SWORDBENCH: EVALUATING ORTHOGONALITY OF STEERING IMAGE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Steering or intervening on model representations at inference time to correct predictions is essential for AI interpretability and safety, yet existing evaluation protocols are limited to ambiguous language modeling tasks. To address this gap, we introduce SWORDBENCH, a benchmark for steering image representations of vision models across multiple backbones and concept removal tasks. Beyond a unified benchmarking suite, we propose new evaluation notions that uncover the second-order effects of orthogonalization among concept activation vectors for pragmatic steering. Specifically, *cross-concept robustness* measures the stability of concept detection performance across inputs orthogonalized against alternative concepts, and *collateral damage* quantifies whether steering inadvertently affects model performance on a downstream task for inputs lacking the bias. We find that although a linear support vector machine exhibits superior separability and orthogonality, it fails to achieve zero collateral damage, often trailing sparse autoencoders. In controlled synthetic settings, both standard baselines and optimization-based methods fail to achieve perfect steering.

1 INTRODUCTION

Self-supervised learning achieves remarkable performance on various language (Gemma Team, 2025), vision (Oquab et al., 2024), and vision-language tasks (Radford et al., 2021; Zhai et al., 2023). Unfortunately, models often rely on shortcut learning (Li et al., 2023b) and spurious correlations to make decisions (Pahde et al., 2023), and are prone to adversarial attacks (Wei et al., 2023), posing a key challenge for AI safety. To this end, interpretability research develops methods that enable understanding the internal behavior of large models, or even correcting their predictions by *steering*, i.e., by intervening on the model's representations at inference time without finetuning (Subramani et al., 2022; Li et al., 2023a; Rimsky et al., 2024; Qiu et al., 2024; Wu et al., 2025).

Steering has been widely studied to control language models at test-time without the unreliability of prompt engineering (Wei et al., 2023) or the misalignment risks of fine-tuning (Betley et al., 2025), where it typically targets abstract concepts such as 'honesty' (Zou et al., 2023). In this work, we make a natural connection that *steering language models* practically mirrors *concept removal in vision models*, where biases can be detected in the model's representation space (Fel et al., 2023a;b; Pahde et al., 2025). Both lines of work rely on the fundamental notion of a *concept activation vector* (CAV, Kim et al., 2018); throughout this paper, we refer to the intervention as *steering* and to the vectors as CAVs. The emerging utility of steering in alignment has spurred the development of numerous algorithms for computing CAVs, exposing a critical need for a unified benchmark to evaluate them. Although a few such evaluation protocols have recently been proposed for steering language models (Zou et al., 2023; Bhalla et al., 2024), notably AXBENCH (Wu et al., 2025), there is no equivalent for vision models, and they share fundamental limitations. Training CAVs for language models is inherently noisy because their multi-token representations are increasingly susceptible to spurious correlations (Tan et al., 2024). Moreover, evaluation metrics for generative models primarily rely on the *LLM-as-a-judge*, an imperfect proxy for human assessment (Wang et al., 2024a; Li et al., 2025). Thus, we seek metrics that can assess CAVs without such confounding factors.

This work: A multidimensional evaluation of steering image representations. We introduce SWORDBENCH, a benchmark for evaluating orthogonality, robustness and potential damage of steering image representations at scale (Figure 1). SWORDBENCH shifts the domain from language

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.

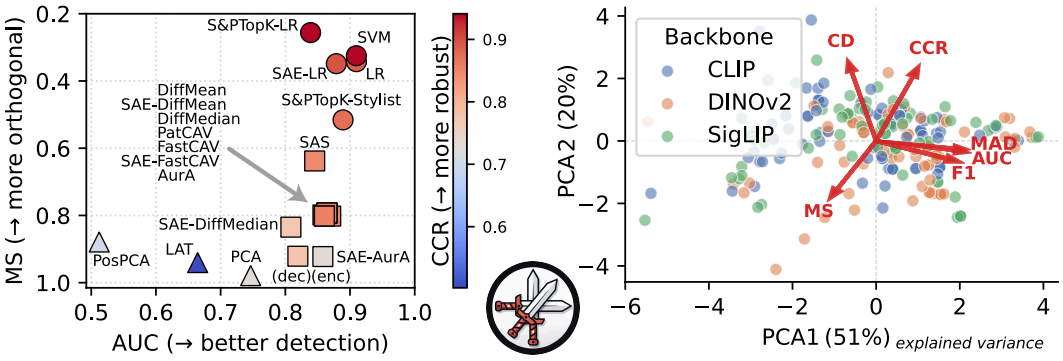

Figure 1: **SWORDBENCH multi-dataset concept activation vectors evaluation.** On the *left*, we report vector metrics for SigLIP image representations CAVs on CelebA (averaged over 39 concepts). Improved disentanglement (lower MS) correlates with higher robustness (CCR), with methods forming three clusters (●, ■, ▲) along the orthogonality and detection axes. On the *right*, we compute a PCA biplot on ISIC using all evaluated backbones to visualize metric correlations, where arrows indicate metric directions. While robustness and collateral damage (CD) appear uncorrelated on ISIC, they show correlation in other datasets (Appendix G).

to vision transformers, focusing on steering modern self-supervised learning architectures such as CLIP (Radford et al., 2021), SigLIP (Zhai et al., 2023) and DINOv2 (Oquab et al., 2024). Relying on *final image representations* used in downstream tasks effectively mitigates the challenge of *"How to steer across layers and tokens?"* and avoids the Hydra effect (McGrath et al., 2023) from benchmarking steering in language models. Furthermore, SWORDBENCH offers ground-truth concept labels across four datasets for bias-removal: CelebA (Liu et al., 2015), ISIC (Codella et al., 2019), Waterbirds (Sagawa et al., 2020) and Counteranimals (Wang et al., 2024b); as well as two controlled synthetic tasks with paired concept counterfactuals, ImageNet-W (Li et al., 2023b) and ImageNet-C (Hendrycks & Dietterich, 2019), which we adapt specifically to evaluate steering.

Our contributions are as follows: **(1) Benchmark.** We introduce SWORDBENCH, a unified collection of vision models, labeled datasets with two *controlled datasets*, standard CAV extraction methods, and performance metrics for evaluating steering. **(2) Metrics.** We propose two novel evaluation notions: *cross-concept robustness* measures the stability of concept detection for inputs orthogonalized against other concepts in the data, and *collateral damage* measures whether concept erasure affects downstream performance on inputs without the concept. Combined, these two metrics reveal the undesirable *second-order effects* of orthogonalization for CAV steering. **(3) Findings.** We use SWORDBENCH to deliver several insights, including evidence that some linear methods achieve separability on par with nonlinear probes, and that standard classification metrics are insufficient proxies for robustness, concept entanglement, or collateral damage. In a controlled setup, we demonstrate that while some methods achieve perfect steering, others do not.

2 RELATED WORK

Concept-based explanations build on the concept activation vector formalism (CAV, Kim et al., 2018) and the *linear representation hypothesis* (Nanda et al., 2023; Park et al., 2024), which posits that semantic concepts are encoded as linear directions in model activations. Beyond supervised linear probes (Alain & Bengio, 2017), a range of extraction methods has been proposed (Zou et al., 2023; Pahde et al., 2025; Schmalwasser et al., 2025), alongside sparse autoencoders (SAE, Templeton et al., 2024; Gao et al., 2025; Zaigrajew et al., 2025; Pach et al., 2025) that decompose polysemantic representations (Bricken et al., 2023; Kopf et al., 2025) into monosemantic features. In parallel, the field has shifted from explanation toward intervention (Geiger et al., 2025): in the language domain, steering manipulates behavioral outputs such as refusal rates or writing style (O'Brien et al., 2025; Rimsky et al., 2024; Venhoff et al., 2025; Wu et al., 2025), while in vision the analogous application is bias removal (Dreyer et al., 2024; Bareeva et al., 2024; Pahde et al., 2025), i.e., concept erasure (Belrose et al., 2023). SWORDBENCH unifies these perspectives by evaluating the bias-removal task and benchmarks 16 CAV methods. For an extended discussion, we refer to Appendix B.1.

3 SWORDBENCH

The ultimate goal of SWORDBENCH is to systemize the evaluation of steering image representations. Moreover, SWORDBENCH shifts the focus from the often noisy, multi-token setting of language models to the more controlled benchmarking environment of vision models. In what follows, we formalize the problem setup in Section 3.1, introduce benchmarked CAV extraction methods in Sec. 3.2, define evaluation metrics in Sec. 3.3, and detail the datasets used for evaluation in Sec. 3.4. Finally, we present the key results from our comprehensive analysis in Sec. 4, while several minor findings are deferred to the Appendix. For clarity, a summary of notation is provided in Appendix A.

3.1 PROBLEM FORMULATION

The global representation advantage. Self-supervised learning vision models (e.g., CLIP, DINOv2, SigLIP) are trained to compress the input into a single, global representation with the [CLS] token or mean-pooled last-layer tokens (Radford et al., 2021; Oquab et al., 2024; Zhai et al., 2023). This is unlike language models, in which semantic information is distributed across tokens, often making the optimal target for intervention ambiguous. Such structural clarity eliminates the uncertainty of selecting *which* token represents the concept or *where* to intervene, thereby avoiding the potential *"Hydra effect"* that occurs when steering intermediate transformer layers (McGrath et al., 2023), which we examine empirically in Appendix C.3. By isolating the representation, SWORDBENCH allows for a rigorous assessment of the vector itself, *decoupling* the quality of the CAV from the complexity of circuit discovery. Importantly, our goal is not to maximize concept discovery within model, but to establish a fair and standardized framework for evaluating steering methods.

Rethinking evaluation. Evaluating *bias removal* on natural data is inherently challenging due to the absence of ground-truth baselines and the lack of counterfactual samples (with and without the concept). Resonating with responsible benchmarking principles, which prioritize reproducible evaluation, SWORDBENCH evaluates removal success exclusively on controlled data using synthetic counterfactuals to ensure ground truth validity. To verify that these vectors generalize and are not merely artifacts of synthetic settings, we evaluate steering on popular non-synthetic datasets along two axes: (1) *detection accuracy*, measuring how accurately the projected CAV identifies the presence of the target concept, and (2) *steering specificity*, showing the ability of CAV to erase the concept without *degrading* performance on concept absent samples, i.e. causing *collateral damage*.

Concept activation vectors. Formally, let ϕ be a model mapping an input $\mathbf{x}$ to a hidden representation $h \in \mathbb{R}^d$, such that $h = \phi(\mathbf{x})$. Within this latent space, a *unit norm* concept activation vector (CAV, Kim et al., 2018), denoted as $\mathbf{v}_c \in \mathbb{R}^d$, represents the linear direction of a human-interpretable concept c. Similarly to linear probes (Alain & Bengio, 2017), $\mathbf{v}_c$ can be derived from a binary dataset $\mathcal{D}_c = \{(h_k, y_k)\}_{k=1}^{2N}$. $\mathcal{D}_c$ is constructed from two (typically) balanced populations of size N: the concept-positive set $\mathbb{A}_c = \{h \mid y = 1\}$ and the concept-negative set $\mathbb{A}_{\bar{c}} = \{h \mid y = 0\}$. N negative samples are drawn from counterfactual pairs for synthetic data or from random images lacking the concept. Guided by our stability analysis (cf. ablations in Section 4), we set $N = 500$ to ensure convergence, unless constrained by data scarcity.

Model setup. We conduct experiments on three frozen vision backbones ϕ: DINOv2 (ViT-B/14), CLIP (ViT-L/14), and SigLIP (ViT-B/16). We focus exclusively on the *final-layer representations*, which compress global semantic information and serve as input to the downstream tasks; in our case, model classifiers ψ. For all predictive tasks, ψ is a lightweight MLP (1–2 layers) trained on the frozen backbone. Model training details are provided in Appendix C.2.

Downstream steering. The core of SWORDBENCH is evaluating the causal effect of concept removal on downstream tasks. We measure this by quantifying the performance shift between the original $\psi(h)$ and steered $\psi(\bar{h})$ representations on a held-out validation set. We implement concept erasure by projecting out the component of the representation parallel to the CAV $\mathbf{v}_c$, effectively flattening the concept direction via **orthogonalization**: $\bar{h} = h - (\mathbf{v}_c^T h)\mathbf{v}_c$. Crucially, we apply this intervention to samples *lacking* the concept and to counterfactual pairs, which allows us to assess both the *side effects* and the *effectiveness* of steering. Ideally, steering should not alter performance on samples where the concept is not present, while for counterfactuals, the steered representation should match the performance of the corresponding natural concept absent samples. We justify the use of orthogonalization as the steering approach for benchmarking in Appendix B.2 and C.4.

3.2 Concept Activation Vector (CAV) Extraction Methods

We benchmark a diverse suite of CAV extraction methods, categorized into *standard linear* approaches and *sparse autoencoders*, some of which were benchmarked in AxBench (Wu et al., 2025). In addition to CAVs, we include *nonlinear probes* to empirically validate the linear representation hypothesis and quantify the limitations of linear CAVs. In Appendix C.5, we provide detailed hyperparameter setups for all the methods.

3.2.1 Standard Linear Methods

Difference-in-Means (**DiffMean**, Li et al., 2023a; Marks & Tegmark, 2024) is a foundational baseline in research on steering, which assumes the concept direction connects the positive and negative class centroids: $\mathbf{v}_{\text{DiffMean}} \coloneqq \frac{1}{|\mathbb{A}_c|}\sum_{h\in\mathbb{A}_c} h - \frac{1}{|\mathbb{A}_{\bar{c}}|}\sum_{h\in\mathbb{A}_{\bar{c}}} h$. Difference-in-Medians (**DiffMedian**) is a robust variant of DiffMean designed to mitigate the effect of outliers. It computes the element-wise median difference: $\mathbf{v}_{\text{DiffMedian}} \coloneqq \text{median}(\mathbb{A}_c) - \text{median}(\mathbb{A}_{\bar{c}})$. We further train a linear Support Vector Machine (**SVM**) with L_2 regularization to find the maximum-margin hyperplane separating sets $\mathbb{A}_c$ and $\mathbb{A}_{\bar{c}}$. The CAV $\mathbf{v}_{\text{SVM}}$ is the normal vector to this hyperplane. Similarly, we train a Logistic Regression (**LR**) model with L_2 regularization, defining $\mathbf{v}_{\text{LR}}$ as the model's coefficient vector.

Fast Concept Activation Vectors (**FastCAV**, Schmalwasser et al., 2025) overcomes the computational costs of iterative training. It computes the direction by centering the positive activations relative to the global mean embedding (μ_{global}) of inputs from the combined sets $\mathbb{A}_c \cup \mathbb{A}_{\bar{c}}$: $\mathbf{v}_{\text{FastCAV}} \coloneqq \frac{1}{|\mathbb{A}_c|}\sum_{h\in\mathbb{A}_c}(h - \mu_{\text{global}})$. Conversely, Pattern-Based CAV (**PatCAV**, Pahde et al., 2025) addresses the directional divergence where classifier weights filter rather than represent the signal, recovering the concept direction directly. Instead of interpreting probe weights, it computes the signal pattern via a closed-form regression solution (Haufe et al., 2014): $\mathbf{v}_{\text{PatCAV}} \coloneqq \frac{\text{Cov}(h,y)}{\text{Var}(y)}$, where y denotes the binary concept labels.

We evaluate two Principal Component Analysis variants: **PCA** computes the first principal component of the combined dataset $\mathbb{A}_c \cup \mathbb{A}_{\bar{c}}$, capturing the dominant variation axis across the distribution. **PosPCA** computes the first principal component of only the positive set $\mathbb{A}_c$, isolating the variance structure specific to the concept expression. Linear Artificial Tomography (**LAT**, Zou et al., 2023) extracts a latent direction by analyzing pairwise activation differences. We construct normalized difference vectors $\delta_{ij} = \frac{(h_i - h_j)}{\|h_i - h_j\|}$ for N pairs such that $h_i \in \mathbb{A}_c$ and $h_j \in \mathbb{A}_{\bar{c}}$. For synthetic datasets, we strictly pair counterfactuals; for real datasets, we use random pairing. Vector $\mathbf{v}_{\text{LAT}}$ is the first principal component of the difference matrix $\Delta \in \mathbb{R}^{N\times d}$ containing vectors δ_{ij} in rows.

AUROC Adaptation (**AurA**, Suau et al., 2024) constructs the CAV by first calculating the AUC of $h^{(j)}$ on $\mathbb{A}_c$ and $\mathbb{A}_{\bar{c}}$, treating each dimension $j \in \{1, \ldots, d\}$ of the represetion h as an individual component. The $\mathbf{v}_{\text{AurA}}$ is then formed by zeroing components where $\text{AUC}(h^{(j)}) \leq 0.5$ and assigning the remaining weights according to the Gini coefficient: $v^{(j)}_{\text{AurA}} \coloneqq 2 \cdot (\text{AUC}(h^{(j)}) - 0.5)$.

3.2.2 SAE-Based Methods

SAEs disentangle polysemantic representations into monosemantic neurons. An SAE consists of an encoder $z = \text{ReLU}(W_{\text{enc}}h + b_{\text{enc}})$ resulting in sparse features $z \in \mathbb{R}^{d_{\text{SAE}}}$, and a decoder $\hat{h} = W_{\text{dec}}z$. We train SAEs for each vision backbone using TopK regularization (Gao et al., 2025) with $k = 64$ and expansion factor 32. Training details are provided in Appendix C.6.

We implement **SAE-DiffMean / Median / FastCAV** by applying statistical aggregations of DiffMean, DiffMedian, and FastCAV directly on the sparse representations z to obtain a latent direction $\mathbf{w}_z$. The final $\mathbf{v}_{\text{SAE}}$ is obtained by decoding this direction to the h space: $\mathbf{v}_{\text{SAE}} \coloneqq W_{\text{dec}}\mathbf{w}_z$. Sparse Activation Steering (**SAS**, Bayat et al., 2025) applies density-based filtering on both $\mathbb{A}_c$ and $\mathbb{A}_{\bar{c}}$ to mitigate noise from rare or spurious features. We select relevant SAE neurons based on activation density and optimize the threshold via cross-validation to maximize AUC. We also exclude neurons active in both sets, assuming that shared features represent context and not the concept. Vector $\mathbf{v}_{\text{SAS}}$ is obtained similarly to SAE-DiffMean, but only on the remaining neurons from z.

For SAE Probe (**SAE-LR**), we train a logistic regression classifier (similar to LR) on the sparse representation z, using L_1 regularization instead to encourage sparsity. The learned weight vector $\mathbf{w}_z$ is

projected back via the decoder $\mathbf{v}_{\text{SAE-LR}} \coloneqq W_{\text{dec}}\mathbf{w}_z$ as in previous SAE methods. Selection and Projection TopK (**S&PTopK**, Barbalau et al., 2025) uses a two-stage CAV extraction process. First, we identify candidate neurons by training an LR with an L_1 penalty or by applying the Stylist (Smeu et al., 2025) ranking method, selecting the indices $\mathcal{S}$ of the Top-K neurons based on their corresponding LR coefficients (fixed at $K = 16$ as originally proposed). Second, we re-train an LR with L_2 penalty exclusively on this subset to obtain refined coefficients $\mathbf{w}_{\mathcal{S}}$. Uniquely, the final CAV projects these coefficients using the *encoder* rows: $\mathbf{v}_{\text{S\&PTopK}} \coloneqq \sum_{k\in\mathcal{S}} \mathbf{w}_{\mathcal{S}}^{(k)} \cdot W_{\text{enc}}[k]$.

Finally, leveraging SAE's disentangled representation and monosemantic neurons, we adapt the AurA to the SAE latent space z and call it **SAE-AurA**. After computing the SAE AurA vector $\mathbf{w}_{\text{SAE-AurA}}$, we decode it back to the h space using either the decoder (W_{dec}) or the encoder (W_{enc}) weights, giving two approaches: $\mathbf{v}_{\text{SAE-AurA}} \coloneqq W\mathbf{w}_{\text{SAE-AurA}}$, where $W \in \{W_{\text{dec}}, W_{\text{enc}}^T\}$.

3.2.3 Nonlinear Probes

To establish an upper bound on concept detection and test the linear hypothesis, we evaluate two nonlinear classifiers. We use a support vector classifier with a RBF kernel (**SVM-RBF**) and **TabPFN** (Hollmann et al., 2025), a state-of-the-art pre-trained transformer for tabular data. As a hyperparameter-free AutoML solution, it serves as a robust and reliable baseline.

3.3 Evaluation Metrics

We decompose the evaluation into (i) *vector metrics* that assess the intrinsic fidelity of the extracted CAVs through detection accuracy and concept entanglement, and (ii) *steering metrics* that measure the causal impact of interventions through debiasing success and damage on model performance.

3.3.1 Vector Metrics

Concept Detection (**AUC**) quantifies the CAV's ability to linearly separate the concept by computing the area under the ROC curve using projection scores $s(h) = \mathbf{v}_c^T h$. For a concept c, this is defined as: $\text{AUC}(\mathbf{v}_c) \coloneqq \frac{1}{|\mathbb{A}_{\bar{c}}||\mathbb{A}_c|} \sum_{h_{\bar{c}}\in\mathbb{A}_{\bar{c}}} \sum_{h_c\in\mathbb{A}_c} \mathbb{I}[\mathbf{v}_c^T h_c > \mathbf{v}_c^T h_{\bar{c}}]$,
where $\mathbb{I}[\cdot]$ is the indicator function. A higher AUC reflects better separation of concept samples.

Maximum Similarity (**MS**) detects if CAVs represent distinct semantic directions by measuring the highest pairwise cosine similarity between all concept vectors in a dataset for a given extraction method. For a set of unit norm concept vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$, the metric for concept c is:

$$\text{MS}(\mathbf{v}_c) \coloneqq \max_{j\neq c} \mathbf{v}_c^T \mathbf{v}_j. \tag{1}$$

Lower values indicate better disentanglement from other concepts, mitigating untargeted steering.

Because MS ignores functional interference, we introduce Cross-Concept Robustness (**CCR**), a proxy metric to preemptively estimate robustness to collateral damage. We measure the stability of the target concept c when removing any other concept j. First, we compute the representation $h_{c\perp j} = h_c - (\mathbf{v}_j^T h_c)\mathbf{v}_j$ orthogonalized with respect to a non-target $\mathbf{v}_j$. Next, CCR is defined as the minimum normalized AUC of target c retained after removing any concept j:

$$\text{CCR}(\mathbf{v}_c) \coloneqq \min_{j\neq c} \frac{\text{AUC}(\mathbf{v}_c \mid h_{c\perp j})}{\text{AUC}(\mathbf{v}_c \mid h_c)}. \tag{2}$$

High CCR values indicate conceptual independence for $\mathbf{v}_c$, while lower scores reflect a dependency on other concepts. In future evaluations, this metric may serve as a proxy for multi-concept steering. We also measure the Mean Activation Difference (**MAD**, Kopf et al., 2024) as detailed in Appendix B.3. However, we omit reporting this metric due to its high correlation with AUC, a relationship shown in Figure 1 and further discussed in Appendix G.

3.3.2 Steering Metrics

Downstream Concept Detection (**F1 Score**) measures the CAVs concept detection performance on the downstream datasets via the F1 score: $\text{F1}(y, \hat{y}) \coloneqq \frac{2\sum_{i=1}^N \mathbb{I}[y_i=1,\hat{y}_i=1]}{\sum_{i=1}^N \mathbb{I}[y_i=1]+\sum_{i=1}^N \mathbb{I}[\hat{y}_i=1]}$, where $\hat{y}_i$ is a binary prediction thresholded using Youden's J statistic on the training set.

We posit that a perfect concept-removal method should have zero effect on samples that do not possess the concept and propose Collateral Damage (**CD**) to quantify this. For the concept-absent subset ($\mathbb{A}_{\bar{c}}$), we measure the drop in downstream accuracy between original and steered representations:

$$\mathrm{CD}(\mathbf{v}_c) \coloneqq \mathrm{Acc}\big(\psi(h) \,|\, h \in \mathbb{A}_{\bar{c}}\big) - \mathrm{Acc}\left(\psi(\bar{h}) \,|\, h \in \mathbb{A}_{\bar{c}}\right). \tag{3}$$

Ideally, CD ≈ 0. A large value indicates that the CAV is entangled with other task-relevant features. Furthermore, for synthetic datasets with exact counterfactual pairs (clean $h \in \mathbb{A}_{\bar{c}}$, infused $h_{\text{concept}} \in \mathbb{A}_c$), we measure Steering Disparity (**SD**), which indicates whether removing the concept from h_{concept} recovers the prediction of the clean counterpart:

$$\mathrm{SD}(\mathbf{v}_c) \coloneqq \frac{\mathrm{Acc}\big(\psi(h) \,|\, h \in \mathbb{A}_{\bar{c}}\big) - \mathrm{Acc}\big(\psi(\bar{h}) \,|\, h \in \mathbb{A}_c\big)}{\mathrm{Acc}\big(\psi(h) \,|\, h \in \mathbb{A}_{\bar{c}}\big) - \mathrm{Acc}\big(\psi(h) \,|\, h \in \mathbb{A}_c\big)}. \tag{4}$$

Ideally, SD ≈ 0, indicating that steering has successfully inverted the concept addition.

3.4 Datasets

Steering success can be conflated with dataset properties. To analyze these factors, we adopt a robust evaluation strategy that uses 4 real-world bias-removal datasets and two controlled synthetic datasets. Dataset splits, concept distributions, and sample visualizations are deferred to Appendix C.1. To ensure reliable generalization, we train all CAVs on balanced samples from the training split and validate vector metrics on the validation split. For downstream tasks, model selection is driven by validation performance, while all final steering is performed on a test split.

CelebA. The CelebA dataset (Liu et al., 2015) is a standard benchmark for bias removal (Sagawa et al., 2020) and CAV evaluation (Pahde et al., 2025). It consists of facial images annotated with 40 binary attributes such as *male*, *smiling*, or *bald*. Unlike other datasets, CelebA lacks a designated target task, so we follow prior work and use the *male* concept as the classification target, excluding it from the evaluated concepts.

ISIC. To evaluate steering in a high-stakes medical domain, we use ISIC 2018 (Codella et al., 2019) for skin lesion classification with eight biases that simulate sampling artifacts (Bissoto et al., 2020). This dataset is widely adopted for bias removal and CAV evaluation (Wu et al., 2023; Pahde et al., 2025). Crucially, due to significant overlap between the training and validation sets, we evaluate CAV metrics on the test split and report results for the four concepts with sufficient sample support.

