CCDiff: Inverse Canonical Correlation Analysis for Discovering Visual Differences in Natural Language

Neelesh Bisht* **Xingjian Li*†** **Zihan Li** **Bo Jiang**
Runmin Jiang **Mostofa Rafid Uddin** **Yang Liu** **Min Xu†**
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, United States
{xingjia2,mxu1}@cs.cmu.edu

Abstract

Set-level visual difference discovery is increasingly important for dataset auditing and for understanding model behavior under distribution shift, yet manually inspecting thousands of images is impractical. This motivates the emerging task of *set difference captioning*, i.e., describing concepts that are more often true for one image set than another. Different from existing approaches which heavily rely on large language models (LLMs), incurring substantial computational cost in terms of time and tokens, we introduce **CCDiff**, an efficient, auditable, statistically grounded, and training-free framework for set difference captioning that significantly reduces LLM dependence during core difference discovery, achieving over $2\times$ speedup and reducing token usage to zero. CCDiff operates in three stages: constructing a shared pool of candidate concepts from domain vocabularies or lightweight captioning models, filtering candidates to obtain set-specific concept pools, and performing *inverse* canonical correlation analysis (CCA) across sets to identify low-correlation directions that correspond to set differences. Candidate concepts are then ranked using a CCDiff score that balances inverse correlation with within-set representativeness. We evaluate CCDiff on VisDiffBench and further validate it on additional domains including MetaShift and MIMIC-CXR. Across benchmarks and real-world settings, CCDiff matches or exceeds prior LLM-based approaches while being substantially more efficient and scalable.

1 Introduction

The increasing deployment of machine learning models in real-world applications ranging from medical imaging [Erickson et al., 2017] and autonomous driving [Janai et al., 2021] to science [Jumper et al., 2021], has heightened concerns about their reliability under distribution shifts [Park et al., 2021, Cascarano et al., 2023, Stocco and Tonella, 2021]. When models encounter data that differs from their training distribution, performance can degrade significantly, posing serious risks in safety-critical domains [Koh et al., 2021, Perone et al., 2018, Alcorn et al., 2019, Wang et al., 2022b]. These risks are further amplified in resource-constrained environments, such as low-infrastructure regions, where access to large-scale compute and proprietary AI tools is limited. In such settings, researchers and engineers often need to reason about differences between entire datasets rather than individual samples, for example, comparing training and deployment data to diagnose distribution shifts [Eyuboglu et al., 2022], auditing datasets to uncover spurious correlations [Ye et al., 2025], or analyzing failures in deployed systems [Jain et al., 2022]. Similar situation arise in applied domains as well; for instance, a doctor may compare sets of medical images to understand how diseased

*Equal contribution.
†Corresponding authors: Xingjian Li and Min Xu.

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

cases differ from healthy ones [Hu et al., 2023]. Supporting these workflows requires dataset-level explanations that are not only interpretable, but also computationally efficient and scalable under strict hardware and budget constraints.

Motivated by these real-world scenarios, a new line of research has emerged that aims to explain distribution shifts in datasets from a data-centric perspective in natural language. The problem is formulated as *Set Difference Captioning*: given two image sets $\mathcal{D}_A$ and $\mathcal{D}_B$, the goal is to discover natural language descriptions that are more often true in $\mathcal{D}_A$ than in $\mathcal{D}_B$. While captioning a single image is usually feasible, a main challenge here is to accurately capture the key differences between image sets. State-of-the-art solutions such as VisDiff [Dunlap et al., 2024] and GSCLIP [Zhu et al., 2022] heavily rely on large language models (LLMs) to generate interpretable semantic descriptions from individual captions. When processing large datasets, these solutions become computationally expensive and impractical in resource-constrained environments.

To address these limitations, we propose CCDiff, an efficient and auditable framework for set difference captioning. CCDiff adapts classical CCA to improve the discovery and ranking modules. We refer to this adaptation as *inverse-canonical correlation analysis (Inverse-CCA)*, which identifies distinctive concepts by examining the least-correlated directions between CLIP [Radford et al., 2021] embeddings of images and candidate textual concepts. Specifically, we first apply cross-set CCA between images from one set and candidate text embeddings from the other, and then focus on canonical components with near-zero correlation, which correspond to textual directions that are poorly explained by the opposing image set. Based on general correlation statistics, our core algorithm is agnostic to the source of candidate concepts [3] and data domains. Crucially, it significantly reduces reliance on large language models during core difference discovery; it operates solely on CLIP embeddings and canonical correlation analysis, making it computationally efficient, scalable, and well-suited for resource-constrained environments.

On the VisDiffBench benchmark [Dunlap et al., 2024], our method surpasses VisDiff on Acc@1 across all difficulty levels while significantly reducing LLM dependence during core difference discovery. We further validate our method on MetaShift and medical imaging dataset MIMIC-CXR [Johnson et al., 2019a], where CCDiff significantly outperforms VisDiff across all difficulty levels, demonstrating that it can capture subtle, domain-specific differences and is practical for use in critical areas such as healthcare. Beyond these benchmarks, CCDiff also uncovers meaningful differences in open-ended real-world settings (Section 4.6). For example, it recovers the human presence failure mode of ResNet reported by VisDiff and additionally surfaces a previously unreported one, scene clutter, which is associated with a 7.4% accuracy drop for ResNet-50. It further reveals that StableDiffusionV2 produces more saturated images with more white borders than StableDiffusionV1, and characterizes memorable images by close-up, attention-grabbing foregrounds rather than low-saliency, background-dominant scenes. By providing an accurate, efficient, and lightweight alternative, CCDiff makes set-difference captioning accessible in resource-constrained and specialized domains, enabling new applications that were previously impractical due to computational constraints.

Our main contributions can be summarized as follows.

1. We introduce CCDiff, a lightweight framework that discovers dataset-level visual differences directly from CLIP embeddings, significantly reducing LLM dependence during core difference discovery.

2. We propose a novel inverse use of classical correlation analysis to solve the problem of multimodal difference discovery.

3. CCDiff achieves superior or competitive accuracy on set difference captioning benchmarks in both general vision and medical domains across difficulty levels.

2 Related Works

Here we focus our discussion on two closely related topics, which are Set Difference Captioning and Canonical Correlation Analysis. A broader review of related directions is discussed in Appendix A.

[3] They can be drawn from existing domain vocabularies, curated word lists, or a lightweight image-captioning model.

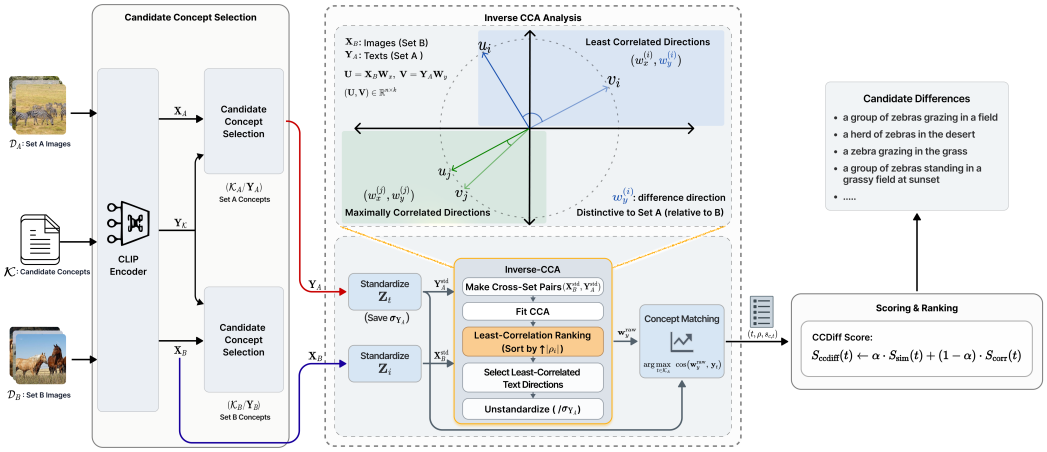

Figure 1: **Overview of the CCDiff framework.** Given two image sets, CCDiff discovers set-level semantic differences using inverse CCA with significantly reduced LLM dependence. This figure illustrates a single directional comparison (e.g., concepts distinctive to set A relative to set B). The same procedure can be applied symmetrically by swapping the two sets to find concepts distinctive to the other set.

2.1 Set Difference Captioning

Different from the traditional task of image difference captioning [Yao et al., 2022, Jhamtani and Berg-Kirkpatrick, 2018, Park et al., 2019b], the emerging research of *set difference captioning* focuses on dataset-level visual difference discovery and explanation by leveraging large vision language models. This task is critical in numerous real-world applications, including dataset auditing (e.g., comparing ImageNet with ImageNetV2), model debugging (identifying systematic failure modes), generative model analysis (contrasting different versions of diffusion models), out-of-distribution detection, cognitive studies (e.g., memorable vs. forgettable images), and data-centric AI workflows (guiding dataset curation). Methods such as VisDiff [Dunlap et al., 2024], GSCLIP [Zhu et al., 2022] leverage generative capabilities together with vision–language representations to discover interpretable semantic descriptions that characterize how in-distribution and shifted datasets differ. However, their reliance on repeated inference with LLMs leads to substantial computational overhead at the ranking stage. Another recent work proposes goal-conditioned discovery of distribution differences between text corpora through natural language predicates [Zhong et al., 2023]. Similarly, much earlier work on difference captioning [Alayrac et al., 2022, Li et al., 2023a, 2025a, Yao et al., 2022] and change captioning [Chang and Ghamisi, 2023, Kim et al., 2021, Park et al., 2019a] aims to describe visual differences between a single pair of images using natural language. While effective for pairwise comparison, these approaches are not designed to characterize differences across large image collections and thus do not scale to dataset-level analysis.

3 Our Method: CCDiff

Given two image sets $\mathcal{D}_A$ and $\mathcal{D}_B$, the goal of set difference captioning is to generate natural language descriptions that are more true for $\mathcal{D}_A$ than $\mathcal{D}_B$ (and vice-versa). While prior work approaches this problem with a proposer-ranker framework, we take a simpler, direct, and statistically grounded approach. We introduce **CCDiff**[4], a method based on **Inverse Canonical Correlation Analysis (inverse-CCA)** that identifies distinctive textual concepts directly from multimodal embeddings, as illustrated in Figure 1 (see Appendix A.1 for details on CCA). Instead of generating and ranking difference candidates with large models, CCDiff analyzes the correlation structure between image and text embeddings using inverse-CCA in a shared representation space to surface concepts that characterize one image set and are either poorly explained or absent in the other.

