Appendix

A Supplement for Section 1 (Introduction)

Here, we present details of the user study in Fig. 1b. The figure shows that logic rule explanations achieve higher human precision than linear-regression-based explanations with local stability, while providing a confidence score that correlates with human precision.

We used a vendor company to recruit three native English speakers for the user study (Sec. 3.3). User studies can be performed by 1) hiring a large number of labelers from platforms like Prolific and AMT or 2) hiring a limited number of experienced annotators from a labeling company. While platforms like Prolific make it easy to find many labelers, they are known to be better suited for cognitively simple tasks and may suffer from errors [54]. Our task is challenging for ordinary labelers, as we require them to carefully reason about which features of an adult are useful for predicting his or her income (the Adult dataset) and compare multiple similar explanations. Thus, we validated the model with more experienced annotators hired through a labeling company. To ensure the labelers have an adequate understanding of the task, we provided them with detailed guidelines and examined their initial labels with feedback when a misunderstanding is detected. Such a close interaction would not be possible in crowdsourcing platforms, which may lead to errors and unreliable results.

Each participant was provided 1,000 and 500 randomly selected explanations from SELOR and SENN, respectively. For each explanation, we test whether it can naturally lead to the model prediction according to human perception. Participants were asked to provide 1) the class label for the explanation and 2) how confident they were in their decision by using a 5-point likert scale (HC, i.e., human confidence). For example, given an explanation "*awesome*, *tasty*", the participant will give the label *positive sentiment* and a high confidence score "5" out of 5. When labels were the same to model predictions, human precision was high. We sampled explanations so that their confidence score from models (MC, i.e., model confidence) was evenly distributed and examined how explanation quality varies with the confidence score. Fig. 1b shows how human precision changes with different levels of model confidence. As shown in the figure, logic rule explanations achieve higher human precision than the linear-regression-based explanations, and the model confidence shows a strong correlation with human precision. Here, human precision is the F1-score of machine prediction for give logic rules using human prediction as the ground-truth labels. Table 6 provides more detailed information about our user study. The logic rule with a higher MC level tends to have higher agreement and HC. Also, the logic rule shows better human precision at most MC levels.

User instruction and labeling detail. We describe the instructions given to participants in the attached guideline file (*Labeling_Guidelines_User_Study_Figure1b.pdf*) with detailed description of the task and labeling examples. Participants received an Excel file containing blank labels, which they were instructed to fill out and return. The snapshot of the Excel file is also attached as a separate file (*Screenshot_User_Study_Figure1b.PNG*). Each participant was paid 22.5\$ per hour and the total budget we spent was 937.5\$ for this task.

Table 6: User study results for human precision on logic rule- and linear-regression-based explanations. HC denotes the average human confidence while MC denotes the machine confidence. Avg. denotes the average number of sentiment agreement, human confidence, and human precision of all data points. (Lv 1: $0.0 \sim 0.2$, Lv 2: $0.2 \sim 0.4$, Lv 3: $0.4 \sim 0.6$, Lv 4: $0.6 \sim 0.8$, Lv 5: $0.8 \sim 1.0$)

(a) Logic rule

	Sentiment Agreement	Avg HC	Human Precision
MC Lv 1	82.67	2.72	52.65
MC Lv 2	86.00	2.88	53.53
MC Lv 3	84.00	3.31	76.19
MC Lv 4	92.67	3.85	89.38
MC Lv 5	95.00	4.07	90.41
Avg.	88.07	3.36	73.32

(b) Linear-regression-based explanation

	Sentiment Agreement	Avg HC	Human Precision
MC Lv 1	78.79	3.63	47.71
MC Lv 2	81.56	3.55	48.69
MC Lv 3	75.95	3.62	52.83
MC Lv 4	79.12	3.56	56.82
MC Lv 5	84.51	3.78	66.17
Avg.	79.96	3.63	54.46

B Supplement for Section 2 (Deep Logic Rule Reasoning)

B.1 Symbols

Table 7 summarizes the symbols used in this paper.

Table 7: The meaning and detailed explanation of each symbol used in the paper.

	Meaning	Detailed Explanation
$\mathbf{x}$	Input sample	Any type of data (e.g. text, tabular)
$\boldsymbol{\alpha}$	Antecedent	Condition to apply the rule
b	Human belief	Common sense that a human believes when they make a decision
y	Consequent	Model's prediction output for the given antecedent
o_i	Atom or logical connective	Atom is the smallest unit of explanation
		Logical connective combines atoms
$\mathbf{o}_i$	Embedding of o_i	Initialized as the average embedding of all training samples that satisfy the atom
$\mathcal{O}$	Set of atoms	
$\mathcal{C}_i$	Set of candidates for o_i	Every candidate should satisfy both global and local constraints (Sec. 2.5)
L	Length of an antecedent	Number of atoms and logical connectives included in an antecedent
N	Number of training data	
$n_{\boldsymbol{\alpha}}$	See detailed explanation	Number of data samples in training data that satisfies the antecedent $\boldsymbol{\alpha}$
$n_{\boldsymbol{\alpha},y}$	See detailed explanation	Number of data samples in training data that satisfies the antecedent $\boldsymbol{\alpha}$ and has the consequent y
$\mathcal{Y}_{\boldsymbol{\alpha}}$	See detailed explanation	Data samples of class y in training data that satisfies the antecedent $\boldsymbol{\alpha}$
$\boldsymbol{\alpha}^{(s)}$	s-th sample of $\boldsymbol{\alpha}$	s-th sampled antecedent in deep antecedent generation
S	Total number of $\boldsymbol{\alpha}^{(s)}$	Set as S = 1 by default
$\Omega(\boldsymbol{\alpha})$	Required number of $\boldsymbol{\alpha}$	The number of explanation required to explain given input
$\mathbf{h}_i$	Hidden state of encoder	The encoder can be any neural sequence encoder such as GRU or Transformer
C	See detailed explanation	Time complexity for computing the consequent of each antecedent
A	See detailed explanation	Number of all feasible antecedents. Usually exponentially increase with $\lvert\mathcal{O}\rvert$ and L (i.e. $\lvert\mathcal{O}\rvert^L$)
A'	See detailed explanation	Number of sampled antecedents for training of neural consequent estimator

