Evaluating Approximate Inference in Bayesian Deep Learning

Andrew Gordon Wilson Pavel Izmailov Matthew D. Hoffman
Yarin Gal Yingzhen Li Melanie F. Pradier Sharad Vikram
Andrew Foong Sanae Lotfi Sebastian Farquhar

March 31, 2021

Abstract

Uncertainty representation is crucial to the safe and reliable deployment of deep learning. Bayesian methods provide a natural mechanism to represent epistemic uncertainty, leading to improved generalization and calibrated predictive distributions. Bayesian methods are particularly promising for deep neural networks, which can represent many different explanations to a given problem corresponding to different settings of parameters. While approximate inference procedures in Bayesian deep learning are improving in scalability and generalization performance, there has been no way of knowing, until now, whether these methods are working as intended, to provide ever more faithful representations of the Bayesian predictive distribution. In this competition we provide the first opportunity to measure the fidelity of approximate inference procedures in deep learning through comparison to Hamiltonian Monte Carlo (HMC). HMC is a highly efficient and well-studied Markov Chain Monte Carlo (MCMC) method that is guaranteed to asymptotically produce samples from the true posterior, but is prohibitively expensive in modern deep learning. To address this computational challenge, we have parallelized the computation over hundreds of tensor processing unit (TPU) devices.

Understanding the fidelity of the approximate inference procedure has extraordinary value beyond the standard approach of measuring generalization on a particular task: if approximate inference is working correctly, then we can expect more reliable and accurate deployment across any number of real-world settings. In this regular competition, we invite the community to evaluate the fidelity of approximate inference procedures across a range of tasks, including image recognition, regression, covariate shift, and medical applications, such as diagnosing diabetic retinopathy. All data are publicly available, and we will release several baselines, including stochastic MCMC, variational methods, and deep ensembles.

Keywords

Uncertainty, safety, robustness, approximate inference, Bayesian deep learning

Competition type

Regular

1 Competition description

1.1 Background and impact

While deep learning has been revolutionary for machine learning, most modern deep learning models cannot represent their uncertainty nor take advantage of the well-studied tools of probability theory. The community has been immensely active in addressing this gap, with the introduction of new deep learning models that use Bayesian inference techniques, and Bayesian models that incorporate deep learning elements. The broad and consistent community interest in these topics is clearly evidenced by the NeurIPS Bayesian Deep Learning workshop being the second largest workshop at the conference every year since 2016. This broad interest is also clear from major tutorials on Bayesian deep learning and uncertainty representation in deep learning at NeurIPS 2019, ICML 2020, and NeurIPS 2021 [Khan, 2019, Wilson, 2020, Tran et al., 2020].

The use of Bayesian techniques in deep learning can be traced back to the 1990s, in seminal works by Radford Neal [Neal, 1996] and David MacKay [MacKay, 1995]. These works gave rise to tools to reason about deep models' confidence, and achieved state-of-the-art performance on many tasks at the time. With a resurgence of deep learning, there has been extraordinary progress in the last five years for scaling approximate Bayesian inference procedures to modern architectures and datasets.

Many of these procedures have provided promising performance on tasks of public interest, such as medical diagnoses and more reliable autonomous driving [Leibig et al., 2017, Filos et al., 2019]. For example, in medical diagnosis, it is not sufficient to simply label an image as pathological or healthy. Instead, we need to make a decision about treatment based on *probabilities* of class labels. For this purpose, Bayesian methods represent *epistemic uncertainty* over different hypotheses for the data, in order to provide a full predictive distribution. This predictive distribution is then crucial in decision making, as it can be combined with a loss function that recognizes asymmetry in outcomes, and that rare mistakes can be extraordinarily costly. A false negative, for instance, is often much more costly than a false positive.

However, there has been no mechanism for understanding whether approximate inference procedures in deep learning are working as intended, and providing a faithful approximation of a Bayesian predictive distribution. Indeed, standard metrics, such as generalization accuracy, or negative log likelihood, provide no way of separating the effects of model specification and inference procedure [Yao et al., 2019].

