ARCHETYPES OF A CONCEPT: SAMPLE-TRACEABLE DECOMPOSITION OF CONCEPT REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept-direction methods summarize a concept with a single vector. This representation is compact, but it obscures variation within the concept and the training examples that support that variation. We introduce **Archetypes of a Concept (AoC)**, a post-hoc method that decomposes concept-positive features into simplex-constrained archetypes. Each archetype is a convex combination of training features and therefore retains explicit sample-level provenance. A fixed concept probe then selects aligned archetypes. New queries are represented by simplex coefficients over the selected dictionary. We establish convex-hull traceability and a margin-based condition for preserving fixed-probe predictions. We also analyze coefficient continuity and selection stability under explicit regularity and score-gap conditions. Together, these properties support evidence inspection and coefficient-space analysis in a frozen feature space.

1 INTRODUCTION

Concept-based explanations seek representations that are easier to inspect than individual input features or opaque hidden units (Doshi-Velez & Kim, 2017; Poeta et al., 2023). A common post-hoc approach represents a concept by a linear direction in a frozen feature space. TCAV, for example, uses a concept activation vector to test how a model output responds along that direction (Kim et al., 2018). The direction provides a useful concept anchor. However, it compresses all positive instances into one axis and does not expose the internal structure of the positive region.

This compression matters when the same concept appears in several visual forms. A color attribute may occur on different object parts, under different viewpoints, or together with different backgrounds. Two samples can therefore receive similar concept scores while relying on visibly different evidence. A single direction records that both samples align with the concept, but it does not describe the modes of variation that produced the alignment. It also provides no direct path from a discovered mode back to the training samples that define it.

We study the following problem. The inputs are a frozen encoder, a binary linear concept probe, and concept-positive training features. The goal is to construct a decomposition with three properties. First, each component should expose the training samples that support it. Second, the components should remain connected to the original probe. Third, the fitted representation should extend to new queries through a fixed-dimensional coefficient vector.

AoC addresses these requirements in the encoder's feature space. It first decomposes the positive region using archetypal analysis. It then screens the learned archetypes with the concept-probe direction. Finally, it represents a query as a convex mixture of the selected archetypes. These stages have distinct roles: the decomposition captures variation, the probe supplies concept conditioning, and coefficient inference supports query-level comparison.

Our approach combines a fixed concept probe with archetypal analysis (Cutler & Breiman, 1994). The archetypes are constrained to the convex hull of concept-positive training features. Consequently, every archetype can be expanded as a vector of nonnegative training-sample weights. This expansion supplies a concrete provenance record in feature space. A probe-based screening step retains archetypes aligned with the original concept direction. Simplex-constrained inference then maps a new query to coefficients over the selected archetypes.

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 summarizes the interface. It supports two complementary views. *Archetype evidence inspection* displays the largest training weights in a row of B. This view identifies the training samples that contribute most strongly to an archetype. *Coefficient-space organization* compares queries through their inferred coefficient vectors. This view identifies queries that use similar mixtures of the selected archetypes.

Our contributions are threefold:

1. We formulate concept-conditioned decomposition as simplex-constrained archetypal analysis over concept-positive features, giving every archetype explicit training-sample provenance.
2. We introduce probe-based archetype screening and simplex coefficient inference, producing a fixed-dimensional representation for new queries while preserving the link to the original concept direction.
3. We characterize the interface through convex-hull traceability and fixed-probe prediction preservation, with complementary continuity and selection-stability results.

Our analysis focuses on the provenance and stability properties of a fitted dictionary in a frozen feature space. Provenance follows from the simplex geometry. Stability additionally depends on probe margins, curvature, or score gaps. We state these conditions separately so that each result has a clear operational interpretation.

2 Related Work

Concept directions and bottlenecks. TCAV represents a concept by a direction in an internal feature space and measures directional task sensitivity (Kim et al., 2018). Its central object is a single separating direction. AoC uses the same type of direction as an anchor, but studies structure within the positive side of the concept. The two representations therefore answer different questions. A concept direction asks whether and how strongly a feature aligns with a concept. AoC asks which sample-grounded modes occur within the aligned region and how a query combines them.

Concept bottleneck models expose concept predictions as an intermediate representation (Koh et al., 2020). Probabilistic, leakage-aware, post-hoc, and label-free variants extend this framework to different supervision settings (Kim et al., 2023; Havasi et al., 2022; Yuksekgonul et al., 2022; Oikarinen et al., 2023). These models typically organize prediction around a vocabulary of concepts. AoC instead takes one binary concept at a time and refines its positive feature region. It is compatible with a pre-existing probe or concept bank and leaves the encoder unchanged.

Concept discovery and representation decomposition. ACE discovers concepts by clustering local image regions (Ghorbani et al., 2019). CRAFT recursively decomposes nonnegative activations and evaluates component importance (Fel et al., 2023). Non-negative concept activation vectors, sparse autoencoders, and sparse linear concept embeddings provide related factorized views of learned representations (Zhang et al., 2021; Huben et al., 2024; Bhalla et al., 2024). These approaches differ in their units of analysis, constraints, and supervision. Some operate on image patches, while others learn activation dictionaries or align features with a textual vocabulary.

AoC is distinguished by two coupled constraints. The dictionary is fitted only on labeled concept-positive features, and every archetype remains inside their convex hull. The first condition makes the decomposition concept-specific. The second preserves an explicit expansion over observed training features. Probe screening is applied after the decomposition. This separates discovery of positive-region variation from selection by the discriminative concept direction.

Sample provenance and archetypal structure. Prototype-based models associate predictions with representative training patterns (Chen et al., 2019). Influence-based methods trace predictions to observations through their effect on model parameters (Koh & Liang, 2017). AoC provides a different form of traceability. Its weights describe how each archetype is constructed in the frozen feature space. They are part of the representation itself rather than an estimate of a training point's influence on a prediction.

Classical archetypal analysis supplies the relevant geometry (Cutler & Breiman, 1994): every archetype is a convex combination of observations. Archetypal SAE similarly constrains dictionary

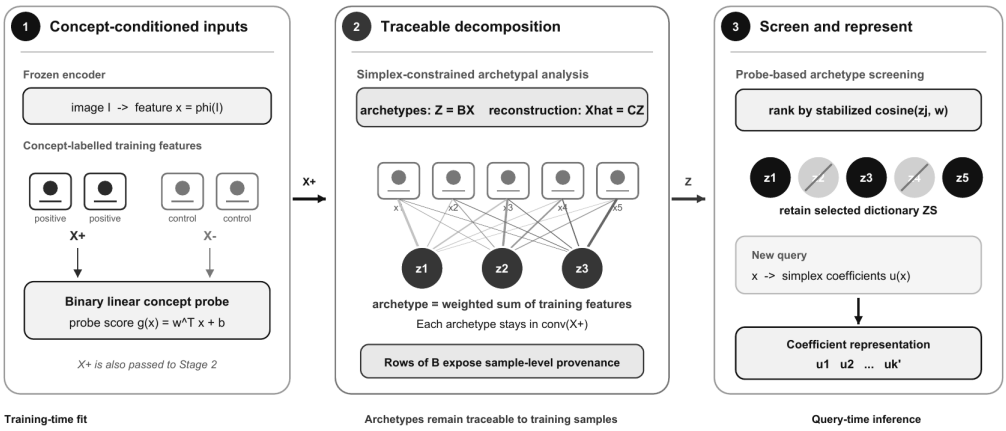


Figure 1: **Overview of AoC.** Concept-labelled features define a linear concept probe. Simplex-constrained archetypal analysis decomposes positive features into sample-traceable archetypes. Probe-based screening retains concept-aligned archetypes, and a new query is represented by simplex coefficients over the selected dictionary.

