Uncertainty quantification for vision regression tasks

Xuanlong Yu

SATIE, Université Paris-Saclay; U2IS, ENSTA Paris, Institute Polytechnique de Paris

	Composition du Jury	
Madalina Olteanu	Professeure, Université Paris Dauphine - PSL	Rapporteuse
Frédéric Pichon	Professeur, Université d'Artois	Rapporteur
David Picard	Professeur, École des Ponts ParisTech	Examinateur
Pauline Trouvé-Peloux	Chargée de recherche, ONERA	Examinatrice
He Wang	Professeur Associé, University College London	Examinateur
Jean-Emmanuel Deschaud	Chargé de recherche, Mines Paris - PSL	Examinateur
	Direction de la thèse	
Emanuel Aldea	Maître de Conférences, Université Paris-Saclay	Directeur
Gianni Franchi	Enseignant chercheur, ENSTA Paris - IPP	Co-encadrant

December 11, 2023

Big picture - Thesis workflow

Thesis objective:

- More efficient uncertainty quantification solutions without changing main task models that are already trained: **auxiliary uncertainty estimators**;
- Discovery of potential solutions for uncertainty quantification in regression: **classification approaches on discretization-induced regression**.

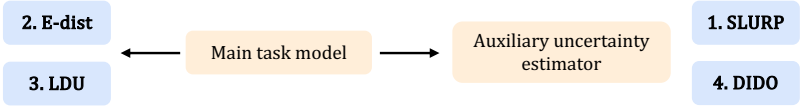
Big picture - Thesis workflow

Thesis objective:

- More efficient uncertainty quantification solutions without changing main task models that are already trained: **auxiliary uncertainty estimators**;
- Discovery of potential solutions for uncertainty quantification in regression: **classification approaches on discretization-induced regression**.

Thesis overview:

Proposed uncertainty quantification solutions	Freezing main task model	Discretization
Simple auxiliary uncertainty estimator (**SLURP**)	✓	✗
Discretization-induced Dirichlet posterior (**DIDO**)	✓	✓
Expectation of distance (**E-dist**)	✗	✓
Latent Discriminant deterministic Uncertainty (**LDU**)	✗	✗



Presentation overview

Current section

Types of uncertainty

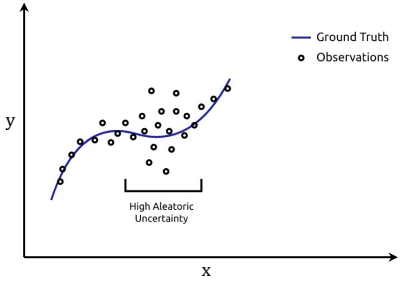
(Abdar et al., 2021; Hüllermeier and Waegeman, 2021)[1]

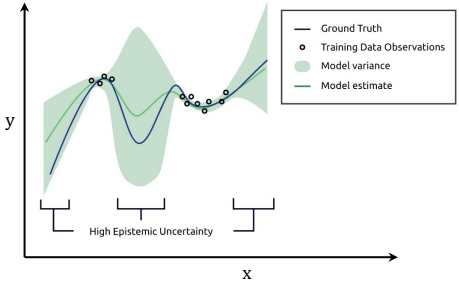
Aleatoric uncertainty: uncertainty inherent in the observation noise (problems caused by sensor quality, natural randomness, etc., that cannot be explained by our data).

Epistemic uncertainty: our ignorance about the correct model that generated the data during inference (lack of knowledge about the data generation process, e.g., out-of-distribution data, which is significantly different from the training distribution).

[1] Moloud Abdar et al. (2021). "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information fusion* 76, pp. 243–297; Eyke Hüllermeier and Willem Waegeman (2021). "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods". In: *Machine Learning* 110, pp. 457–506.

Types of uncertainty[2]





[2] Credits: Huy Nguyen

Bayesian deep neural network (BNN)

Notation

We denote a dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_i^N$, N is the number of image-ground truth pairs, and $\mathcal{D}$ is considered to be independent and identically distributed (i.i.d.). We define a main task network $f_{\boldsymbol{\omega}}$ with parameters $\boldsymbol{\omega}$.

Bayesian deep neural network (BNN)

Notation

We denote a dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_i^N$, N is the number of image-ground truth pairs, and $\mathcal{D}$ is considered to be independent and identically distributed (i.i.d.). We define a main task network $f_{\boldsymbol{\omega}}$ with parameters $\boldsymbol{\omega}$.

BNN (Blundell et al., 2015; Tishby, Levin, and Solla, 1989)[3] is based on marginalization:

$$\mathcal{P}(y|x, \mathcal{D}) = \int \mathcal{P}(y|x, \boldsymbol{\omega})\mathcal{P}(\boldsymbol{\omega}|\mathcal{D})d\boldsymbol{\omega}$$

In practice:

$$\mathcal{P}(y|\mathbf{x}) \simeq \frac{1}{M}\sum_{j=1}^{M}(\mathcal{P}(y|\mathbf{x}, \boldsymbol{\omega}_j)) \text{ with } \boldsymbol{\omega}_j \sim \mathcal{P}(\boldsymbol{\omega}|\mathcal{D})$$

Intractability $\leftarrow$ different techniques to estimate $\mathcal{P}(\boldsymbol{\omega}|\mathcal{D})$.

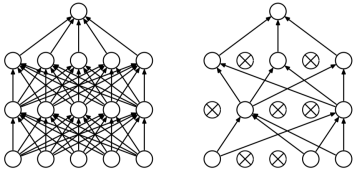
[3] Charles Blundell et al. (2015). "Weight uncertainty in neural network". In: *ICML*; Tishby, Levin, and Solla (1989). "Consistent inference of probabilities in layered networks: predictions and generalizations". In: *IJCNN*. IEEE, pp. 403–409.

Monte-Carlo (MC) Dropout

In MC Dropout (Gal and Ghahramani, 2016)[4], the authors propose to average the predictions of several deep neural networks (DNN) where they apply dropout across the model:

$$\mathcal{P}(y|\mathbf{x}) \simeq \frac{1}{M}\sum_{j=1}^{M}\mathcal{P}(y|\boldsymbol{\omega} \odot b^j, \mathbf{x}) \tag{1}$$

with b^j a vector of the same size of $\boldsymbol{\omega}$ which is a realization of a binomial distribution[5].



(a) Standard Neural Net (b) After applying dropout.

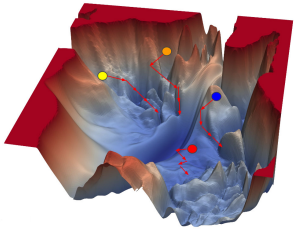
[4] Yarin Gal and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *ICML*.

[5] Image credit: G. Louppe

Deep Ensembles

In Deep Ensembles (Fort, Hu, and Lakshminarayanan, 2019; Lakshminarayanan, Pritzel, and Blundell, 2017)[6], the authors propose to average the predictions of several DNNs with different initial seeds[7]:

$$\mathcal{P}(y|\mathbf{x}) \simeq \frac{1}{M}\sum_{j=1}^{M}\mathcal{P}(y|\mathbf{x},\boldsymbol{\omega}_j) \tag{2}$$



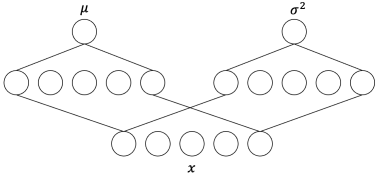
[6] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan (2019). "Deep ensembles: A loss landscape perspective". In: *arXiv preprint arXiv:1912.02757*; Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *NeurIPS*.

[7] Image credit: www.cs.umd.edu/tomg/projects/landscapes/

Learning Gaussian parameters

For regression tasks, the authors of (Kendall and Gal, 2017; Nix and Weigend, 1994)[8] propose to model the outputs of the DNN as the parameters of Gaussian distribution given an input $\mathbf{x}$. The likelihood function is as follows:

$$\mathcal{P}(y|\mathbf{x},\boldsymbol{\omega}) = \frac{1}{\sqrt{2\pi\sigma_{\boldsymbol{\omega}}^2(\mathbf{x})}} \exp \frac{-(y-\mu_{\boldsymbol{\omega}}(\mathbf{x}))^2}{2\sigma_{\boldsymbol{\omega}}^2(\mathbf{x})} \tag{3}$$



[8] Alex 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*.

Evaluation metrics for uncertainty quantification

- AUSE (Area Under Sparsification Error curve) with different accuracy metrics (RMSE, AbsRel, etc.) (Mac Aodha et al., 2012)[9], lower is better ↓

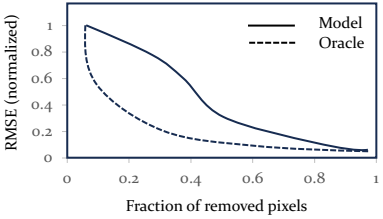


Figure: **Illustrations for uncertainty quantification evaluation - Aleatoric uncertainty estimation evaluation.** Sparsification curves made by model predictions and Oracle. The area between them is AUSE-RMSE.

[9] Oisin Mac Aodha et al. (2012). "Learning a confidence measure for optical flow". In: *IEEE transactions on pattern analysis and machine intelligence* 35.5, pp. 1107–1120.

Evaluation metrics for uncertainty quantification

- Out-of-distribution (OOD)/ In-distribution (ID) example detection (Hendrycks and Gimpel, 2017)[10]: Area Under the ROC Curve (AUROC) and Area Under Precision-Recall Curve (AUPR), higher is better ↑

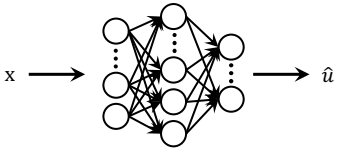


Figure: **Illustrations for uncertainty quantification evaluation - OOD example detection evaluation.** $\hat{u}$ represents the output uncertainty score, u_{in} and u_{out} represent the ground truth uncertainty for ID/OOD examples.

[10] Dan Hendrycks and Kevin Gimpel (2017). "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *ICLR*.

Computer vision task examples

Age estimation

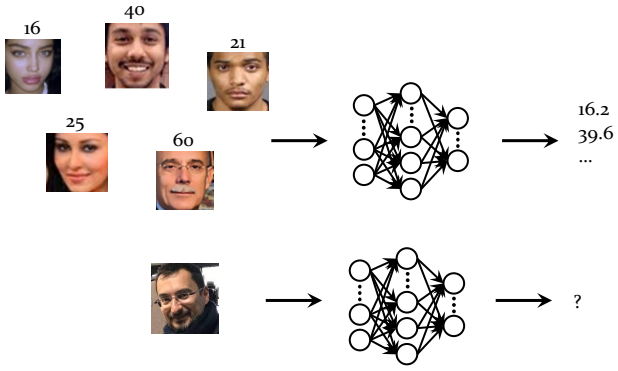


Figure: **An example for age estimation**. Upper: training with labeled facial images; Lower: inferring with a new facial image.

Computer vision task examples

Monocular depth estimation

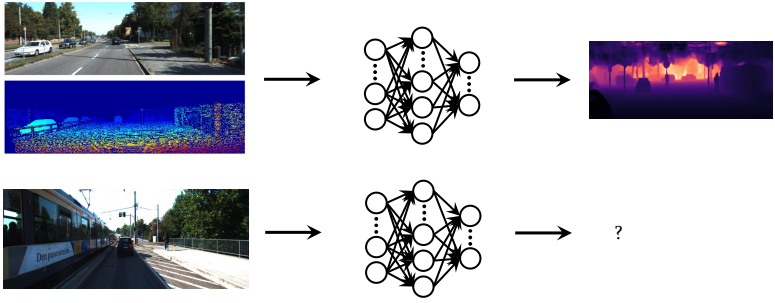


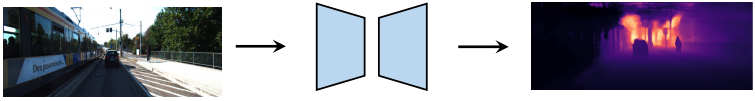
Figure: **An example for monocular depth estimation**[11]. Upper: training with RGB images and the corresponding LIDAR map; Lower: inferring with a new RGB image.

[11] Image credit: KITTI dataset (Geiger et al., 2013)

Current section

General idea for auxiliary uncertainty quantification

Step 1: We obtain a *trained* main task network $f_{\hat{\omega}}$ with parameters $\hat{\omega}$.



Main task model $f_{\hat{\omega}}$

Figure: **Illustration of main task model doing inference.** In the output depth map, the darker pixel represents a closer distance.

General idea for auxiliary uncertainty quantification

Notation

We define a uncertainty estimation network σ_{Θ_1} with parameters Θ_1.

Step 2: We train the auxiliary network σ_{Θ_1} using all the relevant information that we can extract from $\mathcal{D}$ and $f_{\hat{\omega}}$, in order to quantify the uncertainty of the $f_{\hat{\omega}}$ outputs.

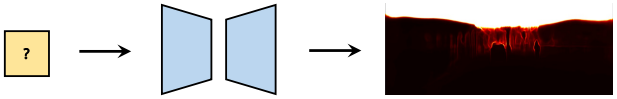


Figure: **Illustration of auxiliary model doing inference.** In the output uncertainty map, the darker pixel represents a smaller uncertainty value.

Problem modeling

We train the auxiliary network σ_{Θ_1} using all the relevant information that we can extract from $\mathcal{D}$ and $f_{\hat{\omega}}$, in order to quantify the uncertainty of the $f_{\hat{\omega}}$ outputs.

- We assume data-dependent noise (Goldberg, Williams, and Bishop, 1997; Nix and Weigend, 1994)[12] follows $\mathcal{N}(0, \boldsymbol{\sigma}^2)$;
- We use the DNN σ_{Θ_1} to estimate the heteroscedastic aleatoric uncertainty u_{alea} (Kendall and Gal, 2017)[13].

