UPPSALA UNIVERSITET

Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision

Fredrik K. Gustafsson

Uppsala University

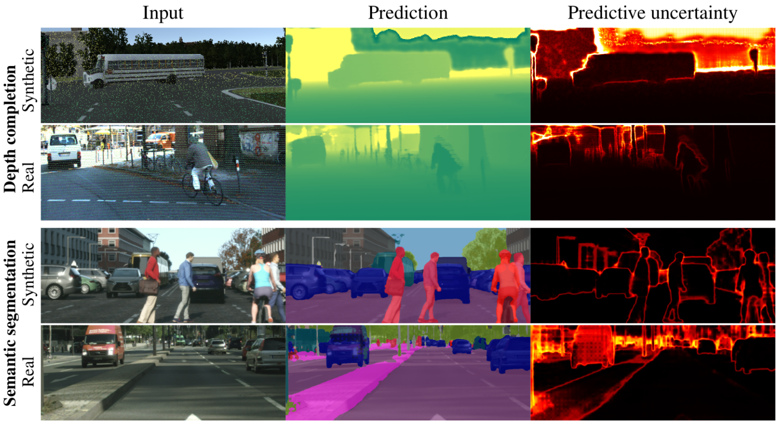
Zenuity

Göteborg, June 18, 2019

Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision

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

- arXiv: https://arxiv.org/abs/1906.01620
- Code: https://github.com/fregu856/evaluating_bdl
- These slides: http://www.fregu856.com



- 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*.

Outline

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

UPPSALA UNIVERSITET

> *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.

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 $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

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 $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

In classification problems, the target space $\mathcal{Y}$ consists of a finite set of C classes. In regression, $\mathcal{Y}$ is continuous, *e.g.* $\mathcal{Y} = \mathbb{R}^K$.

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 $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$, $(x_i, y_i) \sim p(x, y)$.

In classification problems, the target space $\mathcal{Y}$ consists of a finite set of C classes. In regression, $\mathcal{Y}$ is continuous, *e.g.* $\mathcal{Y} = \mathbb{R}^K$.

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 $f_\theta(x) \in \mathcal{U}$.

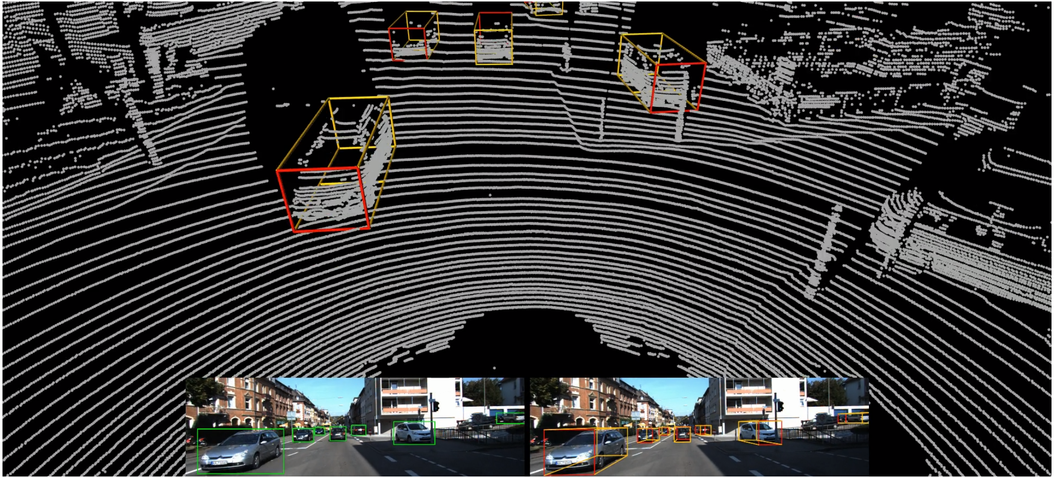
- Given an input x, the “correct” value of y is not always obvious *(what is the correct classification target for an image containing both a cat and a dog?)*.

1. Introduction - Aleatoric uncertainty

- Given an input x, the "correct" value of y is not always obvious *(what is the correct classification target for an image containing both a cat and a dog?)*.
- **Aleatoric** uncertainty captures this inherent and irreducible data uncertainty.

- Given an input x, the "correct" value of y is not always obvious *(what is the correct classification target for an image containing both a cat and a dog?)*.
- **Aleatoric** uncertainty captures this inherent and irreducible data uncertainty.
- *Input-dependent* aleatoric uncertainty is present whenever we expect the 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.

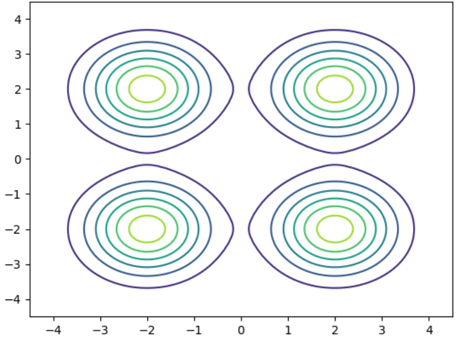
- **Epistemic** uncertainty accounts for uncertainty in the DNN model parameters.

1. Introduction - Epistemic uncertainty

UPPSALA UNIVERSITET

- **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.

- **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

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 $f_\theta(x) \in \mathcal{U}$.

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 $f_\theta(x) \in \mathcal{U}$.

- To estimate input-dependent **aleatoric** uncertainty, it is enough to:
    - Let 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 $f_\theta(x) \in \mathcal{U}$.

- To estimate input-dependent **aleatoric** uncertainty, it is enough to:
  - Let a DNN f_θ output the parameters of some probability distribution, creating a parametric model $p(y|x, \theta)$ of the conditional distribution.
  - Find the maximum-likelihood estimate of the model parameters, $\hat{\theta}_{\text{MLE}}$, by minimizing the negative log-likelihood $-\log p(Y|X, \theta) = -\sum_{i=1}^N \log p(y_i|x_i, \theta)$.

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 $f_\theta(x) \in \mathcal{U}$.

- To estimate input-dependent **aleatoric** uncertainty, it is enough to:
  - Let a DNN f_θ output the parameters of some probability distribution, creating a parametric model $p(y|x, \theta)$ of the conditional distribution.
  - Find the maximum-likelihood estimate of the model parameters, $\hat{\theta}_{\mathrm{MLE}}$, by minimizing the negative log-likelihood $-\log p(Y|X, \theta) = -\sum_{i=1}^{N} \log p(y_i|x_i, \theta)$.
- Given $x^\star$ at test time, the DNN then predicts the *distribution* $p(y^\star|x^\star, \hat{\theta}_{\mathrm{MLE}})$ over $y^\star$, capturing aleatoric uncertainty.

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

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

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.

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):

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

UPPSALA UNIVERSITET

- 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:

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

UPPSALA UNIVERSITET

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

- **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.

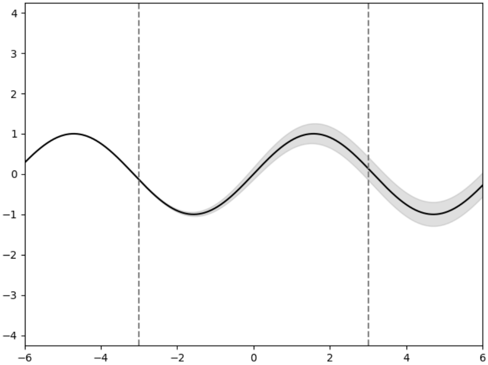
- In practice, an approximate posterior $q(\theta) \approx p(\theta|\mathcal{D})$ has to be used, resulting in:

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

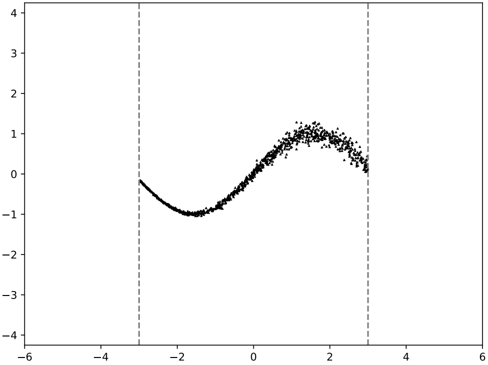
Outline

We consider the following 1D regression problem:

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



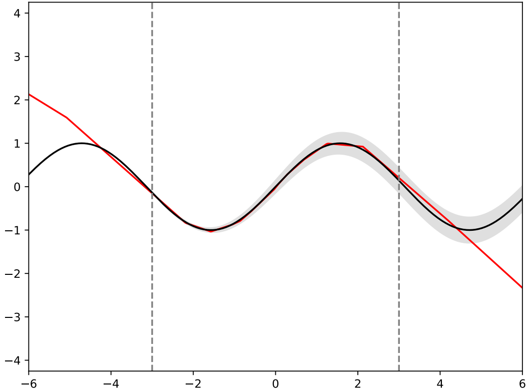
(a) True data generator.



