On the Practicality of Deterministic Epistemic Uncertainty

Janis Postels [*1] **Mattia Segu** [*1] **Tao Sun** [1] **Luca Sieber** [1] **Luc Van Gool** [1] **Fisher Yu** [1] **Federico Tombari** [2 3]

Abstract

A set of novel approaches for estimating epistemic uncertainty in deep neural networks with a single forward pass has recently emerged as a valid alternative to Bayesian Neural Networks. On the premise of informative representations, these deterministic uncertainty methods (DUMs) achieve strong performance on detecting out-of-distribution (OOD) data while adding negligible computational costs at inference time. However, it remains unclear whether DUMs are well calibrated and can seamlessly scale to real-world applications - both prerequisites for their practical deployment. To this end, we first provide a taxonomy of DUMs, and evaluate their calibration under continuous distributional shifts. Then, we extend them to semantic segmentation. We find that, while DUMs scale to realistic vision tasks and perform well on OOD detection, the practicality of current methods is undermined by poor calibration under distributional shifts.

1. Introduction

Despite the dramatic enhancement of predictive performance of deep learning (DL), its adoption remains limited due to unpredictable failure on out-of-distribution (OOD) samples (Blanchard et al., 2011; Muandet et al., 2013) and adversarial attacks (Szegedy et al., 2013). Uncertainty estimation techniques aim at bridging this gap by providing accurate confidence levels on a model's output, allowing for a safe deployment of neural networks (NNs) in safety-critical tasks, *e.g.* autonomous driving or medical applications.

While Bayesian Neural Networks (BNNs) represent the predominant holistic solution for quantifying uncertainty (Hinton & Van Camp, 1993; Neal, 2012), exactly modelling their full posterior is often intractable, and scalable versions usually require expensive variational approximations (Kingma et al., 2015; Gal & Ghahramani, 2016; Zhang et al., 2018; Postels et al., 2019; Loquercio et al., 2020). Moreover, it has recently been shown that true Bayes posterior can also lead to poor uncertainty (Wenzel et al., 2020). Thus, efficient approaches to uncertainty estimation largely remain an open problem, limiting the adoption within real-time applications under strict memory, time and safety requirements.

Recently, a promising line of work emerged for estimating epistemic uncertainty of a NN with a single forward pass while treating its weights deterministically. By regularizing the hidden representations of a model, these methods represent an efficient and scalable solution to epistemic uncertainty estimation and to the related OOD detection problem. In contrast to BNNs, Deterministic Uncertainty Methods (DUMs) quantify epistemic uncertainty using the distribution of latent representations (Alemi et al., 2018; Wu & Goodman, 2020; Charpentier et al., 2020; Mukhoti et al., 2021; Postels et al., 2020; Charpentier et al., 2021) or by replacing the final softmax layer with a distance-sensitive function (Mandelbaum & Weinshall, 2017; Van Amersfoort et al., 2020; Liu et al., 2020; van Amersfoort et al., 2021). Further, these methods have been applied to practical problems such as object detection (Gasperini et al., 2021). While OOD detection is a prerequisite for a safe deployment of DL in previously unseen scenarios, the calibration - *i.e.* how well uncertainty correlates with model performance - of such methods under continuous distributional shifts is equally important. Measuring the calibration of an epistemic uncertainty estimate on shifted data investigates whether it entails information about the predictive performance of the model. This an essential requirement for uncertainty and, unlike OOD detection, an evaluation that is not model-agnostic - i.e. one cannot perform well without taking the predictive model into account. Nonetheless, previous work falls short of investigating calibration, and solely focuses on OOD detection (Mandelbaum & Weinshall, 2017; Van Amersfoort et al., 2020; Wu & Goodman, 2020; van Amersfoort et al., 2021; Mukhoti et al., 2021). Further, DUMs have thus far only been evaluated on toy datasets for binary classification, small-scale image classification tasks (Van Amersfoort et al., 2020; Mukhoti et al., 2021) and toy prediction problems in natural language processing (Liu et al., 2020). Despite claiming to solve practical issues of traditional uncertainty

[*]Equal contribution [1]ETH Zurich [2]Technical University Munich [3]Google. Correspondence to: Janis, Postels <jpostels@ethz.ch>, Mattia, Segu <segum@ethz.ch>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

estimation approaches, the practicality of DUMs remains to be assessed on more challenging tasks.

This work investigates whether recently proposed DUMs are a realistic alternative for practical uncertainty estimation. In particular: (i) we provide the first comprehensive taxonomy of DUMs; (ii) we analyze the calibration of DUMs under synthetic and realistic continuous distributional shifts; (iii) we evaluate the sensitivity of DUMs to their regularization strength; (iv) we scale DUMs to dense prediction tasks, *e.g.* semantic segmentation. Overall, we find that the practicality of many DUMs is undermined by their poor calibration under both synthetic and realistic distributional shifts. Moreover, some techniques for regularizing hidden representations demonstrate only weak correlation with OOD detection and calibration performance.

2. Related Work

Sources of uncertainty. Uncertainty in a model's predictions can arise from two different sources (Der Kiureghian & Ditlevsen, 2009; Kendall & Gal, 2017). While *aleatoric* uncertainty encompasses the noise inherent in the data and is consequently irreducible (Der Kiureghian & Ditlevsen, 2009), *epistemic* uncertainty quantifies the uncertainty associated with choosing the model parameters based on limited information, and vanishes - in principle - in the limit of infinite data. This work is concerned with estimating epistemic uncertainty.

Properties of epistemic uncertainty. This work distinguishes two properties of epistemic uncertainty - its performance on detecting OOD samples and its calibration (i.e. its correlation with model performance under distributional shifts). While the latter has been explored in the case of probabilistic approaches to uncertainty estimation (Snoek et al., 2019), we are the first to investigate the behaviour of DUMs in this scenario. Notably, (Gustafsson et al., 2020) evaluates prominent scalable epistemic uncertainty estimates on semantic segmentation. However, they investigate calibration only on in-distribution data. Further, although (Liu et al., 2020) evaluates the calibration of their approach, they do so exclusively on in-distribution data. Lastly, one line of work focuses on calibrating NNs to achieve good OOD detection (Lee et al., 2018; Bates et al., 2021). However, this tackles the model-agnostic task of OOD detection which is only one part of a good epistemic uncertainty estimate.

BNNs (Neal et al., 2011; Neal, 2012) represent a principled way of measuring uncertainty. However, their intractable posterior distribution requires approximate inference methods, such as Markov Chain Monte-Carlo (Neal et al., 2011) or Variational Inference (VI) (Hinton & Van Camp, 1993). While these methods traditionally struggle with large datasets and architectures, a variety of scalable approaches - often based on VI - have recently emerged.

Deep ensembles, which typically consist of identical models trained from different initializations, have been introduced to the deep learning community by Lakshminarayanan *et al.* (Lakshminarayanan et al., 2017) and extended by (Wen et al., 2020; Dusenberry et al., 2020). While deep ensembles are widely regarded as a strong baseline for estimating epistemic uncertainty, they come with high computational as well as memory costs.

Efficient approaches. Recently, approaches based on stochastic regularization have been developed (Kingma et al., 2015; Gal & Ghahramani, 2016; Zhang et al., 2018; Teye et al., 2018; Osband et al., 2021). By keeping stochasticity at inference time, they estimate uncertainty using multiple forward passes. Another line of work estimates the posterior distribution using the Laplace-approximation (Ritter et al., 2018; Lee et al., 2020; Sharma et al., 2021). Moreover, efficient ensemble methods were proposed producing predictions using a single model (Rupprecht et al., 2017; Dusenberry et al., 2020; Wen et al., 2020; Havasi et al., 2020; Rame et al., 2021). Despite promising results on large-scale tasks and parameter reduction, these methods still require sampling through the model, which can render them impractical given limited compute. To estimate uncertainty in real-time and resource-demanding tasks, recent work has focused on providing uncertainty estimates with a *single forward pass*. One line of work proposes a principled approach for variance propagation in NNs (Postels et al., 2019; Haußmann et al., 2020; Loquercio et al., 2020). These approaches fundamentally differ from DUMs due to their probabilistic treatment of the parameters. Notably, another line of work proposes efficient approaches to estimate aleatoric uncertainty (Bishop, 1994; Oberdiek et al., 2018; Oh et al., 2019).

Recently, DUMs showed promising results on OOD detection. By leveraging distances and densities in the feature space of a NN, these methods provide confidence estimates while adding negligible computational cost. Since they are united in their deterministic treatment of the weights, we term them Deterministic Uncertainty Methods (DUMs). The next section provides a taxonomy of DUMs.

3. Taxonomy for Deterministic Uncertainty Quantification

Since DUMs thus far denote a novel and scattered phenomenon in literature, it is necessary to provide an overview of the research landscape. Note, that we provide more material on existing DUM trends identified in this taxonomy (Sec. A.1) as well as the strength and weaknesses of each individual DUM (Sec. A.1.4). Existing DUMs mostly differentiate along two axis. Firstly, DUMs apply different regularization techniques to equip their representations with the ability to differentiate between in-distribution (ID) and

Table 1. Taxonomy of DUMs. Methods are grouped according to their regularization (**Reg.**) of the hidden representations (rows), and their uncertainty estimation method (columns). For reference: DCS (Mandelbaum & Weinshall, 2017), DUQ (Van Amersfoort et al., 2020), SNGP (Liu et al., 2020), DUE (van Amersfoort et al., 2021), DDU (Mukhoti et al., 2021), DCU (Wu & Goodman, 2020; Winkens et al., 2020), MIR (Postels et al., 2020), Invertible networks(Ardizzone et al., 2018; Nalisnick et al., 2019; Ardizzone et al., 2020), PostNet(Charpentier et al., 2020)

DUMs		**Uncertainty Estimation Method**			
		Discriminative		Generative	
		Class centroid	Gaussian Processes	Gaussian Mixture Models	Normalizing Flows
Reg.	Distance awareness	DCS, DUQ	SNGP, DUE	DDU	-
	Informative representations	-	-	DCU, MIR	Invertible networks, PostNet

OOD data (Sec. 3.1). This is important because the primary goal of DUMs is to quantify epistemic uncertainty while treating the weights of a NN deterministically in order to avoid sampling at inference time. Since epistemic uncertainty is expected to increase on OOD data, the representations of a NN need to be sensitive to the input distribution. However, discriminative models suffer from the fundamental problem of feature collapse (Van Amersfoort et al., 2020; Mukhoti et al., 2021) which has to be counteracted using appropriate regularization. Secondly, DUMs use different methods to estimate uncertainty from such regularized representations (Sec. 3.2). Tab. 1 shows an overview of the resulting taxonomy.

Feature Collapse. Discriminative models can learn to discard a large part of their input information, as exploiting spurious correlations may lead to better performance on the training data distribution (Peters et al., 2017; Segù et al., 2020). Such invariant representations learned may be blind to distributional shifts, resulting in a collapse of OOD embeddings to in-distribution features. This problem is known as *feature collapse* (Van Amersfoort et al., 2020), and it makes OOD detection based on high-level representations impossible.

3.1. Regularization of Representations

We group DUMs according to their approach to mitigating feature collapse. Currently, there are two main paradigms - distance awareness and informative representations - which we discuss in Sec. 3.1.1 and Sec. 3.1.2.

3.1.1. Distance Awareness

Distance-aware representations avoid feature collapse by relating distances between latent representations to distances in the input space. Therefore, one constrains the bi-Lipschitz constant, as it enforces a lower and an upper bound to expansion and contraction performed by a model. A lower bound enforces that different inputs are mapped to distinct representations and, thus, provides a solution to feature collapse. The upper bound enforces smoothness, *i.e.* small changes in the input do not result in large changes in the latent space. While there exist other approaches, *e.g.* (Obukhov et al., 2021), recent proposals have primarily adopted two methods to impose the bi-Lipschitz constraint.

The two-sided **Gradient Penalty** relates changes in the input to changes in feature space by directly constraining the gradient of the input (Van Amersfoort et al., 2020). Note, that this leads to large computational overhead as it requires differentiation of the gradients of the input with respect to the NN's parameters. **Spectral Normalization (SN)** (Miyato et al., 2018) is a less computationally-demanding alternative. SN is applicable to residual layers and normalizes the weights W of each layer using their spectral norm $sn(W)$ to constrain the bi-Lipschitz constant. Various DUMs - SNGP (Liu et al., 2020), DUE (van Amersfoort et al., 2021) and DDU (Mukhoti et al., 2021) - rely on SN to enforce distance-awareness of hidden representations. More details on gradient penalty and SN can be found in the supplement (Sec. A.1.1).

Notably, the bi-Lipschitz constraint is defined with respect to a fix distance measure, which can be difficult to choose for high-dimensional data distributions. For example, SN (Van Amersfoort et al., 2020; Liu et al., 2020; van Amersfoort et al., 2021) corresponds to the L_2 distance. While SN has empirically been found to perform well, it has been suggested (Singla & Feizi, 2019) that popular SN approximations behave sub-optimally, and their interaction with losses, architecture and optimization is yet to be fully understood (Rosca et al., 2020). Principled approaches to providing exact singular values in convolutional layers (Sedghi et al., 2018) result in prohibitive computational complexity. Further, (Smith et al., 2021) provides an explanation for effectiveness of SN in convolutional residual NNs.

3.1.2. Informative Representations

While distance-awareness achieves remarkable performance on OOD detection, it does not explicitly preserve sample-specific information. Thus, depending on the underlying distance metric it may discard useful information about the input or act overly sensitive. An alternative line of work

avoids feature collapse by learning informative representations (Alemi et al., 2018; Wu & Goodman, 2020; Postels et al., 2020; Nalisnick et al., 2019; Ardizzone et al., 2018; 2020), thus forcing discriminative models to preserve information in its hidden representations beyond what is required to solve a task independent of the choice of an underlying distance metric. Notably, while representations that are aware of distances in the input space are also informative, both categories remain fundamentally different in their approach to feature collapse. While distance-awareness is based on the choice of a specific distance metric tying together input and latent space, informative representations incentivize a NN to store more information about the input using an auxiliary task (Postels et al., 2020; Wu & Goodman, 2020) or forbid information loss by construction (Ardizzone et al., 2018; Nalisnick et al., 2019; Ardizzone et al., 2020). There are currently four distinct approaches.

Contrastive learning (Oord et al., 2018) has emerged as an approach for learning representations that are both informative and discriminative and provably maximize the mutual information with the data distribution. This is utilized by Wu *et al.* (Wu & Goodman, 2020) and Winkens *et al.* (Winkens et al., 2020), who apply SimCLR (Chen et al., 2020) to regularize hidden representations for a discriminative task by using a contrastive loss for pretraining and fine-tuning to force representations to discriminate between individual instances.

Reconstruction regularization (Postels et al., 2020) (MIR) instead forces the intermediate activations to fully represent the input. This is achieved by adding a decoder branch fed with the activations of a given layer to reconstruct the input.

Entropy regularization. PostNet (Charpentier et al., 2020) learns the class-conditional distribution of hidden representations end-to-end using a Normalizing Flow (NF) parameterizing a Dirichlet distribution. This allows them to enforce informative representations by implicitly encouraging large entropy of the NF during training. We refer to the supplement for details and further explanations (Sec. A.1.2).

Invertible Neural Networkss (INNs) (Jacobsen et al., 2018; Ardizzone et al., 2018; Nalisnick et al., 2019; Ardizzone et al., 2020), built via a cascade of invertible layers, cannot discard information except at the final classification stage. Consequently, the mutual information between input and hidden representation is maximized by construction. Interestingly, Behrmann *et al.* (Behrmann et al., 2018) showed that a ResNet is invertible if its Lipschitz constant is lower than 1, meaning that invertible ResNets both possess highly-informative representations and satisfy distance-awareness. However, note that this is not a necessary condition for invertibility, and thus information preservation.

3.2. Uncertainty Estimation

There are two directions regarding uncertainty estimation in DUMs - generative and discriminative approaches. While generative approaches use the likelihood produced by an explicit generative model of the distribution of hidden representations as a proxy for uncertainty, discriminative methods directly use the predictions based on regularized representations to quantify uncertainty.

Generative approaches estimate the distribution of hidden representations post-training or end-to-end, and use the likelihood as an uncertainty proxy. Wu *et al.* (Wu & Goodman, 2020) propose a method to estimate the distribution in the feature space, where the variance of the distribution is used as a confidence measure. MIR (Postels et al., 2020), DDU (Mukhoti et al., 2021) and DCU (Winkens et al., 2020) fit a class-conditional GMM to their regularized hidden representations and use the log-likelihood as an epistemic uncertainty proxy. DEUP (Jain et al., 2021) uses the log-likelihood of a normalizing flow in combination with an aleatoric uncertainty estimate to predict the generalization error. A special instance of the generative approaches are INNs as they directly estimate the training data distribution. The likelihood of the input data is used as a proxy for uncertainty. While this idea is appealing, it can lead to training difficulties, imposes strong constraints on the underlying model and still remains susceptible to OOD data (Nalisnick et al., 2018). PostNet (Charpentier et al., 2020) is a hybrid approach which estimates the distribution of hidden representations of each class using a separate NF which is learned in an end-to-end fashion. Its log-likelihoods parameterize a Dirichlet distribution. We categorize PostNet as a generative approach since their epistemic uncertainty is the log-likelihood of the NF associated with the predicted class.

Discriminative approaches use the predictive distribution to quantify uncertainty. Mandelbaum *et al.* (Mandelbaum & Weinshall, 2017) propose to use a Distance-based Confidence Score (DCS) learning a centroid for each class end-to-end. Similarly, DUQ (Van Amersfoort et al., 2020) builds on Radial Basis Function (RBF) networks (LeCun et al., 1998) and proposes a novel centroid updating scheme. Both estimate uncertainty as the distance between the model output and the closest centroid. DUMs adopting SN (Liu et al., 2020; van Amersfoort et al., 2021) (preserving L_2 distances) typically replace the softmax layer with Gaussian processes (GPs) with RBF kernel, extending distance awareness to the output layer. In particular, SNGP (Liu et al., 2020) relies on a Laplace approximation of the GP based on the random Fourier feature (RFF) expansion of the GP posterior (Rasmussen, 2003). DUE (van Amersfoort et al., 2021) uses the inducing point approximation (Titsias, 2009; Hensman et al., 2015), incorporating a large number of inducing points without overfitting (Burt et al., 2019). The uncertainty is derived

as the Dempster-Shafer metric (Liu et al., 2020), resp. the softmax entropy (van Amersfoort et al., 2021).