B.2 Extension to Regression Tasks

Although we mainly focused on classification tasks, SELOR can be applied to regression tasks after a small modification. For regression tasks, we change the modeling of neural consequent estimation from a categorical to a direct prediction. Our neural consequent estimator for regression predicts the value y' instead of $p(y|\boldsymbol{\alpha})$ and coverage $c_{\boldsymbol{\alpha}}$. Then, we maximize $||y' - y||_2$.

B.3 Probability Decomposition

Here we give the proof for Eq. (1):

$$
\begin{aligned}
p(y|\mathbf{x}, b) &= \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\mathbf{x}, b) \\
&= \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) \frac{p(\boldsymbol{\alpha}, \mathbf{x}, b)}{p(\mathbf{x}, b)} \\
&= \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) \cdot \frac{p(\boldsymbol{\alpha}, \mathbf{x}, b)}{p(\mathbf{x}, b)} \cdot \frac{p(b|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})}{p(\boldsymbol{\alpha}, b)} \cdot \frac{p(\boldsymbol{\alpha}|\mathbf{x}) p(\mathbf{x})}{p(\boldsymbol{\alpha}, \mathbf{x})} && (9) \\
&= \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) \cdot p(b|\boldsymbol{\alpha}) \cdot p(\boldsymbol{\alpha}|\mathbf{x}) \cdot \frac{1}{p(b)} \cdot \frac{p(\mathbf{x}) p(b)}{p(\mathbf{x}, b)} \cdot \frac{p(\boldsymbol{\alpha}) p(\boldsymbol{\alpha}, \mathbf{x}, b)}{p(\boldsymbol{\alpha}, \mathbf{x}) p(\boldsymbol{\alpha}, b)} && (10) \\
&= \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) \cdot p(b|\boldsymbol{\alpha}) \cdot p(\boldsymbol{\alpha}|\mathbf{x}) \cdot \frac{1}{p(b)} \cdot \frac{p(\mathbf{x}) p(b)}{p(\mathbf{x}, b)} \cdot \frac{p(\mathbf{x}, b|\boldsymbol{\alpha})}{p(\mathbf{x}|\boldsymbol{\alpha}) p(b|\boldsymbol{\alpha})} && (11) \\
&\propto \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) \cdot p(b|\boldsymbol{\alpha}) \cdot p(\boldsymbol{\alpha}|\mathbf{x}) && (12)
\end{aligned}
$$

There are two assumptions to hold Eq. (12).

Assumption A. For $p(y|\mathbf{x}, b) = \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\mathbf{x}, b)$, we assume that $p(y|\boldsymbol{\alpha}) = p(y|\boldsymbol{\alpha}, \mathbf{x}, b)$. This is decomposed into two assumptions: $p(y|\boldsymbol{\alpha}) = p(y|\boldsymbol{\alpha}, \mathbf{x})$ (A1) and $p(y|\boldsymbol{\alpha}, \mathbf{x}) = p(y|\boldsymbol{\alpha}, \mathbf{x}, b)$ (A2).

Assumption A1 $p(y|\boldsymbol{\alpha}) = p(y|\boldsymbol{\alpha}, \mathbf{x})$ indicates that explanation $\boldsymbol{\alpha}$ contains all information in input $\mathbf{x}$ that is needed to predict y. This formulation compels the model to pass information from $\mathbf{x}$ to y only via explanations, as opposed to other unexplainable parts. This assumption may limit the prediction performance, but it is essential for $\boldsymbol{\alpha}$ to be a trustable explanation for predicting y. Otherwise, there may be a direct connection between y and $\mathbf{x}$ that is unrelated to the explanation $\boldsymbol{\alpha}$. Thus, $\boldsymbol{\alpha}$ may only explain a small portion of the model behavior (e.g., only explain 1% of the change in y) and differ substantially from the ground-truth explanation of the model behavior.

Assumption A2 $p(y|\boldsymbol{\alpha}, \mathbf{x}) = p(y|\boldsymbol{\alpha}, \mathbf{x}, b)$ means that explanation $\boldsymbol{\alpha}$ and input $\mathbf{x}$ contain all of the information in b (human prior preference for explanations) that is needed to predict y. It is intuitive that this assumption holds, as human preference for explanations is unrelated to the current class label.

Assumption B. For $\sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\mathbf{x}, b) \propto \sum_{\boldsymbol{\alpha}} p(b|\boldsymbol{\alpha}) p(y|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\mathbf{x})$, we assume that $p(\mathbf{x}, b) = p(b)p(\mathbf{x})$ and $p(\mathbf{x}, b|\boldsymbol{\alpha}) = p(b|\boldsymbol{\alpha})p(\mathbf{x}|\boldsymbol{\alpha})$.

It means that $\mathbf{x}$ and b are independent no matter which $\boldsymbol{\alpha}$ is given. In other words, seeing input sample ($\mathbf{x}$) does not change the belief in our prior preference for explanations (b), no matter which explanations ($\boldsymbol{\alpha}$) are given, i.e., $p(b) = p(b|\mathbf{x})$ and $p(b|\boldsymbol{\alpha}) = p(b|\mathbf{x}, \boldsymbol{\alpha})$. The rationale for this assumption is that human preferences for explanations are usually fixed and unrelated with the input $\mathbf{x}$. Even if this assumption is not satisfied, it will not have a significant effect on the framework. Only the human prior module must be integrated into the antecedent generation module, which changes from $p(\boldsymbol{\alpha}|\mathbf{x})$ to $p(\boldsymbol{\alpha}|\mathbf{x}, b)$.

