# Tukey g-and-h neural network regression for non-Gaussian data

**Anonymous authors**
**Paper under double-blind review**

## Abstract

This paper addresses non-Gaussian regression with neural networks via the use of the Tukey g-and-h distribution. The Tukey g-and-h transform is a flexible parametric transform with two parameters $g$ and $h$ which, when applied to a standard normal random variable, introduces both skewness and kurtosis, resulting in a distribution commonly called the Tukey g-and-h distribution. Specific values of $g$ and $h$ produce good approximations to other families of distributions, such as the Cauchy and student-t distributions. The flexibility of the Tukey g-and-h distribution has driven its popularity in the statistical community, in applied sciences and finance. In this work we consider the training of a neural network to predict the parameters of a Tukey g-and-h distribution in a regression framework via the minimization of the corresponding negative log-likelihood, despite the latter having no closed-form expression. We demonstrate the efficiency of our procedure in simulated examples and apply our method to a real-world dataset of global crop yield for several types of crops. Finally, we show how we can carry out a goodness-of-fit analysis between the predicted distributions and the test data. A Pytorch implementation is made available on Github and as a Pypi package.

## 1 Introduction

The application of Deep Learning to highly non-linear regression tasks has brought unprecedented results in many real-world applications. In many practical regression problems, it is desirable to predict a probabilistic distribution rather than a single-point value. This is key when machine learning algorithms are used in complex decision-making processes. For instance, if a neural network is used to predict a stock price or the duration of a surgical operation (Ng et al., 2017), confidence intervals will be far more valuable than single point predictions. One must distinguish between different sources of uncertainty: model estimation uncertainty — which we do not consider as it is not the main topic of this paper but which could be addressed via Bayesian Neural Networks (Neal, 2012) or Bootstrapping (Lee et al., 2020) — and what one might call inherent uncertainty, that is, the remaining randomness even if the true regression model was known exactly. We note that inherent uncertainty may not only arise from the generating process, but also from the potentially limited set of features used for regression. One common approach to account for inherent uncertainty consists in predicting, for a given set of features, the two parameters of a Gaussian distribution, rather than a single-point value. This is particularly valuable in the case where the conditional variance of the target also depends on the features. However, the Gaussian assumption itself might be quite restrictive, as it does not allow for skewness and kurtosis of the target variable conditioned on the features. These patterns may arise in the target variable either from the generating process itself, or from using an incomplete set of features. One way to address non-Gaussianity in the target variable is to consider more flexible parametric families of distributions. For instance, one might train a neural network to output the parameters of a mixture of Gaussian distributions rather than a single Gaussian distribution. This might be useful in particular when the conditional distribution of the target is multi-modal, but remains limited, in particular when it comes to modelling heavy tails. It is also worth mentioning non-parametric approaches such as quantile regression (Xu et al., 2017; Taylor, 2000). In this paper, however, we focus on the parametric approach, more specifically on the Tukey g-and-h probability distribution (Jorge & Boris, 1984). This distribution is obtained by applying

the Tukey g-and-h transform with parameters $g$ and $h$ to a standard normal random variable, which is then rescaled by a multiplicative factor $\sigma$ and shifted by an additive constant $\mu$. As such, the Tukey g-and-h distribution has four parameters. The Tukey g-and-h distribution has been widely used in modern statistical geosciences (Xu & Genton, 2017; Jeong et al., 2019), spatio-temporal modelling (Murakami et al., 2021), but also in financial risk analysis (M. Bee & Trapin, 2021). The standard random field approach (Xu & Genton, 2017) consists in modelling spatio-temporal dependence via a Gaussian Process, followed by a pointwise mapping by the Tukey g-and-h transform to incorporate non-Gaussian patterns. In parallel, the application of neural networks to geosciences has grown exponentially, for instance to address problems such as hyper-resolution (Höhlein et al., 2020), forecasting (Ren et al., 2021) or for the parameterization of discretized non-linear PDE solvers (Guillaumin & Zanna, 2021) and yield forecasting (You et al., 2017), (Dado et al., 2020), (Wang et al., 2020). The objective of this work is to propose a method that bridges the gap between Tukey g-and-h random fields and neural network regression. In particular, we see a longer-term incentive to combine Neural Networks to learn complex feature-dependent parameters of a Tukey g-and-h transform with Gaussian Process models to incorporate residual patterns of spatio-temporal dependence. Our paper is organized as follows. In Section 2, we review the Tukey g-and-h transform and present our methodology for training and evaluating neural networks for the prediction of g-and-h distributions. This requires us to obtain the derivatives of the Tukey g-and-h log likelihood with respect to the parameters, which involves the inversion of the Tukey g-and-h transform. There is no known closed form for the latter. We propose the use of binary search to address this issue, which is efficient both numerically and computationally. In Section 3, we demonstrate the benefits of the proposed methodology based on simulated data experiments, while in Section 4 we present an application to a real-world regression problem by applying our methodology to learn patterns of global yield for several crops. Finally, we provide a Pytorch implementation of the Tukey g-and-h loss function made publicly available on the Github account of the authors.

## 2 Methodology

In this section, we present our proposed methodology for the use of the Tukey g-and-h transform in Neural Network regression. We first review the Tukey g-and-h transform and its basic properties. We then consider the evaluation of the negative log-likelihood function, which we will use as our loss function for training. In evaluating the negative log-likelihood, the main challenge lies in the inversion of the Tukey g-and-h transform. We propose to use binary search to address this problem. In comparison to other approaches proposed to approximate the inverse of the Tukey g-and-h transform such as grid search, this entails no approximation other than that incurred by numerical precision. Finally, we discuss how one can assess goodness-of-fit and obtain confidence intervals in Tukey g-and-h Neural Network regression.

### 2.1 The Tukey g-and-h transform

We review some properties of the Tukey g-and-h transform (Xu & Genton, 2017; Jorge & Boris, 1984) and its ability to approximate a wide range of other well-known families of probability distributions when applied to a standard normal random variable, such as student's t-distribution.

