Learning Encoding-Decoding Direction Pairs to Unveil Concepts of Influence in Deep Vision Networks

Alexandros Doumanoglou *aldoum@iti.gr*
Department of Advanced Computing Sciences (DACS), University of Maastricht (UM)
Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH)

Kurt Driessens *kurt.driessens@maastrichtuniversity.nl*
Department of Advanced Computing Sciences (DACS), University of Maastricht (UM)

Dimitrios Zarpalas *zarpalas@iti.gr*
Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH)

Abstract

Empirical evidence shows that deep vision networks often represent concepts as directions in latent space with concept information written along directional components in the vector representation of the input. However, the mechanism to encode (write) and decode (read) concept information to and from vector representations is not directly accessible as it constitutes a latent mechanism that naturally emerges from the training process of the network. Recovering this mechanism unlocks significant potential to open the black-box nature of deep networks, enabling understanding, debugging, and improving deep learning models.

In this work, we propose an unsupervised method to recover this mechanism. For each concept, we explain that under the hypothesis of linear concept representations, this mechanism can be implemented with the help of two directions: the first facilitating encoding of concept information and the second facilitating decoding. Compared to previous matrix decomposition, autoencoder, and dictionary learning approaches which rely on the reconstruction of feature activations, we propose a different perspective to learn these encoding-decoding direction pairs. We base identifying the decoding directions on directional clustering of feature activations and introduce signal vectors to estimate encoding directions under a probabilistic perspective. Unlike most other works, we also take advantage of the network's instructions encoded in its weights to guide our direction search. For this, we illustrate that a novel technique called *Uncertainty Region Alignment* can exploit these instructions to reveal the encoding-decoding mechanism of interpretable concepts that influence the network's predictions.

Our thorough and multifaceted analysis shows that, in controlled, toy settings with synthetic data, our approach can recover the ground-truth encoding-decoding direction pairs. In real-world settings, our method effectively reveals the encoding-decoding mechanism of interpretable concepts, often scoring substantially better in interpretability metrics than other unsupervised baselines, such as PCA and NMF. Finally, we provide concrete applications of how the learned directions can help open the black box and understand global model behavior, explain individual sample predictions in terms of local, spatially-aware, concept contributions and intervene on the network's prediction strategy to provide either counterfactual explanations or correct erroneous model behavior.

Figure 1: Concept Encoding - Decoding under the Linear Representation Hypothesis: Deep networks encode high-level concepts, such as sky or boat, in distinct directions of their latent space, respectively s_1 and s_2. The illustration shows the encoding of two concept latent factors (i.e., the degree of concept presence) ^{1}; ^{2} within a patch's representation x_p by utilizing the concept embedding directions s. Additionally, it demonstrates how a lter w $_1$ can be employed to extract one of these latent factors from the representation. The illustration omits depicting the latent space bias for brevity. We use the terms {encoding direction, concept embedding, signal direction} and the terms { lter, decoding direction} interchangeably throughout this article.

1 Introduction

The linear representation hypothesis suggests that deep neural networks encode high-level concepts in directions of their latent space (Bereska & Gavves, 2024). This hypothesis is su ciently supported by empirical evidence, mostly by the e ectiveness of linear probing. In the latter, a single linear layer can be trained on top of feature representations originating from the upper part of deep neural networks to solve semantic tasks with great success (Szegedy et al., 2014; Alain & Bengio, 2018; Zhou et al., 2018; Kim et al., 2018; Elhage et al., 2022; Nanda et al., 2023). In the concept encoding mechanism suggested by the linear representation hypothesis, the concept content of an input patch is written to its embedding as a linear combination of concept embeddings: directions in the latent space represented as vectors, each one associated with a concept. The corresponding scalar coe cient in the linear combination constitutes the concept's latent factor that encodes the degree of concept presence in the input. Due to its utility in encoding the latent factor to a vector representation, we also refer to the concept embedding as the concept's encoding or signal direction. Retrieving the latent factor back can be accomplished by taking the inner product between the patch embedding and a lter, another vector which we term a decoding direction. (Fig. 1) An encoding-decoding direction pair is interpretable whenever it is related to a concept that is aligned with human intuition; and is in uential whenever it is related to a concept that the network consistently uses to make predictions (Kim et al., 2018; Pfau et al., 2020).

The encoding - decoding mechanism of concepts for a pre-trained network is not directly accessible. Instead, it is a latent mechanism emerged from network training and needs to be inferred by any means of reverse engineering. Recovering this mechanism is of signi cant importance, as accessing it enables understanding representations (Zhou et al., 2018; Kim et al., 2018), assessing in uence of concepts on the network's predictions (Kim et al., 2018; Pfau et al., 2020; Fel et al., 2023b; Pahde et al., 2024), debugging neural networks (Anders et al., 2020), and correcting erroneous model behavior (Anders et al., 2022; Pahde et al., 2023; Neuhaus et al., 2023; Dreyer et al., 2024a).

Previous works that identify concept directions rely either on concept annotations (Zhou et al., 2018; Kim et al., 2018; Pahde et al., 2024) or on a decompose-then-reconstruct regime of feature activations (Zhang et al., 2021; Graziani et al., 2023b; Bricken et al., 2023; Lim et al., 2024; Cunningham et al., 2024; Bussmann et al., 2025). The former su er from the need for concept speculation and the access to annotations while the latter, face the di culty of overcoming unstructured noise that may be present in feature activations in the form of distractors (Haufe et al., 2014; Kindermans et al., 2017) or dark matter (Engels et al., 2025).

Unlike previous approaches, our method relies on neither annotations nor the reconstruction of activations, thereby sidestepping these earlier limitations. Instead, we identify decoding directions via directional clustering of feature activations and estimate encoding directions from a probabilistic perspective. Moreover, in contrast to most prior work, we guide our search for directions by exploiting the network's instructions, i.e. the prediction strategy baked into its weights through a process we call Uncertainty

Figure 2: Our Encoding-Decoding Direction Pairs (EDDP) powers a range of applications, highlighting both the generality and the precision of our approach. The gure summarizes applications that we selected to discuss in this work.

Region Alignment. We conduct a thorough and multifaceted analysis that highlights the e ectiveness of the approach in revealing the encoding-decoding mechanism of (a) the ground-truth concepts in controlled and synthetic toy settings, and (b) interpretable monosemantic concepts in real-world studies of state-of-the-art deep vision architectures. We also demonstrate the utility of our learned directions in obtaining global, local, and counter-factual explanations, as well as identifying and correcting awed model behavior in a toy setting (Fig. 2).

Our comparative analysis primarily considers established matrix decomposition baselines, speci cally Principal Component Analysis (PCA) and Non-Negative Matrix Factorization (NMF). We further extend the comparisons in Appendix Section A.16, where we evaluate our approach against the methods on which we build, namely Unsupervised Interpretable Basis Extraction (UIBE) (Doumanoglou et al., 2023) and Concept Basis Extraction (CBE) (Doumanoglou et al., 2024). Throughout, we focus on settings in which the concept space does not exceed the dimensionality of the embedding space, and we leave a systematic comparison with Sparse AutoEncoders (SAEs) (Sharkey et al., 2022) to future work.

2 Related Work

2.1 Direction Learning

We categorize related work into supervised and unsupervised direction learning approaches. In each category, we go through the previous methods, describing the limitations, di erences, and similarities with our approach.

Supervised Concept Direction Learning

Concept Activation Vectors (CAVs) (Kim et al., 2018; Zhou et al., 2018) use supervised linear classi ers trained on annotated concept datasets to separate representations of samples with a target concept from those without. To solve the task, the resulting classi er weights (lters) must extract (decode) the concept factors from these representations; thus, they approximate concept decoding directions, although they can be inexact due to distractor-noise in the features (Haufe et al., 2014; Kindermans et al., 2017; Pahde et al., 2024). To estimate concept encoding directions, PCAVs (Pahde et al., 2024) take the di erence of means between representations of samples with and without the concept. These methods require both labeled data and prior speci cation of the concept name. In contrast to these probing approaches, the proposed method

is unsupervised, revealing concepts by analyzing the structure of the latent space and thereby avoiding both annotation costs and upfront concept speci cation.

Unsupervised Concept Direction Learning Existing unsupervised direction-learning methods that rely on matrix decomposition (PCA (Graziani et al., 2023a), SVD (Graziani et al., 2023b), NMF (Zhang et al., 2021; Fel et al., 2023b)) can discover concept directions without annotations, but they have key limitations. PCA and SVD are constrained by orthogonality and miss concepts that do not in uence variance (Fel et al., 2023a). NMF, while non-negative, lacks latent space bias and, thus, expressivity, and it has no straightforward decoding directions. To alleviate these shortcomings, at least to some extent, Graziani et al. (2023a;b) combined direction learning with gradient re-ranking, and both Zhang et al. (2021) and Graziani et al. (2023a;b) worked on subgroups of features belonging to the same class category, establishing a class-conditioned protocol for direction learning. In contrast, the proposed method uses a class-agnostic protocol without the need for sample class labels. Additionally, our approach considers non-orthogonal concept directions and accounts for feature space bias, overcoming the limitations of previous methods.

Dictionary Learning (Bricken et al., 2023; Yun et al., 2023) and Sparse Autoencoders (SAEs) (Sharkey et al., 2022; Cunningham et al., 2024; Bussmann et al., 2025) identify decoding-encoding directions driven by the reconstruction of representations after decomposing them into sparse latent factors. They typically consider a latent space larger than the embedding space and may su er from the di culty of overcoming noise in the representations in the form of dark matter (Engels et al., 2025). In contrast, our method is independent of feature reconstruction, unrestricted by the relative sizes between the concept and the embedding space, and unlike these prior works, it implicitly links the learned directions to how the model actually utilizes them. However, in this work, we restrict our experiments to cases where the concept space is no larger than the embedding space and leave experimental comparisons to SAEs with the opposite assumption for future work. Further discussion regarding SAEs is provided in Section A.23.

Moving away from reconstructing representations, the methods of Doumanoglou et al. (2023; 2024) perform directional clustering of feature activations, a process that often unveils the decoding mechanism of interpretable, monosemantic concepts. We build on these previous approaches to estimate decoding directions, removing orthogonality and feature-standardization constraints and introducing new loss terms that preserve or improve concept interpretability while reducing the impact of distractor noise in the lter weights. Compared to the previous attempt (Doumanoglou et al., 2024) of linking directions to their utilization by the network, our Uncertainty Region Alignment approach achieves up to 22.56% relative improvement in interpretability metrics in 3 out of 4 cases. We also address the estimation of concept encoding directions, which these prior works did not address. Further details on this comparison are provided in Section A.3.

2.2 Applications of Directions

Our learned direction pairs enable a series of applications, from global model understanding and detailed spatially-aware local explanations down to model intervention, which allows for counterfactual explanations and model correction. Typical previous enablers of such applications are the Concept Bottleneck Models (CBMs) (Koh et al., 2020). Classic CBMs consist of a backbone feature extractor, a concept bottleneck predicting concept content from features, and a linear head mapping concepts to classes. What makes this architecture interpretable is the fact that the concept vector, i.e, the output of the concept bottleneck, is trained to have each dimension aligned with an interpretable concept. Recent post-hoc CBM variants (Yuksekgonul et al., 2023; Oikarinen et al., 2023) retro t this structure onto existing black-box models by freezing the original backbone, discarding its head, inserting a concept bottleneck projection into concept space, and learning a task-speci c sparse linear head over concepts. To learn concepts without annotations, these works exploit CLIP Radford et al. (2021), and further advances Benou & Raviv (2025) use visual prompting (Shtedritski et al., 2023) to make the bottleneck spatially aware, allowing localization of concepts at the patch level rather than only at the image level. Most recently (Rao et al., 2024), SAEs (Cunningham et al., 2024) have been used to build CBMs directly for CLIP.

To access the advantages of CBMs, these previous approaches require training the last linear head using the model's training data and, sometimes, additional concept-annotated datasets or external models like CLIP. In contrast, the proposed work is non-intrusive: it doesn't need to train new model components and does

not require concept labels. It shows that one can capture many bene ts of concept bottleneck models by identifying directions corresponding to concepts the model already encodes and then using these directions in applications of mechanistic interpretability (Saphra & Wiegre e, 2024; Kästner & Crook, 2024; Bereska & Gavves, 2024).

3 Background

In our formulation, which follows the linear representation hypothesis (Elhage et al., 2022; Nanda et al., 2023), the latent factor of a concept is a scalar denoting the concept's presence in the input, and is embedded in the latent space via multiplication with the concept's encoding direction, also called the concept's embedding or signal direction (Kindermans et al., 2017). For this reason, we also refer to this latent factor as the signal value. Feature activations are considered as linear combinations of signals and noisy directional components called distractors. In the proposed approach, a lter, through the inner product with an input representation, can extract the signal's value and for this reason we term it a decoding direction. Below we provide a more formal explanation of these terms and provide details essential to understand our contributions.

