Quantifying Uncertainty in Neural Networks through Residuals

Dalavai Udbhav Mallanna
udbhav@students.iisertirupati.ac.in
IISER Bhopal
Bhopal, India

Rini Smita Thakur
rinithakur@iiserb.ac.in
IISER Bhopal
Bhopal, India

Rajeev Ranjan Dwivedi
rajeev22@iiserb.ac.in
IISER Bhopal
Bhopal, India

Vinod K Kurmi
vinodkk@iiserb.ac.in
IISER Bhopal
Bhopal, India

ABSTRACT

Regression models are of fundamental importance in explicitly explaining the response variable in terms of covariates. However, point predictions of these models limit them from many real world applications. Heteroscedasticity is common in most real-world scenarios and is hard to model due to its randomness. The Gaussian process generally captures epistemic (model) uncertainty but fails to capture heteroscedastic aleatoric uncertainty. The framework of HetGP inherently captures both epistemic and aleatoric by placing independent GP's priors on both mean function and error term. We propose the posthoc HetGP on the residuals of the trained deterministic neural network to obtain both epistemic and aleatoric uncertainty. The advantage of posthoc HetGP on residuals is that it can be extended to any type of model, since the model is assumed to be black-box that gives point predictions. We demonstrate our approach through simulation studies and UCI regression datasets. The code is available at https://visdomlab.github.io/HetGP/.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Gaussian processes**.

KEYWORDS

Heteroscedasticity, Uncertainty, Gaussian Processes, Regression, Neural Networks

ACM Reference Format:
Dalavai Udbhav Mallanna, Rini Smita Thakur, Rajeev Ranjan Dwivedi, and Vinod K Kurmi. 2024. Quantifying Uncertainty in Neural Networks through Residuals. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679983

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679983

1 INTRODUCTION

Neural networks (NN) finds application in the broad range of the real-world regression and classification problems mapping high dimensional input array of data to the requisite output. NN's maximizes the likelihood parameter which does not give any information about the confidence of the prediction. There is an underlying assumption about the confidence of these mappings, which does not always hold true [22]. NN prediction fails due to biased dataset, model inference errors, domain discrepancy between training and test datasets, etc. Thus, uncertainty quantification (UQ) and prediction are crucial to real-world decision making.

There has been rich literature on quantifying predictive uncertainty in NN ([15], [6], [28], [34]), which can be further decomposed to *aleatoric* and *epistemic* uncertainty. Aleatoric, or data uncertainty, reflects intrinsic stochasticity in data, while epistemic, or model uncertainty, arises from limitations in model itself. Epistemic can be reduced by collecting more data and training a more complex model [1]. Aleatoric is further classified as homoscedastic, which is constant for all inputs and heteroscedastic, which varies with respect to inputs.

Bayesian NN, Gaussian Process (GP), and deterministic UQ methods exist. Bayesian methods are computationally intensive and require network architecture changes, making them difficult to implement. Moreover, sampling requirement in Bayesian methods hinders real-time ability on edge devices. Therefore, to overcome the disadvantages of Bayesian modeling, UQ progressed towards nonparametric and deterministic networks ([2], [34], [21]). Evidential Deep Learning (EDL) is a popular deterministic approach that integrates deep learning and Dempster-Shafer theory ([29]) to quantify predictive uncertainty, resulting in impressive achievements and closed-form expressions of both epistemic and aleatoric uncertainty [28].

On the other hand, GP's ([25]) are highly flexible Bayesian nonparametric models that inherently quantify uncertainty. The standard GP quantifies the epistemic uncertainty by incorporating the distributions of all functions satisfying constraints of the covariance function and data. However, exact GP inference suffers from $O(n^3)$ complexity with number of data points due to the inversion of covariance matrix. There has been significant research on combining deep NNs with GP's ([3], [33], [11], [31]) leveraging expressivity of deep NNs and UQ capabilities of GP's. In particular, ([13]) proposed using a NN to model the mean function of a GP. [7] introduced a different approach that combines NNs and GPs. They proposed

defining conditional distributions over functions given data with a NN parameterising data dependence. However, the above GP-NN combinations change network architecture and homoscedasticity assumption significantly.

