

# Bayesian Optimization Meets Laplace Approximation for Robotic Introspection

Matthias Humt<sup>1</sup>, Jongseok Lee<sup>1</sup> and Rudolph Triebel<sup>1,2</sup>

**Abstract**—In robotics, deep learning (DL) methods are used more and more widely, but their general inability to provide reliable confidence estimates will ultimately lead to fragile and unreliable systems. This impedes the potential deployments of DL methods for long-term autonomy. Therefore, in this paper we introduce a scalable Laplace Approximation (LA) technique to make Deep Neural Networks (DNNs) more introspective, i.e. to enable them to provide accurate assessments of their failure probability for unseen test data. In particular, we propose a novel Bayesian Optimization (BO) algorithm to mitigate their tendency of under-fitting the true weight posterior, so that both the calibration and the accuracy of the predictions can be simultaneously optimized. We demonstrate empirically that the proposed BO approach requires fewer iterations for this when compared to random search, and we show that the proposed framework can be scaled up to large datasets and architectures.

## I. INTRODUCTION

Space exploration, maintenance & inspection and surveillance & reconnaissance are just a few examples of application areas where robots are needed that are capable of long-term autonomous operation. One of the crucial aspects of achieving such long-term autonomy is the addition of the cognitive ability to doubt and understand their own failures and further assess the quality of their own internal states such as sensor data, predictions of a learning method, representation of the world etc. Such knowledge, when examined accurately, can alter the behavior of robots by including the assessed reliability score into the planning and mission management process so that higher mission success rates can be achieved. Such cognitive abilities are called robotic introspection [1].

For Deep Neural Networks (DNNs), the quality of the learned predictor can be assessed by generating distributions rather than a single most-likely prediction. There exists ample evidence that such confidence estimates, if well-calibrated w.r.t the accuracy of the learning algorithm, can lead to improved decision-making and failure avoidance [2]–[4]. Unfortunately, obtaining these confidence estimates for DNNs is often subject to a fidelity-scalability trade-off; highly scalable methods such as MC-dropout [5] often underestimate predictive uncertainty [6] while sampling methods such as HMC [7] are widely recognized as gold standard but do not scale to large datasets and modern DNN architectures. As the requirements vary depending on the use-case, we hypothesize that an easy-to-use method that can provide a

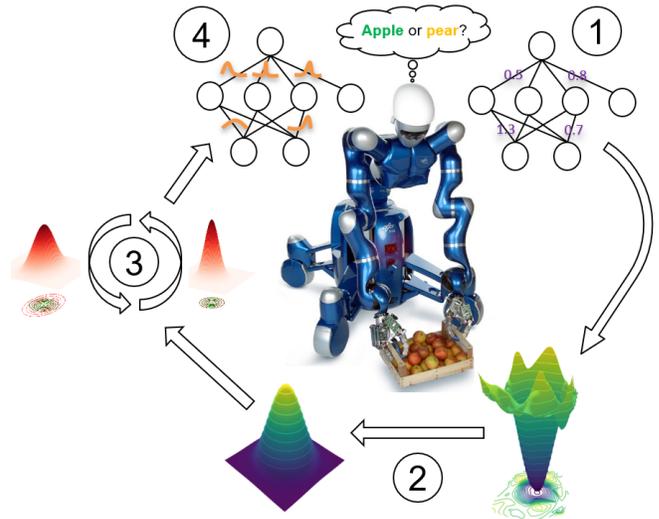


Fig. 1: To equip our robots with introspective qualities, we transform a Deep Neural Network (DNN) (1) into a Bayesian Neural Network (4) using Laplace Approximation (2) which employs information from the inverse curvature (Hessian) of the loss landscape of the DNN (2, right) for the covariance matrix of a multivariate Gaussian as an approximation to the DNNs weight posterior distribution (2, left) and mitigate its tendency of underfitting through Bayesian Optimization (3).

