NEUROLOGIC: FROM NEURAL REPRESENTATIONS TO INTERPRETABLE LOGIC RULES

Anonymous authors
Paper under double-blind review

ABSTRACT

Rule-based explanation methods offer rigorous and globally interpretable insights into neural network behavior. However, existing approaches are mostly limited to small fully connected networks and depend on costly layer-wise rule extraction and substitution processes. These limitations hinder their generalization to more complex architectures such as Transformers. Moreover, existing methods produce shallow, decision-tree-like rules that fail to capture rich, high-level abstractions in complex domains like computer vision and natural language processing. To address these challenges, we propose NEUROLOGIC, a novel framework that extracts interpretable logical rules directly from deep neural networks. Unlike previous methods, NEUROLOGIC can construct logic rules over hidden predicates derived from neural representations at any chosen layer, in contrast to costly layer-wise extraction and rewriting. This flexibility enables broader architectural compatibility and improved scalability. Furthermore, NEUROLOGIC supports richer logical constructs and can incorporate human prior knowledge to ground hidden predicates back to the input space, enhancing interpretability. We validate NEUROLOGIC on Transformer-based sentiment analysis, demonstrating its ability to extract meaningful, interpretable logic rules and provide deeper insights—tasks where existing methods struggle to scale. Our code is available at: `https://github.com/NeuroLogic2026/NeuroLogic`.

1 INTRODUCTION

Among various explanation types for deep neural networks (DNNs), such as attributions (Selvaraju et al., 2017) and hidden semantics (Bau et al., 2017), rule-based methods that generate global logic rules over input sets, rather than local rules for individual samples, offer stronger interpretability and are highly preferred (Pedreschi et al., 2019). However, most existing rule-based explanation methods (Cohen, 1995; Zilke et al., 2016; Zarlenga et al., 2021a; Hemker et al., 2023) suffer from several limitations. We highlight three key issues: (1) they mostly rely on layer-by-layer rule extraction and rewriting to derive final rules, which introduces scalability limitations; (2) they are primarily tailored to fully connected networks and fail to generalize to modern deep neural network architectures such as Transformers; (3) the rules they produce are often shallow and decision-tree-like, lacking the ability to capture high-level abstractions, which limits their effectiveness in complex domains.

To this end, we introduce NEUROLOGIC, a modern rule-based framework designed to address architectural dependence, limited scalability, and the shallow nature of existing decision rules. Our approach is inspired by Neural Activation Patterns (NAPs) (Geng et al., 2023; 2024) which are subsets of neurons that consistently activate for inputs belonging to the same class. Specifically, for any given layer, we identify salient neurons for each class and determine their optimal activation thresholds, converting these neurons into *hidden predicates*. These predicates represent high-level features learned by the model, where a true value indicates the presence of the corresponding feature in a given input. Based on these predicates, NEUROLOGIC constructs first-order logic (FOL) rules in a fully data-driven manner to approximate the internal behavior of neural networks.

The remaining challenge is to ground these hidden predicates in the original input space to ensure interpretability. Unlike existing approaches that can only produce shallow, decision-tree-like rules, NEUROLOGIC features a flexible design that supports a wide range of interpretable surrogate methods, such as program synthesis, to learn rules with richer and more expressive structures. To

demonstrate its capabilities, we apply NEUROLOGIC to extract logic rules from Transformer-based sentiment analysis—a setting where traditional rule-extraction methods struggle to scale. *To the best of our knowledge, this is the first approach capable of extracting global logic rules from modern, complex architectures such as Transformers.* We believe NEUROLOGIC represents a promising step toward opening the black box of DNNs. Our contributions are summarized as follows:

- We propose NEUROLOGIC, a novel framework for extracting interpretable global logic rules from DNNs. By abandoning the costly layer-wise rule extraction and substitution paradigm, NEUROLOGIC achieves greater scalability and broad architectural compatibility.
- Experimental results on small-scale benchmarks demonstrate that NEUROLOGIC produces more compact rules with higher efficiency than state-of-the-art methods, while maintaining strong fidelity and predictive accuracy.
- We further showcase the practical feasibility of NEUROLOGIC in extracting meaningful logic rules and providing insights into the internal mechanisms of Transformers—an area where existing approaches struggle to scale effectively.

2 PRELIMINARY

Neural Networks for Classification Tasks We consider a general deep neural network N used for classification. Let $z_i^l(x)$ denote the value of the i-th neuron at layer l for a given input x. We do not assume any specific form for the transformation between layers, that is, the mapping from z^l to z^{l+1} can be arbitrary. This abstraction allows our analysis to be broadly applied across architectures. The network N as a whole functions as $\mathbf{F}^{<N>} : X \to \mathbb{R}^{|C|}$, mapping an input $x \in X$ from the dataset to a score vector over the class set C. The predicted class is then given by $\hat{c} = \arg\max_{c \in C} \mathbf{F}_c^{<N>}(x)$.

First-Order Logic First-Order Logic (FOL) is a formal language for stating interpretable rules about objects and their relations/attributes. It extends propositional logic by introducing *quantifiers* such as: *Universal quantifier* ($\forall$): meaning "for all", e.g., $\forall x\, p(x)$ means $p(x)$ holds for every x and *Existential quantifier* ($\exists$): meaning "there exists", e.g., $\exists x\, p(x)$ means there exists at least one x for which $p(x)$ holds. We focus on FOL rules in *Disjunctive Normal Form (DNF)*, which are disjunctions (ORs) of conjunctions (ANDs) of *predicates*. A *predicate* is a simple condition or property on the input, e.g., $p_i(x)$. A *clause* is a conjunction (AND) of predicates, such as $p_1(x) \wedge p_2(x) \wedge \neg p_3(x)$. A DNF rule looks like a logical OR of multiple clauses:

$$\forall x, \quad (p_1(x) \wedge p_2(x)) \vee (p_3(x) \wedge \neg p_4(x)) \Rightarrow \textit{Label}(x) = c, \tag{1}$$

meaning that for every input x, if any clause is satisfied, it is assigned to class c. This structured form makes the rules easy to interpret and understand.

3 THE NEUROLOGIC FRAMEWORK

3.1 IDENTIFYING HIDDEN PREDICATES

For a given layer l, we aim to identify a subset of neurons that are highly indicative of a particular class $c \in C$. These neurons form what are known as Neural Activation Patterns (NAPs) (Geng et al., 2023; 2024). A neuron is considered part of the NAP for class c if its activation is consistently higher for inputs from class c compared to inputs from other classes. This behavior suggests that such neurons encode class-specific latent features at layer l, as discussed in (Geng et al., 2024).

To identify the NAP for a specific class c, we evaluate how selectively each neuron responds to class c versus other classes. Since each neuron's activation is a scalar value, we can assess its discriminative power by learning a threshold t. This threshold separates inputs from class c and those from other classes based on activation values.

