Global Interpretability for ProtoPNet Using Rule-Based Explanations

Alec Parise[1], Brian Mac Namee[1]

[1]University College Dublin, Dublin, Ireland

Abstract

Deep learning models are undeniably powerful but often criticised for their "black-box" nature. Prototypical Part Networks (ProtoPNets) address this by providing local, prototype-based explanations for individual predictions. However, while local insights are useful, they fail to capture the model's overall behaviour, a critical shortcoming when domain experts need to diagnose, validate and refine complex models. In this paper we propose a method that converts ProtoPNet activations into human-readable, prototype-based rules using a RIPPER-style induction algorithm. This rule-driven perspective not only elucidates the model's global decision-making process but also offers actionable insights for model debugging and enhancement.

Keywords

Explainable AI, Interpretable Machine Learning, ProtoPNet, RIPPER

1. Introduction

Deep learning models achieve high accuracy but often lack transparency, limiting trust in critical applications [20, 19]. Methods like the Prototypical Part Network (ProtoPNet) [1] have been developed to address this. ProtoPNet augments a CNN with a prototype layer that learns representative image patches; during inference the prediction made for an image is explained by showing a user the patches that most strongly influence the prediction—–a process inspired by prototype theory [26, 11, 12]. While such local explanations offer intuitive, example-based insights, they do not capture the model's global decision process. Global interpretability is essential for understanding model behavior, troubleshooting errors, and refining models, as seen in fields like medical imaging.

In our work, we convert ProtoPNet's local explanations into global, rule-based ones using an adapted RIPPER rule induction algorithm [3], inspired by frameworks such as AIMEE [2]. This approach maps prototype contributions into if-then rules, yielding a human-readable summary of the model's decision-making. We validate our method on the Caltech bird dataset [25] by comparing the rule-based algorithm's adherence, accuracy, and complexity to the original model, addressing challenges like the trade-off between adherence and interpretability and prototype-feature alignment [1, 11, 12, 14].

Late-breaking work, Demos and Doctoral Consortium, colocated with the 3rd World Conference on eXplainable Artificial Intelligence: July 09–11, 2025, Istanbul, Turkey

*Corresponding author.

alec.parise@ucdconnect.ie (A. Parise); brian.macnamee@ucd.ie (B. Mac Namee)

0009-0007-2303-3760 (A. Parise); 0000-0003-2518-0274 (B. Mac Namee)

2. Related Work

Prototype-based neural networks have emerged as a promising approach to enhance interpretability in deep learning. ProtoPNet [1] augments CNNs with a prototype layer that learns representative image patches, providing case-based explanations inspired by prototype theory [26]. Subsequent work has addressed challenges such as redundancy and scalability through methods like prototype merging [16] and decision-tree variants [10]. Alternative interpretability approaches include post-hoc attribution methods (e.g., LIME [24], SHAP [23]) and concept bottleneck models. Despite these advances, achieving a coherent global explanation of model behavior remains challenging. Our work builds on these foundations by transforming ProtoPNet's local explanations into global, rule-based insights via a RIPPER-style induction algorithm.

3. Generating Global Explanations

Our goal is to move from instance-level prototype explanations to a global, rule-based representation of model behavior. We combine a CNN backbone (e.g., ResNet [4]) with ProtoPNet [1] to obtain local interpretability, and then convert the resulting prototype activations into a set of human-readable if-then rules using a RIPPER-style induction algorithm [3]. This section details the process of extracting prototype activations, converting them into binary presence-absence features via a hard gating mechanism, and inducing global rules.

