Condensed Dataset Representations via Latent Prototypical Features

Korbinian Franz Rudolf[1]

[1]Institut fuer Technik der Informationsverarbeitung (ITIV), Karlsruhe Institute of Technologie (KIT), Kaiserstraße 12, 76131 Karlsruhe, Federal Republic of Germany

Abstract

Artificial Intelligence (AI) in cyber-physical systems lacks the credibility that deterministic algorithmic solutions offer. The input-output connection in classical approaches is not as clearly interpretable in most AI systems. To increase model credibility, we propose leveraging the bottleneck created by Prototypical Neural Networks (PNNs) to quantify a model's epistemic uncertainty for out-of-distribution, domain-shift, and bias detection. Through these lenses, we hypothesize that we can gain insight into the neural network's feature distribution. We therefore aim to develop prototypes that serve as traces of the dataset in the latent space, enabling us to analyze the dataset indirectly. To this end, we propose a modular evaluation that leverages latent representations (i.e., the prototypes) to analyze the network's explanations. This proposal outlines the motivation and the theoretical development of the proposed process. Furthermore, we discuss the research questions and the next steps for the dissertation.

Keywords

Prototypical Neural Networks, Epistemic Uncertainty, Credibility, Out-Of-Distribution Detection

1. Introduction and Problem Setting

The ever-increasing use of Artificial Intelligence (AI) systems in everyday activities and in cyber-physical systems, such as software-defined vehicles (SDV), also concerns functionality that influences decision-making processes[1]. Explainability of the inference increases credibility and supports the diagnosis of faults that occur during deployment. Fogg et al.[2] view credibility as a combination of the model's perceived trustworthiness and expertise. Trustworthiness is a subjective, context-dependent attribute. Perceived expertise is the extent to which a model appears knowledgeable in a domain. One way to increase the model's perceived expertise is to increase transparency and to explain its inferences [3].

A model's performance is constrained by the chosen loss function, architecture, including the optimizer, and training dataset. The architecture defines the hypothesis space, while the training dataset influences the optimization towards a hypothesis that minimizes the empirical loss. The training dataset comprises assumptions about the structure of the Operational Design Domain (ODD), the domain in which the model is deployed, and the relationships within it that are relevant to the task. For a fixed model architecture and loss function (i.e., a fixed hypothesis space), the training dataset sets the domain within which the model can learn and therefore constrains the model's expertise. Under this view of expertise, three perspectives open up for analyzing and demonstrating a model's expertise: what information caused the inference result, which features from the dataset are represented by the model, and whether the model has sufficient information about the source domain of a given sample to produce an adequate inference. While the first perspective is a core question in Explainable Artificial Intelligence (xAI), i.e., feature importance [4] and representation learning[5], the second perspective examines how a model, e.g., a neural network, learns and stores information. The number of parameters and the interconnections of neurons in a neural network make the direct analysis of feature importance and data extraction difficult and computationally expensive. Directly viewing the output of, e.g., a feature

Late-breaking work, Demos and Doctoral Consortium, colocated with the 4th World Conference on eXplainable Artificial Intelligence: July 01–03, 2026, Fortaleza, Brazil

korbinian.rudolf@kit.edu (K. F. Rudolf)
www.itiv.kit.edu/21_10753.php (K. F. Rudolf)
0009-0009-0804-7712 (K. F. Rudolf)

encoder and mapping the latent space structure can reduce the effort required for analysis. However, this approach requires a significant number of samples to be embedded in the latent space for mapping. This entails the risk of interpreting apparent correlations as causal relationships.

