A General Framework for Ensemble Distribution Distillation

Jakob Lindqvist, Amanda Olmin, Fredrik Lindsten and Lennart Svensson

2020 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 21–24, 2020, ESPOO, FINLAND

A GENERAL FRAMEWORK FOR ENSEMBLE DISTRIBUTION DISTILLATION

Jakob Lindqvist[*†] *Amanda Olmin*[*‡] *Fredrik Lindsten*[‡] *Lennart Svensson*[†]

[†] Chalmers University of Technology, Department of Electrical Engineering, Gothenburg, Sweden
{jakob.lindqvist, lennart.svensson}@chalmers.se
[‡] Linköping University, Department of Computer and Information Science, Linköping, Sweden
{amanda.olmin, fredrik.lindsten}@liu.se

ABSTRACT

Ensembles of neural networks have shown to give better predictive performance and more reliable uncertainty estimates than individual networks. Additionally, ensembles allow the uncertainty to be decomposed into aleatoric (data) and epistemic (model) components, giving a more complete picture of the predictive uncertainty. Ensemble distillation is the process of compressing an ensemble into a single model, often resulting in a leaner model that still outperforms the individual ensemble members. Unfortunately, standard distillation erases the natural uncertainty decomposition of the ensemble. We present a general framework for distilling both regression and classification ensembles in a way that preserves the decomposition. We demonstrate the desired behaviour of our framework and show that its predictive performance is on par with standard distillation.

Index Terms— Ensemble, distillation, uncertainty

1. INTRODUCTION

Recently, there has been a surge of effort in modelling and estimating the uncertainty in deep neural networks, e.g. [1, 2, 3, 4]. For applications ranging from autonomous vehicles to medical image-analysis, reliable uncertainty estimates are vital. To understand the predictive uncertainty we can decompose it into model, or *epistemic*, uncertainty and inherent, *aleatoric*, noise in the data. This decomposition provides a more complete picture of the uncertainty quantification and is beneficial in applications such as active learning and reinforcement learning.

Ensembles of neural networks have shown to improve model performance and to make predictions more robust [5] as well as to consistently provide good uncertainty estimates. The epistemic uncertainty is naturally characterised in an ensemble as the spread of the predictions. Indeed, since the members are trained in an identical manner, disagreement on a given prediction means that the *model* is uncertain about that prediction.

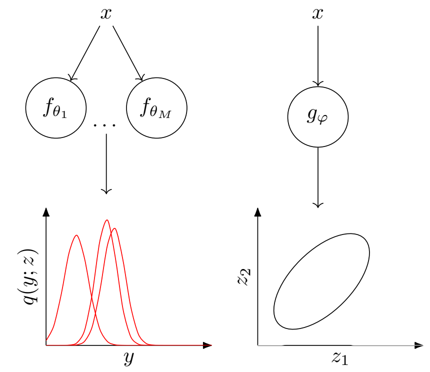

Fig. 1. Schematic view of the general distribution distillation. Here, the output data is modelled with $y|x \sim \mathcal{N}(z(1), \log(1 + e^{z(2)}))$. The ensemble produces several plausible predictive distributions (*left*). The distilled model mimics this by learning a *distribution* over the parameters $[z(1), z(2)]$ that captures the epistemic uncertainty in the model (*right*).

Ensemble state-of-the-art performance on *out of distribution* (*OOD*) data, is attributed to the ability to estimate epistemic uncertainty [6]. However, ensembles are expensive to use at test time, both in terms of memory and computations. It is therefore natural to consider some form of model compression that preserves the rich uncertainty description of the ensemble.

Ensemble distillation is a compression procedure where a distilled network learns to approximate the predictions of an ensemble. The final model is often more robust and performs better than a single network trained on the same data [7]. The drawback of standard ensemble distillation (as done, e.g., by [7]) is that it only considers the mean prediction of the ensemble and thereby the uncertainty decomposition is lost.

To also capture the spread of the ensemble, we pro-

[*] Equal contribution.

