UPPSALA
UNIVERSITET

Ensembling as Approximate Bayesian Inference for Predictive Uncertainty Estimation in Deep Learning

Fredrik K. Gustafsson, Uppsala University
Martin Danelljan, ETH Zurich
Thomas B. Schön, Uppsala University

SSDL19
Norrköping, June 10, 2019

Outline

1. Introduction

> *We need to teach how doubt is not to be feared but welcomed. It's OK to say, "I don't know."*
>
> - Richard P. Feynman

- DNNs have become the go-to approach in computer vision, but generally fail to properly capture the uncertainty inherent in their predictions.
- Estimating this predictive uncertainty can be crucial, for instance in automotive and medical applications.

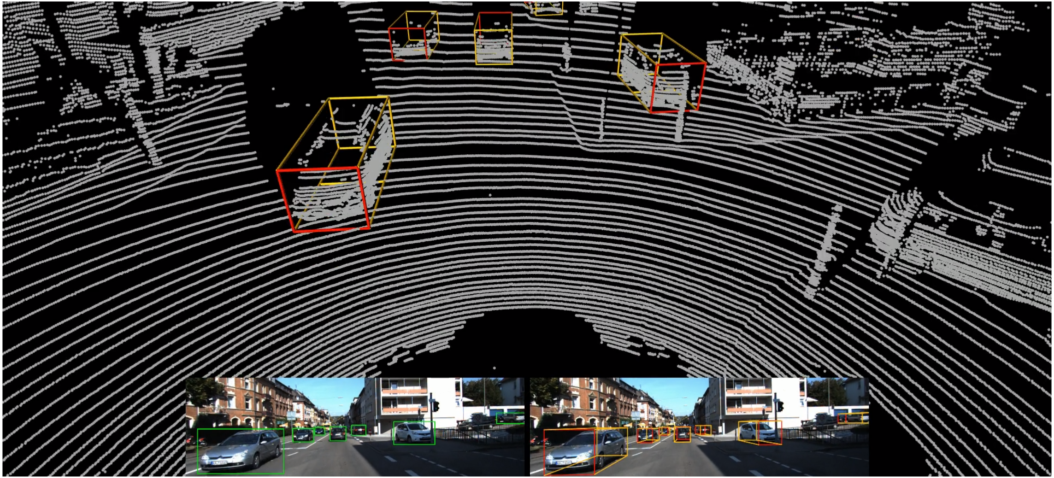
We need to teach how doubt is not to be feared but welcomed. It's OK to say, "I don't know."

\- Richard P. Feynman

- DNNs have become the go-to approach in computer vision, but generally fail to properly capture the uncertainty inherent in their predictions.
- Estimating this predictive uncertainty can be crucial, for instance in automotive and medical applications.
- **Bayesian deep learning** deals with predictive uncertainty by decomposing it into the distinct types of *aleatoric* and *epistemic* uncertainty.

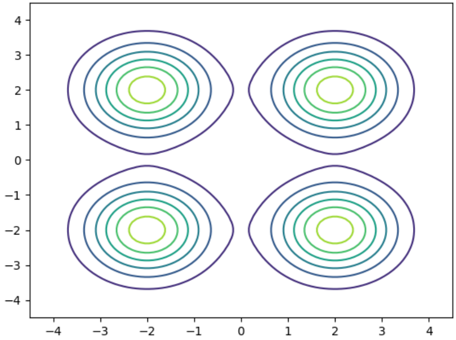
- **Aleatoric** uncertainty captures inherent and irreducible data noise.
- Input-dependent aleatoric uncertainty is present whenever we expect the estimated targets to be inherently more uncertain for some inputs.

- This is true *e.g.* in 3D object detection, where the estimated location of distant objects generally is expected be more uncertain.



- This is also true in semantic segmentation, where image pixels at object boundaries are inherently ambiguous.

- **Epistemic** uncertainty accounts for uncertainty in the DNN model parameters.
- Large epistemic uncertainty is present when a large set of model parameters explains the data (almost) equally well.



