Uncertainty-Guided Edge Learning for Deep Image Regression in Remote Sensing

Anonymous CVPR submission

Paper ID 3809

Abstract

Edge learning refers to training machine learning models deployed on edge platforms, typically using new data accumulated onboard. The computational limitations on edge devices affect not only model optimisation, but also calculation of the predictive uncertainty of the current model on the unlabelled data, which is vital for informing model updating. In this paper, we investigate edge learning in the context of performing deep image regression on a remote sensing satellite, where a deep network is executed by an onboard computer to regress a scalar y from an input image, e.g., *y is the percentage of pixels indicating cloud coverage or land use. We propose an uncertainty-guided edge learning (UGEL) algorithm that can accurately prioritise the data to speed up training convergence of the onboard regression model. Underpinning UGEL is the calculation of predictive uncertainty based on deep beta regression, where a deep network is used to estimate the parameters of a beta distribution for which the target y for an input image has a high likelihood. Compared to established methods for uncertainty estimation that are either too costly on edge devices (e.g., require many forward passes per sample) or make strict assumptions on the predictive distribution (e.g., Gaussian), deep beta regression is computable in a single forward pass and allows more general predictive distributions. Results show that UGEL delivers faster-converging edge learning than active or semi-supervised learning*[1].

1. Introduction

Deploying machine learning models on the edge promises the benefits of lower latency operation, reduced data transmission costs, and enhanced privacy [9, 11, 32, 51]. However, a deployed model invariably needs to be updated to maintain its accuracy. Edge learning (EL) aims to train a machine learning model on the edge [16, 26, 56]. A major theme is performing the update using data accumulated on the platform itself, since such data reflects the actual operating environment.

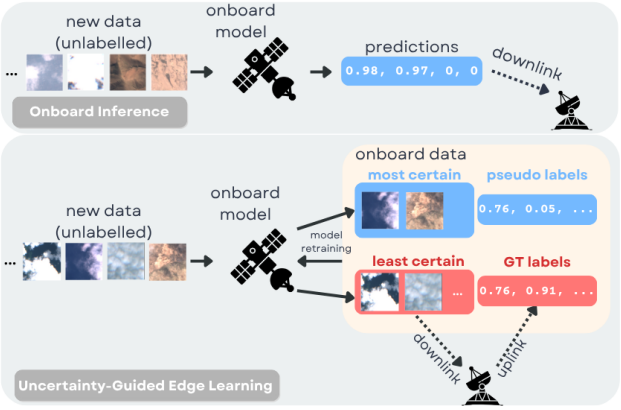

Figure 1. (Top) Onboard inference for responsive RS missions. (Bottom) The proposed UGEL for RS missions.

Progress in space-hardening neural network accelerators for use on orbit has opened up the potential of satellite-borne machine learning [18, 40]. In particular, in remote sensing (RS) missions, by performing inference on Earth imagery collected by onboard imagers and using the prediction for onboard decision making, the time and energy costs of downlinking the data are avoided, thus enabling more responsive missions; see Fig. 1. To motivate using examples, let $f_{\hat{\mathbf{w}}}$ be a deep network with weights $\hat{\mathbf{w}}$, and $\hat{y} = f_{\hat{\mathbf{w}}}(x)$ be the onboard evaluation of $f_{\hat{\mathbf{w}}}$ on an Earth image x. If $\hat{y}$ is the percentage of pixels that are cloud-occluded, x can be discarded if $\hat{y}$ is high, thus saving onboard storage [22]. If $\hat{y}$ is the percentage of pixels indicating land cover, timely survey of environmental impact can be obtained without needing to downlink x for further processing [19].

The need for EL on RS missions arises from the domain gap between the testing data $\mathcal{I} = \{x_i\}_{i=1}^N$ collected onboard and the dataset used to train $f_{\hat{\mathbf{w}}}$ due to inherent sensor discrepancies and drift [17]. Thus, $f_{\hat{\mathbf{w}}}$ needs to be periodically updated to ensure accuracy in long-term operation. Despite the increasing potential for satellite-borne

[1] Source code will be released after the review period.

neural network accelerators, the compute capability will likely be much lower than that of terrestrial versions. Moreover, satellite platforms have stringent power constraints. This demands the update on $f_{\hat{\mathbf{w}}}$ to be computational- and energy-efficient, which invariably translates into using only a small subset $\mathcal{T} \subset \mathcal{I}$ of the unlabelled data to retrain $f_{\hat{\mathbf{w}}}$.

Choosing the subset $\mathcal{T}$ for model retraining and extracting supervisory signals from the unlabelled data are core problems in EL [26]. These tasks are related to active learning (AL) [39], which optimises data subset selection for human labelling, and semi-supervised learning (SSL) [53], which leverages pre-trained models to assign labels to new data. Underpinning the selection of $\mathcal{T}$ is the estimation of the predictive uncertainty of $f_{\hat{\mathbf{w}}}$ on $\mathcal{I}$ [33]. However, if $f_{\hat{\mathbf{w}}}$ has not adapted to the operating environment, balancing the contributions of AL and SSL to selecting $\mathcal{T}$ is difficult.

Moreover, since $|\mathcal{I}| \gg |\mathcal{T}|$, the cost of uncertainty estimation can actually dominate that of retraining $f_{\hat{\mathbf{w}}}$ on $\mathcal{T}$, and an expensive uncertainty estimation technique can exacerbate the computational challenges to EL. For example, the established Monte Carlo (MC) dropout method [20] requires multiple forward passes (*e.g.*, 10 to 30) per sample to collect posterior statistics, which can quickly drain the energy of the edge platform.

By adopting a non-Bayesian approach and placing evidential priors over the Gaussian likelihood function, the seminal deep evidential regression (DER) method [3] directly predicts the hyperparameters of the evidential distribution and avoids multiple forward passes during inference. However, Gaussian predictive uncertainties are unsuitable for some applications, *e.g.*, in the RS tasks of cloud and forest coverage prediction, y is bounded in $[0, 1]$ whereas a Gaussian distribution is unbounded. Applying DER runs the risk of biased uncertainties, and since the accuracy of uncertainty estimation directly influences the usefulness of the selected $\mathcal{T}$ for updating $f_{\hat{\mathbf{w}}}$, the convergence of EL could be impacted. In short, existing uncertainty estimation methods are unsuitable for the task at hand.

