This is a repository copy of *Out-of-distribution object detection through Bayesian uncertainty estimation*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/200330/

Version: Accepted Version

Out-of-distribution Object Detection through Bayesian Uncertainty Estimation

Tianhao Zhang[1], Shenglin Wang[1], Nidhal Bouaynaya[2], Radu Calinescu[3] and Lyudmila Mihaylova[1]

[1] *Automatic Control and Systems Engineering, The University of Sheffield, UK*
[2] *Department of Electrical and Computer Engineering, Rowan University, USA*
[3] *Department of Computer Science, University of York, UK*
Email: {tzhang81@sheffield.ac.uk, swang119@sheffield.ac.uk, bouaynaya@rowan.edu,
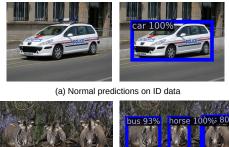radu.calinescu@york.ac.uk, l.s.mihaylova@sheffield.ac.uk}

Abstract—The superior performance of object detectors is often established under the condition that the test samples are in the same distribution as the training data. However, in many practical applications, out-of-distribution (OOD) instances are inevitable and usually lead to uncertainty in the results. In this paper, we propose a novel, intuitive, and scalable probabilistic object detection method for OOD detection. Unlike other uncertainty-modeling methods that either require huge computational costs to infer the weight distributions or rely on model training through synthetic outlier data, our method is able to distinguish between in-distribution (ID) data and OOD data via weight parameter sampling from proposed Gaussian distributions based on pre-trained networks. We demonstrate that our Bayesian object detector can achieve satisfactory OOD identification performance by reducing the FPR95 score by up to 8.19% and increasing the AUROC score by up to 13.94% when trained on BDD100k and VOC datasets as the ID datasets and evaluated on COCO2017 dataset as the OOD dataset.

Index Terms—Out-of-distribution detection, Uncertainty estimation, object detection, deep learning, image classification

I. INTRODUCTION

Out-of-distribution (OOD) detection is a machine learning technique that aims to detect test samples drawn from a distribution that differs from the distribution of the training data, with the definition of distribution to be well-defined according to the application in the target. The goal of OOD detection is to differentiate between inputs that are likely to be part of the distribution the model was trained on, and inputs that are not [4]. The vast majority of modern deep neural networks [5] are deterministic models that provide high-confidence predictions for inputs that are not seen during training, leading to poor generalization and unreliable results, as shown for example in Figure 1. OOD detection is particularly important for safety-related scenarios. For instance, an ideal uncertainty-aware deep object detector built for self-driving cars should be able to recognize the in-distribution (ID) target, such as people and cars, and produce a low-confidence prediction for the OOD objects, such as an animal.

Exploring the uncertainties of the deep learning algorithms can help perform OOD detection [4]. In general, there are two types of uncertainties: *aleatoric uncertainty*, due to the inherent and irreducible uncertainty of the input data, and *epistemic uncertainty* due to various deep learning model

(a) Normal predictions on ID data

(b) Overconfidence predictions on OOD data

(c) Ideal predictions provided by unknown-aware models

Figure 1: (a) An example prediction of PASCAL VOC dataset [1] provided by a standard Faster-RCNN [2] model trained on the same dataset. (b) The same Faster-RCNN model produces overconfident predictions on MS-COCO dataset [3], which is an OOD dataset in this case. (c) The "ideal" predictions provided by the uncertainty-aware object detection model can recognize OOD data.

frameworks [6]. A Bayesian Neural Networks (BNNs) is a type of neural network that incorporates Bayesian methods, such as probability distributions over model parameters, to represent these two types of uncertainties in the model's predictions [7]. BNNs can be implemented for OOD detection by comparing the uncertainties between the model's predictions on a given input and the model's predictions on a set of known ID data. If the uncertainty in the input is high, the input is likely OOD. Also, Bayesian methods can be used to estimate the likelihood of the data under the model. If this likelihood is low, the input is more likely to be OOD data. However, due to the high dimensionality and multi-modality of modern deep neural networks with millions of weight parameters, Bayesian approaches have largely been intractable for state-

of-the-art deep learning object detectors [8]. State-of-the-art deterministic CNNs have achieved huge success on object detection tasks [9]–[11]. A new inquiry is whether it is possible to create a Bayesian object detector that can maintain its outstanding performance on the ID dataset and have the ability to model uncertainty when it encounters the OOD dataset concurrently.

In order to address this problem, we propose a novel and scalable uncertainty-aware Bayesian deep learning method for OOD object detection. We exploit the information contained in the deterministic deep neural network layers when the model is trained in the ID dataset to efficiently approximate the posterior distribution over the weights. Our Bayesian approach focuses on performing Bayesian inference by sampling the predictions from the proposed approximation Gaussian distributions of neural weights. Our Bayesian inference method transforms the deterministic deep neural network layers into probabilistic Bayesian layers with no additional training cost. It makes it possible to maximize the uncertainty quantification abilities of BNNs on large computer vision tasks via BNNs.

