
OPTIMUS-Prime: Minimal & Sufficient Concept Explanations for Deep Vision Models

Arthur Hoarau, Chenrui Zhu, Vu-Linh Nguyen

Abstract

The growing demand for transparency in automated decision-making has propelled eXplainable Artificial Intelligence (XAI) to the forefront of machine learning research. In computer vision, however, existing explanation methods often prioritize end-user accessibility at the expense of formal guarantees, leaving a critical gap between practical utility and theoretical rigor. In this paper, we address this gap by introducing OPTIMUS, a novel framework for generating concept-based visual explanations for deep classification models. OPTIMUS explanations take the form of visual heatmaps that not only remain interpretable to end users, but are grounded in the well-established theory of prime implicants, providing formal guarantees that have been largely absent from existing saliency-based methods. Specifically, OPTIMUS explanations satisfy two desirable properties: sufficiency, ensuring that the highlighted concepts provably guarantee the classifier’s prediction, and minimality, ensuring that no strict subset of those concepts retains this guarantee. Together, these properties yield explanations that are both logically tight and visually coherent. We validate our approach on a visual classification benchmark, demonstrating that OPTIMUS heatmaps naturally and faithfully surface the decision-relevant concepts underlying model predictions.

1 Introduction

The increasing complexity of modern machine learning architectures has created a fundamental tension between predictive performance and interpretability, driving significant interest in explainability methods. A central challenge in deep vision is finding the right balance between explanations that are theoretically grounded from a learning perspective and those that remain accessible to end users.

A common approach to model interpretability relies on saliency maps, which highlight the input regions most influential to a model’s prediction. Several strategies have been proposed in the literature. A first consists in perturbing the inputs and observing the resulting changes in deeper neuron activations [28, 29]. A second strategy, exemplified by Grad-CAM and Guided Grad-CAM [22], is specific to convolutional neural networks (CNNs) and leverages the gradient of a target class score, back-propagated to the last convolutional layer, to produce a coarse localization map. In this paper, we focus on a third strategy, in which the gradient of the predicted class score is back-propagated to the input space in a single forward-backward pass [25]. Intuitively, a high saliency value at pixel i indicates that a small perturbation at that location would significantly alter the model’s output, suggesting that the model relies on that region to reach its decision. Such gradient-based methods, however, lack formal guarantees and can exhibit sensitivity to input noise and model parameter choices. To address some of these limitations, several extensions have been proposed, including Integrated Gradients [27], which accumulates gradients along a path from a baseline to the input, and DeepLIFT [24], which attributes contributions by comparing neuron activations to a reference. Both methods will be considered in this article. Alternatively, concept-based explanation methods such

Link to the code: <https://anonymous.4open.science/r/optimus-prime-075D>

as TCAV [11] probe the network’s internal representations by testing the sensitivity of predictions to user-defined concept directions, offering a higher-level abstraction than pixel-wise saliency.

These techniques belong to a broader family of methods aimed at enhancing the interpretability of machine learning models, collectively known as eXplainable Artificial Intelligence (XAI) [5]. Saliency heatmaps are *post-hoc*, applied after and independently from the training procedure, and partially *model-agnostic*, in the sense that they do not assume a specific internal architecture. The latter qualification is only partially accurate: several backpropagation-based techniques require specific architectures or become intractable for others, yielding *model-specific* explanations, as opposed to purely *model-agnostic* approaches such as SHAP [16] or LIME [21].

In a general classification setting, deep neural networks act as hierarchical feature extractors [15], building increasingly abstract representations of the input space to facilitate decision-making at deeper layers. In this work, we leverage the notion of *concepts* as the semantic information captured at a given layer [2, 3, 10]. In the first layer, concepts correspond to raw pixels while in the final layer, they correspond to class labels (e.g., “dog” or “cat” in an animal classification task). Intermediate layers capture concepts of varying abstraction [20], potentially encoding semantically meaningful features such as “pointy ears” or “vertical pupils”, though these representations can be highly entangled, subject to superposition [7, 4], and remain difficult to define precisely.

In the formal XAI literature, *prime implicants* provide sufficient and subset-minimal explanations for Boolean classifiers [18, 8, 23]: fixing the variables in the explanation is enough to guarantee the model’s prediction, and no proper subset of these variables preserves this guarantee. Transposed to our setting, variables become concepts, and the Boolean function becomes the classifier’s decision rule. An OPTIMUS prime implicant explanation thus identifies the subset-minimal set of concepts that provably guarantees the prediction. This explanation can then be backpropagated into the input space, yielding representations that are both human-interpretable and endowed with formal sufficiency and minimality guarantees.

A closely related contribution was recently published [26], in which the authors propose sufficient and minimal explanations for deep vision. Their approach, however, is restricted to prototype-based network architectures [6], making it strongly *model-specific*. As a consequence, their prime implicant formulation is inherently tied to the prototype structure and cannot be directly applied to general deep networks. We relax this constraint and extend such explanations to a broader class of deep neural networks, moving toward more *model-agnostic* explanations. Furthermore, their method does not backpropagate explanations into the input space and omits end-user visualizations, relying solely on numerical metrics, limiting the intuitive accessibility of the results. Nevertheless, their work introduces the compelling notion of *abductive latent explanations*, arguing that “the pixel-level is not the correct level of abstraction for the human final user of the explanation”. We strongly subscribe to this view, and build upon it throughout this paper.

The remainder of this paper is structured as follows. Section 2 details the proposed OPTIMUS framework. Section 2.1 formalizes the notion of concepts. Section 2.2 extends prime implicants to neural network concepts and introduces a direct search algorithm, while Section 2.3 addresses their backpropagation to the input space. Section 3 validates the framework on a visual Cat-Dog-Bird classification benchmark, demonstrating the effectiveness of OPTIMUS explanations in practice. Section 4 discusses limitations and open challenges, and Section 5 concludes the article.