Table 1: Core notation for AoC.

Symbol	Meaning
$X \in \mathbb{R}^{n \times d}$	Concept-positive training features
$B \in \mathbb{R}^{k \times n}$	Training-sample weights used to construct archetypes
$Z = BX$	Matrix of k archetypes
$C \in \mathbb{R}^{n \times k}$	Training-sample coefficients over the archetypes
$Z_S \in \mathbb{R}^{k' \times d}$	Probe-screened archetype dictionary
$u(x) \in \Delta_{k'}$	Query coefficients over the selected dictionary

atoms to a data convex hull and studies dictionary stability (Fel et al., 2025). AoC conditions this geometry on one binary concept. It fits only the labeled positive region, connects the dictionary to a fixed probe, and adds query-wise simplex inference over the screened archetypes.

AoC thus combines three ideas that are usually treated separately: concept anchoring, sample-weight provenance, and query-level coefficient inference. The result is a post-hoc interface that preserves the original encoder and probe while adding a structured view of within-concept variation.

3 Methodology

AoC takes as input a fixed encoder, concept-positive training examples, and either an existing binary linear concept probe or the positive and negative examples required to train one. It returns a probe-screened archetype dictionary, sample-level evidence weights for each archetype, and a coefficient representation for new queries.

Table 1 summarizes the objects used throughout the method. The notation separates three levels of representation. The matrix X contains observed training features. The matrix Z contains learned archetypes derived from those features. The vector $u(x)$ describes a new query relative to the selected archetypes. Keeping these levels distinct makes it easier to separate training-time provenance from query-time inference.

3.1 Problem Definition

Given an input space $\mathcal{I}$, a fixed encoder $\phi : \mathcal{I} \to \mathbb{R}^d$ maps an input I to $x = \phi(I)$. Feature normalization, when used, is applied before probe training and archetypal decomposition. Throughout

the paper, x denotes the resulting preprocessed feature. The encoder and preprocessing map remain fixed while fitting and using AoC.

Concept source. For a binary concept c, we assume one of two inputs. The first is a previously trained linear probe $g_c(x) = w_c^\top x + b_c$. The second is a collection of positive and negative examples from which such a probe can be trained. In the latter case, an indicator $m_c(I) \in \{0, 1\}$ defines $\mathcal{X}_c^+ = \{\phi(I) : m_c(I) = 1\}$ and $\mathcal{X}_c^- = \{\phi(I) : m_c(I) = 0\}$. AoC models variation within $\mathcal{X}_c^+$. The negative set is used to determine which directions in feature space distinguish the concept from its controls.

Concept activation vector. In case (ii), a linear concept probe distinguishes $\mathcal{X}_c^+$ from $\mathcal{X}_c^-$:

$$g_c(x) = w_c^\top x + b_c \tag{1}$$

where $w_c \in \mathbb{R}^d$ is the concept activation vector (CAV). AoC uses this direction after fitting the positive-region decomposition. This ordering is deliberate. The decomposition first captures variation among positive samples without forcing every archetype to align with the probe. Screening then identifies the subset most closely associated with the discriminative concept direction.

Why decompose only the positive region? The objective is to resolve modes within a known concept, rather than to relearn the binary decision boundary. Mixing positive and control samples in the archetypal dictionary would allow the reconstruction objective to allocate components to the contrast between classes. By fitting Z only on positive samples, the available archetypes instead describe variation that occurs after the concept label is fixed. The probe still uses both positive and control samples, so the later screening step retains access to discriminative information.

3.2 Concept Vector Extraction

We obtain the concept vector by training a linear probe in the fixed feature space of encoder ϕ to distinguish positive concept samples from negative or control samples:

$$f(x) = Wx + b \in \mathbb{R}^2, \quad W \in \mathbb{R}^{2\times d},\ b \in \mathbb{R}^2 \tag{2}$$

The softmax probabilities are denoted by $p(x) = \mathrm{softmax}(f(x))$. We optimize the probe with cross-entropy. Samples in $\mathcal{X}_c^+$ receive label 1, while samples in $\mathcal{X}_c^-$ receive label 0.

The effective separation vector is

$$w_c := W_{1,:} - W_{0,:}, \qquad b_c := b_1 - b_0,$$

because $f_1(x) - f_0(x) = w_c^\top x + b_c$. The difference between the two logit rows is the effective binary separation direction. Using this difference also removes any common shift shared by the two rows of W. The same w_c is used for screening and for the fixed-probe analysis. Screening requires $w_c \neq 0$.

The intercept b_c determines the location of the binary decision boundary. The direction w_c, by contrast, determines how the probe score changes as a feature moves through the representation space. AoC uses w_c for archetype ranking because the ranking is intended to measure directional alignment. The full affine score g_c, including b_c, is retained in the prediction-preservation analysis.

3.3 Concept Decomposition via Archetypal Analysis

We collect the positive training features as $X = [x_1^\top; \cdots; x_n^\top] \in \mathbb{R}^{n\times d}$, where each $x_i \in \mathcal{X}_c^+$. From these features, AoC learns k archetypes. Two sets of coefficients play different roles. The rows of B construct the archetypes from training samples, whereas the rows of C reconstruct training samples from archetypes.

To model the composition of each sample, we adopt a decomposition formulation with convex combination constraints:

$$Z = BX, \quad \hat{X} = CZ = CBX \tag{3}$$

Here $Z = [z_1^\top; \cdots; z_k^\top] \in \mathbb{R}^{k\times d}$ is the archetype matrix. The matrix $B \in \mathbb{R}^{k\times n}$ contains archetype-to-training weights. The matrix $C \in \mathbb{R}^{n\times k}$ contains the training coefficients, and $\hat{X}$ contains the

reconstructed features. We impose row-wise simplex constraints:

$$B_{j,:} \in \Delta_n, \quad C_{i,:} \in \Delta_k,$$
$$\Delta_q := \left\{ u \in \mathbb{R}^q : \; u \geq 0, \; \sum_{r=1}^{q} u_r = 1 \right\} \tag{4}$$

Here $u \geq 0$ is coordinatewise. The constraint on B makes every archetype z_j a convex combination of positive training features. The constraint on C makes every reconstruction $\hat{x}_i$ a convex combination of archetypes. Thus, $B_{j,:}$ answers how archetype j is grounded in the training set. In contrast, $C_{i,:}$ answers how training feature x_i is expressed in the learned dictionary.

The simplex geometry serves two purposes. Nonnegativity prevents cancellation between large positive and negative weights, which would make sample contributions difficult to inspect. The unit-sum constraint fixes the overall scale of each mixture. Together, these constraints also keep every archetype and reconstruction inside the convex hull of the positive features, leading to the provenance result in Section 4.1.

