SOINTER: A NOVEL DEEP ENERGY-BASED INTERPRETATION METHOD FOR EXPLAINING STRUCTURED OUTPUT MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a novel interpretation technique to explain the behavior of structured output models, which learn mappings between an input vector to a set of output variables simultaneously. Because of the complex relationship between the computational path of output variables in structured models, a feature can affect the value of output through other ones. We focus on one of the outputs as the target and try to find the most important features utilized by the structured model to decide on the target in each locality of the input space. In this paper, we assume an arbitrary structured output model is available as a black box and argue how considering the correlations between output variables can improve the explanation performance. The goal is to train a function as an interpreter for the target output variable over the input space. We introduce an energy-based training process for the interpreter function, which effectively considers the structural information incorporated into the model to be explained. The effectiveness of the proposed method is confirmed using a variety of simulated and real data sets.

1 INTRODUCTION

The impressive prediction performance of novel machine learning methods has motivated researchers of different fields to apply these models in challenging problems. However, their complex and non-linear inherence limit the ability to explain what they have learned. Interpretation gets more attention when we want to discover the reasons behind the model's decision and be sure about the trustworthiness and fairness of a trained machine learning model in areas such as medicine, finance, and judgment. Additionally, interpreting a model with a satisfying prediction accuracy in a scientific problem, which results in understanding relationships behind the data, leads to new knowledge about the problem domain.Murdoch et al. (2019)

In many real-world applications, the goal is to map an input variable to a high-dimensional structured output, e.g., image segmentation and sequence labeling. In such problems, the output space includes a set of statistically related random variables. As considering these dependencies can increase the prediction accuracy, many structured output models have been introduced. Many of these methods use graphical models, including random fields, to capture the structural relations between variables. Most define an energy function over these random fields, with a global minimum at the ground truth. Therefore, an inference is needed to find the best configuration of output variables for input by minimizing the energy function in the prediction step. Early efforts to utilize deep neural networks in structured output problems adopt deep networks to extract high-level features from the input vector to incorporate them in calculating the energy function Peng et al. (2009); Chen et al. (2015); Schwing & Urtasun (2015). The computational complexity of the inference step in models that use random fields limits their ability to incorporate complex structures and interactions between output variables. Recent works in Belanger & McCallum (2016); Gygli et al. (2017); Belanger et al. (2017); Graber et al. (2018) propose to adopt deep networks instead of random fields to model the structure of the output space. Nevertheless, complex interactions between problem variables in such models make their interpretation too challenging, specifically when we focus on the model behavior in predicting a single output variable.

This paper attempts to interpret a structured output model by focusing on each output variable separately. Our approach to model interpretation is based on instance-wise feature selection. Its goal is to find the relative importance of each input feature in predicting a single output variable. The subset of important features can vary across the input space. The complicated interactions between computational paths of output variables in structured output models cause critical challenges for finding a subset of important features associated with each output variable. A feature may not be used directly in the computational path of output but affects its value through relations with other outputs. To compute the importance of a feature for a target output variable, we should aggregate its effect on all output variables that are correlated to this target.

Existing approaches of model interpretation can be divided into two groups, model-based and post hoc analysis Murdoch et al. (2019). The model-based interpretation approach encourages machine learning methods that readily provide insight into what the model learned. However, it leads to simple models that are not sufficiently effective for complex structured output problems. Here we follow the post hoc analysis and try to explain the behavior of a trained, structured output model provided as a black box. Many interpretation techniques to find the importance of features as a post hoc analysis have been introduced. Works in Zhou & Troyanskaya (2015); Zeiler & Fergus (2013); Zintgraf et al. (2017) make perturbations to some features and observe their impact on the final prediction. These techniques are computationally inefficient in situations where we search for the most valuable features. Since we should perform a forward propagation for all possible perturbations, in another trend, works in Simonyan et al. (2013); Bach et al. (2015) back-propagate an importance signal from the target output through the network to calculate the critical signal of features by calculating the gradient of the target w.r.t the input features. These models are computationally more efficient than perturbation-based techniques because they need only one pass of propagating. However, they need the structure of the network to be known. As this approach may cause a saturation problem, DeepLIFT Shrikumar et al. (2017) proposes that instead of propagating a gradient signal, the difference of the output from a reference value in terms of the difference of features from a reference value to be considered. In addition to these approaches, other ideas have also been introduced in model interpretation. Authors in Ribeiro et al. (2016) introduce LIME which trains a local interpretable surrogate model to simulate the behavior of a black box model in the vicinity of a sample. It randomly selects a set of instances of the input space around that sample and obtains the black box prediction for them, and trains the surrogate model by this new dataset. Therefore this interpretable model is a good approximation of the black box around the locality of the selected sample. Shapley value, a concept from the game theory, explains how to distribute an obtained payout between coalition players fairly. The work in Lundberg & Lee (2017) proposes the kernel SHAP for approximating the shapely value for each feature as its importance for a prediction. As an information-theoretic perspective on interpretation, the work in Chen et al. (2018) proposes to find a subset of features with the highest mutual information with the output. This subset is expected to involve the most important features for the output.

Existing interpretation techniques can be applied to explain the behavior of a structured model, w.r.t. a single output, by ignoring other output variables. However, none of these approaches consider possible correlations between output variables and only analyze the marginal behavior of the black box on the target. In this paper, we attempt to incorporate the structural information between output variables during training the interpreter. As our goal is to present a local interpreter, which is trained globally as Chen et al. (2018), we train a function over the input space which returns the index of most important features for decision making about the target output. Since the value of other output variables affects the value of the target, incorporating them into the training procedure of an interpreter function may lead to higher performance and decrease our uncertainty about the black box behavior. To the best of our knowledge, this is the first time an interpreter is designed mainly for structured output models, and dependencies between output variables are considered during training the interpreter.

