

# Deep **Anti-Regularized** Ensembles provide reliable out-of-distribution uncertainty quantification

Antoine de Mathelin<sup>1,2</sup>, Francois Deheeger<sup>1</sup>, Mathilde Mougeot<sup>2</sup>, and Nicolas Vayatis<sup>2</sup>

<sup>1</sup> Manufacture Française des Pneumatiques Michelin, Clermont-Ferrand, France  
 {antoine.de-mathelin-de-papigny, francois.deheeger}@michelin.com

<sup>2</sup> Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, Gif-sur-Yvette, France {mathilde.mougeot, nicolas.vayatis}@ens-paris-saclay.fr

**Abstract.** We consider the problem of uncertainty quantification in high dimensional regression and classification for which deep ensemble have proven to be promising methods. Recent observations have shown that deep ensemble often return overconfident estimates outside the training domain, which is a major limitation because shifted distributions are often encountered in real-life scenarios. The principal challenge for this problem is to solve the trade-off between increasing the diversity of the ensemble outputs and making accurate in-distribution predictions. In this work, we show that an ensemble of networks with large weights fitting the training data are likely to meet these two objectives. We derive a simple and practical approach to produce such ensembles, based on an original anti-regularization term penalizing small weights and a control process of the weight increase which maintains the in-distribution loss under an acceptable threshold. The developed approach does not require any out-of-distribution training data neither any trade-off hyper-parameter calibration. We derive a theoretical framework for this approach and show that the proposed optimization can be seen as a "water-filling" problem. Several experiments in both regression and classification settings highlight that Deep Anti-Regularized Ensembles (DARE) significantly improve uncertainty quantification outside the training domain in comparison to recent deep ensembles and out-of-distribution detection methods. All the conducted experiments are reproducible and the source code is available at <https://github.com/antoinedemathelin/DARE>.

**Keywords:** Deep Ensemble · Uncertainty · Out-of-distribution · Anti-regularization

## 1 Introduction

With the adoption of deep learning models in a variety of real-life applications such as autonomous vehicles [4,9], or industrial product certification [23], providing uncertainty quantification for their predictions becomes critical. Indeed, various adaptations of classical uncertainty quantification methods to deep learning predictions have been recently introduced as Bayesian neural networks [22,27],

MC-dropout [11], quantile regression [32] and deep ensembles [17,42,43]. These methods appear to be quite efficient in predicting the uncertainty in the training domain (the domain defined by the training set), called in-distribution or aleatoric uncertainty [1]. However, when dealing with data outside the training distribution, i.e. out-of-distribution data (OOD), the uncertainty estimation often appears to be overconfident [6,20,28]. This is a critical issue, because deep models are often deployed on shifted distributions [24,34,44]; overconfidence on an uncontrolled domain can lead to dramatic consequences in autonomous cars or to poor industrial choices in product design.

One problem to be solved is to increase the output diversity of the ensemble in regions where no data are available during training. This is a very challenging task as neural network outputs are difficult to control outside of the training data regions. In this perspective, contrastive methods make use of real [29,39] or synthetic [14,26,36] auxiliary OOD data to constrain the network output out-of-distribution. However, these approaches can not guarantee prediction diversity for unseen OOD data as the auxiliary sample may not be representative of real OODs encountered by the deployed ensemble. Another set of methods assumes that the diversity of the ensemble outputs is linked to the diversity of the networks' architectures [45], hyper-parameters [43], internal representations [31,38] or weights [5,30]. The main difficulty encountered when using these approaches is to solve the trade-off between increasing the ensemble diversity and providing accurate prediction in-distribution. The current approach to deal with this issue consists in setting a trade-off parameter with hold-out validation [13,21,30] which is time consuming and often penalizes the diversity.

Considering these difficulties, a question that naturally arises is how to ensure important output diversity for any unknown data region while maintaining accurate in-distribution predictions? In this work, we tackle this question with the following reasoning: an ensemble of large weights networks essentially produces outputs with large variance for all data points. Furthermore, to make accurate prediction on the training distribution, the output variance for training data needs to be reduced, which requires that some of the network's weights are also reduced. However, to prevent the output variance from being reduced anywhere other than the training domain, the network weights should then be kept as large as possible. Following this reasoning, we seek an ensemble providing accurate prediction in-distribution and keeping the weights as large as possible.

To meet these two objectives, deviating from traditional recommendations for deep learning training, we propose an anti-regularization process that consists in penalizing small weights during training optimization. To find the right trade-off between increasing the weights and providing accurate prediction in-distribution, we introduce a control process that activates or deactivates the weight increase after each batch update if the training loss is respectively under or above a threshold. Thus, an increase of the weights induces an increase of the prediction variance while the control on the loss enforces accurate in-distribution predictions. Synthetic experiments on toy datasets confirm the efficiency of our proposed approach (cf Figure 1). We observe that the uncertainty estimates of our Deep

Anti-Regularized Ensembles (DARE) increase for any data point deviating from the training domain, whereas, for the vanilla deep ensemble, the uncertainty estimates remain low for some OOD regions.

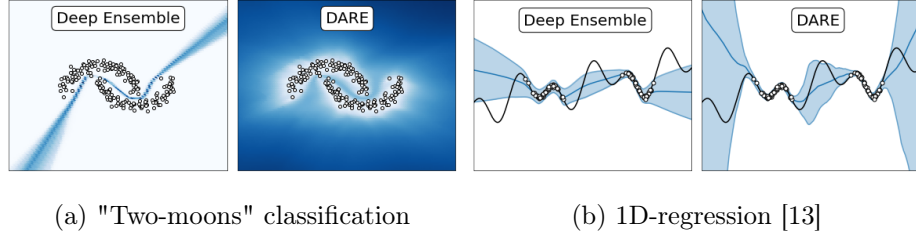


Fig. 1: **Synthetic datasets uncertainty estimation.** White points represent the training data. For each experiment, the ensemble are composed of 20 neural networks. For classification, darker areas correspond to higher predicted uncertainty. For regression, the confidence intervals for  $\pm 2\sigma$  are represented in light blue. The full experiment description is reported in Appendix.

The contributions of the present work are the following:

- A novel and simple anti-regularization strategy is proposed to increase deep ensemble diversity.
- An original control process addresses the trade-off issue between in-distribution accuracy and reliable OOD uncertainty estimates.
- We provide theoretical arguments to understand DARE as a "water-filling" optimization problem where a bounded global amount of variance is dispatched among the network weights.
- A new experimental setup for uncertainty quantification with shifted distribution is developed for regression. Experiments are also conducted for out-of-distribution detection following the setup of [5].

## 2 Deep Anti-Regularized Ensemble

### 2.1 Notations

We consider the supervised learning scenario where  $\mathcal{X} \subset \mathbb{R}^p$  and  $\mathcal{Y} \subset \mathbb{R}^q$  are respectively the input and output space, with  $p, q \in \mathbb{N}$ . The learner has access to a training sample,  $\mathcal{S} = \{(x_1, y_1), \dots, (x_n, y_n)\} \in \mathcal{X} \times \mathcal{Y}$  of size  $n \in \mathbb{N}$ . We consider a set  $\mathcal{H}$  of neural networks  $h_\theta \in \mathcal{H}$  where  $\theta \in \mathbb{R}^d$  refers to the network weights. We consider a loss function  $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$  and define the average error of any  $h_\theta \in \mathcal{H}$  on  $\mathcal{S}$ ,  $\mathcal{L}_\mathcal{S}(h_\theta) = \frac{1}{n} \sum_{(x_i, y_i) \in \mathcal{S}} l(h_\theta(x_i), y_i)$ .

## 2.2 Optimization formulation

The main assumption of this present work is that accurate in-distribution prediction and output diversity for OOD can be obtained by increasing the networks' weights as much as possible while maintaining the training loss under a threshold. We further assume that the variance is increased when the weights of each network is increased. Based on these assumptions, we propose the simple following optimization for every member  $h_\theta \in \mathcal{H}$  of the ensemble:

$$\min_{\theta} \mathcal{L}_S(h_\theta) - \lambda \mathcal{R}(\theta) \quad (1)$$

with  $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}_+$  a monotone function growing with  $\|\theta\|_2$ . The parameter  $\lambda \in \{0, 1\}$  is a binary variable which controls the trade-off between the in-distribution loss and the regularization. At each batch computation,  $\lambda$  is updated as follows:

$$\lambda = \begin{cases} 1 & \text{if } \mathcal{L}_S(h_\theta) \leq \tau \\ 0 & \text{if } \mathcal{L}_S(h_\theta) > \tau \end{cases} \quad (2)$$

with  $\tau \in \mathbb{R}$  a predefined threshold which defines the acceptable level of performance targeted by the learner.

- The first term of the optimization objective in Eq. (1):  $\mathcal{L}_S(h_\theta)$  is the loss in-distribution. This term induces the network to fit the training data which implies smaller in-distribution prediction variances.
- The second term  $-\lambda \mathcal{R}(\theta)$  acts as an "anti-regularization" term which induces an increase of the network weights. This implies a larger variance of the ensemble weights, and therefore a larger prediction variance especially for data "far" from the training distribution on which the network's predictions are not conditioned.

The underlying idea of the proposed optimization is that, to fulfill both objectives: reducing the loss in-distribution and increasing the weights, large weights will appear more likely in front of neurons which are never or weakly activated by the training data. Therefore, if an out-of-distribution data point activates one of these neurons, large values are propagated through the networks which induces larger prediction variances. We show in Section 4, that this claim is supported by theoretical analysis and empirical observations.