Prototypical Neural Networks (PNNs), e.g., ProtoPNet[6], create a bottleneck in the processing of the input by defining cluster centers in the latent space, to which the input embeddings are compared. The assignment, measured by a distance or similarity metric, equals the activation of the prototype with respect to the input. The following inference relies on prototype activation, thereby restricting the information passed to it. This ensures a structural dependency in the computational graph between the prototype activation and the proposed solution to the task. The advantage of this procedure lies in reducing the space the analysis must consider. Instead of creating a map of the latent space by analyzing the embeddings of an input and filling the space element-wise, the prototypes are a limited set of latent representations. The analysis of the prototypes provides insights into the model's learned features and supports understanding of what caused the inference result, thus offering insights for the first two questions concerning the expertise of a neural network.

The third perspective on expertise concerns the epistemic uncertainty of the prediction. Epistemic uncertainty in models is a research direction in its own right[7] and offers a solution to increase a model's perceived expertise by adding an existing approach to a PNN. However, introducing a second addendum, especially with learned parameters, requires it to learn its own statistics and activations to base the estimation on. To avoid adding more features or processes to the model, we propose reusing the feature distribution represented by the prototypes to infer information about epistemic uncertainty. Specifically, the proposed dissertation uses the prototypes to analyze the feature distribution of the training dataset and compare it with samples from the ODD to detect out-of-distribution (OOD) samples, domain shifts, and bias. We interpret these problems as instances of epistemic uncertainty. OOD detection analyzes whether the learned feature distribution covers an input sample [8]. Otherwise, the model lacks information about the feature distribution from which the input sample is drawn, which implies a high epistemic uncertainty. OOD detection is, therefore, a binarization of the epistemic uncertainty into "in-distribution" and "out-of-distribution"[9]. Domain Shift refers to a divergence between the training data and the ODD[10, 11]. In detail, for a subset of the set of input data, the elements of the subset are out of the distribution of the learned feature distribution. Using the same arguments as for OOD detection, domain shift characterizes epistemic uncertainty for a set of inputs.

(Data) Bias is closely related to these problems, especially to subpopulation domain shift[11]. The difference between domain shift and data bias is the reason for the mismatch between the training dataset and the operational domain. While domain shift refers to a change in the operational domain, data bias arises from a misrepresentation of a feature in the training dataset, leading to a mismatch between the training dataset and the ODD[12]. High epistemic uncertainty about an input the model is biased against may result from this underrepresentation.

The dissertation aims to contribute to the state of the art of increasing a model's perceived expertise by

- viewing the prototypes as a trace: a representation of the training dataset, which changes with the data distribution of the dataset,
- utilizing the trace to detect mismatches in the feature distribution as indicators for potential failures of the machine learning model, and
- analyzing the influence of the model architecture on the learned feature distribution.

While the first and last contributions are of theoretical interest, the second aims to support the practical development of safer SDVs. In such systems, innovation increasingly occurs in software, while hardware is reduced to its essential functionality [13]. The proposed approach contributes to this goal by providing a method to validate machine learning components by analyzing epistemic uncertainty during inference, thereby supporting the reliability and safety of AI-based functions.

2. Related Work

Uncertainty estimation for neural networks regularly relies on approximating Bayesian Neural Networks through sampling weights or modeling them as probability distributions[14, 15, 16]. Qian et al.[17] introduce uncertainty measures to a self-explaining model, a Concept Bottleneck Model, by assuming that the intrinsic explanation's value indicates the certainty of the model for that explanation. From this assumption and the model's activation, they derive a non-conformity measure for the explanation based on the Sigmoid Function. The uncertainty is indicated by the size of the top-K explanation set with the best non-conformity measure. While this outlines the epistemic uncertainty, the approach relies on the conformity measure and the activation value of the detected concepts, effectively analyzing the outputs of a multi-label concept-classification network. Instead of defining a single measure to capture epistemic uncertainty, we view it as a multifaceted problem, i.e., as a set of related problems that arise during inference, akin to [9], and aim to identify missing or performance-degrading data sources.