2 Preliminaries and Motivation

Structured output prediction models map an arbitrary n-dimensional feature vector $\mathbf{x} \in \mathcal{X}$ to the output $\mathbf{y} \in \mathcal{Y}$ where $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_d]$ includes a set of correlated variables with known and unknown complex relationships and $\mathcal{Y}$ shows a set of valid configurations.

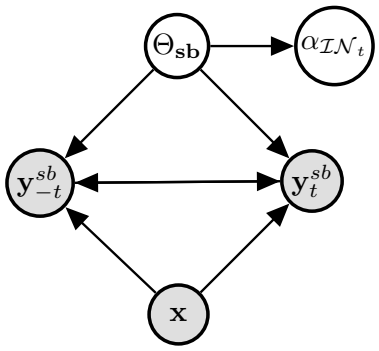

Figure 1: The generative relationship between problem variables.

Now we explain our intuition about an interpreter, which explains the behavior of a structured output model in predicting a single output variable. We assume a structured model is available as a black box about which we do not know. Our goal is to find indices of k important features of $\mathbf{x}$ which affect the black box prediction about the target output $\mathbf{y}_t$. As for different localities of the input space these indices may vary, the proposed interpreter is a function $\mathcal{IN}_t(\mathbf{x};\alpha) : \mathcal{X} \rightarrow \{0,\ 1\}^n$ over the input space with a set of parameters α which returns an n-dimensional k-hot vector. In this vector, the value of 1 shows the indices of selected k important features for target output $\mathbf{y}_t$.

We define $\Theta_{\mathbf{sb}}$ as the set of all parameters and hidden variables inside the structured black box. The probabilistic graphical model of Fig. 1 describes dependencies between problem variables. In this figure, $\mathbf{x}$ shows the input variable, $\mathbf{y}_t^{sb}$ and $\mathbf{y}_{-t}^{sb} = \{\mathbf{y}_i^{sb} | i \neq t\}$ show black box predictions and $\alpha_{\mathcal{IN}_t}$ is the set of parameters of $\mathcal{IN}_t$. The bidirectional edge between $\mathbf{y}_t^{sb}$ and $\mathbf{y}_{-t}^{sb}$ emphasizes the correlation between the outputs of a structured model. In fact $\alpha_{\mathcal{IN}_t}$ is determined based on $\Theta_{\mathbf{sb}}$ and the black box architecture, and the final prediction of the $\mathbf{y}_t^{sb}$ does not directly affect its value. However, here, $\Theta_{\mathbf{sb}}$ is a latent variable which makes active paths between $\alpha_{\mathcal{IN}_t}$ and output values $\mathbf{y}_t^{sb}$ and $\mathbf{y}_{-t}^{sb}$. Therefore $\alpha_{\mathcal{IN}_t}$ and $\mathbf{y}_{-t}^{sb}$ are dependent random variables and we have:

$$H(\alpha_{\mathcal{IN}_t} | \mathbf{x}, \mathbf{y}_t^{sb}) > H(\alpha_{\mathcal{IN}_t} | \mathbf{x}, \mathbf{y}_t^{sb}, \mathbf{y}_{-t}^{sb}) \tag{1}$$

where $H(.|.)$ shows the conditional entropy. We use the strict inequality because $\alpha_{\mathcal{IN}_t}$ and $\mathbf{y}_{-t}^{sb}$ are dependent random variables. The left term measures our uncertainty when we train the interpreter only by observing the target output $\mathbf{y}_t^{sb}$. This inequality confirms that the uncertainty is decreased when we consider observed $\mathbf{y}_{-t}^{sb}$ during estimating $\alpha_{\mathcal{IN}_t}$. Motivated by this fact we propose a training procedure for an interpreter $\mathcal{IN}_t$ which incorporates the structural information of the output space by observing the black box prediction on all output variables.

We call our method SOInter as we propose it to train an Interpreter specifically for Structured Output models.

3 Proposed Method

We consider $p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x})$ as the distribution by which the structured black box predicts the output as follows,

$$\mathbf{y}^{sb} = \arg\max_{\mathbf{y}} p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x}). \tag{2}$$

Our goal is to train the interpreter $\mathcal{IN}_t(\mathbf{x};\alpha)$ which explores a subset of most important features that affects the value of black box prediction on the target output $\mathbf{y}_t$ in each locality of the input space. As the desired interpreter detects the subset of most important features, we expect perturbating other ones does not change the black box prediction of the target $\mathbf{y}_t$. Motivated by this statement, we are encouraged to compare the black box prediction for the target output when a sample and its perturbated version are passed through the black box. The interpreter $\mathcal{IN}_t(\mathbf{x};\alpha)$ returns a k-hot vector in which the value of 1 shows the index of a selected feature. We define $\tilde{\mathbf{x}}$, a perturbated

version of $\mathbf{x}$, as follows,

$$\tilde{\mathbf{x}} = \mathbf{x} \odot \mathcal{IN}_t(\mathbf{x}; \alpha) \tag{3}$$

in which only features selected by the interpreter are included, and other ones are filled with zeros. Therefore by passing $\mathbf{x}$ and $\tilde{\mathbf{x}}$ through the black box, we expect the value of the tth element of predictions to be the same, and we can define a penalty over the value of the target in these two situations. However, since the structure of the black box is unknown and the $\arg\max$ operation in equation 2 is non-differentiable, a loss function that directly compares these two output values can not be used to find the optimal interpreter. Therefore, in the following subsection, we try to achieve a penalty according to the difference between these values for the target, which can transfer the gradient through an interpreter block.

