Discretization-Induced Dirichlet Posterior for Robust Uncertainty Quantification on Regression

Xuanlong Yu, Gianni Franchi, Jindong Gu, Emanuel Aldea

Discretization-Induced Dirichlet Posterior for Robust Uncertainty Quantification on Regression

Xuanlong Yu[1,2], **Gianni Franchi**[2], **Jindong Gu**[3], **Emanuel Aldea**[1]

[1]SATIE, Paris-Saclay University
[2]U2IS, ENSTA Paris, Institut Polytechnique de Paris
[3]University of Oxford

Abstract

Uncertainty quantification is critical for deploying deep neural networks (DNNs) in real-world applications. An Auxiliary Uncertainty Estimator (AuxUE) is one of the most effective means to estimate the uncertainty of the main task prediction without modifying the main task model. To be considered robust, an AuxUE must be capable of maintaining its performance and triggering higher uncertainties while encountering Out-of-Distribution (OOD) inputs, i.e., to provide robust aleatoric and epistemic uncertainty. However, for vision regression tasks, current AuxUE designs are mainly adopted for aleatoric uncertainty estimates, and AuxUE robustness has not been explored. In this work, we propose a generalized AuxUE scheme for more robust uncertainty quantification on regression tasks. Concretely, to achieve a more robust aleatoric uncertainty estimation, different distribution assumptions are considered for heteroscedastic noise, and Laplace distribution is finally chosen to approximate the prediction error. For epistemic uncertainty, we propose a novel solution named Discretization-Induced Dirichlet pOsterior (DIDO), which models the Dirichlet posterior on the discretized prediction error. Extensive experiments on age estimation, monocular depth estimation, and super-resolution tasks show that our proposed method can provide robust uncertainty estimates in the face of noisy inputs and that it can be scalable to both image-level and pixel-wise tasks. Code is available at https://github.com/ENSTA-U2IS/DIDO.

1 Introduction

Uncertainty quantification in deep learning has gained significant attention in recent years (Blundell et al. 2015; Kendall and Gal 2017; Lakshminarayanan, Pritzel, and Blundell 2017; Abdar et al. 2021). Deep Neural Networks (DNNs) frequently provide overconfident predictions and lack uncertainty estimates, especially for regression models outputting single point estimates, affecting the interpretability and credibility of the prediction results.

There are two types of uncertainty in DNNs: unavoidable aleatoric uncertainty caused by data noise, and reducible epistemic or knowledge uncertainty due to insufficient training data (Hüllermeier and Waegeman 2021; Kendall and Gal 2017; Malinin and Gales 2018). Disentangling and estimating them can better guide the decision-making based on DNN predictions. Many seminal methods (Blundell et al. 2015; Gal and Ghahramani 2016; Lakshminarayanan, Pritzel, and Blundell 2017; Kendall and Gal 2017; Wen, Tran, and Ba 2020; Franchi et al. 2022) have been proposed to capture these two types of uncertainty. However, these methods require extensive modifications to the underlying model structure or more computational cost. Furthermore, since DNNs are often designed as task-oriented, obtaining uncertainty estimates by changing the structure of DNNs might reduce main task performance.

As one of the most effective methods, Auxiliary Uncertainty Estimators (AuxUE) (Corbière et al. 2019; Yu, Franchi, and Aldea 2021; Jain et al. 2021; Corbière et al. 2021; Besnier et al. 2021; Upadhyay et al. 2022; Shen et al. 2023) aim to obtain uncertainty estimates without affecting the main task performance. AuxUEs are DNNs that rely on the main task models used for estimating the uncertainty of the main task prediction. They are trained using the input, output, or intermediate features of the *pre-trained* main task model. In practice, the model inputs can be distribution-shifted from the training set, such as samples disturbed by noise (Hendrycks and Dietterich 2019), or even Out-of-Distribution (OOD) data. The pre-trained main task models mainly exhibit aleatoric uncertainty in the outputs given the In-Distribution (ID) inputs. Meanwhile, higher epistemic uncertainty is expected to be raised when OOD data is fed. A robust AuxUE is required in this case to provide robust aleatoric uncertainty estimates when facing In-Distribution (ID) inputs and epistemic uncertainty estimates when encountering OOD inputs. This can help to make effective decisions under anomalies and uncertainty (Guo et al. 2022), such as in autonomous driving (Arnez et al. 2020). Based on these requirements, the prerequisite for a robust AuxUE, thus, is to disentangle the two types of uncertainty. Disentangling can help estimate the epistemic uncertainty and find a more robust aleatoric uncertainty estimation solution.

For vision regression tasks, basic AuxUE addresses only aleatoric uncertainty estimation (Yu, Franchi, and Aldea 2021). Recent works (Upadhyay et al. 2022; Qu et al. 2022) aim to improve the generalization ability of the basic AuxUEs. In DEUP (Jain et al. 2021), the authors propose to add a density estimator based on normalizing flows (Rezende and Mohamed 2015) in the AuxUE, yet challenging to apply on pixel-wise vision tasks. In the current context, both the robustness analysis and modeling of epistemic uncertainty are underexplored for vision regression problems.

To further explore robust aleatoric and epistemic uncertainty estimation in vision regression tasks, in this work, we propose a novel uncertainty quantification solution based on AuxUE. For estimating aleatoric uncertainty, we follow the approach of previous works such as (Nix and Weigend 1994; Kendall and Gal 2017; Yu, Franchi, and Aldea 2021; Upadhyay et al. 2022) and model the heteroscedastic noise using different distribution assumptions. For epistemic uncertainty quantification, we apply a discretization approach to the continuous prediction errors of the main task. This helps to mitigate the numerical impact of the training targets, which may be distributed in a long-tailed manner. With the discretized prediction errors, we propose parameterizing Dirichlet posterior (Sensoy, Kaplan, and Kandemir 2018; Charpentier, Zügner, and Günnemann 2020; Joo, Chung, and Seo 2020) for estimating epistemic uncertainty without relying on OOD data during the training process.

In summary, our contributions are as follows: (1) We propose a generalized AuxUE solution for aleatoric and epistemic uncertainty estimation; (2) We propose Discretization-Induced Dirichlet pOsterior (DIDO), a new epistemic uncertainty estimation strategy for regression, which, to the best of our knowledge, is the only existing work employing this distribution for regression; (3) We demonstrate that assuming the noise which affects the main task predictions to follow Laplace distribution can help AuxUE achieve a more robust aleatoric uncertainty estimation; (4) We propose a new evaluation strategy for the OOD analysis of pixel-wise regression tasks based on systematically non-annotated patterns. We show the robustness and scalability of the proposed generalized AuxUE and DIDO on the age estimation, super-resolution and monocular depth estimation tasks.

2 Related works

Auxiliary uncertainty estimation Auxiliary uncertainty estimation strategies can be divided into two categories: unsupervised and supervised. For the former, Dropout layer injection (Mi et al. 2022; Gal and Ghahramani 2016) samples the network by forward propagations, and (Hornauer and Belagiannis 2022) proposed to use the gradients from the back-propagation. For the latter, AuxUEs are applied to obtain the uncertainty. In addition to regression-oriented ones presented in Section 1, we here introduce classification-oriented solutions. ConfidNet (Corbière et al. 2019) and KLoS (Corbière et al. 2021) learn the true class probability and evidence for the DNNs, respectively. Shen et al. (Shen et al. 2023) apply evidential classification (Joo, Chung, and Seo 2020) to their AuxUE. ObsNet (Besnier et al. 2021) uses adversarial noise to provide more abundant training targets in semantic segmentation task for their AuxUE.

Evidential deep learning and Dirichlet networks Evidential deep learning (Ulmer 2021) (EDL) is a modern application of the Dempster-Shafer Theory (Dempster 1968) to estimate epistemic uncertainty with single forward propagation. In classification tasks, EDL is usually formed as parameterizing a prior (Malinin and Gales 2018, 2019) or a posterior (Joo, Chung, and Seo 2020; Charpentier, Zügner, and Günnemann 2020; Charpentier et al. 2022; Sensoy, Kaplan, and Kandemir 2018) Dirichlet distribution. In regression problems, EDL estimates the parameters of the conjugate prior of Gaussian distribution (Amini et al. 2020; Charpentier et al. 2022; Malinin et al. 2020). Multi-task learning is recently applied to alleviate main task performance degradation due to applying such techniques (Oh and Shin 2022), yet using AuxUE will not affect main task performance. Therefore, we apply EDL to our AuxUE. Moreover, we are the first to apply the Dirichlet network to the regression tasks by discretizing the main task prediction errors.

Robustness of uncertainty estimation A robust uncertainty estimator should show stable performance when encountering images perturbed to varying degrees (Michaelis et al. 2019; Hendrycks and Dietterich 2019; Kamann and Rother 2021). Similar studies are applied to evaluate the robustness of uncertainty estimates (Yeo, Kar, and Zamir 2021; Franchi et al. 2022). Meanwhile, it should provide a higher uncertainty when facing OOD data, such as in classification tasks (Hendrycks and Gimpel 2017; Liang, Li, and Srikant 2018). In image-level regression, we can use the definition of OOD from image classification (Techapanurak and Okatani 2021) in, for example, age estimation task. But for pixel-wise regression tasks, the notion of OOD data is ill-defined. Typical OOD analysis estimates uncertainty on a different dataset than the training dataset (Charpentier et al. 2022). Yet, image patterns that are rarely assigned ground truth values in the training set can also be regarded as OOD. In this work, we also provide a new evaluation strategy for OOD patterns based on outdoor depth estimation to compensate for this experimental shortfall.

3 Method

In this section, we will first provide the notations and the problem settings. We define a training dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_i^N$ where N is the number of images. We consider that $\mathbf{x}, \mathbf{y}$ are drawn from a joint distribution $P(\mathbf{x}, \mathbf{y})$. A pipeline for the main task and auxiliary uncertainty estimation is shown in Fig. 1. We define a main task DNN $f_{\boldsymbol{\omega}}$ with trainable parameters $\boldsymbol{\omega}$ as shown in the blue area in Fig. 1. Similar to (Blundell et al. 2015), we view $f_{\boldsymbol{\omega}}$ as a probabilistic model $P(y|\mathbf{x}, \boldsymbol{\omega})$ which follows a Gaussian distribution $\mathcal{N}(y|\mu, \sigma^2)$ (Bishop and Nasrabadi 2006). The variable σ^2 represents the variance of the noise in the DNN's prediction, and the variable μ is the prediction $\hat{y} = f_{\boldsymbol{\omega}}(\mathbf{x})$ in this case. The noise is considered here to be homoscedastic as all data have the same noise. The parameter $\boldsymbol{\omega}$ is optimized by maximizing the log-likelihood: $\hat{\boldsymbol{\omega}} = \text{argmax}_{\boldsymbol{\omega}} \log\left(P(\mathcal{D}|\boldsymbol{\omega})\right)$ which is often performed by minimizing Negative Log Likelihood (NLL) loss in practice. With the above-mentioned Gaussian assumption on $\hat{y}$, the NLL loss optimizes with the same objective as the Mean Square Error loss (Bishop and Nasrabadi 2006), thus, only the prediction goal y is considered, and the uncertainty modeling is absent in the main task model training objective.

AuxUE aims to obtain this missing uncertainty estimation without modifying $\hat{\boldsymbol{\omega}}$. We consider two DNNs σ_{Θ_1} and σ_{Θ_2} in our generalized AuxUE with parameters Θ_1 and Θ_2, i.e., the two DNNs in the orange area of the Fig. 1. σ_{Θ_1} is

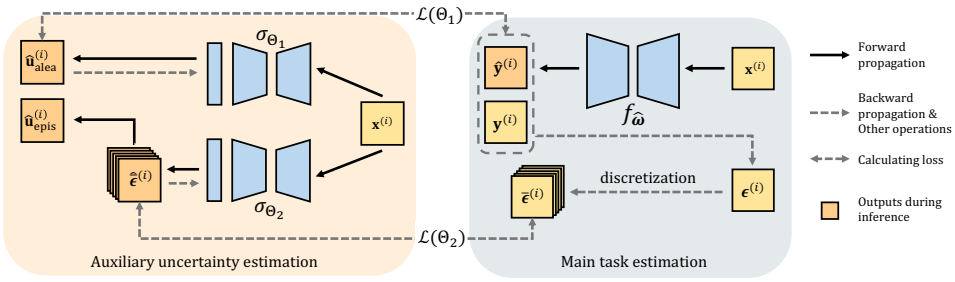


Figure 1: Pipeline of our proposed AuxUE solution. A generalized AuxUE is considered with two DNNs σ_{Θ_1} and σ_{Θ_2} for estimating aleatoric and epistemic uncertainty, respectively. Presented notations are consistent with and described in Section 3. The encoder parts of both DNNs can be shared, we compare the performance in Section 4.3. The input of AuxUE can be the input, output, or intermediate features of $f_{\hat{\boldsymbol{\omega}}}$, we here simplify it to the image $\mathbf{x}^{(i)}$ for brevity.

for estimating aleatoric uncertainty $\mathbf{u}_{\text{alea}}$, and σ_{Θ_2} is for estimating epistemic uncertainty $\mathbf{u}_{\text{epis}}$. The backbone of σ_{Θ_1} and σ_{Θ_2} are based on the basic AuxUEs such as ConfidNet (Corbière et al. 2019), BayesCap (Upadhyay et al. 2022) and SLURP (Yu, Franchi, and Aldea 2021) depending on the tasks. The input of AuxUE can be the input, output, or intermediate features of $f_{\hat{\boldsymbol{\omega}}}$ and it depends on the design of the basic AuxUEs, which is not the focus of this paper. For brevity, we simplify the input of AuxUE to the image $\mathbf{x}$. We detail the inputs for different experiments in Supplementary material (Supp) Section A.

3.1 Aleatoric uncertainty estimation on AuxUE

Based on the preliminaries of the settings, we now start with the first AuxUE σ_{Θ_1}, which addresses $\mathbf{u}_{\text{alea}}$ estimation problem as in SLURP and BayesCap.