To overcome the disadvantage of architectural changes and homoscedasticity assumption of the standard GP, we propose to use Heteroscedastic Gaussian Process (HetGP) over the residuals of the trained network to quantify both aleotoric and epistemic uncertainty as given in Fig (1). Input noise dependent GP regression has independent GP prior on the noise term [8]. To overcome the computational complexity and the intractable posterior for GPs many approximate inference methods are proposed including MCMC ([8]), Variational Bayes ([18]), Laplace Approximation ([35]), Expectation Propogation ([10]), here we adopt the variational sparse approximation of GP through inducing points ([9]), which speeds up the computation by choosing m inducing points over the covariate space. The proposed framework works by taking the residuals of a pretrained network and fitting a HetGP on the residuals during the training time. During the testing time, the mean predictions are given by NN and predictive uncertainty by HetGP simultaneously. Section 2 reviews previous research and our contributions. Sections 3 and 4 describe the methodology of HetGP on residuals using Sparse Variational GP [9]. Section 5 analyzes HetGP on residuals, with results in Section 6. The article concludes in Section 7.

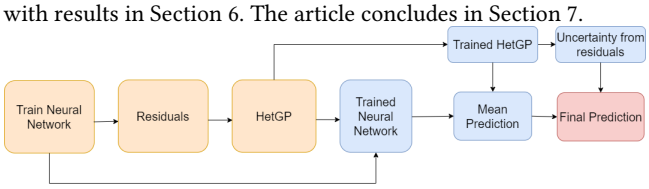

Figure 1: Training and Prediction phases of the proposed method

2 RELATED WORK

Post-hoc UQ in NN's is well studied in literature, ([16]) is the straightforward approach to turn a NN to GP through Laplace approximation, ([32]) used Metropolis Hastings MCMC sampling for uncertainty estimation of trained deterministic networks and is highly computationally complex. Recently, ([20]) have done a thorough analysis on the dropout injection method, which uses dropout layers only during the inference time to obtain epistemic uncertainty prediction. But post-hoc uncertainty through residuals is relatively less explored in the literature, ([24]) is the first one to use a GP with a modified kernel that includes both input and output of the NN to model the residuals of a trained NN to obtain predictive uncertainties. However, their work does not explicitly deal with heteroscedastic data, and using an additive kernel in their framework might account for heteroscedasticity.

Traditional methods ([4]) dealt heteroscedasticity by first fitting the mean function and then fitting the standard deviation from log-residuals. However, the resulting objective function is only convex with respect to a single parameter at a time, not jointly convex in both. To address this for GP's, ([19]) proposed the objective function in terms of natural parametrization.

The de facto standard of obtaining aleatoric uncertainty with NN's involves assuming a specific distribution for the regression targets ([23], [17], [15]). A NN then predicts the parameters of this distribution, typically the input-dependent mean and variance for a heteroscedastic Gaussian distribution. These network parameters are subsequently learned through maximum likelihood estimation (MLE). Several approaches are proposed, including using a separate NN architecture to model the variance ([5]). However, ([27]) pointed out on the performance of these models on their mean function, which can lead to sub par mean function estimates.
The most recent approaches to account for heteroscedasticity in NN include ([30], [12]), where ([12]) inspired by ([19]) adopts the natural parametrization of the Gaussian distribution to obtain modified loss function for the aleatoric uncertainty, Laplace approximation was employed to get the epistemic uncertainty.

Contributions - Our main contributions include enhancing a deterministic-trained NN to account for epistemic and aleatoric uncertainty through residuals without making any modifications to the network architecture or loss functions. While our main focus is on capturing heteroscedasticity, this framework also reduces the error in the NN prediction and make the model well calibrated.

3 PROBLEM STATEMENT

If we have a training dataset $D = \{x_i, y_i\}_{i=1}^{n}$ and NN with pre-trained weights, employed for point prediction $(\hat{\mu}_i)$ given x_i. Our objective is to tackle the issue of heteroscedastic aleatoric uncertainty in NN predictions and enhance the accuracy of the point predictions made by the NN.

4 FRAMEWORK OVERVIEW

The problem is addressed through the residuals of the NN predictions $(\hat{\mu})$ and the observed outcomes y using a HetGP. The HetGP configuration typically employs two GP's, one for representing the latent function and another for estimating the noise that varies with the input. The fusion of the two GP's will produce a joint posterior distribution that encompasses both the latent function and the noise dependent on the input. This distribution is non-Gaussian and cannot be easily solved analytically.

