Supplementary material to
A Reproduction of Ensemble Distribution Distillation

Anonymous Author(s)
Affiliation
Address
email

Contents

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

A The EnD^2 Algorithm

The original paper features an excellent description of the mathematical formulation of the EnD^2 model, but we did not find it immediately obvious how to translate this into an implementation in a modern deep learning framework. For this reason, we will now briefly describe it from an algorithmic-centred perspective using pseudocode and plain English.

The process of training an EnD^2 model is described in Algorithm 1. In practice, the optimization in line 7 can easily be achieved using the standard "fit" method of frameworks such as Keras and PyTorch, by constructing an intermediate dataset and using a custom loss function with a callback for annealing the temperature.

The intermediate dataset is constructed by first adding any auxiliary images to the training images, and then passing the extended image set as input to the ensemble. The ensemble should output an array of logits as described in line 5 of Algorithm 1. The new dataset is then formed by matching each image to its corresponding ensemble logits, using the latter as the target.

The custom loss function is described in Algorithm 2. This formulation includes temperature annealing. This loss function is the only modification necessary to adapt a general classification model into an EnD^2 model, providing it is then trained on an intermediate dataset as described in the previous paragraph. Note that this formulation assumes that the model outputs logits. This output can be converted into Dirichlet probabilities by applying the standard softmax operation.

Algorithm 1: Training algorithm for EnD^2 given an ensemble

Input : Ensemble En outputting logits, training data X (same as the ensemble is trained on), (optional) Out of distribution data X_{OOD}

Output : Trained EnD^2 model

1 **if** X_{OOD} *not None* **then**
2 | $X = [X, X_{OOD}]$ // append OOD data to training set
3 **end**
4 $\phi = En.predict(X)$ // exp(ϕ) are the labels for EnD^2
5 // ϕ is a tensor of logits corresponding to the true distribution, each row corresponds to a model and each column a class. Each matrix corresponds to one image
6 $model_\theta \leftarrow classifier$ //create a new classifier model with weights θ, with logits as output
7 EnD^2 $= argmin_\theta\{Loss_{EnD^2}(\phi, model_\theta(X))\}$ //train model backpropagation
8 return EnD^2

Algorithm 2: loss for EnD^2

Input : Ensemble logits: ϕ, predicted logits: z, temperature: $T = T(t)$, annealing

Output : cost: C

1 $\epsilon = 10^{-8}$ // Smoothing factor
2 $\delta = 1 - 10^{-3}$ // Central smoothing factor
3 $\alpha = e^{z/T(t)}$ // elementwise exponential
4 M = #models
5 N = #classes
6 **for** $i \leftarrow 1$ **to** *M* **do**
7 | $\alpha_{0i} = \sum_j \alpha_{i,j}$ // sum over the classes to produce the precision factor
8 **end**
9 $P_{En} = softmax(\phi/T(t))$ // softmax over classes
10 $P_{En} = \delta(P_{En} - \frac{1}{N}) + \frac{1}{N}$
11 $TIT = \sum_i^N (log(\Gamma(\alpha_i + \epsilon))) - log(\Gamma(\alpha_0 + \epsilon))$ // target independent term, where $log(\Gamma(x)) = log((x-1)!)$
12 $A = \frac{1}{M} \sum_i^M (log(P_{En_i} + \epsilon)$ // mean over ensemble
13 $TDT = -\sum_i^N ((\alpha_i - 1)A_i)$ // target dependent term, sum over classes
14 return $(TDT + TIT)T(t)^2$