The control process is necessary to temper the weight increase, because a large increase of the weights induces an unstable network with reduced accuracy on training data. To be sure to fulfill a performance threshold  $\tau$ , the weight increase is stopped ( $\lambda = 0$ ) until the loss in-distribution comes back under the threshold. Therefore, the resulting ensemble is composed of networks fitting the training data with weights as large as possible.

## 3 Implementation

**Parallel optimization.** Each network of the ensemble is trained independently of the others with the objective of Eq. (1). This approach allows parallel training

of the ensemble. It is theoretically possible that each network ends up reaching the same optimum resulting in no ensemble diversity. However, we observe that this degenerate case never occurs in our experiments due to the random process of the optimization and aleatoric weights initialization.

**Regularization function.** We propose the following choice of regularization function:

$$\mathcal{R}(\theta) = \frac{1}{d} \sum_{k=1}^d \log(\theta_k^2) \quad (3)$$

With  $\theta = (\theta_1, \dots, \theta_k)$  the network weights. The use of the logarithmic function is motivated by the "water-filling" interpretation of DARE (cf. Section 4).

**Control threshold.** The control threshold  $\tau$  should be chosen by the learner based on his targeted error level in-distribution. Smaller  $\tau$  leads to smaller increase of the weights. For  $\tau = -\infty$ , DARE is equivalent to a vanilla deep ensemble. A selection heuristic for  $\tau$  is proposed and discussed in Section 6.

**Optimization algorithm** Each network is trained through batch gradient descent. The optimization algorithm is described in Algorithm 1.

---

**Algorithm 1** Anti-Regularized Network

---

```

1: Inputs: Training set  $\mathcal{S}$ , threshold  $\tau$ , learning rate  $\mu$ 
2: Outputs: network  $h_\theta$ 
3: Init:  $\theta \leftarrow \text{random}$ 
4: while stopping criterion is not reached do
5:   if  $\mathcal{L}_{\mathcal{S}}(h_\theta) \leq \tau$  then
6:      $\theta \leftarrow \theta - \mu \nabla_\theta \left[ \mathcal{L}_{\mathcal{S}}(h_\theta) - \frac{1}{d} \sum_{k=1}^d \log(\theta_k^2) \right]$ 
7:   else
8:      $\theta \leftarrow \theta - \mu \nabla_\theta [\mathcal{L}_{\mathcal{S}}(h_\theta)]$ 
9:   end if
10: end while

```

---

## 4 Theoretical Analysis

The purpose of the following theoretical analysis section is to support the fundamental claim that : "large weights do not activate on in-distribution samples while mostly activate for OOD, leading to larger output variance". We first provide theoretical insights on a simple case of linear networks which is then extended to multi-layer fully-connected networks. In a third part, an in-depth analysis of the layer activations is conducted on a syntetic dataset to illustrate this leveraged assumption.

#### 4.1 Theoretical insights on linear networks

In the following, we propose theoretical insights for anti-regularized ensemble of linear networks (equivalent to linear regression). This theoretical result helps to understand the underlying dynamic of DARE. In particular, we show both following results which hold under appropriate assumptions:

- The weights' variance of the DARE networks is inversely proportional to the variance of the input features (Theorem 1).
- As a consequence of the previous results, the DARE prediction variance is very sensitive to small deviations of low variance input features (Corollary 1).

To formalize and demonstrate these results we consider the regression problem with scalar outputs ( $q = 1$ ) where  $X \in \mathbb{R}^{n \times p}$  and  $y \in \mathbb{R}^n$  are respectively the input and output data of the training set  $\mathcal{S}$ . We consider the ensemble of linear networks  $\mathcal{H} = \{x \rightarrow x^T \theta; \theta \in \mathbb{R}^p\}$  such that, any  $h_\theta \in \mathcal{H}$  is composed of two layers: the input and output layers of respective dimension  $p$  and 1. The loss function is the squared error such that for any  $h_\theta \in \mathcal{H}$  we have  $n\mathcal{L}_\mathcal{S}(h_\theta) = \|X\theta - y\|_2^2$ . To simplify the calculation without losing in generality, we assume that the problem is linear, i.e. there exists  $\theta^* \in \mathbb{R}^p$  such that  $\mathcal{L}_\mathcal{S}(h_{\theta^*}) = 0$ . We denote  $s^2 = (s_1^2, \dots, s_p^2) \in \mathbb{R}_+^p$  the diagonal of the matrix  $\frac{1}{n}X^T X$ . We now consider an anti-regularized ensemble  $\mathcal{H}_\tau$  produced by Algorithm 1. To characterize this ensemble, we make the following assumptions for any  $h_\theta \in \mathcal{H}_\tau$ :

$$\theta \sim \Theta_{\sigma^2}; \quad \mathbb{E}[\theta] = \theta^*, \quad \text{Cov}(\theta) = \text{diag}(\sigma^2) \quad (4)$$

$$\mathbb{E}[\mathcal{L}_\mathcal{S}(h_\theta)] \leq \delta \tau \quad (5)$$

Where  $\delta > 0$  and  $\text{diag}(\sigma^2)$  is the diagonal matrix of values  $\sigma^2 \in \mathbb{R}_+^p$  verifying:

$$\sigma^2 = \underset{\sigma^2 = (\sigma_1^2, \dots, \sigma_p^2)}{\text{argmax}} \sum_{k=1}^p \log(\theta_k^{*2} + \sigma_k^2) \quad (6)$$

As presented in Assumption (4), the anti-regularized ensemble distribution is described by the random variable  $\theta$  centered in  $\theta^*$  with variance  $\sigma^2$ . Assumption (5) implies that  $P(\mathcal{L}_\mathcal{S}(h_\theta) \geq \tau) \leq \delta$ , by Markov inequality, which models the fact that the loss of each member of DARE is maintained above a threshold  $\tau$  thanks to the control process on  $\lambda$  (cf Section 2.2). Definition (6) approximates the impact of the anti-regularization term  $-\mathcal{R}(\theta)$  in the DARE optimization formulation with an upper bound of  $\max_\sigma \mathbb{E}[\mathcal{R}(\theta)]$ . The weights are increased as much as possible while the loss stays under the threshold.

Our first theoretical result shows that the weight variance of the anti-regularized ensemble is solution of a "water-filling" optimization problem [2], and is proportional to  $1/s^2$ , i.e. the inverse of the input features variance.

**Theorem 1.** *There exist a constant  $C > 0$  such that for any  $k \in [1, p]$ , the variance of the  $k^{\text{th}}$  weight component is expressed as follows:*

$$\sigma_k^2 = \max \left[ \frac{C \delta \tau}{s_k^2} - \theta_k^{*2}, 0 \right] \quad (7)$$

*Sketch of Proof.* A detailed proof is reported in Appendix. The proof consists in first noticing that  $\mathbb{E}[\mathcal{L}_{\mathcal{S}}(h_{\theta})] = \sum_{k=1}^p s_k^2 \sigma_k^2$ . By combining this result with Assumptions 5 and 6, we show that  $\sigma^2$  is solution of the above water filling problem:

$$\begin{aligned} & \underset{\sigma^2 \in \mathbb{R}_+^p}{\text{maximize}} && \sum_{k=1}^p \log(\sigma_k^2 + \theta_k^{*2}) \\ & \text{subject to} && \sum_{k=1}^p s_k^2 \sigma_k^2 \leq \delta \tau \end{aligned} \quad (8)$$

As the log function is strictly concave, and the constraints form a compact set on  $\mathbb{R}^p$ , the problem (8) has a unique solution which is given by (7).

The "water-filling" interpretation of the DARE optimization (8) is very insightful:  $\delta \tau$  is the "global variance capacity" that can be dispatched to the network weights. As, it grows as a function of  $\tau$ , the more the learner accept a large error in-distribution, the more the global variance capacity increases. We stress that each weight component has a different "variance cost" equal to  $s_k^2$ : for high feature variance  $s_k^2$ , the increase of the corresponding weight variance  $\sigma_k^2$  penalizes more the training loss. Thus, large weights appear more likely in front of low variance features. Notice also that, when  $\theta_k^{*2}$  is high,  $\frac{C \delta \tau}{s_k^2} - \theta_k^{*2}$  is more likely to be negative, leading to  $\sigma_k = 0$  (cf Eq. (7)). Besides,  $\theta_k^{*2}$  is generally higher for higher  $s_k^2$  as it corresponds to more informative feature, enhancing the effect  $\sigma_k = 0$  for large  $s_k^2$ .

We see the importance of choosing a strictly concave function like the logarithm (cf Section 3), if instead of log, we choose the identity function for instance, then the solution of (8) degenerates to  $\sigma^2 = \left(0, \dots, 0, \frac{\delta \tau}{s_p^2}\right)$  with  $s_p^2$  the lowest feature variance. In this case, all the weight variance is assigned to one component, which reduces the likelihood to detect a deviation of a potential OOD input on another low variance feature.

