Deep Evidential Uncertainty Estimation for Semantic Segmentation under Out-Of-Distribution Obstacles

Siddharth Ancha[1], Philip R. Osteen[2] and Nicholas Roy[1]

Website: https://siddancha.github.io/projects/evidential-segmentation-uncertainty[†]

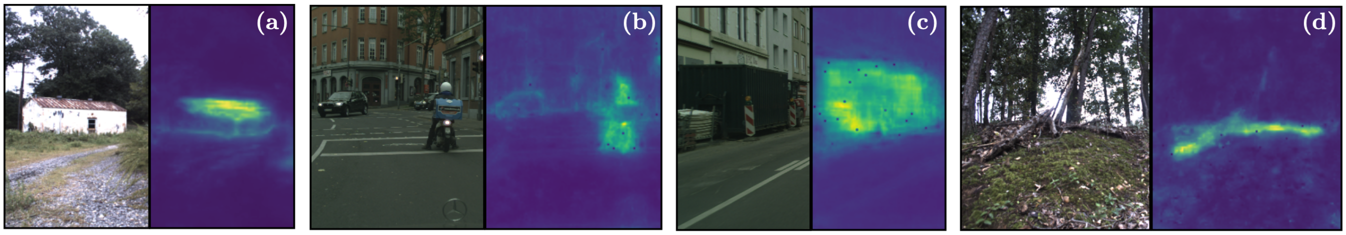


Fig. 1: We present a method for accurate uncertainty estimation in semantic segmentation. Our method produces high epistemic uncertainty for out-of-distribution segments such as (a) structures, (b) motorbikes and (c) containers not present in the training data. It is able to identify long-tail anomalies like (d) fallen trees as out-of-distribution even though standing trees are correctly predicted as in-distribution.

***Abstract*— In order to navigate safely and reliably in novel environments, robots must estimate perceptual uncertainty when confronted with out-of-distribution (OOD) obstacles not seen in training data. We present a method to accurately estimate *pixel-wise* uncertainty in semantic segmentation without requiring real or synthetic OOD examples at training time. From a shared per-pixel latent feature representation, a classification network predicts a categorical distribution over semantic labels, while a normalizing flow estimates the probability density of features under the training distribution. The label distribution and density estimates are combined in a Dirichlet-based evidential uncertainty framework that efficiently computes epistemic and aleatoric uncertainty in a single neural network forward pass. Our method is enabled by three key contributions. First, we simplify the problem of learning a transformation to the training data density by starting from a fitted Gaussian mixture model instead of the conventional standard normal distribution. Second, we learn a richer and more expressive latent pixel representation to aid OOD detection by training a decoder to reconstruct input image patches. Third, we perform theoretical analysis of the loss function used in the evidential uncertainty framework and propose a principled objective that more accurately balances training the classification and density estimation networks. We demonstrate the accuracy of our uncertainty estimation approach under long-tail OOD obstacle classes for semantic segmentation in both off-road and urban driving environments.**

I. INTRODUCTION

Autonomous robotic navigation has become increasingly pervasive in both structured urban environments [23, 62, 19] as well as off-road, unstructured environments like planetary exploration [2, 41], search-and-rescue [29], mines [16] and forests [18]. Navigation systems rely on semantic segmentation of camera images [54, 55, 24] to detect various semantic types and object classes. Robots operating in such real world environments often face “out-of-distribution” (OOD) obstacles that are not well-represented in the training data. However, most deep-learning based semantic segmentation models not only make unexpected errors on OOD examples, but they often have no notion of how incorrect their predictions are.

In order to navigate safely and reliably in novel environments, autonomous robots must estimate *uncertainty* in pixel-wise semantic segmentation to anticipate potential errors. Estimating a high uncertainty for OOD segments can enable the robot to execute cautious, risk-averse behaviors and avoid colliding with such obstacles. Uncertainty in deep neural networks arises from two sources [30]: *aleatoric* uncertainty is the inherent and irreducible uncertainty due to sensor noise and partial observability, whereas *epistemic* uncertainty is due to unfamiliar inputs not well-represented in the limited training dataset. Bayesian neural networks [21, 43] provide a principled framework to estimate *both* uncertainties for an input $\mathbf{x}$ by not just predicting a single categorical distribution $\mathbf{p} = (p_1, \ldots, p_C \mid \sum_{c=1}^{C} p_c = 1)$ over C classes, but by predicting a hierarchical *distribution over distributions* $p(\mathbf{p} \mid \mathbf{x})$ (see Sec. II for details). However, conventional Bayesian uncertainty estimators like variational inference [4, 26, 36], ensembles [34, 56] and MC-Dropout [20] require multiple forward passes through a neural network and are prohibitively slow for real-time robotics applications.

More recently, *evidential* uncertainty [50, 1, 53] estimators such as the *natural posterior network* (NatPN) [6, 7] have emerged as an efficient alternative that directly predict the parameters of a hierarchical Dirichlet uncertainty distribution in a single forward pass (see Fig. 2). First, a latent feature representation $\mathbf{x}^i \in \mathbb{R}^D$ is computed for each pixel indexed by i using an encoder network. Then, Bayesian uncertainty is

[1] S. Ancha and N. Roy are with the Computer Science and Artifical Intelligence Lab (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139, USA. {sancha, nickroy}@csail.mit.edu

[2] P. R. Osteen is with the DEVCOM Army Research Laboratory, Adelphi, MD 20783, USA. philip.r.osteen.civ@army.mil. Distribution Statement A. Approved for public release: distribution unlimited

[†] While this paper is fully self-contained, our project website contains an overview video (3min), qualitative visualizations and detailed background and proofs in a technical report for easy access.

