Piecewise Linear Transformation – Propagating Aleatoric Uncertainty in Neural Networks

Thomas Krapf, Michael Hagn, Paul Miethaner, Alexander Schiller, Lucas Luttner, Bernd Heinrich
Faculty for Computer Science and Data Science, University of Regensburg
{Thomas.Krapf, Michael.Hagn, Paul.Miethaner, Alexander.Schiller, Lucas.Luttner, Bernd.Heinrich}@ur.de

Abstract

Real-world data typically exhibit aleatoric uncertainty which has to be considered during data-driven decision-making to assess the confidence of the decision provided by machine learning models. To propagate aleatoric uncertainty represented by probability distributions (PDs) through neural networks (NNs), both sampling-based and function approximation-based methods have been proposed. However, these methods suffer from significant approximation errors and are not able to accurately represent predictive uncertainty in the NN output. In this paper, we present a novel method, Piecewise Linear Transformation (PLT), for propagating PDs through NNs with piecewise linear activation functions (e.g., ReLU NNs). PLT does not require sampling or specific assumptions about the PDs. Instead, it harnesses the piecewise linear structure of such NNs to determine the propagated PD in the output space. In this way, PLT supports the accurate quantification of predictive uncertainty based on the criterion exactness of the propagated PD. We assess this exactness in theory by showing error bounds for our propagated PD. Further, our experimental evaluation validates that PLT outperforms competing methods on publicly available real-world classification and regression datasets regarding exactness. Thus, the PDs propagated by PLT allow to assess the uncertainty of the provided decisions, offering valuable support.

Introduction

Neural networks (NNs) have been deployed in data-driven decision-making in many tasks and fields, such as autonomous driving (Chen et al. 2021; Huang et al. 2021), medical diagnostics (Takenaka et al. 2020; Yu et al. 2021), cyber security (Pawlicki, Kozik, and Choraś 2022; Vigneswaran et al. 2018), industrial processes (Nunez et al. 2020; Zhang et al. 2021), and many more. However, despite being applied in high-risk and safety-critical domains, traditional NNs only yield deterministic point estimates without further valid information about the confidence or uncertainty of their predictions (Gal 2016; Ayhan and Berens 2018). For example, this is particularly relevant for cyber threat intelligence data, which typically suffer from a high number of missing or uncertain values and extreme class imbalance (because actual cyber-attacks are rather scarce). Indeed, inherent scores such as the softmax output of a NN have been shown to be no valid measure for the confidence of the prediction (Hendrycks and Gimpel 2017; Nguyen, Yosinski, and Clune 2015). Exacerbating this issue, NNs are prone to overconfident predictions (Guo et al. 2017; Wilson and Izmailov 2020) and opaque due to their 'black-box' nature (Roy et al. 2019). Hence, the uncertainty of a prediction needs to be incorporated and accurately assessed in NN-based decision-making. Predictive uncertainty in the NN output can be induced by the ubiquitous noise in the data analyzed (aleatoric uncertainty), by uncertainty in the model and its parameters (epistemic uncertainty), or a combination of both (Gal 2016).

In recent years, many works on uncertainty in NNs have emerged, in which uncertainty is typically represented in probabilistic terms such as probability distributions (PDs). Most of these works focus on epistemic uncertainty, e.g., by incorporating uncertainty in the weights of the NN in a probabilistic setting (e.g., Gal 2016; Kendall and Gal 2017; Goulet, Nguyen, and Amiri 2021). In such Bayesian NNs, each weight parameter is treated as a PD, thus representing uncertainty in the NN itself. In addition, due to the high presence of noise and data quality defects in real-world data, other works focusing on aleatoric uncertainty and its effect on the NN output have been published, which aim to propagate aleatoric uncertainty (represented by PDs) through NNs (e.g., Abdelaziz et al. 2015; Gast and Roth 2018; Jin, Dundar, and Culurciello 2015). However, regarding exactness, these methods suffer from significant approximation errors and are unable to accurately quantify predictive uncertainty in the NN output. This is mainly due to restrictive and approximative assumptions about the PDs in the input, hidden, or output layer (e.g., the assumption of any post-activation PD being a Gaussian). Therefore, the following research question arises:

How can PDs representing aleatoric uncertainty in input data be propagated through NNs regarding exactness?

To address this research question, we propose Piecewise Linear Transformation (PLT) in this paper, a novel method for propagating aleatoric uncertainty. Our main idea is to harness the locally simple piecewise linear structure of NNs with piecewise linear activation functions, such as ReLU or Leaky ReLU NNs (cf. e.g., Hanin and Rolnick 2019; Sattelberg et al. 2020). Note that our method is still generally applicable, since any activation function can be approximated by piecewise linear functions (Hu et al. 2020; Liao et al. 2023). Our main contributions can be summarized as follows:

- First, we propose PLT, a method for propagating PDs representing aleatoric uncertainty in the input data of a NN with piecewise linear activation functions. PLT makes no restrictive assumptions about the characteristics or type of the input PDs (e.g., an assumption about the input PD being Gaussian) and is thus able to propagate arbitrary PDs.
- Second, PLT supports the accurate quantification of predictive uncertainty based on the criterion exactness of the propagated PD in the output.
- Third, we show the exactness of PLT in theory by proving error bounds for our propagated PDs. We evaluate our method on several real-world datasets induced with aleatoric uncertainty represented by PDs, and validate exactness of our propagation compared to results of competing methods from literature.

Related Work

Literature already provides several works aiming to propagate PDs through NNs. These works can be structured in two groups: function approximation-based approaches and sample-based approaches.

The core idea of the first group is to approximate (possibly arbitrary) PDs in the input layer or the hidden layers of a NN with well-known, parametrical PDs to facilitate their propagation: Astudillo and Neto (2011) and Abdelaziz et al. (2015) assume the PDs in each layer to be Gaussian and focus on their propagation through sigmoid NNs. To this end, they approximate the sigmoid activation function with two piecewise exponential functions and derive closed-form formulas for the mean and variance of a post-activation Gaussian on this basis. Considering Leaky ReLU and ReLU NNs, Gast and Roth (2018) use Gaussians to approximate the post-activation PD of each neuron in each layer. To this end, they propose closed-form analytical formulas to obtain the optimal (with respect to the Kullback-Leibler divergence) means and variances for the approximating Gaussians. Jin, Dundar, and Culurciello (2015) follow a similar approach in the context of Convolutional NNs and propose formulas for the mean and variance of a Gaussian after a max-pooling layer. Titensky, Jananthan, and Kepner (2018) also assume the PDs in the hidden layers to be Gaussian. To estimate their mean and covariance matrix, the Extended Kalman Filter technique (cf. Julier and Uhlmann 1997) is applied. Zhang and Shin (2021) approximate input PDs with Gaussian Mixture Models (GMMs) and demonstrate their propagation through one activation layer. Their main idea is that GMMs with a sufficiently high number of components can approximate arbitrary PDs well. Then, the propagation of the whole PD is split up into simpler propagations of individual Gaussian components based on the Unscented Transform technique (cf. Julier and Uhlmann 2004).

In summary, the parametrical form of the approximating (or assumed) PD often enables a closed-form representation, leading to a straightforward propagation of PDs. However, since not all input and post-activation PD can be approximated well by a parametrical PD, propagation via existing function approximation-based approaches results in substantial errors. This holds even for generic NNs when Gaussians are used to approximate post-activation PDs (for a theoretical analysis of the approximation error cf. Theorem 7 and 8 in Appendix D). Moreover, these substantial approximation errors are also evident in our experimental findings (cf. Section 'Evaluation'), providing empirical evidence. Hence, these approaches are not able to perform an exact or near exact propagation of arbitrary input PDs.

The second group of sample-based approaches aims to propagate PDs by mapping a set of samples through the NN based on which either characteristics of the output PD or the output PD itself should be derived. A very well-known sample-based approach is the Monte Carlo simulation which utilizes a high number of random samples drawn from the input PD (e.g., Abdelaziz et al. 2015; Truong 2021). After propagation through the NN, the output samples are used to aggregate an output PD. However, because drawing a very large number of random samples is associated with high computational cost, lightweight sample-based approaches are necessary and have been proposed in literature: Abdelaziz et al. (2015) suggest using the Unscented Transform (UT) technique to estimate the first two statistical moments of the output PD based on a set of specific, systematically chosen samples. Ji, Ren, and Law (2019) aim to find a lower-dimensional 'active' subspace (AS) of the input space which is designed to describe most of the variation of the NN as an input-to-output function. Then, as an approximation of the NN, a less complex response surface defined on the lower-dimensional subspace is estimated. Finally, Monte Carlo samples are propagated through the response surface to approximate the output PD. Smieja et al. (2018) use GMMs to model aleatoric uncertainty in the context of missing data. An analytical formula is derived for the first statistical moment (i.e., the mean) of the activated GMM after a first ReLU hidden layer. In this sense, the uncertainty is discarded after the first layer and the mean (as a single

sample) is propagated through the NN, thus yielding a deterministic output only. Finally, Jia et al. (2019) propose an approach for uncertainty propagation through non-linear systems which aims to compute the first statistical moments of the output PD based on their integral-based definitions. To solve the integrals efficiently, a sparse grid-based technique which identifies a small representative set of grid point samples is chosen.

These lightweight sample-based approaches share the drawback that the whole output PD has to be estimated based on very limited information. More precisely, this information about the output PD is either given by a small set of samples or a small number of statistical moments. However, because equality of a finite set of statistical moments of two PDs does not imply equality of the PDs themselves, this leads to substantial errors. For instance, a Gaussian and a uniform distribution with the same mean and variance still differ substantially due to their different shape. Indeed, we obtain significant errors in our experimental results for lightweight sample-based approaches for this reason (cf. Section 'Evaluation').

Uncertainty Propagation via Piecewise Linear Transformation

In this section, we present our method – Piecewise Linear Transformation (PLT) – for propagating PDs through NNs with piecewise linear activation functions (e.g., ReLU activation). To this end, we first establish the fact that such NNs can be represented by a piecewise linear function. Thus, we begin by deriving an exact formula for the propagation of an arbitrary PD through a single, affine linear mapping. Crucially for our method, we further extend this formula for NNs with piecewise linear activation functions utilizing their piecewise linear form. Finally, we show how PLT can be used to 1) exactly evaluate the propagated PD in arbitrary output space points and 2) obtain a piecewise constant form of the propagated PD on the whole output space.