[4]Code is available at `https://github.com/neeleshbisht99/CCDiff`.

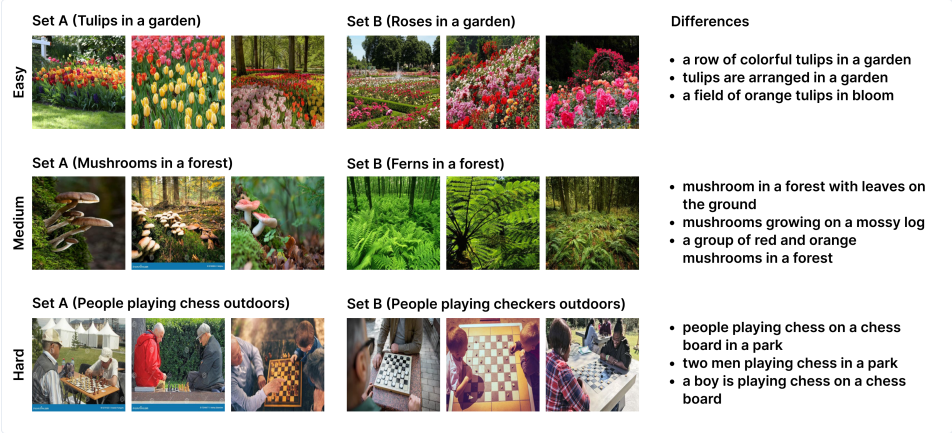

Figure 2: **Examples of CCDiff across difficulty levels.** Representative image-set pairs and the discovered textual differences for easy (top), medium (middle), and hard (bottom) comparisons. Easy cases differ by salient object categories (e.g., tulips vs. roses), medium cases require contextual discrimination within the same scene type (e.g., mushrooms vs. ferns in forests), and hard cases involve fine-grained semantic distinctions under similar visual contexts (e.g., chess vs. checkers played outdoors).

3.1 Overview

CCDiff operates in three core stages:

1. Candidate Concept Pool Creation: We construct a shared pool of candidate textual concepts $\mathcal{K}$ that is agnostic to its source: it can be generated from any captioning model, collected from existing domain vocabularies, or curated from word lists.

2. Candidate Concept Selection: For each image set, we select the most relevant concepts by computing their average semantic similarity to the images in CLIP embedding space and retaining the top-k concepts. This yields set-specific concept pools that are representative of the underlying image distributions.

3. Inverse CCA Analysis & Ranking: We apply CCA between image embeddings from one set and candidate concept embeddings from the *other* set. We identify the *least correlated* canonical components ("anti-correlated" directions) and map them back to candidate concepts to reveal or discover distinctive features. These candidates are then scored and ranked using a metric that combines inverse correlation with within-set similarity.

Together, these stages enable CCDiff to surface concise, interpretable descriptions of how two image sets differ at the dataset level, as illustrated in Figure 2.

3.2 Algorithm

The complete CCDiff algorithm is summarized in Algorithm 1 in Appendix A.1. For each image set $\mathcal{D}_s$ ($s \in \{A, B\}$), we denote the filtered set of candidate concepts as $\mathcal{K}_s$, the corresponding CLIP text embeddings as $\mathbf{Y}_s$ and the relevance score for the j-th concept as $s_{s,j}$, computed during candidate concept selection. For inverse CCA, we consider ordered pairs $c, i \in \{(A, B), (B, A)\}$, where c denotes the set providing candidate concepts and i denotes the set providing images, used for inverse correlation analysis. For each selected concept $t \in \mathcal{K}_c$, we denote its relevance score from candidate concept selection as $s_{c,t}$. We denote by ρ the canonical correlation value associated with each CCA component that produces t. The final ranked lists of distinctive concepts are denoted by $\mathcal{L}_A$ and $\mathcal{L}_B$, corresponding to concepts that are more characteristic of $\mathcal{D}_A$ than $\mathcal{D}_B$, and vice-versa. The method operates in five sequential stages: (1) embedding images and the candidate concept pool via CLIP, (2) candidate concept selection to obtain set-specific candidate sets, (3) standardizing the embeddings, (4) performing inverse-CCA analysis to identify distinctive concepts, and (5) scoring and ranking the resulting candidates. Each stage is detailed in the following sections.

3.3 Candidate Concept Generation

We construct a shared pool of *candidate concepts* $\mathcal{K}$ that captures visual attributes likely to appear across the image sets under comparison. In our implementation, this pool is generated by applying a vision language model (e.g., BLIP-2, used by default to remain consistent with baselines) to produce captions for images from both sets; any off-the-shelf image captioning model or existing domain vocabulary, or curated word list can be used. The resulting captions are treated as candidate concepts and aggregated into a unified 'concept pool'.

Importantly, once embedded via CLIP, the entire difference discovery pipeline operates in the embedding space without further LLM API calls. This contrasts with existing methods such as VisDiff [Dunlap et al., 2024], which require repeated LLM inference for both proposal and ranking. In practical deployment, candidate concepts would typically come from existing domain-specific vocabularies, making CCDiff suitable for applications in resource-constrained real-world settings. This candidate concept pool serves as a lightweight source of visual descriptions from which set-specific concepts are later selected.

3.4 Candidate Concept Selection

To identify the most relevant candidate concepts for each image set, average semantic similarity is used. Given an image set $\mathcal{D}_s$ with N_s images and a candidate concept pool $\mathcal{K}$, we first compute the cosine similarity between each normalized image embedding $\tilde{\mathbf{x}}_i$ and each normalized concept embedding $\tilde{\mathbf{y}}_j$. This yields a similarity matrix $\mathbf{S}_s \in \mathbb{R}^{N_s \times |\mathcal{K}|}$ where each entry $\mathbf{S}_s[i,j] = \cos(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_j)$.

We then compute the mean similarity for each concept across all images in the set:

$$\mathbf{a}_s[j] = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{S}_s[i,j].$$

The resulting vector $\mathbf{a}_s \in \mathbb{R}^{|\mathcal{K}|}$ is the relevance score for each candidate concept. We then select the top-k concepts ranked by $\mathbf{a}_s$ to form set-specific candidate pools $\mathcal{K}_A$ and $\mathcal{K}_B$, with corresponding relevance score vectors $\mathbf{s}_A$ and $\mathbf{s}_B$ for the selected concepts (i.e., $\mathbf{s}_s = \mathbf{a}_s[I_s]$, where I_s contains indices of the selected concepts).

3.5 Inverse-CCA Analysis

The core of CCDiff is an **Inverse Canonical Correlation Analysis (Inverse-CCA)** method that identifies concepts that are distinctive to one image set relative to the other. Given the filtered candidate concept sets $\mathcal{K}_A$ and $\mathcal{K}_B$ and their corresponding CLIP embeddings, we perform Inverse-CCA to analyze cross-set correlations and extract directions that capture set-specific differences. Importantly, individual candidate concepts need not explicitly encode relative differences, inverse-CCA identifies distinguishing directions at the distribution level by analyzing cross-set correlations across the full candidate pool. Here, *inverse* refers to selecting canonical components in increasing order of absolute correlation rather than selecting the most-correlated components, while *back-projection* refers to mapping a selected canonical text direction from the standardized CCA solution into the raw CLIP embedding space.

Cross-Set CCA Analysis For each image set $s \in \{A, B\}$, we apply feature-wise standardization to both image embeddings $\mathbf{X}_s \in \mathbb{R}^{N_s \times d}$ and text embeddings $\mathbf{Y}_s \in \mathbb{R}^{|\mathcal{K}_s| \times d}$, yielding $\mathbf{X}_s^{\text{std}}$ and $\mathbf{Y}_s^{\text{std}}$. Given these standardized embeddings, to identify concepts distinctive to $\mathcal{D}_A$, we analyze the mismatch between images from $\mathcal{D}_B$ and candidate concepts from $\mathcal{D}_A$. We pair standardized image embeddings $\mathbf{X}_B^{\text{std}}$ with standardized text embeddings $\mathbf{Y}_A^{\text{std}}$ using modulo-based sampling, forming paired data $(\mathbf{X}_B^{\text{paired}}, \mathbf{Y}_A^{\text{paired}})$. We then apply CCA to find linear projections $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize correlation between the paired views:

$$\rho = \text{corr}(\mathbf{X}_B^{\text{paired}} \mathbf{w}_x, \mathbf{Y}_A^{\text{paired}} \mathbf{w}_y).$$

CCA yields m component pairs $(\mathbf{w}_x^{(i)}, \mathbf{w}_y^{(i)})$ with corresponding correlations ρ_i, where $m = \min(n_{\text{components}}, d, n-1)$ and n is the number of image-text pairs (which equals the number of

Table 1: Results on VisDiffBench. CCDiff (with BLIP-2 captions and CLIP features) surpasses the best VisDiff pipelines (GPT-4V image-based and BLIP-2 + GPT-4 caption-based proposers with CLIP rankers) on Acc@1 across all difficulty levels, while significantly outperforming other proposer–ranker combinations and avoiding the use of GPT-4 during the core proposal and difference discovery stages. The three CCDiff variants using Linear (LK), Polynomial (PK), and Gaussian (GK) kernels show consistently high accuracy across all difficulty levels.

