A Simple and Explainable Method for Uncertainty Estimation using Attribute Prototype Networks

Claudius Zelenka Andrea Göhring Daniyal Kazempour Maximilian Hünemörder
Lars Schmarje Peer Kröger

Kiel University

{cze, ang, dka, mah, las, pkr}@informatik.uni-kiel.de

Abstract

Deep learning's utility in applications like medical diagnosis, autonomous driving, and natural language processing often hinges on the accurate estimation of uncertainty. Yet, conventional methods for uncertainty estimation face challenges, including high computational cost, difficulties with scalability, or poor interpretability. This paper presents a novel approach to uncertainty estimation using Attribute Prototype Networks (APNs), a method designed for learning robust and interpretable data representations. By leveraging prototype similarity scores, we propose a straightforward way to quantify the uncertainty of predictions, providing explainability and introducing a new technique for detecting out-of-distribution samples based on the distance to the nearest prototype. Our experiments demonstrate that this method offers valuable uncertainty information across several datasets. Our research opens up a new avenue for uncertainty estimation in deep learning, providing a simpler and more explainable solution.

1. Introduction

Uncertainty estimation is a crucial task for many applications of deep learning, such as medical diagnosis, autonomous driving, and natural language processing.

However, most existing methods for uncertainty estimation have some drawbacks, such as computational complexity, scalability issues or poor explainability. Several approaches have been explored in the existing literature.

Gaussian processes (GPs), valued for their probabilistic non-parametric modeling, encounter limitations, especially for large datasets [1]. The high computational cost arises from the cubic time complexity ($O(n^3)$) due to the inversion of the covariance matrix. Further, they suffer from the 'curse of dimensionality', especially when dealing with high-dimensional data.

Bayesian methods, popular choices for applications in machine learning and data science, offer a probabilistic framework for modeling uncertainty [2]. These methods allow for the integration of prior knowledge and observations, yielding predictive distributions that inherently quantify uncertainty. However, the complexity of interpreting the high-dimensional prior and posterior distributions presents a challenge. As a result, extracting actionable insights from Bayesian models can be particularly challenging for complex, high-dimensional problems [3]. Moreover, Bayesian methods often necessitate a reimplementation with a framework such as Pyro [4], making their inclusion into existing classification pipelines difficult.

As another alternative, **variational inference** is a method of approximate Bayesian inference that has also been applied to deep learning problems, including the estimation of uncertainty [5]. One specific application is in the realm of Bayesian neural networks, where variational inference is used to learn the posterior distribution over the network's weights. [6] showed that Dropout CNN after every weight is equivalent to variational approximation of the Bayesian Neural network with Bernoulli distribution prior. It averages over T forward passes through the network at test time (as opposed to upscaling the weight by the dropout ratio in the conventional dropout at test time). It is the Monte Carlo estimation of the predictive distribution and is therefore called **MCDropout**.

The **ensemble neural networks (ENN)** approach requires training several models with different models, architectures or initializations [7]. This approach is based on the assumption that data for which a prediction is uncertain a different model may come to a different result.

Conformal prediction is a model agnostic approach to measuring confidence by providing a set or range of predictions given a confidence threshold. It uses additional, previously unseen samples to measure the conformity of the model with its previous predictions [8]. Conformal prediction can be applied to outlier detection[9], image classification[10] and question answering[11].

Despite these various methods, existing techniques often struggle to capture the uncertainty of out-of-distribution (OOD) samples and lack sufficient explainability [12]. For a more comprehensive review of these techniques, the reader is referred to [13].

Prototype networks (PNs) are a recent approach for learning interpretable and robust representations of data. PNs use prototypes to capture the salient features of different classes or attributes, and measure the similarity between inputs and prototypes to make predictions. This paradigm of using prototypes for salient image parts is explored in various publications [14] [15] [16]. Prototypes have also been applied to semantic segmentation [17].

In [18] uncertainty is calculated by the closest exponential distance to predetermined centroids for every class in a class-weighted feature space. These centroids can be seen as prototypes. While also not explicitly calling them prototypes, a similar distance based idea is used in [19] where an exponential distance measure is applied to logits of one-vs-all classifiers to calculate uncertainty. Finally, the LDU (latent discriminant deterministic uncertainty) method [20] uses cosine similarity to trained prototypes before an uncertainty estimation layer and loss, because they argue that distance measures (L1/L2) lead to instabilities in training [20].

Attribute prototype networks (APNs) are an example of PNs that follows this principle and efficiently uses human attribute annotations. Attributes are properties of a class which are semantically meaningful, an attribute could be if that this bird has a yellow beak. Hence, the attribute prototypes are also meaningful. They have been applied to various tasks, such as zero-shot learning [21] and any-shot learning [22] in image classification. However, there is a lack of research on the use of APNs for uncertainty estimation.

