level of differentiation of the cancer. Hematoxylin and Eosin (H&E)-stained slides are the most common type of histopathology data and the basis for decision making in the clinics. With these properties, H&E are used for mortality prediction of cancer (Wulczyn et al., 2020). In this experiment, we use the H&E slides of non-small cell lung cancers from The Cancer Genome Atlas Program (TCGA)[3] to predict the 5-year survival. The dataset has 1512 whole slide images

[3] `https://www.cancer.gov/tcga`

from 1009 patients, and 352 of them died in 5-years. We split the samples by patients and source institutions into training, validation, and test set, which has 1203, 151, and 158 samples respectively.

The whole slide images contains numerous pixels, so we cropped the slides into tiles at 20x magnification with 1/4 overlapping, resized them to 299×299 with bicubic interpolation, and filtered out the tiles with more than 85% area covered by the background. The representations of each tile is trained with self-supervised momentum contrastive learning (MoCo) (Chen et al., 2020), and the slide-level prediction is made by a multiple-instance learning network (Ilse et al., 2018) trained with the binary label of survival in 5 years.

Weather Forecasting We use the German Weather service dataset[4], which contains quality-controlled rainfall-depth composites from 17 operational Doppler radars. Three precipitation maps from the past 30 minutes serve as an input. The training labels are the 0/1 events of the mean precipitation increasing.

The German Weather service (DWD - Deutshce Wetter Dienst) dataset `https://opendata.dwd.de/weather/radar/` contains quality-controlled rainfall-depth composites from 17 operational DWD Doppler radars. It has a spatial extent of 900x900 km, and covers the entirety of Germany. Data exists since 2006, with a spatial and temporal resolution of 1x1 km and 5 minutes, respectively. The dataset has been used to train RainNet, a pricipitation nowcasting model (Ayzel, 2020).

The network architecture is ResNet18, with 3 input channels and 2 output channels. The input to the network are 3 precipitation maps which cover a fixed area of 300km$\times$300 km in the center of the grid (300$\times$ 300 pixels), set 10 minutes apart. The training, validation and test datasets consist of 20000, 6000 and 3000 samples, respectively, all separated temporally, over the span of 15 years.

Collision Prediction Vehicle collision is one of the leading causes of death in the world. Reliable collision prediction systems which can warn drivers about potential collisions can save a significant number of lives. A standard way to design such a system is to train a convolutional model for identifying if a particular vehicle in the dash-cam video feed might collide with the car in next few seconds. More formally, at time $t = T$ the system tries to predict if any car in the video might collide with our given car in time $t \in [T, T + T_{\text{look-ahead}}]$. Each labelled training sample consists of features $X = (X^{T-\delta}, X^{T-2\delta}, \ldots, X^{T-d\delta})$ and a binary label $Y \in \{0, 1\}$ denoting if an accident will occur in $t \in [T, T + T_{\text{look-ahead}}]$. Each X^t is a tensor with 4 channels where the first 3 channels corresponds to an RGB image of the dashcam view at time $t = t$, and the fourth channel consists of a mask with a bounding box on a particular vehicle of interest. In this work, we use *YouTubeCrash* dataset (Kim et al., 2019) to train and test our model, which uses $\delta = 0.1s$,$T_{\text{look-head}} = 18\delta = 1.8s$, and $d = 3$. Following Kim et al. (2019) we used a VGG network architecture.

The dataset contains 122 accident scenes, and 2096 non-accident scenes, which after feature extraction gives us 2096 positive samples, and 11486 negative samples (the dataset is severely imbalanced, and similar to the Skewed situation in Section 6.1). We further split the dataset into train (6453 samples for label 0, and 1023 samples for label 1), validation (2348 samples for label 0, and 545 samples for label 1), and test (2685 samples for label 0, and 528 samples for label 1) sets. The samples in train, validation and test sets are generated from disjoint scenes/dashcam videos.

[4] `https://opendata.dwd.de/weather/radar/`

I CASE STUDY: CANCER SURVIVAL

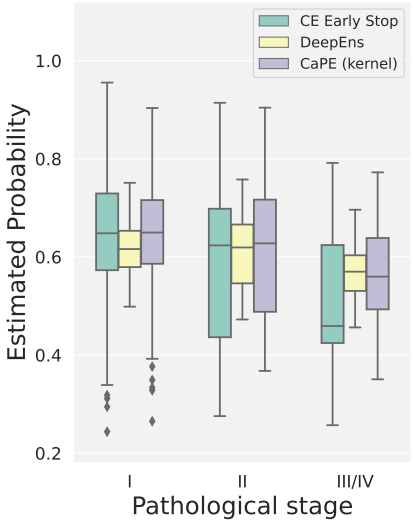


Figure 12: The estimated probability of survival grouped by pathological stages. Deep ensemble has average estimations across all the stages; CE has large variance; CaPE is balanced between two methods.

The case study in Section 7. For cancer survival prediction, we visualize the estimated probabilities on the test set in different pathological stages in Figure 12. In general, patients with earlier stages have higher probabilities of survival, but the confidence intervals of deep ensemble overlap with each other (i.e the model is less discriminative), while CE has a wide confidence interval in probability estimation. CaPE is able to shrink the wide confidence intervals of the CE model as well as retain the differences in probability estimations among stages. It can be attributed to the combination of discriminative loss and calibration loss during training.