3.1 Obtaining a tractable loss function

We consider $\tilde{\mathbf{y}}$ as the black box prediction when masked inputs $\tilde{\mathbf{x}}$ are given to the black box i.e.,

$$\tilde{\mathbf{y}} = \arg\max_{\mathbf{y}} p_{\mathbf{sb}}(\mathbf{y}|\tilde{\mathbf{x}}). \tag{4}$$

We define distribution q_α to estimate the probability density function of $\tilde{\mathbf{y}}$. We assume q_α is from the exponential family and can be parameterized as follows,

$$q_\alpha(\mathbf{y}|\mathbf{x}) = \frac{\exp(-\mathcal{E}_q(\tilde{\mathbf{x}},\ \mathbf{y}))}{\int_{\mathbf{y}\prime} \exp(-\mathcal{E}_q(\tilde{\mathbf{x}},\ \mathbf{y}\prime))} \tag{5}$$

As we define q_α the distribution over the output variables when the masked input is given to the black box, $\mathcal{E}_q$ can be considered as an energy function describing the structural information incorporated into the black box. We train a deep neural network to estimate this energy block, using a set of samples from the input space and their associated outputs predicted by the black box. For an input feature $\mathbf{x}$, the obtained energy function $\mathcal{E}_q(\mathbf{x},\ \mathbf{y})$ has the minimum value when $\mathbf{y}$ is equal to the black box prediction for $\mathbf{x}$. The detailed architecture of this block depends on the data inherence. Different techniques to train an energy network have been introduced recently Belanger & McCallum (2016); Belanger et al. (2017); Gygli et al. (2017) which can be used to train $\mathcal{E}_q$ in a pre-training phase. Here we use the work in Gygli et al. (2017). Since after the pre-training phase, the only unknown parameters of q_α are parameters of $\mathcal{IN}_t$, we show it by the index of α.

Now to find the optimum value for α, we define the following penalty over the log-likelihood of q_α,

$$\mathcal{SL}(\mathbf{x},\ \mathbf{y}_t^{sb},\ \tilde{\mathbf{y}};\ \alpha) = \max\{0,\ (\log q_\alpha([\tilde{\mathbf{y}}_t,\ \tilde{\mathbf{y}}_{-t}]\,|\mathbf{x}) - \log q_\alpha(\left[\mathbf{y}_t^{sb},\ \tilde{\mathbf{y}}_{-t}\right]|\mathbf{x}) + \mathcal{L}(\tilde{\mathbf{y}}_t, \mathbf{y}_t^{sb}))\}. \tag{6}$$

When the target values for an input $\mathbf{x}$ and its perturbated version $\tilde{\mathbf{x}}$ are the same, i.e. $\tilde{\mathbf{y}}_t = \mathbf{y}_t^{sb}$, the penalty value is zero. Otherwise, as $\tilde{\mathbf{y}}$ maximizes q_α, $\mathcal{SL}(\mathbf{x},\ \mathbf{y}_t^{sb},\ \tilde{\mathbf{y}};\ \alpha)$ has a positive value which is proportional to the difference between the desired target value $\mathbf{y}_t^{sb}$ and the most offending answer $\tilde{\mathbf{y}}_t$. Using the log-likelihood of the joint distribution $q_\alpha(.|\mathbf{x})$ in calculating the penalty, which $\tilde{\mathbf{y}}_{-t}$ is the most probable value for other outputs according to the q_α, leads to incorporate the structural information between output variables.

Now we attempt to achieve a more simpler form of equation 6. According to equation 5 the log-likelihood of q_α equals to,

$$\log q_\alpha(\mathbf{y}|\mathbf{x}) = -\mathcal{E}_q(\tilde{\mathbf{x}},\ \mathbf{y}) - \log \int_{\mathbf{y}\prime} \exp(-\mathcal{E}_q(\tilde{\mathbf{x}},\ \mathbf{y}\prime)). \tag{7}$$

Because the second term of equation 7 for both of $\log q_\alpha(\left[\mathbf{y}_t^{sb},\ \tilde{\mathbf{y}}_{-t}\right]|\mathbf{x})$ and $\log q_\alpha([\tilde{\mathbf{y}}_t,\ \tilde{\mathbf{y}}_{-t}]\,|\mathbf{x})$ is the same, we can write the equation 6 as follows,

$$\mathcal{SL}(\mathbf{x},\ \mathbf{y}_t^{sb},\ \tilde{\mathbf{y}};\ \alpha) = \max\{0,\ (\mathcal{E}_q(\tilde{\mathbf{x}},\ \left[\mathbf{y}_t^{sb},\ \tilde{\mathbf{y}}_{-t}\right]) - \mathcal{E}_q(\tilde{\mathbf{x}},\ [\tilde{\mathbf{y}}_t,\ \tilde{\mathbf{y}}_{-t}]) + \mathcal{L}(\tilde{\mathbf{y}}_t, \mathbf{y}_t^{sb}))\}. \tag{8}$$

We propose to iteratively calculate the most offending answer $\tilde{\mathbf{y}}$ for the current q_α and then minimize the penalty function of equation 8 according to α. It is worth mentioning that we consider a constraint only over the $\mathbf{y}_t^{sb}$ as we intend to find the best interpreter for the target. As $\mathcal{E}_q$ is a deep neural network and $\tilde{\mathbf{x}}$ is an element-wise multiplication of $\mathbf{x}$ and $\mathcal{IN}_t(\mathbf{x}; \alpha)$, the gradient over the penalty of equation 8 can be back-propagated through the interpreter block.

