doi:10.1017/S1471068425100318

Symbolic Rule Extraction From Attention-Guided Sparse Representations in Vision Transformers[*]

PARTH PADALKAR and GOPAL GUPTA

The University of Texas at Dallas, USA

(*e-mails:* parth.padalkar@utdallas.edu, gupta@utdallas.edu)

submitted 24 July 2025; revised 24 July 2025; accepted 27 July 2025

Abstract

Recentneuro-symbolic approaches have successfully extracted symbolic rule-sets from Convolutional Neural Network-based models to enhance interpretability. However, applying similar techniques to Vision Transformers (ViTs) remains challenging due to their lack of modular concept detectors and reliance on global self-attention mechanisms. We propose a framework for symbolic rule extraction from ViTs by introducing a sparse concept layer inspired by Sparse Autoencoders (SAEs). This linear layer operates on attention-weighted patch representations and learns a disentangled, binarized representation in which individual neurons activate for high-level visual concepts. To encourage interpretability, we apply a combination of L1 sparsity, entropy minimization, and supervised contrastive loss. These binarized concept activations are used as input to the FOLD-SE-M algorithm, which generates a rule-set in the form of a logic program. Our method achieves a **5.14 %** better classification accuracy than the standard ViT while enabling symbolic reasoning. Crucially, the extracted rule-set is not merely post-hoc but acts as a logic-based decision layer that operates directly on the sparse concept representations. The resulting programs are concise and semantically meaningful. This work is the first to extract executable logic programs from ViTs using sparse symbolic representations, providing a step forward in interpretable and verifiable neuro-symbolic AI.

KEYWORDS: neuro-symbolic AI, mechanistic interpretability, Vision Transformers, explainable AI (XAI), rule-based machine learning, Sparse Autoencoders

1 Introduction

Extracting logic-based rules from neural models has emerged as a central objective in neuro-symbolic AI, driven by the growing demand for interpretability and verifiability in machine learning. As deep learning models continue to scale, they are increasingly deployed in critical applications such as autonomous driving (Kanagaraj *et al.* (2021)), medical diagnosis (Sun *et al.* (2016)), and natural disaster prevention (Ko and Kwak (2012)). In these sensitive domains, incorrect predictions can carry severe

[*] This work was supported by US NSF Grant IIS 1910131, US DoD and industry grants. We thank the members of the ALPS lab at UT Dallas for insightful discussions. We are grateful to the anonymous reviewers for their insightful comments and suggestions for improvement.

consequences, emphasizing the necessity for transparency in decision-making. Many of these applications depend significantly on accurate image classification models, particularly Convolutional Neural Networks (CNNs). Recent frameworks such as NeSyFOLD (Padalkar *et al.* (2024a, b)) and ERIC (Townsend *et al.* (2021)) have successfully demonstrated the extraction of human-interpretable symbolic rule sets from CNNs, providing insights into the underlying reasoning of predictions in vision tasks.

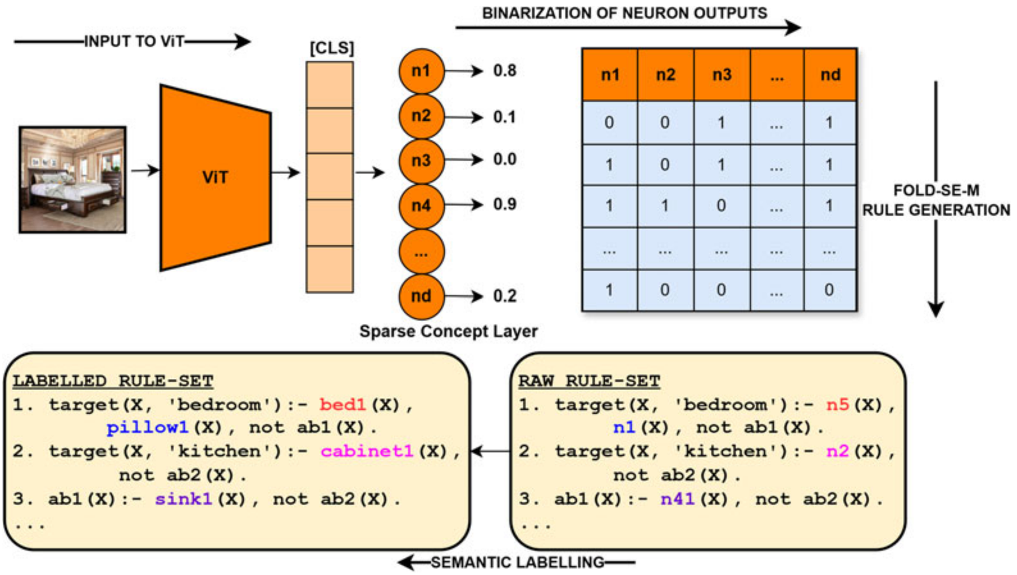
Most of this progress, however, has been concentrated on CNNs. CNNs are composed of filters, that are trainable matrices that learn to detect patterns in local regions of images. Their modular architecture, where individual filters often correspond to distinct visual concepts, makes them particularly suitable for rule extraction. By binarizing the activations of the final layer and using them as input to rule-based machine learning algorithms such as decision trees or FOLD-SE-M (Wang and Gupta (2024)), it is possible to extract the symbolic rule-sets.

In contrast, Vision Transformers (ViTs) (Dosovitskiy *et al.* (2021)) have remained largely inaccessible to symbolic extraction techniques. While ViTs now dominate the field of vision due to their superior performance and flexibility, their reliance on global self-attention and lack of explicit concept detectors pose a significant challenge for rule-based interpretability. ViTs encode information in a distributed and entangled manner, making it unclear how to localize or discretize concept-level representations.

In this work, we take a step toward bridging this gap by introducing a framework – NeSyViT – for extracting logic-based rule sets from ViTs. We modify the standard ViT architecture by replacing the final fully connected classification head with a single linear layer–*sparse concept layer*–trained to produce binarized outputs. The goal is to encourage each neuron in this final layer to correspond to a few distinct high-level concepts. To achieve this, we draw inspiration from Sparse Autoencoders (SAEs) (Ng *et al.* (2011)), incorporating an L1 sparsity loss to ensure that only a small subset of neurons activates for any given image.