We argue that by way of providing prototypes for meaningful attributes, APNs can provide a interpretable way to quantify the uncertainty of predictions by using similarity scores to these attribute prototypes. The contributions of this paper are as follows:

- We propose a **straightforward and novel method for uncertainty estimation** based on APNs.

- We show that our method provides valuable uncertainty information and how these scores **provide explanatory power**. We also analyse the calibration of the uncertainty estimates.

- We further introduce a novel technique for **detecting OOD samples** based on the distance to the nearest attribute prototype.

We strongly believe that our method will offer a new perspective on uncertainty estimation contributing to more interpretable deep learning applications. Thus, our goal is not to beat the SOTA calibration scores on benchmark datasets, but we want to show the direct benefit of attribute information.

The rest of this paper is organized as follows. In the next section we will explain how the similarity to prototypes is calculated, followed by Section 3 with datasets, models and metrics. Then we provide results in Section 4 showing how well we capture uncertainty information on different datasets and how the OOD detection works. Because of it's importance, the explainability part is set in an extra Section followed by a conclusion, in which we discuss advantages, limitations and future work.

2. Methodology

The similarity measure $s(x, p_i)$ is the cornerstone of our approach. The similarity function quantifies the resemblance between an input x and a prototype p_i in the feature space. This function plays a crucial role in defining the notion of proximity that underlies our uncertainty estimation method. We define $s(x, p_i)$ as the cosine similarity between the feature representation of x and the prototype p_i, formulated as:

$$s(x, p_i) = \frac{\langle f(x), p_i \rangle}{||f(x)||_2 ||p_i||_2} \tag{1}$$

where $f(x)$ is the feature extractor output for input x, $\langle ., . \rangle$ denotes the dot product, and $||.||_2$ represents the L_2 norm. The cosine similarity scales the dot product of $f(x)$ and p_i by their magnitudes, ensuring that the similarity score is not influenced by the scale of the features or the prototypes. The result is a value bounded between -1 (completely dissimilar) and 1 (perfectly similar), providing a clear interpretation of the similarity scores.

The proximity function can be defined in terms of the similarity measure. The proximity of an input x to the prototypes is given by:

$$\text{proximity}(x) = \max_i s(x, p_i) \tag{2}$$

where i varies over the total number of prototypes. The proximity score represents the maximum similarity of x to any prototype. This provides a measure of confidence in the classification. The closer the proximity score is to 1, the more confident we are about the class prediction, hence the lower the prediction uncertainty. In our implementation (described in Section 3.2 and illustrated in Figure 1) the feature vector $f(x)$ has a spatial dimension of 7x7 indexed by u and v which allows a rough localization of activations. To take this into account, we calculate the maximum similarity for each spatial element. Hence,

