MOSAIC: Concept Bottlenecks via Text-Anchored Optimal Transport

Sannara Ek
University of Cambridge
Cambridge, United Kingdom
sannara.ek@cl.cam.ac.uk

Chong Tang*
University of Bristol
University of Southampton
Bristol, United Kingdom
chong.tang@bristol.ac.uk

Dirk Koch
University of Manchester
Manchester, United Kingdom
dirk.koch@manchester.ac.uk

Alex Weddell
University of Southampton
Southampton, United Kingdom
asw@ecs.soton.ac.uk

Jagmohan Chauhan
University College London
UCL AI Centre
London, United Kingdom
jagmohan.chauhan@ucl.ac.uk

Robert Mullins
University of Cambridge
Cambridge, United Kingdom
robert.mullins@cl.cam.ac.uk

Abstract

Concept Bottleneck Models (CBMs) provide interpretability but typically incur a penalty in accuracy. Recent CBMs rely on LLM-supplied concept vocabularies that are misaligned with the underlying image evidence and obscure dataset-specific discriminative signals, and reach competitive accuracy only through a trained task-specific linear probe. Methods that instead learn concepts directly from data require sparse-autoencoder pretraining and also classify through a trained probe. We introduce **MOSAIC** (Mixtures Of Sparse Anchored Interpretable Concepts), a CBM whose concepts are discovered directly from image representations via text-anchored optimal transport, a balanced-marginal pass that softly assigns each image to a small set of learned concepts while keeping them tethered to class-text embeddings, and then named post-hoc by a VLM. Across 5 datasets, MOSAIC comes within 2.5 pp of the supervised CLIP linear probe on 8 of 10 dataset-backbone settings (exceeding it on one; scene classification remains the notable exception) and beats raw zero-shot CLIP by $+5.2$ to $+24.4$ top-1 and class-level retrieval R@1 by $+5.7$ to $+27.0$. Beyond classification, the discrete concept basis enables a sparse per-image decomposition that exposes which features drive each prediction and bottleneck compression to as little as 2.5% of dense CLIP.

1 Introduction

Concept Bottleneck Models (CBMs) classify an image by predicting a set of human-interpretable concepts and mapping them to a class label, replacing the opaque end-to-end classifier with a transparent intermediate representation. This intermediate layer makes every prediction auditable at the concept level, supports targeted intervention on incorrect concepts, and permits compositional queries over named visual features. These properties matter most for high-stakes decisions, such as medical diagnosis, scientific claims, or regulated deployment, where a stakeholder must verify, contest, or correct the prediction.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, existing CBMs typically source their concepts in ways that decouple the bottleneck from the actual data. Many language-driven methods predefine a vocabulary via Large Language Models (LLMs) before seeing a single image Oikarinen et al. [2023], Yang et al. [2023], Ozdemir et al. [2026]. Sourced from text alone, these concepts often miss the fine-grained visual primitives that exist in the target dataset, discarding the discriminative signal they carry. Other methods learn concepts from data but train a supervised linear probe on the concept activations to recover competitive accuracy Rao et al. [2024], Yuksekgonul et al. [2022], Koh et al. [2020], introducing a separate failure mode where the probe weights collapse the explanation onto a few class-correlated concepts, turning the concept layer into a linear proxy for class identity rather than an inventory of inspected visual primitives.

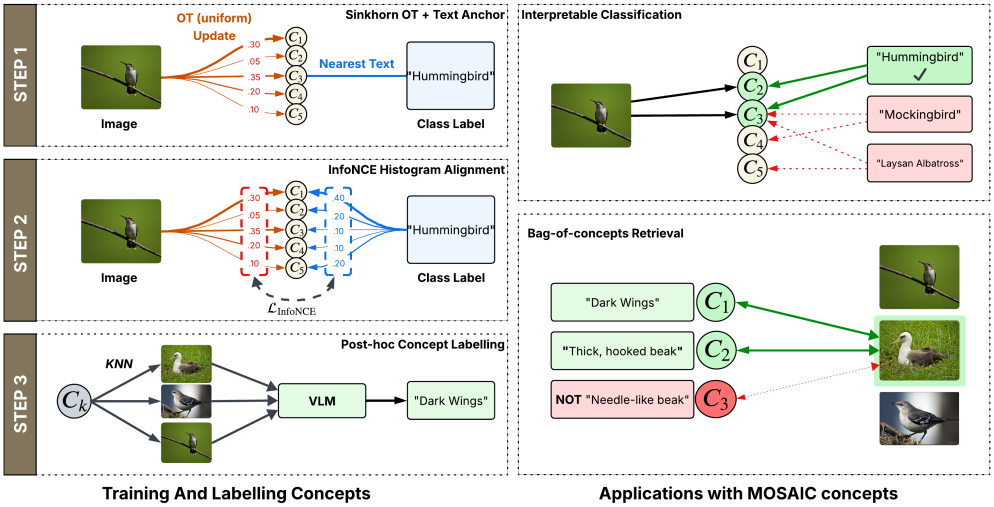

Figure 1: **MOSAIC overview. Step 1** solves a balanced-marginal Sinkhorn OT between image features and K concept directions, each blended toward its nearest class-text anchor. **Step 2** sharpens the codebook by aligning each image's top-k concept histogram with its class-text histogram via InfoNCE. **Step 3** names every concept with a VLM prompted with nearest-neighbour images. Inference encodes an image as a sparse concept histogram and classifies by cosine similarity against per-class histograms.

Both failure modes share one cause, concept axes imposed on the target data rather than learned from it. MOSAIC (Mixtures Of Sparse Anchored Interpretable Concepts) closes this gap by discovering a basis of fine-grained visual concepts directly from the target dataset's image features, anchored to CLIP's text directions, and replacing the trained classifier head with cosine similarity between sparse concept histograms (Figure 1).

MOSAIC decomposes every image into a short, weighted list of active features. Classification is cosine similarity between sparse TF-IDF concept histograms shared by images and class names, with no per-class learned parameters. Each concept is described post-hoc by a Vision-Language Model (VLM) grounded in the images, producing visual descriptions such as *iridescent green plumage* or *curved hooked beak* rather than lexicon-bound class names like *parrot*. This basis supports three capabilities prior CBMs demonstrate only in isolation, namely compositional Boolean retrieval over signed concept queries, audit-trail classification (where every prediction decomposes into named concepts whose captions are independently scored by VLM judges), and concept-level intervention for diagnosing or editing model dependencies on visual primitives.

Empirically, the combination comes within 2.5 pp of the supervised CLIP linear probe on 8 of 10 dataset–backbone settings (exceeding it on one; Places365 is the notable exception, App. A) while compressing to as little as 2.5% of dense CLIP's storage footprint, delivering simultaneously the accuracy of a probe-based CBM and the per-prediction transparency a head-free cosine inference makes possible.

Our contributions are as follows:

- **A data-first concept codebook for CBMs.** We propose MOSAIC, a CBM whose concept basis is discovered from the target dataset's image features rather than predefined externally. Balanced-marginal optimal transport forces even class-seeded centroids to claim image evidence, so concepts encode visual variation rather than class identity, and a small text-anchor blend keeps each concept close to a natural-language direction in CLIP space (Sec. 3.1).
- **Classification by cosine on concept histograms.** Images and class names project into the same codebook as sparse top-k concept histograms, and classification is a single cosine on those histograms with no per-class parameters. An InfoNCE loss aligns image and class-text histograms during training, and each concept is named post-hoc by a VLM from top activating images of six different classes, with names selected to separate each concept from its neighbours (Sec. 3.2, 3.3, 3.4).
- **Evaluation across classification and retrieval.** Across five datasets and two CLIP backbones, MOSAIC outperforms every CBM baseline, comes within 2.5 pp of CLIP-LP on $8/10$ dataset–backbone settings without training a classifier head, and beats class-level CLIP retrieval R@1 by $+5.7$ to $+27.0$ across three datasets and two directions. Three independent VLM judges validate the concept names, and the concepts align with human-annotated attributes 7–$9\times$ more often than those of DN-CBM and SpLiCE. The same pipeline transfers without changes to attribute retrieval and a non-CLIP backbone (Sec. 4, 6, App. O, Q, V).

2 Related Work

2.1 Concept Bottleneck Models

CBMs replace a dense feature vector by activations on a discrete, human-meaningful concept basis before classification, so each prediction decomposes into a short list of named concepts. The original CBM Koh et al. [2020] relies on per-image human concept annotations, the strongest interpretability guarantee but also the strongest supervision requirement, and does not scale beyond datasets that already ship with a concept ontology. Follow-up work flagged concept-leakage and reliability pitfalls of supervised CBMs Margeloiu et al. [2021], Mahinpei et al. [2021] and proposed soft-concept extensions Zarlenga et al. [2022]. The dominant pattern still removes per-image supervision while keeping the bottleneck, leaving open where the concepts come from. Two approaches dominate the modern CBM landscape, detailed in Sec. 2.2. The first **describes first, trains later**, encoding LLM-defined concepts upfront. The second **decomposes first, names later**, decomposing frozen features and then assigning names by nearest-neighbour lookup against a fixed lexicon. MOSAIC **trains first, describes later**, training the codebook by histogram alignment and then naming each concept with a VLM grounded in activating images.

2.2 Two approaches: generated vs. learned concept vocabularies

Concepts from external sources. LF-CBM Oikarinen et al. [2023] and LaBo Yang et al. [2023] query a large language model for a list of attribute strings per class, while P-CBM Yuksekgonul et al. [2022] draws its concepts from ConceptNet. All three encode the strings with CLIP's text tower and use the resulting set as the bottleneck. Two characteristic flaws follow. First, **external-source hallucination**, since the supplied strings need not correspond to features the dataset's images actually contain Bhalla et al. [2024], Ramaswamy et al. [2023], missing the discriminators that fine-grained classification depends on. Second, **a per-task linear-probe step is mandatory** to recover competitive accuracy, adding compute and hyper-parameter search (App. C).

Concepts learned from data. Motivated by the finding that pre-trained VLMs already encode visually grounded concepts recoverable post-hoc from their feature space Zang et al. [2025], a second line of work relaxes the external-vocabulary requirement. DN-CBM Rao et al. [2024] pretrains a sparse autoencoder on an external image corpus and names the resulting neurons via CLIP-text matching, classifying through a trained linear probe (direct head-to-head comparison in App. B); M-CBM Santis et al. [2026] follows a related SAE/mechanistic-explanation route, deriving its CBM from per-neuron interpretations of a frozen network. Both label fixed model directions rather than learning a basis tailored to the target dataset. SpLiCE Bhalla et al. [2024] keeps the basis fixed (a CLIP-text dictionary) and instead learns a per-image sparse code at inference via a LASSO solve. EZPC Ozdemir et al. [2026] keeps the basis fixed (LF-CBM's LLM-generated concept lists, un-

changed) and learns a CLIP-feature projection regularised toward identity, yielding a near-identity map and a dense representation (no compression). Each pays a price at inference: SAE pretraining plus a linear probe (DN-CBM, M-CBM), a slow per-image solver (SpLiCE), or a fixed-size LLM-string basis (EZPC).

We argue that an interpretable CBM should follow three principles implied by these flaws. The concept basis should be discovered from target-domain images rather than supplied by an LLM, so it cannot hallucinate features absent from the data. Classification should be a single cosine similarity over a sparse, named-concept representation, removing the need for a per-task probe. And the number of concepts should be a free parameter, so the codebook can be made smaller or larger without changing how inference works. MOSAIC realises these via image-feature optimal transport (Sec. 3.1), head-free TF-IDF cosine on concept histograms (Sec. 3.4), and an exposed codebook size K (App. I).

3 Method

We present MOSAIC, a data-first CBM with a two-stage training pipeline followed by a post-hoc concept-labelling step. Stage 1 produces an initial codebook of K centroids capturing well-spread visual features in the training data, via an optimal-transport (OT) assignment whose balanced marginals force mass to spread equally across all K centroids in every batch; each centroid is then blended with its nearest class-text anchor to keep the basis aligned with natural-language directions. Stage 2 sharpens the codebook for class discrimination via an InfoNCE contrastive loss between image-side and class-text concept histograms. After training, each concept is named by a VLM prompted with its top activating images, choosing among sampled candidates the name that best separates it from the other concepts (Sec. 3.3). Classification is a single cosine similarity between the image's sparse top-k concept histogram and a class-text histogram, so no classifier weights are learned and a new class is added by encoding its name.

Setup. Let $\phi_I : \mathcal{X} \to \mathbb{R}^D$ and $\phi_T : \mathcal{T} \to \mathbb{R}^D$ denote the frozen CLIP image and text encoders. Given a training set $\{(x_i, y_i)\}_{i=1}^N$ with class names $\{c_1, \ldots, c_C\}$, we compute L_2-normalised embeddings $X_i = \phi_I(x_i)$ and prompt-ensembled class embeddings $T_c = |\Pi|^{-1} \sum_{\pi \in \Pi} \phi_T(\pi(c))$ averaged over the standard 80-template ensemble Π from CLIP Radford et al. [2021] (e.g. *a photo of a c*, *a sculpture of a c*). MOSAIC learns a codebook $C \in \mathbb{R}^{K \times D}$ of text-aligned concept directions, where each of the K centroids becomes a named concept after the post-hoc labelling step. We use *centroid* for the geometric vector inside Stage 1 OT and *concept* for the same codebook entry in its named, interpretable role. Figure 1 (left) summarises the pipeline.

3.1 Text-Anchored Optimal Transport

Initialisation. We initialise the codebook from two complementary sources. The first $\min(K, C)$ slots are seeded with the class-text embeddings $\{T_c\}$, often the dataset's class labels, providing a language-meaningful starting point; when $K < C$ we take the first K embeddings in the dataset's class-index order. The remaining $\max(0, K - C)$ slots are filled with K-means centroids of the training image features, giving early coverage of the data distribution. The combination ensures the codebook starts both linguistically grounded and spread across the feature manifold.

Update via optimal transport. A naive nearest-centroid update lets the codebook collapse to one centroid per class on datasets with many examples per class, producing a class-mean basis equivalent to a linear probe. We avoid this by solving an entropy-regularised OT assignment between each minibatch and the codebook Caron et al. [2020]. For batch $B = \{X_j\}_{j=1}^{|B|}$ we build the cost matrix $\mathbf{M}_{kj} = (1 - \cos(c_k, X_j))/\tau_{\text{cost}}$ ($\tau_{\text{cost}} = 0.1$) and solve

$$\mathbf{P}^\star = \arg \min_{\mathbf{P} \in \mathcal{U}(a,b)} \langle \mathbf{P}, \mathbf{M} \rangle \; - \; \varepsilon H(\mathbf{P}), \tag{1}$$

with marginals $a = \mathbf{1}/K$, $b = \mathbf{1}/|B|$ and $\varepsilon = 0.01$, via Sinkhorn iterations on GPU Cuturi [2013] (20 iters suffice). The uniform a marginal is the principal constraint, forcing every centroid to receive equal assignment mass in every batch. This matters most for the class-seeded centroids: a class-text seed on its own never wins an assignment, and its equal share pulls it into the image manifold, so the codebook encodes within-class variation (colour, pose, viewpoint, part shape) rather than

class identity (App. S). The uniform marginal is also the maximum-entropy prior over codeword usage Asano et al. [2020], Caron et al. [2020]; it shapes per-batch update targets, filtered through the EMA below, rather than forcing uniform usage in the trained codebook (App. U). The OT target for centroid k is the assignment-weighted batch mean $\mathbf{c}_k^{\text{ot}} = \sum_j \tilde{\mathbf{P}}_{kj}\, X_j$, where $\tilde{\mathbf{P}}_{kj} = \mathbf{P}^\star_{kj} / \sum_{j'} \mathbf{P}^\star_{kj'}$ is the row-normalised optimal plan, which feeds the EMA update below.