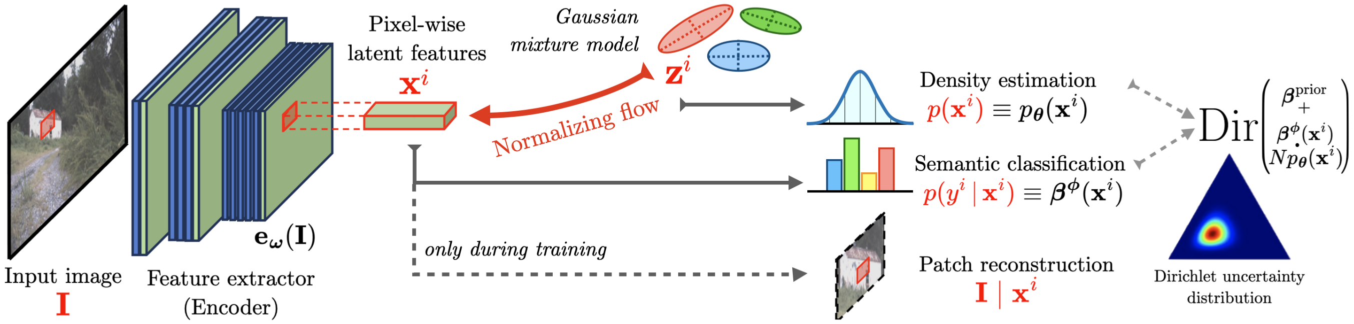

Input image
$\mathbf{I}$
Feature extractor (Encoder)
$\mathbf{e}_{\boldsymbol{\omega}}(\mathbf{I})$
Pixel-wise latent features
$\mathbf{x}^i$
Gaussian mixture model
$\mathbf{z}^i$
Normalizing flow
only during training
Density estimation
$p(\mathbf{x}^i) \equiv p_{\boldsymbol{\theta}}(\mathbf{x}^i)$
Semantic classification
$p(y^i \mid \mathbf{x}^i) \equiv \boldsymbol{\beta}^{\phi}(\mathbf{x}^i)$
Patch reconstruction
$\mathbf{I} \mid \mathbf{x}^i$
$\mathrm{Dir}\left(\boldsymbol{\beta}^{\text{prior}} + \boldsymbol{\beta}^{\phi}(\mathbf{x}^i) \cdot N p_{\boldsymbol{\theta}}(\mathbf{x}^i)\right)$
Dirichlet uncertainty distribution

Fig. 2: *Overview of our method.* A learned feature extractor (encoder), that forms the backbone of standard semantic segmentation pipelines, inputs image $\mathbf{I}$ and outputs a latent feature representation $\mathbf{x}^i$ for each pixel i. A semantic classifier head classifies the latent representation into one of C class labels y^i by predicting $p(y^i \mid \mathbf{x}^i)$; this helps determine aleatoric uncertainty over labels. A normalizing flow learns an invertible transformation from a fitted Gaussian mixture model to the latent distribution. The invertible transformation allows computing the probability density $p(\mathbf{x}^i)$ of a given latent vector $\mathbf{x}^i$ to determine epistemic uncertainty. The classification output and density estimates are combined in an evidential uncertainty framework to produce a Dirichlet-based hierarchical uncertainty distribution. Finally, a decoder inputs the pixel's shared latent representation $\mathbf{x}^i$ to reconstruct a patch in the original image centered at the pixel. Patch reconstruction encourages the feature extractor to learn expressive features $\mathbf{x}^i$ that enable detecting OOD segments. The decoder is not required at test time.

computed in two parts: (i) A linear classifier predicts a categorical distribution $p(y^i \mid \mathbf{x}^i)$ representing aleatoric uncertainty over semantic labels $y^i \in \{1, \ldots, C\}$. (ii) A density estimator predicts the probability density $p(\mathbf{x}^i)$ of the features for pixel i under the training data distribution, representing epistemic uncertainty. A low $p(\mathbf{x}^i)$ corresponds to high uncertainty since a low $p(\mathbf{x}^i)$ means that $\mathbf{x}^i$ is not well-represented in the training data. The density is estimated by learning a *normalizing flow* [48, 46]. Normalizing flows estimate densities $p(\mathbf{x}^i)$ in high-dimensional spaces by learning an invertible and differentiable transformation between $\mathbf{x}$ and samples from a standard normal distribution $\mathbf{z}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}^i \in \mathbb{R}^D$. All components are trained end-to-end, and both uncertainties are combined under the evidential uncertainty framework by interpreting them as parameters of a Dirichlet distribution [60].

In this work, we address three challenges of using evidential uncertainty estimation for semantic segmentation and make the following **main contributions:**

1) Pixel-wise latent semantic features learned using existing segmentation networks may not be sufficiently expressive to identify OOD segments. If this representation *aliases* the input data, i.e., maps OOD objects onto in-distribution features, the density estimator may predict a high probability which is accurate for the features but not the input query as a whole. We propose training a decoder that uses each pixel's latent features to reconstruct an image patch around the pixel. The reconstruction loss forces the latent representation to encode more information that helps detect OOD examples (Sec. III).

2) Normalizing flows are prone to the curse of dimensionality since semantic segmentation empirically requires a high-dimensional latent space (≥ 128) [35, 33, 8], as opposed to fewer latent dimensions (≤ 32) for the kinds of classification tasks considered in prior work [6, 7]. Conventionally, an invertible transformation is learnt between $p(\mathbf{x}^i)$ and the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Our insight is to exploit the fact that normalizing flows can admit any base distribution with a differentiable density. In particular, we use a Gaussian mixture model (GMM) that is first fit to the data distribution. The more powerful base distribution enables the flow to perform at least as well as the GMM even if it learns a simple identity mapping. We show that requiring the normalizing flow to only learn the *residual* transformation between the GMM and $p(\mathbf{x}^i)$ significantly improves performance (Sec. IV).

3) NatPN uses a *Bayesian loss function* [6, 7] that linearly combines the loss for classification and density estimation. We provide an analysis of this loss function and derive a novel combination coefficient that is a function of the predicted density estimate. We show that our proposed combination balances the two losses in a principled way and satisfies a number of desirable theoretical properties. The proposed loss reduces the number of tunable hyperparameters and empirically improves uncertainty estimation (Sec. V).

We validate our uncertainty estimation approach on two semantic segmentation datasets: RUGD [57], an off-road driving dataset, and Cityscapes [11]: an urban driving dataset.

