MUAD: Multiple Uncertainties for Autonomous Driving, a benchmark for multiple uncertainty types and tasks (Supplementary Material)

Gianni Franchi[1†]
gianni.franchi@ensta-paris.fr
Xuanlong Yu[1,2†]
xuanlong.yu@universite-paris-saclay.fr
Andrei Bursuc[3]
andrei.bursuc@valeo.com
Angel Tena[4]
angel.tena@anyverse.ai
Rémi Kazmierczak[1]
remi.kazmierczak@ensta-paris.fr
Séverine Dubuisson[5]
severine.dubuisson@lis-lab.fr
Emanuel Aldea[2]
emanuel.aldea@universite-paris-saclay.fr
David Filliat[1]
david.filliat@ensta-paris.fr

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

Contents

A Multiple Uncertainties for Autonomous Driving benchmark (MUAD)

A.1 Uncertainty and Deep Learning

A DNN is a function f_θ parameterized by a set of parameters θ that takes input data x and outputs a prediction y. The DNN is trained on a training dataset composed of a set $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, with N being the number of data to optimize the parameters θ for a task. Once the DNN is trained, meaning that the optimization of θ on $\mathcal{D}$ is completed, f_θ may be used for inference on new data x^*.

Uncertainty on deep learning may arise mainly from three factors [6]. Firstly it can result from the data acquisition process which introduces some noise. This might be due to the variability in real-world situations. For example, one records training data in certain weather conditions, which subsequently change during inferences. The measurement systems might also introduce errors such as sensor noise. Secondly, uncertainty may result from the DNN building and training process. DNNs are random functions whose parameters θ are initialized randomly and whose training procedure relies on stochastic optimization. Therefore, the resulting neural network is a random function that is most of the time related to a local minimum of the expected loss function (which we denote as the risk). Hence this source of randomness might cause errors in the training procedure of the DNN. Thirdly, the last uncertainty factor is related to the DNN's prediction's uncertainty. Uncertainty could come from the lack of knowledge of the DNN and might be caused by unknown test data.

Based on these factors, we can divide the uncertainty into two kinds: the aleatoric uncertainty and the epistemic uncertainty. The aleatoric uncertainty can be subdivided into two kinds: In-domain uncertainty [2] and Domain-shift uncertainty [13]. In-domain uncertainty occurs when the test data is sampled from the training distribution and is related to the inability of the deep neural network to predict a proper confidence score about the quality of its predictions due to a lack of in-domain knowledge. Domain-shift uncertainty denotes the uncertainty related to an input drawn from a shifted version of the training distribution. Hence, it is caused by the fact the distribution of the training dataset might not encompass enough variability. These two kinds of uncertainty can be reduced by increasing the number of the training dataset. Epistemic uncertainty denotes the uncertainty when the test data is sampled from a distribution that is different and far from the training distribution. Epistemic uncertainty can be categorized into two kinds namely [15] : approximation uncertainty and model uncertainty. The approximation uncertainty is linked to the fact that we optimize the empirical risk instead of the risk. Hence, the optimal DNN's parameters approximate the optimal DNN's parameters of the true risk function. The model uncertainty is linked to the fact that our loss function provides us with a space of solutions that might not include the perfect predictor. For example, the DNN might have different classes between the training and testing set. In this context 'Out of Distribution' samples refers to anomalies in the test set that are data from classes not present in the training set.

B Extra Monocular depth experiments

B.1 Implementation and criterion

Implementation. We train the NeWCRFs [17] model using the same hyperparameters and

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓			AURG↑		
										AbsRel	RMSE	d1	AbsRel	RMSE	d1
Baseline	13.9767	0.1143	0.0444	3.3575	0.5571	0.1443	0.9219	0.9833	0.9933	-	-	-	-	-	-
Deep Ensembles [10]	13.6691	0.1110	0.0419	3.1994	0.6076	0.1400	0.9289	0.9843	0.9945	0.0604	0.2906	0.0431	0.0117	2.4618	0.0215
MC Dropout [5]	13.5602	0.1194	0.0447	3.2090	0.6897	0.1453	0.9193	0.9847	0.9941	0.0610	0.6339	0.0542	0.0161	2.0846	0.0193
Single-PU [9]	14.5896	0.1324	0.0484	3.2298	0.7738	0.1547	0.9054	0.9803	0.9933	0.0807	0.3131	0.0837	0.0042	2.4194	-0.0005
SLURP [16]	13.9767	0.1143	0.0444	3.3575	0.5571	0.1443	0.9219	0.9833	0.9933	0.0477	0.4672	0.0459	0.0252	2.3870	0.0237

