Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2023) 668-683

ECCOMAS
Proceedia

UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12-14 June 2023

ABSTENTION IN REGRESSION

Cristina Garcia-Cardona[1], Jamaludin Mohd-Yusof[1], and Tanmoy Bhattacharya[1]

[1]Los Alamos National Laboratory,
Los Alamos, NM, USA
e-mail: {cgarciac, jamal, tanmoy}@lanl.gov

Abstract. *Identifying noisy samples and outliers at training time is crucial to achieve good performance and improve robustness of deep learning models. However, this process can be difficult and excessively time consuming if attempted by hand, especially for large data sets. Moreover, in realistic situations, there are gradations in our judgement about noisy samples.*

In this work, we extend the Deep Abstention Classifier (DAC) framework to regression problems and devise a deep abstaining regressor (DAR) model. This is a forgiving strategy that limits the detrimental effect of noisy samples on the performance of the predictive model, while, at the same time, retaining information in ambiguous cases.

We apply the new DAR formulation to synthetic and real data sets and demonstrate that the model is able to identify noisy samples, reducing their influence during training, while learning to quantify the uncertainty in the predictions via a heteroscedastic criterion. DAR constitutes a new tool for the machine learning practitioner, producing actionable uncertainty quantification predictions and helping with the automatic curation of data. This may be an efficient way to reduce the burden of manual labelling for evaluation by a human-in-the-loop.

Keywords: Uncertainty Quantification, Deep Abstention Models, Heteroscedastic Loss.

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2023.
doi: 10.7712/120223.10366.19843

1 INTRODUCTION

The deep abstaining classifier (DAC) introduced in [1] is a regular deep neural network model trained for classification but with an additional class, the abstention class, that allows for identifying noisy sample patterns. The DAC is trained with a custom loss function that combines a traditional loss function for the non-abstained samples with an additional term to penalize excess abstention. In this way, the DAC is able to train a robust model, that abstains on the noisy samples, while being able to achieve higher accuracy on the retained samples.

In this work we extend the DAC model to regression problems and show that we can obtain an analogous performance of identifying samples that are noisy in the training set, downgrading their influence during training and synthesizing a model of the data noise that is useful for inferring when the model is confident about its output or abstaining from making a prediction when it is not.

Moreover, to build an actionable deep learning model for regression, we would want to obtain a measure of the quality of the predictions made. Therefore, we also provide a measure of uncertainty in the predictions by adding a second training stage where we use a heteroscedastic loss applied only over the non-abstained samples.

2 PREVIOUS WORK

We previously [1] developed the deep abstaining classifier (DAC) model which introduced the notion of adding a separate unlabeled 'abstention' class to the original 'real' classes. The DAC allows the option of abstaining on those samples when the prediction is low-confidence, rather than forcing the classifier to assign it to one of the real classes. We showed that this can allow the network to learn the characteristics which may make the prediction unreliable, which in turn allows it to make more accurate predictions on the non-abstained set. Examples of note include the ability to learn that a class is unreliable due to mislabeling, as well as detecting unlabeled markers of mislabeling.

For a given input x, denote y to be the predicted class output by the DNN. We define $p_i = p_w(y = i|x)$ (the probability of the ith class given x) as the i^{th} output of the DNN that implements the probability model $p_w(y = i|x)$ (using a softmax function as its final layer) with w being the set of weight matrices of the DNN. For notational brevity, we use p_i in place of $p_w(y = i|x)$ when the input context x is clear.

The standard cross-entropy training loss for DNNs then takes the form

$$\mathcal{L}_{\text{standard}} = -\sum_{i=1}^{k} t_i \log p_i \tag{1}$$

where $t_i \in \{0, 1\}$ is the target for the current sample. The DAC has an additional $k + 1^{\text{st}}$ output p_{k+1} which is meant to indicate the probability of abstention. We train the DAC with the following modified version of the k-class cross-entropy per-sample loss:

$$\mathcal{L}(x) = (1 - p_{k+1}) \left(-\sum_{i=1}^{k} t_i \log \frac{p_i}{1 - p_{k+1}} \right) + \alpha \log \frac{1}{1 - p_{k+1}} \tag{2}$$

where k is the number of classes excluding the abstention class, t_i is the true label of training data for class i: it is one when i is the true label, zero otherwise, $k + 1$ is the abstention class, p_{k+1} is the probability of the abstention class and α adjusts the penalty term for abstention.

This loss function behaves like a regular cross-entropy loss on the original classes and adds an additional loss term, scaled by a tuning parameter α that controls the propensity for abstention. This parameter is tuned during training to guarantee an upper bound on the abstention rate while optimizing the accuracy. A very high value of α means a high penalty for abstaining, driving the model towards no abstention. Conversely, a very low value of α may drive the model to abstain on everything.

The target abstention level is a function of the performance of the base classifier on the current dataset. For example, if the base classifier can obtain *e.g.,* 70% accuracy on a given dataset, then the naive expectation might be that the DAC could then achieve perfect accuracy while abstaining on 30% of the samples, but this tends to be practically impossible because of the pattern-free noise in real world data.

3 ABSTENTION FOR REGRESSION

Given a known matched collection of observed data $\mathcal{D} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{N}$ of N samples, with $\mathbf{x_i} \in \mathbb{R}^d$ representing the i-th input pattern and $\mathbf{y_i} \in \mathbb{R}^{d_{out}}$ the corresponding i-th expected output tuple, a neural network regression model $\mathbf{f}^{\boldsymbol{\omega}}$ can be constructed to synthesize the input-output relation. The optimal set of parameters $\boldsymbol{\omega}^\star$ is estimated by minimizing a loss function measuring the difference between the expected output $\mathbf{y_i}$ and the output computed by the model $\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x_i})$. In the regression case, where the outputs are continuous values, a commonly used loss function is the mean squared error (MSE),

$$\mathrm{MSE}\left(\mathbf{f}^{\boldsymbol{\omega}}; \mathcal{D}\right) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{y}_i - \mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i) \right\|^2 , \tag{3}$$

The model for abstention in regression follows an analogous approach to DAC. We construct the regression model to include an additional output, the *abstention output*, denoted as $\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)_a$ which uses a sigmoid activation function to limit the output range to $[0, 1]$, such that we can use it to indicate when the model abstains from making a prediction. We use a threshold to distinguish non-abstained from abstained predictions. Specifically,

$$\hat{y}_{i,a} = \sigma(\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)_a - 0.5) , \tag{4}$$

where σ represents the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$. Hence, when $\hat{y}_{i,a} > 0.5$ we regard it as an indication that the model is abstaining from making a prediction for the i-th sample point.