(b) Training dataset.

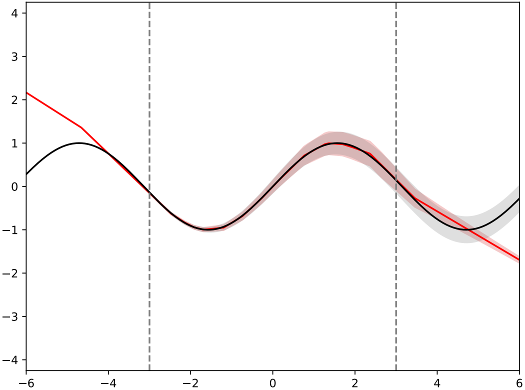
UPPSALA UNIVERSITET

- Following the conventional regression approach, a small 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:

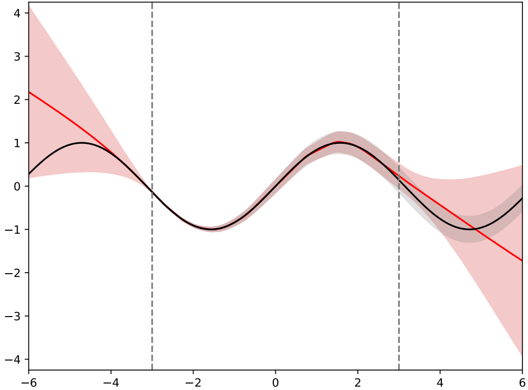


3. Illustrative example - Gaussian model, maximum-likelihood

- 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 [11], is additionally able to predict more reasonable uncertainties in the region where no training data was available:



Outline

Ensembling: create a parametric model $p(y|x,\theta)$ using a DNN f_θ, learn M 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 M 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:

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

Approximate Bayesian inference:

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

UPPSALA UNIVERSITET

Ensembling: create a parametric model $p(y|x,\theta)$ using a DNN f_θ, learn M 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

- Ensembling can thus naturally be viewed as approximate Bayesian inference.

- Ensembling can thus naturally 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 naturally 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 $p(\theta|\mathcal{D}) \propto p(Y|X,\theta)p(\theta)$.

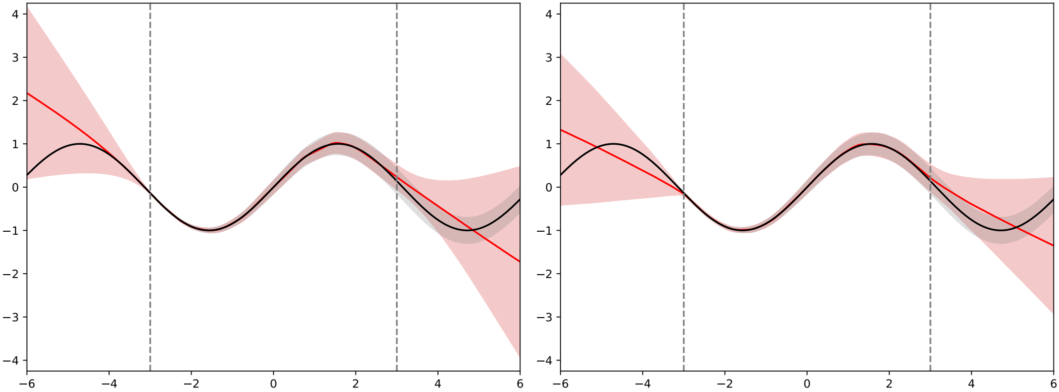
- Ensembling can thus naturally be viewed as approximate Bayesian inference. The level of approximation is determined by the ensemble size M and how well the implicit sampling distribution $\hat{q}(\theta)$ approximates the posterior $p(\theta|\mathcal{D})$.

- 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 naturally 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 [11], even for relatively small values of M:



(a) HMC [11], $M = 1\,000$.

(b) Ensembling, $M = 16$.

Outline

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 [5, 8].

- Our evaluation is motivated by real-world conditions found *e.g.* in automotive applications, where robustness to varying environments and weather conditions is required to ensure safety.

5. Experiments - Evaluation

- Our evaluation is motivated by real-world conditions found *e.g.* in automotive applications, where robustness to varying environments and weather conditions is required to ensure safety.

- Since images captured in these different circumstances could all represent distinctly different regions of the vast input image space, it is infeasible to ensure that all encountered inputs will be well-represented by the training data. Thus, we argue that robustness to **out-of-domain inputs** is crucial.

- Our evaluation is motivated by real-world conditions found *e.g.* in automotive applications, where robustness to varying environments and weather conditions is required to ensure safety.

- Since images captured in these different circumstances could all represent distinctly different regions of the vast input image space, it is infeasible to ensure that all encountered inputs will be well-represented by the training data. Thus, we argue that robustness to **out-of-domain inputs** is crucial.

- To simulate these challenging conditions and test the required robustness, we train models exclusively on **synthetic data** and evaluate the predictive uncertainty on **real-world data**.

- To provide a deeper and more fair analysis, we study all metrics as functions of the number of samples M, not just for a fixed value.

- To provide a deeper and more fair analysis, we study all metrics as functions of the number of samples M, not just for a fixed value.
- To improve rigour of our evaluation, we repeat each experiment multiple times and report results together with the observed variation.

5.1. Illustrative toy problems

- First, we conduct experiments on **illustrative toy problems** to gain insights into how ensembling and MC-dropout fare against other approximate Bayesian inference methods.

5.1. Illustrative toy problems

- First, we conduct experiments on **illustrative toy problems** to gain insights into how ensembling and MC-dropout fare against other approximate Bayesian inference methods.
- We compare with SGLD [13] and SGHMC [2], which are both Stochastic Gradient MCMC (SG-MCMC) [10] methods.

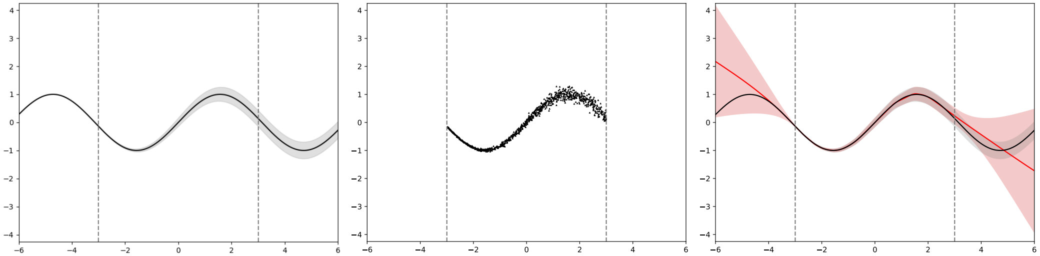
- First, we conduct experiments on **illustrative toy problems** to gain insights into how ensembling and MC-dropout fare against other approximate Bayesian inference methods.
- We compare with SGLD [13] and SGHMC [2], which are both Stochastic Gradient MCMC (SG-MCMC) [10] methods.
- We evaluate the methods by quantitatively measuring how well the obtained predictive distributions approximate that of HMC [11] with $M = 1\,000$ samples and prior $p(\theta) = \mathcal{N}(0, I_P)$.