Method	Proposer	Ranker	PIS-Easy		PIS-Medium		PIS-Hard	
			Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
VisDiff	Feature (BLIP-2)	Feature (CLIP)	0.48	0.69	0.13	0.33	0.12	0.23
VisDiff	Image (LLaVA-1.5)	Feature (CLIP)	0.71	0.81	0.39	0.49	0.28	0.43
VisDiff	Caption (BLIP-2 + GPT-4)	Caption (Vicuna-1.5)	0.60	0.92	0.49	0.77	0.31	0.61
VisDiff	Caption (BLIP-2 + GPT-4)	Image (LLaVA-1.5)	0.78	0.99	0.58	0.80	0.38	0.62
VisDiff	Image (GPT-4V)	Feature (CLIP)	0.95	**1.00**	0.75	**0.87**	0.57	0.74
VisDiff	Caption (BLIP-2 + GPT-4)	Feature (CLIP)	0.88	0.99	0.75	0.86	0.61	**0.80**
CCDiff	Caption (BLIP-2 + CLIP)	Inverse-CCA (LK)	**0.96**	0.99	**0.76**	0.86	**0.70**	0.79
CCDiff	Caption (BLIP-2 + CLIP)	Inverse-CCA (PK)	0.89	0.98	0.68	0.82	0.60	0.75
CCDiff	Caption (BLIP-2 + CLIP)	Inverse-CCA (GK)	0.94	0.99	0.74	0.80	0.62	0.73

images in the set providing the images, i.e., $n = N_i$ for image set $\mathcal{D}_i$). Capping the number of components at m keeps the analysis well-posed when CCA reduces dimensionality. The **inverse-CCA** insight is that components with **low or near-zero correlation** correspond to directions in $\mathcal{D}_A$'s text space that are poorly explained by images from the opposing set $\mathcal{D}_B$, and thus capture set-specific attributes. We therefore sort components by the absolute correlation $|\rho_i|$ in ascending order. For each low correlation component i, we take the corresponding text-space direction $\mathbf{w}_y^{(i)}$ and map it back to the raw embedding space via: $\mathbf{w}_y^{\text{raw}} = \mathbf{w}_y^{(i)} / \boldsymbol{\sigma}_{\mathbf{Y}_A}$, where $\boldsymbol{\sigma}_{\mathbf{Y}_A}$ denotes the per-dimension standard deviation used during $\mathbf{Y}_A$'s standardization. This element-wise division undoes the feature-wise standardization for the selected text direction. We then ℓ_2-normalize the back-projected direction, and its scale does not affect matching because cosine similarity is invariant to positive rescaling. After normalization we compute cosine similarity between $\mathbf{w}_y^{\text{raw}}$ and all original text embeddings in $\mathbf{Y}_A$, selecting the nearest concept t_i as a candidate distinctive description for $\mathcal{D}_A$. We perform the same analysis in the opposite direction to identify concepts distinctive to $\mathcal{D}_B$, by pairing $\mathbf{X}_A^{\text{std}}$ and $\mathbf{Y}_B^{\text{std}}$ and repeating the procedure.

3.6 Scoring & Ranking Distinctive Concepts

Each candidate concept t receives a unified **CCDiff Score** that balances two factors:

1. Inverse Correlation Score: $S_{\text{corr}}(t) = 1 - |\rho_i|$, where ρ_i is the CCA correlation of the component that produced t. Higher values indicate lower correlation with the opposing set.

2. Within-Set Similarity Score: $S_{\text{sim}}(t) = \frac{s_{c,t} - s_c^{\min}}{s_c^{\max} - s_c^{\min}}$, where $s_{c,t}$ is the relevance score for concept t in set c (from Section 3.4), and $s_c^{\min}$ and $s_c^{\max}$ are the minimum and maximum relevance scores.

The final score is as follows:

$$S_{\text{ccdiff}}(t) = \alpha \cdot S_{\text{sim}}(t) + (1 - \alpha) \cdot S_{\text{corr}}(t)$$

where α is a weighting parameter determined empirically. This specific formulation prioritizes concepts that are both **distinctive** (low correlation with the opposing set) and **representative** (well-aligned within their own set). Without the relevance or similarity component, the method may select concepts that are distinctive but rare or poorly aligned with their own set; without the inverse correlation component, it may select concepts that are representative but not distinctive. By combining both factors, CCDiff identifies concepts that effectively differentiate between the two sets while remaining semantically meaningful within their source set. Appendix E.1 provides a controlled synthetic example showing that inverse correlation can rank a stronger cross-set difference above a salient but weaker foreground difference.

Finally, candidates are ranked by $S_{\text{ccdiff}}(t)$ in descending order. This yields two ranked lists: concepts distinctive to $\mathcal{D}_A$ relative to $\mathcal{D}_B$, and concepts distinctive to $\mathcal{D}_B$ relative to $\mathcal{D}_A$.

4 Experiments

4.1 Experimental Settings

Datasets We evaluated our method primarily on the VisDiffBench [Dunlap et al., 2024] benchmark, which consists of 187 paired image sets, each annotated with a ground-truth natural language difference. The core of the benchmark is the **PairedImageSets** collection, which consists of 150 carefully curated image-set pairs designed to cover a wide-spectrum of real-world differences. These pairs are categorized into three difficulty levels (50 pairs each): **Easy** (apparent differences, e.g., "dogs playing in a park" vs. "cats playing in a park"), **Medium** (fine-grained differences, e.g., "SUVs on the road" vs. "Sedans on the road"), and **Hard** (subtle differences, e.g., "People practicing yoga in a mountainous setting" vs. "People meditating in a mountainous setting"). Figure 2 provides qualitative examples from the Easy, Medium, and Hard splits of VisDiffBench.

Implementation Details For image and text embeddings, we use CLIP with the "ViT-bigG-14" backbone. The candidate concept pool is constructed using BLIP-2 [Li et al., 2023b] by default (though the framework is model-agnostic): for each pair $(\mathcal{D}_A, \mathcal{D}_B)$, we sample 20 images per set across three rounds (60 per set, 120 total), generating one caption per image with the prompt *"Describe this image in detail,"* yielding up to 120 candidate concepts per pair. For candidate concept selection (Section 3.4), we select the top-$k = 20$ concepts by average similarity. The final **CCDiff Score** uses a weighting parameter $\alpha = 0.3$, empirically determined to balance distinctiveness and representativeness. Table 16 in Appendix E shows a broad performance plateau across the tested settings spanning $\alpha = 0.3$ to 1.0, and we use $\alpha = 0.3$ unchanged across VisDiffBench, MetaShift, MIMIC-CXR, and BREEDS without per-dataset tuning. All experiments were conducted on a single NVIDIA GeForce RTX 3090 (24GB).

Evaluation To evaluate the performance in VisDiffBench, we follow the VisDiff [Dunlap et al., 2024] evaluation methodology. Given a ranked list of different descriptions produced by CCDiff for a pair $(\mathcal{D}_A, \mathcal{D}_B)$, we use a GPT-4-based evaluator to compute the semantic similarity between each predicted description and the ground-truth difference. The evaluator assigns a score of 0 (no match), 0.5 (partial match), or 1 (perfect match). This automated metric has been shown to align well with human judgment, exhibiting a Pearson correlation of 0.80 with human annotators [Dunlap et al., 2024], consistent with prior findings that large language models can reliably approximate human evaluation [Dubois et al., 2024, Zheng et al., 2023].

4.2 Results on VisDiffBench Dataset

As shown in Table 1, the linear-kernel variant of CCDiff achieves 96% Acc@1 and 99% Acc@5 on the Easy set, 76% Acc@1 and 86% Acc@5 on the Medium set, and 70% Acc@1 and 79% Acc@5 on the Hard set, surpassing the best VisDiff configuration (Image (GPT-4V) with CLIP ranker) on Acc@1 across all difficulty levels while achieving comparable Acc@5. Figure 2 provides qualitative illustrations of these results across difficulty levels.

Notably, CCDiff achieves this performance without relying on LLM inference during core difference discovery, unlike prior LLM-based approaches. The candidate concepts can come from any source, though in our experiments we use BLIP-2 captions, they could equally come from existing vocabularies, or curated word lists (as discussed in section 3.3). We further analyze the sensitivity of CCDiff to the choice of captioning model in Appendix D, showing that while BLIP-2 achieves the best performance, lighter captioning models also yield reasonable results. More importantly, once these concepts are embedded via CLIP, CCDiff performs all subsequent operations (candidate concept selection, inverse-CCA, and scoring-ranking) entirely in the embedding space, without any LLM API calls. While VisDiff requires multiple LLM inferences for proposing and discovering candidate differences, CCDiff operates entirely in CLIP embedding space, making it more efficient and scalable for real-world applications.

Additional qualitative examples across all difficulty levels are provided in Appendix B.5. We further compare CCDiff against naive VLM baselines that directly prompt vision-language models without a structured pipeline. These approaches consistently underperform compared to structured pipelines such as CCDiff across all difficulty levels (see Appendix C). Additional ablation studies, including CCA kernel choice, ranker ablation, and modulo-pairing stability, are provided in Appendix E.

Table 2: Computational efficiency comparison between VisDiff and CCDiff on VisDiffBench, including runtime, LLM token usage, and GPU memory usage.

Method	Runtime (s)	Prompt Tokens	Completion Tokens	Peak GPU Memory (GB)	Total Peak Memory (GB)
VisDiff	108.2	3114.1	255.4	**13.0**	**22.0**
CCDiff	**48.6**	**0**	**0**	13.5	22.5

Table 3: Evaluation on MetaShift (1,379 paired sets, 100 images per set).

Method	Acc@1	Acc@5
VisDiff	0.24	0.46
CCDiff	**0.32**	0.46

To further assess the stability and reliability of the reported results, we provide confidence intervals, paired significance tests, and additional evaluator-reliability details in Appendix B.3. We also analyze candidate-pool coverage, oracle performance, and the remaining ranking gap in Appendix D.1.

Diagnostic Baselines To isolate the contribution of inverse-CCA, we compare against CLIP-only, mean-similarity-difference, and centroid-difference rankers using the same candidate pool and CLIP features. CCDiff matches or outperforms all three rankers across every split, with the largest gain on Hard, where it achieves 0.70 Acc@1 compared to 0.34, 0.56, and 0.56, respectively (Table 14 in Appendix E). Under the same controlled setup, inverse-CCA also outperforms most-correlated CCA, PLS, and PCA variants on all splits, achieving a Hard Acc@1 of 0.70 compared with 0.19, 0.53, and 0.50, respectively, as shown in Table 15 in Appendix E.

4.3 Computational Efficiency

CCDiff is substantially more computationally efficient than VisDiff-style pipelines while surpassing it on Acc@1 across all difficulty levels (Tables 1, 3 and 4). As shown in Table 2, CCDiff achieves more than $2\times$ speedup (48.6s vs. 108.2s) by avoiding iterative LLM-based proposals during core difference discovery, and incurs zero LLM token usage compared to 3,369 tokens per run for VisDiff (roughly 3,000 prompt and 300 completion tokens), amounting to nearly 0.5 million tokens across 150 set pairs. Beyond speed, reducing reliance on API-based LLM calls during core difference discovery also addresses practical concerns around cost and data privacy, making CCDiff particularly suitable for resource-constrained or privacy-sensitive deployments. Both methods have comparable GPU memory usage, with only minor overhead from CCA computation. Storage requirements are identical when using BLIP-2; domain-specific knowledge banks are assumed pre-existing, adding no additional overhead. Furthermore, CCDiff is more interpretable, as it operates entirely via explicit embedding-space operations, enabling fully reproducible, local, and offline analysis without reliance on closed LLM or VLM APIs. This efficiency advantage becomes increasingly pronounced as the number of image-set pairs grows, as VisDiff's cost scales with per-pair LLM inference while CCDiff does not. A stage-level breakdown separating shared candidate generation and external evaluation from core difference discovery and ranking is provided in Appendix B.4.