Counteranimal. To address environmental biases in models like CLIP (Wang et al., 2024b), we include Counteranimal, which categorizes images into *common* (typical background, e.g., a polar bear on ice) and *uncommon* (atypical background, e.g., a polar bear on grass) groups. We train CAVs to capture the direction of spurious background correlations and next test whether neutralizing these concepts enables background-invariant classification. Due to space constraints, we report results only in the Appendix C.1.

ImageNet-W (watermark) and **ImageNet-C** (corruptions). A critical challenge in evaluating real-world data is the lack of ground-truth concept counterfactuals to assess the steering success. To address this, we use ImageNet-W (Li et al., 2023b), which modifies images by overlaying transparent textual *watermarks*, and ImageNet-C (Hendrycks & Dietterich, 2019), which applies one of 19 corruptions, such as glass blur or shot noise. This allows us to generate exact sample pairs $(\mathbf{x}_{\bar{c}}, \mathbf{x}_c)$ and precisely measure the CAV's impact on the concept alone. Such an approach is particularly useful for measuring concept erasure on models sensitive to text, such as CLIP and SigLIP (see e.g. Baniecki et al., 2025) and for evaluating classifier robustness under real-world conditions. For computational efficiency, we limit our evaluation to a diverse subset of 10 classes, targeting the most degrading watermark-class pairs and corruptions (see Appendix C.1).

Waterbirds and **MetaShift.** Waterbirds (Sagawa et al., 2020) is a well-established benchmark for spurious correlation analysis (Wu et al., 2023). It specifically targets the natural correlation between a bird species (*landbirds* or *waterbirds*) and its background (*water* or *land*), which is balanced on the test split. Given the concepts' antipodal structure (see Table 22), we flag this dataset as low-quality and report its results only in the Appendix. We also investigated MetaShift (Liang & Zou, 2022), but excluded it as well due to frequent label conflicts (see Appendix C.1).

Table 1: **DINOv2 representations steering evaluation on CelebA and ISIC.** Values show the mean ± two standard errors (2SE) across concepts. The final CD metric is reported as an absolute value. The best method (row) in each evaluation metric (column) is in **bold** and the second best is underlined. For additional reference, two nonlinear probe baselines are denoted with **gray**. Extended results for the other models, the remaining datasets and metrics are in Appendix D.

	CelebA					ISIC				
Method	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)
Linear SVM	**.88**$_{\pm.03}$	**.97**$_{\pm.01}$	**.17**$_{\pm.03}$	**.56**$_{\pm.06}$	.02$_{\pm.02}$	**.91**$_{\pm.09}$	**.98**$_{\pm.02}$	**.07**$_{\pm.04}$	**.81**$_{\pm.15}$	1.6$_{\pm2.3}$
LR	**.88**$_{\pm.03}$	.93$_{\pm.02}$	.30$_{\pm.05}$	**.56**$_{\pm.07}$	.35$_{\pm.46}$	**.91**$_{\pm.09}$	.97$_{\pm.04}$	.09$_{\pm.03}$	**.81**$_{\pm.16}$	2.3$_{\pm2.8}$
↳ SAE	.84$_{\pm.03}$	.87$_{\pm.03}$	.45$_{\pm.08}$	.51$_{\pm.06}$	.71$_{\pm.84}$	.84$_{\pm.11}$	.97$_{\pm.04}$	.12$_{\pm.05}$	.69$_{\pm.16}$	2.9$_{\pm2.8}$
SAS	.79$_{\pm.04}$	.82$_{\pm.03}$	.72$_{\pm.06}$	.45$_{\pm.07}$	3.7$_{\pm1.5}$	.75$_{\pm.12}$	.94$_{\pm.03}$	.40$_{\pm.18}$	.69$_{\pm.13}$	2.9$_{\pm6.5}$
S&PTopK	.85$_{\pm.03}$	.79$_{\pm.05}$	.64$_{\pm.06}$	.52$_{\pm.07}$	.42$_{\pm.26}$	.87$_{\pm.12}$	.92$_{\pm.08}$	.37$_{\pm.09}$	.77$_{\pm.14}$	**.72**$_{\pm2.0}$
DiffMean	.83$_{\pm.03}$	.82$_{\pm.03}$	.82$_{\pm.05}$	.49$_{\pm.07}$	15.$_{\pm4.8}$	.86$_{\pm.07}$	.93$_{\pm.06}$	.50$_{\pm.25}$	.73$_{\pm.12}$	9.0$_{\pm6.7}$
↳ SAE	.83$_{\pm.03}$	.81$_{\pm.03}$	.84$_{\pm.05}$	.48$_{\pm.07}$	16.$_{\pm5.0}$	.83$_{\pm.08}$	.95$_{\pm.04}$	.50$_{\pm.25}$	.71$_{\pm.12}$	9.5$_{\pm11.}$
DiffMedian	.83$_{\pm.03}$	.83$_{\pm.03}$	.82$_{\pm.05}$	.48$_{\pm.07}$	15.$_{\pm4.7}$	.85$_{\pm.07}$	.94$_{\pm.06}$	.51$_{\pm.30}$	.73$_{\pm.11}$	10.$_{\pm5.8}$
↳ SAE	.78$_{\pm.04}$	.76$_{\pm.04}$	.85$_{\pm.04}$	.44$_{\pm.07}$	7.4$_{\pm3.0}$	.74$_{\pm.07}$	.96$_{\pm.03}$	.48$_{\pm.29}$	.67$_{\pm.13}$	2.3$_{\pm4.9}$
FastCAV	.83$_{\pm.03}$	.82$_{\pm.03}$	.82$_{\pm.05}$	.49$_{\pm.07}$	15.$_{\pm4.9}$	.86$_{\pm.07}$	.93$_{\pm.06}$	.50$_{\pm.25}$	.73$_{\pm.12}$	9.0$_{\pm6.7}$
↳ SAE	.83$_{\pm.03}$	.81$_{\pm.03}$	.84$_{\pm.05}$	.48$_{\pm.07}$	16.$_{\pm5.0}$	.83$_{\pm.08}$	.95$_{\pm.04}$	.50$_{\pm.25}$	.71$_{\pm.12}$	9.5$_{\pm10.}$
PatCAV	.83$_{\pm.03}$	.82$_{\pm.03}$	.82$_{\pm.05}$	.49$_{\pm.07}$	15.$_{\pm4.9}$	.86$_{\pm.07}$	.93$_{\pm.06}$	.50$_{\pm.25}$	.73$_{\pm.12}$	9.0$_{\pm6.7}$
AurA	.83$_{\pm.03}$	.81$_{\pm.03}$	.85$_{\pm.04}$	.49$_{\pm.07}$	.62$_{\pm.22}$	.86$_{\pm.07}$	.93$_{\pm.06}$	.62$_{\pm.18}$	.73$_{\pm.12}$	6.1$_{\pm4.5}$
↳ SAE	.81$_{\pm.03}$	.62$_{\pm.04}$	1.0$_{\pm.00}$	.47$_{\pm.06}$	**.01**$_{\pm.01}$	.83$_{\pm.09}$	.85$_{\pm.12}$	.98$_{\pm.01}$	.71$_{\pm.12}$	1.2$_{\pm1.5}$
PCA	.68$_{\pm.04}$	.60$_{\pm.05}$	.96$_{\pm.02}$	.35$_{\pm.06}$	8.3$_{\pm3.9}$	.68$_{\pm.16}$	.74$_{\pm.26}$	.64$_{\pm.71}$	.59$_{\pm.18}$	13.$_{\pm4.1}$
LAT	.56$_{\pm.02}$	.52$_{\pm.07}$	.96$_{\pm.01}$	.28$_{\pm.06}$	2.8$_{\pm2.0}$	.63$_{\pm.14}$	.43$_{\pm.21}$	.51$_{\pm.92}$	.58$_{\pm.18}$	10.$_{\pm5.1}$
SVM-RBF	**.88**$_{\pm.03}$	–	–	–	–	**.91**$_{\pm.01}$	–	–	–	–
TabPFN	**.88**$_{\pm.03}$	–	–	–	–	**.91**$_{\pm.01}$	–	–	–	–

4 RESULTS

We use SWORDBENCH to deliver several insightful findings regarding steering across multiple CAV methods, datasets, model backbones, and metrics. Extended analyses are provided in the Appendix.

Finding 1: Linear SVM and logistic regression consistently achieve top concept detection accuracy, matching the performance of nonlinear probes across all datasets.

The primary proxy metric used prior to steering analysis is AUC. Across all benchmarks, we observe that linear SVM and LR consistently yield the highest AUC and F1 scores (cf. Tables 1 & 2). Notably, these methods match the performance of complex nonlinear probes, such as TabPFN. We validate this observation on the CelebA and ISIC datasets in Appendix F, where we find no statistically significant difference between these two linear approaches and their nonlinear counterparts.

Finding 2: Optimization-based methods (●) maximize vector separability, excelling in regimes with high concept co-occurrence often found in real-world data; however, they risk overfitting to redundant features in controlled settings. Conversely, while statistical methods (■) may underperform in detection metrics or vector orthogonality, on controlled datasets they identify the core semantic direction more faithfully, as indicated by lower SD, second only to ▲.

We can categorize methods into optimization-based (●), statistical (■), and environment-specific (▲) based on their method type as presented in Figure 1. Environment-specific methods (▲) has weak real-data performance but perfect steering on the synthetic task (Table 2). Among statistical methods (■), FastCAV, DiffMean, or PatCAV, are empirically and mathematically equivalent (see Appendix B.4). We recognize AurA as the best method from this family as it achieves the highest overall performance on both synthetic and real-world data. Comparing optimization-based (●) and statistical (■) methods reveals a critical trade-off: ● excel at disentanglement in complex, real-word

Table 2: **SigLIP representation steering evaluation on ImageNet-W watermark–class pairs (*husky–cat* and *church–goldfish*), and ImageNet-C *Shot Noise*, and *Glass Blur* corruptions.** CAVs are trained on $N = 500$ counterfactual pairs sampled from all classes (All Classes Paired setup). LAT achieves the best SD and CD on ImageNet-W but performs poorly on ImageNet-C, while AurA and DiffMedian excel on both datasets. Values report the mean ± 2SE over five seeds.

	Husky on Cat			Church on Goldfish			Shot Noise		Glass Blur	
Method	F1 (↑)	CD (↓)	SD (↓)	F1 (↑)	CD (↓)	SD (↓)	F1 (↑)	SD (↓)	F1 (↑)	SD (↓)
Linear SVM	$.89_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$.39_{\pm.15}$	$\mathbf{.97}_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.69_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$1.1_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.03}$
LR	$.88_{\pm.07}$	$\mathbf{.00}_{\pm.00}$	$.49_{\pm.27}$	$\underline{.96}_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.66_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.97_{\pm.04}$
↳ SAE	$.71_{\pm.22}$	$.04_{\pm.08}$	$.34_{\pm.09}$	$.86_{\pm.07}$	$\mathbf{.00}_{\pm.00}$	$.73_{\pm.08}$	$\mathbf{1.0}_{\pm.00}$	$1.1_{\pm.03}$	$.98_{\pm.01}$	$.96_{\pm.01}$
SAS	$.85_{\pm.16}$	$.04_{\pm.08}$	$.32_{\pm.28}$	$.83_{\pm.03}$	$\mathbf{.00}_{\pm.00}$	$.49_{\pm.36}$	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.93_{\pm.02}$
S&PTopK	$\underline{.90}_{\pm.02}$	$\mathbf{.00}_{\pm.00}$	$.65_{\pm.20}$	$\underline{.96}_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.66_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$.97_{\pm.02}$	$.99_{\pm.00}$	$.95_{\pm.00}$
DiffMean	$.88_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.11_{\pm.00}$	$.77_{\pm.04}$	$.08_{\pm.10}$	$.19_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.96}_{\pm.06}$	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.02}$
↳ SAE	$.86_{\pm.01}$	$.20_{\pm.00}$	$.24_{\pm.00}$	$.63_{\pm.02}$	$.40_{\pm.00}$	$.56_{\pm.08}$	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.88_{\pm.02}$
DiffMedian	$.87_{\pm.02}$	$\mathbf{.00}_{\pm.00}$	$\underline{.09}_{\pm.05}$	$.80_{\pm.01}$	$.08_{\pm.10}$	$\mathbf{.15}_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.96}_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.00}$
↳ SAE	$.82_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.72_{\pm.19}$	$.68_{\pm.03}$	$\mathbf{.00}_{\pm.00}$	$.53_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$.96_{\pm.01}$	$.99_{\pm.00}$	$\underline{.87}_{\pm.02}$
FastCAV	$.88_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.11_{\pm.00}$	$.77_{\pm.04}$	$.08_{\pm.10}$	$.19_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.96}_{\pm.06}$	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.02}$
↳ SAE	$.86_{\pm.01}$	$.20_{\pm.00}$	$.24_{\pm.00}$	$.63_{\pm.02}$	$.40_{\pm.00}$	$.56_{\pm.08}$	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.88_{\pm.02}$
PatCAV	$.88_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.11_{\pm.00}$	$.77_{\pm.04}$	$.08_{\pm.10}$	$.19_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.96}_{\pm.06}$	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.02}$
AurA	$\mathbf{.95}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.11_{\pm.00}$	$.89_{\pm.02}$	$\mathbf{.00}_{\pm.00}$	$.49_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.96}_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.81}_{\pm.01}$
↳ SAE	$.89_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.24_{\pm.00}$	$\underline{.96}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.66_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$.98_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.96_{\pm.00}$
PCA	$.46_{\pm.04}$	$8.8_{\pm1.9}$	$3.6_{\pm1.0}$	$.20_{\pm.01}$	$9.1_{\pm4.5}$	$2.6_{\pm.71}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.93}_{\pm.04}$	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.04}$
LAT	$.51_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.04}_{\pm.06}$	$.54_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.15}_{\pm.00}$	$.75_{\pm.01}$	$1.6_{\pm.07}$	$.73_{\pm.00}$	$1.5_{\pm.16}$

data such as CelebA or ISIC, yielding better MS and CCR (Figure 2) and CD (Figure 8). In contrast, in the controlled synthetic domain, statistical (■) methods often achieve greater steering success despite lower detection scores (Table 2), as further evidenced in the Appendix. To summarize, • methods excel on entangled, real-world data but risk overfitting on synthetic datasets. On the other hand, ■ methods capture core semantics in controlled regimes but struggle with disentanglement.

Finding 3: While most methods achieve zero collateral damage, only LAT (▲) consistently achieves optimal steering disparity on ImageNet-W, while PCA achieves it on ImageNet-C. This highlights that, in controlled setups, popular methods like DiffMean, SAE, and SVM optimize for axes other than the critical steering metric, resulting in underperformance on that main task.

In synthetic settings (ImageNet-W), while most CAVs minimize collateral damage and possess high F1, only LAT (Table 2), LR, PCA, and SAS (Table 27) achieve perfect steering, with PCA consistently yielding the lowest collateral damage on ImageNet-C. LAT succeeds only under the counterfactual parity assumption and, interestingly, demonstrates that perfect steering does not require high separability (Tables 25 and 27). We highlight that although standard optimization methods (•) like SVM and LR yield peak performance and orthogonality (e.g., Figure 1), both optimization and statistical methods (■) struggle to achieve perfect steering. Instead, LAT and PCA lead in the benchmark for steering on ImageNet-W and ImageNet-C, respectively.

Finding 4: Standard classification metrics, such as AUC and F1, are insufficient proxies for concept entanglement, robustness, collateral damage, or steering disparity.

Performance metrics such as AUC and F1 are inadequate for assessing the safety and fidelity of steering, a limitation highlighted in Findings 2 and 3 and further corroborated by Figure 1. While better disentanglement (lower MS) correlates with higher CCR, (Appendix G), Figure 1 demonstrates that these metrics do not correlate with detection metrics. Furthermore, vector metrics fail to predict steering metrics, motivating the improved evaluation protocols introduced in SWORDBENCH.

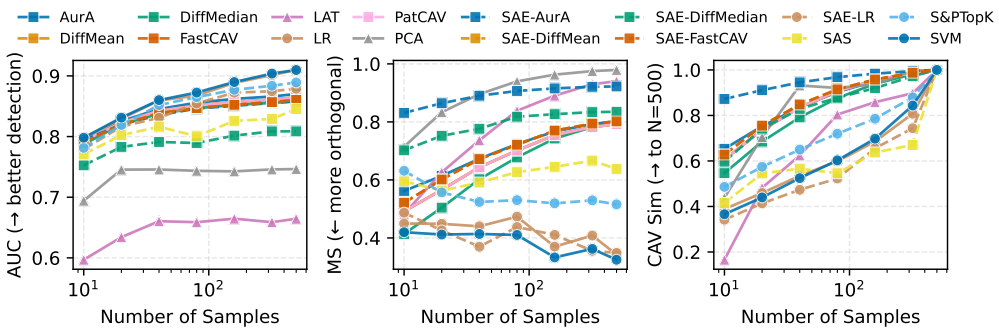

Figure 2: **Sample efficiency analysis on CelebA (SigLIP).** We report AUC (left), MS (middle), and Cosine Similarity to the CAV trained on $N = 500$ (right) as a function of training sample size $N \in \{10, \ldots, 500\}$. Notably, • methods show a decrease in MS as N increases, indicating a more disentangled representation. See Appendix E for additional backbones and datasets results.

Finding 5: Filtering of noisy neurons is essential to recover utility from SAE representations. While methods like SAE-AurA and S&PTopK are justifiable for isolating concepts in complex datasets, they are unnecessary in controlled regimes and add only training overhead.

It is known that directly substituting standard representations h with SAE features typically degrades performance (Kantamneni et al., 2025). However, when applying rigorous filtering strategies such as S&PTopK, SAE methods consistently demonstrate strong separation and high concept disentanglement, often leading to lower CD (Figure 8). While SAE-AurA scores often worse on steering disparity (Table 2), it reduces side effects on real and synthetic data (see Appendix D). Despite the SAE method's high computational overhead and non-trivial optimization, which often yield inconsistent results across backbones and datasets depending on SAE quality, methods like S&PTopK and SAE-AurA achieve high concept detection with low collateral damage on complex datasets. On sets like ImageNet-W/C, baseline methods consistently outperform across all metrics.

Ablations on different vision backbones. We identify SigLIP as the superior backbone for concept detection and collateral damage, matching or outperforming CLIP and DINOv2 on CelebA and ISIC, as confirmed by critical difference plots in Figures 18 and 19. However, DINOv2 offers distinct strengths: it exhibits robustness to watermarking (Table 10) and superior separation performance (Figure 18). This difference was also observed by Tong et al. (2024): while SigLIP is trained to align with text-describable semantic concepts, DINOv2 learns to capture more structural features.

Ablations on the sample-efficiency of CAVs. Figure 2 shows that the optimal number of samples for training CAVs varies by method group. Statistical methods (■) and LAT saturate rapidly, typically stabilizing around $N = 300$, whereas optimization-based (•) continue to improve both AUC and MS, possibly even over $N = 500$, on complex datasets like CelebA. For controlled synthetic domains, saturation occurs much earlier (see Appendix E); on ImageNet-W, statistical methods converge as early as $N = 50$–100, whereas optimization-based prove constant CAV optimization.

5 Conclusion

As image representations become central to real-world applications, reliably steering their behavior is paramount. We introduce SwordBench to unify and extend the field's fragmented steering and evaluation methods. By rigorously evaluating concept activation vectors across visual tasks, our benchmark exposes steering's overlooked second-order effects, insights that existing metrics miss.

Limitations. SwordBench is inherently constrained by the limitations of datasets, some of which suffer from low data quality (e.g., Waterbirds), lack of high-stakes tasks (e.g., CelebA), and sparse concept vocabulary with limited data (e.g., ISIC). While we address this by aggregating findings across a diverse set of tasks, future work should propose controlled datasets that more closely reflect real-world complexity while preserving counterfactual data structure. To focus on 16 CAV methods across several dimensions and model architectures, we leave multi-concept and multi-layer/head steering tasks for future work. In Appendix H we further discuss societal impacts of this work.

AI USE STATEMENT

We used generative AI tools for writing assistance and for coding assistance on auxiliary parts of the codebase, including plotting, data-processing scripts, and refactoring. We did not use generative AI tools for research ideation, and experimental design. All AI-assisted code was reviewed and tested for correctness by at least two authors.

ETHICS STATEMENT

This work introduces an evaluation benchmark and involves no human subjects, user studies, or newly collected data. All datasets are publicly available and used under their original terms. Sources and download links are given in Appendix C. We additionally elaborate on positive and negative societal impacts in Appendix H, and we report negative results prominently to discourage selective benchmarking.

REPRODUCIBILITY STATEMENT

Our benchmark is designed for reproducibility, and we documented every component needed to rerun it. The problem setup and intervention are formalized in Section 3.1, with notation in Appendix A and the justification for orthogonalization as the benchmark intervention in Appendix B.2. All 16 CAV extraction methods are defined in Section 3.2, with complete hyperparameter grids, cross-validation protocols, and selection criteria in Appendix C.5. Metric definitions, ranges, and optimal values are given in Section 3.3 and Appendix A.2. Dataset usage is detailed in Appendix C.1, including splits, concept distributions, stratified sampling strategies, the watermark and corruption pipelines, and our exclusion criteria. Downstream classifier architectures, the training sweep, and model selection are described in Appendix C.2, and SAE architecture, loss, training data, normalization, and hyperparameter grid in Appendix C.6, with reconstruction quality and sparsity reported in Table 13. An anonymized repository, including CAV extraction and scripts reproducing all tables and figures, is provided as supplementary material.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR*, 2017.

Hubert Baniecki, Maximilian Muschalik, Fabian Fumagalli, Barbara Hammer, Eyke Hüllermeier, and Przemyslaw Biecek. Explaining similarity in vision-language encoders with weighted Banzhaf interactions. In *NeurIPS*, 2025.

Antonio Barbalau, Cristian Daniel Paduraru, Teodor Poncu, Alexandru Tifrea, and Elena Burceanu. Rethinking sparse autoencoders: Select-and-project for fairness and control from encoder features alone. In *NeurIPS Workshop on Mechanistic Interpretability*, 2025.

Dilyara Bareeva, Maximilian Dreyer, Frederik Pahde, Wojciech Samek, and Sebastian Lapuschkin. Reactive model correction: Mitigating harm to task-relevant features via conditional bias suppression. In *CVPR Workshops*, 2024.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. In *COLM*, 2025.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *NeurIPS*, 2023.

Jan Betley, Daniel Chee Hian Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs. In *ICML*, 2025.

Usha Bhalla, Suraj Srinivas, Asma Ghandeharioun, and Himabindu Lakkaraju. Towards unifying interpretability and control: Evaluation via intervention. *arXiv preprint arXiv:2411.04430*, 2024.

Alceu Bissoto, Eduardo Valle, and Sandra Avila. Debiasing skin lesion datasets and models? Not so fast. In *CVPR Workshops*, 2020.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning, October 2023. Transformer Circuits Thread.

Gerasimos Chatzoudis, Zhuowei Li, Gemma E Moran, Hao Wang, and Dimitris N Metaxas. Visual sparse steering: Improving zero-shot image classification with sparsity guided steering vectors. *arXiv preprint arXiv:2506.01247*, 2025.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of bert's attention. In *EMNLP Workshop BlackboxNLP*, 2019.

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

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba E. Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. In *NeurIPS*, 2025.

Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. In *NeurIPS*, 2022.

Bartosz Cywiński and Kamil Deja. SAeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. In *ICML*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Maximilian Dreyer, Frederik Pahde, Christopher J Anders, Wojciech Samek, and Sebastian Lapuschkin. From hope to safety: Unlearning biases of deep models via gradient penalization in latent space. In *AAAI*, 2024.

Maximilian Dreyer, Lorenz Hufe, Jim Berend, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. From what to how: Attributing clip's latent components reveals unexpected semantic reliance. *arXiv preprint arXiv:2505.20229*, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *ICLR*, 2025.

Eren Erogullari, Sebastian Lapuschkin, Wojciech Samek, and Frederik Pahde. Post-hoc concept disentanglement: From correlated to isolated concept representations. In *xAI*, 2025.

Thomas Fel, Victor Boutin, Louis Béthune, Remi Cadene, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *NeurIPS*, 2023a.

Thomas Fel, Agustin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept recursive activation factorization for explainability. In *CVPR*, 2023b.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *ICLR*, 2025.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

Gemma Team. Gemma 3 technical report. *arXiv preprint, arXiv:2503.19786*, 2025.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *NeurIPS*, 2019.

Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87:96–110, 2014.

Anna Hedström, Salim I. Amoukou, Tom Bewley, Saumitra Mishra, and Manuela Veloso. To steer or not to steer? Mechanistic error reduction with abstention for language models. In *ICML*, 2025.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *ICLR*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? A case study in sparse probing. In *ICML*, 2025.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *ICML*, 2018.

Laura Kopf, Philine L Bommer, Anna Hedström, Sebastian Lapuschkin, Marina Höhne, and Kirill Bykov. CoSy: Evaluating textual explanations of neurons. In *NeurIPS*, 2024.

Laura Kopf, Nils Feldhus, Kirill Bykov, Philine Lou Bommer, Anna Hedström, Marina M. C. Höhne, and Oliver Eberle. Capturing polysemanticity with PRISM: A multi-concept feature description framework. In *NeurIPS*, 2025.

Patrick Leask, Bart Bussmann, Michael T Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9ca9eHNrdH.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *EMNLP*, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*, volume 36, 2023a.

Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. In *CVPR*, 2023b.

Weixin Liang and James Zou. MetaShift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. In *ICLR*, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *COLM*, 2024.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *EMNLP Workshop BlackboxNLP*, 2023.

Ivica Obadic, Alex Levering, Lars Pennig, Dario Oliveira, Diego Marcos, and Xiaoxiang Zhu. Contrastive pretraining for visual concept explanations of socioeconomic outcomes. In *CVPR*, 2024.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders. *ICML Workshop on Reliable and Responsible Foundation Models*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *NeurIPS*, 2025.

Frederik Pahde, Maximilian Dreyer, Wojciech Samek, and Sebastian Lapuschkin. Reveal to revise: An explainable AI life cycle for iterative bias correction of deep models. In *MICCAI*, 2023.