[12] Paul Goldberg, Christopher Williams, and Christopher Bishop (1997). "Regression with input-dependent noise: A Gaussian process treatment". In: *NeurIPS*; D.A. Nix and A.S. Weigend (1994). "Estimating the mean and variance of the target probability distribution". In: *ICNN*.

[13] Alex Kendall and Yarin Gal (2017). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *NeurIPS*.

Problem modeling

$\widehat{\Theta}_1$ can be achieved by:

$$\widehat{\Theta}_1 = \arg\max_{\Theta_1} P(\mathcal{D}|\hat{\omega}, \Theta_1) = \arg\max_{\Theta_1} \sum_{i=1}^{N} \log(P(y^{(i)}|\mathbf{x}^{(i)}, \hat{\omega}, \Theta_1)) \quad (4)$$

Distribution assumption on the noise affecting $f_\omega(\mathbf{x})$ can follow different distributions, e.g. Gaussian, Laplacian and Generalized Gaussian distribution (Nadarajah, 2005; Upadhyay et al., 2022)[14], which results in different loss functions.

[14] Saralees Nadarajah (2005). "A generalized normal distribution". In: *Journal of Applied statistics* 32.7, pp. 685–694; Uddeshya Upadhyay et al. (2022). "BayesCap: Bayesian Identity Cap for Calibrated Uncertainty in Frozen Neural Networks". In: *ECCV*.

Problem modeling

Gaussian distribution:

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\log(2\pi\sigma_{\Theta_1}(\mathbf{x}^{(i)})) + \frac{(y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)}))^2}{2\sigma_{\Theta_1}(\mathbf{x}^{(i)})} \tag{5}$$

Laplacian distribution:

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\log(2\sigma_{\Theta_1}(\mathbf{x}^{(i)})) + \frac{|y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)})|}{\sigma_{\Theta_1}(\mathbf{x}^{(i)})} \tag{6}$$

Generalized Gaussian distribution:

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)})|}{\hat{\alpha}^{(i)}}\right)^{\hat{\beta}^{(i)}} - \log\frac{\hat{\beta}^{(i)}}{\hat{\alpha}^{(i)}} + \log\Gamma(\frac{1}{\hat{\beta}^{(i)}}) \tag{7}$$

with $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\hat{\alpha}^{(i)}, \hat{\beta}^{(i)})$, and $\Gamma(\cdot)$ represents the Gamma function.

Problem modeling

Gaussian distribution:

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\log(2\pi\sigma_{\Theta_1}(\mathbf{x}^{(i)})) + \frac{(y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)}))^2}{2\sigma_{\Theta_1}(\mathbf{x}^{(i)})} \tag{8}$$

- Aleatoric uncertainty estimation $\hat{u}_{\text{alea}}^{(i)}$, which is output by $\sigma_{\Theta_1}(\mathbf{x}^{(i)})$;
- Error set is denoted as $\boldsymbol{\epsilon} = \{\epsilon^{(i)}\}_{i=1}^{N} = \{(y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)}))^2\}_{i=1}^{N}$;
- The optimal output value of $\sigma_{\Theta_1}(\mathbf{x}^{(i)})$ is $\epsilon^{(i)}$.

Design of Side Learning Uncertainty for Regression Problems

Observations and motivation

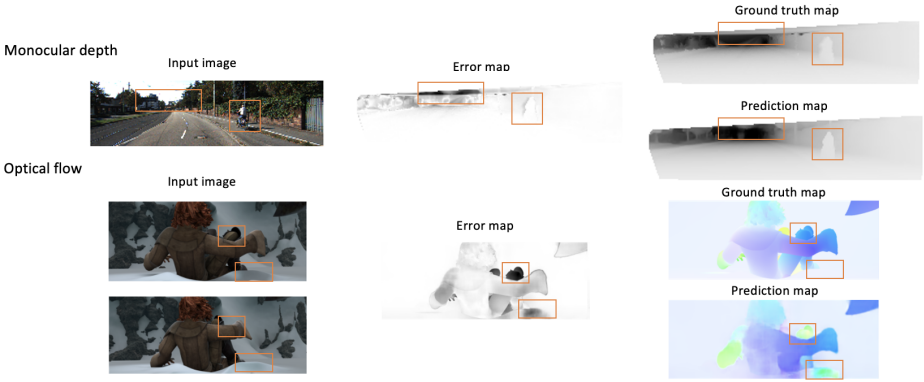


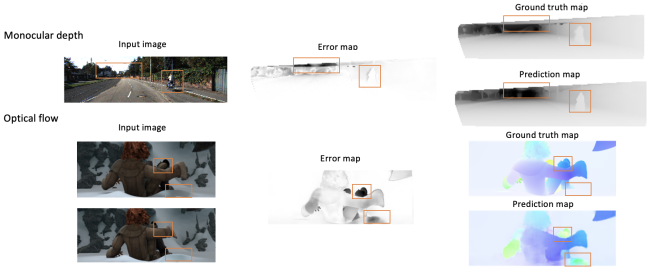
Figure: **Illustration for prediction errors in pixel-wise regression tasks**. Orange boxes highlighted the occluded areas and edges between the objects.

Design of Side Learning Uncertainty for Regression Problems

Observations and motivation

From the visualization of prediction errors:

1. Edges of connected domains in the prediction map ← **edge detection (Y. Liu et al., 2017)[15] on the prediction map**;
2. Hard predictable areas ← **two sources of inputs (image and prediction map)**.



[15] Yun Liu et al. (2017). "Richer convolutional features for edge detection". In: *CVPR*.

Design of Side Learning Uncertainty for Regression Problems

Architecture overview

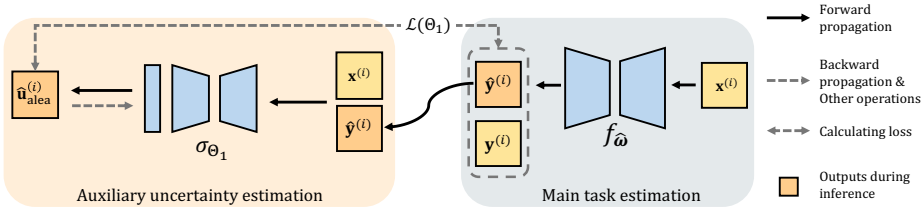


Figure: **Overall processing of SLURP.** Both forward and backward propagations are illustrated.

Design of Side Learning Uncertainty for Regression Problems

Architecture overview

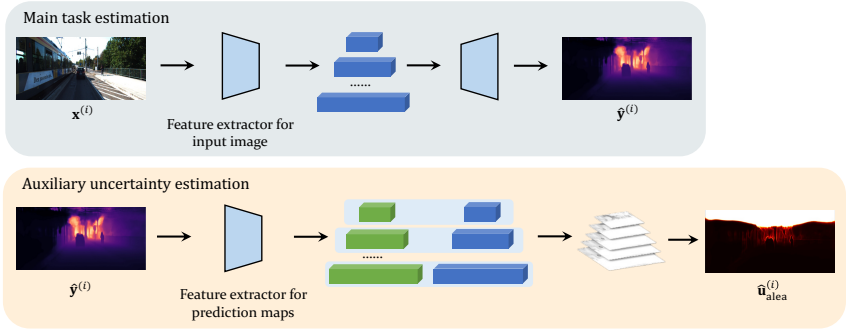


Figure: **Detailed auxiliary network architecture proposed in SLURP.** Green and blue blocks are coarse-to-fine features (Y. Liu et al., 2017)[16] from image and prediction map encoders respectively.

[16] Yun Liu et al. (2017). "Richer convolutional features for edge detection". In: *CVPR*.

Datasets	Criteria	MC	EEns.	Single PU	DEns.	Confid	Ours
KITTI	AUSE-RMSE	8.14	3.17	1.89	**1.68**	1.76	**1.68**
	AUSE-Absrel	9.48	5.02	4.59	4.32	**4.24**	4.36
	AUROC	0.686	0.882	0.882	**0.897**	0.892	0.895
CityScapes	AUSE-RMSE	**9.42**	11.56	9.91	11.47	10.48	9.48
	AUSE-Absrel	9.52	13.14	9.96	9.36	**5.75**	10.90
	AUROC	0.420	0.504	0.386	0.501	**0.519**	0.400
After fine-tuning on CityScapes							
CityScapes	AUSE-RMSE	7.72	8.20	4.35	**3.03**	4.05	3.05
	AUSE-Absrel	8.13	7.50	6.44	6.81	**6.34**	6.55
	AUROC	0.705	0.786	0.741	**0.856**	0.821	0.849
CityScapes Rainy s=1	AUSE-RMSE	7.06	7.29	4.17	3.42	4.89	**3.39**
	AUSE-Absrel	8.73	6.92	6.55	6.68	7.26	**5.62**
	AUROC	0.659	0.757	0.731	0.746	0.697	**0.788**
CityScapes Rainy s=2	AUSE-RMSE	7.14	6.9	4.27	**3.35**	4.68	3.36
	AUSE-Absrel	8.36	6.48	6.79	6.24	6.86	**5.28**
	AUROC	0.667	0.767	0.731	0.756	0.714	**0.794**
CityScapes Rainy s=3	AUSE-RMSE	7.30	6.66	4.35	**3.28**	4.59	3.41
	AUSE-Absrel	8.27	6.03	6.44	5.85	6.64	**5.05**
	AUROC	0.665	0.778	0.742	0.767	0.729	**0.801**
CityScapes Foggy s=1	AUSE-RMSE	7.80	7.82	3.42	3.05	3.98	**3.04**
	AUSE-Absrel	8.36	7.33	6.78	6.58	**6.21**	6.25
	AUROC	0.700	0.783	0.842	**0.852**	0.824	0.847
CityScapes Foggy s=2	AUSE-RMSE	7.82	7.53	3.42	**2.98**	3.86	3.01
	AUSE-Absrel	8.20	7.09	6.55	6.35	**6.02**	6.06
	AUROC	0.704	0.791	0.847	**0.857**	0.833	0.852
CityScapes Foggy s=3	AUSE-RMSE	7.84	7.28	3.48	**2.93**	3.70	3.08
	AUSE-Absrel	7.87	6.80	6.19	6.01	**5.78**	5.80
	AUROC	0.715	0.801	0.851	**0.863**	0.846	0.857

Table: **Uncertainty quantification performance on monocular depth estimation.** s (*e.g* $s = 1$) indicates severity, the higher the s value, the higher the severity.

Task	Criteria	MC	EEns.	Single PU	DEns.	Confid	Ours
Monocular depth	Runtime (ms)	386	144	98	286	106	**88**
	# Param. (M)	**47.0**	141.0	94.0	282.0	94.7	87.2

Table: **Average time cost for processing one image and number of parameters of the model(s)** (main task + uncertainty task).

Current section

Observations and motivation

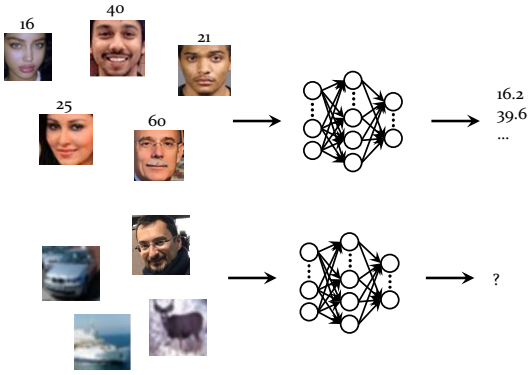


Figure: **An example for age estimation with OOD examples.** Upper: training with labeled facial images; Lower: inferring with unknown images[17].

[17] Image credit: CIFAR10 dataset (Krizhevsky, Hinton, et al., 2009)

Observations and motivation

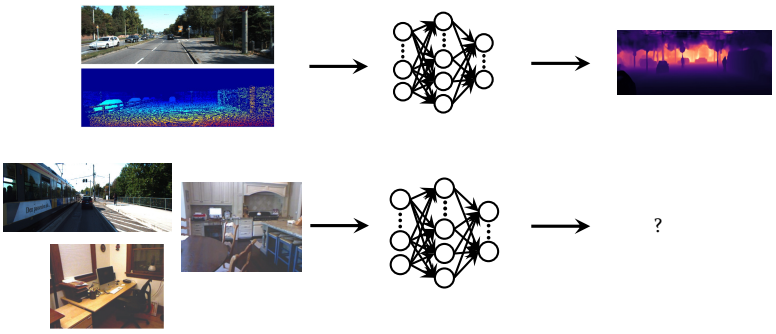


Figure: **An example for depth estimation with OOD examples.** Upper: training with labeled RGB images and the corresponding LIDAR map; Lower: inferring with unknown images[18].

[18] Image credit: NYU v2 dataset (Silberman et al., 2012)

Observations and motivation

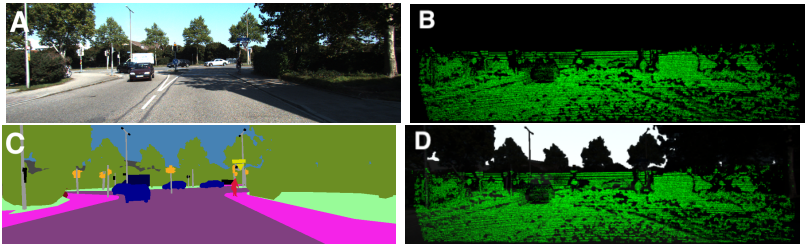


Figure: **Visualization on the OOD example detection in monocular depth estimation.** *A:* input image, *B:* ground truth depth map, green point represent pixels with depth ground truth; *C:* semantic ground truth map, blue point represent pixels with sky pattern; *D:* ID-OOD map, green points represent the depth ground truth, i.e., the ID part. White parts represent the sky pattern, which is the OOD part.