Oh et al.[18] leverage the relationship between epistemic uncertainty and OOD detection by deducing that a sample is OOD from high uncertainty. Their approach measures epistemic uncertainty using an existing method and determines whether a sample is OOD if the uncertainty exceeds a threshold. In contrast, Sun et al.[19] follow a distance-based approach and apply k-Nearest Neighbor in the latent space. If the distance of an embedding to the k-th nearest neighbor is greater than a threshold, the sample is classified as OOD. This relies on detecting clusters in a previously trained latent space. We propose using end-to-end optimized PNNs to enforce clustering around learned prototype centers during training. These prototypes are causal to the inference result and are trained to represent discriminative features, enabling a combination of OOD detection with intrinsic interpretability.

For aligning the learned features of a model to a new domain, Deep CORAL[10] proposes a loss function that minimizes the distance between the covariances of the target domain's features and the source domain's features. The loss alleviates a domain shift but does not detect it. Domain shift detection is related to bias detection: a specific feature or region of the data distribution can degrade performance during inference due to under- or overrepresentation. Anders et al. [20] propose a method for detecting and removing biases in the network by localizing the feature in question in the latent space and fitting its effect on classification, thereby demonstrating effective detection of biases in the latent space, but still requiring a mapping of the latent space.

In the area of Prototypical Neural Networks, Vadillo et al.[21] approach uncertainty estimation by defining one prototype per class as a probability distribution, from which they derive the uncertainty. Wu et al. [22] follow a similar approach: they calculate the cluster centers as the mean of the support sample, but sample the uncertainty from a predefined linear uncertainty distribution. Both approaches assume distributions from which they derive uncertainty to obtain a single value for epistemic uncertainty. We propose outlining the model's uncertainty through the listed perspectives to gain a more holistic view of its performance. While this does not return a single value, it provides insight into the model's capabilities without assuming a family of distributions for the uncertainty.

An alternative approach to PNNs for latent space analysis is linear probing[23] and the related concept of Concept Activation Vectors[24]. These approaches train a linear layer to infer the class or predefined concepts from latent-space embeddings, interpreting classification performance as an indication of whether the embedding contains the relevant information. As a post hoc analysis, this approach is susceptible to learning the same spurious correlations as the model under analysis, thereby providing no independent validation [25]. The proposed dissertation instead relies on prototypes that represent clusters of embeddings, serving as inference-time indicators of feature activation and forced to learn a distribution over the embedded features, rather than a linear separation.

3. Research Questions and Hypothesis

We view prototypes as a means of inferring the feature distribution of the training dataset, given the model's architecture and loss function. To that end, we propose to interpret the prototypes as a trace of

the dataset. This trace is intended to stably condense the dataset's feature distribution under consistent conditions while meaningfully varying with distributional changes. The trace, based on the concept of the matrix trace, is intended to serve as a dataset-level invariant and to provide insights into a dataset's composition after embedding in the latent space. This view is based on the hypothesis that prototypes reflect the feature distribution of the training dataset, conditioned on the task the model is trained to solve, the encoder, and the loss function used.

The dataset embodies the assumptions made during its creation. In ideal dataset construction, each addition of a data sample implies that the sample represents relevant information or a relationship within the operational domain. Therefore, the dataset induces assumptions about the coverage of the operational domain. The second hypothesis of this thesis is that the prototypes, i.e., the dataset's trace, enable evaluation of whether these assumptions hold. If these assumptions do not hold, the dataset does not accurately represent the features that the model will encounter and rely on during deployment. Accurately refers to the appearance of features under similar conditions and frequency in both the training and deployment domains. For example, if a model for pedestrian detection underrepresents children, the detection rate of children significantly decreases[26]. In this case, the model has learned a bias that reduces performance in the operational domain, thereby increasing the risk of traffic accidents.

A third hypothesis driving this research is that an unsupervised learning setting increases the prototypes' ability to reflect the dataset's feature distribution, independent of labels. Specifically, features may be learned as irrelevant in the context of classification because they lack predictive value.