5.1. Illustrative toy problems

- First, we conduct experiments on **illustrative toy problems** to gain insights into how ensembling and MC-dropout fare against other approximate Bayesian inference methods.
- We compare with SGLD [13] and SGHMC [2], which are both Stochastic Gradient MCMC (SG-MCMC) [10] methods.
- We evaluate the methods by quantitatively measuring how well the obtained predictive distributions approximate that of HMC [11] with $M = 1\,000$ samples and prior $p(\theta) = \mathcal{N}(0, I_P)$.
- We thus take as our metric the **KL divergence** $D_{\mathrm{KL}}(p \parallel p_{\mathrm{HMC}})$ with respect to this target predictive distribution p_{HMC}.

- First, we conduct experiments on **illustrative toy problems** to gain insights into how ensembling and MC-dropout fare against other approximate Bayesian inference methods.
- We compare with SGLD [13] and SGHMC [2], which are both Stochastic Gradient MCMC (SG-MCMC) [10] methods.
- We evaluate the methods by quantitatively measuring how well the obtained predictive distributions approximate that of HMC [11] with $M = 1\,000$ samples and prior $p(\theta) = \mathcal{N}(0, I_P)$.
- We thus take as our metric the **KL divergence** $D_{\mathrm{KL}}(p \parallel p_{\mathrm{HMC}})$ with respect to this target predictive distribution p_{HMC}.
- Note that HMC is considered a "gold standard" method for approximate Bayesian inference, but does not scale to large DNNs or large-scale datasets.

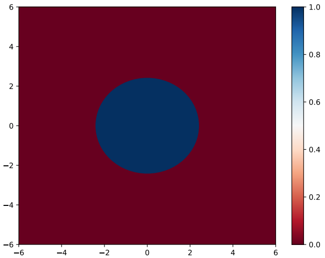
- For **regression**, we study the previously defined 1D problem:



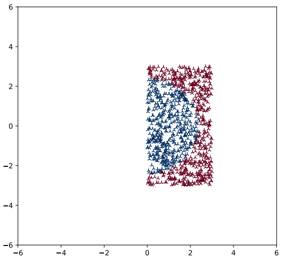
(a) True data generator. **(b)** Training dataset. **(c)** HMC [11] "ground truth".

UPPSALA UNIVERSITET

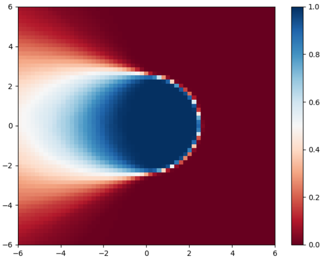
- For **classification**, we study the following binary classification problem:



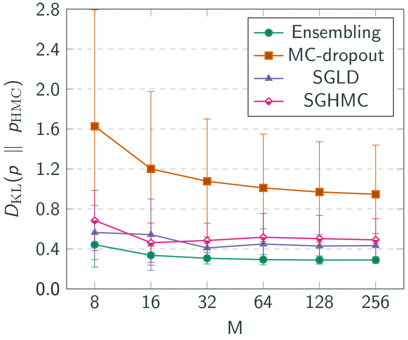
(a) True data generator.



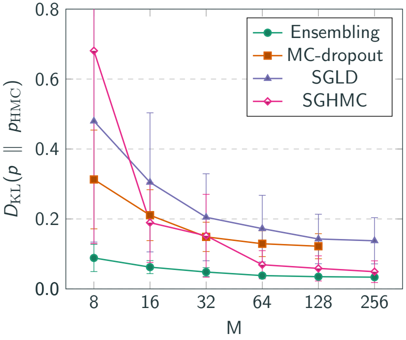
(b) Training dataset.



(c) HMC [11] "ground truth".



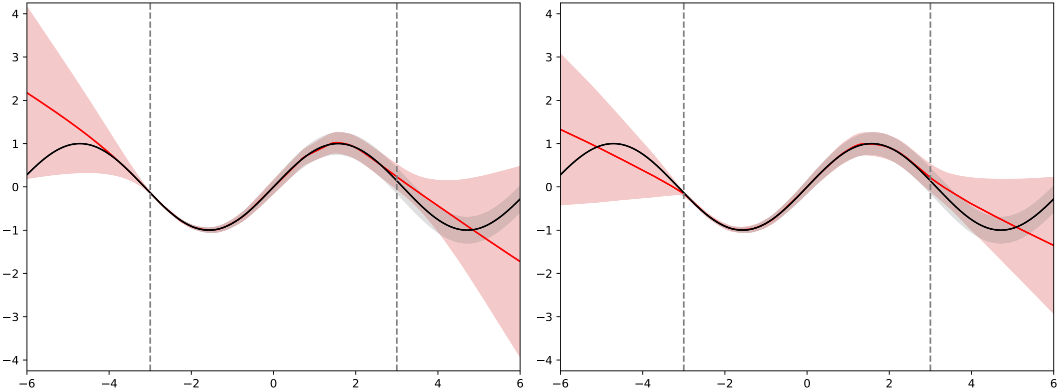
(a) Regression.



(b) Classification.

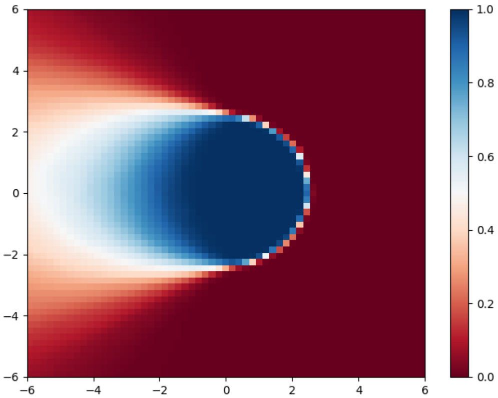
- We observe that **ensembling** consistently outperforms the compared methods, and MC-dropout in particular.

UPPSALA UNIVERSITET

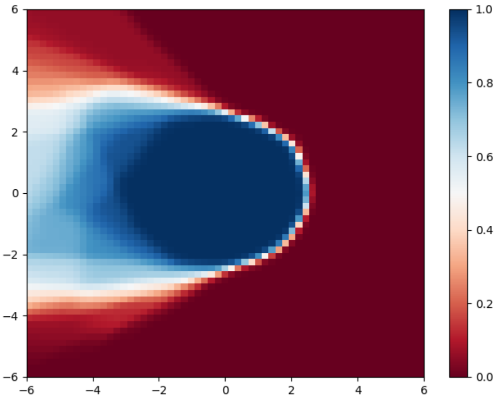


(a) HMC [11], $M = 1\,000$.   **(b)** Ensembling, $M = 16$.

- We observe that **ensembling** provides reasonable approximations to HMC [11], even for relatively small values of M.



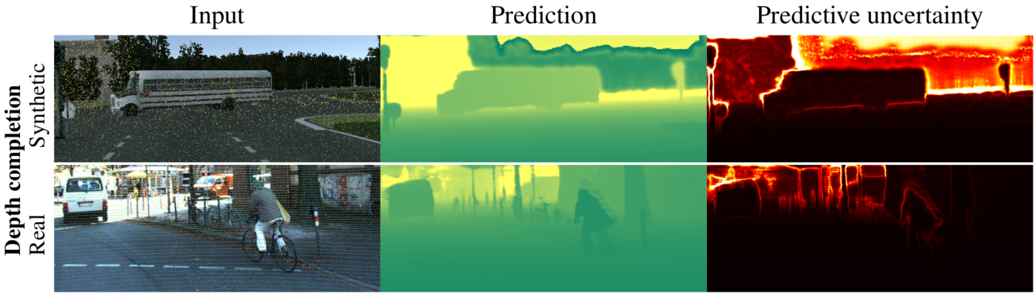
(a) HMC [11], $M = 1\,000$.

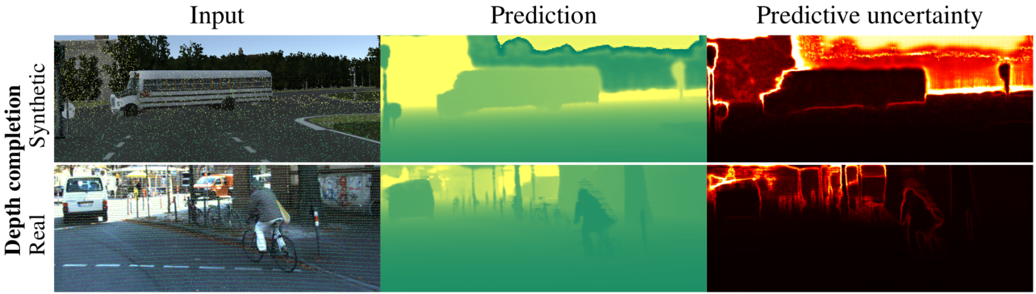


(b) Ensembling, $M = 16$.