Outline

UPPSALA UNIVERSITET

The task is to predict a target value $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$. We are given a training set of i.i.d. sample pairs $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

We view a DNN as a function $f_\theta : \mathcal{X} \to \mathcal{U}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $x \in \mathcal{X}$ to an output $f_\theta(x) \in \mathcal{U}$.

2. Predictive uncertainty estimation using Bayesian deep learning

The task is to predict a target value $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$. We are given a training set of i.i.d. sample pairs $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

We view a DNN as a function $f_\theta : \mathcal{X} \rightarrow \mathcal{U}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $x \in \mathcal{X}$ to an output $f_\theta(x) \in \mathcal{U}$.

- Input-dependent **aleatoric** uncertainty can be estimated by:
  - Letting a DNN f_θ output the parameters of some probability distribution, creating a parametric model $p(y|x, \theta)$ of the conditional distribution.

The task is to predict a target value $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$. We are given a training set of i.i.d. sample pairs $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

We view a DNN as a function $f_\theta : \mathcal{X} \to \mathcal{U}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $x \in \mathcal{X}$ to an output $f_\theta(x) \in \mathcal{U}$.

- Input-dependent **aleatoric** uncertainty can be estimated by:
  - Letting a DNN f_θ output the parameters of some probability distribution, creating a parametric model $p(y|x, \theta)$ of the conditional distribution.
  - Finding the maximum-likelihood estimate of the model parameters, $\hat{\theta}_{\text{MLE}}$, by minimizing $-\log p(Y|X, \theta) = -\sum_{i=1}^{N} \log p(y_i|x_i, \theta)$.

UPPSALA UNIVERSITET

The task is to predict a target value $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$. We are given a training set of i.i.d. sample pairs $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

We view a DNN as a function $f_\theta : \mathcal{X} \rightarrow \mathcal{U}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $x \in \mathcal{X}$ to an output $f_\theta(x) \in \mathcal{U}$.

- Input-dependent **aleatoric** uncertainty can be estimated by:
  - Letting a DNN f_θ output the parameters of some probability distribution, creating a parametric model $p(y|x, \theta)$ of the conditional distribution.
  - Finding the maximum-likelihood estimate of the model parameters, $\hat{\theta}_{\text{MLE}}$, by minimizing $-\log p(Y|X, \theta) = -\sum_{i=1}^{N} \log p(y_i|x_i, \theta)$.
- Given $x^\star$ at test time, the DNN predicts the distribution $p(y^\star|x^\star, \hat{\theta}_{\text{MLE}})$ over $y^\star$.

- In **classification**, a categorical model is commonly used:

$$p(y|x, \theta) = \mathrm{Cat}(y; s_\theta(x)), \quad s_\theta(x) = \mathrm{Softmax}(f_\theta(x)). \tag{1}$$

- In **classification**, a categorical model is commonly used:

$$p(y|x,\theta) = \mathrm{Cat}(y; s_\theta(x)), \quad s_\theta(x) = \mathrm{Softmax}(f_\theta(x)). \tag{1}$$

  - $-\log p(Y|X,\theta)$ corresponds to the standard cross-entropy loss.

- In **classification**, a categorical model is commonly used:

$$p(y|x,\theta) = \mathrm{Cat}(y; s_\theta(x)), \quad s_\theta(x) = \mathrm{Softmax}(f_\theta(x)). \tag{1}$$

  - $-\log p(Y|X,\theta)$ corresponds to the standard cross-entropy loss.

- In **regression**, a Gaussian model can be used (1D case):

$$p(y|x,\theta) = \mathcal{N}\left(y; \mu_\theta(x), \sigma_\theta^2(x)\right), \quad f_\theta(x) = [\,\mu_\theta(x) \quad \log\sigma_\theta^2(x)\,]^\mathsf{T} \in \mathbb{R}^2. \tag{2}$$

2. Predictive uncertainty estimation using BDL - Aleatoric uncertainty

- In **classification**, a categorical model is commonly used:

$$p(y|x,\theta) = \mathrm{Cat}(y; s_\theta(x)), \quad s_\theta(x) = \mathrm{Softmax}(f_\theta(x)). \tag{1}$$

  - $-\log p(Y|X,\theta)$ corresponds to the standard cross-entropy loss.

