Can Uncertainty Quantification Benefit From Label Embeddings? A Case Study on Local Climate Zone Classification

Christoph Schweden, *Graduate Student Member, IEEE*, Katharina Hechinger, Göran Kauermann, and Xiao Xiang Zhu, *Fellow, IEEE*

***Abstract*— Modern deep learning models have achieved superior performance in almost all fields of remote sensing. An often neglected aspect of these models is the quantification and evaluation of predictive uncertainties. Regarding a classification task, this means that the focus of the analysis solely lies on performance metrics such as accuracy or the loss. On the other hand, a notion of uncertainty indicates the model's indecisiveness among the given classes and is essential to understand where the model struggles to classify the data samples. In this work, three levels of uncertainty are distinguished, starting with the typical softmax pseudo-probabilities as level-1 uncertainty. As a next level, the more flexible Dirichlet framework is utilized as model output space, and hereby also, a Bayesian setting with an uninformative prior is considered. For the level-3 uncertainty, an empirical Bayes setting is incorporated where a latent embedding of the label space is iteratively estimated by the marginal likelihood of the fully parameterized label space. The estimated embeddings are then learned by the network in three different settings: Two regression losses use the embeddings directly, while the closed-form solution of the Kullback–Leibler (KL) divergence uses the embedding parameterized as a Dirichlet distribution. To assess the different levels of uncertainty, the label evaluation subset of the So2Sat LCZ42 dataset, which contains label votes from multiple remote sensing experts, is investigated. The predictive uncertainties are evaluated by means of out-of-distribution (OoD) detection and calibration performance. Overall, the embedding-based approaches show strong performance for calibration, while, for the OoD experiments, the Bayesian Dirichlet setting with an uninformative prior achieves the best performance. In conclusion, embedded labels offer a flexible framework for incorporating uncertain or ambiguous labels into a supervised training setup. They could be highly beneficial for applications in fields such as urban planning or disaster response.**

***Index Terms*— Calibration, deep learning, label embedding, land cover classification, uncertainty quantification (UQ).**

Received 14 December 2024; revised 13 March 2025; accepted 21 March 2025. Date of publication 5 May 2025; date of current version 9 May 2025. This work was supported in part by German Federal Ministry of Education and Research (BMBF) in the framework of the International Future AI Laboratory "AI4EO–Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond" under Grant 01DD20001; in part by the Helmholtz Association through the Joint Research School Munich School for Data Science (MUDS); in part by German Federal Ministry for Economic Affairs and Climate Action in the framework of the "National Center of Excellence ML4Earth" under Grant 50EE2201C; and in part by Munich Center for Machine Learning. *(Corresponding author: Xiao Xiang Zhu.)*

Christoph Schweden is with the Department of Data Science in Earth Observation, Technical University of Munich, 80333 Munich, Germany, and also with the Remote Sensing Technology Institute, German Aerospace Center, 82234 Weßling, Germany.

Katharina Hechinger and Göran Kauermann are with the Department of Statistics, LMU Munich, 80799 Munich, Germany.

Xiao Xiang Zhu is with the Department of Data Science in Earth Observation, Technical University of Munich, 80333 Munich, Germany, and also with the Munich Center for Machine Learning, 80333 Munich, Germany (e-mail: xiaoxiang.zhu@tum.de).

Digital Object Identifier 10.1109/TGRS.2025.3562233

I. Introduction

In REMOTE sensing, the advantage of machine and deep learning methods over classical approaches has been shown in practically every aspect and problem setting. Various classes of models and deep neural network (DNN) architectures have been explored recently [2]. DNNs have achieved strong quantitative results in, for example, regression, segmentation, or classification tasks but they often come with a high computational cost of training millions or billions of parameters. Additionally, many DNN architectures represent a closed box that cannot be investigated adequately and quickly. This effectively means that, although solid performance can be achieved, the reason for and how the DNN reached its prediction cannot be explained simply. The first steps toward solving this problem have been summarized recently by [3].

Adding to that, the focus of scientific analyses often lies on the classical performance metrics. These metrics are, for example, represented by the accuracy of a model given a classification task, the mean squared error (mse) loss in the case of a regression problem, or the intersection over union (IoU) in object detection or segmentation tasks. While good performance in a classical view is essential and constantly desired, this view ignores the aspect of uncertainty. A model's uncertainty in its prediction is typically categorized broadly into two streams [4]. For one thing, aleatoric or data uncertainty exists, which naturally appears to some degree in every data source. By definition, this source of uncertainty cannot be reduced as it is inherent in the data.

Furthermore, the uncertainty of the model, termed epistemic uncertainty, quantifies all sources of uncertainty accompanying the modeling process of the data. This includes but not limited to the choice of model or DNN architecture, hyperparameter tuning, regularization or calibration techniques, and loss functions and training regimes in general. In contrast to aleatoric uncertainty, epistemic uncertainty is reducible. This means, in this particular context, that the uncertainty can be reduced by adding more informative data to the model. A reduction is desirable as lower epistemic uncertainty reflects a better

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

choice of a modeling approach for the data at hand. In practical terms and in the context of remote sensing, one could count sensor noise, occlusion, or varying resolution toward aleatoric uncertainty. In contrast, epistemic uncertainty could be steered by the choice of model, hyperparameter optimization, and regularization, among others.

While lower predictive uncertainty estimates are generally considered better, this conclusion is not always proper. First, note that an overall lower uncertainty would effectively translate to a lower epistemic uncertainty, as the data uncertainty is irreducible for a fixed dataset. Yet, the two sources of uncertainty are not typically independent in practice, and their estimation often relies on distributional and statistical assumptions dependent on the uncertainty quantification (UQ) framework. Furthermore, the disentanglement of the two sources of uncertainty is far from trivial [5].

The next problem arises when trying to directly interpret the estimated uncertainties. Apart from the *lower is better* slogan discussed above, a direct comparison between uncertainties of two different training regimes is not possible. Hence, a quantified uncertainty estimate twice as large as another does not directly translate to one prediction being twice as uncertain as the other prediction. Therefore, uncertainty estimates, in general, can be ranked on an ordinal scale, but relative distances cannot be mapped to a feasible metric space. Due to all these shortfalls, UQ methods are almost always linked to a downstream task. The three most common use cases for the classification setting are calibration, active learning, and out-of-distribution (OoD) detection.

For this present work, the focus lies on evaluating uncertainties by means of OoD detection and calibration analysis. A land cover classification dataset with a particularly interesting label structure is investigated: each satellite image was labeled independently by 10 remote sensing experts. The information gained from the multiple annotations was already successfully used for a deep learning pipeline by [6]. In a recent work, [1] explored the idea of transforming the multiple annotations into a latent label embedding. Here, the aim is to underline the effectiveness of such latent embeddings for a deep learning pipeline. In particular, the objective is to specifically incorporate uncertainty in the label space into our learning regime. This procedure differs from many classical UQ frameworks, which either rely on statistical or distributional assumptions or induce a probabilistic notion into the training.

Our core contributions can be summarized as follows.

1) A novel three-level UQ scheme for classification is established, which allows for incorporating prior information into the label space.
2) Within the third level of UQ, the embedded labels and their uncertainties are utilized for two downstream tasks, namely calibration and OoD detection, with the aid of three different loss functions.
3) For the calibration analysis, strong performance of the embedding-based approaches is observed over classical learning regimes, and for OoD detection, the second level UQ method yields the best overall results.

The article is structured as follows. Section II introduces works related to this one, Section III then introduces the methodology in use. Following this, Section IV explains the conducted experiments, which are discussed in Section V before Section VI concludes. The code for this work can be found at github.com/ChrisKo94/Learning_Distributions_LCZ

II. Related Work

The often observed *overconfidence* of DNNs almost always causes unreliable uncertainty estimates, which are unfavorable for all downstream tasks. In order to overcome the model's overconfidence, several so-termed *calibration* techniques have been established [7]. These techniques range from scaling the network's output or label space over replacing the model's output mapping with a distance-aware function to making specific changes to the network's architecture. The related calibration errors are several measures that estimate the degree of overconfidence. To do so, the model confidences (generally given by the pseudo-softmax probabilities) are compared against the accuracy of the model in prespecified subsets of the data. The closer the two measures lie together, the lower the calibration error and, hence, the better the model's degree of calibration.

For the task of active learning, data samples are iteratively added to the model and labeled by a human oracle in the process. The querying function deciding which data instances to add usually relies on an uncertainty quantity. Instances with the highest predictive uncertainty get added, which promises to help the model learn most effectively from the data. Ideally, the model can quickly benefit from added data samples that the model struggled to classify initially and can achieve good test performance after a few active learning iterations. Often, multiple querying strategies are benchmarked against one another and must outperform the random baseline, in which instances are queried entirely randomly. Popular recent approaches involve, exemplarily, a stochastic version of batch acquisition [8], conformal prediction-based acquisition [9], or deep Bayesian active learning [10].

Another application of UQ lies in the field of OoD detection. Here, the model initially gets trained on a separate dataset or a specific subset of the overall dataset. In the latter case, the subset is usually defined by a subset of classes easily distinguishable from the rest of the classes. The reason for this lies in the use case for the estimated uncertainty: after being trained on a particular dataset, the model is confronted with data samples from classes or datasets it has never seen before, so-termed OoD data, at test time. The predictive uncertainty here should ideally reflect the complete unknowingness of the model.

