ANYBOTTLE: A RECIPE TO ONLY KEEP THE CONCEPTS YOU REALLY NEED

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept bottleneck models (CBMs) make predictions inspectable and intervenable by routing them through human-interpretable concepts, but originally required concept annotations. Annotation-free variants remove this requirement, but typically use large concept vocabularies, static at both training and inference, producing bottlenecks larger than any task or prediction needs and harder to inspect. We propose ANYBOTTLE, a single recipe for building compact, task-specific CBMs. ANYBOTTLE assumes only a frozen backbone and an unsupervised concept pool, such as a sparse autoencoder. A black-box teacher trained on the same backbone then guides selection: each round adds the concept that best explains the bottleneck's current failures, with candidates restricted to regions of teacher/student disagreement. Trained with nested dropout over this selection order, the final bottleneck predicts accurately from any concept prefix, so inference spends fewer concepts on inputs it is confident about early and more on hard ones. Since no stage is modality-specific, a new domain and task requires swapping only the backbone and concept pool. Across six vision and two text datasets and two teacher paradigms, ANYBOTTLE yields bottlenecks with fewer concepts and higher concept consistency than annotation-free baselines, while staying close to the black-box reference. Overall, ANYBOTTLE shows that going annotation-free need not mean going large: a small, discovered vocabulary can be as expressive as a much larger, fixed one.

1 INTRODUCTION

Concept bottleneck models (CBMs) provide interpretable decompositions of decisions across domains: a feature extractor maps inputs to human-interpretable concepts, combined via a simple (often linear) predictor. While originally requiring annotated concepts (Koh et al., 2020), recent approaches mainly use CLIP's language alignment, either via CLIP-based supervision (Oikarinen et al., 2023; Yang et al., 2023) or unsupervised concept discovery on a CLIP feature extractor (Rao et al., 2024; Bhalla et al., 2024; Wittenmayer et al., 2026), scaling in a task-agnostic manner without concept-level annotations.

Despite their utility, large, overcomplete concept vocabularies of thousands of candidates hurt interpretability on specialized tasks, where far fewer concepts often suffice. Globally, having many influential concepts causes duplication and feature splitting, complicating debugging and intervention; locally, individual predictions may use more concepts than necessary (Fig. 1, left). Reliance on a CLIP-like backbone (Oikarinen et al., 2023; Bhalla et al., 2024; Shang et al., 2024; Tan et al., 2024) and web-scale concept extractors further limits applicability, as these rarely cover a target domain's fine-grained or specialized concepts.

To address this, we propose ANYBOTTLE, a general recipe for compact, task-specific CBMs that keep only the concepts each output and sample needs. We start from an opaque teacher predictor and an initial, large, task-agnostic concept pool built with a sparse autoencoder (Bricken et al., 2023). Keeping feature extractor and teacher frozen, ANYBOTTLE selects concepts iteratively: each round, we identify the samples with the highest teacher error and add concepts covering parts of these inputs that the teacher uses, but the current concepts do not, yielding a compact concept set that stays task-relevant. The classifier is trained via nested dropout, enabling prediction from any number of concepts and letting inference spend fewer concepts on easy inputs and more on hard ones (Fig. 1, right). This small concept space also enables automatic, agentic naming (Shaham et al., 2024), testing concept meaning via targeted interventions rather than similarity to a fixed CLIP vocabulary (Bhalla

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

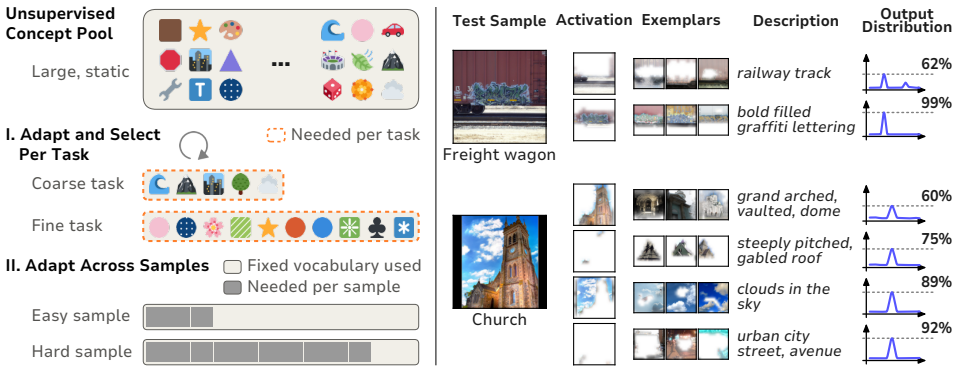


Figure 1: **Only as many concepts as needed.** *Left:* existing annotation-free concept bottlenecks draw from a large, static concept pool, using far more concepts than a given task needs, and the same fixed vocabulary for every sample regardless of difficulty. ANYBOTTLE instead selects a concept set sized to the task, and lets inference use only as many of those concepts as each sample actually requires. *Right:* two test samples illustrating this sample-level adaptivity.

et al., 2024). Finally, since selection assumes only a frozen backbone and an unsupervised concept pool, the recipe transfers across domains and modalities by swapping just these two components. Across six vision and two text datasets, ANYBOTTLE matches or exceeds teacher accuracy using an order of magnitude fewer, and individually more coherent, concepts than annotation-free baselines, resulting in a bottleneck only as large and detailed, as the task and sample demand.

In summary, our contributions are threefold: (i) a teacher-guided concept selection procedure that reduces disagreement with an opaque teacher to build a compact, task-specific bottleneck; (ii) a nested-dropout predictor that exploits this ordering to predict accurately from any concept prefix, spending fewer concepts on confident inputs; and (iii) ANYBOTTLE itself, a recipe requiring no concept annotation or language-aligned backbone, which we show transfers across domains and modalities on vision and text.

2 RELATED WORK

Concept Bottleneck Models (CBMs). CBMs (Koh et al., 2020; Stammer et al., 2021; Knab et al., 2026) mark an important moment in the growing interest in concept-based models (Yeh et al., 2021; Fel et al., 2023; Poeta et al., 2023), promising interpretable predictions and a structured interface for interaction. While the original framework relied on fully supervised concept annotations, subsequent research has relaxed this via unsupervised concepts (Sawada and Nakamura, 2022), leveraging pretrained vision-language models like CLIP for concept extraction (Bhalla et al., 2024; Yang et al., 2023; Oikarinen et al., 2023; Panousis et al., 2024; Yamaguchi et al., 2025), or employing fully unsupervised concept discovery (Stammer et al., 2024; Rao et al., 2024; Schrodi et al., 2025; Schut et al., 2025). Most approaches produce a fixed concept layer, set once and used as-is. ANYBOTTLE instead builds its bottleneck dynamically, keeping only the concepts a teacher's behavior calls for.

Compact and Sparse Concept Selection. A separate line targets compactness directly: how few concepts suffice to solve a task. Chattopadhyay et al. (2023b) cast Orthogonal Matching Pursuit as Information Pursuit (CLIP-IP-OMP), greedily selecting CLIP text-embedding atoms by residual correlation, later extended to a hierarchical dictionary (Nguyen et al., 2026). Res-CBM (Shang et al., 2024) completes a CLIP-text bank with optimizable vectors where existing concepts under-explain a class. OpenCBM (Tan et al., 2024) instead reconstructs an already-trained classifier post hoc via greedy nearest-concept-to-residual steps. Sparse-CBM (Semenov et al., 2024) and UCBM (Schrodi et al., 2025) pursue compactness via Gumbel-softmax and input-dependent gating instead. In text, TBM (Ludan et al., 2023) iteratively prompts an LLM for concepts from misclassified examples, while CT-CBM (Bhan et al., 2025) scores unsupervised candidates by activation importance until a residual branch plateaus. Most select from a language-aligned (CLIP-text or LLM-generated) bank; ANYBOTTLE instead selects greedily by residual over an unsupervised SAE dictionary, filtering candidates by teacher-student attribution disagreement.

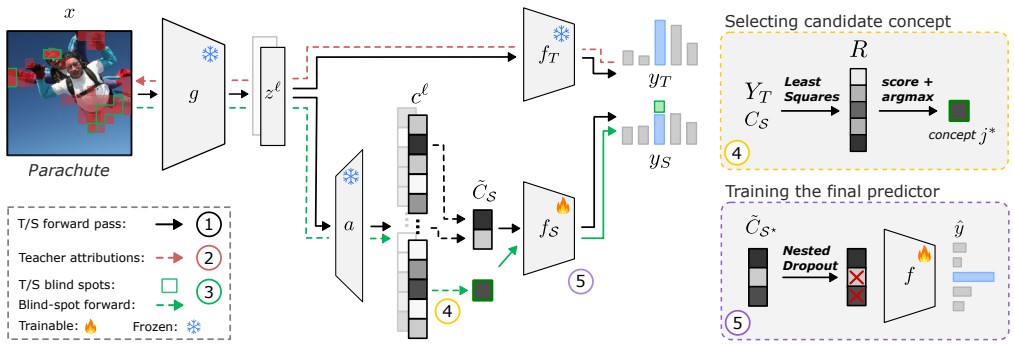


Figure 2: **Overview of AnyBottle.** An input x is encoded by the frozen backbone g into token embeddings z^ℓ, which a sparse autoencoder a maps to concept activations c^ℓ. Each selection round repeats four steps: (1) a forward pass through the teacher f_T and current student f_S; (2) teacher attributions over the input; (3) the resulting teacher/student blind-spots (green), used to restrict the SAE to blind-spot tokens; and (4) scoring the resulting candidates by least-squares residual R against the teacher's output Y_T to pick the next concept $j^\star$, added to $\mathcal{S}$. This repeats until selection converges on $\mathcal{S}^\star$. Once selection terminates, (5) the final predictor f is trained once on rescaled $\tilde{C}_{\mathcal{S}^\star}$ with nested dropout, yielding a head that predicts accurately from any concept prefix.

Adaptive, Variable-Budget Inference. A separate strand asks how many concepts a *given input* needs, and whether one model can serve multiple budgets without retraining, an idea recurring across ordered, prefix-usable representations (Bachmann et al., 2025; Wen et al., 2025). Nested Dropout (Rippel et al., 2014) stochastically drops an ordered suffix of hidden units so any prefix remains usable, an effect Matryoshka Representation Learning (Kusupati et al., 2022) achieves via nested loss terms instead; Matryoshka CBM (Chen et al., 2026) applies this to concept bottlenecks, ordering a given vocabulary and serving fixed budgets via nested heads. The Variational Information Pursuit line (Chattopadhyay et al., 2023a; 2024) instead studies confidence-driven sequential querying, but over a user-defined or LLM-generated query set. AnyBottle instead applies Nested Dropout to its own teacher-discovered vocabulary, letting confidence decide how many concepts to consume.

Sparse Autoencoders as a Concept Source. SAEs (Bricken et al., 2023; Templeton et al., 2024; Bussmann et al., 2025) are increasingly used as a concept source in place of a fixed, LLM/VLM-queried vocabulary (Oikarinen et al., 2023; Shang et al., 2024). Rao et al. (2024) discovers concepts via SAE first and names them afterward; Wittenmayer et al. (2026) pursues a related decomposition at foundation-model scale. Concept-SAE (Ding et al., 2025) and VL-SAE (Shen et al., 2025) instead inject supervision into the SAE objective, aligning a subset of latents to externally specified concepts. Rocchi-Henry et al. (2025) argue CBMs and SAEs are the same geometric object, differing only in how their cone of directions is selected, supervision versus sparse coding; AnyBottle sits between these poles, performing teacher-supervised selection over an already-trained, frozen SAE dictionary.

3 AnyBottle

In this section, we describe AnyBottle's process for obtaining a compact, task-specific CBM from an unsupervised concept pool. It follows the bottom-up approach of performing concept grounding before semantics (Knab et al., 2026). An SAE first grounds a large pool of unnamed patterns (Sec. 3.1). AnyBottle then selects among them by their statistical relationship to a black-box teacher's behavior (Sec. 3.2). The predictor is trained on the final pool with nested dropout for a sample-adaptive concept budget at inference (Sec. 3.3). Naming the selected concepts happens last, after selection is complete (Sec. 3.4). Fig. 2 overviews the recipe (pseudocode: Alg. 1); details follow.

Notation. We adopt the formalism of Knab et al. (2026): a CBM is an input encoder $g : \mathcal{X} \to \mathcal{Z}$, a concept grounding mechanism $h : \mathcal{Z} \to \mathcal{C}$ mapping embeddings to concept activations, and a predictor $f : \mathcal{C} \to \mathcal{Y}$, trained on $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \subseteq (\mathcal{X}, \mathcal{Y})$ with $\mathbf{y}_i$ a class label or regression target. Encoders often operate at the token level (e.g. image patches, word tokens) rather than on a single global embedding: we write $\mathbf{z}^\ell \in \mathbb{R}^D$, $\ell \in \{1, \dots, L\}$, for an individual token's embedding, and $\mathbf{z} = \frac{1}{L}\sum_{\ell=1}^{L} \mathbf{z}^\ell$ for the pooled, global embedding.

ANYBOTTLE follows a teacher-student approach. The teacher is a black-box predictor, e.g. an MLP head, trained on the frozen encoder's global embeddings: $f_T(\mathbf{z}) = \mathbf{y}_T \in \mathbb{R}^{|\mathcal{Y}|}$. ANYBOTTLE's predictor f is trained on the activations $C_{\mathcal{S}^\star}$ of a small, selected subset of concept indices $\mathcal{S}^\star \subseteq \{1, \dots, M\}$, chosen so f predicts $\mathbf{y}_T$ as closely as possible. Since $\mathcal{S}^\star$ is unknown in advance, selection proceeds in rounds: each round, the current predictor $f_{\mathcal{S}}$ is trained on the current candidate set $\mathcal{S}$ and guides which concept to add next (Sec. 3.2). Once selection converges on $\mathcal{S}^\star$, f is trained from $f_{\mathcal{S}}$ with nested dropout (Sec. 3.3) into the final, sample-adaptive predictor.

3.1 OBTAINING THE INITIAL CONCEPT POOL

ANYBOTTLE instantiates the concept space $\mathcal{C}$ with an unsupervised concept pool: a sparse autoencoder (SAE) $a : \mathcal{Z} \to \mathcal{C}$, trained on token embeddings from the frozen encoder g. Applied independently to each token, it gives candidate concept activations $\mathbf{c}^\ell = a(\mathbf{z}^\ell) \in \mathbb{R}^M_{\geq 0}$ for $\ell \in \{1, \dots, L\}$: the activations of M candidate concepts, shared across all inputs. We first train the SAE on a general corpus, without task-specific information. Following Braun et al. (2024), it can then be fine-tuned to the target task: we pass the SAE's reconstructed embedding $\hat{\mathbf{z}}$ through the frozen teacher head in place of $\mathbf{z}$, minimizing the KL divergence between $f_T(\mathbf{z})$ and $f_T(\hat{\mathbf{z}})$ jointly with a standard reconstruction term. In practice, this can make concepts more task-suitable and yield a more compact concept space.

For selection, we pool each concept's token-level activations into a single value per input: $c_{ij} = \max_\ell(c^\ell_{ij})$, keeping concept j's largest activation among all tokens of input i. We compute this once, over the full dataset, giving $C \in \mathbb{R}^{N \times M}$, and reuse it throughout selection. Further note, SAE activations differ widely in strength across concepts, so we rescale before training any predictor. For each concept $j \in \mathcal{S}$, a per-concept scale π_j is set from the p-th percentile of its training-set activations, giving $\tilde{C}_{ij} = \text{clip}\left(\frac{C_{ij}}{\pi_j + \epsilon}, 0, \tau_{\max}\right)$. This outlier-robust rescaling is refit whenever $\mathcal{S}$ changes; both $f_{\mathcal{S}}$ and the final f train and evaluate only on $\tilde{C}$, never raw C.