The abstention-regression loss function follows the original pattern of implementing a convex combination of the base cost and the abstention cost, but, in this case, the base cost is measured via a MSE term. The abstention-MSE loss corresponds to

$$\mathrm{MSE}_{\mathrm{abs}}\left(\mathbf{f}^{\boldsymbol{\omega}}; \mathcal{D}\right) = \frac{1}{N} \sum_{i=1}^{N} [(1 - \hat{y}_{i,a}) \|\mathbf{y}_i - \mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)\|_2^2 - \alpha \log(1 - \hat{y}_{i,a})] , \tag{5}$$

for $\alpha > 0$.

3.1 Alpha Adaptation

An important component of the abstention-MSE loss (5) is the parameter $\alpha > 0$ which establishes the trade-off between minimizing the MSE loss over non-abstained samples and the

fraction of abstained samples. The larger the value of α, the more costly is to have a large fraction of abstention, so this helps to limit the abstention to the samples that have a larger impact on the MSE, in other words, it enhances abstention over the more noisy samples. Conversely, the smaller α is, the cheaper it is to abstain, and thus, the larger the resulting fraction of abstention. This also means that less samples are effectively used to learn the mapping $\mathbf{f}^{\boldsymbol{\omega}}$.

To dynamically capture an appropriate trade-off between learning and abstention, the α parameter is tuned during the training stage. The tuning is based on two complementary goals: (i) to not exceed a target maximum abstention fraction and (ii) to not exceed a target maximum loss. The target maximum abstention fraction should be set to the expected noise in the data, or, if not known, some target abstention fraction based on desired performance. Since the abstention fraction is in the range $[0, 1]$, there is a specific scale to base the setting on. In contrast, the scale of the loss function is arbitrary. Therefore, for setting a target maximum loss and for computing the current loss (over non-abstained samples) we use a normalized MSE, i.e. we divide the MSE by the square of the maximum ground truth value. In training, this is done by batch, so the maximum of the ground truth value per batch is used.

Algorithm 1 Algorithm for Alpha Adaptation

Require: $f_{\text{max-abs}} \in (0, 1)$, $l_{\text{max-abs}} \geq 0$ and $\alpha^{(k)} > 0$
Ensure: $\alpha^{(k+1)} > 0$
  $\mathbf{y}_{\max} \leftarrow \max\{\mathbf{y}_j\}$ for $j \in [1, J]$ i.e. current set of non-abstained samples
  $l \leftarrow \frac{1}{\mathbf{y}_{\max}^2} \frac{1}{J} \sum_j \|\mathbf{y}_j - \mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_j)\|_2^2$ ▷ This is normalized MSE
  $e_l \leftarrow \max\{l - l_{\text{max-abs}}, 0\}$
  $e_f \leftarrow \max\{f - f_{\text{max-abs}}, 0\}$
  $a \leftarrow 1 - e_l + e_f$
  if $a < 0$ **then**
    $a \leftarrow 0.99$
  end if
  $\alpha^{(k+1)} \leftarrow a\, \alpha^{(k)}$

Algorithm 1 describes the process for adapting α, a process that is executed after each epoch. $f_{\text{max-abs}}$ denotes the target maximum abstention fraction while $l_{\text{max-abs}}$ denotes the target maximum normalized non-abstained MSE loss. These values are set at the beginning to reflect performance goals and are kept fixed during training. Note that the updating of a, the α adaptive multiplier, depends on e_l, the difference between target and achieved maximum normalized non-abstained MSE loss, and on e_f, the difference between target and achieved abstention fraction. A positive e_l means that the loss is greater than the target value set. This nudges a to be less than one, which in turn means a smaller α, making it cheaper to abstain. This increases the abstention fraction and potentially decreases the loss by removing noisy samples from having any effect on the first term of eq. (5). On the other hand, a positive e_f means that the abstention fraction is greater than the target set. This pushes a to be greater than one, which increments α, making it more expensive to abstain. This should decrease the abstention fraction, albeit at the price of a possible increment in the loss.

Starting from an initial value of alpha $\alpha^{(0)} > 0$, this adaptive process should achieve a stable α value asymptotically. To avoid huge α swings while training, the factor a can be confined to vary in a specified range (e.g. $a \in [0.8, 1.25]$). The adaptation can be made more sensitive to one or the other target setting by using different multiplier factors (i.e. *gains*) for e_l and e_f

when updating a. Note that we use a validation set during training and base the α adaptation on the metrics computed over this set (not the training set).

3.2 Heteroscedastic Loss

While the abstaining regressor provides a framework to learn regions where the model is not confident about its predictions, it does not provide a metric of uncertainty over the predictions made. Therefore, we combine the abstention formulation with the heteroscedastic model [2] to account for both: outliers, that would adversely affect the learning, and learning the noise level that can reasonably be expected given the training data. The heteroscedastic model assumes a normally distributed noise, which can be posed as the problem of learning the mean and the variance as smooth functions of the input features. Thus, the model is trained to minimize the negative log-likelihood (NLL) of a feature-dependent normal distribution, which can be expressed as,

$$\mathcal{L}_{\text{het}}\left(f^{\boldsymbol{\omega}}, \nu^{\boldsymbol{\omega}}; \mathcal{D}\right) = \frac{1}{J}\sum_{j=1}^{J} \frac{1}{2\nu(\mathbf{x}_j)^2}\left\|y_j - f(\mathbf{x}_j)\right\|^2 + \frac{1}{2}\log \nu(\mathbf{x}_j)^2\,, \tag{6}$$

with $f(\mathbf{x})$ and $\nu(\mathbf{x})^2$ the mean and variance predictions of the model, respectively.

To make use of this heteroscedastic loss, we build a model with sufficient outputs to predict mean $f(\mathbf{x})$, variance $\nu(\mathbf{x})^2$ and abstention $\hat{y}_{i,a}$ as defined in eq. (4), and train the model in stages as described next.