Formally, we consider a neuron to support class c if its activation $z_j^l(x)$ for input x satisfies $z_j^l(x) \geq t$. If this condition holds, we classify x as belonging to class c; otherwise, it is classified as not belonging to c. To quantify the effectiveness of a threshold t, we use the *purity* metric, defined as:

$$\text{Purity}(t) = \frac{\left|\{x \in X_c : z_j^l(x) \geq t\}\right|}{|X_c|} + \frac{\left|\{x \in X_{\neg c} : z_j^l(x) < t\}\right|}{|X_{\neg c}|}$$

Here, X_c denotes the set of inputs from class c, while $X_{\neg c}$ denotes inputs from all other classes. A high purity value means the neuron cleanly separates class c from others, whereas a low value suggests ambiguous or overlapping activation responses. We conduct a linear search to determine the optimal threshold t as its final purity.

3.2 Determining the Logical Rules

Formally, a predicate p_j at layer l, together with its corresponding threshold t_j, is defined as $p_j(x) := \mathbb{I}[z_j^{(l)}(x) \geq t_j]$. A True assignment indicates the presence of the specific latent feature of class c for input x, while a False assignment signifies its absence. Intuitively, the more predicates that fire, the stronger the evidence that x belongs to class c. However, this raises the question: to what extent should we believe that x belongs to class c based on the pattern of predicate activations?

We address this question using a data-driven approach. Let $P_c^{(l)} = \{p_1, ..., p_m\}$ be the m predicates retained for class c. Evaluating $P_c^{(l)}$ on every class example $x \in X_c$ gives a multiset of binary vectors $p(x) \in \{0,1\}^m$. Each distinct vector can be treated as a clause, and the union of all clauses forms a DNF rule:

$$\forall x, \Big(\bigvee_{v \in \mathcal{V}_c} \Big(\bigwedge_{i:v_i=1} p_i(x) \wedge \bigwedge_{i:v_i=0} \neg p_i(x) \Big) \Big) \implies Label(x) = c$$

where $\mathcal{V}_c$ is the set of unique activation vectors for X_c .

3.3 Grounding Predicates to the Input Feature Space

The final step is to *ground* these hidden predicates in the input space to make them human-interpretable. In this work, we present simple approaches for grounding predicates in simple input domains, as well as in the complex input domain of large vocabulary spaces for Transformers.

Grounding Predicates in Simple Input Domains For deep neural networks (DNNs) applied to tasks with simple input domains (e.g., tabular data), we aim to ground each predicate p_j directly in the raw input space. This enables more transparent and interpretable logic rules.

We reframe the grounding task as a supervised classification problem. For a given predicate p_j, we collect input examples where the predicate is activated versus deactivated, and then learn a symbolic function that approximates this distinction.

Formally, for a target class c and predicate p_j, we define the activation set and deactivation set, respectively, as $D_1^{(j)} = \{x \in X_c \mid p_j(x) = 1\}, D_0^{(j)} = \{x \in X_c \mid p_j(x) = 0\}$. These are combined into a labeled dataset $alignD^{(j)} = \{(x, y) \mid x \in D_1^{(j)} \cup D_0^{(j)},\ y = p_j(x)\}$.

Then, to obtain expressive, compositional and human-readable logic rules as explanations, we employ program synthesis to learn a symbolic expression ϕ_j from a domain-specific language (DSL) $\mathcal{L}$. Unlike traditional decision-tree-like rules, the symbolic language $\mathcal{L}$ is richer: a composable grammar over input features that supports not only logical and comparison operators but also linear combinations and nonlinear functions. Specifically, the language includes:

- *Atomic abstractions* formed by applying threshold comparisons to linear or nonlinear functions of the input features, for example,
$$a := f(x) \leq \theta \quad \text{or} \quad f(x) > \theta, \tag{2}$$
where $f(x)$ can be any linear or nonlinear transformation, such as polynomials, trigonometric functions, or other basis expansions.
- *Logical operators* to combine these atomic abstractions into complex expressions:
$$\phi ::= a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2. \tag{3}$$

The synthesis objective is to find an expression $\phi_j \in \mathcal{L}$ that minimizes a combination of classification loss and complexity, formally:

$$\phi_j \in \arg\min_{\phi \in \mathcal{L}} \Big[\mathcal{L}_{\text{cls}}(\phi; D^{(j)}) + \lambda \cdot \Omega(\phi) \Big], \tag{4}$$

where $\mathcal{L}_{\text{cls}}$ measures how well ϕ approximates the predicate activations in $D^{(j)}$, $\Omega(\phi)$ penalizes the complexity of the expression (e.g., number of literals or tree depth), and λ balances the trade-off between accuracy and interpretability.

This grounding approach also supports decision-tree-like rules, which are commonly used in existing methods. In this context, such rules can be viewed as a special case of the above atomic abstractions, where $f(x)$ corresponds to individual features.

A simpler alternative is to leverage off-the-shelf decision tree algorithms: we train a decision tree classifier f_j^{DT} such that

$$f_j^{\text{DT}}(x) \approx p_j(x), \quad \forall x \in X_c. \tag{5}$$

The resulting decision tree provides a simpler rule-based approximation of predicate activations, effectively grounding p_j in the input space in an interpretable manner.

Grounding predicates in the vocabulary space The input space in NLP domains (i.e., vocabulary spaces) is typically extremely large, making it difficult to ground rules onto raw feature vectors. In such domains, it is more effective to incorporate human prior knowledge like words, tokens, or linguistic structure that are more semantically meaningful and ultimately guide the predictions made by transformer-based models (Tenney et al., 2019a). In light of this, we define a set of atomic abstractions over the vocabulary spaces. Each atomic abstraction corresponds to a template specifying keywords along with their associated lexical structures. To ground the learned hidden predicates to this domain knowledge, we leverage causal inference (Zarlenga et al., 2021b; Vig et al., 2020).

Formally, let $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ be the set of atomic abstractions derived from domain knowledge (e.g., keywords or lexical patterns), and let p_j be a learned hidden predicate extracted from the model's internal representations, and x be an input instance (e.g., a text sample).

We define a causal intervention $do(\neg a_i)$ as flipping the truth value of atomic abstraction a_i in the input x (e.g., masking the keyword associated with a_i). The grounding procedure tests whether flipping a_i changes the truth of the hidden predicate p_j:

$$\textit{If} \quad p_j(x) = \textit{True} \quad \textit{and} \quad p_j\big(do(\neg a_i)(x)\big) = \textit{False}, \tag{6}$$

then we infer a causal dependence of p_j on a_i, grounding p_j to the atomic abstraction a_i.

