

Towards scalable uncertainty estimation with deterministic methods, and their fair evaluation

Xuanlong Yu^{1,2}

¹ SATIE, Paris-Saclay University ² U2IS, ENSTA Paris, IP Paris

May 4, 2023

université
PARIS-SACLAY



Papers and co-authors

Latent Discriminant deterministic Uncertainty

Authors: Gianni Franchi*¹, Xuanlong Yu*^{1,2}, Andrei Bursuc³, Emanuel Aldea²,
Séverine Dubuisson⁴, David Filliat¹

17th European Conference on Computer Vision (ECCV), 2022

MUAD: Multiple Uncertainties for Autonomous Driving, a benchmark for multiple uncertainty types and tasks

Authors: Gianni Franchi*¹, Xuanlong Yu*^{1,2}, Andrei Bursuc³, Ángel Tena⁵,
Rémi Kazmierczak¹, Séverine Dubuisson⁴, Emanuel Aldea², David Filliat¹

33rd British Machine Vision Conference (BMVC), 2022

¹ U2IS, ENSTA Paris, IP Paris, ² SATIE, Paris-Saclay University, ³ valeo.ai, ⁴ CNRS, LIS, Aix Marseille University, ⁵ Anyverse

Presentation Overview

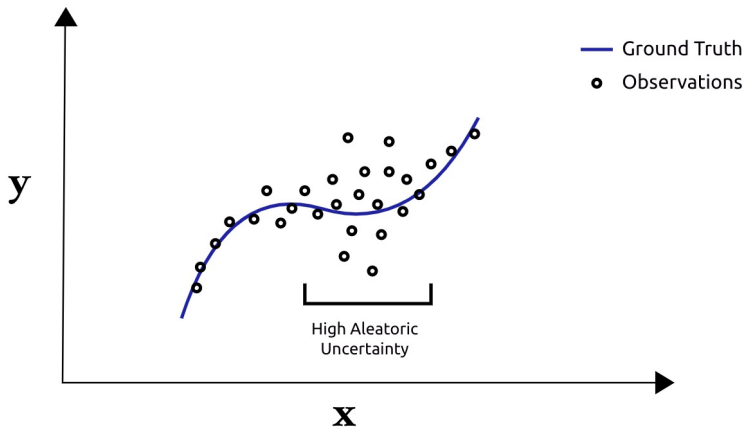
- 1 Uncertainty and Deep learning
- 2 Quantifying DNNs' uncertainty
- 3 Quantifying DNNs' uncertainty with LDU
 - Introduction on LDU
 - Experiments
- 4 Evaluating uncertainty quantification with MUAD
- 5 Conclusions

Types of Uncertainty

Aleatoric uncertainty: uncertainty inherent in the observation noise (problems caused by sensor quality, natural randomness, that cannot be explained by our data).

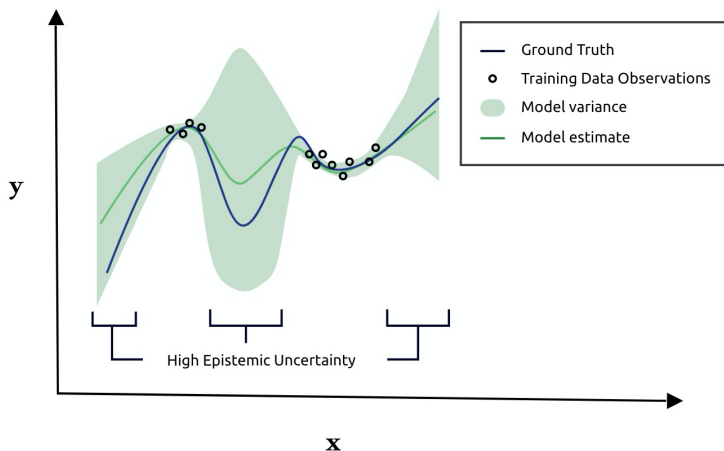
Epistemic uncertainty: our ignorance about the correct model that generated the data (lack of knowledge about the process that generated the data).