This uncertainty is then evaluated against the predictive uncertainty on the initial test dataset, which should generally yield lower uncertainties as the model is already acquainted with the data. Generally, two streams of approaches can be considered, namely those involving OoD data at training time and those that do not require this data for training. For the first, noteworthy advances have been made by energy-based neural networks [11] and prior neural networks [12]. In contrast, for the second approach, distance-aware architectures [13] or kernel-based output mappings [14] have been proposed. For recent advances in the field of remote sensing, see [15], who

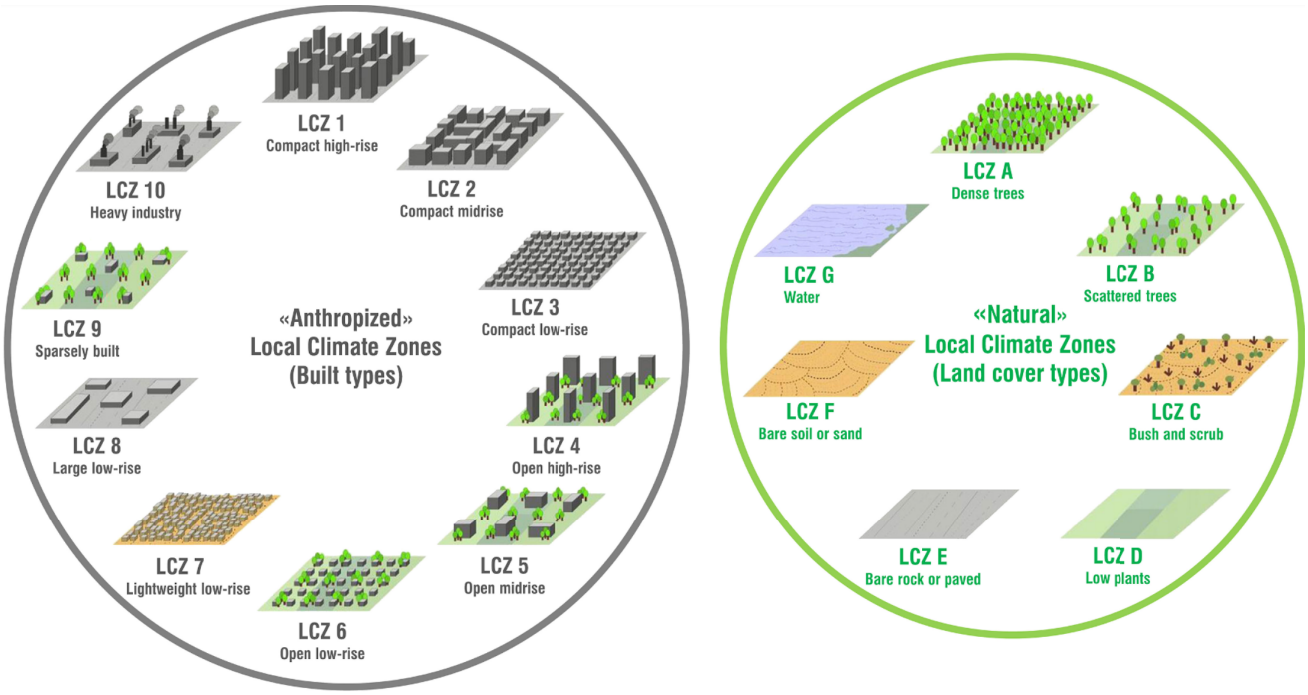

Fig. 1. Labeling scheme of the LCZs as presented in [17].

adapted prior neural networks to satellite imagery, or [16], who evaluated distance-aware neural networks for similar experimental settings.

UQ methods are quite novel in the DNN framework [18] and can generally be distinguished into three main streams. In one of them, single models are replaced by an ensemble of models [19]. The predictions of the individual models are then typically averaged and often lead to increased generalization performance and better model calibration. Next to ensembles, probabilistic methods have proven effective in improving the quality of UQ. Bayesian neural networks (BNNs) [20] are treating the network's weights as distributions, and sample from the network posterior to get a prediction. Another famous example is given by [21], who estimated predictive uncertainties by explicitly accounting for uncertainty via an additional network output node. This output, in turn, yields a posterior prediction from which the uncertainty is sampled.

A second main stream of UQ methods is given by single deterministic networks. These methods are categorized by using a single forward pass through the network for UQ with deterministic network weights. For the uncertainty to effectively measure the distance of a new instance from the training data, distance-aware modules are being built into the network. An early work termed *prior networks* explored the relaxation of the output space into a Dirichlet space and adjusting the loss function to the Kullback–Leibler (KL) divergence accordingly [12]. Similarly, Joo et al. [22] used the analytical solution to the expectation lower bound (ELBO) in a Bayesian belief matching framework, which is equivalent to minimizing the KL divergence. In the framework of deep kernel learning (DKL) [23], recent advances in deterministic UQ were, for example, made by [24], who chose a residual network (ResNet) architecture and modified the network by adding spectral normalization to the weight matrices and an approximate Gaussian process as output mapping. Furthermore, the use of two-sided gradient penalties in combination with trainable centroids in the DNN's latent space was explored by [25]. Here, the uncertainty was derived by the exponentiated distance between the model's latent prediction and the estimated centroids.

Finally, the third stream of deterministic UQ methods is given by the evidential neural networks [26]. The main idea here lies in treating classification model predictions as evidences embedded in the research stream of subjective logic [27], see [28] for a recent survey on the topic. This very active field of research has since then spread to other domains such as regression [29] or multitask learning [30]. For regression applications in particular, however, [31] argued that evidential regression suffers from overparameterization and should, therefore, instead be treated as a heuristic rather than an exact UQ. Recent exploratory works include the application of the variance-based regression approach to a classification setting [32], the theoretical analysis of the epistemic uncertainty estimates [33], and the use of a mixture of Dirichlet distributions coupled with variational inference [34]. Recently, also hybrid approaches have been proposed, which combine the strengths of BNNs with the efficiency and adaptability of deterministic models [35], [36], [37].

In the context of remote sensing, UQ still plays a minor role overall. Only a few papers focus on the methodological investigation of uncertainties, some of which are presented exemplarily in the following. [1], [6], and [38] explored the label uncertainty arising from multiple annotations in-depth for a land cover classification dataset. For the specific task of OoD detection, [15] investigated different versions of prior networks (see [12]) and showed strong results. Similar perfor-

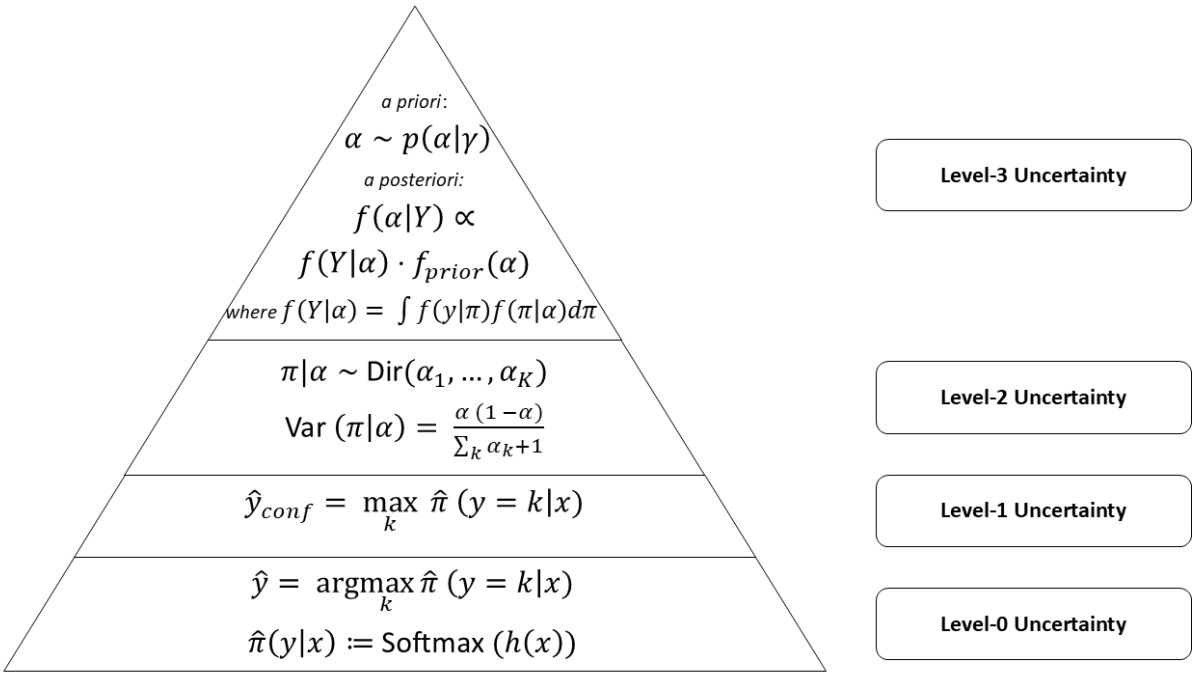

Fig. 2. Visualization of different levels of uncertainty as introduced in this work. Previously, [44] introduced similar levels of uncertainty for binary classification.

mance was achieved with distance-aware architectures [16]. The global estimation of precipitation was supported with statistical data uncertainty by [39]. In the context of cloud removal, [40] proposes a transformer-based architecture with attention and UQ modules, which yields better results and more calibrated probabilities. [41] used an uncertainty-based acquisition function for an active learning framework in the context of hyperspectral image classification. Using a BNN, [42] could effectively use UQ for corn yield prediction. For the analysis of soil moisture, [43] also reverted to uncertainty analysis.

III. Methodology

A. Data Description

The proposed methodology of learning from so-termed *label embeddings* is a generalized framework for dealing with multilabel datasets. More precisely, the framework incorporates information from multiple (expert) annotations per instance by means of a Bayesian statistical model. Since this contribution is extending the work of [1], it also uses the notation introduced there. Consider a labeled (remote sensing) dataset with individual data instances $x_1, \ldots, x_n$. For each image $x_i, i = 1, \ldots, n$, there exist multiple annotations from independent experts $j = 1, \ldots, J_i$. Each expert assigns a class $k = 1, \ldots, K$ to each image (or a subset of the entire dataset), which leads to the entirety of expert votes $V_{ij} \in \{1, \ldots, K\}$. When aggregating the individual expert votes into a single vector for each image i, this leads to the vote vectors.

$$\boldsymbol{Y}_i = (Y_{i1}, \ldots, Y_{iK}), \quad \text{where } Y_{ik} = \sum_{j=1}^{J_i} \mathbf{1}(V_{ij} = k). \tag{1}$$

The vote vectors can be aggregated into soft or hard labels in a variety of ways, which will be explained in more detail in Section IV-A. The dataset studied in this work covers a set of 42 urban conglomerates around the world and is named So2Sat LCZ42 [45]. Each data instance x_i represents a Sentinel-2 satellite image of a homogeneous region of size 320 × 320 m, which corresponds to 32 × 32 pixels (10-m ground sampling distance). A labeling scheme of 17 land cover classes was used for this dataset (see Fig. 1), namely the local climate zones (LCZs) [46]. For the purpose of investigating the reliability of this labeling scheme and process, [45] further chose a subset of 10 European cities, which 10 remote sensing experts independently labeled. This dataset is available online.[1] While an overall agreement rate of ≈85% was found, there exists a noteworthy degree of disagreement among the individual voters.