Table 5: Supervised monocular depth results on **normal set**.

image augmentation parameters used in the official paper for training on KITTI [7], except that we change the batch size to 4 and randomly crop the input image to 512*1024. For the Single-PU [9] models, we perform a multi task training where we train to predict the depth map with the silog loss function provided in the NeWCRFs [17] paper, and we minimize the negative Gaussian log-likelihood loss in order to train to predict the variance. To train the DNN that will predict the variance, we do not optimize the layers that are used to predict the depth map, as explained in [1], as this stabilizes the training. Regarding MC-Dropout [5], we let the dropout layers activated during the inference and perform eight forward passes for each input data during inference and average the predictions. We want to point out that we did not add any additional dropout layers to the model to keep the paper's performance. For the SLURP [16] models, we use the base model as the main task model and train an auxiliary uncertainty estimator. We use the Swin Transformer [12] as used in the base model as an encoder for the auxiliary model and train the auxiliary model for 20 epochs.

Evaluation metrics. To evaluate depth estimations, we use the same metrics as Eigen *et al.* [4] which are standard criteria [11, 17, 18]. For uncertainty quantification evaluation metrics, we use the criteria implementation of Poggi *et al.* [14]: Area Under the Sparsification Error (AUSE) and Area Under the Random Gain (AURG). The Area Under the Sparsification Error is obtained by calculating the difference between the sparsification curve and the oracle sparsification curve. The sparsification curve is obtained by continuously erasing 1% pixels according to the predicted uncertainty and calculating the prediction error for the rest pixels. We can also have an oracle sparsification curve by continuously erasing pixels according to their prediction error. The total difference between the two curves is AUSE. We can evaluate the AUSE for different error metrics such as RMSE, Absrel, and d1, which provide us AUSE-RMSE, AUSE-Absrel, and AUSE-d1. AURG is achieved by calculating the area between the Sparsification curve and a random curve to measure how good the uncertainty estimator is compared to no modeling cases. Similarly, we can achieve AURG-RMSE, AURG-Absrel, and AURG-d1 using different error metrics.

B.2 Full results on supervised monocular depth estimation

In the main paper, due to the space constrain, we can only provide partial results for depth and uncertainty metrics, we here provide full results from Table 5 to Table 11 for different uncertainty quantification solutions introduced in the main paper applied on supervised monocular depth estimation task. Overall, the Deep Ensembles [10] and SLURP [16] can provide better uncertainty estimations on the test sets without perturbations. When weather perturbations exist, MC-Dropout [5] and Deep Ensembles [10] perform better on uncertainty quantification. MC-Dropout can also provide better depth estimations than the other solutions under weather perturbations.

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓ AbsRel	AUSE↓ RMSE	AUSE↓ d1	AURG↑ AbsRel	AURG↑ RMSE	AURG↑ d1
Baseline	19.8427	0.1474	0.0757	5.0053	0.8301	0.2397	0.7861	0.9244	0.9613	-	-	-	-	-	-
Deep Ensembles [illegible]	22.7950	0.1564	0.0850	4.8919	0.8508	0.2759	0.7673	0.9010	0.9419	0.1047	0.7401	0.1823	-0.0103	3.1624	0.0023
MC Dropout [illegible]	21.6959	0.1505	0.0765	4.5799	0.7648	0.2459	0.7980	0.9199	0.9543	0.0980	1.0627	0.1473	-0.0074	2.5851	0.0182
Single-PU [illegible]	24.2069	0.1588	0.0849	4.8648	0.8522	0.2800	0.7727	0.8997	0.9417	0.1115	0.7892	0.1863	-0.0145	3.1099	-0.0025
SLURP [illegible]	19.8429	0.1474	0.0757	5.0053	0.8301	0.2397	0.7861	0.9244	0.9613	0.0898	1.1665	0.1789	-0.0040	2.8036	-0.0037

Table 6: Supervised monocular depth results on **low adv. without OOD set**.