Observations and motivation

From the Out-of-distribution (OOD) scenarios:

1. Image-wise regression tasks: **OOD inputs from OOD datasets** should be assigned to higher uncertainty to make an alarm, as in classification tasks;
2. Pixel-wise regression tasks:
   2.1. Predictions based on **OOD inputs from OOD datasets** should be assigned to higher uncertainty;
   2.2. Predictions based on **OOD patterns from ID datasets** also should be highlighted to make the final prediction more reliable.

Potential solutions

Evidential regression in uncertainty quantification

Evidential regression (Amini et al., 2020)[19] is considered an efficient uncertainty estimation approach, which can capture epistemic uncertainty with a single pass. It explicitly parameterizes and updates the conjugate prior of the assumed distribution for the main task training target.

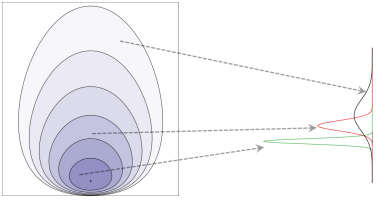


Figure: **Illustration of the relation between Normal-inverse-gamma (NIG) distribution and Gaussian distribution.** Left: visualization of a NIG distribution[20], Right: illustration of sampling Gaussian distributions from the NIG distribution.

[19] Alexander Amini et al. (2020). "Deep evidential regression". In: *NeurIPS*.

[20] Image credit: Wikipedia

Potential solutions

Evidential regression in uncertainty quantification

When we implement it on the auxiliary network, the loss function will be $\mathcal{L}(\Theta_1) = \mathcal{L}_1(\Theta_1) + \lambda_{\text{NIG}} \cdot \mathcal{L}_2(\Theta_1)$ with λ_{NIG} the weight for the regularization term, and:

$$\mathcal{L}_1(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\log(\frac{\pi}{\nu^{(i)}}) - \alpha^{(i)}\log(\Omega^{(i)}) + (\alpha^{(i)} + \frac{1}{2})\log((y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)}))^2\nu^{(i)} + \Omega^{(i)}) + \log(\frac{\Gamma(\alpha^{(i)})}{\Gamma(\alpha^{(i)} + \frac{1}{2})}) \tag{9}$$

$$\mathcal{L}_2(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N} |y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)})| \cdot (2\nu^{(i)} + \alpha^{(i)}) \tag{10}$$

with $\Gamma(\cdot)$ the Gamma function, $\Omega^{(i)} = 2\beta^{(i)}(1 + \nu^{(i)})$, and $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\hat{\alpha}^{(i)}, \hat{\beta}^{(i)}, \hat{\nu}^{(i)})$.

Potential solutions

Evidential regression in uncertainty quantification

When we implement it on the auxiliary network, the loss function will be $\mathcal{L}(\Theta_1) = \mathcal{L}_1(\Theta_1) + \lambda_{\text{NIG}} \cdot \mathcal{L}_2(\Theta_1)$ with λ_{NIG} the weight for the regularization term, and:

$$\mathcal{L}_1(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\log(\frac{\pi}{\nu^{(i)}}) - \alpha^{(i)}\log(\Omega^{(i)}) + (\alpha^{(i)} + \frac{1}{2})\log((y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)}))^2\nu^{(i)} + \Omega^{(i)}) + \log(\frac{\Gamma(\alpha^{(i)})}{\Gamma(\alpha^{(i)} + \frac{1}{2})}) \quad (9)$$

$$\mathcal{L}_2(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N} |y^{(i)} - f_{\hat{\omega}}(\mathbf{x}^{(i)})| \cdot (2\nu^{(i)} + \alpha^{(i)}) \quad (10)$$

with $\Gamma(\cdot)$ the Gamma function, $\Omega^{(i)} = 2\beta^{(i)}(1 + \nu^{(i)})$, and $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\hat{\alpha}^{(i)}, \hat{\beta}^{(i)}, \hat{\nu}^{(i)})$.

- Evidence is an interpretation of the parameters of the conjugate prior distribution (Michael I., 2010)[21], which can describe the uncertainty of the prediction.
- Long-tailed distributed prediction error in Eq. 10 will make the auxiliary network more inclined to give high evidence for most data points.

[21] Jordan Michael I. (2010). "The exponential family: Conjugate priors". In: *Bayesian modeling and inference course*.

Discretization-Induced Dirichlet pOsterior (DIDO)

Overview

Roadmap for introducing DIDO:

1. To reduce the long-tail distribution impact of prediction errors: discretization on prediction error ϵ;
2. Discussion on epistemic uncertainty modeling in auxiliary uncertainty quantification scenario: exploration on using ϵ for epistemic uncertainty estimation;
3. To integrate the above-mentioned components: Dirichlet posterior modeling.

Discretization-Induced Dirichlet pOsterior (DIDO)

Discretization on prediction errors ϵ

To discretize prediction errors ϵ: we compute the different quantiles on ϵ

Discretization-Induced Dirichlet pOsterior (DIDO)

Discretization on prediction errors ϵ

To discretize prediction errors $\boldsymbol{\epsilon}$: we compute the different quantiles on $\boldsymbol{\epsilon}$

1. Sorting the errors in ascending order and creating a new set, denoted by $\boldsymbol{\epsilon}'$, with the same elements as $\boldsymbol{\epsilon}$.

Discretization-Induced Dirichlet pOsterior (DIDO)

Discretization on prediction errors ϵ

To discretize prediction errors $\boldsymbol{\epsilon}$: we compute the different quantiles on $\boldsymbol{\epsilon}$

1. Sorting the errors in ascending order and creating a new set, denoted by $\boldsymbol{\epsilon}'$, with the same elements as $\boldsymbol{\epsilon}$.

2. Dividing $\boldsymbol{\epsilon}'$ into K subsets of equal size, represented by $\{\boldsymbol{\epsilon}_k\}_{k=1}^{K}$. Each error value $\epsilon^{(i)}$ is then replaced by the index of its corresponding subset $k \in [1, K]$, and transformed into a one-hot vector, denoted by $\bar{\boldsymbol{\epsilon}}^{(i)}$, as the final training target:

$$\bar{\boldsymbol{\epsilon}}^{(i)} = [\bar{\epsilon}_1^{(i)} \dots \bar{\epsilon}_k^{(i)} \dots \bar{\epsilon}_K^{(i)}]^{\mathsf{T}} \in \mathbb{R}^K \tag{11}$$

where $\bar{\epsilon}_k^{(i)} = 1$ if $\epsilon^{(i)}$ belongs to the kth subset, and 0 otherwise. Each subset or bin represents a class of error severity. We can achieve a new dataset $\bar{\mathcal{D}} = \{\mathbf{x}^{(i)}, \bar{\boldsymbol{\epsilon}}^{(i)}\}_i^N$.

Discretization-Induced Dirichlet pOsterior (DIDO)

Exploration on using ϵ for epistemic uncertainty estimation

In a Bayesian framework, given an input $\mathbf{x}$, the predictive uncertainty of a DNN is modeled by $P(y|\mathbf{x}, \mathcal{D})$. We can have the following assumptions and simplifications:

[22] Andrey Malinin and Mark Gales (2018). "Predictive uncertainty estimation via prior networks". In: *NeurIPS*.

Discretization-Induced Dirichlet pOsterior (DIDO)

Exploration on using ϵ for epistemic uncertainty estimation

In a Bayesian framework, given an input $\mathbf{x}$, the predictive uncertainty of a DNN is modeled by $P(y|\mathbf{x}, \mathcal{D})$. We can have the following assumptions and simplifications:

1. Since we have a trained main task DNN, and as proposed in (Malinin and Gales, 2018)[22], we assume a point-estimate of $\boldsymbol{\omega}$, and in auxiliary uncertainty estimation case, we have the trained and fixed main task model with parameters $\hat{\boldsymbol{\omega}}$, then we have:

$$P(\boldsymbol{\omega}|\mathcal{D}) = \delta(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}) \rightarrow P(y|\mathbf{x}, \mathcal{D}) \simeq P(y|\mathbf{x}, \hat{\boldsymbol{\omega}}) \tag{12}$$

with δ being the Dirac function.

[22] Andrey Malinin and Mark Gales (2018). "Predictive uncertainty estimation via prior networks". In: *NeurIPS*.

Discretization-Induced Dirichlet pOsterior (DIDO)

Exploration on using ϵ for epistemic uncertainty estimation

2. We follow the previous assumption, i.e., the prediction is drawn from $\mathcal{N}(y|\mu, \sigma^2)$ and according to the previous work (Amini et al., 2020)[23], we denote $\boldsymbol{\alpha}$ as the parameters of prior distributions of (μ, σ^2). Following the same work, we first have:

$$P(\mu, \sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) = P(\mu|\sigma^2, \mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \tag{13}$$

Then, according to Eq. 12, we regard the μ depends only on $\mathbf{x}$ and the main task model $\hat{\boldsymbol{\omega}}$:

$$\begin{aligned} P(\mu, \sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) &= P(\mu|\mathbf{x}, \hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \\ &= \delta(\mu - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}))P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \end{aligned} \tag{14}$$

[23] Alexander Amini et al. (2020). "Deep evidential regression". In: *NeurIPS*.

Discretization-Induced Dirichlet pOsterior (DIDO)

Exploration on using ϵ for epistemic uncertainty estimation

We introduce $\boldsymbol{\alpha}$ and re-write $P(y|\mathbf{x}, \hat{\boldsymbol{\omega}})$ in Eq. 12 as:

$$
\begin{aligned}
P(y|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) &= \iint P(y|\mu, \sigma^2) P(\mu, \sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) d\mu d\sigma^2 \\
&\overset{(a)}{=} \iint P(y|\mu, \sigma^2) P(\mu|\sigma^2, \mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) d\mu d\sigma^2 \\
&= \iint P(y, \mu|\sigma^2, \mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) d\mu d\sigma^2 \\
&\overset{(b)}{=} \int \delta(\mu - f_{\hat{\omega}}(\mathbf{x})) d\mu \int P(y|\sigma^2, \mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) d\sigma^2 \\
&= \int P(y|\mathbf{x}, \sigma^2) P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) d\sigma^2 \qquad (15)
\end{aligned}
$$

where the equality (a) and (b) in Eq. 15 are given by Eq. 13 and Eq. 14, respectively.

Discretization-Induced Dirichlet pOsterior (DIDO)

Exploration on using ϵ for epistemic uncertainty estimation

In summary, we have

$$\begin{aligned} P(y|\mathbf{x},\mathcal{D}) &= \iint P(y|\mathbf{x},\sigma^2)P(\sigma^2|\boldsymbol{\omega})P(\boldsymbol{\omega}|\mathcal{D})d\sigma^2 d\boldsymbol{\omega} \\ &= \int P(y|\mathbf{x},\sigma^2)P(\sigma^2|\mathcal{D})d\sigma^2 \\ &\overset{(a)}{\simeq} \int P(y|\mathbf{x},\sigma^2)P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\sigma^2 \end{aligned} \tag{16}$$

where the approximation (a) in Eq. 16 is given by Eq. 15.

Discretization-Induced Dirichlet pOsterior (DIDO)

Exploration on using ϵ for epistemic uncertainty estimation

In summary, we have

$$
\begin{aligned}
P(y|\mathbf{x}, \mathcal{D}) &= \iint P(y|\mathbf{x}, \sigma^2) P(\sigma^2|\boldsymbol{\omega}) P(\boldsymbol{\omega}|\mathcal{D}) d\sigma^2 d\boldsymbol{\omega} \\
&= \int P(y|\mathbf{x}, \sigma^2) P(\sigma^2|\mathcal{D}) d\sigma^2 \\
&\overset{(a)}{\simeq} \int P(y|\mathbf{x}, \sigma^2) P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) d\sigma^2
\end{aligned} \tag{16}
$$

where the approximation (a) in Eq. 16 is given by Eq. 15.

- We can consider ϵ to be drawn from a distribution parameterized by σ^2;
- $P(\sigma^2|\mathcal{D})$ can describe epistemic uncertainty and the variational approach can be applied (Joo, Chung, and Seo, 2020; Malinin and Gales, 2018)[24]:
$P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \simeq P(\sigma^2|\mathcal{D})$.

[24] Taejong Joo, Uijung Chung, and Min-Gwan Seo (2020). "Being bayesian about categorical probability". In: *ICML*; Andrey Malinin and Mark Gales (2018). "Predictive uncertainty estimation via prior networks". In: *NeurIPS*.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

After discretization:
$P(\boldsymbol{\pi}|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \simeq P(\boldsymbol{\pi}|\bar{\mathcal{D}})$, with $\bar{\mathcal{D}} = \{\mathbf{x}^{(i)}, \bar{\boldsymbol{\epsilon}}^{(i)}\}_i^N$ defined as before.

We adjust $P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}})$ in Eq. 16 to $P(\boldsymbol{\pi}|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}})$, with $\boldsymbol{\alpha}$ the parameters of a conjugate prior of a discrete distribution parameterized by $\boldsymbol{\pi}$.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

We consider $\bar{\epsilon}^{(i)}$ to be drawn from a categorical distribution, and $\boldsymbol{\pi}^{(i)} = (\pi_1^{(i)}, \ldots, \pi_K^{(i)})$ denotes the random variable over this distribution, where $\sum_{k=1}^{K} \pi_k^{(i)} = 1$ and $\pi_k^{(i)} \in [0, 1]$ for $k \in \{1, ..., K\}$.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

We consider $\bar{\epsilon}^{(i)}$ to be drawn from a categorical distribution, and $\boldsymbol{\pi}^{(i)} = (\pi_1^{(i)}, \ldots, \pi_K^{(i)})$ denotes the random variable over this distribution, where $\sum_{k=1}^{K} \pi_k^{(i)} = 1$ and $\pi_k^{(i)} \in [0, 1]$ for $k \in \{1, ..., K\}$.