We consider the data-dependent noise (Goldberg, Williams, and Bishop 1997; Bishop and Quazaz 1996; Nix and Weigend 1994) follows $\mathcal{N}(0, \boldsymbol{\sigma}^2)$. Then we use the DNN σ_{Θ_1} to estimate the heteroscedastic aleatoric uncertainty $\mathbf{u}_{\text{alea}}$ (Nix and Weigend 1994; Kendall and Gal 2017). $\widehat{\Theta}_1$ and the loss function $L(\Theta_1)$ are given by:

$$\widehat{\Theta}_1 = \underset{\Theta_1}{\operatorname{argmax}}\, P(\mathcal{D}|\hat{\boldsymbol{\omega}}, \Theta_1) = \underset{\Theta_1}{\operatorname{argmax}} \sum_{i=1}^{N} \log(P(y^{(i)}|\mathbf{x}^{(i)}, \hat{\boldsymbol{\omega}}, \Theta_1))$$

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\log(\sigma_{\Theta_1}(\mathbf{x}^{(i)})) + \frac{(y^{(i)} - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)}))^2}{2\sigma_{\Theta_1}(\mathbf{x}^{(i)})}\right] \quad (1)$$

The top of the σ_{Θ_1} is an exponential or Softplus function to maintain the output non-negative. The aleatoric uncertainty estimation will be: $\hat{u}^{(i)}_{\text{alea}} = \sigma_{\Theta_1}(\mathbf{x}^{(i)})$. Minimizing $\mathcal{L}(\Theta_1)$ is also equivalent to making σ_{Θ_1} correctly predict the main task errors on the training set according to likelihood maximization. The errors set is denoted as $\boldsymbol{\epsilon} = \{\epsilon^{(i)}\}_{i=1}^{N} = \{(y^{(i)} - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)}))^2\}_{i=1}^{N}$.

Given the fact that distribution assumption on the noise affecting $\hat{y}$ can be different than Gaussian, e.g., Laplacian (Marks et al. 1978) and Generalized Gaussian distribution (Nadarajah 2005; Upadhyay et al. 2022) also been considered in this work, the corresponding loss functions are provided in Supp Section B. The objective remains unchanged: employing AuxUE to estimate and predict the component associated with aleatoric uncertainty using various distribution assumptions. Perturbing input data in various ways with different types of noise makes it challenging to accurately identify the actual noise distribution. Relying on a single distribution assumption and loss function can affect the reliability of aleatoric uncertainty estimates. In Section 4.3, we assess the impact of different distribution assumptions and losses on the robustness of these estimates.

3.2 Epistemic uncertainty estimation on AuxUE

Modeling AuxUEs as formalized in Eq. 1 helps to estimate aleatoric uncertainty for $f_{\hat{\boldsymbol{\omega}}}$. Yet, taking this uncertainty prediction as an indicator for epistemic uncertainty is not methodologically grounded. Evidential learning is considered to be an effective uncertainty estimation approach (Ulmer 2021), which can capture epistemic uncertainty with a single pass as introduced in Section 2. We thus take it as an alternative to implement on AuxUE. In regression tasks, DNN estimates the parameters of the conjugate prior of Gaussian distribution, such as Normal Inverse Gamma (NIG) distribution (Amini et al. 2020). The training will make the model fall back onto a NIG prior for the rare samples by attaching lower evidence to the samples with higher prediction errors using a regularization term in the loss function (Amini et al. 2020). Yet, long-tailed prediction errors make standard AuxUE more inclined to give high evidence for most data points, thereby reducing its ability to estimate epistemic uncertainty. Our experiments also confirmed this tendency.

In contrast to previous works, which consider the *numerical value* of the prediction errors for both aleatoric and epistemic uncertainty estimation, we disentangle them and apply discretization to mitigate numerical bias from long-tailed prediction errors. Specifically, σ_{Θ_1} focuses on aleatoric uncertainty considering the *numerical value* of prediction errors, while for epistemic uncertainty, σ_{Θ_2} will consider the *value-free categories* of the prediction errors. Specifically, we propose Discretization-Induced Dirichlet pOsterior (DIDO), involves discretizing prediction errors and estimating a Dirichlet posterior based on the discrete errors. Further details are provided in the following sections.

3.2.1 Discretization on prediction errors To mitigate numerical bias due to imbalanced data in our prediction error estimation, we employ a balanced discretization approach. Discretization is widely applied in classification approaches for regression (Yu, Franchi, and Aldea 2022). The popular discretization methods can be generally divided into handcrafted (Cao, Wu, and Shen 2017) and adaptive (Bhat, Alhashim, and Wonka 2021). The latter requires computationally expensive components like mini-ViT (Dosovitskiy et al. 2021) to extract global features. Thus, we discretize prediction errors in a handcrafted way.

For pixel-wise scenarios, discretization is applied using per-image prediction errors, and for other cases, such as image-level tasks and 1D signal estimation, we use per-dataset prediction errors. Details and demo-code can be found in Supp Section C.1 and C.2 respectively.

We divide the set of errors $\boldsymbol{\epsilon}$, denoted in Section 3.1, into K subsets, where the kth subset is represented by the subscript k. To do this, we sort the errors in ascending order and create a new set, denoted by $\boldsymbol{\epsilon}'$, with the same elements as $\boldsymbol{\epsilon}$. Then we divide $\boldsymbol{\epsilon}'$ 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. Specifically, the one-hot vector is defined as:

$$\bar{\boldsymbol{\epsilon}}^{(i)} = [\bar{\epsilon}_1^{(i)} \dots \bar{\epsilon}_k^{(i)} \dots \bar{\epsilon}_K^{(i)}]^{\mathrm{T}} \in \mathbb{R}^K \tag{2}$$

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. This process creates a new dataset, denoted by $\bar{\mathcal{D}} = \{\mathbf{x}^{(i)}, \bar{\boldsymbol{\epsilon}}^{(i)}\}_i^N$, consisting of discretized prediction errors represented as one-hot vectors, which serves for training the epistemic uncertainty estimator σ_{Θ_2}.

3.2.2 Modeling epistemic uncertainty using $\boldsymbol{\epsilon}$ in auxiliary uncertainty estimation In a Bayesian framework, given an input $\mathbf{x}$, the predictive uncertainty of a DNN is modeled by $P(y|\mathbf{x}, \mathcal{D})$. Since we have a trained main task DNN, and as proposed in (Malinin and Gales 2018), we assume a point-estimate of $\boldsymbol{\omega}$ (denoted as $\hat{\boldsymbol{\omega}}$), then we have:

$$P(\boldsymbol{\omega}|\mathcal{D}) = \delta(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}) \rightarrow P(y|\mathbf{x}, \mathcal{D}) \approx P(y|\mathbf{x}, \hat{\boldsymbol{\omega}}) \tag{3}$$

with δ being the Dirac function.

We follow the previous assumption, i.e., the prediction is drawn from a Gaussian distribution $\mathcal{N}(y|\mu, \sigma^2)$ and according to (Amini et al. 2020), we denote $\boldsymbol{\alpha}$ as the parameters of the prior distributions of (μ, σ^2) and we have $P(\mu, \sigma^2|\boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) = P(\mu|\sigma^2, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}})P(\sigma^2|\boldsymbol{\alpha}, \boldsymbol{\omega}^*)$. After introducing $\boldsymbol{\alpha}$ and Eq. 3, we can approximate $P(y|\mathbf{x}, \mathcal{D})$ as:

$$\begin{aligned} P(y|\mathbf{x}, \mathcal{D}) &= \iint P(y|\mathbf{x}, \sigma^2)P(\sigma^2|\boldsymbol{\omega})P(\boldsymbol{\omega}|\mathcal{D})d\sigma^2 d\boldsymbol{\omega} \\ &= \int P(y|\mathbf{x}, \sigma^2)P(\sigma^2|\mathcal{D})d\sigma^2 \\ &\approx \int P(y|\mathbf{x}, \sigma^2)P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}})d\sigma^2 \end{aligned} \tag{4}$$

Detailed derivation can be found in Supp Section C.3.

We can consider ϵ to be drawn from a continuous distribution parameterized by σ^2. The discrepancy in variances $P(\sigma^2|\mathcal{D})$ can describe epistemic uncertainty of the final prediction and the variational approach can be applied (Joo, Chung, and Seo 2020; Malinin and Gales 2018): $P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \approx P(\sigma^2|\mathcal{D})$. After discretization, we can transform the approximation to $P(\boldsymbol{\pi}|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \approx P(\boldsymbol{\pi}|\bar{\mathcal{D}})$, with $\bar{\mathcal{D}}$ defined as in Section 3.2.1, $\boldsymbol{\pi}$ the parameters of a discrete distribution and $\boldsymbol{\alpha}$ re-defined as the prior distribution parameters of this discrete distribution. In the next section, we omit $\hat{\boldsymbol{\omega}}$ and $\mathbf{x}$ for the sake of brevity.

3.2.3 Dirichlet posterior for epistemic uncertainty According to the previous discussions on the epistemic uncertainty modeling and error discretization, we model Dirichlet posterior (Sensoy, Kaplan, and Kandemir 2018; Joo, Chung, and Seo 2020; Charpentier et al. 2022) on the discrete errors $\bar{\boldsymbol{\epsilon}}$ to achieve epistemic uncertainty on the main task.

Intuitively, we consider each one-hot prediction error $\bar{\boldsymbol{\epsilon}}^{(i)}$ to be drawn from a categorical distribution, and $\boldsymbol{\pi}^{(i)} = (\pi_1^{(i)}, \dots, \pi_K^{(i)})$ denotes the random variable over this distribution, where $\sum_{k=1}^{K} \pi_k^{(i)} = 1$ and $\pi_k^{(i)} \in [0, 1]$ for $k \in \{1, ..., K\}$. The conjugate prior of categorical distribution is a Dirichlet distribution:

$$P(\boldsymbol{\pi}^{(i)}|\boldsymbol{\alpha}^{(i)}) = \frac{\Gamma(S^{(i)})}{\prod_{k=1}^{K}\Gamma(\alpha_k^{(i)})} \prod_{k=1}^{K} {\pi_k^{(i)}}^{\alpha_k^{(i)}-1} \tag{5}$$

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.

To get access to the epistemic uncertainty, the categorical posterior $P(\boldsymbol{\pi}|\bar{\mathcal{D}})$ is needed, yet it is untractable. Approximating $P(\boldsymbol{\pi}|\bar{\mathcal{D}})$ using Monte-Carlo sampling (Gal and Ghahramani 2016) or ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) comes with an increased computational cost. Instead, we adopt a variational way to learn a Dirichlet distribution in Eq. 5 to approximate $P(\boldsymbol{\pi}|\bar{\mathcal{D}})$ as in (Joo, Chung, and Seo 2020). Here, σ_{Θ_2} outputs the concentration parameters $\boldsymbol{\alpha}$ of $P(\boldsymbol{\pi}|\boldsymbol{\alpha})$, and $\boldsymbol{\alpha}$ update according to the observed inputs. It can also be viewed as collecting the evidence $\boldsymbol{e}$ as a measure for supporting the classification decisions for each class (Sensoy, Kaplan, and Kandemir 2018), akin to estimating the Dirichlet posterior.

Since the numbers of data points are identical for each class in $\bar{\mathcal{D}}$, and no $\boldsymbol{e}^{(i)}$ output before training, we set the initial $\boldsymbol{\alpha}$ as $\mathbf{1}$ so that the Dirichlet concentration parameters can be formed as in (Sensoy, Kaplan, and Kandemir 2018; Charpentier, Zügner, and Günnemann 2020): $\boldsymbol{\alpha}^{(i)} = \boldsymbol{e}^{(i)} + \mathbf{1} = \sigma_{\Theta_2}(\mathbf{x}^{(i)}) + \mathbf{1}$, where $\boldsymbol{e}^{(i)}$ is given by an exponential function on the top of σ_{Θ_2}. Then we minimize the Kullback-Leibler (KL) divergence between the variational distribution $P(\boldsymbol{\pi}|\mathbf{x}, \Theta_2)$ and the true posterior $P(\boldsymbol{\pi}|\bar{\mathcal{D}})$ to achieve $\widehat{\Theta}_2$:

$$\begin{aligned} \widehat{\Theta}_2 &= \underset{\Theta_2}{\operatorname{argmin}}\ \mathrm{KL}[P(\boldsymbol{\pi}|\mathbf{x}, \Theta_2)||P(\boldsymbol{\pi}|\bar{\mathcal{D}})] \\ &= \underset{\Theta_2}{\operatorname{argmin}} - \mathbb{E}_{P(\boldsymbol{\pi}|\mathbf{x}, \Theta_2)}[\log P(\bar{\mathcal{D}}|\boldsymbol{\pi})] + \mathrm{KL}[P(\boldsymbol{\pi}|\mathbf{x}, \Theta_2)||P(\boldsymbol{\pi})] \end{aligned}$$

The loss function will be equivalent to minimizing the negative evidence lower bound (Jordan et al. 1999), considering

the prior distribution $P(\boldsymbol{\pi})$ as $\mathrm{Dir}(\mathbf{1})$:

$$\mathcal{L}(\Theta_2) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}[\bar{\epsilon}_k^{(i)}(\psi(S^{(i)}) - \psi(\alpha_k^{(i)}))] + \lambda \mathrm{KL}(\mathrm{Dir}(\boldsymbol{\alpha}^{(i)})||\mathrm{Dir}(\mathbf{1})) \quad (6)$$

where ψ is the digamma function, λ is a positive hyperparameter for the regularization term and $\bar{\epsilon}$ is given by Eq. 2.

For measuring epistemic uncertainty, we consider using the spread in the Dirichlet distribution (Shen et al. 2023; Charpentier, Zügner, and Günnemann 2020), which is shown in (Shen et al. 2023) to outperform other metrics, e.g. differential entropy. Specifically, the epistemic uncertainty is inversely proportional to the Dirichlet strength: $\hat{u}_{\text{epis}}^{(i)} = \sigma_{\widehat{\Theta}_2}(\mathbf{x}^{(i)}) = \frac{K}{S^{(i)}}$. The class corresponding to the maximum output from σ_{Θ_2} can also represent the aleatoric uncertainty. Yet, this is a rough estimate due to quantization errors and underperforming the other solutions. We provide the corresponding results in Supp Tab. A14. Overall, we take only σ_{Θ_1} output as the aleatoric uncertainty.