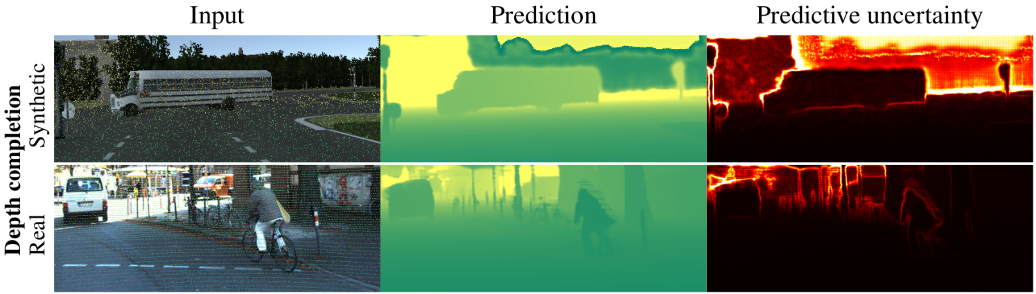
- We observe that **ensembling** provides reasonable approximations to HMC [11], even for relatively small values of M.

5.2. Depth completion

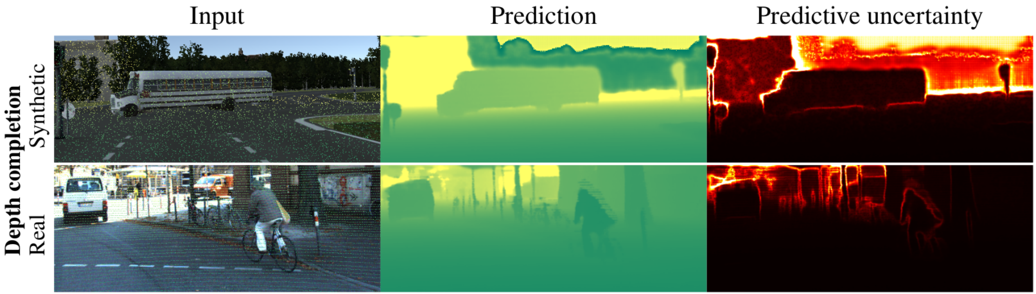




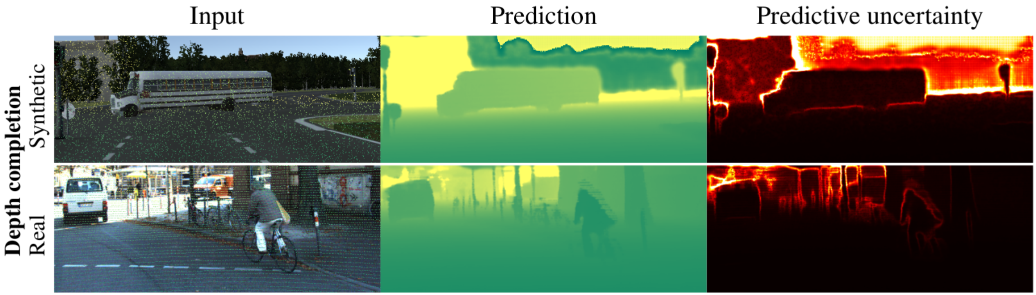
- In **depth completion**, we are given an image $x_{\text{img}} \in \mathbb{R}^{h \times w \times 3}$ and an associated *sparse* depth map $x_{\text{sparse}} \in \mathbb{R}^{h \times w}$. Only non-zero pixels of x_{sparse} correspond to LiDAR depth measurements, projected onto the image plane.



- In **depth completion**, we are given an image $x_{\text{img}} \in \mathbb{R}^{h \times w \times 3}$ and an associated *sparse* depth map $x_{\text{sparse}} \in \mathbb{R}^{h \times w}$. Only non-zero pixels of x_{sparse} correspond to LiDAR depth measurements, projected onto the image plane.
- The goal is to predict a dense depth map $y \in \mathbb{R}^{h \times w}$ of the scene.



- We utilize the **KITTI depth completion** [6, 12] and **Virtual KITTI** [4] datasets.



- We utilize the **KITTI depth completion** [6, 12] and **Virtual KITTI** [4] datasets.
- We train on Virtual KITTI (18 930 examples) and evaluate on KITTI depth completion (1 000 validation examples).

5.2. Depth completion - Model

- We use the DNN model presented by Ma *et al.* [9].

UPPSALA UNIVERSITET

- We use the DNN model presented by Ma *et al.* [9].
- The inputs x_{img}, x_{sparse} are separately processed by initial convolutional layers, concatenated and fed to an encoder-decoder architecture based on ResNet34.

- We use the DNN model presented by Ma *et al.* [9].
- The inputs x_{img}, x_{sparse} are separately processed by initial convolutional layers, concatenated and fed to an encoder-decoder architecture based on ResNet34.
- We employ the **Gaussian model** (2) by duplicating the final convolutional layer, outputting $\mu \in \mathbb{R}^{h \times w}$ and $\log \sigma^2 \in \mathbb{R}^{h \times w}$ instead of the plain depth $\hat{y} \in \mathbb{R}^{h \times w}$.

5.2. Depth completion - Evaluation metrics

- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.

5.2. Depth completion - Evaluation metrics

- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **AUSE**: *Area Under the Sparsification Error curve*, measures how well the ordering of predictions induced by the estimated predictive uncertainty (sorted from least to most uncertain) matches the "oracle" ordering in terms of true prediction error.

5.2. Depth completion - Evaluation metrics

- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **AUSE**: *Area Under the Sparsification Error curve*, measures how well the ordering of predictions induced by the estimated predictive uncertainty (sorted from least to most uncertain) matches the "oracle" ordering in terms of true prediction error. Note that a perfect AUSE score can be achieved even if the true predictive uncertainty is consistently underestimated.

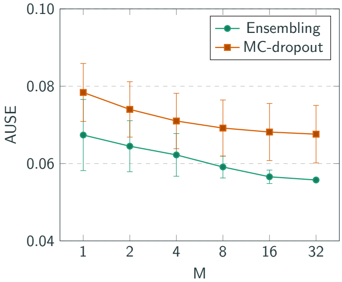
- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **AUSE**: *Area Under the Sparsification Error curve*, measures how well the ordering of predictions induced by the estimated predictive uncertainty (sorted from least to most uncertain) matches the "oracle" ordering in terms of true prediction error. Note that a perfect AUSE score can be achieved even if the true predictive uncertainty is consistently underestimated.
- **Calibration**: our model outputs a Gaussian for each pixel, and we can thus construct prediction intervals of confidence level $p \in]0, 1[$ using the corresponding quantiles.

- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **AUSE**: *Area Under the Sparsification Error curve*, measures how well the ordering of predictions induced by the estimated predictive uncertainty (sorted from least to most uncertain) matches the "oracle" ordering in terms of true prediction error. Note that a perfect AUSE score can be achieved even if the true predictive uncertainty is consistently underestimated.
- **Calibration**: our model outputs a Gaussian for each pixel, and we can thus construct prediction intervals of confidence level $p \in]0,1[$ using the corresponding quantiles. The proportion of pixels for which the prediction interval covers the target is expected to equal $p \in]0,1[$ for a perfectly calibrated model.

