



**HAL**  
open science

## **n-MeRCI: A new Metric to Evaluate the Correlation Between Predictive Uncertainty and True Error**

Michel Moukari, Loïc Simon, Sylvaine Picard, Frédéric Jurie

► **To cite this version:**

Michel Moukari, Loïc Simon, Sylvaine Picard, Frédéric Jurie. n-MeRCI: A new Metric to Evaluate the Correlation Between Predictive Uncertainty and True Error. IEEE/RJS International Conference on Intelligent Robots and Systems (IROS), Nov 2019, Macao, China. hal-02242792

**HAL Id: hal-02242792**

**<https://hal.science/hal-02242792v1>**

Submitted on 19 Aug 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# n-MeRCI: A new Metric to Evaluate the Correlation Between Predictive Uncertainty and True Error

Michel Moukari<sup>1,2</sup>, Loïc Simon<sup>1</sup>, Sylvaine Picard<sup>2</sup> and Frédéric Jurie<sup>1</sup>

**Abstract**—As deep learning applications are becoming more and more pervasive in robotics, the question of evaluating the reliability of inferences becomes a central question in the robotics community. This domain, known as *predictive uncertainty*, has come under the scrutiny of research groups developing Bayesian approaches adapted to deep learning such as Monte Carlo Dropout. Unfortunately, for the time being, the real goal of predictive uncertainty has been swept under the rug. Indeed, these approaches are solely evaluated in terms of raw performance of the network prediction, while the quality of their estimated uncertainty is not assessed. Evaluating such uncertainty prediction quality is especially important in robotics, as actions shall depend on the confidence in perceived information. In this context, the main contribution of this article is to propose a novel metric that is adapted to the evaluation of relative uncertainty assessment and directly applicable to regression with deep neural networks. To experimentally validate this metric, we evaluate it on a toy dataset and then apply it to the task of monocular depth estimation.

## I. INTRODUCTION

The outcome of deep learning these past few years – capable of reaching and even sometimes surpass human performances, *e.g.* for vision tasks [1], [2] – offers novel avenues in robotics, but also raises some novel questions: as pointed out by [3], *how much trust can we put in the predictions of a deep learning system?*, *how can we estimate the uncertainty in a deep network’s predictions?* are among the central questions that have to be addressed before deep learning can safely be used in robotics.

While the robustness of deep learning algorithms to adversarial attacks has been very actively addressed in the recent literature (see *e.g.*, [4]), the question of predicting the uncertainty of predictions made by deep neural networks has received much less attention. Yet, for countless applications it is a matter of critical importance to know whether the estimate output of the algorithm is reliable or not, especially in robotics. In addition, the rare papers which addressed related questions do not evaluate the quality of the predicted uncertainty (*e.g.*, [5] in the context of depth inference from RGB images).

In this paper we take the example of *depth estimation from single monocular images* [6], [5], which is useful for many tasks in robotics (*e.g.*, obstacle avoidance, vehicle guidance, *etc.*). For such a task, deep convolutional neural networks are

trained to regress a depth map from an input RGB image. If one wants to use such algorithms as measurement tools in robotics applications, it is required to have, in addition to the predicted depth, an uncertainty map providing an estimate of the accuracy for each pixel. Uncertainty prediction is indeed a very valuable information and also depends on the content of the images. For example, it is likely that depth estimation on textureless regions will be less accurate, or, that dark images lead to less accurate estimations than well-exposed images.

In the literature of deep learning, some recent techniques have emerged to estimate uncertainties along with the network’s predictions [7], [8], [9]. However, these predictive uncertainty techniques are evaluated either by their influence on the predictions performance, or qualitatively by visually checking the uncertainty maps. This is neither appropriate nor statistically accurate.

In the literature, the most common way to evaluate the predicted confidences is through a sparsification process (see *e.g.*, [10]), which is only measuring if the ranking of the confidences is correct. In domains such as forecasting, standard tools to evaluate predictive uncertainty [11], [12], [13] do not allow to evaluate the quality of the uncertainty estimate independently of the predicted values.

In this context, the main contribution of this paper is a novel evaluation metric that is well adapted to the evaluation of relative uncertainty assessment and is directly applicable to the current practice in deep learning. This metric is evaluated on a toy dataset as well as in the context of depth estimation from a single monocular image. This allows us to draw conclusions regarding several uncertainties methods including some of the most popular proposed in the literature and applied to this task.