To encourage binarization, we apply a sigmoid activation to the linear outputs, constraining them to the $[0, 1]$ range, and introduce an entropy-based loss that pushes activations toward binary extremes (0 or 1). This design enables us, after training, to extract a binary vector for each image that reflects concept-level activations. These vectors are then passed to the FOLD-SE-M rule-based machine learning algorithm Wang and Gupta (2024) to generate a symbolic rule-set in the form of a stratified Answer Set Program for classification. It is crucial that the representations corresponding to images from the same class are well-clustered in the latent space. This promotes the formation of clear decision boundaries that can be effectively exploited by FOLD-SE-M. To enforce this structure, we incorporate a supervised contrastive loss (SupCon) (Khosla *et al.* (2020)), which encourages representations of images with the same label to lie closer together while pushing apart those from different classes. This facilitates learning highly accurate rules.

Finally, each predicate in the extracted rule-set is “semantically labelled,” that is each predicate in the rule-set that corresponds to a neuron is matched to the concepts that the neuron represents. Padalkar et al., introduced an algorithm for automatic semantic labelling of the predicates in the rule-sets extracted from a CNN using FOLD-SE-M



n1	n2	n3	...	nd
0	0	1	...	1
1	0	1	...	1
1	1	0	...	1
...	...	...	...	...
1	0	0	...	0

```
LABELLED RULE-SET
1. target(X, 'bedroom'):- bed1(X),
        pillow1(X), not ab1(X).
2. target(X, 'kitchen'):- cabinet1(X),
        not ab2(X).
3. ab1(X):- sink1(X), not ab2(X).
...
```

```
RAW RULE-SET
1. target(X, 'bedroom'):- n5(X),
        n1(X), not ab1(X).
2. target(X, 'kitchen'):- n2(X),
        not ab2(X).
3. ab1(X):- n41(X), not ab2(X).
...
```

Fig 1. The NeSyViT framework.

Padalkar *et al.* (2024a). We adopt the same algorithm for semantic labelling of the rule-sets we generate. Figure 1 illustrates the NeSyViT framework.

The final Neuro-Symbolic (NeSy) model is a combination of the ViT based feature extractor that generates the binary vector for each image and the rule-set generated by FOLD-SE-M, which classifies the image into a particular class based on which neurons were activated/deactivated (1/0). We demonstrate through our experiments that the NeSy model produced by our framework outperforms the vanilla ViT in terms of classification accuracy, achieving an average improvement of **5.5%**. Notably, this performance gain is achieved while also generating interpretable rule-sets – a significant result, as prior neuro-symbolic frameworks that extract rule-sets from neural models often suffer a drop in accuracy. Our contributions are as follows:

1. We introduce a training method that combines supervised contrastive loss, L1 sparsity, and entropy loss to learn compact and binarized concept-level representations.
2. We propose an end-to-end neuro-symbolic framework, *NeSyViT*, for generating a rule-set from a modified ViT using FOLD-SE-M.
3. We show through experiments that our NeSyViT framework achieves classification accuracy better than the vanilla ViT, while also producing concise rule-sets

2 Background

2.1 Vision Transformers

The ViT (Dosovitskiy *et al.* (2021)) is a deep learning architecture that applies the Transformer model, originally developed for natural language processing, to image classification tasks. Unlike CNNs, which operate on local receptive fields using convolutional filters, ViTs treat an image as a sequence of tokens derived from fixed-size non-overlapping image patches. Specifically, an input image is divided into equal-sized patches (e.g., 16×16 pixels), and each patch is flattened into a vector and linearly projected into a higher-dimensional embedding space.

At the core of these encoder blocks lies the multi-head self-attention mechanism (Vaswani *et al.* (2017)), which enables the model to learn long-range dependencies and contextual relationships between all image patches. Each self-attention layer allows every patch to interact with every other patch, thereby capturing global contextual information at every level of the model. This contrasts with CNNs, where information is typically propagated through hierarchical layers with limited receptive fields.

The self-attention mechanism operates by computing three learned vectors for each patch token: a query (Q), a key (K), and a value (V). The attention score between a given pair of tokens is calculated by taking the dot product of their corresponding query and key vectors. These scores are then scaled and normalized using the softmax function, yielding attention weights that indicate the importance of each patch relative to every other patch in the sequence. The final representation of each patch is obtained as a weighted sum of the value vectors, with the weights determined by the attention scores.

This mechanism allows the ViT to dynamically focus on the most relevant parts of the image depending on the task, thereby enabling rich and flexible modeling of spatial dependencies. Thus, the self-attention layers not only encode contextual information for each patch but also play a central role in determining which visual features contribute most significantly to the model's decision. A special learnable token, known as the `[CLS]` token, is prepended to the patch sequence before being passed into the Transformer. Unlike the patch tokens that represent image regions, the `[CLS]` token acts as a summary placeholder. During self-attention, it aggregates information from all other patches by attending to them across multiple layers. After the final Transformer layer, the embedding corresponding to the `[CLS]` token is typically used for classification, as it encapsulates a global representation of the image.

While this architecture allows ViTs to capture long-range dependencies and global structure, it also presents a major challenge for interpretability. In CNNs, individual filters often learn to detect localized visual patterns or objects (e.g., corners, textures, faces), enabling a degree of modularity and conceptual traceability. In contrast, attention heads are distributed, context-sensitive, and dynamically influenced by the full set of tokens – including the `[CLS]` token. This makes it difficult to associate specific heads or tokens with interpretable visual concepts. Consequently, extracting symbolic, modular representations from ViTs is significantly more challenging than from CNNs. Hence, in this work, we take the `[CLS]` vector and pass it through the sparse concept layer that is optimized to produce sparse outputs, allowing the information encoded in the `[CLS]` token to be disentangled into a small set of active neurons. Each neuron is encouraged to specialize and activate primarily for images of a specific class. This design is inspired by the core principles of Sparse Autoencoders (SAEs) (Ng *et al.* (2011)), which promote compact and interpretable representations through sparsity.

