Input Uncertainty Attribution by Uncertainty Propagation

Benedikt Kantz [1] *Sophie Steger* [1] *Clemens Staudinger* [2] *Christoph Feilmayr* [2]
Johann Wachlmayr [3] *Alexander Haberl* [2] *Stefan Schuster* [2] *Franz Pernkopf* [1 4]

[1]Signal Processing and Speech Communication Laboratory, Technical University Graz, Graz, Austria
[2]voestalpine Stahl GmbH, Linz, Austria
[3]K1-MET GmbH, Linz, Austria
[4]Christian Doppler Laboratory for Dependable Intelligent Systems in Harsh Environments, Graz, Austria

***Abstract*—Attributing uncertainties to the input space elevates the trustworthiness and explainability of machine learning applications. This paper proposes a novel method called Smoothness Constrained Attribution (SCA), which uses the uncertainty propagation mechanism to propagate the output uncertainty back to the input space. This input uncertainty attribution relies solely on test-time data, the trained uncertainty-aware Machine Learning (ML) model, and assumes a smooth input space, resulting in an efficient and simple system. SCA is compared to existing input Uncertainty Attribution Mechanisms (iUCAMs) based on eXplainable Artificial Intelligence (XAI) and an oracle reference using heteroscedastic noise in different synthetic datasets. These evaluations demonstrate the robustness and improvements of SCA compared to existing methods.**

***Index Terms*—Uncertainty, machine learning, uncertainty attribution, explainable artifical intelligence, trustworthy machine learning**

I. Introduction

ML models have seen a steady rise in performance in many application fields in the last years [1]. Besides providing function estimation tools, these ML models can also drive the understanding of the processes they are trying to model. This can be achieved using current XAI methods, providing insights into how these models arrive at certain predictions [2]. Many application fields, furthermore, require, besides being explainable, robustness in the case of noisy data. One path to robustness is Uncertainty Quantification (UCQ), utilizing uncertainty-aware ML models, like Gaussian processes, ensembles or single deterministic methods [1], [3], [4]. These models estimate the aleatoric and epistemic uncertainties, where the former represents the uncertainty inherent within the data, while the latter covers out-of-domain samples, model and training uncertainty, and distribution shifts. These uncertainties are predicted at the models' outputs [3]. Such robust, uncertainty-aware systems are essential for implementing ML as the trust in these systems can be increased significantly by using explanations and uncertainty estimates presented to the end-users [5]. Besides offering more transparency, the use of uncertainty-aware systems can also improve the decision-making process of people when uncertainties are paired with ML predictions [6], [7]. These uncertainty-aware systems can be extended further to predict the driving inputs of the uncertainties through iUCAMs.

For example, aleatoric uncertainty estimations for the input features can be especially helpful when dealing with measurement inaccuracies of unknown origins. iUCAMs are able to provide attributions of these uncertainties to inputs responsible for the uncertainties on a local level. Existing iUCAMs utilize the aleatoric uncertainty provided by uncertainty-aware ML models and use XAI methods to attribute the uncertainties. They use a wide range of methods to indirectly measure the effectiveness of the mechanisms they present [8]–[10]. These methods, however, do not consider a comparison to the ground truth uncertainties at the inputs.[1] They, furthermore, ignore the classical uncertainty propagation framework [11] used in many fields of engineering to determine the uncertainty mapped through a function.

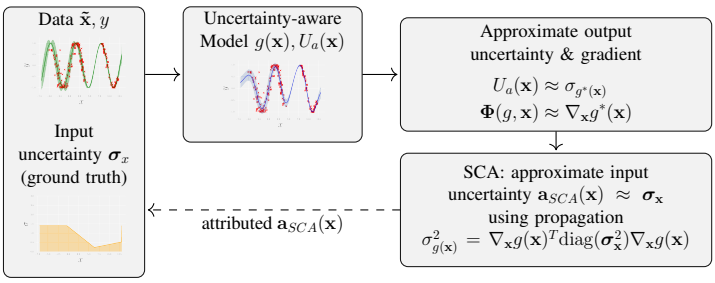


Fig. 1: Overview of the SCA.