3.2 ADAPTIVE CONCEPT SELECTION

Given the pooled activation matrix C and the teacher f_T, ANYBOTTLE iteratively expands a concept selection $\mathcal{S}$, inspired by Orthogonal Matching Pursuit (OMP) (Pati et al., 1993). Each round adds the concept that best reduces the residual between the current selection and the teacher's output, then trains a predictor $f_{\mathcal{S}}$ on the new selection, which guides the next round. We write $\mathbf{c}_j \in \mathbb{R}^N$ for the j-th column of C: concept j's pooled activation across all N inputs.

Throughout this subsection, $C_{\mathcal{S}}$ (the columns of C indexed by $\mathcal{S}$) and $\mathbf{c}_j$ denote these raw, pooled activations, not the rescaled $\tilde{C}$ of Sec. 3.1. Both the residual fit (Eq. 1) and candidate scoring (Eq. 2) instead use an unfit, column-wise normalization, $\|\mathbf{c}_j\|_2$, recomputed each round; this keeps candidate ranking scale-invariant, independent of the rescaling used to train the predictor. Selection is driven by sample- and region-level (attribution) disagreement between f_T and $f_{\mathcal{S}}$, described below.

Computing the Residual. We write $Y_T \in \mathbb{R}^{N \times |\mathcal{Y}|}$ for the teacher's outputs over the full dataset. Each round, we fit $C_{\mathcal{S}}$ to Y_T via least squares, augmenting $C_{\mathcal{S}}$ with a constant column, $\hat{C}_{\mathcal{S}} = [C_{\mathcal{S}}, \mathbf{1}]$, so that $\mathcal{S}$ only needs to explain deviations from Y_T's mean rather than the mean itself: without this column, a concept that fires near-uniformly across the dataset could score as informative purely by matching the residual's mean, rather than by explaining any per-example variance. This yields coefficients $\hat{W} \in \mathbb{R}^{(|\mathcal{S}|+1) \times |\mathcal{Y}|}$ and a residual R:

$$\hat{W} = (\hat{C}_{\mathcal{S}}^\top \hat{C}_{\mathcal{S}})^{-1} \hat{C}_{\mathcal{S}}^\top Y_T, \qquad R = Y_T - \hat{C}_{\mathcal{S}} \hat{W} \tag{1}$$

with W recovered by dropping $\hat{W}$'s intercept row. R then selects the next concept from $\{1, \dots, M\} \setminus \mathcal{S}$, choosing whichever best explains it.

Candidate Proposal via Attribution Disagreement. Scoring every concept against R each round is expensive and unnecessary, since most are irrelevant to the current failure mode. ANYBOTTLE instead restricts candidates using token-level disagreement between the teacher's (f_T) and current predictor's ($f_{\mathcal{S}}$) attributions. We take the k inputs with largest residual norm $\|R_i\|_2$ as focus set $\mathcal{I}$, and for each $i \in \mathcal{I}$ compute a token-importance map for f_T via gradient attribution w.r.t. the ground-truth label; for $f_{\mathcal{S}}$, attribution is computed analytically, each selected concept's contribution

to the target-class logit (its head weight times pooled activation) is assigned to the token where the concept attains its maximum activation. Tokens the teacher attends to but $f_{\mathcal{S}}$ does not are this round's *blind-spots* B_i; summing per-token activations over them, $\text{agg}_i = \sum_{\ell \in B_i} \mathbf{c}^\ell$, and dropping already-selected or below-threshold concepts leaves the r highest-activating candidates per sample.

Aggregating across $\mathcal{I}$, each surviving concept j accumulates a distinct-sample count n_j and mean activation strength δ_j; we discard concepts found in too few samples and rank the rest by $\delta_j \cdot \log(1 + n_j)$, favoring strong, broadly implicated concepts over one-off spikes. The top q form the shortlist $\mathcal{J}$.

Candidate Scoring and Selection. Given the shortlist $\mathcal{J}$ and the residual R, each candidate $j \in \mathcal{J}$ is scored by

$$s_j = \frac{\|\mathbf{c}_j^\top R\|_2}{\|\mathbf{c}_j\|_2 + \epsilon}, \qquad j^\star = \arg\max_j s_j. \tag{2}$$

The top-scoring candidate is added to the pool: $\mathcal{S} \leftarrow \mathcal{S} \cup \{j^\star\}$. Notably, the first round has no prior concept set to fit against the target, so the first concept is instead scored against the mean-centered teacher output (Eq. 1 with $\mathcal{S} = \emptyset$), over the full candidate set: $\mathcal{J} = \{1, \ldots, M\}$. For many-class tasks, this round can instead be jump-started with one heuristically class-specific concept per class, before residual-driven selection proceeds as usual.

Retraining and Stopping. After adding a concept, R is recomputed and $f_{\mathcal{S}}$ retrained on $\tilde{C}_{\mathcal{S}}$, providing validation accuracy for model selection and student attribution for the next round. We track explained variance $\text{EV}_{\mathcal{S}} = 1 - \text{var}(R)/\text{var}(Y_T)$ as a diagnostic: as $\mathcal{S}$ grows, $\text{EV}_{\mathcal{S}}$ approaches the upper bound of the SAE's dictionary, serving as stopping signal and concept-pool-quality diagnostic.

Selection terminates when: (1) no candidate survives attribution-based filtering, or all candidate scores fall below a minimum threshold; (2) the accuracy gap to the teacher falls within a tolerance; (3) neither validation accuracy nor $\text{EV}_{\mathcal{S}}$ improves for a set number of rounds; or (4) a round limit is reached. We then return the best-validation-accuracy concept set $\mathcal{S}^\star$, its predictor $f_{\mathcal{S}^\star}$, and its per-concept scales $\{\pi_j\}_{j \in \mathcal{S}^\star}$, retaining the discovery order for training the final predictor.

3.3 Nested Dropout

Once selection terminates with $\mathcal{S}^\star$ and its discovery order fixed, we initialize a final predictor f from $f_{\mathcal{S}^\star}$ and retrain it on $\tilde{C}_{\mathcal{S}^\star}$ using nested dropout (Rippel et al., 2014). At each training step, we sample a prefix length l uniformly from $\{1, \ldots, |\mathcal{S}^\star|\}$, zero every concept activation beyond the l-th in discovery order, and train f on the resulting $\tilde{C}^{(l)}_{\mathcal{S}^\star}$ with cross-entropy loss against ground-truth labels. Since every step truncates to a new sampled prefix, f learns to predict correctly from any prefix length, not just the full set. This enables an optional adaptive inference mode: concepts are consumed one at a time in discovery order, and inference halts at the shortest prefix l whose predicted class confidence (max softmax of $f(\tilde{C}^{(l)}_{\mathcal{S}^\star})$) exceeds a threshold, with concepts below a minimum per-input activation masked out to retain only those that genuinely fire. This adapts the concept budget per input: fewer concepts for confident inputs, more for harder ones.

3.4 Concept Naming

Because AnyBottle selects only the concepts a task needs, naming applies to a handful rather than the full SAE dictionary. Related work commonly uses a single CLIP-similarity pass (Oikarinen et al., 2023; Yang et al., 2023; Prasse et al., 2025), cheap but prone to missing a concept's exact counterexamples and boundaries. Recent agentic naming methods (Shaham et al., 2024) (cf. Sec. C) instead test these boundaries directly, e.g. by generating or editing interventionist images to check whether a concept tracks color or shape; more tools and testing rounds improve reliability but are too costly over a full, unselected dictionary. Naming AnyBottle's few dozen selected concepts this way remains cheap enough as an automated first pass. Even so, a human should still check these proposals, ideally interactively (Schramowski et al., 2020; Friedrich et al., 2023): AnyBottle's compactness makes checking a few dozen names feasible where checking thousands is not. This also catches what automated naming alone cannot: a well-named concept the predictor may still rely on for the wrong reason, tracking a spurious correlation rather than the task itself; such shortcuts are easier to spot once the compact set is named and reviewed against domain knowledge (Steinmann et al., 2024; Geirhos et al., 2020; Lapuschkin et al., 2019).

4 EXPERIMENTAL EVALUATIONS

We evaluate ANYBOTTLE across multiple datasets and modalities to answer the following questions: does it match or exceed annotation-free CBM baselines on accuracy, compactness, and concept consistency (**RQ1**); does selection close genuine teacher-student disagreement, allocating more concepts to harder tasks (**RQ2**); does nested dropout let inference allocate more concepts to harder individual samples (**RQ3**); which components of the selection mechanism drive its compactness and quality (**RQ4**); and does the recipe transfer to text, across domains and teacher paradigms (**RQ5**).

Datasets. We evaluate on six vision and two text classification datasets, chosen to span coarse- and fine-grained discrimination, small and large label vocabularies, and general versus specialized domains. Vision: *Imagenette* (Howard, 2019) (10-class, coarse), *Dogs-10* (10 fine-grained breeds), *Machines-15* (15 fine-grained machine types) and *Mixed-20* (20 animal and machine classes), *Flowers-102* (Nilsback and Zisserman, 2008) (102-class, fine-grained), and *ISIC-7* (Codella et al., 2019; Tschandl et al., 2018) (7-class dermoscopic skin lesions, a specialised domain). Dogs-10, Machines-15, and Mixed-20 are curated class subsets of ImageNet; full class lists are given in Appendix G. Text: *AG News* (Zhang et al., 2015) (topic classification) and *PubMed-20k* (Dernoncourt and Lee, 2017) (biomedical sentence-role classification). Further details are given in App. G.1.

Models. ANYBOTTLE uses CLIP-DINOiser (Wysoczańska et al., 2024) (vision) or Gemma (Team et al., 2024) (text) as backbone, with concepts from a sparse autoencoder with task-aware fine-tuning. We compare against four annotation-free or compactness-oriented CBM baselines (Label-free CBM (Oikarinen et al., 2023), Res-CBM (Shang et al., 2024), DN-CBM (Rao et al., 2024), UCBM (Schrodi et al., 2025)); backbones, concept sources, and classifier heads for each are given in App. G.2. For vision and the mlp text teacher, the black-box teacher is a two-layer MLP trained per-backbone, keeping reported gaps within one embedding space; the verbalizer teacher trains no parameters. We also compare against a linear probe over the full, unselected SAE dictionary, and one restricted to active neurons only, isolating selection gains from those of any smaller, arbitrary subset. Full configuration and hyperparameter details including automatic concept naming are in App. G.2.

Metrics. Our primary compactness measure is $K = |\mathcal{S}^\star|$, the selected concept set size (for full-dictionary SAE probes, the number of active SAE features). We report balanced accuracy throughout; since methods use different backbones, our primary metric is the accuracy gap Δ between student (or CBM baseline) and teacher (black-box model), on the same backbone. All results are averaged over three random seeds. Since f predicts at any prefix length up to K, we evaluate across prefix lengths and report $\mathrm{C}@q$ for $q \in \{90, 95, 98\}$ (the smallest prefix reaching q percent of the student's own full-budget accuracy), plus $\mathrm{C}@K = K$, the full discovered budget, for reference. To assess concept quality independent of accuracy, we use the C^2 score (Parchami-Araghi et al., 2026) for vision: semantic consistency of each concept's most-activating inputs, averaged over the concept set. ANYBOTTLE's concepts are scored at the region level; the four CBM baselines produce only one scalar activation per image and are scored at the image level. For text, we report an autointerp score (AIS) (Bills et al., 2023; Paulo et al., 2025): an LLM describes a concept from its top-activating documents, a second LLM predicts its activation on held-out documents from that description alone, and the score is the correlation between predicted and true activation. Further details in App. G.3.

4.1 EXPERIMENTAL EVALUATIONS

Adaptive Concept Bottlenecks on par with Black-box Models (RQ1). Fig. 3 evaluates whether ANYBOTTLE matches a black-box teacher's accuracy while using fewer, more consistent concepts than existing baselines across diverse vision tasks, without manual concept annotations. ANYBOTTLE reaches teacher accuracy on most datasets, comes within two points on Dogs-10 and ISIC-7, and surpasses it on Flowers-102: routing through a small, selected concept set instead of the teacher's embedding space costs almost no accuracy. This compactness is not just discarded capacity: it also improves concept consistency. Against a linear probe over the same SAE dictionary, ANYBOTTLE uses 16–60$\times$ fewer concepts on five of six datasets, while matching or improving image-level (and patch-level, cf. Fig. 7) consistency on four. On ISIC-7, the one dataset needing a larger budget, ANYBOTTLE's set is still only half the active SAE dictionary size and beats the probe's accuracy. The pattern holds against the four annotation-free baselines: fewer concepts on all but one dataset, higher image-level consistency on all but two of twenty-four dataset–baseline pairs, and competitive accuracy throughout, with ANYBOTTLE's teacher-gap Δ matching or beating every baseline except

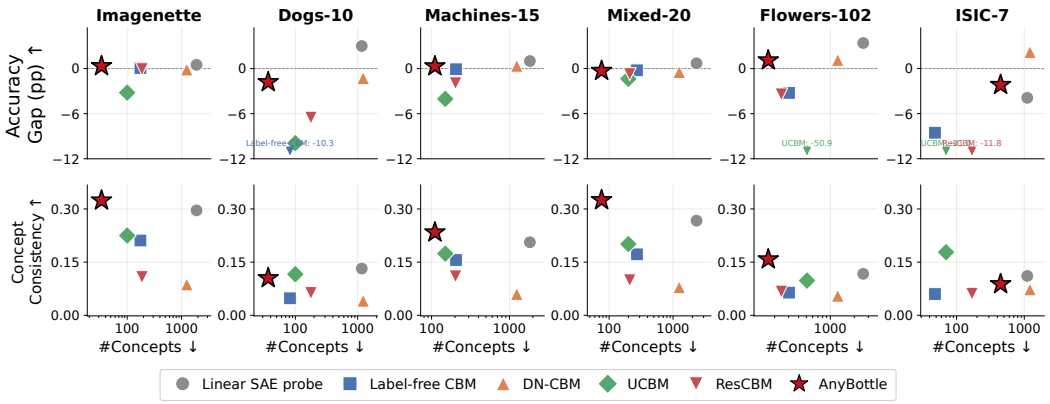


Figure 3: **Top: ANYBOTTLE matches or exceeds black-box teacher accuracy at a fraction of the concepts used.** Accuracy gap (Δ (pp)) to the teacher, plotted against concept count (log scale); ANYBOTTLE reaches the teacher across all six datasets with orders of magnitude fewer concepts than the baselines. **Bottom: ANYBOTTLE's smaller bottleneck is also the more interpretable one.** Concept consistency (C^2_{img}) at the same budgets matches or exceeds every baseline, showing the compactness above is not simply discarded capacity. Points are seed means.

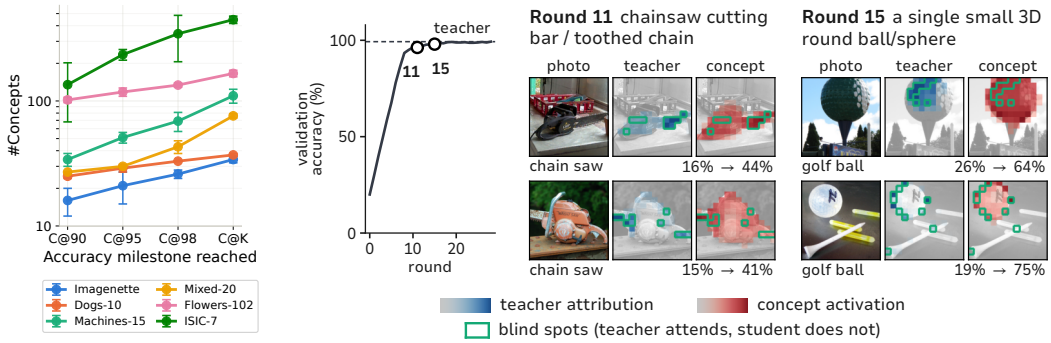


Figure 4: *Left:* **ANYBOTTLE reaches near-full accuracy with a fraction of its concept budget.** Concepts needed to reach 90/95/98% of the student's full-budget accuracy, per dataset; K shown for reference. *Right:* **Early rounds close large blind spots; later rounds refine with diminishing returns.** Round-by-round validation accuracy on Imagenette, with two exemplary rounds detailed.

on ISIC-7, where DN-CBM's edge costs an order of magnitude more concepts. Raw accuracies stay broadly comparable across methods despite different backbones, except for a wider spread on ISIC-7 (Fig. 6). Overall, ANYBOTTLE matches or exceeds black-box teachers across tasks from coarse to fine-grained, at a fraction of the concepts annotation-free baselines require; ISIC-7, with its weak teacher, is the one setting where these gaps narrow across all methods.