3.1 Preliminaries

Let $X \in \mathbb{R}^{H \times W \times D}$ denote the feature representation of an image in an intermediate layer of a deep neural network with spatial dimensions $H, W \in \mathbb{N}^+$ and latent space dimensionality $D \in \mathbb{N}^+$. Let also $x_p \in \mathbb{R}^D$ denote a pixel element of this representation at the spatial location $p = (w, h)$, $w \in \{0, 1, \dots, W-1\}$, $h \in \{0, 1, \dots, H-1\}$. Since x_p is related to a patch within the input image of the network, we also refer to it as patch embedding.

3.2 Signals, Distractors, Filters, Concept-Detectors

In encoding a single concept i, Kindermans et al. (2017); Pahde et al. (2024) proposed a model for the data generation process of feature representations: $x_p = \alpha_p s_i + \beta_p d$; $s_i, d \in \mathbb{R}^D$; $\alpha_p, \beta_p \in \mathbb{R}$. Here, s_i is the signal direction that carries the information of whether x_p is part of concept i. The concept information lies within the signal value α_p. Larger α_p suggests greater con dence that x_p belongs to concept i. d is the distractor direction, modeling noise, or information not related to the concept. β_p follows a random distribution, typically the gaussian or uniform distribution, and is independent of whether x_p belongs to concept i. According to Kindermans et al. (2017), the value of the signal α_p can be extracted using a lter w_i and the inner product: $z_{p,i} = w_i^T x_p = \alpha_p w_i^T s_i + \beta_p w_i^T d$, if we choose $w_i : w_i \perp d$, and $w_i^T s_i = 1$. Let σ denote the sigmoid function. Since stronger values of α_p indicate more con dence in concept presence, when combined with a threshold $b_i \in \mathbb{R}$ that can be learned from data, this lter can be turned into a concept detector: $y_{p,i} = \sigma(z_{p,i} - b_i)$, essentially a binary classi er that can answer the question of whether p belongs to concept i.

With access to the signal value, Haufe et al. (2014); Kindermans et al. (2017) o er a formula to estimate the concept's signal direction:

$$\hat{s}_i = \frac{\mathrm{cov}[x_p, \alpha_p]}{\mathrm{var}[\alpha_p]} = \frac{\mathrm{cov}[x_p, z_{p,i}]}{\mathrm{var}[z_{p,i}]} \tag{1}$$

3.3 Unsupervised Interpretable Direction Learning

Recent research (Doumanoglou et al., 2023) introduced an unsupervised method to identify concepts from the structure of the latent space. Motivated by the directional encoding of concepts, the method partitions the latent space into linear regions, each represented by a hyperplane and a normal vector, forming clusters. Feature activations from an unlabeled concept dataset, possibly activations of images from the network's training set, are assigned to these clusters. The method learns W and b of a feature-to-cluster membership function, a mapping to the semantic space, with $y_p = \sigma(W^T x_p - b) \in [0, 1]^I$; $W \in \mathbb{R}^{D \times I}$; $b \in \mathbb{R}^I$, and I as the cluster count. By softly assigning features to a small number of clusters, the interpretability of the

Figure 3: The core concept of Unsupervised Interpretable Basis Extraction (UIBE Doumanoglou et al. (2023)) is to learn a set of concept detectors, which are essentially binary linear classifiers with learnable filters and biases. These detectors aim to transform feature representations to the soft-binary vector space of concepts in which the newly transformed representations are sparse. In this procedure the input to the method consists of image representations coming from an unlabeled concept dataset. Identifying the concept name behind each detector is done in a post-processing step with a procedure we refer to as Direction Labeling.

clustering is improved. This is grounded in the idea that an image patch generally holds only a few semantic labels from a larger set, reflecting sparsity in the semantic space. Sparsity in the assignments is achieved using two loss terms: the first is Sparsity Loss (L^s), and the second is Maximum Activation Loss (L^{ma}), which ensures binary cluster membership:

$$\begin{aligned} L^s &= E_p\ L_p^s\ ; \quad L^{ma} = E_p\ q_p^T \log_2(y_p)\ ; \\ L_p^s &= H(q_p); \quad q_p = \frac{y_p}{\|y_p\|_1} \end{aligned} \tag{2}$$

with H denoting entropy. Under a different interpretation perspective, a column of W together with a corresponding element of b (i.e., w_i, b_i) forms a linear classifier or concept detector $y_{p,i} = (w_i^T x_p - b_i)$, which means that w_i is a filter and thus acting as a decoding direction. This method also optimizes linear separability by minimizing the inverse of the classification margin $M_i = \frac{1}{\|w_i\|_2}$ (Maximum Margin Loss - L^{mm}) and penalizes clusters with few assignments using the Inactive Classifier Loss - L^{ic} (Doumanoglou et al., 2024) (See Fig. 3 and more details in Section A.1).

3.4 Direction Labeling

When learning directions in an unsupervised manner, the name of the concept represented by each encoding-decoding direction pair is unknown. In a real-world setting, the concept name could be identified by manually inspecting the samples that (maximally) activate the decoding direction. However, in a benchmark setting, where interpretability needs to be quantified in terms of metrics and without human involvement, it is necessary to automate the assignment of a concept label to each of the direction pairs. We refer to this process as direction labeling and employ Network Dissection Bau et al. (2017) for its implementation. In essence, Network Dissection assigns a semantic label to each of the concept detectors based on their segmentation performance in a dataset with annotated concepts. Although it was originally proposed as a method to assign labels to each of the basis vectors in the natural latent space basis, Network Dissection is capable of assigning labels to any set of directions after basis change. Despite possible biases against unsupervised learning due to annotation limitations, we adopt and expand on this labeling protocol as a best-effort approach to evaluate the interpretability of our concept detectors and the rest of the unsupervised baselines.

3.5 Concept Sensitivity Testing

Given the intermediate representation of an image belonging to class k and the direction of concept c in the latent space, RCAV (Pfau et al., 2020) measures the sensitivity of the model to concept c when predicting class k. This is accomplished by perturbing the representation towards the direction of the concept with strength , and subsequently comparing the output probability of the network for the same class before and after the perturbation. Subsequently, an overall dataset score in the range [1; 1] is computed, where zero means inconsistent use of the concept by the model, while extremes indicate consistent positive or negative concept contributions to predicting class k. A statistical test compares concept sensitivity against sensitivity towards random directions to ensure signi cance. In our evaluations, we refer to directions of signi cant in uence in cases where the directions meet the criteria of this statistical signi cance test. Although the initial work used CAVs to test concept sensitivity, a recent study (Pahde et al., 2024) suggests that the appropriate direction for this purpose is the concept's encoding direction, an aspect that we consider in our evaluations.

4 Proposed Method

Figure 4: The proposed method analyzes the latent space to uncover its directional structure. Because many concepts are naturally encoded as speci c directions, this process often reveals the encoding-decoding mechanism of meaningful, monosemantic, and highly interpretable concepts. The gure depicts an overview of the method's components. L denotes loss terms. Purple indicates contributions of this work, while light gray indicates loss terms from Doumanoglou et al. (2023; 2024).

Method Overview

The aim of our unsupervised approach is to reveal the encoding-decoding mechanism of meaningful concepts that in uence the network's predictions. Our method learns encoding-decoding direction pairs that explain the structure of a deep vision network's latent space and takes as input feature representations from an unlabeled concept dataset. The latent space, from which feature representations come, corresponds to a network layer at a given depth. We jointly learn the decoding directions (concept detectors) fW ; bg and the encoding directions (signal vectors) $\hat{S}$ in an end-to-end process.

Our formulation is built upon several key contributions:

1. Flexible and Sparser Clustering: We lift previous constraints on the concept detectors (such as orthogonality between lters and feature space standardization) in Doumanoglou et al. (2023), enabling more exible and less redundant (i.e., sparser) clustering. To preserve or even improve interpretability, we incorporate two novel loss terms: Focal Sparsity Loss and Excessively Active Classi er Loss (Section 4.2).

Figure 5: Left: The learnable parameters of the method $\hat{S}; W; b$ and intermediate variables $z_p; y_p; q_p$. Top Right: Feature manipulation and Uncertainty Region Alignment. Bottom Right: Loss terms L with their dependencies. Purple indicates loss contributions of this work, while light gray indicates loss terms from Doumanoglou et al. (2023; 2024).

2. Encoding Direction Estimation: We incorporate an extended multi-concept signal-distractor model (Section 4.1) and de ne signal vectors ($\hat{S}$) as estimators for the concept encoding directions based on a probabilistic framework (Section 4.3).
3. Network Instruction Driven Search: We introduce Uncertainty Region Alignment (L_{ur}) (Section 4.4), a novel loss that can improve concept interpretability and in uence by aligning concept detector ambiguity with model prediction uncertainty. For the latter, the method makes use of the upper part of the network, after the layer of study, that produces output class probabilities, denoted as f^+.
4. E ective Optimization: The entire set of learning parameters is optimized jointly using an - constrained optimization scheme based on the Augmented Lagrangian Loss (Section 4.5), which allows direct control of clustering quality with easier to tune, interpretable hyper-parameters.

Our method overview and component interconnections are shown in Figures 4 and 5. The used loss terms are described in Table 1, and the learning algorithm is outlined in Section A.2

4.1 Multi-Concept Signal-Distractor Data Model

We introduce an extended signal-distractor data model for the latent space, which models the encoding of multiple concepts. Each patch embedding x_p is considered as a linear combination of latent concept signals $S \in \mathbb{R}^{D \times I}$ and distractors $D \in \mathbb{R}^{D \times F}$. We also consider a latent space bias $c \in \mathbb{R}$, common for all x_p.

$$x_p = S\ _p + D\ _p + c \tag{3}$$

with $_p \in \mathbb{R}^I$ and $_p \in \mathbb{R}^F$. S is a matrix of $I \in \mathbb{N}^+$, D-dimensional, unit-norm concept signal directions and D a matrix denoting a basis for distractor components. Each signal direction encodes the presence of a distinct concept. We apply the same assumptions for individual signal values $_{p;i}$ (the i-th element of $_p$) and distractor coe cients $_{p;f}$ as in Section 3.2.

4.2 Losses to Preserve or Improve Interpretability

We propose Self-Weighted Reduction (R_{SW}) as an aggregation method to estimate the maximum element in a set. Consider the set of elements $\{f_k\}$, $\in \mathbb{R}^+; k \in \mathbb{N}$. The Self-Weighted Reduction is de ned as:

$$R_{SW}(f_k) = \frac{\sum_k {}_k {}_k^{+1}}{\sum_k {}_k} \tag{4}$$

which is equal to the weighted average of elements in $f_k g$ with each element being weighted by $_k$, $1;\ 2\ R^{+}$ a sharpening factor. This aggregation may be seen as a soft di erentiable version of the max operation, since the largest value in the set $f_k g$ is weighted with the largest weight.

Excessively Active Classi er Loss (L^{eac}) This loss penalizes excessively large clusters to prevent trivial solutions in which all inputs are assigned to a single cluster. It relies on a hyper-parameter $2 (0; 1)$, similar to sparse autoencoders Ng et al. (2011), which sets a proportional bound on cluster size. With $_{p;i} 2 [0; 1]$ denoting the membership of patch p for cluster i (Section 3.3), the unreduced loss formula for the i-th cluster is below, with $> 1;\ 2\ R^{+}$ as a sharpening factor, 1 normalizing the loss in the range $[0; 1]$, and p varying across all pixels and image representations in the concept dataset:

$$L_i^{eac} = \frac{1}{1} \text{ReLU}(E_p[y_{p;i}]\ ) \tag{5}$$

The nal loss uses R_{SW} : $L^{eac} = R_{SW}(fL_i^{eac}; i 2 f0; 1; :::; I\ 1gg)$

Focal Sparsity Loss (L^{fs}) Inspired by Focal Loss (Lin et al. (2017)), we introduce Focal Sparsity Loss, which places more emphasis on sparsifying the feature-to-cluster assignments of the most challenging patch embeddings. If L_p^s (2) denotes the Sparsity Loss for pixel p, the Focal Sparsity Loss is de ned as:

$$L^{fs} = \frac{\sum_p {}_p L_p^s}{\sum_p {}_p} \tag{6}$$

i.e. based on a weighted scheme de ned by coe cients $_p 2 [0; 1]$. We choose $_p$ to be proportional to the patch's number of cluster assignments. In detail:

$${}_p = 1\ \ (R_{SW}(fq_{p;i}; i 2 f0; 1; :::; I\ 1gg))\ \ ;\ \ 2\ R^{+} \tag{7}$$

with $$ a sharpening factor and $q_p = \frac{y_p}{\|y_p\|_1}$, calculated as in (2). $R_{SW} fq_{p;i} g 2 [0; 1]$ is approximately equal to $1=k$, with k denoting the patch's number of cluster assignments. Similar to the previous case, in (6) p varies across all spatial elements and image representations in the concept dataset. When we use Focal Sparsity Loss we use it as a replacement for the Sparsity Loss L^s of Doumanoglou et al. (2023).

