Leveraging Bayesian Optimizer-Trained Models for Enhanced Feature Extraction and Out-of-Distribution Detection

A DKU Spring 2023 ECE590K-002 Final Project Report

Yike Guo
DNAS
Duke Kunshan University
Kunshan, China
yike.guo@duke.edu

Yueqian Lin
DNAS
Duke Kunshan University
Kunshan, China
yueqian.lin@duke.edu

Zhixian Zhang
DNAS
Duke Kunshan University
Kunshan, China
zhixian.zhang688@duke.edu

***Abstract*—Uncertainty computation in deep learning is essential for designing robust and reliable systems, which must consider performance measures beyond standard test set accuracy. The problem of detecting out-of-distribution (OOD) is increasingly popular these days due to model safety and robustness concerns. In this work, we adopt a novel approach to train the feature extraction model using a Bayesian-based optimizer. A new data augmentation technique that combines style transfer and pixel-wise picture mixing is also proposed and tested. Several OOD detection methods based on feature extraction are used to prove the feasibility of the Bayesian optimizer.**

***Keywords*—Out-of-distribution detection, Bayesian, Data augmentation, Rectified activations, K-nearest neighbors**

I. Introduction

The concept of out-of-distribution (OOD) detection has been investigated across various fields, such as statistics, machine learning, and deep learning. In scholarly discourse, OOD arises when a model is tested on data that significantly deviates from the data on which it was trained, due to factors such as changes in the underlying data distribution, noise, or intentionally deceptive data. Detecting OOD samples is essential to ensure the reliability and safety of machine learning models in real-world scenarios, as they may produce unreliable or potentially harmful predictions.

As illustrated in Figure 1, a well-trained model for classifying closed-world data with matched training and testing distributions may struggle with open-world data featuring entirely different distributions. In computer vision, Deep Neural Networks (DNNs) have significantly advanced visual recognition systems over the past decade but still fall short of achieving human-level performance in real-world environments, primarily due to their vulnerability to OOD scenarios. These scenarios can involve objects with unusual poses, textures, or shapes, or challenging weather conditions and atypical contexts. Turing Award winners Yoshua Bengio, Geoffrey Hinton, and Yann LeCun have acknowledged this lack of robustness as a core open problem in deep learning [1], which remains largely unsolved despite considerable progress in the field.

The authors thank Prof. Kaizhu Huang and TA Zhiqiang Gao for their support and guidance. The code is available at https://github.com/linyueqian/ECE590K_OOD_Public.

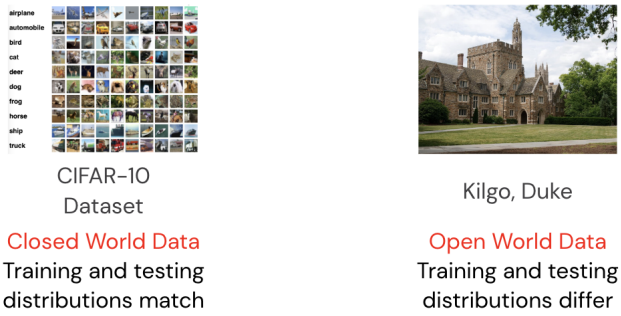

Fig. 1. Real World Example

In the context of machine learning, various methods have been developed to detect OOD samples, including classification-based [2], density-based [3], distance-based [4], and reconstruction-based approaches [5]. Despite their widespread use and remarkable performance in controlled settings, DNNs are often not robust to shifts in the data-generating distribution, resulting in untrustworthy predictions in real-world applications, particularly in safety-critical environments.

In supervised learning settings, the input space is represented by $X = \mathbb{R}^d$, while the label space is denoted by $Y = 1, 2, \ldots, K$. The training dataset is obtained from a joint distribution, $P = X \times Y$, which is assumed to be drawn independently and identically distributed (i.i.d). A neural network is then trained using this dataset, represented by the function $f(x; \theta)$. During testing, however, data may be sourced from a distribution that is different from the training dataset. This out-of-distribution (OOD) data possesses a label set that does not interact with Y and should not be predicted by the model.

Consequently, OOD detection can be formulated as a binary classification problem,

$$G(\mathbf{x}; f) = \begin{cases} 0 & \text{if } \mathbf{x} \sim D_{\text{o.o.d}} \\ 1 & \text{if } \mathbf{x} \sim D_{\text{i.d.}} \end{cases}, \quad (1)$$

where "o.o.d" and "i.d" represent out-of-distribution and in-distribution data, respectively.

Although deep learning has been applied widely in detecting OOD samples, it has some issues which make its application difficult in some fields. For example, it is more likely to overfit when the dataset is small, and it lacks reliable confidence estimates [6]. More importantly, the structure of DNNs neglects the information the data contains and fails to utilize data geometry. However, data geometry is especially crucial for OOD detection. As Bayesian principles exploit the information of data by using Bayes' rule, it has the potential to deal with such issues and the application of Bayesian principles is more likely to improve uncertainty on OOD samples. The use of Bayesian principles for neural networks has been proposed in the 90s, such as MCMC methods [7], Laplace's method [8], and variational inference (VI) [9]–[11], and benefits of Bayesian method have been widely discussed in famous machine-learning books [12], [13]. However, it is rarely used in practice due to computational expenses.