- In **regression**, a Gaussian model can be used (1D case):

$$p(y|x,\theta) = \mathcal{N}\big(y; \mu_\theta(x), \sigma^2_\theta(x)\big), \quad f_\theta(x) = [\, \mu_\theta(x) \quad \log \sigma^2_\theta(x) \,]^\mathsf{T} \in \mathbb{R}^2. \tag{2}$$

  - $-\log p(Y|X,\theta)$ corresponds to the following loss:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{(y_i - \mu_\theta(x_i))^2}{\sigma^2_\theta(x_i)} + \log \sigma^2_\theta(x_i).$$

- **Epistemic** uncertainty can be estimated in a principled manner by performing Bayesian inference.

2. Predictive uncertainty estimation using BDL - Epistemic uncertainty

UPPSALA UNIVERSITET

- **Epistemic** uncertainty can be estimated in a principled manner by performing Bayesian inference. The posterior $p(\theta|\mathcal{D}) \propto p(Y|X,\theta)p(\theta)$ is then utilized to obtain the predictive posterior distribution:

$$p(y^\star|x^\star,\mathcal{D}) = \int p(y^\star|x^\star,\theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{M}\sum_{i=1}^{M} p(y^\star|x^\star,\theta^{(i)}), \quad \theta^{(i)} \sim p(\theta|\mathcal{D}),$$

which captures both **aleatoric** and **epistemic** uncertainty.

- **Epistemic** uncertainty can be estimated in a principled manner by performing Bayesian inference. The posterior $p(\theta|\mathcal{D}) \propto p(Y|X,\theta)p(\theta)$ is then utilized to obtain the predictive posterior distribution:

$$p(y^\star|x^\star,\mathcal{D}) = \int p(y^\star|x^\star,\theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{M}\sum_{i=1}^{M} p(y^\star|x^\star,\theta^{(i)}), \quad \theta^{(i)} \sim p(\theta|\mathcal{D}),$$

  which captures both **aleatoric** and **epistemic** uncertainty.

- In practice, an approximate posterior $q(\theta) \approx p(\theta|\mathcal{D})$ has to be used, resulting in:

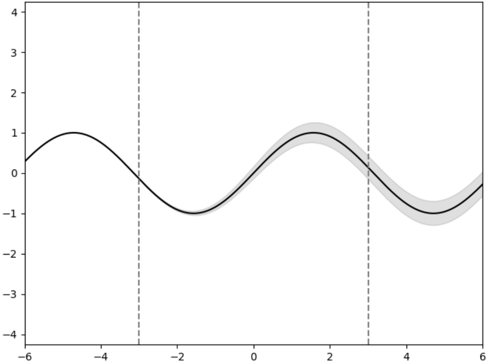
$$\hat{p}(y^\star|x^\star,\mathcal{D}) \triangleq \frac{1}{M}\sum_{i=1}^{M} p(y^\star|x^\star,\theta^{(i)}), \quad \theta^{(i)} \sim q(\theta). \tag{3}$$

Outline

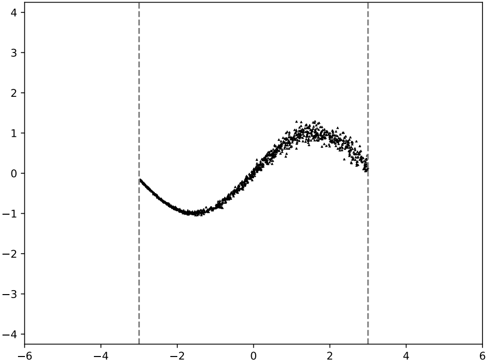
UPPSALA UNIVERSITET

We consider the following 1D regression problem:

$$y \sim \mathcal{N}\big(\mu(x), \sigma^2(x)\big), \qquad \mu(x) = \sin(x), \qquad \sigma(x) = \frac{0.15}{1 + e^{-x}}.$$



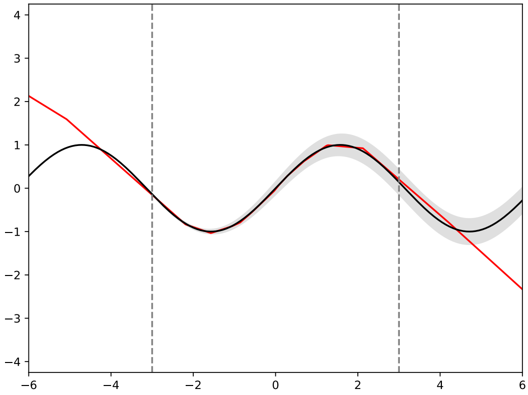
(a) True data generator.