## II. RELATED WORK

*a) Uncertainty estimates:* a common path to the estimate of uncertainties with deep networks is to transform deterministic networks, in which each parameter is a scalar, into probabilistic Bayesian models, where parameters are replaced by probability distributions. Because the inference becomes intractable, several papers rely on approximate solutions. Among the most popular techniques, we count Probabilistic BackPropagation (PBP) [14] or Monte Carlo Dropout [7]. There are also some non-Bayesian ones such as Deep ensembles [8] or the learned attenuation loss proposed by [9] to capture the aleatoric uncertainty in the data.

*b) Assessment of predictive uncertainty:* in the literature, it is usually the quality of the prediction which is

<sup>1</sup>Michel Moukari, Loïc Simon and Frédéric Jurie are with Normandie Univ, UNICAEN, ENSICAEN, CNRS, 6 boulevard du Maréchal Juin 14000 Caen, France {michel.moukari, loic.simon, frederic.jurie}@unicaen.fr

<sup>2</sup>Michel Moukari and Sylvaine Picard are with Safran, 1 rue Geneviève Aube, 78114 Magny-les-Hameaux, France {michel.moukari, sylvaine.picard}@safrangroup.com

measured, not the quality of the uncertainty. Qualitative evaluations are commonly done visually with images of the estimated uncertainties. In terms of quantitative evaluation, the most common metric is sparsification plots which measure how much the estimated uncertainty correlates with the true errors. If the data with the highest variance are removed gradually, the error should monotonically decrease *e.g.*, [10], [15]). As said before, one limitation of these metrics is that they only evaluate the rankings of the confidences.

The *Evaluating Predictive Uncertainty Challenge* [16] proposed two metrics adapted to regression tasks that are still commonly used. The normalized MSE aims at evaluating the quality of a prediction and uses the empirical variance of the test estimates to normalize the scores. This is therefore not a direct way to evaluate the predictive uncertainty. The second one, the average negative log predictive density (NLPD), directly takes into account the estimated uncertainty and penalizes both over and under-confident predictions.

A more in-depth treatment can be found in the forecasting literature. [11] draw up an overview of methods to estimate uncertainty with deterministic models such as: expert assessment, where an expert in the area rates the confidence of a method using his knowledge; model sensitivity analysis, determining how the output response of the model change w.r.t. the inputs and model's parameters; model emulation, consisting of approximating a complex model by a lower-order function, to then apply more easily a sensitivity analysis; spatiotemporal variability, using the spatiotemporal variability as an estimate of the variance; multiple models, where an ensemble of models are used to describe the same domain; and finally data-based approaches, where enough data are available to directly assess the statistical uncertainty related to the model outputs.

Differently, [12] propose to evaluate a predictive uncertainty forecast in terms of calibration and sharpness. Calibration refers to the statistical consistency between the predictive distributions and the observations; sharpness refers to the concentration of the predictive distributions. Several graphical tools are proposed to visually evaluate if the probabilistic forecasts are calibrated and sharp. More recently, [17] propose a method to recalibrate regression algorithms so that the uncertainty estimates lie in the expected confidence interval. They also propose metrics to assess both calibration and sharpness of a predictive distribution. However, they consider them separately while we were driven by the will of assessing both these properties jointly. Following the tracks of [13], we are interested in the scoring rules functions that assess the quality of a forecast, maximizing sharpness subject to calibration.

*c) Limitations of the previous methods for predictive uncertainty assessment:* First, these methods do not allow to evaluate the uncertainty prediction independently of the inferred values; this is the whole distribution which is evaluated. Second, most of these metrics imply that the prediction is a probability distribution, which is seldom true in practice. Instead, existing deep learning methods usually predict uncertainty as being a scalar representing the variance

of some predictions (*e.g.* with Monte Carlo sampling). These remarks motivate the proposed metric allowing to evaluate prediction/uncertainty independently. We call our metric the *normalized Mean Rescaled Confidence Interval* (n-MeRCI). It is based on the calculation of a scaling coefficient, for the interval centered on the prediction to encompass the ground-truth observation. It needs no prior on the distribution, allows to effectively rank different methods of estimations and is parametrized to be robust to outliers.

