Deep Probability Estimation

Sheng Liu [*1] **Aakash Kaku** [*1] **Weicheng Zhu** [*1] **Matan Leibovich** [*2] **Sreyas Mohan** [*1] **Boyang Yu** [1]
Haoxiang Huang [2] **Laure Zanna** [1 2] **Narges Razavian** [3] **Jonathan Niles-Weed** [1 2] **Carlos Fernandez-Granda** [1 2]

Abstract

Reliable probability estimation is of crucial importance in many real-world applications where there is inherent (aleatoric) uncertainty. 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 difference that the objective is to estimate probabilities rather than predicting the specific outcome. This work investigates 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 model (epistemic) uncertainty. For 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. We also give a theoretical analysis of a model for high-dimensional probability estimation which reproduces several of the phenomena evinced in our experiments. Finally, we 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]

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. evaluating the likelihood of each possible outcome for 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 contrast, in classification the class label is usually completely determined by the data (a picture either shows a cat or it does not).

The goal of this work is to investigate probability estimation from high-dimensional data using deep neural networks. Probability estimation is a fundamental problem in machine learning (Murphy, 2013). 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 to improve it (Platt, 1999; Guo et al., 2017; Szegedy et al., 2016; Zhang et al., 2020; Thulasidasan et al., 2020; Mukhoti et al., 2020; Thagaard et al., 2020), including Bayesian neural networks (Gal & Ghahramani, 2016; Wang et al., 2016; Shekhovtsov & Flach, 2019; Postels et al., 2019). However, these works restrict their attention almost exclusively to classification in datasets (e.g. CIFAR-10/100 (Krizhevsky,

[*]Equal contribution [1]Center for Data Science, New York University, New York, USA [2]Courant Institute of Mathematical Sciences, New York University, New York, USA [3]Department of Population Health & Department of Radiology, NYU School of Medicine, New York, USA. Correspondence to: SL, AK, WZ, ML, SM <{shengliu, ark576, jackzhu, ml7557, sm7582}@nyu.edu>.

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

[1]Code available at https://jackzhu727.github.io/deep-probability-estimation/.

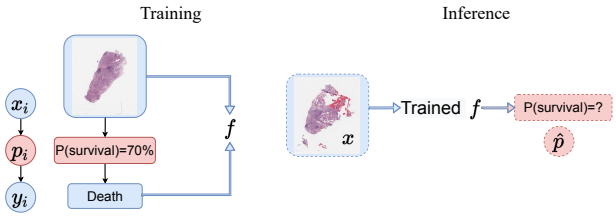

Figure 1. **The probability-estimation problem.** In probability estimation, we assume that each observed outcome y_i (e.g. death or survival in cancer patients) in the training set is randomly generated from a latent unobserved probability p_i associated to the corresponding data $\boldsymbol{x}_i$ (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 $\hat{p} \in [0, 1]$.

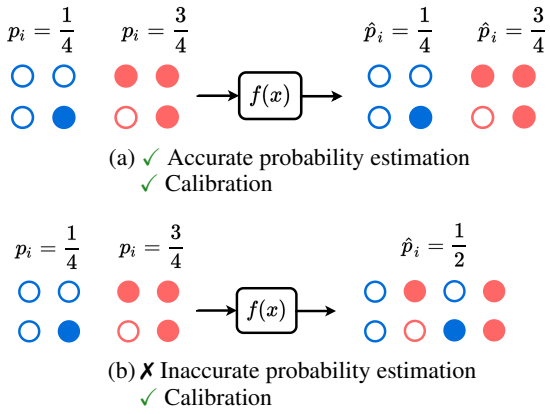

Figure 2. **Calibration is not enough**. Uncolored/colored markers denote $y = 0/1$ outcomes, respectively. Blue/red stand for two classes with different associated ground-truth probabilities (1/4 and 3/4 respectively). (a) The model f retrieves the true probabilities, which requires discriminating between 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%.

2009), or ImageNet (Deng et al., 2009)) where the label itself is *not uncertain*: it quantifies the confidence of the model in its own prediction, *not the probability of an event of interest*. In the literature, this is known as epistemic (model) uncertainty (Hüllermeier & Waegeman, 2021; Tagasovska & Lopez-Paz, 2019). We focus on aleatoric uncertainty, stemming from inherent uncertainty in the problem under study. To formalize this distinction, we propose and rigorously analyze a simple high-dimensional model with *uncertain* labels, and show that even in this simple model, classification-based methods fail to yield good probability estimates.

Probability estimation from high-dimensional data is a problem of critical importance in medical prognostics (Wulczyn et al., 2020), weather prediction (Agrawal et al., 2019), and autonomous driving (Kim et al., 2019). In order to advance deep-learning methodology for probability estimation it is crucial to build appropriate benchmark datasets. Here we build a synthetic dataset and gather three real-world datasets, which we use to systematically evaluate existing methodology. In addition, we propose a novel approach for probability estimation, which outperforms current state-of-the-art methods. Our contributions are the following:

- We give a theoretical analysis of the probability estimation problem for a high-dimensional logistic model, and establish that predictors trained by minimizing the cross entropy loss overfit the observed outcomes and fail to yield calibrated outcomes. However, we show that the predictions *are* well calibrated during the initial stages of training.
- We introduce a new synthetic dataset for probability estimation where a population of people may have a certain disease associated with age. The task is to estimate the probability that they contract the disease from an image of their face. The data are generated using the UTKFaces dataset (Zhang et al., 2017b), 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 Calibrated Probability Estimation (CaPE), a novel technique which modifies the training process so that output probabilities are consistent with empirical probabilities computed from the data. CaPE outperforms existing approaches on most metrics on synthetic and real-world data.

2 Problem Formulation

The goal of probability estimation is to evaluate the likelihood of a certain event of interest, based on observed data. The available training data consist 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 5 years after $\boldsymbol{x}_i$ was collected. The data have inherent uncertainty: y_i, the patient's survival, does not depend deterministically on the histopathology image (due e.g. to comorbidities and other health factors). Instead, we assume that y_i equals 1 with a certain probability p_i associated with $\boldsymbol{x}_i$, as illustrated in Figure 1, because the input data provides key information about the patient's survival chances.

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 $\boldsymbol{x}$ (e.g. the probability of survival for over 5 years for new patients based on their histopathology data). To summarize, this is not just a classification problem, because it involves aleatoric uncertainty. Instead, the goal is to predict the probability of the outcome, which is critical in choosing a course of treatment for the patient.

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 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%.[2]

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:

$$\text{MSE}_p = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - p_i)^2,$$

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

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

Calibration metrics. Ground-truth probabilities are not available for real-world data. 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

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

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) (note however that the definition in (Guo et al., 2017) is specific to classification). Given a probability-estimation model f and a dataset of input data $\boldsymbol{x}_i$ and associated outcomes y_i, $1 \le i \le N$, 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 empirical and predicted probabilities,

$$p_{\text{emp}}^{(b)} = \mathbb{E}\left(y \mid f(\boldsymbol{x}) \in I_b\right) = \frac{1}{|I_b|}\sum_{i \in \text{Index}(I_b)} y_i, \tag{2}$$

$$q^{(b)} = \frac{1}{|I_b|}\sum_{i \in \text{Index}(I_b)} f(\boldsymbol{x}_i), \tag{3}$$

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

$$\text{ECE} = \frac{1}{B}\sum_{b=1}^{B}\left|p_{\text{emp}}^{(b)} - q^{(b)}\right|. \tag{4}$$

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 D.

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

$$\text{Brier} = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - y_i)^2. \tag{5}$$

[2] A perfect model (in terms of probability estimation), assigns 0.25 to the blue class and 0.75 to the red class. To maximize classification accuracy, we predict 1 when the model outputs 0.75 (red examples) and 0 when it outputs 0.25 (blue examples). However, 25% of red examples have an outcome of 0, and 25% of blue examples have an outcome of 1. As a result, the model would only have 75% accuracy.

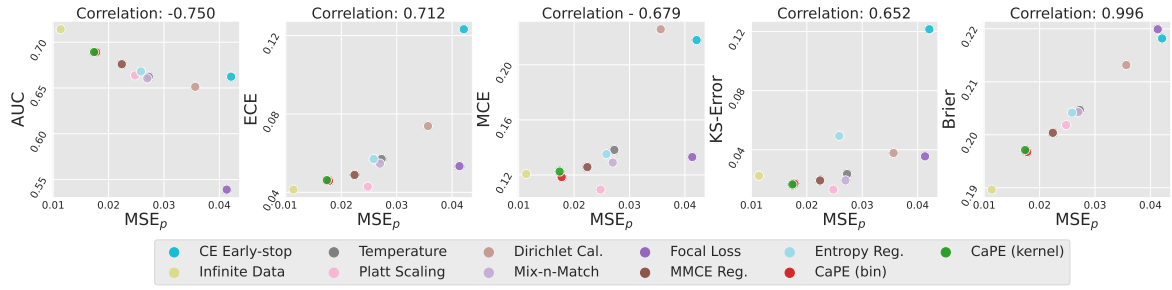

Figure 3. **Evaluating evaluation metrics.** We use synthetic data to compare different metrics to the *gold-standard* MSE$_p$ that uses ground-truth probabilities. Brier score is highly correlated with MSE$_p$, in contrast to the classification metric AUC and the calibration metrics ECE, MCE and KS-Error. The graphs show the results of the proposed method CaPE, as well as the baselines described in Section 7.3 on the *Linear* scenario (see Section 7.1). Results on other scenarios and a similar comparison with KL$_p$ are reported in Appendix F.

This score can be decomposed into two terms associated with calibration and discrimination, as shown in Appendix E. Using the synthetic data in Section 7.1, where the ground-truth probabilities are known, we show that Brier score is indeed a reliable proxy for the *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 F).

4 Early Learning and Memorization

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*, meaning 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, which are highly overparametrized and therefore prone to overfitting (Goodfellow et al., 2016). For classification, deep neural 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 and *memorize* 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). However, 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).

Though early learning and memorization are typically observed when training prediction models based on deep neural networks, we argue that these observations represent a much more general phenomenon, intrinsic to the problem of probability estimation with finite data when the dimension is large. To substantiate this claim, we propose a simple analytical model, where data samples $\boldsymbol{x}_i \in \mathbb{R}^d$ are drawn from a high dimensional normal distribution $\boldsymbol{x}_i \sim \mathcal{N}(0, I_d)$. The probability of each data point is determined by a generalized linear model $p_i(\boldsymbol{\theta}) = \left(1 + e^{-\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle}\right)^{-1}$, with true parameter $\boldsymbol{\theta}^*$. For $k \geq 1$ we denote by $\hat{p}^k$ the predictor obtained by running k iterations of gradient descent on the cross-entropy loss with step size η. We prove the following.

Theorem 4.1 (Informal)**.** *There exists $\kappa^* \in (0, +\infty)$ such that the following holds: if p and n are sufficiently large, the mean squared error of $\hat{p}^k$ decreases monotonically during the first $k = O(1/\eta)$ iterations of gradient descent, but if $\frac{p}{n} > \kappa^*$, then as $k \to \infty$, $\hat{p}^k$ collapses to a predictor that only predicts probabilities 0 and 1.*

A precise statement and proof can be found in Appendix A. Theorem 4.1 identifies a sharp threshold (κ^*) at which the memorization phenomenon occurs and separates early learning stage and memorization. It indicates that even simple generalized linear models exhibit the early learning and memorization phenomena: in high dimensions, predictors obtained by cross-entropy minimization eventually memorize the data. This is likely to plague any overparamterized model, including neural networks. However, this phenomenon does *not* occur if gradient descent is stopped early. This is illustrated in Figures 8 and 7 in Appendix B, which demonstrate that empirically the linear model has this qualitative behavior. These observations motivate our proposed methodology.

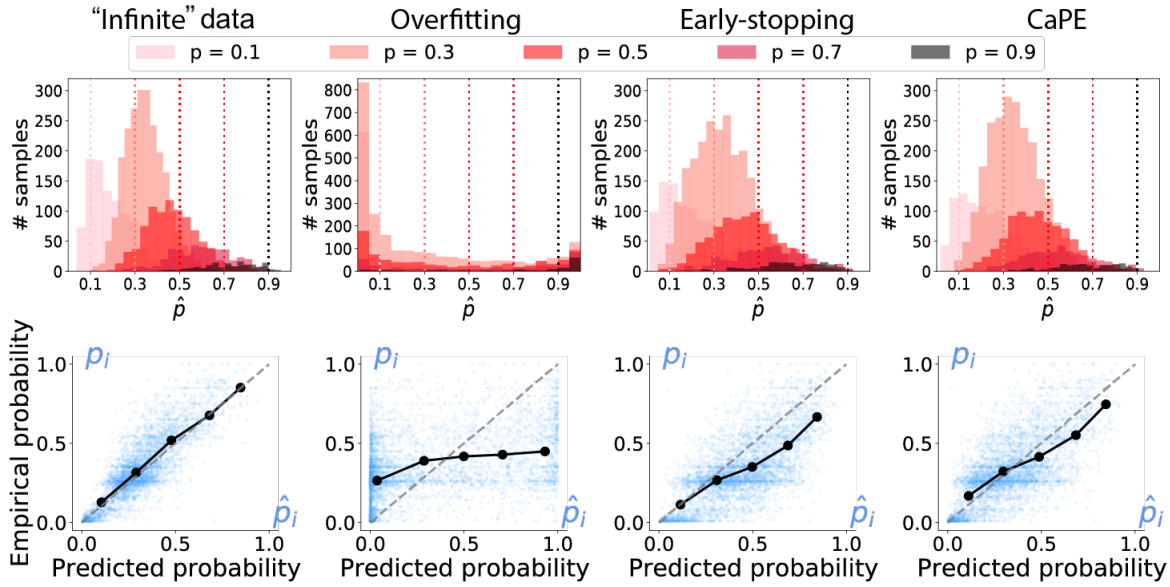

Figure 4. **Miscalibration due to overfitting and how to avoid it.** The top row shows the histogram of predicted probabilities for the synthetic *Discrete* scenario (see Section 7.1). Ideally each histogram should be concentrated around the corresponding value of p. The bottom row shows results for the *Linear* scenario. The horizontal and vertical coordinate of each blue point represent the predicted ($\hat{p}_i$) and true (p_i) probabilities of a test example respectively. We also show a reliability diagram of binned mean predicted and empirical probabilities in black (see Section 3). The dashed line indicates perfect calibration. When trained on *infinite* data obtained by resampling outcome labels at each epoch according to ground-truth probabilities, models minimizing cross-entropy are well calibrated (first column). 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 (i.e. selecting the model based on validation cross-entropy loss), 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. Appendix C.3 shows plots for all synthetic data scenarios.