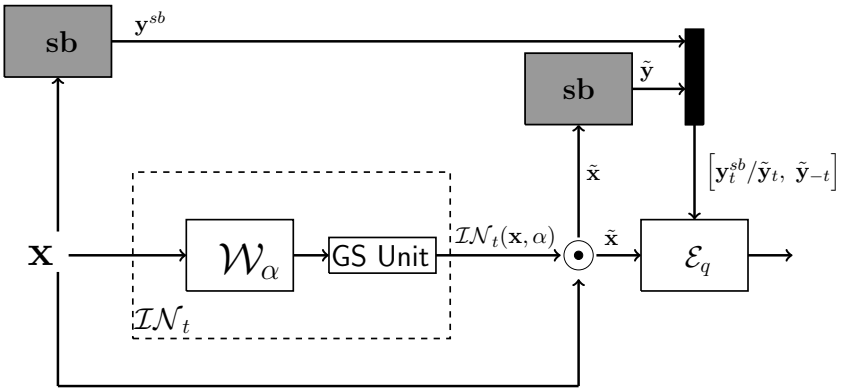

Figure 2: The architecture we use to train the $\mathcal{IN}_t$ for the structured black box determined by **sb**. The interpreter block includes a neural network $\mathcal{W}_\alpha$ and a Gumbel-Softmax (GS) unit. The input feature $\mathbf{x}$ is passed through the $\mathcal{IN}_t$ and a k-hot vector is obtained. The black box prediction is calculated for two input vectors:(1) the feature vector $\mathbf{x}$ and (2) the element-wise multiplication of the $\mathbf{x}$ and $\mathcal{IN}_t(\mathbf{x};\alpha)$. Obtained target outputs $\mathbf{y}_t^{sb}$ and $\tilde{\mathbf{y}}_t$, alongside $\tilde{\mathbf{y}}_{-t}$, are separately passed through energy block $\mathcal{E}_q$.

The final proposed iterative optimization procedure for an interpreter block can be expressed as follows,

$$\alpha^{(k)} = \arg\min_{\alpha} \mathbf{E}_{\mathbf{p}(\mathbf{x})}[\mathcal{SL}(\mathbf{x},\ \mathbf{y}_t^{\mathbf{sb}},\ \tilde{\mathbf{y}}^{(k-1)};\ \alpha)]$$
$$\tilde{\mathbf{y}}^{(k)} = \arg\max_{\mathbf{y}} p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x} \odot \mathcal{IN}_t(\mathbf{x};\alpha^{(k)})) \tag{9}$$

where $\mathbf{y}^{\mathbf{sb}} = \arg\max_{\mathbf{y}} p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x})$. In the first step, the expected penalty of equation 8 is minimized to find the optimum value for the parameter α. The penalty function $\mathcal{SL}$ is minimized when we find a value for α which makes $\left[\mathbf{y}_t^{sb}, \mathbf{y}_{-t}\right]$ the strict maximum of $q_\alpha(.|\mathbf{x})$ with the margin proportional to $\mathcal{L}$. In the second step, the black box prediction for the input which is masked by the output of the current interpreter is calculated.

The initial value $\alpha^{(0)}$ is selected randomly and its associated $\tilde{\mathbf{y}}^{(0)}$ is obtained using the second step of equation 9. The algorithm is continued until the value of the penalty $\mathcal{SL}$ does not considerably change which is usually obtained in less than 100 iterations. Here we consider $\mathcal{L} = I(\mathbf{y}_t^{sb}, \tilde{\mathbf{y}}_t)$ which is an indicator function

3.2 THE INTERPRETER BLOCK

The interpreter $\mathcal{IN}_t$ includes a deep neural network, with a set of parameters α, followed by a Gumbel-Softmax Jang et al. (2017) unit. The detailed architecture of the deep network depends on the inherence of $\mathbf{x}$. The dimension of the interpreter output is the same as the feature vector $\mathbf{x}$. Fig. 2 describes the architecture used for training the interpreter $\mathcal{IN}_t(\mathbf{x},\alpha)$. The output of the deep network $\mathcal{W}_\alpha$ shows the importance of the elements of the feature vector $\mathbf{x}$. To encourage the interpreter to find top k important features associated with the target output $\mathbf{y}_t$, we use the Gumbel-Softmax trick as proposed in Chen et al. (2018). To obtain top k important features, we consider the output of $\mathcal{W}_\alpha(\mathbf{x})$ as parameters of the categorical distribution. Then we can independently draw a sample for k times. Each sample is a one-hot vector in which the element with the value of 1 shows the selected feature. To have a k-hot vector, we can simply get the element-wise maximum of these one-hot vectors. However this sampling process is not differentiable and we use its continuous approximation introduced by the Gumbel-Softmax trick. Considering following random variables,

$$\mathbf{g}_i = -\log(-\log(\mathbf{u}_i)) \tag{10}$$

where $\mathbf{u}_i \sim \text{Uniform}(0,\ 1)$, we can use the re-parameterization trick instead of direct sampling from $\mathcal{W}_\alpha(\mathbf{x})$ as follows:

$$\mathbf{c}_i = \frac{\exp\{\log(\mathcal{W}_\alpha(\mathbf{x})_i + \mathbf{g}_i)/\tau\}}{\Sigma_j \exp\{\log(\mathcal{W}_\alpha(\mathbf{x})_j + \mathbf{g}_j)/\tau\}} \tag{11}$$

The vector $\mathbf{c}$ is the continuous approximation of the sampled one-hot vector. To have k selected features, we draw k vectors $\mathbf{c}^j,\ j = 1 \dots k$ and obtain their element-wise maximum as follows Chen et al. (2018),

$$\mathcal{IN}_t(\mathbf{x}, \alpha)_i = \max_j \{\mathbf{c}_i^j,\ j = 1 \dots k\} \tag{12}$$

Finally, Algorithm 1 of Appendix A summarizes what we propose to train the interpreter $\mathcal{IN}_t$.