(b) Training dataset.

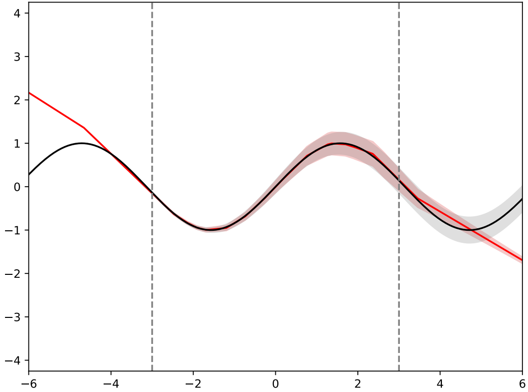
UPPSALA UNIVERSITET

- A DNN trained to directly predict targets, $y^\star = f_{\hat{\theta}}(x^\star)$, via the L^2 loss is able to regress the mean for $x^\star \in [-3, 3]$, but fails to capture any notion of uncertainty:

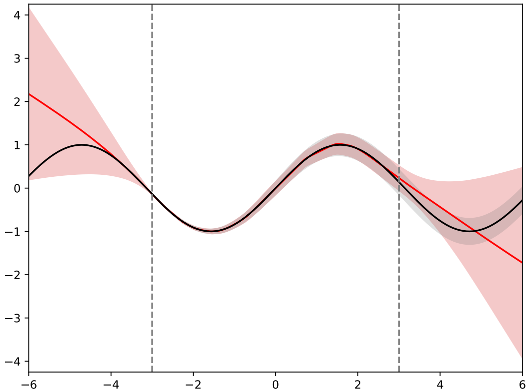


3. Illustrative example - Gaussian model, maximum-likelihood

- A corresponding Gaussian DNN model (2) trained via maximum-likelihood correctly accounts for aleatoric uncertainty, but generates overly confident predictions for inputs $|x^{\star}| > 3$ not seen during training:



- A Gaussian DNN model trained via approximate Bayesian inference (3), with $M = 1\,000$ samples obtained via HMC, is additionally able to predict more reasonable uncertainties in the region where no training data was available:



Outline

4. Ensembling as approximate Bayesian inference

Ensembling: create a parametric model $p(y|x,\theta)$ using a DNN f_θ, learn point estimates $\{\hat{\theta}^{(m)}\}_{m=1}^{M}$ by repeatedly minimizing $-\log p(Y|X,\theta)$ with *random initialization*, and average over the models to obtain the predictive distribution:

$$\hat{p}(y^\star|x^\star) \triangleq \frac{1}{M}\sum_{m=1}^{M} p(y^\star|x^\star,\hat{\theta}^{(m)}). \tag{4}$$

Ensembling: create a parametric model $p(y|x, \theta)$ using a DNN f_θ, learn point estimates $\{\hat{\theta}^{(m)}\}_{m=1}^{M}$ by repeatedly minimizing $-\log p(Y|X, \theta)$ with *random initialization*, and average over the models to obtain the predictive distribution:

$$\hat{p}(y^\star|x^\star) \triangleq \frac{1}{M}\sum_{m=1}^{M} p(y^\star|x^\star, \hat{\theta}^{(m)}). \tag{4}$$

Approximate Bayesian inference:

$$\hat{p}(y^\star|x^\star, \mathcal{D}) \triangleq \frac{1}{M}\sum_{i=1}^{M} p(y^\star|x^\star, \theta^{(i)}), \quad \theta^{(i)} \sim q(\theta) \approx p(\theta|\mathcal{D}). \tag{5}$$