By iterating over all atomic abstractions $a_i \in \mathcal{A}$, we establish a mapping:

$$G : p_j \mapsto \{a_i \in \mathcal{A} \mid \textit{flipping } a_i \textit{ negates } p_j\}, \tag{7}$$

which grounds the hidden predicate p_j in terms of semantically meaningful domain knowledge.

4 EVALUATION

We evaluate NEUROLOGIC across two settings, focusing primarily on its performance in challenging, large-scale scenarios. While detailed results for small-scale benchmarks are provided in Appendix B due to space constraints, those experiments demonstrate that NEUROLOGIC *strikes a favorable balance by effectively combining faithfulness and computational efficiency to outperform existing rule-based methods*. The remainder of this section focuses on transformer-based sentiment analysis, representing a demanding real-world application where existing methods often fail to scale. Our results in this context highlight the practical viability and scalability of NEUROLOGIC.

Class	**Layer 1** (38.92%)		**Layer 2** (43.70%)		**Layer 3** (62.84%)		**Layer 4** (66.92%)		**Layer 5** (78.89%)		**Layer 6** (80.58%)	
	# Clauses	*Length*	*# Clauses*	*Length*	*# Clauses*	*Length*	*# Clauses*	*Length*	*# Clauses*	*Length*	*# Clauses*	*Length*
Anger	70	4.29	91	4.27	91	4.82	75	4.51	42	4.19	33	4.15
Joy	58	3.62	50	4.18	58	4.81	48	4.98	35	5.14	20	4.25
Optimism	34	4.88	32	4.38	47	5.60	49	5.65	26	4.65	23	5.57
Sadness	78	4.76	53	4.36	84	5.25	73	3.78	46	4.72	38	4.92

Table 1: Number of clauses (*# Clauses*) and average clause length (*Length*) for each emotion class. Per-layer rule-set accuracy is shown in parentheses following the layer number.

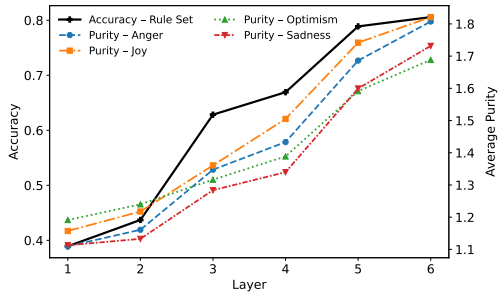


Figure 1: The purity of predicates correlates with accuracy as layers go deeper.

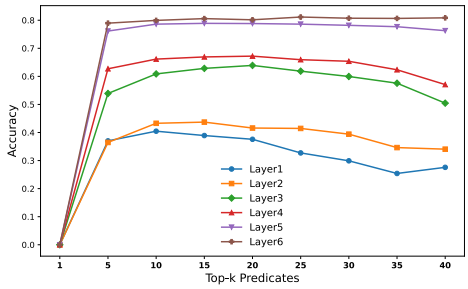


Figure 2: The impact of the number of predicates affects the rule model.

Layer 1		Layer 2		Layer 3		Layer 4		Layer 5		Layer 6	
Keyword	Pattern	Keyword	Pattern	Keyword	Pattern	Keyword	Pattern	Keyword	Pattern	Keyword	Pattern
the	at_end	i	at_start	it	after_verb	sad	at_end	sad	at_end	sad	at_end
i	at_end	i	before_verb	you	after_verb	in	after_subject	sad	after_verb	lost	after_subject
of	at_end	user	at_start	so	after_subject	sad	after_verb	depression	at_end	depression	at_end
when	at_start	user	before_subject	but	at_start	sad	at_start	me	after_subject	sad	after_verb
and	after_subject	a	after_subject	on	after_verb	the	before_verb	at	after_verb	sad	after_subject
at	after_verb	i	after_verb	a	at_end	think	after_subject	sad	after_subject	sad	at_start
be	after_subject	a	after_verb	of	at_start	sad	after_subject	sad	at_start	sadness	at_end
to	at_end	is	after_subject	you	after_subject	depression	at_end	sadness	at_end	depression	at_start
was	after_verb	to	after_verb	it	after_subject	in	at_start	depression	after_verb	depressing	at_end
sad	at_end	i	after_subject	just	at_start	by	after_verb	be	after_verb	depressing	after_subject
when	before_subject	user	before_verb	so	after_verb	user	after_verb	am	after_subject	lost	at_start
like	after_subject	it	at_start	can	after_subject	be	after_subject	was	before_verb	nightmare	at_end
when	before_verb	is	at_start	of	before_verb	think	at_start	at	after_subject	sadness	after_verb
like	after_verb	and	after_verb	can	before_verb	with	after_subject	at	at_start	lost	at_end
are	after_subject	my	after_verb	sad	after_verb	really	after_subject	depressing	at_end	anxiety	after_verb

Table 2: Top 15 keyword linguistic pattern pairs for class *Sadness* learned by the top DNF rules.

Setup and Baselines We evaluate NEUROLOGIC on the *Emotion* task from the TWEETEVAL benchmark, which contains approximately 5,000 posts from Twitter, each labeled with one of four emotions: *Anger*, *Joy*, *Optimism*, or *Sadness* (Barbieri et al., 2020). All experiments use the pretrained model, a 6-layer DistilBERT fine-tuned on the same TweetEval splits (Schmid, 2024; Sanh et al., 2020). The pretrained model has a test accuracy of 80.59%. The model contains approximately 66 million parameters, and we empirically validate that existing methods fail to efficiently scale to this level of complexity. For rule grounding, we approximate predicate-level interventions by masking tokens that instantiate an atomic abstract a_i and flip an active DNF clause to `False`, thereby identifying a_i as its causal grounder. In our study, each a_i is defined as a *(keyword, linguistic pattern)* pair, where the linguistic pattern may include structures such as `at_start`. We benchmark the grounded rules produced by NEUROLOGIC against a classical purely lexical baseline.[1] EMOLEX (Mohammad & Turney, 2013) tags a tweet as *Sadness* whenever it contains any word from its emotion dictionary. This method relies on isolated keyword matching, with syntactical or other linguistic patterns ignored. Additional details are provided in the Appendix C.5.

Identifying Predicates We extract hidden predicates from all six Transformer layers and observe that, as layers deepen, the predicates tend to exhibit higher purity, from average 1.1 to 1.8. This trend also correlates with the test accuracy from around 40 % to 80% of our rule-based model, as illustrated in Figure 1. These results suggest that deeper layers capture more essential and task-relevant decision-making patterns, consistent with prior findings in (Geng et al., 2023; 2024). Another notable observation is that, surprisingly, a small number of predicates, specifically the top five, are often sufficient to explain the model's behavior. As shown in Figure 2, including more predicates beyond this point can even reduce accuracy, particularly in shallower layers (Layers 1 and 2). Middle layers (Layers 3 and 4) are less affected, while deeper layers (Layers 5 and 6) remain relatively stable. Upon closer inspection, we find that this decline is due to the added predicates being noisier and less semantically meaningful, thereby introducing spurious patterns that degrade rule quality.