In this competition we provide the first opportunity to measure the *fidelity* of approximate inference procedures in deep learning through comparison to Hamiltonian Monte

Carlo (HMC) [Neal et al., 2011]. HMC is a highly efficient and well-studied Markov Chain Monte Carlo (MCMC) method that is guaranteed to asymptotically produce samples from the true posterior, but is prohibitively expensive in modern deep learning: HMC can take tens of thousands of training epochs to produce a single sample from the posterior. To address this computational challenge, we have parallelized the computation over hundreds of tensor processing unit (TPU) devices. We provide extensive details for our procedure, as well as comparisons to several popular baselines, in Izmailov et al. [2021].

This competition provides a standardized mechanism for evaluating the fidelity of a wide variety of approaches to approximate inference in deep learning. Each participant is given access to our HMC samples for a variety of architectures across several reference datasets. Participants only need to provide access to the predictive distribution from their procedure, and our evaluation framework then creates a leaderboard of all methods. In the evaluation phase of the competition, participants submit code for their inference procedures on evaluation datasets, and we locally evaluate the fidelity of inference. We consider problems for image recognition, regression, covariate shift, and healthcare. Healthcare applications form the focus of the evaluation datasets. Total runtime of the approximate inference procedures is constrained to no more than ten times standard SGD training, which covers essentially all modern approximate inference procedures, but is several orders of magnitude less expensive than full HMC.

The outcome of the competition will provide an enormous resource for understanding the efficacy of many approximate inference procedures, separating model specification and inference in evaluation, and the design of new inference algorithms which can provide reliable inference at a much lower cost than HMC, which is otherwise inaccessible to machine learning practitioners. More broadly, the development of high fidelity Bayesian inference procedures in deep learning is a crucial component of building safe and robust systems for automatic decision making — which requires faithful representations of uncertainty, and reliable predictive distributions.

1.2 Novelty

Up until now there has been no way to measure the fidelity of approximate inference procedures in deep learning. We have invested significant resources into obtaining, for the first time, high fidelity posterior samples for modern neural networks. Therefore this is a completely new competition. Nothing like it has been proposed in the past.

1.3 Data

The datasets for this competition are derived from publicly available datasets (see Table 1). Each dataset comes with train/test subsets in its original form. However, as the goal of this competition is to evaluate inference accuracy against the exact Bayesian posterior predictive (see Section 1.5 on metrics), the "HMC label distributions" will be generated by

Table 1: **Required Tasks.** Submissions to the competition will be requested to demonstrate their effectiveness at approximating the predictive posterior across a wide range of easily-accessible applications and architectures. For CIFAR datasets, both the standard test set and corrupted versions [Hendrycks and Dietterich, 2019]. For UCI, we will use the regression datasets chosen in Hernández-Lobato and Adams [2015].

Prediction Type	Dataset	Architecture	Metrics Assessed
	Reference Datasets		
Classification	MNIST-(C)	LeNet-5	Top-1 Agreement & Total Variation
	SVHN	LeNet-5	Top-1 Agreement & Total Variation
	CIFAR-10-(C)	ResNet-20	Top-1 Agreement & Total Variation
	CIFAR-100-(C)	ResNet-20	Top-1 Agreement & Total Variation
	IMDB	CNN-LSTM	Top-1 Agreement & Total Variation
Regression	UCI	3x200 FCNN	Wasserstein Distance
	Evaluation Datasets		
Classification	CIFAR-10-(C)	TBD	Top-1 Agreement & Total Variation
	MedMNIST [Yang et al., 2020]	LeNet-5	Top-1 Agreement & Total Variation
	Diabetic Retinopathy [Filos et al., 2019]	ResNet-20	Top-1 Agreement & Total Variation
Regression	UCI-Gap [Foong et al., 2019]	3x200 FCNN	Wasserstein Distance

us to evaluate the submitted algorithms. These "HMC label distributions" will be ready prior to the official launch of the competition, and the generated labels on the datasets used for final evaluation of the submissions (evaluation datasets in Table 1) will be invisible to participants. At the time of writing this proposal, we have already collected a large number of HMC samples on the CIFAR-10, CIFAR-100 and IMDB datasets (see the attached paper in Section 1.7).