In particular, our main contributions are:

- In this work, we propose a scalable and flexible approximate Bayesian inference technique for OOD detection for deep learning object detectors. Our method offers the ability of uncertainty estimation to distinguish the OOD data from the ID data while preserving their outstanding performance on the ID task for large deep learning models. Our framework provides a flexible uncertainty estimation approach by choosing different layers transformed into Bayesian layers during the model inference stage. Different levels of uncertainties can be chosen to be reserved and estimated.

- Our proposed Bayesian inference technique has been demonstrated effective on OOD detection tasks by comprehensively evaluated on typical OOD detection benchmarks. We show in Section IV that compared to other Bayesian methods, our method can reduce the FPR95 score by up to 8.19% and increase the AUROC score by up to 13.94% on BDD and PASCAL VOC as ID datasets and COCO2017 as the OOD dataset.

The rest of the paper is organized in the following way. Section II provides the background of the OOD detection and Bayesian deep learning methods. In Section III, the proposed Bayesian inference framework is introduced in detail. IV shows the comprehensive experimental results and analysis on OOD benchmarks on both object detection and image classification tasks. The conclusions are presented in Section V.

II. RELATED WORK

A. Out-of-distribution Detection

The goal of out-of-distribution detection, or OOD detection, is to identify input data coming out of a different distribution from the training distribution. OOD detection is an important technique used to help neural networks determine their capability boundary [12]. More specifically, given a labeled training dataset of N data pairs as $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$, where $\mathbf{x}_n$ is an input data sample in the domain $\mathcal{X}$, and $\mathbf{y}_n$ its corresponding target value in the domain $\mathcal{Y}$. The goal of OOD detection is to build a detector f that $f(x_1, ..., x_n) = 1, \forall_i, p(x_i) \geq \delta$ and $f(x_1, ..., x_n) = 0, \forall_i, p(x_i) \leq \delta$ where δ is the capability boundary.

OOD detection for classification can be broadly categorized into two main approaches: generative-based methods [13]–[16] and reconstruction-based methods [17]–[20]. A conceptually appealing approach to OOD detection involves fitting a generative model to a data distribution $p(\mathbf{x}; \theta)$ and evaluating the likelihood of unseen samples under this model. The assumption is that OOD samples will be assigned a lower likelihood than in-distribution samples and can be identified using a simple threshold on this value [21]. Some of the state-of-the-art algorithms are: ensembling [22], ODIN [23], energy score [24], Mahalanobis distance [25], Gram matrices based score [26], and GradNorm score [27].

OOD detection for object detection is currently underexplored [28]–[30]. Energy-based [24] has been proved with both mathematical insights and empirical evidence that the energy score is superior to both a softmax-based score and generative-based methods for OOD detection. Based on directly predicting the conditional target density of an energy-based model, a novel framework for OOD detection by adaptively synthesizing virtual outliers that can meaningfully regularize the model's decision boundary during training known as VOS [28] has been proved effective on OOD detection by modeling the aleatoric uncertainty. As a comparison, our method combines the uncertainty estimation for the classification and localization regression with a novel OOD scoring rule.

B. Bayesian Methods for Probabilistic Object Detection

Different from deterministic neural networks, BNNs aim to provide a natural interpretation of uncertainty estimation in deep learning, by inferring distributions over a network's weights $\theta \in \mathcal{W}$ based on the training dataset $\mathcal{D}$.

BNNs produce the predictive distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ by integrating over all values of network weights:

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \int p(\mathbf{y}|\mathbf{x}, \theta)p(\theta|\mathcal{D})\mathrm{d}\theta, \tag{1}$$

where $p(\mathbf{y}|\mathbf{x}, \theta)$ represents the predicted distribution, and $p(\theta|\mathcal{D})$ represents the weight posterior distribution over the dataset [31].

BNNs can be used for OOD detection by comparing the uncertainty of the model's predictions on a given input to the uncertainty of the model's predictions on a set of known in-distribution data. In order to calculate the approximate solutions of the posterior distribution, various techniques have been proposed [32], [32], [33], [33]–[41]. However, due to the high dimensionality and multi-modality of modern deep neural networks with millions of weight parameters, the posterior distribution is intractable for most of the state-of-the-art object detectors.
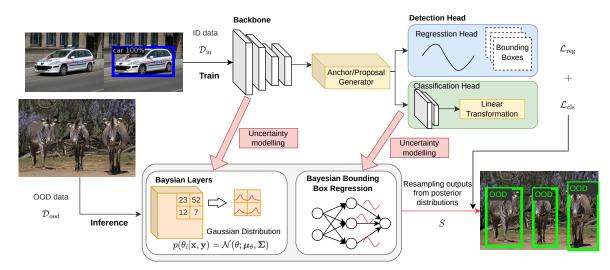
Figure 2: The framework of our proposed Bayesian inference method. The base model is first trained on the ID dataset with the joint object detection loss ($\mathcal{L}_{\text{reg}}, \mathcal{L}_{\text{cls}}$). During the inference stage, we assume the distributions of weight parameters from the chosen layers as class-conditional Gaussians, and sample the results from the low-likelihood region given by the OOD data. The uncertainty estimation produced by the Bayesian layers coming out of the backbone, the classification head or the regression head provides a high uncertainty score S for the OOD data. Combined with the OOD scoring rule with the uncertainty modelling by the Bayesian layers, the OOD data can be separated from the ID data.

