CONCEPTUAL ARCHETYPE DECOMPOSITION FOR TRACING RECONSTRUCTED NEURAL ACTIVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept decompositions of neural activations usually expose a dictionary and reconstruction weights, but inspecting how a reconstructed test feature draws on particular training locations requires an explicit provenance map. We present Conceptual Archetype Decomposition (CAD), a post-hoc explanatory interface derived from classical archetypal analysis. Concepts are convex combinations of full-image training activations, and test features are reconstructed as convex combinations of these fixed concepts. Composing the two coefficient matrices gives a normalized map from each test location through concepts to training locations. This map is exact for the learned decomposition, remains available before optimizer convergence, and its reconstructed features satisfy a perturbation bound when the coefficients are fixed. Existing controlled CUB results describe reconstruction sensitivity to the implementation choices; we distinguish these results from a secondary pairwise logit diagnostic that does not establish prediction fidelity. CAD supplies inspectable training evidence while leaving the original predictor unchanged. The core implementation code is available in the supplementary material.

1 INTRODUCTION

Concept-based explanations describe neural representations using a dictionary of directions and coefficients Kim et al. (2018); Fel et al. (2023). For a fixed vision model, two practical questions follow: which training image regions constitute a concept, and how do those regions enter a reconstruction of a new image? A learned dictionary alone does not answer both questions. For example, CRAFT Fel et al. (2023) uses nonnegative matrix factorization (NMF) Lee & Seung (1999); its dictionary can be inspected through activating examples, but its factorization does not itself retain normalized coefficients linking dictionary atoms to training locations.

We present *Conceptual Archetype Decomposition* (CAD), an archetypal-analysis-derived interface for these questions. Classical archetypal analysis (AA) already supplies the nested convex-combination objective Cutler & Breiman (1994). CAD applies it to full-image sample-location activations of a frozen predictor, retains the training-location identities, and freezes the learned dictionary for test-time reconstruction. A concept index matrix C records how training locations form concepts; a separate composition matrix B_{test} records how concepts reconstruct test locations. Their product $P = B_{\text{test}}^{\top} C$ provides the complete normalized provenance map.

CAD makes the two stages of this decomposition jointly inspectable: each reconstructed test location can be traced through its concept weights to the training activations that constitute those concepts. The simplex constraints give both stages a common normalized scale. Our contributions are:

- A frozen-predictor protocol that connects test locations to concepts and retained training locations using full-image activations and a fixed train-only dictionary.
- A formal account of the composed provenance map, its fixed-coefficient perturbation bound, and the margin conditions needed to turn reconstruction fidelity into prediction agreement.
- A controlled study of reconstruction sensitivity to the composition constraint and activation support, together with an explicit characterization of the class-conditioned pairwise logit score.

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.

2 Related Work

2.1 Traditional interpretability methods

Interpretability research includes post-hoc methods (e.g., Grad-CAM Selvaraju et al. (2017), IG Sundararajan et al. (2017), Shapley Lundberg & Lee (2017), LIME Ribeiro et al. (2016)) and inherently interpretable models (e.g., Concept Bottleneck Models Koh et al. (2020)). We focus on post-hoc concept decomposition for already-trained vision models. Attribution methods such as Grad-CAM and IG highlight regions associated with a prediction; concept decompositions additionally organize intermediate representations into a dictionary that can be inspected across images.

2.2 Research and Application of Concept Decomposition Explainability

Concept-based decomposition methods explain decisions by extracting human-interpretable concepts from intermediate representations. RSVC Kondapaneni et al. (2025b), RDX Kondapaneni et al. (2025a), and ModelDiff Shah et al. (2023) use concept decomposition to study differences between models or training regimes. TCAV Kim et al. (2018) learns concept vectors from user-provided positive/negative sets, but depends on manual concept curation. ACE Ghorbani et al. (2019) automates concept discovery via unsupervised techniques. CRAFT Fel et al. (2023) factorizes intermediate features via NMF Lee & Seung (1999) and requires nonnegative factorization inputs. Its crop-based support construction can differ from the full-image activations used at inference.

CAD retains decomposition coefficients over training locations rather than relying only on post-hoc retrieval of activating examples. This gives explicit provenance; semantic interpretation still requires inspecting the retrieved evidence.

Relation to archetypal analysis and sparse autoencoders. The objective and convex-combination identities in CAD descend from classical AA Cutler & Breiman (1994). Convex NMF Ding et al. (2010) likewise constrains dictionary atoms using data. Archetypal sparse autoencoders (A-SAE) Fel et al. (2025) are especially close: their strict dictionary is anchored in the data convex hull, while sparse nonnegative codes generate conical combinations of dictionary atoms. CAD constrains both stages to simplices, placing reconstructed activations in the dictionary hull, and solves for test coefficients with the dictionary fixed. Relaxed A-SAE additionally permits dictionary deviations from the data hull. Prototype classifiers Chen et al. (2019); Nauta et al. (2021) incorporate prototypes into the predictor itself. CAD uses the AA construction around a frozen multiclass predictor, retaining activation/token identities to expose the composed path from a test reconstruction to its training sources.

3 Method

CAD retains the two stages of a learned decomposition as an explicit interface: training location to concept, and concept to reconstructed test location.

Shape conventions. Let L be the number of retained activation identities per image: $L = h_f w_f$ for a CNN feature grid, or the retained token count for a transformer. An identity is an image plus a spatial location or token index; CLS has image/token provenance but no patch region. The activation matrix is $A \in \mathbb{R}^{nL \times c}$, with each row $A_{j,:} \in \mathbb{R}^{1 \times c}$ being a retained activation vector. We use "location" as shorthand for this activation identity, without assigning a spatial region to CLS. The concept index matrix $C \in \mathbb{R}^{k \times nL}$ is row-wise simplex constrained: $\forall i, \sum_{j=1}^{nL} C_{ij} = 1, C_{ij} \geq 0$. The per-sample composition matrix $B \in \mathbb{R}^{k \times nL}$ is column-wise simplex constrained: $\forall j, \sum_{i=1}^{k} B_{ij} = 1$, $B_{ij} \geq 0$. The concept dictionary is $Z = CA \in \mathbb{R}^{k \times c}$, whose row $Z_{i,:} = \sum_{j=1}^{nL} C_{ij} A_{j,:}$ is a convex combination of training-sample activations. The CAD reconstruction is $\hat{A} = B^\top CA = B^\top Z \in \mathbb{R}^{nL \times c}$, so that $\hat{A}_{\ell,:} = \sum_{i=1}^{k} B_{i\ell} Z_{i,:}$ is, for each sample-location ℓ, a convex combination of concept rows.