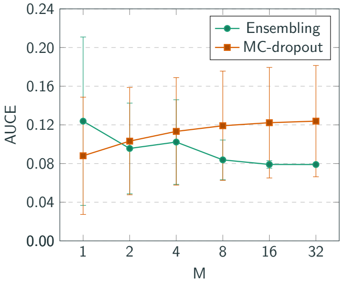
5.2. Depth completion - Evaluation metrics

- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **AUSE**: *Area Under the Sparsification Error curve*, measures how well the ordering of predictions induced by the estimated predictive uncertainty (sorted from least to most uncertain) matches the "oracle" ordering in terms of true prediction error. Note that a perfect AUSE score can be achieved even if the true predictive uncertainty is consistently underestimated.
- **Calibration**: our model outputs a Gaussian for each pixel, and we can thus construct prediction intervals of confidence level $p \in]0, 1[$ using the corresponding quantiles. The proportion of pixels for which the prediction interval covers the target is expected to equal $p \in]0, 1[$ for a perfectly calibrated model.
- We also evaluate in terms of the standard RMSE metric.

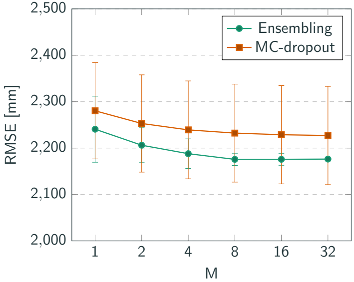
5.2. Depth completion - Results



(a) AUSE, lower is better.

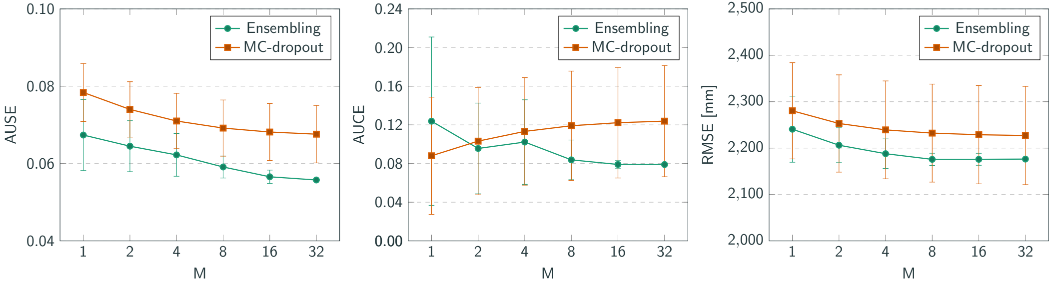


(b) Calibration (AUCE), lower is better.



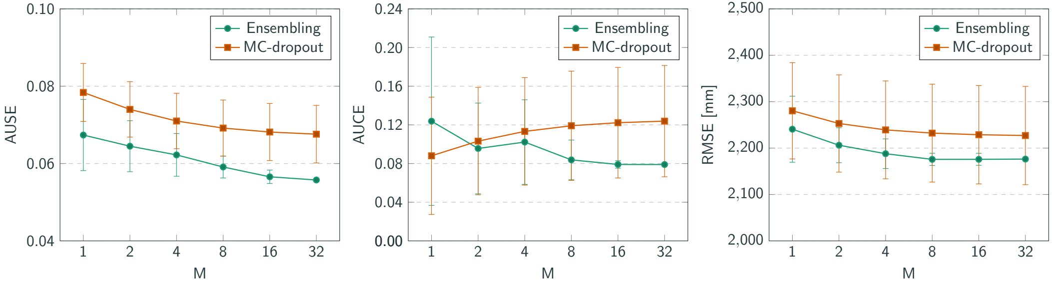
(c) RMSE, lower is better.

UPPSALA UNIVERSITET



(a) AUSE, lower is better. **(b)** Calibration (AUCE), lower is better. **(c)** RMSE, lower is better.

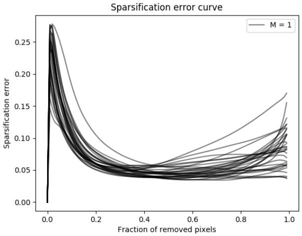
- We observe in **(a)** that ensembling consistently outperforms MC-dropout in terms of AUSE. However, the curves decrease as a function of M in a similar manner.



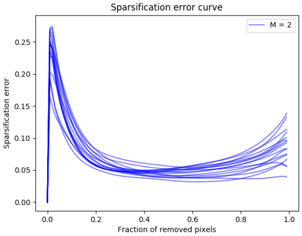
(a) AUSE, lower is better. **(b)** Calibration (AUCE), lower is better. **(c)** RMSE, lower is better.

- We observe in **(a)** that ensembling consistently outperforms MC-dropout in terms of AUSE. However, the curves decrease as a function of M in a similar manner.
- A ranking can be more readily conducted based on **(b)**, where we observe a clearly improving trend for **ensembling**, whereas MC-dropout gets progressively worse.

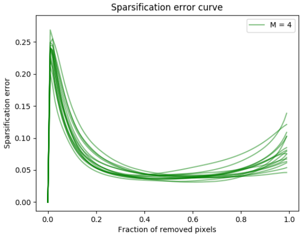
5.2. Depth completion - Results, sparsification, ensembling



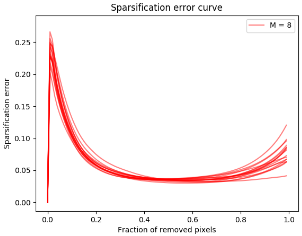
(a) $M = 1$.



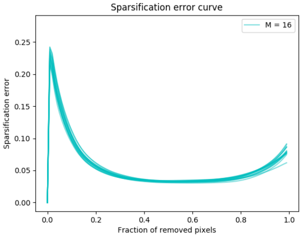
(b) $M = 2$.



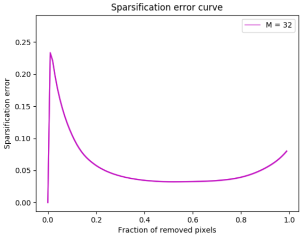
(c) $M = 4$.



(d) $M = 8$.



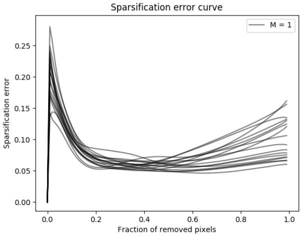
(e) $M = 16$.



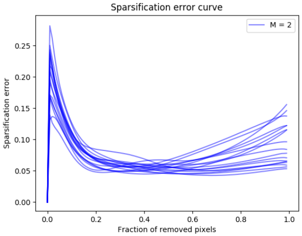
(f) $M = 32$.

5.2. Depth completion - Results, sparsification, MC-dropout

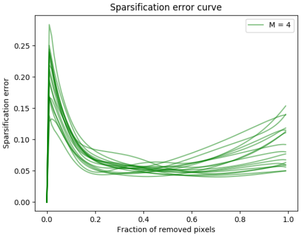
UPPSALA UNIVERSITET



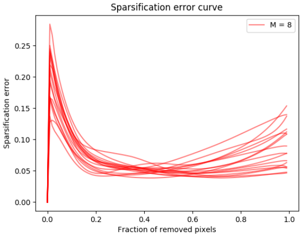
(a) $M = 1$.



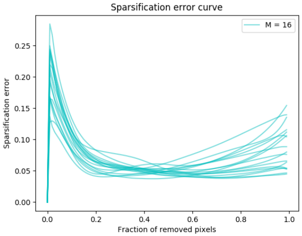
(b) $M = 2$.



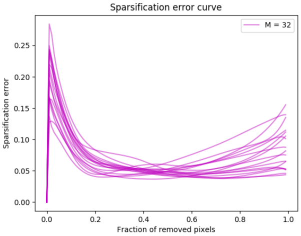
(c) $M = 4$.



(d) $M = 8$.

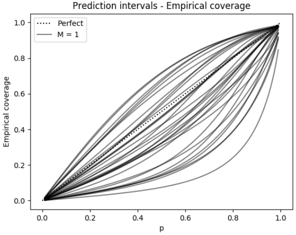


(e) $M = 16$.

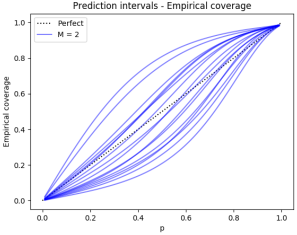


(f) $M = 32$.