From Theorem 1, we now derive the expression of the DARE prediction variance for any data  $x \in \mathbb{R}^p$ :

**Corollary 1.** *Let  $\mathcal{H}_{\tau}$  be the large weights ensemble defined by Assumptions 4, 5, 6 and  $x \in \mathbb{R}^p$  an input data, the variance of prediction for  $h_{\theta} \in \mathcal{H}_{\tau}$  is:*

$$\text{Var}_{\theta \sim \Theta_{\sigma^2}} [h_{\theta}(x)] = \sum_{k=1}^p x_k^2 \max \left[ \frac{C \delta \tau}{s_k^2} - \theta_k^{*2}, 0 \right] \quad (9)$$

We deduce from Corollary 1 that the prediction variance for  $x$  is large when the components  $x_k^2$  are large for features with low variance ( $s_k^2 \ll 1$ ). Thus, the predicted uncertainty of DARE is correlated with deviations in directions in which the training input data has small variance.

## 4.2 Extending the results to multi-layer fully connected networks

Transposed to the hidden layers of a multi-layer neural network, Theorem 1 means that higher network weights likely appear in front of neurons which are weakly activated by the training data. To generalize Corollary 1 to the multi-layer case, we introduce an anti-regularized ensemble  $\mathcal{H}_\tau$  of fully-connected deep neural networks with  $L$  hidden layers and network weights distributed according to  $\Theta_{\sigma^2}$  in  $\mathbb{R}^d$ . We denote  $p_\ell > 0$ , the number of neurons of the layer  $\ell \in [0, L + 1]$ . We denote  $g : \mathbb{R} \rightarrow \mathbb{R}$  the activation function applied at each hidden layer. For every  $\ell \in [0, L]$ , we denote  $\phi_{(\ell)}(X) \in \mathbb{R}^{n \times p_\ell}$  the representation of the  $\ell^{th}$  layer, with  $\ell = 0$  the input layer. For any  $j \in [1, p_{\ell+1}]$ , we denote  $\theta_{(\ell,j)} \in \mathbb{R}^{p_\ell}$  the network weights between the layer  $\ell$  and the  $j^{th}$  component of the layer  $\ell + 1$ .

We assume that, for any random variable  $Z$  in  $\mathbb{R}$  there exists  $\alpha > 0$  such that  $\text{Var}[g(Z)] > \alpha \text{Var}[Z]$ . This condition is true for the leaky-relu activation, and also true for other classic activations as relu, softplus and sigmoid if  $Z$  is centered. We define  $\phi_{(\ell)}^*(X) = \mathbb{E}[\phi_{(\ell)}(X)]$  and denote  $s_{(\ell)}^2$  the diagonal of the matrix  $\frac{1}{n} \phi_{(\ell)}^*(X)^T \phi_{(\ell)}^*(X)$ .

We finally assume that, there is  $\delta > 0$  such that, for any  $\ell \in [0, L]$  and any  $j \in [1, p_{\ell+1}]$ ,  $\phi_{(\ell)}(X)$  and  $\theta_{(\ell,j)}$  verify Assumptions (4), (5), (6) with  $\theta \equiv \theta_{(\ell,j)}$ ,  $X \equiv \phi_{(\ell)}^*(X)$ ,  $y \equiv \phi_{(\ell)}^*(X) \theta_{(\ell,j)}^*$  and  $\theta^* \equiv \theta_{(\ell,j)}^* \equiv \mathbb{E}[\theta_{(\ell,j)}]$

**Theorem 2.** *There exists a constant  $\gamma > 0$  such that, for any  $h_\theta \in \mathcal{H}_\tau$  and  $x \in \mathbb{R}^p$ , the prediction variance verifies:*

$$\text{Var}_{\theta \sim \Theta_{\sigma^2}} [h_\theta(x)] \geq \gamma \sum_{\ell=0}^L \langle \phi_{(\ell)}^*(x)^2, \bar{\sigma}_{(\ell)}^2 \rangle$$

$$\text{With, } \bar{\sigma}_{(\ell)}^2 = \sum_{j=1}^{p_{\ell+1}} \max \left[ \frac{C_{(\ell,j)}}{s_{(\ell)}^2} - \theta_{(\ell,j)}^{*2}, 0 \right]; C_{(\ell,j)} \in \mathbb{R}_+^*$$

Theorem 2 states that the prediction variance for any  $x \in \mathcal{X}$  is a growing function of  $\langle \phi_{(\ell)}^*(x)^2, 1/s_{(\ell)}^2 \rangle$ , which means that the uncertainty predicted by DARE is high if one hidden neuron weakly activated by the training data is strongly activated by  $x$ . Figure 2 shows an illustration of the expected behavior of DARE for one OOD data  $x_{ood}$ . The underlying assumption is that the distance  $D$  between  $x_{ood}$  and the learning domain is expressed on one hidden feature with low variance for the training data.

## 4.3 Verification on a synthetic dataset

We consider the "two-moons" binary classification dataset and use a three hidden-layer network with 100 neurons each and ReLU activations as base network. We use a linear activation for the output layer and the mean squared error as loss function. As we consider a classification task, uncertainty can be obtained with a single network through the formula:  $\min(h_\theta(x)^2, (1 - h_\theta(x))^2)$ . The DARE



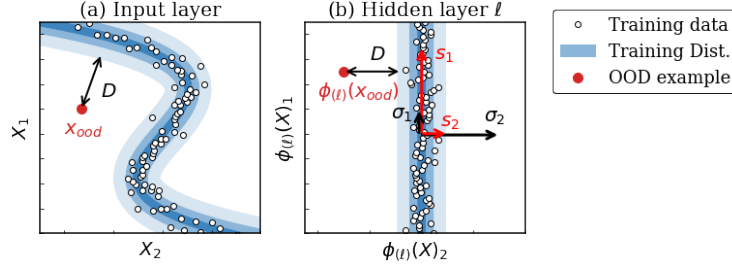


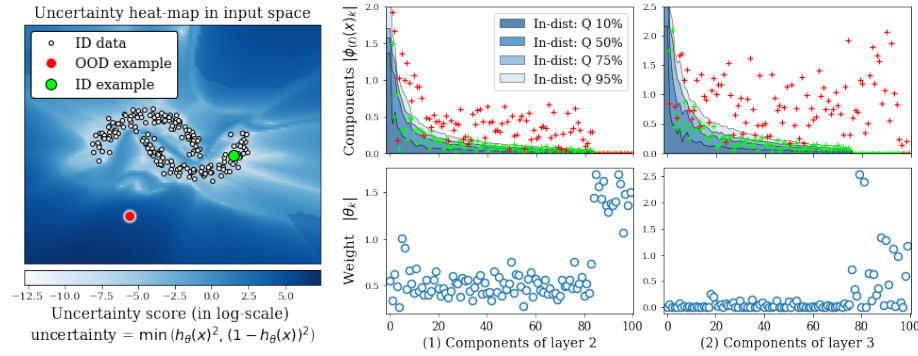
Fig. 2: **Schematic description of DARE behavior:** an OOD data  $x_{ood}$  is distant from  $D > 0$  from the training domain. In a hidden layer  $\ell$  of the network  $h_\theta$ , the training domain is composed of a strong component  $\phi_{(\ell)}(X)_1$  of variance  $s_1^2 \gg 1$  and a weak component  $\phi_{(\ell)}(X)_2$  of variance  $s_2^2 \ll 1$ . In this representation,  $\phi_{(\ell)}(x_{ood})$  is distant from  $D$  to the training domain in the weak direction  $\phi_{(\ell)}(X)_2$ , resulting in  $\text{Var}[h_\theta(x_{ood})] \geq \gamma D^2 \sigma_2^2 \simeq \frac{\gamma D^2}{s_2^2} \gg 1$ .

uncertainty is the sum of the individual network members uncertainty plus the prediction variance between members.

Figure 3 presents the predicted uncertainty heat-map for one DARE network as well as the internal layer representations. We observe that the OOD component values grow from one layer to the next. A correspondence between features with low variance for the training data and large weights can be clearly observed. In the last hidden layer (layer 3), the OOD components are large in direction of low training variance (components 80 to 100) to which correspond large weights. This observation explains the large uncertainty score for the OOD example.

## 5 Related Works

Increasing ensemble diversity has been an enduring paradigm since the early days of the ensemble learning research. At first, diversity was seen as a key feature for improving the generalization ability of the ensembles [3,16,21,46]. Then, with the growing interest in uncertainty quantification, the primary objective of ensemble diversity becomes to produce good uncertainty estimates. In this perspective, a first category of methods propose to increase diversity by using diverse architectures or training conditions among an ensemble of deep neural networks [17,42,43,45]. The underlying idea is that the diversity of architecture or local minima reached by the different networks induces a diversity of predictions. Another category of methods proposes to explicitly impose a diversity constraint in the loss function of the networks. The loss function is then written  $\mathcal{L} + \lambda \mathcal{P}$  where  $\mathcal{L}$  is the loss for the task (e.g. mean squared error or negative log-likelihood (NLL)),  $\mathcal{P}$  is a penalty term which decreases with the diversity of the ensemble and  $\lambda$  is the trade-off parameter between the two terms. Three kinds of penalization are distinguished in the literature. The first kind makes use of training data



**Fig. 3: Internal analysis of a large weights network** The uncertainty produced by one member of an anti-regularized ensemble is presented on the left. On the right, the two figures on top present the expression of the training distribution in the three hidden layers (in blue) compared to the representation of one OOD example (in red). The components are sorted in descending order of training variance. The two bottom figures present the average weight in front of each component, i.e. the mean weights that multiply the layer components to produce the next layer representation.