Using this extended label information for a DNN was previously studied by [6], who transformed the vote vectors into a soft label via the empirical distribution function. An increased generalization and calibration performance was achieved with adjusted labels and loss function. With this idea in mind, a more general UQ framework with two label embedding approximations will be introduced in the following. For the latter, a different approach is taken: with the aid of statistical machine learning, a latent embedding is estimated, which captures all existing labeling information across the entire dataset. This embedding can be seamlessly transformed into the form of a Dirichlet distribution, which can be considered as a second embedding structure. These embeddings will

[1] So2Sat LCZ42 evaluation dataset (link to download: https://mediatum.ub.tum.de/1659039)

eventually be integrated into a DNN as a core part of this work.

B. Uncertainty Framework

Before introducing the concept of label embeddings and their estimation, four different levels of uncertainty can be considered in a general classification problem, which are briefly touched in the following. Note that this differentiation builds upon previous works, in particular [44], which stated the levels for binary classification problems but already hinted at multiclass classification problems. The now following concepts are displayed in Fig. 2. Starting from the bottom, level-0 uncertainty is examined first. This level, in fact, holds no uncertainty at all, as it only considers the single class prediction of a neural network. More precisely, for a neural network h with parameters θ, let us denote the output of the network by $h_\theta(x)$ for a given input x. This output is of dimension K, corresponding to the number of classes considered. Typically, this unconstrained output is shaped into pseudo-probabilities by applying the softmax transformation as follows:

$$\hat{\pi}(y|x) := \text{Softmax}(h_\theta(x)) = \frac{\exp(h_\theta(x))}{\sum_k \exp(h_{\theta,k}(x))}. \tag{2}$$

The resulting point prediction is then given by $\hat{y} = \arg\max A_k \hat{\pi}(y|x)$ and in fact holds no notion of uncertainty. For the next level, therefore, the pseudo-probability associated with this point prediction is included, also often termed model *confidence* and given by $\hat{y}_{\text{conf}} = \max_k \hat{\pi}(y|x)$. This directly yields a level of uncertainty, often denoted via $1 - \hat{y}_{\text{conf}}$.

Often, this level of uncertainty is treated as sufficient, or one resorts to an uncertainty quantity based on the entropy of the softmax. Yet, the uncertainty's uncertainty can be inspected even further by relaxing the distributional assumption of the model's output space. Instead of a categorical distribution defined by the pseudo-softmax probabilities, the more general Dirichlet distribution is considered. This was previously performed by [22], who claim that the expanded label space can represent more rich information. Since the expectation of a Dirichlet distribution always yields a categorical distribution, there can be multiple Dirichlets that, in expectation, yield the same categorical distribution but inherit different levels of uncertainty associated with them. Let, therefore, the probability vector π (of which the model's softmax pseudo-probabilities are an estimate of) further be dependent on hyperparameters $\alpha_1, \ldots, \alpha_K$:

$$\pi|\alpha \sim \text{Dir}(\alpha_1, \ldots, \alpha_K). \tag{3}$$

These hyperparameters can be directly learned by the model, which is a common practice, for example, the *prior networks* [12]. The main advantage of this practice is the variety of possible uncertainty metrics based on the predictive Dirichlet distribution. Many metrics are considered for the experiments in this work; see Section IV-C for more details. As seen in Fig. 2, exemplarily, a notion of uncertainty can be determined for every class prediction given by the variance of the corresponding Dirichlet parameter.

As a novelty, now a third level of uncertainty is introduced. For this, an empirical Bayes framework is utilized to add an additional hierarchy to the existing Dirichlet-categorical distribution. First, a prior distribution of the α's dependence on some γ is assumed. Then, Bayes' theorem is used to find a distributional form proportional to the posterior of α given the data:

$$f(\alpha|Y) \propto f(Y|\alpha) \cdot f_{\text{prior}}(\alpha). \tag{4}$$

Here, the form of the marginal posterior can be received by integrating out π from the full posterior as follows:

$$f(Y|\alpha) = \int f(y|\pi) f(\pi|\alpha) d\pi. \tag{5}$$

The next step is crucial, as it allows us to estimate the α parameters in a supervised and data-driven manner. In particular, the label information of the entire dataset can be embedded into the label embedding parameters. For this, stochastic estimation techniques can be applied to sample from this marginal posterior and hence estimate α, which will be explained in more detail in the following. Note that this approximation now holds information from the entire dataset since the proportional posterior is dependent on Y and iteratively estimated using all existing label information of the training set.

C. Model Training Procedures

1) Estimating the Bayesian Embedding: Given a vote vector $\boldsymbol{Y}_i$ as defined in (1), the overall goal is to estimate a latent variable that incorporates uncertainty among voters about the ground truth label. In particular, the model is then capable of accounting for label ambiguity as opposed to restraining each instance to a single class. Utilizing the idea of a third layer of uncertainty, in the following, an algorithm to estimate the α parameters will be investigated. In order to do this in a data-driven manner, one resorts to an empirical Bayes method to estimate a prior distribution. Therefore, now our estimation approach from [1] is summarized briefly.

First, stepping back to the level-1 uncertainty previously introduced, the distributional assumption of the labels can be viewed slightly differently. Due to the nature of the labeling process, the voted labels can be treated as observations of a multinomial distribution (as opposed to a categorical distribution in the case of softmax pseudo-probabilities) given the sampled probabilities $\boldsymbol{\pi} := (\pi_1, \ldots, \pi_K)$, where K denotes the number of classes considered. The likelihood of the vote vector given the latent probabilities is then given as follows:

$$\boldsymbol{Y}|\boldsymbol{\pi} \sim \text{Mult}(\boldsymbol{\pi}, J). \tag{6}$$

Hence, the observed label votes are treated as if they were drawn from the multinomial distribution above. Again, J stands for the number of voters. A posteriori, as stated in (5), this yields

$$\begin{aligned} P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\alpha}) &= \int f(\boldsymbol{y}|\boldsymbol{\pi}) f(\boldsymbol{\pi}|\boldsymbol{\alpha}) d\boldsymbol{\pi} = \cdots \\ &= \frac{J!}{y_1! \cdots y_K!} \cdot \frac{B(\boldsymbol{\alpha} + \boldsymbol{Y})}{B(\boldsymbol{\alpha})} \end{aligned} \tag{7}$$

for the marginal distribution of the labels given the hyperparameters $\boldsymbol{\alpha}$ after integrating out π. Here, $B(\cdot)$ denotes the beta distribution. As previously outlined, Bayes' theorem as shown in (4) is used next, in order to sample α. The choice of the prior distribution is of uttermost importance here: it can either be specified in advance based on prior knowledge or, if prior knowledge is not available, estimated in a data-driven manner following an empirical Bayes approach. The latter is carried out here.

Note that now, instead of limiting oneself to the constrained α parameters from the conditional Dirichlet distribution in (3), the $\boldsymbol{\alpha}$'s are transformed into $\boldsymbol{Z}$ via

$$(z_1, \ldots, z_K) = \boldsymbol{Z} := \log(\boldsymbol{\alpha}) = (\log(\alpha_1), \ldots, \log(\alpha_K)) \quad (8)$$

which has the advantage of an unconstrained domain. As a prior for $\boldsymbol{Z}$, many options are available, but for simplicity and traceability, a normal distribution is chosen. Utilizing the empirical Bayes approach, the parameters of this $K-$dimensional normal prior distribution are estimated directly from the data using a stochastic version of the EM algorithm (see [47]). Concretely, $\hat{\boldsymbol{Z}} = (\hat{z}_1, \ldots, \hat{z}_K)$ is estimated from the approximate posterior via the mean of the Markov chain Monte Carlo (MCMC) samples from (7). The interested reader is referred to [1] for algorithmic details.

2) Learning the Embeddings With a Neural Network: The estimated embeddings can, in the following, be treated as a soft classification output in the form of Dirichlet parameters:

$$\hat{\boldsymbol{\alpha}}_i = (\hat{\alpha}_{i1}, \ldots, \hat{\alpha}_{iK}) := \exp(\hat{\boldsymbol{Z}}_i). \quad (9)$$

Given the estimated embeddings, the posterior Dirichlet distribution for an image i is formed via the following:

$$\begin{aligned} \mathrm{Dir}_{\mathrm{post},i} &:= \mathrm{Dir}(\hat{\alpha}_{i1}, \ldots, \hat{\alpha}_{iK}) \\ &=: \mathrm{Dir}\big(\hat{\alpha}_{\mathrm{embedding},\,i1}, \ldots, \hat{\alpha}_{\mathrm{embedding},\,iK}\big). \end{aligned} \quad (10)$$

At the same time, the untransformed output (i.e., the *logits*) of a neural network classifier $\boldsymbol{h}_\theta(x) : \mathcal{X} \to \mathbb{R}^K$ is observed, as appearing in (2), when considering a K class classification problem as follows:

$$\boldsymbol{f}_\theta(\boldsymbol{x}_i) = (f_\theta(\boldsymbol{x}_i)_1, \ldots, f_\theta(\boldsymbol{x}_i)_K). \quad (11)$$

To shape the output accordingly, $\exp(\cdot)$ is applied, and the transformation of the network's prediction now feasible for a Dirichlet distribution is given via the following:

$$\hat{\boldsymbol{A}}_{i,\mathrm{model}} := (\exp(f_\theta(\boldsymbol{x}_i)_1), \ldots, \exp(f_\theta(\boldsymbol{x}_i)_K)). \quad (12)$$

