Discerning Decision-Making Process of Deep Neural Networks with Hierarchical Voting Transformation

Anonymous Author(s)

Abstract

Neural network based deep learning techniques have shown great success for numerous applications. While it is expected to understand their intrinsic decision-making processes, these deep neural networks often work in a black-box way. To this end, in this paper, we aim to discern the decision-making processes of neural networks through a hierarchical voting strategy by developing an explainable deep learning model, namely Voting Transformation-based Explainable Neural Network (VOTEN). Specifically, instead of relying on massive feature combinations, VOTEN creatively models expressive single-valued voting functions between explicitly modeled latent concepts to achieve high fitting ability. Along this line, we first theoretically analyze the major components of VOTEN and prove the relationship and advantages of VOTEN compared with Multi-Layer Perceptron (MLP), the basic structure of deep neural networks. Moreover, we design efficient algorithms to improve the model usability by explicitly showing the decision processes of VOTEN. Finally, extensive experiments on multiple real-world datasets clearly validate the performances and explainability of VOTEN.

1 Introduction

Neural network based deep learning techniques have attracted great attention from both academia and industry in the past decade. Compared with classic machine learning models, deep neural networks have much higher expressiveness and adaptability for complicated data input, and thus have made tremendous success in various application domains, such as Computational Vision [26, 46], Natural Language Processing [44, 18], and Recommender Systems [29, 21]. Nevertheless, since deep neural networks usually have complicated connections of hidden units, a long-standing challenge is how to decipher what's inside the black box of models for understanding their intrinsic decision-making processes. Indeed, in many real-world scenarios, such as urban management, finance, and medical treatment, the lack of model explainability makes people less likely to be convinced when the decision-making process of the model is not understandable. This prevents a broader application of deep neural networks.

While many research efforts have been made in developing explainable deep learning models [40, 9], most of existing works focus on post-hoc explanation [4, 16], i.e., designing metrics to measure the feature relevance/contribution to the outputs of a trained model. Although these methods have made progress on finding important features, the decision-making process of deep learning models is still not available for users. For example, users often need to guess the reason why a prediction is made from the relevant features instead of understanding the decision-making process. Indeed, understanding the intrinsic decision-making process of deep neural networks is a non-trivial task. A major reason is that deep neural networks usually involve massive feature combinations to gain expressiveness on fitting complicated functions. During this process, the effect of features and hidden units may be largely coupled with each other. This indicates that the decision logic of models is inherently buried in the massive feature combinations. Meanwhile, it is difficult for human to understand the intrinsic modeling process of deep learning in a natural manner. Therefore, a key point on improving the explainability of deep neural networks is to decouple the feature combinations and make the modeling process consistent with human decision process. In this way, the model will have an explicit decision-making process and become human-understandable.

To this end, in this paper, we propose an explainable deep learning model, namely Voting Transformation-based Explainable Neural Network (VOTEN). Specifically, VOTEN assumes the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

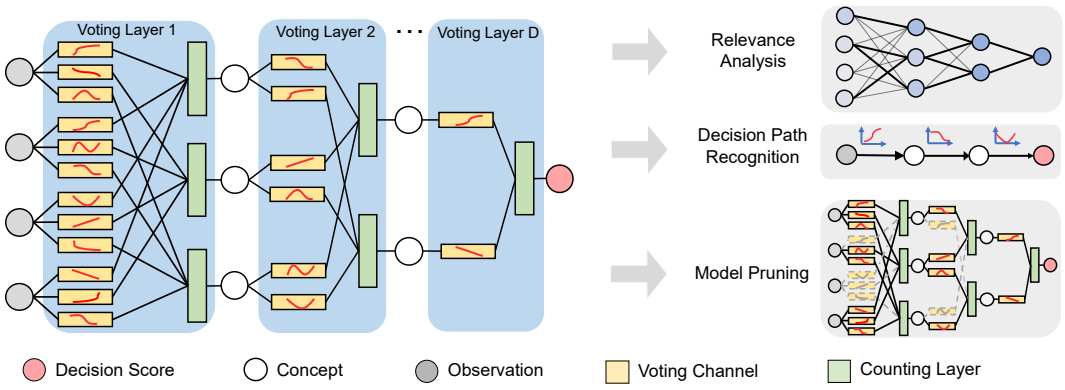


Figure 1: Structure overview of VOTEN.

transformation from the input to the output to be a hierarchical voting process. During this process, lower-level concepts vote for higher-level concepts layer-by-layer in an expressive but explainable way. Instead of relying on massive feature combinations, VOTEN creatively models expressive single-valued voting functions between explicitly modeled hidden concepts to gain expressiveness on fitting complicated functions. This process is explainable for its consistency with human decision-making process. We first theoretically analyze the major components of VOTEN and prove the relationship and advantages of VOTEN compared with Multi-Layer Perceptron (MLP), which is the basic structure of deep neural networks. The results show that MLP can be derived from VOTEN by using the inexpressive voting functions. Accordingly, we further analyze the effect of inherent votings and design efficient algorithms for pruning and explaining VOTEN. Finally, we evaluated VOTEN on multiple real-world datasets with comprehensive experiments. The experimental results demonstrate that VOTEN generally promotes powerful and explanability deep learning. Specifically, VOTEN (1) significantly raises prediction performance; (2) exponentially decreases feature combinations; and (3) supports efficient pruning and effective feature analysis. Meanwhile, we also visualize the intrinsic decision-making process of VOTEN through case studies, which show VOTEN is explainable and can discover meaningful latent concepts.

2 VOTEN

In this section, we introduce the technical details of VOTEN.

