NUQ: Nonparametric Uncertainty Quantification for Deterministic Neural Networks

Nikita Kotelevskii [*1] **Aleksandr Artemenkov** [*1] **Kirill Fedyanin** [1] **Fedor Noskov** [1 2] **Alexander Fishkov** [1]
Aleksandr Petiushko [3 4] **Maxim Panov** [1]

Abstract

This paper proposes a fast and scalable method for uncertainty quantification of machine learning models' predictions. First, we show the principled way to measure the uncertainty of predictions for a classifier based on Nadaraya-Watson's nonparametric estimate of the conditional label distribution. Importantly, the approach allows to disentangle explicitly *aleatoric* and *epistemic* uncertainties. The resulting method works directly in the feature space. However, one can apply it to any neural network by considering an embedding of the data induced by the network. We demonstrate the strong performance of the method in uncertainty estimation tasks on a variety of real-world image datasets, such as MNIST, SVHN, CIFAR-100 and several versions of ImageNet.

1. Introduction

It is crucial in many modern machine learning applications to complement the prediction with a "confidence" score. In particular, deep neural network models, which usually achieve state-of-the-art results in various tasks, are notorious for providing overconfident predictions on data they did not see during training (Nguyen et al., 2015). This issue restricts their wide usage in the fields with high costs of wrong predictions, such as medicine (Miotto et al., 2016), autonomous driving (Levinson et al., 2011; Filos et al., 2020), finance (Brando et al., 2018) and others. Thus, developing a reliable method of quantifying uncertainty is of great interest to researchers, especially practitioners.

The community in recent years made tremendous efforts to develop various uncertainty estimation methods and approaches, including calibration (Guo et al., 2017), ensembling (Lakshminarayanan et al., 2017), Bayesian methods (Gal & Ghahramani, 2016), and many others (Ovadia et al., 2019; Wang et al., 2019). Recently, a series of methods of uncertainty estimation based on the single deterministic neural network model was developed (Van Amersfoort et al., 2020; Liu et al., 2020; van Amersfoort et al., 2021). Their primary focus is on ensuring that embeddings of the data obtained on some layer of a network capture the geometrical relationships between the data samples in the input space, which is done via different regularization strategies. Given this property, one can apply a certain approach to capture uncertainty in the embedding space. The crucial property of these methods is the relatively mild required change in architectures and training procedures, which allows the direct application to the majority of existing deep learning models.

In practice it is usually important to distinguish two types of uncertainty: *aleatoric* and *epistemic* (Der Kiureghian & Ditlevsen, 2009; Kendall & Gal, 2017). The aleatoric uncertainty reflects the internal noise in the data due to class overlap, data markup errors, or other reasons. One can not reduce the uncertainty of this type by providing more data. The epistemic uncertainty reflects the model's imperfectness due to a lack of training data. We can reduce the uncertainty of this type once we get more data. Epistemic uncertainty, thus, may be used to identify *out-of-distribution (OOD) data*. If the model can quantify this type of uncertainty, it may abstain from prediction and address it to a human expert. Also, the ability to quantify epistemic uncertainty helps in active learning (Gal et al., 2017), where lack of "knowledge" naturally indicates in which areas we should label samples. Importantly, there is no universally accepted definition of uncertainty, and diverse, often heuristic treatments are usually used in practice.

Summary of the contributions. In this paper, we develop *a new and theoretically grounded* method of uncertainty quantification applicable to any deterministic neural network model. More specifically, our contributions are as follows.

1. We suggest looking at the pointwise Bayes risk (proba-

[*]Equal contribution [1]Skolkovo Institute of Science and Technology, Moscow, Russia [2]HSE University, Moscow, Russia [3]Lomonosov Moscow State University, Moscow, Russia [4]Work done at Huawei Moscow Research Center, Moscow, Russia. Currently at Artificial Intelligence Research Institute, Moscow, Russia. Correspondence to: Nikita Kotelevskii <nikita.kotelevskii@skoltech.ru>.

bility of the wrong prediction) as to the natural measure of the model prediction uncertainty at a particular data point.

2. We consider the Nadaray-Watson estimator of the conditional label distribution. Its asymptotic Gaussian approximation allows deriving uncertainty estimate based on the upper bound for the risk. We show the proposed estimate's consistency in the classification problem with the reject option.

3. We apply the resulting uncertainty estimation method in the neural network's embedding space. Our approach complements recent works in uncertainty estimation for deterministic neural networks and introduces a new nonparametric way, amenable for uncertainty disentanglement.

4. We implement the method in a scalable manner, which allows it to be used on large datasets such as ImageNet. The experimental results in OOD detection tasks show the significant potential of the proposed approach.

2. Nonparametric Uncertainty Quantification

2.1. Classification under Covariate Shift

In this section and below, for the sake of clarity, we provide derivations for the binary case. We derive a generalization to a multi-class setting in Section A.1.

Let's consider the standard binary classification setup $(X, Y) \in \mathbb{R}^d \times \{0, 1\}$ with $(X, Y) \sim \mathbb{P}$. We assume that we observe the dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of *i.i.d.* points from $\mathbb{P} = \mathbb{P}_{\text{train}}$. Here, X, Y denote random variables, while $\mathbf{x}, y$ are their realisations.

The classical problem in statistics and machine learning is to find a rule $\hat{g}$ based on the dataset $\mathcal{D}$ which approximates the optimal one:

$$g^* = \arg\min_g \mathbb{P}(g(X) \neq Y).$$

Here $g \colon \mathbb{R}^d \to \{0, 1\}$ is any classifier and the probability of wrong classification $\mathbb{P}(g(X) \neq Y)$ is usually called *risk*. The rule g^* is given by the *Bayes optimal classifier*:

$$g^*(\mathbf{x}) = \begin{cases} 1, & \eta(\mathbf{x}) \geq \frac{1}{2}, \\ 0, & \eta(\mathbf{x}) < \frac{1}{2}, \end{cases}$$

where $\eta(\mathbf{x}) = p(Y = 1 \mid X = \mathbf{x})$ which is the conditional probability of Y given $X = \mathbf{x}$ under the distribution $\mathbb{P}$.

In this work, we consider a situation when the distribution of the test samples $\mathbb{P}_{\text{test}}$ is different from the one for the training dataset $\mathbb{P}_{\text{train}}$, i.e. $\mathbb{P}_{\text{test}} \neq \mathbb{P}_{\text{train}}$. Obviously, the rule g^* obtained for $\mathbb{P} = \mathbb{P}_{\text{train}}$ might no longer be optimal if the aim is to minimize the error on the test data $\mathbb{P}_{\text{test}}(g(X) \neq Y)$.

In order to formulate a meaningful estimation problem, some additional assumptions are needed. We assume that the conditional label distribution $p(Y \mid X)$ **is the same** under both $\mathbb{P}_{\text{train}}$ and $\mathbb{P}_{\text{test}}$. This assumption has two important consequences:

1. All the difference between $\mathbb{P}_{\text{train}}$ and $\mathbb{P}_{\text{test}}$ is due to the difference between marginal distributions of X: $p_{\text{train}}(X)$ and $p_{\text{test}}(X)$. The situation when $p_{\text{test}}(X) \neq p_{\text{train}}(X)$ is known as *covariate shift*.

2. The rule g^* is still valid, i.e., optimal under $\mathbb{P}_{\text{test}}$.

However, while the classifier g^* is still optimal under covariate shift, its approximation $\hat{g}$ might be arbitrarily bad. The reason for that is that we can't expect $\hat{g}$ to approximate g^* well in the areas where we have few samples from the training set or don't have them at all. Thus, some special treatment of covariate shift is required.

2.2. Pointwise Risk and Its Estimation

We consider a classification rule $\hat{g}(X) = \hat{g}_{\mathcal{D}}(X)$ constructed based on the dataset $\mathcal{D}$. Let us start from defining pointwise risk of prediction:

$$\mathcal{R}(\mathbf{x}) = \mathbb{P}(\hat{g}(X) \neq Y \mid X = \mathbf{x}),$$

where $\mathbb{P}(\hat{g}(X) \neq Y \mid X = \mathbf{x}) \equiv \mathbb{P}_{\text{train}}(\hat{g}(X) \neq Y \mid X = \mathbf{x}) \equiv \mathbb{P}_{\text{test}}(\hat{g}(X) \neq Y \mid X = \mathbf{x})$ under the assumptions above. The value $\mathcal{R}(\mathbf{x})$ is independent of covariate distribution $p_{\text{test}}(X)$ and essentially allows to define a meaningful target of estimation which is based solely on the quantities known for the training distribution.

Let us note that the total risk value $\mathcal{R}(\mathbf{x})$ admits the following decomposition:

$$\mathcal{R}(\mathbf{x}) = \tilde{\mathcal{R}}(\mathbf{x}) + \mathcal{R}^*(\mathbf{x}),$$

where $\mathcal{R}^*(\mathbf{x}) = \mathbb{P}(g^*(X) \neq Y \mid X = \mathbf{x})$ is Bayes risk and $\tilde{\mathcal{R}}(\mathbf{x}) = \mathbb{P}(\hat{g}(X) \neq Y \mid X = \mathbf{x}) - \mathbb{P}(g^*(X) \neq Y \mid X = \mathbf{x})$ is an excess risk. Here $\mathcal{R}^*(\mathbf{x})$ corresponds to aleatoric uncertainty as it completely depends on the data distribution. Excess risk $\tilde{\mathcal{R}}(\mathbf{x})$ directly measures imperfectness of the model $\hat{g}$ and thus can be seen as a measure of epistemic uncertainty.

To proceed, we first assume that the classifier $\hat{g}$ has the standard form:

$$\hat{g}(\mathbf{x}) = \begin{cases} 1, & \hat{\eta}(\mathbf{x}) \geq \frac{1}{2}, \\ 0, & \hat{\eta}(\mathbf{x}) < \frac{1}{2}, \end{cases}$$

where $\hat{\eta}(\mathbf{x}) = \hat{p}(Y = 1 \mid X = \mathbf{x})$ is an estimate of the conditional density $\eta(\mathbf{x})$.

For such an estimate we can upper bound the excess risk via the following classical inequality (Devroye et al., 2013):

$$\tilde{\mathcal{R}}(\mathbf{x}) \leq 2|\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})|.$$

It allows us to obtain an upper bound for the total risk:

$$\mathcal{R}(\mathbf{x}) \leq \mathcal{L}(\mathbf{x}) = \mathcal{R}^*(\mathbf{x}) + 2|\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})|,$$

where $\mathcal{R}^*(\mathbf{x}) = \min\{\eta(\mathbf{x}), 1 - \eta(\mathbf{x})\}$ in the case of binary classification. While this upper bound still depends on the unknown quantity $\eta(\mathbf{x})$, we will see in the next section that $\mathcal{L}(\mathbf{x})$ allows for an efficient approximation under mild assumptions.

2.3. Nonparametric Uncertainty Quantification

2.3.1. Kernel Density Estimate and Its Asymptotic Distribution

To obtain an estimate of $\mathcal{L}(\mathbf{x})$ and, consequently, bound the risk, we need to consider some particular type of estimator $\hat{\eta}$. In this work, we choose the classical kernel-based Nadaraya-Watson estimator of the conditional label distribution as it allows for a simple description of its asymptotic properties.

Let us denote by $K_h \colon \mathbb{R}^d \mapsto \mathbb{R}$ the multi-dimensional kernel function with bandwidth h. Typically, we consider a multi-dimensional Gaussian kernel, but other choices are also available. We refer readers to Section A.3 for details.