From this transformed prediction, the network's predictive Dirichlet distribution is formed via

$$\begin{aligned} \mathrm{Dir}_{\mathrm{post,\ model},i} &= \mathrm{Dir}(\exp(f_\theta(\boldsymbol{x}_i)_1), \ldots, \exp(f_\theta(\boldsymbol{x}_i)_K)) \\ &=: \mathrm{Dir}\big(\hat{\alpha}_{\mathrm{model},\,i1}, \ldots, \hat{\alpha}_{\mathrm{model},\,iK}\big). \end{aligned} \quad (13)$$

Having these two Dirichlet distributions, the loss function of the neural network can be formulated as the explicit solution of the KL divergence between the predicted Dirichlet of our network and the Dirichlet estimated by the embedding. This loss reads as follows:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \mathrm{KL}\big(\mathrm{Dir}_{\mathrm{post,\ model},i} \,||\, \mathrm{Dir}_{\mathrm{post},i}\big) = \cdots \\ &= \log\Gamma(\hat{\alpha}_{\mathrm{model},\,i0}) - \sum_k \log\Gamma(\hat{\alpha}_{\mathrm{model},\,ik}) \\ &\quad - \log\Gamma(\hat{\alpha}_{\mathrm{embedding},\,i0}) + \sum_k \log\Gamma(\hat{\alpha}_{\mathrm{embedding},\,ik}) \\ &\quad + \sum_k (\hat{\alpha}_{\mathrm{model},\,ik} - \hat{\alpha}_{\mathrm{embedding},\,ik}) \\ &\quad \times \big(\psi(\hat{\alpha}_{\mathrm{model},\,ik}) - \psi(\hat{\alpha}_{\mathrm{model},\,i0})\big). \end{aligned} \quad (14)$$

Here, $\hat{\alpha}_{\mathrm{model},\,i0} = \sum_k \hat{\alpha}_{\mathrm{model},\,ik}$ is set and $\hat{\alpha}_{\mathrm{embedding},\,i0} = \sum_k \hat{\alpha}_{\mathrm{embedding},\,ik}$. $\psi(\cdot)$ denotes the digamma function, namely $\psi(x) = (d/dx)\ln\big(\Gamma(x)\big)$. With the above loss formulation (14), it can be proceeded with training a prespecified DNN as usual, meaning backpropagating the loss with stochastic gradient descent (SGD).

The estimated unbounded embeddings $\hat{\boldsymbol{Z}} = (\hat{z}_1, \ldots, \hat{z}_K)$, on the other hand, reveal a generally unbounded but roughly limited from above and below nature. A natural choice of regression loss function is therefore given by the mse. Given the estimated embeddings and the model prediction, which now gets treated as the predicted embeddings as follows:

$$\begin{aligned} \hat{\boldsymbol{Z}}_{\mathrm{model}} &:= (\hat{z}_{1,\mathrm{model}}, \ldots, \hat{z}_{K,\mathrm{model}}) \\ &= \boldsymbol{f}_\theta(\boldsymbol{x}_i) = \big(f_\theta(\boldsymbol{x}_{i1}), \ldots, f_\theta(\boldsymbol{x}_{iK})\big). \end{aligned} \quad (15)$$

The mse loss function can be written out via

$$\mathrm{MSE}(\hat{\boldsymbol{Z}}_{\mathrm{model}}, \hat{\boldsymbol{Z}}) = \sum_{k=1}^{K} \big(\hat{z}_k - \hat{z}_{k,\mathrm{model}}\big)^2. \quad (16)$$

The Mahalanobis distance (MD) as loss function is of further interest. Particularly because this distance measure relies on an empirical estimation of correlation to scale the squared distance between the prediction and the ground truth. Such an estimation of inter-class correlation was measured during the algorithmic estimation procedure of the label embeddings by [1]. This estimated empirical covariance matrix is hence defined by $\hat{\Sigma}_{\mathrm{embedding}}$ and state the MD loss function as follows:

$$\mathrm{MD}(\hat{\boldsymbol{Z}}_{\mathrm{model}}, \hat{\boldsymbol{Z}}) = \sqrt{(\hat{\boldsymbol{Z}}_{\mathrm{model}} - \hat{\boldsymbol{Z}})^\top \hat{\Sigma}^{-1}_{\mathrm{embedding}} (\hat{\boldsymbol{Z}}_{\mathrm{model}} - \hat{\boldsymbol{Z}})}. \quad (17)$$

3) Simple Dirichlet Embedding: Building upon a strong foundation of Bayesian statistics, a more simple Dirichlet framework with no prior information and without the use of the estimated embeddings from [1] is investigated. For this, a noninformative prior on the α's of the form

$$\alpha_{\mathrm{prior},k} = c \quad \forall\, k = 1, \ldots, K \text{ a priori} \quad (18)$$

is chosen for a constant $c \in \mathbb{N}$. Using the same distributional assumption as in (6), the posterior Dirichlet parameters are looking as follows:

$$\alpha_{\mathrm{post},k} = c + y_k \quad \forall\, k = 1, \ldots, K \text{ a posteriori}. \quad (19)$$

Here, y_k denotes the entry of the vote vector $\boldsymbol{Y}$ as defined in (1) at the kth position. With these posterior Dirichlet parameters, it can be proceeded with training with the loss formulated in (14) as described above.

IV. Experimental Setup

A. General

For the experiments conducted, two different kinds of data splits were considered. Regarding the generalization and calibration experiments, a random split of the dataset between training and testing data was chosen (80/20). For the OoD detection framework, the instances were separated by two superior classes, namely the built classes (1–10) and the non-built classes (11–17 or A–G). Since there are little to no ambiguities between the two superior classes, the separation is clearly defined for the entire dataset. For the models using the expert label votes, the following decisions were made: As for the *one-hot* models, the single label was set to the majority vote of all experts, which is unique for every data instance. The training was then performed using the single label and the cross-entropy loss function. In the case of the *distributional* models, the empirical distribution of the expert votes was taken, namely $\boldsymbol{Y}_i/M$, where M denotes the number of experts. The loss was then constructed using the KL divergence. Another version of models, named *sampled one-hot*, was considered to reflect the spread of label votes per image. Instead of combining the label votes into a distribution, here, the label gets sampled from the different label votes. The sampling probabilities are denoted by the empirical share (i.e., if seven experts voted for class 1 and three experts for class 2, the sampling probabilities are 70% and 30%, respectively). The *simple Dirichlet* models take the Dirichlet parameters defined in (19) and use the exact solution of the KL divergence between two Dirichlets as defined in (14) as a loss function. For the KL embedding models, the same loss is used, but now the Dirichlet parameters are computed using the estimated embeddings as described in (9). For the last two classes of models considered, the raw embeddings were taken as targets. Then, the loss function was modified to the mse (16) in case of the mse embedding models and to the MD (17) in case of the MD embedding models. As a baseline model, Sen2LCZ [48] was chosen due to its simplicity and sovereignty over many state-of-the-art convolutional neural networks. Throughout all models, the hyperparameters were left unchanged. Namely, a stepwise decaying learning rate (initial $2{*}e^{-04}$, 50% decay after five epochs) was used for a maximum of 30 epochs, while an early stopping mechanism was in place based on the validation loss with a patience of five epochs. A batch size of 128 was considered, and the dataset was randomly shuffled with a tracked seed before training was initialized.

Note that due to numerical instabilities of the embedding algorithm for samples of class 7 (lightweight low-rise), the embeddings were estimated based on 16 classes, and samples with majority votes for class 7 were excluded from the analysis. For cases where label votes for class 7 occurred, these votes were added to class 3 (compact low-rise) instead, as there is a known high ambiguity among those two classes. Hence, the models considered in this work all cover only 16 of the 17 LCZs, and the distributional labels were adjusted accordingly. Note that class 7, mostly referring to slums, does not naturally occur in the European cities. The small number of samples in the complete dataset was, therefore, added from non-European cities. It is thus argued that the exclusion of class 7 does not play a crucial role in the label embedding framework, as there is little data support in the original dataset. Additionally, this little support, in combination with convergence issues in the embedding estimation, would lead to a skewed interpretation of the data.

For numerical stability, two substantial modifications were made to the training process. First, small ϵ's were added to the Dirichlet parameters accounting for the cases where they are (near) zero. Furthermore, a temperature [49] to scale the Dirichlet parameters was used because they are too big after being exponentiated (exemplarily, the standard range of the embeddings is $[-10, 10]$, which would be transformed to approximately $[1 \times 10^{-5}, 2.2 \times 10^{5}]$). A temperature of $t = 3$ was used, and the logits were backtransformed at inference time. This means, loss projection was effectively applied at training time and back projection at inference time.

B. Calibration Analysis

The most common performance metric in a classification setting is the accuracy of a classification model, that is, how often the predicted class matches the ground truth. For our present dataset, a ground truth class can be determined by looking at the majority vote of the individual expert label votes. However, this aggregation, of course, ignores the label uncertainty displayed by the spread of the label votes. Therefore, accuracy is solely considered for a *sanity check*, meaning it is investigated whether the model is generally able to learn the classification scheme. Since a lot of images are not perturbed by label uncertainty at all, the accuracy should furthermore not worsen on these samples when switching to a label embedding learning framework.

Of much bigger interest here is the degree of calibration of a classification DNN. Calibration describes the interdependence between the model's confidence in its prediction and the actual observed accuracy of the model. In practice, this means a model should adequately state its uncertainty about its prediction, which is naturally reflected in the model's confidence. The observed mismatch between the confidence and the accuracy of a trained model can be quantified by different types of *calibration errors*. As a first step, a set of bins splitting up the observable range $[0, 1]$ of confidences and accuracies needs to be set. This is typically done in an equidistant manner, but adaptive methods have also been introduced [50].