In conclusion, we propose a generalized AuxUE with two components, namely σ_{Θ_1} and σ_{Θ_2}, to quantify the uncertainty of the prediction given by the main task model. Based on different distribution assumptions on heteroscedastic noise in training data introduced in Section 3.1, we can train σ_{Θ_1} to estimate aleatoric uncertainty. Meanwhile, as described in Section 3.2, applying the proposed DIDO on σ_{Θ_2} and measuring the spread of Dirichlet distribution can help to estimate the epistemic uncertainty. Overall, we integrate the optimization for both uncertainty estimators, and the final loss for training the generalized AuxUE is:

$$\mathcal{L}_{\text{AuxUE}} = \mathcal{L}(\Theta_1) + \mathcal{L}(\Theta_2) \quad (7)$$

For $\mathcal{L}(\Theta_1)$, in addition to the Gaussian NLL, we will test other NLL loss functions according to different distribution assumptions in the experiment.

4 Experiments

In this section, we first show the feasibility of the proposed generalized AuxUE on toy examples. Then, we demonstrate the effectiveness of epistemic uncertainty estimation using the proposed DIDO on age estimation and monocular depth estimation (MDE) tasks, and investigate the robustness of aleatoric uncertainty estimation on MDE task. Due to page limitations, the experiments for an example of OOD detection in tabular data regression and the super-resolution task are provided in Supp Section A.2 and A.4 respectively.

In the result tables, the top two performing methods are highlighted in color. All the results are averaged by three runs. The shar.enc. and sep.enc. denote respectively shared-parameters for the encoders and separate encoders of σ_{Θ_1} and σ_{Θ_2} in the generalized AuxUE. For epistemic uncertainty, we compare our proposed method with the solutions based on modified main DNN: LDU (Franchi et al. 2022), Evidential learning (Evi.) (Amini et al. 2020; Joo, Chung, and Seo 2020) and Deep Ensembles (DEns.) (Lakshminarayanan, Pritzel, and Blundell 2017), as well as training-free methods: Gradient-based uncertainty (Grad.) (Hornauer and Belagiannis 2022), Variance based on Inject-Dropout (Inject.) (Mi et al. 2022).

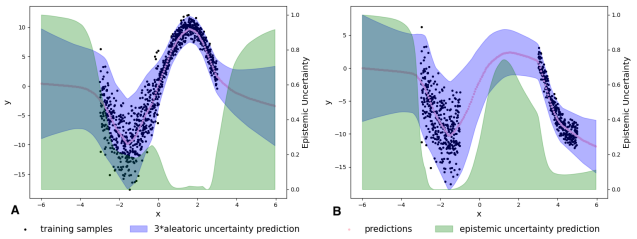


Figure 2: Results on 1D toy examples. Aleatoric and epistemic uncertainty estimations given by our proposed AuxUE are presented respectively as the uncertainty interval and degree (0-1).

The detailed implementations and the main task performance for all experiments are provided in Supp Section A.

4.1 Toy examples: Simple 1D regression

We generate two toy datasets to illustrate uncertainty estimates given by our proposed AuxUE, as shown in Fig. 2. In both examples, a tight aleatoric uncertainty estimation is provided on training data areas. For epistemic uncertainty, in Fig. 2-A, DIDO provides small uncertainty until reaching the unknown inputs $x \notin [-3, 3]$. In Fig. 2-B, we report the 'in-between' uncertainty estimates (Foong et al. 2019). On the in-between part $x \in [-1, 3]$, DIDO can provide higher epistemic uncertainty than in training set regions $x \in [-3, -1]$ and $x \in [3, 5]$. In summary, the generalized AuxUE provides reliable uncertainty estimates in regions where training data is either present or absent.

4.2 Age estimation and OOD detection

Epistemic uncertainty estimation for age estimation is similar to one for classification problems but has rarely been discussed in previous works. We use (unmodified) official ResNet34 (He et al. 2016) checkpoints from Coral (Cao, Mirjalili, and Raschka 2020) as the main task models. Our AuxUE is applied in a ConfidNet (Corbière et al. 2019) style since it is more suitable for image-level tasks.

Evaluation settings and datasets We train the models on AFAD (Niu et al. 2016) training set and choose AFAD test set as the ID dataset for the OOD detection task. We take CIFAR10 (Krizhevsky, Hinton et al. 2009), SVHN (Netzer et al. 2011), MNIST (LeCun 1998), FashionMNIST (Xiao, Rasul, and Vollgraf 2017), Oxford-Pets (Parkhi et al. 2012) and Noise image generated by Pytorch (Paszke et al. 2019) (FakeData) as the OOD datasets. We employ the Areas Under the receiver operating Characteristic (***AUC***) and the Precision-Recall curve (***AUPR***) (higher is better for both) to evaluate OOD detection performance.

Results OOD detection results are shown in Tab. 1. DIDO performs the best on most datasets. The training-free methods also perform well, but we observe that the Gradient-based solution needs inversed uncertainty (inv.) to provide better performance. On the Pets dataset, DIDO performs worse than DEns. and aleatoric uncertainty estimation head σ_{Θ_1}. We argue that images of pets provide features closer to facial information, resulting in higher evidence estimates given by DIDO. While σ_{Θ_1} performs better in this

OOD Dataset	Metrics	AuxUE		Modified main DNN			Training-free	
		Ours σ_{Θ_1}	Ours σ_{Θ_2} DIDO	LDU	Evi.	DEns.	Grad. (inv.)	Inject.
CIFAR10	AUC ↑	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	AUC ↑	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	AUC ↑	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	AUC ↑	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	AUC ↑	**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	AUC ↑	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 1: OOD detection results on Age estimation task. ID data is from Asian Face Age Dataset (AFAD) (Niu et al. 2016).

S	Metrics	Original	+ Ggau	+ Sgau	+ NIG	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	0.013
	AUSE-RMSE ↓	0.204	0.258	**0.202**	0.208	0.205	0.203
	AURG-REL ↑	0.023	0.023	0.023	**0.024**	0.023	0.023
	AURG-RMSE ↑	1.869	1.815	**1.871**	1.865	1.869	1.870
1	AUSE-REL ↓	0.019	0.021	0.019	**0.018**	**0.018**	0.019
	AUSE-RMSE ↓	0.340	0.482	**0.332**	0.335	**0.332**	0.336
	AURG-REL ↑	0.031	0.029	0.031	**0.032**	**0.032**	0.031
	AURG-RMSE ↑	2.357	2.215	**2.365**	2.362	**2.365**	2.361
2	AUSE-REL ↓	0.024	0.026	0.023	**0.022**	**0.022**	0.023
	AUSE-RMSE ↓	0.483	0.707	**0.463**	0.479	0.464	0.468
	AURG-REL ↑	0.038	0.035	**0.039**	**0.039**	**0.039**	0.038
	AURG-RMSE ↑	2.759	2.535	**2.779**	2.763	2.777	2.774
3	AUSE-REL ↓	0.033	0.036	**0.031**	**0.031**	**0.031**	**0.031**
	AUSE-RMSE ↓	0.795	1.176	0.737	0.806	0.749	**0.730**
	AURG-REL ↑	0.047	0.044	**0.049**	**0.049**	**0.049**	**0.049**
	AURG-RMSE ↑	3.243	2.862	3.301	3.232	3.289	**3.308**
4	AUSE-REL ↓	0.056	0.057	0.050	0.053	0.051	**0.049**
	AUSE-RMSE ↓	1.517	2.380	1.364	1.582	1.430	**1.268**
	AURG-REL ↑	0.051	0.051	0.058	0.054	0.056	**0.059**
	AURG-RMSE ↑	3.680	2.817	3.834	3.615	3.767	**3.929**
5	AUSE-REL ↓	0.071	0.082	0.064	0.069	0.066	**0.059**
	AUSE-RMSE ↓	2.202	3.878	2.043	2.414	2.157	**1.760**
	AURG-REL ↑	0.056	0.045	0.063	0.057	0.061	**0.067**
	AURG-RMSE ↑	4.054	2.377	4.213	3.842	4.098	**4.496**

Table 2: Aleatoric uncertainty estimation results on Monocular Depth Estimation. $S = 0$ represents original KITTI dataset and $S > 0$ represents KITTI-C datasets.

case, which can jointly make AuxUE a better uncertainty estimator. Overall, we consider that using generalized AuxUE with DIDO is an alternative that can better detect OOD inputs than ensembling-based solutions.

4.3 Monocular depth estimation task

For the MDE task, we will evaluate both aleatoric and epistemic uncertainty estimation performance based on the AuxUE SLURP (Yu, Franchi, and Aldea 2021). Our generalized AuxUE is also constructed using SLURP as the backbone. We use BTS (Lee et al. 2019) as the main task model and KITTI (Geiger et al. 2013; Uhrig et al. 2017) Eigen-split (Eigen, Puhrsch, and Fergus 2014) training set for training both BTS and AuxUE models.

4.3.1 Aleatoric uncertainty estimation In this section, the goal is to analyze the fundamental performance and robustness of aleatoric uncertainty estimation under different distribution assumptions. We choose simple Gaussian (Sgau) (Nix and Weigend 1994), Laplacian (Lap), Generalized Gaussian (Ggau) (Upadhyay et al. 2022) and Normal-Inverse-Gamma (NIG) (Amini et al. 2020) distributions. We modify the loss functions and the head of the SLURP to output the desired parameters of the distributions.

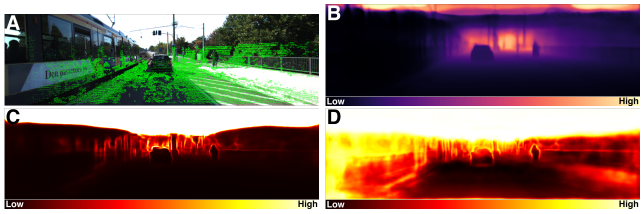


Figure 3: Illustrations of uncertainty estimations for MDE task. A: input image, green points represent pixels with depth groundtruth; B: depth prediction; C and D: aleatoric and epistemic uncertainty estimations. The areas lacking depth groundtruth, e.g. sky and tramway, are assigned high uncertainty using DIDO.

Evaluation settings and datasets We first build Sparsification curves (SC) (Bruhn and Weickert 2006): we achieve predictive SC by computing the prediction error of the remaining pixels after removing a certain partition of pixels (5% in our experiment) each time according to the highest uncertainty estimations. We can also obtain an Oracle SC by removing the pixels according to the highest prediction errors. Then, we have the same metrics used in (Poggi et al. 2020): Area Under the Sparsification Error (***AUSE***, lower is better), and Area Under the Random Gain (***AURG***, higher is better). We choose absolute relative error (REL) and root mean square error (RMSE) as the prediction error metrics.

We generate KITTI-C from KITTI Eigen-split validation set using the code of ImageNet-C (Hendrycks and Dietterich 2019) to have different corruptions on the images to check the robustness of the uncertainty estimation solutions. We apply eighteen perturbations with five severities, including Gaussian noise, shot noise, etc., and take it along with the original KITTI for evaluation.

Results As shown in Tab. 2, the Laplace assumption is more robust when the severity increases, while Gaussian one works better when the noise severity is smaller. We also check the proposed generalized AuxUE with a shared encoder. It shows that the epistemic uncertainty estimation branch affects the robustness of aleatoric uncertainty estimation in this case, especially under stronger noise.

The next sections show epistemic uncertainty estimation results based on different methods. Furthermore, in Supp Tab. A15 and Tab. A16, we also verify whether aleatoric uncertainty methods based on different distribution assumptions can generalize to the OOD data, i.e., provide high uncertainty to the unseen patterns, even without explicitly modeling epistemic uncertainty.

4.3.2 Robustness under dataset change This experiment will explore the predictive uncertainty performance encountering the dataset change. Supervised MDE is an ill-posed problem that heavily depends on the training dataset. In our case, the main task model is trained on the KITTI dataset, so the model will output meaningless results on the indoor data, which should trigger a high uncertainty estimation. The

Metrics	AuxUE with DIDO		Modified main DNN			Training-free	
	Ours σ_{Θ_2} sep. enc.	Ours σ_{Θ_2} shar. enc.	LDU	Evi.	DEns.	Grad.	Inject.
AUC ↑	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 3: Epistemic uncertainty estimation results encountering dataset change on Monocular depth estimation task. The evaluation dataset used here is NYU indoor depth dataset.

S	Metrics	AuxUE with DIDO		Modified main DNN			Training-free	
		Ours σ_{Θ_2} sep. enc.	Ours σ_{Θ_2} shar. enc.	LDU	Evi.	DEns.	Grad. (inv.)	Inject.
0	AUC ↑	**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	AUC ↑	**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	AUC ↑	**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	AUC ↑	**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	AUC ↑	**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	AUC ↑	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 4: 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).

results are shown in Tab. 3.

Evaluation settings and datasets We take ***AUC*** and ***AUPR*** as evaluation metrics. We take all the valid pixels from the KITTI validation set (ID) as the negative samples and the valid pixels from the NYU (Nathan Silberman and Fergus 2012) validation set (OOD) as the positive samples.

Results Tab. 3 shows whether different uncertainty estimators can give correct indications facing the dataset change. Generalized Gaussian and Gradient-based methods can provide competitive results, while our method, especially DIDO, provides the best performance.

4.3.3 Robustness on unseen patterns during training

This experiment focuses on how uncertainty estimators behave on unseen patterns during training. The unseen patterns are drawn from the same dataset distribution as the patterns used in training, and the outputs of the main task model for such patterns may be reasonable. Still, they cannot be evaluated and thus are unreliable. High uncertainty should be assigned to these predictions. Since this topic is rarely considered in MDE, we try to give a benchmark in this work.

Evaluation settings and datasets We select sky areas in KITTI as OOD patterns. This setting is based on the following reasons: due to the generalization ability of MDE DNNs, it is inappropriate to treat all pixels without ground truth as OOD. However, there is consistently no ground truth for the sky parts since LIDAR is used in depth acquisition. During training, sky patterns are masked and never seen by the DNNs (including the AuxUEs). Meanwhile, they are annotated in KITTI semantic segmentation dataset (Alhaija et al. 2018) (200 images), thus can be used for evaluation.