The main difficulty is the computation of posterior distribution and it has been challenging even for approximation methods, such as VI and MCMC, to cope with large datasets such as ImageNet [14]. However, recently proposed natural-gradient VI has been proven to be practical in deep learning [15], and in our work, we demonstrate that Bayesian principles can be applied in the training process as the optimizer of deep learning for OOD detection.

In this report, our key results and contributions are:

1) We present an effective model to solve OOD detection by incorporating the Bayesian optimizer (VOGN) to extract features, Style Transfer Mix as image enhancement, and rectified activations to reduce overconfidence issues.
2) For the training process, on LeNet5, we use VOGN featuring Bayesian principles as an optimizer and demonstrate that it can improve the uncertainty on out-of-distribution data compared with conventional optimizers such as Adam.
3) For image enhancement, we propose a new data augmentation technique called *StyleMix* combining style transfer learning and pixel mixing. We show that the new data augmentation skill can increase the model's accuracy.
4) For addressing overconfidence issues, we incorporate Rectified Activations into our model based on their effectiveness in enhancing the performance of OOD detection, attenuating the overconfidence activations by rectifying the activations at an upper limit $c > 0$.

II. Methods

A. Data Augmentation

As stated in [16], data augmentation skills can improve the performance of the model and expand the limited dataset. To increase the model's accuracy in predicting the in-distribution and out-of-distribution data, some common data augmentation techniques can be used, including rotation, flipping, scaling, cropping, translation, and adding noise. These techniques can be applied to both the input data and the output labels to generate new data samples. For example, in image classification tasks, flipping and rotating the images can create new samples that are still valid for the task.

Another useful technique is transfer learning, which involves using pre-trained models on large datasets to improve the accuracy of a smaller dataset. This can be particularly useful when the available dataset is small or limited, as the pre-trained model can help to generalize to new data and reduce overfitting. In this report, we explore the idea of style transfer and mixing pictures pixel-wise.

The authors of [17] propose a new data augmentation technique called "style augmentation" that uses random style transfer to improve the robustness of CNNs for classification and regression tasks. The technique randomizes texture, contrast, and color while preserving the shape and semantic content. The authors show that data augmentation significantly improves robustness to domain shift and can be used as a simple, domain-agnostic alternative to domain adaptation. An example of the result of the style transfer is shown in Figure 2.

Fig. 2. Style Augmentation Example

We first tested the original Style Augmentation on the CIFAR-10 [18] dataset, which, however, is not increasing the accuracy of the OOD detection as the transferred style appears to be a bit too much change for a 32×32 picture. We then argue that style augmentation can be combined with traditional augmentation techniques to improve network performance. We adopt the algorithm listed in [19] to realize the combination. The algorithm involves mixing an original image with either an augmented version of the original or a randomly selected image from a mixing set, using randomly selected mixing operations such as additive or multiplicative (as shown in Figure 3). The mixing process is repeated a random number of times up to a maximum of k. The mixing set in the original paper includes fractals and feature visualization pictures, but we use the transferred picture instead. The algorithm also includes an augmentation function that randomly applies various operations such as rotate, solarize, and posterize to the input image. The resulting image is returned as the output of the algorithm.

A pseudo-code of our proposed algorithm is shown in Figure 4.

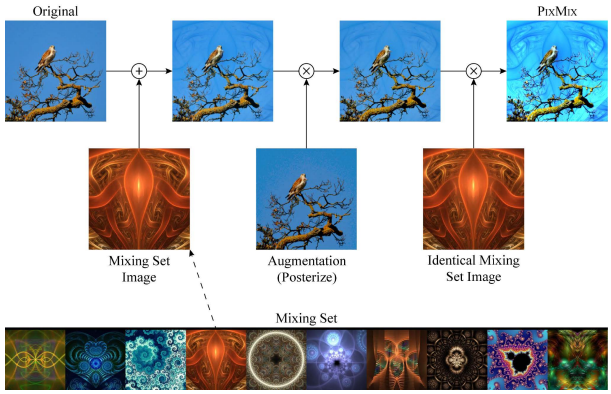

Fig. 3. Pixel Mixing Pipeline

B. Bayesian Optimizer Model

In this project, we apply the natural-gradient VI method (VOGN), proposed by Khan et al. [20]. Although VI is a classic Bayesian estimation method, previous VI methods, proposed by Graves [21], require considerable efforts in tuning and have slow optimization process. VOGN addresses this issue by combining VI and natural-gradient updates [22], which offers a flexible alternative to applying Bayesian principles in deep learning.

1) VI and natural-gradient updates: VI approximates exact Bayesian inference by learning parameters of $q(\theta)$, the best approximation of the true posterior $p(\theta|D)$. The posterior is given by:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \tag{2}$$

where D represents the dataset, $(D|\theta)$ is the likelihood, and $p(\theta)$ is the prior of parameters. By assuming $q(\theta)$ as an exponential family distribution, VI maximizes the Evidence Lower Bound (ELBO) to obtain optimized parameters η:

$$L_{ELBO(\eta)} = E_{q_\eta \theta}[logp(D|\theta] - KL[q(\theta)||p(\theta) \tag{3}$$