The notation of [7] is followed and these bins are denoted by $B_m = (((m-1)/M), (m/M)], \quad m = 1, \ldots, M$. In the experiments, different bin sizes were evaluated, and $M = 20$ was chosen as a reference for display. The indices of the images whose confidences $\hat{p}_i$ fall into the respective bin are then given by $I_m, m = 1, \ldots, M$. The needed quantities are then calculated via the following:

$$\text{acc}(I_m) = \sum_{i \in I_M} \mathbb{1}_{\{\hat{y}_i = \boldsymbol{y}_{\max,i}\}} \quad \text{and} \quad \text{conf}(I_m) = \sum_{i \in I_m} \hat{p}_i. \tag{20}$$

The accuracy per bin is, therefore, calculated as the proportion of correctly classified samples within a bin, while the confidence is the average predicted confidence for all

samples in the bin. In the formula, $\hat{y}_i$ is the predicted class for input $\boldsymbol{x}_i$, which is derived by the index of the maximum of the model's predicted embedding vector and the majority vote of image i is referred to as $\boldsymbol{y}_{\max,i}$. $\hat{p}_i$ then corresponds to the highest predicted probability, which was derived by taking the maximum of the posterior mean of the embeddings:

$$\hat{p}_i := \max_k E(\pi_k|\boldsymbol{Z}) = \max_k \frac{\exp(Z_k)}{\sum_{k'=1}^{K}\exp(Z_{k'})}. \tag{21}$$

The standard expected calibration error (ECE) is then derived via the following:

$$\mathrm{ECE} = \sum_{m=1}^{M}\frac{|I_m|}{n}|\mathrm{acc}(I_m) - \mathrm{conf}(I_m)|. \tag{22}$$

Further versions of the ECE include the maximum calibration error (MCE), which evaluates the maximum gap between confidence and accuracy out of all bins considered.

$$\mathrm{MCE} = \max_m |\,\mathrm{acc}(I_m) - \mathrm{conf}(I_m)\,|. \tag{23}$$

Additionally, the static calibration error (SCE) [50] evaluates calibration performance on a class level. For class k and within bin m, define accuracy and confidence via $\mathrm{acc}(m,k)$ and $\mathrm{conf}(m,k)$, respectively. The SCE is then given by the following:

$$\mathrm{SCE} = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\frac{n_{mk}}{n}|\mathrm{acc}(m,k) - \mathrm{conf}(m,k)| \tag{24}$$

where n_{bk} denotes the number of samples belonging to class k (with respect to the majority vote) within bin b.

Lastly, the KL divergence between the mean of the Dirichlet distribution (i.e., a pseudo-softmax probability distribution) predicted by the model and the mean of the Dirichlet distribution based on the embedding can be interpreted as a measure of generalization performance.

C. OoD Detection

At test time, the model's predictive uncertainty is first evaluated by means of OoD detection. Therefore, two types of samples are presented to the model, namely samples from the classes it was trained on (in-distribution or ID) and samples from the remaining classes (OoD). Then, the model predicts a label embedding for each sample. It is unimportant to look at performance measures such as accuracy, especially since the model cannot correctly predict the OoD data anyway, as it comes from classes unknown to the model.

Instead, a measure of uncertainty accompanying the prediction is focused and this uncertainty is used to test the model's performance to distinguish between ID and OoD data. This is done by means of a binary classifier, which is presented with all predictive uncertainties as scores, and classifies the samples into ID and OoD (0 and 1) according to this score. Ideally, a well-calibrated model will produce low uncertainty scores for ID data and higher uncertainty scores for OoD samples, facilitating effective classification. In classical neural networks, however, the tendency toward highly confident model predictions with little uncertainty values is often observed. This is observed even when the model is fed OoD data, which, in theory, should make the model unsure and, therefore, lead to high uncertainty.

For our dataset at hand, the classes are split into ID and OoD classes by using the intrinsic hierarchical class scheme. This scheme differentiates between built and non-built classes. The latter is composed of the classes A through G (or 11 through 17 if numbered continuously), while the built classes are defined by the initial 10 classes, 1 through 10. Both of the two hierarchical classes can be used for ID and OoD classes, respectively. The hierarchical split between built and non-built classes is justified by the structural differences between these two sets of classes: While the built classes refer to areas of anthropization (see Fig. 1, the non-built classes describe different land cover types. Furthermore, [15] already used this ID/OoD split for an earlier work.

Regarding the uncertainty score, various quantities can be considered. First, the focus lies on measures that take the pseudo-softmax probabilities

$$\hat{\pi}_k := \frac{\hat{\alpha}_{\text{model},\,k}}{\sum_{k'=1}^{K}\hat{\alpha}_{\text{model},\,k'}} \tag{25}$$

and the estimated posterior parameters $\hat{\alpha}_{\text{model},\,k}$ as defined in (13) as starting point. A widely used measure based on the distribution of the softmax pseudo-probabilities is the (Shannon) entropy. For the predictive distribution $\hat{P}(y|\pi)$, this reads as follows:

$$\begin{aligned} &H\left[\mathbb{E}_{p(\pi|x)}\left[P(y|\pi)\right]\right] \\ &= -\sum_{k=1}^{K}\hat{\pi}_k\cdot\log(\hat{\pi}_k) \\ &= -\sum_{k=1}^{K}\frac{\hat{\alpha}_{\text{model},\,k}}{\sum_{k'=1}^{K}\hat{\alpha}_{\text{model},\,k'}}\cdot\log\left(\frac{\hat{\alpha}_{\text{model},\,k}}{\sum_{k'=1}^{K}\hat{\alpha}_{\text{model},\,k'}}\right). \end{aligned} \tag{26}$$

As thoroughly explained by [28], this quantity measures the total uncertainty present in the prediction. The authors furthermore disentangle the total quantity into aleatoric and epistemic uncertainty by means of the expected entropy. This term is derived by taking the expectation of the entropy with respect to the predictive distribution; see [28] for details of the derivation. The expected entropy, which supposedly measures the aleatoric bit of the uncertainty, is quantified via the following:

$$\begin{aligned} &\mathbb{E}_{p(\pi|x)}\left[H\left[P(y|\pi\right]\right] = \cdots \\ &= \sum_{k=1}^{K}\hat{\pi}_k\left(\psi(\hat{\alpha}_{\text{model},\,k}+1) - \psi\left(\sum_{k'=1}^{K}\hat{\alpha}_{\text{model},\,k'}\right)\right) \end{aligned} \tag{27}$$

where ψ again denotes the digamma function $\psi(x) = (d/dx)\ln\left(\Gamma(x)\right)$. Of further interest can be the so-termed *distributional uncertainty*, which [28] depicts via the difference between the total and the aleatoric uncertainty. This quantity measures the amount of information about $\hat{\boldsymbol{\pi}} := (\hat{\pi_1},\ldots,\hat{\pi_K})$ given y and conditioned on x and is also termed mutual

TABLE I
CALIBRATION AND GENERALIZATION METRICS FOR ALL EVALUATED MODELS. ACCURACY METRICS INCLUDE OA, MAA, WAA, AND KAPPA STATISTIC (κ). CALIBRATION METRICS INCLUDE ECE, MCE, SCE, AND KL DIVERGENCE TO EMBEDDING (KL DIV.). BOLD VALUES INDICATE THE BEST-PERFORMING MODEL FOR EACH METRIC

	OA ↑	MAA ↑	WAA ↑	κ ↑	ECE ↓	MCE ↓	SCE ↓	KL-Div. to Emb. ↓
One-hot	90.3	59.8	88.4	88.4	4.4	18.8	0.6	0.40
Distr.	**91.4**	**80.7**	**91.1**	**89.8**	6.6	23.4	0.6	0.32
Sampled One-hot	90.7	76.7	90.2	89.0	6.8	23.3	0.7	0.32
Simple Dirichlet	89.6	72.4	88.7	87.6	4.1	90.0	**0.2**	0.48
KL Embedd.	89.9	66.9	89.3	88.0	**1.7**	**10.2**	0.3	**0.27**
MSE Embedd.	90.5	72.1	89.9	88.8	2.7	12.6	0.3	0.28
MD Embedd.	73.7	31.3	66.7	67.9	10.2	76.3	0.6	0.83

information. It reads as follows:

$$\begin{aligned} &I\left[y, \boldsymbol{\pi}|x\right] \\ &= H\left[\mathbb{E}_{p(\pi|x)}\left[P(y|\pi)\right]\right] - \mathbb{E}_{p(\pi|x)}\left[H\left[P(y|\pi\right]\right] = \cdots \\ &= -\sum_{k=1}^{K} \hat{\pi}_k \Bigg(\log \hat{\pi}_k - \psi(\hat{\alpha}_{\text{model},\,k} + 1) \\ &\qquad + \psi\left(\sum_{k'=1}^{K} \hat{\alpha}_{\text{model},\,k'} + 1\right)\Bigg). \end{aligned} \tag{28}$$

See [28] for a more detailed derivation. Next to the aleatoric and distributional bit of the uncertainty, multiple measures have been established to estimate the epistemic uncertainty. Most notably, the maximum softmax probability (MSP) can be considered, which is given via

$$\text{MSP}(\hat{\boldsymbol{\pi}}) := \max_k \hat{\pi_1}, \ldots, \hat{\pi_K}. \tag{29}$$

This measure is, for example, used by the works of [51] and [25]. Besides that, the Dempster–Shafer metric (DSM) was used by [26]. This measure is founded on the Dempster–Shafer theory of evidence (see [52] for details) and can be computed via the following:

$$\text{DSM}(\hat{\boldsymbol{\pi}}) := \frac{K}{\sum_{k=1}^{K} (\hat{\alpha}_{\text{model},\,k} + 1)}. \tag{30}$$

Lastly, the overall so-termed *evidence* of the predicted Dirichlet distribution can be considered as a measure for epistemic uncertainty as follows:

$$\text{Evidence}(\hat{\boldsymbol{\pi}}) := \sum_{k=1}^{K} \hat{\alpha}_{\text{model},\,k}. \tag{31}$$

V. RESULTS AND DISCUSSION

A. Calibration Results

Now, the quality of the predicted uncertainties is investigated by means of the pseudo-softmax probabilities for all models previously introduced. For these experiments, there is no split between ID and OoD data, rather the models trained on all classes are evaluated on a test bed. In Table I, the calibration and generalization results for the different models are presented.