4. Ensembling as approximate Bayesian inference

UPPSALA UNIVERSITET

Ensembling: create a parametric model $p(y|x,\theta)$ using a DNN f_θ, learn point estimates $\{\hat{\theta}^{(m)}\}_{m=1}^{M}$ by repeatedly minimizing $-\log p(Y|X,\theta)$ with *random initialization*, and average over the models to obtain the predictive distribution:

$$\hat{p}(y^\star|x^\star) \triangleq \frac{1}{M}\sum_{m=1}^{M} p(y^\star|x^\star,\hat{\theta}^{(m)}). \tag{4}$$

Approximate Bayesian inference:

$$\hat{p}(y^\star|x^\star,\mathcal{D}) \triangleq \frac{1}{M}\sum_{i=1}^{M} p(y^\star|x^\star,\theta^{(i)}), \quad \theta^{(i)} \sim q(\theta) \approx p(\theta|\mathcal{D}). \tag{5}$$

- Since $\{\hat{\theta}^{(m)}\}_{m=1}^{M}$ always can be seen as samples from *some* distribution $\hat{q}(\theta)$, we note that (4) and (5) are virtually identical.

4. Ensembling as approximate Bayesian inference

UPPSALA UNIVERSITET

- Ensembling can thus be viewed as approximate Bayesian inference.

- Ensembling can thus be viewed as approximate Bayesian inference. The level of approximation is determined by the ensemble size M and how well the implicit sampling distribution $\hat{q}(\theta)$ approximates the posterior $p(\theta|\mathcal{D})$.

4. Ensembling as approximate Bayesian inference

- Ensembling can thus be viewed as approximate Bayesian inference. The level of approximation is determined by the ensemble size M and how well the implicit sampling distribution $\hat{q}(\theta)$ approximates the posterior $p(\theta|\mathcal{D})$.

- Since $p(Y|X,\theta)$ is *highly* **multi-modal** for DNNs, so is $p(\theta|\mathcal{D}) \propto p(Y|X,\theta)p(\theta)$.

4. Ensembling as approximate Bayesian inference

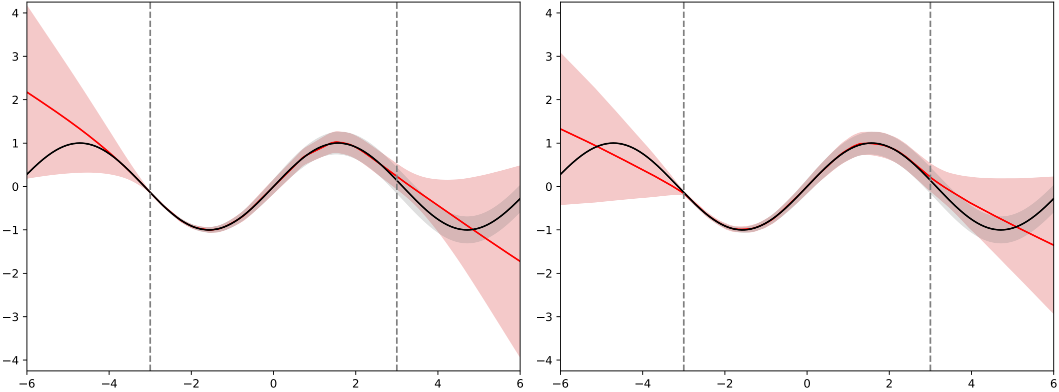
- Ensembling can thus be viewed as approximate Bayesian inference. The level of approximation is determined by the ensemble size M and how well the implicit sampling distribution $\hat{q}(\theta)$ approximates the posterior $p(\theta|\mathcal{D})$.

- Since $p(Y|X, \theta)$ is *highly* **multi-modal** for DNNs, so is $p(\theta|\mathcal{D}) \propto p(Y|X, \theta)p(\theta)$.
- Also, by minimizing $-\log p(Y|X, \theta)$ multiple times using SGD, starting from **randomly chosen** initial points, we are likely to find many different local optima.