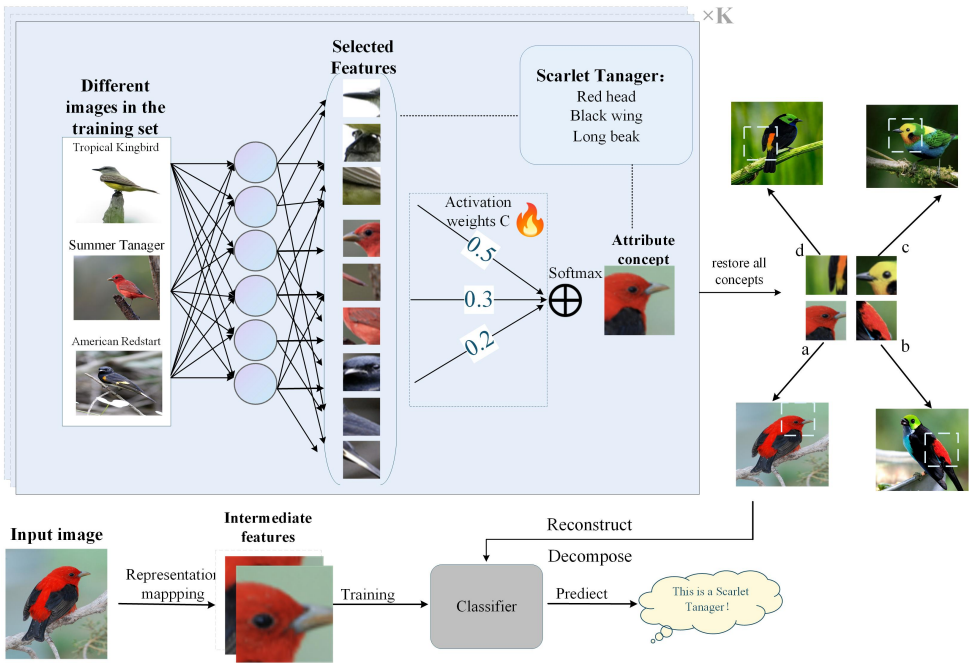

Figure 1: CAD overview. The encoder and classifier are pretrained and frozen: the lower arrow labelled "Training" refers to predictor pretraining, not CAD fitting. CAD fits C, B on training activations, then fixes $Z = CA$ and fits only B_{test} for new inputs. The reconstructed-head output is a diagnostic. Bird crops schematically illustrate concept supports, not reconstructed pixels or verified semantic labels.

3.1 Problem Definition

Given an image $x^{(i)} \in \mathbb{R}^{H_{\text{in}} \times W_{\text{in}} \times 3}$ and its class label $y^{(i)}$, we aim to decompose deep features into concepts with inspectable training supports. Cropping subregions Chen et al. (2019) changes the inputs used for feature extraction; using full images retains their original spatial context. CAD instead operates on full-image activations: a mapping g produces an intermediate activation $A(x^{(i)}) \in \mathbb{R}^{L \times c}$ Goodfellow et al. (2016), and a classifier h yields $f(x^{(i)}) = h(g(x^{(i)}))$, with predicted class $\hat{y}^{(i)} = \operatorname{argmax}_c f_c(x^{(i)})$, which need not equal $y^{(i)}$.

3.2 Preliminaries

CRAFT Fel et al. (2023) factorizes the intermediate-layer activation matrix into a concept dictionary and a non-negative coefficient matrix via NMF Lee & Seung (1999), solving:

$$(U^*, W^*) \in \underset{U,W}{\operatorname{argmin}} \|A - UW\|_F^2, \quad \text{s.t. } U, W \geq 0 \tag{1}$$

where $A \in \mathbb{R}^{nL \times c}$ is the reshaped activation matrix, $W \in \mathbb{R}^{k \times c}$ is the concept dictionary, $U \in \mathbb{R}^{nL \times k}$ is the concept coefficient matrix, and k is the number of concepts. The reconstruction is $A \approx UW$. The NMF formulation does not require or retain coefficients C such that $W = CA$ with row-simplex C; that dictionary provenance is therefore not supplied by its learned factors alone.

3.3 Conceptual Archetype Decomposition

Concept index matrix C. CAD associates every dictionary row with normalized coefficients over training activations. Retaining these coefficients and the corresponding activation identities makes the construction of each learned concept inspectable.

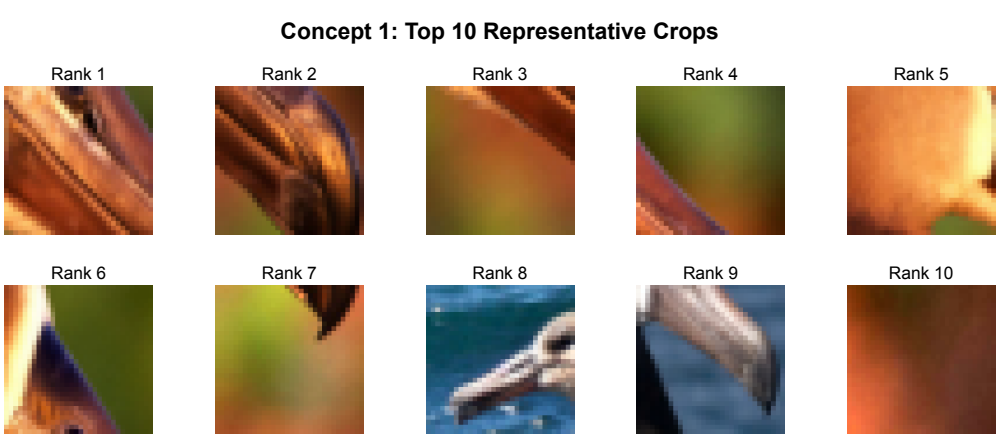

Figure 2: Visualization of CAD concept support on CUB (NF-ResNet50, k=30, seed 0): Concept 1. Each panel shows a training-set image crop centered on the spatial location j whose CAD index value C_{1j} is among the top entries of the row $C_{1,:}$. Crops are arranged left-to-right, top-to-bottom in descending order of C_{1j}, i.e. in descending order of decomposition weight under the learned CAD decomposition. Labels indicate coefficient rank, not coefficient values. These seed-0 examples are separate from the seed-42 main artifacts and seeds 7–12 ablation. A centered crop visualizes a feature location, not its complete receptive field or contextual evidence. Crops are not class-conditional and are retrieved purely from the index matrix.

As illustrated in Figure 1, CAD represents the concept reconstruction process as $\hat{A} = B^\top CA$, with $A \in \mathbb{R}^{nL \times c}$ the activation matrix, $C \in \mathbb{R}^{k \times nL}$ the concept index matrix, and $B \in \mathbb{R}^{k \times nL}$ the per-sample composition matrix (see the shape conventions above). The i-th concept vector is $z_i^\top = (CA)_{i,:} = \sum_{j=1}^{nL} C_{ij} A_{j,:}$, i.e. a non-negative combination of training-sample activations. We enforce $\forall i, \sum_{j=1}^{nL} C_{ij} = 1$ via row-wise softmax,

$$C_{ij} = e^{c_{ij}} \Big/ \sum_{l=1}^{nL} e^{c_{il}}, \tag{2}$$

so each concept is a convex combination of training activations, and the support of $C_{i,:}$ recoverably indexes the contributing (image, activation/token identity) pairs by construction. Unlike CRAFT, which relies on post-hoc activation thresholding to retrieve concept supports, CAD's concept supports are an intrinsic, recoverable index. Figure 2 visualizes the top index sources of Concept 1 (more in Appendix).