Three metrics are applied for evaluating OOD detection performance as shown in Tab. 4. ***AUC*** and ***AUPR***: we select 49 images that are not in the training set and have both depth and semantic segmentation annotations. For each image, we take the sky pixels as the positive class and the pixels with depth ground truth as the negative class. We use AUC and AUPR to assess the uncertainty estimation performance. Note that this metric does not guarantee that the uncertainty of the sky is the largest in the whole uncertainty map. Thus, we have ***Sky-All*** (lower is better): all 200 images with semantic segmentation annotations are selected for evaluation. The ground truth uncertainties are set as $\mathbf{1}$ for the sky areas. Then we normalize the predicted uncertainty, take the sky areas $\hat{\mathbf{u}}_{\text{sky}}$ from the whole uncertainty map and measure: $mean((\mathbf{1} - \hat{\mathbf{u}}_{\text{sky}})^2)$. For simplicity, we denote KITTI Seg-Depth for both evaluation datasets. We also generate a corruption dataset KITTI Seg-Depth-C using the same way in the aleatoric uncertainty estimation section.

Results Fig. 3 shows a qualitative example of typical uncertainty maps computed on KITTI images. More visualizations are presented in Supp Section E. In Tab. 4, the Deep Ensembles and Gradient-based methods can better assign consistent and higher uncertainty to the sky areas, but they are inadequate for identifying the ID and OOD areas. As outlined in Section 3.2.3, DIDO prioritizes rare patterns and then generalizes the uncertainty estimation ability to the unseen patterns. This results in assigning higher uncertainty to some few-shot pixels that have ground truth, making Sky-All results slightly worse. Yet, it can achieve a balanced performance on all the metrics, and at the same time, it maintains robust performance in the presence of noise.

4.4 Ablation study

We conduct the ablation study on the corresponding section in Supp Section D. **Hyperparameters.** We analyze the effect of the number of sets K defined in Section 3.2 for discretization and λ for the regularization term in Eq. 6. **Necessity of using AuxUE.** We also apply DIDO on the main task model to check the impact on main task performance. **Effectiveness of Dirichlet modeling.** We show the effectiveness of the Dirichlet modeling instead of using the normal Categorical modeling based on the discretized prediction errors. For the former, we apply classical cross-entropy on the Softmax outputs given by the AuxUE.

5 Conclusion

In this paper, we propose a new solution for uncertainty quantification on regression problems based on a generalized AuxUE. We design and implement the experiments based on four different regression problems. By modeling heteroscedastic noise using Laplace distribution, the proposed AuxUE can achieve more robust aleatoric uncertainty. Meanwhile, the novel DIDO solution in our AuxUE can provide better epistemic uncertainty estimation performance on both image-level and pixel-wise tasks.

Acknowledgements

We acknowledge the support of the Saclay-IA computing platform. We also thank Mădălina Olteanu for the thought-provoking discussion for the article.

References

Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U. R.; et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76: 243–297.

Alhaija, H.; Mustikovela, S.; Mescheder, L.; Geiger, A.; and Rother, C. 2018. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *International Journal of Computer Vision (IJCV)*.

Amini, A.; Schwarting, W.; Soleimany, A.; and Rus, D. 2020. Deep evidential regression. *NeurIPS*.

Arnez, F.; Espinoza, H.; Radermacher, A.; and Terrier, F. 2020. A comparison of uncertainty estimation approaches in deep learning components for autonomous vehicle applications. *arXiv preprint arXiv:2006.15172*.

Besnier, V.; Bursuc, A.; Picard, D.; and Briot, A. 2021. Triggering Failures: Out-Of-Distribution detection by learning from local adversarial attacks in Semantic Segmentation. In *ICCV*.

Bevilacqua, M.; Roumy, A.; Guillemot, C.; and Alberi-Morel, M. L. 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. *BMVC*.

Bhat, S. F.; Alhashim, I.; and Wonka, P. 2021. Adabins: Depth estimation using adaptive bins. In *CVPR*.

Bishop, C.; and Quazaz, C. 1996. Regression with input-dependent noise: A Bayesian treatment. *NeurIPS*.

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *ICML*.

Bruhn, A.; and Weickert, J. 2006. A confidence measure for variational optic flow methods. *Computational Imaging and Vision*, 31: 283.

Cao, W.; Mirjalili, V.; and Raschka, S. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140: 325–331.

Cao, Y.; Wu, Z.; and Shen, C. 2017. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11): 3174–3182.

Castillo, A.; Escobar, M.; Pérez, J. C.; Romero, A.; Timofte, R.; Van Gool, L.; and Arbelaez, P. 2021. Generalized real-world super-resolution through adversarial robustness. In *ICCV*.

Charpentier, B.; Borchert, O.; Zügner, D.; Geisler, S.; and Günnemann, S. 2022. Natural Posterior Network: Deep Bayesian Predictive Uncertainty for Exponential Family Distributions. In *ICLR*.

Charpentier, B.; Zügner, D.; and Günnemann, S. 2020. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *NeurIPS*.

Corbière, C.; Lafon, M.; Thome, N.; Cord, M.; and Pérez, P. 2021. Beyond First-Order Uncertainty Estimation with Evidential Models for Open-World Recognition. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*.

Corbière, C.; Thome, N.; Bar-Hen, A.; Cord, M.; and Pérez, P. 2019. Addressing failure prediction by learning model confidence. In *NeurIPS*.

Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*.

Dempster, A. P. 1968. A generalization of Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2): 205–232.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Eigen, D.; Puhrsch, C.; and Fergus, R. 2014. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*.

Foong, A. Y.; Li, Y.; Hernández-Lobato, J. M.; and Turner, R. E. 2019. 'In-Between'Uncertainty in Bayesian Neural Networks. *arXiv preprint arXiv:1906.11537*.

Franchi, G.; Yu, X.; Bursuc, A.; Aldea, E.; Dubuisson, S.; and Filliat, D. 2022. Latent Discriminant deterministic Uncertainty. In *ECCV*.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*.

Goldberg, P.; Williams, C.; and Bishop, C. 1997. Regression with input-dependent noise: A Gaussian process treatment. *NeurIPS*.

Guo, Z.; Wan, Z.; Zhang, Q.; Zhao, X.; Chen, F.; Cho, J.-H.; Zhang, Q.; Kaplan, L. M.; Jeong, D. H.; and Jøsang, A. 2022. A Survey on Uncertainty Reasoning and Quantification for Decision Making: Belief Theory Meets Deep Learning. *arXiv preprint arXiv:2206.05675*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *ICLR*.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *ICLR*.

Hornauer, J.; and Belagiannis, V. 2022. Gradient-Based Uncertainty for Monocular Depth Estimation. In *ECCV*.

Hüllermeier, E.; and Waegeman, W. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110: 457–506.

Jain, M.; Lahlou, S.; Nekoei, H.; Butoi, V.; Bertin, P.; Rector-Brooks, J.; Korablyov, M.; and Bengio, Y. 2021. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*.

Joo, T.; Chung, U.; and Seo, M.-G. 2020. Being bayesian about categorical probability. In *ICML*.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine learning*, 37: 183–233.

Kamann, C.; and Rother, C. 2021. Benchmarking the robustness of semantic segmentation models with respect to common corruptions. *IJCV*.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*.

Laves, M.-H.; Ihler, S.; Kortmann, K.-P.; and Ortmaier, T. 2020. Calibration of model uncertainty for dropout variational inference. *arXiv preprint arXiv:2006.11584*.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.

Lee, J. H.; Han, M.-K.; Ko, D. W.; and Suh, I. H. 2019. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *ICLR*.

Malinin, A.; Chervontsev, S.; Provilkov, I.; and Gales, M. 2020. Regression prior networks. *arXiv preprint arXiv:2006.11590*.

Malinin, A.; and Gales, M. 2018. Predictive uncertainty estimation via prior networks. *NeurIPS*.

Malinin, A.; and Gales, M. 2019. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *NeurIPS*.

Marks, R. J.; Wise, G. L.; Haldeman, D. G.; and Whited, J. L. 1978. Detection in Laplace noise. *IEEE Transactions on Aerospace and Electronic Systems*, 866–872.

Martin, D.; Fowlkes, C.; Tal, D.; and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*.

Mi, L.; Wang, H.; Tian, Y.; and Shavit, N. 2022. Training-Free Uncertainty Estimation for Dense Regression: Sensitivity as a Surrogate. In *AAAI*.

Michaelis, C.; Mitzkus, B.; Geirhos, R.; Rusak, E.; Bringmann, O.; Ecker, A. S.; Bethge, M.; and Brendel, W. 2019. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*.

Nadarajah, S. 2005. A generalized normal distribution. *Journal of Applied statistics*, 32(7): 685–694.

Nathan Silberman, P. K., Derek Hoiem; and Fergus, R. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NeurIPS*.

Niu, Z.; Zhou, M.; Wang, L.; Gao, X.; and Hua, G. 2016. Ordinal Regression With Multiple Output CNN for Age Estimation. In *CVPR*.

Nix, D.; and Weigend, A. 1994. Estimating the mean and variance of the target probability distribution. In *ICNN*.

Oh, D.; and Shin, B. 2022. Improving evidential deep learning via multi-task learning. In *AAAI*.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. V. 2012. Cats and Dogs. In *CVPR*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*.

Poggi, M.; Aleotti, F.; Tosi, F.; and Mattoccia, S. 2020. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*.

Qu, H.; Li, Y.; Foo, L. G.; Kuen, J.; Gu, J.; and Liu, J. 2022. Improving the reliability for confidence estimation. In *ECCV*.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *ICML*.

Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential deep learning to quantify classification uncertainty. *NeurIPS*.

Shannon, C. E. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1): 3–55.

Shen, M.; Bu, Y.; Sattigeri, P.; Ghosh, S.; Das, S.; and Wornell, G. 2023. Post-hoc Uncertainty Learning using a Dirichlet Meta-Model. In *AAAI*.

Techapanurak, E.; and Okatani, T. 2021. Practical evaluation of out-of-distribution detection methods for image classification. *arXiv preprint arXiv:2101.02447*.

Uhrig, J.; Schneider, N.; Schneider, L.; Franke, U.; Brox, T.; and Geiger, A. 2017. Sparsity Invariant CNNs. In *3DV*.

Ulmer, D. 2021. A survey on evidential deep learning for single-pass uncertainty estimation. *arXiv preprint arXiv:2110.03051*.

Upadhyay, U.; Karthik, S.; Chen, Y.; Mancini, M.; and Akata, Z. 2022. BayesCap: Bayesian Identity Cap for Calibrated Uncertainty in Frozen Neural Networks. In *ECCV*.

Wen, Y.; Tran, D.; and Ba, J. 2020. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *ICLR*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yeo, T.; Kar, O. F.; and Zamir, A. 2021. Robustness via cross-domain ensembles. In *CVPR*.

Yu, X.; Franchi, G.; and Aldea, E. 2021. SLURP: Side Learning Uncertainty for Regression Problems. In *BMVC*.

Yu, X.; Franchi, G.; and Aldea, E. 2022. On Monocular Depth Estimation and Uncertainty Quantification Using Classification Approaches for Regression. In *ICIP*.

Zeyde, R.; Elad, M.; and Protter, M. 2012. On single image scale-up using sparse-representations. In *ICCS*.

Discretization-Induced Dirichlet Posterior for Robust Uncertainty Quantification on Regression

——— Supplementary Material ———

Hyperparameters	Main task	AuxUE
learning rate	0.001	0.005
# epochs	200	100
batch size	64	64
λ	-	0.001
K	-	5

Table A5: Hyperparameters for 1D signal toy examples.

Hyperparameters	Main task	AuxUE
learning rate	0.001	0.001
# epochs	150	20
batch size	64	64
λ	-	0.0001
K	-	5

Table A6: Hyperparameters for tabular data example.

	MSE ↓	AUC ↑	AUPR ↑
DEns.	**0.646**	0.548	0.250
DIDO	**0.646**	**0.936**	**0.863**

Table A7: Main task and OOD detection performance on tabular data example.

A Supplements for the experiments

A.1 Toy example: 1D signal

A.1.1 Dataset We created two toy datasets for our experiment. Fig.2-A on the main paper was generated as follows: $y = 10\sin(x) + \epsilon$, with ϵ:

$$\begin{cases} \epsilon \sim \mathcal{N}(0,3) & x \in [-3,0] \\ \epsilon \sim \mathcal{N}(0,1) & x \in [0,3] \\ 0 & \text{otherwise} \end{cases}$$

Fig.2-B on the main paper was generated as follows: $y = 10\sin(x) + \epsilon$, with ϵ:

$$\begin{cases} \epsilon \sim \mathcal{N}(0,3) & x \in [-3,-1] \\ \epsilon \sim \mathcal{N}(0,1) & x \in [3,5] \\ 0 & \text{otherwise} \end{cases}$$

A.1.2 Models Our main task model consists of an MLP with four hidden layers with 300 hidden units per layer and ReLU non-linearities. We use a generalized AuxUE method similar to ConfidNet (Corbière et al. 2019).

In particular, the input of the AuxUE is the features from the output of the penultimate layer of the main task model. Thus, in this case, the generalized AuxUE does not need the encoders, as shown in the general process in Fig.1 on the main paper. The architecture of this AuxUE is as follows. σ_{Θ_1} is composed of one fully connected layer (FCL) with an exponential activation function on the top. σ_{Θ_2} is composed of an MLP with a cosine similarity layer and a hidden layer with 300 hidden units per layer, and an exponential activation function on the top.

The reason for using the cosine similarity layer is to decrease the impact of the numerical value. This operation is similar to the fully connected layer operation but simply divides the output by the product of the norm of the layer's inputs (trainable parameters of the linear and input features).

A.1.3 Training The hyperparameters are listed in Tab. A5. As a reminder, λ and K are the hyperparameters specifically for AuxUE (σ_{Θ_2}), which stand for the weight for the regularization term in the loss function, and respectively for the number of the class we set for discretization.

A.2 Tabular data example

A.2.1 Dataset In the tabular data example, we use the red wine quality dataset (Cortez et al. 2009) for the OOD detection task. We randomly separate the dataset in training, validation, and test sets with 72%, 8%, and 20% as the proportions of the whole dataset for each set. We generate the OOD data using the ID test set. We first replicate two test sets as OOD sets, one of which we set all the features in the table to be negative, and the other, we randomly shuffle the values of the features.