5 Calibrated Probability Estimation (CaPE)

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 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,

$$\mathcal{L}_{\mathrm{D}} = -\sum_{i=1}^{N} \left[y_i \log(f(\boldsymbol{x}_i)) + (1 - y_i) \log(1 - f(\boldsymbol{x}_i)) \right]$$

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

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

where p^i_{emp} is an estimate of the conditional probability $\mathbb{P}[y = 1 | f(\boldsymbol{x}) \in I(f(\boldsymbol{x}_i))]$ and $I(f(\boldsymbol{x}_i))$ is a small interval centered at $f(\boldsymbol{x}_i)$. As explained in Section 3 if $f(\boldsymbol{x}_i)$ 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 $p^i_{\mathrm{emp}} = p^{(b_i)}_{\mathrm{emp}}$ in equation 2. (2) CaPE (kernel) where p^i_{emp} is estimated through a moving average with a kernel function (see Appendix G for more details). Both methods are efficiently computed by sorting the predictions $\hat{p}_i$. The calibration loss requires a reasonable estimation of the empirical probabilities $p^{(i)}_{\mathrm{emp}}$, which can be obtained from the model after early learning. Therefore using the calibration loss from the beginning is counterproductive, as demonstrated in Section M. We note that a variant of CaPE can be implemented using a weighted sum of the calibration and discrimination loss.

CaPE is summarized in Algorithm 1. Figures 4 and 5 show that incorporating the calibration-loss minimization step indeed preserves calibration as training proceeds (this is not necessarily expected 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 overfit-

ting, as shown in Figure 5. The experiments with synthetic and real-world data reported in Section 7 suggest that this approach results in accurate probability estimates across a variety of realistic scenarios.

6 Related Work

Neural networks trained for classification often generate a probability associated with their prediction which quantifies its uncertainty. These estimates have been found to be inaccurate in certain situations (Mukhoti et al., 2020; Guo et al., 2017; Zhao & Ermon, 2021) (although a recent study suggests that transformer-based models tend to be well calibrated in vision-based classification tasks (Minderer et al., 2021)). Calibration methods to mitigate this issue 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., 2021; 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. Another approach trains a recalibration model on the outputs of an uncalibrated model (Kuleshov et al., 2018). 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) uses a single model, and ensembles predictions using multiple temperature scaling transformations. Other methods (Lakshminarayanan et al., 2017; Maddox et al., 2019) ensemble multiple models obtained using different initializations. These approaches are compatible with the proposed method CaPE; how to combine them effectively is an interesting future research direction.

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.

Datasets for evaluation The methods discussed in this section were developed for calibration in classification, and 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.

7 Experiments

7.1 Synthetic Dataset: Face-based Risk Prediction

To benchmark probability-estimation methods, we build a synthetic dataset based on UTKFace (Zhang et al., 2017b), containing face 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 $\psi : \mathbb{Z}_{\geq 0} \to [0, 1]$. Then we simulate whether the person actually contracts the illness (label $y_i = 1$) or not ($y_i = 0$) with probability p_i. The probability-estimation task is to estimate the ground-truth probability p_i from the face image x_i using a model that only has access to the images and the binary observations during training. This requires learning to discriminate age and map it to the corresponding risk. We design ψ to create five scenarios, inspired by real-world data (see Appendix I):

- **Linear**: 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: $\psi(z) = 0.2[\mathbb{1}_{\{z>20\}} + \mathbb{1}_{\{z>40\}} + \mathbb{1}_{\{z>60\}} + \mathbb{1}_{\{z>80\}}] + 0.1$

In addition, we report an experiment with simulated probabilistic labels on CIFAR-10 in Appendix J.

7.2 Real-World Datasets

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

Survival of Cancer Patients. Histopathology aims to identify 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 used 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. 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 from The Cancer Genome Atlas Program

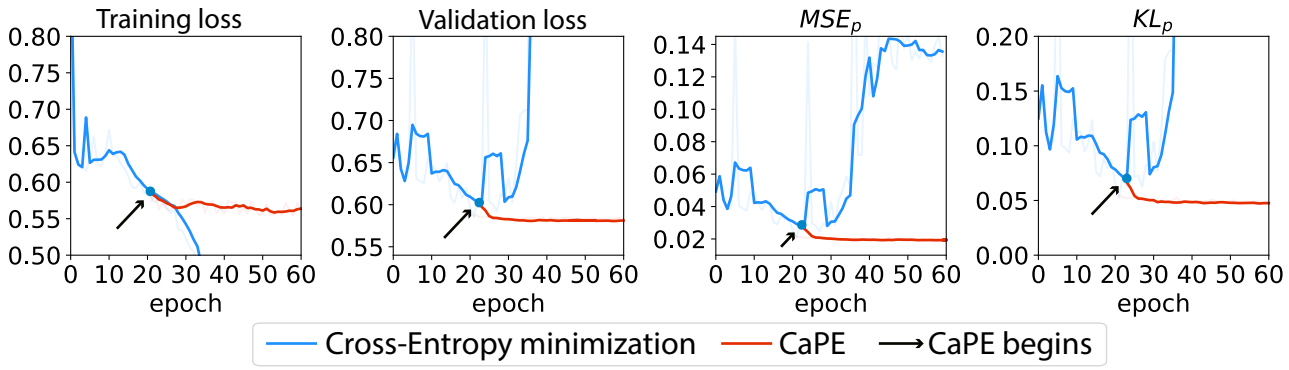

Figure 5. **Calibrated Probability Estimation prevents overfitting.** Comparison between the learning curves of cross-entropy (CE) minimization and the proposed calibrated probability estimation (CaPE), smoothed with a 5-epoch moving average. After an *early-learning* stage where both training and validation 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: $K(p,q) \coloneqq \exp\left[-(p-q)^2/\sigma^2\right]$ ▷ 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
  end for

function BIN(B) ▷ B-number of bins
  $I_1, \cdots I_B \leftarrow$ partitions by quantile of $\{\hat{p}_j\}_{j=1}^N$
  Find b such that $\hat{p}_i \in I_b$
  $\text{Index}(I_b) \leftarrow \{j | \hat{p}_j \in I_b\}$ ▷ 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; K-kernel
  $N_r(i) \leftarrow r$-nearest neighbor of $\hat{p}_i$ (output probability space)
  $Z \leftarrow \sum_{\hat{p}_j \in N_r(i)} K(\hat{p}_i, \hat{p}_j)$ ▷ 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

(TCGA)[3] to estimate the the 5-year survival probability of cancer patients. The outcome distribution is similar to the *Centered* scenario in Section 7.1.

Weather Forecasting. The atmosphere is governed by nonlinear dynamics, hence weather forecast models possess inherent uncertainties (Richardson, 2007). Nowcasting, weather prediction in the near future, 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[4], 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 over the area covered will increase or decrease one hour after the most recent measurement. Three precipitation maps from the past 30 minutes serve as an input. The outcome distribution is similar to the *Linear* scenario in Section 7.1.

Collision Prediction. Vehicle collision is one of the leading causes of death in the world. Reliable collision prediction systems are therefore instrumental in saving human lives. These systems predict potential collisions from dashcam cameras. Collisions are influenced by many unknown factors, and hence are not deterministic. Following (Kim et al., 2019), we use 0.3 seconds of real dashcam videos from `YouTubeCrash` dataset as input, and predict the probability of a collision in the next 2 seconds. The data are very imbalanced as the number of collisions is very low, so the outcome distribution is similar to the *Skewed* scenario in Section 7.1.

7.3 Baselines

We apply existing calibration methods developed for classification to probability estimation (as well as cross-entropy minimization with early-stopping): (1) **Three post-**

[3] https://www.cancer.gov/tcga
[4] https://opendata.dwd.de/weather/radar/

Table 1: Results on synthetic data. All numbers are downscaled by 10^{-2}. Appendix C.1 shows a table with confidence intervals obtained via bootstrapping. * is a model trained via cross-entropy minimization from data obtained by continuous label resampling. No baseline outperforms the proposed method CaPE in any of the scenarios, and CaPE outperforms all the individual baseline models in most scenarios, in both metrics, except for the skewed case, where the difference is statistically insignificant.

Methods	*Linear*		*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
CE + resampled labels*	$1.14_{\pm.04}$	$2.81_{\pm.11}$	$5.34_{\pm.20}$	$14.82_{\pm.51}$	$4.21_{\pm.15}$	$4.21_{\pm.15}$	$4.21_{\pm.15}$	$4.21_{\pm.15}$	$4.21_{\pm.15}$	$4.21_{\pm.15}$
CE early-stop	$4.21_{\pm.15}$	$10.94_{\pm.36}$	$6.16_{\pm.17}$	$17.16_{\pm.48}$	$0.48_{\pm.01}$	$0.98_{\pm.03}$	$0.40_{\pm.01}$	$1.79_{\pm.06}$	$2.24_{\pm.08}$	$5.27_{\pm.17}$
Temperature	$2.73_{\pm.11}$	$6.75_{\pm.25}$	$6.16_{\pm.17}$	$17.09_{\pm.43}$	$0.48_{\pm.01}$	$0.98_{\pm.03}$	$0.40_{\pm.02}$	$1.76_{\pm.06}$	$2.21_{\pm.08}$	$5.15_{\pm.18}$
Platt Scaling	$2.48_{\pm.09}$	$6.07_{\pm.22}$	$5.78_{\pm.19}$	$16.15_{\pm.47}$	$0.41_{\pm.01}$	$0.83_{\pm.03}$	$\mathbf{0.39}_{\pm.01}$	$1.72_{\pm.06}$	$2.06_{\pm.08}$	$4.83_{\pm.17}$
Dirichlet Cal.	$3.56_{\pm.13}$	$9.08_{\pm.29}$	$8.64_{\pm.26}$	$25.18_{\pm.58}$	$0.46_{\pm.01}$	$0.94_{\pm.03}$	$0.47_{\pm.02}$	$2.31_{\pm.07}$	$2.74_{\pm.10}$	$6.53_{\pm.22}$
Focal Loss	$4.13_{\pm.11}$	$10.52_{\pm.28}$	$6.86_{\pm.21}$	$19.46_{\pm.50}$	$0.48_{\pm.01}$	$0.97_{\pm.03}$	$1.28_{\pm.03}$	$1.63_{\pm.66}$	$2.92_{\pm.08}$	$6.77_{\pm.21}$
Mix-n-match	$2.70_{\pm.11}$	$6.72_{\pm.24}$	$6.12_{\pm.17}$	$17.08_{\pm.46}$	$0.48_{\pm.01}$	$0.98_{\pm.03}$	$0.40_{\pm.01}$	$1.75_{\pm.05}$	$2.21_{\pm.08}$	$5.14_{\pm.18}$
Entropy Reg.	$2.58_{\pm.09}$	$6.65_{\pm.21}$	$7.02_{\pm.17}$	$21.16_{\pm.42}$	$0.45_{\pm.01}$	$0.92_{\pm.03}$	$1.18_{\pm.03}$	$10.74_{\pm.65}$	$2.84_{\pm.08}$	$6.62_{\pm.19}$
MMCE Reg.	$2.24_{\pm.08}$	$5.68_{\pm.20}$	$5.35_{\pm.18}$	$15.06_{\pm.49}$	$0.44_{\pm.01}$	$0.90_{\pm.03}$	$0.54_{\pm.02}$	$2.44_{\pm.08}$	$2.09_{\pm.08}$	$4.92_{\pm.18}$
Deep Ensemble	$1.90_{\pm.07}$	$4.55_{\pm.18}$	$5.86_{\pm.22}$	$16.46_{\pm.60}$	$0.44_{\pm.01}$	$0.89_{\pm.03}$	$0.55_{\pm.02}$	$2.58_{\pm.07}$	$1.97_{\pm.08}$	$4.61_{\pm.17}$
CaPE (bin)	$1.78_{\pm.07}$	$4.35_{\pm.16}$	$5.17_{\pm.20}$	$\mathbf{14.27}_{\pm.49}$	$\mathbf{0.38}_{\pm.01}$	$\mathbf{0.78}_{\pm.03}$	$0.40_{\pm.02}$	$1.73_{\pm.06}$	$\mathbf{1.81}_{\pm.08}$	$\mathbf{4.28}_{\pm.18}$
CaPE (kernel)	$\mathbf{1.74}_{\pm.07}$	$4.30_{\pm.17}$	$\mathbf{5.16}_{\pm.20}$	$14.34_{\pm.49}$	$0.40_{\pm.01}$	$0.81_{\pm.03}$	$\mathbf{0.39}_{\pm.01}$	$\mathbf{1.69}_{\pm.06}$	$1.84_{\pm.08}$	$4.35_{\pm.17}$

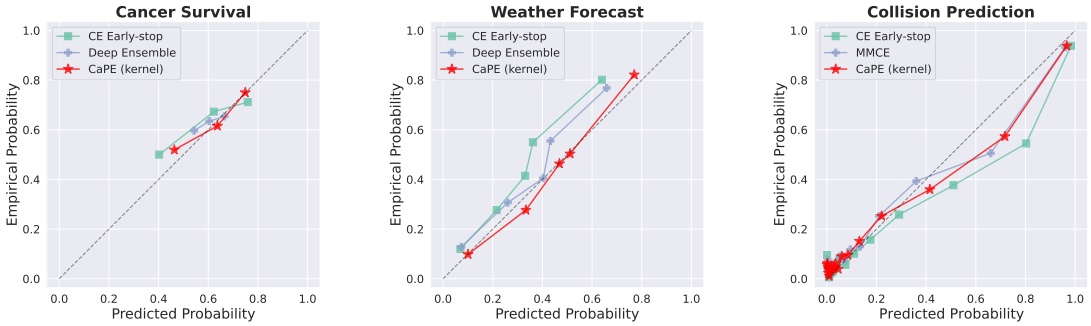

Figure 6. **Reliability diagrams for real-world data.** Reliability diagrams of binned mean predicted and empirical probabilities (see Section 3). The dashed line indicates perfect calibration. The results are computed on test data for cross-entropy minimization with early stopping, the proposed method (CaPE) and the best baseline for each dataset. CaPE produces better calibrated outputs. Appendix C.3 shows additional reliability diagrams.