B.4 Neural Consequent Estimation

The input of the neural consequent estimator is the antecedent embedding, which is obtained by $(\mathbf{o}_1 ..., \mathbf{o}_L)$, where $\mathbf{o}_i$ is the embedding of o_i in $\boldsymbol{\alpha} = (o_1 ..., o_L)$. For each atom, $\mathbf{o}_i$ is initialized as the average embedding of all training samples that satisfy the atom, where the sample embedding can be derived using a pretrained model or f. The embeddings of logical connectives are initialized at random, and can be omitted when there is only one logical connective (e.g., AND). We use the Transformer encoder [43] as the backbone neural network to emphasize the contextual interaction between atoms and logical connectives.

After encoding $\boldsymbol{\alpha}$ with Transformer, an MLP (Multi-Layer Perceptron) layer reduces the representation obtained by mean pooling to a logit. Softmax (multi-class) or sigmoid (two classes) is used to activate the logits to determine the probability for each class $p(y|\boldsymbol{\alpha})$ and the coverage $c_{\boldsymbol{\alpha}}$ of the antecedent $\boldsymbol{\alpha}$, which is converted to the number of observations in the training dataset with $n_{\boldsymbol{\alpha}} = c_{\boldsymbol{\alpha}} N$.

The time complexity of deep logic reasoning is significantly reduced by neural estimation of the consequence (Sec. 2.6).

The neural consequent estimator is pretrained with $A' = 10,000$ sampled rules for each antecedent length (Total $L \times A'$), then used to train the deep antecedent generator with frozen parameters. The following steps are taken to ensure the generality of the rules used in pretraining. To begin, we create the "true matrix" (*tm*), that has the size $(|\mathcal{O}| \times N)$, which indicates whether each input sample satisfies each atoms. Then, by multiplying *tm* and its transpose, we can create a matrix of size $(|\mathcal{O}| \times |\mathcal{O}|)$ that indicates the number of samples that satisfy 2-length antecedents $([o_i, o_j], i, j \in \mathcal{O})$.

Then, we obtain the list of 2-length antecedents whose frequency is larger than a threshold (i.e., *min_df*). From the 2-length antecedent list, we sample $k \times A'$ rules while k is a hyper-parameter larger than 1. We set k to be the same with *min_df* in the experiment. With these $k \times A'$ rules, we can make a new true matrix of size $((k \times A') \times N)$ and repeat the steps to obtain the rules whose frequency is larger than *min_df*. This sampling process takes linear time to A' instead of A, which reduces the time complexity. After the whole process, we can obtain $k \times A'$ number of antecedents for each length. Then we randomly choose A' rules for pretraining of consequent estimator maintaining the balance of labels. In practice, the time spent in sampling process was 1144s for Yelp, 402s for Clickbait, and 456s for Adult dataset in our setting. This time can be even reduced with larger *min_df*.

B.5 Differentiable Learning

In Sec. 2.6, we sampled one antecedent from $p(\boldsymbol{\alpha}^{(s)}|\mathbf{x})$. Naive selection (e.g., selecting the maximum value's index) stops the gradient and prevents differential learning of the neural model. This problem is solved by sampling $\boldsymbol{\alpha}$ with the Straight-Through Gumbel-Softmax function, as shown in Eq. (7). For forward propagation, $\boldsymbol{\alpha}^{(s)} = (\alpha_1^{(s)}..., \alpha_L^{(s)})$ is represented by L discrete one-hot vectors. To derive L input embeddings for the neural consequent estimator in Sec. 2.5, each one-hot vector is multiplied by an embedding matrix of atoms and logical connectives. Differentiable Gumbel-Softmax distribution is used to approximate the gradients during backpropagation.

B.6 Theoretical Analysis of Explanation Stability

For linear-regression-based models like SENN [11], the explanations for similar inputs may be entirely different without specific constraints like the robustness loss, because the main optimization goal for SENN is the local prediction accuracy. Without the robustness loss, the model may find a correct prediction locally for a single instance, but being "surely no more interpretable than any deep neural network" (quoted from the SENN paper). However, this is not the case for the logic rule reasoning framework, because the antecedent generator is trained to optimize two globally consistent rewards (Eq. 3 and Sec. 2.6): human's prior belief about which explanation types are good and the explanation confidence that is measured by the global prediction accuracy over the entire training dataset given the explanation (logic rule). Thus, explanations for similar inputs may be different only when:

1. The optimal (most confident and human-preferred) rules for the inputs are different.
2. There are multiple explanations that achieve the exact same reward.
3. The model has not been trained sufficiently to achieve the optimal result.

In situation 1), SELOR removes the heuristic constraint regarding the similarity of explanations, allowing us to identify the optimal explanations for the two inputs. If an instance A is changed to the instance B by substituting "*very disappointing*" with "*disappointing*", then the best explanation may change from "*very disappointing*" in the instance A to "*awful*" in the instance B. Even if the two instances are similar, their optimal explanations may differ. This is plausible as such a change increases the explanation's confidence. In other words, the radius of validity of an explanation corresponds to inputs that have similar optimal rules. For example, explanation "*very disappointing*"⇒*negative sentiment* can generalize to all instances that satisfy the rule and at the same time do not satisfy the more confident rule. When we want to force the explanations of two inputs to be similar, we can also incorporate a constraint that mimics the robustness loss in SENN into the soft human prior. Situation 2) rarely occurs, as our explanation confidence reward is a real number, not a discrete value. In rare cases where this occurs, it is possible to remedy the situation by using the soft human prior. Situation 3) can be avoided by checking the training loss, the classification accuracy, and the explainability.