The goal of natural-gradient updates is to optimize $L(\eta)$ (can be some loss function in DNN) with respect to η by taking gradient steps as follows:

$$\eta_{t+1} = \eta_t + \beta_t(\eta_t)^{-1}\nabla_\eta \mathcal{L}(\eta_t), \tag{4}$$

where $F(\eta_t)$ is the Fisher information matrix that contains the information geometry of the distribution being optimized, and β_t is the learning rate. The form of the natural-gradient update is similar to the gradient descent in neural networks as described below:

$$\eta_{t+1} = \eta_t + \beta_t \nabla_\eta \mathcal{L}(\eta_t), \tag{5}$$

This shows that the natural-gradient update takes advantage of the information on the distribution by using the Fisher information matrix, which makes it more advanced than the simple gradient descent method. Furthermore, by exploiting features of exponential family distribution, we can simplify the natural-gradient step as:

$$\eta_{t+1} = \eta_t + \beta_t \nabla_{\mathbf{m}} \mathcal{L}_* (\mathbf{m}_t), \tag{6}$$

where $\mathcal{L}_* (\mathbf{m}_t)$ is the same function as $\mathcal{L}(\eta)$ except written in terms of the mean parameters m. From exponential family distribution, $\mathbf{m} = \mathbb{E}_{q_\eta(\theta)}[\phi(\theta)]$, where $\phi(\theta)$ is the sufficient statistics and therefore $\mathbf{m} = \nabla_\eta A(\eta)$.

2) Natural-gradient VI: Now we combine the natural-gradient update and VI by plugging the ELBO (Equation (3)) into the natural-gradient update (Equation (6)). Referred to the appendices in [20], let the prior $p(\theta)$ be an exponential family distribution with natural parameters η_0, then KL term in the ELBO can be simplified as:

$$\nabla_{\mathbf{m}} \text{KL term} = \eta_0 - \eta \tag{7}$$

After putting it back to ELBO, we obtain:

$$\eta_{t+1} = (1-\beta_t)\,\eta_t + \beta_t \left(\eta_0 + \nabla_{\mathbf{m}} \mathbb{E}_{q_{\eta_t}(\theta)}[\log p(\mathcal{D} \mid \theta)]\right), \tag{8}$$

which is presented in detail in [23], which is called "Bayesian learning rule".

3) VOGN: From Equation (6), we obtain:

$$\mathcal{F}_t = \nabla_{\mathbf{m}} \mathbb{E}_{q_{\eta_t}(\theta)}[\log p(\mathcal{D} \mid \theta)] \tag{9}$$

In this section, we now consider a Gaussian approximating family, $q_\eta(theta) = N(\theta; \mu, \Sigma)$ and substitute parameters into Equation (6) to obtain updates of μ and Σ by replacing natural parameters η with μ and σ:

$$\begin{aligned} \boldsymbol{\eta}^{(1)} &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \quad \boldsymbol{\eta}^{(2)} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}, \\ \mathbf{m}^{(1)} &= \boldsymbol{\mu}, \quad \mathbf{m}^{(2)} = \boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma}. \end{aligned} \tag{10}$$

Let the prior be a zero-mean Gaussian, $p(\theta) = N(\theta; 0, \delta^{-1}\mathbf{I})$, and then prior of natural parameters can be written as

$$\eta_0^{(1)} = \mathbf{0}, \eta_0^{(2)} = -\frac{1}{2}\delta\mathbf{I} \tag{11}$$

We now simplify $\nabla_m \mathcal{F}_t$ to be with respect to μ and Σ instead of m by using chain rules, following Appendix B.1 in [24]. Therefore, we obtain:

$$\Sigma_{t+1}^{-1} = (1-\beta_t)\Sigma_t^{-1} + \beta_t(\delta\mathbf{I} - 2\nabla_\Sigma \mathcal{F}_t). \tag{12}$$

$$\mu_{t+1} = \mu_t + \beta_t \Sigma_{t+1}(\nabla_\mu \mathcal{F}_t - \delta\mu_t). \tag{13}$$

To deal with $\nabla_\mu \mathcal{F}_t$ and $\nabla_\Sigma \mathcal{F}_t$ terms, we use Bonnet's and Price's theorems to express equations in terms of the gradient $g(\theta)$ and Hessian of the negative log-likelihood $H(\theta)$, and apply Gauss-Newton matrix to approximate Hessian following [20]. Therefore, we obtain the ultimate algorithm of VOGN:

$$\mu_{t+1} = \mu_t - \alpha_t \frac{\hat{g}(\theta_t) + \tilde{\delta}\mu_t}{s_{t+1} + \tilde{\delta}} \tag{14}$$

$$s_{t+1} = (1-\beta_t)s_t + \beta_t \frac{1}{M} \sum_{i \in \mathcal{M}_t} \left(g_i(\theta_t)^2\right), \tag{15}$$

where $\mathbf{s}_t = (\Sigma_t^{-1} - \delta\mathbf{I})/N$ which makes the calculation simpler.

```python
def style_transfer_mix(img, beta=0.8):
    # Resize the input image to 256x256
    resized_img = cv2.resize(img, (256, 256), interpolation=cv2.INTER_CUBIC)

    # Apply style transfer to the resized image
    stylized_img = style_transfer(resized_img)

    # Mix the stylized image and the original image using the PIXMIX algorithm
    mixed_img = pixmix(stylized_img, img, beta=beta)

    # Return the mixed image
    return mixed_img
```

Fig. 4. The `style_transfer_mix` function in Python.

