Uncertainty Estimation by Density Aware Evidential Deep Learning

Taeseong Yoon [1] **Heeyoung Kim** [1]

Abstract

Evidential deep learning (EDL) has shown remarkable success in uncertainty estimation. However, there is still room for improvement, particularly in out-of-distribution (OOD) detection and classification tasks. The limited OOD detection performance of EDL arises from its inability to reflect the distance between the testing example and training data when quantifying uncertainty, while its limited classification performance stems from its parameterization of the concentration parameters. To address these limitations, we propose a novel method called *Density Aware Evidential Deep Learning (DAEDL)*. DAEDL integrates the feature space density of the testing example with the output of EDL during the prediction stage, while using a novel parameterization that resolves the issues in the conventional parameterization. We prove that DAEDL enjoys a number of favorable theoretical properties. DAEDL demonstrates state-of-the-art performance across diverse downstream tasks related to uncertainty estimation and classification.

1. Introduction

In recent years, artificial intelligence (AI) has achieved remarkable advancements, demonstrating state-of-the-art performance across various domains. Despite these significant strides, applying AI models to real-world problems poses challenges. Relying solely on AI models for critical decision-making is considered risky. To safely deploy AI models in high-risk domains such as healthcare, finance, and manufacturing, they must possess the capability to represent the uncertainty of their outcomes accurately. However, it is widely acknowledged that modern neural networks often lack proper calibration, and struggle to precisely represent the uncertainty associated with their predictions (Guo et al., 2017). Various methods, including deep ensemble (Lakshminarayanan et al., 2017), Monte Carlo dropout (Gal & Ghahramani, 2016), and Bayesian neural networks (Blundell et al., 2015), have been proposed to quantify the uncertainty of AI models. These methods exhibit reasonable performance underpinned by a solid theoretical foundation. Nevertheless, their practical applicability in real-world settings is hindered by the necessity of multiple forward passes, making them less feasible. To address this limitation, researchers have explored models with the capability to quantify uncertainty in a single forward pass, aiming to enhance the practical applicability of uncertainty estimation models for real-world problems.

Dirichlet-based uncertainty (DBU) models (Malinin & Gales, 2018; Sensoy et al., 2018; Malinin & Gales, 2019; Charpentier et al., 2020; 2021; Ulmer et al., 2023) have emerged as a promising avenue among models capable of quantifying uncertainty in a single forward pass. Unlike conventional classification models that directly predict class probabilities, DBU models adopt a distinctive approach by predicting the distribution of class probabilities. Evidential deep learning (EDL) (Sensoy et al., 2018), which employs a Dirichlet distribution to simultaneously quantify belief mass for each class and uncertainty mass, stands out as a prominent example of the DBU models. EDL is distinguished for its simplicity in implementation and impressive uncertainty estimation performance in various tasks, especially in out-of-distribution (OOD) detection (Sensoy et al., 2018; Deng et al., 2023).

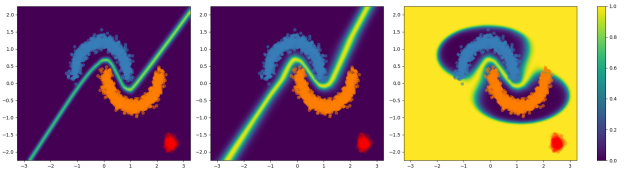
Despite EDL's notable success, there is still room for improvement. First, EDL may fail to accurately reflect the distance between the testing examples and training data when quantifying predictive uncertainty, potentially resulting in a decline in OOD detection performance. To shed light on our hypothesis, we conducted a toy experiment using a two moons dataset (Liu et al., 2020). In Figure 1, uncertainty representations (predictive variance) obtained by Softmax and EDL are depicted in (a) and (b) along with the training data of two classes (blue and orange) and OOD data (red). The ideal uncertainty estimation model should yield low predictive uncertainty for testing examples that are near the training data, with the uncertainty increasing as they move farther away from the training data. However, as illustrated in Figure 1, Softmax and EDL exhibit high uncer-

[1]Department of Industrial and Systems Engineering, KAIST, Daejeon, Republic of Korea. Correspondence to: Heeyoung Kim <heeyoungkim@kaist.ac.kr>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

tainty (yellow) primarily along the decision boundary and low uncertainty (purple) elsewhere. Notably, Softmax and EDL assign low uncertainty even to OOD data. In contrast, our proposed method in (c), which will be detailed below, yielded the desired uncertainty estimation results. Second, EDL exhibits limited classification performance, restricting its suitability for real-world problems, where classification typically takes precedence over uncertainty estimation. We hypothesize that this limitation arises from the conventional parameterization of EDL, specifically the challenge of estimating an appropriate magnitude of the *evidence* in relation to the concentration parameters of the Dirichlet distribution, due to the absence of an explicit range for the magnitude of the evidence.

To address these limitations, we propose a novel method called *Density Aware Evidential Deep Learning (DAEDL)*. First, to enable an uncertainty estimate that reflects the distance between the testing examples and training data, DAEDL integrates the feature space density of the testing example with the output of EDL during the prediction stage. For the density estimation, DAEDL employs Gaussian discriminant analysis (GDA) in the feature space, inspired by Mukhoti et al. (2023), which allows it to estimate the density without additional training. Notably, the feature space encapsulates relevant features essential for both uncertainty estimation and classification, whereas direct density estimation in the input space is computationally demanding and susceptible to the curse of dimensionality (Choi & Jang, 2018; Nalisnick et al., 2019). Second, to overcome the potential limitation of the conventional parameterization of EDL, DAEDL introduces an alternative novel parameterization that resolves the issues arising from the lack of an explicit range for the evidence. Additionally, DAEDL adopts an exponential activation function, in contrast to ReLU in EDL, which allows it to establish a connection with the softmax model, thereby further enhancing classification accuracy.



(a) Softmax **(b) EDL** **(c) DAEDL (ours)**

Figure 1. Uncertainty representations on the two moons dataset. (a) Softmax, (b) EDL, and (c) DAEDL (ours).

We establish that DAEDL exhibits favorable theoretical properties, elucidating the reasons for its superior uncertainty estimation performance over conventional EDL. First, we prove that DAEDL generates a uniform predictive distribution over classes for OOD data. Second, we prove that the predictive distribution of DAEDL can be conceptualized as an *adaptive temperature scaled* softmax model, which has demonstrated effective performance in improving model calibration and OOD detection (Balanya et al., 2022; Joy et al., 2023; Krumpl et al., 2024). Third, we prove that DAEDL can be interpreted as predicting an *input-dependent posterior distribution of the Dirichlet-Categorical model* (Charpentier et al., 2020) with an improper prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{0})$, while typical DBU models (Sensoy et al., 2018; Charpentier et al., 2020; 2021; Deng et al., 2023) can be seen as utilizing a uniform prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{1})$ under this framework. This interpretation implies that DAEDL employs an improper prior in a Bayesian context, thereby addressing the challenge of specifying an appropriate prior distribution. Fourth, we prove that DAEDL's predictive uncertainty of a testing example is proportional to its distance from the training data manifold, under mild assumptions. This property, formally defined as *distance awareness* (Liu et al., 2020), has been demonstrated to enhance the quality of uncertainty estimation.

DAEDL consistently demonstrates state-of-the-art performance across various downstream tasks related to uncertainty estimation, including OOD detection, confidence calibration, and distribution shift detection, as well as achieving superior performance in image classification.

2. Evidential Deep Learning

EDL (Sensoy et al., 2018) stands as one of the pioneering works in the class of DBU models. The development of EDL is grounded in the principles of subjective logic (SL) (Jøsang, 1997; 2016) and Dempster-Shafer Theory of Evidence (DST) (Dempster, 1968; Shafer, 1976). In a classification problem with C classes, DST assigns a belief mass $b_c,\ \forall c \in [C]$ for each class, which measures the evidence in favor of each class, and the uncertainty mass u, which captures the overall uncertainty. These values are all non-negative and subject to the constraint $u + \sum_{c=1}^{C} b_c = 1$. SL models the belief assignment framework of DST using a Dirichlet distribution. The concentration parameters of the Dirichlet distribution, denoted as $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots \alpha_C]$, is parameterized as $\alpha_c = 1 + e_c, \forall c \in [C]$, where e_c denotes the evidence for the cth class. The belief and uncertainty values are computed as $b_c = e_c/\alpha_0$ and $u = C/\alpha_0$, where $\alpha_0 = \sum_{c=1}^{C} \alpha_c$ denotes the precision of the Dirichlet distribution. A higher α_0 corresponds to a sharper and more confident distribution.

EDL performs classification by estimating the evidence vector $\boldsymbol{e} = [e_1, e_2, \cdots, e_C], \forall e_c > 0$, using a neural network. Specifically, the evidence vector of input $\mathbf{x} \in \mathbb{R}^D$ is computed as $\mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = h(g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x})))$, where $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^H$, $g_{\boldsymbol{\phi}} : \mathbb{R}^H \to \mathbb{R}^C$, and $h : \mathbb{R}^C \to \mathbb{R}^C_+$ is the feature extractor, classifier, and activation function, respectively. Here,

$\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ represent the parameters of the feature extractor and classifier, respectively, while D, H, and C denote the dimension of the input data, dimension of feature representations, and the number of classes, respectively. Following the parametrization used in SL, the concentration parameters of the Dirichlet distribution are obtained as

$$\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \mathbf{1} + \mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}), \tag{1}$$

where $\mathbf{1} = [1, 1, \cdots 1] \in \mathbb{R}^C$ is the vector of ones.

EDL is optimized using a loss function that consists of two components: i) expected mean squared error (MSE), responsible for accurate uncertainty-aware classification, and ii) Kullback-Leibler (KL) divergence penalty, ensuring the desired uncertainty behavior of the concentration parameters. For sample $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{y}_i$ is the one-hot encoded label, the loss function is formulated as follows:

$$\begin{aligned}\mathcal{L}^{(i)}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}(\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_i))}[\|\mathbf{y}_i - \boldsymbol{\pi}\|_2^2] \\ + \lambda D_{KL}[\mathrm{Dir}(\tilde{\boldsymbol{\alpha}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_i)) \| \mathrm{Dir}(\mathbf{1})],\end{aligned} \tag{2}$$

where $\tilde{\boldsymbol{\alpha}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_i) = \boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_i) \odot (\mathbf{1} - \mathbf{y}_i) + \mathbf{y}_i$, and λ is a regularization parameter.

3. Density Aware Evidential Deep Learning

3.1. Model Overview

DAEDL closely follows the conventional structure of EDL outlined in Section 2, but introduces two significant modifications designed to address the limitations of EDL: (i) the adoption of an alternative parameterization (Section 3.2), and (ii) the integration of feature space density (Section 3.3). DAEDL is trained using the loss specified in Eq.(2). After training, we obtain the feature extractor $f_{\hat{\boldsymbol{\theta}}}$ and classifier $g_{\hat{\boldsymbol{\phi}}}$ with the optimized parameters $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\phi}}$. The training procedure for DAEDL is presented in Algorithm 1 in Appendix B.

In contrast to conventional EDL, DAEDL employs *spectral normalization* (Miyato et al., 2018) in the feature extractor $f_{\boldsymbol{\theta}}$, facilitating a meaningful density estimate even when a simple density estimator, such as GDA, is used. Spectral normalization has been widely utilized for single forward pass uncertainty estimation models to achieve a regularized feature space (Liu et al., 2020; van Amersfoort et al., 2021; Mukhoti et al., 2023). Notably, we are the first to adopt it for DBU models. By employing spectral normalization, we bound the distance in the feature space by the distance in the input space, preventing feature representations from becoming overly sensitive to meaningless perturbations in the input space (Liu et al., 2020). More detailed descriptions of spectral normalization are provided in Appendix E.

3.2. Alternative Parameterization

We propose a novel parameterization for the concentration parameters of DAEDL, which overcomes the limitations of the conventional scheme in EDL while establishing a connection with the softmax model. The conventional parameterization for the concentration parameters of EDL is expressed as in Eq.(1). This parameterization has inherent limitations due to the challenge of achieving an appropriate balance between $\mathbf{1}$ and $\mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$. As there is no explicit range for $\mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$, $\mathbf{1}$ may dominate $\mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$, potentially leading to counter-intuitive outcomes for the expected class probabilities. For example, in the case with $C = 3$, given a highly likely ID data point $\mathbf{x}_{\mathrm{id}}$ and a highly-peaked evidence vector computed as $\mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_{\mathrm{id}}) = [1, 0, 0]$, the expected class probability is derived as $\bar{\boldsymbol{\pi}}_{\mathrm{id}} = [0.5, 0.25, 0.25]$, contradicting the intuition that the class probability for a highly likely ID data point should be more strongly peaked toward the corresponding class. We hypothesize that these counter-intuitive results may hinder the model from learning the decision boundary accurately, leading to degraded classification performance.

To address the above limitations and enhance the classification performance of EDL, DAEDL introduces two concurrent modifications to the conventional parameterization: i) the removal of $\mathbf{1}$ in Eq.(1) and ii) the adoption of the exponential activation function, instead of ReLU in EDL. We eliminate $\mathbf{1}$ to address the challenge associated with balancing the magnitudes between $\mathbf{1}$ and $\mathbf{e}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$. We argue that $\mathbf{1}$ is not necessarily an essential component in the model, hindering the effectiveness of EDL in the classification task. By removing it, DAEDL allows the model to learn a Dirichlet distribution solely from the data. Subsequently, we adopt an exponential function as the activation function to establish a close connection with the softmax model, replacing ReLU (Sensoy et al., 2018) or Softplus (Deng et al., 2023). Then, the concentration parameters of DAEDL during the training stage can be formulated as $\boldsymbol{\alpha}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \exp\left(g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x}))\right)$. As summarized in Table 1, employing such parameterization aligns the expected class probability of DAEDL with the output of the softmax model. Given that the softmax model generally outperforms conventional EDL in terms of classification accuracy, we expect that DAEDL will improve upon EDL in classification performance.

3.3. Integration of Feature Space Density

DAEDL leverages the feature space density of a testing example during prediction to obtain an uncertainty estimate that reflects its distance from the training data. Specifically, we employ GDA as the density estimator in the feature space, and integrate the estimated feature space density of the testing example with the logits in the prediction stage.

Table 1. Comparison of the concentration parameter (α_c) and expected class probability ($\mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\pi_c]$) during the training stage between the standard EDL (Sensoy et al., 2018; Deng et al., 2023) and DAEDL. $z_c(\mathbf{x}) = (g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x})))_c$ represents the logit for each class $\forall c \in [C]$, and h is an activation function that ensures the non-negativity of the evidence vector (e.g., ReLU (Sensoy et al., 2018) and Softplus (Deng et al., 2023)).