to compute the penalty term. It includes the Negative Correlation method (NegCorr) [37,48] which applies the penalization from [21] to deep ensembles to enforce a negative correlation between the errors of the networks on the training set. Similarly, [33] imposes an orthogonality constraint between the gradients of the ensemble members on training data. Penalizing the similarity between hidden representations of the networks has also been proposed by [31,38] using adversarial training. The second kind of penalization refers to contrastive methods that enforces diversity on potential OOD instances rather than training data. This avoids the issue of being over-conditioned by the training domain that can be encountered by previous methods. In this category, several methods suppose that an unlabeled sample containing OOD is available, [29,39]. Others avoid this restrictive assumption and simulate potential OOD with random uniform data [14,26] or instances localized around the training data [36]. In the last approach, considered by Anchored-Networks (AnchorNet) [30] and Repulsive Deep Ensemble (RDE) [5], the penalization  $\mathcal{P}$  is a function of the network's parameters which forces the networks to reach local minima spaced from each other in parameters space. Our proposed DARE approach relates particularly to these two methods. Our assumption is that imposing weights diversity has more chance to obtain a global output diversity rather than imposing it on specific data regions as done by the two previous kind of penalization. Anchored-Networks appears to be an efficient tool, for instance, in the detection of corrupted data [40], however, it is very hard to set the right anchors and trade-off parameter [35]. Large initial variance can lead to large weight variance but may not converge

to accurate model in-distribution. The DARE approach is more practical as it starts to increase the weights after reaching an acceptable loss threshold which ensures to fit the training data.

## 6 Experiments

The experiments are conducted on both regression and classification datasets. In the majority of previous works, OOD uncertainty quantification is studied in the perspective of OOD detection in the classification setting where examples from other classes / datasets are considered as OOD [5,17,19,41]. For regression, few attempts of uncertainty quantification on shifted datasets have been conducted: [14] separates male and female faces for age prediction dataset and [10,14,36] propose OOD version of several UCI regression datasets [8]. In this work, we propose a novel regression setup for uncertainty estimations on shifted domains based on the CityCam dataset [47] which has been used in several domain adaptation regression experiments [25,49]. Our setup models real-life domain shift scenarios where uncertainty quantification is challenged and offers a clear visual understanding of the domain shifts (cf Figure 4). For the classification experiments, we consider the OOD detection setup developed in [5]. The source code of the experiments is available on GitHub<sup>3</sup>.

### 6.1 Regression experiments on CityCam

We consider the CityCam dataset [47]. This dataset is composed of images recorded from different traffic cameras. The task consists in counting the number of vehicles present on the image, which is useful, for instance, to control the traffic in the city. To get relevant features for the task, we use the features of the last layer of a ResNet50 [12] pretrained on ImageNet [7]. We propose three different kinds of domain shift:

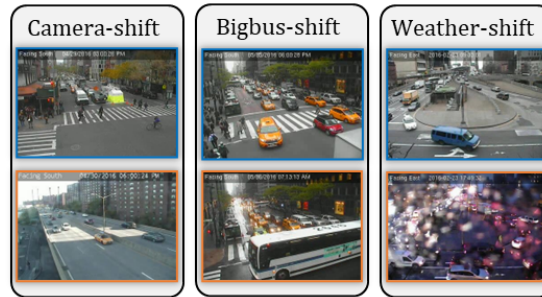


Fig. 4: **CityCam experimental setups.** The top blue images correspond to in-distribution examples and bottom orange images to OOD examples.

<sup>3</sup> <https://github.com/antoinedemathelin/DARE>

**1. Camera-shift:** This experiment uses the images from 10 cameras in the CityCam dataset. For each round, 5 cameras are randomly selected as in-distribution while the 5 remaining cameras are considered as out-of-distribution.

**2. Bigbus-shift:** Images marked as "big-bus" referring to the fact that a bus appears and masks a significant part of the image [47] are used to create the OOD dataset.

**3. Weather-shift:** Blurry images caused by water drops landed on the camera are used as OOD dataset.

These three experiments model real-life uncertainty quantification problems as the generalization of uncertainty estimates to unseen domains (camera-shift), the robustness to changes in data acquisition (weather-shift) and the detection of rare abnormal event (bigbus-shift). Further details on these experimental setups are provided in Appendix.

We consider the following competitors: **Deep-Ensemble (DE)** [17], **Neg-Corr** [37], **AnchorNet** [30], **MOD** [14] and **RDE** [5]. All are deep ensemble methods which focus on increasing the diversity among members. AnchorNet, NegCorr and MOD introduce a penalty term in the loss function multiplied by a trade-off parameter  $\lambda$ . The trade-off  $\lambda$  and the anchor initialization parameter  $\sigma$  for AnchorNet are selected through hold-out validation, as suggested in [14,30]. The kernel bandwidth for RDE is adaptive and chosen with the median heuristic as suggested in the corresponding work [5]. The parameter  $\tau$  of DARE is chosen equal to  $1 + \delta$  times the DE validation loss. The underlying heuristic is to maintain the in-distribution performance level close to that of the vanilla DE. However, a small penalty  $\delta > 0$  should be accepted to allow the weight increase. In these experiments, we intuitively select  $\delta = 0.25$  without further fine-tuning. An ablation study of the impact of the  $\delta$  parameter on the DARE performances is presented in Appendix.

The experiments are performed with ensemble of 5 fully-connected networks with 3 hidden layers of 100 neurons each and ReLU activations. The Adam optimization algorithm is used with learning rate 0.001 [15]. The batch size is chosen equal to 128. The experiments are repeated 5 times to compute standard deviations for the scores. We use the NLL defines in [17] as loss function.

Choosing the right number of optimization epochs to perform is not trivial. Previous works have used a fixed number of epochs [30] or early stopping [14]. For DARE, a natural checkpoint strategy is to save the network if the validation loss of the current epoch is below the threshold  $\tau$ . This process ensures a targeted level of in-distribution performance and allows the weight increase during several epochs. For the competitors, we observed that restoring the weights of the best validation loss epoch provides the best results.

To assess the ensemble quality for the regression experiments, we consider the NLL and expected calibration error (ECE) computed on the out-of-distribution dataset as used in the works [17], [26]. The results are averaged over 5 runs and reported in Table 1. We observe that DARE outperforms other uncertainty quantification methods in both term of NLL and ECE for the bigbus-shift and camera-shift experiments. For the weather-shift experiment, DARE provides

Methods	Camera-shift		Bigbus-shift		Weather-shift	
	NLL	ECE	NLL	ECE	NLL	ECE
DE	6.04 (0.40)	0.28 (0.02)	3.66 (0.17)	0.28 (0.01)	2.25 (0.09)	0.12 (0.02)
AnchorNet	6.03 (0.24)	0.28 (0.02)	3.86 (0.16)	0.29 (0.01)	2.25 (0.12)	0.12 (0.03)
MOD	6.24 (0.89)	0.29 (0.02)	3.66 (0.21)	0.29 (0.01)	2.32 (0.17)	0.15 (0.03)
NegCorr	5.97 (0.54)	0.29 (0.03)	4.02 (0.27)	0.29 (0.01)	2.30 (0.10)	0.14 (0.02)
RDE	5.05 (0.25)	0.27 (0.02)	3.67 (0.11)	0.28 (0.00)	<b>2.22 (0.14)</b>	0.13 (0.04)
DARE	<b>3.64 (0.34)</b>	<b>0.23 (0.03)</b>	<b>2.99 (0.13)</b>	<b>0.25 (0.01)</b>	2.33 (0.13)	<b>0.07 (0.04)</b>

Table 1: Out-of-distribution NLL and ECE for CityCam

	DE	AnchorNet	MOD	NegCorr	RDE	DARE
Cameras	1.03 (0.08)	<b>1.00 (0.07)</b>	1.04 (0.06)	1.04 (0.06)	1.06 (0.06)	1.18 (0.09)
Bigbus	0.89 (0.03)	<b>0.87 (0.02)</b>	0.91 (0.04)	0.89 (0.06)	0.91 (0.03)	1.06 (0.04)
Weather	1.25 (0.04)	<b>1.24 (0.05)</b>	1.26 (0.05)	1.26 (0.06)	1.28 (0.05)	1.27 (0.06)

Table 2: In-distribution NLL for CityCam

the best ECE score. Moreover, these gains in uncertainty estimation come with little penalization of the performance in-distribution as reported in Table 2: in the camera-shift experiment for instance, the in-distribution NLL of DARE is increased of only 0.18 compare to the deep ensemble method, while the NLL out-of-distribution is decreased of 2.4 (cf Table 1). The impact of the  $\lambda$  control in the DARE optimization is presented in Figure 5, we observe that the setting  $\lambda = 0$  (equivalent to DE) leads to a small weight increase whereas the setting  $\lambda = 1$  leads to a loss explosion. The control setting  $\lambda \in \{0, 1\}$  allows the weight increase while maintaining the loss above  $\tau$ .

## 6.2 Classification Experiments

We consider the experimental setup defines in [5] for OOD detection on Fashion-MNIST and CIFAR10. The MNIST dataset is used as OOD dataset for Fashion-MNIST and the SVHN dataset for CIFAR10. We extend the experiments by adding CIFAR10 as OOD for Fashion-MNIST and CIFAR100 as OOD for CIFAR10. Thus, for both experiments, OOD detection is performed on both "Near-OOD" and "Far-OOD" datasets [19].