4.3 Signal Vectors as Concept Signal Estimators

In this paragraph, we consider the new data model that was outlined in Section 4.1 and try to validate whether (1) can be used to estimate the signal direction of a concept. Suppose that our objective is to estimate the signal direction of concept i when we have access to a collection of patch embeddings $_p fx g$ and their signal values $f_{p;i} g$. Starting from the approach of Kindermans et al. (2017), it is easy to prove that whenever $a_{p;i}$ is independent of all $a_{p;j}\ ; j 6= i$ and $_{p;f}$ the following property holds: $\text{cov}[x_p\ \ a_{p;i} s_i; a_{p;i}] = 0$, which, according to Kindermans et al. (2017) directly implies that (1) can be utilized to estimate the direction of the signal.

However, while $a_{p;i}$ can be considered independent of distractor coe cients $_{p;f}$ that represent noise, the independence assumption between $_p a$ and $a_{p;j}\ ; j 6= i$ is easily violated in practice. Some pairs of variables $a_{p;i}$ and $a_{p;j}\ ; j 6= i$ may indeed be independent and this would be the case when concepts i and j belong to di erent groups of mutually-exclusive concepts, for example when concept i belongs to the group of objects and concept j to the group of colors and the concepts of the rst group are independent of those of the second. However, when the concepts i and j belong to the same group of mutually exclusive concepts (for instance, when concept i corresponds to car and concept j to tree and both concepts belong to the group of objects), there may be an anti-correlated relationship between the variables $_p a$ and $a_{p;i}\ ; j 6= i$ due to the fact that whenever $a_{p;i} > b_i$, $a_{p;j} < b_j$. However, the latter bias can be eliminated if we consider only samples with the concept instead of both positive and negative samples. In that case, among that subset of the data, the signal values $a_{p;i}$ and $a_{p;j}$ can be considered independent by assumption, as we now removed the biases $b_i; b_j$ due to sub-sampling. This allows us to still consider (1) as a signal estimator, even under the extended data model of multiple concepts, provided that in the computation of the covariance and variance terms of (1) we subsample the patch embeddings based on their concept label, i.e. keep only samples with

Figure 6: The uncertainty region of the network is de ned as the subspace where all network's predictions are maximally uncertain. The uncertainty region of the concept detectors is de ned as the intersection of all their decision hyperplanes, i.e. the subspace of ambiguous concept predictions. Aligning these two through feature manipulation may improve direction pair interpretability or serve as a balance between the interpretability and in uence of the learned direction pairs.

the concept, instead of additionally considering samples without it. The latter can be easily accomplished when employing the respective concept detector.

We refer to the signal estimator of concept i that is obtained under these conditions as the signal vector $\hat{s}_i$. However, we still require access to the signal values. As explained in Section 3.2, estimating signal values can be attributed to the lters of the concept detectors. They can serve this purpose if the weight vector w_i is orthogonal to all s_j where $j \neq i$, as well as the distractor subspace D.

Thus, we employ the following Filter-Signal Vector Orthogonality Loss when learning the directions:

$$L^{fso} = \sum_{E_{i,j}} \frac{}{((1 - \delta_{i,j}) w_i^T s_j)^2} \tag{8}$$

with $\delta_{i,j}$ the kronecker delta and w; s denoting the L2-normalized lter weights and signal vectors.

To achieve accurate signal value extraction, w_i should additionally be orthogonal to the distractor basis; however, we do not explicitly estimate the distractors. Instead, we use the Uncertainty Region Alignment loss from Section 4.4 to ensure alignment of the directions with utilization by the network.

4.4 Uncertainty Region Alignment to Discover Meaningful Concepts of In uence

The presence or absence of a concept in a representation can provide neutral, supportive, or opposing evidence against the prediction of a class. Since the concept-class pair association is unknown when learning concept directions, a straightforward strategy to perform concept arithmetic on the features in order to nd their utility by the network lacks ground-truth information on how concepts a ect class predictions. To overcome this di culty, we can make a simple but more elegant hypothesis that uncertain network predictions occur when the representation has ambiguous concept information. We propose improving the direction search by aligning the uncertainty regions of the network and the concept detectors. The uncertainty region of the network is the subspace where its predictions are most uncertain, and the uncertainty region of the concept detectors is the subspace where their decision hyperplanes intersect. Figure 6 illustrates the concept of Uncertainty Region Alignment.

To accomplish the alignment, all the patch embeddings x_p in an image are manipulated towards the direction dx_p to arrive at $x'_p = x_p - dx_p$. Based on our estimates of w_i, b_i, and $\hat{s}_i$, we select the direction dx_p so that the shifted x'_p lies at the intersection of the concept detectors' decision hyperplanes, i.e. $w_i^T(x_p - dx_p) - b_i = 0; \forall i$. Then, we ensure the network's prediction for the resulting manipulated image representation is highly

Table 1: Encoding-Decoding Direction Pairs Loss Descriptions. Purple indicates loss contributions of this work, while light gray indicates loss terms from Doumanoglou et al. (2023; 2024)

Symbol	Loss Name	Loss Description
L^{fs}	Focal Sparsity	Ensure sparse patch-to-cluster assignments, emphasizing challenging cases
L^{ma}	Maximum Activation	Push towards binary patch-to-cluster memberships
L^{ic}	Inactive Classi er	Ensure all clusters have a su cient number of members
L^{mm}	Maximum Margin	Ensure cluster members are linearly separated from non-cluster members by a su cient margin
L^{eac}	Excessively Active Classi er	Penalize large clusters to avoid cases such as assigning all patches to a single cluster
L^{fso}	Filter - Signal Orthogonality	More accurate signal value extraction, enabling a more precise estimation of encoding directions
L^{ur} (L^{uur} or L^{cur})	Uncertainty Region Alignment	Discover interpretable concepts of in uence exploiting network instructions.

uncertain, e ectively aligning both uncertainty regions. More speci cally, we de ne two types of feature manipulation for this purpose:

Unconstrained Feature Manipulation (UFM) in which x^0_p corresponds to the closest point of x_p in the uncertainty region of the concept detectors, i.e.: $x^0_p = g_p(x_p; W) = x_p - dx_p$ and dx_p given by:

$$dx_p = (W^T)^+ (W^T x_p - b) \tag{9}$$

with A^+ denoting the pseudo-inverse of A and a detailed derivation provided in Section A.4.

Constrained Feature Manipulation (CFM) in which we additionally restrict feature manipulation to occur within the span of the signal vectors, i.e. $dx_p = \hat{S} v_p$, $v_p \in R^I$, with $\hat{S} \in R^{D \times I}$ denoting the matrix whose columns correspond to the learned signal vectors $\hat{s}_i$; $i \in \{0, 1, \ldots, I-1\}$. The feature manipulation formula for a patch embedding x_p is $h_p(x_p; W; \hat{S}) = x_p - dx_p$, with v_p and dx_p given by:

$$v_p = (W^T \hat{S})^+ (W^T x_p - b)) \tag{10}$$

$$dx_p = \hat{S} v_p = \hat{S}(W^T \hat{S})^+ (W^T x_p - b) \tag{11}$$

and again, a detailed derivation is provided in Section A.4. These manipulations are carried out for all p in the image representation X simultaneously, leading to a manipulated image representation $X^0 = g(X; W)$ or $X^0 = h(X; W; \hat{S})$, based on the manipulation type (Fig. 5).

Finally, the Uncertainty Region Alignment Loss (L^{ur}) is:

$$L^{ur} = E_{X^0} H(f^+(X^0)) \tag{12}$$

with H denoting entropy. When we manipulate the representations via either UFM or CFM, we use the layer's activation function (typically a ReLU) to keep the features in the input domain of the next layer. Throughout the experiments, we will denote the Uncertainty Region Alignment loss with UFM as L^{uur} and with CFM as L^{cur}. We consider the use of L^{uur} and L^{cur} to be mutually-exclusive. We either use the rst or the second type of manipulation when computing the Uncertainty Region Alignment Loss and never both types at the same time.

4.5 Augmented Lagrangian Loss for E ective Direction Learning

The loss terms of the proposed method often con ict with each other. For instance, increasing con dence in the predictions of concept detectors (L^{ma}) con icts with maximizing the separation margin (L^{mm}), and a

large margin may also con ict with sparsity losses (L^s or L^{fs}). Simply using a weighted sum of these losses requires careful architecture and dataset speci c tuning of loss weights. The latter can be laborious and only indirectly controls clustering quality, which is mostly determined by the nal loss values. To avoid this, we adopt an -constrained optimization approach and reformulate our objective as a minimization problem with inequality constraints rather than linearly combining all loss terms. In this formulation, hyper-parameter tuning primarily pertains to target loss values, providing direct control over the clustering quality. We use the Augmented Lagrangian formulation (Hestenes, 1969; Bertsekas, 2014) to solve the following optimization problem:

$$\min \quad {}^{fs}L^{fs} + {}^{ur}L^{ur} \quad \text{s:t:} \quad \begin{cases} L^{ma} & {}^{ma} \\ L^{ic} & {}^{ic} \\ L^{mm} & {}^{mm} \\ L^{eac} & {}^{eac} \\ L^{fso} & {}^{fso} \end{cases} \tag{13}$$

with ^{fs} ; ^{ur} $2 R^{+}$ the only loss weight coe cients. In this formulation, we assign a target value to each individual loss term participating in the inequalities and, as shown in the experiments, subsequently optimize for interpretability (L^{fs} ; L^{uur}), or a balance between interpretability L^{fs} and in uence L^{cur}.

5 Encoding-Decoding Direction Pairs in Applications

Moving forward to the application level, in this Section, we discuss how to read and intervene on concept information (Section 5.1) and how to capitalize on the learned direction pairs to obtain local, detailed, and spatially-aware concept explanations (Section 5.2).

5.1 Reading Concept Information and Intervening on their Encoding

Reading Concept Information from Patch Embeddings: Reading concept information from a patch embedding is equivalent to estimating the signal value encoded in the representation. Filter directions can extract this value; however, this is done with a constant o set derived from the latent space bias c. This o set can be neutralized by calculating the di erence between two projected embeddings (in the direction of the lter), which directly equals the di erence in their signal values. Thus, reading concept information is helpful to be done with respect to the concept information in a point of reference. Two points of reference that we nd useful are:

- The Average Embedding: Estimating the signal value relative to a collection's mean, for instance a collection of samples with or without the concept.
- Point of Uncertainty: A point on the concept detector's decision hyperplane that represents maximum concept ambiguity. Centering around this point ensures that positive values indicate concept presence and negative values indicate concept absence.

Intervening on Concept Encoding: Intervention modi es an embedding's concept content by overwriting its signal value with a target value derived from a reference embedding. To encode the presence or absence of a concept in a patch embedding, we select target values based on:

- The Average Signal Value in representative positive or negative concept samples.
- Top or Bottom Quantiles of Signal Values in a collection of positive or negative concept samples.

Concrete implementation details for reading and intervening on concept information are provided in Section A.5.

Figure 7: Description of the synthetic dataset that we use in the experiments: a) Concept set: {0,1,2} b) Image Classes: {a,b,c} with each class corresponding to a unique pair of concepts. c) Each image is comprised of two patch representations $x_{p_1}; x_{p_2}$.

5.2 Using Encoding-Decoding Direction Pairs and the Regions of Uncertainty to Provide Concept-Based Local Explanations and Detailed Spatially-Aware Concept Contribution Maps

In this Section, we provide details on how the learned Encoding-Decoding Direction Pairs can be leveraged to provide concept-based local explanations for a network's prediction. The core contribution is the introduction of an explanation logit ($l_e = l_c - l_b^m$), which is the difference between the network's class logit for the input image (l_c) and a class logit corresponding to a baseline artificial representation derived from the uncertainty region of the model (l_b^m). We then decompose this l_e into local, concept-specific components.

In Section A.6, we show that, for networks with a Global Average Pooling (GAP) penultimate layer, the explanation logit can be decomposed into four key components: the sample concept contributions, which are input dependent, the baseline concept contributions from an uncertain prediction, a correction factor compensating for imperfect learning convergence, and a final, unexplained residual.