Per-sample composition matrix B. In addition to the concept index matrix C, we also introduce a per-sample composition matrix B, which controls how each sample-location is reconstructed from the learned concepts. Following the shape convention above, B is column-wise simplex constrained, i.e. $\forall j,\ \sum_{i=1}^{k} B_{ij} = 1, B_{ij} \geq 0$. Concretely, the column $B_{:,\ell}$ specifies the convex composition of concepts that reconstructs the activation at sample-location ℓ. We enforce the column-wise simplex via a column-wise softmax, mirroring the operation used for C (see Algorithm 1).

Compared with CRAFT's nonnegative, unnormalized coefficient matrix U, B has two advantages: (i) *unified scale*—each column of B lies on the probability simplex, so concept weights are directly comparable across positions; and (ii) B represents a *normalized composition* rather than an unbounded activation strength, which guarantees that every reconstructed sample-location is a convex combination of concepts in Z. The resulting AA-derived optimization is:

$$
\begin{aligned}
(B^*, C^*) \in \operatorname*{argmin}_{B,C} \left\| A - B^\top C A \right\|_F^2 \\
\text{s.t.} \quad \forall i,\ \sum_{j=1}^{nL} C_{ij} = 1, \quad C_{ij} \geq 0, \\
\forall j,\ \sum_{i=1}^{k} B_{ij} = 1, \quad B_{ij} \geq 0.
\end{aligned}
\tag{3}
$$

Hatted matrices below denote approximate feasible outputs of the practical solver; stars denote ideal minimizers. In the inference phase, we fix the concept dictionary $Z = \hat{C} A_{\text{train}}$ obtained during training and reuse it to reconstruct any new test activation matrix $A_{\text{test}} \in \mathbb{R}^{n_{\text{test}} L \times c}$. Concretely, we solve a separate test-time optimization for a test-time composition matrix $B_{\text{test}} \in \mathbb{R}^{k \times n_{\text{test}} L}$ defined on its own column-wise simplex set

$$
\mathcal{B}_{\text{test}} = \left\{ B \in \mathbb{R}^{k \times n_{\text{test}} L} \;\middle|\; B_{ij} \geq 0,\ \sum_{i=1}^{k} B_{ij} = 1,\ \forall j \right\},
$$

i.e.,

$$
B^*_{\text{test}} \ \in \ \operatorname*{argmin}_{B_{\text{test}} \in \mathcal{B}_{\text{test}}} \left\| A_{\text{test}} - B_{\text{test}}^\top Z \right\|_F^2. \tag{4}
$$

The solver returns an approximate feasible $\hat{B}_{\text{test}}$. Each column provides the convex composition of training-grounded concepts that reconstructs the corresponding test sample-location, so test reconstructions inherit the same simplex/traceability structure as the training-time decomposition.

Composed provenance. Let $m = nL$ and $r = n_{\text{test}} L$. For any feasible C and B_{test}, define

$$
P = B_{\text{test}}^\top C \in \mathbb{R}^{r \times m}, \qquad P \geq 0, \quad P\mathbf{1} = \mathbf{1}, \quad \hat{A}_{\text{test}} = P A_{\text{train}}. \tag{5}
$$

Thus $P_{\ell j} = \sum_i B_{i\ell} C_{ij}$ records the coefficient of training location j in test reconstruction ℓ, summed over concept paths. These identities hold at every feasible iterate, independently of convergence. They provide an exact account of the selected reconstruction in terms of its retained training sources. Section 3.4 describes how to read these paths and bound their response to fixed-coefficient perturbations.

Optimization and cost. Each block subproblem is a convex quadratic program on a product of simplices; joint optimization is nonconvex. Algorithm 1 uses Adam on softmax logits. Matrix products and gradients cost $\mathcal{O}(kmc)$ per iteration, with activation and coefficient storage $\mathcal{O}(mc{+}km)$, retaining provenance in factorized form. Materializing the entire P additionally costs $\mathcal{O}(rmk)$ arithmetic and $\mathcal{O}(rm)$ storage; a single query row can instead be computed on demand. The test stage replaces m by r for its composition matrix and uses the fixed $k \times c$ dictionary.

Algorithm 1 Conceptual Archetype Decomposition (CAD)

Require: Train activations A_{train}; test activations A_{test}; concept count k.

1: **Training:** initialize free logits $\tilde{C}, \tilde{B}$; iteratively form $C{=}\text{rowwise_softmax}(\tilde{C})$, $B{=}\text{colwise_softmax}(\tilde{B})$ and minimize $\|A_{\text{train}} - B^\top C A_{\text{train}}\|_F^2$ in $(\tilde{C}, \tilde{B})$ with the optimizer settings of Section 4.1.

2: **Freeze concepts:** $\hat{C} \leftarrow \text{rowwise_softmax}(\tilde{C})$; $Z \leftarrow \hat{C}\, A_{\text{train}}$.

3: **Test-time:** initialize $\tilde{B}_{\text{test}}$; minimize $\|A_{\text{test}} - B_{\text{test}}^\top Z\|_F^2$ in $\tilde{B}_{\text{test}}$ (with $B_{\text{test}}{=}\text{colwise_softmax}(\tilde{B}_{\text{test}})$, Z frozen) under the test-stage optimizer settings of Section 4.1.

Ensure: $\hat{C}, Z, \hat{B}_{\text{test}}$ and $\hat{A}_{\text{test}} = \hat{B}_{\text{test}}^\top Z$.

Appendix D separates idealized exact block minimization from the practical solver and proves the structural properties used by the interface.

Ideal feasibility and the implemented parameterization. The closed-simplex problem in Eq. (3) has a minimizer because its feasible set is compact and the objective is continuous. Fixing either block gives a convex quadratic problem, and exact block minimization would yield

$$F(B^{t+1}, C^{t+1}) \le F(B^{t+1}, C^t) \le F(B^t, C^t), \quad F(B, C) = \|A - B^\top CA\|_F^2.$$

The nonnegative objective values then converge. Appendix D gives a stationary-accumulation-point result under an additional vanishing-block-change condition. These results concern exact block minimization; the practical Adam updates optimize softmax logits approximately and are not covered by that descent guarantee.

At finite logits, softmax represents strictly positive weights and approaches boundary solutions as logit differences grow. For example, a test activation at an exposed dictionary vertex admits a zero-error solution concentrated on that vertex, whereas a finite-logit mixture gives positive mass to every dictionary row. Accordingly, top-coefficient retrieval summarizes a dense mixture. The retained-mass bound below quantifies the feature error introduced by truncating that display.

3.4 Reading and checking a provenance path

For a held-out location ℓ, the column $B_{:,\ell}$ selects a mixture of concepts, while row $C_{i,:}$ records the training sources of concept i. The individual path weight is $B_{i\ell}C_{ij}$; its sum over concepts is the training-location weight $P_{\ell j}$. A display of the largest entries of $C_{i,:}$ characterizes one dictionary atom, as illustrated in Figure 2. A display of the largest entries of $P_{\ell,:}$ accounts for a particular test reconstruction through the composed weights.