A.2.2 Models and training Our main task model consists of an MLP with four hidden layers with 16, 32, and 16 hidden units in the respective layer and ReLU non-linearities. We use a generalized AuxUE method similar to ConfidNet (Corbière et al. 2019).

In particular, we find it better to provide the tabular data to the AuxUE directly. We use one hidden layer with 16 hidden units followed by ReLU as the feature extractor for σ_{Θ_1} and σ_{Θ_2} uncertainty estimators. For the uncertainty estimators, we use the same ones as in the 1D signal data. The hyperparameters are listed in Tab. A6.

A.2.3 Results We trained three models to build Deep Ensembles (DEns.) (Lakshminarayanan, Pritzel, and Blundell 2017). The epistemic uncertainty estimates are obtained using the variance of DNNs' point estimates. We evaluate the OOD detection performance using AUC and AUPR as the metrics. The results are shown in Tab. A7. The proposed DIDO outperforms the DEns. on OOD detection task using one extra DNN apart from the main task model.

A.3 Age estimation

A.3.1 Model The main task ResNet34 (He et al. 2016) model checkpoints are downloaded from the official GitHub repository of Coral (Cao, Mirjalili, and Raschka 2020).

Hyperparameters	Main task	AuxUE
learning rate	0.0005	0.001
# epochs	200	25
batch size	256	256
λ	-	0.01
K	-	8

Table A8: Hyperparameters for age estimation.

Metrics	Coral	CE	CE +SWS	LDU	Evi.	DEns.
MAE ↓	3.47 ± 0.05	3.60 ± 0.02	3.39	3.41	3.70 ± 0.19	**3.31**
RMSE ↓	4.71 ± 0.06	5.03 ± 0.03	4.52 ± 0.03	4.50	4.72 ± 0.23	**4.40**

Table A9: Main task performance for ResNet34 model based on different methods on age estimation task. The evaluation is based on AFAD (Niu et al. 2016) test set.

We observe that the age estimation result can outperform the one achieved by Coral by applying soft-weighted-sum (SWS) (Yu, Franchi, and Aldea 2022) on the top of the models trained using cross-entropy loss. The goal of SWS is a post-processing operation to transfer the discrete Softmax outputs to the continuous age estimates. For this reason, we use the main task models trained by cross-entropy loss.

As introduced in Section 4.2 of the main paper, the AuxUE DNN is applied in a ConfidNet (Corbière et al. 2019) way. Similarly to the toy example settings, we take the pre-logits (512 features) from the main task model as the inputs of our AuxUE.

For σ_{Θ_1}, we use an MLP with one hidden layer with 512 hidden units and an FCL with an exponential function on the top. For σ_{Θ_2}, we use an MLP with a cosine similarity layer and one hidden layer with 512 hidden units per layer and ReLU non-linearities, followed by an FCL with an exponential function on the top.

A.3.2 Training To train the AuxUE DNN, we use the hyperparameters shown in Tab. A8. We use the same optimizer and batch size as for the main task training, while we use 25 epochs which is much less than training the main task.

A.3.3 Main task and aleatoric uncertainty performance For the age estimation task, we list the main task results in Tab. A9 given by the original Coral, the original cross entropy (CE)-based models and the CE-based models using soft-weighted-sum (SWS). We can see that SWS really improves the main task performance. Furthermore, by adjusting the original model to output the parameters of Gaussian distribution (Nix and Weigend 1994; Kendall and Gal 2017) and training three models like this from scratch, we can achieve the results given by Deep Ensembles (DEns.) (Lakshminarayanan, Pritzel, and Blundell 2017). We also implement LDU (Franchi et al. 2022) and Evidential learning (Evi.) (Joo, Chung, and Seo 2020) based on the ResNet34 backbone. The overall difference among different techniques is not huge, while the adjustments still reduce a bit the age estimation performance. We argue that the adjusted DNNs might achieve comparable performance to the unchanged ones, but more tuning and hyperparameter searching should be required. On the other hand, for aleatoric uncertainty estimation result, for AUSE-RMSE (↓), Ours: 0.067, LDU: 0.056, Evi.: 0.070, DEns.: 0.074, Grad.: 0.073, Inject.: 0.076. Ours is shown to provide comparable results to the other solutions.

A.4 Super-resolution

In the SR task, the noise in the reconstructed image will be irreducible given the noisy low-resolution input, and we consider this uncertainty to be aleatoric. Moreover, we argue that the definition of epistemic uncertainty is rather vague in this task. Therefore, in this section, we use AuxUE to estimate the aleatoric uncertainty based on different distribution assumptions.

A.4.1 Model Similar to the monocular depth estimation experiments in Section 4.3.1 in the main paper, we choose SRGan (Ledig et al. 2017) as the main task model and BayesCap (Upadhyay et al. 2022) as the AuxUE and follow the same training and evaluation settings as in (Upadhyay et al. 2022). The goal is to analyze the fundamental performance and robustness of aleatoric uncertainty estimation under different distribution assumptions. We choose simple Gaussian (Sgau) (Nix and Weigend 1994), Laplacian (Lap), Generalized Gaussian (Ggau) (Upadhyay et al. 2022) and Normal-Inverse-Gamma (NIG) (Amini et al. 2020) distributions on BayesCap (Upadhyay et al. 2022).

We modify the loss functions to output the desired parameters of the distributions. For the architecture adjustments, we only modify the prediction heads on Bayescap. Original BayesCap (Upadhyay et al. 2022) uses multiple Residual blocks (He et al. 2016) followed by three heads which output the three parameters for the Generalized Gaussian distribution, including one as the refined main task prediction. Each head contains a set of convolutional layers + PReLU activation functions. As we apply different distribution assumptions, we use the different numbers of the same heads to construct the variants of BayesCap. Specifically, we use two heads for two Gaussian distribution parameters, two heads for two Laplace distribution parameters, and four heads for four parameters in NIG distribution.

A.4.2 Training We follow the same training settings (batch size, learning rate, weight for the additional identity mapping loss, and the number of epochs) as in the original paper (Upadhyay et al. 2022).

A.4.3 Evaluation settings and datasets We follow (Upadhyay et al. 2022) to use the Uncertainty Calibration Error (***UCE***, lower is better) metric (Laves et al. 2020). It measures the difference between the predicted uncertainty and the prediction error. Specifically, the prediction error and estimated uncertainty are assigned into bins, and the absolute difference between the mean prediction error and mean estimated uncertainty in each bin is calculated. UCE is the sum of the results from all bins.

We use ImageNet (Deng et al. 2009) as the training set for both SRGan and BayesCap models. For uncertainty evaluation, we use Set5 (Bevilacqua et al. 2012), Set14 (Zeyde, Elad, and Protter 2012), and BSDS100 (Martin et al. 2001) as the testing sets. Moreover, we generate Set5-C, Set14-C,

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 A10: 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.

Metrics	Set5	Set14	BSDS100
PSNR ↑	29.40	26.02	25.16
SSIM ↑	0.8472	0.7397	0.6688

Table A11: Main task performance for SRGan model on super-resolution task.

and BSDS100-C using the code of ImageNet-C (Hendrycks and Dietterich 2019) to have different corruptions on the images. We apply the following eighteen perturbations with five severities: Gaussian noise, shot noise, impulse noise, iso noise, defocus blur, glass blur, motion blur, zoom blur, frost, fog, snow, dark, brightness, contrast, pixelated, elastic, color quantization, and JPEG. In the main paper, we mentioned these perturbations in Section 4.3, yet due to the paper limitation, we put the complete list here. Only low-resolution images (inputs) are polluted by noise, while the corresponding high-resolution ground truth images are clean. Castillo et al. (Castillo et al. 2021) applied the noise to the input images during training, while we apply them during inference for robust uncertainty estimation evaluation.

A.4.4 Results As shown in Tab. A10, the Laplacian assumption on the data-dependent noise performs better than all the other assumptions, including the Generalized Gaussian distribution proposed in BayesCap. When the noise severity increases, using the Laplacian assumption can provide more robust uncertainty than the others.

A.4.5 Main task performance In the super-resolution task, we take the main task SRGan (Ledig et al. 2017) model used in BayesCap (Upadhyay et al. 2022). Thus we have the same main task performance as they showed in the paper. We list the results in Tab. A11 as a reminder. The evaluation is based on Set5 (Bevilacqua et al. 2012), Set14 (Zeyde, Elad, and Protter 2012), BSDS100 (Martin et al. 2001) dataset.

Hyperparameters	Main task	AuxUE
start learning rate	1e-4	1e-4
end learning rate	1e-5	1e-5
# epochs	50	8
batch size	4	4
λ	-	0.01
K	-	32

Table A12: Hyperparameters for monocular depth estimation.

A.5 Monocular depth estimation

A.5.1 Model We use SLURP (Yu, Franchi, and Aldea 2021) as the backbone in this experiment. We modify the prediction heads to achieve the uncertainty estimates.

For σ_{Θ_1}, we do not modify the model when the distribution assumption only contains one parameter (except for the main task prediction term). For the distribution assumptions with more than one parameter output, we add one more convolutional layer with ReLU on the top for a fair comparison.

For σ_{Θ_2}, similarly to the one in age estimation, we replace the original head (a single convolutional layer) with the cosine similarity layer followed by two convolutional layers with ReLU activation functions. In the cases where we share the encoders to make the general AuxUE lighter, based on the original SLURP, we doubled the number of features fed into the prediction head. We split them into two sets, feeding them into two prediction heads. The two prediction heads are consistent in the structures mentioned before. We follow (Yu, Franchi, and Aldea 2021) to use the depth output and the encoder features of the main task model BTS (Lee et al. 2019) as the input of the AuxUE.

A.5.2 Training The hyperparameters used during training are listed in Tab. A12. The learning rate decrement is consistent with the main task BTS (Lee et al. 2019) model.

A.5.3 Main task performance In monocular depth estimation, we list in Tab. A13 the results for the methods using modified main task BTS (Lee et al. 2019) models, namely SinglePU (Kendall and Gal 2017), Deep Ensembles (DEns.) (Lakshminarayanan, Pritzel, and Blundell 2017), LDU (Franchi et al. 2022), as well as the original model, which is used for AuxUEs and the training-free methods. Note that we use the evaluation code based on AdaBins (Bhat, Alhashim, and Wonka 2021), which corrected the error made in the BTS evaluation code, and the result will be slightly better than the one claimed in the original BTS. The evaluation is based on KITTI (Geiger et al. 2013) Eigen-split (Eigen, Puhrsch, and Fergus 2014) validation set. As we can see, modifying the model and training in (Kendall and Gal 2017) way will affect the main task performance even after doing Deep Ensembles, LDU can provide competitive results to the original yet only on several metrics. Overall, the AuxUE is necessary to be applied for uncertainty estimation without changing and affecting the main task.

A.5.4 Additional results on aleatoric uncertainty estimation from Dirichlet outputs The class corresponds to the maximum output value of the Dirichlet output, i.e., outputs

Methods	absrel ↓	log10 ↓	rms ↓	sqrel ↓	logrms ↓	d1 ↑	d2 ↑	d3 ↑
Org	**0.056**	**0.025**	2.430	**0.201**	**0.089**	**0.963**	0.994	**0.999**
SinglePU	0.065	0.029	2.606	0.234	0.100	0.952	0.993	0.998
LDU	0.059	0.026	**2.394**	0.203	0.091	0.960	0.994	**0.999**
DEns.	0.060	0.026	2.435	0.202	0.092	0.961	**0.995**	**0.999**

Table A13: Main task performance for original and modified BTS models on monocular depth estimation. The evaluation is based on KITTI (Geiger et al. 2013) Eigen-split (Eigen, Puhrsch, and Fergus 2014) validation set.

S	Metrics	+ Sgau	+ NIG	Ours (+ Lap) shar. enc. σ_{Θ_1}	Ours (+ Lap) sep. enc. σ_{Θ_1}	Ours (DIDO) sep. enc. σ_{Θ_2}
0	AUSE-REL ↓	0.013	**0.012**	0.013	0.013	0.015
	AUSE-RMSE ↓	**0.202**	0.208	0.205	0.203	0.283
	AURG-REL ↑	0.023	**0.024**	0.023	0.023	0.021
	AURG-RMSE ↑	**1.871**	1.865	1.869	1.870	1.791
1	AUSE-REL ↓	0.019	**0.018**	**0.018**	0.019	0.023
	AUSE-RMSE ↓	**0.332**	0.335	**0.332**	0.336	0.474
	AURG-REL ↑	0.031	**0.032**	**0.032**	0.031	0.027
	AURG-RMSE ↑	**2.365**	2.362	**2.365**	2.361	2.223
2	AUSE-REL ↓	0.023	**0.022**	**0.022**	0.023	0.028
	AUSE-RMSE ↓	**0.463**	0.479	0.464	0.468	0.669
	AURG-REL ↑	**0.039**	**0.039**	**0.039**	0.038	0.033
	AURG-RMSE ↑	**2.779**	2.763	2.777	2.774	2.573
3	AUSE-REL ↓	**0.031**	**0.031**	**0.031**	**0.031**	0.040
	AUSE-RMSE ↓	0.737	0.806	0.749	**0.730**	1.093
	AURG-REL ↑	**0.049**	**0.049**	**0.049**	**0.049**	0.040
	AURG-RMSE ↑	3.301	3.232	3.289	**3.308**	2.945
4	AUSE-REL ↓	0.050	0.053	0.051	**0.049**	0.064
	AUSE-RMSE ↓	1.364	1.582	1.430	**1.268**	2.029
	AURG-REL ↑	0.058	0.054	0.056	**0.059**	0.044
	AURG-RMSE ↑	3.834	3.615	3.767	**3.929**	3.168
5	AUSE-REL ↓	0.064	0.069	0.066	**0.059**	0.080
	AUSE-RMSE ↓	2.043	2.414	2.157	**1.760**	3.021
	AURG-REL ↑	0.063	0.057	0.061	**0.067**	0.046
	AURG-RMSE ↑	4.213	3.842	4.098	**4.496**	3.235

Table A14: Aleatoric uncertainty estimation results on Monocular Depth Estimation task. The additional aleatoric uncertainty estimation results are listed in the last column.

of σ_{Θ_2} can also be regarded as aleatoric uncertainty estimation. We provide the corresponding results in Tab. A14. We also put the partial results from the other distribution assumptions to make a comparison. From the last column, we can see that the aleatoric uncertainty given by the Dirichlet outputs underperforms the other solutions on quantitative metrics since the discretization affects the original numerical values of the prediction errors on the ID training data. Yet, the performance reduction is not huge and unacceptable.