For angular feature spaces such as CLIP (Radford et al., 2021), the canonical AoC objective is the stabilized cosine reconstruction loss

$$L_{\mathrm{rec}}(X, \hat{X}) = 1 - \frac{1}{n} \sum_{i=1}^{n} \cos_\varepsilon(\hat{x}_i, x_i), \quad \varepsilon > 0,$$
$$\cos_\varepsilon(u, v) = \frac{u^\top v}{(\|u\|_2 + \varepsilon)\|v\|_2}, \qquad v \neq 0. \tag{5}$$

The fitting problem is

$$\min_{B,C} \; L_{\mathrm{rec}}(X, CBX) \quad \text{s.t.} \quad B_{j,:} \in \Delta_n, \; C_{i,:} \in \Delta_k. \tag{6}$$

The same loss definition is used for query inference. During fitting, the objective encourages each positive feature to be reconstructed by a mixture of archetypes with a similar direction. The optimization is nonconvex, so different initializations can permute the archetypes or produce alternative dictionaries with comparable reconstruction loss. The provenance statement applies to every feasible solution, while comparisons between separate fits require component matching.

For visualization, fix an integer $1 \leq m \leq n$. The *top-m evidence set* of archetype j contains the indices of the m largest entries of $B_{j,:}$. Exact ties are broken by ascending sample index. The resulting set gives a compact view of the samples with the largest contributions. The full row of B remains available when a dense provenance profile is needed.

Positive samples can contain background structure and attributes that co-occur with the target concept. Reconstruction alone may therefore allocate archetypes to factors that explain the positive data but align weakly with the probe. The screening stage uses the original concept direction to rank the fitted archetypes before query inference.

3.4 Concept Screening and Inference

For $\varepsilon_s > 0$, define the screening score

$$s_j = \frac{z_j^\top w_c}{(\|z_j\|_2 + \varepsilon_s)(\|w_c\|_2 + \varepsilon_s)}. \tag{7}$$

Let S contain the indices of the k' largest scores, where $1 \leq k' < k$. Exact ties are broken by ascending archetype index. The selected dictionary is denoted by $Z_S \in \mathbb{R}^{k' \times d}$. The score measures stabilized angular alignment with w_c; it does not use the intercept b_c. Decomposition and screening therefore use complementary information. Decomposition models the geometry of positive samples, while screening uses the discriminative direction learned from positive and negative examples.

For a nonzero query feature x, infer coefficients over the fixed selected dictionary:

$$u(x) \in \arg\min_{u \in \Delta_{k'}} \left\{ \ell(u^\top Z_S, x) + \tfrac{\tau}{2}\|u\|_2^2 \right\}, \qquad \tau \geq 0. \tag{8}$$

where $\ell(y, x) = 1 - \cos_\varepsilon(y, x)$. The coefficient vector lies on the simplex, so its entries are nonnegative and sum to one. A positive τ improves numerical regularity and can promote a unique solution under the conditions in Section 4. The inferred vector $u(x)$ is specific to the query. It is distinct from $C_{i,:}$, which was optimized jointly with the archetypes for a training feature.

AoC procedure.

1. Train or obtain the fixed probe $g_c(x) = w_c^\top x + b_c$.
2. Fit B, C on positive training features and form $Z = BX$.
3. Rank archetypes by s_j and retain the selected dictionary Z_S.
4. For a query x, infer $u(x)$ on $\Delta_{k'}$.
5. Inspect top-m training evidence through B, or compare queries through their coefficient vectors $u(x)$.

The procedure produces two outputs with different interpretations. The rows of B connect archetypes to training evidence and are fixed after fitting. The vector $u(x)$ connects a query to the selected dictionary and is inferred for each new input. Evidence inspection should therefore use B, whereas query comparison should use $u(x)$. Keeping these outputs separate avoids treating a query coefficient as if it were a direct training-sample attribution.

3.5 Practical Interpretation of AoC Outputs

AoC produces two complementary outputs. The row $B_{j,:}$ records how archetype z_j is constructed from concept-positive training features. Its largest entries identify the samples contributing most strongly to that fitted archetype. This is construction provenance rather than a causal influence score or proof that a training sample uniquely determines a model prediction.

The query coefficient vector $u(x)$ has a different interpretation. It describes a new feature relative to the selected archetype dictionary. Queries with similar coefficients use similar mixtures of archetypes, allowing within-concept comparison and retrieval. Unlike B, which is fixed after dictionary learning, $u(x)$ is inferred separately for each query and should not be treated as a direct attribution over training samples.

The decomposition is not generally identifiable. Different initializations may permute the archetypes or produce alternative convex decompositions with similar reconstruction losses. Convex-hull traceability still holds for each fitted solution, but the interpretation of an individual archetype should be supported by its probe-alignment score, highest-weight training samples, and representative query coefficients. Agreement among these views provides stronger evidence for an interpretable within-concept mode than any single output alone.

4 Theoretical Analysis

We analyze four properties of the fixed-dictionary interface. The first two concern the geometry of the representation: sample provenance and preservation of a fixed probe prediction. The remaining two concern local stability: continuity of coefficient inference and stability of the selected archetype set under bounded score perturbations. Each result addresses a different stage of the pipeline in Figure 1.

4.1 Convex-Hull Provenance

Theorem 4.1 (Convex-Hull Traceability)**.** *Let $X = [x_1^\top; \cdots; x_n^\top] \in \mathbb{R}^{n \times d}$ contain the positive sample features. If $B_{j,:} \in \Delta_n$ for all j and $C_{i,:} \in \Delta_k$ for all i, then every archetype and every reconstruction remains in their convex hull:*

$$z_j \in \mathrm{conv}\{x_1, \cdots, x_n\} \quad \forall j, \qquad \hat{x}_i \in \mathrm{conv}\{x_1, \cdots, x_n\} \quad \forall i. \tag{9}$$

where $\mathrm{conv}(\cdot)$ denotes the convex hull.

The proof is provided in Appendix C. The result gives a direct interpretation of the rows of B. An archetype cannot be created from an arbitrary point in feature space; it must be assembled from

observed positive features. The same statement propagates through C, so fitted reconstructions remain in the same convex hull.

The affine probe also interacts cleanly with convex combinations. For any simplex weights p,

$$g_c\left(\sum_t p_t x_t\right) = \sum_t p_t g_c(x_t). \tag{10}$$

Suppose every positive training feature satisfies $g_c(x_t) \geq m > 0$. Equation (10) then implies that every point in their convex hull also has probe score at least m. In this case, the entire fitted positive-region representation remains on the positive side of the probe with the same lower margin. A query outside this region requires the pointwise analysis below.

Corollary 4.2 (Norm Bound and Boundedness)**.** *If all positive samples satisfy $\|x_t\|_2 \leq R$, then for any j, i:*

$$\|z_j\|_2 \leq R, \qquad \|\hat{x}_i\|_2 \leq R \tag{11}$$

The corollary shows that the convex mixtures cannot have a larger norm than the largest positive feature. This boundedness property is useful when studying score perturbations and numerical behavior. Appendix D provides the proof.

4.2 Fixed-Probe Prediction Preservation

Theorem 4.3 (Sufficient condition for fixed-probe prediction preservation)**.** *Fix $g_c(x) = w_c^\top x + b_c$ with $\|w_c\|_2 > 0$, and define $\gamma_c(x) = |g_c(x)|$. If a reconstruction $\hat{x}$ satisfies*

$$\|\hat{x} - x\|_2 < \frac{\gamma_c(x)}{\|w_c\|_2}, \tag{12}$$

then $g_c(\hat{x})$ and $g_c(x)$ have the same sign.

Appendix A proves the result using Cauchy–Schwarz. The ratio $\gamma_c(x)/\|w_c\|_2$ is the Euclidean distance from x to the probe boundary. The theorem therefore has a simple geometric interpretation: reconstruction preserves the binary prediction whenever the reconstruction error is smaller than the available distance to the boundary. Samples with a large probe margin tolerate more error than samples close to the boundary.