2 OPTIMUS-Prime

The OPTIMUS-Prime pipeline is illustrated in Figure 1, which operates in two stages. First, given a prediction from a deep vision model, a linear analysis is performed in the latent concept space of an intermediate layer. Leveraging the linearity of the final classification layer, this analysis identifies a prime implicant (PI): a minimal and sufficient subset of concepts, that provably guarantees the predicted class over all rivals. Second, the selected concepts are projected back into the input space via gradient-based attribution methods, namely Integrated Gradients or DeepLIFT, which attribute each pixel’s relevance with respect to the selected concepts only. The result is an OPTIMUS heatmap in which highlighted regions directly correspond to the minimal and sufficient concepts underlying the model’s decision. This is what gives OPTIMUS its edge over saliency maps: rather than measuring influence, it identifies logical sufficiency with a minimality guarantee.

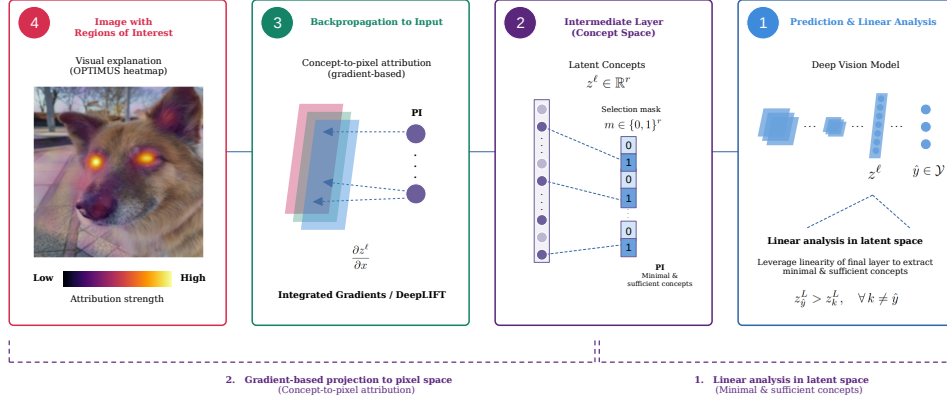


Figure 1: OPTIMUS-Prime: A linear analysis in latent space identifies a prime implicant (PI), a subset of concepts encoded by a selection mask (1-2). These concepts are then backpropagated to the input space via Integrated Gradients or DeepLIFT (3), yielding a visual heatmap where highlighted regions directly correspond to the concepts driving the model’s prediction (4).

The following sections formally detail each component of the framework. Section 2.1 introduces the notion of concepts as the semantic information captured at a given layer. Section 2.2 then defines prime implicants in the context of neural network classifiers and proposes an efficient direct search algorithm. Finally, Section 2.3 addresses the backpropagation of the selected concepts into the input space, bridging the gap between latent guarantees and human-interpretable visualizations.

2.1 OPTIMUS: Optimal concept explanations

We consider a multiclass classification setup with input space $\mathcal{X} \subseteq \mathbb{R}^P$, where P denotes the input dimension, and the label space is $\mathcal{Y} := \{1, \dots, K\}$. We denote by (X, Y) a pair of random variables and (x_n, y_n) i.i.d draws from the unknown joint distribution $P_{X,Y}$ forming a dataset $\mathcal{D} := \{(x_n, y_n)\}_{n=1}^N$ of size N . A neural network classifier is a measurable function $h : \mathcal{X} \rightarrow \mathcal{Y}$, which is implemented as a composition of L layers. For any given input $x \in \mathcal{X}$, we denote $\hat{y}(x) := h(x) \in \mathcal{Y}$ the predicted class. We denote by $z^\ell(x) \in \mathcal{Z}^\ell \subseteq \mathbb{R}^{r^\ell}$ the output of the ℓ -th hidden layer, for $\ell \in \{1, \dots, L-1\}$. We assume that the network can be decomposed as

$$h(x) = \arg \max_{k \in \{1, \dots, K\}} z_k^L(x), \quad (1)$$

where $z^L \in \mathcal{Z}^L \subseteq \mathbb{R}^K$ is the output of the final linear layer, producing one logit z_k^L per class.

Definition 1 (Concept). A concept $\circ^\ell \in \{1, \dots, r^\ell\}$ reflects the semantic information captured in order to activate the $\circ^\ell$ -th neuron $z_{\circ^\ell}^\ell$ in layer ℓ .

The concepts encoded at the input layer $z^0(x) = x$ correspond to raw pixels, while those at the final layer z^L correspond to class-level representations (e.g., the concept “cat” in an animal classification task). One might expect that concepts become increasingly abstract with depth. In practice, however, intermediate-layer concepts rarely align naturally with human-interpretable features [4], such as “pointy ears” or “vertical pupils”, even in tasks where such features are semantically meaningful. Multiple definitions have been proposed to characterize such concepts [7, 26], and we deliberately adopt a minimal one, free of any human-interpretable requirements, so as to preserve the generality of our framework. A related line of work constrains the network architecture itself to produce human-interpretable concept predictions as intermediate outputs [12], at the cost of requiring concept-level supervision at training time, a constraint our framework deliberately avoids.

We introduce OPTIMUS, a novel explanation framework that provides users with a heatmap grounded in a minimal and sufficient set of concepts to preserve the model prediction.

Definition 2 (OPTIMUS explanation). A matrix $\mathbf{m} \in \mathbb{R}^{H \times W}$ where each entry $\mathbf{m}_{i,j}$ reflects the relevance of pixel (i, j) w.r.t. a set of concepts $\mathcal{O}^\ell \subseteq \{1, \dots, r^\ell\}$ in a latent space $\ell \in \{1, \dots, L\}$, is an OPTIMUS explanation if and only if $\mathcal{O}^\ell$ is a prime implicant at layer ℓ .

This definition leverages the notion of prime implicants, formally introduced in the next section, to ensure both the minimality and sufficiency of the highlighted concepts w.r.t. the model prediction. An OPTIMUS is then a human-interpretable projection of minimal latent concepts in the input space.

2.2 Multiclass prime implicants

We now introduce the notion of prime implicants. Informally, a PI is a subset-minimal set of variables sufficient to guarantee the prediction of the model, regardless of the values taken by the remaining variables.