2.1 Structure

The structure overview of VOTEN is shown in Figure 1. Fundamentally, deep neural networks are difficult to be explained because they work in a totally different way from humans thinking. That is, while deep neural networks make the decision according to a unified, inseparable complicated function, humans usually aggregate information to infer intermediate concepts step-by-step. For example, a student spending a long time in the library is likely to be hard-working. A hard-working student may master knowledge well. Then, a student mastering knowledge well may get a high GPA. In this process, we use the concept "long time in the library" to infer the concept "hard-working", then use "hard-working" to infer the concept "master knowledge well". Finally, we infer "high GPA" with "master knowledge well". Along this line, human has specific way on raising our decision model's complexity and making more comprehensive decisions. Usually, we learn to infer from more views with more observations and hidden concepts. During this process, a concept may be inferred from multiple observations, and an observation may be used to infer multiple concepts. For example, "little missing of classes" may also imply "hard-working".

Existing works have made significant progress in designing tools to transform model logic to human-understandable logic. However, we believe a more natural way to raise model explainability is to make its intrinsic decision-making process consistent with humans. Therefore, motivated by human decision-making process, VOTEN explicitly models a small number of concepts and focuses on seizing the quantitative relationship between different levels of individual concepts. For better understanding, observations and concepts in VOTEN can be regarded as *voters* organized in a hierarchical structure. Each voter independently votes for some higher-level concepts based on their value. The higher-level concepts aggregate the votes to get their value estimation and further vote for the next level. During training, the model builds intermediate concepts and learns the quantitative relationship between concepts in adjacent levels for making proper votes.

Formally, for a VOTEN model with D levels of concepts, we use $\mathcal{C}_i^d$ to denote the $i\text{-}th$ concept in the $d\text{-}th$ layer, where $\mathcal{C}_i^0$ denotes an input feature. We refer to transformations between adjacent levels of concepts as a *voting layer*. In the $d\text{-}th$ voting layer, each concept $\mathcal{C}_i^d$ votes for each higher-level concept $\mathcal{C}_j^{d+1}$ with an independent *voting channel* $\mathcal{V}_{i,j}^d$. $\mathcal{V}_{i,j}^d$ takes the value of $\mathcal{C}_i^d$ as the input and votes with a single-valued nonlinear function $f_{i,j}^d : \mathbb{R} \to \mathbb{R}$. Then, a counting layer gets weighted average of the votes and estimates the value of $\mathcal{C}_j^{d+1}$ as

$$x_j^{d+1} = \sum_{k=1}^{n_d} a_{k,j}^d f_{k,j}^d(x_k^d) \qquad \text{s.t. } \forall d, j, \ \sum_{k=1}^{n_d} a_{k,j}^d = 1, \tag{1}$$

where n_d denotes the number of concepts in the $d\text{-}th$ layer, x_k^d denotes the value of $\mathcal{C}_k^d$, $a_{k,j}^d$ denotes the weight of $\mathcal{V}_{i,j}^d$. The concepts of the last layer is regarded as the decision score, which generates the model output with $\boldsymbol{o} = F^{VOTEN}(\boldsymbol{x}^D)$. In particular, to assure the ability of the voting functions on effective concept transformation, VOTEN models each voting function with a voting network. The voting network can be designed in complicated ways without influencing model explainability, as long as the function is still single-valued. For example, we can adopt weight-sharing structures to reduce model complexity.

2.2 Why is VOTEN more explainable than MLP?

In this part, we theoretically discuss VOTEN's advantages over MLP. Specifically, we claim that voting expressiveness is the core proposition of VOTEN that raises explainability.

Theorem 1 *MLP can be derived from a subset of degenerated VOTEN models whose voting functions in the form of $f_{i,j}^d(x_i^d) = w_{i,j}^d \sigma(x_i^d) + b_{i,j}^d$, where σ is a predefined activation function, the scalars $w_{i,j}^d, b_{i,j}^d \in \mathbb{R}$ are trainable parameters.*

Proof. Please refer to Appendix A.

From this point of view, MLP also can be regarded as consistent with human decision-making process. However, MLP is still recognized as difficult to explain. Empirically, concept-transformation-based decision-making is easier to understand with (1) fewer concepts, (2) shorter concept transformation paths, and (3) fewer decision-making patterns. Under VOTEN schema, we show how inexpressive voting channels make MLP disobey these three principles.

Corollary 1 *In MLP, votings from the same concept are linearly correlated.*

This means individual voters in MLP are weak in distinguishing different concepts in the next level. Therefore, MLP relies on highly complicated feature combinations of a large number of deeply tiled voters to achieve high fitting ability. During this process, necessary intermediate information is inherently modeled through combinatorial effects of hidden concepts with inexplicit meanings. Indeed, previous works have proved that human-understandable concepts are inherently mounted in the hidden units of neural network models [25].

Corollary 2 *In MLP, the hypothesis space for voting distribution is limited to scaling and shifting an input distribution.*

This means the voters in MLP may not always achieve significant transformation between different levels of concepts. Then, the input needs to go through a long path of transformations before it contributes to the output. Moreover, the deeply tiled massive concepts make each input feature has massive paths to the output, which brings a large number of possible decision-making patterns of the model. In addition, the effect of votings can easily couple and cancel each other out in the downstream calculations. As a result, separately analyzing the role of individual concepts or decision paths becomes meaningless.

Different from MLP, VOTEN has far more expressive voting functions. It directly models nonlinear transformations between essential intermediate concepts without relying on massive feature combinations and naive transformations. As a result, VOTEN's decision-making process is explicit with only a small number of meaningful concepts, thus is explainable.