Text-anchor blend. A purely image-driven OT update can drift centroids toward directions without a clean linguistic counterpart, since the image-feature manifold spans compositions and textures that lack natural-language analogues. We ground the codebook by blending each centroid toward its nearest class-text embedding. For every centroid c_k we find $a_k = T_{\arg\max_c \cos(c_k, T_c)}$ via FAISS Johnson et al. [2019] inner product and apply the blend

$$\hat{c}_k \;=\; (1-\beta)\,\mathbf{c}_k^{\text{ot}} \;+\; \beta\, a_k, \qquad \beta = 0.3. \tag{2}$$

The small β keeps the anchor influence weak, so the blend grounds the codebook in natural-language directions without undoing the mass-spreading effect of the uniform OT marginal. The blended target $\hat{c}_k$ is the per-step EMA target used below. Sec. 5 ablates this term, a multi-seed sweep over $\beta \in \{0.0, \ldots, 1.0\}$ confirms the choice (App. R), and a per-centroid audit confirms that no centroid collapses onto a class text (App. T).

Overall Stage-1 update. Composing the OT target (1), the text-anchor blend (2), and an EMA, the per-step update for centroid k in Stage 1 is

$$c_k \;\leftarrow\; \pi_{\mathbb{S}}\big[\,(1-\alpha_0)\,c_k \;+\; \alpha_0\,\hat{c}_k\,\big], \tag{3}$$

where $\hat{c}_k$ is the blended target from (2), $\pi_{\mathbb{S}}[\,\cdot\,]$ projects onto the unit sphere, and $\alpha_0 = 0.1$ is the EMA step size, analogous to momentum-encoder updates in self-supervised learning He et al. [2020].

3.2 InfoNCE Contrastive Refinement

Stage 1 shapes the codebook only against the image side, giving a language-aligned geometry that scaffolds the concept basis but does not tie an image's projection through C to its class-text counterpart. Stage 2 closes this gap with an InfoNCE objective that aligns image- and class-text concept histograms, directly optimizing the cosine that classification uses at inference.

Concepts as a soft mixture. We project both an image embedding X_i and a class-text embedding T_c through the codebook into the same K-dimensional space, where the k-th entry of each histogram is the soft assignment of the modality to concept k,

$$\mathbf{h}_i^{\text{img}} = \text{softmax}(X_i C^\top / \tau_{\text{hist}}), \quad \mathbf{h}_c^{\text{txt}} = \text{softmax}(T_c C^\top / \tau_{\text{hist}}), \tag{4}$$

with $\tau_{\text{hist}} = 0.05$, so each image and each class is represented by a distribution over interpretable concepts and classification reduces to matching distributions.

Histogram matching via InfoNCE. To align each image-histogram with its own class-text histogram and separate it from those of other classes, we use an InfoNCE contrastive loss Oord et al. [2018] over the histogram cosines. Writing $\tilde{\mathbf{h}}_i^{\text{img}} = \mathbf{h}_i^{\text{img}} / \|\mathbf{h}_i^{\text{img}}\|_2$ and $\tilde{\mathbf{h}}_c^{\text{txt}} = \mathbf{h}_c^{\text{txt}} / \|\mathbf{h}_c^{\text{txt}}\|_2$ for the L_2-normalised histograms,

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\tilde{\mathbf{h}}_i^{\text{img}} \cdot \tilde{\mathbf{h}}_{y_i}^{\text{txt}} / \tau)}{\sum_c \exp(\tilde{\mathbf{h}}_i^{\text{img}} \cdot \tilde{\mathbf{h}}_c^{\text{txt}} / \tau)}, \qquad \tau = 0.07. \tag{5}$$

Only the concepts C are trained; both CLIP encoders remain frozen.

3.3 Post-hoc Concept Labelling

Each trained centroid c_k is named after training, in two steps (App. E). **Candidates.** We take the top-activating training image of each of the six classes on which c_k fires most, so the shared property cannot be the class identity, and prompt a vision-language model with the grid for a short descriptive phrase naming the visual feature they share. Besides one greedy *first-pass* caption, we sample five candidates from a prompt that lists the names of the four nearest concepts as already taken.

Selection. For a name t, let $\rho(t, k)$ be the Spearman correlation over test images between the CLIP similarity of t to the image's annotated object region and the activation of concept k. The score $s(t, k) = \rho(t, k) - \text{mean}_{j \neq k}\, \rho(t, j)$ rewards a name that tracks its own concept and no other; names are assigned one-to-one to maximise total score, each concept receiving only its own candidates (App. E.1). Naming leaves the codebook untouched, so classification is unaffected. Figure 2 shows three example centroids with their top activating images.



Figure 2: **Example concepts.** One concept per dataset with its top activating images and bold class labels, showing each concept captures a coherent visual feature across different object categories. Labels show each concept's first-pass caption (App. E).

Caption fidelity. The labelling step is informative only if each caption faithfully describes its centroid. We audit every concept name with three independent vision-language models that score the name against its six-class grid on a 0–4 fidelity scale, calibrated against a derangement control where each grid is paired with a different centroid's caption (Sec. 6, App. N). Selection is judge-free; on CUB it raises distinct names from 305 to 437 of 506 (App. E).

3.4 Inference: TF-IDF Weighted Concept Cosine

At test time we replace the dense softmax of Eq. (4) with a top-k encoder applied to both images and class texts, $\mathbf{h}(v) = \text{top-k-softmax}(vC^\top/\tau_{\text{hist}})$ with $k_{\text{enc}} = \max(16, K/8)$. A few concepts activate on almost every image and carry little class-discriminative signal, so we re-weight each histogram by an IDF vector Sparck Jones [1972] $\text{idf}[k] = \log \frac{N+1}{\text{df}[k]+1} + 1$, with $\text{df}[k] = |\{i : \mathbf{h}(\phi_I(x_i))[k] > 10^{-2}\}|$ counted over the training set's top-k_{enc} histograms, and classify by cosine similarity,

$$\hat{y}(x) \;=\; \arg\max_{c}\, \cos\big(\text{idf} \odot \mathbf{h}(\phi_I(x)),\; \text{idf} \odot \mathbf{h}(T_c)\big). \tag{6}$$

No classifier head is learned and no per-class parameters are introduced. Sensitivities of k_{enc}, the τ_{hist} temperature and the IDF presence threshold are reported in App. H.

4 Interpretable Classification and Retrieval

Setup. We evaluate on five datasets (CUB-200 Wah et al. [2011], CIFAR-100 Krizhevsky et al. [2009], ImageNet-1k (IN-1k) Russakovsky et al. [2015] and the ImageNet-100 (IN-100) 100-class subset, Places365 Zhou et al. [2017]) and two CLIP backbones (ViT-L/14-336, ViT-B/16 Dosovitskiy et al. [2021]). All methods share the same 80-prompt class-text ensemble and frozen CLIP features; classification reports top-1 on the official test split. For retrieval we additionally evaluate class-level image↔text retrieval on CUB-200, IN-100 and IN-1k. Image→text ranks the class-text bank (200, 100 and 1000 entries) for 2000 sampled test images, and text→image ranks all test images for each class text, both by IDF-weighted cosine similarity. Full hyperparameters and training procedure are in App. G.

Baselines. We compare against CBMs whose final prediction is a single cosine similarity with no learned classifier head, **CLIP-ZS** Radford et al. [2021], **Z-CBM** (our zero-shot baseline scoring cosine similarity to a fixed concept vocabulary), **SpLiCE-ZS** Bhalla et al. [2024], and **EZPC** Ozdemir

Table 1: **Top-1 classification accuracy (%) across five datasets and two CLIP backbones.** The five *cosine-classifier* columns share the same inference rule: cosine similarity with no learned classifier head (per-baseline details in Sec. 2.2, 4). The two *Reference* columns use a trained classifier head: **LP-CBM** reports the strongest LP-trained CBM per dataset (full breakdown in App. C); **CLIP-LP** is the supervised linear-probe ceiling on the same backbone. All MOSAIC entries are reported at seed 42; a 3-seed verification on this exact protocol (App. R) reproduces every ViT-L/14-336 cell to within 0.3 pp.

		Cosine classifier (no trained head)						Reference (trained head)	
Dataset	Backbone	CLIP	Z-CBM	SpLiCE-ZS	EZPC	**MOSAIC**	ΔLP	LP-CBM	CLIP-LP
CUB-200	L/14-336	61.9	16.2	4.6	62.3	**86.4**	−0.5	78.4	86.9
	B/16	55.0	10.8	4.9	55.1	**78.5**	−2.4	78.4	80.9
CIFAR-100	L/14-336	77.8	49.3	72.3	77.7	**86.2**	−0.3	79.8	86.5
	B/16	68.4	40.2	65.6	68.5	**80.5**	−1.2	79.8	81.7
IN-100	L/14-336	84.8	50.6	82.2	84.7	**90.0**	−0.8	—	90.8
	B/16	77.9	42.9	75.4	78.0	**86.7**	**+0.3**	—	86.4
IN-1k	L/14-336	73.6	30.2	71.5	73.7	**81.7**	−1.8	72.0	83.5
	B/16	65.0	23.0	62.9	65.1	**76.5**	−1.2	72.0	77.7
Places365	L/14-336	38.0	20.4	35.7	37.9	**46.9**	−6.6	45.7	53.5
	B/16	36.3	15.4	34.9	36.3	**46.4**	−6.2	45.7	52.6

et al. [2026]. Of these, MOSAIC and EZPC train per-dataset representations; the rest use only frozen pretrained components. Two reference columns use a trained classifier head, **LP-CBM** (the strongest LP-trained CBM per dataset, drawn from LF-CBM, LaBo, P-CBM, DN-CBM; breakdown in App. C) and **CLIP-LP**, the supervised linear-probe ceiling. CoOp Zhou et al. [2022b] (App. P) is reported separately.

Classification result. Table 1 shows MOSAIC comes within 2.5 pp of CLIP-LP on 8 of 10 dataset–backbone settings without training a classifier head, exceeds LP on IN-100 ViT-B/16 by +0.3 pp, and beats raw CLIP cosine similarity by +5.2 to +24.4 top-1. SpLiCE-ZS and Z-CBM trail CLIP cosine similarity by 1.4–57.3 and 17.6–45.7 pp, and EZPC's near-identity projection stays within 0.4 pp of CLIP, so MOSAIC is the only cosine classifier that clearly improves on CLIP. MOSAIC also outperforms every LP-trained CBM in the LP-CBM column on all four datasets where they report numbers (+8.0 CUB, +6.4 CIFAR-100, +9.7 IN-1k, +1.2 Places365 at L/14-336), so the gain comes from the concept basis itself.

Table 2: **Retrieval R@1 (%)**, ViT-L/14-336. Training-free baselines (Z-CBM, SpLiCE-ZS) use a fixed concept vocabulary; EZPC and MOSAIC train per-dataset under the same inference protocol (cosine similarity, no classifier head). App. D reports MOSAIC's R@5 and MRR.

		Training-free baselines			Trained baseline	Ours
Dataset	Direction	CLIP	Z-CBM	SpLiCE-ZS	EZPC	MOSAIC
CUB-200	i→t	62.5	16.6	4.7	62.6	**87.9**
	t→i	71.5	18.5	5.0	4.0	**98.5**
IN-100	i→t	84.4	48.7	82.1	84.3	**90.1**
	t→i	93.0	32.0	97.0	52.0	**99.0**
IN-1k	i→t	74.0	29.7	71.6	73.7	**81.8**
	t→i	83.4	6.4	87.6	79.5	**94.1**

Retrieval result. Table 2 shows MOSAIC beats CLIP at R@1 on every combination of three datasets and two directions, by +5.7 to +27.0. The largest gains appear where CLIP's text–image alignment is weakest, with MOSAIC reaching 98.5 R@1 on CUB-200 t→i (CLIP 71.5) and 87.9 on CUB-200 i→t (CLIP 62.5). No competing CBM exceeds CLIP in *both* directions, each failing for a different reason: Z-CBM and SpLiCE-ZS collapse on fine-grained CUB-200 ($< 20\%$ R@1) because their concept bases are too generic, and EZPC's matching+KL objective aligns image features to class text in only one direction, matching CLIP on i→t but collapsing on t→i (4.0 on CUB, 52.0 on IN-100). MOSAIC encodes images and class texts in the same concept basis, so retrieval cosine

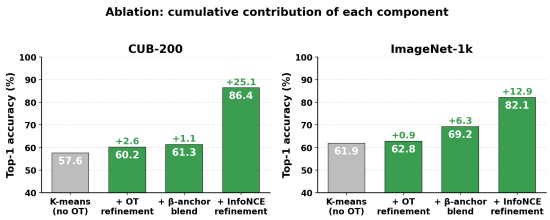

Figure 3: **Cumulative component ablation.** Each bar cumulates one component.

Table 4: **Codebook-init ablation, CUB-200, $K = 512$, ViT-L/14-336.** Test top-1 (%) before and after 10 epochs of Stage-2 InfoNCE, holding all other hyper-parameters fixed.

Init	raw	+Stage-2
random unit vectors	14.58	72.71
text-anchor only	63.03	70.71
K-means only	52.04	73.02
full Stage-1	61.06	**76.92**

is the same operation in both directions. Beyond class-level retrieval, the discrete concept basis enables compositional Boolean queries over signed concepts, with qualitative examples in App. F.

Table 3: **Per-method overhead relative to a frozen CLIP backbone.** † marks SpLiCE's FISTA-dominated latency. Latency is end-to-end classification, from raw image to class index, on a single GH200 GPU. Class-level retrieval shares the same encode-then-cosine path and has identical per-image cost. Storage is reported at $K = 1024$ (the default for CIFAR-100, IN-100 and Places365); MOSAIC's storage scales as $k_{\text{enc}} \cdot (\lceil \log_2 K \rceil + 32)$ bits, with the per-K table in App. J.

Method	External data	Trainable params	Storage/img (bit)	Latency (ms)
CLIP Radford et al. [2021]	—	0	24 576	15.81 ± 0.24
Z-CBM (ZS)	LLM concepts	0	24 576	15.93 ± 0.27
SpLiCE (ZS) Bhalla et al. [2024]	—	0	$\sim$2 000	$32.69 \pm 0.77^{\dagger}$
LF-CBM Oikarinen et al. [2023]	LLM queries	$\sim$10 M	$\sim$5 000	$\sim$15.9
LaBo Yang et al. [2023]	LLM queries	$\sim$10 M	$\sim$5 000	$\sim$15.9
EZPC Ozdemir et al. [2026]	—	$\sim$0.2 M	$\sim$10 000	15.96 ± 0.35
DN-CBM Rao et al. [2024]	600 GB corpus	$\sim$40 M+probe	$\sim$27 000	$\sim$15.9
MOSAIC (ours)	**—**	**$\leq$3 M**	**5,376**	$\mathbf{15.89 \pm 0.25}$

Efficiency. A concept layer that adds substantial latency or storage overhead can offset the interpretability gain it provides. Table 3 compares latency, storage, and training cost across the seven CBM baselines. End-to-end MOSAIC inference takes $\mathbf{15.89 \pm 0.25}$ ms, statistically indistinguishable from raw CLIP cosine (15.81 ± 0.24 ms) and $2\times$ faster than SpLiCE. Per-image storage is **5,376** bits at the default $K = 1024$; $k_{\text{enc}} = \max(16, K/8)$ scales storage with K (per-K breakdown in App. J). At the default K, MOSAIC is comparable to LF-CBM/LaBo ($\sim$5k bits) and well below EZPC ($\sim$10k) and DN-CBM ($\sim$27k). MOSAIC adds $\leq$3M trainable parameters and $\sim$60 min of per-dataset training.

5 Ablation

Component ablation. To isolate the marginal contribution of each pipeline stage, Figure 3 ablates them cumulatively on ViT-L/14-336. Stage **(0)** is pure K-means, **(1)** adds OT refinement, **(2)** adds the text-anchor β-blend, and **(3)** adds InfoNCE. The **β-blend** adds $+1.1$ on CUB and $+6.3$ on IN-1k as the largest stage-1 lever, and **InfoNCE** adds another $+25.1$ CUB and $+12.9$ IN-1k. InfoNCE adds the most because it directly optimizes the inference-time cosine, aligning each image's top-k histogram with its class-text histogram over the codebook fixed by Stage 1.

