Towards a Neurosymbolic Understanding of Hidden Neuron Activations

Abhilekha Dalal [a,*], Rushrukh Rayan [a,**], Adrita Barua [a], Samatha Ereshi Akkamahadevi [a], Avishek Das [a], Cara Widmer [b], Eugene Y. Vasserman [a], Md Kamruzzaman Sarker [b] and Pascal Hitzler [a]

[a] *Computer Science, Kansas State University, KS, USA*
E-mails: adalal@ksu.edu, rushrukh@ksu.edu, adrita@ksu.edu, samatha94@ksu.edu, avishek@ksu.edu, eyv@ksu.edu, hitzler@ksu.edu
[b] *Kairos Research, LLC, OH, USA*
E-mail: cara@kairosresearch.com
[c] *Computer Science, Bowie State University, MD, USA*
E-mail: msarker@bowiestate.edu

Abstract. With the widespread adoption of Deep Learning techniques, the need for explainability and trustworthiness is increasingly critical, especially in safety-sensitive applications and for improved debugging, given the black-box nature of these models. The Explainable AI (XAI) literature offers various helpful techniques; however, many approaches use a secondary deep learning-based model to explain the primary model's decisions or require domain expertise to interpret the explanations. A relatively new approach involves explaining models using high-level, human-understandable concepts. While these methods have proven effective, an intriguing area of exploration lies in using a white-box technique to explain the probing model.

We present a novel, model-agnostic, post-hoc Explainable AI method that provides meaningful interpretations for hidden neuron activations. Our approach leverages a Wikipedia-derived concept hierarchy, encompassing approximately 2 million classes as background knowledge, and uses deductive reasoning-based Concept Induction to generate explanations. Our method demonstrates competitive performance across various evaluation metrics, including statistical evaluation, concept activation analysis, and benchmarking against contemporary methods. Additionally, a specialized study with Large Language Models (LLMs) highlights how LLMs can serve as explainers in a manner similar to our method, showing comparable performance with some trade-offs. Furthermore, we have developed a tool called ConceptLens, enabling users to test custom images and obtain explanations for model decisions. Finally, we introduce an entirely reproducible, end-to-end system that simplifies the process of replicating our system and results.

Keywords: Explainable AI, Neurosymbolic AI, Concept Induction, Background Knowledge

1. Introduction

Deep Learning solutions have been proven to be useful in a plethora of tasks in fields such as Computer Vision, Natural Language Processing, Signal Processing, etc. By tuning numerous neural network connection weights, decisions are driven towards their intended outcome repeatedly during the training process which in turn maximizes the likelihood of expected outcome during the inference phase. Such inferences incorporate a substantial amount

[*] Corresponding author. E-mail: adalal@ksu.edu.
[**] The first and second named author share first authorship of this paper.

of vector/matrix computations which are often untraceable in that, the sheer number of computations renders it impossible to be used as justifications of inference outcomes.

There are numerous techniques available to quantitatively and qualitatively measure such black-box models' performance. However, attaining justification by cleverly peeking into the models' internal mechanisms is a separate field, popularly termed as Explainable AI. The need for Explainable AI techniques designed for Deep Learning applications is manifold such as:

- Bias Detection and Mitigation: Explanations can help unfold underlying potential biases in the data or model which allows adjustments to ensure fairness
- Improving Model Performance: Explanations can serve as additional debugging information providing insight into potential errors, underfitting, or overfitting. Thus, allowing targeted improvements and refinement of models. Additionally, explanation can also highlight what features are most influential in the model's predictions.
- Ensuring Safety and Transparency: Explanations can serve AI Applications in Healthcare, Autonomous Vehicles, Finance by explaining their decisions which permits safe adoption of Deep Learning methods in such critical applications.

There are various families of XAI techniques – based on stages of explanation modeling: Ante-hoc, Post-hoc methods; based on scope of explainability: Global, Local methods; based on output formats: numerical, rule-based, textual, visual [1]. Some of the popular methods belonging to these families are: CAM [73], Grad-CAM [59], LIME [53], SHAP [40]. A relatively recent effort aiming more user-understandable explanations has given to develop Concept-based Explainable AI (C-XAI) methods [50]. Some of the recent C-XAI methods are: T-CAV [30], CAR [13], CaCE [21], ACE [19], ICE [20]. An in-depth review of existing XAI methods is discussed in Section 2.

Many XAI techniques rely on intricate low-level data features projected into a higher-dimensional space in their explanations, limiting their accessibility to users with domain expertise [41, 53, 59]. Some of these methods have shown vulnerability to adversarial tampering; altering attributed features does not induce a change in the model's decision [5, 61, 63]. The C-XAI approaches employ manually selected concepts that are measured for their correlation with model outcomes [13, 30]. However, a significant question remains unanswered: whether the limited set of chosen concepts can offer a comprehensive understanding of the model's decision-making process. The absence of a systematic approach to consider a wide range of potential concepts that may influence the model appears to be the bottleneck. In some techniques [46], a list of frequently occurring English words has been utilized to represent a broad concept pool, which may suffice for general applications but lacks granularity for specialized fields like gene studies or medical diagnoses, as the curation of the concept pool does not provide low-level control over defining natural relationships among concepts. An interesting aspect of XAI technique exploration is to have an explainer method that in itself does not utilize Deep Learning, but instead relies on symbolic, knowledge-based processing. Such an XAI method can be considered as a white-box method which is innately explainable.

Herein, we present a Neurosymbolic XAI approach using symbolic reasoning in the form of Concept Induction. The approach is motivated by several key principles. Firstly, explanations should be understandable to end-users without requiring intimate familiarity with deep learning models. Secondly, there should be a systematic organization of human-understandable concepts with well-defined relationships among them. The extraction of relevant concepts for explaining a deep learning model's decision-making process from this defined concept pool should be automatic, thus eliminating the bottleneck of manual curation prone to confirmation bias. Another significant goal is that the explanation generation technique itself should be inherently interpretable, avoiding the use of black-box methods. Our approach also incorporates a rigorous evaluation protocol encompassing various dimensions.

Concept Induction as core mechanism is based on formal logic reasoning (in the Web Ontology Language OWL [27, 54]) and has originally been developed for Semantic Web [26] applications [37]. The benefits of our approach are: (a) it can be used on unmodified and pre-trained deep learning architectures, (b) it assigns explanation categories (i.e., class labels expressed in OWL) to hidden neurons such that images related to these labels activate the corresponding neuron with high probability, (c) it is inherently self-explanatory as it is based on symbolic deductive reasoning, and (d) it can construct labels from a very large pool of interconnected categories.

We demonstrate that a background knowledge with the skeleton of an ontology coupled with the inherently explainable deductive reasoning (Concept Induction) should be capable of generating meaningful explanations for the deep learning model we wish to explain. To show that our approach can indeed provide meaningful explanations for hidden neuron activation, we instantiate it with a Convolutional Neural Network (CNN) architecture for image scene classification (trained on the ADE20K dataset [76]) and a class hierarchy (i.e., a simple ontology) of approx. $2 \cdot 10^6$ classes derived from Wikipedia as the pool of explanation categories [57].

Our findings suggest that our method performs competitively, as assessed through Concept Activation analysis, which measures the relevance of concepts within the hidden layer activation space, and through statistical evaluation. When compared to other techniques such as CLIP-Dissect [46], a pre-trained multimodal Explainable AI model, and GPT-4 [49], an off-the-shelf Large Language Model, our approach demonstrates both strong quantitative and qualitative performance.

The existing literature emphasizes the importance of labeling neuron with concepts, the focus is mostly on identifying what concepts activate a neuron; corresponding to the notion of recall in information retrieval. We argue that in C-XAI, attempting to explain a Neural Network through concepts is a two-step process. If a neuron is consistently activated when the concept of *Sky* is present in an image (i.e., Recall with respect to neuron label Sky) and is assigned with the concept label of *Sky*; it is equally important to asses the neuron's activation when *only* concepts other than *Sky* e.g., *River, Skyscraper, etc* are present in the images (i.e., Precision with respect to neuron label Sky). If the neuron is activated for many concepts other than *Sky*, the usefulness of such a C-XAI method which attempts to explain a neural network with concepts diminishes. The gap between high recall and low precision, in other words – high false positive rate renders a C-XAI neuron labeling method unreliable. That this occurs is of course not at all unexpected: it is entirely reasonable to assume that any information conveyed by hidden neuron activations be *distributed*, i.e., neurons naturally react to various stimuli, while specific information is indicated by simultaneous activation of neuron groups.

To that extent, we also present an analysis (based on [15]) which shows that our Neurosymbolic C-XAI method (based on [16]) achieves high recall as well as precision when labeling neuron with concepts. We do this by assigning *error margins* to neuron target labels. If a neuron is activated by a stimulus, then the error margin indicates the likelihood that the stimulus indeed falls under the neuron's target label, and this likelihood can be conveyed to the user. The error margins are statistically validated by means of data obtained from a user experiment conducted on Amazon Mechanical Turk.

We also include a special study to test the capability of LLMs as a concept discovery method to be used as a substitute of Concept Induction [7]. Our method discussed in Section 4 uses a heuristic implementation called ECII (Efficient Concept Induction from Individuals) [55] for explanation generation. We were interested to assess LLM's common-sense reasoning capability leveraging their vast domain knowledge for automated concept discovery in the same setting of Scene Classification using a CNN model. We have used GPT-4 to label neurons with high-level concepts through prompt engineering by essentially replacing ECII. Acknowledging the apparent trade-off of this method being a black-box XAI method as opposed to ECII being a white-box XAI method, human assessment conducted through Amazon Mechanical Turk to assess how meaningful the generated explanations are to humans, we find that while human-generated explanations remain superior, concepts derived from GPT-4 are more comprehensible to humans compared to those generated by ECII.

Core contributions of the paper are as follows.

1. A novel zero-shot model-agnostic C-XAI method that explains existing pre-trained deep learning models through high-level human understandable concepts, utilizing symbolic reasoning over an ontology (or Knowledge Graph schema) as the source of explanation, which achieves state-of-the-art performance and is explainable by its nature.
2. A method to automatically extract *relevant concepts* through Concept Induction for any concept-based Explainable AI method, eliminating the need for manual selection of Label Hypothesis concepts.
3. An in-depth comparison of explanation sources using statistical analysis for the hidden neuron perspective and Concept Activation analysis for the hidden layer perspective of our approach, a pre-trained multimodal Explainable AI method (CLIP-Dissect [46]), and a Large Language Model (GPT-4 [49]).
4. Introduction of error margins to neuron target labels to provide a quantitative measure of confidence for concept detection in Image Analysis tasks.
5. A fully automated end-to-end system to use Concept Induction to interpret neurons' in terms of concepts in a CNN [4], discussed in Section 6.

6. ConceptLens: A demonstrator designed to represent the concepts that trigger neuron activations in a CNN [14], discussed in Section 7.

Our work shows that combining symbolic reasoning with LLMs offers a powerful approach for producing explainable, human-understandable insights from deep learning models. This combination promises to improve both the interpretability and the performance of XAI techniques, providing more trustworthy and reliable AI systems.

The structure of this paper is as follows: in Section 2, we discuss some of the important related research efforts. In Section 4, we present the main method Concept Induction and core findings. Following that, in Section 5 we discuss the use of LLMs as a substitute for Concept Induction, in Section 6 we present the end-to-end automated tool, and in Section 7 we discuss the tool ConceptLens. In Section 9 we conclude.

This paper is an extended merger of several conference contributions: [16] is the central one for the overall narrative; [15] is an extension with a finer-grained analysis; [7] goes in detail on using LLMs as an alternative to concept induction; [4] reports on our automation of the analysis process (see Section 6); This paper extends these by providing a joint perspective, additional literature review, more discussion, and a demonstrator system (see Section 7) previously only reported as a pre-print [14].

2. Related Work

The need for explainable AI (XAI) has gained significant momentum since the 1970s with the growing complexity and opacity of deep learning models [36]. As AI is increasingly applied in diverse domains, explaining the rationale behind AI decisions is critical for trust and transparency [3, 23, 43]. Various methods have been proposed to achieve explainability, categorized primarily into approaches that focus on understanding features (e.g., feature summarizing [53, 58]) and those that focus on the model's internal units (e.g., node summarizing [8, 75]). Model-agnostic methods such as LIME [53] and SHAP [40] aim to explain model predictions by assessing feature importance, while other techniques rely on counterfactual questions for human interpretability [68]. However, feature attribution methods like LIME and SHAP face challenges such as instability [5] and bias susceptibility [63]. Another such work, Individual Conditional Expectation (ICE) [20] is a tool to visualize complex relationships between predictors and responses, allowing for a more granular view than traditional partial dependence plots. Though generating ICE plots can be computationally intensive, their model-agnostic nature allows them to interpret various "black box" models, enhancing flexibility across algorithms. [73] presents a pixel attribution method that uses global average pooling and Class Activation Mapping (CAM) to enable convolutional neural networks (CNNs) to perform object localization, even when only trained on image-level labels. Another work Grad-CAM [59] generalizes CAM by using the gradients of target classes flowing into the last convolutional layer to produce localization maps, thus making it compatible with a broader range of CNN models. Pixel attribution techniques, although useful for image-based models, encounter limitations with activation functions like ReLU and are prone to adversarial attacks [33, 61]. [29] introduces a framework for interpreting image representation features by identifying human-understandable concepts through contrasting high- and low-activation images. But the framework depends on a pre-trained vision-language model (i.e., CLIP), which may lack sufficient representation when applied to models trained on niche or uncommon datasets.

Recent works have introduced concept-based approaches, which provide human-understandable explanations by linking model behavior to predefined concepts. For instance, methods like TCAV [30] use human-provided concepts, while ACE [19] utilizes image segmentation and clustering to derive automated concepts. However, these approaches may lose information during segmentation or fail to capture low-level details. In another work, the limitations of TCAV approach are addressed for concept-based explanations in deep neural networks and concept activation region (CAR) [13] is introduced. It allows for the nonlinear separability of concepts in the latent space, offering better accuracy and alignment with human-understandable concepts. In [21] the authors introduce the Causal Concept Effect (CaCE) to measure the causal impact of high-level, human-interpretable concepts on a classifier's predictions, aiming to reduce confounding errors common in correlation-based interpretability methods. Although CaCE estimation relies on the accuracy of generative models, such as VAEs [34], which may not fully capture the true causal relationships in complex, real-world datasets. Other methods such as Concept Bottleneck Models

(CBM) [35] and Post-hoc CBM [72] attempt to map neural network models to human-interpretable concepts, but they often rely on hand-picked concepts, requiring significant human input and manual curation. [69, 71] make use of Concept Bottleneck techniques to achieve interpretability in Image Classification. [69] represents an image solely by the presence/absence of concepts learned through training over the target task without explicit supervision over the concepts. [71] uses GPT-3 to produce factual sentences about categories to form candidate concepts. [60, 64] study performing interventional interactions by updating concept values to rectify predictive outputs of the model. [10] extends CBMs to interactive prediction settings by developing an interaction policy which, at prediction time, chooses which concepts to request a label for. [31] introduces probabilistic concept-embeddings which models uncertainty in concept prediction and provides explanations based on the concept and its corresponding uncertainty.

The application of background knowledge, including the use of large ontologies, has been explored to generate more automated and systematic explanations. Semantic Web technologies [11, 17] and methods like Concept Induction [51, 56] have demonstrated the utility of formal logic and structured data to explain deep learning models, though these approaches often focus on input-output relationships rather than internal model activations. While methods such as Network Dissection (e.g., [75]) provide valuable insights by mapping hidden units with semantic concepts by comparing neuron activations against a pre-defined set of labels (typically derived from human-annotated datasets), they do not capture the full hierarchical and dynamic nature of learned concepts, nor do they incorporate an explicit reasoning process. Notably, CLIP-Dissect [46] employs zero-shot learning to associate images with labels using a pre-trained CLIP model, but this method is limited by its accuracy in predicting labels from hidden layers and its transferability across domains. Building upon this, Label-Free Concept Bottleneck Models [47] leverage GPT-4 [49] for concept generation, but similar to CLIP-Dissect, they face limitations in explainability and domain adaptability. [22] propose a novel knowledge-aware neuron interpretation framework to explain model predictions for image scene classification, using core concepts of a scene based on a knowledge graph, ConceptNet. In [6], neural networks do not make task predictions directly, but they build syntactic rule structures using concept embeddings. The Deep Concept Reasoner executes these rules on meaningful concept truth degrees to provide semantically-consistent and differentiable predictions. [65] uses Segment Anything Model (SAM) in a lightweight per-input equivalent scheme to enable efficient explanation with a surrogate model. [45] introduces quantization for sparse decision layers in an iterative fine-tuning loop which leads to a quantized self-explaining neural network.

Recent trends highlight the potential of large language models (LLMs) to bridge the gap between model complexity and human-understandable explanations. LLMs like GPT-3 and GPT-4 have been used in few-shot learning contexts to generate concepts with minimal human intervention [47], providing a scalable solution to automated concept discovery. However, these approaches still require post-processing to filter and refine generated concepts for practical use [12, 70]. While LLMs show promise in automating concept generation, challenges remain in aligning explanations with human common sense and ensuring that they cater to diverse user needs, whether system developers or end-users.

Table 1

Comparison of key features across explainability methods.

Feature/Methods	Pixel-attribution (CAM/Grad-CAM)	Feature-attribution (LIME/SHAP)	Concept-based (TCAV/ACE/CAR)	Zero-shot (CLIP-Dissect)	LLM-based (GPT-4)	**CI**
White-box reasoning	No	No	No	No	No	**Yes**
Ontology-driven concept pool	No	No	Partially[1]	No	No	**Yes**
Precision and Recall	No	No	No	No	No	**Yes**
Model-agnostic	Yes	Yes	Yes	Yes	Yes	**Yes**
End-to-end automation	No	No	No	Partialy[2]	No	**Yes**
Leverages large background knowledge	No	No	No	No[3]	Yes	**Yes**

Our approach distinguishes itself by leveraging symbolic deductive reasoning over a comprehensive background knowledge base derived from Wikipedia, comprising approximately 2 million interconnected classes to generate explanations. Unlike methods that depend on manual selection or post-hoc filtering of candidate concepts, our framework systematically extracts human-understandable labels directly from this knowledge base, reducing potential biases and ensuring scalability. Moreover, by operating as a white-box system, Concept Induction provides inherent transparency: each explanation can be traced back to logical reasoning steps, which contrasts with black-box methods as discussed above that do not reveal the underlying rationale behind their output. In this way, our approach not only offers improved interpretability but also facilitates a more scalable and systematic framework for understanding and comparing neuron activations.

3. Methodology Overview

Before diving into the detailed methodology, we provide a concise "Preliminaries" overview of our system architecture, training protocol, and concept-analysis pipeline (see Figure 1). This roadmap highlights the key components—neural network training, Concept Induction, and Concept Activation Analysis—each of which is fully elaborated in the subsequent sections.

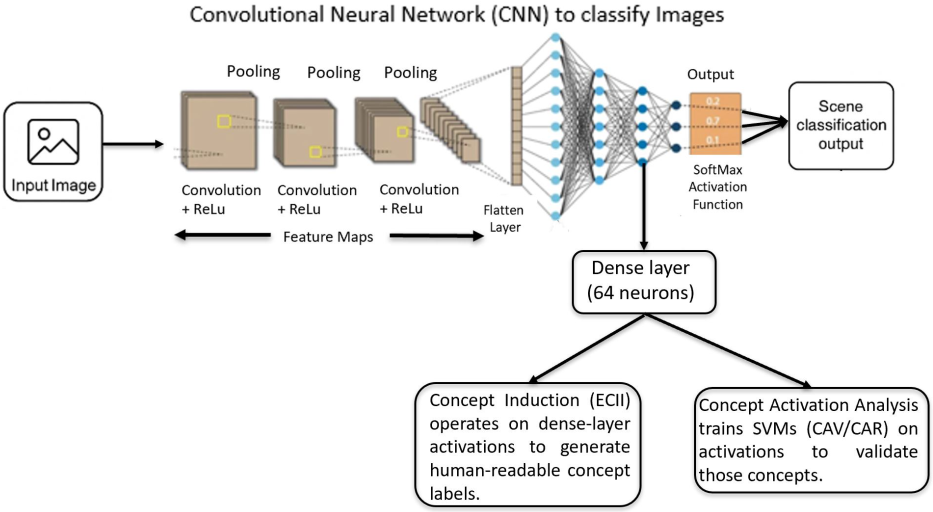

Convolutional Neural Network (CNN) to classify Images
Input Image
Pooling
Pooling
Pooling
Convolution + ReLu
Convolution + ReLu
Convolution + ReLu
Feature Maps
Flatten Layer
Output
0.2
0.7
0.1
SoftMax Activation Function
Scene classification output
Dense layer (64 neurons)
Concept Induction (ECII) operates on dense-layer activations to generate human-readable concept labels.
Concept Activation Analysis trains SVMs (CAV/CAR) on activations to validate those concepts.

Fig. 1. Overview: An input image dataset passes through a CNN (ResNet50V2) architecture with hidden layers to produce a scene classification output. The 64-unit dense layer (highlighted) feeds into two analysis modules: (1) Concept Induction (ECII), which generates human-readable concept labels from neuron activations, and (2) Concept Activation Analysis (CAV/CAR), which trains SVMs on the same activations to validate those concepts.