Note first that the typical one-hot encoded model achieves the second-highest values in the classical performance metrics. Next to the classical overall accuracy (OA), the macro average accuracy (MAA), which takes the average of the class-wise accuracies, and the weighted average accuracy (WAA), which weighs the class-wise accuracies by the class share in the dataset, are displayed. The κ statistic [53] furthermore describes the inter-rater agreement between the prediction and the label. This phenomenon should receive little attention because there is no kind of uncertainty presented to the model. Therefore, the model solely focuses on classifying the single ground truth class correctly. The slight performance increase of the model trained with the soft labels based on the empirical distribution of the votes (*Distr.*) can be potentially explained by smoother training enabled via the natural regularization of the labels.

When now focusing on the calibration metrics as introduced earlier, the clear advantage of the embedding-based approaches becomes apparent. The ECE of the KL embedding-based model is reduced by more than 50% compared to traditional models such as the one-hot or distributional label-based models. The message is similar for the mse loss in comparison with embedding-based models. These strong calibration results imply the practical usage of the embeddings for remote sensing classification tasks since a high degree of reliability is thereby enforced. Solely for the MD loss, the results are not convincing. It is argued that the underlying correlation structure of the embeddings might be too complex to enable smooth training and can, on the other hand, hinder calibration performance. The look at MCE reveals a picture similar to that of the ECE. Note that the MCE depends highly on the choice of number and width of bins, therefore the high MCE of the simple Dirichlet model is not problematic. The SCE is also generally much lower for the embedding-based models, yet here the simple Dirichlet model can also shine and beats the embedding-based models by a small margin. Note, however, that the gap to the traditional models remains extremely high. As another performance metric of interest, the KL divergence between the models' predictions and the categorical distribution defined over the latent embeddings of the test set (i.e., the

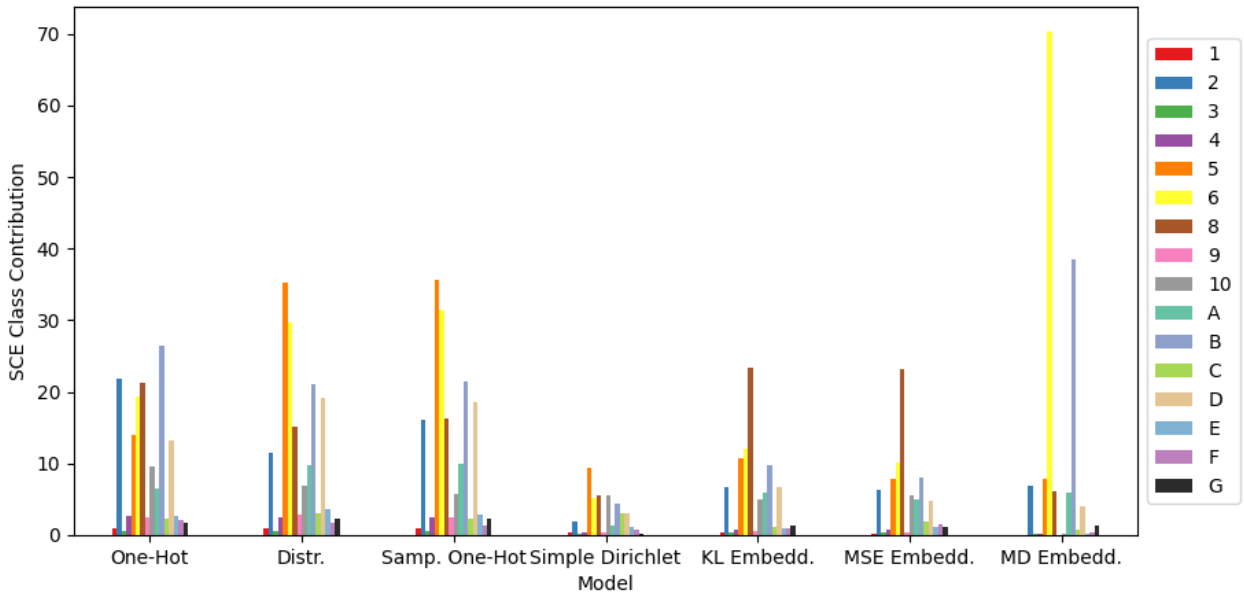

Fig. 3. SCE contributions per class for all models. Classes: 1 (compact high-rise), 2 (compact midrise), 3 (compact low-rise), 4 (open high-rise), 5 (open midrise), 6 (open low-rise), 8 (large low-rise), 9 (sparsely built), 10 (heavy industry), A (dense trees), B (scattered trees), C (bush and scrub), D (low plants), E (bare rock or paved), F (bare soil or sand), and G (water).

expectation of the Dirichlet) is considered. This metric can be seen as a measure of generalization performance toward the unseen embedding at test time. With regard to this metric, the embedding-based approaches clearly outperform all other models by a large margin.

For a closer look at calibration performance, Fig. 3 shows the individual class contributions to the overall SCE by method. In more detail, the individual summands of the sum over $k = 1, \ldots, K$ in (24) are visualized and explain the individual class contributions to the overall SCE. The SCE is a more fine-grained calibration measure than the ECE, as it derives calibration performance on a class level. In the analysis, now the entirety of the softmax probability distribution is considered. This closer look reveals that despite a few outliers (e.g., class 8 and class E), the Bayesian embedding approaches, with the exception of the seemingly too vague Mahalanobis embedding approach, lead to increased calibration performance on a class level. The construction of the Bayesian embeddings backs this up: via integrating label uncertainty among samples across the entire dataset, a more realistic and uncertainty-aware embedding is created. With this embedding, the entire softmax probability vector better reflects the label uncertainty present in the data. Regarding the outliers, the low calibration performance can be partially explained by a small share of the respective classes in the data (hence little support for effective and confident learning) and small confusion to other classes (hence spatially separated embeddings leading to suboptimal learning). For an in-depth analysis of the class votes, see [38].

B. OoD Detection

The OoD detection performance for all models described in Section IV-A is shown in Table II. As indicated by the first two columns, all models are trained in two different scenarios: for once, the models are trained using randomly 80% of the data samples where the majority vote was cast for a built class (1–10), of which 25% are reserved for validation at the end of each epoch. The remaining 20% is reserved for the testing stage. At this stage, OoD data is presented to the model belonging to the non-built classes (A–G), with respect to the class voted for by the majority of the experts. Identically, the models in the second scenario are trained with parts of the data belonging to the non-built classes (with respect to the majority vote) and are then presented with OoD data from built classes. Based on the initial label evaluation study performed by [45], these two superior classes, namely built and non-built, are close to mutually independent. This means almost no votes were cast for the other overarching class, different from the overarching class the majority voted class belongs to.

In a first step, all uncertainty metrics presented in Section IV-C are computed for the ID and OoD test samples. For all predicted uncertainties, the corresponding areas under the curve (AUCs) are displayed wherever computable. In more detail, for the softmax-based approaches (*one-hot*, *distributional*, and *sampled one-hot*), uncertainty metrics that are based on Dirichlet parameters cannot be computed. These metrics are namely the DSM, the *expected entropy*, the overall *evidence*, and the *distributional uncertainty*. Note that these AUCs are based on a binary classifier, which is constructed to differentiate between ID and OoD data samples only based on the predictive uncertainties that are presented at test time. Both the area under the precision–recall curve (AUPR), which compares the precision and recall of the binary classifier, and the area under the receiver operator curve (AUROC), which compares the recall and the false positive (FP) rate of the binary classifier, are investigated for a varying decision threshold. In more detail, given the true positives (TPs), true negatives (TNs), FPs, and false negatives (FNs) of the classifier results,

TABLE II

OoD Detection Results for All Models and a Range of Different Uncertainty Metrics. Two Scenarios Are Considered, Where Built and Non-Built Classes Are Used for ID and OoD, Respectively, and Vice Versa. Areas Under Curves Are Calculated for a Binary Classifier Built Upon the Predictive Uncertainties. The OoD Uncertainties Are Labeled With 1; the ID Uncertainties Are Labeled With 0. The Binary Classifier Then Tries to Separate the Two Kinds of Uncertainties by Estimating a Threshold. AUROC = Area Under Precision–Recall Curve. AUPR = Area Under Precision–Recall Curve, DSM = Dempster–Shafer Metric (30), MSP = 1 − Maximum Softmax Probability (29), Pred.Ent. = Predictive Entropy (26), Exp.Ent. = Expected Entropy (27), Evid. = Evidence (31), Distr.Unc. = Distributional Uncertainty (28). The Best Results Are Indicated in Bold

ID Data	OoD Data	Setting	AUROC ↑						AUPR ↑					
			DSM	MSP	Pred.Ent.	Exp.Ent.	Evidence	Distr. Unc.	DSM	MSP	Pred.Ent.	Exp.Ent.	Evid.	Distr. Unc.
Built-up	Non-built	One-Hot		78.8	79.8					76.9	77.5			
		Distr.		82.4	78.4					77.5	73.4			
		Sampled One-Hot		79.0	73.3					74.0	69.9			
		Simple Dirichlet	81.7	**82.7**	80.9	81.7	18.3	79.3	74.9	**78.1**	74.8	74.9	33.8	74.1
		KL Embedd.	70.0	66.1	68.3	70.0	30.0	66.5	61.7	61.6	63.7	61.7	37.4	63.0
		MSE Embedd.	44.7	35.5	36.0	59.9	55.3	53.1	57.6	40.8	43.5	58.3	58.3	58.8
		MD Embedd.	45.8	51.8	47.1	48.0	54.2	53.9	49.3	56.6	51.4	47.4	50.2	50.4
Non-built	Built-up	One-Hot		96.2	96.7					93.2	94.6			
		Distr.		95.3	96.4					92.7	94.8			
		Sampled One-Hot		94.7	95.6					92.4	94.1			
		Simple Dirichlet	**96.7**	96.6	96.7	96.7	3.3	66.4	94.2	95.3	**95.6**	94.2	31.1	78.3
		KL Embedd.	95.6	93.4	93.3	95.6	4.4	74.5	95.1	91.6	91.7	95.1	31.1	81.8
		MSE Embedd.	56.9	70.4	64.3	31.2	43.1	52.4	57.2	67.5	63.4	39.1	48.7	55.5
		MD Embedd.	92.2	92.7	94.0	9.7	7.8	9.8	91.6	88.2	91.8	32.3	31.9	33.3

the precision is computed as [TP/(TP + FP)], the recall via [TP/(TP + FN)], and the FP rate via [FP/(FP + TN)]. In order to make the AUCs expressive, the size of the OoD dataset and the ID test set was balanced (namely, whichever dataset was larger was cropped to the size of the smaller one). If this balancing is ignored, the resulting plot would be skewed as there are far more non-built data samples than from the built classes.