Frederik Pahde, Maximilian Dreyer, Moritz Weckbecker, Leander Weber, Christopher J. Anders, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. Navigating neural space: Revisiting concept activation vectors to overcome directional divergence. In *ICLR*, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*, 2024.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M. Ponti, and Shay B. Cohen. Spectral editing of activations for large language model alignment. In *NeurIPS*, 2024.

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

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *ACL*, 2020.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In *ICML*, 2022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *ACL*, 2024.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*, 2020.

Laines Schmalwasser, Niklas Penzel, Joachim Denzler, and Julia Niebling. FastCAV: Efficient computation of concept activation vectors for explaining deep neural networks. In *ICML*, 2025.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 2022.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

Stefan Smeu, Elena Burceanu, Emanuela Haller, and Andrei Liviu Nicolicioiu. Robust novelty detection through style-conscious feature ranking. In *WACV*, 2025.

Samuel Soo, Wesley Teng, Chandrasekaran Balaganesh, Tan Guoxian, and Ming YAN. Interpretable steering of large language models with feature guided activation additions. In *ICLR Workshop on Building Trust in Language Models and Applications*, 2025.

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodriguez. Whispering experts: Neural interventions for toxicity mitigation in language models. In *ICML*, 2024.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *ACL*, 2022.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. In *NeurIPS*, 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, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, 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, May 2024. Transformer Circuits Thread.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. In *ICLR Workshop on Reasoning and Planning for Large Language Models*, 2025.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. *California Institute of Technology*, 2011.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *ACL*, 2024a.

Qizhou Wang, Yong Lin, Yongqiang Chen, Ludwig Schmidt, Bo Han, and Tong Zhang. A sober look at the robustness of clips to spurious features. *NeurIPS*, 37:122484–122523, 2024b.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *NeurIPS*, 2023.

Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. In *ICML*, 2023.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. AxBench: Steering LLMs? Even simple baselines outperform sparse autoencoders. In *ICML*, 2025.

Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting CLIP with hierarchical sparse autoencoders. In *ICML*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

APPENDIX FOR "SWORDBENCH: EVALUATING ORTHOGONALITY OF STEERING IMAGE REPRESENTATIONS"

A Notation and Definitions **16**

A.1 Notation . . . 16

A.2 Metrics Definitions . . . 17

B Supplementary Related Work **18**

B.1 Concept Steering: Representations, SAEs, and Bias Mitigation . . . 18

B.2 Steering Intervention . . . 18

B.3 Mean Activation Difference (MAD) . . . 19

B.4 The equivalence of FastCAV, PatCAV, and DiffMean . . . 19

C Experimental Setup and Training Details **21**

C.1 Details of the Datasets/Tasks . . . 21

C.2 Downstream Model Training . . . 28

C.3 Choice of the Steering Layer . . . 29

C.4 Steering Magnitude Ablation . . . 31

C.5 CAV Training Details . . . 32

C.6 SAE Implementation Details . . . 33

C.7 Compute Resources . . . 35

D Extended Metric Results **36**

D.1 Counteranimals Results . . . 43

D.2 Critical Difference Analysis . . . 44

E Extended Ablation on Sample Efficiency **45**

F Performance Comparison Between Linear and Nonlinear Methods **48**

G Extended Metrics Analysis **50**

H Broader Impact **53**

A Notation and Definitions

A.1 Notation

Table 3: Main notations used in the paper.

Notation	Meaning
	Models & representations
ϕ	Frozen vision model backbone (e.g., CLIP, DINOv2)
h	Activation vector (representation) from the final layer of ϕ, $h \in \mathbb{R}^d$
ψ	Downstream linear classifier
d	Dimensionality of the representation space
	Datasets & concepts
$\mathbb{A}_c, \mathbb{A}_{\bar{c}}$	Sets of samples where the concept is present ($y = 1$) and absent ($y = 0$)
N	Number of samples per balanced set
$\mathbf{x}$	A clean sample and its concept-infused variant (synthetic data)
y	Binary concept label
	Steering & SAEs
$\mathbf{v}_c$	Concept activation vector (CAV)
$\bar{h}$	Steered representation (after removal/orthogonalization)
z	Sparse latent feature from Sparse Autoencoder (SAE)
$W_{\text{enc}}, W_{\text{dec}}$	SAE Encoder and Decoder weight matrices
$\mathcal{S}$	Set of selected Top-K feature indices (for S&PTopK)
	Vector evaluation metrics
AUC	Detection Accuracy (Linear Separability)
MAD	Mean Activation Difference
MS	Maximum Similarity (Orthogonality/Disentanglement)
CCR	Cross-Concept Robustness (Functional Independence)
	Downstream steering metrics
Acc	Classification accuracy
CD	Collateral Damage (Safety metric on $\mathbb{A}_{\bar{c}}$)
SD	Steering Disparity (Impact metric on counterfactual pairs $\mathbb{A}_c$)
F1	Harmonic mean of precision and recall

A.2 METRICS DEFINITIONS

Table 4: Definition of metrics used in the paper.

Metric	Captures	Range	Optimal
$\mathrm{AUC}(v_c) = \frac{1}{\lvert A_{\bar{c}}\rvert\lvert A_c\rvert} \sum_{h_{\bar{c}} \in A_{\bar{c}}} \sum_{h_c \in A_c} I[v_c^T h_c > v_c^T h_{\bar{c}}]$	separability of projections	$[0, 1]$	1
$\mathrm{MS}(v_c) = \max_{j \neq c} v_c^T v_j$	worst-case cosine sim. to another CAV	$[-1, 1]$	0
$\mathrm{CCR}(v_c) = \min_{j \neq c} \frac{\mathrm{AUC}(v_c \mid h_{c \perp j})}{\mathrm{AUC}(v_c \mid h_c)}$	worst-case AUC relative change after orthogonalization	≥ 0	0
$\mathrm{MAD}(A_{\bar{c}}, A_c) = \frac{\frac{1}{\lvert A_c\rvert} \sum_{h \in A_c} v_c^T h - \frac{1}{\lvert A_{\bar{c}}\rvert} \sum_{h \in A_{\bar{c}}} v_c^T h}{\sqrt{\frac{1}{\lvert A_{\bar{c}}\rvert - 1} \sum_{h \in A_{\bar{c}}} (v_c^T h - \mu_0)^2}}$	standardized projection gap (redundant with AUC, see App. B.3)	≥ 0	high
$\mathrm{F1}(y, \hat{y}) = \frac{2 \sum_i I(y_i = 1 \wedge \hat{y}_i = 1)}{\sum_i I(y_i = 1) + \sum_i I(\hat{y}_i = 1)}$	downstream detection (Youden threshold)	$[0, 1]$	1
$\mathrm{CD}(v_c) = \mathrm{Acc}(\psi(h) \mid h \in A_{\bar{c}}) - \mathrm{Acc}(\psi(\bar{h}) \mid h \in A_{\bar{c}})$	accuracy drop on concept-absent samples	$[-1, 1]$	0
$\mathrm{SD}(v_c) = \frac{\mathrm{Acc}(\psi(h) \mid h \in A_{\bar{c}}) - \mathrm{Acc}(\psi(\bar{h}) \mid h \in A_c)}{\mathrm{Acc}(\psi(h) \mid h \in A_{\bar{c}}) - \mathrm{Acc}(\psi(h) \mid h \in A_c)}$	normalized gap on counterfactual pairs	≥ 0	0

B SUPPLEMENTARY RELATED WORK

B.1 CONCEPT STEERING: REPRESENTATIONS, SAES, AND BIAS MITIGATION

Section 2 summarizes the lines of work that SWORDBENCH builds on. In this section, we expand on the topic of related works in the field of steering.

Concept activation vectors. Concept-based explanations gained significant prominence with the formalism of the concept activation vector (Kim et al., 2018). This framework builds upon the *linear representation hypothesis* (Nanda et al., 2023; Park et al., 2024), which posits that semantic concepts can be encoded as linear directions within the intermediate model activations. While typically these directions are identified using supervised linear probes (Alain & Bengio, 2017), various alternative extraction methods have since been proposed (Fel et al., 2023a; Pahde et al., 2025; Schmalwasser et al., 2025). Although the universality of this hypothesis is debated (Crabbé & van der Schaar, 2022; Engels et al., 2025), it remains the foundational assumption for steering across vision (Ghorbani et al., 2019; Obadic et al., 2024) and language (Clark et al., 2019; Belrose et al., 2023). We empirically verify this hypothesis by comparing the performance of linear CAVs with that of non-linear probes.

Superposition and sparse autoencoders. A critical challenge of interpretability is *superposition* (Elhage et al., 2022) and *polysemanticity* (Bricken et al., 2023; Kopf et al., 2025), in which individual neurons in representations simultaneously activate for multiple, unrelated concepts. This entanglement suggests that standard linear steering may inadvertently affect unrelated concepts. To address this, recent research on sparse autoencoders (SAEs, Templeton et al., 2024; Gao et al., 2025; Zaigrajew et al., 2025; Pach et al., 2025) aims to decompose dense representations into *monosemantic*, sparse features without supervision. While these disentangled features should yield 'cleaner' CAVs than simple linear probes, empirical evidence suggests this is not often the case (Bhalla et al., 2024; Kantamneni et al., 2025). SWORDBENCH evaluates a range of SAE-based steering methods across both standard CAVs (e.g. FastCAV, Schmalwasser et al., 2025) applied to SAE representations, as well as recent SAE-specific algorithms (Bayat et al., 2025) such as S&PTopK (Barbalau et al., 2025).

Steering and bias mitigation. While CAVs began as tools for explanation, the field has shifted toward interventional methods (Geiger et al., 2025). In this paradigm, vectors are used interventionally, i.e., added to or subtracted from activation, to *causally* alter the model's behavior (Subramani et al., 2022; Zou et al., 2023; Cywiński & Deja, 2025). In the language domain, steering is widely used to manipulate behavioral outputs, such as refusal rates or writing style (O'Brien et al., 2025; Rimsky et al., 2024; Venhoff et al., 2025; Hedström et al., 2025; Wu et al., 2025). However, in the vision domain, a critical application is *bias removal* (Dreyer et al., 2024; Bareeva et al., 2024; Pahde et al., 2025; Erogullari et al., 2025), where sensitive attributes, e.g,. race, are suppressed to improve the model. SWORDBENCH unifies these perspectives by evaluating the bias removal task, which is equivalent to the *concept erasure* steering task used in the language domain (Belrose et al., 2023).

B.2 STEERING INTERVENTION

After extracting a concept vector $\mathbf{v}_c$, steering interventions are performed by adding $\mathbf{v}_c$ to the original representation h with a scaling parameter α (Rimsky et al., 2024; Soo et al., 2025; Wu et al., 2025):

$$\bar{h} = h + \alpha \mathbf{v}_c. \tag{5}$$

Here, α takes a negative value for concept removal and a positive one for addition. Since excessive steering can degrade model performance (O'Brien et al., 2025), α must often be tuned for a specific concept task. How to tune is an open research question (Hedström et al., 2025). Moreover, **this optimization becomes non-trivial when a single vector is evaluated across multiple metrics**: one must either report results with α optimized separately for each metric or aggregate them through a principled heuristic. Neither approach has a clear canonical preference to apply for the fair comparison. While AxBench (Wu et al., 2025) aggregates metrics using a harmonic mean, we argue this may lead to biased optimization. Therefore, we focus on the concept erasure task via orthogonalization, eliminating the need to tune steering parameters.

In concept erasure, the objective is to remove specific information from the representation space. INLP (Ravfogel et al., 2020) observes that simple orthogonalization often fails to fully eliminate a concept, motivating more sophisticated approaches. They propose an iterative procedure that repeats orthogonalization until a linear classifier can no longer distinguish the concept from the activation space. RLACE (Ravfogel et al., 2022) later framed this as a linear minimax game solved via convex relaxation. However, because CAVs are frequently correlated with other concepts (exhibiting non-zero MS), such iterative procedures may inflict substantial collateral damage to unrelated features while aggressively maximizing concept removal. Notably, these procedures require optimizing the number of iterations for each separate CAV and task, which hinders fair benchmarking. Given that simple orthogonalization is often sufficient on controlled synthetic datasets, we do not use INLP in the SWORDBENCH. Furthermore, we omit LEACE (Belrose et al., 2023), which applies an affine transformation (centering and whitening) before projection, because the CAV methods are often evaluated on the raw rather than transformed activations. Further, estimating covariance requires a significant number of samples, which is not feasible in our benchmarking procedures, where $N \approx p$. Applying such transformations might unfairly distort or compromise method performance.

We therefore use simple orthogonalization as our primary benchmark steering intervention. We emphasize that our goal is not to optimize for the best possible steering performance, but rather to compare existing methods on equal footing and present several findings, as presented in Section 4.

B.3 MEAN ACTIVATION DIFFERENCE (MAD)

To move beyond ranking metrics such as AUC, we adopt MAD from neuron interpretability literature (Kopf et al., 2024). MAD quantifies the standardized mean projection score between positive ($\mathbb{A}_c$) and negative ($\mathbb{A}_{\bar{c}}$) populations. MAD is defined as:

$$\text{MAD}(\mathbb{A}_{\bar{c}}, \mathbb{A}_c) = \frac{\frac{1}{|\mathbb{A}_c|}\sum_{h\in\mathbb{A}_c}\mathbf{v}_c^T h - \frac{1}{|\mathbb{A}_{\bar{c}}|}\sum_{h\in\mathbb{A}_{\bar{c}}}\mathbf{v}_c^T h}{\sqrt{\frac{1}{|\mathbb{A}_{\bar{c}}|-1}\sum_{h\in\mathbb{A}_{\bar{c}}}(\mathbf{v}_c^T h - \mu_0)^2}}, \tag{6}$$

where $\mu_0 = \frac{1}{|\mathbb{A}_{\bar{c}}|}\sum_{h\in\mathbb{A}_{\bar{c}}}\mathbf{v}_c^T h$ is the mean activation of the negative group. Higher MAD values indicate a stronger signal-to-noise ratio for the concept direction.

However, in Figure 14 and further in Appendix G, we demonstrate that MAD correlates highly with AUC, questioning the necessity of MAD given that AUC is a more established metric.

B.4 THE EQUIVALENCE OF FASTCAV, PATCAV, AND DIFFMEAN

During preliminary results generation, we found that three CAV extraction methods—FastCAV, PatCAV, and DiffMean—produced similar results across all metrics. Notably, across all experiments, their CAVs have cosine similarities of over 0.99. As we detail further in this section, their mathematical definitions, under assumptions fulfilled in our experimental setup, are equal, and only numerical errors may introduce any difference between them.

Let us first recall that the DiffMean method can be formulated as follows:

$$\mathbf{v}_{\text{DiffMean}} = \frac{1}{N}\sum_{h\in\mathbb{A}_c} h - \frac{1}{N}\sum_{h\in\mathbb{A}_{\bar{c}}} h \tag{7}$$

A similar definition can be derived for FastCAV under the assumption that $|\mathbb{A}_{\bar{c}}| = |\mathbb{A}_c| = N$, which holds in our experimental setup. We start the derivation with the initial FastCAV definition:

$$\mathbf{v}_{\text{FastCAV}} = \frac{1}{N}\sum_{h\in\mathbb{A}_c}(h - \mu_{\text{global}}) = \frac{1}{N}\sum_{h\in\mathbb{A}_c} h - \frac{1}{2N}\sum_{h\in\mathbb{A}_{\bar{c}}\cup\mathbb{A}_c} h \tag{8}$$

$$= \frac{1}{N}\sum_{h\in\mathbb{A}_c} h - \frac{1}{2N}\sum_{h\in\mathbb{A}_{\bar{c}}} h - \frac{1}{2N}\sum_{h\in\mathbb{A}_c} h \tag{9}$$

$$= \frac{1}{2N}\sum_{h\in\mathbb{A}_c} h - \frac{1}{2N}\sum_{h\in\mathbb{A}_{\bar{c}}} h \tag{10}$$

$$= \frac{1}{2}\left(\frac{1}{N}\sum_{h\in\mathbb{A}_c} h - \frac{1}{N}\sum_{h\in\mathbb{A}_{\bar{c}}} h\right) \tag{11}$$

which is the same as the DiffMean up to the $\frac{1}{2}$ scalar, which can be omitted since all CAVs are normalized.

A similar derivation for PatCAV is described in (Pahde et al., 2025, appendix B.3), where the important assumption is that the concept labels are binary, which holds true throughout our experiments as well.

C EXPERIMENTAL SETUP AND TRAINING DETAILS

Below, we detail the datasets, model configurations (CAV, SAE, and downstream classifiers), and the specific hyperparameters used in our evaluation.

C.1 DETAILS OF THE DATASETS/TASKS

In this section, we detail the class and concept distributions for all datasets used in SWORDBENCH. Throughout this section, we bold concepts in the train and validation splits (unless stated otherwise) when the sample count is fewer than 500 in either set. Significantly low counts may indicate potential instability in CAV extraction.

C.1.1 ISIC

The ISIC dataset is sourced from the benchmark repository (Wu et al., 2023), which derives from the original ISIC 2018 Task 1 & 2 challenge (Codella et al., 2019) with artifact annotations provided in Bissoto et al. (2020). The authors manually annotated 8 artifacts (e.g., rulers, markers) and constructed five overlapping "trap" sets intended to simulate varying bias intensities. The original ISIC 2018 dataset is under the CC-BY-NC 4.0 License and is openly available via the link: https://drive.google.com/uc?id=1Os34EapIAJM34DrwZMw2rRRJij3HAUDV

While the creators of the trap sets claim distinct bias factors for each version, our empirical analysis indicated that Traps 1–5 exhibit statistically similar bias distributions and concept overlaps. Furthermore, upon investigating the splits, we observed significant overlap in the image data between the sets; consequently, we **evaluate only on Trap Set 1**. We report the distribution of the specific sample in Table 5. Following the protocol proposed by Wu et al. (2023), we strictly use the test split for all CAV evaluations to ensure validity, as the validation splits in this dataset are subsets of the training data.

We excluded concepts with fewer than 500 training samples, labeled as *Insufficient Data* in Table 5. We aggregate results over the remaining four attributes: *dark_corner*, *hair*, *gel_bubble*, and *ruler*.

Table 5: **ISIC (trap 1) dataset distribution**. Total samples: train (1,822), val (259), test (772). Counts below 500 in train or test split are **bold**. The confounded class column identifies the class negatively impacted in the test set due to correlation reversal (e.g., *ruler* is strongly correlated with *malignant* in the train set, but acts as a confounder for *benign* in the test set). The percent value denotes the fraction of samples in the given split that contain a concept. Note that concepts may co-occur, so percentages in columns do not sum up to 100%.

Image	Concept	Confounded class	Train data		Val data		Test data	
			benign	malignant	benign	malignant	benign	malignant
	**dark_corner**	benign	**386** (27%)	**223** (61%)	51 (25%)	29 (57%)	**340** (55%)	**10** (7%)
	**hair**	malignant	1,045 (72%)	**147** (41%)	153 (74%)	17 (33%)	**209** (34%)	**116** (75%)
	**gel_border**	*Insufficient Data*	**192** (13%)	**89** (24%)	34 (16%)	14 (28%)	**397** (64%)	**0** (0%)
	**gel_bubble**	malignant	847 (58%)	**128** (35%)	104 (50%)	17 (33%)	**164** (27%)	**129** (84%)
	**ruler**	benign	**446** (31%)	**295** (81%)	69 (33%)	45 (88%)	529 (86%)	**13** (8%)
	**ink**	*Insufficient Data*	**57** (4%)	**63** (17%)	7 (3%)	8 (16%)	**322** (52%)	**0** (0%)
	**patches**	*Insufficient Data*	**111** (8%)	**2** (1%)	21 (10%)	0 (0%)	**74** (12%)	**0** (0%)
	**vasc**	*Insufficient Data*	**246** (17%)	**109** (30%)	28 (14%)	16 (31%)	**90** (15%)	**49** (32%)

C.1.2 CELEBA

The CelebA dataset contains facial images with attribute annotations but lacks a designated downstream target. This dataset is available under a Non-Commercial Research Only agreement and can be loaded via `torchvision.datasets.CelebA`. Given that gender is a prevalent confounding factor (Table 6) and is a popular attribute to evaluate (Zaigrajew et al., 2025; Sagawa et al., 2020), we select *Male* as our primary classification target. Hence, we evaluate and average all vector and steering results over the remaining 39 attributes. Detailed statistics of the dataset are provided in Table 6.

Table 6: **CelebA dataset distribution**. Total samples: train $(162,770)$, val $(19,867)$, test $(19,962)$. Concepts with fewer than 500 samples in train or val splits are **bolded**. Overlap statistics report the conditional probability $P(\text{confounder} \mid \text{concept})$ in the training set. High overlap (e.g., Mustache and Male) highlights risks where CAVs may learn confounders rather than targets.

Concept	Train	Val	Test	$P(\textbf{conf} \mid \textbf{c})$	Confounding concept
Mustache	6,642 (4.1%)	1,003 (5.0%)	772 (3.9%)	99.96%	Male
5_o_Clock_Shadow	18,177 (11.2%)	2,345 (11.8%)	1,994 (10.0%)	99.91%	Male
Wearing_Lipstick	76,437 (47.0%)	8,860 (44.6%)	10,418 (52.2%)	99.91%	No_Beard
Heavy_Makeup	62,555 (38.4%)	7,751 (39.0%)	8,084 (40.5%)	99.91%	No_Beard
Goatee	10,337 (6.4%)	1,464 (7.4%)	915 (4.6%)	99.90%	Male
Sideburns	9,156 (5.6%)	1,367 (6.9%)	926 (4.6%)	99.90%	Male
Wearing_Necktie	11,890 (7.3%)	1,443 (7.3%)	1,399 (7.0%)	99.76%	Male
Bald	3,713 (2.3%)	**411** (2.1%)	423 (2.1%)	99.63%	Male
Rosy_Cheeks	10,525 (6.5%)	1,358 (6.8%)	1,432 (7.2%)	99.26%	No_Beard
Blond_Hair	24,267 (14.9%)	3,056 (15.4%)	2,660 (13.3%)	98.70%	No_Beard
Wearing_Earrings	30,362 (18.7%)	3,789 (19.1%)	4,125 (20.7%)	97.81%	No_Beard
Wearing_Necklace	19,764 (12.1%)	2,396 (12.1%)	2,753 (13.8%)	97.33%	No_Beard
Arched_Eyebrows	43,278 (26.6%)	5,134 (25.8%)	5,678 (28.4%)	95.96%	No_Beard
Bangs	24,685 (15.2%)	2,915 (14.7%)	3,109 (15.6%)	95.13%	No_Beard
Pale_Skin	7,005 (4.3%)	856 (4.3%)	840 (4.2%)	94.30%	No_Beard
Attractive	83,603 (51.4%)	10,332 (52.0%)	9,898 (49.6%)	93.19%	Young
Wavy_Hair	51,982 (31.9%)	5,495 (27.7%)	7,267 (36.4%)	92.18%	No_Beard
High_Cheekbones	73,645 (45.2%)	8,926 (44.9%)	9,618 (48.2%)	91.19%	No_Beard
Pointy_Nose	44,846 (27.6%)	5,660 (28.5%)	5,704 (28.6%)	89.29%	No_Beard
Brown_Hair	33,192 (20.4%)	4,793 (24.1%)	3,587 (18.0%)	89.22%	No_Beard
Double_Chin	7,571 (4.7%)	975 (4.9%)	913 (4.6%)	87.91%	Male
Smiling	78,080 (48.0%)	9,602 (48.3%)	9,987 (50.0%)	87.83%	No_Beard
Chubby	9,389 (5.8%)	1,216 (6.1%)	1,058 (5.3%)	87.65%	Male
Oval_Face	46,101 (28.3%)	5,565 (28.0%)	5,901 (29.6%)	87.14%	No_Beard
Mouth_Slightly_Open	78,486 (48.2%)	9,573 (48.2%)	9,883 (49.5%)	86.78%	No_Beard
Black_Hair	38,906 (23.9%)	4,144 (20.9%)	5,422 (27.2%)	86.42%	Young
Bushy_Eyebrows	23,386 (14.4%)	2,831 (14.3%)	2,586 (13.0%)	86.12%	Young
Young	126,788 (77.9%)	14,832 (74.7%)	15,114 (75.7%)	85.88%	No_Beard
Big_Lips	39,213 (24.1%)	3,044 (15.3%)	6,528 (32.7%)	85.37%	Young
Gray_Hair	6,896 (4.2%)	967 (4.9%)	636 (3.2%)	85.13%	Male
Straight_Hair	33,947 (20.9%)	4,085 (20.6%)	4,190 (21.0%)	85.09%	No_Beard
Narrow_Eyes	18,869 (11.6%)	1,492 (7.5%)	2,968 (14.9%)	83.25%	No_Beard
Blurry	8,362 (5.1%)	940 (4.7%)	1,010 (5.1%)	82.82%	No_Beard
No_Beard	135,779 (83.4%)	16,338 (82.2%)	17,041 (85.4%)	79.57%	Young
Eyeglasses	10,521 (6.5%)	1,383 (7.0%)	1,289 (6.5%)	79.42%	Male
Receding_Hairline	13,040 (8.0%)	1,429 (7.2%)	1,694 (8.5%)	76.48%	No_Beard
Big_Nose	38,341 (23.6%)	4,943 (24.9%)	4,232 (21.2%)	74.57%	Male
Bags_Under_Eyes	33,280 (20.5%)	4,121 (20.7%)	4,045 (20.3%)	72.77%	No_Beard
Wearing_Hat	8,039 (4.9%)	940 (4.7%)	839 (4.2%)	70.66%	Young
Male	68,261 (41.9%)	8,458 (42.6%)	7,715 (38.7%)	63.30%	Young

C.1.3 WATERBIRDS

Waterbirds Sagawa et al. (2020) is a modified version of the Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011), in which image backgrounds are replaced to create spurious correlations between binary bird classes (*landbird*, *waterbird*) and backgrounds (*land*, *water*). This creates rare groups: *landbird* on *water* and *waterbird* on *land*. Table 7 presents the distribution, showing that while the training data is highly skewed, the validation and test splits balance backgrounds within each class to rigorously test robustness. The dataset is available from the Stanford DRO repository under CC BY 4.0 license at https://downloads.cs.stanford.edu/nlp/data/dro/waterbird_complete95_forest2water2.tar.gz.