The Tukey g-and-h transform with parameters $g \in \mathbb{R}$ and $h \geq 0$ is defined by,

$$\tau_{g,h}(z) = \frac{\exp(gz) - 1}{g} \exp\left(\frac{1}{2} h z^2\right), \quad \forall z \in \mathbb{R}, \tag{1}$$

when $g \neq 0$. The case $g = 0$ is obtained by continuous extension,

$$\tau_{0,h}(z) = z \exp\left(\frac{1}{2} h z^2\right), \quad \forall z \in \mathbb{R}.$$

When both $g$ and $h$ are zero, the transform is just the identity function. For fixed $g$ and $h$ the transform $\tau_{g,h}(z)$ is an increasing function in $z$.

Let $Z$ be a standard normal random variable, and define,

$$\begin{cases} \widetilde{Z} & = \tau_{g,h}(Z), \\ Y & = \mu + \sigma \widetilde{Z}. \end{cases} \tag{2}$$

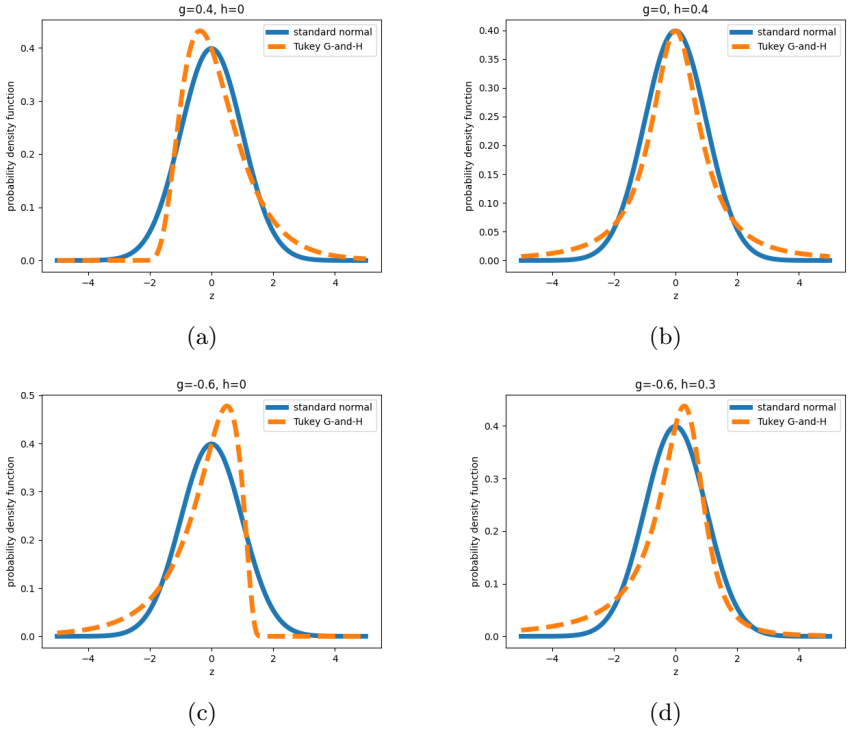



Figure 1: Comparison of the probability density function of a Tukey g-and-h distributed random variable with that of a standard normal random variable, for different values of $g$ and $h$.

where $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ roughly control the first two moments of the transformed random variable $Y$. The parameters $g$ and $h$ roughly control, respectively, the skewness and kurtosis, which are zero for the non-transformed variable $Z$. A positive value of $g$ will incur positive skewness in $Y$ — see Figure 1a— while a negative value of $g$ will incur negative skewness — see Figure 1c. Note that as $g$ and $h$ converge to zero, $Y$ converges in distribution towards a Gaussian random variable with mean $\mu$ and standard deviation $\sigma$. In this manuscript, we will say that the transformed random variable $\widetilde{Y}$ follows a G-and-H distribution. In Figure 1 we show the probability density function of $\widetilde{Z}$ for 4 combinations of the parameters $g$ and $h$, superimposed with the probability density function of a standard normal distribution. The Tukey g-and-h distribution provides a good approximation to a wide range of standard probability distributions (Jorge & Boris, 1984).

### 2.2 Training via negative likelihood loss minimization

We consider a regression problem with features $\mathbf{X}$ and target $Y$. Rather than only training a neural network to predict the conditional expectation of $Y$ given $\mathbf{X}$, we wish to approximate the full conditional distribution of $Y$ given $\mathbf{X}$ by training the neural network to predict a g-and-h distribution — that is, predict in terms of $\mathbf{X}$ the four parameters $\mu, \sigma, g$ and $h$ of such a distribution. As such, the output of our neural network should have a total of 4 neurons, one for each parameter of a g-and-h distribution. In this section, we provide details on how to train a neural network for Tukey g-and-h regression via stochastic gradient descent on the negative log-likelihood. Following standard properties of transformed random variables, the transformed random variable $Y$ defined in (2) admits a probability density function which can be expressed in terms of the probability density function $f_Z(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$ of the standard normal random variable $Z$. Specifically,

$$f_Y(y) = \frac{1}{\sigma \; \tau'_{g,h}\left(\tau^{-1}_{g,h}\left(\frac{y-\mu}{\sigma}\right)\right)} f_Z\left(\tau^{-1}_{g,h}\left(\frac{y-\mu}{\sigma}\right)\right),$$

where $\tau'_{g,h}(z)$ is the derivative of the Tukey g-and-h transform $\tau_{g,h}(\cdot)$ given by,

$$\tau'_{g,h}(z) = \left[\exp(gz) + hz\frac{\exp(gz)-1}{g}\right]\exp\left(\frac{1}{2}hz^2\right),$$

and $\tau^{-1}_{g,h}$ is the inverse function of the Tukey g-and-h transform for fixed parameters $g$ and $h$.