II. Problem Formulation

Given a 2-D RGB image $\mathbf{I} \in \{0, \ldots, 255\}^{3 \cdot W \cdot H}$ encountered by a mobile robot in urban or off-road environments, our goal is to classify each pixel $i \in \mathbf{I}$ with a distribution over per-pixel semantic labels $y^i \in \mathbb{C} = \{1, \ldots, C\}$. More precisely, given a training dataset of images $\mathcal{D} = \{\mathbf{I}_{0:t}\}$, its associated training labels, and a query image $\mathbf{I}$, we wish to learn

$$p(y^i \mid \mathbf{I}, \mathcal{D}) \tag{1}$$

Firstly, for each pixel (indexed by i) we assume a learned feature representation $\mathbf{x}^i \in \mathbb{R}^D$ computed from an encoding of the raw image $\mathbf{e}^i(\mathbf{I}) = \mathbf{x}^i$. Secondly, in order to accurately capture both aleatoric and epistemic uncertainty, we use hierarchical Bayesian uncertainty [21, 30, 43] to factor the model and compute $p(y^i \mid \mathbf{x}^i, \mathcal{D})$ as follows:

$$\underbrace{p(y^i \mid \mathbf{x}^i, \mathcal{D})}_{\text{label posterior}} = \int_{\mathbf{p}} \overbrace{p(y^i \mid \mathbf{p}^i)}^{\textit{aleatoric}} \underbrace{\overbrace{p(\mathbf{p}^i \mid \mathbf{x}^i, \mathcal{D})}^{\textit{epistemic}}}_{\text{distributional posterior}} \mathrm{d}\mathbf{p}. \tag{2}$$

In this model, the $p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D})$, is known as the *distributional uncertainty* [37] or *distributional posterior* (shown in red throughout the paper). Aleatoric uncertainty is expressed by the (average) spread of the lower-level label distribution $p(y^i \,|\, \mathbf{p}^i)$. Epistemic uncertainty is expressed by the spread of the the higher-level distribution $p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D})$, measuring the disagreement between the different $\mathbf{p}^i$ sampled from $p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D})$. Thus, the overall uncertainty (of the label posterior $p(y^i \,|\, \mathbf{x}^i, \mathcal{D})$) for pixel i is a combination of epistemic and aleatoric uncertainty. Note that the label posterior $p(y^i \,|\, \mathbf{x}^i, \mathcal{D})$ is simply the mean of the distributional posterior $p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D})$ and y^i is independent of $\mathbf{x}^i$ conditioned on $\mathbf{p}^i$ since the distributional posterior is computed from $\mathbf{x}^i$.

We compute $p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D})$ by repurposing a standard semantic segmentation architecture into a feature extractor (or encoder) and a pixel-wise classification head, shown in Fig. 2. In this work, we build upon the feature pyramid network [35], which is a fast and accurate semantic segmentation method. First, the encoder $\mathbf{e}_{\boldsymbol{\omega}}(\cdot)$, parametrized by weights $\boldsymbol{\omega}$ takes as input an RGB image $\mathbf{I}$ and extracts a latent feature vector $\mathbf{x}^i = \mathbf{e}^i_{\boldsymbol{\omega}}(\mathbf{I}) \in \mathbb{R}^D$ for each pixel i. Given latent features $\mathbf{x}^i \in \mathbb{R}^D$ for pixel i, we compute its distributional uncertainty $p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D})$ by expressing it as a closed-form Dirichlet distribution [60] over the probability simplex Δ^C. We follow the approach of the *natural posterior network* (NatPN) [6, 7] and parametrize the Dirichlet as:

$$p(\mathbf{p}^i \,|\, \mathbf{x}^i, \mathcal{D}) = \mathrm{Dir}\Big(\mathbf{p}^i \,\Big|\, \underbrace{\boldsymbol{\beta}^{\text{prior}}}_{(1,\ldots,1)} + \underbrace{\boldsymbol{\beta}^{\boldsymbol{\phi}}(\mathbf{x}^i)}_{\text{classification output}} \cdot N \underbrace{p_{\boldsymbol{\theta}}(\mathbf{x}^i)}_{\text{flow output}}\Big) \quad (3)$$

The above form is motivated by exact Bayesian inference for a Dirichlet distribution [60]. The Dirichlet parameters combine a categorical distribution $\boldsymbol{\beta}^{\boldsymbol{\phi}}(\mathbf{x}^i) \in \Delta^C$ over semantic classes predicted by a classification head with weights $\boldsymbol{\phi}$, a density estimate $p_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}_{\geq 0}$, and a distributional prior $\mathrm{Dir}(\mathbf{p}^i \,|\, \boldsymbol{\beta}^{\text{prior}})$ where $\boldsymbol{\beta}^{\text{prior}} = (1, \ldots, 1)$ corresponds to a uniform distribution over the probability simplex Δ^C.

The pixel feature density $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$ is the crucial component for estimating epistemic uncertainty, as it indicates how familiar the feature vector is in the training dataset. The density is scaled by a constant N so that it can be interpreted as the "evidence" for pixel features $\mathbf{x}^i$: $Np_{\boldsymbol{\theta}}(\mathbf{x}^i)$ is the *effective count* of $\mathbf{x}^i$ in the training dataset. When $p_{\boldsymbol{\theta}}(\mathbf{x})$ is high, the magnitude of the Dirichlet parameter increases and corresponds to a peaked distribution around the classifier's prediction $\boldsymbol{\beta}^{\boldsymbol{\phi}}(\mathbf{x}^i)$. However, as $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$ decreases to zero, the Dirichlet diffuses to $\mathrm{Dir}(\mathbf{p}^i \,|\, \boldsymbol{\beta}^{\text{prior}})$ i.e. the uniform distribution.

The advantage of using Dirichlet-based evidential uncertainty is that a *single* forward pass through the encoder, classifier and density estimator produces the full distributional uncertainty in Eqn. 3. Furthermore, the label posterior (Eqn. 2), epistemic and aleatoric uncertainty can be computed efficiently in closed-form for the Dirichlet distribution [37]. In contrast, conventional Bayesian uncertainty methods [4, 26, 36, 34, 56, 20] require multiple forward passes to compute distributional uncertainty, which can be prohibitively expensive for real-time robotics applications. All components of our method can be trained end-to-end using a single Bayesian loss function [6, 7].