Then, the conditional probability estimate is expressed as:

$$\hat{\eta}(\mathbf{x}) = \frac{\sum_{i=1}^N \mathbb{1}[y_i = 1] \cdot K_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^N K_h(\mathbf{x} - \mathbf{x}_i)}, \qquad (1)$$

where y_i is either 0 or 1.

The difference between $\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})$ for properly chosen bandwidth h converges in distribution as follows (see, e.g. (Powell, 2010)):

$$\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x}) \to \mathcal{N}\left(0, \frac{1}{N}\frac{\sigma^2(\mathbf{x})}{p(\mathbf{x})}\int [K_h(\mathbf{u})]^2 d\mathbf{u}\right), \quad (2)$$

where N is the number of data points in the training set, and $\sigma^2(\mathbf{x})$ is the standard deviation of the data label at point $\mathbf{x}$.

Now we are equipped with an estimate of the distribution for $\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})$. Let us denote by $\tau(\mathbf{x})$ the standard deviation of a Gaussian from equation (2):

$$\tau^2(\mathbf{x}) = \frac{1}{N}\frac{\sigma^2(\mathbf{x})}{p(\mathbf{x})}\int [K_h(\mathbf{u})]^2 d\mathbf{u}.$$

In the following sections, we first show how to use the obtained property for uncertainty estimation, and then we will show how it can be computed.

2.3.2. Total, Aleatoric and Epistemic Uncertainty and Their Estimates

In this work, we suggest a particular uncertainty quantification procedure inspired by the derivation above which we call *Nonparametric Uncertainty Quantification (NUQ)*. More specifically, we suggest to consider the following measure of the total uncertainty:

$$\mathbf{U}_t(\mathbf{x}) = \min\{\eta(\mathbf{x}), 1 - \eta(\mathbf{x})\} + 2\sqrt{\frac{2}{\pi}}\tau(\mathbf{x}),$$

which is obtained by considering an asymptotic approximation of

$$\mathbb{E}_{\mathcal{D}}\mathcal{L}(\mathbf{x}) = \min\{\eta(\mathbf{x}), 1 - \eta(\mathbf{x})\} + 2\mathbb{E}_{\mathcal{D}}\left|\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})\right|$$

in a view of (2) and the fact, that $\mathbb{E}|\xi| = \text{std}(\xi)\sqrt{\frac{2}{\pi}}$ for the zero-mean normal variable ξ. The resulting estimate upper bounds the average error of estimation at point $\mathbf{x}$ and thus indeed can be used as the measure of total uncertainty.

We also can write the corresponding measures of aleatoric and epistemic uncertainties:

$$\mathbf{U}_a(\mathbf{x}) = \min\{\eta(\mathbf{x}), 1 - \eta(\mathbf{x})\}, \mathbf{U}_e(\mathbf{x}) = 2\sqrt{\frac{2}{\pi}}\tau(\mathbf{x}). \quad (3)$$

Finally, the data-driven uncertainty estimates $\hat{\mathbf{U}}_a(\mathbf{x}), \hat{\mathbf{U}}_e(\mathbf{x})$ and $\hat{\mathbf{U}}_t(\mathbf{x})$ can be obtained via plug-in using estimates $\hat{\eta}(\mathbf{x})$, $\hat{\sigma}(\mathbf{x})$, $\hat{p}(\mathbf{x})$ and, consequently, $\hat{\tau}^2(\mathbf{x}) = \frac{1}{N}\frac{\hat{\sigma}^2(\mathbf{x})}{\hat{p}(\mathbf{x})}\int \left[K_h(\mathbf{u})\right]^2 d\mathbf{u}$. In the next section we discuss the efficient computation of these estimates and the choice of the hyperparameters.

2.3.3. How to Compute Uncertainty Estimates?

The computation of nonparametric estimate (1) involves a sum over the whole available data. This could be intractable in practice when we are working with large datasets. However, the typical kernel K_h quickly approaches zero with the increase of the norm of the argument: $\|\mathbf{x} - \mathbf{x}_i\|$. Thus, we can use an approximation of kernel estimate: instead of the sum over all elements in the dataset, we consider the contribution of only several nearest neighbors. It requires a fast algorithm for finding the nearest neighbors. For this purpose, we use the approach of (Malkov & Yashunin, 2018) based on Hierarchical Navigable Small World graphs (HNSW). It provides a fast, scalable, and easy-to-use solution to the computation of nearest neighbors.

Algorithm 1 NUQ inference algorithm.

Input: Training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, inference point $\mathbf{z}$
Output: Prediction $\hat{g}(\mathbf{z})$ and uncertainty estimate $\hat{\mathbf{U}}_t(\mathbf{z})$

$\{\mathbf{x}_{i_k}\}_{k=1}^K \leftarrow K$ nearest neighbors of $\mathbf{z}$ among $\{\mathbf{x}_i\}_{i=1}^N$
$\hat{\eta}_c(\mathbf{z}) \leftarrow \frac{\sum_{k=1}^K K_h(\mathbf{x}_{i_k}-\mathbf{z})\mathbb{1}[y_{i_k}=c]}{\sum_{k=1}^K K_h(\mathbf{x}_{i_k}-\mathbf{z})}$
$\hat{\sigma}_c^2(\mathbf{z}) = \hat{\eta}_c(\mathbf{z})\big(1-\hat{\eta}_c(\mathbf{z})\big)$
$\hat{g}(\mathbf{z}) \leftarrow \operatorname{argmax}_c \ \hat{\eta}_c(\mathbf{z})$
$\hat{p}(\mathbf{z}) \leftarrow$ either KDE: $\frac{1}{Nh^d}\sum_{k=1}^K K_h(\mathbf{x}_{i_k}-\mathbf{z})$ or GMM
$\hat{\tau}^2(\mathbf{z}) \leftarrow \frac{1}{N}\frac{\max_c \hat{\sigma}_c^2(\mathbf{z})}{\hat{p}(\mathbf{z})}\int\big[K_h(\mathbf{u})\big]^2 d\mathbf{u}$
$\hat{\mathbf{U}}_t(\mathbf{z}) \leftarrow \min_c\big\{1-\hat{\eta}_c(\mathbf{z})\big\} + 2\sqrt{\frac{2}{\pi}}\hat{\tau}(\mathbf{z})$

The other design choice is related to the estimate of marginal density $\hat{p}(\mathbf{x})$. While it is natural to use kernel estimates within the considered framework, one can also use other density estimates. In particular, we found Gaussian Mixture Models (GMMs) useful in some cases; see details in Section 4.

To summarize the proposed method and to ease understanding, we provide a pseudo-code, see Algorithm 1. Please note that the algorithm is generalized there to the multiclass case, see details in Section A.1.

The only remaining unspecified ingredient of the procedure is the choice of bandwidth h for KDE and number of neighbors K. In this work, we suggest simply optimizing the classification quality of the Nadaraya-Watson classifier, i.e., h and K are chosen to obtain the best classification quality measured by cross-validation procedure on the training set.

2.4. Consistency of NUQ-based classification with reject option

Above we obtained uncertainty estimates that characterize the classical risk of prediction. However, they are also helpful to solve the formal problem of classification with reject option. In this problem, for any input $\mathbf{x}$ we can choose either we perform prediction or reject it. Following (Chow, 1970), we assume that in case of prediction we pay binary price depending whether the prediction was correct or not, while in the case of rejection we pay the constant price $\lambda \in (0, 1)$. For this task the risk function is

$$\mathcal{R}_\lambda(\mathbf{x}) = \mathcal{R}(\mathbf{x})\mathbb{1}\{\alpha(\mathbf{x}) = 0\} + \lambda\mathbb{1}\{\alpha(\mathbf{x}) = 1\},$$

where $\alpha(\mathbf{x})$ is an indicator of the rejection.

The minimizer of $\mathcal{R}_\lambda(\mathbf{x})$ is given by the optimal Bayes classifier $g^*(\mathbf{x})$ and the abstention function

$$\alpha^*(\mathbf{x}) = \begin{cases} 0, & \mathcal{R}^*(\mathbf{x}) \le \lambda, \\ 1, & \mathcal{R}^*(\mathbf{x}) > \lambda. \end{cases}$$

To approximate $\alpha^*(\mathbf{x})$, we utilize hypothesis testing:

$$H_0\colon \mathcal{R}^*(\mathbf{x}) > \lambda \text{ vs. } H_1\colon \mathcal{R}^*(\mathbf{x}) \leqslant \lambda.$$

We choose the confidence level $\beta > 0$ and consider the statistic

$$\hat{\mathbf{U}}_\beta(\mathbf{x}) = \min\big\{\hat{\eta}(\mathbf{x}), 1-\hat{\eta}(\mathbf{x})\big\} + z_{1-\beta}\hat{\tau}(\mathbf{x}),$$

where $z_{1-\beta}$ is $1-\beta$ quantile of the standard normal distribution. The statistic $\hat{\mathbf{U}}_\beta(\mathbf{x})$ combines the plug-in estimate of the Bayes risk $\min\big\{\hat{\eta}(\mathbf{x}), 1-\hat{\eta}(\mathbf{x})\big\}$ and the term $z_{1-\beta}\hat{\tau}(\mathbf{x})$ accounting for the confidence of estimation. The resulting abstention rule is given by:

$$\hat{\alpha}_\beta(\mathbf{x}) = \begin{cases} 0, & \hat{\mathbf{U}}_\beta(\mathbf{x}) \le \lambda, \\ 1, & \hat{\mathbf{U}}_\beta(\mathbf{x}) > \lambda. \end{cases}$$

Finally, if we consider the pair of kernel classifier $\hat{g}(\mathbf{x})$ and $\hat{\alpha}_\beta(\mathbf{x})$ then we can prove the consistency result for the corresponding risk $\hat{\mathcal{R}}_\lambda(\mathbf{x})$ under standard assumptions on nonparametric densities, see Section A.2 for details.

Theorem 2.1. *Suppose that assumptions A.1-A.3 hold and $p(\mathbf{x}) > 0$, the bandwidth $h \to 0$ and $Nh^d \to \infty$ as N tends to infinity. Then, for any $\beta < 1/2$:*

$$\mathbb{E}_{\mathcal{D}}\hat{\mathcal{R}}_\lambda(\mathbf{x}) - \mathcal{R}^*_\lambda(\mathbf{x}) \underset{N\to\infty}{\longrightarrow} 0.$$

Moreover, if the support of X_i is compact,

$$\mathbb{E}\left\{\mathbb{E}_{\mathcal{D}}\hat{\mathcal{R}}_\lambda(X) - \mathcal{R}^*_\lambda(X)\right\} \underset{N\to\infty}{\longrightarrow} 0,$$

where X is a random variable distributed according to $p(\mathbf{x})$ and independent of all X_i, $i \in [N]$.

This result shows the validity of the NUQ-based abstention procedure. Interesting future work is to obtain a precise convergence rate for the method. It should be possible based on the finite sample bounds provided in Section A.2.

3. Related Work

The notion of uncertainty naturally appears in Bayesian statistics (Gelman et al., 2013), and, thus, Bayesian methods are often used for uncertainty quantification. Exact Bayesian inference is computationally intractable, and approximations are used. Two popular ideas are Markov Chain Monte Carlo sampling (MCMC; Neal et al. (2011)) and Variational Inference (VI; Blei et al. (2017)). MCMC has theoretical guarantees to be asymptotically unbiased but high computational cost. VI-based approaches (Rezende & Mohamed, 2015; Dinh et al., 2017; Papamakarios et al., 2021; Kobyzev et al., 2020) are more scalable, they are biased and at least

double the number of parameters. That's why some alternatives are considered, such as the Bayesian treatment of Monte-Carlo dropout (Gal & Ghahramani, 2016).