4. Evaluation of Deterministic Epistemic Uncertainty

We investigate whether a deterministic treatment of the weights of a NN as proposed by DUMs not only detects OOD well but also yields well calibrated epistemic uncertainty, and scales to realistic vision tasks. Therefore, our experiments are comprised of two parts. Firstly, we evaluate DUMs on image classification, where we measure their calibration under synthetic corruptions (Sec. 4.1.1) and sensitivity to their regularization strength. We also evaluate DUMs on OOD detection in Sec. A.3.2. Then, we extend DUMs to a large-scale dense prediction task - semantic segmentation (Sec. 4.2) - where we evaluate their calibration on synthetic corruptions (Sec. 4.2.1) based on Cityscapes as well as on more realistic distributional shifts (Sec. 4.2.2) based on data collected in the simulation environment CARLA (Dosovitskiy et al., 2017).

Baselines. We compare DUMs with two baselines for epistemic uncertainty - Monte-Carlo (MC) dropout (Gal & Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017). Moreover, we report the softmax entropy of a vanilla NN as a simple baseline. We refer to the supplement for details on uncertainty estimation in our baselines. Note, that the softmax entropy is expected to yield suboptimal calibration under distributional shifts since it quantifies aleatoric uncertainty while adding no computational overhead.

Methods. We evaluate DUQ (Van Amersfoort et al., 2020), SNGP (Liu et al., 2020) and DDU (Mukhoti et al., 2021) as representatives of distance-awareness, since these cover both techniques - SN and gradient penalty - and apply different techniques for uncertainty estimation. We exclude DUE (van Amersfoort et al., 2021) since it provides limited additional insights given SNGP. Moreover, we exclude DCS (Mandelbaum & Weinshall, 2017) since it only leads to a marginal improvement in their own experiments and their contrastive loss only operates on class centroids and, thus, is not expected to lead to distance awareness within clusters. Furthermore, we evaluate MIR (Postels et al., 2020), DCU (Winkens et al., 2020) and PostNet (Charpentier et al., 2020) as representatives of informative representations. However, we do not scale DCU and PostNet to semantic segmentation. DCU with its constrastive pretraining based on SimCLR (Chen et al., 2020) is computationally too demanding due to large batch sizes. PostNet does not scale to semantic segmentation due to instabilities arising from the end-to-end training of the NF for learning the distribution of hidden representations. They require a small hidden dimension ($\leq$10) which already leads to poor testset performance on CIFAR100. Further, we do not evaluate methods based on invertible neural networks (Nalisnick et al., 2019; Ardizzone et al., 2020) since they 1) enforce strict constraints on the underlying architecture (*e.g.* fixed dimensionality of hidden representations) and often lead to training instabilities.

Calibration metrics. Typical calibration metrics are Expected Calibration Error (ECE) (Naeini et al., 2015) and Brier score (BRIER, 1950). However, since most DUMs, except SNGP, do not provide uncertainty in form of a probabilistic forecast, we cannot rely on measuring the calibration of probabilities. Thus, we will exploit another desired property of uncertainty to quantify calibration, namely the ability to distinguish correct from incorrect predictions. In fact, this property is a relaxation of calibrated probabilities, as it is independent of the absolute value of uncertainty estimates. It solely relies on the ability of an uncertainty estimate to sort predictions according to their correctness. We assess the calibration of uncertainty estimates under distributional shifts using two metrics. Firstly, we report the *Area Under the Receiver Operating Characteristic (AUROC)* obtained when separating correct and incorrect predictions based on uncertainty. Moreover, we introduce a new metric, *Relative Area Under the Lift Curve (rAULC)*, based on the Area Under the Lift Curve (AULC) (Vuk & Curk, 2006). The AULC is obtained by ordering the predictions according to increasing uncertainty and plotting the performance of all samples with an uncertainty value smaller than a certain quantile of the uncertainty against the quantile itself.

Formally, given a set of uncertainty quantiles $q_i \in [0, 1]$, $i \in [1, ..., S]$, with some quantile step width $0 < s < 1$ and the function $F(q_i)$ which returns the accuracy of all samples with uncertainty $u < q_i$, the AULC is defined as $AULC = -1 + \sum_{i \in [1,...,S]} s \frac{F(q_i)}{F_R(q_i)}$. Here, $F_R(\cdot)$ refers to a baseline uncertainty estimate that corresponds to random guessing. We subtract 1 to shift the performance of the random baseline to zero. Note, if an uncertainty estimate is anti-correlated with a models' performance, this score can also be negative. To alleviate a bias towards better performing models, we further compute the rAULC by dividing the AULC by the AULC of a hypothetical (optimal) uncertainty estimation that perfectly orders samples according to model performance. In classification we measure AUROC and rAULC on the image-level, in semantic segmentation on the pixel-level. In all experiments we set the quantile step width to $s = \frac{1}{N}$, where N is the number of predictions.

We compute AUROC and rAULC on continuous distributional shifts 1) across all severities of distributional shifts (including the clean testset) and 2) for each severity separately. We use the former method to establish a quantitative comparison among the methods and the latter to depict the calibration evolution qualitatively as a function of the distributional shift's severity.

4.1. Image Classification

Datasets. We train DUMs on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2014) and evaluate on the corrupted versions of their test set CIFAR10/100-C (Hendrycks & Dietterich, 2019) (Sec. 4.1.1). These include 15 synthetic corruptions, each with 5 levels of severity. Moreover, we explore how sensitive the calibration of DUMs is to the choice of regularization strength on MNIST (LeCun, 1998) and FashionMNIST (Xiao et al., 2017)).

Models and optimization. Each method shares the same backbone architecture and uses a method-specific prediction head. When training on CIFAR-10/100, the backbone architecture is a ResNet-50 (He et al., 2016) For the experiments regarding hyperparameter sensitivity on MNIST and Fashion-MNIST, we employ a multilayer perceptron (MLP) as feature extractor with 3 hidden layers of 100 dimensions each and ReLU activation functions. Each DUM has a hyperparameter for the regularization of its hidden representations. We choose the hyperparameter such that it minimizes the validation loss. All results are averaged over 5 independent runs. The standard deviation and optimization details can be found in the supplement where not present in the main paper. Moreover, we provide per-sample training and inference runtimes for each method in the supplement.

4.1.1. Continuous Distributional Shifts

Tab. 2 reports testset accuracy and calibration for the baselines and DUMs. AUROC and rAULC are computed for each corruption across all severity levels, then averaged over all corruptions. Further, Fig. 1 depicts our metrics depending on the severity of corruptions. We generally observe that ensembles and MC dropout demonstrate best performance in terms of calibration. Further, SNGP is the only DUM that consistently outperforms the softmax entropy. Overall we observe that DUMs using the distribution of hidden representations for estimating epistemic uncertainty yield worse calibration. Among these we find that regularizing hidden representations by enforcing distance awareness (DDU) yields the worst calibration. The superior performance of DCU (Winkens et al., 2020) in terms of testset accuracy originates from their extensive contrastive pretraining which includes extensive data augmentation and a prolonged training schedule. We note that DUQ (Van Amersfoort et al., 2020) did not converge on CIFAR100 due to training instabilities. These arise from maintaining the class centroids, which become very noisy for 100 classes with only 600 samples per class.

Moreover, we report OOD detection performance of models used in Tab. 2 and Fig. 1 in the supplement. Interestingly, despite competitive performance on OOD detection DUMs fall short in terms of uncertainty calibration compared to MC dropout and deep ensembles.

4.1.2. Sensitivity to Hyperparameters

We are interested in the impact of the regularization strength on the uncertainty calibration. Therefore, we train DUQ (Van Amersfoort et al., 2020), SNGP (Liu et al., 2020), MIR (Postels et al., 2020) and DDU (Mukhoti et al., 2021) using various regularization strengths on MNIST and evaluate on continuously shifted data by rotating from 0 to 180 degrees in steps of 20 degrees. Fig. 2 depicts the test accuracy against the rAULC for various regularization strengths. Only for MIR we observe a clear, positive correlation between regularization strength and calibration. Moreover, Tab. 3 reports the corresponding Pearson/Spearman correlation coefficients. The supplement depicts similar results on FashionMNIST (Xiao et al., 2017) as well as for OOD detection performance.

4.2. Semantic Segmentation

This section evaluates whether DUMs seamlessly scale to realistic vision tasks and compares their behaviour under synthetic and realistic continuous distributional shifts with the softmax entropy, MC dropout and ensembles. Therefore, we apply MIR (Postels et al., 2020), SNGP (Liu et al., 2020) and DDU (Mukhoti et al., 2021) to semantic segmentation. Note that DUQ (Van Amersfoort et al., 2020) did not converge on this task.

We consider semantic segmentation as a multidimensional classification problem, where each pixel of the output mask represents an independent classification problem. Given an image $\mathbf{x}$ with n pixels $\mathbf{y} = \{y_1, \cdots, y_n\}$, the predictive distribution factorizes according to $p(\mathbf{y} \mid \mathbf{x}) = p(y_1 \mid \mathbf{x})p(y_2 \mid \mathbf{x}) \cdots p(y_n \mid \mathbf{x})$. We evaluate the calibration of the pixel-level uncertainty in our experiments.

Datasets. We evaluate on synthetic distributional shifts using a corrupted version of Cityscapes (Cordts et al., 2016) (Cityscapes-C (Michaelis et al., 2019)) which contains the same corruptions as CIFAR10/100-C. To further benchmark DUMs in a realistically and continuously changing environment, we collect a synthetic dataset for semantic segmentation. We use the CARLA Simulator (Dosovitskiy et al., 2017) and leverage the SHIFT dataset (Sun et al., 2022) toolkit for rendering images and segmentation masks under controlled distributional shifts in a driving scenario. The classes definition is aligned with the CityScape dataset (Cordts et al., 2016). Training data is collected from four towns in CARLA. We produce 32 sequences from each town. Vehicles and pedestrians are randomly generated for each sequence. Every sequence has 500 frames with a sampling rate of 10 FPS. We uniformly sample a validation set. We introduce continuous distributional shifts by varying the time-of-the-day and weather conditions (visual examples

Table 2. We compare Softmax, MC Dropout (Gal & Ghahramani, 2016), Deep Ensembles, SNGP, DDU, MIR, DUQ, DCU and PostNet on CIFAR10/100-C. We evaluate the accuracy (ACC) on the uncorrupted testset, AUROC and rAULC. Ensembles and MC dropout demonstrate better uncertainty calibration than most DUMs. Only SNGP consistently outperforms the softmax entropy. DCU's superior performance is expected since it uses expensive contrastive pretraining. DUQ did not converge on CIFAR100-C due to training instabilities arising from dynamically updated cluster centroids.

Method	CIFAR10-C			CIFAR100-C		
	ACC	AUROC	rAULC	ACC	AUROC	rAULC
Softmax entropy	0.882	0.782	0.708	0.610	0.762	0.596
MC Dropout (Gal & Ghahramani, 2016)	0.885	**0.866**	0.829	0.615	0.818	**0.726**
Ensemble (Lakshminarayanan et al., 2017)	0.910	0.85 0	**0.833**	0.628	**0.824**	0.713
SNGP (Liu et al., 2020)	0.903	0.833	0.766	0.611	0.788	0.623
DDU (Mukhoti et al., 2021)	0.884	0.673	0.441	0.609	0.635	0.339
MIR (Postels et al., 2020)	0.889	0.79	0.697	0.617	0.726	0.514
DUQ (Van Amersfoort et al., 2020)	0.860	0.773	0.614	-	-	-
DCU (Winkens et al., 2020)	**0.945**	0.794	0.706	**0.642**	0.750	0.558
PostNet (Charpentier et al., 2020)	0.882	0.784	0.676	0.520	0.743	0.603

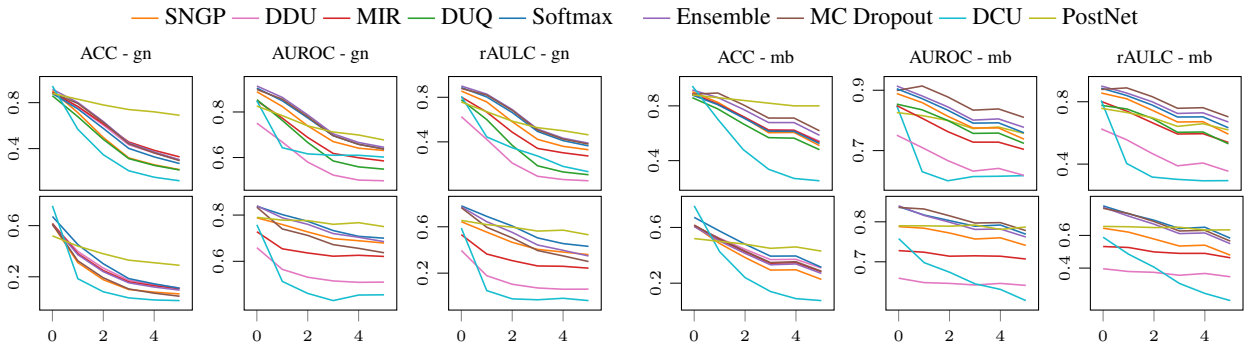


Figure 1. Softmax entropy, ensembles (Lakshminarayanan et al., 2017), MC dropout (Gal & Ghahramani, 2016), DUQ (Van Amersfoort et al., 2020), SNGP (Liu et al., 2020), MIR (Postels et al., 2020), DDU (Mukhoti et al., 2021) and DCU (Winkens et al., 2020) on CIFAR10-C (upper row) and CIFAR100-C (lower row) (Hendrycks & Dietterich, 2019). We show the accuracy, AUROC and rAULC on the corruptions gaussian_noise (gn) and motion_blur (mb) against the corruption severity. While all methods, except DCU, demonstrate a similar accuracy, DUMs - in particular methods based on generative modeling of hidden representations - yield worse calibration. DUQ did not converge on CIFAR100. Other corruptions are included in the supplement.

Table 3. Quantitative correlation between rAULC and regularization strength using Pearson/Spearman's rank correlation coefficient. MIR (Postels et al., 2020) demonstrates the largest correlation.

Metric	DUQ	DDU	SNGP	MIR
Pearson	-0.83	-0.30	0.58	0.76
Spearman	-0.83	-0.43	0.61	0.96

and details on data collection are in the supplement). The time-of-the-day is parameterized by the sun's altitude angle, where $90°$ means mid-day (training data) and the $0°$ means dust/dawn. We produce samples with altitude angles from $90°$ to $15°$ by steps of $5°$, and $15°$ to $-5°$, where the environment changes sharply, in $1°$ steps. In order to continuously change the weather conditions, we increase the magnitude of the rain in four steps (see supplement for visual examples). We refer to this dataset as CARLA-C.

Backbone. We adopt Dilated ResNet (DRN) (Yu & Koltun, 2016; Yu et al., 2017) as semantic segmentation backbone since it is based on residual connections allowing the use SN. Using dilated convolutions it improves spatial accuracy, achieving satisfactory results on CityScapes (Cordts et al., 2016). We adopt the variant DRN-A-50. All results are averaged across 5 independent repetitions.

SNGP. DRN uses 1×1 convolutions at the last layer to map the latest feature map to the predicted segmentation mask. This works under the assumption that all pixels in the output mask are i.i.d. random variables. Following this intuition, we extend SNGP to semantic segmentation by fitting a $GP : \mathbb{R}^Z \to \mathbb{R}^C$ at pixel level that maps from the

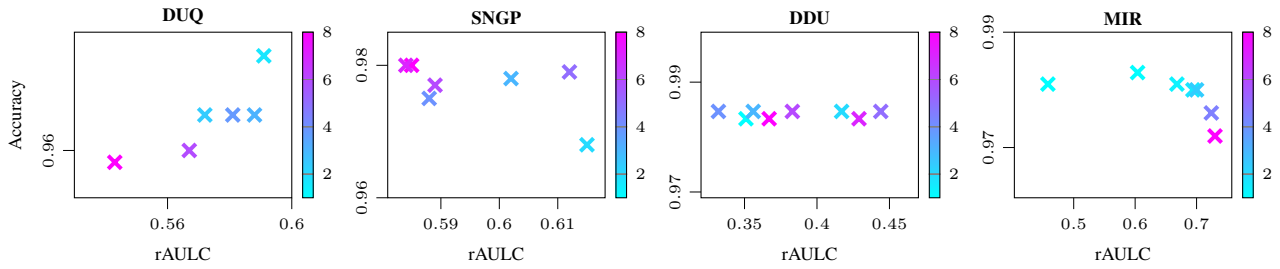


Figure 2. We analyze the sensitivity of DUQ (Van Amersfoort et al., 2020), SNGP (Liu et al., 2020), MIR (Postels et al., 2020) and DDU (Mukhoti et al., 2021) to their regularization strength. Therefore, we train models on MNIST and evaluate on continuously shifted data by rotating from 0 to 180 degrees in steps of 20 degrees. We plot the accuracy on the unperturbed testset against the rAULC computed using data from all levels of perturbation. Only for MIR we observe a clear, positive correlation between regularization strength and calibration. For DDU and SNGP, smaller regularization parameter denotes stronger regularization.

Table 4. We compare semantic segmentation using Softmax, MC Dropout (Gal & Ghahramani, 2016), Deep Ensembles, SNGP, DDU and MIR on Cityscapes-C and CARLA-C. We evaluate the mean Intersection over Union (mIoU) on the uncorrupted testset and AUROC/rAULC across all levels of corruption. Again, ensembles and MC dropout yield better calibrated uncertainty than most DUMs. Most notably, only SNGP consistently outperforms the softmax entropy. DUMs using an explicit generative model of hidden representations to estimate uncertainty perform particularly bad on realistic distributional shifts (CARLA-C).

Method	**Cityscapes-C**			**CARLA-C**		
	mIoU	AUROC	rAULC	mIoU	AUROC	rAULC
Softmax	0.503	0.815	0.737	0.422	0.854	0.818
MC Dropout (Gal & Ghahramani, 2016)	0.506	**0.846**	**0.785**	0.410	0.843	0.730
Ensemble (Lakshminarayanan et al., 2017)	**0.525**	0.835	0.751	**0.428**	**0.863**	0.812
SNGP (Liu et al., 2020)	0.519	0.833	0.759	0.424	0.853	**0.813**
DDU (Mukhoti et al., 2021)	0.505	0.731	0.542	0.408	0.467	-0.038
MIR (Postels et al., 2020)	0.504	0.729	0.564	0.412	0.744	0.619

deep feature dimension z to the number of classes c. By keeping the GP kernel parameters shared across all pixels, we simulate a 1×1 convolutional GP, *i.e.* $\sigma : (H_l \times W_l \times Z) \rightarrow (H_l \times W_l \times C)$, where σ convolves the GP, H_l and W_l are, respectively, feature map height and width at layer l, Z is the number of latent features and C is the number of output classes. For details about the GP we refer to (Liu et al., 2020) or the supplement.

MIR and DDU require fitting the distribution of hidden representations. We fit a Gaussian mixture model (GMM) with 20 components (*i.e.* number of classes) to each spatial location of the hidden representations using features extracted from the training data independently. This assumes that the distribution is translation invariant and factorizes along the spatial dimensions of the latent space. Pixel-level uncertainties are then computed using bi-cubic interpolation following a similar procedure as the one proposed in (Blum et al., 2019). We refer to the supplement for more details.