To summarise, our key contributions are:

- We propose uncertainty-guided edge learning (UGEL) that unites AL and SSL in a principled manner through a common uncertainty estimation step. We will show clear evidence that combining human labels and pseudo-labels on small subsets of $\mathcal{T}$ chosen in UGEL allows the updating of $f_{\hat{\mathbf{w}}}$ to converge much more rapidly than AL, SSL, or a simple combination of AL and SSL [41].
- We propose the usage of deep beta regression (DBR), where a deep network predicts the beta distribution for which the target y for an input image x has a high likelihood, as the uncertainty estimator for UGEL. This permits more flexible predictive distributions and respects the target bound $y \in [0, 1]$. Also crucially, the differential entropy of beta distribution is a theoretically justified uncertainty estimate which can be computed in one forward pass, which is attractive for edge application.
- We demonstrate that the proposed framework consistently outperforms strong baselines across multiple image regression tasks with various modern lightweight backbones through extensive and rigorous experiments.

2. Related works

The need to update machine learning models deployed on the edge motivates EL [16, 26, 56]. In addition to computational efficiency, major considerations in EL include model simplification, collaborative learning, and learning from partially labelled or unlabelled data—we focus on the latter in this paper. Techniques for EL from unlabelled data include unsupervised learning, self-supervised learning, transfer learning, AL and SSL, the latter two being the most relevant for our work. In the following, we will survey the relevant developments, including the core topic of uncertainty estimation.

2.1. Active learning

The general aim of AL [30, 39, 44, 48] is selecting a small subset of $\mathcal{I}$ for human (groundtruth) annotation, which is then used to retrain $f_{\hat{\mathbf{w}}}$. Finding the most appropriate subset under the manual labelling and data transfer budget (in settings where the subset needs to be transferred off board for human labelling) is a key AL problem. Broadly speaking, there are three strategies for AL: uncertainty-based [21, 42, 52], where samples with low prediction confidence by the current model are prioritised, diversity-based [43, 47], where samples that are representative of the underlying data distribution are selected, and hybrid techniques [4, 28], that combine the above approaches.

While clear benefits to non-deep learning methods have been recorded [44], the utility of AL on deep learning is less obvious [36]. It has been conjectured that the small number of samples selected, which is generally of the same size as training batches, dilutes the effectiveness of AL for deep learning [49].

2.2. Semi-supervised learning

SSL focusses on settings where we have a limited number of labelled (groundtruth) samples but a large number of potentially useful unlabelled data [53]. Broadly the SSL loss combines supervised loss, unsupervised loss and relevant regularisation terms. Major paradigms include generative methods [14, 15, 46], where the data distribution is approximated before the joint data-label distribution is evaluated on the new samples, graph-based methods [2, 23, 55], where a neighbourhood structure is estimated for label propagation, consistency regularisation methods [6, 7, 45], which assume model predictions remain unchanged under realistic perturbations to data or model weights, and pseudo-labelling

methods [10, 29, 37], which generate labels for the new data based on one or more models trained on the labelled subset.

In EL settings where there is domain gap or shift, SSL should not rely on groundtruth samples acquired prior to deployment, since this will lead to bias in the decisions on the new unlabelled samples. We argue that the groundtruth samples should be extracted from the unlabelled samples directly, specifically using AL to request manual annotations on the appropriate subset of samples.

2.3. Active semi-supervised learning

Active semi-supervised learning (ASSL) combines AL and SSL to alleviate their respective shortcomings. However, previous works on ASSL [12, 31] assume all data to be located on a monolithic platform (*e.g.*, a data centre) without restrictions on the computational and communication budget inherent to EL. More fundamentally, previous ASSL works combine AL and SSL simplistically, whereby AL is used to procure the small set of groundtruth samples for SSL.

We argue that more principled ways to conduct ASSL could lead to further benefits, *e.g.*, cross leveraging the intermediate results of AL and SSL to improve accuracy and/or achieve computational savings. To this end, Sec. 3 will describe our proposed UGEL algorithm.

2.4. Uncertainty estimation

As mentioned above, uncertainty estimation is a key step in AL. The main approaches to uncertainty estimation are Bayesian inference, ensemble methods, and deterministic methods [50]. Due to the resource constraints of edge devices, ensemble methods—which require storing multiple models, performing several inferences per input, and updating all models in the ensemble when necessary—are not suitable and hence will not be considered.

Given training data $\mathcal{K} = \{(x_k, y_k)\}_{k=1}^{K}$, Bayesian inference defines the predictive distribution on new sample x as

$$p(y \mid x, \mathcal{K}) = \int_{\mathbf{w}} p(y \mid x, \mathbf{w}) p(\mathbf{w} \mid \mathcal{K}) d\mathbf{w}, \qquad (1)$$

where $p(\mathbf{w} \mid \mathcal{K})$ is the posterior of model weights $\mathbf{w}$, and $p(y \mid x, \mathbf{w})$ is the likelihood. MC dropout [20] approximates (1) efficiently by mimicking a sample from the posterior distribution: first, a deep network $f_{\hat{\mathbf{w}}}$ is trained on $\mathcal{K}$ using dropout, where $\hat{\mathbf{w}}$ are the learned weights. Then, P forward passes of x through $f_{\hat{\mathbf{w}}}$ are conducted, with stochastic dropouts on $\hat{\mathbf{w}}$ in each pass, yielding P estimates $\{\hat{y}^{(p)}\}_{p=1}^{P}$. The mean and variance of (1) are then obtained respectively as

$$\hat{y} = \frac{1}{P}\sum_{p=1}^{P} \hat{y}^{(p)}, \quad \hat{\sigma}^2 = \frac{1}{P}\sum_{p=1}^{P} (\hat{y}^{(p)} - \hat{y})^2, \qquad (2)$$

with $\hat{y}$ and $\hat{\sigma}$ interpreted as the prediction at x and its uncertainty. Since P evaluations of $f_{\hat{\mathbf{w}}}$ are required, MC dropout can be costly on edge devices if P is large, *e.g.*, $P = 30$ as suggested by Gal and Ghahramani [20].

DER [3], which is representative of deterministic methods, jointly predicts the regression target and its corresponding evidence to estimate predictive uncertainty. Under the assumption that regression targets follows a Gaussian distribution of unknown mean $\mu \sim \mathcal{N}(\gamma, \sigma^2 \upsilon^{-1})$ and unknown variance $\sigma \sim \Gamma^{-1}(\alpha, \beta)$, DER optimises the model to infer the parameters $(\gamma, \upsilon, \alpha, \beta)$ of the evidential distribution given input x, which are then used to estimate the aleatoric and epistemic uncertainties

$$h_a = \mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}, \quad h_e = Var[\mu] = \frac{\beta}{\upsilon(\alpha - 1)}. \qquad (3)$$

The sum $h_a + h_e$ is used as the predictive uncertainty on x [1]. As we will show later, using DER in selecting $\mathcal{T}$ in our UGEL method did not lead to consistently better performance than random sampling.