3.1. Research Questions

Validating the hypothesis requires answering two central questions: first, whether prototypes can serve as trace characteristics that provide interpretable links between network activations and the dataset's feature distribution; and second, how post hoc analysis of prototype explanations can yield reliable information about epistemic uncertainty. These questions are closely related to the challenges of domain shift, OOD, and bias detection, and we interpret these challenges as potential failures arising from high epistemic uncertainty.

Several concrete questions arise from the two broader questions. The first question asks how the prototypes must be defined concretely to be usable as a trace (RQ1). Furthermore, the trace's expressiveness depends on the distribution of the represented features across the prototypes. Therefore, we ask whether the prototypes reflect the feature distribution in the dataset (RQ1.1) and how changes in feature distribution and initialization parameters affect the prototypes' semantics (RQ1.2). This additional question is important for understanding whether the trace is an invariant of the feature distribution. However, these changes in the representations must reflect and respond to changes in the feature distribution while remaining semantically consistent (RQ1.3). The second question concerns the usefulness of the prototype-based trace. To be useful, the trace must predict and model the distributional fit of input samples. Additionally, it must match the performance of existing OOD, domain-shift, and bias detection methods (RQ2). This stability of the representations extends to changes in the model's architecture. Given a fixed feature distribution in a dataset, we examine how changes to the model's encoder and task-specific head affect the learned representations (RQ3). In this context, we examine whether unsupervised learning increases the prototype's ability to reflect the dataset's feature distribution. In summary, the questions are:

RQ1. How must prototypes be defined to act as a trace for the dataset in the context of the model and the loss function?

- RQ1.1 Does the feature distribution learned by prototypes reflect the feature distribution of the training dataset?
- RQ1.2 How sensitive are prototype semantics to changes in feature distribution and initialization?
- RQ1.3 Do prototypes remain semantically consistent while reflecting actual distributional changes?

RQ2. Can a prototype-based trace match the performance of the current state of the art OOD, domain shift, and bias detectors?

RQ3. To what extent are trace-based conclusions consistent across different model architectures?

RQ3.1 Does unsupervised learning increase the prototypes' capabilities to reflect the feature distribution of the training dataset?

RQ1 and RQ3 are evaluated in parallel, as prototype definitions and model architectures are mutually dependent. The evaluation considers multiple prototype definitions and architectures, assessing both explanation quality, following the evaluation protocol of ProtoPNet[6], and trace quality. Trace quality is measured by the similarity between the training feature distribution and the prototype-induced distribution in the latent space, using metrics such as Maximum Mean Discrepancy (MMD) and quantization error. The evaluation is performed on CUB-200-2011[27] due to its broad feature spectrum and established use in the prototype community. The trace is considered effective in representing the feature distribution when the distributional fit between the embedded samples and the distribution induced by the prototypes changes significantly as the embedded features change. Furthermore, the features represented by the prototypes must remain consistent across runs, as validated by analyzing the prototypes' activations on the labeled attributes of the CUB-200-2011 dataset.

RQ2 evaluates the proposed trace-based detection methods against state-of-the-art approaches for OOD, domain shift, and bias detection. The evaluation follows established benchmarking protocols and datasets, using OpenOOD v1.5[28] for OOD detection, ACDC[29] for domain shift, and CelebA[30] and KITTI[31] for bias detection.

4. Preliminary Results

The research is currently transitioning from the conceptual design and literature review to practical implementation of models and experiments in the context of SDVs. PNNs were chosen as the interpretable model architecture, in contrast to, e.g., the related architecture of Concept Bottleneck Models[32], for their explicitly exposed learned representations. For the architecture in which we embed the prototypes, we chose an autoencoder to learn high-level features that can be applied, e.g., to classification. The prototypes are inserted into the autoencoder's latent space, which serves as a bottleneck. We examine whether the absence of the need to learn relationships between inputs and labels is advantageous for representing a more comprehensive feature distribution of the training dataset, compared with a latent space trained for classification.

