# VISIONLOGIC: DISCOVERING AND GROUNDING DECISION-RELEVANT VISUAL CONCEPTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Concept-based explanations help users understand vision models through recognizable visual patterns. However, existing methods often rely on correlational signals without directly validating which image cues support prediction-relevant internal features. To this end, we introduce VISIONLOGIC, a post-hoc framework that grounds these features in visual concepts through intervention-based validation. VISIONLOGIC first identifies compact sets of features whose contributions reproduce the model's original prediction. It then represents their activation states as predicates using class-specific thresholds. An iterative refinement procedure grounds these predicates in visual regions through ablation tests. A region is accepted when its removal deactivates the corresponding predicate, linking the feature's numerical role to visual evidence in the input. The same predicates allow us to examine how features are activated, selected, and reused across images and classes. Across CNNs and vision transformers on ImageNet-1k, we find that only a few features are selected to explain each prediction, and frequently active features are not always selected. In a large-scale human evaluation with 465 participants, VISIONLOGIC significantly improves participants' understanding of model behavior over established methods ACE and CRAFT.

## 1 INTRODUCTION

Deep learning-based vision models have achieved remarkable success across numerous tasks, yet their *black-box* nature remains a major obstacle to trustworthy AI. This challenge persists as vision architectures evolve from Convolutional Neural Networks (CNNs) (LeCun et al., 1998; He et al., 2016) to Vision Transformers (ViTs) (Dosovitskiy et al., 2021). To address this issue, a wide range of interpretability methods have been proposed. Among them, concept-based explanations (Nguyen et al., 2016; Bau et al., 2017; Kim et al., 2018; Ghorbani et al., 2019; Fel et al., 2023; Achtibat et al., 2023) have attracted particular interest because they uncover high-level semantic concepts rather than only providing low-level attribution maps such as Grad-CAM (Selvaraju et al., 2017).

However, concept discovery often relies on *correlational evidence*, leaving the connection between recognizable concepts and prediction-relevant internal features insufficiently grounded (Lopez-Paz et al., 2017; Zhang et al., 2023; Goyal et al., 2019). For instance, TCAV (Kim et al., 2018) learns concept directions in activation space and measures prediction sensitivity, while ACE (Ghorbani et al., 2019) applies clustering to identify recurring concepts from image segments. These methods identify meaningful patterns and assess their relevance, but a concept's association with a class does not by itself establish which image cues support a specific internal feature. Without this validation, visually plausible explanations may attribute a prediction to the wrong image cues. This can mislead users about the model's behavior, particularly when identifying biases or understanding its failures.

We address this gap with VISIONLOGIC, a post-hoc framework for discovering and grounding decision-relevant visual concepts through intervention-based validation. Our approach operates in three stages. First, we identify compact sets of features whose contributions through the final classification head reproduce the model's original prediction. Second, we represent their activation states as predicates by fitting class-specific thresholds. This provides an intermediate symbolic representation for understanding how vision models make decisions.

Third, we ground these predicates in visual regions using an iterative procedure that combines neuron-specific attribution maps with ablation tests. By testing whether removing a region deactivates the



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

corresponding predicate, we connect internal features to supporting visual evidence. Foreground segmentation (Zheng et al., 2024) highlights the visual content of these regions, and recurring patterns across images help interpret the features as human-understandable concepts. Together, the three stages provide a common representation for visual explanations and global analysis of feature use.

Our large-scale human evaluation with 465 participants demonstrates the practical value of the discovered concepts. Compared with established concept-based methods ACE and CRAFT, VISIONLOGIC significantly improves participants' understanding of model behavior in tasks involving bias detection and the discovery of non-obvious strategies. Experiments on CNNs and ViTs on ImageNet-1k (Deng et al., 2009) show that features can be frequently active yet rarely selected to explain a prediction. Typically, only a few features are selected for each image, and some features are reused across hundreds of classes. Our contributions are summarized as follows:

- We propose VISIONLOGIC, a post-hoc framework that identifies compact sets of features whose contributions reproduce the model's prediction and represents their activation states as predicates for concept-based explanation.
- We develop an iterative grounding procedure that combines neuron-specific attribution maps with ablation tests to ground these predicates in visual regions.
- We conduct a large-scale human evaluation with 465 participants, demonstrating significant improvements over ACE and CRAFT in participants' understanding of model behavior.
- We analyze how features are activated, selected, and reused across CNNs and ViTs, connecting explanations of individual predictions to broader patterns of feature use across classes.

## 2 RELATED WORK

Existing approaches to interpreting vision models increasingly focus on *concept-based methods*, which aim to link internal representations to human-interpretable concepts rather than pixel-level attributions (Selvaraju et al., 2017; Chattopadhyay et al., 2018; Wang et al., 2020). Early studies explore the *hidden semantics* of neural networks by visualizing what individual neurons or layers encode. For instance, Nguyen et al. (2016) generate synthetic images that maximally activate specific neurons, while Mahendran & Vedaldi (2015) reconstruct inputs from intermediate feature maps to reveal the information preserved at different network depths. Network Dissection (Bau et al., 2017) further provides a systematic framework to quantify the alignment between hidden units and semantic concepts across diverse architectures. These methods offer valuable insights into model representations, motivating further study of how the identified features support individual predictions. Subsequent work establishes explicit links between model activations and human-interpretable concepts. NET2VEC (Fong & Vedaldi, 2018) trains linear predictors on activation patterns to align feature maps with semantic labels, while TCAV (Kim et al., 2018) learns concept directions in activation space and measures prediction sensitivity along these directions. These approaches help identify what a model represents and assess the relevance of predefined concepts.

More recent methods automate concept discovery. For instance, ACE (Ghorbani et al., 2019) applies unsupervised clustering to representations of image segments and scores the resulting concepts using TCAV. It also evaluates concept importance through addition and removal experiments. ICE (Zhang et al., 2021) introduces invertible concept-based explanations using non-negative concept activation vectors to improve interpretability and fidelity. CRAFT (Fel et al., 2023) further combines recursive concept discovery with sensitivity analysis to measure how concept perturbations affect model predictions. Related work on causal concept effects, such as CACE (Goyal et al., 2019), estimates how intervening on concept presence changes classifier predictions.

However, concept discovery often relies on *correlational evidence*, leaving the connection between recognizable concepts and prediction-relevant internal features insufficiently grounded. We address this gap with VISIONLOGIC, which directly tests whether candidate visual regions support the internal features selected to explain a prediction. The resulting explanations help users better understand the visual evidence behind a model's behavior than those provided by ACE and CRAFT, as demonstrated by our large-scale human evaluation.

# 3 THE VISIONLOGIC FRAMEWORK

VISIONLOGIC explains a frozen vision classifier in three stages. First, we identify a compact set of features whose contributions reproduce the model's prediction. Second, we represent their activation states as predicates using class-specific thresholds, allowing comparison across images and classes. Third, we ground these predicates in visual regions through ablation tests.

## 3.1 SELECTING COMPACT SETS OF FEATURES

Let $z(x) \in \mathbb{R}^d$ denote the feature activations supplied to the model's final linear classification head. The logit for class $k$ and the model prediction are

$$F_k(x) = b_k + \sum_{j=1}^{d} W_{kj} z_j(x), \qquad \hat{y}(x) = \arg\max_{k \in \{1,\dots,C\}} F_k(x). \tag{1}$$

For an image predicted as $c = \hat{y}(x)$, the term $W_{cj} z_j(x)$ gives feature $j$'s contribution to the class-$c$ logit. We sort features by decreasing contribution, obtaining the order $\pi_x = (\pi_1, \dots, \pi_d)$. Starting from the full bias vector, we add their contributions to all class logits in this order:

$$F_k^{(m)}(x) = b_k + \sum_{r=1}^{m} W_{k\pi_r} z_{\pi_r}(x), \qquad k \in \{1, \dots, C\}. \tag{2}$$

We stop at the first nonempty prefix that reproduces the original prediction:

$$m^*(x) = \min\{m \in \{1, \dots, d\} : \arg\max_k F_k^{(m)}(x) = c\}, \qquad P(x) = (\pi_1, \dots, \pi_{m^*(x)}). \tag{3}$$

Here, $P(x)$ lists the selected features in contribution order.

The selection procedure accounts for each feature's contribution to both the predicted class and its competitors. In particular, the margin against class $k$ after selecting the first $m$ features is

$$M_{c,k}^{(m)}(x) = b_c - b_k + \sum_{r=1}^{m} (W_{c\pi_r} - W_{k\pi_r}) z_{\pi_r}(x). \tag{4}$$

Keeping the activations of the features in $P(x)$ and setting all others to zero preserves the head's prediction.

We record each selected feature as $(j, +)$ if $z_j(x) \geq 0$ and $(j, -)$ otherwise. These pairs form the ordered sequence $S(x)$, preserving the order in $P(x)$. The sign refers to the activation $z_j(x)$ supplied to the head, rather than the contribution $W_{cj} z_j(x)$, and does not assume a particular activation function. Feature selection requires no classifier retraining, gradients, or concept annotations; sorting costs $O(d \log d)$ and accumulating the contributions takes at most $O(Cd)$ operations.

## 3.2 REPRESENTING FEATURE ACTIVATIONS AS PREDICATES

The selected features explain individual predictions through their numerical contributions. To compare their activation states across images and ground these features in image cues, we define predicates using class-specific thresholds. We write $p_{c,(j,s)}$ for the predicate associated with class $c$ and signed feature $(j, s)$. When $c$ is clear from context, we abbreviate it as $p_{(j,s)}$. Across classes, identical abbreviated labels refer to the same signed feature but may use different thresholds.

Let $\mathcal{D}_c^{\text{src}}$ denote the retained source examples assigned to model-predicted class $c$. We use the features selected by the procedure in Section 3.1, recorded in $S(x)$, to define the predicates and fit their thresholds. The same ordered sequence of signed features may occur in different classes. These selections explain the model's original predictions and do not assign classes independently.

**Class-specific thresholds.** For each signed feature, we collect the source images in class $c$ for which it is selected:

$$\mathcal{I}_{c,(j,s)} = \{x \in \mathcal{D}_c^{\text{src}} : (j, s) \in S(x)\}, \qquad \mathcal{V}_c = \{(j, s) : \mathcal{I}_{c,(j,s)} \neq \varnothing\}. \tag{5}$$

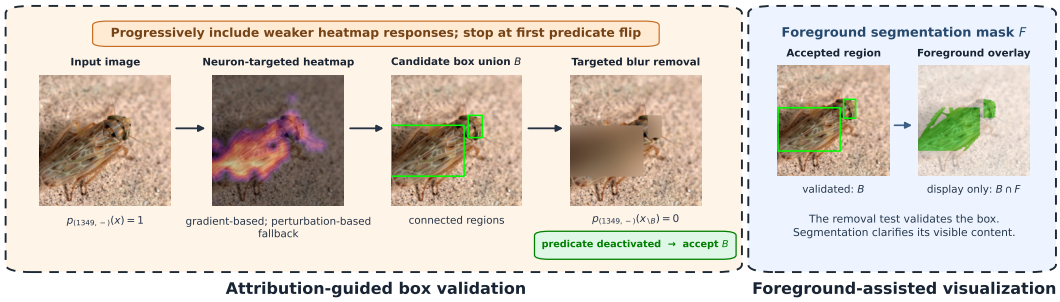



Figure 1: **Intervention-based visual grounding.** Neuron-specific attribution maps guide the search for candidate regions. We ablate these regions by replacing their content with a blurred version. If the predicate remains active, we iteratively refine the proposals by including weaker heatmap responses. The first union of boxes whose ablation deactivates the target predicate is accepted. Foreground segmentation highlights the visual content within the accepted boxes.