### III. PROPOSED METRIC

#### A. Notations and definitions

We assume a regression setting with a training set of  $N$  samples denoted as  $\mathcal{T} = \{(x_i, y_i^*)\}$  where  $x_i \in \mathbb{R}^d$  is the input vector of dimension  $d$  and  $y_i^* \in \mathbb{R}$  the scalar value to be predicted. In a standard deterministic regression framework, the regressor is a function defined as  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ , where  $f_\theta(x)$  is an estimator of the true function  $f(x)$ , and  $\hat{y} = f_\theta(x)$  is the prediction made by the model. Learning such a regressor consists in finding the parameters  $\theta$  minimizing an empirical loss defined by  $\frac{1}{N} \sum_{i=1}^N L(f_\theta(x_i), y_i^*)$  where  $L(\cdot, \cdot)$  accounts for the proximity of its two arguments (*e.g.* a squared difference). In a probabilistic framework, we predict a whole distribution  $P_y$  instead of a deterministic prediction. This distribution has a probability density function  $p_y(x)$  and a cumulative distribution  $F_y$ . In practice,  $P_y$  is often characterized by its mean value, which is usually used as the prediction  $\hat{y}$ , and by its standard deviation  $\sigma$ . Different statistical approaches to perform such kinds of predictions in deep learning shall be reviewed in section IV.

The previous notations assume the predictions to be defined through a distribution. However, most deep learning regression methods merely output a scalar value  $\hat{y}$  along with a standard deviation  $\sigma$ . The proposed method is based on such  $\sigma$  values.

#### B. Objective

We denote the true error as the difference between estimated values and actual values:  $\epsilon_i = |\hat{y}_i - y_i^*|$ . The Mean Absolute Error is defined as:  $MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i^*|$ .

Our aim is to define a metric with the following properties:

- Be scale independent and robust to outliers. Being scale independent is important as it avoids to advantage methods predicting large uncertainties. It normalizes uncertainties in terms of sharpness.
- Reflect the correlation between the true error  $\epsilon_i$  and the estimated uncertainty  $\sigma_i$ , independently of the estimated values  $y_i$ .
- Give meaningful values. Being greater or equal to 1 means the estimation  $\sigma_i$  is worse than a constant uncertainty estimator (equivalent to have no estimation of the uncertainty). Being equal to 0 means the prediction perfectly fits with the actual errors (lower bound).

#### C. Mean Rescaled Confidence Interval

Regarding the first property, inspired by the notion of calibration and with scale invariance in mind, we propose

to rescale all the  $\sigma_i$ 's by a common factor  $\lambda$  so that the rescaled uncertainties  $\tilde{\sigma}_i = \lambda\sigma_i$  are compatible with the observed errors, that is to say  $|\hat{y}_i - y_i^*| \leq \tilde{\sigma}_i$ . Robustness to outliers is achieved by relaxing this constraint such that it is verified by only a certain number of the samples. For example, one can choose to compute the MeRCI score on 95% of the data considered to be reliable (inliners). In other words, we look for the smallest factor  $\lambda^\alpha$  such that  $\alpha\%$  of the observed errors are less than their corresponding rescaled predicted uncertainty.

Considering also the second property, the Mean Rescaled Confidence Interval that we propose is an average uncertainty, performed over the  $\lambda$ -rescaled estimated uncertainties and averaged over the number of evaluated points  $N$ :

$$\text{MeRCI}^\alpha = \frac{1}{N} \sum_{i=1}^N \lambda^\alpha \sigma_i \quad (1)$$

To obtain an efficient computation of the scaling factor, we first evaluate all ratios  $\lambda_i = \frac{|\hat{y}_i - y_i^*|}{\sigma_i}$  and then extract the  $\alpha^{\text{th}}$  percentile which corresponds exactly to the desired value.

MeRCI is inherently insensitive to scaling and allows us to compare methods with different ranges of uncertainties.