Let $f : \mathcal{Z} \rightarrow \mathbb{R}^K$ be a linear multiclass classifier function such that

$$f_k(x) = \sum_{q=1}^r x_q w_q^{(k)} + b^{(k)}, \quad k \in \{1, \dots, K\}, \quad (2)$$

defined on any multi-dimensional box (in this article, we consider the unit hypercube). Let consider an instance $x^* \in [0, 1]^r$ such that class c is predicted

$$f_c(x^*) > f_k(x^*), \quad \forall k \in \{1, \dots, K\} \setminus c. \quad (3)$$

For each rival class $k \in \{1, \dots, K\} \setminus c$, let respectively define the dominance weights and biases

$$\Delta_q^{(k)} := w_q^{(c)} - w_q^{(k)}, \quad \delta^{(k)} := b^{(c)} - b^{(k)}. \quad (4)$$

The dominance margin between class c and class k is defined as

$$f_c(x) - f_k(x) = \delta^{(k)} + \sum_{q=1}^r x_q \Delta_q^{(k)}. \quad (5)$$

Let $I = \{1, \dots, r\}$ be the set of all feature indices. For a given input $x^* \in \mathcal{Z}$ and a subset $S \subseteq I$, we interpret S as the set of fixed features, whose values are locked to x_q^* , while the remaining features $I \setminus S$ are free to vary in $[0, 1]^r$. The worst-case margin for class k under this partial freedom is defined as

$$\Gamma_k(S) := \delta^{(k)} + \sum_{q \in S} x_q^* \Delta_q^{(k)} + \sum_{q \in I \setminus S} \min_{v \in [0, 1]} v \Delta_q^{(k)}. \quad (6)$$

Definition 3 (Multiclass implicant). A candidate $S \subseteq I$ is a multiclass implicant for class c at x^* iff $\forall x \in [0, 1]^r$ such that $\forall q \in S, x_q = x_q^*$

$$f_c(x) > f_k(x), \quad \forall k \in \{1, \dots, K\} \setminus c. \quad (7)$$

Equivalently, S is a multiclass implicant iff

$$\Gamma_k(S) > 0, \quad \forall k \in \{1, \dots, K\} \setminus c. \quad (8)$$

Definition 4 (Multiclass prime implicant). A subset $S \subseteq I$ is a multiclass prime implicant (PI) for class c at x^* iff S is an implicant¹ and

$$\exists q \in S, \exists k \in \{1, \dots, K\} \setminus c, \quad \Gamma_k(S \setminus \{q\}) \leq 0. \quad (9)$$

A multiclass PI is a subset-minimal set of features ensuring dominance over all rival classes simultaneously. Recent extensions of formal explanations to more expressive classifiers [9] further motivate the need for scalable PI search procedures in the multiclass setting.

¹A singleton implicant is considered prime.

2.2.1 Prime implicant search

Finding a prime implicant in the linear binary classifier has been studied by [19]. Their work also considers the search for a cardinality-minimum PI in the binary setting. The objective is to obtain an PI explanation as small as possible while still guaranteeing the prediction. This is particularly relevant in image and visual classification, where large explanations may be difficult to interpret. Inspired by this approach, we extend the search procedure to the multiclass setting.

Given an instance x^* , we propose the following direct search by starting from the complete set of features I . Let $c = \arg \max_k f_k(x^*)$ be the predicted class of the classifier. For each rival class $k \neq c$ let define

$$a_k := \delta^{(k)} + \sum_{q=1}^r \min_{v \in [0,1]} v \Delta_q^{(k)}, \quad g_q^{(k)} := x_q^* \Delta_q^{(k)} - \min_{v \in [0,1]} v \Delta_q^{(k)}, \quad (10)$$

such that the worst-case margin can be rewritten as

$$\Gamma_k(S) = a_k + \sum_{q \in S} g_q^{(k)}. \quad (11)$$

A feature $q \in S$ can then be removed if $\Gamma_k(S \setminus \{q\}) > 0, \forall k \neq c$, which can be simplified for computation as

$$a_k + \sum_{j \in S \setminus \{q\}} g_j^{(k)} > 0, \quad \forall k \in \{1, \dots, K\} \setminus c. \quad (12)$$

Proposition 1. *Assume that x^* is classified as class c . Starting from $S = I$, by repeatedly removing any feature $q \in S$ such that $\forall k \neq c \quad \Gamma_k(S \setminus \{q\}) > 0$, the resulting set S is a multiclass PI when no further removal can guarantee the constraint.*

Note, however, that the final PI depends on the order in which removable features are examined.

Proof for Proposition 1 is provided in Appendix B. The quantities a_k and $g_q^{(k)}$ can be precomputed for all $q \in I$ and all rival classes $k \in \{1, \dots, K\} \setminus c$ in $O(Kr)$ time. Testing whether a feature $q \in S$ can be removed costs $O(K)$. Hence, pruning features in a fixed order costs $O(Kr)$.

In the binary case, variables are ranked once and for all according to their margin contribution, and then greedily selected in decreasing order of contribution, or, similar to Proposition 1, removed in ascending order of contribution. In the multiclass case, however, each variable contributes differently to the various pairwise margin, therefore no single static ordering is available. Hence, the search for a cardinality-minimum multiclass PI can be formulated as a combinatorial optimization problem. Let introduce binary variables $s_q \in \{0, 1\}$, where $s_q = 1$ feature q is selected. The problem becomes

$$\begin{aligned} & \min_{s \in \{0,1\}^r} \sum_{q=1}^r s_q, \\ & \text{subject to} \quad a_k + \sum_{q=1}^r s_q g_q^{(k)} > 0, \quad \forall k \neq c. \end{aligned} \quad (13)$$

Any optimal solution is a multiclass implicant of minimum cardinality.

Note that an exact solution to find a minimum-cardinality multiclass PI is possible via mixed-integer linear programming by replacing strict inequality 0 by some fixed tolerance $\varepsilon > 0$. Nevertheless, the problem remains combinatorial in general and becomes impractical for large feature sets. We therefore propose some adaptive greedy rules. One can select among removable features. We thus define the LOWGAIN as

$$q^* \in \arg \min_{q \in R(S)} \sum_{k \neq c} g_q^{(k)}, \quad (14)$$

which removes the features with the smallest total contribution to the margins. Elements $g_q^{(k)}$ can be precomputed in $O(Kr)$ and sorted once in $O(r \log r)$. Hence, the total complexity is $O(Kr + r \log r)$. We also define an alternative MAXMIN rule as

$$q^* \in \arg \max_{q \in R(S)} \min_{k \neq c} \Gamma_k(S \setminus \{q\}), \quad (15)$$

which removes the feature that keeps the remaining worst-case margin as large as possible. Recomputing such a greedy choice by scanning all candidates at each step yields a worst-case complexity of $O(Kr^2)$. The algorithm for finding a multiclass PI is provided in Appendix C.