What is uncertainty in machine/deep learning¹



¹Credits: Huy Nguyen

What is uncertainty in machine/deep learning²



²Credits: Huy Nguyen

Bayesian DNN

Bayesian DNN (Blundell et al., 2015)³ is based on marginalization instead of MAP optimization.

$$\begin{aligned}\mathcal{P}(Y|X) &= \mathbb{E}_{\omega \sim \mathcal{P}(\omega|D_I)} (\mathcal{P}(Y|X, \omega)) \\ \mathcal{P}(Y|X) &= \int \mathcal{P}(Y|X, \omega) \mathcal{P}(\omega|D_I) d\omega\end{aligned}$$

In practice:

$$\mathcal{P}(Y|X) \simeq \frac{1}{N_{\text{model}}} \sum_{i=1}^{N_{\text{model}}} (\mathcal{P}(Y|X, \omega_i)) \text{ with } \omega_i \sim \mathcal{P}(\omega|D_I)$$

Intractability : different techniques to estimate $\mathcal{P}(\omega|D_I)$.

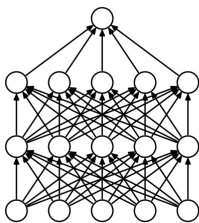
³Charles Blundell et al. (2015). "Weight uncertainty in neural network". In: *ICML*.

MC dropout

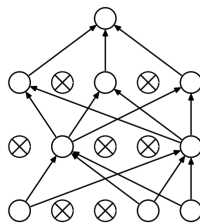
In MC Dropout (Gal and Ghahramani, 2016)⁴, the authors propose to average the predictions of several DNNs where they apply dropout across the model:

$$\mathcal{P}(y^*|x^*) = \frac{1}{N_{\text{model}}} \sum_{j=1}^{N_{\text{model}}} \mathcal{P}(y^*|\omega(t^*) \odot b^j, x^*) \quad (1)$$

with b^j a vector of the same size of $\omega(t^*)$ which is a realization of a binomial distribution.



(a) Standard Neural Net



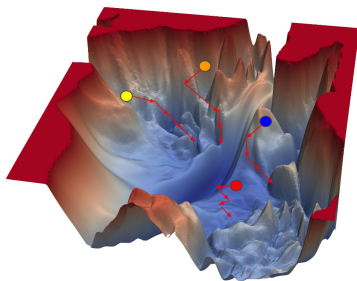
(b) After applying dropout.

⁴Yarin Gal and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *ICML*.

Deep Ensembles

In Deep Ensembles (Lakshminarayanan, Pritzel, and Blundell, 2017)⁵, the authors propose to average the predictions of several DNNs with different initial seeds:

$$\mathcal{P}(y^*|x^*) = \frac{1}{N_{\text{model}}} \sum_{j=1}^{N_{\text{model}}} \mathcal{P}(y^*|\omega^j(t^*), x^*) \quad (2)$$

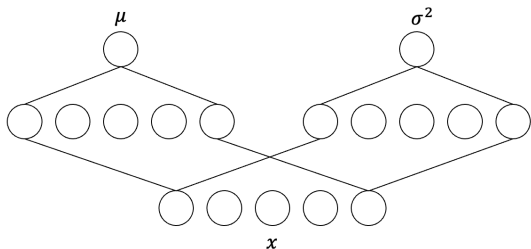


⁵Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *NeurIPS*.

Learning Gaussian parameters

For regression tasks, the authors of (Kendall and Gal, 2017; Nix and Weigend, 1994)⁶ propose to model the outputs of the DNN as the parameters of Gaussian distribution given an input x . The likelihood function is as follows:

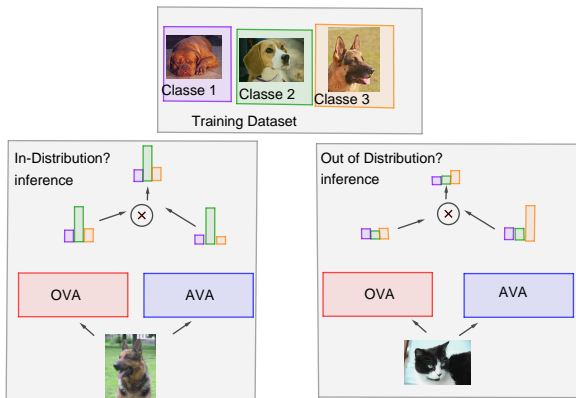
$$\mathcal{P}(y|x, \omega) = \frac{1}{\sqrt{2\pi\sigma^2(x)}} \exp \frac{-[y - \mu(x)]^2}{2\sigma^2(x)} \quad (3)$$



⁶Alex Kendall and Yarin Gal (2017). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *NeurIPS*; D.A. Nix and A.S. Weigend (1994). "Estimating the mean and variance of the target probability distribution". In: *ICNN*.

OVNNI

We notice in OVNNI (Franchi et al., 2021)⁷ that an ensemble of DNNs trained to classify One class vs All the other classes (OVA) quantifies the uncertainty better.



⁷ Gianni Franchi et al. (2021). "One Versus all for deep Neural Network Incertitude (OVNNI) quantification". In: *IEEE Access*.

OVADM

In the traditional DNN, the **logit outputs** of a neural network are calculated from the latent space embeddings through an affine transformation $f_{\omega}(x)_k = w_k^T h_{\omega}(x) + b_k$. The probability distribution is then calculated through the softmax normalization :

$$P(y_k|x, \omega) = \frac{\exp(w_k^T h_{\omega}(x) + b_k)}{\sum_k \exp(w_k^T h_{\omega}(x) + b_k)}$$

In OVADM (Padhy et al., 2020)⁸, the authors propose to use a **Distinction Maximization** logit, hence :

$$f_{\omega}(x)_k = -\|h_{\omega}(x) - w_k\|$$

and they also use an OVA training strategy :

$$P(y_k|x, \omega) = \frac{2}{1 + \exp(-f_{\omega}(x)_k)} = \frac{2}{1 + \exp(\|h_{\omega}(x) - w_k\|)}$$

⁸Shreyas Padhy et al. (2020). "Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks". In: *ICML Workshops*.

Deterministic Uncertainty Methods (DUMs)

The authors of (Van Amersfoort et al., 2020)⁹ consider that a DNN $f_{\omega}(\cdot)$ with trainable parameters ω is composed of two main blocks: a feature extractor h_{ω} and a head g_{ω} , such that $f_{\omega}(x) = (g_{\omega} \circ h_{\omega})(x)$

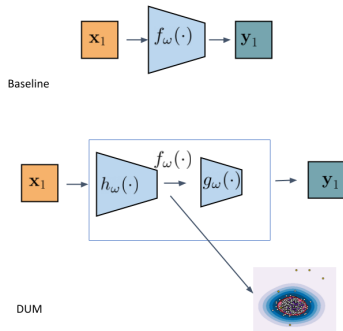


Figure: An illustration of Deterministic Uncertainty Methods

⁹Joost Van Amersfoort et al. (2020). "Uncertainty estimation using a single deep deterministic neural network". In: *ICML*.

Deterministic Uncertainty Methods (DUMs)

$h_\omega(x)$ computes a latent representation from x .

To avoid feature collapse (Van Amersfoort et al., 2020)¹⁰, they consider that $h_\omega(x)$ is a bi-Lipschitz DNN which implies that for any pair of inputs x_1 and x_2 from \mathcal{X} :

$$L_1 \|x_1 - x_2\| \leq \|h_\omega(x_1) - h_\omega(x_2)\| \leq L_2 \|x_1 - x_2\| \quad (4)$$

¹⁰Joost Van Amersfoort et al. (2020). "Uncertainty estimation using a single deep deterministic neural network". In: *ICML*.