2.2 FOLD-SE-M

The FOLD-SE-M algorithm Wang and Gupta (2024) that we employ in our framework, learns a rule-set from data as a *default theory*. Default logic is a non-monotonic logic used to formalize commonsense reasoning. A default D is expressed as:

$$D = \frac{A : \mathbf{M}B}{\Gamma} \tag{1}$$

Equation 1 states that the conclusion Γ can be inferred if pre-requisite A holds and B is justified. $\mathbf{M}B$ stands for "it is consistent to believe B." Normal logic programs can encode a default theory quite elegantly (Gelfond and Kahl (2014)). A default of the form:

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n : \mathbf{M}\neg\beta_1, \mathbf{M}\neg\beta_2 \ldots \mathbf{M}\neg\beta_m}{\gamma}$$

can be formalized as the normal logic programming rule:

$$\gamma \texttt{ :- } \alpha_1, \alpha_2, \ldots, \alpha_n, \texttt{not}\ \beta_1, \texttt{not}\ \beta_2, \ldots, \texttt{not}\ \beta_m.$$

where α's and β's are positive predicates and `not` represents negation-as-failure. We call such rules *default rules*. Thus, the default

$$\frac{bird(X) : M\neg penguin(X)}{flies(X)}$$

will be represented as the following default rule in normal logic programming:

```
flies(X) :- bird(X), not penguin(X).
```

We call `bird(X)`, the condition that allows us to jump to the default conclusion that `X` flies, the *default part* of the rule, and `not penguin(X)` the *exception part* of the rule.

FOLD-SE-M (Wang and Gupta (2024)) is a Rule Based Machine Learning algorithm. It generates a rule-set from tabular data, comprising rules in the form described above. The complete rule-set can be viewed as a stratified answer set program (a stratified Answer Set Programming (ASP) rule-set has no cycles through negation (Baral (2003))). It uses special `abx` predicates to represent the exception part of a rule where `x` is a unique numerical identifier. FOLD-SE-M incrementally generates literals for *default rules* that cover positive examples while avoiding covering negative examples. It then swaps the positive and negative examples and calls itself recursively to learn exceptions to the default when there are still negative examples falsely covered.

FOLD-SE-M has been shown to produce more compact rule-sets than even decision tree algorithms, while matching or surpassing them in classification accuracy (Wang and Gupta (2024)). It has also demonstrated competitive performance compared to state-of-the-art models such as XGBoost and Multi-Layer Perceptrons. The succinctness of the rule-sets generated by FOLD-SE-M stems from its decision-list-like representation, which imposes a strict top-down execution order. This means that the rule interpreter sequentially evaluates each rule – starting from the top – and stops as soon as a rule fires, reducing redundancy and improving interpretability. This structure is particularly advantageous because the resulting rule-sets can be directly used as input to the s(CASP) goal-directed ASP interpreter, which inherently follows a top-down execution strategy. As a result, FOLD-SE-M not only produces interpretable and compact models but also serves as a bridge between tabular data and symbolic knowledge representation within a powerful formalism like ASP. However, applying FOLD-SE-M directly to image data is not straightforward due to its original design for tabular inputs. In this work, we take a significant step beyond prior efforts such as NeSyFOLD by enabling image classification using ViTs within the symbolic domain. We achieve this by transforming image representations into sparse binary vectors that can be interpreted by FOLD-SE-M, thereby generating ASP rule-sets. Notably, FOLD-SE-M consistently produces some of the most succinct rule-sets among symbolic learners, which directly enhances interpretability in this neurosymbolic pipeline.

There are 2 tunable hyperparameters, *ratio*, and *tail*. The *ratio* controls the upper bound on the number of false positives to the number of true positives implied by the default part of a rule. The *tail* controls the limit of the minimum number of training examples a rule can cover.

3 Methodology

Our aim is to extract a symbolic rule-set from the ViT and use it for final classification. To this end, we first modify the architecture of a standard ViT by replacing the final classification head – placed after the last self-attention layer– with a single linear layer of dimension D. The `[CLS]` token's vector which has the aggregated global information feeds directly into this layer. This layer outputs a D-dimensional vector, to which we apply a sigmoid function element-wise, constraining all values to lie in the range $[0, 1]$. Each value in this vector could then be interpreted as the activation of a distinct neuron. There are three key objectives in designing this representation:

1. The model needs to produce similar binary vectors for images of the same class, and dissimilar ones for different classes, to allow the FOLD-SE-M rule-based machine learning alogrithm to identify meaningful decision boundaries.
2. The output values had to be pushed as close as possible to either 0 or 1, since after training, we binarize these vectors by rounding. Values near 0.5 would introduce ambiguity and loss of information.
3. The output vectors need to be sparse, with most values being 0 and only a few being 1. This sparsity ensures that rule-sets remain compact and interpretable, as fewer neurons would appear as predicates in the learned rules.

We enforce these three properties using a combination of supervised contrastive loss, entropy minimization, and L1 sparsity loss, described in detail below.

Supervised Contrastive Loss: To ensure that images from the same class produce similar sparse representations, we incorporate the *Supervised Contrastive Loss* (SupCon) (Khosla *et al.* (2020)). This loss encourages the D-dimensional vectors corresponding to samples of the same class to cluster together in the latent space, while pushing apart those from different classes. Such class-wise clustering is essential for enabling a rule-based learner, such as FOLD-SE-M, to identify clean symbolic decision boundaries.

Let $\mathbf{z}_i \in \mathbb{R}^D$ denote the normalized sparse representation of sample i in a batch $\mathcal{B}$, and let y_i be its corresponding class label. In contrastive learning, we refer to $\mathbf{z}_i$ as the *anchor*, and we compare it to the remaining samples in the batch. Those with the same label are treated as *positives*, while the rest are considered *negatives*.

The SupCon loss for a single anchor i is defined as:

$$\mathcal{L}^{i}_{\text{supcon}} = -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{a \in \mathcal{A}(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \tag{2}$$

where:

- $P(i)$ is the set of indices in the batch with the same label as i (i.e., the positives for anchor i),
- $\mathcal{A}(i)$ is the set of all indices in the batch excluding i itself,
- $\tau > 0$ is a temperature parameter that controls the sharpness of the distribution.