We first elaborate on the mathematical structure of NNs with piecewise linear activation functions. Without loss of generality, we exclusively focus on NNs of such structure for the remainder of this paper. Following Sattelberg et al. (2020), we recall definitions of the necessary mathematical concepts of polytopes and piecewise linear functions. The piecewise linear structure of NNs has been recognized in literature and is key in the discussion of expressiveness and complexity of NNs (e.g., Hanin and Rolnick 2019).

Definition 1 (Polytope, polytopic subdivision). *Let $m \geq 0$ be an integer. A bounded convex polyhedron $A \subset \mathbb{R}^m$ is called a polytope. A polytopic subdivision of a bounded set $D \subset \mathbb{R}^m$ is a set of finitely many polytopes $\mathbb{A} = \{A_1, A_2, \dots, A_k\}$ such that $D = \bigcup_i A_i$. A polytopic subdivision is called disjoint if for every $i \neq j$ such that $\dim(A_i) = \dim(A_j)$ the intersection of A_i and A_j is either empty or a polytope of lower dimension $\dim(A_i \cap A_j) < \dim(A_i)$.*

Definition 2 (Piecewise linear function, linear regions). *A function $f: D \to \mathbb{R}^n$ is called piecewise linear with respect to a polytopic subdivision $\mathbb{A} = \{A_1, A_2, \dots, A_k\}$ of D if the restriction $f_i := f|_{A_i}$ is an affine linear function for all $A_i \in \mathbb{A}$. In this case, we call $A_1, A_2, \dots, A_k$ linear regions of f.*

Definition 3 (NNs as piecewise linear functions). *Let $D \subset \mathbb{R}^m$ and $f: D \to \mathbb{R}^n$ be the function representing a NN (with n-dimensional output space). Then the function f has the form*

$$f(x) = \begin{cases} W_1 x + b_1, & if\ M_1 x \leq c_1, \\ W_2 x + b_2, & if\ M_2 x \leq c_2, \\ \dots & \\ W_t x + b_t, & if\ M_t x \leq c_t \end{cases} \quad (1)$$

with $W_i \in \mathbb{R}^{n \times m}, M_i \in \mathbb{R}^{u_i \times m}, b_i \in \mathbb{R}^n, c_i \in \mathbb{R}^{u_i}$ for $t \in \mathbb{N}, u_i \in \mathbb{N}, 1 \leq i \leq t$.

The inequalities $M_i x \leq c_i$ in Eq. 1 divide D into polytopes $A_i = \{x \in D \mid M_i x \leq c_i\}$, which are disjoint (in the sense of Definition 1) and satisfy $D = \bigcup_{i=1}^{t} A_i$. On each polytope A_i, f is given by an affine linear function defined by W_i and b_i. Thus, f is a piecewise linear function with respect to the linear regions A_i, which in particular also define a polytopic subdivision of D. In order to rigorously define probability density functions (PDFs) on such subdivisions in the input, hidden, and output layers of NNs, we now introduce our notion of piecewise PDFs.

Definition 4 (Piecewise PDF). *Let $\mathcal{A} = \mathfrak{B}(D)$ be the Borel σ-Algebra of $D \subset \mathbb{R}^m$. We call a function $P: \mathcal{A} \to \mathbb{R}$ piecewise with respect to a set of subsets $\mathbb{A} = \{A_1, A_2, \dots, A_k | A_i \subset D\}$ if there exists a set of functions $\{p_i: A_i \to \mathbb{R}\}$, such that $P(X) = \sum_{i=1}^{k} \int_{A_i \cap X} p_i \mathrm{d}\mu_{A_i}$ holds for all $X \in \mathcal{A}$. Hereby μ_{A_i} denotes the $\dim(A_i)$-dimensional Lebesgue measure. If P is additionally a PD on D, P is called a piecewise PD (PPD) on D with respect to $\mathbb{A}$, and the set of underlying functions p_i is called the piecewise PDF (PPDF).*

An example for a PPD and its associated PPDF is provided in Appendix A. In other words, a PD on D is piecewise with respect to a set $\mathbb{A} = \{A_1, A_2, \dots, A_k\}$ if it is described by parts of PDFs with each part being restricted to a subset $A_i \in \mathbb{A}$. In particular, any PD defined by a PDF $p: D \to \mathbb{R}$ satisfies this condition if $\dim(A_i) = m$ for all i. Crucially however, our notation of a PPDF allows to describe PDs which are (partially) defined by PDFs on degenerate polytopes. This case of a polytopic subdivision containing degenerate polytopes occurs regularly, especially in the output space of NNs as a consequence of degenerate linear matrices W_i (cf. Eq. 1). Hence, propagated PDs in general cannot be described by traditional PDFs (with respect to the Lebesgue measure of the output space), but a more general notion of PDFs as in Definition 4 is needed.

For the following discussions, we fix the notation of an arbitrary bounded subset $D \subset \mathbb{R}^m$ together with a disjoint polytopic subdivision $\mathbb{A} = \{A_1, A_2, \dots, A_k\}$. If D itself is a polytope, we may assume without loss of generality that D is not degenerate itself (i.e., that $\dim(D) = m$). Moreover, let $f: D \to \mathbb{R}^n$ be a function representing a NN such that f is piecewise linear with respect to $\mathbb{A}$. Further, let P be a PPD on D with a PPDF p with respect to $\mathbb{A}$ and functions p_i on A_i. In this setting, we refer to D as the *input space* and to P as the *input PD*.

Crucial for the exactness of our method, we next address dependencies between neurons in the same layer. Thus, we first model the input layer of a NN with $m \in \mathbb{N}$ real-valued input neurons as a multivariate random variable (RV) $\bar{X}: (\Omega, \mathcal{F}, \mu) \to \mathbb{R}^m$ for a probability space $(\Omega, \mathcal{F}, \mu)$. Following common notation, we denote the pushforward measure on $\mathbb{R}^m$ induced by the measure μ via the RV $\bar{X}$ by $\bar{X}_*\mu$.

As the RVs representing neurons in subsequent layers emanate from the same RV representing the neurons in the input layer, they obviously exhibit substantial dependencies (for a deeper analysis, cf. Appendix D). However, these dependencies are often disregarded in literature and independence between the univariate RVs representing single neurons in the same layer of a NN is typically assumed (Gast and Roth 2018; Wang, Shi, and Yeung 2016; Wu et al. 2018; Jin, Dundar, and Culurciello 2015). Even for the input layer, this assumption of independence severely restricts the set of input PD that can be considered for propagation. In Theorem 8 (Appendix D) we prove that even in the simple case of Gaussians, disregarding dependency substantially impairs the exactness of the propagation. Therefore, instead of considering univariate PDFs of each neuron in a layer individually, we utilize their joint PPDF as a whole. Hence, we preserve dependencies between the neurons in all layers of the NN (including the input layer) because the information of dependency is contained in the joint PPDF. Note that the PPDFs of single neurons can easily be deduced from the joint PPDF by aggregating their respective marginal PPDF.

For the propagation of a PPDF through a NN via PLT, we first determine the linear regions that the input PPDF lies in, and then aggregate the desired output PPDF based on the formulas for PDFs on single linear regions. Therefore, let $W: \mathbb{R}^m \to \mathbb{R}^n$ be a surjective linear map and p_X be a (traditional) PDF describing the input PD P on a linear region. As any linear map W is surjective onto its image, we may replace W by $W: \mathbb{R}^m \to \mathrm{Im}(W)$ without loss of generality. By the Radon-Nikodym Theorem the pushforward measure W_*P admits a PDF with respect to the n-dimensional Lebesgue measure. We denote this PDF by W_*p_X, i.e., $W_*P = W_*p_X \mathrm{d}x$. In other words, the pushforward of the PD P admits the pushforward of the PDF p_X as a PDF again.

Theorem 1 (Propagation of PDFs through linear operators). *Let $W: \mathbb{R}^m \to \mathbb{R}^n$ be a surjective linear map and p_X be a PDF as above. Then $p_Y = W_*p_X$ follows the formula*

$$p_Y(y) = \int_{W^{-1}(y)} p_X(x) \left|\det \widetilde{W}^{-1}\right| \mathrm{d}x$$

almost everywhere. Hereby $\widetilde{W}$ is given by the restriction of W to $\ker W^\perp$. If W is bijective, this formula simplifies to

$$p_Y(y) = p_X(W^{-1}y) |\det W^{-1}| .$$

Proof. Let $Y \subset \mathbb{R}^n$ be a measurable subset and $\pi: \mathbb{R}^m \to \ker W^\perp$ the orthogonal projection. By transformation of variables, we get

$$\int_Y p_Y(y)\mathrm{d}y = \int_{W^{-1}Y} p_X(x)\mathrm{d}x = \int_{\ker W + \pi(W^{-1}Y)} p_X(x)\mathrm{d}x$$
$$= \int_{\pi(W^{-1}Y)} \int_{\ker W} p_X(x+y)\mathrm{d}x\mathrm{d}y$$
$$= \int_Y \int_{\ker W} p_X\left(x + \widetilde{W}^{-1}y\right) \left|\det \widetilde{W}^{-1}\right| \mathrm{d}x\mathrm{d}y .$$

Since a measurable preimage under W exists for any measurable subset $Y \subset \mathbb{R}^n$, this yields the desired identity almost everywhere. □

Theorem 1 presents an exact formula for the propagation of PDFs through affine linear operators representing the NN on its linear regions. By utilizing joint PDFs in this formula, we preserve the dependencies between the neurons and thus retain crucial information for exact propagation. Next, we extend this result to piecewise linear functions and PPDFs.

Theorem 2 (Propagation of PPDFs through piecewise linear functions). *For all $1 \le i \le k$ let f_i denote the affine linear map such that $f|_{A_i} = f_i$. Then the (propagated) PPD f_*P is represented by the PPDF f_*p defined by*

$$f_*p(y) = \sum_{\substack{i=1 \\ y \in f_i(A_i)}}^{k} \frac{1}{\det \widetilde{f_i}} \int_{f_i^{-1}(y) \cap A_i} p_i(x)\mathrm{d}x. \qquad (2)$$

In this formula, $\widetilde{f_i}$ denotes the restriction of f_i to $\ker(f_i)^\perp$. Moreover, the integrals are defined with respect to the $\dim(A_i)$-dimensional Lebesgue measure.