Thus, $\mathcal{V}_c$ contains the signed features selected in at least one source image of class $c$. We define a predicate for each of these features, with threshold and Boolean value

$$T_{c,(j,+)} = \min_{x \in \mathcal{I}_{c,(j,+)}} z_j(x), \qquad p_{c,(j,+)}(x) = \mathbf{1}\{z_j(x) \geq T_{c,(j,+)}\},$$
$$T_{c,(j,-)} = \max_{x \in \mathcal{I}_{c,(j,-)}} z_j(x), \qquad p_{c,(j,-)}(x) = \mathbf{1}\{z_j(x) \leq T_{c,(j,-)}\}. \tag{6}$$

Class-specific thresholds allow the same signed feature to have different activation criteria across classes. This design is motivated by feature polysemanticity: a shared feature may correspond to different visual concepts in different classes.

**Active and selected features.** For an image $x$, we use the predicates defined for its predicted class $c = \hat{y}(x)$. We distinguish three sets of signed features:

$$A_c(x) = \{(j,s) \in \mathcal{V}_c : p_{c,(j,s)}(x) = 1\},$$
$$Q_c(x) = \mathrm{set}(S(x)) \cap \mathcal{V}_c, \qquad J_c(x) = A_c(x) \cap Q_c(x). \tag{7}$$

Here, $A_c(x)$ contains features whose predicates are active, $Q_c(x)$ contains selected features with predicates defined from the source images, and $J_c(x)$ contains features satisfying both conditions. Selected features outside $\mathcal{V}_c$ remain part of the numerical explanation but have no fitted predicate for class $c$.

On source images, selection implies activation because the thresholds are fitted from the selected occurrences. For held-out images, we apply the same feature selection procedure, but a selected feature need not satisfy its fixed source threshold. For visual explanation, we ground the predicates associated with $J_c(x)$: their features are selected for the image, and their thresholds are satisfied.

### 3.3 Grounding Predicates in Visual Concepts

The final stage grounds abstract predicates in visual concepts through ablation tests, as illustrated in Figure 1. Given an image $x$, we consider the predicates associated with $J_c(x)$, which are active and whose features are selected to explain the prediction (Section 3.2). For each predicate, we blur a candidate region (Fong & Vedaldi, 2017) to obtain $x'$ and recompute the predicate. A change from active to inactive provides evidence that the region supports its activation, directly connecting the selected internal feature to visual evidence in the image.

To locate these regions, we use neuron-specific attribution maps that target the predicate's signed neuron activation rather than the final class score. We begin with Grad-CAM (Selvaraju et al., 2017) and draw tight bounding boxes around connected areas with strong heatmap responses. When several disjoint areas are highlighted, their boxes are ablated together. If the ablation does not deactivate the predicate, we progressively include areas with weaker heatmap responses. If the Grad-CAM search

fails, we repeat the search using another attribution method such as Score-CAM (Wang et al., 2020). Using different attribution methods provides complementary proposals and can expand the search space. We accept the first region whose ablation deactivates the predicate. If all searches fail, the predicate remains unlocalized for that image. In short, attribution guides where to search, while the ablation test determines which regions are accepted.

To make the accepted regions easier to interpret, we intersect their boxes with segmentation masks. We use an off-the-shelf foreground segmentation model, BiRefNet (Zheng et al., 2024), in our ImageNet experiments. Other segmentation methods, such as SAM (Kirillov et al., 2023), could also be used when separate object masks are desired. The resulting intersections, shown as colored overlays, highlight the segmented content, while the boxes show the full regions tested by ablation. We do not perform further ablation tests on these intersections because relevant image cues may include object contours or surrounding context that the masks exclude. By inspecting these visualizations across images, we identify recurring patterns that help interpret concepts associated with each predicate.

We present the complete algorithm, ablation operator, and search settings in Appendix C. We further validate our design choices by evaluating the attribution methods and comparing the guided search with a strong area-matched search baseline in Tables 2 and 3.

## 4 Experiments

We evaluate three complementary aspects of VisionLogic's explanations. Section 4.1 analyzes the class-specific numerical predicates. Section 4.2 evaluates the usefulness of the concept explanations in human model-understanding tasks, and Section 4.3 examines representative visual concepts.

### 4.1 Global Analysis of Feature Use

**Setup.** We examine how features are activated, selected, and reused in ViT-B/16 (Dosovitskiy et al., 2021), ResNet-50 (He et al., 2016), ConvNeXt-Base (Liu et al., 2022), and Swin-T (Liu et al., 2021). Our analysis distinguishes features whose predicates are active, represented by $A_c(x)$, from those selected to explain the prediction, represented by $Q_c(x)$ (Section 3.2). For each population $\mathcal{D}_c$ grouped by model-predicted class, we measure each predicate's activation and selection frequencies. We also examine feature reuse across classes and the number of active and selected predicates per image. Thresholds are fitted on source images and kept fixed for held-out evaluation.

**Feature activation and selection.** Figure 2 compares activation and selection frequencies within each predicted class. We observe that many predicates are active in a large proportion of source images, while their associated features are selected much less often. For example, in ViT class 326, predicate $p_{(257,+)}$ is active in 78.1% of examples and selected in 10.0%. In ConvNeXt class 713, the corresponding frequencies for predicate $p_{(90,-)}$ are 93.0% and 9.3%, respectively. On source images, selected features always satisfy their predicates because the thresholds are fitted from these activations. What stands out across all four models is how frequently a predicate remains active even when its feature is not selected. A feature can therefore be active in many images without

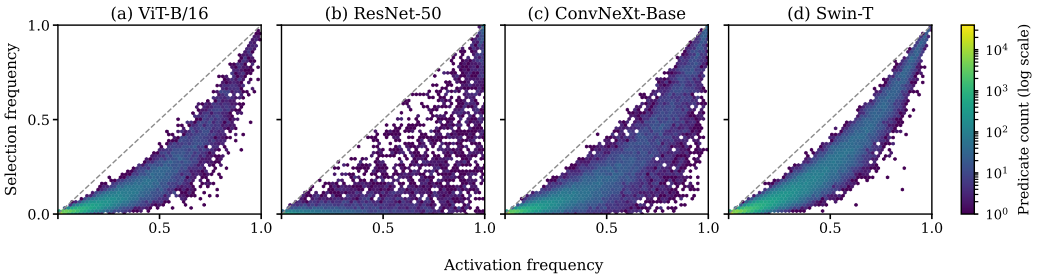



Figure 2: **Activation and selection frequencies on source images in four models.** Each hexbin counts class-specific predicates, with a shared logarithmic color scale. The dashed line indicates equal frequencies. Selection implies activation on source images by threshold construction. The large gaps below the line show that many frequently active features are selected much less often.

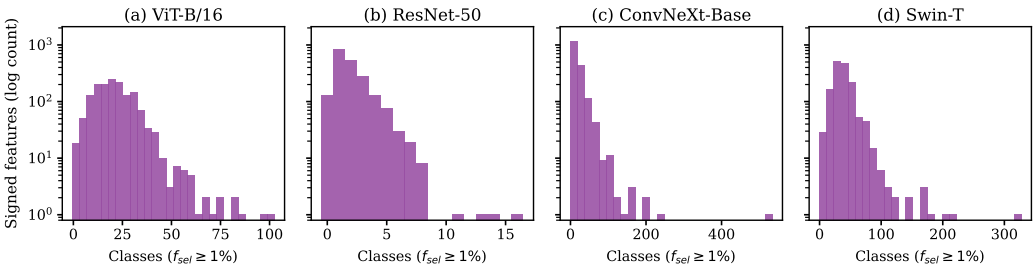



Figure 3: **Cross-class reuse of signed features in four models.** For each feature selected in the source images, we count the classes in which it is selected in at least 1% of source examples. The vertical axis is shared and logarithmic; horizontal ranges differ across models. Predicates for shared features use class-specific thresholds, and feature indices are specific to each model.

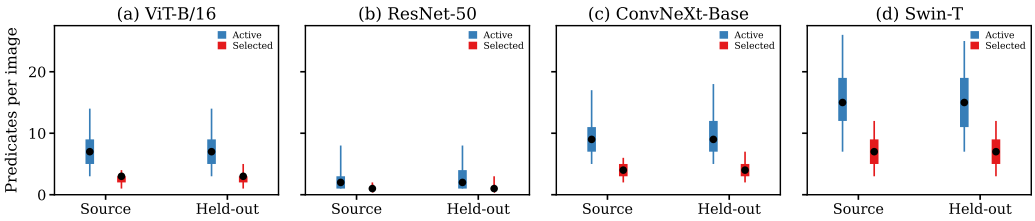



Figure 4: **Active and selected predicate counts per image.** For each model and data split, thin intervals span the 5th–95th percentiles, thick intervals show the interquartile range, and black points mark medians. These are distribution summaries, not confidence intervals.

often appearing in the compact sets that reproduce their predictions. This distinction helps separate common activation patterns from frequent use in our explanations.

**Features shared across classes.** Figure 3 shows how widely the same features are selected across different classes. Note that we count positive and negative activations of each feature separately within each model. For ConvNeXt, feature $(817, +)$ is selected in at least 1% of source examples in 534 classes, and in at least 5% of examples in 465 classes. For Swin, feature $(606, +)$ reaches the same cutoffs in 327 and 241 classes, respectively. Feature reuse varies considerably: some features are selected in only a few classes, while others recur across hundreds. These shared features connect explanations across classes, although their predicates use class-specific thresholds. Such reuse also raises the question of polysemanticity: does a feature represent one concept or different concepts across classes? Section 4.3 explores this question through grounded visual examples.

**Active and selected predicates per image.** Figure 4 summarizes how many predicates are active and selected for each image. The median active counts are 7, 2, 9, and 15 for ViT, ResNet, ConvNeXt, and Swin, respectively, while the corresponding median selected counts are 3, 1, 4, and 7. We apply the same feature selection procedure to source and held-out images. The median counts are unchanged between the two populations, and the interquartile ranges remain similar. Across all four models, the explanations typically involve only a few selected predicates, even when more are active. ResNet produces particularly compact explanations, with a median of one selected predicate per image. This means that one feature's contribution is often enough to reproduce the final head's prediction.

**Predicates on held-out images.** We examine whether predicates defined from the source images remain applicable to held-out images. Among all selected feature occurrences in held-out images, 98.6–99.5% have a predicate defined for the same class and signed feature in the source data. Of these matched occurrences, 97.4–98.9% satisfy the corresponding fixed threshold. Thus, most selected features on unseen images are covered by the source vocabulary and satisfy its thresholds without refitting. These strong generalization results support the simple feature-selection and predicate-construction procedure in Sections 3.1–3.2. Appendix D reports the detailed results and the concentration of feature selections within each class.