3.2.1 Optimization of Abstention and Heteroscedastic Loss

A straightforward combination of the abstention concept with the heteroscedastic model produces the following loss function,

$$\begin{aligned}\mathcal{L}_{\text{abs-het}}\left(f^{\boldsymbol{\omega}}, \nu^{\boldsymbol{\omega}}; \mathcal{D}\right) \;=\; & \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2\nu(\mathbf{x}_i)^2}(1-\hat{y}_{i,a})\left\|y_i - f(\mathbf{x}_i)\right\|^2\right. \\ & \left. + \frac{1}{2}(1-\hat{y}_{i,a})\log \nu(\mathbf{x}_i)^2 - \alpha\log(1-\hat{y}_{i,a})\right]\,. \end{aligned} \tag{7}$$

We find, however, that learning to infer abstention and simultaneously learning to infer the uncertainty yields an unstable or abstention-free result. These two objectives compete against each other, since abstention can be substituted by a very large point-wise variance in expression (7) (see Sec. 4.1.3 for an illustration). Rather than devise stabilization mechanisms that could complicate α adaptation, we choose to split the training in two distinct stages instead.

3.2.2 Abstention and Heteroscedastic Loss in Two Stages

We train our model in two stages. In the first stage, we train an abstaining regressor, driven by MSE_{abs} (5), to learn to infer abstention. In the second stage, we use only the non-abstained samples in the training set, and driven by $\mathcal{L}_{\text{het}}$ (6), we learn the heteroscedastic model, but limit the updates only to the last layer of the model. We restrict the re-training to the last layer since we still want to preserve the abstention ability of the model and be able to apply it when inferring over new samples.

While the abstention stage extends seamlessly to regression problems with multiple outputs, the heteroscedastic estimation does not. For simplicity, for multiple regression outputs, we consider the mean and variance for each output independently, i.e. we use a univariate normal NLL and discard covariance terms (or equivalently, learn a diagonal covariance matrix in a multivariate setup).

4 NUMERICAL EXPERIMENTS

We demonstrate the performance of the abstention in regression model in synthetic data as well as in publicly available data. The following subsections describe the experiments performed and the obtained results.

4.1 Synthetic Data

To fully evaluate the model we construct a synthetic data set where we have complete information about the ground truth, the noise level and the outliers.

4.1.1 Data Generation

We use the Hill model,

$$H(x; r_1, r_2, k, h) = \begin{cases} r_1 + (r_2 - r_1)\frac{k^h}{k^h + x^h} & \text{for } h \geq 0\,, \\ \\ r_1 + (r_2 - r_1)\frac{k^{-h}}{k^{-h} + x^{-h}} & \text{for } h < 0\,, \end{cases} \tag{8}$$

with parameters $r_1, r_2, k > 0$ and h to build synthetic data sets in order to evaluate the models against the known ground truth. Specifically, a Hill model with parameters $r_1 = 10$, $r_2 = 30$, $k = 10$ and $h = -6$ is used as the base function, $y = H(x)$ with $x \in [1, 20]$. This corresponds to a monotonically decreasing 1D signal.

In order to simulate noisy input and outliers, we add a 1D Gaussian noise to the output. The Gaussian noise has mean zero and standard deviation given by another Hill model with parameters $r_{1n} = 5.48$, $r_{2n} = 3.16$, $k = 10$ and $h = -6$. In this way we simulate a heteroscedastic, i.e., feature-dependent, noise model, with noise increasing for higher x-coordinate values. The outliers come from generating an additional correlated input feature equal to the product of the base Hill function times a random normal variable with standard deviation of 0.1. For the samples where this second feature has an absolute value greater than 2.0, we further corrupt the output by replacing it with a random normal variable with zero mean and a standard deviation of 30.0. In equations, $x_1 \in [1, 20]$ and

$$\begin{aligned} x_2 &= H(x_1; r_1, r_2, k, h)\, \mathcal{N}(x_1; 0, 0.1)\,, \\ y &= \begin{cases} H(x_1; r_1, r_2, k, h) + \mathcal{N}\left(x_1; 0, H(x_1; r_{1n}, r_{2n}, k, h)\right) & \text{for } |x_2| \leq 2\,, \\ \\ \mathcal{N}(x_1; 0, 30.0) & \text{for } |x_2| > 2\,. \end{cases} \end{aligned}$$

Since the second feature x_2 tends to have larger values for smaller x_1-coordinate values, the outliers also tend to be located at lower x_1-coordinate values. Figure 1 displays one such realization, with the *left* plot illustrating the (x_1, x_2) feature correlations, while the *right* plot shows the corrupted output, for simplicity, in the plane y vs. x_1. Note that with this random generation process the MSE of the noisy data with respect to the clean data is of the order of 500.

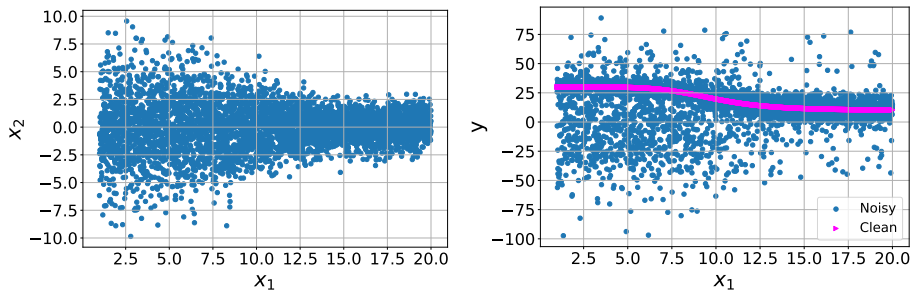


Figure 1: Synthetic data generation. Left: Feature correlation, outliers correspond to $|x_2| > 2$. Right: Output y with respect to feature x_1; 'Clean' corresponds to $H(x_1; r_1, r_2, k, h)$. For this realization, outliers are 28.7% of data and the MSE vs. clean is 446.43.