Constructing Rules Based on Figure 2, we select the top-15 predicates to construct the DNF rules, meaning that each clause initially consists of 15 predicates. However, after distillation, we find that,

[1]To the best of our knowledge, no existing rule-extraction baseline is available for this task.

on average, fewer than five predicates are retained, as reported in Table 1. As a stand-alone classifier, the rule set distilled from *Layer 6* achieves an accuracy of *80.58%*, on par with the neural model's accuracy (80.59%). Notably, the distilled DNF rule sets primarily consist of positive predicates, with negations rarely appearing. This indicates that the underlying neuron activations function more like *selective filters*, each tuned to respond to specific input patterns rather than suppressing irrelevant ones. This aligns with the intuition that deeper transformer layers develop specialized units that favor and reinforce certain semantic or structural patterns, making the logic rules not only more compact but also more interpretable and faithful to the model's decision boundaries.

Grounding Rules To simplify our analysis, we focus on the *Sadness* class and the highest-scoring DNF rule per layer in Table 2. We claim this is empirically justified: Figure 3 (Appendix) shows that the class accuracy for each layer is explained significantly by the top DNF rule, so it effectively "decides" whether an example is labelled *Sadness* or not while the other rules handle outliers and more nuanced examples. In the earlier layers 1–2, high-frequency function keywords such as *the*, *i*, *of*, and *at* mostly describe surface positions i.e `at_end`. These words don't include any *Sadness* emotional keywords but rather provides syntactic cues like subject boundaries and sentence structuring. This observation mirrors earlier probing attempts on Transformer layers (Tenney et al., 2019b; Peters et al., 2018). In mid-layers 3–4, the introduction of explicit *Sad* keywords (*sad*, *depression*) starts to mix in with anchors like *in* and *you*. This indicates a slow transition where emotional content is starting to get attended to, but overall linguistic patterns that encode local syntax are still required for rules to fire. Finally, in the deep layers 5–6, it is evident that the top rule fires nearly exclusively on keywords that convey *Sadness* (*sad*, *lost*, textit depression, *nightmare*, *anxiety*, *bad*). Each keyword appears numerous times paired with different linguistic patterns, with certain keywords being refined and pushed up (*lost, sadness, sad, depression*). Additionally, we also see a pattern collapse in later layers where many of the same keywords appear with multiple patterns. Together, these trends show that deeper predicates become less about local syntax and more about whether a salient semantic token is present anywhere in the input–an observation shared in many other findings (de Vries et al., 2020; Peters et al., 2018).

5 Related Work

Interpreting neural networks with logic rules has been explored since before the deep learning era. These approaches are typically categorized into two groups: pedagogical and decompositional methods (Zhang et al., 2021; Craven & Shavlik, 1994). Pedagogical approaches approximate the network in an end-to-end manner. For example, classic decision tree algorithms such as CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993) have been adapted to extract decision trees from trained neural networks (Craven & Shavlik, 1995; Krishnan et al., 1999; Boz, 2002). In contrast, decompositional methods leverage internal network information, such as structure and learned weights, to extract rules by analyzing the model's internal connections. A core challenge in rule extraction lies in identifying layerwise value ranges through these connections and mapping them back to input features. While recent works have explored more efficient search strategies (Zilke et al., 2016; Zarlenga et al., 2021a), these methods typically scale only to very small networks due to the exponential growth of the search space with the number of attributes. Our proposed method, NeuroLogic, combines the efficiency of pedagogical approaches with the faithfulness of decompositional ones, making it scalable to modern DNN models. Its flexible design also enables the generation of more abstract and interpretable rules, moving beyond the limitations of shallow, decision tree–style explanations.

6 Conclusion

In this work, we introduce NeuroLogic, a novel framework for extracting interpretable logic rules from modern deep neural networks. NeuroLogic abandons the costly paradigm of layer-wise rule extraction and substitution, enabling greater scalability and architectural compatibility. Its decoupled design allows for flexible grounding, supporting the generation of more abstract and interpretable rules. We demonstrate the practical feasibility of NeuroLogic in extracting meaningful logic rules and providing deeper insights into the inner workings of Transformers.

REFERENCES

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644–1650, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.148. URL https://aclanthology.org/2020.findings-emnlp.148/.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3319–3327. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.354. URL https://doi.org/10.1109/CVPR.2017.354.

Rudolf K Bock, A Chilingarian, M Gaug, Frantisek Hakl, Thomas Hengstebeck, Marcel Jiřina, Jan Klaschka, Emil Kotrč, Petr Savickỳ, Sherry Towers, et al. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 516(2-3):511–528, 2004.

Olcay Boz. Extracting decision trees from trained neural networks. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pp. 456–461. ACM, 2002. doi: 10.1145/775047.775113. URL https://doi.org/10.1145/775047.775113.

Leo Breiman, J. H. Friedman, Richard A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

William W. Cohen. Fast effective rule induction. In Armand Prieditis and Stuart Russell (eds.), *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pp. 115–123. Morgan Kaufmann, 1995. doi: 10.1016/B978-1-55860-377-6.50023-2. URL https://doi.org/10.1016/b978-1-55860-377-6.50023-2.

Mark W. Craven and Jude W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In William W. Cohen and Haym Hirsh (eds.), *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*, pp. 37–45. Morgan Kaufmann, 1994. doi: 10.1016/B978-1-55860-335-6.50013-1. URL https://doi.org/10.1016/b978-1-55860-335-6.50013-1.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo (eds.), *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, pp. 24–30. MIT Press, 1995. URL http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.

Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. What's so special about BERT's layers? a closer look at the NLP pipeline in monolingual and multilingual models. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4339–4350, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.389. URL https://aclanthology.org/2020.findings-emnlp.389/.

Chuqin Geng, Nham Le, Xiaojie Xu, Zhaoyue Wang, Arie Gurfinkel, and Xujie Si. Towards reliable neural specifications. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11196–11212. PMLR, 2023. URL https://proceedings.mlr.press/v202/geng23a.html.

Chuqin Geng, Zhaoyue Wang, Haolin Ye, Saifei Liao, and Xujie Si. Learning minimal nap specifications for neural network verification. *arXiv preprint arXiv:2404.04662*, 2024.