4.3 Conditional Coefficient Continuity

We next ask how the inferred coefficients change when the query feature changes. Fix Z_S and define $J(u; x) = \ell(u^\top Z_S, x) + \tau\|u\|_2^2/2$. Let $K \subseteq \Delta_{k'}$ be a nonempty fixed compact convex set, and let N be a neighborhood of query features. Two standard regularity conditions make the sensitivity statement precise.

Assumption 4.4 (Uniform parameter regularity)**.** For all $u \in K$ and $x, x' \in N$, J is differentiable in u, continuous on K, and

$$\|\nabla_u J(u; x) - \nabla_u J(u; x')\|_2 \leq L\|x - x'\|_2. \tag{13}$$

Assumption 4.5 (Uniform strong convexity)**.** For every $x \in N$, $J(\cdot; x)$ is μ-strongly convex on K with the same $\mu > 0$:

$$\langle \nabla_u J(u; x) - \nabla_u J(v; x), u - v\rangle \geq \mu\|u - v\|_2^2. \tag{14}$$

Theorem 4.6 (Conditional coefficient continuity)**.** *Under Assumptions 4.4–4.5, $u_K(x) = \arg\min_{u \in K} J(u; x)$ exists uniquely and*

$$\|u_K(x) - u_K(x')\|_2 \leq (L/\mu)\|x - x'\|_2, \qquad x, x' \in N. \tag{15}$$

Appendix B gives the variational-inequality proof. The bound separates two effects. The constant L measures how strongly the coefficient gradient responds to a change in the query. The constant μ measures how strongly the objective identifies its minimizing coefficient vector. A smaller ratio L/μ therefore corresponds to a less sensitive representation. For squared-error inference, $\tau > 0$ supplies strong convexity. Cosine inference uses the stated curvature assumption.

4.4 SCREENING-SET STABILITY

Lemma 4.7 (Screening-gap condition)**.** *Let S be the top-k' set under scores s_j. Suppose component-matched perturbed scores satisfy $|s'_j - s_j| \le \eta$ for all j. If the original descending score gap satisfies*

$$s_{(k')} - s_{(k'+1)} > 2\eta,$$

then the selected set is unchanged.

The condition compares the score perturbation with the gap at the selection boundary. If the gap is larger than twice the maximum perturbation, a selected archetype cannot exchange order with an unselected one. Formally, for $i \in S$ and $j \notin S$, $s'_i - s'_j \ge s_i - s_j - 2\eta > 0$. Across independently fitted dictionaries, the same comparison is applied after matching corresponding components.

4.5 SCOPE OF THE ANALYSIS

The four results answer progressively different questions. Convex-hull membership establishes where an archetype comes from. The margin condition determines when reconstruction preserves a probe decision. The continuity theorem controls the response of coefficients to a query perturbation, and the score-gap lemma controls the discrete screening decision. The latter two results are conditional on curvature, gradient regularity, component matching, and a nonzero score gap. Human-semantic interpretation is evaluated empirically rather than inferred from these geometric properties.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING AND EVALUATION ENDPOINT

The reported experiments cover two forms of visual concepts. CUB provides fine-grained bird attributes (Wah et al., 2011), while DTD provides texture categories (Cimpoi et al., 2014). This pairing probes whether the representation behaves similarly for localized object attributes and distributed texture patterns. Features come from fixed CLIP-RN50 and CLIP-ViT-B/16 encoders (Radford et al., 2021), together with EfficientNet-B0 (Tan & Le, 2019). Appendix K records the implementation path, optimization budgets, and configuration details.

The evaluation contains several endpoints, each addressing a different property. Fitted coefficient accuracy measures whether the AoC representation separates positive and control samples. The evidence panels inspect which training samples receive the largest archetype weights. Retrieval compares queries through their coefficient vectors. Perturbation experiments measure changes in rankings and neighbors. We report these endpoints separately because no single metric captures provenance, separability, semantic concentration, and robustness at once.

Coefficient-space separability. Positive and control samples are balanced to the smaller pool size. They form $D = \{(x_i, y_i)\}_{i=1}^{N}$. A two-class probe fitted on D provides the concept direction. AoC fits its dictionary on the positive features and screens the resulting archetypes. It then infers coefficients u_i for every positive and control sample. A new two-class linear classifier $h_\theta(u)$ is trained and evaluated on these coefficients:

$$\widehat{A}_{\text{fit}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\arg\max_r h_{\hat{\theta},r}(u_i) = y_i\}. \tag{16}$$

Here $\hat{\theta}$ is fitted by cross-entropy on the same (u_i, y_i) used for evaluation. We therefore call $\widehat{A}_{\text{fit}}$ *fitted coefficient-space separability*. It measures how clearly the labels can be separated in the coefficient representation. It is distinct from held-out prediction accuracy and from the fixed-probe preservation result in Theorem 4.3. In particular, the refitted classifier may separate coefficient mixtures even when the original probe is positive on their reconstructed features.

Appendix H reports the four-row AoC–CRAFT comparison. On the two CUB settings, AoC is higher by 6.6 and 4.7 percentage points. The DTD differences are smaller and mixed: $+0.9$ and -1.8 points. This contrast suggests that the relative behavior depends on the structure of the concept family and feature space.

The recorded elapsed ratios include feature extraction, probe fitting, and decomposition. They are accumulated across retained-size evaluations in a shared-cache worker. Results are selected by the largest fitted reduced accuracy within each dataset/model/method/concept/seed group, using seeds 7, 42, and 123. Appendix K gives the selection and timing details.

Baseline, screening, and retrieval comparisons. The supplementary comparisons show that performance varies by representation and screening rule. On CUB/RN50, AoC minus PCA is -1.93 percentage points in fitted reduced accuracy. At $k = 10, k' = 3$, alignment screening minus random screening is -3.86 points. These comparisons indicate that neither the decomposition nor the screening rule dominates in every recorded setting.

The retrieval protocol uses positive queries and a balanced positive/control candidate pool. It applies cosine similarity to each representation and treats other positives as relevant after excluding the query itself. The scalar CAV baseline is also cosine-normalized, which produces ties among scores with the same sign. We therefore interpret the retrieval result as a comparison under this particular relevance rule, rather than as a universal measure of instance similarity.

5.2 Supplementary diagnostics

The sign-preservation analysis contains 6,702 reconstructed positive samples across 36 runs. It uses the stabilized ratio $\rho = \|\hat{x} - x\|(\|w_c\| + \epsilon)/(|g_c(x)| + \epsilon)$ with $\epsilon = 10^{-12}$. A total of 265 samples (3.95%) fall below 1, and all 265 preserve the probe sign. Overall preservation is 96.4% when averaged over runs and 94.8% when weighted by sample count. The stabilized ratio differs slightly from the strict condition $\|\hat{x} - x\|\|w_c\| < |g_c(x)|$. We use it as an empirical reconstruction diagnostic, while Appendix I.1 provides the exact comparison and reconstruction procedure.

GPT matching scores are higher for top-weight archetype evidence in the displayed distributions. Appendix J compares the main judge with 11 additional VLMs. Appendix M records the evaluation prompt. Feature-space insertion and deletion curves are reported in Appendix E. Together, these analyses provide complementary views of semantic alignment and response to ranked archetypes.

