WHEN ARE CONCEPT BOTTLENECK MODEL EXPLANATIONS FAITHFUL AND COMPACT?

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept bottleneck models (CBMs) are neural classifiers that allow to explain their decisions via high-level concepts, potentially enabling understanding, steering and debugging. However, their explanations are often derived *heuristically*. Building on *formal* explainability – we argue they should also be faithful, *i.e.*, not misreport which concepts actually matter. We show that, for widespread CBM architectures, including recent VLM-based variants, faithful explanations must include all concepts in the bottleneck, compromising interpretability when this is large. This result applies to *both* heuristic and faithful-by-construction formal explanations. To encourage the existence of compact faithful explanations, we suggest *i*) modeling concepts *probabilistically* as binary or categorical random variables (rather than logits), and *ii*) employing per-concept training-time sparsification via *group lasso* (rather than regular elastic net). We also extend algorithms from formal explainability to CBMs, and show they outperform natural heuristics in terms of guarantees and explanation size. Overall, our work warns against naïve interpretability claims and provides formal conditions and practical strategies for ensuring CBMs are as interpretable as advertised.

1 INTRODUCTION

Concept bottleneck models (CBMs) (Koh et al., 2020; Ji et al., 2025; Knab et al., 2026) are "explainable-by-design" neural classifiers that predict high-level concepts (*e.g.*, objects present in an input image) and use these to infer a label (*e.g.*, the image's class), see Fig. 1 (left). By enabling stakeholders to inspect the set of concept activations – the titular *bottleneck* – CBMs aim to support interpretation (Bogani et al., 2026), steering (Shin et al., 2023), and interactive debugging (Stammer et al., 2021). Yet, reliable interaction requires CBMs to produce explanations that are both ***compact*** (Ramaswamy et al., 2023; Schrodi et al., 2025), thus supporting understanding, and ***faithful*** to the CBM's decision process, so as not to mislead stakeholders (Marques-Silva, 2023; Teso et al., 2023).

Neither property is guaranteed in practice. On the one hand, bottlenecks can include hundreds or even thousands (Yang et al., 2023) of activations, well beyond what stakeholders can grasp. The go-to solution is to compute a small explanation – that is, a subset of the bottleneck – $\mathcal{E}$ using heuristic algorithms, *e.g.*, by taking the k highest activations, but see Fig. 1 and Section 3 for alternatives (Yuksekgonul et al., 2023; Yang et al., 2023; Oikarinen et al., 2023; Srivastava et al., 2024).

On the other, as we will show, *heuristic explanations can be unfaithful*, *i.e.*, they may misreport which concepts do vs. do not determine the prediction. Specifically, a subset $\mathcal{E}$ that supports different predictions is insufficient to justify either (Marques-Silva, 2023; Darwiche, 2023; Soria et al., 2026). To see why, consider the heuristic explanation $\mathcal{E}$ in Fig. 1 (right), computed on the `CUB` bird classification dataset (Wah et al., 2011). It supports predictions for two different classes, `Vesper Sparrow` and `Black-billed Cuckoo`. Since the activations in $\mathcal{E}$ are identical for both predictions, these cannot be determined by $\mathcal{E}$ alone: they must depend on activations *outside* $\mathcal{E}$.

We seek to understand when compact faithful explanations can be obtained. As a first contribution, we demonstrate that this is not always possible: unless a CBM is sufficiently constrained, ***faithful explanations must include all concepts in the bottleneck***, no matter what algorithm is used to compute them. We show that this depends on how a CBM constrains the concept activations and the inference layer mapping concepts to labels and that many existing CBM architectures fail

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

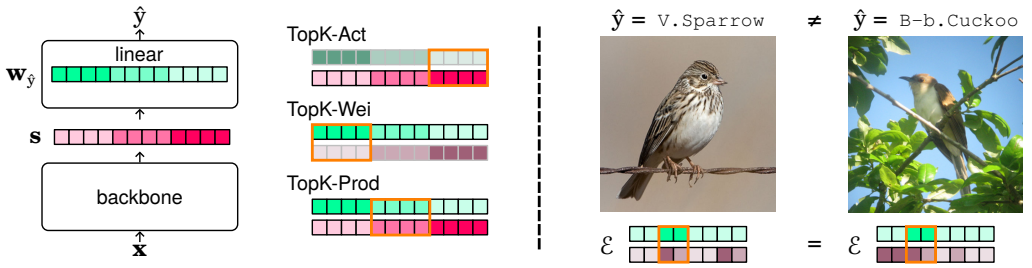


Figure 1: **Current heuristic explanation for CBMs can be unfaithful** as shown for a CBM combining a neural *backbone* generating concept activations $\mathbf{s} \in \mathcal{S}$ and a linear *inference layer* with per-class weights $\mathbf{w}_y$, with $y \in [m]$. **Left:** Standard heuristics compute explanations $\mathcal{E}$ by looking at the magnitude of activations, weights, or both, cf. Eq. (2). **Right:** A heuristic explanation $\mathcal{E}$ that covers the $k = 2$ top weights: it justifies two decisions on the ground that the same two activations attain the same values. Yet, the decisions differ: the CBM must be looking at activations outside of $\mathcal{E}$ to produce them. Hence, $\mathcal{E}$ is insufficient to fully justify either decision.

on both fronts. As a second contribution, we discuss two simple but effective *strategies that encourage CBMs to admit compact faithful explanations*, namely: *i*) modeling concept activations as probability distributions and not logits; and *ii*) applying structured, rather than unstructured, sparsity to the inference layer via group lasso. As a third contribution, we propose *replacing heuristics with algorithms from formal explainability* (Marques-Silva, 2023; Darwiche, 2023), which ensure faithfulness, and introduce `EA+`, an exact algorithm geared for compactness. Our experiments in Section 5 highlight how our modeling strategies enable finding more compact faithful explanations for regular and VLM-based CBMs (Oikarinen et al., 2023; Yang et al., 2023; Debole et al., 2026a) and how `EA+` outperforms heuristic alternatives while remaining computationally affordable.

Contributions. We: *i*) Study the faithfulness of CBM explanations, proving that, for many regular and VLM-based CBMs, faithful explanations cannot be compact (Section 4); *ii*) From an algorithmic perspective, extend exact explainers from FXAI that ensure faithfulness to favor compactness (Section 3); *iii*) From a modeling perspective, propose two simple but effective strategies that constrain CBMs to allow more compact faithful explanations (Section 4.2); *iv*) Present an extensive experimental evaluation showcasing the underlying issues and the benefits of our contributions (Section 5).

2 Concept Bottleneck Models

Notation. We denote scalar constants in lower-case (x), vector constants in bold ($\mathbf{x}$), scalar and vector variables in upper-case (X, $\mathbf{X}$), and sets in upper-case italics ($\mathcal{X}$). We abbreviate $\{1, \ldots, n\}$ to $[n]$ and, given a vector $\mathbf{x} \in \mathbb{R}^n$, we write $\mathbf{x}_\mathcal{I}$ for its restriction to the indices in $\mathcal{I} \subseteq [n]$.

Concept bottleneck models. CBMs (Koh et al., 2020) are neural classifiers that combine a *backbone* $g : \mathcal{X} \to \mathcal{S}$ mapping inputs (*e.g.*, images) $\mathbf{x} \in \mathcal{X}$ to a bottleneck of b concept *activations* $\mathbf{s} \in \mathcal{S}$, and an *inference layer* $f : \mathcal{S} \to \mathbb{R}^m$ mapping activations to m per-class logits; see Fig. 1 (left). The prediction $\hat{y} \in [m]$ is the highest-scoring class. Concept activations $\mathbf{s}$ are traditionally modelled as *independent logits* (Koh et al., 2020; Shin et al., 2023; Park et al., 2025), *i.e.*, $\mathcal{S} \equiv \mathbb{R}^b$, and only translated into probabilities for computing the loss. Most commonly, the inference layer is a *linear model* of the form $f(\mathbf{s}) = W\mathbf{s} + \mathbf{w}_0$, where $W \in \mathbb{R}^{m \times b}$ and $\mathbf{w}_0 \in \mathbb{R}^m$ are parameters. CBMs define a loss for labels and one for concepts:

$$\ell^Y(f, g) := \widehat{\mathbb{E}}_{(\mathbf{x}_i, y_i)}\big[-\log p(y_i \mid \mathbf{x}_i)\big], \quad \ell^C(f, g) := \widehat{\mathbb{E}}_{(\mathbf{x}_i, \mathbf{c}_i)}\big[-\tfrac{1}{b}\textstyle\sum_{j \in [b]} \log p(c_{ij} \mid \mathbf{x}_i)\big]. \quad (1)$$

Here, $y_i \in [m]$ are task labels and $\mathbf{c}_i \in \{0, 1\}^b$ concept labels; the distributions are $p(y \mid \mathbf{x}) = \mathrm{softmax}((f \circ g)(\mathbf{x}))_y$ and $p(c_j \mid \mathbf{x}) = \mathrm{sigmoid}(g(\mathbf{x})_j)$; and the expectations are estimated on a densely annotated training set $\{(\mathbf{x}_i, \mathbf{c}_i, y_i)\}$. The concept loss aims to align concepts to human semantics, but see Section 6 for a discussion of failure cases. The exact loss depends on whether the model is trained *independently* (each module fit on corresponding labels), *sequentially* (backbone fit on concept labels, inference layer on concept predictions), or *jointly* (end-to-end with a convex combination of the two losses).

This backbone-plus-linear-classifier setup acts as a blueprint for most CBM variants – see (Ji et al., 2025; Knab et al., 2026) for overviews – and for other concept-based architectures, like self-explainable (Alvarez Melis & Jaakkola, 2018) and prototype-based (Chen et al., 2019) neural nets.

VLM-aided CBMs. Concept supervision is key for ensuring learned concepts are human-aligned (Bortolotti et al., 2025; Fokkema et al., 2025), yet seldom available. For datasets lacking it, "VLM-CBMs" bypass this issue by asking an LLM to generate a concept vocabulary (*i.e.*, textual descriptions of task-relevant concepts) and then obtaining weak concept supervision from a Vision-Language Model (VLM) like CLIP (Radford et al., 2021), typically by measuring the similarity between concept descriptions and images in the VLM's embedding space (Yuksekgonul et al., 2023; Oikarinen et al., 2023; Yang et al., 2023; Tan et al., 2024a; Kazmierczak et al., 2026; Debole et al., 2026a). These activations are used either to train a backbone or employed directly for inference.

3 Compactness and Faithfulness

By decomposing predictions into higher-level concepts, CBMs aim to facilitate mental simulation and reliability assessment (Rudin, 2019; Narayanan et al., 2018; Bogani et al., 2026), human intervention (Shin et al., 2023; Espinosa Zarlenga et al., 2023), and interactive debugging (Stammer et al., 2021; Bontempelli et al., 2023). Reliable interaction, however, requires CBMs to supply stakeholders with concept-level explanations that are both *compact* – so as to enable understanding – and *faithful* to their decision process – so as not to mislead stakeholders.

Compactness. Taking the *whole* bottleneck to serve as explanation is rarely useful, as CBM vocabularies can comprise hundreds of entries (Lage et al., 2019; Barker et al., 2023; Schrodi et al., 2025), well beyond what stakeholders can grasp. E.g., the default vocabulary of the `CUB` bird classification task (Wah et al. (2011), illustrated in Fig. 1, right) comprises 312 activations, and LLM-generated vocabularies for the same task count up to 10,000 (Yang et al., 2023).

Practitioners often construct a smaller explanation by extracting a subset $\mathcal{E} \subset [b]$ of concept activations. Given a prediction $(\mathbf{s}, \hat{y})$ and a target size $k < b$, three well-known *heuristic algorithms* for doing so (shown in Fig. 1, middle) are picking the k activations[1] with the highest value (Chen et al., 2019; Ley et al., 2026), the largest-magnitude weights (Yuksekgonul et al., 2023; Yang et al., 2023), or a combination thereof (Oikarinen et al., 2023; Srivastava et al., 2024):

$$\mathcal{E} = \text{argtop}_k\{s_i\}, \qquad \mathcal{E} = \text{argtop}_k\{|w_{\hat{y},i}|\}, \qquad \mathcal{E} = \text{argtop}_k\{|s_i \cdot w_{\hat{y},i}|\}. \tag{2}$$

One exception are Selective CBMs (Schrodi et al., 2025), which construct $\mathcal{E}$ via gating. These heuristics are sometimes paired with *training-time sparsification* penalties encouraging CBM predictions to rely only on few concepts. We will detail these and Selective CBMs in Section 4.2.

Faithfulness. Another essential requirement is that $\mathcal{E}$ covers all and only those concepts that determine the target prediction but not others (Marques-Silva, 2023; Darwiche, 2023). To see why, consider the example in Fig. 1 (right): despite covering the $k = 2$ highest-weight activations (as per Eq. (2)), $\mathcal{E}$ supports predictions for two different classes, `Vesper Sparrow` and `Black-billed Cuckoo`. Since the same activations within $\mathcal{E}$ are compatible with different predictions, the predicted class cannot be determined by $\mathcal{E}$ alone, but it must depend on activations *outside* $\mathcal{E}$. In other words, $\mathcal{E}$ is insufficient to fully justify either decision. Unfaithful explanations $\mathcal{E}$ can mislead users, compromising model assessment and reliable interaction.

As illustrated by the example above, the core requirement for a faithful explanation is that $\mathcal{E}$ cannot simultaneously back up predictions for different classes (Shih et al., 2018; Ignatiev, 2020). Surprisingly, this aspect has been off the radar in the CBM literature, and the go-to heuristics from Eq. (2) have been implicitly assumed to be faithful. We show this is not the case. Following the FXAI literature, explanations satisfying this requirement are called *abductive explanations*[2] (AXps) (Marques-Silva, 2023), which we extend to CBMs as follows:

Definition 1. *Given a CBM (g, f), a prediction $(\mathbf{s}, \hat{y})$, and a set of attainable concept activation vectors $\mathcal{S}$, then $\mathcal{E} \subseteq [b]$ is said to be a* weak AXp *for $\hat{y}$ with respect to $\mathcal{S}$ if and only if*

$$\forall \mathbf{s}' \in \mathcal{S} \,.\, \left(\mathbf{s}'_{\mathcal{E}} = \mathbf{s}_{\mathcal{E}}\right) \implies \left(\hat{y} \in \text{argmax}_{y \in [m]} f(\mathbf{s}')_y\right). \tag{3}$$

[1] Alternatively, one can fix a weight, activation, or score threshold $\tau \geq 0$. Our results extend to this case.

[2] Aka *sufficient reasons* (Shih et al., 2018) and *prime implicant explanations* (Ignatiev et al., 2019).

A weak AXp *that is also subset-minimal (s.t. no proper subset $\mathcal{E}' \subset \mathcal{E}$ is a weak* AXp*), is an* AXp.

The definition requires *sufficiency* – an adversary cannot construct any counterexample $\mathbf{s}' \in \mathcal{S}$ that matches $\mathbf{s}$ on all concepts in $\mathcal{E}$ and yet entails a different class[3] – and *subset-minimality* – all selected concepts should be needed for this to hold. As shown in Fig. 1 (right) and by our experiments in Section 5.3, the heuristics in Eq. (2) provide no such guarantees for commonly used values of k.