Concept initialisation. This ablation isolates Stage-1 by holding Stage-2 fixed at 10 epochs and varying only the initialisation (Table 4). All four variants train on the 80% seen-class split, so absolute numbers sit below the all-class Tab. 1 by design. The full Stage-1+2 pipeline leads by $+3.9$ pp over the next-best init (K-means only). Random unit-vector and K-means inits each recover most of the gap on their own, confirming InfoNCE is essential; pure text-anchor init under-performs both, since centroids collapse onto class-text directions and InfoNCE has too little room to refine. The OT + β-blend combination provides both geometric coverage and language alignment, and neither alone matches it. App. I sweeps codebook size K from 64 to 2048 and finds accuracy saturates near $K = 1024$, with $K = 64$ already recovering 71.3% on IN-1k at $40\times$ compression over dense CLIP.

Inference rule. Holding the trained codebook fixed and varying only the scoring function isolates the histogram-cosine inference rule. We compare R0 one-to-one CLIP cosine $\cos(\phi_I(x), \phi_T(c))$; R1 dense-softmax histogram cosine, $\cos(\mathrm{softmax}(XC^\top/\tau), \mathrm{softmax}(T_cC^\top/\tau))$; R2 top-$k$ sparse histogram cosine; and R3 MOSAIC (R2 + TF-IDF). Table 5 shows that R3 beats R0 by $+24.4$ pp on CUB-200, $+5.2$ on IN-100, and $+8.1$ on IN-1k. Sparsity and TF-IDF each change accuracy by ≤ 0.3 pp on top of the dense softmax, so the softmax step is the principal operation.

Table 5: **Inference-rule ablation.** Top-1 (%) with a fixed trained codebook, varying only the scoring rule. R3 is MOSAIC's default; bold marks the inference rule used in the main paper. On CUB-200 the no-IDF top-k rule (R2) is numerically slightly higher than R3, but we adopt R3 for consistency with the other datasets.

	R0 CLIP-ZS	R1 softmax	R2 top-k	R3 MOSAIC
CUB-200	61.94	86.28	86.42	**86.35**
IN-100	84.82	89.96	89.98	**90.04**
IN-1k	73.62	81.32	81.41	**81.69**

6 Interpretability Analysis

We probe the codebook from three angles, asking whether the post-hoc captions describe what each centroid encodes, whether the centroids recover hand-labelled visual attributes, and whether the surfaced concepts contribute to the classifier's predictions.

Table 6: **Cross-VLM fidelity, three independent judges.** Mean $\pm$ std of fidelity ratings on a 0–4 scale, averaged across LLaVA-1.6-Mistral-7B, InternVL3.5-30B-A3B, and Gemma-3-27b-it. MATCHED = concept name paired with its own six-class image grid; SHUFFLED = derangement (different concept's name). The matched–shuffled gap is judge-agnostic (per-judge breakdown in Tab. 20, App. N).

Dataset	matched	shuffled	gap
CUB-200	3.25 ± 0.04	1.20 ± 0.39	$+2.05\pm0.43$
IN-100	2.76 ± 0.35	0.62 ± 0.38	$+2.14\pm0.22$
IN-1k	2.66 ± 0.36	0.26 ± 0.24	$+2.39\pm0.21$

VLM-description fidelity. This test rules out captions that sound plausible without matching their centroid. We name each centroid with Qwen3.5-35B-A3B Qwen Team [2026] (Sec. 3.3, App. E) and audit every concept name on its six-class grid with three independent VLM judges, LLaVA-1.6-Mistral-7B Liu et al. [2023], InternVL3.5-30B-A3B Chen et al. [2024], and Gemma-3-27b-it Team et al. [2024], on a 0–4 fidelity scale. Judges rate *matched* (name paired with its own grid) versus *shuffled* (derangement, grid paired with another concept's name). Matched names score 2.66–3.25 on average across the three datasets, while the derangement collapses to 0.26–1.20 (Table 6); the matched–shuffled gap holds across every individual (judge, dataset) entry (Tab. 20, App. N). Figure 2 shows three example concepts with their top activating images. Concept non-redundancy and per-image diversity statistics are in App. L.

Concept–attribute alignment (CUB-200). Discovered concepts risk corresponding to arbitrary feature directions rather than human-recognizable visual primitives. CUB-200's hand-labelled binary attributes give a ground truth for measuring alignment. We count a concept and an attribute as aligned when the Spearman rank correlation ρ between the concept's activation and the binary attribute label on the test split reaches $|\rho| \geq 0.3$. At $K = 512$, 60.9% of the concepts active on the test set align with at least one attribute. Most of them thus recover human-named primitives without any attribute supervision, while the rest capture compositional or finer-grained structure that CUB's labelled vocabulary does not enumerate. An attribute-level retrieval test using these correlations is in App. O, and a label-free interpretability proxy over the full K codebook is in App. M.

Deletion faithfulness. A CBM's per-prediction explanation is only meaningful if removing the highlighted concepts breaks the prediction. Otherwise the displayed activations correlate with the

outcome without producing it. Following the deletion-based paradigm of ERASER DeYoung et al. [2020], Jacovi and Goldberg [2020], we zero out the top-M most-active concepts per test image and re-classify. A faithful explanation produces a sharp accuracy drop, an unfaithful one does not. At $M = 5$, deletion costs -84.7 top-1 pp on CUB-200 ($86.4\% \rightarrow 1.7\%$) and -76.4 pp on IN-1k ($81.7\% \rightarrow 5.3\%$), well below the corresponding CLIP zero-shot baselines (61.9% on CUB, 73.6% on IN-1k), showing residual CLIP knowledge cannot recover the prediction. The displayed top-M concepts are thus the operative basis of each prediction, giving users a per-image audit trail that traces directly to the classifier's decision (full curves in App. K). In contrast, removing the 64 most-used concepts from every image costs only 0.8 to 8.5 pp, so predictions draw on the whole codebook rather than on a few frequent concepts (App. U).

7 Conclusion and Limitations

MOSAIC discovers a sparse, named concept basis from the target dataset and classifies by cosine on histograms, coming within 2.5 pp of the supervised CLIP linear probe on $8/10$ dataset–backbone settings (exceeding it on one; Places365 is the notable exception) without a classifier head, beating every LP-trained CBM by $+1.2$ to $+9.7$ pp, and lifting CLIP retrieval R@1 by $+5.7$ to $+27.0$ pp across three datasets and both directions. The same discrete representation supports three capabilities prior CBMs offer in isolation, namely compositional Boolean retrieval (App. F), per-prediction concept decompositions whose names are audited by three independent VLM judges (Sec. 6, App. N), and concept-level intervention via deletion (App. K). Together they make the concept layer interactive, so every prediction can be verified, contested, or corrected rather than only inspected.

Limitations. **(i)** MOSAIC's codebook is dataset-specific, so the compute saved by skipping an external concept corpus is offset by per-dataset training. **(ii)** Concept names come from a frozen VLM, and our cross-VLM audit (App. N) is itself VLM-based, so individual centroids can still be miscaptioned and downstream claims that depend on caption accuracy inherit this dependence. **(iii)** Head-free cosine applies a single global IDF weighting, so concepts predictive for one class but uninformative for another are weighted identically, which leaves a residual gap on scene classification where the diagnostic feature subset varies sharply across classes (App. A). **(iv)** Stage-1 assumes an equal-mass prior over concepts, which may under-serve frequent concepts on strongly long-tailed data (App. S), and **(v)** each centroid is blended toward its nearest class-text anchor without a semantic-adequacy check (App. R).

Acknowledgments and Disclosure of Funding

The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR) McIntosh-Smith et al. [2024]. Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Business, Innovation, Science and Trade (DBIST) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), under the project Perfect Recollection for Clearer Insight (grant number EP/Y036077/1). The authors declare no competing interests.

References

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=7UyBKTFrtd`.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4443–4458, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4198–4205, 2020.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7(3):535–547, 2019.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/koh20a.html`.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

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

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

Simon McIntosh-Smith, Sadaf Alam, and Christopher Woods. Isambard-AI: a leadership-class supercomputer optimised specifically for artificial intelligence. In *Proceedings of the Cray User Group*, pages 44–54, 2024.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=FlCg47MNvBA`.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Onat Ozdemir, Anders Christensen, Stephan Alaniz, Zeynep Akata, and Emre Akbas. Explaining clip zero-shot predictions through concepts. *arXiv preprint arXiv:2603.28211*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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*, pages 8748–8763. PmLR, 2021.

Vikram V Ramaswamy, Sunnie SY Kim, Ruth Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10932–10941, 2023.

Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, pages 444–461. Springer, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Antonio De Santis, Schrasing Tong, Marco Brambilla, and Lalana Kagal. Learning concept bottleneck models from mechanistic explanations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gdEWoxhb70.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

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

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19187–19197, 2023.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022. URL https://openreview.net/forum?id=HAMeOIRD_g9.

Yuan Zang, Tian Yun, Hao Tan, Trung Bui, and Chen Sun. Pre-trained vision-language models learn discoverable visual concepts. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Vq0wMFBjo2.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. Concept embedding models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=HXCPA2GXf_.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022b.

A CLIP-Linear-Probe Protocol

CLIP-LP setup. The CLIP-LP column of Table 1 is computed with `sklearn.linear_model.LogisticRegression`, L-BFGS solver, inverse regularisation $C = 10$, on L_2-normalised CLIP image features (no augmentation, single-crop, the same 80-prompt class embedding ensemble is used downstream of the probe at evaluation). Probes are trained on each dataset's official train split and evaluated on the official test split; no validation-set tuning is applied to the probe regularisation, so these numbers are a conservative lower bound on the supervised LP ceiling rather than an optimised one. CLIP-LP is included in the main table as the rightmost reference column so the ΔLP gap is visible alongside the cosine-classifier baselines; MOSAIC recovers $\geq 97\%$ of LP on 8 of 10 dataset–backbone settings without training a classifier, exceeds LP on IN-100 ViT-B/16 by $+0.3$ pp, and the only settings with a substantial residual gap are Places365 (-6.2/-6.6 pp), where supervised classifier weights yield more accuracy gain than the codebook can recover from frozen features alone.

Why Places365 has the largest gap. Scene classification is class-conditionally heterogeneous: a *beach* class is recognised via warm-colour, horizon-line, and sand-texture activations, whereas *church indoor* is recognised via vaulted-ceiling, arch, and window-symmetry activations, and these per-class weightings are not implicit in cosine similarity on a shared concept basis. A trained LP head can learn arbitrary per-class concept reweightings and recovers the gap; MOSAIC's head-free cosine applies a single global IDF weighting, so concepts that are predictive for one class but uninformative for another are weighted identically. This is by design, since class-conditional reweighting reintroduces the LP-head failure mode we explicitly avoid (Sec. 1), but it leaves a residual gap on tasks like scene classification where the diagnostic feature subset varies sharply across classes. Tasks where most classes are distinguished by the same axes (CUB-200 fine-grained bird parts, ImageNet object categories) do not exhibit this gap.

B DN-CBM Comparison

DN-CBM Rao et al. [2024] is the closest data-driven analogue to MOSAIC. Instead of an LLM-supplied vocabulary, it pretrains a sparse autoencoder on the CC3M corpus and names the resulting neurons via CLIP-text matching, then trains a linear probe for classification. It is therefore the natural reference for isolating the contribution of data-driven concept discovery, modulo two design choices that differ from MOSAIC. It relies on a 600 GB external corpus (vs. MOSAIC, which uses no external corpus), and it requires a per-task linear probe at inference (vs. MOSAIC's head-free cosine similarity). We reproduce DN-CBM using the authors' official pretrained SAE weights, a GPU SGD linear probe for large-scale robustness, and their paper-native backbones (Table 7). MOSAIC at ViT-L/14-336 outperforms DN-CBM at both of its native backbones on every dataset except Places365 with RN50 (-3.4 pp).

Backbone and resolution. DN-CBM released SAE checkpoints only for CLIP ViT-L/14 at 224 px and RN50, and retraining its SAE requires the 600 GB CC3M corpus, so its entries cannot be run at our ViT-L/14-336 setting. To remove the mismatch we also run MOSAIC at DN-CBM's exact backbone and resolution (ViT-L/14, 224 px). There MOSAIC leads DN-CBM on every dataset with a ViT-L/14 entry, by $+7.0$ / $+7.1$ / $+2.5$ / $+0.9$ pp on CUB-200 / CIFAR-100 / IN-100 / Places365, while dropping from 336 to 224 px costs MOSAIC about 1 pp on CUB-200 ($86.4 \rightarrow 85.4$). MOSAIC on the weaker ViT-B/16 backbone (Tab. 1) also matches or exceeds DN-CBM at ViT-L/14-224 (CUB-200 78.5 vs. 78.4, CIFAR-100 80.5 vs. 79.8, Places365 46.4 vs. 45.7). All other columns of Tab. 1 are backbone-, resolution- and prompt-matched and re-run by us.

C Comparison with Classifier-Head-Trained CBMs

The main text (Section 4) compares only methods that classify by cosine similarity in some feature space (no learned classifier weights). Several widely cited CBM methods (LF-CBM Oikarinen et al. [2023], LaBo Yang et al. [2023], P-CBM Yuksekgonul et al. [2022]) train a linear probe on concept activations at inference, so they fall outside that comparison. We provide their reported numbers here for context.

Table 7: **Comparison with DN-CBM.** DN-CBM runs at its native backbones; ours is ViT-L/14-336 from the main table, and ViT-L/14 at 224 px, DN-CBM's exact backbone and resolution.

Dataset	DN-CBM (ViT-L/14, 224)	DN-CBM (RN50)	**Ours** (ViT-L/14, 224)	**Ours**
CUB-200	78.4	68.9	85.4	**86.4**
CIFAR-100	79.8	68.5	**86.9**	86.2
IN-100	87.2	82.1	89.7	**90.0**
IN-1k	N/A†	46.0	80.8	**81.7**
Places365	45.7	**50.3**	46.6	46.9

†DN-CBM ViT-L/14 LP failed to converge on IN-1k even with SGD+cosine.

Table 8: **Comparison with classifier-head-trained CBMs.** Rows with † are *cited verbatim from the respective papers*; other rows are our own runs. Dashes mean the paper did not evaluate that (dataset, method) pair; we do not fill them because backbones differ. The most directly comparable entry is **LaBo at ViT-L/14** (224×224 input resolution); MOSAIC uses ViT-L/14-336 (336×336 input), which provides an inherent advantage on fine-grained tasks, so the LaBo gap is an overestimate of the backbone-matched margin (the same caveat applies to DN-CBM in App. B). For datasets evaluated on Tab. 1 we substitute CIFAR-10 here for IN-100 to enable comparison with prior LaBo and P-CBM tables.

Method	Inference	Backbone	CUB-200	CIFAR-100	CIFAR-10	Places365	IN-1k
LF-CBM† Oikarinen et al. [2023]	LP	CLIP RN50/ViT-B/16	74.59	65.27	86.37	43.71	71.98
P-CBM† Yuksekgonul et al. [2022]	LP	CLIP RN50	—	56.00	84.50	—	—
P-CBM-h† Yuksekgonul et al. [2022]	LP + res.	CLIP RN50	—	69.90	87.60	—	—
LaBo† Yang et al. [2023]	LP	**CLIP ViT-L/14**	81.90	86.04	**97.75**	—	**83.97**
LaBo† *(avg 11 datasets)*	LP	CLIP ViT-L/14		85.72 (average)			
DN-CBM (our repro)	LP	CLIP ViT-L/14 (224)	78.43	79.76	—	45.70	N/A
CLIP-LP (our repro, reference)	LP	CLIP ViT-L/14-336	86.92	86.53	97.70	53.50	83.49
MOSAIC (ours)	**cosine**	**CLIP ViT-L/14-336**	**86.38**	**86.16**	97.69	46.93	81.69

LF-CBM's per-dataset backbone choice is mixed (RN50 for ImageNet, ViT-B/16 for others); the same weights table is reported from their GitHub README. P-CBM / P-CBM-h numbers are from the LaBo paper's Table 2 comparison (CLIP RN50). LaBo's ViT-L/14 row is its paper's Table 10 (full data), and the LaBo *average-of-11* row is their paper's Table 1 Full-shot mean. We do not re-run these methods ourselves; all cited entries are reproducible from the source papers.