978-1-7281-6662-9/20/$31.00 ©2020 IEEE

pose to learn a *distribution* over the ensemble predictions. Instead of mimicking the task of the ensemble, the training objective of the distilled network will be to predict the parameters of this distribution. See fig. 1 for a schematic illustration.

Recently, a special case of this approach was proposed for classification problems, using a Dirichlet distribution to model ensemble predictions [1]. Here we present a general framework for ensemble distribution distillation of both classification and regression models, as well as other predictive models. Our framework is more generally applicable than previous works and allows for greater flexibility in the description of the ensemble.

2. BACKGROUND

Probabilistic predictive models Given a set of pairs of inputs and targets $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, a probabilistic predictive model approximates the true conditional probability distribution $p(y|x, \mathcal{D})$, with $q(y; f_\theta(x))$, where q belongs to some family of distributions parameterised by f_θ. In this paper, $z = f_\theta(x)$ is the output of a neural network that maps x to a parameter vector z for $q(y; z)$.

The network parameters θ are optimised in order to maximise the likelihood of data with respect to $q(y; f_\theta(x))$. In practice we minimise the negative logarithm of the likelihood (*NLL*),

$$\mathcal{L}(\theta) = -\mathbb{E}_{p(x,y)}\left[\log q(y; z = f_\theta(x))\right]. \qquad (1)$$

Uncertainty quantification The uncertainty in a model's prediction can be characterised using the estimated conditional probability $q(y; f_\theta(x)) \approx p(y|x)$. However, when reasoning about the uncertainty it is useful to distinguish between epistemic uncertainty in the model parameters θ and aleatoric noise in the data [2].

For a fixed value of θ, the model $q(y; f_\theta(x))$ will only capture aleatoric uncertainty. Conceptually, we can address this limitation with a Bayesian approach, learning a posterior distribution over the model parameters $p(\theta|\mathcal{D})$ and expressing the predictive distribution for a data point x^* as

$$p(y^*|x^*, \mathcal{D}) = \int \underbrace{p(y^*|x^*, \theta)}_{\text{aleatoric}} \underbrace{p(\theta|\mathcal{D})}_{\text{epistemic}} d\theta. \qquad (2)$$

More specifically, we can use this approach to define the different types of uncertainty:

$$\text{Total: } U_{\text{tot}} = I\left[p(y|x, \mathcal{D})\right], \qquad (3a)$$

$$\text{Aleatoric: } U_{\text{ale}} = \mathbb{E}_{p(\theta|\mathcal{D})}\left(I\left[p(y|x, \theta)\right]\right), \qquad (3b)$$

$$\text{Epistemic: } U_{\text{epi}} = U_{\text{tot}} - U_{\text{ale}}, \qquad (3c)$$

where I is some uncertainty measure, such as variance, entropy or differential entropy.

Ensembles Computing the posterior distribution over model parameters $p(\theta|\mathcal{D})$ is intractable in most cases when f_θ is given by a deep neural network. Although many approximate Bayesian methods have been proposed, e.g. [8, 9], a simple alternative is to use an ensemble of networks. This has been found to have very competitive empirical performance [6, 5].

Training an ensemble with M members means that we train M networks independently, resulting in M identically distributed models $\{f_{\theta_j}\}_{j=1}^M$. To ensure diversity in the ensemble, random initialisation of the same network architecture and randomly sampled mini-batches are commonly considered enough.

In addition to increased performance, ensembles also provide a natural estimate of the epistemic uncertainty. Specifically, we can use the spread of the ensemble, $\frac{1}{M}\sum_{j=1}^M \delta_{\theta_j}(\theta)$ as a plug-in replacement of the Bayesian posterior $p(\theta \mid \mathcal{D})$ in (3) to compute the different types of uncertainties (previously explored, e.g., by [2, 5, 10, 1]).

Ensemble distillation Because of their memory usage and computational cost at test time, ensembles are good targets for model distillation [11, 7]. In ensemble distillation, a single, distilled, model $g_\varphi(x)$ is trained to mimic the predictions made by the ensemble, after which the ensemble itself can be discarded.