Proof. First consider a single polytope $A_i \in \mathbb{A}$ together with $p_i: A_i \to \mathbb{R}$ and the function $f_i: \mathbb{R}^m \to \mathbb{R}^n$. The function p_i defines a measure M on A_i and we aim to find f_*p_i on $\mathbb{R}^n$ which defines the pushforward measure f_*M. This problem can be traced back to the formula for linear operators by extending p_i to a function $\overline{p_i}$ on $\mathbb{R}^m$ by

$$\overline{p_i}(x) := \begin{cases} p_i(x), & if\ x \in A_i, \\ 0, & else. \end{cases}$$

The pushforward measure associated to $\overline{p_i}$ is also given by f_*M, and from Theorem 1 (with $(f_i)_*p_i = p_Y$) we obtain

$$(f_i)_*p_i(y) = \frac{1}{\det(\widetilde{f_i})} \int_{f_i^{-1}(y)} \overline{p_i}(x)\mathrm{d}x$$
$$= \frac{1}{\det(\widetilde{f_i})} \int_{f_i^{-1}(y) \cap A_i} p_i(x)\mathrm{d}x.$$

Note that $(f_i)_*p_i(y)$ only attains nonzero values if $y \in f(A_i)$. Hence, it is fully described by its restriction to $f(A_i) = f_i(A_i)$, which also is a polytope. As the intersection of images of different polytopes under f can be non-empty, the formula for the pushforward of a PPD along f has to be

obtained by taking the sum over all polytopes in the subdivision $\mathbb{A}$. Moreover, the PD f_*P on $f(D)$ is also piecewise and can be described by a PPDF with respect to $f(\mathbb{A}) \coloneqq \{f(A_1), \dots, f(A_k)\}$ and the functions $(f_i)_* p_i \colon f(A_i) \to \mathbb{R}$. In summary, the statement given by Eq. 2 follows. □

The formula in Eq. 2 for input PD propagation is now applicable to NNs, which poses a vital step of our PLT method proposed in this paper. However, it involves computing integrals of the form as in Eq. 2, which is not possible in closed form for arbitrary PPDFs, e.g., PPDFs without a closed analytical formula. In order to cope with this problem, we next introduce a grid-based approximation technique. We simplify the above formula in Eq. 2 for such approximate PPDFs, enabling a tractable propagation. In particular, by approximating the PPDF on a fine-grained grid, the shape of the input PD can be preserved all the way through the propagation process.

In a first step, we approximate p with another piecewise constant PPDF p'. We construct this approximate PPDF to be piecewise with respect to a very fine-grained grid of polytopes to minimize the approximation error. A discussion of this theoretical error is part of the following section and is deepened further in Appendix C. More precisely, the initial polytopic subdivision $\mathbb{A} = \{A_1, A_2, \dots, A_k\}$ of D is refined as follows: First, each polytope A_i is subdivided into $\dim(A_i)$-dimensional simplices by applying Delaunay triangulation (Delaunay 1934). As a result, we obtain a (more fine-grained) polytopic subdivision of D solely consisting of simplices. Moreover, any subdivision of this form can be subdivided further using the edgewise subdivision technique (cf. Edelsbrunner and Grayson 2000) satisfying the property that for any $b \in \mathbb{N}$, each simplex of dimension d can be subdivided into b^d sub-simplices of dimension d and equal volume. In this way, an arbitrarily, evenly fine-grained grid can be obtained (Edelsbrunner and Grayson 2000, also cf. Appendix B for more details).

To control the granularity of the grid, we fix a small threshold $\varepsilon > 0$. We require that for each (simplicial) polytope A in the final subdivision, every edge of A is shorter than ε (i.e., $dist(c_1, c_2) < \varepsilon, \forall c_1, c_2 \in C(A)$ with $C(A)$ denoting the set of all vertices of A). This can either be achieved by choosing the subdivision parameter b large enough or by iteratively applying the edgewise subdivision. The result is a fine-grained polytopic subdivision

$$\mathbb{A}^{\varepsilon}_{edge} = \{A^{\varepsilon}_{1,1}, A^{\varepsilon}_{1,2}, \dots, A^{\varepsilon}_{1,l_{1,\varepsilon}}, A^{\varepsilon}_{2,1}, \dots, A^{\varepsilon}_{k,l_{k,\varepsilon}}\}, \tag{3}$$

where $l_{i,\varepsilon}$ denotes the number of edgewise simplices of the polytope A_i, and the length of any edge of any simplex $A^{\varepsilon}_{i,j} \subset A_i$ is smaller than ε. We then define the approximative PPDF p' by

$$p' \colon D \to \mathbb{R}, p'(x) = c_{ij} \quad \text{with } c_{ij} \coloneqq \frac{1}{\dim(A_i) + 1} \sum_{c \in C\left(A^{\varepsilon}_{i,j}\right)} p_i(c), \tag{4}$$

if $x \in A^{\varepsilon}_{i,j} \in \mathbb{A}^{\varepsilon}_{edge}$.

By this definition, p' is constant on each edgewise simplex $A^{\varepsilon}_{i,j}$, attaining the average value of p evaluated in the vertices of $A^{\varepsilon}_{i,j}$. Applying Theorem 2 to p' now yields the following result:

Theorem 3. *If the PPDF p is approximated using the piecewise constant PPDF p' given by Eq. 4, the formula from Theorem 2 simplifies to:*

$$f_* p'(y) = \sum_{i=1}^{k} \sum_{\substack{j=1, \\ y \in f_i(A_{i,j})}}^{l_i} \frac{c_{ij}}{\det \overline{f_i}} \operatorname{vol}\left(f_i^{-1}(y) \cap A_{i,j}\right). \tag{5}$$

Proof. Follows directly from Theorem 2 and the definition of piecewise constant PPDFs in Eq. 4. □

With Eq. 5, we have derived a formula based on PLT to evaluate the propagated PPDF at any point $y \in \mathbb{R}^n$ in the output space. Hence, we can obtain the propagated PPD as a whole by iteratively applying Eq. 5 on the points of a grid over the output space (e.g., the grid induced by the edgewise subdivision on the output space as described above).

Moreover – instead of evaluating a set of grid points in the output space – PLT can also be used to infer the whole propagated PPDF from an input grid (for which again the edgewise subdivision is a natural choice). More precisely, the core idea is to map the input grid points together with their closely adjacent probability mass into the output space via the simple linear operator of the respective linear region. Finally, the probability mass is assigned to a cuboid bin in the output space. By defining a fine-grained grid of such cuboid bins on the output space, we also obtain the output PPDF in a piecewise constant form, similar to a (multidimensional) histogram. In Theorem 5 (cf. Appendix C) we show that this piecewise constant form also can be rendered arbitrarily exact for bins with increasingly small volume.

Formally, let $\mathbb{A}_{edge}$ be the edgewise subdivision of the input space as in Eq. 3. Then, the function f representing the NN is also affine linear on each $A_{i,j} \in \mathbb{A}_{edge}$. Moreover, let S be a bin grid of disjoint bins of equal size on the output space. For a n-dimensional output space we can choose cuboid bins with side lengths $l_1, \dots, l_n$ for each dimension, ensuring equal size of bins and easier computation. Since on each polytope $A_{i,j}$ with center t_{ij} the input PPDF is given by a constant value $c_{ij} \in \mathbb{R}$ (cf. Eq. 4), the probability mass of $A_{i,j}$ (equal to $c_{ij} \cdot \operatorname{vol}(A_{ij})$) can be propagated and assigned to the bin $B \in S$ containing $f(t_{ij})$. Thus, the estimated constant PPDF on a bin B is given by

$$f_* p(y) = \frac{1}{l_1 \cdot \dots \cdot l_n} \sum_{i=1}^{k} \sum_{\substack{j \\ f(t_{ij}) \in B}} \operatorname{vol}\left(A_{i,j}\right) \cdot c_{ij} \tag{6}$$

for all $y \in B$. While the formula in Eq. 5 is particularly advantageous when evaluating the output PPDF in a certain set of points, propagation via Eq. 6 yields the output PPDF as a whole, since all probability mass in the input is propagated and assigned to the respective bin in the output space. In Theorem 5 (cf. Appendix C) we prove that for increasingly fine subdivision $\mathbb{A}_{edge}$ and bin grid S this PPDF described in Eq. 6 becomes arbitrarily exact.

Evaluation

We first provide a theoretical evaluation by establishing the mathematical exactness of PLT. We discuss rigorous bounds for possible approximation errors made by PLT for which detailed proofs are provided in Appendix C. In Lemma 3 (cf. Appendix C) we show that for piecewise constant approximations of arbitrary input PDs as defined in Eq. 4, the approximation error converges to zero for increasingly small diameters (i.e., the maximum distance of two vertices) of edgewise simplices $A_{i,j}^{\varepsilon}$. As ε can be chosen arbitrarily small and the diameter of any $A_{i,j}^{\varepsilon}$ is smaller than ε by construction of the edgewise subdivision, the approximation error also becomes arbitrarily small. As a direct consequence of the simple piecewise linear form of NNs, we further show in Theorem 4 (cf. Appendix C) that the error after propagation through the NN in the output space is always bounded by the approximation error in the input space. From the combination of these results the theoretical exactness of our method and our formula of the propagated PD as in Eq. 6 follows: Since we obtain an approximative piecewise constant output PPDF with respect to a cuboid bin grid, the approximation error in the output again becomes arbitrarily small for an increasingly fine-grained bin grid. Hence, the PPDF in Eq. 6 indeed converges towards the true output PD (for details cf. Theorem 5, Appendix C).

In the following, we aim to substantiate our theoretical results with empirical evidence. To this end, we evaluate our method on a broad range of publicly available real-world datasets from various domains for both classification and regression tasks. Details about the datasets are provided in Table 2 (cf. Appendix E). We randomly split each dataset into training and test dataset and train a standard ReLU NN for classification or regression depending on the task associated to the dataset.

Since the datasets do not exhibit aleatoric uncertainty represented by PDs, we induce uncertainty according to the following procedure: In a first step, we analyze each dataset regarding feature importance. On this basis, we select a set of features with high feature importance. Each feature value of these selected features is labeled as uncertain with a fixed probability (e.g., 50 percent, cf. '%unc' in Table 1). By focusing on features with high importance for the output result, we make sure that the induced aleatoric uncertainty indeed affects the NN prediction and non-trivial predictive uncertainty can be observed. For each data instance containing uncertain feature values, the multiple imputation method MICE (van Buuren and Groothuis-Oudshoorn 2011) is deployed (as if the values were missing), leading to a set of multiple values suggested for imputation. Finally, for each instance we apply a Gaussian kernel density estimation on this set of imputation values, thus obtaining a (potentially multivariate) continuous PD[1] over its uncertain features.

