CONCEPTDIFF: CONTRASTIVE CONCEPT DISCOVERY FOR DATASET COMPARISON

Anonymous authors
Paper under double-blind review

ABSTRACT

Identifying how two datasets differ in their semantics is a basic primitive for model understanding, data analysis, and scientific discovery. It is also hard to find: differences can be subtle, their form unknown in advance, and there are usually several. Current methods require foundation models to inspect the datasets, and propose natural-language hypotheses regarding the difference. This makes them expensive, and it limits them to differences the model thinks to propose. In this paper, we argue that differences should instead be learned directly from data. We formalize this as contrastive concept discovery (CCD), the task of finding a small set of dataset difference concepts that are discriminative (each separates the two datasets), monosemantic (each encodes a single factor), and diverse (together they cover distinct differences). We then introduce ConceptDiff, a lightweight, domain-agnostic method based on training a regularized concept-based classifier designed to learn diverse concepts. Across text, image, and video use-cases, we find that ConceptDiff recovers ground truth dataset differences better than LLM-based, clustering, and sparse-autoencoder baselines, and consistently finds concepts that more discriminative, diverse and monosemantic.

1 INTRODUCTION

Given a target dataset A and a reference dataset B, what does A contain that B does not? This is a task we call contrastive concept discovery (CCD): finding the k concepts that best capture what is present in one dataset but absent from another. This has obvious uses, such as characterizing dataset drift or comparing dataset versions. But it also has several less obvious uses. Consider *safety-critical scene analysis* in autonomous driving. To characterize safety, we need to know when safety-critical events happen for our existing driving policy. For example, do critical scenarios occur more with fast-moving pedestrians, or with poor weather conditions? Or do the failures stem from obscure, rare scenarios that we did not consider a priori? We can answer this using CCD; let A be safety-critical driving clips and B be normal driving footage. The k concepts are then the road scenarios over-represented in safety-critical events. Similarly, CCD can make *AI text detection* explainable, where the question is which textual patterns signal AI use. Here, we let A be AI-generated text and B be human writing. The concepts are then the stylistic patterns that distinguish them. *Model failure analysis* is another use case, where A is the inputs a model gets wrong and B those it gets right, and the concepts are different error modes. The same structure appears well outside machine learning: in biological cell perturbation screens, where A is cells receiving an intervention and B unperturbed controls, shared concepts help determine the statistical effects of the perturbation. In credit risk analysis, A is loans that defaulted and B loans repaid at a matched approval score, and lenders must disclose the primary reasons behind a decision rather than a score. In general, insights about dataset contrasts are essential across many use cases, making CCD a core computational primitive.

Importantly, for a CCD method to be broadly applicable, it must satisfy three desiderata. First, it must be *domain-agnostic*, applicable across data modalities rather than being specific to one domain. Second, it must *scale*, since datasets like driving logs or internet-scale text can contain millions of examples. And finally, it must be able to capture diverse *human-interpretable* semantic differences to provide task-specific insights. Several communities have proposed variations of this task, but their methods fail to achieve these desiderata. In data mining, *emerging pattern mining* and *contrast set mining* search for combinations of attribute values, such as "age $>$ 60 and smoker," that are much more frequent in one group than another (Dong & Li, 1999; Bay & Pazzani, 2001). The closely

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.

related *subgroup discovery* searches for subsets of the data, defined by such attribute rules, in which some outcome of interest is unusually common (Wrobel, 1997). These methods are interpretable by construction, but only because they search over a fixed, predefined vocabulary of attributes, which makes them neither general nor scalable to raw images or text. In statistics and machine learning, *contrastive PCA* finds directions along which the target data varies a lot but the reference data varies little, which surfaces structure specific to the target (Abid et al., 2018). It is general and scales well, but its directions are linear mixtures of raw features, not human-interpretable concepts. More recently, language-model pipelines have a model read samples from two text corpora, propose natural-language hypotheses about how they differ, and keep the hypotheses that best separate them (Zhong et al., 2022; 2023; Dunlap et al., 2024). These are general in principle and produce interpretable output, but they are costly to run at scale, and they focus on outputting a single discriminative hypothesis, not multiple diverse ones. In computational biology, *differential expression analysis* tests each gene for a change in activity between two conditions, and *enrichment analysis* asks whether predefined sets of related genes, such as those in a known pathway, are changed together (Love et al., 2014; Subramanian et al., 2005). Both scale well and are interpretable within biology, but only because they rely on a curated, domain-specific vocabulary of genes and pathways, and are not applicable outside this scope.

To address these issues, we first formalize the task of contrastive concept discovery, discovering k concepts that are (1) *discriminative*, capturing what A contains but B does not; (2) *diverse*, covering distinct differences; and (3) *monosemantic*, meaning that each concept encodes a single domain-specific or task-specific concept. We then propose ConceptDiff, a learning-based framework for contrastive concept discovery. The core idea is to learn a concept-based classifier that discriminates between A and B, with concept regularizers encouraging diversity. Crucially, the classifier operates on the principle of multiple-instance learning, meaning that it outputs A if and only if at least one of the concepts is sufficiently active. One of our main findings is that ensuring diversity and anti-collapse also encourages monosemanticity of concepts, according to VLM-based metrics. The framework is domain-agnostic and lightweight, and so it scales to large datasets, while still producing task-specific concepts.

Distinction from Standard Concept-Based Classifiers. ConceptDiff is closely related to concept-based classifiers (CBCs) like ProtoPNet (Chen et al., 2019) and Concept Bottleneck Models (Koh et al., 2020). The key difference lies in its goals: standard CBCs are symmetric, aiming to recover all discriminative concepts, whereas ConceptDiff only cares about concepts that are present in A but not in B, and is hence inherently asymmetric. This is a critical distinction: consider the driving example, with A the safety-critical clips and B ordinary footage. A symmetric concept-based classifier can encode concepts in B such as empty highways, parked cars, and clear weather; these can be discriminative, but say nothing about what makes a scene dangerous. The asymmetric variant is critical for obtaining safety-specific concepts. In terms of motivation, in standard CBCs the classifier is the product and concepts explain model predictions, whereas in ConceptDiff, the concepts are the product and the classifier is only a means of finding them.

Our overall contributions are as follows:

- We formulate *contrastive concept discovery* as a task of finding k scalar concepts that are discriminative, diverse, and monosemantic, helping us reason about dataset differences.
- We present *ConceptDiff*, a lightweight method that learns k concept detectors to directly discriminate datasets via a regularized concept-based classifier design.
- We perform experimental validation of ConceptDiff across datasets with known concept ground truths on CelebA, and demonstrate that it recovers these latent differences better than baselines. We also perform case studies on (1) safety critical video analysis, (2) explainable AI text detection, (3) error slice discovery, finding that ConceptDiff better recovers discriminative concepts at lower computational cost than LLM-based baselines.

2 Related Work

Describing dataset differences. One popular line of work to extract dataset differences uses language-model pipelines propose natural-language hypotheses about how two datasets differ and

re-rank them with a verifier (Zhong et al., 2022; 2023; Dunlap et al., 2024). The limitation of these pipelines is that they are tuned towards extracting and naming the single best hypothesis as opposed to obtaining a diverse set. In addition, they suffer from high compute cost stemming from language model usage. An alternate line of work uses SAEs (Movva et al., 2025; Jiang et al., 2025) to instead fit a sparse autoencoder first and select discriminative latents afterward. A critical limitation of both these and previous language-model pipelines is that hypothesis generation is independent of discriminability, limiting their ability to extract subtle, discriminative concepts. ProtoSumm (Babbar et al., 2025) also learns an interpretable classifier for learning dataset differences. However, they extracts prototypes, whereas ConceptDiff involves concepts; furthermore ProtoSumm learns prototypes for both datasets symmetrically; ConceptDiff is inherently assymetric and learns only what concepts are predictive of target dataset A.

Concept-based methods. Our work builds on the literature on concept-based methods. The dominant line of work here aims to "reverse-engineer" models via concepts either by clustering segments (Ghorbani et al., 2019), factorizing activations (Fel et al., 2023), or training SAEs (Bricken et al., 2023; Fel et al., 2025). We are instead motivated by discovering structure in datasets as opposed to reverse engineering models. Concept bottleneck models (Koh et al., 2020) and TCAV (Kim et al., 2018) require pre-specification of the concepts before learning, whereas ConceptDiff discovers discriminative concepts in a data-driven manner without human input. Label-free and post-hoc concept bottlenecks (Oikarinen et al., 2023; Yuksekgonul et al., 2023) and SpLiCE (Bhalla et al., 2024) express representations in a vocabulary of text concepts providing flexible concept-based decompositions, but they are applicable to dataset description rather than contrast; as the vocabulary is fixed in advance. Further, they are specific to image domain due to dependency on CLIP-style models. Discover-then-Name (Rao et al., 2024) discovers concepts first but, like the SAE pipelines, does so without regard to the contrast.

Appendix A.1 covers further related work.