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, (2) **Two Ensemble Methods**: Mix-n-Match (Zhang et al., 2020) applied to best CE model, and Deep Ensemble (Lakshminarayanan et al., 2017) with 5 networks, and (3) **Three Modified Training methods**: Focal loss (Mukhoti et al., 2020), entropy-maximizing loss (Pereyra et al., 2017), and MMCE regularization (Kumar et al., 2018). Appendix H provides a detailed description. For our experiments on synthetic data, we also compare against a model trained on a **large amount of data** by repeatedly sampling new outcomes from the ground-truth probabilities at each epoch. This provides a best-case reference for each scenario. We refer to Appendix H, for a detailed discussion of the baselines and hyperparameter optimization.

8 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 best-case baseline with resampled labels 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 met-

Table 2: Results on cancer-survival prediction, weather forecasting, and collision prediction. All numbers are downscaled by 10^{-2}. Tables with all the metrics described in Section 3 are provided in Appendix C.2. The proposed method CaPE outperforms existing techniques in terms of Brier score, the metric that best captures probability-estimation accuracy.

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**

rics 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 most other methods. This demonstrates that enforcing better probability estimation 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, demonstrating that CaPE produces outputs that are better calibrated on real data.

Figure 6 also shows that each real-world dataset closely aligns with a particular synthetic scenario: cancer survival with *Centered*; weather forecasting with *Linear*; 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 (see Appendix L for further analysis based on pathological stages). Accordingly, deep ensemble also underperforms in the synthetic scenarios where ground-truth probabilities are clustered closely (*Sigmoid*, *Centered*), but is effective for *Linear*. 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 evaluating methodology for probability estimation.

9 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 provide theoretical analysis showing that early learning and memorization are fundamental phenomena in high-dimensional probability estimation. Motivated by this, we propose a novel approach that outperforms existing approaches on our simulated and real-world benchmarks. An important application for probability estimation is in the context of survival analysis, which can be recast as estimation of conditional probabilities (Lee et al., 2018; Shamout et al., 2020; Goldstein et al., 2021). Another interesting research direction is to consider problems with several possible uncertain outcomes (analogous to multiclass classification).

Acknowledgements

The authors gratefully acknowledge support from Alzheimer's Association grant AARG-NTF-21- 848627 (S.L.), NIH grant R01 LM01331 (A.K., B.Y.), NSF grants HDR-1940097 (M.L.), DMS2009752 (C.F.G., S.L.) and NRT-1922658 (A.K., S.L., S.M., B.Y., W.Z.), and the NYU Langone Health Predictive Analytics Unit (N.R., W.Z.). Funding for the Multiscale Machine Learning In coupled Earth System Modeling (M2LInES) project was provided to Laure Zanna by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program. We would also like to thank Juan Argote for useful discussions, and the reviewers for their comments.

References

Agrawal, S., Barrington, L., Bromberg, C., Burge, J., Gazen, C., and Hickey, J. Machine learning for precipitation nowcasting from radar images. *arXiv preprint arXiv:1912.12132*, 2019.

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

Buja, A., Stuetzle, W., and Shen, Y. Loss functions for bi-

nary class probability estimation and classification: Structure and applications," manuscript, available at www-stat.wharton.upenn.edu/ buja, 2005.

Candes, E. J. and Sur, P. The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression, 2018.

Chen, X., Fan, H., Girshick, R. B., and He, K. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. URL https://arxiv.org/abs/2003.04297.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. 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.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

Goldstein, M., Han, X., Puli, A., Perotte, A. J., and Ranganath, R. X-cal: Explicit calibration for survival analysis, 2021.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT Press, 2016.

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

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., and Hartley, R. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=eQe8DEWNN2W.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In Dy, J. and Krause, A. (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.

Kim, H., Lee, K., Hwang, G., and Suh, C. 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.

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

Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pp. 2796–2804. PMLR, 2018.

Kull, M., Filho, T. M. S., and Flach, P. 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.

Kull, M., Perelló-Nieto, M., Kängsepp, M., de Menezes e Silva Filho, T., Song, H., and Flach, P. A. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. In *NeurIPS*, 2019.

Kumar, A., Sarawagi, S., and Jain, U. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, 2018.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Lee, C., Zame, W., Yoon, J., and van der Schaar, M. Deephit: A deep learning approach to survival analysis with competing risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11842.

Liang, G., Zhang, Y., Wang, X., and Jacobs, N. Improved trainable calibration method for neural networks on medical imaging classification. In *British Machine Vision Conference (BMVC)*, 2020.

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Maddox, W., Garipov, T., Izmailov, P., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. *CoRR*, abs/1902.02476, 2019. URL http://arxiv.org/abs/1902.02476.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. S., and Dokania, P. Calibrating deep neural networks using focal loss. *ArXiv*, abs/2002.09437, 2020.

Murphy, K. P. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020. URL https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., and Hinton, G. E. Regularizing neural networks by penalizing confident output distributions. *CoRR*, abs/1701.06548, 2017. URL http://arxiv.org/abs/1701.06548.

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

Postels, J., Ferroni, F., Coskun, H., Navab, N., and Tombari, F. Sampling-free epistemic uncertainty estimation using approximated variance propagation. *CoRR*, abs/1908.00598, 2019. URL http://arxiv.org/abs/1908.00598.

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., et al. Skillful precipitation nowcasting using deep generative models of radar. *arXiv preprint arXiv:2104.00954*, 2021.

Richardson, L. F. *Weather prediction by numerical process*. Cambridge university press, 2007.

Shamout, F. E., Shen, Y., Wu, N., Kaku, A., Park, J., Makino, T., Jastrzebski, S., Wang, D., Zhang, B., Dogra, S., Cao, M., Razavian, N., Kudlowitz, D., Azour, L., Moore, W., Lui, Y. W., Aphinyanaphongs, Y., Fernandez-Granda, C., and Geras, K. J. An artificial intelligence system for predicting the deterioration of COVID-19 patients in the emergency department. *CoRR*, abs/2008.01774, 2020. URL https://arxiv.org/abs/2008.01774.

Shekhovtsov, A. and Flach, B. Feed-forward propagation in probabilistic neural networks with categorical and max layers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkMuPjRcKQ.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data, 2018.

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

Tagasovska, N. and Lopez-Paz, D. Single-model uncertainties for deep learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/73c03186765e199c116224b68adc5fa0-Paper.pdf.

Thagaard, J., Hauberg, S., van der Vegt, B., Ebstrup, T., Hansen, J. D., and Dahl, A. B. Can you trust predictive uncertainty under real dataset shifts in digital pathology? In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 824–833. Springer, 2020.

Thulasidasan, S., Chennupati, G., Bilmes, J., Bhattacharya, T., and Michalak, S. On mixup training: Improved calibration and predictive uncertainty for deep neural networks, 2020.

Vershynin, R. *High Dimensional Probability*. Cambridge university press, 2018.

Wang, H., Shi, X., and Yeung, D. Natural-parameter networks: A class of probabilistic neural networks. *CoRR*, abs/1611.00448, 2016. URL http://arxiv.org/abs/1611.00448.

Wulczyn, E., Steiner, D. F., Xu, Z., Sadhwani, A., Wang, H., Flament, I., Mermel, C., Chen, P.-H. C., Liu, Y., and Stumpe, M. C. Deep learning-based survival prediction for multiple cancer types using histopathology images. *PLoS ONE*, 15, 2020.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., and Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2020.

Yao, Q., Yang, H., Han, B., Niu, G., and Kwok, J. T.-Y. Searching to exploit memorization effect in learning with noisy labels. In *International Conference on Machine Learning*, pp. 10789–10798. PMLR, 2020.

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

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR*, 2017a.

Zhang, H., Cissé, M., Dauphin, Y., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412, 2018.

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

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

Zhao, S. and Ermon, S. Right decisions from wrong predictions: A mechanism design alternative to individual calibration. In *International Conference on Artificial Intelligence and Statistics*, pp. 2683–2691. PMLR, 2021.

A Analytical Model

In this appendix, we propose a simple analytical model that helps to demonstrate the early learning and memorization phenomena in probability estimation. Consider the following standard logistic regression problem: we are given i.i.d. data $\{(y_i, \boldsymbol{x_i})\}_{i\leq n}$ where $y_i \in \{+1, -1\}$ are labels[5] and $\boldsymbol{x_i} \sim \mathcal{N}(0, I_p)$ are standard Gaussian covariate vectors. Given $\boldsymbol{x_i} = \boldsymbol{x}$, the label y_i is distributed according to

$$\mathbb{P}(y_i = 1|\boldsymbol{x_i} = \boldsymbol{x}) = \sigma(\langle \boldsymbol{\theta}_p^*, \boldsymbol{x}\rangle), \tag{6}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function and $\boldsymbol{\theta}_p^* \in \mathbb{R}^p$ is the ground-truth parameter with Euclidean norm $\|\boldsymbol{\theta}_p^*\| = \gamma > 0$. Throughout, we assume that we are working in the high dimensional regime: the ratio $\frac{p}{n} \to \kappa \in (0, +\infty)$ as $n \to \infty$. For convenience, we denote N_0 an integer such that $\frac{p}{n} < 2\kappa$ for $\forall n > N_0$. In Section B of the appendix, we provide a numerical example to illustrate this theoretical model.

Given a fresh sample $\boldsymbol{z} \in \mathbb{R}^p$, we predict the probability of the associated label being one via $\sigma(\langle \hat{\boldsymbol{\theta}}, \boldsymbol{z}\rangle)$ for some $\hat{\boldsymbol{\theta}} \in \mathbb{R}^p$, and we train the estimator $\hat{\boldsymbol{\theta}}$ through gradient descent to minimize the cross-entropy:

$$L_n(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\left\{\left(\frac{1+y_i}{2}\right)\log\sigma(\langle\boldsymbol{\theta}, \boldsymbol{x_i}\rangle) + \left(\frac{1-y_i}{2}\right)\log(1-\sigma(\langle\boldsymbol{\theta}, \boldsymbol{x_i}\rangle))\right\} \tag{7}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left\{-\frac{1}{2}y_i\langle\boldsymbol{\theta}, \boldsymbol{x_i}\rangle + \log\left(e^{\frac{1}{2}\langle\boldsymbol{\theta}, \boldsymbol{x_i}\rangle} + e^{-\frac{1}{2}\langle\boldsymbol{\theta}, \boldsymbol{x_i}\rangle}\right)\right\} \tag{8}$$

$$\hat{\boldsymbol{\theta}}_{\eta,n}^{k+1} = \hat{\boldsymbol{\theta}}_{\eta,n}^{k} - \eta\nabla L_n(\hat{\boldsymbol{\theta}}_{\eta,n}^{k}), \quad k = 0, 1, ... \tag{9}$$

where we choose constant step size $\eta < \delta$, for some prefixed $\delta > 0$, to be determined later. For the initialization, we draw $\hat{\boldsymbol{\theta}}_{\eta,n}^{0}$ uniformly from the sphere of radius γ_0.

Define the mean squared error

$$\text{MSE}_{\eta,n}(k) := \mathbb{E}[(\sigma(\langle\hat{\boldsymbol{\theta}}_{\eta,n}^{k}, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle))^2], \tag{10}$$

where $z \sim \mathcal{N}(0, I_p)$ is a fresh Gaussian vector, independent of the data. Note that the expectation is taken with respect to all sources of randomness (including the randomness of data and initialization).

We present two main theorems that summarize our results. The first theorem states that for sufficiently large n and p, the mean squared error decreases during the initial iterates of gradient descent. This phenomenon justifies the name "early learning": during early stages of training, the predictions obtained by the iterates of gradient descent improve.

Theorem A.1. *For any $n > N^*$, the function $k \mapsto MSE_{\eta,n}(k)$ decreases for $k \in [0, \frac{T^*}{\eta})$, where the constants $N^* = N^*(\eta, \gamma, \gamma_0)$ depends on η, γ, γ_0 and $T^* = T^*(\kappa, \gamma, \gamma_0, \delta) > 0$ depends on κ, γ, γ_0, and δ.*

The second theorem, however, states that when the dimension is sufficiently large (with respect to the number of samples), the predictor eventually overfits, and converges to a predictor that outputs only the probabilities 0 and 1.

Theorem A.2. *Let $p(\boldsymbol{z}) = \lim_{k\to\infty}\sigma(\langle\hat{\boldsymbol{\theta}}_{\eta,n}^{k}, \boldsymbol{z}\rangle)$. Then there exists a $\kappa^* = \kappa^*(\gamma)$, such that when $n \to +\infty$ and $\frac{p}{n} \to \kappa > \kappa^*$, we have*

$$\mathbb{P}(\{p(z) \in \{0, 1\} \text{ for a.e. } \boldsymbol{z} \in \mathbb{R}^p\}) \to 1.$$

We begin with the proof of Theorem A.2.

Proof of Theorem A.2. Define the event

$$S_n = \{\exists \boldsymbol{w} \in \mathbb{R}^p \text{ such that for } \forall i \leq n : y_i\langle\boldsymbol{w}, \boldsymbol{x}_i\rangle > 0\}.$$

[5] For mathematical convenience, in this appendix we work with labels in $\{\pm 1\}$ rather than $\{0, 1\}$.

For some large $C_G > 0$ to be determined later, define the event

$$G_n = \{\|X\|_2 \leq C_G(\sqrt{n} + \sqrt{p})\},$$

where $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathbb{R}^{p\times n}$ is the covariate matrix and $\|X\|_2$ denotes the maximum singular value of X. Pick $\delta < \frac{2}{C_G^2(\sqrt{\kappa}+1)^2}$.