To evaluate our method with respect to the criterion exactness, the deviation of the propagated PD obtained by our method from an exact ground truth has to be quantified. We generate this ground truth output PD by utilizing a Monte Carlo simulation with a very high sample count. To ensure a high quality of this ground truth, we start with a fixed number of samples that are propagated through the NN and compare the resulting output PD to a more refined PD obtained analogously but with twice the number of samples. This process is iterated until this increasingly exact sequence of PDs converges, i.e., until the L1-distance between two consecutive PDs falls under a fixed, small threshold. The generation of a high-quality ground truth via Monte Carlo simulation comes with enormous computational effort, which we undertook once for each data instance in the datasets analyzed.

The PDs propagated by PLT and the other existing and applicable methods from literature[2] (cf. Related Work) are then evaluated against this ground truth. More precisely, we calculate L1-, L2-, and Hellinger distance between the ground truth and the PD obtained by each method as these are well-known and meaningful standard metrics for PDFs (for definitions and more details, cf. Appendix C). Evaluating deterministic performance metrics such as accuracy or mean squared error is not desirable in this context because of two reasons: First, the output PDs would be condensed into single, deterministic values, resulting in a significant loss of information. Second, the uncertainty-based ground truth often induces 'true' labels different to the ones given by the certain dataset. By quantifying PDF-based distances between the PD and the ground truth, more general metrics accounting for both of these points are considered.

The experimental results presented in Table 1 substantiate that PLT significantly outperforms existing methods regarding exactness across all metrics, thus confirming our theoretical results and validating the ability our of method for accurate uncertainty propagation with respect to exactness.

[1] The uncertain datasets resulting from this procedure are provided in the supplementary material.

[2] Code for the approach of Zhang and Shin (2021) was requested, but not provided by the authors. Therefore, this method could not be considered in the evaluation.

Datasets	% unc	PLT (ours)			ADF (Gast and Roth)			AS (Ji, Ren, and Law)			UT (Abdelaziz et al.)		
		L1	L2	H	L1	L2	H	L1	L2	H	L1	L2	H
Appendicitis	25	**0.029**	**1.048**	**0.040**	1.623	27.490	0.813	0.333	8.129	0.198	0.889	25.615	0.539
	50	**0.045**	**3.381**	**0.052**	1.293	68.298	0.669	0.969	20.239	0.497	0.983	67.555	0.556
Banana	25	**0.114**	**0.694**	**0.104**	1.332	1.995	0.759	1.198	2.167	0.609	1.018	1.985	0.656
	50	**0.079**	**0.316**	**0.072**	1.346	1.434	0.761	1.313	1.780	0.657	1.037	1.433	0.669
Balance	25	**0.038**	**1.375**	**0.047**	0.793	40.805	0.405	0.502	19.539	0.264	0.654	38.936	0.321
	50	**0.044**	**1.222**	**0.042**	0.776	27.221	0.383	0.560	18.718	0.298	0.625	27.439	0.289
Bands	25	**0.130**	**22.331**	**0.078**	1.738	134.118	0.875	0.422	25.562	0.249	14.978	426.529	1.067
	50	**0.220**	**15.676**	**0.121**	1.695	76.088	0.861	0.452	18.525	0.270	8.602	221.450	1.238
Boston	25	**0.131**	**0.661**	**0.097**	0.859	1.734	0.429	0.368	1.078	0.193	0.565	1.330	0.301
	50	**0.083**	**0.156**	**0.065**	0.782	1.170	0.380	0.300	0.578	0.162	0.516	0.750	0.259
Breast	25	**0.151**	**5.301**	**0.099**	1.628	64.911	0.827	0.281	18.933	0.161	9.204	151.424	1.052
	50	**0.230**	**5.857**	**0.131**	1.632	50.158	0.826	0.281	9.110	0.173	22.927	486.339	1.431
California	25	**0.147**	**11.925**	**0.074**	0.987	97.197	0.617	0.335	29.937	0.193	0.898	73.451	0.458
	50	**0.304**	**19.492**	**0.140**	0.836	64.378	0.435	0.415	30.829	0.242	0.750	54.400	0.387
Diabetes	25	**0.014**	**0.078**	**0.009**	0.534	3.057	0.266	0.259	1.588	0.136	0.420	2.235	0.219
	50	**0.012**	**0.059**	**0.008**	0.522	2.556	0.250	0.235	1.284	0.121	0.415	1.943	0.210
Iris	25	**0.023**	**0.330**	**0.037**	1.854	11.643	0.920	0.564	6.082	0.333	0.999	11.710	0.695
	50	**0.029**	**0.153**	**0.035**	1.856	4.204	0.907	0.380	1.368	0.259	0.999	4.257	0.701
Machine	25	**0.031**	**5.032**	**0.027**	0.873	121.193	0.422	0.180	20.733	0.099	0.736	95.224	0.345
	50	**0.034**	**4.652**	**0.032**	0.820	88.004	0.392	0.186	16.689	0.103	0.661	67.060	0.305
Real estate	25	**0.021**	**2.381**	**0.011**	0.596	77.677	0.294	0.133	12.939	0.076	0.399	46.363	0.191
	50	**0.015**	**1.752**	**0.009**	0.613	70.441	0.296	0.148	13.043	0.083	0.430	44.167	0.207
Vehicle	25	**0.035**	**26.123**	**0.029**	0.887	460.033	0.471	0.103	60.106	0.059	0.681	275.497	0.340
	50	**0.045**	**25.491**	**0.038**	0.888	396.444	0.467	0.100	52.507	0.049	0.718	268.618	0.358
Wine	25	**0.042**	**7.587**	**0.043**	1.742	159.540	0.882	0.592	61.310	0.326	1.036	160.301	0.704
	50	**0.043**	**41.778**	**0.046**	1.736	902.390	0.870	0.801	115.544	0.422	1.696	910.188	0.778

Table 1: Experimental Results (cf. Table 2 in Appendix E for details on the datasets)

Discussion and Conclusion

In this paper, we proposed PLT, a novel method for the propagation of aleatoric uncertainty through NNs that is applicable to arbitrary PDs in the input space. To this end, we introduced the notion of PPDFs, which allows to generalize the concept of PDFs to the polytopic subdivisions occurring in NNs. By propagating a joint PD across all uncertain features, our method is able to preserve vital dependencies between neurons in each NN layer. We provided mathematical proofs that neuron dependencies must be considered as the simplifying assumption of independence leads to large approximation errors (cf. ADF in Table 1). Further, we showed that our method is able to approximate the true propagated PD up to an arbitrarily small error, allowing us to accurately quantify predictive uncertainty. We evaluated our method on a broad range of real-world datasets for both classification and regression tasks, where it achieved higher exactness than competing methods from literature. In particular, our evaluation shows that methods making restrictive assumptions (ADF and UT), such as independence of neurons in NN layers or Gaussian form of PDs, exhibit high errors regarding exactness, thus confirming our theoretical findings. Similarly, a competing sample-based method (AS) suffered from the drawback that complex PDs have to be estimated from limited information and yielded a worse exactness than PLT.

Moving forward, our method has broad application potential across various domains, particularly in areas where uncertainty and its quantification play a pivotal role, such as medicine, finance, and high-risk environments (e.g., self-driving cars). One limitation to acknowledge is that PLT requires an already trained NN to propagate aleatoric uncertainty. Hence, the quality of our propagated PD is subject to the quality of the given NN model. Moreover, despite posing a potential starting point, uncertainty during the training phase of a NN is not yet addressed by PLT. Additionally, our results only hold for NNs with piecewise linear activation functions, thus PLT cannot be directly applied to NNs with other activation functions. However, as mentioned earlier, any activation function can be approximated by a piecewise linear function in order to apply PLT. Another intriguing direction to explore is the integration of PLT into different NN architectures (e.g., CNNs). These challenges represent interesting avenues for future research.

Appendix

Appendix A: Propagation of Densities

In this section, some additional information about PPDs and their propagation through linear operators is provided. We show a simple example of a PD on a union of polytopes of different dimensions which does not admit a PDF with respect to the Lebesgue measure in the traditional sense but does admit a PPDF in our sense with respect to Lebesgue measures of different dimensions. Additionally, we give a Corollary to Theorem 1 which shows that the dimension of the output PD is bounded by that of the input PD.

Example 1 (Piecewise probability distribution). *First, we provide a simple 2-dimensional example of a PD given by a PPDF on a polytopic subdivision consisting of 1-dimensional and 2-dimensional polytopes. Consider the 1-dimensional polytopes* A_0 *and* A_1 *defined by the closed intervals* $[0, 0.5] \times [2]$ *and* $[0.5, 1] \times [2]$*, respectively, and the 2-dimensional polytope* A_2 *given by the rectangle* $[0, 0.5375] \times [0,1]$*. Let* $D = A_0 \cup A_1 \cup A_2$*. The functions* $p_0 \colon A_0 \to \mathbb{R}, (x_1, x_2) \to 3x_1$*,* $p_1 \colon A_1 \to \mathbb{R}, (x_1, x_2) \to 0.3x_1^2$ *and* $p_2 \colon A_2 \to \mathbb{R}, (x_1, x_2) \to 1$ *define a piecewise function* $P \colon \mathcal{B}(D) \to \mathbb{R}$ *as in Definition 4. Additionally, the function* P *is* σ*-additive and satisfies* $P(D) = P(A_0) + P(A_1) + P(A_2) = 1$*. Note the first two summands are given by integrals with respect to the 1-dimensional Lebesgue measure whereas the third summand is given by an integral with respect to the 2-dimensional Lebesgue measure. It follows that* P *defines a PPD and the set* (p_0, p_1, p_2) *is the associated PPDF.*

Corollary 1 (Dimensionality is bounded). *We define the dimensionality of a PPDF to be the dimension of its support, i.e., the dimension of the highest-dimensional polytope on which the PPDF is defined. In the situation of Theorem 1, let* $d \in \mathbb{N}, d \leq m$*, be the dimensionality of* p_X*. Then the dimensionality* d' *of* $p_Y = W_* p_X$ *is bounded by* d*, i.e.,* $d' \leq d$*.*
Proof. The support $supp(p_X) \subset \mathbb{R}^m$ is a d-dimensional manifold (with $d \leq m$). If W is injective, the support $supp(p_Y) \subset \mathbb{R}^n$ again is a d-dimensional manifold. If W is non-injective (i.e. $\ker(W)$ is a k-dimensional subspace of $\mathbb{R}^m, k \geq 1$), the dimensionality of $supp(p_Y)$ is $d - i$ with $0 \leq i \leq k$, depending on the number of basis vectors of the kernel that generate $supp(p_X)$. □

Appendix B: Edgewise Subdivision

In PLT, we use edgewise subdivision to obtain a fine-grained grid of simplices in the input space on which the input PD is approximated by a piecewise constant PPDF. We now elaborate on edgewise subdivision by first explaining the construction of the subdivision and then discussing some useful properties.