4.2.1. Cityscapes Corrupted

We evaluate the softmax entropy, ensembles (Lakshminarayanan et al., 2017), MC dropout (Gal & Ghahramani, 2016), SNGP (Liu et al., 2020), MIR (Postels et al., 2020), and DDU (Mukhoti et al., 2021) on Cityscapes-C (Michaelis et al., 2019). Tab. 4 depicts mean Intersection over Union (mIoU) and calibration performance in terms of AUROC and rAULC. Ensembles and MC dropout yield the best calibration, while among DUMs only SNGP consistently ourperforms the softmax entropy. A qualitative visualization of the AUROC and rAULC depending on the corruption strength is in the supplement (Fig. 15, Fig. 16, and Fig. 17).

4.2.2. Realistic Continuous Distributional Shifts

Similarly, we evaluate the softmax entropy, ensembles (Lakshminarayanan et al., 2017), MC dropout (Gal & Ghahramani, 2016), SNGP (Liu et al., 2020), MIR (Postels et al., 2020), and DDU (Mukhoti et al., 2021) on CARLA-C. Tab. 4 depicts mIoU and calibration performance in terms of AUROC and rAULC. Ensembles yield the best calibration. Among DUMs only SNGP consistently ourperforms the softmax entropy which is in line with the results on image classification (Sec. 4.1.1). Further, we show the qualitative behaviour of AUROC and rAULC depending on the

corruption strength in the supplement (Fig. 18).

5. Conclusion & Discussion

This work investigates the shortcomings of a recent trend in research on deterministic epistemic uncertainty estimation. To this end, we provide the first taxonomy of DUMs. Moreover, we verify that DUMs indeed scale to realistic vision tasks in terms of predictive performance. However we find that (i) the epistemic uncertainty of many DUMs is not well calibrated under distributional shifts and (ii) the regularization strength in methods based on distance awareness does not correlate strongly with OOD detection and calibration.

DUMs recently showed good OOD detection performance and are interesting for practical applications in need of efficient uncertainty quantification. We established that DUMs mainly differ in their regularization technique for countering feature collapse and their approach to quantifying uncertainty (Sec. 3).

Regarding the calibration under continuous distributional shifts of DUMs, we observe that such uncertainty estimates are considerably worse calibrated than scalable Bayesian methods. This is the case for image classification (Sec. 4.1.1) as well as semantic segmentation (Sec. 4.2) and for synthetic as well as more realistic distributional shifts. SNGP (Liu et al., 2020) denotes the only DUM that consistently yields better calibrated uncertainties under continuous distributional shifts than the softmax entropy. Simultaneously, SNGP is the only DUM which derives their uncertainty from its predictive distribution.

In particular, we find that methods relying on the distribution of hidden representations to quantify uncertainty (Winkens et al., 2020; Postels et al., 2020; Mukhoti et al., 2021) are poorly calibrated. It is understandable that these methods are worse calibrated than SNGP since they do not take into account the predictive distribution. They assume that locations in the feature space entail information about the correctness of predictions. While this is arguably true, features also contain additional information that renders them sub-optimal for judging the correctness of predictions due to ambiguities. Overall, this underlines the necessity to refrain from DUMs that purely rely on distances or log-likelihoods in the feature space when well calibrated uncertainties are required.

Interestingly, one may argue that DUMs simply do not scale to realistic vision tasks and, for that reason, are not well calibration in Sec. 4.2. However, note that DUMs in fact demonstrate strong predictive performance in such scenarios (see Tab. 4) and have been shown to perform well on OOD detection on realistic vision datasets (*e.g.* pixel-wise anomaly detection (Blum et al., 2019)). Therefore, we can conclude that this is not a problem of scaling DUMs to realistic scenarios but rather an inherent problem which these types of uncertainty estimates.

Moreover, another desirable property of DUMs would be that the strength of the feature space regularization correlates with the quality of the uncertainty - both in terms of calibration as well as OOD detection. Due to the original purpose of most DUMs, this would be at least expected for OOD detection. However, we do not observe this for bi-Lipschitz regularization (Sec. 4.1.1). We hypothesize that this originates from the fact that these regularization techniques rely on an underlying distance metric - *i.e.* L_2 distance. Such distance metrics are not meaningful in the case of high-dimensional data, *i.e.* images. We hope that our findings will foster future research on making these promising family of methods better calibrated and more broadly applicable.

6. Acknowledgement

This work was partially supported by Google and by the Max Planck ETH Center for Learning Systems.

References

Alemi, A. A., Fischer, I., and Dillon, J. V. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

Ardizzone, L., Mackowiak, R., Rother, C., and Köthe, U. Training normalizing flows with the information bottleneck for competitive generative classification. *Advances in Neural Information Processing Systems*, 33, 2020.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279*, 2021.

Behrmann, J., Grathwohl, W., Chen, R., Duvenaud, D., and Jacobsen, J. Invertible residual networks. arxiv e-prints. *arXiv preprint arXiv:1811.00995*, 2018.

Bishop, C. M. Mixture density networks. 1994.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24: 2178–2186, 2011.

Blum, H., Sarlin, P.-E., Nieto, J., Siegwart, R., and Cadena, C. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *arXiv preprint arXiv:1904.03215*, 2019.

BRIER, G. W. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Burt, D., Rasmussen, C. E., and Van Der Wilk, M. Rates of convergence for sparse variational gaussian process regression. In *International Conference on Machine Learning*, pp. 862–871. PMLR, 2019.

Charpentier, B., Zügner, D., and Günnemann, S. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33:1356–1367, 2020.

Charpentier, B., Borchert, O., Zügner, D., Geisler, S., and Günnemann, S. Natural posterior network: Deep bayesian predictive uncertainty for exponential family distributions. *arXiv preprint arXiv:2105.04471*, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

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

Daunizeau, J. Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables. *arXiv preprint arXiv:1703.00091*, 2017.

Der Kiureghian, A. and Ditlevsen, O. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792. PMLR, 2020.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1183–1192. JMLR. org, 2017.

Gasperini, S., Haug, J., Mahani, M.-A. N., Marcos-Ramiro, A., Navab, N., Busam, B., and Tombari, F. Certainnet: Sampling-free uncertainty estimation for object detection. *IEEE Robotics and Automation Letters*, 7(2):698–705, 2021.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Gustafsson, F. K., Danelljan, M., and Schon, T. B. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 318–319, 2020.

Haußmann, M., Hamprecht, F. A., and Kandemir, M. Sampling-free variational inference of bayesian neural networks by variance backpropagation. In *Uncertainty in Artificial Intelligence*, pp. 563–573. PMLR, 2020.

Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., Dai, A. M., and Tran, D. Training independent subnetworks for robust prediction. In

International Conference on Learning Representations, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pp. 351–360. PMLR, 2015.

Hinton, G. E. and Van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.

Jacobsen, J.-H., Smeulders, A., and Oyallon, E. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

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

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5580–5590, 2017.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pp. 2575–2583, 2015.

Krizhevsky, A., Nair, V., and Hinton, G. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55:5, 2014.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.

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

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J., Humt, M., Feng, J., and Triebel, R. Estimating model uncertainty of neural networks in sparse information form. In *International Conference on Machine Learning*, pp. 5702–5713. PMLR, 2020.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Conference on Neural Information Processing Systems*, 2020.

Loquercio, A., Segu, M., and Scaramuzza, D. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020.

Mandelbaum, A. and Weinshall, D. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844*, 2017.

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

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P. H., and Gal, Y. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv preprint arXiv:2102.11582*, 2021.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2018.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pp. 4723–4732. PMLR, 2019.

Neal, R. M. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Oberdiek, P., Rottmann, M., and Gottschalk, H. Classification uncertainty of deep neural networks based on gradient information. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pp. 113–125. Springer, 2018.

Obukhov, A., Rakhuba, M., Liniger, A., Huang, Z., Georgoulis, S., Dai, D., and Van Gool, L. Spectral tensor train parameterization of deep learning layers. In *International Conference on Artificial Intelligence and Statistics*, pp. 3547–3555. PMLR, 2021.

Oh, S. J., Murphy, K., Pan, J., Roth, J., Schroff, F., and Gallagher, A. Modeling uncertainty with hedged instance embedding. *Proceedings of the International Conference on Learning Representations*, 2019.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Osband, I., Wen, Z., Asghari, M., Ibrahimi, M., Lu, X., and Van Roy, B. Epistemic neural networks. *arXiv preprint arXiv:2107.08924*, 2021.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Postels, J., Ferroni, F., Coskun, H., Navab, N., and Tombari, F. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2931–2940, 2019.

Postels, J., Blum, H., Strümpler, Y., Cadena, C., Siegwart, R., Van Gool, L., and Tombari, F. The hidden uncertainty in a neural networks activations. *arXiv preprint arXiv:2012.03082*, 2020.

Rame, A., Sun, R., and Cord, M. Mixmo: Mixing multiple inputs for multiple outputs via deep subnetworks. *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Ritter, H., Botev, A., and Barber, D. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

Rosca, M., Weber, T., Gretton, A., and Mohamed, S. A case for new neural network smoothness constraints. *arXiv preprint arXiv:2012.07969*, 2020.

Rupprecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., and Hager, G. D. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3591–3600, 2017.

Sedghi, H., Gupta, V., and Long, P. M. The singular values of convolutional layers. *arXiv preprint arXiv:1805.10408*, 2018.

Segù, M., Tonioni, A., and Tombari, F. Batch normalization embeddings for deep domain generalization. *arXiv preprint arXiv:2011.12672*, 2020.

Sharma, A., Azizan, N., and Pavone, M. Sketching curvature for efficient out-of-distribution detection for deep neural networks. *arXiv preprint arXiv:2102.12567*, 2021.

Singla, S. and Feizi, S. Bounding singular values of convolution layers. *arXiv preprint arXiv:1911.10258*, 2019.

Smith, L., van Amersfoort, J., Huang, H., Roberts, S., and Gal, Y. Can convolutional resnets approximately preserve input distances? a frequency analysis perspective. *arXiv preprint arXiv:2106.02469*, 2021.

Snoek, J., Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J., Ren, J., and Nado, Z. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

Sun, T., Segù, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., and Yu, F. SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation. In *Computer Vision and Pattern Recognition*, 2022.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Teye, M., Azizpour, H., and Smith, K. Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning*, pp. 4907–4916, 2018.

Titsias, M. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pp. 567–574. PMLR, 2009.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. Improving deterministic uncertainty estimation in deep learning for classification and regression. *arXiv preprint arXiv:2102.11409*, 2021.

Vuk, M. and Curk, T. Roc curve, lift chart and calibration plot. *Metodoloski zvezki*, 3(1):89, 2006.

Wen, Y., Tran, D., and Ba, J. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

Wenzel, F., Roth, K., Veeling, B., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. How good is the bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pp. 10248–10259. PMLR, 2020.

Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

Wu, M. and Goodman, N. A simple framework for uncertainty in contrastive learning. *arXiv preprint arXiv:2010.02038*, 2020.

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

Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

Yu, F., Koltun, V., and Funkhouser, T. Dilated residual networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pp. 5852–5861. PMLR, 2018.

A. Appendix

We here provide additional theoretical background, implementation and optimization details, additional results, comparisons and ablation studies. In particular, we report additional details on the theoretical background necessary to understand DUMs' design choices in Sec. A.1. We describe techniques used for enforcing Lipschitz constraints Sec. A.1.1 and informative representations Sec. A.1.2 in more detail. Moreover, Sec. A.1.3 summarizes common choices of uncertainty estimation techniques for discriminative and generative DUMs.

Sec. A.3.2/Sec. A.3.3 show additional results for image classification/semantic segmentation. We report optimization/implementation details in Sec. A.4/Sec. A.5. Furthermore, we describe the quantification of uncertainty for image classification Sec. A.5.3 and semantic segmentation Sec. A.5.4. Finally, we provide details on the data collection process in CARLA and examples from the sequences collected for semantic segmentation Sec. A.6.

A.1. DUMs - Fundamentals

Subsequently, we provide a more detailed introduction of the most common concepts applied by DUMs. Furthermore, we discuss each DUM individually in light of the introduced taxonomy and highlight its strengths and weaknesses. Initially (Sec. A.1.1, Sec. A.1.2 and Sec. A.1.3) maintain a modularized perspective of DUMs by describing individual components including their advantages and disadvantages. Subsequently (Sec. A.1.4), we shed light on each DUMs individually using insights from the modularized considerations.

In particular, We describe regularization techniques, including Lipschitz regularization (Sec. A.1.1) for enforcing distance awareness and informative representations (Sec. A.1.2). Sec. A.1.3 summarizes discriminative and generative approaches to uncertainty estimation in DUMs. Finally, Sec. A.1.4 discusses each individual method used in our empirical comparison.

A.1.1. Regularization Techniques - Distance Awareness

The fundamental idea of distance-aware hidden representations is to avoid feature collapse by enforcing distances between latent representations to mirror distances in the input space. This can be achieved by constraining the Lipschitz constant, as it enforces a lower and an upper bound to expansion and contraction performed by an underlying neural network. More formally, given any pair of inputs x_1 and x_2 the following lower and upper bounds must hold for the resulting activation of a feature extractor f_θ with parameters θ: $c_1||x_1 - x_2||_I \leq ||f_\theta(x_1) - f_\theta(x_2)||_F \leq c_2||x_1 - x_2||_I$. c_1 and c_2 denote respectively the lower and upper bound for the Lipschitz constant, and $||\cdot||_I$ and $||\cdot||_F$ are the chosen metrics in the input and feature space respectively.

Recent proposals have primarily adopted two methods to impose this constraint.

Gradient Penalty. First introduced to regularize the Lipschitz constant in GAN training (Gulrajani et al., 2017), a two-sided gradient penalty is used as an additional loss term to enforce sensitivity of the feature space to changes in the input by DUQ (Van Amersfoort et al., 2020). The gradient penalty is formulated as an additional loss term that regularises the Frobenius norm $||J||_F$ of the Jacobian J of a NN to enforce a bi-Lipschitz constraint. Therefore, the training loss of a NN is typically enhanced with the absolute difference between $||J||_F$ and some chosen positive constant.

Given a model g and an input x, regularising the Frobenius norm $||J||_F$ of its Jacobian J constraints its Lipschitz constant. Therefore, the following two-sided gradient penalty is used: $\lambda \left[||\nabla_x g(x)||_F - 1\right]^2$, where λ is the regularization strength, $||\cdot||_2$ is the L_2 norm, the target bi-Lipschitz constant is 1. For more details, refer to (Van Amersfoort et al., 2020).

Spectral Normalization. The two-sided gradient penalty described above requires backpropagating through the Jacobian of a NN and is, thus, computationally demanding. A more efficient technique is SN (Miyato et al., 2018). For each layer $g : \mathbf{h}_{in} \rightarrow \mathbf{h}_{out}$, SN normalizes the weights W of each layer using their spectral norm $sn(W)$ to constrain the bi-Lipschitz constant. Thus, weight matrices are normalized according to: $W_{sn} = \frac{W}{c \cdot sn(W)}$. This effectively constrains the layer's Lipschitz norm $||g||_{Lip} = sup_{\mathbf{h}} sn(\nabla g(\mathbf{h}))$, where $sn(A)$ is the spectral norm of the matrix A, which is equivalent to its largest singular value. Consequently, SN normalizes the spectral norm of the weights W of each layer to satisfy the soft-Lipschitz constraint $sn(W) = c$ (hard- if the Lipschitz constant $c = 1$): $W_{sn} = W/sn(W)$. Note, that spectral normalization requires residual layers. We refer to (Liu et al., 2020) for further details.

Runtime. Let N denote the number of parameters of the underlying neural network and B denote the batch size used during training of the underlying discriminative task. Gradient penalty leads to additional runtime/memory cost of $O(NB)$.

This originates from backpropagation through the gradients of the input which essentially doubles the computation during backpropagation. Spectral normalization leads to additional runtime/memory cost of $O(N)$ since its complexity equates to applying the affine layers of a model additionally on a single sample.

Overall, we summarize the advantages and disadvantages of each regularization technique enforcing distance aware representations.

- **Distance awareness (general):**
  - **Advantages:** Can be used in combination with GPs and RBF kernels which both assume distance-aware inputs.
  - **Disadvantages:** Assumes an underlying distance metric (e.g. L_2). This can be unsuitable/problematic for some data distributions (e.g. images). Does not correlate with OOD detection performance.
- **Gradient Penalty:**
  - **Advantages:** Architecture-agnostic.
  - **Disadvantages:** High computational and memory costs due to backpropagation through the input's gradients.
- **Spectral Normalization:**
  - **Advantages:** Computationally more efficient compared to gradient penalty.
  - **Disadvantages:** Not architecture-agnostic. Requires the use of residual layers.

A.1.2. Informative Representations

Unlike approaches enforcing distance aware representations, informative representations do not rely on an underlying distance metric. These approaches rather aim at maximizing the mutual information between input data distribution and the distribution of hidden representations heuristically (Postels et al., 2020) or exactly (Winkens et al., 2020; Charpentier et al., 2020). Subsequently, we discuss the different approaches in greater detail.

Constrastive learning. DCU (Winkens et al., 2020) first pretrains its model using contrastive learning (Chen et al., 2020). Subsequently, they finetune on the actual classification task by training simultaneously on a classification and a contrastive learning objective. This approach provably encourages the model to increase the mutual information between the input distribution and the distribution of hidden representations (Oord et al., 2018). A disadvantage of this approach is the large batch size required for training the contrastive objective. Furthermore, it heavily depends on the underlying data augmentations which need to be tailored to the discriminative task at hand (e.g. classification).

Reconstruction regularization. The authors of MIR (Postels et al., 2020) try to heuristically increase the information content about the input in the hidden representations. Therefore, they require the model to be able to reconstruct its input from its hidden representations using a separate decoder module during training. The entire approach can also be viewed as a constrained autoencoder, where the constraint is the objective of the discriminative task at hand (e.g. classification). While this approach is only a heuristic, it is more agnostic of the underlying discriminative task.