We first observed that a direct application of DARE to this setup leads to negative results. The first issue comes from the softmax activation at the end layer, which cancels the effect of increasing the weights. Indeed, the softmax activation inverses the correlation between large outputs and high uncertainty, resulting in over-confidence for OOD data. A second issue comes from the use of convolutional layers in the CIFAR10 experiment. These layers have fewer parameters than fully-connected ones, which reduces the possibility of increasing the weights in front of weakly activated neurons.

To overcome these difficulties, considering that DARE is efficient for fully-connected networks and linear end-activation, we remove the softmax activation at the end-layer of the DARE networks and use the mean squared error. Uncertainty

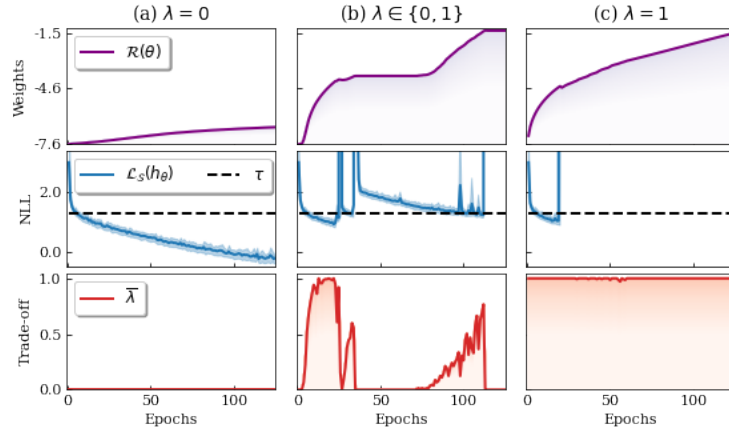


Fig. 5: **Training behaviors for different  $\lambda$  settings.** Evolution of the weights, the training loss and the average trade-off parameter over the epochs ((a)  $\equiv$  DE and (b)  $\equiv$  DARE).

scores are then obtained by computing the distance between the predicted logits and the one-hot-encoded vector of the predicted class. For the CIFAR10 experiment, we use a pretrained ResNet32 and replace the end layer by deep ensemble of fully-connected networks. This setup is more practical, as only one ResNet32 is required, which reduces the need of computational resources [18].

The obtained results are reported in Table 3 for DARE and the competitors. To fully evaluate the impact of the DARE optimization, we add the results obtained with a Deep Ensemble trained with the mean squared error (DE (MSE)) which is equivalent to a DARE with  $\lambda = 0$ . We train 5 networks in each ensemble and repeat the experiments 5 times. The AUROC metric is used, computed with the the uncertainty score defined previously for DARE and DE (MSE) and the entropy for the other methods. We observe that DARE globally improves the OOD detection. For instance, in Fashion-MNSIT, we observe an improvement of 8 points on CIFAR10 and 34 points on MNIST, with a lost of only 2 points of in-distribution accuracy.

Methods	SVHN	CIFAR10		Accuracy	Fashion-MNIST		
		CIFAR10	CIFAR100		CIFAR10	MNIST	Accuracy
DE (NLL)	90.9 (0.4)	86.4 (0.2)	<b>91.8 (0.1)</b>		89.7 (0.9)	62.7 (6.2)	<b>89.2 (0.2)</b>
AnchorNet	91.0 (0.3)	<b>86.5 (0.2)</b>	<b>91.8 (0.0)</b>		88.8 (1.1)	68.7 (6.2)	89.1 (0.2)
MOD	91.3 (0.3)	86.3 (0.3)	91.7 (0.1)		89.4 (1.7)	60.8 (2.7)	88.7 (0.4)
NegCorr	91.3 (0.4)	86.3 (0.4)	91.7 (0.1)		91.5 (0.8)	68.9 (4.5)	86.1 (0.6)
RDE	91.2 (0.5)	86.4 (0.3)	<b>91.8 (0.1)</b>		90.1 (0.9)	70.9 (5.8)	89.1 (0.3)
DE (MSE)	85.9 (1.2)	77.7 (0.8)	91.7 (0.1)		96.5 (0.5)	93.0 (5.3)	88.6 (0.1)
DARE	<b>92.6 (0.7)</b>	82.7 (0.5)	<b>91.8 (0.1)</b>		<b>97.7 (0.5)</b>	<b>97.4 (1.3)</b>	87.2 (0.2)

Table 3: **OOD detection results.** AUROC scores and ID accuracy are reported

## 7 Limitations and Perspectives

For now, the efficiency of DARE is limited to fully-connected neural network with piece-wise linear activation. We have seen, however, that DARE can benefit to a final fully-connected network head placed on top of deep features obtained with convolutions. Thanks to the practical aspect of DARE, the method can be combined to other deep ensemble or OOD detection methods. One can use a specific training process and then apply DARE afterwards to increase diversity. Future work will consider a Bayesian version of DARE by adding gaussian noise with increasing variance to the weights of pretrained networks.

## References

1. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* **76**, 243–297 (2021)
2. Boyd, S., Vandenberghe, L., Foybusovich, L.: Convex optimization. *IEEE Transactions on Automatic Control* **51**(11), 1859–1859 (2006)
3. Brown, G., Wyatt, J.L., Tino, P., Bengio, Y.: Managing diversity in regression ensembles. *Journal of machine learning research* **6**(9) (2005)
4. Choi, J., Chun, D., Kim, H., Lee, H.J.: Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 502–511 (2019)
5. D’Angelo, F., Fortuin, V.: Repulsive deep ensembles are bayesian. *Advances in Neural Information Processing Systems* **34** (2021)
6. D’Angelo, F., Henning, C.: Uncertainty-based out-of-distribution detection requires suitable function space priors. *arXiv preprint arXiv:2110.06020* (2021)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255. Ieee (2009)
8. Dua, D., Graff, C.: UCI machine learning repository (2017), <http://archive.ics.uci.edu/ml>
9. Feng, D., Rosenbaum, L., Dietmayer, K.: Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. pp. 3266–3273. IEEE (2018)
10. Foong, A.Y., Li, Y., Hernández-Lobato, J.M., Turner, R.E.: ‘in-between’ uncertainty in bayesian neural networks. *arXiv preprint arXiv:1906.11537* (2019)
11. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: *international conference on machine learning*. pp. 1050–1059. PMLR (2016)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
13. Jain, S.D., Grauman, K.: Active image segmentation propagation. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2864–2873 (2016)



14. Jain, S., Liu, G., Mueller, J., Gifford, D.: Maximizing overall diversity for improved uncertainty estimates in deep ensembles. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 34, pp. 4264–4271 (2020)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015)
16. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning* **51**(2), 181–207 (2003)
17. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* **30** (2017)
18. Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., Batra, D.: Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314* (2015)
19. Liu, J.Z., Padhy, S., Ren, J., Lin, Z., Wen, Y., Jerfel, G., Nado, Z., Snoek, J., Tran, D., Lakshminarayanan, B.: A simple approach to improve single-model deep uncertainty via distance-awareness. *arXiv preprint arXiv:2205.00403* (2022)
20. Liu, Y., Pagliardini, M., Chavdarova, T., Stich, S.U.: The peril of popular deep learning uncertainty estimation methods. *arXiv preprint arXiv:2112.05000* (2021)
21. Liu, Y., Yao, X.: Ensemble learning via negative correlation. *Neural networks* **12**(10), 1399–1404 (1999)
22. Mackay, D.J.C.: *Bayesian methods for adaptive models*. Ph.D. thesis, California Institute of Technology (1992)
23. Mamalet, F., Jenn, E., Flandin, G., Delseny, H., Gabreau, C., Gauffriau, A., Beaudouin, B., Ponsolle, L., Alecu, L., Bonnin, H., et al.: *White Paper Machine Learning in Certified Systems*. Ph.D. thesis, IRT Saint Exupéry; ANITI (2021)
24. de Mathelin, A., Deheeger, F., Mougeot, M., Vayatis, N.: Handling distribution shift in tire design. In: *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications* (2021)
25. de Mathelin, A., Richard, G., Deheeger, F., Mougeot, M., Vayatis, N.: Adversarial weighting for domain adaptation in regression. *arXiv preprint arXiv:2006.08251* (2020)
26. Mehrtens, H.A., González, C., Mukhopadhyay, A.: Improving robustness and calibration in ensembles with diversity regularization. *arXiv preprint arXiv:2201.10908* (2022)
27. Neal, R.M.: *Bayesian learning for neural networks*, vol. 118. Springer Science & Business Media (2012)
28. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.: Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* **32** (2019)
29. Pagliardini, M., Jaggi, M., Fleuret, F., Karimireddy, S.P.: Agree to disagree: Diversity through disagreement for better transferability. *arXiv preprint arXiv:2202.04414* (2022)
30. Pearce, T., Zaki, M., Brintrup, A., Anastassacos, N., Neely, A.: Uncertainty in neural networks: Bayesian ensembling. *stat* **1050**, 12 (2018)
31. Ramé, A., Cord, M.: Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. In: *ICLR 2021-9th International Conference on Learning Representations* (2021)