Ensemble distillation is most prevalent in classification, where the ensemble members $\{f_{\theta_j}\}_{j=1}^M$ each predict a probability vector over classes, $p_j = f_{\theta_j}$. The distilled model g_φ is also trained as a classifier using cross-entropy loss, but with the "soft targets" $\overline{p} = \frac{1}{M}\sum_j p_j$, rather than the hard targets y. Distillation of regression models has received comparatively little attention.

3. DISTRIBUTION DISTILLATION

In this section we first discuss an interpretation of "vanilla" distillation as a KL minimisation problem. We then propose a general framework for distilling the distribution over the ensemble in a way that preserves the possibility of uncertainty decomposition.

Distillation as KL minimisation The above approach for distilling ensembles of classification models using cross-entropy loss is equivalent to interpreting the ensemble as a mixture of categorical distributions $y \sim \text{Cat}(p)$ and to minimise the KL-divergence between the distilled model and the mixture,

$$\text{KL}\left[\text{Cat}(y; \overline{p}) \| \text{Cat}(y; g_\varphi(x))\right] = \overline{p}\ \log(g_\varphi) + C, \quad (4)$$

where $\overline{p}$ is the soft target from the ensemble.

Similarly, if we let both ensemble members and the distilled model parameterise some distribution over $y|x$ for a regression task, we can minimise the KL-divergence between the mixture of the predictive distributions described by the ensemble and the distilled model. For instance, if both the ensemble members and the distilled model are assumed to be Gaussian, we get

$$\mathrm{KL}\left[\frac{1}{M}\sum_{j=1}^{M}\mathcal{N}(y;z_j=f_{\theta_j}(x))\|\mathcal{N}(y;g_\varphi(x))\right], \quad (5)$$

with $g_\varphi(x) = [\mu_\varphi(x), \sigma^2_\varphi(x)]$. Recent works have also used the KL-divergence interpretation, [12] for classification and [13] for both classification and regression. We call this approach *mixture distillation*.

A general framework for distribution distillation The mixture distillation method captures the total uncertainty of the model but not the epistemic. To address this limitation we propose a new framework for distillation where the distilled network predicts a distribution over the parameters $\{z_j = f_{\theta_j}(x)\}_{j=1}^M$ produced by the ensemble. That is, the distilled network predicts parameters for a higher-order distribution v instead of the parameters for a distribution over the output as in mixture distillation. The distilled network, like the ensemble members, is trained by minimising a *NLL*, but where we use the output of the ensemble $\{z_j = f_{\theta_j}(x)\}_{j=1}^M$ as the target:

$$\mathcal{L}_{\mathrm{DD}}(\varphi) = -\,\mathbb{E}_{p(x)}\left[\frac{1}{M}\sum_{j=1}^{M}\log v(z_j|g_\varphi(x))\right]. \quad (6)$$

Note that the expectation is taken w.r.t. the marginal distribution over the inputs and that we use the ensemble output in place of a ground truth. Hence, the distillation process does not require annotated data.

A key property of the proposed framework is that it is generic and applies to more than one problem class, including both classification and regression. The generality of our framework is related to the fact that there is freedom in choosing the parameterisation of the predictive distribution $q(y;z)$. How z is interpreted can differ both between and within problem classes. For example, in a classification setting, z could represent either a soft-max-transformed probability vector or the untransformed vector in *logit* space. This is in contrast with the work by [1], that only considers distribution distillation for classification, for one choice of parameterisation.

