Interpreting Uncertainty: Understanding Neural Network Decisions with Conceptual Hierarchies

Angie Boggust
MIT CSAIL
Cambridge, USA
aboggust@csail.mit.edu

Arvind Satyanarayan
MIT CSAIL
Cambridge, USA

Hendrik Strobelt
University of Konstanz
Konstanz, Germany

***Abstract*—Understanding neural network uncertainty is essential to comprehend model behavior, ensure safe deployment, and intervene appropriately. However, existing uncertainty values only describe the model's decision, ignoring its consideration of other choices and masking the reason for the uncertainty. By leveraging human knowledge about related decisions, we expand single uncertainty values into a hierarchy of concepts, creating an *uncertainty fingerprint*. Unlike an uncertainty value, an uncertainty fingerprint describes the model's confidence in every possible decision, distinguishing how the model proceeded from a broad idea to its precise prediction. Using hierarchical entropy, we represent and compare fingerprints based on the model's decision-making process to uncover patterns of uncertainty. Doing so facilitates important model analysis tasks in image classification settings, including categorizing types of model uncertainty, identifying common failure modes, and reasoning about a model's decision.**

***Index Terms*—Machine learning, Uncertainty, Complex hierarchies, Knowledge modeling**

I. Introduction

As deep neural networks are deployed in high-stakes applications, like cancer diagnoses [1], it is increasingly important to understand when and why networks are uncertain. Currently, uncertainty estimation algorithms, such as Bayesian Neural Networks [2], ensembling [3], dropout [4], and model calibration [5], approximate how certain a model is in its decisions and present this information as a single confidence value. These confidence values provide important context about the model's decision-making process, for instance, uncovering when a correct prediction was actually a random guess. Confidence values are widely used to analyze model behavior and increase human trust and acceptance of model-based recommendation systems [6].

While uncertainty estimation expands human insight into model decision-making, using a single quantitative value masks the reasoning behind the uncertainty. Quantitative values provide the model's probabilistic confidence in its decision but do not describe the reason for its confidence. For instance, in an image classification setting, a model might express that it is 20% confident in its decision. However, humans have a rich vocabulary to describe their confusion and might say they are uncertain because there are multiple possible objects or the image is corrupted. Understanding the reason for the uncertainty is critical to make an appropriate intervention, like modifying the modeling task to handle multi-object images or addressing data quality issues.

The second limitation of confidence values is that they ignore relationships between the model's confidence in its decision and other possible decisions. While uncertainty estimation procedures can estimate the model's confidence for every possible decision (e.g., every output class), the relationships between those decisions are ignored. However, understanding other options the model was considering and their relationship to its ultimate decision are crucial to understanding the model's reasoning process. For example, suppose our model is only 20% confident that the image is a `trout`. Then, it is essential to know whether the other 80% is distributed across other fish species versus unrelated classes. Depending on the task, we may be comfortable if our model can not differentiate fine-grained species of fish but concerned if the model can not distinguish a `trout` from a `tractor`.

To operationalize uncertainty values and overcome their limitations, we utilize the conceptual relationships between model output classes. Dataset classes inherit the semantic relationships in language. As a result, datasets, like CIFAR-100 [7], contain built-in conceptual hierarchies with parent/child relationships between the output classes and more abstract concepts. By propagating uncertainty values throughout the conceptual hierarchy, we expand the number and complexity of concepts we can reason about. The result is an *uncertainty fingerprint* representing the model's confidence for every concept in the hierarchy for a given input (Fig. 1). Unlike an uncertainty value, an uncertainty fingerprint describes the model's confidence in every possible decision. Instead of only having uncertainty values for output classes, like `trout`, we now know the model's confidence in a range of higher-level concepts, like `fish`. Uncertainty fingerprints give us a more detailed vocabulary to describe and categorize uncertainty by distinguishing how the model proceeded from a broad idea to its precise prediction.