Table 1: **Utility scores in three application scenarios.** The Utility benchmark measures how well explanations help users identify general rules that transfer to unseen instances. During training, participants are shown images along with the explanations and model predictions, and are asked to infer the underlying decision rules. At test time, the benchmark evaluates participants' accuracy in predicting the model's output on novel images. Higher Utility scores indicate that the explanations provide more useful information for understanding the model's behavior on new samples. For each scenario, the first and second best results are in **bold** and underlined respectively. VISIONLOGIC *achieves higher Utility scores than* ACE *and* CRAFT *across all three scenarios, with statistically significant improvements over both methods in the first two.*

| | Husky vs. Wolf | | | | Otter vs. Beaver | | | | Kit Fox vs. Red Fox | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Session n° | 1 | 2 | 3 | Utility | 1 | 2 | 3 | Utility | 1 | 2 | 3 | Utility |
| Baseline | 65.7 | 68.6 | 70.3 | 1.00 | 84.4 | 90.3 | 92.2 | <u>1.00</u> | 84.1 | 89.0 | 84.1 | **1.00** |
| Control | 55.3 | 63.6 | 70.0 | 0.92 | 85.1 | 88.3 | 92.9 | <u>1.00</u> | 80.8 | 79.2 | 79.2 | 0.93 |
| ACE | 60.4 | 71.1 | 74.6 | <u>1.01</u> | 80.4 | 85.7 | 90.5 | 0.96 | 80.6 | 83.2 | 76.2 | 0.93 |
| CRAFT | 55.5 | 60.8 | 65.3 | 0.89 | 86.3 | 90.9 | 90.9 | <u>1.00</u> | 76.8 | 81.8 | 76.8 | 0.92 |
| VISIONLOGIC | 74.8 | 90.0 | 91.0 | **1.25** | 96.8 | 98.4 | 99.2 | **1.10** | 84.1 | 84.5 | 82.9 | <u>0.98</u> |

### 4.2 HUMAN EVALUATION OF ABLATION-GROUNDED CONCEPTS

**Setup.** We evaluate the practical utility of our proposed VISIONLOGIC, alongside prior concept-based methods ACE and CRAFT, in terms of *concept explanations* using the human-in-the-loop framework of Colin et al. (2022), which assesses how well explanations help participants understand a model's behavior across three real-world scenarios: (1) detecting bias in AI decisions (using Husky vs. Wolf classification), (2) identifying novel model strategies that are non-obvious to untrained observers (using Otter vs. Beaver classification), and (3) understanding failure cases (using Kit Fox vs. Red Fox classification). Each scenario adopts the meta-predictor paradigm: during the training phase, participants study example images paired with explanation images and model outputs, then predict the model's output on unseen images without access to the corresponding explanations.

In total, 531 participants were recruited from Prolific (Prolific, 2024), of whom 465 passed screening criteria aligned with Colin et al. (2022). We designed questionnaires for five conditions: (1) baseline (no explanation), (2) control (bottom-up saliency maps (Simonyan et al., 2014)), (3) ACE (Ghorbani et al., 2019), (4) CRAFT (Fel et al., 2023), and (5) VISIONLOGIC (ours), where ACE and CRAFT are considered state-of-the-art. To ensure fair comparison on the adapted datasets, all methods were re-run under identical experimental settings. Following prior work (Colin et al., 2022; Fel et al., 2023), we report the *utility score*, defined as participants' average test accuracy across the three sessions, normalized by the baseline; higher values indicate more effective human understanding of the model. Additional experimental details, including the participant recruitment process and the number of images used in each session and phase, are provided in Appendix E.1.

**Results.** Table 1 shows that VISIONLOGIC achieves higher utility than ACE and CRAFT across three scenarios, with statistically significant gains over both in the first two. Scores of 1.25 and 1.10 represent improvements of 25% and 10% over the no-explanation baseline, respectively. In the third scenario, we observe the same trend as prior work (Fel et al., 2023), where no existing method provides more effective information than Baseline. Nevertheless, VISIONLOGIC achieves the highest utility among the evaluated explanation methods, approaching the no-explanation baseline at 0.98.

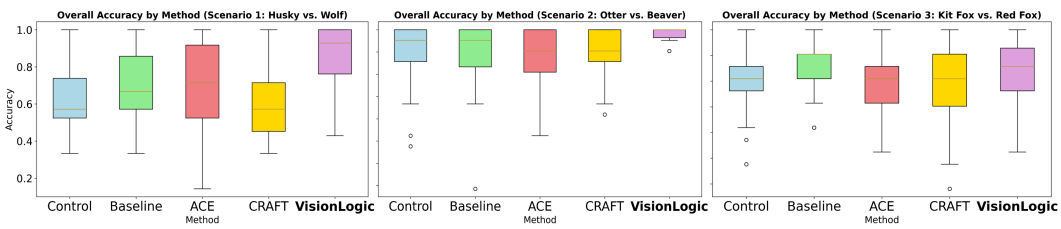



Figure 5: **Participant accuracy distributions for each explanation method.** VISIONLOGIC achieves higher mean accuracy than the ACE and CRAFT baselines across all three scenarios.

To rigorously examine the performance gain of our method over Baseline, Control, ACE, and CRAFT in the evaluation shown in Table 1, we perform statistical tests on the collected data. We initially follow the procedure of Colin et al. (2022): an analysis of variance (ANOVA; Scheffe, 1999) followed by Tukey's honestly significant difference (HSD) test for pairwise comparisons (Tukey, 1949). However, we noticed that the assumption checks for normality and homogeneity, which are crucial to the validity of parametric test results (Öztuna et al., 2006), are absent from prior works (Fel et al., 2023). To this end, we performed a complete statistical testing procedure with assumption checks. Figure 5 shows that the data is highly skewed as test accuracy is upper bounded by 1.0, hence violating the normality assumption. Therefore, non-parametric statistics are more appropriate for testing whether there is any statistically significant difference between the five conditions listed.

A Kruskal-Wallis test (McKight & Najab, 2010) with a null hypothesis of *"All condition distributions are identical"* and an alternative hypothesis of *"At least one condition distribution differs"* at a significance level of 0.05 rejects the null hypothesis with $p = 3.4 \times 10^{-5},\ 8.83 \times 10^{-3}$ for the first two scenarios. We then utilize Dunn's test (Dinno, 2015) to analyze pairwise differences with Bonferroni correction (Weisstein, 2004) applied. In the first scenario, VISIONLOGIC is shown to be *significantly* better than all other conditions with $p = 3.03 \times 10^{-2},\ 4.00 \times 10^{-4},\ 2.41 \times 10^{-2},\ p < 0.001$,

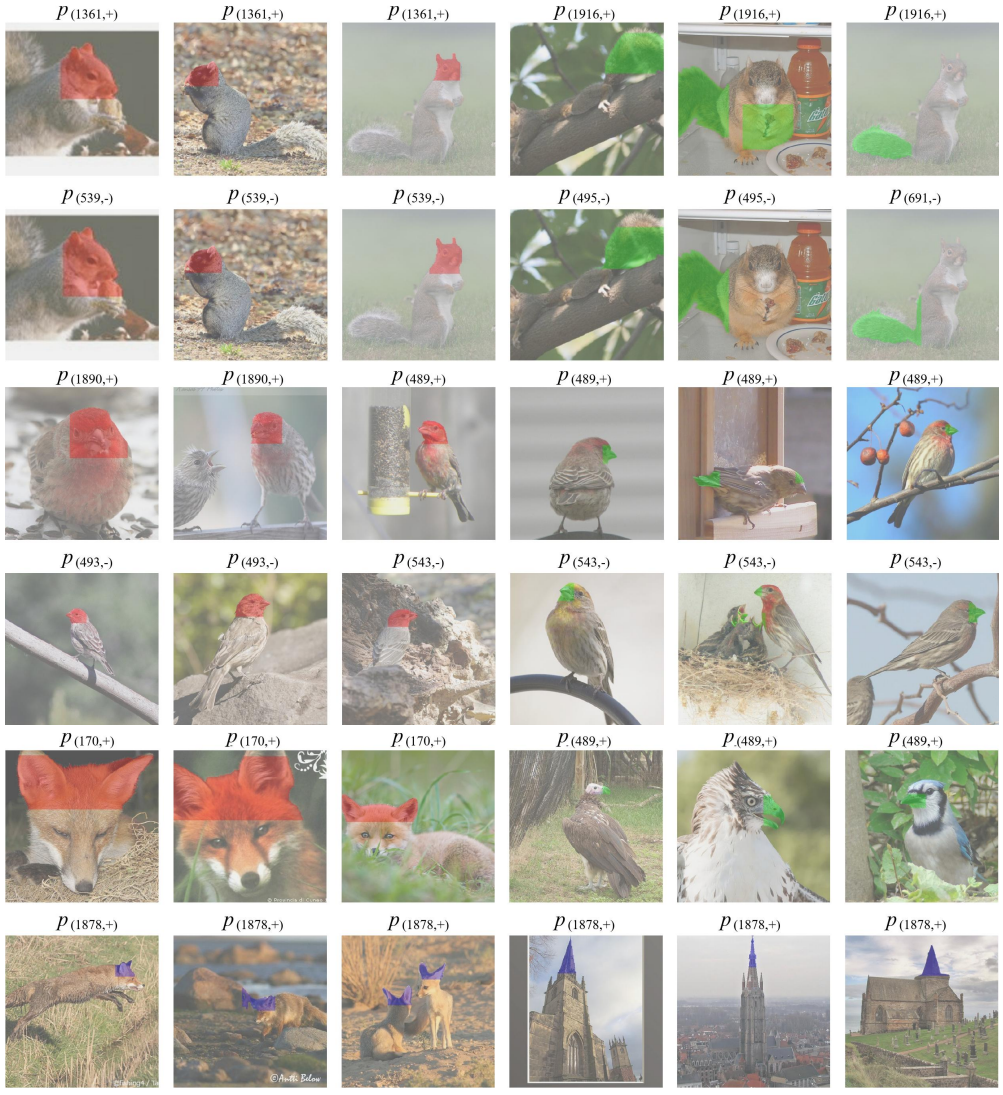



Figure 6: **Human-recognizable visual concepts grounded through predicates.** In each image, the predicate appears above the frame, and the colored overlay highlights the associated visual content.

respectively, suggesting that our method is effective in helping participants detect biases in the model. In the second scenario, VISIONLOGIC is shown to be *significantly* more effective than ACE and CRAFT with $p = 4.5 \times 10^{-3}$ and $3.09 \times 10^{-2}$ respectively, supporting its improvement over prior methods on the task of identifying non-obvious visual cues. Full test details, including the statistical tests involved, assumption checks, corrections, and test statistics are reported in Appendix E.2.

### 4.3 QUALITATIVE ANALYSIS OF VISUAL CONCEPTS

We present human-interpretable visual concepts encoded by predicates discovered in both ResNet and ViT. Each concept judgment is based on consistent visual inspection across many instances; Figure 6 shows representative cases. The 2nd and 4th rows are from ViT, the others from ResNet. Sampled concepts include *squirrel head*, *squirrel tail/paws*, *bird head*, *bird beak*, *fox ears*, and *church tops*.

**Polysemanticity.** We observe a many-to-many relationship between predicates and concepts (polysemanticity). Within a class, regions associated with one predicate can contain different visual concepts; across classes, identical abbreviated labels indicate reuse of the same (signed) feature. Conversely, regions associated with different predicates can depict the same concept. We interpret these recurring visual associations as evidence of polysemanticity, with ablation validating the combined region. For example, in the 5th image of row 1, both the tail and paws of the squirrel are captured by $p_{(1916,+)}$ (ResNet). Predicate $p_{(1878,+)}$ simultaneously encodes *fox ears* and *church tops*; while semantically different, both share a triangular geometry, which may explain the reuse. Predicate $p_{(489,+)}$ (rows 3 and 5) frequently encodes *bird beak* across species (*jay*, *vulture*, *kite*, etc.), and recurs in local explanations for bird images, indicating an influential role in inference.