Enforcing faithfulness. If guarantees are desired, a better option is to resort to *exact algorithms* for computing AXps. Concretely, we build on the simple but flexible Algorithm 1 of Marques-Silva (2023), denoted EA (Exact Algorithm). Starting from the full bottleneck, $\mathcal{E} = [b]$, it iteratively removes unneeded activations following an arbitrary order (*e.g.*, activation index), invoking an *oracle* to check whether the candidate $\mathcal{E}$ admits a counterexample $\mathbf{s}' \in \mathcal{S}$. Doing so can be computationally expensive – in fact, NP-hard – in general (Shih et al., 2018; Ignatiev et al., 2019). However, the decision function of CBMs is linear and $\mathcal{S}$ is a conjunction of linear inequalities; therefore, the oracle boils down to a linear program that can be solved efficiently (see Section A.1).

We propose a variant of EA biased toward more compact explanations. Our algorithm, denoted EA+, grows a candidate $\mathcal{E} = \varnothing$ into a weak AXp by adding groups of activations following a heuristic-informed and shrinks the result until it is subset minimal. This is akin to Algorithm 1 of Soria et al. (2026), except EA+ *i*) computes three AXps – one for each heuristic in Eq. (2) – and picks the smallest, and *ii*) employs binary search during the growth stage, and as such requires only a logarithmic number of oracle calls. As shown in Section D.1, EA+'s explanations are always AXps, whereas heuristics explanations often are not, while remaining computationally affordable in practice. Pseudo-code for all algorithms is listed and discussed in Section A.

4 The Trade-off Between Faithfulness and Compactness

Our goal is to construct explanations $\mathcal{E}$ that are both *compact* and *faithful*. These properties, however, are at odds: any (weak) AXp must pin all activations that can be (even adversarially) manipulated to yield a different prediction.[4] Hence, the more leeway a CBM offers to construct counterexamples, the larger (weak) AXps have to be. We proceed to make this argument exact. The next result shows how two architectural properties – (**P1**) the structure of the activation space $\mathcal{S}$, and (**P2**) the structure of the inference layer's weights W – determine whether a given $\mathcal{E} \subset [b]$ can be a weak AXp. We begin from *unbounded* activations $\mathcal{S} = (-\infty, \infty)^b$ and analyze the other cases to Section B.

Proposition 1 (Informal)**.** *Fix any CBM prediction $(\mathbf{s}, \hat{y}) \in \mathcal{S} \times [m]$ and explanation $\mathcal{E} \subset [b]$, and let $\delta\mathbf{w}^{(j)} := \mathbf{w}_j - \mathbf{w}_{\hat{y}} \in \mathbb{R}^b$ be the delta between the weight vectors of classes j and $\hat{y}$. If (**P1**) $\mathcal{S} = (-\infty, \infty)^b$, and (**P2**) there exists an activation $i \in [b] \setminus \mathcal{E}$ and a class $j \neq \hat{y}$ s.t. $\delta w_i^{(j)} \neq 0$, then $\mathcal{E}$ is **not** a weak* AXp.

The proof can be found in Section B. Hence, if $\mathcal{S}$ is unbounded, fixing the prediction $\hat{y}$ requires $\mathcal{E}$ to fix *all* concept activations j that link it to any alternative y' via a non-zero weight difference $\delta w_i^{(j)}$. Together with our earlier observation that $\mathcal{E} = [b]$ is a (weak) AXp, this entails that for regular CBMs, which (**P1**) model activations as raw logits and (**P2**) employ dense linear layers, *no* AXp *smaller than the entire bottleneck exists at all*.[5] This affects both heuristic and exact algorithms.

Proposition 1 suggests – and Proposition 3 confirms – that **P1** and **P2** indeed control for the existence and size of (weak) AXps: intuitively, larger activation spaces $\mathcal{S}$ leave more room for manipulating concept activations not fixed by $\mathcal{E}$, while non-zero learned weights, especially if large, translate these into higher scores for non-predicted classes (see Section B.3 for a detailed discussion). Thus, appropriately constraining $\mathcal{S}$ and W can reduce the adversary's ability to construct working counterexamples, meaning (weak) AXps do not have to cover as many activations to safeguard the prediction.

[3] If a CBM predicts two classes when the same subset of activations attain exactly the same values, these classes cannot be faithfully explained. This is unlikely to happen for discriminatively trained models.

[4] E.g., a weak AXp can always be obtained by taking $\mathcal{E} = [b]$: since this fixes the whole bottleneck, there exists no counterexample. However, as we discussed, such an explanation is unusable if b is large.

[5] In the binary case ($m = 2$), the top layer has a single output (the score of the positive class) and a single weight vector $\mathbf{w} \in \mathbb{R}^b$. This is equivalent to a top layer with two outputs and symmetric per-class weight vectors, that is, $\mathbf{w}_0 = -\mathbf{w}_1$. Hence, the precondition $w_{\hat{y},j} - w_{y',j} \neq 0$ becomes $w_{\hat{y},j} \neq 0$: Proposition 1 triggers whenever an independent, unbounded activation c_j not in $\mathcal{E}$ is associated to a non-zero weight.

Table 1: **Representative CBMs** grouped by domain of concept activations $\mathcal{S}$ (**P1**) and structure of the inference layer's weights W (**P2**). Constraining $\mathcal{S}$ and W can make it more difficult to construct counterexamples that flip the prediction, allowing (weak) AXp to be smaller. Few CBMs employ probabilistic activations and none group lasso.

ACT. $\mathcal{S}$	WEI. W	REPRESENTATIVE MODELS (NOT EXHAUSTIVE)
$(-\infty, \infty)^b$	Dense	CBM (Koh et al., 2020), ProbCBM (Kim et al., 2023), MCBM (Almudévar et al., 2026)
$(-\infty, \infty)^b$	Elastic Net	LF-CBM (Oikarinen et al., 2023), VLG-CBM (Srivastava et al., 2024)
$[\alpha, \infty)^b, \alpha \geq 0$	Elastic Net	Selective CBM (Schrodi et al., 2025)
$[-1, 1]^b$	Dense	LaBo (Yang et al., 2023)
$\subseteq [0, 1]^b$	Dense	CB2M (Steinmann et al., 2024), GlanceNets (Marconato et al., 2022)

4.1 EXISTING CBMS VS. PROPOSITION 1

Next, we discuss existing CBMs in light of **P1** and **P2**, summarizing our findings in Table 1.

Concept activation space. Standard CBMs (Koh et al., 2020; Shin et al., 2023; Park et al., 2025) and extensions (Sawada & Nakamura, 2022; Wu et al., 2022; Huang et al., 2024; Almudévar et al., 2026) employ independent, unbounded activations, and thus trigger Proposition 1. Despite the name, probabilistic CBMs do the same (Kim et al., 2023). VLM-CBMs with a backbone (Oikarinen et al., 2023; Srivastava et al., 2024) do supervise it with bounded concept annotations, yet feed the inference layer unbounded activations. Post-hoc CBMs (Yuksekgonul et al., 2023) lack a backbone, but they employ unbounded dot products as activations. All these models leave the adversary unconstrained. Selective CBMs (Schrodi et al., 2025) pass activations through an input-dependent gating function $\max(0, \mathbf{s} - \mathbf{o})$ with a learnable bias $\mathbf{o}$.[6] While this prevents Proposition 1 from applying, it allows pushing activations not in $\mathcal{E}$ to $+\infty$, which can flip the label whenever these are associated to positive (rather than non-zero) weight differences $\delta w_i^{(j)}$ (see Proposition 3 for a formal statement); as we show in Section D.5, this inflates faithful explanations in practice. LaBo (Yang et al., 2023) fares best in that its activations are cosine similarities living in $\mathcal{S} = [-1, 1]^b$, yet we show in Section 5.2 this does not guarantee (weak) AXps are compact unless the weights of the inference layer are also appropriately constrained.

Inference layer weights. Standard CBMs do not regularize W and as such do not encourage learning non-zero differences $\delta\mathbf{w}^{(j)}$. Recent alternatives – like Selective CBMs (Schrodi et al., 2025) and most VLM-CBMs (Yuksekgonul et al., 2023; Oikarinen et al., 2023; Srivastava et al., 2024) – sparsify W through an *elastic net* penalty (Zou & Hastie, 2005), thus encouraging predictions to depend on few concepts. This penalty takes the form $\alpha\|W\|_1 + (1-\alpha)\|W\|_2^2/2$, where α mediates between the L_1 and L_2 terms. While the elastic net encourages *unstructured* sparsity, it still allows learning *pairs* of entries satisfying $\delta\mathbf{w}^{(j)} \neq \mathbf{0}$, and, as we show in Section 5.2, it struggles to produce compact faithful explanations for VLM-CBMs.

4.2 BEYOND PROPOSITION 1 WITH PROBABILISTIC CONCEPTS AND GROUP LASSO

Probabilistic modeling of concept activations. We propose to constrain $\mathcal{S}$ by *modeling concepts probabilistically*, *i.e.*, by passing activations of binary (resp. categorical) concepts through sigmoid (resp. softmax) activations. Besides ensuring activations are bounded to $[0, 1]^b$, doing so allows exploiting mutual exclusivity of categorical concepts. To see why this matters, imagine encoding a categorical "color" concept with three values – red, green and blue – as three softmax outputs $s_{\mathtt{r}}$, $s_{\mathtt{g}}$, $s_{\mathtt{b}}$. Critically, $s_{\mathtt{r}} + s_{\mathtt{g}} + s_{\mathtt{b}} = 1$. Suppose red is predicted with high probability, say, $s_{\mathtt{r}} = 0.95$. If $\mathcal{E}$ covers $s_{\mathtt{r}}$, the adversary is left little room to construct a counterexample $\mathbf{s}'$: at most, it can shift the residual 0.05 mass between $s_{\mathtt{g}}$ and $s_{\mathtt{b}}$, which is unlikely to flip the prediction. Formally, this shrinks $\mathcal{S}$ from $[0, 1]^b$ to a product of per-concept simplices: the two only match if all concepts are binary.

Structured weight sparsity. We suggest to constrain the weights W via *group lasso* (Yuan & Lin, 2006), treating each activation as a separate group. This penalty takes the form $\sum_i \|\mathbf{w}_i\|_2$, where i iterates over activations, so as to actively discourage each of them from contributing to any class. In turn, this encourages all differences $w_{y',j'} - w_{\hat{y},j'}$ to be exactly zero.

[6] The JumpReLU/TopK-based variants (Schrodi et al., 2025) behave analogously.

Benefits & Limitations. As we will show empirically in Section 5, both proposed strategies are effective at shrinking the size of (weak) AXps. Probabilistic modeling of concepts is not new (see *e.g.*, (Marconato et al., 2022; Steinmann et al., 2024; Tan et al., 2024b), but it has never been linked to the compactness of faithful explanations. In contrast, despite being a widely known technique, no existing CBM architecture employs group lasso. As we will see, probabilistic modeling does not come at any accuracy cost, while excessive sparsification *can* lower downstream accuracy (see Sections 5.2 and 5.3) and as such has to be applied with care. We stress, however, that even our strategies, while helpful, cannot ensure large bottlenecks always admit *small* AXps. We will briefly outline other potentially promising remedies in Section 7.

5 Experiments

We address the following research questions: **Q1**: Does modeling concepts probabilistically enable more compact AXps? **Q2**: Does group lasso shrink AXps further? **Q3**: How does `EA+` fare compared to the heuristic algorithms in Eq. (2)? Due to space constraints, we leave some experimental details to Section C and additional results to Section D.

Datasets. We evaluate four CBM datasets spanning different number of classes m, concept activations b, and ratio of binary vs. categorical concepts, cf. Table 2. `CelebA` (Liu et al., 2015) comprises 203K face images annotated with 40 binary concepts, split into 163K training / 20K validation / 20K test images. The goal is to predict `sex` from the remaining 39 concepts. `Derm7pt` (Kawahara et al., 2018) includes 1K dermoscopic skin-lesion cases annotated with diagnostic criteria, one-hot encoded into 28 activations, and a 5-class diagnosis label obtained by grouping 20 raw diagnoses. `WBCAtt` (Tsutsui et al., 2023) annotates 10K white blood cell microscopy images with 11 categorical morphological attributes, one-hot encoded into 31 activations, and a 5-class cell-type label. `CUB` (Wah et al., 2011) involves classifying bird images into one of 200 classes using 28 categorical concepts (multi-hot encoded into 312 activations). It comprises 5K training and 5K test images; we carve about 10% of the training set for validation.

Table 2: **Datasets.** Cat. is the # of categorical concepts.

Dataset	m	b	Cat.	Train	Val	Test
`CelebA`	2	39	0	162,770	19,867	19,962
`Derm7pt`	5	28	7	413	203	395
`WBCAtt`	5	31	11	6,169	1,030	3,099
`CUB`	200	312	28	5,394	600	5,794

Models. For all models, the backbone is a frozen pre-trained image encoder – OpenCLIP's `ViT-L-14-quickgelu` (Ilharco et al., 2021) – followed by a linear concept probe. The inference layer is a logistic regressor, either dense or sparsified via elastic net (`enet`) or group lasso (`glasso`). For regular CBMs, we take the concept vocabularies and annotations from the dataset and evaluate logit ($\mathcal{S} = (-\infty, +\infty)^b$), ReLU ($\mathcal{S} = [0, +\infty)^b$), and probabilistic ($\mathcal{S} \subseteq [0, 1]^b$) concept activations. For the latter, we employ softmax activations for categorical concepts and sigmoids for binary ones, as per Section 4.2. We resort to sequential training, as it achieves competitive performance while reducing the amount of label-specific information affecting the concepts. For VLM-CBMs, we employ LLM-generated vocabularies from `LaBo` (Yang et al., 2023), `LF-CBM` (Oikarinen et al., 2023), and `VLG-CBM` (Srivastava et al., 2024) and compute concept activations as cosine similarities in CLIP's embedding space, as customary (Debole et al., 2026a). Following Section 4.1, we ensure $\mathcal{S} = [-1, 1]^b$ by employing the obtained activations for inference, like `LaBo`.[7] Due to space constraints, we defer our analysis of Selective CBMs to Section D.5.

Metrics. For each model, we report the test set label and concept F_1 score (denoted F_1^Y and F_1^C). The latter is micro-averaged over concept activations when these are independent (*i.e.*, logits and ReLUs) and over concepts otherwise (*i.e.*, for probabilities). For each algorithm, we explain a random subset of 1K test instances; for `Derm7pt` we explain all 395 test points; for the `LaBo` `CUB` vocabulary, which has 10K entries, we explain 100 instances. For each model, *we test faithfulness following Definition 1 using the model's declared activation space* $\mathcal{S}$; as we discuss in Section C.1, this guarantees our weak AXps are robust to all possible counterexamples, regardless of whether are produced by the backbone or by (potentially malicious) human interventions.

[7] The original `LF-CBM` and `VLG-CBM` employ a different VLM and a separate backbone; this explains the accuracy drop in our results compared to the original papers. This difference is irrelevant for our purposes.