Grouping the first two components in the previous decomposition and exploiting the linearity of GAP, the explanation logit enables the derivation of Concept Contribution Maps (CCMs), which provide a spatially-aware breakdown of the explanation logit in terms of its constituent concept contributions. CCMs differ from previous approaches (Zhou et al., 2018; Graziani et al., 2023a) in that they do not simply localize the concepts in the input image, but they quantify the contributions of each concept across patches to the explanation logit, including both positive and negative contributions.

The contribution of a specific concept i from an image patch p to the explanation logit is calculated as the product of two factors:

- A Global Contribution Factor, referred to as Concept-Class Relation Coefficient (CCRC) ($w_c^T s_i$), independent of the individual image input, with w_c denoting the network's class vector.
- A Local Contribution Factor ($\hat{v}_{p;i} - \hat{v}_{p;i}^b$): A local, sample dependent, spatially-aware factor representing the difference in concept information between the actual image patch ($\hat{v}_{p;i}$) and the baseline patch ($\hat{v}_{p;i}^b$).

The CCM is a heatmap that visualizes the multiplication of these two factors across the image, precisely showing which regions and concepts positively or negatively drove the final classification decision. Further details are provided in Section A.6.

6 Experiment on Synthetic Data

In this Section, we test our method on synthetic data which follow the data generation process detailed in Section 4.1. We validate three aspects: a) the effectiveness of signal vectors in estimating the concept's encoding direction given ground-truth signal values, b) the efficacy of our unsupervised approach to reliably identify the ground-truth concept encoding-decoding direction pairs, under challenging conditions

Table 2: Evaluating the performance of the concept detectors in classifying patch embeddings in the experiment on synthetic data. The metric is Intersection over Union (IoU). Rows correspond to concept detectors and columns to ground-truth concept classes. Clearly, each detector is aligned with one distinct ground-truth concept.

		Concept		
		#0	#1	#2
Detector	#0	0	1.0	0
	#1	1.0	0	0
	#2	0	0	1.0

for conventional techniques, c) the necessity of Filter-Signal Orthogonality loss L^{so}_f for reliable estimation of encoding directions.

We consider synthetic image representations with two spatial elements $p_1; p_2$, i.e. $W = 2$ and $H = 1$. Every pixel p is presumed to be associated with just one concept from a set of $I = 3$ concepts. Let $c(p) \in \{0; 1; 2\}$ represent the concept label of p, and $k \in \{a; b; c\}$ denote an image class. We construct image representations as follows: for $k = a$, $c(p_1) = 0$ and $c(p_2) = 1$; for $k = b$, $c(p_1) = 0$ and $c(p_2) = 2$; and for $k = c$, $c(p_1) = 1$ and $c(p_2) = 2$ (Fig. 7). We set the embedding space dimensionality to $D = 8$, the size of the distractor basis to $F = 2$ and randomly create unit-norm vectors to construct the matrices S and D. Using those principles and hyper-parameters we generate patch embeddings $x_{p_1}; x_{p_2}$ for each image according to (3). The latent signal values and distractor coefficients follow the uniform distribution: $\alpha_{p;i} \sim U(0:0; 2:25)$ if p is not part of concept i and $\alpha_{p;i} \sim U(2:75; 5:0)$, otherwise, while $\alpha_{p;f} \sim U(0; 5:0)$ is independent of the patch's concept label. We introduce a bias of $c = 10$ across all dimensions of the representations to maintain them in the positive quartile, similar to the impact of a ReLU layer. We generate a balanced dataset with each class being represented by 1000 images.

Using this dataset, we train a network to predict the image class based on the concept content of its patches. The network we use is composed of just two layers (corresponding to the top part of a potentially larger deep network). The first is an average-pooling layer, and the second is a linear layer with $K = 3$ output classes. After training, the network attains 96.33% accuracy on a test set, randomly generated based on the previous principles. More details regarding the setup of this experiment are provided in Section A.8.

6.1 Evaluation of the Signal-Vector Estimator

Based on the synthetic pixel dataset $\{x_{p_1}\} \cup \{x_{p_2}\}$ and the ground-truth signal values $\alpha_{p;i}$, we put the estimator of Kindermans et al. (2017) and our signal vectors under test. The difference between the two estimators is the sub-sampling procedure that we proposed in Section 4.3. Given the ground-truth matrix of signal directions S we are able to calculate cosine similarities between the estimated signal directions and the respective ground-truth. The results of this experiment are depicted in Fig. 8 (in blue and green). Since we use the ground-truth signal values, this experiment evaluates the efficacy of signal vectors under perfect input conditions. The figure showcases the effectiveness of our proposed sub-sampling procedure when using (1), as in all cases, signal vectors demonstrate perfect alignment with the ground-truth directions, while in the absence of sub-sampling, the estimation is less reliable. Conclusively, the sub-sampling technique that we proposed in Section 4.3 adapts more precisely to our extended data model.

6.2 Evaluation of the Encoding-Decoding Direction Pairs

In this experiment, signal vectors are jointly learned with the concept detectors, employing every proposed facet of the method presented in Section 4. Concept detectors are evaluated for their ability to detect each one of the ground-truth concepts, while signal vectors are evaluated for their alignment with the ground-truth concept encoding directions.

Decoding Directions (Concept Detectors): Using the synthetic pixel dataset $\{x_{p_1}\} \cup \{x_{p_2}\}$, we assess the ability of each learned concept detector to detect each of the ground-truth concepts. Since the method is unsupervised, the name of the concept that each detector identifies is unknown. For this reason, we need

Figure 8: 1) Cosine similarity between the ground-truth concept encoding directions and their estimation via Kindermans et al (1) calculated without our proposed feature sub-sampling. In this experiment, the estimation of signal directions uses ground-truth signal values. 2) Cosine similarity between the ground-truth concept encoding directions and their estimation via Signal-Vectors, i.e. using (1) calculated with our proposed feature sub-sampling. Similarly to the previous experiment this one uses ground-truth signal values. Under ideal input conditions, Signal-Vectors are proven to be faithful signal estimators. 3) Cosine similarity between ground-truth concept encoding directions and their estimation via our proposed Encoding-Decoding Direction Pairs (EDDP). In constrast to the previous two experiments, in this one Signal Vectors are learned in an unsupervised manner using all the aspects of the proposed method. Without access to ground-truth signal values, our approach is capable of recovering the ground-truth concept encoding directions. 4) Cosine similarity between the ground-truth concept encoding directions and their estimation via our proposed Encoding-Decoding Direction Pairs (EDDP) but without using the Filter-Signal Orthogonality Loss L^{fso}. In that case, the encoding direction estimation is inferior compared to (3).

to implement direction labeling (Section 3.4). In Table 2, we present the Intersection over Union scores for each detector against actual concept classes. Zero values indicate complete purity and no mixing of the concepts, while scores of 1.0 indicate perfect concept detection. As an additional step, we also examine how well the learned lters extract signal values from representations. Since distractors and the latent space bias c are not directly estimated by our method, we can only quantify the extracted signal values as deviations from the dataset's average value (See also Section 5.1, (29) and Section A.9). The Root Mean Squared Error (RMSE) between these extracted values and the ground truth, after subtracting the mean signal value, is noted as 0:06. We also veri ed that in this experiment, where the distractor components are independent of the concept content contained in the features, the learned lters were orthogonal to all the vectors in the distractor basis D, without explicitly enforcing this to occur.

Encoding Directions: Fig. 8 - orange indicates perfect alignment between the learned signal vectors and the ground-truth concept encoding directions. We stress the fact that, in this experiment, signal vectors are driven by signal values extracted by the concept detectors, and this is di erent from Section 6.1, where signal vectors were tested under ideal input conditions.

6.3 Ablation Study

We also consider a case study where we learn the direction pairs by omitting L^{fso} from the objective function, i.e., learning the direction pairs without (/wo) it. In that case, the concept detectors still discovered the ground-truth directional clusters, as in Table 2. However, the decoding directions were less reliable in extracting the ground-truth signal values. The RMSE between the extracted values and the ground truth, after subtracting the mean signal value, was found to be 0:31, which is substantially inferior to the standard case discussed above. The latter has an impact on the delity of the signal vectors. Fig. 8 purple visualizes the cosine similarity between the learned signal vectors and the ground-truth encoding directions for this case. The experiment shows that the signal direction estimation is less e ective compared to when L^{fso} is taken into account.

6.4 Evaluation against other Baselines

The ground-truth encoding directions of this example (Table 15 in Section A.8) contain both positive and negative components, and the relationship among the encoding directions (and the distractors) is, in general, not orthogonal. In theory, and without the need for practical experiments, this example cannot be addressed by NMF, K-Means, or PCA. NMF would produce a signal basis of non-negative components. Similarly, the cluster centers of K-Means would point toward the positive quartile where the centroids reside. Finally, the non-orthogonal nature of the ground truth encoding directions implies that the PCA's solution space is insu cient.

7 Experiments on Deep Image Classi ers

In this Section, we apply our method in real-world scenarios. In the following sections, we a) assess the faithfulness of the learned encoding-decoding direction pairs, b) provide qualitative segmentation results obtained by utilizing our learned concept detectors, c) compare with unsupervised baselines in interpretability and in uence terms, d) conduct an ablation study to quantify the contribution of each proposed component, e) compare with supervised direction learning, aiming to provide more insights on the interpretability and in uence of the directions, f) provide application use cases in which the learned direction pairs assist in providing global, local, and counterfactual model explanations, g) provide an example application in model correction.

7.1 Direction Learning and Evaluation Protocol

We apply our Encoding-Decoding Direction Pairs (EDDP) on the last convolution layer of ve di erent networks: ResNet18 (He et al., 2016) trained on Places365 (Zhou et al., 2017), ResNet50 (He et al., 2016) trained on Moments in Time (MiT) (Monfort et al., 2019), as well as E cientNet (b0) (Tan & Le, 2019), Inception-v3 (Szegedy et al., 2016), and VGG16 (Simonyan & Zisserman, 2014) trained on ImageNet (Deng et al., 2009). For all the networks that we study, when learning the direction pairs, we use the Broden (Bau et al., 2017) concept dataset, except for ResNet50, for which we use Broden-Action Ramakrishnan et al. (2019). The concept datasets feature dense pixel annotations; Broden includes 1197 concepts across 63K images in 5 concept categories (object, part, material, texture, color), while Broden-Action incorporates an additional action category with 210 labels and 23K more images. We emphasize that when learning the direction pairs, we do not make use of the annotations that complement the concept datasets.

For baselines, we consider directional clusterings based on either PCA, NMF, or the Natural latent space basis (referred to as Natural). For our EDDP, we examine two variants that are di erentiated based on the choice of feature manipulation strategy used in Uncertainty Region Alignment. We use EDDP-U to refer to UFM and EDDP-C to CFM. Unless stated otherwise, our Encoding-Decoding Direction Pairs use the Augmented Lagrangian Loss from Section 4.5, all losses from Sections 4.2 and 3.3 (except L^{st}), along with either L^{uur} (EDDP-U) or L^{cur} and L^{fso} (EDDP-C) from Sections 4.3 and 4.4.

After learning the directions with any unsupervised method, we use Network Dissection (Bau et al., 2017) and the annotations available in the concept datasets to label them. As described in Section 3.4, we make a basis change by projecting the network feature activations onto the learned decoding directions (for PCA and EDPP) or by directly utilizing the concept factors learned by NMF. More details on the experimental setup for all methods, including baselines, are provided in Section A.13, while in Section A.10, we discuss how our evaluation pipeline implicitly captures human judgment.

7.2 Faithfulness Assessment of the Encoding-Decoding Direction Pairs

In this Section, we aim to evaluate the faithfulness of the encoding-decoding direction pairs, i.e. to assess whether the signal value extracted by the decoding direction is indeed encoded in the direction of the signal vector. Although this was validated in synthetic data (Section 6), in this paragraph, we make the same evaluation in the case of real data, in the absence of ground truth. Based on activation maximization (Nguyen et al., 2019), we propose the following novel approach to address this evaluation.

Figure 9: Our approach to assess the faithfulness of the direction pair: Starting from an image with the concept, we use deep dream to maximize the concept's decoding direction and collect the patch embeddings from the dream images at the regions with the concept. Subsequently, we t a parametric line to those features in order to estimate the direction towards which they are moving when dreaming. We term this direction as the dreaming direction.

Figure 10: Dreaming pre-images from the process of estimating the dreaming directions. Left: Maximizing a concept detector learned for the latent space of ResNet18 and labeled with the concept name skyscraper. Right: Maximizing a concept detector learned for the latent space of E cientNet and labeled with the concept name tvmonitor. Both images regard directions learned with EDDP-C.