2.2.2 OPTIMUS Prime implicant

This section outlines the assumptions and requirements under which the proposed framework maintains its theoretical guarantees.

An OPTIMUS prime implicant is defined at a layer ℓ that is imperatively followed by a linear function. This is naturally satisfied at layer $L - 1$, which is directly followed by the final linear layer, an architecture adopted by most neural networks. The second requirement is more restrictive: the relative input space must be bounded. This can be enforced by applying a sigmoid activation function at the penultimate layer, constraining the representations to the hypercube $[0, 1]^{r^{L-1}}$. However, such a restriction may be too costly in practice, as modern activation functions such as ReLU are positively unbounded. One possible remedy is to replace the sigmoid by an unbounded activation and instead sample the instance space to derive statistical bounds, at the cost of relaxing our exact theoretical guarantees to weaker PAC statistical guarantees.

An OPTIMUS prime implicant $\mathcal{O}^\ell$ provides the following guarantee. For an observation x^* , $\forall z^\ell(x) \in [0, 1]^{r^\ell}$ such that $z_o^\ell(x) = z_o^\ell(x^*)$ for all $o \in \mathcal{O}^\ell$

$$z_{\hat{y}(x^*)}^L(x) > z_k^L(x), \quad \forall k \neq \hat{y}(x^*), \quad (16)$$

and $\mathcal{O}^\ell$ is minimal with respect to this property.

In other words, the concepts $o \in \mathcal{O}^\ell$ are sufficient to guarantee the prediction $\hat{y}(x^*)$, and minimal in the sense that removing any single concept from $\mathcal{O}^\ell$ may alter the prediction.

2.3 Visual gradient backpropagation

In this section, we focus on the backpropagation of a specific concept’s activation into the input space. For this purpose, we use Integrated Gradients [27] and DeepLIFT [24].

2.3.1 Integrated Gradients

For this method, we need to compare our input x to a baseline x' . In our image setup, the baseline x' simply is a uniformly black image. Let $\gamma(\sigma)$ be a linear path between x' and x such that

$$\gamma(\sigma) = x' + \sigma \times (x - x'), \quad \sigma \in [0, 1]. \quad (17)$$

Given a function $F : \mathcal{X} \rightarrow [0, 1]$, Integrated Gradients IG along the “ i -th” dimension are defined as the path integral of the gradient along the linear path from the baseline to the target

$$\text{IG}_i(x) = (x_i - x'_i) \times \int_{\sigma=0}^1 \frac{\partial F(\gamma(\sigma))}{\partial x_i} d\sigma. \quad (18)$$

Intuitively, rather than relying on the gradient at a single point, which may be uninformative or saturated, IG accumulates gradient information along the entire path from the baseline to the input, yielding a more robust and complete picture of each feature’s contribution. To empirically obtain the IG, an approximation can be computed on points occurring at sufficiently small intervals

$$\widehat{\text{IG}}_i(x) = (x_i - x'_i) \times \frac{1}{M} \sum_{m=1}^M \frac{\partial F(\gamma(\frac{m}{M}))}{\partial x_i}, \quad (19)$$

where $m \in [50, 300]$. In practice, m interpolation steps are sufficient to obtain a reliable approximation, with $[50, 300]$ being the standard range recommended in the original work. Given the linearity of our last layer L , the IG of a subset of concepts $\mathcal{O}^{L-1}$ at layer $L - 1$ can be expressed as

$$\text{IG}_i^{\text{OPTIMUS}}(x) = (x_i - x'_i) \times \frac{1}{M} \sum_{m=1}^M \sum_{k \in \mathcal{O}^{L-1}} w_{\hat{y},k}^L \times \frac{\partial z_k^{L-1}(\gamma(\frac{m}{M}))}{\partial x_i}, \quad (20)$$

where $w_{\hat{y},k}^L$ represent the weights of the last linear layer, predicting class $\hat{y}$. Restricting the sum to concepts $k \in \mathcal{O}^{L-1}$ rather than the full set of neurons effectively filters out irrelevant activations, ensuring that only the minimal and sufficient concepts contribute to the final heatmap.



Figure 2: DeepLIFT-OPTIMUS explanations.

2.3.2 DeepLIFT

While Integrated Gradients is computationally expensive due to its integration over $[0, 1]$ and the repeated backpropagation of gradients, DeepLIFT proposes a single-point estimate instead. We first define $\Delta z_i^\ell = z_i^\ell(x) - z_i^\ell(x')$ as the activation difference with respect to the baseline x' introduced earlier. A multiplier $m_{\Delta z_i^\ell \Delta z_j^{\ell+1}}$ is then introduced as a finite-difference approximation of the partial derivative $\frac{\partial z_j^{\ell+1}}{\partial z_i^\ell}$ (when the baseline x' is close to x), defined as follows

$$m_{\Delta z_i^\ell \Delta z_j^{\ell+1}} = \frac{\Delta z_j^{\ell+1}}{\Delta z_i^\ell}. \quad (21)$$

A chain rule can then be applied to propagate the multipliers across the network

$$m_{\Delta z_i^\ell \Delta z_j^{\ell+2}} = \sum_t^{r^{\ell+1}} m_{\Delta z_i^\ell \Delta z_t^{\ell+1}} \times m_{\Delta z_t^{\ell+1} \Delta z_j^{\ell+2}}. \quad (22)$$

The final attribution score is then obtained by backpropagating these multipliers through the entire network

$$\text{DeepLIFT}_i = (x_i - x'_i) \times m_{\Delta z_i^0 \Delta z_j^L}. \quad (23)$$

Once again, exploiting the linearity of our last layer L , the DeepLIFT scores restricted to a subset of concepts $\mathcal{O}^{L-1}$ at layer $L-1$ can be expressed as

$$\text{DeepLIFT}_i^{\text{OPTIMUS}} = (x_i - x'_i) \times \sum_{k \in \mathcal{O}^{L-1}} w_{\hat{y},k}^L m_{\Delta z_i^0 \Delta z_k^{L-1}}. \quad (24)$$

In this section, we showed that both DeepLIFT-OPTIMUS and IG-OPTIMUS can be expressed as a weighted sum over the previous layer's heatmap

$$\text{DeepLIFT}_i^{\text{OPTIMUS}} = \sum_{k \in \mathcal{O}^{L-1}} w_{\hat{y},k}^L \times \text{DeepLIFT}_k^{L-1}, \quad (25)$$

and

$$\text{IG}_i^{\text{OPTIMUS}} = \sum_{k \in \mathcal{O}^{L-1}} w_{\hat{y},k}^L \times \widehat{\text{IG}}_k^{L-1}. \quad (26)$$

Here, the superscript indicates the target layer and the subscript identifies the target neuron with respect to which the attribution is computed. This formulation means that our implementation can directly leverage existing libraries to back-propagate concept-level attributions into the image space. The following section outlines the OPTIMUS-Prime framework applied to a practical visual classification task.