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓ AbsRel	AUSE↓ RMSE	AUSE↓ d1	AURG↑ AbsRel	AURG↑ RMSE	AURG↑ d1
Baseline	27.2917	0.2072	0.1148	6.9890	1.5990	0.3603	0.6316	0.8275	0.9028	-	-	-	-	-	-
Deep Ensembles [illegible]	34.7624	0.2429	0.1478	7.4977	1.9794	0.4674	0.5657	0.7643	0.8507	0.1529	1.1824	0.3031	-0.0117	4.6140	-0.0044
MC Dropout [illegible]	30.5442	0.2073	0.1142	6.2782	1.3762	0.3652	0.6567	0.8292	0.8992	0.1277	1.3819	0.2169	-0.0055	3.5187	0.0393
Single-PU [illegible]	41.9847	0.2480	0.1588	7.6797	2.1362	0.5295	0.5708	0.7586	0.8435	0.1706	1.7402	0.3318	-0.0220	4.2634	-0.0322
SLURP [illegible]	27.2917	0.2072	0.1148	6.9890	1.5990	0.3603	0.6316	0.8275	0.9028	0.1281	1.7066	0.2740	-0.0100	3.7188	-0.0024

Table 7: Supervised monocular depth results on **high adv. without OOD set**.

B.3 Self-supervised monocular depth estimation

In this section, we provide the self-supervised monocular depth results for MUAD. In order to provide a wider variety of urban scenarios, there are no consecutive frames in MUAD, but still provides pictures taken by the left and right cameras. We provide self-supervised monocular depth results on MUAD in Table 12 using DIFFNet [18] and left-right consistency [8] strategy. DIFFNet is one of the SOTA on KITTI outdoor dataset [7]. We train a DIFFNet model with 12 images as the batch size, randomly crop the image to 512*1024, and train 20 epochs in total.

We observe that OOD objects have less impact on the results of monocular depth estimation in the Self-supervised monocular depth. According to [3], monocular depth estimation based on left-right coherence is sensitive to illumination conditions, particularly to object shadows. However, our results on the *Normal set* and *Overhead sun set* do not seem to confirm this point. We believe that DNNs learn depth without necessarily paying much attention to shadows; hence they have no impact on the performance of the self-supervised monocular depth model.

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓			AURG↑		
										AbsRel	RMSE	d1	AbsRel	RMSE	d1
Baseline	12.4227	0.0895	0.0387	3.6461	0.4083	0.1257	0.9513	0.9909	0.9969	-	-	-	-	-	-
Deep Ensembles []	11.7212	0.0829	0.0351	3.4788	0.3867	0.1188	0.9553	0.9903	0.9967	0.0553	0.3363	0.0098	-0.0041	2.6248	0.0336
MC Dropout []	12.0129	0.0915	0.0389	3.4074	0.3888	0.1263	0.9475	0.9902	0.9969	0.0576	0.7856	0.0308	-0.0019	2.0452	0.0199
Single-PU []	12.4754	0.1052	0.0437	3.5463	0.4210	0.1344	0.9461	0.9895	0.9966	0.0788	0.3576	0.0308	-0.0189	2.5430	0.0212
SLURP []	12.4227	0.0895	0.0387	3.6461	0.4083	0.1257	0.9513	0.9909	0.9969	0.0328	0.5248	0.0100	0.0222	2.5207	0.0373

Table 8: Supervised monocular depth results on **normal test set with Overhead Sun**.

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓			AURG↑		
										AbsRel	RMSE	d1	AbsRel	RMSE	d1
Baseline	16.4332	0.1250	0.0525	3.6157	0.5875	0.1747	0.8956	0.9602	0.9783	-	-	-	-	-	-
Deep Ensembles []	16.3795	0.1142	0.0503	3.4465	0.4812	0.1724	0.9027	0.9600	0.9777	0.0739	0.4268	0.0563	-0.0016	2.4750	0.0296
MC Dropout []	16.1976	0.1277	0.0525	3.4437	0.5923	0.1744	0.8934	0.9620	0.9799	0.0720	0.7253	0.0649	0.0104	2.1331	0.0292
Single-PU []	17.1019	0.1319	0.0561	3.4628	0.5126	0.1833	0.8884	0.9580	0.9777	0.0948	0.4474	0.0872	-0.0135	2.4091	0.0103
SLURP []	16.4332	0.1250	0.0525	3.6157	0.5875	0.1747	0.8956	0.9602	0.9783	0.0681	0.7208	0.0852	0.0121	2.2899	0.0054

Table 9: Supervised monocular depth results on **OOD set**.