Based on these architectural and task choices, we define the proposed process (see Fig. 1) as a two-part system to provide diagnostic signals about the model's performance. The approach combines an interpretable neural network, i.e., a prototypical neural network, with an evaluation process that analyzes its explanations. An encoder embeds the input data, either from the training dataset or the operational domain, into the latent space. The embeddings are compared and assigned to prototypes. The comparison result is passed to the task-specific head, i.e., a decoder, which computes the inference result. The evaluation process relies on the prototypes. This entails OOD detection, domain shift detection, and bias detection. The purpose of this evaluation component is to generate further insights into the network's reliability through the lens of uncertainty. The prototypes provide explanations for the inference, from which the evaluation process infers the model's uncertainty and the dataset's composition. These evaluation elements will be developed and integrated into a modular testing process.

5. Next Steps and Expected Contributions

A key contribution of the proposed dissertation is the development of a separate evaluation process for prototype-based explanations, using a trace interpretation of prototypes to assess distributional fit and epistemic uncertainty from multiple perspectives. Concretely, this process evaluates the distributional

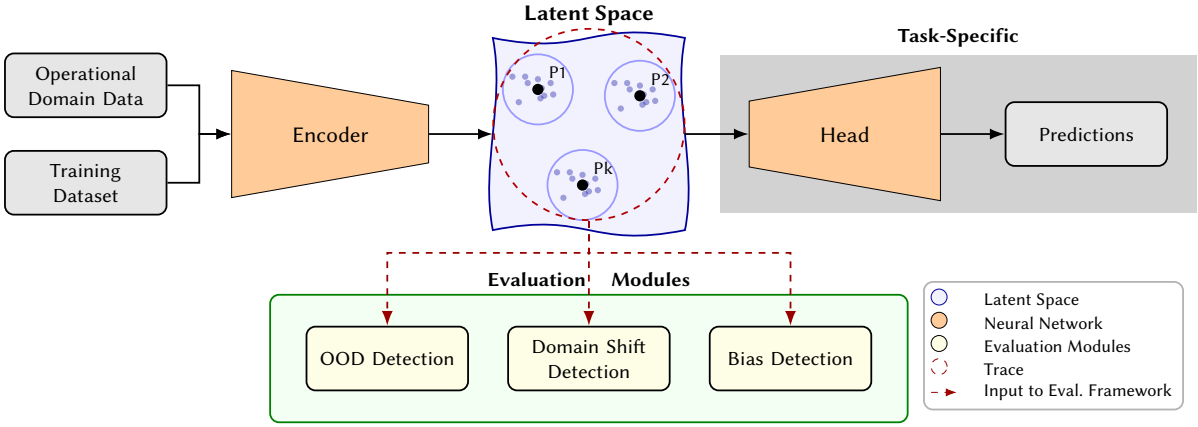

Figure 1: Overview of the model architecture and evaluation process. The encoder embeds the input data, originating either from the training dataset or the operational domain, into a latent space. The similarity between the embedded features and the learned prototypes determines how samples are assigned to clusters. During evaluation, each analysis module (domain shift, out-of-distribution, and bias detection) uses these assignments and prototype similarity scores (i.e., the dataset trace) to evaluate its respective task. To preserve a causal link between explanation and inference, the task-specific head processes the prototype–embedding similarities to produce the final predictions.

match between the training dataset and the ODD through self-contained modules for OOD detection, domain shift detection, and bias detection, developed in this order of priority. The dissertation is structured into three phases: defining the trace and model architecture, independently developing and evaluating each module, and integrating them into a complete system benchmarked against existing baselines.

The theoretical goals are architecture- and domain-agnostic: treating data sources and model components as interchangeable allows the results to transfer across domains by exchanging the data source. In the SDV context, the resulting diagnostic signals support both acute in-operation monitoring and post-hoc analysis of traffic incidents. Beyond the benchmarks in RQ2, we additionally evaluate on the KITTI dataset[31] and the nuScenes dataset[33] to assess applicability to perception tasks; the derived deficiency indicators are expected to inform dataset extension[34] or the closing of the simulation-to-reality gap[35].