2.3 Explaining the effect of votings

In VOTEN, individual voting channels play explicit roles in influencing the model decision. In this part, we analyze the effect of votings. First, we use a concept function $g_i^d : \mathbb{R}^{n_0} \to \mathbb{R}$ to represent the transformation from model inputs $\boldsymbol{x}^0 \in \mathbb{R}^{n_0}$ to $\mathcal{C}_i^d$, which decides the meaning of the concept.

Definition 1 *In VOTEN, we refer to two concept functions g and g' as equivalent iff. there exists an invertible function Φ, so that $\forall \boldsymbol{x}^0 \in \mathbb{R}^{n_0},\ \Phi(g(\boldsymbol{x}^0)) = g'(\boldsymbol{x}^0)$.*

Since Φ is invertible, the outputs of two equivalent concept functions have one-to-one correspondence over all the possible inputs. Then, they can effect equally in the decision-making process.

Theorem 2 *In VOTEN, if replacing a concept function with an equivalent form, there exists a way to replace its voting functions so that all the downstream concept functions stay unchanged.*

Proof. Please refer to Appendix A.

To analyze how voting channels effect on concept functions, we can reformulate Equation 1 as

$$x_j^{d+1} = g_j^{d+1}(\boldsymbol{x}^0) = \sum_{i=1}^{n_d} a_{i,j}^d (f_{i,j}^d(g_i^d(\boldsymbol{x}^0)) - \mathbb{E}_{\boldsymbol{x}}[f_{i,j}^d(g_i^d(\boldsymbol{x}))] + b_j^{d+1}, \tag{2}$$

where $\mathbb{E}_{\boldsymbol{x}}[f_{i,j}^d(g_i^d(\boldsymbol{x}))]$ denotes the expectation of $f_{i,j}^d$ over all the instances, b_j^{d+1} is a sample-independent bias. In particular,

$$b_j^{d+1} = \sum_{i=1}^{n_d} a_{i,j}^d \mathbb{E}_{\boldsymbol{x}}[f_{i,j}^d(g_i^d(\boldsymbol{x}))] = \mathbb{E}_{\boldsymbol{x}}[g_j^{d+1}(\boldsymbol{x})]. \tag{3}$$

For simplicity, we use $\overline{h_{i,j}^d}$ to denote $\mathbb{E}_{\boldsymbol{x}}[f_{i,j}^d(g_i^d(\boldsymbol{x}))]$. Notably, by adding an arbitrary bias to g_j^d, we obtain an equivalence of the original concept. According to Theorem 2, we can construct a model with exactly the same expression (i.e., equivalent concepts and the same predictions) as the previous one. This implies that VOTEN may converge to models with differed internal bias but exactly the same decision-making process, indicating VOTEN explanation should be independent of concept bias. According to Equation 2 and 3, voting channels' average only influence concept bias while the voting deviation $f_{i,j}^d(g_i^d(\boldsymbol{x}^0)) - \overline{h_{i,j}^d}$ reveals the effect of $\mathcal{V}_{i,j}^d$ for the decision. This can be intuitively explained as each voter can vote with different basic scores and only the deviation from the basic score reflects their judgement for a specific instance. Similarly, from the global view, we reformulate the concept function as $x_j^{d+1} = \sum_i a_{i,j}^d \sigma_{i,j}^d K_{i,j}^d(\boldsymbol{x}^0) + b_j^{d+1}$, where $K_{i,j}^d(\boldsymbol{x}^0) = \frac{f_{i,j}^d(g_i^d(\boldsymbol{x}^0)) - \overline{h_{i,j}^d}}{\sigma_{i,j}^d}$. Notably, $K_{i,j}^d(\cdot)$ generates a distribution with mean 0 and variance 1 over all the instances. Therefore, $a_{i,j}^d$ and $\sigma_{i,j}^d$ jointly decide the overall effect of votings. Specifically, the counting layer explicitly adjusts $a_{i,j}^d$ so that reliable voters have stronger influences. Meanwhile, the voter implicitly adjusts $\sigma_{i,j}^d$. When the concept is less related to the target concept and cannot support proper votes, they decrease $\sigma_{i,j}^d$ and tend to always vote the basic score to avoid disturbing the model. Otherwise, they increase $\sigma_{i,j}^d$ and vote confidently to lead the model to correct estimation.

Based on the above analysis, we can easily design algorithms to ease both VOTEN local and global explanation, such as recognizing decision paths and quantifying the concept/feature relevance to the prediction, which can be found in Appendix B and Appendix C.

2.4 VOTEN supports effective model pruning

In VOTEN, concepts are estimated by averaging the votes. This means we can delete a voting channel while keeping the physical meaning of the target concept unchanged. This supports effective link pruning. Specifically, since only the voting deviation from the average decides the effect, we can assume the absent channel votes the basic score regardless of the input. Formally, the value of $\mathcal{C}_j^{d+1}$ is estimated as $\hat{x}_j^{d+1} = \sum_{i=1}^{n_d} I_{i,j}^d a_{i,j}^d (f_{i,j}^d(x_i^d) - \overline{h_{i,j}^d}) + b_j^{d+1}$, where $I_{i,j}^d \in \{0, 1\}$ indicates if $\mathcal{V}_{i,j}^d$ is not absent. Furthermore, since only involving a small number of voting channels, we can achieve network pruning on VOTEN by exhaustively exploring how the performance will get influenced if some voting channels are absent, which is infeasible for MLP. In MLP, we usually prune unimportant hidden units to reduce model complexity. However, usually not all the voting channels from an important concept are necessary. In VOTEN, these unnecessary concept transformations can be further eliminated to not only reduce model complexity but also raises the explainability of the model. In Appendix D, we give an efficient VOTEN pruning algorithm with a lazy updating strategy.