Deep Ensemble (Lakshminarayanan et al., 2017) is usually considered as a quite strong yet expensive approach. A series of papers developed ways of approximating the distribution obtained by an ensemble of models by a single probabilistic model (Malinin & Gales, 2018; Malinin et al., 2020; Sensoy et al., 2018). These methods require changing the training procedure and having more parameters to train.

Recently, another popular type of model for uncertainty quantification was proposed. Specifically, it was proposed to consider a single deterministic neural network model and only apply mild changes to the architecture and training procedure. The crucial idea behind these methods is to ensure that the embedding space induced by the network captures the geometry of the input space. Usually, it is reached via some regularization techniques such as weight clipping, gradient penalty, and spectral normalization (Van Amersfoort et al., 2020; van Amersfoort et al., 2021; Mukhoti et al., 2021; Liu et al., 2020). Deep neural networks trained in this way achieve results comparable with standard approaches.

For uncertainty estimation with such networks variety of methods were proposed. In DUQ (Van Amersfoort et al., 2020), an RBF layer is added to the network with a custom procedure to adjust the centroid points (in embedding space). The downside of the method is its inability to distinguish aleatoric and epistemic uncertainty. Another approach to capture epistemic uncertainty was proposed in the DDU approach (Mukhoti et al., 2021) which uses Gaussian mixture model to estimate the density of objects in embedding space of a trained neural network. The density values are then used as a confidence measure. SNGP (Liu et al., 2020) and DUE (van Amersfoort et al., 2021) are similar but use a Gaussian process as the final layer, requiring estimating covariance with the use of inducing points or RFF expansion.

One may wonder about the difference between NUQ and DDU, especially when the Gaussian Mixture Model is used for covariate density approximation. The difference is twofold:

- NUQ is based on the rigorous derivation of total, aleatoric and epistemic uncertainties, while DDU suggests a heuristic way to capture epistemic one;
- Even when GMM is exploited, an additional term $\sigma^2(x)$ in NUQ affects the result. It is there for purpose as apparently noise in the data influences model quality and, correspondingly, epistemic uncertainty.

Another branch of work suggests using out-of-distribution data explicitly while training the model (Malinin & Gales, 2018; Jain et al., 2021). This could be beneficial, but only in some specific cases when we know what the out-of-distribution data looks like in advance.

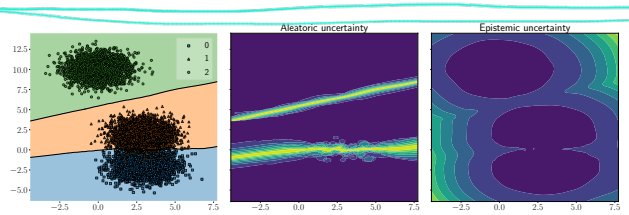


Figure 1: Left plot shows the data and the result of the classification by the Bayes classifier based on the nonparametric estimate of conditional density. The next two plots show different types of uncertainties: aleatoric and epistemic. The lighter color, the higher uncertainty. We see that the former does not increase as we go away from training data, while the latter does.

There is a wide range of papers discussing classification with the reject option. Most likely, the problem was firstly studied by Chow in papers (Chow, 1957; 1970). Moreover, in the article (Chow, 1970) he introduced a risk function used across this paper. Herbei & Wegkamp (2006) studied an optimal procedure for this risk and provided a plug-in rule. Then people investigated either empirical risk minimization among a class of hypotheses (see (Bartlett & Wegkamp, 2008; Cortes et al., 2016)) or other types of risk (see (Denis & Hebiri, 2015; El-Yaniv & Wiener, 2010; LEI, 2014)). Besides, a number of practical works were presented, see, for example, (Grandvalet et al., 2009; Geifman & El-Yaniv, 2019; Nadeem et al., 2009).

4. Experiments

4.1. Toy Example

We start this section with the application of the proposed *Nonparametric Uncertainty Quantification (NUQ)* method to a toy example. As a dataset, we use a 2-dimensional mixture of three Gaussians with centers at points [3, -2], [3, 2], [0, 10], and variance equal to 1. Each Gaussian is treated as a separate class (see Figure 1, the leftmost panel).

We consider the Bayes classifier based on the nonparametric estimate of the conditional density (1) and which takes samples from these Gaussians as an input. We compute aleatoric and epistemic uncertainty values according to equations (3). Bandwidth was selected according to our proposed approach, explained in Section 2.3.3. Classification results and uncertainties for this toy problem are presented in Figure 1. The central and the rightmost plots present aleatoric and epistemic uncertainty estimates obtained. The uncertainty measures show the desired behavior: aleatoric

uncertainty is large in-between the classes, while epistemic uncertainty increases with the distance to the training data.

We also conducted another toy experiment, where we know how exact uncertainties should look like, and we refer readers to Section A.6 to see the details.

4.2. Image Classification Datasets

In this section, we consider a series of experiments on image datasets. In contrast to the toy example above, where raw data was passed into Bayesian classifier directly, here we should do the following:

1. Train a parametric model, typically a neural network. In what follows, we call this model a *base model*.
2. Fit NUQ afterward, using the embeddings that the base model extracts from the training set.

We emphasize that NUQ is the postprocessing method, and it is fitted on top of the embeddings of the base model. In the experiments of this section, we use logits as extracted features, if not explicitly stated otherwise. However, other options are also possible; see Section A.7.1.

Following SNGP and DDU, we use spectral normalization to train the base model to achieve bi-Lipschitz property and avoid feature collapse problem. However, NUQ works sufficiently good even without this regularization (see Table 6 in Section A.7.1).

We compare popular measures of uncertainty which do not require significant modifications to model architectures and training procedures. More specifically, we consider:

1. Maximum probability (MaxProb):$1 - \max_c p(y = c \mid \mathbf{x})$;
2. Entropy:$-\sum_{c=1}^{C} p(y = c \mid \mathbf{x}) \log p(y = c \mid \mathbf{x})$;
3. Monte-Carlo dropout (Gal & Ghahramani, 2016);
4. Ensemble of models trained with different random seeds;
5. Test-Time Augmentation (TTA) – augmentation, applied to data at inference time;
6. DDU (Mukhoti et al., 2021) involves Gaussian Mixture Model (GMM)-like an approximation of extracted features to predict uncertainties.
7. SNGP (Liu et al., 2020) approach uses Random Fourier Features to approximate the Gaussian process on the last layer of a neural network while using spectral normalization to preserve the adequate distance in the embedding space;
8. DUQ (Van Amersfoort et al., 2020) approach suggests using RBF networks with regularized Jacobian w.r.t. input. Then, for each class c a weight matrix W_c and centroid center e_c are trained. Models prediction and the associated uncertainty are argmax and max over c correspondingly, applied to RBF kernel, where they use both W_c and e_c.

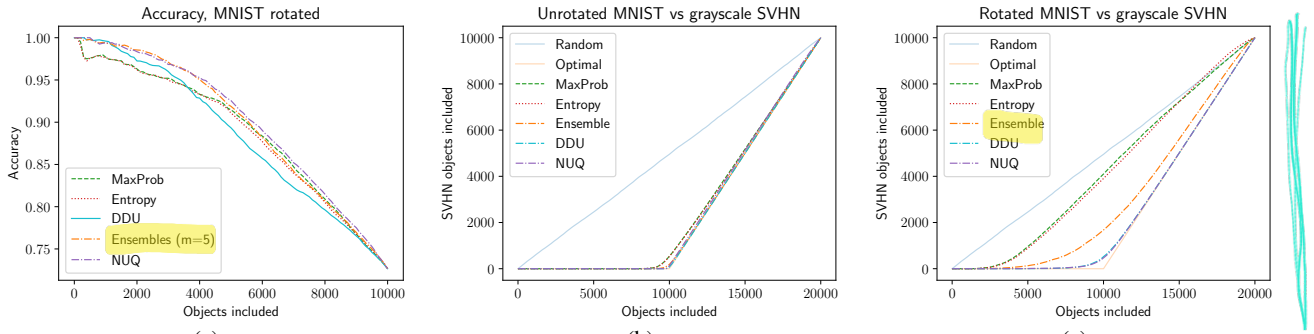
For Monte-Carlo dropout, Ensembles, and TTA, we first compute average predicted class probabilities and then compute their entropy. More details can be found in Section A.4.

Before we proceed, we want to address one important question. One may ask whether nonparametric classification method used in NUQ, trained on some embedding from the base model, has any relation to the original neural network. To reassure the reader, we provide an argument that it well approximates the predictions of the base model and NUQ-based uncertainty estimates can be used for the base model as well. Specifically, we compute the agreement between predictions obtained from the Bayes classifier based on kernel estimate (i.e. the one used in NUQ) and base models' predictions. This metric formally can be defined as $\text{agreement}(\hat{p}, p) = \frac{1}{n}\sum_{i=1}^{n} I\left[\arg\max_j \hat{p}(y = j \mid \mathbf{x}_i) = \arg\max_j p(y = j \mid \mathbf{x}_i)\right]$. For CIFAR-100 (see more experiments on this dataset in Section 4.2.3), this metric gives us the agreement of 0.975, which tells that the approach is accurate.

4.2.1. Rotated MNIST

The first example is classification under covariate shift on MNIST (LeCun et al., 2010). We train a small convolutional neural network with three convolution layers, see Section A.4. This is the base model we use to obtain logits for the input objects. We consider a particular instance of distribution shift for evaluation by using a test set of MNIST images rotated at a random angle in the range from 30 to 45 degrees. This set contains 10000 images. The range of angles reassures that the data does not look like the original MNIST data, though many resulting pictures can still remind the ones from training.

This experiment considers two simple baselines: MaxProb and Entropy-based uncertainty estimates of the base model. We compliment them with two more challenging baselines: Deep ensemble and DDU (Mukhoti et al., 2021) (as the most successful among deterministic methods). We compare them all with NUQ-based estimate of epistemic uncertainty $\hat{\mathbf{U}}_e(\mathbf{x})$. To evaluate the quality of the uncertainty estimates, we sort the objects from the test dataset in order of ascending uncertainties. Then we obtain the model's predictions and plot how accuracy changes with the number of objects taken into consideration; see Figure 2a. The valid uncertainty estimation method is expected to produce the



(a) (b) (c)

Figure 2: (a) Accuracy for images sorted by uncertainty on rotated MNIST. (b) Number of SVHN images included into consideration vs unrotated MNIST. In this simpler version, even the basic entropy manages to achieve a good result. (c) More challenging task – number of SVHN images included into consideration vs rotated MNIST. NUQ still distinguish between datasets with close to an optimal solution.

plot with accuracy decreasing when more objects are taken into account. Moreover, the higher is the plot, the better is the quality of the corresponding uncertainty estimate.

We see that the plots for all the considered methods show the expected trend, while uncertainties obtained by NUQ are more reliable. NUQ distinctly outperforms DDU and has comparable performance with deep ensembles.

4.2.2. MNIST vs. SVHN

To make the problem more challenging, we consider the SVHN dataset (Netzer et al., 2011), convert it to grayscale, and resize it to the shape of 28x28. The size of this additional SVHN-based dataset is again 10000. We take the base model trained on MNIST from the previous section and consider the problem of OOD detection with SVHN being the OOD dataset. As in-distribution data, we first consider the test set of 10000 MNIST images. We again compute uncertainties for each object on this concatenated dataset (10000 of MNIST and 10000 of SVHN) and sort them by their uncertainties in ascending order. The goal for uncertainty quantification methods is to sort all objects so that all MNIST images have lower uncertainty values than SVHN ones. Note that the optimal decision rule in this case is a ReLU-shaped function, with a break at point 10000.