We introduce hierarchical entropy metrics to analyze uncertainty fingerprints at scale and generate a global understanding of network uncertainty. Hierarchical entropy encodes the distribution of uncertainty at each level of the hierarchy and represents how certain the model was at each level of abstraction. Using entropy encodes how the model became certain in its decision while ignoring which decision the model

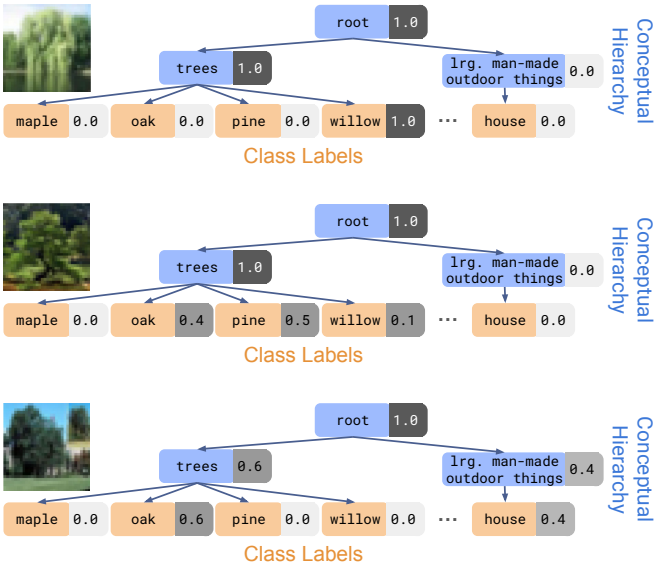

Fig. 1. Interpreting uncertainty reveals the model's decision-making process. By propagating the model's confidence through a conceptual hierarchy, we create an *uncertainty fingerprint* describing the model's behavior. Here we show uncertainty fingerprints for three CIFAR-100 [7] images. The model is fully confident the top image contains a `willow`. However, in the middle example, the model is sure the image comprises a `tree` but uncertain if it is an `oak`, `pine`, or `willow`. And in the bottom example, the model is split between multiple objects (`oak` or `house`) in the image.

made. For instance, the hierarchical entropy of an input that the model is confident is a `trout` will be identical to an input that the model is confident is a `tractor`. As a result, clustering inputs based on hierarchical entropy reveals patterns of model decision-making regardless of semantic content.

Interpreting uncertainty supports critical model analysis tasks, including categorizing types of model uncertainty, identifying common failure modes, and reasoning about a model's decision. By combining uncertainty fingerprints and hierarchical entropy, we encode model inputs into an uncertainty-based embedding space. The embedding space represents the model's decision-making patterns and can reveal common types of model uncertainty and identify outliers. Expanding the vocabulary to describe model uncertainty can lead to critical insights into model behavior that inform better modeling procedures, human intervention, and model deployment.

II. Related Work

To gain trust and adoption, neural networks must communicate their uncertainty. Human stakeholders need to understand *how confident the model is in its decision*, *what other options it considered*, and *how it arrived at this decision* to ensure proper development and safe deployment.

As a result, research has developed techniques to measure the model's confidence in its decision. Bayesian machine learning [2] inherently includes uncertainty by relying on probabilistic models that output confidence bounds for their decisions. However, deep learning models are deterministic and must depend on uncertainty estimation techniques to measure uncertainty. The most straightforward approach is to scale the model's output via a softmax function to extract confidence probabilities for each class. However, many deep learning models are highly overconfident, so their confidences may not accurately represent the probability of a correct output [8]. Calibration techniques such as temperature scaling [9] and histogram binning [5] ensure the output probabilities match the empirical frequencies found in the data. Other uncertainty estimation techniques mimic probabilistic models by adding dropout noise during inference [4] or ensembling many models independently trained on the same data [3].

Once uncertainty is extracted, it can be categorized as either epistemic or aleatoric [10, 11]. Epistemic uncertainty indicates that the network is uncertain because of a lack of information and encompasses cases where multiple model parameters can separate the input data. Increasing the amount of data can reduce the amount of epistemic uncertainty. Aleatoric uncertainty is due to observational noise. For instance, noise may come from labeling errors or human disagreements about labels. Aleatoric uncertainty is not reducible because even optimal model parameters will not accurately separate the data.