We train a convolutional neural network (ResNet50V2) on the ADE20K scene-classification task (10 classes, around 6200 images). All layers are fine-tuned for 30 epochs with early stopping (patience 3, lr=0.001) using categorical cross-entropy loss. This yields a stable 87% validation accuracy, ensuring the model is sufficiently reliable for downstream explanation without over- or under-fitting.

Next, we extract explanations at the network's final dense layer (64 neurons). In the Concept Induction step, each neuron's strongly activating images (*atleast* 80% of its peak response) and weakly activating images (*atmost*

[1] TCAV/ACE/CAR require manual or clustering-based concept selection

[2] CLIP-Dissect automates label assignment but not full pipeline from neuron activation to explanation.

[3] CLIP-Dissect uses a predefined English-word concept pool (via CLIP's vocabulary), not a structured ontology or large knowledge graph.

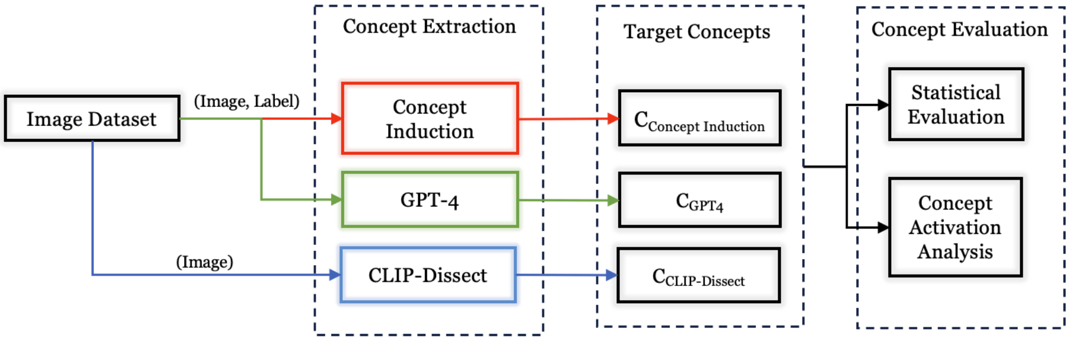


Fig. 2. An overview of the complete pipeline explored in this paper where Concept Extraction outlines the methods used to extract Target Concepts and Concept Evaluation outlines the evaluation methods.

20%) are combined with a large background ontology (approximately 2 million Wikipedia classes) and fed into ECII (Efficient Concept Induction from Individuals). ECII returns a small set of candidate labels (e.g. “skyscraper,” “cross_walk”) whose class expressions best separate the positive and negative image sets. We also explore an alternative LLM-based labeling: prompting GPT-4 directly on the same activation sets to propose high-level concepts, enabling a comparison between symbolic- and language-model explainers.

Each proposed label yields a hypothesis (“Concept X drives Neuron N”). In hypothesis confirmation, we retrieve new images for Concept X via Google Images, measure neuron activations on those images, and apply a Mann–Whitney U test to ensure “target” images activate N significantly more than “non-target” images ($p < 0.05$). Labels that pass are deemed confirmed neuron-concept associations. To assess how broadly each concept lives in the dense-layer space, we perform Concept Activation Analysis: for every confirmed concept, we train two SVMs on dense-layer activations (linear=CAV, nonlinear=CAR) to distinguish images with versus without that concept. The resulting test accuracy quantifies each concept’s global footprint.

Finally, we introduce an error-margin analysis: for each neuron–concept pair, we compute the likelihood that an activation truly corresponds to the concept (precision margin) as well as the concept’s recall. This yields statistically validated confidence bounds on every explanation. All steps—from input image through CNN, Concept Induction or GPT-4 labeling, hypothesis testing, activation analysis, and error-margin computation—are integrated into our end-to-end tool, ConceptLens. The following sections unpack each component in detail.

4. A Neurosymbolic Approach with Concept Induction

We explore and evaluate three concrete methods to generate high-level concepts for explaining hidden neuron activations. Fig. 2 is a high-level depiction of our workflow. Fig. 2 and its components are further discussed below and throughout the paper. In section 4.1 we present preparations regarding the scenario, the CNN training, and Concept Induction. In section 4.2 we provide details on our three label hypothesis generation approaches. In section 4.3 we describe our different evaluation protocols. In section 4.4 we provide evaluation results, followed by additional discussion in section 4.5.

4.1. Preliminaries

In this section, we describe the experimental setup that underpins our evaluation of Concept Induction. We begin by outlining the scenario used to demonstrate our approach, including the selection of image data, training of a CNN, and the integration of background knowledge for concept extraction. These preparatory steps set the stage for a detailed explanation of our methodology, which is further elaborated in the following sub-sections.

4.1.1. Scenario and CNN Training

We use a scene classification from images scenario to demonstrate our approach, drawing from the ADE20K dataset [76] which contains more than 27,000 images over 365 scenes, extensively annotated with pixel-level objects and object part labels. *The annotations are not used for CNN training*, but rather only for generating label hypotheses that we will describe in Section 4.2.1.

We train a classifier for the following scene categories: "bathroom," "bedroom," "building facade," "conference room," "dining room," "highway," "kitchen," "living room," "skyscraper," and "street." We selected scene categories with the highest number of images, and we deliberately include some scene categories that should have overlapping annotated objects – we believe this makes the hidden node activation analysis more interesting. We did not previously conduct any experiments on any other scene selections, i.e., *we did not change our scene selections based on any preliminary analyses*.

We trained a number of CNN architectures in order to use the one with highest accuracy, namely Vgg16 [62], InceptionV3 [66] and different versions of Resnet – Resnet50, Resnet50V2, Resnet101, Resnet152V2 [24, 25]. Each neural network was fine-tuned with a dataset of 6,187 images (training and validation set) of size 224 × 224 for 30 epochs with early stopping[4] to avoid overfitting. We used Adam as our optimization algorithm, with a categorical cross-entropy loss function and a learning rate of 0.001.

We select Resnet50V2 because it achieves the highest accuracy (see Table 2). Note that for our investigations, which focus on explainability of hidden neuron activations, achieving a very high accuracy for the scene classification task is not essential, but a reasonably high accuracy is necessary when considering models which would be useful in practice.

Table 2

Performance (accuracy) of different architectures on the ADE20K dataset. The system we used, based on performance, is **in bold**.

Architectures	Training acc	Validation acc
Vgg16	80.05%	46.22%
InceptionV3	89.02%	51.43%
Resnet50	35.01%	26.56%
Resnet50V2	**87.60%**	**86.46%**
Resnet101	53.97%	53.57%
Resnet152V2	94.53%	51.04%

4.1.2. Concept Induction

Concept Induction [37] is based on deductive reasoning over description logics, i.e., over logics relevant to ontologies, knowledge graphs, and generally the Semantic Web field [26, 27] including the W3C OWL standard [54]. Concept Induction has been demonstrated in other scenarios to produce meaningful labels for human interpretation [70]. A Concept Induction system accepts three inputs,

- a set of positive examples P,
- a set of negative examples N, and
- a knowledge base (or ontology) K,

all expressed as description logic theories, and all examples $x \in P \cup N$ occur as individuals (constants) in K. It returns description logic class expressions E such that $K \models E(p)$ for all $p \in P$ and $K \not\models E(q)$ for all $q \in N$. If no such class expressions exist, then it returns approximations for E together with a number of accuracy measures.

For scalability reasons [57], we use the heuristic Concept Induction system ECII [55] together with a background knowledge base that consists only of a hierarchy of approximately 2 million classes, curated from the Wikipedia

[4] monitor validation loss; patience 3; restore weights

concept hierarchy and presented in [57]. We use *coverage* as accuracy measure, defined as

$$\text{coverage}(E) = \frac{|\{p \in P \mid K \models E(p)\}| + |\{n \in N \mid K \not\models E(n)\}|}{|P \cup N|}, \tag{1}$$

with P, N, K as above.

For our setting, positive and negative example sets contain images from ADE20K, i.e., we include the images in the background knowledge by linking them to the class hierarchy. For this, we use the object annotations available for the ADE20K images, but only part of the annotations for simplicity and scalability. More precisely, we only use the information that certain objects (such as *Windows*) occur in certain images, and we do not make use of any of the richer annotations such as those related to segmentation.[5] All objects from all images are then mapped to classes in the class hierarchy using the Levenshtein string similarity metric [38] with edit distance 0. This metric computes the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another, and we normalize the result to assess the degree of similarity between the strings. Mapping is in fact automated using the "combine ontologies" function of ECII.

4.2. Generating Label Hypotheses

In the following, we detail the components shown in Fig. 2. We explain our use of Concept Induction for generating explanatory concepts, followed by our utilization of CLIP-Dissect and GPT-4 for the same. We describe our three evaluation approaches in Section 4.3.

4.2.1. Generating Label Hypotheses using Concept Induction

The general idea for generating label hypotheses using Concept Induction is as follows: given a hidden neuron, P is a set of inputs (i.e., in this case, images) to the deep learning system that activate the neuron, and N is a set of inputs that do not activate the neuron (where P and N are the sets of positive and negative examples, respectively). As mentioned above, inputs are annotated with classes from the background knowledge for Concept Induction, but these annotations and the background knowledge are not part of the input to the deep learning system. ECII generates a label hypothesis[6] for the given neuron on inputs P, N, and the background knowledge.

We first feed 1,370 ADE20K images to our trained Resnet50V2 and retrieve the activations of the dense layer. We chose to look at the dense layer because previous studies indicate [48] that earlier layers of a CNN respond to low level features such as lines, stripes, textures, colors, while layers near the final layer respond to higher-level features such as face, box, road, etc. The higher-level features align better with the nature of our background knowledge. The dense layer consists of 64 neurons, and we analyze each separately. Activation patterns involving more than one neuron are likely also informative in the sense that information may be distributed among several neurons, but this will be part of future investigations.

For each neuron, we calculate the maximum activation value across all images. We then take the positive example set P to consist of all images that activate the neuron with *at least* 80% of the maximum activation value, and the negative example set N to consist of all images that activate the neuron with *at most* 20% of the maximum activation value (or do not activate it at all). We selected these thresholds as our best guess (further refinement may be possible in future) based on experimental observations to ensure that the positive set is predominantly comprised of images in which the target concept is clearly expressed, while the negative set is limited to images with minimal or no activation, thereby reducing overlap and enhancing the reliability of the subsequent concept extraction. The highest scoring response of running ECII on these sets, together with the background knowledge described above, is shown in Table 3 for each neuron, together with the coverage of the ECII response. For each neuron, we call its corresponding label the *target label*, e.g., neuron 0 has target label "building." Note that some target labels consist of two concepts, e.g., "footboard, chain" for neuron 49 – this occurs if the corresponding ECII response carries two

[5] In principle, complex annotations in the form of sets of OWL axioms could of course be used, if a Concept Induction system is used that can deal with them, such as DL-Learner [37]. However DL-Learner does not quite scale to our size of background knowledge and task [56].

[6] In fact, it generates several, ranked, but we use only the highest ranked one for now.

class expressions joined by a logical conjunction, i.e., in this example “footboard ⊓ chain” (as description logic expression) or footboard(x) ∧ chain(x) expressed in first-order predicate logic.

We give an example, depicted in Figure 3, for neuron 1. The green and red boxed images show positive and negative examples for neuron 1. Concept Induction yields “cross_walk” as target label. The example is continued below.

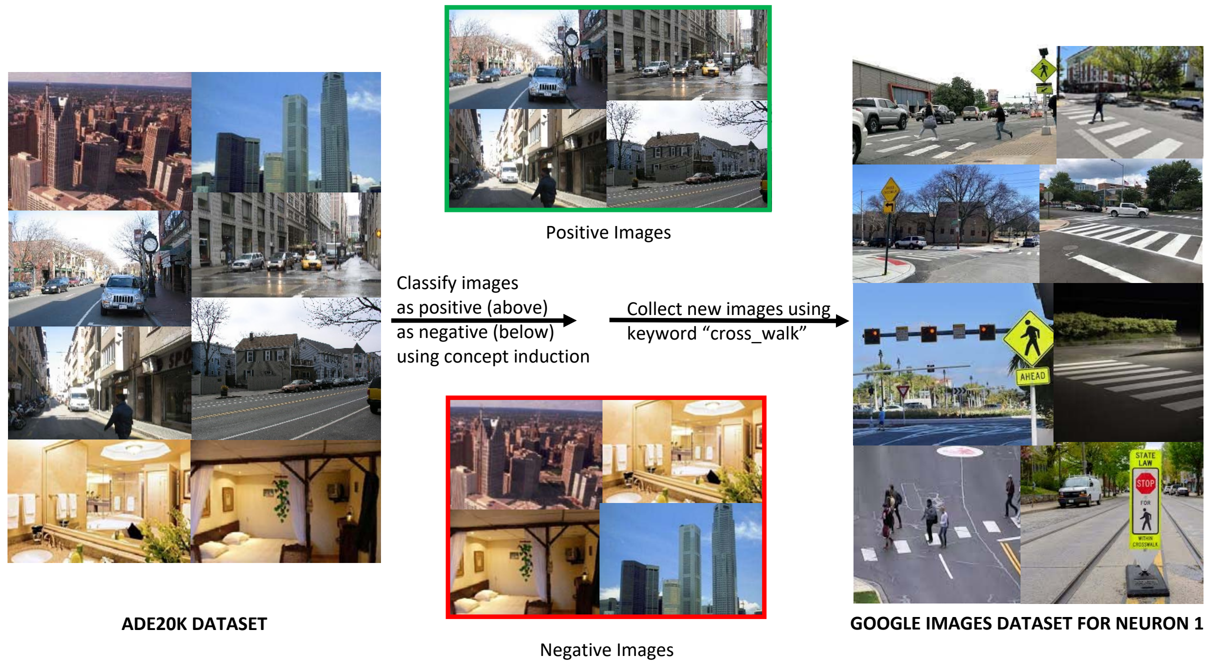

Positive Images
Classify images as positive (above) as negative (below) using concept induction
Collect new images using keyword “cross_walk”
ADE20K DATASET
Negative Images
GOOGLE IMAGES DATASET FOR NEURON 1
AHEAD
STATE LAW
STOP
FOR
WITHIN CROSSWALK

Fig. 3. Example of images that were used for generating and confirming the label hypothesis for neuron 1.

4.2.2. *CLIP-Dissect*

CLIP-Dissect [46] is a zero-shot Explainable AI method that associates high-level concepts with individual neurons in a designated layer. It utilizes the pre-trained multimodal model CLIP [52] to project a set of concepts and a set of images into shared embedding space. Using Weighted Pointwise Mutual Information, it assesses the similarities between concepts and images in the hidden layer activation space to assign a concept to a neuron.

First, CLIP-Dissect uses a set of the most common 20,000 English vocabulary words as concepts. Then, we collect activations from our ResNet50v2 trained model for the ADE20K test images. This results in a matrix of dimensions (Number of Images × 64), where each row in the matrix represents an image through its 64 hidden neuron activation values. With these two sets of input, CLIP-Dissect assigns a label to each neuron such that the neuron is most activated when the corresponding concept is present in the image. This yields 22 unique concepts for the 64 neurons, with duplicate concepts for several neurons.

4.2.3. *GPT-4*

We employ a Large Language Model (LLM) for concept selection. Specifically, we use GPT-4, which represents the latest advancement in generative models and offers improved reliability, outperforming existing LLMs across various tasks [49]. These models appear capable of generating concepts essential for distinguishing between different image classes when prompted effectively [47].

For this approach, we use the same positive (P) and negative (N) example sets from Section 4.1.2, with some minor adjustments: For Concept Induction, the negative example set (N) comprises all images that activate the neuron with at most 20% of the maximum activation value. Due to constraints on having a large number of negative image tags as input to GPT-4, we select only one image per class of images for each neuron to create the negative

Table 3

Concept Induction – The omitted neurons were not activated by any image, i.e., their maximum activation value was 0. Images: Number of images used per label. Target %: Percentage of target images activating the neuron above 80% of its maximum activation. Non-Target %: The same, but for all other images. **Bold** denotes the 20 neurons whose labels are considered confirmed.

Neuron	Obtained Label(s)	Images	Coverage	Target %	Non-Target %
0	**building**	**164**	**0.997**	**89.024**	**72.328**
1	**cross_walk**	**186**	**0.994**	**88.710**	**28.923**
3	**night_table**	**157**	**0.987**	**90.446**	**56.714**
6	dishcloth, toaster	106	0.999	16.038	39.078
7	toothbrush, pipage	112	0.991	75.893	59.436
8	**shower_stall, cistern**	**136**	**0.995**	**100.000**	**53.186**
11	river_water	157	0.995	31.847	22.309
12	baseboard, dish_rag	108	0.993	75.926	48.248
14	rocking_horse, rocker	86	0.985	54.651	47.816
16	**mountain, bushes**	**108**	**0.995**	**87.037**	**24.969**
17	stem	133	0.993	30.827	31.800
18	**slope**	**139**	**0.983**	**92.086**	**69.919**
19	**wardrobe, air_conditioning**	**110**	**0.999**	**89.091**	**65.034**
20	fire_hydrant	158	0.990	5.696	13.233
22	**skyscraper**	**156**	**0.992**	**99.359**	**54.893**
23	fire_escape	162	0.996	61.111	18.311
25	spatula, nuts	126	0.999	2.381	0.883
26	skyscraper, river	112	0.995	77.679	35.489
27	manhole, left_arm	85	0.996	35.294	26.640
28	flooring, fluorescent_tube	115	1.000	38.261	33.198
29	**lid, soap_dispenser**	**131**	**0.998**	**99.237**	**78.571**
30	**teapot, saucepan**	**108**	**0.998**	**81.481**	**47.984**
31	fire_escape	162	0.961	77.160	63.147
33	tanklid, slipper	81	0.987	41.975	30.214
34	left_foot, mouth	110	0.994	20.909	49.216
35	utensils_canister, body	111	0.999	7.207	11.223
36	**tap, crapper**	**92**	**0.997**	**89.130**	**70.606**
37	cistern, doorcase	101	0.999	21.782	24.147
38	letter_box, go_cart	125	0.999	28.000	31.314
39	side_rail	148	0.980	35.811	34.687
40	sculpture, side_rail	119	0.995	25.210	21.224
41	**open_fireplace, coffee_table**	**122**	**0.992**	**88.525**	**16.381**
42	pillar, stretcher	117	0.998	52.137	42.169
43	**central_reservation**	**157**	**0.986**	**95.541**	**84.973**
44	saucepan, dishrack	120	0.997	69.167	36.157
46	Casserole	157	0.999	45.223	36.394
48	**road**	**167**	**0.984**	**100.000**	**73.932**
49	**footboard, chain**	**126**	**0.982**	**88.889**	**66.702**
50	night_table	157	0.972	65.605	62.735
51	**road, car**	**84**	**0.999**	**98.810**	**48.571**
53	pylon, posters	104	0.985	11.538	17.332
54	**skyscraper**	**156**	**0.987**	**98.718**	**70.432**
56	**flusher, soap_dish**	**212**	**0.997**	**90.094**	**63.552**
57	**shower_stall, screen_door**	**133**	**0.999**	**98.496**	**31.747**
58	plank, casserole	80	0.998	3.750	3.925
59	manhole, left_arm	85	0.994	35.294	21.589
60	paper_towels, jar	87	0.999	0.000	1.246
61	ornament, saucepan	102	0.995	43.137	17.274
62	sideboard	100	0.991	21.000	29.734
63	**edifice, skyscraper**	**178**	**0.999**	**92.135**	**48.761**

example set (*N*). The positive image set (*P*) remain unchanged, given its smaller size. All these images are sourced from the ADE20K dataset as before and are labeled with object tags present in the image.

Object tags from these images are passed into GPT-4 via the OpenAI API using prompts to generate explanations aimed at discerning the distinguishing features present in the positive set (*P*) that were absent in the negative set (*N*). These explanations were treated as concepts, and we generated a top-three list of concepts for each neuron using zero-shot prompting. For each neuron, we ran the prompt with the following parameters:

- Positive example set: object tags of all positive images (*P*)
- Negative example set: object tags of all negative images (*N*)
- Prompt question: Generate the top three classes of objects or general scenario that better represent what images in the positive set (*P*) have but the images in the negative set (*N*) do not.

We employ the most recent version of the GPT-4 model for this task, with the model's temperature set to 0 and top_p to 1. These parameters significantly influence the output diversity of GPT-4: higher temperatures (e.g., 0.7) lead to more varied and imaginative text, whereas lower temperatures (e.g., 0.2) produce more focused and deterministic responses. Setting the temperature to 0 theoretically selects the most probable token at each step, with minor variations possible due to GPU computation nuances even under deterministic settings. In contrast to temperature sampling, which modulates randomness in token selection, top_p sampling restricts token selection to a subset (the nucleus) based on a cumulative probability mass threshold (top_p). OpenAI's documentation advises adjusting either temperature or top_p but not both simultaneously to control model behavior effectively. For our study, setting the temperature to 0 ensured consistency and reproducibility across outputs. More detailed information regarding the experimental setup and complete prompt can be found in section 5 below.