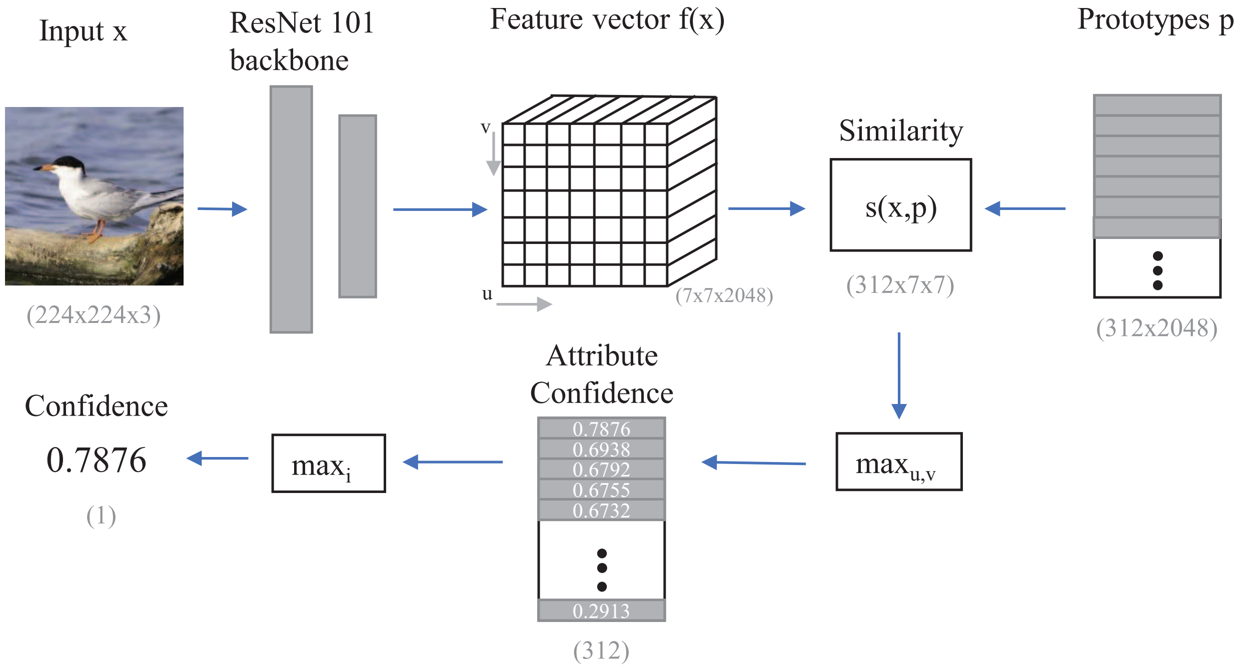


Figure 1: The calculation of confidence with an Attribute-Prototype model.

$$\text{proximity}_{u,v}(x) = \max_{i} \max_{u,v} s(x_{u,v}, p_i) \tag{3}$$

where u and v are the index variables for possible spatial dimensions of $f(x)$ as shown in Figure 1.

For a more intuitive confidence measure we normalize the proximity ($0 =$ not confident, $1 =$ highly confident):

$$p(x) = \frac{\text{proximity}(x) + 1}{2} \tag{4}$$

The general assumption of our Out-of-Distribution (OOD) detection approach is that images with low confidence, i.e., very high uncertainty are less likely to be from the same distribution as the training images. As for OOD detection, we introduce the concept of a threshold τ. Inputs with proximity scores less than this threshold are considered uncertain or potentially OOD. Since the confidence is in the range of 0 to 1, a reasonable choice for τ might be a small positive number in this range, allowing for a tolerance of inputs with very low similarity to all prototypes. Formally:

$$\text{OOD}(x) = \begin{cases} 1, & \text{if } p(x) < \tau \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\text{OOD}(x) = 1$ denotes an OOD sample and $\text{OOD}(x) = 0$ denotes an in-distribution sample. Note that OOD detection is also called anomaly detection or outlier detection, depending on applications.

3. Experimental setup

3.1. Datasets

We evaluate our method on three widely used datasets: CUB-200-2011 (CUB) [23], SUN Attribute (SUN) [24], and Animals with Attributes 2 (AWA2) [25]. These datasets are selected for their diverse nature and the availability of attribute information, which is crucial for our Attribute Prototype Networks (APNs). The CUB dataset contains 11,788 images of 200 bird species, each associated with a set of attributes. The SUN dataset consists of 14,340 images from 717 categories, each with a unique set of attributes. The AWA2 dataset includes 37,322 images of 50 animal classes, each associated with 85 attributes.

3.2. Models

Generalized Zero-Shot Learning (GZSL) is a machine learning paradigm that extends Zero-Shot Learning (ZSL) with the goal of recognizing instances from both seen and unseen classes [26]. Seen classes are those for which training data is available, enabling supervised learning models to infer the relationships. In contrast, unseen classes refer to categories without training examples; these must be inferred indirectly via shared attributes or other indirect information [27]. The key challenge in GZSL is to minimize the bias towards seen classes while improving recognition performance on unseen classes, which often leads to the so-called 'domain shift' problem [28]. This approach has a wide range of applications, particularly in areas such as object and activity recognition, where it is unrealistic to obtain labeled data for all possible classes [29].

We use attribute prototype models by [21] trained on the dataset explained in Section 3.1. Our model is based on a pretrained ResNet-101 [30] backbone and designed for GZSL. It provides us with a 2048 element prototype vector for every attribute defined in the dataset which helps in predicting labels of either seen or unseen classes. For our experiments we use models trained on the Generalized Zero-Shot Learning task. For comparison against other baselines of uncertainty estimation please refer to future work in Section 6.

In Figure 1, we illustrate the architecture during inference including feature and prototype dimensions for an image from the CUB dataset.

3.3. Metrics

Calibration refers to the agreement between predicted uncertainties and true outcomes. We use several calibration metrics, which focus on different aspects of calibration error and the Out-of-Distribution (OOD) detection accuracy to evaluate our method:

Expected Calibration Error (ECE): ECE is a popular metric for evaluating the calibration of a model [31] [32]. It measures the average discrepancy between the model's confidence and the accuracy of its predictions. To calculate ECE, we first sort the predictions by their confidence into M bins. Then, for each bin B_m, with samples per bin $|B_m|$, we calculate the absolute difference between the bin's *accuracy* $acc(B_m)$ and the average *confidence* $conf(B_m)$, that are defined as follows:

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i)$$

where $\mathbb{1}(.)$ is the indicator function of all correctly classified samples; $\hat{y}_i$ and y_i denote the predicted and true class label for a given sample i.

and

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} p(i)$$

where $p(i)$ denotes the confidence for a sample i.