4.1.2 Model Architecture and Training

For the synthetic data, we train a 3-layer model with dense connections, 200 units and elu activation functions in the first two layers and 3 units in the last layer corresponding to $f(\mathbf{x})$, $\nu(\mathbf{x})$ and $f(\mathbf{x})_a$. The abstention output uses a sigmoid activation function while the other two outputs use a linear activation function. We use a stochastic gradient descent (SGD) optimizer [3] with learning rate of 10^{-5} and Nesterov acceleration with 0.9 momentum and 10^{-6} decay, for 1000 epochs with batch size of 12 in the abstention stage and 500 epochs and learning rate of 10^{-6} in the heteroscedastic stage (Nesterov acceleration and other parameters are held the same as in the abstention stage). We generate 4500 points and do a partition of 56% for training, 19% for validation and 25% for testing. We implement and train all our models using the Keras [4] interface of Tensorflow [5] v2.8.1.

4.1.3 Results

As described in Algorithm 1, there are three parameters that control the alpha adaptation and we configure them as follows. We set the target maximum abstention fraction in the order of the known noise level $f_{\text{max-abs}} = 0.33$. We set the target maximum normalized non-abstained MSE loss to a relatively small value to learn a tight regression function $l_{\text{max-abs}} = 0.045$. We set the initial α to a low value that should encourage abstention $\alpha^{(0)} = 0.1$.

Optimization of Abstention and Heteroscedastic Loss Figs. 2 and 3 show the results obtained when optimizing loss (7) via only one-stage. Fig 2 displays the evolution of different loss and metrics during training both in the training set (in black) and a validation set (in cyan)[1]. The top-left plot corresponds to the optimized $\mathcal{L}_{\text{abs-het}}$ loss, eq. (7), with an α value that is being adapted to achieve the targets set for $f_{\text{max-abs}}$ and $l_{\text{max-abs}}$. The top-right plot displays the evolution of the normalized non-abstained MSE loss l in Algorithm 1. The bottom-left shows the evolution of the abstention fraction while the bottom-right, the evolution of $\alpha^{(k)}$ (in green). These plots illustrate that stabilizing the minimization when both abstention and heteroscesdastic loss are active is not a trivial task. In this case, α keeps oscillating, and even when that oscillation is small it translates into large oscillations in the normalized MSE and in the abstention fraction.

[1] We use the same settings for the optimizer as in the abstention stage described before for the two-stage process.

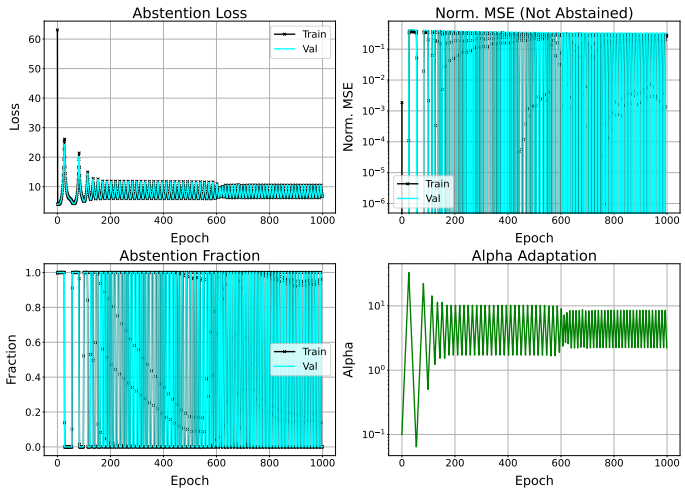


Figure 2: Training evolution for (7). Top-Left: Evolution of (7) with adaptive α. Top-Right: Evolution of l, normalized MSE in Algorithm 1. Bot.-Left: Evolution of abstention fraction. Bot.-Right: Evolution of α.

The evaluation of the model obtained after 1000 training epochs is shown in Fig. 3. Blue points correspond to ground-truth, yellow crosses correspond to the learned function and the gray-shaded region corresponds to a radius of two predicted standard deviations centered on the predicted mean. The abstention fraction is zero (no points marked red) and a large standard deviation is predicted to account for the noise present in the dataset.

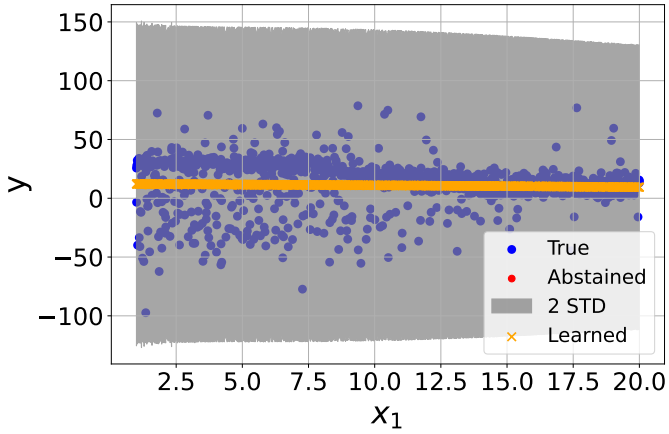


Figure 3: Evaluation on the testing set of the model trained using the abstention-regression loss of (7).

Abstention and Heteroscedastic Loss in Two Stages Fig. 4 displays the evolution of different metrics when optimizing loss (5) evaluated in the training set (in black) and the validation set (in cyan). The top-left plot corresponds to the optimized $\mathrm{MSE}_{\mathrm{abs}}$ loss (5), with an α value that is being adapted to achieve the targets set for $f_{\text{max-abs}}$ and $l_{\text{max-abs}}$. The other plots are as before, top-right displays the evolution of the normalized non-abstained MSE loss l in Algorithm 1, bottom-left shows the evolution of the abstention fraction and bottom-right, the

evolution of $\alpha^{(k)}$ (in green). It can be observed that at the beginning of training, the value of α is small, which makes it cheap to abstain, up to the point that the abstention is 100%, which also explains why l becomes zero. However, in order to satisfy the target abstention, α increases, penalizing the second term in eq. (5) more strongly, making it more costly to abstain. α stabilizes to a value that allows to satisfy both targets: $f_{\text{max-abs}} = 0.33$ and $l_{\text{max-abs}} = 0.045$.