Stratified sampling strategy. For Waterbirds, we use stratified sampling to ensure CAVs capture the background concept rather than the bird species. For the *land* concept, positives are drawn from the majority class (*landbird* on *land*) and negatives from the same bird class but with the opposite background (*landbird* on *water*). Similarly, for *water*, we contrast *waterbird* on *water* (positive) vs. *waterbird* on *land* (negative). Our preliminary experiments confirmed that naive random sampling produced CAVs aligned with bird features rather than the environment. To sum up, we investigate the CAV behavior when negative samples are drawn exclusively from the opposing bias group.

Table 7: **Waterbirds dataset distribution**. Total samples: train $(4,795)$, val $(1,199)$, test $(5,794)$. Concepts with fewer than 500 samples in train or val splits are **bolded**. In contrast to the balanced validation and test sets, the training set exhibits extreme imbalance.

Image	Concept	Confounded class	Train data	Val data	Test data
	**Water** (with landbird)	landbird	**184** (5%)	**466** (50%)	2,255 (50%)
	**Land** (with landbird)	—	3,498 (95%)	**467** (50%)	2,255 (50%)
	**Water** (with waterbird)	—	1,057 (95%)	**133** (50%)	642 (50%)
	**Land** (with waterbird)	waterbird	**56** (5%)	**133** (50%)	642 (50%)

C.1.4 METASHIFT

This dataset was created with detecting data shift in mind (Liang & Zou, 2022). It contains 410 classes (e.g., *cat*, *dog*, *bus*, *horse*) split into co-occurring objects in the image. For example, the *cat* class is split into cats with *keyboard*, *sink*, or *box*. We decided to exclude this data from our



Figure 3: **Examples of images from the MetaShift dataset.** Many of the samples are incorrectly labeled for both the class (*dog* or *car*) and the corresponding concept (*shelf*, *sofa*, *bed*, and *bike*). Here, the first two images are labeled as dogs with a concept *shelf* and *bike*, respectively. Another two images are labeled as cats with a concept *sofa* and *bed*, respectively. In addition to label mismatches, some images do not contain any of the concepts or classes claimed to be present.

benchmark due to multiple instances of mislabeled data we found during manual exploration of the dataset. For example, images presented in Figure 3 show images that lack either the class (e.g., *dog*) or the co-occurring concept (e.g., *bike*). We acknowledge that we restricted our analysis to a subset of this dataset containing only cat and dog images. The dataset is available from https://github.com/Weixin-Liang/MetaShift under the MIT License.

C.1.5 COUNTERANIMAL

The Counteranimal dataset (Wang et al., 2024b) was introduced as an additional validation for models from the CLIP family that highlights their biases and limitations not previously detected on datasets such as ImageNet (Deng et al., 2009) or LAION (Schuhmann et al., 2022). It contains images of animals collected from the iNaturalist application, notably after manually filtering for low-quality or ambiguous-label images. The dataset contains 45 animals, whose labels align with ImageNet-1k classes, and is split into two parts: *common* and *uncommon*. The *common* split contains images of animals in their typical habitat. Consequently, *uncommon* split covers images of animals with an uncommon background, e.g., polar bears on the grass. Figure 4 shows an example. The authors verified that CLIP models perform decent classification in a zero-shot setup on *common* split, but they also observed a significant drop in performance on *uncommon* split, which highlights that CLIP bases its prediction partially on the context – a background in the image – rather than on the object in the image itself. The benchmark results on this dataset are in Appendix D.1. The dataset is available at https://counteranimal.github.io/ under an unknown license.



Figure 4: **Example of Counteranimal dataset.** An animal is presented in its common environment (left) or with an atypical background (right).

Sampling strategy. We use the *common* split of Counteranimal dataset as a train and validation set in our experiments, with a random split between those in a ratio of 9:1. As a test set, we use all images from *uncommon* split. This approach leverages the strengthening of the background bias in both the training and validation sets, which can later be removed using debiasing methods on the test set. The full statistics are provided in Table 9. Because the background labels are semantically similar (e.g., grass, green, forest), we decided to merge some labels as shown in Table 8. Further, we discarded classes that represent background types that cannot be explained by a more general group.

Table 8: **Counteranimal dataset concept grouping.**

grouped concept	original concepts
water	above water, underwater, water
ground	earth, ground, rock, sand
green	grass, green, shrub, tree, tree and grass
snow	snow, white background
sky	white or blue, sky
not_water	not in water, without water

Table 9: **Counteranimal dataset distribution**. Counts for each class across splits, along with environment group annotations. Total samples: 10,774 (Train: 5,485; Validation: 610; Test: 4,679).

Class	train data		validation data		test data	
	count	group	count	group	count	group
African crocodile	82	ground	9	ground	84	green
Arctic fox	111	snow	12	snow	238	green
Vulture	132	sky	15	sky	98	green
agama	304	ground	34	ground	142	green
beaver	93	water	10	water	100	green
bighorn	85	green	9	green	95	ground
black stork	73	green	8	green	149	sky
black swan	184	water	20	water	106	ground
box turtle	58	green	7	green	200	ground
brambling	105	green	12	green	111	sky
bulbul	110	sky	12	sky	185	green
bullfrog	247	water	27	water	158	not_water
centipede	55	snow	6	snow	104	green
cicada	130	green	14	green	88	other
common iguana	45	ground	5	ground	120	green
dung beetle	58	ground	7	ground	92	other
flamingo	177	water	20	water	101	sky
garter snake	70	green	8	green	249	ground
harvestman	451	green	50	green	125	ground
hognose snake	183	ground	20	ground	123	green
hyena	311	green	35	green	100	other
ice bear	112	snow	13	snow	109	green
jaguar	59	water	6	water	226	green
king snake	205	ground	23	ground	98	green
loggerhead	198	water	22	water	91	not_water
mink	69	green	8	green	101	water
ostrich	185	ground	21	ground	113	water
otter	117	water	13	water	102	green
pelican	209	water	23	water	98	sky
prairie chicken	233	green	26	green	87	snow
ptarmigan	51	snow	6	snow	107	green
red fox	129	green	14	green	105	other
sea lion	52	ground	6	ground	94	water
tarantula	208	ground	23	ground	158	green
water ouzel	234	water	26	water	159	not_water
water snake	136	water	15	water	163	ground
whiptail	224	ground	25	ground	100	other

C.1.6 IMAGENET-W (WATERMARK)

We use the ImageNet-1k dataset (Deng et al., 2009) from Hugging Face `ILSVRC/imagenet-1k` (ImageNet Agreement), sampling 10 diverse classes following Baniecki et al. (2025) to optimize computational efficiency without sacrificing evaluation depth. Our ImageNet-W dataset features a balanced distribution across 10 distinct classes: tractor, goldfish, cat, husky, banana, pizza, plane, ball, church, and iPod. It contains 11700 training samples, 1300 validation samples, and 500 test samples. Each individual class is uniformly represented by 1,170 training images, 130 validation images, and 50 test images, with these per-class quantities corresponding exactly to 10% of the official dataset split. To ensure robust evaluation, we repartition the official splits as follows: (i) Test set: The official ImageNet Validation split (50 images per class). (ii) The official training split is partitioned into new training (90%) and validation (10%) sets per class. During downstream training, we augment each sample with its corresponding class watermark to increase the model's sensitivity to watermarks. The watermark augmentation pipeline, adapted from Li et al. (2023b), generates counterfactual pairs on the fly. The visual effect of this augmentation is presented in Figure 5.

Confounding watermarks. For downstream steering, we restrict our evaluation to high-degradation watermark-class pairs (Table 10), defined as pairs in which the watermark text causes a significant drop in classification accuracy, serving as a strong confounder. Notably, we find that text watermarks do not degrade DINOv2 performance (0% accuracy drop). We hypothesize this is due to DINOv2's vision-only self-supervised objective, which, unlike text-aligned models such as CLIP and SigLIP, does not learn to attend to text hints. Therefore, we report steering results primarily for the text-aligned models where the artifact poses a genuine safety risk.



Figure 5: **Example of watermarking in ImageNet-W**. *Left*: A clean image of a *Tractor*. *Right*: The same image infused with the watermark text `church`. The watermark overlays transparent text, which is clearly visible but does not obscure the underlying structural details of the object.

Table 10: **ImageNet-W confounding pairs**. The confounding watermark column indicates the specific text overlay used to induce spurious correlation. The *Degradation Accuracy Drop* reports the performance gap between clean and watermarked samples. DINOv2 is robust to watermarks.

Visual class	Confounding watermark (text)	Degradation accuracy drop		
		CLIP	DINOv2	SigLIP
cat	*husky*	42%	0%	16%
goldfish	*church*	24%	0%	12%

C.1.7 IMAGENET-C (CORRUPTIONS)

ImageNet-C shares a similar structure with ImageNet-W; however, rather than adding watermarks to the original ImageNet dataset, it applies one of 19 corruptions defined by Hendrycks & Dietterich (2019) and presented in Figure 7. Each corruption can be applied at five different severity levels, as visualized for three representative concepts in Figure 6. Given the consistent performance drop under *Shot Noise* and *Gaussian Noise* at severity 4 for both concepts (Figure 7), we evaluate downstream tasks across five seeds for both corruptions and present the detailed results in Appendix D. These accuracy drops remain consistent across various backbones for only those two corruptions.

We exclude severity level 5 from our evaluation, as the severe degradation of the original image often obscures the target object, rendering classification impractical.



Figure 6: **Example of corruptions in ImageNet-C.** We visualize three sample corruptions: *Fog*, *Frost*, and *Gaussian Noise*. Each row displays the corruption across increasing severity levels.

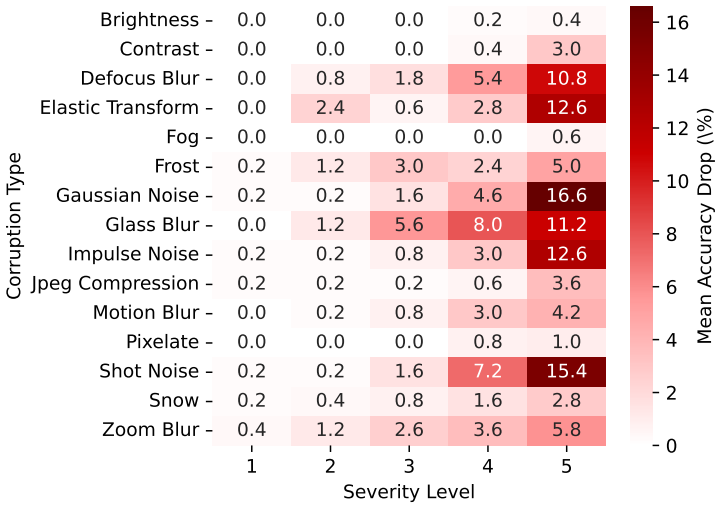

Figure 7: **Heatmap of corruptions $\times$ severity impact on accuracy drop for CLIP.** *Gaussian Noise* visibly degrades performance from severity 3, whereas *Shot Noise* degrades it from severity 4. The largest drop in accuracy is observed at severity level 5, but, as shown in Figure 6, we exclude this level from our experiments because it alters images too drastically compared to lower severities.

C.2 Downstream Model Training

We train lightweight classifiers ψ on top of the frozen backbone representations h to evaluate steering performance. We use the following openly available backbone models with default hyperparameters:

- CLIP ViT-L/14 (Radford et al., 2021): loaded via the clip Python library `https://github.com/openai/CLIP` `ViT-L/14` (MIT License)
- DINOv2 ViT-B/14 (Oquab et al., 2024): `facebookresearch/dinov2 | dinov2_vitb14`, loaded via PyTorch Hub (Apache License 2.0, CC-BY-4.0 License)
- SigLIP-2 ViT-B/16 (Zhai et al., 2023): `google/siglip2-base-patch16-224` loaded from Hugging Face (Apache License 2.0)

All downstream models are trained using the AdamW optimizer with a cosine decay learning rate schedule and a linear warmup of 2 epochs (initial $lr = 10^{-6}$). For every dataset, we train and evaluate two distinct architectures, selecting the one that maximizes validation performance:

- linear probe (LP): A single linear layer mapping representations directly to class logits.
- multi-layer perceptron (MLP): A two-layer network with ReLU activation and Dropout, designed to capture nonlinear decision boundaries. The hidden dimension d_{hidden} is set to the largest power of two less than the input dimension d (e.g., for a ViT-L/14 with $d = 768$, the hidden layer is set to 512).

We perform a grid search to select the optimal configuration for each dataset, choosing the model with the highest F1 score on the validation set. The number of epochs is tuned per dataset to ensure convergence: 4 for ImageNet-W, 8 for Waterbirds, and 25 for ISIC. The search space and fixed training parameters are detailed in Table 11.

Table 11: **Downstream training configuration.** We performed a full sweep over learning rates and weight decay for both architectures to ensure robust baselines.

Fixed parameters		**Hyperparameter grid**		
	Value		LP	MLP
optimizer	AdamW	learning rate	{1e-2, 1e-3, 5e-4}	{1e-2, 1e-3, 5e-4}
batch size	256	dropout	N/A	{0.0, 0.5}
scheduler	cosine + warmup	weight decay	{0.0, 1e-4}	{0.0, 1e-4}

C.3 Choice of the Steering Layer

In Section 3.1, we justified steering the global image representation, as models are trained to extract a global semantic representation of the whole image at the output, which also sidesteps the potential "Hydra effect" (McGrath et al., 2023). Prior work in vision follows this setup by choosing the penultimate layer, i.e., the layer before the classifier head, as in our setup (Pahde et al., 2025; Erogullari et al., 2025). To test this empirically in this work, we steer at $\{100, 83, 67, 50\}\%$ of the network depth, where 100% denotes the last transformer block before pooling or projection, and compare with the final output on all classes paired setup in ImageNet-W (Tables 12 to 14) and CelebA (Table 15).

On ImageNet-W with CLIP, 100% matches or improves on output SD, but from 83% downward disparity rises to $\approx$1 for all methods, even though SVM, DiffMean, and AurA still detect the concept at these layers, which is consistent with Hydra-style self-repair. On SigLIP, this collapse appears only at 50%, and intermediate layers can lower SD, but F1 drops sharply for all methods except SVM. On CelebA, performance at 100% is on par with the output layer, and AUC generally decreases at earlier layers (e.g., AurA on CLIP, .86 $\rightarrow$ .71 at 50%), while the ranking of methods by AUC and MS is preserved at every depth. CCR increases for weaker methods such as LAT, likely due to the degenerate case of weak vectors indicated by a low AUC score. Therefore, specific method–backbone combinations can benefit from other layers, but no layer is consistently better than the output layer across methods and backbones. Together with avoiding the collapse observed at intermediate layers, this motivates using the final output representation in our main analysis.

Table 12: **ImageNet-W layer ablation on CD (↓).** CAVs are extracted and applied at layers $\{24, 20, 16, 12\}$ of 24 for CLIP ViT-L/14 and $\{12, 10, 8, 6\}$ of 12 for SigLIP ViT-B/16; *output* denotes the final output representation used in the main paper. Values represent the mean $\pm$ two standard errors across 5 seeds, and we indicate the best depth within each backbone per row metric in **Bold** while the second best are underlined.

		CLIP					SigLIP				
Method	**Task**	output	100%	83%	67%	50%	output	100%	83%	67%	50%
LAT	Husky	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	.28$_{\pm.16}$	**.00**$_{\pm.00}$	2.0$_{\pm.64}$	.36$_{\pm.15}$
	Church	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	.20$_{\pm.00}$	**.00**$_{\pm.00}$	.72$_{\pm.52}$	.28$_{\pm.16}$
SVM	Husky	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$
	Church	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$
DiffMean	Husky	.40$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	1.12$_{\pm.10}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$
	Church	.20$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	.08$_{\pm.10}$	1.68$_{\pm.10}$	.04$_{\pm.08}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$
AurA	Husky	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$
	Church	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$

Table 13: **ImageNet-W layer ablation on SD (↓).** The setup is identical to that of Table 12.

		CLIP					SigLIP				
Method	**Task**	output	100%	83%	67%	50%	output	100%	83%	67%	50%
LAT	Husky	**.00**$_{\pm.00}$	**.00**$_{\pm.00}$	1.14$_{\pm.00}$	1.01$_{\pm.06}$	1.0$_{\pm.00}$	.04$_{\pm.06}$	.47$_{\pm.53}$	$-$**.01**$_{\pm.00}$	.24$_{\pm.11}$	1.0$_{\pm.51}$
	Church	**.00**$_{\pm.00}$	.07$_{\pm.03}$	1.07$_{\pm.03}$	0.77$_{\pm.23}$	.97$_{\pm.07}$	**.15**$_{\pm.00}$	**.15**$_{\pm.00}$	.25$_{\pm.08}$	.86$_{\pm.07}$	.93$_{\pm.14}$
SVM	Husky	.83$_{\pm.12}$	**.66**$_{\pm.24}$	1.00$_{\pm.00}$	0.97$_{\pm.02}$	1.0$_{\pm.00}$	.39$_{\pm.15}$	**.32**$_{\pm.06}$	.39$_{\pm.37}$	.54$_{\pm.23}$	1.1$_{\pm.23}$
	Church	**.43**$_{\pm.18}$	.57$_{\pm.23}$	1.13$_{\pm.10}$	1.22$_{\pm.04}$	1.0$_{\pm.00}$	.69$_{\pm.07}$	.66$_{\pm.00}$	**.56**$_{\pm.20}$	.69$_{\pm.13}$	1.1$_{\pm.20}$
DiffMean	Husky	.40$_{\pm.09}$	**.00**$_{\pm.00}$	0.93$_{\pm.02}$	1.02$_{\pm.04}$	.93$_{\pm.02}$	.11$_{\pm.00}$	1.6$_{\pm.00}$	$-$**.01**$_{\pm.00}$	$-$**.01**$_{\pm.00}$	.37$_{\pm.00}$
	Church	.17$_{\pm.00}$	**.00**$_{\pm.00}$	1.25$_{\pm.00}$	1.43$_{\pm.03}$	.88$_{\pm.04}$	.19$_{\pm.07}$	.66$_{\pm.00}$	$-$**.02**$_{\pm.00}$	.49$_{\pm.00}$	.63$_{\pm.07}$
AurA	Husky	.23$_{\pm.02}$	**.21**$_{\pm.02}$	1.05$_{\pm.00}$	1.05$_{\pm.00}$	1.0$_{\pm.00}$	**.11**$_{\pm.00}$	.24$_{\pm.00}$	.32$_{\pm.06}$	.75$_{\pm.00}$	.87$_{\pm.00}$
	Church	**.08**$_{\pm.00}$	.17$_{\pm.00}$	1.27$_{\pm.03}$	1.17$_{\pm.00}$	1.0$_{\pm.00}$	.49$_{\pm.00}$	**.22**$_{\pm.08}$	.66$_{\pm.00}$	.83$_{\pm.00}$	1.1$_{\pm.08}$

Table 14: **ImageNet-W layer ablation on F1 (↑).** The setup is identical to that of Table 12. Unlike the results in Table 2 and in Appendix D, the Youden-J threshold is fitted on the validation split rather than the training split, and F1 is reported on the test split.

		CLIP					SigLIP				
Method	**Task**	output	100%	83%	67%	50%	output	100%	83%	67%	50%
LAT	Husky	$.53_{\pm.02}$	$.53_{\pm.03}$	$.52_{\pm.04}$	$\mathbf{.56}_{\pm.23}$	$.38_{\pm.04}$	$\mathbf{.52}_{\pm.01}$	$.18_{\pm.00}$	$.21_{\pm.01}$	$.22_{\pm.01}$	$.20_{\pm.00}$
	Church	$.63_{\pm.01}$	$\mathbf{.64}_{\pm.02}$	$.55_{\pm.05}$	$.60_{\pm.29}$	$.63_{\pm.08}$	$\mathbf{.51}_{\pm.02}$	$.21_{\pm.04}$	$.23_{\pm.04}$	$.17_{\pm.00}$	$.17_{\pm.00}$
SVM	Husky	$.92_{\pm.03}$	$\mathbf{.95}_{\pm.01}$	$.94_{\pm.02}$	$.93_{\pm.02}$	$.89_{\pm.05}$	$\mathbf{.91}_{\pm.04}$	$.85_{\pm.06}$	$.83_{\pm.05}$	$\mathbf{.91}_{\pm.03}$	$.85_{\pm.05}$
	Church	$.96_{\pm.02}$	$.96_{\pm.04}$	$\mathbf{.97}_{\pm.01}$	$.95_{\pm.00}$	$.96_{\pm.02}$	$\mathbf{.97}_{\pm.00}$	$.91_{\pm.04}$	$.93_{\pm.04}$	$.95_{\pm.01}$	$.92_{\pm.04}$
DiffMean	Husky	$.89_{\pm.02}$	$.86_{\pm.02}$	$\mathbf{.93}_{\pm.00}$	$.87_{\pm.01}$	$.67_{\pm.03}$	$\mathbf{.90}_{\pm.01}$	$.20_{\pm.00}$	$.40_{\pm.02}$	$.36_{\pm.02}$	$.32_{\pm.01}$
	Church	$\mathbf{.97}_{\pm.01}$	$.94_{\pm.01}$	$.96_{\pm.00}$	$.96_{\pm.00}$	$.77_{\pm.01}$	$\mathbf{.84}_{\pm.02}$	$.19_{\pm.01}$	$.63_{\pm.04}$	$.56_{\pm.00}$	$.45_{\pm.01}$
AurA	Husky	$.85_{\pm.05}$	$.92_{\pm.00}$	$\mathbf{.94}_{\pm.00}$	$.90_{\pm.00}$	$.57_{\pm.06}$	$\mathbf{.93}_{\pm.03}$	$.39_{\pm.03}$	$.43_{\pm.01}$	$.27_{\pm.01}$	$.26_{\pm.00}$
	Church	$\mathbf{.99}_{\pm.00}$	$.94_{\pm.01}$	$.96_{\pm.00}$	$.95_{\pm.00}$	$.77_{\pm.01}$	$\mathbf{.90}_{\pm.02}$	$.47_{\pm.03}$	$.57_{\pm.01}$	$.49_{\pm.02}$	$.39_{\pm.01}$

Table 15: **CelebA layer ablation on vector metrics.** CAVs are extracted at $\{100, 83, 67, 50\}\%$ of the network depth, i.e., layers $\{24, 20, 16, 12\}$ of 24 for CLIP ViT-L/14 and $\{12, 10, 8, 6\}$ of 12 for DINOv2 ViT-B/14 and SigLIP ViT-B/16, and compared with the *output* representation used in the main paper. Values represent the mean $\pm$ two standard errors across concepts and we indicate the best depth within each backbone per row metric in **Bold** while the second best are underlined.