Predictions and uncertainty quantification The advantage of our proposed distillation framework is that it produces a network which not only models the ensemble predictions but also its epistemic uncertainty, encoded in the distribution $v(z;g_\varphi(x))$. The distilled network can be used to make predictions through the marginal predictive distribution,

$$\tilde{q}(y;g_\varphi(x)) = \int q(y;z)v(z;g_\varphi(x))dz. \quad (7)$$

Similarly to (3), it can also be used for computing the total and aleatoric uncertainties,

$$\text{Total: } U_{\text{tot}} = I\left[\tilde{q}(y;g_\varphi(x))\right], \quad (8a)$$

$$\text{Aleatoric: } U_{\text{ale}} = \mathbb{E}_{v(z;g_\varphi(x))}(I\left[q(y;z)\right]). \quad (8b)$$

If the involved expectations are intractable we can approximate them by sampling. Let $z_t \sim v(z;g_\varphi(x))$, $t = 1,\ldots,T$ be independent draws from the distilled distribution. Then

$$U_{\text{tot}} \approx I\left[\frac{1}{T}\sum_{t=1}^{T}q(y;z_t)\right], \quad U_{\text{ale}} \approx \frac{1}{T}\sum_{t=1}^{T}I\left[q(y;z_t)\right].$$

The epistemic uncertainty is given by (3c).

4. EXPERIMENTS

We evaluate our proposed framework in both regression and classification settings.[1] Note that the purpose of the distillation is to compress the ensemble for efficiency. We expect that this compression comes at the price of a performance drop. Hence, the purpose of the illustration is not to show that a distilled model outperforms an ensemble, but rather that it has comparable performance at a fraction of the cost in memory and computation.

4.1. Regression

Regression is an under-explored topic in distillation. Here, we demonstrate how our framework can be used in that setting. First, we present regression distillation on a toy problem and then illustrate its performance on some real-world datasets.

Regression toy example The example data set is the same as used by [14] and is a sinusoidal curve with heteroscedastic noise

$$y(x) = \sin(x) + \varepsilon(x),\ \varepsilon \sim \mathcal{N}\left(0, \frac{0.15}{1+e^{-x}}\right). \quad (9)$$

An ensemble with $M = 10$ members, each member with a single hidden layer, predicts M normal distributions on the form $\mathcal{N}(y; z(1), \log(1 + e^{z(2)}))$. The ensemble is trained on $N = 1000$ pairs $\{(x_i, y_i)\}_{i=1}^N$ with x_i sampled uniformly on $[-3, 3]$. To illustrate behaviour on *OOD* data, we evaluate the ensemble on data sampled uniformly on the larger interval $[-5, 5]$.

The aleatoric and epistemic uncertainty is calculated according to (3b) and (3c), respectively, using variance as a measure of uncertainty. The average mean prediction and decomposed uncertainty are shown in fig. 2(a).

The ensemble is distilled to a single network, parameterising a diagonal normal distribution over $z =$