The framework of HetGP assuming the response variable generated by normal distribution is given by,

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2(x_i)) \tag{1}$$

Here, the functions $f(x)$ and $\log(\sigma^2(x))$ are the independent GP's. We get a standard GP when the noise variance term is independent of data. In general heteroscedasticity arises from the non-constant variance of the error terms. Here, we have variance of error as a function of data. During the training stage, the discrepancies (residuals (r_i)) are computed as the differences between observed outcomes (y_i) and predictions $(\hat{\mu}_i)$ generated by the NN on the training set calculated as $r_i = y_i - \hat{\mu}_i$. Now, we have the residuals r_i and can be further broken down into,

$$r_i = y_i - f_i + f_i - \hat{\mu}_i = \epsilon_i + (f_i - \hat{\mu}_i) \tag{2}$$

Where, ϵ_i is the observation error or noise and the $(f_i - \hat{\mu}_i)$ is the modelling error. The observation error is random, and the modelling error is non-random. This makes our residuals r_i a random variable with gaussian distribution with mean $\mu_{r_i} = (f_i - \hat{\mu}_i)$ and variance $(\sigma^2_{r_i} = g_i = \text{var}(\epsilon(x_i))$

$$r_i \sim \mathcal{N}(\mu_{r_i}, \sigma^2_{r_i}) \tag{3}$$

Since, we assume heteroscedasticity in our data the variance of the residuals is a function of data. To account for this, we place a

heteroscedastic GP prior over the residuals to independently model both the mean function and the log-noise. Let $\mu_{r_i} = m(x_i)$ is the mean function of the residuals and $\text{var}(\epsilon(x_i)) = \sigma^2_{r_i}$. So, we can place independent GP priors over $m(x_i)$ and $\sigma^2_{r_i}$ and to ensure positive variance, we use log link function for the second GP.

We have $\mu_{r_i} \sim GP(0, K_1)$ and $\log(\sigma^2_{r_i}) \sim GP(0, K_2)$, where K_1 and K_2 are the kernel functions of the respective independent GPs. Assume that $z_i = \log(\sigma^2_{r_i})$. and thus the HetGP uses two GPs: the z-process, which learns the input-dependent noise level, and the μ-process, which recovers the unknown mean function, which implies that $r_i \sim \mathcal{N}(\mu_{r_i}, e^{z_i})$. Over the test output, the resulting posterior predictive is given by,

$$p(r_i^*|x^*, \theta_\mu, \theta_z, D) = \int p(r_i^*|x^*, \theta_\mu, z, z^*, D)p(z, z^*|x^*, \theta_z, D)dzdz^*$$

Where $z = (z_1, \ldots, z_n)^T$ are the log noise variances at training inputs X and z^* is the log noise variance at the test input x^*. Here, the full posterior $p(z, z^*|x^*, \theta_z, D)$ of the noises is analytically intractable and we accomodate the variational sparse approximation to obtain the approximate full posterior. Since, we have two independent GP's for mean and noise, with separate link functions, the framework is similar to chained GP ([26]). For simplicity, let us assume $\mu_i = f_i, z_i = g_i$. Then the variational approximation of the posterior is given by,

$$p(f, g, u_f, u_g|r) \approx p(f|u_f)p(g|u_g)q(u_f)q(u_g) \tag{4}$$

Where, $u_f = f(t), u_g = g(t)$ are the inducing points at locations $T = \{t_i\}_{i=1}^m$. We can write the lower bound for the log-marginal from the Jensen's inequality, which is given by,

$$\log p(r) \geq \int q(f)q(g) \log p(r|f, g)dfdg - \text{KL}(q(u_f)||p(u_f)) - \text{KL}(q(u_g)||p(u_g)) \tag{5}$$

Where the first term is the variational expectation, second and third terms are the prior KL (Kullback–Leibler) terms which acts as a regularizers for the objective function. Here maximizing this log-marginal is equivalent to minimizing the Kullback–Leibler-divergence between true posterior and the approximate posterior.

Now, for a new independent test points, the predictive distribution for each data pair is given by,

$$p(r_i^*|r_i, x_i) = \int p(r_i^*|f_i^*, g_i^*)q(f_i^*)q(g_i^*)df_i^*dg_i^* \tag{6}$$

where $q(f^*)$ and $q(g^*)$ are the variational distributions of the respective GP's posteriors.