C. Rectified Activations

Rectified activation is a simple but very effective method for enhancing the performance of out-of-distribution detection [25]. Figure 5 shows the output of the penultimate layer of ResNet. The solid line shows the mean value and the shaded area shows the standard deviation. The mean of the ID data is higher than the OOD data, but the variance of the OOD data are higher than the ID data. As a result, such a high value of output can undesirably impact the confidence of the OOD data, leading to some over confidence of OOD data. The above observation yields an simple and effective method of clipping the activations above a threshold. To be specific, the over confidence activations can be attenuated by rectifying the activations at an upper limit $c > 0$. What worth noting is that this process can be done without any modifications to the pre-trained model.

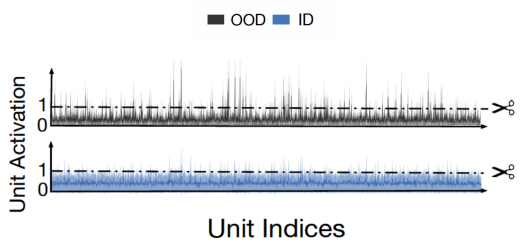

Fig. 5. Distribution of penultimate layer of ID and OOD data

For instance, we consider a pre-trained neural network model parameterized by , which encodes an input space $\mathbb{R}^d$ to a feature space with dimension m. This feature extractor is denoted by $h_\theta(\mathbf{x}) \in \mathbb{R}^m$, which ends up with the penultimate layer of the network. The weight matrix that maps $h(\mathbf{x})$ to the output logits $f(\mathbf{x})$ is denoted by $\mathbf{W} \in \mathbb{R}^{m \times K}$, where K is the total number of classes in the ID dataset. The rectified activation of the penultimate layer of the network works as follows:

$$\bar{h}(\mathbf{x}) = min(h(\mathbf{x}), c), \tag{16}$$

As a result, the operation obliterates the activations above the threshold c, to mitigate the effect of overconfidence of the OOD data. The output logits of the model after the rectified activation is:

$$f^{\text{ReAct}}(\mathbf{x}; \theta) = \mathbf{W}^\top \bar{h}(\mathbf{x}) + \mathbf{b}, \tag{17}$$

where $\mathbf{b} \in \mathbb{R}^K$ is the bias term. When the threshold $c = \infty$, it means the output is not influenced by the rectified layer. In practice, we choose the c according to the value that 90% of the features of ID data is not truncated by the rectified activation.

During inference time, the output after the rectified activation can be leveraged by various postprocessors by the following criteria:

$$G_\lambda\left(\mathbf{x}; f^{\text{ReAct}}\right) = \begin{cases} \text{in} & S\left(\mathbf{x}; f^{\text{ReAct}}\right) \geq \lambda \\ \text{out} & S\left(\mathbf{x}; f^{\text{ReAct}}\right) < \lambda \end{cases}, \tag{18}$$

where $S(\mathbf{x}; f)$ is the scoring function. The threshold λ is chosen based on the resulting high fraction of ID data is correctly classified (i.e. 95%).

D. Out-of-Distribution Post Processors

In this section, we provide a brief overview of post-hoc methods, focusing on their general principles and popular techniques such as k-Nearest Neighbors (KNN), energy-based methods, and maximum softmax probability. Post-hoc methods for OOD detection, also known as post-processors, aim to identify OOD samples by analyzing the features or outputs generated by a pre-trained classifier network. These methods do not require modification to the network architecture or retraining, making them a flexible and efficient option for OOD detection. Post-hoc techniques rely on the assumption that the feature embeddings or output scores of OOD samples are significantly different from those of in-distribution (ID) samples.

1) The K-Nearest Neighbors (KNN) approach: The authors of [26] propose a deep k-Nearest Neighbor (KNN) approach for detecting out-of-distribution (OOD) samples. This method falls under the category of distance-based techniques, which

assumes that OOD samples are located far away from in-distribution (ID) data in the feature embedding space. Unlike previous distance-based methods that relied on parametric density estimation and assumed multivariate Gaussian distributions, the authors suggest a non-parametric density estimation using nearest neighbors for OOD detection.

Although the KNN approach is relatively straightforward, it has not been thoroughly explored in the context of OOD detection. To determine whether an input is OOD, the method calculates the k-th nearest neighbor distance between the embedding of each test image and the training set, using a simple threshold-based criterion. Following the approach in rectified activations, the method uses the normalized penultimate feature $z = \frac{\phi(x)}{|\phi(x)|_2}$ for OOD detection, where ϕ is the feature encoder.

During testing, the method obtains the normalized feature vector z for a test sample x and computes the Euclidean distances $||z_i - z^*||_2$ with respect to the embedding vectors $z_i \in \mathbb{Z}_n$. The data sequence $\mathbb{Z}_n$ is then reordered based on the increasing distance $||z_i - z^*||_2$, and the reordered sequence is denoted as $\mathbb{Z}'_n = (z^{(1)}, z^{(2)}, \ldots, z^{(n)})$. The OOD detection decision function is given by:

$$G\left(z^*; k\right) = 1\left\{-r_k\left(z^*\right) \geq \lambda\right\}, \tag{19}$$

where $r_k(z^*) = ||z^- z^{(k)}||_2$ is the distance to the k-th nearest neighbor and $1\cdot$ is the indicator function. The threshold λ is selected to ensure that a high proportion of ID data (e.g., 95%) is correctly classified. This threshold does not depend on OOD data.

The KNN approach provides a more adaptable and potentially more precise method for detecting OOD samples because it does not rely on strong distributional assumptions about the learned feature space.