Given a collection of features $\mathbf{x}_i$ and response variables $y_i$, $i = 1, \ldots, n$, we denote $\mu_i, \sigma_i, g_i$ and $h_i$ the 4 components of the neural network's output for input $\mathbf{x}_i$, corresponding to the four parameters of a G-and-H distribution. If we denote $\widehat{z}_i = \tau^{-1}_{g_i,h_i}\left(\frac{y-\mu_i}{\sigma_i}\right)$, the negative log-likelihood function can be expressed in terms of the neural network's parameters $\theta$ as,

$$\begin{aligned} l(\theta) &= \sum_i \log \tau'_{g_i,h_i}(\hat{z}_i) + \sum_i \log\sigma_i + \frac{1}{2}\sum_i \hat{z}_i^2 \\ &= \sum_i \log\left[\exp(g\hat{z}_i) + h\hat{z}_i\frac{\exp(g\hat{z}_i)-1}{g}\right] + \sum_i \log\sigma_i + \sum_i \frac{1+h_i}{2}\hat{z}_i^2. \end{aligned} \tag{3}$$

The dependence of $g_i$, $h_i$, $\sigma_i$ and $\hat{z}_i$ on the neural network's parameters $\theta$ is left implicit in the notation for simplicity. The evaluation of the $\widehat{z}_i$'s and their gradient with respect to the parameters $g$ and $h$ — these are required to carry out Stochastic Gradient Descent — poses the main challenge to the evaluation of the loss. One previously proposed approach is to approximate the inverse Tukey g-and-h transform using a grid of values $z_0, \ldots, z_p$ for which $\tau_{g,h}$ is evaluated (Xu & Genton, 2017). In contrast, we propose an approach which is not limited in accuracy other than by that of the precision of the numerical implementation. More specifically, since the Tukey g-and-h transform is an increasing function of $z$, its inverse at any given point can be obtained efficiently by binary search. The only requirement is that we start the algorithm with a large enough range. We note that this binary search need only be applied once for every feedforward call of the Stochastic Gradient Algorithm.

Finally, in order to apply Stochastic Gradient Descent algorithms, we also require the derivatives of the inverse transform with respect to $g$, $h$ and $\widetilde{z}$ due to the terms $\hat{z}_i$ in the loss function (3). First, we have,

$$\frac{\partial \tau^{-1}_{g,h}}{\partial \widetilde{z}}(\widetilde{z}) = \frac{1}{\tau'_{g,h}(\tau^{-1}_{g,h}(\widetilde{z}))}. \tag{4}$$

Then, we derive,

$$\frac{\partial \tau_{g,h}}{\partial g}(z) = \frac{\exp(gz)(gz-1)+1}{g^2}\exp(\frac{1}{2}hz^2), \tag{5}$$

$$\frac{\partial \tau_{g,h}}{\partial h}(z) = \frac{z^2}{2}\tau_{g,h}(z). \tag{6}$$

We then write,

$$\tau_{g,h}\left(\tau^{-1}_{g,h}(\widetilde{z})\right) = \widetilde{z}, \tag{7}$$

and obtain the desired quantities after taking the derivative of the above equation with respect to $g$ and $h$ respectively, by application of the chain rule,

$$\frac{\partial \tau^{-1}_{g,h}}{\partial g}(\widetilde{z}) = -\frac{\frac{\partial \tau_{g,h}}{\partial g}\left(\tau^{-1}_{g,h}(\widetilde{z})\right)}{\tau'_{g,h}(\tau^{-1}_{g,h}(\widetilde{z}))}, \tag{8}$$

$$\frac{\partial \tau^{-1}_{g,h}}{\partial h}(\widetilde{z}) = -\frac{\frac{\partial \tau_{g,h}}{\partial h}\left(\tau^{-1}_{g,h}(\widetilde{z})\right)}{\tau'_{g,h}(\tau^{-1}_{g,h}(\widetilde{z}))}. \tag{9}$$

By defining a *Function* object in PyTorch that encapsulates the inverse Tukey transform and its derivatives with respect to $g$, $h$ and $\widetilde{z}$, we can obtain the gradient of the negative log-likelihood function via automatic differentiation. The code provided alongside this manuscript takes care of implementing these computations, so that the user only needs to declare a TukeyGandHLoss object and provide the four parameters output by their neural network for each data point of a mini-batch.

### 2.3 Prediction intervals

One benefit of the Tukey g-and-h distribution for the modelling of non-Gaussian random variables is that we can easily obtain prediction intervals for the target variable. Let $0 < \alpha < 1$. Due to the continuous and increasing nature of the Tukey g-and-h transform, we immediately obtain,

$$F_{\widetilde{Z}}^{-1}(\alpha) = \tau_{g,h}(\Phi^{-1}(\alpha)), \tag{10}$$

where $F_{\widetilde{Z}}^{-1}$ is the inverse cumulative distribution function of the transformed random variable, and $\Phi^{-1}$ is the inverse cumulative distribution function of the standard normal distribution.

Denote $\widehat{\theta}$ the parameters of the trained neural network. For an input $\mathbf{x}$, write

$$\mu(\mathbf{x};\widehat{\theta}), \sigma(\mathbf{x};\widehat{\theta}), g(\mathbf{x};\widehat{\theta}) \text{ and } h(\mathbf{x};\widehat{\theta}) \tag{11}$$

for the four parameters of the g-and-h distribution output by the neural network with parameters $\widehat{\theta}$ when provided with input $\mathbf{x}$. An $\alpha$-level confidence interval for a feature $\mathbf{x}$ is provided by,

$$\Big[\mu(\mathbf{x};\widehat{\theta}) + \sigma(\mathbf{x};\widehat{\theta})\tau_{g(\mathbf{x};\widehat{\theta}),h(\mathbf{x};\widehat{\theta})}(-z_{1-\alpha/2}) \ ,$$
$$\mu(\mathbf{x};\widehat{\theta}) + \sigma(\mathbf{x};\widehat{\theta})\tau_{g(\mathbf{x};\widehat{\theta}),h(\mathbf{x};\widehat{\theta})}(z_{1-\alpha/2})\Big] \ ,$$