Entropy regularization. PostNet (Charpentier et al., 2020) learns the distribution of hidden representations end-to-end during training of the discriminative model. They parameterize the distribution using one normalizing flow (radial flow) per class. While they do not explicitly mention the problem of feature collapse, their entropy regularization loss fulfills this purpose. In particular, they maximize the entropy of the predicted Dirichlet distribution $D(\alpha^{(i)})$ parameterized by $\alpha^{(i)} = (\alpha_1^{(i)}, \dots, \alpha_c^{(i)})$ for a classification problem with c classes. Each $\alpha_j^{(i)}$ is given by $\alpha_j^{(i)} = \beta^{prior} + N_c P(z^{(i)}|c, \phi)$. Here, β^{prior} denotes a constant prior term shared across classes, N_c denotes the number of occurrences of class c in the training set, $z^{(i)}$ denotes the hidden representation of some input $x^{(i)}$ and $P(z^{(i)}|c, \phi)$ denotes the radial flow associated with class c with parameters ϕ. Importantly, β^{prior} is set to 1 in their experiments which leads to $\alpha_j^{(i)} \geq 1 \forall j \in [1, \dots, c]$. PostNet then encourages large entropies of the Dirichlet distribution during training. The entropy is given by

$$H(D(\alpha^{(i)})) = \log(B(\alpha^{(i)})) + (\alpha_0^{(i)} - c)\psi(\alpha_0^{(i)}) - \sum_{j=0}^{c}(\alpha_j^{(i)} - 1)\psi(\alpha_j^{(i)}) \tag{1}$$

where B is there beta-function, ψ is the Digamma function and $\alpha_0^{(i)} = \sum_{j=1}^{c} \alpha_j^{(i)}$. Importantly, this function has a global maximum for $\alpha_j^{(i)} = 1 \forall j \in [1, \dots c]$. Since $\beta^{prior} = 1$, this term further encourages the normalizing flows to produce

likelihoods close to zero. Thus, it encourages large values of the negative log-likelihoods and, consequently, entropies under each radial flow.

Runtime. Let N denote the number of parameters of the underlying neural network and B denote the batch size used during training of the underlying discriminative task. Contrastive learning leads to additional runtime/memory cost of $O(N)$. This originates from the fact that it requires large batch sizes and thus likely increases the batch size compared to the original discriminative task. Reconstruction regularization (Postels et al., 2020) and entropy regularization (Charpentier et al., 2020) lead to additional runtime/memory cost of $O(B)$, since the size of the decoder model in reconstruction regularization (Postels et al., 2020), and resp. the normalizing flows in entropy regularization (Charpentier et al., 2020), are in principal independent of the size of the original model.

Overall, we summarize the advantages and disadvantages of each regularization technique enforcing informative representations.

- **Informative representations (general):**
  - **Advantages:** Does not rely on an underlying distance metric.
  - **Disadvantages:** When paired with generative modeling of hidden representations, strong regularization is expected to show similar pathologies as explicit generative models trained directly on the data distribution (Nalisnick et al., 2018). Moreover, it cannot be paired with RBF kernels and GPs approximations based on RBF kernels, since they work under the assumption that also the feature extractor is a distance-preserving function.
- **Contrastive Learning:**
  - **Advantages:** Is shown to simultaneously boost predictive performance (Chen et al., 2020; Winkens et al., 2020) - particularly on classification. Architecture-agnostic. Provably maximizes mutual-information between input distribution and distribution of hidden representations (Oord et al., 2018).
  - **Disadvantages:** High computational and memory costs due to the necessity of large batch sizes. Contrastive learning needs to be tailored (e.g. data augmentations) to the underlying discriminative task.
- **Reconstruction regularization:**
  - **Advantages:** Architecture-agnostic.
  - **Disadvantages:** Only heuristically maximizes mutual information between input distribution and distribution of hidden representations.
- **Entropy regularization:**
  - **Advantages:** Assuming a deterministic neural network, enforcing large entropy in the latent space equates maximizing mutual-information between input distribution and distribution of hidden representations.
  - **Disadvantages:** Not architecture-agnostic, since it requires Batch Normalization prior to entropy regularized hidden representations for training stabilization. Further, requires low-dimensional hidden representations.

A.1.3. Uncertainty Estimation

Training a feature extractor under the regularization constraints imposed by distance awareness (Sec. 3.1.1, Sec. A.1.1) or representation informativeness (Sec. 3.1.2) allows to leverage intermediate representations to quantify uncertainty over network's predictions. Extending Sec. 3.2, we here distinguish between *generative* and *discriminative* approaches to uncertainty quantification in DUMs and provide a detailed categorization of such techniques.

Generative approaches. Given a model trained under some above-discussed regularization constraint, generative approaches estimate the distribution of hidden representations by fitting density models on the regularized feature space, and use the likelihood as uncertainty metric to detect OOD samples. MIR (Postels et al., 2020), DDU (Mukhoti et al., 2021) and DCU (Winkens et al., 2020) learn the density of hidden representations post-training based on the features observed on the training data. In contrast, PostNet (Charpentier et al., 2020) learns the density model end-to-end with the underlying discriminative model.

Predominantly, class-conditional GMMs are fitted on the regularized intermediate feature space to estimate its distribution, as done in MIR (Postels et al., 2020), DDU (Mukhoti et al., 2021) and DCU (Winkens et al., 2020) - i.e. one multi-variate

gaussian per class, to the hidden representations post training. Subsequently, log-likelihood (Postels et al., 2020) or the log-likelihood of the mixture component associated with the predicted class (Mukhoti et al., 2021; Winkens et al., 2020) is used as a proxy for epistemic uncertainty.

On the other hand, PostNet (Charpentier et al., 2020) learns the class-conditional distribution of hidden representations end-to-end using normalizing flows (in particular radial flows). They learn one normalizing flow per class. In their work the class-conditional distribution is used to parameterize a Dirichlet distribution. Following PostNet's notation, the parameters $\alpha^{(i)}$ of the Dirichlet distribution associated with a particular sample $x^{(i)}$ are given by $\alpha_c^{(i)} = \beta^{prior} + \beta^i$ with $\beta^i = N_c P(z^{(i)}|c, \phi)$. Here, c denotes the class, β^{prior} a constant prior term shared across all classes, N_c the number of samples observed in class c and $P(z^{(i)}|c, \phi)$ (ϕ are the parameters of the normalizing flow) is the probability of observing the hidden representation $z^{(i)}$ given the normalizing flow associated with class c. Ultimately, epistemic uncertainty is then quantified as the maximum alpha among all classes. Thus, the epistemic uncertainty is directly derived from the likelihood of the normalizing flow associated with the predicted class of the NN, which mostly corresponds to the normalizing flow with the maximum likelihood assuming a balanced class distribution. We refer to (Charpentier et al., 2020) for a more detailed treatment.

Empirically, we find generative approaches to show worse calibration. The underlying assumption of generative approaches to uncertainty estimation is that locations in feature space entail information about the correctness of predictions. While this is arguably true, features also contain additional information that which can render them suboptimal for judging the correctness of predictions due to ambiguities.

Discriminative While generative approaches use the likelihood produced by an explicit generative model fit to the distribution of regularized hidden representations to quantify uncertainty, discriminative methods directly rely on the predictions based on regularized representations.

Centroid-based techniques use distances between points in the latent space to parameterize predictions. Centroids are defined with respect to the distribution of the feature space generated by the training set. Mandelbaum *et al.* (Mandelbaum & Weinshall, 2017) propose to use a Distance-based Confidence Score (DCS) to estimate local density at a point as the Euclidean distance in the embedded space between the point and its k nearest neighbors in the training set. Similarly, DUQ (Van Amersfoort et al., 2020) builds on Radial Basis Function (RBF) networks (LeCun et al., 1998), which requires the preservation of input distances in the output space which is achieved using the gradient penalty. The class-specific centroids used in the RBF kernel are maintained as a running mean of the features observed for each class.

Other methods are based on the idea that, since GPs with RBF kernels are distance preserving functions (Liu et al., 2020), they can be combined with regularization techniques that enforce distance awareness of the feature extractor (Liu et al., 2020; van Amersfoort et al., 2021) to obtain an end-to-end distance-aware model. The uncertainty can then be computed at the network's output level as the Dempster-Shafer metric (Liu et al., 2020) or the softmax entropy (van Amersfoort et al., 2021). Based on this intuitive idea, SNGP (Liu et al., 2020) and DUE (van Amersfoort et al., 2021) simply rely on different approximations of the GP, adopting respectively the Laplace approximation based on the random Fourier feature (RFF) expansion of the GP posterior (Rasmussen, 2003) and the inducing point approximation (Titsias, 2009; Hensman et al., 2015). Another minor difference lies in the spectral normalization algorithm, with DUE providing a SN implementation also for batch normalization layers. While both methods rely on spectral-normalized feature extractors, they could in principle be applied together with any distance-preserving regularization technique. For example, the GPs could be placed on top of a feature extractor trained with gradient penalty (Van Amersfoort et al., 2020) to regularize the bi-Lipschtiz constant.

A.1.4. A Method-oriented Perspective on Individual DUMs

Here, we discuss each DUMs in our empirical comparison individually. Furthermore, Tab. 5 provides a comparison of DUMs used in our empirical evaluation regarding properties which are interesting for practitioners. We consider four characteristics: Architecture/task constraints, well calibrated uncertainties and computational/memory overhead. We consider the computational/memory overhead of a method not minimal when it scales with at least $O(N)$ where N denotes the number of parameters of the underlying neural network. This is the case for contrastive learning in DCU (Winkens et al., 2020) due to large batch sizes, the gradient penalty in DUQ (Van Amersfoort et al., 2020) due to backpropagation through the gradients of a neural network's input and spectral normalization in SNGP (Liu et al., 2020) and DDU (Mukhoti et al., 2021).

SNGP (Liu et al., 2020) uses spectral normalization for regularizing hidden representations (see Sec. A.1.1). While this

Method	Architecture constraints	Task constraints	Well-calibrated	Minimal computational/memory cost
SNGP(Liu et al., 2020)	residual layers	classification	✓	✗
DUQ(Van Amersfoort et al., 2020)	-	classification	✗	✗
DDU(Mukhoti et al., 2021)	residual layers	-	✗	✗
DCU(Winkens et al., 2020)	-	classification	✗	✗
MIR(Postels et al., 2020)	-	-	✗	✓
PostNet(Charpentier et al., 2020)	batch normalization	classification	✗	✓

Table 5. Qualitative comparison of different DUMs used in our empirical comparison. We are interested in four characteristics that are interested from a practical perspective: Architecture/task constraints, well calibrated uncertainties and computational/memory overhead. We consider the computational/memory overhead of a method not minimal when it scales with at least $O(N)$ where N denotes the number of parameters of the underlying neural network. This is the case for contrastive learning in DCU (Winkens et al., 2020) due to large batch sizes, the gradient penalty in DUQ (Van Amersfoort et al., 2020) due to backpropagation through the gradients of a neural network's input and spectral normalization in SNGP (Liu et al., 2020) and DDU (Mukhoti et al., 2021).

denotes an efficient approach to enforcing distance-aware representations, it renders them dependent on an underlying distance metric in the input space. Moreover, they require the underlying model to be composed of residual layers. Furthermore, we find that enforcing distance awareness, does not directly correlate with OOD detection performance. SNGP estimates uncertainty by replacing the softmax layer with a GP based on the RBF kernel. In particular, they use a Laplace approximation of the GP. They estimate epistemic uncertainty using the Dempster-Shafer metric (Liu et al., 2020). We find that SNGP yields reasonable uncertainty calibration.

DUQ (Van Amersfoort et al., 2020) prevents feature collapse by enforcing distance-aware hidden representations. Distance-awareness is enforced using the gradient penalty. While the latter allows DUQ to be model agnostic, it dramatically increases the computational/memory cost at training time. Moreover, following the general drawbacks of enforcing distance-aware representations it depends on an underlying distance metric in the input space. DUQ is only applicable to classification and replaces the softmax output layer using an RBF kernel which compares observed representations to centroids where each class in the classification problem is associated with one centroid. These centroids are updated using a running mean during the training of the model. This renders DUQ sensitive to instabilities at training time in case the mean updates are noisy. For example, the latter case can arise when the number of classes in the classification problem becomes large.

DDU (Mukhoti et al., 2021) uses spectral normalization for regularizing hidden representations (see Sec. A.1.1). While this denotes an efficient approach to enforcing distance-aware representations, it renders them dependent on an underlying distance metric in the input space. Moreover, they require the underlying model to be composed of residual layers. In order to estimate uncertainty, DDU estimates the distribution of hidden representations of the penultimate layer using a class-conditional GMM, i.e. they train one multivariate Gaussian per class. Then, epistemic uncertainty is approximated as the negative log-likelihood of the mixture components with the highest probability. This approach to uncertainty estimation allows DDU to be applied across different tasks. However, due to the generative approach to uncertainty estimation it yields poorly calibrated uncertainty.

DCU (Winkens et al., 2020) enforces informative representations at training time to counter feature collapse. To this end DCU uses constrastive learning (see Sec. A.1.2) as a regularization objective. While this approach has theoretical guarantees for maximizing the information content in the hidden representations and is architecture-agnostic, in practically requires very large batch size to generate hard negative samples at training time (Chen et al., 2020). Moreover, while the contrastive learning objective boosts performance on classification, it is not directly transferable to other tasks than classification since those may require a different set of data augmentations which are essential to the success of contrastive learning (Chen et al., 2020). In order to estimate uncertainty, DCU also estimates the distribution of hidden representations of the penultimate layer using a class-conditional GMM, i.e. they train one multivariate Gaussian per class. Then, epistemic uncertainty is approximated as the negative log-likelihood of the mixture components with the highest highest probability. This approach to uncertainty estimation allows DCU to be applied across different tasks. However, due to the generative approach to uncertainty estimation it yields poorly calibrated uncertainty.

MIR (Postels et al., 2020) regularizes hidden representations using reconstruction regularization. While this approach only heuristically increases the information content in the hidden representations, it is efficient and architecture- and task-agnostic.

In order to estimate uncertainty, MIR also estimates the distribution of hidden representations of the penultimate layer using a class-conditional GMM, i.e. they train one multivariate Gaussian per class. Then, epistemic uncertainty is approximated as the negative marginal log-likelihood. This approach to uncertainty estimation allows DCU to be applied across different tasks. However, due to the generative approach to uncertainty estimation it yields poorly calibrated uncertainty.

PostNet (Charpentier et al., 2020) learns the distribution of hidden representations end-to-end which allows them to regularize its entropy directly (see Sec. A.1.2). While this is theoretically guaranteed to lead to high information content in the hidden representations for deterministic models, it leads to some difficulties during training. To ensure stability during training the hidden representations are require to be low dimension. Furthermore, it is required to apply a Batch Normalization layer directly prior to hidden representations which distribution is estimated. While PostNet's approach can be generalized to other predictive distributions, they focus on Dirichlet distributions and thus only classification. PostNet estimates the distribution of hidden representations using one radial flow per class. Uncertainty is estimated using the likelihood of the radial flow with the maximum likelihood. Precisely, they multiply the likelihood with the elements observed for a particular class in the training set which is usually constant in their/our experiments and add one to the result. In accordance with other approaches that use generative modeling of hidden representations for uncertainty estimation, we observe poor calibration of epistemic uncertainty.

A.2. Intuition for Poor Calibration

Given a neural network NN trained on a dataset $D = (X, Y)$ with $X = \{x_i\}_{i \in |D|}$ and $Y = \{y_i\}_{i \in |D|}$, some DUMs (Winkens et al., 2020; Postels et al., 2020; Mukhoti et al., 2021) require modeling of the intermediate activations $p(z|X, Y)$ of the NN to derive estimates of the predictive uncertainty. For this reason, different types of regularization over the feature space are applied with the aim of making the latent distribution representative of the input one. Since this only captures the data distribution through the lense of a fixed set of model parameters, we argue that a key ingredient is missing to account for the total variability over latent distribution.

Taking into account the distribution $p(\theta|X, Y)$ over networks' parameters θ, which is approximated by a surrogate distribution $q(\theta)$, we obtain for the distribution $p(z|X, Y)$:

$$\begin{aligned} p(z|X,Y) &= \int_x p(z|x,X,Y)p(x|X,Y)dx \\ &= \int_x \left(\int_\theta p(z|x,\theta)p(\theta|X,Y)d\theta \right) p(x|X,Y)dx \\ &= \int_x \left(\int_\theta p(z|x,\theta)q(\theta)d\theta \right) p(x|X,Y)dx \\ &= \int_\theta \left(\int_x p(z|x,\theta)p(x|X,Y)dx \right) q(\theta)d\theta \end{aligned} \tag{2}$$

DUMs typically assume that network weights are fixed, i.e. they are distributed according to a Dirac delta function:

$$p(\theta|X,Y) \approx q(\theta) = \delta(\theta) = \begin{cases} +\infty, & \theta = \hat{\theta} \\ 0, & \theta \neq \hat{\theta} \end{cases} \tag{3}$$

subject to $\int_{-\infty}^{\infty} \delta(x)\, dx = 1$. Then,

$$\begin{aligned} p(z|X,Y) &= \int_\theta \left(\int_x p(z|x,\theta)p(x|X,Y)dx \right) q(\theta)d\theta \\ &= \int_x \left(p(z|x,\hat{\theta}) \right) p(x|X,Y)dx \end{aligned} \tag{4}$$

Intuitively, DUMs only approximate the inner integral over the distribution of inputs x, since only the distribution over the input space is taken into account and weights $\hat{\theta}$ are fixed. While this justifies the good performance of DUMs on OOD

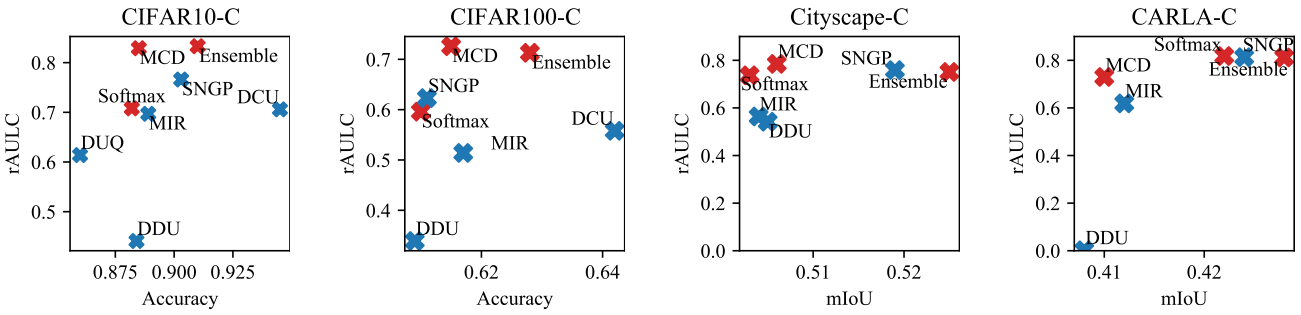


Figure 3. Scatter plots for Accuracy/mIoU versus rAULC on 4 testsets, based on Tab. 2 and 4. The baselines (red) usually occupy the top part of the figure, confirming their effectiveness of uncertainty calibration. Among the DUMs (blue), SNGP and MIR are closer to the region of baselines than others. For realistic shifts (CARLA-C), a significant drop of uncertainty calibration performance can be found for DDU. ("MCD"="MC Dropout")

detection, failing to model the weights distribution may not account for the total variability over latent distributions, thus underestimating the output predictive uncertainty and making it a bad hint for networks' expected error.