Although three concepts were generated for each neuron, we selected only one concept per neuron for analysis, resulting in 64 unique concepts, with several neurons having duplicate concepts.

4.3. Concept Evaluation Protocols

We describe the two evaluations, Statistical and Concept Activation Analysis, that we have performed for each of the concept selection methods, as depicted in Fig.2. We also describe an additional Error Margin Analysis, in section 4.3.3, that goes deeper on the Concept Induction scenario.

4.3.1. Statistical Evaluation

Confirming Label Hypotheses The three approaches described above produce label hypotheses for all investigated neurons – hypotheses that we will confirm or reject by testing the labels with new images. We use each of the target labels to search Google Images with the labels as keywords (requiring responses to be returned for *both* keywords if the label is a conjunction of classes, for Concept Induction). We call each such image a *target image* for the corresponding label or neuron. We use Imageye[7] to automatically retrieve the images, collecting up to 200 images that appear first in the Google Images search results, filtering for images in JPEG format and with a minimum size of 224x224 pixels (conforming to the size and format of ADE20K images).

For each retrieval label, we use 80% of the obtained images, reserving the remaining 20% for the statistical evaluation described later in the section. The number of images used in the hypothesis confirmation step, for each label, is given in the tables. These images are fed to the network to check (a) whether the target neuron (with the retrieval label as target label) activates, and (b) whether any other neurons activate. The Target % column of Tables 3, 4, and 5 show the percentage of the target images that activate each neuron.

Returning to our example neuron 1 in the Concept Induction case (Fig. 3), 88.710% of the images retrieved with the label "cross_walk" activate it. However, this neuron activates only for 28.923% (indicated in the Non-Target % column) of images retrieved using all other labels excluding "cross_walk."

We define a target label for a neuron to be *confirmed* if it activates for $\geqslant 80\%$ of its target images regardless of how much or how often it activates for non-target images. The cut-offs for neuron activation and label hypothesis

[7] https://chrome.google.com/webstore/detail/image-downloader-imageye/agionbommeaifngbhincahgmoflcikhm

Table 4

CLIP-Dissect – The omitted neurons were not activated by any image, i.e., their maximum activation value was 0. Images: Number of images used per label. Target %: Percentage of target images activating the neuron above 80% of its maximum activation. Non-Target %: The same, but for all other images. **Bold** denotes the 8 neurons whose labels are considered confirmed.

Neuron	Obtained Label(s)	Images	Target %	Non-target%
0	restaurants	140	55.000	59.295
1	restaurants	140	32.143	33.851
3	**dresser**	**171**	**95.322**	**66.199**
6	dining	153	7.190	50.195
7	**bathroom**	**153**	**93.333**	**44.113**
8	restaurants	140	24.286	37.957
11	highway	153	14.063	25.153
12	street	140	5.797	50.253
14	file	160	54.375	69.867
16	bathroom	171	2.000	31.722
17	furnished	169	62.130	36.390
18	**dining**	**153**	**93.464**	**74.448**
19	bathroom	149	77.333	56.471
20	buildings	107	13.725	19.610
22	road	258	51.550	46.487
23	bedroom	123	0.637	18.823
25	restaurants	140	12.857	5.044
26	restaurants	140	2.143	44.552
27	bedroom	150	2.548	27.763
28	dining	153	9.150	40.747
29	street	150	78.261	66.277
30	bed	150	29.375	36.154
31	mississauga	146	30.137	57.175
33	**bathroom**	**150**	**80.667**	**32.955**
34	microwave	102	3.922	50.240
35	roundtable	72	16.667	14.932
36	municipal	154	51.299	67.002
37	bed	160	8.125	17.670
38	**bathroom**	**150**	**90.667**	**32.566**
39	restaurants	140	26.429	39.961
40	dining	153	5.882	32.143
41	bedroom	157	64.968	34.428
42	room	156	35.897	45.206
43	**highways**	**128**	**100.000**	**61.900**
44	buildings	153	9.150	38.377
46	restaurants	140	23.571	33.269
48	bedroom	157	8.917	60.241
49	**bedroom**	**157**	**95.541**	**55.917**
50	**bedroom**	**157**	**100.000**	**62.744**
51	bedroom	157	4.459	51.951
53	kitchens	155	50.968	24.886
54	dining	153	13.725	62.857
56	bedroom	157	1.911	45.676
58	buildings	153	0.654	10.455
59	buildings	153	35.294	24.156
61	street	69	1.449	14.697
62	street	69	24.638	44.722
63	bathroom	150	16.667	47.584

Table 5

GPT-4 – The omitted neurons were not activated by any image, i.e., their maximum activation value was 0. Images: Number of images used per label. Target %: Percentage of target images activating the neuron above 80% of its maximum activation. Non-Target %: The same, but for all other images. **Bold** denotes the 27 neurons whose labels are considered confirmed.

Neuron	Obtained Label(s)	Images	Target %	Non-target%
0	Urban Landscape	176	54.545	59.078
1	**Street Scene**	**164**	**92.073**	**29.884**
3	**Bedroom**	**165**	**97.576**	**62.967**
6	**Kitchen**	**171**	**86.550**	**51.733**
7	Indoor Home Decor	177	66.102	44.793
8	**Bathroom**	**164**	**98.780**	**47.897**
11	Kitchen Scene	167	41.916	26.281
12	Indoor Home Setting	164	62.805	47.205
14	**Living Room**	**164**	**82.317**	**65.053**
16	Urban Landscape	176	73.864	28.290
17	**Dining Room**	**159**	**93.711**	**46.339**
18	**Outdoor Scenery**	**164**	**92.073**	**73.852**
19	Indoor Home Decor	177	29.379	45.571
20	Street Scene	164	68.902	14.305
22	**Street Scene**	**164**	**90.244**	**51.273**
23	**Street Scene**	**164**	**81.098**	**19.507**
25	Kitchen	171	21.637	5.628
26	Cityscape	156	73.718	28.023
27	Urban Transportation	163	66.871	30.152
28	Classroom	162	60.494	60.494
29	**Bathroom**	**164**	**91.463**	**68.926**
30	**Kitchen**	**171**	**90.643**	**41.724**
31	**Urban Street Scene**	**163**	**80.864**	**67.201**
33	Bathroom	164	74.390	37.272
34	Eyeglasses	168	65.476	45.208
35	Kitchen	171	66.667	13.224
36	**Bathroom**	**164**	**95.122**	**61.704**
37	Bathroom	164	43.902	10.487
38	**Living Room**	**164**	**94.512**	**56.087**
39	**Bicycle**	**156**	**82.692**	**46.328**
40	Living Room	164	70.122	24.156
41	**Living Room**	**164**	**95.122**	**41.616**
42	Living Room	164	48.780	46.431
43	**Outdoor Urban Scene**	**163**	**91.411**	**57.925**
44	**Kitchen Scene**	**167**	**86.826**	**45.721**
46	Kitchen Scene	167	43.114	31.155
48	**Urban Street Scene**	**163**	**99.383**	**55.061**
49	**Bedroom**	**165**	**95.758**	**36.120**
50	**Living Room**	**164**	**93.902**	**62.756**
51	**Street Scene**	**164**	**98.171**	**43.830**
53	Street Scene	164	57.317	23.575
54	Home Interior	165	26.061	63.216
56	**Toilet Brush**	**165**	**94.545**	**35.095**
57	**Bathroom Interior**	**165**	**95.092**	**41.549**
58	Kitchen Scenario	165	29.268	11.096
59	**Urban Street Scene**	**163**	**87.037**	**26.217**
60	Kitchen	171	0.585	1.691
61	Kitchen	171	60.819	11.810
62	**Dining Room**	**159**	**94.969**	**44.128**
63	**Cityscape**	**156**	**95.513**	**47.791**

confirmation are chosen to ensure strong association and responsiveness to images retrieved under the target label, but 80% is somewhat arbitrary and could be chosen differently.

For our example neuron 1, we retrieve 233 new images with the keyword “cross_walk,” 186 of which (80%) are used in this step. 165 of these images, i.e., 88.710% activate neuron 1. Since $88.710 \geqslant 80$, we consider the label “cross_walk” confirmed for neuron 1.

After this step, we arrive at a list of 19 (distinct) *confirmed* labels from Concept-Induction, 5 (distinct) *confirmed* labels from CLIP-Dissect, and 14 (distinct) *confirmed* labels from GPT-4, as listed in Table 6.

Label Validation After generating the confirmed labels (as above), we evaluate the node labeling using the remaining images from those retrieved from Google Images as described earlier. Results are shown in Table 6, omitting neurons that were not activated by any image, i.e., their maximum activation value was 0.

We consider each neuron-label pair (rows in Table 6) to be a hypothesis, e.g., for neuron 1 in Table 6, the hypothesis is that it activates more strongly for images retrieved using the keyword “cross_walk” than for images retrieved using other keywords. The corresponding null hypothesis is that activation values are *not* different. Table 6 shows the 20 hypotheses from Concept Induction to test, corresponding to the 20 neurons with confirmed labels from method Concept Induction (recall that a double label such as neuron 16’s “mountain, bushes” is treated as one label consisting of the conjunction of the two keywords.)

Similarly, Table 6 also lists the 8 hypotheses to test, corresponding to the 8 neurons with confirmed labels from method CLIP-Dissect, and the 27 hypotheses to test, corresponding to the 27 neurons with confirmed labels from method GPT-4.

There is no reason to assume that activation values would follow a normal distribution, or that the preconditions of the central limit theorem would be satisfied. We therefore base our statistical assessment on the Mann-Whitney U test [42] which is a non-parametric test that does not require a normal distribution. Essentially, by comparing the ranks of the observations in the two groups, the test allows us to determine if there is a statistically significant difference in the activation percentages between the target and non-target labels.

The resulting z-scores and p-values are shown in Table 6 and are further discussed in Section 4.4. For our running example (neuron 1), we analyze the remaining 47 target images (20% of the images retrieved during the label hypothesis confirmation step). Of these, 43 (91.49%) activate the neuron with a mean and median activation of 4.17 and 4.13, respectively. Of the remaining (non-target) images in the evaluation (the sum of the image column in Table 6 Concept Induction Section minus 47), only 28.94% activate neuron 1 for a mean of 0.67 and a median of 0.00. The Mann-Whitney U test yields a z-score of -8.92 and $p < 0.00001$. The negative z-score indicates that the activation values for non-target images are indeed lower than for the target images, rejecting the null hypothesis.

It is instructive to have another look at our example neuron 1 for the Concept Induction case. The images depicted on the left in Fig. 4 – target images not activating the neuron – are mostly computer-generated as opposed to photographic images as in the ADE20K dataset. The lower right image does not actually show the ground at the crosswalk, but mostly sky and only indirect evidence for a crosswalk by means of signage, which may be part of the reason why the neuron does not activate. The right-hand images are non-target images that activate the neuron. We may conjecture that other road elements, prevalent in these pictures, may have triggered the neuron. We also note that several images show bushes or plants – particularly interesting because the ECII response with the third-highest coverage score is “bushes, bush” with a coverage score of 0.993 and 48.052% of images retrieved using this label actually activate the neuron (the second response for this neuron is also “cross_walk”). It appears that Concept Induction results should be further improvable by taking additional Concept Induction returns into consideration. While we will not entirely follow through on this idea in this paper, we look into it to some extent in Section 4.3.3.

4.3.2. *Concept Activation Analysis*

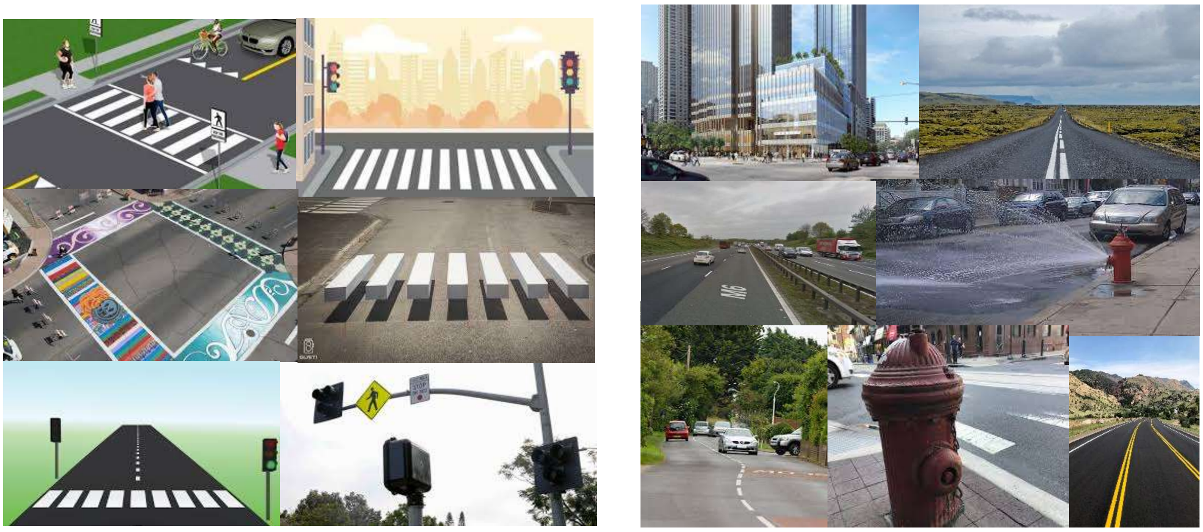
Concept Induction is a separate process from the neural network based processes. Leveraging the strength of the background knowledge, it outputs a list of high-level concepts based on single neuron activation patterns. A question we can ask is: can we find existence or absence of such concepts in the full hidden layer activation space?

To that extent, we employ *Concept Activation* [13, 30], which is a concept-based explainable AI technique which works with a *pre-defined* set of concepts. It attempts at explaining a pre-trained model by measuring the presence of *concepts* in hidden-layer activations of a given image for a particular layer. For the purpose of comparative analysis, we evaluate all candidate concepts (label hypotheses), obtained from all three methods, through Concept Activation

Table 6

Evaluation details for all three approaches as discussed in Section 4.3.1. Images: Number of images used for evaluation. # Activations: (targ(et)): Percentage of target images activating the neuron;(non-t):Same for all other images used in the evaluation. Mean/Median (targ(et)/non-t(arget)): Mean/median activation value for target and non-target images, respectively.

Neuron	Label(s)	Images	# Activations (%)		Mean		Median		z-score	p-value
			targ	non-t	targ	non-t	targ	non-t		
	Concept Induction									
0	building	42	80.95	73.40	2.08	1.81	2.00	1.50	-1.28	0.0995
1	cross_walk	47	91.49	28.94	4.17	0.67	4.13	0.00	-8.92	<.00001
3	night_table	40	100.00	55.71	2.52	1.05	2.50	0.35	-6.84	<.00001
8	shower_stall, cistern	35	100.00	54.40	5.26	1.35	5.34	0.32	-8.30	<.00001
16	mountain, bushes	27	100.00	25.42	2.33	0.67	2.17	0.00	-6.72	<.00001
18	slope	35	91.43	68.85	1.59	1.37	1.44	1.00	-2.03	0.0209
19	wardrobe, air_conditioning	28	89.29	65.81	2.30	1.28	2.30	0.84	-4.00	<.00001
22	skyscraper	39	97.44	56.16	3.97	1.28	4.42	0.33	-7.74	<.00001
29	lid, soap_dispenser	33	100.00	80.47	4.38	2.14	4.15	1.74	-5.92	<.00001
30	teapot, saucepan	27	85.19	49.93	2.52	1.05	2.23	0.00	-4.28	<.00001
36	tap, crapper	23	91.30	70.78	3.24	1.75	2.82	1.29	-3.59	<.00001
41	open_fireplace, coffee_table	31	80.65	15.11	2.03	0.14	2.12	0.00	-7.15	<.00001
43	central_reservation	40	97.50	85.42	7.43	3.71	8.08	3.60	-5.94	<.00001
48	road	42	100.00	74.46	6.15	2.68	6.65	2.30	-7.78	<.00001
49	footboard, chain	32	84.38	66.41	2.63	1.67	2.30	1.17	-2.58	0.0049
51	road, car	21	100.00	47.65	5.32	1.52	5.62	0.00	-6.03	<.00001
54	skyscraper	39	100.00	71.78	4.14	1.61	4.08	1.12	-7.60	<.00001
56	flusher, soap_dish	53	92.45	64.29	3.47	1.48	3.08	0.86	-6.47	<.00001
57	shower_stall, screen_door	34	97.06	32.31	2.60	0.61	2.53	0.00	-7.55	<.00001
63	edifice, skyscraper	45	88.89	48.38	2.41	0.83	2.36	0.00	-6.73	<.00001
	CLIP-Dissect									
3	dresser	43	93.02	64.61	2.59	1.42	2.62	0.68	5.01	<0.0001
7	bathroom	46	89.47	41.56	2.02	1.01	2.15	0.00	5.45	<0.0001
18	dining	36	94.87	76.82	3.01	1.85	3.11	1.44	4.52	<0.0001
33	bathroom	38	71.05	34.02	1.28	0.47	0.95	0.00	4.91	<0.0001
38	bathroom	38	84.21	31.71	1.79	0.54	1.83	0.00	7.14	<0.0001
43	highways	32	100.00	63.87	7.00	3.14	6.39	2.64	6.17	<0.0001
49	bedroom	40	97.50	55.77	3.48	1.63	3.43	0.63	6.05	<0.0001
50	bedroom	40	97.50	63.21	4.56	1.30	4.60	0.66	8.70	<0.0001
	GPT-4									
1	Street Scene	42	90.50	30.40	3.80	0.70	4.20	0.00	-9.62	<0.0001
3	Bedroom	42	97.60	63.40	4.70	1.20	4.90	0.70	-9.05	<0.0001
6	Kitchen	43	83.70	52.00	2.40	1.00	2.00	0.10	-5.06	<0.0001
8	Bathroom	41	100.00	44.10	4.10	1.00	4.10	0.00	-9.57	<0.0001
14	Living Room	41	78.00	67.50	1.40	1.30	1.20	0.90	-0.77	0.4413
17	Dining Room	40	97.50	45.90	2.20	0.60	2.50	0.00	-8.29	<0.0001
18	Outdoor Scenery	41	100.00	76.10	2.30	1.50	2.20	1.20	-3.96	<0.0001
22	Street Scene	42	90.50	50.10	3.00	1.40	3.30	0.00	-5.95	<0.0001
23	Street Scene	42	85.70	20.70	2.40	0.30	2.10	0.00	-10.83	<0.0001
29	Bathroom	41	90.20	68.40	2.60	1.50	2.40	1.00	-4.05	<0.0001
30	Kitchen	43	86.00	38.60	2.60	0.80	2.70	0.00	-7.22	<0.0001
31	Urban Street Scene	41	80.50	65.70	1.80	1.30	1.70	0.90	-2.4	0.164
36	Bathroom	41	100.00	61.30	3.10	1.20	2.80	0.60	-7.48	<0.0001
38	Living Room	41	92.70	54.30	2.00	1.00	2.20	0.30	-5.53	<0.0001
39	Bicycle	39	84.60	47.40	2.10	0.90	2.40	0.00	-5.64	<0.0001
41	Living Room	41	97.60	42.00	2.60	0.60	2.30	0.00	-9.31	<0.0001
43	Outdoor Urban Scene	41	92.70	56.30	4.10	2.40	4.30	1.00	-4.42	<0.0001
44	Kitchen Scene	42	81.00	43.40	2.30	1.00	2.10	0.00	-5.43	<0.0001
48	Urban Street Scene	41	100.00	52.60	4.90	2.30	4.80	0.40	-6.03	<0.0001
49	Bedroom	42	95.20	35.00	3.80	0.70	4.00	0.00	-10.31	<0.0001
50	Living Room	41	97.60	63.90	3.00	1.20	2.60	0.60	-6.78	<0.0001
51	Street Scene	42	95.20	42.90	5.70	1.50	6.10	0.00	-9.05	<0.0001
56	Toilet Brush	42	97.60	34.60	3.60	0.70	3.60	0.00	-10.48	<0.0001
57	Bathroom Interior	41	92.70	40.50	3.00	0.80	2.90	0.00	-8.35	<0.0001
59	Urban Street Scene	41	82.90	26.30	2.70	0.50	2.50	0.00	-9.06	<0.0001
62	Dining Room	40	90.00	43.90	3.30	0.80	3.70	0.00	-8.64	<0.0001
63	Cityscape	39	97.40	48.50	2.80	0.70	2.40	0.00	-8.76	<0.0001



Target images not activating neuron 1

Non-target images activating neuron 1

Fig. 4. Examples of some Google images used: target images ("cross_walk") that did not activate the neuron; non-target images from labels like "central_reservation," "road and car," and "fire_hydrant" that activated the neuron.

Analysis. Note that we do not restrict this analysis to only confirmed concepts, as the Concept Activation Analysis approach has not been developed with such a confirmation step as part of it.