Definition 5 (Corners and dimensions of a polytope). *Let* $m \geq 0$ *be an integer and* $A \subset \mathbb{R}^m$ *a polytope. There is a unique minimal set of points* $C(A)$ *in* $\mathbb{R}^m$ *such that* A *is the convex hull of these points. We call an element of* $C(A)$ *a corner of* A*. The dimension* $dim(A)$ *is given by the dimension of the affine hyperplane spanned by* $C(A)$*.*
We call

$$h(A) = \max_{a_i, a_j \in C(A)} |a_i - a_j|$$

the diameter of A *and set*

$$\rho(A) = 2 \sup\{R > 0 \colon B_R(x_0) \subset A, x_0 \in A\}$$

as the diameter of the biggest inner ball contained in A*.*

We now follow Edelsbrunner & Grayson (2000) to construct the edgewise subdivision of a simplex. Let A be a simplex of dimension d. For a natural number $k \in \mathbb{N}$, each 1-dimensional facet between exactly two corners of the simplex is divided into $k \in \mathbb{N}$ pieces of equal length creating a grid of points. The edgewise subdivision of A comprises the k^d (sub-)simplices spanned by adjacent (in direction of the vectors spanning the simplex) points of the grid. The edgewise subdivision for $k = 2$ on a 2-simplex and a 3-simplex is illustrated in Figure 1.

The subdivision of A obtained by this method is denoted by $\mathbb{A}^k_{edge}(A)$. It follows immediately from the definition of edgewise subdivision that for sufficiently large $k \in \mathbb{N}$ the distance between any two corners of any simplex in $\mathbb{A}^k_{edge}(A)$ becomes arbitrarily small. Thus, for every $\varepsilon > 0$ there exists a $k_\varepsilon \in \mathbb{N}$ such that $d(c_1, c_2) < \varepsilon$ for all $c_1, c_2 \in C(A_i)$ and every simplex $A_i \in \mathbb{A}^{k_\varepsilon}_{edge}(A)$. The minimal choice of $k_\varepsilon \in \mathbb{N}$ depends on A.

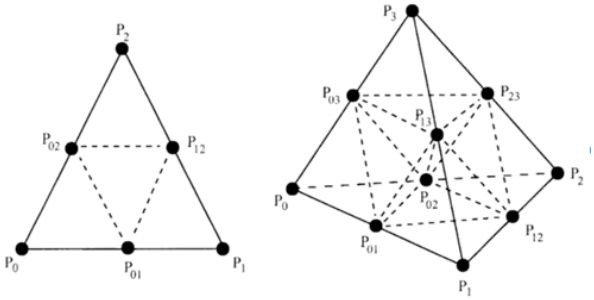

Figure 1: Edgewise Subdivision of a 2-simplex (left) and a 3-simplex (right) for $k = 2$ (Edelsbrunner & Grayson, 2000)

Lemma 1. *Let $k, l \in \mathbb{N}$. Then edgewise subdivision with respect to kl is equivalent to first subdividing with respect to l and then further subdividing each simplex in $\mathbb{A}^l_{edge}(A)$ with respect to k, i.e.,*

$$\mathbb{A}^{kl}_{edge}(A) = \mathbb{A}^k_{edge}(\,\mathbb{A}^l_{edge}(A)).$$

Proof. Edelsbrunner & Grayson (2000). □

The simplices in $\mathbb{A}^k_{edge}(A)$ can be assigned to congruence classes, i.e., two simplices belong to the same congruence class if and only if they are congruent. We denote the set of congruence classes of $\mathbb{A}^k_{edge}(A)$ by $Cong\left(\mathbb{A}^k_{edge}(A)\right)$. It can be shown that there exists an upper bound for the number of congruence classes which does not depend on k:

Lemma 2. *For any simplex A of dimension d, the set $\mathbb{A}^k_{edge}(A)$ has at most $d!/2$ congruence classes.*
Proof. Edelsbrunner & Grayson (2000). □

Appendix C: Error Estimation

In this subsection we study the error made when approximating the PPDF p with a piecewise polynomial density p' of degree k. In particular, choosing constant polynomials yields an error estimation for the method described in the main chapter. We measure the error by means of p-norms and the Hellinger distance, which we will define in the following. We will then provide upper bounds for the approximation error both in the input (Lemma 3) and output space (Theorem 4) and show that if the input subdivision and the output bin grid are fine enough, our method PLT is able to approximate the true propagated PD arbitrarily closely (Theorem 5).

Definition 6 (p-Norm). *Let $(\Omega, \mathcal{A}, \mu)$ be a measure space and $f: \Omega \to \mathbb{R}$ a measurable function. The p-norm of f is defined as*

$$\|f\|_p = \|f\|_{L^p(\Omega)} = \left(\int_\Omega |f|^p \, \mathrm{d}\mu\right)^{\frac{1}{p}}$$

for $p \in [1, \infty)$ and

$$\|f\|_\infty = \|f\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}_{x \in \Omega} |f(x)|$$

for $p = \infty$. We say $f \in L^p(\Omega)$ if $\|f\|_p < \infty$.

In our case, we always consider a measure space $\Omega \subset \mathbb{R}^m$ together with the Borel σ-Algebra $\mathcal{A} = \mathcal{B}(\Omega)$ and the Lebesgue measure μ .

Definition 7 (σ-finite measure). *Let $(\Omega, \mathcal{A}, \mu)$ be a measure space. The measure μ is called σ-finite if Ω can be covered by countably many measurable sets $B_1, B_2, \ldots \in \mathcal{A}$ such that $\mu(B_i) < \infty$ for all i.*

Definition 8 (Absolutely continuous measure). *Let μ_1, μ_2 be two measures on a measureable space X. The measure μ_1 is absolutely continuous with respect to μ_2 if every measurable set $B \subset X$ with $\mu_2(B) = 0$ also satisfies $\mu_1(B) = 0$.*

Definition 9 (Hellinger distance). *Let P_1, P_2 be PDs on a measurable space X. Choose a σ-finite measure μ with respect to which P_1 and P_2 are absolutely continuous. Note that such a measure always exists (for example, $\mu = P_1 + P_2$ is a possible choice). Then by the Radon-Nikodym Theorem P_1 and P_2 have PDFs p_1 and p_2, respectively, on X with respect to μ. The Hellinger distance between P_1 and P_2 is defined as*

$$H(P_1, P_2) = \sqrt{\frac{1}{2}\int_X \left(\sqrt{p_1} - \sqrt{p_2}\right)^2 \mathrm{d}\mu}.$$

By Pollard (2001), $H(P_1, P_2)$ does not depend on the choice of μ. Hence, the Hellinger distance is well-defined. The Hellinger distance defines a metric on the set of measures on the measurable space X. The maximum value of the Hellinger distance between P_1 and P_2 is 1, which is attained if and only if P_1 and P_2 are mutually singular (Pollard 2001), i.e., there exists a measurable subset S of X such that $P_1(S) = 0 = P_2(X \backslash S)$.

For a symmetric semi-positive definite (s.s.p.d.) $n \times n$-matrix Σ, we set $supp(\Sigma) := \mathrm{Im}(\Sigma)$. This is the unique linear subspace V of $\mathbb{R}^n$ such that the normal distribution $N(0, \Sigma)$ admits a PDF with respect to the Lebesgue measure on V.
To denote the Hellinger distance between normal distributions, we will use the abbreviation $H(\Sigma_1, \Sigma_2) := H(N(0, \Sigma_1), N(0, \Sigma_2))$.

We will now provide error estimations for our approximation of the true PPDF. While we only use a constant approximation of the PPDF on each polytope in $\mathbb{A}_{edge}(A_i)$, we give an upper bound for the approximation error in a more general case where for each polytope, the PPDF values on $C(A)$ are interpolated by a polynomial of degree k such that for any polynomial PPDF of degree $\leq k$ the interpolation polynomial agrees with the PPDF. Note that our constant approximation fulfills this condition for $k = 0$. We start by estimating the approximation error of the polynomial approximation by means of the 2-norm. We consider the case where the PPDF p on a polytope A_i is $(k+1)$-times differentiable in a weak sense and its weak derivatives have finite 2-norm. In particular, every function p for which the first $k+1$ (classic) derivatives exist and have finite 2-norm, is an element of $W^{k+1,2}(A_i)$.

Lemma 3 (Approximation error, 2-norm). *Let $p \in W^{k+1,2}(A_i)$ (i.e., p is $(k+1)$-times differentiable in a weak sense and the weak derivatives have finite 2-norm). Let p' be the polynomial approximation of degree k on an edgewise subdivision $\mathbb{A}^{\varepsilon}_{i,edge}(A_i)$ of A_i. Then there exist $C_0, C_1 > 0$ such that*

$$\|p - p'\|_{L^2(A_i)} \leq C_0 h^{k+1} \|D^{k+1} p\|_{L^2(A_i)},$$
$$\|\nabla p - \nabla p'\|_{L^2(A_i)} \leq C_1 \sigma h^k \|D^{k+1} p\|_{L^2(A_i)},$$

where

$$h = \max_{A \in \mathbb{A}^{\varepsilon}_{i,edge}} h(A),$$
$$\sigma = \max_{A \in \mathbb{A}^{\varepsilon}_{i,edge}} \frac{h(A)}{\rho(A)} \text{ and}$$
$$\|D^{k+1} p\|_{L^2(A_i)} := \left(\int_{A_i} \sum_{|\alpha|=k+1} |D^{\alpha} p|^2 \mathrm{d}x\right)^{\frac{1}{2}}$$

where $\alpha \in \mathbb{N}_0^m$ is the associated multi-index vector with $\sum_i \alpha_i = k+1$. The constants C_0 and C_1 depend on both k and A_i. Moreover, we have

$$\lim_{\varepsilon \to 0} \|p - p'\|_{L^2(A_i)} = 0,$$
$$\lim_{\varepsilon \to 0} \|\nabla p - \nabla p'\|_{L^2(A_i)} = 0$$

Note that p' is dependent on $\varepsilon > 0$ as it is defined based on the subdivision $\mathbb{A}^{\varepsilon}_{i,edge}$.