For NUQ we use epistemic uncertainty $\hat{\mathbf{U}}_e(\mathbf{x})$ in this experiment. In Figure 2b we plot the number of objects included from the SVHN dataset. It is seen that Ensembles, NUQ, and DDU overperform MaxProb and Entropy. All of these three methods perform almost as an optimal decision.

Next, we consider more challenging problem of separation between rotated MNIST (see Section 4.2.1) and SVHN. We expect it is harder to distinguish between them as rotated MNIST images differ from those used to train the network. However, Figure 2c shows that NUQ still does a very good job and allows for almost perfect separation. Interestingly, this example shows that ensembles are worse than DDU and NUQ. The performance of the last two is visually almost identical.

4.2.3. CIFAR-100

To reinforce our results on simpler datasets, we further conduct experiments on more challenging CIFAR-100 (Krizhevsky, 2009). We want our model to detect the unconventional samples, and thus we treat the out-of-distribution detection as a binary classification task (OOD/not-OOD) by uncertainty score, and we report the ROC-AUC for that task. Following the setup from the recent works (Van Amersfoort et al., 2020; van Amersfoort et al., 2021; Sastry & Oore, 2020), we use SVHN, LSUN (Yu et al., 2015) and Smooth (Hein et al., 2019) datasets as OOD datasets.

We trained the ResNet-50 model from scratch on CIFAR-100. For this more challenging problem, we extend the list of our competitors. Specifically, in addition to the previous ones, we consider Monte-Carlo dropout (Gal & Ghahramani, 2016), SNGP (Liu et al., 2020) and DUQ (Van Amersfoort et al., 2020). Following DDU (Mukhoti et al., 2021) and SNGP (Liu et al., 2020), which are suggested to be trained with spectral normalization (Miyato et al., 2018) to ensure the bi-Lipschitz constraint for mappings at each layer, we train NUQ's base model in the same fashion. However, according to our additional experiments (see Table 6 in Section A.7.1), it is not crucial for NUQ, and, in principle, it could be applied to any neural network without any changes.

For DUQ (Van Amersfoort et al., 2020) to achieve the bi-Lipschitz property, authors suggested using gradient penalty during training. Unfortunately, the experiments in the paper were conducted only on simple CIFAR-10 and SVHN, and we found it complicated to achieve a good performance score using gradient penalty on CIFAR-100 and ImageNet.

OOD dataset	MaxProb*	Entropy*	Dropout	Ensemble	TTA	DUQ*	SNGP*	DDU*	NUQ*
SVHN	79.7±1.3	81.1±1.6	77.6±2.5	82.9±0.9	81.6±1.2	88.7±6.3	86.2±7.4	**89.6±1.6**	**89.7±1.6**
LSUN	81.5±2.0	83.0±2.1	76.8±5.1	86.5±0.8	85.0±2.7	90.8±6.7	83.7±8.6	**92.1±0.6**	**92.3±0.6**
Smooth	76.6±3.5	77.8±5.2	63.3±3.8	83.7±1.2	73.2±10.8	91.1±8.4	60.9±12.5	**97.1±3.1**	**96.8±3.8**

Table 1: OOD detection for CIFAR-100 in-distribution dataset with ResNet-50 neural network. The top two results are shown in bold. Evaluation is done for three models trained with different seeds to estimate the standard deviation. Methods requiring a single pass over the data to compute uncertainty estimates are marked with *.

OOD dataset	MaxProb*	Entropy*	TTA	Ensemble	DDU*	DUQ*	SNGP*	NUQ*
ImageNet-R	80.4	83.6	85.8	84.4	80.1	73.3	85.0	**99.5**
ImageNet-O	28.2	29.1	30.5	51.9	74.1	71.4	75.8	**82.4**

Table 2: ROC-AUC score for ImageNet out-of-distribution detection tasks for different methods. Methods requiring a single pass over the data to compute uncertainty estimates are marked with *.

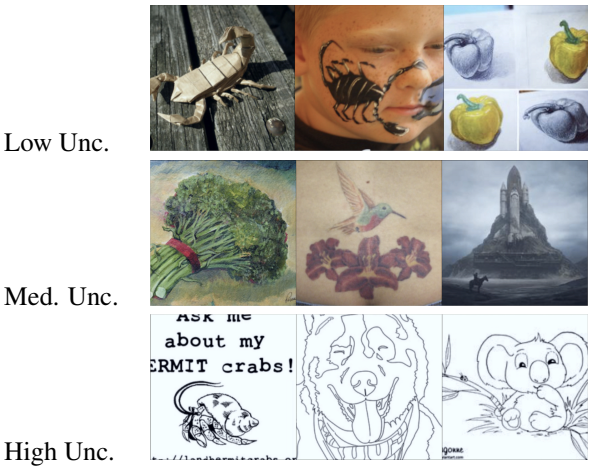


Figure 3: Typical OOD images for different levels of uncertainty as predicted by NUQ.

Thus, we decided to use spectral normalization for DUQ as well. See more details in Section A.7.3.

For Deep Ensembles, we fixed the number of models to 5. Still, additional experiments were conducted (see Section A.7.2) where we changed the number of models, both for CIFAR-100 and ImageNet experiments.

In this experiment, NUQ was applied to the features from the penultimate layer, and the density estimate is given by GMM, as it provides the best results (see the results for other choices of hyperparameters in Section A.7.1).

The results are presented in Table 1. We can clearly see that NUQ and DDU show close results while outperforming the competitors with a significant margin.

4.2.4. IMAGENET

To evaluate the method's applicability to the large-scale data, we have applied our approach to the ImageNet (Deng et al., 2009) dataset. As OOD data we used the ImageNet-O (Hendrycks et al., 2021b) and ImageNet-R (Hendrycks et al., 2021a) datasets. ImageNet-O consists of images from classes not found in the standard ImageNet dataset. ImageNet-R contains different artistic renditions of ImageNet classes.

In contrast to the previous experiment, we found that for NUQ, it is more beneficial to use KDE as a density estimator $p(x)$, rather than Gaussian Mixture Model.

The results are summarized in Table 2. We see that for ImageNet-O, many methods show good OOD detection quality, but NUQ achieves an almost perfect result. For ImageNet-R simple approaches completely fail while DDU, SNGP, and DUQ perform pretty well, and NUQ shows the best result with a large margin.

It is interesting that unlike in the CIFAR-100 experiment, for ImageNet NUQ significantly outperforms DDU. We conjecture that GMM struggles to approximate density here as an embedding structure is much more complicated for ImageNet compared to CIFAR-100 (see some visualizations in Section A.5). NUQ is beneficial in this case as KDE is much more flexible than GMM and provides a better result.

Additionally, we looked at some typical samples from ImageNet-R with low, moderate, and high levels of uncertainty as assigned by NUQ, see Figure 3. Here, low, medium, and high uncertainties correspond to 10, 50, and 90% quantiles of epistemic uncertainty distribution for images from ImageNet-R dataset. We observe that uncertainty values corresponds well to these images' intuitive degree of complexity compared to the original ImageNet data.

5. Conclusions

This work proposes NUQ, a new principled uncertainty estimation method that applies to a wide range of neural network models. It does not require retraining the model and acts as a postprocessing step working in the embedding space induced by the neural network. NUQ significantly outperforms the competing approaches with only re-

cently proposed DDU method (Mukhoti et al., 2021) showing comparable results. Importantly, in the most practical example of OOD detection for ImageNet data, NUQ shows the best results with a significant margin. All the code to reproduce the experiments is available at `https://github.com/stat-ml/NUQ`.

We hope that our work opens a new perspective on uncertainty quantification methods for deterministic neural networks. We also believe that NUQ is suitable for in-depth theoretical investigation, which we defer to future work.

Bibliography

Bartlett, P. L. and Wegkamp, M. H. Classification with a Reject Option using a Hinge Loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008. ISSN 1533-7928.

Blei, D. M., Kucukelbir, A., et al. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Brando, A., Rodríguez-Serrano, J. A., et al. Uncertainty Modelling in Deep Networks: Forecasting Short and Noisy Series. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 325–340. Springer, 2018.

Chow, C. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1): 41–46, January 1970. ISSN 1557-9654. doi: 10.1109/TIT.1970.1054406. Conference Name: IEEE Transactions on Information Theory.

Chow, C. K. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, December 1957. ISSN 0367-9950. doi: 10.1109/TEC.1957.5222035. Conference Name: IRE Transactions on Electronic Computers.

Cortes, C., DeSalvo, G., and Mohri, M. Learning with Rejection. In *ALT*, 2016. doi: 10.1007/978-3-319-46379-7_5.

Deng, J., Dong, W., et al. Imagenet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Denis, C. and Hebiri, M. Consistency of plug-in confidence sets for classification in semi-supervised learning. 2015. doi: 10.1080/10485252.2019.1689241.

Der Kiureghian, A. and Ditlevsen, O. Aleatory or Epistemic? Does It Matter? *Structural Safety*, 31(2):105–112, 2009.

Devroye, L., Györfi, L., and Lugosi, G. *A Probabilistic Theory of Pattern Recognition*, volume 31. Springer Science & Business Media, 2013.

Dinh, L., Sohl-Dickstein, J., et al. Density Estimation using Real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

El-Yaniv, R. and Wiener, Y. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010. ISSN 1533-7928.

Filos, A., Tigkas, P., et al. Can Autonomous Vehicles Identify, Recover from, and Adapt to Distribution Shifts? In *International Conference on Machine Learning*, pp. 3145–3153. PMLR, 2020.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Balcan, M. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1050–1059. JMLR.org, 2016.

Gal, Y., Islam, R., et al. Deep Bayesian Active Learning with Image Data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR, 2017.

Geifman, Y. and El-Yaniv, R. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2151–2159. PMLR, May 2019. ISSN: 2640-3498.

Gelman, A., Carlin, J. B., et al. *Bayesian Data Analysis*. CRC press, 2013.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., and Canu, S. Support Vector Machines with a Reject Option. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009.

Guo, C., Pleiss, G., et al. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Hein, M., Andriushchenko, M., et al. Why Relu Networks Yield High-Confidence Predictions Far away from the Training Data and How to Mitigate the Problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

Hendrycks, D., Basart, S., et al. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *ICCV*, 2021a.

Hendrycks, D., Zhao, K., et al. Natural Adversarial Examples. *CVPR*, 2021b.

Herbei, R. and Wegkamp, M. H. Classification with Reject Option. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 34(4):709–721, 2006. ISSN 0319-5724. Publisher: [Statistical Society of Canada, Wiley].

Jain, M., Lahlou, S., Nekoei, H., Butoi, V., Bertin, P., Rector-Brooks, J., Korablyov, M., and Bengio, Y. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*, 2021.

Kendall, A. and Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5574–5584, 2017.

Kobyzev, I., Prince, S., et al. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Lakshminarayanan, B., Pritzel, A., et al. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NIPS*, 2017.

LeCun, Y., Cortes, C., et al. MNIST Handwritten Digit Database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

LEI, J. Classification with confidence. *Biometrika*, 101(4): 755–769, 2014. ISSN 0006-3444. Publisher: [Oxford University Press, Biometrika Trust].

Levinson, J., Askeland, J., et al. Towards Fully Autonomous Driving: Systems and Algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 163–168. IEEE, 2011.

Liu, J. Z., Lin, Z., et al. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Malinin, A. and Gales, M. J. F. Predictive Uncertainty Estimation via Prior Networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7047–7058, 2018.

Malinin, A., Mlodozeniec, B., et al. Ensemble Distribution Distillation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Malkov, Y. A. and Yashunin, D. A. Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2018.