with $z_\alpha \equiv F_Z^{-1}(\alpha)$. While this approach is simple and computationally efficient, it was noted in (Xu & Genton, 2017) that it may lead to unreasonably large prediction intervals when the skewness parameter $g$ is large in absolute value. Following their approach, one might instead consider the following prediction interval,

$$\Big[\mu(\mathbf{x};\widehat{\theta}) + \sigma(\mathbf{x};\widehat{\theta})\tau_{g(\mathbf{x};\widehat{\theta}),h(\mathbf{x};\widehat{\theta})}(-z_{1-\alpha+\gamma}) \ ,$$
$$\mu(\mathbf{x};\widehat{\theta}) + \sigma(\mathbf{x};\widehat{\theta})\tau_{g(\mathbf{x};\widehat{\theta}),h(\mathbf{x};\widehat{\theta})}(z_{1-\gamma})\Big] \ ,$$

where $0 \leq \gamma \leq \alpha$ is chosen so that it minimizes the length of the prediction interval. We note that the prediction intervals obtained in this fashion do not account for the uncertainty in $\widehat{\theta}$. While this is not the focus of this paper, a potential means of addressing this issue might be the use of bootstrapping (Sluijterman et al., 2023).

### 2.4 Goodness-of-fit analysis

The standard approach to goodness-of-fit analysis in Machine Learning relies on evaluating the loss function on a test dataset. In the case where we predict a parametric probability distribution rather than point values, we might also be interested in comparing the distribution of the predicted target values with that of the observed targets. In general, this poses some difficulty as the predicted distribution of the target depends on the features. In the case of Tukey g-and-h Neural Network regression, however, we can easily address this problem. We compare the empirical distribution of the $\hat{z}_i$'s to the standard normal distribution via standard methods such as quantile-quantile plots. Naturally, one should keep in mind that there could be compensation effects in such an analysis: a misfit could compensate another misfit. We investigate this approach both in the simulated and real-world data experiments of the next sections. A somewhat equivalent approach is to compute *uniform residuals* by applying the PIT (Probability Integral Transform) on the $\hat{z}_i$'s. Specifically, we define $u_i = \Phi(\hat{z}_i)$ where $\Phi$ is used to denote the cumulative distribution function of a standard normal random variable. We can then compare the empirical distribution of the $u_i$'s to that of a uniform random variable on the interval $[0, 1]$. Alternatively, we might want to plot the $u_i$'s in terms of the features — ideally, we should not observe any pattern of dependence.

## 3 Simulated data experiments

In this section we present some results on Tukey g-and-h neural network regression in a simulated setting. We first study the case where the true distribution of the target variable, conditioned on the features, is indeed

a Tukey g-and-h distribution. We then assess the robustness of the proposed methodology in a misspecified setting where the target variable follows a student-t distribution, conditionally on the features.

### 3.1 No model misspecification

We first treat the case where the true conditional distribution of the target variable $Y$ actually belongs to the family of Tukey g-and-h distributions. Specifically, we consider a scalar feature $X$ uniformly distributed on the interval $[0, 1]$,

$$X \sim \mathcal{U}([0, 1]). \tag{12}$$

The response variable $Y$ conditionally on $X$ is written as,

$$Y|X \sim \sigma(X)\tau_{g(X),h(X)}(Z) + \mu(X), \tag{13}$$

with $Z \sim \mathcal{N}(0, 1)$. For the purposes of this simulated data experiment, the deterministic functions $\sigma(\cdot)$, $\mu(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are set to arbitrarily chosen closed-form functions. From these definitions, we simulate a dataset of size 40,000.

We define a neural network with 5 fully-connected layers, with a scalar input (the feature $x$) and 4 outputs corresponding to the 4 functions of $X$ that appear in (13). We train this neural network using training and validation data according to a standard 80% versus 20% split. Exact details of the training procedure and functional form of the true regression functions are available in the code provided as supplementary material.

In Figure 2 we show the resulting fit between the true regression functions and the functions learnt by the neural network. In Figure 3 we compare the fitted Tukey-g-and-h distribution, and a fitted Gaussian distribution for different values of the feature $X$. Unsurprisingly, we observe that in the case of a left skew of the target variable (e.g. x=0.1, top left), the mode of the fitted Gaussian distribution is underestimated. On the contrary, in the case of a right skew of the target variable (e.g. x=0.9, bottom right), the mode of the fitted Gaussian distribution is overestimated.

### 3.2 Model misspecification

We now present a simulated data experiment subject to model misspecification. Specifically, we set the target variable to follow a t-distribution, the parameters of which are controlled by arbitrarily specified functions,

$$Y|X \sim \mu(X) + \sigma(X)T, \quad T \sim t_{\nu(X)}, \tag{14}$$

where $t_\nu$ is used to denote student's t-distribution with $\nu$ degrees of freedom. Naturally, if we know the true distribution of the target variable to follow a t-distribution, we are likely to be better off directly using that in our regression model. Here our intent is to show that our procedure is robust to a misspecified setting. As we mentionned earlier, the Tukey g-and-h provides a good approximation to student's t distribution. It is therefore natural to expect from our method that it should be robust to this slight departure from the model distribution. In stronger cases of misspecification, such as that of a multi-modal distribution, there is no reason to expect good performance from our Tukey g-and-h approach in general. One could however adapt Mixture Density Networks (Bishop, 1994) to mixtures of Tukey g-and-h distributions. While in principle, as mentioned by their author, Mixture Density Networks can approximate any distribution, the number of required components may be very large for heavy tailed distributions, an issue that could be addressed by replacing the Gaussian components of the mixture with Tukey g-and-h components. In Figure 4 we compare the fitted distribution for four values of the scalar input. The Tukey g-and-h recovers the shape of the distribution of the target variable, in comparison to the Gaussian neural network regression. This is validated by the training and validation losses, which are shown in Figure 5. Finally, we carry out a goodness-of-fit analysis by following the procedure described in Section 2.4, see Figure 9 in Appendix A, where we show a histogram of the $\widehat{z}_i$ residuals against the probability density function of a standard normal variable.