For each candidate concept, a set of images are collected using Imageye (exactly as described above) and a concept classifier (i.e., a Support Vector Machine) is trained. The dataset given to the concept classifier requires some pre-processing:

i. The dataset for one concept classifier consists of images that exhibit the presence of the concept under description and with images where the said concept is absent. As the concept classifier will output the existence or absence of a concept, we assign the images to have labels 0 (when concept is absent) and 1 (when concept is present).

ii. Since we are interested in finding the concepts in the hidden layer activation space, not in the image pixel space, we need to transform the image pixel values to their activation values. To achieve that, the dataset is passed across the ResNet50V2 pre-trained model as it is the network we wish to explain. The activation values of each image in the dense layer is saved. If the dense layer consists of 64 neurons, then we end up with a matrix of dimensions (no. of images × 64).

The transformed dataset is split into train (80%) and test (20%) datasets. Thereafter, a Support Vector Machine (SVM) is trained using the train split. We have used both linear (Concept Activation Vector, CAV) and non-linear (Concept Activation Region, CAR) kernel to see which decision boundary separates the presence/absence of a concept best. Once the concept classifier is trained, a test dataset is used to see to what extent the concept classifier can classify the presence/absence of concepts in the hidden layer activation space.

We use Concept Induction, CLIP-Dissect, and GPT-4 as Concept Extraction mechanisms. Thereafter we use Concept Activation analysis to measure to what extent such concepts are identifiable in the hidden layer activation space. We adopt two different kernels through CAV and CAR to train an SVM and then test the classifiers on unseen image data. Tables 15, 16, and 17 represent the test accuracies for the concepts extracted by Concept Induction, CLIP-Dissect, and GPT-4. Table 7 represents the results of the Mann-Whitney U test performed over the test accuracies

Table 7

Summary of Concept Activation Analysis results of Concept Induction, CLIP-Dissect, and GPT-4 using Mann-Whitney U test

Method	CAV		CAR	
	z-score	p-value	z-score	p-value
Concept Induction x CLIP-Dissect	0.1252	0.9004	-0.8717	0.3834
CLIP-Dissect x GPT-4	1.7494	0.0801	**1.9680**	**0.0488**
Concept Induction x GPT-4	**2.1560**	**0.0308**	1.7792	0.0751

obtained from all 3 approaches. Table 12 shows the Mean, Median, and Standard Deviation of the test accuracies for each of the 3 approaches.

4.3.3. Additional Error Margin Analysis for Concept Induction

In this section, we outline our technical approach for assessing neuron-label associations through *error-margin* analysis (Non-target Label Activation Percentage, or Non-TLA). Non-TLA represents the percentage of images *not falling under the target label* that activate a neuron that carries the target label as per the prior analysis. Similarly, Target Label Activation Percentage, TLA, represents the percentage of images falling under the target label that activate the neuron that carries the target label.

To obtain error margins, we calculate activation percentages for both target labels and non-target labels per neuron based on Google Images retrieved from the labels as search terms, and we also take into account activation patterns of neuron groups for semantically related labels, analyzing TLA and Non-TLA across different cutoff values. We then use images from the ADE20K dataset [76], with annotations improved thorugh Amazon Mechanical Turk, to statistically validate the error-margins obtained earlier.

Computation of Non-TLA Concept Induction generates a number of concept labels for each neuron unit, ranked by some accuracy measure. Herein, we consider the Top-3 labels (ranked by coverage score) for each of the 64 neurons in the dense layer. Using the Target-Label image dataset (each image falls under the target label), the TLA is calculated, and, using a Non-target Label image dataset (none of the images contain the target label), the Non-TLA is calculated. To obtain a nuanced understanding of how activation levels affect the reliability of the neuron–concept association, we calculate TLA and Non-TLA for each neuron at specified activation value thresholds, namely > 0, > 20%, > 40%, and > 60% of the max activation value that was recorded for that neuron. These thresholds are our best guess for balancing sensitivity and specificity, and we acknowledge they are heuristic and may be refined in future work. For example, (see Table 8), neuron 43 activates at > 40% of its max activation value in about 19.7% of images *not* showing a central reservation.

Neuron Ensembles for Concept Associations The distribution of input information across simultaneously activated neurons necessitates the investigation of neuron ensemble activations at different cut-off activation values. However, an exhaustive analysis of all neuron ensembles does not scale as even just 64 neurons give rise to 2^{64} possible neuron ensembles. We deal with this by considering only ensembles of neurons that activate for semantically related labels. For example, the concept *building* activates both neurons 0 and 63 (see Table 8); we evaluate all images from Non-target Label image dataset as well as Target Label image dataset *separately*, activating neurons 0 and 63 at the specified cut-off activation values, to calculate TLA and Non-TLA.

In scenarios where a concept activates more than two neurons, our analysis encompasses all possible combinations of pairs, triples, etc., of neurons (see *skyscraper* in Table 8). We then narrow our focus to a list of highly associated concepts corresponding to the neurons (see the Concepts column in Table 8), that demonstrate TLA exceeding 80%, i.e., those neurons with high recall.

Annotations of ADE20K Dataset The analysis just described yields *error-margins* associated with each concept, for each of the chosen activation thresholds listed in Table 8. For example, the concept *buffet* has an *error-margin* of 12.374 for the Non-TLA of > 20%: Our analysis suggests the *hypothesis* that at most 12.374% of *non-buffet* images activate the neuron unit 62 at 20% of its max activation value. In other words, the *error-margin* at Non-TLA of > 20% for the concept *buffet* is 12.374%. If this hypothesis can be substantiated, then upon presentation of a new input to the network, activation of neuron 62 of at least 20% of its max activation value means that a *buffet* has been detected, and that this detection is *wrong* in at most about 12.374% of cases.

In order to substantiate our hypotheses, we analyse neuron activation values for new inputs, more precisely for images taken from the ADE20K dataset. We take advantage of the fact that ADE20K images already carry rich object annotations, however we have observed that they are still too incomplete for our purpuses. Therefore we made use of Amazon Mechanical Turk via the Cloud Research platform, to add missing annotations from a list of concepts derived from Table 8 to 1,050 randomly chosen ADE20K images.

For this set of 1,050 ADE20K images, we conducted a user study through Amazon Mechanical Turk using the Cloud Research platform, to annotate images based on a list of concepts derived from Table8. The study protocol was reviewed and approved by the Institutional Review Board (IRB) at Kansas State University and was deemed exempt under the criteria outlined in the Federal Policy for the Protection of Human Subjects, 45 CFR §104(d), category: Exempt Category 2 Subsection ii. The study was conducted in 35 batches (each batch containing 30 images), with 5 participants per study compensated with $5 for completing the task. The task was estimated to take approximately 40 minutes, equivalent to $7.50 per hour.

For each image, users were presented with a list of concepts (a concise form of concepts from Table 8) to choose from, including buffet, building, building and dome, central_reservation, clamp_lamp and clamp, closet and air-conditioning, cross_walk, edifice and skyscraper, faucet and flusher, field, flusher and soap_dish, footboard and chain, hedgerow and hedge, lid and soap_dispenser, mountain, mountain and bushes, night_table, open_fireplace and coffee_table, pillow, potty and flusher, road, road and automobile, road and car, route, route and car, shower_stall and cistern, Shower_stall and screen_door, skyscraper, slope, tap and crapper, tap and shower_screen, teapot and saucepan, wardrobe and air-conditioning.

Users were allowed to select multiple concepts for each image, indicating all concepts that applied to the given image. These selected concepts were considered annotations for the respective image.

Validating Neuron-Concept Associations To assess the validity of the *error-margins* retrieved from the Google Image dataset for all concepts in Table 8, we look at activations yielded by ADE20K images, and hypothesize that they are similar or lower (i.e., not higher), for non-target images. Non-TLA are computed across the predefined cut-off activation thresholds. Selected values can be found in Table 9. For example, the central reservation neuron 43 mentioned above activates above its 40% max activation threshold for about 14.9% of ADE20K non-target images (not showing central reservations), while it activates for about 19.7% of Google non-target images.

Both single-neuron and neuron ensemble activations are considered and shown in Table 9.

4.4. Results

For the given test dataset split of ADE20K, we looked at Concept Induction, CLIP-Dissect, and GPT-4 for extracting relevant candidate concepts. Subsequently, we conducted two analyses from different perspectives.

i. For each neuron of the dense layer, we identify the concepts that activate them the most (Statistical Evaluation).
ii. For each concept, we measure its degree of relevance across the entire dense layer activation space (Concept Activation Analysis).

We will now bring together the results. We will also present results from the additional error margin analysis.

4.4.1. Comparison of Concept Extraction Approaches

The combination of the two evaluation perspectives – a detailed examination of how each neuron unit functions and a broader view of how the dense layer operates as a whole – enables us to gain a comprehensive insight into the inner workings of hidden layer computations.

Regarding statistical evaluation, we rigorously assess the significance of differences in activation percentages between target and non-target labels for each confirmed label hypothesis. We compute the z-score and p-value using the non-parametric Mann-Whitney U test. Additionally, we calculate the Mean and Median for both target and non-target labels to further characterize the results. In the Concept Activation Analysis, we evaluate the effectiveness of concepts across several dimensions. Initially, we assess each concept classifier considering both linear (CAV) and non-linear (CAR) decision boundary based on the presence and absence of each concept. To validate that the concept classifier's test accuracy is not merely coincidental, we conduct K-fold cross-validation and calculate p-values. Additionally, we compute the Mean, Median, and Standard Deviation, and perform the Mann-Whitney U

Table 8

Non-target Label Activation Percentages (Non-TLA) for Google dataset: The table showcases a refined selection, inclusive of concepts and neuron ensembles with targ(et) activation > 80%. Non-t: percentage of non-target images that activate the neuron(s) associated with the concept being analyzed across various activation thresholds.

Concepts	Neuron	**targ %>0**	Non-target % for different threshold values			
			non-t >0	non-t > 20%	non-t > 40%	non-t > 60%
buffet	62	**83.607**	32.714	12.374	3.708	0.825
building	0	**89.024**	72.328	39.552	12.040	2.276
building	0, 63	**80.164**	43.375	12.314	2.276	0.182
building and dome	0	**90.400**	78.185	45.133	14.643	2.639
central_reservation	43	**95.541**	84.973	57.993	19.734	2.913
clamp_lamp and clamp	7	**95.139**	59.504	29.229	9.000	1.652
closet and air_conditioning	19	**86.891**	71.054	38.491	10.135	1.267
cross_walk	1	**88.770**	28.241	6.800	1.524	0.521
edifice and skyscraper	63	**92.135**	48.761	21.786	8.379	2.229
faucet and flusher	29	**95.695**	78.562	37.862	12.104	1.873
field	18	**91.824**	65.333	30.207	8.183	1.656
flusher and soap_dish	56	**90.094**	63.552	29.901	7.695	1.148
footboard and chain	49	**88.889**	66.702	40.399	17.064	4.399
hedgerow and hedge	54	**91.165**	68.527	30.421	7.685	1.352
lid and soap_dispenser	29	**99.237**	78.571	34.989	9.052	1.485
mountain and bushes	16	**87.037**	24.969	10.424	4.666	1.937
mountain and bush	16	**87.037**	24.969	10.424	4.666	1.937
mountain	43	**99.367**	88.516	64.169	23.112	4.326
night_table	3	**90.446**	56.714	27.691	7.691	1.137
open_fireplace and coffee_table	41	**88.525**	16.381	4.325	0.812	0.088
pillow	3	**98.214**	61.250	28.228	7.249	1.001
pillow	50	**99.405**	66.834	24.242	4.101	0.530
pillow	3, 50	**97.605**	46.492	9.634	0.988	0.049
potty and flusher	29	**88.525**	76.830	36.537	10.755	1.932
road and car	51	**98.810**	48.571	25.373	8.399	3.261
road and automobile	51	**92.560**	41.466	16.055	3.301	0.701
road	48	**100.000**	76.789	47.897	18.843	3.803
road	48, 51	**97.099**	44.592	17.727	3.471	0.702
route	48	**100.000**	80.834	51.873	21.034	4.979
route and car	51	**92.628**	47.408	18.871	4.081	1.416
route	48, 51	**94.334**	45.089	18.937	4.809	1.169
shower_stall and cistern	8	**100.000**	53.186	24.788	8.485	1.372
Shower_stall and screen_door	57	**98.496**	31.747	12.876	4.121	1.026
slope	18	**92.143**	64.503	29.976	6.894	1.200
tap and crapper	36	**89.130**	70.606	36.839	13.696	2.511
tap and shower_screen	36	**86.250**	72.584	32.574	7.836	0.860
teapot and saucepan	30	**81.481**	47.984	18.577	4.367	0.845

Concepts	Neuron	**targ %>0**	Non-target % for different threshold values			
			non-t >0	non-t > 20%	non-t > 40%	non-t > 60%
wardrobe and air_conditioning	19	**89.091**	65.034	31.795	6.958	1.145
skyscraper	22	**99.359**	54.893	21.914	0.977	0.977
skyscraper	54	**98.718**	70.432	26.851	7.050	0.941
skyscraper	63	**94.393**	51.612	20.618	5.775	1.143
skyscraper	22, 26	**82.116**	22.274	3.423	0.292	0.004
skyscraper	26, 54	**82.225**	28.782	5.444	0.703	0.054
skyscraper	22, 54	**97.165**	47.422	7.910	0.465	0.000
skyscraper	22, 63	**96.947**	36.408	5.521	0.449	0.008
skyscraper	26, 63	**81.788**	21.421	3.335	0.534	0.088
skyscraper	54, 63	**96.074**	37.149	5.594	0.615	0.046
skyscraper	22, 26, 54	**81.461**	18.940	2.363	0.169	0.000
skyscraper	22, 26, 63	**81.243**	15.252	1.706	0.184	0.004
skyscraper	22, 54, 63	**95.420**	29.090	3.023	0.234	0.000
skyscraper	26, 54, 63	**81.134**	16.823	1.975	0.350	0.023
skyscraper	22, 26, 54, 63	**80.589**	13.093	0.872	0.015	0.000

test to quantify the statistical significance of the test accuracies. This comprehensive approach ensures a robust evaluation of the concepts' performance in activating the hidden layer.

Our findings suggest that Concept Induction consistently performs well in all evaluations conducted – Statistical Evaluation, Concept Activation Analysis, and also Error Margin Analysis. From the statistical evaluation, it is evident that Concept Induction achieves better performance than that of CLIP-Dissect and GPT-4. In the Concept Activation Analysis, quantitative measures reveal that Concept Induction achieves comparable performance to CLIP-Dissect, with GPT-4 exhibiting the lowest performance. Conversely, the Concept Induction approach demonstrates several notable qualitative advantages over both CLIP-Dissect and GPT-4:

- CLIP-Dissect and GPT-4 are black-box models used as a concept extraction method to explain a probing network, which in this case is a CNN model, i.e., this approach to explainability is itself not readily explainable. In contrast, Concept Induction, serving as a concept extraction method, inherently offers explainability as it operates on deductive reasoning principles.
- CLIP-Dissect relies on a common English vocabulary (about 20K words) as the pool of concepts, whereas Concept Induction is supported by a meticulously constructed background knowledge (in this case with about 2M concepts), affording greater control over the definition of explanations through hierarchical relationships.
- While GPT-4/CLIP-Dissect emulate intuitive and rapid decision-making processes, Concept Induction follows a systematic and logic-based decision-making approach – thereby rendering our approach to be explainable by nature.

The results in Table 6 show that Concept Induction analysis with large-scale background knowledge yields meaningful labels that stably explain neuron activation. Of the 20 null hypotheses from Concept Induction, 19 are rejected at $p < 0.05$, but most (all except neurons 0, 18 and 49) are rejected at much lower p-values. Only neuron 0's null hypothesis could not be rejected. With CLIP-Dissect, all 8 null hypotheses are rejected at $p < 0.05$, and with GPT-4, 25 out of 27 null hypotheses are rejected at $p < 0.05$, with exceptions for neurons 14 and 31. Excluding repeating concepts, Concept Induction yields **19** statistically validated hypotheses, CLIP-Dissect yields **5**, and GPT-4 yields **12**.

The Non-Target % column of Table 3 provides some insight into the results for neurons 0, 18, 49 and neurons 14, 31 from Table 5: target and non-target values for these neurons are closer to each other. Likewise, differences between target and non-target values for mean activation values and median activation values in Table 6 are smaller for these neurons. This hints at ways to improve label hypothesis generation or confirmation, and we will discuss this and other ideas for further improvement below under possible future work.

Table 9

Non-target Label Activation Percentages (Non-TLA) for ADE20K and Google Image dataset: Non-t: percentage of non-target label images that activate the neuron(s) associated with the concept being analyzed across various activation thresholds.

Concepts	non-t >0		non-t >20%		non-t >40%		non-t >60%	
	google	ADE20K	google	ADE20K	google	ADE20K	google	ADE20K
buffet	32.714	40.135	12.374	25.817	3.708	9.470	0.825	1.804
building	43.375	11.458	12.314	5.208	2.276	1.458	0.182	0.000
building and dome	78.185	26.170	45.133	5.893	14.643	0.867	2.639	0.000
central_reservation	84.973	44.893	57.993	34.343	19.734	14.927	2.913	3.816
clamp_lamp and clamp	59.504	27.273	29.229	19.170	9.000	8.300	1.652	1.976
closet and air_conditioning	71.054	30.168	38.491	15.620	10.135	5.513	1.267	1.378
cross_walk	28.241	21.474	6.800	16.391	1.524	9.784	0.521	2.922
edifice and skyscraper	48.761	24.187	21.786	8.453	8.379	1.300	2.229	0.260
faucet and flusher	78.562	56.967	37.862	30.580	12.104	11.097	1.873	1.850
field	65.333	66.161	30.207	30.043	8.183	10.412	1.656	2.386
flusher and soap_dish	63.552	19.481	29.901	10.035	7.695	3.896	1.148	0.236
footboard and chain	66.702	27.975	40.399	13.671	17.064	5.063	4.399	1.013
hedgerow and hedge	68.527	45.120	30.421	28.390	7.685	13.308	1.352	2.028
lid and soap_dispenser	78.571	57.512	34.989	18.427	9.052	2.817	1.485	0.352
mountain	88.516	45.144	64.169	33.725	23.112	16.115	4.326	3.842
mountain and bushes	24.969	28.331	10.424	16.573	4.666	6.607	1.937	1.904
night_table	56.714	30.534	27.691	15.267	7.691	5.954	1.137	1.679
open_fireplace and coffee_table	16.381	26.139	4.325	10.590	0.812	2.413	0.088	0.268
pillow	46.492	12.500	9.634	3.869	0.988	1.190	0.049	0.149
potty and flusher	76.830	58.410	36.537	24.194	10.755	4.608	1.932	1.152
road	44.592	8.501	17.727	6.955	3.471	4.328	0.702	0.927
road and automobile	41.466	17.604	16.055	14.497	3.301	8.728	0.701	2.811
road and car	48.571	14.815	25.373	11.704	8.399	6.074	3.261	1.333
route	45.089	12.349	18.937	10.241	4.809	5.723	1.169	1.807
route and car	47.408	17.073	18.871	14.204	4.081	7.461	1.416	2.152
shower_stall and cistern	53.186	25.982	24.788	9.700	8.485	4.965	1.372	1.039
Shower_stall and screen_door	31.747	24.910	12.876	14.320	4.121	5.897	1.026	1.203
skyscraper	13.093	3.009	0.872	0.463	0.015	0.231	0.000	0.116
slope	64.503	66.520	29.976	29.967	6.894	9.879	1.200	1.976
tap and crapper	70.606	62.225	36.839	12.861	13.696	4.890	2.511	0.611
tap and shower_screen	72.584	62.621	32.574	13.180	7.836	4.733	0.860	0.607
teapot and saucepan	47.984	23.632	18.577	11.176	4.367	6.519	0.845	1.281
wardrobe and air_conditioning	65.034	30.525	31.795	16.160	6.958	5.525	1.145	0.967

Mann-Whitney U results show that, for most neurons listed in Table 6 (with $p < 0.00001$), activation values of target images are *overwhelmingly* higher than that of non-target images. The negative z-scores with high absolute values informally indicate the same, as do the mean and median values. Neurons 16 and 49 of Table 6 Concept Induction section, for which the hypotheses also hold but with $p < 0.05$ and $p < 0.01$, respectively, still exhibit statistically significant higher activation values for target than for non-target images, but not overwhelmingly so. This can also be informally seen from lower absolute values of the z-scores, and from smaller differences between the means and the medians.

For the Concept Activation Analysis evaluation (see Table 12), Concept Induction yields **69** unique concepts with Mean Test Accuracy of **0.9154** (CAV) and **0.9150** (CAR). CLIP-Dissect identifies **22** concepts with Mean Test

Accuracy of **0.9160** (CAV) and **0.9259** (CAR). GPT-4 produces **21** concepts with Mean Test Accuracy of **0.8757** (CAV) and **0.8887** (CAR). Although, based solely on the numeric values of Mean Test Accuracy, CLIP-Dissect demonstrates a slightly superior performance compared to Concept Induction, and GPT-4 performs the least, we contend that the substantially higher number of concepts generated by Concept Induction allows CLIP-Dissect to achieve a marginally higher test accuracy. By considering the top 22 (equal to the number of concepts generated by CLIP-Dissect) test accuracies of concepts extracted by Concept Induction, the Mean Test Accuracy increases to **0.9599** (CAV) and **0.9584** (CAR). For statistical confirmation, we conduct a p-value test for K-fold cross validation, wherein all concepts in Concept Activation analysis achieve $p < 0.05$. Using a Mann-Whitney U test, we statistically ascertain that CLIP-Dissect outperforms GPT-4 in terms of CAR, and Concept Induction surpasses GPT on CAV (see Table 7).