Latent Discriminant Deterministic Uncertainty (LDU)

We denote $z \in \mathbb{R}^n$ the latent representation of dimension n of x , i.e., $z = h_\omega(x)$, that is given as input to the **Distinction Maximization (DM)** layer. Given a set $p_\omega = \{p_i\}_{i=1}^m$, of m vectors ($p_i \in \mathbb{R}^n$) that are trainable, we define the DM layer as follows:

$$\text{DM}_\rho(z) = [-\|z - p_1\|, \dots, -\|z - p_m\|]^\top \quad (5)$$

Latent Discriminant Deterministic Uncertainty (LDU)

Our DNN can be written as:

$$f_{\omega}(x) = [g_{\omega} \circ (\exp(-DM_p(h_{\omega})))](x) \quad (6)$$

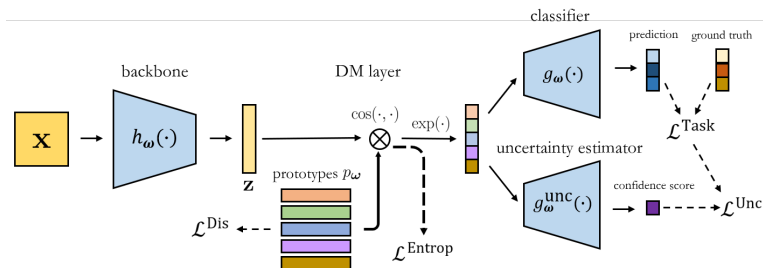


Figure: Overview of LDU

Latent Discriminant Deterministic Uncertainty (LDU)

Our training loss is equal to:

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{Task}} + \lambda(\mathcal{L}^{\text{Entrop}} + \mathcal{L}^{\text{Dis}} + \mathcal{L}^{\text{Unc}}) \quad (7)$$

We add a loss to force the prototypes to be dissimilar:

$$\mathcal{L}^{\text{Dis}} = - \sum_{i < j} \|p_i - p_j\|.$$

We also add one loss to constrain the latent representation to stay close to different prototypes:

$$\mathcal{L}^{\text{Entrop}} = \sum_{i=1}^n \sigma(\text{DM}_p(h_\omega))_i \cdot \log(\sigma(\text{DM}_p(h_\omega))_i),$$

Latent Discriminant Deterministic Uncertainty (LDU)

We propose to train g_{ω}^{unc} to predict the error of the DNN (Corbière et al., 2019; Yu, Franchi, and Aldea, 2021)¹¹, which helps us link the prototypes to the uncertainty.

Given an input data x , its groundtruth y (y can be a scalar or a vector if we deal with regression) and, its loss $\mathcal{L}^{\text{Task}}(g_{\omega}(x), y)$, we train g_{ω}^{unc} by minimizing:

$$\mathcal{L}^{\text{Unc}} = \text{BCE}([g_{\omega}^{\text{unc}} \circ (\exp(-DM_{\rho}(h_{\omega})))](x), \mathcal{L}^{\text{Task}}(g_{\omega}(x), y)),$$

¹¹Charles Corbière et al. (2019). "Addressing failure prediction by learning model confidence". In: *NeurIPS*; Xuanlong Yu, Gianni Franchi, and Emanuel Aldea (2021). "SLURP: Side Learning Uncertainty for Regression Problems". In: *BMVC*.

Experiments

Toy example

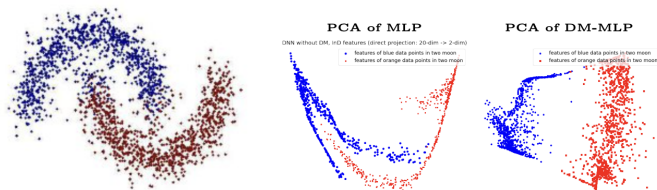


Figure: PCA 2D projection on the left of a standard MLP and on the right of a DM-MLP trained on the two moons dataset. Blue and red points indicate the features of data points of the two classes, respectively.