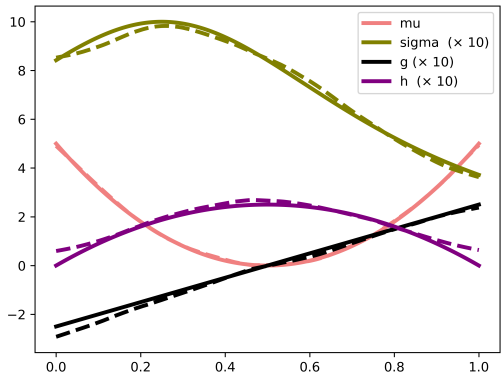



Figure 2: True regression functions (solid) and trained regression functions (dashed) on 10000 observations.

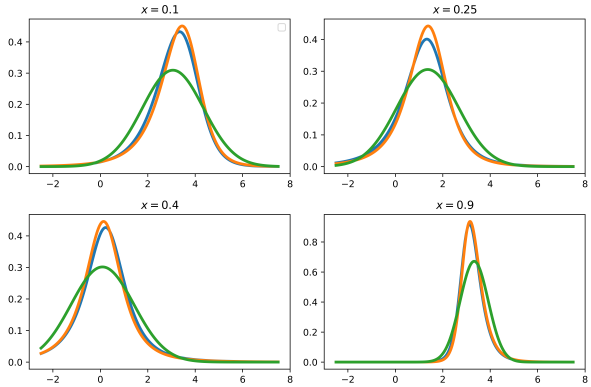



Figure 3: Comparison of true (blue) and fitted (Tukey g-and-h in orange vs Gaussian in green) probability density functions at four values of the scalar feature $x$, in the case where the target variable follows a Tukey g-and-h distribution.

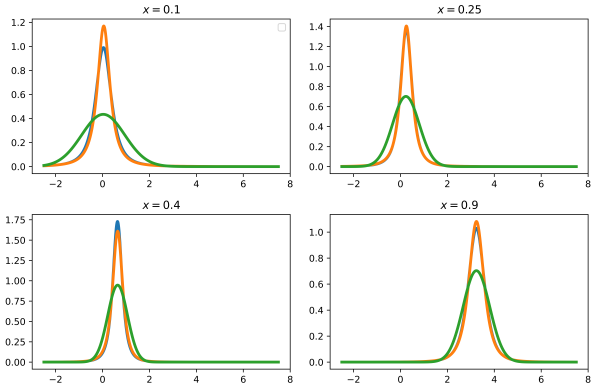



Figure 4: Comparison of true (blue) and fitted (Tukey g-and-h in orange vs Gaussian in green) probability density functions at four values of the scalar feature $x$, in the case where the target variable follows a t-distribution.

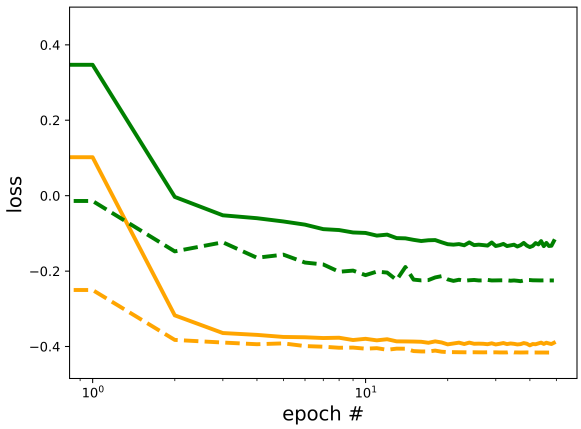



Figure 5: Tukey g-and-h (orange) and Gaussian (green) losses over training epochs for training (solid) and validation (dashed) data for a t-distributed target variable.

## 4 Real data experiment: an application to crop yield predictions

Food security is widely recognised as one of the most urgent challenges we currently face globally (Intergovernmental Panel on Climate Change (IPCC), 2023). This concern has grown in significance due to the changing climate and its warming effects. With the increasing occurrence of extreme weather events and alterations in weather patterns, our food system, especially in certain regions, has become highly vulnerable to the impacts of climate change. Agriculture is both one of the sectors most susceptible to climate change and a significant contributor to it (Mahowald et al., 2017). Therefore, it is essential to consider both mitigation and adaptation strategies, as well as transforming agricultural practices to promote sustainability and resilience. A key objective is to develop more reliable and scalable methods for monitoring global crop conditions promptly and transparently, while also exploring how we can adapt agriculture to mitigate the effects of climate change.

The key areas of research in AI for agriculture include crop mapping, crop type mapping, field boundary delineation, yield estimation, and pests and disease detection. While there are other use cases such as crop suitability, the aforementioned categories encompass the majority of research conducted in this field. We can determine yield at various scales, assess crop conditions, and combine the model outputs with other data sources, such as temperature and rainfall predictions, to derive comprehensive insights. These insights can aid in decision-making for farmers and policymakers, facilitating informed choices regarding yields, diseases, and other relevant factors.

We propose to apply our proposed methodology to global crop yield prediction since food security in general and yield prediction in particular pose one of the most pressing issues in AI for agriculture. Extreme events can lead to crop yield declines, resulting in financial losses and threats to food security. Crop yield prediction involves predicting the volume or weight of crops to be harvested in each unit area. This is a regression problem where the goal is to estimate crop yield per unit area, representing the rate of production. There is extensive literature on crop yield prediction, where it can be considered on a plant, farm, global or regional scales. In this work, we consider a global scale and therefore utilise a global dataset for crop yield prediction. Shuai et al. predicted maize yield in the United States at the pixel level, providing estimates in tonnes per hectare (Shuai & Basso, 2022). The paper by You et al. (2017) introduced a different approach, employing deep Gaussian processes to predict crop yield based on remote sensing data (You et al., 2017), with focus on estimating yield at the county scale rather than at the individual pixel level.