In this paper, we propose a *Smoothness Constrained Attribution (SCA)*. Furthermore, we evaluate our SCA and existing iUCAMs by comparing the attributed input uncertainty to the real perturbations using simulated datasets. The existing approaches [8]–[10] only utilize the output space to estimate each

This work is supported by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology, and Development, the Christian Doppler Research Association, and the voestalpine Stahl GmbH.

[1]These ground truth uncertainties are available in simulated scenarios

input feature's contributions to a model's uncertainty. Our SCA method, shown in Figure 1, first uses a trained uncertainty-aware ML model. This model provides the function $g(\mathbf{x})$ and aleatoric uncertainty $U_a(\mathbf{x})$. The output uncertainty $\sigma_{g(\mathbf{x})}$ can also be approximated using the gradient $\nabla_{\mathbf{x}} g(\mathbf{x})$ and input uncertainty $\sigma_{\mathbf{x}}$ using the uncertainty propagation [11]. This is then used to attribute the uncertainties back towards the input space using a simple least squares problem. The proposed attribution mechanism can not only identify the features driving the uncertainty but also quantify the uncertainty within the input space accurately, as we show in the evaluation. Furthermore, we introduce a source-free test-time SCA approach, as this kind of system is especially advantageous as the initial training data (source) or further existing samples are not required to determine the input uncertainties. This independence from source samples is achieved by adding Gaussian noise to the test-time sample, which enables us to solve the least squares problem. Hence, only the trained uncertainty-aware ML model is necessary. Additionally, we provide an oracle-based reference that only depends on the simulated heteroscedastic input noise and no ML model. The input noise is propagated using the uncertainty propagation formula to calculate the output uncertainty. The input uncertainty is then recovered by gathering neighbors of a data sample, applying our SCA, and compared to the simulated input noise.

II. Input Uncertainty Attribution

A. Deriving SCA

First, assuming a linear function $\mathbf{g}(\mathbf{x})$ with a $d \times d$ matrix $\mathbf{A}$ and vector $\mathbf{b} \in \mathbb{R}^d$ in the form of

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}. \tag{1}$$

The uncertainty $\mathbf{\Sigma_g}$ of this function $\mathbf{g}(\mathbf{x})$ can be propagated using a known uncertainty $\mathbf{\Sigma_x}$ of $\mathbf{x} \in \mathbb{R}^d$ [11], i.e.

$$\mathbf{\Sigma_g} = \mathbf{A}^T \mathbf{\Sigma_x} \mathbf{A}. \tag{2}$$

If the function $\mathbf{g}(\mathbf{x})$ is nonlinear, we can still approximate it as a linear function locally using a Taylor expansion such as

$$\mathbf{g}(\mathbf{x}') = \mathbf{g}(\mathbf{x}) + \mathbf{J}_{\mathbf{x}}^T(\mathbf{x}' - \mathbf{x}) + \mathcal{O}(\mathbf{x}' - \mathbf{x}), \tag{3}$$

where $\mathbf{J}_{\mathbf{x}}^T(\mathbf{x}' - \mathbf{x})$ is a local linear approximation, with $\mathbf{J_x}$ being the Jacobian of $\mathbf{g}(\mathbf{x})$. Then the uncertainty can therefore still be propagated at $\mathbf{x}$ using

$$\mathbf{\Sigma_g} = \mathbf{J}_{\mathbf{x}}^T \mathbf{\Sigma_x} \mathbf{J_x}. \tag{4}$$