[1] Code available at github.com/jackonelli/ensemble_distr_distillation

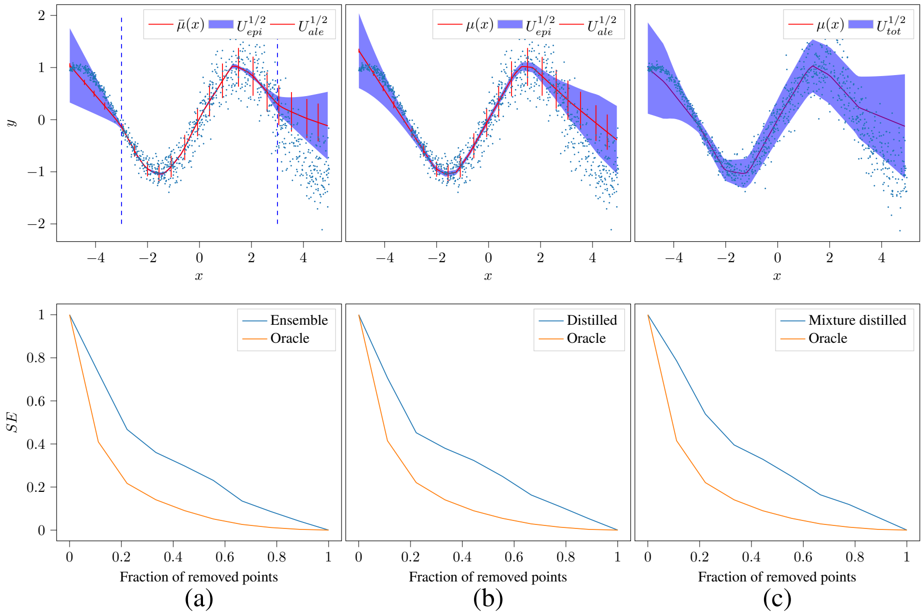

Fig. 2. **Top:** Mean prediction and uncertainty estimation on toy data in (9). (a) Ensemble, trained on data on the interval $[-3, 3]$. (b) Our framework. (c) Mixture distillation. Distilled networks are trained only on ensemble predictions on x sampled uniformly on $[-5, 5]$. Our framework preserves the uncertainty decomposition, whereas the mixture distillation only estimates the total uncertainty. **Bottom:** Sparsification plots of the toy data set for the respective models.

$[z(1), z(2)]$. This network has 2 hidden layers with 10 neurons each. Training is done on the ensemble predictions on inputs drawn from $U[-5, 5]$. We emphasize that the distillation training is unsupervised and does not require ground truth values y. In addition, we note that the distribution parameterised by the distilled network differs from that of the ensemble members, since it is a distribution over parameters and not over the output of the network. The distilled network is evaluated in the same way as the ensemble and the result is shown in fig. 2(b). The results indicate that our framework successfully distills the ensemble while retaining its rich uncertainty description.

For comparison, we also train a network with mixture distillation. We use the same architecture as for the distribution distillation, but optimise the KL-divergence in (5). In fig. 2(c) the distilled mean and total uncertainty are shown, but the uncertainty decomposition is no longer available. Sparsification error (*SE*) measure the error decay when the most uncertain data are removed [15]. *SE* plots in fig. 2 confirm that both distilled networks are able to capture the total uncertainty.

UCI data We use the UCI data [16] and perform an experiment with the setup described in [17]. We distill an ensemble of $M = 10$ networks. Individual ensemble members have a single hidden layer with 50 neurons. The distilled model has a single hidden layer of 75 neurons, trained only on ensemble predictions.

We measure root mean squared error (*RMSE*), *NLL* and area under sparsification error plot (*AUSE*) for both models and the results are compared in table 1. Each data set is split into 5 train-test folds for which both models are re-trained and tested. The ensemble consistently outperforms the distilled model, which is expected since the objective for the distillation is to mimic the ensemble. Still, the distilled model is performing well in all metrics, with confidence intervals computed over independent replications largely overlapping those of the ensemble.

Datasets	RMSE		NLL		AUSE	
	Ensemble	Distilled	Ensemble	Distilled	Ensemble	Distilled
concrete	8.08 ± 2.38	8.64 ± 1.82	3.47 ± 0.23	3.80 ± 0.14	0.36 ± 0.12	0.34 ± 0.08
wine	0.65 ± 0.02	0.65 ± 0.02	0.99 ± 0.01	1.05 ± 0.02	0.50 ± 0.02	0.58 ± 0.04
yacht	2.86 ± 0.26	3.42 ± 0.32	3.41 ± 0.11	4.30 ± 0.14	0.28 ± 0.02	0.34 ± 0.11
kin8nm	0.11 ± 0.02	0.12 ± 0.02	-0.72 ± 0.31	-0.27 ± 0.43	0.30 ± 0.04	0.37 ± 0.07
power plant	4.31 ± 0.20	4.33 ± 0.23	3.10 ± 0.19	3.67 ± 0.41	0.57 ± 0.06	0.64 ± 0.10

Table 1. Results on regression benchmark datasets comparing *RMSE*, *NLL* and *AUSE* for the ensemble and our distillation. Lower is better for all three metrics.

4.2. Classification

The presented framework is evaluated for classification on the CIFAR-10 dataset [18]. We include our model in the benchmark in [6] to measure accuracy and expected calibration error (*ECE*) [3] on *OOD* data.