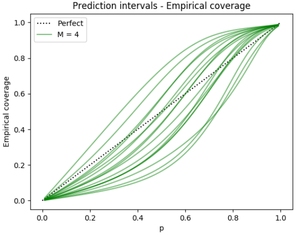
5.2. Depth completion - Results, calibration, ensembling



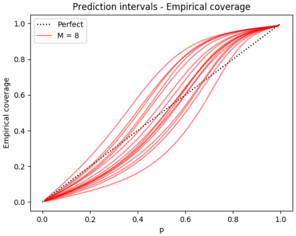
(a) $M = 1$.



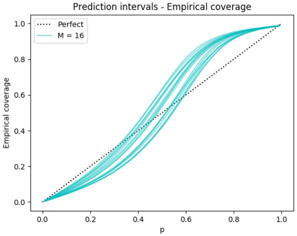
(b) $M = 2$.



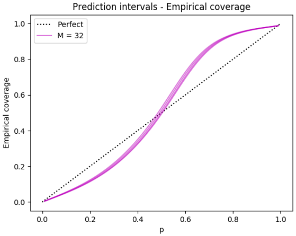
(c) $M = 4$.



(d) $M = 8$.

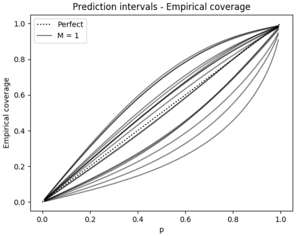


(e) $M = 16$.

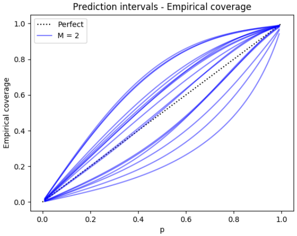


(f) $M = 32$.

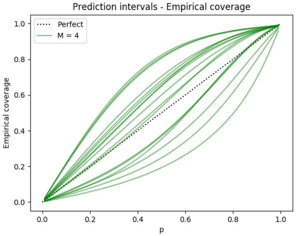
5.2. Depth completion - Results, calibration, MC-dropout



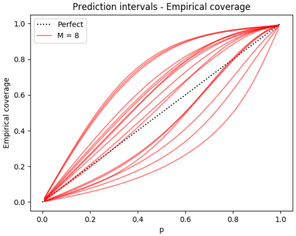
(a) $M = 1$.



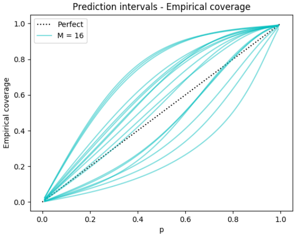
(b) $M = 2$.



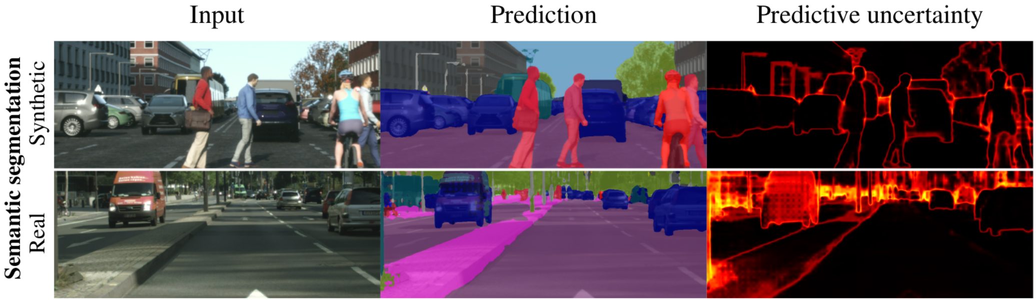
(c) $M = 4$.



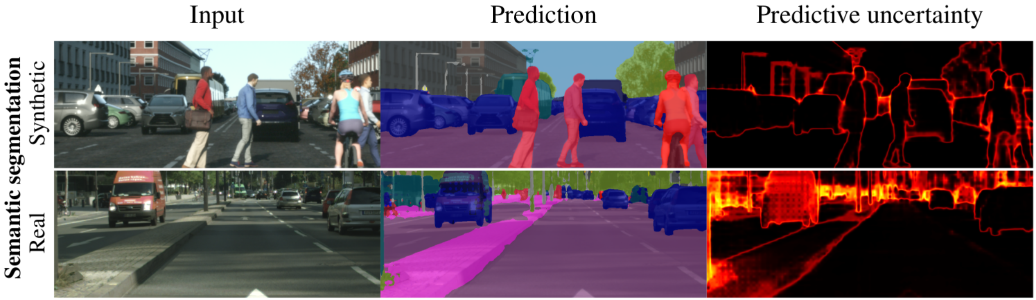
(d) $M = 8$.



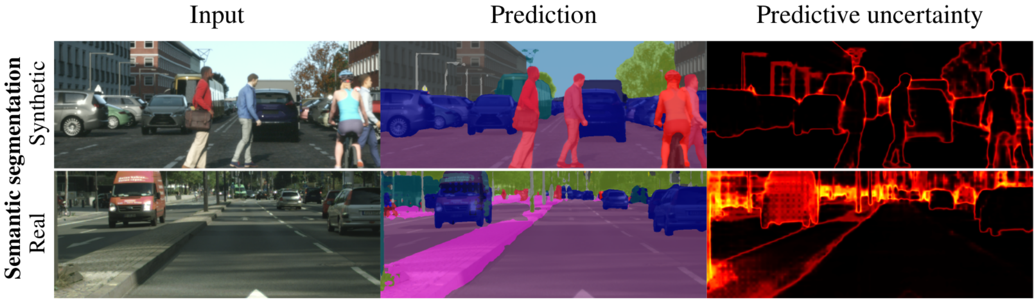
(e) $M = 16$.


Input
Prediction
Predictive uncertainty
Semantic segmentation
Synthetic
Real

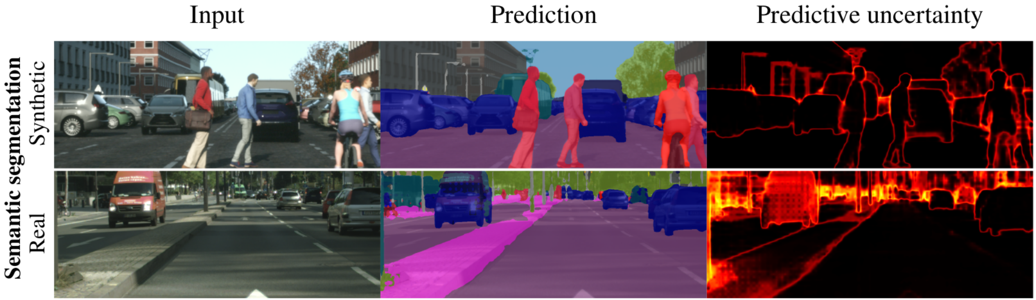
5.3. Street-scene semantic segmentation



- In **street-scene semantic segmentation**, we are given an image $x \in \mathbb{R}^{h \times w \times 3}$.
- The goal is to predict y of size $h \times w$, in which each pixel is assigned to one of C different class labels (road, sidewalk, car, etc.).



- We utilize the **Cityscapes** [3] and **Synscapes** [14] datasets.



- We utilize the **Cityscapes** [3] and **Synscapes** [14] datasets.
- We train on Synscapes (2 975 examples) and evaluate on Cityscapes (500 validation examples).

- We use the DeepLabv3 DNN model presented by Chen *et al.* [1].

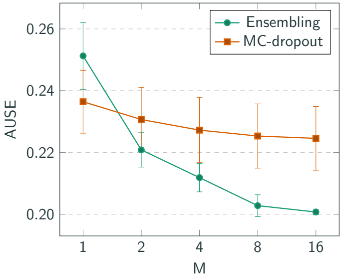
- We use the DeepLabv3 DNN model presented by Chen *et al.* [1].
- The input image x is processed by a ResNet101 and then fed to an ASPP module, outputting logits at 1/8 of the original resolution. These are then upsampled to image resolution using bilinear interpolation.

- We use the DeepLabv3 DNN model presented by Chen *et al.* [1].
- The input image x is processed by a ResNet101 and then fed to an ASPP module, outputting logits at 1/8 of the original resolution. These are then upsampled to image resolution using bilinear interpolation.
- The conventional **Categorical model** (1) is thus used for each pixel.