C Supplement for Section 3 (Experiment)

C.1 Datasets

We use the following three datasets for experiments. Table 8 reports the number of data points for each dataset that we used for training, validation, and testing. **Yelp** classifies reviews of local businesses into positive or negative sentiment [44]. For Yelp, we use a down-sampled subset (10%) for training, as per existing work [39]. We split the test dataset and used half of them for the validation dataset. **Clickbait News Detection** from Kaggle labels whether a news article is a clickbait [45], and we use the "*news*" and "*clickbait*" classes in the dataset. We split the train data into train and validation. **Adult** from the UCI machine learning repository [46] is an imbalanced tabular dataset that provides labels about whether the annual income of an adult is more than \$50K/yr or not. We split the data points into train, validation, and test datasets.

C.2 Implementation Details

Hyperparameter settings. The backbone models for textual data (i.e., BERT, RoBERTa) follow the original setting, and the model for tabular data (i.e., DNN) consists of network with three fully-connected layers with ReLU activation layers (i.e., FC-ReLU-FC-ReLU-FC) with 512 hidden dimensions. We employ GRU [42] as a sequential encoder for deep antecedent generation, and Transformer [43] as a neural model for consequent estimation, respectively. For deep antecedent generation, neural consequent estimation, and other baseline models, we set the hidden dimension $|h|$ as the default BERT and RoBERTa embedding size (i.e., 768) for textual data and 512 for tabular data. For training of SELOR, cross-entropy loss is used for optimization on the probability predicted by the consequent estimator for the antecedents extracted by the antecedent generator. For RCN, we extract a predefined rule set by following the original work [39]. In particular, the predefined rules are decision paths in random forests with 100 estimators and a maximum depth of four. After excluding stopwords, we limit atoms in textual data to only derive from the top-5000 most frequent words. Tabular data uses both categorical and numerical features for atoms while the threshold of numerical features is set to the 25th, 50th, 75th percentiles of data. The length of antecedent L (i.e., the number of atoms from recursive deep antecedent generation) is set to 4. The minimum document frequency is set to 200, and the number of rules for pretraining the neural consequent estimator is set to $10,000$.

We introduce hyper-parameters in training our model and baselines. Note that the same hyper-parameters are used for training baselines, the neural consequent estimator, and the deep antecedent generator for the all datasets. The base backbone network and self-explainable models are trained 10 epochs. The batch size is set to 16, the largest size that can be trained on our GPU. For optimization, we employ Adam optimizer with a learning rate of $1e-5$, and ExponentialLR scheduler with γ 0.95. For the learning rate, the one with the best performance is selected after experiments on $5e-5$, $4e-5$, $3e-5$, $2e-5$, and $1e-5$. For SENN, a set of token embeddings from the pretrained language model (i.e., BERT and RoBERTa) are utilized as inputs and are considered to be interpretable basic concepts for textual data experiments. In the case of tabular data, raw input features are used. We follow the implementation and hyper-parameter settings for training as in the original work [11]. Optimizer or scheduler are also set to be the same as other baselines for a fair comparison. One NVIDIA A100 is used for each experiment.

Details about selecting atom candidates. We ensure that atoms have a consistent form with baselines for fair comparison. In current implementation, we only consider atoms that contains the information about existence of a word for given instance (e.g., "*awesome* ≥ 1") for textual datasets. This enables a comparison with explainable models that highlights the words based on their importance weight. We choose top 5000 frequent words in vocabulary set for atom candidates in main experiments in Sec. 3. The result with other number of atoms is shown in Sec. C.5.4. The result with For tabular

Table 8: Dataset statistics. The labeling ratio shows whether the data is imbalanced between classes. All data, including training, validation, and test data, is split into the same ratio.

Dataset	# for training	# for validation	# for test	Prediction Labels	Label Ratio
Yelp	56000	19000	19000	Negative, Positive	1 : 1
Clickbait	18330	1312	1312	News, Clickbait	3.9 : 1
Adult	39073	4884	4885	<=50K, >50K	3.2 : 1

Table 9: Comparison of classification performance measured in F1. The average results from five runs are shown. The best results among self-explaining models are marked in **bold**, and the highlighted cells indicate a similar or better result compared with the unexplainable backbone (Base). The numbers in subscript indicates the standard error of the result.

	Yelp		Clickbait		Adult	Average
	BERT	RoBERTa	BERT	RoBERTa	DNN	
Base	96.20 $_{0.0541}$	97.16 $_{0.0672}$	72.84 $_{0.9302}$	74.25 $_{0.7763}$	76.15 $_{0.2522}$	83.32
SENN	95.12 $_{0.1995}$	96.07 $_{0.1180}$	69.09 $_{0.9550}$	70.99 $_{0.5076}$	71.69 $_{0.7681}$	80.59
RCN	**96.38** $_{0.0089}$	**97.36** $_{0.0049}$	68.80 $_{0.1359}$	68.64 $_{0.1467}$	**77.35** $_{0.0309}$	81.77
SELOR	**96.26** $_{0.0445}$	**97.13** $_{0.0642}$	**71.12** $_{0.5479}$	**74.20** $_{0.5009}$	**77.37** $_{0.0541}$	**83.34**

datasets, we choose different strategies based on feature types. For categorical features, whether the instance belongs to a certain category or not becomes an atom. For example, in the Adult dataset, "*marital-status* == *Married*" indicates the person in the given instance is married. For numerical features, we calculate 25th, 50th, 75th percentiles of each feature distribution for the threshold. We use whether the feature of a given sample is larger or smaller than the threshold as atoms to obtain thresholds and use those values to determine the over or under presence of each feature in the given sample. For example, the feature "*age*" of the Adult dataset has thresholds 28, 37, and 48, which lead to atoms like "*age* $\geq$ 28", "*age* < 28", "*age* $\geq$ 37", "*age* < 37", "*age* $\geq$ 48", and "*age* < 48". This form is consistent with the atoms in our baseline RCN [39], which uses random forests for rule creation.