Reading the table. The most directly comparable entry is **LaBo at ViT-L/14**, which classifies with a trained linear probe. MOSAIC stays competitive with classifier-free cosine similarity alone, ahead by $+4.48$ pp on CUB-200 and $+0.12$ pp on CIFAR-100, level on CIFAR-10 (-0.06 pp) and 2.28 pp behind on IN-1k, although our inference regime is strictly simpler. On CIFAR-10, MOSAIC **ties our own CLIP-LP upper bound to within 0.01 pp** (97.69 vs 97.70), a cosine-classifier CBM matching the supervised linear probe on the same backbone. At smaller backbones (RN50 / ViT-B/16) the LP-based CBMs land at 56–75% on CUB-200, CIFAR-100 and IN-1k; our ViT-L/14-336 numbers clearly dominate but the comparison is not backbone-matched. Beyond LaBo on IN-1k, the one other caveat is CLIP-LP itself, which is not a CBM but an upper-bound reference. Across the five datasets it ties us on CIFAR-10 (0.01 pp), exceeds us by ≤ 1.8 pp on CUB-200, CIFAR-100, and IN-1k, and exceeds us by 6.57 pp on Places365 where head-free cosine cannot express the class-conditional feature reweighting that scene classification rewards (a known limitation we discuss in Sec. 7), all while requiring a trained classifier head at inference.

Codebook size. MOSAIC uses fewer concepts than most LP-trained baselines and is freely configurable down to $K = 64$ (Table 9). LF-CBM and EZPC size K to the LLM-generated list per dataset, ranging from 208 on CUB-200 to $4\,523$ on IN-1k. LaBo's top-N-per-class scheme produces $5\,000$–$50\,000$ concepts. DN-CBM and SpLiCE use a single fixed dictionary across datasets ($6\,144$ SAE features at ViT-L/14, $8\,192$ at RN50, and SpLiCE's $10\,000$-word vocabulary); in our SpLiCE-ZS and Z-CBM runs the vocabulary is dataset-specific, with 312 / 594 / $3\,294$ concepts on CUB-200 / IN-100 / IN-1k. MOSAIC's K is strictly smaller than every baseline on IN-1k and Places365 and comparable to LF-CBM/EZPC on CUB-200 and CIFAR-100, while App. I shows $K = 64$ already recovers 71.3% on IN-1k at a regime no other CBM reaches.

Table 9: **Codebook size K (number of concepts) per method per dataset.** LF-CBM and EZPC use the LLM-generated list size; LaBo uses top-$N = 50$ concepts per class; DN-CBM uses a fixed SAE pretrained on CC3M (6 144 features at ViT-L/14, 8 192 at RN50); SpLiCE uses a fixed 10 000-concept dictionary, replaced in our SpLiCE-ZS runs by the dataset-specific vocabulary of Z-CBM. "–" means the method does not report numbers on that dataset. MOSAIC's K is strictly smaller than every baseline on IN-1k and Places365, comparable to LF-CBM/EZPC on CUB-200 and CIFAR-100, and freely tunable down to $K = 64$ (App. I).

Method	CUB-200	CIFAR-100	IN-100	IN-1k	Places365
LF-CBM Oikarinen et al. [2023]	208	824	—	4 523	2 207
LaBo Yang et al. [2023]	∼10 000	∼5 000	∼5 000	∼50 000	∼18 250
P-CBM Yuksekgonul et al. [2022]	—	∼250	—	—	—
DN-CBM Rao et al. [2024]	6 144	6 144	6 144	6 144	6 144
SpLiCE Bhalla et al. [2024]	10 000	10 000	10 000	10 000	10 000
EZPC Ozdemir et al. [2026]	208	824	—	4 523	2 207
MOSAIC (ours)	**512**	**1 024**	**1 024**	**4 096**	**1 024**

D Full Retrieval Table

Table 10 extends the main retrieval table (Table 2) with R@5 and Mean Reciprocal Rank (MRR) for every (dataset, direction) entry at the same protocol (cosine similarity on IDF-weighted concept histograms, no classifier head).

Table 10: **Full retrieval metrics for MOSAIC**, all three class-level retrieval datasets, ViT-L/14-336. i→t R@1 is reported under the retrieval protocol (cosine ranking on histograms with no temperature calibration), which differs in temperature handling from the classification protocol of Tab. 1; small numerical gaps between the two (e.g. CUB 87.85 R@1 vs. 86.4 top-1) reflect this and not a data split change.

Dataset	Direction	R@1	R@5	MRR
CUB-200	i→t	87.85	98.40	0.925
	t→i	98.50	100.00	0.992
IN-100	i→t	90.10	99.05	0.941
	t→i	99.00	100.00	0.995
IN-1k	i→t	81.80	97.10	0.886
	t→i	94.10	99.70	0.967

E VLM Concept Discovery

For each trained codebook centroid we ask Qwen3.5-35B-A3B Qwen Team [2026] to describe the visual feature shared across its top activating images. Naming has two steps: a first pass that captions every centroid once, described in the paragraphs below, and a selection step that samples contrastive candidates and picks each concept's name among them (App. E.1). The galleries in this section and the examples in App. F are labelled with first-pass captions; the selected names for every concept, with per-concept scores and provenance, accompany the code. Figures 4, 5, and 6 show three representative concepts per dataset alongside their captions; the selection criterion is described in the next subsection.

On CUB-200 (Fig. 4) the codebook captures fine-grained avian features such as bold black markings, wing posture, and feather texture, none of which corresponds to a single species.

On IN-100 (Fig. 5) the same procedure returns broader cross-category descriptors such as repeating body pattern, sinuous shape, and open-jaw posture, features that generalise across the much wider ImageNet class set.

On the full IN-1k codebook (Fig. 6), where $K = 4096$ centroids cover 1000 classes, captions span postures, facial expressions, and scene-context features.

Concept #202: Bold black markings



Concept #298: Extended wing configuration



Concept #489: Soft downy feather texture

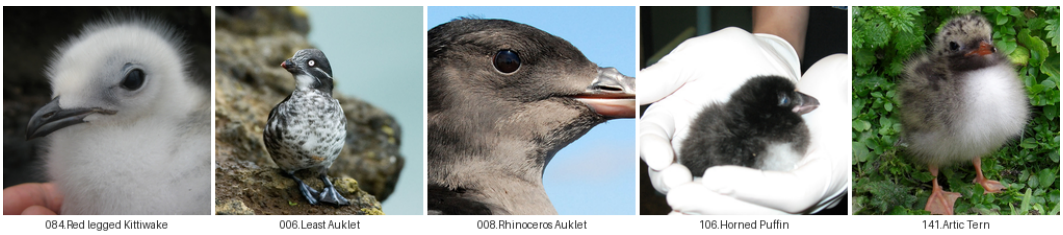

Figure 4: **Concept gallery on CUB-200.** Three concepts with their Qwen3.5 descriptions; each row shows top-activating images sampled one-per-class. Descriptions are produced with no label supervision and no class names appear in the captions.

Concept #96: High-contrast repeating body pattern



Concept #225: Sinuous curved elongated shape



Concept #902: open-jaw posture



Figure 5: **Concept gallery on IN-100.** Same layout as Fig. 4. Concepts span animals, postures, and texture patterns rather than single ImageNet classes.

Concept #216: Curled resting posture



Concept #258: Smiling facial expression



Concept #284: Green grassy ground texture



Figure 6: **Concept gallery on IN-1k.** Same layout as Fig. 4. At $K = 4096$ the codebook still yields concise feature-level descriptors that span many classes.

Selection criterion. Every codebook concept is VLM-captioned ($K = 512$ for CUB, 1024 for IN-100, 4096 for IN-1k). The gallery figures show the top-6 concepts by a composite score that rewards (i) genuine cross-class activation (`n_distinct_classes` ≥ 3 in the top-activating images), (ii) SHARED or DISCRIMINATIVE quality tier (excluding diffuse prompt-boilerplate concepts), and (iii) description-level diversity (a Jaccard token filter with ≤ 0.5 overlap on the VLM descriptions, so the gallery does not show the same feature twice when several centroids activate on near-duplicate visual content).

Prompting. The VLM receives a 2×3 grid of top-6 cross-class activating images plus a prompt that (i) declares the images are activation-rank-ordered (top row carries the strongest activations, so the model should weight them more heavily); (ii) tells the model the images come from *different* object categories so the shared property cannot be the subject; (iii) frames the task abstractly (imagine you are describing the visual element to a graphic designer who must reuse it on a completely different object) to push the model away from species-naming; (iv) constrains the answer to a SHORT *descriptive phrase* of ≤ 3 words built from adjective + attribute (*not* a noun naming the object) and forbids conjoining multiple attributes with and / with / commas, so each concept is named on a single diagnostic axis; and (v) requires the response to end with the literal token `Final answer:` followed by the phrase, so reasoning-mode chain-of-thought can be parsed cleanly. The full prompt body is:

> *You are naming a learned visual concept. The grid below shows 6 images that a vision model has clustered together because they share ONE underlying visual feature. Your job is to label that feature with a SHORT descriptive phrase (NOT a noun naming the object), the kind of primitive visual attribute that would belong in an interpretable feature vocabulary.*
>
> *The images are arranged in activation-rank order: the top row contains the three strongest-activating images. Weight the top row more heavily.*
>
> *The images come from DIFFERENT object categories, so the shared feature is NOT the subject or species. The label must be abstract enough to apply across categories: a property a graphic designer could re-use on a completely different object.*
>
> *Reply with a SHORT descriptive phrase (3 words or fewer) naming the visual feature itself, not what is in the image.*
>
> *Format rules (strictly enforced):*
>
> *(1) End your response with a single line in the exact form: `Final answer: <descriptive phrase>`, where the phrase is at most 3 words, no trailing punctuation.*
> *(2) Output a DESCRIPTIVE phrase (adjectives + attribute), NOT a noun naming the object. iridescent green is good; green plumage is acceptable; parrot is wrong.*
> *(3) The phrase MUST NOT begin with All images, Each image, The images, A photo of, An image of, Features of, Something that, Close-up of, or any other preamble.*
> *(4) Use ONLY abstract visual descriptors. NEVER name a species, animal, object class, or category (no snakes, dogs, birds, cars, people, reptiles). If the images happen to share a category, ignore that and name the underlying visual property.*
> *(5) ONE feature only. Do NOT conjoin multiple attributes with and / with / commas. Pick THE most diagnostic axis.*
> *(6) Choose the axis from: colour, texture, shape, pattern, material, lighting, posture, composition.*
>
> *Examples of clean descriptive labels: iridescent green; matte scaly; curved pointed; horizontally striped; wet glossy; metallic ridged; warm earthy; spiral coiled.*

If a completion overruns the 3-word cap, the captioner re-prompts with a tighter limit (down to 2 words) up to two times before accepting the long answer; this absorbs the long-tail of verbose completions without falling back to a generic descriptor.

Post-processor. Two regex stages in `clean_caption` (`ezpc/vlm_concept_discovery.py`) clean the raw VLM output before any caption is shown in a figure or sent to a judge. **Stage A (thinking-aloud strip).** Reasoning-mode VLMs occasionally leak chain-of-thought debris past the `Final answer:` parser ($\sim$80 / 4096 captions on IN-1k, $\sim$17 / 1024 on IN-100): arrow-style self-edits (`"X pattern" -> "X"`; we keep the post-arrow segment), reasoning preambles (*Let's…, Looking at…, Another thought:, From the list:, So…, Therefore…*), bullet markers, dangling quote marks around the actual answer, and trailing meta-commentary (*– that's listed as a…, is overly generic*); each is matched and stripped, and any caption longer than 10 words is truncated to its first sentence. **Stage B (category-noun strip).** The remaining $\sim$1–3% of completions that still lead with a category noun (Snakes with…, Birds in…) are caught by matching

`[adj{0,2}]` + `category-noun` + `connector` (where *connector* $\in$ {with, of, having, showing, featuring, displaying, being}) and stripping the prefix; bare leading categories with no connector are replaced with the abstract token Subjects. The category-noun list covers the dominant CUB-200, IN-100/1k surfaces (snakes, dogs, cats, birds, reptiles, mammals, fish, insects, etc.). All gallery captions in Figs. 4–6 pass through both stages.

Decoding. Generation is greedy (`do_sample=False`) with `max_new_tokens=1024` (large enough to fit the reasoning-mode prefix and the final answer line); on Qwen3.5 processors that support it we pass `enable_thinking=False` to the chat template, which suppresses the explicit reasoning block where supported but still emits the same final-answer-line trailer. The post-processor strips everything before the last `Final answer:` marker; if no marker is present (rare, $< 1\%$ of outputs across all three datasets), it falls back to the last non-empty line. A second pass strips any preamble that survived rule (2) (e.g. a leading All images feature…) so the returned label always begins with the attribute itself.

Cross-class grids. Candidate generation shows each concept a grid that spans six classes: it recomputes the concept's activations over the full training split and takes the top image from each of the six highest-activating classes. This rules out class identity as the shared property, which matters most on fine-grained data: on CUB-200 a concept's six top-activating images span only 1.8 classes on average.

E.1 Contrastive name selection

Captioning each concept independently tends to give neighbouring concepts the same locally correct phrase: on CUB-200 the 506 first-pass captions contain 305 distinct strings, and "bright yellow underparts" labels 13 concepts. Concepts that share a caption are still distinct centroids (mean cosine 0.77, against 0.59 for random pairs), so name selection generates contrastive candidates and chooses among them with a judge-free criterion, raising the number of distinct names to 437 (Tab. 11).

Candidates. Each concept is shown its six-class grid (above) with the first-pass prompt, extended by the rule *"The following labels are ALREADY USED by the most similar concepts in this codebook: [list]. Your label MUST be different from every one of them, and must name the property that separates THIS group from those. If the obvious label is already taken, choose a finer distinction on the same axis (a more specific shade, a more specific placement, or a more specific texture)."* The list holds the first-pass names of the four nearest concepts by centroid cosine, and the word limit is four. We sample five candidates per concept at temperature 0.9, discard candidates outside 2–6 words or containing a category noun, and apply a stricter filter for reasoning text that matches its structure (quotation, self-questions, references to grid positions, list items and clauses cut off mid-phrase) rather than a phrase list.

Scoring. For a name t and concept k, $\rho(t, k)$ is the Spearman correlation, over test images, between the CLIP similarity of t to the image cropped to its annotated object box (CUB-200 boxes; the ILSVRC validation boxes for ImageNet) and the activation of concept k on the full image. The score $s(t, k) = \rho(t, k) - \frac{1}{K-1} \sum_{j \neq k} \rho(t, j)$ rewards a name that predicts its own concept and penalises one that predicts others, which raw correlation would not do for generic names. Scoring the object crop rather than the whole image means a name that describes the habitat no longer tracks the concept. We score on the test split because ImageNet's object boxes exist only for its validation images; names are metadata and enter no reported accuracy.

Assignment. Names are assigned one-to-one by the Hungarian algorithm to maximise total score, treating names that differ only in case or word order as one, and a concept may only receive a candidate generated from its own grid. Without that restriction the assignment scores higher while drawing 92% of its names from other concepts' candidates, which optimises the correlation proxy rather than the naming. The first-pass name is kept unless the assigned candidate beats it by 0.005 and itself passes the candidate filters; a concept with no valid candidate keeps its first-pass name if that name is valid, and is otherwise left unnamed. Candidates that name a class are rejected: on CUB-200 the head noun of each species name ("Gull", "Warbler"), and on ImageNet, where class names are ordinary English, any class name outside a list of words used descriptively, or a name

of one of the five classes on which the concept itself fires most. One CUB-200 concept (#384) is named by hand with its first-pass caption, "hummingbird hovering in flight", because its best-scoring candidate describes an object absent from most of its images; it is the only name that contains a class word.

Effect. Table 11 compares the first-pass and selected names. Selection picks a sampled candidate for 61–63% of concepts, more than halves the share of concepts whose name is shared with another on every dataset, and cuts the share of names with negative s, which track the average other concept better than their own, from 9–11% to 1–5%. The same captioner suffices throughout: on CUB-200, a larger candidate generator (Qwen3.8-27B) brings no further gain.