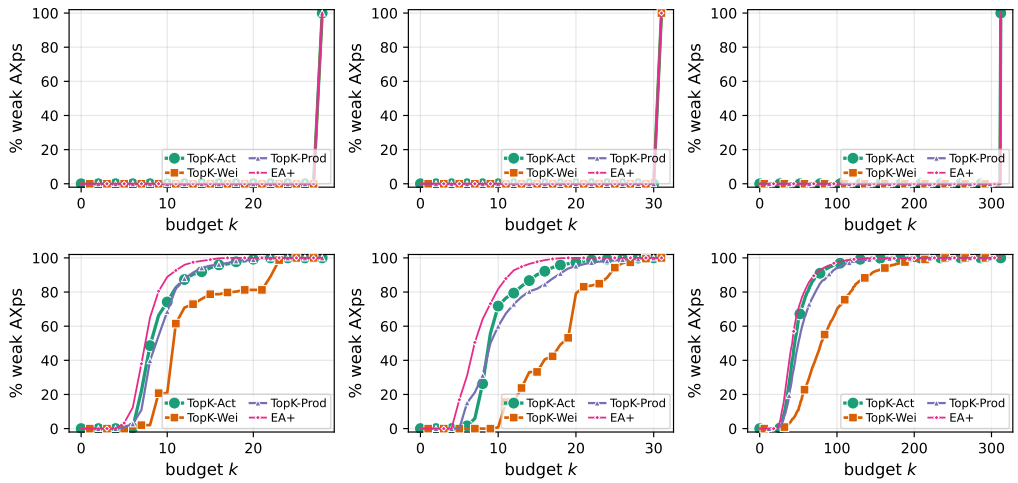


Figure 2: **Modeling concepts as probabilities shrinks** AXps on `Derm7pt` (left), `WBCAtt` (middle), and `CUB` (right). Top row: logits. Bottom row: probabilities. `CelebA` is left to Section D.2.

5.1 Q1: Probabilities Shrink Faithful Explanations At No Accuracy Cost

We compute *faithfulness-compactness curves* (FCCs) plotting the cumulative percentage of explained instances admitting a (weak) AXp within a specified size budget k: the faster they grow, the more instances admit a smaller faithful explanation. In Fig. 2, we report FCCs for dense CBMs: logit activations above and probabilities below. Colors represent different algorithms: the `TopK` heuristics in Eq. (2) (`Act` in **green**, `Wei` in **orange**, `Prod` in **violet**) and `EA+` (**pink**).

Logits are pathological. For all algorithms, the curves for logit activations are squashed to zero until the explanation size budget reaches the bottleneck size b. This empirically confirms Proposition 1: logits leave the adversary unconstrained, meaning that – since the inference layer is dense – no (weak) AXp smaller than the entire bottleneck exists at all.

Table 3: **Probabilities are competitive with logits** in terms of F_1 score.

	F_1^C		F_1^Y	
Dataset	prob	logit	prob	logit
`CelebA`	0.760	0.760	0.989	0.991
`Derm7pt`	0.676	0.670	0.653	0.666
`WBCAtt`	0.877	0.872	0.943	0.948
`CUB`	0.796	0.781	0.817	0.832

Probabilities shrink faithful explanations of all algorithms without hindering accuracy. For probabilistic activations, the FCCs grow much quicker, meaning several explained instances admit smaller (weak) AXps. Focusing on `EA+`, the FCC shows that about 80% instances (Y-axis) admit a faithful explanation of size ≤ 9 in `Derm7pt` (down from 28 for logits), ≤ 10 in `WBCAtt` (31 for logits), and ≤ 60 in `CUB` (312 for logits). The results for `CelebA` in Section D.2 are analogous. Comparing the FCCs of the heuristics in the top and bottom rows shows they also require a smaller budget k to find weak AXps: `TopK-Act` and `TopK-Prod` fare best, *i.e.*, explain compactly and faithfully almost as many instances as `EA+`, while `TopK-Wei` lags behind. Overall, these results confirm that modeling concepts probabilistically is effective at preventing counterexamples. Crucially, these benefits come at no cost as CBMs with logit and probabilistic activations perform equally well: Table 3 shows that they attain almost identical F_1^Y and F_1^C, with the worst-case gap being 2 percentage points (pp).

5.2 Q2: Group Lasso Outperforms Elastic Net

Next, we compare `enet` and `glasso` on CBMs with probabilistic activations and VLM-CBMs with cosine similarity activations. We focus here on the average size of the (weak) AXps found by the various algorithms, and report the FCCs for all models and datasets in Sections D.3 and D.4. Note that cranking up sparsity strength λ helps finding faithful explanations (per **P2**) but also sinks performance. To ensure a fair comparison, for each sparse CBM we maximize λ under the constraint that its F_1^Y is at most 10% worse than the dense CBM's on the validation split, as shown in Table 5. We fix `enet`'s α to 0.99, Yuksekgonul et al. (2023) and Oikarinen et al. (2023)'s default.

Table 4: **`glasso` consistently yields smaller faithful explanations than `enet`**. Explanation size is reported as a percentage (*i.e.*, normalized by bottleneck size b) for concision. **Bold** indicates best in column and red no shrinkage w.r.t. the dense baseline.

	Derm7pt ($b = 28$)				WBCAtt ($b = 31$)				CUB ($b = 312$)			
	ACT	WEI	PROD	EA+	ACT	WEI	PROD	EA+	ACT	WEI	PROD	EA+
dense	34.1	47.3	35.1	29.6	34.0	57.6	35.6	26.3	16.6	28.6	18.5	15.4
enet	27.7	36.9	36.9	9.5	29.9	31.0	30.9	**7.3**	**13.1**	87.0	87.0	10.2
glasso	**27.4**	**8.1**	**7.8**	**7.3**	**28.1**	**11.7**	**13.6**	8.1	14.3	**9.8**	**9.4**	**7.7**
	LF-CBM ($b = 202$)				VLG-CBM ($b = 419$)				LaBo ($b = 10{,}000$)			
	ACT	WEI	PROD	EA+	ACT	WEI	PROD	EA+	ACT	WEI	PROD	EA+
dense	100.0	99.9	99.9	99.5	99.9	99.8	99.8	99.4	99.8	99.6	99.6	99.6
enet	100.0	99.9	99.9	82.9	100.0	99.9	99.9	68.7	99.9	99.1	99.1	46.9
glasso	99.7	**19.3**	**19.4**	**17.5**	99.8	**17.1**	**17.1**	**15.9**	99.9	**6.6**	**6.7**	**6.5**

`enet` only partially works. For CBMs (Table 4, top), in most cases `enet` shrinks explanation sizes, but in three cases (in red) it actively regresses: barely on `Derm7pt` (+1.8pp for `TopK-Prod`) and substantially on `CUB` (+58.4pp for `TopK-Wei` and +68.5pp for `TopK-Prod`). This is because at $\alpha = 0.99$, which favors L_1 over L_2, it tends to learn larger non-zero weights, which allow the adversary to flip the prediction with smaller activations deltas (as larger weights help convert these into comparatively larger changes to the class scores). `enet` also fares poorly with VLM-CBMs, which have less constrained concept activations: it often fails to shrink explanations (see the red entries), except for `EA+` (`LF-CBM`: −16.6pp; `VLG-CBM`: −30.7pp; `LaBo`: −52.7pp).

Table 5: **`glasso` shrinks faithful explanations at less than 10% F_1^Y cost**. VLM-CBMs have no F_1^C as they have no ground-truth annotations.

	F_1^C	F_1^Y		
DATASET		DENSE	enet	glasso
CUB	0.796	0.817	0.759	0.756
Derm7pt	0.676	0.653	0.648	0.638
WBCAtt	0.877	0.943	0.916	0.927
LF-CBM	–	0.545	0.589	0.564
VLG-CBM	–	0.626	0.632	0.576
LaBo	–	0.835	0.753	0.766

`glasso` performs better and is key for VLM-CBMs. `glasso` always yields smaller faithful explanations compared to the dense baseline. E.g., for `EA+`, the improvement amounts to `Derm7pt`: −22.3pp; `WBCAtt`: −18.3pp; `CUB`: −7.7pp. It only loses to `enet` by a small margin in two cases: `WBCAtt`/`EA+` (explanation size 7.3% vs 8.1%) and `CUB`/`TopK-Act` (13.1% vs 14.3%). Moreover, Table 4 (bottom) shows that `glasso` is essential for VLM-CBMs, where it substantially outperforms `enet` (except for `TopK-Act`, which misbehaves in all cases): with `glasso`, `EA+` explanations take 17.5% of `LF-CBM`'s bottleneck (whereas `enet` requires 82.9% of it), 15.9% of `VLG-CBM`'s (versus 68.7%), and only 6.5% of `LaBo`'s (versus 46.9%).

5.3 Q3: EA+ FINDS SMALLER FAITHFUL EXPLANATIONS

Finally, we assess how the various algorithms fare in recovering compact (weak) AXps, focusing on properly constrained CBMs to enable meaningful comparison. Additionally, we report an in-depth analysis of runtimes, number of oracle calls, and subset-minimality in Section D.1.

`EA+` finds AXps at a lower budget k. While `EA+` finds k automatically, the `TopK` heuristics take it as input: if too small, they *cannot* find any weak AXp. This is what the FCCs in Fig. 2 (bottom row) show for CBMs with probabilistic concepts: *setting k too low yields no weak AXp at all*. Moreover, compared to `EA+`, heuristics need a larger k to find a weak AXp. This is expected: `EA+` returns the smallest of three AXps, each computed by refining a weak AXp obtained by ranking activations using weights, activations, or their product, cf. Section A. Among heuristics, `TopK-Act` and `TopK-Prod` work best. This is due to modeling categorical concepts as probabilities: as discussed in Section 4.2, they pin the largest activations first, constraining the other activations via mutual exclusivity in turn. This advantage disappears when all concepts are binary, as shown on `CelebA` in Section D.2 and by VLM-CBMs. `EA+` also yields smaller weak AXps for VLM-CBMs, as shown in Table 4. The gap is substantial for `enet`-sparsified models: the best heuristic yields explanations with average size $> 99\%$ of the bottleneck across vocabularies, while `EA+` attains 82.9% (`LF-CBM`), 68.7% (`VLG-CBM`) and 46.9% (`LaBo`). On `glasso`, `EA+` still wins but with a smaller margin, attaining 17.5%, 15.9% and 6.5% versus 19.3%, 17.1% and 6.6% for the runner-up.

6 Related Work

Formal XAI sets itself apart from classic XAI (Guidotti et al., 2018) by seeking explanations with formal guarantees (Marques-Silva, 2023; Darwiche, 2023). Most works focus on discrete features, and as such do not apply to CBMs with continuous concepts, although a handful of CBM variants do employ discrete concepts, *e.g.*, (Havasi et al., 2022; Vandenhirtz et al., 2024; Knab et al., 2026); we plan to investigate their faithfulness in future work. We build on the well-known EA algorithm due to its flexibility and good empirical performance. More refined algorithms for computing *size-minimal* AXps of linear classifiers with polynomial delay (Marques-Silva et al., 2020; Cooper & Marques-Silva, 2023; Fernandes & Rocha, 2026) are restricted to binary tasks and cannot replace EA+ in our experiments. Our theoretical contribution is novel. Compatibly with it, the algorithms of Marques-Silva et al. (2020) require continuous features to be bounded. Our work, however, studies not only existence of AXps but also modeling choices (beyond boundedness) that control their *size*.

Three papers lie closest to our work. Srivastava et al. (2024) show that top-5 heuristic explanations of VLM-CBM decisions often allow flipping predictions; our results systematize their observation. Soria et al. (2026) introduce FXAI algorithms for computing AXps of prototype neural nets, where concept activations encode similarities to learned prototypical embeddings. Our theory backs up their findings but also allows us to identify architectural choices that facilitate the existence of more compact (weak) AXps. Their Algorithm 1 is close to EA+ except it ranks activations by absolute value only, which performs poorly for independent activations (see CelebA in Section D.2) and for VLM-CBMs (see Section 5.3). Finally, Paul et al. (2024) design an FXAI algorithm for neuro-symbolic models that, similarly to CBMs, pair concept extraction with (logic) inference (Marconato et al., 2026). Whereas they seek hierarchical AXps (covering both inputs and concepts) robust to local (ε-ball) perturbations, we construct concept-level AXps robust to all counterexamples in $\mathcal{S}$.

Transparency of CBMs. Most critiques of CBMs study human-alignment of the learned concepts. Some works show that, without concept supervision, CBMs can learn arbitrarily misaligned concepts (Bortolotti et al., 2025); others look at the impact of replacing expert supervision with VLM annotations (Kazmierczak et al., 2026; Debole et al., 2026a); others still show that, even when concepts are supervised, CBMs can push accuracy by encoding spurious information into the concepts (Mahinpei et al., 2021; Margeloiu et al., 2021; Raman et al., 2025) and propose several remedies (Havasi et al., 2022; Marconato et al., 2022; Huang et al., 2024; Fokkema et al., 2025; Almudévar et al., 2026) (but see Zarlenga (2026) for counterarguments). While concept alignment is sometimes referred to as "concept faithfulness" (Schmalwasser et al., 2026), none of these works consider the faithfulness of CBM *explanations*.

7 Conclusion

We study the faithfulness of CBM explanations and adapt ideas and techniques from FXAI to these models. Our work shows that many existing CBMs do not admit compact faithful explanations; that probabilistic modeling and structured sparsity enable retrieving substantially more compact (weak) AXps for regular and VLM-based CBMs; and that, while existing heuristic strategies for computing compact CBM explanations often fail to identify (even weak) AXps, our exact algorithm does. At a high level, our work suggests that (large) concept bottlenecks should not be naïvely equated with faithful interpretability.

In future work, we will study how to shrink faithful explanations further. A first direction involves squeezing (Schrodi et al., 2025; Yan et al., 2023) and enlarging (Feng et al., 2025) bottlenecks *on demand* while ensuring faithfulness. A second direction involves relaxing the notion of faithfulness, *e.g.*, by adapting *probabilistic* variants (Wang et al., 2021; Azzolin et al., 2025; Subercaseaux et al., 2025) that require the prediction holds with high probability. Finally, while we require explanations to be robust to both natural and (possibly out-of-distribution) human interventions, we plan to study how to shrink AXps by considering only counterexamples achievable by the backbone, leveraging approximations from the neural net verification literature (Singh et al., 2019; Zhang et al., 2018) to keep EA+ tractable. As a side benefit, this would automatically account for dependencies between input variables, which FXAI algorithms (including ours) typically ignore.

AI USE STATEMENT

We used generative AI tools for writing a first version of the code, for support when deriving the proofs in Section B, for scanning the literature in search of relevant publications we originally missed, and for reviewing the manuscript before submission. We have not used generative AI tools for other tasks with required disclosure. The code was manually reviewed by the authors. The proofs were manually written and carefully checked by the authors. The figures were made by the authors. The manuscript was written by the authors. The bibliography was manually compiled by the authors using bibentries sourced from Google Scholar. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Experimental details are provided in Section 5 and Section D. **The code is provided in the supplementary material** and will be released on GitHub upon acceptance. It includes a script to fully reproduce the experiments from scratch. We ran it twice and obtained the same results.