Looking at the results of non-built or vegetation as OoD detection, the difficulty of the task becomes immediately visible. Apparently, for the given space of considered models and uncertainty metrics, there is an upper limit of ≈83% in AUC. Overall, the performance of the embedding-based approaches is lacking a bit behind the other models for all uncertainty metrics considered. Exemplarily, the KL embedding model only achieves 66.1% AUROC for the MSP uncertainty compared to the distributional model with 82.4%. This can potentially be explained by a high level of data-specific uncertainty in the embedding space for the built classes. By learning this data-specific uncertainty during training, the predictive ID uncertainty remains data-specific and hinders the binary classifier from successfully differentiating between ID and OoD uncertainty. Note that this phenomenon is not surprising, as the estimated voting uncertainty is purposely fed, which is highly data-specific, to the model via the Bayesian embeddings. Hence, the differentiation between training and test data (from a different data regime) is less obvious. The big advantage of such uncertainty-aware training lies within the better-calibrated uncertainties, as was seen in the calibration analysis.

For the opposite case where the built-up classes are OoD, the simple Dirichlet model achieves an AUROC of 96.7% compared to the KL embedding model's 95.6% for the DSM uncertainty (see Table II). Regarding the AUPR, the gap is less narrow; here the simple Dirichlet achieves 95.6% and whereas the best embedding-based model (MD embedding) only achieves 91.8%. This overall best result emphasizes the advantage of Dirichlet embedding models, particularly with the uninformative prior. The uninformativeness a priori ensures training without too strong a focus on the ID data (as opposed to the Bayesian embedding models). This then helps to better deal with OoD samples at test time. Strong results can be hereby achieved across many of the uncertainty metrics considered. Note that three of them can only be calculated if a predictive Dirichlet distribution is formed by the model. This strong performance of the simple Dirichlet model across AUROC and AUPR for various uncertainty metrics underlines the robustness of the proposed approach. Fig. 4(a) shows the relatively clear separation of ID and OoD uncertainties (MSP as uncertainty metric) of this model. While most of the ID uncertainties are centered near zero and sparsely scattered outwards, the OoD uncertainties are mostly taking much higher values. This leads to a generally good separability of the two datasets. On the right side, the corresponding ROC curve [see Fig. 4(b)] is displayed, showing a moderate increase of the curve for varying the decision threshold.

In the opposite case, where built samples are presented as OoD data, a much higher level of AUC performance can be seen. This can be attributed to the clear separation of the individual non-built classes. The clear separation, in turn, yields very confident models. When presented with built samples, which are fundamentally different in their nature,

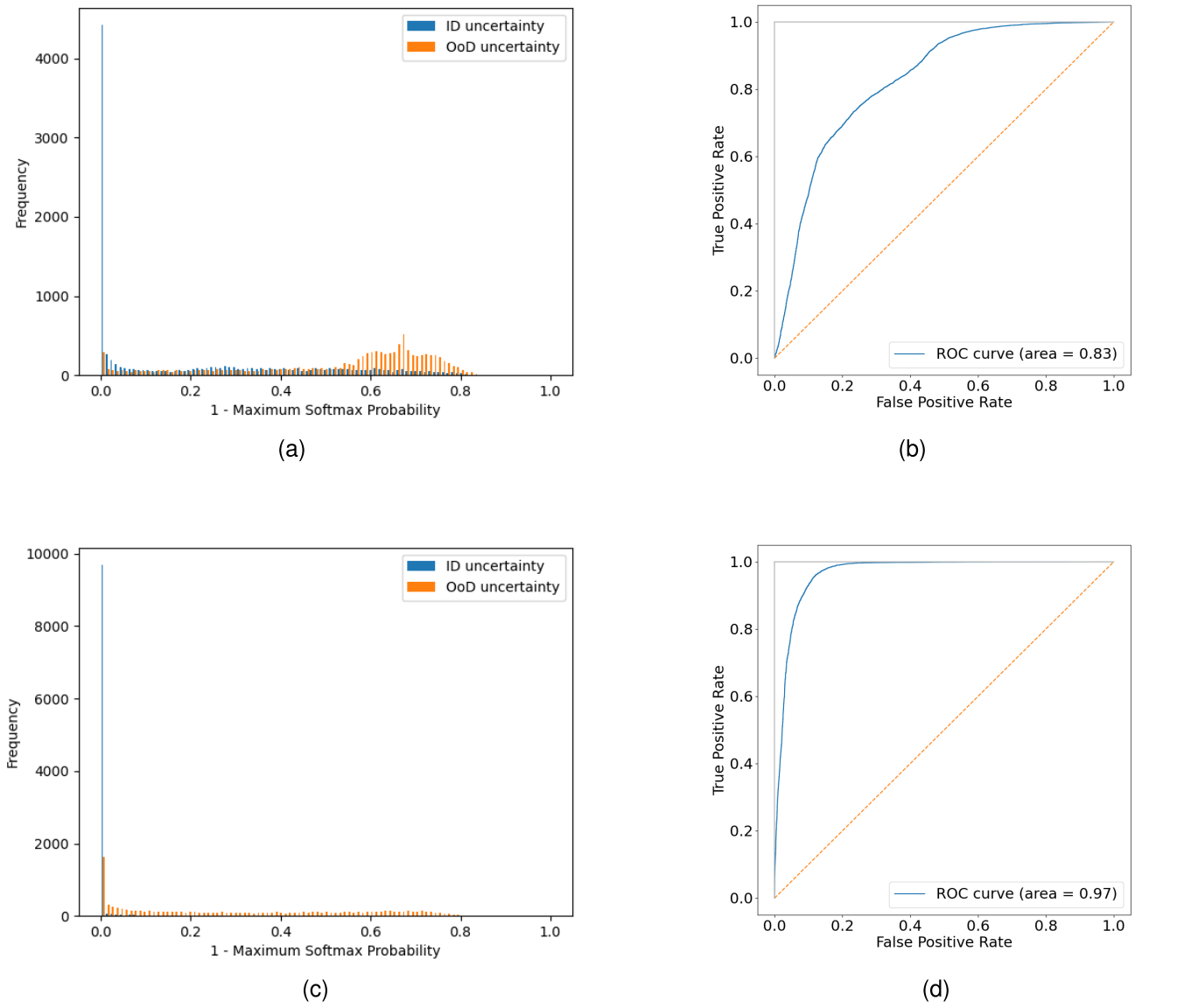

Fig. 4. Exemplary (a) and (c) histograms of predictive uncertainty scores and (b) and (d) ROC curves for binary OoD classifier. The two plots show the interdependence between the grouped uncertainty quantities and the performance of the binary OoD detector.

a small predictive uncertainty is sufficient to effectively distinguish this sample from the confident (hence near-zero) ID uncertainties. This phenomenon becomes clearly visible in Fig. 4(c), where a large spread of OoD uncertainties stands against a pile of near-zero ID uncertainties in the case of the best-performing model (simple Dirichlet). This is then reflected in a near-perfect ROC curve and AUROC metric depicted in Fig. 4(d). Again, the uninformativeness a priori seemingly helps to keep the model cautious of OoD samples at test time.

An interesting observation of this OoD setting is that the Bayesian embedding-based approaches are performing similarly well to the other models. It can be concluded that the overall lower inter-class uncertainty of the non-built classes, which is visible in both the label distributions and the corresponding estimated embeddings, helps sufficiently well to make the training OoD-aware. Especially when solely looking at distribution-based uncertainty metrics such as expected entropy and distributional uncertainty, the Bayesian embedding models generally achieve higher detection performance than the Dirichlet-embedding models, although they are trained with data-specific uncertainty.

Generally speaking, a lot of uncertainty metrics that were considered might not be suitable for the present OoD detection task or the investigated models. The *expected entropy*, theoretically measuring the aleatoric part of the uncertainty, can only give sense-making results for the simple Dirichlet model and the KL embedding. This is because all results below 50% can be discarded; they are worse than random guessing. Following this, the *distributional uncertainty* is also mismatched, likely because it includes the expected entropy as a term. The performance result using the *evidence* as

uncertainty metric is also underperforming. It is believed this is due to the fact that this quantity is unnormalized and does not align with the unconstrained embedding space on which it is based. Even for the somewhat contained Dirichlet space, this metric is not suitable.

VI. Conclusion

This study introduced a three-level uncertainty scheme, including a Bayesian embedding setup that demonstrated superior calibration across multiple metrics. For the third-level uncertainty, two types of label embeddings were investigated: a Bayesian setup to estimate an unconstrained label embedding as well as a Dirichlet setup with an uninformative prior. A core idea of the Bayesian embedding approach is the loosening of the restrictions of typical label spaces defined, for example, Dirac or categorical distributions. These embeddings have then been incorporated into a deep learning framework through various transformations and adapted loss functions. In particular, the embeddings were, in one case, utilized as Dirichlet parameters of a predictive distribution, which was then learned by means of the explicit solution of the KL divergence between Dirichlet distributions. In other cases, the untransformed embeddings were directly learned by the network with the aid of two popular regression loss functions.

In the studied dataset LCZ42, label uncertainty is present in the form of multiple expert annotations per image. The aim of the Bayesian embedding approaches is, therefore, to project this uncertainty into the embedding space and, in the following, make the network learn in an uncertainty-aware manner. Similarly, the uncertainty is projected into the Dirichlet space for the Dirichlet embedding. Two separate test beds were constructed to evaluate the approaches, namely an OoD detection framework that evaluates the model's predictive uncertainties by means of different uncertainty metrics and a calibration framework that examines the pseudo-softmax probabilities. In the calibration analysis, the Bayesian embedding approaches, with the partial exception of the MD loss, can show significant improvements in terms of calibration performance. The most common performance metrics are reduced by more than 50% compared to traditional learning regimes. In particular, the KL embedding model reduced the ECE by over 60% compared to the one-hot model, as shown in Table II. In the OoD detection experiments, the simple Dirichlet approach could show strong OoD detection results compared to models learning with a constrained label space. Regarding the AUROC metric, for example, the Dirichlet model achieved 95.6% as opposed to the 94.8% achieved by the distributional model. Here, the uninformative prior seemingly helped best to differentiate unseen OoD data from the ID data as seen during training. The Bayesian embedding models, which were trained with data-specific uncertainty, could not sufficiently well detect OoD data.