This analysis leads us to the following conclusion: Among the three approaches we evaluate, Concept Induction demonstrates superior performance both in the quantity of high-quality concepts generated and in the relevance of these concepts within the hidden layer activation space. Furthermore, our approach possesses inherent explainability as it does not depend on any pre-trained black-box model to identify candidate concepts. However, there are undoubtedly trade-offs involved in selecting among the three approaches, which we elaborate on in Section 5.4.

Based on the results obtained from the Statistical Evaluation and Concept Activation analysis, our approach introduces a novel zero-shot, model-agnostic Explainable AI technique. This technique offers insights into the hidden layer activation space by utilizing high-level, human-understandable concepts. Leveraging deductive reasoning over background knowledge, our approach inherently provides explainability while also achieving competitive performance, thus confirming our initial hypothesis.

4.4.2. Error Margin Analysis

For a statistical evaluation of our error margin values, we treat each row, representing a concept-error pair at each threshold level, from Table 9, as an individual hypothesis. For example, the *error-margin* (Non-TLA) for the concept "central reservation" under the > 40 threshold constitutes one hypothesis. This way, we get $33 \times 4 = 132$ hypotheses to test.

We conduct Mann-Whitney U tests (MWU) [42] with the null hypothesis (H0) stating that there is no difference in Non-TLA across both datasets, while the alternative hypothesis (H1) posits that Non-TLA in Google Images is greater than in the ADE20K dataset. We choose the MWU test for its robustness with non-parametric data and its aptitude for comparing distributions of independent samples. As our Non-TLA data may not adhere to normality and we are comparing distinct datasets (Google Images and ADE20K), the MWU test provides a reliable means to analyze differences in Non-TLA.

Table 10 presents a comparison of Non-TLA between the Google Images and ADE20K datasets for all concepts. Each row represents a concept, with columns displaying the percentage of non-target label images activating associated neuron(s) in both datasets. The p-values from the MWU test indicate the significance of differences in Non-TLA between the datasets. The analysis reveals a consistent trend of decreased Non-TLA in the ADE20K dataset compared to Google Images across various threshold categories. Among the 33 hypotheses tested for the category of Non-TLA > 0, 13 were rejected at a significance level of p < 0.05. Similarly, for Non-TLA > 20%, 15 hypotheses were rejected at the same significance level. In the case of Non-TLA > 40%, 21 hypotheses were rejected, while for Non-TLA > 60%, 23 hypotheses were rejected, all at a p-value < 0.05. Concepts with p-value < 0.05 are deemed statistically significant and are identified as *confirmed* concepts, subject to further scrutiny for their reliability and potential implications.

After confirming concepts using the MWU, we proceed to validate them further using Wilcoxon signed-rank tests. To calculate the Wilcoxon test, we used an online website calculator called the Wilcoxon signed-rank test calculator by Statistics Kingdom 2017.[8] We employ the Wilcoxon test, with the hypothesis that the difference between Non-TLA of ADE20K and Google Image dataset would be less than or equal to zero (H0), while the alternative hypothesis (H1) suggested a decrease in Non-TLA in the ADE20K dataset compared to the Google image dataset. Each threshold serves as an individual hypothesis for the Wilcoxon test, with Non-TLA of the *confirmed* concepts for Google and ADE20K datasets grouped accordingly. For instance, all confirmed Non-TLA > 0 for

[8] http://www.statskingdom.com/170median_mann_whitney.html

Table 10

Statistical Evaluation for *confirmed* concepts (concepts getting *p*-value <0.05 for MWU): Non-t: percentage of non-target label images activating the associated neuron(s) analyzed across various activation thresholds.

Concepts	Google	ADE20K	p-values
	non-t >0		
building	43.37468	11.45833	0.018471
building and dome	78.185	26.16984	6.06E-05
central_reservation	84.97336	44.89338	1.75E-66
closet and air_conditioning	71.05416	30.16845	0.009373
edifice and skyscraper	48.76092	24.18726	0.016058
faucet and flusher	78.562	56.96671	9.19E-07
footboard and chain	66.702	27.97468	0.000284
lid and soap_dispenser	78.57143	57.51174	0.00218
pillow	46.49232	12.5	4.21E-23
potty and flusher	76.82974	58.41014	1.39E-07
shower_stall and cistern	53.1865	25.98152	0.016657
tap and crapper	70.60579	62.22494	6.17E-08
tap and shower_screen	72.584	62.62136	0.007024
Wilcoxon signed rank test (non-t >0)			**0.0001221**
	non-t >20 %		
building	12.31365	5.208333	1.72E-17
building and dome	45.13343	5.892548	1.37E-23
clamp_lamp and clamp	29.2287	19.16996	1.57E-07
closet and air_conditioning	38.4913	15.62021	0.000287
edifice and skyscraper	21.78641	8.452536	5.80E-17
faucet and flusher	37.86209	30.57953	1.80E-15
lid and soap_dispenser	34.98939	18.42723	2.74E-15
mountain and bushes	10.42437	16.57335	3.25E-06
pillow	9.634389	3.869048	3.49E-49
potty and flusher	36.53659	24.19355	3.69E-18
Shower_stall and screen_door	12.87584	14.3201	0.035051
skyscraper	0.872071	0.462963	1.99E-05
tap and crapper	36.83933	12.86064	0.000114
tap and shower_screen	32.5745	13.17961	3.22E-14
wardrobe and air_conditioning	31.79496	16.16022	2.18E-11
Wilcoxon signed rank test (non-t > 20%)			**0.0004272**
	non-t >40 %		
building	2.27609	1.458333	3.16E-19
building and dome	14.64338	0.866551	6.28E-20
central_reservation	19.73357	14.92705	1.18E-05
clamp_lamp and clamp	9.000096	8.300395	2.79E-31
closet and air_conditioning	10.1354	5.513017	6.38E-09
cross_walk	1.52392	9.78399	0.000572
edifice and skyscraper	8.37939	1.30039	5.06E-17
faucet and flusher	12.10377	11.09741	2.90E-24
field	8.183384	10.41215	3.82E-05
flusher and soap_dish	7.695067	3.896104	4.26E-08
lid and soap_dispenser	9.052334	2.816901	2.04E-19
mountain and bushes	4.666314	6.606943	1.28E-12
pillow	0.988239	1.190476	1.37E-23
potty and flusher	10.75519	4.608295	1.97E-09

road	3.471037	4.327666	0.033105
road and car	8.399088	6.074074	0.009958
Shower_stall and screen_door	4.120976	5.89651	1.13E-07
skyscraper	0.015367	0.231481	2.47E-30
slope	6.893903	9.879254	1.14E-07
tap and shower_screen	7.835857	4.73301	2.05E-12
wardrobe and air_conditioning	6.9579	5.524862	1.70E-19
Wilcoxon signed rank test (non-t > 40%)			**0.0479**
	non-t > 60%		
building	0.182087	0	1.08E-07
building and dome	2.639495	0	5.70E-10
central_reservation	2.912966	3.815937	1.50E-07
clamp_lamp and clamp	1.652099	1.976285	4.24E-19
closet and air_conditioning	1.266925	1.378254	2.50E-07
cross_walk	0.520833	2.92249	0.000171
edifice and skyscraper	2.228561	0.260078	4.80E-07
faucet and flusher	1.872623	1.849568	0.008524
field	1.655819	2.386117	1.43E-09
flusher and soap_dish	1.147982	0.236128	3.03E-13
lid and soap_dispenser	1.485149	0.352113	3.10E-07
mountain and bushes	1.936961	1.903695	9.96E-12
pillow	0.048848	0.14881	1.04E-09
potty and flusher	1.931664	1.152074	0.010232
road	0.701794	0.927357	0.000445
road and car	3.261441	1.333333	3.79E-05
route and car	1.415601	2.15208	0.000137
shower_stall and cistern	1.372089	1.039261	0.031085
Shower_stall and screen_door	1.025822	1.203369	9.36E-11
skyscraper	0	0.115741	6.15E-26
slope	1.200192	1.975851	2.39E-10
tap and shower_screen	0.859795	0.606796	3.67E-08
wardrobe and air_conditioning	1.144971	0.966851	1.52E-14
Wilcoxon signed rank test (non-t > 60%)			**0.05803**

both datasets constitute one hypothesis, while those > 20% form another. The p-values, denoting the significance of the test results, are displayed at the bottom of the table. Remarkably, the obtained p-values for each threshold suggest the rejection of the null hypothesis, indicating statistically significant differences in Non-TLA between the datasets when considered separately. A p-value < 0.05 from this test would indicate a statistically significant decrease in Non-TLA in the ADE20K dataset compared to the Google dataset, further strengthening our findings and highlighting that the error estimates from the Google image data hold, or are even bettered by, the ADE20K images.

We also examine all *confirmed* concepts from all thresholds together in the Wilcoxon test with the same alternative hypothesis ((H1) suggested a decrease in Non-TLA in the ADE20K dataset compared to the Google image dataset), which provides a comprehensive overview of the differences in Non-TLA between the Google and ADE20K datasets across various levels of activation thresholds. This approach aggregates the results from individual thresholds, offering a more consolidated perspective on the overall significance of the differences observed. In our analysis, obtaining a p-value of 5.633e-7, which is less than 0.05, implies the rejection of the null hypothesis. This indicates a statistically significant decrease in Non-TLA in the ADE20K dataset compared to the Google Image dataset when considering all thresholds collectively.

Table 11

Count of statistically confirmed Concepts from each method (Table 11) such that their percentage of target activation is binned into 3 regions based on their degree of relevance.

Method	90-100%	80-89%	<80%
Concept Induction	14	6	0
GPT-4	10	4	0
CLIP-Dissect	4	1	0

Table 12

Mean, Median, and Standard Deviation (SD) of Concept Activation Analysis Test Accuracies, and Count of Concepts with their Concept Classifier Test Accuracies binned into 3 regions – High (90-100%), Medium (80-89%), and Low (<80%) relevance

Method	CAV			CAR			Count of Concepts		
	Mean	Median	SD	Mean	Median	SD	90-100%	80-89%	<80%
Concept Induction	0.9154	0.9230	0.0449	0.9150	0.9310	0.0465	46	22	1
CLIP-Dissect	0.9160	0.9146	0.0389	0.9259	0.9293	0.0443	17	5	0
GPT-4	0.8757	0.8863	0.0817	0.8887	0.9024	0.0690	11	9	1

4.5. Further Discussion

From the statistical evaluation, based on the percentage of target activation and from Concept Activation Analysis, based on the concepts' test accuracies, we can categorize all confirmed concepts into three regions: high (90-100%), medium (80-89%), and low ($< 80\%$) relevance concepts. Tables 11 and 12 show that Concept Induction produces a notably larger number of high-relevance concepts compared to other methods. Table 6, shows 8 and 27 statistically confirmed concepts from the CLIP-Dissect and GPT-4 method, respectively. However, upon closer examination, it becomes evident that some concepts are duplicated across the tables.

Disregarding the duplicates, we have only 5 and 14 confirmed concepts from CLIP-Dissect or GPT-4, respectively, as opposed to 18 from Concept Induction.

This difference is likely due to Concept Induction's reliance on rich background knowledge, necessitating additional preprocessing but offering additional value. While a candidate concept pool of 20K English vocabulary words for off-the-shelf GPT-4 may not be universally effective, Concept Induction's ability to generate extensive, high-relevance concepts underscores the importance of well-engineered background knowledge.

If an application does not require comprehensive concept-based explanations, CLIP-Dissect/GPT-4 may serve as a useful solution, especially when time is limited. However, for detailed concept-based analysis, preparing background knowledge and leveraging Concept Induction is crucial. For CLIP-Dissect/GPT-4, it is unclear how to meticulously craft the pool of candidate concepts since it is difficult to manually curate a static set that is broad enough to capture all pertinent concepts while remaining specific enough to avoid noisy or ambiguous labels. By employing a background knowledge base, it is possible to define a large pool of potential explanations, tailored to the application scenario, with additional relationships among concepts. For example, in a medical diagnostic application, an ideal candidate pool would include specialized clinical terminology (e.g., "cardiomegaly" or "pleural effusion") that is essential for accurate interpretation – an adjustment that is hard to achieve with a generic vocabulary. Concept Induction facilitates deductive reasoning utilizing this background knowledge, inherently offering transparency and flexibility in shaping the candidate concept pool.

While it is important to investigate methods that assess the relevance of concepts in hidden layer computations within a given candidate pool, it is equally, if not more, vital to thoughtfully design this pool. Neglecting this aspect could result in — (a) missing domain-critical concepts essential for gaining insights into hidden layer computations and (b) introducing noisy or ambiguous concepts that can lead to spurious activations and misleading explanations. Our ontology-driven approach mitigates both risks by integrating rich background knowledge and extract meaningful concepts from it.

Our focus on dense layer activations, while providing valuable insights, represents only a part of what the deep representation encodes. The dense layer likely relates to clear-cut concepts that separate output classes, aligning well with our goal of identifying high-level, interpretable concepts. However, these concepts are influenced by combinations of features from previous layers. This limitation underscores the complex nature of deep neural networks, where concepts identified at the dense layer result from hierarchical feature compositions throughout the network. While our method offers meaningful insights into these high-level concepts, it may not fully capture the nuanced feature interactions in earlier layers. Nonetheless, focusing on the dense layer allows us to extract concepts more directly relevant to the network's final decision-making process, balancing interpretability with the complexity of internal representations. Future work could explore extending our method to analyze concept formation across multiple layers, potentially revealing a more comprehensive picture of the network's decision-making process.

One drawback of utilizing Concept Induction (and GPT-4) is its dependency on object annotations, which serve as data points in the background knowledge. In contrast, CLIP-Dissect operates without the need for labels and can function with any provided set of images.

We view this as a trade-off that must be carefully considered based on the application scenario. If the application is broad and does not demand a meticulous design of candidate concepts, then employing approaches like CLIP-Dissect can be advantageous. Conversely, for applications that are focused or specialized, CLIP-Dissect may only provide broadly relevant concepts.

Our focus has been primarily on assessing the comparative effectiveness of Concept Induction within the confines of Convolutional Neural Network architecture using ADE20K Image data. Nevertheless, it is imperative to investigate its suitability across different architectures and with diverse datasets. Given the model-agnostic nature of our approach, our results suggest its potential applicability across a range of neural network architectures, datasets, and modalities. While we utilized a Wikipedia Concept Hierarchy comprising 2 million concepts, it would be intriguing to observe the outcomes of our approach when powered by a domain-specific Knowledge Graph in specialized domains such as Medical Diagnosis.

The error margins derived from our analysis significantly enhance the interpretability and reliability of neural networks. These margins provide a quantitative measure of confidence for concept detection in image analysis tasks. For instance, when a neuron associated with a specific concept (e.g., "buffet") activates above a certain threshold, the error margin allows us to estimate the likelihood that the image actually contains that concept.

Our study demonstrates the robustness of error margin methodology across diverse datasets without assuming identical neuron-concept associations between Google Images and ADE20K. Instead, our primary goal was to validate the generalizability of error margins across these distinct datasets. In our experiments, we observed varying neuron-concept associations across datasets. For instance, while neuron 62 prominently associated with 'buffet' in Google Images, its activation pattern in ADE20K showed similarities but also notable differences. These variations stem from differences in dataset characteristics, training specificity, and concept granularity. Importantly, these differences strengthen our findings. The methodology's ability to produce statistically significant results despite these variations underscores its robustness and broad applicability. This adaptability is crucial for real-world applications requiring reliable concept detection and interpretability across diverse data contexts.

Our statistical analysis, employing Mann-Whitney U and Wilcoxon signed-rank tests, reveals significant differences in non-target label activations (Non-TLA) between the ADE20K dataset and the Google Images dataset. Crucially, the lower Non-TLA values observed in the ADE20K dataset validate our error margins and underscore their reliability. This validation is important for several reasons:

- **Generalizability**: The fact that error margins derived from the Google Images dataset generalize well to the more structured and annotated ADE20K dataset indicates that our method is not confined to a specific dataset. This enhances the broader applicability of our approach.
- **Reliability**: The reduced Non-TLA in the ADE20K dataset suggests that neuron activation patterns are more precise and reliable when tested on a well-annotated dataset. This finding assures that the calculated error margins are robust and can be trusted for practical AI applications.
- **Foundation for Future Work**: Validating our error margins across different datasets provides a strong foundation for future research, encouraging further exploration of neuron activation patterns and their implications for model explainability.

These error margins significantly enhance the interpretability of neural network decisions by quantifying the reliability of neuron-concept associations. This offers a more nuanced understanding of how the network processes information, going beyond simple neuron labeling to provide insights into the degree of certainty with which we can interpret a neuron's activation. Such information is crucial for building trust in AI systems, especially in critical decision-making scenarios.

Furthermore, our error margin analysis can guide the refinement of neural architectures. By identifying neurons or neuron ensembles with high precision for specific concepts, we can inform targeted improvements in network design. For example, architectures could be optimized to enhance the precision of key concept detections, potentially leading to more efficient and accurate models.

In summary, our analysis demonstrates that the concept associations and error margins derived from our method are both reliable and generalizable. These findings contribute significantly to the field of explainable AI by providing a validated approach to understanding and improving the interpretability of neural networks, paving the way for more advanced and trustworthy AI systems.

5. A Special Study: Concept Induction using LLM

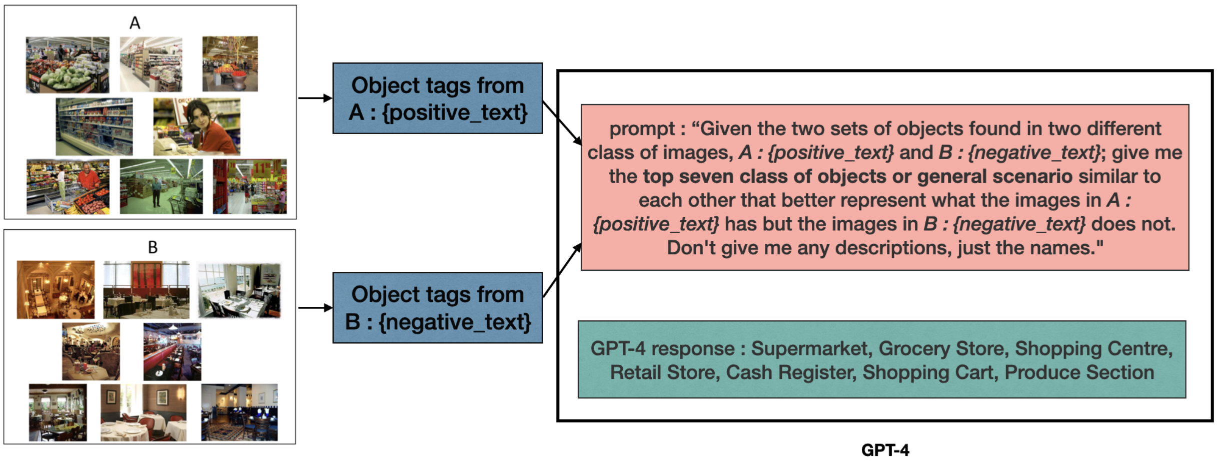
We explore the potential of a Large Language Model (LLM), specifically GPT-4, by leveraging its domain knowledge and common-sense capability, to generate high-level concepts that are meaningful as explanations for humans, for our specific setting of image classification. We use minimal textual object information available in the data via prompting to facilitate this process. To evaluate the output, we compare the concepts generated by the LLM with two other methods: concepts generated by humans and the ECII heuristic concept induction system. Since there is no established metric to determine the human understandability of concepts, we conducted a human study to assess the effectiveness of the LLM-generated concepts. Our findings indicate that while human-generated explanations remain superior, concepts derived from GPT-4 are more comprehensible to humans compared to those generated by ECII. The prompting approach we detail and evaluate below was also used for the GPT-4 based label hypothesis generation described in section 4.2.3.

Expanding upon the framework introduced in Section 4, our goal is to explore the feasibility of replacing the ECII model with a Large Language Model (LLM) to produce explanations that remain meaningful and coherent. The objective is to identify "good" concepts that make sense to humans and can later be validated by mapping them with a Deep Neural Network (DNN) to accurately describe what neurons perceive. We utilized the GPT-4 [49] model to generate meaningful explanations for a specific scene classification task, which was done using a logistic regression algorithm that classified images into scene categories based on semantic tags of objects present in each image. The explanations are generated using Prompt Engineering [18] via the OpenAI API. Unlike logical-deduction-based systems such as ECII, which are limited by background knowledge, an LLM like GPT-4 can leverage its common-sense reasoning capability and vast domain knowledge to produce more comprehensive concepts. In [70], the quality of explanations generated by concept induction was assessed and found to be more meaningful than semi-random explanations but less accurate than human-generated (gold standard) ones. Our objective is to evaluate the extent to which explanations generated by LLMs align with human-generated explanations and potentially surpass the concept induction system in terms of accuracy and comprehensibility.