		CLIP					DINOv2					SigLIP				
Method	**Metric**	output	100%	83%	67%	50%	output	100%	83%	67%	50%	output	100%	83%	67%	50%
LAT	AUC (↑)	$\mathbf{.63}_{\pm.02}$	$.61_{\pm.03}$	$.56_{\pm.02}$	$.55_{\pm.02}$	$.55_{\pm.02}$	$\mathbf{.56}_{\pm.02}$	$\mathbf{.56}_{\pm.02}$	$\mathbf{.56}_{\pm.02}$	$\mathbf{.56}_{\pm.02}$	$.54_{\pm.01}$	$.66_{\pm.03}$	$\mathbf{.68}_{\pm.03}$	$\mathbf{.68}_{\pm.03}$	$.64_{\pm.03}$	$.52_{\pm.01}$
	MS (↓)	$.91_{\pm.02}$	$\mathbf{.89}_{\pm.02}$	$.96_{\pm.01}$	$.97_{\pm.01}$	$.99_{\pm.00}$	$.96_{\pm.01}$	$.96_{\pm.01}$	$.98_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{.94}_{\pm.03}$	$\mathbf{.94}_{\pm.02}$	$.95_{\pm.01}$	$.95_{\pm.02}$	$.97_{\pm.01}$	$.99_{\pm.00}$
	CCR (↑)	$.61_{\pm.05}$	$.62_{\pm.04}$	$.55_{\pm.05}$	$.73_{\pm.04}$	$\mathbf{.77}_{\pm.03}$	$.52_{\pm.07}$	$.51_{\pm.06}$	$.68_{\pm.05}$	$.69_{\pm.05}$	$\mathbf{.72}_{\pm.04}$	$.50_{\pm.05}$	$.52_{\pm.06}$	$.47_{\pm.06}$	$.54_{\pm.05}$	$\mathbf{.57}_{\pm.06}$
SVM	AUC (↑)	$\mathbf{.90}_{\pm.03}$	$\mathbf{.90}_{\pm.02}$	$\mathbf{.90}_{\pm.03}$	$.87_{\pm.03}$	$.84_{\pm.03}$	$\mathbf{.88}_{\pm.03}$	$\mathbf{.88}_{\pm.03}$	$.86_{\pm.03}$	$.84_{\pm.03}$	$.82_{\pm.03}$	$.91_{\pm.03}$	$.91_{\pm.02}$	$\mathbf{.92}_{\pm.02}$	$.91_{\pm.03}$	$.89_{\pm.03}$
	MS (↓)	$.35_{\pm.05}$	$\mathbf{.32}_{\pm.05}$	$.34_{\pm.05}$	$.34_{\pm.05}$	$.35_{\pm.05}$	$\mathbf{.17}_{\pm.03}$	$.20_{\pm.03}$	$.29_{\pm.05}$	$.26_{\pm.04}$	$.33_{\pm.05}$	$.32_{\pm.06}$	$.29_{\pm.05}$	$\mathbf{.24}_{\pm.04}$	$.34_{\pm.05}$	$.32_{\pm.05}$
	CCR (↑)	$\mathbf{.94}_{\pm.01}$	$.93_{\pm.02}$	$.93_{\pm.02}$	$.91_{\pm.02}$	$.86_{\pm.03}$	$\mathbf{.97}_{\pm.01}$	$\mathbf{.97}_{\pm.01}$	$.88_{\pm.03}$	$.90_{\pm.03}$	$.84_{\pm.03}$	$\mathbf{.94}_{\pm.02}$	$.92_{\pm.02}$	$.90_{\pm.03}$	$.90_{\pm.03}$	$.90_{\pm.03}$
DiffMean	AUC (↑)	$.85_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$.84_{\pm.03}$	$.76_{\pm.03}$	$.70_{\pm.03}$	$\mathbf{.83}_{\pm.03}$	$\mathbf{.83}_{\pm.03}$	$.75_{\pm.03}$	$.74_{\pm.03}$	$.76_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$.82_{\pm.03}$	$.79_{\pm.03}$
	MS (↓)	$.77_{\pm.06}$	$.76_{\pm.06}$	$.79_{\pm.05}$	$.76_{\pm.04}$	$\mathbf{.75}_{\pm.04}$	$.82_{\pm.05}$	$.83_{\pm.05}$	$.83_{\pm.04}$	$.80_{\pm.05}$	$\mathbf{.78}_{\pm.05}$	$\mathbf{.79}_{\pm.06}$	$\mathbf{.79}_{\pm.06}$	$.82_{\pm.06}$	$.83_{\pm.05}$	$.83_{\pm.05}$
	CCR (↑)	$\mathbf{.87}_{\pm.03}$	$\mathbf{.87}_{\pm.03}$	$.84_{\pm.03}$	$.84_{\pm.03}$	$\mathbf{.87}_{\pm.02}$	$.82_{\pm.03}$	$.82_{\pm.03}$	$.82_{\pm.03}$	$\mathbf{.84}_{\pm.03}$	$.83_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$.85_{\pm.03}$	$.81_{\pm.03}$
AurA	AUC (↑)	$\mathbf{.86}_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$.84_{\pm.03}$	$.76_{\pm.03}$	$.71_{\pm.03}$	$\mathbf{.83}_{\pm.03}$	$\mathbf{.83}_{\pm.03}$	$.76_{\pm.03}$	$.74_{\pm.03}$	$.76_{\pm.03}$	$\mathbf{.87}_{\pm.03}$	$.86_{\pm.03}$	$.84_{\pm.03}$	$.75_{\pm.03}$	$.68_{\pm.02}$
	MS (↓)	$.78_{\pm.05}$	$.80_{\pm.04}$	$.81_{\pm.04}$	$.78_{\pm.03}$	$\mathbf{.77}_{\pm.03}$	$.85_{\pm.04}$	$.85_{\pm.04}$	$.85_{\pm.03}$	$.82_{\pm.04}$	$\mathbf{.81}_{\pm.04}$	$\mathbf{.80}_{\pm.05}$	$.82_{\pm.04}$	$.84_{\pm.04}$	$.85_{\pm.05}$	$.84_{\pm.05}$
	CCR (↑)	$\mathbf{.86}_{\pm.03}$	$\mathbf{.86}_{\pm.03}$	$.83_{\pm.02}$	$.84_{\pm.02}$	$\mathbf{.86}_{\pm.02}$	$.81_{\pm.03}$	$.81_{\pm.03}$	$.82_{\pm.03}$	$\mathbf{.83}_{\pm.03}$	$\mathbf{.83}_{\pm.02}$	$.85_{\pm.03}$	$.86_{\pm.03}$	$.87_{\pm.03}$	$.91_{\pm.03}$	$\mathbf{.94}_{\pm.02}$

C.4 Steering Magnitude Ablation

We ablate the steering magnitude by introducing a scaling parameter α into the orthogonalization procedure:

$$\bar{h} = h - \alpha(\mathbf{v}_c^T h)\mathbf{v}_c.$$

The results are presented in Table 16, and Table 17. Smaller values of the α parameter lead to better CD, but worse SD. We note that there is no single α value that can be considered as an optimal for a trade-off between the steering effectiveness, which is captured by SD, and reducing the negative side-effects of steering, which is measured by CD. In the main experiments, we use $\alpha = 1.0$. We motivate this decision by the two factors. First, $\alpha = 1.0$ is the only value with a clear geometric interpretation: it fully removes the component of h along $\mathbf{v}_c$, following the standard procedure of linear concept removal (Ravfogel et al., 2020; Belrose et al., 2023; Pahde et al., 2025), whereas $\alpha < 1$ leaves residual concept signal and $\alpha > 1$ flips the sign of the concept projection. Second, it makes the intervention parameter-free. Tuning α would require selecting a value per method, concept, and metric, or aggregating metrics through a heuristic such as the harmonic mean used in AxBench (Wu et al., 2025), both of which would bias the comparison toward methods whose optimal α happens to align with the chosen criterion (see Appendix B.2 for deeper discussion).

Table 16: **Effect of orthogonalization strength on CelebA for Linear SVM.** We compute CD (↓) across different α magnitudes where $\alpha = 1$ is the full orthogonal projection used throughout the paper and $\alpha = 0$ is the uncorrected baseline. Values represent the mean ± two standard errors over 39 concepts.

Backbone	**α value**					
	0.00	0.25	0.50	0.75	1.00	1.25
CLIP	$.00_{\pm.00}$	$.00_{\pm.00}$	$.02_{\pm.02}$	$.08_{\pm.06}$	$.20_{\pm.21}$	$.47_{\pm.50}$
DINOv2	$.00_{\pm.00}$	$.00_{\pm.00}$	$.01_{\pm.01}$	$.01_{\pm.01}$	$.02_{\pm.02}$	$.03_{\pm.03}$
SigLIP	$.00_{\pm.00}$	$.02_{\pm.01}$	$.04_{\pm.04}$	$.12_{\pm.12}$	$.32_{\pm.37}$	$.85_{\pm1.1}$

Table 17: **Effect of orthogonalization strength on ImageNet-W.** We report CD (↓) and SD (↓) for the All Classes Paired CAV. Values represent the mean ± two standard errors over 5 seeds.

Scenario	**Backbone**	**Method**	**Metric**	**α value**					
				0.00	0.25	0.50	0.75	1.00	1.25
Church	CLIP	LAT	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.04_{\pm.08}$
			SD	$1.0_{\pm.00}$	$.50_{\pm.00}$	$.15_{\pm.03}$	$.08_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
		SVM	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
			SD	$1.0_{\pm.00}$	$.78_{\pm.15}$	$.65_{\pm.20}$	$.60_{\pm.20}$	$.43_{\pm.18}$	$.38_{\pm.21}$
	SigLIP	LAT	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
			SD	$1.0_{\pm.00}$	$.66_{\pm.00}$	$.56_{\pm.08}$	$.49_{\pm.00}$	$.15_{\pm.00}$	$.15_{\pm.00}$
		SVM	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
			SD	$1.0_{\pm.00}$	$.83_{\pm.00}$	$.73_{\pm.08}$	$.69_{\pm.07}$	$.69_{\pm.07}$	$.66_{\pm.00}$
Husky	CLIP	LAT	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
			SD	$1.0_{\pm.00}$	$.67_{\pm.00}$	$.29_{\pm.00}$	$.14_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
		SVM	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
			SD	$1.0_{\pm.00}$	$.97_{\pm.04}$	$.94_{\pm.07}$	$.87_{\pm.13}$	$.83_{\pm.12}$	$.81_{\pm.14}$
	SigLIP	LAT	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.04_{\pm.08}$
			SD	$1.0_{\pm.00}$	$.37_{\pm.00}$	$.24_{\pm.00}$	$.11_{\pm.00}$	$.04_{\pm.06}$	$-.01_{\pm.00}$
		SVM	CD	$0.0_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$	$.00_{\pm.00}$
			SD	$1.0_{\pm.00}$	$.85_{\pm.15}$	$.62_{\pm.24}$	$.52_{\pm.25}$	$.39_{\pm.15}$	$.37_{\pm.14}$

C.5 CAV Training Details

In this section, we detail the hyperparameter optimization protocols for the CAV extraction methods. Importantly, all performance metrics reported in the main paper are computed on strictly held-out validation sets.

We selected specific backbone versions for DINOv2 (ViT-B/14), CLIP (ViT-L/14), and SigLIP (ViT-B/16) for shared output dimensionality of $d = 768$. This consistency facilitates fair CAV evaluations and permits a fixed SAE expansion rate across all models, simplifying the experimental setup.

All downstream models and SAE models were trained on NVIDIA A100. Concept Activation Vectors were trained on CPU, and vector metrics were similarly extracted using CPU resources. Steering metrics were extracted on NVIDIA A100 GPUs to accommodate the significant computational demands of the backbone models and downstream classifiers.

C.5.1 Optimization-Based Linear CAVs

For methods requiring iterative optimization, specifically logistic regression (LR) and linear SVM, we adopt the robust training pipeline from (Kantamneni et al., 2025).[1] To accommodate varying concept frequencies, we employ an adaptive cross-validation strategy:

- small datasets (< 128 samples): we use stratified K-fold CV (default $K = 5$).
- large datasets (≥ 128 samples): we use a stratified shuffle split with a single validation fold, sized at $\min(0.2 \times N, 100)$.

Grid search. For all SVM- and LR-based CAV extraction methods, we optimize the regularization strength parameter C over 20 logarithmically spaced values in the range $[10^{-3}, 10^{3}]$. All linear models are trained with:

- class balancing: weights are adjusted inversely proportional to class frequencies to handle imbalances,
- intercept: fitting is enabled,
- convergence: a maximum of 10,000 iterations.

The optimal C is selected to maximize AUC on validation splits from the training set. The final CAV is obtained by retraining on the full training set using the best C. Additionally, S&PTopK was tested with LR- (including C optimization) and Stylist-based neuron selection. In Figure 1, we refer to S&PTopK LR as S&PTopK_probe to distinguish it from standard LR. For SAE-AurA, we use W_{dec} and W_{enc} as the feature directions. The whole grid was evaluated for each CAV independently.

C.5.2 Nonlinear Probes

SVC with RBF kernel (SVM-RBF). We standardize features and perform a grid search using 5-fold cross-validation over 105 parameter combinations:

- C: 15 logarithmically spaced values in $[10^{-3}, 10^{3}]$.
- γ: A discrete set $\{10^{-4}, 5 \times 10^{-3}, 10^{-3}, 10^{-2}, 10^{-1}, \texttt{scale}, \texttt{auto}\}$.

TabPFN. We use the transformer-based TabPFN (Hollmann et al., 2023) version 2.5, which performs well on tabular datasets. As a pre-trained network, it requires no hyperparameter tuning, providing a deterministic, robust nonlinear baseline out of the box.

C.5.3 SAE Density Search

For density-based sparse autoencoder methods–SAE-Density and SAE-DensityCommon–the optimal activation density threshold τ varies significantly across concepts. We employ a 5-fold CV sweep to adaptively select this threshold:

[1] https://github.com/JoshEngels/SAE-Probes

1. search space: we define a range of candidate density thresholds d (from 0.3 to 1.0, i.e., features firing in more than $d \times 100\%$ of samples), with step size 0.01,
2. evaluation: for each candidate d:
   (a) we compute the steering vector on the training fold using features filtered by d,
   (b) we evaluate the vector's separation ability (AUC) on the validation fold,
3. selection: we select the density d^* that yields the highest mean validation AUC,
4. refinement: the final vector is re-computed on the full dataset using the optimal density threshold d^*.

C.6 SAE Implementation Details

We train Sparse Autoencoders (SAEs) with Top-K sparsity (Gao et al., 2025) and hyperparameter settings based on (Costa et al., 2025). Our implementation utilizes the `overcomplete` Python package[2], which provides efficient kernels for SAE training. We chose TopK SAE over newer variants (Zaigrajew et al., 2025; Costa et al., 2025) due to its wide adoption, simple training, and strong performance (Chatzoudis et al., 2025; Leask et al., 2025; Dreyer et al., 2025).

Architecture. We employ the Top-K SAE architecture, which enforces sparsity by strictly retaining only the k largest activations in the latent code. Given an input $h \in \mathbb{R}^d$, the forward pass is defined as:

$$\mathbf{z}_{\text{pre}} = \text{ReLU}(W_{\text{enc}}(h - \mathbf{b}_d)) + b_{\text{enc}}) \tag{12}$$

$$\mathbf{z} = \text{TopK}(\mathbf{z}_{\text{pre}}, k) \tag{13}$$

$$\hat{h} = W_{\text{dec}}\mathbf{z} + \mathbf{b}_d \tag{14}$$

where $W_{\text{enc}} \in \mathbb{R}^{m \times d}$ is the encoder, $W_{dec} \in \mathbb{R}^{d \times m}$ is the decoder (dictionary), $m = r \cdot d$ is the number of latent features determined by the expansion rate r, and $\hat{h}$ is a reconstructed input. The `overcomplete` implementation omits the explicit decoder bias parameter $\mathbf{b}_d$ in the final reconstruction step, which we acknowledge may influence the model's ability to recover the mean shift.

Loss Function. The training objective minimizes the reconstruction error combined with an auxiliary reanimation regularizer to mitigate dead features (latents that are rarely active):

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{aux}} \tag{15}$$

where $\mathcal{L}_{\text{recon}} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$ is the Mean Squared Error, and $\mathcal{L}_{\text{aux}}$ is the reanimation loss implemented in `overcomplete`. We set the auxiliary weight $\lambda = 10^{-7}$.

Training data & preprocessing. SAEs are trained on image embeddings extracted from the CC3M dataset (Sharma et al., 2018), a standard choice for pre-training vision models. CC3M is openly available via Hugging Face `pixparse/cc3m-wds` CC BY 4.0 license. Validation is performed on ImageNet-1k embeddings from the vision model. Following standard practice,[3] we normalize embeddings before SAE training and rescale SAE outputs to the original magnitude during evaluation. Specifically, we normalize by the RMS of norms, scaling so that $\sqrt{\mathbb{E}[|x|^2]} = \sqrt{m}$, rather than the mean norm $\mathbb{E}[|x|] = \sqrt{m}$ described in the reference. Our primary motivation for this normalization is to keep the sparsity penalty λ comparable across representation sizes d. Since we use a TopK SAE with no λ parameter, the precise normalization criterion has little impact. We have empirically verified that training with such normalization yields marginally better reconstruction quality than without normalization, and therefore retain it.

Training configuration. We perform a grid search over learning rates and Top-K values (54 total combinations). Optimization uses AdamW (no weight decay) with gradient clipping (1.0). We employ the same cosine scheduler with linear warmup described in Appendix C.2. Detailed settings are provided in Table 18.

[2] https://github.com/jkminder/overcomplete

[3] https://transformer-circuits.pub/2024/april-update/index.html

Table 18: **SAE training configuration.** We utilize a fixed expansion rate while sweeping over learning rates and Top-K values. A large batch size and consistent seed ensure stable, reproducible training across 50 epochs.

Fixed parameters		**Hyperparameter grid**	
Setting	Value	Hyperparameter	Values
optimizer	AdamW ($\epsilon = 6.25\text{e-}10$)	learning rate	{1e-3, 5e-4, 1e-4}
batch size	4096	Top-K	{32, 64, 128}
epochs	50		
scheduler	Cosine + 10-Epoch Warmup		
grad clip	1.0		
weight decay	0		
expansion rate	32		
seed	42		

C.6.1 SAE Performance Metrics

Table 19: **SAE Performance across architectures and datasets.** We report reconstruction quality (R^2) separately for concept-present ($\mathbb{A}_c$) and concept-absent ($\mathbb{A}_{\bar{c}}$) splits to assess potential bias. For ImageNet-1k validation, we report only ($\mathbb{A}_c$), as it represents the entire validation set. For downstream tasks, we report results based on the validation set used in the CAV evaluation; however, we do not report the dead-features. This metric was recorded only during training.

Dataset	**Model**	L_0	R^2		**Dead Feature Ratio**
			$\mathbb{A}_c$	$\mathbb{A}_{\bar{c}}$	
ImageNet-1k	CLIP	64	.91	-	5%
	DINOv2	64	.80	-	0%
	SigLIP	64	.92	-	3%
CelebA	CLIP	64	.89	.89	-
	DINOv2	64	.85	.85	-
	SigLIP	64	.91	.91	-
Waterbirds	CLIP	64	.89	.89	-
	DINOv2	64	.71	.70	-
	SigLIP	64	.91	.90	-
ISIC	CLIP	64	.86	.86	-
	DINOv2	64	.69	.70	-
	SigLIP	64	.89	.89	-
Counteranimal	CLIP	64	.92	.92	-
	DINOv2	64	.77	.77	-
	SigLIP	64	.94	.93	-
ImageNet-C	CLIP	64	.87	.92	-
	DINOv2	64	.76	.84	-
	SigLIP	64	.91	.93	-
ImageNet-W	CLIP	64	.87	.92	-
	SigLIP	64	.91	.93	-

Table 19 summarizes the performance of the selected SAEs across all datasets. For model selection, we prioritized the trade-off between reconstruction fidelity (R^2) and sparsity (L_0), ultimately selecting Top-K = 64 for all reported experiments. We report:

- R^2 score: the coefficient of determination for reconstruction quality.
- L_0 sparsity: the average number of non-zero active features per sample.

- dead feature ratio: The fraction of features that never activate across the entire training set.

C.7 Compute Resources

We estimate the compute resources needed to reproduce all results in this paper for ca. 5K A100 GPU hours plus 10K CPU hours on a high-RAM machine. We effectively used $2-3\times$ more compute for preliminary and failed experiments. Note that CAV methods use cached representations from the backbones and thus are highly efficient to compute on a CPU; see specific compute times in Tables 22 to 24. In contrast, training downstream models and SAEs, as well as computing the metrics, was performed on a GPU.

D EXTENDED METRIC RESULTS

Tables 20–31 extend the evaluations from Section 4 by reporting metrics for all three vision backbones across the datasets introduced in Section 3.4. Metrics are aggregated across the concepts specified for each dataset in Appendix C.1. For ImageNet-W and ImageNet-C, results are further aggregated over five random seeds. We report the mean ± two standard errors (2SE) across these concepts or seeds for ImageNet-W/C. The counteranimals results are presented in Appendix D.1. In addition, we provide a critical difference ranking in Appendix D.2 to determine which CelebA-based method results in the minimum collateral damage. Consistent with Tables 1–2, we report only the best-performing variant for each method—SAE-AurA (Encoder/Decoder), S&PTopK (LR/Stylist), and PCA/PosPCA—selected based on overall performance.

Table 20: **Evaluating methods for steering image representations across two backbones CLIP nd SigLIP on the CelebA dataset.** Expanding on Table 1, we compare performance across two additional backbones: CLIP and DINOv2 backbone models. Columns report Concept Detection (AUC), Cross-Concept Robustness (CCR), Maximum Similarity (MS), and Collateral Damage (CD). Values represent the mean ± two standard errors across concepts, and we indicate the best method in each evaluation metric in **Bold** while the second best are underlined. Results on the other two backbones overlap with previous results, indicating that SVM and LR have comparable performance to nonlinear models and also provide better disentangled representations than statistical methods or even SAE-based version of the methods.

	CLIP					SigLIP				
Method	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)
Linear SVM	**.90**$_{\pm.03}$	**.94**$_{\pm.01}$	.35$_{\pm.05}$	**.59**$_{\pm.06}$	0.20$_{\pm0.21}$	**.91**$_{\pm.03}$	**.94**$_{\pm.02}$	.32$_{\pm.06}$	**.62**$_{\pm.06}$	0.32$_{\pm0.37}$
LR	**.90**$_{\pm.03}$	.89$_{\pm.03}$	.40$_{\pm.06}$	**.59**$_{\pm.06}$	1.23$_{\pm1.30}$	**.91**$_{\pm.02}$	.88$_{\pm.04}$	.34$_{\pm.06}$	.61$_{\pm.06}$	1.27$_{\pm1.82}$
↳ SAE	.87$_{\pm.03}$	.90$_{\pm.02}$	.42$_{\pm.07}$	.53$_{\pm.07}$	0.21$_{\pm0.16}$	.88$_{\pm.03}$	.89$_{\pm.03}$	.35$_{\pm.06}$	.56$_{\pm.07}$	0.47$_{\pm0.62}$
SAS	.80$_{\pm.04}$	.86$_{\pm.03}$	.61$_{\pm.08}$	.47$_{\pm.06}$	5.69$_{\pm3.14}$	.85$_{\pm.04}$	.85$_{\pm.03}$	.64$_{\pm.07}$	.53$_{\pm.07}$	5.70$_{\pm4.55}$
S&PTopK	.80$_{\pm.04}$	.92$_{\pm.02}$	**.24**$_{\pm.04}$	.48$_{\pm.07}$	**0.00**$_{\pm0.01}$	.84$_{\pm.04}$	**.94**$_{\pm.02}$	**.26**$_{\pm.05}$	.53$_{\pm.07}$	**0.03**$_{\pm0.02}$
DiffMean	.85$_{\pm.03}$	.87$_{\pm.03}$	.77$_{\pm.06}$	.51$_{\pm.06}$	17.6$_{\pm5.5}$	.86$_{\pm.03}$	.86$_{\pm.03}$	.79$_{\pm.06}$	.53$_{\pm.06}$	11.3$_{\pm3.8}$
↳ SAE	.85$_{\pm.03}$	.86$_{\pm.02}$	.78$_{\pm.06}$	.51$_{\pm.06}$	17.7$_{\pm5.4}$	.86$_{\pm.03}$	.85$_{\pm.03}$	.80$_{\pm.06}$	.53$_{\pm.06}$	10.4$_{\pm3.5}$
DiffMedian	.84$_{\pm.03}$	.88$_{\pm.03}$	.78$_{\pm.06}$	.50$_{\pm.06}$	18.3$_{\pm5.6}$	.86$_{\pm.03}$	.87$_{\pm.03}$	.80$_{\pm.06}$	.53$_{\pm.06}$	11.3$_{\pm3.7}$
↳ SAE	.77$_{\pm.04}$	.79$_{\pm.04}$	.72$_{\pm.06}$	.44$_{\pm.06}$	2.74$_{\pm2.19}$	.81$_{\pm.03}$	.77$_{\pm.04}$	.83$_{\pm.06}$	.48$_{\pm.06}$	12.1$_{\pm3.4}$
FastCAV	.85$_{\pm.03}$	.87$_{\pm.03}$	.77$_{\pm.06}$	.51$_{\pm.06}$	17.6$_{\pm5.5}$	.86$_{\pm.03}$	.86$_{\pm.03}$	.79$_{\pm.06}$	.53$_{\pm.06}$	11.3$_{\pm3.8}$
↳ SAE	.85$_{\pm.03}$	.86$_{\pm.02}$	.78$_{\pm.06}$	.51$_{\pm.06}$	17.7$_{\pm5.4}$	.86$_{\pm.03}$	.85$_{\pm.03}$	.80$_{\pm.06}$	.53$_{\pm.06}$	10.4$_{\pm3.5}$
PatCAV	.85$_{\pm.03}$	.87$_{\pm.03}$	.77$_{\pm.06}$	.51$_{\pm.06}$	17.6$_{\pm5.5}$	.86$_{\pm.03}$	.86$_{\pm.03}$	.79$_{\pm.06}$	.53$_{\pm.06}$	11.3$_{\pm3.8}$
AurA	.86$_{\pm.03}$	.86$_{\pm.03}$	.78$_{\pm.05}$	.52$_{\pm.06}$	0.58$_{\pm0.32}$	.87$_{\pm.03}$	.85$_{\pm.03}$	.80$_{\pm.05}$	.55$_{\pm.06}$	0.24$_{\pm0.11}$
↳ SAE	.83$_{\pm.03}$	.73$_{\pm.03}$	.94$_{\pm.01}$	.48$_{\pm.06}$	0.12$_{\pm0.05}$	.86$_{\pm.03}$	.73$_{\pm.04}$	.92$_{\pm.02}$	.54$_{\pm.06}$	0.04$_{\pm0.02}$
PCA	.75$_{\pm.04}$	.74$_{\pm.03}$	.97$_{\pm.01}$	.42$_{\pm.07}$	49.5$_{\pm4.8}$	.75$_{\pm.04}$	.73$_{\pm.04}$	.98$_{\pm.01}$	.43$_{\pm.06}$	32.9$_{\pm3.1}$
LAT	.63$_{\pm.02}$	.61$_{\pm.05}$	.91$_{\pm.02}$	.35$_{\pm.06}$	33.9$_{\pm7.6}$	.66$_{\pm.03}$	.50$_{\pm.05}$	.94$_{\pm.02}$	.38$_{\pm.06}$	25.1$_{\pm5.3}$
SVM-RBF	.90$_{\pm.03}$	–	–	–	–	.91$_{\pm.02}$	–	–	–	–
TabPFN	.90$_{\pm.03}$	–	–	–	–	.91$_{\pm.03}$	–	–	–	–

Table 21: **Evaluating methods for steering image representations of CLIP and SigLIP models on the ISIC dataset.** Expanding on Table 1, we compare performance across two additional backbones: CLIP and DINOv2 backbone models. Columns report Concept Detection (AUC), Cross-Concept Robustness (CCR), Maximum Cosine Similarity (MS), Downstream Concept Detection (F1), and Collateral Damage (CD). We indicate the best method for each evaluation metric in **Bold**, while the second-best methods are underlined.