3 Experiments

For our experiments, we consider a Cat-Dog-Bird classification task [17] with a 70/15/15 train/val/test split. The encoder architecture is described below. The following sections report results for DeepLIFT-OPTIMUS. An identical analysis for IG-OPTIMUS is provided in the Appendix. Code is available at the link given on the first page of the article.



Figure 3: On the left: DeepLIFT-OPTIMUS. On the right: the difference between full concepts highlights and OPTIMUS (positive difference in mauve), highlighting unnecessary concepts.

3.1 Experimental setup

The encoder maps $x \in \mathbb{R}^{3 \times 224 \times 224}$ to $z \in \mathbb{R}^{512 \times 14 \times 14}$ through four Conv-BN-ReLU-MaxPool blocks (channels: 32, 64, 128, 256), followed by two Conv-BN blocks (512 channels, stride 1) with a final Sigmoid activation. All convolutions use 3×3 kernels with padding 1. During training, the backbone is followed by adaptive average pooling, a dropout layer ($p = 0.3$), and two fully-connected layers (512→256 with ReLU, then 256→3) for 3-class classification. At test time, the backbone is frozen and replaced by a single linear probe (512 → 3) preceded by adaptive average pooling. The model is optimized with cross-entropy loss and Adam ($\text{lr} = 10^{-3}$), with a batch size of 64, trained for 20 epochs on the full network and 10 additional epochs for the linear probe.

3.2 DeepLIFT-OPTIMUS

Once the model is trained, a PI can be computed (cf. Section 2.2) on the final linear layer, whose inputs are the last backbone features bounded in $[0, 1]$ by the Sigmoid activation. As shown in Section 2.3, DeepLIFT-OPTIMUS can be expressed as a weighted sum of DeepLIFT attributions over the preceding layer. We therefore use the neuron-wise DeepLIFT implementation from Captum [13].

Figure 2 presents our OPTIMUS explanations. The original image is shown on the left, while the heatmap is on the right, ranging from transparent purple to yellow, highlighting the sufficient and minimal concepts required to sustain the model’s prediction.

Figure 3 illustrates the difference between the full neuron-layer activation and the OPTIMUS explanation, highlighting unnecessary concepts. Depending on the concepts captured by each neuron, non-OPTIMUS concepts may partially overlap with the selected ones. The left panel shows the OPTIMUS explanation as described above; the right panel displays the difference between all concept activations and the OPTIMUS subset, highlighted in mauve. Finally, Figure 4 provides additional visualizations of unnecessary concepts, as defined above.

3.3 Analytic evaluation of OPTIMUS explanations

To complement the qualitative visualizations, we evaluate the computation of OPTIMUS explanations in terms of PI size and runtime. We compare the two proposed greedy rules, LOWGAIN and MAXMIN, against a UNORDERED baseline, which removes features according to their original index order and does not use any margin-based criterion. Detailed results are reported in Appendix D.

Overall, LOWGAIN provides the best trade-off between compactness (length) and efficiency: it typically yields PI sizes close to MAXMIN, while being substantially faster in high-dimensional settings. This makes it a practical heuristic for repeated PI computations, including enumeration-oriented analyses. The UNORDERED baseline is often fastest but produces larger PIs, whereas MAXMIN remains a useful compactness-oriented reference for optimization-based searches. The artificial-data results further characterize how PI size and runtime scale with the number of variables and classes, providing a benchmark for future comparisons.

4 Discussion

OPTIMUS-Prime remains partially *model-specific*, as a few assumptions must hold. The first constraint is the boundedness of the concept space at the targeted layer. As discussed in Section 2.2, this requirement can be relaxed at the cost of trading formal guarantees for PAC statistical ones. The second, non-removable assumption is the linearity of the targeted layer. While final layers are commonly linear in practice, this prevents OPTIMUS-Prime from being applicable to arbitrary *black-box*



Figure 4: Difference between DeepLIFT and DeepLIFT-OPTIMUS: unnecessary concepts.

models. Additional minor constraints arise from the use of DeepLIFT [24] and Integrated Gradients [27], whose applicability similarly depends on the underlying architecture. Furthermore, recent work has raised concerns about the reliability of gradient-based attribution methods, highlighting sensitivity to baseline choice and input perturbations [1], limitations that our formal PI framework partially addresses by grounding explanations in provable sufficiency rather than gradient sensitivity.

A recurring observation from our experiments is that multiple neurons can encode the same concept [15], a phenomenon partly attributable to the simplicity of the classification tasks considered. While this redundancy is harmless in practice, one may nonetheless prefer disentangled concept representations. This can typically be addressed at inference time through a variety of existing techniques [14, 7, 12], which we leave as a natural extension of the present framework.

Finally, we reflect on the semantic meaning of latent concepts, which can sometimes remain too abstract to be directly interpretable by humans. This is precisely why we advocate for input-space heatmaps as the primary medium for communicating explanations. Prior work has shown that neurons can be inherently interpretable across a range of architectures [10, 2, 3]. For instance, word embedding concepts appear to encode semantic directions [20], while superposition phenomena have been documented in more recent settings [7, 4]. We deliberately leave these challenges to the practitioner, in keeping with our goal of designing a framework that is as *model-agnostic* as possible, and that can be naturally composed with existing interpretability tools to address architecture-specific constraints.

5 Conclusion

In this paper, we present OPTIMUS-Prime, a new framework for deep visual explanations introducing OPTIMUS explanations. These explanations are grounded in theoretical derivations that provide strong guarantees while maintaining high interpretability: the highlighted concepts are both sufficient and minimal to sustain the model’s prediction.