In all of the following metrics a lower deviation between accuracy and confidence indicates a lower error. In an ideal case the discrepancy between accuracy and confidence is zero, meaning that in all of the metrics a lower score indicates a lower error and hence a better calibration of the model.

The ECE is the weighted average of these differences, with the weights being the number of samples per bin. It is defined as follows:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

with n denoting the number of samples. Lower ECE values indicate better calibration. If the classifiers are perfectly calibrated, the ECE score is 0.

In scenarios where it is imperative to have a reliable confidence measure this is achieved by minimizing the maximum possible deviation between accuracy and confidence. For this purpose the so called **Maximum Calibration Error (MCE)** is defined as follows (see [33] and [31]):

$$MCE = \max_{m \in \{1,\ldots,M\}} |acc(B_m) - conf(B_m)|$$

To account for the multi-class setting the authors in [34] proposed the **Static Calibration Error (SCE)** which performs binning for each class probability. It computes the calibration error within each bin followed by an averaging across all bins:

$$SCE = \frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M} \frac{B_{mk}}{n} |acc(B_{mk}) - conf(B_{mk})|$$

where K denotes the total number of classes and B_{mk} denotes the number of predictions for a particular bin m for a class label k.

While ECE, MCE and SCE rely on static and equally sized bin intervals, the **Adaptive Calibration Error (ACE)** [33] relies on a binning with adaptive intervals working on the premise that each bin contains an equal number of predictions. The motivation for such an adaptive interval binning emerges from the need to focus on regions in the probability distributions where the majority of predictions are performed, rather than regions that contain lower number of predictions. ACE is computed as follows:

$$ACE = \frac{1}{KR} \sum_{k=1}^{K} \sum_{r=1}^{R} |acc(B_{rk}) - conf(B_{rk})|$$

where r denotes the calibration range, that refer to the intervals within which the predicted probabilities of a model

Table 1: Accuracy values in % for GZSL based on pretrained models by [21]

Dataset	Accuracy	
	Unseen	Seen
SUN	40.3472	35.1163
AWA2	58.5871	79.4288
CUB	64.4873	70.6856

Table 2: Calibration metrics on seen classes. All metrics are calculated with 10 bins.

Dataset	ECE	MCE	SCE	ACE
SUN	0.0646	0.7060	31.9606	0.0724
AWA2	0.2030	0.6240	2.7877	0.2108
CUB	0.0275	0.8260	8.2269	0.0442

Table 3: Calibration metrics on unseen classes. All metrics are calculated with 10 bins.

Dataset	ECE	MCE	SCE	ACE
SUN	0.0329	0.6357	1.3612	0.0340
AWA2	0.0214	0.6502	0.4179	0.0498
CUB	0.0831	0.8202	2.2845	0.0832

are expected to reflect the true likelihood of an event occuring. Furthermore R denotes the number of ranges.

Out-of-Distribution (OOD) Detection: We evaluate the OOD detection capability of our model by measuring its accuracy in identifying OOD samples. We consider samples from one dataset as in-distribution and samples from another datasets as OOD. For example, when evaluating on the AWA2 dataset, we consider AWA2 as in-distribution and SUN as OOD. We calculate the OOD detection accuracy as the proportion of correct OOD/in-distribution classifications.

4. Results & Discussion

4.1. Uncertainty Metrics

Let us delve into the evaluation of the uncertainty estimation by analyzing calibration metrics. Our goal is to show how we capture uncertainty estimation with our confidence score and give the reader an impression on how results across the three datasets CUB, AWA2, and SUN can look like. For CUB 312 , for AWA2 85 and for SUN 102 attribute prototypes are detected.

As per Table 1, the baseline accuracies for Generalized Zero-Shot Learning (GZSL) obtained from pretrained models [21] show varying results across datasets. For the unseen case, SUN exhibits the lowest accuracy of 40.35%, while AWA2 and CUB have higher accuracies of 58.59% and 64.49%, respectively. When examining the seen classes, AWA2 shows a distinct increase in accuracy to 79.43%, while the CUB dataset maintains a relatively high accuracy of 70.69%, and the accuracy for SUN is lower with 35.12%. These results serve as our performance baseline and match the results reported in [21]. The strong differences in accuracy are an advantage for our evaluation as it allows us to see how well our uncertainty method performs with attribute prototype predictors of varying strength.