The included datasets (original version) are released under MIT/CC/Apache licences, all of them allow distribution and adaptation for non-commercial use. The generation of the "HMC label distributions" is conducted by running the Hamiltonian Monte Carlo algorithm on network architectures mentioned in Table 1. The HMC computation is parallelized over hundreds of tensor processing unit (TPU) devices to ensure high fidelity posterior samples. This process is automated, no human annotation is required. We aim to open-source the "HMC label distributions" after the competition under an appropriate licence for non-commercial use. When releasing, we will include a statement on the algorithmic labelling procedure, and we will make it clear that the HMC simulation results should not be regarded as ground truths from the underlying data distribution.

These datasets cover different types of supervised learning tasks (regression and classification), and the evaluation is conducted for both in-distribution and out-of-distribution scenarios. Also the datasets and reference architectures in Table 1 are selected to enable the collection of reliable HMC simulation results. We believe these are comprehensive benchmarks to make the competition interesting and draw conclusive results.

1.4 Tasks and application scenarios

Bayesian deep learning methods have countless potential applications precisely because inferring the Bayesian posterior distribution is such a powerful principled way to incorporate the information contained in a training dataset. These scenarios include:

- Safe medical diagnostics: automatically handling clear-cut diagnoses while elevating difficult decisions to medical professionals who can request further scans.
- Rare or under-represented inputs: recognizing the uncertainty present when individuals come from groups that are under-represented in datasets and seeking guidance from experts.
- Covariate shift: identifying situations where covariate shift makes model predictions unreliable, for example in autonomous driving.

In recent work, researchers have made great progress on specific metrics associated with these sorts of applications of Bayesian deep learning (e.g., Lakshminarayanan et al. [2017], Leibig et al. [2017], Maddox et al. [2019], Filos et al. [2019], Ovadia et al. [2019]).

However, prior work has mostly focused on simple metrics like accuracy, log-likelihood, and rejection accuracy. It is possible to score highly on these in individual cases even if a method generalizes poorly to other tasks due to issues with approximate inference. Rather than focus on generalization error for a set of tasks, we instead wish to understand which approximate inference methods are performing as intended to produce *high quality posterior approximations*, leading to more calibrated expectations of their applicability across a broad range of settings.

To this end, we ask the participants to implement approximate inference for the application scenarios listed in Table 1. We aim to balance the goal of demonstrating performance across as wide a range of settings as possible, while at the same time relying on datasets and settings that will be familiar to as many researchers as possible and accessible to researchers with computational and budgetary constraints. All of the datasets involved are publicly available and can be found as part of the `torchvision` or `tensorflow` packages, the UCI repository, or as a Kaggle download. In each case, contestants will provide a predictive posterior distribution rather than a single prediction, which will be compared to a reference result from Hamiltonian Monte Carlo.

For the datasets listed as *reference datasets*, we will provide the HMC checkpoints and the corresponding predictive distributions to the participants. These datasets can be used to develop and calibrate the solutions. The datasets listed under *evaluation datasets* will be used to evaluate the submissions. The participants will submit the training scripts to produce the predictive distributions on these datasets. We will compare the predictive distributions to the private HMC checkpoints that we will not release publically until the end of the competition.

Table 2: **Baselines and example results.** The agreement and total variation metrics for the deep ensembles and SG-MCMC variations. The methods were trained on the CIFAR-10 train set, and we report the results on the original CIFAR-10 test set and the corrupted test sets from CIFAR-10-C. For CIFAR-10-C we report the mean and standard deviation of the metrics over the different corruptions and corruption intensities.

			SG-MCMC			
Dataset	Metric	Deep Ensembles	SGLD	SGHMC	SGHMC-CLR	SGHMC-CLR-Prec
CIFAR-10 Test	Agreement	91.7	91.8	92.2	**92.8**	**92.8**
	Total Variation	0.104	0.106	0.105	0.095	**0.092**
CIFAR-10-C	Agreement	80.1 ± 9.5	78.9 ± 10.9	79.9 ± 10.3	81.77 ± 8.8	$\mathbf{82.5 \pm 8.4}$
	Total Variation	0.204 ± 0.073	0.205 ± 0.076	0.193 ± 0.069	0.183 ± 0.064	$\mathbf{0.172 \pm 0.06}$

1.5 Metrics

The submissions will be evaluated based on the similarity of their predictive distribution to the predictive distribution approximated by a long run of multiple Hamiltonian Monte Carlo chains. Let us denote the target predictive distribution approximated by HMC for an input x by $\hat{p}(y|x)$, and let $p(y|x)$ be the predictive distribution from a submission to the competition.