Proof. The first two inequalities are derived in Ciarlet (2002). To show the claims about the limit of the errors for $\varepsilon \to 0$, it suffices to show that σ and $\|D^{k+1} p\|_{L^2(A_i)}$ are bounded and that $h \leq \varepsilon$. For two congruent simplices $A_i, A_j \in c$ (where c denotes the associated congruence class) it is clear that

$$h(A_i)/\rho(A_i) = h(A_j)/\rho(A_j) =: \sigma_c$$

holds and σ_c is finite. By Lemma 2 $\mathbb{A}^{\varepsilon}_{i,edge}$ comprises at most $d!/2$ congruence classes. Hence, $\sigma = \max_c \sigma_c < \infty$ can be chosen as the maximum of the finite set $\{\sigma_c \mid c \in Cong(\mathbb{A}^{\varepsilon}_{i,edge})\}$ and is independent of ε. Moreover, $\|D^{k+1} p\|_{L^2(A_i)} < \infty$ holds by definition as $p \in W^{k+1,2}(A_i)$. It follows immediately from the construction of edgewise subdivision that for any $l \in \mathbb{N}$ we have

$h(A_{i,j}) \leq 1/l \cdot h(A_i)$ for all simplices $A_{i,j} \in \mathbb{A}^l_{i,edge}(A_i)$. By definition of $\mathbb{A}^\varepsilon_{i,edge}$, $l \in \mathbb{N}$ is chosen large enough such that $h(A_{i,j}) \leq \varepsilon$ holds for all $A_{i,j} \in \mathbb{A}^l_{i,edge}(A_i)$ which yields the claim. □

The constants $\|D^{k+1}p\|_{L^2(A_i)}$ and σ can be interpreted as follows: If the $(k+1)$-th derivatives of p attain large values or variations on major parts of A_i, it is difficult to approximate p in these regions by polynomial interpolation. In this case, the value of $\|D^{k+1}p\|_{L^2(A_i)}$ (and hence, the right-hand side of the error estimation) is rather large. The value $\sigma = \max_{A \in \mathbb{A}^\varepsilon_{i,edge}} \frac{h(A)}{\rho(A)}$ is a measure of how geometrically "flat" the simplices in $\mathbb{A}^\varepsilon_{i,edge}$ are. For a flat simplex the longest 1-dimensional facet of the simplex is relatively large when compared to the diameter of its largest inner circle. Thus, a polynomial interpolation of a function defined on this simplex is prone to error due to the relatively large distances of the points used for interpolation. A large value of σ accounts for this condition.

The approximation error on $\mathbb{A}^\varepsilon_{i,edge}(A_i)$ with respect to the 2-norm is of order $O(\varepsilon^{k+1})$ for $\|p - p'\|_{L^2(A_i)}$ and $O(\varepsilon^k)$ for $\|\nabla p - \nabla p'\|_{L^2(A_i)}$, respectively. Thus, for a higher-degree interpolation, the approximation error in the input space decreases faster. However, for any degree $k > 0$, the propagation of p' through a NN can no longer be done in an exact manner as it involves calculating integrals of p' over polytopes. Therefore, we only use constant approximations of the true PDF for our method PLT.

Theorem 4 (Error propagation, p-norm). *Let p_1, p_2 be PDFs on a polytope A, let $p \in [1, \infty)$ and $f: A \to \mathbb{R}^m$ a linear function on A. Then there exists $C > 0$ such that*

$$\|f_*(p_1) - f_*(p_2)\|^p_{L_p} \leq C \cdot \|p_1 - p_2\|^p_{L^p}$$

Proof. Let $\tilde{f}$ be the restriction of f to $\ker f^\perp$ and $c := |\det(\tilde{f}^{-1})|$. Using Hölder inequality and triangle inequality for integrals, we get

$$\begin{aligned}
&\|f_*p_1 - f_*p_2\|^p_{L_p} \\
&= \int_{f(A)} \left| \int_{f^{-1}(x) \cap A} c \cdot p_1(y) \mathrm{d}y - \int_{f^{-1}(x) \cap A} c \cdot p_2(y) \mathrm{d}y \right|^p \mathrm{d}x \\
&= \int_{f(A)} \left| \int_{f^{-1}(x) \cap A} c \cdot \big(p_1(y) - p_2(y)\big) \mathrm{d}y \right|^p \mathrm{d}x \leq C \int_{f(A)} \int_{f^{-1}(x) \cap A} |p_1(y) - p_2(y)|^p \, \mathrm{d}y \, \mathrm{d}x \\
&= C \int_A |p_1(z) - p_2(z)|^p \, \mathrm{d}z = C \|p_1 - p_2\|^p_{L^p}
\end{aligned}$$

for some constant C which depends on A and $\tilde{f}$. □

Finally, we prove that our propagated piecewise constant PPDF given by formula (6) can approximate the true output PD P_y arbitrarily closely if the bin grid in the output space and the edgewise subdivision of the input polytopes are chosen fine enough.

Theorem 5. *Let P be a PPD in the input space of a neural network f. Then the output distribution obtained by PLT can approximate the propagated PPD $P_y = f_*P$ up to an arbitrarily small error.*

Proof. We first claim that for a given bin grid S in the output space, our method can approximate the true probability mass of P_y in each bin $b \in S$ up to an arbitrarily small error. Consider a subdivision $\mathbb{A}_{edge}$ of the input space D such that f is linear and the input PD is approximated by a piecewise constant PD P' given by a constant PDF $c_i \in \mathbb{R}$ on each $A_i \in \mathbb{A}_{edge}$ as defined in Eq. 3. We denote the pushforward of P' with respect to f by $P'_y = f_*P'$. Let $b \in S$ and $\varepsilon > 0$. The entire probability mass of P' in each A_i is assigned to the output bin containing the image $f(t_i)$ of the center t_i of A_i. We first show that the probability mass of $P_y{}'$ in B can be approximated up to an error ε by PLT. Denote the output PD obtained by PLT by P_{PLT}. For any $A_i \in \mathbb{A}_{edge}$, the fraction of the probability mass $P'(A_i)$ that is propagated into B is correct if $f(A_i) \subset B$ or $f(A_i) \cap B = \emptyset$. If $f(A_i)$ is only partially contained in B, then a fraction of the probability mass $P(A_i)$ is either incorrectly propagated into B (if $f(t_i) \in B$) or incorrectly not propagated into B (if $f(t_i) \notin B$). By the construction of the edgewise subdivision, the total volume (and, therefore, the total probability mass with respect to P') of all polytopes A_i with images $f(A_i)$ not contained in a single bin becomes arbitrarily small if the subdivision is chosen fine enough. In particular, there is a $K \in \mathbb{N}$ such that $\left|P'_y(B) - P_{PLT}(B)\right| < \varepsilon/2$ for all $B \in S$ and all $k \geq K$ if $\mathbb{A}^k_{edge}$ is chosen as the input subdivision. Furthermore, the difference between P and P' becomes arbitrarily small for fine enough subdivision by Lemma 3. Hence, the difference between their respective pushforwards also becomes arbitrarily small. Therefore, there exists a $K' \in \mathbb{N}$ such that $\left|P_y(B) - P'_y(B)\right| < \varepsilon/2$ for all $b \in S$ and for

all $k \geq K'$ if $\mathbb{A}_{edge}^{k}$ is chosen as input subdivision. It follows immediately that $\left|P_y(B) - P_{PLT}(B)\right| < \varepsilon$ for all bins $B \in S$ if k is chosen large enough which proves our first claim.

It is well-known that the true output distribution can be approximated arbitrarily closely by a piecewise constant PD if the underlying bin grid S is chosen fine-granular enough. Together with the first claim, this yields that for fine enough bin grid and input subdivision, the piecewise constant output PD obtained by PLT can approximate the true output PPDF up to an arbitrarily small error. □

Thus, we have shown that for increasingly fine-grained subdivision in the input and bin grid in the output, the result of PLT converges to the true propagated distribution.

Appendix D: Dependencies of Neurons

In this section, we will show that in general neurons in the same layer of a NN are dependent and that the assumption of independence between neurons can lead to large errors in the propagated PD. To achieve this, we consider a setting where the input neurons are given by independent Gaussian distributions, i.e., the input PD is a Gaussian with diagonal covariance matrix. We proof that even in this case, the neurons in the first hidden layer are only independent if the weight matrix satisfies specific requirements (Lemma 5). We then analyze the difference between the true distribution in a NN layer and the distribution resulting from assuming independent neurons, measured by the Hellinger distance introduced in Appendix C. We give a maximality criterion to determine when the Hellinger distance between two normal distributions assumes the maximal value of 1 (Lemma 6). We then use this maximality criterion to proof theorems yielding a wide range of examples of covariance matrices and weight matrices where the assumption of independent neurons in the hidden layer results in a maximal Hellinger distance of 1 between the true PD in the hidden layer and the PD obtained by assuming independence (Theorems 7 and 8). Finally, we show some formulas for the Hellinger distance between a Gaussian distribution with non-diagonal covariance matrix and the Gaussian distribution resulting from restricting the covariance matrix to its diagonal (Theorem 9). These allow quantifying the error that would be made even in the input space if the input neurons are dependent and normally distributed but are modelled as independent. The formula depends on a set of eigenvalues, and it can be seen that it may result in large errors, measured by the Hellinger distance, as well.

Let k be a positive integer, μ be an element of $\mathbb{R}^k$ and Σ be a symmetric positive definite (s.p.d.) $k \times k$-matrix. We denote the PDF of the Gaussian distribution with mean μ and covariance matrix Σ by $N(\mu, \Sigma)$ and the identity matrix of dimension n by E_n.