	EDL	**DAEDL**
α_c	$1 + h(z_c(\mathbf{x}))$	$\exp(z_c(\mathbf{x}))$
$\mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\pi_c]$	$\frac{1+h(z_c(\mathbf{x}))}{C+\sum_{c=1}^{C} h(z_c(\mathbf{x}))}$	$\frac{\exp(z_c(\mathbf{x}))}{\sum_{c=1}^{C}\exp(z_c(\mathbf{x}))}$

Density estimation. DAEDL employs GDA as a density estimator in the feature space, inspired by its efficiency and the ability to operate without additional training (Mukhoti et al., 2023). Given the training dataset $\mathcal{D}_{\text{tr}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ and the trained feature extractor $f_{\hat{\boldsymbol{\theta}}}$, the parameters of GDA for each class $\forall c \in [C]$ are obtained as follows:

$$\hat{\omega}_c = \frac{N_c}{N}, \quad \hat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{\{i:y_i=c\}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i),$$

$$\hat{\Sigma}_c = \frac{1}{N_c - 1}\sum_{\{i:y_i=c\}} (f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_c)(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_c)^T,$$

where $\hat{\omega}_c$, $\hat{\boldsymbol{\mu}}_c$, and $\hat{\Sigma}_c$ represent the weight, mean vector, and covariance matrix for each class, respectively. In addition, N denotes the total number of training data points, and $N_c = \sum_{i=1}^{N} \mathbb{1}_{\{y_i=c\}}$ represents the number of data points for each class $\forall c \in [C]$. The density estimation algorithm for DAEDL is provided in Algorithm 2 in Appendix B.

Prediction. In the prediction stage, we estimate the feature space density of the testing example and integrate it with the output of EDL to obtain a reliable uncertainty estimate that reflects the distance between the testing example and the training data. The graphical representation of the prediction stage is depicted in Figure 2.

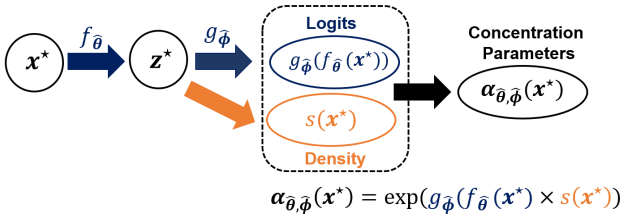


Figure 2. Graphical representation of the prediction stage of DAEDL. The process begins with the computation of the logits (blue), followed by the estimation of the normalized feature space density of the testing example (orange). These two components are integrated to derive the concentration parameters (black).

First, we estimate the feature space density of a testing example using a fitted GDA model on the training data. For testing example $\mathbf{x}^\star \in \mathbb{R}^D$, the feature space density is obtained as follows:

$$p(\mathbf{z}^\star = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star)) = \sum_{c=1}^{C} \hat{\omega}_c \, \mathcal{N}(\mathbf{z}^\star = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star) | \hat{\boldsymbol{\mu}}_c, \hat{\Sigma}_c),$$

where $\mathbf{z}^\star \in \mathbb{R}^H$ is a feature representation of $\mathbf{x}^\star$. In practice, we first calculate the logarithm of the feature space density, and then normalize it to a range between $[0, 1]$ to avoid challenges related to parameter divergence, leveraging the minimum and maximum values observed in the logarithm of the feature space density of the training data $\mathcal{X}_{\text{tr}} = \{\mathbf{x}_i\}_{i=1}^{N}$ and the Clip function defined as $\mathrm{Clip}(x) = \max(0, \min(1, x))$. The normalized feature space density for $\mathbf{x}^\star$ is expressed as follows:

$$s(\mathbf{x}^\star) = \mathrm{Clip}\left(\frac{\log p(\mathbf{z}^\star = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star)) - d_{\min}}{d_{\max} - d_{\min}}\right),$$

where $d_{\min} = \min_{\mathbf{x}\in\mathcal{X}_{tr}} \left\{\log p(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}))\right\}$ and $d_{\max} = \max_{\mathbf{x}\in\mathcal{X}_{tr}} \left\{\log p(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}))\right\}$.

Subsequently, we combine the normalized feature space density (i.e., $s(\mathbf{x}^\star)$) with the logits (i.e., $g_{\hat{\boldsymbol{\phi}}}(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star))$ to obtain the concentration parameters of DAEDL. Specifically, we multiply the normalized feature space density with the logits, and apply the exponential activation function to derive the concentration parameters as follows:

$$\boldsymbol{\alpha}_{\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\phi}}}(\mathbf{x}^\star) = \exp\left(g_{\hat{\boldsymbol{\phi}}}(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star)) \times s(\mathbf{x}^\star)\right).$$

This process can be interpreted as scaling the logits with the estimated confidence level of the prediction before applying the activation function. This integration strategy demonstrates effective empirical performance, underpinned by favorable theoretical properties, including Theorem 4.3, Theorem 4.4, and Corollary 4.5. The algorithm for the prediction of DAEDL is provided in Algorithm 3 in Appendix B.

4. Theoretical Analysis

We establish the theoretical foundations of DAEDL. First, we prove that DAEDL generates a uniform predictive distribution over classes for highly likely OOD testing examples (Theorem 4.1). This indicates that DAEDL effectively quantifies uncertainty for OOD data, whereas EDL fails to quantify uncertainty for testing examples distant from the training data, as illustrated in Figure 1.

Second, we prove that DAEDL can be interpreted as predicting an *input-dependent posterior distribution of Dirichlet-Categorical model* (Charpentier et al., 2020) with an improper prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{0})$ in the Bayesian context (Theorem 4.2). This indicates that DAEDL addresses the chal-

lenge of prior specification by using an improper prior, enabling the model to learn the Dirichlet distribution solely from the data.

Third, we prove that the predictive distribution of DAEDL can be conceptualized as an *adaptive temperature scaled* (Joy et al., 2023) softmax model (Theorem 4.3). This indicates that DAEDL inherits the advantages of adaptive temperature scaling, which is known for its effectiveness in enhancing model calibration and OOD detection performance (Balanya et al., 2022; Joy et al., 2023; Krumpl et al., 2024).

Finally, we prove that the predictive uncertainty of a testing example estimated using DAEDL is proportional to the distance between the testing example and training data manifold in both the feature space (Theorem 4.4) and input space (Corollary 4.5) under mild conditions. This property is formally defined as *distance awareness* (Liu et al., 2020) and has been established as a beneficial condition for obtaining high-quality uncertainty estimates.

Theorem 4.1. ***(Uniform Prediction for OOD Data)*** *As the distance between the testing example $\mathbf{x}^\star_{\text{ood}}$ and training data in the input space diverges, i.e., $\mathbb{E}_{\mathbf{x}'\sim\mathcal{X}_{\text{tr}}}\|\mathbf{x}^\star_{\text{ood}}-\mathbf{x}'\|_2 \to \infty$, the predictive distribution of DAEDL converges to the uniform distribution over classes $\forall c \in [C]$, i.e., $p(y|\mathbf{x}^\star_{\text{ood}}) \to \mathcal{U}\{1, C\}$. Additionally, the concentration parameters of DAEDL converges to $\mathbf{1}$, i.e., $\boldsymbol{\alpha}(\mathbf{x}^\star_{\text{ood}}) \to \mathbf{1}$.*

Theorem 4.2. ***(Bayesian Interpretation of DAEDL)*** *In the Bayesian context, DAEDL can be interpreted as predicting an input-dependent posterior distribution of the Dirichlet-Categorical model with an improper prior $\boldsymbol{\pi} \sim \text{Dir}(\mathbf{0})$.*

Theorem 4.3. ***(Relationship with Temperature Scaling)*** *The predictive distribution of DAEDL aligns with the adaptive temperature scaled softmax model:*

$$p(y|\mathbf{x}^\star) = \text{Cat}(\bar{\boldsymbol{\pi}}), \quad \bar{\boldsymbol{\pi}} = \sigma\left(\mathbf{z}(\mathbf{x}^\star)/T(\mathbf{x}^\star)\right),$$

where $\mathbf{z}(\mathbf{x}^\star) = g_{\hat{\boldsymbol{\phi}}}(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star))$ is the logits and $T(\mathbf{x}^\star) = 1/s(\mathbf{x}^\star)$ is a sample-dependent temperature.

Theorem 4.4. ***(Feature Distance Awareness)*** *The predictive distribution of the testing example $\mathbf{x}^\star$ obtained by DAEDL is distance aware in the feature space, i.e., $u(\mathbf{z}^\star) = \nu\big(\mathbb{E}_{\mathbf{z}'\sim\mathcal{Z}_{\text{tr}}}\|\mathbf{z}^\star - \mathbf{z}'\|_2\big)$, where u is an uncertainty measure, ν is a monotonic function, $\mathbf{z}^\star = f_{\boldsymbol{\theta}}(\mathbf{x}^\star)$ is the feature representation of $\mathbf{x}^\star$, and $\mathcal{Z}_{tr}$ is the set of feature representations of the training data.*

Corollary 4.5. ***(Input Distance Awareness)*** *If $f_{\boldsymbol{\theta}}$ is constructed using residual blocks (e.g., ResNet), the predictive distribution of $\mathbf{x}^\star$ obtained by DAEDL is distance aware in the input space, i.e., $u(\mathbf{x}^\star) = \nu\big(\mathbb{E}_{\mathbf{x}'\sim\mathcal{X}_{\text{tr}}}\|\mathbf{x}^\star - \mathbf{x}'\|_2\big)$.*

Assumptions, proofs, and more detailed explanations of the theoretical results are provided in Appendix A.

5. Related Works

DBU models. Distinctions among DBU models have emerged in various aspects, including parameterization, the requirement of OOD data, loss functions, and regularizers. KL-PN (Malinin & Gales, 2018) is trained to minimize simultaneously the KL divergence towards a peaked Dirichlet distribution for ID data and a uniform Dirichlet distribution for OOD data. RKL-PN (Malinin & Gales, 2019) employs the reverse KL divergence instead, arguing that using the KL divergence results in a multimodal target distribution, leading to undesirable uncertainty representation. Nandy et al. (2020) argued that RKL-PN struggles to distinguish ID data with high aleatoric uncertainty (Malinin & Gales, 2018) from OOD data and proposed a novel loss function to maximize the representation gap between ID and OOD data. However, these models require OOD data for training, which is often an unrealistic assumption in practice.

The Posterior Network (PostNet) (Charpentier et al., 2020) predicts the posterior Dirichlet distribution by utilizing feature space density estimated through Normalizing Flow (Rezende & Mohamed, 2015). The Natural Posterior Network (NatPN) (Charpentier et al., 2021) extends the PostNet to arbitrary distributions within the exponential family. However, because these models heavily rely on the feature space density for uncertainty quantification, they may encounter difficulties in practical scenarios where obtaining high-quality feature space is not always feasible.

EDL (Sensoy et al., 2018), as discussed in Section 2, is another notable instance of the DBU model. Haußmann et al. (2019) proposed a Bayesian version of EDL trained with the marginal likelihood regularized by the Probably Approximately Correct bound regularizer. Additionally, Tsiligkaridis (2021) suggested using the l_p loss combined with Rényi divergence regularizer. $\mathcal{I}$-EDL (Deng et al., 2023) incorporated Fisher information to weigh the importance of each class during training, demonstrating significant performance gains over EDL. Moreover, alternative DBU models exist, adopting artificial OOD data generation for training (Sensoy et al., 2020; Hu et al., 2021), knowledge distillation (Malinin et al., 2019; Fathullah & Gales, 2022), or posterior Dirichlet distribution prediction by variational inference (Chen et al., 2018; Joo et al., 2020). However, the uncertainty quantified by these models may fail to reflect the distance between the testing example and the training data, hindering their effectiveness in OOD detection.

Other single forward pass uncertainty models. Alternative models capable of quantifying uncertainty in a single forward pass typically involve i) regularization of the feature space, and ii) uncertainty estimation. To obtain the regularized feature space, spectral normalization (Liu et al., 2020; van Amersfoort et al., 2021; Kotelevskii et al., 2022; Mukhoti et al., 2023) and gradient penalty (Van Amers-

foort et al., 2020) have been used. Subsequently, Gaussian process (Liu et al., 2020; van Amersfoort et al., 2021), Radial Basis Function network (Van Amersfoort et al., 2020), GDA (Mukhoti et al., 2023), and kernel density estimator (Kotelevskii et al., 2022) have been used for estimating uncertainty in a single forward pass. However, these models often require substantial modifications to the model structure or additional computational costs for uncertainty estimation.

6. Experiments

6.1. Experimental Settings

Tasks. We conducted extensive experiments across various downstream tasks related to uncertainty estimation and classification. Our primary goal was to evaluate whether DAEDL successfully addressed the limitations of conventional EDL, thereby enhancing uncertainty estimation and classification performance. Additionally, we aimed to empirically validate the theoretical advancements presented in Section 4. The specific questions explored through the experiments are outlined below, with the most relevant properties listed in parentheses.

- *Q1*. Does DAEDL improve uncertainty estimation by leveraging feature space density? (Theorem 4.1)
- *Q2*. Does DAEDL improve classification by using the new parameterization? (Theorem 4.2)
- *Q3*. Does DAEDL improve confidence calibration? (Theorem 4.3)
- *Q4*. Does the uncertainty quantified by DAEDL reflect the distance between the testing example and training data? (Theorem 4.4, Corollary 4.5)

To address *Q1*, we performed OOD detection (Section 6.2) as a downstream task to evaluate the quality of uncertainty estimation. *Q2* was evaluated by an image classification (Section 6.3) task. For *Q3*, a confidence calibration (Section 6.3) task was executed in two manners: i) conducting a misclassified image detection task, and ii) measuring the Brier score. To address *Q4*, we conducted distribution shift detection (Section 6.4). Similar to *Q1*, OOD detection was applied as a downstream task to assess the quality of the uncertainty estimate. Specifically, we systematically performed a series of OOD detection tasks using progressively generated OOD datasets. These datasets were created by applying various types of corruption to ID data, with the severity of corruption increasing sequentially. Note that each experiment not only evaluates the corresponding question but also relates to other questions. For instance, OOD detection is also related to *Q3* and *Q4*. For each experiment, the mean and standard deviation of the results averaged over five runs were reported. The code for our model is available at https://github.com/TaeseongYoon/DAEDL.