The total loss is averaged over all anchors in the batch:

$$\mathcal{L}_{\text{supcon}} = \frac{1}{N} \sum_{i \in \mathcal{B}} \mathcal{L}^{i}_{\text{supcon}} \tag{3}$$

where N is the number of images in the batch $\mathcal{B}$. By minimizing this loss, the model is encouraged to produce sparse vectors that are tightly clustered for images of the same class and distinct from those of other classes. This structure makes the representations well-suited for interpretable rule extraction.

Entropy Minimization: To encourage the sparse representations to become binarized – that is close to either 0 or 1 – we apply an *entropy minimization* loss on the output of the sigmoid-activated linear layer. Since this linear layer uses a sigmoid function, each neuron's activation lies in the range $[0, 1]$. Ideally, we want these activations to converge toward discrete values (0 or 1) to allow lossless binarization post-training.

We treat each neuron's output as a Bernoulli random variable and compute its entropy using the standard binary entropy formula. Given a batch of N images where $\mathbf{z}_i \in \mathbb{R}^D$ is the D-dimensional output for image i, the entropy loss is computed as:

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{ND} \sum_{i=1}^{N} \sum_{j=1}^{D} \left[z_{i,j} \log(z_{i,j} + \epsilon) + (1 - z_{i,j}) \log(1 - z_{i,j} + \epsilon) \right], \tag{4}$$

where $z_{i,j}$ is the activation of neuron j for image i, and ϵ is a small constant added for numerical stability.

Minimizing this entropy term encourages each activation to move closer to either 0 or 1, making the final binarization step more reliable. This is especially important for downstream symbolic rule extraction, where crisp binary features are essential for learning accurate and interpretable logic programs.

L1 Sparsity Loss: To promote interpretability and ensure that only a few neurons activate for each image, we incorporate an *L1 sparsity loss* on the output of the final linear layer. Sparse representations are crucial for generating concise rule-sets, as they limit the number of active predicates per image, reducing the complexity of the learned logic programs.

Formally, given a batch of N vectors $\mathbf{z}_1, \ldots, \mathbf{z}_N$, where $\mathbf{z}_i \in \mathbb{R}^D$ is the sigmoid-activated output for image i, the L1 sparsity loss is defined as:

$$\mathcal{L}_{\text{sparsity}} = \frac{1}{ND} \sum_{i=1}^{N} \sum_{j=1}^{D} |z_{i,j}| \tag{5}$$

This loss penalizes large activation values and encourages most neurons to remain close to zero, allowing only a few dimensions to be active for any given input.

The total loss then becomes:

$$\mathcal{L}_{Total} = \alpha \mathcal{L}_{supcon} + \beta \mathcal{L}_{entropy} + \gamma \mathcal{L}_{sparsity} \tag{6}$$

where α, β and γ are weights to control the impact of each loss term. Finally, after training the model using the combined loss $\mathcal{L}_{\text{Total}}$, we collect the sparse representation vectors for all images in the training set. These vectors are binarized – each value thresholded at 0.5 – to form a binary table, where each row corresponds to an image and each column to a neuron in the final linear layer. This binarization table is used as input to the FOLD-SE-M algorithm, which generates a symbolic rule-set in the form of a stratified Answer Set Program. Each predicate in the resulting rule-set corresponds directly to a neuron in the final layer, enabling symbolic reasoning over the learned representations.

Semantic Labelling Algorithm: To assign human-understandable meanings to these predicates, we employ the semantic labeling algorithm introduced by Padalkar et al., (Padalkar et al. (2024a)). For each neuron, we identify the top-10 images that activate it most strongly. We then compute attention-based heatmaps for these images to localize the spatial regions that the neuron focuses on. These heatmaps are intersected with the corresponding pre-annotated semantic segmentation masks. A semantic segmentation mask is a pixel-level annotation of an image where each individual pixel is assigned a specific semantic category label, such as bed, refrigerator, or sink. These masks provide a fine-grained understanding of the objects and regions present in the image. By overlapping the neuron's activation heatmap with these masks, we can accurately measure how much a particular neuron responds to specific visual concepts, based on the spatial alignment between the activated regions and the labeled object areas in the image. For each neuron, we compute the average Intersection-over-Union (IoU) between its heatmaps and each available concept label. The label with the highest average IoU is then assigned to the predicate corresponding to that neuron, thus grounding the symbolic rule-set in interpretable visual concepts.

Using the rule-set for image classification: After training, each new image is classified using the learned symbolic rule-set in the following way. First, the image is passed through the modified ViT, which outputs a D-dimensional latent vector from the sparse concept layer. Each of the D dimensions corresponds to a specific neuron, which may be associated with a visual concept. This vector is then binarized by applying a threshold of 0.5 to each dimension – values greater than 0.5 are treated as `True`, and others as `False`. This binary vector effectively represents the presence or absence of the learned concepts in the image.

The binarized vector is then input to the FOLD-SE-M rule interpreter. The symbolic rule-set consists of logic rules of the form `target(X, ‘class’) :- predicate1(X), not predicate2(X), ...`. Each predicate in the rule corresponds to a specific neuron (or concept) in the sparse vector. During inference, FOLD-SE-M checks each rule sequentially, determining whether its body is satisfied based on the current binary vector. As soon as a rule evaluates to true, its head – containing the predicted class – is returned as the final prediction.

This symbolic classification process is not only accurate but also interpretable, as one can trace exactly which predicates (neurons) triggered the classification. Furthermore, the resulting rule-set can be fed into the s(CASP) (Arias *et al.* (2018)) goal-directed ASP interpreter, which provides justifications or explanations in the form of symbolic derivations for the predicted label. This end-to-end approach tightly integrates deep visual features with symbolic reasoning, enabling verifiable and explainable image classification.

4 Experiments

We conducted experiments to address the following research questions:

Q1: How does the classification accuracy of the neuro-symbolic model produced by the NeSyViT framework compare to that of a standard vanilla ViT?

Q2: What is the typical size of the rule-set generated by the NeSyViT framework, and how compact are the resulting rules?

Q3: How does our framework scale as the number of classes increases?

Q4: What do the neuron activation patterns look like for each class, and how well do they align with human-interpretable semantic concepts?

Q5: How well can the semantic labelling algorithm meant for CNN-based frameworks such as NeSyFOLD, be directly adopted for our NeSyViT framework?