For classification tasks, we will consider two primary metrics: *agreement* and *total variation*. Let $D_{test} = \{x_i\}_{i=1}^n$ be the test dataset. Then we define the agreement between $\hat{p}$ and p as the fraction of the test data points for which the top-1 predictions of $\hat{p}$ and p agree:

$$\text{agreement}(\hat{p}, p) = \frac{1}{n} \sum_{i=1}^{n} I[\arg\max_j \hat{p}(y = j|x_i) = \arg\max_j p(y = j|x_i)], \tag{1}$$

where $I[\cdot]$ is the indicator function. The agreement metric measures how well the submission is able to capture the top-1 predictions of the Bayesian model average. Higher is better.

We define the total variation metric between $\hat{p}$ and p as the total variation distance between the predictive distributions averaged over the test data points:

$$\text{TV}(\hat{p}, p) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \sum_j \left| \hat{p}(y = j|x_i) - p(y = j|x_i) \right|. \tag{2}$$

The total variation metric captures how well the full predictive distributions of $\hat{p}$ and p agree. In order to achieve a low total variation score, the submission has to capture not only the top-1 prediction of HMC, but also all of the class-probabilities. Lower is better.

For regression tasks, we will consider the Wasserstein-2 distance between $\hat{p}$ and p. Since p is provided as a set of sampled predictions for each example, the metric will be calculated

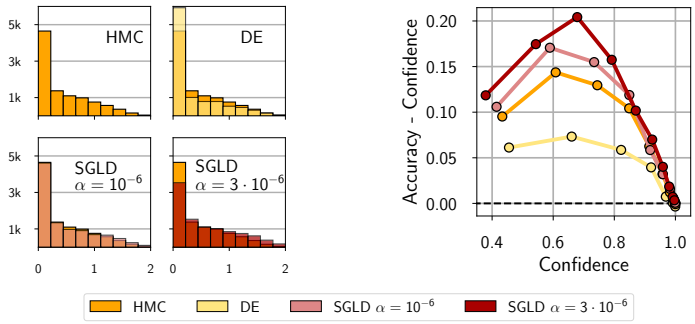

Figure 1: Distribution of predictive entropies (**left**) and calibration curve (**right**) of posterior predictive distributions for HMC, SGLD (with learning rates $\alpha = 10^{-6}$ and $3 \cdot 10^{-6}$), and deep ensemble on ResNet20-FRN on CIFAR-10. On the left, for all methods, except HMC we plot a pair of histograms: for HMC and for the corresdponding method. Deep ensembles provide more confident predictions than HMC, SGLD with high learning rate is undercondident, while SGLD with $\alpha = 10^{-6}$ matches HMC well.

as a point-wise W_2 distance:

$$W_2(\hat{p}, p) = \inf_I \sqrt{\sum_{i \in I, j} |p_i - \hat{p}_j|^2}, \tag{3}$$

where I are possible orderings of points. Lower is better.

In addition to these scores, the submissions will be constrained in training (and prediction) time (s): the time taken for the training script to perform approximate inference and return the full predictive distribution over the evaluation dataset. We require the training time to be no more than the equivalent of 3000 SGD training epochs. We will request the participants to provide the scripts for the several winning submissions, and run them on our hardware to estimate the training time.

Submissions will receive a performance score which is a weighted average of (1 − agreement), total variation, and Wasserstein distance averaged over the test problems.

To estimate the error-bars in the scores, we will split our HMC samples into several subsets, to form several targets to evaluate the submissions against. We will compute the scores using each of these targets and compute their standard deviation.

1.6 Baselines, code, and material provided

As baselines, we will provide Deep Ensembles, Mean-Field Variational Inference, Monte-Carlo Dropout and several variations of Stochastic Gradient MCMC (SG-MCMC) methods. For these methods we will provide the results and an implementation (starting kit) in the JAX framework. We will also provide an implementation (starting kit) of some of these

methods in the PyTorch framework. We provide preliminary results on the CIFAR-10 and CIFAR-10-C test sets for the deep ensemble and SG-MCMC baselines in Table 2. In Figure 1 we visualize the predictive entropy distribution and the calibration curve for HMC, deep ensembles and SGLD at different learning rates.