4 Experiments

We evaluate the performance of our proposed interpreter on both synthetic and real datasets. In section 4.1, we define two arbitrary energy functions to synthesize structured data. We compare the performance of SOInter with two well-known interpretation techniques, Lime and Kernel-Shap, which are frequently used to evaluate the performance of interpretation methods, and L2X Chen et al. (2018) which proposes an information-Theoretic method for interpretation. None of these techniques are specifically designed for structured models. Indeed, they only consider the target output and ignoring other ones. In section 4.2 and 4.3, the efficiency of SOInter is shown with two real text and image datasets.

4.1 Synthetic Data

Here we define two arbitrary energy functions on input vector $\mathbf{x}$ and output variables $\mathbf{y}$, $\mathcal{E}_1$ and $\mathcal{E}_2$ in equation 13, which are linear and non-linear functions respectively according to the input features.

$$\begin{aligned}
\mathcal{E}_1 &= (\mathbf{x}_1\mathbf{y}_1 + \mathbf{x}_4)(1 - \mathbf{y}_2) + (\mathbf{x}_2(1 - \mathbf{y}_1) + \mathbf{x}_3)\mathbf{y}_2 \\
\mathcal{E}_2 &= (\sin(\mathbf{x}_1)\mathbf{y}_1\mathbf{y}_3 + |\mathbf{x}_4|)\,(1 - \mathbf{y}_2)\mathbf{y}_4 + \left(\exp(\frac{\mathbf{x}_2}{10} - 1)(1 - \mathbf{y}_1)(1 - \mathbf{y}_3) + \mathbf{x}_3\right)\mathbf{y}_2(1 - \mathbf{y}_4)
\end{aligned} \tag{13}$$

Input features are randomly generated using the standard normal distribution. Output variables are considered as binary discrete variables. For each input vector $\mathbf{x}$, we found the corresponding output by the following optimization,

$$\mathbf{y}^* = \arg\min_{\mathbf{y}}\ \mathcal{E}(\mathbf{x},\ \mathbf{y}) \tag{14}$$

where $\mathcal{E}$ shows the energy function from which we attempt to generate data. $\mathcal{E}_1$ describes the energy value over a structured output of size 2 and $\mathcal{E}_2$ describes an output of size 4. For each scenarios, we simulate input vectors with the dimension of 5, 10, 15 and 20.

For each generated dataset, we train a structured prediction energy network introduced in Gygli et al. (2017). As it has the sufficient ability to learn energy functions in equation 13, we can assume it has successfully captured the important features with a negligible error rate. We adopt each interpretation techniques to explain trained energy networks. According to equation 13 first four features affect the value of outputs. Fig. 3 compares the accuracy of results obtained by each method during the interpretation. Diagrams of Fig. 3 show results for target outputs $\mathbf{y}_1$ and $\mathbf{y}_2$ in $\mathcal{E}_1$ and two arbitrary outputs $\mathbf{y}_3$ and $\mathbf{y}_4$ in $\mathcal{E}_2$. As there may be randomness in interpretation methods, we run each interpreter five times for each dataset. Each line in the diagrams shows the average value, and the highlighted area shows the standard deviation. SOInter has an overall better performance compared to others.

As the accuracy measures the exact match of the subset of important features with ground truths, we consider an order for features and report the median rank of important ones in Fig. 4 as proposed in Chen et al. (2018). As the first four features are the solution, the desired median rank is 2.5 in all situations. As shown, SOInter has the nearest median rank to 2.5 nearly in all cases.

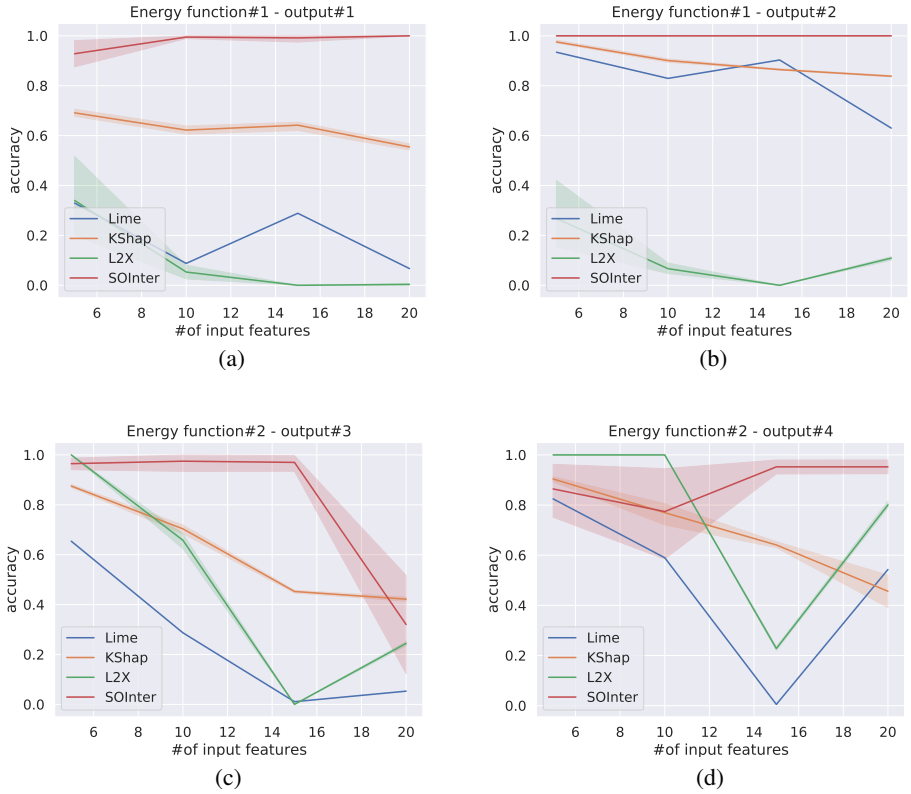

Figure 3: The accuracy of Lime, Kernel-Shap, L2X and SOInter as a function of input size. For each energy function $\mathcal{E}1$ and $\mathcal{E}2$ results on two outputs are reported. The SOInter performance is overall better than others.