32. Romano, Y., Patterson, E., Candes, E.: Conformalized quantile regression. *Advances in neural information processing systems* **32** (2019)
33. Ross, A., Pan, W., Celi, L., Doshi-Velez, F.: Ensembles of locally independent prediction models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 34, pp. 5527–5536 (2020)
34. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: *Proceedings of the 11th European Conference on Computer Vision: Part IV*. p. 213–226. ECCV’10, Springer-Verlag, Berlin, Heidelberg (2010)
35. Scalia, G., Grambow, C.A., Pernici, B., Li, Y.P., Green, W.H.: Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of chemical information and modeling* **60**(6), 2697–2717 (2020)
36. Segonne, P., Zainchkovskyy, Y., Hauberg, S.: Robust uncertainty estimates with out-of-distribution pseudo-inputs training. *arXiv preprint arXiv:2201.05890* (2022)
37. Shui, C., Mozafari, A.S., Marek, J., Hedhli, I., Gagné, C.: Diversity regularization in deep ensembles. *arXiv preprint arXiv:1802.07881* (2018)
38. Sinha, S., Bharadhwaj, H., Goyal, A., Larochelle, H., Garg, A., Shkurti, F.: Dibs: Diversity inducing information bottleneck in model ensembles. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 35, pp. 9666–9674 (2021)
39. Tifrea, A., Stavarache, E.P., Yang, F.: Semi-supervised novelty detection using ensembles with regularized disagreement. In: *The 38th Conference on Uncertainty in Artificial Intelligence* (2022)
40. Ulmer, D., Meijerink, L., Cinà, G.: Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In: *Machine Learning for Health*. pp. 341–354. PMLR (2020)
41. Van Amersfoort, J., Smith, L., Teh, Y.W., Gal, Y.: Uncertainty estimation using a single deep deterministic neural network. In: *International conference on machine learning*. pp. 9690–9700. PMLR (2020)
42. Wen, Y., Tran, D., Ba, J.: Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715* (2020)
43. Wenzel, F., Snoek, J., Tran, D., Jenatton, R.: Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems* **33**, 6514–6527 (2020)
44. Xu, Y., Du, B., Zhang, L., Zhang, Q., Wang, G., Zhang, L.: Self-ensembling attention networks: Addressing domain shift for semantic segmentation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 33, pp. 5581–5588 (2019)
45. Zaidi, S., Zela, A., Elsken, T., Holmes, C.C., Hutter, F., Teh, Y.: Neural ensemble search for uncertainty estimation and dataset shift. *Advances in Neural Information Processing Systems* **34**, 7898–7911 (2021)
46. Zhang, D., Chen, S., Zhou, Z.H., Yang, Q.: Constraint projections for ensemble learning. In: *AAAI*. pp. 758–763 (2008)
47. Zhang, S., Wu, G., Costeira, J.P., Moura, J.M.: Understanding traffic density from large-scale web camera data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5898–5907 (2017)
48. Zhang, S., Liu, M., Yan, J.: The diversified ensemble neural network. *Advances in Neural Information Processing Systems* **33**, 16001–16011 (2020)
49. Zhao, H., Zhang, S., Wu, G., Moura, J.M.F., Costeira, J.P., Gordon, G.J.: Adversarial multiple source domain adaptation. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 31, pp. 8559–8570. Curran Associates, Inc. (2018)

## A Proof of Theorem 1

We first remind the assumptions and notations used for the Theorem:

**Notations:**

- $X \in \mathbb{R}^{n \times p}$  and  $y \in \mathbb{R}^n$  are respectively the input and output data which compose the training set  $\mathcal{S}$ .
- $\mathcal{H} = \{x \rightarrow x^T \theta; \theta \in \mathbb{R}^p\}$  is the ensemble of linear networks.
- $\mathcal{L}_{\mathcal{S}}(h_{\theta}) = \frac{1}{n} \|X\theta - y\|_2^2$  is the mean squared error.
- $s^2 = (s_1^2, \dots, s_p^2) \in \mathbb{R}_+^p$  is the diagonal of the matrix  $\frac{1}{n} X^T X$ .
- $\Theta_{\sigma^2}$  is the weights distribution of the anti-regularized ensemble  $\mathcal{H}_{\tau} \subset \mathcal{H}$ , with  $\sigma^2$  the weights variance.

**Assumptions:**

$$\exists \theta^* \in \mathbb{R}^p; X\theta^* = y \quad (10)$$

For any  $h_{\theta} \in \mathcal{H}_{\tau}$ :

$$\theta \sim \Theta_{\sigma^2}; \quad \mathbb{E}[\theta] = \theta^*, \quad \text{Cov}(\theta) = \text{diag}(\sigma^2) \quad (11)$$

$$\mathbb{E}[\mathcal{L}_{\mathcal{S}}(h_{\theta})] \leq \delta \tau \quad (12)$$

Where  $\delta > 0$  and  $\text{diag}(\sigma^2)$  is the diagonal matrix of values  $\sigma^2 \in \mathbb{R}_+^p$  verifying:

$$\sigma^2 = \underset{\sigma^2 = (\sigma_1^2, \dots, \sigma_p^2)}{\text{argmax}} \sum_{k=1}^p \log(\theta_k^{*2} + \sigma_k^2) \quad (13)$$

**Theorem A.1.** *There exist a constant  $C > 0$  such that for any  $k \in [1, p]$ , the variance of the  $k^{\text{th}}$  weight component is expressed as follows:*

$$\sigma_k^2 = \max \left[ \frac{C \delta \tau}{s_k^2} - \theta_k^{*2}, 0 \right] \quad (14)$$

*Proof.* Let's introduce the variable  $z \sim \mathcal{N}(0, \text{diag}(\sigma^2))$ , verifying:  $\theta = \theta^* + z$ . From Assumption 12, we derive that:

$$\begin{aligned} \mathcal{L}_{\mathcal{S}}(h_{\theta}) &= \frac{1}{n} \|X\theta - y\|_2^2 \\ &= \frac{1}{n} \|X(\theta^* + z) - y\|_2^2 \\ &= \frac{1}{n} \|X\theta^* - y + Xz\|_2^2 \\ &= \frac{1}{n} \|Xz\|_2^2 \quad (\text{by definition of } \theta^*) \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^p \sum_{j=1}^p X_{ik} X_{ij} z_k z_j \end{aligned} \quad (15)$$

Thus, we have:

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_S(h_\theta)] &= \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^p \sum_{j=1}^p X_{ik} X_{ij} \mathbb{E}[z_k z_j] \\
&= \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^p X_{ik}^2 \sigma_k^2 \quad (\text{by definition of } z) \\
&= \sum_{k=1}^p \left( \frac{1}{n} \sum_{i=1}^n X_{ik}^2 \right) \sigma_k^2 \\
&= \sum_{k=1}^p s_k^2 \sigma_k^2
\end{aligned} \tag{16}$$

Combining this results with Assumption 13, we show that  $\sigma^2$  verifies:

$$\begin{aligned}
&\underset{\sigma^2 \in \mathbb{R}_+^p}{\text{maximize}} \quad \sum_{k=1}^p \log(\sigma_k^2 + \theta_k^{*2}) \\
&\text{subject to} \quad \sum_{k=1}^p s_k^2 \sigma_k^2 \leq \delta \tau
\end{aligned} \tag{17}$$

This expression is a "water-filling" problem (cf [2] Example 5.2) with weighted constraint. The inequality constraint  $\sum_{k=1}^p s_k^2 \sigma_k^2 \leq \delta \tau$  can be written as an equality constraint, as any increase of  $\sigma_k^2$  induces an increase of the objective function.

To solve this problem, we introduce the Lagrange multipliers  $\mu \in \mathbb{R}_+^p$  for the constraints  $\sigma^2 \geq 0$  and the multiplier  $\alpha \in \mathbb{R}$  for the constraint  $\sum_{k=1}^p s_k^2 \sigma_k^2 = \delta \tau$ . By considering the Lagrangian as a function of  $\sigma^2$ , the KKT conditions are then written:

$$\frac{-1}{\theta_k^{*2} + \sigma_k^2} + \alpha s_k^2 - \mu_k = 0 \quad \forall k \in [1, p] \tag{18}$$

$$\sigma_k^2 \geq 0, \mu_k \geq 0, \mu_k \sigma_k^2 = 0 \quad \forall k \in [1, p] \quad \text{and} \quad \sum_{k=1}^p s_k^2 \sigma_k^2 = \delta \tau \tag{19}$$

Leading to:

$$\sigma_k^2 \left( \frac{-1}{\theta_k^{*2} + \sigma_k^2} + \alpha s_k^2 \right) = 0 \quad \forall k \in [1, p] \tag{20}$$

$$\sigma_k^2 \geq 0, \alpha s_k^2 \geq \frac{1}{\theta_k^{*2} + \sigma_k^2}, \quad \forall k \in [1, p] \quad \text{and} \quad \sum_{k=1}^p s_k^2 \sigma_k^2 = \delta \tau \tag{21}$$

Then, for any  $k \in [1, p]$ , we have:

$$\sigma_k^2 = 0 \quad \text{or} \quad \sigma^2 = \frac{1}{\alpha s_k^2} - \theta_k^{*2} \tag{22}$$

As  $\sigma_k^2 \geq 0$ , we deduce that:

$$\sigma_k^2 = \max \left[ \frac{1}{\alpha s_k^2} - \theta_k^{*2}, 0 \right] \quad (23)$$

With  $\alpha$  verifying:

$$\begin{aligned} \sum_{k=1}^p s_k^2 \max \left[ \frac{1}{\alpha s_k^2} - \theta_k^{*2}, 0 \right] &= \delta \tau \\ \sum_{k=1}^p \max \left[ \frac{1}{\alpha \delta \tau} - \frac{s_k^2 \theta_k^{*2}}{\delta \tau}, 0 \right] &= 1 \end{aligned} \quad (24)$$

By defining  $C = \frac{1}{\alpha \delta \tau}$ , we have:

$$\sum_{k=1}^p \max \left[ C - \frac{s_k^2 \theta_k^{*2}}{\delta \tau}, 0 \right] = 1 \quad (25)$$

Which imposes  $C > 0$ .