We investigate a yearly global yield dataset (Iizumi, 2019) for maize, wheat, rice, and soybean, spanning the period 1981-2016. For each year and each crop, the yield is provided in tons per hectare on a global grid with 0.5' spatial resolution. For instance, Figure 6 shows the global yields for maize in 2010. Our goal is to

learn, for each crop, a parametric distribution of the yield as a function of latitude, longitude, and year. We compare two approaches:

1. Gaussian prediction, based on a negative Gaussian likelihood loss function.
2. Tukey g-and-h prediction, based on a negative Tukey g-and-h likelihood loss function, as described in Section 2.2.

Our rationale for leveraging neural networks for this problem is that we expect a strongly non-linear dependence of the Tukey g-and-h parameters with respect to latitude and longitude. The architecture of our neural network is a follows: a sequence of fully connected layers with ReLU activations and batch normalization. The neural network takes the latitude and longitude as inputs to the first hidden layer, and additionally the year is concatenated to the penultimate hidden layer. The two approaches use the same neural network architecture, with the exception that they output a different number of values. For the first approach, we output two values, corresponding to the two parameters of a Gaussian distribution, its mean and variance. For the second approach, we output four values, one for each of the four parameters that define a Tukey g-and-h distribution. Training is performed according to the Adam algorithm (Kingma & Ba, 2015), and with the use of a scheduler that reduces the learning rate by a factor of 10 at epochs 10, 15, 20, 30 and 40 from an initial value of 1e−4. Finally, we use a batch size of 4096.

To compare the two approaches, we split the data, by using years 1985, 1995, 2005, 2015 for validation and 1986, 1996, 2006, 2016 for testing. All remaining years from the dataset are used for training.

To assess each approach, we compute residuals that are expected to follow a standard normal distribution under the posited models, according to the approach mentioned in Section 2.4. Therefore, we report quantile-quantile plots of those residuals with respect to the quantiles of the standard normal distribution. This is shown in Figure 7a for the validation data and Figure 8 for the test data. We note that for both the validation and test datasets, the quantile-quantile plots show a better fit for the proposed method that predicts a Tukey g-and-h distribution.

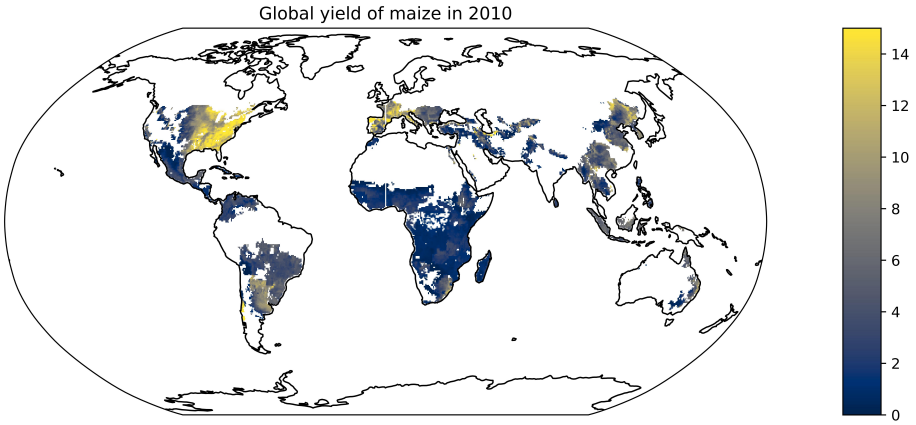
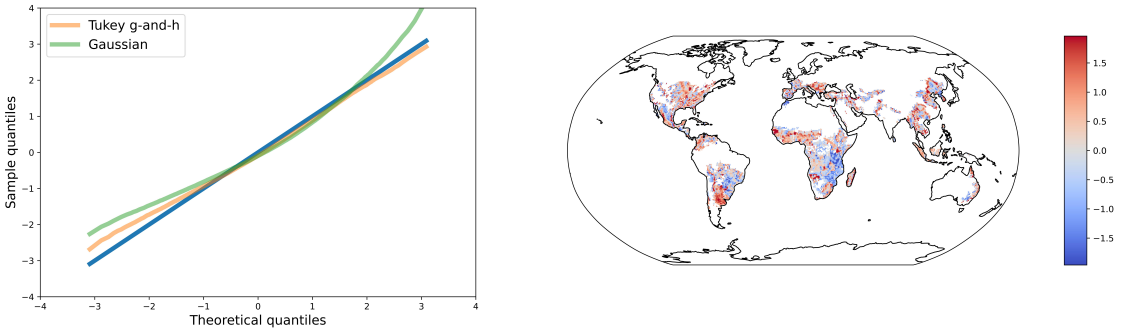



Figure 6: Global yield of maize in ton per hectare in 2010 on a 0.5' spatial-resolution grid

In future work, we will investigate the use of weather indicators to more accurately predict the yield. Average and extreme weather conditions over the year naturally play a key role in determining the yield. Additionally, the spatial dependence of weather conditions results in spatial dependence of the yield, which we wish to remove as much as possible from a methodological point of view.




(a) (b)

Figure 7: (a) quantile-quantile plot of the maize yield residuals from the validation data with respect to the standard normal distribution, (b) global map of residuals from the validation data (year 2005)

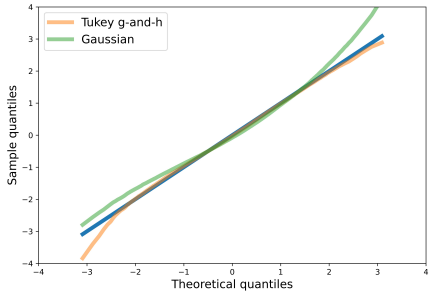



Figure 8: quantile-quantile plot of the maize yield residuals from the test data with respect to the standard normal distribution

## 5 Conclusion

The Tukey g-and-h distribution has a strong history of research and applications in environmental sciences and other fields such as financial modeling. It offers a good trade-off between a limited number of parameters and the ability to approximate a wide range of skewed and heavy-tailed probability density functions. Applications of Deep Neural Networks is also becoming more and more prevalent in the aforementioned fields. While the exact form of the Tukey g-and-h log-likelihood has no known closed-form solution, we show in this paper that we can still train a Deep Neural Network to predict a Tukey g-and-h distribution. While the standard approach in Tukey g-and-h random fields is to assume that a unique Tukey g-and-h is applied pointwise, here we allow the parameters of the transform to be features-dependent, and learn the corresponding mapping via standard neural network training techniques. This can naturally be extended to multi-modal Tukey g-and-h distributions with 4 output neurons for each mode and a softmax over $p$ neurons where $p$ is the number of modes. A natural direction for future work would be to address the remaining dependence in the target variable. For instance, looking at Figure 7b, it is clear that there remains some spatial dependence between residuals at neighbouring locations. Additional features — such as weather patterns for this application— may further account for this remaining dependence. However for applications such as spatial interpolation, it will be judicious to explicitly model and estimate a Gaussian covariance kernel on the residuals.