We assume a single-output function $g(\mathbf{x})$ and constrain $\mathbf{\Sigma_x}$ to an *uncorrelated* uncertainty matrix $\mathrm{diag}(\boldsymbol{\sigma}_{\mathbf{x}}^2)$, thus constraining the solution to be independent between the features of the input. Then, Eq. 4 simplifies to

$$\sigma_{g(\mathbf{x})}^2 = \nabla_{\mathbf{x}} g(\mathbf{x})^T \mathrm{diag}(\boldsymbol{\sigma}_{\mathbf{x}}^2) \nabla_{\mathbf{x}} g(\mathbf{x}) = \tag{5}$$

$$= \left(\nabla_{\mathbf{x}} g(\mathbf{x})^2\right)^T \boldsymbol{\sigma}_{\mathbf{x}}^2, \tag{6}$$

using a simple dot-product between the element-wise square of the gradient $\nabla_{\mathbf{x}} g(\mathbf{x})$ and the input uncertainties $\boldsymbol{\sigma}_{\mathbf{x}}^2$. If the output uncertainty $\sigma_{g(\mathbf{x}_i)}^2$ is known but the input uncertainty $\boldsymbol{\sigma}_{\mathbf{x}}^2$ is unknown, an underdetermined problem of the form

$$\hat{\boldsymbol{\sigma}}_{\mathbf{x}}^2 = \arg\min_{\boldsymbol{\sigma}_{\mathbf{x}}^2} \left((\nabla_{\mathbf{x}} g(\mathbf{x})^2)^T \boldsymbol{\sigma}_{\mathbf{x}}^2 - \sigma_{g(\mathbf{x}_i)}^2 \right)^2 \tag{7}$$

has to be solved. By assuming that the neighborhood $\{\mathbf{x}_h | \mathbf{x}_h \in K\text{−neighbourhood of } \mathbf{x}_i\}$ around $\mathbf{x}_i$ has a similar input uncertainty $\boldsymbol{\sigma}_{\mathbf{x}_i}^2 \approx \boldsymbol{\sigma}_{\mathbf{x}_h}^2$ and is locally smooth around K neighbors [12], we can minimize over $\{\mathbf{x}_h\}$:

$$\hat{\boldsymbol{\sigma}}_{\mathbf{x}_i}^2 = \arg\min_{\boldsymbol{\sigma}_{\mathbf{x}_i}^2} \sum_{h=0}^{K} \left(\left(\nabla_{\mathbf{x}_h} g(\mathbf{x}_h)^2\right)^T \boldsymbol{\sigma}_{\mathbf{x}_i}^2 - \sigma_{g(\mathbf{x}_h)}^2 \right)^2. \tag{8}$$

This can be easily formulated as a least squares problem, assuming $K > d$ and linear independence of the gradients, using

$$\mathbf{B} = \begin{pmatrix} \left(\nabla_{\mathbf{x}_0} g(\mathbf{x}_0)^2\right)^T \\ \vdots \\ \left(\nabla_{\mathbf{x}_K} g(\mathbf{x}_K)^2\right)^T \end{pmatrix} \text{ and } \mathbf{s} = \begin{pmatrix} \sigma_{g(\mathbf{x}_0)}^2 \\ \vdots \\ \sigma_{g(\mathbf{x}_K)}^2 \end{pmatrix}.$$

We can find $\boldsymbol{\sigma}_{\mathbf{x}_i}$ by solving the least squares problem over all neighboring samples $\mathbf{x}_h$, i.e.

$$\hat{\boldsymbol{\sigma}}_{\mathbf{x}_i}^2 = \arg\min_{\boldsymbol{\sigma}_{\mathbf{x}_i}} \left\| \mathbf{B}\boldsymbol{\sigma}_{\mathbf{x}_i}^2 - \mathbf{s} \right\|_2^2. \tag{9}$$

This unconstrained least squares problem can be solved using the Moore-Penrose pseudo-inverse of $\mathbf{B}$, i.e.

$$\hat{\boldsymbol{\sigma}}_{\mathbf{x}_i,\text{unc.}}^2 = \left(\mathbf{B}^T\mathbf{B}\right)^{-1} \mathbf{B}^T\mathbf{s}. \tag{10}$$

The problem can be extended by a regularization term $\frac{\lambda}{2} \left\| \boldsymbol{\sigma}_{\mathbf{x}_i} \right\|_2^2$ to penalize greater uncertainties and overfitting:

$$\hat{\boldsymbol{\sigma}}_{\mathbf{x}_i}^2 = \arg\min_{\boldsymbol{\sigma}_{\mathbf{x}_i}} \left\| \mathbf{B}\boldsymbol{\sigma}_{\mathbf{x}_i}^2 - \mathbf{s} \right\|_2^2 + \frac{\lambda}{2} \left\| \boldsymbol{\sigma}_{\mathbf{x}_i}^2 \right\|_2^2 \tag{11}$$