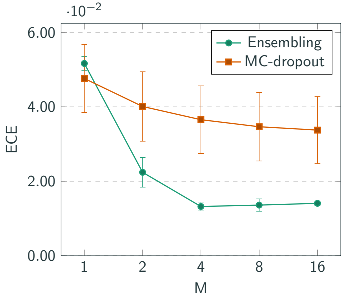
- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.

- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **Calibration**: all predictions are partitioned into L bins based on the maximum assigned confidence. For each bin, the difference between the average predicted confidence and the actual accuracy is then computed, and ECE *(Expected Calibration Error)* is obtained as the weighted average of these differences.

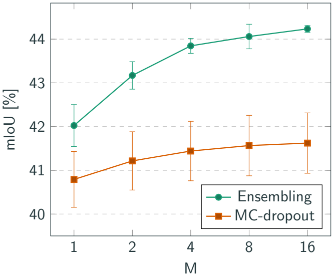
- We evaluate the methods in terms of quality of the estimated predictive uncertainty, as measured by the *relative* **AUSE** metric [7] and the *absolute* measure of uncertainty **calibration**.
- **Calibration**: all predictions are partitioned into L bins based on the maximum assigned confidence. For each bin, the difference between the average predicted confidence and the actual accuracy is then computed, and ECE *(Expected Calibration Error)* is obtained as the weighted average of these differences.
- We also evaluate in terms of the standard mIoU metric.



(a) AUSE, lower is better.

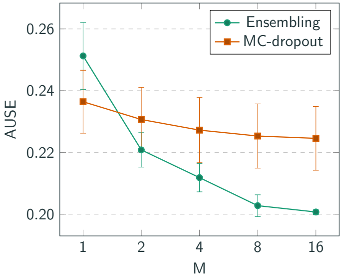


(b) Calibration (ECE), lower is better.

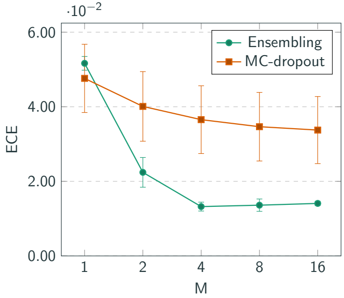


(c) mIoU, higher is better.

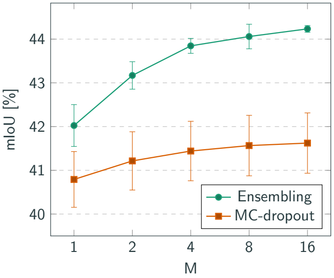
UPPSALA UNIVERSITET



(a) AUSE, lower is better.

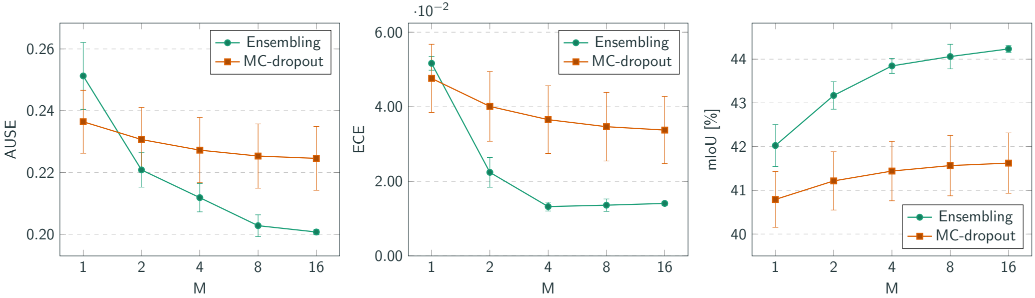


(b) Calibration (ECE), lower is better.



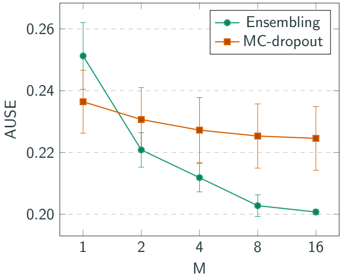
(c) mIoU, higher is better.

- Note that $M = 1$ corresponds to the baseline of only estimating aleatoric uncertainty.

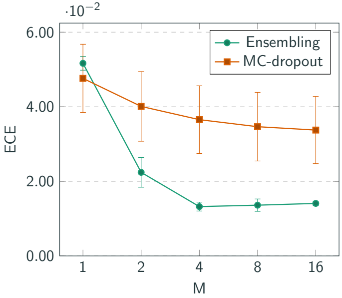


(a) AUSE, lower is better. **(b)** Calibration (ECE), lower is better. **(c)** mIoU, higher is better.

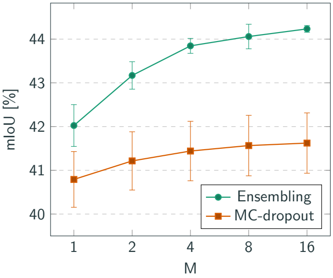
- Note that $M = 1$ corresponds to the baseline of only estimating aleatoric uncertainty. The metrics clearly improve as functions of M for both methods, demonstrating the importance of epistemic uncertainty estimation.



(a) AUSE, lower is better.

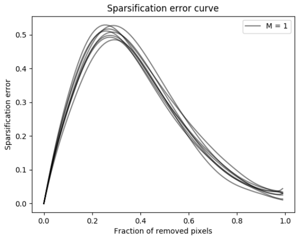


(b) Calibration (ECE), lower is better.

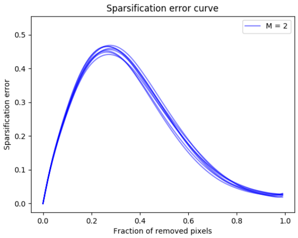


(c) mIoU, higher is better.

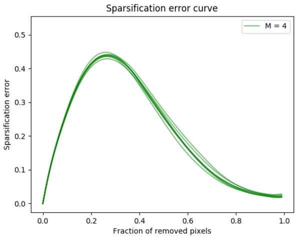
- Note that $M = 1$ corresponds to the baseline of only estimating aleatoric uncertainty. The metrics clearly improve as functions of M for both methods, demonstrating the importance of epistemic uncertainty estimation.
- We observe that the rate of improvement is generally greater for **ensembling**.



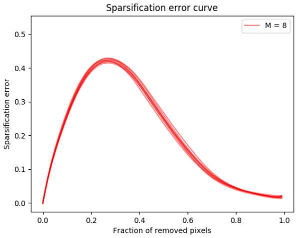
(a) $M = 1$.



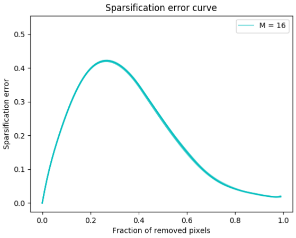
(b) $M = 2$.



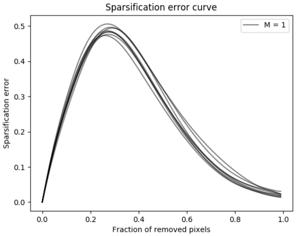
(c) $M = 4$.



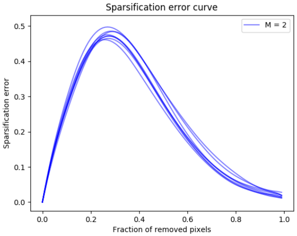
(d) $M = 8$.



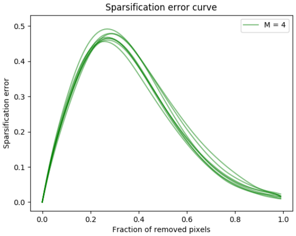
(e) $M = 16$.



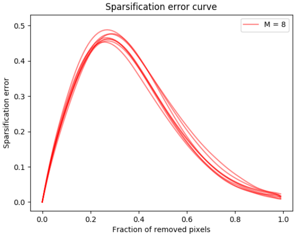
(a) $M = 1$.



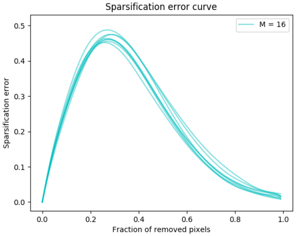
(b) $M = 2$.