It is possible that different atoms associated with the same feature appear in the same explanation, for example, as in our tabular dataset (e.g., "*age* $\geq$ 37" and "*age* $\geq$ 48" for the feature "*age*"). In such a situation, we remove the redundant atoms after the explanation has been generated (e.g., removing "*age* $\geq$ 37"). Note that the generated atoms will not be conflicted with each other. For example, "*age* $\geq$ 48" and "*age* < 37" will not be generated simultaneously in one explanation, because the condition for generation is that the corresponding instance satisfies both atoms. This is enforced by the local constraint introduced in Sec. 2.5. We find that such a post-processing step of removing redundant atoms is easy to implement and has reasonably good explainability and prediction performance. It is also possible to eliminate redundant atoms during explanation generation. One possible way is to create the atoms so that they do not overlap (e.g., creating "*age* $\geq$ 48", "48 > *age* $\geq$ 37", "37 > *age* $\geq$ 28", "28 > *age*" for feature "*age*"). However, this may make it impossible to flexibly combine different thresholds (e.g., generating "48 > *age* $\geq$ 28"). Another way is to apply a mask to the model so that it assigns zero probability to an already chosen feature or a redundant atom. This can be implemented by carefully setting the local constraint in Sec. 2.5.

C.3 Prediction Performance in F1-score

We also provide the prediction performance of SELOR and other self-explainable baselines in Table 9. The result shows that our method successfully maintains the representation ability of deep learning.

C.4 User Study Details

Explanation generation process. Here, we introduce how we generate the explanations.

- **LIME** is distributed as a Python package, and we use *lime_text* and *lime_tabular* to generate explanations. The number of disturbances is set to 3,000 for textual data. We choose the words that are consistent with the prediction having positive weights as explanation. To reduce the incongruity with other explanations, we hide the score provided by LIME and join chosen the predicates.
- **Anchor** is initialized with an empty set. For every iteration, multiple candidate anchors are produced by extending the current anchor by one additional predicate. Then, the model selects the set of predicates with the highest precision as an anchor while perturbing the other predicates. This process repeats until it satisfies the precision constraint of probability 0.95.
- **SENN** defines the interpretable basis concepts $h(x)$ from the input $\mathbf{x}$, and learns the relevance value $\theta(x)$ which is an interpretable weight in relation to each concept (i.e., $f(x) = \sum_i \theta(x)_i \cdot h(x)_i$). We choose the set of top-k predicates with the highest positive relevance value as an interpretation for the given input $\mathbf{x}$. k is set to 5. We remove meaningless words (such as "-", " ") by post-

Table 10: User study results on human precision. Nine participants P1-P9 were asked to annotate whether an explanation is a reasonable rationale for the prediction. For each compared method, we report the percentage of explanations that are considered good (a, b) or best (c, d). Avg. and Agr. denote the average and inter-participant agreement, respectively. P-values from t-test indicates the statistical significance of the experiment. We mark one star (*) if the p-value is lower than 0.05. Best results are highlighted in **bold**.

(a) Percentage of good (Yelp)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	Avg.	Agr.	P-value
Lime	88	82	**96**	90	92	90	98	88	84	89.8	84.4	8.68 E-04*
Anchor	86	74	92	86	84	84	90	78	86	84.4	87.7	1.12 E-07*
SENN	26	22	18	32	26	30	80	32	44	34.4	72.3	1.40 E-51*
RCN	70	32	6	70	62	74	88	76	**98**	64.0	77.6	7.26 E-13*
SELOR	**90**	**84**	**96**	**98**	**100**	**96**	**100**	**92**	94	**94.4**	93.9	-

(b) Percentage of good (Adult)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	Avg.	Agr.	P-value
Lime	**88**	24	88	26	76	2	6	26	48	42.7	57.1	6.09 E-54*
Anchor	30	38	32	54	30	84	68	94	44	52.7	59.9	5.56 E-18*
SENN	**88**	16	**90**	30	82	4	8	38	58	46.0	51.5	1.18 E-41*
RCN	78	70	86	70	56	4	18	86	80	60.9	53.2	2.83 E-27*
SELOR	84	**88**	**90**	**98**	**84**	**98**	**86**	**100**	**88**	**90.7**	85.7	-

(c) Percentage of best (Yelp)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	Avg.	Agr.	P-value
Lime	30	36	24	40	44	34	14	**48**	38	34.2	67.6	8.87 E-03*
Anchor	24	20	**36**	16	8	12	16	10	20	18.0	83.6	5.63 E-18*
SENN	2	4	4	2	2	2	4	0	2	2.4	96.3	6.84 E-40*
RCN	2	2	2	0	0	0	6	6	**0**	2.0	96.3	6.84 E-40*
SELOR	**44**	**40**	**36**	**48**	**50**	**54**	**64**	40	**44**	**46.7**	64.8	-

(d) Percentage of best (Adult)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	Avg.	Agr.	P-value
Lime	0	2	0	0	8	0	0	0	2	1.3	96.7	1.72 E-64*
Anchor	22	18	20	16	2	2	16	2	26	13.8	82.9	1.23 E-35*
SENN	6	10	8	8	34	4	2	2	12	9.6	83.3	2.30 E-36*
RCN	10	10	8	12	10	0	6	10	26	10.2	82.9	1.23 E-35*
SELOR	**62**	**60**	**64**	**64**	**46**	**94**	**76**	**86**	**34**	**65.1**	58.4	-