We claim that $S_n \cap G_n \subset E_n = \{p(\boldsymbol{z}) \in \{0, 1\} \text{ for a.e. } \boldsymbol{z} \in \mathbb{R}^p\}$. Indeed, assume that S_n and G_n happen. Then the step size η satisfies

$$\eta < \delta < \frac{2}{C_G^2(\sqrt{\kappa}+1)^2} < \frac{2n}{\|X\|_2^2}.$$

On the event S_n, the data points are separable, and Soudry et al. (2018, Theorem 3) implies that as $k \to \infty$,

$$\left\|\hat{\boldsymbol{\theta}}^k_{\eta,n}\right\|_2 \to +\infty,$$
$$\frac{\hat{\boldsymbol{\theta}}^k_{\eta,n}}{\left\|\hat{\boldsymbol{\theta}}^k_{\eta,n}\right\|_2} \to \hat{\boldsymbol{\theta}}^{MM},$$

where $\hat{\boldsymbol{\theta}}^{MM} \in \mathbb{R}^p$ denotes the max-margin separator. Thus for any $\boldsymbol{z} \in \mathbb{R}^p$ with $\langle \boldsymbol{z}, \hat{\boldsymbol{\theta}}^{MM}\rangle > 0$ (resp. < 0), we have $\langle z, \hat{\boldsymbol{\theta}}^k_{\eta,n}\rangle \to +\infty$ (resp. $-\infty$), and thus $p(\boldsymbol{z}) = \lim_{k\to\infty} \sigma(\langle \hat{\boldsymbol{\theta}}^k_{\eta,n}, \boldsymbol{z}\rangle) = 1$ (resp. $= 0$). Since $\{\boldsymbol{z} \in \mathbb{R}^p : \langle \boldsymbol{z}, \hat{\boldsymbol{\theta}}^{MM}\rangle = 0\}$ has Lebesgue measure 0, we have shown that $S_n \cap G_n \subset E_n$.

It remains to show that S_n and G_n hold with probability tending to one. Candes & Sur (2018, Theorem 1) show that there exists a finite threshold κ^*, such that when $\frac{p}{n} \to \kappa > \kappa^*$, it holds that $\mathbb{P}(S_n) \to 1$. Moreover, by standard results from random matrix theory [see e.g. (Vershynin, 2018), Theorem 4.4.5], there exists some constant C_G such that $\mathbb{P}(G_n) \geq 1 - 2e^{-n} \to 1$. Therefore, we can conclude that

$$\mathbb{P}(E_n) \geq \mathbb{P}(S_n \cap G_n) \to 1.$$

□

In the remainder of the appendix, we prove Theorem A.1. Consider the following gradient flow, which is the analogue of gradient descent in the $\eta \to 0$ limit.

$$\frac{d\hat{\boldsymbol{\theta}}^t_n}{dt} = -\nabla L_n(\hat{\boldsymbol{\theta}}^t_n), \quad \forall t \geq 0. \tag{11}$$

Set $\hat{\boldsymbol{\theta}}^0_n = \hat{\boldsymbol{\theta}}^0_{\eta,n}$ and define the corresponding mean square error

$$\text{MSE}_n(t) := \mathbb{E}[(\sigma(\langle \hat{\boldsymbol{\theta}}^t_n, \boldsymbol{z}\rangle) - \sigma(\langle \boldsymbol{\theta}^*_p, \boldsymbol{z}\rangle))^2]. \tag{12}$$

We will first show that the mean squared error decreases along the trajectory of gradient flow, and then establish that this result extends to the iterates of gradient descent. We denote $\text{MSE}'_n(t)$ as the derivative of MSE_n at time $t \geq 0$.

Lemma A.3. *(a)* $\lim_{n\to\infty} MSE'_n(0) < 0$.

(b) There exists some positive constants $N_1(\gamma, \gamma_0)$, $T_1(\gamma, \gamma_0)$ and $C_1(\gamma, \gamma_0)$, such that for any $n > N_1$ and $t < T_1$, it holds that

$$MSE'_n(t) \leq -C_1.$$

Proof. (a) For convenience, we write equation (11) in matrix form

$$\frac{d\hat{\boldsymbol{\theta}}^t_n}{dt} = \frac{1}{n} X(\boldsymbol{y}' - \sigma(X^T \hat{\boldsymbol{\theta}}^t_n)),$$

where $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathbb{R}^{p\times n}$, $\boldsymbol{y}' = (\frac{y_1+1}{2}, ..., \frac{y_n+1}{2})^T \in \mathbb{R}^n$ and $\sigma(X^T\hat{\boldsymbol{\theta}}_n^t) = (\sigma(\langle \boldsymbol{x}_1, \hat{\boldsymbol{\theta}}_n^t\rangle), ..., \sigma(\langle \boldsymbol{x}_n, \hat{\boldsymbol{\theta}}_n^t\rangle))^T \in \mathbb{R}^n$. We first calculate the derivative of the mean square prediction error:

$$\begin{aligned}
\text{MSE}_n'(t) &= \mathbb{E}\left[\left(\sigma(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle)\right)\sigma'(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle)\langle\frac{d\hat{\boldsymbol{\theta}}_n^t}{dt}, \boldsymbol{z}\rangle\right] \\
&= \mathbb{E}\left[\left(\sigma(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle)\right)\sigma'(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle)\frac{1}{n}(\boldsymbol{y}' - \sigma(X^T\hat{\boldsymbol{\theta}}_n^t))^T X^T z\right] \\
&= \mathbb{E}\left[\left(\sigma(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle)\right)\sigma'(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle)\frac{1}{n}(\sigma(X^T\boldsymbol{\theta}_p^*) - \sigma(X^T\hat{\boldsymbol{\theta}}_n^t))^T X^T z\right],
\end{aligned}$$

where in the first line we used dominated convergence theorem, and in the last line we used $\mathbb{E}[\boldsymbol{y}'|X] = \sigma(X^T\boldsymbol{\theta}_p^*)$.

Thus,

$$\begin{aligned}
\text{MSE}_n'(0) &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left(\sigma(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle)\right)\sigma'(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle)\left(\sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{x}_i\rangle) - \sigma(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{x}_i\rangle)\right)\langle\boldsymbol{x}_i, \boldsymbol{z}\rangle\right] \\
&= \mathbb{E}\left[\left(\sigma(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle)\right)\sigma'(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle)\left(\sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{x}_1\rangle) - \sigma(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{x}_1\rangle)\right)\langle\boldsymbol{x}_1, \boldsymbol{z}\rangle\right],
\end{aligned}$$

where $\hat{\boldsymbol{\theta}}_n^0$, $\boldsymbol{z}$, and $\boldsymbol{x}_1$ are mutually independent. Let $\mu(b) := \mathbb{E}[\sigma'(Z_b)]$, where the scalar $Z_b \sim \mathcal{N}(0, b^2)$. Conditioned on $(\boldsymbol{z}, \hat{\boldsymbol{\theta}}_n^0)$, the pair $(\langle\boldsymbol{x}_1, \boldsymbol{z}\rangle, \langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{x}_1\rangle)$ is multivariate Gaussian with covariance $\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle$. By the Gaussian integration by parts formula,

$$\mathbb{E}[\sigma(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{x}_1\rangle)\langle\boldsymbol{x}_1, \boldsymbol{z}\rangle|\boldsymbol{z}, \hat{\boldsymbol{\theta}}_n^0] = \mathbb{E}[\sigma'(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{x}_1\rangle)|\boldsymbol{z}, \hat{\boldsymbol{\theta}}_n^0]\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle = \mu(\gamma_0)\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle.$$

Similarly,

$$\mathbb{E}[\sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{x}_1\rangle)\langle\boldsymbol{x}_1, \boldsymbol{z}\rangle|\boldsymbol{z}, \hat{\boldsymbol{\theta}}_n^0] = \mu(\gamma)\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle.$$

So

$$\text{MSE}_n'(0) = \mathbb{E}\left[\left(\sigma(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle)\right)\sigma'(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle)\left(\mu(\gamma)\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle - \mu(\gamma_0)\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle\right)\right],$$

Since $\hat{\boldsymbol{\theta}}_n^0 \sim \text{Unif}(\mathbb{S}^{p-1}(\gamma_0))$, we have $\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{\theta}_p^*\rangle \to 0$ in probability as $p \to \infty$. Thus, the pair $(\langle\hat{\boldsymbol{\theta}}_n^0, \boldsymbol{z}\rangle, \langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle) \xrightarrow{d} (Z_0, Z)$, where Z_0 and Z are independent centered Gaussian random variables with variance γ_0 and γ, respectively.

We hence compute

$$\begin{aligned}
\lim_{p\to\infty}\text{MSE}_n'(0) &= \mathbb{E}[(\sigma(Z_0) - \sigma(Z))\sigma'(Z_0)(\mu(\gamma)Z - \mu(\gamma_0)Z_0)] \\
&= -\gamma^2\mu(\gamma)^2\mu(\gamma_0) - \mu(\gamma_0)\mathbb{E}[(\sigma(Z_0) - \frac{1}{2})\sigma'(Z_0)Z_0],
\end{aligned}$$

where in the last inequality we used Gaussian integration by parts and $\mathbb{E}[\sigma(Z)] = \frac{1}{2}$. Since $\mu(b) > 0$ for all $b \geq 0$ and $(\sigma(x) - \frac{1}{2})\sigma'(x)x \geq 0$ for all $x \in \mathbb{R}$, we conclude that $\lim_{n\to\infty}\text{MSE}_n'(0) < 0$.

(b) We first show that the second derivative of MSE_n is uniformly bounded. Write $f(x) = (\sigma(x) - \sigma(\langle\boldsymbol{\theta}_p^*, \boldsymbol{z}\rangle))^2$. Since σ, σ' and σ'' are all bounded, we have

$$\begin{aligned}
|\text{MSE}_n''(t)| &= \left|\mathbb{E}\left[f''(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle)\langle\frac{d\hat{\boldsymbol{\theta}}_n^t}{dt}, \boldsymbol{z}\rangle^2\right] + \mathbb{E}\left[f'(\langle\hat{\boldsymbol{\theta}}_n^t, \boldsymbol{z}\rangle)\langle\frac{d^2\hat{\boldsymbol{\theta}}_n^t}{dt^2}, \boldsymbol{z}\rangle\right]\right| \\
&\leq C_1\left(\mathbb{E}\left[\langle\frac{d\hat{\boldsymbol{\theta}}_n^t}{dt}, \boldsymbol{z}\rangle^2\right] + \mathbb{E}\left[\left|\langle\frac{d^2\hat{\boldsymbol{\theta}}_n^t}{dt^2}, \boldsymbol{z}\rangle\right|\right]\right) \\
&\leq C_2\left(\mathbb{E}\left[\left\|\frac{d\hat{\boldsymbol{\theta}}_n^t}{dt}\right\|_2^2\right] + \mathbb{E}\left[\left\|\frac{d^2\hat{\boldsymbol{\theta}}_n^t}{dt^2}\right\|_2\right]\right),
\end{aligned}$$

where in the first equality, we use dominated convergence to interchange differentiation and expectation, and in the second inequality we used the fact that $\boldsymbol{z} \sim \mathcal{N}(0, I_p)$ is independent of the data.

By a simple calculation,

$$\left\| \frac{d\hat{\boldsymbol{\theta}}_n^t}{dt} \right\|_2 \leq \frac{1}{n} \|X\|_2 \left\| \boldsymbol{y}' - \sigma(X^T \hat{\boldsymbol{\theta}}_n^t) \right\|_2 \leq \frac{2\sqrt{p}\,\|X\|_2}{n},$$
$$\left\| \frac{d^2\hat{\boldsymbol{\theta}}_n^t}{dt^2} \right\|_2 = \left\| \frac{1}{n^2} X D_t X^T X (\boldsymbol{y}' - \sigma(X^T \hat{\boldsymbol{\theta}}_n^t)) \right\|_2 \leq \frac{2\sqrt{p}\,\|X\|_2^3}{n^2},$$

where $D_t = \mathrm{Diag}(\sigma'(\langle \hat{\boldsymbol{\theta}}_n^t, \boldsymbol{x}_i \rangle))_{i=1}^n$ is a diagonal matrix with $\|D_t\|_2 \leq 1$. By standard random matrix results [see e.g. (Vershynin, 2018), Theorem 4.4.5],

$$\mathbb{E}[\|X\|_2^2] \leq C(\sqrt{p} + \sqrt{n})^2, \quad \mathbb{E}[\|X\|_2^3] \leq C(\sqrt{p} + \sqrt{n})^3.$$

Whence, for any $t \geq 0$ and $n > N_0$,

$$|\mathrm{MSE}_n''(t)| \leq C_3 \left(\frac{p(\sqrt{p} + \sqrt{n})^2}{n^2} + \frac{\sqrt{p}(\sqrt{p} + \sqrt{n})^3}{n^2} \right) := L,$$

where $L = C_4(k^2 + 1)$ is a constant that only depends on κ.

Let $\tau = \tau(\gamma, \gamma_0) = \lim_{n \to \infty} \mathrm{MSE}_n'(0)$, and let $N_1 = N_1(\gamma, \gamma_0) > N_0$ satisfy $\mathrm{MSE}_n'(0) \leq \frac{\tau}{2}$ for any $n > N_1$. By Claim (a), $\tau < 0$. Then for any $t \leq T_1(\gamma, \gamma_0) = \frac{-\tau(\gamma, \gamma_0)}{4L}$,

$$\mathrm{MSE}_n'(t) = \mathrm{MSE}_n'(0) + (\mathrm{MSE}_n'(t) - \mathrm{MSE}_n'(0)) \leq \frac{\tau}{2} + Lt \leq \frac{\tau}{4}.$$

□

Lemma A.1 essentially shows that within a small constant time window, MSE_n' is negative, and thus the mean square error of gradient flow decreases. The next lemma shows that the mean square error of gradient flow is a good approximation of the mean square error of gradient descent. Before stating the next lemma, we extend gradient descent (9) to a piecewise linear function, i.e.

$$\frac{d\tilde{\boldsymbol{\theta}}_{\eta,n}^t}{dt} = -\nabla L_n(\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor t \rfloor}), \quad \forall t \geq 0, \tag{13}$$

where $\lfloor t \rfloor := \max\{k\eta : k\eta \leq t, k \in \mathbb{N}\}$. In particular, $\tilde{\boldsymbol{\theta}}_{\eta,n}^{k\eta} = \hat{\boldsymbol{\theta}}_{\eta,n}^k$.

Lemma A.4. *(a) For any $t > 0$ and $n > N_0$,*

$$\sup_{s \leq t} \|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\| \leq C_2(\kappa, t, \delta, \gamma_0)\eta t.$$

with probability $1 - 2e^{-n}$, where $C_2(\kappa, t, \delta, \gamma_0) > 0$ is a constant that depends on κ, t, δ and γ_0.
(b) For any $t > 0$ and $n > N_0$,

$$\sup_{k \leq t/\eta} |\mathit{MSE}_n(k\eta) - \mathit{MSE}_{\eta,n}(k)| \leq 4C_2(\kappa, t, \delta, \gamma_0)\eta t + 8e^{-n}.$$

Proof. (a) We first bound $\sup_{s \leq t} \|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\|$.