E. Other approaches

Energy-based methods, on the other hand, focus on the energy values computed from the feature embeddings. These methods assume that OOD samples have higher energy values compared to ID samples, as the classifier has not seen these inputs before. By computing energy values for each input and comparing them against a threshold, energy-based methods can effectively distinguish between OOD and ID samples.

Another common post-hoc approach is analyzing the maximum softmax probability (MSP) of the classifier's output. Under this method, it is expected that the MSP for OOD samples will be lower than that of ID samples, as the classifier should be less confident in its predictions for unfamiliar data. By comparing the MSP values against a threshold, the method can identify OOD samples.

III. Experiments

A. Experimental Setup

In this experiment, we consider the problem of in-distribution classification on the CIFAR-10 [18] dataset and out-of-distribution (OOD) classification on three other datasets, namely SVHN [27], MNIST, [28] and Texture [29]. The feature extraction model comprises two architectures: LeNet [30] and ResNet [31], which are popular deep-learning models for image classification tasks. We use two optimization algorithms, namely Adam and Bayesian optimizers (using VOGN algorithm), to train the model seperately. We evaluate the performance of the models using three different methods introduce in the previous section: K Nearest Neighbors, Energy-based, and Maximum Softmax Probability. We also use data augmentation techniques, specifically Style Transfer Mix, to augment the dataset and improve model performance. We trained the models on a high-performance computing system equipped with an RTX TITAN GPU with 24G of memory. The experiment aims to investigate the robustness and generalization of the model against OOD samples. We report the performance of the models using standard metrics, including accuracy, precision, recall, and F1-score, and analyze the results to draw insights into the suitability of the models for real-world applications. Overall, this experiment provides a comprehensive evaluation of different techniques for in-distribution and OOD classification tasks on image datasets.

B. Metrics

In the context of Out-of-Distribution (OOD) detection, several performance metrics are commonly used. The False Positive Rate at 95% True Negative Rate (FPR95) measures the percentage of out-of-distribution samples that are misclassified as in-distribution at a fixed True Negative Rate of 95%. The Area Under the Receiver Operating Characteristic curve (AUROC) is a common metric for binary classification problems, and measures the trade-off between the True Positive Rate and the False Positive Rate. The Area Under the Input-dependent Confidence Score curve (AUIN) measures the confidence of the model's prediction for a given input, and is used to evaluate the model's ability to detect samples that are out-of-distribution.

Additionally, other metrics such as the False Negative Rate (FNR), Accuracy (ACC), and End-to-end Accuracy (End-to-end ACC) are also used to evaluate the performance of OOD detection methods. The FNR measures the percentage of in-distribution samples that are misclassified as out-of-distribution, while the ACC measures the percentage of correctly classified in-distribution samples. The End-to-end ACC takes into account both the accuracy of the original model and the effectiveness of the OOD detection method in detecting out-of-distribution samples. The formula for end-to-end accuracy is $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, where TP is the number of true positives, and FP is the number of false positives. Together, these performance metrics provide a comprehensive evaluation of the model's ability to classify and detect inputs and can be used to guide the development and improvement of OOD detection methods.

Similar to literature [34], we also use predictive entropy as one metric to test the out-of-distribution performance. Predictive entropy is given by $\sum_{k=1}^{K} -p_{ik}\log(p_{ik})$, where k represents the number of classes of the dataset and i represents the image index. Ideally, as the entropy represents the stability

of the model, a model with high entropy indicates that it is unsure about which class images belong to. For OOD detection, a good model will have more examples with lower entropy on in-distribution data, which indicates that the model is confident in the classes of input images. In contrast, on out-of-distribution data, a good model will have more examples with lower entropy, which shows that the model is uncertain of the inputs.

C. Main Results and Analysis

Firstly, We present the results of the data augmentation in Table I. With the same OOD detection method and model architecture, our proposed Style Mix Transfer reduces the average FPR95 by 3.56% and AUROC by 0.86%, compared to the performance of the same OOD detection method and model architecture without data augmentation. The corresponding end-to-end accuracy also increases 0.08%.

We also show the predictive entropy histograms to compare the performance of Adam and Bayesian optimizers, illustrated in Figure 6 and Figure 7. Ideally, we want the predictive entropy to be high on out-of-distribution data and to be low on in-distribution data. On LeNet5 with batch normalization, the Bayesian optimizer shows the desired result: entropies are lower in in-distribution data and higher in out-of-distribution data compared with Adam.

To further prove our results, we further explain below four instances as examples. Figure 8 shows an ID instance that both Adam and Bayesian classify correctly as ID data, but the confidence score output by Adam (0.4489) is much lower than Bayesian (1.4416). This indicates that Bayesian method is more confidence on in-distribution data, which is up to our expectation. Figure 9 shows an ID instance where Adam mistakenly categorizes it as OOD with confidence score = 0.6325 but Bayesian correctly classifies it as ID with the confidence score = 0.41. Figure 11 is an OOD instance that both Adam and Bayesian correctly classify. However, Adam has higher confidence score (1.6029) than Bayesian method (0.7948). Since lower confidence score in out-of-distribution data represents higher uncertainty, this demonstrates that Bayesian method can improve the uncertainty on out-of-distribution samples. Figure 10 is an OOD instance that Adam mistakenly classifies it as ID with confidence score = 1.4615 while the Bayesian correctly classifies it as OOD data with confidence score = 0.3577. All four instances show that the Bayesian optimizer can identify in-distribution data with more confidence and recognize out-of-distribution data with improved uncertainty, which outperforms the Adam optimizer in performance.