#### D. Normalized Mean Rescaled Confidence Interval

Regarding the third property, a range of relevant values can easily be exposed for this metric. On one hand, the MeRCI score is always lower bounded by the corresponding score for an ideal uncertainty prediction called the *oracle*. This oracle merely predicts an uncertainty equal to the actual observed error, which yields to  $\lambda^\alpha = 1$  and a MeRCI score reducing to the Mean Absolute Error (MAE). On the other hand, any uncertainty prediction that performs worse than a *constant* uncertainty predictor (i.e. a non-informative predictor) is clearly not worth pursuing. As a result, the score obtained by the constant predictor,  $\max^\alpha(|\hat{y}_i - y_i|)$ , provides a form of upper bound.  $\max^\alpha$  is computed on the same extracted  $\alpha^{\text{th}}$  percentile, as for  $\text{MeRCI}^\alpha$ . By defining n-MeRCI as:

$$\text{n-MeRCI}^\alpha = \frac{\frac{1}{N} \sum_{i=1}^N \lambda^\alpha \sigma_i - \text{MAE}}{\max^\alpha(|\hat{y}_i - y_i|) - \text{MAE}} \quad (2)$$

we obtain the expected properties: it is scale invariant, is equal to 1 when the uncertainty predictor predicts a constant value for all data points, and is equal to 0 when the prediction is perfect (oracle). The score is therefore to be minimized. Moreover, when the uncertainty prediction is worse than using a constant uncertainty predictor, the score is greater than 1. This allows us to characterize the predictor as useless.

n-MeRCI is easy to interpret: a n-MeRCI value of 0.5 means the uncertainties are, on average, two times larger than the oracle. In its construction, the n-MeRCI score follows the track outlined by [12]. It reflects the best possible sharpness under the constraint that the uncertainty estimate is calibrated, in the sense that it is consistent with 95% of the samples. Being able to be robust to outliers is important as it avoids a few data points to strongly impact the whole measure, as shown in Section V-B. Also, the oracle is the

sharpest possible uncertainty predictor once the estimated values  $\hat{y}_i$  are fixed.

## IV. PREDICTIVE UNCERTAINTY METHODS

We consider several approaches to perform regression tasks with predictive uncertainty. First of all, we take the most used and popular methods in the literature of deep learning, among which we find Bayesian techniques. Due to its small overhead, *Monte Carlo Dropout (MCD)* [7] is one such technique. At test time, the dropout is kept activated to perform Monte Carlo sampling. The final prediction is the average of the different estimates and their standard deviation is used as a measure of uncertainty.

We also explore some popular ensemble techniques. In ensemble learning, multiple algorithms are used to obtain a predictive performance. Here, the different deterministic outputs are combined together (usually averaged) to obtain the final prediction and their standard deviation is associated to the degree of uncertainty of the prediction. A classical ensemble approach consists in training the same network several times with different initial random seeds. It is the approach employed in Deep Ensembles [8], and we refer to this method as *Multi Inits (MI)*. Another popular ensemble way to estimate uncertainty is *Bagging* [18]. It involves using several subsets of the dataset to train several networks and compute the mean/variance of the different outputs.

In addition to these standard methods, inspired by state-of-the-art, we also propose several techniques to predict uncertainty in a deep learning framework. First, we propose to consider the network at different stages of the training as being as many different networks as there are epochs, once the plateau is reached. Assuming that, we can simulate an ensemble method coming with no additional cost, with standard training procedures. However, unlike snapshot ensembles [19], we do not use a particular learning rate policy during the learning stage. Moreover, we aim to use this kind of ensemble technique seeking more to produce a meaningful uncertainty map than enhancing the overall prediction. We refer to this method as *Multi Epochs (ME)*. On a par with multiple initializations, we propose to use several architectures and refer to this ensemble technique as *Multi Networks (MN)*. Here again, we average the outputs to obtain the prediction and use their standard deviation as the measure of uncertainty. Finally, we think that because we want to measure the correlation between the uncertainty and the error, it is possible to learn a neural network to infer the confidence of a former neural network. The output of this network that learns the error will be therefore the uncertainty estimate paired with the prediction of the initial network. It is referred as *Learned Error (LE)*.

## V. EXPERIMENTAL VALIDATION ON A TOY DATASET

### A. Experimental protocol

The toy dataset corresponds to a simple one-dimensional regression task. The data consists of 20 training examples, sampled uniformly in the interval  $[-4; 4]$ . Their corresponding targets are generated as  $y = x^3 + \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, 3^2)$ .