ANYBOTTLE Allocates Concepts Where Accuracy Demands Them (RQ2). Fig. 4 shows concepts needed by the nested-dropout head to reach 90%, 95%, and 98% of its own full-budget accuracy, alongside the full discovered budget C@K. Most accuracy comes early: on Imagenette, Dogs-10, and Mixed-20, C@90 needs 27 concepts or fewer, with C@98 only a handful more. Machines-15 and Flowers-102 need a larger initial budget, matching their finer granularity, but show the same diminishing cost. ISIC-7 is the exception: C@90 already needs 135 concepts, and closing to C@98 needs more than double that, reflecting both task complexity (difficult and unbalanced classes) and a weak, noisy teacher. Fig. 4 (right) shows what drives this pattern round by round on Imagenette: round 11 adds a toothed-chain concept refining chainsaw recognition, round 15 a small-spheres concept improving golf-ball recognition, both closing blind spots the teacher attends to but the student does not. Early rounds close large, class-defining blind spots, while later rounds refine coverage with diminishing accuracy gain, confirming that allocation tracks task difficulty by closing genuine teacher-student disagreement, not by adding concepts uniformly.

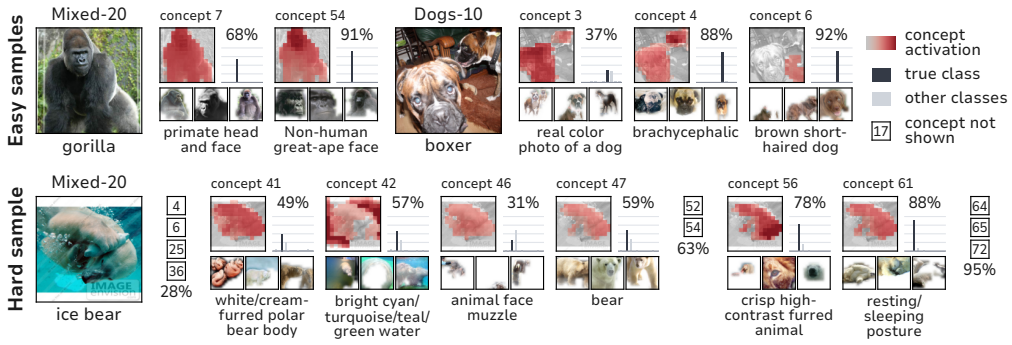


Figure 5: **Adaptive inference uses more concepts only on samples that need them.** Two easy samples (top) and one hard sample (bottom) given the task and selected concept set. For each sample, the activating concepts to reach 90% confidence are shown, together with three exemplars. Boxed numbers mark firing concepts left out for visibility.

Adaptive Inference Allocates Concepts by Sample Difficulty (RQ3). Fig. 5 shows the nested-dropout head consuming concepts in discovery order at inference, for two easy samples and one hard sample. Both easy samples converge quickly: two concepts take a gorilla, and three a boxer, past 90% confidence. The hard sample, an ice bear (Mixed-20), draws on far more of the budget, crossing 50% confidence only after 5 concepts and needing 14 for 95%. This is nested dropout's intended effect: a head accurate at any prefix length lets inference stop as soon as the prefix suffices, rather than always spending the full budget K.

Table 1: **Every component of AnyBottle's selection facilitates compact, consistent, and accurate concept sets.** Ablation over selection strategy, averaged over vision datasets. LS: least-squares scoring (vs. random selection). Loc: localized candidate search. PF: prefiltering.

LS	Loc	PF	K ↓	C^2_{patch} ↑	Δ ↑ (%)
✓	✓	✓	145.0	0.362	−0.4
✓	✓	–	165.7	0.345	−0.9
✓	–	–	190.5	0.314	−1.3
–	–	–	231.3	0.272	−2.1

Ablating Selection Strategy (RQ4). To test AnyBottle's selection mechanism, Tab. 1 removes the prefiltering (PF), the localization step (Loc), and the least-squares (LS) scoring in turn. Removing any part progressively reduces compactness and concept consistency while increasing the accuracy gap to the teacher, confirming that each stage contributes to finding compact, consistent and accurate concept sets. Per-dataset results in Tab. 3. Tab. 4 in the appendix reports two additional ablations. Continuous versus discrete concept representations: AnyBottle works with both types, though a discrete set usually has to be larger for the same accuracy. Additionally, task-aware SAE fine-tuning helps most when the pretrained dictionary is poorly matched to the target domain, notably on ISIC-7 and Flowers-102.

Generalization Across Domains (RQ5). Tab. 2 tests whether AnyBottle transfers beyond vision by swapping only the backbone and concept pool, along two axes: domain (general-topic AG News vs. specialized biomedical PubMed-20k) and teacher paradigm (mlp-clf, an MLP head on a frozen LM's hidden state, vs. verb-clf, a verbalizer teacher reading predictions directly off a frozen, instruction-tuned model's next-token distribution), each with its own backbones. Against mlp-clf, AnyBottle closely matches teacher accuracy on both datasets and matches or outperforms the linear SAE probe in both accuracy and interpretability, using or-

Table 2: **AnyBottle generalizes across domains and teacher paradigms.** Evaluation of AnyBottle on two text datasets with a trained mlp-clf teacher and a zero-shot verbalizer teacher, verb-clf. We report concept count K, AutoInterp score (AIS), teacher gap (Δ), and absolute balanced accuracy (Acc).

	Method	K ↓	AIS ↑	Δ (%) ↑	Acc (%) ↑
AG News	SAE probe	31477	0.327±0.009	−0.12±0.38	90.42±0.38
	AnyBottle (mlp)	93.7±6.4	0.469±0.006	−0.18±0.25	90.36±0.25
	SAE probe	9710	0.157±0.001	+18.57±0.98	88.25±0.98
	AnyBottle (verb)	30.0±5.0	0.094±0.020	+0.82±1.89	70.50±1.89
PubMed	SAE probe	25893	0.453±0.000	−2.47±0.87	73.58±0.87
	AnyBottle (mlp)	100.0±2.6	0.526±0.004	−0.02±0.23	76.03±0.23
	SAE probe	7853	0.087±0.000	+43.04±1.66	70.92±1.66
	AnyBottle (verb)	23.3±2.5	0.146±0.012	+15.93±2.04	43.81±2.04

ders of magnitude fewer concepts. This mirrors the vision results: a small, selected concept set recovers near-teacher accuracy regardless of domain. The verb-clf performs worse overall, only slightly above chance on PubMed-20k. ANYBOTTLE again matches the teacher on AG News and substantially outperforms it on PubMed-20k, but falls short of the linear SAE probe there: teacher quality caps what ANYBOTTLE's selection can recover. This confirms ANYBOTTLE generalizes across teacher paradigms, while showing its limits when paired with a weak teacher.

5 DISCUSSION & LIMITATIONS

ANYBOTTLE combines several known components into one recipe: SAE-based concept discovery, teacher-student distillation via OMP-style residual selection, nested-dropout prediction, and agentic, automated concept naming. Together, they yield task-specific CBMs that are compact and accurate without concept annotations. We discuss the main limitations below.

Faithfulness, Not Correctness. A concept is selected for reducing disagreement with the teacher, not for being correct: a teacher relying on a shortcut is matched by a bottleneck that reproduces it. However, ANYBOTTLE's small concept budget should make such shortcuts easier to spot and revise in a human-in-the-loop setting (Schramowski et al., 2020; Stammer et al., 2021).

Limits of the Concept Pool, Backbone and Teacher. ANYBOTTLE's performance is bound by its backbone and pretrained components: it cannot find a concept absent from the SAE dictionary, even if the teacher relies on other dimensions. Task-aware fine-tuning helps thereby, but is harder with scarce target-domain data. And while ANYBOTTLE can outperform a poor teacher to some extent, its performance remains tied to the teacher's guidance. Because concepts are scalar, non-negative activations rather than embeddings, concept leakage is bounded, and discrete activations can reduce it further. Sophisticated automated naming, e.g., MAIA, identifies concept meaning more accurately than coarse image-text similarity, which helps users spot unintended semantics. Its counterfactual boundary testing can further clarify what a discovered concept represents. A separate concern is whether a concept correlating strongly with the label is a shortcut for a finer-grained one. We argue that this is not automatic: if a task's true separating factor is coarse, a coarse concept is a sufficient decomposition, not a shortcut. Since selection ties concept count to task difficulty by construction, it stops once the teacher's residual is explained.

Computational Efficiency. ANYBOTTLE's iterative loop trains a predictor f_S each round, but training a linear head is cheap, so cost scales with rounds, not model complexity. Rounds can be reduced by adding multiple concepts at once or jump-starting selection for harder problems. This cost is offset downstream, where the smaller bottleneck substantially cuts the cost of concept naming.

6 CONCLUSION

We presented ANYBOTTLE, a recipe for building compact, task-specific concept bottlenecks from only a frozen backbone and an unsupervised concept pool: teacher-guided selection yields compact concept sets, globally and per sample, and nested dropout adapts their size to what each prediction needs. Across diverse domains and datasets, ANYBOTTLE matches or exceeds teacher accuracy with an order of magnitude fewer concepts than annotation-free baselines, using a single recipe that transfers across domains and modalities. Overall, ANYBOTTLE shows that annotation-free concept bottlenecks need not trade compactness for coverage: a small, discovered vocabulary can be as expressive as a much larger, fixed one, while remaining tractable to inspect.

Future work. Several extensions follow from Sec. 3's design choices. The concept pool need not be an SAE: any unsupervised decomposition exposing per-token activations, e.g. Bağcı et al. (2026), is compatible. A logic-based head (Vemuri et al., 2026) could replace the linear predictor, pairing naturally with compositional naming (Mu and Andreas, 2020) as an alternative to MAIA. The recipe also extends to other tasks as long as suitable teachers are available. ANYBOTTLE could be extended with an explicit grounding-verification step, checking that a concept's evidence is not merely residual-reducing but genuinely correct, as Fang et al. (2026) do for medical imaging. Finally, human or automated verification of MAIA's naming remains open.

REPRODUCIBILITY STATEMENT

An interactive visualiser lets readers click through ANYBOTTLE's iterative concept selection process for our vision dataset evaluations (Sec. B, link). We also provide anonymous repo for ANYBOTTLE at https://anonymous.4open.science/r/AnyBottle-Code-4E76/. Appendix G specifies datasets (Sec. G.1), backbones, teachers, SAE architectures, fine-tuning objectives, and hyperparameters for both modalities (Sec. G.2), and Appendix G.3 defines every reported metric precisely, including C@q, C^2, and the AutoInterp score. Appendix G.2 also documents implementation details of every annotation-free CBM baseline and G.4 shows the prompts for MAIA evaluations. Appendix E reports per-dataset breakdowns underlying every averaged result in the main text, the accuracy-gap stopping tolerance ablation (Sec. E.6), and the full text results (Sec. E.7). All results are averaged over three seeds, with model selection throughout by held-out validation accuracy.

AI USE STATEMENT

We used generative AI tools for the following required-disclosure tasks: refining hypotheses and design or feedback on research methodology and experiments (e.g. framing and stress-testing the discussion points in Section 5, and aspects of ablation and evaluation design); implementing methods, via agentic coding that wrote and executed code for ANYBOTTLE and the baselines; and interpreting results, assisting in reading experimental outputs when drafting the corresponding text. We did not use generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, or provide ingredients for or assist in writing proofs, and did not use generative AI tools for translation, dataset cleaning or reformatting, or qualitative/thematic data analysis.

Additionally, we used generative AI tools for the following recommended-disclosure tasks: suggesting experimental parameters; creating and modifying scientific figures; editing code; identifying relevant literature and formatting references; and drafting and restructuring parts of the manuscript (text, section structure, and tables) to improve readability.

The authors reviewed AI-generated text, figures, and a portion of the AI-generated and AI-executed code visually for plausibility; we did not systematically test or verify all AI-generated code, nor exhaustively review all of it. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025.

Doğukan Bağcı, Bernt Schiele, Simone Schaub-Meyer, Jonas Fischer, and Robin Hesse. Interpretability without tradeoffs: Disentangling polysemanticity at equal predictive performance. *CoRR*, abs/2605.31304, 2026.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, 2017.

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

Milan Bhan, Yann Choho, Jean-Noël Vittaut, Nicolas Chesneau, Pierre Moreau, and Marie-Jeanne Lesot. Towards achieving concept completeness for textual concept bottleneck models. In *EMNLP (Findings)*, pages 2007–2024, 2025.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neu-

rons in language models. *OpenAI Blog*, 2023. URL https://openai.com/index/language-models-can-explain-neurons-in-language-models/.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning. *Advances in Neural Information Processing Systems*, 37:107286–107325, 2024.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, 2023.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=m25T5rAy43.

Josep Lopez Camuñas, Christy Li, Tamar Rott Shaham, Antonio Torralba, and Agata Lapedriza. Openmaia: a multimodal automated interpretability agent based on open-source models. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.

Aditya Chattopadhyay, Kwan Ho Ryan Chan, Benjamin David Haeffele, Donald Geman, and Rene Vidal. Variational information pursuit for interpretable predictions. In *International Conference on Learning Representations (ICLR)*, 2023a.

Aditya Chattopadhyay, Ryan Pilgrim, and Rene Vidal. Information maximization perspective of orthogonal matching pursuit with applications to explainable AI. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Aditya Chattopadhyay, Kwan Ho Ryan Chan, and Rene Vidal. Bootstrapping variational information pursuit with large language and vision models for interpretable image classification. In *International Conference on Learning Representations*, volume 2024, pages 33246–33272, 2024.

Ziye Chen, Hongbin Lin, Jie Li, and Lijie Hu. Matryoshka concept bottleneck models. *arXiv preprint arXiv:2605.20612*, 2026.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Franck Dernoncourt and Ji Young Lee. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing.

Jianrong Ding, Muxi Chen, Chenchen Zhao, and Qiang Xu. Concept-SAE: active causal probing of visual model behavior. *CoRR*, abs/2509.22015, 2025.

Yingying Fang, Haijie Xu, Shuang Wu, Mariathasan Anish, and Guang Yang. Towards fine-grained and verifiable concept bottleneck models. *CoRR*, abs/2605.14210, 2026.

Thomas Fel, Victor Boutin, Louis Béthune, Rémi Cadène, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology for exploring the mitigation of shortcut behaviour. *Nature Machine Intelligence*, 5(3):319–330, 2023.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations (ICLR)*, 2022.

Jeremy Howard. Imagenette, 2019. URL https://github.com/fastai/imagenette/.

Been Kim, John Hewitt, Neel Nanda, Noah Fiedel, and Oyvind Tafjord. Because we have llms, we can and should pursue agentic interpretability. *arXiv preprint arXiv:2506.12152*, 2025.

Patrick Knab, David Steinmann, Christian Bartelt, Kristian Kersting, Bernt Schiele, Thomas Seidl, Udo Schlegel, and Wolfgang Stammer. What's in the bottle? a survey and roadmap of concept bottleneck models. *Transactions on Machine Learning Research*, 2026.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, pages 5338–5348, 2020.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1–8, 2019.