III. Latent features via patch reconstruction

A standard segmentation network is trained by extracting features $\mathbf{x}^i = \mathbf{e}^i_{\boldsymbol{\omega}}(\mathbf{I})$ for each pixel i using an encoder $\mathbf{e}_{\boldsymbol{\omega}}$ to learn the features, and minimizing a semantic classification loss for the predicted categorical distribution $\boldsymbol{\beta}^{\boldsymbol{\phi}}(\mathbf{x}^i)$. However, latent features that are sufficient for semantically classifying a pixel may be insufficient for detecting OOD objects. For example, assume that the training dataset contains classes such as sky, grass and mud, but not many obstacles. In that case, a semantic classifier might only need to learn color features for accurate classification. It may discard other features such as texture and shape that are important to differentiate obstacles such as a brown vehicle from mud or a white building from the sky. Charpentier et al. [7] indeed note that their density estimator can only identify OOD features that pertain to the semantic classification task.

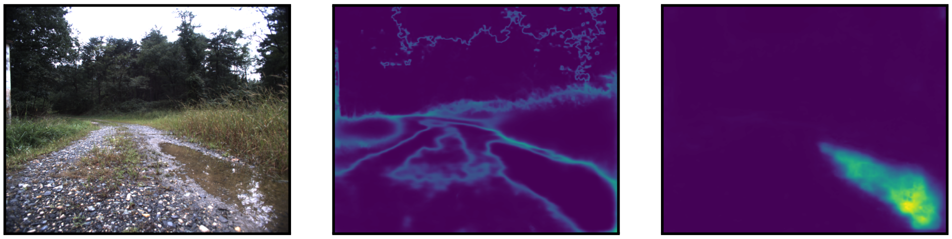
We propose an auxiliary decoder network that takes as input the latent representation $\mathbf{x}^i$ of each pixel i, and learns to reconstruct a 60 pixel × 60 pixel patch centered at pixel i (see Fig. 2). We call this a *patch-decoder* since it decodes the latent representation to an image patch. A similar idea was introduced in Richter and Roy [49] to autoencode images for image-level anomaly detection. The patch decoder is trained to minimize the smooth-L1 distance [22] between the predicted reconstruction and the ground truth image patch. The reconstruction objective encourages the latent representation to learn expressive features that enable identifying OOD segments. The decoder provides an auxiliary reconstruction loss for training the shared latent features; however, the decoder is not required at test time and is discarded. Therefore, the patch decoder does not add any computational overhead to our method during deployment. Sec. VI shows that training an auxiliary patch decoder significantly improves uncertainty estimation in the presence of OOD segments.

IV. Improving normalizing flows via GMMs

In order to to estimate the density $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$ of pixel-wise latent features $\mathbf{x}^i \in \mathbb{R}^D$, we use a normalizing flow model [46], which transforms a simple, fixed base distribution $p(\mathbf{z}^i)$, $\mathbf{z}^i \in \mathbb{R}^D$, conventionally the standard normal distribution $\mathcal{N}(\mathbf{z}^i \,|\, \mathbf{0}, \mathbf{1})$, to a more complex distribution using a sequence of T invertible functions $\{f^{\boldsymbol{\theta}}_t : \mathbb{R}^D \to \mathbb{R}^D\}_{t=1}^T$ parametrized by weights $\boldsymbol{\theta}$:

$$\text{(Sampling)} \quad \mathbf{z}^i \sim p(\mathbf{z}^i) = \mathbf{z}^i_0 \xrightarrow{f^{\boldsymbol{\theta}}_1} \mathbf{z}^i_1 \xrightarrow{f^{\boldsymbol{\theta}}_2} \cdots \xrightarrow{f^{\boldsymbol{\theta}}_T} \mathbf{z}^i_T = \mathbf{x}^i \quad (4)$$

Each learnable transformation $f^{\boldsymbol{\theta}}_t$ is designed to be invertible and differentiable by construction [46, 9, 48], and their Jacobians are efficient to compute. We use residual flows [9], a state-of-the art normalizing flow method that uses a sequence of invertible residual networks [3] guaranteed to be invertible by Lipschitz continuity. These properties allow us to compute the probability density of a pixel's latent features $\mathbf{x}^i$ using



(a) Test-time query image **(b)** Estimated *aleatoric* uncertainty **(c)** Estimated *epistemic* uncertainty

Fig. 3: (a) The input/query image at test time containing a puddle of water; water or puddles were never seen by the model during training. **(b)** Aleatoric uncertainty is the inherent and irreducible uncertainty due to ambiguous labels, and is estimated to be high at the boundaries between semantic classes. **(c)** Estimated epistemic uncertainty is high at unfamiliar regions not well represented in the training data. Although aleatoric uncertainty is high only at the boundary of the puddle (due to misclassification), epistemic uncertainty is high throughout the puddle. Therefore, we distinguish between, model and predict both uncertainties in a Bayesian evidential framework.

the change-of-variables formula:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^i) = \log p(\mathbf{z}^i) - \sum_{t=1}^{T} \log \left| \det \frac{\partial f^{\boldsymbol{\theta}}(\mathbf{z}_t^i)}{\partial \mathbf{z}_t^i} \right| \qquad (5)$$

$$\text{(Density estimation)} \quad \text{where } \mathbf{z}_t^i = {f^{\boldsymbol{\theta}}}_t^{-1} \circ \cdots \circ {f^{\boldsymbol{\theta}}}_T^{-1}(\mathbf{x}^i) \qquad (6)$$

The target distribution of pixel latent features $\mathbf{x}^i$ we want to learn is the distribution induced by the encoder $\mathbf{x}^i = \mathbf{e}_{\boldsymbol{\omega}}^i(\mathbf{I})$ with weights $\boldsymbol{\omega}$; we represent this target by $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$. However, the target density is multimodal and complex; therefore, the target transformation from the simple base density $\mathcal{N}(\mathbf{0}, \mathbf{1})$ to $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$ is also complex and challenging for the normalizing flow to learn. This is especially true in high dimensional ($D \geq 128$) latent spaces $\mathbf{x}^i \in \mathbb{R}^D$ used for semantic segmentation.

Our insight is to ease this learning problem by exploiting the fact that normalizing flows can admit any base density that is computable and differentiable. In particular, we attempt to bridge the gap to $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$ by using a more complex, learnable base density: a Gaussian mixture model (GMM) $p_{\mathrm{GMM}}(\cdot)$.