3.1. Extracting Prototype Activations

ProtoPNet augments a CNN with a prototype layer that learns representative image patches. The process involves:

1. **Feature Extraction:** An input image x is passed through the convolutional layers in the CNN backbone. Each convolutional filter extracts local features, and as the image propagates through the network, these features are spatially organized into a feature map—a multi-dimensional array where each element corresponds to a small, localized region (or "patch") of the original image.
2. **Prototype Comparison:** Each patch is compared to a set of learned prototypes using a similarity metric (typically the negative squared Euclidean distance).
3. **Activation Scoring:** A max-pooling operation aggregates the similarity scores for each prototype p_j:
$$a(p_j, x) = \max_{z \in f(x)} \text{sim}(z, p_j),$$
where $f(x)$ denotes the feature map and $\text{sim}(z, p_j)$ the similarity between patch z and prototype p_j. This score is then weighted by a non-negative coefficient via a ReLU activation:
$$s(p_j, x) = a(p_j, x) \times \text{ReLU}(w(p_j)).$$

The resulting prototype-activation profile highlights the image regions that most influence the classification.

3.2. Binary Feature Creation via Hard Gating Method

To induce interpretable rules, we convert the continuous activation scores $s(p_j, x)$ into binary presence-absence indicators using a hard gating mechanism. This approach discretizes the activations while retaining a differentiable approximation during training. The process involves three key steps:

1. **Noise Injection:** To simulate stochastic sampling and encourage exploration of binary states, we add Gumbel noise to the activation score:

$$y(p_j, x) = s(p_j, x) + G(p_j, x),$$

   where the Gumbel noise is generated as

$$G(p_j, x) = -\log\big(-\log(U(p_j, x))\big) \quad \text{with} \quad U(p_j, x) \sim \text{Uniform}(0, 1).$$

   This transformation converts uniformly distributed random values into noise that effectively approximates the extreme value distribution, which is useful for modeling the binary decision process.
2. **Soft Gating:** The noisy score $y(p_j, x)$ is then passed through a temperature-scaled sigmoid function to obtain a soft probability:

$$\pi(p_j, x) = \sigma\left(\frac{y(p_j, x)}{T}\right).$$

   The temperature parameter T controls the sharpness of the sigmoid output: a lower T results in a steeper function, closely approximating a hard threshold, while a higher T produces a smoother transition. This soft gating step maintains differentiability, which is critical for gradient-based optimization during training.
3. **Thresholding:** Finally, the soft probability $\pi(p_j, x)$ is converted into a binary decision using a threshold of 0.5:

$$\text{presence}(p_j, x) = \begin{cases} 1, & \pi(p_j, x) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

   This discretization yields a clear binary indicator of whether prototype p_j is considered active in the image x. The resulting binary presence-absence matrix is then used as input for the RIPPER-style rule induction algorithm.

Overall, this hard gating method effectively captures uncertainty in the activation signals while providing a differentiable pathway for training and a clear binary representation for downstream rule extraction.

3.3. Adapted RIPPER Rule Induction

After converting each image's continuous prototype activations into binary indicators, we train a RIPPER-style rule learner [3] to form a global, logical view of prototype-based classifications. In

standard RIPPER, a one-vs-all approach iteratively adds or removes conditions (e.g., "Prototype i is present") to maximize target class coverage while minimizing misclassifications, followed by pruning to avoid overfitting.

Our implementation extends RIPPER to better accommodate prototype-based features by:

1. **Cross-Class Penalty:** Adding a term in the First Order Inductive Learner (FOIL) [6] gain calculation to penalize conditions referencing non-target prototypes.
2. **Merging Redundant Single-Literal Rules:** Combining multiple single-literal rules for the same class into a single rule with an *AND* clause to reduce redundancy while preserving coverage and accuracy.
3. **Top- N Prototype-Based Fallbacks:** Constructing minimal fallback rules using the top-N frequently activated prototypes when a class remains uncovered.
4. **Adaptive Rule Growth and Pruning:** Iteratively growing rules by adding conditions that maximize modified FOIL gain, then pruning using a hold-out set and removing a fraction of positive examples to prevent redundancy.

This adapted RIPPER method yields a global set of rules consistent with ProtoPNet's decisions, providing a structured framework to analyze the model's internal logic.