first line-of-attack with a balanced fidelity-scalability would be beneficial for save long-term autonomous operation in robotics. For this purpose, we introduce a scalable LA technique [8], [9] in the context of robotic introspection. As depicted in Fig. 1, LA computes the DNNs posterior with a Gaussian distribution around a local minimum, where its mean is the maximum a-posteriori (MAP) estimate of the DNNs parameters (i.e. the trained weights) while the covariance matrix is specified by the scaled inverse of the DNNs Hessian matrix (i.e. the matrix of 2nd order derivatives of the loss w.r.t the weights). Here, the Hessian matrix can be obtained via 2nd order Taylor series expansion on the log-likelihood and requires relaxed assumptions on the DNN architecture (e.g. no mandatory dropout layers). By exploiting Kronecker factorization of the Hessian, more expressive posterior families than the Bernoulli distribution or diagonal approximations of the covariance matrix can further be modelled, even in a large scale setting that is relevant for many robotic applications [10]. This was demonstrated in [11], where LA induced uncertainty was successfully

<sup>1</sup> Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. matthias.humt@dlr.de

<sup>2</sup> Computer Vision Group, Technical University of Munich, Garching, Germany

employed to improve robustness and accuracy of deep sensor fusion for end-to-end trainable visual-inertial odometry.

More importantly, we propose a new Bayesian Optimization (BO) technique to address the under-fitting problem of LA which is its inherent drawback arising from (i) the use of 2nd order Taylor series expansion on the DNNs local log-likelihood restricting the assumed posterior to be a bell-shaped distribution (i.e. it cannot handle an asymmetrical true posterior and places a large probability mass in low probable sets of parameters), and (ii) a typical over-parameterization of DNNs which result in an undetermined Hessian matrix. Opposed to the brute force grid search strategy used in [9], we empirically show that the proposed BO technique can provide faster convergence for better calibration and accuracy of the predictions. In addition, as we show that the proposed framework can be scalably applied to ImageNet, we demonstrate an alternative approach to MC-dropout, which is a standard approach to obtaining model (i.e. epistemic) uncertainty in DNNs.

In summary, our contributions are:

- We propose to employ a scalable yet straight forward Laplace Approximation technique in the context of robotic introspection.
- We propose a Bayesian Optimization technique to tackle the under-fitting problem of Laplace Approximation.
- We demonstrate empirically that the proposed technique requires fewer iterations when compared to a random search, and the resulting framework scales to large network architectures and datasets.

## II. RELATED WORK

Uncertainty estimation in DNNs has received increasing attention in various communities and a variety of methods have been proposed. Broadly, we build upon the paradigm of Bayesian Deep Learning (BDL) which aims to estimate the posterior distribution over the weights of DNNs, which allows for predictive uncertainty to be computed by marginalization over the possible values of weights [8]. MC-dropout [5] is a prominent example within this umbrella, where the authors showed that training a DNN with dropout layers is in some sense equivalent to performing variational inference - an optimization based approach to estimating the posterior distribution rather than sampling. While MC-dropout has been widely adopted in the robotics community and beyond [12]–[14], some drawbacks of the framework include (i) the need for dropout layers while most modern networks are trained with batch normalization, and (ii) lack of expressiveness of the posterior family. Concurrently, more sophisticated methods based on sampling [7] and variational inference [15] have been widely studied for DNNs. However, these arguably complex Bayesian inference procedures have not been popular as they are often not scalable or fragile [16]. We instead build upon LA [8], [9], one of the simplest and deterministic inference methods with an attempt to make the framework more practical for real-world robotics applications. We are aware that there exists alternatives to BDL, namely post-hoc calibration methods [17], [18],

ensembles [19], probabilistic programming [20], [21] etc. but these approaches are rather orthogonal to our work.

The importance of robotic introspection for decision making has been first emphasized in [1], where the authors demonstrated that on top of standard metrics for classification, namely precision and recall, the quality of confidence estimates are an additional crucial measure that can help to avoid catastrophic failures of robots, and can also be utilized for active learning [3]. Since then, [22]–[24] have proposed to learn the failures of a perception module with a DNN, and [25] demonstrated a failure recovery method. The backbone of these methods is to perform extensive data collection in the regime where the perception module fails, and use such data for training a DNN based failure predictor. Coping with non-stationary data [1] (also called open-set conditions in [13]) by means of unsupervised novelty detection is a further notable class of methods that attempted to achieve robotic introspection [26], [27]. Our method relies on Bayesian principles and is not tied to a specific task but can also be applied to pruning, active learning, continual learning, and 2nd order optimization for DNNs, which are other crucial frontiers towards long-term autonomy.