For the effect of rectified layer, we can see from Table III that adding a ReAct layer to the Lenet5 model has positive effect on the FPR95 and AUROC value in OOD detection. Meaning that the ReAct method indeed mitigates the negative effect of overconfidence regarding to OOD samples.

IV. Discussions

In this study, we introduce a novel data augmentation technique, Style Transfer Mix, which demonstrates marginally improved performance compared to the original dataset without any data augmentation. It is crucial to acknowledge that the instances within our utilized dataset consist of 32×32 images, potentially limiting the capacity to fully exhibit the advantages of incorporating style transfer. We expect that employing a more sophisticated model and a dataset with real-world settings will yield enhanced performance outcomes.

Regarding Bayesian optimizer (VOGN), in our project, we directly apply the model proposed by [12] and the implementation structure presented by [15]. However, the Bayesian optimizer method advocated by previous research is aimed at solving common optimization problems and can be applied to all cases related to deep learning, and it is not targeted at solving OOD detection problems. Our project further explores the availability and efficiency of Bayesian principles applied to OOD detection by comparing the performance of Bayesian and Adam optimizers and therefore provides strong support for the proposal of previous related research.

Nevertheless, as mentioned before, the Bayesian optimizer applied in our project is not OOD-oriented. For future work, we reframe the Bayesian optimizer method by looking for representative losses that can better measure the difference between distributions, such as Wasserstein loss.

We observe that OOD samples can cause unusual activation patterns in neural networks, and the effectiveness of the rectified activation method in mitigating this issue is demonstrated. However, further research is necessary to fully understand the underlying reasons for this phenomenon. Our study highlights the advantages and disadvantages of BatchNorm, which can result in unusually high unit activations when applied to out-of-distribution data without adjustments. In future work, we suggest analyzing the neural network's training and evaluation mechanisms to provide a better explanation for the observed activation patterns in Figure 5 and to identify ways to improve the performance of OOD detection in neural networks.

V. Conclusion

In our study, we present a novel model designed to effectively address the out-of-distribution (OOD) detection problem. This model incorporates Bayesian principles as an optimizer to facilitate superior feature extraction, leverages Style Transfer Mix for image enhancement, and employs rectified activations to mitigate overconfidence issues. Our findings indicate that the Bayesian optimizer surpasses traditional optimization methods, such as Adam, in performance. Furthermore, the integration of data augmentation techniques, such as Style Transfer Mix, and the utilization of rectified activations contribute to improved uncertainty estimation and reduced overconfidence in OOD data.

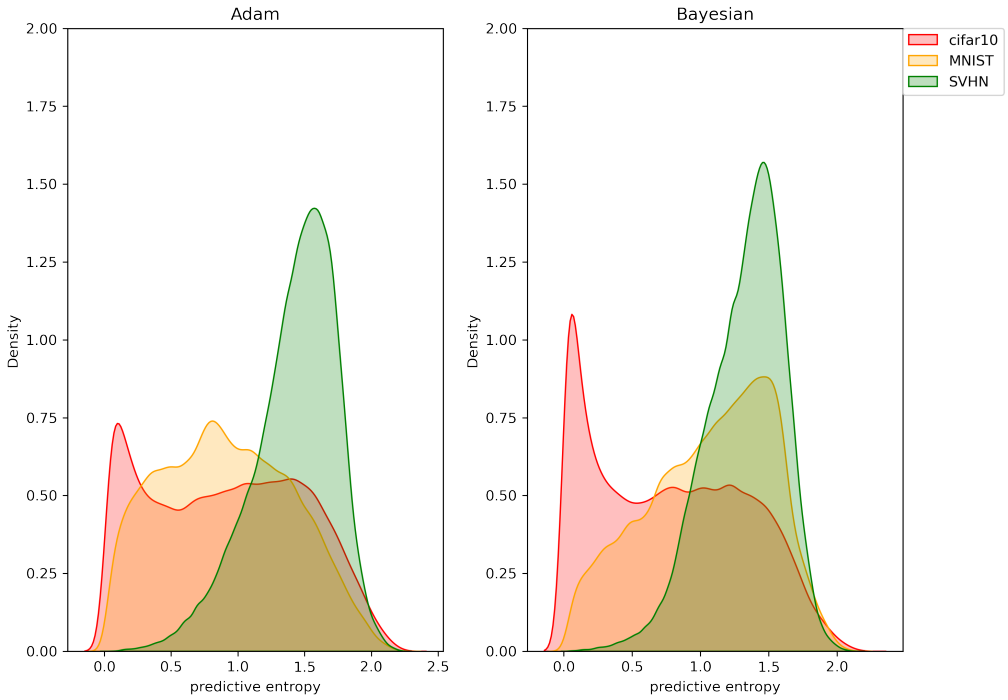

Fig. 6. Histograms of predictive entropy trained on LeNet5 and two separate optimizers, Adam and Bayesian. We clearly see that entropies are lower on in-distribution data (CIFAR-10) and higher on out-of-distribution data (MNIST and SVHN) optimized by the Bayesian method.