So, the distribution of the residuals at the new test point x_* is given by $\hat{r}_*|X, \hat{y}, r, x_* \sim \mathcal{N}(\bar{\hat{r}}_*, \text{var}(\hat{r_*}))$. We now notice that the NN predictions can be calibrated through the mean prediction of the residuals, so that the final calibrated prediction with uncertainty information is given by,

$$\hat{y}'_* \sim \mathcal{N}(\hat{y_*} + \bar{\hat{r}}_*, \text{var}(\hat{r}_*)) \tag{7}$$

Here, $(\text{var}(\hat{r_*}))$ is the variance of the error term and accounts for the heteroscedasticity of the new input points. It is also evident that the NN is deterministic implying that $\text{var}(\hat{y}_*) = 0$. This gives us that the variance of the final prediction which account for epistemic uncertainty is only given by the epistemic uncertainty of HetGP. However, without changing the architecture or training set, this method additionally provides the means to calibrate predictions of NN alongside meaningful uncertainty information.

5 ANALYSIS OF HETGP ON RESIDUALS

This section shows theoretically how applying a HetGP to trained NN residuals can capture heteroscedasticity and provide uncertainty information. This method reduces prediction errors, improving NN calibration. Due to their expressive power, NNs are good at interpreting complex data patterns, while GPs handle uncertainty well. Removing variable noise and simplifying the complex structure makes fitting a HetGP to the remaining NN errors easier. This approach takes advantage of NN's predictive capabilities while also providing useful uncertainty estimates.

As discussed in ([27]) the main reasons for under performance of a NN model when using the common NLL loss for heteroscedastic aleatoric noise is due to initial badly-fit regions receiving increasingly less weightage in the loss which results in premature convergence. Here, we try to avoid this problem by training the NN with homoscedastic assumption and capturing heteroscedasticity through residuals. For example, let us consider the case where our data generating process is defined by,

$$y_i = f(x_i) + \epsilon(x_i) = f_1(x_i) + f_2(x_i) + \epsilon(x_i)$$

Here, $f(x_i)$ represents the true mean function, while $\epsilon(x_i)$ represents the associated heteroscedasticity. The true function f can be decomposed into two parts: f_1, which is well captured by the mean function of HetGP, and f_2, which is considered as noise. This is particularly evident in situations involving complex mean and noise functions, such as image regression. For example, a pretrained NN that has already learned a certain level of complexity can be utilised to its advantage. The mean function of HetGP may capture this complexity, but it reduces generalisation. The unsatisfactory result may have been due to difficult kernel selection and high-dimensional data. By accurately estimating complexity variance as noise, the HetGP optimiser avoids mismodelling complexity. Residual components are determined by,

$$y_i - f_{NN}(x_i) = r(x_i) + \epsilon(x_i) = r_{f_1}(x_i) + r_{f_2}(x_i) + \epsilon(x_i)$$

Where $f_{NN}(x_i)$ is the trained NN and in this case, it is easier for the mean function of HetGP to learn the residuals than the true function (f) itself and the remaining noisy part of the residuals is easily modelled by noise function of HetGP.

From Theorem 5.1 in ([24]), we can see that fitting HetGP on residuals is useful only when $E[f_2^2(x)] - E[r^2_{f_1}(x)] > 0$, which says that fitting on the residuals is helpful when NN has already captured some complex structure. This statement and theorem is clearly justified through the simulation study (3, 4). This emphasises the need to train NNs without overfitting.

6 NUMERICAL EXAMPLES

6.1 Simulation study

We train a HetGP NN, where the network outputs both mean and standard deviation as a function of data. We then compute the residuals corresponding the true response. Now, this looks like the case of a heteroscedastic regression, where the variance of the error term is a function of data and the residuals (r_i) will be a function of data. Fitting a HetGP on residuals is equivalent to fitting a heteroscedastic regression on the original formulation. First, let us consider a data generating process with simple mean function