Though the introduced techniques can capture the label uncertainty arising from the expert labeling pipeline in a broad range, these techniques are, of course, limited to the number and diversity of experts considered. While the 10 experts provide a solid basis for label uncertainty, the question arises as to whether more labels per image or more images are desirable. [54] discusses this question in an NLP context and concludes that more labels can sharpen the latent class borders better than in the opposite case. For satellite image classification, answering this question is an open issue. Next to label-based embeddings, an interesting direction would be to incorporate semantic information of the different LCZ classes into an embedding, potentially with an unsupervised, semi-supervised, or self-supervised approach. This stream is left for future research. Additionally, in terms of generalization of the proposed framework, it would be wise to broaden the scope to other datasets with a similar label structure. Unfortunately, to the best of our knowledge, there does not exist a similar large-scale dataset in the remote sensing domain. For an extensive analysis of the proposed framework, which lies out of the scope of this journal, the interested reader may be referred to [55] for an overview of many datasets with noisy and ambiguous labels across domains.

References

[1] K. Hechinger, C. Koller, X. Xiang Zhu, and G. Kauermann, "Human-in-the-loop: Towards label embeddings for measuring classification difficulty," 2023, *arXiv:2311.08874*.

[2] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[3] A. Höhl et al., "Opening the black-box: A systematic review on explainable AI in remote sensing," 2024, *arXiv:2402.13791*.

[4] E. Hullermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, 2021.

[5] M. Valdenegro-Toro and D. S. Mori, "A deeper look into aleatoric and epistemic uncertainty disentanglement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1508–1516.

[6] C. Koller, G. Kauermann, and X. X. Zhu, "Going beyond one-hot encoding in classification: Can human uncertainty improve model performance in Earth observation?" *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5400211.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Intl. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 1321–1330.

[8] A. Kirsch, S. Farquhar, P. Atighehchian, A. Jesson, F. Branchaud-Charron, and Y. Gal, "Stochastic batch acquisition: A simple baseline for deep active learning," 2021, *arXiv:2106.12059*.

[9] S. Matiz and K. E. Barner, "Inductive conformal predictor for convolutional neural networks: Applications to active learning for image classification," *Pattern Recognit.*, vol. 90, pp. 172–182, Jun. 2019.

[10] A. Kirsch, J. V. Amersfoort, and Y. Gal, "BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 7026–7037.

[11] W. Liu, X. Wang, D. J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2020, pp. 21464–21475.

[12] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Dec. 2018, pp. 7047–7058.

[13] J. Z. Liu et al., "A simple approach to improve single-model deep uncertainty via distance-awareness," *J. Mach. Learn. Res.*, vol. 24, no. 42, pp. 1–63, 2023.

[14] J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal, "On feature collapse and deep kernel learning for single forward pass uncertainty," 2021, *arXiv:2102.11409*.

[15] J. Gawlikowski, S. Saha, A. Kruspe, and X. X. Zhu, "An advanced Dirichlet prior network for out-of-distribution detection in remote sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5616819.

[16] C. Koller, P. Jung, and X. X. Zhu, "Can land cover classification models benefit from distance-aware architectures?" *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024.

[17] M. Migliari, E. Briche, J. Despax, L. Chesne, and O. Baverel, "A cross-analysis matrix comparing multi-site local climate zone trends: Application to identify sustainable built morphologies improving summer daytime urban microclimate," *Sustain. Futures*, vol. 7, Jun. 2024, Art. no. 100162.

[18] J. Gawlikowski et al., "A survey of uncertainty in deep neural networks," *Artif. Intell. Rev.*, vol. 56, no. 1, pp. 1513–1589, Jul. 2023.

[19] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2016, pp. 6405–6416.

[20] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. Cham, Springer, 2012.

[21] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision," in *Proc. Adv. Neural Inf. Process. Syst.*, Mar. 2017, pp. 5580–5590.

[22] T. Joo, U. Chung, and M.-G. Seo, "Being Bayesian about categorical probability," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4950–4961.

[23] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep Kernel learning," in *Proc. 19th Int. Conf. Artif. Intell. Statist.*, 2016, pp. 370–378.

[24] J. Z. Liu et al., "A simple approach to improve single-model deep uncertainty via distance-awareness," *arXiv:2205.00403*, 2022.

[25] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Uncertainty estimation using a single deep deterministic neural network," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9690–9700.

[26] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018, pp. 9690–9700.

[27] A. Jsang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Cham, Switzerland: Springer, 2018.

[28] D. T. Ulmer, C. Hardmeier, and J. Frellsen, "Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation," *Trans. Mach. Learn. Res.*, 2023. [Online]. Available: https://openreview.net/forum?id=xqS8k9E75c

[29] A. Amini, W. Schwarting, A. P. Soleimany, and D. Rus, "Deep evidential regression," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 14927–14937.

[30] D. Oh and B. Shin, "Improving evidential deep learning via multi-task learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7895–7903.

[31] N. Meinert, J. Gawlikowski, and A. Lavin, "The unreasonable effectiveness of deep evidential regression," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, Jun. 2023, pp. 9134–9142.

[32] R. Duan, B. Caffo, H. X. Bai, H. I. Sair, and C. Jones, "Evidential uncertainty quantification: A variance-based perspective," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2121–2130.

[33] M. Jürgens, N. Meinert, V. Bengs, E. Hüllermeier, and W. Waegeman, "Is epistemic uncertainty faithfully represented by evidential deep learning methods?" 2024, *arXiv:2402.09056*.

[34] J. J. Ryu et al., "Improved evidential deep learning via a mixture of Dirichlet distributions," 2024, *arXiv:2402.06160*.

[35] D. T. Chang, "Hybrid Bayesian neural networks with functional probabilistic layers," 2021, *arXiv:2107.07014*.

[36] S. Jose, S. Louis, S. Dascalu, and S. Liu, "Transfer learning-based hybrid approach for Bayesian network structure learning," *Int. J. Artif. Intell. Tools*, vol. 31, no. 7, Nov. 2022, Art. no. 2260003.

[37] S. I. Allec and M. Ziatdinov, "Active and transfer learning with partially Bayesian neural networks for materials and chemicals," 2025, *arXiv:2501.00952*.

[38] K. Hechinger, X. X. Zhu, and G. Kauermann, "Categorising the world into local climate zones: Towards quantifying labelling uncertainty for machine learning models," *J. Roy. Stat. Soc. Ser. C, Appl. Statist.*, vol. 73, no. 1, pp. 143–161, Jan. 2024.

[39] J. Dong et al., "Statistical uncertainty analysis-based precipitation merging (SUPER): A new framework for improved global precipitation estimation," *Remote Sens. Environ.*, vol. 283, Dec. 2022, Art. no. 113299.

[40] P. Ebel, V. S. F. Garnot, M. Schmitt, J. D. Wegner, and X. X. Zhu, "UnCRtainTS: Uncertainty quantification for cloud removal in optical satellite time series," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 2086–2096.

[41] X. Liao, B. Tu, J. Li, and A. Plaza, "Class-wise graph embedding-based active learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5522813.

[42] Y. Ma, Z. Zhang, Y. Kang, and M. Özdoğan, "Corn yield prediction and uncertainty analysis based on remotely sensed variables using a Bayesian neural network approach," *Remote Sens. Environ.*, vol. 259, Jun. 2021, Art. no. 112408.

[43] J. Liu et al., "Uncertainty analysis of eleven multisource soil moisture products in the third pole environment based on the three-corned hat method," *Remote Sens. Environ.*, vol. 255, Mar. 2021, Art. no. 112225.

[44] E. Hüllermeier, "Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures?" in *Proc. Uncertainty Artif. Intell.*, Jan. 2022, pp. 2282–2292.

[45] X. X. Zhu et al., "So2Sat LCZ42: A benchmark data set for the classification of global local climate zones [Software and data sets]," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 3, pp. 76–89, Sep. 2020.

[46] I. D. Stewart and T. R. Oke, "Local climate zones for urban temperature studies," *Bull. Amer. Meteorological Soc.*, vol. 93, no. 12, pp. 1879–1900, Dec. 2012.

[47] G. Celeux, D. Chauveau, and J. Diebolt, "Stochastic versions of the em algorithm: An experimental study in the mixture case," *J. Stat. Comput. Simul.*, vol. 55, no. 4, pp. 287–314, Nov. 1996.

[48] C. Qiu, X. Tong, M. Schmitt, B. Bechtel, and X. X. Zhu, "Multilevel feature fusion-based CNN for local climate zone classification from sentinel-2 images: Benchmark results on the So2Sat LCZ42 dataset," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2793–2806, 2020.

[49] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[50] J. Nixon et al., "Measuring calibration in deep learning," in *Proc. CVPR workshops*, Jun. 2019, pp. 1–9. [Online]. Available:

[51] B. Charpentier, D. Zügner, and S. Günnemann, "Posterior network: Uncertainty estimation without OOD samples via density-based pseudo-counts," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 1356–1367.

[52] G. Shafer, *A Mathematical Theory of Evidence*, vol. 42. Princeton, NJ, USA: Princeton Univ. Press, 1976.

[53] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

[54] C. Gruber, K. Hechinger, M. Aßenmacher, G. Kauermann, and B. Plank, "More labels or cases? Assessing label variation in natural language inference," in *Proc. 3rd Workshop Understand. Implicit Underspecified Lang.*, 2024, pp. 22–32.

[55] L. Schmarje et al., "Is one annotation enough?—A data-centric image classification benchmark for noisy and ambiguous label estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 33215–33232.