Datasets. To evaluate the OOD detection performance, we used MNIST (LeCun, 1998) and CIFAR-10 (Krizhevsky et al., 2009) as ID datasets. We used FMNIST (Xiao et al., 2017) and KMNIST (Clanuwat et al., 2018) as OOD datasets for MNIST. For CIFAR-10, we used SVHN (Netzer et al., 2011) and CIFAR-100 (Krizhevsky et al., 2009) as OOD datasets. We evaluated the classification accuracy and confidence calibration performance of our model using MNIST and CIFAR-10. To evaluate the performance of distribution shift detection, we employed MNIST-C (Mu & Gilmer, 2019a) and CIFAR-10-C (Hendrycks & Dietterich, 2019) as the OOD dataset for MNIST and CIFAR-10, respectively. MNIST-C and CIFAR-10-C are datasets created by applying continuous distribution shifts to MNIST and CIFAR-10, respectively. More detailed descriptions of the datasets are provided in Appendix C.1.

Implementation. For a fair comparison, we followed the settings of Charpentier et al. (2020) and Deng et al. (2023). We used 3 convolutional layers and 3 dense layers for MNIST, and used VGG-16 (Simonyan & Zisserman, 2014) for CIFAR-10. The optimal hyperparameters were determined through grid search. Additionally, we used a learning rate scheduler and early stopping based on the validation loss. More detailed explanations of the implementation are provided in Appendix C.2.

Baselines. We compared our model with representative DBU models. Following the setup of Deng et al. (2023), our evaluation included the following competing models: KL-PN (Malinin & Gales, 2018), RKL-PN (Malinin & Gales, 2019), PostNet (Charpentier et al., 2020), EDL (Sensoy et al., 2018), and $\mathcal{I}$-EDL (Deng et al., 2023). In Section 6.2 and Section 6.3, we also included a comparison with Dropout (Gal & Ghahramani, 2016), which still demonstrates state-of-the-art uncertainty estimation performance in various tasks. In Section 6.4, we further compared our model with MSP (Hendrycks & Gimpel, 2016), a standard baseline for distribution shift detection. Following Charpentier et al. (2020) and Deng et al. (2023), we used the set of data points generated by uniform noise as an OOD dataset for KL-PN and RKL-PN, which require an OOD dataset for training, to ensure a fair comparison.

6.2. OOD Detection

We evaluated the OOD detection performance of DAEDL in comparison to baseline methods. Our evaluation consisted of two steps. First, we calculated OOD scores for both the ID test dataset and the OOD dataset using specific uncertainty measures. Following Charpentier et al. (2020) and Deng et al. (2023), for DBU models, we used $\max_c\{\mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\pi_c]\}$ (i.e., maximum expected class proba-

Table 2. AUPR scores of OOD detection based on aleatoric and epistemic uncertainty. A → B denotes that A is employed as an ID dataset, while B is utilized as an OOD dataset. "ALEA." and "EPIS." indicate that the results were obtained by employing aleatoric and epistemic uncertainty measures as an OOD score, respectively. The first four lines, excluding the results of CIFAR-10 → CIFAR-100, were obtained from Charpentier et al. (2020). The remaining results were obtained from Deng et al. (2023).

	MNIST → KMNIST		MNIST → FMNIST		CIFAR-10 → SVHN		CIFAR-10 → CIFAR-100	
	ALEA.	EPIS.	ALEA.	EPIS.	ALEA.	EPIS.	ALEA.	EPIS.
DROPOUT	94.00 ± 0.1	-	96.56 ± 0.2	-	51.39 ± 0.1	-	45.57 ± 1.0	-
KL-PN	92.97 ± 1.2	93.39 ± 1.0	98.44 ± 1.0	98.16 ± 0.0	43.96 ± 1.9	43.23 ± 2.3	61.41 ± 2.8	61.53 ± 3.4
RKL-PN	60.76 ± 2.9	53.76 ± 3.4	78.45 ± 3.1	72.18 ± 3.6	53.61 ± 1.1	49.37 ± 0.8	55.42 ± 2.6	54.74 ± 2.8
POSTNET	95.75 ± 0.2	94.59 ± 0.3	97.78 ± 0.2	97.24 ± 0.3	80.21 ± 0.2	77.71 ± 0.3	81.96 ± 0.8	82.06 ± 0.8
EDL	97.02 ± 0.8	96.31 ± 2.0	98.10 ± 0.4	98.08 ± 0.4	78.87 ± 3.5	79.12 ± 3.7	84.30 ± 0.7	84.18 ± 0.7
$\mathcal{I}$-EDL	98.34 ± 0.2	98.33 ± 0.2	98.89 ± 0.3	98.86 ± 0.3	83.26 ± 2.4	82.96 ± 2.2	85.35 ± 0.7	84.84 ± 0.6
DAEDL	$\mathbf{99.90 \pm 0.0}$	$\mathbf{99.92 \pm 0.0}$	$\mathbf{99.83 \pm 0.0}$	$\mathbf{99.87 \pm 0.0}$	$\mathbf{85.50 \pm 1.4}$	$\mathbf{85.54 \pm 1.4}$	$\mathbf{88.16 \pm 0.1}$	$\mathbf{88.19 \pm 0.1}$

bility) to measure aleatoric uncertainty, while we used α_0 (i.e., precision of the Dirichlet distribution) as the measure of epistemic uncertainty. For Dropout, we adopted $\max_c \pi_c$ (i.e., maximum class probability) as the aleatoric uncertainty measure. Second, using the calculated OOD scores, we computed the area under the precision-recall curve (AUPR) score, assigning label 1 to ID and 0 to OOD data, to evaluate the OOD detection performance.

Based on the AUPR scores reported in Table 2, DAEDL demonstrates state-of-the-art performance across all evaluated tasks. Specifically, DAEDL outperformed the runner-up method ($\mathcal{I}$-EDL) by noteworthy margins, achieving improvements of 1.56, 1.59, 0.94, and 1.01 on MNIST, as well as 2.24, 2.58, 2.81, and 3.35 on CIFAR-10. OOD detection results with an additional performance metric, the area under the receiver operating characteristic curve (AUROC), are provided in Appendix D.1.

6.3. Image Classification & Confidence Calibration

We conducted both image classification and confidence calibration tasks using DAEDL and the baseline methods. For the image classification task, we evaluated performance using the test accuracy. For the confidence calibration task, we evaluated performance in two different ways. First, we evaluated the misclassified image detection performance using the AUPR score. Specifically, we first split the ID test dataset into two groups based on the classification results: one with correctly classified data and the other with misclassified data. Then, we calculated the confidence score for each group using a specific confidence measure: $\max_c \{\mathbb{E}_{\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha})}[\pi_c]\}$ for DBU models and $\max_c \pi_c$ for Dropout, following Charpentier et al. (2020) and Deng et al. (2023). Using the calculated confidence scores, we computed the AUPR scores with label 1 for correctly classified data and 0 for misclassified ones, to evaluate the misclassified image detection performance. Second, we measured the Brier score (Brier, 1950), which is a standard metric used to assess the calibration of the model (Gneiting & Raftery, 2007). A lower value of the Brier score indicates better performance.

As shown in Table 3, DAEDL exhibits state-of-the-art performance in image classification, outperforming the runner-up method ($\mathcal{I}$-EDL) by a significant margin of 1.91. Moreover, DAEDL achieved state-of-the-art performance in confidence calibration, surpassing the respective runner-up methods, $\mathcal{I}$-EDL and PostNet, by 0.36 in the AUPR score and by 8.57 in the Brier score. Despite the primary focus of the DBU models on uncertainty estimation, achieving high classification accuracy remains crucial. Therefore, DAEDL holds a distinct advantage by achieving the best performance in both classification and confidence calibration.

Table 3. The results of image classification and confidence calibration on CIFAR-10. The first four lines present the results from Charpentier et al. (2020). The test accuracy and AUPR of EDL and $\mathcal{I}$-EDL are obtained from Deng et al. (2023).

	TEST ACC.	AUPR	BRIER
DROPOUT	82.84 ± 0.1	97.15 ± 0.0	27.15 ± 0.2
KL-PN	27.46 ± 1.7	50.61 ± 4.0	87.28 ± 1.0
RKL-PN	64.76 ± 0.3	86.11 ± 0.4	54.73 ± 0.4
POSTNET	84.85 ± 0.0	97.76 ± 0.0	22.84 ± 0.0
EDL	83.55 ± 0.6	97.86 ± 0.2	33.38 ± 2.0
$\mathcal{I}$-EDL	89.20 ± 0.3	98.72 ± 0.1	35.20 ± 0.8
DAEDL	$\mathbf{91.11 \pm 0.2}$	$\mathbf{99.08 \pm 0.0}$	$\mathbf{14.27 \pm 0.2}$

6.4. Distribution Shift Detection

We conducted distribution shift detection using an OOD dataset created by applying distribution shift (corruption) to the ID dataset. We used the static MNIST-C (Mu & Gilmer, 2019a) and CIFAR-10-C (Hendrycks & Dietterich, 2019) datasets as OOD datasets, paired with the MNIST and CIFAR-10 datasets as ID datasets, respectively. The static MNIST-C dataset has a fixed severity level of corruption,

Table 4. AUPR scores of distribution shift detection based on aleatoric uncertainty. $\mathcal{C} \in \{1, 2, 3, 4, 5\}$ denotes the severity level of the corruptions in CIFAR-10-C. The results are averaged over 19 different corruptions for each severity level.

	MNIST → MNIST-C	**CIFAR-10 → CIFAR-10-C**				
		$\mathcal{C} = 1$	$\mathcal{C} = 2$	$\mathcal{C} = 3$	$\mathcal{C} = 4$	$\mathcal{C} = 5$
MSP	78.54 ± 0.3	56.39 ± 0.7	61.88 ± 1.1	65.86 ± 1.3	69.91 ± 1.5	75.01 ± 1.8
EDL	82.75 ± 0.8	54.76 ± 0.3	59.01 ± 0.4	62.46 ± 0.5	65.87 ± 0.6	70.21 ± 0.8
$\mathcal{I}$-EDL	86.06 ± 0.5	56.33 ± 0.2	61.52 ± 0.5	65.44 ± 0.5	69.45 ± 0.5	74.56 ± 0.5
DAEDL	$\mathbf{92.43 \pm 0.3}$	$\mathbf{57.89 \pm 0.3}$	$\mathbf{63.23 \pm 0.4}$	$\mathbf{67.53 \pm 0.4}$	$\mathbf{72.21 \pm 0.4}$	$\mathbf{77.74 \pm 0.4}$

Table 5. Ablation study results on CIFAR-10. "AUPR" represents the performance of misclassified image detection on the CIFAR-10 dataset. The results of EDL (DAEDL without EXP, DE, and SN) are from Deng et al. (2023)

					CIFAR10 → SVHN		**CIFAR-10 → CIFAR-100**	
EXP	**DE**	**SN**	**TEST.ACC**	**AUPR**	**ALEA.**	**EPIS.**	**ALEA.**	**EPIS.**
✗	✗	✗	83.55 ± 0.6	97.86 ± 0.2	78.87 ± 3.5	79.12 ± 3.7	84.30 ± 0.7	84.18 ± 0.7
✓	✗	✗	88.59 ± 0.4	98.42 ± 0.1	80.39 ± 2.0	80.45 ± 1.9	83.62 ± 0.9	83.67 ± 0.9
✓	✓	✗	88.59 ± 0.4	99.01 ± 0.0	85.02 ± 0.7	85.04 ± 0.7	87.48 ± 0.2	87.50 ± 0.1
✓	✗	✓	$\mathbf{91.11 \pm 0.2}$	99.04 ± 0.0	84.53 ± 0.9	84.55 ± 0.9	87.52 ± 0.2	87.54 ± 0.2
✓	✓	✓	$\mathbf{91.11 \pm 0.2}$	$\mathbf{99.08 \pm 0.0}$	$\mathbf{85.50 \pm 1.4}$	$\mathbf{85.54 \pm 1.4}$	$\mathbf{88.16 \pm 0.1}$	$\mathbf{88.19 \pm 0.1}$

whereas the CIFAR-10-C dataset has five severity levels of corruption. In the CIFAR-10-C experiments, our objective was to assess the performance of DAEDL in detecting both the presence of corruption and its severity. We measured the performance using AUPR.

As shown in Table 4, DAEDL achieved state-of-the-art performance across all tasks. Specifically, DAEDL outperformed the runner-up method (MSP) by 6.37 in the MNIST-C experiments, and by 1.50, 1.35, 1.67, 2.30, and 2.73 in the CIFAR-10-C experiments for the severity level $\mathcal{C}$ of 1, 2, 3, 4, and 5, respectively. Notably, the performance of EDL and $\mathcal{I}$-EDL was even worse than MSP, which is based on the softmax model, known for its inefficacy in quantifying uncertainty. This inferior performance of EDL models may arise from the unique challenges of distribution shift detection. In typical OOD detection tasks, large distances between OOD testing examples and training data allow for effective discrimination, even if the uncertainty estimates do not accurately reflect the distance. However, in distribution shift detection, OOD data undergo a distribution shift from ID data, with only minor distances from the ID data. Therefore, precise uncertainty estimation is particularly crucial for successful distribution shift detection. DAEDL effectively conducted distribution shift detection by leveraging the feature space density of the testing example to account for its distance from the training data. More experimental results, including additional uncertainty measures, performance measures, and graphical representations of the results for each corruption scenario, are provided in Appendix D.2.

6.5. Ablation Study

We conducted an ablation study of DAEDL on CIFAR-10 to evaluate the contributions of its key components: i) alternative parameterization (EXP), ii) feature space density integration (DE), and iii) spectral normalization (SN). We conducted OOD detection, image classification, and misclassified image detection, using the same OOD datasets and procedures described in Section 6.2 and Section 6.3. The results, presented in Table 5, demonstrate that all components of DAEDL significantly contributed to enhancing overall performance. Specifically, EDL equipped with the proposed parameterization achieved significantly higher classification accuracy, while maintaining comparable uncertainty estimation performance to that of EDL. Moreover, the integration of feature space density resulted in a significant enhancement in uncertainty estimation performance. Furthermore, the application of spectral normalization resulted in an overall improvement in DAEDL's performance. The ablation study results for MNIST are provided in Appendix D.3.

7. Conclusion

We proposed a novel method, DAEDL, which improves the classification and OOD detection performance of EDL. DAEDL achieves this improvement by incorporating the feature space density of testing examples during prediction and introducing a new parameterization of the concentration parameters of the Dirichlet distribution. We demonstrated the effectiveness of DAEDL both theoretically and empirically, showcasing its favorable theoretical properties and

state-of-the-art performance in various downstream tasks related to uncertainty estimation and classification. A potential future research direction includes extending DAEDL for regression tasks.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2023R1A2C2005453, RS-2023-00218913).