[Q1, Q2, Q3] Classification Accuracy, Rule-set Size and Scalability The central goal of the NeSyViT framework is to produce a neuro-symbolic model that maintains high classification accuracy while offering symbolic interpretability through a compact rule-set. Ideally, the accuracy of the interpretable model should match or exceed that of its purely neural counterpart (i.e., the vanilla ViT), and the resulting rule-set should remain as concise as possible. We use rule-set size as a proxy for interpretability, based on the findings of Lage et al. (2019), who demonstrated through human-subject evaluations that larger rule-sets are significantly more difficult to interpret. Thus, achieving high accuracy alongside a small rule-set is critical for balancing performance and human-understandability.

Setup: We use the ViT-Base architecture from the `timm` library, which processes input images of size 224 × 224 using non-overlapping 16 × 16 patches. The model consists of 12 Transformer blocks, each with 12 attention heads, and produces a `[CLS]` token embedding of dimension 768. As described earlier, we modify this architecture by removing the final classification head and replacing it with a single linear layer of output dimension 128.

Table 1. *Comparison between the NeSyViT and the Vanilla ViT. Bold values are better*

Data	Model	Accuracy (%)	Rules	Unique Predicates	Size
P2	NeSyViT	**100 ± 0.00**	2 ± 0.00	2 ± 0.00	2 ± 0.00
	Vanilla	99 ± 0.00	-	-	-
P3.1	NeSyViT	**99 ± 0.00**	3 ± 0.00	3 ± 0.00	3 ± 0.00
	Vanilla	98 ± 0.00	-	-	-
P3.2	NeSyViT	**99 ± 0.00**	3 ± 0.00	3 ± 0.40	3 ± 0.00
	Vanilla	97 ± 0.00	-	-	-
P3.3	NeSyViT	**98 ± 0.00**	5 ± 0.40	3 ± 0.49	5 ± 0.40
	Vanilla	91 ± 0.00	-	-	-
P5	NeSyViT	**98 ± 0.00**	5 ± 0.00	5 ± 0.40	5 ± 0.80
	Vanilla	90 ± 0.00	-	-	-
P10	NeSyViT	**94 ± 0.00**	10 ± 0.49	12 ± 1.47	14 ± 1.67
	Vanilla	76 ± 0.00	-	-	-
GT43	NeSyViT	98 ± 0.00	43 ± 0.00	44 ± 1.90	51 ± 3.12
	Vanilla	**99 ± 0.00**	-	-	-
Mean Stats	NeSyViT	**98 ± 0.00**	10.14 ± 0.13	10.30 ± 0.67	11.85 ± 0.86
	Vanilla	92.86 ± 0.00	-	-	-

All experiments are conducted using this configuration with weights pretrained on ImageNet-1k to construct the NeSy model. The vanilla ViT refers to the same base architecture without any modifications. We used the AdamW optimizer along with a cosine annealing learning rate scheduler. The model was trained for 50 epochs using the combined loss consisting of supervised contrastive loss, entropy minimization, and L1 sparsity, as previously described. For rule-set generation, we employed the FOLD-SE-M algorithm. The weights assigned to each loss component, as well as additional training and architectural hyperparameters, are provided in the Appendix.

We evaluated both our NeSy model and the vanilla ViT on subsets of two benchmark datasets (1): the *Places* dataset (Zhou et al., 2018), which contains images of various indoor and outdoor scenes, and the *German Traffic Sign Recognition Benchmark* (GTSRB) (Stallkamp et al., 2012), which consists of images of traffic signposts. From the *Places* dataset, we constructed multiple class subsets of increasing number of classes to gauge the scalability of NeSyViT: two classes (*P2*), three classes (*P3.1*), five classes (*P5*), and ten classes (*P10*). The *P2* subset includes *bathroom* and *bedroom* images. *P3.1* extends this by adding *kitchen*, while *P5* further incorporates *dining room* and *living room*. *P10* adds five additional classes: *home office*, *office*, *waiting room*, *conference room*, and *hotel room*. Additionally, we include two alternative three-class subsets: *P3.2* with *desert road*, *forest road*, and *street*, and *P3.3* with *desert road*, *driveway*, and *highway*. For each class, we sampled $5,000$ images, creating a 4k/1k train-test split, and used the official validation set without modification.

The *GTSRB* (*GT43*) dataset contains 43 traffic sign classes. We used the official test set of $12.6k$ images and performed an $80/20$ train-validation split on the remaining data, resulting in approximately $21k$ training and $5k$ validation images.

https://doi.org/10.1017/S1471068425100318

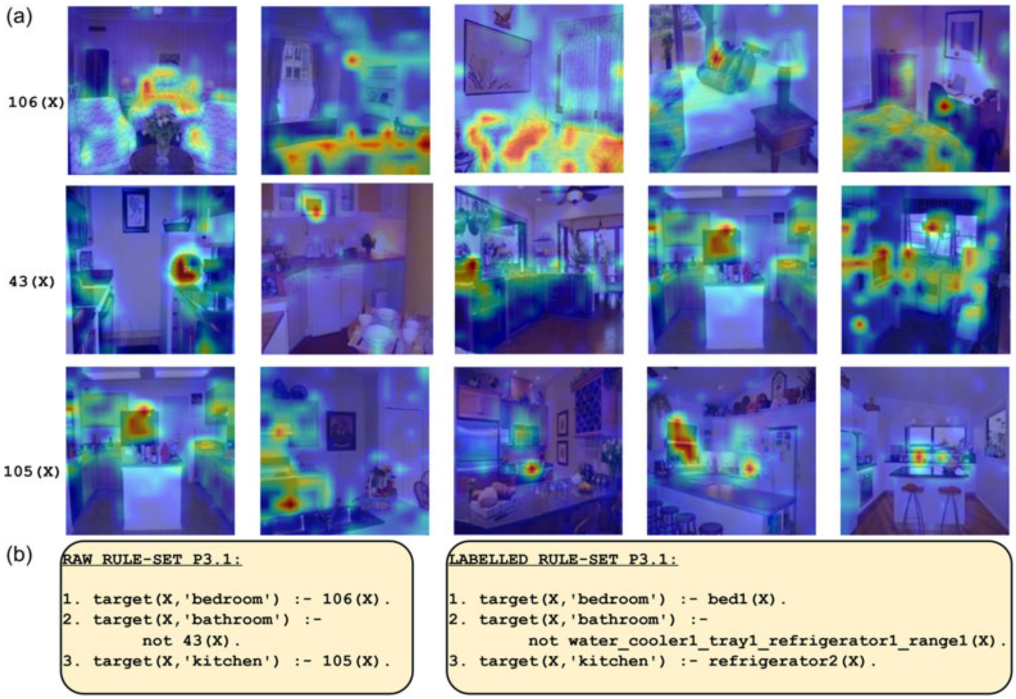
Table 2. *Comparison of relative % accuracy change w.r.t. the vanilla model and rule-set size between NeSyFOLD and NeSyViT*

