# Supplementary Material

**Anonymous Author(s)**
Affiliation
Address
email

## 1 Mean Prediction and UQ of prediction

Given the main advantage of Deep Evidence Regression over other UQ aware deep learning methods like Bayesian NN, esembling etc, is due to existence of analytical solution for both predictions and unceratinty from NN output, without the need for sampling. Hence this section details the derivation of mean prediction and total uncertainty.

### 1.1 Mean Prediction

$$\text{We define the mean prediction as } E[Z|\alpha,\beta] \tag{1}$$
$$\text{where } Z = E[y_i] \tag{2}$$
$$\text{Now given } y_i \sim Weibull(k,\lambda) \tag{3}$$
$$Z = E[\lambda * \Gamma(1+\frac{1}{k})] = E(\lambda) * \Gamma(1+\frac{1}{k}) \quad \text{(k is known)} \tag{4}$$
$$E[\lambda] = \int_\lambda \lambda p(\lambda) d\lambda \tag{5}$$
$$\tag{6}$$

Hence to solve for mean prediction we need to find pdf $p(\lambda)$. Because we know $\theta = \lambda^k \sim \Gamma^{-1}(\alpha,\beta)$, we can use change of variable to find pdf of $\lambda$ [2].

$$p(\lambda|\alpha,\beta) = p_\theta(\lambda^k) * |\frac{d\lambda^k}{d\lambda}| \tag{7}$$
$$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\frac{1}{\lambda^k})^{\alpha+1} \exp{(-\frac{\beta}{\lambda^k})} * |k\lambda^{k-1}| \tag{8}$$
$$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\frac{1}{\lambda^k})^{\alpha+1} \exp{(-\frac{\beta}{\lambda^k})} * k\lambda^{k-1} \quad \text{(given } \lambda, k > 0) \tag{9}$$

Hence,

$$
\begin{aligned}
E[\lambda] &= \int_{\lambda} \lambda \frac{\beta^{\alpha}}{\Gamma(\alpha)} (\frac{1}{\lambda^k})^{\alpha+1} \exp{(-\frac{\beta}{\lambda^k})} * k\lambda^{k-1} d\lambda & (10)\\
&= \frac{k\beta^{\alpha}}{\Gamma(\alpha)} \int_{\lambda=0}^{\infty} \frac{1}{\lambda^{k\alpha+k-k}} \exp(\frac{\beta}{\lambda^k}) d\lambda & (11)\\
&\text{Substituting } t = 1/\lambda \text{, we get:} & (12)\\
dt &= -1/\lambda^2 d\lambda, & (13)\\
E[\lambda] &= \frac{k\beta^{\alpha}}{\Gamma(\alpha)} \int_{t=0}^{\infty} t^{k\alpha-2} \exp(-\beta t^k) dt & (14)\\
&\text{By table of integrals at [3]} & (15)\\
\int_0^{\infty} y^m e^{-by^k}, dy &= \frac{\Gamma\left(\frac{m+1}{k}\right)}{kb^{(m+1)/k}} & (16)\\
\implies E[\lambda] &= \frac{k\beta^{\alpha}}{\Gamma(\alpha)} * \Gamma(\frac{k\alpha-1}{k}) * \frac{1}{k} * \frac{1}{\beta^{\frac{k\alpha-1}{k}}} & (17)
\end{aligned}
$$

Hence we get the mean prediction as:

$$
\begin{aligned}
Z = E[y_i|\alpha,\beta] &= E(\lambda) * \Gamma(1+\frac{1}{k}) & (18)\\
= \frac{k\beta^{\alpha}}{\Gamma(\alpha)} * \Gamma(\frac{k\alpha-1}{k}) * \frac{1}{k} * \frac{1}{\beta^{\frac{k\alpha-1}{k}}} &* \Gamma(1+\frac{1}{k}) & (19)\\
&= \Gamma(1+\frac{1}{k}) \frac{1}{\Gamma(\alpha)} \Gamma(\alpha - \frac{1}{k}) * \beta^{1/k} & (20)
\end{aligned}
$$