3. Uncertainty-guided edge learning

Let $f_{\hat{\mathbf{w}}}$ be the deep regression model deployed onboard that was pre-trained with an *initial* labelled dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^{M}$, where $y_j \in [0, 1]$. Details on training $f_{\hat{\mathbf{w}}}$ will be given in Sec. 4. Our goal is updating $f_{\hat{\mathbf{w}}}$ using the unlabelled data $\mathcal{I} = \{x_i\}_{i=1}^{N}$ collected onboard, where $N \gg M$.

The proposed UGEL algorithm for updating $f_{\hat{\mathbf{w}}}$ using $\mathcal{I}$ iteratively selects a subset $\mathcal{T} \subset \mathcal{I}$, labels the subset, and updates $f_{\hat{\mathbf{w}}}$ across a number of rounds—a single iteration of UGEL is defined in Alg. 1. To facilitate labelling, a *twin model* $f_{\mathring{\mathbf{w}}}$ is also maintained on the edge. Both $f_{\hat{\mathbf{w}}}$ and $f_{\mathring{\mathbf{w}}}$ have the same architecture and were pre-trained on the same data $\mathcal{D}$, but initialised differently, thus $\hat{\mathbf{w}} \neq \mathring{\mathbf{w}}$. Details of the main steps in Alg. 1 are given in the following.

3.1. Subset selection

Let $h_{\hat{\mathbf{w}}}(x_i)$ be the estimated measure of predictive uncertainty of $f_{\hat{\mathbf{w}}}$ on x_i. Since UGEL is agnostic to the uncertainty estimation method, we will leave the description of the proposed DBR method to Sec. 4.

In each round of UGEL, $\mathcal{T} = \mathcal{T}_U \cup \mathcal{T}_C$ is composed from

- A subset $\mathcal{T}_U \subset \mathcal{I}$ of size B_U on which $f_{\hat{\mathbf{w}}}$ is the *most uncertain*, *i.e.*,

$$\mathcal{T}_U = \underset{\mathcal{S} \subset \mathcal{I},\ |\mathcal{S}| = B_U}{\arg\max} \sum_{x_i \in \mathcal{S}} h_{\hat{\mathbf{w}}}(x_i). \qquad (4)$$

- A subset $\mathcal{T}_C \subset \mathcal{I}$ of size B_C on which $f_{\hat{\mathbf{w}}}$ is the *most certain*, *i.e.*,

$$\mathcal{T}_C = \underset{\mathcal{S} \subset \mathcal{I},\ |\mathcal{S}| = B_C}{\arg\min} \sum_{x_i \in \mathcal{S}} h_{\hat{\mathbf{w}}}(x_i). \qquad (5)$$

By construction, $\mathcal{T}_U \cap \mathcal{T}_C = \emptyset$, hence $|\mathcal{T}| = B_U + B_C$.

Algorithm 1 UGEL (single round)

Inputs:
$f_{\hat{\mathbf{w}}}$: regression model
$h_{\hat{\mathbf{w}}}$: uncertainty estimator
$f_{\mathring{\mathbf{w}}}$: twin model
$\mathcal{D} = \{(x_j, y_j)\}_{j=1}^{M} := (\mathcal{X}, \mathcal{Y})$: init. labelled dataset
$\mathcal{I} = \{x_i\}_{i=1}^{N}$: unlabelled data (typ. $N \gg M$)
B_U: #samples for human annot. (req. $B_U \leq N$)
B_C: #samples for model annot. (req. $B_C \leq N - B_U$)
τ: loss coefficient
▷ **Select data subset**
$\mathcal{T}_U \leftarrow \underset{\mathcal{Z} \subset \mathcal{I},\ |\mathcal{Z}| = B_U}{\arg\max} \sum_{x_i \in \mathcal{Z}} h_{\hat{\mathbf{w}}}(x_i)$
$\mathcal{T}_C \leftarrow \underset{\mathcal{Z} \subset \mathcal{I},\ |\mathcal{Z}| = B_C}{\arg\min} \sum_{x_i \in \mathcal{Z}} h_{\hat{\mathbf{w}}}(x_i)$
▷ **Label data subset**
$\mathcal{Y}_U \leftarrow$ manually annotate $\mathcal{T}_U$
$\hat{\mathcal{Y}}_C \leftarrow f_{\hat{\mathbf{w}}}(\mathcal{T}_C)$ // Pseudo-labels
$\mathring{\mathcal{Y}}_C \leftarrow f_{\mathring{\mathbf{w}}}(\mathcal{T}_C)$ // Pseudo-labels
▷ **Retrain model(s)**
$(\mathcal{X}, \mathcal{Y}) \leftarrow (\mathcal{X}, \mathcal{Y}) \cup (\mathcal{T}_U, \mathcal{Y}_U)$
$\hat{\mathcal{Y}} \leftarrow f_{\hat{\mathbf{w}}}(\mathcal{X})$ // Prediction
$\mathring{\mathcal{Y}} \leftarrow f_{\mathring{\mathbf{w}}}(\mathcal{X})$ // Prediction
$L \leftarrow \mathcal{L}_s(\hat{\mathcal{Y}}, \mathcal{Y}) + \mathcal{L}_s(\mathring{\mathcal{Y}}, \mathcal{Y}) + \tau \mathcal{L}_{RMSE}(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}_C, \mathring{\mathcal{Y}} \cup \mathring{\mathcal{Y}}_C)$
$f_{\hat{\mathbf{w}}} \leftarrow$ model_update$(f_{\hat{\mathbf{w}}}, L)$
$f_{\mathring{\mathbf{w}}} \leftarrow$ model_update$(f_{\mathring{\mathbf{w}}}, L)$
Outputs:
$f_{\hat{\mathbf{w}}}$: updated regression model
$f_{\mathring{\mathbf{w}}}$: updated twin model
$\mathcal{D} \leftarrow (\mathcal{X}, \mathcal{Y})$: expanded labelled dataset
$\mathcal{I} \leftarrow \mathcal{I} \setminus \mathcal{T}_U$: reduced unlabelled data

3.2. Label assignment

Labelling of $\mathcal{T}$ is achieved as follows:

- $\mathcal{T}_U$ is transferred off board for manual annotation, yielding the label set $\mathcal{Y}_U$. Therefore, size B_U should reflect the annotation budget, which depends on the available human effort and communication bandwidth.
- $\mathcal{T}_C$ is labelled by evaluating $f_{\hat{\mathbf{w}}}$ and $f_{\mathring{\mathbf{w}}}$ to yield the label sets $\hat{\mathcal{Y}}_C$ and $\mathring{\mathcal{Y}}_C$. Since $\mathcal{T}_C$ has no groundtruth labels, $\hat{\mathcal{Y}}_C$ and $\mathring{\mathcal{Y}}_C$ play the role of *pseudo-labels*.

Note that in the context of satellite-borne machine learning, $\mathcal{T}_U$ needs to be downlinked with the labels $\mathcal{Y}_U$ subsequently uplinked. Hence B_U should be small (more details below).