Data	% Accuracy Change		Rule-set Size	
	NeSyFOLD	NeSyViT	NeSyFOLD	NeSyViT
P2	−4	**+1**	12	**2**
P3.1	−7	**+1**	16	**3**
P3.2	−4	**+2**	7	**3**
P3.3	−7	**+7**	23	**5**
P5	−15	**+8**	30	**5**
P10	−21	**+18**	65	**14**
GT43	−13	**−1**	99	**51**
Mean Stats	−10.14	**+5.14**	36	**11.85**

We run each experiment 5 times with random train-test splits and then report the average metrics in Table 1. We closely follow the experimental setup proposed by Padalkar et al. (2024a) for evaluating the NeSyFOLD framework, which uses FOLD-SE-M to extract rule-sets from CNNs. Specifically, we compare the drop (or gain) in accuracy observed while using NeSyFOLD as compared to the vanilla CNN, against the corresponding change in accuracy when using NeSyViT as compared to the Vanilla ViT. This relative comparison highlights the effectiveness of our approach and is summarized in Table 2.

Result: Table 1 presents a comparison between the NeSy model generated using NeSyViT and the Vanilla ViT. All metrics are reported over five independent runs. Accuracy is computed on the test set for both models. The “Rules” column reports the average and standard deviation of the number of rules generated. “Unique Predicates” indicates the number of distinct predicates used across the rule-set, and “Size” refers to the total number of predicate occurrences in the bodies of all rules. Recall that we use rule-set size as a proxy for interpretability – smaller rule-sets are generally easier to understand.

The most significant observation is that our method achieves higher average accuracy than the Vanilla ViT. This is particularly noteworthy, as extracting interpretable rule-sets by binarizing internal representations typically results in some loss of information and a subsequent drop in performance. However, our approach mitigates this issue through the use of an entropy loss, which encourages the outputs of the sparse concept layer to be close to either 0 or 1 during training. As a result, thresholding these outputs at 0.5 post-training introduces minimal distortion. Furthermore, the supervised contrastive loss ensures that the resulting binary vectors are well-clustered by class, enabling FOLD-SE-M to learn accurate and class-discriminative rule-sets. Finally, the L1 sparsity loss encourages only a small subset of neurons to activate for each image, resulting in compact and interpretable rule-sets. Notice that for the *P10* dataset, the accuracy improvement over vanilla is **18%**, which is a huge improvement and shows the merit of our approach when scaling to larger number of classes.



```
RAW RULE-SET P3.1:

1. target(X,'bedroom') :- 106(X).
2. target(X,'bathroom') :-
        not 43(X).
3. target(X,'kitchen') :- 105(X).
```

```
LABELLED RULE-SET P3.1:

1. target(X,'bedroom') :- bed1(X).
2. target(X,'bathroom') :-
        not water_cooler1_tray1_refrigerator1_range1(X).
3. target(X,'kitchen') :- refrigerator2(X).
```

Fig 2. **a)** The top-5 images overlayed with activation heat-maps for neurons `106, 43` and `105` when rules are extracted for the *P3.1* dataset containing classes "bathroom," "bedroom" and "kitchen." **b)** The raw rule-set and the labelled rule-set when NeSyViT is employed on the *P3.1* dataset.

To provide context for the significance of our results, Table 2 presents the percentage change in accuracy of the NeSy model relative to its vanilla counterpart for both NeSyViT and NeSyFOLD (which uses a CNN instead of a ViT). We also report the average rule-set size generated by each framework. Notably, NeSyViT improves upon the accuracy of the Vanilla ViT by an average of **5.14%** – a substantial gain, especially when contrasted with NeSyFOLD, which suffers an average accuracy drop of **10.14%** compared to the vanilla CNN. Additionally, NeSyFOLD shows a consistent degradation in performance as the number of classes increases, largely due to the information loss introduced by post-hoc binarization. In contrast, our method consistently yields accuracy improvements across datasets, with the sole exception of *GT43*, where we observe a minor drop of 1 percentage point.

Furthermore, the rule-sets generated by NeSyViT are significantly more compact, with an average size of 11.85, which is **67%** smaller than those produced by NeSyFOLD – further emphasizing the interpretability of our approach.

[Q4, Q5] Neuron Activation Patterns and Automatic Semantic Labelling Efficacy: A key component of interpretability in neuro-symbolic frameworks is the alignment of individual neurons with semantically meaningful concepts relevant to specific classes.

Setup: To evaluate this alignment, we apply the previously described, semantic labelling algorithm to assign human-interpretable concept names to each predicate in the rule-set, where each predicate corresponds to a neuron in the final sparse concept layer. Initially, the raw rule-set produced by the FOLD-SE-M algorithm uses only neuron indices as predicate names. The semantic labelling process replaces these with descriptive concept labels derived by analyzing the top-10 most activating images per neuron. These images are overlaid with attention-based heatmaps and compared with pre-annotated semantic segmentation masks to identify the dominant concept for each neuron.

In Figure 2a), we visualize the top-5 most activating images for each predicate in the labelled rule-set for the *P3.1* dataset, which contains the classes “bathroom,” “bedroom,” and “kitchen.” Figure 2b) shows a side-by-side comparison of the raw and semantically labelled rule-sets, illustrating how the abstract neuron indices are transformed into interpretable concept-based rules.