As the number of input features is increased, the performance of methods is generally degraded. This is because the ratio of important features compared to the size of the input vector is decreased, which can lead to confusion of the interpreter.

However, the obtained results confirm the robustness of SOInter for the more significant number of input features. Thus the proposed method is more reliable when the size of the input vector is large.

4.2 Multi-label Classification on Bibtex Dataset

Bibtex is a standard dataset for the multi-label classification of texts. Each sample in Bibtex involves an input feature vector corresponding to 1836 words mapped to a 159-dimensional output vector. Elements of the output vector are associated with a set of tags that describes the sample subject. We train a structured prediction energy network (SPEN) as a multi-label classifier on Bibtex with a desirable accuracy as shown in Gygli et al. (2017). A SPEN as a structured black box is a challenging benchmark for an interpreter because of its ability to capture more complicated relations between output variables. During interpreting this classifier with SOInter, we select an output variable, i.e., a tag, as a target of explanation and find the top 30 features related to this tag for each sample. According to SPEN decisions, we aggregate those top features over all samples for each tag and find the top 30 features which are expected to be correlated to this tag. Table 1 shows the general top 30 features for different 3 tags. More results are provided in Table 3 of Appendix A. As shown, word sets are meaningfully correlated to their corresponding tags. In addition, we highlight bold words that are correlated to each tag confirmed by human experts.

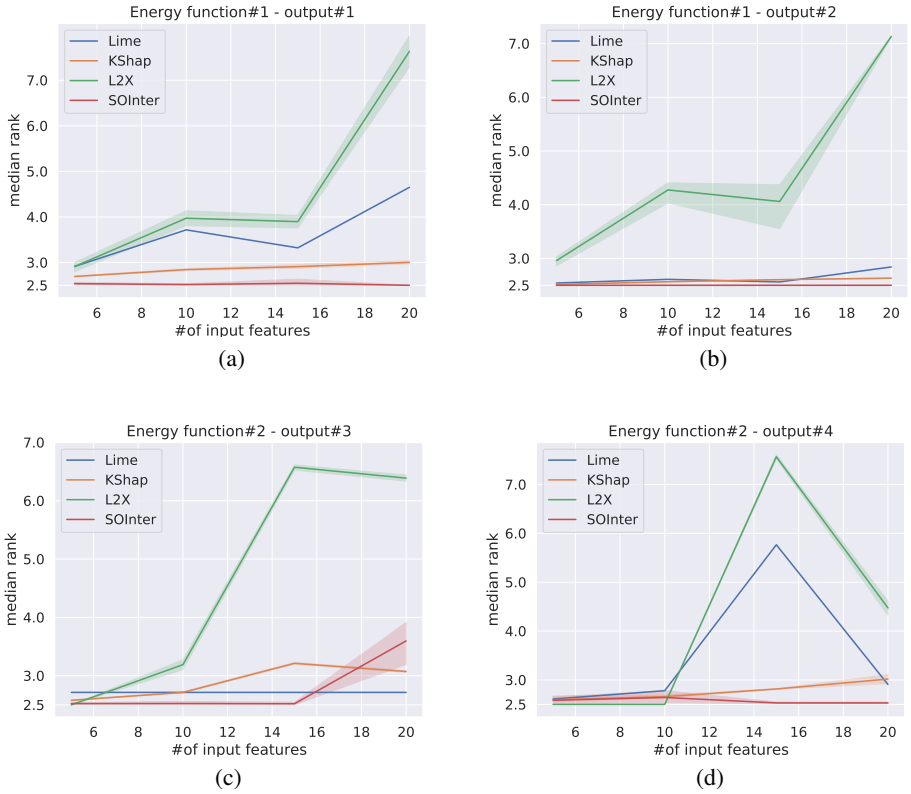

Figure 4: The median rank obtained by Lime, Kernel-Shap, L2X and SOInter as a function of input size. For each energy function $\mathcal{E}1$ and $\mathcal{E}2$ results on two outputs are reported. The ground truth value in all situations is 2.5. The SOInter performance is overall better than others.

Table 1: Results on Bibtex dataset

TAG	TOP 30 IMPORTANT FEATURES ASSOCIATED TO EACH TAG
GAMES	**GAMES-LEARNING-GAME-DESIGN**-HOW-EXPERIENCES-NEW-SOCIAL-IDEA FUTURE-MEDIUM-BEYOND-SCHOOL-LOGIC-CONTEXTS-OPPORTUNITIES **COMPUTERS-COMPUTER**-POINT-KNOW-TEACHERS-EDUCATIONAL-ARGUE BUILDING-DEVELOP-**VIDEO**-EDUCATION-KINDS-NEED-DEMONSTRATES
HCI	**INTERFACES- USER - CASE-** P- **LEARNING** - E - B - CONTENT **SEMANTIC** -ENERGY- CHEMICAL - MOLECULAR - DENSITY 2004 -APPLIED-BETTA-**SIMULATIONS**-FISH-2000-MINIMAL APPLICATIONS-COGNITIVE-SPLENDENS-PHYSICS-CONFERENCE THESE-EDUCATION-APOLIPOPROTEIN-EFFICIENT-OBSERVED
MOLECULAR	**MOLECULAR-BIOINFORMATICS-GENOME-DYNAMICS-STRUCTURES** SMALL-FORCE-VELOCITY-**PARTICLES-SEQUENCE**-EXTERNAL WHILE-**MOLECULES**-FLUID-PARTIAL-**PROTEINS**-THREE-SEMANTIC WEB-ORGANIZED-FORCES-ACID-VERSUS-MODEL-MOTIVATION REDUCING-REVERSE-**BIOLOGICAL**-ALGORITHMS-ADDRESSED