Our training of the GMM-based normalizing flow occurs in three stages. First, we train the encoder $\mathbf{e}_{\boldsymbol{\omega}}(\cdot)$, segmentation head $\boldsymbol{\beta}^{\boldsymbol{\phi}}(\cdot)$, and the decoder jointly using classification and reconstruction losses. Second, we freeze the weights $\boldsymbol{\omega}$ of the encoder and fit $p_{\mathrm{GMM}}(\mathbf{x}^i)$ to the pixel latent feature distribution $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$ using the EM algorithm [14]. Crucially, the gap between an optimized $p_{\mathrm{GMM}}(\mathbf{x}^i)$ and $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$ is expected to be smaller than the gap between $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$, since $p_{\mathrm{GMM}}(\cdot)$ is strictly more expressive than $\mathcal{N}(\mathbf{0}, \mathbf{1})$. Finally, we train the classifier head $\boldsymbol{\beta}^{\boldsymbol{\phi}}(\mathbf{x}^i)$ and normalizing flow $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$ jointly using the Bayesian loss function [6, 7] (see Sec. V), but using $p_{\mathrm{GMM}}(\mathbf{z}^i)$ as the base density for the normalizing flow instead of $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

The above procedure makes the learning problem easier because even if the flow learns a simple identity transform $f_t^{\boldsymbol{\theta}}(\mathbf{z}_t^i) = \mathbf{z}_t^i \; \forall t$, we have that $p_{\boldsymbol{\omega}}(\mathbf{x}^i) = p_{\mathrm{GMM}}(\mathbf{x}^i)$. Therefore, the normalizing flow is guaranteed to perform at least as well as the GMM. The GMM can be viewed as a clever "initialization" that offsets the burden on the normalizing flow to learn a simpler *residual* transport between $p_{\mathrm{GMM}}(\mathbf{x}^i)$ and $p_{\boldsymbol{\omega}}(\mathbf{x}^i)$. In Sec. VI, we show that using a fitted GMM as the base density significantly outperforms either using only normalizing flows or only GMMs.

V. Improving Training Loss via Density Estimates

We follow NatPN [6, 7] and train our full architecture end-to-end by minimizing a single *Bayesian loss function*:

$$\arg\min_{\boldsymbol{\phi}} \sum_{n} \Big[\underbrace{-\mathbb{E}_{\mathbf{p} \sim \mathrm{Dir}^{(n)}} \log \big(y^{(n)} \,\big|\, \mathbf{p} \big)}_{\text{expected cross-entropy loss term}} - \lambda \underbrace{\mathbb{H}\big(\mathrm{Dir}^{(n)}\big)}_{\text{entropy regularization term}} \Big]$$

$$\text{where } \mathrm{Dir}^{(n)} := \mathrm{Dir}\big(\mathbf{p} \,\big|\, \boldsymbol{\beta}^{\mathrm{prior}} + N\, p_{\boldsymbol{\theta}}(\mathbf{x}^{(n)})\, \boldsymbol{\beta}^{\boldsymbol{\phi}}(\mathbf{x}^{(n)})\big) \qquad (7)$$

where the training dataset $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{|\mathcal{D}|}$ contains pixel features $\mathbf{x}^{(n)}$ extracted by the encoder, and corresponding ground truth labels $y^{(n)}$. The loss balances between two terms: (i) an expected cross-entropy loss that trains the classifier to predict the conditional label distribution $p(y^i \mid \mathbf{x}^i)$, and (ii) an entropy regularization term that prevents the predicted Dirichlet distribution from being excessively concentrated. Charpentier et al. [6, 7] hand-tune λ and set it to a constant value for all $\mathbf{x}^{(n)} \in \mathbb{R}^D$.

We propose a principled value for λ by analyzing the loss as an evidence lower-bound (ELBO) [31] for exact Bayesian inference in Dirichlet distributions. The standard Bayesian ELBO contains two terms [31]: a *prior* term that corresponds to the entropy regularization term, and a *likelihood* term that corresponds to the the cross-entropy loss. Our insight is to note that the prior term occurs only once for each unique $\mathbf{x} \in \mathcal{D}$ in the dataset (all unique data points should be equally regularized). However, the likelihood term is contributed by every occurrence of $\mathbf{x}$ in the dataset. Therefore, the relative weight of the two terms should depend on the frequency of $\mathbf{x}$ — its *effective count* or "evidence" — in the dataset, which is modeled as a scaled version of its probability density $N p_{\boldsymbol{\theta}}(\mathbf{x})$. The higher the value of $N p_{\boldsymbol{\theta}}(\mathbf{x})$, the lower the relative weight of entropy regularization should be.

Therefore, we propose an analytical value for the coefficient: $\lambda := (N p_{\boldsymbol{\theta}}(\mathbf{x}^{(n)}))^{-1}$ i.e. the reciprocal of the (scaled) predicted density. Importantly, our proposed λ is not a constant but is a function of pixel features $\mathbf{x}^{(n)}$. Our proposed coefficient can be further justified by proving[1] that (i) when $\lambda \neq (N p_{\boldsymbol{\theta}}(\mathbf{x}^{(n)}))^{-1}$, the true posterior may not minimize the Bayesian loss function. But when $\lambda = (N p_{\boldsymbol{\theta}}(\mathbf{x}^{(n)}))^{-1}$ (ii) the true posterior is the *unique global minimum* of the

[1] The proof is omitted due to space constraints.

Bayesian loss, and (iii) the Bayesian loss function trains the normalizing flow to maximize its predicted density $p_{\boldsymbol{\theta}}(\mathbf{x}^{(n)})$ on the training data. In Sec. VI, we empirically show that optimizing under our proposed value $\lambda = (N p_{\boldsymbol{\theta}}(\mathbf{x}^{(n)}))^{-1}$ produces more accurate uncertainty estimates than using a constant value for all $\mathbf{x}^{(n)}$.