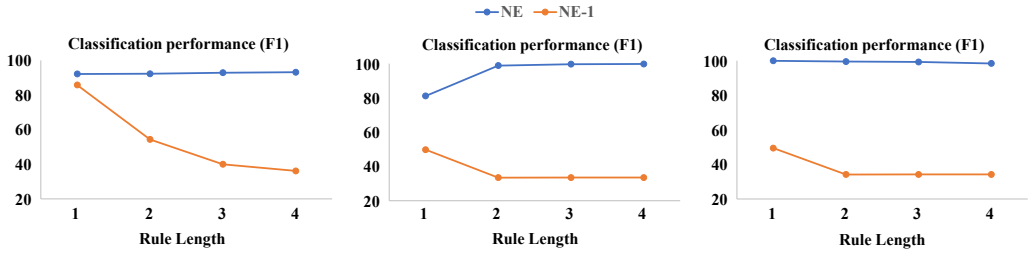
processing. To reduce incongruity with other explanations, we hide the score provided by SENN and join the chosen predicates.

- **RCN** chooses a rule from a predefined rule set made by random forest. As the random forest is trained with the bag-of-words of training data, the form of the rule also aligns with the frequency of words.
- **SELOR** recursive deep antecedent generation chooses atoms with the largest weight sequentially. All our atoms are existence of a word (e.g., if "*good*" exists), so a rule becomes the list of words. We join these words to explain the given sample.

User instruction and labeling detail. We provided instructions for participants in the form of the guideline file (*Labeling_Guidelines_User_Study_Table3.pdf*) with detailed description of the task and labeling examples. Participants received an Excel file with empty labels, which they were instructed to fill out and return. The snapshot of the Excel file is also attached as a separate file (*Screenshot_User_Study_Table3_1.PNG*, *Screenshot_User_Study_Table3_2.PNG*). We originally allowed multiple choices as best explanations, but labelers found it unclear how to decide two explanations are equally good. As this guideline led to confusion and further lower agreement among labelers, we updated the guideline to allow only one best explanation. We conducted the user study twice. During the first survey, we hired three participants. For Yelp dataset, each participant was paid 22.5\$ per hour with the total budget 45\$. For Adult dataset, each participant was paid 7.5\$ per hour

Table 11: Performance comparison of SELOR and a fully transparent model, Random Forest. The backbone model of textual dataset for SELOR is RoBERTa.

	Yelp		Clickbait		Adult	
	F1	AUC	F1	AUC	F1	AUC
Random Forest	73.03	80.40	44.29	60.25	65.60	66.15
SELOR	97.13	97.78	74.20	64.14	77.37	70.36



(a) Yelp (b) Clickbait (c) Adult

Figure 5: The prediction performance of consequent estimator with varying length of antecedents. NE-1 denotes the estimator which is only trained with length-1 antecedents.

with total budget 37.5$. At the second survey, we hired six participants, and each participant was paid 7$ per hour for both datasets. The total budget we spent in the second survey was 56$.

Results of all participants. Table 10 provides more detailed result including that of each participant.

Further discussion on user study results. Table 10a shows that participants have a low level of agreement on RCN. This is because people have varying preferences for the logical connective NOT. NOT denotes that the prediction is made due to the absence of a particular feature in the text. One participant (P3) considered most explanations that contained NOT to be noisy because s/he seldomly made decisions based on the absence of a word.

C.5 Additional Experimental Results

We describe additional experimental results to support the prediction performance and explanation quality of SELOR.

C.5.1 Comparison with Fully Transparent Model

Tree-based models are popular explainable models because their decision process is fully transparent. However, fully transparent models such as decision trees and random forests cannot achieve comparable prediction performance to deep models as shown in Table 11.

C.5.2 Effectiveness of Neural Consequent Estimator

The Fig. 5 shows the prediction performance of our neural consequent estimator (NE) for antecedents of varying length. Our consequent estimator shows reasonable performance in most cases. NE-1 is the estimator that is only pretrained with length-1 antecedents and hence cannot learn the relationship among atoms. Its prediction ability dramatically drops for rules longer than 1.

Also, we explore the effect of neural consequent estimator to the overall model performance. Fig. 6 demonstrates that SELOR is not highly sensitive to the number of samples used in pretraining the neural consequent estimator, although it requires a minimum level of prediction ability. Additionally, a larger number of samples are needed for more difficult dataset such as Clickbait.

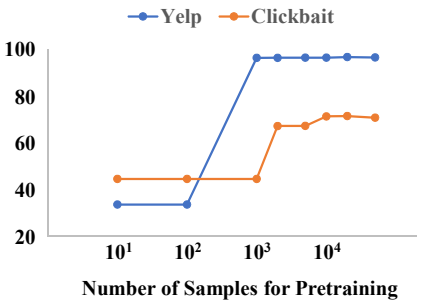


Figure 6: The performance of SELOR with varying number of samples used in pretraining of the neural consequent estimator.