This ℓ_2-regularized objective yields the SCA, i.e.

$$\mathbf{a}_{SCA}(\mathbf{x}_i) = \hat{\boldsymbol{\sigma}}_{\mathbf{x}_i,\ell_2} = \sqrt{\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I}\right)^{-1} \mathbf{B}^T\mathbf{s}}. \tag{12}$$

B. Approximations & Implementation of SCA

We approximate the gradient $\nabla_{\mathbf{x}} g^*(\mathbf{x})$ of the ground truth function $g^*(\mathbf{x})$ through the XAI method SmoothGrad (SG) [13] performed on the model $g(\mathbf{x})$, denoted as $\mathbf{\Phi}(g, \mathbf{x})$. The uncertainty $\sigma_{g^*(\mathbf{x})}^2$ is assumed to be correctly estimated by an uncertainty-aware ML model. We use the Light Gradient Boosting Machine (LightGBM) [14], where a Prediction Interval (PI) is fitted using quantile regression. This PI is converted to a scalar aleatoric uncertainty $U_a(\mathbf{x})$ by assuming a Gaussian distribution and fitting it to the prediction interval. This model uses two sub-models for the prediction interval $[g_-(\mathbf{x}), g_+(\mathbf{x})]$ to estimate the probability that a sample is within the PI. Additionally, a mean function $g(x)$ is learned from the data. Furthermore, the PI of the LightGBM is calibrated using *inductive conformal prediction* [15] to ensure that the probability that the output is within this PI with a guaranteed probability. This calibration ensures correct magnitudes of the

uncertainties [3], [16]. This calibrated $U_a(\mathbf{x})$ is assumed to approximate $\sigma_{g^*(\mathbf{x})}$.

To ensure a sufficient number of samples locally, we use $K = d^2$ neighbors. These neighbors are either determined through a k-Nearest Neighbors (kNN) approach or by sampling from a local Gaussian distribution around the test sample $\mathbf{x}_i$ with a small enough $\boldsymbol{\sigma}_s$.

III. Evaluation

Our evaluation framework takes samples from a known data-generating distribution and nonlinear functions. These samples are perturbed using known, heteroscedastic noise levels $\boldsymbol{\sigma}_\mathbf{x}$ to introduce aleatoric uncertainty in the input space artificially. This noise is linearly scaled using a noise level l before adding it to the inputs to evaluate the methods for their noise robustness. The noisy data is used to train and calibrate the LightGBM to provide the uncertainties and the prediction function $g(x)$. The calibrated estimated uncertainties are then either explained using XAI or propagated to the input space by SCA. All of these outputs are finally compared and scored with respect to their ground truth values using the Mean Squared Error (MSE), resulting in their scores s_{ℓ_2} [17].

A. Data-generating Processes

The dataset-generating process consists of two parts, the data-generating distribution $\mathcal{X}$ and the non-linear ground truth functions $g^*(\mathbf{x})$. We employ simple uniform and Gaussian distributions to generate the data $\mathbf{x}_i \sim \mathcal{X}, \mathbf{x}_i = [x_1^i, \ldots, x_d^i]^T$, where d represents the dimensionality of the dataset. Each dataset consists of 5000 samples for training, 500 for calibration, and 500 for testing. We sample 20 datasets and perform an evaluation run for each set.

1) Polynomial Datasets: This dataset is intended to validate our system on low-dimensionality tasks with simple nonlinear functions to visualize and compare the different models and methods quickly. To this end, the data-generating distributions $\mathcal{X}_j$ of each dimension are chosen to be either simple uniform distributions or Gaussian Mixture Models (GMMs) with two components. The nonlinear mapping functions are quadratic, i.e.

$$g^*(x_1, x_2) = k_1 x_1^2 + k_2 x_2^2 + k_3 x_1 x_2 + k_4 x_1 + k_5 x_2 + k_6, \quad (13)$$

where the coefficients $k_l, l \in \{1, \ldots, 6\}$ are drawn from $k_l \sim \mathrm{Uniform}(0, 1)$ once for each run.