	autompg	concrete	energy	gas	machine	parkinsons	protein	wine(r)	yatch
RMSE (↓)									
NN	0.067 ± 0.003	0.056 ± 0.003	0.014 ± 0.003	0.037 ± 0.026	0.089 ± 0.004	0.032 ± 0.015	0.142 ± 0.001	0.075 ± 0.004	0.020 ± 0.009
GP (r)	0.066 ± 0.001	0.053 ± 0.002	0.010 ± 0.001	0.019 ± 0.006	0.089 ± 0.003	0.028 ± 0.011	0.141 ± 0.001	0.072 ± 0.003	0.015 ± 0.003
EDL	0.070 ± 0.003	0.064 ± 0.009	0.025 ± 0.011	0.027 ± 0.008	0.089 ± 0.004	**0.021** ± 0.002	0.148 ± 0.001	**0.069** ± 0.003	0.044 ± 0.018
HetGP (r)	**0.065** ± 0.001	**0.053** ± 0.002	**0.009** ± 7e-4	**0.019** ± 0.005	**0.088** ± 0.003	0.027 ± 0.013	**0.140** ± 0.001	0.072 ± 0.002	**0.016** ± 0.005
ECE (↓)									
NN	0.022 ± 0.008	0.017 ± 0.006	0.007 ± 0.003	0.023 ± 0.019	0.056 ± 0.002	0.007 ± 0.004	0.019 ± 0.004	0.022 ± 0.005	0.009 ± 0.006
GP (r)	0.017 ± 0.002	**0.013** ± 0.001	**0.003** ± 1e-4	**0.004** ± 0.001	**0.055** ± 0.002	0.004 ± 0.002	**0.014** ± 0.001	0.020 ± 0.002	**0.004** ± 0.001
EDL	0.025 ± 0.006	0.027 ± 0.013	0.017 ± 0.011	0.014 ± 0.010	0.058 ± 0.003	**0.004** ± 0.001	0.022 ± 0.003	0.026 ± 0.008	0.026 ± 0.018
HetGP (r)	**0.016** ± 0.001	**0.013** ± 0.001	**0.003** ± 1e-4	0.005 ± 0.003	0.056 ± 0.002	0.004 ± 0.002	**0.014** ± 0.001	**0.019** ± 0.003	0.004 ± 0.002
NLPD (↓)									
GP (r)	1.294 ± 0.028	1.429 ± 0.051	2.853 ± 0.288	1.815 ± 0.548	0.931 ± 0.074	2.013 ± 0.456	0.516 ± 0.011	**0.824** ± 0.201	2.232 ± 0.821
EDL	1.242 ± 0.056	1.201 ± 0.219	2.033 ± 0.490	1.931 ± 0.798	**0.362** ± 0.284	2.583 ± 0.111	0.428 ± 0.044	1.162 ± 0.147	1.545 ± 0.427
HetGP (r)	**0.984** ± 0.013	**0.876** ± 0.031	**1.098** ± 0.027	**0.890** ± 0.051	0.866 ± 0.013	**1.165** ± 0.272	**0.242** ± 0.009	1.138 ± 0.036	**0.932** ± 0.176

Table 1: Performance metrics (RMSE, ECE, NLPD) for different models across various UCI regression datasets.

and noise $y = 0.5x^2 + 0.25x^3 + \mathcal{N}\left(0, (2x\sin(x))^2\right)$. Figure (2) shows the comparison of aleatoric of NN's with aleatoric uncertainty obtained from fitting HetGP and evidential model. The portion of 95% coverage is obtained as 95.3% for NN that outputs mean and standard deviation, 95.7% for HetGP on residuals, 97% for evidential regression. This plot shows that even though the mean function is almost linear and we can capture the heteroscedasticity as function of data through residuals. It is clear that the evidential model covers more number of points than it need to in the 95% confidence plot.

The idea is to take any trained NN and use HetGP to calculate predictive uncertainty with residuals as the response variable. The plot shows smooth uncertainty bounds from squared exponential kernel versus sharp relu curves.

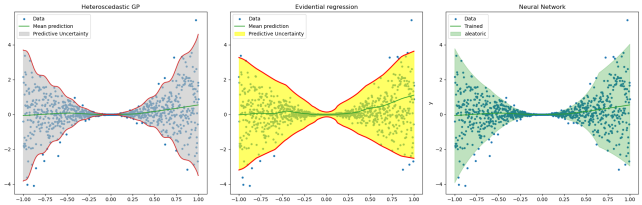

Figure 2: Comparison of Aleatoric uncertainty in HetGP on residuals, NN and Evidential model

Now, we consider an example with more complex mean and noise by generating 1000 samples from $y = \text{sinc}(x) + \epsilon \cdot S_X$, where $x \in (-10, 10)$, $\epsilon \sim \mathcal{N}(0, 1)$ and $S_X = 0.05 + \left(1 + \frac{1}{1+e^{-0.2x}}\right)(1 + \sin(2x)) \cdot 0.2$. The plots (Figure 2, Figure 3) clearly shows the capability of the proposed method in obtaining heteroscedastic aleatoric uncertainty even on highly noisy data. From the plot of residuals (Fig: 2) we can clearly see that the mean function is almost constant and the HetGP has to focus only on capturing the heteroscedasticity. If suppose, we try to fit the original data directly with HetGP, then either the resulting mean or noise function is underfit. So, NN acts a preprocessing method to capture the complexities in the mean function and leaving out the noise to HetGP to model through residuals.