C.5.3 Using Different Logical Connectives

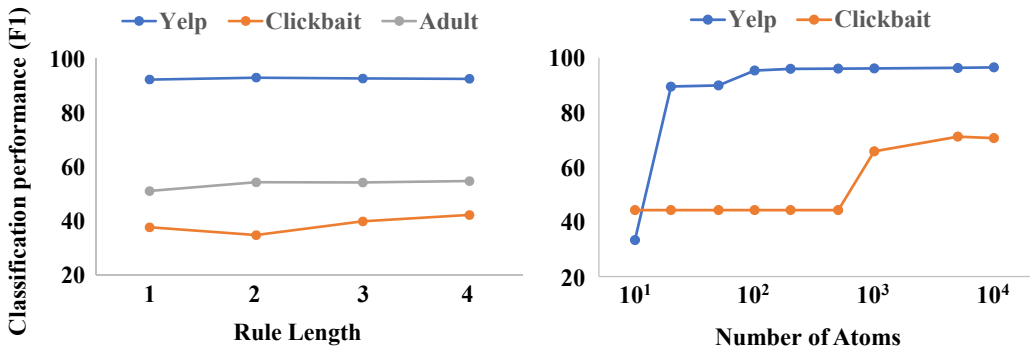
We investigate the performance in terms of F1 of different logical connectives on Yelp using BERT as a base model. First, joining atoms with logical connectives **OR** leads to a prediction performance of 96.21, which is similar to the original model using the **AND** connectives. We also change half of atoms to non-existence rules, which indicates the non-existence of a word (e.g. "NOT *awesome*" means the given instance does not contain the word "*awesome*"). The performance changes to 94.46, and this is natural as the information capacity of non-existence is usually smaller than the existence rules. Additionally, we try **ORDERED AND**, which considers the order of atoms. For example, "*not* BEFORE *happy*" and "*happy* BEFORE *not*" will be treated as different antecedents although they have the same words in atoms. Its performance is 96.93, as the amount of information in the rule increases.

C.5.4 Hyper-Parameter Sensitivity Analysis

We conduct analysis to test sensitivity of two hyper-parameters: antecedent length L and number of atoms $|\mathcal{O}|$.

Impact of hyper-parameters on prediction performance. Fig. 7a shows that SELOR is not sensitive to the length of antecedent although longer antecedents yield better result in general. Fig 7b shows that the number of atoms required for good performance varied by datasets. The more difficult dataset, Clickbait, requires a larger number of atoms to get reasonable performance. However, after certain points, the prediction performance of our method becomes insensitive to number of atoms.

Impact of hyper-parameters on explainability. Table 12 show how human precision of explanations change with the antecedent length. Antecedents of all lengths, including short antecedents with only one atom, offer a certain level of explainability; The average percentage of good for Length 1 antecedent is 79.7%. Meanwhile, longer antecedents tend to improve human precision. This indicates the longer antecedents contain more useful information for decision making as it has more chances to find a good atom, resulting in greater precision. Note that the *length of antecedent* is the maximum length of the antecedent; our method can automatically generate shorter antecedents than the default length by electing the NULL atom. Table 13 shows how human precision of explanations change with the number of candidate atoms. In particular, $1,000$ means that we use the top $1,000$ frequent words as candidate atoms. The explanation quality increases with increasing number of atoms, up to a certain points(i.e., $1,000$ atoms). After this point, there is no statistically significant gain in explainability, demonstrating that SELOR requires a reasonable size of approximately $1,000$ atoms to provide a good explanation. This finding aligns with the observations in [55], which shows that analyzing and explaining text contents such as restaurant reviews and news articles does not require a large vocabulary.



(a) Antecedent Length (b) Number of Atoms

Figure 7: The predictive performance of our model with varying antecedent lengths and the number of atoms.

Table 12: User study results on human precision with varying antecedent lengths. Participants P1-P6 were asked to annotate whether an explanation is a reasonable rationale for the prediction. For each length, we report the percentage of explanations that are considered good (a) or best (b). Avg. and Agr. denote the average and inter-participant agreement, respectively. P-values from t-test indicates the statistical significance of the experiment. We mark one star (*) if the p-value is lower than 0.05. Best results are highlighted in **bold**.

(a) Percentage of good

	P1	P2	P3	P4	P5	P6	Avg.	Agr.	P-value
Length 1	76	74	76	76	96	80	79.7	87.6	1.31 E-11*
Length 2	92	90	86	94	**100**	84	91.0	91.3	8.93 E-04*
Length 3	96	94	88	92	**100**	84	92.3	89.7	3.73 E-03*
Length 4	**100**	**100**	**90**	**98**	**100**	**86**	**95.6**	91.9	-

(b) Percentage of best

	P1	P2	P3	P4	P5	P6	Avg.	Agr.	P-value
Length 1	8	4	2	6	10	8	6.3	94.3	2.38 E-43*
Length 2	24	26	10	12	14	10	16.0	81.6	6.98 E-23*
Length 3	20	28	18	28	14	26	22.3	81.7	1.58 E-16*
Length 4	**56**	**50**	**68**	**62**	**72**	**58**	**61.0**	66.5	-

Table 13: User study results on human precision with varying number of atoms. Participants P1-P6 were asked to annotate whether an explanation is a reasonable rationale for the prediction. For each length, we report the percentage of explanations that are considered good (a) or best (b). Avg. and Agr. denote the average and inter-participant agreement, respectively. P-values from t-test indicate the statistical significance of the experiment. We mark one star (*) if the p-value is lower or close to 0.05. Best results are highlighted in **bold**.

(a) Percentage of good

# Atoms	P1	P2	P3	P4	P5	P6	Avg.	Agr.	P-value
10	10	12	14	8	26	24	15.7	94.5	5.86 E-02*
100	34	40	46	34	64	58	46.0	94.5	5.86 E-02*
1000	84	82	86	78	**100**	80	85.0	94.1	1.58 E-01
5000	**100**	**98**	**98**	**90**	**100**	86	**95.3**	91.7	**-**
10000	96	96	94	88	**100**	**90**	94.0	91.7	2.06 E-01

(b) Percentage of best

# Atoms	P1	P2	P3	P4	P5	P6	Avg.	Agr.	P-value
10	0	2	4	0	0	0	1.0	77.1	6.25 E-5**
100	6	6	4	6	4	2	4.7	76.9	8.16 E-5**
1000	30	38	32	30	44	34	34.7	73.5	2.45 E-3**
5000	**54**	**56**	**56**	**52**	**48**	**50**	**52.7**	71.2	**-**
10000	46	48	52	46	**48**	46	47.7	74.3	2.20 E-1

Relation between prediction performance and explainability. Throughout Fig. 7, Table 12, and Table 13, we could not find concrete evidence for a trade-off between explainability and prediction performance. Rather, we found models with good explainability also produce good prediction performance (i.e., models with antecedent length 2 to 4, models having number of atoms $1,000$ or more atoms). This is consistent with our framework $p(y|\mathbf{x}, b) = \sum_{\boldsymbol{\alpha}} p(y|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|\mathbf{x}, b)$, which passes information from input to prediction only via explanations, as opposed to other unexplainable parts. Thus, the expressivity of explanations and the capacity of the model are tightly related. If the hyperparameter settings significantly constrain the expressivity of the explanations (such as limiting the number of atoms to 10), both explanation quality and predictive performance will decrease significantly.