1.7 Tutorial and documentation

We describe our HMC implementation in detail, as well as baseline comparisons, at the following URL: `https://cims.nyu.edu/~andrewgw/bnnhmc.pdf` [Izmailov et al., 2021]. We will additionally provide a detailed documentation of the API for submissions to the competition, and tutorial resources on Bayesian deep learning. We provided a tutorial on Bayesian deep learning at ICML 2020 [Wilson, 2020].

2 Organizational aspects

2.1 Protocol

The participants will be provided with:

1. Links to publicly available datasets as well as data loading scripts in JAX and PyTorch frameworks in Python.

2. A list of reference architectures for each dataset and reference implementations in JAX and PyTorch frameworks in Python.

3. A reference implementation of a standard training procedure for each of the datasets and architectures in JAX and PyTorch frameworks in Python.

4. A reference implementation of the baselines listed in Section 1.6 in the JAX framework in Python.

5. HMC samples and the corresponding predictive distributions for the reference dataset-model pairs (see Section 1.4).

Although the test datasets for these publicly available datasets can be used by participants, they will not have access to the benchmark HMC samples against which their methods will be evaluated. Overfitting to the true test labels will not result in an improvement in the fidelity of the approximation of the "HMC label distributions".

The participants will submit the predictive distributions produced by their training scripts on each of the evaluation datasets. We will also request the authors of the winning (top 5) entries to submit their training scripts that receives the train and test dataset as input and produces the predictions on the test data; we will provide the participants with detailed instructions on how the training script needs to be structured, the format of the

inputs and outputs. We will run each of the provided training scripts on the same hardware with a time limit and evaluate the predictions. Hardware and time limits will be chosen at a later date based on available resources, but will aim to offer enough computation to enable a wide variety of methods while allowing researchers from a range of backgrounds to compete. We intend to provide participants with free access to cloud compute resources.

For each submission, we will record both the performance metrics and usability metrics. These will all be reported alongside method meta-data on a website associated with the competition (such as approximate inference method and key hyperparameters). Participants will be strongly encouraged to make their submissions open source under the MIT license, although we will allow submissions that only report the required structured meta-data.

To prevent cheating we will manually check the code submitted by the participants.

2.2 Rules

Draft of the rules:

- The goal of this competition is to accurately approximate the posterior predictive distribution on a range of neural network inference tasks, measured against accurate but expensive Hamiltonian Monte Carlo inference.

- Please submit the predictive distributions for the evaluation datapoints as a CSV file. For classification problems, the file should contain N rows and C columns, where N is the number of datapoints and C is the number of classes; the entry in position (i, c) should contain the predictive confidence for class c (softmax output, on the scale of 0 to 1) for the i-th datapoint. For regression problems, the file should contain N rows and 100 columns, where each line contains 100 samples from the predictive distribution for the corresponding datapoint.

- The authors of the winning submissions will be requested to submit their training scripts. The training script takes the training data as inputs saves and returns the predictions on the test data as outputs. We will provide detailed instructions later.

- The scripts have to run on `hardware` under the time limit of `time limit` (`hardware` and `time limit` will be decided later). Submissions will be evaluated on their training and inference time as described in Metrics.

- The scripts may not use any extra data or checkpoints not provided or produced by the scripts themselves, e.g. download extra data or use pre-computed checkpoints.

- The competition will be held in two separate tracks: *light track* and *extended track*. We invite all participants to take part in both tracks or just the light track of the competition if they prefer.

  - For the light track we will use the Diabetic Retinopathy dataset.
  - For the extended track we will use the Diabetic Retinopathy, CIFAR-10, UCI-Gap and MedMNIST datasets.

- Each participant can submit up to 3 CSV files with predictions at the evaluation stage. The participants will be scored based on their best submission.

- Optionally, you can also submit the CSV files produced by your scripts run locally on any of the reference datasets. The submissions will be scored using the same metrics that are used for the final evaluation and registered on a public leaderboard. The number of submissions for the public leaderboard is unlimited.