Evaluation metrics for uncertainty quantification

Classification tasks:

- Calibration: ECE (Expected Calibration Error (Guo et al., 2017)¹²); **lower is better**
- OOD detection: AUC and AUPR; **higher is better**

Regression tasks:

- AUSE (Area Under Sparsification Error curve); **lower is better**

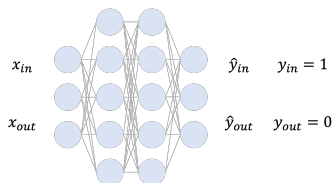
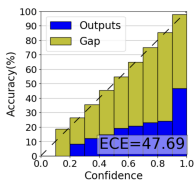


Figure: Left: an example for ECE; Right: OOD detection and evaluation.

¹²Chuan Guo et al. (2017). "On calibration of modern neural networks". In: *ICML*.

Classification tasks



Figure: Datasets used in OOD detection task in classification. Left: CIFAR10 training set, Right: SVHN evaluation set.

Classification results

Method	CIFAR-10			
	Acc \uparrow	AUC \uparrow	AUPR \uparrow	ECE \downarrow
Baseline (MCP)	88.02	0.8032	0.8713	0.5126
MCP lipz.	88.50	0.8403	0.9058	0.3820
Deep Ensembles	89.96	0.8513	0.9087	0.4249
SNGP	88.45	0.8447	0.9139	0.4254
DUQ	89.9	0.8446	0.9144	0.5695
DUE	87.54	0.8434	0.9082	0.4313
DDU	87.87	0.8199	0.8754	0.3820
MIR	87.95	0.7574	0.8556	0.4004
LDU #p = 128	87.95	0.8721	0.9147	0.4933
LDU #p = 64	88.06	0.8625	0.9070	0.5010
LDU #p = 32	87.83	0.8129	0.8900	0.5264
LDU #p = 16	88.33	0.8479	0.9094	0.4975

Table: Comparative results for **image classification tasks**. We evaluate on CIFAR-10 for the tasks: in-domain classification, and out-of-distribution detection with SVHN. Results are averaged over three seeds.

Semantic segmentation task

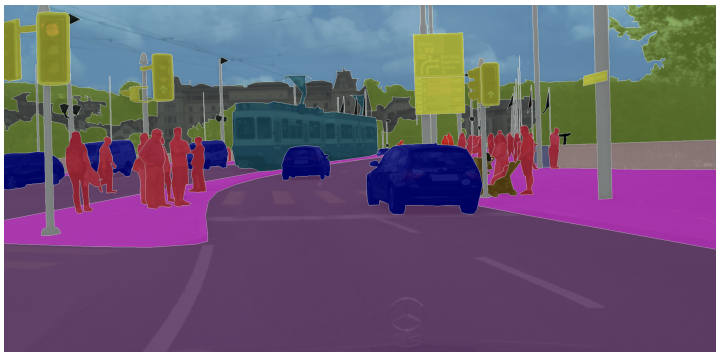


Figure: An example from the Cityscapes dataset.

Semantic segmentation results

Method	Cityscapes		Cityscapes-C lvl 1		Cityscapes-C lvl 2		Cityscapes-C lvl 3		Cityscapes-C lvl 4		Cityscapes-C lvl 5	
	mIoU \uparrow	ECE \downarrow	mIoU \uparrow	ECE \downarrow	mIoU \uparrow	ECE \downarrow	mIoU \uparrow	ECE \downarrow	mIoU \uparrow	ECE \downarrow	mIoU \uparrow	ECE \downarrow
Baseline (MCP)	76.84%	0.1180	51.59%	0.1793	41.45%	0.2291	35.67%	0.2136	30.12%	0.1970	24.84%	0.2131
Baseline (MCP) lipz.	58.38%	0.1037	44.70%	0.1211	38.04%	0.1475	32.70%	0.1802	25.35%	0.2047	18.36%	0.2948
MC-Dropout	71.88%	0.1157	53.61%	0.1501	42.02%	0.2531	35.91%	0.1718	29.52%	0.1947	25.61%	0.2184
Deep Ensembles	77.23%	0.1139	54.98%	0.1422	44.63%	0.1902	38.00%	0.1851	32.14%	0.1602	28.74%	0.1729
LDU (ours)	76.62%	0.0893	52.00%	0.1371	43.02%	0.1314	37.17%	0.1702	32.27%	0.1314	27.30%	0.1712

Table: Comparative results for semantic segmentation on Cityscapes and Cityscapes-C (Hendrycks and Dietterich, 2019)¹³.