Perturbation results give ranking Kendall $\tau = 0.576$ under Gaussian noise and $\tau = 0.364$ under PGD. Under PGD, retrieval overlap@5 is 0.218. The contamination analysis further shows that top-weight evidence can contain correlated non-target attributes. These diagnostics distinguish three separate questions: whether an archetype is traceable to samples, whether its representation is stable to perturbations, and whether its evidence is semantically concentrated. Full results appear in Appendices F–J.

6 Conclusion

We introduced AoC, a sample-traceable decomposition of fixed binary concept representations. The method separates three operations that are often conflated. Archetypal decomposition captures variation among positive samples. Probe-based screening connects that variation to the original concept direction. Query-wise inference provides coefficients for downstream comparison and organization.

The simplex constraints make each archetype traceable to training features. The analysis formalizes this convex-hull provenance and gives a margin-based certificate for fixed-probe prediction preservation. Conditional results further relate coefficient sensitivity to curvature and selection stability to a score gap.

The experiments illustrate both the utility and the setting dependence of the representation. Coefficient separability, evidence inspection, retrieval, and perturbation analyses evaluate different aspects of the interface and need not rank methods identically. AoC is therefore most naturally used as an inspection and representation tool in a frozen feature space. Semantic concentration, robustness, and downstream relevance remain properties to evaluate for the application at hand.

AI USE STATEMENT

Generative AI tools assisted with language polishing, structural revisions, LaTeX formatting, and the presentation of mathematical expressions. The authors remain responsible for verifying all assertions, proofs, citations, code, figures, and reported results.

REFERENCES

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2711–2721, 2023.

Thomas Fel, Ekdeep Singh Lubana, Jacob S Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.

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

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

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

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. Pmlr, 2020.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.

Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey. *ACM Computing Surveys*, 2023.

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 *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PmLR, 2019.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, 2011.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models, 2022.

Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11682–11690, 2021.

A Proof of Theorem 4.3

Proof. We have $g_c(\hat{x}) - g_c(x) = w_c^\top(\hat{x} - x)$. By the Cauchy–Schwarz inequality,

$$|g_c(\hat{x}) - g_c(x)| \le \|w_c\|_2 \, \|\hat{x} - x\|_2 < \gamma_c(x) = |g_c(x)|. \tag{17}$$

Therefore, $g_c(\hat{x})$ cannot cross 0, so $g_c(\hat{x})$ and $g_c(x)$ have the same sign. This implies that a reconstruction error smaller than the safety radius preserves the binary decision of the fixed concept probe. □

B Proof of Theorem 4.6

Proof. Compactness of K and continuity give existence for every $x \in N$; uniform strong convexity gives uniqueness. Write $u = u_K(x)$ and $v = u_K(x')$. The first-order variational inequalities on the same convex set yield

$$\langle \nabla J(u; x), v - u \rangle \ge 0, \qquad \langle \nabla J(v; x'), u - v \rangle \ge 0.$$

Strong monotonicity at x and these inequalities imply

$$\begin{aligned} \mu\|u - v\|^2 &\le \langle \nabla J(u; x) - \nabla J(v; x), u - v \rangle \\ &\le \langle \nabla J(v; x') - \nabla J(v; x), u - v \rangle \le L\|x - x'\|\|u - v\|. \end{aligned}$$

Divide by $\|u - v\|$ unless it is zero. Uniform curvature for every x is essential; curvature only at x_0 plus gradient continuity cannot imply nearby uniqueness.

□

C Proof of Theorem 4.1

Proof. Since $B_{j,:} \in \Delta_n$, each archetype is

$$z_j = \sum_{t=1}^{n} B_{j,t} x_t, \tag{18}$$

where $B_{j,t} \ge 0$ and $\sum_t B_{j,t} = 1$. Hence $z_j \in \mathrm{conv}\{x_1, \ldots, x_n\}$ for every j. Since $C_{i,:} \in \Delta_k$, $\hat{x}_i$ is a convex combination of $\{z_j\}_{j=1}^k$. Substituting the expression for z_j into $\hat{x}_i = \sum_j C_{i,j} z_j$, we obtain:

$$\hat{x}_i = \sum_{t=1}^{n} \left(\sum_{j=1}^{k} C_{i,j} B_{j,t} \right) x_t. \tag{19}$$

Let $\alpha_{i,t} := \sum_j C_{i,j} B_{j,t}$. Since elements of C and B are non-negative, $\alpha_{i,t} \ge 0$. Furthermore,

$$\sum_t \alpha_{i,t} = \sum_j C_{i,j} \sum_t B_{j,t} = 1. \tag{20}$$

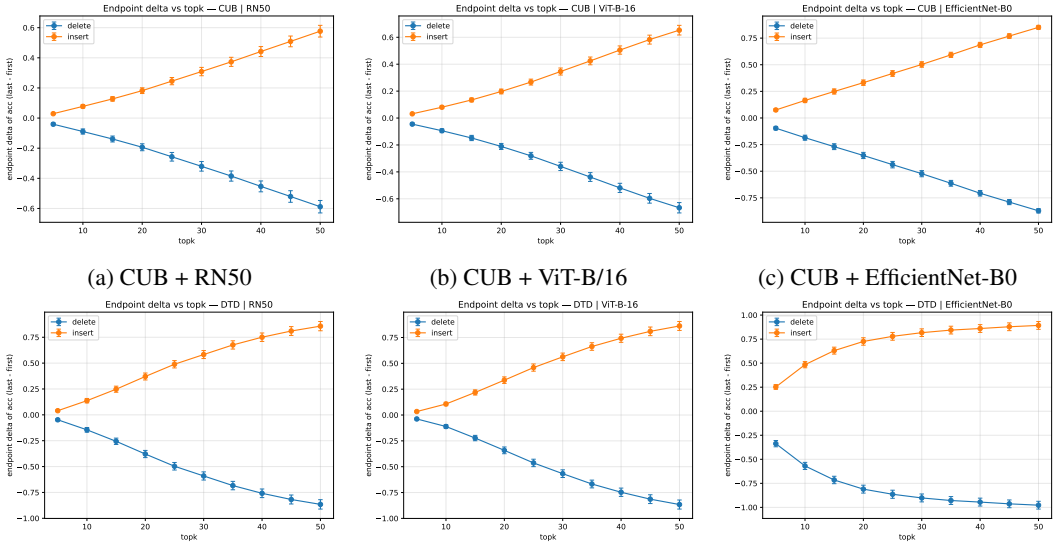
Thus, $\hat{x}_i$ is also a convex combination of $\{x_t\}$, implying it belongs to their convex hull. □

D Proof of Corollary 4.2

Proof. The Euclidean norm is a convex function. Therefore, for any convex combination $\sum_t \alpha_t x_t$ with $\alpha \in \Delta_n$:

$$\left\| \sum_t \alpha_t x_t \right\|_2 \le \sum_t \alpha_t \|x_t\|_2 \le \sum_t \alpha_t R = R \tag{21}$$

Applying this inequality first to $z_j = \sum_t B_{j,t} x_t$ and then to $\hat{x}_i = \sum_j C_{i,j} z_j$ completes the proof. □



(a) CUB + RN50 (b) CUB + ViT-B/16 (c) CUB + EfficientNet-B0

(d) DTD + RN50 (e) DTD + ViT-B/16 (f) DTD + EfficientNet-B0

Figure 2: Endpoint delta curves for top-k sequential analysis. Top row: CUB; bottom row: DTD; each column corresponds to one visual encoder. The curves report cumulative feature-space response as ranked archetypes are inserted or deleted under the recorded protocol.