Incorporating the lack of knowledge over the network's weights is a promising direction to make DUMs well calibrated.

A.3. Additional Results

A.3.1. Additional Visualization of Quantitative Results on Continuous Distributional Shifts

Fig. 3 provides additional visualization of the test accuracy versus calibration performance for the methods compared in Tab. 2 and 4.

A.3.2. Image Classification

OOD Detection. Tab. 6 shows quantitative results on detecting OOD data for DUMs, MC dropout and deep ensembles trained on CIFAR10/100. We observe that DUMs are able to outperform MC dropout and deep ensembles on OOD detection.

Table 6. OOD detection performance when training on CIFAR10/100 and testing on various other datasets. We report AUROC averaged across 5 independent trainings.

	OOD Data	STL10	SVHN	CIFAR100
CIFAR10	MC Dropout (Gal & Ghahramani, 2016)	0.686 ± 0.004	0.885 ± 0.002	0.82 ± 0.003
	Ensemble (Lakshminarayanan et al., 2017)	**0.875 ± 0.001**	0.937 ± 0.009	0.758 ± 0.003
	DUQ (Van Amersfoort et al., 2020)	0.633 ± 0.008	0.843 ± 0.016	0.766 ± 0.003
	SNGP (Liu et al., 2020)	0.726 ± 0.007	0.925 ± 0.02	0.861 ± 0.004
	MIR (Postels et al., 2020)	0.752 ± 0.015	0.916 ± 0.025	0.840 ± 0.007
	DDU (Mukhoti et al., 2021)	0.737 ± 0.018	0.663 ± 0.073	0.638 ± 0.004
	DCU (Winkens et al., 2020)	0.725 ± 0.027	**0.992 ± 0.014**	**0.921 ± 0.014**
	OOD Data	STL10	SVHN	CIFAR10
CIFAR100	MC Dropout (Gal & Ghahramani, 2016)	0.772 ± 0.004	0.846 ± 0.01	0.735 ± 0.002
	Ensemble (Lakshminarayanan et al., 2017)	**0.801 ± 0.014**	0.741 ± 0.003	0.756 ± 0.007
	DUQ (Van Amersfoort et al., 2020)	-	-	-
	SNGP (Liu et al., 2020)	0.744 ± 0.02	0.795 ± 0.112	0.686 ± 0.007
	MIR (Postels et al., 2020)	0.789 ± 0.025	0.809 ± 0.031	0.663 ± 0.004
	DDU (Mukhoti et al., 2021)	0.698 ± 0.021	0.809 ± 0.056	**0.764 ± 0.019**
	DCU (Winkens et al., 2020)	0.798 ± 0.019	**0.978 ± 0.005**	0.755 ± 0.024

Sensitivity to regularization strength. We provide additional ablation studies on the sensitivity to regularization strength for different methods on MNIST (Fig. 4) and FashionMNIST (Fig. 5) These results confirm the findings of the

main manuscript, *i.e.* that only MIR and Dropout are sensible to regularization strength, while DUMs based on Lipschitz regularization are not influenced by the regularization strength.

Corruption Severity Analysis. We show the classification accuracy, AUROC and rAULC for each method on the CIFAR10-C (Fig. 9, Fig. 10, Fig. 11) and CIFAR100-C (Fig. 12, Fig. 13, Fig. 14) datasets under different types of corruptions. Each type of corruption is applied at 5 increasing levels of severity. For ease of visualization, we split the 15 different types of corruptions applied on the CIFAR10-C and CIFAR100-C datasets into 3 different figures each. Each figure shows 5 different types of corruptions.

While all methods demonstrate a similar predictive performance pattern, DUMs - in particular methods based on generative modeling of hidden representations - yield worse calibration across corruption severities.

Training/Inference Runtime Comparison. We report the per sample training and inference runtime in Tab. 7. The runtimes were measured on a single V100 using CIFAR10 and a ResNet50 backbone.

Table 7. Per sample runtime during training and inference on CIFAR10. The runtimes of MC dropout were obtained using 10 samples.

Method	Training Runtime [ms]	Inference Runtime [ms]
Softmax	1.14	0.36
MC Dropout (Gal & Ghahramani, 2016)	1.13	5.17
DUQ (Van Amersfoort et al., 2020)	3.68	0.35
SNGP (Liu et al., 2020)	2.47	0.44
DUE (Mukhoti et al., 2021)	2.26	0.49
MIR (Postels et al., 2020)	1.34	0.55
DDU (Mukhoti et al., 2021)	2.26	0.49

A.3.3. Semantic Segmentation

Examples of segmentation and uncertainty masks. We show qualitative examples of predicted masks, error masks and uncertainty masks for Softmax, MC dropout, SNGP and MIR on semantic segmentation. Fig. 6 illustrates examples under *minimal* distributional shift (*i.e.* Azimuth angle of the sun $= 85°$ and Fig. 7 under *maximal* distributional shift (*i.e.* Azimuth angle of the sun $= -5°$. We show the input image (Input), the segmentation ground truth (GT), the predicted segmentation mask (Prediction), the error mask (Error) and the uncertainty mask (Uncertainty). The error mask is computed as a boolean mask with True values when a pixel is predicted wrongly (yellow) and False (blue) when the prediction is instead correct. The uncertainty mask is preprocessed to facilitate visualization. In particular, we first compute mean μ and standard deviation σ of per-pixel uncertainties over each uncertainty mask. Then, the uncertainty mask is clipped between $[\mu - 2\sigma, \mu + 2\sigma]$. Finally, the uncertainty mask is normalized between 0 and 1 before being visualized.

While the softmax entropy provides decent uncertainty estimates under minimal distributional shift, it tends to be overconfident under severe distributional shift. In particular, Softmax models are only uncertain close to object borders, but they are confident about large portions of the image that are instead predicted wrongly. This can be observed in Fig. 7, where all models tend to predict the entire sky wrongly (assigned to 'building' class), but the Softmax model is the most confident about its predictions of the sky being correct. DUMs and MC dropout do a better job at recognizing wrong predictions under sever domain shift by outputting higher uncertainty values.

Qualitative Behaviour on Continuous Distributional Shifts We show the mIoU, AUROC and rAULC for each method on the Carla (Fig. 18) and Cityscapes (Fig. 15, Fig. 16, Fig. 17) datasets under different levels of corruptions. For ease of visualization, we split the 15 different types of corruptions applied on the Cityscapes dataset into 3 different figures, showing 5 types of corruptions each.

While all methods demonstrate a similar mIoU pattern, DUMs - in particular methods based on generative modeling of hidden representations - yield worse calibration across corruption severities.

A.4. Training Details

We provide training and optimization details for all evaluated methods. All methods using spectral normalization use 1 power iteration. Hyperparameters were chosen to minimize the validation loss.

A.4.1. IMAGE CLASSIFICATION - MNIST/FASHIONMNIST

All methods trained on MNIST/FashionMNIST used a MLP as backbone with 3 hidden layers of 100 dimensions each and ReLU activation functions. We used a batch size of 128 samples and trained for 200 epochs. No data augmentation is performed.

Softmax and Deep ensembles. We used for the single softmax model the Adam optimizer with learning rate 0.003, and L_2 weight regularization 0.0001. When using ensembles, 10 models are trained from different random initializations.

MC dropout. We used for all baselines the Adam optimizer with learning rate 0.003, dropout rate 0.4 and L_2 weight regularization 0.0001.

We found the optimal SN coefficient to be 7, with the GP approximation using 10 (number of classes) inducing points initialized using k-means over 10000 samples.

DUQ We trained DUQ with the SGD optimizer with learning rate 0.01, L_2 weight regularization 0.0001, and a multi-step learning rate decay policy with decay rate 0.3 and decay steps at the epochs 10, 20. Length-scale for the RBF kernel is 0.1 and optimal gradient penalty loss weight is 0 where we searched along the grid $[0.0, 0.0000001, 0.0000003, 0.000001, 0.000003, 0.00001, 0.00003, 0.0001, 0.0003, 0.001, 0.005, 0.01, 0.025, 0.05, 0.075, 0.1, 0.2, 0.5]$.

DDU. We trained DDU with the Adam optimizer with learning rate 0.001, L_2 weight regularization 0.0001, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 100, 200, 300. We found the optimal SN coefficient to be 6 searching along the grid $[1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 15]$. The GMM is fitted by estimating the empirical mean and covariance matrix of the representations on the training data associated with each class.

SNGP. We trained SNGP with the SGD optimizer with learning rate 0.05, L_2 weight regularization 0.0003, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 60, 120, 160. We found the optimal SN coefficient to be 6 searching along the grid $[1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 15]$., with the GP approximation using 10 hidden dimensions, lengthscale 2 and mean field factor 30.

MIR. We trained MIR with the Adam optimizer with learning rate 0.001, and L_2 weight regularization 0.0001. We found the optimal reconstruction loss weight to be 1 after searching along the grid $[0.0, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0, 50.0, 100.0]$.

A.4.2. IMAGE CLASSIFICATION - CIFAR10/SVHN

When training on CIFAR-10/SVHN, we use a ResNet-18 (He et al., 2016) as backbone. The dimensionality of the last feature space encoded with the ResNet backbone is 100 for all methods. We used a batch size of 128 samples and trained for 400 epochs. The training set is augmented with common data augmentation techniques. We apply random horizontal flips, random brightness augmentation with maximum delta 0.2 and random contrast adjustment with multiplier lower bound 0.8 and upper bound 1.2.

Softmax and Deep ensembles. We used for the single softmax model the Adam optimizer with learning rate 0.003, L_2 weight regularization 0.0001, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 250, 300, 400. When using ensembles, 10 models are trained from different random initializations.

MC dropout. We used for all baselines the Adam optimizer with learning rate 0.003, dropout rate 0.3, L_2 weight regularization 0.0001, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 250, 300, 400.

DUE We trained DUE with the SGD optimizer with learning rate 0.01, L_2 weight regularization 0.0005, dropout rate 0.1, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 100, 200, 300. We found the optimal SN coefficient to be 7 for SVHN and 9 for CIFAR-10, with the GP approximation using 10 (number of classes) inducing points initialized using k-means over 10000 samples.

DUQ We trained DUE with the SGD optimizer with learning rate 0.01, L_2 weight regularization 0.0001, dropout rate 0.1, and a multi-step learning rate decay policy with decay rate 0.3 and decay steps at the epochs 200, 250. 300. Lengthscale for the RBF kernel is 0.1 and optimal gradient penalty loss weight is 0

DDU. We trained DDU with the Adam optimizer with learning rate 0.001, L_2 weight regularization 0.0001, dropout rate 0.3, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 80, 120, 180. We found

the optimal SN coefficient to be 7. The GMM fit on top of the pretrained feature extractor is trained for 100 epochs and is fit with 64 batches.

SNGP. We trained SNGP with the SGD optimizer with learning rate 0.05, L_2 weight regularization 0.0004, dropout rate 0.1, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 100, 200, 300. We found the optimal SN coefficient to be 7, with the GP approximation using 10 hidden dimensions, lengthscale 2 and mean field factor 30.

MIR. We trained MIR with the Adam optimizer with learning rate 0.003, L_2 weight regularization 0.0001, dropout rate 0.1, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 150, 200, 250, 300. We found the optimal reconstruction loss weight to be 1.

A.4.3. SEMANTIC SEGMENTATION.

When training on semantic segmentation, we use a DRN (Yu & Koltun, 2016; Yu et al., 2017) (DRN-A-50) as backbone. We used a batch size of 4 samples and trained for 200 epochs. Images are rescaled to size 400×640 The training set is augmented with common data augmentation techniques. All training samples are augmented with random cropping with factor 0.8. We apply random horizontal flips, random brightness augmentation with maximum delta 0.2 and random contrast adjustment with multiplier lower bound 0.8 and upper bound 1.2.

Softmax. We used for the single softmax model the Adam optimizer with learning rate 0.0004, L_2 weight regularization 0.0001, and a multi-step learning rate decay policy with decay rate 0.3 and decay steps at the epochs 30, 60, 90, 120.

MC dropout. We used for all baselines the Adam optimizer with learning rate 0.0004, dropout rate 0.4, L_2 weight regularization 0.0001, and a multi-step learning rate decay policy with decay rate 0.3 and decay steps at the epochs 30, 60, 90, 120.

SNGP. We trained SNGP with the SGD optimizer with learning rate 0.0002, L_2 weight regularization 0.0003, dropout rate 0.1, and a multi-step learning rate decay policy with decay rate 0.2 and decay steps at the epochs 20, 40, 60, 80, 100. We found the optimal SN coefficient to be 6, with the GP approximation using 128 hidden dimensions, lengthscale 2 and mean field factor 25.

MIR. We trained MIR with the Adam optimizer with learning rate 0.0002, L_2 weight regularization 0.0001, dropout rate 0.1, and a multi-step learning rate decay policy with decay rate 0.3 and decay steps at the epochs 30, 60, 90, 120. We found the optimal reconstruction loss weight to be 1.

A.5. Implementation Details.

All methods were re-implemented in Tensorflow 2.0. We payed attention to all the details reported in each paper and we run all experiments for each method multiple times to account for stochasticity, *i.e.* 5 times for classification and 3 times for segmentation. When an implementation was publicly available, we relied on it. This is the case for DUQ (https://github.com/y0ast/deterministic-uncertainty-quantification), SNGP (https://github.com/google/uncertainty-baselines/blob/master/baselines/imagenet/sngp.py) and DUE (https://github.com/y0ast/DUE).

SNGP. We follow the publicly available implementation of SNGP, which, compared to the implementation described in the original paper, proposes to further reduce the computational overhead of the GP approximation by replacing the Monte-Carlo averaging with the mean-field approximation (Daunizeau, 2017). This is especially relevant in large-scale tasks like semantic segmentation, were it is important to reduce the computational overload.

A.5.1. IMAGE CLASSIFICATION

DUE. Note that only DUE uses a SN approximation also for the batch normalization layer. All other methods only restrict the Lipschitz constant of convolutional and fully connected layers.

MIR only differs from regular softmax models in its decoder module used for the reconstruction regularization loss (Postels et al., 2020). When training MLP architectures the decoder is comprised of two fully-connected layer. The first has a ReLU activation function and 200 output neruons. The second has a linear activation function and its output dimensionality equals that of the models' input data. When training convolutional neural networks the decoder is comprised of four blocks

of transpose convolutions, batch normalization layers and ReLU activation functions that gradually upscale the hidden representations to the dimensionality of the input data. These four blocks are followed by a 1x1 convolution with linear activation function.

A.5.2. SEMANTIC SEGMENTATION

MIR. Similar to image classification, MIR only differs from regular segmentation models in its decoder module used for the reconstruction regularization loss (Postels et al., 2020). The decoder module is comprised of a single point-wise feed forward layer that maps the hidden representations $\mathbf{z} \in \mathbb{R}^{W_z \times H_z \times C_z}$ to $\mathbf{z} \in \mathbb{R}^{W_z \times H_z \times 3}$. Subsequently, the result is bilinearly upsampled to the image resolution on which we compute the reconstruction loss.

A.5.3. UNCERTAINTY DERIVATION.

We provide details on the estimation of uncertainty for the baseline methods. For details on the uncertainty derivation in DUMs, please refer to Sec. 3 of the main paper or to the original paper of each amalysed method.

Softmax. In case of the softmax baseline we estimate uncertainty using the entropy of the predictive distribution parameterized by the neural network. Given an input $\mathbf{x}$ the entropy H is given by $H(\mathbf{y}|\mathbf{x}) = \sum -p(\mathbf{y}|\mathbf{x}) \log(p(\mathbf{y}|\mathbf{x}))$ where $p(\mathbf{y}|\mathbf{x})$ are the softmax probabilities.

MC dropout and deep ensembles. We following (Gal et al., 2017) and compute epistemic uncertainty as the conditional mutual information between the weights $\mathbf{w}$ and the predictions $\hat{y}$ given the input $\mathbf{x}$. Given an input $\mathbf{x}$ and a set of weights $\mathbf{w}$ we observe the predictive distribution $p(\hat{y}|x, w)$. Then epistemic uncertainty u_{ep} is calculated by approximating the mutual information conditioned on the input $\mathbf{x}$:

$$
\begin{aligned}
u_{ep} &= I\left(\hat{y}, w|x\right) \\
&= H(\hat{y}|x) - H(\hat{y}|w, x) \\
&= E_{y \sim p(\hat{y}|x)}\left[-\log(p(\hat{y}|x))\right] - u_{al}
\end{aligned}
$$

where u_{al} denotes the aleatoric uncertainty. Here, $p(\hat{y}|x) = \int \mathrm{d}w p(w) p(\hat{y}|x, w)$ is evaluated using a finite set of samples/ensemble members.

A.5.4. UNCERTAINTY DERIVATION FOR SEMANTIC SEGMENTATION.

We derive uncertainty estimates for each method for semantic segmentation. We average pixel-level uncertainties under the assumption that all pixels are represented by i.i.d. variables.

Uncertainty. In our experiments on continuous distributional shifts we want to estimate pixel-level uncertainty for the output map.

MIR estimates epistemic uncertainty using the likelihood of hidden representations $\mathbf{z} \in \mathbb{R}^{W_z \times H_z \times C_z}$. Since $\mathbf{z}$ is high-dimensional in our experiments, we assume that it factorizes along W_z and H_z and is translation invariant. Formally, $p(\mathbf{z}) = \prod_i^{W_z} \prod_j^{H_z} p_\theta(\mathbf{z}_{ij})$ where $\mathbf{z}_{ij} \in \mathbb{R}^{W_z \times H_z}$ and θ is shared across W_z and H_z.

We parameterize p_θ with a GMM with $n = 10$ components where each component has a full covariance matrix. We fit the GMM on 100000 hidden representations ($\mathbf{z}_{ij} \in \mathbb{R}^{C_z}$) randomly picked from the training dataset post-training. Since $C_z = 1024$ is still high-dimensional, we first apply PCA to reduce its dimensionality to 32.

In the dilated resnet architecture used for semantic segmentation the latent representation $\mathbf{z}$ is passed through a point-wise feedforward layer $f : \mathbb{R}^{W_z \times H_z \times C_z} \mapsto \mathbb{R}^{W_z \times H_z \times 3}$ and, subsequently, bilinearly upsampled to image resolution ($\mathbb{R}^{W \times H \times K}$) where K is the number of classes. We could estimate the global, *i.e.*image-level, uncertainty of an input, by providing the negative log-likelihood of the factorizing distribution. However, in order to also obtain pixel-wise uncertainties using MIR, we first compute the negative log-likelihood (*i.e.*epistemic uncertainty) associated with each latent representation $\mathbf{z}_{ij}$. Then, we bilinearly upsample the negative log-likelihoods and use the result as proxy for pixel-wise epistemic uncertainty. If we wanted to obtain a global, *i.e.*image-level, uncertainty we could average pixel-level uncertainties.