- Ensembling can thus be viewed as approximate Bayesian inference. The level of approximation is determined by the ensemble size M and how well the implicit sampling distribution $\hat{q}(\theta)$ approximates the posterior $p(\theta|\mathcal{D})$.

- Since $p(Y|X,\theta)$ is *highly* **multi-modal** for DNNs, so is $p(\theta|\mathcal{D}) \propto p(Y|X,\theta)p(\theta)$.
- Also, by minimizing $-\log p(Y|X,\theta)$ multiple times using SGD, starting from **randomly chosen** initial points, we are likely to find many different local optima.
- Ensembling can thus generate a compact set of samples $\{\hat{\theta}^{(m)}\}_{m=1}^{M}$ that captures the important aspect of multi-modality in $p(\theta|\mathcal{D})$.

- On the 1D regression problem, we observe that ensembling provides reasonable approximations to HMC, even for relatively small values of M:



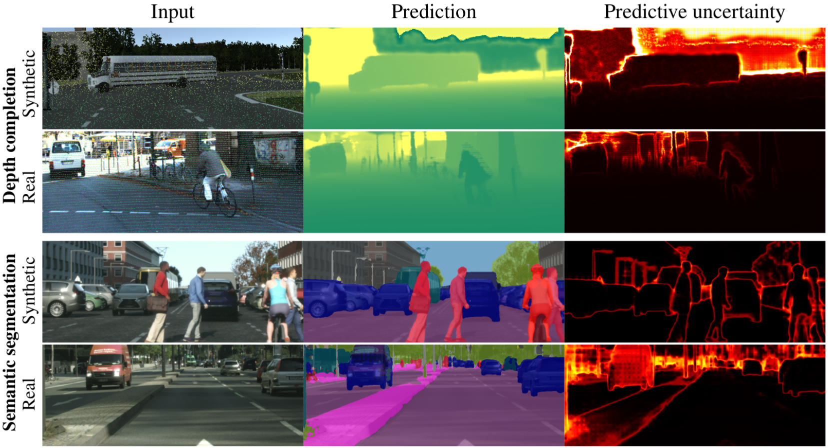
(a) HMC, $M = 1\,000$.

(b) Ensembling, $M = 16$.

Outline

UPPSALA UNIVERSITET

- Our extended abstract led to the paper **Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision**.
  - arXiv: https://arxiv.org/abs/1906.01620
  - Code: https://github.com/fregu856/evaluating_bdl



UPPSALA UNIVERSITET

- **Contributions:**
  - We propose an evaluation framework for predictive uncertainty estimation that is specifically designed to test the robustness required in real-world vision applications.
  - We perform an extensive comparison of **ensembling** and **MC-dropout** on the tasks of **depth completion** and **street-scene semantic segmentation**.

- **Contributions:**
  - We propose an evaluation framework for predictive uncertainty estimation that is specifically designed to test the robustness required in real-world vision applications.
  - We perform an extensive comparison of **ensembling** and **MC-dropout** on the tasks of **depth completion** and **street-scene semantic segmentation**.

MC-dropout: simple and scalable method for epistemic uncertainty estimation. Entails using *dropout* also at test time and averaging M stochastic forward passes on the same input. Can be interpreted as performing variational inference with a Bernoulli variational distribution.

- To simulate challenging conditions found *e.g.* in automotive applications, where robustness to **out-of-domain inputs** is required to ensure safety, we train models exclusively on **synthetic data** (Virtual KITTI[1], Synscapes[2]) and evaluate the predictive uncertainty on **real-world data** (KITTI[3], Cityscapes[4]).

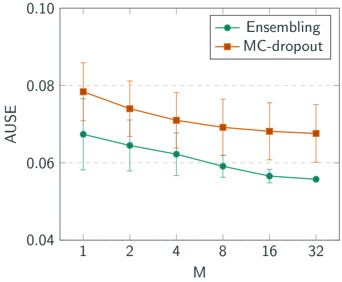
[1] https://europe.naverlabs.com/Research/Computer-Vision/Proxy-Virtual-Worlds/
[2] https://7dlabs.com/synscapes-overview
[3] http://www.cvlibs.net/datasets/kitti/
[4] https://www.cityscapes-dataset.com/

- To simulate challenging conditions found *e.g.* in automotive applications, where robustness to **out-of-domain inputs** is required to ensure safety, we train models exclusively on **synthetic data** (Virtual KITTI[1], Synscapes[2]) and evaluate the predictive uncertainty on **real-world data** (KITTI[3], Cityscapes[4]).