Konstantin Hemker, Zohreh Shams, and Mateja Jamnik. Cgxplain: Rule-based deep neural network explanations using dual linear programs. In Hao Chen and Luyang Luo (eds.), *Trustworthy Machine Learning for Healthcare - First International Workshop, TML4H 2023, Virtual Event, May 4, 2023, Proceedings*, volume 13932 of *Lecture Notes in Computer Science*, pp. 60–72. Springer, 2023. doi: 10.1007/978-3-031-39539-0_6. URL https://doi.org/10.1007/978-3-031-39539-0_6.

R. Krishnan, G. Sivakumar, and P. Bhattacharya. Extracting decision trees from trained neural networks. *Pattern Recognit.*, 32(12):1999–2009, 1999. doi: 10.1016/S0031-3203(98)00181-2. URL https://doi.org/10.1016/S0031-3203(98)00181-2.

Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013. doi: 10.1111/j.1467-8640.2012.00460.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.2012.00460.x.

Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, and Franco Turini. Meaningful explanations of black box AI decision systems. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 9780–9784. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33019780. URL https://doi.org/10.1609/aaai.v33i01.33019780.

Bernard Pereira, Suet-Feung Chin, Oscar M Rueda, Hans-Kristian Moen Vollan, Elena Provenzano, Helen A Bardwell, Michelle Pugh, Linda Jones, Roslin Russell, Stephen-John Sammut, et al. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature communications*, 7(1):11479, 2016.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1179. URL https://aclanthology.org/D18-1179/.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. ISBN 1-55860-238-0.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Philipp Schmid. philschmid/DistilBERT-tweet-eval-emotion. https://huggingface.co/philschmid/DistilBERT-tweet-eval-emotion, 2024. Hugging Face model card, version accessed 31 Jul 2025.

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4593–4601. Association for Computational Linguistics, 2019a. doi: 10.18653/V1/P19-1452. URL https://doi.org/10.18653/v1/p19-1452.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. Causal mediation analysis for interpreting neural NLP: the case of gender bias. *CoRR*, abs/2004.12265, 2020. URL https://arxiv.org/abs/2004.12265.

Mateo Espinosa Zarlenga, Zohreh Shams, and Mateja Jamnik. Efficient decompositional rule extraction for deep neural networks. *CoRR*, abs/2111.12628, 2021a. URL https://arxiv.org/abs/2111.12628.

Mateo Espinosa Zarlenga, Zohreh Shams, and Mateja Jamnik. Efficient decompositional rule extraction for deep neural networks. *arXiv preprint arXiv:2111.12628*, 2021b.

Yu Zhang, Peter Tiňo, Ales Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.*, 5(5):726–742, 2021. doi: 10.1109/TETCI.2021.3100641. URL https://doi.org/10.1109/TETCI.2021.3100641.

Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. Deepred–rule extraction from deep neural networks. In *Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016, Proceedings 19*, pp. 457–473. Springer, 2016.

A Additional Details on Small-Scale Benchmarks

All experiments were conducted on a desktop equipped with a 2GHz Intel i7 processor and 32 GB of RAM. For each baseline, we used the original implementation and followed the authors' recommended hyperparameters to ensure a fair comparison. We performed all experiments across five different random folds to initialize the train-test splits, the random initialization of the DNN, and the random inputs for the baselines. Regarding the metric of average clause length, there appears to be a discrepancy in how it is computed in (Zarlenga et al., 2021a) and (Hemker et al., 2023). Specifically, Zarlenga et al. (2021a) seems to underestimate the average clause length. To ensure consistency and accuracy, we adopt the computation method used in (Hemker et al., 2023).

To maintain consistency, we used the same DNN topology (i.e., number and depth of layers) as in the experiments reported by (Zarlenga et al., 2021a). For NeuroLogic, we applied it to the last hidden layer and used the C5.0 decision tree as the grounding method for optimal efficiency. Below is a detailed description of each dataset:

MAGIC. The MAGIC dataset simulates the detection of high-energy gamma particles versus background cosmic hadrons using imaging signals captured by a ground-based atmospheric Cherenkov telescope (Bock et al., 2004). It consists of 19,020 samples with 10 handcrafted features extracted from the telescope's "shower images." The dataset is moderately imbalanced, with approximately 35% of instances belonging to the minority (gamma) class.

Metabric-ER. This biomedical dataset is constructed from the METABRIC cohort and focuses on predicting Estrogen Receptor (ER) status—a key immunohistochemical marker for breast cancer—based on 1,000 features, including tumor characteristics, gene expression levels, clinical variables, and survival indicators. Of the 1,980 patients, roughly 24% are ER-positive, indicating the presence of hormone receptors that influence tumor growth.

Metabric-Hist. Also derived from the METABRIC cohort (Pereira et al., 2016), this dataset uses the mRNA expression profiles of 1,694 patients (spanning 1,004 genes) to classify tumors into two major histological subtypes: Invasive Lobular Carcinoma (ILC) and Invasive Ductal Carcinoma (IDC). Positive diagnoses (ILC) account for only 8.7% of all samples, resulting in a highly imbalanced classification setting.

XOR. A synthetic dataset commonly used as a benchmark for rule-based models. Each instance $\mathbf{x}^{(i)} \in [0,1]^{10}$ is sampled independently from a uniform distribution. Labels are assigned according to a non-linear XOR logic over the first two dimensions:

$$y^{(i)} = \text{round}(x_1^{(i)}) \oplus \text{round}(x_2^{(i)}),$$

where $\oplus$ denotes the logical XOR operation. The dataset contains 1,000 instances.

B Evaluation on Small-Scale Benchmarks

Setup and Baselines We evaluate NeuroLogic against popular rule-based explanation methods C5.0 [2], ECLAIRE (Zarlenga et al., 2021a), and CGXPLAIN (Hemker et al., 2023) on four standard interpretability benchmarks: XOR, MB-ER, MB-HIST, and MAGIC. For each baseline, we use the original implementation and follow the authors' recommended hyperparameters. We evaluate all methods using five metrics: accuracy, fidelity (agreement with the original model), runtime, number of clauses, and average clause length, to assess both interpretability and performance.

Notably, NeuroLogic consistently produces the most concise explanations, as reflected by both the number of clauses and the average clause length. In particular, it generates rule sets with substantially shorter average clause lengths—for example, on MB-HIST, it achieves 2.7 ± 0.3 compared to 27.8 ± 7.6 by the previous state-of-the-art, CGXPLAIN. This conciseness, along with fewer clauses, directly enhances interpretability and readability by reducing overall rule complexity. These results highlight a key advantage of NeuroLogic and align with our design goal of improving interpretability.

[2] This represents the use of the C5.0 decision tree algorithm to learn rules in an end-to-end manner.