A.6. Dataset

To benchmark our model on data with realistically and continuously changing environment, we collect a synthetic dataset for semantic segmentation. We use the CARLA Simulator (Dosovitskiy et al., 2017) for rendering the images and segmentation masks. The classes definition is aligned with the CityScape dataset (Cordts et al., 2016). In order to obtain a fair comparison, all the OOD data are sampled with the same trajectory and the environmental objects, except for the time-of-the-day or weather parameters.

In-domain data The data is collected from 4 towns in CARLA. We produce 32 sequences from each town. The distribution of the vehicles and pedestrians are randomly generated for each sequence. Every sequence has has 500 frames with a sampling rate of 10 FPS. From them we randomly sample the training and validation set.

Out-of-domain data Here, we consider the time-of-the-day and the rain strength as the parameters for the continuous changing environment. In practice, these two parameters have major influence for autonomous driving tasks.

The change of the time-of-the-day is illustrated in Fig. 8 (first and second row). The time-of-the-day is parametrized by the Sun's altitude angle, where $90°$ means the mid-day and the $0°$ means the dusk or dawn. Here, we produce samples with the altitude angle changes from $90°$ to $15°$ by step of $5°$, and $15°$ to $-5°$ by step of $1°$ where the environment changes shapely. From these examples, we can confirm that the change of time-of-the-day leads to the major change in the lightness, color and visibility of the sky, roads and the buildings nearby. The effect of rain strength is demonstrated in Fig. 8 (bottom). Here the cloudiness and, ground wetness and ground reflection are the main changing parameters.

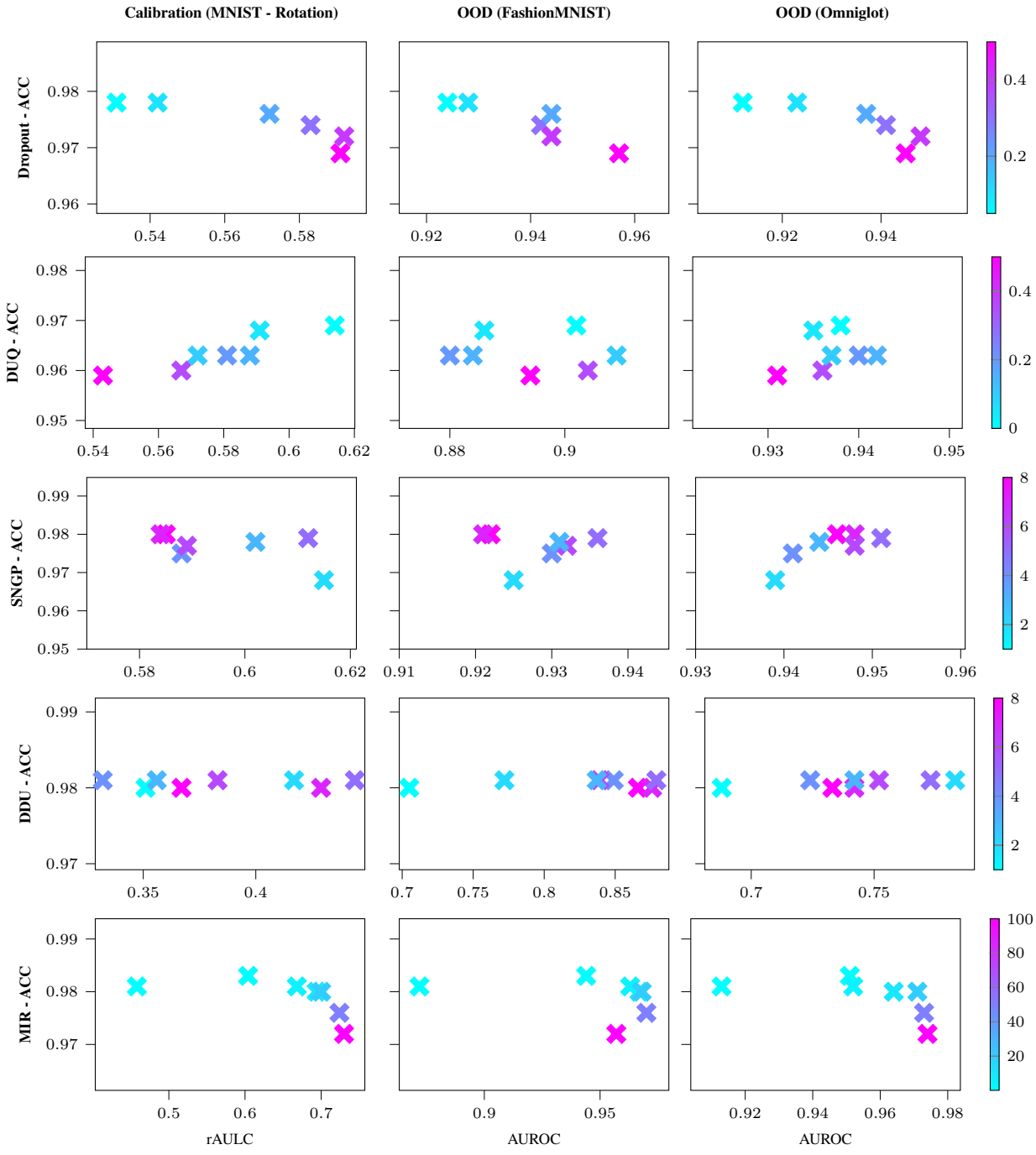


Figure 4. Trained on MNIST. Vertical axis: Test accuracy. Horizontal axis: rAULC (left), AUROC against FashionMNIST (center) and Omniglot (right) for Dropout (1st row), DUQ (2nd row), SNGP (3rd row), DDU (4th row) and MIR (5th row) using different regularization strength. For SNGP a larger hyperparameter corresponds to less regularization. For Dropout and MIR we observe a correlation between regularization strength and performance.

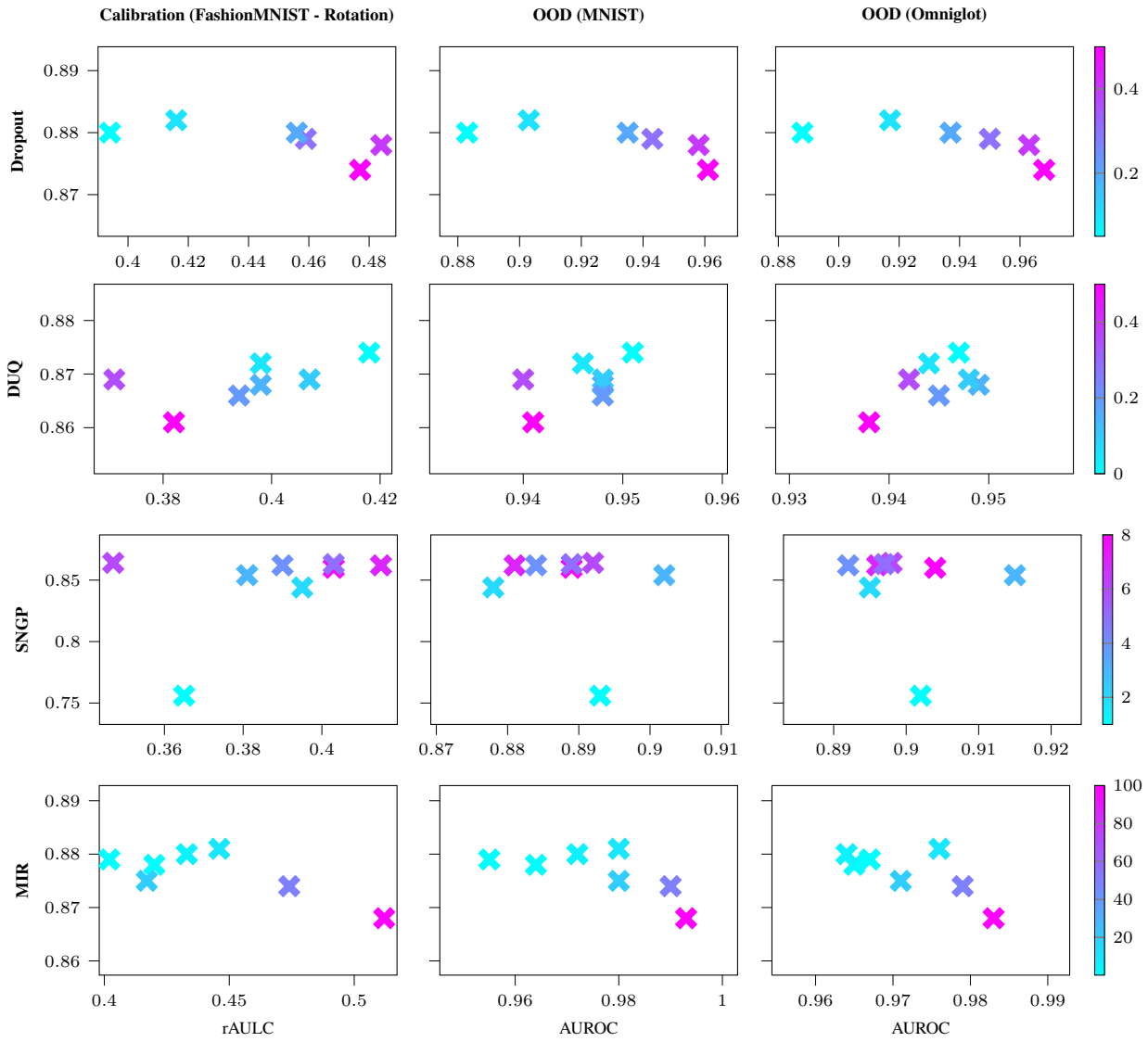


Figure 5. Trained on FashionMNIST. Vertical axis: Test accuracy. Horizontal axis: rAULC (left), AUROC against MNIST (center) and Omniglot (right) for Dropout (1st row), DUQ (2nd row), SNGP (3rd row) and MIR (4th row) using different regularization strength. For SNGP a larger hyperparameter corresponds to less regularization. For Dropout and MIR we observe a correlation between regularization strength and performance.

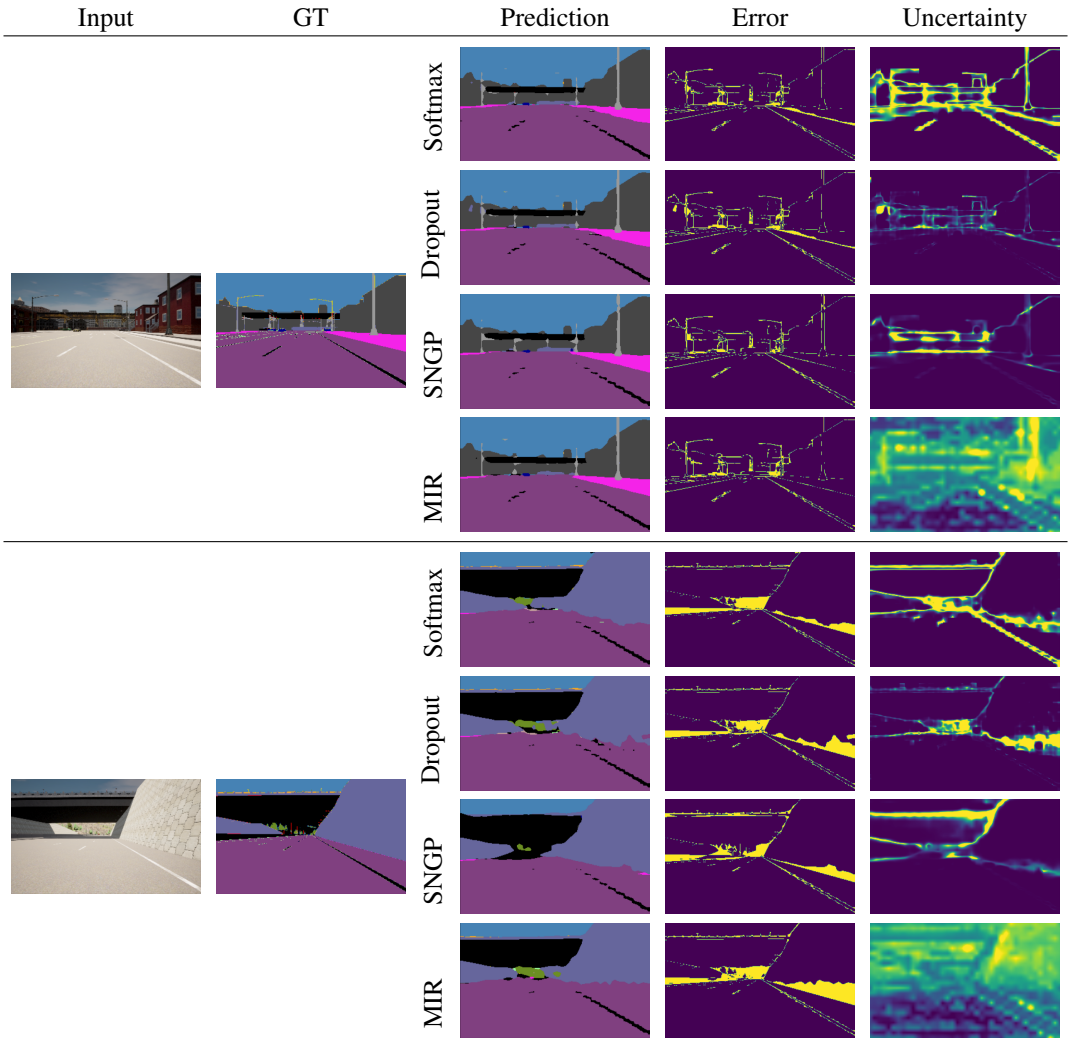


Figure 6. Qualitative comparison of uncertainty from Softmax, MC Dropout, SNGP and MIR under minimal time-of-the-day distribution shift (*i.e.* Azimuth angle of the sun $= 85°$). We show the input image (Input) and the ground truth mask (GT), and we report for each method the predicted segmentation mask (Prediction), the error mask (Error) and the uncertainty mask (Uncertainty).

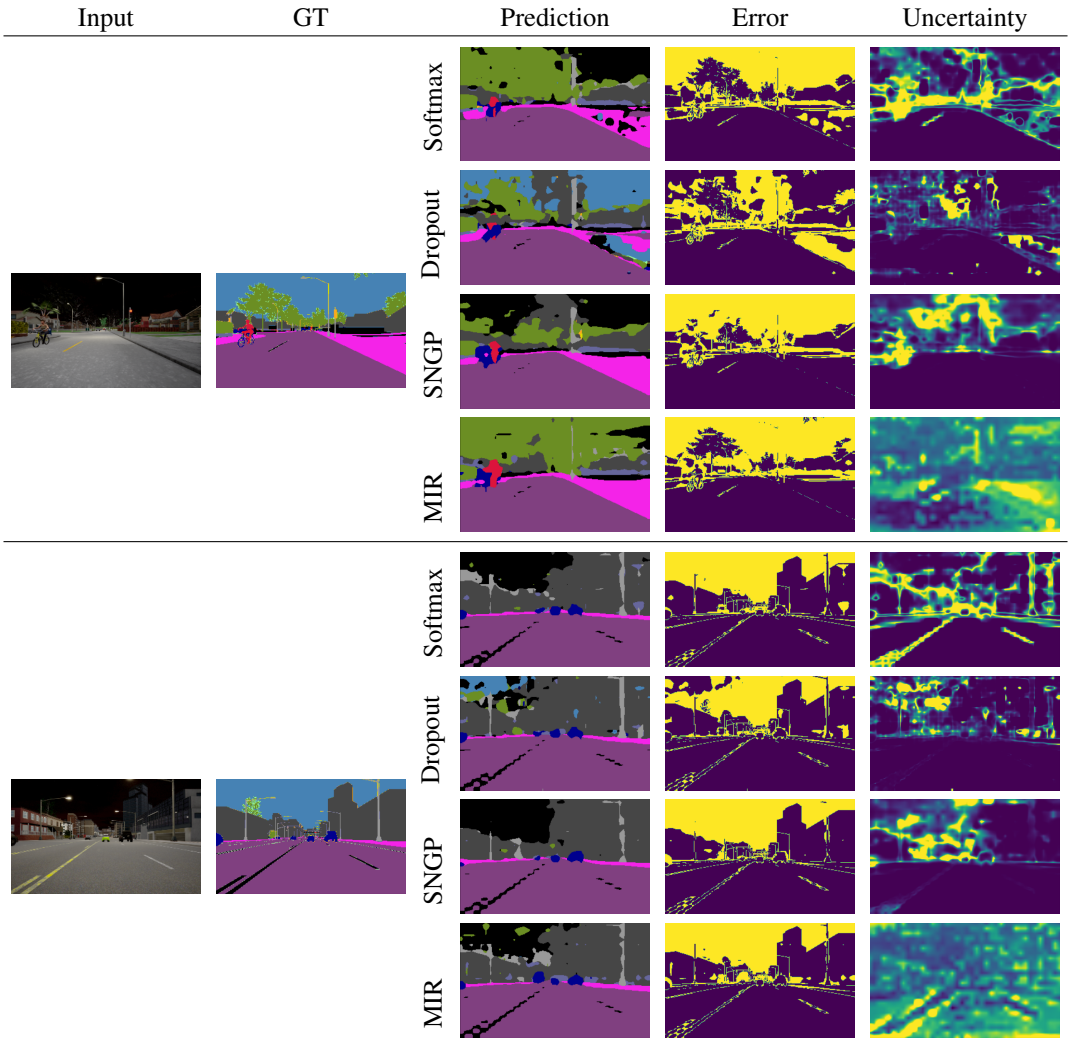
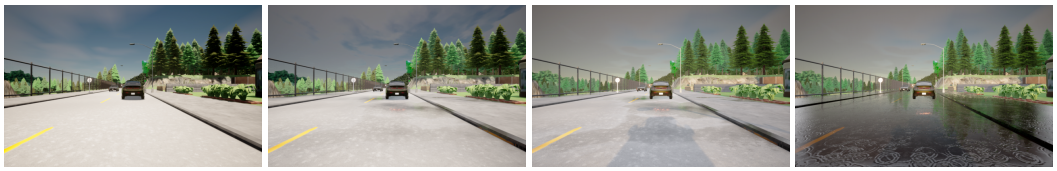


Figure 7. Qualitative comparison of uncertainty from Softmax, MC Dropout, SNGP and MIR under maximal time-of-the-day distribution shift (*i.e.* Azimuth angle of the sun $= -5^{\circ}$). We show the input image (Input) and the ground truth mask (GT), and we report for each method the predicted segmentation mask (Prediction), the error mask (Error) and the uncertainty mask (Uncertainty).