Normalization and fixed-coefficient perturbations. The simplex constraints imply $P \ge 0$ and $P\mathbf{1} = \mathbf{1}$. For fixed coefficients and training-row perturbations e_j, convex averaging gives

$$\|\widetilde{\hat{A}}_{\ell,:} - \hat{A}_{\ell,:}\| \le \sum_j P_{\ell j}\|e_j\| \le \max_j \|e_j\|. \tag{6}$$

Appendix E proves this for any norm and any feasible iterate, with B, C, and the encoder held fixed. The factors and activation identities specify the reconstruction to which the bound applies.

Coefficient rank versus contribution. A larger $P_{\ell j}$ means a larger scalar mixture weight, not necessarily a larger vector contribution $\|P_{\ell j}A_j\|$ or a larger head-level effect. For a feasible transfer of mass δ from source j' to j, the reconstructed activation changes by $\delta(A_j - A_{j'})$. This follows directly by subtracting the two mixtures, with $0 \le \delta \le P_{\ell j'}$. When the two activations coincide, their weights can change without changing the reconstruction. Such ambiguity is compatible with exact provenance of the selected coefficients; exactness does not imply uniqueness or semantic identifiability.

Retained mass in a truncated display. A short list of top-weight training locations does not preserve the complete dense mixture. For a displayed set S, define its retained mass $\rho_\ell = \sum_{j\in S} P_{\ell j}$. Reporting ρ_ℓ distinguishes a concentrated explanation from a visually similar list that accounts for little of the reconstruction. For $\rho_\ell > 0$, a renormalized displayed feature is $a_S = \rho_\ell^{-1} \sum_{j\in S} P_{\ell j}A_j$. If $\rho_\ell < 1$, define the omitted mixture a_{S^c} analogously. Then

$$\hat{A}_{\ell,:} = \rho_\ell a_S + (1-\rho_\ell)a_{S^c}, \qquad \|\hat{A}_{\ell,:} - a_S\| \le (1-\rho_\ell)\Delta_A, \tag{7}$$

where $\Delta_A = \max_{j,j'} \|A_j - A_{j'}\|$ is the training-activation diameter. To see the bound, expand $a_S - a_{S^c}$ as a convex combination of differences $A_j - A_{j'}$ and apply the triangle inequality. If $\rho_\ell = 1$, the displayed mixture is exact. This gives a deterministic feature-error bound for a chosen display. The crop illustrations report coefficient ranks; applying the bound to them additionally requires their retained masses.

Inspecting a test location. For a selected test location ℓ, compute $P_{\ell,:} = B_{\mathrm{test},:,\ell}^\top C$ and rank its entries with their training-image and activation identities. For any retrieved source j, the terms $B_{i\ell}C_{ij}$ show how its weight is distributed across concept paths. A source can receive mass through several concepts, so summing before ranking captures contributions split across concept galleries.

Summing $P_{\ell j}$ over all retained locations from one source image gives that image's total mixture weight, and these image weights also sum to one. This supports inspection first by image and then by spatial location or token. These views can be computed for a queried row without storing the full test-by-training matrix.

Choosing how much evidence to display. Equation (7) gives a principled way to choose the length of a source list. Sort the entries of $P_{\ell,:}$ and retain the smallest prefix whose cumulative mass reaches a chosen threshold $\tau \in (0, 1]$. Its renormalized mixture has feature error at most $(1 - \tau)\Delta_A$ relative to the full reconstruction. Concentrated rows may reach the threshold with few sources; diffuse rows require longer lists. Thus a common mass threshold provides a consistent coverage criterion across test locations, while a fixed crop count can cover different fractions of their mixtures. An inspection record can include the source identities, original coefficients, retained mass, and reconstruction residual together. The residual describes the fit to the test activation, whereas retained mass describes the completeness of the displayed training evidence.

4 Experiment

4.1 Experimental setup

Following Fel et al. (2023); Colin et al. (2022); Sundararajan et al. (2017), we evaluate on CUB Wah et al. (2011) (200 fine-grained bird classes) and ImageNet100 images Deng et al. (2009), with two backbones: NF-ResNet50 Brock et al. (2021) and ViT-B/32 Dosovitskiy et al. (2021). The class-conditioned experiment compares CAD with PCA Jolliffe (1986). CRAFT Fel et al. (2023) is excluded because the ViT inputs violate its nonnegativity assumption and the exact NF-ResNet50 input admissibility remains unverified. A separate controlled CUB/NF-ResNet50 sweep over $k \in \{10, 20, 30, 40, 50, 75, 100\}$ and six random seeds examines sensitivity to the B-simplex constraint and full-image support.

Implementation details. The released CUB models use finetuned 200-way heads and true labels. ImageNet100 evaluation selects 100 mapped outputs of the pretrained 1,000-way head. NF-ResNet50 features are extracted before its head. The released ViT-B/32 path retains all 50 tokens, including CLS; its 5×10 reshape is a storage layout, not a grid of 50 image regions. Only the 49 patch tokens have spatial patch identities. CUB attributes come from `attributes/image_attribute_labels.txt`, using per-image `image_id`, `attribute_id`, and `is_present`; certainty and class-level attribute replacement are not used. Attributes evaluate retrieved supports after fitting, not the factorization objective.

CAD uses Adam on softmax logits. The main classwise results use seed 42 and report dispersion across classes. The controlled global CUB ablation uses seven concept counts and seeds 7–12. Its launcher sets training/test-solve iteration caps to 1000/300 for $k \leq 20$, 1200/350 for $30 \leq k \leq 50$, and 1500/400 otherwise, with learning rate 10^{-2}. The main classwise solver has a separate 3000-step default. Appendix F details the two protocols and their aggregation rules.

Historical train/test reconstruction aggregates appear separately in Appendix C, together with their unresolved normalization and weighting definitions.

4.2 Class-Conditional Reconstruction Score

Pairwise diagnostic (Table 1). Write $p_i = \operatorname{argmax}_y s_y(A_i)$ for the original prediction and y_i for the true label. The original diagnostic compares only $\{y_i, p_i\}$ under reconstructed logits. Its indicator I_i is one if y_i wins that pairwise comparison, with the same tie rule used in evaluation; $I_i = 1$ automatically if $p_i = y_i$. Define $a = N^{-1} \sum_i \mathbf{1}[p_i = y_i]$ and, when there are original errors, $q = \sum_{i:p_i \neq y_i} I_i / \#\{i : p_i \neq y_i\}$. Then

$$S_{\text{pair}} = N^{-1} \sum_i I_i = a + (1 - a)q, \qquad a \leq S_{\text{pair}} \leq 1. \tag{8}$$

Table 1: Gold-over-original pairwise preference (%): mean ± standard deviation across classes at seed 42. The score compares the true class with the original predicted class under reconstructed logits, as defined in Eq. (8); it is distinct from multiclass top-1 accuracy.