A.5.5 Full results on epistemic uncertainty estimation Tab. A15 shows the results on the robustness of different methods for epistemic uncertainty estimation under dataset change. Tab. A16 provides the results on evaluating the robustness of the uncertainty estimation methods on unseen patterns during training. As we can see, in Tab. A16, most distribution assumptions can help AuxUE achieve good AUC and AUPR results, which shows that these AuxUEs all fit the ID data well. Yet, they can not assign consistent and higher uncertainty to the sky areas.

A.5.6 Procedures and illustration for building sky pattern as the OOD examples In pixel-wise regression, when the main task model is applied to the same scenario as the one used in the training process, there will still be some OOD patterns in the image. For example, given an image shown Fig. A4-A, the corresponding depth ground truth

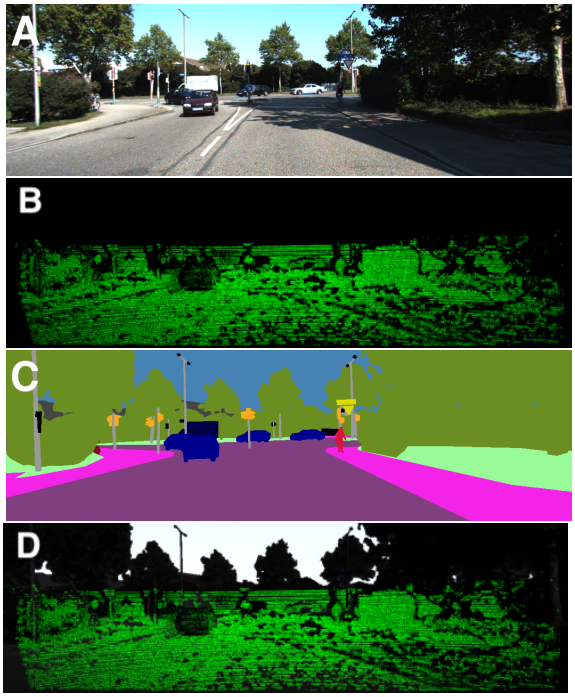


Figure A4: Visualization on the OOD example detection in monocular depth estimation. A: input image, B: ground truth depth map, green points represent pixels with depth ground truth; C: semantic ground truth map, blue areas represent pixels with sky pattern; D: ID-OOD map, green points represent the depth ground truth, i.e., the ID part. White parts represent the sky pattern, which is the OOD part.

map is provided in Fig. A4-B, with the annotated pixels colored in green. We can see that a big part of the pixels are not annotated by LIDAR because the orientation and the nature of LIDAR prevent it from covering all the ranges in the scene. We consider that the sky pattern is not annotated with depth value, but it can be labeled in the semantic map shown in Fig. A4-C. In this case, when we have a monocular depth estimation task, we can take the sky pattern as OOD examples and the pixels with depth annotations as the ID examples, as shown in Fig. A4-D, where the OOD patterns are colored in white and the ID patterns are colored in green.

B Loss functions based on different distribution assumptions

B.1 Loss functions

We follow the notations in Sec 3 of the main paper and construct the loss functions based on different distribution assumptions. The following loss functions are used in the toy example, age estimation, and monocular depth estimation.

Laplace (Lap) distribution

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N} \log(2\sigma_{\Theta_1}(\mathbf{x}^{(i)})) + \frac{|y^{(i)} - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)})|}{\sigma_{\Theta_1}(\mathbf{x}^{(i)})} \tag{A8}$$

	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 A15: Epistemic uncertainty estimation results encountering dataset change on Monocular depth estimation task. The evaluation dataset used here is NYU indoor depth dataset.

		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 A16: 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).

we choose this distribution assumption for aleatoric uncertainty estimation in our generalized AuxUE solution.

Generalized Gaussian (Ggau) distribution

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|y^{(i)} - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)})|}{\hat{\alpha}^{(i)}}\right)^{\hat{\beta}^{(i)}} - \log\frac{\hat{\beta}^{(i)}}{\hat{\alpha}^{(i)}} + \log\Gamma(\frac{1}{\hat{\beta}^{(i)}}) \tag{A9}$$

with $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\hat{\alpha}^{(i)}, \hat{\beta}^{(i)})$, which means σ_{Θ_1} will output two other components (except for $\tilde{y}^{(i)}$ defined in (Upadhyay et al. 2022) which stands for $f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)})$ in our case) for Generalized Gaussian distribution.

Normal Inverse Gamma (NIG) distribution

$$\begin{aligned}\mathcal{L}_1(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}&\frac{1}{2}\log(\frac{\pi}{\nu^{(i)}}) - \alpha^{(i)}\log(\Omega^{(i)}) \\ &+ (\alpha^{(i)} + \frac{1}{2})\log((y^{(i)} - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)}))^2\nu^{(i)} + \Omega^{(i)}) \\ &+ \log(\frac{\Gamma(\alpha^{(i)})}{\Gamma(\alpha^{(i)} + \frac{1}{2})})\end{aligned} \tag{A10}$$

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

$$\mathcal{L}(\Theta_1) = \mathcal{L}_1(\Theta_1) + \lambda_{\text{NIG}} \cdot \mathcal{L}_2(\Theta_1) \tag{A11}$$

where $\Omega^{(i)} = 2\beta^{(i)}(1+\nu^{(i)})$. $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\hat{\alpha}^{(i)}, \hat{\beta}^{(i)}, \hat{\nu}^{(i)})$, which means σ_{Θ_1} will output three other components (except for $\gamma^{(i)}$ defined in (Amini et al. 2020) which stands for $f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)})$ in our case) for Generalized Gaussian distribution. In our experiment, we set $\lambda_{\text{NIG}} = 0.01$.

B.2 Modifications in Super-resolution task

In the BayesCap pipeline, the authors discover that, in their AuxUE, the reconstruction of the main task prediction can increase the uncertainty estimation performance. The reconstructed main task prediction is denoted as $\tilde{\mathbf{y}}$. We thus follow this idea in practice but only in the super-resolution experiment to have a fair comparison for different distribution assumptions. The loss function will have a slight difference and an additional identity mapping loss (Upadhyay et al. 2022) than the proposed one in the main paper. We list the modified loss functions as follows. The weight for the identity mapping loss $\lambda_{\text{Identity}}$ is set the same as in the original BayesCap.

Laplace (Lap) distribution

$$\begin{aligned}\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}&\log(2\sigma_{\Theta_1}(\mathbf{x}^{(i)})) + \frac{|y^{(i)} - \tilde{y}^{(i)}|}{b^{(i)}} \\ &+ \lambda_{\text{Identity}} \cdot |f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}^{(i)}) - \tilde{y}^{(i)}|^2\end{aligned} \tag{A12}$$

where $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\tilde{y}^{(i)}, b^{(i)})$.

Generalized Gaussian (Ggau) distribution

$$\mathcal{L}(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|y^{(i)}-\tilde{y}^{(i)}|}{\hat{\alpha}^{(i)}}\right)^{\hat{\beta}^{(i)}} - \log\frac{\hat{\beta}^{(i)}}{\hat{\alpha}^{(i)}} + \log\Gamma(\frac{1}{\hat{\beta}^{(i)}}) + \lambda_{\text{Identity}}\cdot|f_{\boldsymbol{\omega}}(\mathbf{x}^{(i)})-\tilde{y}^{(i)}|^2 \quad \text{(A13)}$$

where $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\tilde{y}^{(i)}, \hat{\alpha}^{(i)}, \hat{\beta}^{(i)})$.

Normal Inverse Gamma (NIG) distribution

$$\mathcal{L}_1(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\log(\frac{\pi}{\nu^{(i)}}) - \alpha^{(i)}\log(\Omega^{(i)}) + (\alpha^{(i)}+\frac{1}{2})\log((y^{(i)}-\tilde{y}^{(i)})^2\nu^{(i)}+\Omega^{(i)}) + \log(\frac{\Gamma(\alpha^{(i)})}{\Gamma(\alpha^{(i)}+\frac{1}{2})}) \quad \text{(A14)}$$

$$\mathcal{L}_2(\Theta_1) = \frac{1}{N}\sum_{i=1}^{N}|y^{(i)}-\tilde{y}^{(i)}|\cdot(2\nu^{(i)}+\alpha^{(i)}) \quad \text{(A15)}$$

$$\mathcal{L}(\Theta_1) = \mathcal{L}_1(\Theta_1) + \lambda_{\text{NIG}}\cdot\mathcal{L}_2(\Theta_1) + \lambda_{\text{Identity}}\cdot|f_{\boldsymbol{\omega}}(\mathbf{x}^{(i)})-\tilde{y}^{(i)}|^2 \quad \text{(A16)}$$

where $\Omega^{(i)} = 2\beta^{(i)}(1+\nu^{(i)})$, and $\sigma_{\Theta_1}(\mathbf{x}^{(i)}) = (\tilde{y}^{(i)}, \hat{\alpha}^{(i)}, \hat{\beta}^{(i)}, \nu^{(i)})$. In our experiment, we set $\lambda_{\text{NIG}} = 0.01$.

C Epistemic uncertainty estimation on AuxUE using prediction errors

C.1 Discretization for pixel-wise regression tasks

Given an image input $\mathbf{x}^{(i)}$, we can achieve an output map $\hat{\mathbf{y}}^{(i)}$. We consider the error map $\boldsymbol{\epsilon}^{(i)}$ contains $J^{(i)}$ valid pixels, and subscript j as the indicator of the pixel. The values are sorted in ascending order, denoted by $\boldsymbol{\epsilon}'^{(i)}$, with the same elements as $\boldsymbol{\epsilon}^{(i)}$. We divide $\boldsymbol{\epsilon}'^{(i)}$ into K subsets of equal size, represented by $\{\boldsymbol{\epsilon}_k^{(i)}\}_{k=1}^K$:

$$\left\{\boldsymbol{\epsilon}_k^{(i)} \mid \epsilon^{(i)}_{\lfloor J^{(i)}*\frac{k-1}{K}\rceil} \le \epsilon_j^{(i)} < \epsilon^{(i)}_{\lfloor J^{(i)}*\frac{k}{K}\rceil}\right\}_{k=1}^{K} \quad \text{(A17)}$$

$\lfloor\cdot\rceil$ denotes the rounding operation. Each value in $\boldsymbol{\epsilon}_k^{(i)}$ is in the range of $\lfloor J^{(i)}*\frac{k-1}{K}\rceil$th and $\lfloor J^{(i)}*\frac{k}{K}\rceil$th value of the whole prediction error set $\boldsymbol{\epsilon}^{(i)}$. Each error value $\epsilon_j^{(i)}$ is then replaced by the index of its corresponding subset $k\in[1,K]$ and transformed into a one-hot vector, denoted by $\bar{\boldsymbol{\epsilon}}_j^{(i)}$, as the final training target. Specifically, the one-hot vector is defined as:

$$\bar{\boldsymbol{\epsilon}}_j^{(i)} = [\bar{\epsilon}_{j,1}^{(i)}\dots\bar{\epsilon}_{j,k}^{(i)}\dots\bar{\epsilon}_{j,K}^{(i)}]^{\mathrm{T}} \in \mathbb{R}^K \quad \text{(A18)}$$

where $\bar{\epsilon}_{j,k}^{(i)} = 1$ if $\epsilon_j^{(i)}$ belongs to the kth subset, and 0 otherwise.

C.2 Demo code for discretization operation

For the sake of clarification, we provide the demonstration code in DemoCode 1 directly in Python and PyTorch. The function `discretization_imagelevel` and `discretization_pixelwise` represent the discretization for image-level and pixel-wise tasks, respectively. Note that some modifications might be needed when deploying them in practice.

C.3 Modeling epistemic uncertainty using prediction errors

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. Since we have a trained main task DNN, and as proposed in (Malinin and Gales 2018), 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:

$$P(\boldsymbol{\omega}|\mathcal{D}) = \delta(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}) \rightarrow P(y|\mathbf{x},\mathcal{D}) \approx P(y|\mathbf{x},\hat{\boldsymbol{\omega}}) \quad \text{(A19)}$$

with δ being the Dirac function.

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

$$P(\mu,\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}}) = P(\mu|\sigma^2,\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}}) \quad \text{(A20)}$$

According to Eq. A19, we regard the μ depends only on $\mathbf{x}$ and the main task model $\hat{\boldsymbol{\omega}}$:

$$\begin{aligned} P(\mu,\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}}) &= P(\mu|\mathbf{x},\hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}}) \\ &= \delta(\mu - f_{\hat{\boldsymbol{\omega}}}(\mathbf{x}))P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}}) \end{aligned} \quad \text{(A21)}$$

We introduce $\boldsymbol{\alpha}$ and re-write $P(y|\mathbf{x},\hat{\boldsymbol{\omega}})$ in Eq. A19 as:

$$\begin{aligned} P(y|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}}) &= \iint P(y|\mu,\sigma^2)P(\mu,\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\mu d\sigma^2 \\ &\overset{(a)}{=} \iint P(y|\mu,\sigma^2)P(\mu|\sigma^2,\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\mu d\sigma^2 \\ &= \iint P(y,\mu|\sigma^2,\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\mu d\sigma^2 \\ &\overset{(b)}{=} \int\delta(\mu-f_{\hat{\omega}}(\mathbf{x}))d\mu\int P(y|\sigma^2,\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\sigma^2 \\ &= \int P(y|\mathbf{x},\sigma^2)P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\sigma^2 \end{aligned} \quad \text{(A22)}$$

where the equality (a) and (b) in Eq. A22 are given by Eq. A20 and Eq. A21, respectively.