REFERENCES

Antonio Almudévar, José Miguel Hernández Lobato, and Alfonso Ortega. There was never a bottleneck in concept bottleneck models. In *ICLR*, 2026.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *NeurIPS*, 2018.

Steve Azzolin, Antonio Longa, Stefano Teso, and Andrea Passerini. Reconsidering faithfulness in regular, self-explainable and domain invariant gnns. In *ICLR*, 2025.

Matthew Barker, Katherine M Collins, Krishnamurthy Dvijotham, Adrian Weller, and Umang Bhatt. Selective concept models: Permitting stakeholder customisation at test-time. In *HCOMP*, 2023.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2009.

Alessandro Bogani, Nicola Debole, Emanuele Marconato, Andrea Pugnana, Katya Tentori, and Andrea Passerini. Are concept bottleneck models effective as decision-support systems? *arXiv:2608.25581*, 2026.

Andrea Bontempelli, Stefano Teso, Katya Tentori, Fausto Giunchiglia, and Andrea Passerini. Concept-level debugging of part-prototype networks. In *ICLR*, 2023.

Samuele Bortolotti, Emanuele Marconato, Paolo Morettin, Andrea Passerini, and Stefano Teso. Shortcuts and identifiability in concept-based models from a neuro-symbolic lens. In *NeurIPS*, 2025.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. *NeurIPS*, 2019.

Martin C Cooper and João Marques-Silva. Tractability of explaining classifier decisions. *Artificial Intelligence*, 2023.

Adnan Darwiche. Logic for Explainable AI. In *Annual ACM/IEEE Symposium on Logic in Computer Science*, 2023.

Nicola Debole, Pietro Barbiero, Francesco Giannini, Andrea Passerini, Stefano Teso, and Emanuele Marconato. If concept bottlenecks are the question, are foundation models the answer? *Machine Learning*, 2026a.

Nicola Debole, Andrea Passerini, Stefano Teso, Andrea Pugnana, and Emanuele Marconato. Concepts worth having: Refining vlm-guided concept bottleneck models with minimal annotations. *arXiv:2605.16405*, 2026b.

Mateo Espinosa Zarlenga, Katie Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: Intervention-aware concept embedding models. *NeurIPS*, 2023.

Jean Feng, Avni Kothari, Luke Zier, Chandan Singh, and Yan Shuo Tan. Bayesian concept bottleneck models with llm priors. *NeurIPS*, 2025.

Gleilson Pedro Fernandes and Thiago Alves Rocha. Concisely explaining the doubt: Minimum-size abductive explanations for linear models with a reject option. *World Conference on Explainable Artificial Intelligence*, 2026.

Hidde Fokkema, Tim van Erven, and Sara Magliacane. Sample-efficient learning of concepts with theoretical guarantees: from data to concepts without interventions. *NeurIPS*, 2025.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 2018.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. *NeurIPS*, 35:23386–23397, 2022.

Qihan Huang, Jie Song, Jingwen Hu, Haofei Zhang, Yong Wang, and Mingli Song. On the concept trustworthiness in concept bottleneck models. In *AAAI*, 2024.

Alexey Ignatiev. Towards trustable explainable ai. In *IJCAI-PRICAI*, 2020.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *AAAI*, 2019.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.

Yang Ji, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. A comprehensive survey on self-interpretable neural networks. *Proceedings of the IEEE*, 2025.

Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of biomedical and health informatics*, 2018.

Rémi Kazmierczak, Steve Azzolin, Goran Frehse, Eloïse Berthier, and Gianni Franchi. Enhancing concept localization in CLIP-based concept bottleneck models. *TMLR*, 2026.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *ICML*, 2023.

Patrick Knab, David Steinmann, Christian Bartelt, Kristian Kersting, Bernt Schiele, Thomas Seidl, Udo Schlegel, and Wolfgang Stammer. What2019s in the bottle? a survey and roadmap of concept bottleneck models. *TMLR*, 2026.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020.

Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Samuel J Gershman, and Finale Doshi-Velez. Human evaluation of models built for interpretability. In *HCOMP*, 2019.

Dan Ley, Giang Nguyen, Himabindu Lakkaraju, and Julius Adebayo. Prototype language models. *arXiv:2607.00510*, 2026.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 1989.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE international conference on computer vision*, 2015.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *arXiv:2106.13314*, 2021.

Emanuele Marconato, Andrea Passerini, and Stefano Teso. GlanceNets: Interpretabile, Leak-proof Concept-based Models. In *NeurIPS*, 2022.

Emanuele Marconato, Samuele Bortolotti, Emile Van Krieken, Paolo Morettin, Elena Umili, Antonio Vergari, Efthymia Tsamoura, Andrea Passerini, and Stefano Teso. Symbol grounding in neuro-symbolic ai: A gentle introduction to reasoning shortcuts. *Journal of Artificial Intelligence Research*, 2026.

Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv:2105.04289*, 2021.

Joao Marques-Silva. Logic-based explainability in machine learning. In *Reasoning Web. Causality, Explanations and Declarative Knowledge: 18th International Summer School 2022, Tutorial Lectures*. 2023.

Joao Marques-Silva, Thomas Gerspacher, Martin Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive bayes and other linear classifiers with polynomial time and delay. *NeurIPS*, 2020.

Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv:1802.00682*, 2018.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023.

Seonghwan Park, Jueun Mun, Donghyun Oh, and Namhoon Lee. An analysis of concept bottleneck models: Measuring, understanding, and mitigating the impact of noisy annotations. *NeurIPS*, 2025.

Sushmita Paul, Jinqiang Yu, Jip J Dekker, Alexey Ignatiev, and Peter J Stuckey. Formal explanations for neuro-symbolic ai. *arXiv:2410.14219*, 2024.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Naveen Raman, Mateo Espinosa Zarlenga, Juyeon Heo, and Mateja Jamnik. Do concept bottleneck models respect localities? *TMLR*, 2025.

Vikram V Ramaswamy, Sunnie SY Kim, Ruth Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability. In *CVPR*, 2023.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 2019.

Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 2022.

Laines Schmalwasser, Jan Blunk, Niklas Penzel, Julia Niebling, and Joachim Denzler. On the faithfulness of post-hoc concept bottleneck models. In *ECCV*, 2026.

Simon Schrodi, Julian Schur, Max Argus, and Thomas Brox. Selective concept bottleneck models without predefined concepts. *TMLR*, 2025.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. In *IJCAI*, 2018.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *ICML*, 2023.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *POPL*, 2019.

Jules Soria, Zakaria Chihani, Julien Girard-Satabin, Alban Grastien, Romain Xu-Darme, and Daniela Cancila. Formal abductive latent explanations for prototype-based networks. In *AAAI*, 2026.

Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. VLG-CBM: Training Concept Bottleneck Models with Vision-Language Guidance. In *NeurIPS*, 2024.

Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *CVPR*, 2021.

David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to intervene on concept bottlenecks. In *ICML*, 2024.

Bernardo Subercaseaux, Marcelo Arenas, and Kuldeep S Meel. Probabilistic explanations for linear models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Andong Tan, Fengtao Zhou, and Hao Chen. Explain via any concept: Concept bottleneck model with open vocabulary concepts. In *ECCV*, 2024a.

Zhen Tan, Tianlong Chen, Zhenyu Zhang, and Huan Liu. Sparsity-guided holistic explanation for llms with interpretable inference-time intervention. In *AAAI*, 2024b.

Stefano Teso, Öznur Alkan, Wolfang Stammer, and Elizabeth Daly. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, 2023.

Satoshi Tsutsui, Winnie Pang, and Bihan Wen. Wbcatt: A white blood cell dataset annotated with detailed morphological attributes. *NeurIPS*, 2023.

Moritz Vandenhirtz, Sonia Laguna, Ričards Marcinkevičs, and Julia E Vogt. Stochastic concept bottleneck models. *NeurIPS*, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Eric Wang, Pasha Khosravi, and Guy Van den Broeck. Probabilistic sufficient explanations. In *IJCAI*, 2021.

Carissa Wu, Sonali Parbhoo, Marton Havasi, and Finale Doshi-Velez. Learning optimal summaries of clinical time-series with concept bottleneck models. In *Machine Learning for Healthcare Conference*, 2022.

An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *ICCV*, 2023.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, 2023.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2006.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*, 2023.

Mateo Espinosa Zarlenga. In defense of information leakage in concept-based models. *ICML*, 2026.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *NeurIPS*, 2018.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2005.

A ALGORITHMIC DETAILS

The algorithms we evaluate are the following: *i*) `TopK-Act`, `TopK-Wei`, and `TopK-Prod`, three heuristic strategies that require a target explanation size k as input; *ii*) `EA`, a general exact algorithm, appearing as Algorithm 1 in Marques-Silva (2023), which does *not* require k as input; and *iii*) `EA+`, our improved `EA`. All pseudo-code assumes access to a CBM (f, g) with a linear inference layer and to a prediction $(\mathbf{s}, \hat{y})$ to be explained; we leave these implicit for brevity.

A.1 FINDING A COUNTEREXAMPLE

Most algorithms rely on a procedure ORACLE$(\mathcal{E}, \mathcal{S})$ for checking whether a given explanation $\mathcal{E}$ admits constructing a counter-example $\mathbf{s}' \in \mathcal{S}$. Given $(\mathbf{s}, \hat{y})$, a vector $\mathbf{s}'$ is a valid counterexample if it matches $\mathbf{s}$ on all activations covered by $\mathcal{E}$ but yields a different prediction $y' \neq \hat{y}$, that is:

$$\big(\mathbf{s}'_{\mathcal{E}} = \mathbf{s}_{\mathcal{E}}\big) \;\wedge\; \big(\hat{y} \notin \operatorname{argmax}_{y \in [m]} f(\mathbf{s}')_y\big). \tag{4}$$

We are dealing with *linear* inference layers of the form $f(\mathbf{s}) = W\mathbf{s} + \mathbf{w}_0$. Hence, letting $\mathbf{w}_y \in \mathbb{R}^b$ be the vector of inference layer weights W associated to class $y \in [m]$, and $w_{y,0}$ be the corresponding bias term, the second term in Eq. (4) translates to:

$$\langle \mathbf{w}_{y'}, \mathbf{s}' \rangle + \mathbf{w}_{y',0} > \langle \mathbf{w}_{\hat{y}}, \mathbf{s}' \rangle + \mathbf{w}_{\hat{y},0}. \tag{5}$$

We ignore follow-up monotonic transformations, *e.g.*, sigmoid and softmax activations, as they do not alter the ranking between classes. In order to find a counterexample, the oracle has to find a class $y' \neq \hat{y}$ out of $m-1$ possible alternatives and a vector $\mathbf{s}' \in \mathcal{S}$ that satisfies Eq. (5). This can be readily done by solving, for each candidate y', the following optimization problem:

$$\max_{\mathbf{s}'} \quad \langle \mathbf{w}_{y'} - \mathbf{w}_{\hat{y}}, \mathbf{s}' \rangle + (\mathbf{w}_{y',0} - \mathbf{w}_{\hat{y},0}) \tag{6}$$

$$\text{s.t.} \quad \mathbf{s}'_{\mathcal{E}} = \mathbf{s}_{\mathcal{E}} \wedge \mathbf{s}' \in \mathcal{S} \tag{7}$$

The objective is *linear* in $\mathbf{s}'$. At the same time, in our setting the feasible set $\mathcal{S}$ can be encoded as a (conjunction of) *linear constraints*. For cosine similarities ($\mathcal{S} = [-1, 1]^b$), ReLU activations ($\mathcal{S} = [0, +\infty)^b$), and raw logits ($\mathcal{S} = (-\infty, +\infty)^b$), the constraint is a conjunction of axis-aligned open or closed bounds $a_j \leq s'_j \leq b_j$. For probabilistic activations ($\mathcal{S} \equiv \boldsymbol{\delta}\mathbf{w} \subseteq [0, 1]^b$, where $\boldsymbol{\delta}\mathbf{w}$ is a product of simplices) it is a conjunction of per-concept linear constraints $\sum_j s_{i,j} = 1$, where i runs over concepts and j over activations encoding that concept, and non-negativity constraints $s_j \geq 0$ running over all activations. Hence, finding a counterexample boils down to a linear program (LP).

In short: the oracle is a collection of $m-1$ LPs. If the LP for class y' admits a positive objective, its solution $\mathbf{s}'$ is a valid counterexample for $\mathcal{E}$ within $\mathcal{S}$, and the oracle returns it. Otherwise, if no LP yields a positive objective, $\mathcal{E}$ admits no counterexample and the oracle returns `false`.

A.2 THE HEURISTIC ALGORITHMS

These are the three strategies introduced in Eq. (2). Given a decision $(\mathbf{s}, \hat{y})$ and a budget k, they sort the b concept activations by either largest activation (`TopK-Act`), largest absolute weight (`TopK-Wei`), or largest absolute score (activation times weight) (`TopK-Prod`), and construct an explanation $\mathcal{E}$ by taking the top k activations. This procedure, however, does not guarantee the output is an AXp (faithful and subset-minimal) or even a weak AXp (faithful but not subset-minimal). Crucially, *the likelihood of finding a (weak)* AXp *increases with the budget k, making a fair comparison with `EA+` difficult, as the latter does not require to fix k.*

Algorithm 1 TOPK$^\star(\mathcal{S}, b, h)$

```
π ← [b] decreasing by h's criterion
for k = 1, ..., b do
    if not ORACLE(π_{1:k}, S) then
        return π_{1:k}
return [b]
```

For this reason, rather than fixing k *a priori*, we compute it dynamically by taking the smallest k such that the resulting explanation a (weak) AXps. This is precisely what `TopK*` (Algorithm 1) does: it first ranks the b concept activations according to value, absolute weight, or absolute score – depending on the choice of heuristic h – and then checks prefixes of increasing size k until it finds a (weak) AXp. If no proper prefix is faithful, it returns the entire bottleneck. This ensures the resulting explanation is a weak AXp at the cost of at most b calls to ORACLE.

A.3 THE EA ALGORITHM

This algorithm is a direct application of the SHRINK procedure in Algorithm 2. The procedure takes a weak AXp $\mathcal{E} \subseteq [b]$ as input and iterates over all indices $j \in \mathcal{E}$ in some order (in practice, lowest to highest, for simplicity): for each j, it checks whether $\mathcal{E}$ remains a weak AXp – *i.e.*, whether it still admits no counterexamples, using ORACLE – even if j is removed. In this case, it removes j, otherwise it keeps it. Note that *the output is guaranteed to be a subset-minimal weak* AXp, that is, an AXp. The EA algorithm simply invokes SHRINK on the full bottleneck $\mathcal{E} = [b]$, which is a weak AXp by construction. The oracle is called exactly b times.

Algorithm 2 SHRINK$(\mathcal{S}, \mathcal{E})$

for $j \in \mathcal{E}$ **do**
  if not ORACLE$(\mathcal{E} \setminus \{j\}, \mathcal{S})$ **then**
    $\mathcal{E} \leftarrow \mathcal{E} \setminus \{j\}$
return $\mathcal{E}$

A.4 THE EA+ ALGORITHM

EA ignores instance-specific information (*e.g.*, the values and weights of the concept activations) and in doing so may return an overly large AXp. At the same time, it requires b oracle calls, which can be problematic for large bottlenecks.