The rules ensure that the training scripts submitted by the participants will produce the predictive distribution based on the train and test sets provided as inputs. To prevent cheating, we will manually check the submissions to make sure that they follow the rules. We intend to use CodaLab to host the competition.

2.3 Schedule and readiness

Competition time line:

1. **By July 1:** we will collect HMC checkpoints on the reference and evaluation datasets. We will also set up and test the submission system on CodaLab.

2. **July 15:** We open the submission system and announce the beginning of the competition.

3. **Oct 15:** The evaluation phase starts. The participants are invited to submit their training scripts.

4. **Oct 31:** The evaluation stage ends.

5. **Nov 15:** We announce the results.

6. **Dec 13-14:** Winning participants are invited to present at the conference.

2.4 Competition promotion

The organizers have extensive experience successfully hosting, administrating, and advertising highly attended workshops. There are several mailing lists we will use to circulate the call for the competition, such as the NYU CS, Math, and Data Science lists, MILA, the Cornell MLDG list, Connectionists, ML News, CMU MLD, Oxford CS and statistics, Google Research, DeepMind, Microsoft Research. We are also active on social media, with collectively more than $50,000$ followers are Twitter, for example. We will additionally

build a website for the competition and circulate it widely. We will make a particular effort to reach out to affinity groups such as Black in AI, LatinX in AI, Women in Machine Learning, and Queer in AI.

3 Resources

3.1 Organizing team

Andrew Gordon Wilson (coordinator, evaluator) is an Assistant Professor in the Courant Institute of Mathematical Sciences and Center for Data Science at New York University. Andrew works on methodology for Bayesian deep learning, with a focus on scalable and reliable representations of uncertainty. He has applied his work to active learning for malaria incidence forecasting, understanding the effects of vaccine introduction for measles prevalance, and uncertainty representations for decision making in autonomous driving. Andrew provided an ICML 2020 tutorial on Bayesian deep learning. Andrew has also organized ten highly attended workshops across ICML and NeurIPS on Bayesian deep learning, generative modelling, and interpretable machine learning.

Yarin Gal (coordinator, evaluator) is an Associate Professor of Machine Learning at the University of Oxford holding positions also as a Tutorial Fellow in Computer Science at Christ Church college, Oxford, and a Turing AI Fellowship at the Turing Institute. He obtained his PhD from the Machine Learning group at the University of Cambridge, and was a Research Fellow at St Catherine's college, Cambridge. Yarin works on Bayesian deep learning with applications in automotive and medical, is the lead organiser of the Bayesian deep learning workshops at NeurIPS, and co-organiser of the uncertainty track at the Brain Tumor Segmentation Challenge at MICCAI.

Yingzhen Li (coordinator, beta tester) is a Lecturer (equiv. US assistant professor) in Machine Learning at the Department of Computing, Imperial College London, UK. Before that she was a senior researcher at Microsoft Research Cambridge, and previously she has interned at Disney Research. She received her PhD in engineering from the University of Cambridge. She has worked extensively on approximate inference methods with applications to Bayesian deep learning and deep generative models, and her work has been applied in industrial systems and implemented in deep learning frameworks (e.g. Tensorflow Probability and Pyro). She gave an invited tutorial on Advances in Approximate Inference at NeurIPS 2020. She was a co-organiser of the Advances in Approximate Bayesian Inference (AABI) symposium in 2020/2021, NeurIPS 2020 Bayesian Deep Learning meet-up, and ICLR 2021 workshop on Neural Compression.

Pavel Izmailov (coordinator, data provider, baseline method provider) is a fourth year PhD student at New York University working with Andrew Gordon Wilson. Pavel is working on Bayesian deep learning with a focus on scalability, uncertainty estimation and the fidelity of the posterior approximation. Pavel will provide the baseline method implementations and participate in the evaluation of the submissions, as well as administrate

and beta-test the submission platform.

Matthew D. Hoffman (coordinator, evaluator, platform administrator) is a research scientist at Google. His research focuses on probabilistic modeling and inference algorithms, with a particular interest in methods that can take advantage of modern software and hardware systems. Prior to joining Google, he was a research scientist at Adobe Research and a postdoc in the Statistics Department at Columbia University, where he was part of the team that developed the statistical modeling and inference package Stan. He has a Ph.D. in Computer Science from Princeton University.