4.4 Evaluation on MetaShift

To further evaluate generalization beyond VisDiffBench, we conduct experiments on MetaShift [Liang and Zou, 2022], a large-scale and diverse collection of 1,379 paired image sets constructed from Visual Genome, representing a $7\times$–$8\times$ increase in scale over VisDiffBench. As shown in Table 3, CCDiff outperforms VisDiff on Acc@1 by approximately 33% relatively (0.32 vs. 0.24) while achieving equal Acc@5 (0.46), despite the increased scale and noise. The lower absolute performance of both methods reflects the weaker semantic separation and higher noise inherent to MetaShift [Dunlap et al., 2024], making the task inherently more challenging than VisDiffBench, yet CCDiff's relative gain demonstrates that it generalizes to larger and less curated settings. Additional dataset preparation and implementation details, together with further qualitative examples, are provided in Appendix F.

Table 4: Acc@1 performance comparison on MIMIC-CXR paired sets across difficulty levels. Both methods use the same domain-adapted CXR-CLIP and fine-tuned BLIP models.

Method	Easy	Medium	Hard
VisDiff	0.59	0.55	0.39
CCDiff	**0.73**	**0.68**	**0.42**

4.5 Medical Imaging Evaluation on MIMIC-CXR

To validate CCDiff's applicability in real-world critical domains, we evaluate on MIMIC-CXR [Johnson et al., 2019b], a chest X-ray dataset requiring precise, interpretable difference explanations under computational constraints. From a publicly available subset of approximately 30,000 images, we construct 30 paired sets (10 easy, 10 medium, 10 hard) across 24 clinically meaningful radiological categories. As shown in Table 4, CCDiff outperforms VisDiff across all difficulty splits, achieving Acc@1 scores of 73%, 68%, and 42% on Easy, Medium, and Hard, respectively, compared to 59%, 55%, and 39% for VisDiff. CCDiff's stronger performance can be attributed to its robustness to noisy or imperfect captions, which are common in specialized domains. Unlike VisDiff, which relies purely on language understanding and is thus sensitive to caption quality, CCDiff incorporates image–text cross-set correlations during both discovery and ranking, ensuring that extracted differences reflect the underlying image distributions rather than caption artifacts alone. Full dataset-construction details and additional qualitative examples are provided in Appendix G.

4.6 Real-World Applications

To demonstrate that CCDiff generalizes beyond benchmarks, we reproduce three representative application scenarios from VisDiff [Dunlap et al., 2024]: analyzing model failure modes (§4.6.1), characterizing generative model differences (§4.6.2), and understanding image memorability (§4.6.3).

4.6.1 Finding Failure Modes of ResNet

Following VisDiff [Dunlap et al., 2024], we contrast ImageNet images misclassified by *both* ResNet-50 and ResNet-101 [He et al., 2016] ($\mathcal{D}_A$) against correctly classified images ($\mathcal{D}_B$). CCDiff recovers the *human presence* signal reported by VisDiff, and additionally surfaces a previously unreported failure mode: *scene clutter*. Quantitative validation confirms that both factors correlate with consistent accuracy drops across ResNet-50, ResNet-101, and their ensemble, with the clutter effect larger than the human-presence effect (7.4% vs. 2.9% for ResNet-50). Full details are provided in Appendix H.

4.6.2 Comparing Versions of Stable Diffusion

Following VisDiff [Dunlap et al., 2024], we compare StableDiffusionV1 (SDv1) and StableDiffusionV2 (SDv2), proposing differences on 1,634 PartiPrompts [Yu et al., 2022] images and validating on 10,000 unseen DiffusionDB [Wang et al., 2023] images. CCDiff's top hypotheses: "strong color contrast", "vibrant colors", "framed artworks", and "single centered subjects", are consistent with VisDiff's findings, while additionally surfacing "single centered subjects" as a novel structural difference not reported by VisDiff. Quantitative validation confirms that SDv2 produces higher saturation and more white borders than SDv1, and these differences generalize to unseen prompts. Full details are provided in Appendix I.

4.6.3 Describing Memorability in Images

Following VisDiff [Dunlap et al., 2024], we apply CCDiff to the LaMem dataset [Khosla et al., 2015], dividing images into the most memorable 25th percentile ($\mathcal{D}_A$) and the least memorable 25th percentile ($\mathcal{D}_B$). CCDiff recovers the broad categories reported by VisDiff (human presence, close-up views, humorous settings) while providing finer-grained characterization: memorable images are associated with close-up, attention-grabbing foregrounds, while forgettable images are dominated by low-saliency, background-dominant scenes. Using CLIP to classify each LaMem test image against the top-25 CCDiff-derived descriptions, CCDiff achieves a classification accuracy of 87.06%, compared to 89.8% reported by VisDiff [Dunlap et al., 2024]. Full details are provided in Appendix J.

4.7 Evaluation on BREEDS

To further evaluate CCDiff, we construct 50 paired image sets from sibling ImageNet superclasses in BREEDS [Santurkar et al., 2021], with 100 images per set. As shown in Table 24, CCDiff achieves 0.73 Acc@1 and 0.92 Acc@5, performing on par with VisDiff (0.76 and 0.91, respectively), while avoiding LLM inference during core difference discovery, with full dataset construction and evaluation details provided in Appendix M.

4.8 Evaluation on Abstract Visual Differences

Beyond semantic differences, CCDiff can also identify abstract texture differences. We further evaluate CCDiff on DTD [Cimpoi et al., 2014], constructing 50 paired sets of distinct texture categories with 100 images per set and using a texture vocabulary as the candidate concept pool. As shown in Table 25, CCDiff identifies the distinctive texture with high accuracy, reaching up to 0.89 Acc@1 and 0.99 Acc@5. Full experimental details, qualitative examples, and the limitation for low-level image distortions are provided in Appendix N.

5 Limitations

CCDiff's effectiveness is inherently constrained by the coverage and quality of its candidate concept pool. Because differences are identified only within this fixed set of textual concepts, the method cannot account for semantic distinctions absent from the concept pool. This design trades expressive flexibility for efficiency, enabling scalability with significantly reduced LLM dependence, while bounding the space of discoverable differences. In practice, achieving accurate and nuanced differences therefore requires careful selection of the concept source, such as domain-specific lexicons or lightweight captioning models.

6 Conclusion

This paper presents CCDiff, a lightweight framework for set-difference captioning that leverages inverse canonical correlation analysis to explain differences between two image sets. CCDiff addresses the key limitation of prior frameworks that rely on large language models to identify and propose dataset-level differences by operating directly on CLIP embeddings, significantly reducing LLM dependence during core difference discovery. Extensive experiments across multiple datasets demonstrate that CCDiff is both robust and competitive. In benchmarks such as VisDiffBench, MetaShift, and MIMIC-CXR, CCDiff matches or exceeds prior LLM-dependent methods while being substantially more efficient and scalable.

Acknowledgments and Disclosure of Funding

This work was supported in part by U.S. NIH grant R35GM158094. The authors declare no competing interests.

References

Shotaro Akaho. A kernel method for canonical correlation analysis, 2007. URL https://arxiv.org/abs/cs/0609071.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

Michael A. Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects, 2019. URL https://arxiv.org/abs/1811.11553.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/andrew13.html.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, 4:IV–876, 2003. URL https://api.semanticscholar.org/CorpusID:7691428.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pages 213–229. Springer International Publishing, 2020.

Anna Cascarano, Jordi Mur-Petit, Jerónimo Hernández-González, Marina Camacho, Nina de Toro Eadie, Polyxeni Gkontra, Marc Chadeau-Hyam, Jordi Vitrià, and Karim Lekadir. Machine and deep learning for longitudinal biomedical data: a review of methods and applications. *Artif. Intell. Rev.*, 56(Suppl 2):1711–1771, August 2023. ISSN 0269-2821. doi: 10.1007/s10462-023-10561-w. URL https://doi.org/10.1007/s10462-023-10561-w.

Shizhen Chang and Pedram Ghamisi. Changes to captions: An attentive network for remote sensing change captioning. *IEEE Transactions on Image Processing*, 32:6047–6060, 2023. ISSN 1941-0042. doi: 10.1109/tip.2023.3328224. URL http://dx.doi.org/10.1109/TIP.2023.3328224.

Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Automated data slicing for model validation:a big data - ai integration approach, 2019. URL https://arxiv.org/abs/1807.06068.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. URL https://openaccess.thecvf.com/content_cvpr_2014/html/Cimpoi_Describing_Textures_in_2014_CVPR_paper.html.

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, 2021. URL https://arxiv.org/abs/2107.00758.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024. URL https://arxiv.org/abs/2305.14387.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. Machine learning for medical imaging. *Radiographics*, 37(2):505–515, March 2017.

Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. Zero-shot out-of-distribution detection based on the pre-trained model clip. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6568–6576, June 2022. ISSN 2159-5399. doi: 10.1609/aaai.v36i6.20610. URL http://dx.doi.org/10.1609/aaai.v36i6.20610.

Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared A. Dunnmon, James Y. Zou, and Christopher R'e. Domino: Discovering systematic errors with cross-modal embeddings. *ArXiv*, abs/2203.14960, 2022. URL https://api.semanticscholar.org/CorpusID:247779010.

Kenji Fukumizu, Francis R. Bach, and Arthur Gretton. Statistical consistency of kernel canonical correlation analysis. *J. Mach. Learn. Res.*, 8:361–383, May 2007. ISSN 1532-4435.

Gene H. Golub and Hongyuan Zha. The canonical correlations of matrix pairs and their numerical computation. In Adam Bojanczyk and George Cybenko, editors, *Linear Algebra for Signal Processing*, pages 27–49, New York, NY, 1995. Springer New York. ISBN 978-1-4612-4228-4.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Christiaan Heij and {B (Berend)} Roorda. A modified canonical correlation approach to approximate state space modelling. In *Proceedings of the 30-th Conference on Decision and Control, IEEE*, pages 1343–1348, 1991.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2018. URL https://arxiv.org/abs/1610.02136.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features, 2022. URL https://arxiv.org/abs/2201.11114.

HAROLD HOTELLING. Relations between two sets of variates*. *Biometrika*, 28(3-4):321–377, 12 1936. ISSN 0006-3444. doi: 10.1093/biomet/28.3-4.321. URL https://doi.org/10.1093/biomet/28.3-4.321.