In summary, we have

$$\begin{aligned} P(y|\mathbf{x},\mathcal{D}) &= \iint P(y|\mathbf{x},\sigma^2)P(\sigma^2|\boldsymbol{\omega})P(\boldsymbol{\omega}|\mathcal{D})d\sigma^2 d\boldsymbol{\omega} \\ &= \int P(y|\mathbf{x},\sigma^2)P(\sigma^2|\mathcal{D})d\sigma^2 \\ &\overset{(a)}{\approx} \int P(y|\mathbf{x},\sigma^2)P(\sigma^2|\mathbf{x},\boldsymbol{\alpha},\hat{\boldsymbol{\omega}})d\sigma^2 \end{aligned} \quad \text{(A23)}$$

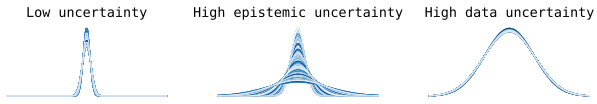


Figure A5: Visualizations on desired behaviors of regression results in auxiliary uncertainty estimation scenario. Different types of uncertainty result in different distributions of the variance under Gaussian assumption.

where the approximation (a) in Eq. A23 is given by Eq. A22.

In this case, we first consider ϵ to be drawn from a continuous distribution parameterized by σ^2. Furthermore, we argue that $P(\sigma^2|\mathcal{D})$ describes the epistemic uncertainty when we have a trained and fixed main task model and the variational approach can be applied (Joo, Chung, and Seo 2020; Malinin and Gales 2018): $P(\sigma^2|\mathbf{x}, \boldsymbol{\alpha}, \hat{\boldsymbol{\omega}}) \approx P(\sigma^2|\mathcal{D})$. It shows the special case of the approximation for the posterior over y, where the mean is fixed and only variances differ. Similar to (Malinin et al. 2020), we can illustrate this by using the ensembles of the regression results as in Figure A5. The discrepancy in variances determines the epistemic uncertainty of the final prediction.

D Ablation study

The ablation studies are based on the monocular depth estimation task.

D.1 Hyperparameters

There are two main hyperparameters in our proposed DIDO: the number of the sets K in discretization and λ for the regularization term in the loss function (Eq. 4. in the main paper). In this section, we analyze the effect of these two hyperparameters.

The evaluations are based on epistemic uncertainty estimation on unseen patterns and dataset change detection.

The effect of K is shown in Fig. A6a and Fig. A6c. We test $K = \{8, 16, 32, 64\}$ and λ is fixed to 0.01. The AUC and AUPR performance decrease when we have bigger K, while on the Sky-All metric, bigger K provides better results. When the evaluation dataset is changed from KITTI to NYU (Nathan Silberman and Fergus 2012), bigger K can also provide better AUC and AUPR in identifying the change. We choose $K = 32$ to have a balanced performance.

The effect of λ is shown in Fig. A6b and Fig. A6d. We test $\lambda = \{1e-4, 1e-3, 1e-2, 1e-1\}$ and K is fixed to 32. We can see the AUC and AUPR performance decrease when we use bigger λ during training, while on the Sky-All metric, it performs better when using bigger λ during training. For the dataset change experiment, we can see when $\lambda = 0.01$, the DIDO can provide the best AUC and AUPR. We choose $\lambda = 0.01$ in the end for our model.

D.2 Necessity of using AuxUE

In this experiment, we apply DIDO on the main task BTS (Lee et al. 2019) model to see the impact on the main task performance and the uncertainty estimation performance. The comparison will only be conducted with the other modified main task models for fairness. In particular, we first adjust the BTS model to Single Predictive Uncertainty (Nix and Weigend 1994; Kendall and Gal 2017) variant (BTS-SinglePU), i.e., we first add an aleatoric uncertainty estimation head parallel to and identical to the depth estimation head on top of the original model. Then we add the same head for DIDO applied on σ_{Θ_2}. Thus there are three prediction heads on the modified BTS model corresponding to depth prediction, aleatoric uncertainty estimation and epistemic uncertainty estimation. We denote this variant as BTS-DIDO. We will compare BTS-DIDO with the original BTS model (Org) and the BTS-SinglePU model to check the impact on the main task. We also compare the BTS-DIDO with AuxUE + original BTS, BTS-DEns. and BTS-SinglePU models to check the uncertainty estimation performance.

To train the BTS-SinglePU models, we follow the original BTS settings for hyperparameters in training. We change the loss function to Gaussian NLL loss. For BTS-DIDO, we use the same hyperparameters as we used in AuxUEs for DIDO modeling, i.e., $K = 32$ and $\lambda = 0.01$. For the other hyperparameters, such as the batch size and learning rate, we follow the original BTS settings.

During training, we found that combining DIDO directly with the BTS will make the training unstable: the loss will be exploded after around fifteen epochs. As shown in Tab. A17, for the main task performance, the original BTS can outperform the others even for BTS-DEns. We argue that there are two reasons that might result in the performance reduction: BTS-DEns. component models (BTS-SinglePU models) are adjusted for the uncertainty output; SiLog loss (Eigen, Puhrsch, and Fergus 2014), which is specifically applied to the MDE task, is replaced by the Gaussian negative log-likelihood loss. However, when the noise severity increases ($S > 3$), BTS-DIDO and BTS-DEns. can perform better than the others. In particular, BTS-DIDO shows a more robust performance given the inputs with heavy perturbations.

For the uncertainty estimation performance, as shown in Tab. A18, BTS-DIDO and AuxUE achieve similar performance on AUC and AUPR metrics and on dataset change detection. While on Sky-All, AuxUE works slightly better than BTS-DIDO. For aleatoric uncertainty, since the main task performances for different models are different, the comparison can only be a reference. With a sacrifice on the main task performance, BTS-DIDO has the potential to achieve good uncertainty estimation performance.

We argue that it is necessary to use AuxUE to keep the main task performance when the input is relatively clean. Meanwhile, the good performance on BTS-DIDO under high severity perturbations makes it meaningful to work on stabilizing the training for DIDO-based models in the future.

D.3 Effectiveness of Dirichlet modeling

We show the effectiveness of the Dirichlet modeling instead of using the standard categorical modeling based on discretized prediction errors. For categorical modeling, we also choose $K = 32$ classes for discretization. We change the activation function on the top of σ_{Θ_2} from the ReLU function

```python
import torch
import torch.nn.functional as F

def discretization_imagelevel(data_loader, x, y, f_omega, K):
    """Discretization operation on image-level tasks.
    Args:
        data_loader: training or validation set data loader
        x: input image
        y: ground truth target
        f_omega: trained main DNN
        K: number of classes
    Returns:
        epsilon_bar: one-hot prediction error
    """
    epsilon = []
    for x_item, y_item in data_loader:
        epsilon.append(abs(y_item - f_omega(x_item)))
    epsilon = torch.cat(epsilon, dim=0)
    quantiles = torch.quantile(epsilon, torch.tensor(range(K+1)/K))
    classes = torch.tensor(range(K))
    for i, (q1, q2, c) in enumerate(zip(quantiles[:-1], quantiles[1:], classes)):
        if i == 0:
            mask_q1 = epsilon >= q1
        else:
            mask_q1 = epsilon > q1
        mask_q2 = epsilon <= q2
        mask_q = torch.logical_and(mask_q1, mask_q2)
        temp[mask_q] = c
    epsilon_bar = F.one_hot(temp, K)
    return epsilon_bar

def discretization_pixelwise(x, y, f_omega, K):
    """Discretization operation on pixel-wise tasks.
    Args:
        x: input image
        y: ground truth map
        f_omega: trained main DNN
        K: number of classes
    Returns:
        epsilon_bar: one-hot prediction error map
    """
    epsilon = abs(y - f_omega(x))
    temp = torch.zeros_like(epsilon)
    quantiles = torch.quantile(epsilon, torch.tensor(range(K+1)/K))
    classes = torch.tensor(range(K))
    for i, (q1, q2, c) in enumerate(zip(quantiles[:-1], quantiles[1:], classes)):
        if i == 0:
            mask_q1 = epsilon >= q1
        else:
            mask_q1 = epsilon > q1
        mask_q2 = epsilon <= q2
        mask_q = torch.logical_and(mask_q1, mask_q2)
        temp[mask_q] = c
    epsilon_bar = F.one_hot(temp, K)
    return epsilon_bar
```

DemoCode 1: Discretization code in Python. Both image-level and pixel-wise cases are provided. Note that the exact code in practice might need some modifications based on this.

to the Softmax function, then apply classical cross-entropy on the Softmax outputs. For measuring uncertainty, we use the Shannon-Entropy (Shannon 2001) on the Softmax outputs. As shown in Fig. A7, the Dirichlet modeling outperforms the Categorical modeling on all three metrics w.r.t. the OOD pattern detection. On the dataset change experiment, Categorial modeling provides 90.37 for AUC and 96.83 for AUPR, which underperforms the results given by Dirichlet modeling. This study shows the effectiveness of DIDO and the use of evidential learning in the AuxUE.

E More visualizations

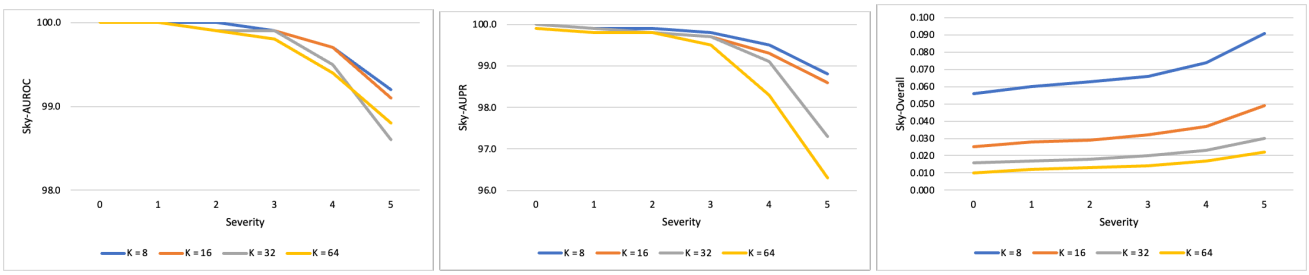
Fig. A8 shows more visualizations on monocular depth estimation. For aleatoric uncertainty estimation maps, since some of the values on the unseen part (mostly the upper part of the map) are extremely high ($>$1e4), we **clip the values** to the maximum predicted value on the pixels with ground truth for better illustration. As uncertainty estimates show, our proposed DIDO can highlight the patterns rarely appearing throughout the whole dataset, e.g., the windshield of the car, the underside of the car, the barbed wire fence, and the upper part of the image like the sky. However, only the sky part is a pattern that must have no ground truth depth values and have semantic segmentation annotations. This is the reason we chose only the sky as the OOD pattern. Note that DIDO won't always highlight the areas without ground truth. For instance, we can see DIDO does not always assign higher uncertainty on the parts with no ground truth for the body of the car or the road, since some of these patterns might have ground truth on the other images in the training set or share similar patterns which have ground truth values.

S	Methods	Main task performance							
		absrel ↓	log10 ↓	rms ↓	sqrel ↓	logrms ↓	d1 ↑	d2 ↑	d3 ↑
0	Org + AuxUE	**0.056**	**0.025**	**2.430**	**0.201**	**0.089**	**0.963**	0.994	**0.999**
	BTS-SinglePU	0.065	0.029	2.606	0.234	0.100	0.952	0.993	0.998
	BTS-DEns.	0.060	0.026	2.435	0.202	0.092	0.961	**0.995**	**0.999**
	BTS-DIDO	0.061	0.027	2.574	0.236	0.098	0.954	0.992	0.998
1	Org + AuxUE	**0.077**	**0.036**	**3.185**	**0.370**	**0.129**	**0.919**	**0.977**	**0.992**
	BTS-SinglePU	0.094	0.043	3.581	0.476	0.149	0.890	0.969	0.989
	BTS-DEns.	0.087	0.040	3.415	0.422	0.138	0.902	0.974	**0.992**
	BTS-DIDO	0.088	0.040	3.453	0.456	0.143	0.898	0.972	0.991
2	Org + AuxUE	**0.096**	**0.047**	**3.861**	**0.571**	**0.168**	**0.876**	**0.954**	0.979
	BTS-SinglePU	0.116	0.057	4.359	0.735	0.192	0.835	0.939	0.973
	BTS-DEns.	0.109	0.053	4.189	0.661	0.178	0.848	0.947	0.979
	BTS-DIDO	0.108	0.051	4.169	0.670	0.178	0.851	0.948	**0.980**
3	Org + AuxUE	**0.130**	0.069	**4.905**	**0.985**	0.237	**0.805**	0.908	0.949
	BTS-SinglePU	0.149	0.078	5.357	1.140	0.253	0.760	0.890	0.944
	BTS-DEns.	0.140	0.073	5.184	1.031	0.234	0.772	0.904	0.955
	BTS-DIDO	0.134	**0.067**	5.134	1.003	**0.228**	0.789	**0.912**	**0.961**
4	Org + AuxUE	0.195	0.117	6.591	1.888	0.370	0.680	0.808	0.874
	BTS-SinglePU	0.195	0.110	6.649	1.786	0.341	0.662	0.816	0.894
	BTS-DEns.	0.186	0.103	6.485	1.649	0.317	0.667	0.833	0.911
	BTS-DIDO	**0.170**	**0.089**	**6.292**	**1.485**	**0.293**	**0.711**	**0.862**	**0.930**
5	Org + AuxUE	0.265	0.172	8.259	2.932	0.508	0.555	0.696	0.783
	BTS-SinglePU	0.231	0.135	7.731	2.328	0.410	0.585	0.757	0.853
	BTS-DEns.	0.222	0.127	7.584	2.190	0.386	0.587	0.772	0.871
	BTS-DIDO	**0.211**	**0.116**	**7.484**	**2.065**	**0.367**	**0.621**	**0.799**	**0.890**

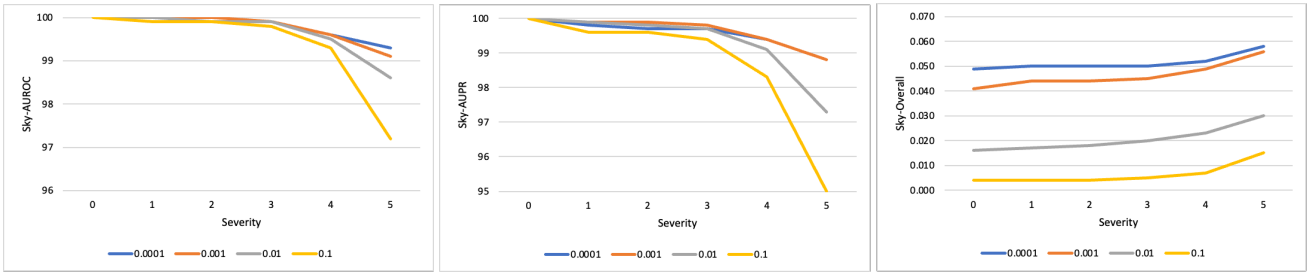
Table A17: Ablation study on the necessity of using AuxUE. Main task performance comparison on KITTI and KITTI-C.