¹³ Dan Hendrycks and Thomas Dietterich (2019). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *ICLR*.

Monocular depth task

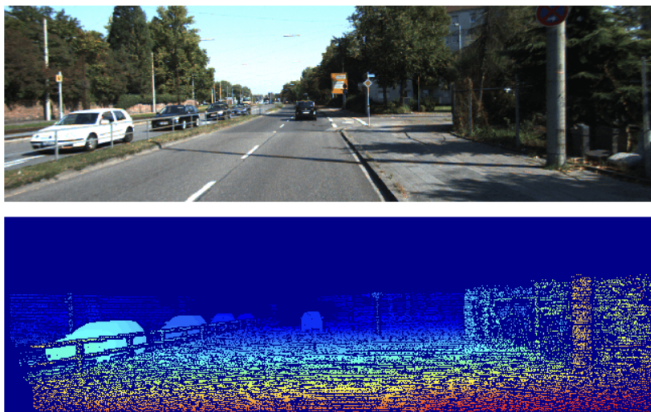


Figure: An example from the **KITTI dataset**. Upper: RGB Image; Lower: depth ground truth by LIDAR.

Monocular depth results

Method	Depth performance								Uncertainty performance	
	d1 \uparrow	d2 \uparrow	d3 \uparrow	Abs Rel \downarrow	Sq Rel \downarrow	RMSE \downarrow	RMSE log \downarrow	log10 \downarrow	AUSE RMSE \downarrow	AUSE Absrel \downarrow
Baseline	0.955	0.993	0.998	0.060	0.249	2.798	0.096	0.027	-	-
Deep Ensembles	0.956	0.993	0.999	0.060	0.236	2.700	0.094	0.026	0.08	0.21
MC-Dropout	0.945	0.992	0.998	0.072	0.287	2.902	0.107	0.031	0.46	0.50
Single-PU	0.949	0.991	0.998	0.064	0.263	2.796	0.101	0.029	0.08	0.21
Infer-noise	0.955	0.993	0.998	0.060	0.249	2.798	0.096	0.027	0.33	0.48
LDU #p = 5, $\lambda = 1.0$	0.954	0.993	0.998	0.063	0.253	2.768	0.098	0.027	0.08	0.21
LDU #p = 15, $\lambda = 0.1$	0.954	0.993	0.998	0.062	0.249	2.769	0.098	0.027	0.10	0.28
LDU #p = 30, $\lambda = 0.1$	0.955	0.992	0.998	0.061	0.248	2.757	0.097	0.027	0.09	0.26

Table: Comparative results for monocular depth estimation on KITTI eigen-split validation set.

Evaluating uncertainty quantification with MUAD

Overview of the different datasets for uncertainty on autonomous driving:

Dataset	Adversarial annotations	Fog	Night	Rain	Snow	Classes	Out of distribution	Depth	Object detection 2D/3D	Instance segmentation
Foggy Driving	101	✓	-	-	-	19	-	-	✓	-
Foggy Zurich	40	✓	-	-	-	19	-	-	-	-
Nighttime Driving	50	-	✓	-	-	19	-	-	-	-
Dark Zurich	201	-	✓	-	-	19	-	-	-	-
Raincover	326	-	✓	✓	-	3	-	-	-	-
WildDash	226	✓	✓	✓	✓	19	-	-	-	-
BDD100K	1346	✓	✓	✓	✓	19	-	-	-	-
ACDC	4006	✓	✓	✓	✓	19	-	✓	✓	-
Virtual KITTI 2	21260	✓	-	✓	-	14	-	✓	✓	✓
Fishyscapes	373	-	-	-	-	19+2	✓	-	-	-
LostAndFound	1203	-	-	-	-	19+9	✓	-	-	-
RoadObstacle21	327	-	✓	-	✓	19+1	✓	-	-	-
RoadAnomaly21	100	-	-	-	✓	19+1	✓	-	-	-
Streethazard	6625	-	-	-	-	13+250	✓	-	-	-
BDD anomaly	810	✓	✓	✓	✓	17+2	✓	-	-	-
MUAD	10413	✓	✓	✓	✓	16+9	✓	✓	✓	✓

Table: Comparative overview of the different datasets for uncertainty on autonomous driving.