Erdong Hu, Longteng Guo, Tongtian Yue, Zijia Zhao, Shuning Xue, and Jing Liu. Onediff: A generalist model for image difference captioning, 2025. URL https://arxiv.org/abs/2407.05645.

Xinyue Hu, Lin Gu, Qiyuan An, Mengliang Zhang, Liangchen Liu, Kazuma Kobayashi, Tatsuya Harada, Ronald M. Summers, and Yingying Zhu. Expert knowledge-aware image difference graph representation learning for difference-aware medical visual question answering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4156–4165. ACM, August 2023. doi: 10.1145/3580305.3599819. URL http://dx.doi.org/10.1145/3580305.3599819.

Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space, 2022. URL https://arxiv.org/abs/2206.14754.

Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state of the art, 2021. URL https://arxiv.org/abs/1704.05519.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images, 2018. URL https://arxiv.org/abs/1808.10584.

Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Roger G. Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6(1):317, Dec 2019a. ISSN 2052-4463. doi: 10.1038/s41597-019-0322-0. URL https://doi.org/10.1038/s41597-019-0322-0.

Alistair E. W. Johnson, Tom J. Pollard, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih ying Deng, Yifan Peng, Zhiyong Lu, Roger G. Mark, Seth J. Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs, 2019b. URL https://arxiv.org/abs/1901.07042.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon Kohl, Andrew Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 07 2021. doi: 10.1038/s41586-021-03819-2.

Aditya Khosla, Akhil S. Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *International Conference on Computer Vision (ICCV)*, 2015.

Hoeseong Kim, Jongseok Kim, Hyungseok Lee, Hyunsung Park, and Gunhee Kim. Viewpoint-agnostic change captioning with cycle consistency. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2075–2084. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00210. URL https://doi.org/10.1109/ICCV48922.2021.00210.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts, 2021. URL https://arxiv.org/abs/2012.07421.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning, 2023a. URL https://arxiv.org/abs/2306.05425.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Joshua Adrian Cahyono, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning, 2025a. URL https://arxiv.org/abs/2305.03726.

Jinglun Li, Kaixun Jiang, Zhaoyu Chen, Bo Lin, Yao Tang, Weifeng Ge, and Wenqiang Zhang. Synthesizing near-boundary ood samples for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4496–4506, October 2025b.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023b. URL https://arxiv.org/abs/2301.12597.

Xingjian Li, Qiming Zhao, Neelesh Bisht, Mostofa Rafid Uddin, Jin Yu Kim, Bryan Zhang, and Min Xu. Diffcam: Data-driven saliency maps by capturing feature differences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10327–10337, June 2025c.

Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts, 2022. URL https://arxiv.org/abs/2202.06523.

Kevin Z. Lin and Nancy R. Zhang. Quantifying common and distinct information in single-cell multimodal data with tilted canonical correlation analysis. *Proceedings of the National Academy of Sciences*, 120(32):e2303647120, 2023. doi: 10.1073/pnas.2303647120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2303647120.

Hal Lindsey, J. T. Webster, and Sharon K. Halpern. Canonical correlation as a discriminant tool in a periodontal problem. *Biometrical Journal*, 27:257–264, 1985. URL https://api.semanticscholar.org/CorpusID:120008659.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, *Artificial Neural Networks — ICANN 2001*, pages 353–360, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44668-2.

Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations, 2022. URL https://arxiv.org/abs/2211.13445.

Jun Nie, Yonggang Zhang, Zhen Fang, Tongliang Liu, Bo Han, and Xinmei Tian. Out-of-distribution detection with negative prompts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nanyAujl6e.

Chunjong Park, Anas Awadalla, Tadayoshi Kohno, and Shwetak Patel. Reliable and trustworthy machine learning for health using dataset shift detection, 2021. URL https://arxiv.org/abs/2110.14019.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019a.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning, 2019b. URL https://arxiv.org/abs/1901.02527.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale, 2023. URL https://arxiv.org/abs/2303.14186.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world, 2023. URL https://arxiv.org/abs/2306.14824.

Christian Samuel Perone, Pedro Lemos Ballester, Rodrigo C. Barros, and Julien Cohen-Adad. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194:1–11, 2018. URL https://api.semanticscholar.org/CorpusID:53432181.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL https://arxiv.org/abs/1602.04938.

Juho Rousu, Daniel D Agranoff, Olugbemiro Sodeinde, John Shawe-Taylor, and Delmiro Fernandez-Reyes. Biomarker discovery by sparse canonical correlation analysis of complex clinical phenotypes of tuberculosis and malaria. *PLoS Comput. Biol.*, 9(4):e1003018, April 2013.

Dylan Sam, Devin Willmott, Joao D. Semedo, and J. Zico Kolter. Finetuning clip to reason about pairwise differences, 2025. URL https://arxiv.org/abs/2409.09721.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. BREEDS: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mQPBmvyAuk.

B K Sarkar and Chiranjib Chakraborty. DNA pattern recognition using canonical correlation algorithm. *J. Biosci.*, 40(4):709–719, October 2015.

Stephan V. Schell and William A. Gardner. Programmable canonical correlation analysis: a flexible framework for blind adaptive spatial filtering. *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 647–652 vol.1, 1993. URL https://api.semanticscholar.org/CorpusID:29687139.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

Jose A Seoane, Colin Campbell, Ian N M Day, Juan P Casas, and Tom R Gaunt. Canonical correlation analysis for gene-based pleiotropy discovery. *PLoS Comput. Biol.*, 10(10):e1003876, October 2014.

Jiahang Sha, Jingxuan Bao, Kefei Liu, Shu Yang, Zixuan Wen, Junhao Wen, Yuhan Cui, Boning Tong, Jason H Moore, Andrew J Saykin, Christos Davatzikos, Qi Long, Li Shen, and Alzheimer's Disease Neuroimaging Initiative. Preference matrix guided sparse canonical correlation analysis for mining brain imaging genetic associations in alzheimer's disease. *Methods*, 218:27–38, October 2023.

Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. Modeldiff: A framework for comparing learning algorithms, 2022. URL https://arxiv.org/abs/2211.12491.

Li Shen and Paul M Thompson. Brain imaging genomics: Integrated analysis and machine learning. *Proc. IEEE Inst. Electr. Electron. Eng.*, 108(1):125–162, January 2020.

Andrea Stocco and Paolo Tonella. Confidence-driven weighted retraining for predicting safety-critical failures in autonomous driving systems. *Journal of Software (Malden, Ma)*, 34, 2021. URL https://api.semanticscholar.org/CorpusID:241784606.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language, 2022a. URL https://arxiv.org/abs/2205.14100.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization, 2022b. URL https://arxiv.org/abs/2103.03097.

Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 893–911, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.51. URL https://aclanthology.org/2023.acl-long.51/.

K. W. Wong, P. C. W. Fung, and C. C. Lau. Study of the mathematical approximations made in the basis-correlation method and those made in the canonical-transformation method for an interacting bose gas. *Phys. Rev. A*, 22:1272–1286, Sep 1980. doi: 10.1103/PhysRevA.22.1272. URL https://link.aps.org/doi/10.1103/PhysRevA.22.1272.

Linli Yao, Weiying Wang, and Qin Jin. Image difference captioning with pre-training and contrastive learning, 2022. URL https://arxiv.org/abs/2202.04298.

Wenqian Ye, Luyang Jiang, Eric Xie, Guangtao Zheng, Yunsheng Ma, Xu Cao, Dongliang Guo, Daiqing Qi, Zeyu He, Yijun Tian, Megan Coffee, Zhe Zeng, Sheng Li, Ting-hao, Huang, Ziran Wang, James M. Rehg, Henry Kautz, and Aidong Zhang. The clever hans mirage: A comprehensive survey on spurious correlations in machine learning, 2025. URL https://arxiv.org/abs/2402.12715.

Kihyun You, Jawook Gu, Jiyeon Ham, Beomhee Park, Jiho Kim, Eun K. Hong, Woonhyuk Baek, and Byungseok Roh. *CXR-CLIP: Toward Large Scale Chest X-ray Language-Image Pre-training*, page 101–111. Springer Nature Switzerland, 2023. ISBN 9783031438950. doi: 10.1007/978-3-031-43895-0_10. URL http://dx.doi.org/10.1007/978-3-031-43895-0_10.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=AFDcYJKhND. Featured Certification.

Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S. Yu. From canonical correlation analysis to self-supervised graph neural networks, 2021. URL https://arxiv.org/abs/2106.12484.

Shizhen Zhao, Jiahui Liu, Xin Wen, Haoru Tan, and Xiaojuan Qi. Equipping vision foundation model with mixture of experts for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1751–1761, October 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. Goal driven discovery of distributional differences via language descriptions, 2023. URL https://arxiv.org/abs/2302.14233.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015. URL https://arxiv.org/abs/1512.04150.

Zhuoping Zhou, Davoud Ataee Tarzanagh, Bojian Hou, Boning Tong, Jia Xu, Yanbo Feng, Qi Long, and Li Shen. Fair canonical correlation analysis, 2023. URL https://arxiv.org/abs/2309.15809.

Zhiying Zhu, Weixin Liang, and James Zou. Gsclip : A framework for explaining distribution shifts in natural language, 2022. URL https://arxiv.org/abs/2206.15007.

A Additional Related Work

A.1 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a classical multivariate statistical technique [HOTELLING, 1936] that identifies pairs of linear projections of two multivariate variable sets such that the projected variables are maximally correlated. For two random variable $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^m$, their covariance matrix is $\Sigma_{XY} = Cov(X, Y)$. CCA aims at seeking two vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$ such that the correlation

$$\rho = corr(a^\top X, b^\top Y) = \frac{a^\top \Sigma_{XY} b}{\sqrt{a^\top \Sigma_{XX} a}\sqrt{b^\top \Sigma_{YY} b}}$$

is maximized. Formally the objective is:

$$\max_{a,b} a^\top \Sigma_{XY} b, s.t. a^\top \Sigma_{XX} a = b^\top \Sigma_{YY} b = 1.$$

For multi-dimensional projections, CCA finds a sequence of orthogonal projection pairs that are maximally correlated and uncorrelated with previous pairs [Golub and Zha, 1995, Zhang et al., 2021]. Kernel canonical correlation analysis (Kernel CCA) introduced by [Akaho, 2007, Melzer et al., 2001, Bach and Jordan, 2003] extends CCA to nonlinear-relationships by mapping the data into reproducing kernel Hilbert spaces. Given two random variable X and Y, kernel CCA aims to find maximally correlated nonlinear projections, restricted to reproducing kernel Hilbert spaces with corresponding kernels. Both CCA and KCCA are techniques for learning representation that are maximally predictive across two views of the data [Fukumizu et al., 2007, Andrew et al., 2013].