3.3. Model retraining

The loss function L is a weighted combination of supervised loss $\mathcal{L}_s$ and semi-supervised loss $\mathcal{L}_{RMSE}$ (precise definitions of the losses will be given in Sec. 4). $\mathcal{L}_s$ depends on the human-labelled samples only (including $(\mathcal{T}_U, \mathcal{Y}_U)$) while $\mathcal{L}_{RMSE}$ measures the inconsistency of the prediction of $f_{\hat{\mathbf{w}}}$ and $f_{\mathring{\mathbf{w}}}$ on the data considered thus far (including the pseudo-labelled samples). Intuitively, minimising $\mathcal{L}_{RMSE}$ induces cross supervision [13] between the primary and twin models.

The overall loss L is then used to update both $f_{\hat{\mathbf{w}}}$ and $f_{\mathring{\mathbf{w}}}$. Then, $(\mathcal{T}_U, \mathcal{Y}_U)$ is appended to $\mathcal{D}$, $\mathcal{T}_U$ is removed from $\mathcal{I}$, and UGEL can proceed to the next round if $|\mathcal{I}| \geq B_U$.

3.4. Intuition behind UGEL

UGEL allows to balance the contributions of human- and pseudo-labelled data, both chosen carefully using an uncertainty estimator, to speed up the training convergence. Executing UGEL sequentially will progressively expand $\mathcal{D}$ and reduce $\mathcal{I}$. However, since $\mathcal{D}$ is initially small (*e.g.*, $M = 12$ in typical online learning settings) and $\mathcal{T}_U$ is kept small to minimise human labelling effort and data transfer load (*e.g.*, $B_U = 6$ in our experiments), $f_{\hat{\mathbf{w}}}$, $f_{\mathring{\mathbf{w}}}$ and $h_{\hat{\mathbf{w}}}$ tend to be inaccurate and uninformative initially. Using all pseudo-labelled samples from $\mathcal{I}$, which dwarfs $\mathcal{D}$ (*e.g.*, size N of $\mathcal{I}$ can be $10k$), will lead to divergence in training. The size of $\mathcal{T}_C$ should thus reflect the size of $\mathcal{D}$. To this end, we use $B_C = |\mathcal{D}|$, *i.e.*, $\mathcal{T}_C$ is small initially and grows with $\mathcal{D}$.

Following from the above, we can also see that UGEL bridges AL, SSL and ASSL through the parameter settings:

- If $B_U > 0$ and $B_C = 0$, UGEL reduces to uncertainty-based AL [21, 42].
- If $B_U = 0$ and $B_C = N$, UGEL can be seen as SSL with cross pseudo supervision [13] (in this setting, UGEL will terminate after one round).
- If $B_U > 0$ and $B_C = N - B_U$, UGEL reduces to the ASSL algorithm of Yao et al. [54] that uses all human- and pseudo-labelled samples in the training.

Sec. 6 will show that being able to balance the relative contributions of human- and pseudo-labelled samples allows UGEL to converge much faster than AL, SSL and ASSL.

3.5. Computational considerations

In each round of UGEL, the uncertainty estimator $h_{\hat{\mathbf{w}}}$ needs to be applied on the unlabelled set $\mathcal{I}$ of size N. The samples involved in retraining $f_{\hat{\mathbf{w}}}$ and $f_{\mathring{\mathbf{w}}}$ are $\mathcal{D}$, $\mathcal{T}_U$ and $\mathcal{T}_C$, which are of sizes M, B_U and B_C respectively. Since N is much larger than M and B_U, the total cost of uncertainty estimations in each round dominates that of retraining the models. This argues for a computationally efficient $h_{\hat{\mathbf{w}}}$.

4. Deep beta regression

We focus on regressing a scalar $y \in [0, 1]$ from an input image x due its relevance to RS applications (see Sec. 1).

Our method DBR trains a deep network $f_{\mathbf{w}}$ that outputs the parameters of a beta distribution given x. The probability density function of a beta distribution has a support on

$[0,1]$ and two shape parameters $\alpha > 0$ and $\beta > 0$, *i.e.*,

$$p(y \mid \alpha, \beta) = \frac{1}{Beta(\alpha,\beta)} y^{\alpha-1}(1-y)^{\beta-1}, \quad y \in [0,1], \tag{6}$$

where $Beta(\alpha,\beta)$ is the beta function. We use an alternative parameterisation via the mean $\mu = \alpha/(\alpha+\beta) \in (0,1)$ and the precision $\nu = \alpha + \beta > 0$, leading to

$$p(y|\mu,\nu) = \frac{\Gamma(\nu)}{\Gamma(\mu\nu)\Gamma\big((1-\mu)\nu\big)} y^{\mu\nu-1}(1-y)^{(1-\mu)\nu-1}, \tag{7}$$

where $\Gamma(\cdot)$ is the gamma function. Hence the output range of $f_{\mathbf{w}}$ is $(0,1) \times \mathbb{R}_+$. Given the prediction $(\hat{\mu}, \hat{\nu}) = f_{\mathbf{w}}(x)$, we take $\hat{\mu}$ directly as the scalar prediction $\hat{y}$.

4.1. Predictive uncertainty

We can directly calculate the *differential entropy* [34], which measures the predictive uncertainty of the fitted beta distribution under mean-precision parameterisation as

$$\begin{aligned} h_{\mathbf{w}}(x) = \log \frac{\Gamma(\hat{\mu}\hat{\nu})\Gamma\big((1-\hat{\mu})\hat{\nu}\big)}{\Gamma(\hat{\nu})} \\ +(\hat{\nu}-2)\psi(\hat{\nu}) - (\hat{\mu}\hat{\nu}-1)\psi(\hat{\mu}\hat{\nu}) \\ -\big((1-\hat{\mu})\hat{\nu}-1\big)\psi\big((1-\hat{\mu})\hat{\nu}\big), \end{aligned} \tag{8}$$

where $\psi(\cdot)$ is the digamma function. Note that the range of (8) is $[-\infty, 0]$. A higher entropy indicates that the predicted beta distribution has a shape closer to the uniform distribution, revealing the model is uncertain about the value of $\hat{y}$ in $[0,1]$. Conversely, a lower entropy arises from a more concentrated density, indicating higher certainty.

In contrast to MC dropout, computing $h_{\mathbf{w}}(x)$ in DBR depends on values $(\hat{\mu}, \hat{\nu})$ obtained from a single forward pass of $f_{\mathbf{w}}$. In contrast to DER, the predictive density respects the range $[0,1]$. Fig. 2 provides qualitative comparisons between the uncertainty estimation methods. As will be presented in Sec. 6, the groundtruth targets frequently occupy the extremes (0 and 1) in real image regression datasets, thus, handling cases at the boundaries is vital.