	Method	Accuracy (%)	Fidelity (%)	Runtime (s)	Number of Clauses	Avg Clause Length
XOR	C5.0	52.6 ± 0.2	53.0 ± 0.2	$\mathbf{0.1 \pm 0.0}$	1 ± 0	1 ± 0
	ECLAIRE	91.8 ± 1.0	91.4 ± 2.4	6.2 ± 0.4	87.0 ± 16.2	263.0 ± 49.1
	CGXPLAIN	$\mathbf{96.7 \pm 1.7}$	$\mathbf{92.4 \pm 1.1}$	9.1 ± 1.8	$\mathbf{3.6 \pm 1.8}$	10.4 ± 7.2
	NEUROLOGIC	89.6 ± 1.9	90.3 ± 1.6	1.2 ± 0.3	10.8 ± 3.5	$\mathbf{6.8 \pm 2.0}$
MB-ER	C5.0	92.7 ± 0.9	89.3 ± 1.0	20.3 ± 0.8	21.8 ± 3	72.4 ± 14.5
	ECLAIRE	$\mathbf{94.1 \pm 1.6}$	$\mathbf{94.7 \pm 0.2}$	123.5 ± 36.8	48.3 ± 15.3	137.6 ± 24.7
	CGXPLAIN	92.4 ± 0.7	94.7 ± 0.9	462.7 ± 34.0	5.9 ± 1.1	21.8 ± 3.4
	NEUROLOGIC	92.8 ± 0.9	92.7 ± 1.4	$\mathbf{6.0 \pm 1.2}$	$\mathbf{5.8 \pm 1.0}$	$\mathbf{3.7 \pm 0.2}$
MB-HIST	C5.0	87.9 ± 0.9	89.3 ± 1.0	16.06 ± 0.64	12.8 ± 3.1	35.2 ± 11.3
	ECLAIRE	88.9 ± 2.3	89.4 ± 1.8	174.5 ± 73.2	30.0 ± 12.4	74.7 ± 15.7
	CGXPLAIN	89.4 ± 2.5	89.1 ± 3.6	285.3 ± 10.3	5.2 ± 1.9	27.8 ± 7.6
	NEUROLOGIC	$\mathbf{90.7 \pm 0.9}$	$\mathbf{92.0 \pm 3.5}$	$\mathbf{2.3 \pm 0.2}$	$\mathbf{3.6 \pm 1.6}$	$\mathbf{2.7 \pm 0.3}$
MAGIC	C5.0	82.8 ± 0.9	85.4 ± 2.5	$\mathbf{1.9 \pm 0.1}$	57.8 ± 4.5	208.7 ± 37.6
	ECLAIRE	$\mathbf{84.6 \pm 0.5}$	87.4 ± 1.2	240.0 ± 35.9	392.2 ± 73.9	1513.4 ± 317.8
	CGXPLAIN	84.4 ± 0.8	$\mathbf{91.5 \pm 1.3}$	44.6 ± 2.9	7.4 ± 0.8	11.6 ± 1.9
	NEUROLOGIC	$\mathbf{84.6 \pm 0.5}$	90.8 ± 0.7	17.0 ± 1.5	$\mathbf{6.0 \pm 0.0}$	$\mathbf{3.6 \pm 0.1}$

Table 3: Comparison of rule-based explanation methods across different benchmarks. The best results are highlighted in bold.

By avoiding the costly layer-wise rule extraction and substitution paradigm employed by ECLAIRE and CGXPLAIN, NEUROLOGIC achieves significantly higher efficiency. Although C5.0 can be faster in some cases by directly extracting rules from DNNs, it often suffers from lower fidelity, reduced accuracy, or the generation of overly complex rule sets. For example, while C5.0 can complete rule extraction on XOR in just 0.1 seconds, its accuracy is only around 52%. In contrast, NEUROLOGIC consistently achieves strong performance in both fidelity and accuracy across all benchmarks. These results demonstrate that NEUROLOGIC strikes a favorable balance by effectively combining interpretability, computational efficiency, and faithfulness, outperforming existing rule-based methods.

C ADDITIONAL DETAILS ON TRANSFORMER-BASED SENTIMENT ANALYSIS

All experiments are conducted on a machine running Ubuntu 22.04 LTS, equipped with an NVIDIA A100 GPU (40 GB VRAM), 85 GB of RAM, and dual Intel Xeon CPUs.

EmoLex. We use the NRC Word-Emotion Association Lexicon (Mohammad & Turney, 2013). Tweets are lower-cased and split into alphabetical word fragments with regex. The tweet is assigned emotion e iff any word appears in the EmoLex list for e. No lemmatisation, emoji handling, or other heuristics are applied.

C.1 GROUNDING RULE TEMPLATES PROCEDURE

Given DNFs extracted in §3.2, we ground each DNF to lexical templates of the underlying text. Our implementation (`causal_word_lexical_batched` does the following:

Implementation We use **spaCy 3.7** (`en_core_web_sm`) for sentence segmentation, POS tags, and dependency arcs. 1) **Causal test.** For every neuron-predicate in the learned DNF, we *mask* one candidate word. If the forward pass flips the DNF class prediction i.e any predicate in the DNF flips, the word is deemed causal. We then fit this word into the possible templates.
2) **Template types.** Once a word is deemed *causal*, we map it to the first matching template in the following order:

1. **IS_HASHTAG**: word starts with "#".
2. **AT_START / AT_END**: word's index within its sentence falls in the first or last 20 % of tokens ($\alpha = 0.20$).

3. **BEFORE/AFTER_SUBJECT**: using `spaCy`, locate the first `nsubj/nsubjpass`; the word is BEFORE or AFTER if it appears within a $\pm$ 6-token window of that subject.
4. **BEFORE/AFTER_VERB**: same window logic around the first main `VERB`.
5. **EXISTS**: general template, applied to all templates.

This assignment yields the (*word*, *template*) pair that forms the grounded rules.
3) **Scoring & ordering.** For every (*word*, *template*) rule, we compute a *support score*

$$s = \mathrm{idf}(w)\,\frac{\texttt{flips}(w,t)}{\texttt{total}(w,t)}, \quad \mathrm{idf}(w) = \log\frac{N_{\mathrm{docs}}+1}{\mathrm{df}(w)+1}.$$

Templates with $s \geq \tau$ ($\tau = 0.03$) are kept. The final rule list for each class is sorted in **descending** s so highest score appears first.

C.2 Top DNF rule accuracy for each class

We report the class-wise accuracy achieved by the top DNF rule at each layer in Figure 3. The results show that each layer's behavior can be effectively and consistently explained by its corresponding top DNF rule, demonstrating a strong alignment between the rule and the model's internal representations.

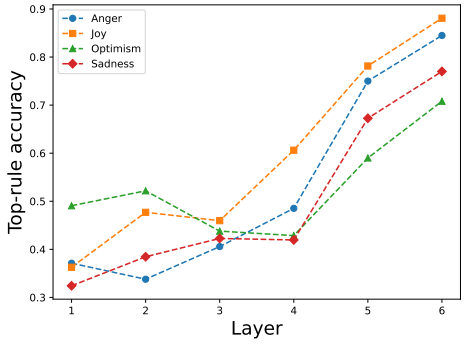


Figure 3: Top DNF rule accuracy for each class by layer.