## III. BAYESIAN HYPERPARAMETER OPTIMIZATION FOR LAPLACE APPROXIMATION

### A. Laplace Approximation

Let  $f_{\mathbf{W}}(\mathbf{x})$  be a DNN with weights  $\mathbf{W}$  that transforms inputs  $\mathbf{x}$  into outputs  $\hat{\mathbf{y}}$ . Here,  $\mathbf{x}$  are images and  $\hat{\mathbf{y}}$  are predicted class probabilities with corresponding ground truth labels  $\mathbf{y}$ . After training using the data  $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ , the weights  $\mathbf{W}^*$  are the mode of the networks posterior probability distribution which can be approximated by a 2nd order Taylor expansion around this mode which is commonly referred to as Laplace Approximation (LA),

$$\log p(\mathbf{W}|\mathcal{D}) \approx \log p(\mathbf{W}^*|\mathcal{D}) - \frac{1}{2}(\mathbf{W} - \mathbf{W}^*)^T H(\mathbf{W} - \mathbf{W}^*), \quad (1)$$

where  $H$  is the average Hessian of the negative log posterior. Taking the exponential, the right hand side of eq. 1 is of Gaussian functional form, resulting in

$$\mathbf{W} \sim \mathcal{N}(\mathbf{W}^*, H^{-1}). \quad (2)$$

Sampling from this posterior allows to perform approximate Bayesian inference for an unseen datum  $\mathbf{x}^*$  using Monte Carlo integration

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{W})p(\mathbf{W}|\mathcal{D})d\mathbf{W} \quad (3)$$

$$\approx \frac{1}{T} \sum_{t=1}^T p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{W}_t), \quad (4)$$

where  $\mathbf{W}_t$  are weight samples from eq. 2. Note that those samples can be drawn efficiently using a matrix normal distribution [9]. Because the Hessian of a NN is in general too large to be computed, stored and inverted, it is usually approximated by the diagonal or block-diagonal and Kronecker factored (KFAC) [28] Fisher information matrix or the

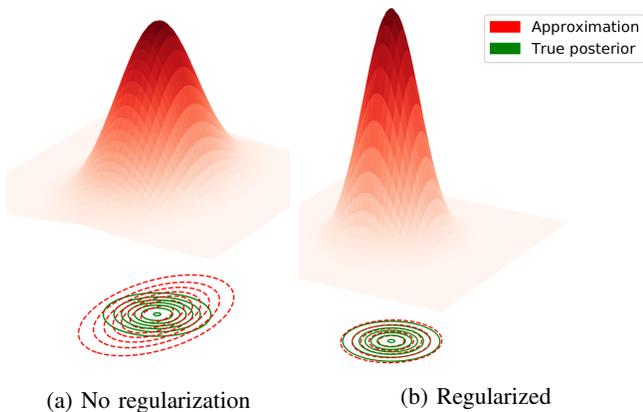


Fig. 2: An illustration showing how Bayesian Optimization can help to remove the under-fitting problem where LA puts high probability mass on areas of low true probability (left). This is achieved by regularizing the covariance matrix by optimizing the hyperparameters  $N$  and  $\tau$  so that the under-fitting areas are removed (right).

Generalized Gauss-Newton matrix [29]. For generality we will refer to any of these second-order matrices as *curvature matrices*  $C_\ell$  for  $\ell \in (0, L)$  layers of the DNN.

Several problems can arise using this posterior approximation. (1) The curvature matrix might not be positive semi-definite (p.s.d), so it cannot be inverted. (2) The approximations that made the problem tractable might lead to an overestimation of the variance of the Gaussian distribution in certain directions in parameter space. (3) The approximation of the NNs posterior using a multivariate Gaussian distribution itself might place probability mass in low probability areas of the true posterior [9]. See Fig. 2.

To mitigate those problems, regularization can be used. Following [9], we make use of the simple regularization scheme

$$C = N\mathbb{E} \left[ -\frac{\partial \log p(\mathcal{D}|\mathbf{W})}{\partial \mathbf{W}^2} \right] + \tau I, \quad (5)$$

where  $\tau$  can be interpreted as the precision of a Gaussian prior on the weights of the DNN, equivalent to  $L_2$ -regularization, and  $N$  the size of the dataset, but they can also simply be treated as hyperparameters to be optimized w.r.t. to a designated performance metric on a validation set.

Note, that a single set of regularization parameters is used for all  $C_{\ell \in L}$ , so one might wonder if maintaining multiple sets of hyperparameters for, e.g., layers with curvature of dissimilar magnitude, could result in superior performance. We will come back to this in section III-C.