2) Styblinski-Tang Dataset: The Styblinski-Tang data-generating function [18] enables the creation of high-dimensional datasets with complex polynomial functions through simple linear combinations. We extended this formula by a parameter k_j to create non-uniform mixtures of dimensions, resulting in a modified Styblinski-Tang function of the form

$$g^*(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^{d} k_j (x_j^4 - 16 x_j^2 + 5 x_j). \quad (14)$$

The parameters k_j are sampled using $k_j \sim \mathrm{Uniform}(0, 1)$ once. The samples $\mathbf{x}$ are sampled using $x_j \sim \mathrm{Uniform}(-5, 5)$.

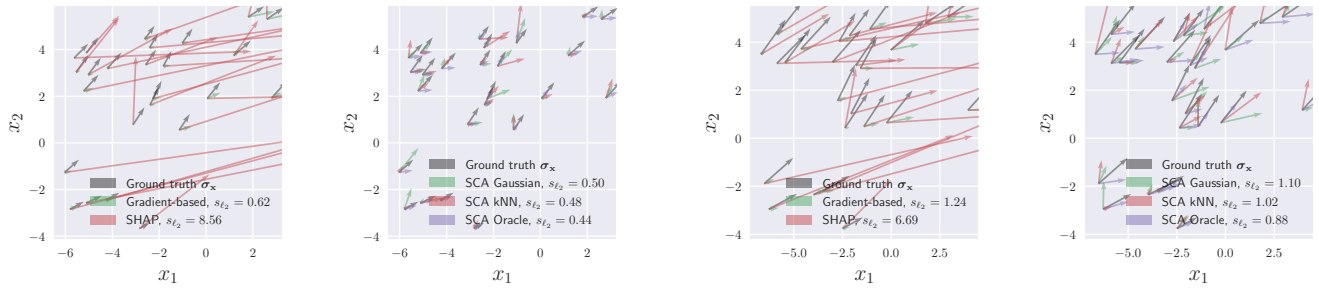
B. Input Uncertainty Attribution Methods and References

- Our *SCA Gaussian* method uses the SCA framework from above with $K = d^2$ neighbors generated using sampling of a Gaussian centered on a test data sample $\mathbf{x}_i$ with a small $\boldsymbol{\sigma}_s$. The gradient $\nabla_\mathbf{x} g(\mathbf{x})$ of each of those samples is approximated using SG [13] with another Gaussian sampling step for a more accurate gradient [17], applied to the model $g(\mathbf{x})$. This method requires just a single test time sample.
- Our second approach, *SCA kNN* uses the same approximations as the Gaussian SCA but uses kNN of the test sample to determine the neighborhood for both the gradient and SCA.
- The first XAI-based approach, *Gradient-based* explains the aleatoric uncertainty directly [8]. The explanations for this approach are generated by an effect-based explainer, which is SG with kNN in this case. The SG produces two gradient approximations $\boldsymbol{\Phi}(g_-, \mathbf{x}), \boldsymbol{\Phi}(g_+, \mathbf{x})$ from the explanations as the uncertainty is modeled using quantiles, which are averaged into the attribution $\mathbf{a}_{\mathrm{XAI}}(\mathbf{x})$
- The second XAI approach, *SHAP*, trades the gradient-based explanations for the SHapley Additive exPlanations (SHAP) framework [9], [19] but attributes the uncertainty in the same way as the Gradient-based iUCAM by explaining both prediction intervals and averaging them.
- Our first reference iUCAM, called *SCA Oracle* utilizes our SCA in an ideal setting, which assumes a perfect uncertainty-aware ML model. The necessary output uncertainty $\sigma_{g(\mathbf{x})}$ is determined by utilizing the uncertainty propagation formula from Eq. 6, the known input noise $\boldsymbol{\sigma}_\mathbf{x}$, and the ground-truth gradient $\nabla_\mathbf{x} g^*(\mathbf{x})$ from the nonlinear function $g^*(\mathbf{x})$. No uncertainty-aware ML model is used here. The smoothness constraint is assumed, and the gathering of the neighbors using kNN is used for SCA.
- Finally, the second reference we use is a *Predict 0* baseline, always predicting no uncertainty in input space. This baseline serves as a worst-case benchmark.