We also find cases where a single concept is captured by multiple predicates. For the *fox* examples (rows 5–6), both $p_{(1878,+)}$ and $p_{(170,+)}$ encode *fox ears*. A predicate may also attend to multiple instances of the same concept within an image: in row 6, image 3 (two foxes), $p_{(1878,+)}$ captures both pairs of ears. Such behaviors highlight the role of predicates as global concept detectors rather than local ones.

**CNNs vs. Transformers.** Similar concepts are found in both model families (see rows 1–2, where both capture the same concept on the same image). In these examples, ViT tends to involve *more* predicates per visual concept, whereas ResNet yields sparser, more distinct encodings. These observations complement the quantitative analysis of feature use in Section 4.1. We hypothesize two factors behind this difference: (1) activation functions (ReLU vs. GELU) induce different predicate sparsity through positive and negative activation branches; and (2) convolution versus attention imposes different inductive biases on concept formation. Validating these hypotheses and examining how architectural differences shape the discovered concepts remain directions for future work.

## 5 CONCLUSION

Existing concept-based explanation methods for vision models often rely on *correlational evidence*, leaving the connection between recognizable concepts and prediction-relevant internal features insufficiently grounded. To address this gap, we propose VISIONLOGIC, a post-hoc framework that grounds these features in visual concepts through intervention-based validation. Beyond explaining individual predictions, the same predicates allow us to examine how features are activated, selected, and reused across images and classes. Experiments on CNNs and ViTs show that compact sets of features can reproduce a prediction even when many more features are active. They also reveal extensive feature reuse across classes. Our large-scale human evaluation with 465 participants further demonstrates the practical value of the discovered concepts: VISIONLOGIC's explanations achieve higher Utility scores than ACE and CRAFT across three model-understanding tasks. We envision this work as a step toward bridging the gap between recognizable visual concepts and an understanding of how vision models make decisions.

**Limitations.** Our grounding procedure uses localized blur interventions that preserve image content outside the candidate regions. However, these interventions can still introduce distribution shift, so changes in feature activation may reflect intervention artifacts as well as the removal of relevant visual evidence. Distinguishing these effects remains an open challenge in explainable AI.

## AI USE STATEMENT

The authors developed the research ideas, methods, algorithms, and experimental design. Generative AI tools assisted with code implementation, debugging, experiment execution, and preparation of result summaries under the authors' direction, as well as writing refinement and presentation. The authors reviewed and debugged the AI-assisted code, verified the reported results, and retained responsibility for all methodological decisions and interpretation of findings. The authors take full responsibility for the final manuscript, code, and results.

## ETHICS STATEMENT

The human study received institutional research ethics approval, anonymized for review, and included informed consent, participant screening, and compensation, as detailed in Appendix E.1.

## REPRODUCIBILITY STATEMENT

Sections 3.1–3.2 describe how we select compact sets of features and represent their activation states as predicates using class-specific thresholds. Appendix A describes the models, data, and hardware. Appendix B provides the feature-selection and threshold-fitting algorithm and implementation details. Appendix C specifies the iterative grounding procedure, attribution methods, blur intervention, foreground visualization, and comparison with the area-matched search baseline. Appendix D defines the image populations and metrics used to analyze feature activation, selection, and reuse. Appendix E details the human-study protocol and statistical analyses.

## REFERENCES

Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance propagation. *Nat. Mac. Intell.*, 5(9):1006–1019, 2023. doi: 10.1038/S42256-023-00711-8. URL https://doi.org/10.1038/s42256-023-00711-8.

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

Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pp. 839–847. IEEE Computer Society, 2018. doi: 10.1109/WACV.2018.00097. URL https://doi.org/10.1109/WACV.2018.00097.

Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. What I cannot predict, I do not understand: A human-centered evaluation framework for explainability methods. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/13113e938f2957891c0c5e8df811dd01-Abstract-Conference.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Alexis Dinno. Nonparametric pairwise multiple comparisons in independent groups using dunn's test. *The Stata Journal*, 15(1):292–300, 2015.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Thomas Fel, Agustin Martin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: concept recursive activation factorization for explainability. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 2711–2721. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00266. URL https://doi.org/10.1109/CVPR52729.2023.00266.

Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8730–8738. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00910. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Fong_Net2Vec_Quantifying_and_CVPR_2018_paper.html.

Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 3449–3457. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.371. URL https://doi.org/10.1109/ICCV.2017.371.

Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim. Towards automatic concept-based explanations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9273–9282, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/77d2afcb31f6493e350fca61764efb9a-Abstract.html.

Yash Goyal, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *CoRR*, abs/1907.07165, 2019. URL http://arxiv.org/abs/1907.07165.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2673–2682. PMLR, 2018. URL http://proceedings.mlr.press/v80/kim18d.html.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3992–4003. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00371. URL https://doi.org/10.1109/ICCV51070.2023.00371.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://doi.org/10.1109/5.726791.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 9992–10002. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00986. URL https://doi.org/10.1109/ICCV48922.2021.00986.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11966–11976. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01167. URL https://doi.org/10.1109/CVPR52688.2022.01167.

David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Schölkopf, and Léon Bottou. Discovering causal signals in images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 58–66. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.14. URL https://doi.org/10.1109/CVPR.2017.14.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 5188–5196. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7299155. URL https://doi.org/10.1109/CVPR.2015.7299155.

Patrick E McKight and Julius Najab. Kruskal-wallis test. *The corsini encyclopedia of psychology*, pp. 1–1, 2010.

Anh Mai Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on*

*Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3387–3395, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5d79099fcdf499f12b79770834c0164a-Abstract.html.

Derya Öztuna, Atilla Halil Elhan, and Ersöz Tüccar. Investigation of four different normality tests in terms of type 1 error rate and power under different distributions. *Turkish Journal of Medical Sciences*, 36(3):171–176, 2006.

Prolific. Prolific. https://www.prolific.com, 2024. Participants were recruited using Prolific (www.prolific.com). Accessed: April 24, 2025.

Henry Scheffe. *The analysis of variance*. John Wiley & Sons, 1999.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 618–626. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.74. URL https://doi.org/10.1109/ICCV.2017.74.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

John W Tukey. Comparing individual means in the analysis of variance. *Biometrics*, pp. 99–114, 1949.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pp. 111–119. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00020. URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w1/Wang_Score-CAM_Score-Weighted_Visual_Explanations_for_Convolutional_Neural_Networks_CVPRW_2020_paper.html.

Eric W Weisstein. Bonferroni correction. *https://mathworld. wolfram. com/*, 2004.

Kexuan Zhang, Qiyu Sun, Chaoqiang Zhao, and Yang Tang. Causal reasoning in typical computer vision tasks. *CoRR*, abs/2307.13992, 2023. doi: 10.48550/ARXIV.2307.13992. URL https://doi.org/10.48550/arXiv.2307.13992.

Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. Invertible concept-based explanations for CNN models with non-negative concept activation vectors. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 11682–11690. AAAI Press, 2021. doi: 10.1609/AAAI.V35I13.17389. URL https://doi.org/10.1609/aaai.v35i13.17389.

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024. doi: 10.26599/AIR.2024.9150038. URL https://www.sciopen.com/article/10.26599/AIR.2024.9150038.

# A EXPERIMENTAL SETUP

**Models and data.** We study feature activation, selection, and reuse in four pretrained ImageNet-1k models: ResNet-50 (He et al., 2016), ConvNeXt-Base (Liu et al., 2022), ViT-B/16 (Dosovitskiy et al., 2021), and Swin-T (Liu et al., 2021). For each model, we select compact sets of features whose contributions through the final classification head reproduce its predictions. We then use the selected source activations to fit class-specific thresholds and apply these thresholds to held-out images without refitting. Section 4.1 presents the main findings, and Appendix D gives the image populations and evaluation metrics. The following sections explain how we select features, construct and ground predicates, analyze feature use, and evaluate the resulting explanations with human participants.

**Hardware.** We ran the experiments on Ubuntu 22.04 LTS with an AMD EPYC™ 7532 processor (32 cores), 128 GB RAM, and a single NVIDIA A100 GPU with 40 GB memory. The pretrained vision models are kept fixed throughout feature selection, predicate construction, and visual grounding.

# B FEATURE SELECTION AND PREDICATE IMPLEMENTATION

**Source image selection.** We use the ImageNet-1k training split as the source population and the ImageNet validation split as held-out data. For each class, we use its correctly classified training images as the source population for computing the numerical contributions and predicates described in Sections 3.1–3.2. Appendix D gives the resulting source and held-out populations.

**Feature ordering and numerical implementation.** We order features by decreasing contribution to the predicted class using `np.argsort(contributions, kind='stable')[::-1]`. The same convention is used for all four models and for both source and held-out images, giving a consistent ordering when contributions are equal. Predictions are obtained with `np.argmax`, which chooses the first class index in a tie. We apply the stopping condition directly, without adding a tolerance or margin. The positive activation branch includes zero; negative features $(j, -)$ are stored as $j + d$.

The full prediction is computed from `matrix @ weights.T + bias`. To find a compact set of features that reproduces it, we begin with `bias.copy()` and add each feature's contribution to all class logits in order (Section 3.1). Matrix multiplication and sequential addition can produce small floating-point differences. We access the final classification heads through `heads.head` in ViT, `fc` in ResNet, `classifier.2` in ConvNeXt, and `head` in Swin.

**Feature selection and predicate construction.** For each source image, we identify a compact set of features whose contributions through the frozen classification head reproduce its prediction. We record these features in contribution order as $S(x)$ and use their activations to fit the class-specific thresholds. Threshold fitting uses exactly this selection, ending at the first prefix that recovers the prediction. Every selected source sequence reproduces the original head prediction. Features shared across classes contribute to separate class-specific predicates, allowing their activation thresholds to differ even when the selected sequences are identical.

**Held-out evaluation.** For a held-out image, we select features using the same procedure and evaluate the predicates fitted for its predicted class (Algorithm 1). Their thresholds remain fixed, allowing us to examine how well the source predicates apply to unseen images. We obtain the active set $A_c(x)$ and selected set $Q_c(x)$, then ground the predicates associated with their intersection $J_c(x)$ in visual regions.

# C GROUNDING PREDICATES IN VISUAL CONCEPTS

**Input and predicate selection.** We resize the shorter side of each image to 256 pixels using bilinear interpolation and take a $224 \times 224$ center crop. Standard ImageNet normalization is applied before classifier evaluation. Given an image $x$, we fix its original prediction $c = \hat{y}(x)$ and consider the

**Algorithm 1:** Feature selection, predicate construction, and held-out evaluation

**Input:** Frozen representation and classification head $(z, W, b)$; retained source images; shared stable sorting convention.

**Output:** Ordered selected features $S$; class-specific predicate vocabularies and thresholds $\{\mathcal{V}_c, T_c\}$; active and selected sets for held-out images.