## B. Bayesian Optimization

*Bayesian Optimization* (BO) is a form of sequential, model-based optimization scheme for finding the minimum of an expensive to evaluate and potentially noisy black-box function  $q(r)$ . Black-box functions means, that we do not have access to gradient information and the function is too expensive to evaluate to compute numerical derivatives.

Instead,  $q(r)$  is approximated by a probabilistic model, trained alongside the optimization procedure. The next best candidate  $r_{n+1}$  in the search space  $\mathcal{R}$  to be evaluated by  $q(r)$  is found by first computing the analytical posterior distribution of the probabilistic model and then minimizing an easy to evaluate *utility function*  $u(r)$  based on this posterior, with the goal of finding an optimal point  $r^*$ . More formally we get

$$r_{n+1} = \arg \min_{r \in \mathcal{R}} u(r) \quad (6)$$

$$r^* = \arg \min_{r \in \mathcal{R}} q(r). \quad (7)$$

Typical choices for  $u$  include *Expected Improvement* (EI) or *Lower Confidence Bound* (LCB) while a common choice of model is the *Gaussian Process* (GP) [30]. This optimization procedure stands in contrast to the commonly used grid or random search, where candidate points are either predefined (grid search) or sampled randomly from some predefined distribution (random search).

An typical example in the BO domain is DNN training, where the next best set of hyperparameters—corresponding to the next best candidate  $r_{n+1}$ —like the learning rate, regularization strength or dropout rate, cannot be directly inferred using e.g. gradient information and the performance of the chosen hyperparameters can only be evaluated after hours, days or even weeks of training, making it an extremely expensive black-box function  $q(r)$ .

## C. Bayesian Optimization meets Laplace Approximation

We face the same problem when searching for an optimal set of regularization parameters for LA, i.e.  $r = (N, \tau)$ . As can be seen from eq. 3,  $T$  forward passes through the network need to be performed for each data point in the dataset where larger  $T$  yield more accurate approximations. Using DNNs and a large datasets, evaluating the performance of one set of  $N$  and  $\tau$  is a significant computational burden and is further complicated due to noisy results stemming from the stochastic sampling procedure of LA. However, as explained earlier, the regularization parameters have a defining influence on the performance of LA, as excessive regularization reduces the probabilistic network output to a deterministic one while too little regularization hurts the networks accuracy.

The problem is further aggravated when separate sets of hyperparameters for different layers are introduced, as we are operating under the curse of dimensionality. We therefore propose, with a particular focus on the aforementioned multiple-set case, the use of BO of LA hyperparameter search. In particular, we make use of a GP model in conjunction with the *Hedge* utility function, which chooses probabilistically between standard choices like EI and LCB [30], to find a single or multiple sets of regularization parameters for LA.

This is done by an independent optimization of the three utility functions EI, LCB and *Probability of Improvement* (PI), each returning a candidate for the next evaluation  $n+1$ .

The best proposal is picked using a normalized and uniformly initialized weight vector  $\mathbf{w}^n$

$$p(\mathbf{w}^n) = \frac{\mathbf{w}^n}{w_1^n + w_2^n + w_3^n},$$

which is updated using cost  $c_i^n$  for utility function  $i \in \{1, 2, 3\}$  at evaluation  $n$  from the chosen performance metric as

$$w_i^{n+1} = w_i^n \cdot \beta^{c_i^n},$$

where  $\beta \in [0, 1]$  is a hyperparameter of the Hedge algorithm [31].

The most common performance metrics for classification are accuracy and negative log-likelihood (NLL), i.e. the average cross entropy of the target class  $y_{ik} = 1$  for sample  $i$  and the predicted probability  $\hat{y}_{ik}$ , though they are insufficient to capture important notions when reliable uncertainty estimates are required. As was shown in [18], modern DNNs are severely miscalibrated, providing accurate yet highly overconfident predictions where the predictive uncertainty does not reflect the empirical frequency of being right or wrong. And while NLL captures notions of uncertainty in the form of predictive entropy, it does not account for the fact that DNNs can overfit w.r.t. classification *and* probabilistic error, s.t. NLL increases while accuracy further improves. Next to accuracy, we therefore make use of *expected calibration error* (ECE) [18] as the (binned) weighted difference between accuracy and confidence, to provide a more faithful representation of the classifiers performance, ignoring additional benefits of BNNs s.a. superior detection capabilities of out-of-domain examples or increased resilience to adversarial attacks [9] in this discussion for the sake of brevity. We subsequently provide definitions of the previously introduced performance metrics for reference.