3 Contrastive Concept Discovery & ConceptDiff

3.1 Contrastive Concept Discovery (CCD)

We are given two unlabeled datasets $A = \{x_1^A, x_2^A, \dots\}$ and $B = \{x_1^B, x_2^B, \dots\}$ drawn from a common input space $\mathcal{X}$. The task is to find k scalar functions $S_1, \dots, S_k : \mathcal{X} \to \mathbb{R}$, which we call *concepts*, that together explain what distinguishes A from B.

Let $M_{A|B}(S)$ be any performance measure of how well a score S separates A (higher scores) from B. It is oriented so that higher is better (e.g.: balanced accuracy, AUC, 1 - EER). We evaluate the quality of the concepts with three performance metrics.

Discriminativeness. A good concept should separate A from B. We measure this as the average separation across concepts, $\text{Disc} = \frac{1}{k}\sum_{i=1}^{k} M_{A|B}(S_i)$.

Coverage. Good concepts should collectively capture the difference between A and B, so that most examples in A are flagged by at least one concept. To put scores on a common scale, we map each score to its percentile under B, $\tilde{S}_i(x) = \Pr_{x' \sim B}[S_i(x') \le S_i(x)]$, and combine concepts by taking the maximum. We measure how well the combined score separates A from B, $\text{Cov} = M_{A|B}\big(\max_i \tilde{S}_i\big)$.

Monosemanticity. A good concept should track a single domain-specific pattern, so a description of that pattern should be enough to predict the concept's scores. We measure this with a simulation score (Bills et al., 2023), in two steps. First, an explainer agent (human or LLM) is shown $\mathcal{T}_i$, the n highest-scoring examples under S_i, and describes the factor c_i they share. Second, a simulator agent, given only this description, sees a held-out set of examples spanning the range of S_i. For each example x, it predicts $\hat{c}_i(x)$, the degree to which x exhibits c_i. We report the rank correlation between predictions and scores, averaged across concepts, $\text{Mono} = \frac{1}{k}\sum_{i=1}^{k} \rho\big(\hat{c}_i, S_i\big)$, where ρ is Spearman's correlation over the held-out set. Values above 0 mean the description alone lets the simulator predict which examples score higher. In principle, if the explainer agent is a domain expert, and they are able to identify a single interpretable difference, then a concept can be considered monosemantic. In practice, we may use VLM or LLM judges as human proxies.

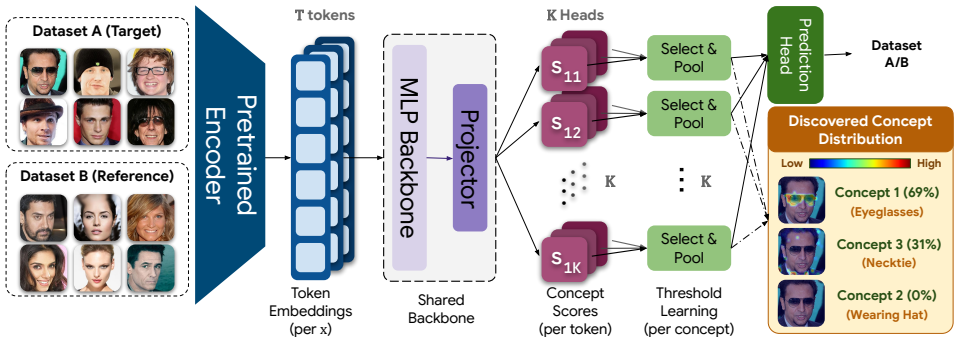

Figure 1: **ConceptDiff architecture.** A frozen pretrained encoder represents each example as T tokens. A shared backbone then transforms each token embedding into a feature vector, which k learned linear detectors score for evidence of each concept. Each detector learns a threshold that softly weights token scores before pooling them into example-level concept evidence. At inference, ConceptDiff returns per-example concept distributions and token-level evidence maps.

3.2 CONCEPTDIFF

We propose ConceptDiff, a framework that discovers k contrastive concepts satisfying the three CCD desiderata described above. The idea is to train a classifier that separates A from B, with an architectural design that includes an intermediate concept layer, and uses the multiple-instance principle, predicting A if and only if at least one concept is sufficiently active. Diversity and anti-collapse regularizers on the concept layer keep the concepts distinct.

The model works in four stages: (1) encode each item into tokens with a pre-trained encoder, (2) score every token against each concept, (3) let each concept select and pool the tokens it claims, and (4) combine the concept evidence into a single prediction (see Fig. 1).

3.2.1 MODEL ARCHITECTURE DESIGN.

Stage 1: Token encoding. A pre-trained encoder maps each item x to a sequence of token embeddings $h_1, \ldots, h_T$. The choice to use token embeddings rather than a single pooled embedding of the whole item is deliberate. While a pooled embedding mixes every part of the input into one vector, tokens keep the parts separate, letting us attach concepts to specific regions of the input.

Stage 2: Concept scoring. A small MLP ϕ maps each token to a feature vector, and k linear detectors score it as $s_{t,i} = w_i^\top \phi(h_t)$, where $s_{t,i}$ measures how strongly token t exhibits concept i. Note that the concepts are thus non-linear functions of the input representations.

Stage 3: Per-concept token selection and global pooling. Most input tokens will not contain information that allows discriminating between A and B, and thus it is critical to identify and localize the discriminative tokens to assign concept labels. This stage handles this, where for each concept, tokens which strongly activate for that concept are selected; and global input-level concept scores are produced. Specifically, each concept has a learned threshold θ_i and softly claims the tokens whose scores clear it via $g_{t,i}$. Its evidence $S_i(x)$ is the mean of its own scores over the claimed tokens:

$$g_{t,i} = \sigma\Big(\tfrac{s_{t,i}-\theta_i}{\tau_{\text{sel}}}\Big), \qquad S_i(x) = \frac{\sum_t g_{t,i}\, s_{t,i}}{1+\sum_t g_{t,i}}, \tag{1}$$

where σ is the logistic sigmoid and τ_{sel} is a selection temperature. Letting concepts pick tokens is inspired by the expert-choice routing mechanism of (Zhou et al., 2022), applied here to concepts, with an inclusion of a learnt threshold.

Stage 4: Classification via Multiple Instance Learning. The item logit is a smooth maximum over concept evidence:

$$f(x) = \tau \log \sum_{i=1}^{k} \exp\big(S_i(x)/\tau\big) - b, \qquad P(y{=}1 \mid x) = \sigma\big(f(x)\big), \tag{2}$$

where $y = 1$ for items from A and $y = 0$ for items from B, τ is a temperature and b is a learnable bias. This is the multiple-instance assumption written out directly: an item belongs to A if some concept fires on it.

3.2.2 REGULARIZED TRAINING OBJECTIVE

The main problem with naive training of the classifier described above is the possibility of concept collapse, i.e., a single concept can learn to encode the entire A vs B classification signal, completely bypassing the CCD desiderata. We therefore use regularizers to ensure that such concept collapse is avoided, and concept diversity is encouraged.

All regularizers are computed over X_A, the items from A in the minibatch, with $N_A = |X_A|$. The balance and commitment losses use each item's concept-assignment distribution, or prevalence, $\pi(x) \in \mathbb{R}^k$, with $\pi_i(x) = \exp(S_i(x)/\tau)/\sum_{j=1}^{k} \exp(S_j(x)/\tau)$, using the same temperature τ as in Eq. equation 2. The regularizers are:

Diversity. For each pair of concepts $i \neq j$, let $r_{i,j}$ be the Pearson correlation between the concept evidences S_i and S_j over the items in X_A. The diversity loss is the mean squared off-diagonal correlation. This prevents the usual collapse in which every detector learns the same A-vs-B direction.

Balance. We include a balance regularizer to ensure that the most activating concept in a batch is spread out, and no concept dominates uniformly. Let $\bar{\pi} = \frac{1}{N_A}\sum_{x \in X_A} \pi(x)$ be the batch-mean prevalence. The balance loss is its normalized negative entropy. In practice, we start from a large value and anneal its weight λ_{bal} during training.

Commitment. We also include a commitment regularizer to ensure that each example is mostly explained by one concept, and not an uninterpretable aggregate of several ones. The commitment loss is the normalized mean entropy of the per-item prevalence:

$$\begin{gathered}\mathcal{L}_{\text{div}} = \frac{1}{k(k-1)}\sum_{i \neq j} r_{i,j}^2, \quad \mathcal{L}_{\text{bal}} = 1 + \frac{1}{\log k}\sum_{i=1}^{k} \bar{\pi}_i \log \bar{\pi}_i, \\ \mathcal{L}_{\text{com}} = -\frac{1}{N_A \log k}\sum_{x \in X_A}\sum_{i=1}^{k} \pi_i(x) \log \pi_i(x).\end{gathered} \tag{3}$$

The total loss is the classification loss (Binary Cross Entropy, BCE) plus the three regularizers:

$$\mathcal{L} = \text{BCE}\big(f(x), y\big) + \lambda_{\text{div}}\mathcal{L}_{\text{div}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}} + \lambda_{\text{com}}\mathcal{L}_{\text{com}}. \tag{4}$$

4 EXPERIMENTS

We first evaluate ConceptDiff on constructed image-dataset pairs with known semantic groups, testing whether it recovers multiple differences rather than merely separating the datasets (Sec. 4.2). We then investigate the usefulness of our approach across three case studies: generated versus human or edited writing (Sec. 4.3), crash versus normal-driving videos (Sec. 4.4), and discovering error slices by comparing a model's incorrect and correct predictions (Sec. 4.5).