Compute

$$\begin{aligned}
\frac{d}{dt} \left\| \hat{\boldsymbol{\theta}}_n^t \right\|_2 &\leq \left\| \frac{d\hat{\boldsymbol{\theta}}_n^t}{dt} \right\|_2 \\
&= \left\| \frac{1}{n} X(\boldsymbol{y}' - \sigma(\boldsymbol{0})) + \frac{1}{n} X(\sigma(\boldsymbol{0}) - \sigma(X^T \hat{\boldsymbol{\theta}}_n^t)) \right\|_2 \\
&\leq \frac{2\sqrt{p}\,\|X\|_2}{n} + \frac{\|X\|_2^2 \,\|\hat{\boldsymbol{\theta}}_n^t\|_2}{n},
\end{aligned}$$

where we used $\|\sigma(\boldsymbol{x}) - \sigma(\boldsymbol{y})\|_2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$. By Grönwall's inequality,

$$\left\|\hat{\boldsymbol{\theta}}_n^t\right\|_2 \leq e^{\frac{\|X\|_2^2}{n}t}\left(\left\|\hat{\boldsymbol{\theta}}_n^0\right\|_2 + \frac{2\sqrt{p}\,\|X\|_2}{n}t\right). \tag{14}$$

Next, compute

$$\begin{aligned}
\frac{d}{dt}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 &\leq \left\|\frac{d}{dt}\left(\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right)\right\|_2 \\
&= \left\|\frac{1}{n}X(\sigma(X^T\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor}) - \sigma(X^T\hat{\boldsymbol{\theta}}_n^s)\right\|_2 \\
&\leq \frac{\|X\|_2^2}{n}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor} - \hat{\boldsymbol{\theta}}_n^s\right\|_2 \\
&\leq \frac{\|X\|_2^2}{n}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 + \frac{\|X\|_2^2}{n}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor} - \tilde{\boldsymbol{\theta}}_{\eta,n}^s\right\|_2 \\
&\leq \frac{\|X\|_2^2}{n}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 + \frac{\|X\|_2^2}{n}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor} - \tilde{\boldsymbol{\theta}}_{\eta,n}^s\right\|_2.
\end{aligned}$$

Since $\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor} = (s - \lfloor s\rfloor)\frac{1}{n}X(\boldsymbol{y}' - \sigma(X^T\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor}))$, we can bound

$$\begin{aligned}
\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor}\right\|_2 &\leq \frac{2\eta\sqrt{p}\,\|X\|_2}{n} + \frac{\eta\,\|X\|_2^2\,\|\tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor}\|_2}{n} \\
&\leq \frac{2\eta\sqrt{p}\,\|X\|_2}{n} + \frac{\eta\,\|X\|_2^2\,\|\hat{\boldsymbol{\theta}}_n^{\lfloor s\rfloor}\|_2}{n} + \frac{\eta\,\|X\|_2^2}{n}\left\|\hat{\boldsymbol{\theta}}_n^{\lfloor s\rfloor} - \tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor s\rfloor}\right\|_2.
\end{aligned}$$

Combining the above inequalities, we get

$$\begin{aligned}
\frac{d}{dt}\sup_{s\leq t}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 &\leq \frac{d}{dt}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^t - \hat{\boldsymbol{\theta}}_n^t\right\|_2 \\
&\leq \frac{\|X\|_2^2}{n}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^t - \hat{\boldsymbol{\theta}}_n^t\right\|_2 + \frac{\|X\|_2^2}{n}\left(\frac{2\eta\sqrt{p}\,\|X\|_2}{n} + \frac{\eta\,\|X\|_2^2\,\|\hat{\boldsymbol{\theta}}_n^{\lfloor t\rfloor}\|_2}{n} + \frac{\eta\,\|X\|_2^2}{n}\left\|\hat{\boldsymbol{\theta}}_n^{\lfloor t\rfloor} - \tilde{\boldsymbol{\theta}}_{\eta,n}^{\lfloor t\rfloor}\right\|_2\right) \\
&\leq \left(\frac{\|X\|_2^2}{n} + \frac{\delta\,\|X\|_2^4}{n^2}\right)\sup_{s\leq t}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 + \eta\left(\frac{2\sqrt{p}\,\|X\|_2^3}{n^2} + \frac{\|X\|_2^4\,\|\hat{\boldsymbol{\theta}}_n^{\lfloor t\rfloor}\|}{n^2}\right).
\end{aligned}$$

Together with inequality (14), by Grönwall's inequality we eventually have

$$\sup_{s\leq t}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 \leq \eta t\left(\frac{2\sqrt{p}\,\|X\|_2^3}{n^2} + \frac{\gamma_0\,\|X\|_2^4\,e^{\frac{\|X\|_2^2}{n}t}}{n^2} + \frac{2t\sqrt{p}\,\|X\|_2^5\,e^{\frac{\|X\|_2^2}{n}t}}{n^3}\right)e^{(\frac{\|X\|_2^2}{n} + \frac{\delta\|X\|_2^4}{n^2})t}. \tag{15}$$

Again, by a standard random matrix result [see e.g. (Vershynin, 2018), Theorem 4.4.5], it holds that $\|X\|_2 \leq C(\sqrt{p} + \sqrt{n})$, with probability $1 - 2e^{-n}$. Plug this bound into inequality (15), we get for any $n > N_0$,

$$\sup_{s\leq t}\left\|\tilde{\boldsymbol{\theta}}_{\eta,n}^s - \hat{\boldsymbol{\theta}}_n^s\right\|_2 \leq \eta t C_2(\kappa, t, \delta, \gamma_0), \tag{16}$$

with probability $1 - 2e^{-n}$, where $C_2(\kappa, t, \delta, \gamma_0) = C(t + \gamma_0 + 1)(\kappa^6 + 1)e^{C(\delta+1)(\kappa^2+1)t}$, for some $C > 0$.

(b) Define the event $A_n = \left\{ \sup_{s\le t} \left\| \tilde{\boldsymbol{\theta}}^s_{\eta,n} - \hat{\boldsymbol{\theta}}^s_n \right\|_2 \le \eta t C_2(\kappa, t, \delta, \gamma_0) \right\}$. Since $|\sigma|, |\sigma'| < 1$, we have for any $n > N_0$,

$$\begin{aligned}
\sup_{k\le t/\eta} |\text{MSE}_n(k\eta) - \text{MSE}_{\eta,n}(k)| &\le \sup_{k\le t/\eta} 4\mathbb{E}[|\sigma(\langle \hat{\boldsymbol{\theta}}^{k\eta}_n, \boldsymbol{z}\rangle) - \sigma(\langle \hat{\boldsymbol{\theta}}^{k}_{\eta,n}, \boldsymbol{z}\rangle)|] \\
&= \sup_{k\le t/\eta} 4\mathbb{E}[|\sigma(\langle \hat{\boldsymbol{\theta}}^{k\eta}_n, \boldsymbol{z}\rangle) - \sigma(\langle \hat{\boldsymbol{\theta}}^{k}_{\eta,n}, \boldsymbol{z}\rangle)| \mathbb{1}_{A_n}] + 8e^{-n} \\
&\le \sup_{k\le t/\eta} 4\mathbb{E}\left[\left\| \hat{\boldsymbol{\theta}}^{k\eta}_n - \hat{\boldsymbol{\theta}}^{k}_{\eta,n} \right\|_2 \mathbb{1}_{A_n} \right] + 8e^{-n} \\
&\le 4\mathbb{E}\left[\sup_{k\le t/\eta} \left\| \hat{\boldsymbol{\theta}}^{k\eta}_n - \hat{\boldsymbol{\theta}}^{k}_{\eta,n} \right\|_2 \mathbb{1}_{A_n} \right] + 8e^{-n} \\
&= 4\mathbb{E}\left[\sup_{k\le t/\eta} \left\| \hat{\boldsymbol{\theta}}^{k\eta}_n - \tilde{\boldsymbol{\theta}}^{k\eta}_{\eta,n} \right\|_2 \mathbb{1}_{A_n} \right] + 8e^{-n} \\
&\le 4C_2(\kappa, t, \delta, \gamma_0) t\eta + 8e^{-n}
\end{aligned}$$

□

We are now ready to prove Theorem A.1.

Proof of Theorem A.1. Pick a small $T_2(\kappa, \delta, \gamma_0) > 0$ that satisfy $C_2(\kappa, T_2, \delta, \gamma_0) T_2 \le \frac{C_1(\gamma,\gamma_0)}{32}$. This is possible since $C_2(\kappa, t, \delta, \gamma_0)$ is bounded as $t \to 0$. Let N_1 and T_1 be as in Lemma A.3, and let $T^*(\gamma, \gamma_0, \kappa, \delta) = \min\{T_1(\gamma, \gamma_0), T_2(\kappa, \delta, \gamma_0)\}$. For any $k < \frac{T^*}{\eta} - 1$,

$$\begin{aligned}
\text{MSE}_{\eta,n}(k+1) - \text{MSE}_{\eta,n}(k) &= (\text{MSE}_n((k+1)\eta) - \text{MSE}_n(k\eta)) + (\text{MSE}_{\eta,n}(k+1) - \text{MSE}_n((k+1)\eta)) \\
&\quad + (\text{MSE}_{\eta,n}(k) - \text{MSE}_n((k)\eta)) \\
&\le (\text{MSE}_n((k+1)\eta) - \text{MSE}_n(k\eta)) + 2 \sup_{m\le T^*/\eta} |\text{MSE}_n(m\eta) - \text{MSE}_{\eta,n}(m)|
\end{aligned}$$

By Lemma A.3.(b) and the mean value theorem, for any $n > N_1$,

$$\text{MSE}_n((k+1)\eta) - \text{MSE}_n(k\eta) \le -C_1(\gamma, \gamma_0)\eta.$$

By Lemma A.4.(b), it holds that for any $n > N_0$,

$$\begin{aligned}
\sup_{m\le T^*/\eta} |\text{MSE}_n(m\eta) - \text{MSE}_{\eta,n}(m)| &\le 4C_2(\kappa, T^*, \delta, \gamma_0) T^* \eta + 8e^{-n} \\
&\le \frac{C_1(\gamma, \gamma_0)\eta}{8} + 8e^{-n}.
\end{aligned}$$

Let $N_2(\eta, \gamma, \gamma_0)$ satisfy $8e^{-N_2} < \frac{C_1(\gamma,\gamma_0)\eta}{8}$, and let $N^*(\eta, \gamma, \gamma_0) = \max\{N_1(\gamma, \gamma_0), N_2(\eta, \gamma, \gamma_0), \}$. We therefore have for any $n > N^*$ and $k < \frac{T^*}{\eta} - 1$,

$$\begin{aligned}
\text{MSE}_{\eta,n}(k+1) - \text{MSE}_{\eta,n}(k) &\le -C_1(\gamma, \gamma_0)\eta + \frac{C_1(\gamma, \gamma_0)\eta}{4} + 16e^{-N^*} \\
&= -\frac{C_1(\gamma, \gamma_0)\eta}{2} < 0.
\end{aligned}$$

□

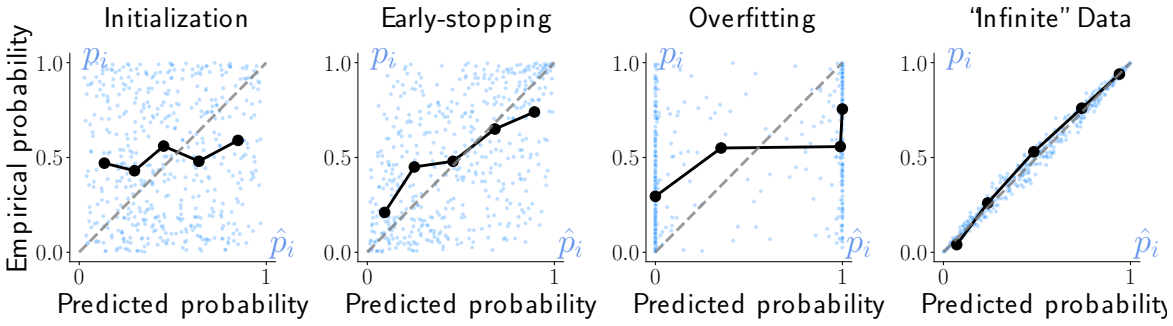

Figure 7. **Reliability curves for a linear model (Appendix B)**. The horizontal and vertical coordinate of each blue point represent the predicted ($\hat{p}_i$) and true (p_i) probabilities of a test example respectively. We also show a reliability diagram of binned mean predicted and empirical probabilities in black (see Section 3). The dashed diagonal line indicates perfect calibration. The model is initialized randomly (first column) and initially improves the estimation, with the reliability curve trending towards the diagonal line (second column). However, because the dataset is finite, it eventually overfits, and the predicted probabilities collapse to 0 or 1 (third column). When labels are resampled to generate large amount of data, the estimates converge to the ground-truth probability labels (right column).

B Early Learning and Memorization in a Linear Model

We provide a numerical example that illustrates the theoretical results of Section 4, and demonstrates the similarities between the behavior of linear and deep-learning models in high dimensions.

We train a logistic regression model in an overparametrized regime ($\frac{p}{n} = 1$). To generate the features, we draw $\{\boldsymbol{x}_i\}_{i=1}^{500}$, $\boldsymbol{x}_i \in \mathbb{R}^{500}$ i.i.d. Gaussian random variables, $\boldsymbol{x}_i \sim \mathcal{N}(0, I_{500})$. The ground-truth unobserved probability labels p_i are generated according to equation 6. i.e. $p_i = \sigma\left(\langle \boldsymbol{\theta}^*, \boldsymbol{x}_i \rangle\right)$. The true parameter $\boldsymbol{\theta}^* = (1, 0, \ldots, 0)$ is fixed to equal the first standard basis vector, and $\sigma(x) = (1 + e^x)^{-1}$ is the sigmoid function.

The 0-1 labels $\{y_i\}_{i=1}^{500}$ are drawn from Bernoulli distribution with probability p_i, $y_i \sim \text{Bern}(p_i)$.