C.3 Code

All code used for our experiments is available in the following GitHub repository: `github.com/NeuroLogic2026/NeuroLogic`.

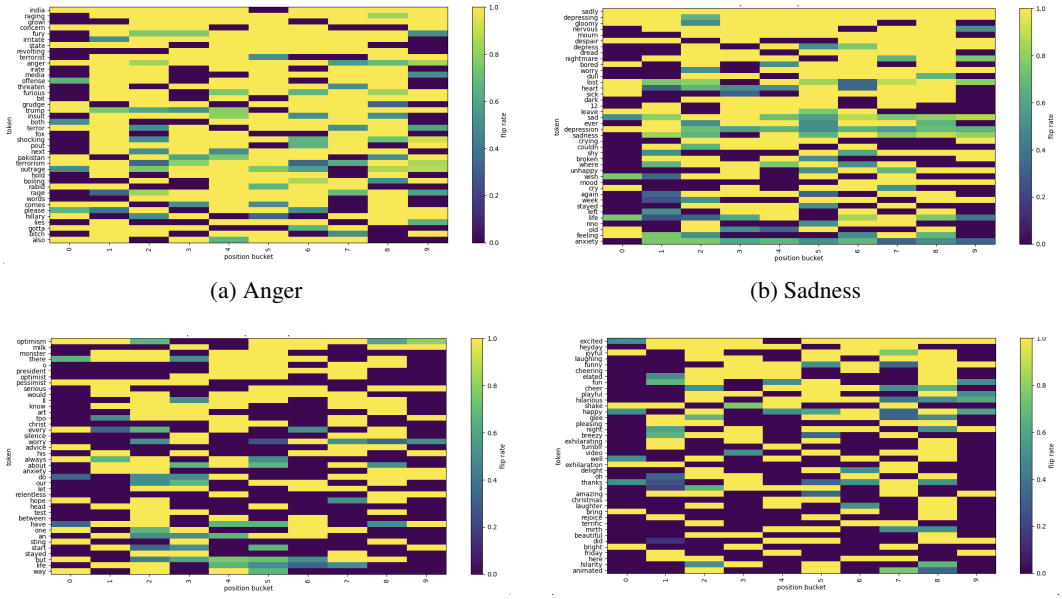
C.4 Token Position Analysis

Figures 4a, 4b, 4c, and 4d present results for the classes *Anger*, *Sadness*, *Optimism*, and *Joy*, respectively. We identify *causal tokens*—words whose masking flips the activation of at least one class-specific predicate neuron. These words are grouped into 10 buckets based on their relative position within the input.

C.5 Compare with Baseline

EmoLex Baseline. We use the NRC Word-Emotion Association Lexicon, EmoLex (Mohammad & Turney, 2013). Tweets are lower-cased and split into alphabetical word fragments with regex. The tweet is assigned emotion e iff any word appears in the EmoLex list for e. No lemmatisation, emoji handling, or other heuristics are applied.

Table 4 compares the top-10 token cues for the class *Sadness* extracted by each method. NeuroLogic's top-10 list preserves core sadness cues like *sad*, *depression*, *sadness*, *depressing*, *sadly*, *depressed*, *mourn*, *anxiety* while promoting unique contextual hits like *nightmare* and *never* in place of more noisy terms like *terrorism* or *feeling*. Concretely, our method lifts the F1 from 0.297 to 0.499 by stripping out noisy cross-class terms without losing coverage.



(a) Anger (b) Sadness

(c) Optimism (d) Joy

Figure 4: Heat maps of keywords by positional bucket for different emotion classes.

	EMOLEX	NeuroLogic
	depression	sad
	bad	depression
	lost	lost
	terrorism	depressing
	sadness	sadness
	awful	sadly
	anxiety	mourn
	depressed	nightmare
	feeling	anxiety
	offended	never
F1	0.297	**0.499**

Table 4: Top-10 words for the *Sadness* class from EMOLEX and NEUROLOGIC. The bottom row reports the F1 scores.

C.6 TOP GROUNDED RULES

Tables 5, 6, 7, 8, 9, 10 present the top five grounded rules, i.e., (*Keyword*, *Template*) pairs, associated with the top five neurons in each corresponding layer, respectively.

Neuron	Class	Purity	Keyword	Template	Flips	Rate
438	optimism	1.2304	is	before_verb	14	1.00
			a	exists	14	1.00
			you	before_verb	32	1.00
			you	at_start	25	1.00
			you	after_subject	22	1.00
683	optimism	1.2237	is	before_verb	17	1.00
			a	exists	13	1.00
			you	before_verb	21	1.00
			you	at_start	18	1.00
			you	after_subject	16	1.00
568	optimism	1.2081	is	before_verb	20	1.00
			a	exists	17	1.00
			you	before_verb	22	1.00
			you	at_start	22	1.00
			you	after_subject	19	1.00
389	anger	1.2037	my	at_end	12	1.00
			it	at_start	74	1.00
			it	exists	55	1.00
			it	before_verb	52	1.00
			it	at_end	43	1.00
757	optimism	1.2006	is	before_verb	15	1.00
			a	exists	13	1.00
			you	before_verb	26	1.00
			you	at_start	23	1.00
			you	after_subject	18	1.00

Table 5: Top-5 grounded rules in layer 1.

Neuron	Class	Purity	Keyword	Template	Flips	Rate
734	anger	1.1921	it	at_start	61	1.00
			it	after_verb	73	1.00
			s	after_subject	62	1.00
			that	after_subject	84	1.00
			that	after_verb	87	1.00
110	anger	1.1875	it	at_start	6	1.00
			it	after_verb	6	1.00
			s	after_subject	5	1.00
			that	after_subject	3	1.00
			that	after_verb	1	1.00
756	anger	1.1739	it	at_start	52	1.00
			it	after_verb	63	1.00
			s	after_subject	61	1.00
			that	after_subject	64	1.00
			that	after_verb	70	1.00
635	anger	1.1722	it	at_start	53	1.00
			it	after_verb	60	1.00
			s	after_subject	65	1.00
			that	after_subject	69	1.00
			that	after_verb	78	1.00
453	anger	1.1628	it	at_start	52	1.00
			it	after_verb	62	1.00
			s	after_subject	64	1.00
			that	after_subject	69	1.00
			that	after_verb	74	1.00

Table 6: Top-5 grounded rules in layer 2.