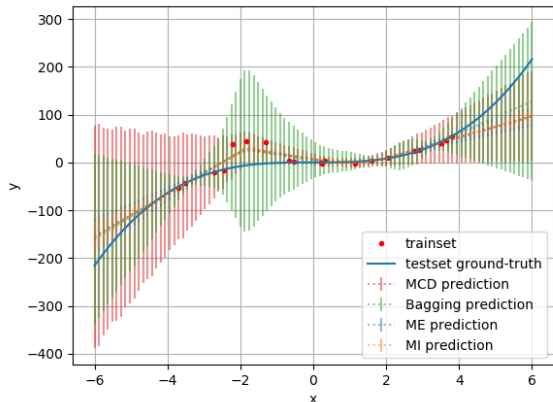


Fig. 1: The toy dataset.

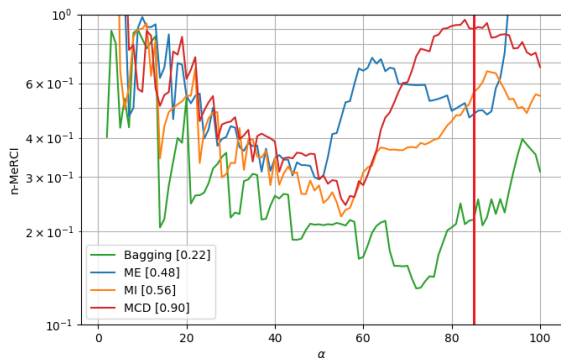


Fig. 2: n-MeRCI w.r.t.  $\alpha$  parameter. Scores in brackets are displayed for  $\alpha = 85$  which corresponds to the actual inliners rate (marked by the red line).

We add a high bias to each training point in the interval  $[-2.3; -1.3]$  to simulate some outliers, as illustrated in Figure 1. A similar dataset has already been used to qualitatively evaluate predictive uncertainty in [14] and [8].

Given  $x$ , we learn to regress  $y$  using a neural network with one hidden layer consisting of 100 neurons. We use the ReLU non-linearity and a dropout with dropping probability of 0.2 on the hidden layer. We use 4 different techniques that we presented earlier: Multi Inits, Bagging, Monte Carlo Dropout and Multi Epochs. For each one we compute 20 runs. Their average predictions and uncertainty results are also illustrated in Fig. 1. As expected, we see that the predictions are less accurate outside of the training set as well as in the outliers interval, with higher variance in these areas (displayed with  $\times 10$  scaling for better visualization).

### B. Robustness to outliers

Unlike classical metrics, n-MeRCI can be tailored towards obtaining more or less tolerance to erroneous data by tuning the percentile value. To gain better understanding, we plot the evolution of n-MeRCI with respect to the chosen percentile in Fig. 2. In this toy dataset, the ratio of corrupted data is

known in advance and is equal to 15% (3 data points).

As expected, the evolution of the n-MeRCI score decreases as the number of inliners increases, to some extent. Indeed, taking into account more correctly predicted points improves the score. However, although there is 85% of outliers, we observe a deterioration around  $\alpha = 60\%$ . This is due to the fact that most predictors present high variances around the outliers area, for correct estimations (see Fig. 1). This encompasses around 40% of the points.

In practice, the percentile should be chosen according to the number of supposed outliers in the dataset. Therefore in later experiments we use the 95th percentile, giving some latitude to handle up to 5% of possible outliers since we know that ground-truth is not perfect for the dataset we use.

### C. Comparisons of the uncertainty prediction methods on the toy dataset

Finally, we evaluate the n-MeRCI scores obtained for each uncertainty estimation technique. We expect Bagging to be the top technique because it benefits from external sources of data. We also expect Multi Epochs to be one of the worst. Indeed, the network is likely to converge to a local minimum and therefore there won't be much informative variance across epochs.

Although very noisy due to the number of data points, the plot confirms our intuitions. Indeed, Fig. 2 show the actual performances and looking at the specific value of  $\alpha=0.85$ , we see that: Bagging is best (n-MeRCI=0.22), Multi Epochs is worse (n-MeRCI=0.48) and the worst is Monte Carlo Dropout (n-MeRCI=0.9). We also note that, on overall, predictions are far from being perfect. It is understandable as training points cover only a small part of the distribution, making estimation more difficult.