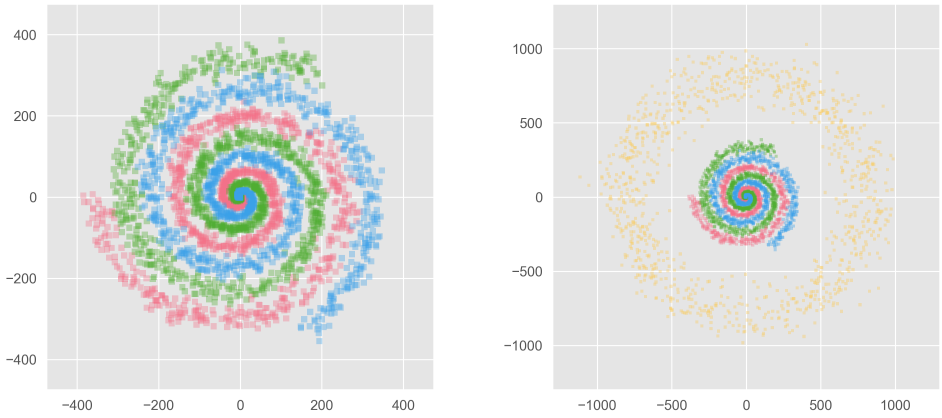


Figure 1: The synthetic, spiral dataset.

B Experiments on Synthetic Data

B.1 Methodology

The goal with these experiments is to provide qualitative justification for Claim 5 and illustrate the inner workings of EnD^2. We also provide some new experiments on temperature annealing and the size of the auxiliary dataset, to visualize their effect.

B.1.1 Dataset

To illustrate the model, Malinin et. al. use a synthetic dataset in $\mathbb{R}^2$. Our rendering of this dataset can be seen in Figure 1. This is advantageous since it enables plotting both knowledge and data uncertainty over the entire data manifold, giving a qualitative understanding of whether the algorithm works or not, in contrast to higher dimensional data (images, etc.) that cannot be plotted. The dataset itself looks like a spiral, divided into three classes shaped as spiralling arms of increasing radius. The spirals are centred and almost symmetric around the origin. Furthermore, they have increased noise and overlap with radius, which leads us to believe that uncertainty should vary as well. In addition to the spiral data an OOD data-set, referred to as the AUX data-set is also used, which takes the form of a ring slightly outside the spiral.

For the experiments, 1000 samples per ID class are used, both for training and test. The number of AUX samples was also 1000. This is the same setting as the original paper. The generation of the data uses the original paper's code, but the hyperparameters were not specified. Our hyperparameters can be found in our code. We manually searched for hyperparameters, so that our plot would look as close to theirs, but the exact correspondence is probably not achieved.

B.1.2 Model description and hyperparameters

The original paper does not specify what type of neural network was used for classification. We were also unable to find it in the (unofficial) code. Instead, we chose to use a simple DNN with four hidden layers, each of width 64 with ReLu-activation functions, trained by minimizing the categorical cross-entropy using the Adam-optimizer, all with standard `tf.keras` settings, for 85 epochs. EnD and EnD^2 used the same base model but was instead trained for 500 epochs.

B.1.3 Experimental setup and code

On the output of an ensemble of 100 models, all differently randomly initialized, we train EnD and EnD^2 both with and without auxiliary data, using an initial temperature of 1, as in the paper. Doing this, we observed that the training diverges for many initialisations, mainly for EnD$^2_{+\texttt{AUX}}$. Thus, we also used an initial temperature of $T = 2.5$, both with and without annealing. The annealing schedule was $T = 2.5$ between epoch 0 and 200, linearly decreasing to 1 between

Table 1: Classification error on Spiral Dataset, compared with [1]. Error bars are 95%-confidence intervals assuming normal distribution. Note that our results likely use a different base model and training procedure than the original paper, since it was not specified there.

ERR↓	IND	ENSM	EnD	EnD^2	EnD $_{\texttt{+AUX}}$	EnD^2 $_{\texttt{+AUX}}$	EnD^2 $_{\texttt{+AUX,ANN}}$	EnD^2 $_{\texttt{+AUX,T=2.5}}$	EnD^2 $_{\texttt{+AUX20}}$
Our results	8.20$\pm$0.67	2.3$\pm NA$	3.90$\pm$0.65	3.86$\pm$0.70	2.61$\pm$0.11	4.67$\pm$3.26	3.30$\pm$0.59	3.45$\pm$0.96	5.0$\pm$1.54
Paper [1]	13.21	12.37	12.39	12.47	12.41	12.40	-	-	-

epoch 200 and 400 and 1 between epoch 400 and 500. Additionally, we also trained a model EnD$^2_{\texttt{+AUX20}}$, with only 20 samples from the auxiliary dataset.