Our algorithm, EA+, implements two simple modifications that aim to lessen both issues, see Algorithm 3. It invokes TopK* three times, once for each heuristic h, obtaining three weak AXp $\mathcal{E}_h$. Then it applies the SHRINK procedure for removing possibly useless activations to each of them. Out of the three resulting AXps, it returns the smallest.

Moreover, rather than invoking Algorithm 1, which costs b oracle calls, it employs a more efficient variant that employs binary search instead, listed in Algorithm 4. The goal is to find the shortest prefix of the bottleneck – sorted according to the given heuristic h – that is faithful. Rather than scanning prefixes linearly as Algorithm 1 does, it employs binary search: in each iteration, it checks whether the prefix delimited by $(1, \frac{1}{2}\lfloor \alpha + \beta \rfloor)$ admits a counterexample. If it does not, it is faithful, so the algorithm halves it (by decreasing β), and repeats. If it does, it is *not* faithful, so the algorithm enlarges it (by increasing α), and repeats. This modification reduces the number of oracle calls to $O(\log_2 b)$. This offsets the total cost of having to call the procedure for three times total, once per heuristic.

Algorithm 3 EA+$(\mathcal{S}, b)$

for each heuristic h **do**
  $\mathcal{E}_h \leftarrow$ TOPK$^\star_{bs}(\mathcal{S}, b, h)$
  $\mathcal{E}_h \leftarrow$ SHRINK$(\mathcal{S}, \mathcal{E}_h)$
return $\operatorname{argmin}_h |\mathcal{E}_h|$

Algorithm 4 TOPK$^\star_{bs}(\mathcal{S}, b, h)$

$\pi \leftarrow [b]$ decreasing by h's criterion
$\alpha \leftarrow 0$; $\beta \leftarrow b$
while $\alpha < \beta$ **do**
  $\mu \leftarrow \lfloor \frac{1}{2}(\alpha + \beta) \rfloor$
  if not ORACLE$(\pi_{1:\mu}, \mathcal{S})$ **then**
    $\beta \leftarrow \mu$
  else
    $\alpha \leftarrow \mu + 1$
return $\pi_{1:\alpha}$

B CONDITIONS FOR (IN)EXISTENCE OF WEAK AXPS

Recall that an explanation is a weak AXp insofar as it does not allow to *flip* the CBM's prediction by manipulating the concept activations it does not cover. In this section, we make explicit the conditions for flipping the label of an arbitrary inference layer weights W for counterexamples taken from an activation space of the form

$$\mathcal{S} = \bigtimes_{i=1}^{b} [\alpha_i, \beta_i]. \tag{8}$$

Here, $\alpha_i, \beta_i \in \mathbb{R}$ are arbitrary (finite or infinite) per-activation constants with $\alpha_i < \beta_i$. This will be our departure point for discussing four different continuous activation spaces $\mathcal{S}$ most common in the CBM literature (as discussed in Section 4.1), namely:

- **Unbounded**: $\mathcal{S} = (-\infty, \infty)^b$, *e.g.*, when concept activations are unconstrained logits, as in many CBM architectures (see Table 1).
- **Bounded-from-below**: $\mathcal{S} = [\alpha, \infty)^b$, *e.g.*, as in Selective CBMs (Barker et al., 2023) where activations are only retained if they exceed a threshold $\alpha \geq 0$.
- **Bounded**: $\mathcal{S} = [\alpha, \beta]^b$, *e.g.*, in the case of independent probabilistic activations, in which case $\alpha = 0$, $\beta = 1$.
- **Bounded-on-simplices**: $\mathcal{S} \subseteq [0, 1]^b$, a product of per-concept simplices, representing the case of probabilistic modeling with (binary and) categorical concepts.

Notation. We focus on concept activations $\mathbf{s} \in \mathcal{S}$ associated to an input $\mathbf{x} \in \mathcal{X}$. Recall that the inference layer is a linear function $f : \mathcal{S} \to \mathbb{R}^m$, *i.e.*,

$$f(\mathbf{s}) = W\mathbf{s} + \mathbf{w}_0 \tag{9}$$

where $W \in \mathbb{R}^{m \times b}$ and $\mathbf{w}_0 \in \mathbb{R}^m$, and the prediction is

$$\hat{y} := \operatorname*{argmax}_{y \in [m]} f_y(\mathbf{s}), \quad \text{where} \quad f_y(\mathbf{s}) = \mathbf{w}_y^\top \mathbf{s} + w_{0,y}. \tag{10}$$

For any class $y' \in [m]$, we use $\mathbf{w}_{y'} \in \mathbb{R}^b$ to indicate its associated weights. Given a non-predicted class j and a prediction $\hat{y}$, we denote the deltas between their associated weights and biases as:

$$\delta\mathbf{w}^{(j)} := \mathbf{w}_j - \mathbf{w}_{\hat{y}}, \quad \text{and} \quad \delta w_0^{(j)} := w_{0,j} - w_{0,\hat{y}} \tag{11}$$

and the difference between their logits as

$$\delta\mu^{(j)}(\mathbf{s}) := f_j(\mathbf{s}) - f_{\hat{y}}(\mathbf{s}) = \delta\mathbf{w}^{(j)} \cdot \mathbf{s} + \delta w_0^{(j)}. \tag{12}$$

Finally, for any real $a \in \mathbb{R}$, we write

$$a^+ := \max(0, a), \quad a^- := \max(0, -a). \tag{13}$$

B.1 THE INDEPENDENT CONCEPT ACTIVATIONS CASE

Throughout, we will use the shorthands $\bar{\mathcal{E}} := [b] \setminus \mathcal{E}$ and $\mathcal{S}_\mathcal{E}(\mathbf{s}) = \{\mathbf{s}' \in \mathcal{S} : \mathbf{s}'_\mathcal{E} = \mathbf{s}_\mathcal{E}\}$. First, we identify general conditions for label flipping in a bounded asymmetric domain when fixing a subset $\mathcal{E} \subset [b]$ of concept activations. Our lemma follows on the footsteps of Marques-Silva et al. (2020).

Lemma 2. *Let $\mathcal{S} = \bigtimes_{i=1}^{b} [\alpha_i, \beta_i]$. Take any $\mathbf{s} \in \mathcal{S}$, a prediction $\hat{y} \in \operatorname{argmax}_y f_y(\mathbf{s})$, a non-predicted class $j \in [m]$, and a subset $\mathcal{E} \subset [b]$. Let $\mu_j := \max_{\mathbf{s}^* \in \mathcal{S}_\mathcal{E}(\mathbf{s})} \delta\mu_j(\mathbf{s}^*)$ be the largest logit difference between classes j and $\hat{y}$ achievable by manipulating concept activations outside of $\mathcal{E}$. This quantity can be rewritten as:*

$$\mu_j = \delta\mu^{(j)}(\mathbf{s}) + \sum_{i \in \bar{\mathcal{E}}} \left[(\delta w_i^{(j)})^+ (\beta_i - s_i) + (\delta w_i^{(j)})^- (s_i - \alpha_i) \right]. \tag{14}$$

Proof. Recall that the difference in logits for activation j is given by

$$\delta\mu^{(j)}(\mathbf{s}) = \delta\mathbf{w}^{(j)} \cdot \mathbf{s} + \delta w_0^{(j)}. \tag{15}$$

We expand the dot product as follows

$$\delta\mu^{(j)}(\mathbf{s}) = \sum_{i\in\mathcal{E}} \delta w_i^{(j)} s_i + \sum_{i\in\bar{\mathcal{E}}} \delta w_i^{(j)} s_i + \delta w_0^{(j)} \,. \tag{16}$$

Now, take any non-fixed concept activation $i \in \bar{\mathcal{E}}$. Following (Marques-Silva, 2023), we maximize the logit of class j by setting s_i to β_i if $\delta w_i^{(j)}$ is positive and to α_i if $\delta w_i^{(j)}$ is negative for each $i \in \bar{\mathcal{E}}$. Plugging this into Eq. (16) and maximizing shows that μ_j, the maximum difference in logits attainable for class j, is:

$$\begin{aligned}
\max_{\mathbf{s}^*\in\mathcal{S}_{\mathcal{E}}(\mathbf{s})} \delta\mu_j(\mathbf{s}^*) &= \sum_{i\in\mathcal{E}} \delta w_i^{(j)} s_i + \sum_{i\in\bar{\mathcal{E}}} \Big[(\delta w_i^{(j)})^+\beta_i - (\delta w_i^{(j)})^-\alpha_i\Big] + \delta w_0^{(j)} && (17)\\
&= \underbrace{\sum_{i\in\mathcal{E}} \delta w_i^{(j)} s_i + \sum_{i\in\bar{\mathcal{E}}} \delta w_i^{(j)} s_i + \delta w_0^{(j)}}_{\delta\mu^{(j)}(\mathbf{s})} \\
&\qquad - \sum_{i\in\bar{\mathcal{E}}} \delta w_i^{(j)} s_i + \sum_{i\in\bar{\mathcal{E}}} \Big[(\delta w_i^{(j)})^+\beta_i - (\delta w_i^{(j)})^-\alpha_i\Big] && (18)\\
&= \delta\mu^{(j)}(\mathbf{s}) + \sum_{i\in\bar{\mathcal{E}}} \Big[(\delta w_i^{(j)})^+(\beta_i - s_i) + (\delta w_i^{(j)})^-(s_i - \alpha_i)\Big] && (19)
\end{aligned}$$

This proves the claim. □

The reason we care about μ_j is that it allows us to rewrite the condition under which a label-flipping counterexample exists as follows: there must exist a class $j \neq \hat{y}$ such that $\mu_j > 0$, *i.e.*, that attains a larger logit than the prediction $\hat{y}$. Equivalently, we can write $\max_{j\neq\hat{y}} \mu_j > 0$. Next, we exploit Lemma 2 to identify explicity conditions under which different choices $\mathcal{S}$ admit (weak) AXps smaller than the bottleneck:

Proposition 3 (Extended Proposition 1)**.** *For any CBM prediction $(\mathbf{s}, \hat{y}) \in \mathcal{S} \times [m]$, an explanation $\mathcal{E} \subset [b]$ smaller than the entire bottleneck **cannot** be a weak* AXp *unless:*

1. ***Unbounded** ($\mathcal{S} = (-\infty, \infty)^b$) There exists no concept activation $i \in \bar{\mathcal{E}}$ and alternative class $j \neq \hat{y}$ such that $\delta w_i^{(j)} \neq 0$.*
2. ***Bounded from below**: ($\mathcal{S} = [\alpha, \infty)^b$.) There exists no concept activation $i \in \bar{\mathcal{E}}$ and alternative class $j \neq \hat{y}$ such that $\delta w_i^{(j)} > 0$.*
3. ***Bounded symmetric**: ($\mathcal{S} = [\alpha, \beta]^b$) There exists no alternative class $j \neq \hat{y}$ such that $\max_{j\neq\hat{y}} \mu_j > 0$, where μ_j can be written as*

$$\mu_j = \delta\mu_j(\bar{\mathbf{s}}) + \frac{\beta - \alpha}{2} \|\delta\mathbf{w}_{\bar{\mathcal{E}}}^{(j)}\|_1 \,; \tag{20}$$

   and $\bar{\mathbf{s}}$ is such that $\bar{\mathbf{s}}_{\bar{\mathcal{E}}} = \frac{\alpha+\beta}{2}\mathbf{1}_{\bar{\mathcal{E}}}$ is the mid-point between α and β for all coordinates in $\bar{\mathcal{E}}$.

4. ***Bounded asymmetric**: ($\mathcal{S} = \bigtimes_{i=1}^b [\alpha_i, \beta_i]$). There exists no alternative class $j \neq \hat{y}$ such that $\max_{j\neq\hat{y}} \mu_j > 0$, where μ_j can be written as*

$$\mu_j = \delta\mu^{(j)}(\mathbf{s}) + \sum_{i\in\bar{\mathcal{E}}} \Big[(\delta w_i^{(j)})^+(\beta_i - s_i) + (\delta w_i^{(j)})^-(s_i - \alpha_i)\Big] \,; \tag{21}$$

Proof. Recall that finding the conditions for $\mathcal{E}$ to be an AXp is equivalent to find conditions for which a label flip cannot occur. Since cases satisfy the conditions of Lemma 2, and we will exploit it below. We begin from the most general case.

Bounded asymmetric. Eq. (21) follows directly from the lemma. An adversary can flip the label if and only if there exist a counterexample $\mathbf{s}^* \in \mathcal{S}_{\mathcal{E}}(\mathbf{s})$ such that $\operatorname{argmax}_j f_j(\mathbf{s}^*) \neq \operatorname{argmax}_j f_j(\mathbf{s})$, or, equivalently, $\max_{j\neq\hat{y}} \mu_j > 0$. If $\mu_j > 0$, then $\max_{\mathbf{s}^*\in\mathcal{S}_{\mathcal{E}}(\mathbf{s})} f_j(\mathbf{s}^*) - f_{\hat{y}}(\mathbf{s}^*) > 0$, that is, $\mathbf{s}^*$

ensures class j attains a larger logit than the prediction $\hat{y}$. Conversely, if $\max_j \mu_j \leq 0$, no $\mathbf{s}^*$ can induce a label flip.