4.1 COMMON EXPERIMENTAL PROTOCOL

Baselines. We compare four approaches to discovering dataset differences and, where applicable, standardize their outputs to K concepts with continuous example-level scores. *Clustering:* k-means groups target examples and scores each concept by similarity to its centroid. *Sparse features:* We adapt HypotheSAEs (Movva et al., 2025) to learn a sparse feature dictionary, then select the K directions with the highest validation AUROC for distinguishing $\mathcal{D}_A$ from $\mathcal{D}_B$. *Language-based proposals:* We use VisDiff (Dunlap et al., 2024) for images and TextDiff (Zhong et al., 2022) for text, adapting each to propose and score natural-language descriptions of dataset differences. *Spatial prototypes:* ProtoSumm (Babbar et al., 2025) learns discriminative prototypes and scores examples by their maximum patch similarity; our adaptation trains a feature projection on top of a frozen encoder. For k-means and HypotheSAEs, we evaluate both frozen pretrained features (*pre.*) and features from a classifier trained to distinguish the two datasets (*cls.*). Appendix A.2 provides baseline implementation details.

Evaluation. We evaluate the three criteria defined in Sec. 3 and report reproducibility as an additional diagnostic. *Discriminability:* Concept-level equal error rate (cEER↓) measures how well

Table 1: Recovery of planted image differences. Across 12 constructed comparisons ($K = 3$), ConceptDiff leads in group recovery, discriminability, coverage, and regional monosemanticity. Subscripts give sample SD across three seed-level scenario averages. **Bold** and underline mark the best and second-best score; whole-image and regional scores are ranked separately.

	Discriminability		Coverage		Monosemanticity		Reliability
Method	ARI (↑)	cEER (↓)	Act (↑)	Cov (↑)	ΔCoh (↑)	ρ (↑)	Stab (↑)
k-means (pre.)	$0.268_{0.002}$	$0.189_{0.001}$	$\mathbf{1.00}_{0.00}$	$0.395_{0.001}$	$\mathbf{0.173}_{0.000}$	0.496	**0.990**
k-means (cls.)	$\underline{0.275}_{0.029}$	$0.111_{0.002}$	$\underline{0.98}_{0.02}$	$0.747_{0.004}$	$0.077_{0.006}$	0.543	0.469
HypotheSAEs (pre.)	$0.117_{0.024}$	$0.155_{0.007}$	$\mathbf{1.00}_{0.00}$	$0.533_{0.008}$	$\underline{0.118}^{\mathcal{R}}_{0.003}$	$\underline{0.510}^{\mathcal{R}}$	0.157
HypotheSAEs (cls.)	$0.131_{0.033}$	$\underline{0.099}_{0.006}$	$0.86_{0.05}$	$\underline{0.751}_{0.009}$	$0.065_{0.007}$	0.450	0.213
VisDiff	$0.139_{0.030}$	$0.183_{0.013}$	**1.00**	$0.488_{0.020}$	$\underline{0.123}_{0.007}$	$\underline{0.616}$	0.248
ProtoSumm	$0.153_{0.060}$	$0.178_{0.033}$	$\mathbf{1.00}_{0.00}$	$0.349_{0.052}$	$0.066^{\mathcal{R}}_{0.004}$	$0.273^{\mathcal{R}}$	0.125
ConceptDiff (ours)	$\mathbf{0.460}_{0.020}$	$\mathbf{0.062}_{0.001}$	$\mathbf{1.00}_{0.00}$	$\mathbf{0.827}_{0.007}$	$\mathbf{0.132}^{\mathcal{R}}_{0.005}$	$\mathbf{0.538}^{\mathcal{R}}$	$\underline{0.522}$

$^{\mathcal{R}}$ indicates monosemanticity calculated based on concept region.

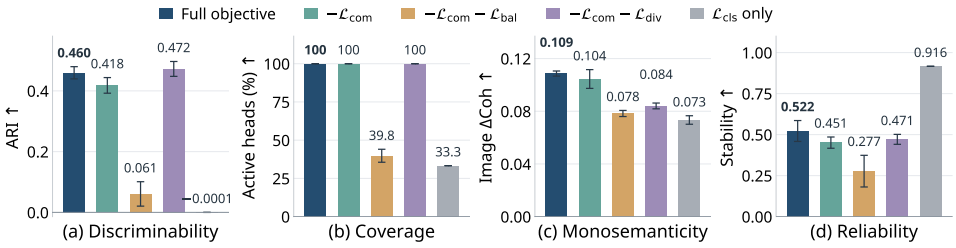

Figure 2: Evaluation of complementary roles of the loss terms across concept recovery, active concepts, coherence, and stability. Training on classification only objective leads to concept collapse.

each active concept separates its assigned target examples from reference examples. On constructed testset, adjusted Rand index (ARI↑) additionally measures recovery of the known target groups. *Coverage:* Cov↑ is the fraction of target examples detected by at least one concept, using a threshold calibrated on reference validation examples at a nominal 5% false-positive rate. We report the active-concept fraction (Act↑) alongside it to expose solutions in which several concepts collapse. *Monosemanticity:* Concept coherence (ΔCoh↑) measures consistency in an independent embedding space, while the simulation score (ρ↑) tests whether a concept description predicts its activations on held-out examples. *Reliability:* Cross-seed stability (Stab↑) measures whether independent runs produce similar assignments; we interpret it alongside Act because a consistently collapsed solution can also be highly stable. Appendix A.3 gives the full protocols.

4.2 Multi-Concept Recovery on Constructed Image Testsets

Benchmark construction. We construct 12 image-dataset pairs $(\mathcal{D}_A, \mathcal{D}_B)$ with known target groups to evaluate recovery of the planted differences. Seven cases use CelebA-HQ Karras et al. (2017) attributes, including accessories, hair characteristics, and co-occurring attributes; five use ImageNet-1K Deng et al. (2009), ImageNet-R Hendrycks et al. (2021), and ImageNet* Vendrow et al. (2023). Each target set has three annotated groups, and all methods receive the same budget of $K = 3$ concepts. Appendix A.4.1 details the construction of each testset.

Recovering and localizing differences. On the constructed testcases, ConceptDiff most accurately recovers the planted groups while achieving the lowest cEER and highest coverage (Table 1). Pretrained k-means yields stable, coherent whole-image groups but weaker recovery and coverage. On the other hand, classifier-trained features improve separation and coverage for k-means and HypotheSAEs, while reducing k-means coherence. ConceptDiff also achieves the highest regional coherence and simulation score among methods with patch-level attributions. In the CelebA-HQ example, its eyeglasses, necktie, and hat concepts highlight the regions containing those attributes (Fig. 3). Thus, the recovered groups are supported by localized evidence as well as dataset-level discrimination. Additional qualitative examples appear in Appendix A.4.2.

Table 2: **Concept discovery on EditLens: Gemini-generated vs. human writing** ($K = 4$; 492 test pairs). Best and second-best means are **bold** and underlined, respectively, excluding Act.

	Discriminability	Coverage		Monosemanticity		Reliability
Method	cEER (↓)	Act (↑)	Cov (↑)	ΔCoh (↑)	ρ (↑)	Stab (↑)
k-means (pre.)	$0.0270_{0.0000}$	$1.00_{0.00}$	$0.801_{0.000}$	$\underline{0.271}_{0.000}$	$\underline{0.562}$	**1.000**
k-means (cls.)	$\mathbf{0.0002}_{0.0002}$	$1.00_{0.00}$	$\mathbf{1.000}_{0.000}$	$0.146_{0.032}$	0.456	0.257
HypotheSAEs (pre.)	$0.0962_{0.0129}$	$1.00_{0.00}$	$\underline{0.839}_{0.279}$	$\mathbf{0.277}_{0.030}$	0.507	0.089
HypotheSAEs (cls.)	$0.0038_{0.0066}$	$0.42_{0.29}$	$\mathbf{1.000}_{0.000}$	$0.123_{0.041}$	0.147	0.004
TextDiff	$0.2532_{0.0200}$	$1.00_{0.00}$	$0.279_{0.147}$	$0.253_{0.032}$	0.440	0.150
ConceptDiff (ours)	$\underline{0.0021}_{0.0004}$	$1.00_{0.00}$	$\mathbf{1.000}_{0.000}$	$0.227_{0.002}$	**0.670**	$\underline{0.927}$

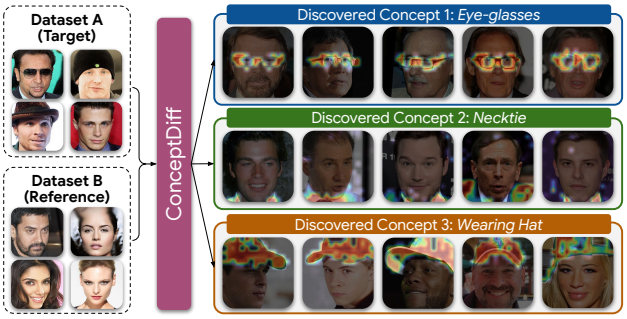

Figure 3: ConceptDiff recovers differentiating concepts (eyeglasses, neckties, and hats) in CelebA-HQ faces with and without accessories. Highlighted regions identify the supporting difference.