Although OPTIMUS-Prime is broadly and almost transparently applicable to many deep architectures, the framework remains subject to a few assumptions. Notably, the targeted layer must be linear, and the preceding layer’s output must be bounded to preserve our theoretical guarantees. In practice, the semantic quality of the explanations further depends on the information captured within the targeted neurons, which may be entangled or superposed. We therefore leave users free to design alternative architectures that best suit their needs. A natural and promising extension of this work would be the application of OPTIMUS-Prime to recent transformer-based architectures, which we leave for future work.

References

- [1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, pages 9525–9536, Red Hook, NY, USA, December 2018. Curran Associates Inc.
- [2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. *2017 IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, July 2017.
- [3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, December 2020.
 - [4] Brian Cheung, Alexander Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
 - [5] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020.
 - [6] Omid Davoodi, Shayan Mohammadizadehsamakosh, and Majid Komeili. On the interpretability of part-prototype based classifiers: a human centric analysis. *Scientific Reports*, 13(1):23088, December 2023.
 - [7] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy Models of Superposition, September 2022.
 - [8] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and Joao Marques-Silva. From contrastive to abductive explanations and back again. In *International Conference of the Italian Association for Artificial Intelligence*, 2020.
 - [9] Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva. On Explaining Decision Trees, October 2020. *arXiv:2010.11034* [cs].
 - [10] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and Understanding Recurrent Networks, November 2015. *arXiv:1506.02078* [cs].
 - [11] Been Kim, M. Wattenberg, J. Gilmer, Carrie J. Cai, James Wexler, F. Viégas, and R. Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, November 2017.
 - [12] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML’20*, pages 5338–5348. JMLR.org, July 2020.
 - [13] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqu Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, September 2020. *arXiv:2009.07896* [cs].
 - [14] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited, July 2019. *arXiv:1905.00414* [cs].
 - [15] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent Learning: Do different neural networks learn the same representations? In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, pages 196–212. PMLR, December 2015.
 - [16] S. Lundberg and S-I. Lee. A unified approach to interpreting model predictions. In *Proc. of NIPS’17*, 2017.
 - [17] MahmoudNoor. High Resolution Cat-Dog-Bird Image Dataset (13000).
 - [18] Joao Marques-Silva. Logic-based explainability in machine learning, 2023.
 - [19] Joao Marques-Silva, Thomas Gerspacher, Martin Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive bayes and other linear classifiers with polynomial time and delay. *Advances in Neural Information Processing Systems*, 33:20590–20600, 2020.

- [20] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.
- [21] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *SIGKDD*, pages 1135–1144, 2016.
- [22] 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. *International Journal of Computer Vision*, 128(2):336–359, February 2020. arXiv:1610.02391 [cs].
- [23] A. Shih, A. Choi, and A. Darwiche. A symbolic approach to explaining bayesian network classifiers. In *Proc. of IJCAI'18*, pages 5103–5111, 2018.
- [24] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, pages 3145–3153, Sydney, NSW, Australia, August 2017. JMLR.org.
- [25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, April 2014. arXiv:1312.6034 [cs].
- [26] Jules Soria, Zakaria Chihani, Julien Girard-Satabin, Alban Grastien, Romain Xu-Darme, and Daniela Cancila. Formal Abductive Latent Explanations for Prototype-Based Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(30):25590–25598, March 2026.
- [27] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, pages 3319–3328, Sydney, NSW, Australia, August 2017. JMLR.org.
- [28] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8689, pages 818–833. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computer Science.
- [29] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis, February 2017. arXiv:1702.04595 [cs].

A Notations, Acronyms and Datasets

Notations/Acronyms	Description
$\mathcal{X}$	Input space $\mathcal{X} \subseteq \mathbb{R}^P$
P	Input dimension
$\mathcal{Y}$	Label space $\{1, \dots, K\}$
(x_n, y_n)	i.i.d. draw from the joint distribution $P_{X,Y}$
$\mathcal{D}$	Dataset $\{(x_n, y_n)\}_{n=1}^N$
N	Number of training instances
h	Neural network classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$
L	Number of layers in the network
K	Number of classes
$\hat{y}(x)$	Predicted class for input x
$z^\ell(x)$	Output of the ℓ -th hidden layer, $z^\ell \in \mathbb{R}^{r^\ell}$
r^ℓ	Number of neurons at layer ℓ
$\circ^\ell$	A concept index in $\{1, \dots, r^\ell\}$ at layer ℓ
$\mathcal{O}^\ell$	OPTIMUS prime implicant (set of concepts) at layer ℓ
m	OPTIMUS explanation heatmap
f	Linear multiclass classifier function $f : \mathcal{Z} \rightarrow \mathbb{R}^K$
x^*	A given instance with predicted class c
I	Set of all feature indices $\{1, \dots, r\}$
S	Candidate subset of features $S \subseteq I$
$\Gamma_k(S)$	Worst-case dominance margin for class k given fixed features S
$g_q^{(k)}$	Margin gain of feature q w.r.t. rival class k
x'	Baseline input (uniformly black image)
$\gamma(\sigma)$	Linear path between baseline x' and input x
$\text{IG}_i(x)$	Integrated Gradients attribution at pixel i
$\widehat{\text{IG}}_i(x)$	Empirical approximation of $\text{IG}_i(x)$ with M steps
M	Number of interpolation steps for IG (typically in $[50, 300]$)
$\text{IG}_i^{\text{OPTIMUS}}(x)$	OPTIMUS-restricted Integrated Gradients at pixel i
DeepLIFT_i	DeepLIFT attribution score at pixel i
$\text{DeepLIFT}_i^{\text{OPTIMUS}}$	OPTIMUS-restricted DeepLIFT attribution at pixel i
$w_{\hat{y},k}^L$	Weight of the last linear layer for predicted class $\hat{y}$ at neuron k

Table 1: Notations and Acronyms

B Proof for Proposition 1

Proposition 1. *Assume that x^* is classified as class c . Starting from $S = I$, by repeatedly removing any feature $q \in S$ such that $\forall k \neq c \quad \Gamma_k(S \setminus \{q\}) > 0$, the resulting set S is a multiclass PI when no further removal can guarantee the constraint.*

Proof. Since $\Gamma_k(I) > 0$ for all $k \neq c$, the initial set I is a multiclass implicant. At each step, a feature is removed only if the remaining set still satisfies $\Gamma_k(S) > 0$, $\forall k \neq c$. Thus, the implicant property is preserved throughout the procedure. At termination, no feature $q \in S$ can be removed while preserving all dominance constraints. Hence, for every $q \in S$, there exists some rival class $k \neq c$ such that $\Gamma_k(S \setminus \{q\}) \leq 0$. Therefore, no subset obtained by deleting one more feature remains an implicant, which means that S is subset-minimal, and hence a multiclass PI. $\square$

Table 2: Analytic comparison of multiclass PI search on real data and image embeddings.