We use stochastic gradient descent (to fit a logistic regression model with parameter $\boldsymbol{\theta} \in \mathbb{R}^{500}$ on the simulated data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{500}$, using a cross-entropy loss function. We compare the model trained on two datasets: (a) finite $(\boldsymbol{x}, y)$ pairs, and (b) a large amount of data set, generated by repeatedly resampling new outcomes y_i from the ground-truth probabilities p_i at every iteration.

Figure 7 shows that the linear model trained with cross entropy begins by improving the estimate of the true probabilities (at the early-learning stage), but eventually memorizes the 0-1 labels (the overfitting stage). Figure 8 illustrates that the MSE_p of cross entropy training increases when the model starts memorizing the 0-1 labels, as predicted by the results of Appendix A.

C Additional Results

We present here supplementary results to the ones presented in Section 8.

C.1 Face-based Risk Prediction

Full evaluation with confidence intervals derived using 1000 bootstraps for the five simulated scenarios are examined: *Linear* (Table.3); *Sigmoid* (Table.4); *Centered* (Table.5); *Skewed* (Table.6); *Discrete* (Table.7). Note that all numbers are upscaled by 10^{-2} in the tables.

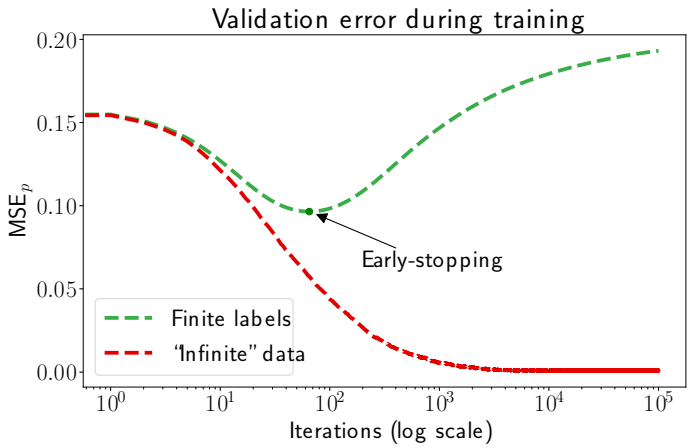

Figure 8. **Validation loss for training a linear model (Appendix B).** Training with finite data (green line) decreases MSE_p initially but eventually overfits to the 0-1 labels. Training with resampled labels (red line) avoids overfitting and results in accurate estimation of the true probabilities.

Table 3: Performance on Face-based Risk Prediction. *Linear* scenario. All numbers are downscaled by 10^{-2}.

Linear	ECE	MCE	KS	Brier	MSE_p	KL_p
CE + label resampling	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

C.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); Collision Prediction (Table 10).

Table 4: Performance on Face-based Risk Prediction. *Sigmoid* scenario. All numbers are downscaled by 10^{-2}.

Sigmoid	ECE	MCE	KS	Brier	MSE_p	KL_p
CE + label resampling	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 5: Performance on Face-based Risk Prediction. *Centered* scenario. All numbers are downscaled by 10^{-2}.

Centered	ECE	MCE	KS	Brier	MSE_p	KL_p
CE + label resampling	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

C.3 Additional Reliability Diagram

Figure 9 shows additional reliability curves on the real world data, supplementing the ones illustrated in Figure 6. Figure 10 shows reliability curves for the different synthetic data scenarios.

D Kolmogorov-Smirnov Error

We derive the KS-error, mentioned in Section 3.

For a calibrated estimator

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

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

Hence

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

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

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

Table 6: Performance on Face-based Risk Prediction. *Skewed* scenario. All numbers are downscaled by 10^{-2}.

Skewed	ECE	MCE	KS	Brier	MSE_p	KL_p
CE + label resampling	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 7: Performance on Face-based Risk Prediction. *Discrete* scenario. All numbers are downscaled by 10^{-2}.

Discrete	ECE	MCE	KS	Brier	MSE_p	KL_p
CE + label resampling	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

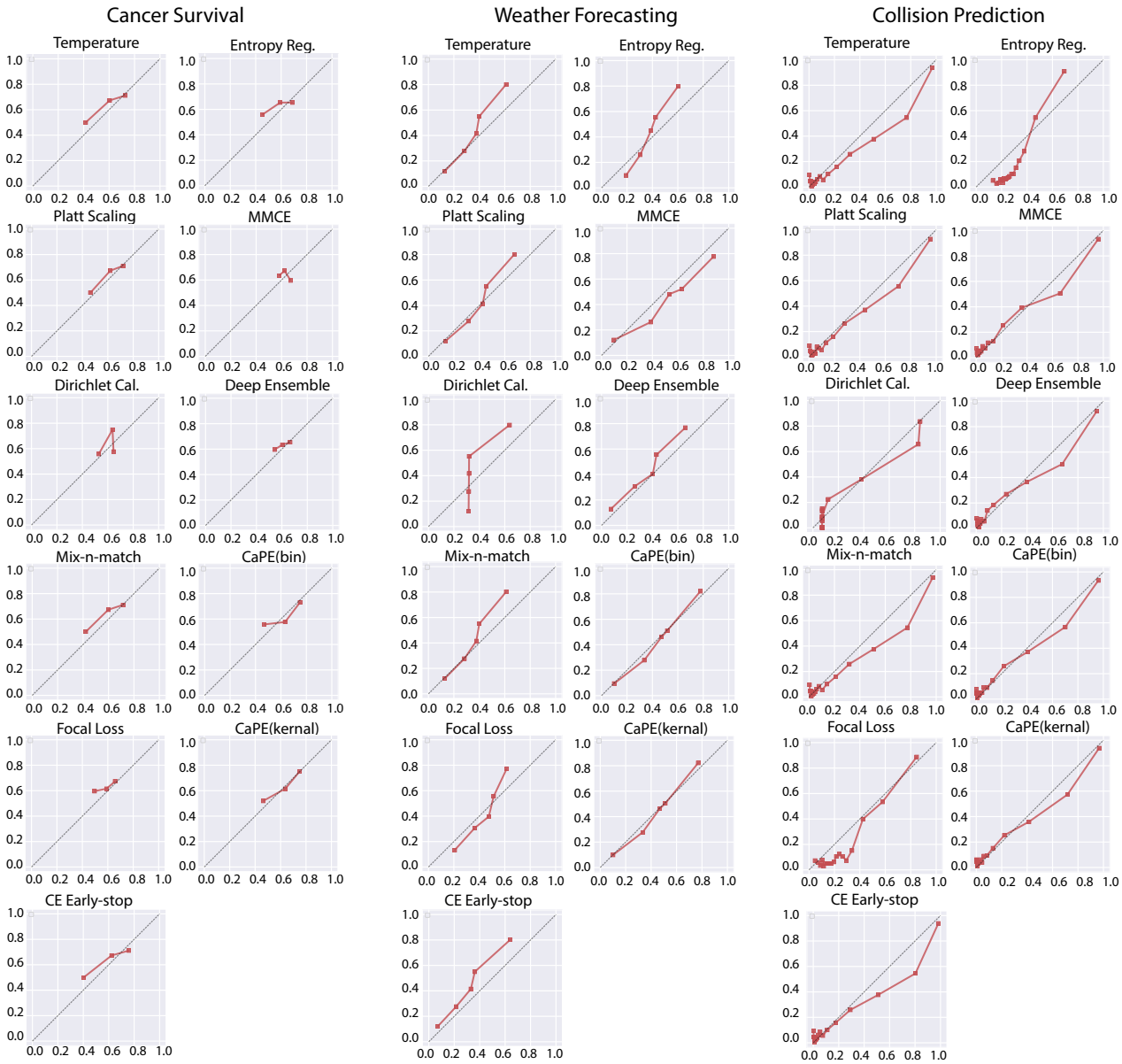

Figure 9. Reliability diagrams of all the baselines on the real-world datasets. We train all baseline methods on each of the datasets and plot the empirical probability(y-axis) against predicted probability(x-axis). The axis labels are removed to ease visualization (they are the same as in Figure 6).

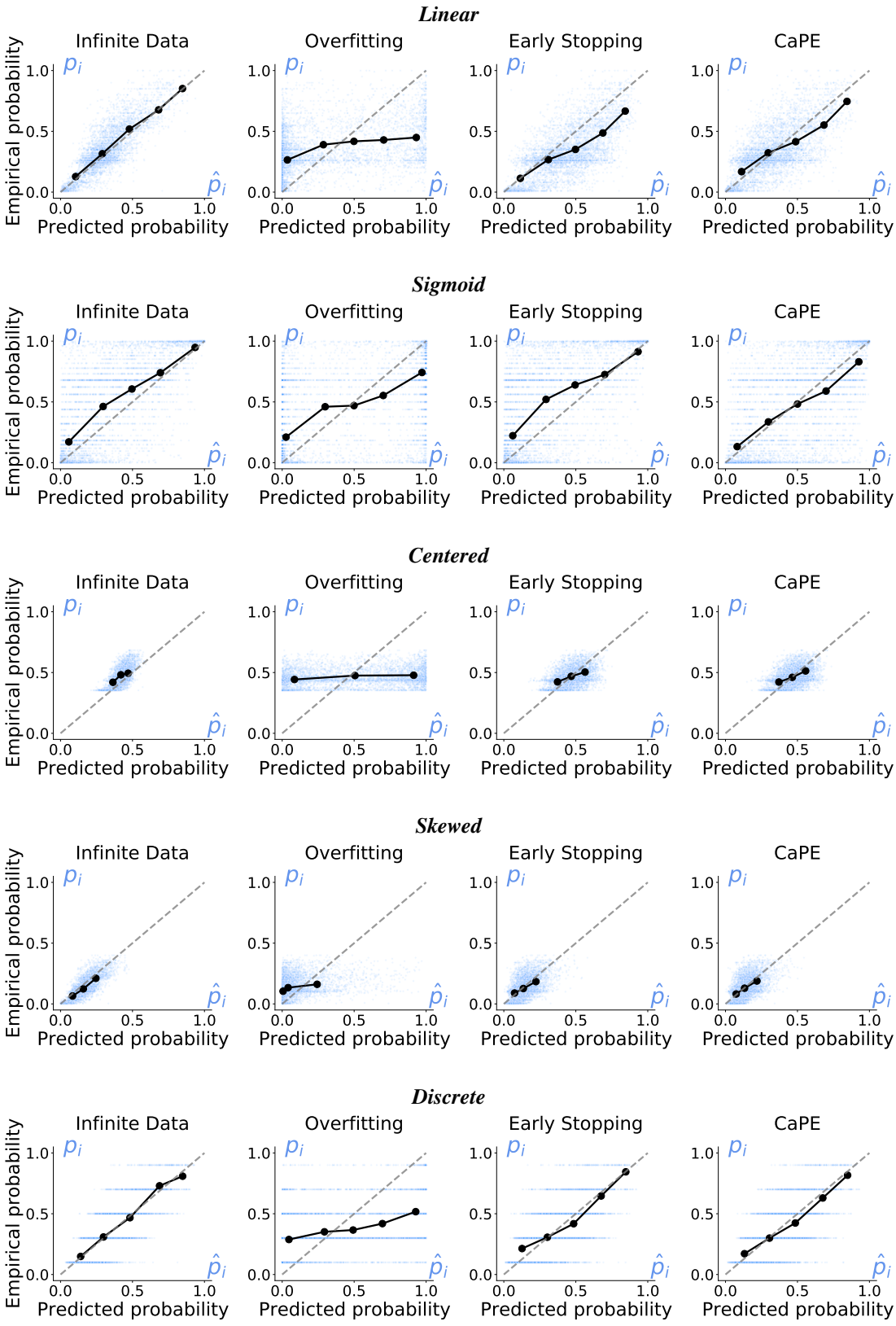

Figure 10. Reliability diagrams for different synthetic data scenarios. We can see that CaPE outperforms early stopping, prevents overfitting, and achieves a performance on par with training on large amount of resampled data.

Table 8: Baselines with full metrics for cancer survival. All numbers are downscaled by 10^{-2}.

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 9: Baselines with full metrics for weather prediction. All numbers are downscaled by 10^{-2}.

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 10: Baselines with full metrics for collision prediction. All numbers are downscaled by 10^{-2}.

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)	85.70	3.16	12.21	30.61	8.18	2.13
CaPE (kernel)	**85.95**	3.22	13.32	30.44	**8.13**	2.10

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

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

The KS error is defined as

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

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.

E Brier Score Decomposition

We present here a decomposition of 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 can output one of 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 $\hat{q}_k$ and $\bar{q}_k$ the empirical probability over S_k, 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 empirical probabilities $\bar{q}_k$ substituting for the true labels. The second term, refinement, is an estimate of the confidence in determining $\bar{q}_k$. 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 $\bar{q}_k$ tend towards 0 or 1. Thus, this term penalizes empirically calibrated predictors, with low discriminative power, as in Figure 2.

F Metric Comparison

Figure 11 shows the correlation between different calibration and accuracy metrics, and two gold-standard metric that use ground truth probabilities: MSE_p and KL-divergence. The correlations are computed using all five scenarios in our Face-based Risk Prediction synthetic dataset.

G 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 5.

We wish to estimate 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 $\hat{p} \in I(q)$,

$$\mathbb{P}(y = 1 | f(\boldsymbol{x}) \in I(q)) \approx \frac{1}{|I(q)|} \sum_{\hat{p} \in I(q)} \mathbb{P}(y = 1 | f(\boldsymbol{x}) = p). \tag{17}$$

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

$$\mathbb{P}(y = 1 | f(\boldsymbol{x}) \in I(q)) \approx \frac{1}{|\text{Index}(I(q))|} \sum_{f(\boldsymbol{x}_i) \in I(q)} y_i, \tag{18}$$

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

Alternatively, we can use kernel estimation:

$$\mathbb{P}(y = 1 | f(\boldsymbol{x}) \in I(q)) \approx \frac{1}{Z} \sum_{\hat{p} \in I(q)} \mathbb{P}(y = 1 | f(\boldsymbol{x}) = \hat{p}) \cdot \exp\left(-\frac{(p - q)^2}{\sigma^2}\right), \tag{19}$$

Figure 11. Correlation between MSE$_p$ and other metrics on synthetic data. Brier score achieves the most consistent correlation with MSE$_p$.