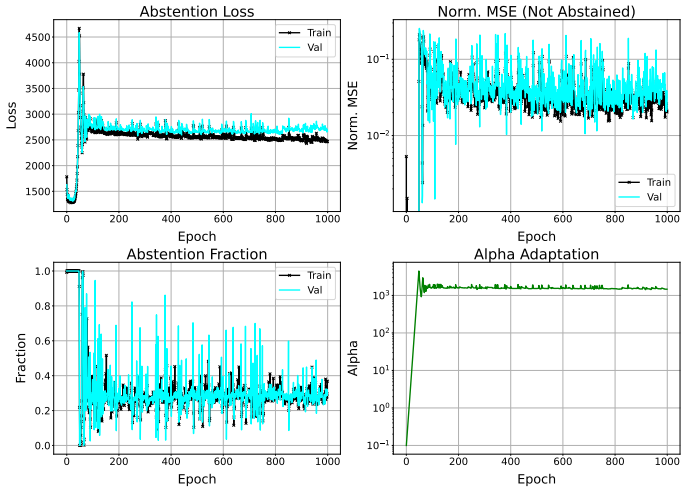


Figure 4: Training evolution. Top-Left: Evolution of eq. (5) with adaptive α. Top-Right: Evolution of l, normalized MSE in Algorithm 1. Bot.-Left: Evolution of abstention fraction. Bot.-Right: Evolution of α.

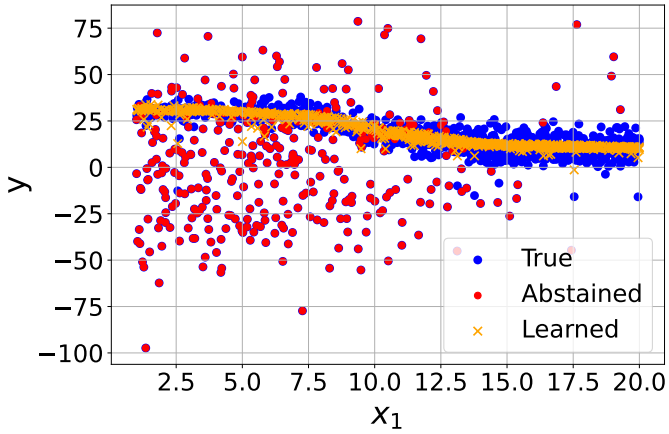


Figure 5: Abstention result for regression: MSE. Abstention in test set: 30.4%. Data contamination: 28.7%.

Fig. 5 shows the results of the model when applied to testing data. Blue points correspond to ground-truth. Red points correspond to abstained samples, i.e. ground truth points where the model is not confident and abstains of making a prediction. Yellow crosses correspond to the learned function. The model abstains on 31.5% of the testing samples, showing not only a good quantitative match with the expected level of noise, but also being able to correctly identify and abstain from making predictions over the more noisy samples in the testing set.

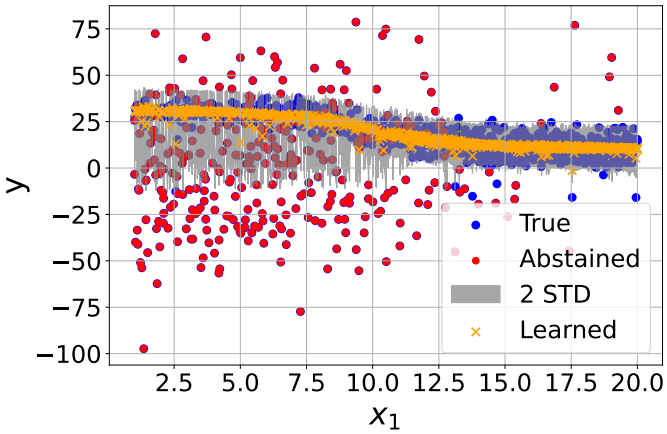


Figure 6: Evaluation on the testing set of the model trained using an abstention stage followed by a heteroscedastic stage.

Fig. 6 shows the heteroscedastic regression abstention model. Once the abstained samples are identified, a second training stage is used to learn a heteroscesdastic model from the non-abstained samples. The heteroscedastic model is trained to predict mean and standard deviation of a univariate normally distributed noise model, with the standard deviation useful as a measure of the uncertainty in the predictions. The plot shows results of the heteroscedastic abstention regression model when applied to testing data as a gray-shaded region that corresponds to a radius of two standard deviations centered on the mean prediction. The shaded region should account for 95.45% of the predictions. As can be seen, the shaded region does effectively cover most of the blue distribution while discarding many of the red abstained samples from the estimation of uncertainty.

Sweeping Abstention Fraction and MSE Loss Targets The performance for different combinations of target maximum abstention fraction $f_{\text{max-abs}}$ and target maximum normalized non-abstained MSE loss $l_{\text{max-abs}}$ is described next. We use a grid with $f_{\text{max-abs}}$ taking values in $[0.25, 0.29, 0.33, 0.37, 0.41, 0.45]$ and $l_{\text{max-abs}}$ in [0.03, 0.05, 0.07, 0.1, 0.12]. We use the same architecture described before and repeat 10 times the training for each combination of target maximum abstention fraction and target maximum normalized loss. We generate 3000 points and for each repetition we perform a different partition of about 64% samples for training, 11% for validation and 25% for testing. We again use a SGD with learning rate of 10^{-4} and Nesterov acceleration with 0.9 momentum and 10^{-6} decay, for 200 epochs with batch size of 12 in the abstention stage and 200 epochs and learning rate of 10^{-6} in the heteroscedastic stage (Nesterov acceleration and other parameters same as in the abstention stage).

Figure 7 (*left*) displays the abstention fraction obtained over the test set when sweeping the different combinations of abstention fraction and loss target, while (*right*) plots the MSE obtained over the test set. These figures show that the abstention fractions obtained are close to the ones set and that the MSE obtained is smaller when more abstention fraction is allowed. Figure 8 displays MSE and R^2 obtained at the first stage without and with abstention. It demonstrates that abstention improves all metrics producing lower MSE and higher R^2. Figure 9 shows MSE, R^2 and NLL obtained over the test set when training with abstention on the first stage and followed by a heteroscedastic stage. Training an heteroscedastic stage, over non abstained samples and updating only the last layer, does not change the abstention predictions, as demon-

strated by the minimal changes in MSE and R^2, while producing a mechanism to estimate NLL.