Miotto, R., Li, L., et al. Deep Patient: an Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports*, 6(1):1–10, 2016.

Miyato, T., Kataoka, T., et al. Spectral Normalization for Generative Adversarial Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Mukhoti, J., Kirsch, A., et al. Deterministic Neural Networks with Appropriate Inductive Biases Capture Epistemic and Aleatoric Uncertainty. *CoRR*, abs/2102.11582, 2021.

Nadeem, M. S. A., Zucker, J.-D., and Hanczar, B. Accuracy-Rejection Curves (ARCs) for Comparing Classification Methods with a Reject Option. In *Proceedings of the third International Workshop on Machine Learning in Systems Biology*, pp. 65–81. PMLR, March 2009. ISSN: 1938-7228.

Neal, R. M. et al. MCMC using Hamiltonian Dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

Netzer, Y., Wang, T., et al. Reading Digits in Natural Images with Unsupervised Feature Learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, A. M., Yosinski, J., et al. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 427–436, 2015.

Ovadia, Y., Fertig, E., et al. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *NeurIPS*, 2019.

Papamakarios, G., Nalisnick, E., et al. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Paszke, A., Gross, S., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Powell, J. L. Notes On Nonparametric Regression Estimation. *Manuscript*, 2010.

Rezende, D. and Mohamed, S. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015.

Sastry, C. S. and Oore, S. Detecting Out-of-Distribution Examples with Gram Matrices. In *ICML*, 2020.

Sensoy, M., Kaplan, L. M., et al. Evidential Deep Learning to Quantify Classification Uncertainty. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3183–3193, 2018.

Van Amersfoort, J., Smith, L., et al. Uncertainty Estimation using a Single Deep Deterministic Neural Network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

van Amersfoort, J., Smith, L., et al. Improving Deterministic Uncertainty Estimation in Deep Learning for Classification and Regression. *CoRR*, abs/2102.11409, 2021.

Wang, G., Li, W., et al. Aleatoric Uncertainty Estimation with Test-Time Augmentation for Medical Image Segmentation with Convolutional Neural Networks. *Neurocomputing*, 335:34 – 45, 2019.

Yu, F., Zhang, Y., et al. LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop. *CoRR*, abs/1506.03365, 2015.

A. Supplementary Material

A.1. Multiclass Generalization for Uncertainties

In this section we show, how our method can be generalized from binary classification to multiclass problems. Consider data pairs $(X, Y) \sim \mathbb{P}$. Now, $X \in \mathbb{R}^d$ and $Y \in \{1, \dots, C\}$, where C is the number of classes. We also denote $\eta_c(\mathbf{x}) = \mathbb{P}(Y = c \mid X = \mathbf{x})$.

Let us start with the Bayes risk:

$$\mathbb{P}(Y \neq g^*(X) \mid X = \mathbf{x}) = 1 - \mathbb{P}(Y = g^*(X) \mid X = \mathbf{x}) = 1 - \max_c \eta_c(\mathbf{x}) = \min_c \big\{1 - \eta_c(\mathbf{x})\big\},$$

where $g^*(\mathbf{x}) := \arg\max_c \eta_c(\mathbf{x})$ is the Bayes optimal classifier.

Let us further move to the excess risk and denote by $\hat{\eta}_c(\mathbf{x})$ some estimator of conditional probability. Analogously, $g(\mathbf{x}) := \arg\max_c \hat{\eta}_c(\mathbf{x})$ and we can bound the excess risk in the following way:

$$\begin{aligned}
&\mathbb{P}(Y \neq g(X) \mid X = \mathbf{x}) - \mathbb{P}(Y \neq g^*(X) \mid X = \mathbf{x}) = \eta_{g^*(\mathbf{x})}(\mathbf{x}) - \eta_{g(\mathbf{x})}(\mathbf{x}) \\
&= \eta_{g^*(\mathbf{x})}(\mathbf{x}) - \hat{\eta}_{g^*(\mathbf{x})}(\mathbf{x}) + \hat{\eta}_{g^*(\mathbf{x})}(\mathbf{x}) - \hat{\eta}_{g(\mathbf{x})}(\mathbf{x}) + \hat{\eta}_{g(\mathbf{x})}(\mathbf{x}) - \eta_{g(\mathbf{x})}(\mathbf{x}) \\
&\leq \big|\eta_{g^*(\mathbf{x})}(\mathbf{x}) - \hat{\eta}_{g^*(\mathbf{x})}(\mathbf{x})\big| + \big|\eta_{g(\mathbf{x})}(\mathbf{x}) - \hat{\eta}_{g(\mathbf{x})}(\mathbf{x})\big|,
\end{aligned}$$

where we used the fact that $\hat{\eta}_{g^*(\mathbf{x})}(\mathbf{x}) - \hat{\eta}_{g(x)}(\mathbf{x}) \leq 0$ for any $\mathbf{x}$.

The expectation of the right hand side in the case of kernel density estimator can be upper bounded by $2\sqrt{\frac{2}{\pi}}\tau(\mathbf{x})$, where

$$\tau^2(\mathbf{x}) = \frac{1}{N} \frac{\max_c \big\{\sigma_c^2(\mathbf{x})\big\}}{p(\mathbf{x})} \int \big[K_h(\mathbf{u})\big]^2 d\mathbf{u}$$

and $\sigma_c^2(\mathbf{x}) = \eta_c(\mathbf{x})\big(1 - \eta_c(\mathbf{x})\big)$. Total uncertainty for multiclass problem is thus

$$\mathbf{U}_t(\mathbf{x}) = \min_c \big\{1 - \eta_c(\mathbf{x})\big\} + 2\sqrt{\frac{2}{\pi}}\tau(\mathbf{x}).$$

A.2. Consistency of NUQ-based Classification with reject option

In this section we derive a non-asymptotic upper bound of excess risk used to obtain the consistency result in Subsection 2.4. First, using results of the previous section, notice, that for an arbitrary rejection rule $\alpha(\mathbf{x})$ the excess risk of $\mathcal{R}_\lambda(\mathbf{x})$ is at most

$$\mathbb{E}_{\mathcal{D}} \left\{\mathcal{R}_\lambda(\mathbf{x}) - \mathcal{R}^*_\lambda(\mathbf{x})\right\} \leqslant 2\mathbb{E}_{\mathcal{D}} \left\{\max_{c \in C} |\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x})| \mathbb{1}\{\alpha(\mathbf{x}) = 0\}\right\} + |\lambda - \mathcal{R}^*(\mathbf{x})| \mathbb{P}_{\mathcal{D}} \left(\alpha(\mathbf{x}) \neq \alpha^*(\mathbf{x})\right).$$

As previously,

$$\frac{\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x})}{\tau_c(\mathbf{x})} \to \mathcal{N}(0, 1), \quad \tau_c(\mathbf{x}) = \|K\|_2 \sqrt{\frac{\eta_c(\mathbf{x})(1 - \eta_c(\mathbf{x}))}{Nh^d p(\mathbf{x})}}.$$

Thus, asymptotically

$$\mathbb{P}\left(\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x}) \leqslant z_\beta \|K\|_2 \sqrt{\frac{\eta_c(\mathbf{x})(1 - \eta_c(\mathbf{x}))}{Nh^d \hat{p}(\mathbf{x})}}\right) \leqslant \beta.$$

Consequently,

$$\mathbb{P}\left(\min_c \{1 - \hat{\eta}_c(\mathbf{x})\} \leqslant \lambda - \max_c z_{1-\beta} \|K\|_2 \sqrt{\frac{\hat{\eta}_c(\mathbf{x})(1 - \hat{\eta}_c(\mathbf{x}))}{Nh^d \hat{p}(\mathbf{x})}}\right) \leqslant \beta |C|.$$

Algorithm 2 Acceptance testing for classification

Input: Samples (X_i, Y_i), bandwidth h, parameters λ, β
Output: Accept or reject the regression result

Calculate $\hat{p}(\mathbf{x}), \hat{\eta}_c(\mathbf{x})$ for $c \in C$;
if *the density estimation* $\hat{p}(\mathbf{x}) > 0$ *and the criterion holds:*

$$\min_c(1 - \hat{\eta}_c(\mathbf{x})) \leqslant \lambda - \max_c z_{1-\beta/|\mathcal{C}|}\|K\|_2\sqrt{\frac{\hat{\eta}_c(\mathbf{x})(1-\hat{\eta}_c(\mathbf{x}))}{Nh^d\hat{p}(\mathbf{x})}}$$

then
| Accept results of the regression
else
| Reject
end

That leads us to the procedure described as Algorithm 2.

We formulate a number of mild assumptions:

Assumption A.1. There exist Hessians of functions $\eta_c(\mathbf{x})$, $c \in \mathcal{C}$, and their spectral norms are bounded by some constant M_η.

Assumption A.2. L_2-norms of ∇p and $\nabla^2 p$ are bounded by constants L_d and M_d respectively, i.e.

$$\|\nabla p(\mathbf{x})\|_2 = \sqrt{\sum_i (\nabla_i p(x))^2} \leqslant L_d, \quad \sup_{\mathbf{x}:\|\mathbf{x}\|_2=1} \|\mathbf{x}^T\nabla^2 p(\mathbf{x})\|_2 \leqslant M_d.$$

Assumption A.3. There exist finite values

$$\max_{\mathbf{t}} K(\mathbf{t}), \quad \int_{\mathbb{R}^d} \|\mathbf{t}\|_2 K^2(\mathbf{t})d\mathbf{t}, \quad \int_{\mathbb{R}^d} K^2(\mathbf{t})d\mathbf{t}, \quad \int_{\mathbb{R}^d} \|\mathbf{t}\|_2^2 K(\mathbf{t})d\mathbf{t},$$

while

$$\int_{\mathbb{R}^d} \mathbf{t}K(\mathbf{t})d\mathbf{t} = 0.$$

Under these assumptions, we state the following theorem:

Theorem A.4. *Suppose that assumptions A.1-A.3 hold,* $p(\mathbf{x}) > 0$*, and* $\beta < 1/2$*. Define* $\Delta = |\lambda - \mathcal{R}^*(\mathbf{x})|$*. Then, if* $\lambda < \mathcal{R}^*(\mathbf{x})$

$$\mathbb{E}_{\mathcal{D}}\left\{\mathcal{R}_\lambda(\mathbf{x}) - \mathcal{R}^*_\lambda(\mathbf{x})\right\} \leqslant |\mathcal{C}|A_\Delta,$$

where

$$A_\Delta = 2\left\{R \wedge (\mathbb{1}\{\Delta \leqslant h^2\kappa_\Delta\} \vee q_\Delta)\right\} + \Delta(\mathbb{1}\{\Delta \leqslant h^2\kappa_\Delta\} \vee q_\Delta),$$

$$R = 2h^2\kappa_\Delta + 2\sqrt{\frac{\pi\left\{12\left(\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|_2 K^2(\mathbf{t})d\mathbf{t}\right) + \max_{\mathbf{t}} K(\mathbf{t})\right\}}{Nh^d p(\mathbf{x})}},$$

$$q_\Delta = \exp\left(-\frac{1}{2}\frac{Nh^d p(\mathbf{x})(\Delta - h^2\kappa_\Delta)^2}{(1+\Delta)^2\left(\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|K^2(\mathbf{t})d\mathbf{t}\right) + \frac{1}{3}(\Delta - h^2\kappa_\Delta)\max_{\mathbf{t}} K(\mathbf{t})}\right),$$

and

$$\kappa_\Delta = \frac{1}{p(\mathbf{x})}\left(M_d + 2dM_\eta L_d + 2dL_d\sqrt{M_\eta}\right)\int_{\mathbb{R}^d}\|t\|_2^2 K(\mathbf{t})d\mathbf{t}.$$

If $\mathcal{R}^*(\mathbf{x}) \leqslant \Delta$,

$$\mathbb{E}_{\mathcal{D}}\mathcal{R}(\mathbf{x}) - \mathcal{R}^*(\mathbf{x}) \leqslant |\mathcal{C}| \left(R + \mathbb{1}\left\{ \Delta \leqslant h^2\kappa_\Delta + z_{1-\beta/|\mathcal{C}|}\|K\|_2 \sqrt{\frac{1}{2Nh^2p(\mathbf{x})}} \right\} \vee \tilde{q}_\Delta + \mathbb{1}\left\{1/2 \leqslant h^2\kappa_p\right\} \vee q_p \right).$$

Here $\tilde{q}_\Delta$ *differs from* q_Δ *by replacing* $h^2\kappa_\Delta$ *with* $h^2\kappa_\Delta + z_{1-\beta/|\mathcal{C}|}\|K\|_2\sqrt{\frac{1}{2Nh^2p(\mathbf{x})}}$, *while*

$$q_p = \exp\left(-\frac{\frac{1}{2}Nh^2p(\mathbf{x})(1/2 - h^2\kappa_p)^2}{\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|_2 K^2(\mathbf{t}) + \max_{\mathbf{t}} K(\mathbf{t})(1/2 - h^2\kappa_p)} \right),$$
$$\kappa_p = \frac{M_d}{2p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|^2 K(\mathbf{t})d\mathbf{t}.$$

Notice, that Theorem 2.1 follows from Theorem A.4 as all the terms in the upper bound tend to zero when $h \to 0$ and $Nh^d \to \infty$ with $N \to \infty$.