Categorizing uncertainty as epistemic or aleatoric can inform important decisions about data collection. However, these categorizations are orthogonal to our goal of using uncertainty to understand model behavior. For example, if we showed our model an out-of-distribution input, its uncertainty would be epistemic. However, we would still have questions about what the model thought the input might be or how it reasoned about this new input. By combining model uncertainty with conceptual hierarchies, we can understand the model's confidence in various concepts and begin to answer these questions. Using our method, we can identify options the model considered, find a high-level concept the model was confident about, and uncover inputs where the model was similarly uncertain.

We rely on conceptual hierarchies to expand the vocabulary models use to describe their uncertainty. Conceptual hierarchies are human representations of knowledge, such as relationships between words [12] or evolutionary taxonomies of organisms [13]. In machine learning, conceptual hierarchies are built into many tasks, such as image classification [7], medical diagnostics [14], and text prediction [12]. Even datasets that do not explicitly include a hierarchy have been incorporated into existing conceptual hierarchies by matching their output classes with corresponding concept nodes [15].

Interpretability research has also focused on better understanding machine learning model decisions [16, 17]. Feature visualization [18, 19], attribution [20, 21, 22], dimensionality reduction [23, 24], and combinations of multiple methods [25, 26] help users reason about neural networks. These methods allow users to explore how models represent semantic relationships like features important to a specific class or images similar to one another in latent space. However, they rarely incorporate confidence values and primarily surface the model's semantic representations. In contrast, interpreting uncertainty uncovers patterns in model decision-making based on the similarity of its confusion.

III. Interpreting Uncertainty

Understanding model uncertainty is challenging. While models output their confidence in each output class, they do not consider the relationships between the classes. To make sense of model reasoning, humans must manually examine a complex collection of its confidence values. If the model is confident in one output class, the confidence values will be easy to understand. However, if the confidence is distributed across many classes, humans must identify relationships between the classes to reverse engineer the model's reasoning. To improve the uncertainty analysis process, we synthesized four design goals that interpreting uncertainty must achieve.

1. **Describe the model's confidence at varying levels of abstraction.** Existing uncertainty estimation techniques expose the model's uncertainty at the output granularity. Interpreting uncertainty should communicate uncertainty at various levels of abstraction to express patterns like *the model is confident this image contains a fish but does not know the species*.
2. **Represent the model's decision-making pattern.** Interpreting the model's decision-making pattern from a list of tens, hundreds, or thousands of uncertainty values is cognitively challenging. Interpreting uncertainty should make it easy to understand how the model came to its prediction and what other options it considered.
3. **Enable large-scale analysis of model uncertainty.** Analyzing the model's confidence values describes its reasoning on an input. However, exploring model uncertainty across an entire dataset can uncover behavior patterns and failure modes. Interpreting uncertainty should support the analysis of single inputs and datasets.
4. **Support post hoc interpretation on a wide variety of models and datasets.** Understanding model uncertainty is critical for thorough development and safe deployment. Interpreting uncertainty should apply to many model frameworks, tasks, and modalities.

To achieve our design goals, we incorporate the output classes into a conceptual hierarchy that expands the number and complexity of concepts in which the model can express its uncertainty (Design Goal 1). By propagating the model's output uncertainties through the hierarchy, we create an *uncertainty fingerprint* that represents how the model proceeded from a broad concept to a precise prediction (Design Goal 2). We encode each uncertainty fingerprint as a hierarchical entropy vector representing the model's uncertainty at each level of abstraction (Design Goals 1 and 2). Comparing hierarchical entropy vectors identifies global uncertainty patterns and reveals recurring model and dataset failure modes (Design Goal 3). Our method only requires uncertainty estimates and a conceptual hierarchy related to the task, making it usable across various data modalities, model architectures, and downstream applications (Design Goal 4).

A. Integrating Conceptual Hierarchies with Model Outputs

Conceptual hierarchies represent human knowledge by encoding relevant concepts and the parent/child relationships between them. For example, conceptual hierarchies often describe the relationships between objects, and many machine learning datasets, such as CIFAR-100 [7], are built on top of them. By linking the model's output labels into these conceptual hierarchies, we can describe the model's confidence in additional and more complex concepts.