Josh Magnus Ludan, Qing Lyu, Yue Yang, Liam Dugan, Mark Yatskar, and Chris Callison-Burch. Interpretable-by-design text understanding with iteratively generated concept bottleneck. *arXiv preprint arXiv:2310.19660*, 2023.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Zhi Nguyen, Katherine Ding, and Rene Vidal. Hierarchical concept embedding & pursuit for interpretable image classification. *CoRR*, abs/2602.11448, 2026.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: automatic description of neuron representations in deep vision networks. In *International Conference on Learning Representations (ICLR)*, 2023.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023.

Konstantinos Panousis, Dino Ienco, and Diego Marcos. Coarse-to-fine concept bottleneck models. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 105171–105199, 2024.

Amin Parchami-Araghi, Sukrut Rao, Jonas Fischer, and Bernt Schiele. Fact: Faithful concept traces for explaining neural network decisions. *Advances in Neural Information Processing Systems*, 38: 122410–122453, 2026.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *International Conference on Machine Learning (ICML)*, 2025.

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

Katharina Prasse, Patrick Knab, Sascha Marton, Christian Bartelt, and Margret Keuper. Dcbm: Data-efficient visual concept bottleneck models. In *International Conference on Machine Learning (ICML)*, pages 49752–49782, 2025.

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 (ECCV)*, pages 444–461, 2024.

Oren Rippel, Michael A. Gelbart, and Ryan P. Adams. Learning ordered representations with nested dropout. *International Conference on Machine Learning (ICML)*, pages 1746–1754, 2014.

Alexandre Rocchi-Henry, Thomas Fel, and Gianni Franchi. A geometric unification of concept learning with concept cones. *CoRR*, abs/2512.07355, 2025.

Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

Simon Schrodi, Julian Schur, Max Argus, and Thomas Brox. Selective concept bottleneck models without predefined concepts. *Transactions on Machine Learning Research*, 2025. Preprint titled "Concept Bottleneck Models Without Predefined Concepts".

Lisa Schut, Nenad Tomašev, Thomas McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human–ai knowledge gap through concept discovery and transfer in alphazero. *Proceedings of the National Academy of Sciences*, 122(13):e2406675122, 2025.

Andrei Semenov, Vladimir Ivanov, Aleksandr Beznosikov, and Alexander Gasnikov. Sparse concept bottleneck models: Gumbel tricks in contrastive learning. *CoRR*, abs/2404.03323, 2024.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *International Conference on Machine Learning (ICML)*, 2024.

Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11030–11040, 2024.

Shufan Shen, Junshu Sun, Qingming Huang, and Shuhui Wang. Vl-SAE: interpreting and enhancing vision-language alignment with a unified concept set. *CoRR*, abs/2510.21323, 2025.

Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3619–3629, 2021.

Wolfgang Stammer, Antonia Wüst, David Steinmann, and Kristian Kersting. Neural concept binder. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 71792–71830, 2024.

David Steinmann, Felix Divo, Maurice Kraus, Antonia Wüst, Lukas Struppek, Felix Friedrich, and Kristian Kersting. Navigating shortcuts, spurious correlations, and confounders: From origins via detection to mitigation. *arXiv preprint arXiv:2412.05152*, 2024.

Andong Tan, Fengtao Zhou, and Hao Chen. Explain via any concept: Concept bottleneck model with open vocabulary concepts. In *European Conference on Computer Vision (ECCV)*, pages 123–138, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5:180161, 2018. doi: 10.1038/sdata.2018.161.

Deepika SN Vemuri, Gautham Bellamkonda, Aditya Pola, and Vineeth N Balasubramanian. Logiccbms: Logic-enhanced concept-based learning. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6039–6048. IEEE, 2026.

Xin Wen, Bingchen Zhao, Ismail Elezi, Jiankang Deng, and Xiaojuan Qi. "principal components" enable a new language of images. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16641–16651. IEEE, 2025.

Kai Wittenmayer, Sukrut Rao, Amin Parchami-Araghi, Bernt Schiele, and Jonas Fischer. CFM: language-aligned concept foundation model for vision. In *European Conference on Computer Vision (ECCV)*, 2026.

Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. In *European Conference on Computer Vision*, pages 320–337. Springer, 2024.

Shin'ya Yamaguchi, Kosuke Nishida, Daiki Chijiwa, and Yasutoshi Ida. Zero-shot concept bottleneck models. *CoRR*, abs/2502.09018, 2025.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19187–19197, 2023.

Chih-Kuan Yeh, Been Kim, and Pradeep Ravikumar. Human-centered concept explanations for neural networks. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, volume 342 of *Frontiers in Artificial Intelligence and Applications*, pages 337–352. IOS Press, 2021.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

Supplementary Materials

A Impact Statement

This work aims to improve the interpretability of machine learning systems by producing concept bottleneck models that use fewer, more coherent concepts than existing annotation free approaches. Improved interpretability tools of this kind can support model debugging, regulatory compliance, and human oversight in deployed systems, including in specialized domains such as medical imaging, where our ISIC-7 evaluation is one example.

We note several considerations specific to this method. First, AnyBottle's concept bottleneck is trained to match a black box teacher's behavior rather than to independently verify correctness; as discussed in Sec. 5, if the teacher relies on a spurious correlation or shortcut, a faithful bottleneck may reproduce rather than expose that shortcut. Practitioners should not treat a compact, coherent bottleneck as proof that the underlying teacher is free of such issues. Second, concept names are generated by an automated agent without a mandatory human verification step in our experiments; we recommend human review before any such bottleneck is used to inform consequential decisions, particularly in safety critical domains. Third, as with any interpretability method, there is a risk that a plausible sounding concept based explanation is used to build unwarranted trust in a system's predictions, independent of whether the explanation is accurate; we see this as a general risk for the concept bottleneck literature rather than one specific to this work.

Our experiments use publicly available benchmark datasets (Imagenette, ImageNet derived subsets, Flowers-102, ISIC-7, AG News, PubMed-20k) and do not involve new data collection or human subjects. We are not aware of dual use concerns beyond those generally applicable to interpretability and machine learning research.

B AnyBottle Visualiser

We provide an anonymous link for readers to investigate the iterative concept selection process of AnyBottle of our vision dataset evaluations here: https://anonymous.4open.science/w/AnyBottle-Visualizer-E56F

C Additional Related Works

Automatic Interpretability Agents. Assigning human-readable semantics to a grounded concept (Knab et al., 2026) is typically done via a single, cheap pass, matching activating inputs against a fixed or open vocabulary (Bau et al., 2017; Oikarinen and Weng, 2023) or a trained captioning model (Hernandez et al., 2022). MAIA (Shaham et al., 2024) instead replaces this with an agent that iteratively designs and runs experiments on a concept before committing to a description, at a substantially higher per-concept cost; comparable agentic prescreening has since been explored elsewhere (Kim et al., 2025). AnyBottle's compactness and its naming choice are not independent decisions: MAIA is impractical over an SAE dictionary of thousands of atoms, but affordable once only a handful of concepts survive selection, a connection we are not aware of prior CBM work making explicit, though (Rao et al., 2024; Wittenmayer et al., 2026; Ding et al., 2025) face the same tradeoff implicitly, choosing cheaper, non-agentic naming over their respective concept pools.

D Algorithm

We provide an algorithmic overview of AnyBottle in Alg. 1.

E Additional Results

In this section we provide results from additional evaluations.

Algorithm 1 ANYBOTTLE

Cache teacher outputs Y_T and concept activations C over full concept pool ▷ Section Sec. 3.1
$R \leftarrow Y_T$ mean-centered; $\mathcal{S} \leftarrow \{\arg\max_{j\in\{1,\dots,M\}} \text{score}(\mathbf{c}_j, R)\}$ ▷ initial concept, scored against the intercept-only residual
repeat ▷ Section Sec. 3.2
  $R \leftarrow$ residual of least squares fit of $\mathcal{S}$ against Y_T
  $\mathcal{I} \leftarrow$ inputs with largest residual $\|R_i\|_2$
  $\mathcal{J} \leftarrow$ candidate concepts at teacher/student attribution *blind-spots* on $\mathcal{I}$
  $j^\star \leftarrow \arg\max_{j\in\mathcal{J}} \text{score}(\mathbf{c}_j, R)$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{j^\star\}$
  retrain current predictor $f_\mathcal{S}$ on $\tilde{C}_\mathcal{S}$; track best validation accuracy so far
until no valid candidate, accuracy gap closed, accuracy/EV plateau reached, or round limit reached
$\mathcal{S}^\star \leftarrow$ concept set of the best validation-accuracy round
$f \leftarrow$ retrain from $f_{\mathcal{S}^\star}$ on $\tilde{C}_{\mathcal{S}^\star}$ with nested dropout ▷ Section Sec. 3.3
automatically name each concept in $\mathcal{S}^\star$, subject to human review ▷ Section Sec. 3.4
return $\mathcal{S}^\star$, f

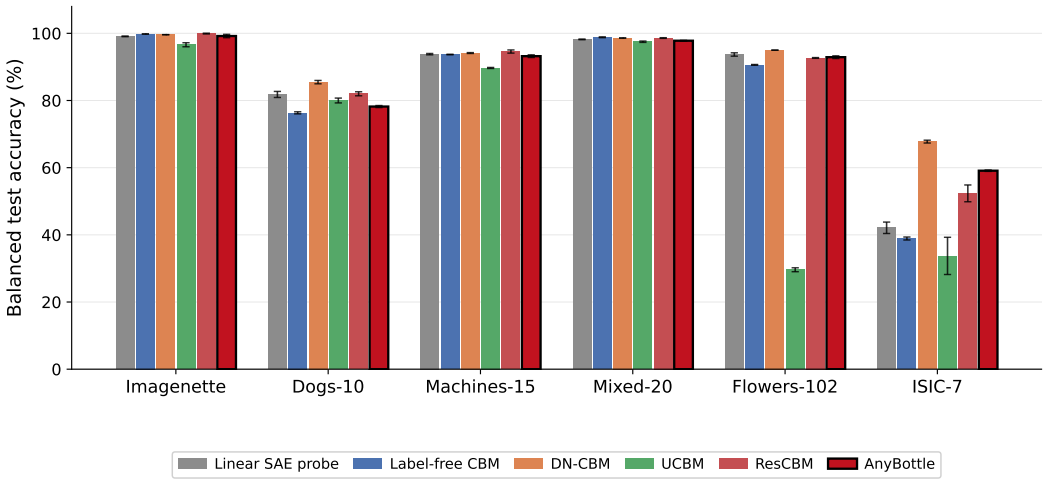


Figure 6: Raw accuracy of CBM baselines across vision datasets.

E.1 RAW ACCURACY ACROSS VISION DATASETS

Fig. 6 reports raw balanced test accuracy for every method on every dataset, as a supplement to the teacher gap Δ reported in the main text (Fig. 2). Because each method uses its own backbone, these numbers are not directly comparable across methods: a higher bar does not imply a better bottleneck, since it may instead reflect a stronger or more task-suited backbone. We include this figure only to show that no method's raw accuracy is degenerate in absolute terms; the teacher gap Δ, which normalizes for each method's own backbone, remains the primary accuracy metric throughout the paper. Accuracies are broadly comparable across methods on most datasets, with the exception of ISIC-7, where the spread across methods is largest.

E.2 PATCH-LEVEL CONCEPT CONSISTENCY VERSUS THE LINEAR SAE PROBE

Fig. 7 extends the image-level C^2 comparison in the main text (Fig. 3) to the patch (region) level. This comparison is only possible against the linear SAE probe, not against the four CBM baselines: ANYBOTTLE and the linear probe both operate on the same underlying SAE, whose concepts are grounded in individual patches, so patch-level activating evidence exists for every concept in the dictionary regardless of whether it was selected. The four CBM baselines produce only a single scalar

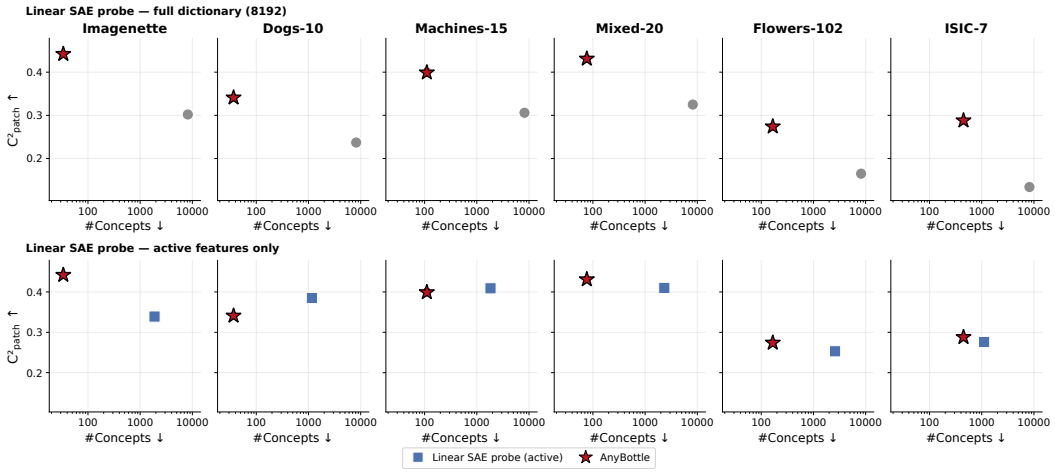


Figure 7: Patch-level concept consistency (C^2_{patch}) versus number of concepts, across six datasets. *Top row*: Linear SAE probe evaluated at its full dictionary size (8192) compared to AnyBottle. *Bottom row*: the same comparison restricted to the Linear SAE probe's active features only, shown at a finer x-axis scale. AnyBottle achieves comparable or higher C^2_{patch} at substantially fewer concepts.

activation per image per concept, with no sub-image localization, so no patch-level consistency score can be computed for them.

Top row: against the full, unselected dictionary (8192 features), ANYBOTTLE achieves higher C^2_{patch} than the linear probe on every dataset, while using two to three orders of magnitude fewer concepts. Bottom row: restricting the linear probe to only its active features (Sec. G.3) narrows the concept-count gap substantially, but ANYBOTTLE still matches or exceeds the active-feature probe's consistency on four of six datasets (Imagenette, Mixed-20, Flowers-102, ISIC-7); Dogs-10 is the one dataset where the active-feature probe is clearly more consistent, and Machines-15 shows the two within each other's error bars. This indicates that ANYBOTTLE's compactness is not simply discarding low-consistency concepts that an unconstrained active-feature baseline would also avoid, selection recovers comparable or better concept quality at a small fraction of the concept count on most, though not all, datasets.

E.3 PER-DATASET SELECTION STRATEGY ABLATION

Tab. 3 reports the per-dataset values underlying the averaged ablation in Tab. 1, showing concept set size K, C^2_{patch} consistency and teacher accuracy gap Δ. We ablate the different selection components: prefiltering (PF), localization (Loc), and candidate least squares scoring (LS). C^2_{patch} ordering is largely consistent across datasets: the full method has the highest consistency, and removing components generally reduces consistency. Concept count K generally follows the same ordering: removing successive parts generally reduces compactness of the final concept selection, as less suited concepts are selected. On Mixed-20 and Dogs-10 the no-localization configuration selects a concept pool close in size to the full method's, despite the drop in consistency and in accuracy gap. When only considering the teacher gap Δ, there is no configuration that always outperforms over all datasets. While the full selection performs best on average, other configurations can perform slightly better on individual datasets. Overall, the detailed per-dataset analysis confirms the findings from Tab. 1. While the full selection mechanism is not unanimously better in all metrics for all datasets, removing individual components lowers the quality of the resulting concept selection in one or more aspects.