Suppose that we evaluate the faithfulness of a particular direction pair w, s and let b denote the respective concept detector's bias. We use the concept detector (w, b) to collect a set of images containing the concept. Subsequently, with these images as an initialization point, we use Deep Dream (Olah et al., 2017) to compute pre-images (Mahendran & Vedaldi, 2015; 2016), that is, images that maximize the decoding direction w at the spatial locations in which the concept was initially found. In this manner, we start from an image containing the concept and ask Deep Dream to amplify (in the input pixel space) whatever concept the detector can detect. We keep running the activation maximization loop for K iterations and keep a record of how the features (at the locations with the concept) evolve while optimizing. Finally, we use all the recorded features collected during the dreaming optimization to t a parametric line and estimate the direction that best describes feature evolvement while dreaming. We call the direction of this line the dreaming direction (Fig. 9). We conclude by comparing the dreaming direction of the concept for each image with the respective signal vector in terms of cosine similarity.

In Figures 11 and 12, we provide histograms of these cosine similarities when applying our method to ResNet18 and E cientNet. In all cases, even those without considering L^{fso}, approximately 90% of the signal vectors have cosine similarity with the dreaming directions above 0.7, indicating su cient faithfulness. The e ect of L^{fso} is to push the distribution to become more Gaussian. To validate the e ectiveness of the sub-sampling strategy proposed in Section 4.3, we also provide histograms of cosine similarities between signal vectors and dreaming directions in the absence of sub-sampling. The results are depicted in Fig. 13 where, in that case, the deviation of the signal vectors from the dreaming direction is evident. Example dreaming pre-images for two cases are provided in Fig. 10 and further details and experiments may be found in Sections A.11 and A.12.

7.3 Interpretability Evaluation of Concept Detectors: Qualitative Segmentation Results and Interpretability Statistics

Figure 14 presents qualitative segmentation results obtained using our concept detectors, con rming that they are largely monosemantic. These visualizations rely on Network Dissection to generate segmentation maps and calculate the Intersection over Union (IoU) scores, which represent the detector's performance across the entire concept dataset's validation split. A detector quali es as interpretable if its IoU score

Figure 11: Cosine similarity histogram between dreaming directions and signal vectors. Left: Directions learned for ResNet18 trained on Places365 (I = 448). Right: Directions learned for E cientNet trained on ImageNet (I = 1120). These histograms regard EDDP-C, i.e. directions learned with L ^{cur} and L ^{fso} .

Figure 12: Cosine similarity histogram between dreaming directions and signal vectors. Left: Directions learned for ResNet18 trained on Places365 (I = 448). Right: Directions learned for E cientNet trained on ImageNet (I = 1120). These histograms regard EDDP-U, i.e. directions learned with L ^{uur} but without L ^{fso} .

surpasses the prede ned threshold of 0:04. We provide detailed quantitative interpretability statistics for each network architecture, exempli ed by Table 3 for ResNet18 and other Tables in Section A.17. More extensive qualitative comparisons and statistics against unsupervised baselines are provided in the same Section of the Appendix.

Table 3: Network Dissection statistics for ResNet18 trained on Places365. For each concept category in Broden, we report two numbers: First, the number of concept detectors that were labeled with the name of a concept belonging to the category and second, the number of unique concept labels from the category that have been assigned to the set of the concept detectors. The table summarizes statistics for unsupervised methods and for di erent values of the cluster count hyper-parameter I.

		ResNet18 / Places365						
I	Method	Color	Object	Part	Material	Scene	Texture	Total
384	PCA	0 / 0	22 / 13	5 / 2	1 / 1	28 / 16	117 / 32	173 / 64
	NMF	0 / 0	94 / 42	7 / 4	2 / 2	189 / 101	47 / 26	339 / 175
	EDDP-U	2 / 2	118 / 54	8 / 7	3 / 3	151 / 114	21 / 17	303 / 197
	EDDP-C	2 / 2	117 / 51	9 / 6	4 / 4	153 / 114	22 / 17	307 / 194
448	PCA	0 / 0	22 / 13	5 / 2	1 / 1	28 / 16	127 / 33	183 / 65
	NMF	0 / 0	95 / 46	9 / 7	1 / 1	234 / 120	33 / 20	372 / 194
	EDDP-U	2 / 2	137 / 62	6 / 6	5 / 5	193 / 129	21 / 18	364 / 222
	EDDP-C	2 / 2	136 / 60	5 / 5	5 / 5	194 / 133	22 / 18	364 / 223
512	Natural	0 / 0	107 / 43	11 / 8	1 / 1	269 / 135	26 / 17	414 / 204
	PCA	0 / 0	22 / 13	5 / 2	1 / 1	27 / 15	125 / 32	180 / 63
	EDDP-U	1 / 1	127 / 66	10 / 10	4 / 4	236 / 161	29 / 22	407 / 264
	EDDP-C	1 / 1	128 / 66	11 / 11	4 / 4	236 / 165	27 / 22	407 / 269

As an exemplary interpretation of the report, Table 3 shows that when learning direction pairs for ResNet18 with I = 512, Network Dissection reported that EDDP-C can identify 269 di erent concepts in the following

Figure 13: Cosine similarity histogram between dreaming directions and signal vectors learned without sub-sampling. Left: Directions learned for ResNet18 trained on Places365 (I = 448). Right: Directions learned for E cientNet trained on ImageNet (I = 1120). These histograms regard EDDP-U, i.e. directions learned with L^{uur} but without L^{fso}.

Figure 14: Qualitative segmentations using the concept detectors learned with our method as reported by Network Dissection. Rows correspond to concept detectors learned for di erent networks. For ResNet18 and E cientNet the results were obtained using CFM while for Inception-v3, VGG16 and Resnet50 using UFM.

categories: 66 objects, 165 scenes, 11 parts, 4 materials, 22 textures, and 1 color. The total number of interpretable concept detectors is 407. This preliminary statistical report, together with the visualizations, is provided in order to o er intuition regarding the interpretability of the concept detectors. A rigorous evaluation with comparisons is conducted in the Sections that follow.

7.4 Metrics

7.4.1 Evaluation of Clustering Quality via Cluster Statistics

We report two statistics for directional clustering. Coverage is the percentage of patches in the concept dataset that receive at least one positive prediction from the concept detectors. Equivalently, it is the percentage of pixels assigned to at least one cluster or explained by the clustering. Redundancy is the average number of detectors that classify each patch positively. Equivalently, it is the average number of clusters to which each patch belongs.

7.4.2 Evaluation of the Decoding Directions: Interpretability

Classi cation Let T_i denote the binary classi cation task of predicting whether a patch belongs to concept i. Given the concept label l, which was assigned to a concept detector with direction labeling, we assess the detector's performance on T_i using the standard binary classi cation metrics of Precision, Recall, F1, and Average Precision (AP). For each metric, we report average and standard deviation scores by aggregating across all detectors in the learned set.

Segmentation We additionally consider the performance of concept detectors in the binary semantic segmentation task and employ two metrics from Doumanoglou et al. (2023). Speci cally, let $_i(c;K)$ the Intersection Over Union for concept detector i in identifying concept c within the dataset K. De ne $c_i^? = \text{argmax}_c \ _i(c;K_t)$, indicating the concept label detected best by concept detector i within the training subset of the dataset (K_t). With K_v as the validation subset, we use the following interpretability scores S^1 and S^2:

$$S^1 = \int_0^1 \sum_{i=0}^{X} 1_{x \quad _i(c_i;K_v)} \, d \tag{14}$$

$$S^2 = \int_0^1 \left| c_i^? \; j \, 9 i : \quad _i(c_i^?;K_v) \quad g \right| d \tag{15}$$

The rst metric S^1 counts concept detectors with an IoU performance that exceeds a score threshold . The second metric S^2 uses the cardinality of the set j:j to count the unique concept labels detected by the concept detectors with IoU above . Both metrics become threshold-agnostic, by integrating over all 2 [0;1]. Viewing it from a di erent angle, S^1 pertains to the segmentation e ciency of each concept detector individually, whereas S^2 concerns the clustering diversity, meaning the capability of the detector ensemble to recognize a wide array of distinct concepts. Finally, we also report the standard mean Intersection over Union (mIoU) score, which is the average IoU performance score aggregated across all the detectors in the set.

Monosemanticity Even though Precision is a natural classi cation metric that can capture the monosemanticity of the clustering, it has the drawback of relying on the explicit labels that are available in the concept dataset. To overcome this labeling limitation, and inspired by Dreyer et al. (2024b), we additionally quantify the monosemanticity of the concepts identi ed by the concept detectors by utilizing the embedding space of CLIP ViT/B-16 (Radford et al., 2021). In particular, for each concept detector, we pick images from the validation split of the concept dataset for which the detector is most con dent about the presence of the concept. We select 100 unique images from the top con dent predictions. Subsequently, we crop a rectangle around the area of the concept and obtain the CLIP embedding for this crop. Let k_1^i and k_2^i denote image indices in the set of selected images, with $k_1^i, k_2^i \in \{0,1,\ldots,99\}$ and i denoting the i-th concept detector. We use the following monosemanticity metric, which measures the average distance between CLIP embeddings $(e_{k_1^i}; e_{k_2^i})$ within a cluster, aggregated over all clusters:

$$M = \mathop{E}_i \left[\mathop{E}_{k_1^i \neq k_2^i} \left\| e_{k_1^i} - e_{k_2^i} \right\|_2 \right] \tag{16}$$

Table 4: Comparative Analysis of our Encoding-Decoding Direction Pairs (EDDP) against unsupervised baselines. This table summarizes metrics for ResNet18 trained on Places365. EDDP-U stands for using Unconstrained Feature Manipulation and EDDP-C for using Constrained Feature Manipulation.

				ResNet18 / Places365								
I	Method	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S ^{1} "	S^2 "	mIoU "	I ^{1} "
384	PCA	0.72	7.27	0.26 0.13	0.14 0.11	0.18 0.11	0.17 0.11	7.47	16.57	5.6	0.04	0.73
	NMF	0.57	2.06	0.62 0.19	0.21 0.14	0.31 0.17	0.4 0.18	6.88	29.56	17.43	0.08	0.68
	EDDP-U	0.86	1.29	0.82 0.21	0.23 0.16	0.33 0.18	0.53 0.19	6.61	41.73	28.01	0.12	0.58
	EDDP-C	0.86	1.32	0.81 0.21	0.23 0.16	0.33 0.17	0.5 0.19	6.67	41.5	27.49	0.12	0.61
448	PCA	0.75	8.89	0.25 0.12	0.14 0.11	0.17 0.11	0.16 0.1	7.48	18.06	5.62	0.04	0.73
	NMF	0.51	1.79	0.67 0.18	0.21 0.14	0.31 0.17	0.43 0.18	6.96	31.77	18.16	0.07	0.69
	EDDP-U	0.86	1.32	0.83 0.19	0.22 0.16	0.32 0.18	0.53 0.19	6.64	47.5	32.07	0.11	0.59
	EDDP-C	0.86	1.33	0.82 0.2	0.21 0.16	0.31 0.18	0.49 0.19	6.72	46.3	31.34	0.11	0.61
512	Natural	0.49	1.79	0.69 0.18	0.2 0.13	0.3 0.16	0.43 0.17	7.02	34.61	18.74	0.07	0.69
	PCA	0.77	10.05	0.25 0.12	0.14 0.11	0.17 0.11	0.16 0.1	7.5	19.41	5.62	0.04	0.74
	EDDP-U	0.86	1.28	0.82 0.22	0.21 0.17	0.31 0.19	0.51 0.21	6.66	52.51	37.78	0.11	0.61
	EDDP-C	0.86	1.28	0.81 0.24	0.2 0.16	0.29 0.19	0.47 0.2	6.73	50.29	36.99	0.1	0.63

7.4.3 Evaluation of the Encoding Directions: In uence

We use RCAV Pfau et al. (2020) to assess the ability of our method to identify concepts in uential to model predictions and relate concept in uence to the network's concept sensitivity. In the ablation studies, two metrics summarize the results: Signi cant Direction Count (SDC) and Signi cant Class-Direction Pairs (SCDP). SDC represents the number of signal vectors that signi cantly in uence at least one model class, while SCDP tallies class-direction pairs where the signal vectors signi cantly a ect the class. Let $S_{C;i;k}$ denote the sensitivity of the model to concept i when predicting class k. When comparing against unsupervised baselines, we use the following in uence metric I^1 which is de ned as:

$$I^1 = E_{i;k}\left[\,|S_{C;i;k}|\,\right] \tag{17}$$

i.e., the average sensitivity of the model across all concepts and classes.