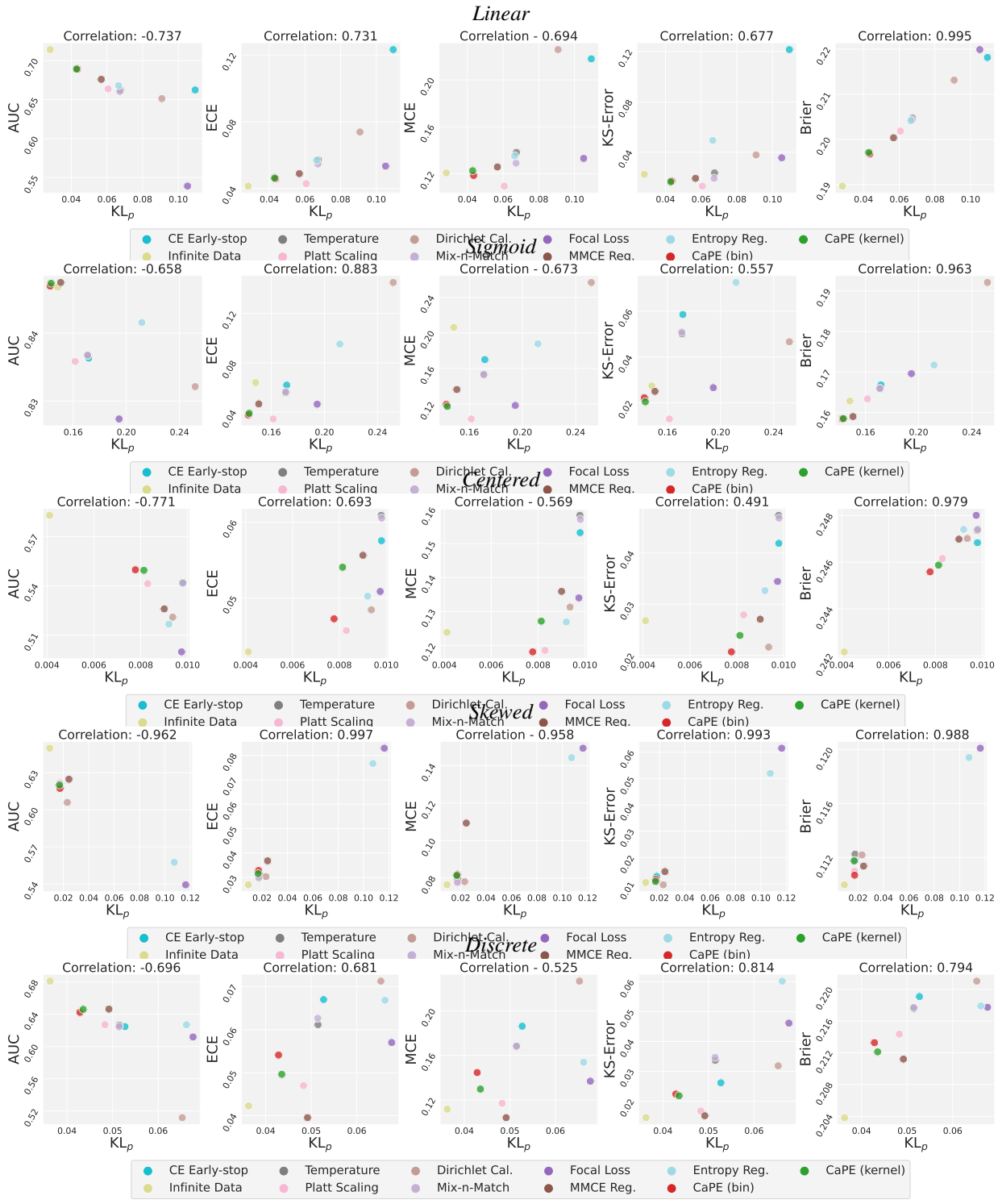

Figure 12. Correlation between KL_p and other metrics on synthetic data. Brier score achieves the most consistent correlation with KL_p.

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

$$\mathbb{P}\left(y=1|f(\boldsymbol{x}) \in I(q)\right) \approx \frac{1}{Z} \sum_{f(\boldsymbol{x}_i) \in I(q)} y_i \exp\left(-\frac{\left(f(\boldsymbol{x}_i)-q\right)^2}{\sigma^2}\right). \tag{20}$$

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 $\mathbb{P}\left(y=1|f(\boldsymbol{x}) \in [Q_{b-1}, Q_b]\right),\ b=1,\ldots,B, Q_0=0$. 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,

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

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 kernel estimation:

$$p^i_{\text{emp}} = \frac{\sum_{k \in \text{NN}(i,r)} K\left(i,k\right) y_k}{\sum_{k \in \text{NN}(i,r)} K\left(i,k\right)}. \tag{21}$$

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

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

with hyperparameter σ.

H Calibration Baselines

This section provides a review of the baseline methods, discussed in Section 7.

Post-processing Postprocessing consists of finding a function $f : [0,1] \to [0,1]$, that transforms the model outputs $\hat{p}_i \to f(\hat{p}_i)$ to improve their calibration.

- In Platt scaling (Platt, 1999) the model predictions are used as inputs to a logistic regression model optimized using a validation set,
$$f_1(\hat{p}_i) = \sigma\left(W^T \hat{p}_i + b\right) \tag{22}$$
where $W \in \mathbb{R}^2, b \in \mathbb{R}$ and σ is the sigmoid function.
- Temperature scaling (Guo et al., 2017) is a single-parameter variant of Platt Scaling where we change a temperature parameter in the logistic function.
- Beta/Dirichlet calibration (Dir-ODIR) (Kull et al., 2017; 2019) assumes that the probabilities can be parametrized by a Beta/Dirichlet distribution i.e.
$$f_j \sim \text{Beta}(\alpha^{(j)}, \beta^{(j)}) \tag{23}$$
Assuming the prior to be $p(y=j) = \pi_j, \pi_j \in [0,1]$, we have $P(y|f_j) \propto \pi_j f_j$, and then $\alpha^{(j)}, \beta^{(j)}$ are estimated by maximizing the posterior.

Ensembling These calibration methods simultaneously train several neural networks, varying parameters in the training process. The final output is a function of all the different outputs.

- 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

$$f_j(\hat{p}_i) = w_1 g_j(\hat{p}_i, T) + w_2 g_j(\hat{p}_i, 0) + w_3 g_j(\hat{p}_i, \infty). \tag{24}$$

  After ensembling the parametric temperature scaling, Mix-n-Match applies non-parametric isotonic regression.
- Deep ensemble (Lakshminarayanan et al., 2017) trains M copies of the neural network with different initializations. The final estimate is the average of all single model outputs

$$p(y_i \mid x_i) = \frac{1}{M} \sum_{j}^{M} p_{\theta_j}(y_i \mid x_i). \tag{25}$$

Modified training These calibration methods train the neural networks from end to end, modifying the training process to improve calibration.

- Confidence penalty (Pereyra et al., 2017) penalizes low entropy output distributions (confidence penalty),

$$\mathcal{L}(\theta) = -\sum_i \log p_\theta(y_i|x_i) - \beta H(p_\theta(y_i|x_i)) \tag{26}$$

- Focal loss (Mukhoti et al., 2020) maximizes entropy while minimizing the KL divergence between the predicted and the target distributions. It also regularizes the weights of the model to avoid overfitting,

$$\mathcal{L}(\theta) = -\sum_i (1 - p_\theta(y_i|x_i))^\beta \log p_\theta(y_i|x_i), \quad \beta \in \mathbb{R}. \tag{27}$$

- Kernel MMCE (Kumar et al., 2018) is a reproducing kernel Hilbert space (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 follows,

$$\text{MMCE}^2(\mathcal{D}) = \sum_{i,j} \frac{(c_i - r_i)(c_j - r_j)k(r_i, r_j)}{m^2} \tag{28}$$

  where $k(r_i, r_j)$ is a kernel function. MMCE is optimized together with the cross entropy loss as a regularization term weighted by a scaling parameter $\lambda \in \mathbb{R}$

$$\mathcal{L}(\theta) = -\sum_i \log p_\theta(y_i|x_i) + \beta \left(\text{MMCE}^2(D)\right)^{\frac{1}{2}}. \tag{29}$$

Hyperparameter tuning for baseline methods Most postprocessing methods do not involve hyperparameters, and are optimized based on a validation set. The modified training methods all use a single hyperparameter to control the strength of regularization. We tune the hyperparameter, using a validation set. Figure 13 shows that MMCE is robust to the choice of hyperparameter. Focal loss and entropy regularization result in inferior performance to that of MMCE for all hyperparameter choices.

I Synthetic data experiments

We use a ResNet-18 backbone architecture for all our experiments with synthetic data.

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. In order to evaluate the performance of the model for probability estimation, the held-out test set contains the ground truth probabilities p_i, in addition to x_i and y_i. Note that we do not use the ground-truth probability labels p_i values during training or inference - we only use them to compare the performance of different models.

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

Table 11: Results of on CIFAR 10 with simulated probabilistic labels. CaPE outperforms a CE early stopped model.

Name	CE early stop	CaPE (bin)	CaPE (kd)
MSE_p	0.0297	0.0252	**0.0247**
KL_p	0.4051	**0.2468**	0.2598

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.15.

J Additional Synthetic-Data Experiments on CIFAR-10

We simulated probabilistic labels for CIFAR-10 to perform additional experiments. Each of the ten classes of CIFAR-10 was assigned a different ground-truth probability. To this end we assigned an integer c between 0 and 9 to each class and set the corresponding probability equal to $c/10$. The training and validation sets were built by assigning each image x_i to a binary label y_i sampled from the corresponding class probability p_i. Note that we do not use the ground-truth probability labels during training or inference - we only use them to evaluate the performance on the test set.

Table 11 shows the results on the test set. We again observe that CaPE outperforms the cross-entropy baseline based on early stopping.

K Additional Details on Real-World Data and Experiments

We present here supplementary information for the real-world datasets used in our experiments.

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)[6] to predict the 5-year survival. The dataset has 1512 whole slide images 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 contain numerous pixels, so we cropped the slides into tiles at 10x 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 are trained with self-supervised momentum contrastive learning (MoCo) (Chen et al., 2020), and the slide-level prediction is obtained from 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[7], 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 indicating whether the mean precipitation increases (1) or not (0).

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

[6] https://www.cancer.gov/tcga
[7] https://opendata.dwd.de/weather/radar/

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-16 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 7.1). 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.

L Analysis of Cancer Survival Results

For cancer survival prediction, we visualize the estimated probabilities on the test set in different pathological stages in Figure 16. In general, patients in earlier stages should have higher probabilities of survival. Deep ensemble produces similar probability estimates for all stages (i.e the model is less discriminative). Cross-entropy minimization (CE) is more discriminative, but has very wide confidence intervals. CaPE is more discriminative than deep ensemble, while having narrower confidence intervals than CE.

M Calibrating from the Beginning

CaPE exploits a calibration-based cost function to improve its probability estimates without overfitting. The empirical probabilities in this loss are computed from the model itself. Consequently, applying this strategy from the beginning of training can be counterproductive, because the model predictions are essentially random. This is demonstrated in the following table, which compares CaPE with a model trained using the calibration loss from the beginning (in the same way as CaPE, alternating with cross-entropy minimization).

Table 12: Comparison between CaPE and a model that uses the calibration loss from the beginning (in the same way as CaPE, alternating with cross-entropy minimization) on synthetic data. All numbers are downscaled by 10^{-2}.

Methods	*Linear*		*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
Bin (start)	2.59	6.81	8.07	22.10	0.48	0.98	0.51	2.37	2.74	6.36
Kernel (start)	2.23	5.68	7.60	21.15	0.54	1.10	0.68	2.84	2.40	5.63
Bin (CaPE)	1.83	4.46	5.29	14.59	**0.38**	**0.78**	0.40	1.72	**1.83**	**4.31**
Kernel (CaPE)	**1.81**	**4.41**	**5.22**	**14.47**	0.40	0.81	**0.39**	**1.70**	1.85	4.36

N Correspondence of Real-World Datasets and the Scenarios of the Simulated Dataset

Figure 17 illustrates the similarity between the empirical probability curves of different real-world datasets and the different scenarios of our synthetic dataset. For the cancer survival dataset, the empirical probabilities are clustered in the center (0.4-0.6) similar to the *Centered* scenario. For the weather forecasting dataset, the probabilities are uniformly distributed across 0.1-0.8 similar to the *Linear* scenario. For the collision prediction dataset, the majority of the data points are clustered in the lower probability region which makes it similar to the *Skewed* scenario.