CCA and its variants have been widely applied across diverse scientific domains, including medicine [Lin and Zhang, 2023, Lindsey et al., 1985], physics [Wong et al., 1980], time-series modeling [Heij and Roorda, 1991], and signal processing [Schell and Gardner, 1993], among others [Zhou et al., 2023]. Beyond these areas, CCA has also seen extensive use in neuroscience, machine learning, and bioinformatics [Shen and Thompson, 2020, Sha et al., 2023], with numerous applications in computational biology [Seoane et al., 2014, Rousu et al., 2013, Sarkar and Chakraborty, 2015]. This breadth of applications highlights the versatility of CCA as a general-purpose tool for extracting relationships between multivariate variables, making it a foundational technique in scientific research [Zhou et al., 2023].

A.2 Out-of-distribution Detection

A core challenge in reliable machine learning is the detection of distribution shifts, as models often fail silently on out-of-distribution (OOD) data, raising critical safety concerns in deployment [Hendrycks and Gimpel, 2018]. Recent work on OOD detection for foundation models has pursued diverse strategies, particularly leveraging vision–language models such as CLIP. One line of work augments the CLIP model with a text generation module to produce candidate unseen labels, computing an OOD score based on the relative softmax mass assigned to generated versus known class label embeddings [Esmaeilpour et al., 2022]. Another approach learns class-specific negative CLIP prompts, classifying inputs as OOD when their image embeddings exhibit higher similarity to negative prompts than to positive class prompts [Nie et al., 2024]. Another complementary zero-shot approach leverages the inherent alignment between vision and language features in CLIP to compute OOD Score via Maximum Concept Matching (MCM) [Ming et al., 2022]. In parallel some of the previous work focuses on deriving OOD detection scores directly from model confidence outputs, such as maximum softmax probability [Hendrycks and Gimpel, 2018], synthesizing near-boundary OOD samples to explicitly refine the in- and out-of-distribution decision boundary [Li et al., 2025b], or adapting foundation models to better separate in- and out-of-distribution data [Zhao et al., 2025]. While effective at detecting the presence of a distribution shift, these approaches are fundamentally limited in explaining the nature of the shift in a way that is actionable or understandable to users. Our work differs with this line of research by describing the shifts or differences with natural languages.

A.3 Foundation Models for Difference Captioning

A parallel line of work focuses on solving the difference captioning task through training specialized foundation models on paired image data, following a supervised or contrastive learning paradigm.

Algorithm 1 CCDiff: Discover Visual Differences via Inverse CCA

Require: Image Sets $\mathcal{D}_A, \mathcal{D}_B$; Concept Bank $\mathcal{K}$;
Ensure: Ranked distinctive concepts $\mathcal{L}_A, \mathcal{L}_B$
Embed Images and Concept Bank
$\mathbf{X}_A, \mathbf{X}_B \leftarrow \text{CLIP}_{\text{image}}(\mathcal{D}_A), \text{CLIP}_{\text{image}}(\mathcal{D}_B)$
$\mathbf{Y}_\mathcal{K} \leftarrow \text{CLIP}_{\text{text}}(\mathcal{K})$
Concept Selection via Average Similarity
for $s \in \{A, B\}$ **do** ▷ Process each image set $\mathcal{D}_s$
  $\mathbf{S} \leftarrow \tilde{\mathbf{X}}_s \tilde{\mathbf{Y}}_\mathcal{K}^\top$ ▷ Cosine similarity matrix
  $\mathbf{a}_s \leftarrow \text{mean}(\mathbf{S}, \text{axis} = 0)$ ▷ Avg. similarity per concept
  Select top-k indices by $\mathbf{a}_s \rightarrow I_s$
  $\mathcal{K}_s \leftarrow \{\mathcal{K}[i] : i \in I_s\}$
  $\mathbf{Y}_s \leftarrow \mathbf{Y}_\mathcal{K}[I_s]$
  $\mathbf{s}_s \leftarrow \mathbf{a}_s[I_s]$ ▷ Relevance scores
end for
Standardize Image and Concept Embeddings
$\mathbf{X}_s^{\text{std}} \leftarrow \text{Standardize}(\mathbf{X}_s),\ \forall s \in \{A, B\}$
$\mathbf{Y}_s^{\text{std}} \leftarrow \text{Standardize}(\mathbf{Y}_s),\ \forall s \in \{A, B\}$
Inverse CCA Analysis
for each ordered pair $(c, i) \in \{(A, B), (B, A)\}$ **do**
  Pair images from $\mathbf{X}_i^{\text{std}}$ with $\mathbf{Y}_c^{\text{std}}$ (modulo pairing)
  Fit CCA and rank components by ascending $|\rho_i|$
  for each low-correlation component **do**
    Map text direction to closest concept t in $\mathcal{K}_c$
    Store candidate: $(t, \rho, s_{c,t})$ ▷ $s_{c,t}$ is relevance score for concept t in set c
  end for
end for
Score and Rank Concepts
for each candidate tuple $(t, \rho, s_{c,t})$ with set c **do**
  $S_{\text{corr}}(t) \leftarrow 1 - |\rho|$
  $S_{\text{sim}}(t) \leftarrow (s_{c,t} - s_c^{\min})/(s_c^{\max} - s_c^{\min})$
  $S_{\text{ccdiff}}(t) \leftarrow \alpha \cdot S_{\text{sim}}(t) + (1 - \alpha) \cdot S_{\text{corr}}(t)$
end for
Sort candidates by $S_{\text{ccdiff}}(t)$ descending $\rightarrow \mathcal{L}_A, \mathcal{L}_B$
return $\mathcal{L}_A, \mathcal{L}_B$

For example, [Yao et al., 2022] proposes a self-supervised pre-training framework that leverages tasks such as masked language modeling and fine-grained difference alignment, supported by cross-task data augmentation. However, these approaches remain constrained by a heavy dependence on domain-specific pre-training and multi-stage optimization, resulting in significant computational overhead and limited adaptability to new domains and resource-constrained settings. Similarly, OneDiff [Hu et al., 2025] proposes a generalist model. While achieving strong performance, it relies on a two-stage training pipeline that requires large-scale supervised data and substantial GPU resource, making it impractical for lightweight settings. Another related approach [Sam et al., 2025] proposes fine-tuning CLIP to understand differences between images, but this method also required additional training with synthetic data, adding further computational cost.

Notably, all of these approaches focus on modeling differences between individual image pairs through training-intensive pipelines, whereas CCDiff instead targets set-level difference discovery directly from frozen multimodal embeddings without additional model training.

A.4 Explainable AI

The work of set difference captioning is also relevant to the field of explanaible AI [Dunlap et al., 2024]. The evolution of explainable AI has recently expanded from local feature attribution, which explains individual predictions by highlighting influential input features (e,g, LIME [Ribeiro et al., 2016], CAM [Zhou et al., 2015], Grad-CAM [Selvaraju et al., 2019], Diff-CAM [Li et al., 2025c], SHAP [Lundberg and Lee, 2017], MILAN [Hernandez et al., 2022] etc.) towards data-centric and dataset-level explanations. This emerging direction seeks to understand the model behaviour through the structure and composition of the data itself, for e.g. by identifying influential training

Table 5: 95% bootstrap confidence intervals for CCDiff over the 50 image-set pairs in each VisDiff-Bench split.

Split	Acc@1	95% CI	Acc@5	95% CI
Easy	0.96	$[0.92, 0.99]$	0.99	$[0.97, 1.00]$
Medium	0.76	$[0.68, 0.84]$	0.86	$[0.80, 0.92]$
Hard	0.70	$[0.62, 0.78]$	0.79	$[0.70, 0.87]$

samples that that exert a significant influence on model outputs [Park et al., 2023, Shah et al., 2022], or discovering interpretable data subgroups and distributional difference [Chung et al., 2019, d'Eon et al., 2021, Eyuboglu et al., 2022, Dunlap et al., 2024]. Our set difference captioning formulation aligns naturally with these dataset-level explainability objectives. In this context, CCDiff introduces a purely data-driven, model-agnostic approach to explainability: instead of interpreting why a model makes a certain decision, it directly characterizes the distributional difference between two image sets in natural language, providing a data-centric explanation for differential model performance. By moving from pixel-level saliency to set-level semantic descriptions, CCDiff enables high level debugging and insight into dataset shifts, effectively bridging dataset understanding and explainable-AI without relying on model internal or instance-specific attributions.

A.5 Relation to Multimodal Retrieval

More broadly, unlike traditional multimodal retrieval, which ranks candidates against a single image query, set difference captioning requires identifying descriptions that reflect distributional differences across entire image sets. While CCDiff ranks textual candidates from a pool, similar in form to retrieval, its objective is fundamentally different: it seeks concepts that are discriminative at the set level, rather than semantically aligned to any individual image.

B Additional Details on VisDiffBench Evaluation

B.1 Dataset Details

Each set contains 100 images, retrieved using paired sentences as search queries. While VisDiffBench also includes a small number of paired sets from ImageNet-R and ImageNet*, which contain AI-generated or edited images, we restrict our quantitative evaluation to the PairedImageSets subset to ensure reliable benchmarking, as these pairs are reliable human-annotated collections of real images.

B.2 Deduplication

Since multiple text-space directions from low-correlation CCA components can map to the same textual concept, we remove duplicate candidates in the final ranked lists after scoring. For each unique text description, we retain only the instance with the highest $S_{\text{ccdiff}}(t)$ score, ensuring that each distinctive concept appears at most once in the final output.

B.3 Statistical Analysis and Evaluator Reliability

Confidence Intervals To quantify uncertainty in the reported accuracies, we compute 95% confidence intervals for CCDiff using bootstrap resampling over the 50 image-set pairs in each VisDiffBench split. As shown in Table 5, the intervals remain concentrated around the reported Acc@1 and Acc@5 values, indicating that the estimates are stable under resampling. On Hard, the marginal CCDiff Acc@1 interval of $[0.62, 0.78]$ lies above the VisDiff point estimate of 0.61, although the paired comparison below is not statistically significant.