E.4 CONTINUOUS AND DISCRETE BOTTLENECKS

ANYBOTTLE Supports Both Continuous and Discrete Bottlenecks. Tab. 4 (left) compares discrete (binary) and continuous (percentile-normalised) concept representations. Both are natively supported by ANYBOTTLE's selection procedure, and the choice can be made per deployment:

Table 3: Per-dataset results underlying the averaged ablation in Table Tab. 1 (main text). Ablations of least-squares scoring (LS), localized candidate search (Loc), and prefiltering (PF). Δ = balanced test accuracy − balanced teacher accuracy. C^2_{patch} is the patch-level concept consistency. Mean±std over 3 seeds.

Dataset	LS	Loc	PF	K ↓	C^2_{patch} ↑	Δ ↑ (%)
Imagenette	✓	✓	✓	$34.0_{\pm 2.0}$	$0.442_{\pm 0.005}$	$+0.3_{\pm 0.6}$
	✓	✓	–	$47.3_{\pm 3.5}$	$0.402_{\pm 0.008}$	$-0.9_{\pm 0.4}$
	✓	–	–	$87.3_{\pm 1.5}$	$0.348_{\pm 0.003}$	$+0.4$
	–	–	–	$184.7_{\pm 22.8}$	$0.269_{\pm 0.015}$	$+0.2$
Dogs-10	✓	✓	✓	$37.0_{\pm 1.0}$	$0.341_{\pm 0.002}$	$-1.8_{\pm 0.3}$
	✓	✓	–	$49.0_{\pm 0.0}$	$0.345_{\pm 0.000}$	$-2.0_{\pm 0.2}$
	✓	–	–	$38.3_{\pm 1.2}$	$0.331_{\pm 0.000}$	-2.3
	–	–	–	$128.7_{\pm 20.6}$	$0.280_{\pm 0.011}$	-1.4
Machines-15	✓	✓	✓	$110.0_{\pm 14.0}$	$0.399_{\pm 0.009}$	$+0.3_{\pm 0.4}$
	✓	✓	–	$219.0_{\pm 80.9}$	$0.377_{\pm 0.021}$	$+0.8_{\pm 0.4}$
	✓	–	–	$167.3_{\pm 40.3}$	$0.344_{\pm 0.016}$	$+0.5$
	–	–	–	$265.0_{\pm 10.6}$	$0.308_{\pm 0.006}$	$+0.4$
Mixed-20	✓	✓	✓	$76.0_{\pm 3.0}$	$0.431_{\pm 0.006}$	$-0.3_{\pm 0.1}$
	✓	✓	–	$137.7_{\pm 2.1}$	$0.423_{\pm 0.000}$	$+0.1_{\pm 0.1}$
	✓	–	–	$78.0_{\pm 3.5}$	$0.408_{\pm 0.005}$	-0.6
	–	–	–	$150.0_{\pm 25.2}$	$0.335_{\pm 0.012}$	-0.9
Flowers-102	✓	✓	✓	$166.0_{\pm 10.0}$	$0.274_{\pm 0.006}$	$+1.1_{\pm 0.4}$
	✓	✓	–	$156.0_{\pm 9.2}$	$0.270_{\pm 0.003}$	$+0.7_{\pm 0.4}$
	✓	–	–	$291.0_{\pm 4.4}$	$0.245_{\pm 0.000}$	$+0.4$
	–	–	–	$230.7_{\pm 14.6}$	$0.235_{\pm 0.009}$	$+1.0$
ISIC-7	✓	✓	✓	$447.0_{\pm 30.0}$	$0.288_{\pm 0.001}$	$-2.2_{\pm 0.2}$
	✓	✓	–	$385.3_{\pm 67.8}$	$0.254_{\pm 0.000}$	$-3.9_{\pm 0.6}$
	✓	–	–	$481.0_{\pm 12.5}$	$0.205_{\pm 0.001}$	-6.0
	–	–	–	$428.3_{\pm 90.4}$	$0.208_{\pm 0.006}$	-11.7

discrete bottlenecks give the strictly present/absent semantics that much of the CBM literature builds on, useful for symbolic integration or settings where a binary interface is preferred, while continuous bottlenecks use fewer concepts on every dataset and match or improve both the teacher-accuracy gap and patch-level consistency on most of them. We use continuous representations by default, but the discrete variant remains a valid option within the same framework.

E.5 Task-Aware SAE Fine-Tuning

Task-Aware SAE Fine-Tuning Helps Most Where the Pretrained Dictionary Is Mismatched. Tab. 4 (right) compares the CC12M-pretrained SAE against the same SAE fine-tuned on-task via a KL loss to the teacher. On Imagenette, Machines-15, and Flowers-102, fine-tuning reduces concept count and improves consistency together; on Dogs-10 and Mixed-20, where the class differences are coarse and pretrained dictionary is already well matched, it offers less clear benefit. ISIC-7 is the clearest case: fine-tuning substantially closes the accuracy gap and improves consistency, at the cost of a larger concept budget, consistent with the pretrained dictionary being poorly matched to dermoscopic imagery. Across all six datasets, fine-tuning does not come at a systematic cost to concept consistency, indicating it adapts which features are selected.

Table 4: Ablation studies on concept representation (left) and SAE fine-tuning (right). **Left**: effect of concept representation with a Task-aware: discrete vs continuous concept scores. **Right**: effect of SAE fine-tuning with continuous concepts: task-agnostic pretrained vs. task-aware finetuned. Δ = balanced test accuracy − balanced teacher accuracy; C^2_{patch} is the patch-level concept consistency. Mean±std over 3 seeds.

Dataset	Config	#Concepts ↓	C^2_{patch} ↑	Δ ↑ (%)
Imagenette	Discrete	90.0±9.0	0.38±0.012	+0.1±0.4
	Continuous	34.0±2.0	0.442±0.005	+0.3±0.6
Dogs-10	Discrete	89.0±3.0	0.317±0.004	+0.3±0.8
	Continuous	37.0±1.0	0.341±0.002	−1.8±0.3
Machines-15	Discrete	141.0±38.0	0.405±0.015	−0.9±1.1
	Continuous	110.0±14.0	0.399±0.009	+0.3±0.4
Mixed-20	Discrete	79.0±5.0	0.443±0.006	−1.2±0.3
	Continuous	76.0±3.0	0.431±0.006	−0.3±0.1
Flowers-102	Discrete	335.0±24.0	0.25±0.001	+0.7±0.2
	Continuous	166.0±10.0	0.274±0.006	+1.1±0.4
ISIC-7	Discrete	465.0±55.0	0.285±0.003	−4.6±2.1
	Continuous	447.0±30.0	0.288±0.001	−2.2±0.2

Dataset	Config	#Concepts ↓	C^2_{patch} ↑	Δ ↑ (%)
Imagenette	Pretrained	81.0±1.0	0.306±0.0	−0.1±0.3
	Task-aware	34.0±2.0	0.442±0.005	+0.3±0.6
Dogs-10	Pretrained	36.0±9.0	0.388±0.029	−1.4±0.9
	Task-aware	37.0±1.0	0.341±0.002	−1.8±0.3
Machines-15	Pretrained	182.0±6.0	0.355±0.003	+0.4±0.5
	Task-aware	110.0±14.0	0.399±0.009	+0.3±0.4
Mixed-20	Pretrained	58.0±0.0	0.464±0.0	−0.5±0.2
	Task-aware	76.0±3.0	0.431±0.006	−0.3±0.1
Flowers-102	Pretrained	404.0±138.0	0.24±0.002	−1.9±0.6
	Task-aware	166.0±10.0	0.274±0.006	+1.1±0.4
ISIC-7	Pretrained	358.0±27.0	0.263±0.001	−12.4±0.4
	Task-aware	447.0±30.0	0.288±0.001	−2.2±0.2

Table 5: Effect of the stopping tolerance δ_{acc} on AnyBottle, across all six vision datasets. Δ = Accuracy − Teacher (balanced accuracy). C^2_{patch} is the patch-level (localised) consistency score; C^2_{img} is the image-level score. Mean±std over 3 seeds.

Dataset	δ_{acc}	#Concepts	C^2_{patch} ↑	C^2_{img} ↑	Δ_{val} (%)	Δ_{test} (%)
Imagenette	0.0	33.7±2.1	0.442±0.005	0.324±0.003	+0.09	+0.33
	0.1	33.7±2.1	0.442±0.005	0.324±0.003	+0.09	+0.33
	0.2	25.0±4.6	0.453±0.007	0.335±0.007	−0.19	−0.27
Dogs-10	0.0	36.7±0.6	0.341±0.002	0.105±0.001	+0.05	−1.80
	0.1	36.7±0.6	0.341±0.002	0.105±0.001	+0.05	−1.80
	0.2	36.7±0.6	0.341±0.002	0.105±0.001	+0.05	−1.80
Machines-15	0.0	143.0±66.2	0.391±0.033	0.233±0.028	+0.10	+0.27
	0.1	110.3±13.7	0.399±0.009	0.234±0.009	+0.48	+0.31
	0.2	89.0±9.5	0.414±0.010	0.254±0.011	−0.06	−0.57
Mixed-20	0.0	82.0±4.0	0.406±0.009	0.303±0.008	+0.19	−0.60
	0.1	76.3±3.1	0.431±0.006	0.325±0.006	+0.08	−0.30
	0.2	71.3±3.8	0.428±0.007	0.321±0.005	+0.00	−0.63
Flowers-102	0.0	197.3±13.3	0.265±0.002	0.152±0.002	+1.37	+1.59
	0.1	166.3±10.1	0.274±0.006	0.158±0.003	+0.78	+1.10
	0.2	188.0±3.5	0.267±0.001	0.154±0.001	+0.91	+1.38
ISIC-7	0.0	447.0±29.6	0.288±0.001	0.088±0.000	−3.08	−2.17
	0.1	447.0±29.6	0.288±0.001	0.088±0.000	−3.08	−2.17
	0.2	404.3±91.0	0.290±0.004	0.088±0.001	−8.75	−2.53

E.6 Influence of the Accuracy-Gap Stopping Tolerance

Tab. 5 reports the effect of the stopping tolerance δ_{acc}, which controls how much validation accuracy gap to the teacher is tolerated before selection halts and concepts are fixed. Tightening δ_{acc} from 0.0 toward 0.2 generally reduces the size of the selected concept sets, at a dataset-dependent cost to Δ_{val}. On the other hand, in several scenarios changing the tolerance has no effect, since selection already halts on a different criterion first. Because model selection is driven by Δ_{val} rather than Δ_{test}, a gap between the two is diagnostic of the stopping criterion fitting validation-specific noise rather than true generalization: on ISIC-7, attempting to match the validation performance of the teacher comes at a cost in test performance instead. This is most likely due to ISIC-7's weak, noisy teacher and small, unbalanced validation split (Sec. 4). This suggests δ_{acc} is better treated as a budget-accuracy trade-off knob than as a target to tune aggressively: on well-behaved datasets it trades accuracy for compactness predictably, but where the teacher is unreliable, the validation signal can turn misleading exactly as the tolerance tightens.

Table 6: Across domains and teacher paradigms, ANYBOTTLE matches near-teacher accuracy with orders of magnitude fewer concepts than a linear probe over the full SAE dictionary. We report concept count K, concepts needed to reach 90% and 95% of the student accuracy (C@90/95), AutoInterp score (AIS), and balanced accuracy gap to teacher (Δ) and in absolute terms (Acc).

Dataset	Method	#Concepts ↓	C@90 ↓	C@95 ↓	AIS ↑	Δ (%) ↑	Acc (%) ↑
AG News	Linear SAE probe	31477	–	–	$0.327_{\pm 0.009}$	$-0.12_{\pm 0.38}$	$90.42_{\pm 0.38}$
	mlp-clf (gemma-2-2B)	$93.7_{\pm 6.4}$	$63_{\pm 1}$	$64_{\pm 0}$	$0.469_{\pm 0.006}$	$-0.18_{\pm 0.25}$	$90.36_{\pm 0.25}$
	Linear SAE probe	9710	–	–	$0.157_{\pm 0.001}$	$+18.57_{\pm 0.98}$	$88.25_{\pm 0.98}$
	verb-clf (gemma-3-1B-it)	$30.0_{\pm 5.0}$	$27_{\pm 10}$	$30_{\pm 6}$	$0.094_{\pm 0.020}$	$+0.82_{\pm 1.89}$	$70.50_{\pm 1.89}$
PubMed-20k	Linear SAE probe	25893	–	–	$0.453_{\pm 0.000}$	$-2.47_{\pm 0.87}$	$73.58_{\pm 0.87}$
	mlp-clf (gemma-2-2B)	$100.0_{\pm 2.6}$	$47_{\pm 10}$	$62_{\pm 1}$	$0.526_{\pm 0.004}$	$-0.02_{\pm 0.23}$	$76.03_{\pm 0.23}$
	Linear SAE probe	7853	–	–	$0.087_{\pm 0.000}$	$+43.04_{\pm 1.66}$	$70.92_{\pm 1.66}$
	verb-clf (gemma-3-1B-it)	$23.3_{\pm 2.5}$	$20_{\pm 4}$	$23_{\pm 3}$	$0.146_{\pm 0.012}$	$+15.93_{\pm 2.04}$	$43.81_{\pm 2.04}$

E.7 GENERALIZATION TO TEXT IN DETAIL

Tab. 6 extends Tab. 2 (main text) with C@90 and C@95, showing how much of each text dataset's concept budget K is needed to reach near-full accuracy. The pattern differs by teacher paradigm. For the mlp-clf teacher, ANYBOTTLE con reach most of its maximum accuracy with a smaller fraction of the concepts. For the verb-clf teacher, in contrast, nearly the entire (much smaller) concept pool is needed throughout on both datasets.

F QUALITATIVE EXAMPLES

This section extends the qualitative results in the main text (Sec. 4) to all six vision datasets, and adds a further qualitative view of concept consistency not otherwise shown.

Selection round by round, all datasets. Fig. 8 extends Fig. 4 (main text, Imagenette only) to all six datasets, showing the validation accuracy trajectory and three representative rounds' concepts and blind-spots per dataset. Note: Machines-15, Mixed-20, Flowers-102, and ISIC-7 use the class-focused jump-start round described in Sec. 3.2, so their earliest round reflects this heuristic seed. Round numbers here are counted from the start of residual-driven selection that follows. The pattern from the main text holds throughout: early rounds close large, class-defining blind-spots, while later rounds refine coverage with diminishing accuracy gain. ISIC-7 stands out for needing far more rounds (314) than any other dataset to plateau, consistent with the results discussed in Sec. 4.

Adaptive inference, all datasets. Fig. 9 and Fig. 10 extend Fig. 5 to all datasets, following the same selection and layout: for each dataset we show two correctly classified test images, one which needs a large number of firing concepts to reach 90% confidence and which needs only a few. The same pattern recurs across datasets: easy images converge within a handful of concepts, while harder images draw on a much larger share of the discovered budget.

Concept consistency, qualitatively. Fig. 11 and Fig. 12 complement the quantitative C^2 score with a direct visual check: five randomly drawn concepts per dataset, each shown with its ten most strongly activating test images. Most concepts are immediately recognizable and hold up across all ten images, not just the first one or two, e.g. Dogs-10's *"erect pointed animal ear at top of head"* and Flowers-102's *"round funnel/trumpet-shaped flower corolla"* pick out a consistent visual pattern throughout. A few concepts are visibly looser, most notably on ISIC-7, where patterns like *"thin dark body hairs on pale skin"* and *"pale normal skin with fine light micro-texture"* vary considerably in shape and framing across their top-activating images, and on Machines-15, where the broader *"vehicles / means of transportation"* concept spans several vehicle types rather than one specific pattern. This mirrors the quantitative results, where ISIC-7 shows the lowest C^2 scores of the six datasets (Fig. 3), consistent with its more specialized, harder-to-disentangle domain and the less adapted SAE concept pool.