## VI. PREDICTIVE UNCERTAINTIES FOR MONOCULAR DEPTH ESTIMATION

### A. Experimental protocol

This section presents experiments on depth estimation from monocular images on the NYU-Depth v2 indoor dataset [20]. We use the set of 16K images proposed by [21]. Also, we use their hourglass-type architecture, which backbone is a dilated ResNet200 pre-trained on ImageNet for the encoder part, and 4 up-projection modules followed by a final convolution for the decoder. We then add dropout with probability 0.2 after the first 3 up-projection modules, in order to perform Monte-Carlo sampling at test time for the corresponding experiment.

### B. Qualitative evaluation

We first show a qualitative evaluation obtained for the monocular depth estimation. In Fig. 3, we show an example of a test image 3(a), the prediction of one predictor 3(b) and finally the absolute error between this prediction and the ground truth 3(c); all methods predictions are close to each other so we choose to only display one for visualization. According to the absolute error map, a few demarcated regions

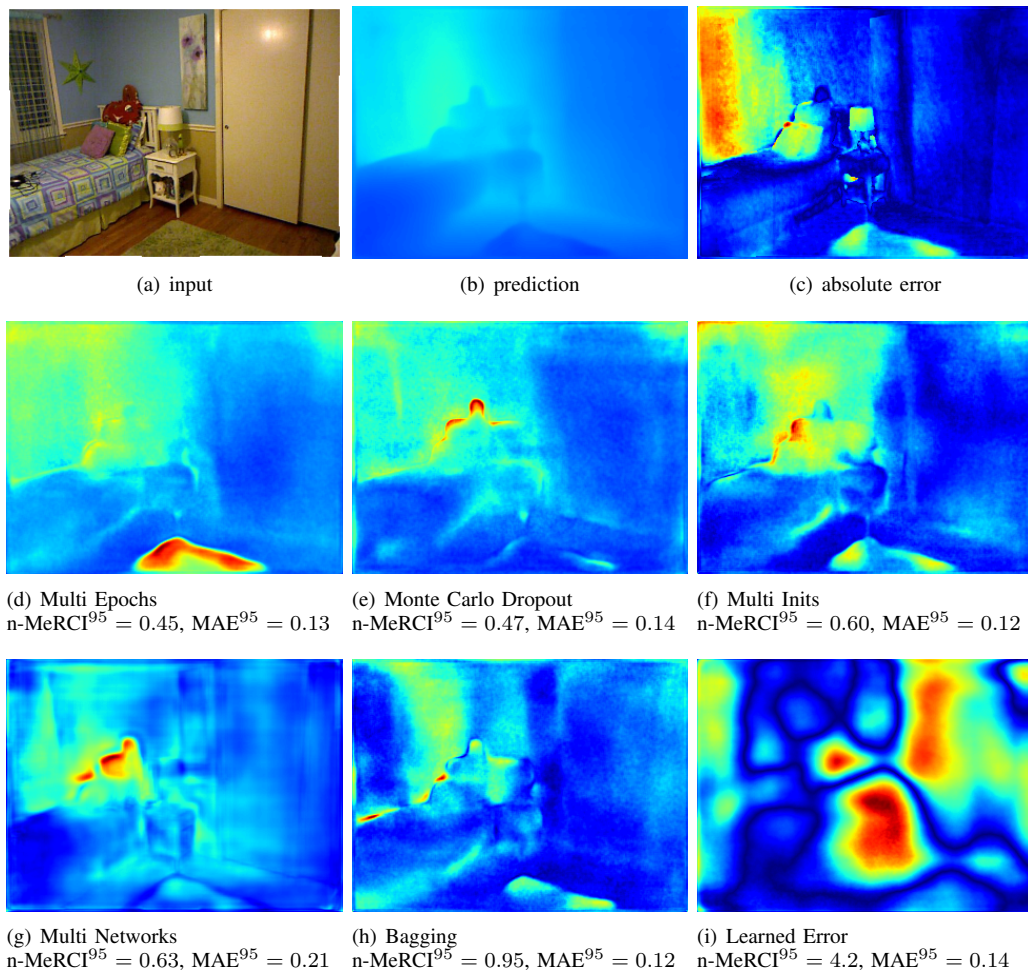


Fig. 3: A NYU Depth v2 test case. First row: the test image along with a predicted depth map and the corresponding absolute errors. Remaining rows: predictive uncertainties, ranked according to their n-MeRCI scores.

should be marked as uncertain: the carpet, the bedside lamp, the pillow and the window.