Impact Statement

Accurate uncertainty estimation is crucial for ensuring the safe deployment of AI models, particularly in high-risk domains such as healthcare, finance, and manufacturing. However, existing uncertainty estimation models encounter practical challenges, such as the need for multiple forward passes, substantial modifications to the neural network structure, limited classification performance, and sensitive hyperparameters. In this paper, we introduce DAEDL, a novel approach capable of producing high-quality uncertainty estimates in a single forward pass. DAEDL is comprised of detachable components easily integrable into existing network structures, and excels in classification tasks while operating with non-sensitive hyperparameters. Moreover, DAEDL demonstrates remarkable performance in OOD and distribution shift detection, highlighting its potential impact in addressing real-world challenges effectively. Despite its promising advancements, it is essential to acknowledge the limitations of DAEDL. In complex scenarios, EDL outputs and the feature space density may not accurately capture uncertainty, potentially limiting DAEDL's effectiveness. Therefore, practitioners must carefully assess the suitability of DAEDL for their specific problem domains before solely relying on its outcomes.

References

Balanya, S. A., Maroñas, J., and Ramos, D. Adaptive temperature scaling for robust calibration of deep neural networks. *arXiv preprint arXiv:2208.00461*, 2022.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Bui, H. M. and Liu, A. Density-softmax: Scalable and distance-aware uncertainty estimation under distribution shifts. *arXiv preprint arXiv:2302.06495*, 2023.

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 uncertainty for exponential family distributions. *arXiv preprint arXiv:2105.04471*, 2021.

Chen, W., Shen, Y., Jin, H., and Wang, W. A variational dirichlet framework for out-of-distribution detection. *arXiv preprint arXiv:1811.07308*, 2018.

Choi, H. and Jang, E. Generative ensembles for robust anomaly detection. 2018.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

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

Deng, D., Chen, G., Yu, Y., Liu, F., and Heng, P.-A. Uncertainty estimation by fisher information-based evidential deep learning. In *International Conference on Machine Learning*, pp. 7596–7616. PMLR, 2023.

Fathullah, Y. and Gales, M. J. Self-distribution distillation: efficient uncertainty estimation. In *Uncertainty in Artificial Intelligence*, pp. 663–673. PMLR, 2022.

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. PMLR, 2016.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Haußmann, M., Gerwinn, S., and Kandemir, M. Bayesian evidential deep learning with pac regularization. *arXiv preprint arXiv:1906.00816*, 2019.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hu, Y., Ou, Y., Zhao, X., Cho, J.-H., and Chen, F. Multidimensional uncertainty-aware evidential neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7815–7822, 2021.

Joo, T., Chung, U., and Seo, M.-G. Being bayesian about categorical probability. In *International conference on machine learning*, pp. 4950–4961. PMLR, 2020.

Jøsang, A. Artificial reasoning with subjective logic. In *Proceedings of the second Australian workshop on commonsense reasoning*, volume 48, pp. 34. Citeseer, 1997.

Jøsang, A. *Subjective logic*, volume 3. Springer, 2016.

Joy, T., Pinto, F., Lim, S.-N., Torr, P. H., and Dokania, P. K. Sample-dependent adaptive temperature scaling for improved calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14919–14926, 2023.

Kotelevskii, N., Artemenkov, A., Fedyanin, K., Noskov, F., Fishkov, A., Shelmanov, A., Vazhentsev, A., Petiushko, A., and Panov, M. Nonparametric uncertainty quantification for single deterministic neural network. *Advances in Neural Information Processing Systems*, 35:36308–36323, 2022.

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

Krumpl, G., Avenhaus, H., Possegger, H., and Bischof, H. Ats: Adaptive temperature scaling for enhancing out-of-distribution detection methods. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3864–3873, 2024.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

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

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

Malinin, A. and Gales, M. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

Malinin, A., Mlodozeniec, B., and Gales, M. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

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

Mu, N. and Gilmer, J. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019a.

Mu, N. and Gilmer, J. MNIST-C: A Robustness Benchmark for Computer Vision, June 2019b. URL https://doi.org/10.5281/zenodo.3239543.

Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P. H., and Gal, Y. Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24384–24394, 2023.

Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2019.

Nandy, J., Hsu, W., and Lee, M. L. Towards maximizing the representation gap between in-domain & out-of-distribution examples. *Advances in Neural Information Processing Systems*, 33:9239–9250, 2020.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Sensoy, M., Kaplan, L., and Kandemir, M. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Sensoy, M., Kaplan, L., Cerutti, F., and Saleki, M. Uncertainty-aware deep classifiers using generative models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5620–5627, 2020.

Shafer, G. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tsiligkaridis, T. Information aware max-norm dirichlet networks for predictive uncertainty estimation. *Neural Networks*, 135:105–114, 2021.

Ulmer, D., Hardmeier, C., and Frellsen, J. Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation. *Transactions on Machine Learning Research*, 2023.

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. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

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

Appendix for the Paper
"Uncertainty Estimation by Density Aware Evidential Deep Learning"

A. Additional Explanation for Theoretical Analysis

In this section, we provide an additional explanation about the theoretical analysis (Section 4) of DAEDL. First, we outline the assumptions and lemmas that are utilized throughout the study. Second, we present detailed proofs of the theorems. Third, we provide an additional description of the Bayesian interpretation of DAEDL, which is established in Theorem 4.2. Finally, we offer additional insights into Theorem 4.3, clarifying the relationship between EDL, AdaTS, and DAEDL. Here, $\mathcal{X}_{\text{tr}}$ and $\mathcal{Z}_{tr} = \{\mathbf{x} = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}),\ \forall \mathbf{x} \in \mathcal{X}_{\text{tr}}\}$ denote the sets of training data and their feature representations. In addition, $\mathbf{x}^\star$ and $\mathbf{z}^\star = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star)$ represent the testing example and its feature representation, respectively.

A.1. Assumptions & Lemmas

We state the assumptions that are required to prove the theorems. First, we assume that if the distance between the testing example and the training data diverges in the input space, it will also diverge in the feature space (Assumption A.1). Assumption A.1 is used to prove Theorem 4.1. Second, we assume that the normalized feature space density (i.e., $s(\mathbf{x}^\star)$) is monontonically decreasing with respect to the distance between the testing example and the training data in the feature space (i.e., $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{\text{tr}}} \|\mathbf{z}^\star - \mathbf{z}'\|_2$) (Assumption A.2). Assumption A.2 is used to prove Theorem 4.4 and Corollary 4.5.

Assumption A.1. If $\mathbb{E}_{\mathbf{x}' \sim \mathcal{X}_{\text{tr}}} \|\mathbf{x}^\star - \mathbf{x}'\|_2 \to \infty$, then $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{\text{tr}}} \|\mathbf{z}^\star - \mathbf{z}'\|_2 \to \infty$.

Assumption A.2. $s(\mathbf{x}^\star)$ is monotonically decreasing with respect to $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{\text{tr}}} \|\mathbf{z}^\star - \mathbf{z}'\|_2$.

We provide the lemmas that are required to prove the theorems.

Lemma A.3. *If spectral normalization is applied to $f_{\boldsymbol{\theta}}$, then $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{tr}} \|\mathbf{z}^\star - \mathbf{z}'\|_2$ is bounded by $\mathbb{E}_{\mathbf{x}' \sim \mathcal{X}_{tr}} \|\mathbf{x}^\star - \mathbf{x}'\|_2$.*

Proof. If spectral normalization is applied to $f_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$ is 1-Lipschitz continuous (Miyato et al., 2018). In other words, for data points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^D$ and their corresponding feature representations $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^H$, where $\mathbf{z}_1 = f_{\boldsymbol{\theta}}(\mathbf{x}_1)$ and $\mathbf{z}_2 = f_{\boldsymbol{\theta}}(\mathbf{x}_2)$, the inequality $\|\mathbf{z}_1 - \mathbf{z}_2\|_2 \leq \|\mathbf{x}_1 - \mathbf{x}_2\|_2$ holds. By generalizing this inequality, we obtain :

$$\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{tr}} \|\mathbf{z}^\star - \mathbf{z}'\|_2 \leq \mathbb{E}_{\mathbf{x}' \sim \mathcal{X}_{tr}} \|\mathbf{x}^\star - \mathbf{x}'\|_2.$$

□

Lemma A.4. *(Lemma 5 of Charpentier et al. (2021)) Let $p(\mathbf{z}^\star; \hat{\boldsymbol{\alpha}})$ be parameterized with a Gaussian Mixture Model (GMM). Then $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{tr}} \|\mathbf{z}^\star - \mathbf{z}'\|_2 \to \infty$ implies that $p(\mathbf{z}^\star; \hat{\boldsymbol{\alpha}}) \to 0$.*

Lemma A.5. *(Modified from Proposition 5.5. of Bui & Liu (2023)) $u(\mathbf{z}^\star)$ is monotonically decreasing with respect to $s(\mathbf{x}^\star)$ on the interval $(0, 1]$.*

Proof. Suppose that we employ the predictive entropy (i.e., $u(\mathbf{z}^\star) = \mathcal{H}[p(y|\mathbf{z}^\star)]$) as the uncertainty measure for DAEDL. Then, the uncertainty of $\mathbf{x}^\star$ quantified by DAEDL can be expressed as follows: $u(\mathbf{z}^\star) = \mathcal{H}[\sigma(g_{\hat{\boldsymbol{\phi}}}(\mathbf{z}^\star) \times s(\mathbf{x}^\star)]$. This formulation mirrors the uncertainty presented in the proof of Proposition 5.5 in Bui & Liu (2023), differing only in the method for obtaining the normalized feature space density. Therefore, substituting $s(\mathbf{x}^\star)$ for $p(\mathbf{z}^\star; \boldsymbol{\alpha})$ in the proof of the corresponding theorem allows us to conclude the proof. □

Lemma A.6. *(Proposition 1 of Liu et al. (2020)) Consider a hidden mapping $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{H}$ with residual architecture $f_{\boldsymbol{\theta}} = f_{L-1} \circ \cdots f_2 \circ f_1$, where $f_l(\mathbf{x}) = \mathbf{x} + g_l(\mathbf{x})$. If for $0 < \alpha \leq 1$, all g_l's are α-Lipschitz, i.e., $\|g_l(\mathbf{x}) - g_l(\mathbf{x}')\|_{\mathcal{H}} \leq \alpha \|\mathbf{x} - \mathbf{x}'\|_{\mathcal{X}}$, $\forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X}$. Then,*

$$L_1 \|\mathbf{x} - \mathbf{x}'\|_{\mathcal{X}} \leq \|f_{\boldsymbol{\theta}}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\mathbf{x}')\|_{\mathcal{H}} \leq L_2 \|\mathbf{x} - \mathbf{x}'\|_{\mathcal{X}},$$

where $L_1 = (1 - \alpha)^{L-1}$ and $L_2 = (1 + \alpha)^{L-1}$, i.e., $f_{\boldsymbol{\theta}}$ is distance preserving.

A.2. Proofs of Theorems

We prove Theorem 4.3, Theorem 4.1, Theorem 4.4, and Corollary 4.5 sequentially.

Proof for Theorem 4.3 Let $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\phi}}$ be the optimal parameters of the feature extractor and the classifier, respectively, which are obtained by training. For $\forall c \in [1, 2, \cdots C]$, the predictive distribution of testing example $\mathbf{x}^\star$ obtained by DAEDL is expressed as follows:

$$\begin{aligned} p(y = c|\mathbf{x}^\star; \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) &= \int p(y = c|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\boldsymbol{\alpha}, \mathbf{x}^\star; \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) d\boldsymbol{\pi} \\ &= \int \pi_c \operatorname{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}, \mathbf{x}^\star; \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) d\boldsymbol{\pi} \\ &= \mathbb{E}_{\boldsymbol{\pi} \sim \operatorname{Dir}(\boldsymbol{\alpha})}[\pi_c] \\ &= \frac{\alpha_c}{\sum_{c=1}^{C} \alpha_c}. \end{aligned} \tag{3}$$

Additionally, the concentration parameters of DAEDL can be expressed as follows:

$$\boldsymbol{\alpha}(\mathbf{x}^\star) = \exp(\mathbf{z}(\mathbf{x}^\star)/T(\mathbf{x}^\star)), \tag{4}$$

where $\mathbf{z}(\mathbf{x}^\star) = g_{\hat{\boldsymbol{\phi}}}(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star))$ and $T(\mathbf{x}^\star) = 1/s(\mathbf{x}^\star)$. Plugging Eq.(4) into Eq.(3), the predictive distribution of $\mathbf{x}^\star$ obtained by DAEDL can be expressed as follows:

$$p(y|\mathbf{x}^\star) = \operatorname{Cat}(\bar{\boldsymbol{\pi}}), \quad \bar{\boldsymbol{\pi}} = \sigma\left(\mathbf{z}(\mathbf{x}^\star)/\, T(\mathbf{x}^\star)\right). \tag{5}$$

Here, we omitted the parameters $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\phi}}$ for notational simplicity. From Eq.(5), we can conclude that the predictive distribution of $\mathbf{x}^\star$ obtained by DAEDL aligns with the adaptive temperature scaled softmax model.