IV. Results

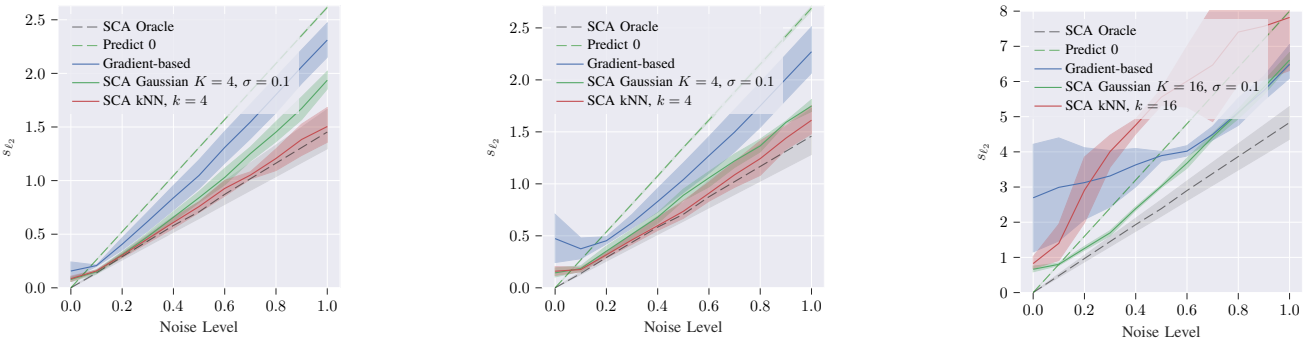
A. Qualitative Input Uncertainty Attribution Evaluation

First, the different iUCAM methods are applied on a simple test set in $\mathbb{R}^2$, providing a qualitative overview. Figure 2 shows the input uncertainties for $\mathbf{x} \in \mathbb{R}^2$, i.e. we have an attribution of each tested method. The figure shows the same evaluation performed two times with different levels of noise applied and being attributed back. The results with the lower noise levels in Figure 2a show that the magnitude and direction of the attribution methods, either SHAP or the Gradient-based approach, do not match the ground truth magnitude. These problems persist throughout higher noise levels, shown in Figure 2b. Our SCA approaches and reference are, in comparison, much more accurate, as shown in Figures 2a and 2b. The SCA kNN sampling performs similar to the SCA Gaussian in both settings. The SCA's improvement can be attributed to the improved magnitude estimation. The direction



(a) $l = 0.30$ (b) $l = 0.60$

Fig. 2: Vectors of the uncertainty attribution in $\mathbb{R}^2$ input space using SHAP attribution, Gradient-based attribution, SCA Gaussian, SCA kNN and the SCA Oracle over two different noise settings on a small subset of the polynomial GMM dataset.



(a) Polynomial (b) Polynomial GMM (c) Styblinski-Tang, $d = 4$

Fig. 3: Results of the uncertainty attribution noise sweep using the LightGBM on different dataset, including baselines, averaged over 20 runs, the shaded area represents the 10th to 90th percentile.

of the input uncertainty attributions compared to the ground truth is also much closer. These three SCA systems perform better than the existing XAI-based methods on this toy dataset. Another observation is that the performance of the SCA kNN and the SCA Gaussian is highly dependent on the region that the data is sampled from and how dense this region is present in the dataset. The reference SCA Oracle performs best as it does not depend on a trained ML model. However, it cannot be applied to a real-world dataset, as the reference uncertainties and gradients are usually unknown.