Proof of Theorem A.4. For a reminder, the excess risk is

$$\mathbb{E}_{\mathcal{D}}\left\{\mathcal{R}_\lambda(\mathbf{x}) - \mathcal{R}^*_\lambda(\mathbf{x})\right\} \leqslant 2\mathbb{E}_{\mathcal{D}}\left\{\max_{c\in\mathcal{C}} |\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x})|\mathbb{1}\{\alpha(\mathbf{x}) = 0\}\right\} + \Delta \cdot \mathbb{P}_{\mathcal{D}}\left(\alpha(\mathbf{x}) \neq \alpha^*(\mathbf{x})\right).$$

First, rewrite the expectation as an integral

$$\begin{aligned}
\mathbb{E}|\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})|\mathbb{1}\{\alpha(\mathbf{x}) = 0\} &= \int_0^{+\infty} \mathbb{P}(|\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})|\mathbb{1}\{\alpha(\mathbf{x}) = 0\} \geqslant t)dt \\
&= \int_0^{+\infty} \mathbb{P}\left(|\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})| \geqslant t \text{ and } \alpha(\mathbf{x}) = 0\right) dt \\
&\leqslant \int_0^{+\infty} \mathbb{P}\left(|\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})| \geqslant t \text{ and } \sum_i K_h(X_i - \mathbf{x}) \neq 0\right) dt \\
&\leqslant \int_0^{1} \mathbb{P}\left(|\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})| \geqslant t \text{ and } \sum_i K_h(X_i - \mathbf{x}) \neq 0\right) dt,
\end{aligned}$$

since we abstain if $\hat{p}(\mathbf{x}) = 0$. Due to Lemma A.5,

$$\begin{aligned}
&\int_0^{1} \mathbb{P}\left(|\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})| \geqslant t \text{ and } \sum_i K_h(X_i - \mathbf{x}) \neq 0\right) dt \leqslant \\
&\leqslant 2h^2\kappa + 2\sqrt{\frac{\pi\left\{\left(\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|_2 K^2(\mathbf{t})d\mathbf{t}\right) + \max_{\mathbf{t}} K(\mathbf{t})\right\}}{2Nh^dp(\mathbf{x})}}.
\end{aligned}$$

Here we use the Poisson integral. Denote this upper bound by R.

Now, assume $\lambda < \mathcal{R}^*(\mathbf{x})$**.** Then the excess risk can be estimated in the following way:

$$\begin{aligned}
&2\mathbb{E}_{\mathcal{D}}\left\{\max_{c\in C} |\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x})|\mathbb{1}\{\alpha(\mathbf{x}) = 0\}\right\} \leqslant 2\left[R \wedge \mathbb{P}\left(\alpha(\mathbf{x}) = 0\right)\right], \\
&\Delta \cdot \mathbb{P}\left(\alpha(\mathbf{x}) \neq \alpha^*(\mathbf{x})\right) = \Delta\mathbb{P}\left(\alpha(\mathbf{x}) = 0\right), \\
&\mathbb{P}(\alpha(\mathbf{x}) = 0) = \mathbb{P}\left(\sum_i K_h(X_i - \mathbf{x}) > 0 \text{ and } \min_c(1 - \hat{\eta}_c(\mathbf{x})) \leqslant \lambda - z_{1-\beta/|\mathcal{C}|}\|K\|_2\sqrt{\frac{\hat{\eta}(x)(1-\hat{\eta}(x))}{Nh^d\hat{p}(\mathbf{x})}}\right).
\end{aligned}$$

The event from the RHS implies that there is $c \in C$ such that

$$\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x}) \geqslant \Delta, \tag{4}$$

and, consequently,

$$\mathbb{P}(\alpha(\mathbf{x}) = 0) \leqslant \sum_{c \in \mathcal{C}} \mathbb{P}\left(\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x}) \geqslant \Delta \text{ and } \sum_i K_h(X_i - \mathbf{x}) > 0\right).$$

The upper bound was obtained using Lemma A.5.

Finally, consider the case $\mathcal{R}^*(\mathbf{x}) \geqslant \lambda$. Then, we estimate

$$\mathbb{E}_{\mathcal{D}}\{\max_c |\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x})| \mathbb{1}\{\alpha(\mathbf{x}) = 0\}\} \leqslant R|\mathcal{C}|,$$

and

$$\begin{aligned}
\mathbb{P}(\alpha(\mathbf{x}) = 1) &\leqslant \sum_{c \in \mathcal{C}} \mathbb{P}\left(\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x}) \geqslant \Delta - z_{1-\beta/|\mathcal{C}|}\|K\|_2 \max_c \sqrt{\frac{\hat{\eta}_c(\mathbf{x})(1-\hat{\eta}_c(\mathbf{x}))}{Nh^d \hat{p}(\mathbf{x})}} \text{ or } \hat{p}(\mathbf{x}) = 0\right) \\
&\leqslant \sum_{c \in \mathcal{C}} \mathbb{P}\left(\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x}) \geqslant \Delta - z_{1-\beta/|\mathcal{C}|}\|K\|_2 \sqrt{\frac{1}{2Nh^d p(\mathbf{x})}} \text{ and } \hat{p}(\mathbf{x}) > 0\right) \\
&+ |\mathcal{C}| \mathbb{P}\left(\hat{p}(\mathbf{x}) \leqslant \frac{p(\mathbf{x})}{2}\right).
\end{aligned}$$

Similarly to Lemma A.5, we bound the last probability by Bernstein's inequality:

$$\mathbb{P}\left(\hat{p}(\mathbf{x}) \leqslant \frac{p(\mathbf{x})}{2}\right) \leqslant \mathbb{1}\left\{\frac{1}{2} \leqslant h^2 \kappa_p\right\} \vee \exp\left(-\frac{\frac{1}{2}h^d N p(\mathbf{x})\left(\frac{1}{2} - h_p^\kappa\right)}{\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|_2 K^2(\mathbf{t}) + \max_{\mathbf{t}} K(\mathbf{t})(1/2 - h^2\kappa_p)}\right).$$

□

Lemma A.5. *Suppose all conditions of Theorem A.4 holds. Then, for any non-negative r*

$$\begin{aligned}
&\mathbb{P}(\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x}) \geqslant r \textit{ and } \sum_i K_h(X_i - \mathbf{x}) > 0) \leqslant \\
&\leqslant \mathbb{1}\{r \leqslant h^2\kappa\} \vee \exp\left\{-\frac{1}{2}\frac{Nh^d p(\mathbf{x})(r - h^2\kappa)^2}{(1+r)^2\left(\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|_2 K^2(\mathbf{t})d\mathbf{t}\right) + (\max_{\mathbf{t}} K(\mathbf{t}) + r)|r - h^2\kappa|}\right\}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}(\eta_c(\mathbf{x}) - \hat{\eta}_c(\mathbf{x}) \geqslant r \textit{ and } \sum_i K_h(X_i - \mathbf{x}) > 0) \leqslant \\
&\leqslant \mathbb{1}\{r \leqslant h^2\kappa\} \vee \exp\left\{-\frac{1}{2}\frac{Nh^d p(\mathbf{x})(r - h^2\kappa)^2}{(1+r)^2\left(\|K\|_2^2 + \frac{hL_d}{p(\mathbf{x})}\int_{\mathbb{R}^d}\|\mathbf{t}\|_2 K^2(\mathbf{t})d\mathbf{t}\right) + (\max_{\mathbf{t}} K(\mathbf{t}) + r)(r - h^2\kappa)}\right\}.
\end{aligned}$$

Proof. Let us prove the first inequality, the second one can be proved in the same way. Since $\hat{p}(\mathbf{x}) > 0$, we can multiply both sides of the inequality $\hat{\eta}_c(\mathbf{x}) - \eta_c(\mathbf{x}) \geqslant r$ by $\sum_i K_h(X_i - \mathbf{x})$. Thus, the inner event implies

$$\sum_i \mathbb{1}\{Y_i = c\}K_h(X_i - \mathbf{x}) - \eta_c(\mathbf{x})K_h(X_i - \mathbf{x}) - rK_h(X_i - \mathbf{x}) \geqslant 0.$$

Define

$$\begin{aligned}
e_i &= \mathbb{1}\{Y_i = c\}K_h(X_i - \mathbf{x}) - \eta_c(\mathbf{x})K_h(X_i - \mathbf{x}) - rK_h(X_i - \mathbf{x}), \\
e &= \mathbb{E}e_i = \mathbb{E}\{\eta_c(X_i) - \eta_c(\mathbf{x})\}K_h(X_i - \mathbf{x}) - r \cdot \mathbb{E}K_h(X_i - \mathbf{x}).
\end{aligned}$$

In order to write a concentration we should analyze e. Inequalities

$$\|\nabla\eta_c(\mathbf{x}) - \nabla\eta_c(\mathbf{y})\| \leqslant \sqrt{d}M_\eta\|\mathbf{x}-\mathbf{y}\|,$$
$$\left|\int_0^\lambda \nabla_i\eta_c(\mathbf{x}+se_i)ds\right| \leqslant |\eta_c(\mathbf{x}+\lambda e_i) - \eta_c(\mathbf{x})| \leqslant 1,$$

which hold for each λ, $\mathbf{x}$ and $\mathbf{y}$, guarantee us that the norm of the gradient $\nabla\eta_c(\mathbf{x})$ is bounded by

$$L_\eta = 2d\sqrt{M_\eta}.$$

Moreover, non-negativity of $p(\mathbf{x})$, the L_d-Lipschitz property and its L_1-norm imply that $p(\mathbf{x})$ is bounded by $2L_d d$. Then, Taylor's expansion delivers the following:

$$\begin{aligned}
|\mathbb{E}\{\eta_c(X_1) - \eta_c(\mathbf{x})\}K_h(X_1-\mathbf{x})| &= \left|\int_{\mathbb{R}^d}(\eta(\mathbf{x}') - \eta(\mathbf{x}))K_h(\mathbf{x}'-\mathbf{x})p(\mathbf{x}')d\mathbf{x}'\right| \\
&\leqslant \left|h\int_{\mathbb{R}^d}\langle\nabla\eta(\mathbf{x}), t\rangle K(t)p(\mathbf{x}+ht)dt\right| + h^2 dM_\eta L_d\int_{\mathbb{R}^d}\|\mathbf{t}\|_2^2 K(\mathbf{t})d\mathbf{t} \\
&\leqslant h^2 L_d d\left(\sqrt{M_\eta} + M_\eta\right)\int_{\mathbb{R}^d}\|\mathbf{t}\|_2^2 K(\mathbf{t})d\mathbf{t}.
\end{aligned}$$