We represent conceptual hierarchies as directed acyclic graphs (DAG). In each DAG, the leaves are the output classes, the root is an abstract concept, and every path from the root to a leaf is the same length. The DAG consists of m nodes: $N = \{n_0, ..., n_{m-1}\}$. The nodes are split into h levels: $L = \{l_0, ..., l_{h-1}\}$, where the leaf nodes are contained in l_0. The CIFAR-100 [7] DAG corresponds to its built-in hierarchy ($h = 3$, $m = 121$, $|l_0| = 100$). The leaves are the classes, the second level contains the superclasses, and the root is an abstract node connecting all superclasses.

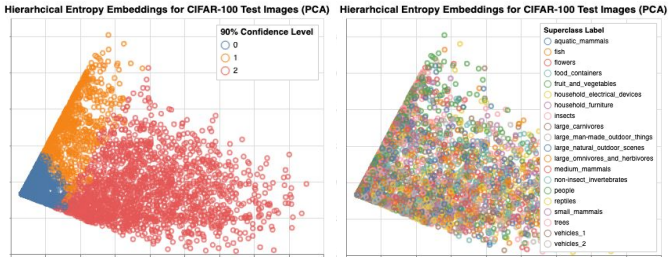

Fig. 2. Hierarchical entropy encodes how the model proceeded from a broad concept to a precise prediction. Here, we show two PCA projections [23] of the hierarchical entropy of CIFAR-100 [7] test images. The points are colored by the level where the model is 90% confident in a single node (left) and their superclass label (right). Since hierarchical entropy encodes the model's decision-making process, we see a correlation between hierarchical entropy and the confidence level. There is no correlation between hierarchical entropy and the superclass label because, unlike other interpretability methods, hierarchical entropy explicitly ignores input semantics.

B. Describing Confidence via Uncertainty Fingerprints

We use the conceptual hierarchy to represent the model's uncertainty. For a given input, we compute the model's confidence for every output class: $C = \{c_0, ..., c_{|l_0|-1}\}$ where $\sum C = 1$. Each output confidence c_i corresponds to its leaf node n_i. In our examples, we use the softmax probabilities of the model's output.

Next, we assign uncertainties to every node in the hierarchy, creating an *uncertainty fingerprint* (Fig. 1). An uncertainty fingerprint represents the model's decision-making process on an input. It is a collection of confidence values for every node in the hierarchy: $F = \{f_0, ..., f_{m-1}\}$. The fingerprint confidence of a node is the sum of the confidence of every reachable leaf node.

$$f_i = \sum \{c_j \, \forall \, c_j \in C \mid \text{a path exists from } n_i \text{ to } n_j\}$$

Since leaf nodes can only reach themselves, they inherit the model's confidence in their class ($f_i = c_i$ for $n_i \in l_0$).

C. Encoding Fingerprints with Hierarchical Entropy

Uncertainty fingerprints represent the model's behavior on a single input, but comparing multiple fingerprints allows us

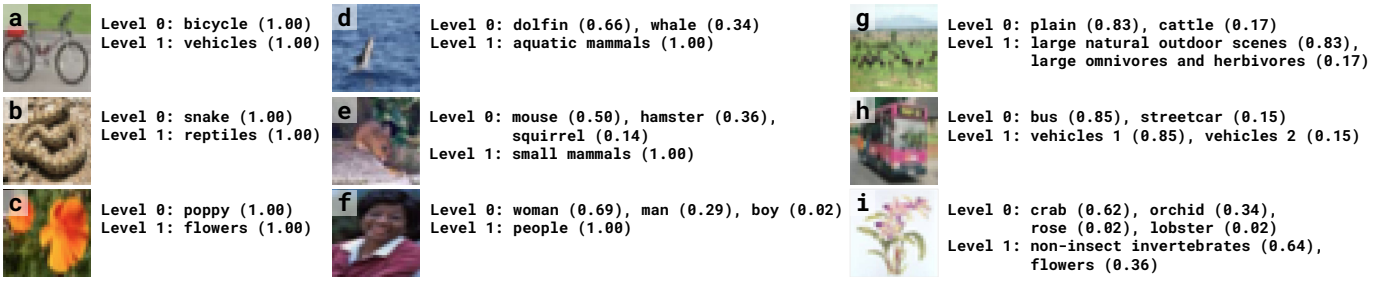

Fig. 3. Examples of the model's decision-making process on nine images from the CIFAR-100 test set. Each image is accompanied by its fingerprint nodes that have non-zero confidence. The left column contains examples where the model was confident in its level 0 prediction. The middle column comprises examples where the model was uncertain until the level 1 superclasses. These examples reveal cases where the model has yet to learn the right level of abstraction (e.g., type of rodent), or the image is hard to precisely categorize (e.g., a small `aquatic mammal` in a large body of water). The rightmost column includes instances where the model was not confident until the root node. These examples reveal dataset complexities (e.g., multi-object images), poor hierarchical categorizations (e.g., two `vehicle` superclasses), and model confusion (e.g., not differentiating a `flower` from a `crab`).