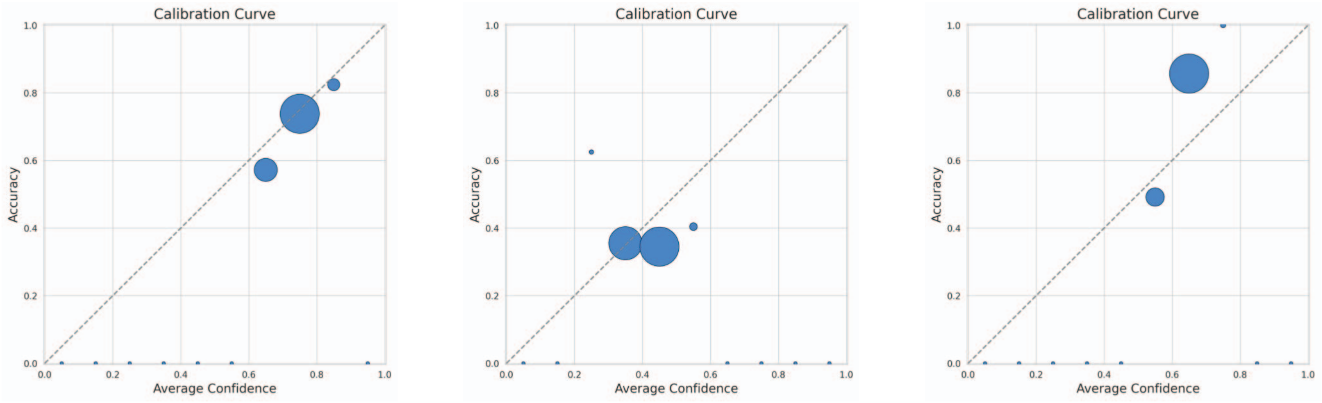
To assess the calibration quality of the novel uncertainty estimation method, we use four metrics: Expected Calibration Error (ECE), Maximum Calibration Error (MCE), Static Calibration Error (SCE), and Adaptive Calibration Error (ACE). A definition of these metrics is given in Section 3.3. Calibration refers to the agreement between predicted uncertainties and true outcomes. Ideally, for predictions made with a confidence of $x\%$, the proportion of correct predictions should be close to $x\%$.

Table 2 and Table 3 show the calibration metrics for the seen and unseen classes. For the seen classes, the ECE scores vary across the datasets: they are relatively low for SUN and CUB, suggesting that the confidence levels of the model for these datasets are reasonably consistent with its accuracy. However, the AWA2 dataset shows a higher ECE score, indicating some discrepancy between the model's confidence and its accuracy. The MCE scores highlight the presence of miscalibration under some circumstances, as indicated by relatively high maximum calibration errors, except for the AWA2 dataset which has the lowest MCE scores. Which means that while the average error is high, the maximum error is lower. This underlines the necessity for a combination of calibration metrics.

When we look at the unseen classes (Table 3), we notice that the ECE scores for the SUN and CUB dataset remain low, similar to the seen case. The MCE score are slightly lower in this scenario, indicating that the model's worst-case calibration is better for unseen data. For the AWA2 it is the opposite, the ECE score are much lower and the MCE score remains on the same level. Indicating that the calibration is better on average, but the maximum calibration error remains the same.

However, when we look at the SCE, a more holistic calibration measure accounting for the calibration of individual prediction scores, we observe high values in both seen and unseen cases, especially for the SUN and CUB datasets in the seen classes. High SCE scores may suggest that the model's miscalibration is pervasive across all confidence levels, not just the extremes, which shows potential for using calibration methods such as Temperature scaling [31] or Isotonic Regression [35].