Table 11: **First-pass versus selected concept names.** Distinct names are exact strings, ignoring case. s is the discriminative score of App. E.1; "pairs > 0.8" is the share of name pairs whose CLIP text embeddings exceed cosine 0.8.

	CUB-200		IN-100		IN-1k	
	first pass	selected	first pass	selected	first pass	selected
Named concepts	506	506	994	994	3975	3920
Distinct names	305	437	651	860	3098	3570
Concepts sharing a name (%)	56	20	48	20	33	14
Name pairs > 0.8 (%)	5.1	2.1	2.1	1.4	0.8	0.6
Mean score s	0.133	0.176	0.139	0.195	0.114	0.143
Names with negative s (%)	10.1	1.2	10.9	2.0	8.8	4.5
New name (%)	—	63	—	61	—	61

F Interpretable Retrieval Examples

Beyond the standard retrieval numbers in Sec. 4, the codebook supports two retrieval modes CLIP cannot natively express. Examples are selected automatically by `ezpc/demo_curator.py`, which ranks queries by the human-readability of their constituent concept labels (favouring attribute-style tokens over class-name templates and penalising duplicate labels), so the selection process is reproducible.

F.1 Multi-concept Boolean queries

A query is a signed set of concepts $\{+c_1, +c_2, -c_3, \ldots\}$; the score for an image is $\sum_{+c} \mathbf{h}[c] - \sum_{-c} \mathbf{h}[c]$, and we retrieve the top-k images by that score. Such compositional queries, especially with hard negation, are difficult to express with a single text prompt and dense CLIP cosine similarity; the discrete concept basis makes them mechanical.

To make the compositional structure visible, the figures below show each query *incrementally*: the first row uses only $+c_1$ (broadly mixed images sharing the first concept), the second adds $+c_2$ (more specific intersection), and the third subtracts $-c_3$ (suppresses a contrastive feature). Class names appear under each retrieved thumbnail so the narrowing effect is observable directly.

Figure 7 shows a CUB-200 query whose three steps progressively narrow the retrieval set, demonstrating that each Boolean operator over named concepts produces a visible change in the gallery.

A second CUB-200 query (Fig. 8) repeats the same incremental structure with a different concept triple, to confirm that the narrowing behaviour is not specific to a single query.

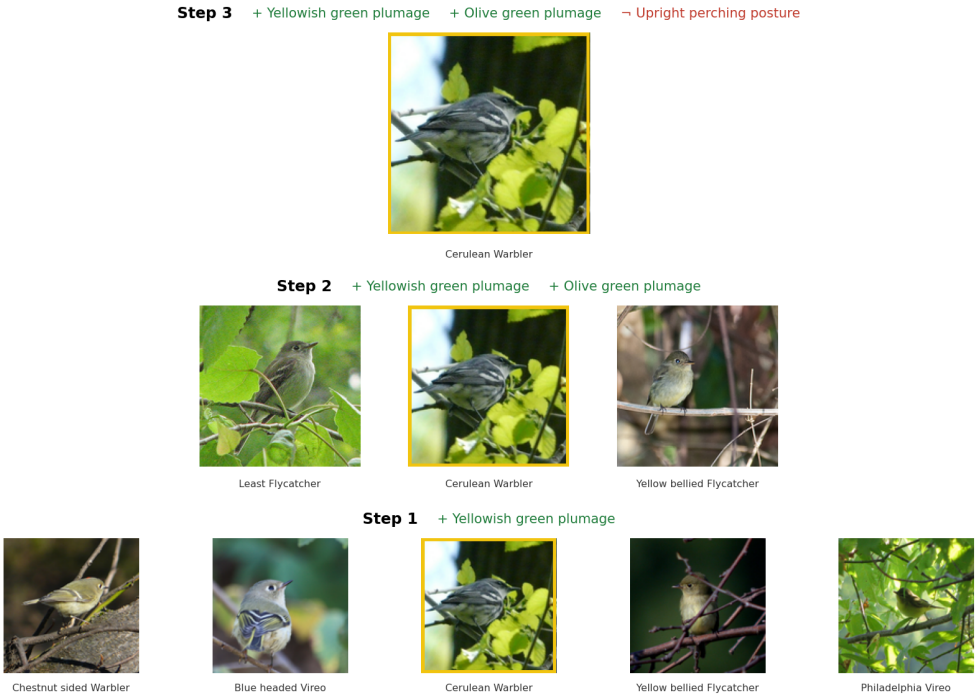

Figure 7: **Incremental composition on CUB-200.** Each row shows the top-8 retrievals after adding one more concept to the query. *Step 1* ($+A$ alone) returns a visually mixed set of images that share concept A. *Step 2* ($+A+B$) narrows to images activating both concepts. *Step 3* ($+A+B-C$) suppresses the negated concept and yields the most specific subset. Class names appear under each thumbnail.

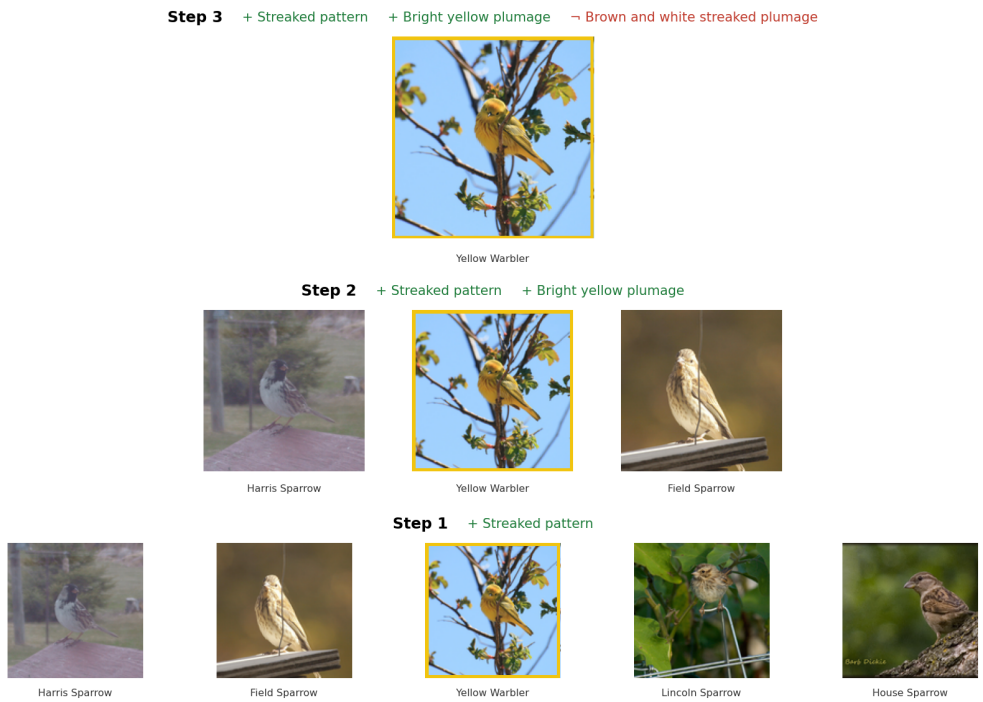

Figure 8: **Incremental composition, second CUB-200 query.** Same layout as Fig. 7: each step adds a concept and the retrieval narrows from a broadly similar set to a specific feature combination. Compositional Boolean queries are mechanical in MOSAIC's discrete concept basis but difficult to express with a single text prompt and dense CLIP cosine similarity.

F.2 Interpretable image-to-image similarity

For two images, we score their similarity by the weighted cosine similarity of their concept histograms; because every image decomposes into a sparse, weighted mixture of named concepts, the per-concept contributions are exposed directly. Each similarity-explanation figure therefore shows *both* images' individual concept-weight profiles (blue and orange bars) along with the per-concept shared weight ($\min$(blue, orange), hatched). The shared bars recover the classical *why are these two similar?* reading; the coloured bars expose that each image *separately* activates an interpretable subset of concepts, which is the essential CBM property.

Figure 9 shows the first CUB-200 example (two crested waterbirds) where the per-concept bar chart exposes a crest-related similarity that single-cosine CLIP cannot localise.

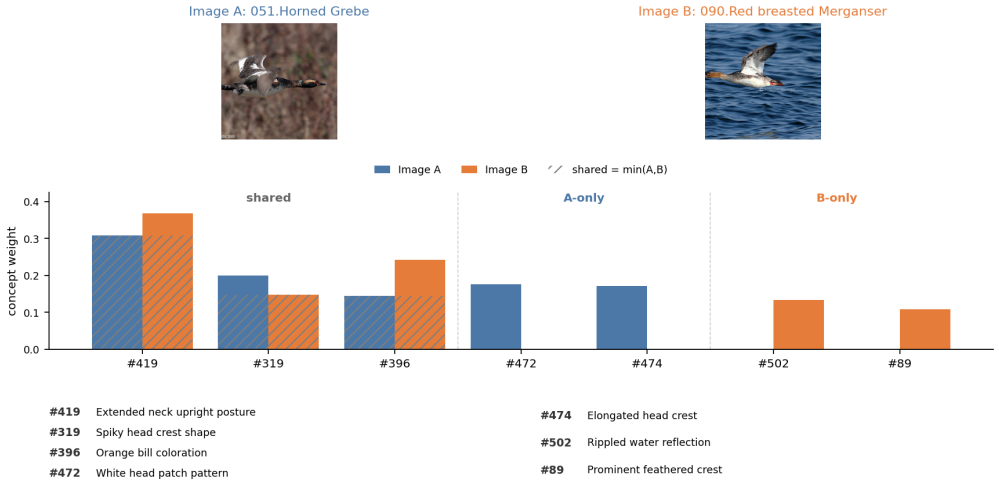

Figure 9: **Similarity explanation on CUB-200.** A Horned Grebe (Image A) compared to a Red-breasted Merganser (Image B), two crested waterbirds. Blue bars show Image A's per-concept weights; orange bars show Image B's; the hatched overlay is the shared component $\min(h^A, h^B)$ that drives the cross-class similarity score. The two birds share an extended neck-upright posture (#419), spiky head crest (#319), and orange bill (#396); they diverge on Image A's white head-patch (#472) and elongated head crest (#474) versus Image B's rippled water context (#502) and prominent feathered crest (#89). Several crest-related concepts coexist because the two species crest differently; the per-concept bars expose this rather than collapse it to a scalar.

A second CUB-200 pair (Fig. 10) replaces the waterbird example with two perching songbirds, showing that the mode-3 visualisation generalises across CUB families.

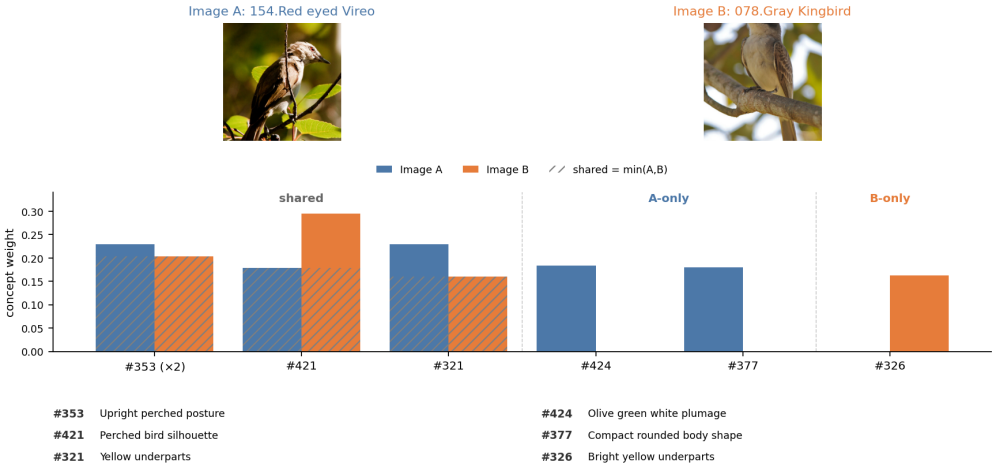

Figure 10: **Red-eyed Vireo vs. Gray Kingbird.** Both species share an upright perched posture (#353) and yellow underparts (#321); they diverge on Image A's olive green and white plumage (#424) versus Image B's compact rounded body shape (#377).

ImageNet examples. Figures 11 and 12 give analogous examples for IN-100.

The Boolean-query example in Fig. 11 demonstrates how compositional retrieval narrows from a broad pool of bumpy marine creatures down to a single radially-symmetric sea anemone in three steps, using only signed concept names rather than free-text prompts.

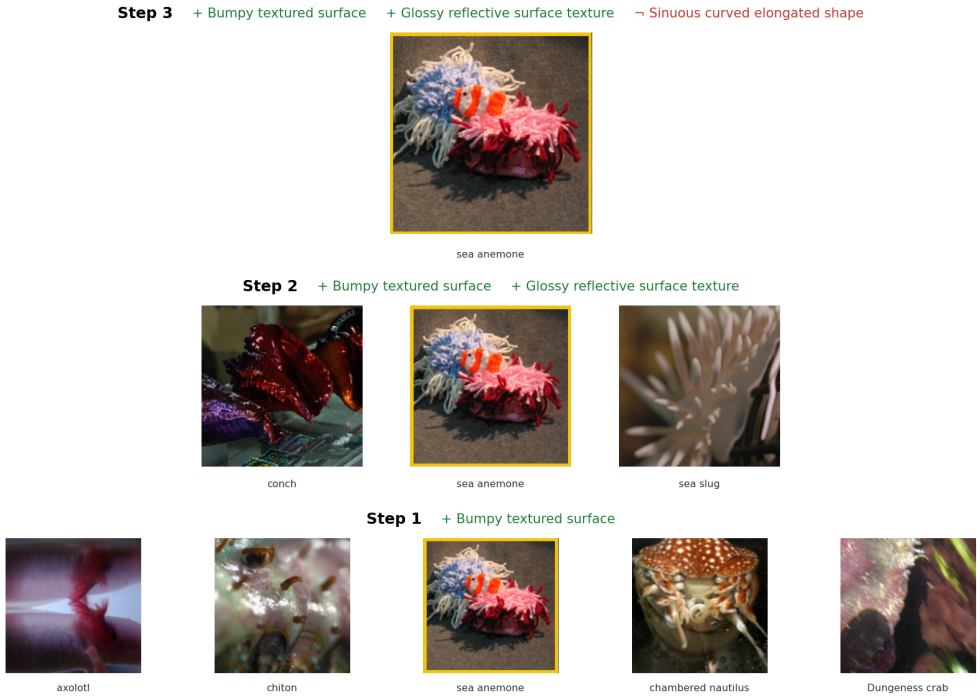

Figure 11: **Incremental composition on IN-100.** Same layout as Fig. 7. Query $\{+$*bumpy textured surface*, $+$*glossy reflective surface*, $-$*sinuous elongated shape*$\}$. Step 1 returns a broad set of bumpy marine creatures (axolotl, chiton, sea anemone, chambered nautilus, Dungeness crab); Step 2 keeps only the glossy ones (conch, sea anemone, sea slug); Step 3 excludes elongated shapes, leaving the radially-symmetric sea anemone. Class names appear under each thumbnail.

The mode-3 IN-100 example (Fig. 12) compares two morphologically distinct invertebrates and shows that the per-concept bar chart recovers shared shape and surface concepts even between an arthropod and an arachnid.