E Top-k Sequential Verification: Endpoint Delta and Response Sensitivity

Fig. 2 reports, across datasets and backbones, the cumulative *endpoint delta* in accuracy from the start to the end of progressively *inserting/deleting* the top-k archetypes according to our importance ranking. The top row contains the CUB panels for RN50, ViT-B/16, and EfficientNet-B0 (Figs. 2a, 2b, and 2c); the bottom row contains the corresponding DTD panels (Figs. 2d, 2e, and 2f). The plots summarize feature-space endpoint changes as progressively more ranked archetypes are inserted or deleted. Their interpretation depends on the reconstruction rule, coefficient handling, zero-component baseline, and uncertainty unit. We therefore use them as descriptive response curves and compare their shapes across datasets and backbones.

F Qualitative AoC vs. CRAFT Panel on CUB

G Archetype Evidence and Coefficient-Space Retrieval

Figure 4 contrasts top-weight archetype evidence with coefficient-space retrieval under the same candidate vocabulary. The left stack is referenced as Fig. 4a and the right stack as Fig. 4b.

Top-weight archetype evidence. As shown in Fig. 4, these examples are the training images with the largest entries in an archetype's weight vector. They expose the samples that contribute most strongly to the feature-space archetype. The accompanying vocabulary-naming scores summarize their alignment with the candidate words.

Coefficient-space retrieval. The comparison examples are nearest neighbors in coefficient space, organizing queries according to their inferred archetype mixtures. The reported protocol uses its specified candidate pool, similarity function, and relevance rule.

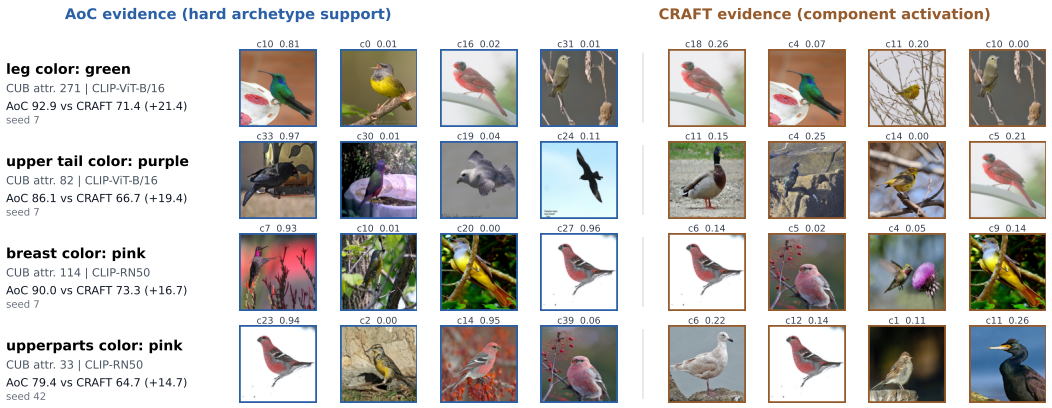
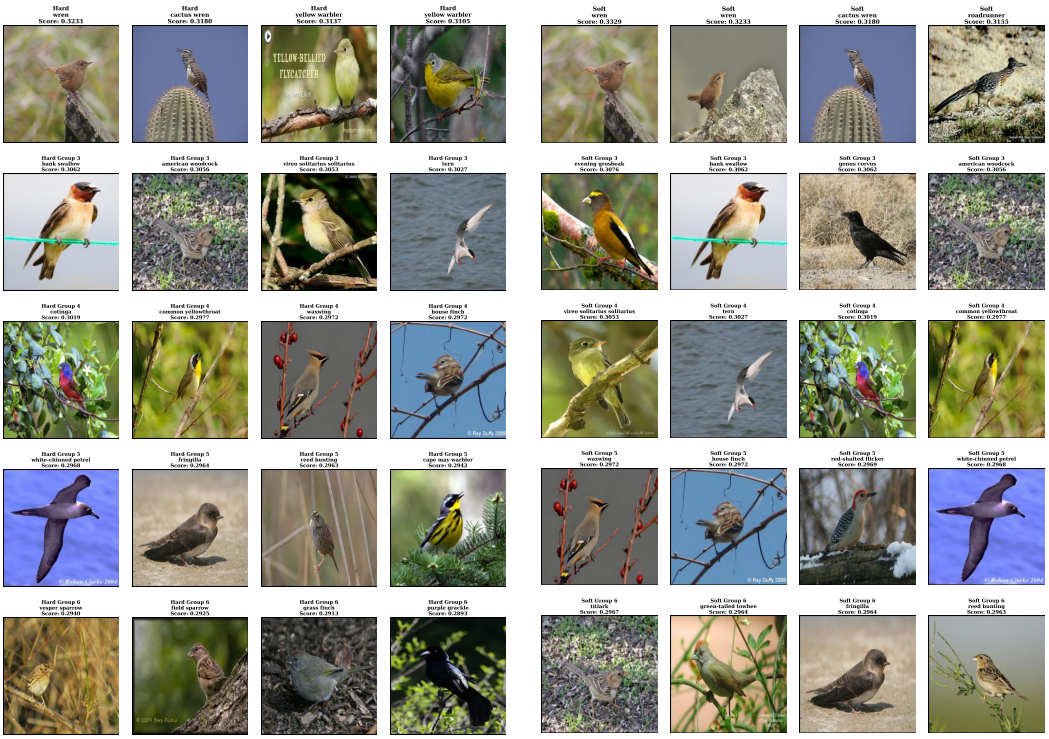


Figure 3: Illustrative AoC and CRAFT evidence panels for matched CUB concept–seed pairs with large AoC differences. Blue panels show samples receiving the largest AoC archetype weights; brown panels show samples with high CRAFT component activations. For these fine-grained attributes, AoC evidence emphasizes the target visual cue, whereas the displayed CRAFT examples include more whole-bird appearance and nearby co-occurring cues.



(a) Top-weight archetype evidence: training images with the largest weights in the displayed archetypes.

(b) Coefficient-space retrieval: nearest neighbors under the recorded coefficient representation.

Figure 4: Comparison of top-weight archetype evidence and coefficient-space retrieval. Each side aggregates five groups (best match and Groups 3–6). Candidate words and GPT matching scores are outputs of the semantic-evaluation protocol.

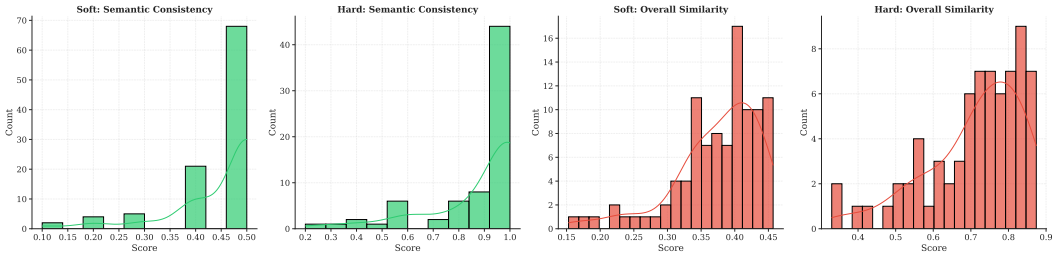


Figure 5: GPT matching distributions for top-weight evidence and coefficient-space retrieval on CUB/CLIP-RN50 under the recorded candidate vocabulary and evaluation pool.