We display the 6 uncertainty estimates obtained with the different methods in Fig. 3. Only two methods show high uncertainty in the region corresponding to the window: namely Multi Epochs and Monte Carlo Dropout. This region is clearly the most important as it is the one with the highest errors according to the error map Fig. 3(c). Furthermore, Multi Epochs also displays high uncertainty at the carpet location, while Monte Carlo Dropout misses this place. This is the reason why n-MeRCI ranked these methods as the two top ones, but favors Multi Epochs. Meanwhile, we see that Multi Networks, Multi Inits or Bagging have poor correlation with the true error. Indeed, they present high uncertainty around the pillow but misses the other important spots: either the carpet and bedside lamp for Multi Networks or the window for Multi Inits and Bagging. Multi Networks also captures variance on the edges of the RGB image which is inconsistent with the true error. This is due to the fact that some of the different networks involved in the estimations produce sharper outputs but at the price of some

artifacts transferred from the RGB images [21]. Eventually, Learned Error completely fails to capture the relevant error information, which is correctly evaluated by n-MeRCI.

### C. Quantitative evaluation

Quantitative results are presented in Fig. 4 (MAE) and Fig. 5 (n-MeRCI). We withdraw 5% of the worst points in these experiments and n-MeRCI is computed locally for depth intervals of size 0.1 meter. First of all, it makes sense to inspect the MAE, shown on Fig. 4, as selecting a method purely on its ability to predict the uncertainty, without taking care of the precision, would be risky. The different methods are very close in terms of precision, with a slight advantage to Multi Epochs (MAE=0.29). The worse one is Learned Error, which makes sense as it does not average several predictions (MAE=0.34). We also observe that MAE increase as depth grows. Fig. 5 shows n-MeRCI scores for the different methods. In the same way as before, we see that n-MeRCI worsen as depth increase, which means that uncertainties are less accurate for higher depth intervals. This makes sense as these intervals are statistically less represented in the NYU dataset. Similarly to the single-image

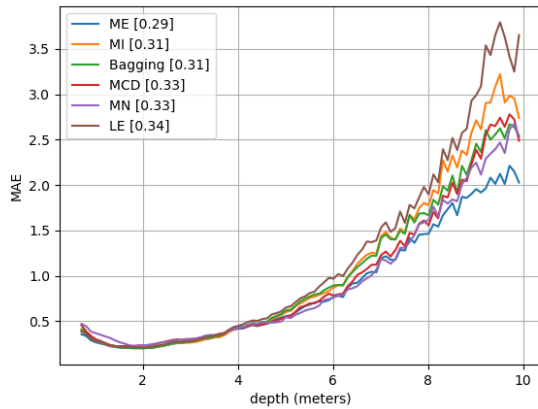


Fig. 4: MAE w.r.t. depth intervals for 95% of the points. Average MAE scores are displayed in brackets.

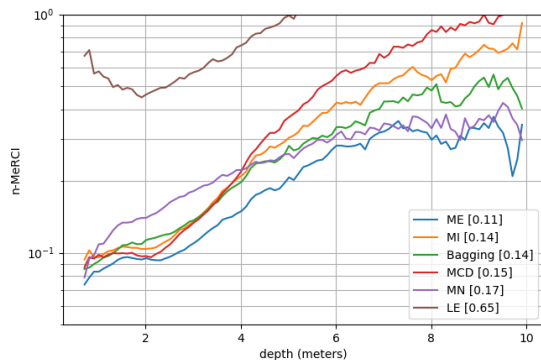


Fig. 5: n-MeRCI w.r.t. depth intervals ( $\alpha=95\%$ ). Average n-MeRCI<sup>95</sup> scores are displayed in brackets.

qualitative evaluation, n-MeRCI underlines Learned Error as the worst forecast (n-MeRCI = 0.65) and Multi Epochs as the best one (n-MeRCI = 0.11). This favored rank is supported by the fact that training a large network on a complex task and with a stochastic minimization scheme naturally yields oscillations around a minimum. What is more, along the last epochs, the network will focus on learning what is still improvable: *i.e.* regions with high errors. Thus, using the standard deviation of estimations obtained across epochs allows to capture these variations and highlights error-prone regions. On the other hand, the results obtained with LE show how difficult it is to regress uncertainties from images with a feed-forward network, without using any statistical properties of the main network.