**Procedure** SELECTFEATURES($x$)**:**
- $c \leftarrow \arg\max_k (Wz(x) + b)_k$
- $\pi \leftarrow$ decreasing contribution order of $W_{cj} z_j(x)$
- $\ell \leftarrow b$; $S \leftarrow ()$
- **for** $r = 1, \ldots, d$ **do**
  - $j \leftarrow \pi_r$; $s \leftarrow +$ if $z_j(x) \geq 0$, otherwise $-$
  - $\ell \leftarrow \ell + W_{:j} z_j(x)$; append $(j, s)$ to $S$
  - **if** $\arg\max_k \ell_k = c$ **then**
    - **break**
- **return** $(c, S)$

**foreach** *retained source image* $x$ **do**
- $(c, S(x)) \leftarrow$ SELECTFEATURES($x$); store the predicted class $c$

Initialize $\mathcal{V}_c \leftarrow \varnothing$ for each source class

**foreach** *retained source tuple* $(x, c, S(x))$ **do**
- **foreach** $(j, s) \in S(x)$ **do**
  - **if** $(j, s) \notin \mathcal{V}_c$ **then**
    - $T_{c,(j,s)} \leftarrow +\infty$ for $s = +$, otherwise $-\infty$
    - $\mathcal{V}_c \leftarrow \mathcal{V}_c \cup \{(j, s)\}$
  - $T_{c,(j,s)} \leftarrow \min(T_{c,(j,s)}, z_j(x))$ for $s = +$; otherwise use $\max$

Freeze all $\mathcal{V}_c$ and $T_{c,(j,s)}$

**foreach** *held-out image* $x$ **do**
- $(c, S(x)) \leftarrow$ SELECTFEATURES($x$)
- **if** $\mathcal{V}_c$ *is unavailable* **then**
  - continue
- $A_c(x) \leftarrow \{(j, s) \in \mathcal{V}_c : p_{c,(j,s)}(x) = 1\}$
- $Q_c(x) \leftarrow \text{set}(S(x)) \cap \mathcal{V}_c$
- $J_c(x) \leftarrow A_c(x) \cap Q_c(x)$
- Record $(c, S(x), A_c(x), Q_c(x), J_c(x))$ and selected features outside $\mathcal{V}_c$

predicates associated with $J_c(x) = A_c(x) \cap Q_c(x)$ (Section 3.2). These predicates are active on the image, and their features are selected to explain its prediction. Their class-specific thresholds are fitted on the source images and remain fixed during grounding. We examine each predicate independently to identify the visual regions supporting its activation. Selected features without an active source predicate remain in the numerical explanation but are not grounded visually.

**Neuron-specific attribution maps.** To locate candidate regions, we use attribution maps that target the signed activation $sz_j$ supplied to the final classification head, with $s = 1$ for the positive branch and $s = -1$ for the negative branch. The maps therefore highlight regions associated with the feature we aim to ground. For Grad-CAM (Selvaraju et al., 2017), let $A_k$ be channel $k$ at the chosen spatial layer and $\Omega$ its spatial index set. We compute

$$\gamma_k = \frac{1}{|\Omega|} \sum_{u \in \Omega} \frac{\partial (sz_j)}{\partial A_k(u)}, \qquad H_{\text{grad}} = \mathcal{N}\left( \text{ReLU}\left( \sum_k \gamma_k A_k \right) \right). \tag{8}$$

If the Grad-CAM search fails to find an accepted region, we repeat the search with Score-CAM (Wang et al., 2020). Let $M_k$ be the bilinearly upsampled and spatially min–max-normalized map of $A_k$, and $\bar{x}$ the normalized classifier input. We compute

$$a_k = sz_j(\bar{x} \odot M_k), \qquad \omega_k = \frac{\exp(a_k)}{\sum_\ell \exp(a_\ell)}, \qquad H_{\text{score}} = \mathcal{N}\left( \text{ReLU}\left( \sum_k \omega_k A_k \right) \right), \tag{9}$$

where $z_j$ in Equation 9 is evaluated on the masked normalized input, and $\mathcal{N}$ denotes spatial normalization and resizing to the input grid. For negative predicates, both methods target decreasing $z_j$.

We follow the `pytorch-grad-cam` implementation conventions. Score-CAM evaluates masks from all spatial channels in batches of eight, sharing these forward evaluations across neuron targets. A softmax over the signed neuron scores gives the channel weights. The weighted spatial sum is then clipped at zero, normalized, resized, and normalized again to produce the heatmap. Input masks use bilinear interpolation with `align_corners=True` and a $10^{-8}$ denominator stabilizer. Both attribution methods provide candidate regions for the subsequent ablation tests.

We obtain spatial activations from `layer4.2` in ResNet-50, `features.7` in ConvNeXt-Base, `encoder.layers.encoder_layer_11.ln_1` in ViT-B/16, and `features.7.1.norm1` in Swin-T. For ViT, we remove the class token and reshape the remaining tokens into a $14 \times 14$ grid. For Swin, we permute channels-last features to channels-first. These spatial layers provide the maps used for localization, while the attribution target is the signed activation of the selected feature at the final classification head. The chosen layer determines the heatmap's spatial resolution.

**Iterative region refinement and ablation.** Algorithm 2 gives the grounding procedure described in Section 3.3. We first propose boxes around areas with strong Grad-CAM responses. If ablating these regions leaves the predicate active, we progressively include weaker responses and test the resulting boxes. When the complete Grad-CAM search fails, we repeat the procedure with Score-CAM. We accept the first union of boxes whose ablation deactivates the predicate. If neither search succeeds, the predicate remains unlocalized for that image.

Let $x$ be the original $224 \times 224$ RGB crop, $B$ a candidate union of boxes, and $G_{20}(x)$ the image blurred with a Gaussian blur radius of 20 pixels. We ablate $B$ by replacing its content with the corresponding blurred pixels:

$$x^{\setminus B} = (1 - \mathbf{1}_B) \odot x + \mathbf{1}_B \odot G_{20}(x). \tag{10}$$

Throughout the search, we keep the original class, activation sign, and threshold fixed. A nonempty candidate is accepted when

$$p_{c,(j,s)}(x) = 1 \quad \text{and} \quad p_{c,(j,s)}(x^{\setminus B}) = 0. \tag{11}$$

**Algorithm 2:** Iterative visual grounding with neuron-specific attribution and ablation tests

**Input:** RGB image $x$; predicate $p = p_{c,(j,s)}$ with $(j, s) \in J_c(x)$; fixed class, sign, and threshold
**Output:** Accepted box union $B$ and foreground overlay mask $D$, or NoRegion

$x_{\text{blur}} \leftarrow \text{GaussianBlur}(x, \text{radius} = 20)$
**foreach** $m \in (\text{GradCAM}, \text{ScoreCAM})$ **do**
  $H \leftarrow \text{SignedNeuronAttribution}(m, x, j, s)$
  **foreach** $\alpha \in (0.60, 0.55, \ldots, 0.10)$ **do**
    $U_\alpha \leftarrow \{u : H(u) \geq \alpha \max(H)\}$ if $\max(H) > 0$, otherwise $\varnothing$
    $\mathcal{C}_\alpha \leftarrow \text{ConnectedComponents}(U_\alpha, \text{connectivity} = 8)$
    $B \leftarrow \bigcup_{C \in \mathcal{C}_\alpha} \text{TightBox}(C)$
    **if** $B \neq \varnothing$ **then**
      $x' \leftarrow (1 - \mathbf{1}_B) \odot x + \mathbf{1}_B \odot x_{\text{blur}}$
      **if** $p(x') = 0$ **then**
        $F \leftarrow \{u : \text{BiRefNet}(x)(u) \geq 0.5\}$
        $D \leftarrow B \cap F$ `// Foreground visualization`
        **return** $(B, D)$
**return** NoRegion

For each attribution map, we use eleven cutoffs $\alpha = 0.60, 0.55, \ldots, 0.10$, relative to the maximum heatmap response. At each cutoff, we find all nonempty 8-connected components and draw tight axis-aligned boxes around them. We retain small components as well as large ones and combine their boxes into a single ablation mask. The boxes have no added margins or foreground restriction. A zero heatmap provides no candidate regions for that attribution method.

As the cutoff decreases, weaker responses enter the candidate region, potentially enlarging or merging its boxes. A larger region need not produce a larger change in the target activation, so we recompute the predicate after each ablation. All boxes for one predicate are tested together, and each predicate is examined separately using the original image. Once a region passes the test, we retain it without further area minimization or random restarts.

We compute `PIL.ImageFilter.GaussianBlur(radius=20)` once on the original RGB crop and use the corresponding blurred pixels for every candidate region. The radius is measured at the $224 \times 224$ input resolution; it is not a 20-pixel kernel width. Each ablation starts from the original image and is applied before model normalization, leaving pixels outside the boxes unchanged. A positive predicate becomes inactive when $z_j(x') < T_{c,(j,+)}$, while a negative predicate becomes inactive when $z_j(x') > T_{c,(j,-)}$. Equality remains active. This threshold crossing determines acceptance, without requiring a change in the predicted class or a separate amount of activation suppression.

Blur removes fine detail while preserving some low-frequency structure. It can therefore leave relevant cues intact or introduce artifacts at region boundaries. When a search fails, we leave the predicate unlocalized because none of the tested regions deactivated it under the chosen blur operator and cutoff schedule.

**Foreground visualization.** We use the pretrained `ZhengPeng7/BiRefNet` model (Zheng et al., 2024) to highlight the foreground content within accepted regions. We resize the original crop to $1024 \times 1024$ and apply ImageNet normalization before segmentation. A sigmoid converts the final output to foreground probabilities, which are resized bilinearly to $224 \times 224$ with `align_corners=False` and thresholded at $0.5$. We use this mask directly, without output min–max normalization or dilation. Its intersection with the accepted boxes gives the foreground overlay, without further restriction to the heatmap. An empty intersection produces no overlay but leaves the accepted boxes available for inspection.

We provide two complementary views of each explanation. Solid colored boundaries show the full boxes tested by ablation, while colored overlays on a lightened image highlight their foreground content. Regions associated with the same predicate share a color, including disconnected boxes, and different predicates receive different colors. Where overlays overlap, their colors are averaged. Since segmentation depends only on the image, its mask can be computed once and reused for all predicates.

Foreground segmentation makes the accepted regions easier to interpret, as described in Section 3.3. We do not apply further ablation tests to the intersections because relevant cues may include object contours or surrounding context excluded by the masks. The box view preserves this context, while the overlays help readers identify recurring visual patterns and interpret the concepts associated with each predicate.

**Grounding evaluation setup.** We evaluate the attribution methods and the combined guided search on the same seed-42 sample of 100 ImageNet validation images for ViT-B/16 and ResNet-50. The evaluation includes all predicates associated with $J_c(x)$: 276 predicate occurrences across 99 eligible ViT images and 129 across 100 ResNet images. All methods use the same heatmap cutoffs, connected-region box construction, and predicate-deactivation test described above.

We report predicate success, the fraction of tested predicate occurrences for which an accepted region is found, and image coverage, the fraction of eligible images with at least one accepted region. To assess the size of the resulting explanations, we also report occupancy, the fraction of image area covered by an accepted box union. Occupancy is summarized over successful predicate tests.