Bounded symmetric. This is a corollary of Lemma 2, as the symmetric domain $[\alpha, \beta]^b$ is a special case of the asymmetric case. Denoting with $\bar{\mathbf{s}} \in \mathcal{S}_{\mathcal{E}}(\mathbf{s})$ the activation with $\bar{\mathbf{s}}_{\bar{\mathcal{E}}} = \frac{\alpha+\beta}{2}\mathbf{1}_{\bar{\mathcal{E}}}$, we can write

$$\max_{\mathbf{s}^* \in \mathcal{S}_{\mathcal{E}}(\mathbf{s})} \delta\mu_j(\mathbf{s}^*) = \sum_{i \in \mathcal{E}} \delta w_i^{(j)} s_i + \sum_{i \in \bar{\mathcal{E}}} \max\left(\delta w_i^{(j)}\beta, \delta w_i^{(j)}\alpha\right) + \delta w_0^{(j)} \tag{22}$$

$$= \underbrace{\sum_{i \in \mathcal{E}} \delta w_i^{(j)} s_i + \sum_{i \in \bar{\mathcal{E}}} \delta w_i^{(j)} \bar{s}_i + \delta w_0^{(j)}}_{\delta\mu_j(\bar{\mathbf{s}})} - \sum_{i \in \bar{\mathcal{E}}} \delta w_i^{(j)} \bar{s}_i + \sum_{i \in \bar{\mathcal{E}}} \max\left(\delta w_i^{(j)}\beta, \delta w_i^{(j)}\alpha\right) \tag{23}$$

$$= \delta\mu_j(\bar{\mathbf{s}}) - \frac{\alpha+\beta}{2} \sum_{i \in \bar{\mathcal{E}}} \delta w_i^{(j)} + \sum_{i \in \bar{\mathcal{E}}} \max\left(\delta w_i^{(j)}\beta, \delta w_i^{(j)}\alpha\right) \tag{24}$$

Now we make use of $\max(a, b) = (a + b + |a - b|)/2$ to rewrite

$$\max\left(\delta w_i^{(j)}\beta, \delta w_i^{(j)}\alpha\right) = \frac{(\alpha+\beta)}{2}\delta w_i^{(j)} + \frac{(\beta-\alpha)}{2}|\delta w_i^{(j)}| \tag{25}$$

and substituting back to the previous expression we get the result:

$$\max_{\mathbf{s}^* \in \mathcal{S}_{\mathcal{E}}(\mathbf{s})} \delta\mu_j(\mathbf{s}^*) = \delta\mu_j(\bar{\mathbf{s}}) - \frac{\alpha+\beta}{2} \sum_{i \in \bar{\mathcal{E}}} \delta w_i^{(j)} + \sum_{i \in \bar{\mathcal{E}}} \left[\frac{(\alpha+\beta)}{2}\delta w_i^{(j)} + \frac{(\beta-\alpha)}{2}|\delta w_i^{(j)}|\right] \tag{26}$$

$$= \delta\mu_j(\bar{\mathbf{s}}) + \frac{\beta-\alpha}{2}\|\delta\mathbf{w}_{\bar{\mathcal{E}}}^{(j)}\|_1 \tag{27}$$

Bounded from below. We first check if there are any positive weights. Because $\mathcal{S}$ is only bounded from below, by rewriting the logit difference expression as

$$\delta\mu_j(\mathbf{s}^*) = \delta\mu^{(j)}(\mathbf{s}) + \sum_{i \in \bar{\mathcal{E}}} \max(\delta w_i^{(j)}, 0)(s_i^* - s_i) + \sum_{i \in \bar{\mathcal{E}}} \max(-\delta w_i^{(j)}, 0)(s_i - s_i^*) \tag{28}$$

we can see that the quantity $(s_i^* - s_i)$ is also unbounded. Therefore, any positive weight $\delta w_i^{(j)}$ would naturally allow for $\mathbf{s}^* \in \mathcal{S}$ with fixed $\mathbf{s}_{\mathcal{E}}^* = \mathbf{s}_{\mathcal{E}}$ that give positive $\ell\mu_j(\mathbf{s}^*)$. If $\exists j \neq \hat{y}, \exists i \in \bar{\mathcal{E}}$, such that $\delta w_i^{(j)} > 0$, we get

$$\lim_{s_i^* \to \infty} \delta\mu_j(\mathbf{s}^*) = \infty\,. \tag{29}$$

When these conditions are not met, *i.e.*, for all $j \neq \hat{y}$ and $i \in \bar{\mathcal{E}}$, $\delta w_i^{(j)} < 0$, the μ_j expression can be derived from Lemma 2 and simplifies as[8]

$$\mu_j = \delta\mu^{(j)}(\mathbf{s}) + \sum_{i \in \bar{\mathcal{E}}} \left[\underbrace{(\delta w_i^{(j)})^+}_{=0}(\beta - s_i) + (\delta w_i^{(j)})^-(s_i - \alpha)\right] \tag{30}$$

$$= \delta\mu^{(j)}(\mathbf{s}) - \sum_{i \in \bar{\mathcal{E}}} \delta w_i^{(j)}(s_i - \alpha) \tag{31}$$

Unbounded. follows directly from the previous case. Since the domain is unbounded, we have that if $\exists j \neq \hat{y}, \exists i \in \bar{\mathcal{E}}$, such that $\delta w_i^{(j)} > 0$,

$$\lim_{s_i^* \to \infty} \delta\mu_j(\mathbf{s}^*) = \infty \tag{32}$$

or, if $\exists j \neq \hat{y}, \exists i \in \bar{\mathcal{E}}$, such that $\delta w_i^{(j)} < 0$,

$$\lim_{s_i^* \to -\infty} \delta\mu_j(\mathbf{s}^*) = \infty\,. \tag{33}$$

This shows that non-zero weight differences in $\bar{\mathcal{E}}$ lead to a label flip for unbounded domains. □

[8] To be precise, here we are taking the convention here that $0 \cdot (+\infty) = 0$ as resulting from the limit of $\beta \to \infty$.

B.2 The Probabilistic Concept Activations Case

Proposition 3 only considers independent concept activations. When modeling concept activations as distributions, activations referring to alternative values of the same categorical concept (*e.g.*, different colors) must sum to one, so independence no longer holds. We study this case separately.

Let $\mathcal{S} \subsetneq [0,1]^b$ be the probability simplex for $t > 1$ categorical concepts. For each concept $c \in [t]$, the number of values it can take are set to n_c, such that

$$b = \textstyle\sum_{c=1}^{t} n_c \,. \tag{34}$$

We indicate with $B_1, \ldots, B_t$ the partitions induced by the t concepts over the b total activations. The probability simplex is given by

$$\mathcal{S} = \bigtimes_{c=1}^{t} \Delta^{n_c-1}, \quad \Delta^{n_c-1} := \{\mathbf{z} \in [0,1]^{n_c} : \sum_{i \in B_c} z_i = 1\} \,. \tag{35}$$

For a given $\mathcal{E} \in [b]$, we denote with $F_c := B_c \setminus \mathcal{E}$ the free coordinates per concept c that are not fixed by $\mathcal{E}$. Because activations in $\mathcal{E}$ are fixed, the per-concept residual probability mass is given by

$$\gamma := \sum_{\mathcal{E} \in F_c} s_i = 1 - \sum_{i \in B_c} s_i \,. \tag{36}$$

We use this to prove how μ_j changes:

Proposition 4. *Let $\mathcal{S} = \bigtimes_{c=1}^{t} \Delta^{n_c-1}$, let $\mathbf{s} \in \mathcal{S}$, let $\hat{y}$ be the argmax of $f(\mathbf{s})$ and let $\mathcal{E} \subset [b]$ be fixed. Define, for $j \neq \hat{y}$,*

$$\mu_j := \delta\mu^{(j)}(\mathbf{s}) + \sum_{c=1}^{t} \rho_c, \quad \rho_c := \left[\gamma_c \max_{i \in F_c} \delta w_i^{(j)} - \sum_{i \in F_c} \delta w_i^{(j)} \, s_i \right] \tag{37}$$

with the convention $\rho_c = 0$ when $F_c = \varnothing$. A label flip is possible if and only if $\max_{j \neq \hat{y}} \mu_j > 0$.

Proof. We start from the logit difference expression for $f_j - f_{\hat{y}}$ and consider a generic $\mathbf{s}^* \in \mathcal{S}_{\mathcal{E}}(\mathbf{s})$

$$\delta\mu_j(\mathbf{s}^*) = \delta\mu^{(j)}(\mathbf{s}) + \sum_{i \in \bar{\mathcal{E}}} \delta w_i^{(j)} (s_i^* - s_i) \tag{38}$$

$$= \delta\mu^{(j)}(\mathbf{s}) + \sum_{c=1}^{t} \sum_{i \in F_c} \delta w_i^{(j)} (s_i^* - s_i) \,, \tag{39}$$

where the last step follows by the fact that only activations in F_c are allowed to differ to the original $\mathbf{s}$. Because activations are bounded in each $c \in [t]$, the maximum gain per concept is given by placing all residual probability mass μ_c to the highest weight difference, *i.e.*,

$$\max_{\mathbf{s}^* \in \mathcal{S}} \sum_{i \in F_c} \delta w_i^{(j)} (s_i^* - s_i) = \gamma_c \max_{i \in F_c} \delta w_i^{(j)} \,. \tag{40}$$

Therefore,

$$\max_{\mathbf{s}^* \in \mathcal{S}_{\mathcal{E}}(\mathbf{s})} \delta\mu_j(\mathbf{s}^*) = \delta\mu^{(j)}(\mathbf{s}) + \sum_{c=1}^{t} \left[\gamma_c \max_{i \in F_c} \delta w_i^{(j)} - \sum_{i \in F_c} \delta w_i^{(j)} \, s_i \right] = \mu_j \,. \tag{41}$$

Clearly, if $\mu_j > 0$ a label flip is possible and, viceversa, if $\mu_j < 0$ no label flip can happen. □

B.3 Discussion

We briefly discuss our theoretical results. First of all, the **unbounded** case recovers Proposition 1: it shows that unless $\mathcal{E}$ covers all unbounded activations associated to a non-zero weight difference between the prediction $\hat{y}$ and an alternative class $j \neq \hat{y}$, then it cannot be a weak AXp. Hence, any weak AXp is forced to cover *all* concept activations linking the prediction to any alternative class through a non-zero weight difference. The more such activations a CBM admits, the more

(weak) AXp size blows up. This is precisely what happens with dense CBMs with unbounded logit activations in Section 5.1. It also backs up our suggestion to implement `glasso` at training time, as doing so is likely to reduce the number of non-zero weight differences, in turn shrinking the size of weak AXps.

The **bounded from below** case recovers both CBMs modeling concepts using ReLU activations and Selective CBMs (Schrodi et al., 2025). The logic is the same as for the unbounded case above, except that now AXps have to cover all concept activations associated to a *positive* weight difference, while *negative* weight differences can be left uncovered. While this can shrink explanation sizes, CBMs do not control for weight polarity during training. In contrast, our `glasso` strategy encourages differences to be zero, typically reducing the number of (positive and negative) weight differences between classes. This yields the improvements reported in Section D.5.

The **bounded symmetric** case shows that what matters are the ranges of activations $[\alpha, \beta]$ and the range of weight differences $\delta\mathbf{w}_i^{(j)}$ for $i \in \bar{\mathcal{E}}$. Smaller activation ranges leave less room for manipulating the concept activations, and smaller weight differences leave less room for translating these into a higher score for the alternative class j. This backs up our discussion of cosine similarity activations in Section 4.1, which constrain activations to $[-1, 1]^b$ and are relevant to VLM-CBMs, and also partially motivates our suggestion to model concepts probabilistically, which constrains activations to $[0, 1]^b$. It also suggests that shrinking the range of the learned weights, which `enet` and `glasso` both implicitly do, is beneficial for constraining the adversary in the bounded case. The **bounded asymmetric** case conveys essentially the same message.

Finally, the **probabilistic** case shows that the exact condition for label flipping is stricter than that for the pure bounded (symmetric and asymmetric) case, meaning that the product-of-simplices structure of the activation space prevents it even more from constructing counterexamples. Since the concept activation space is fixed, the only knob for reducing the number of possible counterexamples – and therefore shrinking weak AXps – is the magnitude of the learned weights, which both `enet` and `glasso` can shrink. Further improvements may be gained by designing sparsification penalties targeting the quantities reported in our propositions; we will explore these in future work.

C EXPERIMENTAL DETAILS

We ran all experiments on a 64-core machine with 256 GiB RAM and two RTX 5090 GPUs. These are used only for computing the CLIP embeddings and nothing else. Excluding this step, our experiments take about one full day to complete in this setup.

Data preparation. For each dataset, we use the released train/validation/test partitions; for `CUB`, which has no validation split, we carve one via stratified sampling from the training split. Images are RGB and passed through CLIP's own preprocessing transform without extra resizing or normalization. We discuss the preparation of each dataset in turn.

For `CelebA` (Liu et al., 2015), we follow the pipeline of Debole et al. (2026a). We translate each of the 40 binary concept annotations from $\{\pm 1\}$ to $\{0, 1\}$, and predict `Male` from the others. Rather than subsampling, as in (Debole et al., 2026b), we keep all splits as they are. We stress that `CelebA` has no categorical concepts and therefore `glasso` does not apply to it.

For `Derm7pt` (Kawahara et al., 2018), we take the seven dermoscopic checklist criteria from the `meta.csv` file shipping with the dataset and one-hot encode them into 28 activations. Following (Kawahara et al., 2018), we also collapse the 20 raw diagnoses into 5 classes (basal cell carcinoma, melanoma, nevus, seborrheic keratosis, and a miscellaneous class) by matching keywords. This reproduces their Table I. Depending on the experiment, Kawahara et al. (2018) perform one additional round of label merging; we skip this step.

For `WBCAtt` (Tsutsui et al., 2023) we one-hot encode the 11 morphological attributes into 31 activations, and use the released 5-way cell-type label.

For `CUB` (Wah et al., 2011), we follow Koh et al. (2020) by replacing per-image concept annotations with class-level concepts, using only data from the training split to determine the aggregation. Koh et al. (2020) aggregation step, however, does not treat categorical concepts are as such. Specifically, it thresholds each attribute independently, which often does not yield a one-hot vector useful for training categorical concepts. Moreover, Koh et al. (2020) filter out activations that do not contribute to at least 10 classes, as these are too sparse/noisy to be predicted with high accuracy ((Koh et al., 2020, Appendix A)), and only 112 activations survive this stage. This also removes categories from categorical concepts. Since we are interested in evaluating the interpretability of CBMs and the benefits of modeling concept activations probabilistically, rather than sheer prediction accuracy, we depart from their processing. First, we aggregate annotations by taking, within each of the 28 categorical concepts the value with the largest share of positive votes, with ties broken at random. Moreover, we skip the filtering step.

LLM-generated vocabularies All VLM-CBM vocabularies are taken directly from the LF-CBM, VLM-CBM and LaBo GitHub repositories and used as-is to compute, for each image, concept activations. Concretely, we compute the cosine similarity between each image embedding and each concept textual description, both normalized, obtaining concept activation that individually range in $[-1, 1]$. For the LF-CBM and VLM-CBM vocabularies, we filter out those activations that do not meet a fixed 0.25 threshold (computed on the training split, to avoid contamination), as done by Oikarinen et al. (2023), yielding bottleneck sizes of $b = 202$ for the LF-CBM vocabulary and $b = 419$ for the VLG-CBM one. Since the LaBo vocabulary is pre-filtered via submodular optimization (Yang et al., 2023), we keep all activations.

Architectures. For all models, the concept probes and dense inference layers are `sklearn` (Pedregosa et al., 2011) logistic regressors fit using L-BFGS (Liu & Nocedal, 1989) with default hyperparameters. `enet-` and `glasso`-sparsified inference layers are fit with FISTA (Beck & Teboulle, 2009) with step size $1/L$, where L is the (approximate) Lipschitz constant of the loss, ensuring stability and quick convergence.

C.1 Concept Activation Spaces: In- vs. Out-of-distribution Counterexamples

In all experiments, we employ a model-specific activation space $\mathcal{S}$ for both testing whether heuristic explanations are (weak) AXps, and for constructing AXps with the exact algorithms. As anticipated in Sections 4.1 and 4.2, we set $\mathcal{S} = (-\infty, \infty)^b$ for logit-based CBMs, $\mathcal{S} = [0, \infty)^b$ for ReLU CBMs and Selective CBMs, $\mathcal{S} = [-1, 1]^b$ for VLM-CBMs with cosine similarity activations, and $\mathcal{S} \subseteq [0, 1]^b$ for CBMs modeling concepts probabilistically.