Data / backbone	k	CAD	PCA
CUB / NF-ResNet50	10	99.5 ± 5.4	34.5 ± 40.3
	20	99.5 ± 4.3	44.6 ± 38.8
	30	99.4 ± 4.9	52.3 ± 36.4
	40	99.4 ± 5.4	54.1 ± 36.3
	50	99.3 ± 5.3	56.1 ± 34.6
CUB / ViT-B/32	10	99.6 ± 2.5	65.6 ± 34.5
	20	99.8 ± 1.8	64.7 ± 31.8
	30	100.0 ± 0.3	65.2 ± 29.2
	40	99.9 ± 0.5	65.1 ± 29.1
	50	100.0 ± 0.0	66.5 ± 27.8
ImageNet100 / NF-ResNet50	10	99.9 ± 0.8	95.0 ± 19.0
	20	99.9 ± 0.9	93.5 ± 20.0
	30	99.8 ± 1.0	95.1 ± 15.8
	40	99.8 ± 1.3	95.3 ± 12.7
	50	99.6 ± 1.2	95.4 ± 13.8
ImageNet100 / ViT-B/32	10	97.6 ± 8.8	90.7 ± 19.5
	20	98.5 ± 5.1	89.8 ± 16.8
	30	99.0 ± 3.9	90.5 ± 12.8
	40	99.3 ± 1.9	88.1 ± 16.2
	50	99.3 ± 2.9	88.1 ± 16.7

For the class-macro table, let G be the number of evaluated classes and S_g, a_g, q_g the corresponding classwise quantities. Then

$$S_{\text{macro}} = \frac{1}{G} \sum_g S_g, \qquad S_g = \begin{cases} a_g + (1 - a_g) q_g, & a_g < 1, \\ 1, & a_g = 1. \end{cases}$$

The class-macro score averages the classwise quantities S_g, retaining the within-class products $(1 - a_g)q_g$ where $a_g < 1$. Original and reconstructed predictions use the evaluated 200 CUB outputs or the selected 100 ImageNet outputs, and $\mathbf{1}[\cdot]$ denotes the indicator. Equation (8) separates the contribution of originally correct examples from pairwise preference on original errors. When $a = 1$, the score is one regardless of the reconstructed logits and q is undefined. Thus Table 1 measures pairwise preference, not prediction preservation or multiclass error correction. The archived tie policy remains unresolved, and computing numerical q values requires the unavailable original error counts.

The class-conditioned protocol uses the true class to select its decomposition, so it evaluates reconstruction with class information available rather than deployment-time classification. Full-label-space behavior is measured separately by reconstructed top-1, $N^{-1} \sum_i \mathbf{1}[\text{argmax}_y \, s_y(\hat{A}_i) = y_i]$, and original-prediction agreement, $N^{-1} \sum_i \mathbf{1}[\text{argmax}_y \, s_y(\hat{A}_i) = p_i]$. These metrics are reported for the global ablation below.

When reconstruction preserves a decision. For a frozen linear head $s_c(z) = w_c^\top z + b_c$, let p be the unique original prediction and $e = \hat{z} - z$. A sufficient condition for agreement is

$$s_p(z) - s_c(z) > \|w_p - w_c\|_* \, \|e\| \quad \text{for all } c \neq p. \tag{9}$$

Here z is the vector entering the affine head, including any pooling or token aggregation applied to A. The condition compares each image's reconstruction error with its original classification margins. Applying it requires per-image margins and head-input errors; the reported average MSE does not certify individual predictions. Appendix E gives the proof, an extension to Lipschitz logits, and an example showing that feature-error and logit-error orderings can differ. Table 2 therefore reports feature MSE, reconstructed top-1, and agreement as separate diagnostics.

Table 2: Controlled CUB/NF-ResNet50 ablation, mean over seven concept counts $k \in \{10, 20, 30, 40, 50, 75, 100\}$ and six seeds (42 settings per variant). MSE is per-entry squared feature error; agreement and top-1 are fractions. Agreement compares with the original prediction; top-1 compares with true CUB labels under the finetuned 200-way head.

Variant	B simplex	Support	MSE ↓	Agree. ↑	Top-1 ↑
Full CAD	Yes	full-image	159.24	0.129	0.103
No B simplex	No	full-image	3.77×10^5	0.010	0.006
Crop-based AA	Yes	crops	265.70	0.063	0.055

4.3 Ablation of CAD Design Choices

The controlled CUB/NF-ResNet50 ablation examines two structural choices: the B-simplex reconstruction constraint and full-image activation support. The sweep covers seven concept counts and six seeds (7–12), yielding 42 settings for each variant. Table 2 reports averages over all settings.

Ablation results. Full CAD has the lowest mean feature MSE and the highest mean agreement and top-1 among the three variants under the reported solver settings. Test MSE increases from 159.24 to 3.77×10^5 when the B simplex is removed, and to 265.70 with crop support. Reported per-setting times are 96.9 seconds for CAD and 127.3 seconds for crop-based AA. These results characterize the sensitivity of this implementation to normalization and support construction. Removing normalization enlarges the feasible set while changing the numerical parameterization; the observed test differences do not isolate optimization from generalization effects. Appendix B separately records the retrieval statistics and their unresolved matching definitions.

Unit of comparison. The class-conditioned comparison and the global ablation use different decomposition selection rules. We report them separately, with the sweep averages summarizing variation across concept counts, solver budgets, and seeds. Appendix F specifies the sampling and aggregation protocol.

5 Limitations

CAD provides provenance for a selected feature decomposition; semantic interpretation and causal influence require additional evidence. Convex-hull membership concerns activation geometry and does not ensure membership in the data manifold. The qualitative examples visualize ranked C supports; a held-out demonstration of the composed B_{test}/P interface and measurement of its inspection benefit remain open. The experiments cover two vision backbone families, with a global design ablation on CUB/NF-ResNet50 and no matched SAE/A-SAE sweep. The global agreement and top-1 values in Table 2 limit the use of these reconstructions as substitutes for the original predictor. Broader behavioral comparisons require matched evaluation protocols. For sensitive collections, inspecting retrieved training crops also requires appropriate data-access controls.

6 Conclusion

CAD turns an AA-derived factorization into an explicit test-location-to-concept-to-training-location interface for a frozen vision model. Retaining and composing the two coefficient matrices gives each reconstructed activation a normalized account of its training sources at any feasible iterate. The dictionary provides a shared vocabulary across test inputs, while the composed coefficients specify how individual reconstructions draw on that vocabulary and its underlying training evidence.

Fixed-coefficient perturbation and retained-mass bounds connect these inspection choices to changes in the reconstructed feature. In the controlled CUB ablation, full CAD achieves the lowest mean feature error among the tested design variants under the evaluated solver settings. The resulting interface makes the training sources of a learned reconstruction available for explicit inspection while preserving the original predictor as the object of study.

AI USE STATEMENT

Generative AI tools assisted with manuscript refinement, language clarity and readability, and LaTeX formatting. The authors are responsible for the final manuscript.