3 Experiment

To evaluate the effectiveness and explainability of VOTEN for seizing comprehensive decision-making patterns. We conducted experiments with 4 large public datasets, including Context-aware

Table 1: Model Performance. We conducted 10 independent runs on each dataset and show the average $\pm$ standard deviation of AUC and AP. In particular, for multi-classifications, we estimated the macro average of each metric. We also did significance test, where * and ** denote significantly (i.e., p-value ≤ 0.05) and very significantly (i.e., p-value ≤ 0.01) worse than VOTEN.

Model	MR		RP		CT		CI	
	AP	AUC	AP	AUC	AP	AUC	AP	AUC
DecisionTree	0.2557^{**}	0.6762^{**}	0.0177^{**}	0.5134^{**}	0.8119^{**}	0.9396^{**}	0.2514^{**}	0.7250^{**}
$\pm$	0.0007	0.0096	0.0001	0.0014	0.0010	0.0005	0.0022	0.0018
RandomForest	0.4624^{**}	0.9007^{**}	0.0349^{**}	0.6566^{**}	0.9763^{**}	0.9979^{**}	0.6348^{**}	0.9388^{**}
$\pm$	0.0010	0.0006	0.0011	0.0052	0.0003	0.0001	0.0014	0.0005
LightGBM	0.4817^{**}	0.9209^{**}	0.0525^{*}	0.7305^{*}	0.9753^{**}	0.9965^{**}	**0.6972**	**0.9566**
$\pm$	0.0007	0.0001	0.0017	0.0016	0.0003	0.0001	0.0001	0.0001
MLP	0.4870^{**}	0.9199^{**}	0.0516^{**}	0.7250^{**}	0.9662^{**}	0.9965^{**}	0.6215^{**}	0.9454^{**}
$\pm$	0.0040	0.0008	0.0014	0.0017	0.0018	0.0002	0.0030	0.0004
VOTEN	**0.5007**	**0.9237**	**0.0550**	**0.7322**	**0.9783**	**0.9983**	0.6522	0.9508
$\pm$	0.0041	0.0005	0.0001	0.0003	0.0013	0.0001	0.0019	0.0002

Multi-Modal Transportation Recommendation (MR) [1, 48], IJCAI-18 Search Conversion Rate Prediction (RP) [2], Forest Cover Type Prediction (CT) [10], and Census-Income Prediction (CI) [34]. The detailed descriptions of experimental setup can be found in Appendix E.

3.1 Performance Evaluation: Can VOTEN achieve higher performance than MLP?

We used two widely adopted metrics for imbalanced classification performance evaluation, including Area Under ROC Curve (AUC) [12] and average precision (AP) [17], whose higher value means higher performance. In Table 1, we compare the performance of VOTEN with several baselines (see Appendix E.2). For each dataset, we have carefully tuned the parameters of the baselines to achieve their best performance. Especially, detailed analysis on MLP parameters can be found in Appendix E.3. It can be observed that, while deep neural networks are powerful when incorporated with purposely designed modules or prior knowledge for specific tasks, its standard form (i.e., MLP) without task-specific structures may perform worse than LightGBM, which have also been shown by many previous studies [41, 7, 3]. Indeed, LightGBM is believed to be powerful in handling structured data and often appears as the major model of the top solutions in data-mining competitions [48, 31]. In contrast, VOTEN, also in its standard form, significantly outperforms MLP for all these tasks, even better than LightGBM (except for CI, which contains mostly discrete features and is naturally suitable for tree-based models). This shows the effectiveness of VOTEN in terms of handling real-world problems. Indeed, VOTEN is more suitable for handling data-mining tasks since it is consistent with the practical decision-making process. It should be noticed that, similar to MLP, VOTEN is a standard and generic model that can be easily expanded to task-specific networks to raise model performance (for example, we can replace MLP with VOTEN in DeepFM [21]). While this paper focuses on the generic performance of standard VOTEN, it shows the possibility of building more powerful deep learning solutions for a wide range of applications.

3.2 Explanation Complexity: Is the decision-making process of VOTEN recognizable?

In Table 2, we compare the explanation complexity of MLP and VOTEN. As we have discussed in Section 2.2, we use the number of feature combinations, the length of decision paths, and the number of possible decision-making patterns to show the explanation complexity of a model. In particular, we trained two VOTEN models with different settings for each dataset. Specifically, "VOTEN$^-$" is comparable to the best performance of MLP, with the least feature combinations. "VOTEN" is the one with the best performance. It can be observed that VOTEN greatly reduces feature combinations and decision paths. For example, on the MR dataset, MLP needs 7 voting layers that each contain 128 concepts. This brings an exponentially large number of long decision paths. In contrast, VOTEN achieves comparable performance with 16 hidden concepts in total. Then, each feature only has 16 possible paths with a length of 2 to reach the output. This significantly eases the understanding of the decision-making process. For some datasets, VOTEN with no intermediate

Table 2: Explanation complexity. "#C/L" counts hidden concepts in each layer. "#P/F" counts possible decision paths from each feature.