## 1.2 UQ of Prediction

We quantify the total uncertainty as $Var(Z)$ with defined as above, i.e. $Z = E[y_i]$

$$
\begin{aligned}
Var(Z) &= Var(\lambda * \Gamma(1+\frac{1}{k})) & (21)\\
Var(Z|\alpha,\beta) &= Var(\lambda) * \Gamma^2(1+\frac{1}{k}) & (22)\\
&= (E[\lambda^2] - E[\lambda]) * \Gamma^2(1+\frac{1}{k}) & (23)\\
&\text{With } E(\lambda) \text{ defined as in 18, we only need } E(\lambda^2) & (24)\\
&\text{Similar to approach outlined in 1.1, we get:} & (25)\\
E[\lambda^2] &= \int_{\lambda} \lambda^2 p(\lambda) d\lambda & (26)\\
E[\lambda^2] &= \frac{k\beta^{\alpha}}{\Gamma(\alpha)} * \Gamma(\frac{k\alpha-2}{k}) * \frac{1}{k} * \frac{1}{\beta^{\frac{k\alpha-2}{k}}} & (27)\\
\implies E[Z^2] &= \Gamma^2(1+\frac{1}{k}) \frac{1}{\Gamma(\alpha)} \Gamma(\alpha - \frac{2}{k}) * \beta^{2/k} & (28)
\end{aligned}
$$

Similar to approach outlined in 1.1, we get:

$$
E[\lambda^2|\alpha,\beta] = \Gamma(\frac{k\alpha-2}{k}) \frac{\beta^{2/k}}{\Gamma(\alpha)} \tag{29}
$$

Hence we can write

$$Var(Z) = \Gamma^2(1 + \frac{1}{k}) * \left[\Gamma(\frac{k\alpha - 2}{k})\frac{\beta^{2/k}}{\Gamma(\alpha)} - \left(\Gamma(\frac{k\alpha - 1}{k})\frac{\beta^{1/k}}{\Gamma(\alpha)}\right)^2\right] \tag{30}$$

or

$$Var(Z) \propto \frac{\beta^{2/k}}{\Gamma(\alpha)^2}\left[\Gamma(\alpha)\Gamma(\frac{k\alpha - 2}{k}) - \Gamma^2(\frac{k\alpha - 1}{k})\right]$$

### 1.3 Validation of proofs

Here we generate a target variable following a Weibull distribution. The target variable is generated as:

$y_i = x_i^2 + \epsilon, \epsilon \sim Weibull(k = 1.2, \lambda = 0.2)$

The train set is comprised of $x \in [0, 3]$ while test set is $x \in [0, 4]$.

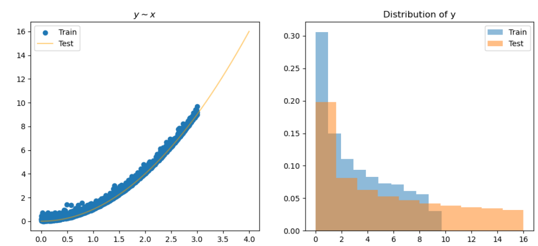


Figure 1: y vs x and Distribution of y(right) for synthetic data

Since our approach assumes known $k$, k is estimated from the training set.

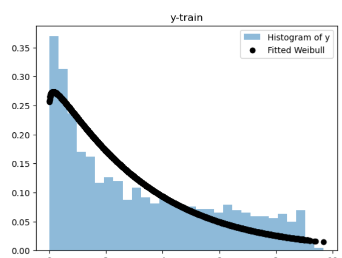


Figure 2: Weibull Fit on Train data

To confirm that analytical calculations outlined above, we have also created the mean prediction after sampling from the NN outputs. Firstly we sample $\theta$ from $\Gamma(\alpha, \beta)$. $\lambda$ is then calculated as the k-th root of $\theta$ or $\lambda = \theta^{1/k}$. Finally, the response variable $y_i$ can be sampled as $Weibull(\lambda, k)$. The consistency between the results from sampling and analytical calculations, supports the analytical calculations in 18.