Paired Significance Tests Since CCDiff and VisDiff are evaluated on the same image-set pairs, we apply a paired bootstrap to their per-pair differences. Table 6 compares CCDiff with the VisDiff configuration using the *Caption (BLIP-2 + GPT-4)* proposer and *Feature (CLIP)* ranker. For Acc@1, CCDiff significantly outperforms VisDiff Overall ($\Delta = +0.06$, $p = 0.024$), and the gain is also significant on Easy ($\Delta = +0.08$, $p = 0.005$). The Medium and Hard Acc@1 differences are positive but not statistically significant, with $p = 0.894$ and $p = 0.152$, respectively, reflecting the limited

Table 6: Paired-bootstrap comparison of CCDiff and the VisDiff caption-based configuration on VisDiffBench. Δ denotes CCDiff minus VisDiff.

Metric	Split	CCDiff	VisDiff	Δ	95% CI	p
Acc@1	Easy	0.96	0.88	$+0.08$	$[+0.03, +0.14]$	0.005
Acc@1	Medium	0.76	0.75	$+0.01$	$[-0.07, +0.10]$	0.894
Acc@1	Hard	0.70	0.61	$+0.09$	$[-0.03, +0.21]$	0.152
Acc@1	Overall	0.81	0.75	$+0.06$	$[+0.01, +0.11]$	0.024
Acc@5	Easy	0.99	0.99	$+0.00$	$[-0.03, +0.03]$	1.000
Acc@5	Medium	0.86	0.86	$+0.00$	$[-0.08, +0.08]$	1.000
Acc@5	Hard	0.79	0.80	-0.01	$[-0.09, +0.09]$	1.000
Acc@5	Overall	0.88	0.88	$+0.00$	$[-0.04, +0.04]$	1.000

sample size of 50 pairs per split. In particular, the paired Hard interval of $[-0.03, +0.21]$ contains zero. Acc@5 remains at parity across the splits because both methods almost always recover the correct difference within the top-5, leaving little headroom on this metric.

The per-split estimates should be interpreted in light of the 50-pair sample size. We additionally evaluate at larger scale on MetaShift with 1,379 pairs and on BREEDS with 50 pairs, where CCDiff remains competitive with VisDiff across the reported metrics.

Evaluator Reliability We follow the exact GPT-4-based evaluation protocol used by VisDiff [Dunlap et al., 2024]. The evaluator assigns a raw score of 0, 1, or 2, corresponding to no match, a partial match, or a perfect match, which is normalized to 0, 0.5, or 1 for reporting. This protocol has been validated against human annotators with a Pearson correlation of $r = 0.80$ [Dunlap et al., 2024], consistent with prior findings that LLM-based evaluators can approximate human judgment [Dubois et al., 2024, Zheng et al., 2023]. We use the same protocol for CCDiff and VisDiff, supporting direct comparability and reproducibility, and rely on this existing human validation rather than conducting a new human study.

B.4 Stage-Level Computational Breakdown

Table 7 separates the shared candidate-generation and external evaluation stages from the core difference-discovery and ranking pipeline. Both methods use BLIP-2 for candidate generation and the same external GPT-4 judge for benchmark evaluation. In the core pipeline, VisDiff makes 3 GPT-4 proposer calls and uses approximately 3,369 tokens before a separate CLIP ranking pass. CCDiff makes 0 LLM calls and uses 0 LLM tokens because inverse-CCA replaces the GPT-4 proposer and CLIP reranker in a single pass. Consequently, CCDiff incurs no GPT-4 API cost during core difference discovery and ranking. Thus, all of VisDiff's LLM cost lies in the difference-discovery and ranking pipeline, while candidate generation is shared. The external GPT-4 evaluation is identical for both methods and is excluded from the call and token comparison. Including the shared stages and external evaluation, total runtime is 108.2 seconds for VisDiff and 48.6 seconds for CCDiff. Peak and total GPU memory are comparable at 13.0/22.0 GB for VisDiff and 13.5/22.5 GB for CCDiff, respectively, as reported in Table 2.

B.5 Additional Qualitative Results on VisDiffBench

Figure 3 presents additional qualitative examples from VisDiffBench across Easy, Medium, and Hard difficulty levels, comparing differences discovered by VisDiff and CCDiff.

C Comparison with Naive VLM Baselines

To evaluate whether directly prompting vision-language models can serve as an alternative, we prompt LLaVA-1.5 and GPT-4o to generate the top-5 differences between two image sets in a single pass, without any intermediate proposal or ranking stage. The generated outputs are evaluated using the same GPT-4-based evaluator as CCDiff and VisDiff.

As shown in Table 8, while GPT-4o improves over LLaVA-1.5, both approaches remain consistently below structured pipelines such as VisDiff and CCDiff (Table 1), particularly on harder splits. This

Table 7: Stage-level computational breakdown for VisDiff and CCDiff on VisDiffBench. LLM calls and tokens are counted through ranking. The shared external GPT-4 judge is excluded from call and token accounting and shown with dashes.

Stage	Method	Component	LLM Calls	LLM Tokens	Runtime (s)
Load + setup	Both	Config + data	–	–	1.2
Candidate generation	Both	BLIP-2	0	0	31.3
Difference discovery	VisDiff	GPT-4 proposer	3	∼3,369	41.9
	CCDiff	inverse-CCA	0	0	13.4
Ranking	VisDiff	CLIP	0	0	31.1
	CCDiff	inverse-CCA (same pass)	0	0	0.0
Evaluation (GPT-4 judge)	Both	GPT-4	–	–	2.7
Total	VisDiff	–	3	∼3,369	108.2
	CCDiff	–	0	0	48.6

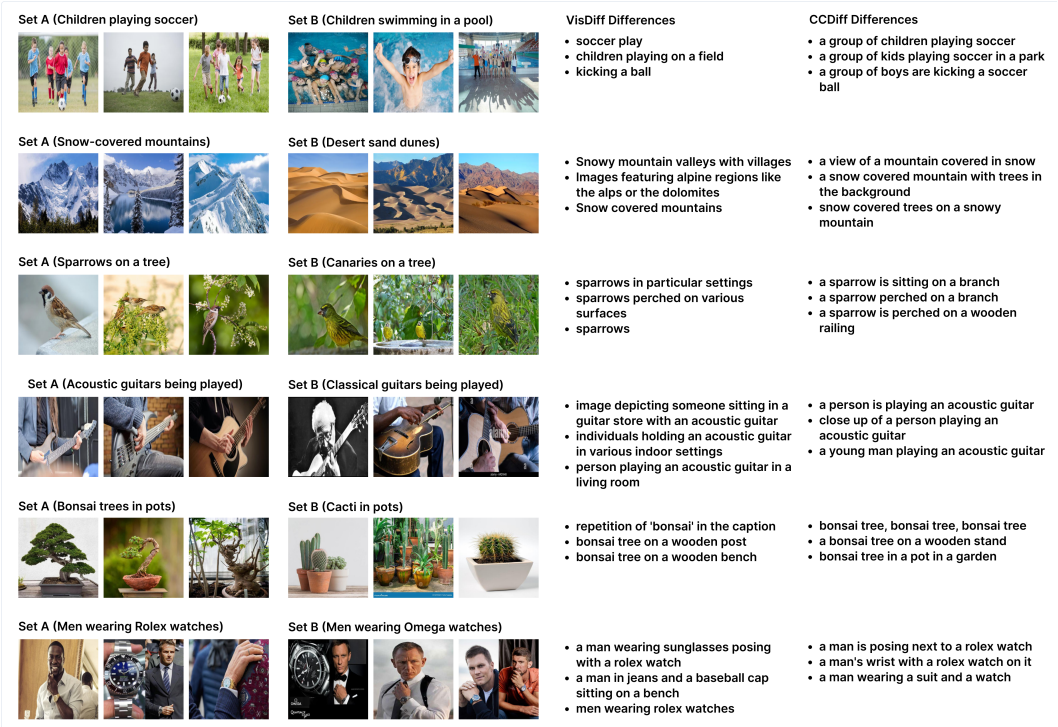

Figure 3: Additional qualitative examples from VisDiffBench across difficulty levels (Easy, Medium, Hard), comparing differences discovered by VisDiff and CCDiff for representative image-set pairs.

highlights that directly prompting VLMs does not reliably capture the most discriminative set-level distinctions, as they lack explicit mechanisms to contrast distributions across image sets.

D Sensitivity to Candidate Concept Quality

Since CCDiff operates on a pool of candidate textual concepts $\mathcal{K}$ that is agnostic to how these concepts are obtained, a natural question is how the choice of captioning model affects difference discovery performance.

To study this effect, we evaluate CCDiff using candidate concepts generated by captioning models of varying capacity. As shown in Table 9, BLIP-2 achieves the strongest performance across all difficulty levels, with a clear margin across all splits. However, despite the large disparity in model size between BLIP-2 and lighter captioning models such as Kosmos [Peng et al., 2023] and GIT [Wang et al.,

Table 8: Comparison with naive VLM baselines on VisDiffBench. Models are prompted to directly generate the top-5 differences without any proposal or ranking stage, and evaluated using the same GPT-4-based evaluator as CCDiff and VisDiff.

Model	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
LLaVA-1.5	0.44	0.60	0.24	0.44	0.31	0.49
GPT-4o	0.83	0.98	0.64	0.86	0.46	0.71

Table 9: Performance of CCDiff using different image captioning models for candidate concept generation on VisDiffBench. All methods use CLIP embeddings and the same inverse-CCA ranking pipeline.

Captioner	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
BLIP-2	**0.96**	**0.99**	**0.76**	**0.86**	**0.70**	**0.79**
Kosmos	0.90	0.95	0.59	0.71	0.52	0.67
GIT	0.87	0.96	0.68	0.75	0.48	0.62

2022a], they still achieve reasonable performance. This result suggests that difference discovery is relatively robust to the choice of captioning model, and does not strictly require large or proprietary language models. Even smaller captioning models are able to surface sufficiently informative candidate concepts for CCDiff to identify meaningful dataset-level differences. The improved performance of BLIP-2 can be attributed to higher-quality and more descriptive captions, likely stemming from its greater model capacity, but good results can still be achieved with significantly more lightweight alternatives.