4. Experiment Design

This section describes the design of an experiment performed to evaluate the approach to generate global explanations for ProtoPNet models described in Section 3. The evaluation uses 10 randomly selected classes from the Caltech Bird Dataset [25].300 images from these classes were randomly for training the ProtoPNet an this set was expanded to 1794 augmented images (using random flips, skewing, and other perturbations) to enhance robustness.

For binarized feature extraction, 280 test images spanning the 10 classes were sampled from the original dataset. Seventy percent of these images were used for training and cross-validating the RIPPER rule induction model, with the remaining 30% reserved for testing the ability of the rule-based representation to replicate ProtoPNet's behavior.

To further ensure the reliability of our rule induction process, we performed dynamic hyperparameter tuning using the **Optuna** framework [27] in conjunction with 5-fold cross validation. Rather than setting rule induction hyperparameters (e.g., `max_conditions`, `min_coverage`, `prune_size`, etc.) arbitrarily, Optuna systematically explores the search space of possible values—leveraging TPE (Tree-structured Parzen Estimator) [28] to converge toward optimal configurations.

We assessed the quality of the generated explanations with three metrics:

- **Adherence:** The percentage of cases where the RIPPER model's predictions agree with ProtoPNet on a hold-out test set.
- **Accuracy:** The RIPPERmodel's prediction accuracy compared to ground truth labels and ProtoPNet's performance.
- **Complexity:** Measured by the total number of extracted rules and the average number of conditions per rule.

These metrics were computed both overall and per class. ProtoPNet achieved a baseline accuracy of 85.13% on the test set. Our evaluation focuses on whether the extracted rules faithfully capture ProtoPNet's internal reasoning while remaining interpretable for end users.

5. Results

Figure 1 shows examples of three separate rules extracted from the trained ProtoPNet using the our approach. These rules offer a global explanation of how the model identifies classes based on the presence or absence of specific prototypes. The accompanying miniature images depict the actual prototype patches learned by ProtoPNet, illustrating the visual features associated with each prototype.

By requiring certain prototypes to be *present* or *absent*, each rule provides insight into the decision-making process. For example, one rule states:

> IF Contains Prototype 78 AND Contains Prototype 77 AND NOT Contains Prototype 22 THEN class=7

This particular rule highlights the significance of certain plumage patterns (Prototypes 78 and 77), while simultaneously indicating the absence of another plumage patterns (Prototype 22) for predicting the Black Billed Cuckoo. Collectively, the three rules in the figure illustrate how a RIPPER-style surrogate can provide multiple, human-readable explanations for what a ProtoPNet model has learned.

To assess the quality of these global explanations, we evaluated their adherence and accuracy (see Section 4). Our approach achieved an adherence of 70.37% and a rule-based accuracy of 71.60%. Additionally, the method produced a total of 71 rules with an average of 2.42 conditions per rule. These metrics demonstrate that this is a promising approach to effectively captures key decision cues while maintaining interpretability. Overall, by integrating both the presence and absence of prototypes, the approach extracts a comprehensive yet concise set of rules that elucidate the model's learned knowledge, thereby offering the potential to streamline analysis and enhance human interpretability.

6. Conclusion

This paper describes an approach to bridge the gap between local, prototype-based explanations and a global, rule-based perspective. By moving beyond single-image explanations, it reveals common patterns that a model has learned, offering a high-level, comprehensible map of how prototypes shape model decisions.

Future work will compare this strategy to established approaches such as decision trees and concept bottleneck models, in order to further contextualize its performance and scalability. Additionally, we plan to extend our evaluation framework by incorporating alignment with domain expert knowledge, analyzing coverage and specificity, validating generalizability on new or perturbed data, and conducting user studies to assess interpretability. We will also investifgate modifications to the approach including different gating mechanisms, different rule extraction algorithms, and the integration of human feedback into the rule building process.