We use the conjugate prior of categorical distribution, which is a Dirichlet distribution, as the variational distribution:

$$P(\boldsymbol{\pi}^{(i)}|\boldsymbol{\alpha}^{(i)}) = \frac{\Gamma(S^{(i)})}{\prod_{k=1}^{K} \Gamma(\alpha_k^{(i)})} \prod_{k=1}^{K} {\pi_k^{(i)}}^{\alpha_k^{(i)}-1} \tag{17}$$

with $\Gamma(\cdot)$ the Gamma function, $\boldsymbol{\alpha}^{(i)}$ positive concentration parameters of Dirichlet distribution and $S^{(i)} = \sum_{k=1}^{K} \alpha_k^{(i)}$ the Dirichlet strength.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

Notation

We define another uncertainty estimation network σ_{Θ_2} with parameters Θ_2.

The auxiliary network σ_{Θ_2} will output and update $\boldsymbol{\alpha}$:

- $\boldsymbol{\alpha}$ will be initialized as $\mathbf{1}$;
- $\boldsymbol{\alpha}^{(i)} = \sigma_{\Theta_2}(\mathbf{x}^{(i)}) + \mathbf{1}$, where an exponential function on the top of σ_{Θ_2} will remain the output positive.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

We minimize the Kullback-Leibler (KL) divergence between the variational distribution $P(\boldsymbol{\pi}|\Theta_2)$ and the true posterior $P(\boldsymbol{\pi}|\bar{\mathcal{D}})$ to achieve $\widehat{\Theta}_2$ (Charpentier, Zügner, and Günnemann, 2020; Joo, Chung, and Seo, 2020)[25]:

$$\begin{aligned}\widehat{\Theta}_2 &= \arg\min_{\Theta_2} \ \mathrm{KL}[P(\boldsymbol{\pi}|\Theta_2)||P(\boldsymbol{\pi}|\bar{\mathcal{D}})] \\ &= \arg\min_{\Theta_2} \int P(\boldsymbol{\pi}|\Theta_2) \log \frac{P(\boldsymbol{\pi}|\Theta_2)}{P(\boldsymbol{\pi})P(\bar{\mathcal{D}}|\boldsymbol{\pi})} \\ &= \arg\min_{\Theta_2} -\mathbb{E}_{P(\boldsymbol{\pi}|\Theta_2)}[\log P(\bar{\mathcal{D}}|\boldsymbol{\pi})] + \mathrm{KL}[P(\boldsymbol{\pi}|\Theta_2)||P(\boldsymbol{\pi})] \end{aligned} \tag{18}$$

[25] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann (2020). "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts". In: *NeurIPS*; Taejong Joo, Uijung Chung, and Min-Gwan Seo (2020). "Being bayesian about categorical probability". In: *ICML*.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

The loss function will be equivalent to minimizing the negative evidence lower bound (Jordan et al., 1999)[26], considering the prior distribution $P(\boldsymbol{\pi})$ as $\text{Dir}(\mathbf{1})$:

$$\mathcal{L}(\Theta_2) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left[\bar{\epsilon}_k^{(i)}\left(\psi(S^{(i)}) - \psi(\alpha_k^{(i)})\right)\right] + \lambda \text{KL}\left(\text{Dir}(\boldsymbol{\alpha}^{(i)})||\text{Dir}(\mathbf{1})\right) \tag{19}$$

where ψ is the digamma function, λ is a positive hyperparameter for the regularization term.

[26] Michael I Jordan et al. (1999). "An introduction to variational methods for graphical models". In: *Machine learning* 37, pp. 183–233.

Discretization-Induced Dirichlet pOsterior (DIDO)

Dirichlet posterior for epistemic uncertainty

For measuring epistemic uncertainty, we consider using the pseudo-counts of the evidence from the Dirichlet distribution (Charpentier, Zügner, and Günnemann, 2020; Shen et al., 2023)[27]. The epistemic uncertainty is inversely proportional to the Dirichlet strength:

$$\hat{u}_{\text{epis}}^{(i)} = \sigma_{\widehat{\Theta}_2}(\mathbf{x}^{(i)}) = \frac{K}{S^{(i)}} \tag{20}$$

[27] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann (2020). "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts". In: *NeurIPS*; Maohao Shen et al. (2023). "Post-hoc Uncertainty Learning using a Dirichlet Meta-Model". In: *AAAI*.

Generalized auxiliary networks with DIDO

Integration

We propose:

- A generalized auxiliary uncertainty estimator with two components, namely σ_{Θ_1} and σ_{Θ_2}, to quantify the uncertainty of the prediction given by the main task model;
- A loss function for training the generalized auxiliary network:

$$\mathcal{L}_{\text{AuxUE}} = \mathcal{L}(\Theta_1) + \mathcal{L}(\Theta_2) \tag{21}$$

- Estimation for two types of uncertainty via:

$$\hat{u}^{(i)}_{\text{alea}} = \sigma_{\widehat{\Theta}_1}(\mathbf{x}^{(i)}) \quad \text{and} \quad \hat{u}^{(i)}_{\text{epis}} = \sigma_{\widehat{\Theta}_2}(\mathbf{x}^{(i)}) = \frac{K}{S^{(i)}} \tag{22}$$

Generalized auxiliary networks with DIDO

Architecture overview

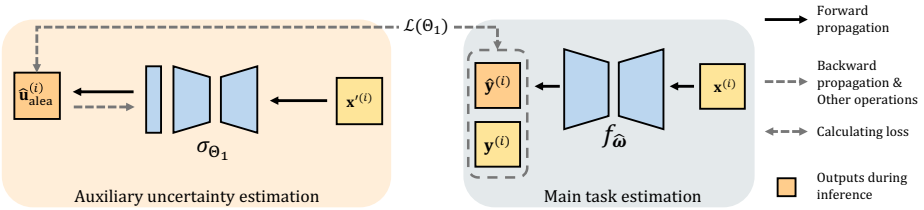


Figure: **Overall processing of the simple auxiliary network.** The input of σ_{Θ_1} can be the combinations of the input, output (Upadhyay et al., 2022; Yu, Franchi, and Aldea, 2021)[28], or intermediate features (Corbière et al., 2019)[29] of $f_{\hat{\boldsymbol{\omega}}}$, we here simplify it to $\mathbf{x}'^{(i)}$ for brevity.

[28] Uddeshya Upadhyay et al. (2022). "BayesCap: Bayesian Identity Cap for Calibrated Uncertainty in Frozen Neural Networks". In: *ECCV*; Xuanlong Yu, Gianni Franchi, and Emanuel Aldea (2021). "SLURP: Side Learning Uncertainty for Regression Problems". In: *BMVC*.

[29] Charles Corbière et al. (2019). "Addressing failure prediction by learning model confidence". In: *NeurIPS*.

Generalized auxiliary networks with DIDO

Architecture overview

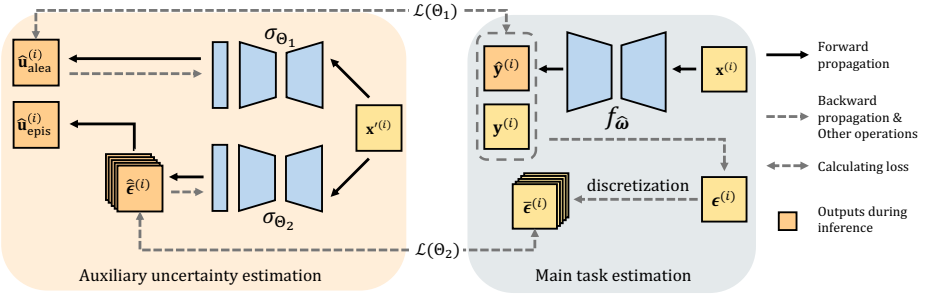


Figure: **Overall processing of generalized auxiliary network with *separated* encoders.** The input of σ_{Θ_1} and σ_{Θ_2} can be the combinations of the input, output, or intermediate features of $f_{\hat{\boldsymbol{\omega}}}$, we here simplify it to $\mathbf{x}'^{(i)}$ for brevity.

Generalized auxiliary networks with DIDO

Architecture overview

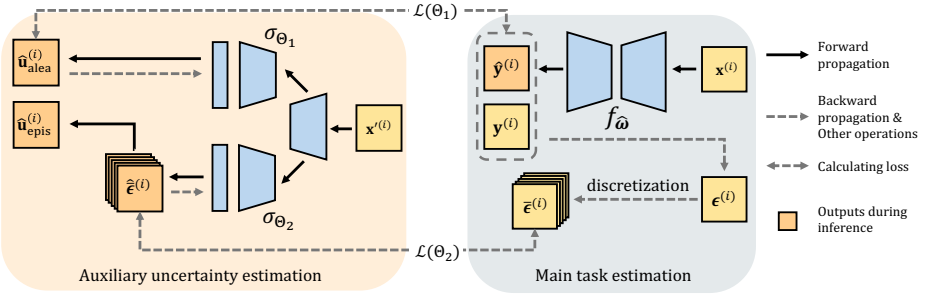


Figure: **Overall processing of generalized auxiliary network with a *shared* encoder.** The input of σ_{Θ_1} and σ_{Θ_2} can be the combinations of the input, output, or intermediate features of $f_{\hat{\boldsymbol{\omega}}}$, we here simplify it to $\mathbf{x}'^{(i)}$ for brevity.

Generalized auxiliary networks with DIDO

Experiment - Toy examples

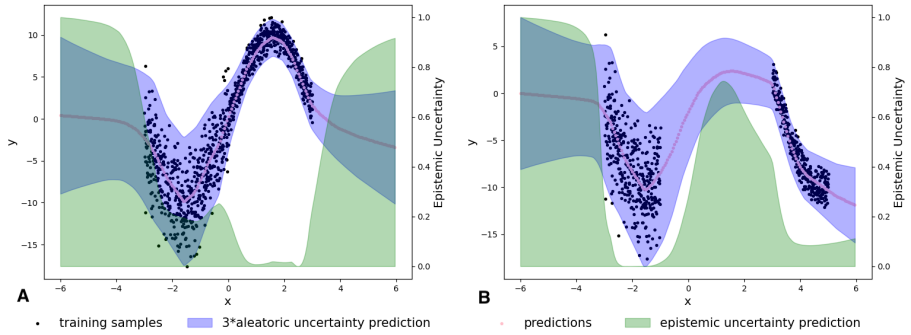


Figure: **Results on 1D toy examples.** Aleatoric and epistemic uncertainty estimations given by our proposed AuxUE are presented respectively as the prediction interval and the uncertainty degree (0-1).

Experiment - Age estimation and OOD detection

		AuxUE		Modified main DNN			Training-free	
OOD Dataset	Metrics	Ours σ_{Θ_1}	Ours σ_{Θ_2} DIDO	LDU	Evi.	DEns.	Grad. (inv.)	Inject.
CIFAR10	AUROC ↑	96.0	**100**	95.2	50.0	99.2	**100**	94.5
	AUPR ↑	91.7	**100**	88.3	23.4	95.1	**100**	87.3
SVHN	AUROC ↑	98.3	**100**	94.8	50.0	99.2	**100**	94.0
	AUPR ↑	98.1	**100**	93.2	44.3	97.8	**100**	92.5
MNIST	AUROC ↑	97.8	**100**	97.6	50.0	99.6	**100**	98.8
	AUPR ↑	93.9	**100**	93.8	23.4	97.2	**100**	96.9
Fashion MNIST	AUROC ↑	97.7	**100**	95.6	50.0	99.1	**100**	97.7
	AUPR ↑	94.0	**100**	89.3	23.4	93.8	**100**	94.2
Oxford Pets	AUROC ↑	**82.9**	55.9	31.5	50.1	56.1	50.7	48.6
	AUPR ↑	**53.3**	23.9	12.5	18.5	21.3	19.6	20.3
Fake Data	AUROC ↑	67.0	**80.8**	70.0	50.0	33.2	45.9	45.1
	AUPR ↑	59.7	**70.2**	58.8	49.5	37.8	46.3	44.6

Table: **OOD detection results on Age estimation task.** ID data is from Asian Face Age Dataset (AFAD) (Niu et al., 2016).

Experiment - Robustness under dataset change

	AuxUE with DIDO		Modified main DNN			Training-free	
Metrics	Ours σ_{Θ_2} sep. enc.	Ours σ_{Θ_2} shar. enc.	LDU	Evi.	DEns.	Grad.	Inject.
AUROC ↑	98.1	**98.4**	58.1	70.6	62.1	78.4	18.3
AUPR ↑	99.3	**99.4**	79.5	77.8	76.7	92.6	62.3

Table: **Epistemic uncertainty estimation results encountering dataset change on Monocular depth estimation task.** The evaluation dataset used here is NYU indoor depth dataset.

Experiment - Robustness on unseen patterns during training

		AuxUE with DIDO		Modified main DNN			Training-free	
S	Metrics	Ours σ_{Θ_2} sep. enc.	Ours σ_{Θ_2} shar. enc.	LDU	Evi.	DEns.	Grad. (inv.)	Inject.
0	AUROC ↑	**100.0**	99.9	96.5	76.7	93.5	85.6	58.4
	AUPR ↑	**100.0**	99.0	93.8	42.6	70.0	76.3	28.1
	Sky-All ↓	0.015	0.018	0.278	0.986	0.005	**0.001**	0.800
1	AUROC ↑	**100.0**	99.9	96.3	69.7	92.8	76.9	58.5
	AUPR ↑	**99.9**	98.9	93.5	37.4	68.0	69.8	28.2
	Sky-All ↓	0.016	0.018	0.277	0.988	0.005	**0.002**	0.799
2	AUROC ↑	**99.9**	**99.9**	95.9	65.4	92.3	75.6	58.4
	AUPR ↑	**99.8**	98.8	93.0	34.5	67.0	67.8	28.1
	Sky-All ↓	0.017	0.018	0.280	0.990	0.005	**0.002**	0.803
3	AUROC ↑	**99.9**	99.7	95.9	62.3	91.6	73.6	58.4
	AUPR ↑	**99.7**	98.1	92.8	32.8	65.7	64.5	28.2
	Sky-All ↓	0.018	0.020	0.283	0.992	0.005	**0.002**	0.809
4	AUROC ↑	**99.6**	99.5	96.1	58.8	91.8	71.3	58.4
	AUPR ↑	**99.1**	97.2	92.9	31.2	67.2	60.0	28.3
	Sky-All ↓	0.023	0.022	0.288	0.994	0.005	**0.002**	0.819
5	AUROC ↑	98.5	**99.0**	96.5	58.5	92.2	66.8	57.8
	AUPR ↑	**97.1**	96.1	93.7	32.8	70.4	53.8	28.2
	Sky-All ↓	0.035	0.026	0.295	0.996	0.005	**0.002**	0.839

Table: **Epistemic uncertainty estimation results encountering unseen pattern on Monocular depth estimation task.** The evaluation datasets used here are KITTI Seg-Depth (S=0) and KITTI Seg-Depth-C (S>0).