7.5 Interpretability and In uence Comparison with Established Unsupervised Baselines

The previous unsupervised state-of-the-art (Graziani et al., 2023a;b; Zhang et al., 2021; Fel et al., 2023b) has demonstrated e ectiveness in revealing interpretable directions in the latent space of deep networks. However, as outlined in Section A.13, these approaches employed a class-conditioned protocol, tying their analyses to speci c class labels. In this work, we instead operate under a class-agnostic protocol, which seeks concepts across the entire latent space, independent of class. This fundamental di erence in the approaches renders a direct performance comparison with previous implementations inapplicable. Therefore, we utilize PCA and NMF, the core matrix decomposition methods upon which these prior works were built, as our principal baselines for a fair, classical comparison. We also move beyond subjective human ratings typically used for evaluation and instead develop a rigorous, best-e ort quantitative framework to compare interpretability and in uence with these baselines, with a more detailed discussion on how our pipeline captures human judgment provided in Section A.10.

In Section A.16, we also compare with the methods we extend (Doumanoglou et al., 2023; 2024), but under a di erent protocol for a fair comparison. For each network, we consider three sizes for the clustering, namely I = 3=4D, I = 7=8D, and I = D. It is worth noting that when I = D, NMF trivially resolves to the natural latent space basis, i.e., the neuron directions. Additionally, when studying E cientNet, we disregard NMF since this network uses the SiLU activation function (Ramachandran et al., 2017), which allows negative feature activations, rendering NMF non-applicable. Tables 4, 5, 6, 7, and 8 summarize the comparative analysis for ResNet18, E cientNet, Inception-v3, VGG16, and ResNet50, respectively. In the following subsections, we provide a detailed discussion of these tables.

7.5.1 Case Study: ResNet18 / Places365

Clustering Quality: Regarding dataset coverage, both our EDDP variants explain 86% of the dataset regardless of cluster count. Speci cally, this means that 86% of the patch embeddings in the concept dataset have been assigned to at least one cluster. This number is substantially higher than the coverage of the clustering obtained via NMF (which is at most 57%) or the natural latent space basis (49%) and still signi cantly higher than the coverage of the clustering obtained via PCA (at most 77%). As far as redundancy is concerned, the EDDP variants minimize multiple cluster assignments to individual patches, as they attain the lowest redundancy scores (at most 1:33 cluster assignments per patch). While NMF and the natural basis rank 3rd in redundancy terms with more than 1:77 assignments per patch, PCA is ranked last with the same number exceeding 7:27.

Interpretability: In classi cation metrics, the clustering of EDDP-U is the most interpretable of all, scoring higher than alternative clusterings, especially in Precision and AP terms. In most cases, clustering with EDDP-C attains scores close to clustering with EDDP-U. Thus, EDDP-C is ranked 2nd in all classi cation metrics except in the case of F1 score and $I = 512$, in which the natural basis is slightly better by a score point of 0:01. With only the latter exception, in all classi cation metrics, NMF and the natural basis are ranked 3rd, while PCA is ranked last.

The high precision of our concept detectors (approximately 0:82 on average) indicates that the learned directions are highly monosemantic. The latter is additionally con rmed by the monosemanticity metric based on CLIP embedding distances, in which EDDP variants are attributed the lowest M score. Comparing with the rest of the approaches, all of the clusterings obtained by NMF, the natural basis, and PCA are less monosemantic, often by a large margin (0:67 precision for NMF, 0.26 precision for PCA and 0:69 for the natural basis).

Similar conclusions can be drawn from the segmentation metrics, in which the concept detectors of EDDP score substantially higher than the previous approaches. For example, when $I = 448$, EDDP-C achieves 46:3 score points in S^1, surpassing NMF's 31:77, and reaches 0:11 mIoU points in contrast to NMF's 0:07. Finally, the EDDP variants also achieve the most diverse clustering, covering a variety of visual concepts, as justi ed by the substantially higher values of S^2 compared to the rest of the approaches. Furthermore, this diversity is consistently improved with the increase in cluster count, a phenomenon that is less prominent for the rest of the baselines.

In uence: To rank the methods based on the network's sensitivity to the identi ed concepts (metric I^1), starting from the most in uential clustering, we would list PCA rst, then NMF, followed by the Natural basis, and nally EDDP-C and EDDP-U. In this case, it appears that there is a clear anti-correlated relationship between interpretability and in uence, with the most interpretable clustering being less in uential on the overall model outcomes. We will be discussing this further at the end of the section.

7.5.2 Case Study: E cientNet / ImageNet

Table 5: Comparative Analysis of our Encoding-Decoding Direction Pairs (EDDP) against unsupervised baselines. This table summarizes metrics for E cientNet (b0) trained on ImageNet. EDDP-U stands for using Unconstrained Feature Manipulation and EDDP-C for using Constrained Feature Manipulation.

				E cientNet / ImageNet								
I	Method	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "	I^1 "
960	PCA	0.63	19.29	0.29 0.08	0.13 0.07	0.18 0.07	0.19 0.07	7.44	35.26	2.62	0.04	0.95
	EDDP-U	0.85	1.51	0.75 0.22	0.2 0.17	0.27 0.17	0.38 0.16	6.74	27.92	14.71	0.03	0.94
	EDDP-C	0.85	1.48	0.72 0.23	0.2 0.18	0.27 0.17	0.38 0.17	6.75	25.92	14.45	0.03	0.94
1120	PCA	0.67	24.7	0.28 0.07	0.13 0.07	0.17 0.07	0.19 0.07	7.44	42.33	2.7	0.04	0.95
	EDDP-U	0.85	1.61	0.76 0.21	0.2 0.17	0.28 0.17	0.39 0.17	6.73	31.36	16.54	0.03	0.94
	EDDP-C	0.85	1.54	0.74 0.22	0.2 0.18	0.28 0.19	0.39 0.17	6.74	28.62	16.38	0.03	0.94
1280	Natural	0.51	5.44	0.56 0.13	0.14 0.07	0.21 0.1	0.29 0.12	7.35	35.94	7.09	0.03	0.95
	PCA	0.68	26.71	0.28 0.08	0.13 0.07	0.17 0.07	0.18 0.07	7.44	48.01	3.12	0.04	0.95
	EDDP-U	0.85	1.65	0.74 0.21	0.19 0.16	0.27 0.16	0.36 0.17	6.81	40.38	19.12	0.03	0.94
	EDDP-C	0.85	1.6	0.73 0.21	0.19 0.17	0.26 0.17	0.36 0.17	6.88	36.95	18.98	0.03	0.94

Clustering Quality: Clusters created by the EDDP variants attain a Coverage score of 85%, while the best values for PCA and the natural basis in the same metric are 68% and 51%, respectively. In Redundancy terms, EDDP achieves scores less than 1:65, while the minimum scores for PCA and the natural basis are 19:29 and 5:44, respectively.

Interpretability: In terms of classi cation metrics, the directional clusterings obtained via EDDP-U and EDDP-C are approximately equally interpretable and notably more interpretable than the directional clustering of PCA. For instance, when I = 1120, the mean Precision of PCA's concept detectors is 0:28, the mean Recall 0:13, the mean F1 score 0:17, while the mean AP is equal to 0:19. For EDDP-U, the mean scores for the same metrics are 0:76 for Precision, 0:2 for Recall, 0:28 for F1 score, and 0:39 for AP. Similar conclusions about the improved interpretability of EDDP can be drawn when comparing with the natural latent space basis in the case of I = 1280.

Regarding monosemanticity, the concept detectors of EDDP attain a mean Precision score which is better than 0:72, in contrast to the ones of PCA, which attain mean Precision scores below 0:29. The same gap between PCA and EDDP is similarly re ected in the M score. For example, when I = 1120, the M score of EDDP-C is 6:74 while the score of PCA is 7:44. EDDP also demonstrates a substantial improvement in monosemanticity compared to the natural basis. EDDP-C attains a mean Precision score of 0:73, while the score of the natural basis for the same metric is only 0:56. Additionally, EDDP-C attains 6:88 points in terms of M, which is signi cantly lower than the 7:35 points of the natural basis.

In the context of segmentation, when I = 1120, PCA's concept detectors exhibit improved performance concerning S^1, exceeding EDDP-U by roughly 12 points. In mIoU terms, PCA still scores higher (0:04), but EDDP follows closely (0:03). Yet, PCA exhibits a narrow diversity in the clustering, discovering substantially fewer concepts than EDDP, as re ected in S^2 in which EDDP-U surpasses PCA by up to approximately 14 points. When considering the natural basis as an alternative directional clustering, EDDP scores better in terms of S^1 and S^2 and equally well in mIoU.

In uence: With respect to the impact of the learned directions on the network predictions, E cientNet shows a signi cant sensitivity to the direction set determined by PCA or the neuron directions from the natural basis, with a I^1 score of 0:95. Even though the model is less in uenced by the directions learned with EDDP, this is only by a small margin, as the I^1 scores for EDDP variants are 0:94.

7.5.3 Case Study: Inception-v3 / ImageNet

Table 6: Comparative Analysis of our Encoding-Decoding Direction Pairs (EDDP) against unsupervised baselines. This table summarizes metrics for Inception-v3 trained on ImageNet. EDDP-U stands for using Unconstrained Feature Manipulation and EDDP-C for using Constrained Feature Manipulation.

				Inception-v3 / ImageNet								
I	Method	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "	I^1 "
1536	PCA	0.7	32.29	0.26 0.12	0.12 0.1	0.16 0.1	0.17 0.1	7.39	54.5	4.19	0.04	0.95
	NMF	0.7	9.26	0.53 0.17	0.16 0.12	0.24 0.14	0.28 0.16	7.24	70.68	14.92	0.05	0.95
	EDDP-U	0.87	2.41	0.84 0.18	0.28 0.24	0.38 0.26	0.53 0.26	6.79	156.46	28.03	0.11	0.93
	EDDP-C	0.85	1.9	0.86 0.19	0.23 0.16	0.33 0.19	0.58 0.24	6.87	125.64	27.46	0.08	0.94
1792	PCA	0.68	33.26	0.28 0.12	0.12 0.1	0.17 0.1	0.17 0.1	7.39	59.18	4.1	0.03	0.95
	NMF	0.64	7.71	0.6 0.18	0.16 0.13	0.25 0.16	0.32 0.17	7.29	72.86	15.86	0.04	0.95
	EDDP-U	0.9	2.96	0.86 0.16	0.33 0.25	0.43 0.26	0.58 0.25	6.77	210.44	25.77	0.13	0.93
	EDDP-C	0.86	2.19	0.89 0.16	0.24 0.16	0.36 0.19	0.62 0.21	6.92	174.0	24.69	0.1	0.94
2048	Natural	0.63	10.29	0.58 0.17	0.17 0.13	0.25 0.15	0.31 0.17	7.35	84.62	14.27	0.04	0.95
	PCA	0.71	42.35	0.26 0.12	0.12 0.1	0.16 0.1	0.17 0.1	7.38	69.87	4.24	0.03	0.95
	EDDP-U	0.86	2.36	0.85 0.2	0.31 0.29	0.4 0.29	0.54 0.28	6.95	170.81	32.13	0.09	0.94
	EDDP-C	0.93	3.98	0.75 0.18	0.16 0.12	0.24 0.14	0.38 0.16	7.09	92.67	31.18	0.05	0.94

Clustering Quality: On par with the experiments on the previous architectures, EDDP variants explain more than the 85% of the concept dataset, as noted by the respective Coverage scores. This score is still signi cantly higher than the ones attained from NMF (up to 70%), the natural basis (63%) or PCA (up to 71%). As far as Redundancy is concerned, on one hand, EDDP does not exceed 3:98 cluster assignments per patch. On the other hand, NMF assigns a patch to more than 7:71 clusters, while for PCA this number

exceeds 32:29. Finally, in the clustering obtained by the use of the natural basis, each patch is assigned to 10:29 clusters.

Interpretability: For I = 1536 and I = 1792, the concept detectors of the EDDP variants score substantially higher in the classi cation metrics than the rest of the approaches. For example, when I = 1792, EDDP-C attains a Precision score of 0:89, whereas the same score for NMF is 0:6 and for PCA 0:28. For the same I, EDDP-U attains an F1 score of 0:43, which is a notable improvement over NMF's 0:25 and PCA's 0:17. Finally, in terms of AP, EDDP-C achieves a score of 0:62 points, which is signi cantly more competitive than the score of NMF (0:32) and PCA (0:17). For I = 2048, concept detectors from EDDP-U consistently score higher in the classi cation metrics than any other approach. Unlike what we have seen in the rest of the experiments, in that case, EDDP-C did not follow EDDP-U closely in the classi cation metrics. Instead, it is clearly ranked 2nd, surpassing PCA and the natural basis in Precision and AP while remaining comparable with the natural basis in Recall and F1 score.