We conclude then, that there exists  $C > 0$  such that, for any  $k \in [1, p]$ :

$$\sigma_k^2 = \max \left[ \frac{C \delta \tau}{s_k^2} - \theta_k^{*2}, 0 \right] \quad (26)$$

## B Proof of Corollary 1

**Corollary B.1.** *Let  $\mathcal{H}_\tau$  be the large weights ensemble defined by Assumptions 11, 12, 13 and  $x \in \mathbb{R}^p$  an input data, the variance of prediction for  $h_\theta \in \mathcal{H}_\tau$  is:*

$$\text{Var}_{\theta \sim \Theta_{\sigma^2}} [h_\theta(x)] = \sum_{k=1}^p x_k^2 \max \left[ \frac{C \delta \tau}{s_k^2} - \theta_k^{*2}, 0 \right] \quad (27)$$

*Proof.* Let's consider  $x \in \mathbb{R}^p$  and  $h_\theta \in \mathcal{H}_\tau$ :

$$\begin{aligned} \text{Var}_{\theta \sim \Theta_{\sigma^2}} [h_\theta(x)] &= \text{Var}_{\theta \sim \Theta_{\sigma^2}} \left[ \sum_{k=1}^p x_k \theta_k \right] \\ &= \sum_{k=1}^p x_k^2 \sigma_k^2 \quad (\text{by Assumption 11}) \end{aligned} \quad (28)$$

## C Proof of Theorem 2

**Notations:**

- $X \in \mathbb{R}^{n \times p}$  and  $y \in \mathbb{R}^n$  are respectively the input and output data which compose the training set  $\mathcal{S}$ .

- $\mathcal{H}_\tau$  is an anti-regularized ensemble of fully-connected deep neural networks with  $L$  hidden layers and network weights distributed according to  $\Theta_{\sigma^2}$  in  $\mathbb{R}^d$ .
- For any  $\ell \in [0, L+1]$ ,  $p_\ell$  is the number of neurons of the  $\ell^{th}$  layer.
- $g : \mathbb{R} \rightarrow \mathbb{R}$  is the activation function applied at each hidden layer.
- For every  $\ell \in [0, L]$ , we denote  $\phi_{(\ell)}(X) \in \mathbb{R}^{n \times p}$  the representation of the  $\ell^{th}$  layer, with  $\ell = 0$  the input layer.
- For any  $j \in [1, p_{\ell+1}]$ , we denote  $\theta_{(\ell,j)} \in \mathbb{R}^p$  the network weights between the layer  $\ell$  and the  $j^{th}$  component of the layer  $\ell + 1$ .
- For any  $\ell \in [0, L]$ ,  $\phi_{(\ell)}^*(X) = \mathbb{E}[\phi_{(\ell)}(X)]$  and  $s_{(\ell)}^2$  is the diagonal of the matrix  $\frac{1}{n} \phi_{(\ell)}^*(X)^T \phi_{(\ell)}^*(X)$ .

### Assumptions:

For any random variable  $Z$  in  $\mathbb{R}$  there exists  $\alpha > 0$  such that:

$$\text{Var}[g(Z)] > \alpha \text{Var}[Z]. \quad (29)$$

There is  $\delta > 0$  such that, for any  $\ell \in [0, L]$  and any  $j \in [1, p_{\ell+1}]$ ,  $\phi_{(\ell)}(X)$  and  $\theta_{(\ell,j)}$  verify Assumptions 11, 12, 13 with

$$\theta \equiv \theta_{(\ell,j)}, X \equiv \phi_{(\ell)}^*(X), y \equiv \phi_{(\ell)}^*(X) \theta_{(\ell,j)}^* \text{ and } \theta^* \equiv \theta_{(\ell,j)}^* \equiv \mathbb{E}[\theta_{(\ell,j)}] \quad (30)$$

**Theorem C.1.** *There exists a constant  $\gamma > 0$  such that, for any  $h_\theta \in \mathcal{H}_\tau$  and  $x \in \mathbb{R}^p$ , the prediction variance verifies:*

$$\text{Var}_{\theta \sim \Theta_{\sigma^2}} [h_\theta(x)] \geq \gamma \sum_{\ell=0}^L \langle \phi_{(\ell)}^*(x)^2, \bar{\sigma}_{(\ell)}^2 \rangle$$

With,  $\bar{\sigma}_{(\ell)}^2 = \sum_{j=1}^{p_{\ell+1}} \max \left[ \frac{C_{(\ell,j)}}{s_{(\ell)}^2} - \theta_{(\ell,j)}^*{}^2, 0 \right]; C_{(\ell,j)} \in \mathbb{R}_+^*$

*Proof.* Let's consider  $x \in \mathbb{R}^p$  and  $h_\theta \in \mathcal{H}_\tau$ :

Considering Assumption 30, we apply Theorem A.1 to  $\phi_{(\ell)}(x)_j$  for any  $\ell \in [0, L]$  and any  $j \in [1, p_{\ell+1}]$ , such that:

$$\text{Var}[\theta_{(\ell,j)}] = \sigma_{(\ell,j)}^2 = \max \left[ \frac{C_{(\ell,j)}}{s_{(\ell)}^2} - \theta_{(\ell,j)}^*{}^2, 0 \right] \quad (31)$$

With  $C_{(\ell,j)} \in \mathbb{R}_+^*$ . Notice that the max function is applied element-wise.

For any  $\ell \in [0, L-1]$  and any  $j \in [1, p_{\ell+1}]$ , the following inequalities are verified:

$$\begin{aligned}
\text{Var} [\phi_{(\ell+1)}(x)_j] &= \text{Var} [g(\phi_{(\ell)}(x)\theta_{(\ell,j)})] \\
&\geq \alpha \text{Var} [\phi_{(\ell)}(x)\theta_{(\ell,j)}] \quad (\text{by Assumption 29}) \\
&\geq \alpha \text{Var} \left[ \sum_{k=1}^{p_\ell} \phi_{(\ell)}(x)_k \theta_{(\ell,j,k)} \right] \\
&\geq \alpha \sum_{k=1}^{p_\ell} \text{Var} [\phi_{(\ell)}(x)_k \theta_{(\ell,j,k)}] \quad (\phi_{(\ell)}(x)_k \theta_{(\ell,j,k)} \text{ independents}) \\
&\geq \alpha \sum_{k=1}^{p_\ell} \text{Var} [\phi_{(\ell)}(x)_k] \text{Var} [\theta_{(\ell,j,k)}] + \mathbb{E} [\phi_{(\ell)}(x)_k]^2 \text{Var} [\theta_{(\ell,j,k)}] \\
&\quad + \mathbb{E} [\theta_{(\ell,j,k)}]^2 \text{Var} [\phi_{(\ell)}(x)_k] \\
&\geq \alpha \sum_{k=1}^{p_\ell} \text{Var} [\phi_{(\ell)}(x)_k] \left( \sigma_{(\ell,j,k)}^2 + \theta_{(\ell,j,k)}^{*2} \right) + \phi_{(\ell)}^*(x)_k^2 \sigma_{(\ell,j,k)}^2 \\
&\geq \alpha \langle \phi_{(\ell)}^*(x)^2, \sigma_{(\ell,j)}^2 \rangle + \alpha \sum_{k=1}^{p_\ell} \text{Var} [\phi_{(\ell)}(x)_k] \left( \sigma_{(\ell,j,k)}^2 + \theta_{(\ell,j,k)}^{*2} \right) \\
&\geq \alpha \langle \phi_{(\ell)}^*(x)^2, \sigma_{(\ell,j)}^2 \rangle + \alpha \gamma_{(\ell,j)} \sum_{k=1}^{p_\ell} \text{Var} [\phi_{(\ell)}(x)_k]
\end{aligned} \tag{32}$$

With  $\gamma_{(\ell,j)} \geq \min_{k \in [1, p_\ell]} \left( \sigma_{(\ell,j,k)}^2 + \theta_{(\ell,j,k)}^{*2} \right) = \min_{k \in [1, p_\ell]} \max \left[ \frac{C_{(\ell,j)}}{s_{(\ell)}^2}, \theta_{(\ell,j)}^{*2} \right] > 0$

The fourth inequality comes from the variance decomposition of the product of two independent variables:  $X \perp\!\!\!\perp Y \implies \text{Var}[XY] = \text{Var}[X]\text{Var}[Y] + \text{Var}[X]\mathbb{E}[Y]^2 + \text{Var}[Y]\mathbb{E}[X]^2$ .