Neuron	Class	Purity	Keyword	Template	Flips	Rate
597	joy	1.4149	amazing	after_verb	1	1.00
509	anger	1.3988	it	at_end	16	1.00
			that	before_subject	8	1.00
			he	before_verb	23	1.00
			he	at_start	19	1.00
			user	after_subject	16	1.00
495	anger	1.3886	it	at_end	10	1.00
			that	before_subject	14	1.00
			he	before_verb	10	1.00
			he	at_start	12	1.00
			user	after_subject	13	1.00
652	joy	1.3743	amazing	after_verb	28	1.00
			live	after_verb	27	1.00
			ly	before_verb	27	1.00
			ly	at_start	27	1.00
			musically	at_end	27	1.00
734	anger	1.3660	it	at_end	6	1.00
			that	before_subject	2	1.00
			he	before_verb	1	1.00
			he	at_start	1	1.00
			user	after_subject	3	1.00

Table 7: Top-5 grounded rules in layer 3.

Neuron	Class	Purity	Keyword	Template	Flips	Rate
499	joy	1.8198	today	at_start	1	1.00
258	joy	1.7694	heyday	after_verb	1	1.00
698	joy	1.7653	heyday	after_verb	1	1.00
			today	at_start	3	1.00
221	joy	1.7426	today	at_start	1	1.00
535	joy	1.7384	heyday	after_verb	2	1.00
			glee	before_subject	6	1.00
			today	at_start	2	1.00

Table 9: Top-5 grounded rules in layer 5.

Neuron	Class	Purity	Keyword	Template	Flips	Rate
597	joy	1.5896	is	before_verb	1	1.00
232	joy	1.5464	is	before_verb	4	1.00
			amazing	after_verb	1	1.00
			i	after_subject	6	0.96
			i	after_verb	8	0.96
			this	after_verb	5	0.96
66	joy	1.5405	is	before_verb	2	1.00
			this	after_verb	1	0.96
399	anger	1.5323	s	exists	2	1.00
			fucking	after_subject	4	1.00
			but	before_subject	1	1.00
			but	at_start	1	1.00
			but	before_verb	1	1.00
71	joy	1.5262	is	before_verb	8	1.00
			amazing	after_verb	1	1.00
			i	after_subject	20	0.96
			i	after_verb	17	0.96
			this	after_verb	9	0.96

Table 8: Top-5 grounded rules in layer 4.

Neuron	Class	Purity	Keyword	Template	Flips	Rate
122	joy	1.8478	laughter	after_subject	2	1.00
			hilarious	after_subject	1	0.88
344	joy	1.8359	playful	exists	1	1.00
			smiling	after_subject	2	1.00
			laughter	at_end	1	1.00
			laughter	after_subject	1	1.00
			hilarious	after_subject	2	0.88
497	joy	1.8342	laughter	after_subject	1	1.00
			hilarious	after_subject	1	0.88
212	joy	1.8330	playful	exists	1	1.00
			omg	before_subject	1	1.00
			smiling	after_subject	2	1.00
			laughter	at_end	1	1.00
			laughter	after_subject	3	1.00
452	joy	1.8261	laughter	after_subject	1	1.00
			hilarious	after_subject	1	0.88

Table 10: Top-5 grounded rules in layer 6.

C.7 TOP THREE RULES PER CLASS FOR EVERY LAYER

Tables 11 and 12 present the top three grounded rules for each class across all layers.

Class	Keyword	Template	Flips	Total	Rate	Neuron
Layer 1						
anger	terrorism	after_subject	19	19	1.00	125
anger	am	at_start	19	19	1.00	125
anger	terrorism	after_subject	17	19	1.00	695
joy	ly	before_verb	27	27	1.00	505
joy	ly	at_start	27	27	1.00	505
joy	musically	at_end	27	27	1.00	505
optimism	can	after_subject	17	19	1.00	563
optimism	your	after_verb	18	21	1.00	438
optimism	your	after_verb	18	21	1.00	563
sadness	want	after_subject	17	19	1.00	52
sadness	lost	after_subject	24	30	1.00	52
sadness	want	after_subject	16	19	1.00	679
Layer 2						
anger	have	after_subject	67	86	1.00	298
anger	with	after_verb	67	84	1.00	298
anger	have	after_subject	65	86	1.00	136
sadness	my	after_verb	83	94	1.00	698
sadness	is	after_subject	96	110	1.00	712
sadness	my	after_verb	72	94	1.00	712
Layer 3						
anger	people	before_verb	27	32	1.00	189
anger	why	before_verb	23	30	1.00	189
anger	why	before_subject	23	31	1.00	189
joy	ly	before_verb	27	27	1.00	652
joy	ly	before_verb	27	27	1.00	28
joy	ly	at_start	27	27	1.00	652
optimism	be	after_subject	22	26	1.00	459
optimism	be	after_subject	22	26	1.00	416
optimism	be	after_subject	19	26	1.00	157
sadness	sad	at_end	25	27	1.00	316
sadness	sad	at_end	24	27	1.00	498
sadness	sad	after_verb	23	26	1.00	305

Table 11: Top-3 grounded rules per class (Layers 1 to 3).

Class	Token	Keyword	Template	Total	Rate	Neuron
Layer 4						
anger	fucking	after_subject	20	20	1.00	434
anger	anger	after_verb	18	19	1.00	92
anger	terrorism	after_subject	17	19	1.00	92
joy	amazing	after_verb	29	31	1.00	95
joy	this	after_verb	41	48	0.96	95
joy	is	before_verb	19	25	1.00	95
optimism	be	after_subject	22	26	1.00	416
optimism	you	at_start	26	34	0.97	416
optimism	you	before_verb	29	38	1.00	416
sadness	sad	at_end	19	27	1.00	305
sadness	sad	at_end	19	27	1.00	23
sadness	sad	at_end	17	27	1.00	296
Layer 5						
anger	awful	at_end	15	19	0.84	92
anger	angry	after_subject	22	27	0.93	92
anger	angry	after_verb	21	27	0.89	92
optimism	it	at_start	9	24	0.83	430
optimism	it	before_verb	6	21	0.81	430
optimism	is	at_start	8	29	0.76	459
sadness	sad	at_end	23	27	0.93	246
sadness	sadness	exists	17	23	0.83	433
sadness	sad	at_end	22	27	0.93	433
Layer 6						
anger	anger	after_verb	14	19	0.89	531
anger	awful	at_end	13	19	0.74	15
anger	anger	after_verb	13	19	0.89	603
optimism	not	after_subject	14	19	0.74	142
optimism	s	after_subject	13	20	0.75	142
optimism	user	exists	15	20	0.80	142
sadness	sadness	exists	17	23	0.78	298
sadness	sad	exists	25	31	0.81	298
sadness	sad	at_end	21	27	0.89	242

Table 12: Top-3 grounded rules per class (Layers 4 to 6).