Markov Chain Monte Carlo (MCMC) methods are well-recognised methods for inference, including those combined with neural networks, for instance, the Hamiltonian Monte Carlo (HMC) [32]. However, the biggest disadvantage is that requiring full gradients makes HMC computationally intractable for modern neural networks. Later the HMC framework has been extended into stochastic gradient HMC (SGHMC) [33], which allows stochastic gradients to be used in Bayesian inference. As an alternative, stochastic gradient Langevin dynamics (SGLD) [37] employs first-order Langevin dynamics in the stochastic gradient process.

Monte-Carlo Dropout aims to reduce the computational cost for approximating the true posterior distributions [42]. It is a Bayesian inference method implemented based on the regularization techniques with Stochastic Gradient Descent (SGD) during the training stage, which links dropout-based neural network training to Variational Inference (VI) in BNNs. This method is later extended [43] by proving that the optimal dropout rate can be found by treating the dropout probability as a hyperparameter. During test time, samples from the approximate posterior distribution are generated by performing inference multiple times with dropout enabled: $p(\mathbf{y}|\mathbf{x}, \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} p(\mathbf{y}|\mathbf{x}, \theta_t)$. However, recent researchers are questioning whether MC Dropout is a Bayesian method [44]. The implementation of MC Dropout fundamentally depends on the dropout layers being used as the regularization method during the training stage. Since 2015, Batch Normalization (BN) [45] has been applied to normalize the activations of a layer in a neural network by adjusting and scaling the activations to reduce internal covariate shifts. Due to the problem of variance shift problem when using BN and Dropout at the same time [46], BN has been proven as a more effective normalization method, which leads to Dropout being barely used in modern deep neural networks. This makes it difficult to use MC Dropout as a Bayesian inference method in advanced deep-learning models.

Direct Modeling Methods refer to those methods which assume certain probability distributions over the weight parameters in the neural networks, and then directly predict parameters for such distributions through output layers [31]. Some methods aim at modifying the network's output layers and the loss function to model the uncertainty through a single forward pass. A softmax function with a Gaussian distribution is used to replace the standard softmax function with the cross-entropy loss [6] for classification uncertainty quantification. For object detection tasks, some methods focused on modeling the uncertainty coming out of the bounding box regression [47], [48]. Furthermore, several works propose to estimate higher-order conjugate priors in addition to directly predicting the output probability distributions [49]. Stochastic Gradient Descent (SGD) Based Approximations [50] use the iterates of averaged SGD as an MCMC sampler. Stochastic Weight Average Gaussian (SWAG) [40] for Bayesian model averaging and uncertainty estimation proves that the posterior distribution over neural network parameters is close to Gaussian in the subspace spanned by the trajectory of SGD.

III. METHOD

Our proposed Bayesian object detection inference framework is illustrated in Figure 2. One significant advantage of our method is that we can select which layers of the model to sample. Different levels of uncertainties can be represented

by choosing the different numbers of Bayesian layers. The experimental results of choosing different layers to perform OOD detection are shown in section IV.

A. Bayesian Neural Networks for Object Detection

Followed by the definition of the data in section II, the goal of OOD detection for object detection is to provide predictions based on the training data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{b}_i, \mathbf{y}_i\}_{i=1}^{N}$, where $\mathbf{x} \in \mathcal{X}$ denotes the input image, $\mathbf{b} \in \mathbb{R}^4$ represents the bounding box coordinates associated with object instances in the image, and $\mathbf{y} \in \mathcal{Y}$ is semantic labels to its corresponding target for K classes classification. In stead of forming the parameters through maximum a-posterior (MAP) optimization $\hat{\boldsymbol{\theta}}_{\text{MAP}} = \text{argmax}_{\theta} p(\theta|\mathcal{D})$, BNNs aim to produce the predictive classification distribution $p_{\boldsymbol{\theta}}(\hat{\mathbf{y}}|\hat{\mathbf{x}})$ and the bounding box regression distribution $p_{\theta}(\hat{\mathbf{b}}|\hat{\mathbf{y}}, \hat{\mathbf{x}})$ through a Monte Carlo sampling procedure: $p(\hat{\mathbf{y}}|\mathcal{D}, \hat{\mathbf{x}}) \approx \frac{1}{T} \sum_{t=1}^{T} p(\hat{\mathbf{y}}|\theta_l, \hat{\mathbf{x}}), \quad \boldsymbol{\theta}_l \sim p(\theta|\mathcal{D})$, where the $\hat{\mathbf{y}}$ denotes the output predictions of the test input data $\hat{\mathbf{x}}$, and $\hat{\mathbf{b}}$.