These figures also highlight potential limitations of the used automated naming rather than of the concepts themselves: on Imagenette, the concept named *"gas-station fuel nozzle / car fuel-filler inlet"* fires on several images whose gripped, handle-like protrusions more plausibly belong to a chainsaw's

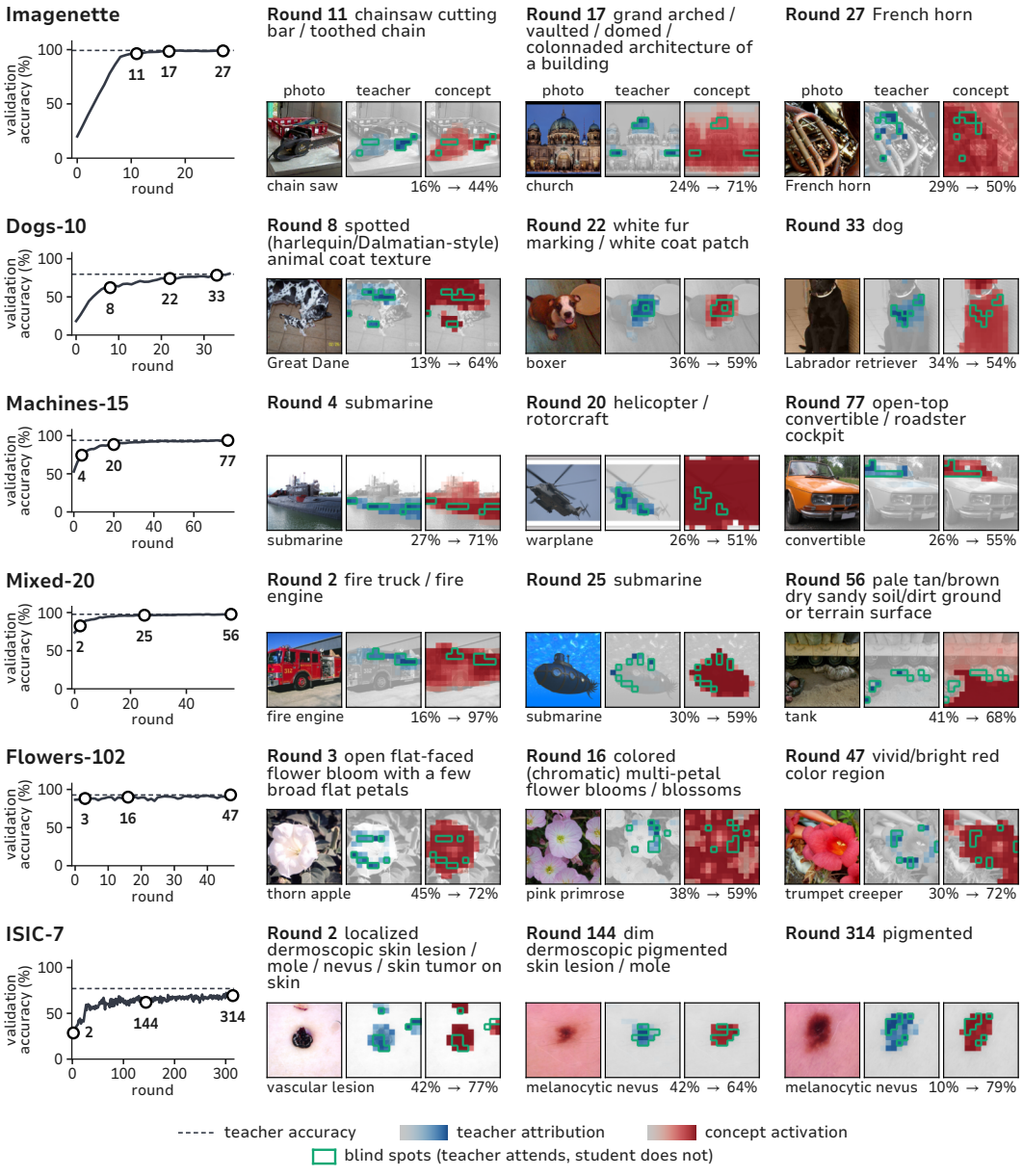


Figure 8: **Concept selection round by round on all six vision datasets.** For each dataset, we show the validation accuracy of the current predictor $f_{\mathcal{S}}$ (left) and the concepts and blind spots for three rounds (right). The dashed line is the teacher's validation accuracy. For each round, we show one training image from its focus set on which the added concept fires, with the teacher's attribution and the concept's activation. Green outlines mark the round's blind spots, i.e. the tokens the teacher attends to, but $f_{\mathcal{S}}$ does not. Below each image are its class and the probability $f_{\mathcal{S}}$ assigns to it before $\rightarrow$ after the round.

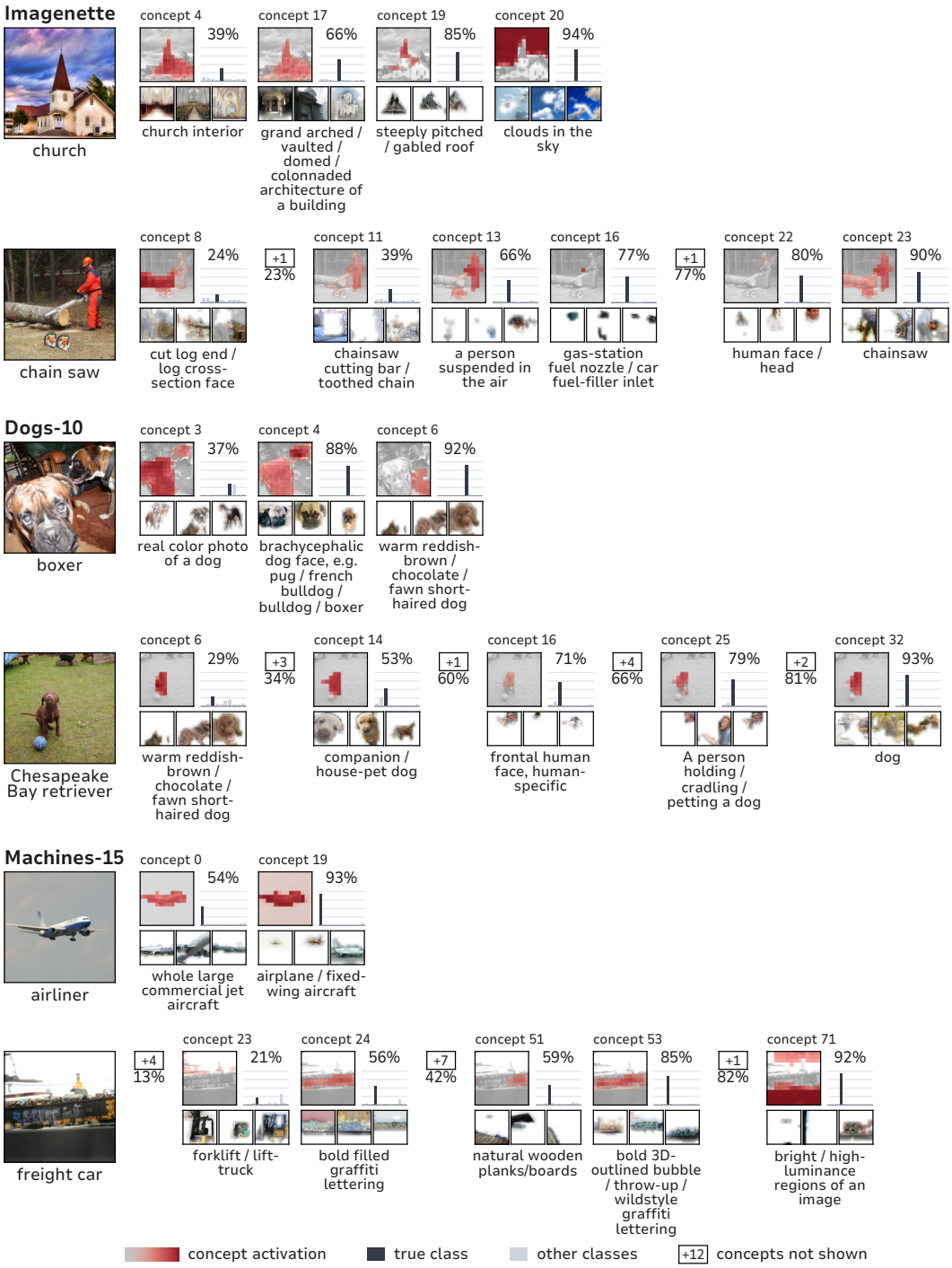


Figure 9: **Adaptive inference on the other five datasets.** Two test images per dataset, one per row, as Fig. 5 shows them for Mixed-20. Imagenette, Dogs-10 and Machines-15 here; Flowers-102 and ISIC-7 in Fig. 10.

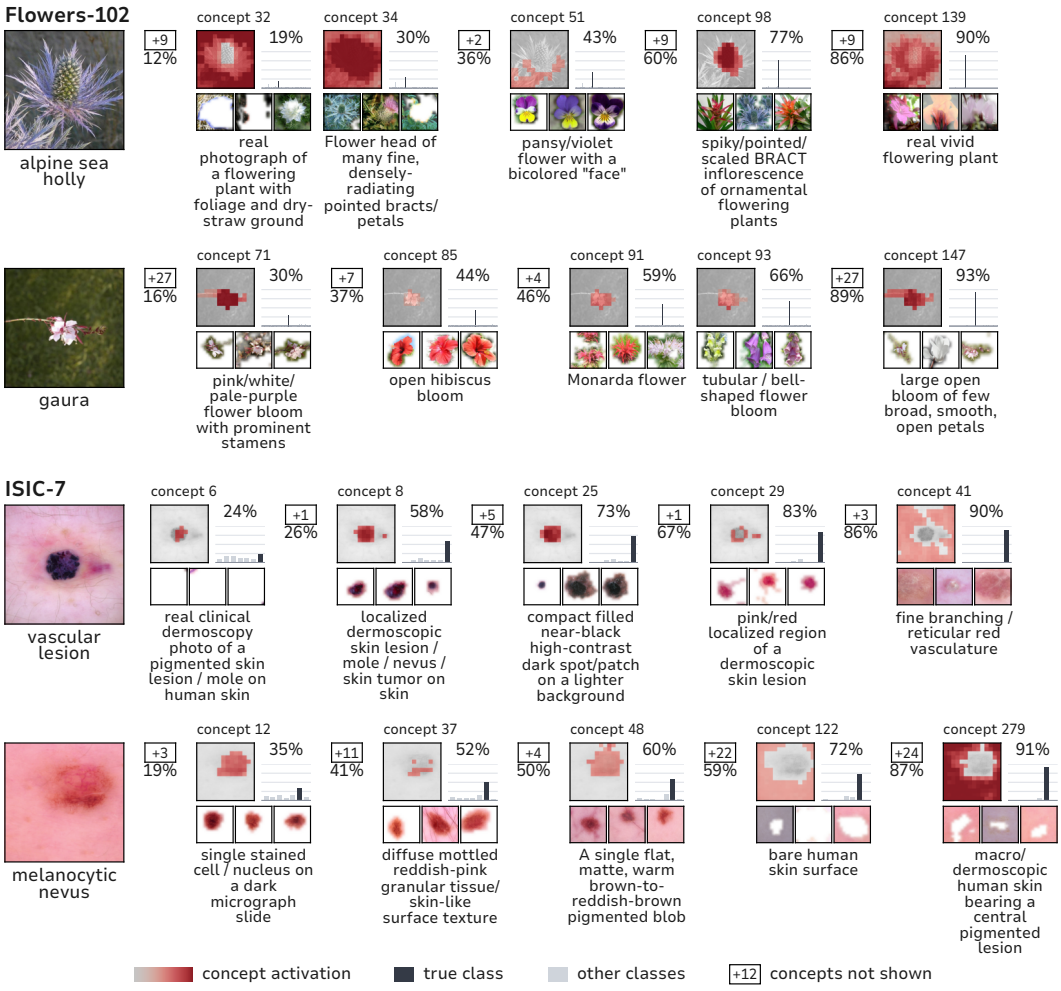


Figure 10: **Adaptive inference on Flowers-102 and ISIC-7**, continued from Fig. 9.

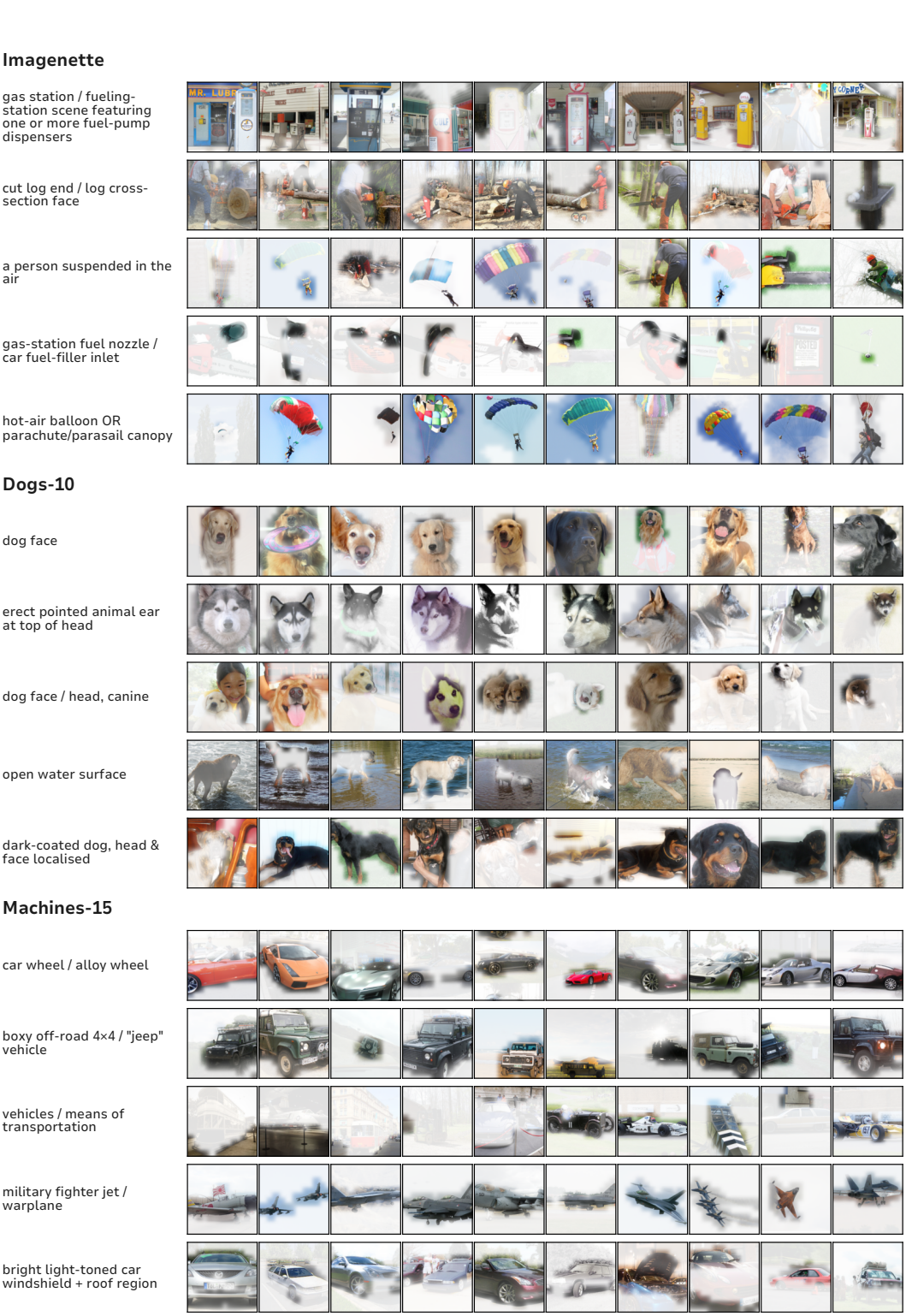


Figure 11: **Concept examples.** Five concepts per dataset, drawn at random, one per row, each with the ten test images it fires on most strongly. Each image keeps its colour where the concept fires, scaled to its own strongest patch, and fades elsewhere. Imagenette, Dogs-10 and Machines-15 here; Mixed-20, Flowers-102 and ISIC-7 in Fig. 12.