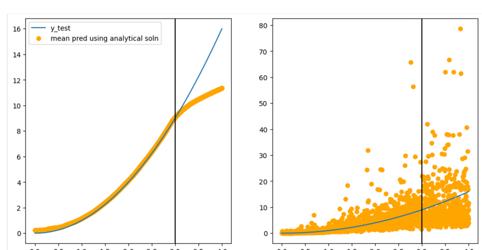


Figure 3: Mean prediction using analytical equation(left) vs from sampling (right)

## 2 Experiment Code and Results

The code and corresponding dataset is shared alongwith

### 2.1 Experiments on synthetic data

In the experiment, both approaches utilized an identical neural network architecture consisting of five hidden layers, with each layer comprising 200 neurons. To enhance the model's performance, hyperparameter optimization was conducted on the regularization cost, denoted as 'c.' This optimization involved varying the value of 'c' within the logarithmic space ranging from 0.000001 to 0.1. For each lambda value, the best model was selected based on the value of 'c' that minimized the training loss, adhering to the outlined procedure. This approach allowed for fine-tuning the regularization parameter and ensuring that the chosen models were optimized for the given experiment.

Qualitatively we can see that the benchmark version either captures no uncertainty in prediction or amplifies it way too much for different c values.

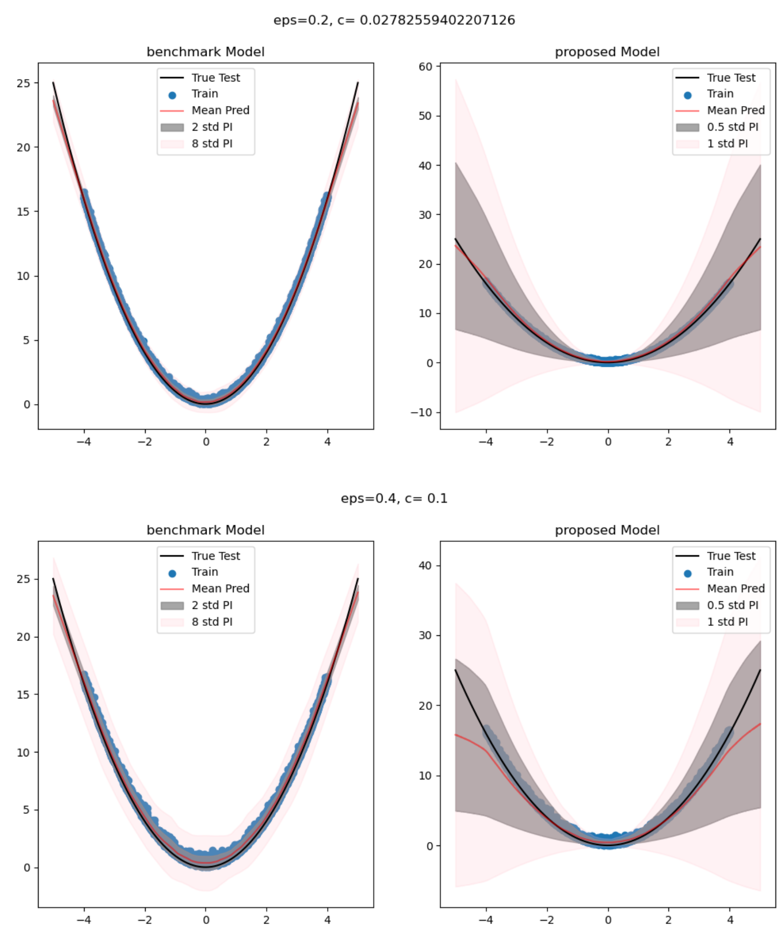


Figure 4: Deep evidence regression (left) vs Weibull evidence Regression (right). We see that uncertainty is much better captured by proposed version.

### 2.2 Experiments on Recovery data

The dataset under consideration pertains to peer to peer mortgage lending data during the period of 2007 to 2014 sourced from Kaggle [1]. However, the data does not include the loss given default values. Instead, the recovery rate has been used as a proxy, which is calculated as the ratio of recoveries made to the origination amount. The dataset contains approximately 46 variables denoted as 'x,' which include features such as the time since the loan was issued, debt-to-income ratio (DTI), joint applicant status, and delinquency status, among others. In total, the dataset comprises around 23,000 rows.