to analyze the model's decision-making processes across the entire dataset. One way to compare fingerprints is to compare the uncertainty at each node. Two fingerprints would be similar if they had similar uncertainty values on identical nodes, so the inputs would likely be from related classes. However, we are interested in how a model makes its decisions, not the semantic content of input, so we use hierarchical entropy. Hierarchical entropy encodes how the model proceeds from a broad idea (root) to a precise concept (leaf) and ignores the model's semantic categorization of an input (Fig. 2).

Hierarchical entropy measures the entropy of the model's confidence for each level in the conceptual hierarchy. It takes in an uncertainty fingerprint and outputs a vector of length h representing the uncertainty at each level.

$$\mathrm{HE}_k(F) = -\sum\{f_i * log(f_i)\ \forall\, f_i \in F \,|\, n_i \in l_k\}$$

Hierarchical entropy treats each level of the fingerprint as a probability distribution of the model's confidence in the level's nodes. Given an uncertainty fingerprint can be thought of as levels of probability distributions, hierarchical entropy measures how certain the model is at each level of abstraction. If the model is fully confident in a single node, then that level's entropy will be 0. On the other hand, if the model is very uncertain and its confidence is distributed across many nodes in a level, that level's entropy will be higher.

IV. Model Uncertainty Analysis

Analyzing model uncertainty can improve our understanding of model behavior. For example, understanding uncertainty is essential when scrutinizing the performance of a new model, trading-off collaboration in an integrated workflow, and identifying failure modes of a high-stakes model. Interpreting uncertainty supports these use cases by revealing the model's reasoning for a specific decision, uncovering typical failure modes, and discovering dataset limitations.

In this case study, we are interpreting uncertainty to better understand the performance of an image classification model. We trained a ResNet20 [27] on CIFAR-100 [7] using cross entropy loss optimized via stochastic gradient descent with Nesterov momentum [28] and data augmentation [29]. The resulting model achieves 68% accuracy on the test set. This accuracy is low, so let's interpret the model's uncertainty to discover how it is making decisions and what types of uncertainty it faces. Using the CIFAR-100 conceptual hierarchy, we create uncertainty fingerprints for every image in the test set and compute their hierarchical entropy vectors.

Given the model's poor performance, it is essential to understand how confident the model is in its predictions. If the model is highly confident, it could indicate the model is overfitting to the training data and needs more regularization. On the other hand, if the model is uncertain in its predictions, it likely has yet to learn the distinguishing features of classes or the right level of abstraction. Fig. 2 shows the hierarchical entropy vectors colored by the conceptual level where the model becomes 90% confident in a single node. We see a distribution of points where the model is confident in its output (blue), uncertain until the superclass (orange), and unsure until the abstract root concept (red). The distribution balance indicates the model is not very confident in its predictions. We see many cases where the model has learned the superclass but not the fine-grained output class (yellow). This type of failure might be acceptable if our downstream task does not require high specificity. However, it is concerning we see many instances where the model is not confident until the root node (red). These are instances where the model is confused across many different output classes and superclasses. Let's examine what is causing the model to be so highly confused.