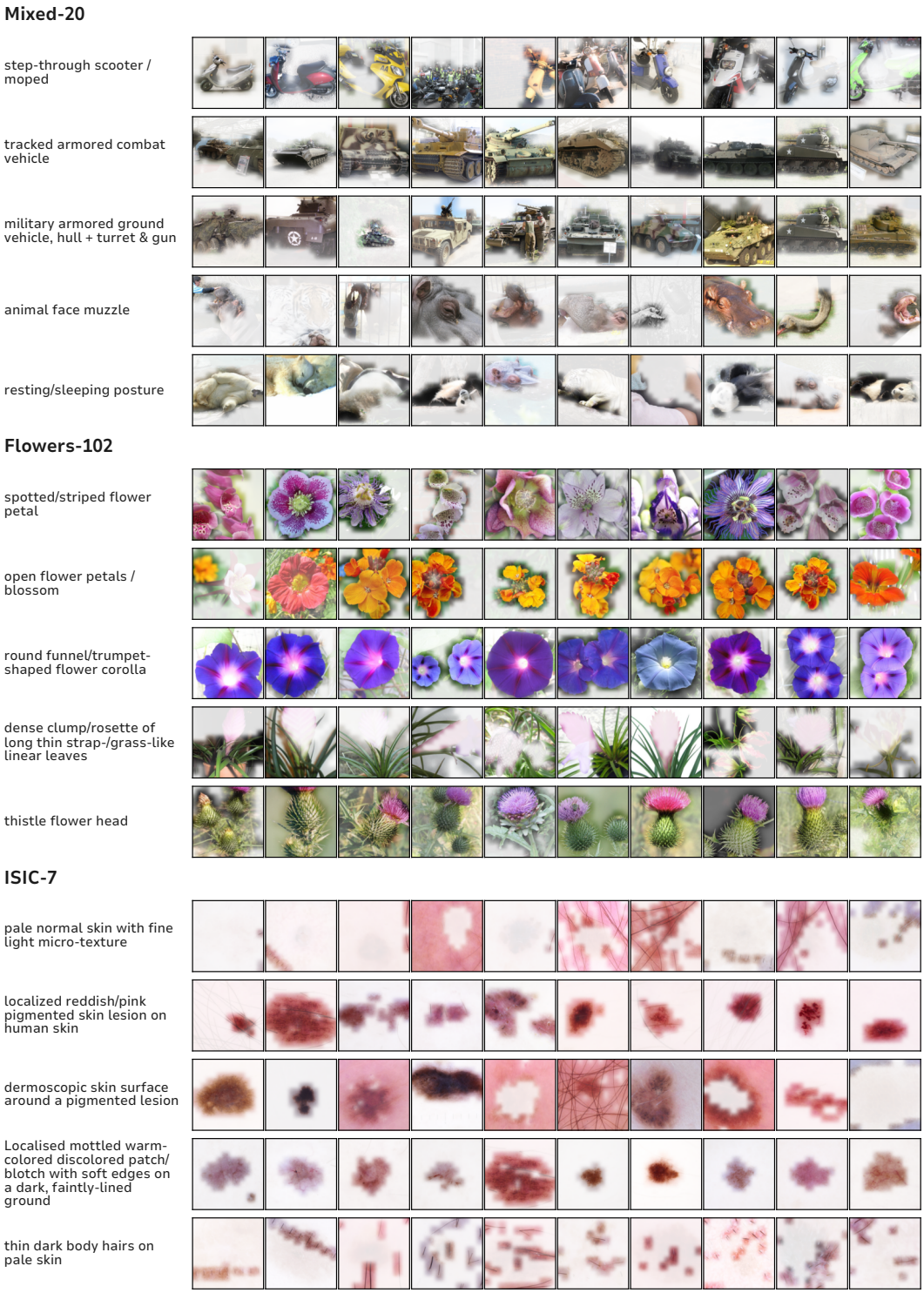


Figure 12: **Concept examples**, continued from Fig. 11: Mixed-20, Flowers-102 and ISIC-7.

Table 7: ImageNet-1k classes used in our three curated subsets.

Subset	Classes
Dogs (10)	Golden retriever, Labrador retriever, Chesapeake Bay retriever, German shepherd, Rottweiler, Doberman, Boxer, Great Dane, Siberian husky, Eskimo dog
Machines (15)	Sports car, Station wagon, Convertible, Jeep, Pickup, Garbage truck, Trailer truck, Steam locomotive, Streetcar, Freight car, Airliner, Warplane, Speedboat, Submarine, Forklift
Mixed (20)	*Animals:* Lion, Tiger, African elephant, Zebra, Giant panda, Polar bear, Gorilla, Flamingo, Ostrich, Hippopotamus; *Machines:* Sports car, Warplane, Submarine, Steam locomotive, Forklift, Ambulance, Tank, Space shuttle, Fire engine, Motor scooter

trigger housing than a fuel nozzle, a visual confusion likely inherited from Imagenette's own chain saw class. The concept's activating region is consistent, a handled mechanical fixture, but MAIA's single-pass name is more specific than the pattern it describes actually supports. We discuss such naming-accuracy risk in Sec. 5: compactness makes such cases tractable to catch on inspection, as here, but does not by itself guarantee the automatically generated name is correct, underscoring the value of human review before deployment.

G Details on Experimental Setup

G.1 Datasets

Vision. We evaluate on six image classification datasets chosen to span a range of task types: *Imagenette*, a coarse-grained 10-class subset of ImageNet; *Dogs-10*, a fine-grained 10-breed subset requiring within-category discrimination; *Machines-15*, a coarse multi-class task over mechanical object categories; *Mixed-20*, a heterogeneous 20-class task combining categories from multiple sources; *Flowers-102*, a large-vocabulary, fine-grained 102-class botanical task; and *ISIC-7*, a 7-class dermoscopic skin lesion classification task representing a specialised scientific domain with a comparatively weak teacher. This selection deliberately covers coarse and fine-grained discrimination, small and large label vocabularies, and both generic and domain-specific imagery, to test whether concept selection behaves consistently across task difficulty rather than only on standard benchmarks.

ImageNet subsets. To study how the difficulty of the discrimination task affects the learned bottleneck, we curate three subsets of ImageNet-1k (Deng et al., 2009). **ImageNet-Dogs** (10 classes) is a set of dog breeds. It includes groups of visually similar breeds (three retrievers, plus the Siberian husky and Eskimo dog) so that the task cannot be solved from coarse cues alone. **ImageNet-Machines** (15 classes) contains vehicles and machines from road, rail, water, air and industrial settings; while broader in scope than the dog breeds, it still groups several visually similar classes (e.g. sports car, convertible, and pickup, or the various trucks), giving it comparable fine-grained discrimination in parts of its label set. **ImageNet-Mixed** (20 classes) is a control condition that combines ten distinct animals with ten distinct machines. Table 7 lists the classes. For each subset we sample 700 training images per class and split them 90/10 into train and validation using a fixed seed (42). The official ImageNet validation images of the selected classes serve as the held-out test set.

Text. We evaluate on two text classification datasets with different tasks and domains: *AG News* (4-class topic classification, general-domain) and *PubMed-20k* (5-class sentence role classification in biomedical abstracts, specialised domain). As in the vision setting, this spans a general-domain and a specialised, domain-specific text classification task.

G.2 Model and Training Details

G.2.1 Vision Backbones and Teachers

AnyBottle's vision backbone is CLIP-DINOiser (Wysoczańska et al., 2024): a CLIP ViT-B/16 model (`laion2b_s34b_b88k` weights) combined with DINO-style correlation pooling. This is a different backbone from the plain CLIP ViT-B/16 used by Label-free CBM and DN-CBM. It shares its CLIP weights (`laion2b`) with Res-CBM's substituted backbone, but the two backbones are otherwise different.

For every baseline and for AnyBottle, the teacher is a two-layer MLP (Linear($d \to 512$)–ReLU–Dropout(0.1)–Linear($512 \to C$)), trained on that method's own frozen backbone features via Adam (lr 10^{-3}, weight decay 10^{-4}, batch 256, 40 epochs), with model selection by balanced accuracy on a held-out validation split. Each baseline's teacher is trained separately, on that baseline's own backbone, so reported teacher–student gaps stay within one embedding space.

Label-free CBM (Oikarinen et al., 2023). Backbone: CLIP ViT-B/16 (`openai` weights), used both as the feature encoder and as the concept-scoring model, following the paper's `clip_ViT-B/16` configuration. Concepts are generated once with GPT-4o-mini (the paper's original `text-davinci-002` is deprecated) using the paper's three prompt templates, then passed through the paper's five filtering steps (length, cosine similarity to the class name, cosine similarity to other concepts, mean CLIP activation, and post-projection similarity) unchanged. A linear projection layer is fit to maximize the CLIP-Dissect $\cos^3$ similarity objective, and the final sparse layer is fit with GLM-SAGA elastic-net regression ($\alpha = 0.99$), tuned toward the paper's target of 25–35 non-zero weights per class.

Res-CBM (Shang et al., 2024). Backbone: CLIP ViT-B/16 (`laion2b` weights), chosen to keep this baseline in the same CLIP family as the rest of our comparisons rather than the original paper's RN50/ViT-L-14, so absolute numbers are not directly comparable to the published results. The candidate concept bank is built from Visual Genome object, attribute, and relationship terms, intersected with a CAV concept library and ConceptNet neighbours of the class names, following the paper's construction. Concepts are added incrementally over D discovery rounds (D set per dataset, following the paper's $\min(2C, 64)$ heuristic), each round initializing a new concept vector, training it, and mapping it to its nearest candidate via the paper's rank-based similarity loss.

DN-CBM (Rao et al., 2024). Backbone: CLIP ViT-B/16 (`openai` weights), matching Label-free CBM's backbone so the two CLIP-based baselines are directly comparable to each other. Concepts come from the authors' released sparse autoencoder checkpoint (512→4096, expansion factor 8, trained on CC3M), used frozen and forward-only, with concept names taken from the authors' released CLIP-Dissect-style name assignments. A single linear probe is trained on the resulting 4096-dimensional sparse code with an ℓ_1 penalty, following the paper's ViT-B/16 configuration; because the SAE dictionary is large relative to the test sets, explained variance for this baseline is close to 1.0 and should be read as the concept space spanning the teacher's output space rather than as evidence of a tight fit.

UCBM (Schrodi et al., 2025). Backbone: a frozen, ImageNet-supervised ResNet-50, used up to (excluding) global average pooling, following the paper's primary configuration; on our four ImageNet-derived datasets this backbone is in-distribution, so the teacher is near ceiling and the resulting gap is correspondingly small, while Flowers-102 and ISIC-7 are genuinely out-of-distribution for it. Concepts are discovered once per dataset via non-negative matrix factorization over cropped, class-conditioned activations, with the number of concepts per class set proportionally to the number of classes, following the paper's default factor. The classifier applies a cosine-similarity projection onto the discovered dictionary, an input-dependent gate that suppresses low-similarity concepts, concept dropout, and a sparse linear head, trained with the paper's elastic-net objective.

Comparability. By default, each baseline uses a different backbone and concept source, and each has its own separately-trained teacher (above), so absolute accuracy is not comparable across baselines. For every baseline including AnyBottle, we report the teacher–student balanced accuracy gap and concept-count metrics within that baseline's own embedding space, under a common evaluation

protocol: identical datasets, train/val/test splits, and three seeds (42, 123, 7) across all methods, with model selection by held-out validation balanced accuracy throughout (the original papers variously select on the training set, on a train/val split, or on the test set; we standardize all baselines, and ANYBOTTLE, to a genuine held-out validation split for a symmetric comparison). We follow each paper's method code and hyperparameters as closely as possible.

G.2.2 VISION SAE: ARCHITECTURE AND FINE-TUNING

ANYBOTTLE's concept pool is a Matryoshka BatchTopK SAE with activation dimension 512 and dictionary size 8192. It uses TopK activation with $k = 12$ active latents per token, and has 6 nested Matryoshka groups. The SAE is trained on CLIP-DINOiser patch tokens over a CC12M corpus, with learning rate 10^{-4} for 118,935 steps. ANYBOTTLE's selection procedure does not use the Matryoshka nesting structure and works with any SAE that exposes per-token activations over a fixed dictionary. The SAE's TopK activation ($k = 12$) affects only its own reconstruction; it does not restrict ANYBOTTLE's initial candidate pool, which always spans all 8192 dictionary atoms.

We fine-tune one SAE checkpoint per dataset, in each case starting from the same CC12M-pretrained checkpoint. As detailed in Sec. 3.1, we use the KL divergence of the teacher output distribution on standard embeddings and reconstructed SAE embeddings as objective for this fine-tuning.

G.2.3 TEXT BACKBONES AND TEACHERS

We consider two teacher paradigms. The *MLP teacher* (mlp-clf) mirrors the vision setup: a frozen, pretrained language model (Gemma-2-2B) is read via activation hooks, and a two-layer MLP head with one hidden layer of 512 units is trained on the model's last-token hidden state to predict the class label. Only this head is trained; the backbone stays frozen. The *verbalizer teacher* (verb-clf) has no vision analogue: a frozen, instruction-tuned language model (Gemma-3-1B-it) is prompted directly, via a chat template, to name the class in natural language. Its prediction is read off the model's own next-token distribution, restricted to one single-token surface form per class (AG News: *World*, *Sports*, *Business*, *Technology*, the last standing in for the label *Sci/Tech*; PubMed: *Background*, *Objective*, *Methods*, *Results*, *Conclusion*) and renormalised over those tokens. Prompts are listed in Tab. 8.

Table 8: Verbalizer prompts (wrapped in the Gemma-3 chat template) and class-to-token mappings.

Dataset	Prompt	Class → token
AG News	`Article: "{text}"` `Which topic best fits this article: World, Sports, Business, or Technology? Answer with exactly one word.`	World, Sports, Business, Sci/Tech → Technology
PubMed 20k RCT	`Sentence from a medical research abstract: "{text}"` `Does this sentence state the study's background, objective, methods, results, or conclusion? Answer with exactly one word.`	Background, Objective, Methods, Results, Conclusion

G.2.4 TEXT SAE: ARCHITECTURE PER TEACHER

For the MLP-teacher (Gemma-2-2B), concepts come from a SAEBench-released TopK sparse autoencoder, dictionary size 64,512, read at layer 19. For the verbalizer-teacher (Gemma-3-1B-it), concepts come from a Gemma Scope sparse autoencoder, dictionary size 16,384, read at layer 17. This layer is fixed by configuration for all datasets. As on vision, we fine-tune one SAE checkpoint per dataset from a shared pretrained starting point, here using the `teacher_kl` objective, again selected over the alternative fine-tuning modes by validation-set performance. A redundancy filter discards candidate concepts whose SAE decoder direction is too cosine-similar (threshold 0.6) to an already-selected concept due to the overly large SAE concept pool.

G.2.5 Linear Probe and Full-Dictionary Baselines

For both modalities, we compare against a linear probe trained on the full, unselected SAE dictionary, and one restricted to only the neurons active on the evaluation data.

G.2.6 Hyperparameters

Tab. 9 lists the remaining hyperparameters governing selection and training, for vision and text where they differ.

Table 9: Hyperparameters used throughout AnyBottle's selection and training pipeline. Where vision and text differ, both values are given; a single value applies to both. For text, G2 = Gemma-2-2B MLP-teacher runs, G3 = Gemma-3-1B-it verbalizer runs. Accuracy thresholds are in percentage points.