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓			AURG↑		
										AbsRel	RMSE	d1	AbsRel	RMSE	d1
Baseline	24.2098	2.6367	0.0980	4.7962	10.3942	0.3066	0.7134	0.8775	0.9280	-	-	-	-	-	-
Deep Ensembles []	25.9658	1.8097	0.1009	4.7072	5.1183	0.3237	0.7091	0.8652	0.9174	0.1292	0.6917	0.2091	0.1164	3.1474	0.0067
MC Dropout []	25.3372	3.9252	0.0924	4.3635	22.9193	0.2971	0.7437	0.8829	0.9287	0.2062	0.9267	0.1843	0.0598	2.6365	0.0125
Single-PU []	27.3008	4.3492	0.1009	4.7161	28.5999	0.3284	0.7140	0.8638	0.9174	0.4815	0.7444	0.2104	-0.0210	3.1238	0.0039
SLURP []	24.2098	2.6366	0.0980	4.7962	10.3930	0.3066	0.7134	0.8775	0.9280	0.2116	1.0715	0.2229	0.0682	2.8043	-0.0116

Table 10: Supervised monocular depth results on **low adv. with OOD set**.

Methods	silog↓	AbsRel↓	log10↓	RMSE↓	SqRel↓	log_RMSE	d1↑	d2↑	d3↑	AUSE↓			AURG↑		
										AbsRel	RMSE	d1	AbsRel	RMSE	d1
Baseline	32.1516	0.4588	0.1448	6.9160	10.0794	0.4422	0.5549	0.7727	0.8587	-	-	-	-	-	-
Deep Ensembles []	37.4423	0.3308	0.1672	7.4105	2.7108	0.5183	0.5209	0.7277	0.8179	0.1509	1.0724	0.2720	0.0347	4.8398	0.0285
MC Dropout []	34.0965	0.5448	0.1351	6.1764	14.0074	0.4229	0.6096	0.7933	0.8672	0.3137	1.2454	0.2394	0.0811	3.7196	0.0288
Single-PU []	42.7338	0.3513	0.1735	7.6272	5.0461	0.5606	0.5289	0.7224	0.8106	0.1556	1.3474	0.2768	0.0611	4.7969	0.0232
SLURP []	32.1516	0.4588	0.1448	6.9160	10.0794	0.4422	0.5549	0.7727	0.8587	0.1514	1.5640	0.2737	0.1437	3.9450	0.0134

Table 11: Supervised monocular depth results on **high adv. with OOD set**.

Evaluation sets	AbsRel ↓	log10 ↓	RMSE ↓	SqRel ↓	log_RMSE ↓	d1 ↑	d2 ↑	d3 ↑
Normal	0.365	0.111	5.646	2.234	0.350	0.638	0.874	0.919
Overhead sun	0.174	0.079	5.875	1.426	0.249	0.693	0.953	0.978
low adv. without OOD	0.312	0.185	10.472	3.951	0.586	0.442	0.716	0.824
high adv. without OOD	0.510	0.432	15.578	8.513	1.194	0.227	0.417	0.531
OOD	0.312	0.101	6.170	2.663	0.331	0.648	0.899	0.941
low adv. with OOD	1.462	0.192	9.356	6.054	0.601	0.431	0.697	0.807
high adv. with OOD	1.141	0.415	14.415	25.281	1.194	0.236	0.426	0.543

Table 12: Self-supervised monocular depth results on all test sets given by DIFFNet [18].

References

[1] Akari Asai, Daiki Ikami, and Kiyoharu Aizawa. Multi-task learning based on separable formulation of depth estimation and its uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[2] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

[3] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

[6] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv:2107.03342*, 2021.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[8] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[9] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[10] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

[11] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv:1907.10326*, 2019.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[13] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[14] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, 2020.

[15] Omer Faruk Tuna, Ferhat Ozgur Catak, and M Taner Eskil. Exploiting epistemic uncertainty of the deep learning models to generate adversarial samples. *arXiv preprint arXiv:2102.04150*, 2021.

[16] Xuanlong Yu, Gianni Franchi, and Emanuel Aldea. SLURP: Side learning uncertainty for regression problems. In *BMVC*, 2021.

[17] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. NeW CRFs: Neural window fully-connected CRFs for monocular depth estimation. In *CVPR*, 2022.

[18] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021.