	CLIP					SigLIP				
Method	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)
Linear SVM	$\mathbf{.90}_{\pm.11}$	$\underline{.97}_{\pm.03}$	$\underline{.10}_{\pm.05}$	$\mathbf{.80}_{\pm.14}$	$4.14_{\pm3.63}$	$\mathbf{.89}_{\pm.11}$	$.95_{\pm.05}$	$\underline{.11}_{\pm.08}$	$\underline{.78}_{\pm.15}$	$1.67_{\pm2.31}$
LR	$\underline{.89}_{\pm.11}$	$\underline{.97}_{\pm.03}$	$.11_{\pm.04}$	$\underline{.79}_{\pm.13}$	$4.65_{\pm3.98}$	$\mathbf{.89}_{\pm.11}$	$.94_{\pm.06}$	$\mathbf{.10}_{\pm.06}$	$\mathbf{.79}_{\pm.16}$	$1.76_{\pm2.63}$
↳ SAE	$.83_{\pm.14}$	$\mathbf{.99}_{\pm.01}$	$.17_{\pm.08}$	$.73_{\pm.16}$	$\underline{1.55}_{\pm2.47}$	$\underline{.83}_{\pm.14}$	$\mathbf{.98}_{\pm.02}$	$.17_{\pm.06}$	$.74_{\pm.16}$	$5.12_{\pm3.26}$
SAS	$.71_{\pm.14}$	$\underline{.97}_{\pm.04}$	$.30_{\pm.11}$	$.66_{\pm.15}$	$\mathbf{1.25}_{\pm8.64}$	$.73_{\pm.07}$	$\underline{.97}_{\pm.03}$	$.34_{\pm.15}$	$.68_{\pm.13}$	$2.03_{\pm4.86}$
S&PTopK	$.74_{\pm.12}$	$\mathbf{.99}_{\pm.01}$	$\mathbf{.05}_{\pm.02}$	$.64_{\pm.18}$	$1.95_{\pm1.56}$	$.79_{\pm.12}$	$\underline{.97}_{\pm.04}$	$.12_{\pm.07}$	$.70_{\pm.17}$	$\underline{0.62}_{\pm1.34}$
DiffMean	$.81_{\pm.13}$	$.94_{\pm.05}$	$.51_{\pm.29}$	$.70_{\pm.14}$	$4.36_{\pm8.19}$	$.82_{\pm.10}$	$.94_{\pm.05}$	$.53_{\pm.25}$	$.71_{\pm.14}$	$7.70_{\pm10.6}$
↳ SAE	$.78_{\pm.13}$	$.94_{\pm.07}$	$.56_{\pm.26}$	$.68_{\pm.14}$	$2.75_{\pm4.17}$	$.80_{\pm.09}$	$.94_{\pm.05}$	$.51_{\pm.23}$	$.70_{\pm.14}$	$3.16_{\pm5.55}$
DiffMedian	$.80_{\pm.13}$	$.95_{\pm.04}$	$.52_{\pm.29}$	$.70_{\pm.14}$	$2.78_{\pm7.52}$	$.81_{\pm.10}$	$.94_{\pm.05}$	$.52_{\pm.27}$	$.71_{\pm.14}$	$8.44_{\pm10.5}$
↳ SAE	$.64_{\pm.08}$	$.93_{\pm.07}$	$.70_{\pm.04}$	$.64_{\pm.12}$	$2.83_{\pm9.07}$	$.77_{\pm.12}$	$.93_{\pm.04}$	$.41_{\pm.23}$	$.59_{\pm.31}$	$3.58_{\pm3.78}$
FastCAV	$.81_{\pm.13}$	$.94_{\pm.05}$	$.51_{\pm.29}$	$.70_{\pm.14}$	$4.36_{\pm8.19}$	$.82_{\pm.10}$	$.94_{\pm.05}$	$.53_{\pm.25}$	$.71_{\pm.14}$	$7.70_{\pm10.6}$
↳ SAE	$.78_{\pm.13}$	$.94_{\pm.07}$	$.56_{\pm.26}$	$.68_{\pm.14}$	$2.75_{\pm4.17}$	$.80_{\pm.09}$	$.94_{\pm.05}$	$.51_{\pm.23}$	$.70_{\pm.14}$	$3.16_{\pm5.55}$
PatCAV	$.81_{\pm.13}$	$.94_{\pm.05}$	$.51_{\pm.29}$	$.70_{\pm.14}$	$4.36_{\pm8.19}$	$.82_{\pm.10}$	$.94_{\pm.05}$	$.53_{\pm.25}$	$.71_{\pm.14}$	$7.70_{\pm10.6}$
AurA	$.80_{\pm.12}$	$.93_{\pm.05}$	$.61_{\pm.18}$	$.70_{\pm.14}$	$2.76_{\pm4.58}$	$\underline{.83}_{\pm.10}$	$.91_{\pm.08}$	$.64_{\pm.16}$	$.71_{\pm.14}$	$3.35_{\pm5.45}$
↳ SAE	$.76_{\pm.11}$	$.86_{\pm.12}$	$.88_{\pm.09}$	$.69_{\pm.12}$	$3.73_{\pm4.71}$	$.77_{\pm.08}$	$.87_{\pm.09}$	$.83_{\pm.10}$	$.69_{\pm.14}$	$\mathbf{0.43}_{\pm4.28}$
PCA	$.54_{\pm.05}$	$.86_{\pm.22}$	$1.00_{\pm.00}$	$.45_{\pm.13}$	$2.50_{\pm1.80}$	$.49_{\pm.02}$	$.67_{\pm.17}$	$.99_{\pm.00}$	$.57_{\pm.10}$	$1.91_{\pm3.62}$
LAT	$.53_{\pm.05}$	$.84_{\pm.05}$	$.99_{\pm.00}$	$.45_{\pm.14}$	$2.70_{\pm1.62}$	$.49_{\pm.02}$	$.65_{\pm.16}$	$.99_{\pm.00}$	$.56_{\pm.11}$	$1.27_{\pm3.07}$

Table 22: **Evaluating methods for steering image representations of CLIP model on the Waterbirds dataset.** Values show the results for both concepts ("land" on the left and "water" on the right). We report computation time of each method (Time) in seconds. For MS, we report absolute values because most methods yield negative scores, indicating that the CAVs for "land" and "water" are nearly antipodal, with the exception of AurA, which has non-negative values. We attribute AurA's performance to its targeted use of negative samples during training. It utilizes them strictly for neuron filtering, steering only the isolated neurons to increase concept presence, rather than actively pushing the representation away from the negative concept. For AurA SAE, CCR results exceed 1.0, demonstrating that orthogonalization improved performance beyond the original score.

Method	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)	Time (s)
Linear SVM	$\mathbf{.98}/\mathbf{.98}$	$.96/.86$	$-.70/-.70$	$\mathbf{.92}/\mathbf{.92}$	$5.63/4.73$	$0.49/1.18$
LR	$\mathbf{.98}/\mathbf{.98}$	$.95/.81$	$-.71/-.71$	$\mathbf{.92}/\mathbf{.92}$	$5.21/4.80$	$0.53/0.83$
↳ SAE	$\mathbf{.98}/\underline{.97}$	$.80/.72$	$-.83/-.83$	$.89/\underline{.89}$	$5.01/3.56$	$1.08/1.92$
SAS	$.91/.91$	$.87/.82$	$-.57/-.57$	$.82/.84$	$18.0/2.76$	$0.86/1.60$
S&PTopK	$\mathbf{.98}/\underline{.97}$	$.97/.95$	$\underline{-.50}/\underline{-.50}$	$\underline{.90}/\underline{.89}$	$4.14/2.11$	$0.23/0.33$
DiffMean	$.96/.94$	$.93/.83$	$-.76/-.76$	$.89/\underline{.89}$	$3.38/1.93$	$\mathbf{0.00}/\mathbf{0.00}$
↳ SAE	$.96/.93$	$.90/.81$	$-.77/-.77$	$.88/.88$	$\mathbf{0.28}/0.69$	$0.13/0.05$
DiffMedian	$.96/.94$	$.95/.84$	$-.72/-.72$	$\underline{.90}/.88$	$3.45/3.38$	$\mathbf{0.00}/\mathbf{0.00}$
↳ SAE	$.93/.94$	$.82/.80$	$-.71/-.71$	$.83/.86$	$\underline{0.72}/2.42$	$0.12/0.05$
FastCAV	$.96/.94$	$.93/.83$	$-.76/-.76$	$.89/\underline{.89}$	$3.38/1.93$	$\mathbf{0.00}/\mathbf{0.00}$
↳ SAE	$.96/.93$	$.90/.81$	$-.77/-.77$	$.88/.88$	$\mathbf{0.28}/0.69$	$0.13/0.04$
PatCAV	$.96/.94$	$.93/.83$	$-.76/-.76$	$.89/\underline{.89}$	$3.38/1.93$	$\mathbf{0.00}/\mathbf{0.00}$
AurA	$\underline{.97}/.95$	$\underline{1.0}/\underline{1.0}$	$+\mathbf{.04}/+\mathbf{.04}$	$.89/\underline{.89}$	$4.14/2.00$	$0.81/0.75$
↳ SAE	$.90/.89$	$\mathbf{1.1}/\mathbf{1.1}$	$+.69/+.69$	$.81/.83$	$2.45/\underline{0.45}$	$25.9/24.7$
PCA	$.84/.75$	$\underline{1.0}/.78$	$-.75/-.75$	$.73/.69$	$3.49/3.97$	$\underline{0.02}/0.05$
LAT	$.66/.66$	$.95/.97$	$-.65/-.65$	$.59/.52$	$0.79/\mathbf{0.03}$	$0.06/\underline{0.03}$

Table 23: **DINOv2 representations steering evaluation on Waterbirds.** In addition to AurA, PCA also exhibits the non-negative values for MS.

Method	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)	Time (s)
Linear SVM	**.97**/**.94**	0.98/0.94	−.33/ − .33	**.91**/.83	2.55/1.00	0.55/0.86
LR	**.97**/**.94**	0.98/0.87	−.39/ − .39	.90/**.84**	4.00/0.90	0.80/1.34
↳ SAE	.92/.83	0.95/0.89	−.28/ − .28	.82/.76	2.49/0.66	1.01/1.83
SAS	.63/.66	1.00/1.00	**−.02**/**−.02**	.60/.59	0.38/0.83	1.09/2.20
S&PTopK	.90/.88	0.87/0.87	−.64/ − .64	.80/.75	0.93/0.59	0.36/0.38
DiffMean	.92/.84	0.93/0.88	−.50/ − .50	.83/.80	4.52/1.69	0.01/**0.00**
↳ SAE	.86/.76	0.93/0.91	−.38/ − .38	.77/.71	4.59/0.90	0.13/0.04
DiffMedian	.91/.84	0.94/0.89	−.44/ − .44	.83/.78	4.38/1.52	**0.00**/**0.00**
↳ SAE	.52/.46	1.00/1.00	**−.02**/**−.02**	.00/.00	2.28/1.66	0.16/0.06
FastCAV	.92/.84	0.93/0.88	−.50/ − .50	.83/.80	4.52/1.69	0.04/0.04
↳ SAE	.86/.76	0.93/0.91	−.38/ − .38	.77/.71	4.59/0.90	0.42/0.14
PatCAV	.92/.84	0.93/0.88	−.50/ − .50	.83/.80	4.52/1.69	**0.00**/**0.00**
AurA	.92/.87	1.01/1.01	+.11/+.11	.83/.82	3.24/1.14	1.15/0.76
↳ SAE	.71/.73	**1.17**/**1.11**	+.98/ + .98	.66/.54	0.24/0.10	26.1/24.8
PCA	.51/.53	1.00/1.00	**+.02**/**+.02**	.49/.30	**0.07**/0.10	0.04/0.02
LAT	.47/.54	1.00/1.00	**−.02**/**−.02**	.24/.24	0.28/**0.07**	0.03/0.04

Table 24: **SigLIP representations steering evaluation on Waterbirds.**

Method	AUC (↑)	CCR (↑)	MS (↓)	F1 (↑)	CD (↓)	Time (s)
Linear SVM	**.99**/.98	0.95/0.80	−.80/ − .80	**.94**/.92	11.6/2.52	0.78/1.11
LR	**.99**/.98	0.95/0.79	−.80/ − .80	**.94**/.92	11.6/2.49	0.61/0.89
↳ SAE	**.99**/**.99**	0.94/0.79	−.64/ − .64	.90/.91	11.7/2.35	1.04/1.79
SAS	.96/**.99**	0.78/0.84	−.70/ − .70	.87/.90	8.46/**0.14**	1.23/1.98
S&PTopK	**.99**/**.99**	0.97/0.86	−.54/ − .68	**.94**/**.94**	8.32/1.73	1.08/0.35
DiffMean	.97/.97	0.92/0.75	−.82/ − .82	.91/.89	6.73/2.42	0.14/0.07
↳ SAE	.97/.97	0.89/0.73	−.84/ − .84	.89/.88	6.21/1.42	0.13/0.05
DiffMedian	.97/.97	0.95/0.77	−.76/ − .76	.91/.90	7.35/2.04	0.01/0.02
↳ SAE	.95/.97	0.82/0.86	−.66/ − .66	.86/.90	10.6/2.55	0.15/0.05
FastCAV	.97/.97	0.92/0.75	−.82/ − .82	.91/.89	6.73/2.42	0.06/0.05
↳ SAE	.97/.97	0.89/0.73	−.84/ − .84	.89/.88	6.21/1.42	0.29/0.05
PatCAV	.97/.97	0.92/0.75	−.82/ − .82	.91/.89	6.73/2.42	**0.00**/**0.00**
AurA	.97/.96	1.00/1.00	**+.04**/**+.04**	.90/.90	**1.10**/0.21	1.40/0.78
↳ SAE	.93/.96	**1.04**/**1.01**	+.56/ + .56	.84/.84	5.25/1.31	26.0/24.2
PCA	.77/.48	1.02/0.92	−.18/−.18	.71/.20	13.3/0.24	0.07/0.01
LAT	.57/.53	1.02/**1.01**	+.19/ + .19	.56/.35	5.49/0.21	0.02/0.02

Table 25: **Evaluating results of steering on CLIP on ImageNet-W husky-cat watermark class pair in three data setups.** The results are presented for three CAV training setups: Vision class paired, where samples were drawn from *Cat* class samples; All Classes Paired, from random samples from all 10 classes; and All Classes Unpaired, where we broke counterfactuality by applying a watermark to a different set of images. Columns report Downstream Concept Detection (F1) and Steering Disparity (SD). Values are represented with the mean ± two standard errors across 5 seeds, where the best method (row) in each evaluation metric (column) is in **bold** and the second best is underlined. We don't display CD as only PCA, DiffMean, FastCAV, DiffMedian (with their SAE version) reported deviation from 0 with LAT on the unpaired one.

	Vision Class Paired			**All Classes Paired**			**All Classes Unpaired**		
Method	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.03}$	$.57_{\pm.36}$	$\mathbf{.99}_{\pm.00}$	$.91_{\pm.04}$	$.83_{\pm.12}$	$\mathbf{.99}_{\pm.00}$	$\underline{.94}_{\pm.02}$	$.63_{\pm.16}$
LR	$\mathbf{1.0}_{\pm.00}$	$.92_{\pm.02}$	$.58_{\pm.42}$	$\mathbf{.99}_{\pm.00}$	$\underline{.94}_{\pm.01}$	$.67_{\pm.28}$	$\mathbf{.99}_{\pm.01}$	$.91_{\pm.02}$	$.44_{\pm.21}$
↳ SAE	$\underline{.99}_{\pm.01}$	$.82_{\pm.03}$	$.71_{\pm.00}$	$\underline{.98}_{\pm.02}$	$.78_{\pm.10}$	$.69_{\pm.02}$	$\underline{.98}_{\pm.01}$	$.77_{\pm.11}$	$.74_{\pm.08}$
SAS	$\underline{.99}_{\pm.01}$	$.90_{\pm.06}$	$\mathbf{.05}_{\pm.00}$	$\mathbf{.99}_{\pm.00}$	$.91_{\pm.03}$	$.23_{\pm.08}$	$\mathbf{.99}_{\pm.01}$	$.92_{\pm.01}$	$\underline{.16}_{\pm.06}$
S&PTopK	$\underline{.99}_{\pm.01}$	$\underline{.95}_{\pm.02}$	$.55_{\pm.20}$	$\mathbf{.99}_{\pm.00}$	$.92_{\pm.04}$	$.75_{\pm.06}$	$\mathbf{.99}_{\pm.00}$	$\underline{.94}_{\pm.04}$	$.73_{\pm.02}$
DiffMean	$\underline{.99}_{\pm.00}$	$.84_{\pm.01}$	$.41_{\pm.06}$	$\underline{.98}_{\pm.00}$	$.88_{\pm.02}$	$.40_{\pm.09}$	$\underline{.98}_{\pm.01}$	$.90_{\pm.02}$	$.36_{\pm.13}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.81_{\pm.01}$	$1.1_{\pm.04}$	$\underline{.98}_{\pm.00}$	$.86_{\pm.02}$	$.95_{\pm.00}$	$\underline{.98}_{\pm.01}$	$.87_{\pm.02}$	$1.0_{\pm.09}$
DiffMedian	$\underline{.99}_{\pm.00}$	$.85_{\pm.01}$	$.41_{\pm.07}$	$\underline{.98}_{\pm.00}$	$.89_{\pm.02}$	$.30_{\pm.12}$	$\underline{.98}_{\pm.00}$	$.87_{\pm.01}$	$.36_{\pm.13}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.72_{\pm.01}$	$1.5_{\pm.08}$	$.97_{\pm.00}$	$.80_{\pm.03}$	$1.3_{\pm.07}$	$.97_{\pm.01}$	$.78_{\pm.01}$	$1.4_{\pm.09}$
FastCAV	$\underline{.99}_{\pm.00}$	$.84_{\pm.01}$	$.41_{\pm.06}$	$\underline{.98}_{\pm.00}$	$.88_{\pm.02}$	$.40_{\pm.09}$	$\underline{.98}_{\pm.01}$	$.90_{\pm.02}$	$.36_{\pm.13}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.81_{\pm.01}$	$1.1_{\pm.04}$	$\underline{.98}_{\pm.00}$	$.86_{\pm.02}$	$.95_{\pm.00}$	$\underline{.98}_{\pm.01}$	$.87_{\pm.02}$	$1.0_{\pm.09}$
PatCAV	$\underline{.99}_{\pm.00}$	$.84_{\pm.01}$	$.41_{\pm.06}$	$\underline{.98}_{\pm.00}$	$.88_{\pm.02}$	$.40_{\pm.09}$	$\underline{.98}_{\pm.01}$	$.90_{\pm.02}$	$.36_{\pm.13}$
AurA	$\underline{.99}_{\pm.00}$	$.84_{\pm.01}$	$.24_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.92_{\pm.02}$	$.23_{\pm.02}$	$\underline{.98}_{\pm.00}$	$.91_{\pm.04}$	$.23_{\pm.02}$
↳ SAE	$\underline{.99}_{\pm.01}$	$\mathbf{.97}_{\pm.00}$	$.30_{\pm.02}$	$\mathbf{.99}_{\pm.00}$	$\mathbf{.96}_{\pm.01}$	$.33_{\pm.03}$	$\underline{.98}_{\pm.00}$	$\mathbf{.96}_{\pm.02}$	$.51_{\pm.06}$
PCA	$.86_{\pm.10}$	$.49_{\pm.19}$	$1.1_{\pm.22}$	$.86_{\pm.04}$	$.69_{\pm.02}$	$\underline{.18}_{\pm.25}$	$.91_{\pm.03}$	$.79_{\pm.07}$	$\mathbf{.05}_{\pm.04}$
LAT	$.73_{\pm.03}$	$.42_{\pm.03}$	$\underline{.06}_{\pm.02}$	$.83_{\pm.01}$	$.50_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.55_{\pm.02}$	$.34_{\pm.06}$	$1.3_{\pm.39}$

Table 26: **Evaluating results of steering on SigLIP on ImageNet-W husky-cat watermark class pair in three data setups.** Non-zero CD was observed in SAS, SAE-FastCAV, and all classes in the PCA setup, with unpaired only for LAT.

	Vision Class Paired			All Classes Paired			All Classes Unpaired		
Method	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.92}_{\pm.01}$	$.16_{\pm.10}$	$\mathbf{.99}_{\pm.00}$	$.89_{\pm.05}$	$.39_{\pm.15}$	$\mathbf{.99}_{\pm.00}$	$\underline{.94}_{\pm.02}$	$.29_{\pm.06}$
LR	$\mathbf{1.0}_{\pm.00}$	$\underline{.90}_{\pm.01}$	$\underline{.09}_{\pm.15}$	$\mathbf{.99}_{\pm.00}$	$.88_{\pm.07}$	$.49_{\pm.27}$	$\mathbf{.99}_{\pm.00}$	$.93_{\pm.02}$	$.24_{\pm.08}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.87_{\pm.04}$	$.39_{\pm.19}$	$\underline{.95}_{\pm.04}$	$.71_{\pm.22}$	$.34_{\pm.09}$	$\underline{.99}_{\pm.00}$	$.85_{\pm.05}$	$.34_{\pm.05}$
SAS	$\underline{.99}_{\pm.01}$	$.85_{\pm.05}$	$.16_{\pm.25}$	$\underline{.98}_{\pm.01}$	$.85_{\pm.16}$	$.32_{\pm.28}$	$\mathbf{.99}_{\pm.00}$	$.90_{\pm.05}$	$.24_{\pm.20}$
S&PTopK	$\underline{.99}_{\pm.00}$	$.89_{\pm.02}$	$.29_{\pm.06}$	$\mathbf{.99}_{\pm.00}$	$\underline{.90}_{\pm.02}$	$.65_{\pm.20}$	$\mathbf{.99}_{\pm.00}$	$.90_{\pm.01}$	$.49_{\pm.14}$
DiffMean	$\underline{.99}_{\pm.00}$	$.78_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.88_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.89_{\pm.02}$	$\underline{.11}_{\pm.00}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.68_{\pm.01}$	$.37_{\pm.08}$	$\mathbf{.99}_{\pm.00}$	$.86_{\pm.01}$	$.24_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.86_{\pm.02}$	$.32_{\pm.15}$
DiffMedian	$\underline{.99}_{\pm.00}$	$.75_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.87_{\pm.02}$	$\underline{.09}_{\pm.05}$	$\underline{.98}_{\pm.00}$	$.89_{\pm.04}$	$\mathbf{.09}_{\pm.05}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.66_{\pm.01}$	$.32_{\pm.06}$	$\mathbf{.99}_{\pm.00}$	$.82_{\pm.01}$	$.72_{\pm.19}$	$\mathbf{.99}_{\pm.00}$	$.85_{\pm.02}$	$.59_{\pm.19}$
FastCAV	$\underline{.99}_{\pm.00}$	$.78_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.88_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.89_{\pm.02}$	$\underline{.11}_{\pm.00}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.68_{\pm.01}$	$.37_{\pm.08}$	$\mathbf{.99}_{\pm.00}$	$.86_{\pm.01}$	$.24_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.86_{\pm.02}$	$.32_{\pm.15}$
PatCAV	$\underline{.99}_{\pm.00}$	$.78_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.88_{\pm.01}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.89_{\pm.02}$	$\underline{.11}_{\pm.00}$
AurA	$\underline{.99}_{\pm.00}$	$\underline{.90}_{\pm.00}$	$.11_{\pm.00}$	$\mathbf{.99}_{\pm.00}$	$\mathbf{.95}_{\pm.00}$	$.11_{\pm.00}$	$\underline{.98}_{\pm.00}$	$\mathbf{.95}_{\pm.01}$	$\underline{.11}_{\pm.00}$
↳ SAE	$.99_{\pm.00}$	$.84_{\pm.01}$	$.24_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.89_{\pm.01}$	$.24_{\pm.00}$	$\underline{.98}_{\pm.00}$	$.90_{\pm.02}$	$.24_{\pm.00}$
PCA	$.66_{\pm.02}$	$.53_{\pm.02}$	$.87_{\pm.00}$	$.60_{\pm.02}$	$.46_{\pm.04}$	$3.6_{\pm1.0}$	$.59_{\pm.02}$	$.41_{\pm.04}$	$3.5_{\pm.67}$
LAT	$.91_{\pm.01}$	$.38_{\pm.01}$	$\mathbf{.01}_{\pm.05}$	$.81_{\pm.01}$	$.51_{\pm.01}$	$\mathbf{.04}_{\pm.06}$	$.54_{\pm.02}$	$.35_{\pm.05}$	$3.1_{\pm1.1}$

Table 27: **Evaluating results of steering on CLIP on ImageNet-W church-goldfish watermark class pair in three data setups.** Non-zero CD was observed in PCA, DiffMean, PatCAV, FastCAV, and DiffMedian (including their SAE variants). Additionally, for the LAT in the unpaired version.

	Vision Class Paired			All Classes Paired			All Classes Unpaired		
Method	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$\underline{.97}_{\pm.02}$	$\underline{.08}_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.98}_{\pm.02}$	$.43_{\pm.18}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.98}_{\pm.01}$	$.35_{\pm.08}$
LR	$\mathbf{1.0}_{\pm.00}$	$.95_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.97}_{\pm.01}$	$.32_{\pm.23}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.97}_{\pm.01}$	$.23_{\pm.14}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.75_{\pm.09}$	$.15_{\pm.06}$	$.99_{\pm.00}$	$.81_{\pm.11}$	$.53_{\pm.04}$	$.99_{\pm.00}$	$.92_{\pm.03}$	$.53_{\pm.11}$
SAS	$\mathbf{1.0}_{\pm.00}$	$.70_{\pm.06}$	$\mathbf{.00}_{\pm.00}$	$.97_{\pm.01}$	$.89_{\pm.06}$	$\underline{.07}_{\pm.06}$	$.97_{\pm.01}$	$.88_{\pm.08}$	$\mathbf{.05}_{\pm.07}$
S&PTopK	$\mathbf{1.0}_{\pm.00}$	$.87_{\pm.07}$	$.37_{\pm.12}$	$\underline{.99}_{\pm.00}$	$.96_{\pm.01}$	$.60_{\pm.16}$	$\underline{.99}_{\pm.00}$	$.96_{\pm.01}$	$.52_{\pm.03}$
DiffMean	$\mathbf{1.0}_{\pm.00}$	$.92_{\pm.01}$	$\underline{.08}_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.94_{\pm.00}$	$.17_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.15_{\pm.03}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.66_{\pm.01}$	$.23_{\pm.03}$	$\underline{.99}_{\pm.00}$	$.93_{\pm.00}$	$.38_{\pm.07}$	$\underline{.99}_{\pm.00}$	$.93_{\pm.01}$	$.35_{\pm.10}$
DiffMedian	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.00}$	$\underline{.08}_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.17_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.15_{\pm.03}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.60_{\pm.01}$	$1.9_{\pm.26}$	$.97_{\pm.00}$	$.88_{\pm.02}$	$.93_{\pm.13}$	$.97_{\pm.01}$	$.88_{\pm.03}$	$.93_{\pm.34}$
FastCAV	$\mathbf{1.0}_{\pm.00}$	$.92_{\pm.01}$	$\underline{.08}_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.94_{\pm.00}$	$.17_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.15_{\pm.03}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.66_{\pm.01}$	$.23_{\pm.03}$	$\underline{.99}_{\pm.00}$	$.93_{\pm.00}$	$.38_{\pm.07}$	$\underline{.99}_{\pm.00}$	$.93_{\pm.01}$	$.35_{\pm.10}$
PatCAV	$\mathbf{1.0}_{\pm.00}$	$.92_{\pm.01}$	$\underline{.08}_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.94_{\pm.00}$	$.17_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.15_{\pm.03}$
AurA	$\mathbf{1.0}_{\pm.00}$	$.96_{\pm.00}$	$\underline{.08}_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.96_{\pm.01}$	$.08_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.96_{\pm.00}$	$\underline{.08}_{\pm.00}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.99}_{\pm.00}$	$.50_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.96_{\pm.01}$	$.50_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\underline{.97}_{\pm.01}$	$.50_{\pm.00}$
PCA	$\mathbf{1.0}_{\pm.00}$	$.81_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.98_{\pm.00}$	$.94_{\pm.01}$	$.13_{\pm.04}$	$.98_{\pm.00}$	$.87_{\pm.10}$	$.17_{\pm.00}$
LAT	$.55_{\pm.06}$	$.17_{\pm.08}$	$.20_{\pm.07}$	$.86_{\pm.00}$	$.62_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.53_{\pm.01}$	$.23_{\pm.10}$	$3.6_{\pm.56}$

Table 28: **Evaluating results of steering on SigLIP on ImageNet-W church-goldfish watermark class pair in three data setups.** Non-zero CD was observed across three settings. For Vision Class Paired, this occurred with SAE-FastCAV, SAE-DiffMean, and SAE-DiffMedian. For All Classes Paired, non-zero CD was observed in PCA, as well as the DiffMean, PatCAV, FastCAV, and DiffMedian methods (including their SAE variants). Finally, for All Classes Unpaired, non-zero CD was seen with PCA, LAT, and the aforementioned methods (including their SAE variants).