Finally, the ACE scores by way of adaptive interval bin-

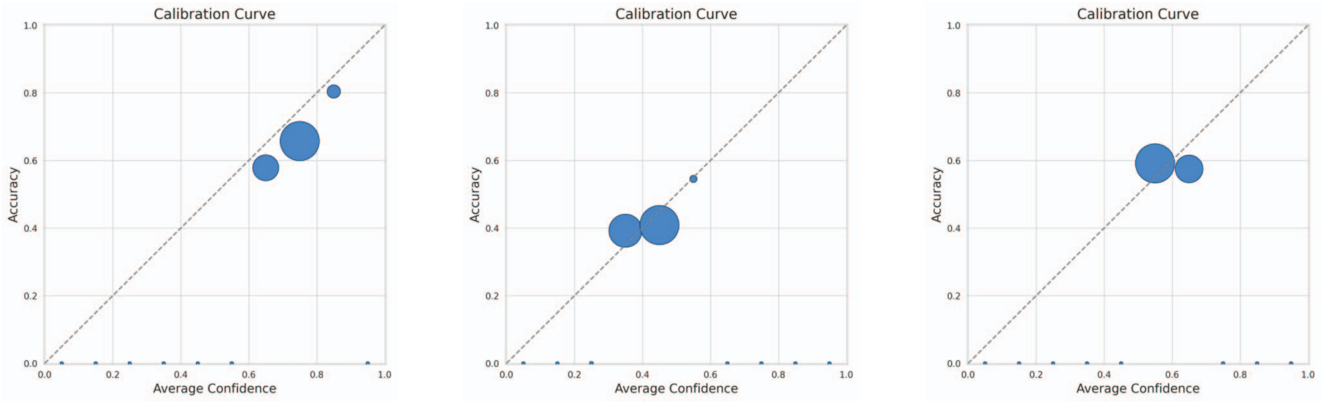


(a) CUB

(b) SUN

(c) AWA2

Figure 2: Calibration curves for seen classes with GZSL models. The size of the dots indicates the number of datapoints in the respective bin.



(a) CUB

(b) SUN

(c) AWA2

Figure 3: Calibration curves for unseen classes with GZSL models. The size of the dots indicates the number of datapoints in the respective bin.

ning provide a different view on the prediction scores, focusing on the majority of predictions.

They remain quite low in both seen and unseen cases for the CUB and SUN dataset. For the AWA2 dataset it shown the same trend as in the ECE scores. This suggests that the model's calibration quality is robust to such a change in interpretation.

In conclusion, the novel uncertainty estimation method generally demonstrates adequate calibration in terms of ECE and ACE. Furthermore, the method shows room for improvement in dealing with worst-case scenarios (MCE) and in maintaining a consistent calibration across all confidence levels (SCE).

For a visual confirmation we offer the calibration curves which show accuracy vs. average confidence in 10 bins in Figure 2 and 3. For the CUB dataset we can confirm the very good calibration, the blue dots are almost on the perfect calibration line confidence matching accuracy, while for the SUN and AWA2 dataset we see still good values and but not as good calibration. The large difference in ECE scores between seen and unseen classes for the AWA2 dataset is clearly visible. We suspect that the generally higher accuracy for the CUB dataset also lead to better attribute prototypes.

4.2. Out-of-Distribution Detection

In Figure 4 we show the cumulative distribution of the confidence values of the GZSL trained on the AWA2 dataset applied to the AWA2 dataset. For these in-distribution data points, we see a clear S-curve of values between 0.55 and 0.70. In Figure 5 we show the previous AWA2 model applied to the SUN dataset, thus the OOD case. Again we

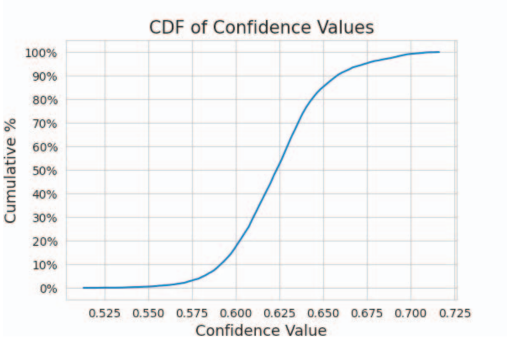


Figure 4: CDF of the confidence values of a AWA2 model applied on AWA2 data (seen classes)

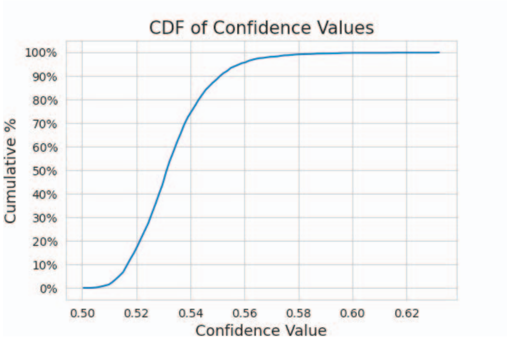


Figure 5: CDF of the confidence values of a AWA2 model applied on SUN data (seen classes)