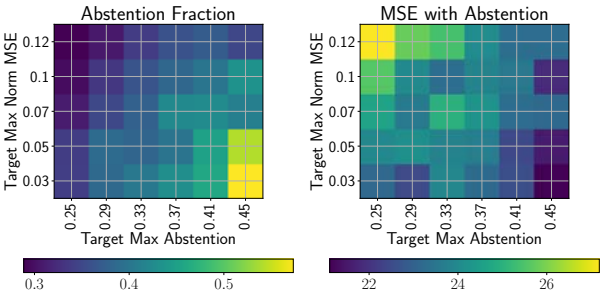


Figure 7: Comparison of targets set vs. obtained over test set.

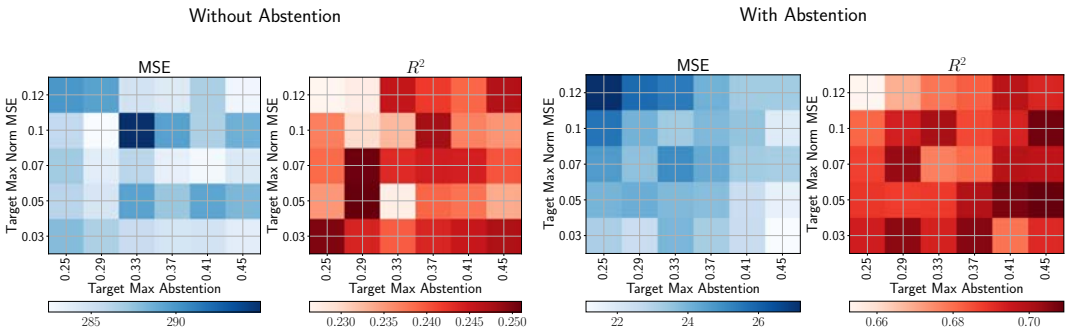


Figure 8: Summary MSE and R^2 evaluated at the end of first stage over the test set. Left: without abstention. Right: with abstention. In blue plots, smaller is better; in red plots, larger is better.

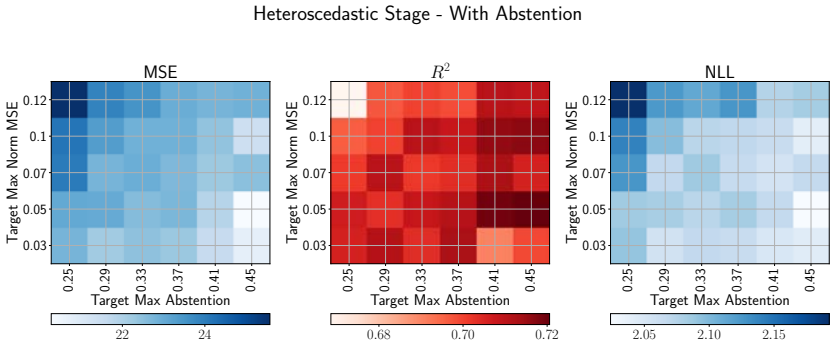


Figure 9: Summary MSE, R^2 and NLL with abstention after heteroscedastic stage evaluated over the test set. In blue plots, smaller is better; in red plots, larger is better.

4.2 Public Data

To provide a broader evaluation, we apply our abstention regression model to different public UCI regression datasets, using the same experimental setup as demonstrated by [6, 7]. Briefly,

a 90%-10% training-testing partition is used to evaluate model performance. The models are constructed with one hidden layer of 50 units followed by ReLU activation. For the mid-to-large sized 'Protein' and 'Year' datasets, the model uses 100 units in the hidden layer. Models are trained during 40 epochs with batch size of 1 (except the 'Year' dataset, see below). Metrics, i.e. root MSE (RMSE) and negative log-likelihood (NLL), are calculated using 20 repetitions for smaller datasets, using 5 repetitions for the mid-sized 'Protein' dataset, and 2 repetitions for the large 'Year' dataset. The mean $\pm$ the standard deviation are reported.

We run these evaluations optimizing: (i) the MSE loss of eq. (3) and (ii) the heteroscedastic loss of eq. (6). Tables 1 and 2 describe the datasets, including number of samples N, the dimensionality d and the number of samples in the testing partition N_{test}, and display the results obtained. These are labeled MSE and HET (for the heteroscedastic loss). Notice that we do not optimize the model hyperparemeters and do not apply sophisticated UQ models, however, the values obtained are close to the ones reported, serving the purpose of providing a familiar baseline.

				RMSE			
				MSE	**HET**	**ABS**	
Dataset	N	d	N_{test}			**Best**	**Worst**
Concrete	1,030	8	103	6.54±0.66	12.63±1.43	5.96±0.44	7.30±4.05
Energy	768	8	77	1.04±0.14	3.15±0.82	0.85±0.17	1.91±0.34
Naval	11,934	16	1,194	0.008±0.001	0.003±0.001	0.011±0.004	0.013±0.007
Power	9,568	4	957	4.33±0.29	4.46±0.32	4.39±0.15	4.44±0.15
Protein	45,730	9	11,433	4.39±0.08	4.76±0.03	2.44±0.05	2.97±0.30
Wine	1,599	11	160	0.63±0.03	0.66±0.04	0.64±0.04	0.64±0.04
Yacht	308	6	31	1.16±0.51	3.99±1.70	0.54±0.16	0.69±0.34
Year	515,345	90	51,535	9.34±0.01	9.50±0.17	9.35±0.14	9.61±0.09

Table 1: RMSE for UCI regression datasets.

				NLL		
				HET	**ABS**	
Dataset	N	d	N_{test}		**Best**	**Worst**
Concrete	1,030	8	103	2.88±0.12	2.32±0.15	2.62±0.46
Energy	768	8	77	1.04±0.25	0.33±0.31	0.86±0.21
Naval	11,934	16	1,194	-5.02±0.15	-3.06±0.20	-2.96±0.23
Power	9,568	4	957	1.99±0.08	1.99±0.06	2.03±0.10
Protein	45,730	9	11,433	1.96±0.02	1.35±0.07	1.51±0.07
Wine	1,599	11	160	0.11±0.15	0.12±0.08	0.24±0.34
Yacht	308	6	31	-0.02±0.37	0.11±0.16	0.27±0.44
Year	515,345	90	51,535	2.73±0.01	3.56±0.69	7.98±4.91

Table 2: NLL for UCI regression datasets.