6.2 UCI regression data

Here, we consider the publicly available UCI regression datasets, where all the 9 datasets are real world regression problems ([14]) and

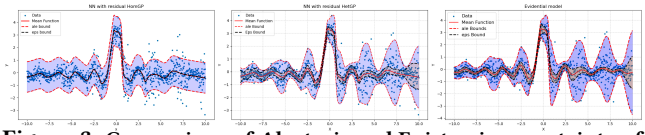

Figure 3: Comparison of Aleatoric and Epistemic uncertainty of different models

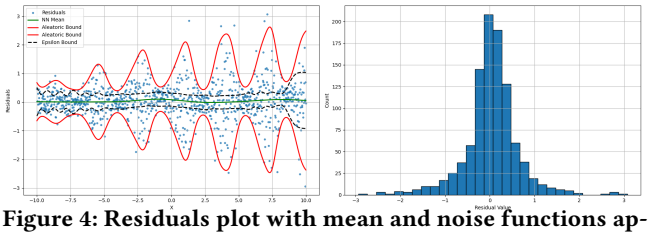

Figure 4: Residuals plot with mean and noise functions approximated by HetGP and histogram of residuals

each dataset is normalized and split into train and test where the NN model is trained on train data and the HetGP is trained on residuals with NN fitted on train data. Then, the test data is used to predict on both NN and HetGP. Evidential model is directly trained on train and tested on test data. We also considered the homoscedastic GP on residuals for comparison with the performance of HetGP. For all the datasets, 50 inducing points are chosen with squared exponential kernels. NaturalGradient and Adam optimise hyperparameters with 0.1 and 0.01 step sizes.

Root Mean Square Error (RMSE) is used here to measure the error between model predictions and the true outcomes; calibration of the model with Expected Calibration Error (ECE); quality of the uncertainty estimates by Negative Log Predictive Density (NLPD), all the experiments are run for 10 times by shuffling the dataset before train-test split each time and the mean and standard deviation of the metrics are reported. While the primary goal of HetGP on residuals is to improve pretrained NN rather than build a new model for prediction from start, statistical tests show that this method outperforms NN on all datasets, with 6 out of 9 datasets showing significant improvements over other approaches. Table (1) displays RMSE, ECE, and NLPD metrics for all models, with GP (r) and HetGP(r) fitting residuals.

7 DISCUSSION AND CONCLUSION

Here, we presented a novel approach on obtaining heteroscedastic aleatoric and epistemic uncertainty estimates from a trained deterministic NN, our proposed method need no modifications to the network architecture and works post-hoc on residuals to capture the inherent heteroscedasticity in the data. Our approach involves variational sparse GPs and can easily scale to large datasets. We have also shown that the proposed method is on par with the state of art deterministic uncertainty networks. As a part of future work, we aim to extend this framework to classification tasks and assess the reliability of epistemic uncertainty.

REFERENCES

[1] Moloud Abdar et al. 2021. A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Information Fusion*, 76, 243–297. DOI: https://doi.org/10.1016/j.inffus.2021.05.008.

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. 2020. Deep evidential regression. *Advances in neural information processing systems*, 33, 14927–14937.

[3] John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. 2017. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*.

[4] Gavin C. Cawley, Nicola L.C. Talbot, Robert J. Foxall, Stephen R. Dorling, and Danilo P. Mandic. 2004. Heteroscedastic kernel ridge regression. *Neurocomputing*, 57, 105–124. DOI: https://doi.org/10.1016/j.neucom.2004.01.005.

[5] Nicki S Detlefsen, Martin Jørgensen, and Søren Hauberg. 2019. Reliable training and estimation of variance networks. *arXiv preprint arXiv:1906.03260*.

[6] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[7] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. 2018. Conditional neural processes. In *International conference on machine learning*. PMLR, 1704–1713.