VI. EXPERIMENTS

A. Dataset and experimental design

We wish to estimate pixel-wise distributional uncertainty in the presence of OOD obstacles. From the full semantic label set $\mathbb{C} = \{1, \ldots, C\}$, we define a subset $\mathbb{C}_{\text{ID}} \subset \mathbb{C}$ as the set of in-distribution classes, and the remaining subset $\mathbb{C}_{\text{OOD}} = \mathbb{C} \setminus \mathbb{C}_{\text{ID}}$ as out-of-distribution classes. For the purposes of this OOD evaluation, we split the dataset such that pixels in the training dataset $\mathcal{D}$ belong exclusively to $\mathbb{C}_{\text{ID}}$, whereas test images contain pixels drawn from both $\mathbb{C}_{\text{ID}}$ and $\mathbb{C}_{\text{OOD}}$. Importantly, we assume that the learner has no access to any amount of real or synthetic OOD data at training time, and we do not make any assumptions about the nature of OOD data that the model can expect at test time. We validate our approach using two datasets:

1. RUGD dataset [57]: a real-world dataset containing camera images collected in off-road environments using mobile robot platforms and manually labeled for semantic segmentation. It contains 7,453 labeled images from 17 scenes. We split the 24 semantic categories as 16 in-distribution labels: $\mathbb{C}_{\text{ID}}$ = {`dirt`, `sand`, `grass`, `tree`, `pole`, `sky`, `asphalt`, `gravel`, `mulch`, `rock-bed`, `log`, `fence`, `bush`, `sign`, `rock`, `concrete`}, and 8 OOD labels corresponding to "obstacle" classes: $\mathbb{C}_{\text{OOD}}$ = {`vehicle`, `container/generic-object`, `building`, `bicycle`, `person`, `bridge`, `picnic-table`, `water`}.

2. Cityscapes dataset [11]: a real-world dataset containing camera images collected from a vehicle driving in urban German streets. It contains 3,475 finely labeled images collected from 50 cities. We split the 19 semantic categories as 15 in-distribution labels: $\mathbb{C}_{\text{ID}}$ = {`road`, `sidewalk`, `building`, `wall`, `fence`, `pole`, `traffic-light`, `traffic-sign`, `vegetation`, `terrain`, `sky`, `person`, `rider`, `car`, `truck`} and 4 OOD labels corresponding to rarer obstacles: { `bus`, `train`, `motorcycle`, `bicycle` }.

B. Evaluation metrics

We evaluate our approach using the following metrics. The results of our evaluation are presented in Table I.

(i) Overall uncertainty evaluation: We evaluate the overall uncertainty of the label posterior distribution $p(y^i \,|\, \mathbf{x}^i, \mathcal{D})$. We use the log-probability of the true label under the predicted distribution, and the Brier score [5, 45, 6]: $\sum_c (p(y^i = c \,|\, \mathbf{x}^i, \mathcal{D}) - y^*_c)^2$, where $y^*_c \in \{0, 1\}$ is a binary variable indicating whether the ground truth label is c or not. Both log-probability and Brier scores are *strictly-proper scoring rules* [45] i.e. they are uniquely optimized by the ground truth probability distribution and simultaneously evaluate refinement and calibration [45, 13, 42]. They are commonly used to evaluate uncertainty [45, 6, 25]. We also compute the expected calibration error (ECE) [25], a commonly used confidence calibration metric that measures the discrepancy between segmentation confidence $p \in [0, 1]$ and the empirical probability that a label predicted with confidence p was actually correct. Finally, we also evaluate the power of uncertainty to predict segmentation errors. We report the area under the precision-recall curve between overall uncertainty (measured as the entropy of $p(y^i \,|\, \mathbf{x}^i, \mathcal{D})$) and the classification error. A higher correlation between uncertainty and errors will produce a better score.

(ii) Density estimation: Since this work focuses on methods that that estimate epistemic uncertainty using density estimators, we additionally evaluate the accuracy of the density estimation. We report the log-density on in-distribution examples (higher is better) and OOD examples (lower is better). We also compute the area under the precision-recall curve between density, and whether the example is in-distribution (highlighted in blue in Table I).

(iii) Semantic segmentation accuracy: we also compute the mean intersection over union (mIoU), a standard semantic metric to evaluate semantic segmentation.

C. Baselines and ablations

We compare against multiple baselines in Table I. **GMM** (rows 2, 4): We first train the feature extractor (encoder), classification head and patch-decoder shown in Fig. 2 without a normalizing flow model. Then, we fit a GMM with a small (20) and large (200) number of components on the trained features. **NatPN** [7] (row 5): We adapt the vanilla NatPN architecture for semantic segmentation without adding any of our main contributions. The entire NatPN architecture is trained end-to-end. **Autoencoder-only** (row 1): We compare against the anomaly detection method of Richter and Roy [49] by using the patch-based autoencoder to predict pixel-wise reconstruction scores. The scores are treated analogous to the (negative of) density. **Nearest-neighbor search** (row 3): This baseline is akin to memorizing the training dataset. After jointly training the encoder, classifier and patch-decoder, we collect 50,000 randomly sampled pixel latent vectors computed on training images. For a given latent vector $\mathbf{x}^i$ of pixel i at test time, we output the distance of the nearest neighbor in the collected set. This is treated analogous to the (negative of) its density. Since the latter two methods do not output a normalized probability density $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$, we do not report any metrics that rely on $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$ for rows 1 and 3. Finally, we compare against an ensemble of 20 independently trained segmentation networks (row 6) that average predictions across members. Since ensembles don't compute densities $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$, we do not report density-based metrics. We also perform ablation experiments in rows 7-9, removing one of the three key contributions of this work at a time. Normalizing flows in rows 8-10 use the GMM with 20 components (row 2) as the base distribution. The key takeaways from these experiments are as follows.