The last inequality in (32) provides an expression of recurrence for  $\text{Var} [\phi_{(\ell)}(x)_j]$ . We notice that,  $\text{Var} [\phi_{(0)}(x)_j] = \text{Var} [x_j] = 0$  for any  $j \in [1, p_0]$ . For the last layer, we have:

$$\text{Var} [h_\theta(x)] = \text{Var} [\phi_{(L)}(x)\theta_{(L,1)}] \geq \langle \phi_{(L)}^*(x)^2, \sigma_{(L,1)}^2 \rangle + \gamma_{(L,1)} \sum_{k=1}^{p_L} \text{Var} [\phi_{(L)}(x)_k] \tag{33}$$

We pose  $\tilde{\gamma} = \min_{\ell \in [0, L]} \min_{j \in [1, p_{\ell+1}]} \gamma_{(\ell,j)}$  and  $\gamma = \min[1, \tilde{\gamma}, \tilde{\gamma}\alpha^L] > 0$ . We then deduce, by considering the recurrence expression, that:

$$\text{Var} [h_\theta(x)] \geq \gamma \sum_{\ell=0}^L \langle \phi_{(\ell)}^*(x)^2, \bar{\sigma}_{(\ell)}^2 \rangle \tag{34}$$

With,  $\bar{\sigma}_{(\ell)}^2 = \sum_{j=1}^{p_{\ell+1}} \sigma_{(\ell,j)}^2 = \sum_{j=1}^{p_{\ell+1}} \max \left[ \frac{C_{(\ell,j)}}{s_{(\ell)}^2} - \theta_{(\ell,j)}^{*2}, 0 \right]$

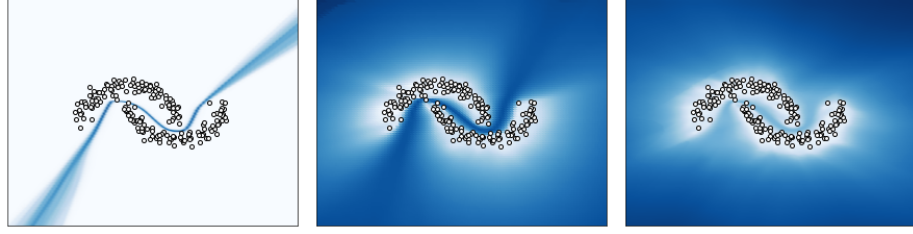
## D Synthetic Experiments

We consider the two synthetic experiments : Two moons classification and 1D regression :

- **Two Moons Classification** : 200 random data points are generated from the two moons generator<sup>4</sup> to form the training set. We consider the three uncertainty quantification methods: Deep Ensemble (NLL), Deep Ensemble (MSE) and Deep Anti-Regularized Ensemble (DARE). The first method implements a softmax activation function at the end layer and use the multi-classification NLL as loss function. The two others use a linear activation at the end layer and the mean squared error as loss function. A fully-connected network of three layers with 100 neurons each and ReLU activations is used as base network. Each ensemble is composed of 20 networks. The Adam [15] optimization is used with learning rate 0.001, batch size 32 and 500 epochs. For DARE, the threshold is set to  $\tau = 0.001$ . To compute the uncertainty scores, the entropy metric [17] is used for Deep Ensemble (NLL) while Deep Ensemble (MSE) and DARE use the uncertainty metric for classification with MSE loss function defined in Appendix F. We report the uncertainty map produced by each method in Figure 6. We also report the results for the OOD detection task in Figure 7. To produce this last results, we sample 50 random points from the two moons generator which acts as validation data. We compute the uncertainty scores on the validation set and use the 95% percentiles as threshold for OOD detection. Any data points below this threshold are considered as in-distribution data (in white) and the others as OOD (in dark blue).
- **1D Regression** : We reproduce the synthetic univariate Regression experiment from [13]. We consider the three methods: Deep Ensemble (NLL), Deep Ensemble (MSE) DARE. We use the gaussian NLL loss for regression defined in [17] for Deep Ensemble (NLL) and DARE. The mean squared error is used as loss function for Deep Ensemble (MSE). The threshold for DARE is set to  $\tau = 0.1$ . For any data  $x \in \mathbb{R}$ , each method, return two values  $\mu_x$  and  $\sigma_x$  (cf [17]). We report in Figure 8 the confidence intervals  $[\mu_x - 2\sigma_x, \mu_x + 2\sigma_x]$  (in light blue) for any  $x$  in the input range.

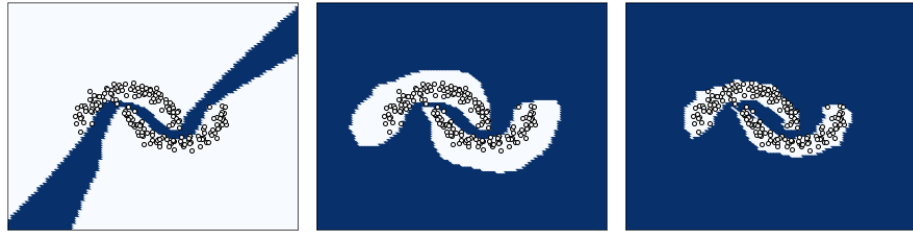
---

<sup>4</sup> [https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make\\_moons.html](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html)



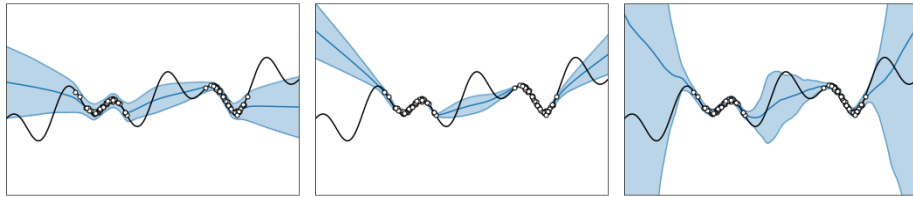
(a) Deep Ensemble (NLL) (b) Deep Ensemble (MSE) (c) DARE

Fig. 6: **Two-moons uncertainty estimation** Darker areas correspond to higher predicted uncertainty.



(a) Deep Ensemble (NLL) (b) Deep Ensemble (MSE) (c) DARE

Fig. 7: **OOD detection**. Data classified as OOD are in dark blue OOD while the ones classified as in-distribution are in white.



(a) Deep Ensemble (NLL) (b) Deep Ensemble (MSE) (c) DARE

Fig. 8: **1D Regression uncertainty estimation**. the confidence intervals for  $\pm 2\sigma$  are given in light blue.



## E CityCam Experiments

We consider the three following experiments:

- **Weather-shift**: in this experiment, we select the images of the three cameras n°164, 166 and 572 and use the images recorded during February the 23<sup>th</sup>. During this particular day, the weather significantly change between the beginning and the end of the day. Thus, to evaluate the robustness of the method to a change in the weather condition, we split the dataset in two subsets: we consider the images recorded before 2 pm as in-distribution and the others as out-of-distribution. we observe a shift in the weather condition around 4 pm which produces a visual shift in the images. After 4 pm, the rain starts to fall and progressively damages the visual rendering of the images due to the reverberation. This is particularly the case for camera n°164, as the water drops have landed on the camera and blur the images. Each dataset is composed of around 2500 images.
- **Camera-shift**: this experiment uses the images from 10 cameras in the CityCam dataset. For each round, 5 cameras are randomly selected as in-distribution while the 5 remaining cameras are considered as out-of-distribution. In average, each set is composed of around 20000 images.
- **Bigbus-shift**: The CityCam dataset contains images marked as "big-bus" referring to the fact that a bus appears and masks an significant part of the image [47]. We then select the 5 cameras for which some images are marked as "big-bus" and use this marker to split the dataset between in-distribution and out-of-distribution samples. The in-distribution set is composed of around 17000 images while the other set contains around 1000 images.

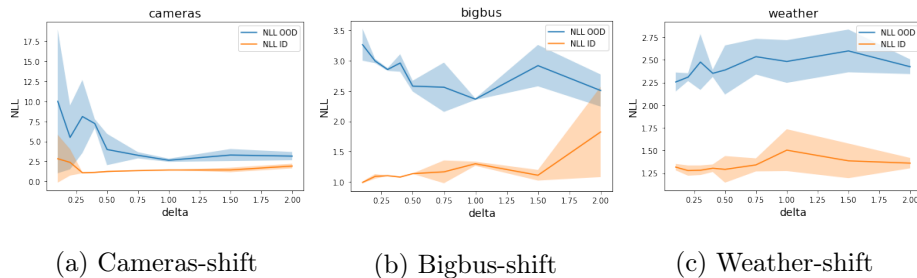


Fig. 9: **Ablation study for the  $\delta$  parameter.** The DARE NLL is reported for different values of  $\delta$  with  $\tau = 1 + \delta$ . We globally observe a decrease of the OOD NLL and a slight increase of the in-distribution NLL when  $\delta$  increases.

## F OOD Detection Experiments

For the OOD detection experiments, we follow the setup of [5]. For the Fashion-MNIST experiment, a three layers fully-connected network is used as base network,

with ReLU activations. We use the Adam optimizer [15] with learning rate 0.001 and a batch size of 128. The maximal number of epochs is set to 50. For the CIFAR10 experiment, the same base network is used on top of a pretrained ResNet32. We consider the same optimization parameters.

For DARE and Deep Ensemble (MSE), the uncertainty score is computed as follows:

$$\text{OOD-score}(x) = \frac{1}{M} \sum_{k=1}^M \|h_k(x) - \hat{y}_k\|_2^2 + \frac{1}{M} \sum_{k=1}^M \left\| h_k(x) - \frac{1}{M} \sum_{i=1}^M h_i(x) \right\|_2^2 \quad (35)$$

With  $M$  the number of network in the ensemble,  $h_k$  the network members and  $\hat{y}_k$  the one-hot-encoded vector of the predicted class multiplied by the number of classes.