For the abstention case, we use the same model architecture, sets partitions and training setup, but we sweep different combinations of target maximum abstention fraction $f_{\text{max-abs}}$ and target maximum normalized non-abstained MSE loss $l_{\text{max-abs}}$. We use a grid with $f_{\text{max-abs}}$ taking values in $[0.3, 0.6]$ (for all the datasets) and $l_{\text{max-abs}}$ as described in Table 3 for each dataset. We initialize $\alpha^{(0)} = 1$ for 'Concrete', 'Energy', 'Naval, 'Wine' and 'Yacht'; $\alpha^{(0)} = 1 \times 10^8$ for 'Power'; $\alpha^{(0)} = 11$ for 'Protein' and $\alpha^{(0)} = 1.1 \times 10^6$ for 'Year'.

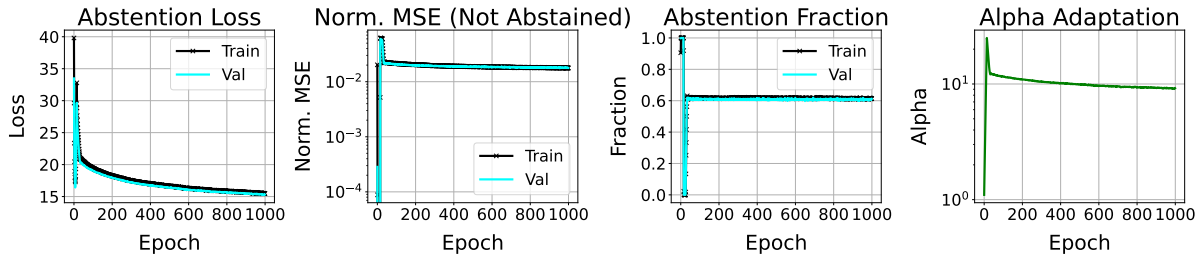
Training the 'Year' Dataset: For the 'Year' dataset, we initialize $\alpha^{(0)} = 1.1 \times 10^6$ to prevent abstention in all the training samples. Although the asymptotic convergence of the abstention model does not depend on α initialization [1], it leads to different results for limited iteration windows. Under the experimental setup used in previous works, consisting on training during 40 epochs with batch size of 1, it is not possible to reach the range of effective α adaptation. Therefore, in this case we run for 100 epochs with batch size of 100. For a fairer comparison, we use this same setup for training the corresponding MSE and heteroscedastic models.

		Abstention Fraction			
Dataset	$f_{\text{max-abs}}$	$l_{\text{max-abs}}$			
		1.00×10^{-6}	4.64×10^{-6}	2.15×10^{-5}	1.00×10^{-4}
Concrete	0.3	0.30±0.45	0.20±0.40	0.35±0.47	0.30±0.45
	0.6	0.40±0.48	0.40±0.48	0.40±0.48	0.45±0.49
		1.00×10^{-4}	4.64×10^{-4}	2.15×10^{-3}	1.00×10^{-2}
Energy	0.3	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
	0.6	0.44±0.15	0.41±0.18	0.49±0.05	0.47±0.12
		1×10^{-6}	4.64×10^{-6}	2.15×10^{-5}	1.00×10^{-4}
Naval	0.3	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
	0.6	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		5.01×10^{-5}	6.30×10^{-5}	7.94×10^{-5}	1.00×10^{-4}
Power	0.3	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
	0.6	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		1.00×10^{-4}	1.00×10^{-3}	1.00×10^{-2}	
Protein	0.3	0.70±0.04	0.69±0.03	0.59±0.29	
	0.6	0.78±0.01	0.62±0.31	0.62±0.31	
		1.00×10^{-6}	4.64×10^{-6}	2.15×10^{-5}	1.00×10^{-4}
Wine	0.3	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
	0.6	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
		1.00×10^{-6}	4.64×10^{-6}	2.15×10^{-5}	1.00×10^{-4}
Yacht	0.3	0.29±0.16	0.31±0.15	0.25±0.15	0.25±0.18
	0.6	0.34±0.21	0.32±0.21	0.33±0.22	0.34±0.22
		3.16×10^{-6}	3.16×10^{-5}	3.16×10^{-4}	
Year	0.3	0.00±0.00	0.00±0.00	0.00±0.00	
	0.6	0.00±0.00	0.00±0.00	0.00±0.00	

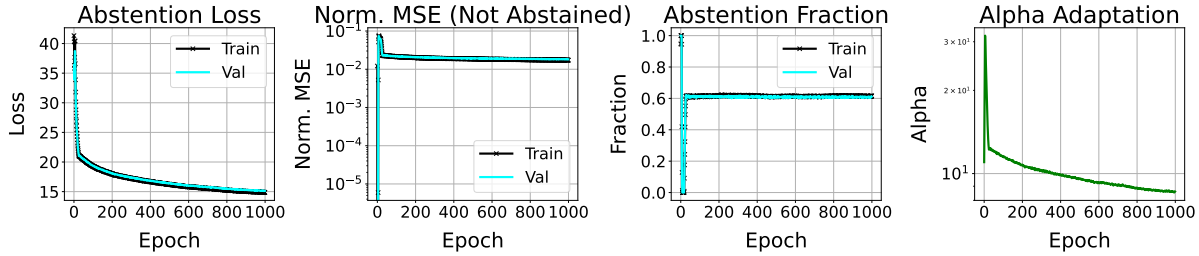
Table 3: Abstention fraction obtained in testing for UCI regression datasets.