Sampling images and their uncertainty fingerprints reveals how the model reasoned about each input. Our model is confident about easy-to-distinguish images like a well-framed `bicycle` (Fig. 3a) or a closeup of a `poppy` (Fig. 3c). However, in some cases, our model struggles to categorize an image at the correct level of abstraction. For instance, our model is not sure what type of rodent is in Fig. 3e or the age and gender of the `person` in Fig. 3f. Our analysis also exposes dataset issues. Some images are challenging to classify precisely, like a small `aquatic mammal` jumping out of a large body of water (Fig. 3d). Other images contain multiple valid objects, like the `plain` of grazing `cattle` (Fig. 3g). The fingerprints also reveal that the conceptual hierarchy is imprecise. Two

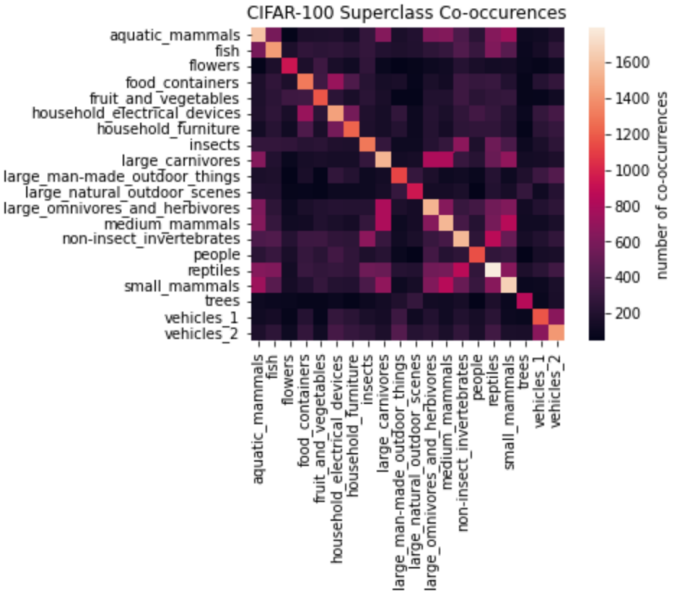

Fig. 4. The model's co-occurrence matrix of the CIFAR-100 [7] superclasses. A co-occurrence occurs if each superclass has an output subclass with non-zero confidence in the same uncertainty fingerprint. The co-occurrence matrix reveals the model's most frequent source of uncertainty is a lack of classification precision. Superclasses are most frequently confused with themselves or related superclasses, like `vehicles 1` and `vehicles 2`.

superclasses include highly similar concepts — `vehicles 1` and `vehicles 2` (Fig. 3h). Finally, we see examples where the model is confused by an image that is easily classifiable by a human. For example, Fig. 3i clearly contains a `flower`, but the model is unsure whether it is an `orchid` or a `crab`.

The fingerprints reveal many reasons for our model's uncertainty and low performance, but how often do these failures occur, and which ones are most prevalent? To answer this, we create a co-occurrence matrix of the level 1 superclasses (Fig. 4). When two level 0 output nodes have non-zero confidence in a fingerprint, their level 1 superclasses co-occur. For example, in Fig. 3g, `large natural outdoor scenes` and `large omnivores and herbivores` co-occur; in Fig. 3d, `aquatic mammals` co-occurs with itself. There is no co-occurrence in the fingerprint for Fig. 3a because the model is only confident in one output class. Analyzing the co-occurrence matrix reveals that the most significant cause of our model's uncertainty is an inability to classify an image at the fine-grained output level. Most co-occurrences occur along the diagonal, meaning most uncertainty exists within the same superclass. For example, `reptiles` co-occurs with itself over 1700 times, meaning our model struggles to distinguish between members of the `reptiles` superclass: `crocodile`, `dinosaur`, `lizard`, `snake`, and `turtle`. Our imprecise conceptual hierarchy is also a major cause of model uncertainty. As we saw in our previous analysis, the model often confuses classes in the `vehicles 1` and `vehicles 2` superclasses. Other than that, our model has few egregious confusions. Most other co-occurrences are between superclasses containing animals: `aquatic mammals`, `large carnivores`, `large omnivores and herbivores`, `medium mammals`, `reptiles`, and `small mammals`. While these confusions indicate our model lacks precision, we do not see many blatant confusions of unrelated superclasses.

By interpreting the uncertainty of our image classification model, we discovered reasons for its poor performance. Hierarchical entropy encoded our images into an uncertainty embedding space that exposed a distribution of decision types. Looking at uncertainty fingerprints from different regions of the embedding space revealed that our model confidently classified clear images but was less sure about obscure images. We also discovered dataset limitations, including overlapping superclasses and multi-object images. By looking at fingerprint node co-occurrences, we found our model's largest source of uncertainty was an inability to classify images precisely at the output level. These insights helped us understand our model's behavior and could inform model improvements.