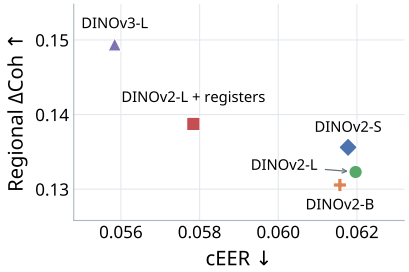

Figure 4: Better token representations can further improve the quality of discovered differences. DINOv3-L discovers the most discriminative and coherent differences.

Effects of the training objective and backbone. Fig. 2 shows how the training objective supports recovery of multiple concepts. Removing commitment reduces ARI, while removing balance leaves most concept heads inactive. Furthermore, classification-only training yields a stable solution but collapses to a single active concept. With the full objective, DINOv3-L improves both cEER and regional coherence over DINOv2-L (Fig. 4), suggesting that stronger visual representations further improve the recovered differences.

4.3 Case Study: Generated Writing Compared with Human and Edited Text

Experimental setup. We apply ConceptDiff to four-token groups of contextual embeddings from a frozen T5-large encoder in place of image patches, leaving the objective and aggregation unchanged. On EditLens Thai et al. (2026), we compare Gemini 2.5 Flash generations with source-matched human writing (492 test pairs) and Gemini-edited writing (81 test pairs), using $K = 4$ in both cases. We additionally evaluate three pairwise comparisons among Claude, Gemini, and GPT on StoryScope Russell et al. (2026); their results appear in Appendix A.5.

Results. On both EditLens case studies, ConceptDiff achieves near-zero cEER with full coverage and a high simulation score while keeping all four concepts active (Table 2). On the other hand, classifier-based k-means separates the sources slightly better, but produces concepts that are less stable and less describable. A similar pattern holds across the three StoryScope case studies: ConceptDiff maintains all heads with near-zero cEER, whereas classifier-based baselines often leave heads inactive despite strong source separation (Appendix A.5). Meanwhile, frozen clustering sometimes yields more coherent or stable groups, so these results support consistent multi-concept discrimination rather than superiority on every measure.

Discovered differences. The text-concepts expose distinct ways generated writing differs from its matched human references. Fig. 5 highlights supporting spans in sensory fiction, explanatory prose, news, and reviews, showing that the contrast draws on different cues across writing genres. Interestingly, increasing K reveals finer distinctions within these broad groups. For example, sensory

Table 3: Concept discovery in car crash videos. Crashes versus normal driving with frozen V-JEPA 2.1 ($K = 6$; three seeds). Best and second-best results are **bold** and underlined, respectively. ΔCoh uses centered X-CLIP embeddings to evaluate monosemanticity.

	Discriminability	Coverage		Monosemanticity	Reliability
Method	cEER ↓	Act ↑	Cov ↑	ΔCoh ↑	Stab ↑
k-means (pre.)	$0.089_{0.003}$	$\mathbf{1.00}_{0.00}$	$0.439_{0.004}$	$\underline{0.141}_{0.002}$	**0.981**
k-means (cls.)	$\mathbf{0.000}_{0.000}$	$\underline{0.89}_{0.10}$	$\mathbf{1.000}_{0.000}$	$0.109_{0.016}$	0.232
HypotheSAEs (pre.)	$\underline{0.017}_{0.013}$	$0.61_{0.10}$	$0.978_{0.010}$	$0.043_{0.006}$	$\underline{0.904}$
HypotheSAEs (cls.)†	$\mathbf{0.000}_{0.000}$	$0.22_{0.10}$	$\mathbf{1.000}_{0.000}$	$0.100_{0.029}$	0.333
ProtoSumm†	$\mathbf{0.000}_{0.000}$	$0.22_{0.10}$	$\underline{0.979}_{0.036}$	$0.018_{0.031}$	0.333
ConceptDiff (ours)	$\mathbf{0.000}_{0.0001}$	$\mathbf{1.00}_{0.00}$	$\mathbf{1.000}_{0.000}$	$\mathbf{0.151}_{0.008}$	0.383

† indicates concept collapse due to low active concepts.

Table 4: Slice discovery on DCBench CelebA (143 slices, 142 problems; $K = 5$; 3 seeds). Bold indicates best performance. †Uses true labels and predicted probs.

Method	Slice AP (↑)	cEER (↓)
k-means (pre.)	$0.742_{0.028}$	$0.309_{0.009}$
k-means (cls.)	$0.726_{0.028}$	$0.289_{0.013}$
SAE (pre.)	$0.740_{0.030}$	$0.242_{0.010}$
SAE (cls.)	$0.687_{0.029}$	$0.248_{0.012}$
ProtoSumm	$0.732_{0.029}$	$0.241_{0.005}$
Domino†	$0.718_{0.029}$	$\mathbf{0.162}_{0.012}$
ConceptDiff (ours)	$\mathbf{0.807}_{0.023}$	$0.169_{0.007}$

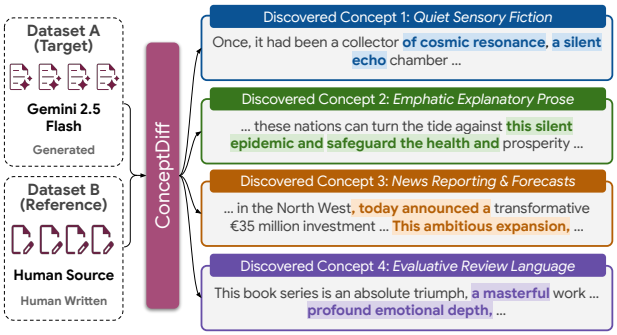

Figure 5: Qualitative examples of localized textual differences. Highlighted spans support concepts discovered in generated versus human writing.

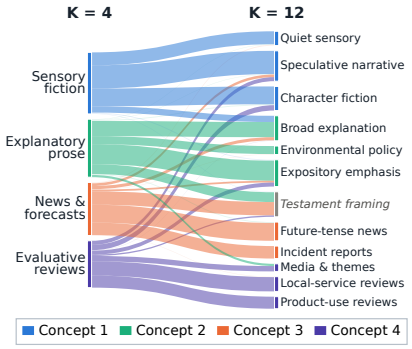

Figure 6: Broader writing differences divide into more specific groups as the K increases from 4 to 12 (see Appendix A.5.3).

fiction separates into quiet sensory, speculative, and character-focused narratives at $K = 12$ (Fig. 6). Together, these examples show that ConceptDiff can isolate broad, recurring differences and resolve them into finer-grained patterns as the concept budget grows, while grounding the discovered concepts in supporting text spans. The full concept evolution with K is presented in Appendix A.5.3.

4.4 Case Study: Understanding Safety-Critical Driving Scenarios

Experimental setup. We compare onset-aligned crash clips with normal driving from the Car Crash Dataset Bao et al. (2020). All methods use frozen V-JEPA 2.1 features extracted from 16 frames at 4 fps, with a concept budget of $K = 6$. This setting tests whether the methods can identify multiple coherent differences without ground-truth concept labels in real-world setting.

Results. The video results reveal a gap between separating the two sources and using the concept budget meaningfully. Several baselines achieve near-zero cEER while assigning most clips to only one or two heads; frozen k-means produces more stable groups but has weaker discrimination and coverage. ConceptDiff keeps all six heads active, separates the sources, and achieves the highest coherence (Table 3). Fig. 7 illustrates two of its discovered groups: night driving, with evidence concentrated in the sky, and a motorcycle rider's viewpoint, with evidence concentrated on the road. These groups capture recognizable properties of the clips, but do not isolate distinct collision mechanisms. We believe this may reflect limitations of the V-JEPA 2.1 features used here. As the image backbone comparison suggests, stronger representations can improve concept discovery; video features that better encode motion and interactions may similarly yield more specific collision concepts.