#### IV. EXPERIMENTS

To empirically validate the effectiveness, efficiency and scalability of BO for LA we conduct a small scale comparison of BO and random search on the CIFAR-10 [32] dataset and a large scale experiment on ImageNet [33], using publicly available pre-trained networks to facilitate reproducibility. All experiments were implemented using PyTorch in Python and we make the code publicly available.

##### A. Small scale experiment

In the small scale experiment we make use of a ResNet variant [34] with 18 layers (ResNet18), pre-trained on CIFAR-10<sup>1</sup> to 93.32% validation accuracy (and therefore conversely an error of 6.68%) on random 50% of the official validation set, which we will use for the hyperparameter optimization. The ECE is 4.57% resulting in a baseline cost of 11.25 as the sum of classification error and ECE which we will use as performance measure.

We use our own LA implementation with  $T = 30$  posterior samples from the diagonal posterior approximation computed

using the CIFAR-10 training set with the same augmentation techniques as applied during network training<sup>2</sup>.

Figure 3 shows the lowest cost found by BO and RS for 50 evaluations averaged over 20 runs (solid line) and one standard deviation (shaded area) when searching for one set of hyperparameters for LA.

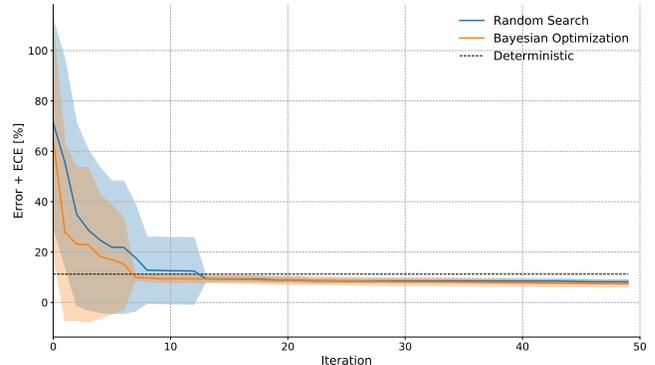


Fig. 3: Average performance of BO and RS for 50 evaluations using ResNet18 on CIFAR-10 with 2 hyperparameters. We note that BO converges faster than random search which is evident in the transient phase (i.e. before iteration 15). This shows that the proposed BO can provide a more powerful scheme to optimize calibration and accuracy simultaneously.

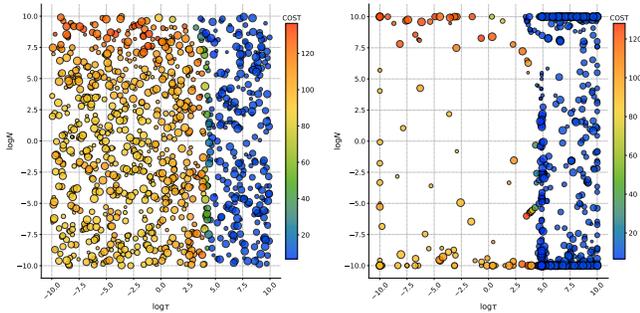
As can be observed, BO for LA is able to surpass the performance of the standard DNN (black dashed line) after seven evaluations compared to 13 for RS and additionally finds a marginally superior final set of parameters. We can further see, that the variance of the RS performance is much higher for a small number of evaluations, making it a less reliable choice for a high cost setting. We additionally visualize the behavior of both algorithms by showing their sampling strategies side by side (Fig. 4). Clearly, BO makes better use of its sampling budget by largely excluding the high cost area of insufficient regularization.

We conduct an additional small scale experiment with the exact same setup as before except for the introduction of an additional set of hyperparameters. The first set is used to regularize the convolutional layers while the second set regularizes the final fully connected layer. While this is mostly done to highlight the benefit of BO over RS in higher dimensional search spaces, it is reasonable to expect performance gains from regularizing individual layers differently, as the magnitude of their curvature matrices varies significantly. The results are shown in Fig. 5. Again, BO converges faster than RS and is able to find slightly superior sets of hyperparameters. The results are summarized in table I.