Data	Model	Performance		#C/L	Depth	#P/F
		AP	AUC			
MR	MLP	0.487	0.920	128	7	2^{49}
	VOTEN$^-$	0.497	0.922	16	1	16
	VOTEN	**0.501**	**0.924**	16	2	256
RP	MLP	0.052	0.725	12	3	1,728
	VOTEN$^-$	0.055	0.732	0	0	1
	VOTEN	**0.055**	**0.732**	8	2	64
CT	MLP	0.966	0.996	128	7	2^{49}
	VOTEN$^-$	0.969	0.997	32	2	1,024
	VOTEN	**0.978**	**0.998**	64	2	4,096
CI	MLP	0.622	0.945	32	2	1,024
	VOTEN$^-$	0.652	0.950	0	0	1
	VOTEN	**0.652**	**0.951**	4	2	16

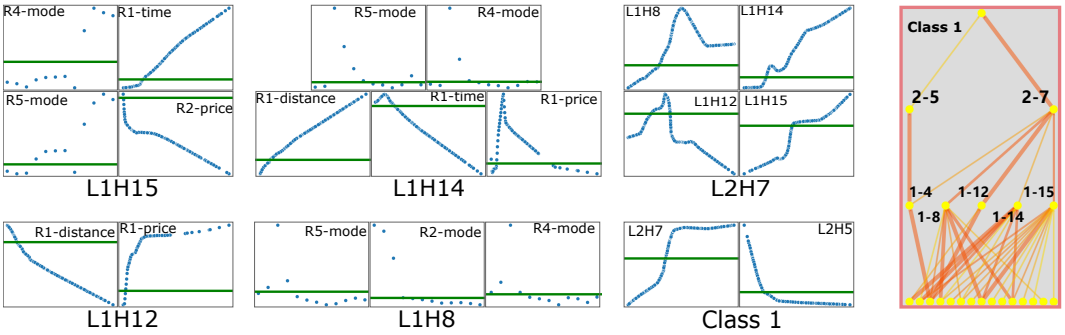


Figure 2: Case study of VOTEN global decision-making process in MR dataset. We show the important paths from inputs to one output, where wide lines indicate important voting channels. The concept LxHy indicates the y-th concept in the x-th layer. We visualize the important voting functions for each concept along the paths relevant to L2H7, where blue lines show the function while green lines show the average vote.

concepts (i.e., features directly vote for the prediction) can achieve comparable performance to MLP. Moreover, even when reaching its best performance, VOTEN still has a much smaller explanation complexity than MLP. It should be noticed that we have listed the maximum possible number of decision paths for convincing illustration. In practice, we can easily distinguish a fewer number of important decision paths, which will be shown in the following experiments.

3.3 Case Study: How to explain a VOTEN model?

We show how to explain a VOTEN model with an example in the MR dataset. Specifically, we first analyze the global decision-making process and discover the meaning of concepts by observing voting functions on important decision paths. Then, we locally explain how the inputs of an instance hierarchically vote the final prediction.

Task Description. The task is to recommend the transport mode for online map app users, given a user and an Origin-Destination (OD) pair. The features mainly contain user portraits and an ordered list of recommended plans of the map app. Each plan consists of transport mode, time, distance, and price. We deleted the first recommended plan's mode information since it is too strongly correlated with the label (many users choose the first recommendation as default). In this way, we can better observe how the model incorporates complicated information for meaningful decisions.

Global Decision-Making Process. We focus on the decision-making process for the class "Subway". First, we discover the important transformations from observations to the prediction with our decision path recognition algorithm. The results are shown in Figure 2. Next, we analyze the meaning of concepts from the bottom to the top. L1H8 gets larger when modes of more recommended plans are "Subway", which we regard as "*OD pair with flexible subway-based plans*". L1H12 decreases with higher prices and lower distances, which we regard as "*OD pair's distance-cost performance*". L1H14 observes if the OD pair is distant but still available with inexpensive and fast transportation. Besides, it also observes if modes of public transportation ever appear in the plan list. Therefore, we regard L1H14 as "*distant OD pair with convenient and economical public transportation*". L1H15 is sensitive to a time-consuming top-1 plan. It also observes the other plans' modes and prices to estimate if trading money for efficiency is infeasible. Therefore, we regard L1H15 as "*no choice but a time-consuming transportation*". In the second layer, L2H7 is estimated based on L1H8, L1H12, L1H14, and L1H15. It gets higher if many subway-based plans available (higher L1H8), transportation with balanced distance-cost is recommended (has a peak for L1H12), distant OD pair but still has convenient public transportation (higher L1H14), or costly time-saving transportation is infeasible (higher L1H15). Comprehensively considering these reasons, it implies "*OD pair suitable for subway transportation*, which votes for the score of "Subway" with a monotonic transformation. Along the other path, L1H4 is a concept "*OD pair with inexpensive transportation*". Then, L2H5 is also about the price since it is mainly based on L1H4. It should be noticed that its estimation may still be enhanced by other lower-level concepts when dealing with specific instances, although they are less important from the global view. Finally, L2H5 and L2H7 vote nonlinearly to the score so that the model can make accurate quantitative predictions. With the above analysis, we can qualitatively understand the logic of VOTEN on recommending "Subway". Actually, the decision-making process is quantitatively more complicated and can handle more special cases. In practice, domain experts

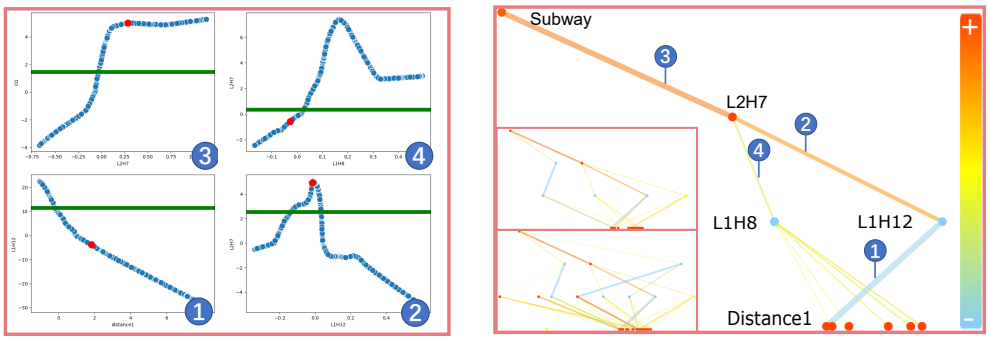


Figure 3: Case study of VOTEN local decision-making process in MR dataset, where wider and darker lines indicate stronger influence (negatively in blue and positively in red). The bottom left shows filtered paths when we gradually bring up the thresholds. We also show the important voting functions, where the green lines show the average and red points show vote for the current instance.