Similarly,

$$|\mathbb{E}K_h(X_i-\mathbf{x}) - p(\mathbf{x})| \leqslant \frac{h^2}{2}M_d\int_{\mathbb{R}^d}\|\mathbf{t}\|_2^2 K(\mathbf{t})d\mathbf{t}.$$

Thus,

$$(-e) \geqslant p(\mathbf{x})r - \frac{h^2}{2}\left(M_d + 2dM_\eta L_d + 2dL_d\sqrt{M_\eta}\right)\int_{\mathbb{R}^d}\|\mathbf{t}\|_2^2 K(\mathbf{t})d\mathbf{t} = p(\mathbf{x})(r - h^2\kappa).$$

If $e > 0$ we estimate the probability by 1. Otherwise, we utilize Bernstein's inequality:

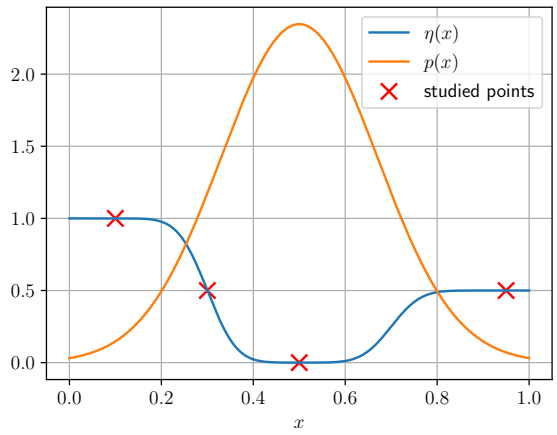
$$\begin{aligned}
\mathbb{P}\left(\sum_i e_i - Ne \geqslant N(-e)\right) &\leqslant \exp\left(-\frac{\frac{1}{2}N^2e^2}{\sum_i \operatorname{Var} e_i + \frac{1}{3}h^{-d}\max_t K(t)N(-e)}\right) \\
&\leqslant \exp\left(-\frac{\frac{1}{2}Ne^2}{\mathbb{E}e_1^2 + \frac{1}{3}h^{-d}\max_t K(t)(-e)}\right) \\
&\leqslant \exp\left(-\frac{\frac{1}{2}Ne^2}{(1+r)^2\mathbb{E}K_h^2(X_i-\mathbf{x}) + h^{-d}(\max_{\mathbf{t}} K(\mathbf{t}) + r)(-e)}\right). \qquad (5)
\end{aligned}$$

We estimate $\mathbb{E}K_h^2(X_i-\mathbf{x})$ as follows:

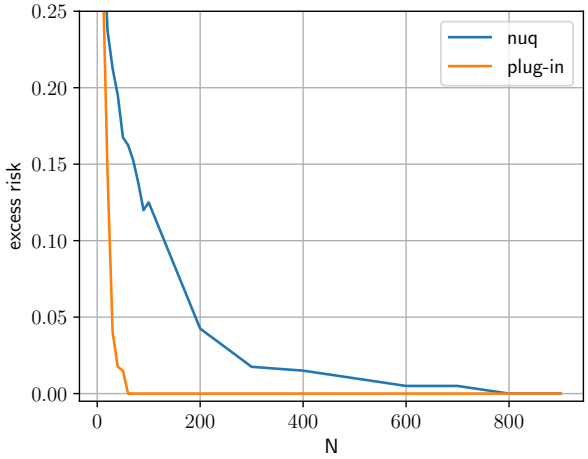
$$\mathbb{E}K_h^2(X_i-\mathbf{x}) = h^{-d}\int_{\mathbb{R}^d}K^2(\mathbf{t})p(\mathbf{x}+h\mathbf{t})dt \leqslant h^{-d}\left(p(\mathbf{x})\|K\|_2^2 + hL_d\int_{\mathbb{R}^d}\|\mathbf{t}\|K^2(\mathbf{t})d\mathbf{t}\right).$$

□

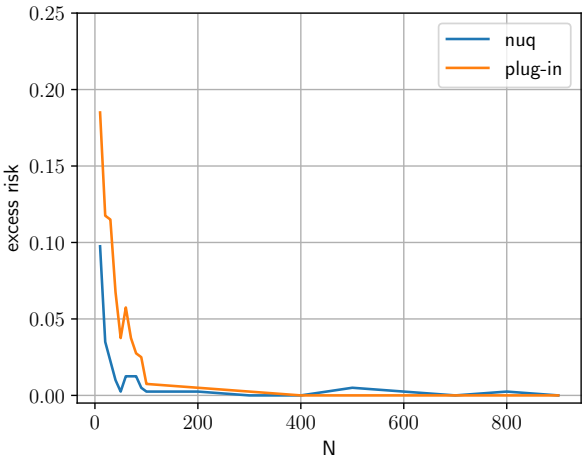
For illustration, we present some experimental results, see Figure 4. We consider smoothed step function as $\eta(x)$, while data X_i is distributed according to the normal distribution with mean 0.5 and variance 0.04 (see Figure 4a). We study point-wise excess risk of NUQ and the plug-in rule. For the points with high covariate mass (Figures 4c and 4d) methods show comparable results. NUQ is useful for points lying with low covariate density, see Figure 4e. However, for points without any label noise (Figure 4b) plug-in is still better as it quickly learns the correct class while NUQ is more conservative.



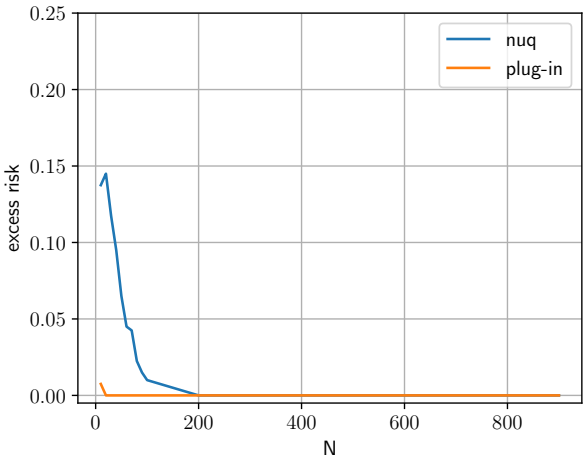
(a) data



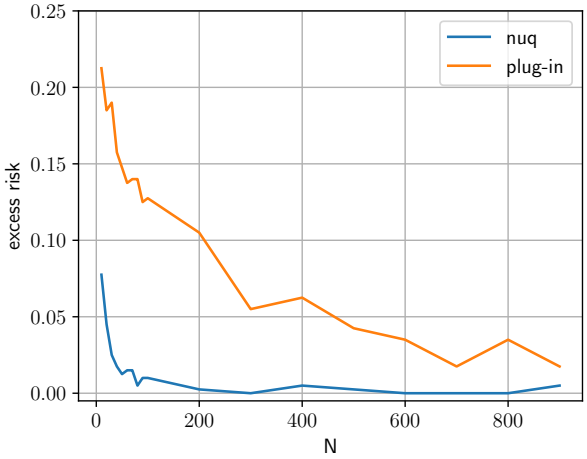
(b) $x = 0.1$



(c) $x = 0.3$



(d) $x = 0.5$



(e) $x = 0.95$

Figure 4: Excess risk at studied points marked on Figure 4a. We choose RBF-kernel and quantile $1 - \beta$ to be equal 0.95.

A.3. Choice of Kernel for Uncertainty Quantification

In this section, we study the choice of a kernel for uncertainty quantification. First, let us rewrite $K_h(\mathbf{u})$ as follows:

$$K_h(\mathbf{u}) = \prod_{i=1}^{d} K\left(\frac{u_i}{h}\right) = \prod_{i=1}^{d} K(z_i).$$

We consider the different choices of kernels, see Table 3.

Kernel name	**Formula** $K(z)$	**Integral** $\int K_h(\mathbf{u})^2 d\mathbf{u}$
Gaussian (RBF)	$\frac{1}{\sqrt{2\pi}} \exp\left\{-z^2\right\}$	$\frac{h^d}{2\sqrt{\pi}}$
Sigmoid	$\frac{2}{\pi} \frac{1}{\exp\{-z\}+\exp\{z\}}$	$\frac{2h^d}{\pi^2}$
Logistic	$\frac{1}{\exp\{-z\}+2+\exp\{z\}}$	$\frac{h^d}{6}$

Table 3: Different types of kernels $K(z)$ considered and corresponding values of the integral $\int \widetilde{K_h}(\mathbf{u})^2 d\mathbf{u}$.

A.4. Architectures and Hyperparameters

Base Model. For CIFAR-100 and ImageNet-like datasets, we are using ResNet50 as a base model, with or without spectral normalization (Miyato et al., 2018). For the spectral normalization, we use 3 iterations of the power method. We use a ResNet50 architecture with implementation from PyTorch (Paszke et al., 2019). This architecture was implemented for the ImageNet dataset; thus, for the CIFAR-100, we had to adapt it. We changed the first convolutional layer and used kernel size 3x3 with stride 1 and padding 1 (instead of kernel size 7x7 with stride 2 and padding 3 used for ImageNet). For CIFAR-100, we train the model for 200 epochs with an SGD optimizer, starting with a learning rate of 0.1 and decaying it 5 times on 60, 120, and 160 epoch. For ImageNet, we train the model for 90 epochs with an SGD optimizer and learning rate decaying 10 times every 30 epochs.

For MNIST, we train a small convolutional neural network with three convolutional layers with padding of 1 and kernel size of 3. Each of these layers is followed by a batch normalization layer. Finally, it has a linear layer with Softmax activation. This network achieves an accuracy of 0.99 on the holdout set.

We refer readers to our code for more specific details.

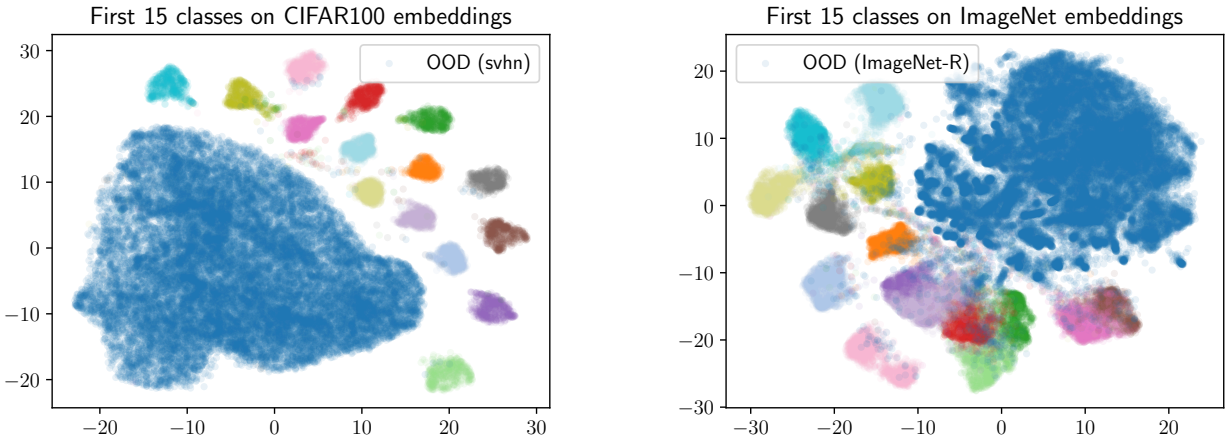
Ensemble. For ensemble we use a combination of 5 base models trained with different random seeds.