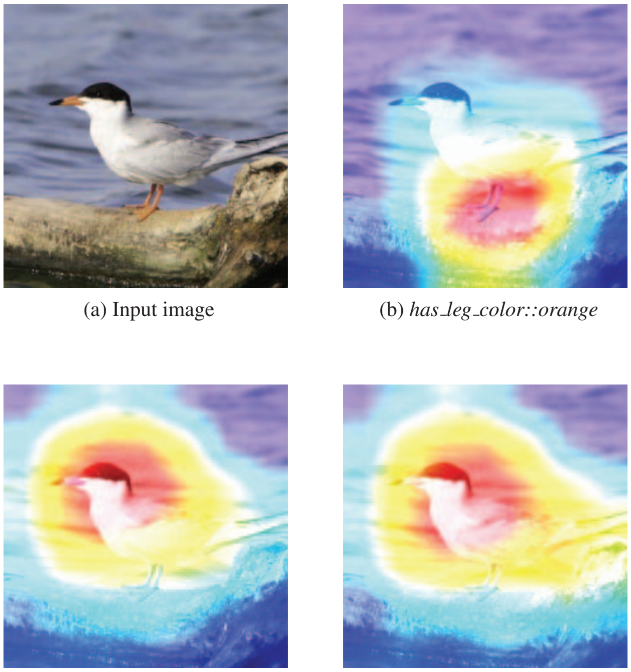
see an S-curve, however this time between 0.50 and 0.58. There is only a very small overlap of the curves between the CDFs in Figure 4 and Figure 5. Thus, we can set a threshold value $\tau = 0.57$ and get an almost perfect separation of OOD and non-OOD samples, which results in ca. 97% OOD detection accuracy. For the SUN/CUB model/dataset combinations we observed a similar separation.

Overall we can conclude that our method of directly estimating the confidence from the distance to attribute prototypes is an effective method for OOD detection.

5. Explainability

In Eq. 3, we calculate the proximity using the maximum proximity over all attribute prototypes. This calculation serves to enhance the explainability of the uncertainty or confidence. To do this, we first analyze which individual attribute prototype is closest to our image.

Consider an example image of a bird called Forster's tern from the CUB dataset (Table 4). The image is classified as a 'Forsters_Tern' (class 146) using the GZSL model by [21] with a confidence of 0.7876.



(a) Input image

(b) *has_leg_color::orange*

(c) *has_head_pattern::capped*

(d) *has_underparts_color::white*

Figure 6: Confidence scores as a normalized heatmap for TOP 3 closest prototype attributes (see Table 4) for an image of a Forster's tern bird from the CUB dataset.

Next, we calculate the proximity for each of these closest attributes across each spatial dimension of the feature vector. Afterward, this proximity is converted into a confidence score, ranging from 0 to 1. Subsequently, we overlay this mask onto the original image, adjusting the color scaling for clarity as shown in Figures 6b to 6d.

As we can observe in Figure 6, the determined attributes align with the visual aspects of the image and are entirely appropriate for this bird species. We can see that the network correctly identifies the bird's orange leg color and the capped head. This means that attribute prototypes provide a method for interpretable results and with our method this transfers to an interpretable uncertainty.

Table 4: Attributes and Confidence for an image of a Forster's tern from the CUB dataset in Figure 6a.

Attribute Name	Confidence
has_leg_color::orange	0.7876
has_head_pattern::capped	0.6938
has_underparts_color::white	0.6792
has_belly_color::white	0.6755

6. Conclusion

The results show that the attribute prototypes can be used for uncertainty estimation, but there are also limitations to this approach. We want to conclude this article by listing these advantages and limitations.
Advantages:

- **Interpretability and explainability**: The proximity measure directly correlates to how similar the input is to known prototypes, making it relatively easy to interpret compared to some other uncertainty measures like ENN where we receive only the uncertainty.

- **Straightforward**: As this approach is integrated within a deep learning framework, it scales well with high dimensional data and large datasets. It is direct and straightforward. If attribute prototype information is available it can be easily evaluated. Compared to other approaches such as Ensemble or MC methods that require multiple models and or multiple evaluations this is a considerable advantage.

- **OOD detection**: This method allows not only for uncertainty estimation but also for OOD detection by setting a threshold on the proximity value.

- **Different working principle**: The working principle of our method is different to other uncertainty estimation methods that relying on logits, stochastic processes, ensembles etc. This means that a combination with these methods is plausible in future work.

Limitations:

- **Threshold dependence**: The choice of the threshold for OOD detection could be problem-dependent and may require careful tuning.

- **Prototype dependence**: The effectiveness of this method heavily relies on how well the prototypes can capture the characteristics of the classes. If the prototypes are not representative enough, the proximity measure may not effectively estimate uncertainty.

- **Competitiveness needs validation**: Our method shows good results on three datasets. Nevertheless, we believe that comparing it with other non-APN methods will be valuable. Hence we would like to compare with other methods [18] and [20] as well as OOD-detection algorithms [36] by adapting them to attribute prototype data in future work.

Because of these advantages we believe our novel method will offer a new perspective towards more interpretable deep learning applications of uncertainty estimation.

References

[1] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When gaussian process meets big data: A review of scalable gps,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[2] R. M. Neal, “Bayesian learning for neural networks,” *Springer Science & Business Media*, 2012.