First, we report that semantic segmentation accuracy (measured by mIoU) is very similar across methods. In the RUGD and Cityscapes experiments, the mIoU of all method (except deep ensembles) lies in 29.8±0.9 and 48.7±0.7, respectively

	Evaluating the overall uncertainty $p(y^i \mid \mathbf{x}^i, \mathcal{D})$								Evaluating the density estimator $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$						Time (ms)
	log-probability		Brier score		ECE		AUC-PR curve unc. v. acc.		log-density of ID latents		log-density of OOD latents		AUC-PR curve density v. ID		
	↑		↓		↓		↑		↑		↓		↑		↓
Autoencoder only [49]	*Not Applicable*								*Not Applicable*				0.049	0.554	15.8
GMM *20 components*	-1.278	-0.449	0.451	0.285	0.279	0.093	0.584	0.417	-180.1	-144.3	-212.8	-154.3	0.275	0.704	86.4
Nearest neighbor search	*Not Applicable*								*Not Applicable*				0.316	0.762	129.8
GMM *200 components*	-1.053	-0.409	0.386	0.128	0.184	0.079	0.612	0.457	-146.3	-135.6	-237.1	-162.8	0.353	0.783	700.7
Nat. Posterior Network [7]	-1.441	-0.732	0.547	0.507	0.395	0.422	0.417	0.268	-175.3	-163.3	-127.6	-138.5	0.063	0.611	47.4
Deep Ensemble *20 models* [34]	-1.017	-0.372	0.272	0.116	0.125	0.081	0.670	0.484	*Not Applicable*						1364.2
Ours *without* GMM	-1.392	-0.649	0.526	0.295	0.473	0.291	0.442	0.360	-168.0	-152.8	-138.1	-148.9	0.072	0.613	47.4
Ours *without* patch decoder	-1.335	-0.579	0.499	0.325	0.421	0.360	0.518	0.332	-157.6	-159.8	-140.1	-152.8	0.087	0.631	117.9
Ours *without* corrected loss	-0.968	**-0.356**	**0.270**	0.118	**0.111**	0.073	0.652	0.493	-124.6	-125.7	-302.1	-185.5	0.441	0.819	117.9
Ours (GMM + PD + corrected loss)	**-0.937**	-0.363	0.278	**0.112**	**0.111**	**0.052**	**0.680**	**0.501**	**-116.0**	**-116.5**	**-319.9**	**-192.5**	**0.502**	**0.838**	117.9

TABLE I: Measuring the accuracy of uncertainty estimation on the 🟥 RUGD [57] dataset (red) and the 🟩 Cityscapes [11] dataset (green) with held out (OOD) semantic classes at train time. We compute standard metrics for the accuracy of overall uncertainty (label posterior $p(y^i \mid \mathbf{x}^i, \mathcal{D})$) in the left half, as well as accuracy of the density estimates $p_{\boldsymbol{\theta}}(\mathbf{x}^i)$ in the right half.

whereas the 90% confidence intervals of each method is at least 1.8 and 1.4 respectively. Deep ensembles have the highest segmentation accuracy of 32.3 and 51.2 respectively. Ensembling tends to improve classification performance; however they are an order of magnitude slower than our method (see timing column in Table I) due to running multiple forward passes, and perform worse at uncertainty estimation.

Each of the three contributions (patch-based decoder, GMM-based normalizing flow, corrected Bayesian loss) improve performance on both datasets. In particular, the performance drops significantly when the normalizing flow does not use the GMM as its base distribution (rows 5, 7 vs. row 10). The performance also drops significantly when the patch-based decoder is not used to learn a good latent representation (rows 5, 8 vs. row 10). These results indicate that normalizing flows are challenging to train on high-dimensional latent distributions when initialized with a standard normal distribution. Normalizing flows combined with GMMs also outperform GMMs alone. In addition, semantic features learned without the guidance of a reconstruction loss fail to learn a representation conducive for OOD uncertainty estimation.

VII. Related Work

Ulmer et al. [53] provides a review of evidential methods for uncertainty estimation in deep learning. Aside from using density estimators for evidential deep learning [6, 7], older works either use regularization losses [50, 1], small amounts of OOD training data [37, 38] or distill the uncertainty from an ensemble into a Dirichlet distribution [40]. Amini et al. [1], Malinin et al. [39] apply the evidential uncertainty framework to continuous regression problems; this is orthogonal to our work as we focus on discrete (pixel-wise) classification. Relatedly, NatPN [7] (that we build upon and compare against) proposes evidential uncertainty estimation for a general class of exponential family distributions that encompasses both regression and classification.

Sirohi et al. [51] use the evidential uncertainty framework [50] to estimate per-pixel uncertainties. However, they focus on *panoptic segmentation* i.e. segmenting every object instance, and propose evaluation metrics tailored for this task. In contrast, we focus on *semantic segmentation*. Furthermore, we use a more recent form of evidential uncertainty estimation [7] that explicit trains a density estimator (normalizing flow) to estimate epistemic uncertainty, instead of relying on auxiliary regularization terms [50]. Petek et al. [47] combine evidential uncertainty for segmentation [50] with evidential deep learning [1] for bounding-box regression to perform robust localization from monocular images and HD maps.

Normalizing flows have shown to fail for OOD anomaly detection in prior work [44, 32, 10, 63] when they are trained directly on high dimensional image inputs. Kirichenko et al. [32], Jiang et al. [28] show that this is mitigated when applying normalizing flows to lower-dimensional latent features extracted from a neural network; we follow this approach in our work. Prior works have explored using GMMs with normalizing flows – either using a hand-crafted GMM [27] or jointly optimizing the GMM with the NF [52]. We first fit a GMM to the training data before training the NF. Using more complex 'resampling base distributions' [52, 17] could be a promising future direction for our work.

VIII. Conclusions

In this work, we develop an approach to accurately estimate pixel-wise Bayesian uncertainty for semantic segmentation in off-road and urban driving environments in the presence of out-of-distribution (OOD) obstacles not seen in the training data. Using an evidential deep learning framework that efficiently estimates both epistemic and aleatoric uncertainty in a single forward pass, this work establishes the importance of (i) encoding latent features essential for OOD detection by learning to reconstruct the input image, (ii) simplifying density estimation in high-dimensional latent spaces by learning residual transforms starting from pre-trained models, and (iii) balancing the level of regularization in the training objective informed by theoretical analysis, towards accurate and efficient uncertainty modeling. We hope this work paves the way towards inferring and harnessing uncertainty in perception systems as a critical tool to enable safe and reliable robot navigation in novel environments.

ACKNOWLEDGMENT

This material is based upon work supported by the DEVCOM Army Research Laboratory under Grant No. W911NF-21-2-0150. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DEVCOM Army Research Laboratory and DARPA.

REFERENCES

[1] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937, 2020.