While the straightforward idea is to infer the posterior distribution $p(\mathbf{y}|\mathbf{x}, \theta)$ based on the training dataset $\mathcal{D}$ during the training stage. Variational Inference (VI) implies fitting a Gaussian variational posterior approximation over the weights of neural networks, and then approximating the prior distribution through Kullback–Leibler (KL) divergence [35]. VI is later generalized through the reparameterization trick for DNNs [34], [51]. Same as other Bayesian approximation methods introduced in Section II, VI methods are empirically hard to be deployed on deep neural networks with numerous parameters.

B. Weights Sampling from the Bayesian Layers

Instead of training to approximate the posterior distribution $p(\mathbf{y}|\mathbf{x}, \theta)$, our key idea is that, given a pre-trained model, we assume the uncertainty representation of the weight parameters of different Bayesian layers from class-conditional multivariate Gaussian distributions:

$$p(\theta_l|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\theta; \boldsymbol{\mu}_{\theta}, \boldsymbol{\Sigma}) \tag{2}$$

where $\boldsymbol{\mu}_{\theta}$ is the Gaussian mean, $\boldsymbol{\Sigma}$ is the related covariance matrix, and θ_l is weight parameters corresponding to different layers in the neural networks. The ability to model epistemic uncertainty can be achieved by sampling the weights from the proposed prior distribution.

To formulate the parameters of the proposed class-conditional Gaussian distributions, we propose that the sample class means $\hat{\boldsymbol{\mu}}_{\theta}$ are chosen as the values of the weight parameters θ_{pre} from the pre-trained models. The true posterior weight distribution $p(\theta|\mathcal{D}, \omega)$, following data assimilation $\mathcal{D}$, is proportional to the product of the prior weight distribution $p(\theta|\omega)$ and the likelihood $p(\mathcal{D}|\theta)$. Instead of approximation methods such as MCMC [52], a particular weight prior is chosen corresponding to the weight decay ω. Typically, weight decay is used to regularize DNNs, corresponding to explicit L2 regularization during the SGD [53] training process without momentum. When SGD is used with momentum, implicit

regularization still exists, producing a vague prior on the weights of the deep neural network. This regularizer can be given an explicit Gaussian-like form [54], corresponding to a prior distribution on the weights.

Furthermore, we are sampling the weights θ_l from the ϵ-likelihood region of the estimated class-conditional distribution:

$$\theta_l \sim \frac{1}{(2\pi)^{m/2}|\hat{\boldsymbol{\Sigma}}|^{1/2}} e^{\left(-\frac{1}{2}(\theta_l - \hat{\boldsymbol{\mu}}_{\theta})^{\top}\hat{\boldsymbol{\Sigma}}^{-1}(\theta_l - \hat{\boldsymbol{\mu}}_{\theta})\right)} < \epsilon \tag{3}$$

where $\hat{\theta}_l \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{\theta}, \hat{\boldsymbol{\Sigma}})$ denotes the proposed posterior distribution for layer l, which are in the sub-level set based on the likelihood. The hyperparameter ϵ is chosen to be sufficiently small as the sampled weight parameters will not be too far away from the original value causing the potential decrease of the performance of the model.

C. OOD Scoring Rule

An appropriate scoring rule $S(p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{b}))$ is required to distinguish the OOD from the ID data. Scoring rules can be further divided into *local* and *non-local* rules [55]. As a local scoring rule, the negative Log Likelihood (NLL) function [56] evaluates a predictive distribution based on its value only at the true target. NLL measure the quality of predicted probability distributions of a test dataset by: NLL $= -\frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \log p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{D})$, where $\mathbf{x}_n$ is a test data point, and $\mathbf{y}_n$ its corresponding ground truth label. NLL ranges in $(-\infty, +\infty)$, with a lower NLL score indicating a better fitting predictive distribution for that specific ground truth label. However, NLL has its limitation in learning and evaluating bounding box predictive distributions. Energy Score [57] is a proper and non-local scoring rule as an alternative for learning and evaluating multivariate Gaussian predictive distributions. Energy score uses a non-probabilistic energy function to attribute lower values to in-distribution data and higher values to out-of-distribution data. Energy score has been proven efficient compared to the standard softmax score for OOD detection on both image classification and object detection tasks [24], [28]. For object detection tasks, the object-level energy score function [58] can be defined as

$$E(\mathbf{x}, \mathbf{b}; \theta) = -T \log \sum_{k=1}^{K} \exp(f_k((\mathbf{x}, \mathbf{b}); \theta)) \tag{4}$$

where the $f_k((\mathbf{x}, \mathbf{b}); \boldsymbol{\theta})$ is the logit output for class k in the classification head. T represents the temperature parameter of the Gibbs distribution [24]. In this case, it is set as a constant hyperparameter.