4.5 Error Slice Discovery: Model Errors vs. Correct Predictions

One important application of ConceptDiff is identifying coherent error patterns from model errors, which is called an error slice. Here, we set A to be the validation examples a trained model misclas-

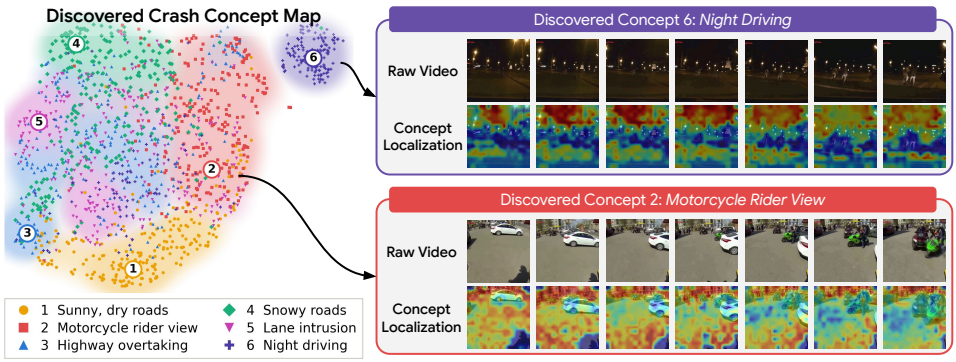

Figure 7: Concepts discovered in crash videos. Left: clips grouped by their assigned concepts. Right: example videos and their localized evidence heatmap for the "night driving" and "motorcycle rider view" concepts.

sifies and B to those it classifies correctly, so each ConceptDiff head defines a candidate *error slice*, i.e., a set of coherent errors, whose efficacy we evaluate here.

We use known classifier errors from DCBench (Eyuboglu et al., 2022), stemming from injected spurious correlations, to evaluate the efficacy of error slice discovery. Specifically, out of a total of 520 CelebA-based problem instances, we experiment with a subset of 142 problem instances, filtered from the criteria that the error slice must have least 70% of annotated members being misclassified. This subset better matches our notion of error slices, as coherent subsets of error samples that are incorrect for the same reason. On the other hand, the majority of DCBench instances contain problems where $< 50\%$ of the annotated ground-truth slice were correctly classified, which we omit. For metrics, we use *Error-only AP* to measure recovery of known error slices, and*cEER* is the per-concept discriminability used in previous experiments. We average seeds, then slices, classes, and problems, and compute 95% intervals from 1,000 paired bootstrap resamples over problems to report results.

We observe in Table 4 that ConceptDiff outperforms baselines such as k-means, SAEs, ProtoSumm and Domino; and obtains error clusters most aligned with the ground truth error slices. Further, ConceptDiff also obtains highly discriminative concepts on par with Domino.

5 Conclusion

ConceptDiff grew out of the practical use-case of analyzing safety-critical events in autonomous driving logs. In that setting, existing LLM-based methods did not scale to the size of the logs, and SAE- and clustering-based methods produced generic concepts rather than the safety-relevant ones we needed. ConceptDiff addressed both problems by learning differences directly from the contrast between two datasets, using a lightweight model. Our experiments show that the same framework applies across images, text, and video, as well as to error slice discovery. More broadly, we believe the framing of contrastive concept discovery matters beyond any single method. Treating dataset comparison as learning a small set of discriminative, monosemantic, and diverse concepts gives the problem a common target and common metrics, and we hope future work builds on it.

Limitations. Because ConceptDiff was empirically driven, we lack a rigorous first-principles account of its effectiveness, and hence are unable to present theoretical results on concept quality. Another limitation is that ConceptDiff promotes monosemanticity only indirectly, through its diversity and commitment losses. We tried promoting it directly with perceptual regularizers, such as robustness regularizers, but they were ineffective. Thus principled theory for ConceptDiff, and whether monosemanticity-promoting regularizers exist, remain open problems.

AI USE STATEMENT

We used generative AI tools extensively for brainstorming architecture design ideas, and experimentation, including coding, implementing baselines, and iterative refinement of our method ConceptDiff. We have not used generative AI tools for developing the conceptual framework or the metrics, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for writing an initial draft of the paper which we edited, and for initial literature survey. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work introduces a algorithmic contribution to dataset difference literature, helpful in data analysis and debugging models; as such we do not see any explicit ethical issues that may arise here.

REPRODUCIBILITY STATEMENT

We ensure that the main paper and the supplementary material contains the necessary details to reproduce the experiments. This includes model design, metrics and dataset details. We plan to provide the full source code to reproduce the experiments following internal review.

REFERENCES

Abubakar Abid, Martin J. Zhang, Vivek K. Bagaria, and James Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications*, 9(1):2134, 2018. doi: 10.1038/s41467-018-04608-8.

Varun Babbar, Zhicheng Guo, and Cynthia Rudin. "what is different between these datasets?" a framework for explaining data distribution shifts. *Journal of Machine Learning Research*, 26 (180):1–64, 2025. URL https://www.jmlr.org/papers/v26/24-0352.html.

Wentao Bao, Qi Yu, and Yu Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *ACM Multimedia Conference*, May 2020.

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

Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 73–82. ACM, 2001.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P. Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/adf7ee2dcf142b0e11888e72b43fcb75-Abstract.html.

Dami Choi, Vincent Huang, Kevin Meng, Daniel D. Johnson, Jacob Steinhardt, and Sarah Schwettmann. Scaling automatic neuron description. Transluce blog, 2024. URL https://transluce.org/neuron-descriptions.

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

Greg d'Eon, Jason d'Eon, James R. Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pp. 1962–1981, 2022. doi: 10.1145/3531146.3533240.

Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 43–52, 1999. doi: 10.1145/312129.312191.

Lisa Dunlap, Yuhui Zhang, Xiaohan Wang, Ruiqi Zhong, Trevor Darrell, Jacob Steinhardt, Joseph E. Gonzalez, and Serena Yeung-Levy. Describing differences in image sets with natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24199–24208, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Dunlap_Describing_Differences_in_Image_Sets_with_Natural_Language_CVPR_2024_paper.pdf.

Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=FPCMqjI0jXN.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2711–2721, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Fel_CRAFT_Concept_Recursive_Activation_FacTorization_for_Explainability_CVPR_2023_paper.html.

Thomas Fel, Ekdeep Singh Lubana, Jacob S. Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba E. Ba, and Talia Konkle. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 16543–16572, 2025. URL https://proceedings.mlr.press/v267/fel25a.html.

Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1902.03129.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=99RpBVpLiX.

Nick Jiang, Xiaoqing Sun, Lisa Dunlap, Lewis Smith, and Neel Nanda. Interpretable embeddings with sparse autoencoders: A data analysis toolkit, 2025. URL https://arxiv.org/abs/2512.10092.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 29223–29264, 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *PMLR*, pp. 2668–2677, 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Discovering and mitigating visual biases through keyword explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11082–11092, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Kim_Discovering_and_Mitigating_Visual_Biases_through_Keyword_Explanation_CVPR_2024_paper.html.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *PMLR*, pp. 5338–5348, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, 2014. doi: 10.1186/s13059-014-0550-8.

Rajiv Movva, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson. Sparse autoencoders for hypothesis generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 44997–45023, 2025. URL https://proceedings.mlr.press/v267/movva25a.html.

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

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

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 48393–48421, 2025. URL https://proceedings.mlr.press/v267/paulo25a.html.

Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *Computer Vision – ECCV 2024*, volume 15135 of *Lecture Notes in Computer Science*, pp. 444–461. Springer, 2024. doi: 10.1007/978-3-031-72980-5_26.

Jenna Russell, Rishanth Rajendhran, Chau Minh Pham, Mohit Iyyer, and John Wieting. StoryScope: Investigating idiosyncrasies in AI fiction, 2026. URL https://arxiv.org/abs/2604.03136.

Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43): 15545–15550, 2005. doi: 10.1073/pnas.0506580102.

Katherine Thai, Bradley Emi, Elyas Masrour, and Mohit Iyyer. EditLens: Quantifying the extent of AI editing in text. In *International Conference on Learning Representations (ICLR)*, 2026.

Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865*, 2023.

Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Principles of Data Mining and Knowledge Discovery (PKDD)*, volume 1263 of *Lecture Notes in Computer Science*, pp. 78–87. Springer, 1997. doi: 10.1007/3-540-63223-9_108.

Sriram Yenamandra, Pratik Ramesh, Viraj Prabhu, and Judy Hoffman. FACTS: First amplify correlations and then slice to discover bias. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4794–4804, 2023.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023.

Ruiqi Zhong, Charlie Snell, Dan Klein, and Jacob Steinhardt. Describing differences between text distributions with natural language. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *PMLR*, pp. 27099–27116, 2022. URL https://proceedings.mlr.press/v162/zhong22a.html.

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. Goal driven discovery of distributional differences via language descriptions. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2302.14233.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7114, 2022.

A Appendix

A.1 Extended Related Work

Section 2 covers the work closest to ConceptDiff. This appendix discusses the remaining related work.