4.2. Architecture and training

Let f be the deep neural network employed in this study. Given labelled dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^M := (\mathcal{X}, \mathcal{Y})$, the training of $f_{\mathbf{w}}$ is formulated as multitask optimisation, in which both the standard regression error (root mean-squared error) and the negative log-likelihood are minimised simultaneously. Let $(\hat{\mathcal{U}}, \hat{\mathcal{V}}) = f_{\mathbf{w}}(\mathcal{X})$, where $\hat{\mathcal{U}} = \{\hat{\mu}_j\}_{j=1}^M$ and $\hat{\mathcal{V}} = \{\hat{\nu}_j\}_{j=1}^M$. The loss is

$$\mathcal{L}_s = \mathcal{L}_{RMSE}(\hat{\mathcal{U}}, \mathcal{Y}) + \lambda \mathcal{L}_{NLL}(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \mathcal{Y}), \tag{9}$$

where λ weighs the relative importance of the two tasks,

$$\mathcal{L}_{RMSE}(\hat{\mathcal{U}}, \mathcal{Y}) = \sqrt{\sum_{j=1}^{M} \frac{(\hat{\mu}_j - y_j)^2}{M}} \tag{10}$$

evaluates the regression error, while minimising

$$\begin{aligned} \mathcal{L}_{NLL}(\hat{\mathcal{U}}, \hat{\mathcal{V}}, \mathcal{Y}) = \frac{1}{M}\sum_{j=1}^{M} \log \frac{\Gamma(\hat{\mu}_j\hat{\nu}_j)\Gamma\big((1-\hat{\mu}_j)\hat{\nu}_j\big)}{\Gamma(\hat{\nu}_j)} \\ -(\hat{\mu}_j\hat{\nu}_j - 1)\log(y_j) - \big((1-\hat{\mu}_j)\hat{\nu}_j - 1\big)\log(1-y_j) \end{aligned} \tag{11}$$

fits the predictive beta density for uncertainty estimation. Refer to the supp material for details on the derivations.

5. Datasets

We first describe the RS datasets used in the experiments.

5.1. Real scenes with real clouds from Landsat-8 (RSRC-L8)

We used real RGB images from the 38-Cloud dataset [35] which contained groundtruth cloud masks. We selected 18 scenes for EL and the remaining 20 scenes were used as a testing data from unseen before scenes. We cropped all scenes and their ground-truth masks into 128x128 non-overlapping patches, and the cloud coverage percentage of each patch was obtained by counting the foreground pixels in the masks. In total, we have $20,294$ patches for EL, and $22,528$ for testing; see Fig. 3.

5.2. Real scenes with real clouds from Sentinel-2 (RSRC-S2)

We sourced real base images from CloudSEN12 [5], a large-scale expert-labelled cloud segmentation dataset built upon ESA's Sentinel-2 L1C imagery. Due to the large size of the corpus (nearly 1 TB), we selected $5,000$ based images (about 10%) for EL and $2,000$ based images for testing. As above, we cropped each based image into 128x128 patches and used bands 4-3-2, which can be processed to RGB images. The ground-truth masks of CloudSEN12 had 4 classes: 0-clear, 1-thick cloud, 2-thin cloud, 3-cloud shadow. We grouped classes 0 & 3 as non-cloud and classes 1 & 2 as cloud before calculating the groundtruth cloud percentage. In total, we have $80,000$ patches for EL and $32,000$ patches for testing.

5.3. Real scenes with land coverage (RSLC)

We sourced real RGB images for land cover segmentation from LandCover.ai [8]. The background class (uncovered) relates to open land (*e.g.*, grass, fields) while the rest (*e.g.*, building, woodland, water and road) are subsumed under the foreground class (covered). Based on this assignment,

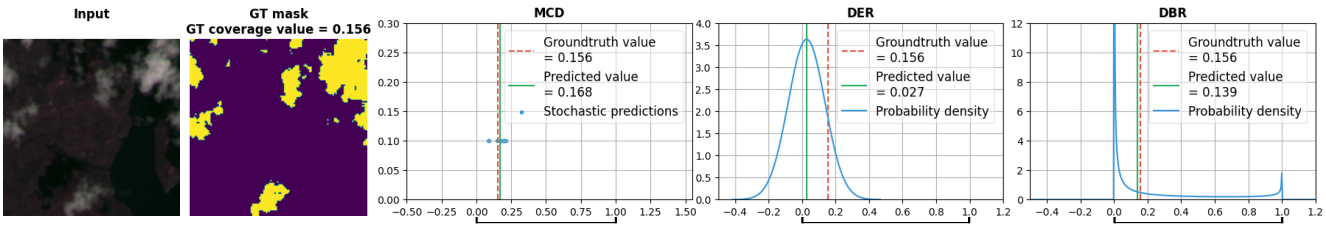

Figure 2. Qualitative comparison of MC Dropout, DER and DBR in uncertainty estimation for cloud coverage prediction. The first two panels show the input image x and the foreground mask corresponding to the groundtruth coverage value y. The remaining panels show the three uncertainty estimation methods. Note that the predictive distribution of DER (a Gaussian) extends beyond the target range $[0, 1]$, while that of DBR (a Beta) lies within $[0, 1]$.

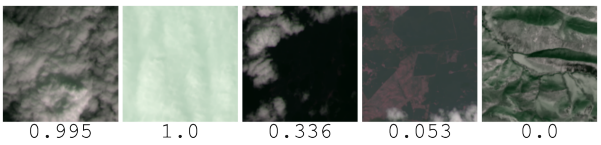

Figure 3. Sample images from `RSRC-L8` with reg. targets.

the percentage of pixels corresponding to land cover were computed. As above, 128x128 pixel patches were extracted, resulting in $154,056$ patches for EL and $25,279$ patches for testing; see Fig. 4 for sample images.

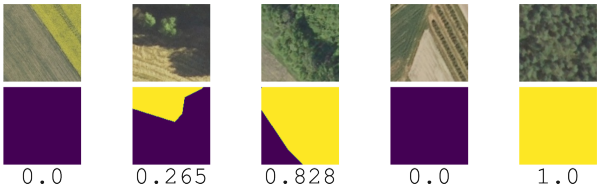

Figure 4. Sample images from `RSLC` with regression targets.

5.4. Dataset statistics

Fig. 5 plots the histograms of the groundtruth regression labels y in the three datasets above. The histograms are bimodal, with the two extremes $y = 0$ and $y = 1$ being much more frequent, though non-trivial number of samples span the range $[0, 1]$. Crucially, the fact that many groundtruth labels occupy the extremes argues for uncertainty estimation that handles boundaries correctly.