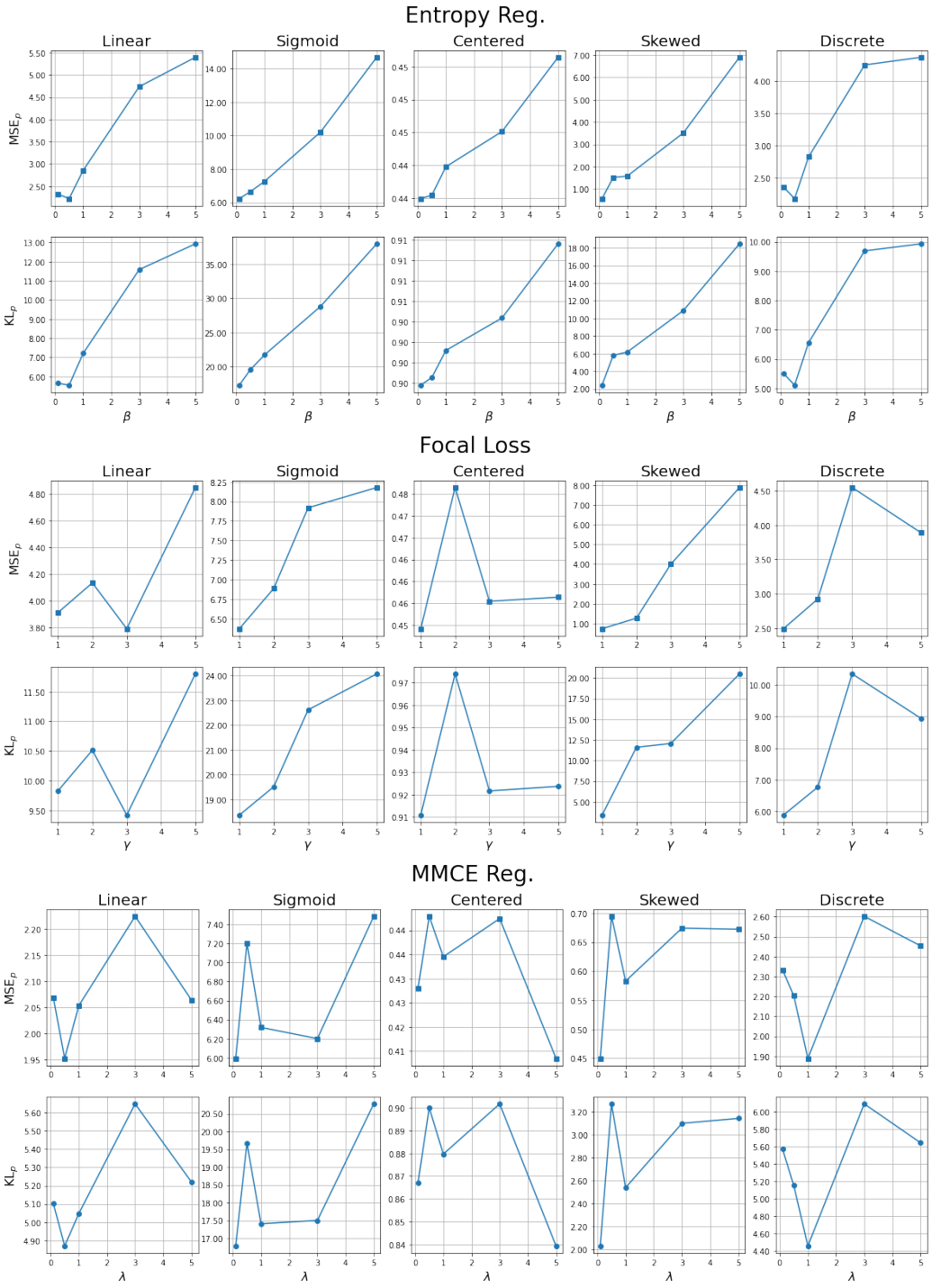

Figure 13. The hyperparameter tuning of baseline methods.

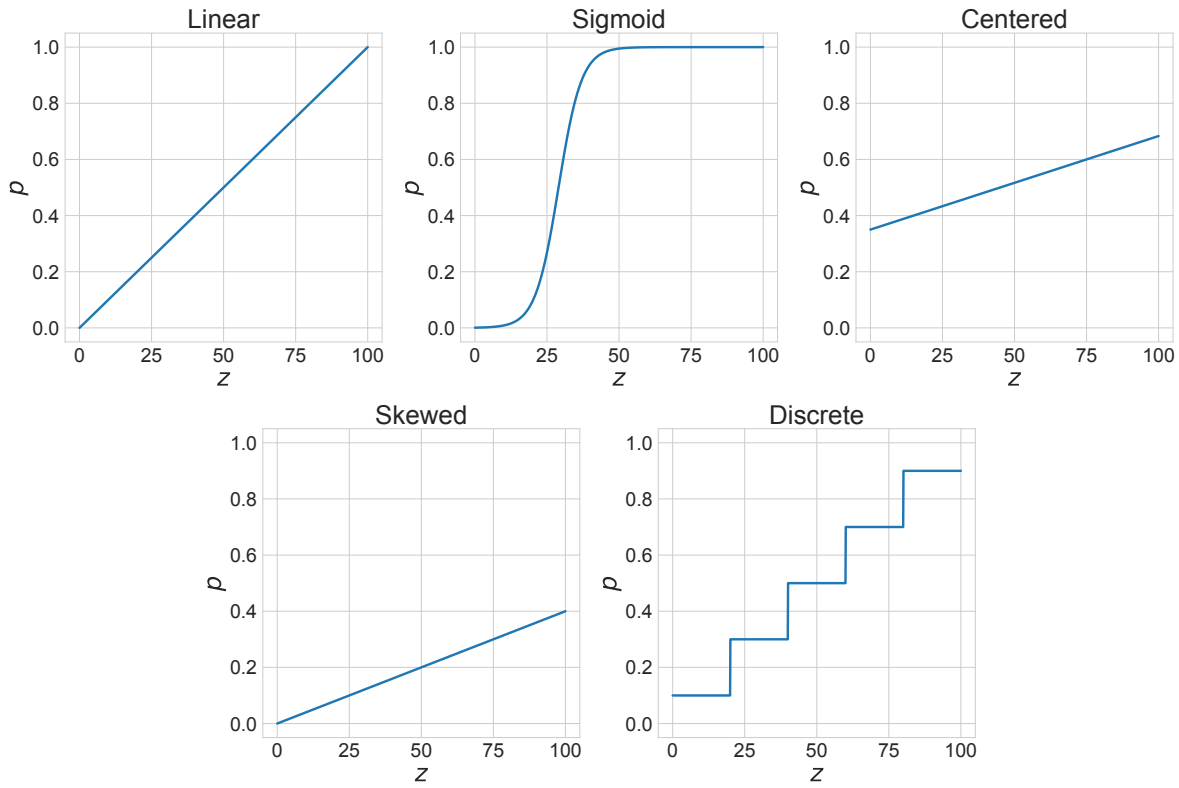

Figure 14. Illustration of the function $\psi(z)$ used to generate the different synthetic-data scenarios.

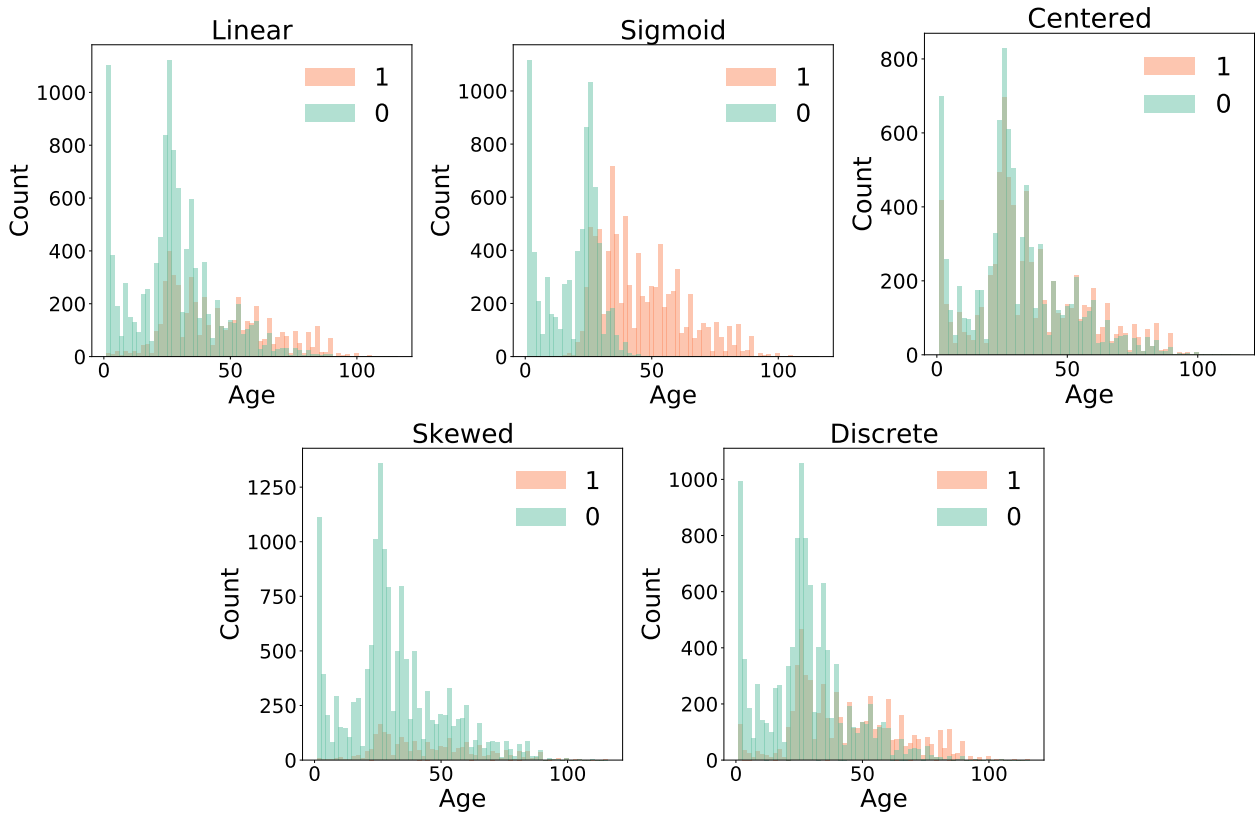

Figure 15. Histograms of the outcomes (y_i) for the different synthetic-data scenarios.

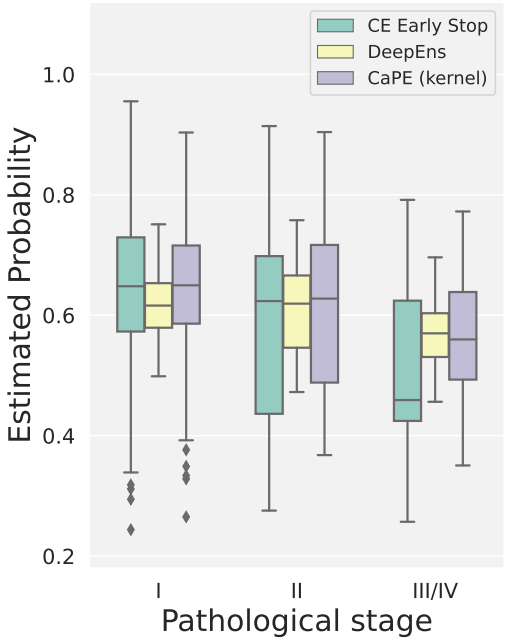

Figure 16. Estimated probability of survival grouped by pathological stages. The plot shows median, samples between 25th to 75th percentile in the box, samples between 0th and 100th percentile on the line, and the outliers as dots. Deep ensemble produces similar probability estimates for patients across all the stages; CE is more discriminative but has a very large variance; CaPE achieves a trade-off between the two baselines.

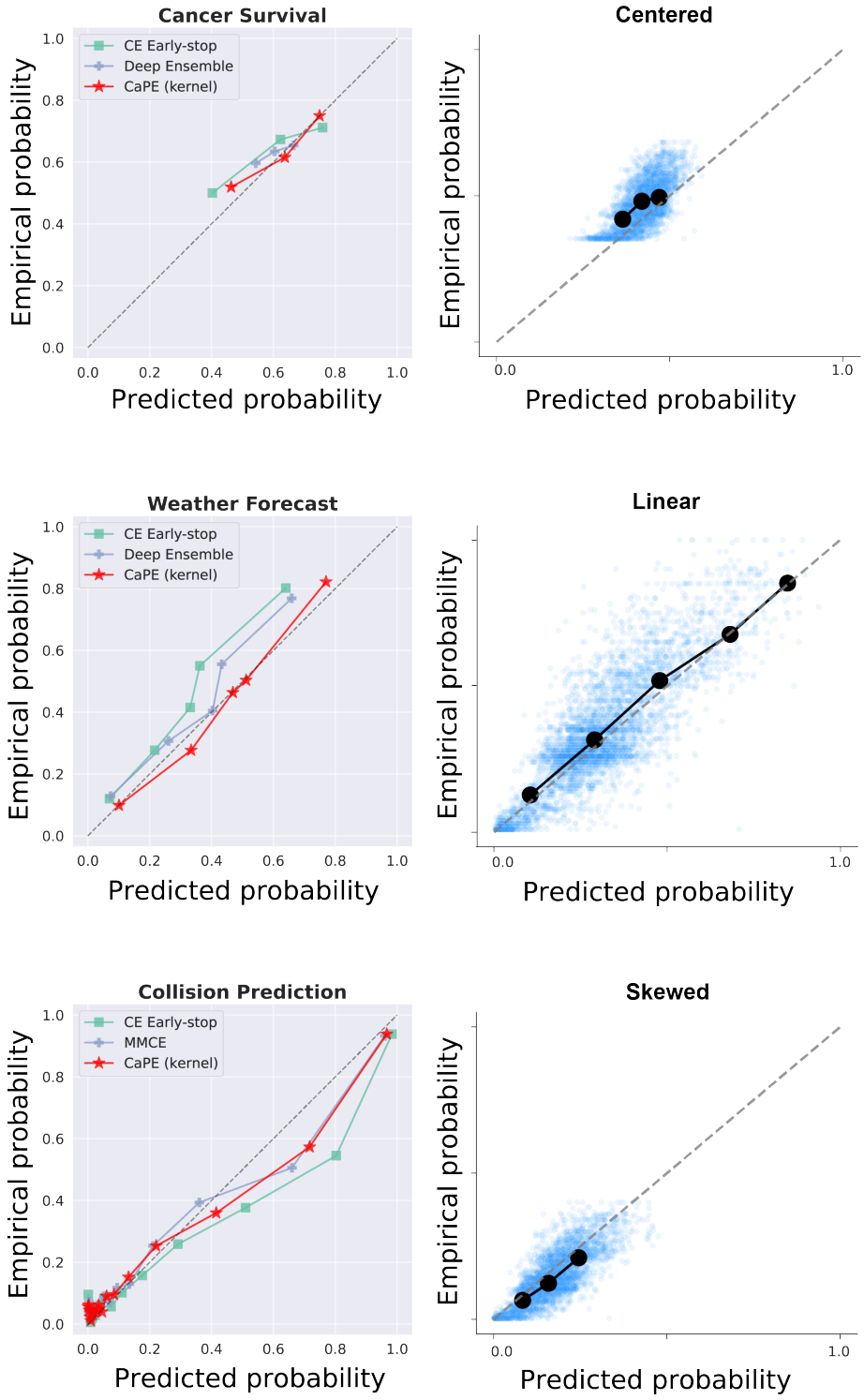

Figure 17. **Comparison of reliability diagrams for real-world data with different scenarios of simulated data**. For the cancer survival dataset, the empirical probabilities are clustered in around (0.4-0.6), similar to the *Centered* scenario. For the weather forecasting dataset, the probabilities are uniformly distributed across 0.1-0.8, similar to the *Linear* scenario. For the collision prediction dataset, the majority of the output probabilities are clustered in the lower probability region, similar to the *Skewed* scenario.