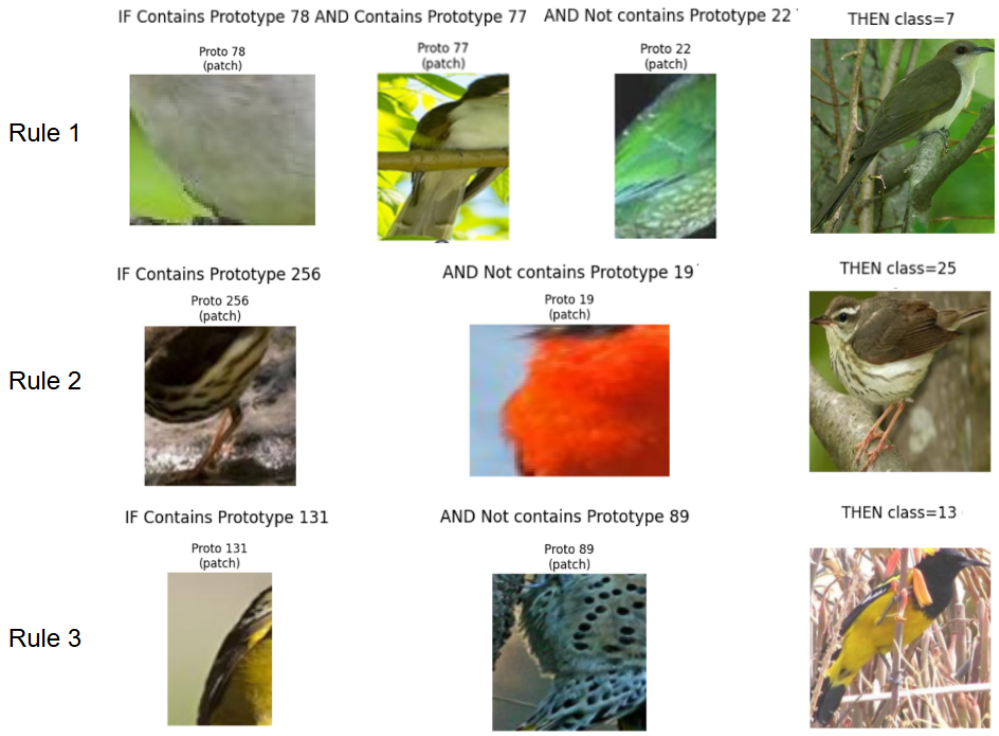

Figure 1: Example of the rules extracted from a trained ProtoPNet model to provide a global explanation of what it has learned. Each rule defines how the presence or absence of certain prototypes indicates class membership.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

Declaration on Generative AI

The authors have not employed any generative-AI tools in the preparation of this article.

References

[1] Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: *This looks like that: Deep learning for interpretable image recognition.* Advances in Neural Information Processing Systems, 32 (2019).

[2] Piorkowski, M., et al.: *AIMEE: An Interactive Model Explanation Environment for Deep Neural Networks.* In: Proceedings of the [Conference] (2023).