Test-Time Augmentation (TTA). For TTA, we use a base model and apply different transformations on the inference stage. Images of CIFAR-100 are randomly cropped with padding 4, randomly horizontally flipped, and randomly rotated up to 15 degrees. ImageNet is randomly cropped from 256 to 224, randomly horizontally flipped, and the color was jittered (0.02).

Spectrally Normalized Models. For SNGP, DDU and NUQ, we need spectral normalized models to extract features. We wrapped each convolutional and linear layer with spectral normalization (PyTorch implementation). We used 3 iterations of the power method in our experiments.

A.5. Performance difference on CIFAR-100 and ImageNet

One of the things that caught our attention is the superior performance of NUQ on ImageNet, given that it has very similar results with DDU on CIFAR-100. One of our hypotheses was that embeddings have a more complex and multi-modal distribution for a more complex ImageNet dataset compared to simpler CIFAR-100. To check this, we made t-SNE based embeddings of out-of-distribution and test data (see Figure 5). While we understand the limitation of this type of visualization, the ImageNet embeddings appear to be much more irregular compared to well-shaped clusters for CIFAR-100. Because of that, the modeling of the class with a single Gaussian (in DDU) might not work very well for ImageNet. NUQ approach performs the modeling of distributions in a much more flexible way, which is beneficial for approximating complex distributions. We hypothesize that this is the reason for the NUQ's superior performance.



(a) (b)

Figure 5: Embeddings space visualization for CIFAR (a) and ImageNet (b). We present the embeddings for first 15 classes on test dataset (in various colors) and all the embeddings for out-of-distribution datasets (in blue). The OOD dataset for CIFAR is SVHN and for ImageNet it is ImageNet-R.

A.6. Toy experiment on Detecting Actual Aleatoric and Epistemic Uncertainties

In this section, we conduct a toy experiment, for which we explicitly know what should be the true probability of class one, as well as the true data density.

Let us consider a binary classification problem. Our dataset consists of 5000 samples from three different one-dimensional Gaussians, located to mix classes. Colors denote class label: red - 0; green - 1 (see Figure 6a). For this particular data model, we can compute the conditional probability of a data point x belongs to class 1: $\eta(x) = p(Y = 1 \mid X = x)$. We build an estimate of this conditional using our Nadaraya-Watson kernel-based approach $\hat{\eta}(x)$. Further, we generate a uniform grid, and for each point of this grid, using our method, we can upper bound difference between the true conditional and our approximation. This difference, according to our approach, is considered as an epistemic uncertainty (see Figure 6b). The green line in this plot denotes an absolute difference between the true conditional and our approximation. The red line denotes our epistemic uncertainty. From the picture, we can see that our epistemic uncertainty approximates the probabilities difference well. Next, we show how our aleatoric uncertainty relates to the true class 1 conditional probability. In the Figure 6c we show true conditional distribution $\eta(x)$ (orange line) and our approximation of the aleatoric uncertainty $\min\{\hat{\eta}(x), 1 - \hat{\eta}(x)\}$ (red line). We can see that our approximation is high exactly in the same regions where the true conditional is absolutely unsure about the class label.

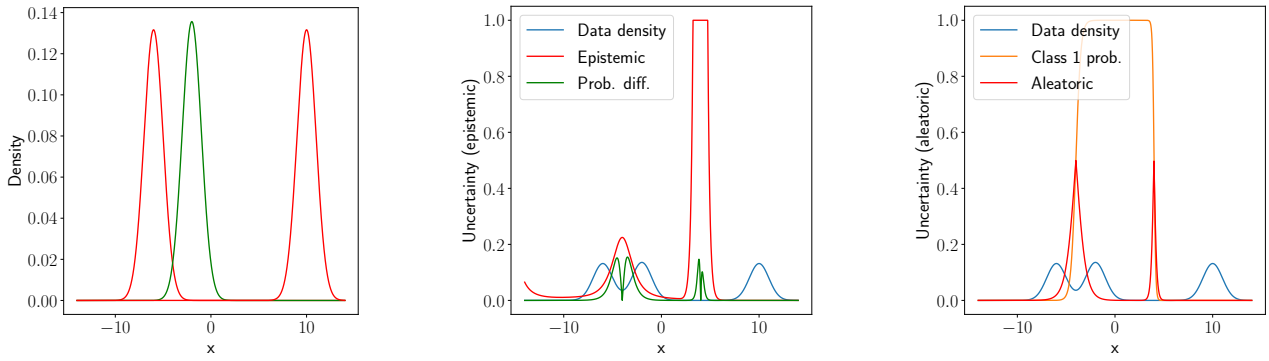
A.7. Additional Experiments on Image Datasets

A.7.1. Ablation Study on CIFAR-100

We need an estimator of marginal density $p(\mathbf{x})$ for our method, and there exist different options. We consider kernel method with RBF kernel and logistic kernel and Gaussian mixtures model (GMM). There is also a question about which embeddings to use – the DDU paper (Mukhoti et al., 2021) proposes to take the features from the second last layer; while logits from the last layer represent a reasonable choice as well. To validate the options, we conducted some ablation study on out-of-distribution detection for the CIFAR-100 dataset, similar to the main experiment.

First, we compare the DDU and NUQ on embeddings from the second last and last layers (Table 4) on SVHN, LSUN, and Smooth datasets. Secondly, we compare the NUQ method on RBF, logistic kernel, and GMM for both last and penultimate layer embeddings (Table 5). As we can see from the tables, the optimal option is GMM density on the penultimate layer.

Kernel-based methods rely on the "reasonable" geometry of the embedding space, meaning that embeddings of similar images should be not too far and different images should not collapse into a single point. Our motivation to use spectral



(a) (b) (c)

Figure 6: (a): Mixture of one dimensional Gaussians we took samples from. Color denotes class label. (b): Epistemic uncertainty our model assigns to data points. Note that the uncertainty is quite high in the region of 3-5. For the sake of visualization, we clipped the maximum value to be 1. (c): Our approximation of aleatoric uncertainty is built along with the true conditional probability.

	DDU, features	DDU, logits	NUQ, features	NUQ, logits
SVHN	89.6±1.6	88.2±0.6	89.7±1.6	88.2±0.6
LSUN	92.1±0.6	90.9±0.4	92.3±0.6	90.9±0.4
Smooth	97.1±3.1	96.3±4.1	96.8±3.8	96.2±4.1

Table 4: Comparison of DDU and NUQ predictions on different type of embeddings: logits (last layer) and features (second last layer).

normalization during training is to make the embedding space smooth with respect to input images. We have conducted an extra ablation study, comparing the result for feature extractors with and without spectral normalization, see Table 6. The results confirm our hypothesis, as the spectral-normalized version performs better, though the NUQ beats the baseline even without applying the modification to the ResNet training. We also show here that entropy performs better than maximum probability as an uncertainty measure.

A.7.2. Sensitivity to Ensemble Size

In this section, we explore the ensemble model performance regarding the number of models. From Table 7, we can see that 5 models is a reasonable amount number for CIFAR-100. For the ImageNet dataset (Table 8), increasing the number of models gives a steady gain, but still 5 models provide the gain within the error margin.

A.7.3. Additional experiments with DUQ

DUQ (Van Amersfoort et al., 2020) is one of the baselines in the paper. We chose it as similarly to our method it requires only a single forward pass and uses post-processing for embeddings. In the original article, the method shows a good result on a relatively small dataset CIFAR-10 with a ResNet18 model. We tried to train it on CIFAR-100 and ImageNet with a larger model, ResNet50, but there were difficulties with training process as method failed to converge to the model of reasonable quality. We believe the cause of the problem was gradient penalty as a regularization method, and we switched to spectral normalization instead. DUQ training requires a balance between hyperparameters such as length scale, momentum, and learning rate, so we initially trained the model with a pre-trained feature extractor. With careful selection of parameters, we managed to train end-to-end as well, and we observed improvement in all experiments (see Tables 9 and 10), although methods like DDU and NUQ had more stable training in our experience and a better final result.

	RBF, f	RBF, l	Logistic, f	Logistic, l	GMM, f	GMM, l
SVHN	84.4±3.2	84.7±3.1	84.8±2.9	86.7±2.6	89.7±1.6	88.2±0.6
LSUN	88.2±1.0	88.1±0.8	88.5±4.0	90.3±1.0	92.3±0.6	90.9±0.4
Smooth	85.5±6.8	87.7±9.4	86.2±8.2	90.8±7.8	96.8±3.8	96.2±4.1

Table 5: Probability density methods comparison – radial basis function kernel (RBF), logistic kernel, Gaussian mixture models (GMM). 'f' (features) marks models, built on embeddings from a second last layer and 'l' (logits) is for the ones built on embeddings from a last layer.

OOD dataset	DDU	DDU (spectral)	NUQ	NUQ (spectral)
SVHN	88.7±4.3	89.6±1.6	86.8±1.2	89.7±1.6
LSUN	91.3±0.9	92.1±0.6	91.2±1.1	92.3±0.6
Smooth	95.7±1.2	97.1±3.1	95.5±1.3	96.8±3.8

Table 6: Comparing the influence of spectral normalization on the model performance for OOD detection, ROC-AUC.

OOD dataset	2	3	5	7	10
SVHN	82.3 ± 1.3	82.4 ± 0.7	82.9 ± 0.9	82.7 ± 0.7	82.6 ± 0.5
LSUN	85.1 ± 0.6	85.9 ± 0.6	86.5 ± 0.8	87.1 ± 0.6	87.1 ± 0.6
Smooth	83.7 ± 6.5	83.4 ± 3.2	83.7 ± 1.2	83.3 ± 1.5	83.2 ± 1.6

Table 7: Ablation study for ensemble size on CIFAR100 in-distribution dataset for OOD detection. The number of models above 5 give almost no gain (even some loss sometimes) within an error margin.

OOD dataset	2	3	5	7	9
ImageNet-O	49.8	50.8	51.9	52.1	52.6
ImageNet-R	84.7	85.3	85.8	86	86.1

Table 8: Ablation study for ensemble size on ImageNet out-of-distribution detection task.

OOD dataset	Ensembles	TTA	DDU	NUQ	DUQ Head	DUQ end-to-end
SVHN	82.9 ± 0.9	81.6 ± 1.2	89.6 ± 1.6	**89.7 ± 1.6**	83.6 ± 4.0	88.7 ± 6.3
LSUN	86.5 ± 0.8	85.0 ± 2.7	92.1 ± 0.6	**92.3 ± 0.6**	87.2 ± 2.1	90.8 ± 6.7
Smooth	83.7 ± 1.2	73.2 ± 10.8	**97.1 ± 3.1**	96.8 ± 3.8	83.8 ± 11.4	91.1 ± 8.4

Table 9: Performance on OOD detection for CIFAR-100

OOD dataset	Ensemble	DDU*	NUQ (spectral)*	DUQ Head	DUQ end-to-end
ImageNet-R	84.4	74.2	**99.5**	57.4	73.3
ImageNet-O	51.9	74.1	**82.4**	67.3	71.4

Table 10: Performance on OOD detection on ImageNet