Regarding monosemanticity, EDDP-U attains Precision scores close to 0:84, while for the cases of I = 1536 and I = 1792, EDDP-C exceeds this number, achieving scores up to 0:89 but attains a lower score of 0:75 when I = 2048. The best Precision score for NMF is 0:6 for I = 1792, while for PCA this score is 0:28. Finally, when clustering with the directions of the neurons, i.e. the natural basis, the Precision score is 0:58. When measuring monosemanticity with the CLIP embedding distances, the EDDP variants achieve a substantially lower score compared to the rest of the approaches. For instance, when I = 1792, the EDDP-U clustering attains a score of 6:77, EDDP-C 6:92, NMF 7:29 and PCA 7:39. Interestingly, unlike what we have seen when using the labeled dataset in terms of Precision, when I = 2048 EDDP-C and EDDP-U score similarly in M, with EDDP-U attaining a score of 6:95 and EDDP-C 7:09.

As far as segmentation is concerned, the ranking of the methods is consistent across all metrics: EDDP-U comes rst, followed by EDDP-C, then NMF or the natural basis, and nally PCA. Beyond just ranking, the gap in the scores between EDDP and other approaches is large. For instance, when I = 1536, the EDDP-U score in terms of S^1 is 156:46 when NMF's is 70:68. Similarly, for the same case, EDDP-U's ^{2}Sis 28:03 when NMF's is 14:92, while for mIoU, EDDP-U's score is 0:11 when NMF's is 0:05.

In uence: Regarding the sensitivity of the model with respect to the identi ed concepts, Inception-v3 is more sensitive to the PCA and NMF directions with a I^1 score of 0:95. The network is slightly less sensitive to the concepts identi ed by EDDP-C and EDDP-U, but once again by a small margin. EDDP-C attains a I^1 score of 0:94 and EDDP-U a score in {0:93; 0:94g.

7.5.4 Case Study: VGG16 / ImageNet

Table 7: Comparative Analysis of our Encoding-Decoding Direction Pairs (EDDP) against unsupervised baselines. This table summarizes metrics for VGG16 trained on ImageNet. EDDP-U stands for using Unconstrained Feature Manipulation and EDDP-C for using Constrained Feature Manipulation.

				VGG16 / ImageNet								
I	Method	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "	I^1 "
384	PCA	0.64	7.31	0.29 0.09	0.15 0.07	0.19 0.07	0.16 0.07	7.85	22.44	2.98	0.06	0.87
	NMF	0.55	1.6	0.6 0.16	0.17 0.11	0.26 0.13	0.31 0.14	7.44	22.73	9.7	0.06	0.87
	EDDP-U	0.82	1.17	0.61 0.18	0.18 0.09	0.27 0.09	0.31 0.12	6.88	29.98	9.39	0.08	0.91
	EDDP-C	0.79	1.07	0.63 0.18	0.16 0.09	0.24 0.09	0.31 0.12	6.92	26.23	8.89	0.07	0.9
448	PCA	0.67	8.91	0.28 0.08	0.15 0.07	0.19 0.07	0.16 0.06	7.86	25.8	3.12	0.06	0.87
	NMF	0.5	1.44	0.62 0.16	0.16 0.1	0.25 0.12	0.31 0.14	7.44	24.82	9.86	0.06	0.87
	EDDP-U	0.84	1.35	0.59 0.18	0.18 0.09	0.26 0.09	0.3 0.11	6.93	34.67	10.29	0.08	0.91
	EDDP-C	0.82	1.22	0.61 0.18	0.16 0.08	0.24 0.09	0.3 0.11	6.93	30.91	9.26	0.07	0.9
512	Natural	0.58	2.12	0.58 0.16	0.16 0.1	0.25 0.12	0.29 0.12	7.5	30.61	10.28	0.06	0.87
	PCA	0.69	9.62	0.29 0.08	0.15 0.06	0.19 0.07	0.16 0.06	7.87	30.25	3.51	0.06	0.86
	EDDP-U	0.85	1.43	0.61 0.18	0.18 0.08	0.27 0.09	0.31 0.11	6.95	41.2	11.11	0.08	0.91
	EDDP-C	0.85	1.37	0.62 0.18	0.15 0.09	0.24 0.09	0.3 0.11	7.09	33.91	10.83	0.07	0.9

Clustering Quality: In terms of clustering quality metrics, the EDDP variants attain higher Coverage and lower Redundancy scores than either one of NMF, PCA, or the natural basis, in a similar fashion as in the rest of the architectures.

Interpretability: Unlike the other architectures, in classi cation metrics, NMF and the EDDP variants perform on par. For example, when I = 448, NMF attains a Precision score of 0:62 exceeding EDDP-U's 0:59. In Recall and F1 score terms, EDDP-U achieves the highest scores of 0:18 and 0:26, respectively, while for NMF these numbers are 0:16 and 0:25. In terms of AP, NMF scores 0:31 while both EDDP variants attain a score of 0:30. In most cases, all of NMF, the natural basis, and EDDP substantially surpass the clustering of PCA in the classi cation metrics.

Even though EDDP, NMF, and the natural latent space basis score similarly in terms of Precision, if monosemanticity is measured via the CLIP embedding distances, the EDDP variants achieve the best scores, often by a substantial margin. For instance, when I = 384, NMF achieves an M score of 7:44, PCA a score of 7:85, while EDDP-U a score of 6:88.

Finally, in segmentation metrics, EDDP variants achieve the highest scores in S^1 and mIoU, while in S^2, EDDP-U is ranked 1st in two of the three cases (I = 448 and I = 512) while ranked 2nd, after NMF in the third case (I = 384).

In uence: Regarding the sensitivity of VGG16 with respect to the concepts, the network is most sensitive to the directions learned with EDDP-U (I^1 score of 0:91), followed by EDDP-C (I^1 score of 0:9), and nally, by NMF, the natural basis, or PCA, with I^1 scores close to 0:87.

7.5.5 Case Study: ResNet50 / Moments in Time

Table 8: Comparative Analysis of our Encoding-Decoding Direction Pairs (EDDP) against unsupervised baselines. This table summarizes metrics for ResNet50 trained on Moments in Time (MiT). EDDP-U stands for using Unconstrained Feature Manipulation and EDDP-C for using Constrained Feature Manipulation.

				ResNet50 / MiT								
I	Method	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "	I^1 "
1536	PCA	0.7	30.77	0.16 0.06	0.1 0.05	0.12 0.05	0.1 0.04	7.34	58.51	3.55	0.04	0.88
	NMF	0.54	9.07	0.46 0.17	0.16 0.11	0.23 0.13	0.27 0.13	6.91	73.38	17.31	0.05	0.85
	EDDP-U	0.84	2.13	0.72 0.22	0.17 0.12	0.26 0.15	0.36 0.16	6.25	127.83	33.67	0.08	0.85
	EDDP-C	0.88	2.37	0.75 0.19	0.17 0.11	0.26 0.13	0.45 0.14	6.49	98.52	34.81	0.06	0.84
1792	PCA	0.72	36.06	0.16 0.06	0.1 0.05	0.12 0.05	0.1 0.04	7.35	65.82	3.58	0.04	0.88
	NMF	0.42	6.02	0.57 0.15	0.15 0.08	0.23 0.1	0.3 0.11	6.95	77.02	19.0	0.04	0.85
	EDDP-U	0.85	2.47	0.78 0.16	0.19 0.1	0.29 0.13	0.4 0.14	6.24	162.86	34.17	0.09	0.85
	EDDP-C	0.88	2.62	0.76 0.16	0.16 0.11	0.25 0.13	0.44 0.13	6.5	113.9	36.68	0.06	0.84
2048	Natural	0.5	11.18	0.47 0.16	0.15 0.09	0.22 0.11	0.26 0.12	6.97	90.35	16.63	0.04	0.85
	PCA	0.76	43.2	0.15 0.06	0.1 0.05	0.12 0.05	0.1 0.04	7.35	74.3	3.61	0.04	0.88
	EDDP-U	0.86	3.15	0.72 0.2	0.21 0.13	0.31 0.16	0.43 0.16	6.29	209.2	33.04	0.1	0.85
	EDDP-C	0.88	2.83	0.72 0.16	0.15 0.11	0.24 0.12	0.43 0.13	6.61	126.46	35.98	0.06	0.84

Clustering Quality: Once again, in terms of clustering quality, the EDDP variants demonstrate both larger Coverage and lower Redundancy compared to the rest of the approaches, often with a large margin. When I = 1792, the coverage score for EDDP-C is 0:88 while for NMF it is 0:42. The gap between the two is more than twice the score of NMF. Similarly, the redundancy score of EDDP-C is 2:62 whereas NMF's is 6:02, with a similar conclusion regarding the gap between the two.

Interpretability: The clustering obtained via EDDP is more interpretable in terms of classi cation metrics, especially in Precision, AP, and sometimes in the F1 score as well. Regardless of whether monosemanticity is measured via Precision or distances between CLIP embeddings, the EDDP variants score better, similar to the rest of the experiments.

When considering segmentation, both EDDP variants demonstrate consistent superiority in terms of S^1 and S^2 and mIoU. Interestingly, in this experiment, EDDP-C scores the highest in S^2, indicating a clustering that captures a larger variety of concepts.

In uence: In terms of the concept sensitivity metric I^1, NMF, EDDP-C, EDDP-U and the natural basis perform on par, with an approximate score of 0:85. On the contrary, PCA achieves a score of 0:88, which puts it further ahead of the competition.

7.5.6 Discussion

From the previous experiments, we can draw the following, mostly straightforward conclusions: a) EDDP's directional clusters are monosemantic, often surpassing the unsupervised baselines by a signi cant margin. b) EDDP also excels in terms of F1 score, where EDDP-U is consistently ranked rst among the competition, and EDDP-C is often ranked second. c) In terms of segmentation and speci cally S^1, EDDP-U is consistently ranked rst, except for E cientNet, in which the clustering of PCA scores better, yet with substantially inferior performance in most of the other interpretability metrics. In the same metric, EDDP-C is often ranked second. d) In terms of discovering a diverse set of concepts (S^2), the EDDP variants score the best, often by a large margin, except for one case in VGG16 where they score on par with the best method. e) In terms of in uence, in all cases except VGG16, the EDDP variants are ranked last. However, in many cases, the respective score is not far behind the competition.

In most cases, and in a broader sense, we observe that interpretability is somewhat negatively correlated with in uence. Less interpretable directions may represent complex interactions (i.e. correspond to less pure human-understandable concepts) and, as a result, they are more impactful on the network predictions, in the sense that they a ect many output classes. On the contrary, more interpretable directions may have a more targeted impact on speci c output classes (Gao et al., 2025), and thus score lower in the respective metric.

7.6 Ablation Study

7.6.1 Interpretability Losses

Table 9: Ablation study with respect to the interpretability losses. For ResNet18 I = 448, and for E cientNet I = 1120.

Ortho	L^{fs}	L^{eac}	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "
ResNet18 / Places365												
3	7	7	0.89	2.25	0.71 0.19	0.3 0.18	0.4 0.19	0.47 0.19	6.44	54.28	26.4	0.13
7	7	7	0.86	1.09	0.88 0.17	0.18 0.12	0.29 0.15	0.54 0.17	6.57	40.92	19.1	0.1
7	3	7	0.83	1.13	0.85 0.19	0.19 0.14	0.29 0.16	0.53 0.18	6.57	39.26	25.36	0.1
7	3	3	0.85	1.29	0.83 0.19	0.22 0.15	0.31 0.17	0.52 0.18	6.64	47.14	29.0	0.11
E cientNet / ImageNet												
3	7	7	0.9	3.36	0.67 0.19	0.17 0.14	0.25 0.14	0.34 0.14	7.0	46.9	14.42	0.04
7	7	7	0.84	1.29	0.78 0.23	0.23 0.21	0.3 0.2	0.39 0.18	6.81	19.54	13.29	0.02
7	3	7	0.86	1.59	0.78 0.2	0.2 0.16	0.29 0.16	0.4 0.17	6.68	25.42	14.1	0.03
7	3	3	0.85	1.57	0.74 0.2	0.22 0.18	0.3 0.17	0.38 0.17	6.68	28.72	16.02	0.03

The motivation behind the loss terms that we introduced in Section 4.2 is to sustain or improve the interpretability of the clustering when moving from an orthogonal decoding direction set to a non-orthogonal one. In Table 9 we present interpretability metrics for the cases when we learn the directions with L^{fs}, both L^{fs} and L^{eac}, or without any of them. We also provide metrics for the case of orthogonal decoding directions (marked as Ortho). The latter is equivalent to learning with the method of Doumanoglou et al. (2023) using the Augmented Lagrangian loss of Section 4.5. We provide results regarding the directions learned for ResNet18 and E cientNet. In these experiments, we exclude any form of Uncertainty Region Alignment.