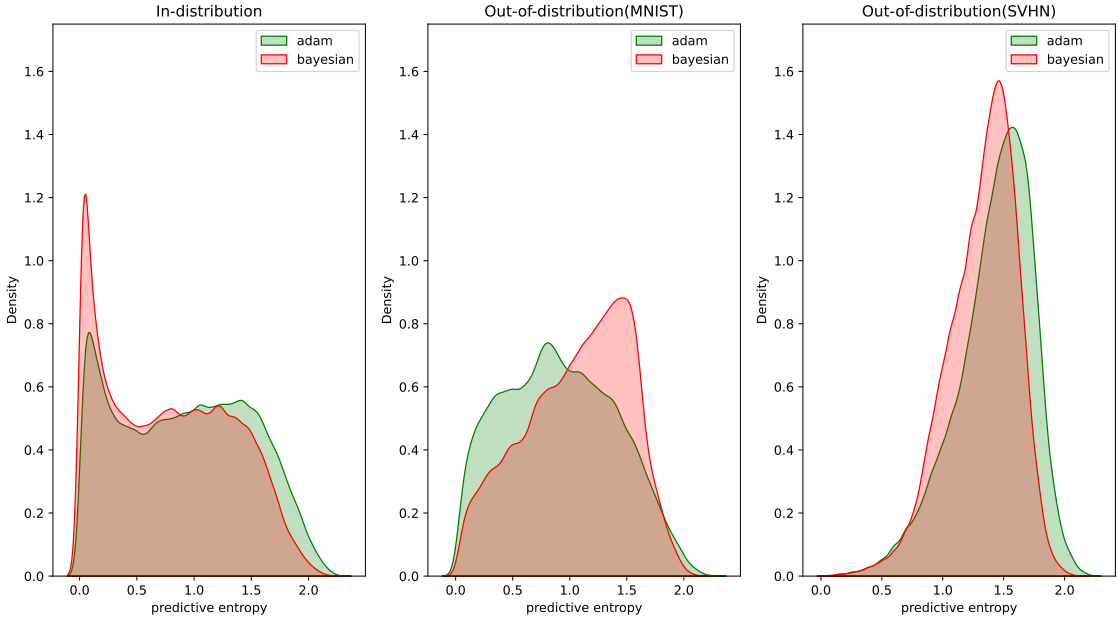

Fig. 7. A follow-up of Figure 6. From left to right: in-distribution data (CIFAR10), out-of-distribution data (MNIST), out-of-distribution data(SVHN). We can observe that the Bayesian optimizer has lower entropies in in-distribution data and higher in out-of-distribution data

TABLE I
PERFORMANCE METRICS FOR OOD DETECTION METHOD USING DATA AUGMENTATION

Model name	Optimizer	OOD detection method	OOD dataset	FPR95	AUROC	AUIN
ResNet18	Adam	knn	SVHN	72.14	87.73	82.64
			MNIST	66.31	89.63	80.10
			Texture	61.51	87.82	93.12
			AVG	66.65	88.39	85.29
Without data augmentation FNR: 5.01, Acc: 94.23, End-to-end Acc: 91.32						
ResNet18	Adam	knn	SVHN	68.73	88.21	83.30
			MNIST	61.52	90.72	81.75
			Texture	59.01	88.83	93.77
			AVG	63.09	89.25	86.27
With `style_transfer_mix` FNR: 5.01, Acc: 94.23, End-to-end Acc: 91.40						

TABLE II
PERFORMANCE METRICS FOR OOD DETECTION METHOD USING BAYESIAN AND ADAM OPTIMIER

Model name	Optimizer	OOD detection method	OOD dataset	FPR95	AUROC	AUIN
Lenet5	Adam	knn	SVHN	98.37	63.91	31.05
			MNIST	98.72	71.08	78.28
			Texture	79.65	73.39	82.60
			AVG	92.25	69.46	63.98
FNR: 5.01, Acc: 47.63, End-to-end Acc: 46.26						
Lenet5	Bayesian	knn	SVHN	98.23	49.62	13.51
			MNIST	99.96	58.17	67.72
			Texture	90.85	60.40	72.58
			AVG	96.35	56.07	51.27
FNR: 5.01, Acc: 67.23, End-to-end Acc: 65.06						

TABLE III
PERFORMANCE METRICS FOR OOD DETECTION METHOD USING RECTIFIED ACTIVATION

Model name	Optimizer	OOD detection method	OOD dataset	FPR95	AUROC	AUIN
Lenet5	Adam	knn	SVHN	98.41	63.87	32.23
			MNIST	99.92	59.55	71.75
			Texture	77.61	75.50	84.35
			AVG	91.98	66.31	62.78
Without rectified layer FNR: 5.01, Acc: 48,81, End-to-end Acc: 47.56						
Lenet5	Adam	knn	SVHN	98.02	64.51	31.62
			MNIST	99.21	70.69	78.40
			Texture	79.38	73.39	82.60
			AVG	92.20	69.53	64.21
With rectified layer FNR: 5.01, Acc: 47.63, End-to-end Acc: 46.32						

Fig. 8. ID Instance 1

Fig. 9. ID Instance 2

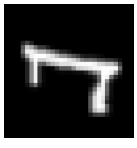

Fig. 10. OOD Instance 1

Fig. 11. OOD Instance 2

REFERENCES

[1] J. Chen, "Why a major AI Revolution is coming, but it's not what you think — AAAI 2020," Medium, Aug. 18, 2020. https://towardsdatascience.com/why-a-major-ai-revolution-is-coming-but-its-not-what-you-think-aaai-2020-aedbe2a3928f (accessed Apr. 22, 2023).

[2] A. Ballas and C. Diou, "Multi-layer representation learning for robust OOD Image Classification," Proceedings of the 12th Hellenic Conference on Artificial Intelligence, 2022.