Lemma 4 (Propagation of Gaussian distributions). *Let W be a non-singular $k \times k$-matrix. Then $W_* N(\mu, \Sigma) = N(W\mu, W\Sigma W^T)$.*

Proof. By the bijective case of Theorem 1, we can directly compute

$$
\begin{aligned}
&W_* N(\mu, \Sigma)(y) = N(\mu, \Sigma)(W^{-1}y)|W^{-1}| \\
&= \frac{|W^{-1}|}{(\sqrt{(2\pi)^k|\Sigma|}} \exp\left(-\frac{1}{2}(W^{-1}y - \mu)^T\Sigma^{-1}(W^{-1}y - \mu)\right) \\
&= \frac{1}{|W|\sqrt{(2\pi)^k|\Sigma|}} \exp\left(-\frac{1}{2}\left(W^{-1}(y - W\mu)\right)^T\Sigma^{-1}\left(W^{-1}(y - W\mu)\right)\right) \\
&= \frac{1}{\sqrt{(2\pi)^k|W\Sigma W^T|}} \exp\left(-\frac{1}{2}(y - W\mu)^T W^{-T}\Sigma^{-1}W^{-1}(y - W\mu)\right) = N\left(W\mu, W\Sigma W^T\right)
\end{aligned}
$$

which proves the claim. □

Definition 10. *We call a $k \times k$-matrix W permutation diagonal if W is diagonal up to a permutation of its columns.*

Lemma 5. *Let W be a non-singular $k \times k$-matrix such that for all diagonal s.p.d. matrices Σ the matrix $W\Sigma W^T$ is diagonal. Then W is permutation diagonal.*

Proof. By replacing W with W^{T} we can assume that $W^{\mathrm{T}}\Sigma W$ is diagonal for each diagonal s.p.d. matrix Σ. Denote the columns of W by $(w_i)_{i=1,\dots,k}$. For any $i \neq j$, we obtain

$$0 = \left(W^{\mathrm{T}}\Sigma W\right)_{i,j} = e_i^{\mathrm{T}} W^{\mathrm{T}}\Sigma W e_j = w_i^{\mathrm{T}}\Sigma w_j.$$

In other words, w_i is orthogonal to Σw_j for all $i \neq j$. As W is non-singular and $\Sigma = E_k$ is a valid choice, $(w_i)_i$ is an orthogonal basis. Now fix an index j. Because Σw_j is orthogonal to w_i for all $i \neq j$, it follows that Σw_j is a multiple of w_j. Since this holds

for all diagonal s.p.d. matrices Σ, we can conclude that the vector w_j has a single non-zero entry. Since W is non-singular, this proves the claim. □

Theorem 6. *We denote by pr_i the projection to the i-th component. For a non-singular $k \times k$-matrix W the following are equivalent:*

1. *For each probability measure μ on $\mathbb{R}^k$ such that the family of random variables $(pr_i)_{i=1,\dots,k}$ is independent, $(pr_i W)_{i=1,\dots,k}$ is independent as well.*
2. *The matrix W is permutation-diagonal.*

Proof. We first show that 2 implies 1. Let W be permutation-diagonal. Then $(\mathrm{pr}_i W)_i = \left(a_i \mathrm{pr}_{\sigma(i)}\right)_i$ for a permutation σ and a_i in $\mathbb{R}$. The independence of $(\mathrm{pr}_i)_{i=1,\dots,k}$ implies the independence of $\left(a_i \mathrm{pr}_{\sigma(i)}\right)_{i=1,\dots,k}$. To prove the converse, specializing condition 1 to non-degenerate normal distributions and applying Lemma 4 yields that for each s.p.d. diagonal matrix Σ the matrix $W\Sigma W^{\mathrm{T}}$ is diagonal. By Lemma 5, this implies that W is permutation-diagonal. □

Lemma 6 (Maximality criterion). *Let Σ_1, Σ_2 be s.s.p.d. $n \times n$ matrices. Then $H(\Sigma_1, \Sigma_2) = 1$ if and only if $supp(\Sigma_1) \neq supp(\Sigma_2)$.*

Proof. As stated in Appendix C, $H(\Sigma_1, \Sigma_2) = 1$ if and only if $N(0, \Sigma_1)$ and $N(0, \Sigma_2)$ are mutually singular. This is equivalent to $supp(\Sigma_1) \neq supp(\Sigma_2)$. □

For the following discussion, we fix a $k \times n$ matrix W and a s.s.p.d. matrix Σ. The matrix Σ defines a scalar product $\langle \cdot, \cdot \rangle_\Sigma$ on $\mathbb{R}^n$ by $\langle a, b \rangle_\Sigma = a^T \Sigma b$ for $a, b \in \mathbb{R}^n$. We denote by w^i the i-th row of W viewed as a column vector. Further let $(e_i)_{i=1,\dots,k}$ denote the standard basis of $\mathbb{R}^k$. For any matrix M, we denote by M_d the diagonal matrix containing only the diagonal entries of M, i.e., $(M_d)_{ij} = M_{ij}$ for $i = j$ and $M_{ij} = 0$ for $i \neq j$.

Lemma 7. *The Hellinger distance $H(W\Sigma W^T, (W\Sigma W^T)_d)$ is maximal if and only if*

$$supp(W\Sigma W^T) \neq \left\langle e_i \middle| \langle w^i, \Sigma w^i \rangle \neq 0 \right\rangle.$$

Proof. By the maximality criterion, it suffices to show $\left\langle e_i \middle| \langle w^i, \Sigma w^i \rangle \neq 0 \right\rangle = supp((W\Sigma W^T)_d)$. The support of $(W\Sigma W^T)_d$ is given by $\langle e_i | ((W\Sigma W^T)_d)_{ii} \neq 0 \rangle$. By definition the i-th diagonal entry of $(W\Sigma W^T)_d$ is given by $\langle w^i, \Sigma w^i \rangle$ and the claim follows immediately. □

Theorem 7. *If W is not surjective and does not have a zero row and Σ is non-singular, then*

$$H(W\Sigma W^T, (W\Sigma W^T)_d) = 1.$$

Proof. Since W is not surjective, we have $\dim supp(W\Sigma W^T) \neq k$. As each row of W is non-zero the dimension of $\left\langle e_i \middle| \langle w^i, w^i \rangle_\Sigma \neq 0 \right\rangle = \mathbb{R}^k$ is k. Thus, it follows from Lemma 7 that $H(W\Sigma W^T, (W\Sigma W^T)_d) = 1$. □

For s in $\{1, \dots, n\}$, we denote by D_s the $n \times n$ matrix satisfying $(D_s)_{s,s} = 1$ where every other matrix entry is 0.

Lemma 8. *If the s-th column of W has 2 non-zero entries, then $H(WD_sW^T, (WD_sW^T)_d) = 1$.*

Proof. Since D_s has rank 1, it follows that WD_sW^T has rank at most 1. In other words, $supp(WD_sW^T)$ has dimension at most 1. We can compute

$$(WD_sW^T)_d = diag\left(W_{1,s}^2, \dots, W_{k,s}^2\right).$$

Therefore, the dimension of $supp((WD_sW^T)_d)$ is given by the number of non-zero entries of the s-th column of the matrix W, which is greater than 1. This implies $supp(WD_sW^T) \neq supp((WD_sW^T)_d)$. □

Using the maximality criterion from Lemma 6, we characterize the matrices W for which there exists some s.s.p.d. matrix Σ with the property that $H(W\Sigma W^T, (W\Sigma W^T)_d) = 1$.

Theorem 8. *The following are equivalent:*

1. *There exists a diagonal s.s.p.d. matrix Σ such that $H(W\Sigma W^T, (W\Sigma W^T)_d) = 1$.*
2. *The matrix W has a column with at least 2 non-zero entries.*

Proof. The fact that 2 implies 1 follows from Lemma 8 as the matrix D_s is s.s.p.d for all $s = 1, \dots, n$. We show that 2 implies 1 via contradiction. Assume that W does not have a column with at least 2 non-zero entries and let Σ be a diagonal s.s.p.d matrix. Then a direct computation shows $W\Sigma W^T = (W\Sigma W^T)_d$, which implies $H(W\Sigma W^T, (W\Sigma W^T)_d) = 0 \neq 1$. □

From Theorems 7 and 8, we get a large set of diagonal s.s.p.d matrices Σ and matrices W for which the Hellinger distance $H(W\Sigma W^T, (W\Sigma W^T)_d)$ is maximal. In particular, this means that if the input distribution is given by a normal distribution with covariance matrix Σ and the weight matrix of the first layer of a NN is given by a matrix W such that W and Σ fulfill the conditions discussed in Theorem 7 or Theorem 8, the assumption of independent neurons in the first hidden layer of the neural network results in a distribution $N(0, (W\Sigma W^T)_d)$ with maximal Hellinger distance $H(W\Sigma W^T, (W\Sigma W^T)_d) = 1$ to the true distribution $N(0, W\Sigma W^T)$ in the first hidden layer. Note that a mean of 0 can be assumed here without loss of generality.

In the remainder of this section, we want to give a formula for the distance $H(W\Sigma W^T, (W\Sigma W^T)_d)$ – under appropriate assumptions on W and Σ – if it is not maximal. More precisely, we will consider the case $supp(W\Sigma W^T) = supp((W\Sigma W^T)_d) = \mathbb{R}^k$.

Lemma 9. *Let Σ_1, Σ_2 be s.p.d. matrices, then*

$$H^2(\Sigma_1, \Sigma_2) = 1 - \left(\frac{\det(\Sigma_1)\det(\Sigma_2)}{\det\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^2} \right)^{\frac{1}{4}}.$$

Proof. Pardo (2005). □

For a s.p.d. matrix Σ, we define the correlation matrix associated to Σ by

$$\Sigma_{cor} := \sqrt{\Sigma_d}^{-1} \Sigma \sqrt{\Sigma_d}^{-1}.$$

Here, $\sqrt{\Sigma_d}$ is computed by applying the square root to each entry of Σ_d. Further, $\sqrt{\Sigma_d}^{-1}$ exists as each entry of Σ_d is positive. By the multiplicativity of the determinant, it follows that $H(\Sigma, \Sigma_d) = H(\Sigma_{cor}, 1)$.

Theorem 9. *Denote by $(\mu_i)_{i=1,\dots,n}$ the eigenvalues of Σ_{cor}. Then the equation*

$$H^2(\Sigma, \Sigma_d) = H^2(\Sigma_{cor}, E_n) = 1 - \left(\prod_{i=1}^{n} \frac{\mu_i}{\left(\frac{1+\mu_i}{2}\right)^2} \right)^{\frac{1}{4}}$$

holds.

Proof. We compute

$$H^2(\Sigma, \Sigma_{\mathrm{d}}) = H^2(\Sigma_{cor}, E_n) = 1 - \frac{\det(\Sigma_{cor})^{\frac{1}{4}} \det(E_n)^{\frac{1}{4}}}{\det\left(\frac{\Sigma_{cor}+1}{2}\right)^{\frac{1}{2}}} = 1 - \left(\prod_{i=1}^{n} \frac{\mu_i}{\left(\frac{1+\mu_i}{2}\right)^2} \right)^{\frac{1}{4}}.$$

Here we used that the eigenvalues of $\frac{1}{2}(\Sigma_{cor} + E_n)$ are $\left(\frac{1}{2}(\mu_i + 1)\right)_i$. □

Note that if Σ is diagonal, Σ_{cor} is equal to E_n and the above formula yields a Hellinger distance of 0. Applying Theorem 9 to $W\Sigma W^T$ we obtain:

Corollary 2. *If W is surjective and Σ s.p.d, then*

$$H^2(W\Sigma W^T, (W\Sigma W^T)_d) = 1 - \left(\prod_{i=1}^{n} \frac{\sigma_i}{\left(\frac{1+\sigma_i}{2}\right)^2} \right)^{\frac{1}{4}},$$

where $(\sigma_i)_{i=1,\dots,k}$ denotes the family of eigenvalues of $(W\Sigma W^T)_{cor}$.