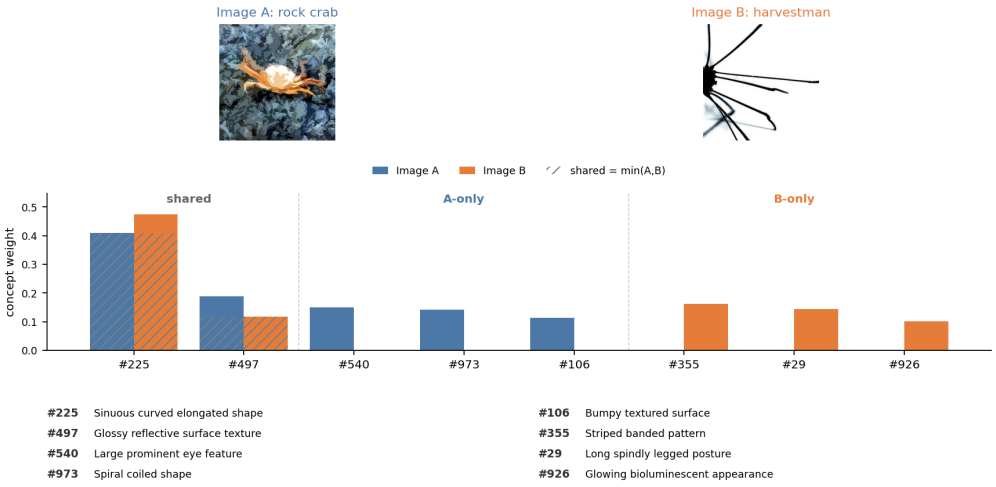

Figure 12: **Rock crab vs. harvestman on IN-100.** Both species share a sinuous curved shape (#225) and glossy reflective surface (#497); they diverge on Image A's large prominent eye and spiral coiled shape (#540, #973) versus Image B's striped banded pattern and long spindly legs (#355, #29).

G Training and Reproducibility Details

Hyperparameters (all datasets).

- Sinkhorn OT: $\varepsilon = 0.01$, cost $= (1 - \cos)/0.1$, EMA decay 0.9, 30 refinement epochs.
- Text-anchor blend $\beta = 0.3$.
- Encoding: top-$k_{\text{enc}} = \max(16, K/8)$ softmax at $\tau_{\text{hist}} = 0.05$.
- Stage-2 contrastive: Adam, lr 5×10^{-4}, 10 epochs, $\tau = 0.07$.
- Codebook sizes: $K = 512$ (CUB), 1024 (CIFAR-100, IN-100, Places365), 4096 (IN-1k).
- Prompts: OpenAI 80-template ensemble.

Hyperparameter-selection protocol. All inference-time settings — $k_{\text{enc}} = \max(16, K/8)$, $\tau_{\text{hist}} = 0.05$, and the IDF presence threshold 10^{-2} (Sec. 3.4) — were fixed once as implementation defaults and used unchanged across all five datasets, both CLIP backbones, and the SigLIP transfer; none was tuned on a validation split or per dataset. The sweeps in App. H were run afterwards purely as robustness checks; the shipped defaults are not the per-dataset test optima of those sweeps (e.g., on CUB-200 the no-IDF rule scores $+0.07$ pp and $\tau_{\text{hist}} = 0.02$ scores $+0.36$ pp above the defaults, and the IDF threshold 3×10^{-2} scores $+0.33$ pp on CUB-200 and 3×10^{-3} scores $+0.02$ pp on IN-1k, Tab. 14), which could not occur under post-hoc selection. The threshold 10^{-2} is on the order of the uniform top-k softmax mass $1/k_{\text{enc}}$, not a tuned constant.

Compute. All experiments on NVIDIA GH200 Grace Hopper GPUs. A single-dataset training run takes 15–30 min (CLIP features cached once per backbone). For IN-1k ($N \sim 1.3$M train images), the one-time CLIP feature encoding is the dominant cost ($\sim$1 h); the codebook training itself accounts for $\sim$15–20 min.

H Soundness Ablations

Stage 2 InfoNCE uses a dense softmax over all K centroids (needed for differentiability), while inference uses a top-k softmax with $k_{\text{enc}} = \max(16, K/8)$; and MOSAIC exposes two temperature

scalars ($\tau_{\text{hist}} = 0.05$ for histograms, $\tau = 0.07$ for InfoNCE). We sweep both to measure their effect on final accuracy.

Sparsity at inference. We fix the trained codebook and vary the inference-time sparsity k_{enc} over $\{16, 32, K/8, K/4, K/2, K\}$. The $k_{\text{enc}} = K$ row is the *train-regime* inference rule (same as Stage 2 training); the other rows are progressively sparser. Table 12 reports top-1 accuracy. Accuracy varies by at most 0.74 pp across the full range, and the default ($K/8$) is within 0.24 pp of the dense configuration ($k_{\text{enc}} = K$) on every dataset. The dense-softmax training regime transfers cleanly to the sparse-softmax inference rule.

Table 12: **Sparsity at inference.** Top-1 (%) on the test split as we vary inference-time k_{enc}, holding the trained codebook fixed. Default marked $\star$; K column is the dense-softmax rule used during Stage-2 training. Accuracy varies by ≤ 0.74 pp across the full range.

k_{enc}	16	32	$K/8^\star$	$K/4$	$K/2$	K
CUB-200	86.24	86.37	86.35	86.52	86.47	86.38
IN-100	89.64	89.54	90.04	89.98	89.96	89.86
IN-1k	81.51	81.69	81.69	81.65	81.58	81.55

Temperature. Table 13 sweeps the inference histogram temperature over $\{0.02, 0.05, 0.1, 0.2, 0.5\}$ and, on CUB-200, retrains Stage 2 with a 5×4 grid over $(\tau, \tau_{\text{hist}})$. Within ± 0.02 of the defaults accuracy is stable (≤ 1.6 pp on IN-100, ≤ 0.25 pp on CUB). Beyond $\tau_{\text{hist}} \geq 0.2$ the softmax flattens toward uniform and accuracy collapses (62.9% on IN-100 at $\tau = 0.5$). The 0.02–0.10 interval is the working range.

Table 13: **Temperature ablation.** Top-1 (%). Defaults marked $\star$. Top: inference-time τ_{hist} sweep at fixed $k_{\text{enc}} = K/8$. Bottom: Stage-2 retraining grid on CUB-200 over $(\tau, \tau_{\text{hist}})$, evaluated at the default inference config.

τ_{hist} (inference)	0.02	$0.05^\star$	0.10	0.20	0.50
CUB-200	86.71	86.35	75.94	54.56	42.10
IN-100	88.42	90.04	86.68	73.06	62.90
IN-1k	80.17	81.69	75.85	52.89	43.06

CUB-200: $\tau \setminus \tau_{\text{hist}}$	0.02	$0.05^\star$	0.10	0.20
0.03	75.53	85.64	83.60	76.61
0.05	76.13	86.14	84.66	72.85
$0.07^\star$	76.92	**86.35**	84.67	69.92
0.10	78.86	86.74	83.88	67.95
0.20	81.50	86.18	82.38	63.10

IDF presence threshold. The IDF vector counts a concept as present in a training image when its histogram weight exceeds 10^{-2} (Sec. 3.4). Freezing the trained codebooks, we sweep this threshold over seven values and add a no-IDF row (Tab. 14). All variants lie within 0.59 / 0.44 / 0.33 pp of each other on CUB-200 / IN-100 / IN-1k, and the default is within 0.33 pp of the best variant on every dataset (the best on IN-100). At $t = 0$ the document frequency is membership in the top-k_{enc} support, so the rule also runs without any threshold, within 0.44 pp of the default. The scoring-rule ablation of Sec. 5 already bounds the whole IDF term at ≤ 0.3 pp over the sparse histogram.

Table 14: **IDF presence-threshold sweep.** Top-1 (%), trained codebooks frozen, ViT-L/14-336. Default marked $\star$; best per row in bold.

t	0	10^{-4}	10^{-3}	$3{\cdot}10^{-3}$	$10^{-2\star}$	$3{\cdot}10^{-2}$	10^{-1}	no IDF
CUB-200	86.09	86.09	86.09	86.16	86.35	**86.68**	86.24	86.42
IN-100	89.60	89.60	89.60	89.84	**90.04**	89.70	89.82	89.98
IN-1k	81.54	81.54	81.62	**81.71**	81.69	81.59	81.38	81.41

IDF transfer / OOD sensitivity. The IDF vector is computed on training histograms (Sec. 3.4); if the test distribution shifts, a stale IDF may mis-weight concepts whose marginal frequency has changed. Our five-dataset evaluation is in-distribution by construction (standard classification splits), so this caveat does not affect the reported numbers. For deployments facing domain shift, the IDF vector is a single-pass statistic that can be re-estimated on unlabelled target-domain samples without retraining the codebook.

I Codebook Size Sweep

Sweeping $K \in \{64, 128, 256, 512, 1024, 2048\}$ on CUB-200 and IN-1k (Fig. 13), $K = 64$ already recovers 71.3% on IN-1k at $\sim$608 bits/image (**40**$\times$ compression vs. CLIP's 24 576 dense bits, just -2.3 below CLIP). Accuracy saturates around $K = 1024$ ($+9$ over CLIP at $\sim 4.5\times$ compression). Figure 13 plots the full sweep.

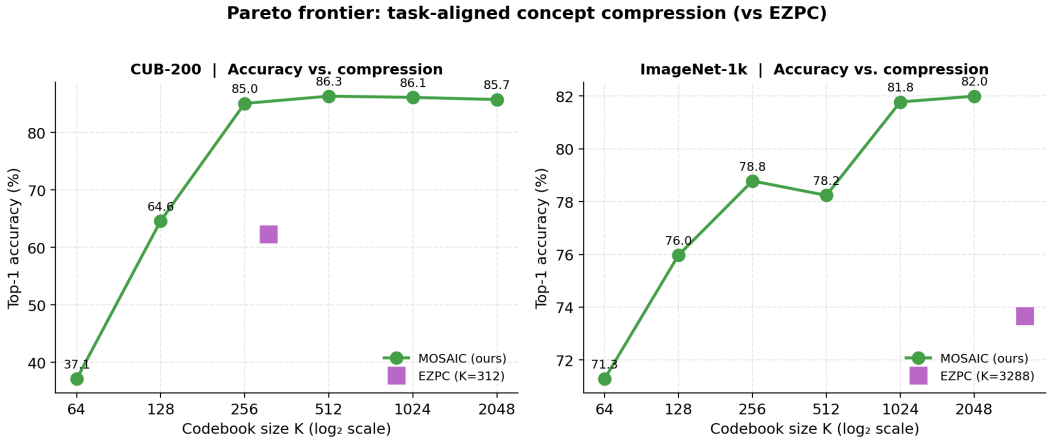

Figure 13: **Pareto frontier: accuracy vs. codebook size.** MOSAIC (green) spans $40\times$ to $2\times$ compression and reaches 86 top-1 by $K = 256$ on CUB and $\sim$82 on IN-1k by $K = 1024$. EZPC (purple square) achieves negligible compression with no accuracy gain over its starting basis. On CUB the phase transition at $K = n_{\text{classes}} = 200$ is visible.

J Efficiency and Overhead

We measure the overhead MOSAIC adds on top of a frozen CLIP backbone and compare against seven concept-bottleneck baselines along five axes: external pretraining data required, per-dataset trainable parameters, per-dataset training time, inference-time storage per image, and per-image inference latency (Table 15). Latency is reported in two parts: **embedding**, the cached-feature to method-specific concept representation cost, which is the per-method overhead in feature space, and **full pipeline**, the end-to-end cost from a raw image to a class index, including the CLIP ViT-L/14-336 forward pass. Numbers are mean$\pm$std over 100 runs for embedding and 50 images for full pipeline, on a single GH200 GPU, with all five measured methods interleaved per measurement step so that one-time CUDA cold-start cost is spread evenly across methods rather than falling on whichever method is timed first. SpLiCE incurs a 200-iteration FISTA solve in feature space that dominates its embedding time, while LF-CBM, LaBo, and DN-CBM use the standard linear-projection inference path shared with EZPC, and we do not separately benchmark them.

Storage–accuracy Pareto at smaller K. MOSAIC compresses further by choosing smaller K. The codebook-size sweep in App. I shows that at $K = 64$ each image occupies only $\sim$608 bits, a $40\times$ compression over dense CLIP, while still recovering 71.3% top-1 on IN-1k, 2 points below raw CLIP cosine similarity. No other CBM baseline reaches this regime, because all alternatives either have no compression mechanism, as with DN-CBM and EZPC, or collapse below CLIP before compression takes effect, as with Z-CBM and SpLiCE.

Table 15: **Per-method overhead relative to a frozen CLIP backbone.** Bold entries highlight the MOSAIC row and outliers († marks SpLiCE's FISTA-dominated latency, the only non-trivial slowdown). Latency entries marked ∼(est.) share EZPC's linear-projection inference path; others are precise mean±std. ‡ MOSAIC's storage scales with K via $k_{\text{enc}} \cdot (\lceil \log_2 K \rceil + 32)$ bits per image: 608 bit at $K = 64$, 5 376 bit at $K = 1024$ (default for CIFAR-100, IN-100, Places365), 22 528 bit at $K = 4096$ (default for IN-1k); CUB-200 at $K = 512$ uses 2 624 bit.

Method	External data	Trainable params	Training time (IN-1k)	Storage/img	Embed. (ms)	Full pipeline (ms)
CLIP Radford et al. [2021]	—	0	0	24 576 bit	0.073 ± 0.006	15.81 ± 0.24
Z-CBM (ZS)	LLM concepts	0	0	24 576 bit	0.169 ± 0.010	15.93 ± 0.27
SpLiCE (ZS) Bhalla et al. [2024]	—	0	0	∼2 000 bit	$18.09 \pm 0.61^{\dagger}$	$32.69 \pm 0.77^{\dagger}$
LF-CBM Oikarinen et al. [2023]	LLM queries	∼10 M	∼30 min	∼5 000 bit	∼0.15 (est.)	∼15.9 (est.)
LaBo Yang et al. [2023]	LLM queries	∼10 M	∼60 min	∼5 000 bit	∼0.15 (est.)	∼15.9 (est.)
EZPC Ozdemir et al. [2026]	—	∼0.2 M	∼45 min	∼10 000 bit	0.123 ± 0.008	15.96 ± 0.35
DN-CBM Rao et al. [2024]	CC3M (600 GB)	∼40 M + probe	∼1–3 days + ∼1 h	∼27 000 bit	∼0.15 (est.)	∼15.9 (est.)
MOSAIC (ours)	**—**	**≤3 M**	**∼60 min**	**608–22 528 bit‡**	$\mathbf{0.203 \pm 0.013}$	$\mathbf{15.89 \pm 0.25}$

K Deletion Faithfulness

A concept decomposition is informative only if the classifier uses those concepts to make its prediction, rather than emitting them as post-hoc labels that happen to correlate with the outcome. Faithfulness Jacovi and Goldberg [2020] distinguishes the two cases. If the concepts contribute to the prediction, removing them should break it. If they are decorative, the prediction should survive intact. For each test image we zero out the top-M most-active concepts in its histogram and re-classify, see Fig. 14. Deleting the top-5 concepts costs -84.7 top-1 pp on CUB-200 and -76.4 pp on IN-1k (reported in Sec. 6); the full curves on the codebooks of Tab. 1 for CUB-200, IN-100 and IN-1k drop monotonically (values in Table 16), with the IDF weighting and L_2 normalisation applied before deletion as at inference.

Table 16: **Deletion faithfulness on the codebooks of Tab. 1.** Top-1 (%) after zeroing the M most active concepts of each test image, ViT-L/14-336.

M	0	1	3	5	10	20
CUB-200	86.35	53.90	4.83	1.69	0.29	0.09
IN-100	90.04	80.42	65.72	49.04	9.54	1.74
IN-1k	81.69	59.20	15.97	5.33	1.60	0.56

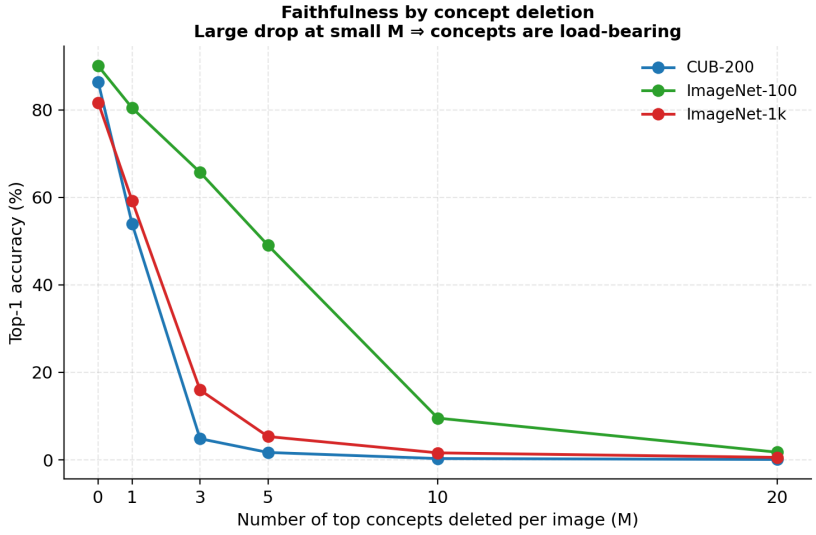

Figure 14: **Faithfulness by concept deletion.** Accuracy drops sharply when the top-M concepts per image are zeroed out (codebooks of Tab. 1; values in Table 16).

