

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 $D = \cup_i A_i$. 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 $f: D \rightarrow \mathbb{R}^n$ 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 $f: D \rightarrow \mathbb{R}^n$ be the function representing a NN (with n -dimensional output space). Then the function f has the form*

$$f(x) = \begin{cases} W_1x + b_1, & \text{if } M_1x \leq c_1, \\ W_2x + b_2, & \text{if } M_2x \leq c_2, \\ \dots & \dots \\ W_t x + b_t, & \text{if } M_t x \leq c_t \end{cases} \quad (1)$$

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 $D = \cup_{i=1}^t A_i$. 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 $P: \mathcal{A} \rightarrow \mathbb{R}$ piecewise with respect to a set of subsets $\mathbb{A} = \{A_1, A_2, \dots, A_k \mid A_i \subset D\}$ if there exists a set of functions $\{p_i: A_i \rightarrow \mathbb{R}\}$, such that $P(X) = \sum_{i=1}^k \int_{A_i \cap X} p_i d\mu_{A_i}$ 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 $p: D \rightarrow \mathbb{R}$ 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 $f: D \rightarrow \mathbb{R}^n$ 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) $\bar{X}: (\Omega, \mathcal{F}, \mu) \rightarrow \mathbb{R}^m$ 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 $W: \mathbb{R}^m \rightarrow \mathbb{R}^n$ 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 $W: \mathbb{R}^m \rightarrow \text{Im}(W)$ 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., $W_*P = W_*p_X dx$. 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 $W: \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a surjective linear map and p_X be a PDF as above. Then $p_Y = W_*p_X$ follows the formula*

$$p_Y(y) = \int_{W^{-1}(y)} p_X(x) |\det \tilde{W}^{-1}| dx$$

almost everywhere. Hereby \tilde{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 $\pi: \mathbb{R}^m \rightarrow \ker W^\perp$ the orthogonal projection. By transformation of variables, we get

$$\begin{aligned} \int_Y p_Y(y) dy &= \int_{W^{-1}Y} p_X(x) dx = \int_{\ker W + \pi(W^{-1}Y)} p_X(x) dx \\ &= \int_{\pi(W^{-1}Y)} \int_{\ker W} p_X(x + y) dx dy \\ &= \int_Y \int_{\ker W} p_X(x + \tilde{W}^{-1}y) |\det \tilde{W}^{-1}| dx dy. \end{aligned}$$

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

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 $1 \leq i \leq k$ 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*

$$f_*p(y) = \sum_{\substack{i=1 \\ y \in f_i(A_i)}}^k \frac{1}{\det \tilde{f}_i} \int_{f_i^{-1}(y) \cap A_i} p_i(x) dx. \quad (2)$$

In this formula, \tilde{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 $p_i: A_i \rightarrow \mathbb{R}$ and the function $f_i: \mathbb{R}^m \rightarrow \mathbb{R}^n$. 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 \bar{p}_i on \mathbb{R}^m by

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

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

$$\begin{aligned} (f_i)_*p_i(y) &= \frac{1}{\det(\tilde{f}_i)} \int_{f_i^{-1}(y)} \bar{p}_i(x) dx \\ &= \frac{1}{\det(\tilde{f}_i)} \int_{f_i^{-1}(y) \cap A_i} p_i(x) dx. \end{aligned}$$

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 $f(\mathbb{A}) := \{f(A_1), \dots, f(A_k)\}$ and the functions $(f_i)_*p_i: f(A_i) \rightarrow \mathbb{R}$. In summary, the statement given by Eq. 2 follows. \square

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., $\text{dist}(c_1, c_2) < \varepsilon, \forall c_1, c_2 \in C(A)$ 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

$$\mathbb{A}_{edge}^\varepsilon = \{A_{1,1}^\varepsilon, A_{1,2}^\varepsilon, \dots, A_{1,l_{1,\varepsilon}}^\varepsilon, A_{2,1}^\varepsilon, \dots, A_{k,l_{k,\varepsilon}}^\varepsilon\}, \quad (3)$$

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

$$p': D \rightarrow \mathbb{R}, p'(x) = c_{ij} \\ \text{with } c_{ij} := \frac{1}{\dim(A_i) + 1} \sum_{c \in C(A_{i,j}^\varepsilon)} p_i(c), \quad (4)$$

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

By this definition, p' is constant on each edgewise simplex $A_{i,j}^\varepsilon$, attaining the average value of p evaluated in the vertices of $A_{i,j}^\varepsilon$. 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:*

$$f_*p'(y) = \sum_{i=1}^k \sum_{\substack{j=1, \\ y \in f_i(A_{i,j})}}^{l_i} \frac{c_{ij}}{\det \tilde{f}_i} \text{vol}(f_i^{-1}(y) \cap A_{i,j}). \quad (5)$$

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

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 $c_{ij} \cdot \text{vol}(A_{i,j})$) 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

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

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 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 PD¹ 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 literature² (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 of our 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.

Data-sets	% 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
Appen- dicitis	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
Califor- nia	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
Diabe- tes	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
Ma- chine	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.

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.; Dunder, 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.