REPRODUCIBILITY STATEMENT

The supplementary archive contains the CAD implementation, entry-point notes, a dependency inventory, and a file manifest. Its code README lists the required data and checkpoints, external utilities, and entry-point issues that must be addressed before execution. Section 4.1 and Appendix F specify the solver settings and evaluation protocols for the reported results. Appendices B and C retain the historical statistics with their unresolved metric and aggregation definitions. Appendix D provides the proofs and their assumptions.

REFERENCES

Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International conference on machine learning*, pp. 1059–1071. PMLR, 2021.

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

Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. What I cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in neural information processing systems*, 35:2832–2845, 2022.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Chris Ding, Tao Li, and Michael I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, 2010.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

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 E. Ba, and Talia Konkle. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16543–16572. PMLR, 2025. URL https://proceedings.mlr.press/v267/fel25a.html.

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

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

Ian T Jolliffe. *Principal Component Analysis*. Springer, 1986.

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.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pp. 5338–5348. PMLR, 2020.

Neehar Kondapaneni, Oisin Mac Aodha, and Pietro Perona. Representational difference explanations. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 71478–71527. Curran Associates, Inc., 2025a. doi: 10.52202/085713-2402. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6794f555524c9069e26970a408d353cc-Paper-Conference.pdf.

Neehar Kondapaneni, Oisin Mac Aodha, and Pietro Perona. Representational similarity via interpretable visual concepts. *arXiv preprint arXiv:2503.15699*, 2025b.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14933–14943, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should I trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. Modeldiff: A framework for comparing learning algorithms. In *International Conference on Machine Learning*, pp. 30646–30688. PMLR, 2023.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

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.

A MORE VISUALIZATIONS

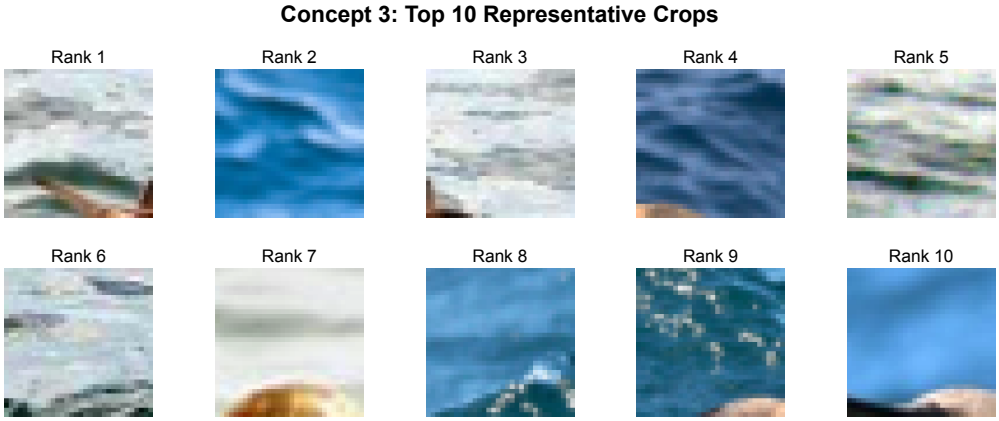

Figure 3: Visualization of CAD concept support on CUB (NF-ResNet50, k=30, seed 0): Concept 3. Crops are retrieved by descending C_{3j} from the learned index matrix and arranged in the same order as Figure 2. Each panel corresponds to one (image, spatial location) pair j entering the convex decomposition of concept z_3; labels indicate coefficient rank, not coefficient values. This seed-0 illustration is separate from the quantitative sweeps. The displayed crop need not cover the feature's complete receptive field.

We use Figure 3 to show the visualization results of another example on the CUB dataset.

B HISTORICAL RETRIEVAL STATISTICS

Table 3 records CAD top-K support statistics from the CUB/NF-ResNet50 controlled sweep. Attribute annotations belong to source images and are inherited by displayed crops, so their spatial scope is image-level. The matching objective, denominator, retrieval K, unmatched-concept handling, and zero-norm cosine policy remain unresolved. These entries are retained as historical records and are excluded from semantic-quality comparisons.

Table 3: Historical support statistics across concept counts, with the recorded means and standard deviations. Unique image ratio denotes the fraction of distinct source images. "Attribute precision" retains the recorded metric name, with unresolved matching and denominator definitions; it is distinct from the separately recorded "Hungarian AP" below. Pairwise cosine measures embedding similarity and ranges over $[-1, 1]$ for nonzero vectors.

k	Unique image ratio	Attribute precision	Pairwise cosine
10	0.985 ± 0.015	0.804 ± 0.040	0.999 ± 0.001
20	0.978 ± 0.013	0.830 ± 0.013	0.992 ± 0.003
30	0.961 ± 0.004	0.821 ± 0.022	0.982 ± 0.008
40	0.958 ± 0.008	0.847 ± 0.021	0.966 ± 0.009
50	0.956 ± 0.008	0.833 ± 0.020	0.944 ± 0.013
75	0.929 ± 0.008	0.837 ± 0.018	0.945 ± 0.013
100	0.923 ± 0.010	0.832 ± 0.009	0.921 ± 0.013

The ablation also records a statistic labelled "Hungarian AP": CAD 0.468, no-B-simplex 0.403, and crop-AA 0.461. Its exact definition and relation to the attribute-precision column remain unresolved, so it is retained separately under its recorded name.

C HISTORICAL RECONSTRUCTION AGGREGATES

Table 4: Historical reconstruction aggregates from the classwise concept-budget sweep, shown as recorded mean ± spread. Exact MSE normalization, class/budget weighting, and the meaning of the spread remain unresolved; the entries are excluded from inferential comparisons. $\Delta = \mathrm{MSE}_{\mathrm{test}} - \mathrm{MSE}_{\mathrm{train}}$ retains the recorded values, which may differ from subtraction of rounded entries. Feature scales differ across backbones. CRAFT is omitted for the input-admissibility reasons in Section 4.1.