Generalized auxiliary networks with DIDO

Qualitative results

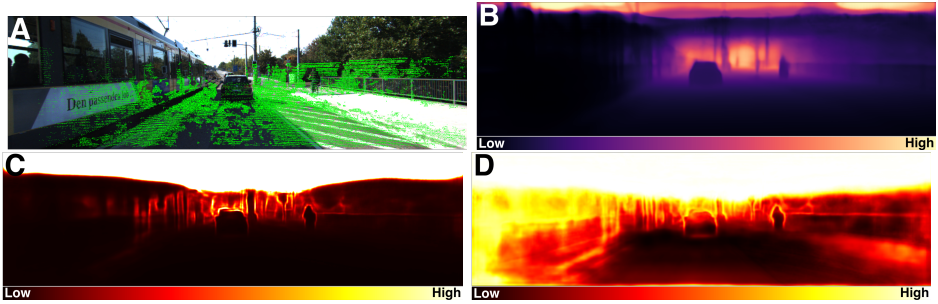


Figure: **Illustrations of uncertainty estimations for MDE task.** *A:* input image, green points represent pixels with depth ground truth; *B:* depth prediction; *C and D:* aleatoric and epistemic uncertainty estimations. The areas lacking depth ground truth, e.g. sky and tramway, are assigned high uncertainty using DIDO.

Current section

Conclusion

In this thesis, we proposed:

1. Auxiliary networks for uncertainty quantification: no adjustment on the main task model - **SLURP**, **DIDO**
2. Discretization on continuous targets for uncertainty quantification: from a novel point of view to estimate uncertainty for regression tasks - **E-dist**, **DIDO**
3. Pluggable and scalable distinction maximization layer: better latent feature representation and uncertainty quantification - **LDU**

To end up with, we provide a visualization to summarize the works.

* represents equal contribution

SLURP: Side Learning Uncertainty for Regression Problems
Authors: Xuanlong Yu[1,2], Gianni Franchi[2], Emanuel Aldea[1]
*32nd British Machine Vision Conference (**BMVC**), 2021*

On Monocular Depth Estimation and Uncertainty Quantification using Classification Approaches for Regression
Authors: Xuanlong Yu[1,2], Gianni Franchi[2], Emanuel Aldea[1]
*IEEE International Conference on Image Processing (**ICIP**), 2022 (Oral)*

Latent Discriminant deterministic Uncertainty
Authors: Gianni Franchi*[2], Xuanlong Yu*[1,2], Andrei Bursuc[3], Emanuel Aldea[1], Séverine Dubuisson[4], David Filliat[2]
*European Conference on Computer Vision (**ECCV**), 2022*

MUAD: Multiple Uncertainties for Autonomous Driving, a benchmark for multiple uncertainty types and tasks
Authors: Gianni Franchi*[2], Xuanlong Yu*[1,2], Andrei Bursuc[3], Ángel Tena[5], Rémi Kazmierczak[2], Séverine Dubuisson[4], Emanuel Aldea[1], David Filliat[2]
*33rd British Machine Vision Conference (**BMVC**), 2022*

The Robust Semantic Segmentation UNCV2023 Challenge Results
Authors: Xuanlong Yu[1,2], et al.
*International Conference on Computer Vision Workshop (**ICCVW**), 2023*

Discretization-Induced Dirichlet Posterior for Robust Uncertainty Quantification on Regression
Authors: Xuanlong Yu[1,2], Gianni Franchi[2], Jindong Gu[6], Emanuel Aldea[1]
*The 38th Annual AAAI Conference on Artificial Intelligence (**AAAI**), 2024*

[1] *SATIE, Paris-Saclay U.*, [2] *U2IS, ENSTA Paris*, [3] *valeo.ai*, [4] *LIS, Aix Marseille U.*, [5] *Anyverse*, [6] *U. of Oxford*

Thank you for your attention.

All references:

- Abdar, Moloud et al. (2021). "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information fusion* 76, pp. 243–297.
- Amini, Alexander et al. (2020). "Deep evidential regression". In: *NeurIPS*.
- Blundell, Charles et al. (2015). "Weight uncertainty in neural network". In: *ICML*.
- Charpentier, Bertrand, Daniel Zügner, and Stephan Günnemann (2020). "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts". In: *NeurIPS*.
- Corbière, Charles et al. (2019). "Addressing failure prediction by learning model confidence". In: *NeurIPS*.
- Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8, pp. 861–874.

- Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan (2019). "Deep ensembles: A loss landscape perspective". In: *arXiv preprint arXiv:1912.02757*.
- Franchi, Gianni, Xuanlong Yu, Andrei Bursuc, Emanuel Aldea, et al. (2022). "Latent Discriminant deterministic Uncertainty". In: *ECCV*.
- Franchi, Gianni, Xuanlong Yu, Andrei Bursuc, Angel Tena, et al. (2022). "MUAD: Multiple Uncertainties for Autonomous Driving benchmark for multiple uncertainty types and tasks". In: *BMVC*.
- Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *ICML*.
- Geiger, Andreas et al. (2013). "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11, pp. 1231–1237.
- Goldberg, Paul, Christopher Williams, and Christopher Bishop (1997). "Regression with input-dependent noise: A Gaussian process treatment". In: *NeurIPS*.

- Guo, Chuan et al. (2017). "On calibration of modern neural networks". In.
- Gustafsson, Fredrik K, Martin Danelljan, and Thomas B Schon (2020). "Evaluating scalable bayesian deep learning methods for robust computer vision". In: *CVPR workshops*, pp. 318–319.
- Hendrycks, Dan and Kevin Gimpel (2017). "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *ICLR*.
- Hüllermeier, Eyke and Willem Waegeman (2021). "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods". In: *Machine Learning* 110, pp. 457–506.
- Joo, Taejong, Uijung Chung, and Min-Gwan Seo (2020). "Being bayesian about categorical probability". In: *ICML*.
- Jordan, Michael I et al. (1999). "An introduction to variational methods for graphical models". In: *Machine learning* 37, pp. 183–233.
- Kendall, Alex and Yarin Gal (2017). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *NeurIPS*.

- Krizhevsky, Alex, Geoffrey Hinton, et al. (2009). *Learning multiple layers of features from tiny images*. Tech. rep.
- Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *NeurIPS*.
- Lee, Jin Han et al. (2019). "From big to small: Multi-scale local planar guidance for monocular depth estimation". In: *arXiv preprint arXiv:1907.10326*.
- Liu, Yun et al. (2017). "Richer convolutional features for edge detection". In: *CVPR*.
- Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *CVPR*.
- Mac Aodha, Oisin et al. (2012). "Learning a confidence measure for optical flow". In: *IEEE transactions on pattern analysis and machine intelligence* 35.5, pp. 1107–1120.

- Macêdo, David et al. (2021). "Entropic out-of-distribution detection". In: *IJCNN*.
- Malinin, Andrey and Mark Gales (2018). "Predictive uncertainty estimation via prior networks". In: *NeurIPS*.
- Marks, Robert J et al. (1978). "Detection in Laplace noise". In: *IEEE Transactions on Aerospace and Electronic Systems*.
- Michael I., Jordan (2010). "The exponential family: Conjugate priors". In: *Bayesian modeling and inference course*.
- Nadarajah, Saralees (2005). "A generalized normal distribution". In: *Journal of Applied statistics* 32.7, pp. 685–694.
- Niu, Zhenxing et al. (2016). "Ordinal Regression With Multiple Output CNN for Age Estimation". In: *CVPR*.
- Nix, D.A. and A.S. Weigend (1994). "Estimating the mean and variance of the target probability distribution". In: *ICNN*.
- Shen, Maohao et al. (2023). "Post-hoc Uncertainty Learning using a Dirichlet Meta-Model". In: *AAAI*.

- Silberman, Nathan et al. (2012). "Indoor segmentation and support inference from rgbd images". In: *ECCV*.
- Tishby, Levin, and Solla (1989). "Consistent inference of probabilities in layered networks: predictions and generalizations". In: *IJCNN*. IEEE, pp. 403–409.
- Ulmer, Dennis, Christian Hardmeier, and Jes Frellsen (2023). "Prior and Posterior Networks: A Survey on Evidential Deep Learning Methods For Uncertainty Estimation". In: *Transactions on Machine Learning Research*.
- Upadhyay, Uddeshya et al. (2022). "BayesCap: Bayesian Identity Cap for Calibrated Uncertainty in Frozen Neural Networks". In: *ECCV*.
- Van Amersfoort, Joost et al. (2020). "Uncertainty estimation using a single deep deterministic neural network". In.
- Yu, Xuanlong, Gianni Franchi, and Emanuel Aldea (2021). "SLURP: Side Learning Uncertainty for Regression Problems". In: *BMVC*.
- — (2022). "On Monocular Depth Estimation and Uncertainty Quantification Using Classification Approaches for Regression". In: *ICIP*.

- Yu, Xuanlong, Gianni Franchi, Jindong Gu, et al. (2024). "Discretization-Induced Dirichlet Posterior for Robust Uncertainty Quantification on Regression". In: *AAAI*.
- Yu, Xuanlong, Yi Zuo, et al. (2023). "The Robust Semantic Segmentation UNCV2023 Challenge Results". In: *ICCV workshop*, pp. 4618–4628.

Current section

Discretization-induced regression

Survey on monocular depth estimation task

Notation

We denote the depth table a K dimensional vector $\bar{\mathbf{d}} \in \mathbb{R}^K$, the discrete predicted depth map as $\hat{\mathbf{y}}$, and the final continuous predicted depth map as $\hat{\mathbf{d}}$.

Three key components of discretization-induced regression (classification approaches for regression (CAR):

- Discretization: handcrafted/learned depth table ($\bar{\mathbf{d}}/\hat{\bar{\mathbf{d}}}$);
- Loss function: designs of training targets ($\mathbf{y}$);
- Post-processing: transformations of the prediction map from discrete ($\hat{\mathbf{y}}$) to continuous ($\hat{\mathbf{d}}$).

Discretization-induced regression

Survey on monocular depth estimation task

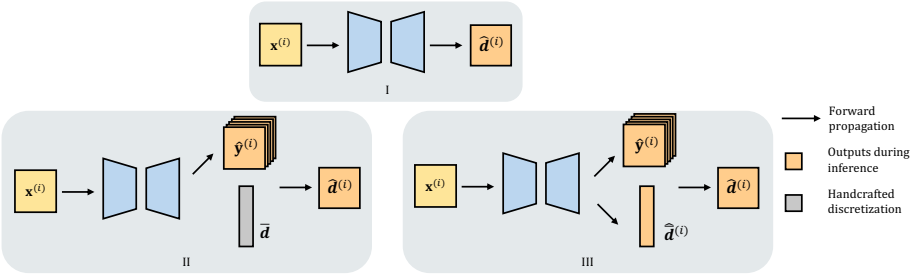


Figure: **Three types of architectures for monocular depth estimation task**. *I*. original regression version; *II*. modeling using handcrafted discretization; *III*. modeling using adaptive discretization.

Discretization-induced regression

Survey on monocular depth estimation task

CAR-MDEs		Discretization				Post processing
		Fully Handcrafted			Adaptive	
		Handcrafted + One-hot	Handcrafted + Ordinal	Handcrafted + Smooth		
Loss function	KL divergence/ Weighted CE loss	-	-	SORN [1]	-	Argmax
	CE loss	-	-	Cao et al. [2] ([3])	-	Soft weighted sum
	Multiple BCE loss	Li et al. [4] ([5, 6])	-	-	-	
	Regression loss	-	-	Yang et al. [7]	-	
	Ordinal Regression loss	DS-SIDENet [8]	-	-	Adabins [9]	
		-	DORN [10] ([11, 12])	-	-	Ordinal sum

Figure: **Summary on CAR-MDE solutions.** In parentheses are methods that use the corresponding schemes as part of their solutions.

[1] Diaz, et al., 'Soft labels for ordinal regression,' in CVPR, 2019.
[2] Cao, et al., 'Estimating depth from monocular images as classification using deep fully convolutional residual networks,' TCSVT, 2017.
[3] Yin, et al., 'Enforcing geometric constraints of virtual normal for depth prediction,' in ICCV, 2019.
[4] Li, et al., 'Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference,' Pattern Recognition, 2018.
[5] Li, et al., 'Deep attention-based classification network for robust depth prediction,' in ACCV, 2018.
[6] Liebel, et al., 'Multidepth: Single-image depth estimation via multi-task regression and classification,' in ITSC, 2019.
[7] Yang, et al., 'Inferring distributions over depth from a single image,' in IROS, 2019.
[8] Ren, et al., 'Deep robust single image depth estimation neural network using scene understanding,' in CVPRW, 2019.
[9] Bhat, et al., 'Adabins: Depth estimation using adaptive bins,' in CVPR, 2021.
[10] Fu, et al., 'Deep ordinal regression network for monocular depth estimation,' in CVPR, 2018.
[11] Lo, et al., 'Depth estimation from monocular images and sparse radar using deep ordinal regression network,' in ICIP, 2021.
[12] Phan, et al., 'Ordinal depth classification using region-based self-attention,' in ICPR, 2021.