Table 2: AoC vs. CRAFT results. Accuracy is fitted coefficient-probe accuracy (mean ± std); Win is the matched-pair fraction; Elapsed is the recorded mean cumulative time ratio under the shared-cache execution protocol.

Dataset	Backbone	AoC	CRAFT	Δ	Win / Elapsed
CUB	CLIP-RN50	72.5±8.7	65.9±8.5	+6.6	96.7% / 1.50×
CUB	CLIP-ViT-B/16	73.3±8.7	68.6±8.4	+4.7	92.0% / 1.78×
DTD	CLIP-RN50	81.2±5.3	80.3±5.9	+0.9	56.7% / 1.10×
DTD	CLIP-ViT-B/16	82.0±4.8	83.8±6.0	-1.8	26.7% / 1.16×

H Comparison Protocol and Results

Table 2 reports the paired AoC–CRAFT summary. Accuracy is measured using the refitted coefficient classifier described in Section 5. Appendix K records the selection rule and effective optimization budgets; the three seeds for each concept are grouped observations of that concept.

I Additional Stability and Robustness Analyses

This section reports supplementary diagnostics for label preservation, perturbation robustness, few-shot direction estimation, and cross-concept attribute concentration.

I.1 Label-Preservation Stress Test

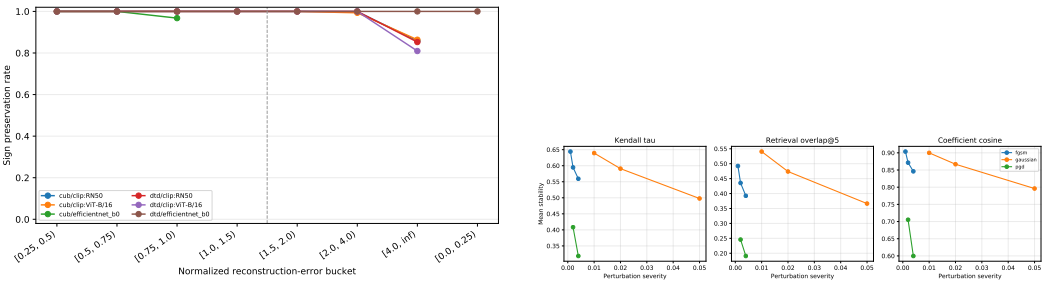
Theorem 4.3 relates label preservation to reconstruction error and concept-probe margin. For each reconstructed concept-positive feature, the diagnostic computes

$$\rho(x) = \frac{\|\hat{x} - x\|_2(\|w_c\|_2 + \epsilon)}{|w_c^\top x + b_c| + \epsilon}, \qquad \epsilon = 10^{-12},$$

and groups samples by $\rho(x)$. Across 36 AoC runs, all below-threshold cases ($\rho(x) < 1$; $265/265$ samples) preserve the probe sign. The overall preservation rate is 96.4% averaged over runs and 94.8% sample-weighted across 6,702 reconstructed positives. Figure 6a shows perfect preservation in the low-error buckets and a decrease in the largest-error bucket. The below-threshold subset covers $265/6702 = 3.95\%$ of samples. Because ρ includes ϵ, the exact theorem certificate remains the strict inequality $\|\hat{x} - x\|\|w_c\| < |g_c(x)|$; the displayed statistic is its stabilized empirical counterpart. Reconstructions use the retained full-dictionary coefficients $\hat{x}_i = \sum_{j \in S} U_{ij} z_j$ without subset re-inference.

I.2 Noise and Adversarial Robustness

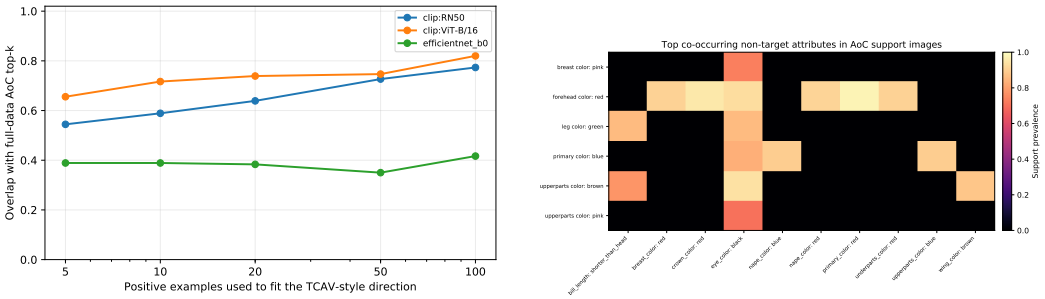
We next summarize the noise, FGSM, and PGD robustness outputs. Under Gaussian perturbations, mean reduced accuracy decreases by 2.1 percentage points and coefficient-ranking Kendall τ is 0.576. FGSM gives Kendall $\tau = 0.599$ with an 11.8-point accuracy decrease. PGD gives Kendall $\tau = 0.364$ and retrieval overlap@5 of 0.218. Figure 6b visualizes the increasing degradation from Gaussian



(a) Label-preservation reliability curve.

(b) Coefficient/ranking stability under perturbations.

Figure 6: Additional stability diagnostics. Left: concept-probe sign preservation as a function of the stabilized normalized reconstruction error. Right: coefficient-space stability under Gaussian noise, FGSM, and PGD.



(a) Few-shot TCAV-direction proxy.

(b) Correlated-attribute analysis.

Figure 7: Additional representation diagnostics. Left: overlap between few-shot TCAV-direction rankings and full-data AoC top-10 archetype rankings as the number of positives increases. Right: correlated non-target attributes in the highest-weight evidence for target CUB attributes.

noise to adversarial perturbations. The recorded summary does not include all attack hyperparameters, so these values are reported as empirical robustness diagnostics.

I.3 Few-Shot TCAV-Direction Proxy

To study limited concept supervision, we estimate a TCAV-style direction from a small positive set and matched negatives, score the fixed AoC archetypes, and compare the resulting top-10 set with the full-data ranking. The archetype basis is held fixed, isolating the effect of estimating the concept direction. Mean overlap@10 rises from 0.53 with 5 positives to 0.67 with 100 positives, while score Spearman correlation rises from 0.63 to 0.96. Figure 7a shows the corresponding recovery trend.

I.4 Cross-Concept Contamination and Semantic Drift

Finally, we measure correlated non-target attributes in the highest-weight AoC evidence for each target CUB attribute. The target attribute is present in every inspected evidence set, while the average maximum non-target prevalence is 0.832 and several non-target attributes exceed prevalence 0.5. Figure 7b shows this co-occurrence structure at the attribute-pair level, highlighting residual entanglement with adjacent color parts and common bird morphology.

J Multi-VLM Semantic-Judge Agreement

Because the semantic-alignment analysis uses GPT-4.1 as a judge, we compare its outputs with 11 additional VLM judges on the same 171 CUB/CLIP-RN50 image groups, with five images per group. The judge pool spans Anthropic, Google, Meta-Llama, Mistral, and Qwen model families (Table 3).

Table 3: VLM judge pool used in the multi-judge agreement analysis.