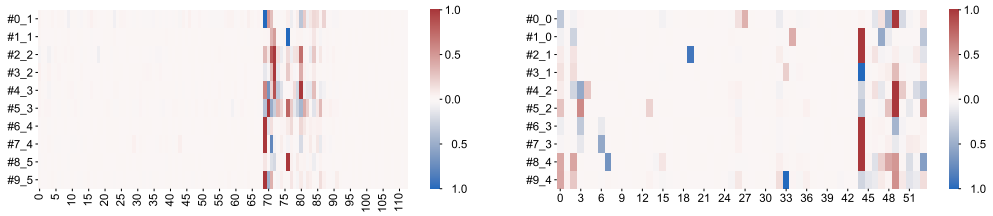
can thoroughly analyze the shape and gradients of the voting functions to get more insights into the concepts. This may help researchers to find new concepts and develop new theories, especially in fields such as psychology and management, where scientists work on finding mechanisms linking observations to outcomes. Extra visualizations on global decision path and voting functions can be found in Appendix F and Appendix G.

Local Decision-Making Process. Then, we analyze the decision paths for specific instances. In Figure 3, we visualize the most important paths for a sample with a high score for "Subway", which are filtered with a small threshold in our local decision path recognition algorithm. Based on the global analysis, we can easily tell how the observations gradually transformed to higher-level concepts. Specifically, for L1H12, the vote from price is filtered out, showing the OD pair's price is normal over the dataset. However, it finds the OD pair to be distant, which contributes to a relatively larger *distance-cost performance* than average, given the normal price. This indicates the plan to be relatively more economical. L2H7 observes the value of L1H12, and finds that the OD pair can choose transportation whose *distance-cost performance* more balanced than average cases (reaches the peak of voting). In this case, L2H7 votes high for subway, which is an economical and balanced transportation mode. On the other decision path, L1H8 thinks "Subway" may not be suitable since subway-based plans seldom appear in the list, thus votes negatively for L2H7. But L1H12 makes a very confident judgment based on the balance of distance-price performance, which dominates L2H7 and makes the model predict correctly. Interestingly, we find one of the most important strategies in this task is to guess the transportation mode most recommended by the app (the information that we hide in prior), which is reasonable. On the one hand, the app's recommender system trained with abundant information can naturally achieve high performance. On the other hand, many users will click the first recommendation as default. In addition to this strategy, the model will use more important decision paths to achieve more accurate predictions. As we gradually increase the threshold, more decision-making patterns appeared. Extra visualizations on local explanation of other datasets can be found in Appendix H.

3.4 Relevance Analysis: Can VOTEN help quantify feature relevance?

Propagation-Based Relevance. Motivated by relevance propagation [9], we propose an algorithm (see Appendix D) to quantify the relevance of features and concepts, based on the important decision paths. The short decision paths of VOTEN decreases error accumulation during the propagation and enables more accurate relevance estimation. In Figure 4, we visualize the propagation-based feature relevance with heatmaps. Obviously, features from #69 to #86 are generally important in the MR dataset, among which the first several features (information about the top-1 plan in the list) are the most relevant. Furthermore, the relevant features vary for different samples in terms of different classes, which indicates VOTEN to predict in multiple patterns. For example, Figure 4(b) shows that VOTEN assigns high score to class 2 for sample 4 and sample 5 with different reasons. Specifically, feature #3 and #53 contribute positively for sample 5 while negatively for sample 4. Instead, #50 is more positively relevant for sample 4. Extra visualizations can be found in Appendix I.

Single-Sighted Prediction Strength. We can also estimate the feature relevance from the view of model performance when the prediction is supported by a single voter in some layer. Specifically, we disable the other voters in a similar way as we do in model pruning. Table 3 shows 5 MR features achieving the highest AUC when conducting single-sighted prediction for class 0 and



(a) MR (b) CT

Figure 4: Heatmap for the propogation-based relevance in VOTEN. The x-axis represents features and the y-axis represents the instance-output pairs, where $\#I\_O$ indicates the feature relevance of the I-th instance to class O. Red means positive relevance while blue means negative relevance.

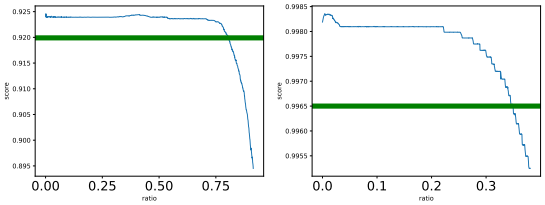
Table 3: Features with top-5 single-sighted prediction strength for class 0 and class 1 of MR dataset in VOTEN. "Input" indicates the AUC of ranking the samples according to the feature's value.