L Concept Non-redundancy

A codebook with K centroids is only useful if the centroids capture different features, otherwise it is a redundant recoding of CLIP geometry. We measure this two ways on all three datasets at the default codebooks $K = 512, 1024, 4096$. First, the geometric pairwise cosine between centroid vectors in CLIP feature space. Second, the semantic pairwise cosine between the concept names re-encoded by CLIP's text tower. Centroids are spread in CLIP feature space, with median pairwise cosines of 0.59 on CUB, 0.37 on IN-100 and 0.06 on IN-1k; the fine-grained CUB codebook is the most compact, since all of its concepts describe birds. The names are distinct as well: median name cosine is 0.56, 0.59 and 0.56, only 2.1%, 1.4% and 0.6% of name pairs exceed cosine 0.80, and 86.4%, 86.5% and 91.1% of names are distinct strings, against 60.3%, 65.5% and 77.9% for the first-pass names (App. E.1).

Per-image diversity. A complementary question is how many distinct concepts each prediction recruits. Table 17 reports per-image concept-activation statistics over each full test split. The median top-1 concept weight is only 0.08–0.11 (no single concept dominates), and the effective rank $(1/\sum_k h(x)_k^2)$ of each image's concept distribution is 40–197, so each prediction recruits many distinct concepts rather than acting as a single class detector.

Table 17: **Per-image concept-activation statistics** over each test split (5 794 / 5 000 / 50 000 images). Median top-1 concept weight is 0.08–0.11 (no single concept dominates); effective rank $(1/\sum_k h(x)_k^2)$ and $N_{50\%}$ (the rank at which cumulative concept weight reaches 50%) both grow with codebook size K.

Dataset	K	top-1 (median)	eff. rank (median)	$N_{50\%}$ (median)
CUB-200	512	0.11	40	11
IN-100	1024	0.09	58	12
IN-1k	4096	0.08	197	44

M Interpretability Rate over the Full Codebook

This appendix reports a label-free interpretability proxy applied to every centroid in the codebook, complementing the qualitative galleries with a quantitative tier-distribution at the full K. For every centroid we use a quick heuristic on the class-activation distribution: CONCENTRATED if the top class activation exceeds twice the second *and* the peak mean activation exceeds an absolute gate of 0.05 (filtering near-inactive centroids from the concentrated tier), CROSS-CLASS if ≥ 3 classes have activation $\geq 0.8\times$ the peak (i.e. a shared feature), else AMBIGUOUS. Results (Table 18): 50.0% / 58.8% / 58.1% of centroids pass the heuristic on CUB / IN-100 / IN-1k respectively, where the default codebooks are $K = 512, 1024, 4096$. The tail (AMBIGUOUS) is real and we report it rather than hiding it, but it measures class-activation *concentration*, not visual coherence: a concept whose activations spread across many classes fails the heuristic by construction even when it is a perfectly legible shared attribute. CUB's higher ambiguous-rate (50%) is expected: fine-grained species-specific features (e.g. subtle wing-pattern variations) are harder to surface with a heuristic that relies only on class-activation concentration, and they often distribute across many neighbouring species without any one dominating. The CUB galleries in App. F confirm that many of these ambiguous concepts are in fact visually coherent cross-class attributes; the proxy under-counts.

Table 18: **Interpretability rate over the full codebook.** Per-centroid tier distribution under the label-free proxy. Interp. = concentrated + cross-class.

	Concentrated	Cross-class	Ambiguous	Interp. rate
CUB-200, $K = 512$	43.2%	6.8%	50.0%	$\mathbf{50.0}\%$
IN-100, $K = 1024$	57.4%	1.4%	41.2%	$\mathbf{58.8}\%$
IN-1k, $K = 4096$	52.7%	5.4%	41.9%	$\mathbf{58.1}\%$

Decision mass and caption fidelity by tier. Each tier's share of test-time decision mass matches its share of the codebook (Tab. 19), and each tier's captions pass a judge-free fidelity check. On

every tier and dataset, the CLIP cosine between a centroid and the text embedding of its first-pass caption exceeds the cosine to a deranged caption. The tiers therefore measure how class-specific a concept is, not how well it can be described.

Table 19: **Decision mass and caption fidelity by tier.** Share of concepts, share of test-time histogram mass, and the judge-free fidelity margin: mean cosine between centroid and own first-pass caption minus that with a deranged caption.

	CUB-200			IN-100			IN-1k		
Tier	concepts	mass	margin	concepts	mass	margin	concepts	mass	margin
Concentrated	43.2	42.6	+0.061	57.4	56.3	+0.069	52.7	57.2	+0.093
Cross-class	6.8	5.9	+0.054	1.4	1.2	+0.062	5.4	3.2	+0.071
Ambiguous	50.0	51.6	+0.041	41.2	42.4	+0.056	41.8	39.6	+0.077

N Cross-VLM Fidelity: Methodology

This appendix expands the methodology behind Table 6 (reported in Sec. 6).

To avoid using a single VLM for both description and scoring, we generate captions with Qwen3.5-35B-A3B (as in Sec. 6) and rate every captioned concept with *three independent VLM judges*, LLaVA-1.6-Mistral-7B Liu et al. [2023], InternVL3.5-30B-A3B Chen et al. [2024], and Gemma-3-27b-it Team et al. [2024]. Each judge scores the (name, image-grid) pair on a 0–4 scale answering *does this description accurately describe the shared visual feature across these images?* Rating is run on every named concept in each dataset (506, 994 and 3 920 for CUB, IN-100, IN-1k), each name shown with its six-class grid (App. E.1), with greedy decoding and an identical prompt for all judges. No rating failed to parse.

Negative control. A high matched score is only credible if the same judge can also reject wrong description-image pairings. We therefore re-run each judge on a derangement, where each concept's image grid is paired with a different concept's caption. If the judge were lenient (blanket-positive), shuffled pairs would still score high. Table 20 reports the full per-(judge, dataset) breakdown that the main-body Table 6 aggregates into per-dataset means: across the nine (judge, dataset) cells, shuffled $\% \geq 3$ stays at 1.9–25.1% and the matched–shuffled gap stays at 50–70 pp on every cell. The gap is judge-agnostic, three independent VLM judges converge on the same calibration signal, ruling out artefacts from any single rater.

Selected versus first-pass names. To measure what name selection adds, we also rate the first-pass captions on the same six-class grids (Tab. 20, right). The selected names widen the matched–shuffled gap in all nine (judge, dataset) cells, by 1–11 pp: on ImageNet mainly through higher matched agreement (+5 to +10 pp), on CUB-200 through fewer deranged names being accepted.

Table 20: **Cross-VLM fidelity, full per-(judge, dataset) breakdown.** Each judge rates every concept name against its six-class image grid on a 0–4 scale. MATCHED = name paired with its own grid; SHUFFLED = derangement (different concept's name). Gap is the matched-vs-shuffled $\% \geq 3$ difference. This is the source for the per-dataset means in Tab. 6. Right: the first-pass captions rated on the same grids.

		Selected names				First-pass captions	
Judge	Dataset	mean (matched)	$\% \geq 3$ (matched)	$\% \geq 3$ (shuffled)	gap	$\% \geq 3$ (matched)	gap
LLaVA-1.6-Mistral-7B	CUB-200	3.27	**81.8%**	25.1%	+56.7	82.8%	+53.6
	IN-100	2.37	**57.9%**	8.0%	+49.9	52.4%	+43.8
	IN-1k	2.26	**54.0%**	2.0%	+52.1	47.1%	+44.9
InternVL3.5-30B-A3B	CUB-200	3.27	**78.1%**	12.8%	+65.2	78.7%	+64.0
	IN-100	2.84	**59.4%**	5.2%	+54.1	51.6%	+45.9
	IN-1k	2.73	**58.2%**	1.9%	+56.4	48.3%	+46.3
Gemma-3-27b-it	CUB-200	3.21	**90.5%**	24.9%	+65.6	88.9%	+60.7
	IN-100	3.06	**77.9%**	13.0%	+64.9	68.2%	+54.1
	IN-1k	2.98	**75.8%**	5.8%	+70.0	65.9%	+60.1

O CUB-200 Attribute Retrieval

Class-level retrieval on classification datasets saturates (every strong backbone scores $> 95\%$ R@1) and largely reduces to the classification task. A more informative capability test uses *non-class* queries. CUB-200 annotates every image with 312 binary attributes (e.g. `has_bill_shape::hooked`, `has_underparts_color::white`); we phrase each as a prompt (a bird that has bill shape hooked) and rank test images by three rules:

- **CLIP prompt**: $\cos(\phi_I(x), \phi_T(\text{prompt}))$.
- **MOSAIC nearest**: for each attribute, find the codebook centroid closest in CLIP-text space and rank images by their activation on it.
- **MOSAIC soft**: score $= \sum_k h(x)_k \cdot \cos(c_k, \phi_T(\text{prompt}))$.

Table 21 reports MAP and R@K averaged over the 311 attributes with ≥ 5 positive images. General scores are relatively low because positives are sparse (≤ 30 per attribute out of ~ 5800 test images) and the labels are visually subtle and noisy; the relative deltas are what matters. MOSAIC-soft leads by $+2.4$ MAP over CLIP's prompt baseline without any attribute-level supervision. MOSAIC-nearest, which relies on a single centroid, trails CLIP by ~ 1.7 MAP. Attribute presence is additive over several codebook centroids (a hooked bill activates multiple bill-shape-adjacent concepts), and the soft rule aggregates this evidence while the nearest rule discards it.

Table 21: **CUB-200 attribute retrieval.** Per-attribute retrieval over 311 attributes with ≥ 5 positive images (ViT-L/14-336). Higher is better.

	MAP	R@50	R@100	R@200
CLIP prompt	0.181	0.045	0.081	0.134
MOSAIC nearest	0.164	0.038	0.068	0.112
MOSAIC soft	**0.205**	**0.046**	**0.088**	**0.149**

P Additional Baseline: Soft-Prompt Adaptation (CoOp)

CBM baselines are not the only way to adapt frozen CLIP on a target dataset. The canonical non-CBM alternative is CoOp Zhou et al. [2022b], which learns a short sequence of soft-prompt tokens prepended to each class name and supervises them with cross-entropy on CLIP's image-text cosine similarity; later prompt-tuning extensions such as CoCoOp Zhou et al. [2022a] and Visual Prompt Tuning Jia et al. [2022] share the same opacity property. CoOp reaches strong accuracy but produces an opaque learned prompt; its predictions are not decomposable into named concepts.

We run CoOp with the standard unified-context configuration ($M = 16$, SGD lr $= 0.002$, cosine schedule, 200 epochs at batch 32 on CUB-200 and 30 epochs at batch 256 on the larger datasets) on the same five datasets and two backbones (Table 22). MOSAIC beats CoOp on every pair, by $+0.3$ to $+2.9$ pp. CoOp's compute scales linearly in the number of classes because it recomputes all class-text embeddings at every step; IN-1k $\times$ ViT-L/14-336 was outside our wall-clock budget. The accuracy gaps are small, but MOSAIC offers a decomposable concept basis and a constant-per-step inference cost that CoOp does not.

Q Non-CLIP Backbone: SigLIP-B/16

We replace the CLIP backbone with SigLIP-B/16 (`ViT-B-16-SigLIP`, WebLI pretraining) to check that MOSAIC is not tied to CLIP's feature geometry. SigLIP uses a sigmoid pair-wise loss (vs CLIP's softmax contrastive) and a different pretraining corpus (WebLI vs WIT-400M).

On SigLIP-B/16 (Table 23), MOSAIC reaches 95.2% of the SigLIP linear-probe ceiling on CUB-200 (79.4 vs 83.4) and 99.8% on IN-100 (88.1 vs 88.3), consistent with the CLIP results in the main paper. EZPC, the only CBM baseline that runs unchanged on SigLIP, tracks raw SigLIP cosine similarity (47.5 / 81.8) and offers no gain over the backbone.

Table 22: **CoOp comparison.** Top-1 (%). CoOp is run with $M = 16$ context tokens, batch $= 256$, 30 epochs (CUB rows use the paper's canonical 200-epoch configuration; convergence plateaued at $\sim$140 epochs). IN-1k $\times$ ViT-L/14-336 was infeasible in our wall-clock budget ($\geq$25 h at any batch size because CoOp recomputes 1000 class-text embeddings every step); we omit that entry. † = partial training on IN-1k $\times$ ViT-B/16: epoch 15 of 30 at wall-clock cap (CoOp's top-1 trajectory had flattened at $\sim$73.5: $73.05 \to 72.55 \to 72.92 \to 73.42 \to 73.47 \to 73.68$ at epochs $1, 3, 6, 9, 12, 15$; the remaining 15 epochs would add at most $\sim$ 0.5 pp by extrapolation). MOSAIC beats CoOp by $+0.3$–$+2.9$ pp on every reported pair, while CoOp produces an opaque learned prompt and no decomposable concept basis.

Backbone	Method	CUB-200	CIFAR-100	IN-100	IN-1k	Places365
ViT-B/16	CoOp	77.60	79.06	85.34	73.68†	45.02
	MOSAIC	**78.51**	**80.50**	**86.70**	**76.54**	**46.37**
ViT-L/14-336	CoOp	84.92	84.85	89.74	—	46.49
	MOSAIC	**86.38**	**86.16**	**90.04**	**81.69**	**46.93**

Table 23: **SigLIP-B/16 classification.** Top-1 (%). MOSAIC closes 95–100% of the LP ceiling on both datasets under a non-CLIP backbone.

Method (backbone: SigLIP-B/16)	CUB-200	IN-100
Zero-shot cosine (raw backbone)	47.39	81.66
Z-CBM	4.68	50.78
SpLiCE-ZS	0.52	39.34
EZPC	47.46	81.76
Z-CBM + linear probe	72.80	87.86
MOSAIC (ours)	**79.41**	**88.10**
Linear probe on raw backbone (ceiling)	83.43	88.30
MOSAIC % of LP ceiling	95.2%	99.8%

R Text-Anchor Blend β Sensitivity

The text-anchor blend weight β in Eq. (2) is fixed at $\beta = 0.3$ across all main-paper experiments. We verify this choice with a multi-seed sweep over the full dial $\beta \in \{0.0, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0\}$ on the full Tab. 1 grid (five datasets, ViT-L/14-336, $n = 3$ seeds: $\{42, 0, 1\}$; mean $\pm$ std reported). The endpoints matter for interpreting the anchor's role: $\beta = 0$ removes the anchor entirely and costs at most 0.69 pp (CUB-200) and $\leq$0.1 pp elsewhere — accuracy does not come from pulling centroids toward class-text directions — while $\beta = 1.0$ (pure anchor target) degrades the cross-dataset mean by 0.46 pp, most sharply on IN-100 (-0.87 pp), consistent with the text anchor overriding the image-driven OT signal; the corresponding collapse failure mode is strongest under pure text-anchor *initialization* (Sec. 5 init ablation). Between the endpoints the curve has a broad plateau over $\beta \in [0.2, 0.5]$ (cross-dataset means span 0.11 pp), and $\beta = 0.3$ sits in its middle, within 0.12 pp of the per-dataset best on every dataset.

Table 24: **Multi-seed β-sensitivity at the Tab. 1 classification protocol.** Top-1 accuracy (%), $n = 3$ seeds, ViT-L/14-336. Per-row maximum in bold; cross-dataset mean ($\bar{\cdot}$) in the bottom row.