[3] B. Charpentier, D. Zügner, and S. Günnemann, "Posterior network: Uncertainty estimation without OOD samples via density-based Pseudo-Counts," arXiv.org, 22-Oct-2020. [Online]. Available: https://arxiv.org/abs/2006.09239. [Accessed: 19-Apr-2023].

[4] C. Gonzalez et al., "Distance-based detection of out-of-distribution silent failures for Covid-19 lung lesion segmentation." arXiv, Aug. 05, 2022. doi: 10.48550/arXiv.2208.03217.

[5] S. M. Kahya, M. S. Yavuz, and E. Steinbach, "Reconstruction-based Out-of-Distribution Detection for Short-Range FMCW Radar." arXiv, Feb. 27, 2023. doi: 10.48550/arXiv.2302.14192.

[6] J. Bradshaw, A. G. de G. Matthews, and Z. Ghahramani, "Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks." arXiv, Jul. 08, 2017. doi: 10.48550/arXiv.1707.02476.

[7] R. M. Neal, Bayesian Learning for Neural Networks, vol. 118. in Lecture Notes in Statistics, vol. 118. New York, NY: Springer New York, 1996. doi: 10.1007/978-1-4612-0745-0.

[8] D. Mackay, "Bayesian Methods for Adaptive Models," PhD thesis, California Institute of Technology, 1991.

[9] G. Rey E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights | Proceedings of the sixth annual conference on Computational learning theory," Accessed: Apr. 23, 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/168304.168306

[10] L. K. Saul, T. Jaakkola, and M. I. Jordan, "Mean Field Theory for Sigmoid Belief Networks." arXiv, Feb. 29, 1996. Accessed: Apr. 23, 2023. [Online]. Available: http://arxiv.org/abs/cs/9603102

[11] C. Peterson, "A Mean Field Theory Learning Algorithm for Neural Networks".

[12] C. M. Bishop, Pattern recognition and machine learning. in Information science and statistics. New York: Springer, 2006.

[13] D. MacKay, "Information Theory, Inference, and Learning Algorithms".

[14] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge." arXiv, Jan. 29, 2015. doi: 10.48550/arXiv.1409.0575.

[15] K. Osawa et al., "Practical Deep Learning with Bayesian Principles." arXiv, Oct. 29, 2019. Accessed: Apr. 20, 2023. [Online]. Available: http://arxiv.org/abs/1906.02506

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 1, p. 60, Jul. 2019, doi: 10.1186/s40537-019-0197-0.

[17] P. T. Jackson, A. Atapour-Abarghouei, S. Bonner, T. Breckon, and B. Obara, "Style Augmentation: Data Augmentation via Style Randomization." arXiv, Apr. 12, 2019. Accessed: Apr. 19, 2023. [Online]. Available: http://arxiv.org/abs/1809.05375

[18] "CIFAR-10 and CIFAR-100 datasets." https://www.cs.toronto.edu/ kriz/-cifar.html (accessed Apr. 23, 2023).

[19] D. Hendrycks et al., "PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures." arXiv, Mar. 29, 2022. Accessed: Apr. 19, 2023. [Online]. Available: http://arxiv.org/abs/2112.05135

[20] M. E. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava, "Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam." arXiv, Aug. 02, 2018. doi: 10.48550/arXiv.1806.04854.

[21] A. Graves, "Practical Variational Inference for Neural Networks," in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2011.

[22] G. Zhang, S. Sun, D. Duvenaud, and R. Grosse, "Noisy Natural Gradient as Variational Inference." arXiv, Feb. 26, 2018. doi: 10.48550/arXiv.1712.02390.

[23] M. E. Khan and H. Rue, "The Bayesian Learning Rule." arXiv, Mar. 18, 2022. Accessed: Apr. 23, 2023. [Online]. Available: http://arxiv.org/abs/2107.04562

[24] M. E. Khan and W. Lin, "Conjugate-Computation Variational Inference: Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models." arXiv, Apr. 13, 2017. Accessed: Apr. 23, 2023. [Online]. Available: http://arxiv.org/abs/1703.04265

[25] Y. Sun, C. Guo, and Y. Li, "ReAct: Out-of-distribution Detection With Rectified Activations." arXiv, Nov. 24, 2021. doi: 10.48550/arXiv.2111.12797.

[26] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-Distribution Detection with Deep Nearest Neighbors." arXiv, Dec. 07, 2022. Accessed: Apr. 20, 2023. [Online]. Available: http://arxiv.org/abs/2204.06507

[27] "The Street View House Numbers (SVHN) Dataset." http://ufldl.stanford.edu/housenumbers/ (accessed Apr. 23, 2023).

[28] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." http://yann.lecun.com/exdb/mnist/ (accessed Apr. 23, 2023).

[29] "Describable Textures Dataset." https://www.robots.ox.ac.uk/ vgg/-data/dtd/ (accessed Apr. 23, 2023).

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." arXiv, Dec. 10, 2015. doi: 10.48550/arXiv.1512.03385.

[32] K. Osawa et al., "Practical Deep Learning with Bayesian Principles." arXiv, Oct. 29, 2019. doi: 10.48550/arXiv.1906.02506.

[33] D. Barber and C. M. Bishop, "Ensemble Learning in Bayesian Neural Networks".

[34] D. Hendrycks and K. Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." arXiv, Oct. 03, 2018. doi: 10.48550/arXiv.1610.02136.