	Vision Class Paired			All Classes Paired			All Classes Unpaired		
Method	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)	AUC (↑)	F1 (↑)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$\underline{.93}_{\pm.03}$	$.49_{\pm.19}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.97}_{\pm.01}$	$.69_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.98}_{\pm.02}$	$.69_{\pm.07}$
LR	$\mathbf{1.0}_{\pm.00}$	$.80_{\pm.13}$	$.25_{\pm.20}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.96}_{\pm.01}$	$.66_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.98}_{\pm.02}$	$.69_{\pm.07}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.71_{\pm.03}$	$.32_{\pm.15}$	$\underline{.99}_{\pm.00}$	$.86_{\pm.07}$	$.73_{\pm.08}$	$\underline{.99}_{\pm.00}$	$.87_{\pm.09}$	$.66_{\pm.00}$
SAS	$\mathbf{1.0}_{\pm.00}$	$.66_{\pm.06}$	$\mathbf{.02}_{\pm.07}$	$.98_{\pm.01}$	$.83_{\pm.03}$	$.49_{\pm.36}$	$.98_{\pm.01}$	$.76_{\pm.08}$	$.56_{\pm.14}$
S&PTopK	$\mathbf{1.0}_{\pm.00}$	$.87_{\pm.07}$	$.66_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\underline{.96}_{\pm.01}$	$.66_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\underline{.96}_{\pm.01}$	$.66_{\pm.00}$
DiffMean	$\mathbf{1.0}_{\pm.00}$	$.55_{\pm.01}$	$\underline{.15}_{\pm.00}$	$.98_{\pm.00}$	$.77_{\pm.04}$	$\underline{.19}_{\pm.07}$	$.98_{\pm.00}$	$.71_{\pm.10}$	$\underline{.22}_{\pm.08}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.41_{\pm.01}$	$.53_{\pm.07}$	$.97_{\pm.00}$	$.63_{\pm.02}$	$.56_{\pm.08}$	$.97_{\pm.01}$	$.66_{\pm.11}$	$.63_{\pm.07}$
DiffMedian	$\mathbf{1.0}_{\pm.00}$	$.54_{\pm.01}$	$\underline{.15}_{\pm.00}$	$.98_{\pm.00}$	$.80_{\pm.01}$	$\mathbf{.15}_{\pm.00}$	$.97_{\pm.00}$	$.71_{\pm.11}$	$\mathbf{.15}_{\pm.00}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.37_{\pm.03}$	$.83_{\pm.11}$	$.97_{\pm.00}$	$.68_{\pm.03}$	$.53_{\pm.07}$	$.97_{\pm.01}$	$.69_{\pm.03}$	$.56_{\pm.08}$
FastCAV	$\mathbf{1.0}_{\pm.00}$	$.55_{\pm.01}$	$\underline{.15}_{\pm.00}$	$.98_{\pm.00}$	$.77_{\pm.04}$	$\underline{.19}_{\pm.07}$	$.98_{\pm.00}$	$.71_{\pm.10}$	$\underline{.22}_{\pm.08}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.41_{\pm.01}$	$.53_{\pm.07}$	$.97_{\pm.00}$	$.63_{\pm.02}$	$.56_{\pm.08}$	$.97_{\pm.01}$	$.66_{\pm.11}$	$.63_{\pm.07}$
PatCAV	$\mathbf{1.0}_{\pm.00}$	$.55_{\pm.01}$	$\underline{.15}_{\pm.00}$	$.98_{\pm.00}$	$.77_{\pm.04}$	$\underline{.19}_{\pm.07}$	$.98_{\pm.00}$	$.71_{\pm.10}$	$\underline{.22}_{\pm.08}$
AurA	$\mathbf{1.0}_{\pm.00}$	$.62_{\pm.01}$	$.49_{\pm.00}$	$.98_{\pm.00}$	$.89_{\pm.02}$	$.49_{\pm.00}$	$.98_{\pm.00}$	$.87_{\pm.06}$	$.49_{\pm.00}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.98}_{\pm.00}$	$.66_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\underline{.96}_{\pm.00}$	$.66_{\pm.00}$	$\underline{.99}_{\pm.00}$	$.95_{\pm.01}$	$.66_{\pm.00}$
PCA	$.98_{\pm.00}$	$.46_{\pm.00}$	$\underline{.15}_{\pm.00}$	$.55_{\pm.03}$	$.20_{\pm.01}$	$2.6_{\pm.71}$	$.56_{\pm.05}$	$.21_{\pm.05}$	$3.6_{\pm1.9}$
LAT	$.75_{\pm.01}$	$.35_{\pm.01}$	$.42_{\pm.08}$	$.74_{\pm.01}$	$.54_{\pm.01}$	$\mathbf{.15}_{\pm.00}$	$.52_{\pm.02}$	$.24_{\pm.08}$	$2.2_{\pm.86}$

Table 29: **Evaluating steering representations on ImageNet-C *Shot Noise* and *Glass Blur* corruptions on CLIP.** The best method (row) in each evaluation metric (column) is in **bold** and the second best is underlined.

	Shot Noise					Glass Blur				
Method	AUC (↑)	MS (↓)	F1 (↑)	CD (↓)	SD (↓)	AUC (↑)	MS (↓)	F1 (↑)	CD (↓)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$\underline{.98}_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.86_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.74_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.16_{\pm.05}$
LR	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.86_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.75_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{.00}_{\pm.00}$	$\underline{1.13}_{\pm.03}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.96}_{\pm.04}$	$\underline{.99}_{\pm.32}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.74_{\pm.05}$	$\underline{.99}_{\pm.10}$	$\mathbf{.00}_{\pm.00}$	$1.19_{\pm.01}$
SAS	$\mathbf{1.0}_{\pm.00}$	$\underline{.98}_{\pm.03}$	$\underline{.99}_{\pm.15}$	$\mathbf{.00}_{\pm.00}$	$.98_{\pm.16}$	$\mathbf{1.0}_{\pm.00}$	$.77_{\pm.03}$	$\underline{.99}_{\pm.18}$	$\mathbf{.00}_{\pm.00}$	$1.28_{\pm.03}$
S&PTopK	$\mathbf{1.0}_{\pm.00}$	$\underline{.98}_{\pm.01}$	$\underline{.99}_{\pm.23}$	$\mathbf{.00}_{\pm.00}$	$.91_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.81_{\pm.03}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.15_{\pm.03}$
DiffMean	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.82_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.73}_{\pm.00}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.15_{\pm.02}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.83_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.83_{\pm.00}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.21_{\pm.03}$
DiffMedian	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{.00}_{\pm.00}$	$.85_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.73}_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{.00}_{\pm.00}$	$1.17_{\pm.01}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$\underline{.79}_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.82_{\pm.01}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.15_{\pm.02}$
FastCAV	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.82_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.73}_{\pm.00}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.15_{\pm.02}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.83_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.83_{\pm.00}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.21_{\pm.03}$
PatCAV	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.82_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.73}_{\pm.00}$	$\underline{.99}_{\pm.05}$	$\mathbf{.00}_{\pm.00}$	$1.15_{\pm.02}$
AurA	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.94_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.76_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{.00}_{\pm.00}$	$1.18_{\pm.01}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\underline{.99}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.94_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$.89_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{.00}_{\pm.00}$	$1.15_{\pm.01}$
PCA	$\mathbf{1.0}_{\pm.00}$	$1.0_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.70}_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.72}_{\pm.01}$	$\underline{.99}_{\pm.07}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{1.12}_{\pm.03}$
LAT	$.76_{\pm.04}$	$.97_{\pm.02}$	$.72_{\pm.21}$	$\mathbf{.00}_{\pm.00}$	$1.4_{\pm.21}$	$.67_{\pm.07}$	$.81_{\pm.04}$	$.71_{\pm.82}$	$.04_{\pm.08}$	$2.67_{\pm.23}$

Table 30: **Evaluating steering representations on ImageNet-C *Shot Noise* and *Glass Blur* corruptions DINOv2.** The best method (row) in each evaluation metric (column) is in **bold** and the second best is underlined.

	Shot Noise					Glass Blur				
Method	AUC (↑)	MS (↓)	F1 (↑)	CD (↓)	SD (↓)	AUC (↑)	MS (↓)	F1 (↑)	CD (↓)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$.91_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.64_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.03_{\pm.03}$
LR	$\mathbf{1.0}_{\pm.00}$	$.94_{\pm.02}$	$\underline{.99}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.66_{\pm.02}$	$\underline{.99}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.03_{\pm.02}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.72}_{\pm.03}$	$.98_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$1.0_{\pm.07}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.48}_{\pm.04}$	$\underline{.99}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{0.95}_{\pm.05}$
SAS	$.97_{\pm.00}$	$.96_{\pm.02}$	$.91_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.01}$	$.98_{\pm.00}$	$.74_{\pm.04}$	$.95_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$0.97_{\pm.00}$
S&PTopK	$\underline{.99}_{\pm.00}$	$.92_{\pm.02}$	$.96_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$1.0_{\pm.01}$	$\underline{.99}_{\pm.00}$	$.72_{\pm.04}$	$.95_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{0.94}_{\pm.01}$
DiffMean	$\mathbf{1.0}_{\pm.00}$	$.96_{\pm.00}$	$.97_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.97}_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.63_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.03_{\pm.01}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.95_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{1.0}_{\pm.00}$	$.76_{\pm.01}$	$.97_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.09_{\pm.02}$
DiffMedian	$\mathbf{1.0}_{\pm.00}$	$.95_{\pm.00}$	$.97_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.0_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.64_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.09_{\pm.02}$
↳ SAE	$.94_{\pm.00}$	$1.0_{\pm.00}$	$.87_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.2_{\pm.00}$	$.96_{\pm.00}$	$1.0_{\pm.00}$	$.91_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{0.95}_{\pm.00}$
FastCAV	$\mathbf{1.0}_{\pm.00}$	$.96_{\pm.00}$	$.97_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.97}_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.63_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.03_{\pm.01}$
↳ SAE	$\underline{.99}_{\pm.00}$	$.95_{\pm.00}$	$.95_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.99}_{\pm.04}$	$\mathbf{1.0}_{\pm.00}$	$.76_{\pm.01}$	$.97_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.09_{\pm.02}$
PatCAV	$\mathbf{1.0}_{\pm.00}$	$.96_{\pm.00}$	$.97_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.97}_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.63_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.03_{\pm.01}$
AurA	$\underline{.99}_{\pm.00}$	$.96_{\pm.00}$	$.96_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.0_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.68_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.04_{\pm.01}$
↳ SAE	$\underline{.99}_{\pm.00}$	$1.0_{\pm.00}$	$.95_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.00}$	$.99_{\pm.00}$	$.99_{\pm.00}$	$.96_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$0.98_{\pm.00}$
PCA	$.84_{\pm.06}$	$\underline{.79}_{\pm.13}$	$.82_{\pm.05}$	$.40_{\pm.52}$	$2.3_{\pm.42}$	$.98_{\pm.00}$	$\underline{.51}_{\pm.07}$	$.94_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$1.29_{\pm.07}$
LAT	$.68_{\pm.05}$	$.90_{\pm.01}$	$.64_{\pm.05}$	$\underline{.12}_{\pm.10}$	$1.1_{\pm.07}$	$.54_{\pm.04}$	$.72_{\pm.03}$	$.51_{\pm.10}$	$\underline{.20}_{\pm.13}$	$0.98_{\pm.08}$

Table 31: **Evaluating steering representations on ImageNet-C *Shot Noise* and *Glass Blur* corruptions SigLIP.** The best method (row) in each evaluation metric (column) is in **bold** and the second best is underlined.

	Shot Noise					Glass Blur				
Method	AUC (↑)	MS (↓)	F1 (↑)	CD (↓)	SD (↓)	AUC (↑)	MS (↓)	F1 (↑)	CD (↓)	SD (↓)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$\underline{.69}_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.01}$	$1.0_{\pm.03}$
LR	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.99_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.75_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.97_{\pm.04}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.92}_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$1.1_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.50}_{\pm.11}$	$.98_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.96_{\pm.01}$
SAS	$\mathbf{1.0}_{\pm.00}$	$.98_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.99_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.70_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.93_{\pm.02}$
S&PTopK	$\mathbf{1.0}_{\pm.00}$	$\underline{.97}_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.97_{\pm.02}$	$\mathbf{1.0}_{\pm.00}$	$.81_{\pm.03}$	$.99_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$.95_{\pm.00}$
DiffMean	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.96}_{\pm.06}$	$\mathbf{1.0}_{\pm.00}$	$.76_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.91_{\pm.02}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.99_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.81_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.88_{\pm.02}$
DiffMedian	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.96}_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.77_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.91_{\pm.00}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.98_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.96}_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.81_{\pm.01}$	$.99_{\pm.01}$	$\mathbf{.00}_{\pm.00}$	$\underline{.87}_{\pm.02}$
FastCAV	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.96}_{\pm.06}$	$\mathbf{1.0}_{\pm.00}$	$.76_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.91_{\pm.02}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.99_{\pm.05}$	$\mathbf{1.0}_{\pm.00}$	$.81_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.88_{\pm.02}$
PatCAV	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.96}_{\pm.06}$	$\mathbf{1.0}_{\pm.00}$	$.76_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.91_{\pm.02}$
AurA	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\underline{.96}_{\pm.03}$	$\mathbf{1.0}_{\pm.00}$	$.77_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.81}_{\pm.01}$
↳ SAE	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.98_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$.84_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.96_{\pm.00}$
PCA	$\mathbf{1.0}_{\pm.00}$	$.99_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$\mathbf{.93}_{\pm.04}$	$\mathbf{1.0}_{\pm.00}$	$.80_{\pm.01}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{.00}_{\pm.00}$	$.91_{\pm.04}$
LAT	$.55_{\pm.06}$	$.98_{\pm.01}$	$.75_{\pm.05}$	$.36_{\pm.23}$	$1.6_{\pm.07}$	$.50_{\pm.05}$	$.80_{\pm.03}$	$.73_{\pm.04}$	$.60_{\pm.72}$	$1.5_{\pm.16}$

D.1 COUNTERANIMALS RESULTS

In prior benchmarks such as CelebA and ISIC, models can exploit spurious correlations between concepts and class labels, effectively using concepts as shortcuts for classification. This behavior arises due to strong class–concept co-occurrence in the training data. In the CounterAnimal dataset, this shortcut phenomenon is further amplified. Here, specific animals not only co-occur with particular backgrounds but are deterministically associated with them. As a result, models may tend to strongly overfit to background cues rather than learning the underlying object representations, leading to severe bias and poor generalization to novel contexts. To better assess debiasing in this setting, we modify the evaluation protocol as follows.

First, we train a classifier on the original training set and evaluate it on the unmodified test set. Next, we repeat the procedure using data in which all identified concepts have been orthogonalized from the training set. This setup aims to eliminate the model's reliance on spurious concept-based signals, beyond simply orthogonalizing concepts during inference. The results are summarized in Table 32. The first three columns report standard CAV-based metrics used throughout this work. The columns "Diff Acc" and "Diff F1" denote the performance differences between models trained on orthogonalized data and those trained on the original data. Overall, most methods yield improvements in downstream classification performance after debiasing. However, perfect concept separability—as achieved by Linear SVM and Logistic Regression—does not necessarily translate into the largest gains in classification accuracy or F1 score. This strengthens our assumption that concept detectability alone is insufficient for effective debiasing.

Table 32: **Evaluating debiasing on CLIP for Counteranimal dataset.** Columns "Diff Acc" and "Diff F1" refer to the difference in the classification task between the model that operated on data orthogonalized to all concepts, and the model using the original data.

Method	AUC (↑)	CCR (↑)	MS (↓)	Diff Acc (↑)	Diff F1 (↑)	Time (s)
Linear SVM	$\mathbf{1.0}_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\mathbf{-.08}_{\pm.04}$	1.9	1.9	$1.4_{\pm.59}$
LR	$\mathbf{1.0}_{\pm.00}$	$\mathbf{1.0}_{\pm.00}$	$\underline{-.07}_{\pm.03}$	1.6	<u>2.0</u>	$1.2_{\pm.48}$
↳ SAE	$\underline{.99}_{\pm.01}$	$\underline{.99}_{\pm.01}$	$-.01_{\pm.08}$	<u>2.1</u>	1.8	$2.1_{\pm.50}$
SAS	$.91_{\pm.07}$	$.95_{\pm.07}$	$.07_{\pm.03}$	.24	.48	$5.2_{\pm3.8}$
S&PTopK	$.97_{\pm.03}$	$\underline{.99}_{\pm.01}$	$.16_{\pm.08}$	1.1	.81	$.88_{\pm.25}$
DiffMean	$.97_{\pm.02}$	$.98_{\pm.02}$	$.00_{\pm.08}$	1.4	1.5	$\mathbf{.00}_{\pm.01}$
↳ SAE	$.97_{\pm.02}$	$.98_{\pm.02}$	$.00_{\pm.08}$	.94	.84	$.26_{\pm.09}$
DiffMedian	$.96_{\pm.02}$	$.98_{\pm.02}$	$.00_{\pm.08}$	<u>2.2</u>	**2.3**	$\underline{.01}_{\pm.01}$
↳ SAE	$.87_{\pm.10}$	$.95_{\pm.06}$	$.15_{\pm.16}$	$-.56$	-1.2	$.26_{\pm.07}$
FastCAV	$.97_{\pm.02}$	$.98_{\pm.02}$	$.00_{\pm.08}$	1.4	1.5	$\mathbf{.00}_{\pm.00}$
↳ SAE	$.97_{\pm.02}$	$.98_{\pm.02}$	$.00_{\pm.08}$	.92	.81	$.25_{\pm.07}$
PatCAV	$.97_{\pm.02}$	$.98_{\pm.02}$	$.00_{\pm.08}$	1.4	1.5	$\mathbf{.00}_{\pm.00}$
AurA	$.97_{\pm.02}$	$\underline{.99}_{\pm.01}$	$.33_{\pm.05}$	**2.4**	<u>2.2</u>	$.90_{\pm.03}$
↳ SAE	$.96_{\pm.03}$	$.96_{\pm.04}$	$.70_{\pm.01}$	1.4	1.5	$28_{\pm.91}$
PCA	$.73_{\pm.09}$	$.90_{\pm.11}$	$.58_{\pm.14}$	$-.66$	$-.71$	$.08_{\pm.04}$
LAT	$.60_{\pm.10}$	$.87_{\pm.11}$	$.59_{\pm.24}$	-1.8	-1.8	$.05_{\pm.05}$

D.2 CRITICAL DIFFERENCE ANALYSIS

In Figure 8, we analyze the critical differences between CAV methods regarding collateral damage (CD) on the CelebA dataset. The results show that S&PTopK achieves the lowest CD on two out of the three backbones. The slightly worse results on DINOv2 may be attributed to the SAE's weak performance on this specific backbone (see Table 19). SVM and SAE-AurA follow closely as top-performing methods. Excluding AurA, we see that on all backbones, statistical methods often lag behind optimization-based approaches.

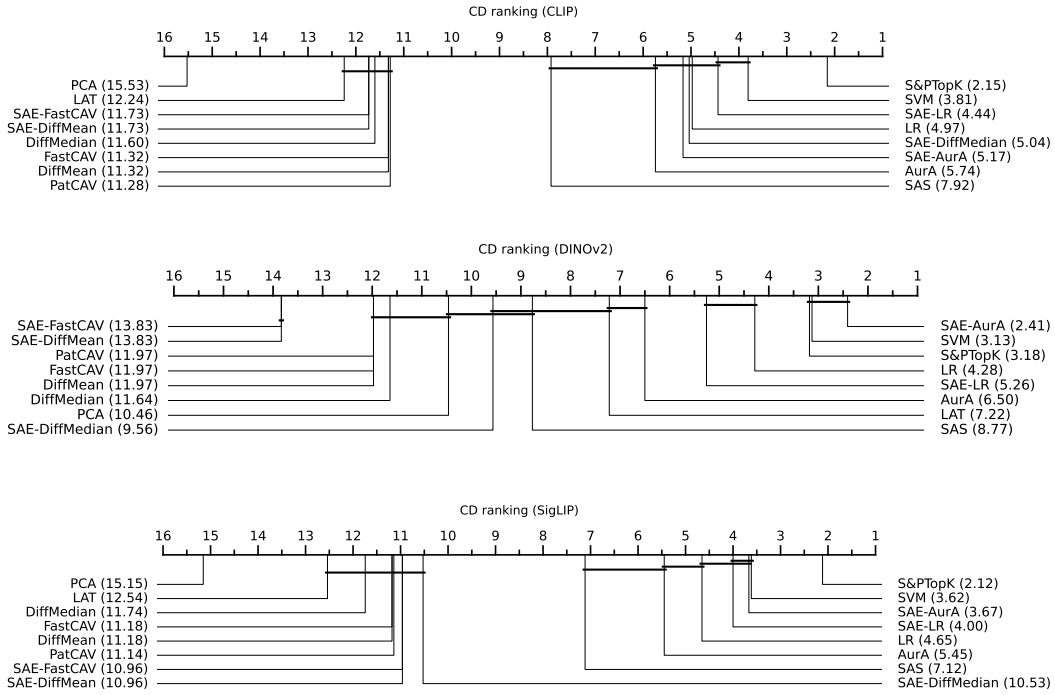

Figure 8: **Critical difference analysis of collateral damage.** The plots display results for CLIP (top), DINOv2 (middle), and SigLIP (bottom). Across all backbones, optimization-based methods consistently outperform statistical and environment-specific approaches, with the notable exception of AurA and SAE-AurA.

E Extended Ablation on Sample Efficiency

Extending the ablations from Section 4, Figure 10 presents sample efficiency ablations for additional backbones on CelebA (CLIP, DINOv2) and Figure 11 the ISIC dataset on all backbones. These results reinforce our primary finding: methods consistently cluster into three distinct groups: optimization-based (•), statistical (■), and "environment-specific" (▲), regardless of the backbone or dataset.

Furthermore, Figure 9 examines the ImageNet-W dataset (*husky* watermark on *cat* class scenario). We observe that data requirements depend heavily on the specific training data setup. Paired setups require significantly fewer samples to establish a stable direction. Consistently, statistical methods demonstrate superior data efficiency, saturating early, whereas optimization-based methods continue to refine their trajectory up to the maximum $N = 500$ samples.

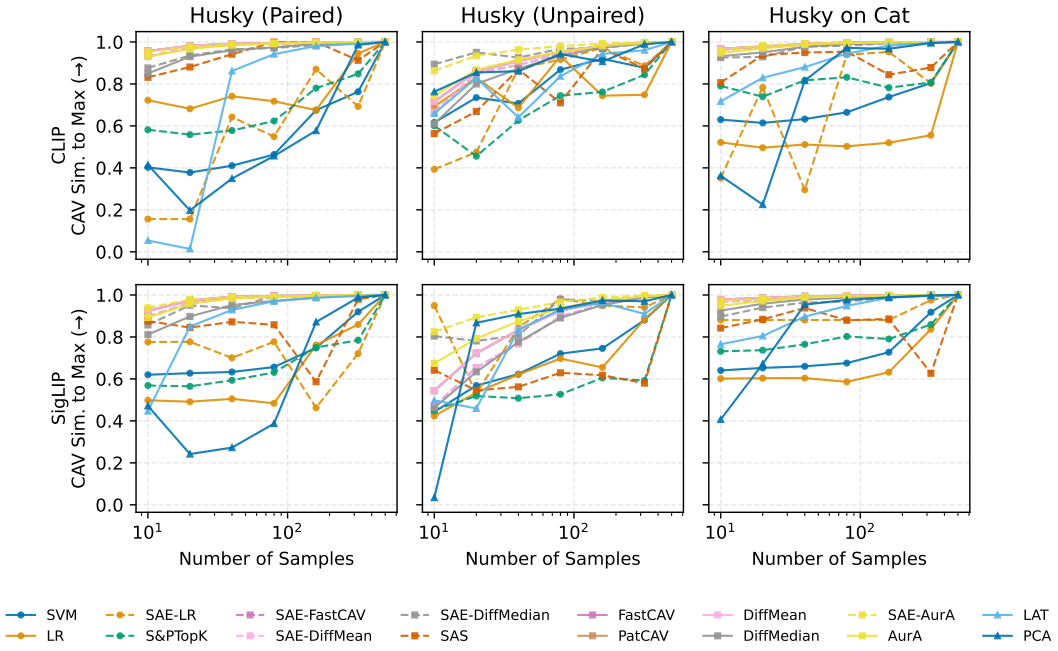

Figure 9: **Sample efficiency on ImageNet-W (husky/cat Watermark).** We compare SigLIP (top row) and CLIP (bottom row) across three setups: watermark on all classes, unpaired samples, and paired (cat-only) watermarks. Results indicate that paired setups require the least amount of data to converge. Notably, optimization-based methods (•) utilize the full range of samples to optimize, whereas statistical (■) methods stabilize much earlier.