As discussed before, concept induction is a symbolic reasoning task that can be done using provably correct [37] or heuristic [55] deduction algorithms over description logic knowledge bases. In this section, we attempt to make use of pre-trained LLMs to produce results that are comparable to or even better than those obtained from a concept induction system. In other words, we are making use of an LLM to do better than a symbolic-reasoning-based algorithm, at least in a specific setting.

5.1. Approach

Our approach and evaluation setting is essentially the same as in [70], however instead of their comparison of explanations generated by (1) humans, (2) concept induction, and (3) a semi-random process, we compare (1) human, (2) concept induction, and (3) GPT-4 prompting. We went into the study with the hypothesis that explanations produced by GPT-4 would outperform those produced by concept induction in terms of "meaningfulness to humans," but that they would still not be as good as the human-generated gold standard.


A
B
Object tags from A : {positive_text}
Object tags from B : {negative_text}
GPT-4

prompt : "Given the two sets of objects found in two different class of images, *A : {positive_text}* and *B : {negative_text}*; give me the top seven class of objects or general scenario similar to each other that better represent what the images in *A : {positive_text}* has but the images in *B : {negative_text}* does not. Don't give me any descriptions, just the names."

GPT-4 response : Supermarket, Grocery Store, Shopping Centre, Retail Store, Cash Register, Shopping Cart, Produce Section

Fig. 5. Prompting Method: The GPT-4 model was prompted using the exact prompt mentioned in the image. Here, the positive and negative text indicates the object tags present in the images. The resulting set of seven concepts is mentioned in the GPT-4 response.

5.1.1. *Input Dataset*

As in [70], we used the object tags associated with images from the ADE20K dataset [74, 76] as input, in this case for the GPT-4 model via the OpenAI API. As discussed previously, this dataset contains approximately 20,000 human-curated images annotated with scene categories and object tags present in the images. We used a selection of 45 image set pairs. Each image set pair consists of two groups of natural images representing distinct scene categories (A and B), with a total of 90 scene categories across all sets. Each set within a pair consisted of eight images selected at random from a particular category.

These image set pairs were curated in the previous study [70], and we adopted the same set of pairs to maintain consistency. Although the object tags in the dataset indicate not only the presence of an object but also details such as the number of objects and occlusions, we focused solely on the object labels for our analysis, disregarding additional annotations.

To generate explanations from the GPT-4 model, we fed the object tags of the images into the model using prompts. Our objective was to describe what distinguished Category A from Category B in each image set pair, where each image set belongs to a specific scene category. These descriptions were defined as "concepts," and for each image set, we produced a list of seven concepts. We tried to come up with concepts that encompass tangible objects depicted in the images (e.g., tree or bench) or general categories that align with the theme of the images (e.g., park or garden).

To prompt the GPT-4 model effectively, we experimented with different prompting techniques to obtain the most reasonable concepts. Our approach involved using a straightforward technique that leveraged only the object labels from each image set category. We instructed the GPT-4 model to differentiate between the two categories based on their object tags. Object tags, as the name suggests, could be anything physically present in the images. For example, the object tags coming from category A in Figure 5 include object labels such as stands, food, wall, tomatoes, bag, register, weighing machine, shopping carts, person, etc.

Similarly, the ECII system also used the same object tags to generate concepts. For the ECII model, all object tags from the images are automatically mapped to classes in the Wikipedia class hierarchy using the Levenshtein string similarity metric [39] with an edit distance of 0. The algorithm then assessed the images based on their object tags and returned a rating of how well concepts matched images in Category A but not Category B. ECII explanations were then created by taking the seven highest-rated unique concepts. This alignment allowed us to compare the concepts generated by our approach with those produced by the ECII system.

The process and the prompt used for interacting with the GPT-4 model are illustrated in Figure 5.

5.1.2. Prompting the model

We used the latest version of the GPT-4 model for our prompt. We utilized zero-shot prompting with specific parameters, setting the temperature to 0.5 and top_p to 1. The temperature parameter in GPT-4 controls the level of creativity or randomness in the generated text. When predicting the next token from a vocabulary of size N, the model uses a softmax distribution of the form $\text{softmax}(x_i/T)$ for $i = 1, \ldots, N$, where T is the temperature. This distribution assigns probabilities to each token (x_i) in the vocabulary, influencing the likelihood of selecting each word. Lowering the temperature favors words with higher probabilities, leading to more predictable and less creative responses when the model randomly samples the next word. Top_p sampling is an alternative to temperature sampling. It limits the consideration from all possible tokens to a subset of tokens (the nucleus) whose cumulative probability mass reaches a specified threshold (top_p). OpenAI recommends adjusting one of these parameters but not both simultaneously for optimal control over text generation. In our prompts, we set the model's temperature to a lower value (0.5) to ensure more consistent and reproducible answers across different sets. Here, we didn't set the temperature to 0 as we wanted to see some creative responses from the GPT-4 model in tasks where the image set categories (e.g., Category A and B) contain similar objects, to test if the model can distinguish them using human-like intuitive behavior. In figure 5, we can see that all the object tags coming from sets A and B are given in the prompt, and it was asked to distinguish between them. Here as it becomes a long prompt with all the object tags for both categories, we mention them twice in our prompt, once at the beginning and once at the end, which seems to be helpful for the GPT model to produce better results and remember the object tags. In our prompts, we aimed to generate generic concepts or object classes that mimic the ontology classes positioned somewhere in the middle of the hierarchy used by ECII. These intermediate classes are designed to capture a broader range of specific child classes, providing a bridge between more general concepts and highly specific subclasses within the ontology structure. It is asked to provide the top seven concepts based on the instruction. We generate a list of seven concepts for each set following this method.

5.2. Evaluation

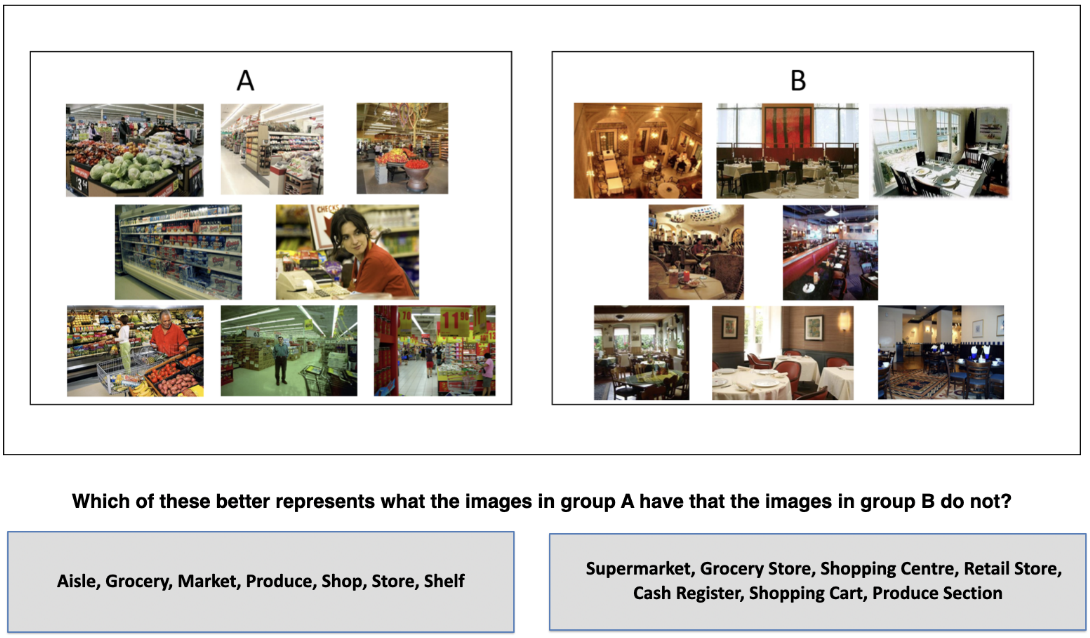
To evaluate the concepts generated from GPT-4 we ran a study through Amazon Mechanical Turk using the Cloud Research platform. Our goal was to assess the quality of LLM explanations (i.e., GPT-4 explanations) compared to both human-generated ("gold standard") explanations and ECII explanations.

We recruited 300 participants through Mechanical Turk, compensating each participant with $5 for completing the task, which was estimated to take approximately 40 minutes (equivalent to $7.50 per hour based on the minimum legal wage in the USA). Based on the previous study [70], we aimed for a sample size of at least 89 unique participant judgments per trial to estimate the parameters (medium effect size of f2 = 0.15 and 95% power) of the Bradley-Terry model [9], which is used to evaluate the survey results. This required collecting data from 300 participants, resulting in a total of 100 observations per trial after accounting for potential exclusions.

Across all questions, each participant encountered three types of explanations, although only two explanation types were compared in any given question. Each participant was asked to choose the more accurate explanation using a two-alternative forced choice design. For each pair of image sets, participants answered three questions comparing (1) Human versus ECII explanation; (2) Human versus LLM (GPT-4) explanation; and (3) LLM versus ECII explanation. For each pair of image sets (A and B), a given participant completed all three comparisons.

The 45 pairs of image sets in this study resulted in a total of 135 unique target questions. Participants were randomly assigned to 15 image sets (45 questions in total), ensuring that image sets were counterbalanced across participants to receive an equal number of responses.

For all image sets, ECII explanations and Human "gold standard" explanations were created in a previous study [70]. In this work, we generated LLM (GPT-4) explanations following the method described in Section 5.1. To form the ECII explanations, the object tags of the images were provided to the ECII algorithm, then the seven highest-rated unique concepts were selected based on the ranking of the F1 score. Human "gold standard" explanations were crafted by presenting image sets (without object or scene category tags) to three human raters, selecting concepts unanimously mentioned by all three, then by two raters, and finally filling in randomly selected concepts until seven unique concepts were reached.



Which of these better represents what the images in group A have that the images in group B do not?

Aisle, Grocery, Market, Produce, Shop, Store, Shelf

Supermarket, Grocery Store, Shopping Centre, Retail Store, Cash Register, Shopping Cart, Produce Section

Fig. 6. Survey interface, with human explanation presented on the left and LLM explanation on the right.

In addition to the 45 image sets, five "catch trial" image sets were used to verify participant attention. These catch-trial image sets included two types of explanations: human explanations generated similarly to other human gold standard explanations, and explanations consisting of completely random concepts generated from a word generator to serve as obviously inaccurate explanations.

After providing consent, participants received brief training on the task, including instructions on how concepts and explanations were defined in the study. They then began answering questions, with the 50 questions (45 assigned targets and 5 catch trials) presented in random order. Figure 6 illustrates the stimuli presentation and response options shown to participants.

5.3. Results

Prior to analysis, participant responses to catch trials were evaluated, and participants who failed more than one catch trial were excluded from further analysis. Among the 300 participants, 253 did not fail any catch trials, while 22 participants failed exactly one trial, and 35 participants failed more than one trial. The 35 participants who failed multiple trials were excluded from all subsequent analyses, resulting in a total of 265 participants included in the analyses.

Across all image sets, human explanations were overwhelmingly preferred over ECII explanations (chosen 3282 times versus 693 times; 83% preference) and over LLM (GPT-4) explanations (chosen 2762 times versus 1213 times; 69% preference). Additionally, LLM explanations were preferred over ECII explanations (chosen 2514 times compared to 1461 times; 63% preference). See Figure 7.

Participants' pairwise judgments were utilized in a Bradley-Terry analysis [67] to derive "ability scores" for each type of explanation, reflecting the extent to which each explanation type was preferred by participants. The Bradley-Terry model uses data where entities are compared pairwise, and the outcome (win/loss, preference ranking, etc.) is observed. From these comparisons, the model estimates the abilities θ_i such that the observed outcomes are statistically likely. The estimation process typically involves fitting the model to the pairwise comparison data to find the best-fitting values of θ_i for each entity. These estimates reflect the latent abilities or strengths of the entities relative to each other. Ability scores were calculated for each of the 45 image set pairs based on the pairwise

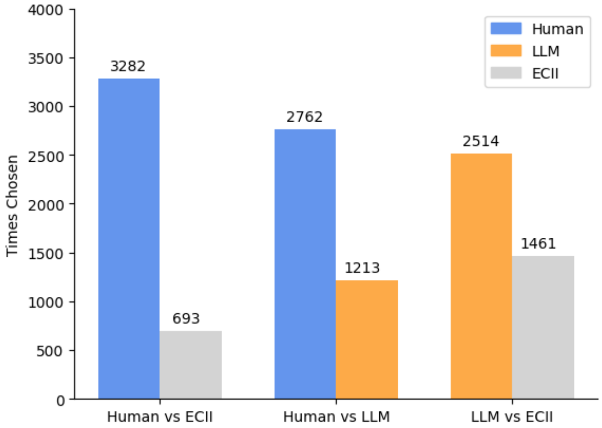


Fig. 7. Number of times participants chose different explanation types.

Table 13

p-values of the ability scores among different explanation types from Tukey's HSD Test

Comparison pairs	p-value
Human_explanation vs ECII_explanation	<0.0001
LLM_explanation vs ECII_explanation	=0.0004
Human_explanation vs LLM_explanation	<0.0001

comparison data (win/loss) for each type of explanation. The analysis of these ability scores demonstrated that human explanations had the highest scores (M = 1.77, SD = 0.978), followed by LLM explanations (M = 0.724, SD = 1.16), with a significant overall difference (F(2) = 46.28, $p < 0.001$, $\eta^2 = 0.41$). Here, ECII explanations served as the reference point and were set to 0, with the ability scores for human and LLM explanations indicating their preference over ECII explanations.

A post hoc analysis using Tukey's Honestly Significant Difference (HSD) test [2] was conducted to determine which specific group means are significantly different from each other. When comparing multiple group means, the Tukey post hoc test is preferred over multiple t-tests [32] because it adjusts for multiple comparisons, controlling the overall Type I error rate [44]. Conducting multiple t-tests increases the risk of false positives, while the Tukey test maintains the integrity of statistical conclusions by adjusting the significance levels appropriately. This test confirmed significant differences in ability scores between human vs. ECII explanations and human vs. LLM explanations (both $p < 0.0001$), as well as between LLM vs. ECII explanations ($p = 0.0004$) (Table 13). These low p-values indicate that the observed differences in ability scores are highly significant and unlikely to have occurred by random chance alone.

The individual ability scores for human and LLM explanations for each image set pair are detailed in Table 14.

The source code, input data, and raw result files related to the evaluation tasks (i.e., survey questionnaires, and collected responses) are available online.[9]

5.4. Discussion

The analysis of the results presented in Table 14 provides evidence supporting our hypothesis that LLM (GPT-4) explanations are more meaningful for humans compared to ECII-generated ones. Human-generated explanations were consistently preferred as the most accurate in describing differences between image categories, followed by

[9] https://github.com/AdritaBarua/Concept-Induction-using-LLMs-a-user-experiment-for-assessment

Table 14

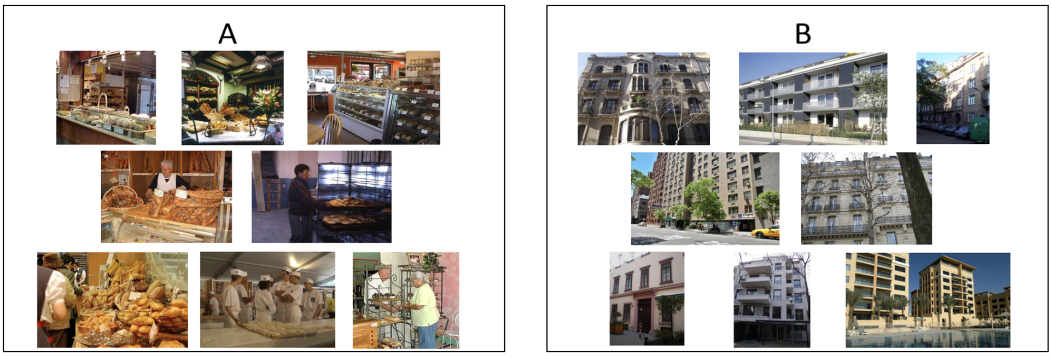
Ability Scores and Number of Wins for Human (H), ECII (E), and LLM (L) explanations. ECII explanations were set as the reference point in the Bradley-Terry analysis and so their ability scores were always equal to 0, and thus are not displayed here.

Image Set	H.Ability	LLM.Ability	HvE Wins	HvL Wins	LvE Wins
Set 1: Bedroom v Park	1.47	-1.05	72-12	74-10	18-66
Set 2: Living Room v Parking Lot	2.64	2.76	84-3	38-49	79-8
Set 3: Office v Playground	1.12	0.34	74-15	54-35	45-44
Set 4: Airport v Amusement Park	1.93	0.77	77-13	70-20	63-27
Set 5: Bathroom v Art Studio	1.05	1.47	67-20	32-55	68-19
Set 6: Beauty Salon v Forest Path	0.72	-0.86	63-25	69-19	22-66
Set 7: Bookstore v Child Room	1.72	1.79	76-15	45-46	79-12
Set 8: Hotel Room v Cockpit	0.65	-1.68	62-28	79-11	11-79
Set 9: Shoe Store v Alcove	0.79	1.52	63-24	25-62	68-19
Set 10: Alley v Wet Bar	2.74	1.85	85-6	65-26	79-12
Set 11: Closet v Construction Site	1.98	1.14	77-8	57-28	62-23
Set 12: Gazebo v Bowling Alley	2.64	-1.03	85-2	81-6	19-68
Set 13: Garage v Hallway	0.42	-0.09	49-39	59-29	46-42
Set 14: Laundromat v Pantry	1.86	1.18	75-14	61-28	70-19
Set 15: Conference Room v Waterfall	2.42	-0.45	85-3	79-9	30-58
Set 16: Home Office v Bow	1.83	1.58	77-13	51-39	75-15
Set 17: Dining Room v Kitchen	0.24	0.33	45-41	44-42	53-33
Set 18: Fast Food v Office Building	2.58	0.24	84-4	78-10	47-41
Set 19: Jacuzzi v Greenhouse	3.08	2.13	88-5	68-25	84-9
Set 20: Gymnasium v Corridor	2.76	1.63	83-6	68-21	75-14
Set 21: Bus v Broadleaf Forest	2.24	-0.59	77-8	80-5	30-55
Set 22: Casino v Arrival Gate	1.77	1.11	73-13	57-29	65-21
Set 23: Library v Gas Station	0.92	-1.02	61-31	85-7	29-63
Set 24: Valley v Yard	2.66	1.17	85-7	76-16	71-21
Set 25: Mountain v Coast	0.45	-0.64	50-36	67-19	32-54
Set 26: Dinette Vehicle v Farm Field	0.88	-0.62	69-23	71-21	28-64
Set 27: Poolroom v Driveway	-0.72	-0.12	30-58	30-58	40-48
Set 28: Bridge v Auditorium	1.95	1.9	80-10	45-45	77-13
Set 29: Museum v Youth Hostel	1.24	-1.04	68-20	80-8	23-65
Set 30: Supermarket v Restaurant	2.12	2.97	75-8	24-59	78-5
Set 31: Classroom v Archive	1.18	0.06	65-18	61-22	41-42
Set 32: Dentist Office v Ballroom	2.94	1.29	85-5	76-14	71-19
Set 33: Lighthouse v River	1.68	1.81	73-14	41-46	75-12
Set 34: Creek v Basement	4.46	2.85	86-4	78-12	88-2
Set 35: Building Facade v Ocean	1.69	0.77	77-16	68-25	65-28
Set 36: Courthouse v Parking Garage	2.95	1.15	82-7	79-10	70-19
Set 37: Balcony v Skyscraper	3.18	0.8	84-4	81-7	61-27
Set 38: Game Room v Waiting Room	0.68	0.09	63-29	57-35	46-46
Set 39: Landing Deck v Window Seat	2.72	2.15	86-4	56-34	79-11
Set 40: Bar v Warehouse	1.35	0.47	73-15	59-29	51-37
Set 41: Bakery v Apartment Building	0.99	1.98	63-21	21-63	72-12
Set 42: Needleleaf Forest v Playroom	2.41	1.14	81-8	70-19	68-21
Set 43: Outdoor Window v Roundabout	2.14	0.53	84-8	75-17	56-36
Set 44: Reception v Golf Course	2.16	0.99	76-9	65-20	62-23
Set 45: Staircase v Plaza	1.09	0.04	65-21	63-23	43-43

LLM explanations, with ECII explanations found as the least accurate. The preference for human-generated explanations over LLM explanations is expected given the messy nature of generalized Large Language Models. These models, trained on vast and diverse datasets, can produce responses that lack precision and clarity because of their broad generalization. This can lead to explanations that are sometimes inaccurate or unclear, making human-generated explanations generally more reliable and preferred. Also, there is potential for further refinement in prompting techniques using varied hyper-parameters (e.g., temperature and top-p). However, LLM explanations demonstrated notable explanatory power, suggesting their usability in concept generation.