Slice discovery. Slice discovery audits a trained classifier by finding subsets of evaluation data, or slices, on which it fails. These methods share the same inputs: a model, labeled examples, and the model's per-example errors or losses. Domino (Eyuboglu et al., 2022), our primary baseline, fits an error-aware mixture model in a cross-modal embedding space and names each slice with text. Spotlight (d'Eon et al., 2022) searches the model's representation space for a contiguous high-loss region, Jain et al. (2023) fit a direction per class that separates errors from correct predictions, FACTS (Yenamandra et al., 2023) amplifies a model's spurious correlations before slicing, and B2T (Kim et al., 2024) extracts keywords from captions of misclassified images. In each case, the output is a set of whole examples, or an error direction tied to the failures of a single model. ConceptDiff applied to slice discovery instead takes two datasets in where A holds a model's errors and B its correct predictions. Its output is k decorrelated concept detectors that score new items and localize the tokens responsible for each concept; making it distinct from the slice discovery literature.

Evaluating concepts. Evaluating the human-interpretability of concept representations is an important theme in interpretability. Early works like Network Dissection (Bau et al., 2017) measure the interpretability of individual neurons by their overlap with densely annotated concepts, and CLIP-Dissect (Oikarinen & Weng, 2023) removes the need for annotations by matching units to concepts with CLIP. We score monosemanticity with the explainer–simulator protocol of Bills et al. (2023), which has been scaled to every neuron of a model (Choi et al., 2024) and to millions of SAE features (Paulo et al., 2025). SAEBench (Karvonen et al., 2025) scores SAEs on downstream tasks, and Pach et al. (2025) measure the monosemanticity of SAE features in vision–language models using CLIP models as reference. In our work, we use both LLM-based and CLIP-based models for constructing monosemanticity measures.

A.2 Baseline Implementation Details

We describe the baseline implementations used for the constructed image benchmark. Each method produces K candidate concepts for $\mathcal{D}_A$ and a continuous score $s_k(x)$ for each concept–example pair. Concepts are learned from the training split, with validation data used for feature selection where specified. Ground-truth semantic groups are used only for evaluation.

k-means. We fit K centroids to representations of target training examples. Each example is assigned to its nearest centroid, and cosine similarities to the centroids provide its concept scores. The pretrained variant (*pre.*) uses pooled embeddings from a frozen encoder. The classifier variant (*cls.*) uses penultimate-layer features from an MLP trained to distinguish $\mathcal{D}_A$ from $\mathcal{D}_B$ over the same frozen encoder. This comparison tests how adapting the representation for dataset discrimination affects the resulting groups.

HypotheSAEs. Our adaptation of HypotheSAEs (Movva et al., 2025) trains a sparse autoencoder on representations from both datasets. We rank the learned features by validation AUROC for distinguishing $\mathcal{D}_A$ from $\mathcal{D}_B$ and retain the top K as concepts. The pretrained variant operates on frozen patch features, with each image scored by its maximum feature activation across patches. The classifier variant instead operates on penultimate-layer features from an MLP trained for dataset discrimination over frozen encoder features.

Language-based proposals. We use VisDiff (Dunlap et al., 2024) for images and TextDiff (Zhong et al., 2022) for text. Both propose natural-language descriptions of dataset differences and score examples against these descriptions. Our VisDiff adaptation uses a multimodal language model to inspect sampled training images from both datasets directly. Descriptions are scored by CLIP

text–image similarity, ranked by training AUROC, and the top K are retained. Our TextDiff adaptation uses a language model to propose textual differences and scores examples by text-embedding similarity to the selected descriptions.

ProtoSumm. Our adaptation of ProtoSumm (Babbar et al., 2025) learns K prototypes for each dataset through a trainable projection of frozen patch features. The objective combines dataset classification, clustering, separation, and cross-view prototype-affinity losses. After training, each prototype is projected onto its nearest projected training patch from the corresponding dataset. The K prototypes associated with $\mathcal{D}_A$ define the reported concepts. Each concept scores an image by its maximum cosine similarity to the image's projected patches.

A.3 Evaluation Protocol

We evaluate discriminability, coverage, and monosemanticity, and report cross-seed stability as an additional reliability measure. Let $S_k(x)$ denote the example-level score of concept k, with higher scores indicating stronger evidence. We retain the nearest-centroid assignments for k-means; for the other methods, we use $a(x) = \arg\max_k S_k(x)$.

Concept discriminability and recovery. Each concept is evaluated as a detector for the target examples assigned to it. Its positive set is

$$\mathcal{P}_k = \{x \in \mathcal{D}_A^{\text{test}} : a(x) = k\},$$

and its negative set is $\mathcal{D}_B^{\text{test}}$. Using $S_k(x)$, we construct the empirical ROC curve and select the point minimizing the absolute difference between the false-positive and false-negative rates. We estimate EER_k as the average of these two rates at that point.

A concept is considered active if it has at least five assigned target test examples. Defining $\mathcal{K}_{\text{act}} = \{k : |\mathcal{P}_k| \geq 5\}$, we report

$$\text{cEER} = \frac{1}{|\mathcal{K}_{\text{act}}|} \sum_{k \in \mathcal{K}_{\text{act}}} \text{EER}_k, \qquad \text{Act} = \frac{|\mathcal{K}_{\text{act}}|}{K}.$$

cEER is undefined when no concept is active. Because each method determines its own positive sets, we interpret cEER jointly with Act and coverage. On constructed testbeds, we additionally report adjusted Rand index (ARI) between target test assignments and ground-truth semantic groups. ARI is invariant to permutations of concept labels and requires no label matching.

Coverage and calibration. Coverage measures whether at least one concept detects each target example. We define the union score $u(x) = \max_k S_k(x)$ and calibrate one threshold per method, task, and run:

$$\tau = Q_{0.95}\big(\{u(x) : x \in \mathcal{D}_B^{\text{val}}\}\big).$$

Here, $Q_{0.95}$ is the empirical 95th percentile, computed using linear interpolation. Coverage is

$$\text{Cov} = \frac{1}{|\mathcal{D}_A^{\text{test}}|} \sum_{x \in \mathcal{D}_A^{\text{test}}} \mathbf{1}\{u(x) \geq \tau\}.$$

The realized reference false-positive rate is computed by applying the same threshold and inequality to $\mathcal{D}_B^{\text{test}}$.

This procedure uses raw concept scores, without per-concept percentile normalization. The nominal 5% calibration level does not guarantee a 5% test false-positive rate: finite validation samples, tied scores, and differences between validation and test distributions can affect the realized rate. We record this rate as a calibration diagnostic.

Concept coherence. For each concept, we rank all target test examples by $S_k(x)$ and select the highest- and lowest-scoring sets, $\mathcal{T}_k$ and $\mathcal{B}_k$. Each contains $n = \min(25, \lfloor |\mathcal{D}_A^{\text{test}}|/2 \rfloor)$ examples. Selection is independent of concept assignment. Using embeddings from an independent evaluation encoder, we compute

$$\Delta\text{Coh}_k = \overline{\cos}(\mathcal{T}_k, \mathcal{T}_k) - \overline{\cos}(\mathcal{T}_k, \mathcal{B}_k), \tag{5}$$

where $\overline{\cos}$ denotes mean pairwise cosine similarity, excluding self-pairs within a set. We report the mean over all K concepts.

We use SigLIP2 for images, X-CLIP for videos, and MPNet for text. Higher coherence indicates that strongly activating examples are more mutually similar than they are to weakly activating examples in this independent representation. For the centered variant, $\Delta\mathrm{Coh}^c$, we subtract the mean embedding of the full target pool across training, validation, and test splits, then normalize each embedding before applying Eq. 5. The video results use this centered X-CLIP variant.

Region-based coherence. For image methods with patch-level attributions, $\Delta\mathrm{Coh}^{\mathcal{R}}$ applies Eq. 5 to evidence crops from the same top and bottom examples. For each image and concept, we select the $\max(4, \lfloor P/10 \rfloor)$ highest-scoring patches, where P is the number of patches. We form their bounding box, add margins of 8% of the image width on each horizontal side and 8% of the image height on each vertical side, and clip the box to the image boundaries. The resulting crop is padded to a square with a white background and encoded with SigLIP2.

For centered region coherence, the centering mean is computed over all selected top and bottom crop embeddings across concepts. Whole-image and region scores are reported and compared separately.

Simulation scoring. Following the explainer–simulator approach (Bills et al., 2023), we test whether a natural-language description predicts a concept's scores on held-out examples. For each concept, an explainer receives up to 12 highest-scoring target training examples and describes their shared pattern in one or two sentences. A separate simulator call receives this description and up to 24 target validation examples, sampled at approximately uniform positions in the concept-score ranking and presented in descending score order. It rates each example from 0 to 10 without receiving the numerical concept scores.

We compute Spearman correlation between the simulator ratings and $S_k(x)$ and average across successfully scored concepts. Undefined correlations, including those caused by constant ratings, are assigned zero. Unparseable simulator responses are retried once; concepts with a second parsing failure are excluded from the average. For text, the models receive excerpts from the beginning, middle, and end of each document, with up to 170 words per excerpt.

For the image region-based variant $\rho^{\mathcal{R}}$, the explainer sees full images with brightness modulated by the concept's patch scores, emphasizing stronger evidence and dimming weaker evidence. The simulator receives unmodified validation images without attribution overlays. Thus, regional simulation evaluates descriptions derived from localized evidence, whereas regional coherence directly evaluates evidence crops.