Discretization-induced regression

Survey on monocular depth estimation task

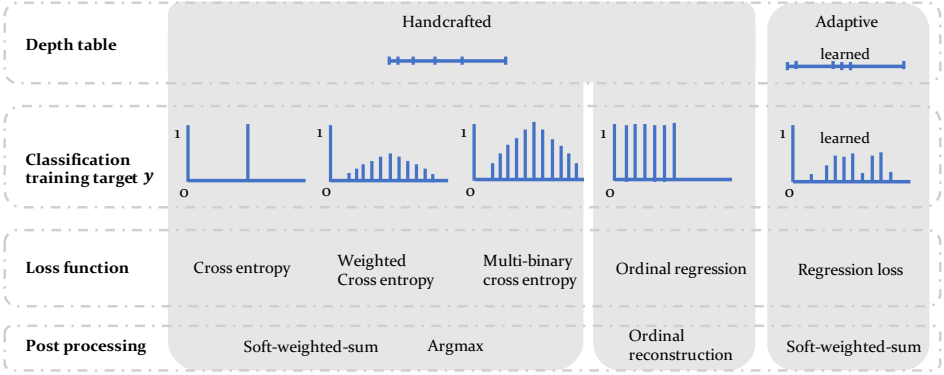


Figure: **Implementation details in CAR-MDE solutions.**

Uncertainty quantification in discretization-induced regression

Expectation of distance

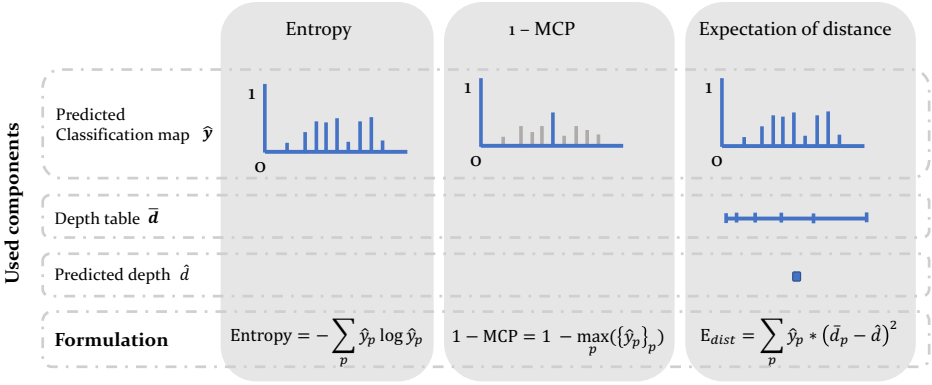


Figure: **Illustration of uncertainty estimators.** Footnote p denotes the p_{th} class (bin) in the vector for each pixel.

Expectation of distance: $E_{dist} = \sum_p (\hat{y}_p * (\bar{d}_p - \hat{d})^2)$

Backbones	BTS								FCN								K
Metrics	$\delta 1\uparrow$	$\delta 2\uparrow$	$\delta 3\uparrow$	Abs Rel$\downarrow$	Sq Rel$\downarrow$	RMSE$\downarrow$	RMSE log$\downarrow$	log10$\downarrow$	$\delta 1\uparrow$	$\delta 2\uparrow$	$\delta 3\uparrow$	Abs Rel$\downarrow$	Sq Rel$\downarrow$	RMSE$\downarrow$	RMSE log$\downarrow$	log10$\downarrow$	
DORN	**0.952**	**0.992**	**0.998**	0.069	**0.267**	**2.802**	**0.103**	**0.029**	**0.940**	0.990	**0.998**	0.076	**0.292**	**2.962**	**0.113**	0.033	80
Cao et al.	0.945	**0.992**	**0.998**	0.077	0.292	2.988	0.111	0.034	0.928	0.989	**0.998**	0.084	0.344	3.223	0.122	0.036	50
Li et al.	0.950	0.990	**0.998**	0.070	0.287	2.928	0.106	0.030	**0.940**	0.988	0.997	0.075	0.314	3.190	0.116	0.033	150
SORN	0.947	**0.992**	**0.998**	0.071	0.290	2.929	0.107	0.031	0.863	0.976	0.995	0.119	0.563	3.938	0.163	0.051	120
Yang et al.	0.951	0.991	**0.998**	**0.065**	0.276	2.897	**0.103**	**0.029**	**0.940**	0.989	0.997	**0.072**	0.302	3.096	**0.113**	**0.032**	128
DS-SIDE	0.950	0.991	**0.998**	0.071	0.275	2.886	0.106	0.032	0.931	0.990	**0.998**	0.079	0.331	3.353	0.119	0.035	80
Adabins	0.935	0.990	**0.998**	0.078	0.347	3.143	0.114	0.033	0.937	**0.991**	**0.998**	0.079	0.331	3.027	**0.113**	0.033	256
DORN	0.952	0.992	**0.998**	0.069	**0.267**	**2.802**	**0.103**	**0.029**	**0.940**	**0.990**	**0.998**	**0.076**	**0.292**	**2.962**	0.113	0.033	80
Cao et al.	**0.953**	0.991	**0.998**	**0.066**	0.268	2.857	**0.103**	**0.029**	0.934	0.989	0.997	**0.076**	0.319	3.124	0.117	0.033	80
Li et al.	0.949	0.990	0.997	0.087	0.305	2.982	0.116	0.037	0.933	0.988	0.997	0.096	0.350	3.157	0.126	0.040	80
SORN	0.949	**0.993**	**0.998**	0.072	0.283	2.902	0.106	0.031	0.863	0.976	0.995	0.122	0.573	3.950	0.165	0.052	80
Yang et al.	0.948	0.991	**0.998**	0.070	0.284	2.973	0.107	0.031	**0.940**	**0.990**	0.997	**0.076**	0.308	3.067	0.115	0.034	80
DS-SIDE	0.950	0.991	**0.998**	0.071	0.275	2.886	0.106	0.032	0.931	**0.990**	**0.998**	0.079	0.331	3.353	0.119	0.035	80
Adabins	0.933	0.989	**0.998**	0.079	0.357	3.203	0.116	0.033	0.937	**0.990**	**0.998**	**0.076**	0.318	3.062	**0.112**	**0.032**	80
Org	0.955	0.993	0.998	0.060	0.249	2.798	0.096	0.027	0.944	0.992	0.998	0.069	0.275	2.938	0.107	0.030	1
MC-Dropout	0.941	0.992	0.998	0.083	0.308	2.910	0.114	0.035	0.918	0.984	0.996	0.085	0.369	3.157	0.125	0.036	1
Deep Ensembles	0.957	0.993	0.999	0.059	0.233	2.688	0.093	0.026	0.946	0.992	0.998	0.068	0.269	2.923	0.106	0.030	1

Table: **Depth accuracy evaluations.** Org: the original BTS (Lee et al., 2019)[30] model and the regression version applied on FCN (Long, Shelhamer, and Darrell, 2015)[31] model.

[30] Jin Han Lee et al. (2019). "From big to small: Multi-scale local planar guidance for monocular depth estimation". In: *arXiv preprint arXiv:1907.10326*.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *CVPR*.

Backbones	BTS						FCN						K
Metrics	AUSE RMSE↓			AUSE AbsRel↓			AUSE RMSE↓			AUSE AbsRel↓			
Methods	1-MCP	S-Entr	E-Dist	1-MCP	S-Entr	E-Dist	1-MCP	S-Entr	E-Dist	1-MCP	S-Entr	E-Dist	
Cao et al.	0.542	0.770	0.133	0.382	0.424	0.411	0.532	0.701	0.127	0.354	0.375	0.375	50
Li et al.	0.174	0.153	0.187	0.276	0.262	0.409	0.137	0.138	0.132	0.241	0.235	0.259	150
SORN	1.371	1.394	0.157	0.939	0.982	0.427	1.244	1.283	0.170	0.754	0.755	0.451	120
Yang et al.	0.141	0.145	**0.094**	0.232	**0.219**	0.256	0.142	0.161	**0.111**	**0.225**	0.226	0.247	128
DS-SIDE	0.698	0.806	0.293	0.525	0.544	0.397	1.212	1.331	0.995	0.630	0.722	0.484	80
Adabins	0.855	0.827	0.179	0.536	0.527	0.377	0.984	0.945	0.191	0.608	0.589	0.398	256
DORN	0.188	0.158	0.128	0.530	0.430	0.303	0.202	0.165	0.135	0.593	0.445	0.283	80
Cao et al.	0.371	0.476	0.119	0.323	0.329	0.356	0.349	0.393	0.117	0.284	0.265	0.308	80
Li et al.	0.206	0.178	0.181	0.355	0.348	0.449	0.170	0.163	0.124	0.318	0.310	0.333	80
SORN	1.367	1.390	0.157	0.900	0.941	0.444	1.228	1.275	0.175	0.725	0.737	0.473	80
Yang et al.	0.194	0.179	**0.099**	0.273	**0.259**	0.271	0.156	0.169	**0.104**	0.258	**0.252**	0.274	80
DS-SIDE	0.698	0.806	0.293	0.525	0.544	0.397	1.212	1.331	0.995	0.630	0.722	0.484	80
Adabins	0.823	0.683	0.181	0.499	0.450	0.360	0.775	0.710	0.234	0.502	0.478	0.391	80
DORN	0.188	0.158	0.128	0.530	0.430	0.303	0.202	0.165	0.135	0.593	0.445	0.283	80
MC-Dropout	0.460 (**Variance**)			0.501 (**Variance**)			0.322 (**Variance**)			0.456 (**Variance**)			1
Deep Ensembles	0.165 (**Variance**)			0.261 (**Variance**)			0.184 (**Variance**)			0.290 (**Variance**)			1

Table: **Depth uncertainty quantification evaluations.** MC-Dropout (Gal and Ghahramani, 2016)[32] and Deep Ensembles (Lakshminarayanan, Pritzel, and Blundell, 2017)[33] will only provide the uncertainty with one method.

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

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

Current section

Latent Discriminant deterministic Uncertainty (LDU)

Notation

We denote $f_\omega(\mathbf{x}) = (g_\omega \circ h_\omega)(\mathbf{x})$ with h_ω and g_ω the feature extractor and the prediction head respectively. $\mathbf{z}{=}h_\omega(\mathbf{x})$ with $\mathbf{z} \in \mathbb{R}^n$ the latent representation of dimension n of $\mathbf{x}$.

Latent Discriminant deterministic Uncertainty (LDU)

Notation

We denote $f_\omega(\mathbf{x}) = (g_\omega \circ h_\omega)(\mathbf{x})$ with h_ω and g_ω the feature extractor and the prediction head respectively. $\mathbf{z}{=}h_\omega(\mathbf{x})$ with $\mathbf{z} \in \mathbb{R}^n$ the latent representation of dimension n of $\mathbf{x}$.

Given a set $\mathbf{p}_\omega{=}\{\mathbf{p}_i\}_{i=1}^m$, of m vectors ($\mathbf{p}_i{\in}\mathbb{R}^n$) that are trainable, we define the Distinction maximization (DM) (Macêdo et al., 2021)[34] layer as follows:

$$\mathrm{DM}_p(\mathbf{z}) = \left[-\|\mathbf{z}-\mathbf{p}_1\|, \ldots, -\|\mathbf{z}-\mathbf{p}_m\|\right]^\top \tag{23}$$

[34] David Macêdo et al. (2021). "Entropic out-of-distribution detection". In: *IJCNN*.

Latent Discriminant deterministic Uncertainty (LDU)

Note that the DM layer is a Lipschitz function:

$$\| \exp(-\mathrm{DM}(\mathbf{z}_1)) - \exp(-\mathrm{DM}(\mathbf{z}_2)) \| \leq k \|\mathbf{z}_1 - \mathbf{z}_2\|, \text{with } k \in \mathbb{R}^+ \qquad (24)$$

Authors of (Van Amersfoort et al., 2020)[35] show the **feature collapse** phenomenon and propose to turn $h_\omega(\mathbf{x})$ a bi-Lipschitz DNN to alleviate this phenomenon.
We relax the bi-Lipschitz constraint by applying the DM layer into the DNN to add scalability.

[35] Joost Van Amersfoort et al. (2020). "Uncertainty estimation using a single deep deterministic neural network". In.

Latent Discriminant deterministic Uncertainty (LDU)

Introduction on LDU

Our DNN can be written as:

$$f_{\omega}(\mathbf{x}) = [g_{\omega} \circ (\exp(-\mathrm{DM}_p(h_{\omega})))](\mathbf{x}) \tag{25}$$

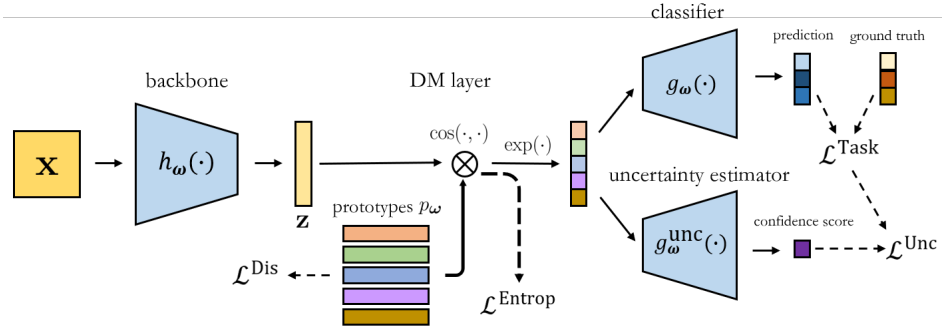


Figure: **Overview of LDU**

Latent Discriminant deterministic Uncertainty (LDU)

Introduction on LDU

We propose to train g_ω^{unc} to predict the error of the DNN (Corbière et al., 2019; Yu, Franchi, and Aldea, 2021)[36], which helps us link the prototypes to the uncertainty.