Natural vs. out-of-distribution counterexamples. In all cases, the adversary can construct counterexamples by manipulating the activations not covered by explanations *independently from each another* – except for CBMs modelling concepts probabilistically, where activations of the same categorical concept are mutually constrained to sum to 1. This setup ignores correlations between concept activations, meaning (weak) AXps are required to be robust even against "out-of-distribution" (OOD) counterexamples that the bottleneck does not naturally output. While this choice does not invalidate our modeling and algorithmic contributions, we wish to elucidate the rationale behind it.

First, even though OOD counterexamples cannot be generated by the backbone, they can be manually constructed by (possibly ill-intentioned) stakeholders intervening on the model. By ensuring our activation spaces $\mathcal{S}$ match the full range of attainable activation vectors, *we explicitly ensure our (weak)* AXp*s are robust against all natural and human-constructed counterexamples*. Second, ensuring an explanation does not admit naturally occurring counterexamples – with guarantees – requires inferring the image of the bottleneck, or at least to build a tractable outer approximation thereof. This is highly non-trivial (Singh et al., 2019; Zhang et al., 2018), making it technically challenging to construct explanations robust only to naturally occurring counterexamples. Regardless, while focusing on natural counterexamples only downgrades the guarantees that AXps provide, doing so *can* in principle shrink AXp size further. Hence, as mentioned in Section 7, we plan to investigate how to do so in future work.

D ADDITIONAL EXPERIMENTS

In this section, we report the following additional results:

Section D.1 Presents an in-depth algorithmic comparison.

Section D.2 Extends Q1 to logits vs. *ReLU activations* vs. probabilities, including `CelebA`.

Section D.3 Extends Q2 to probability-based CBMs, including `CelebA`.

Section D.4 Extends Q2 to *all* three available VLM-CBM vocabularies.

Section D.5 Evaluates AXp sizes for Selective CBMs (Schrodi et al., 2025).

D.1 EXTRA: ALGORITHMIC COMPARISON

We compare `EA+` against the original exact algorithm from (Marques-Silva, 2023), denoted `EA`, and the three heuristic algorithms – `TopK-Act`, `TopK-Wei`, and `TopK-Prod` – along several dimensions: explanation size, computational cost, and percentage of output explanations that are both faithful and subset-minimal (as per Definition 1). All tests are carried out on a random sample of 100 test instances, kept fixed across experiments.

Since CBMs with raw logits exhibit pathologic behavior, we focus on CBMs that model concepts with either probabilities ($\mathcal{S} \subseteq [0,1]^b$) and VLM-CBMs modeling them as cosine similarities ($\mathcal{S} = [-1,1]^b$), giving all algorithms a chance of producing compact explanations. `CelebA` has no categorical concepts that `glasso` can work with, so we only consider dense CBMs for this dataset.

D.1.1 EA+ OUTPUTS SMALLER AXPS THAN EA

		CBM				CBM + `glasso`			
		EA		EA+		EA		EA+	
DATASET	b	SIZE	CALLS	SIZE	CALLS	SIZE	CALLS	SIZE	CALLS
`CelebA`	39	20.8 ± 4.6	39	**12.6** ± 3.7	102	–	–	–	–
`Derm7pt`	28	17.1 ± 1.8	28	**8.3** ± 2.0	53	3.1 ± 2.9	28	**2.1** ± 0.8	33
`WBCAtt`	31	9.9 ± 3.6	31	**8.2** ± 3.0	60	2.6 ± 0.9	31	2.4 ± 0.8	37
`CUB`	312	169.5 ± 28.5	312	**46.3** ± 18.6	221	47.0 ± 9.9	312	**23.5** ± 3.6	134

Table 6: **Comparison between `EA` and our `EA+`** on regular CBMs with prob. activations. Bold indicates a size improvement of at least one activation.

		VLM-CBM				VLM-CBM + `glasso`			
		EA		EA+		EA		EA+	
DATASET	b	SIZE	CALLS	SIZE	CALLS	SIZE	CALLS	SIZE	CALLS
`LF-CBM`	202	201.0 ± 0.8	202	200.9 ± 0.9	633	36.9 ± 5.8	202	**35.3** ± 5.9	308
`VLG-CBM`	419	416.5 ± 1.5	419	416.3 ± 1.5	1286	78.0 ± 8.4	419	**66.5** ± 11.3	593

Table 7: **Comparison between `EA` and our `EA+`** on VLM-CBMs. Bold indicates a size improvement of at least one activation.

As a first experiment, we compare `EA` and `EA+` in terms of explanation size and number of oracle invocations, averaged across the 100 test instances. Both algorithms are exact in that they are guaranteed to always output (subset-minimal) AXps, but neither ensures these are size-minimal. The results on dense and `glasso`-sparsified CBMs are reported in Table 6 and those for VLM-CBMs in Table 7.

`EA+` always produces smaller AXps. Let us begin from explanation size. The tables indicate that `EA+` always produces smaller explanations compared to the original algorithm. This is expected considering it *i*) exploits information about the instance and the model to determine a ranking over activations, and *ii*) picks the smallest AXp for all three possible heuristic rankings, cf. Section A.

The size reduction is more noticeable for dense CBMs. E.g., for CUB, it drops from nearly 170 activations to about 46. The improvement is more limited for glasso-sparsified CBMs, especially on Derm7pt and WBCAtt. Owing to the benefits of structured sparsification, EA already performs very well. Still, on CUB under glasso, EA explanations cover about 47 activations, while EA+ only 24.

The size delta is minimal for dense VLM-CBMs, regardless of vocabulary, due to cosine similarity activations failing to properly constrain the adversary, while a gap forms for sparse VLM-CBMs, with improvements of about 1.5 and 11.5 activations depending on the chosen vocabulary.

EA+ makes less than $3\times$ oracle calls per run. Recall that EA+ runs each TopK* once, each time making a logarithmic number of oracle calls, whereas EA calls the oracle exactly b times with no early exit. The tables show that, in practice, EA tends to make fewer total calls in many cases – *e.g.*, dense CelebA/dense (39 vs. 102), Derm7pt/dense (28 vs. 53), WBCAtt/dense (31 vs 60), Derm7pt/sparse (28 vs. 33), WBCAtt (31 vs. 37). The worst offenders are the dense VLM-CBMs, where EA+ makes as much as three times the number of calls (202 vs. 633 and 419 vs. 1286). This is because concept activations are independent, meaning the SHRINK step has to test each of the activations covered by the candidate weak AXps individually. However, EA+ makes *fewer* calls in selected cases, and specifically for dense and sparse CBMs on CUB (312 vs. 221 and 312 vs. 134, respectively).

In short, in the worst case, EA+ can be more demanding than EA, but it typically does not pay the entire $3\times$ cost. In fact, while EA+ invokes TopK* three times – which helps with finding smaller AXps – thanks to binary search it can pay less than the full $3\times$ price. As we will show next, despite making more oracle calls, EA+ remains affordable in terms of wall-clock time.

D.1.2 EA+ IS COMPUTATIONALLY AFFORDABLE

Next, we compare the practical computational cost EA+ with that of the TopK heuristics. To this end, we measure wall-clock times and the number of oracle calls, averaged across the 100 test explanations. Recall that EA+ is guaranteed to output an AXp, while the TopK heuristics likely don't unless the budget k is sufficiently large. For fairness, we compare here EA+ against the TopK$^*_{bs}$ algorithm (Algorithm 4), which selects k automatically to *ensure* the output is a (weak) AXp and makes use of *binary search* to cut on the number of oracle calls. This gives it two substantial advantages over heuristics as used in the literature.

		CBM				CBM + glasso			
		TopK*		EA+		TopK*		EA+	
Dataset	b	Calls	Time	Calls	Time	Calls	Time	Calls	Time
CelebA	39	6	0.001	102	0.007	–	–	–	–
Derm7pt	28	6	0.004	53	0.025	6	0.004	33	0.016
WBCAtt	31	6	0.005	60	0.040	6	0.006	37	0.026
CUB	312	9	0.704	221	16.607	9	0.704	134	10.667

Table 8: **Runtime cost of all algorithms for regular CBMs**. Time is measured in seconds.

		VLM-CBM				VLM-CBM + glasso			
		TopK*		EA+		TopK*		EA+	
Dataset	b	Calls	Time	Calls	Time	Calls	Time	Calls	Time
LF-CBM	202	8	0.005	633	0.188	8	0.006	308	0.147
VLG-CBM	419	9	0.012	1286	1.095	9	0.020	593	1.215

Table 9: **Runtime cost of all algorithms for VLM-CBMs**. Time is measured in seconds.

The results for CBMs in Table 8 and those for VLM-CBMs in Table 9 tell the same story: EA+ makes far more calls to the oracle than the TopK$^*_{bs}$ algorithm, yet it remains affordable. The reason behind the extra calls is the checking phase: TopK$^*_{bs}$ only invokes the oracle a logarithmic number of times (thanks to binary search) for growing an explanation; on the other hand, like EA, EA+ invokes

it an additional linear number of times for ensuring subset minimality. However, we stress that, in the worst case, computing a (compact) AXp with EA+ takes 11s on CUB for dense CBMs. While slower, it is very much affordable. The benefit, as we will see momentarily, is that EA+ always outputs smaller explanations than the heuristics.

D.1.3 EA+ FINDS THE MOST COMPACT EXPLANATIONS

	CBM				CBM + glasso			
DATASET	ACT	WEI	PROD	EA+	ACT	WEI	PROD	EA+
CelebA	35.0 ± 3.8	15.7 ± 4.9	29.6 ± 5.1	**12.6 ± 3.7**	–	–	–	–
Derm7pt	9.5 ± 2.8	13.4 ± 5.0	9.7 ± 2.5	**8.3 ± 2.0**	7.7 ± 1.8	2.3 ± 0.7	2.2 ± 0.9	**2.1 ± 0.8**
WBCAtt	10.7 ± 4.3	17.5 ± 4.8	11.0 ± 4.9	**8.2 ± 3.0**	8.7 ± 3.2	3.5 ± 0.9	4.1 ± 1.3	**2.4 ± 0.8**
CUB	50.3 ± 19.0	84.7 ± 36.8	55.4 ± 21.8	**46.3 ± 18.6**	44.4 ± 14.6	30.0 ± 5.2	28.2 ± 4.3	**23.5 ± 3.6**

Table 10: **Comparison between explanation sizes** across datasets for regular CBMs.

	CBM				CBM + glasso			
DATASET	ACT	WEI	PROD	EA+	ACT	WEI	PROD	EA+
LF-CBM	202.0 ± 0.2	201.8 ± 0.5	201.8 ± 0.5	**200.9 ± 0.9**	201.3 ± 1.3	39.0 ± 6.8	39.1 ± 6.8	**35.3 ± 5.9**
VLG-CBM	418.6 ± 0.6	418.1 ± 1.1	418.2 ± 1.1	**416.3 ± 1.5**	418.0 ± 2.1	71.6 ± 12.2	71.7 ± 12.2	**66.5 ± 11.3**

Table 11: **Comparison between explanation sizes** for CUB using VLM-CBM vocabularies.

Next, we evaluate the average size of the explanations extracted by the various algorithms. We report the results for regular and glasso-sparsified CBMs in Table 10, marking the smallest average explanations for each model in bold. As expected, EA+– thanks to the shrink step (Algorithm 3) – *always* finds smaller explanations, and this holds for both dense and sparse CBMs.

Results for CBMs. For dense CBMs (Table 10, left), compared to the runner-up, EA+ achieves average size 12.6 (vs. 15.7 for the runner up, bottleneck size $b = 39$) in CelebA; 8.3 (vs. 9.7, $b = 28$) for Derm7pt; 8.2 (vs. 10.7, $b = 31$) for WBCAtt; and 46.3 (vs. 50.3, $b = 312$) for CUB. The difference is in-between 1.4 activations on average for Derm7pt and 9 for CUB. For glasso-sparsified CBMs (Table 10, right), the gap tightens, further supporting our usage of glasso to contrast Proposition 1. The only algorithm showing no benefit from glasso is TopK-Act, which does not directly depend on the learned weights. EA+ still outperforms all heuristics, but (compared to the runner-up) by less than one activation on average on Derm7pt, about 1 activation in WBCAtt, and 5 activations on CUB.

Results for VLM-CBMs. We report additional results on dense and glasso-sparsified VLM-CBMs in Table 11. Recall that the LF-CBM vocabulary comprises 202 activations and the VLG-CBM one 419. Here, concept activations are cosine similarities, which explains the large explanations output by all algorithms – and the small gaps between them – when using dense inference layers (Table 11, left): this is yet another instance of Proposition 1 and it is consistent with the poor results reported for cosine similarity activations in Section 5.2. The picture is more favorable for sparse CBMs, with explanation sizes dropping sharply for all algorithms (except TopK-Act, as above). Again, EA+ is the winner across the board, with an advantage of 4 activations on average for the LF-CBM vocabulary and of 5 activations for the VLG-CBM one.

Summary. While EA+ provides a modest advantage, size-wise, over the competitors, it is by far the most reliable alternative, as there is no obvious ranking between heuristics, barring poor interactions between TopK-Act with sparsification.

D.1.4 EA+ ALWAYS FINDS AXPS, HEURISTICS VERY OFTEN DO NOT

We evaluate whether the algorithms are capable of computing explanations that are both faithful (sufficient to defend the prediction) *and* subset-minimal, as required by Definition 1. Since all algorithms, including our TopK heuristics, always return a *weak* AXp, as explained in Section A, we only need to check how frequently these are also subset-minimal.

Dataset	TopK Act	TopK Wei	TopK Prod	EA+
CelebA	0%	10%	1%	**100%**
Derm7pt	37%	2%	37%	**100%**
WBCAtt	9%	0%	12%	**100%**
CUB	5%	1%	5%	**100%**
Average	12.8%	3.5%	13.8%	**100%**

Dataset	TopK Act	TopK Wei	TopK Prod	EA+
CelebA	–	–	–	–
Derm7pt	0%	81%	85%	**100%**
WBCAtt	0%	18%	22%	**100%**
CUB	0%	1%	1%	**100%**
Average	0%	33.4%	36%	**100%**

Table 12: **Whereas EA+ always computes AXps, heuristics do not**. **Left**: dense CBMs. **Right**: glasso-sparsified CBMs.

We compute the *percentage of subset-minimal explanations* across 100 test instances and report the results for regular CBMs in Table 12. The table indicates that, as expected, EA+ retrieves subset-minimal AXps for every instance, model and dataset. The heuristics, on the other hand, never achieve 100% subset-minimality. For dense CBMs, they attain at best 37% subset-minimality (on Derm7pt with TopK-Act and TopK-Prod), but the percentage can go as low as 0% (on CelebA with TopK-Act). The weight-based heuristics, TopK-Wei and TopK-Prod, fare better on glasso-sparsified CBMs, in that they manage to achieve high percentage of subset-minimality on Derm7pt (up to 85%); yet they still fail catastrophically on CUB, where the best heuristic attains 1% subset-minimality. In conclusion, these results demonstrate the advantages of EA+ over common heuristics.

D.2 Extended Q1: Logits vs. ReLUs vs. Probabilities

We report a comparison of all common concept activation modeling strategies – logits, ReLU activations, probabilities – in Fig. 3.