Cross-seed stability. For independently seeded runs, we compute ARI between each pair of concept assignments on the same target test examples and average over all run pairs. This measures reproducibility without requiring ground-truth semantic groups. We interpret stability alongside Act because repeated collapse to a single group can also produce high agreement. The corresponding tables specify the seed counts and whether each reported metric is computed from an individual run or averaged across runs.

A.4 Constructed Image Testsets

A.4.1 Benchmark Details

We construct 12 dataset pairs $(\mathcal{D}_A, \mathcal{D}_B)$ spanning local attributes, object categories, visual renditions, and scene contexts. Each target dataset contains three annotated groups, and all methods use the same concept budget, $K = 3$. Figure 8 summarizes the target and reference groups and their sizes. Group annotations define the evaluation partition; concept learning uses only target-versus-reference membership.

CelebA-HQ scenarios. Seven scenarios use facial attribute annotations to define the target and reference sets. These comparisons cover accessories, hair color, facial hair, hairline, adornments, aging-related attributes, and attribute co-occurrence. Their construction tests several aspects of concept recovery: accessories occupy different image regions, hairline attributes share a common

#		scenario	Dataset A classes (K=3 each)	Dataset B classes	N_A / N_B
CelebA-HQ — faces, attribute predicates	1	celeba_accessories	eyeglasses (600), wearing_hat (600), necktie (600)	plain_face (1800)	1800 / 1800
	2	celeba_hair_color	blond_hair (600), brown_hair (600), gray_hair (600)	black_hair (600), brown_hair (600), bald (499)	1800 / 1699
	3	celeba_facial_hair	stubble (900), goatee (620), mustache (372)	clean_shaven (1892)	1892 / 1892
	4	celeba_hairline	bangs (900), receding_hairline (900), bald (439)	regular_hairline (2239)	2239 / 2239
	5	celeba_multiactive	eyeglasses_only (872), smiling_only (900), eyeglasses_smiling (596)	neither (2368)	2368 / 2368
	6	celeba_adornments	lipstick (900), earrings (656), necklace (395)	unadorned (1951)	1951 / 1951
	7	celeba_aging_signals	eye_bags (900), gray_hair (900), receding_hairline (900)	young_none (2700)	2700 / 2700
ImageNet / ImageNet-R / ImageNet*	8	imagenet_neckwear	bolo_tie (1300), bow_tie (1300), windsor_tie (1213)	cardigan (1300), jersey (1285), sweatshirt (931)	3813 / 3516
	9	imagenet_headwear	cowboy_hat (1300), mortarboard (1300), sombrero (1300)	suit (618), lab_coat (509), groom (457)	3900 / 1584
	10	imagenet_media	sketch (1500), tattoo (1004), toy (993)	natural (4850)	3497 / 4850
	11	imagenet_star_contexts	in_the_snow (483), in_the_rain (335), at_night (261)	base (500)	1079 / 500
	12	imagenet_emergency	ambulance (1300), fire_engine (1300), police_van (1300)	minivan (1300), pickup (1300), sports_car (1300)	3900 / 3900

Figure 8: Construction of the 12 image testsets. Each row specifies the three target groups, the reference groups, and their sample counts across all splits. The scenarios span local attributes, shared and co-occurring attributes, object categories, visual renditions, and scene contexts.

region, and eyeglasses and smiling can occur separately or together. The hair-color scenario deliberately includes brown hair in both datasets, introducing a shared semantic group alongside the distinguishing groups.

Attribute predicates specify group membership. For facial hair, both datasets are restricted to examples annotated as male; for hairline, hats are excluded from both datasets. In the aging-signals scenario, groups follow the priority gray hair, eye bags, then receding hairline, giving each target example one evaluation label while allowing attributes to co-occur.

ImageNet-family scenarios. The remaining five scenarios use ImageNet-1K, ImageNet-R, and ImageNet*. Neckwear and headwear compare related accessory categories, while emergency vehicles provide a comparison based on object categories. The rendition scenario contrasts sketches, tattoos, and toy depictions from ImageNet-R with natural photographs drawn from its object-class vocabulary. The context scenario compares snow, rain, and nighttime edits with base images. During construction, CLIP-based filtering retains context edits that exhibit the intended condition and reduces hat-containing examples in the headwear reference set.

Sampling and splits. We sample examples deterministically within each group, subject to availability and scenario-specific caps. Each target and reference group is split into approximately 70% training, 10% validation, and 20% test examples. All methods use the same split manifests. Group and dataset sizes can differ; the counts in Figure 8 include all three splits.

A.4.2 Additional Qualitative Results

Figure 9 presents additional concept discovery examples and their spatial evidence maps. Viewing the grouped images alongside their evidence maps connects the recovered groups to the visual regions supporting their concept scores, complementing the recovery and regional coherence metrics.

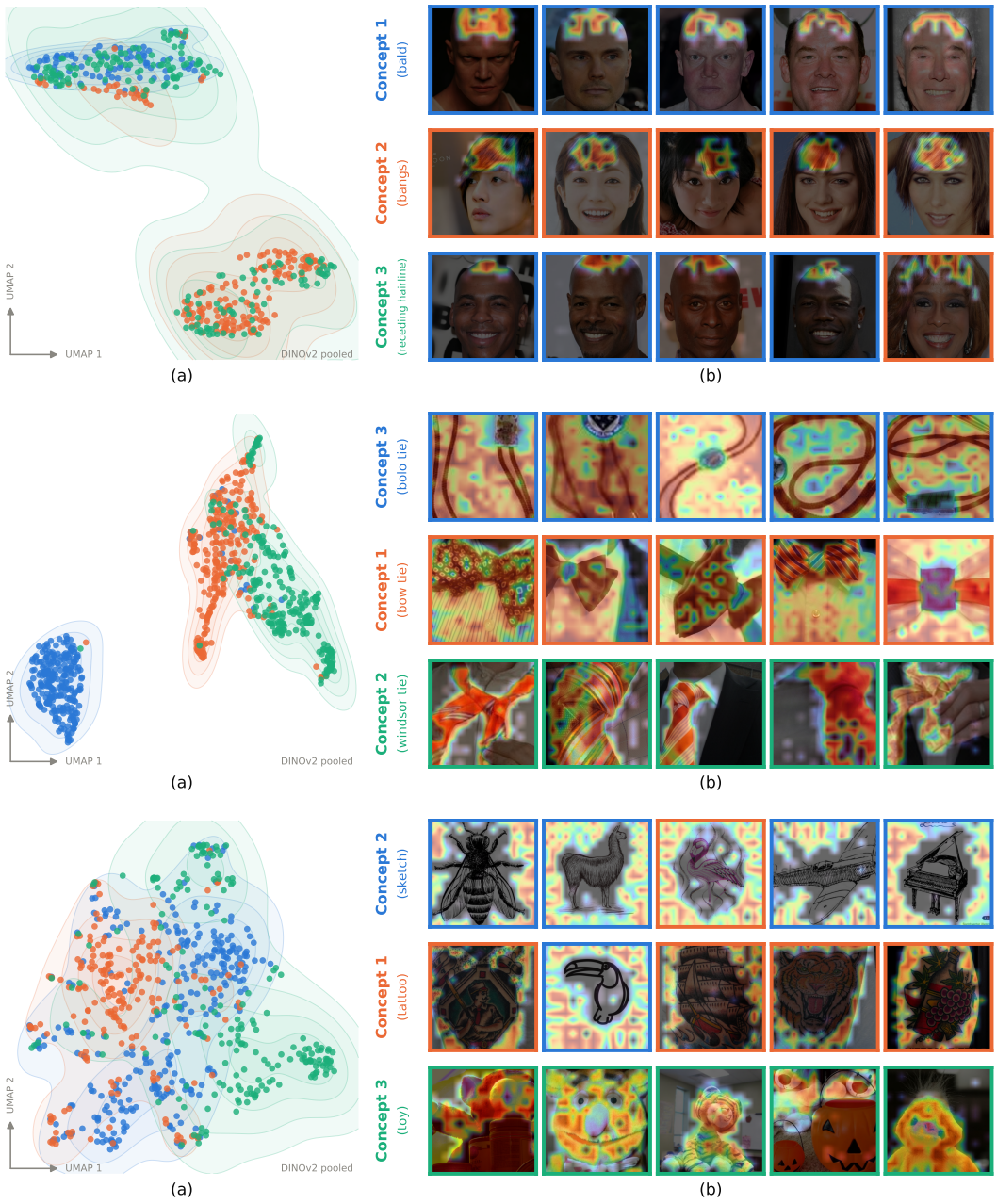

Figure 9: Additional qualitative results on the constructed image testsets. Example images illustrate the discovered groups, while concept-specific evidence maps highlight the regions supporting their scores.