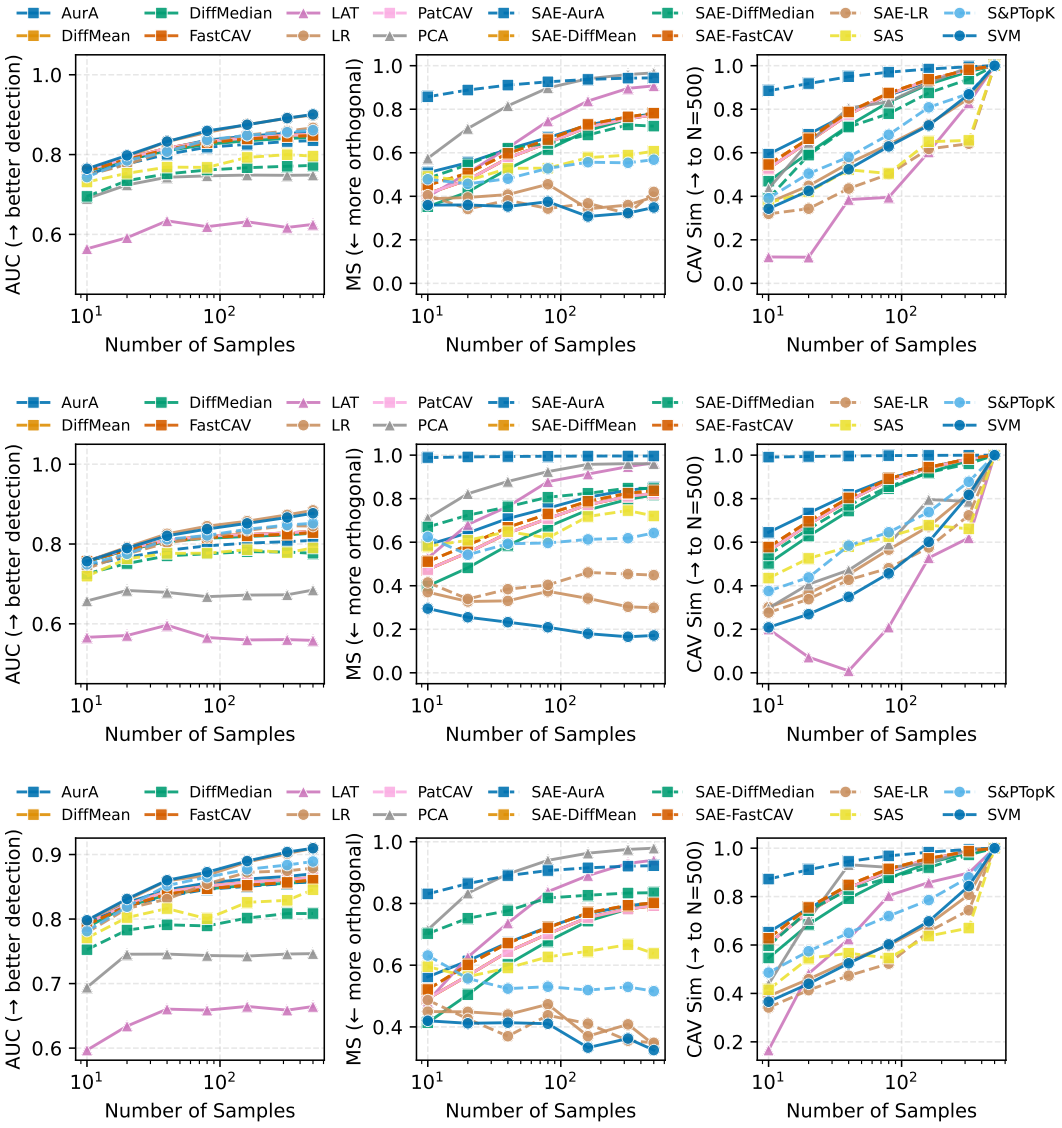

Figure 10: **Extended sample efficiency ablations.** Complementing Figure 2, we report metrics as a function of training sample size for CelebA with CLIP (*top*), CelebA with DINOv2 (*middle*), and CelebA with SigLIP (*bottom*). Convergence trends remain consistent across all backbones, with the exception of SAE-AurA on DINOv2. The DINOv2 SAE-AurA has already converged at just 10 samples, demonstrating only marginal subsequent changes, as indicated by an increase in AUC.

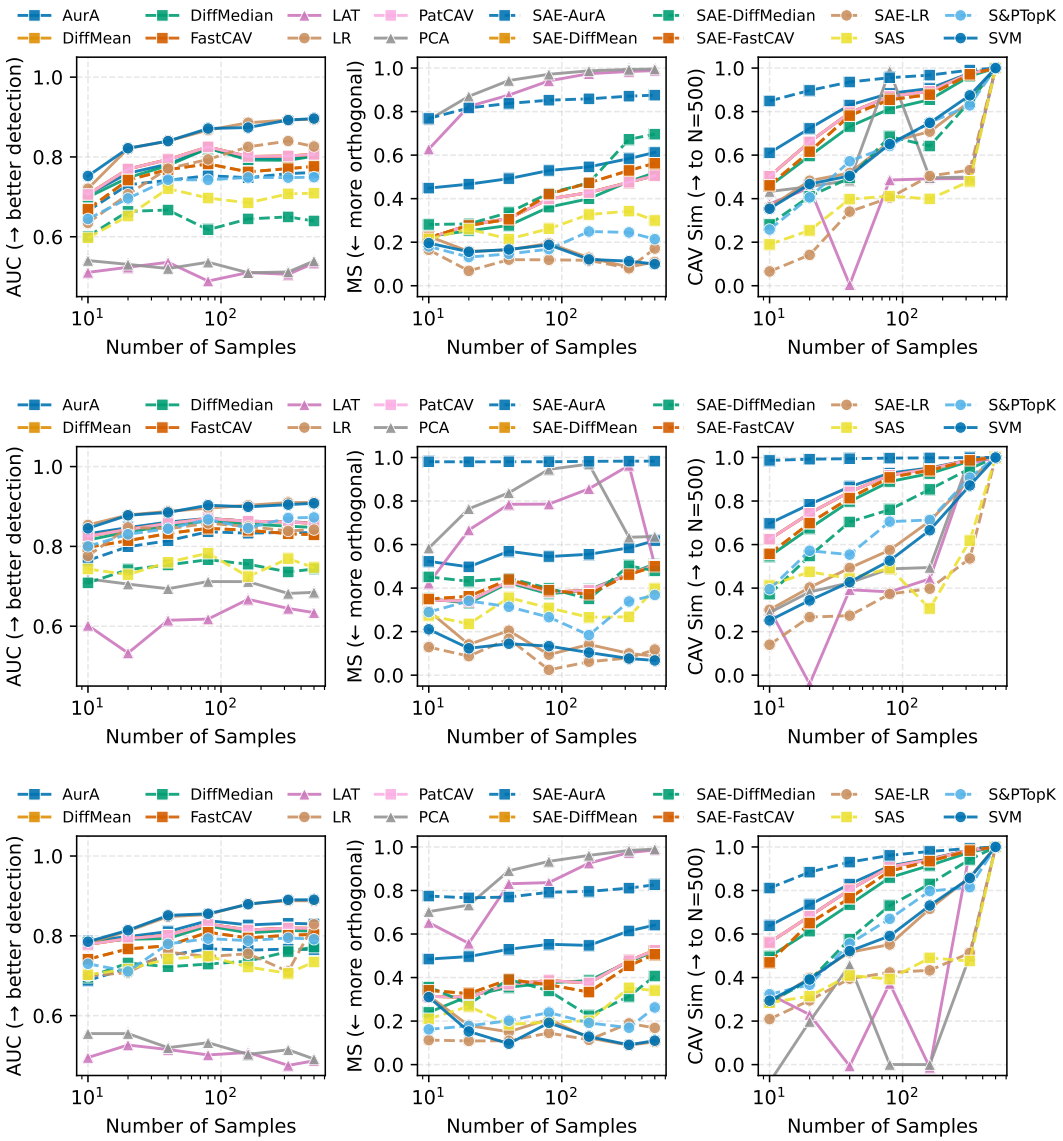

Figure 11: **Sample efficiency ablations for ISIC.** We report metrics as a function of training sample size for: ISIC with CLIP (*top*); ISIC with DINOv2 (*middle*), and ISIC with SigLIP (*bottom*).

F Performance Comparison Between Linear and Nonlinear Methods

This section validates the hypothesis from Finding 1: that top-performing linear CAVs like linear SVM and LR (but not SAE-LR) achieve separation accuracy (AUC) comparable to complex nonlinear probes. We visualize the win-ratio between linear and nonlinear methods for the CelebA and ISIC datasets. (Waterbirds is excluded due to insufficient concept count, and synthetic datasets are excluded due to ceiling effects where AUC ≈ 1.0).

Although raw win rates in Figures 12 and 13 suggest a marginal advantage for nonlinear probes (TabPFN, SVM-RBF), a two-sided Wilcoxon signed-rank test (Table 33), paired by concept, shows that for linear SVM and LR the null hypothesis of no difference cannot be rejected ($p > 0.05$). In contrast, SAE-LR on ISIC frequently yields a significant result ($p < 0.05$), and its win rates indicate that SAE-based adaptation consistently underperforms relative to linear probes.

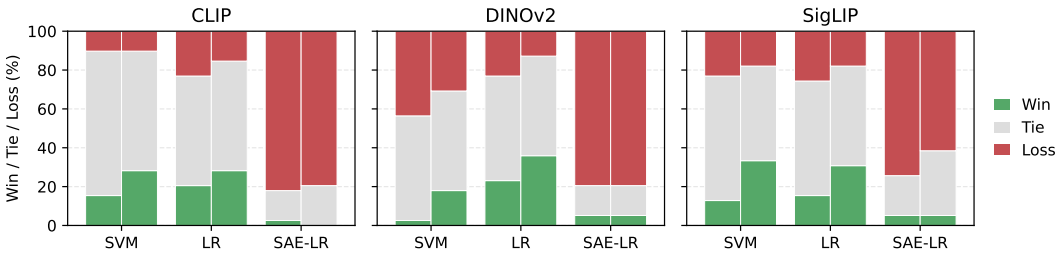

Figure 12: **Win-rate comparison on CelebA: linear CAVs vs. nonlinear probes.** We present the win-rates of linear methods (linear SVM, LR, SAE-LR) against nonlinear baselines *left* TabPFN), (*right* SVM-RBF. Results are rounded to 2 significant digits. While linear SVM and LR maintain competitive performance (high tie/win rates), SAE-LR demonstrates a high loss ratio. Among nonlinear probes, TabPFN generally outperforms SVM-RBF.

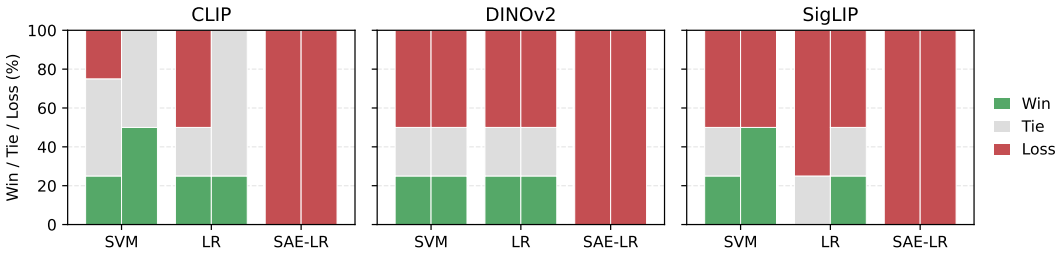

Figure 13: **Win-rate comparison on ISIC: linear CAVs vs. nonlinear probes.** Similar to CelebA Figure 12, we report win-rates against *left* TabPFN, *right* SVM-RBF on all backbones. Linear SVM and LR perform comparably to nonlinear methods, whereas SAE-LR significantly underperforms.

Table 33: **Statistical significance of performance differences (two-sided Wilcoxon signed-rank test).** We compare the AUC distributions of each method against nonlinear probes (TabPFN and SVM-RBF). Values are p-values; $p < 0.05$ (**bold**) indicates a statistically significant difference. High p-values for SVM and LR suggest no significant gap relative to nonlinear probes, whereas SAE-LR frequently on CelebA shows significant underperformance (see also win rates in Figures 12 and 13).

Backbone	CAV	ISIC		CelebA	
		TabPFN	SVM-RBF	TabPFN	SVM-RBF
SigLIP	SVM	0.250	0.875	0.512	0.054
	LR	0.125	1.000	0.634	0.104
	SAE-LR	0.125	0.125	**0.000**	**0.000**
DINOv2	SVM	0.625	0.875	**0.000**	0.329
	LR	0.875	0.875	0.595	**0.001**
	SAE-LR	0.125	0.125	**0.000**	**0.000**
CLIP	SVM	0.875	0.875	0.070	**0.007**
	LR	0.875	0.875	0.918	0.127
	SAE-LR	0.125	0.125	**0.000**	**0.000**

G Extended Metrics Analysis

In this section, we provide detailed visualizations and statistical analyses to substantiate Finding 4, demonstrating the relationships and the critical distinctiveness between vector and steering metrics.

To analyze the interplay among the metrics defined in Section 3.3, we employ a PCA biplot visualization in Figure 14, extending the previous analysis to cover multiple datasets. This allows us to project the multi-dimensional metric space onto two principal axes for CelebA, Waterbirds, and ImageNet-W. From the biplots, we find 5 key observations:

- **Direction 1: predictive fidelity (AUC, F1, MAD).** The biplots reveal that MAD, F1 score, and AUC are highly collinear. This strong correlation implies that MAD provides redundant information when AUC or F1 is available. We further confirm this in Figures 15 and 16, which demonstrate a high correlation between these metrics across all backbones.
- **Direction 2: disentanglement & robustness (MS, CCR).** We observe a strong correlation between Cross Concept Robustness (CCR) and Maximum Similarity (MS). For the CelebA task (Figure 14, middle), these vectors point in opposite directions, which is intuitive as lower MS (more orthogonal vectors) should correspond to higher robustness (higher CCR) against interference from other concepts. In the Waterbirds task (Figure 14, left), CCR and MS appear positively correlated (same direction). As detailed in Table 22 this is an artifact of the dataset's concept binary structure: the "land" and "water" concepts are essentially inverse vectors. Thus, high negative similarity to one implies lower robustness.
- **Orthogonality of steering metrics.** Crucially, side-effect metrics (CD and CCR) and Disparity appear orthogonal to the predictive cluster (AUC/F1) Figures 1 and 14. This lack of correlation, further supported by the quantitative results in Tables 1 and 2, confirms that high predictive performance does not imply steering effects.
- **Impact of CAVs similarity and their robustness.** Strong similarity between two CAVs (dark red or blue color in Figure 17, left) is required for observing a loss in predictive performance after orthogonalization of one CAV against the other (dark blue in Figure 17, right). Notably, the Cross-Concept Robustness is not symmetric, i.e., one CAV might be more robust to orthogonalization than the other in a pair.
- **Impact of vision models.** The overall performance of the CAV extraction methods is strongly influenced by the vision model that serves as the backbone for all computations. The superiority of a backbone depends on the specific task, which is presented in Figures 18–19. Notably, DINOv2 serves as the best backbone for the ISIC task, although SigLIP incurs lower collateral damage. For CelebA, SigLIP is the dominant model, trailing CLIP only on the MS metric.

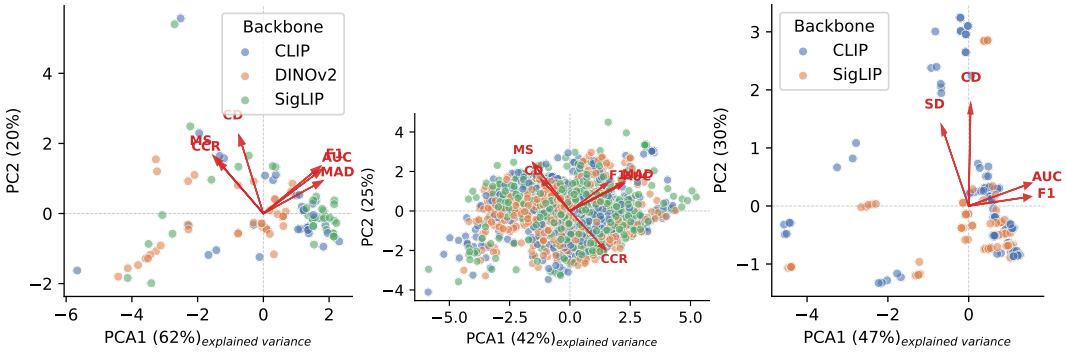

Figure 14: **PCA biplots of metric correlations.** We project all evaluation metrics onto the first two principal components across all backbones for Waterbirds (left), CelebA (middle), and ImageNet-W (right). The arrows indicate the direction of each metric. Predictive metrics (AUC, F1, MAD) consistently overlap, while MS and CCR form distinct correlated directions.

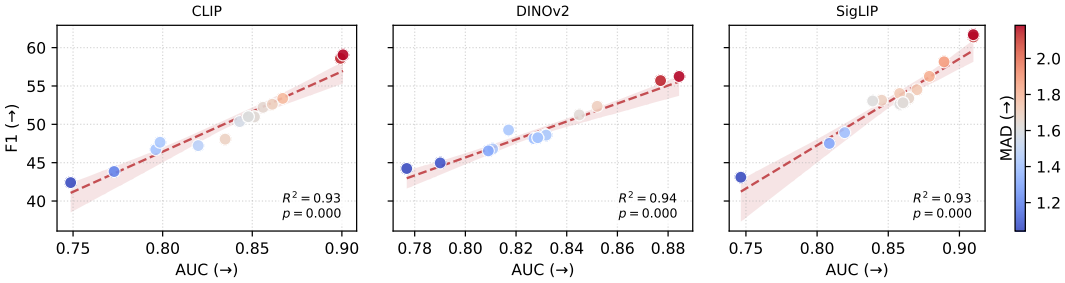

Figure 15: **Correlation between AUC and F1 on CelebA.** We report linear regression statistics (R^2 and p-value) for CLIP (left), DINOv2 (middle), and SigLIP (right). All backbones show a statistically significant positive correlation ($p \approx 0, R^2 > 0.9$), confirming that intrinsic concept detection (AUC) reliably predicts downstream performance (F1).

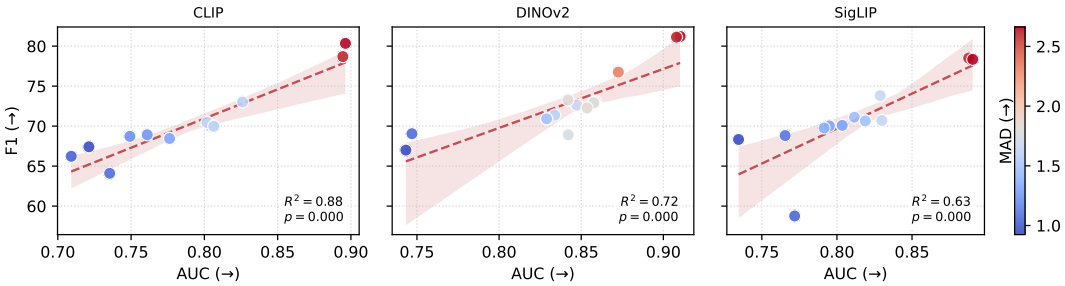

Figure 16: **Correlation between AUC and F1 on ISIC.** Across all backbones, we observe a strong correlation ($p \approx 0, R^2 > 0.6$) between the F1 and AUC metrics. MAD shows a clear and strong correlation with both of AUC and F1, which is why we exclude it from our results.

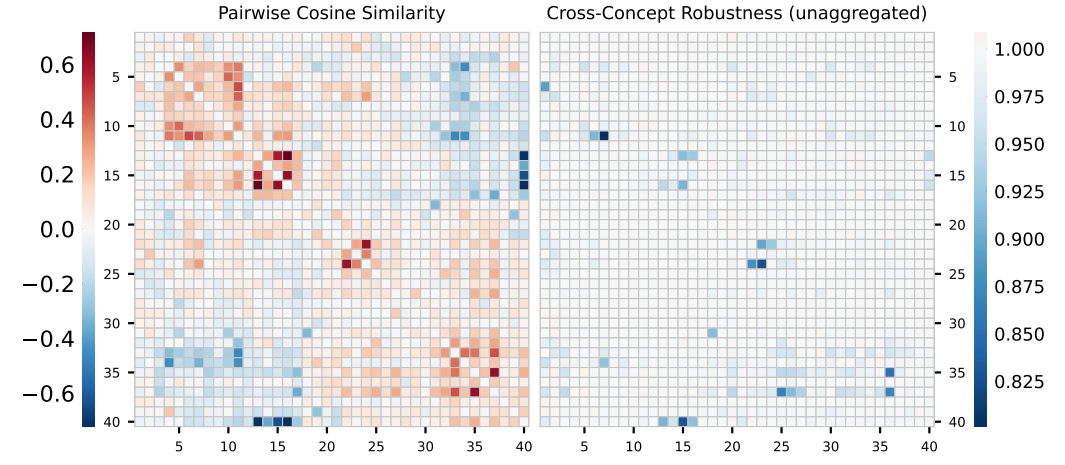

Figure 17: **Heatmap comparison of MS vs CCR on CelebA (using linear SVM).** Left: Pairwise Cosine Similarity between concept vectors. Right: Pairwise Cross Concept Robustness (CCR). We choose linear SVM as CAVs from this method have the highest disentanglement. Although the CCR heatmap is sparser, the non-zero off-diagonal elements largely overlap with high-similarity regions in the left plot. This indicates that high vector similarity (entanglement) drives low robustness (interference), justifying the use of Maximum Similarity (MS) as a worst-case proxy for robustness.

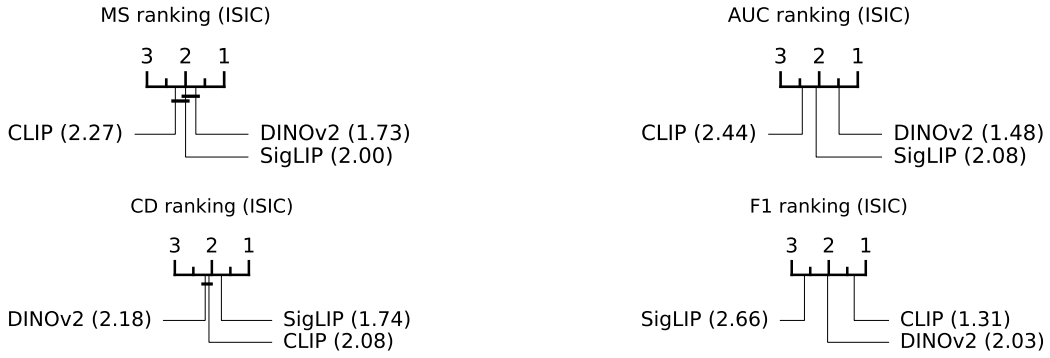

Figure 18: **Critical difference analysis of backbone performance on ISIC** (Wilcoxon's test, $\alpha = 0.1$). We report average rank comparisons for concept similarity (MS, top left), detection performance (AUC, top right), collateral damage (CD, bottom left), and F1 detection score (bottom right). On this dataset, we observe that the optimal backbone varies depending on the specific metric being evaluated. For MS, the performance of all backbones is statistically indistinguishable.

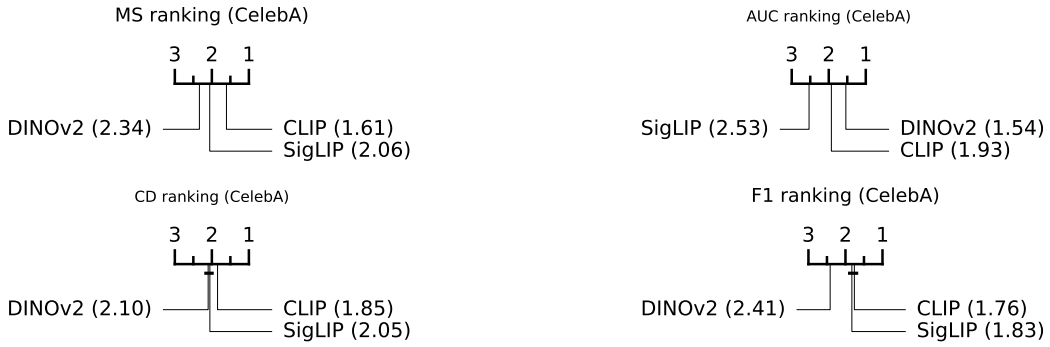

Figure 19: **Critical difference analysis of backbone performance on CelebA** (Wilcoxon's test, $\alpha = 0.1$). We report average rank comparisons for concept similarity (MS, top left), detection performance (AUC, top right), collateral damage (CD, bottom left), and F1 detection score (bottom right). Although SigLIP is the top-performing backbone for overall concept detection on both vector and downstream tasks, CLIP performs significantly better on MS.

H BROADER IMPACT

SWORDBENCH contributes to interpretability, controllability, and fairness research on vision models, motivated by high-stakes settings where representation-level shortcuts cause harm: dermatology classifiers latching onto acquisition artifacts rather than lesion morphology (Bissoto et al., 2020; Wu et al., 2023), demographic correlates contaminating face attribute predictions (Sagawa et al., 2020), and contrastive backbones relying on context rather than the object itself (Wang et al., 2024b). By explicitly measuring *collateral damage* and *cross-concept robustness*, our benchmark discourages a common failure mode in this area: declaring success on separability or downstream accuracy alone, while leaving unrelated capabilities silently degraded.

Steering is, however, fundamentally dual-use. Any method that subtracts a concept can, with a sign flip or tuned coefficient, amplify or inject one: propagating sensitive attributes, weakening refusal behaviors, or covertly altering predictions along demographic or commercial axes. We further caution that strong linear erasure performance does not guarantee that information has been removed from the representation; it removes only the linearly accessible component along the chosen direction (Belrose et al., 2023; Ravfogel et al., 2020), and our own Findings 3–4 show that vector and downstream metrics can disagree in non-obvious ways. SWORDBENCH scores are evidence about a method, not a fairness certificate for any deployed system. To mitigate these risks, we release SWORDBENCH as an open benchmark and invite contributions of new concepts, datasets, and metrics. We prominently report negative results to discourage selective benchmarking.