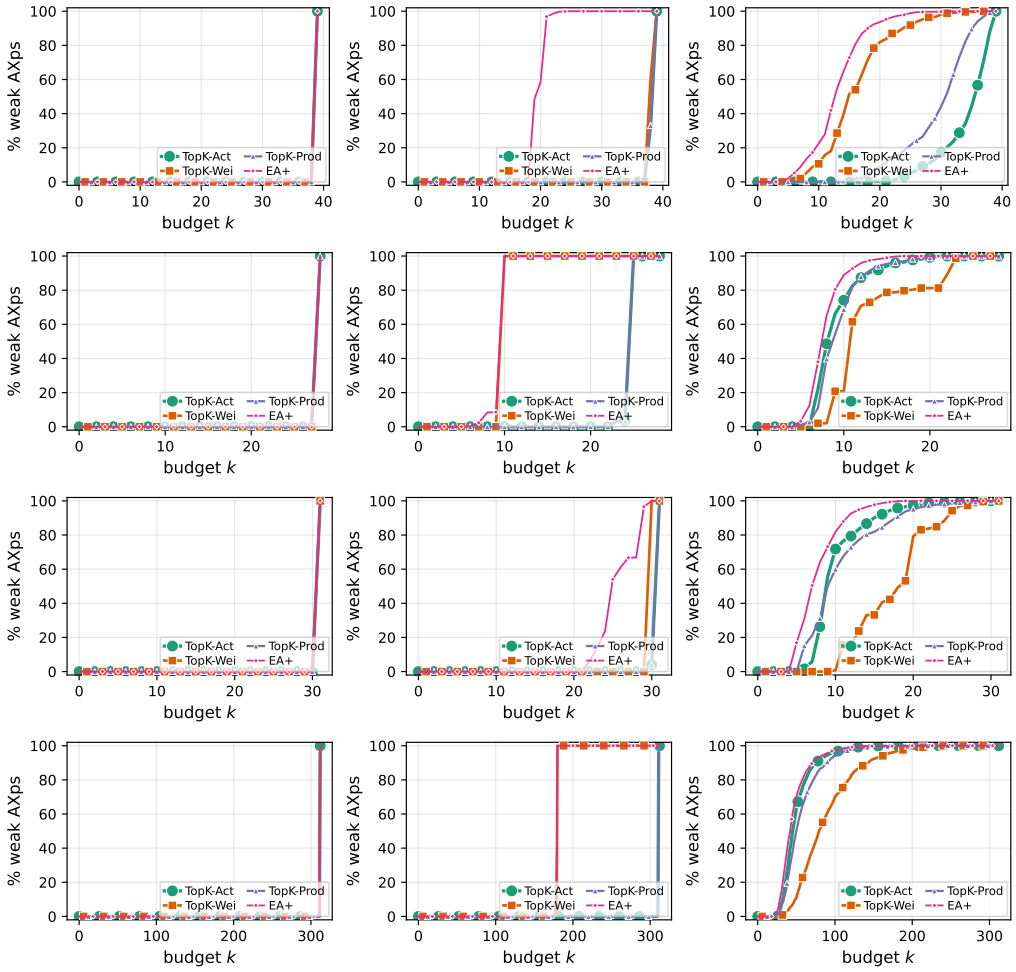


Figure 3: **Modeling concepts as probabilities shrinks** AXps **the most** on all datasets. From the top: `CelebA`, `Derm7pt`, `WBCAtt`, `CUB`). Left to right: logits, ReLU, probabilities.

ReLUs sit in-between logits and probabilities. All datasets display the same trend: logits ($\mathcal{C} = (-\infty, +\infty)^b$) are pathologic, probabilities ($\mathcal{C} \subseteq [0,1]^b$) enable more compact AXps for all algorithms, and ReLU activations ($\mathcal{C} = [0, \infty)^b$) sit in-between, as shown by the curves in the middle row. This is perfectly in line with Proposition 1.

`TopK-Act` and `TopK-Prod` underperform on `CelebA`. This is evident when comparing the curves of the heuristic algorithms in `CelebA` (top row) and on the other datasets (other rows). The reason is simple. This is because `CelebA` only has binary concepts, which means $\mathcal{S} = [0,1]^b$, with no simplex structure, leaving the adversary less constrained than in other datasets. This directly penalizes heuristics focusing on the highest activations.

D.3 Extended Q2: enet vs. glasso for Probabilities

Fig. 4 reports a complete comparison of `enet` vs. `glasso` sparsification for all datasets with categorical concepts, as these are needed to define the groups used by `glasso`. These include `Derm7pt`, `WBCAtt`, and `CUB`, but not `CelebA`. As in Section 5.2, we focus on CBMs with probabilistic activations, which are much better behaved than CBMs with logit activations (as per Section 5.1).

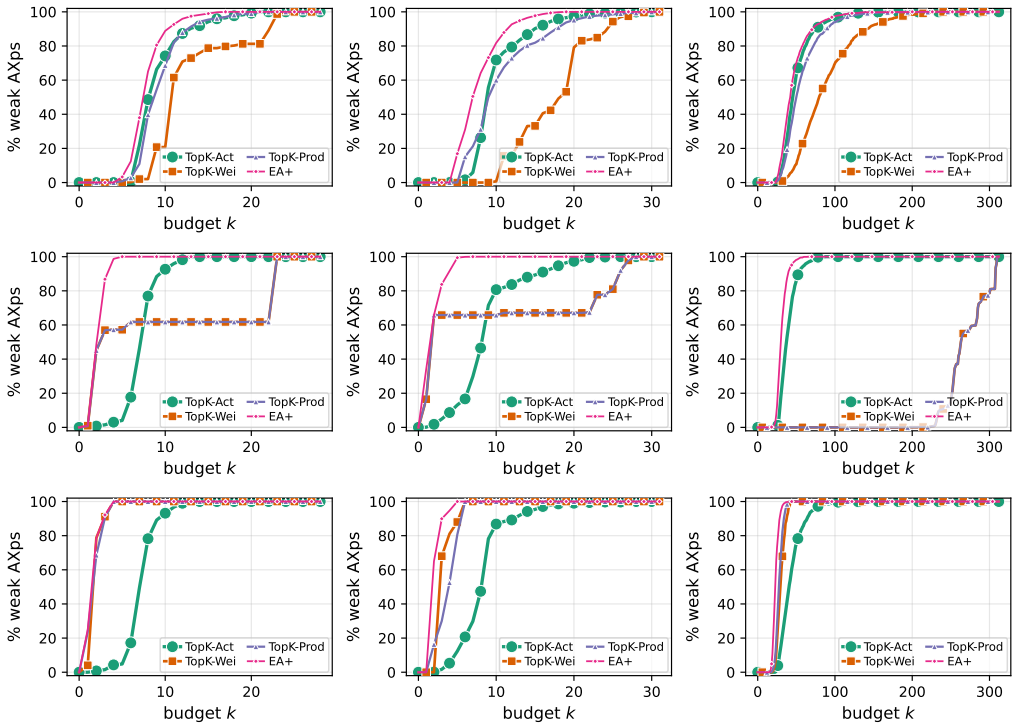


Figure 4: **`enet` (middle) works partially, `glasso` (below) works consistently.**

`enet` partially works, `glasso` is more consistent. The results are compatible with those reported in Section 5.2 for `CUB`: while `enet` primarily helps `EA+`, it makes some heuristics – `TopK-Wei` and `TopK-Prod`– misbehave as it learns larger non-zero weights. On the other hand, `glasso` helps more consistently across algorithms. At the same time, `TopK-Act` does not benefit from sparsification as much for either penalty.

D.4 Extended Q2: Sparsity on VLM-CBMs

Here, we report the complete FCCs for all VLM-CBM vocabularies. Recall that concept activations are modelled as cosine similarities.

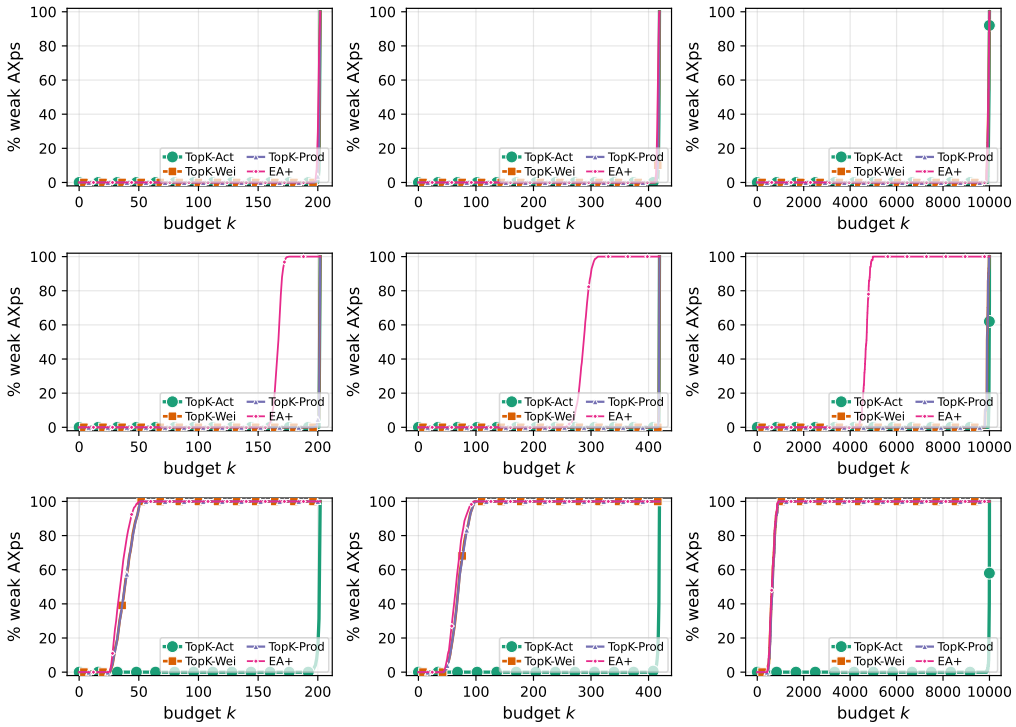


Figure 5: **Q3: For VLM-CBMs, `glasso` outperforms other sparsification strategies**, as shown here with `CUB`. Each VLM-CBM employs its own LLM-generated vocabulary, with `LaBo`'s sporting $10,000$ activations.

`glasso` outperforms `enet` on all vocabularies. The FCCs in Fig. 5 – dense above, `enet` in the middle, and `glasso` at the bottom – indicate a large margin in the percentage of explained examples admitting a smaller weak AXp. As before, despite employing bounded concept activations, dense VLM-CBMs display pathologic behavior, precisely because their weights are dense. Introducing `enet` (middle row) helps: focusing on `EA+` for `LF-CBM`, about 80% explained instances admit (weak) AXps of size $\sim$ 160 (where the full bottleneck holds 202 activations); for `VLG-CBM`, of size $\sim$ 300 (compared to 419); for `LaBo`, of size $\sim$ 5,000 (compared to 10,000). The heuristic algorithm, however, do not see any size improvement, backing up our observations in Section 5.2. In stark contrast, `glasso` dramatically shrinks explanations of all algorithms except `TopK-Act` (again, as noted in Section 5.2).

VLM-CBMs resist compactly faithful explanations. Despite the substantial improvements given by `glasso`, faithful explanations of size 100 (for `VLG-CBM`) and up to 500 (for `LaBo`) are unlikely to be human interpretable. This supports our key message: large concept bottlenecks should not be (always, at least) equated with reliable interpretability. We discuss possible ways to overcome this roadblock in Section 7 and leave an extensive evaluation to future work.

D.5 EXTRA: SELECTIVECBM

Compared to standard CBMs, Selective CBMs (Schrodi et al., 2025) sparsify both concept activations (per-input) and weights (across inputs) by combining an instance-dependent gating function with an elastic net penalty.

To assess the benefits of this double sparsification strategy, we compare Selective CBMs against: *i*) a baseline, namely dense CBMs with ReLU concept activations; and *ii*) sparsified CBMs (`enet` for `CelebA` and `glasso` for `WBCAtt`) with logit activations. We apply them to two datasets, `CelebA` (no categorical concepts, *i.e.*, `glasso` is unnecessary) and `WBCAtt` (several categorical concepts, *i.e.*, `glasso` is useful). As usual, we maximize λ subject to the constraint that F_1^Y is at most 10% worse than the dense baseline, for fairness.

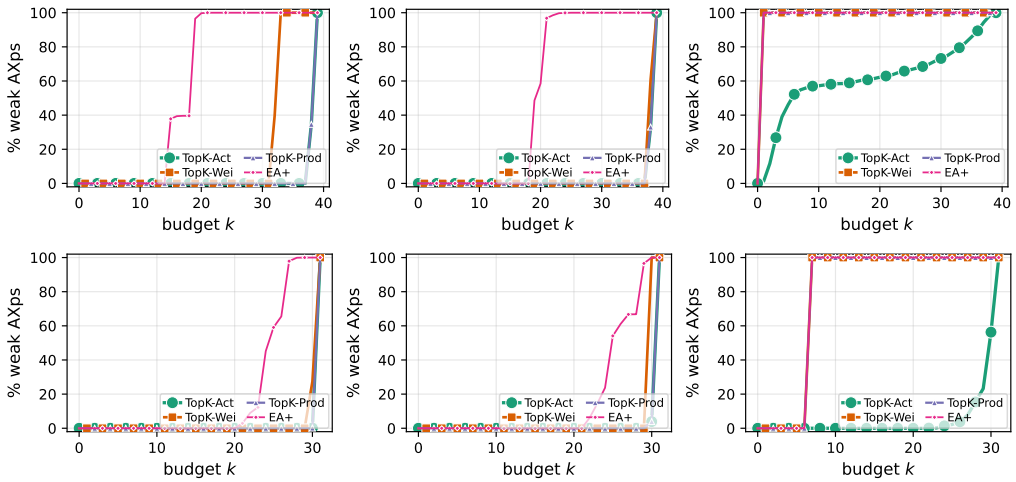


Figure 6: **Selective CBMs behave like ReLU CBMs and worse than sparsified CBMs** in terms of AXp size, shown here for `CelebA` (top) and `WBCAtt` (bottom).

Selective CBMs behave similarly to ReLU CBMs. Fig. 6 shows that Selective CBMs (left) behave similarly to ReLU CBMs (middle), *i.e.*, they provide a moderate improvement to AXp size compared to regular CBMs (which are pathological, as shown in Section 5.1). The improvement applies chiefly to `EA+` (in **pink**), whereas all heuristics continue to behave poorly. However, like ReLU CBMs, Selective CBMs underperform compared to appropriately sparsified CBMs even when these employ raw logits (right). This is most visible for `EA+`, `TopK-Wei`, and `TopK-Prod`, which all display the same trend, whereas `TopK-Act` lags behind by a fair amount.

These results support our arguments in Sections B and 4.1 that, from the perspective of AXps, SelectiveCBM's learnable gating function is equivalent to a pure ReLU: neither prevents the construction of counterexamples whenever the prediction can be flipped by pushing a concept activation s_j to $+\infty$. The only difference is the threshold for constructing the counterexample: for ReLUs it is 0, for Selective CBMs it is a possibly non-zero positive constant (*i.e.*, the learned gating threshold).