Proof for Theorem 4.1 First, $\mathbb{E}_{\mathbf{x}' \sim \mathcal{X}_{tr}} \|\mathbf{x}^\star_{\text{ood}} - \mathbf{x}'\|_2 \to \infty$ implies that $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{tr}} \|\mathbf{z}^\star_{\text{ood}} - \mathbf{z}'\|_2 \to \infty$ (Assumption A.1). Second, $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{tr}} \|\mathbf{z}^\star_{\text{ood}} - \mathbf{z}'\|_2 \to \infty$ implies that $p(\mathbf{z}^\star_{\text{ood}} = f_{\boldsymbol{\theta}}(\mathbf{z}^\star_{\text{ood}})) \to 0$ (Lemma A.4). Therefore, the following inequality holds for $\mathbf{z}^\star_{\text{ood}}$:

$$\log p(\mathbf{z}^\star_{\text{ood}}) \leq \min_{\mathbf{x} \in \mathcal{X}_{tr}} \{\log p(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}))\}.$$

By the definition of the normalizing function s, $p(\mathbf{z}^\star_{\text{ood}} = f_{\boldsymbol{\theta}}(\mathbf{z}^\star_{\text{ood}})) \to 0$ implies that $s(\mathbf{x}^\star_{\text{ood}}) \to 0$ and $T(\mathbf{x}^\star_{\text{ood}}) \to \infty$. Plugging these into Eq.(4) and Eq.(5), the concentration parameters and predictive distribution of $\mathbf{x}^\star_{\text{ood}}$ obtained by DAEDL are derived as follows:

$$\boldsymbol{\alpha}(\mathbf{x}^\star_{\text{ood}}) \to \mathbf{1}, \quad p(y|\mathbf{x}^\star_{\text{ood}}) \to \mathcal{U}\{1, C\}.$$

Proof for Theorem 4.4 The predictive distribution of the testing example $\mathbf{x}^\star$ obtained by DAEDL can be expressed as follows:

$$p(y|\mathbf{z}^\star) = \sigma(g_{\boldsymbol{\phi}}(\mathbf{z}^\star) \times s(\mathbf{x}^\star)),$$

where $\mathbf{z}^\star = f_{\boldsymbol{\theta}}(\mathbf{x}^\star)$ is the feature representation of $\mathbf{x}^\star$. Suppose that we employ the predictive entropy (i.e., $u(\mathbf{z}^\star) = \mathcal{H}[p(y|\mathbf{z}^\star)]$) as the uncertainty measure for DAEDL. Then, the uncertainty of $\mathbf{x}^\star$ quantified by DAEDL can be expressed as follows:

$$u(\mathbf{z}^\star) = \mathcal{H}[(\sigma(g_{\boldsymbol{\phi}}(\mathbf{z}^\star) \times s(\mathbf{x}^\star)].$$

First, $s(\mathbf{x}^\star)$ is monotonically decreasing with respect to $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{\text{tr}}} \|\mathbf{z}^\star - \mathbf{z}'\|_2$ (Assumption A.2). Second, $u(\mathbf{z}^\star)$ is monotonically decreasing with respect to $s(\mathbf{x}^\star)$ (Lemma A.5). Combining these results, it follows that $u(\mathbf{z}^\star)$ is monotonically increasing with respect to $\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{\text{tr}}} \|\mathbf{z}^\star - \mathbf{z}'\|_2$. In other words,

$$u(\mathbf{z}^\star) = \nu\big(\mathbb{E}_{\mathbf{z}' \sim \mathcal{Z}_{\text{tr}}} \|\mathbf{z}^\star - \mathbf{z}'\|_2\big)$$

holds for a monotonic function ν. Therefore, the predictive distribution of $\mathbf{x}^\star$ obtained by DAEDL is *distance aware* in the feature space.

Proof for Corollary 4.5 First, if spectral normalization is applied to $f_{\boldsymbol{\theta}}$, and $f_{\boldsymbol{\theta}}$ is constructed by the residual blocks, $f_{\boldsymbol{\theta}}$ is *distance preserving* (Lemma A.6). Second, the uncertainty of $\mathbf{x}^\star$ obtained by DAEDL is monotonically increasing with respect to the feature space distance (Theorem 4.4). In other words, the output layer is *distance aware*. Consequently, the combination of *distance preserving* feature extractor and *distance aware* output layers leads to the *distance awareness* in the input space (Section 2.2 of Liu et al. (2020)). Therefore, we can conclude that if $f_{\boldsymbol{\theta}}$ is constructed by the residual blocks, the predictive distribution of $\mathbf{x}^\star$ obtained by DAEDL is *distance aware* in the input space.

A.3. Additional Explanation about Theorem 4.2

We provide a detailed explanation about Theorem 4.2. First, we interpret the DBU models in the Bayesian context within the framework of an *input-dependent Dirichlet-Categorical model* (Charpentier et al., 2020). Second, we analyze the limitations of the conventional prior specifications utilized in DBU models. Finally, we interpret DAEDL under the same framework to underscore its strength. In particular, DAEDL can be interpreted as predicting a posterior distribution of input-dependent Dirichlet Categorical model using an improper prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{0})$. This approach mitigates the challenge of prior specification that occurs in common DBU models, allowing our model to learn the appropriate posterior Dirichlet distribution from the data.

DBU model under the input-dependent Dirichlet Categorical model framework. In a Bayesian context, DBU models can be interpreted under the framework of an *input-dependent Dirichlet-Categorical model* (Charpentier et al., 2020). Intuitively, the goal of the DBU model is to predict the posterior Dirichlet distribution for the data using a neural network. The concentration parameters of the prior Dirichlet distribution for these models are determined based on the prior belief about the class counts. When there is no prior information, the parameters of the prior are conventionally set as $\boldsymbol{\alpha}_{\mathrm{prior}}(\mathbf{x}_i) = \mathbf{1}$. To estimate the concentration parameters of the posterior Dirichlet distribution, we need to obtain the class counts from the observations. However, in the absence of such observations, DBU models predict *pseudo-observations* $\{\tilde{y}_i^{(j)}\}_{j=1}^N$ for each data point $\mathbf{x}_i$, utilizing a neural network. More specifically, DBU models predict *pseudo-counts* (i.e., class counts of the pseudo-observations) $\alpha_{\mathrm{data}}^{(c)} = \sum_{j=1}^N \mathbb{1}_{\{\tilde{y}_j^{(j)}=c\}}$ for each class $\forall c \in [C]$. Then, the concentration parameters of the posterior Dirichlet distribution can be obtained in a closed form, leveraging the conjugacy of the Dirichlet and Categorical distributions. In summary, the DBU model can be expressed as a Bayesian model as follows:

$$
\begin{aligned}
\textbf{Prior} \quad & \boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha}_{\mathrm{prior}}(\mathbf{x}_i)), \\
\textbf{Likelihood} \quad & \{\tilde{y}_i^{(j)}\}_{j=1}^N | \boldsymbol{\pi} \sim \mathrm{Cat}(\boldsymbol{\pi}). \\
\textbf{Posterior} \quad & \boldsymbol{\pi} | \{\tilde{y}_i^{(j)}\}_{j=1}^N \sim \mathrm{Dir}(\boldsymbol{\alpha}_{\mathrm{post}}(\mathbf{x}_i)), \quad \boldsymbol{\alpha}_{\mathrm{post}}(\mathbf{x}_i) = \boldsymbol{\alpha}_{\mathrm{prior}}(\mathbf{x}_i) + \boldsymbol{\alpha}_{\mathrm{data}}(\mathbf{x}_i),
\end{aligned}
$$

where $\boldsymbol{\pi}$ is a class probability and the pseudo-counts are computed as $\alpha_{\mathrm{data}}^{(c)}(\mathbf{x}_i) = \sum_{j=1}^N \mathbb{1}_{\{\tilde{y}_j^{(j)}=c\}}, \forall c \in [C]$. The core aspect of the DBU model involves predicting pseudo-counts using a neural network and computing the posterior Dirichlet distribution. Specifically, the pseudo-counts are obtained as follows:

$$
\boldsymbol{\alpha}_{\mathrm{data}}(\mathbf{x}_i) = h(g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x}_i))),
$$

where $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^H$, $g_{\boldsymbol{\phi}} : \mathbb{R}^H \to \mathbb{R}^C$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and h are the feature extractor, classifier, parameters of the feature extractor, parameters of the classifier, and activation function, respectively. With the conventional choice for the concentration parameters of the prior Dirichlet distribution ($\boldsymbol{\alpha}_{\mathrm{prior}}(\mathbf{x}_i) = \mathbf{1}$), the concentration parameters of the posterior Dirichlet distribution of the DBU models can be expressed as follows:

$$
\boldsymbol{\alpha}_{\mathrm{post}}(\mathbf{x}_i) = \mathbf{1} + h(g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x}_i))).
$$

Representative DBU models, including PostNet (Charpentier et al., 2020), NatPN (Charpentier et al., 2021), EDL (Sensoy et al., 2018), and $\mathcal{I}$-EDL (Deng et al., 2023), can be interpreted within the framework explained above. The comparison of the DBU models under the input-dependent Dirichlet-Categorical model framework is detailed in Table 6.

Limitation of conventional parameterization of DBU models. For a typical Dirichlet-Categorical model in Bayesian statistics, a commonly employed non-informative prior is a uniform Dirichlet distribution (i.e., $\mathrm{Dir}(\mathbf{1})$). Thus, it is natural for models like PostNet and NatPN to adopt this prior. Moreover, EDL models (Sensoy et al., 2018; Deng et al., 2023) can

Table 6. The comparison between typical DBU models (Sensoy et al., 2018; Charpentier et al., 2020; 2021; Deng et al., 2023) and DAEDL in the Bayesian context. N_c denotes the number of observations for each class (Charpentier et al., 2020), and N_H is a hyperparameter that corresponds to the certainty budget (Charpentier et al., 2021). $\boldsymbol{\beta}$ denotes the parameter of the density estimator.

	PRIOR	PSEUDO-COUNTS	POSTERIOR
POSTNET	$\alpha^{(c)}_{\text{prior}} = 1$	$\alpha^{(c)}_{\text{data}} = N_c \times p(\mathbf{z}^\star; \boldsymbol{\beta}^{(c)})$	$\alpha^{(c)}_{\text{post}} = 1 + N_c \times p(\mathbf{z}^\star; \boldsymbol{\beta}^{(c)})$
NATPN	$\alpha^{(c)}_{\text{prior}} = 1$	$\alpha^{(c)}_{\text{data}} = N_H \times p(\mathbf{z}^\star; \boldsymbol{\beta}) \times (g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c$	$\alpha^{(c)}_{\text{post}} = 1 + N_H \times p(\mathbf{z}^\star; \boldsymbol{\beta}) \times (g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c$
EDL	$\alpha^{(c)}_{\text{prior}} = 1$	$\alpha^{(c)}_{\text{data}} = \text{ReLU}((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c)$	$\alpha^{(c)}_{\text{post}} = 1 + \text{ReLU}((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c)$
$\mathcal{I}$-EDL	$\alpha^{(c)}_{\text{prior}} = 1$	$\alpha^{(c)}_{\text{data}} = \text{Softplus}((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c)$	$\alpha^{(c)}_{\text{post}} = 1 + \text{Softplus}((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c)$
DAEDL (TRAINING)	$\alpha^{(c)}_{\text{prior}} = 0$	$\alpha^{(c)}_{\text{data}} = \exp((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c)$	$\alpha^{(c)}_{\text{post}} = \exp((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c)$
DAEDL (PREDICTION)	$\alpha^{(c)}_{\text{prior}} = 0$	$\alpha^{(c)}_{\text{data}} = \exp((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c \times s(\mathbf{z}^\star))$	$\alpha^{(c)}_{\text{post}} = \exp((g_{\boldsymbol{\phi}}(\mathbf{z}^\star))_c \times s(\mathbf{z}^\star; \boldsymbol{\beta}))$

also be interpreted as using a uniform Dirichlet prior implicitly. Theorem A.7 establishes that typical DBU models can be interpreted as using $\boldsymbol{\pi} \sim \text{Dir}(\mathbf{1})$ under this framework. However, this seemingly reasonable choice of the concentration parameters of the prior Dirichlet distribution in the DBU models poses inherent challenges. Notably, the mechanism for obtaining the posterior distribution differs from typical Bayesian models. Obtaining the parameters of the posterior in the DBU models involves pseudo-counts, which are estimated by a neural network, rather than actual class counts. From Table 6, we can see that the range of the pseudo-count is $[0, \infty)$. Consequently, several failure cases arise due to the unconstrained range of the pseudo-counts. Below, we illustrate some representative failure cases.

Theorem A.7. ***(Bayesian Interpretation of DBU models)*** *In the Bayesian context, typical DBU models can be interpreted as predicting an input-dependent posterior distribution of the Dirichlet-Categorical model with a uniform prior* $\boldsymbol{\pi} \sim \text{Dir}(\mathbf{1})$.

(i) $C = 3, \quad \boldsymbol{\alpha}_{\text{data}}(\mathbf{x}^\star_1) = [0.1, 0.01, 0.01]$

Consider the scenario where the magnitude of the concentration parameters of the prior Dirichlet distribution dominates the pseudo-counts in the classification task with $C = 3$. Suppose that $\mathbf{x}^\star_1$ is a highly likely ID testing example that corresponds to the first class, with the pseudo-counts computed as $\boldsymbol{\alpha}_{\text{data}}(\mathbf{x}^\star_1) = [0.1, 0.01, 0.01]$. Then, the concentration parameters of the posterior are estimated as $\boldsymbol{\alpha}_{\text{post}}(\mathbf{x}^\star_1) = [1.1, 1.01, 1.01]$. However, the maximum expected class probability is computed as $1.1/3.12$, yielding a value close to $1/3$. Then, the highly likely ID data point will be misclassified as OOD.

ii) $C = 10, \quad \boldsymbol{\alpha}_{\text{data}}(\mathbf{x}^\star_2) = [10, 0, 0, \cdots, 0]$

Consider the scenario where pseudo-counts exist only for one class in the classification task with $C = 10$. Suppose that $\mathbf{x}^\star_2$ is a highly likely ID testing example that corresponds to the first class, with the pseudo-counts computed as $\boldsymbol{\alpha}_{\text{data}}(\mathbf{x}^\star_2) = [10, 0, 0, \cdots, 0]$. Then, the concentration parameters of the posterior is estimated as $\boldsymbol{\alpha}_{\text{post}}(\mathbf{x}_2) = [11, 1, 1, \cdots, 1]$. Subsequently, the maximum expected class probability is computed as $11/20 = 0.55$. This result is counter-intuitive, as typical classification models (e.g., softmax) usually output the maximum class probability close to 1 for a highly ID data point like $\mathbf{x}^\star_2$.

These counter-intuitive results in the expected class probability occur due to the inherent challenge of achieving the appropriate balance in the magnitude between the concentration parameters of the prior Dirichlet distribution and the pseudo-counts. To overcome this problem, it is necessary to employ the optimal concentration parameters of the prior Dirichlet distribution that maintains the right balance with the pseudo-counts. In the absence of specific information, it is reasonable to set the concentration parameter of the prior Dirichlet distribution for each class to be the same (i.e., $\boldsymbol{\alpha}_{\text{prior}} = \alpha\mathbf{1}$), where α represents the magnitude and $\mathbf{1} \in \mathbb{R}^C$. However, determining the optimal magnitude α is non-trivial, because of the unconstrained range of the magnitude of the pseudo-counts. While an optimal value of α in the DBU models might exist in some contexts, $\alpha = 1$ is applied in most tasks. However, as illustrated above, numerous counter-intuitive results may occur in the expected class probability attributed to setting $\alpha = 1$. We hypothesize that prior misspecification in DBU models can be a significant factor that contributes to the limited classification performance of DBU models. DAEDL successfully addresses this limitation by employing an alternative prior that eliminates the need for this nonessential balancing.