It should be noted that, even though we did not strive for state-of-the-art performance by making use of the rather simplistic diagonal curvature approximation, LA could sig-

<sup>1</sup>Publicly available at <https://github.com/huyvnphan/PyTorch-CIFAR10>

<sup>2</sup>We opted for the simple diagonal approximation as opposed to more sophisticated approximations like KFAC as our focus lies on the benefits of using BO for LA and not on achieving state-of-the-art performance.



(a) Sampling behavior of RS. (b) Sampling behavior of BO.

Fig. 4: While RS samples the search space uniformly at random, BO focuses the search on certain areas, largely avoiding the high cost area to the left. High cost is depicted from red to orange while low cost is shown in blue. This demonstrates the efficiency of the proposed algorithm in improving the under-fitting problem of Laplace Approximation.

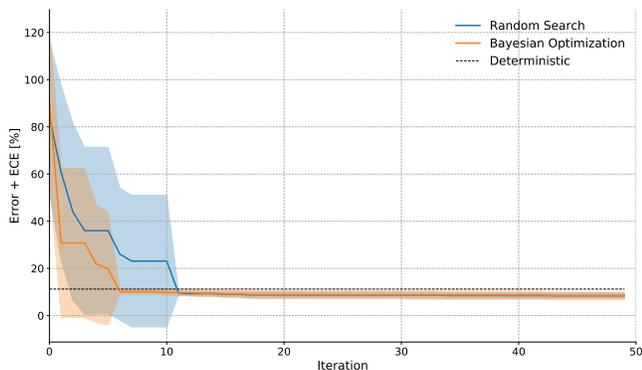


Fig. 5: Average performance of BO and RS for 50 evaluations using ResNet18 on CIFAR-10 with 4 hyperparameters. As the number of hyperparameters grow, and as we treat to regularize the posterior of each layers separately, BO further outperforms random search w.r.t convergence rate.

nificantly improve the performance of ResNet18 on CIFAR-10, with a slight improvement in accuracy and a major reduction of mis-calibration, which might not be immediately visible from the preceding figures.

### B. Large scale experiment

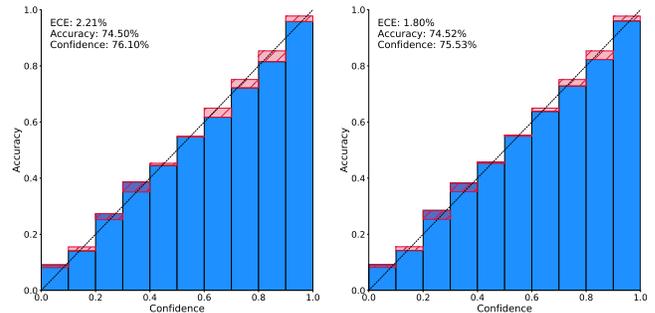
We finally conduct a large scale experiment on ImageNet where BO is especially useful, because every single performance evaluation of the chosen LA hyperparameters is extremely costly. We change the network architecture to DenseNet121 [35] which comes pre-trained on ImageNet as part of the *torchvision* PyTorch module and achieves a baseline accuracy of 74.42% and ECE of 2.63% on a randomly chose 50% subset of the official ILSVRC-2012 validation set. The other half is used to perform the LA hyperparameter optimization. Guided by the results from the previous experiments we define a fixed search budget of 10 evaluations for both RS and BO.

The results are depicted in figure 6 in the form of

TABLE I: Small scale results after convergence. Higher is better for accuracy (Acc) while lower is better for expected calibration error (ECE). After convergence, the combination of LA and BO achieves better calibration performance. Furthermore, accuracy of the predictions has improved over the deterministic counterpart due to the marginalization over the possible weights.

	Baseline		LA + RS		LA + BO	
	Acc.	ECE	Acc.	ECE	Acc.	ECE
2 parameters	93.32	4.57	93.68	0.41	93.6	<b>0.18</b>
4 parameters	93.32	4.57	93.8	0.7	93.62	<b>0.17</b>

two *Reliability Diagrams* inspired by [18], which visualize accuracy as a function of confidence by showing the average accuracy per confidence bin as blue bars, and their difference (the calibration gap) as red bars on top. A perfectly calibrated network exhibits the same accuracy in each bin as the corresponding confidence dictates, resulting in the identity function, indicated by the dashed black line.