All 7 models were trained 20 times, with different random initialisations. To make sure they converged, the test error was calculated. In cases test error was above 10%, it was deemed as non-convergence, and not taken into account. Among the converged ones, the mean error and the 95%-confidence interval around the mean is calculated, assuming a normal distribution. This means that for cases with fewer samples, the confidence interval is larger.

The main goal of this experiment is to visually show the total uncertainty, the data uncertainty and the knowledge uncertainty. They were calculated as specified in [1] and [2], for the grid $[-2000, 2000] \times [-2000, 2000]$ at all coordinates divisible by four, for a total of 10^6 points.

The full code is available at https://anonymous.4open.science/r/4ee2c9ef-295f-44e2-8214-f0818b932817/.

B.1.4 Computational requirements

The experiments were run on the CPU of a normal laptop (2.7 GHz Dual-Core i5). The total time to reproduce the ensemble of 100 models and all 20 repetitions of all 7 tested distillation methods, is around 5 to 6 hours.

B.2 Results

B.2.1 Classification accuracy

In Table 1, the classification accuracy from our experiment and the original paper is reported. We see that

- the ensemble outperforms the individual models, and that all distillation methods perform closer to the ensemble, than an individual model.
- the best performance is achieved by EnD with auxiliary data.
- using annealing or not when starting at $T = 2.5$ does not affect the final classification accuracy.

B.2.2 Visualization of uncertainty

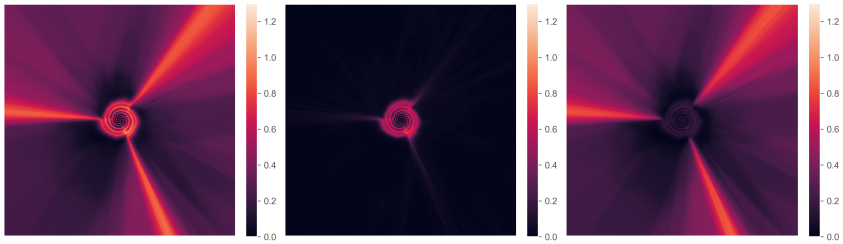
The total, data and knowledge uncertainty is plotted in Figure 2 for a grid of 10^6 points. In contrast to the original paper, we fix the scale of the colour bar for better comparability between plots.

We observe that

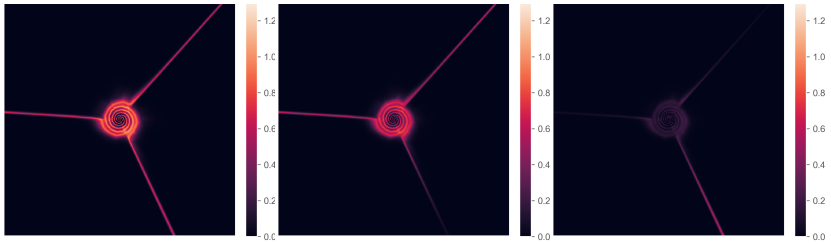
- EnD^2 is not able to emulate the uncertainty landscape of the ensemble, but EnD$^2_{\texttt{+AUX}}$ can approximate it fairly well.
- Starting at a higher temperature ($T = 2.5$) and using annealing produces similar results as starting at temperature 1, but starting at temperature 2.5 and keeping it there for the entire training duration does not capture the true uncertainty.
- Using a smaller auxiliary dataset gives a worse approximation of the ensemble's uncertainty landscape.