The model architecture for Benchmark model is as follows:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_100 (Dense)           (None, 1)                 46

 dense_101 (Dense)           (None, 350)               700

 dense_102 (Dense)           (None, 300)               105300

 dense_103 (Dense)           (None, 300)               90300

 dense_104 (Dense)           (None, 250)               75250

 dense_105 (Dense)           (None, 250)               62750

 dense_106 (Dense)           (None, 200)               50200

 dense_107 (Dense)           (None, 200)               40200

 dense_108 (Dense)           (None, 200)               40200

 dense_normal_gamma_5 (Dense  (None, 4)                804
 NormalGamma)

=================================================================
Total params: 465,750
Trainable params: 465,750
Non-trainable params: 0
_________________________________________________________________
```

The model architecture for Proposed model is as follows:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_110 (Dense)           (None, 1)                 46

 dense_111 (Dense)           (None, 350)               700

 dense_112 (Dense)           (None, 300)               105300

 dense_113 (Dense)           (None, 300)               90300

 dense_114 (Dense)           (None, 250)               75250

 dense_115 (Dense)           (None, 250)               62750

 dense_116 (Dense)           (None, 200)               50200

 dense_117 (Dense)           (None, 200)               40200

 dense_118 (Dense)           (None, 200)               40200

```

```
 dense_weibull_gamma_5 (Dens  (None, 2)               402
 eWeibullGamma)

=================================================================
Total params: 465,348
Trainable params: 465,348
Non-trainable params: 0
_________________________________________________________________
```

Hyperparameter optimization was conducted on the regularization cost, denoted as 'c.' This optimization involved varying the value of 'c' within the linear space ranging from 0.4 to 1.2 for proposed version and 0.04 to 0.12 for benchmark version. For each lambda value, the best model was selected based on the value of 'c' that minimized the training loss, adhering to the outlined procedure. This approach allowed for fine-tuning the regularization parameter and ensuring that the chosen models were optimized for the given experiment. Finally NN is trained for 4 times with the best c value.

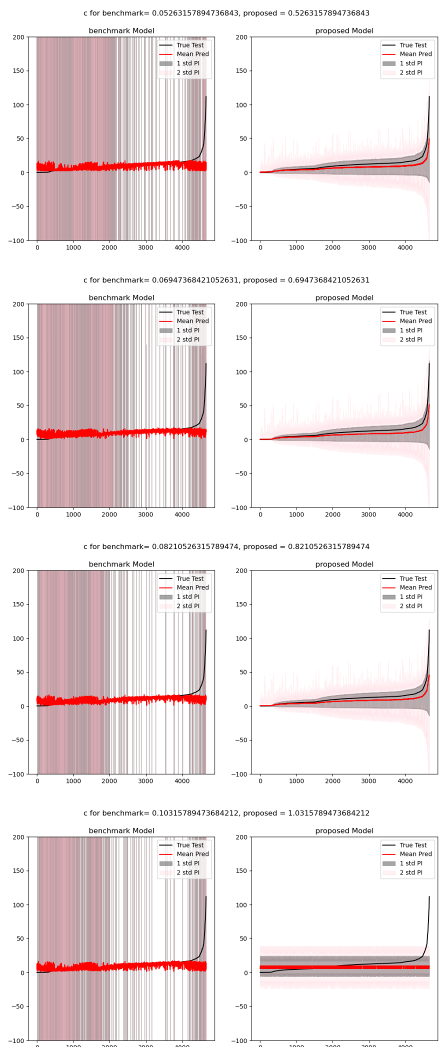


Figure 5: Deep evidence regression (left) vs Weibull evidence Regression (right). We see that uncertainty is much better captured by proposed version.

# References

[1] Shawn Sun. Loan data for credit risk modeling. https://www.kaggle.com/datasets/shawnysun/loan-data-for-credit-risk-modeling, 2021.

[2] Wikipedia contributors. Change of variables — Wikipedia, the free encyclopedia, 2023. [Online; accessed 1-May-2023].

[3] Wikipedia contributors. List of integrals of exponential functions — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_integrals_of_exponential_functions&oldid=1142840420, 2023. [Online; accessed 1-May-2023].