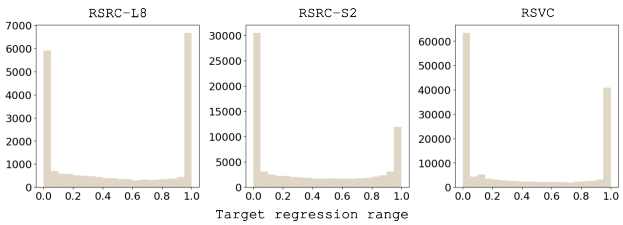

Figure 5. Histogram of groundtruth regression labels.

6. Experiments

6.1. Benchmarking on RS applications

We benchmarked UGEL against representative methods for model learning/updating in image-based RS applications. The specific pipelines compared were:

- Chen et al. [13]'s SSL algorithm based on cross pseudo supervision for image semantic segmentation that we modified for image regression (predicting scalar y).
- Yao et al. [54]'s ASSL algorithm that combines AL and SSL for cloud segmentation, which we also modified for image regression.
- The proposed UGEL with DBR uncertainty estimation. The hyperparameters used for each round of UGEL were $B_U = 6$, $B_C = |\mathcal{D}|$ and $\tau = 2$.

For completeness, we also compared against

- Using BALD [21] to train an image regressor. BALD is a seminal AL method that uses MC dropout for uncertainty estimation in the acquisition function that selects $\mathcal{T}_U$.
- Baseline AL method with random sampling as the acquisition function. Munjal et al. [36] demonstrated that random sampling performs competitively with most AL methods in fair experimental settings, validating its role as a bona fide AL baseline.

For the methods that require human labelling of $\mathcal{T}_U$, we take the groundtruth labels of $\mathcal{T}_U$ as the human labels.

We adopted the standard batch-mode active learning setup [4], initialising with $M = 12$ labelled samples and using a batch size of $b = 6$ to respect the communication limit B_U for annotation. B_U is kept small to reflect the practical constraints of satellite communication. As discussed in Sec. 1, satellites operate under stringent computational and energy limitations, which necessitates the use of lightweight deep neural networks. Accordingly, this work focuses on compact yet effective architectures. All regression models employed widely adopted lightweight backbone architectures (ResNet18 [24], MobileNetv3 [25], or MobileNetv4 [38]) and were trained with the Adam optimizer [27] (learning rate = 0.001). In each round, the models were randomly re-initialised and retrained for 12

epochs. Re-initialisation at the beginning of each round is required, since continuing training from the previous checkpoint degrades model performance due to catastrophic forgetting. Performance was evaluated using RMSE on the test set after each round. Each experiment consisted of as many rounds as feasible on our HPC, and the RMSE values were recorded. To ensure robustness and comparability, experiments were repeated 10 times to obtain mean, std dev and p-values. Results on `RSRC-L8` and `RSLC` are presented in Fig. 6 for models using ResNet18 and MobileNetV3. Results on `RSRC-S2` and for the MobileNetV4 backbone are included in the supp material. The SSL and ASSL experiments were limited to fewer rounds due to computational constraints. Each ASSL round processes all remaining unlabeled samples, greatly increasing computational cost and runtime compared with UGEL with DBR or active learning methods (Rand and BALD). Extending ASSL further would require substantially more training time and resources without providing additional methodological insight

AL with random sampling and BALD reduced the testing RMSE over time, but with large fluctuations, particularly for random sampling. Moreover, BALD did not outperform random sampling; in fact, in `RSLC` random sampling reduced the RMSE faster than BALD. This observation is consistent with [36]. In contrast to the AL methods, UGEL with DBR consistently yielded lower RMSE and standard deviations, indicating that incorporating supervisory signals from the most certain unlabelled instances enhances model convergence and learning stability.

UGEL with DBR substantially outperformed Yao et al. [54]'s ASSL method. While ASSL initially provided noticeable improvement over using AL alone, its performance plateaued thereafter, likely due to a large number of noisy pseudo-labels diluting the human-labelled samples. In contrast, UGEL sustained the benefit of SSL by leveraging only the most certain pseudo-labels, resulting in consistent performance gains. This highlights the importance of selectively integrating pseudo-labels guided by uncertainty.

Note that the gap between all methods reduce with the number of rounds, since asymptotically all the new data are eventually selected for labelling. The key differentiating factor for EL is reducing the testing error using the least number of samples to respect labelling and bandwidth limits. Tab. 1 shows p-values from comparing UGEL with the other methods, complementing the results shown in Fig. 6. The performance gaps are meaningful since most of the p-values are < 0.05. Note that outcomes in different number of rounds are shown since the datasets are of distinct sizes, which yielded different convergence speeds.

6.2. UGEL ablation studies

Two ablation tests were conducted:

T1 Effects of diff. uncertainty estimation methods on UGEL.

	Test	Rd. 2	Rd. 4	Rd. 6	Rd. 8
RSRC-S2	UGEL vs Rand	0.019	0.001	0.005	0.002
	UGEL vs BALD	0.042	0.002	0.053	0.002
	UGEL vs SSL	0.014	0.001	0.005	0.001
	UGEL vs ASSL	0.016	0.008	0.008	0.008
	Test	**Rd. 5**	**Rd. 10**	**Rd. 15**	**Rd. 20**
RSRC-L8	UGEL vs Rand	0.024	0.001	0.002	0.009
	UGEL vs BALD	0.036	0.002	0.002	0.001
	UGEL vs SSL	0.001	0.001	0.001	0.001
	UGEL vs ASSL	0.001	0.001	0.001	0.001
	Test	**Rd. 1**	**Rd. 2**	**Rd. 3**	**Rd. 4**
RSLC	UGEL vs Rand	0.188	0.001	0.005	0.003
	UGEL vs BALD	0.002	0.001	0.001	0.001
	UGEL vs SSL	0.839	0.001	0.001	0.001
	UGEL vs ASSL	0.577	0.042	0.024	0.019

Table 1. Wilcoxon signed-rank test p-values comparing UGEL with DBR against other methods on all datasets using the ResNet18 backbone.

T2 Effects of different groundtruth regression label distributions on the performance of UGEL.

For **T1**, in place of DBR, we used the following uncertainty estimation algorithms in UGEL: MC dropout (MCD) with $P = 10$, DER and a "dummy" method (RAN) that assigns constant uncertainty thus amounting to randomly sampling the data. The experimental setup, hyperparameter configurations, and evaluation metric were kept consistent with those described in the preceding benchmarking experiment. The utilised backbone is ResNet18. Results on `RSRC-S2` are in Fig. 7, while those for `RSRC-L8` and `RSLC` are in the supp material.

The results show that using DBR in UGEL outperformed the other uncertainty methods, with a noticeable gap in the earlier rounds. The statistical significance test results in Tab. 2 confirm that using DBR in UGEL achieved significantly lower regression errors than other methods most of the time (p-values < 0.05).