In the following, the dissertation will implement an autoencoder with prototypes in the latent space and define suitable losses for use in the unsupervised learning setting. On this basis, an interpretability evaluation will be performed to assess the usability of the prototypes. The evaluation process can then be developed based on the autoencoder. Each module of the evaluation process is developed and evaluated independently to achieve a modular process. .

Acknowledgments

The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the InnovationCampus Future Mobility (ICM).

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT (GPT-5) and Grammarly in order to: Grammar and spelling check. After using these tools/services, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] Y. Li, Z. Huang, Basics and Applications of AI in ADAS and Autonomous Vehicles, in: Y. Li, H. Shi (Eds.), Advanced Driver Assistance Systems and Autonomous Vehicles: From Fundamentals to Applications, Springer Nature, Singapore, 2022, pp. 17–48. doi:10.1007/978-981-19-5053-7_2.

[2] B. J. Fogg, H. Tseng, The elements of computer credibility, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '99, Association for Computing Machinery, New York, NY, USA, 1999, pp. 80–87. doi:10.1145/302979.303001.

[3] L. Jantzen, M. Philipp, N. Tagalidou, M. Bui, R. Kempen, Trust Me, I'm Transparent: Describing AI Systems Using Global Explanations, International Journal of Human–Computer Interaction 0 (2025) 1–23. doi:10.1080/10447318.2025.2606953.

[4] T. Speith, A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 2239–2250. doi:10.1145/3531146.3534639.

[5] Y. Bengio, A. Courville, P. Vincent, Representation Learning: A Review and New Perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1798–1828. doi:10.1109/TPAMI.2013.50.

[6] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, C. Rudin, This looks like that: Deep learning for interpretable image recognition, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 801, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 8930–8941.

[7] W. He, Z. Jiang, T. Xiao, Z. Xu, Y. Li, A Survey on Uncertainty Quantification Methods for Deep Learning, ACM Comput. Surv. 58 (2026) 179:1–179:35. doi:10.1145/3786319.

[8] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, S. Vernekar, Improving Reconstruction Autoencoder Out-of-distribution Detection with Mahalanobis Distance, https://arxiv.org/abs/1812.02765v1, 2018.

[9] B. Mucsányi, M. Kirchhof, S. J. Oh, Benchmarking Uncertainty Disentanglement: Specialized Uncertainties for Specialized Tasks, Advances in Neural Information Processing Systems 37 (2024) 50972–51038. doi:10.52202/079017-1614.

[10] B. Sun, K. Saenko, Deep CORAL: Correlation Alignment for Deep Domain Adaptation, in: G. Hua, H. Jégou (Eds.), Computer Vision – ECCV 2016 Workshops, Springer International Publishing, Cham, 2016, pp. 443–450. doi:10.1007/978-3-319-49409-8_35.

[11] S. Santurkar, D. Tsipras, A. Madry, BREEDS: Benchmarks for Subpopulation Shift, in: International Conference on Learning Representations, 2020.

[12] K. Mavrogiorgos, A. Kiourtis, A. Mavrogiorgou, A. Menychtas, D. Kyriazis, Bias in Machine Learning: A Literature Review, Applied Sciences 14 (2024). doi:10.3390/app14198860.

[13] Z. Liu, W. Zhang, F. Zhao, Impact, Challenges and Prospect of Software-Defined Vehicles, Automotive Innovation 5 (2022) 180–194. doi:10.1007/s42154-022-00179-z.

[14] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, in: Proceedings of the 32nd International Conference on Machine Learning - Volume 37, volume 37 of *ICML'15*, JMLR.org, Lille, France, 2015, pp. 1613–1622.

[15] Y. Gal, Z. Ghahramani, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, in: Proceedings of The 33rd International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.