Proof. By assumption Σ is s.p.d., which means that $W\Sigma W^T$ is s.p.d. as well. Hence, we can apply Theorem 9 which proves the claim. □

Based on Theorem 9, we see that if the true input PD contains dependencies between neurons, the assumption of independence can lead to large errors depending on the eigenvalues of Σ_{cor}. Corollary 2 can be used to find more examples of covariance matrices and weight matrices where the assumption of independence in the propagated distribution can lead to large values of the Hellinger distance. In Theorems 7 and 8, we have already seen that even the maximal value of 1 can be attained depending on the weight matrix of the NN layer. Thus, we have shown that even under the restrictive assumption that the input neurons are independent and normally distributed, neglecting potential dependencies between neurons in subsequent layers generally results in large errors between the propagated PD and the true PD.

Appendix E: Datasets and Models

In this section, we provide additional information about the datasets on which our method was evaluated. The evaluation was performed on datasets widely used for classification and regression tasks which are shown in Table 2. All datasets are publicly available, and sources are provided.

Dataset	Features	Instances	Reference
Appendicitis	7	106	Wang, Zhang and Min 2019
Banana	2	5300	Jaichandaran 2023
Balance	4	625	Siegler 1976
Bands	35	541	Evans 1994
Boston	13	506	Harrison and Rubinfeld 1978
Breast	30	569	Wolberg 1990
California	8	20640	Kelley Pace and Barry 1997
Diabetes	10	442	Efron et al. 2004
Iris	4	150	Fisher 1936
Machine	9	209	Feldmesser 1987
Real Estate	7	414	Unknown 2018
Vehicle	18	847	Mowforth and Shepherd 1987
Wine	13	178	Aeberhard and Forani 1992

Table 2: Datasets

For each dataset, instances with naturally missing attribute values were excluded before deleting values and applying MICE imputation as described in the evaluation. We provide the code used for our experiments in a publicly available repository[3].

[3] https://github.com/URWI2/Piecewise-Linear-Transformation

References

Abdelaziz, A. H.; Watanabe, S.; Hershey, J. R.; Vincent, E.; and Kolossa, D. 2015. Uncertainty Propagation through Deep Neural Networks. In Interspeech.

Astudillo, R. F. and Neto, J. 2011. Propagation of Uncertainty through Multilayer Perceptrons for Robust Automatic Speech Recognition. In Interspeech.

Ayhan, M. S. and Berens, P. 2018. Test-Time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks. In Proceedings of the 1st Conference on Medical Imaging with Deep Learning.

Chen, L.; Lin, S.; Lu, X.; Cao, D.; Wu, H.; Guo, C.; Liu, C.; and Wang, F.-Y. 2021. Deep Neural Network Based Vehicle and Pedestrian Detection for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*. 22 (6): 3234–3246. doi.org/10.1109/tits.2020.2993926.

Delaunay, B. 1934. Sur la Sphere Vide. *Bulletin of Academy of Sciences of the USSR*. 7 (6): 793–800.

Edelsbrunner, H. and Grayson, D. R. 2000. Edgewise Subdivision of a Simplex. *Discrete and Computational Geometry*. 24 (4): 707–719. doi.org/10.1007/s4540010063.

Gal, Y. 2016. Uncertainty in Deep Learning. PhD dissertation, Department of Engineering, University of Cambridge, Cambridge.

Gast, J. and Roth, S. 2018. Lightweight Probabilistic Deep Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 3369–3378. IEEE Computer Society.

Goulet, J.-A.; Nguyen, L. H.; and Amiri, S. 2021. Tractable Approximate Gaussian Inference for Bayesian Neural Networks. *The Journal of Machine Learning Research*. 22 (1): 11374–11396. doi.org/10.5555/3546258.3546509.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. 2017. On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning: 1321–1330.

Hanin, B. and Rolnick, D. 2019. Deep ReLU Networks Have Surprisingly Few Activation Patterns. In Proceedings of the 33rd Conference on Neural Information Processing Systems: 361–370.

Hendrycks, D. and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. arXiv:1610.02136v3.

Hu, X.; Liu, W.; Bian, J.; and Pei, J. 2020. Measuring Model Complexity of Neural Networks with Curve Activation Functions. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Huang, Z.; Lv, C.; Xing, Y.; and Wu, J. 2021. Multi-Modal Sensor Fusion-Based Deep Neural Network for End-to-End Autonomous Driving With Scene Understanding. *IEEE Sensors Journal*. 21 (10): 11781–11790. doi.org/10.1109/jsen.2020.3003121.

Ji, W.; Ren, Z.; and Law, C. 2019. Uncertainty Propagation in Deep Neural Network Using Active Subspace. arXiv:1903.03989.

Jia, X. Y.; Jiang, C.; Fu, C. M.; Ni, B. Y.; Wang, C. S.; and Ping, M. H. 2019. Uncertainty Propagation Analysis by an Extended Sparse Grid Technique. *Frontiers of Mechanical Engineering*. 14 (1): 33–46.

Jin, J.; Dundar, A.; and Culurciello, E. 2015. Robust Convolutional Neural Networks under Adversarial Noise. arXiv:1511.06306v2.

Julier, S. J. and Uhlmann, J. K. 1997. New Extension of the Kalman Filter to Nonlinear Systems. In Signal Processing, Sensor Fusion, and Target Recognition VI: 182–193.

Julier, S. J. and Uhlmann, J. K. 2004. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*. 92 (3): 401–422. doi.org/10.1109/jproc.2003.823141.

Kendall, A. and Gal, Y. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Proceedings of the 31st Conference on Neural Information Processing Systems: 5574–5584.

Liao, X.; Zhou, T.; Zhang, L.; Hu, X.; and Peng, Y. 2023. A Method for Calculating the Derivative of Activation Functions Based on Piecewise Linear Approximation. *Electronics*. 12 (2): 267. doi.org/10.3390/electronics12020267.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In 2015 IEEE Conference on Computer Vision and Pattern Recognition.

Nunez, F.; Langarica, S.; Diaz, P.; Torres, M.; and Salas, J. C. 2020. Neural Network-Based Model Predictive Control of a Paste Thickener Over an Industrial Internet Platform. *IEEE Transactions on Industrial Informatics*. 16 (4): 2859–2867. doi.org/10.1109/tii.2019.2953275.

Pawlicki, M.; Kozik, R.; and Choraś, M. 2022. A Survey on Neural Networks for (Cyber-) Security and (Cyber-) Security of Neural Networks. *Neurocomputing*. 500: 1075–1087. doi.org/10.1016/j.neucom.2022.06.002.

Roy, A.; Conjeti, S.; Navab, N.; and Wachinger, C. 2019. Bayesian QuickNAT: Model Uncertainty in Deep Whole-Brain Segmentation for Structure-Wise Quality Control. *NeuroImage*. 195: 11–22.

Sattelberg, B.; Cavalieri, R.; Kirby, M.; Peterson, C.; and Beveridge, R. 2020. Locally Linear Attributes of ReLU Neural Networks. arXiv:2012.01940.

Smieja, M.; Struski, Ł.; Tabor, J.; Zieliński, B.; and Spurek, P. 2018. Processing of Missing Data by Neural Networks. arXiv:1805.07405v3.

Takenaka, K.; Ohtsuka, K.; Fujii, T.; Negi, M.; Suzuki, K.; Shimizu, H.; Oshima, S.; Akiyama, S.; Motobayashi, M.; Nagahori, M.; Saito, E.; Matsuoka, K.; and Watanabe, M. 2020. Development and Validation of a Deep Neural Network for Accurate Evaluation of Endoscopic Images from Patients With Ulcerative Colitis. *Gastroenterology*. 158 (8): 2150–2157. doi.org/10.1053/j.gastro.2020.02.012.

Titensky, J. S.; Jananthan, H.; and Kepner, J. 2018. Uncertainty Propagation in Deep Neural Networks Using Extended Kalman Filtering. In 2018 IEEE MIT Undergraduate Research Technology Conference: 1–4.

Truong, D. 2021. Estimating the Impact of COVID-19 on Air Travel in the Medium and Long Term Using Neural Network and Monte Carlo Simulation. *Journal of Air Transport*

Management. 96: 102126. doi.org/10.1016/j.jairtraman.2021.102126.

van Buuren, S. and Groothuis-Oudshoorn, K. 2011. MICE: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*. 45 (3): 1–67. doi.org/10.18637/jss.v045.i03.

Vigneswaran, R. K.; Vinayakumar, R.; Soman, K. P.; and Poornachandran, P. 2018. Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. In 9th International Conference on Computing, Communication and Networking Technologies.

Wang, H.; Shi, X. j.; and Yeung, D.-Y. 2016. Natural-Parameter Networks: A Class of Probabilistic Neural Networks. In Proceedings of the 30th Conference on Neural Information Processing Systems: 118–126.

Wilson, A. and Izmailov, P. 2020. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In Proceedings of the 34th Conference on Neural Information Processing Systems: 4697–4708.

Wu, A.; Nowozin, S.; Meeds, E.; Turner, R. E.; Hernández-Lobato, J. M.; and Gaunt, A. L. 2018. Deterministic Variational Inference for Robust Bayesian Neural Networks. arXiv:1810.03958v2.

Yu, H.; Yang, L. T.; Zhang, Q.; Armstrong, D.; and Deen, M. J. 2021. Convolutional Neural Networks for Medical Image Analysis: State-of-the-Art, Comparisons, Improvement and Perspectives. *Neurocomputing*. 444: 92–110. doi.org/10.1016/j.neucom.2020.04.157.

Zhang, B. and Shin, Y. C. 2021. An Adaptive Gaussian Mixture Method for Nonlinear Uncertainty Propagation in Neural Networks. *Neurocomputing*. 458: 170–183. doi.org/10.1016/j.neucom.2021.06.007.

Zhang, W.; Li, X.; Ma, H.; Luo, Z.; and Li, X. 2021. Federated Learning for Machinery Fault Diagnosis with Dynamic Validation and Self-Supervision. *Knowledge-Based Systems*. 213: 106679. doi.org/10.1016/j.knosys.2020.106679.