C.6 Explanation Stability

Do explanations keep the same in different runs? We conduct experiments to confirm that our model usually generates unique explanations for the same instances in different runs. Comparing the model explanations trained with 5 seeds reveals that, on average, 90.04% of atoms were shared by explanations from different seeds, and 71.27% were identical on Yelp. This comparison suggests that our model generates a unique explanation for the same instance, even in the absence of a direct controlling factor. The reason why we can generate unique explanations is that we optimize the explanation generator with two globally consistent rewards in Eq. 2: 1) human's prior belief about which explanation types are good and 2) the explanation (rule) confidence that is measured by the global prediction accuracy over the entire training corpus given the rule. Since the second reward is a real number instead of a discrete value and has a globally consistent meaning, the optimal explanation is usually unique and stable, leading to similar results when trained with different random seeds.

Do similar instances lead to similar explanations? Table 14 shows examples of generated explanations for similar inputs. SELOR successfully maintains its explanation when minor changes are made to input words, but suggests a new explanation when critical changes are made. In case (a), for example, our method provides the same explanation when the words "*pizza*" and "*waiters*" are changed to "*pasta*" and "*servers*". However, when sentiment-related words such as "*cold*" and "*rude*" are changed, it adapts to the new words and gives a new explanation.

Table 14: Generated explanations of samples and their perturbation. The manually changed words are highlighted in **bold**

Case	Sample	Model Explanation	Prediction
(a)	This place is awful. The pizza was cold, and the waiters were rude.	awful, cold, rude	Negative
	This place is awful. The **pasta** was cold, and the **servers** were rude.	awful, cold, rude	Negative
	This place is awful. The pizza was **undercooked**, and the waiters were **unfriendly**.	awful, undercooked, unfriendly	Negative
(b)	I love here. It was an amazing experience to eat a cheesy macaroni.	love, amazing, cheesy	Positive
	I **recommend** here. It was **a happy** experience to eat a cheesy macaroni.	recommend, happy, cheesy	Positive
	I **hate** here. It was **a bad** experience to eat a cheesy macaroni.	hate, bad, experience	Negative
	I ordered three tacos and all 3 were downright lousy. Can't remember the last time I had food this bad. The shrimp taco was overbreaded and in a sickly sweet sauce, the shredded beef taco was very tiny and thankfully, I can't remember what the third taco tasted like. To the reviewer who posted that these tacos are top notch.... what are you smoking? I waited forever to get my food and saw numerous other people who came in after me get their food. Waiter was MIA. Not coming back....ever.	lousy, bad, waited, forever	Negative

(c)	I ordered three tacos and all 3 were downright lousy. Can't remember the last time I had food this bad. The shrimp taco was overbreaded and in a sickly sweet sauce, the shredded beef taco was very tiny and thankfully, I can't remember what the third taco tasted like. To the reviewer who posted that these tacos are top notch.... what are you smoking? I waited **a little** to get my food and saw numerous other people who came in after me get their food. Waiter was MIA. Not coming back....ever.	lousy, bad, waited, not	Negative
	I ordered three **awful, terrible** tacos and all 3 were downright lousy. Can't remember the last time I had food this bad. The shrimp taco was overbreaded and in a sickly sweet sauce, the shredded beef taco was very tiny and thankfully, I can't remember what the third taco tasted like. To the reviewer who posted that these tacos are top notch.... what are you smoking? I waited forever to get my food and saw numerous other people who came in after me get their food. Waiter was MIA. Not coming back....ever.	awful, terrible, waited, forever	Negative
(d)	I had an amazing 4 course meal here with my family from philadlephia. my father runs a farmers market there and was very impressed with their use of seasonal and local foods. We had an amazing pork belly salad and I had duck wrapped in bacon and stuffed with pate, which sounds insanely heavy, but it was not; the portion was small enough not to be overwhelmed and it was not overly greasy at all. It was a fantastic meal. I think l'etoile is on par with top restaurants in bigger cities.	amazing, family, stuffed, fantastic	Positive
	I had **a great** 4 course meal here with my family from philadlephia. my father runs a farmers market there and was very impressed with their use of seasonal and local foods. We had an amazing pork belly salad and I had duck wrapped in bacon and stuffed with pate, which sounds insanely heavy, but it was not; the portion was small enough not to be overwhelmed and it was not overly greasy at all. It was a fantastic meal. I think l'etoile is on par with top restaurants in bigger cities.	great, family, amazing, fantastic	Positive
	I had an **awful** 4 course meal here with my family from philadlephia. my father runs a farmers market there and was very **disappointed** with their use of seasonal and local foods. We had a **terrible** pork belly salad and I had duck wrapped in bacon and stuffed with pate, which sounds insanely heavy; **and it was right**; the portion was **too small to be full** and it **was overly** greasy at all. It was a **bad** meal. I think l'etoile is on par with **bad** restaurants in bigger cities.	awful, disappointed, terrible, bad	Negative