Given an input data $\mathbf{x}$, its groundtruth y, and its loss $\mathcal{L}^{\text{Task}}(g_\omega(\mathbf{x}), y)$, we train g_ω^{unc} by minimizing:

$$\mathcal{L}^{\text{Unc}} = \text{BCE}\left(\left[g_\omega^{\text{unc}} \circ (\exp(-\text{DM}_p(h_\omega)))\right](\mathbf{x}), \quad \mathcal{L}^{\text{Task}}(g_\omega(\mathbf{x}), y)\right) \tag{26}$$

[36] Charles 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*.

Latent Discriminant deterministic Uncertainty (LDU)

Introduction on LDU

To constrain the latent representation to stay close to different prototypes:

$$\mathcal{L}^{\text{Entrop}} = \sum_{i=1}^{n} \sigma(\text{DM}_p(h_\omega))_i \cdot \log(\sigma(\text{DM}_p(h_\omega))_i), \tag{27}$$

To force the prototypes to be dissimilar:

$$\mathcal{L}^{\text{Dis}} = -\sum_{i<j} \|\mathbf{p}_i - \mathbf{p}_j\|. \tag{28}$$

Total loss function:

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{Task}} + \lambda_{\text{LDU}}(\mathcal{L}^{\text{Entrop}} + \mathcal{L}^{\text{Dis}} + \mathcal{L}^{\text{Unc}}) \tag{29}$$

with λ_{LDU} the hyperparameter for the additional losses.

Experiments

Toy example

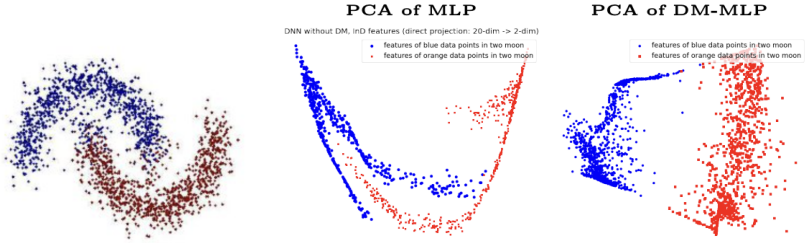


Figure: **PCA 2D projection for the latent features.** Left: features from a standard MLP; Right: features from a DM-MLP. Both networks are trained on the two moons dataset. Blue and red points indicate the features of data points of the two classes, respectively.

Experiments

Monocular depth results

Method	Depth performance								Uncertainty performance	
	d1↑	d2↑	d3↑	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	log10↓	AUSE RMSE↓	AUSE Absrel↓
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 $\#\mathbf{p} = 5$, $\lambda = 1.0$	0.954	**0.993**	0.998	0.063	0.253	2.768	0.098	0.027	**0.08**	**0.21**
LDU $\#\mathbf{p} = 15$, $\lambda = 0.1$	0.954	**0.993**	0.998	0.062	0.249	2.769	0.098	0.027	0.10	0.28
LDU $\#\mathbf{p} = 30$, $\lambda = 0.1$	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 dataset.**

Experiments

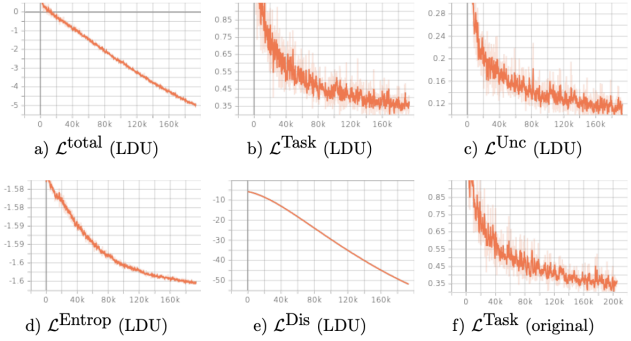


Fig. A1. Illustrations on training curves for different losses. Figure a) - e): training curves for the model with LDU modifications; Figure f): training curve for the original model. The $\mathcal{L}^{\text{Task}}$ is silog loss for depth regression as defined in BTS [46].

Current section

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	-	-	-	-
Raincouver	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

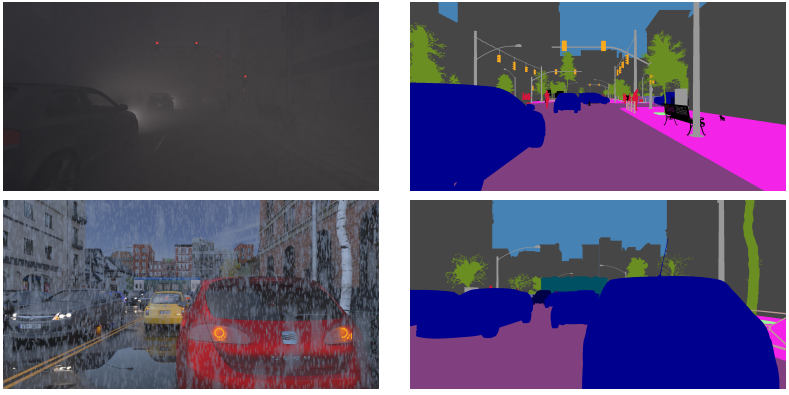


Figure: **Snapshots from the MUAD dataset** showing different types of adverse conditions and events to evaluate perception models.

MUAD dataset

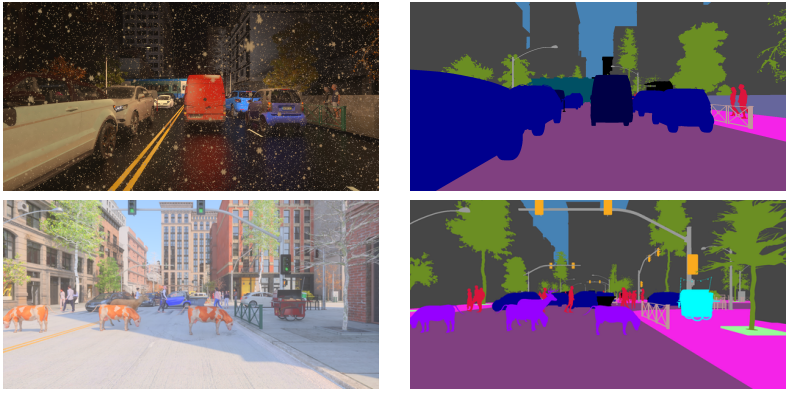


Figure: **Snapshots from the MUAD dataset** showing different types of adverse conditions and events to evaluate perception models.

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.

Methods	normal set					low adv. without OOD set					high adv. without OOD set					normal set overhead sun				
	Depth results			Uncertainty results		Depth results			Uncertainty results		Depth results			Uncertainty results		Depth results			Uncertainty results	
	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓
Baseline	0.922	0.114	3.357	-	-	0.786	**0.147**	5.005	-	-	0.632	**0.207**	6.989	-	-	0.951	0.090	3.646	-	-
Deep Ensembles	**0.929**	**0.111**	**3.199**	**0.291**	0.060	0.767	0.156	4.892	**0.740**	0.105	0.566	0.243	7.498	**1.182**	0.153	**0.955**	**0.083**	3.479	**0.336**	0.055
MC Dropout	0.919	0.119	3.209	0.634	0.061	**0.798**	0.151	**4.580**	1.063	0.098	**0.657**	**0.207**	**6.278**	1.382	**0.128**	0.948	0.092	**3.407**	0.786	0.058
Single-PU	0.905	0.132	3.230	0.313	0.081	0.773	0.159	4.865	0.789	0.112	0.571	0.248	7.680	1.740	0.171	0.946	0.105	3.546	0.358	0.079
SLURP	0.922	0.114	3.357	0.467	**0.048**	0.786	**0.147**	5.005	1.167	**0.090**	0.632	**0.207**	6.989	1.707	**0.128**	0.951	0.090	3.646	0.525	**0.033**

Methods	OOD set					low adv. with OOD set					high adv. with OOD set				
	Depth results			Uncertainty results		Depth results			Uncertainty results		Depth results			Uncertainty results	
	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓	d1 ↑	AbsRel ↓	RMSE ↓	AUSE RMSE ↓	AUSE Absrel ↓
Baseline	0.896	0.125	3.616	-	-	0.713	2.637	4.764	-	-	0.555	0.459	6.916	-	-
Deep Ensembles	**0.903**	**0.114**	3.447	**0.427**	0.074	0.709	**1.810**	4.707	**0.692**	**0.129**	0.521	**0.331**	7.411	**1.072**	**0.151**
MC Dropout	0.893	0.145	**3.432**	0.724	0.080	**0.744**	3.925	**4.364**	0.927	0.206	**0.610**	0.545	**6.176**	1.245	0.314
Single-PU	0.888	0.132	3.463	0.447	0.095	0.714	4.349	4.716	0.744	0.482	0.529	0.351	7.627	1.347	0.156
SLURP	0.896	0.125	3.616	0.721	**0.068**	0.713	2.637	4.764	1.072	0.212	0.555	0.459	6.916	1.564	**0.151**

Table: **Comparative results for monocular depth on MUAD**. We use NeWCRFs as the based DNN for monocular depth task.

	KITTI							
Training set	d1↑	d2↑	d3↑	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	SILog↓
KITTI	0.975	0.997	0.999	0.052	0.148	2.072	0.078	6.9859
Virtual KITTI 2	0.835	0.957	0.989	0.129	0.706	4.039	0.177	15.534
MUAD	0.731	0.927	0.983	0.187	1.059	4.754	0.227	18.581

Table: **Comparative results for monocular depth estimation simple domain adaptation from MUAD to KITTI eigen-split**. First row is the original baseline, the second and the third rows are the performance of the model trained directly on Virtual KITTI 2 and MUAD respectively.

Monocular depth estimation results

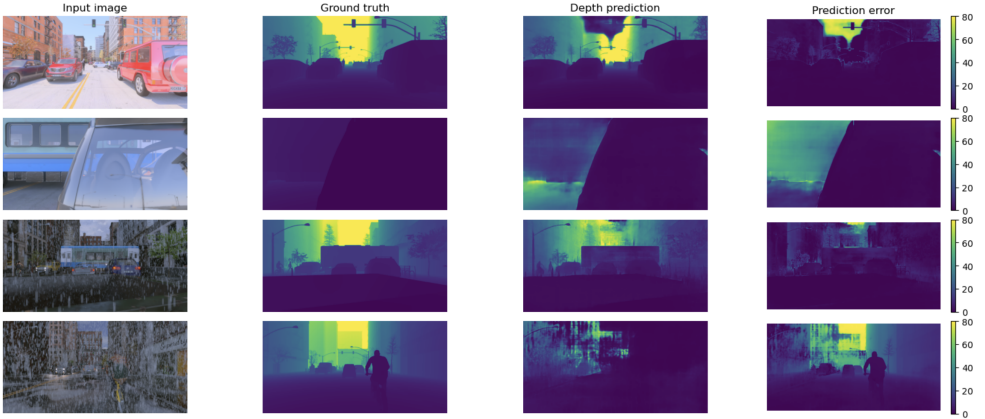


Figure: **Visualization of monocular depth estimates on the testing images given by the NeWCRFs model.** The first row and the second rows show the clear weather cases, and the rest show the rainy cases. The train and bicycle on the last three inputs are

Semantic segmentation results

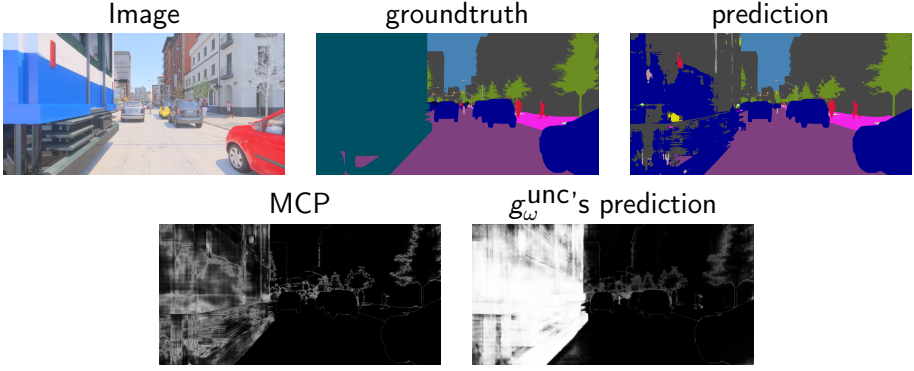


Figure: Illustration of the different confidence scores on one image of MUAD. Note that the class `train`, `bicycle`, `Stand food` and the `animals` are OOD.

Semantic segmentation results

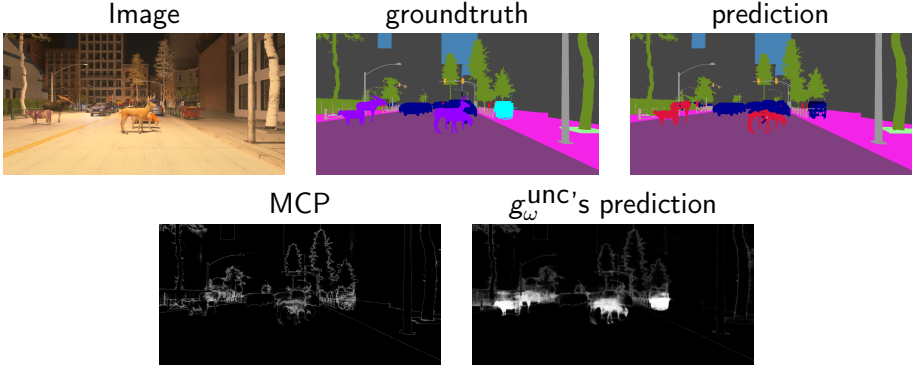


Figure: Illustration of the different confidence scores on one image of MUAD. Note that the class `train`, `bicycle`, `Stand food` and the `animals` are OOD.