[2] John Bares, Martial Hebert, Takeo Kanade, Eric Krotkov, Tom Mitchell, Reid Simmons, and William Whittaker. Ambler: An autonomous rover for planetary exploration. *Computer*, 22(6):18–26, 1989.

[3] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning (ICML)*, pages 573–582. PMLR, 2019.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*, pages 1613–1622. PMLR, 2015.

[5] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1): 1–3, 1950.

[6] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without OOD samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33: 1356–1367, 2020.

[7] Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural posterior network: Deep Bayesian predictive uncertainty for exponential family distributions. In *International Conference on Learning Representations*, 2022.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017.

[9] Ricky T. Q. Chen, Jens Behrmann, David K. Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[10] Hyunsun Choi, Eric Jang, and Alexander A Alemi. WAIC, but why? Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. URL `https://www.cityscapes-dataset.com/`.

[12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

[13] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[14] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[15] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

[16] David D. Fan, Kyohei Otsu, Yuki Kubo, Anushri Dixit, Joel Burdick, and Ali akbar Agha-mohammadi. STEP: Stochastic Traversability Evaluation and Planning for Risk-Aware Off-road Navigation. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.021.

[17] Jianxiang Feng, Jongseok Lee, Simon Geisler, Stephan Günnemann, and Rudolph Triebel. Topology-matching normalizing flows for out-of-distribution detection in robot learning. In *7th Annual Conference on Robot Learning (CoRL)*, 2023.

[18] Jonas Frey, David Hoeller, Shehryar Khattak, and Marco Hutter. Locomotion policy guided traversability learning using volumetric representations of complex environments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5722–5729. IEEE, 2022.

[19] Hironobu Fujiyoshi, Tsubasa Hirakawa, and Takayoshi Yamashita. Deep learning-based image recognition for autonomous driving. *IATSS research*, 43(4):244–252, 2019.

[20] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on*

Machine Learning (ICML), pages 1050–1059. PMLR, 2016.

[21] Yarin Gal et al. Uncertainty in deep learning. *Ph.D. Thesis, University of Cambridge*, 2016.

[22] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[23] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37 (3):362–386, 2020.

[24] Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. GA-Nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments. *IEEE Robotics and Automation Letters*, 7(3):8138–8145, 2022.

[25] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330. PMLR, 2017.

[26] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning (ICML)*, pages 1861–1869. PMLR, 2015.

[27] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In *International Conference on Machine Learning (ICML)*, pages 4615–4630. PMLR, 2020.

[28] Chiyu Max Jiang, Mahyar Najibi, Charles R Qi, Yin Zhou, and Dragomir Anguelov. Improving the intra-class long-tail in 3D detection via rare example mining. In *European Conference on Computer Vision*, pages 158–175. Springer, 2022.

[29] George Kantor, Sanjiv Singh, Ronald Peterson, Daniela Rus, Aveek Das, Vijay Kumar, Guilherme Pereira, and John Spletzer. Distributed search and rescue with robot and sensor teams. In *Field and service robotics: Recent advances in reserch and applications*, pages 529–538. Springer, 2006.

[30] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

[33] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9808, 2020.

[34] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[36] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[37] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[38] Andrey Malinin and Mark Gales. Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales. Regression prior networks. *arXiv preprint arXiv:2006.11590*, 2020.

[40] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020.

[41] Mauro Massari, Giovanni Giardini, Franco Bernelli-Zazzera, et al. Autonomous navigation system for planetary exploration rover based on artificial potential fields. In *Proceedings of Dynamics and Control of Systems and Structures in Space (DCSSS) 6th Conference*, pages 153–162, 2004.

[42] Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600, 1973.

[43] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[44] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, (ICLR), New Orleans, LA, USA*, 2019.

[45] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[46] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research (JMLR)*, 22(1):2617–2680, 2021.

[47] Kürsat Petek, Kshitij Sirohi, Daniel Büscher, and Wolfram Burgard. Robust monocular localization in sparse

hd maps leveraging multi-task uncertainty estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4163–4169. IEEE, 2022.

[48] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*, pages 1530–1538. PMLR, 2015.

[49] Charles Richter and Nicholas Roy. Safe Visual navigation via deep learning and novelty detection. In *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017. doi: 10.15607/RSS.2017.XIII.064.

[50] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[51] Kshitij Sirohi, Sajad Marvi, Daniel Büscher, and Wolfram Burgard. Uncertainty-aware panoptic segmentation. *IEEE Robotics and Automation Letters (RAL)*, 8(5):2629–2636, 2023.

[52] Vincent Stimper, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Resampling base distributions of normalizing flows. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4915–4936. PMLR, 2022.

[53] Dennis Thomas Ulmer, Christian Hardmeier, and Jes Frellsen. Prior and Posterior Networks: A Survey on Evidential Deep Learning Methods For Uncertainty Estimation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[54] Abhinav Valada, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *2016 International Symposium on Experimental Robotics*, pages 465–477. Springer, 2017.

[55] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. AdapNet: Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4644–4651. IEEE, 2017.

[56] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[57] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007. IEEE, 2019. URL `http://rugd.vision/`.

[58] Wikipedia contributors. Beta function: Wikipedia, The Free Encyclopedia, 2023. URL `https://en.wikipedia.org/w/index.php?title=Beta_function&oldid=1166085901`. [Online; accessed 28-July-2023].

[59] Wikipedia contributors. Digamma function: Wikipedia, The Free Encyclopedia, 2023. URL `https://en.wikipedia.org/w/index.php?title=Digamma_function&oldid=1160610038`. [Online; accessed 15-July-2023].

[60] Wikipedia contributors. Dirichlet distribution: Wikipedia, The Free Encyclopedia, 2023. URL `https://en.wikipedia.org/w/index.php?title=Dirichlet_distribution&oldid=1167012235`. [Online; accessed 27-July-2023].

[61] Wikipedia contributors. Gamma function: Wikipedia, The Free Encyclopedia, 2023. URL `https://en.wikipedia.org/w/index.php?title=Gamma_function&oldid=1167040765`. [Online; accessed 28-July-2023].

[62] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

[63] Lily Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning (ICLR)*, pages 12427–12436. PMLR, 2021.