(a) Random Search

(b) Bayesian Optimization

Fig. 6: Reliability diagrams for LA of DenseNet121 with hyperparameter optimization using RS and BO.

Even though both methods find regularization parameters that outperform the deterministic network, BO achieves lower ECE and slightly higher accuracy (see figure labels) within the given optimization budget. As the deterministic network was already relatively well calibrated (ECE of 2.63%), absolute performance gains can mostly be seen for confidence bin 0.6 – 0.7, where the BO results show significantly lower overconfidence.

## V. CONCLUSION

In this work, we have presented a practical approach based on Laplace Approximation for estimating the model uncertainty in deep learning methods, and further proposed to use a novel Bayesian optimization technique to regularize the obtained representation of model uncertainty to better fit the ground truth. We demonstrated both performance and scalability by outperforming random search as a strong baseline and further demonstrated the relevancy in robotics by showing the scalability of our approach on the ImageNet dataset using a large deep neural network.

## REFERENCES

- [1] H. Grimmer, R. Paul, R. Triebel, and I. Posner, "Knowing when we don't know: Introspective classification for mission-critical decision making," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4531–4538.
- [2] H. Grimmer, R. Triebel, R. Paul, and I. Posner, "Introspective classification for robot perception," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 743–762, 2016.
- [3] R. Triebel, H. Grimmer, R. Paul, and I. Posner, "Driven learning for driving: How introspection improves semantic mapping," in *Robotics Research*. Springer, 2016, pp. 449–465.
- [4] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. Weller, "Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning," in *International Joint Conferences on Artificial Intelligence, Inc.*, 2017.
- [5] 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.
- [6] Y. Li and Y. Gal, "Dropout inference in bayesian neural networks with alpha-divergences," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2052–2061.
- [7] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.
- [8] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.
- [9] H. Ritter, A. Botev, and D. Barber, "A scalable laplace approximation for neural networks," in *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, vol. 6. International Conference on Representation Learning, 2018.
- [10] J. Lee, M. Humt, J. Feng, and R. Triebel, "Estimating model uncertainty of neural networks in sparse information form," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- [11] K. Shinde, J. Lee, M. Humt, A. Sezgin, and R. Triebel, "Learning multiplicative interactions with bayesian neural networks for visual-inertial odometry," in *Workshop on AI for Autonomous Driving (AIAD) of the 37th International Conference on Machine Learning*, 2020.
- [12] J. Feng, M. Durner, Z.-C. Marton, F. Balint-Benczedi, and R. Triebel, "Introspective robot perception using smoothed predictions from bayesian neural networks," in *2019 International Symposium on Robotics Research (ISRR)*, 2019.
- [13] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.
- [14] G. Kahn, A. Villafior, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," 2017.
- [15] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, ser. COLT '93. New York, NY, USA: ACM, 1993, pp. 5–13.
- [16] A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernandez-Lobato, and A. L. Gaunt, "Deterministic variational inference for robust bayesian neural networks," in *International Conference on Learning Representations*, 2019.
- [17] J. Wenger, H. Kjellström, and R. Triebel, "Non-parametric calibration for classification," ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. Online: PMLR, 26–28 Aug 2020, pp. 178–190. [Online]. Available: <http://proceedings.mlr.press/v108/wenger20a.html>
- [18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1321–1330.
- [19] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.
- [20] D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei, "Edward: A library for probabilistic modeling, inference, and criticism," 2017.
- [21] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 973–978, 2019.
- [22] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1743–1750.
- [23] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: autonomously identifying perception failures for self-driving cars," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, 2018.
- [24] C. Gurău, D. Rao, C. H. Tong, and I. Posner, "Learn from experience: probabilistic prediction of perception performance to avoid failure," *The International Journal of Robotics Research*, vol. 37, no. 9, pp. 981–995, 2018.
- [25] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5824–5829.
- [26] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics, Science and Systems (RSS)*, 2017.
- [27] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," in *Conference on Robot Learning*. PMLR, 2020, pp. 384–393.
- [28] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *International conference on machine learning*, 2015, pp. 2408–2417.
- [29] A. Botev, H. Ritter, and D. Barber, "Practical gauss-newton optimisation for deep learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 557–565.
- [30] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [31] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.
- [32] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.