The distilled model predicts a diagonal normal distribution $v(z; g_\varphi(x)) = \mathcal{N}(z; \mu_\varphi(x), \Sigma_\varphi(x))$ over ensemble logits $z = [z(1), \dots, z(K-1)]$, using class K as a reference class. We base the model on a ResNet architecture [19] with 20 layers[2] and train it using ensembles

[2] Based on code from https://github.com/kuangliu/pytorch-cifar/blob/master/models/resnet.py

of size $M = 10$ from [6]. The *OOD* data used in the experiments comes from applying 16 corruptions, such as Gaussian noise and changes to the contrast, with five levels of severity to CIFAR-10 test images [20].

In addition to comparing the performance of our model to that of the models constructed in [6], we train and include in the results one distribution distilled model parameterising a Dirichlet distribution according to [1] using a temperature annealing schedule for the soft-max function. We also include one mixture distilled model obtained with the KL-divergence objective in (4). The accuracy and *ECE* obtained with each model over the corrupted datasets and over five repeats are displayed in fig. 3. The *ECE* is a measure of misalignment between confidence and predictive accuracy, and is used to assess the validity of a model's uncertainty estimates.

In terms of *ECE*, our distribution distilled model performs comparable to the ensemble and is one of the best performing models on the corrupted data. Among the best-performing models, we also find the dropout model that bases it predictions on sampling by applying dropout during test time. In contrast to this model, our model requires only one forward pass through the network at test time.

The distribution distilled model has a slightly lower accuracy than the ensemble and the mixture distilled model, at least on in-distribution data. This indicates a trade-off between cost, in terms of computation and memory as well as a trade-off between the two objectives of estimating epistemic uncertainty and predicting the mean. However, our model is more cost-efficient than the ensemble and the ability to preserve the uncertainty decomposition of the ensemble proves valuable on *OOD* data. The same conclusion can be drawn from the performance of the Dirichlet distilled model.

5. DISCUSSION AND CONCLUSION

We highlight possible extensions to the presented framework and summarise the contributions of this paper.

Possible extensions The *NLL* in equation (6) allows for learning the two tasks of making predictions and representing uncertainty, but it lacks a way of adjusting the trade-off between the tasks. In addition, it does not offer any possibility of including the annotated data that do exist. It would be of relevance to investigate how model performance could benefit from changes to the loss function.

For classification we have parameterised z in $q(y; z)$ as logits. Using the standard parameterisation of q in terms of the probability vector instead, the distillation process can have difficulties in distinguishing small (but potentially important) differences in the class probabilities. The flexibility of alleviating this problem by performing the distillation directly in logit space as opposed to specifying a temperature annealing schedule as in [1] is of interest for further study.

Conclusions We have proposed a general framework for ensemble compression that maintains the rich description of predictive uncertainty, a key advantage of ensembles. Specifically, the compressed model estimates both epistemic and aleatoric uncertainty. Contrary to previous work, our framework applies to both regression and classification. We have demonstrated that this framework can result in compressed models with performance that is highly competitive with the state-of-the-art. Furthermore, compared to using a full ensemble, or other methods that are able to capture epistemic uncertainty (e.g. MC dropout or VI), our distilled model is simple and efficient to use at test time and has favorable storage cost.

ACKNOWLEDGMENTS

This work was supported by the Wallenberg Al, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

6. REFERENCES

[1] A. Malinin, B. Mlodozeniec, and M. Gales, "Ensemble Distribution Distillation," *arXiv:1905.00076*, Apr 2019.

[2] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *NeurIPS*, 2017.

[3] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *ICML*, 2017.

[4] D. Widmann, F. Lindsten, and D. Zachariah, "Calibration tests in multi-class classification: A unifying framework," in *NeurIPS*. 2019.

[5] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *NeurIPS*, 2017.