Result: Examining the labelled rule-set in Figure 2b), we observe that just three rules are sufficient to achieve a classification accuracy of 99%. Consider Rule 2: `target(X, ‘bathroom’) :- not water_cooler1_tray1_refrigerator1_range1(X).` At first glance, this rule may appear counterintuitive – why would the absence of several kitchen-related concepts imply that the image depicts a bathroom rather than a bedroom? The explanation lies in the execution strategy of FOLD-SE-M, which evaluates rules in a top-down fashion. Rule 2 only activates if Rule 1 fails to fire. Rule 1 is: `target(X, ‘bedroom’) :- bed1(X).` Thus, if the image lacks the “bed” concept, Rule 1 is bypassed, and Rule 2 checks for the absence of kitchen concepts; if none are present, the image is classified as a “bathroom.” Rule 3: `target(X, ‘kitchen’) :- refrigerator2(X)` fires only if the previous two rules do not apply and the “refrigerator” concept is present, thereby classifying the image as kitchen.”

While the logical flow of the rules is valid, the quality of semantic labelling is less reliable. The labelling pipeline we adapted from NeSyFOLD was designed for CNNs, where modular filters make concept isolation more straightforward. In ViTs, however, neurons often attend to multiple regions and concepts simultaneously, making the learned representations less disentangled. As a result, some neuron labels can be misleading. This limitation is evident in Figure 2a: neuron 106 consistently activates for the “bed” concept in bedroom images, but neurons 43 and 105 do not display clear concept selectivity and instead respond to mixed features. This highlights the challenge of isolating concepts in self-attention-based architectures.

While the automatic labelling approach may have limitations, the visualization of neuron activation heatmaps remains a powerful tool. These heatmaps allow for manual inspection of concept associations and provide valuable interpretability cues. We believe this combination of automated rule generation and visual validation offers a promising direction. Also, since the extracted rule-set is an Answer Set Program we can get justification of any prediction using the s(CASP) (Arias *et al.* (2018)) ASP interpreter. For completeness, we provide the labelled rule-sets for all datasets used in our experiments in Table 1, in the Appendix.

5 Related works

The original ViT architecture (Dosovitskiy *et al.* (2021)) demonstrated that with sufficient data and computational resources, self-attention mechanisms could outperform CNNs in vision tasks. Since then, numerous variants have improved upon ViT's design. DeiT (Touvron *et al.* (2021)) introduced a data-efficient training strategy enabling ViTs to perform well even without massive datasets. Swin Transformer (Liu *et al.* (2021)) proposed a hierarchical design with shifted windows, improving both efficiency and performance on dense prediction tasks. Another notable extension includes ConvNeXt (Liu *et al.* (2022)), which bridges the gap between convolutional and transformer-based architectures while being faithful to the original CNN. These models aim to balance accuracy, parameter efficiency, and computational cost.

Although newer ViT variants often achieve higher accuracy on large-scale benchmarks such as ImageNet-1k and ImageNet-21k, the base ViT model remains a widely used and standardized backbone for evaluating interpretability methods. Since our primary goal is to explore interpretability through rule extraction – not to push state-of-the-art classification accuracy – we adopt the base ViT to isolate and evaluate the neurosymbolic contributions of our framework. Our results remain valid and meaningful in the context of interpretability research and can easily be extended to stronger ViT variants in future work.

The integration of symbolic reasoning with deep learning has led to various approaches for extracting interpretable rules from neural networks. While initial efforts focused on CNNs, recent studies have begun exploring ViTs. VisionLogic (Geng *et al.* (2025)) introduces a framework that transforms neurons in the final fully connected layer of deep vision models into predicates, grounding them into visual concepts through causal validation. ViT-NeT (Kim *et al.* (2022)) presents a neural tree decoder that interprets the decision-making process of ViTs. By routing images hierarchically through a tree structure, it provides transparent and interpretable classifications, addressing the trade-off between model complexity and interoperability. These methods primarily focus on post-hoc interpretability or rely on additional structures for explanation. In contrast, our approach integrates a sparse linear layer directly into the ViT architecture, enabling the extraction of executable logic programs without auxiliary components.

Enhancing the interpretability of ViTs has been a subject of extensive research, leading to various methodologies. INTR (Paul *et al.* (2024)) proposes a proactive approach, asking each class to search for itself in an image. This idea is realized via a Transformer encoder-decoder where "class-specific" queries (one for each class) are learnt as input to the decoder, enabling each class to localize its patterns in an image via cross-attention. Each class is attended to very distinctly hence the cross-attention weights provide a interpretation of the prediction. LeGrad (Bousselham *et al.* (2024)) introduces an explainability method specifically designed for ViTs. By computing gradients with respect to attention maps and aggregating signals across layers, LeGrad produces explainability maps that offer insights into the model's focus areas during decision-making. While these approaches enhance the transparency of ViTs, they do not produce symbolic or executable explanations. Our method differs by generating logic programs that serve as the final decision layer, enabling symbolic reasoning directly from the model's internal representations.

The pursuit of sparse and disentangled representations in ViTs has led to innovative methodologies. Recent work integrates sparse autoencoders (SAEs) with ViTs to improve interpretability. Joseph *et al.* (2025) trained SAEs on CLIP's ViT and found that manipulating a small set of steerable features can enhance performance and robustness. PatchSAE (Lim *et al.* (2025)) introduces method to extract granular visual concepts and study how these influence predictions, showing that most adaptation gains stem from existing features in the pre-trained model. Our approach builds upon these concepts by integrating a sparse linear layer into the ViT architecture and using it for extracting a symbolic rule-set which gives a global explanation of the model and a justification can be obtained for each prediction.

6 Conclusion and future work

In this work, we proposed a novel neuro-symbolic framework for interpretable image classification using ViTs. By replacing the final classification layer with a linear layer (sparse concept layer) producing sigmoid outputs, and introducing a unified loss function – comprising supervised contrastive loss for better class separation in latent space, entropy loss for sharper binarization, and sparsity loss for concept selectivity – we generated sparse binary vectors that represent images in a disentangled manner. These vectors serve as inputs to the FOLD-SE-M algorithm, which produces stratified Answer Set Programs in the form of concise rule-sets. The resulting neuro-symbolic (NeSy) model, composed of the modified ViT and the learned rule-set, outperforms the vanilla ViT by an average of **5.4%** in accuracy.