(a) Midday (90°) (b) Afternoon (45°) (c) Afternoon (15°) (d) Evening (10°)

(e) Dusk (2°) (f) Sunset (0°) (g) Sunset (-2°) (h) Night ($\leq -5^\circ$)



(i) Clear (j) Cloudy (k) Small rain (l) Heavy rain

Figure 8. Changing of the time-of-the-day and the weather

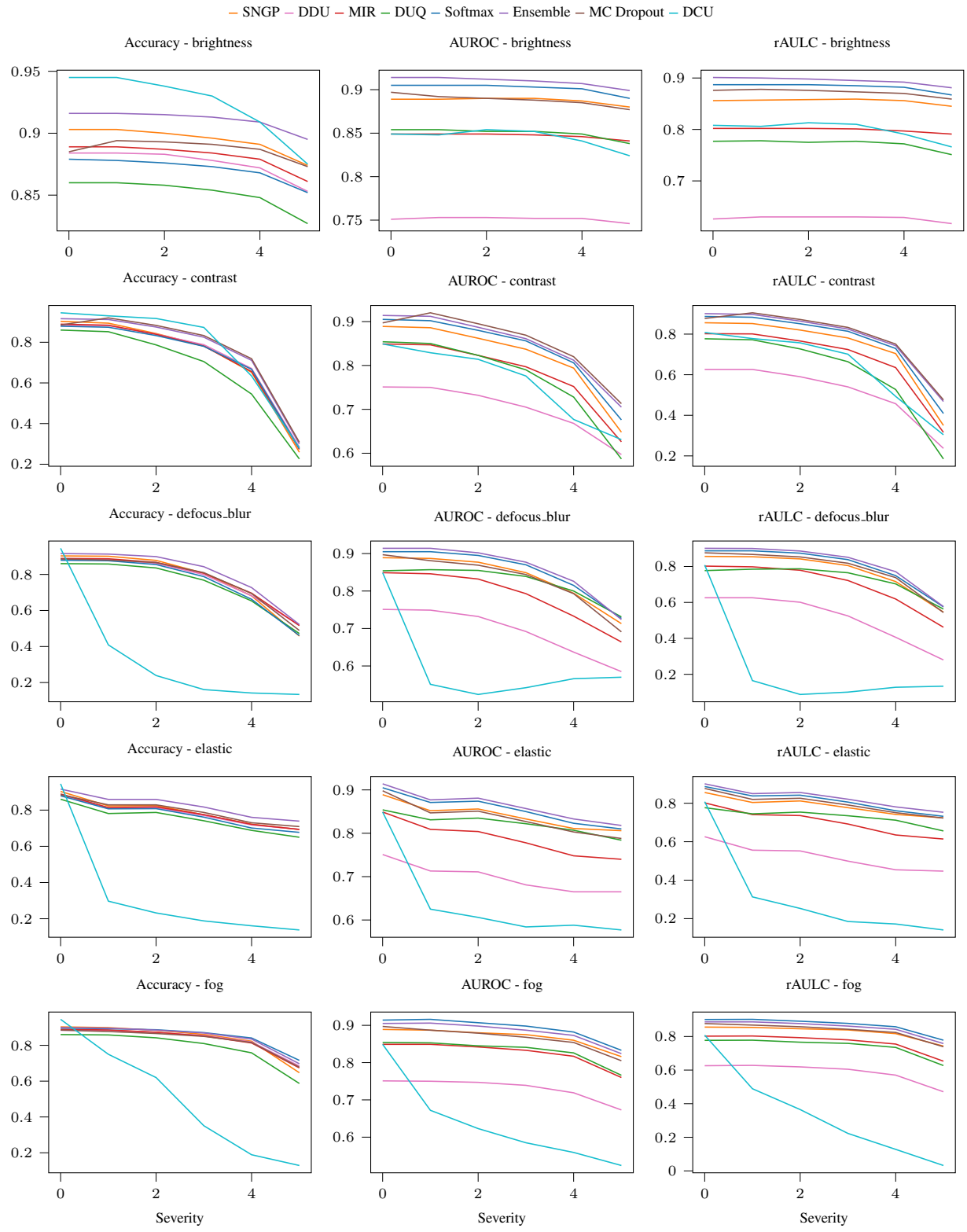


Figure 9. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR10-C dataset. We show the accuracy, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): brightness, contrast, defocus blur, elastic, fog.

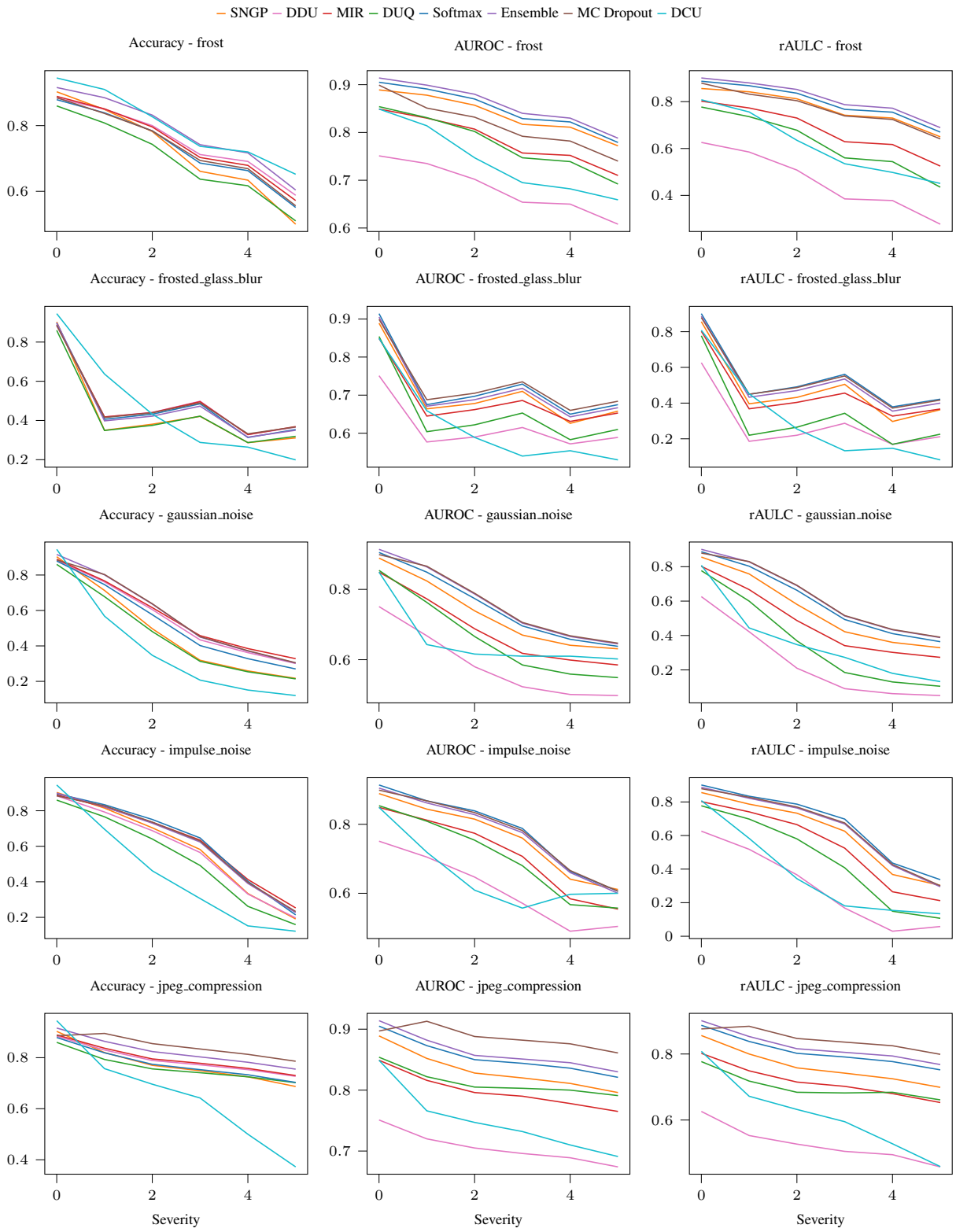


Figure 10. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR10-C dataset. We show the accuracy, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): frost, frosted glass blur, gaussian noise, impulse noise, jpeg compression.

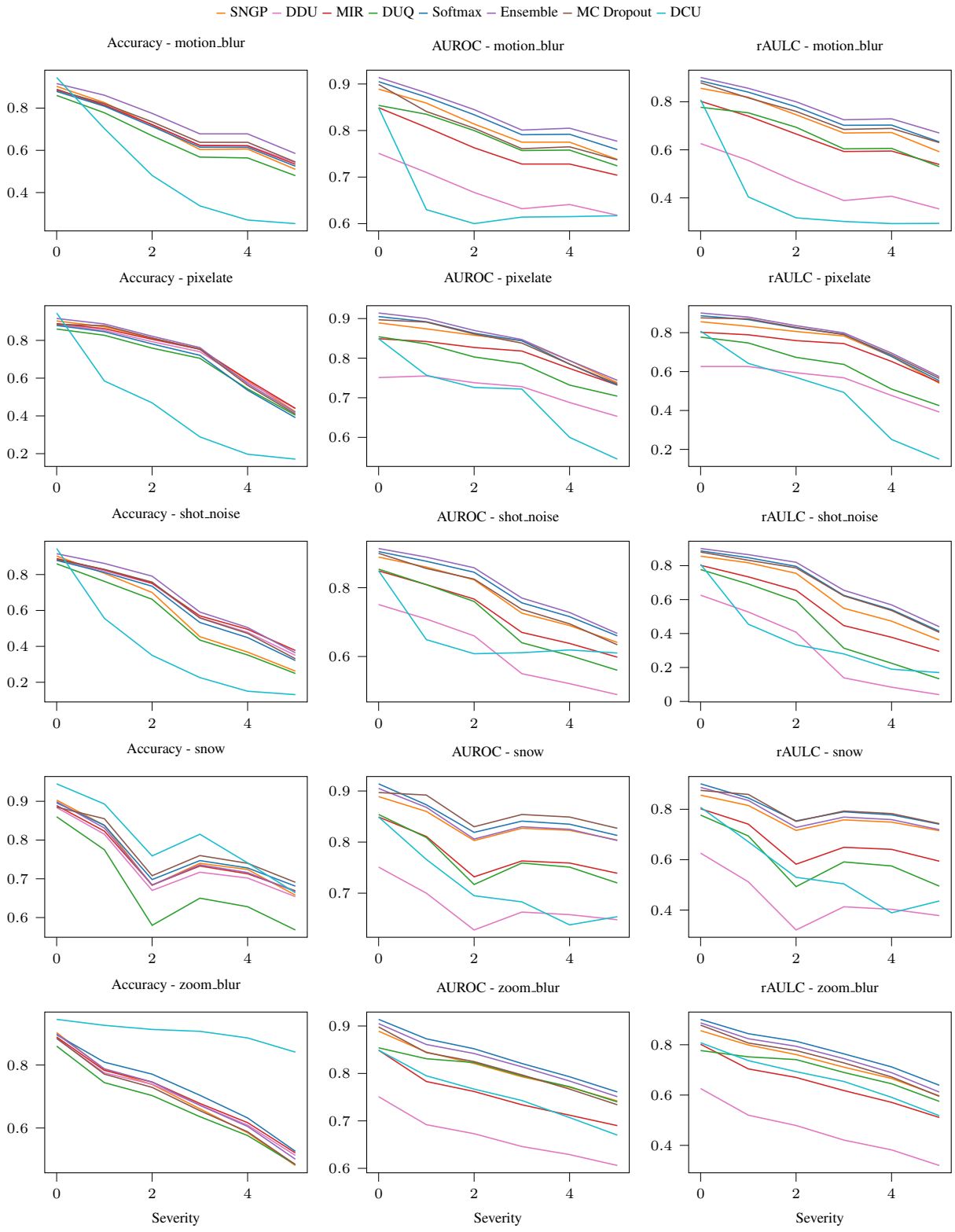


Figure 11. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR10-C dataset. We show the accuracy, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): motion blur, pixelate, shot noise, snow, zoom blur.

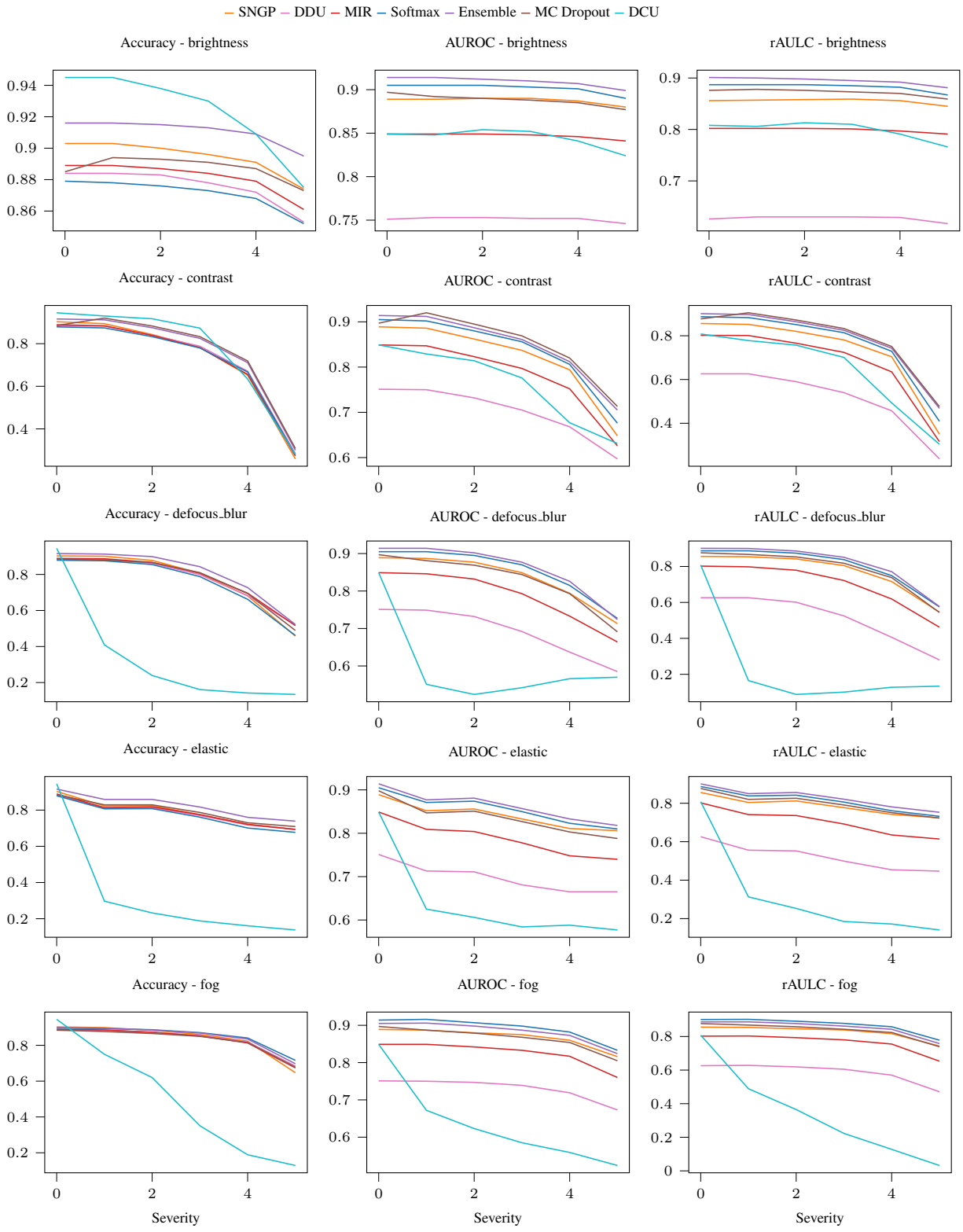


Figure 12. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR100-C dataset. We show the accuracy, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): brightness, contrast, defocus blur, elastic, fog.

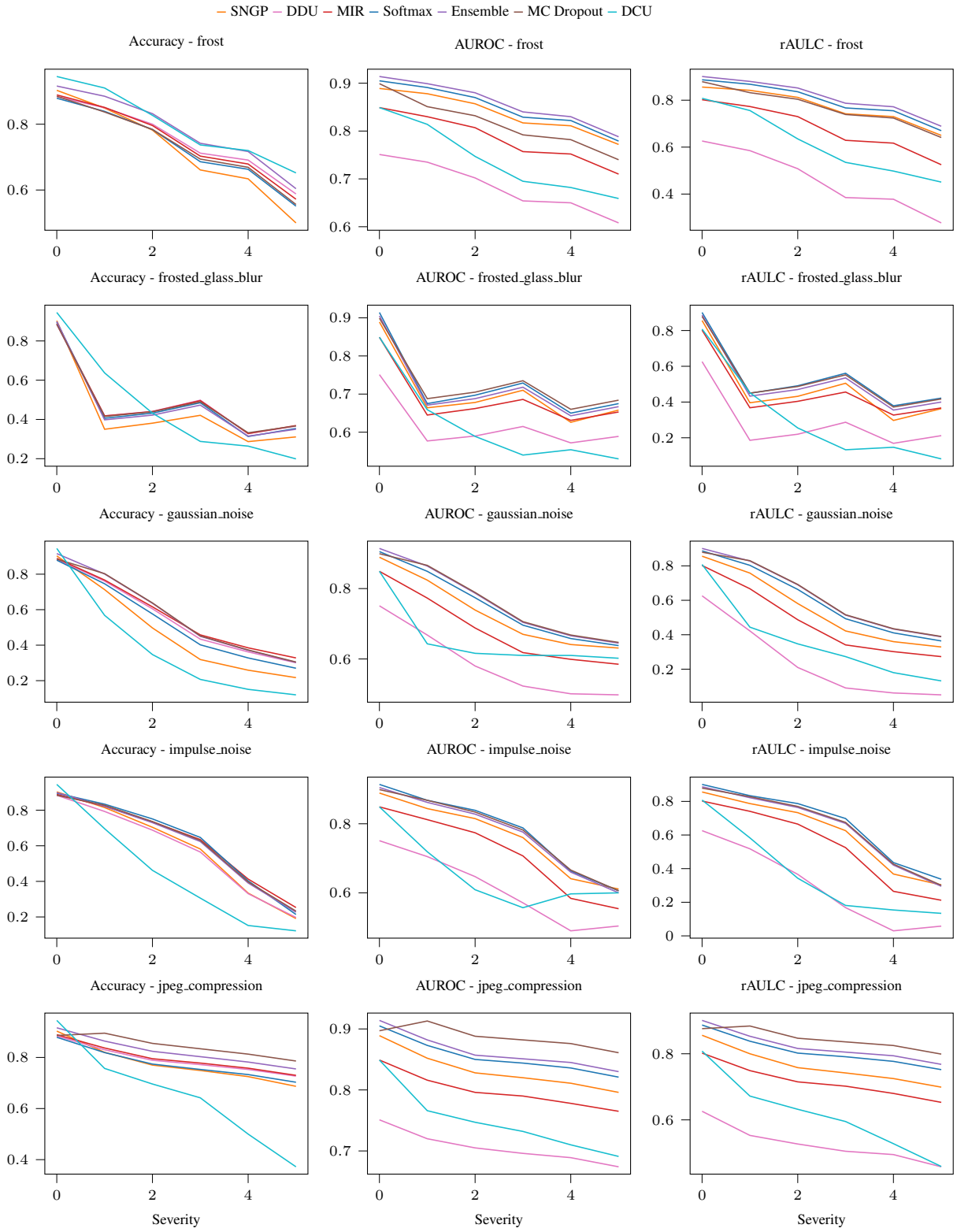


Figure 13. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR100-C dataset. We show the accuracy, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): frost, frosted glass blur, gaussian noise, impulse noise, jpeg compression.