Current section

Experiment - Aleatoric uncertainty estimation on Monocular Depth Estimation

S	Metrics	Original	+ Ggau	+ Sgau	+ NIG Uncer1	+ NIG Uncer2	Ours (DIDO) sep. enc. σ_{Θ_2}	Ours (+ Lap) shar. enc. σ_{Θ_1}	Ours (+ Lap) sep. enc. σ_{Θ_1}
0	AUSE-REL ↓	0.013	0.014	0.013	**0.012**	0.013	0.050	0.013	0.013
	AUSE-RMSE ↓	0.204	0.258	**0.202**	0.208	0.205	3.277	0.205	0.203
	AURG-REL ↑	0.023	0.023	0.023	**0.024**	0.023	-0.016	0.023	0.023
	AURG-RMSE ↑	1.869	1.815	**1.871**	1.865	1.868	-1.453	1.869	1.870
1	AUSE-REL ↓	0.019	0.021	0.019	**0.018**	0.020	0.064	**0.018**	0.019
	AUSE-RMSE ↓	0.340	0.482	**0.332**	0.335	0.350	4.085	**0.332**	0.336
	AURG-REL ↑	0.031	0.029	0.031	**0.032**	0.030	-0.014	**0.032**	0.031
	AURG-RMSE ↑	2.357	2.215	**2.365**	2.362	2.347	-1.388	**2.365**	2.361
2	AUSE-REL ↓	0.024	0.026	0.023	**0.022**	0.025	0.077	**0.022**	0.023
	AUSE-RMSE ↓	0.483	0.707	**0.463**	0.479	0.505	4.788	0.464	0.468
	AURG-REL ↑	0.038	0.035	**0.039**	**0.039**	0.037	-0.016	**0.039**	0.038
	AURG-RMSE ↑	2.759	2.535	**2.779**	2.763	2.737	-1.546	2.777	2.774
3	AUSE-REL ↓	0.033	0.036	**0.031**	**0.031**	0.035	0.099	**0.031**	**0.031**
	AUSE-RMSE ↓	0.795	1.176	0.737	0.806	0.846	5.743	0.749	**0.730**
	AURG-REL ↑	0.047	0.044	**0.049**	**0.049**	0.045	-0.019	**0.049**	**0.049**
	AURG-RMSE ↑	3.243	2.862	3.301	3.232	3.192	-1.705	3.289	**3.308**
4	AUSE-REL ↓	0.056	0.057	0.050	0.053	0.059	0.125	0.051	**0.049**
	AUSE-RMSE ↓	1.517	2.380	1.364	1.582	1.607	5.743	1.430	**1.268**
	AURG-REL ↑	0.051	0.051	0.058	0.054	0.049	-0.019	0.056	**0.059**
	AURG-RMSE ↑	3.680	2.817	3.834	3.615	3.590	-1.705	3.767	**3.929**
5	AUSE-REL ↓	0.071	0.082	0.064	0.069	0.071	0.140	0.066	**0.059**
	AUSE-RMSE ↓	2.202	3.878	2.043	2.414	2.307	7.354	2.157	**1.760**
	AURG-REL ↑	0.056	0.045	0.063	0.057	0.055	-0.014	0.061	**0.067**
	AURG-RMSE ↑	4.054	2.377	4.213	3.842	3.949	-1.098	4.098	**4.496**

Table: **Aleatoric uncertainty estimation results on Monocular Depth Estimation.** Datasets used are KITTI (S=0) and KITTI-C (S>0). All results are averaged by three runs.

		Auxiliary Uncertainty Estimator (SLURP)							Modified main DNN				Training-free	
S	Metrics	Original	+ Ggau	+ Sgau	+ NIG	Ours σ_{Θ_1} sep. enc.(+Lap)	Ours σ_{Θ_2} sep. enc.(DIDO)	Ours σ_{Θ_2} shar. enc.(DIDO)	Single PU	LDU	Evi.	DEns.	Grad. (inv.)	Inject.
0	AUC ↑	99.1	96.8	99.0	90.9	99.9	**100.0**	99.9	89.0	96.5	76.7	93.5	85.6	58.4
	AUPR ↑	94.6	80.3	91.6	57.6	99.7	**100.0**	99.0	62.0	93.8	42.6	70.0	76.3	28.1
	Sky-All ↓	0.643	0.277	0.934	0.983	0.961	0.015	0.018	0.005	0.278	0.986	0.005	**0.001**	0.800
1	AUC ↑	98.4	96.0	99.0	90.4	99.8	**100.0**	99.9	86.9	96.3	69.7	92.8	76.9	58.5
	AUPR ↑	93.3	77.9	92.4	57.4	99.5	**99.9**	98.9	59.1	93.5	37.4	68.0	69.8	28.2
	Sky-All ↓	0.742	0.274	0.935	0.978	0.962	0.016	0.018	0.005	0.277	0.988	0.005	**0.002**	0.799
2	AUC ↑	97.6	95.6	99.0	90.2	99.7	**99.9**	**99.9**	86.6	95.9	65.4	92.3	75.6	58.4
	AUPR↑	91.9	76.5	92.8	57.5	99.3	**99.8**	98.8	58.9	93.0	34.5	67.0	67.8	28.1
	Sky-All ↓	0.784	0.274	0.937	0.973	0.962	0.017	0.018	0.005	0.280	0.990	0.005	**0.002**	0.803
3	AUC ↑	96.8	95.0	98.9	90.0	99.4	**99.9**	99.7	86.6	95.9	62.3	91.6	73.6	58.4
	AUPR ↑	90.5	75.1	92.9	58.2	98.9	**99.7**	98.1	59.5	92.8	32.8	65.7	64.5	28.2
	Sky-All ↓	0.815	0.277	0.938	0.965	0.960	0.018	0.020	0.005	0.283	0.992	0.005	**0.002**	0.809
4	AUC ↑	94.9	93.2	98.5	89.8	99.0	**99.6**	99.5	87.2	96.1	58.8	91.8	71.3	58.4
	AUPR ↑	87.2	71.1	92.4	59.8	98.2	**99.1**	97.2	61.7	92.9	31.2	67.2	60.0	28.3
	Sky-All ↓	0.868	0.284	0.940	0.945	0.959	0.023	0.022	0.005	0.288	0.994	0.005	**0.002**	0.819
5	AUC ↑	92.5	90.3	97.6	89.6	98.2	98.5	**99.0**	87.5	96.5	58.5	92.2	66.8	57.8
	AUPR ↑	83.8	66.6	91.4	63.6	96.8	**97.1**	96.1	64.6	93.7	32.8	70.4	53.8	28.2
	Sky-All ↓	0.902	0.299	0.943	0.909	0.959	0.035	0.026	0.005	0.295	0.996	0.005	**0.002**	0.839

Table: **Epistemic uncertainty estimation results encountering unseen pattern on Monocular depth estimation task.** The evaluation datasets used here are KITTI Seg-Depth (S=0) and KITTI Seg-Depth-C (S>0).

	Auxiliary Uncertainty Estimator (SLURP)							Modified task DNN				Training-free	
Metrics	Original	+ Ggau	+ Sgau	+ NIG	Ours σ_{Θ_1}sep. enc.(+Lap)	Ours σ_{Θ_2}sep. enc.(DIDO)	Ours σ_{Θ_2}shar. enc.(DIDO)	Single PU	LDU	Evi.	DEns.	Grad.	Inject.
AUC ↑	59.8	80.9	74.5	57.0	65.4	98.1	**98.4**	64.2	58.1	70.6	62.1	78.4	18.3
AUPR ↑	76.7	90.9	88.4	75.5	82.5	99.3	**99.4**	78.3	79.5	77.8	76.7	92.6	62.3

Table: **Epistemic uncertainty estimation results encountering dataset change on Monocular depth estimation task.** The evaluation dataset used here is NYU indoor depth dataset.

Super Resolution (Metric: UCE ↓)

Dataset	S	Original (Ggau)	+ Sgau	+ NIG	Ours σ_{Θ_1} (+ Lap)
Set5	0	0.0088	0.0083	**0.0018**	0.0019
	1	0.0186	0.0180	**0.0156**	0.0157
	2	0.0253	0.0243	**0.0226**	0.0227
	3	0.0363	0.0341	0.0333	**0.0332**
	4	0.0434	0.0394	0.0392	**0.0389**
	5	0.0525	0.0462	0.0464	**0.0040**
Set14	0	0.0137	0.0092	**0.0040**	**0.0040**
	1	0.0221	0.0195	0.0176	**0.0174**
	2	0.0281	0.0255	0.0241	**0.0240**
	3	0.0350	0.0318	0.0310	**0.0308**
	4	0.0408	0.0368	0.0364	**0.0362**
	5	0.0509	0.0465	0.0465	**0.0461**
BSDS100	0	0.0124	0.0071	0.0036	**0.0033**
	1	0.0204	0.0174	0.0162	**0.0160**
	2	0.0271	0.0237	0.0229	**0.0227**
	3	0.0332	0.0288	0.0286	**0.0358**
	4	0.0425	0.0363	0.0363	**0.0358**
	5	0.0539	0.0459	0.0460	**0.0453**

Table: **Aleatoric uncertainty estimation results on Super-Resolution task.** Datasets with an S (severity) greater than 1 are the -C variants of the corresponding clean dataset.

Datasets	Criteria	MC	EEns.	Single PU	DEns.	Confid	MHP	Ours-light	Ours
FlyingChairs	AUSE-EPE	2.75	1.97	1.28	1.26	1.92	1.88	**1.16**	1.20
	AUROC	0.896	0.900	**0.977**	0.959	0.945	0.936	**0.977**	0.974
KITTI	AUSE-EPE	3.57	4.41	3.71	**3.45**	6.56	5.48	5.20	4.69
	AUROC	0.870	**0.904**	0.848	0.866	0.687	0.854	0.791	0.800
Sintel Clean	AUSE-EPE	3.33	2.89	2.74	3.02	5.28	**2.61**	3.02	2.91
	AUROC	0.861	0.825	**0.925**	0.895	0.767	0.886	0.889	0.896
Sintel Final	AUSE-EPE	3.30	3.02	3.09	3.05	6.06	**2.71**	2.95	2.86
	AUROC	0.858	0.814	**0.916**	0.899	0.728	0.878	0.901	0.906

Table: **Uncertainty quantification performance on optical flow task.** Bold value: result with the best performance. Blue value: second performance.

Task	Criteria	MC	EEns.	Single-PU	DEns.	Confid	MHP	Ours-light	Ours
Monocular depth	Runtime (ms)	386	144	98	286	106	-	-	**88**
	# Param. (M)	**47.0**	141.0	94.0	282.0	94.7	-	-	87.2
Optical Flow	Runtime (ms)	79	65	**64**	66	65	67	65	76
	# Param. (M)	**38.7**	116.1	**38.7**	116.3	78.6	78.8	57.0	105.3

Table: Average time cost for processing one image and number of parameters of the model(s) (main task + uncertainty task).

$$
\begin{aligned}
\mathbb{E}_{P(\boldsymbol{\pi}|\Theta_{\mathbf{2}})}[\log P(\tilde{\mathcal{D}}|\boldsymbol{\pi})] = \mathbb{E}_{Dir(\boldsymbol{\pi}|\boldsymbol{\alpha})}[\log P(y|\boldsymbol{\pi}, x)] &= \mathbb{E}_{Dir(\boldsymbol{\pi}|\boldsymbol{\alpha})}[\log \boldsymbol{\pi}_y] \\
&= \int \log \boldsymbol{\pi}_y Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}) d\boldsymbol{\pi} \\
&= \int_{\mathbf{0}}^{\mathbf{1}} \log \boldsymbol{\pi}_y Beta(\alpha_y, S - \alpha_y) d\boldsymbol{\pi}_y \\
&= \int_{\mathbf{0}}^{\mathbf{1}} \log \boldsymbol{\pi}_y \frac{\boldsymbol{\pi}_y^{\alpha_y - \mathbf{1}} (1 - \boldsymbol{\pi}_y)^{S - \alpha_y - \mathbf{1}}}{B(\alpha_y, S - \alpha_y)} d\boldsymbol{\pi}_y \\
&= \frac{\int_{\mathbf{0}}^{\mathbf{1}} \frac{d\boldsymbol{\pi}_y^{\boldsymbol{\alpha_y - 1}}}{d\alpha_y} (1 - \boldsymbol{\pi}_y)^{S - \alpha_y - \mathbf{1}} d\boldsymbol{\pi}_y}{B(\alpha_y, S - \alpha_y)} \\
&= \frac{\frac{d}{d\alpha_y} \int_{\mathbf{0}}^{\mathbf{1}} \boldsymbol{\pi}_y^{\alpha_y - \mathbf{1}} (1 - \boldsymbol{\pi_y})^{S - \alpha_y - \mathbf{1}} d\boldsymbol{\pi}_y}{B(\alpha_y, S - \alpha_y)} \\
&= \frac{1}{B(\alpha_y, S - \alpha_y)} \frac{dB(\alpha_y, S - \alpha_y)}{d\alpha_y} \\
&= \frac{d \log B(\alpha_y, S - \alpha_y)}{d\alpha_y} \\
&= \frac{d \left(\log \Gamma(\alpha_y) + \log \Gamma(S - \alpha_y) - \log \Gamma(S) \right)}{d\alpha_y} \\
&= \frac{d \log \Gamma(\alpha_y)}{d\alpha_y} - \frac{d \log \Gamma(S)}{dS} \\
&= \psi(\alpha_y) - \psi(S) \qquad (30)
\end{aligned}
$$