During inference, given a test input $\hat{\mathbf{x}}$, the object detector produces a bounding box prediction $\hat{\mathbf{b}}$ through the Bayesian model consists of sampled parameters θ_l. The OOD uncertainty score for the predicted object (x^*, b^*) is given by:

$$S(p_{\theta}(\hat{\mathbf{x}}), \hat{\mathbf{b}}) = \frac{\exp(-\phi(E(\hat{\mathbf{x}}, \hat{\mathbf{b}}; \theta_l)))}{1 + \exp(-\phi(E(\hat{\mathbf{x}}, \hat{\mathbf{b}}; \theta_l)))}. \tag{5}$$

Method	ID: Berkeley DeepDrive-100k			ID: PASCAL-VOC		
	FPR95↓	AUROC↑	mAP(ID)	FPR95↓	AUROC↑	mAP(ID)
Faster-RCNN (Baseline) [2]	82.34	51.39	31.2	70.99	58.37	48.7
MC Dropout [42]	78.64	53.12	31.1	72.19	57.14	48.7
SWAG [40]	68.75	57.41	31.2	65.14	62.45	48.7
BayesOD [47]	72.64	55.84	**32.4**	60.16	64.62	**48.9**
Energy score [24]	60.06	77.48	31.2	56.89	83.69	48.7
VOS [28]	44.27	86.87	31.3	47.53	88.7	**48.9**
Faster-RCNN with Bayesian Layers	73.42±2.31	56.42±1.25	31.2	66.15±2.15	68.63±2.45	48.7
BayesOD with Bayesian Layers	64.15±1.69	59.26±0.93	**32.4**	58.14±2.31	72.12±1.63	**48.9**
Energy Score with Bayesian Layers	56.42±2.08	82.13±1.18	31.2	54.8±1.25	83.99±0.81	48.7
VOS with Bayesian Layers	**43.45±2.21**	**87.25±1.08**	31.3	**46.55±1.11**	**88.75±0.60**	**48.9**

Table I: Main results of the comparison between the proposed method and competitive out-of-distribution detection methods. All methods are implemented based on Faster-RCNN with ResNet-50 as the backbone, and then evaluated on COCO2017 as the OOD dataset. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. **Bold** numbers are superior results. We report standard deviations estimated across 5 runs.

$\phi(\cdot)$ can be a trainable nonlinear function that allows flexible energy surface learning. In this case, $\phi(\cdot)$ is set to be a hyperparameter for inference only.

For OOD detection, ID and OOD objects are distinguished through the following equation:

$$G(\hat{\mathbf{x}}, \hat{\mathbf{b}}) = \begin{cases} 1 & \text{if } S(p_\theta(\hat{\mathbf{x}}), \hat{\mathbf{b}}) \geq \gamma, \\ 0 & \text{if } S(p_\theta(\hat{\mathbf{x}}), \hat{\mathbf{b}}) < \gamma. \end{cases} \quad (6)$$

where $G(\hat{\mathbf{x}}, \hat{\mathbf{b}}) = 1$ implies that the current object is ID, and $G(\hat{\mathbf{x}}, \hat{\mathbf{b}}) = 0$ means the OOD data. The threshold γ is typically chosen so that a high fraction of ID data (e.g., 95%) is correctly classified.

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our proposed Bayesian inference method on a wide range of OOD benchmarks with different base models for both object detection and image classification tasks, respectively in Section IV-A and in Section IV-B. The visualization results are shown in Figure 3.

A. Evaluation on Object Detection

a) Experimental Setup: To evaluate our method, we use PASCAL VOC[1] [1] and Berkeley DeepDrive 100K (BDD-100k[2]) [59] datasets as the ID training data, MS-COCO [3] as the OOD data. PASCAL VOC 2007 and PASCAL VOC 2012 containing 21493 images for train/validation/test data are used. BDD-100k contains 70,000/10,000/20,000 images for train/val/test with 1.8M objects. COCO2017 includes 123,287 images and 886,284 instances.