[16] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, in: Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[17] W. Qian, C. Zhao, Y. Li, F. Ma, C. Zhang, M. Huai, Towards Modeling Uncertainties of Self-Explaining Neural Networks via Conformal Prediction, Proceedings of the AAAI Conference on Artificial Intelligence 38 (2024) 14651–14659. doi:10.1609/aaai.v38i13.29382.

[18] D. Oh, D. Ji, O. Kwon, Y. Hyun, Boosting Out-of-Distribution Image Detection With Epistemic Uncertainty, IEEE Access 10 (2022) 109289–109298. doi:10.1109/ACCESS.2022.3213667.
[19] Y. Sun, Y. Ming, X. Zhu, Y. Li, Out-of-Distribution Detection with Deep Nearest Neighbors, arXiv, 2022. doi:10.48550/ARXIV.2204.06507.
[20] C. J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, S. Lapuschkin, Finding and removing Clever Hans: Using explanation methods to debug and improve deep models, Information Fusion 77 (2022) 261–295. doi:10.1016/j.inffus.2021.07.015.
[21] J. Vadillo, R. Santana, J. A. Lozano, M. Kwiatkowska, Uncertainty-Aware Explanations Through Probabilistic Self-Explainable Neural Networks, 2025. doi:10.48550/arXiv.2403.13740. arXiv:2403.13740.
[22] W. Wu, T. Zou, D. Guo, L. Zhang, K. Wang, X. Li, Fault Diagnosis for China Space Station Circulating Pumps: Prototypical Network with Uncertainty Theory, Symmetry 15 (2023) 903. doi:10.3390/sym15040903.
[23] G. Alain, Y. Bengio, Understanding intermediate layers using linear classifier probes, 2017. URL: https://openreview.net/forum?id=ryF7rTqgl.
[24] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 2668–2677. URL: https://proceedings.mlr.press/v80/kim18d.html.
[25] Y. Belinkov, Probing Classifiers: Promises, Shortcomings, and Advances, Computational Linguistics 48 (2022) 207–219. doi:10.1162/coli_a_00422.
[26] X. Li, Z. Chen, J. M. Zhang, F. Sarro, Y. Zhang, X. Liu, Bias behind the Wheel: Fairness Testing of Autonomous Driving Systems, ACM Trans. Softw. Eng. Methodol. 34 (2025) 82:1–82:24. doi:10.1145/3702989.
[27] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset (2011).
[28] J. Zhang, J. Yang, P. Wang, H. Wang, Y. Lin, H. Zhang, Y. Sun, X. Du, Y. Li, Z. Liu, Y. Chen, H. Li, OpenOOD v1.5: Enhanced Benchmark for Out-of-Distribution Detection, 2024. doi:10.48550/arXiv.2306.09301. arXiv:2306.09301.
[29] C. Sakaridis, D. Dai, L. Van Gool, ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.
[30] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.
[31] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset, The International Journal of Robotics Research 32 (2013) 1231–1237. doi:10.1177/0278364913491297.
[32] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, P. Liang, Concept Bottleneck Models, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 5338–5348.
[33] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuScenes: A multimodal dataset for autonomous driving, 2020. doi:10.48550/arXiv.1903.11027. arXiv:1903.11027.
[34] P. Reis, P. Rigoll, E. Sax, Behavior Forests: Real-Time Discovery of Dynamic Behavior for Data Selection, in: 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), IEEE, Edmonton, AB, Canada, 2024, pp. 1962–1967. doi:10.1109/ITSC58415.2024.10920178.
[35] C. Steinhauser, J. Ransiek, J. Langner, E. Sax, Contextualizing the Sim-to-Real Gap for Hardware-in-the-Loop Testing of Highly Automated Driving Functions, in: 2024 8th International Conference on System Reliability and Safety (ICSRS), IEEE, Sicily, Italy, 2024, pp. 404–411. doi:10.1109/ICSRS63046.2024.10927458.