S	Methods	Aleatoric uncertainty estimation				Epsitemic uncertainty: Unseen pattern			Dataset change	
		AUSE-REL ↓	AUSE-RMSE ↓	AURG-REL ↑	AURG-RMSE ↑	AUC ↑	AUPR ↑	Sky-All ↓	AUC ↑	AUPR ↑
0	Org + AuxUE	**0.013**	0.203	0.023	1.870	**100.0**	**100.0**	0.015	98.1	99.3
	BTS-SinglePU	0.016	0.222	0.026	1.978	89.0	62.0	**0.005**	64.2	78.3
	DEns.	0.014	**0.195**	0.024	1.866	93.5	70.0	**0.005**	62.1	76.7
	BTS-DIDO	**0.013**	0.207	**0.028**	**1.990**	**100.0**	**100.0**	0.017	**98.5**	**99.5**
1	Org + AuxUE	0.019	0.336	0.031	2.361	**100.0**	**99.9**	0.016		
	BTS-SinglePU	0.021	0.330	0.038	**2.657**	86.9	59.1	**0.005**		
	BTS-DEns.	0.019	**0.285**	0.036	2.573	92.8	68.0	**0.005**		
	BTS-DIDO	**0.017**	0.308	**0.041**	2.608	**100.0**	**99.9**	0.027		
2	Org + AuxUE	0.023	0.468	0.038	2.774	99.9	99.8	0.017		
	BTS-SinglePU	0.026	0.443	0.046	**3.150**	86.6	58.9	**0.005**		
	BTS-DEns.	0.022	**0.387**	0.044	3.078	92.3	67.0	**0.005**		
	BTS-DIDO	**0.021**	0.396	**0.050**	3.093	**100.0**	**99.9**	0.033		
3	Org + AuxUE	0.031	0.730	0.049	3.308	**99.9**	99.7	0.018		
	BTS-SinglePU	0.031	0.619	0.055	3.719	86.6	59.5	**0.005**		
	BTS-DEns.	0.027	0.526	0.054	3.685	91.6	65.7	**0.005**		
	BTS-DIDO	**0.023**	**0.500**	**0.062**	**3.749**	**99.9**	**99.8**	0.036		
4	Org + AuxUE	0.049	1.268	0.059	3.929	99.6	99.1	0.023		
	BTS-SinglePU	0.038	0.905	0.067	4.345	87.2	61.7	**0.005**		
	BTS-DEns.	0.032	0.734	0.067	4.401	91.8	67.2	**0.005**		
	BTS-DIDO	**0.029**	**0.680**	**0.074**	**4.446**	**99.8**	**99.7**	0.041		
5	Org + AuxUE	0.059	1.760	0.067	4.496	98.5	97.1	0.035		
	BTS-SinglePU	0.045	1.202	0.075	4.831	87.5	64.6	**0.005**		
	BTS-DEns.	**0.036**	**0.890**	0.080	**5.044**	92.2	70.4	**0.005**		
	BTS-DIDO	**0.036**	1.010	**0.084**	4.964	**99.5**	**99.3**	0.051		

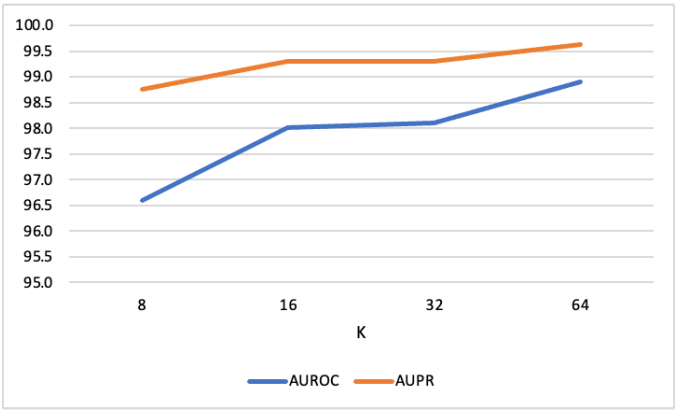
Table A18: Ablation study on the necessity of using AuxUE. Epistemic uncertainty estimation performance comparison on KITTI and KITTI-C. On clean KITTI, the extra columns stand for the dataset change experiment.



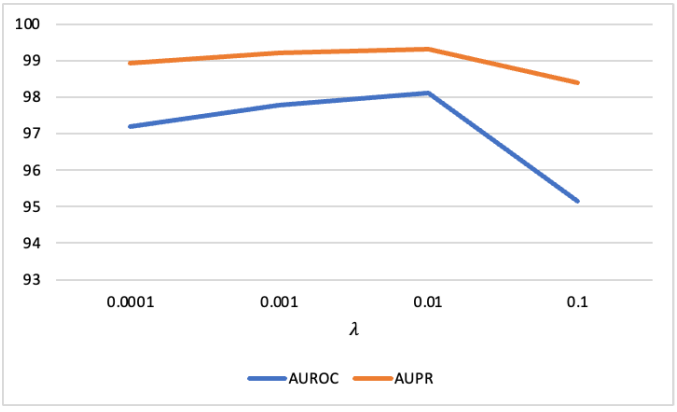
(a) Ablation study on K for DIDO on unseen patterns detection in KITTI dataset. The results are given by DIDO-based AuxUE with different numbers of classes (K) in discretization.



(b) Ablation study on λ for DIDO on unseen patterns detection in KITTI dataset. The results are given by DIDO-based AuxUE with ($K = 32$) trained by using different λ for the regularization term in loss $L(\Theta_2)$.



(c) Ablation study on K for DIDO on dataset change detection in monocular depth estimation. The evaluation is made by taking KITTI outdoor dataset as the In-Distribution data and NYU indoor dataset as the Out-of-Distribution data.



(d) Ablation study on λ for DIDO on dataset change detection in monocular depth estimation. The evaluation is made by taking KITTI outdoor dataset as the In-Distribution data and NYU indoor dataset as the Out-of-Distribution data.

Figure A6: Ablation study on hyperparameters for DIDO on monocular depth estimation.

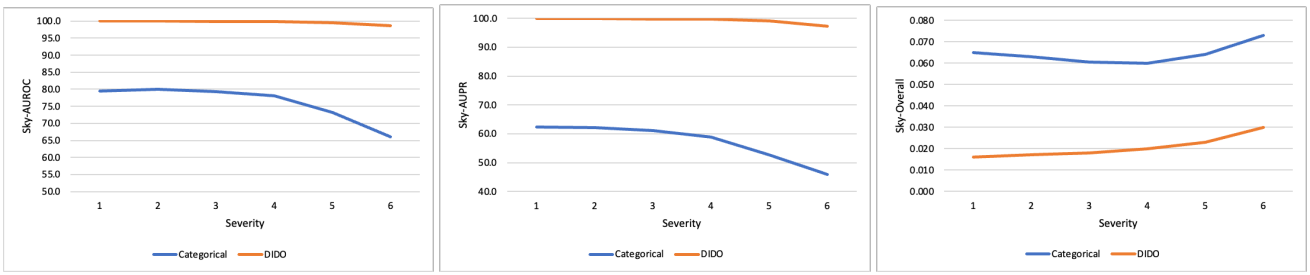
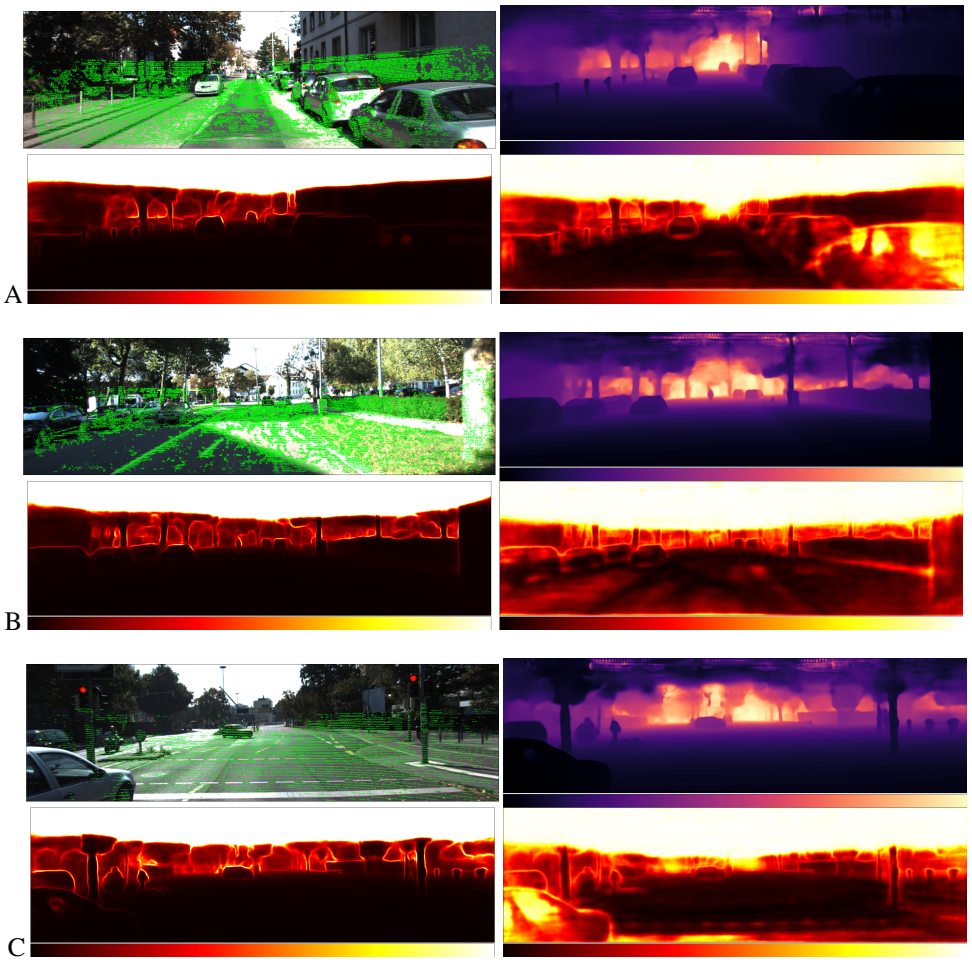


Figure A7: Ablation study on the effectiveness of Dirichlet modeling for DIDO on monocular depth estimation. $K = 32$ for both Categorical and Dirichlet modeling cases.



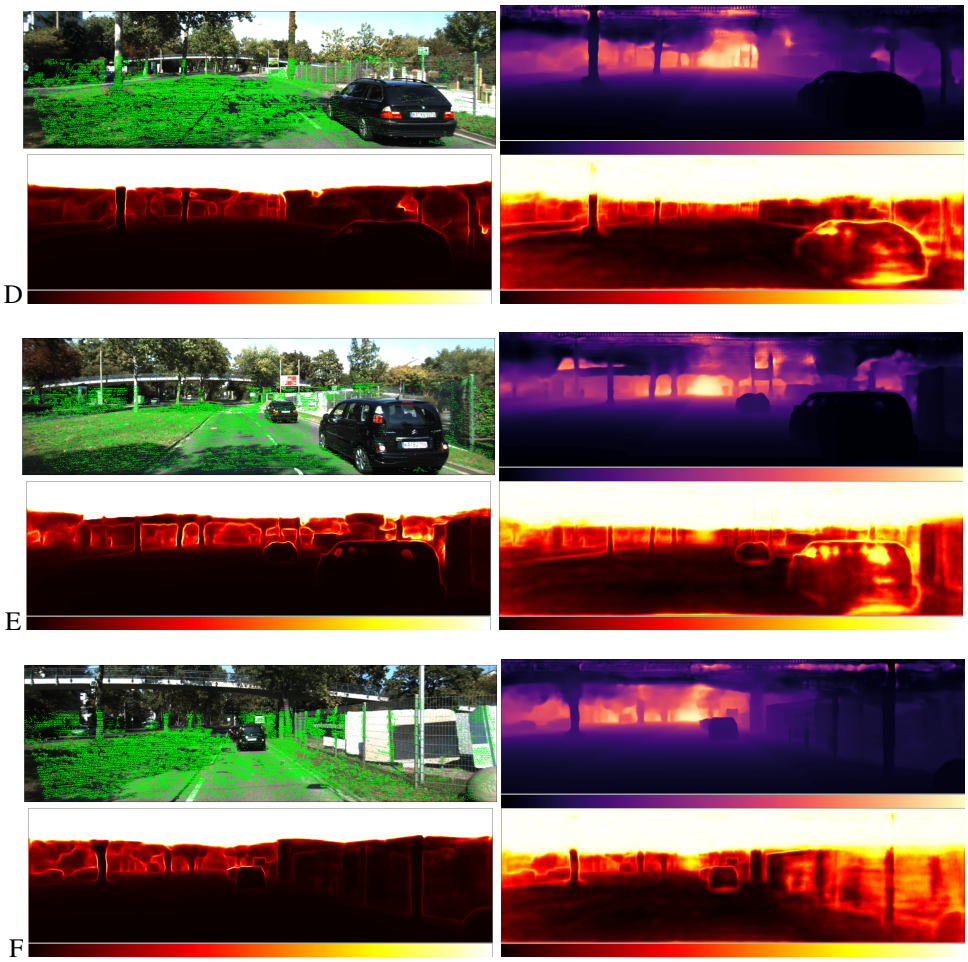


Figure A8: Visualizations on monocular depth estimations and corresponding uncertainty quantification results. The color bars and the image orders follow the ones in Fig.3 of the main paper.