[8] Paul Goldberg, Christopher Williams, and Christopher Bishop. 1997. Regression with input-dependent noise: a gaussian process treatment. In *Advances in Neural Information Processing Systems*. M. Jordan, M. Kearns, and S. Solla, (Eds.) Vol. 10. MIT Press. https://proceedings.neurips.cc/paper_files/paper/1997/file/afe434653a898da20044041262b3ac74-Paper.pdf.

[9] James Hensman, Alexander Matthews, and Zoubin Ghahramani. 2015. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*. PMLR, 351–360.

[10] Daniel Hernández-Lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H Lampert, and Novi Quadrianto. 2014. Mind the nuisance: gaussian process classification using privileged noise. *Advances in Neural Information Processing Systems*, 27.

[11] Wenbing Huang, Deli Zhao, Fuchun Sun, Huaping Liu, and Edward Chang. 2015. Scalable gaussian process regression using deep neural networks. In *Twenty-fourth international joint conference on artificial intelligence*. Citeseer.

[12] Alexander Immer, Emanuele Palumbo, Alexander Marx, and Julia E Vogt. 2023. Effective bayesian heteroscedastic regression with deep neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=A6EquH0enk.

[13] Tomoharu Iwata and Zoubin Ghahramani. 2017. Improving output uncertainty estimation and generalization in deep learning via neural network gaussian processes. *arXiv preprint arXiv:1707.05922*.

[14] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. 2017. The uci machine learning repository. https://archive.ics.uci.edu. (2017).

[15] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.

[16] Mohammad Emtiyaz E Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. 2019. Approximate inference turns deep networks into gaussian processes. *Advances in neural information processing systems*, 32.

[17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

[18] Miguel Lázaro-Gredilla and Michalis K Titsias. 2011. Variational heteroscedastic gaussian process regression. In *ICML*, 841–848.

[19] Quoc V. Le, Alex J. Smola, and Stéphane Canu. 2005. Heteroscedastic gaussian process regression. In (ICML '05). Association for Computing Machinery, Bonn, Germany, 489–496. ISBN: 1595931805. DOI: 10.1145/1102351.1102413.

[20] Emanuele Ledda, Giorgio Fumera, and Fabio Roli. 2023. Dropout injection at test time for post hoc uncertainty quantification in neural networks. *Information Sciences*, 645, 119356.

[21] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33, 7498–7512.

[22] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: de-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33, 20673–20684.

[23] D.A. Nix and A.S. Weigend. 1994. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 1, 55–60 vol.1. DOI: 10.1109/ICNN.1994.374138.

[24] Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. 2020. Quantifying point-prediction uncertainty in neural networks via residual estimation with an i/o kernel. In *International Conference on Learning Representations*.

[25] 2004. *Gaussian processes in machine learning*. Springer Berlin Heidelberg. ISBN: 978-3-540-28650-9. DOI: 10.1007/978-3-540-28650-9_4.

[26] Alan D Saul, James Hensman, Aki Vehtari, and Neil D Lawrence. 2016. Chained gaussian processes. In *Artificial intelligence and statistics*. PMLR, 1431–1440.

[27] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. 2022. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*.

[28] Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31.

[29] Glenn Shafer. 1976. *A mathematical theory of evidence*. Vol. 42. Princeton university press.

[30] Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. 2023. Faithful heteroscedastic regression with neural networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 5593–5613.

[31] Xiang Sun, Seongyoon Kim, and Jung-Il Choi. 2022. Recurrent neural network-induced gaussian process. *Neurocomputing*, 509, 75–84. DOI: https://doi.org/10.1016/j.neucom.2022.07.066.

[32] Katarína Tóthová, L'ubor Ladickỳ, Daniel Thul, Marc Pollefeys, and Ender Konukoglu. 2022. Quantification of predictive uncertainty via inference-time sampling. In *International Workshop on Uncertainty for Safe Utilization of Machine Learning in Medical Imaging*. Springer, 14–25.

[33] Gia-Lac Tran, Edwin V Bonilla, John Cunningham, Pietro Michiardi, and Maurizio Filippone. 2019. Calibrating deep convolutional gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1554–1563.

[34] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*. PMLR, 9690–9700.

[35] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. 2012. Bayesian modeling with gaussian processes using the gpstuff toolbox. *arXiv preprint arXiv:1206.5754*.