MUAD dataset

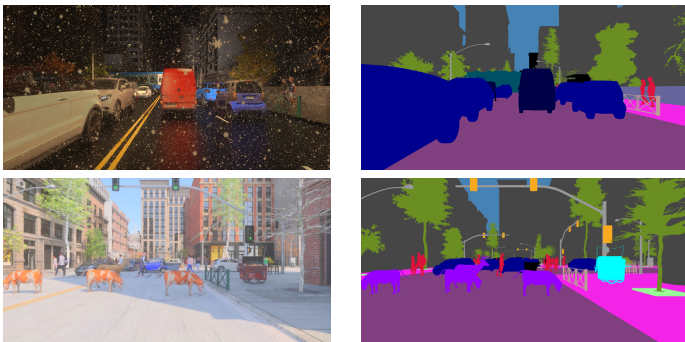


Figure: Snapshots from the MUAD dataset showing different types of adverse conditions and events to evaluate perception models.

MUAD dataset

10413 annotated images: 3420 images in the train set, 492 in the validation set, and 6501 in the test set. 2/3 being day images and 1/3 night images.

3 types of adversity conditions with 2 intensity levels: **Fog, Rain, Snow.**

21 classes: 19 ID classes (same as Cityscapes), 2 OOD classes (object anomalies and animals).

7 test sets: Normal sets, Normal set overhead sun, OOD set, Low adv. Set High adv. Set, Low adv. with OOD set, High adv. with OOD set.

4 supported tasks: Semantic segmentation, Depth estimation, Object detection 2D/3D, Instance segmentation.

Semantic segmentation results

Method	normal set		OOD set				low adv. set				high adv. set						
	mIoU \uparrow	ECE \downarrow	mIoU \uparrow	ECE \downarrow	AUROC \uparrow	AUPR \uparrow	FPR \downarrow	mIoU \uparrow	ECE \downarrow	AUROC \uparrow	AUPR \uparrow	FPR \downarrow	mIoU \uparrow	ECE \downarrow	AUROC \uparrow	AUPR \uparrow	FPR \downarrow
Baseline (MCP)	68.90%	0.0138	57.32%	0.0607	0.8624	0.2604	0.3943	31.84%	0.3078	0.6349	0.1185	0.6746	18.94%	0.4356	0.6023	0.1073	0.7547
Baseline (MCP) lipz.	53.96%	0.01398	45.97%	0.0601	0.8419	0.2035	0.3940	16.79%	0.3336	0.6303	0.1051	0.7262	7.8%	0.4244	0.5542	0.0901	0.8243
MIR	53.96%	0.01398	45.97%	0.0601	0.6223	0.1469	0.8406	16.79%	0.3336	0.5143	0.1035	0.8708	7.8%	0.4244	0.4470	0.0885	0.9093
MC-Dropout	65.33%	0.0173	55.62%	0.0645	0.8439	0.2225	0.4575	33.38%	0.1329	0.7506	0.1545	0.5807	20.77%	0.3809	0.6864	0.1185	0.6751
Deep Ensembles	69.80%	0.01296	58.29%	0.0588	0.871	0.2802	0.3760	34.91%	0.2447	0.6543	0.1212	0.6425	20.19%	0.4227	0.6101	0.1162	0.7212
LDU (ours)	69.32%	0.01356	58.29%	0.0594	0.8816	0.4418	0.3548	36.12%	0.2674	0.7779	0.2898	0.5381	21.15%	0.4231	0.7107	0.2186	0.6412

Table: Comparative results for semantic segmentation on MUAD.