Provider / family	Model identifier
Anthropic	`anthropic/claude-3-haiku`
Anthropic	`anthropic/claude-haiku-4.5`
Google	`google/gemini-2.0-flash-001`
Google	`google/gemini-2.0-flash-lite-001`
Google	`google/gemini-2.5-flash-lite`
Meta-Llama	`meta-llama/llama-4-scout`
Mistral	`mistralai/mistral-small-3.2-24b-instruct`
Qwen	`qwen/qwen2.5-vl-72b-instruct`
Qwen	`qwen/qwen3-vl-235b-a22b-instruct`
Qwen	`qwen/qwen3-vl-32b-instruct`
Qwen	`qwen/qwen3-vl-8b-instruct`

Table 4: Agreement between GPT-4.1 semantic matching and 11 independent VLM judges on 171 CUB/CLIP-RN50 concept groups. Scores use the same 0–100 image-level scale as the main GPT prompt.

Summary over judges	Label Agr.	κ	Spearman	Pearson	MAE
Mean	59.6%	0.369	0.673	0.690	15.35
Median	60.8%	0.386	0.720	0.732	14.94
Minimum	48.0%	0.183	0.328	0.315	12.43
Maximum	64.4%	0.440	0.786	0.805	19.10

Mean Spearman and Pearson correlations are 0.673 and 0.690, respectively, and 9 of the 11 judges have Spearman correlation of at least 0.60 with GPT-4.1. Mean label agreement is 59.6% and mean Cohen's κ is 0.369, indicating stronger agreement in score ordering than in discrete label assignment. These results measure cross-model consistency under the shared vocabulary and prompt; human calibration remains a separate evaluation.

K Implementation and Configuration Details

Table 5 records the phase-2 implementation used to interpret Table 2. It distinguishes class defaults from launcher overrides and specifies the observed optimization and evaluation paths.

Table 5: Solver configuration recorded in `main.py` and `AA.py`, including class defaults and the phase-2 call path.

Item	Observed source behavior
Fitting / inference	Adam on softmax logits; no explicit simplex projection
Initialization	Gaussian logits for B, C; normalized NNLS initialization for subset inference, with fallback
Class defaults	3000 fit/inference iterations and learning rate 10^{-2}; phase-2 launcher overrides the iteration budget
Cosine objective	PyTorch cosine with each norm clamped at 10^{-8}
Regularization	No $\tau\|u\|^2/2$ term in the inspected default subset path
Probe endpoint	Refit linear classifier on coefficients, measured on its fitting data
Data split	Phase-2 launcher passes `train`

The phase-2 launcher uses 500 fit/transform steps and 300 probe steps. AoC evaluates $k \in \{10, 20, 40\}$ and valid pairs with $k' \in \{5, 10, 20, 40\}$; the CRAFT adaptation evaluates $k \in \{10, 20\}$ and $k' \in \{5, 10, 20\}$. The paired summaries contain 300 concept–seed observations per CUB backbone (100 concepts and three seeds) and 90 per DTD backbone (30 concepts and three seeds). Rows are selected by fitted accuracy. The CRAFT comparison uses a patch-fit NMF adaptation, and the timer covers feature extraction, probe fitting, and decomposition accumulated across the retained-size loop. These implementation choices define the scope of the comparison in Table 2.

Table 6: Auxiliary compute summary. Log timings are reported separately from the cumulative comparison times described in Appendix K.

Item	Recorded value	Caveat
Configuration coverage	33/33 rows reported	Metadata summary
Runtime coverage	4/7 compute blocks	Peak memory not logged
Recorded host	2× L40S; 503.6 GiB RAM	Host snapshot
OpenRouter VLM analysis	1,880 calls; 2.60M tokens; $1.00	Provider-managed inference
Robustness logs	17.97 s mean/run; 4.49 h sum	Parsed from auxiliary logs
Auxiliary fit logs	11.80 s mean/run; 1.40 h sum	Parsed from CRAFT/NMF logs

L Compute and Reproducibility Details

Table 6 summarizes the available configuration and compute records. All 33 extracted configuration rows report datasets, backbones, splits, seeds, k/k' sweeps, and method variants, while runtime is available for 4 of 7 compute blocks. Peak memory and frozen-feature extraction time were not recorded for every run. The available host snapshot contains two NVIDIA L40S GPUs with 46,068 MiB each, 24 CPU cores, and 503.6 GiB system RAM.

M Prompt for Visual Concept Evaluation

The semantic evaluation calls GPT-4.1 once per instance with temperature 0 and a maximum output length of 4096 tokens, without repeated sampling or self-consistency voting. The prompt is reproduced below for transparency. It requests ten candidate words while specifying five scores per image; this wording is retained to match the recorded evaluation protocol.

```
SYSTEM_PROMPT = """
You are a strict and conservative visual concept evaluator for academic research.

Your task is to evaluate how well a set of images matches a list of "top words"
     representing a concept.

You will be provided with:
1. A concept ID.
2. A list of 10 "top words" (ranked by relevance).
3. A set of 5 images (img01 to img05).

Your output MUST be a valid JSON object following the given schema.

========================
EVALUATION PROCEDURE
========================

You MUST follow this procedure for EACH image:

Step 1: Coarse-domain check
- Determine whether the image belongs to the coarse semantic domain implied by the top
     words.
- Coarse-domain for this concept is defined by the dominant high-ranked words.
  (e.g., if top words include "hummingbird", "hummer", or "feeder", then the domain is
   "hummingbird and/or hummingbird feeder scene".)
- If an image does not contain the core domain entities, label it as "outlier".
- Outliers must have best_score <= 20 and all word scores <= 20.

Step 2: Fine-grained matching
- Only if the image belongs to the domain, evaluate similarity to EACH top word.
- Assign a score from 0 to 100 for every word.

========================
LABEL ASSIGNMENT RULES
========================

Use labels conservatively:

- "strong":
  Assign ONLY if the image clearly and unambiguously depicts ONE specific top word,
  and this word is visually dominant over all other top words.
```

```
  There must be little to no plausible confusion.
  If both "feeder" and "hummingbird/hummer" are clearly visible and dominant, you SHOULD
     assign "strong",
  unless the view is extremely ambiguous.

- "medium":
  Assign if the image belongs to the correct domain but:
    * multiple top words are plausible, OR
    * the visual evidence is incomplete or ambiguous, OR
    * the concept is present but not dominant.

- "outlier":
  Assign if the image does not belong to the domain at all.

If there is any uncertainty, you MUST choose "medium" instead of "strong".

========================
SCORING GUIDELINES
========================

Scores must be realistic and well-calibrated:

- 90--100: almost certain visual match
- 70--89: highly likely
- 40--69: weak or partial match
- 0--39: irrelevant or very weak

Use the full range of 0--100 when appropriate.
For "medium" cases, best_score should typically be between 60 and 85.

========================
SUMMARY METRICS
========================

K = 5

purity_strong = (# of images with label == "strong") / K
semantic_consistency = (# of images with label == "strong" OR "medium") / K
overall_similarity = (1/K) * sum(best_score_of_each_image / 100)

Round all summary metrics to 4 decimal places.

========================
OUTPUT CONSTRAINTS
========================

- Output ONLY the raw JSON object.
- No explanations.
- No markdown.
- Follow the schema exactly.
- Ensure:
  * best_word is one of the top words
  * best_score matches the corresponding score
  * each image has exactly 5 scores
  * summary metrics are numerically correct.

This evaluation is used for quantitative analysis in concept-based model research.
Accuracy, calibration, and conservativeness are more important than confidence.
"""

USER_PROMPT_TEMPLATE = """
Concept ID: {concept_id}

Top Words (Ranked):
{top_words_str}

Images:
(Attached 5 images with IDs: img01 to img05)

Please evaluate each image against the top words and provide the summary metrics.
"""
```