## References


Christopher M. Bishop. Mixture density networks. Technical report, Aston University, Birmingham, 1994. Unpublished.

Walter T Dado, Jillian M Deines, Rinkal Patel, Sang-Zi Liang, and David B Lobell. High-resolution soybean yield mapping across the US midwest using subfield harvester data. *Remote Sensing*, 12(21):3471, 2020.

Arthur P. Guillaumin and Laure Zanna. Stochastic-Deep Learning parameterization of ocean momentum forcing. *Journal of Advances in Modeling Earth Systems*, 13(9):e2021MS002534, 2021.

Kevin Höhlein, Michael Kern, Timothy Hewson, and Rüdiger Westermann. A comparative study of convolutional neural network models for wind field downscaling. *Meteorological Applications*, 27(6):e1961, 2020.

Toshichika Iizumi. Global dataset of historical yields v1.2 and v1.3 aligned version, 2019.

Intergovernmental Panel on Climate Change (IPCC). *Summary for Policymakers. In: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)].* IPCC, Geneva, Switzerland, 2023. doi: 10.59327/IPCC/AR6-9789291691647.001.

Jaehong Jeong, Yuan Yan, Stefano Castruccio, and Marc G Genton. A stochastic generator of global monthly wind energy with Tukey g-and-h autoregressive processes. *Statistica Sinica*, 29(3):1105–1126, 2019.

Martinez Jorge and Iglewicz Boris. Some properties of the Tukey g and h family of distributions. *Communications in Statistics - Theory and Methods*, 13(3):353–369, 1984.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye Teh. Bootstrapping neural processes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6606–6615. Curran Associates, Inc., 2020.

J. Hambuckers M. Bee and L. Trapin. Estimating large losses in insurance analytics and operational risk using the g-and-h distribution. *Quantitative Finance*, 21(7):1207–1221, 2021.

Natalie M Mahowald, Daniel S Ward, Scott C Doney, Peter G Hess, and James T Randerson. Are the impacts of land use on warming underestimated in climate policy? *Environmental Research Letters*, 12 (9):094016, 2017.

Daisuke Murakami, Gareth W. Peters, Tomoko Matsui, and Yoshiki Yamagata. Spatio-temporal analysis of urban heatwaves using Tukey g-and-h random field models. *IEEE Access*, 9:79869–79888, 2021.

Radford M. Neal. Bayesian Learning for Neural Networks. volume 118 of *Lecture Notes in Statistics*. Springer New York, NY, 2012.

Nathan H Ng, Rodney A Gabriel, Julian McAuley, Charles Elkan, and Zachary C Lipton. Predicting surgery duration with neural heteroscedastic regression. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pp. 100–111. PMLR, 18–19 Aug 2017.

Xiaoli Ren, Xiaoyong Li, Kaijun Ren, Junqiang Song, Zichen Xu, Kefeng Deng, and Xiang Wang. Deep Learning-based weather prediction: A survey. *Big Data Research*, 23:100178, 2021.

Guanyuan Shuai and Bruno Basso. Subfield maize yield prediction improves when in-season crop water deficit is included in remote sensing imagery-based models. *Remote Sensing of Environment*, 272:112938, 2022. doi: https://doi.org/10.1016/j.rse.2022.112938.

Laurens Sluijterman, Eric Cator, and Tom Heskes. Confident neural network regression with bootstrapped deep ensembles, 2023. arXiv 1711.07128.

James W. Taylor. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting*, 19(4):299–311, 2000.

Sherrie Wang, Stefania Di Tommaso, Jillian M Deines, and David B Lobell. Mapping twenty years of corn and soybean across the US midwest using the landsat archive. *Scientific Data*, 7(1):307, 2020.

Ganggang Xu and Marc G Genton. Tukey g-and-h random fields. *Journal of the American Statistical Association*, 112(519):1236–1249, 2017.

Qifa Xu, Kai Deng, Cuixia Jiang, Fang Sun, and Xue Huang. Composite quantile regression neural network with applications. *Expert Systems with Applications*, 76:129–139, 2017.

Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep Gaussian Process for crop yield prediction based on remote sensing data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

## A Additional figures from the simulated data experiments

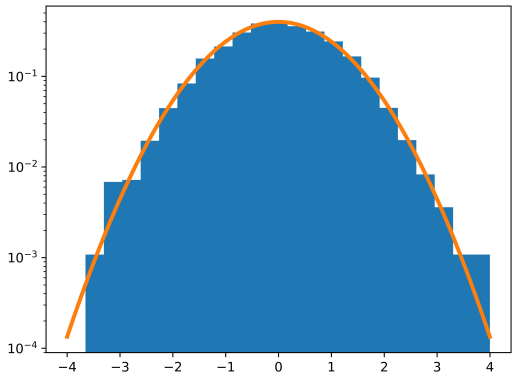



Figure 9: Log-density histogram of standardized residuals on validation data obtained from the Tukey g-and-h neural network trained on simulated t-distributed target variables versus the probability density function of a standard normal random variable.