Semantic segmentation results

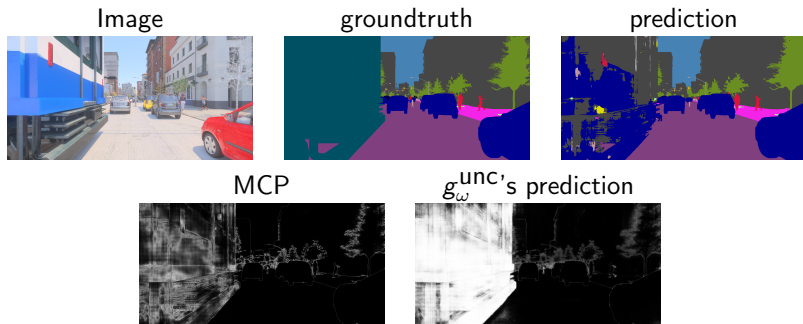


Figure: Illustration of the different confidence scores on one image of MUAD. Note that the class train, bicycle, Stand food and the animals are OOD.

Semantic segmentation results

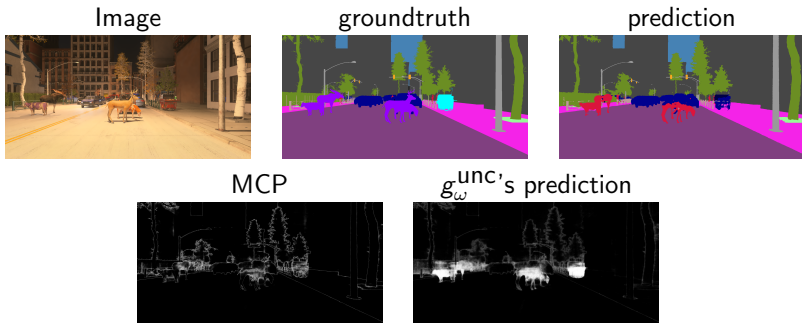


Figure: Illustration of the different confidence scores on one image of MUAD. Note that the class train, bicycle, Stand food and the animals are OOD.

Conclusions

Scalable uncertainty estimation with deterministic method:

We propose LDU: a modification for DNNs to better estimate their predictive uncertainty.

LDU reaches results comparable to Deep Ensembles with less computational cost.

MUAD dataset:

We provide a new synthetic dataset with multiple uncertainty sources for autonomous driving.

Other information

Latent Discriminant deterministic Uncertainty

<https://arxiv.org/abs/2207.10130>















MUAD: Multiple Uncertainties for Autonomous Driving, a benchmark for multiple uncertainty types and tasks

<https://muad-dataset.github.io/>



All references:

-  [Blundell, Charles et al. \(2015\)](#). “Weight uncertainty in neural network”. In: *ICML*.
-  [Corbière, Charles et al. \(2019\)](#). “Addressing failure prediction by learning model confidence”. In: *NeurIPS*.
-  [Franchi, Gianni et al. \(2021\)](#). “One Versus all for deep Neural Network Incertitude (OVNNI) quantification”. In: *IEEE Access*.
-  [Gal, Yarin and Zoubin Ghahramani \(2016\)](#). “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *ICML*.
-  [Guo, Chuan et al. \(2017\)](#). “On calibration of modern neural networks”. In: *ICML*.
-  [Hendrycks, Dan and Thomas Dietterich \(2019\)](#). “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *ICLR*.
-  [Kendall, Alex and Yarin Gal \(2017\)](#). “What uncertainties do we need in bayesian deep learning for computer vision?” In: *NeurIPS*.

-  Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2017). “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *NeurIPS*.
-  Nix, D.A. and A.S. Weigend (1994). “Estimating the mean and variance of the target probability distribution”. In: *ICNN*.
-  Padhy, Shreyas et al. (2020). “Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks”. In: *ICML Workshops*.
-  Van Amersfoort, Joost et al. (2020). “Uncertainty estimation using a single deep deterministic neural network”. In: *ICML*.
-  Yu, Xuanlong, Gianni Franchi, and Emanuel Aldea (2021). “SLURP: Side Learning Uncertainty for Regression Problems”. In: *BMVC*.