In both network case studies, starting without our interpretability losses and gradually adding them one after the other, we observe that the di erent variations a) compare similarly in coverage and redundancy terms and b) achieve comparable performance in classi cation and monosemanticity metrics. However, the impact of the loss terms on the segmentation metrics is more signi cant, especially in terms of S^1 and S^2. We emphasize the improvement in S^2 since this metric captures the variety of the identi ed concepts. Compared to the orthogonal direction set, directions learned without orthogonality constraints a) result in a less redundant clustering, b) the monosemanticity of the clustering in terms of Precision is improved, and c) the clustering becomes more diverse, capturing a variety of di erent concepts.

Table 10: Ablation study with respect to the Unconstrained Uncertainty Region Alignment loss. For ResNet18 I = 448, for E cientNet I = 1120, and for ResNet50 I = 1792.

Ortho	L^{uur}	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "
ResNet18 / Places365											
3	7	0.89	2.25	0.71 0.19	0.3 0.18	0.4 0.19	0.47 0.19	6.44	54.28	26.4	0.13
3	3	0.88	2.2	0.71 0.18	0.31 0.18	0.41 0.19	0.48 0.19	6.52	54.56	26.06	0.13
7	7	0.85	1.29	0.83 0.19	0.22 0.15	0.31 0.17	0.52 0.18	6.64	47.14	29.0	0.11
7	3	0.86	1.32	0.83 0.19	0.22 0.16	0.32 0.18	0.53 0.19	6.64	47.5	32.07	0.11
E cientNet / ImageNet											
3	7	0.9	3.36	0.67 0.19	0.17 0.14	0.25 0.14	0.34 0.14	7.0	46.9	14.42	0.04
3	3	0.9	3.34	0.68 0.2	0.17 0.14	0.25 0.14	0.36 0.15	6.98	52.94	16.38	0.05
7	7	0.85	1.57	0.74 0.2	0.22 0.18	0.3 0.17	0.38 0.17	6.68	28.72	16.02	0.03
7	3	0.85	1.61	0.76 0.21	0.2 0.17	0.28 0.17	0.39 0.17	6.73	31.36	16.54	0.03
ResNet50 / MiT											
3	7	0.88	3.44	0.72 0.15	0.16 0.1	0.25 0.13	0.35 0.14	6.47	121.67	29.41	0.07
3	3	0.88	3.31	0.73 0.16	0.18 0.1	0.27 0.13	0.37 0.13	6.49	137.76	33.0	0.08
7	7	0.83	2.42	0.75 0.19	0.15 0.12	0.24 0.15	0.32 0.14	6.1	157.54	30.81	0.09
7	3	0.85	2.47	0.78 0.16	0.19 0.1	0.29 0.13	0.4 0.14	6.24	162.86	34.17	0.09

Table 11: Ablation study with respect to the Uncertainty Region Alignment losses. For ResNet18 I = 448, for E cientNet I = 1120, and for ResNet50 I = 1792.

L^{uur}	L^{cur}	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	M #	S^1 "	S^2 "	mIoU "	SDC "	SCDP "
ResNet18 / Places365													
7	7	0.85	1.29	0.83 0.19	0.22 0.15	0.31 0.17	0.52 0.18	6.64	47.14	29.0	0.11	291	2104
3	7	0.86	1.32	0.83 0.19	0.22 0.16	0.32 0.18	0.53 0.19	6.64	47.5	32.07	0.11	264	1565
7	3	0.86	1.33	0.82 0.2	0.21 0.16	0.31 0.18	0.49 0.19	6.72	46.3	31.34	0.11	304	1692
E cientNet / ImageNet													
7	7	0.85	1.57	0.74 0.2	0.22 0.18	0.3 0.17	0.38 0.17	6.68	28.72	16.02	0.03	93	216
3	7	0.85	1.61	0.76 0.21	0.2 0.17	0.28 0.17	0.39 0.17	6.73	31.36	16.54	0.03	107	243
7	3	0.85	1.54	0.74 0.22	0.2 0.18	0.28 0.19	0.39 0.17	6.74	28.62	16.38	0.03	673	1484
ResNet50 / MiT													
7	7	0.83	2.42	0.75 0.19	0.15 0.12	0.24 0.15	0.32 0.14	6.1	157.54	30.81	0.09	370	548
3	7	0.85	2.47	0.78 0.16	0.19 0.1	0.29 0.13	0.4 0.14	6.24	162.86	34.17	0.09	393	570
7	3	0.88	2.62	0.76 0.16	0.16 0.11	0.25 0.13	0.44 0.13	6.5	113.9	36.68	0.06	1353	3913

7.6.2 Uncertainty Region Alignment

In this section, we study how Uncertainty Region Alignment may a ect the interpretability of clustering and the discovery of concepts with signi cant in uence. For this study, we consider three network architectures: ResNet18, E cientNet, and ResNet50.

In Table 10 we compare, in quality and interpretability terms, the clusterings obtained with and without the Unconstrained Uncertainty Region Alignment loss (L^{uur}). We consider learning the decoding directions both with and without orthogonality constraints (Ortho). Some simple conclusions that can be drawn from the experiments are: a) Coverage may be improved when using it but Redundancy can be slightly worsened, b) the Precision and AP of the concept detectors are consistently improved by its use, c) it consistently improves the performance of concept detectors in terms of S^1, and d) L^{uur} can signi cantly improve the variety of concepts captured by the direction set (S^2 metric).

In Table 11 we focus on comparing between the two Uncertainty Region Alignment variants by additionally considering metrics of concept in uence. Some key takeaways drawn from these experiments for using L^{cur} are: a) Coverage may be improved by its use, but Redundancy can be a bit worsened, b) the concept detectors' interpretability in terms of classi cation metrics is slightly inferior but somewhat comparable to when using L^{uur}, c) in two of the three cases (ResNet18 and E cientNet) the segmentation metrics across Uncertainty Region Alignment variants are mostly comparable, but in the third case (ResNet50) L^{cur} leads to a more diverse clustering (improved S^2) but at the cost of a lower S^1 score. d) In all cases, when comparing the two uncertainty region variants, the in uence metrics favor L^{cur}. In two of the three cases (E cientNet and ResNet50) this improvement is substantially higher than what can be achieved when using L^{uur}. For a more complete picture regarding the e ect of L^{fso}, which is used in EDDP-C but not in EDDP-U, see also Section A.15.

Overall, by this ablation study, we could conclude that the use of L^{uur} can improve the diversity of the clustering by capturing a larger variety of concepts, compared to when not using it. Using L^{cur} instead of L^{uur} may perform on par or improve this diversity, while substantially increasing the in uence of the learned signal vectors on the network predictions, in a statistically signi cant sense.

7.7 Insights on Interpretability and In uence in comparison to Supervised Direction Learning

In this section, we compare our method with supervised direction learning, aiming to draw insights regarding interpretability and in uence of concept directions. For this reason, we experiment with the last convolutional layer of ResNet18 trained on Places365. We consider our method with the constrained feature manipulation strategy, and two supervised learning approaches: a) IBD (Zhou et al., 2018) for supervised learning of decoding directions (i.e. concept detectors) and b) Pattern-CAVs (PCAVs) (Pahde et al., 2024) for supervised estimation of concept encoding directions. The supervised concept detectors and PCAVs are learned for the labels assigned to EDDP's concept detectors at the direction labeling phase (Section 3.4).

Interpretability: In the interpretability analysis, we consider three variants for the proposed method: a) the exact outcome of our method, marked as EDDP, b) combining directions with a shared label (post initial learning) using a binary linear classi cation layer which classi es representations positively if any detector with the same label does (marked as EDDP - Linear OR), and c) considering the learned directions but optimizing the classi cation threshold in a supervised manner, in a post-learning step, to enhance the F1 Score (marked as EDDP /w sup thres). This last approach assesses direction alignment to the concept by relaxing the sparsity constraint of the method. Table 12 summarizes metrics for this comparison. Results show that the classi ers of the proposed method achieve high Precision, comparable to the supervised ones, but su er from low recall. The latter could be related to the strict sparsity objective of EDDP, as IBD exhibits signi cantly higher Redundancy. Combining classi ers with the same label improves recall, which indicates that di erent detectors capture di erent parts of the dataset, while supervised optimization of the classi cation bias further enhances F1 scores by relaxing sparsity.

In uence: In the concept in uence analysis, we aim to answer whether PCAVs, that correspond to directions of high interpretability due to learning them with supervision, are more in uential to the network predictions than signal vectors. Since Network Dissection can assign identical labels to multiple Encoding-Decoding Direction Pairs, direct comparison with Pattern-CAVs becomes less straightforward. We propose the following metric to help us draw some conclusions. Let $j \in \{0, 1, \dots, N_l - 1\}$ index EDDP's signal vectors sharing the same concept label l, and $S^l_{j,k}$ represent the RCAV sensitivity score of the network relative to the j-th signal vector of label l when predicting class k. Similarly, let $S^l_{P,k}$ denote the sensitivity score of the network relative to the Pattern-CAV for the same label and class. Inspired by RCAV, we regard signal vectors as noise vectors and compare the sensitivity of the network with respect to PCAVs and signal vectors that share the same label. We de ne a metric I^2, which, when above 0.5, indicates that PCAVs have more in uence than signal vectors on the network predictions at the statistical signi cance level of $\alpha = 0.05$ with

Table 12: Comparison of EDDP (with CFM) with supervised direction learning Zhou et al. (2018). Comparing between: a) standard EDDP, b) combined concept detectors (Linear-OR) c) EDDP with the thresholds of concept detectors learned with supervision in a post initial direction learning step (/w sup thres), and d) IBD: a set of classi ers learned in a supervised way. The network here is ResNet18 and I = 512.

		ResNet18 / Places365							
Method	Coverage "	Redundancy #	Precision "	Recall "	F1 "	AP "	S^1 "	S^2 "	mIoU "
IBD Zhou et al. (2018)	0.95	3.08	0.84 0.13	0.6 0.17	0.68 0.15	0.77 0.15	54.71	54.71	0.2
EDDP	0.86	1.28	0.81 0.24	0.2 0.16	0.29 0.19	0.47 0.2	50.29	36.99	0.1
EDDP - Linear OR	N/A	N/A	0.73 0.25	0.35 0.26	0.4 0.22	0.53 0.21	30.77	30.77	0.11
EDDP - /w sup thres	N/A	N/A	0.6 0.19	0.43 0.18	0.49 0.18	0.47 0.2	N/A	N/A	N/A

Bonferroni correction:

$$I^2 = E_{l;k}\left[1(p_{l;k} < \frac{}{N_l})\right]$$

$$p_{l;k} = \frac{1}{N_l}\sum_j 1\ jS^l_{j;k}j\ \ jS_{P;k}^{\ l}j$$

For RCAV = 5:0, I^2 is equal to 0:34, which is less than 0:5, indicating that the Pattern-CAVs are less in uential on the network predictions than signal vectors. This outcome could also be linked to the discussion in Section 7.5.6. It appears that less interpretable directions (in this case, EDDP's) have more in uence on the network predictions compared to more interpretable ones.

7.8 Application: Global Model Explanations via Concept Sensitivity Testing

Figure 15: Concept In uence Diagram for ResNet18 trained on Places365. The model is sensitive to the depicted concepts with an absolute score above 0.99. (We use RCAV to quantify the sensitivity, and re-scale the score to $[\ 1; 1]$) Positive in uencing and negative in uencing concepts are provided. The number of concepts have been limited to 10. When concepts appear more than once, they correspond to di erent signal directions (as labeling the classi ers with NetDissect may assign the same concept name to more than one directions.) Here we report results for EDDP-C and I = 512.

Figure 16: Concept In uence Diagram for ResNet50 trained on Moments in Time (MiT). The model is sensitive to the depicted concepts with an absolute score above 0.99. (We use RCAV to quantify the sensitivity, and re-scale the score to $[\ 1; 1]$) Positive in uencing and negative in uencing concepts are provided. The number of concepts have been limited to 10. When concepts appear more than once, they correspond to di erent signal directions (as labeling the classi ers with NetDissect may assign the same concept name to more than one directions.) Here we report results for EDDP-C and I = 2048.

Figures 15,40,41,42 depict concrete examples of how each concept's signal direction impacts the ResNet18's class predictions. Figures 16, 43 and 44 depict similar examples for ResNet50. Concepts appearing more than once correspond to di erent directions that have been attributed the same label. Seemingly irrelevant concepts with positive in uence may have three possible explanations: a) the network has some sensitivity to those concepts (as ResNet18's top1 accuracy is 56.51% and ResNet50's 28.4%) b) their impact might be