Sebastian Farquhar (coordinator, evaluator, beta tester) is a Junior Research Fellow-elect at Christ Church, University of Oxford and DPhil candidate in the OATML research group at the University of Oxford. His work focuses on approximate inference methods for Bayesian deep learning including application-centred evaluations of inference quality and methods for assessing the performance cost of specific approximation choices. He co-organized the Bayesian Deep Learning meetup at NeurIPS 2020.

Melanie F. Pradier (coordinator, evaluator, beta tester) is a Research Scientist at Microsoft Research (Cambridge, UK). Until 2020, Melanie was a Postdoctoral Fellow at Harvard University working on probabilistic models for healthcare applications. She received her PhD on Bayesian nonparametrics from University Carlos III in 2017. Melanie co-organized the NeurIPS 2020 workshop "I can't believe it's not better!", as well as the workshop "Big data in human genetics: opportunities and challenges?" at the European Society of Human Genomics in 2016. Melanie has also worked at Sony EU Research Center in Stuttgart, Sony Corporation R&D in Tokyo, and the Memorial Sloan-Kettering Cancer Center in New York.

Andrew Foong (evaluator, beta tester) is a PhD candidate at the University of Cambridge. His work focuses on the intersection of probabilistic modelling and deep learning, with work on Bayesian neural networks, meta-learning, and modelling equivariance.

Sanae Lotfi (coordinator, evaluator, beta tester, platform administrator) is a PhD candidate at New York University. Her research interests include generalization in deep learning, optimization, the geometric structure of loss landscapes, and Bayesian methods.

Sharad Vikram (coordinator, platform administrator, baseline method provider) is a researcher at Google. He is working on probabilistic machine learning, focusing on Bayesian modeling and inference at scale, along with probabilistic programming. He completed his Ph.D. in Computer Science at U.C. San Diego. Sharad will particularly work on the HMC baseline samples and the submission platform.

Overall, this is a highly diverse team, spanning multiple institutions across both academia and industry research, junior and senior researchers, multiple ethnicities and genders, and complementary expertise and backgrounds.

3.2 Resources provided by organizers, including prizes

- **Community**: We will provide a public form to facilitate communication between participants and organizers.
- **Computing resources**: We will provide computing resources necessary for evaluating all inference methods in the reference phase of the competition, and running and evaluating all methods in the evaluation phase. We will also provide tutorial resources on approximate inference and Bayesian deep learning.
- **Prize**: We will provide a first prize of $2000 and a second prize of $500. We will also invite and advertise paper submissions for the three winning entries.

References

Angelos Filos, Sebastian Farquhar, Aidan N Gomez, Tim G J Rudner, Zachary Kenton, Lewis Smith, Milad Alizadeh, Arnoud de Kroon, and Yarin Gal. Benchmarking bayesian deep learning with diabetic retinopathy diagnosis. 2019.

Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between'uncertainty in bayesian neural networks. *arXiv preprint arXiv:1906.11537*, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *ICLR*, March 2019.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are Bayesian neural network posteriors really like? 2021. URL `http://cims.nyu.edu/~andrewgw/bnnhmc.pdf`.

Emtiyaz Khan. Deep learning with Bayesian principles, 2019. URL `https://www.youtube.com/watch?v=2wFb46Q8kmA`.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. 2017.

Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. 7(1), 2017.

David JC MacKay. Probable networks and plausible predictions?a review of practical Bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505, 1995.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. 2019.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

R.M. Neal. *Bayesian Learning for Neural Networks*. Springer Verlag, 1996. ISBN 0387947248.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. 2019.

Dustin Tran, Jasper Snoek, and Balaji Lakshminarayanan. Practical uncertainty estimation and out-of-distribution robustness in deep learning, 2020. URL `https://slideslive.com/38935801/practical-uncertainty-estimation-outofdistribution-robustness-in-deep-learning`.

Andrew Gordon Wilson. Bayesian deep learning and a probabilistic perpsective of model construction, 2020. URL `https://www.youtube.com/watch?v=E1qhGw8QxqY`.

Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. *arXiv preprint arXiv:2010.14925*, 2020.

Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of uncertainty quantification for bayesian neural network inference. *arXiv preprint arXiv:1906.09686*, 2019.