[3] Cohen, W.W.: *Fast Effective Rule Induction.* Machine Learning, 2, 115–137 (1995).
[4] He, K., Zhang, X., Ren, S., Sun, J.: *Deep Residual Learning for Image Recognition.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778 (2016).
[5] Chiaburu, T., Haußer, F., Bießmann, F.: *CoProNN: Concept-based Prototypical Nearest Neighbors for Explaining Vision Models.* arXiv preprint arXiv:2404.14830 (2024).
[6] Richards, B. L., Mooney, R. J.: *First-order Theory Revision.* In: *Machine Learning Proceedings 1991*, pp. 447–451. Elsevier (1991).
[7] Bontempelli, A., Teso, S., Tentori, K., Giunchiglia, F., Passerini, A.: *Concept-level debugging of part-prototype networks.* arXiv preprint arXiv:2205.15769 (2022).
[8] Heinrich, R., Sick, B., Scholz, C.: *AudioProtoPNet: An interpretable deep learning model for bird sound classification.* arXiv preprint arXiv:2404.10420 (2024).
[9] Sinhamahapatra, P., Shit, S., Sekuboyina, A., Husseini, M., Schinz, D., Lenhart, N., Menze, J., Kirschke, J., Roscher, K., Guennemann, S.: *Enhancing Interpretability of Vertebrae Fracture Grading using Human-interpretable Prototypes.* arXiv preprint arXiv:2404.02830 (2024).
[10] Nauta, M., Van Bree, R., Seifert, C.: *Neural Prototype Trees for Interpretable Fine-Grained Image Recognition.* In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14933–14943 (2021).
[11] Sourati, Z., Deshpande, D., Ilievski, F., Gashteovski, K., Saralajew, S.: *Robust Text Classification: Analyzing Prototype-Based Networks.* arXiv preprint arXiv:2311.06647 (2023).
[12] Narayanan, A., Bergen, K.J.: *Prototype-Based Methods in Explainable AI and Emerging Opportunities in the Geosciences.* arXiv preprint arXiv:2410.19856 (2024).
[13] Kim, B., Rudin, C., Shah, J.A.: *The Bayesian Case Model: A Generative Approach for Case-Based Reasoning and Prototype Classification.* Advances in Neural Information Processing Systems, 27 (2014).
[14] Rymarczyk, D., Struski, Ł., Górszczak, M., Lewandowska, K., Tabor, J., Zieliński, B.: *Interpretable image classification with differentiable prototypes assignment.* In: European Conference on Computer Vision, 351–368 (2022).
[15] Li, A.J., Netzorg, R., Cheng, Z., Zhang, Z., Yu, B.: *Improving Prototypical Visual Explanations with Reward Reweighing, Reselection, and Retraining.* In: Proceedings of the Forty-first International Conference on Machine Learning (2023).
[16] Rymarczyk, D., Struski, Ł., Tabor, J., Zieliński, B.: *Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification.* In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 1420–1430 (2021).
[17] Netzorg, R., Li, J., Yu, B.: *Improving prototypical part networks with reward reweighing, reselection, and retraining.* arXiv preprint arXiv:2307.03887 (2023).
[18] Vilone, G., Longo, L.: *A quantitative evaluation of global, rule-based explanations of post-hoc, model agnostic methods.* Frontiers in Artificial Intelligence, 4, 717899 (2021).
[19] Rudin, C.: *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.* Nature Machine Intelligence, 1(5), 206–215 (2019).
[20] Lipton, Z.C.: *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.* Queue, 16(3), 31–57 (2018).
[21] Wang, Y.: *A comparative analysis of model agnostic techniques for explainable artificial intelligence.* Research Reports on Computer Science, 25–33 (2024).
[22] Yuksekgonul, M., Wang, M., Zou, J.: *Post-hoc concept bottleneck models.* arXiv preprint

arXiv:2205.15480 (2022).
[23] Lundberg, S.M., Lee, S.-I.: *A unified approach to interpreting model predictions*. arXiv preprint arXiv:1705.07874 (2017).
[24] Ribeiro, M.T., Singh, S., Guestrin, C.: *"Why should I trust you?" Explaining the predictions of any classifier*. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144 (2016).
[25] Krizhevsky, A.: *Learning Multiple Layers of Features from Tiny Images*. Technical Report, University of Toronto (2009). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[26] Rosch, E. H.: *Natural Categories*. Cognitive Psychology, 4(3), 328–350 (1973). Elsevier.
[27] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2019)*, pp. 2623–2631. ACM (2019)
[28] Watanabe, S.: *Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance*. arXiv preprint arXiv:2304.11127 (2023).
[29] Jang, E., Gu, S., Poole, B.: *Categorical Reparameterization with Gumbel-Softmax*. arXiv preprint arXiv:1611.01144 (2016).
[30] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5, 115–133. Springer.
[31] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*.
[32] Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
[33] Louizos, C., Welling, M., and Kingma, D. P. (2017). Learning sparse neural networks through L_0 regularization. *arXiv preprint arXiv:1712.01312*.
[34] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.