In Tables 1 and 2, we report the *Best* and the *Worst* results obtained in the grid sweep. It can be seen that in most cases the abstention model matches or improves over the baseline MSE and heteroscedastic models, while it has slightly worse performance in a few cases. Table 3 contrasts the target maximum abstention fraction specified $f_{\text{max-abs}}$ vs. the abstention fraction obtained in testing. Results have been segregated in function of the different target maximum normalized non-abstained MSE loss $l_{\text{max-abs}}$ set. Mean $\pm$ the standard deviation over the different repetitions are reported. Note that under this training protocol, loss minimization is not fully converged and considerable standard deviations are observed in some cases. In addition, note that the obtained abstention fraction respects the target $f_{\text{max-abs}}$ specified in most cases. In some cases the model does not abstain, in which case the result is usually close to the baseline case. Note that because of the different loss function, the trajectory of the model during training can be different from original model, resulting in slightly different performance even without abstention. When the model does abstain, the performance is generally improved over the baseline, even though the training is not fully converged in many cases. This demonstrates

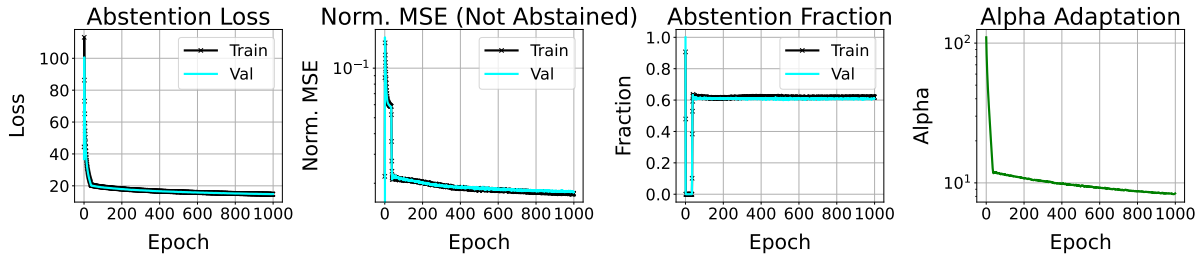
the ability of the approach to produce a robust model that identifies noisy patterns and achieves higher accuracy on the retained samples.



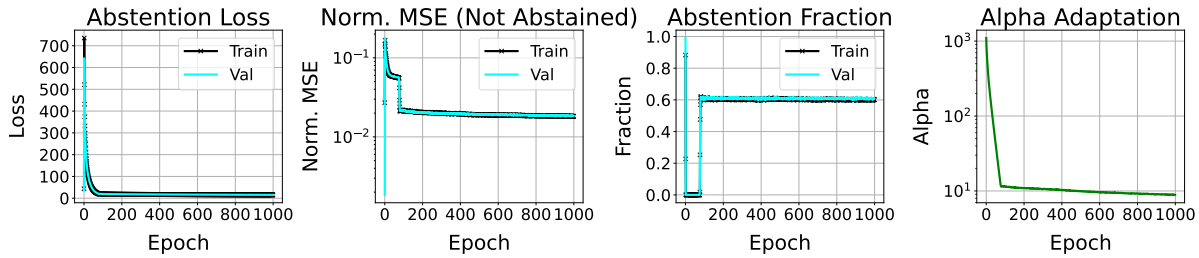
(a) $\alpha^{(0)} = 1.10$



(b) $\alpha^{(0)} = 11.00$



(c) $\alpha^{(0)} = 110.00$



(d) $\alpha^{(0)} = 1100.00$

Figure 10: Training evolution for the 'Protein' dataset with $l_{\text{max-abs}} = 1.00 \times 10^{-2}$ and $f_{\text{max-abs}} = 0.6$. Left-Column: Evolution of eq. (5) with adaptive α. Second-Column: Evolution of l, normalized MSE in Algorithm 1. Third-Column: Evolution of abstention fraction. Right-Column: Evolution of α.

To further illustrate the alpha adaptation and convergence of the proposed model, we choose the 'Protein' dataset and run it for 1,000 epochs with batch size of 100 for $l_{\text{max-abs}} = 1.00 \times 10^{-2}$ and $f_{\text{max-abs}} = 0.6$ using values of $\alpha^{(0)} = 1.10, 11.00, 110.00$ and 1100.00. Results for one realization are displayed in Figures 10a, 10b and 10c and 10d. It can be seen that, independent

of the initialization of α, the model converges to similar values of loss (15.18 ± 0.24), normalized MSE (0.018 ± 0.0003), abstention fraction (0.609 ± 0.0018) and α (8.76 ± 0.29). In all cases, the normalized MSE loss obtained, 0.018, is slightly larger than the target normalized loss, 0.01, and the abstention obtained, 60.9%, is slightly larger than the target abstention set, 60%. This is usually an indication of the best trade-off of noise and fraction of noisy samples achievable within the specific dataset for the given target conditions. Since there is no *a-priori* way of knowing what is the intrinsic noise structure in a dataset, it is useful to be able to run under different target balances and employ the obtained normalized MSE loss and abstention fraction as validation criteria.

5 CONCLUSION

In this work we extend the deep abstention classifier model to a formulation that makes it suitable to tackle regression problems. We introduce a two-stage training process that is able to label noisy samples as outliers, down-weighting their influence during training and identifying regions where the model is not confident and abstains from making predictions in testing. The second stage in the training process uses a heteroscedastic loss and allows to predict a standard deviation, following a univariate normal distribution model, which can then be used to estimate confidence intervals for the non-abstained predictions.

To allow for different maximum abstention fraction and maximum MSE loss target configurations, we use an adaptive criterion that dynamically balances these two, sometimes competing, goals during training. This adaptation may require small learning rates (10^{-4} or 10^{-5}) for stability. An even smaller learning rate (10^{-6}) may be necessary during the heteroscedastic stage to avoid NaN issues.

We demonstrate the performance of the model through extensive numerical experiments using synthetic and real datasets. The results obtained show the abstaining regression model is able to detect noisy samples and discard them from the regression fit while yielding robust detection of low-confidence samples and good estimations of uncertainty during testing.

ACKNOWLEDGEMENTS

This work has been supported by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health, and was performed under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory under Contract DE-AC5206NA25396. Approved for public release LA-UR-23-22064.

REFERENCES

[1] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, and J. Mohd-Yusof, "Combating label noise in deep learning using abstention," in *Proceedings International Conference on Machine Learning*, 2019.

[2] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 5574–5584. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf

[3] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1139–1147. [Online]. Available: https://proceedings.mlr.press/v28/sutskever13.html

[4] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[6] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proceedings International Conference on Machine Learning ICML'15*, vol. 37, July 2015, pp. 1861 – 1869.

[7] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, U. von Luxburg, I. Guyon, S. Bengio, H. Wallach, and R. Fergus, Eds., vol. 31. Curran Associates, Inc., 2017, pp. 6405–6416. [Online]. Available: https://dl.acm.org/doi/10.5555/3295222.3295387