Model	Class 0					Class 1				
	Top-1	Top-2	Top-3	Top-4	Top-5	Top-1	Top-2	Top-3	Top-4	Top-5
VOTEN	0.675	0.653	0.647	0.606	0.592	0.747	0.632	0.625	0.587	0.566
Input	0.596	0.533	0.572	0.600	0.586	0.535	0.527	0.534	0.555	0.525

class 1 in VOTEN. As a comparison, we also show the AUC of directly ranking with the feature, which reveals the feature's linear correlation with the prediction. In particular, since there can be negative correlations, we evaluate AUC for the rank in increasing and decreasing order, and use the larger one as the performance. Moreover, Figure 5 shows the results on all the features. It can be observed that, even if approximated into single-sighted, VOTEN significantly raises AUC, showing VOTEN to recognize important features and strengthen their effectiveness with nonlinear transformations. Especially, while feature #85 originally seems not correlated with the output, VOTEN finds it in practice nonlinearly very relevant to the output. This proves the effectiveness of VOTEN on quantifying the inherent nonlinear relationships between the observations and the prediction. Interestingly, VOTEN weakens the effect of some features (e.g., feature #3 for RP) to prevent them from disturbing the prediction. Extra visualizations can be found in Appendix J.

3.5 Pruning Experiment

We conducted pruning experiments on VOTEN for MR and CT datasets with our pruning algorithm. As we gradually deleted voting channels, we monitored the change of AUC during this process, which are shown in Figure 6. For MR, AUC is still near 0.924 after pruning nearly half of the voting channels. For CT, AUC is still near 0.998 after pruning a quarter of the channels. Interestingly, proper pruning may slightly raise model performance, which is reasonable as a simple model has less chance of over-fitting. In practice, operations like fine-tuning can be adopted to further raise the performance of the pruned VOTEN model. Then, the model can be further compressed without affecting the prediction much. These results prove VOTEN to support effective pruning, which is helpful. We can use complicated information for training and easily prune the model to decrease the complexity both for storage, calculation, and explanation.

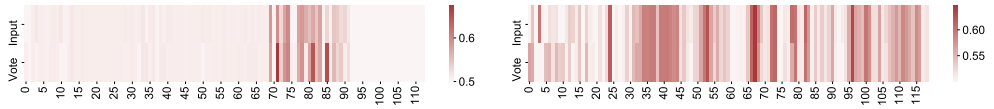


(a) MR (b) CT

Figure 6: VOTEN pruning experiments. The x-axis shows the ratio of pruned channels while the y-axis shows AUC. The green line shows MLP's performance.

4 Related Work

Post-Hoc Deep Learning Explanation. Post-Hoc explaining algorithms analyze the relevance of features in a model-free way, mainly including propagation-based methods [27, 5] and perturbation-based methods [49, 46]. Propagation-based methods propagate the relevance score backward to the inputs. For example, Simonyan *et al.* [40] generates saliency maps with the gradients of the output category with respect to the inputs. Bach *et al.* [9] proposed Layer-wise Relevance Propagation (LRP), which designed effective rules for the propagation. Perturbation-based methods explain model behavior by observing how the output reacts to purposely perturbed or constructed inputs. For example, Local Interpretable Model-Agnostic Explanations (LIME) [38] trains a local explainable



(a) MR (Class 0) (b) RP

Figure 5: Single-sighted prediction strength in VOTEN. We also show the AUC of linearly ranking the samples as comparisons. Darker color means higher AUC.

approximation model around the prediction with randomly perturbed features and the corresponding outputs. SHapley Additive exPlanations (SHAP) [30] estimates the Shapley value of features to measure their contribution to model performance. In addition to features, some works estimate concept importance for a model [20, 43, 25]. For example, Kim *et al.* [25] learn the representation of human-understandable concepts with labeled concept-relevant examples and estimate concept sensitivity according to the directional derivative towards the concepts. Along this line, abundant works have been proposed to further raise the effectiveness of post-hoc interpretation algorithms [28, 45, 13]. However, these algorithms regard models as blackboxes and heuristically explain with their own metrics, which cannot give explicit understandings of the actual decision-making process. Different from existing works, we proposed a naturally understandable neural network model.

Intrinsically Explainable Machine Learning Techniques. Intrinsically explainable models can be explained without relying on post-hoc algorithms [8, 4], mainly including classic models such as logistic regression [33], linear support vector machine [23], decision trees [39], generalized additive models [22] and Bayesian models [37]. Recently, Agarwal *et al.* [6] proposed Neural Additive Model that predicts with a linear combination of neural networks. However, all these models usually have tight restrictions on the hypothesis space, which limits their fitting ability on complicated real-world problems. Based on these methods, complicated models are developed for higher performance. However, even with intrinsically explainable base models, these complicated models still need post-hoc algorithms for explanation [8]. For example, ensemble tree models [14, 24] predict with a large number of weak learners. However, the joint decision-making process of massive decision trees is difficult to understand. Kernel functions [35] are incorporated in support vector machines to seize high dimensional feature interactions. However, the dimension transformation is implicit and not understandable. In recent years, researchers also try to design explainable neural network models by incorporating purposely designed task-specific constraints or structures [15, 47, 36]. For example, Qin *et al.* [36] proposed a person-job fit network, which used attention layers to find important talent working experience for different job requirements. However, these models cannot be adopted by the general tasks. Besides, they only provide heuristic and domain-specific intermediate information instead of telling the complete decision-making process. Different from these works, we aim at a general explainable neural network model, which has an intrinsically explainable decision-making process while retaining the high fitting ability.

5 Concluding Remarks

In this paper, we have proposed an explainable deep learning model, VOTEN. Specifically, we theoretically analyzed the major components of VOTEN and discussed its priority over MLP, and accordingly proposed some efficient algorithms to raise the model usability. Experimental results on multiple real-world datasets clearly demonstrated that VOTEN can significantly improve the explainability and performance of deep learning.