[3] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[4] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman, “Pyro: Deep universal probabilistic programming,” *J. Mach. Learn. Res.*, vol. 20, pp. 28:1–28:6, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-403.html

[5] J. Swiatkowski, K. Roth, B. Veeling, L. Tran, J. Dillon, J. Snoek, S. Mandt, T. Salimans, R. Jenatton, and S. Nowozin, “The k-tied normal distribution: A compact parameterization of gaussian mean field posteriors in bayesian neural networks,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 9289–9299.

[6] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html

[7] Y. Chen, H. Chang, J. Meng, and D. Zhang, “Ensemble neural networks (enn): A gradient-free stochastic method,” *Neural Networks*, vol. 110, pp. 170–185, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608018303319

[8] A. N. Angelopoulos and S. Bates, “A gentle introduction to conformal prediction and distribution-free uncertainty quantification,” *CoRR*, vol. abs/2107.07511, 2021. [Online]. Available: https://arxiv.org/abs/2107.07511

[9] S. Bates, E. Candès, L. Lei, Y. Romano, and M. Sesia, “Testing for outliers with conformal p-values,” *The Annals of Statistics*, vol. 51, no. 1, pp. 149–178, 2023.

[10] A. N. Angelopoulos, S. Bates, M. I. Jordan, and J. Malik, “Uncertainty sets for image classifiers using conformal prediction,” in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=eNdiU_DbM9

[11] B. Kumar, C. Lu, G. Gupta, A. Palepu, D. Bellamy, R. Raskar, and A. Beam, “Conformal prediction with large language models for multi-choice question answering,” *arXiv preprint arXiv:2305.18404*, 2023.

[12] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection: A survey,” *arXiv preprint arXiv:2110.11334*, 2021.

[13] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Information fusion*, vol. 76, pp. 243–297, 2021.

[14] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, “Robust classification with convolutional prototype learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3474–3482.

[15] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, “This looks like that: deep learning for interpretable image recognition,” *Advances in neural information processing systems*, vol. 32, 2019.

[16] M. Nauta, A. Jutte, J. C. Provoost, and C. Seifert, “This looks like that, because ... explaining prototypes for interpretable image recognition,” in *PKDD/ECML Workshops*, 2020.

[17] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, “Rethinking semantic segmentation: A prototype view,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2582–2593.

[18] J. R. van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in *International conference on machine learning*. PMLR, 2020, pp. 9690–9700. [Online]. Available: https://api.semanticscholar.org/CorpusID:211987967

[19] S. Padhy, Z. Nado, J. J. Ren, J. Z. Liu, J. Snoek, and B. Lakshminarayanan, “Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks,” *ArXiv*, vol. abs/2007.05134, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220486366

[20] G. Franchi, X. Yu, A. Bursuc, E. Aldea, S. Dubuisson, and D. Filliat, “Latent discriminant deterministic uncertainty,” in *European Conference on Computer Vision*. Cham: Springer Nature Switzerland, 2022, pp. 243–260. [Online]. Available: https://api.semanticscholar.org/CorpusID:250921065

[21] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, “Attribute prototype network for zero-shot learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 969–21 980, 2020.

[22] ——, “Attribute prototype network for any-shot learning,” *International Journal of Computer Vision*, vol. 130, pp. 1735 – 1753, 2022.

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-ucsd birds-200-2011 dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[24] G. Patterson, C. Xu, H. Su, and J. Hays, “The sun attribute database: Beyond categories for deeper scene understanding,” *International Journal of Computer Vision*, vol. 108, pp. 59–81, 2014.

[25] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 951–958.

[26] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2019.

[27] Y. Xian, B. Schiele, and Z. Akata, “Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4582–4591.

[28] W.-L. Chao, H. Liu, B. Peng, K. Saenko, and T. Darrell, “An empirical study and analysis of generalized zero-shot learning for object recognition in the wild,” in *Proceedings of the European conference on computer vision*, 2016, pp. 52–68.

[29] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, “Transductive multi-view zero-shot learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2332–2345, 2015.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[32] I. H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions,” *SN Computer Science*, vol. 2, no. 9, pp. 1–35, 2021.

[33] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, “Measuring calibration in deep learning.” in *CVPR workshops*, vol. 2, no. 7, 2019.

[34] H. Gweon and H. Yu, “How reliable is your reliability diagram?” *Pattern Recognition Letters*, vol. 125, pp. 687–693, 2019.

[35] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.

[36] O. Dinari and O. Freifeld, “Variational- and metric-based deep latent space for out-of-distribution detection,” in *Conference on Uncertainty in Artificial Intelligence*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252898943