## VII. CONCLUSIONS

This paper has proposed a metric, the Normalized Mean Rescaled Confidence Interval (n-MeRCI), dedicated to the assessment of predictive uncertainty. We showed that the proposed metric behaves as expected on a toy dataset, while giving latitude to be robust to outliers if needed (thanks to its percentile parameter). n-MeRCI has the advantage of being

directly applicable to regression tasks with deep learning, without needing any prior on the distribution. Finally, we used n-MeRCI to evaluate the merits of various uncertainty estimation methods applied to the problem of monocular depth estimation, and ranked them w.r.t. such merits. We showed that Multi Epochs, which is one of the fastest, is the one giving the best prediction of the uncertainties.

## REFERENCES

- [1] I. Kokkinos, “Surpassing humans in boundary detection using deep learning,” *CoRR*, abs/1511.07386, vol. 2, 2015.
- [2] C. Lu and X. Tang, “Surpassing human-level face verification performance on lfw with gaussianface,” in *AAAI*, 2015, pp. 3811–3819.
- [3] N. Sünderhauf, O. Brock, W. J. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Uprocroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, “The limits and potentials of deep learning for robotics,” *I. J. Robotics Res.*, vol. 37, no. 4-5, pp. 405–420, 2018.
- [4] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, 2016, pp. 372–387.
- [5] C. Liu, J. Gu, K. Kim, S. Narasimhan, and J. Kautz, “Neural RGB->D Sensing: Depth and Uncertainty from a Video Camera,” *arXiv:1901.02571 [cs]*, Jan. 2019.
- [6] F. Liu, C. Shen, G. Lin, and I. D. R. 0001, “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, pp. 2024–2039, 2016.
- [7] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, 2016, pp. 1050–1059.
- [8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in Neural Information Processing Systems*, 2017.
- [9] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in *Advances in Neural Information Processing Systems*, 2017, pp. 5580–5590.
- [10] Xiaoyan Hu and P. Mordohai, “A Quantitative Evaluation of Confidence Measures for Stereo Vision,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2121–2133, Nov. 2012.
- [11] L. Uusitalo, A. Lehtinen, I. Helle, and K. Myrberg, “An overview of methods to evaluate uncertainty of deterministic models in decision support,” *Environmental Modelling & Software*, vol. 63, 2015.
- [12] T. Gneiting, F. Balabdaoui, and A. E. Raftery, “Probabilistic forecasts, calibration and sharpness,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243–268, 2007.
- [13] T. Gneiting and A. E. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.
- [14] J. M. Hernández-Lobato and R. Adams, “Probabilistic backpropagation for scalable learning of bayesian neural networks,” in *International Conference on Machine Learning*, 2015, pp. 1861–1869.
- [15] M. Gong and Y. Yang, “Fast unambiguous stereo matching using reliability-based dynamic programming,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 998–1003, 2005. [Online]. Available: <https://doi.org/10.1109/TPAMI.2005.120>
- [16] J. Quinero-Candela, C. E. Rasmussen, F. Sinz, O. Bousquet, and B. Schölkopf, “Evaluating predictive uncertainty challenge,” in *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, 2006.
- [17] V. Kuleshov, N. Fenner, and S. Ermon, “Accurate uncertainties for deep learning using calibrated regression,” *International Conference on Machine Learning*, 2018.
- [18] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [19] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, “Snapshot ensembles: Train 1, get m for free,” *ICLR*, 2017.
- [20] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.
- [21] M. Moukari, S. Picard, L. Simon, and F. Jurie, “Deep multi-scale architectures for monocular depth estimation,” in *Image Processing (ICIP), 2018 IEEE International Conference on*. IEEE, 2018.