## B Crop yield for maize

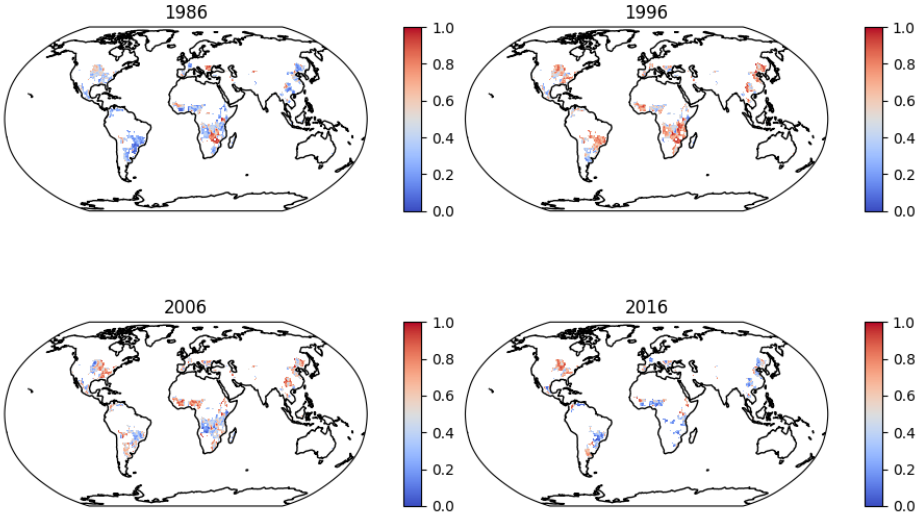



Figure 10: Map of uniform residuals for maize yield on test years

## C Crop yield for rice

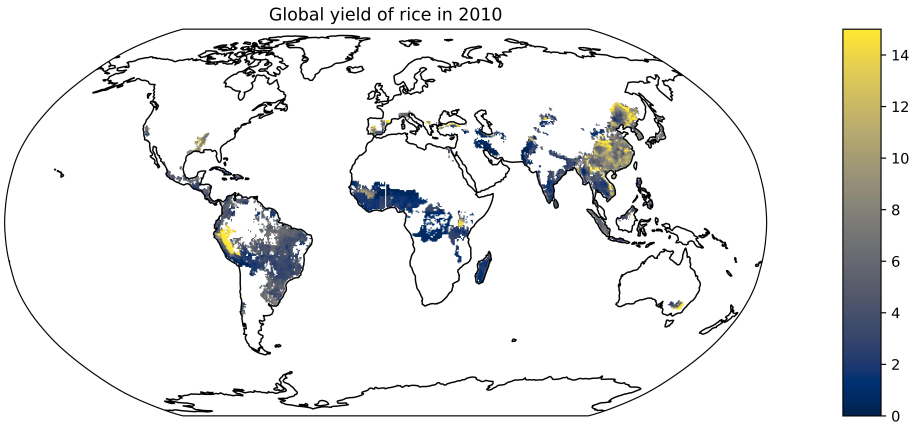



Figure 11: Gloabl yield of rice in ton per hectar in 2010 on a 0.5' spatial-resolution grid.

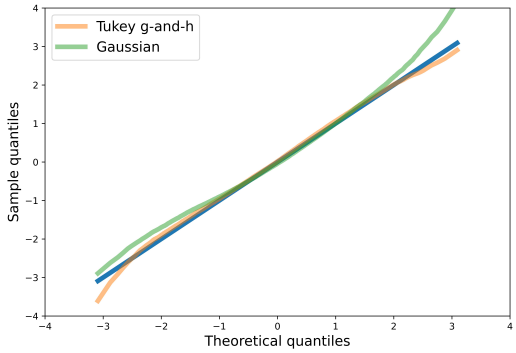



Figure 12: quantile-quantile plot of the rice residuals from the test data for methods 2 and 3 with respect to the standard normal distribution.

## D Crop yield for wheat

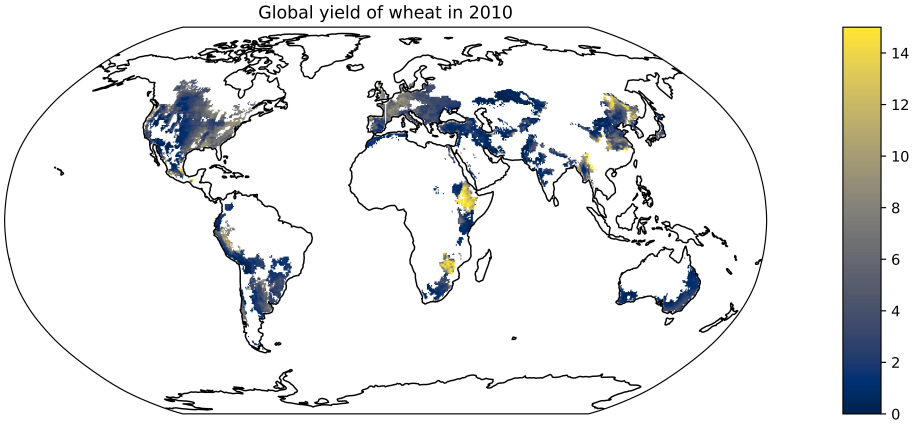



Figure 13: Gloabl yield of wheat in ton per hectar in 2010 on a 0.5' spatial-resolution grid.

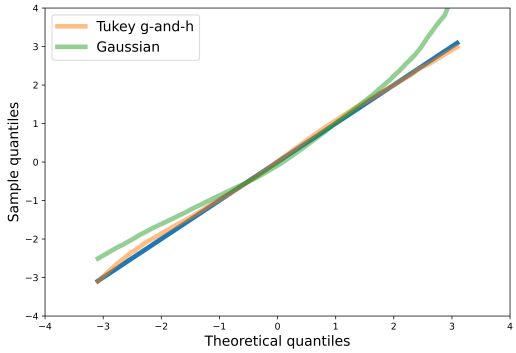



Figure 14: quantile-quantile plot of the wheat residuals from the test data for methods 2 and 3 with respect to the standard normal distribution.