The models are implemented based on the Detectron2 library [60]. Faster-RCNN with ResNet-50 backbone [61] is used as the base framework for object detection. We choose sample numbers of 1000 for each single test run. The Bayesian layers W_l are chosen to be the 2D Convolutional layers in the main results listed in Table I. Faster R-CNN model with a ResNet-50 backbone has a total of 52 Conv2D layers including 49 Conv2D layers from the ResNet-50 backbone and 3 Conv2D layers from the RPN and detection head. The

impact of selecting different layers to be Bayesian is discussed in Section IV-A0d.

b) Evaluation Metrics: Two evaluation metrics are used to evaluate the performance of proposed models on OOD detection, which are: (1)FPR95, the false positive rate of OOD samples when the true positive rate of ID samples is at 95%; (2)AUROC, computes Area Under the Receiver Operating Characteristic Curve. In addition, mean average precision (mAP) is reported to represent the performance on the ID task.

c) Bayesian Layers are Effective on OOD Detection: Table I shows that in comparison to different pre-trained base models, algorithms with Bayesian layers can provide varying degrees of performance enhancement. Our proposed method is compared with several baselines introduced in Section II, including Faster-RCNN (Baseline) [2], MC Dropout [42], SWAG [40], BayesOD [47], Energy score [24], and VOS [28]. Several popular Bayesian deep learning methods often used as benchmarks are included, as well as outperforming OOD detection algorithms as comparisons to prove the efficiency of inference with Bayesian layers on OOD detection. The comparison precisely highlights the benefits of incorporating Bayesian layers for OOD inference. The experimental results show that all baselines inferred by Bayesian layers have increased OOD detection performance. Results shown in Table III are produced based on the convolutional layers from the backbone and the detection head chosen to be transformed into Bayesian layers.

Compared to each base model, inference with Bayesian layers increases the performance for both experiments with two ID datasets. Some baselines rely on a classification model trained primarily for the ID classification task. Due to the existence of a classification head, these methods can be naturally extended to the object detection model. MC Dropout and SWAG are similar Bayesian methods to our method which is also scalable to deep learning models and

[1]PASCAL-VOC consists of the following ID labels: Person, Car, Bicycle, Boat, Bus, Motorbike, Train, Airplane, Chair, Bottle, Dining Table, Potted Plant, TV, Sofa, Bird, Cat, Cow, Dog, Horse, Sheep.

[2]BDD-100k consists of ID labels: Pedestrian, Rider, Car, Truck, Bus, Train, Motorcycle, Bicycle, Traffic light, Traffic sign.
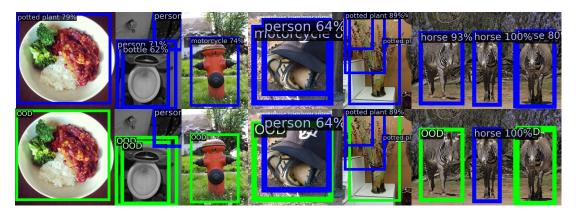
Figure 3: The visualization results of OOD detection. Images are taken from COCO2017. The first row of visualization results is provided by a vanilla Faster-RCNN. The bottom row of visualization results is produced when inference with Bayesian Layers. Models are trained on PASCAL-VOC and evaluated on COCO. Detected objects with blue bounding boxes are classified as one of the ID classes, while objects detected in green bounding boxes indicate that they are recognized as OOD objects.

Inference Layer Selection	FPR95↓ / AUROC↑
Deterministic Faster-RCNN (Baseline)	82.34 / 51.39
Conv2D layers (Backbone) as Bayesian Layers	73.73 / 56.14
Linear layers (Backbone) as Bayesian Layers	75.25 / 54.73
Conv2D layers (Backbone+head) as Bayesian Layers	**73.42 / 56.42**
Linear layers (Backbone+head) as Bayesian Layers	75.82 / 54.12
Full layers as Bayesian Layers	74.12 / 55.73

Table II: FPR and AUROC results of different inference Bayesian layers selections. BDD 100k is the ID training data, COCO is the OOD data.

focus on the posterior distribution approximation during the inference stage. Compared to MC Dropout, inference with Bayesian Layers on Faster-RCNN improved the FPR95 scores by **7.53**% on BDD-100k and **8.19**% on Pascal VOC, and AUROC scores by **4.55**% on BDD-100k and **13.94**% on Pascal VOC. Moreover, our method preserves the high accuracy on the original in-distribution task (measured by mAP) as long as choosing the appropriate baseline.

Most existing OOD detection methods focus on modeling a single type of uncertainty. The epistemic uncertainty comes from the classification branch modeling the classifier or the regression branch modeling the bounding box regression. Other methods, such as VOS, rely on synthesis data outliers to measure the aleatoric uncertainty. Inference with Bayesian layers makes it possible to quantify the uncertainties from the classification head and the regression head at the same time by choosing to transform Bayesian Layers from backbone and detection heads together. This provides a theoretical explanation of the performance improvement of Bayesian layers being applied to all baselines. Inference with Bayesian layers offers a tool to model the epistemic uncertainty coming out of every corner of the network architecture. Combined with the aleatoric uncertainties modeling allows further performance improvement for OOD detection.