We compared our framework, **NeSyViT**, with **NeSyFOLD**, a similar approach for CNNs, and found that NeSyViT produces rule-sets that are **67% smaller** on average while improving upon the baseline ViT's accuracy – whereas NeSyFOLD suffers a **10.14%** drop in performance compared to its CNN counterpart. These results demonstrate the potential of attention-based architectures for interpretable, logic-driven classification.

Finally, we investigated the semantic interpretability of neurons in our framework. We observed that the automatic semantic labelling algorithm used in NeSyFOLD – based on overlap with segmentation masks – struggles in ViTs due to limited neuron monosemanticity. However, we argue that this challenge can be partially mitigated by visualizing attention-guided heatmaps for the small set of neurons that appear in the rule-set, enabling manual inspection and insight into the learned concepts.

Future work will focus on improving neuron disentanglement and enhancing monosemanticity using architectural or training refinements. Another promising direction explored in CNN-based interpretability methods is bias correction using extracted rule-sets (Padalkar et al. (2024c)) which could be the natural next step for this work along with counterfactual generation (Dasgupta et al. (2025)). We also plan to explore using multimodal LLMs like GPT-4o for concept labeling, enabling an automatic semantic annotation pipeline that does not rely on pixel-level segmentation masks.

Supplementary material

The supplementary material for this article can be found at https://doi.org/10.1017/S1471068425100318.

Competing interests

The author(s) declare none.

References

Arias, J., Carro, M., Salazar, E., Marple, K. and Gupta, G. 2018. Constraint answer set programming without grounding. *Theory and Practice of Logic Programming 18*, 337–354.

Baral, C. 2003. *Knowledge Representation, Reasoning and Declarative Problem Solving.* Cambridge University Press.

Bousselham, W., Boggust, A., Chaybouti, S., Strobelt, H. and Kuehne, H. 2024. Legrad: An explainability method for vision transformers via feature formation sensitivity. arXiv:2404.03214.

Dasgupta, S., Shakerin, F., Arias, J., Salazar, E. and Gupta, G. 2025. C3G: Causally constrained counterfactual generation. In *Practical Aspects of Declarative Languages*, E. Erdem and G. Vidal, Eds. Springer Nature Switzerland, Cham, 215–232.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. 2021. An image is worth 16 × 16 words: Transformers for image recognition at scale. In *Proc. 9th International Conference on Learning Representations*, Virtual Event, Austria, May 3-7, 2021. https://openreview.net/forum?id=YicbFdNTTy.

Gelfond, M. and Kahl, Y. 2014. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach.* Cambridge University Press.

Geng, C., Jiang, Y., Zhao, Z., Ye, H., Wang, Z. and Si, X. 2025. Learning interpretable logic rules from deep vision models. arXiv:2503.10547.

Joseph, S., Suresh, P., Goldfarb, E., Hufe, L., Gandelsman, Y., Graham, R., Bzdok, D., Samek, W. and Richards, B. A. 2025. Steering clip's vision transformer with sparse autoencoders. arXiv:2504.08729.

Kanagaraj, N., Hicks, D., Goyal, A., Tiwari, S., and Singh, G. 2021. Deep learning using computer vision in self driving cars for lane and traffic sign detection. *International Journal of System Assurance Engineering and Management* 12, 1011–1025.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C. and Krishnan, D. 2020. Supervised contrastive learning. In *Proc. Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, NeurIPS 2020. https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.

Kim, S., Nam, J. and Ko, B. C. 2022. Vit-net: Interpretable vision transformers with neural tree decoder. In International conference on machine learning, PMLR, 11162–11172.

Ko, B. and Kwak, S. 2012. Survey of computer vision-based natural disaster warning systems. In *Optical Engineering*, 2012, Vol. 51, 070901-1–070901-11.

Lage, I., Chen, E., He, J., Narayanan, M., Kim, B., Gershman, S. J. and Doshi-Velez, F. 2019. Human evaluation of models built for interpretability. In *Proc. Seventh AAAI Conference on Human Computation and Crowdsourcing*, HCOMP 2019, Vol. 7, 59–67.

Lim, H., Choi, J., Choo, J. and Schneider, S. 2025. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *Proc. The Thirteenth International Conference on Learning Representations*, ICLR 2025.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 10012–10022.

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T. and Xie, S. 2022. A convnet for the 2020s. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022, 11966–11976.

Ng, A. 2011. Sparse autoencoder. CS294A Lecture Notes, 72, 1–19.

Padalkar, P., Ślusarz, N., Komendantskaya, E. and Gupta, G. 2024c. A neurosymbolic framework for bias correction in convolutional neural networks. *Theory and Practice of Logic Programming*, 24, 4, 644–662.

Padalkar, P., Wang, H. and Gupta, G. 2024b. Using logic programming and kernel-grouping for improving interpretability of convolutional neural networks. In Proc. PADL 2024b, Vol. 14512 of LNCS, Springer, 134–150.

Padalkar, P., Wang, H. and Gupta, G. 2024a . Nesyfold: A framework for interpretable image classification. In Proc. AAAI 2024a, AAAI Press, 4378–4387.

Paul, D., Chowdhury, A., Xiong, X., Chang, F.-J., Carlyn, D., Stevens, S., Provost, K., Karpatne, A., Carstens, B., Rubenstein, D. I., Stewart, C. V., Berger-Wolf, T. Y., Su, Y. and Chao, W.-L. 2024. A simple interpretable transformer for fine-grained image classification and analysis. arXiv:2311.04157.

Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks* 32, 323–332.

Sun, W., Zheng, B. and Qian, W. 2016. Computer aided lung cancer diagnosis with deep learning algorithms. In SPIE Medical Imaging 2016.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In International conference on machine learning, PMLR, 10347–10357.

Townsend, J., Kasioumis, T. and Inakoshi, H. 2021. Eric: Extracting relations inferred from convolutions. In Computer Vision – ACCV 2021, Springer International, Cham, 206–222.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L.u and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Eds. Vol. 30, 5998–6008.

Wang, H. and Gupta, G. 2024. FOLD-SE: An efficient rule-based machine learning algorithm with scalable explainability. In Proc. PADL 2024, Vol. 14512 of LNCS, Springer, 37–53.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. 2018. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 1452–1464.