Bayesian interpretation of DAEDL. We interpret DAEDL within the input-dependent Dirichlet-Categorical model framework to provide insights into how DAEDL resolves the limitation in the conventional parameterization of DBU models. As established in Theorem 4.2, DAEDL can be interpreted as predicting the input-dependent posterior distribution of the Dirichlet-Categorical model with an improper prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{0})$. DAEDL essentially eliminates the challenge of determining the optimal parameter value of the prior by setting it to zero. Although the improper prior is not a valid probability distribution, the posterior Dirichlet distribution is valid if all pseudo-counts are positive, which holds in practice for neural network models. The improper prior exhibits favorable theoretical properties, such as being a uniform prior in the log-scale of the parameter (i.e., $\log \boldsymbol{\pi}$) (Gelman et al., 1995).

To summarize, while typical DBU models can be interpreted as using a uniform prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{1})$ in the Bayesian context, DAEDL can be viewed as employing an improper prior $\boldsymbol{\pi} \sim \mathrm{Dir}(\mathbf{0})$. While opting for a uniform prior is standard in typical models, it poses challenges within the context of DBU models, due to the challenge of achieving an appropriate balance between the parameter of the prior and pseudo-counts. This challenge may lead to counter-intuitive results in the expected class probabilities, which could be a potential reason for the limited performance of DBU models in classification tasks. Therefore, within the DBU model framework, we assert that employing an improper prior $\mathrm{Dir}(\mathbf{0})$ is a more practical choice.

A.4. Additional Explanation about Theorem 4.3

We provide an additional insight into Theorem 4.3. To begin, we define the EDL equipped with the proposed parameterization as EDL$^+$ for notational convenience. Despite the existence of other models related to adaptive temperature scaling, here we focus on Adaptive Temperature Scaling (AdaTS) (Joy et al., 2023), which serves as a representative model.

We first clarify the relationship between the four models: i) Softmax, ii) EDL$^+$, iii) AdaTS, and iv) DAEDL. Figure 3 illustrates the relationship of these models graphically. The orange arrow signifies the enhancement in the model calibration performance achieved by adaptive temperature scaling. The blue arrow denotes the improvement in the uncertainty estimation performance of the model achieved by adopting the EDL structure. As demonstrated in Figure 3, DAEDL can be interpreted in two manners: i) calibrated version of EDL$^+$, and ii) AdaTS with uncertainty estimation ability. First, DAEDL can be interpreted as the calibrated version of EDL$^+$, accomplished by dividing the logits by the sample-dependent temperature before applying the exponential activation function. Given the demonstrated effectiveness of *adaptive temperature scaling* in improving model calibration and OOD detection (Balanya et al., 2022; Joy et al., 2023; Krumpl et al., 2024), we can expect that DAEDL will exhibit similar benefits and improvements over EDL$^+$. Second, DAEDL can be interpreted as AdaTS with uncertainty estimation capability. Given the superior uncertainty estimation performance of the EDL$^+$ compared to the softmax model, DAEDL is expected to exhibit improved uncertainty estimation performance over AdaTS. In summary, DAEDL can be interpreted as a model created by integrating enhancements from two distinct directions to the softmax model, to improve both calibration and uncertainty estimation performance.

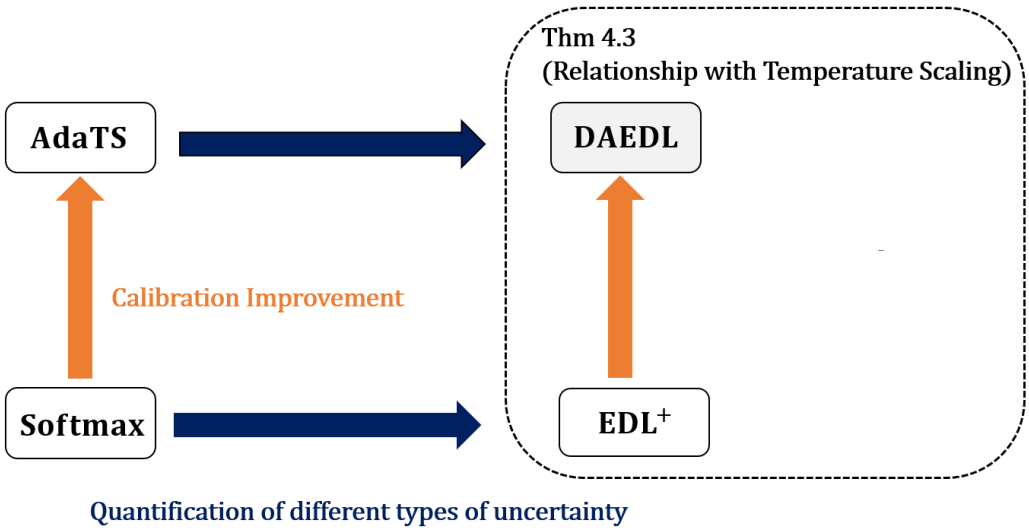


Figure 3. Relationship between Softmax, EDL$^+$, AdaTS, and DAEDL

B. Algorithm

We present the algorithms for DAEDL. Specifically, the algorithms for training, density estimation, and prediction are provided in Algorithm 1, Algorithm 2, and Algorithm 3, respectively. These algorithms are intuitive and easy to implement.

Algorithm 1 DAEDL Training

Input: Training data $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, initial model parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$, maximum epoch M, learning rate η, batch size B, regularization parameter λ
for $i = 1$ **to** M **do**
  SGD update $\left\{\boldsymbol{\theta} = \{W^{(l)}, b^{(l)}\}_{l=1}^L,\ \boldsymbol{\phi}\right\}$.
  Apply spectral normalization to $\{W^{(l)}\}_{l=1}^L$.
end for
Output: Trained model parameter $\{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}\}$

Algorithm 2 DAEDL Density Estimation

Input: Training data $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, trained feature extractor parameter $\hat{\boldsymbol{\theta}}$
for $c = 1$ **to** C **do**
  $\hat{\omega}_c \leftarrow \frac{N_c}{N},\ \ N_c = \sum_{i=1}^N \mathbb{1}_{\{y_i=c\}}$
  $\hat{\boldsymbol{\mu}}_c \leftarrow \frac{1}{N_c} \sum_{\{i:y_i=c\}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i)$
  $\hat{\Sigma}_c \leftarrow \frac{1}{N_c - 1} \sum_{\{i:y_i=c\}} (f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_c)(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_c)^T$
end for
Output: GDA parameter $\hat{\boldsymbol{\omega}} = \{\hat{\omega}_c\}_{c=1}^C, \hat{\boldsymbol{\mu}} = \{\hat{\boldsymbol{\mu}}_c\}_{c=1}^C, \hat{\boldsymbol{\Sigma}} = \{\hat{\Sigma}_c\}_{c=1}^C$

Algorithm 3 DAEDL Prediction

Input: Testing example $\mathbf{x}^\star$, trained model parameter $\{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}\}$, GDA parameter $\hat{\boldsymbol{\omega}},\ \hat{\boldsymbol{\mu}},\ \hat{\boldsymbol{\Sigma}}$
Estimate the log feature space density: $\log p(\mathbf{z}^\star = f_{\boldsymbol{\theta}}(\mathbf{x}^\star) | \hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\mu}}, \hat{\Sigma})$
Compute the normalized feature space density: $s(\mathbf{x}^\star) = \text{Clip}\left(\frac{\log p(\mathbf{z}^\star = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star)) - d_{\min}}{d_{\max} - d_{\min}}\right)$
Compute the concentration parameters: $\boldsymbol{\alpha}_{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}}(\mathbf{x}^\star) = \exp\left(g_{\hat{\boldsymbol{\phi}}}(f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}^\star)) \times s(\mathbf{x}^\star)\right)$
Output: Concentration parameters $\boldsymbol{\alpha}_{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}}(\mathbf{x}^\star)$

C. Experimental Details

In this section, we provide the details about our experiments. First, we provide a detailed description of the datasets employed in our experiments. Second, we provide the implementation details.

C.1. Datasets

Following Charpentier et al. (2020) and Deng et al. (2023), we mainly use two image classification datasets : (i) MNIST (LeCun, 1998) and (ii) CIFAR-10 (Krizhevsky et al., 2009) as our ID dataset. For MNIST, Kuzushiji-MNIST (KMNIST) (Clanuwat et al., 2018) and FashionMNIST (FMNIST) (Xiao et al., 2017) serve as OOD datasets. Regarding CIFAR-10, Street View House Numbers (SVHN) (Netzer et al., 2011) and CIFAR-100 (Krizhevsky et al., 2009) were employed as OOD datasets. Furthermore, to assess our model's capability for detecting distribution shifts, we utilize MNIST-C (Mu & Gilmer, 2019a) and CIFAR-10-C (Hendrycks & Dietterich, 2019), which are created by introducing distribution shifts (corruption) to MNIST and CIFAR-10, respectively. A detailed description of each dataset is provided below.

MNIST (LeCun, 1998) is a dataset comprising grayscale images of handwritten digits (0 through 9). MNIST is widely used as a benchmark dataset for evaluating machine learning algorithms. It comprises 60,000 training images and 10,000

testing images. Each image is represented as a $1 \times 28 \times 28$ tensor. We partitioned the training samples into a training set and a validation set with a ratio of $0.8 : 0.2$.

Kuzushiji-MNIST (KMNIST) (Clanuwat et al., 2018) serves as a more challenging alternative of the MNIST dataset. Similar to MNIST, it comprises grayscale images represented by $1 \times 28 \times 28$ tensor and includes 60,000 training images and 10,000 testing images. However, KMNIST features a handwritten Japanese Hiragana script, originally designed for researching the recognition of historical Japanese characters. In our experiments, we employed KMNIST as an OOD dataset for MNIST.

FashionMNIST (FMNIST) (Xiao et al., 2017) serves as another alternative to the MNIST dataset. Similar to MNIST and KMNIST, it comprises grayscale images represented by $1 \times 28 \times 28$ tensor and includes 60,000 training images and 10,000 testing images. However, the FMNIST dataset features images of various fashion items. Specifically, it consists of 10 classes, each representing specific fashion items such as T-shirt, trouser, pullover, and others. In our experiments, we employed FMNIST as an OOD dataset for MNIST.

CIFAR-10 (Krizhevsky et al., 2009) is a dataset comprising color images of different animals and objects. CIFAR-10 is widely used as a benchmark dataset for evaluating machine learning algorithms. It comprises of 50,000 training images and 10,000 testing images. The dataset is divided into ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each image in the dataset is represented as $3 \times 32 \times 32$ tensor. We partitioned the training samples into a training set and a validation set with a ratio of $0.95 : 0.05$.

Street View House Numbers (SVHN) (Netzer et al., 2011) is a dataset consisting of cropped images of house numbers from Google Street View. The dataset is divided into 10 classes, 1 for each digit. There are 73,257 training images, 26,032 testing images, and 531,131 additional images. In our experiments, we employed SVHN as an OOD dataset for CIFAR-10.

CIFAR-100 (Krizhevsky et al., 2009) is an extension of the CIFAR-10 dataset. It comprises 50,000 training images and 10,000 testing images, divided among 100 different classes. Each image in this dataset is represented as a $3 \times 32 \times 32$ tensor. In our experiments, we employed CIFAR-100 as an OOD dataset for CIFAR-10.

MNIST-C (Mu & Gilmer, 2019a) is a dataset created by applying 15 different types of corruptions (distribution shift) to the MNIST test set. These corruptions include shot noise, impulse noise, glass blur, motion blur, shear, scale, rotate, brightness, translate, stripe, fog, spatter, dotted line, zigzag, and canny edges. In our experiments, we utilized a static MNIST-C dataset, where severity levels are fixed (Mu & Gilmer, 2019b), as an OOD dataset for the distribution shift detection task.

CIFAR-10-C (Hendrycks & Dietterich, 2019) is a dataset generated by applying 19 different types of corruption (distribution shift) to the CIFAR-10 test set. These corruptions include Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelate, jpeg compression, speckle noise, Gaussian blur, splatter, and saturate. Each corruption is applied at five different severity levels $\mathcal{C} \in \{1, 2, 3, 4, 5\}$. In our experiments, we employed CIFAR-10-C as an OOD dataset for the distribution shift detection task. In essence, conducting distribution shift detection for CIFAR-10-C is akin to performing OOD detection with 95 (19×5) different OOD datasets.

C.2. Implementation Details

For a fair comparison, we followed Charpentier et al. (2020) and Deng et al. (2023) regarding the choice of the backbone. Specifically, we implemented a configuration of 3 convolutional layers and 3 dense layers (ConvNet) when utilizing MNIST as the ID dataset. When CIFAR-10 served as the ID dataset, we used VGG-16. For VGG-16, dropout with a rate of 0.5 was applied. FMNIST and KMNIST were used as OOD datasets for MNIST, while SVHN and CIFAR-100 were employed as OOD datasets for CIFAR-10. To prevent overfitting, early stopping based on the validation loss was implemented for both datasets. In the case of MNIST, training extended up to 50 epochs with a batch size of 64, and for CIFAR-10, we trained up to 100 epochs with the same batch size. The Adam optimizer and LambdaLR scheduler were employed for both datasets. The learning rate (η) and regularization parameter (λ) were determined through a grid search, yielding the optimal values of

$(\eta, \lambda) = (10^{-3}, 5 \times 10^{-2})$. Notably, DAEDL was robust to the hyperparameter choice and only required minimal tuning. A summary of the implementation details is presented in Table 7.

Table 7. Implementation details of our experiments. B, p_{drop}, η, lr_λ, λ, and $T_{\max}$ denote the batch size, dropout rate, learning rate, parameter of the scheduler, regularization parameter, and maximum epoch, respectively.

ID DATASET	BACKBONE	OPTIMIZER	SCHEDULER	B	p_{drop}	η	lr_λ	λ	$T_{\max}$
MNIST	CONVNET	ADAM	LAMBDALR	64	-	10^{-3}	0.95^{epochs}	5×10^{-2}	50
CIFAR-10	VGG-16	ADAM	LAMBDALR	64	0.5	10^{-3}	0.95^{epochs}	5×10^{-2}	100

D. Additional Experimental Results

D.1. Additional Results in OOD Detection (Section 6.2)

Table 8 presents the area under the receiver operating characteristic curve (AUROC) scores measured for the OOD detection performance. We can observe that DAEDL outperformed the competitors in all tasks.

Table 8. AUROC scores of OOD detection based on aleatoric and epistemic uncertainty. The results of EDL and $\mathcal{I}$-EDL were obtained from Deng et al. (2023).