For **T2**, we generated and added synthetic cloud masks on clearsky patches on RS images to produce deep regression datasets with different groundtruth label distributions. UGEL with the different uncertainty estimation methods was then executed on the datasets, and the results show that DBR still provided the best regression accuracy with high statistical significance. Due to page limits, we refer the reader to the supp material for details of this experiment.

6.3. Runtime comparisons

To assess runtime in a satellite-borne EL context, we executed UGEL with different uncertainty methods on an Nvidia Jetson Orin NX, which was identified as viable for satellite operation [18, 40]. The employed backbone is

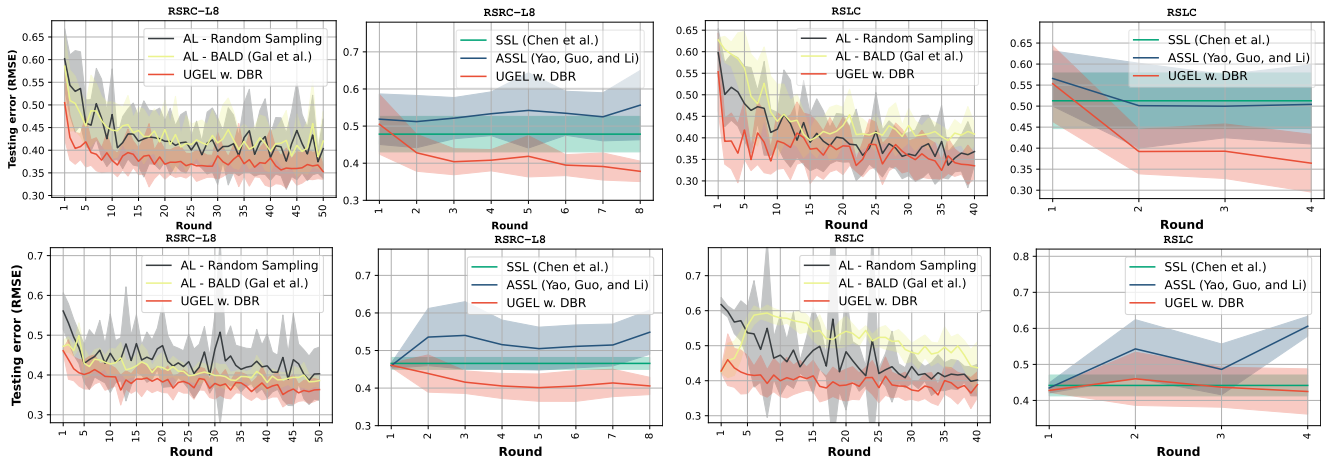

Figure 6. Panels 1–4 (top row): Benchmarking UGEL with DBR against AL (BALD [21] and random sampling), SSL [13] and ASSL [54] on RSRC−L8 and RSLC using a ResNet18 backbone. Panels 5–8 (bottom row): Same comparisons but using MobileNetV3 as the backbone. Bolded curves and shaded regions respectively indicate mean RMSE testing error and standard deviation over 10 runs.

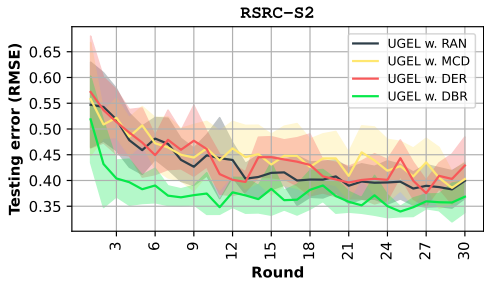

Figure 7. Comparison of different uncertainty estimation methods in UGEL on RSRC−S2 using the ResNet18 backbone. Bolded curves and shaded regions respectively indicate mean RMSE testing error and std dev over 10 runs.

	Test	Rd. 5	Rd. 10	Rd. 15	Rd. 20
RSRC−S2	DBR vs RAN	0.005	0.002	0.116	0.019
	DBR vs MCD	0.001	0.003	0.08	0.003
	DBR vs DER	0.001	0.001	0.005	0.032
RSRC−L8	DBR vs RAN	0.094	0.003	0.003	0.068
	DBR vs MCD	0.003	0.001	0.004	0.003
	DBR vs DER	0.244	0.003	0.011	0.002
RSLC	DBR vs RAN	0.348	0.423	0.216	0.348
	DBR vs MCD	0.042	0.005	0.116	0.042
	DBR vs DER	0.138	0.042	0.065	0.042

Table 2. Wilcoxon signed-rank test p-values comparing using DBR and other methods using the ResNet18 backbone, sampled every 5 rounds.

ResNet18. Fig. 3 records the runtime on RSRC−S2 (see the supp material for the runtimes on RSRC−L8 and RSLC), where it is clear that the computational cost of uncertainty estimation was much greater than that of model retraining, and that MC dropout was costlier than DER and DBR.

Method	Uncertain. est.↓	Model retrain.↓
UGEL w. RAN	n/a	7.05-11.64
UGEL w. MCD	495.84-514.18	7.18-12.54
UGEL w. DER	82.01-85.47	7.23-13.29
UGEL w. DBR	68.04-74.53	7.43-12.02

Table 3. Runtime (minimum and maximum, in seconds) for different uncertainty estimation methods during the first 10 rounds of UGEL on the RSRC−S2 dataset using the ResNet18 backbone.

7. Conclusions and limitations

We proposed an uncertainty-guided edge learning (UGEL) algorithm that was integrated with an efficient uncertainty estimation method based on deep beta regression (DBR). Benchmarking on real remote sensing (RS) datasets demonstrated the superiority of UGEL with DBR over active and semi-supervised learning methods. Ablation studies confirmed that DBR effectively guided the selection of informative unlabeled samples for model updates in remote sensing applications. Overall, the results indicate that the proposed framework is well-suited for accelerating on-board deep network training for RS applications.

The improvement provided by the proposed framework declines with increased object diversity, as shown in the results on RSLC. The use of the twin model in UGEL also increases computational costs, memory usage, and power consumption during training. In the future, we plan to develop a more adaptive uncertainty estimation method for such conditions and a more efficient semi-supervised learning method without employing a twin model.