Table 5: **EditLens: generated versus edited writing** ($K = 4$; 81 test pairs). Entries are mean $\pm$ sample SD over three seeds where shown. Act is the mean active-concept fraction; ρ uses seed 0; Stab is mean pairwise ARI across seeds. HypotheSAEs uses pooled representations.

	Discriminability	Coverage		Monosemanticity		Reliability
Method	cEER ↓	Act ↑	Cov ↑	ΔCoh ↑	ρ ↑	Stab ↑
k-means (frozen)	0.0726 ± 0.0009	1.00	0.309 ± 0.000	0.0899 ± 0.0000	0.651	1.000
k-means (classifier)	0.0368 ± 0.0048	1.00	0.979 ± 0.007	0.0686 ± 0.0198	0.562	0.496
HypotheSAEs (frozen)	0.3213 ± 0.1408	0.42	0.761 ± 0.413	0.0185 ± 0.0449	0.756	0.090
HypotheSAEs (classifier)	0.0358 ± 0.0077	0.58	1.000 ± 0.000	0.0130 ± 0.0162	0.468	0.175
TextDiff	0.3307 ± 0.0574	0.83	0.202 ± 0.124	0.1638 ± 0.0224	0.209	0.053
ConceptDiff (ours)	0.0223 ± 0.0069	1.00	1.000 ± 0.000	0.0928 ± 0.0046	0.660	0.957

A.5 Additional Text Experiments

We extend the main text study with a generated-versus-edited comparison on EditLens, three comparisons between generative models on StoryScope, and an analysis of concept evolution as the concept budget increases. We retain the text representation and evaluation protocol described in Sec. 4.3 and Appendix A.3.

A.5.1 EditLens: Generated versus Edited Writing

We compare Gemini-generated writing with Gemini-edited writing using 81 matched test pairs and $K = 4$. Because both datasets involve the same model, this comparison examines differences between generating and editing text, extending the generated-versus-human comparison in the main paper.

Table 5 shows that ConceptDiff achieves the lowest cEER while retaining all four concepts and high cross-seed stability. Classifier-based HypotheSAEs also separates the datasets effectively, but leaves several concepts inactive. Conversely, frozen HypotheSAEs achieves the highest simulation score and TextDiff the highest coherence, yet both distinguish the datasets less effectively. These results illustrate why describability and coherence must be assessed alongside discrimination and concept usage. With only 13 reference validation pairs, coverage is sensitive to threshold calibration; we therefore base the discrimination comparison primarily on cEER.

A.5.2 StoryScope: Differences between Generative Models

We compare Claude, Gemini, and GPT using responses to matched StoryScope prompts. Each model serves in turn as the target, with responses from the other two models forming the reference set. These are three one-versus-rest comparisons. We use $K = 6$ for Claude and $K = 4$ for Gemini and GPT.

Across all three comparisons, ConceptDiff maintains the full concept budget while achieving low cEER and near-complete target coverage (Table 6). Classifier-based baselines attain even lower cEER, but frequently use fewer concepts; their simulation scores are also consistently below ConceptDiff's. Thus, highly accurate source separation can coexist with a limited or poorly describable concept set. Frozen k-means remains more coherent and stable, while providing weaker discrimination and lower coverage. The comparison therefore highlights a trade-off between reproducible grouping in the pretrained representation and learning concepts that distinguish the generating models.

A.5.3 Concept Evolution with K

We examine how the discovered differences change with $K \in \{4, 6, 8, 12\}$ on the EditLens generated-versus-human comparison. Each setting is fitted independently, and Figure 10 tracks the assignments of the same held-out documents across settings.

Table 6: **StoryScope: comparisons between generative models.** Each target model is compared with the other two. Entries are mean ± sample SD over three seeds where shown. Act is the mean active-concept fraction; ρ uses seed 0; Stab is mean pairwise ARI across seeds. HypotheSAEs uses pooled representations. Displayed zeros are rounded.

	Discriminability	Coverage		Monosemanticity		Reliability
Method	cEER ↓	Act ↑	Cov ↑	ΔCoh ↑	ρ ↑	Stab ↑
Claude vs. GPT and Gemini ($K = 6$)						
k-means (frozen)	0.1150 ± 0.0041	1.00	0.514 ± 0.012	0.1467 ± 0.0009	0.439	0.606
k-means (classifier)	0.0000 ± 0.0001	0.83	1.000 ± 0.000	0.0367 ± 0.0077	0.023	0.263
HypotheSAEs (frozen)	0.0718 ± 0.0015	1.00	1.000 ± 0.000	0.0188 ± 0.0065	0.195	0.081
HypotheSAEs (classifier)	0.0001 ± 0.0001	0.22	1.000 ± 0.000	-0.0087 ± 0.0062	0.053	0.007
TextDiff	0.2939 ± 0.0801	0.56	0.051 ± 0.010	0.0474 ± 0.0056	0.298	0.051
ConceptDiff (ours)	0.0054 ± 0.0004	1.00	0.997 ± 0.001	0.0379 ± 0.0094	0.214	0.245
Gemini vs. Claude and GPT ($K = 4$)						
k-means (frozen)	0.0523 ± 0.0010	1.00	0.874 ± 0.002	0.1295 ± 0.0016	0.381	0.819
k-means (classifier)	0.0000 ± 0.0000	0.83	1.000 ± 0.000	0.0957 ± 0.0087	−0.013	0.214
HypotheSAEs (frozen)	0.0959 ± 0.0084	1.00	1.000 ± 0.000	0.0561 ± 0.0035	0.148	0.042
HypotheSAEs (classifier)	0.0000 ± 0.0000	0.33	1.000 ± 0.000	0.0694 ± 0.0090	−0.191	0.333
TextDiff	0.2954 ± 0.0586	0.92	0.081 ± 0.013	0.0936 ± 0.0056	0.343	0.102
ConceptDiff (ours)	0.0002 ± 0.0000	1.00	1.000 ± 0.000	0.0730 ± 0.0091	0.308	0.304
GPT vs. Claude and Gemini ($K = 4$)						
k-means (frozen)	0.1075 ± 0.0033	1.00	0.550 ± 0.017	0.1472 ± 0.0011	0.341	0.939
k-means (classifier)	0.0000 ± 0.0000	1.00	1.000 ± 0.000	0.0654 ± 0.0144	0.113	0.285
HypotheSAEs (frozen)	0.1040 ± 0.0065	1.00	1.000 ± 0.000	0.0442 ± 0.0094	0.460	0.010
HypotheSAEs (classifier)	0.0000 ± 0.0000	0.25	1.000 ± 0.000	0.0744 ± 0.0186	0.215	0.333
TextDiff	0.2904 ± 0.0214	0.92	0.107 ± 0.007	0.0641 ± 0.0098	0.393	0.000
ConceptDiff (ours)	0.0044 ± 0.0030	1.00	1.000 ± 0.000	0.0861 ± 0.0109	0.346	0.224

At $K = 4$, the concepts capture broad patterns associated with sensory fiction, explanatory prose, news and forecasts, and evaluative reviews. Larger budgets reveal more specific distinctions: review-related patterns separate into product-use and local-service reviews, while news-related patterns distinguish future-oriented announcements from incident reports. The additional concepts also capture patterns shared across broader groups, including asides and hedging and "testament" framing. Increasing K therefore supports both finer distinctions within prominent dataset differences and the isolation of recurring language patterns across those groups.

The transitions include both splitting and regrouping, as each column represents a separately learned concept set. The figure illustrates how the concept budget changes the granularity of the comparison.

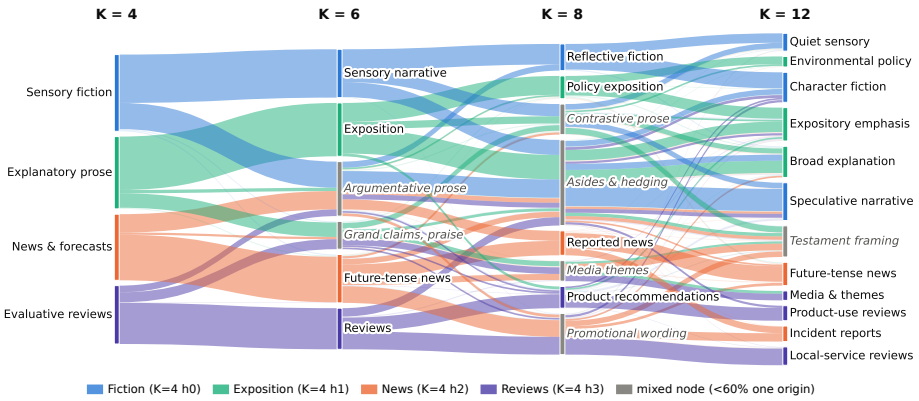

Figure 10: Concept evolution on EditLens as K increases from 4 to 12. Bands trace the assignments of the same held-out documents, with widths proportional to document counts. Colors indicate their $K = 4$ origins; gray nodes have no single origin contributing at least 60% of their members. Independently fitted concept sets reveal finer distinctions within broad groups and shared patterns across groups.