It is important to note the variability in LLM performance across different image sets. In some cases, LLM explanations were chosen relatively more frequently than in others, with some instances showing LLM explanations being preferred more often than human explanations. Conversely, in other image sets, LLM explanations were chosen less often than ECII explanations. For instance, in Set 41 (see Figure 8), explanations generated by LLM are more comprehensive in identifying images of a bakery, while human-generated explanations also perform adequately. However, the concept "Women" included in the human-generated list is not as relevant for capturing the overall scene depicted in these images.

Example (Set - 41):



Which of these better represents what the images in group A have that the images in group B do not?

LLM Explanation:
- Bakery Shop, Pastry Display, Bread Shelf, Indoor Scene, Cake Counter, Bread Basket, Food Showcase

Human Explanation:
- Bake, Bakery, Bread, Indoor, Product, Store, Woman

ECII Explanation:
- Basket, Bread, Cake, Ceiling, Floor, Person, Wall

Fig. 8. Example of different explanation types for Set 41: Bakery v Apartment Building

On the other hand, ECII concepts only identify the object names present in the images and fail to capture the broader category of the scenes (i.e., bakery). In most cases where LLM explanations fall short, they tend to introduce concepts that are unrelated to the images. For example, in Set 6 (see Figure 9), LLM produced a concept like "Public Transport," which is contextually incorrect. One potential reason for this is the presence of object names (such as streetcar, tram, tramcar, swivel chair, trolley car, armchair) in the input images, which could be erroneously associated with public transport. Based on these examples, it is speculated that when GPT-4 was prompted to generate generic scenarios based on object tags, it attempted to produce seven distinct concepts. Limiting the number of concepts might lead to clearer explanations that are more pertinent. Additionally, running prompts to ask for simple object names rather than generic scenarios akin to ECII-generated explanations could yield different outputs that may prove useful. This suggests there is certainly still room for improvement in LLM explanations, but that on average there is promising evidence that LLMs can produce explanations that successfully describe the differences

Example (Set - 6):

Which of these better represents what the images in group A have that the images in group B do not?

LLM Explanation:

- Interior Decor, Furniture, Lighting Fixtures, Home Appliances, Building Structures, Personal Care Items, Public Transport

Human Explanation:

- Business, Chair, Indoor, Light fixture, Mirror, Salon, Sink

ECII Explanation:

- Ceiling, Floor, Hair dryer, Mirror, Sink, Swivel chair, Wall

Fig. 9. Example of different explanation types for Set 6: Beauty Salon v Forest Path

between two groups of data. Moreover, variability could be introduced by human participants. In our study, human explanations were preferred over ECII 83% of the time, whereas in the previous study [70] with the same settings, the preference ratio was 87%.

6. Tool: End-to-End Automated Neuron Interpretation using Concept Induction

In this section, we report on an implementation that end-to-end automates the concept induction and statistical evaluation workflow. Our system uses automation in four stages (see Figure 10) to streamline processes and to enhance efficiency.

Stage 1: Model Training and Data Configuration Initially, our automation pipeline trains and configures a CNN model using the ADE20K dataset [76]. This process is executed on Beocat [28], a high-performance computing environment optimized for managing extensive datasets. A Bash script automates job scheduling, resource allocation via SLURM, initializes the Python environment, securely clones the stage 1 repository from GitHub, and installs the necessary dependencies to establish the training environment. We employ a ResNet50V2 architecture implemented in TensorFlow, fine-tuned to enhance model performance using techniques such as data augmentation, early stopping, and batch normalization.

Stage 2: Parallelized Concept Induction and Label Hypothesis Generation We used the concept induction process to generate label hypotheses for each of the 64 neuron activations in the CNN's dense layer using the heuristic Concept Induction system ECII [55]. We automated the simultaneous execution of tasks for all 64 neurons by employing parallel processing with a SLURM-configured Bash script in Beocat. The script initializes the environment, installs necessary Java and Maven dependencies, and clones the latest stage 2 repository from GitHub. Each neuron-specific configuration file from Stage 1 was used to generate semantic concepts, producing output concept files with hypothesized labels and coverage scores using the background knowledge base from the Wikipedia concept hierarchy.

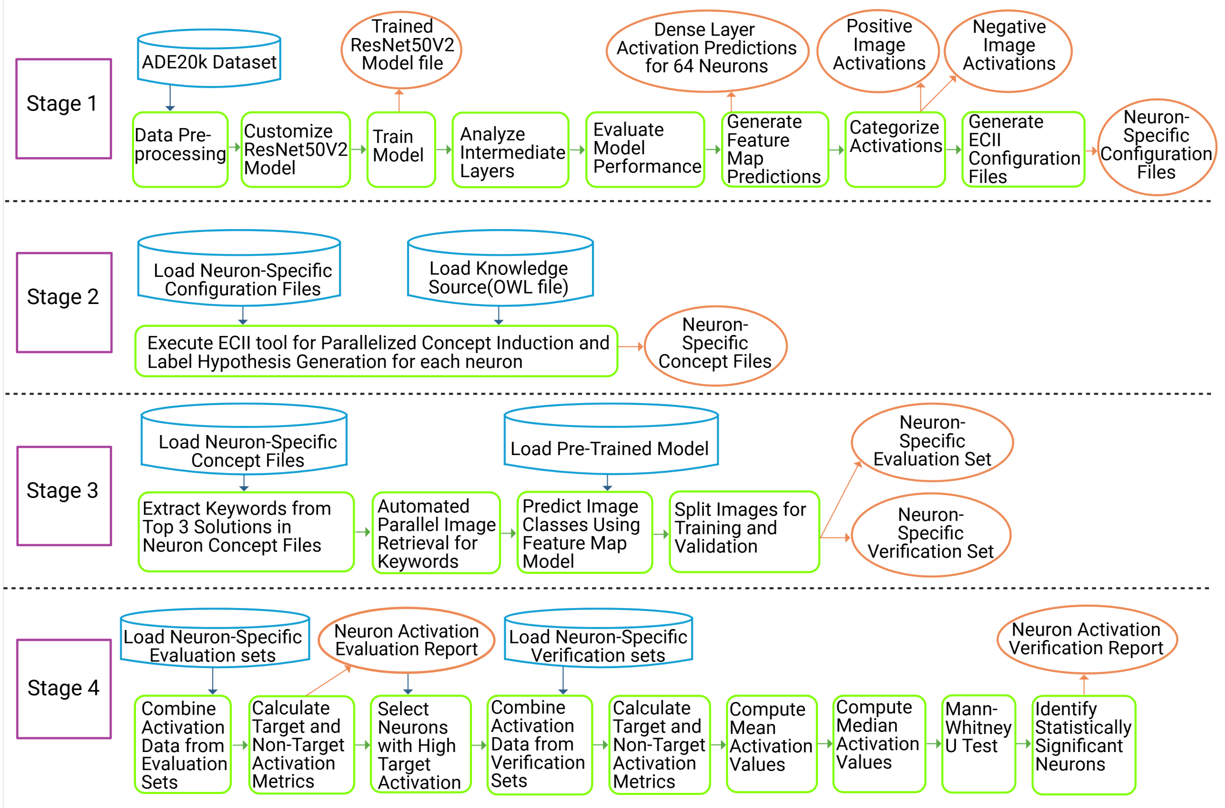

Stage 1
ADE20k Dataset
Data Pre-processing
Customize ResNet50V2 Model
Trained ResNet50V2 Model file
Train Model
Analyze Intermediate Layers
Evaluate Model Performance
Dense Layer Activation Predictions for 64 Neurons
Generate Feature Map Predictions
Positive Image Activations
Categorize Activations
Negative Image Activations
Generate ECII Configuration Files
Neuron-Specific Configuration Files
Stage 2
Load Neuron-Specific Configuration Files
Load Knowledge Source(OWL file)
Execute ECII tool for Parallelized Concept Induction and Label Hypothesis Generation for each neuron
Neuron-Specific Concept Files
Stage 3
Load Neuron-Specific Concept Files
Load Pre-Trained Model
Extract Keywords from Top 3 Solutions in Neuron Concept Files
Automated Parallel Image Retrieval for Keywords
Predict Image Classes Using Feature Map Model
Split Images for Training and Validation
Neuron-Specific Evaluation Set
Neuron-Specific Verification Set
Stage 4
Load Neuron-Specific Evaluation sets
Neuron Activation Evaluation Report
Load Neuron-Specific Verification sets
Neuron Activation Verification Report
Combine Activation Data from Evaluation Sets
Calculate Target and Non-Target Activation Metrics
Select Neurons with High Target Activation
Combine Activation Data from Verification Sets
Calculate Target and Non-Target Activation Metrics
Compute Mean Activation Values
Compute Median Activation Values
Mann-Whitney U Test
Identify Statistically Significant Neurons

Fig. 10. Automated four-stage pipeline for analyzing neuron activations, inducing concepts, and evaluating neuron significance using a ResNet50V2 model and ECII tool, created with BioRender.com.

Stage 3: Parallelized Image Retrieval and Classification Image retrieval and classification were automated for all neurons to validate the label hypotheses generated in Stage 2. A Bash script manages parallel task execution using SLURM, generating indices for neurons with configuration files. It clones the Stage 3 project repository, sets up the environment, installs dependencies. The script runs a Python program that utilizes the pygoogle_image library to extract labels from the top 3 solutions for each neuron, retrieves 100 images per label from Google, and classifies them using the trained CNN model. Retrieved images are divided into evaluation and verification sets for statistical analysis.

Stage 4: Statistical Analysis and Verification of Neuron Activations Label hypotheses are validated through statistical analysis of neuron activations. A Bash script sets up the environment, clones the stage 4 repository, and installs dependencies. The script runs a Python program that combines activation data from evaluation and verification sets, generates summary statistics, and conducts a Mann-Whitney U test [42] to compare activation values for target and non-target images. Evaluation sets, containing images that strongly activate neurons, provide initial activation metrics. Verification sets undergo further statistical testing to confirm the accuracy and robustness of the label hypotheses.

7. Tool: Demo

We provide a visualizing tool, *ConceptLens*, that quantifies the uncertainty and imprecision in neural concept labels through error margins. *ConceptLens* makes use of, and displays, identified concepts (using the Concept Induction approach) as well as confidence values based on the above discussed error-margin analysis. This approach

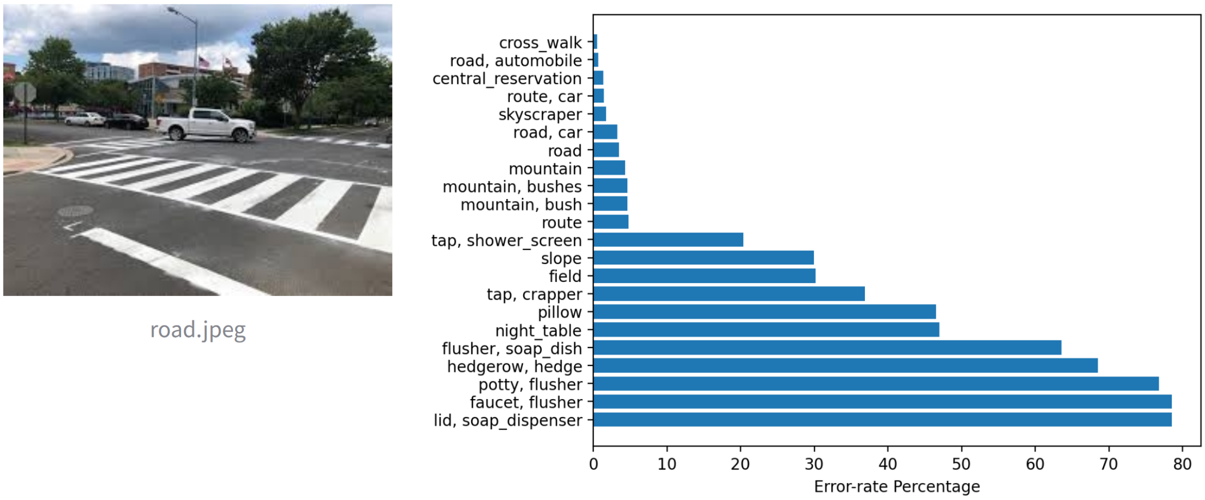


Fig. 11. Example image and output, ConceptLens demonstrator.

allows users to see not only what stimuli activate specific neurons but also how confidently these neurons respond to different inputs. The demonstrator is currently restricted to the system trained as described in section 4.1.

User Interface *ConceptLens* features a user-friendly interface that allows users to upload images and receive real-time visualizations of neuron activations. The main components of the interface include:

1. *Image Upload and Selection:* Users can upload their images or choose from a curated gallery. The tool supports a wide range of images, although results may vary for images outside the 10 classes it was primarily trained on: bathroom, bedroom, building facade, conference room, dining room, highway, kitchen, living room, skyscraper, and street.
2. *Concept Detection and Visualization:* ConceptLens processes the uploaded image through trained CNN and Concept Induction to detect relevant concepts. The detected concepts are then presented as bar chart visualization and their corresponding error-margin percentages, providing users with a clear understanding of the network's predictions.
3. *Error-Margin Display:* The interface highlights the error-margin percentages for each detected concept, allowing users to gauge the confidence of the network's predictions. Lower percentages indicate higher confidence in the concept detection.

Demonstration The ConceptLens demonstrator is available online,[10] together with a video for a preview of its features[11] and the source code is available on GitHub.[12] Fig. 11 shows a screenshot of a ConceptLens output example. Note the (relatively) small error margins for the top mentioned detected concepts, most of which are clearly present in the image.

8. Limitations and Future Work

Despite the strong performance and interpretability demonstrated by our neurosymbolic Concept Induction framework, several limitations remain:

[10] https://conceptlens.streamlit.app/Explore_ConceptLens
[11] https://youtu.be/yLYig1IjB9Y
[12] https://github.com/abhilekha-dalal/ConceptLens

1. *Single-layer focus:-* We restrict our analysis to a single dense layer's activations, yet deep networks encode hierarchical features across many layers. In future work, we will extend Concept Induction and Concept Activation Analysis to convolutional layers and to combinations of neurons, to reveal how concepts emerge and interact throughout the network.
2. *Dependence on labeled data:-* Our error-margin computations and positive/negative concept sets require image annotations, which may be costly to obtain. We plan to explore semi-supervised or weakly supervised approaches—e.g., leveraging pseudo-labels or active learning—to reduce annotation requirements while maintaining explanation fidelity.
3. *Single-dataset evaluation:-* All experiments use the ADE20K dataset, which may limit generality. We intend to validate our approach on additional domains (such as medical or satellite imagery) and on other network architectures (e.g., Vision Transformers), to assess robustness across data modalities and model families.
4. *Fixed background-knowledge scale:-* We employ a 2 million-class Wikipedia-derived ontology, but domain-specific tasks may benefit from smaller, more focused knowledge graphs. Future work will investigate the trade-offs of ontology size and specificity, including experiments with specialized medical or scientific ontologies.

By acknowledging these limitations and outlining concrete next steps, we aim to guide future enhancements of neurosymbolic explainability methods.

9. Conclusion

In this study, we demonstrate and evaluate the use of Concept Induction as a post-hoc Explainable AI tool. Our findings indicate that indeed Concept Induction over a background knowledge provides detailed insights into the otherwise black-box nature of hidden layer computations. It is a neurosymbolic approach where the generation of explanations is not a black-box process, which has practical advantage across many applications, e.g., where the explanation generation process necessitates provable correctness. Of course such an advantage is only achieved at a cost, in our case that is – requiring labeled data. We view this approach not as a replacement of existing non-white-box Explainable AI methods, rather we introduce a novel neurosymbolic approach to complement the existing techniques, especially for situations where provable correctness is of utmost importance regardless of the cost.

Additionally, we provide a systematic approach to evaluate neuron-labeling by means of high-level concepts obtained through Concept Induction. In our evaluation protocol, we not only measure the explanation performance by asking *"What Concept(s) activate a neuron the most?"*, but also *"Given Concept X activates a neuron, how likely it is that the said neuron is activated by other concepts?"*, we argue that the latter is necessary for a trustworthy Explainable AI technique.

The process of achieving explanations using Concept Induction involves many steps, as such we also provide an automated end-to-end system; thereby reducing any manual efforts. We hope that such an automated system gets rid of impediments towards replicating the results and would nurture future research opportunities.

In further work, we intend to extend our work to diverse datasets and various neural network architectures. Additionally, we aim to enhance model interpretability by exploring additional concept induction results, using different background knowledge, across various neuron layers.

Acknowledgement. The authors acknowledge partial funding under National Science Foundation grants 2119753 and 2333782.

References

[1] S. A. and S. R., A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends, *Decision Analytics Journal* **7** (2023), 100230. doi:https://doi.org/10.1016/j.dajour.2023.100230. https://www.sciencedirect.com/science/article/pii/S277266222300070X.

[2] H. Abdi and L.J. Williams, Tukey's honestly significant difference (HSD) test, *Encyclopedia of research design* **3**(1) (2010), 1–5.

[3] A. Adadi and M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (XAI), *IEEE access* **6** (2018), 52138–52160.

[4] S.E. Akkamahadevi, A. Dalal and P. Hitzler, Automating CNN Neuron Interpretation using Concept Induction, in: *ISWC 2024 Demo Proceedings*, 2024.

[5] D. Alvarez-Melis and T.S. Jaakkola, On the Robustness of Interpretability Methods (2018), cite arxiv:1806.08049, presented at 2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), Stockholm, Sweden. http://arxiv.org/abs/1806.08049.

[6] P. Barbiero, G. Ciravegna, F. Giannini, M.E. Zarlenga, L.C. Magister, A. Tonda, P. Lió, F. Precioso, M. Jamnik and G. Marra, Interpretable neural-symbolic concept reasoning, in: *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, JMLR.org, 2023.

[7] A. Barua, C.L. Widmer and P. Hitzler, Concept Induction Using LLMs: A User Experiment for Assessment, in: *Neural-Symbolic Learning and Reasoning – 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, Part II*, T.R. Besold, A. d'Avila Garcez, E. Jiménez-Ruiz, R. Confalonieri, P. Madhyastha and B. Wagner, eds, Lecture Notes in Computer Science, Vol. 14980, Springer, 2024, pp. 132–148. doi:10.1007/978-3-031-71170-1_13.

[8] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou and A. Torralba, Understanding the role of individual units in a deep neural network, *Proceedings of the National Academy of Sciences* **117**(48) (2020), 30071–30078.

[9] R.A. Bradley and M.E. Terry, Rank analysis of incomplete block designs: I. The method of paired comparisons, *Biometrika* **39**(3/4) (1952), 324–345.

[10] K. Chauhan, R. Tiwari, J. Freyberg, P. Shenoy and K. Dvijotham, Interactive Concept Bottleneck Models, *Proceedings of the AAAI Conference on Artificial Intelligence* **37**(5) (2023), 5948–5955. doi:10.1609/aaai.v37i5.25736. https://ojs.aaai.org/index.php/AAAI/article/view/25736.

[11] R. Confalonieri, T. Weyde, T.R. Besold and F.M. del Prado Martín, TREPAN Reloaded: A Knowledge-Driven Approach to Explaining Black-Box Models, in: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, G.D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín and J. Lang, eds, Frontiers in Artificial Intelligence and Applications, Vol. 325, IOS Press, 2020, pp. 2457–2464. doi:10.3233/FAIA200378.

[12] R. Confalonieri, T. Weyde, T.R. Besold and F.M. del Prado Martín, Using ontologies to enhance human understandability of global post-hoc explanations of black-box models, *Artificial Intelligence* **296** (2021), 103471.

[13] J. Crabbé and M. van der Schaar, Concept Activation Regions: A Generalized Framework For Concept-Based Explanations, in: *Advances in Neural Information Processing Systems*, Vol. 35, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho and A. Oh, eds, Curran Associates, Inc., 2022, pp. 2590–2607.

[14] A. Dalal and P. Hitzler, ConceptLens: from Pixels to Understanding, 2024. https://arxiv.org/abs/2410.05311.

[15] A. Dalal, R. Rayan and P. Hitzler, Error-Margin Analysis for Hidden Neuron Activation Labels, in: *Neural-Symbolic Learning and Reasoning - 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, Part II*, T.R. Besold, A. d'Avila Garcez, E. Jiménez-Ruiz, R. Confalonieri, P. Madhyastha and B. Wagner, eds, Lecture Notes in Computer Science, Vol. 14980, Springer, 2024, pp. 149–164. doi:10.1007/978-3-031-71170-1_14.

[16] A. Dalal, R. Rayan, A. Barua, E.Y. Vasserman, M.K. Sarker and P. Hitzler, On the Value of Labeled Data and Symbolic Methods for Hidden Neuron Activation Analysis, in: *Neural-Symbolic Learning and Reasoning – 18th International Conference, NeSy 2024, Barcelona, Spain, September 9-12, 2024, Proceedings, Part II*, T.R. Besold, A. d'Avila Garcez, E. Jiménez-Ruiz, R. Confalonieri, P. Madhyastha and B. Wagner, eds, Lecture Notes in Computer Science, Vol. 14980, Springer, 2024, pp. 109–131. doi:10.1007/978-3-031-71170-1_12.

[17] N. Díaz-Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes and F. Herrera, EXplainable Neural-Symbolic Learning (X-NeSyL) methodology to fuse deep learning representations with expert knowledge graphs: The MonuMAI cultural heritage use case, *Information Fusion* **79** (2022), 58–83.