**Attribution methods and search order.** Table 2 compares Grad-CAM, Score-CAM, and their sequential combination. Grad-CAM produces more compact accepted regions, with median occupancy of 23.0% for ViT and 41.2% for ResNet, compared with 33.1% and 57.9% for Score-CAM. Score-CAM, however, succeeds on more predicates. Their combination achieves the highest predicate success and image coverage among the evaluated methods, while keeping median occupancy closer to Grad-CAM.

Table 2: **Comparison of attribution methods for visual grounding.** All methods use the same 100-image sample per model and predicate-deactivation test. Image coverage is measured over images containing at least one eligible predicate. Median occupancy is computed over successful predicate tests. The combined search applies Grad-CAM first and Score-CAM only when the Grad-CAM search fails.

| Model | Attribution method | Predicate success | Image coverage | Median occupancy |
|---|---|---|---|---|
| ViT-B/16 | Grad-CAM | 242/276 (87.7%) | 98/99 (99.0%) | 23.0% |
| | Score-CAM | 254/276 (92.0%) | 97/99 (98.0%) | 33.1% |
| | Grad-CAM → Score-CAM | **274/276 (99.3%)** | **99/99 (100.0%)** | 24.0% |
| ResNet-50 | Grad-CAM | 95/129 (73.6%) | 79/100 (79.0%) | 41.2% |
| | Score-CAM | 109/129 (84.5%) | 86/100 (86.0%) | 57.9% |
| | Grad-CAM → Score-CAM | **116/129 (89.9%)** | **90/100 (90.0%)** | 46.3% |

Table 3: **Guided grounding versus area-matched search on 100 images per model.** The baseline mirrors both stages of the guided search with approximately area-matched proposals placed at random locations. Image coverage refers to the guided search. Occupancy is summarized over successful predicate tests as the median percentage of image area covered; brackets give the interquartile range.

| Model | Guided success | Baseline success | Image coverage | Guided occupancy | Baseline occupancy |
|---|---|---|---|---|---|
| ViT-B/16 | 274/276 (99.3%) | 256/276 (92.8%) | 99/99 (100%) | 24.0% [11.8, 50.0] | 27.9% [16.8, 56.4] |
| ResNet-50 | 116/129 (89.9%) | 105/129 (81.4%) | 90/100 (90.0%) | 46.3% [24.8, 70.8] | 60.7% [37.0, 78.7] |

Score-CAM recovers 32 of 34 failed Grad-CAM predicate tests for ViT and 21 of 34 for ResNet, showing that the two methods provide complementary proposals. These results support our choice to begin with Grad-CAM, which requires one backward pass and supplies most accepted regions, and invoke Score-CAM when needed. This ordering improves the availability of grounded explanations while avoiding the additional Score-CAM computation whenever Grad-CAM succeeds.

**Comparison with area-matched search.** To examine the benefit of attribution-guided proposals, we compare the combined search with a strong area-matched search baseline. At each cutoff, the baseline randomly places a rectangle with approximately the same area and enclosing aspect ratio as the guided box union. It mirrors both stages of the guided search and succeeds if either corresponding search finds a region that deactivates the predicate. Matching proposal area, search opportunities, and the acceptance test makes this a controlled comparison of the benefit of attribution-based spatial guidance.

For both architectures, the guided search deactivates more predicates than the area-matched baseline (Table 3). Its accepted regions also have lower median occupancy: 24.0% versus 27.9% for ViT and 46.3% versus 60.7% for ResNet. These results support our use of neuron-specific attribution to guide iterative refinement toward visual regions that pass the ablation test.

## D Additional analysis of feature use

This appendix provides the definitions, image populations, and additional results for the analysis in Section 4.1. For each model, $p_{c,(j,s)}$ represents the activation state of signed feature $(j, s)$ using a threshold $T_{c,(j,s)}$ fitted for predicted class $c$. Positive predicates are active at or above the minimum selected source activation, while negative predicates are active at or below the corresponding maximum. We apply the same feature selection procedure to source and held-out images, selecting compact sets whose contributions reproduce the original prediction. Thresholds are fitted on the selected source activations and remain fixed for held-out evaluation. Each image is analyzed using the predicates defined for its predicted class. The same features may be selected across different classes, with separate thresholds describing their activation states in each class.

We select source images using the correct-classification procedure described in Appendix B. Table 4 summarizes the resulting populations for each model. Held-out evaluation uses all 50,000 ImageNet validation images, grouped by model prediction. ConvNeXt has no source predicates for class 744,

Table 4: **Image populations and predicate counts.** Dim. gives the number of features supplied to the classification head, and Vocab. gives the median number of source predicates per predicted class. Source and Held-out report median active/selected predicate counts per image. Models follow the order used in the main figures.

| Model | Dim. | Source images | Vocab. | Source | Held-out |
|---|---|---|---|---|---|
| ViT-B/16 | 768 | 1,206,951 | 95 | 7/3 | 7/3 |
| ResNet-50 | 2,048 | 1,169,769 | 7 | 2/1 | 2/1 |
| ConvNeXt-Base | 1,024 | 1,216,462 | 81 | 9/4 | 9/4 |
| Swin-T | 768 | 1,153,631 | 133 | 15/7 | 15/7 |

so the 34 validation images predicted as this class are excluded from analyses requiring a source vocabulary. These analyses therefore use 49,966 ConvNeXt images. We compare distributions across models, with each model evaluated on its own predicted-class groups.

**Activation and selection frequencies.** Let $\mathcal{D}_c$ contain the source or held-out images assigned to predicted class $c$, with $|\mathcal{D}_c| > 0$. For each signed feature $(j, s) \in \mathcal{V}_c$, we measure its predicate's activation frequency and its feature's selection frequency:

$$f_{\text{act}}(c, j, s) = \frac{\sum_{x \in \mathcal{D}_c} p_{c,(j,s)}(x)}{|\mathcal{D}_c|}, \qquad f_{\text{sel}}(c, j, s) = \frac{\sum_{x \in \mathcal{D}_c} \mathbf{1}\{(j, s) \in S(x)\}}{|\mathcal{D}_c|}. \tag{12}$$

These frequencies describe the proportion of all images in the class for which the predicate is active or its feature is selected. We obtain $S(x)$ using the same feature selection procedure for both populations and evaluate predicates with their fixed source thresholds. For cross-class reuse, we count how many classes meet a given selection-frequency cutoff for the same feature $(j, s)$. Positive and negative activations are counted separately, and the predicates associated with a shared feature retain their class-specific thresholds.

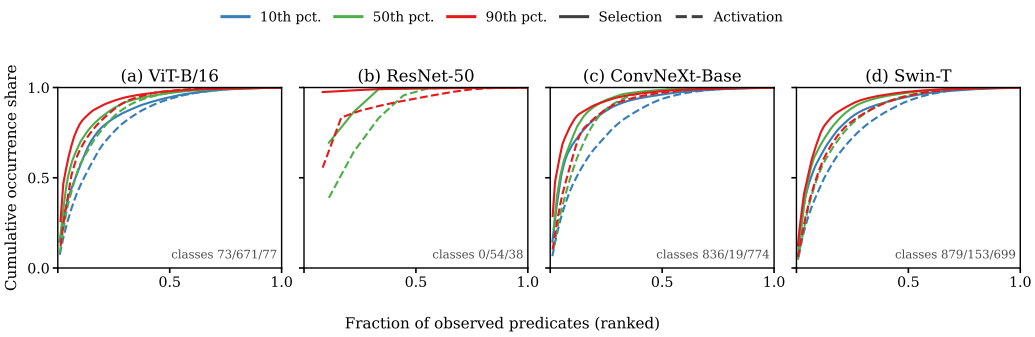



Figure 7: **Within-class concentration of predicate activations and selections on source images.** Predicates are ranked separately by activation and selection frequency, and curves show the cumulative share of occurrences. Colors identify classes nearest the 10th, 50th, and 90th percentiles of selection concentration, measured by $1 - N_{80}/|\mathcal{V}_c|$. Solid lines show selection and dashed lines activation; both axes are normalized to $[0, 1]$.

**Within-class concentration.** For ViT, ResNet, ConvNeXt, and Swin, the median source vocabulary contains 95, 7, 81, and 133 predicates, respectively. However, only a small subset accounts for most selections: the median number needed to cover 80% of within-class selected occurrences, $N_{80}$, is 16, 1.5, 11, and 23. The median fractions of the vocabulary needed to reach this coverage are 0.165, 0.222, 0.136, and 0.177 for selection, compared with 0.225, 0.364, 0.211, and 0.257 for activation. Thus, selections concentrate on fewer predicates than activations across all four models. Figure 7 illustrates classes near the 10th, 50th, and 90th percentiles of selection concentration. In the smaller held-out class groups, the relative concentration of selection and activation varies, so this ordering is clearest in the source results.

**Predicates on held-out images.** We assess how well the source predicates cover features selected on unseen images. A selected occurrence is *eligible* when the source vocabulary contains a predicate for its predicted class and signed feature. The eligible fractions of all held-out selected occurrences are 98.566%, 99.512%, 98.920%, and 99.468% for ViT, ResNet, ConvNeXt, and Swin, respectively. Among these eligible occurrences, 97.357%, 98.927%, 98.359%, and 98.428% also satisfy their fixed source thresholds; we call this *threshold retention*. Source retention is 100% because the thresholds are fitted from selected source activations. The high held-out coverage and retention show that most selected features on unseen images can be represented by the source predicates without refitting, supporting the feature-selection and predicate-construction procedure in Sections 3.1–3.2.

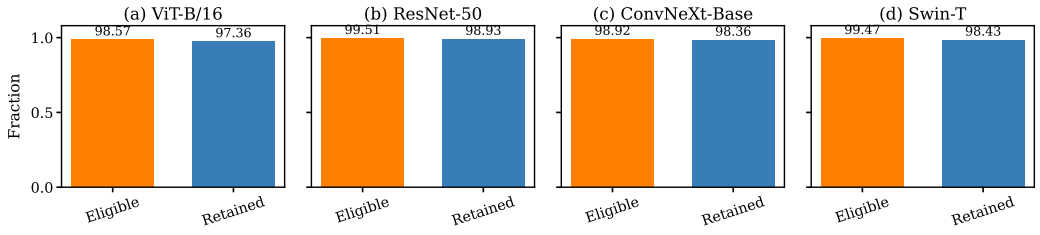



Figure 8: **Source predicate coverage and threshold retention on held-out images.** Eligible shows the percentage of all selected occurrences with a matching class-specific source predicate. Retained shows the percentage of eligible occurrences that satisfy the corresponding fixed threshold. Values are printed above the bars, showing how well the source predicates cover selected features and retain activation on unseen images.

# E HUMAN EVALUATION

## E.1 EXPERIMENT SETUP

We describe the evaluation setup in detail as follows.

**Explanation stimuli.** The VISIONLOGIC explanations shown to participants were generated using the bounding-box grounding procedure in Section 3.3, with the settings specified in Appendix C. We used the accepted bounding boxes to show the grounded concepts, without the optional foreground-segmentation overlays. The ACE and CRAFT explanations were also presented using bounding boxes. Using a common region-display format helps control presentation differences when comparing concept usefulness.

**Participants.** We recruited participants from Prolific (Prolific, 2024), a popular online platform for human evaluation of research projects. The users recruited have high qualifications as we requested a task acceptance rate greater than 98%. Our questionnaire was approved by the Social Sciences, Humanities & Education REB at University X, and all users provided informed consent before they started the experiment. The questionnaire was designed to be completed in 8-10 minutes and users who passed the screening received US $ 2.00 upon completion.