Dataset	Classes	Variables	Rules	PI size	Time (ms)
Iris	3	4	MAXMIN	3.333 $\pm$ 0.516	0.046
Iris	3	4	LOWGAIN	3.333 $\pm$ 0.516	0.055
Iris	3	4	UNORDERED	3.333 $\pm$ 0.516	0.042
Wine	3	13	MAXMIN	8.315 $\pm$ 1.537	0.220
Wine	3	13	LOWGAIN	8.315 $\pm$ 1.537	0.094
Wine	3	13	UNORDERED	9.796 $\pm$ 1.192	0.081
Digits	10	64	MAXMIN	29.656 $\pm$ 4.337	4.864
Digits	10	64	LOWGAIN	30.144 $\pm$ 4.350	0.371
Digits	10	64	UNORDERED	37.619 $\pm$ 4.258	0.290
Image embeddings	3	512	MAXMIN	353.814 $\pm$ 41.086	330.997
Image embeddings	3	512	LOWGAIN	371.986 $\pm$ 43.110	6.203
Image embeddings	3	512	UNORDERED	434.436 $\pm$ 27.962	4.972

C Algorithm for finding a multiclass PI

Algorithm 1: Finding one multiclass PI.

Input: Feature set I , feature dimension r , predicted class c , and greedy rule $\mathcal{G}$.

Output: A multiclass PI $S \subseteq I$.

Initialize $S \leftarrow I$

Sort the features in ascending order according to $\mathcal{G}$: $(q_{(1)}, \dots, q_{(r)}) \leftarrow \text{Order}_{\mathcal{G}}(I)$

for $i = 1, \dots, r$ **do**

if $q_{(i)} \in E$ **and** $\Gamma_k(S \setminus \{q_{(i)}\}) > 0, \forall k \neq c$ **then**
 $S \leftarrow S \setminus \{q_{(i)}\}$

return S

D Additional analytic evaluation

Tables 2 report the average size and computation time of the PIs obtained by each greedy rule. Lower PI size indicates a more compact explanation, while lower runtime indicates a more efficient search procedure. Bold values indicate the shortest PI size or the quickest computation time within each dataset. The results show that the same trend persists in this regime: MAXMIN gives the smallest PIs, UNORDERED is the fastest, and LOWGAIN achieves an intermediate trade-off between PI length and computation time.

For the image embeddings, the number of variables is much larger, as each instance is represented by a 512-dimensional feature vector extracted from the image model. This setting therefore provides a higher-dimensional evaluation scenario, closer to the latent representations used in deep vision models.

Figures 5–7 report PI size and computation time as functions of the number of variables on artificially generated data. This setting allows us to vary both the feature dimension and the number of classes. For each number of classes, the left plot shows the average PI size with standard deviation as a shaded region, while the right plot shows the average computation time.

As expected, PI size and runtime generally increase with the number of variables. The UNORDERED rule is consistently faster, but usually returns larger PIs. In contrast, MAXMIN and LOWGAIN produce PI sizes that remain relatively close, while MAXMIN becomes substantially more expensive as the feature dimension grows.

We also observe that MAXMIN tends to run faster when the number of classes increases, for a fixed number of variables. A possible explanation is that additional classes impose tighter dominance constraints, leaving fewer removable features and making the choice of q more efficient, although this typically leads to larger PIs. The raw numerical results are reported in Table 3.

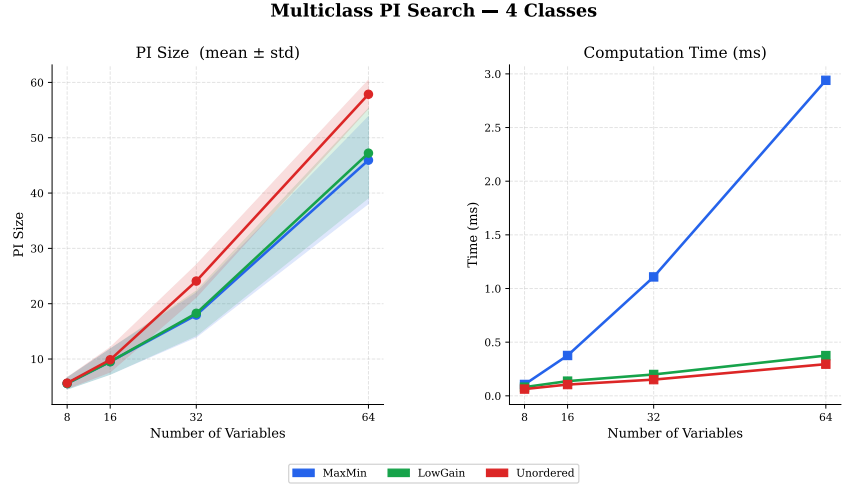


Figure 5: Artificial-data comparison of 4-class PI search vs. number of variables.