(c) $M = 4$.

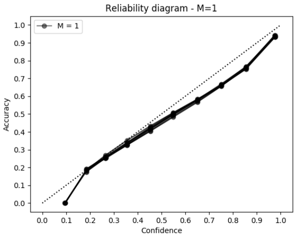


(d) $M = 8$.

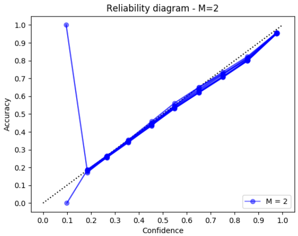


(e) $M = 16$.

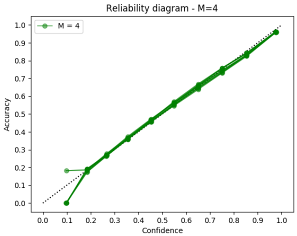
5.3. Semantic segmentation - Results, calibration, ensembling



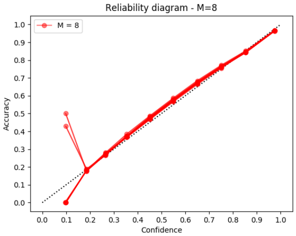
(a) $M = 1$.



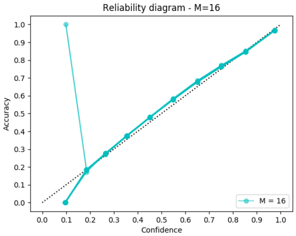
(b) $M = 2$.



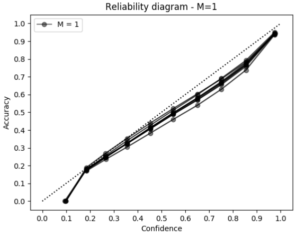
(c) $M = 4$.



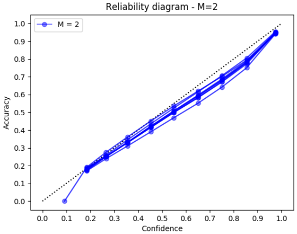
(d) $M = 8$.



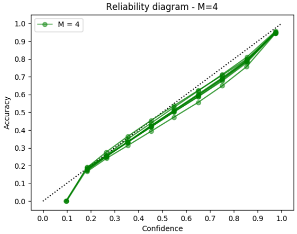
(e) $M = 16$.



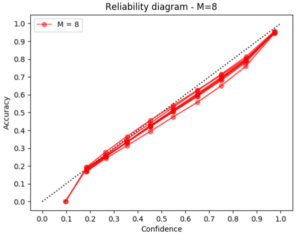
(a) $M = 1$.



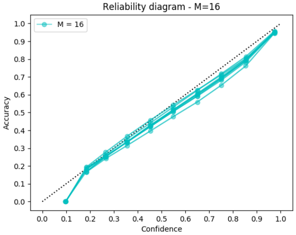
(b) $M = 2$.



(c) $M = 4$.



(d) $M = 8$.



(e) $M = 16$.

UPPSALA UNIVERSITET

- **Video:** https://youtu.be/CabPVqtzsOI.
- All shown results are for **ensembling** with $M = 8$.

5.4. Qualitative results - Depth completion

- **Video:** https://youtu.be/CabPVqtzsOI.
- **Depth completion**: 8:22 - 14:18.
  - Trained on Virtual KITTI, evaluated on Virtual KITTI (synthetic to synthetic): 8:22.
  - Trained on Virtual KITTI, evaluated on KITTI (synthetic to real): 9:26.
- The input image, input sparse depth map, ground truth depth map, prediction, predictive uncertainty, aleatoric uncertainty and epistemic uncertainty are visualized.
- Black: minimum uncertainty, white: maximum uncertainty.

5.4. Qualitative results - Street-scene semantic segmentation

- **Video:** `https://youtu.be/CabPVqtzsOI`.
- **Street-scene semantic segmentation**: 0:00 - 8:22.
  - Trained on Cityscapes, evaluated on Cityscapes (real to real): 0:00.
  - Trained on Synscapes, evaluated on Cityscapes (synthetic to real): 2:30.
  - Trained on Synscapes, evaluated on Synscapes (synthetic to synthetic): 5:00.
  - Trained on Cityscapes, evaluated on Synscapes (real to synthetic): 6:41
- On Cityscapes, the input image, prediction and predictive entropy are visualized.
- On Synscapes, the input image, ground truth, prediction and predictive entropy are visualized.
- Black: minimum uncertainty, white: maximum uncertainty.

Outline

UPPSALA UNIVERSITET

6. Discussion

UPPSALA UNIVERSITET

- The results of our comparison suggest that **ensembling** consistently provides more reliable and useful uncertainty estimates than MC-dropout.

6. Discussion

- The results of our comparison suggest that **ensembling** consistently provides more reliable and useful uncertainty estimates than MC-dropout.
- MC-dropout has a large design-space compared to ensembling, and while careful tuning of MC-dropout potentially could close the performance gap, the simplicity and general applicability of ensembling must be considered key **strengths**.

6. Discussion

- The results of our comparison suggest that **ensembling** consistently provides more reliable and useful uncertainty estimates than MC-dropout.
- MC-dropout has a large design-space compared to ensembling, and while careful tuning of MC-dropout potentially could close the performance gap, the simplicity and general applicability of ensembling must be considered key **strengths**.
- The main **drawback** of both methods is the computational cost at test time that grows linearly with M, limiting real-time applicability.

6. Discussion

- The results of our comparison suggest that **ensembling** consistently provides more reliable and useful uncertainty estimates than MC-dropout.
- MC-dropout has a large design-space compared to ensembling, and while careful tuning of MC-dropout potentially could close the performance gap, the simplicity and general applicability of ensembling must be considered key **strengths**.
- The main **drawback** of both methods is the computational cost at test time that grows linearly with M, limiting real-time applicability.
    - Here, **future work** includes exploring the effect of model pruning techniques on predictive uncertainty quality.

6. Discussion

- The results of our comparison suggest that **ensembling** consistently provides more reliable and useful uncertainty estimates than MC-dropout.
- MC-dropout has a large design-space compared to ensembling, and while careful tuning of MC-dropout potentially could close the performance gap, the simplicity and general applicability of ensembling must be considered key **strengths**.
- The main **drawback** of both methods is the computational cost at test time that grows linearly with M, limiting real-time applicability.
  - Here, **future work** includes exploring the effect of model pruning techniques on predictive uncertainty quality. For ensembling, sharing early stages of the DNN among ensemble members is also an interesting future direction.

6. Discussion

- The results of our comparison suggest that **ensembling** consistently provides more reliable and useful uncertainty estimates than MC-dropout.
- MC-dropout has a large design-space compared to ensembling, and while careful tuning of MC-dropout potentially could close the performance gap, the simplicity and general applicability of ensembling must be considered key **strengths**.
- The main **drawback** of both methods is the computational cost at test time that grows linearly with M, limiting real-time applicability.
  - Here, **future work** includes exploring the effect of model pruning techniques on predictive uncertainty quality. For ensembling, sharing early stages of the DNN among ensemble members is also an interesting future direction.
- A **weakness** of ensembling is the additional training required, which also scales linearly with M. The training of different ensemble members can however be performed in parallel, making it less of an issue in practice given appropriate computing infrastructure.

Outline

UPPSALA UNIVERSITET

7. 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 in the case of DNNs.

7. 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 in the case of 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.

7. 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 in the case of 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.

References

[1] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[2] T. Chen, E. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning (ICML)*, pages 1683–1691, 2014.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[4] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[7] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Bro. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018.

[8] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[9] F. Ma, G. V. Cavalheiro, and S. Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[10] Y.-A. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2917–2925, 2015.

[11] R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2:113–162, 2011.

[12] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant CNNs. In *International Conference on 3D Vision (3DV)*, 2017.

[13] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, pages 681–688, 2011.

[14] M. Wrenninge and J. Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018.

Contact

Fredrik K. Gustafsson, Uppsala University

fredrik.gustafsson@it.uu.se

www.fregu856.com