V. Discussion

Interpreting uncertainty operationalizes model uncertainty values to provide insight into model behavior. By integrating conceptual hierarchies with output classes, we include new concepts at different levels of abstraction, expanding the vocabulary we can use to describe a model's uncertainty. Using the hierarchy, we create uncertainty fingerprints for each input that expresses how the model proceeded from an abstract concept to its precise prediction. Comparing the decision-making pattern of multiple fingerprints with hierarchical entropy enables global analysis of a model's behavior. In a case study, interpreting the uncertainty of an image classification model categorizes types of model confusion and uncovers common model pathologies that impact performance.

A fundamental interpretation choice of our method is to align model explanations with existing human knowledge. The human stakeholders who develop and interpret machine learning have a learned hierarchical worldview, so we design models that express uncertainty within the same hierarchical structure. While other interpretability methods output convoluted explanations (e.g., feature visualizations [18]), aligning model confidence with human priors facilitates efficient exploration of model behavior.

To express model uncertainty at multiple levels of human abstraction, we require conceptual hierarchies integrated with machine learning datasets. Many research datasets contain hierarchical structures [7, 12, 14]; however, real-world datasets or data from poorly understood domains may not. Future work incorporating new data into existing hierarchies or inducing conceptual hierarchies from flat datasets could address this limitation and expand the applications of interpreting uncertainty.

Acknowledgment

This research is partially supported by IEEE Computational Intelligence Society Graduate Student Research Grant 2022.

References

[1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[2] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. W. 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.

[3] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6402–6413.

[4] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*, M. Balcan and K. Q. Weinberger, Eds., vol. 48. JMLR, 2016, pp. 1050–1059.

[5] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *International Conference on Machine Learning (ICML)*, 2001, pp. 609–616.

[6] A. Bussone, S. Stumpf, and D. O'Sullivan, "The role of explanations on trust and reliance in clinical decision support systems," in *International Conference on Healthcare Informatics (ICHI)*. IEEE Computer Society, 2015, pp. 160–169.

[7] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[8] N. Thain, A. Pearce, J. Snoek, and M. Pushkarna, "Are model predictions probabilities?" *Google PAIR Explorables*, 2022, https://pair.withgoogle.com/explorables/uncertainty-calibration/.

[9] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, vol. 70. PMLR, 2017, pp. 1321–1330.

[10] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[11] C. R. Fox and G. Ülkümen, "Distinguishing two dimensions of uncertainty," in *Perspectives on Thinking, Judging and Decision Making*, G. K. Wibecke Brun, Gideon Keren and Henry, Eds. Universitetsforlaget, 2011, ch. 1, pp. 1–14.

[12] G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[13] R. Kulkarni, "Tree of life," 2021. [Online]. Available: https://www.kaggle.com/ds/1295752

[14] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[15] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.

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

[17] A. Rai, "Explainable AI: From black box to glass box," *Journal of the Academy of Marketing Science*, vol. 48, no. 1, pp. 137–141, 2020.

[18] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[19] D. Erhan, Y. Bengio, A. C. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," 2009.

[20] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 3319–3328.

[21] B. Carter, J. Mueller, S. Jain, and D. K. Gifford, "What made you do this? Understanding black-box decisions with sufficient input subsets," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2019, pp. 567–576.

[22] A. Boggust, B. Hoover, A. Satyanarayan, and H. Strobelt, "Shared Interest: Measuring human-AI alignment to identify recurring patterns in model behavior," in *Conference on Human Factors in Computing Systems (CHI)*. ACM, 2022, pp. 10:1–10:17.

[23] I. T. Jolliffe, "Principal components in regression analysis," in *Principal Component Analysis*. Springer, 1986, pp. 129–155.

[24] A. Boggust, B. Carter, and A. Satyanarayan, "Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples," in *International Conference on Intelligent User Interfaces (IUI)*, 2022, pp. 746–766.

[25] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, 2018, https://distill.pub/2018/building-blocks.

[26] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah, "Activation atlas," *Distill*, 2019, https://distill.pub/2019/activation-atlas.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[28] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2013, pp. 1139–1147.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 630–645.