C Computational requirements for reproduction

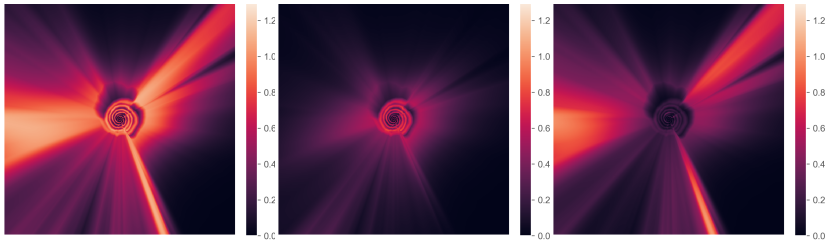
In this section, we report the computational resources used for this reproduction. The running time of the major experiments on CIFAR-10 is expressed in time on an RTX 2070. For easier comparison, we also report the equivalent cost when running on a V100 GPU on Google Cloud for $2.48 per hour, given a relative performance of 2.89 versus an



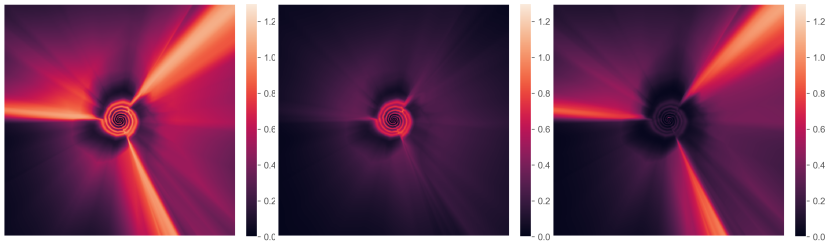
(a) Ensm. Tot. Unct. (b) Ensm. Data Unct. (c) Ensm. Know. Unct.



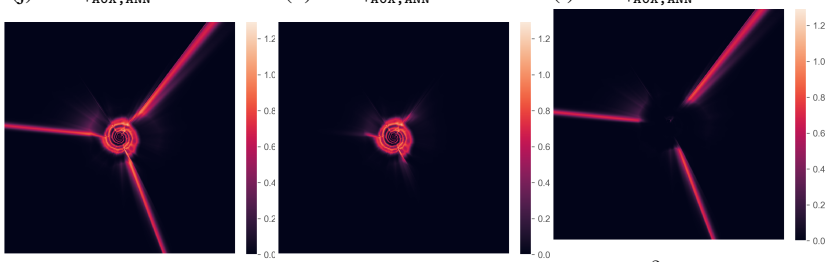
(d) EnD^2 Tot Unct. (e) EnD^2 Data Unct. (f) EnD^2 Know. Unct.



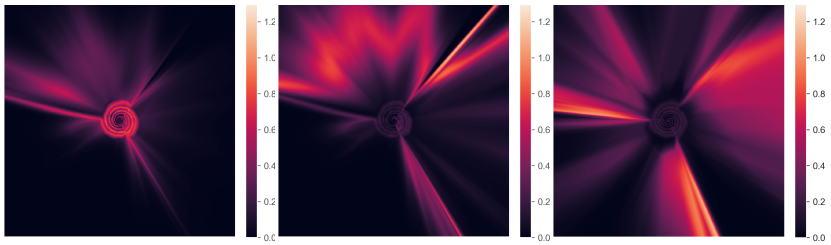
(g) EnD$^2_{\texttt{+AUX}}$ Tot Unct. (h) EnD$^2_{\texttt{+AUX}}$ Data Unct. (i) EnD$^2_{\texttt{+AUX}}$ Know. Unct.



(j) EnD$^2_{\texttt{+AUX,ANN}}$ Total Unct. (k) EnD$^2_{\texttt{+AUX,ANN}}$ Data Unct. (l) EnD$^2_{\texttt{+AUX,ANN}}$ Know. Unct.



(m) EnD$^2_{\texttt{+AUX,T=2.5}}$ Tot. Unct. (n) EnD$^2_{\texttt{+AUX,T=2.5}}$ Data Unct. (o) EnD$^2_{\texttt{+AUX,T=2.5}}$ Know. Unct.