4.3 IMAGE SEGMENTATION ON WEIZMANN-HORSE DATASET

Image segmentation is another structured output learning task in which the image is partitioned into semantic regions. Here we again train a SPEN for segmentation of 24×24 Weizmann-horse dataset. Each image of this dataset is partitioned into two regions which determines the borders of the horse. During the interpretation, we consider a pixel as the target and find pixels of the image that affect the target's output.

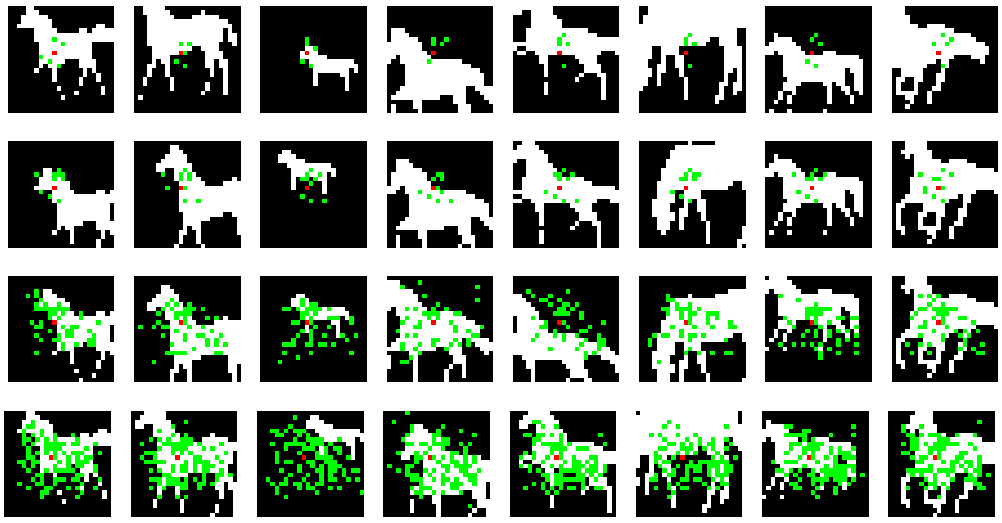

Figure 5: The red pixel shows the target and green ones are important features selected by SOInter. The number of important features is considered 5, 10, 50 and 100 respectively for images of each row. Important features are mostly placed in the locality of the target.

In Fig. 5 the pixel in $[10,\ 10]$ is considered as target and the interpretation results for arbitrary images are shown. We do experiments for different numbers of important features of 5, 10, 50, and 100. The red pixel shows the target, and green ones are obtained important input features as expected green pixels are placed in the locality of the target.

Table 2 compares the accuracy and post-accuracy of the SPEN in finding the label of the target during the segmentation process. Post-accuracy measures the model performance when only the selected features are passed through the SPEN and un-selected ones are masked with zero. These measures are shown for different choices for the number of important features. The negligible difference between the accuracy and post-accuracy confirms that the interpreter selects pixels which are the most effective ones in labeling the target. In addition, the increased value of the post-accuracy in some situations shows that omitted features confuse the segmentation model, i.e. SPEN, as a noise signal. These observations show that SOInter successfully selects important features which affect the SPEN decision for the target output.

Table 2: Accuracy and post accuracy for target pixel segmentation

Num. of imp. features	**5**	**10**	**50**	**100**
ACCURACY	0.877	0.872	0.868	0.866
POST-ACCURACY	0.872	0.855	0.868	0.872

5 CONCLUSION

We have presented SOInter, an interpreter for explaining structured output models. We focused on a single output variable of a structured model, available as a black box, as the target. Then we train a function over the input space, which returns a subset of important features for the black box to decide on the target. This is the first time an interpreter has been designed explicitly for structured output models to the best of our knowledge. These models learn complex relations between output variables which ignoring them while interpreting a single output can decline the explanation performance. We used an energy model to learn the structural information of the black box and utilize it during the interpreter's training. The effectiveness of SO-Inter is confirmed using synthetic and real structured datasets.

REFERENCES

S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Mller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015.

D. Belanger and A. McCallum. Structured prediction energy networks. In *Proceedings of the 33th International Conference on Machine Learning (ICML 2016)*, pp. 983–992, 2016.

D. Belanger, B. Yang, and A. McCallum. End-to-end learning for structured prediction energy networks. *arXiv preprint arXiv:1703.05667*, 2017.

J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, pp. ?, 2018.

Liang-Chieh Chen, Alexander Schwing, Alan Yuille, and Raquel Urtasun. Learning deep structured models. In *International Conference on Machine Learning*, pp. 1785–1794. PMLR, 2015.

Colin Graber, Ofer Meshi, and Alexander Schwing. Deep structured prediction with nonlinear output transformations. *arXiv preprint arXiv:1811.00539*, 2018.

M. Gygli, M. Norouzi, and A. Angelova. Deep value networks learn to evaluate and iteratively refine structured outputs. *arXiv preprint arXiv:1703.04363*, 2017.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *stat*, pp. 1050:1, 2017.

S. M. Lundberg and S. I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pp. 4765–4774, 2017.

W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

Jian Peng, Liefeng Bo, and Jinbo Xu. Conditional neural fields. *Advances in neural information processing systems*, 22:1419–1427, 2009.