d) Impact of Bayesian Layers Selection: To further investigate the impact of choosing different layers to be transformed into Bayesian layers during the Bayesian inference, we test five different architectures on COCO as the OOD dataset when trained on BDD-100k as ID data. Table II shows the mean FPR95 and AUROC results of 3 runs. Specifically, we consider two types of neural network layers: (i) Convolutional Layer: As the primary building block of a CNN. The convolutional layer computes the convolutional operation of the input images using kernel filters to extract fundamental features. The model captures epistemic uncertainties when meeting the OOD data by replacing deterministic weight parameters with the proposed Gaussian distributions. (ii) Linear Layer: A linear layer, also known as a fully connected layer. It is capable of learning an average rate of correlation between the output and the input without a bias. Linear layers are frequently modified by Bayesian methods since they contain limited parameters leading to less training and computational inference cost.

B. Evaluation on Image Classification

We demonstrate that Bayesian Layers are also effective at common image classification benchmarks beyond object detection. In this case, CIFAR-10 [62] with standard train/val splits is used as the ID training data. WideResNet-40 [63] is chosen to be the base neural network architecture. For image classification, the cross-entropy loss is used as the training objective. All models are evaluated on six OOD datasets: `Textures` [64], `SVHN` [65], `Places365` [66], `LSUN-C` [67], `LSUN-Resize` [67], and `iSUN` [68]. The comparisons are shown in Table III, with results averaged over six test datasets. We demonstrate that inferencing with Bayesian layers for OOD detection has different levels of performance improvement compared to each benchmark, without sacrificing the ID test classification accuracy (94.84% on pretrained WideResNet) and extra training requirement.

Base model $\mathcal{D}_{in}^{test}$	Method	FPR95 ↓	AUROC ↑
	WideResNet (Baseline)	51.04	90.90
	Bayesian Neural Networks	53.04	92.45
	MC dropout	54.32	88.90
WideResNet	SWAG	38.25	92.91
	Energy score	33.01	91.88
CIFAR-10	Vitual Outliers	24.87	94.06
	Baseline + Bayesian Layers (Ours)	**48.25 $\pm$ 0.91**	**92.27 $\pm$ 2.24**
	Energy Score + Bayesian Layers (Ours)	**32.33 $\pm$ 0.46**	**93.52 $\pm$ 1.82**
	VOS + Bayesian Layers (Ours)	**23.68 $\pm$ 0.52**	**94.36 $\pm$ 0.93**

Table III: OOD detection results of the proposed Bayesian methods on image classification. All models listed are built based on the WideResNet-40 architecture. All models are first trained on CIFAR-10 dataset and then evaluated on several OOD datasets. The mean PR95 and AUROC results are given.

V. CONCLUSION

In this work, we propose a novel and scalable Bayesian deep learning inference framework for OOD detection. Transforming the deterministic deep neural layers into Bayesian layers by replacing the weights parameters trained by point estimates with assumed Gaussian distributions is able to provide the uncertainty estimation during the inference stage. We show that inference with superficial Bayesian layers can improve the performance of various benchmarks for OOD detection. Unlike other Bayesian deep learning methods, our proposed method does not require extra training, making it scalable for modern advanced deep-learning-based object detectors. For future work, we would like to explore the impact on different assumed distributions for Bayesian layers and the uncertainty quantification method for Bayesian layers. In addition, our approach can be valuable to other machine learning tasks than object detection or image classification. We hope future research will increase attention toward a broader view of OOD uncertainty estimation from a Bayesian layers perspective.

REFERENCES

[1] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions Pattern Analysis and Machine Intelligence.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of Eur. Conf. Computer Vision*, 2014.

[4] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv preprint arXiv:2110.11334*, 2021.

[5] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[6] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[7] R. M. Neal, "Bayesian learning for neural networks," Ph.D. dissertation, University of Toronto, 1995.

[8] R. Li, C. Zhang, H. Zhou, C. Shi, and Y. Luo, "Out-of-distribution identification: Let detector tell which I am not sure," in *Computer Vision – Proceedings of ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 638–654.

[9] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li *et al.*, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," *arXiv preprint arXiv:2211.05778*, 2022.

[10] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," *CoRR*, vol. abs/2012.07177, 2020. [Online]. Available: https://arxiv.org/abs/2012.07177

[11] R. Dagli and A. M. Shaikh, "Cppe-5: Medical personal protective equipment dataset," 2021.

[12] M. S. Graham, W. H. L. Pinaya, P.-D. Tudosiu, P. Nachev, S. Ourselin, and M. J. Cardoso, "Denoising diffusion models for out-of-distribution detection," 2022. [Online]. Available: https://arxiv.org/abs/2211.07740

[13] H. Choi and E. Jang, "Generative ensembles for robust anomaly detection," *arXiv preprint arXiv:1810.01392*, 2018. [Online]. Available: http://arxiv.org/abs/1810.01392

[14] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Proceedings of Advances in Neural Information Processing Systems*, 2019, pp. 14 680–14 691.