Dataset	$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$ (used)	$\beta = 0.5$	$\beta = 0.7$	$\beta = 1.0$
CUB-200	85.66±0.12	85.70±0.11	85.97±0.08	86.35±0.06	**86.47±0.11**	86.46±0.16	86.34±0.12
CIFAR-100	86.10±0.03	86.17±0.06	**86.28±0.10**	86.19±0.06	86.20±0.02	85.90±0.09	85.78±0.11
IN-100	90.22±0.20	90.09±0.13	90.27±0.12	90.30±0.24	**90.31±0.32**	**90.31±0.42**	89.43±0.18
IN-1k	81.70±0.03	81.56±0.07	**81.80±0.12**	81.79±0.12	81.75±0.06	81.76±0.12	81.37±0.12
Places365	**47.16±0.14**	47.03±0.10	47.00±0.19	47.14±0.05	47.11±0.13	46.85±0.05	46.54±0.23
$\bar{\cdot}$	78.17	78.11	78.26	78.35	**78.37**	78.26	77.89

Findings. Differences between adjacent β values are small and within seed std on most cells. $\beta = 0.3$ lies within 0.12 pp of the per-dataset best on all five datasets (CUB-200 0.12, CIFAR-100 0.09, IN-100 0.01, IN-1k 0.01, Places365 0.02), and its cross-dataset mean (78.35) is within 0.02 pp

of the best ($\beta = 0.5$), so a single value serves every dataset. Cross-dataset means span only 0.24 pp across $\beta \in \{0.1, 0.2, 0.3\}$ ($78.11 \rightarrow 78.26 \rightarrow 78.35$), so Tab. 1 is not sensitive to small variations in β. The single-seed values in Tab. 1 (seed 42) lie within 0.3 pp of the multi-seed mean on every dataset (largest gap IN-100, 90.04 vs. 90.30), so the published numbers are seed-robust.

S Relaxed (Unbalanced) OT Marginal

We test the uniform row marginal a of Eq. (1) by replacing the hard row constraint with a KL penalty of strength τ (unbalanced Sinkhorn: each row update is scaled by $\tau/(\tau + \varepsilon)$, verified against POT, with the column marginal kept exact). We sweep $\tau \in \{0.05, 0.5, 5\}$ against the balanced default, three seeds each, on CUB-200 ($K = 512$) and IN-100 ($K = 1024$). Besides top-1 we record the centroids that receive no usage, the Stage-1-only accuracy, and how class-like the class-seeded slots become: the median over seeded slots of $\max_c \cos(c_k, T_c)$ (Tab. 25).

Mild relaxation ($\tau = 5$) is free. Strong relaxation leaves exactly the class-seeded centroids, 200 on CUB-200 and 100 on IN-100, at their class-text initialisation (max-cos 1.00) with zero transport mass, and costs 3.2 pp end-to-end and 14.5 pp at Stage 1 on CUB-200. A text-free control isolates the cause: with k-means initialisation, relaxing to $\tau = 0.05$ strands no centroid and costs nothing at Stage 1 (CUB-200 59.4% balanced, 60.6% relaxed). The constraint's load-bearing role is therefore to pull the class-seeded slots into the image manifold, so that class-identity directions cannot persist in the codebook (Sec. 3.1). It does not force uniform usage: the Stage-1 usage Gini under the balanced prior is 0.30 on CUB-200 and 0.12 on IN-100, and the trained Stage-1+2 codebooks reach 0.24 / 0.33 / 0.45 on CUB-200 / IN-100 / IN-1k (App. U).

Table 25: **Uniform versus KL-relaxed OT marginal.** Top-1 (%), mean ± std over 3 seeds; Stage-1 top-1 and usage at seed 42. Max-cos: median over class-seeded slots of $\max_c \cos(c_k, T_c)$. Unused: centroids with zero usage.

	CUB-200 ($K = 512$)				IN-100 ($K = 1024$)			
τ	top-1	Stage-1	max-cos	unused	top-1	Stage-1	max-cos	unused
balanced	86.35 ± 0.06	61.4	0.70	0	90.30 ± 0.24	85.1	0.64	0
5	86.31 ± 0.22	54.5	0.75	0	90.39 ± 0.14	84.9	0.65	0
0.5	83.15 ± 0.46	46.7	1.00	200	89.77 ± 0.37	82.6	1.00	100
0.05	83.27 ± 0.57	46.9	1.00	200	90.05 ± 0.03	83.1	1.00	100

k-means initialisation (text-free control), CUB-200, Stage-1: balanced 59.4, $\tau = 0.05$ 60.6; unused 0 in both

T Text-Anchor Adequacy Audit

A nearest class text always exists computationally, so its adequacy as an anchor has to be checked separately. For every centroid of the codebooks of Tab. 1 we compute $s_k = \max_c \cos(c_k, T_c)$, its cosine to the nearest class text (Tab. 26). No centroid exceeds 0.8 on any dataset, while class texts have a median cosine of 0.78–0.81 to their nearest other class text, so no centroid has become a copy of a class anchor. The IN-100 and IN-1k centroids stay in the image manifold, at nearly the same cosine to class text as raw CLIP image embeddings (median 0.27–0.31, the modality gap), while the fine-grained CUB-200 centroids sit closer to their class texts (median 0.62). Anchors act as coarse regional grounding rather than per-centroid targets: 913 anchors serve the $4\,096$ IN-1k centroids, and the anchor is re-selected in every batch from the current centroid, so a centroid that drifts toward a texture or composition switches anchors rather than being pulled back. The per-step pull toward the anchor is $\alpha_0 \beta = 0.03$, against 0.07 toward the OT image target. On IN-1k the cross-class tier, the case where a class anchor is least appropriate, sits furthest from all anchors (median 0.26).

U Concept Utilization

Deletion faithfulness (App. K) shows that the highlighted concepts matter; this appendix asks how widely usage is spread, and whether predictions rest on a small recurring subset. The analysis is inference-only on the seed-42 codebooks of Tab. 1, with concepts ranked once by their total

Table 26: **Cosine from each centroid to its nearest class text**, codebooks of Tab. 1, ViT-L/14-336. Baselines: median cosine from each class text to its nearest other class text, and from raw image embeddings to their nearest class text.

	CUB-200	IN-100	IN-1k
Median s_k (10th–90th pct.)	0.62 (0.59–0.71)	0.31 (0.28–0.34)	0.30 (0.25–0.34)
Centroids with $s_k > 0.8$	0%	0%	0%
Class text to nearest class text	0.81	0.78	0.81
Image to nearest class text	0.31	0.29	0.27
Distinct anchors used / classes	200/200	93/100	913/1000
Median s_k: concentrated / cross-class / ambiguous	0.62/0.61/0.62	0.32/0.30/0.31	0.31/0.26/0.29

activation mass over the training split (Tab. 27). Almost every concept carries usage, and half of it is spread over hundreds of concepts. Keeping only the 64 most-used concepts loses 34–68 pp, yet deleting the same 64 costs only 0.8–8.5 pp, so none of them is individually load-bearing. Table 28 gives the full retention and deletion curves.

Table 27: **Concept utilization**, codebooks of Tab. 1, ViT-L/14-336.

	CUB-200 ($K = 512$)	IN-100 ($K = 1024$)	IN-1k ($K = 4096$)
Concepts with non-zero training df	512	1024	3754
Concepts holding 50% of training usage	173	294	947
Usage Gini (training)	0.24	0.33	0.45
Distinct concepts in any test top-5	501	923	3516
Median per-image effective rank (test)	40	58	197
Test top-1, keeping only the 64 most-used	29.3	56.3	13.6
Test top-1, deleting the 64 most-used	77.8	83.8	80.9
Test top-1, full codebook	86.3	90.0	81.7

Table 28: **Retention and deletion curves.** Test top-1 (%) keeping only the N most-used concepts (top), and after deleting the N most-used concepts (bottom).

Keep N	16	32	64	128	256	512	1024	2048	4096
CUB-200	9.9	17.9	29.3	47.2	73.6	86.3	—	—	—
IN-100	30.1	41.2	56.3	70.6	82.0	89.0	90.0	—	—
IN-1k	2.6	5.8	13.6	23.7	37.1	51.3	67.3	79.8	81.7
Delete N	0	1	2	4	8	16	32	64	
CUB-200	86.3	86.3	86.2	86.0	85.2	83.7	82.0	77.8	
IN-100	90.0	89.4	89.3	88.9	87.6	86.8	85.1	83.8	
IN-1k	81.7	81.7	81.7	81.7	81.7	81.5	81.4	80.9	

V Interpretability Comparison with DN-CBM and SpLiCE

We compare MOSAIC, DN-CBM and SpLiCE as interpretability methods on CUB-200 under identical metric code, each with its own classifier: MOSAIC's head-free cosine, DN-CBM's official SAE at ViT-L/14-224 with a linear probe on its activations, and SpLiCE's FISTA decomposition over the 312-attribute vocabulary with concept-space cosine (Tab. 29). Because the methods differ tenfold in active concepts per image, deletion removes a fraction of each image's own active concepts rather than a fixed top-M, measured against each method's own accuracy. Alignment uses the Spearman correlation of Sec. 6 against CUB's human-annotated attributes, averaged over concepts that fire at least once on the test split, since 5 370 of DN-CBM's 6 144 units never do. MOSAIC is the most faithful (it keeps 1.0% of its accuracy after its top 10% of active concepts are removed, DN-CBM 15.7%), uses almost all of its dictionary, and aligns best with the human attributes on both views, including the per-attribute view, which favours larger dictionaries.

Table 29: **Interpretability head-to-head on CUB-200.** *SpLiCE activates a median of four concepts per image, so every fraction removes at least one, and its zero-shot base accuracy is 4.6%, so its retention ratios are uninformative.

	MOSAIC	DN-CBM	SpLiCE
Dictionary size / median active per image	$512/64$	$6144/607$	$312/4$
Dictionary ever active on the test split	$\mathbf{99.0}\%$	12.6%	43.6%
Top-1 with own classifier	86.3	82.4	4.6
Accuracy retained, top 2% of active deleted	$\mathbf{19.2}\%$	43.2%	$58.5\%^*$
Accuracy retained, top 10% of active deleted	$\mathbf{1.0}\%$	15.7%	$58.5\%^*$
Mean best $\lvert\rho\rvert$, active concepts	$\mathbf{0.342}$	0.179	0.147
Active concepts with $\lvert\rho\rvert \geq 0.3$	$\mathbf{60.9}\%$	7.1%	8.8%
Mean best $\lvert\rho\rvert$ per attribute	$\mathbf{0.227}$	0.191	0.189
Attributes with $\lvert\rho\rvert \geq 0.3$	$\mathbf{23.7}\%$	13.8%	16.0%

W Cross-Dataset Transfer

MOSAIC learns its vocabulary from the target data, so we test how a codebook transfers to a domain it was not trained on. We freeze a codebook trained on IN-100 or IN-1k, keep its source IDF, add the target's classes by encoding their names, and report top-1 over the target dataset's classes (Tab. 30). Objects transfer to objects: the IN-1k codebook reaches 62.8% on CIFAR-100, up from 25.7% with the IN-100 codebook. Scenes (26.5%) and fine-grained birds (20.6%) rely on discriminative features, such as wing bars and bill shapes, that an object-trained codebook does not contain, which is why MOSAIC learns its vocabulary from the target data. Transfer leaves the source codebook intact: it still scores 81.1% on its own 1000 classes. What matters is whether the source vocabulary contains the target's discriminative features, not how large the source is. Learning the codebook once over a broad image–caption corpus is the natural route to transfer; Stages 1–2 permit it, since class names only anchor centroids and stratify training batches.

Table 30: **Frozen-codebook transfer.** Top-1 (%) on the target dataset's classes, ViT-L/14-336. CLIP zero-shot from Tab. 1 for reference.

Source codebook	CIFAR-100	Places365	CUB-200
IN-100 ($K = 1024$)	25.7	5.3	21.0
IN-1k ($K = 4096$)	62.8	26.5	20.6
CLIP zero-shot	77.8	38.0	61.9

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Three quantitative claims appear in the abstract and Sec. 1. MOSAIC (i) comes within 2.5 pp of CLIP-LP on $8/10$ dataset–backbone settings, exceeding it on one, with Places365 stated as the exception (Tab. 1, App. A), (ii) outperforms every CBM baseline in our study (Tabs. 1, 2), and (iii) compresses to as little as 2.5% of dense CLIP (App. I).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Sec. 7 discusses five limitations. The codebook is dataset-specific (per-dataset Stage-1+2 training), concept names depend on a frozen VLM and are audited by VLM judges (App. N), head-free cosine with a single global IDF leaves a residual gap on scene classification where the diagnostic feature subset varies sharply across classes (Places365, -6 pp, App. A), Stage-1 assumes an equal-mass concept prior, and nearest class-text anchors carry no semantic-adequacy check.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: No new theoretical results. The OT/Sinkhorn formulation (Sec. 3.1) reuses Cuturi Cuturi [2013].

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Sec. 3 specifies the full pipeline. App. G lists every hyperparameter, App. H the sparsity and temperature sensitivities, and App. E the full VLM prompt with its post-processor, the name refinement and the selection rule.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [Yes]

   Justification: Code, training scripts, pre-trained per-dataset codebooks and concept names are available at `https://github.com/Sannaraek/MOSAIC`. All datasets used (CUB-200, CIFAR-100, ImageNet-100/1k, Places365) are public benchmarks.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: App. G lists Sinkhorn ε, EMA decay, β, k_{enc}, both temperatures, the Adam learning rate, Stage-2 epochs, and per-dataset K. Sensitivities reported in App. H.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Latency in Tab. 3 is mean±std over 50–100 runs. A 3-seed verification (seeds $\{42, 0, 1\}$) on the headline protocol reproduces every Tab. 1 cell to within 0.3 pp (App. R), so the single-seed numbers in Tab. 1 are seed-robust. Classification and retrieval numbers in the body are reported at seed 42 for exact reproducibility, and codebook-size robustness is reported in App. I.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Single NVIDIA GH200 GPU. Per-dataset training 15–30 min once CLIP features are cached. ImageNet-1k feature encoding is the dominant one-time cost (~ 1 h). Full per-method overhead in App. J.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: Public benchmarks and publicly released models only. Upstream licences respected.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: MOSAIC is an interpretability tool. It releases no generative model, no scraped data, and no high-risk artefact. **Positive**: per-prediction concept decomposition (Sec. 3.4) supports audit and intervention in deployments where opaque end-to-end scores are insufficient. **Negative**: post-hoc captions are not verified labels and could mislead users into over-trusting plausible-sounding names that do not faithfully describe the underlying centroid (*named* $\neq$ *verified*). We mitigate via the cross-VLM fidelity audit (App. N) and deletion-faithfulness analysis (App. K). High-stakes deployments should gate on a held-out cross-VLM audit plus domain-expert review of the concept gallery rather than the captions alone.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We release an algorithm and per-dataset codebooks only. No generative model, no new dataset, no high-misuse artefact.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets (CUB-200 Wah et al. [2011], CIFAR-100 Krizhevsky et al. [2009], ImageNet-100/1k Russakovsky et al. [2015], Places365 Zhou et al. [2017]) and models (CLIP Radford et al. [2021], SigLIP Zhai et al. [2023], Qwen3.5 Qwen Team [2026], LLaVA-1.6 Liu et al. [2023], InternVL3.5 Chen et al. [2024], Gemma-3 Team et al. [2024]) are cited at first use under their public research licences.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The public codebase ships with README, inline docstrings, trained per-dataset codebooks, and the concept names with per-concept scores and provenance, alongside the training and labelling scripts.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

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

Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Qwen3.5-35B-A3B Qwen Team [2026] generates the first-pass captions and the candidate names from which the concept names are selected (Sec. 3.3). LLaVA-1.6-Mistral-7B Liu et al. [2023], InternVL3.5-30B-A3B Chen et al. [2024], and Gemma-3-27b-it Team et al. [2024] score those labels as independent judges (App. N).

    Guidelines:

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