(p) EnD$^2_{\texttt{+AUX20}}$ Tot. Unct. (q) EnD$^2_{\texttt{+AUX20}}$ Data Unct. (r) EnD$^2_{\texttt{+AUX20}}$ Know. Unct.

Figure 2: Recreation of Figure 3 in [1], showing uncertainties over entire data manifold.

Table 2: Computation requirements for major experiments, and which claims they test. GPU time refers to time on an NVIDIA GeForce RTX 2070. Equivalent cost represents the cost if run on a V100 on Google cloud, for $2.48 per hour.

Experiment	**Models**	**GPU minutes per model**	**GPU days**	**Equivalent cost in $**
Ensemble, training	400	16	4.44	91.53
Ensemble, labeling	400	0.45	0.13	2.57
Ensemble, inference	400	0.23	0.06	1.32
Evaluation, claim 1 and 2	15	51	0.53	10.94
Size ablation, training, claim 3	112	51	3.97	81.69
Temperature ablation, training, claim 4	27	51	0.96	19.69
3-class ensemble, training, claim 5	100	5.25	0.36	7.51
Total			11.413	235.06

Table 3: OOD ROC-AUC↑ on CIFAR-10 (in) and LSUN (out), normalized to ensemble results. Error bounds signify two standard deviations, taken over three models.

Unc.	IND	ENSM	EnD	EnD^2	$\text{EnD}_{\texttt{+AUX}}$	$\text{EnD}^2_{\texttt{+AUX}}$	$\text{PN}_{\texttt{+AUX}}$
Tot. our	$0.96_{\pm 0.00}$	$1.00_{\pm NA}$	$1.00_{\pm 0.01}$	$0.98_{\pm 0.00}$	$1.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$\mathbf{1.02}_{\pm 0.01}$
Tot. paper	$0.97_{\pm 0.01}$	$1.00_{\pm NA}$	$0.94_{\pm 0.01}$	$0.97_{\pm 0.01}$	$0.94_{\pm 0.01}$	$1.00_{\pm 0.01}$	$\mathbf{1.01}_{\pm 0.01}$
Know., our	-	$1.00_{\pm NA}$	-	$0.95_{\pm 0.01}$	-	$0.99_{\pm 0.01}$	$\mathbf{1.02}_{\pm 0.00}$
Know., paper	-	$1.00_{\pm NA}$	-	$0.98_{\pm 0.01}$	-	$0.99_{\pm 0.01}$	$\mathbf{1.01}_{\pm 0.01}$

RTX 2070[1]. Note that these figures represent the time to reproduce only the final experiments. We estimate that the total GPU time used for this reproduction, including experimentation and bug-hunting, to be 3 to 5 times as long. The full data can be seen in Table 2.

D Histograms

To compare ensembles, EnD^2 and $\text{EnD}^2_{\texttt{+AUX}}$ on the CIFAR-10 and 3-class CIFAR-10 datasets, we provide histograms of data and knowledge uncertainty for in- and out-of-domain-distribution, in Figure 3 and 4.

E Relative performance of EnD^2 compared to ensemble and original article

In Tables 3 and 4 of the main report we report several measures for the 7 different models tested. For better comparability, we here also provide the values normalized to the ensembles' performance, both for our experiments, and for the original paper, in Table 4 and 3.

[1]Benchmark taken from https://timdettmers.com/2020/09/07/which-gpu-for-deep-learning/

Table 4: Classification metrics on CIFAR-10, normalized to ensemble results. Error bounds signify two standard deviations, taken over three models.