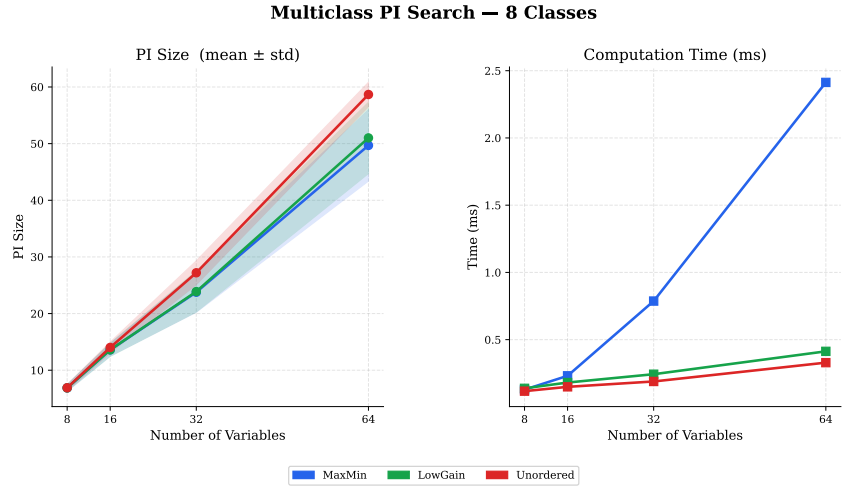


Figure 6: Artificial-data comparison of 8-class PI search vs. number of variables.

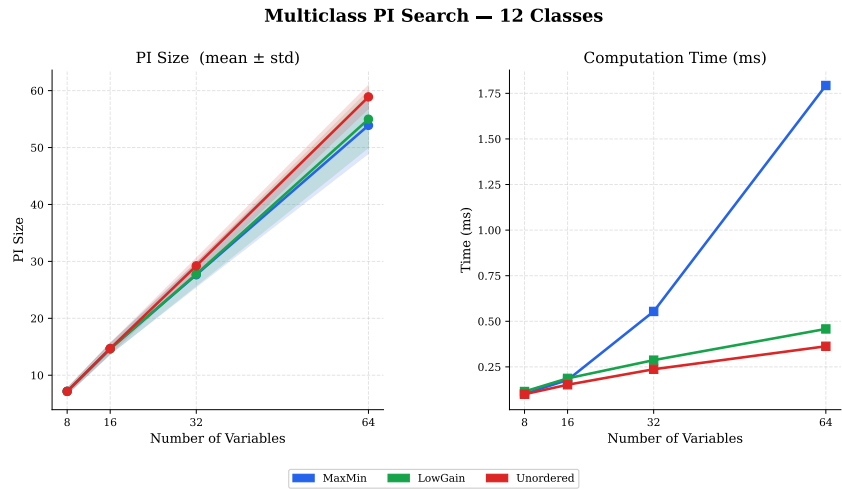


Figure 7: Artificial-data comparison of 12-class PI search vs. number of variables.

Table 3: Artificial-data comparison of multiclass PI search as a function of the number of classes and variables.

Classes	Variables	Rules	PI size	Time (ms)
4	8	MAXMIN	5.575 ± 0.997	0.106
4	8	LOWGAIN	5.567 ± 0.998	0.081
4	8	UNORDERED	5.650 ± 0.972	0.063
4	16	MAXMIN	9.525 ± 2.254	0.376
4	16	LOWGAIN	9.517 ± 2.258	0.137
4	16	UNORDERED	9.867 ± 2.183	0.105
4	32	MAXMIN	17.950 ± 3.951	1.108
4	32	LOWGAIN	18.267 ± 4.010	0.199
4	32	UNORDERED	24.092 ± 2.890	0.151
4	64	MAXMIN	45.950 ± 7.822	2.940
4	64	LOWGAIN	47.217 ± 8.039	0.375
4	64	UNORDERED	57.858 ± 2.521	0.295
8	8	MAXMIN	6.875 ± 0.571	0.128
8	8	LOWGAIN	6.875 ± 0.571	0.138
8	8	UNORDERED	6.900 ± 0.554	0.117
8	16	MAXMIN	13.592 ± 1.107	0.231
8	16	LOWGAIN	13.600 ± 1.150	0.181
8	16	UNORDERED	14.017 ± 0.931	0.149
8	32	MAXMIN	23.767 ± 3.528	0.787
8	32	LOWGAIN	23.883 ± 3.601	0.243
8	32	UNORDERED	27.200 ± 2.096	0.189
8	64	MAXMIN	49.683 ± 6.284	2.413
8	64	LOWGAIN	51.008 ± 6.275	0.413
8	64	UNORDERED	58.675 ± 2.134	0.329
12	8	MAXMIN	7.167 ± 0.506	0.102
12	8	LOWGAIN	7.167 ± 0.506	0.115
12	8	UNORDERED	7.167 ± 0.506	0.100
12	16	MAXMIN	14.650 ± 0.900	0.178
12	16	LOWGAIN	14.633 ± 0.921	0.187
12	16	UNORDERED	14.692 ± 0.874	0.152
12	32	MAXMIN	27.650 ± 2.104	0.554
12	32	LOWGAIN	27.758 ± 1.983	0.287
12	32	UNORDERED	29.217 ± 1.311	0.237
12	64	MAXMIN	53.892 ± 4.905	1.793
12	64	LOWGAIN	54.983 ± 4.956	0.458
12	64	UNORDERED	58.900 ± 2.002	0.363

E Experimental results with Integrated Gradients



Figure 8: IG-OPTIMUS explanations.



Figure 9: Difference between IG and IG-OPTIMUS: unnecessary concepts.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims of the paper regards theoretical guarantees in the concept space. This is formally explicated in Section 2.

Guidelines:

- The answer [\[N/A\]](#) means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [\[No\]](#) or [\[N/A\]](#) answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limits of the proposed framework are identified and extensively discussed in the Discussion section.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The full set of assumptions is extensively discussed in Section 2 and in the Discussion section. Proofs are either referenced through citations if already existing, or provided in the paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Complete transparency is given with regard to the experimental results. A link to our anonymized code is also available in the article. However, we provide an already-trained model in order to enhance reproducibility on light hardware. Reviewers are free to retrain the model with the provided architecture.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Open access to the data and code is provided directly in the paper as well as through a link to our repository.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: All training details are provided. We used standard architectures, well understood and easily reproducible.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: Standard deviations are provided with the plotted curves. However, we do not provide statistical significance tests.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not include experiments.
- The authors should answer [\[Yes\]](#) if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Our experiments rely on well documented architectures. In order to enhance reproductibility, we made our shared code CPU-compatible for light architectures thanks to pre-training.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We believe this work respect the provided Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Our contribution does not deviate or innovate from existing XAI literature regarding positive or negative societal impacts. However, this is slightly discussed in the introduction.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Authors and creators are properly cited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: New assets (i.e. OPTIMUS explanations) are well documents, a code implementation is also provided.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.