**Statistics.** For the Husky vs. Wolf scenario, $n = 161$ participants passed screening and filtering, respectively $n = 31, 25, 40, 35, 30$ for control, baseline, ACE, CRAFT and VISIONLOGIC.

For the Otter vs. Beaver scenario[1], $n = 114$ participants passed screening and filtering, respectively $n = 22, 22, 27, 25, 18$ for control, baseline, ACE, CRAFT, and VISIONLOGIC.

For the Kit Fox vs. Red Fox scenario, $n = 190$ participants passed screening and filtering, respectively $n = 35, 35, 45, 40, 35$ for control, baseline, ACE, CRAFT, and VISIONLOGIC.

[1]This task replaces the legacy "Leaves" dataset from Colin et al. (2022) that is no longer available.

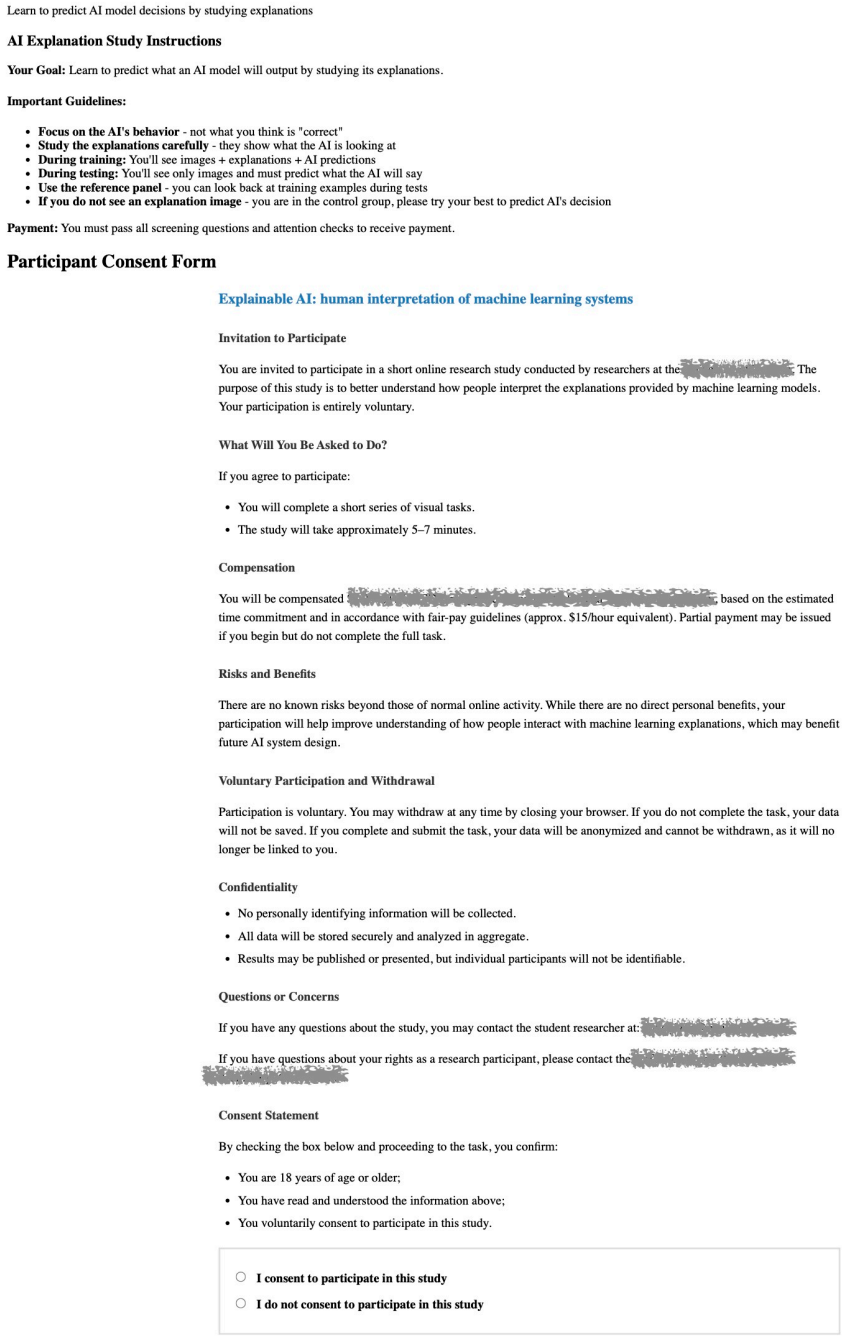
**Study design.** We followed the experimental design proposed by Colin et al. (2022), which quantitatively assesses to what degree a concept-based explanation method can help a human observer understand the behavior of an AI model. Each participant is only tested on a single condition (control, baseline, ACE, CRAFT, or VISIONLOGIC) to avoid possible experimental confounds.

The screening, training, and testing phases follow the procedure described in Colin et al. (2022). After granting consent (shown in Figure 9), participants are presented with detailed instructions and the study goal: learning to predict which class *an AI model will predict* for a given image. They start with a practice session that shows simple images along with their explanations and model prediction, followed by some unseen images without explanations, where they are expected to answer the model's prediction correctly. Participants who failed this session were not allowed to proceed with the study. They are then tested on their understanding of the study goal by a short quiz (exactly formulated as in Colin et al. (2022)). Again, participants who failed this session were not allowed to proceed.

After the above two screening phases, participants went through 3 sessions of training phases (5, 10, and 15 images, respectively, with explanations and model decisions) followed by a testing phase (7 new images without explanation). The answers for the testing phases are collected to compute the main result. Figure 10 shows an example of the training phase. As described in Colin et al. (2022), we implemented a reservoir (Figure 11) for training images as a reference point for participants during the testing phase. The last test image of each session is an image from the reservoir, and participants who incorrectly answered the last test question were filtered out.

### E.2 STATISTICAL TEST RESULTS FOR SIGNIFICANCE

We provide the statistical test results in Figures 12, 13, 14, 15, 16, and 17.



Learn to predict AI model decisions by studying explanations

**AI Explanation Study Instructions**

**Your Goal:** Learn to predict what an AI model will output by studying its explanations.

**Important Guidelines:**

- **Focus on the AI's behavior** - not what you think is "correct"
- **Study the explanations carefully** - they show what the AI is looking at
- **During training:** You'll see images + explanations + AI predictions
- **During testing:** You'll see only images and must predict what the AI will say
- **Use the reference panel** - you can look back at training examples during tests
- **If you do not see an explanation image** - you are in the control group, please try your best to predict AI's decision

**Payment:** You must pass all screening questions and attention checks to receive payment.

**Participant Consent Form**

**Explainable AI: human interpretation of machine learning systems**

**Invitation to Participate**

You are invited to participate in a short online research study conducted by researchers at the [redacted]. The purpose of this study is to better understand how people interpret the explanations provided by machine learning models. Your participation is entirely voluntary.

**What Will You Be Asked to Do?**

If you agree to participate:

- You will complete a short series of visual tasks.
- The study will take approximately 5–7 minutes.

**Compensation**

You will be compensated [redacted], based on the estimated time commitment and in accordance with fair-pay guidelines (approx. $15/hour equivalent). Partial payment may be issued if you begin but do not complete the full task.

**Risks and Benefits**

There are no known risks beyond those of normal online activity. While there are no direct personal benefits, your participation will help improve understanding of how people interact with machine learning explanations, which may benefit future AI system design.

**Voluntary Participation and Withdrawal**

Participation is voluntary. You may withdraw at any time by closing your browser. If you do not complete the task, your data will not be saved. If you complete and submit the task, your data will be anonymized and cannot be withdrawn, as it will no longer be linked to you.

**Confidentiality**

- No personally identifying information will be collected.
- All data will be stored securely and analyzed in aggregate.
- Results may be published or presented, but individual participants will not be identifiable.

**Questions or Concerns**

If you have any questions about the study, you may contact the student researcher at: [redacted]

If you have questions about your rights as a research participant, please contact the [redacted]

**Consent Statement**

By checking the box below and proceeding to the task, you confirm:

- You are 18 years of age or older;
- You have read and understood the information above;
- You voluntarily consent to participate in this study.

○ **I consent to participate in this study**

○ **I do not consent to participate in this study**

Figure 9: The online questionnaire begins with a consent form.




Figure 10: The online questionnaire displaying an example of the training session with the original image, the explanation, and the model prediction.