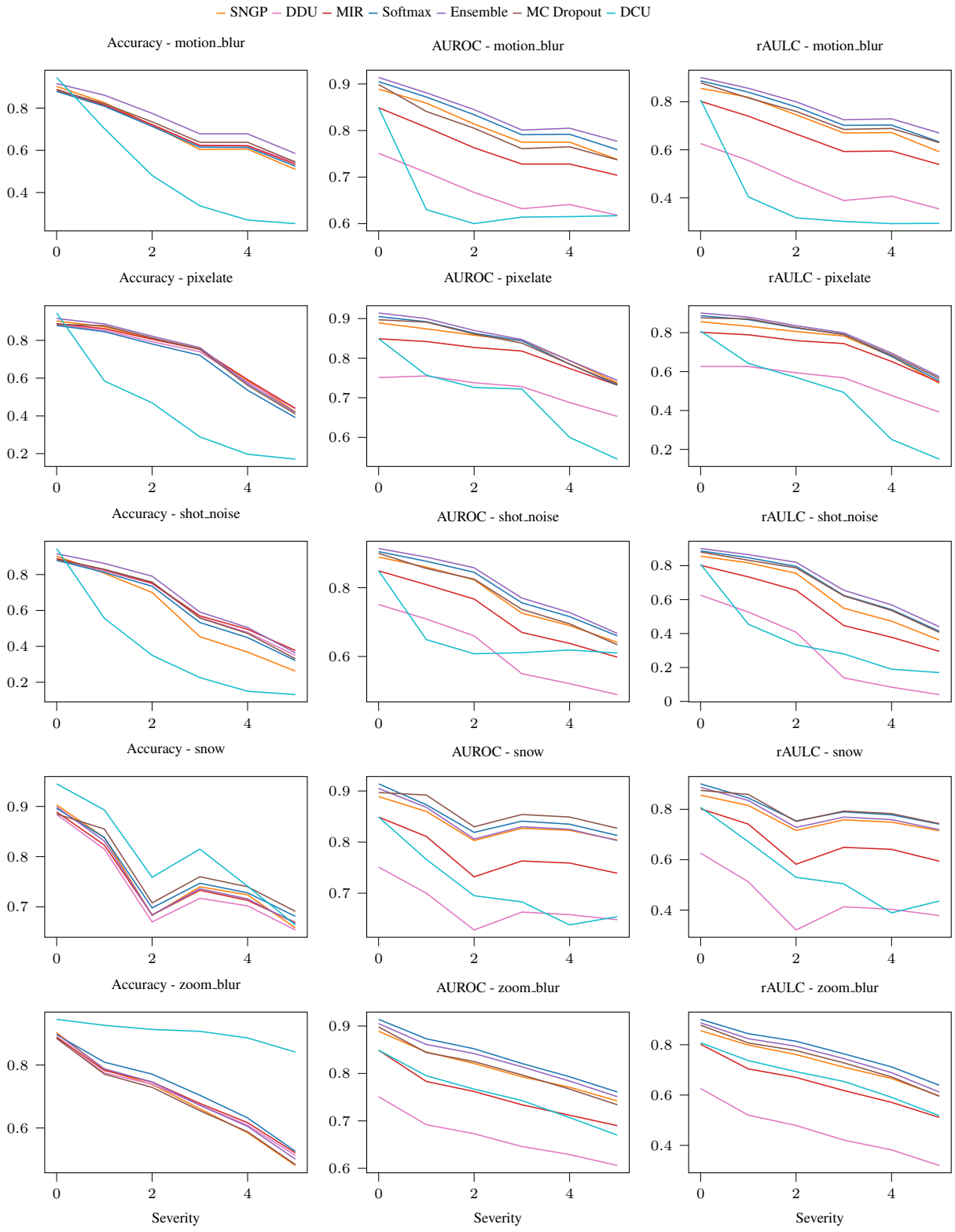


Figure 14. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR100-C dataset. We show the accuracy, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): motion blur, pixelate, shot noise, snow, zoom blur.

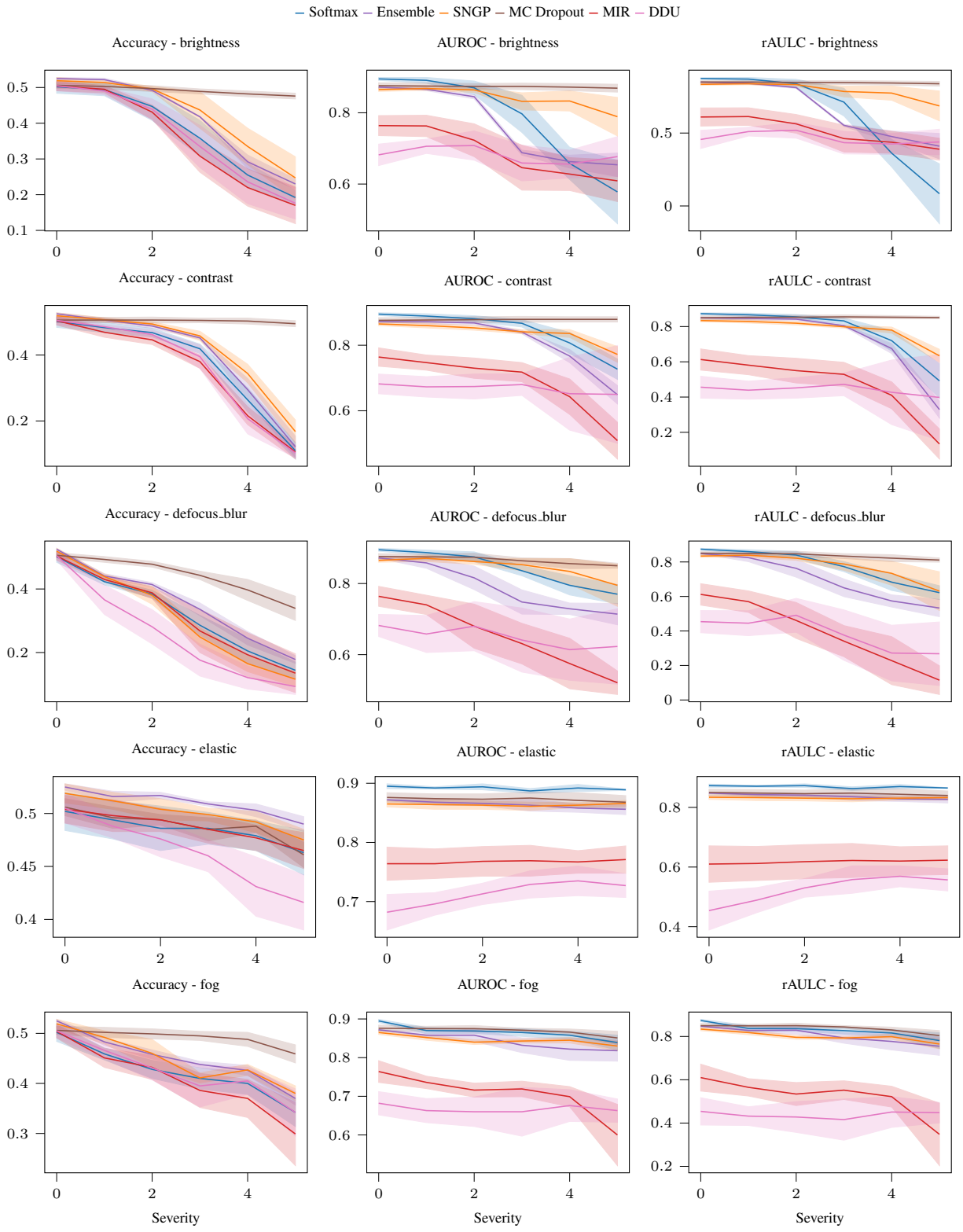


Figure 15. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR100-C dataset. We show the mIoU, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): brightness, contrast, defocus blur, elastic, fog.

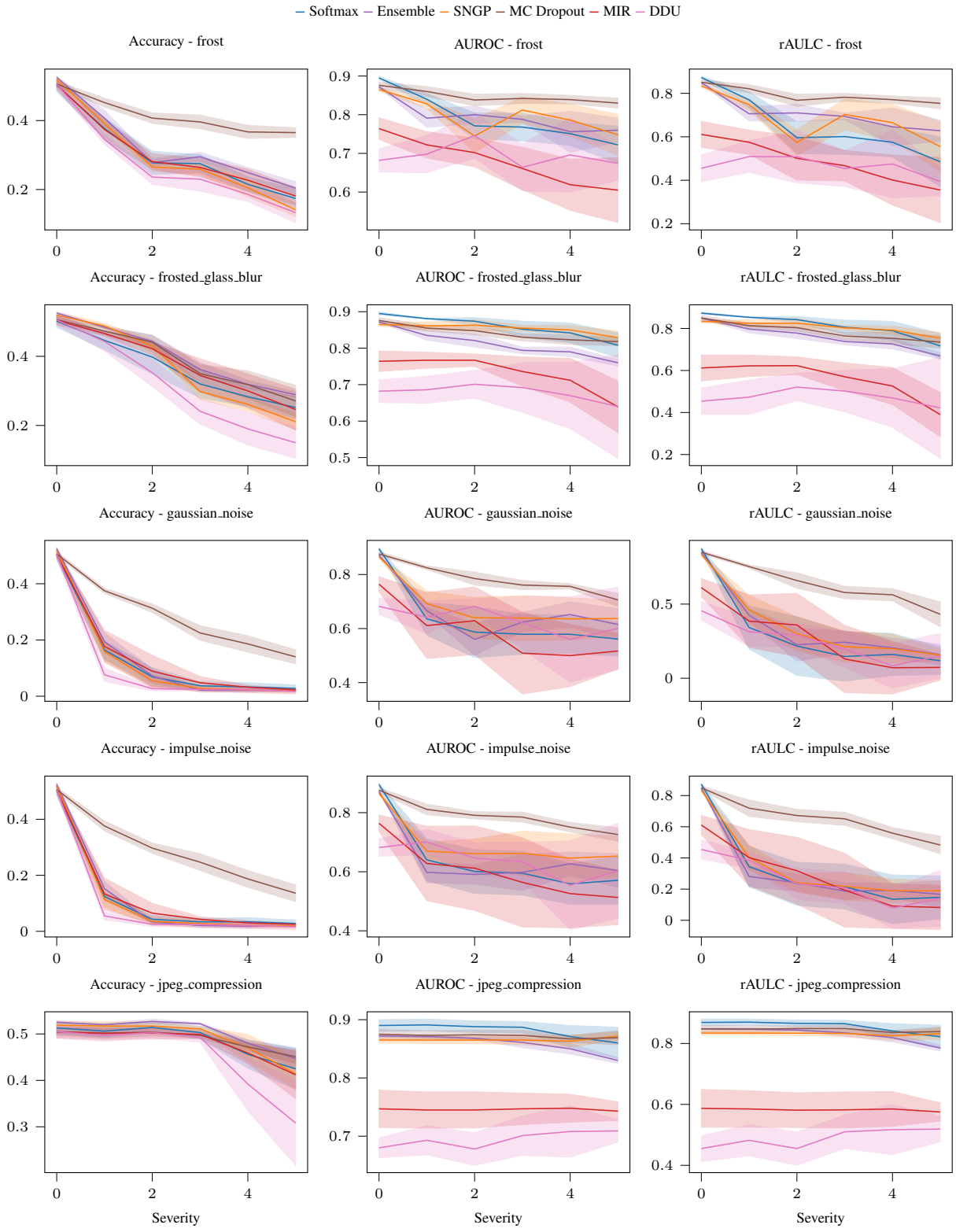


Figure 16. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the CIFAR100-C dataset. We show the mIoU, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): frost, frosted glass blur, gaussian noise, impulse noise, jpeg compression.

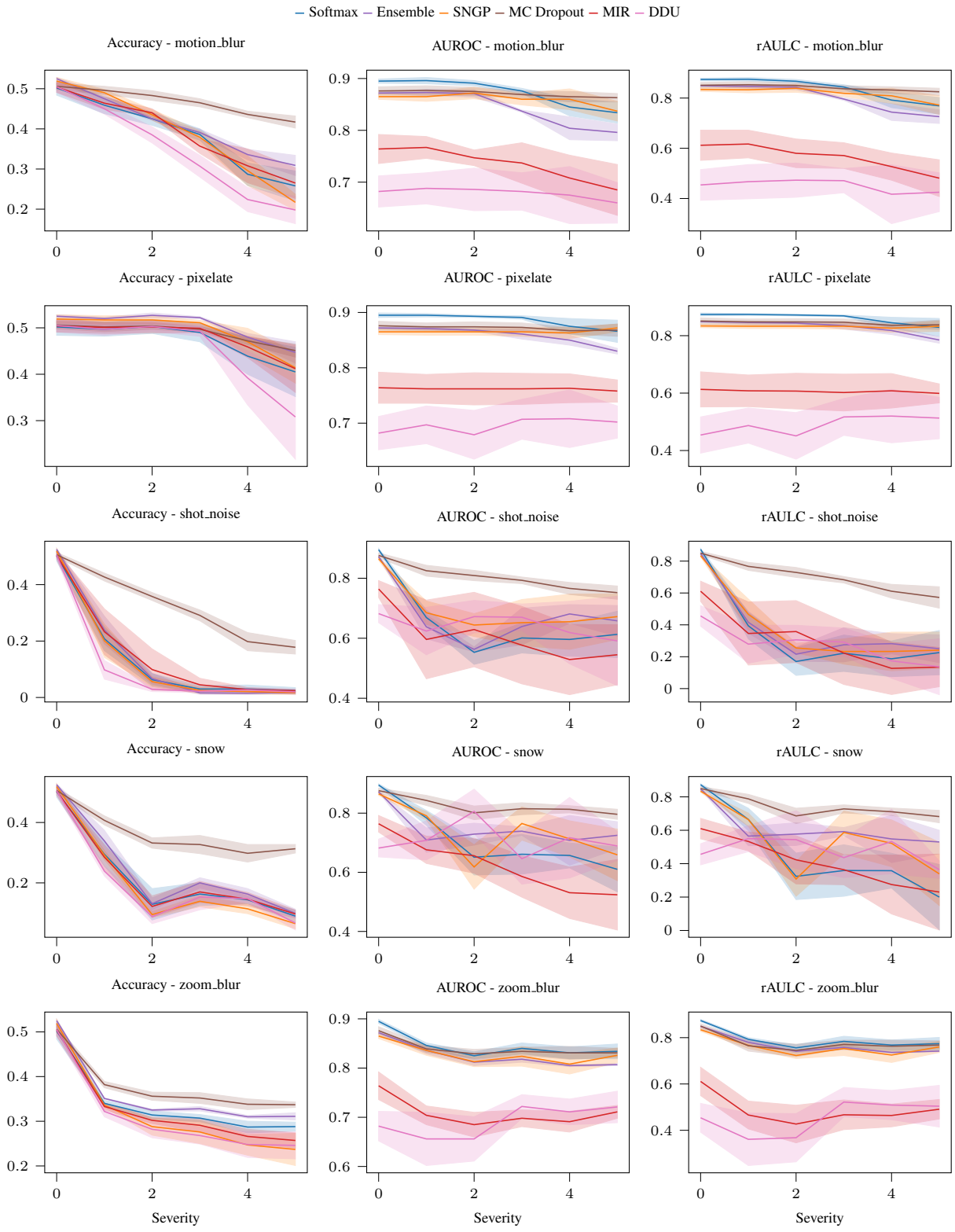


Figure 17. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the Cityscapes-C dataset. We show the mIoU, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): motion blur, pixelate, shot noise, snow, zoom blur.

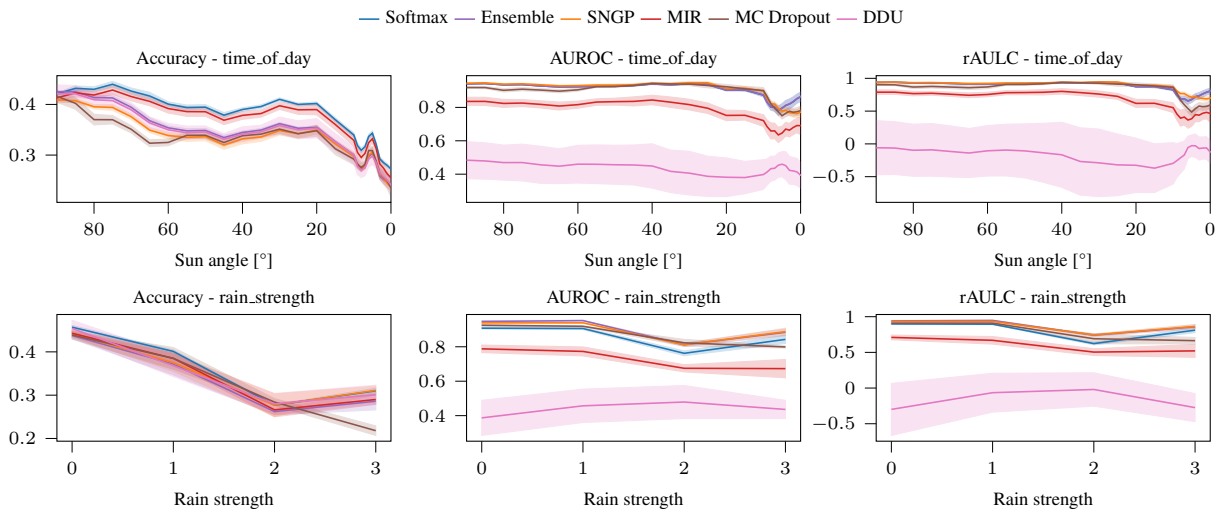


Figure 18. We here compare the performance of DUMs and of the baselines under different corruption types and severities applied on the Carla-C dataset. We show the mIoU, AUROC and rAULC (vertical axis) for each method depending on the corruption severity (horizontal axis) of the following corruption types (listed from top to bottom): time of day, rain.