Crit.	IND	ENSM	EnD	EnD^2	$\text{EnD}_{\texttt{+AUX}}$	$\text{EnD}^2_{\texttt{+AUX}}$	$\text{PN}_{\texttt{+AUX}}$
ERR↓, our	$1.12_{\pm 0.08}$	$1.00_{\pm NA}$	$\mathbf{0.99}_{\pm 0.06}$	$1.13_{\pm 0.02}$	$1.13_{\pm 0.02}$	$1.16_{\pm 0.01}$	$1.14_{\pm 0.04}$
ERR↓, paper	$1.29_{\pm 0.06}$	$\mathbf{1.00}_{\pm NA}$	$1.08_{\pm 0.05}$	$1.18_{\pm 0.03}$	$1.08_{\pm 0.03}$	$1.11_{\pm 0.06}$	$1.21_{\pm 0.10}$
PRR↑, our	$0.87_{\pm 0.02}$	$\mathbf{1.00}_{\pm NA}$	$0.98_{\pm 0.00}$	$0.96_{\pm 0.01}$	$0.98_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.70_{\pm 0.12}$
PRR↑, paper	$0.97_{\pm 0.01}$	$\mathbf{1.00}_{\pm NA}$	$0.98_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.98_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.94_{\pm 0.02}$
ECE↓, our	$41.37_{\pm 0.35}$	$1.00_{\pm NA}$	$\mathbf{0.94}_{\pm 0.05}$	$1.45_{\pm 0.13}$	$1.08_{\pm 0.19}$	$1.85_{\pm 0.29}$	$5.69_{\pm 0.37}$
ECE↓, paper	$1.69_{\pm 0.31}$	$1.00_{\pm NA}$	$2.00_{\pm 0.15}$	$\mathbf{0.77}_{\pm 0.15}$	$2.00_{\pm 0.46}$	$1.69_{\pm 0.31}$	$9.23_{\pm 0.54}$
NLL↓, our	$6.38_{\pm 0.04}$	$\mathbf{1.00}_{\pm NA}$	$1.06_{\pm 0.04}$	$1.35_{\pm 0.02}$	$1.19_{\pm 0.01}$	$1.38_{\pm 0.01}$	$1.86_{\pm 0.04}$
NLL↓, paper	$1.32_{\pm 0.05}$	$\mathbf{1.00}_{\pm NA}$	$1.16_{\pm 0.05}$	$1.32_{\pm 0.05}$	$1.16_{\pm 0.05}$	$1.26_{\pm 0.00}$	$2.00_{\pm 0.05}$