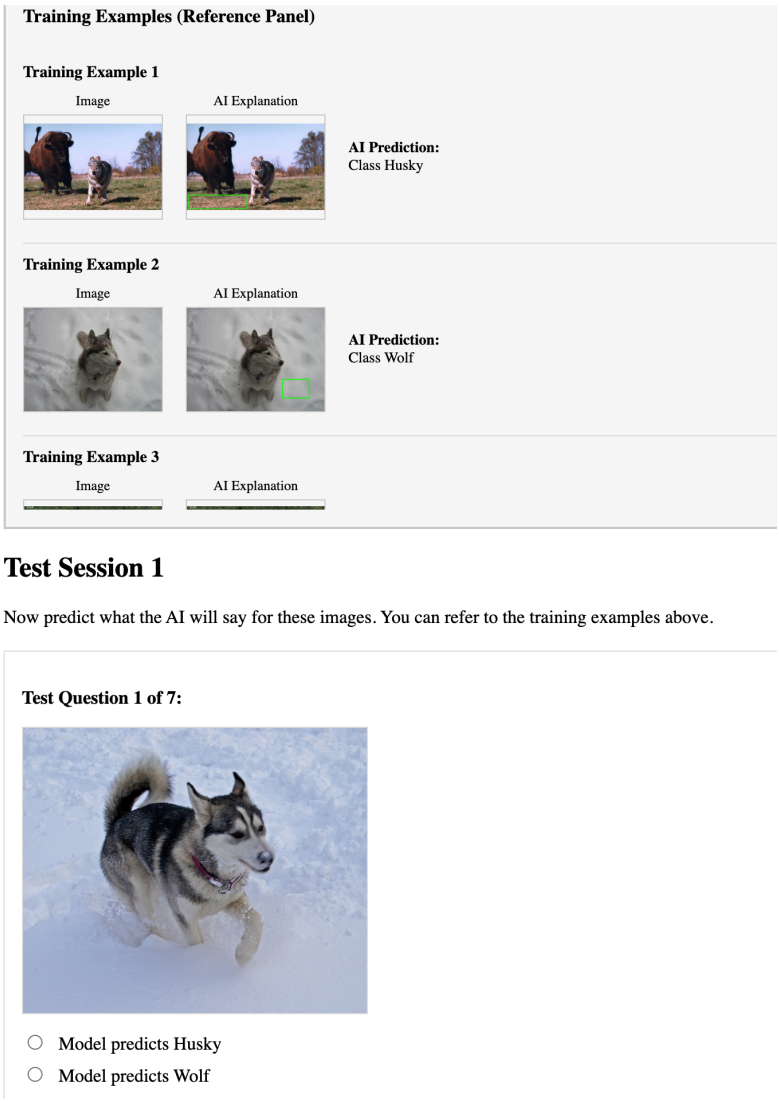



Figure 11: The online questionnaire displaying the testing session with a reservoir containing all examples during the training phase.

```
Starting analysis for EXP1
======================================================================
Loading exp1 data files...
============================================================
✓ Control : 31 participants loaded from Control_exp1_individual_results.csv
✓ Baseline: 25 participants loaded from Baseline_exp1_individual_results.csv
✓ ACE     : 40 participants loaded from ACE_exp1_individual_results.csv
✓ CRAFT   : 35 participants loaded from CRAFT_exp1_individual_results.csv
✓ LOGIC   : 30 participants loaded from LOGIC_exp1_individual_results.csv

Total participants loaded: 161
Methods found: ['ACE', 'Baseline', 'CRAFT', 'Control', 'LOGIC']
============================================================
NORMALITY TEST (Shapiro-Wilk)
============================================================
H0: Data is normally distributed
H1: Data is not normally distributed
Significance level: 0.05
------------------------------------------------------------
Control  | n=31 | W=0.9372 | p=0.0689 | Normal
Baseline | n=25 | W=0.9521 | p=0.2790 | Normal
ACE      | n=40 | W=0.9404 | p=0.0357 | Not Normal
CRAFT    | n=35 | W=0.9376 | p=0.0473 | Not Normal
LOGIC    | n=30 | W=0.8189 | p=0.0001 | Not Normal

Overall normality assumption: ✗ VIOLATED
============================================================
HOMOGENEITY OF VARIANCE TEST (Levene's Test)
============================================================
H0: All groups have equal variances
H1: At least one group has different variance
Significance level: 0.05
------------------------------------------------------------
Levene's statistic: 1.3810
p-value: 0.2430
Homogeneity assumption: ✓ MET

Group variances:
Method    |  n | Variance |    SD    |    SE
-------------------------------------------------
Control   | 31 |   0.0334 |   0.1827 |   0.0328
Baseline  | 25 |   0.0385 |   0.1962 |   0.0392
ACE       | 40 |   0.0593 |   0.2435 |   0.0385
CRAFT     | 35 |   0.0317 |   0.1780 |   0.0301
```

Figure 12: Complete statistical test results for scenario 1, page 1.

```
LOGIC    | 30 |   0.0299 |   0.1730 |   0.0316

============================================================
NON-PARAMETRIC ANALYSIS
============================================================
============================================================
KRUSKAL-WALLIS TEST (Non-parametric)
============================================================
H0: All group distributions are identical
H1: At least one group distribution differs
Significance level: 0.05
------------------------------------------------------------
H-statistic: 25.8440
p-value: 0.000034
Degrees of freedom: 4
Result: ✓ SIGNIFICANT

→ Significant differences found between method distributions.
→ Proceeding to Dunn's post-hoc test...

============================================================
DUNN'S TEST (Non-parametric post-hoc)
============================================================
Pairwise p-values (Bonferroni corrected):
              ACE  Baseline   CRAFT  Control   LOGIC
ACE        1.0000    1.0000  0.5956   1.0000  0.0241
Baseline   1.0000    1.0000  1.0000   1.0000  0.0303
CRAFT      0.5956    1.0000  1.0000   1.0000  0.0000
Control    1.0000    1.0000  1.0000   1.0000  0.0004
LOGIC      0.0241    0.0303  0.0000   0.0004  1.0000

Pairwise Comparisons Summary (Bonferroni corrected, α = 0.05):
======================================================================
ACE      vs Baseline | p:  1.0000
ACE      vs CRAFT    | p:  0.5956
ACE      vs Control  | p:  1.0000
ACE      vs LOGIC    | p:  0.0241 ***
Baseline vs CRAFT    | p:  1.0000
Baseline vs Control  | p:  1.0000
Baseline vs LOGIC    | p:  0.0303 ***
CRAFT    vs Control  | p:  1.0000
CRAFT    vs LOGIC    | p:  0.0000 ***
Control  vs LOGIC    | p:  0.0004 ***
```

Figure 13: Complete statistical test results for scenario 1, page 2.

```
Starting analysis for EXP2
======================================================================
Loading exp2 data files...
============================================================
✓ Control : 22 participants loaded from Control_exp2_individual_results.csv
✓ Baseline: 22 participants loaded from Baseline_exp2_individual_results.csv
✓ ACE     : 27 participants loaded from ACE_exp2_individual_results.csv
✓ CRAFT   : 25 participants loaded from CRAFT_exp2_individual_results.csv
✓ LOGIC   : 18 participants loaded from LOGIC_exp2_individual_results.csv

Total participants loaded: 114
Methods found: ['ACE', 'Baseline', 'CRAFT', 'Control', 'LOGIC']
============================================================
NORMALITY TEST (Shapiro-Wilk)
============================================================
H0: Data is normally distributed
H1: Data is not normally distributed
Significance level: 0.05
------------------------------------------------------------
Control  | n=22 | W=0.7308 | p=0.0000 | Not Normal
Baseline | n=22 | W=0.6701 | p=0.0000 | Not Normal
ACE      | n=27 | W=0.8543 | p=0.0014 | Not Normal
CRAFT    | n=25 | W=0.8510 | p=0.0018 | Not Normal
LOGIC    | n=18 | W=0.6087 | p=0.0000 | Not Normal

Overall normality assumption: ✗ VIOLATED
============================================================
HOMOGENEITY OF VARIANCE TEST (Levene's Test)
============================================================
H0: All groups have equal variances
H1: At least one group has different variance
Significance level: 0.05
------------------------------------------------------------
Levene's statistic: 2.5161
p-value: 0.0455
Homogeneity assumption: ✗ VIOLATED

Group variances:
Method    |  n | Variance |    SD    |    SE
-------------------------------------------------
Control  |  22 |   0.0230 |   0.1517 |   0.0323
Baseline |  22 |   0.0282 |   0.1678 |   0.0358
ACE      |  27 |   0.0222 |   0.1491 |   0.0287
CRAFT    |  25 |   0.0123 |   0.1111 |   0.0222
```

Figure 14: Complete statistical test results for scenario 2, page 1.

```
LOGIC    | 18 |   0.0011 |   0.0332 |   0.0078

============================================================
NON-PARAMETRIC ANALYSIS
============================================================

============================================================
KRUSKAL-WALLIS TEST (Non-parametric)
============================================================
H0: All group distributions are identical
H1: At least one group distribution differs
Significance level: 0.05
------------------------------------------------------------
H-statistic: 13.5628
p-value: 0.008830
Degrees of freedom: 4
Result: ✓ SIGNIFICANT

→ Significant differences found between method distributions.
→ Proceeding to Dunn's post-hoc test...
============================================================
DUNN'S TEST (Non-parametric post-hoc)
============================================================
Pairwise p-values (Bonferroni corrected):
              ACE  Baseline   CRAFT  Control   LOGIC
ACE        1.0000    1.0000  1.0000   1.0000  0.0045
Baseline   1.0000    1.0000  1.0000   1.0000  0.1556
CRAFT      1.0000    1.0000  1.0000   1.0000  0.0309
Control    1.0000    1.0000  1.0000   1.0000  0.1463
LOGIC      0.0045    0.1556  0.0309   0.1463  1.0000

Pairwise Comparisons Summary (Bonferroni corrected, α = 0.05):
======================================================================
ACE      vs Baseline | p:  1.0000
ACE      vs CRAFT    | p:  1.0000
ACE      vs Control  | p:  1.0000
ACE      vs LOGIC    | p:  0.0045 ***
Baseline vs CRAFT    | p:  1.0000
Baseline vs Control  | p:  1.0000
Baseline vs LOGIC    | p:  0.1556
CRAFT    vs Control  | p:  1.0000
CRAFT    vs LOGIC    | p:  0.0309 ***
Control  vs LOGIC    | p:  0.1463
```

Figure 15: Complete statistical test results for scenario 2, page 2.

```
Starting analysis for EXP3
======================================================================
Loading exp3 data files...
============================================================
✓ Control : 35 participants loaded from Control_exp3_individual_results.csv
✓ Baseline: 35 participants loaded from Baseline_exp3_individual_results.csv
✓ ACE     : 45 participants loaded from ACE_exp3_individual_results.csv
✓ CRAFT   : 40 participants loaded from CRAFT_exp3_individual_results.csv
✓ LOGIC   : 35 participants loaded from LOGIC_exp3_individual_results.csv

Total participants loaded: 190
Methods found: ['ACE', 'Baseline', 'CRAFT', 'Control', 'LOGIC']
============================================================
NORMALITY TEST (Shapiro-Wilk)
============================================================
H0: Data is normally distributed
H1: Data is not normally distributed
Significance level: 0.05
------------------------------------------------------------
Control  | n=35 | W=0.9552 | p=0.1641 | Normal
Baseline | n=35 | W=0.9030 | p=0.0048 | Not Normal
ACE      | n=45 | W=0.9359 | p=0.0153 | Not Normal
CRAFT    | n=40 | W=0.8808 | p=0.0006 | Not Normal
LOGIC    | n=35 | W=0.9488 | p=0.1038 | Normal

Overall normality assumption: ✗ VIOLATED
============================================================
HOMOGENEITY OF VARIANCE TEST (Levene's Test)
============================================================
H0: All groups have equal variances
H1: At least one group has different variance
Significance level: 0.05
------------------------------------------------------------
Levene's statistic: 1.7067
p-value: 0.1503
Homogeneity assumption: ✓ MET

Group variances:
Method    |  n | Variance |    SD    |    SE
-------------------------------------------------
Control  |  35 |   0.0133 |   0.1151 |   0.0195
Baseline |  35 |   0.0100 |   0.1000 |   0.0169
ACE      |  45 |   0.0122 |   0.1106 |   0.0165
CRAFT    |  40 |   0.0270 |   0.1642 |   0.0260
```

Figure 16: Complete statistical test results for scenario 3, page 1.

```
LOGIC    | 35 |   0.0134 |   0.1156 |   0.0195

============================================================
NON-PARAMETRIC ANALYSIS
============================================================

============================================================
KRUSKAL-WALLIS TEST (Non-parametric)
============================================================
H0: All group distributions are identical
H1: At least one group distribution differs
Significance level: 0.05
------------------------------------------------------------
H-statistic: 9.1296
p-value: 0.057940
Degrees of freedom: 4
Result: ✗ NOT SIGNIFICANT

→ No significant differences found between method distributions.

============================================================
DUNN'S TEST (Non-parametric post-hoc)
============================================================
Pairwise p-values (Bonferroni corrected):
              ACE  Baseline   CRAFT  Control  LOGIC
ACE        1.0000    0.1503  1.0000   1.0000    1.0
Baseline   0.1503    1.0000  0.4246   0.1384    1.0
CRAFT      1.0000    0.4246  1.0000   1.0000    1.0
Control    1.0000    0.1384  1.0000   1.0000    1.0
LOGIC      1.0000    1.0000  1.0000   1.0000    1.0

Pairwise Comparisons Summary (Bonferroni corrected, α = 0.05):
======================================================================
ACE      vs Baseline | p:  0.1503
ACE      vs CRAFT    | p:  1.0000
ACE      vs Control  | p:  1.0000
ACE      vs LOGIC    | p:  1.0000
Baseline vs CRAFT    | p:  0.4246
Baseline vs Control  | p:  0.1384
Baseline vs LOGIC    | p:  1.0000
CRAFT    vs Control  | p:  1.0000
CRAFT    vs LOGIC    | p:  1.0000
Control  vs LOGIC    | p:  1.0000
```

Figure 17: Complete statistical test results for scenario 3, page 2. The overall test does not reject at the 0.05 level.