Limitations. In this paper, we focused on comparing VOTEN with MLP, which is the generic and basic structure of deep learning models. Many powerful problem-specific structures can be derived from MLP by adding operations such as weight sharing (e.g., CNN). Similar to MLP, VOTEN is a basic and generic structure. It can be adopted to problem-specific models (e.g., we can simply use VOTEN to replace MLPs in deepFM [21] or MMoE [32]). Indeed, recent studies show that if properly designed, simple MLP-based structure achieves comparable performance to complicated SOTA models [19, 42]. VOTEN's advantages over MLP provides great possibility on further improving a wide range of deep learning applications. In the future, we will also explore building VOTEN-based task-specific structures. In addition, since VOTEN automatically extracts concepts during training, human effort is needed to observe the voting functions for understanding the concepts, which is a common issue in unsupervised concept modeling, such as Latent Dirichlet Allocation [11]. In the future, we will work on easing the concept understanding of VOTEN, such as recognizing concept-related samples or aligning VOTEN with human-understandable concepts.

References

[1] Context-aware multi-modal transportation recommendation. https://dianshi.bce.baidu.com/competition/29/question.

[2] Ijcai-18 alimama sponsored search conversion rate(cvr) prediction contest. https://tianchi.aliyun.com/competition/entrance/231647/information.

[3] Lightgbm vs neural network. https://mljar.com/machine-learning/lightgbm-vs-neural-network/.

[4] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[5] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[6] Rishabh Agarwal, Nicholas Frosst, Xuezhou Zhang, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *arXiv preprint arXiv:2004.13912*, 2020.

[7] Aneseh Alvanpour, Sumit Kumar Das, Christopher Kevin Robinson, Olfa Nasraoui, and Dan Popa. Robot failure mode prediction with explainable machine learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 61–66. IEEE, 2020.

[8] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[9] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[10] Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

[11] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[12] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[13] Steven Bramhall, Hayley Horn, Michael Tieu, and Nibhrat Lohia. Qlime-a quadratic local interpretable model-agnostic explanation approach. *SMU Data Science Review*, 3(1):4, 2020.

[14] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[15] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 765–774, 2019.

[16] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020.

[17] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[19] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv e-prints*, pages arXiv–2105, 2021.

[20] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019.

[21] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[22] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

[23] Shujun Huang, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, and Wayne Xu. Applications of support vector machine (svm) learning in cancer genomics. *Cancer Genomics-Proteomics*, 15(1):41–51, 2018.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[25] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[27] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[28] Heyi Li, Yunke Tian, Klaus Mueller, and Xin Chen. Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. *Image and Vision Computing*, 83:70–86, 2019.

[29] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1754–1763, 2018.

[30] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[31] Zhipeng Luo, Jianqiang Huang, Ke Hu, Xue Li, and Peng Zhang. Accuair: Winning solution to air quality prediction for kdd cup 2018. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1842–1850, 2019.

[32] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1930–1939, 2018.

[33] Edward C Norton, Bryan E Dowd, and Matthew L Maciejewski. Marginal effects—quantifying the effect of changes in risk factors in logistic regression models. *Jama*, 321(13):1304–1305, 2019.

[34] Nikunj C Oza and Stuart Russell. Experimental comparisons of online and batch versions of bagging and boosting. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 359–364, 2001.

[35] Luis Carlos Padierna, Martin Carpio, Alfonso Rojas-Dominguez, Hector Puga, and Hector Fraire. A novel formulation of orthogonal polynomial kernel functions for svm classifiers: the gegenbauer family. *Pattern Recognition*, 84:211–225, 2018.

[36] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 25–34, 2018.

[37] Adrian E Raftery. Bayesian model selection in social research. *Sociological methodology*, pages 111–163, 1995.

[38] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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*, pages 1135–1144, 2016.

[39] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[41] Ying Sun, Fuzhen Zhuang, Hengshu Zhu, Qi Zhang, Qing He, and Hui Xiong. Market-oriented job skill valuation with cooperative composition neural network. *Nature Communications*, 12(1):1–12, 2021.

[42] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[43] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Towards global explanations of convolutional neural networks with concept attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8652–8661, 2020.

[44] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[45] Muhammad Rehman Zafar and Naimul Mefraz Khan. Dlime: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*, 2019.

[46] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[47] Yingying Zhang, Shengsheng Qian, Quan Fang, and Changsheng Xu. Multi-modal knowledge-aware hierarchical attention network for explainable medical question answering. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1089–1097, 2019.

[48] Wenjun Zhou, Taposh Dutta Roy, and Iryna Skrypnyk. The kdd cup 2019 report. *ACM SIGKDD Explorations Newsletter*, 22(1):8–17, 2020.

[49] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We have described the limitations and future works in Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work aims to raise the explainability of deep learning. There is no potential negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] No assumptions for theoretical analysis in this paper.
   (b) Did you include complete proofs of all theoretical results? [Yes] We have provided complete proofs for our theoretical results in Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The supplemental material contains code and instructions to reproduce the main experimental results of this paper. The original public dataset can be downloaded by readers on their own, which we have provided the URLs in the code package.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We have specified these training details, please refer to Appendix E.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We have reported error bars after running experiment multiple times. In addition, we have conducted significant test. Related information can be found in Table 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We have reported these information, please refer to Appendix E.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Our work use public dataset. We have cited the creators.
   (b) Did you mention the license of the assets? [No] We are sorry that we cannot find the license of the datasets. Specifically, MR and PR can be downloaded from the competition and CT and CI are publicly available in UCI repository. The licenses are not specified.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We have submitted our code as supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We have mentioned that we use public datasets, please refer to Section 3 and Appendix E.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Our data contain no such information. We have emphasized that all the data are anomalous, please refer to Appendix E.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No human subject involved.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No human subject involved.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No human subject involved.