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021. 3

[2] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR, 2019. 2

[3] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020. 2, 3

[4] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019. 2, 6

[5] Cesar Aybar, Luis Ysuhuaylas, Jhomira Loja, Karen Gonzales, Fernando Herrera, Lesly Bautista, Roy Yali, Angie Flores, Lissette Diaz, Nicole Cuenca, et al. Cloudsen12, a global dataset for semantic understanding of cloud and cloud shadow in sentinel-2. *Scientific data*, 9(1):782, 2022. 5

[6] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 2

[7] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 2

[8] Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Tomasz Dziedzic, and Anna Zambrzycka. Landcover. ai: Dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1102–1110, 2021. 5

[9] Han Cai, Ji Lin, Yujun Lin, Zhijian Liu, Haotian Tang, Hanrui Wang, Ligeng Zhu, and Song Han. Enable deep learning on mobile devices: Methods, systems, and applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 27(3):1–50, 2022. 1

[10] Dong-Dong Chen, Wei Wang, and Wei Gao Zhihua Zhou. Tri-net for semi-supervised deep learning. In *Proceedings of twenty-seventh international joint conference on artificial intelligence*, pages 2014–2020, 2018. 3

[11] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019. 1

[12] Sijin Chen, Yingyun Yang, and Yan Hua. Semi-supervised active learning for object detection. *Electronics*, 12(2):375, 2023. 3

[13] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2613–2622, 2021. 4, 6, 8

[14] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017. 2

[15] Remi Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016. 2

[16] Sauptik Dhar, Junyao Guo, Jiayi Liu, Samarth Tripathi, Unmesh Kurup, and Mohak Shah. A survey of on-device machine learning: An algorithms and learning theory perspective. *ACM Transactions on Internet of Things*, 2(3):1–49, 2021. 1, 2

[17] Andrew Du, Anh-Dzung Doan, Yee Wei Law, and Tat-Jun Chin. Domain adaptation for satellite-borne hyperspectral cloud detection. *arXiv preprint arXiv:2309.02150*, 2023. 1

[18] Michael A. Felix, Windy S. Slater, Derrek C. Landauer, Ryan E. Pinson, and Benjamin B.W. Rutherford. Total ionizing dose radiation testing of nvidia jetson orin nx system on module. In *2024 IEEE Space Computing Conference (SCC)*, pages 116–121, 2024. 1, 7

[19] Matt Finer, Sidney Novoa, Mikaela J Weisse, Rachael Petersen, Joseph Mascaro, Tamia Souto, Forest Stearns, and Raúl García Martinez. Combating deforestation: From satellite to intervention. *Science*, 360(6395):1303–1305, 2018. 1

[20] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 2, 3

[21] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR, 2017. 2, 4, 6, 8

[22] Gianluca Giuffrida, Lorenzo Diana, Francesco De Gioia, Gionata Benelli, Gabriele Meoni, Massimiliano Donati, and Luca Fanucci. Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images. *Remote Sensing*, 12(14):2205, 2020. 1

[23] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[25] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 6

[26] Aymen Rayane Khouas, Mohamed Reda Bouadjenek, Hakim Hacid, and Sunil Aryal. Training machine learning models at the edge: A survey. *arXiv preprint arXiv:2403.02619*, 2024. 1, 2

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[28] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019. 2

[29] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896. Atlanta, 2013. 3

[30] Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. A survey on deep active learning: Recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 2

[31] Wanli Ma, Oktay Karakus, and Paul L Rosin. Integrating semi-supervised and active learning for semantic segmentation. *arXiv preprint arXiv:2501.19227*, 2025. 3

[32] Miguel Angel Quiroz Martinez. Deep learning for edge computing: A survey. *Artificial Intelligence, Computer and Software Engineering Advances: Proceedings of the CIT 2020 Volume 1*, 1326:79, 2021. 1

[33] Ryan G McClarren, Penrose McClarren, and R Penrose. Uncertainty quantification and predictive computational science. 2018. 2

[34] Joseph Victor Michalowicz, Jonathan M Nichols, and Frank Bucholtz. *Handbook of differential entropy*. Crc Press, 2013. 5

[35] Sorour Mohajerani and Parvaneh Saeedi. Cloud-net: An end-to-end cloud detection algorithm for landsat 8 imagery. In *IGARSS 2019-2019 IEEE international geoscience and remote sensing symposium*, pages 1029–1032. IEEE, 2019. 5

[36] Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. Towards robust and reproducible active learning using neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 223–232, 2022. 2, 6, 7

[37] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152, 2018. 3

[38] Danfeng Qin, Chas Leichner, Manolis Delakis, Marco Fornoni, Shixin Luo, Fan Yang, Weijun Wang, Colby Banbury, Chengxi Ye, Berkin Akin, et al. Mobilenetv4: Universal models for the mobile ecosystem. In *European Conference on Computer Vision*, pages 78–96. Springer, 2024. 6

[39] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021. 2

[40] Ivan Rodriguez-Ferrandez, Leonidas Kosmidis, Maris Tali, David Steenari, Alex Hands, and Camille Bélanger-Champagne. Proton evaluation of single event effects in the nvidia gpu orin som: Understanding radiation vulnerabilities beyond the soc. In *2024 IEEE 30th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 1–7, 2024. 1, 7

[41] Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. Deep bayesian active semi-supervised learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 158–164. IEEE, 2018. 2

[42] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International symposium on intelligent data analysis*, pages 309–318. Springer, 2001. 2, 4

[43] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 2

[44] Burr Settles. Active learning literature survey. 2009. 2

[45] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2

[46] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015. 2

[47] Wei Tan, Lan Du, and Wray Buntine. Diversity enhanced active learning with strictly proper scoring rules. *Advances in Neural Information Processing Systems*, 34:10906–10918, 2021. 2

[48] Alaa Tharwat and Wolfram Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4):820, 2023. 2

[49] Devis Tuia, Konrad Schindler, Begüm Demir, Xiao Xiang Zhu, Mrinalini Kochupillai, Sašo Džeroski, Jan N van Rijn, Holger H Hoos, Fabio Del Frate, Mihai Datcu, et al. Artificial intelligence to advance earth observation: A review of models, recent trends, and pathways forward. *IEEE Geoscience and Remote Sensing Magazine*, 2024. 2

[50] Ke Wang, Chongqiang Shen, Xingcan Li, and Jianbo Lu. Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications. *IEEE Transactions on Intelligent Transportation Systems*, 2025. 3

[51] Xiaofei Wang, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE communications surveys & tutorials*, 22(2):869–904, 2020. 1

[52] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. The label complexity of active learning from observational data. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[53] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE transactions on knowledge and data engineering*, 35(9):8934–8954, 2022. 2

[54] Xudong Yao, Qing Guo, and An Li. Cloud detection in optical remote sensing images with deep semi-supervised and active learning. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023. 4, 6, 7, 8

[55] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. Towards deeper graph neural networks

with differentiable group normalization. *Advances in neural information processing systems*, 33:4917–4928, 2020. 2

[56] Shuai Zhu, Thiemo Voigt, Fatemeh Rahimian, and Jeonggil Ko. On-device training: A first overview on existing systems. *ACM transactions on sensor networks*, 20(6):1–39, 2024. 1, 2