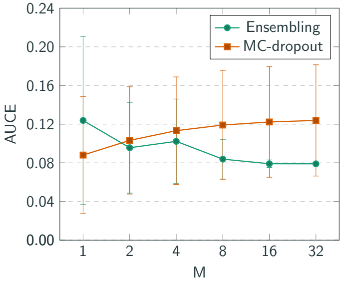
- We evaluate the methods in terms of the *relative* **AUSE** metric (how well the ordering of predictions in terms of estimated uncertainty matches the "oracle" ordering in terms of true prediction error) and the *absolute* measure of **calibration**.

[1] https://europe.naverlabs.com/Research/Computer-Vision/Proxy-Virtual-Worlds/
[2] https://7dlabs.com/synscapes-overview
[3] http://www.cvlibs.net/datasets/kitti/
[4] https://www.cityscapes-dataset.com/

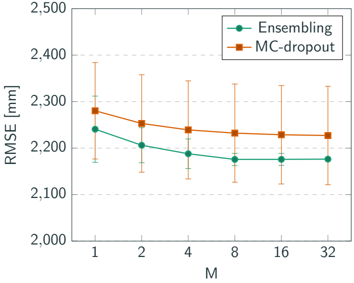
Depth completion:



(a) AUSE, lower is better.

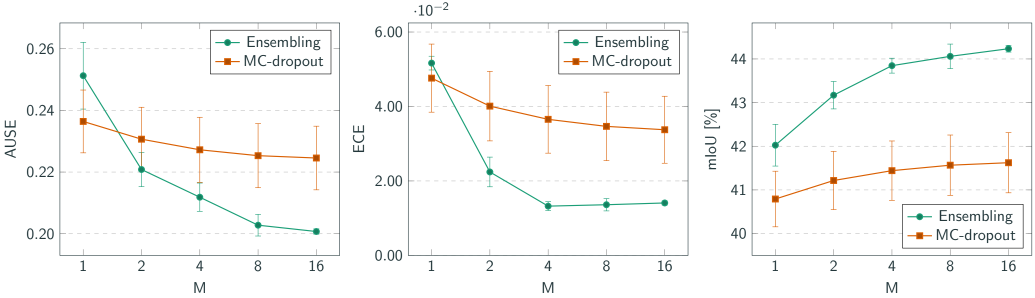


(b) Calibration (AUCE), lower is better.



(c) RMSE, lower is better.

Street-scene semantic segmentation:



(a) AUSE, lower is better. **(b)** Calibration (ECE), lower is better. **(c)** mIoU, higher is better.

UPPSALA UNIVERSITET

Video: https://youtu.be/CabPVqtzsOI.

Outline

6. Conclusion

- We noted that ensembling naturally can be viewed as an approximate Bayesian inference method, and provided some intuition for why it should be a reasonable approximation specifically for DNNs.

6. Conclusion

- We noted that ensembling naturally can be viewed as an approximate Bayesian inference method, and provided some intuition for why it should be a reasonable approximation specifically for DNNs.

- We proposed an evaluation framework for predictive uncertainty estimation that is specifically designed to test the robustness required in real-world computer vision applications.

6. Conclusion

UPPSALA UNIVERSITET

- We noted that ensembling naturally can be viewed as an approximate Bayesian inference method, and provided some intuition for why it should be a reasonable approximation specifically for DNNs.
- We proposed an evaluation framework for predictive uncertainty estimation that is specifically designed to test the robustness required in real-world computer vision applications.
- We performed an extensive comparison of ensembling and MC-dropout on the tasks of depth completion and street-scene semantic segmentation, the results of which suggest that **ensembling** consistently provides more reliable and useful predictive uncertainty estimates.

Contact

Fredrik K. Gustafsson, Uppsala University
fredrik.gustafsson@it.uu.se
www.fregu856.com