Stage	Parameter	Vision	Text
Backbone / Teacher	Teacher backbone	CLIP-DINOiser (ViT-B/16), CLS	G2: Gemma-2-2B, last-token resid. (layer 19) / G3: Gemma-3-1B-it (frozen verbalizer)
	Teacher head	MLP, hidden dim 512, dropout 0.1 (G3: none, frozen LM)	
	Teacher optimizer	Adam, lr 10^{-3}, weight decay 10^{-4} (G3: n/a)	
	Teacher batch size / epochs	256 / 40	full-batch / 60 (early stop, patience 12)
	Teacher selection metric	balanced val. accuracy	
SAE	Architecture	Matryoshka BatchTopK	G2: TopK (SAEBench) / G3: JumpReLU (Gemma Scope 2)
	Dictionary size M	8192	G2: 64,512 / G3: 16,384
	Sparsity	$k = 12$	G2: $k = 40$ / G3: target $L_0 \approx 60$
	Pretraining corpus	CC12M	G2: The Pile (400M tokens) / G3: Gemma 3 pretraining distribution + chat data (oasst1, LMSYS-Chat-1M)
SAE Fine-tuning	Objective	`teacher_kl_recon`	`teacher_kl`
	Fine-tuning teacher	patch-attention MLP (frozen)[b]	same teacher as discovery
	KL temperature / recon. weight	1.0 / 0.1	1.0 / 0
	Optimizer	Adam, lr 10^{-4}	
	Batch size / max epochs	64 / 40 (ISIC: 20)	32 / 20
	Early stopping	val. CE increase, patience 5, 200 warmup steps	best val. loss epoch
Rescaling	Percentile p	99	
	Clip value $\tau_{\max}$, ϵ	1.0, 10^{-6}	
Candidate Proposal	Focus set size k	200	G2: 800 (AG News) / 500 (PubMed); G3: 500
	Blind-spot token count b	10	G2: 6 (AG News) / 10 (PubMed); G3: 10
	Per-sample top candidates r	5 (10 for Flowers, ISIC)	30
	Min distinct-sample count $d_{\min}$	3	
	Shortlist size q	30	
	Activation threshold θ	0.01	
	Redundancy filter (cosine threshold)	disabled	0.6
	Class-focused jump-start round	enabled: Machines-15, Mixed-20, Flowers-102, ISIC-7	–
Candidate Scoring	Scoring ϵ (Eq. 2)	10^{-8}	
	Minimum score threshold $s_{\min}$	10^{-3} (immediate stop)	10^{-3} (3 consecutive rounds)
Retraining / Stopping	Student optimizer	lr 10^{-2}, weight decay 10^{-4}	
	Student batch / epochs per round	128 / 5 (64 / 3–10)[a]	32 / 5
	Accuracy gap tolerance δ_{acc}	0.1	0.0
	Validation plateau margin η	0.01	0.1
	EV plateau margin η_{EV}	0.001	
	Patience P	10 / 20[a]	30
	Max rounds $T_{\max}$	300 / 500[a]	200
Nested Dropout	Prefix sampling	uniform over $\{1, 2, 4, \ldots, 2^i\} \cup \{\lvert\mathcal{S}^\star\rvert\}$	
	Epochs / lr / patience	100 / 10^{-3} / 20	
Evaluation	Seeds	42, 123, 7 (data split fixed, seed 42)	

G.2.7 Agentic Concept Naming

We use the open-source OpenMAIA implementation (Camuñas et al., 2025) with Qwen3-VL-8B-Instruct as both the interpretability agent and the image-description model, served locally via vLLM (32k context). For each selected concept, MAIA runs its full agentic loop: in every round it writes Python experiments that are executed against our system, with access to the concept's activations and per-patch activation heatmaps on arbitrary images, its 15 top-activating images from the held-out probe set, and captioning. It can also synthesize and edit images with FLUX.1-dev and FLUX.1-Kontext-dev (4-bit quantized; 25 and 15 denoising steps, guidance 5.5 and 2.5) to test hypotheses causally. The loop ends when the agent returns a final description; after 25 rounds it is prompted to conclude. The agent outputs a short label, a description, and a self-reported confidence score. We provide prompts in Sec. G.4.

G.3 Metrics

Since each baseline uses a different backbone, concept source, and teacher, absolute accuracy is not comparable across methods. We instead report each method's own teacher student accuracy gap, explained variance, and concept count within its own embedding space, under an identical protocol: identical datasets, train/val/test splits, and three seeds across all methods.

Number of concepts. We report $K = |\mathcal{S}^\star|$, the size of the concept set returned by selection, as the primary measure of bottleneck compactness. For full dictionary SAE comparisons (the linear SAE

probe baseline), we report the number of active SAE features in place of K (see below). Enabled by the nested dropout head, ?method's final head f can predict at any prefix length $k \leq K$. After training, f is evaluated across a range of prefix lengths on both validation and test data, producing a balanced accuracy versus concept budget profile $\mathrm{acc}(k)$, where $\mathrm{acc}(\cdot)$ denotes balanced accuracy throughout, and $\mathrm{acc}(K)$ is reported as the model's overall accuracy. We define

$$\mathrm{C@}q = \min\left\{k : \mathrm{acc}(k) \geq \frac{q}{100} \cdot \mathrm{acc}(K)\right\}, \tag{3}$$

the smallest prefix length at which test balanced accuracy first reaches q percent of the student's own full-budget accuracy $\mathrm{acc}(K)$, for $q \in \{90, 95, 98\}$, characterising how much of the discovered budget is actually needed to reach near-ceiling performance, rather than how many concepts were discovered in total. Unlike C@90/95/98, K is not itself an accuracy milestone: it is the total selected budget, reported alongside C@q for reference.

Active SAE features. The full-dictionary linear SAE probe uses every SAE feature, so we report its number of *active* features in place of K. A feature counts as active if it fires on at least 5 examples of the held-out test split. It fires on an example if its max-pooled activation is above zero. We use the same active subset as the probe's reference set for the C^2 and AIS comparisons, and the AIS features are sampled from this subset, stratified by activation frequency. This subset is computed after the probe has been trained and is used only for reporting and evaluation: the probe is trained on, and evaluated with, the full dictionary.

Explained variance (EV). As a running diagnostic during selection, we track $\mathrm{EV}_S = 1 - \mathrm{var}(R)/\mathrm{var}(Y_T)$, the fraction of teacher output variance explained by the current concept set S. We report EV alongside K and accuracy where relevant, as an indicator of concept pool quality independent of the stopping criterion reached.

Balanced accuracy. As several datasets have substantial class imbalance, we report balanced accuracy, the average of per-class recall

$$\mathrm{BAcc} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \frac{\text{correctly predicted}_c}{\text{total}_c}. \tag{4}$$

Concept consistency (C^2). Accuracy alone does not indicate whether the selected concepts are individually coherent, i.e. whether each concept's high-activation examples share a consistent, describable pattern rather than firing on an arbitrary mix of inputs. We adopt the C^2 score (Parchami-Araghi et al., 2026) to quantify this: for each selected concept, C^2 measures the semantic consistency of the set of inputs (or input regions) that most strongly activate it, and we report the mean C^2 over the selected concepts. For AnyBottle, this is computed at the *patch* level, since each SAE concept's activating evidence is localized to specific image patches. The four CBM baselines we compare against instead produce a single scalar activation per image per concept, either a text-concept cosine similarity or a global-feature projection, so their C^2 is necessarily computed at the coarser *image* level; we report this distinction alongside each result and note that region-level and image-level C^2 are not directly comparable; the baseline scores serve to indicate whether a method's concepts are coherent *at the granularity it operates at*, not to rank fine-grained localization ability the baselines were never designed to provide. We additionally compare AnyBottle against two references on its own SAE pool: a linear probe over the full, unselected SAE dictionary (M concepts), and a linear probe restricted to only the subset of SAE neurons that are ever active on the evaluation data (*active* neurons), both trained under the same protocol as the student. Together with K, C^2 lets us distinguish genuine gains in concept quality from AnyBottle's selection procedure from gains that would follow from using any smaller, arbitrary subset of the dictionary.

AutoInterp score (AIS). For text, where concepts do not localize to spatial regions, we assess concept coherence following the automated interpretability protocol of (Bills et al., 2023; Paulo et al., 2025). For each concept, an LLM judge is shown the concept's top activating documents from the evaluation split and produces a single sentence description of the pattern. The same judge is then given only this description, together with a held out set of ten strongly activating and ten weakly or non activating documents, and rates per document how strongly it expects the concept to fire. The

AutoInterp score for a concept is the Pearson correlation between these predicted ratings and the concept's true activations on the same held out documents; a concept is excluded from the aggregate, rather than scored as zero, when this correlation is undefined (fewer than two held out documents, or zero variance in predicted or true activations). We report AIS as the unweighted mean over concepts with a defined value. The judge is Qwen2.5 14B Instruct, decoded greedily for determinism. For ANYBOTTLE, AIS covers the discovered concept set $\mathcal{S}^\star$ in full. For the linear SAE probe baseline, scoring the full dictionary is infeasible. We instead restrict to the active subset (activation greater than zero on at least five test documents), partition it into five quantile bins by document activation frequency, and draw 200 latents without replacement, split evenly across bins. This stratified draw is repeated over 20 independent seeds, and we report the mean and standard deviation of AIS across the 20 draws in our provided results.

G.4 PROMPTS

MAIA prompts. After 25 rounds the MAIA agent is prompted to conclude (Listing 4). The agent receives the system prompt (Listing 1), the user prompt (Listing 2), and the exemplar evidence (Listing 3), in that order.

Listing 1: MAIA system prompt (compact variant for 32k-context local VLMs).

```
You are MAIA, an interpretability agent labeling one visual concept/neuron in an AnyBottle
    SAE/CBM run.

You already have initialized objects named `system` and `tools`. Never import, construct,
    overwrite, or reassign
`System`, `Tools`, `system`, or `tools`.

Available calls:
- `tools.dataset_exemplars(system)` returns top activating real exemplars as `(activation,
    image_b64)` pairs.
- `tools.text2image(list[str])` generates synthetic images from text prompts.
- `tools.edit_images(base_images: list[str], editing_prompts: list[str])` edits given base
    images/prompts.
- `system.call_neuron(images)` returns `(activation_list, activation_map_list)` for images.
- `tools.display(...)` records images/text/results for you to inspect after execution.

Use only existing `system` and `tools`. Write experiments as Python code blocks. Prefer
    small experiments:
1-3 prompts or edits per code block. After seeing results, update hypotheses. When confident,
     stop writing code and
return final output exactly as:
[DESCRIPTION]: <specific visual concept>
[LABEL 1]: <short label>
[LABEL 2]: <optional second label>
[CONFIDENCE]: <0-100> - <one short reason, based on the evidence you gathered>
```

Listing 2: MAIA user prompt.

```
Your task is to identify the visual concept that maximally activates the current neuron/
    concept.

You are given ground-truth top activating exemplar evidence in the conversation. Use it as
    primary evidence.
You may run small synthetic `tools.text2image` or `tools.edit_images` experiments to test
    hypotheses, then call
`system.call_neuron` and `tools.display` to view activations/results.

Rules:
- Do not use examples from instructions as evidence.
- Do not reinitialize `system` or `tools`.
- Do not write a final description in the same response as code. If you write code, wait for
     execution results.
- Keep generated prompts specific and very few.

Begin by proposing one small experiment, or provide the final answer if the exemplar
    evidence is already decisive.

Important for this AnyBottle run: API examples in the system prompt (for example dogs, cats,
     grass, landscapes, or generated images) are usage examples only. They are not evidence
     about the current neuron. Never use those examples in the final label unless they are
    visibly present in the actual attached top activating exemplars.
```

```
The objects named `system` and `tools` are already initialized for the current AnyBottle SAE
      concept. Do not import, construct, overwrite, or reassign `System`, `Tools`, `system`,
      or `tools`. In code blocks, call methods on the existing objects, for example `tools.
     text2image(...)`, `tools.edit_images(...)`, `system.call_neuron(...)`, and `tools.
     dataset_exemplars(system)`.

Localisation: `system.call_neuron(images)` returns, per image, the concept's activation and
     a turbo heatmap of where in the image it responds. For a closer look use `overlays,
     info = system.concept_activation_map(images)` -- `info[i]` has a small `grid` (g x g
     ints 0-100, row 0 = top of the image) and a `summary` with `peak_xy` (x,y image
     fractions), `active_fraction`, `top10pct_mass` (near 1.0 = a sharp point, ~0.1 = spread
      evenly) and `quadrant_mass` [top-left, top-right, bottom-left, bottom-right]. Use it
     to tell a localised object/part (peak, low active_fraction, high top10pct_mass) from a
     whole-scene or texture concept (spread out). Reflect this in the label, e.g. 'dog's
     face (localised)' vs 'outdoor grass texture (diffuse)'.

Confidence: whatever final-answer format the instructions above specify, add one more line
     at the end -- exactly:
[CONFIDENCE]: <0-100> - <one sentence on what would raise or lower it, e.g. how many
     exemplars fit vs. don't, whether experiments causally confirmed the hypothesis or you
     only had exemplar evidence, conflicting concepts in the top exemplars>. Base it on the
     evidence you actually gathered, not general confidence in the task.

Attribute decomposition: an object-level label (e.g. 'dog') is a starting hypothesis, not a
     final answer. Before finalizing, decompose it along these dimensions and identify which
      one actually drives the activation:
  - shape/geometry (e.g. the silhouette or outline, independent of what fills it)
  - texture/material (e.g. fur, fabric weave, metal sheen, independent of the object's
     identity)
  - color (does the concept survive recoloring/desaturation, or does it collapse?)
  - part vs. whole (is it the whole object, or one localised part, e.g. ears, snout, a
     specific edge?)
  - context/background (does it need the surrounding scene, or does it fire on the object/
     part alone, cropped out?)
If `tools.edit_images`/`tools.text2image` are available, use them to test these directly: e.
     g. recolor or desaturate an exemplar and re-run `system.call_neuron` to see if
     activation survives; swap or blur the texture while keeping the same silhouette; crop
     to just the candidate part vs. the full object vs. just the background. An attribute
     the concept does NOT need should be explicitly ruled out in your hypothesis list, not
     just omitted. If those tools are unavailable in this run, still push the label as far
     down this hierarchy as the exemplar evidence alone supports (e.g. note consistent fur
     texture across varied dog breeds/poses as evidence for texture over shape), and say in
     [CONFIDENCE] that this wasn't causally tested.
```

Listing 3: Exemplar-evidence message (template; braces are filled in per concept).

```
Ground-truth visual evidence for the current neuron/concept follows. These are the actual
     mined exemplars for this run; API examples in the system prompt are not evidence and
     must not be used for the final label.

Top activating exemplar activations: {a_1, ..., a_4}

Top activating exemplar summary from the image-description model:
{summary of the top-4 exemplars}

For this local-VLM run, the raw exemplar image payloads were used to produce the summary
     above but are not repeated in this reasoning prompt because they exceed the model
     context window. Base your final description and labels on this visual summary and any
     subsequent displayed experiment results.
```

Listing 4: Final-answer prompt, issued after 25 rounds without a final answer (from OpenMAIA).

```
Please provide you final description in the following format:
[DESCRIPTION]: <final description> ## Your description should be selective (e.g. very
     specific: "dogs running on the grass" and not just "dog") and complete (e.g. include
     all relevant aspects the neuron is selective for). In cases where the neuron is
     selective for more than one concept, include in your description a list of all the
     concepts separated by logical "OR".
[LABEL]: <final label drived from the hypothesis or hypotheses> ## a label for the neuron
     generated from the hypothesis (or hypotheses) you are most confident in after running
     all experiments. They should be concise and complete, for example, "grass surrounding
     animals", "curved rims of cylindrical objects", "text displayed on computer screens", "
     the blue sky background behind a bridge", and "wheels on cars" are all appropriate. You
      should capture the concept(s) the neuron is selective for. Only list multiple
     hypotheses if the neuron is selective for multiple distinct concepts. List your
     hypotheses in the format:
[LABEL 1]: <label 1>
[LABEL 2]: <label 2>
```