B. Input Uncertainty Attribution Robustness Analysis

Similar properties to the qualitative example can be observed for the final attribution evaluation over increasing noise levels l, shown in Figure 3. Beginning with the existing methods, the Gradient-based iUCAM results in the worst performance scores across almost all datasets and noise settings. In the higher dimensional setting in Figure 3c it only matches the SCA in the regime of high noise. The SCA is the best-performing attribution methods across the lower-dimensional benchmarks of Figures 3a and 3b. Our approach mostly struggles with the higher dimensional setting and high noise in Figure 3c, most likely due to worse uncertainty estimation performance of the LightGBM. Notwithstanding, the SCA Gaussian achieves the closest result to the SCA Oracle throughout all tested settings.

V. Conclusion

Uncertainty-aware ML models are used in combination with XAI methods to attribute uncertainties to the input space. These systems, however, are not evaluated with regard to the input perturbations and disregard the propagation of uncertainty. This paper alleviates these omissions by introducing SCA. SCA assumes a smooth input space and a well-trained uncertainty-aware ML model to approximate the gradient and uncertainties present in a sample and can, without any further data, accurately predict the local input uncertainties. This source-free uncertainty attribution enables users to estimate the input uncertainties of individual features efficiently.

This framework is validated on several datasets of different dimensionalities and with varying noise levels of heteroscedastic noise, verifying its robustness to noise and estimating the capabilities for further applications. Nevertheless, SCA outperforms all existing XAI-based attribution methods throughout our tested noise range and datasets. We observe limited performance of SCA on datasets with more input dimensions, which we aim to improve in future works.

REFERENCES

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001081

[2] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[3] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. S1, p. 1513–1589, Jul. 2023. [Online]. Available: http://dx.doi.org/10.1007/s10462-023-10562-9

[4] S. Steger, C. Knoll, B. Klein, H. Fröning, and F. Pernkopf, "Function space diversity for uncertainty prediction via repulsive last-layer ensembles," in *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=FbMN9HjgHI

[5] U. Bhatt, J. Antorán, Y. Zhang, Q. V. Liao, P. Sattigeri, R. Fogliato, G. Melançon, R. Krishnan, J. Stanley, O. Tickoo, L. Nachman, R. Chunara, M. Srikumar, A. Weller, and A. Xiang, "Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty," in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 401–413. [Online]. Available: https://doi.org/10.1145/3461702.3462571

[6] L. R. Marusich, J. Z. Bakdash, Y. Zhou, and M. Kantarcioglu, "Using ai uncertainty quantification to improve human decision-making," 2024. [Online]. Available: https://arxiv.org/abs/2309.10852

[7] J. C. Cresswell, Y. Sui, B. Kumar, and N. Vouitsis, "Conformal prediction sets improve human decision making," 2024. [Online]. Available: https://arxiv.org/abs/2401.13744

[8] H. Wang, D. Joshi, S. Wang, and Q. Ji, "Gradient-based uncertainty attribution for explainable bayesian deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 044–12 053.

[9] P. Iversen, S. Witzke, K. Baum, and B. Y. Renard, "Identifying drivers of predictive aleatoric uncertainty," 2024.

[10] J. Antoran, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato, "Getting a {clue}: A method for explaining uncertainty estimates," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=XSLF1XFq5h

[11] K. O. Arras, "An introduction to error propagation: derivation, meaning and examples of equation cy= fx cx fxt," ETH Zurich, Tech. Rep., 1998.

[12] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, p. 674–679.

[13] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[15] V. Vovk, "Conditional validity of inductive conformal predictors," *Machine Learning*, vol. 92, no. 2-3, pp. 349–376, May 2013. [Online]. Available: http://dx.doi.org/10.1007/s10994-013-5355-6

[16] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," 2022.

[17] B. Kantz, C. Staudinger, C. Feilmayr, J. Wachlmayr, A. Haberl, S. Schuster, and F. Pernkopf, "Robustness of explainable artificial intelligence in industrial process modelling," 2024. [Online]. Available: https://arxiv.org/abs/2407.09127

[18] M. Styblinski and T.-S. Tang, "Experiments in non-convex optimization: Stochastic approximation with function smoothing and simulated annealing," *Neural Networks*, vol. 3, no. 4, pp. 467–483, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/089360809090029K

[19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.