M. T. Ribeiro, S. Singh, and C. 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*, pp. 11351144, 2016.

Alexander G Schwing and Raquel Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3145–3153, 2017.

K. Simonyan, V. Andrea, and Z. Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *arXiv preprint arXiv:1311.2901*, 2013.

J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, pp. 12:9314, 2015.

M. L. Zintgraf, S. T. Cohen, and T. Adel. Visualizing deep neural network decisions: Prediction difference analysis. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017.

A APPENDIX

Algorithm 1 The training procedure of the interpreter $\mathcal{IN}_t(\mathbf{x}; \alpha)$

function INTERPRETER-BLOCK($\mathbf{x},\ \alpha$)
  return Gumbel-Softmax ($\mathcal{W}_\alpha(\mathbf{x})$)
end function

function SL-CALCULATOR($\mathbf{x},\ \tilde{\mathbf{y}},\ \alpha$)
  $\tilde{\mathbf{x}} \leftarrow \mathbf{x} \odot$ Interpreter-Block($\mathbf{x},\ \alpha$)
  $\mathbf{y}^{sb} \leftarrow \arg\max_{\mathbf{y}} p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x})$
  $\mathcal{L} \leftarrow |\tilde{\mathbf{y}}_t - \mathbf{y}_t^{sb}|$
  $\mathcal{SL} \leftarrow \max\{0,\ \mathcal{E}_q(\tilde{\mathbf{x}},\ [\mathbf{y}_t^{sb},\ \tilde{\mathbf{y}}_{-t}]) - \mathcal{E}_q(\tilde{\mathbf{x}},\ [\tilde{\mathbf{y}}_t,\ \tilde{\mathbf{y}}_{-t}]) + \mathcal{L}\}$
  return $\mathcal{SL}$
end function

function TRAINING-INTERPRETER(t)
  $\alpha^{(0)} \leftarrow$ Sample Standard Normal distribution
  $\tilde{\mathbf{y}}_i{}^{(0)} \leftarrow \arg\max_{\mathbf{y}} p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x} \odot$ Interpreter-Block($\mathbf{x}_i,\ \alpha^{(0)}$), for $i = 1...N$
  $k = 1$
  while the penalty does not change considerably **do**
    $\alpha^{(k)} =$ Adam-minimizer$_\alpha$ Σ_i^NSL-calculator($\mathbf{x}_i,\ \tilde{\mathbf{y}}_i{}^{(k-1)},\ \alpha$)
    $\tilde{\mathbf{y}}_i{}^{(k)} =$ Adam-minimizer$_{\mathbf{y}}$ $- p_{\mathbf{sb}}(\mathbf{y}|\mathbf{x}_i \odot$ Interpreter-Block($\mathbf{x}_i,\ \alpha^{(k)}$)), for $i = 1...N$
    $k = k + 1$
  end while
end function

Table 3: Results on Bibtex dataset

TAG	TOP 30 IMPORTANT FEATURES ASSOCIATED TO EACH TAG
CLUSTERING	**DATA**-**CLUSTERING**-E-MORE-THAT-WERE-TYPE-5-ANNUAL-REAL **CLUSTER**-APPLICATIONS-**OPTIMIZED**-FUNCTIONAL-SAME-RETRIEVAL DESIGN-AUTOMATIC-PROCEEDINGS-PROBLEM-GIVEN-QUERY-THESE SELECTION-ALSO-CHEMISTRY-HAS-EFFECTIVE-BOUND-INFORMATION
DEFFUSION	TERM-**WEIGHT**-**DIFFUSION**-MOBILITY-CONFERENCE-OFTEN DECREASE-IMAGING-SCIENCE-INCREASED-DISCOVERY-SOLUTION QUALITY-E-ENZYME-**FUNCTION**-PROPOSE-WHICH-POSSIBLE DEVELOPERS-SUBJECTS-MUCH-AVAILABLE-APPROACH-PAPER YEARS-DETAIL-HETEROGENEOUS-IDEAS-ENGINEERING
ELECTROCHEMISTRY	REVIEW-**QUANTUM**-NETWORKS-PROPOSE-IT-WORKSHOP-INTERNATIONAL WE-DISCUSS-COMPUTER-FIRST-COLLABORATIVE-WEB-**MECHANICAL** CONTEXT-**ELECTROCHEMICAL**-DO-FOUND-**APOLIPOPROTEIN** **LIPOPROTEIN**-**ELECTRODE**-APPROACH-**PHYSICS**-AMPEROMETRIC-HIGH STATISTICAL-LANGUAGE-APPLICATIONS-CONCEPTUAL-IMMUNOASSAY
GRAPH	**OBSERVATIONS**-SERUM-ACIDS-OBJECTS-PAST-UNIT PARADIGM-**NODES**-**FREQUENCIES**-PERFORMED-**GRAPHS**-**SOCIAL** WEAK-MEASUREMENTS-PROCEDURES-ANTI-ANTIBODY-FACT EASY-AT-LITERATURE-RELATION-PATHWAY-PARAMETER ADAPTATION-CREATING-UNIVERSAL-DISCOVERY-FAMILY-COST
ONTOLOGY	**ONTOLOGY**-**LANGUAGES**-**IUPAP**-XXIII-TOP-**INTEGRATE**-KNOWN KEY-STATISTICAL-GIVEN-BOOK-**PHYSICS**-CONFERENCE DISCUSSED-METHODS-SPLENDENS-OBSERVED-C-SHOW-DETERMINE PROPERTIES-**MECHANISMS**-EVALUATING-MONITORING INTERNATIONAL-SOFTWARE-IMAGES-CONVENTIONAL-FOUND-FISH