[15] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, "Input complexity and out-of-distribution detection with likelihood-based generative models," in *International Conference on Learning Representations*, 2020.

[16] Z. Xiao, Q. Yan, and Y. Amit, "Likelihood regret: An out-of-distribution detection score for variational auto-encoder," *Proceedings of Advances in Neural Information Processing Systems*, vol. 33, 2020.

[17] O. Lyudchik, "Outlier detection using autoencoders," Tech. Rep., 2016.

[18] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal processing*, vol. 99, pp. 215–249, 2014.

[19] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance," *arXiv preprint arXiv:1812.02765*, 2018.

[20] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proceedings of International Conference on Learning Representations*, 2018.

[21] C. M. Bishop, "Novelty detection and neural network validation," *IEE Proceedings-Vision, Image and Signal processing*, vol. 141, no. 4, pp. 217–222, 1994.

[22] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proceedings of Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[23] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations, ICLR 2018*, 2018.

[24] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Proceedings of Advances in Neural Information Processing Systems*, 2020.

[25] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proceedings of Advances in Neural Information Processing Systems*, 2018, pp. 7167–7177.

[26] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with gram matrices," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, vol. 119, 2020, pp. 8491–8501.

[27] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," in *Proceedings of Advances in Neural Information Processing Systems*, 2021.

[28] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021*, 2021.

[29] C. Huang, V. D. Nguyen, V. Abdelzad, C. G. Mannes, L. Rowe, B. Therien, R. Salay, and K. Czarnecki, "Out-of-distribution detection for lidar-based 3d object detection," in *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 4265–4271.

[30] M. A. Santana, R. Calinescu, and C. Paterson, "Risk-aware real-time object detection," in *Proceedings of the 2022 18th European Dependable Computing Conference (EDCC)*, 2022, pp. 105–108.

[31] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9961–9980, 2022.

[32] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[33] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *Proceedings of Int. Conf. Machine Learning*, 2014, pp. 1683–1691.

[34] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of Int. Conf. Machine Learning*, 2015.

[35] A. Graves, "Practical variational inference for neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2011.

[36] S. Depeweg, J. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning," in *Proceedings of Int. Conf. Machine Learning*, 2018.

[37] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proceedings of Int. Conf. Machine Learning*, 2011.

[38] X. Chen, J. D. Lee, X. T. Tong, Y. Zhang *et al.*, "Statistical inference for model parameters in stochastic gradient descent," *The Annals of Statistics*, vol. 48, no. 1, pp. 251–273, 2020.

[39] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate bayesian inference," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4873–4907, 2017.

[40] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for bayesian uncertainty in deep learning," in *Proceedings of Advances in Neural Information Processing Systems*, 2019.

[41] H. Ritter, A. Botev, and D. Barber, "A scalable laplace approximation for neural networks," in *6th International Conference on Learning Representations (ICLR)*, 2018.

[42] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[43] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[44] L. L. Folgoc, V. Baltatzis, S. Desai, A. Devaraj, S. Ellis, O. E. M. Manzanera, A. Nair, H. Qiu, J. A. Schnabel, and B. Glocker, "Is MC dropout bayesian?" *CoRR*, vol. abs/2110.04286, 2021. [Online]. Available: https://arxiv.org/abs/2110.04286

[45] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*. PMLR, 2015, pp. 448–456.

[46] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 2677–2685.

[47] A. Harakeh, M. Smart, and S. L. Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," in *Proceedings of IEEE Int. Conf. Robotics and Automation*, 2020.

[48] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, 2019.

[49] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep evidential regression," *arXiv preprint arXiv:1910.02600*, 2019.

[50] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," *CoRR*, vol. abs/1708.02182, 2017. [Online]. Available: http://arxiv.org/abs/1708.02182

[51] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," *Proceedings of Advances in Neural Information Processing Systems*, vol. 28, 2015.

[52] C. Rasmussen, "A practical monte carlo implementation of bayesian learning," *Advances in Neural Information Processing Systems*, vol. 8, 1995.

[53] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[54] ——, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[55] A. Harakeh and S. L. Waslander, "Estimating and evaluating regression predictive uncertainty in deep object detectors," in *Proceedings of International Conference on Learning Representations*, 2021.

[56] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[57] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[58] W. Liu, X. Wang, J. D. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proceedings of Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html

[59] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 2020, pp. 2633–2642.

[60] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[62] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," vol. Technical Report, 2009.

[63] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[64] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, 2014, pp. 3606–3613.

[65] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proceedings of NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[66] B. Zhou, À. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions Pattern Analysis and Machine Intelligence.*, vol. 40, no. 6, pp. 1452–1464, 2018.

[67] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[68] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," *arXiv preprint arXiv:1504.06755*, 2015.