	MNIST → KMNIST		**MNIST → FMNIST**		**CIFAR-10 → SVHN**		**CIFAR-10 → CIFAR-100**	
	ALEA.	**EPIS.**	**ALEA.**	**EPIS.**	**ALEA.**	**EPIS.**	**ALEA.**	**EPIS.**
EDL	96.59 ± 0.6	96.18 ± 1.3	96.49 ± 0.8	96.22 ± 1.3	80.64 ± 4.2	81.06 ± 4.5	80.96 ± 0.8	80.63 ± 1.0
$\mathcal{I}$-EDL	98.00 ± 0.3	97.97 ± 0.3	97.99 ± 0.3	97.97 ± 0.3	87.58 ± 2.0	86.79 ± 1.3	83.55 ± 0.7	82.15 ± 0.5
DAEDL	$\mathbf{99.88 \pm 0.0}$	$\mathbf{99.90 \pm 0.0}$	$\mathbf{99.77 \pm 0.1}$	$\mathbf{99.82 \pm 0.0}$	$\mathbf{89.10 \pm 1.0}$	$\mathbf{89.24 \pm 1.0}$	$\mathbf{85.94 \pm 0.1}$	$\mathbf{86.04 \pm 0.1}$

D.2. Additional Results in Distribution Shift Detection (Section 6.4)

MNIST → MNIST-C Table 9 demonstrates the results for the distribution shift detection task on the MNIST-C dataset. The results with two different score metrics (AUPR and AUROC) and uncertainty measures (aleatoric and epistemic) are provided. We can observe that DAEDL outperforms the competitors by a significant margin regardless of the score metric and uncertainty measures.

Table 9. AUPR and AUROC scores of distribution shift detection based on aleatoric and epistemic uncertainty estimates for the MNIST-C dataset. Alea.AUPR indicates that aleatoric uncertainty is employed as an uncertainty measure while AUPR is utilized as a performance metric. Similarly, Alea.AUROC denotes the results with aleatoric uncertainty and AUROC. In addition, EPIS.AUPR and EPIS.AUROC denote the results obtained by employing epistemic uncertainty while utilizing AUPR and AUROC as a performance metric, respectively.

	ALEA. AUPR	**EPIS. AUPR**	**ALEA. AUROC**	**EPIS. AUROC**
MSP	78.54 ± 0.3	-	79.99 ± 0.4	-
EDL	82.75 ± 0.8	82.75 ± 0.8	82.59 ± 0.6	82.59 ± 0.6
$\mathcal{I}$-EDL	86.06 ± 0.5	86.04 ± 0.5	85.83 ± 0.7	85.80 ± 0.7
DAEDL	$\mathbf{92.43 \pm 0.3}$	$\mathbf{92.51 \pm 0.3}$	$\mathbf{91.84 \pm 0.3}$	$\mathbf{91.99 \pm 0.3}$

CIFAR-10 → CIFAR-10-C Figure 4 demonstrates four sub-figures that represent the results of the distribution shift detection task. The four sub-figures differ by the uncertainty measure (aleatoric and epistemic) and the performance metric (AUPR and AUROC). The results indicate that DAEDL consistently outperforms the competitors regardless of the score and uncertainty metrics. Notably, the performance gap between DAEDL and the competitors is higher when aleatoric uncertainty was applied as an uncertainty measure. As aleatoric uncertainty corresponds to the inherent uncertainty within the data, it is a more suitable measure for capturing the distribution shift that occurred in the data. Therefore, the results align with our expectations.

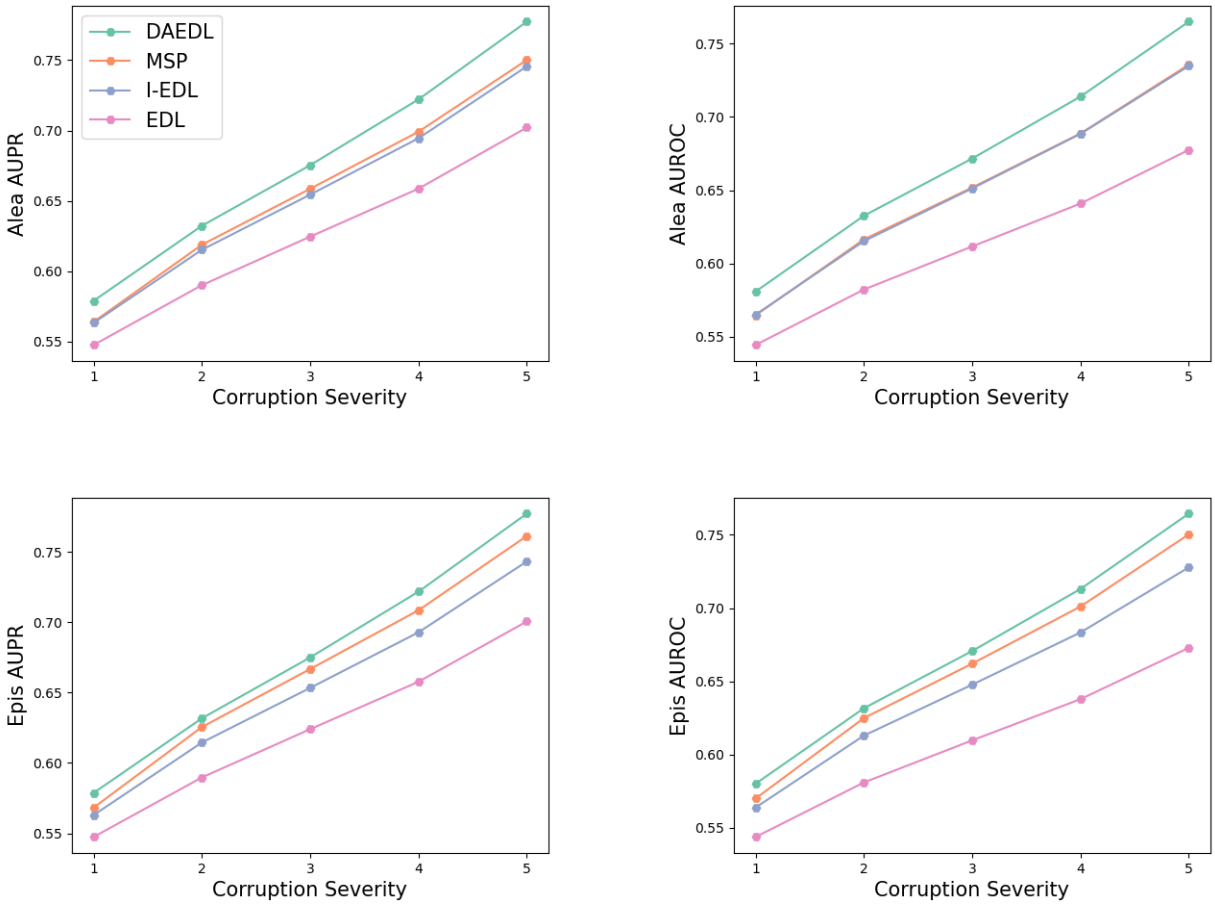


Figure 4. The plot shows the average scores for distribution shift detection in the CIFAR-10-C dataset. The scores were obtained by averaging over 5 independent runs. The score for each run was obtained by averaging over 19 different corruptions. The left-side figures show the results using AUPR, while the right-side figures present the outcomes with AUROC. Additionally, the top row displays the results obtained using aleatoric uncertainty, and the bottom row features the results obtained using epistemic uncertainty.

Results for various corruption types on CIFAR-10 $\rightarrow$ CIFAR-10-C Figure 5, Figure 6, Figure 7, and Figure 8 demonstrate the graphical illustrations of the results for 19 different corruptions. The four figures differ by the uncertainty measure (aleatoric and epistemic) and the performance metric (AUPR and AUROC). In each figure, 19 sub-figures differ by the type of corruption applied. The list of corruptions includes Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelate, jpeg compression, speckle noise, Gaussian blur, splatter, and saturate. From the figures, we can observe that DAEDL outperformed the competitors in most of the corruption types and severity levels.

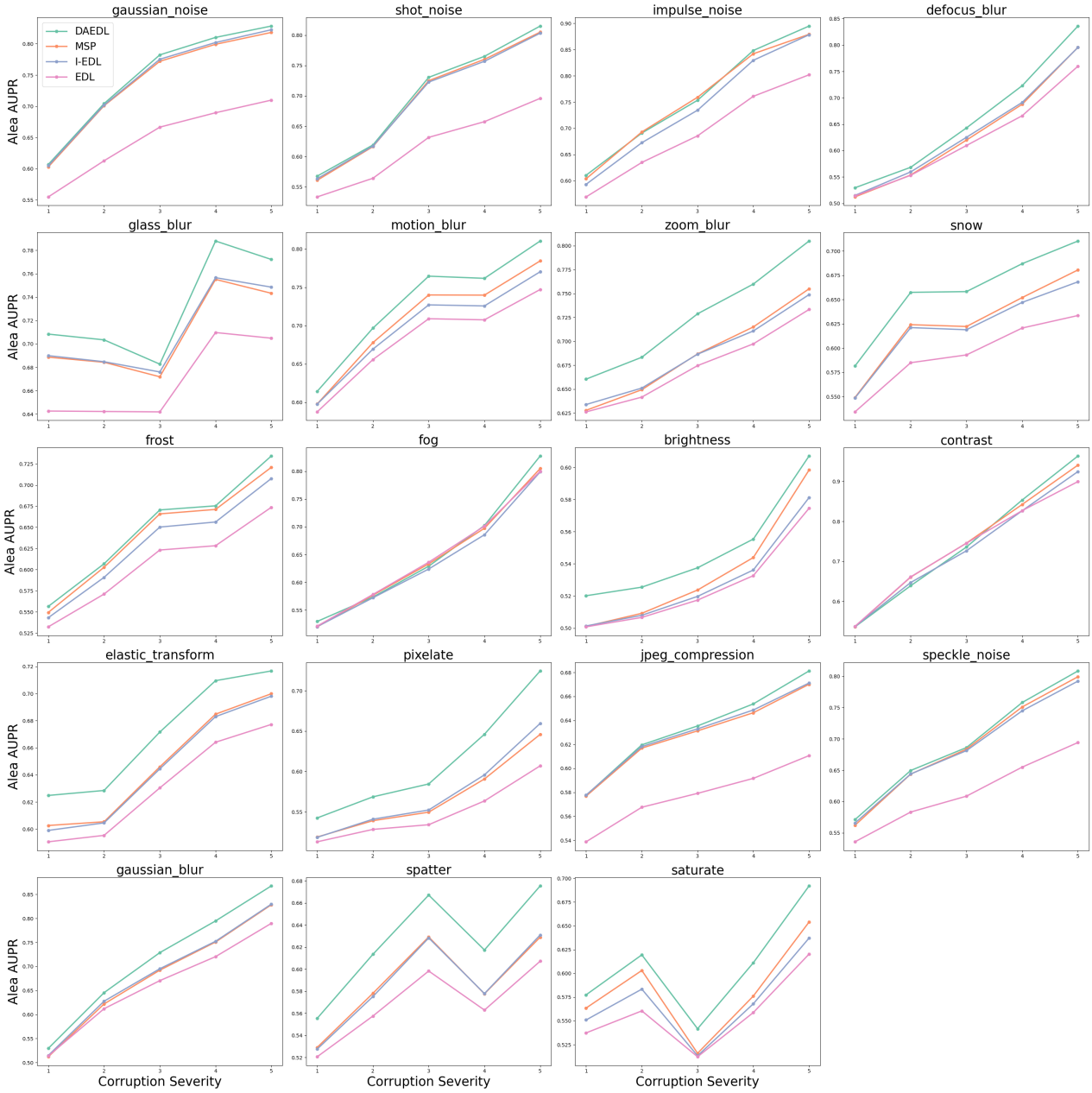


Figure 5. AUPR scores for distribution shift detection using aleatoric uncertainty estimates across 19 different corruptions in the CIFAR-10-C dataset

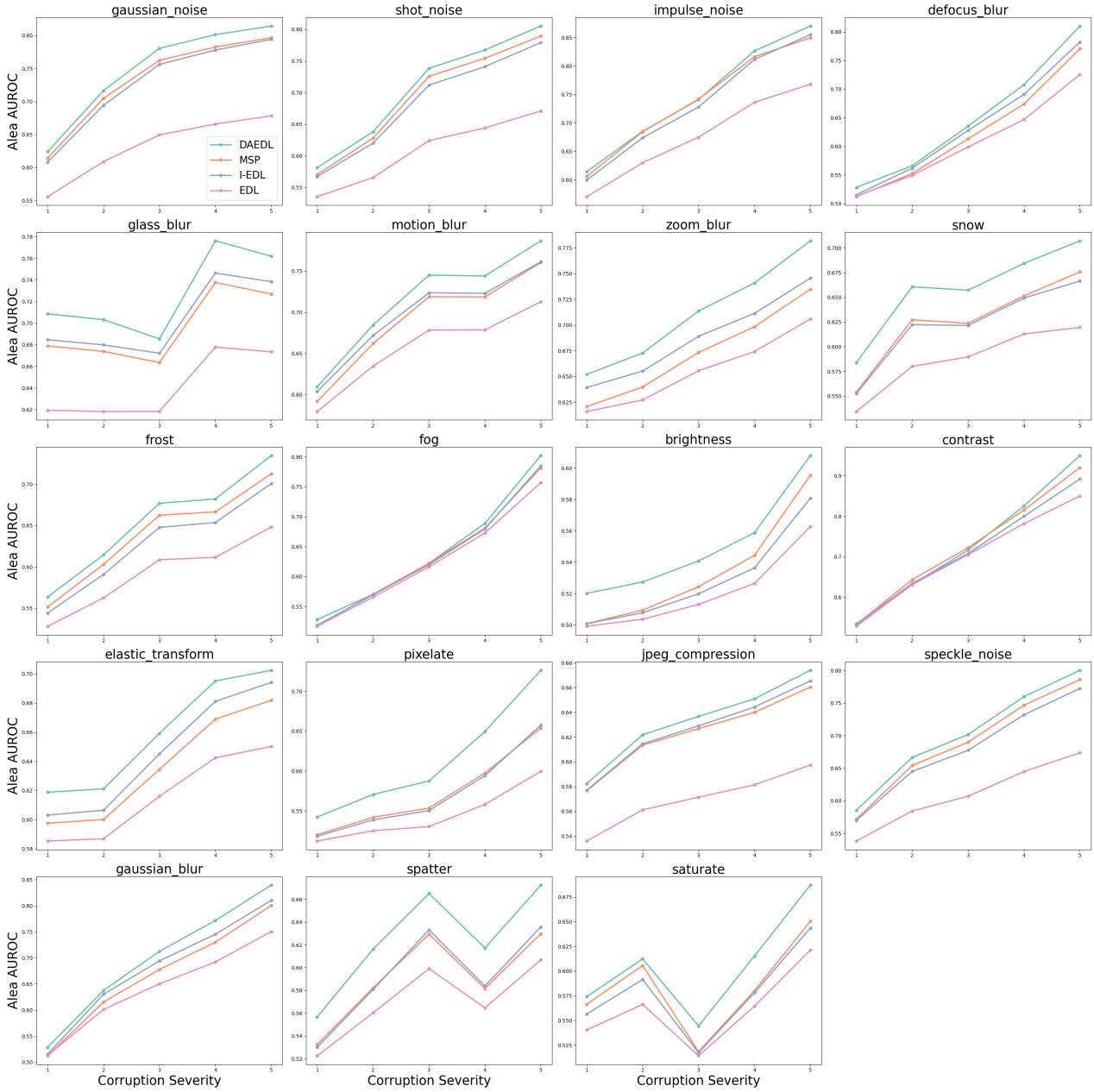


Figure 6. AUROC scores for distribution shift detection using aleatoric uncertainty estimates across 19 different corruptions in the CIFAR-10-C dataset