Data / backbone	Method	Train	Test	Δ
CUB / NF-ResNet50	CAD	86.74 ± 22.37	111.48 ± 24.03	**24.73**
	PCA	5.56 ± 1.10	176.33 ± 40.48	170.78
CUB / ViT-B/32	CAD	1.14 ± 0.17	1.40 ± 0.16	**0.26**
	PCA	0.93 ± 0.30	1.80 ± 0.21	0.87
ImageNet100 / NF-ResNet50	CAD	0.89 ± 0.18	0.97 ± 0.19	**0.08**
	PCA	0.19 ± 0.03	1.20 ± 0.20	1.02
ImageNet100 / ViT-B/32	CAD	1.46 ± 0.16	1.58 ± 0.16	**0.12**
	PCA	1.06 ± 0.29	1.85 ± 0.20	0.79

D THEORETICAL VERIFICATION

Theorem 1 (Idealized exact block minimization)**.** *Let $A \in \mathbb{R}^{nL \times c}$ be fixed. Consider the CAD optimization problem*

$$\min_{B \in \mathcal{B},\, C \in \mathcal{C}} \; F(B, C) := \left\| A - B^\top C A \right\|_F^2, \tag{10}$$

where

$$\mathcal{C} = \left\{ C \in \mathbb{R}^{k \times nL} \;\middle|\; C_{ij} \geq 0,\ \sum_{j=1}^{nL} C_{ij} = 1,\ \forall i \right\}$$

is the row-wise simplex set for C, and

$$\mathcal{B} = \left\{ B \in \mathbb{R}^{k \times nL} \;\middle|\; B_{ij} \geq 0,\ \sum_{i=1}^{k} B_{ij} = 1,\ \forall j \right\}$$

is the column-wise simplex set for B. Then the alternating minimization scheme

$$B^{(t+1)} \in \operatorname*{argmin}_{B \in \mathcal{B}} F(B, C^{(t)}),$$
$$C^{(t+1)} \in \operatorname*{argmin}_{C \in \mathcal{C}} F(B^{(t+1)}, C).$$

generates a sequence $\{(B^{(t)}, C^{(t)})\}_{t \geq 0}$ such that:

1. *The objective value $F(B^{(t)}, C^{(t)})$ is monotonically non-increasing.*
2. *If additionally $\|B^{(t+1)} - B^{(t)}\|_F \to 0$, every accumulation point is stationary for problem* (10)*.*

The theorem establishes monotone objective values and, under the additional condition, stationary accumulation points; convergence of the entire parameter sequence requires further assumptions. It applies to exact block updates. The implemented Adam solver instead operates on softmax logits, whose finite values parameterize the relative interior of the simplices.

Proof. We first note that for fixed C, the map $B \mapsto B^\top (CA)$ is linear, so the objective $F(B, C)$ is a Frobenius squared loss of a linear function of B and is therefore convex in B:

$$F(B, C) = \|A\|_F^2 - 2\langle A, B^\top C A \rangle + \|B^\top C A\|_F^2.$$

Since the feasible set $\mathcal{B}$ (the column-wise simplex set) is a convex and compact polytope, the subproblem

$$\min_{B \in \mathcal{B}} F(B, C)$$

is a convex quadratic program with linear equality and inequality constraints, and thus admits at least one global minimizer. Similarly, for fixed B, the map $C \mapsto B^\top C A$ is linear, so $F(B, C)$ is convex in C, and the subproblem

$$\min_{C \in \mathcal{C}} F(B, C)$$

is also a convex quadratic program over the row-wise simplex (a compact convex polytope), and admits a global minimizer.

By construction of the alternating minimization scheme, we have

$$F(B^{(t+1)}, C^{(t)}) \leq F(B^{(t)}, C^{(t)}),$$
$$F(B^{(t+1)}, C^{(t+1)}) \leq F(B^{(t+1)}, C^{(t)}),$$

which implies that the sequence $\{F(B^{(t)}, C^{(t)})\}$ is monotonically non-increasing. Since $F(B, C) \geq 0$ for all feasible (B, C), the sequence is bounded from below and therefore convergent. Moreover, the feasible sets $\mathcal{B}$ and $\mathcal{C}$ are compact, implying that the sequence $\{(B^{(t)}, C^{(t)})\}$ admits at least one accumulation point. To prove the conditional stationary-point claim, take a convergent subsequence $(B^{(t_j)}, C^{(t_j)}) \to (B^*, C^*)$. Exact C minimization at the preceding iteration gives $F(B^{(t_j)}, C^{(t_j)}) \leq F(B^{(t_j)}, C)$ for all feasible C. Continuity yields $F(B^*, C^*) \leq F(B^*, C)$. Exact B minimization gives $F(B^{(t_j+1)}, C^{(t_j)}) \leq F(B, C^{(t_j)})$ for all feasible B. Under the additional vanishing-block-change condition, $B^{(t_j+1)} \to B^*$, so $F(B^*, C^*) \leq F(B, C^*)$. The two blockwise variational inequalities combine on the product feasible set to give first-order stationarity, equivalently the simplex-constrained KKT conditions. □

The following provenance identities hold at every feasible iterate, including intermediate Adam iterates, because they follow directly from the simplex constraints.

Theorem 2 (Convex and Sample-Traceable Concept Decomposition)**.** *Let $A \in \mathbb{R}^{nL \times c}$ denote the activation matrix extracted from a fixed intermediate layer of a neural network, and let $C \in \mathcal{C}$ be the concept index matrix obtained by CAD, where $\mathcal{C}$ is the row-wise probability simplex set. Then each concept vector $z_i \in \mathbb{R}^c$ constructed by CAD satisfies*

$$z_i = A^\top C_{i:}^\top = \sum_{j=1}^{nL} C_{ij} A_j^\top, \tag{11}$$

and is therefore a convex combination of activation vectors associated with training samples.

Moreover, the index matrix C establishes an explicit and recoverable correspondence between each concept z_i and its contributing training activations $\{A_{j,:} : C_{ij} > 0\}$. The retained coefficients and activation identities give every CAD concept image/activation-level provenance under the learned linear decomposition. Spatially indexed rows additionally provide region-level provenance.

Proof. By construction, the concept index matrix C is constrained to the row-wise simplex $\mathcal{C}$, i.e.,

$$C_{ij} \geq 0, \qquad \sum_{j=1}^{nL} C_{ij} = 1 \quad \forall i.$$

As a result, for each concept index i, the corresponding concept vector is given by

$$z_i = A^\top C_{i:}^\top = \sum_{j=1}^{nL} C_{ij} A_j^\top,$$

which is a convex combination of the activation vectors $\{A_j\}_{j=1}^{nL}$. Since each activation vector A_j originates from a retained activation/token identity of a specific training sample, the stored coefficients C_{ij} specify each sample-location pair's term in this particular decomposition. Different coefficient matrices can represent the same concept vector, so the coefficients need not be identifiable from z_i

alone. Consequently, the support of $C_{i:}$ provides an explicit and recoverable index set identifying the training samples and retained activation identities that constitute the concept. Therefore, each concept extracted by CAD admits a decomposition-level provenance grounded in the training data, without requiring post-hoc sampling or thresholding procedures, establishing image/activation-level traceability under the learned linear decomposition. Spatial rows additionally carry region identities; CLS does not. □

Theorem 2 identifies the training sources of each concept. The next result specifies their individual coefficients and the change in the concept vector induced by transferring weight between two sources.