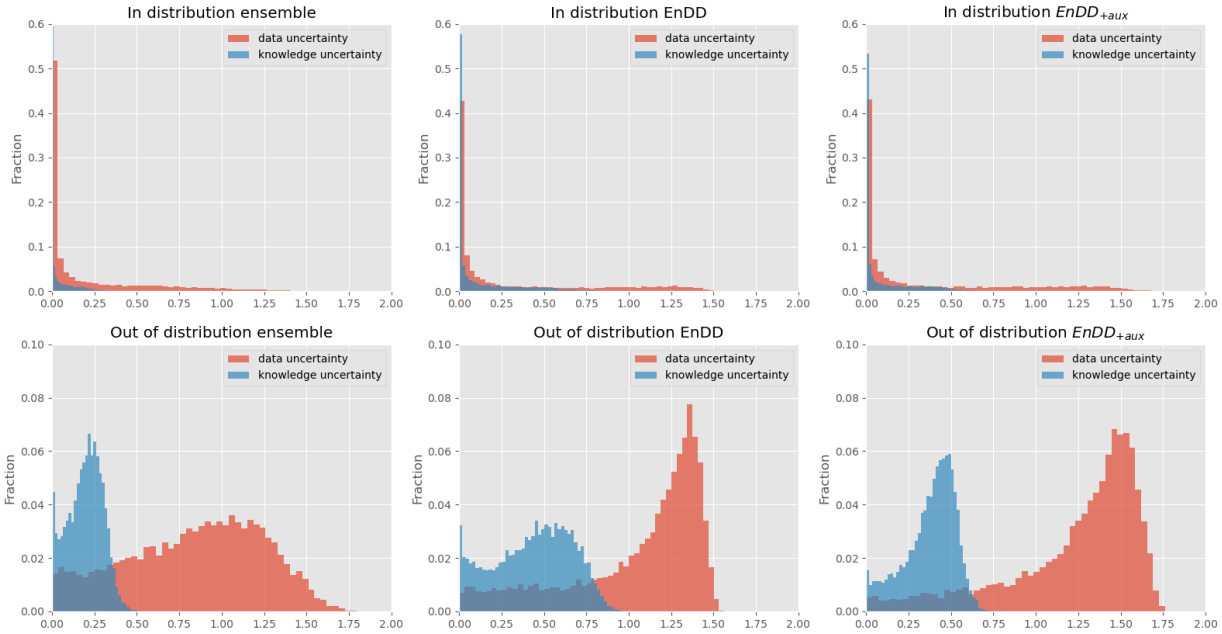


Figure 3: Data/knowledge uncertainty-distributions for ensemble, EnD^2 and EnD$^2_{\texttt{+AUX}}$.

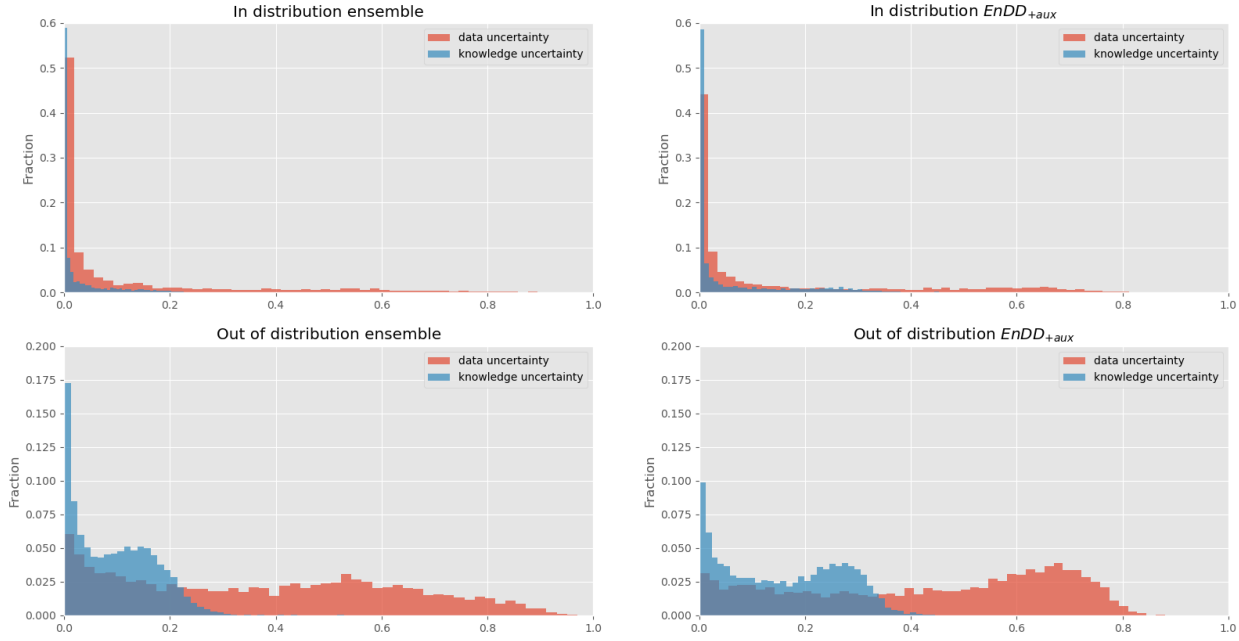


Figure 4: Data/knowledge uncertainty-distributions for ensemble and EnD$^2_{\texttt{+AUX}}$ on the 3-class CIFAR10 dataset

References

[1] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. "Ensemble Distribution Distillation". In: *International Conference on Learning Representations (ICLR)*. 2020.

[2] Andrey Malinin and Mark Gales. "Predictive Uncertainty Estimation via Prior Networks". In: *Advances in Neural Information Processing Systems 31*. 2018.