[18] S. Ekin, Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices, 2023. doi:10.36227/techrxiv.22683919.

[19] A. Ghorbani, J. Wexler, J.Y. Zou and B. Kim, Towards Automatic Concept-based Explanations, in: *Advances in Neural Information Processing Systems*, Vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds, Curran Associates, Inc., 2019.

[20] A. Goldstein, A. Kapelner, J. Bleich and E. Pitkin, Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation, *Journal of Computational and Graphical Statistics* **24**(1) (2015), 44–65. doi:10.1080/10618600.2014.907095.

[21] Y. Goyal, U. Shalit and B. Kim, Explaining Classifiers with Causal Concept Effect (CaCE), *CoRR* **abs/1907.07165** (2019). http://arxiv.org/abs/1907.07165.

[22] Y. Guan, F. Lécué, J. Chen, R. Li and J. Z. Pan, Knowledge-Aware Neuron Interpretation for Scene Classification, *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(3) (2024), 1950–1958. doi:10.1609/aaai.v38i3.27965. https://ojs.aaai.org/index.php/AAAI/article/view/27965.

[23] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf and G.-Z. Yang, XAI – Explainable artificial intelligence, *Science robotics* **4**(37) (2019), eaay7120.

[24] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] K. He, X. Zhang, S. Ren and J. Sun, Identity mappings in deep residual networks, in: *European conference on computer vision*, Springer, 2016, pp. 630–645.

[26] P. Hitzler, A review of the semantic web field, *Commun. ACM* **64**(2) (2021), 76–83. doi:10.1145/3397512.

[27] P. Hitzler, M. Krötzsch and S. Rudolph, *Foundations of Semantic Web Technologies*, Chapman and Hall/CRC Press, 2010. ISBN 9781420090505. http://www.semantic-web-book.org/.

[28] K. Hutson, D. Andresen, A. Tygart and D. Turner, Managing a heterogeneous cluster, in: *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, 2019, pp. 1–6.

[29] N. Kalibhat, S. Bhardwaj, C.B. Bruss, H. Firooz, M. Sanjabi and S. Feizi, Identifying interpretable subspaces in image representations, in: *International Conference on Machine Learning*, PMLR, 2023, pp. 15623–15638.

[30] B. Kim, M. Wattenberg, J. Gilmer, C.J. Cai, J. Wexler, F.B. Viégas and R. Sayres, Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV), in: *Proceedings of the International Conference on Machine Learning (ICML)*, J.G. Dy and A. Krause, eds, Proceedings of Machine Learning Research, Vol. 80, PMLR, 2018, pp. 2673–2682. http://proceedings.mlr.press/v80/kim18d.html.

[31] E. Kim, D. Jung, S. Park, S. Kim and S. Yoon, Probabilistic Concept Bottleneck Models, in: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato and J. Scarlett, eds, Proceedings of Machine Learning Research, Vol. 202, PMLR, 2023, pp. 16521–16540. https://proceedings.mlr.press/v202/kim23g.html.

[32] T.K. Kim, T test as a parametric statistic, *Korean journal of anesthesiology* **68**(6) (2015), 540–546.

[33] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K.T. Schütt, S. Dähne, D. Erhan and B. Kim, *The (Un)Reliability of Saliency Methods*, in: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer-Verlag, Berlin, Heidelberg, 2022, pp. 267–280–. ISBN 978-3-030-28953-9. https://doi.org/10.1007/978-3-030-28954-6_14.

[34] D.P. Kingma and M. Welling, Auto-Encoding Variational Bayes, in: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2014. http://arxiv.org/abs/1312.6114.

[35] P.W. Koh, T. Nguyen, Y.S. Tang, S. Mussmann, E. Pierson, B. Kim and P. Liang, Concept Bottleneck Models, in: *Proceedings of the 37th International Conference on Machine Learning*, H.D. III and A. Singh, eds, Proceedings of Machine Learning Research, Vol. 119, PMLR, 2020, pp. 5338–5348. https://proceedings.mlr.press/v119/koh20a.html.

[36] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature* **521**(7553) (2015), 436–444.

[37] J. Lehmann and P. Hitzler, Concept learning in description logics using refinement operators, *Mach. Learn.* **78**(1–2) (2010), 203–250. doi:10.1007/s10994-009-5146-2.

[38] V.I. Levenshtein, On the Minimal Redundancy of Binary Error-Correcting Codes, *Inf. Control.* **28**(4) (1975), 268–291. doi:10.1016/S0019-9958(75)90300-9.

[39] V.I. Levenshtein, On the minimal redundancy of binary error-correcting codes, *Information and Control* **28**(4) (1975), 268–291.

[40] S.M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, in: *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30, 2017.

[41] S.M. Lundberg and S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: *Advances in Neural Information Processing Systems 30*, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, Curran Associates, Inc., 2017, pp. 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[42] P.E. McKnight and J. Najab, Mann-Whitney U Test, in: *The Corsini Encyclopedia of Psychology*, Wiley, 2010.

[43] D. Minh, H.X. Wang, Y.F. Li and T.N. Nguyen, Explainable artificial intelligence: A comprehensive review, *Artificial Intelligence Review* (2022), 1–66.

[44] A. Nanda, B.B. Mohapatra, A.P.K. Mahapatra, A.P.K. Mahapatra and A.P.K. Mahapatra, Multiple comparison test by Tukey's honestly significant difference (HSD): Do the confident level control type I error, *International Journal of Statistics and Applied Mathematics* **6**(1) (2021), 59–65.

[45] T. Norrenbrock, M. Rudolph and B. Rosenhahn, Q-SENN: Quantized Self-Explaining Neural Networks, *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(19) (2024), 21482–21491. doi:10.1609/aaai.v38i19.30145. https://ojs.aaai.org/index.php/AAAI/article/view/30145.

[46] T. Oikarinen and T.-W. Weng, CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks, in: *International Conference on Learning Representations*, ICLR, 2023. https://openreview.net/forum?id=iPWiwWHc1V.

[47] T. Oikarinen, S. Das, L.M. Nguyen and T.-W. Weng, Label-free Concept Bottleneck Models, in: *The Eleventh International Conference on Learning Representations*, ICLR, 2023. https://openreview.net/forum?id=FlCg47MNvBA.

[48] C. Olah, A. Mordvintsev and L. Schubert, Feature Visualization, *Distill* (2017), https://distill.pub/2017/feature-visualization. doi:10.23915/distill.00007.

[49] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H.W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S.P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S.S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N.S. Keskar, T. Khan, L. Kilpatrick, J.W. Kim, C. Kim, Y. Kim, J.H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C.M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S.M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko,

P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H.P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V.H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F.P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M.B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J.F.C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J.J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk and B. Zoph, GPT-4 Technical Report, 2024. https://arxiv.org/abs/2303.08774.

[50] E. Poeta, G. Ciravegna, E. Pastor, T. Cerquitelli and E. Baralis, Concept-based Explainable Artificial Intelligence: A Survey, 2023. https://arxiv.org/abs/2312.12936.

[51] T. Procko, T. Elvira, O. Ochoa and N. Del Rio, An Exploration of Explainable Machine Learning Using Semantic Web Technology, in: *IEEE 16th International Conference on Semantic Computing (ICSC)*, IEEE, 2022, pp. 143–146.

[52] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, Learning Transferable Visual Models From Natural Language Supervision, in: *International Conference on Machine Learning*, PMLR, Vol. 139, 2021.

[53] M.T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1135–1144–. ISBN 9781450342322. doi:10.1145/2939672.2939778.

[54] S. Rudolph, M. Krötzsch, P. Patel-Schneider, P. Hitzler and B. Parsia, OWL 2 Web Ontology Language Primer (Second Edition), W3C Recommendation, W3C, 2012, https://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[55] M.K. Sarker and P. Hitzler, Efficient Concept Induction for Description Logics, in: *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI) The Thirty-First Innovative Applications of Artificial Intelligence Conference (IAAI), The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI)*, AAAI Press, 2019, pp. 3036–3043. doi:10.1609/aaai.v33i01.33013036.

[56] M.K. Sarker, N. Xie, D. Doran, M.L. Raymer and P. Hitzler, Explaining Trained Neural Networks with Semantic Web Technologies: First Steps, in: *Proceedings of the Twelfth International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, T.R. Besold, A.S. d'Avila Garcez and I. Noble, eds, CEUR Workshop Proceedings, Vol. 2003, CEUR-WS.org, 2017. https://ceur-ws.org/Vol-2003/NeSy17_paper4.pdf.

[57] M.K. Sarker, J. Schwartz, P. Hitzler, L. Zhou, S. Nadella, B.S. Minnery, I. Juvina, M.L. Raymer and W.R. Aue, Wikipedia Knowledge Graph for Explainable AI, in: *Proceedings of the Knowledge Graphs and Semantic Web Second Iberoamerican Conference and First Indo-American Conference (KGSWC)*, B. Villazón-Terrazas, F. Ortiz-Rodríguez, S.M. Tiwari and S.K. Shandilya, eds, Communications in Computer and Information Science, Vol. 1232, Springer, 2020, pp. 72–87. doi:10.1007/978-3-030-65384-2_6.

[58] R.R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh and D. Batra, Grad-CAM: Why did you say that?, Vol. abs/1611.07450, 2016. http://arxiv.org/abs/1611.07450.

[59] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, in: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626. doi:10.1109/ICCV.2017.74.

[60] S. Shin, Y. Jo, S. Ahn and N. Lee, A Closer Look at the Intervention Procedure of Concept Bottleneck Models, in: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato and J. Scarlett, eds, Proceedings of Machine Learning Research, Vol. 202, PMLR, 2023, pp. 31504–31520. https://proceedings.mlr.press/v202/shin23a.html.

[61] A. Shrikumar, P. Greenside and A. Kundaje, Learning Important Features Through Propagating Activation Differences, in: *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y.W. Teh, eds, Proceedings of Machine Learning Research, Vol. 70, PMLR, 2017, pp. 3145–3153. https://proceedings.mlr.press/v70/shrikumar17a.html.

[62] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: *3rd International Conference on Learning Representations (ICLR 2015)*, Computational and Biological Learning Society, 2015.

[63] D. Slack, S. Hilgard, E. Jia, S. Singh and H. Lakkaraju, Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods, in: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 180–186–. ISBN 9781450371100. doi:10.1145/3375627.3375830.

[64] D. Steinmann, W. Stammer, F. Friedrich and K. Kersting, Learning to Intervene on Concept Bottlenecks, in: *Proceedings of the 41st International Conference on Machine Learning*, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett and F. Berkenkamp, eds, Proceedings of Machine Learning Research, Vol. 235, PMLR, 2024, pp. 46556–46571. https://proceedings.mlr.press/v235/steinmann24a.html.

[65] A. Sun, P. Ma, Y. Yuan and S. Wang, Explain Any Concept: Segment Anything Meets Concept-Based Explanation, in: *Advances in Neural Information Processing Systems*, Vol. 36, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt and S. Levine, eds, Curran Associates, Inc., 2023, pp. 21826–21840. https://proceedings.neurips.cc/paper_files/paper/2023/file/44cdeb5ab7da31d9b5cd88fd44e3da84-Paper-Conference.pdf.

[66] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the inception architecture for computer vision, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[67] H. Turner and D. Firth, Bradley-Terry Models in R: The BradleyTerry2 Package, *Journal of Statistical Software* **48**(9) (2012), 1–21–. doi:10.18637/jss.v048.i09. https://www.jstatsoft.org/index.php/jss/article/view/v048i09.

[68] S. Wachter, B.D. Mittelstadt and C. Russell, Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR, *CoRR* **abs/1711.00399** (2017). http://arxiv.org/abs/1711.00399.

[69] B. Wang, L. Li, Y. Nakashima and H. Nagahara, Learning Bottleneck Concepts in Image Classification, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 10962–10971.

[70] C.L. Widmer, M.K. Sarker, S. Nadella, J. Fiechter, I. Juvina, B. Minnery, P. Hitzler, J. Schwartz and M. Raymer, Towards human-compatible XAI: Explaining data differentials with concept induction over background knowledge, *Journal of Web Semantics* **79** (2023), 100807. doi:https://doi.org/10.1016/j.websem.2023.100807. https://www.sciencedirect.com/science/article/pii/S1570826823000367.

[71] Y. Yang, A. Panagopoulou, S. Zhou, D. Jin, C. Callison-Burch and M. Yatskar, Language in a bottle: Language model guided concept bottlenecks for interpretable image classification, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19187–19197.

[72] M. Yuksekgonul, M. Wang and J. Zou, Post-hoc Concept Bottleneck Models, in: *The Eleventh International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=nA5AZ8CEyow.

[73] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, Learning Deep Features for Discriminative Localization, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[74] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, Scene Parsing through ADE20K Dataset, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 5122–5130. https://doi.org/10.1109/CVPR.2017.544.

[75] B. Zhou, D. Bau, A. Oliva and A. Torralba, Interpreting deep visual representations via network dissection, *IEEE transactions on pattern analysis and machine intelligence* **41**(9) (2018), 2131–2145.

[76] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso and A. Torralba, Semantic understanding of scenes through the ADE20K dataset, *International Journal of Computer Vision* **127**(3) (2019), 302–321.

Appendix A. Appendices

Appendix. Appendix B: Detailed Concept-Activation Results

The following tables present the full per-concept train and test accuracies for Concept Induction (Table 15), CLIP-Dissect (Table 16), and GPT-4 (Table 17). These detailed results underlie the summary statistics reported in Table 12 and Table 7 of the main text.

Table 15

Concept Accuracy in Hidden Layer Activation Space of Concepts extracted using Concept Induction.

Concept Name	CAR		CAV	
	Train Acc.	Test Acc.	Train Acc.	Test Acc.
Air Conditioner	0.8994	0.8415	0.811	0.8659
Baseboard	0.875	0.8717	0.8846	0.9102
Body	0.9035	0.8857	0.8642	0.9
Building	0.9085	0.9404	0.8262	0.8690
Bushes	0.9150	0.9487	0.9477	0.9743
Car	0.9464	0.9571	0.925	0.9429
Casserole	0.9458	0.9375	0.9808	0.975
Central Reservation	0.8694	0.9	0.8917	0.9
Chain	0.9556	0.9677	0.9637	0.9677
Cistern	0.8734	0.8375	0.8449	0.8875
Coffee Table	0.9047	0.9523	0.8988	0.9166
Crapper	0.8516	0.8043	0.8571	0.8695
Cross Walk	0.9166	0.9468	0.9247	0.9361
Dishcloth	0.9055	0.9375	0.9685	0.9531
Dish Rack	0.9375	0.9583	0.9843	0.9375
Dishrag	0.8603	0.9285	0.9144	0.9464
Doorcase	0.8936	0.8611	0.8581	0.8194
Edifice	0.9487	0.9642	0.9548	0.9523
Fire Hydrant	0.9171	0.9625	0.9171	0.925
Fire Escape	0.8950	0.9146	0.9104	0.8902
Flooring	0.8841	0.9166	0.8871	0.9047
Flusher	0.8722	0.8285	0.9014	0.9285
Fluorescent Tube	0.9006	0.9625	0.9358	0.9125
Footboard	0.9268	0.9519	0.9585	0.9423
Go Cart	0.9378	0.9512	0.9254	0.9390
Jar	0.9059	0.9333	0.9572	0.9666
Left Arm	0.8549	0.8536	0.8858	0.8658
Left Foot	0.8734	0.8658	0.8703	0.8536
Letter Box	0.8901	0.8636	0.875	0.9242
Lid	0.8622	0.9047	0.8712	0.8809
Manhole	0.9349	0.8953	0.9349	0.9302
Mountain	0.9426	0.95	0.9745	0.9625
Mouth	0.8963	0.9268	0.9481	0.9512
Night Table	0.8917	0.875	0.9235	0.8875
Nuts	0.9223	0.9134	0.9417	0.9230
Open Fireplace	0.9129	0.9222	0.9101	0.9333
Ornament	0.8910	0.9375	0.9198	0.9625
Paper Towels	0.9021	0.9166	0.9239	0.9166
Pillar	0.8372	0.8837	0.7732	0.8372
Pipage	0.84239	0.7826	0.7826	0.7391
Plank	0.8719	0.9523	0.9146	0.9047
Posters	0.8806	0.9230	0.8806	0.9230
Pylon	0.8397	0.8125	0.8205	0.8375
River	0.9430	0.925	0.9399	0.925
River Water	0.9554	0.9375	0.9617	0.9375
Road	0.9221	0.9642	0.9461	0.9404
Rocker	0.8953	0.9545	0.9457	0.8939
Rocking Horse	0.9173	0.9310	0.9347	0.9655
Saucepan	0.9561	0.9827	1	0.9827

Concept Name	CAR		CAV	
	Train Acc.	Test Acc.	Train Acc.	Test Acc.
Screen Door	0.9076	0.9375	0.9235	0.925
Sculpture	0.8242	0.8333	0.8788	0.8571
Shower Stall	0.9409	0.9722	0.9652	0.9583
Sideboard	0.91	0.94	0.965	0.92
Side Rail	0.9054	0.9459	0.8986	0.9054
Skyscraper	0.9455	0.9743	0.9615	0.9743
Slipper	0.9262	0.9456	0.9617	0.9565
Slope	0.8705	0.8714	0.9208	0.8857
Soap Dish	0.8733	0.8589	0.8474	0.8589
Soap Dispenser	0.88	0.9375	0.916	0.9531
Spatula	0.9017	0.9431	0.9219	0.9204
Stem	0.8834	0.8676	0.8383	0.8382
Stretcher	0.89375	0.9375	0.9312	0.9375
Tank Lid	0.8947	0.8846	0.8848	0.8717
Tap	0.8198	0.8536	0.8354	0.8902
Teapot	0.9365	0.9411	0.9552	0.9779
Toaster	0.927	0.9714	0.9197	0.9736
Toothbrush	0.9198	0.9125	0.9198	0.9
Utensils Canister	0.9262	0.925	0.9487	0.9375
Wardrobe	0.9375	0.95	0.9188	0.9125

Table 16

Concept Accuracy in Hidden Layer Activation Space of Concepts extracted using CLIP-Dissect.

Concept Name	CAR		CAV	
	Train Acc.	Test Acc.	Train Acc.	Test Acc.
Bathroom	0.9700	0.9474	0.9400	0.9474
Bed	0.9587	0.9500	0.9437	0.9125
Bedroom	0.9167	0.9167	0.9137	0.9048
Buildings	0.9321	0.9230	0.8990	0.8974
Dallas	0.9447	0.9318	0.9750	0.9545
Dining	0.9294	0.9125	0.8907	0.9000
Dresser	0.9762	0.9625	0.9650	0.9500
File	0.9837	0.9750	0.9681	0.9500
Furnished	0.8843	0.8875	0.8762	0.8625
Highways	0.9396	0.9375	0.9679	0.9531
Interstate	0.9293	0.9268	0.8593	0.8536
Kitchen	9848	0.9743	0.9590	0.9487
Legislature	0.9149	0.9000	0.9156	0.9000
Microwave	0.9803	0.9807	0.9873	0.9807
Mississauga	0.9041	0.9054	0.9467	0.9324
Municipal	0.8679	0.8461	0.9298	0.9102
Restaurants	0.9850	0.9722	0.9692	0.9583
Road	0.9362	0.9250	0.9387	0.9250
Room	0.8653	0.8125	0.8273	0.8250
Roundtable	0.9405	0.9473	0.9136	0.8947
Valencia	0.8735	0.8625	0.8781	0.875
Street	0.9830	0.9722	0.9347	0.9167

Table 17

Concept Accuracy in Hidden Layer Activation Space of Concepts extracted using GPT-4.

Concept Name	CAR		CAV	
	Train Acc.	Test Acc.	Train Acc.	Test Acc.
Bedroom	0.9851	0.9761	0.9660	0.9523
Bathroom	0.9176	0.9024	0.9068	0.8902
Bathroom Interior	0.9273	0.9146	0.9241	0.9268
Bicycle	0.9787	0.9615	0.9887	0.9871
Cityscape	0.9438	0.9358	0.9894	0.9743
Classroom	0.8981	0.8780	0.9012	0.8536
Dining Room	0.9256	0.9125	0.8942	0.8875
Eyeglasses	0.9813	0.9883	0.9883	0.9883
Home Interior	0.8515	0.8452	0.8363	0.8214
Indoor Home Decor	0.8428	0.8333	0.8418	0.8222
Indoor Home Setting	0.6713	0.6785	0.6890	0.6666
Kitchen	0.9122	0.9302	0.9122	0.9186
Kitchen Scene	0.8562	0.8571	0.8022	0.7976
Living Room	0.8963	0.8658	0.8658	0.8414
Outdoor Scenery	0.9135	0.9024	0.9054	0.9024
Outdoor Urban Scene	0.8343	0.8170	0.7650	0.7317
Street Scene	0.8819	0.8809	0.8568	0.8690
Toilet Brush	0.9815	0.9761	0.9727	0.9642
Urban Landscape	0.8665	0.8636	0.8922	0.8863
Urban Street Scene	0.9140	0.9024	0.8757	0.8658
Urban Transportation	0.8412	0.8414	0.8251	0.8414