[6] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *NeurIPS*, 2019.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

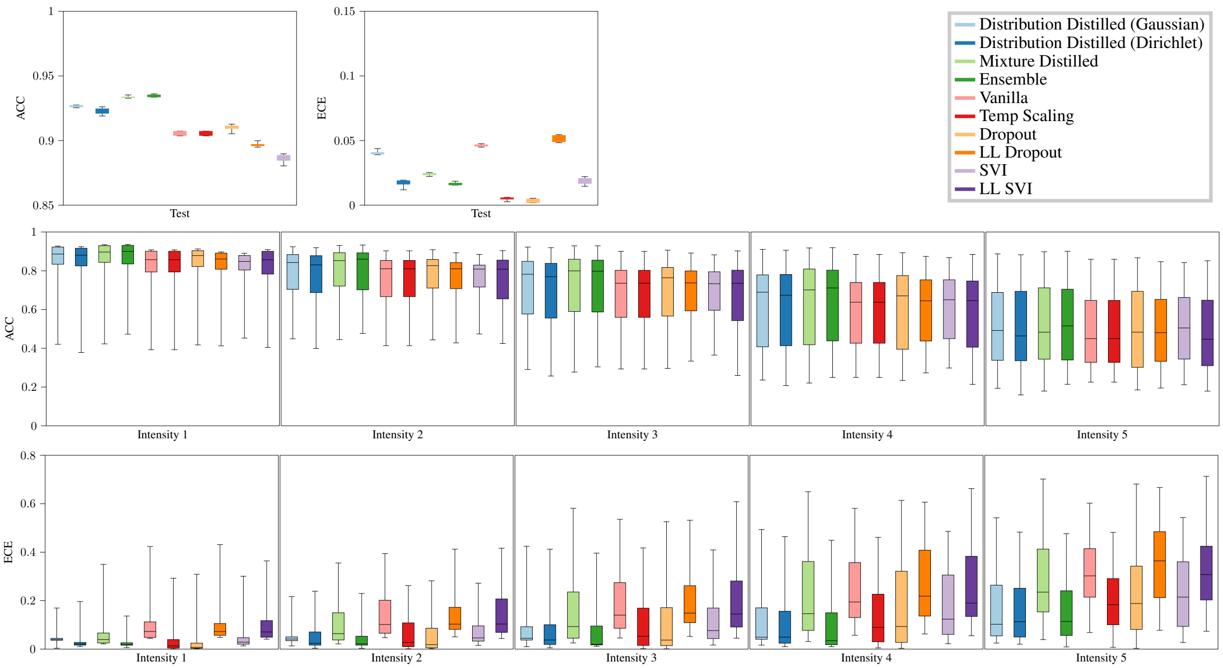

Fig. 3. Model accuracy and *ECE* across CIFAR-10 test data and *OOD* data consisting of CIFAR-10 data distorted with 16 different corruptions applied at an intensity scale ranging from 1 to 5. Boxes display minimum, maximum and median together with first and third quartiles of the accuracy and *ECE*, respectively. See [6] for details on the problem setup and explanations of the competing methods (from *Vanilla* to *LL SVI* in the list).

[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *ICML*, 2015.

[9] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016.

[10] A. Malinin and M. Gales, "Predictive Uncertainty Estimation via Prior Networks," in *NeurIPS*, 2018.

[11] C. Buciluă, R. Caruana, and A. Niculescu Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowledge discovery and data mining*, 2006.

[12] E. Englesson and H. Azizpour, "Efficient Evaluation-Time Uncertainty Estimation by Improved Distillation," in *ICML Workshops, Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

[13] L. Tran, B. S. Veeling, K. Roth, J. Swiatkowski, J. V. Dillon, J. Snoek, S. Mand t, T. Salimans, S. Nowozin, and R. Jenatton, "Hydra: Preserving Ensemble Diversity for Model Distillation," *arXiv:2001.04694*, Jan 2020.

[14] F. K. Gustafsson, M. Danelljan, and T. B. Schön, "Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision," *arXiv:1906.01620*, Jun 2019.

[15] C. Kondermann, R. Mester, and C. Garbe, "A statistical confidence measure for optical flows," in *ECCV*, 2008.

[16] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[17] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks," in *ICML*, 2015.

[18] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.

[20] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *ICLR*, 2019.