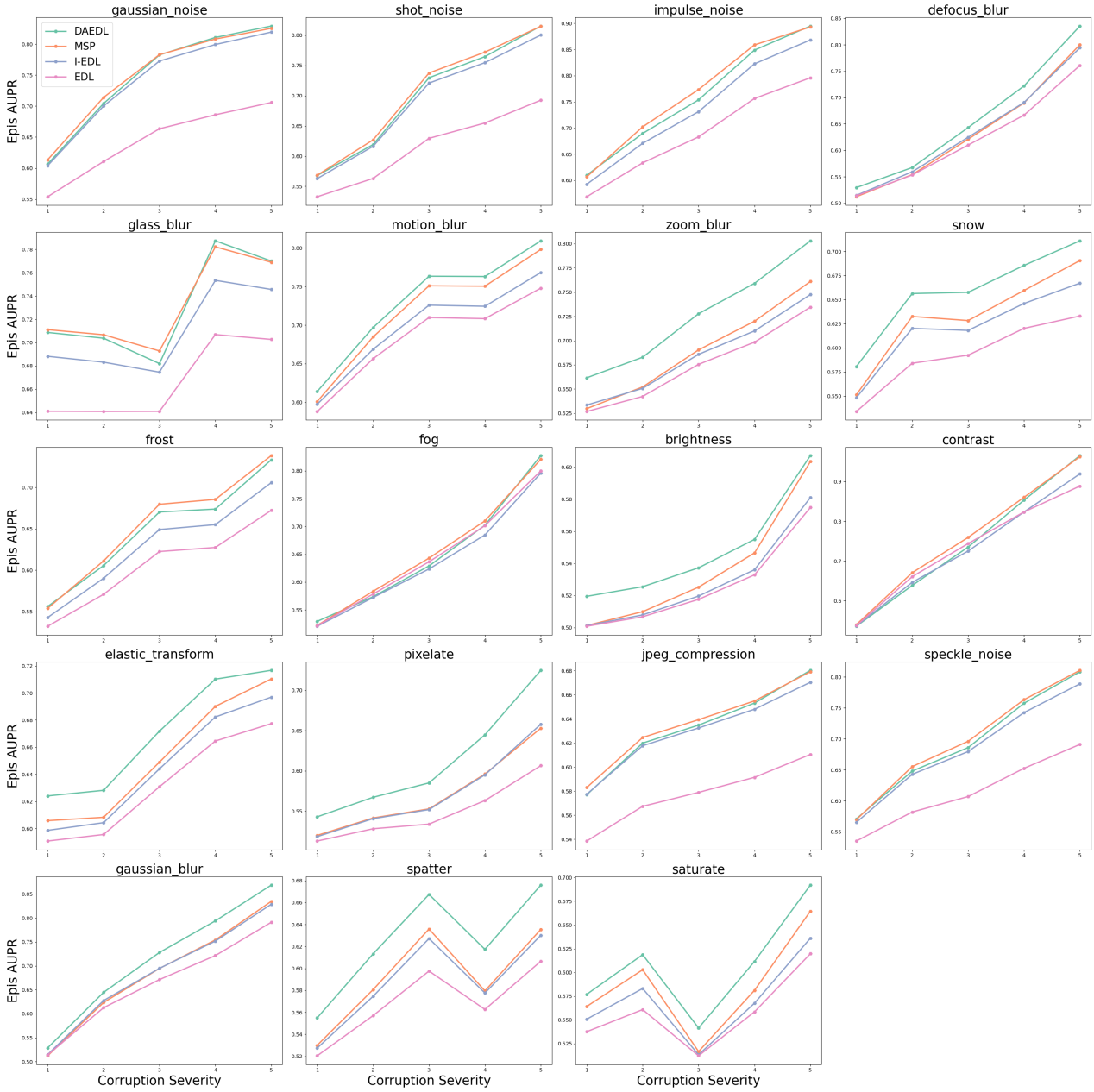


Figure 7. AUPR scores for distribution shift detection using epistemic uncertainty estimates across 19 different corruptions in CIFAR-10-C dataset.

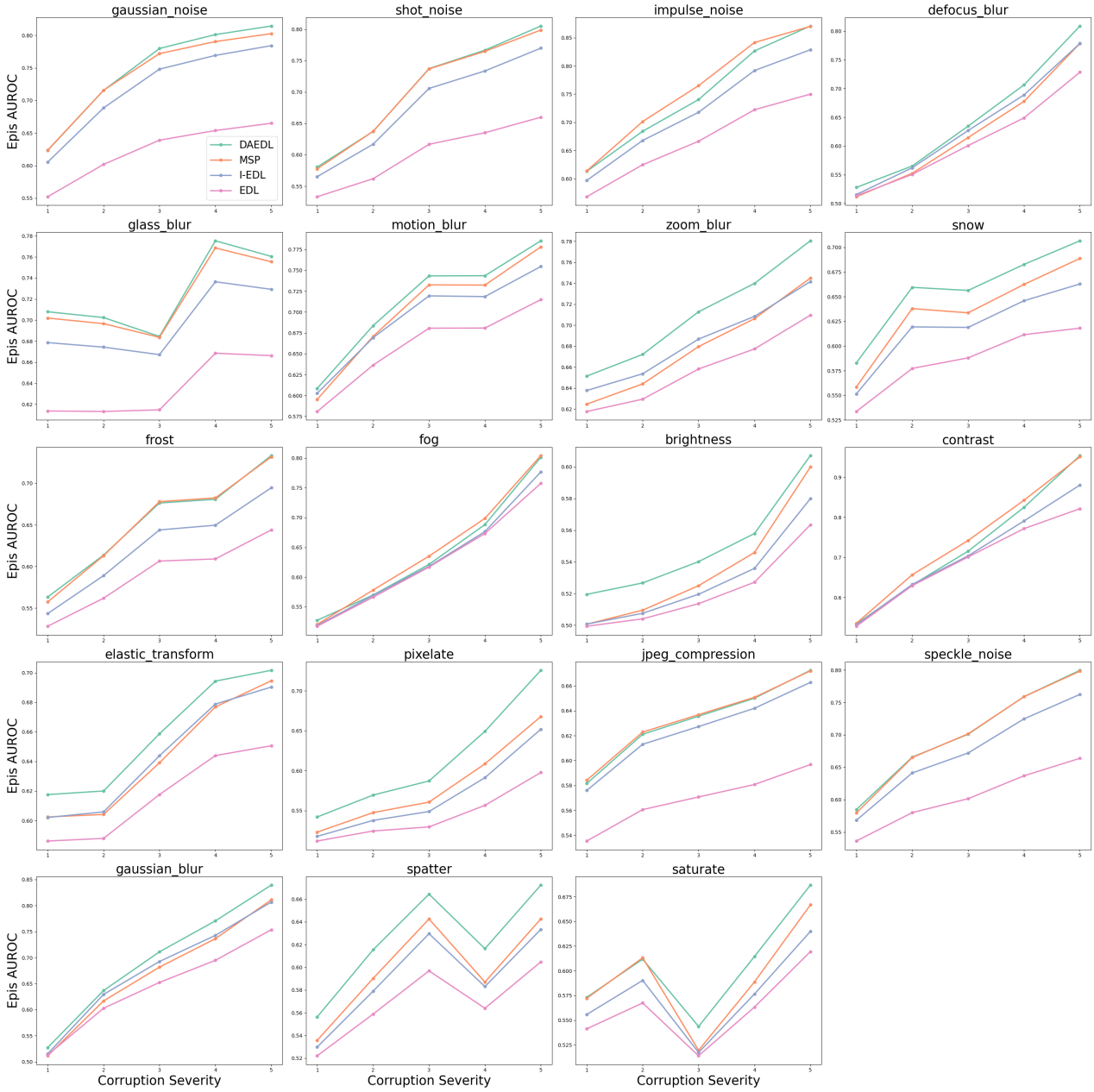


Figure 8. AUROC scores for distribution shift detection using epistemic uncertainty estimates across 19 different corruptions in the CIFAR-10-C dataset

D.3. Additional Results in Ablation Study (Section 6.5)

Table 10 presents the results of the ablation study conducted on the MNIST dataset. We can observe that each component of our model is effective individually and collectively demonstrates a synergistic effect, thereby enhancing the performance of EDL.

Table 10. Ablation study results on MNIST. The results of EDL (DAEDL without EXP, DE, and SN) were from Deng et al. (2023)

			MNIST → KMNIST		MNIST → FMNIST	
EXP	DE	SN	ALEA.	EPIS.	ALEA.	EPIS.
✗	✗	✗	97.02 ± 0.8	96.34 ± 2.0	98.10 ± 0.4	98.08 ± 0.4
✓	✗	✗	98.86 ± 0.0	98.89 ± 0.0	99.35 ± 0.0	99.48 ± 0.0
✓	✓	✗	99.74 ± 0.0	99.76 ± 0.0	99.65 ± 0.1	99.67 ± 0.1
✓	✗	✓	98.86 ± 0.1	98.90 ± 0.1	99.40 ± 0.1	99.50 ± 0.1
✓	✓	✓	$\mathbf{99.90 \pm 0.0}$	$\mathbf{99.92 \pm 0.0}$	$\mathbf{99.83 \pm 0.0}$	$\mathbf{99.87 \pm 0.0}$

E. Additional Explanations about the Concepts

To enhance the self-contained nature of our paper, we have included supplementary explanations for the concepts utilized in our study.

Dirichlet Distribution. The probability density function (PDF) of a Dirichlet distribution is formulated as follows:

$$\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{c=1}^{C} \alpha_c)}{\prod_{c=1}^{C} \Gamma(\alpha_c)} \prod_{c=1}^{C} \pi_c^{\alpha_c - 1},$$

where $\boldsymbol{\pi} \in \Delta^{C-1}$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_C], \forall \alpha_c > 0$ being the concentration parameters. Here, the expected probability for the cth class can be calculated as follows:

$$\bar{\pi}_c = \mathbb{E}_{\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha})}[\pi_c] = \frac{\alpha_c}{\alpha_0}.$$

In this paper, we utilize $\max_c \bar{\pi}_c$ (i.e., maximum expected class probability) and α_0 (i.e., precision of the Dirichlet distribution) as the measures of aleatoric and epistemic uncertainty, respectively. In addition to these metrics, DBU models provide various types of uncertainty measures in a closed form. For more comprehensive analysis and derivation of such measures, please refer to Ulmer et al. (2023).

Spectral Normalization. Spectral normalization (Miyato et al., 2018) is a technique that is applied to ensure a regularized feature space. It has been widely utilized in the context of single forward pass uncertainty estimation models. Specifically, spectral normalization is applied by estimating the spectral norm of the weight matrices and dividing the weight matrices by their spectral norm. Suppose that $f_{\boldsymbol{\theta}}$ is an L-layer neural network with weight matrices $\boldsymbol{\theta} = \{W^{(l)}\}_{l=1}^{L}$. We estimate the spectral norm $\sigma(W^{(l)})$ of the weight matrices for each layer using the power iteration method (Miyato et al., 2018; Gouk et al., 2021) and normalize the weights by dividing it by the corresponding spectral norm as follows:

$$W^{(l)} \leftarrow \frac{W^{(l)}}{\sigma(W^{(l)})}, \quad \sigma(W^{(l)}) = \max_{h:h \neq 0} \frac{\|W^{(l)}h\|_2}{\|h\|_2}.$$

For the updated $f_{\boldsymbol{\theta}}$ with the spectral normalized weight, the Lipschitz norm $\|f_{\boldsymbol{\theta}}\|_{\mathrm{Lip}}$ is bounded above by 1. Thus, we can ensure that the feature space distance is bounded by the input space distance. In other words, inequality $\|f_{\boldsymbol{\theta}}(\boldsymbol{x}_1) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_2)\|_2 \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$ holds $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^D$. This prevents the feature representations from being overly sensitive to the meaningless perturbation in the input space and ensures the representation to be more informative (Liu et al., 2020).

In this paper, we utilize spectral normalization to obtain a regularized feature space that enables meaningful feature space density estimates with GDA.

Distance Awareness. Distance awareness is a beneficial property that ensures the model to obtain high-quality uncertainty estimates. It has been widely utilized in the context of single forward pass uncertainty estimation models. Formally, distance awareness is defined as Definition E.1

Definition E.1. (Distance Awareness) (Liu et al., 2020) The predictive distribution $p(y|\mathbf{x})$ is *distance aware* if there exist $u(\mathbf{x})$, a summary statistic of $p(y|\mathbf{x})$ that quantifies model uncertainty that reflects distance between $\mathbf{x}$ and the training data with respect to $\|\|_{\mathcal{X}}$ i.e., $u(\mathbf{x}) = v(d(\mathbf{x}, \mathcal{X}))$, where v is a a monotonic function and $d(\mathbf{x}, \mathcal{X}) = \mathbb{E}_{\mathbf{x}' \sim \mathcal{X}} \|\mathbf{x} - \mathbf{x}'\|_{\mathcal{X}}^2$ is the distance between $\mathbf{x}$ and the training data domain.

In this paper, we prove that DAEDL satisfies *distance awareness* with respect to both the feature space and input space under certain conditions (Theorem 4.4, Corollary 4.5).