Theorem 3 (Index Traceability and Coefficient-Level Attribution)**.** *Let $z_i \in \mathbb{R}^c$ be the i-th concept obtained by CAD and let $C_{i:} \in \mathbb{R}^{nL}$ denote the corresponding row of the concept index matrix. Then the mapping*

$$\phi_i : \{1, \dots, nL\} \to \mathbb{R}, \quad \phi_i(j) = C_{ij},$$

specifies a coefficient-level attribution measure from training activations to the concept z_i under the CAD decomposition, satisfying the following properties:

1. *(*Non-negativity*) $\phi_i(j) \geq 0$ for all j.*
2. *(*Normalization*) $\sum_{j=1}^{nL} \phi_i(j) = 1$.*
3. *(*Coefficient-transfer identity*) For any feasible simplex-preserving perturbation $C'_{i:} = C_{i:} + \delta(e_j - e_{j'})$ with $\delta > 0$, the concept changes by $z'_i - z_i = \delta(A_j^\top - A_{j'}^\top)$; in particular, raising C_{ij} at the expense of $C_{ij'}$ increases the relative contribution of A_j versus $A_{j'}$ to z_i.*

Consequently, each index value C_{ij} specifies the exact coefficient of training activation A_j (and its region when a spatial identity exists) in the convex combination defining z_i, and ranking by $\phi_i(j)$ orders training activations by their decomposition weights.

Remark. Ranking by ϕ_i orders training activations by their coefficients in the selected decomposition. A deterministic display can break ties using the retained source index. This coefficient ordering is distinct from vector-contribution magnitude, classifier effect, and semantic relevance.

Proof. From the CAD construction, each concept z_i is generated as a linear mapping

$$z_i = \mathcal{L}_A(C_{i:}) := A^\top C_{i:}^\top,$$

where $\mathcal{L}_A : \mathbb{R}^{nL} \to \mathbb{R}^c$ is linear in $C_{i:}$.

Because $C_{i:}$ lies on the probability simplex, we have

$$C_{ij} \geq 0 \quad \text{and} \quad \sum_{j=1}^{nL} C_{ij} = 1,$$

which directly implies non-negativity and normalization of the attribution scores.

To establish the coefficient-transfer identity, consider a feasible perturbation of the index vector along coordinates j and j' that preserves the simplex constraint:

$$C'_{i:} = C_{i:} + \delta e_j - \delta e_{j'}, \quad \delta > 0,$$

where e_j is the standard basis vector and j' is chosen such that $C_{ij'} \geq \delta$. Then by linearity of $\mathcal{L}_A$, the induced change in the concept representation is

$$z'_i - z_i = \delta(A_j^\top - A_{j'}^\top),$$

which shows that, holding the simplex constraint fixed, raising C_{ij} at the expense of $C_{ij'}$ moves z_i along the direction $A_j^\top - A_{j'}^\top$, i.e. increases the relative contribution of A_j versus $A_{j'}$ to z_i.

Since each activation A_j corresponds to a retained activation/token identity within a specific training image, the index value C_{ij} specifies the coefficient with which that training-activation enters the convex decomposition of z_i. Furthermore, for any two activations j_1 and j_2, the inequality $C_{ij_1} > C_{ij_2}$ means that A_{j_1} enters the convex decomposition of z_i with a strictly larger weight than A_{j_2}, yielding an ordering by decomposition weight, with possible ties. □

E COMPOSED PROVENANCE AND DECISION BOUNDS

For $D = B_{\text{test}}^{\top}$, the simplex constraints give $D \geq 0$, $C \geq 0$, $D\mathbf{1} = \mathbf{1}$, and $C\mathbf{1} = \mathbf{1}$. Therefore $P = DC$ is nonnegative and $P\mathbf{1} = D(C\mathbf{1}) = \mathbf{1}$. Expanding the products proves Eq. (5). If training row j changes by e_j while D, C are fixed, the triangle inequality gives, for any norm,

$$\|\widetilde{\hat{A}}_{\ell,:} - \hat{A}_{\ell,:}\| = \left\| \sum_j P_{\ell j} e_j \right\| \leq \sum_j P_{\ell j} \|e_j\| \leq \max_j \|e_j\|.$$

This convex-averaging bound applies with the fitted coefficients, support identities, and encoder fixed.

For the linear-head claim in Eq. (9), each competing class satisfies

$$s_p(\hat{z}) - s_c(\hat{z}) = s_p(z) - s_c(z) + (w_p - w_c)^{\top} e \geq s_p(z) - s_c(z) - \|w_p - w_c\|_* \|e\| > 0.$$

The result applies to the actual vector entering the head; when A is spatial, pooling or token aggregation must be included in the map from A to z. More generally, if each logit is L_c-Lipschitz in the selected representation, a sufficient condition is $s_p(z) - s_c(z) > (L_p + L_c)\|e\|$ for every $c \neq p$.

Neither bound ranks observed logit errors across methods. For example, let $W = (1, 0)$, $e_1 = (1, 0)^{\top}$, and $e_2 = (0, 2)^{\top}$. Then $\|e_1\|_2 < \|e_2\|_2$, but $\|We_1\|_2 = 1 > 0 = \|We_2\|_2$. A feature reconstruction can therefore have lower error and a worse head-level discrepancy.

F EXPERIMENTAL PROTOCOLS AND EVALUATION DETAILS

Class-conditioned reconstruction. The main pairwise experiment uses the true class to select the activation subset and its class-specific decomposition. Table 1 reports the mean and standard deviation across classes at seed 42. This protocol evaluates reconstruction with class information available; the reported dispersion measures class-to-class variation. The historical reconstruction aggregates in Appendix C are reported separately because their exact MSE normalization and class/budget weighting remain unresolved.

Global CUB ablation. The controlled CUB/NF-ResNet50 ablation fits a global decomposition for each of seven concept counts, $k \in \{10, 20, 30, 40, 50, 75, 100\}$, and six seeds, 7–12, yielding 42 settings per variant. The launcher requests at most 30 training and 30 test images per class. Training/test-solve iteration caps are 1000/300 for $k \leq 20$, 1200/350 for $30 \leq k \leq 50$, and 1500/400 for larger k, with learning rate 10^{-2}. The sweep therefore varies both representation capacity and solver budget. Its averages summarize performance across these settings; they do not estimate uncertainty at a fixed concept count. The global and class-conditioned protocols use different decomposition selection rules and are reported separately.

Ablation metrics and aggregation. The ablation uses a CUB-finetuned 200-way classification head. Feature MSE is the mean squared error per feature entry. Reconstructed top-1 accuracy compares predictions from reconstructed features with true CUB labels, while prediction agreement compares them with the original model predictions; both are reported as fractions. Table 2 averages each metric over the 42 settings. The full-CAD means are 159.235446 for feature MSE, 0.129210 for agreement, and 0.102586 for top-1, rounded to 159.24, 0.129, and 0.103 in the table.

Attribute and spatial identities. CUB attributes are indexed by image identity using the `image_id`, `attribute_id`, and `is_present` fields in `attributes/image_attribute_labels.txt`. Certainty fields and class-majority replacement are not used. Attribute annotations enter retrieval evaluation after fitting; true-class selection separately defines the class-conditioned reconstruction protocol. The ViT cache retains CLS and 49 patch tokens in a 5×10 storage layout. Only the 49 patch tokens carry spatial identities for image-region retrieval; CLS has an image identity but no corresponding image region.

Baseline scope. CRAFT requires nonnegative input features. The signed ViT caches violate this requirement, and admissibility of the exact NF-ResNet50 caches remains unverified. CRAFT scores are therefore excluded from the comparisons.