D.1 Candidate Pool Analysis

Coverage and Oracle Performance The candidate pool matters for any set difference captioning framework, including VisDiff, because every method degrades when the pool does not contain the distinguishing concept. To measure this dependence, for each pair we run the GPT-4 judge over all candidates and retain the best score. Candidate-pool recall records whether any candidate is correct, while oracle Acc@1 is the best possible score under perfect ranking. As shown in Table 10, recall is 1.00 on every VisDiffBench split, indicating that the BLIP-2 pool almost always contains the relevant concept. Oracle Acc@1 reaches 1.00, 0.88, and 0.82 on Easy, Medium, and Hard, respectively, while CCDiff reaches 0.96, 0.76, and 0.70. CCDiff therefore performs close to the oracle ceiling, indicating that the remaining gap is primarily a ranking problem. Moreover, the matched ranking and latent-method ablations use the same BLIP-2 candidate pool and CLIP embeddings, so their controlled gains isolate the inverse-CCA mechanism rather than the pool.

Task-Specific Candidate Sources The practicality of manually curated or no-caption vocabularies depends on the data domain because any candidate source must cover the concepts needed to express the possible differences. For some tasks, an online vocabulary or knowledge bank can provide this coverage. In medical imaging, established radiology lexicons already enumerate many findings of interest and can be used directly as a candidate pool, although our reported MIMIC-CXR experiment uses captions from a domain-adapted BLIP model. When no suitable vocabulary exists, we use BLIP captions grounded in the images. For VisDiffBench, we could not find an online vocabulary that covers its fine-grained and image-specific differences, such as ankle vs. knee-high boots, so we construct the pool from captions following common practice. For DTD, a fixed texture lexicon generated with a single LLM prompt is reused across all pairs, with CCDiff operating on it unchanged. There is therefore no universal recipe, and candidate-pool design remains task-specific, with the goal of covering the possible differences between the pairs of interest.

Alternative Candidate Generator To test whether the performance gain depends on BLIP-2, we evaluate a candidate pool generated by LLaVA-1.5 and compare CCDiff against VisDiff under the same LLaVA-based proposer. As shown in Table 11, CCDiff outperforms VisDiff by a wide margin on every split, most notably on Hard, where it achieves 0.57 Acc@1 compared to 0.28. This result

Table 10: Candidate-pool recall and oracle Acc@1 on VisDiffBench. Recall records whether the pool contains a correct candidate, while oracle Acc@1 is the best possible score under perfect ranking.

Split	Recall	Oracle Acc@1	CCDiff Acc@1
Easy	1.00	1.00	0.96
Medium	1.00	0.88	0.76
Hard	1.00	0.82	0.70

Table 11: Comparison of VisDiff and CCDiff using the same LLaVA-1.5-generated candidate pool on VisDiffBench.

Method	Proposer	Ranker	Easy		Medium		Hard	
			Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
VisDiff	Image (LLaVA-1.5)	Feature (CLIP)	0.71	0.81	0.39	0.49	0.28	0.43
CCDiff	Image (LLaVA-1.5)	inverse-CCA	**0.92**	**0.97**	**0.73**	**0.77**	**0.57**	**0.65**

further confirms that the gain comes from the inverse-CCA discovery and ranking mechanism rather than a particular captioning model, and that CCDiff improves over VisDiff with either BLIP-2 or LLaVA-generated candidates. Both methods perform somewhat worse with the LLaVA pool than with BLIP-2, suggesting that BLIP-2 captions are better suited as a candidate source for this task. The result is also consistent with CCDiff showing larger improvements in settings with subtler differences or noisier candidate captions, such as MetaShift.

Noisy Candidate Pools To stress-test the candidate pool directly, we construct two pools with very different signal-to-noise ratios: a small noisy pool with 50 total candidates, including only 10 true difference concepts, and a large noisy pool with 18,000 total candidates, including only 20–30 true difference concepts, with the remaining candidates serving as random distractors. As shown in Table 12, CCDiff remains robust even when the true concepts form a small fraction of the pool. As described in Section 3.4, before inverse-CCA and final ranking, each concept is scored by its average CLIP similarity to its own set's images, and only the top-$k = 20$ concepts are retained for each set. Random distractors have no systematic relationship to a given image set and therefore receive low, noise-like relevance scores, while true difference concepts are grounded in the images and consistently score higher. This filtering prevents CCDiff from being overwhelmed by scale. The large noisy pool performs slightly better than the small noisy pool because its larger set of candidates gives more opportunities for a true concept or a close paraphrase to survive selection.

Coverage Limitation Although the initial candidate pool is shared, the pool is filtered independently for each set, so opposing concepts are not guaranteed to be retained in both set-specific pools. No ranking method can recover a distinguishing concept that is absent from the candidate pool. The recall and oracle results therefore establish strong empirical coverage for the tested VisDiffBench pools rather than a universal coverage guarantee.

E Ablation Study

Kernel Choice As shown in Table 1, the polynomial and Gaussian kernel variants of CCDiff show strong, consistent performance across all difficulty levels (e.g., achieving 89% and 94% Acc@1 on the Easy split, 68% and 74% on Medium, and 60% and 62% on Hard, respectively). The close performance across kernel choices (linear, polynomial, Gaussian) suggests that the core inverse-CCA framework is stable and not overly sensitive to specific similarity metrics. This empirical result suggests that the discriminative directions for set-level discovery in CLIP embedding space are well captured by linear correlations, and the least-correlated directions identified by CCDiff represent robust semantic axes that do not require complex non-linear projections.

Ranker Ablation To isolate the contribution of inverse-CCA as a ranking module, we compare two configurations: (1) *VisDiff*: GPT-4V proposals with CLIP ranking, and (2) *CCDiff (ablation)*: GPT-4V proposals with inverse-CCA ranking. As shown in Table 13, the CCDiff ranker achieves

Table 12: Effect of candidate-pool quality and scale on CCDiff. Results are reported as Acc@1 and Acc@5 across VisDiffBench difficulty levels.

Candidate Pool	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
Small noisy (50, 10 differences)	0.93	0.97	0.67	0.76	0.60	0.74
Large noisy (18,000, 20–30 differences)	**0.97**	**1.00**	0.69	0.81	0.65	0.77
Default clean BLIP-2	0.96	0.99	**0.76**	**0.86**	**0.70**	**0.79**

Table 13: Ablation study comparing VisDiff and CCDiff (GPT-4V proposals + inverse-CCA ranker) on VisDiffBench across difficulty levels.

Method	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
VisDiff	**0.95**	**1.00**	**0.75**	**0.87**	0.57	0.74
CCDiff	0.91	0.97	0.71	0.83	**0.64**	**0.79**

comparable performance on Easy and Medium splits, and notably improves on Hard (0.64 vs. 0.57 Acc@1), suggesting that inverse-CCA better captures subtle, fine-grained differences.

Pairing Stability To construct cross-set pairs for CCA, we use modulo-based pairing, which satisfies CCA's requirement of equal sample sizes without imposing semantic alignment between individual image–text pairs. Since inverse-CCA captures global distributional structure across the two views rather than instance-level correspondences, the recovered directions remain stable across different pairings. Across 5 runs with different random modulo pairings, we observe consistent performance: Easy (Acc@1: 0.96 ± 0.01, Acc@5: 0.99 ± 0.01), Medium (Acc@1: 0.76 ± 0.01, Acc@5: 0.86 ± 0.02), and Hard (Acc@1: 0.70 ± 0.01, Acc@5: 0.79 ± 0.01), confirming robustness to the specific pairing heuristic.

Matched Candidate-Pool Ranking Baselines To isolate whether the performance gain comes from inverse-CCA rather than the candidate pool or CLIP features, we compare four methods using the same candidate pool and CLIP features, changing only the ranking rule. As shown in Table 14, CCDiff matches or outperforms the three CLIP-based baselines on every split, with the widest margin on Hard. On Easy, the differences are obvious, so simple CLIP similarity is sufficient and the baselines remain competitive. On Hard, where the two sets overlap most, the baseline Acc@1 scores drop to 0.34, 0.56, and 0.56, while CCDiff remains at 0.70. The CLIP-only baseline ranks candidates by average CLIP similarity, the mean-similarity-difference baseline uses the difference in average CLIP similarity between the two sets, and the centroid-difference baseline uses the difference between their CLIP centroids. These baselines summarize the sets through average or centroid differences, whereas inverse-CCA uses the image–text cross-set correlations of both sets, capturing how the two distributions relate rather than only their average gap. By selecting the least-correlated canonical directions, inverse-CCA suppresses attributes shared by both sets and isolates the set-specific difference, providing the strongest benefit on the Hard split. Since the candidate pool and CLIP features are shared across all four methods, this gain comes from the ranking rather than candidate quality.

CCA Projection and Selection Variants Standard CCA without inverse projection, forward-CCA-only scoring, and selection of the most-correlated directions reduce to the same variant in our implementation, which ranks candidates using the most-correlated rather than the least-correlated directions. We evaluate this variant alongside PLS and PCA using the same candidate pool and CLIP features, as shown in Table 15. The PCA baseline uses PCA with LDA and Lasso because plain PCA is unsupervised and does not identify which directions separate the two sets. LDA provides the discriminative direction and Lasso selects the contributing components, giving PCA a fair and competitive form rather than treating it as purely unsupervised.

Inverse-CCA outperforms every alternative on all splits, with the widest margin on Hard. Selecting the most-correlated directions collapses on Hard to 0.19 Acc@1 because these directions emphasize what the two sets share rather than what differs, demonstrating the contribution of the inverse,

Table 14: Matched candidate-pool ranking comparison on VisDiffBench. All methods use the same candidate pool and CLIP features, changing only the ranking rule.

Ranker	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
CLIP-only	**0.96**	**0.99**	**0.76**	0.83	0.34	0.70
Mean-sim diff	0.47	0.89	0.67	0.81	0.56	0.76
Centroid diff	0.79	0.95	0.67	0.81	0.56	0.76
CCDiff (inverse-CCA)	**0.96**	**0.99**	**0.76**	**0.86**	**0.70**	**0.79**

Table 15: Comparison of inverse-CCA with alternative projection and component-selection variants on VisDiffBench. All variants use the same candidate pool and CLIP features.

Variant	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
Most-correlated (standard / forward CCA)	0.84	0.95	0.61	0.77	0.19	0.33
PLS	0.65	0.86	0.57	0.72	0.53	0.66
PCA (+ LDA + Lasso)	0.62	0.93	0.53	0.76	0.50	0.66
CCDiff (inverse-CCA)	**0.96**	**0.99**	**0.76**	**0.86**	**0.70**	**0.79**

least-correlation selection. PLS maximizes covariance rather than correlation, so its least directions are dominated by low-variance noise, leading to weaker performance across all splits. PCA also underperforms because its projection is learned from the images alone and does not capture the image–text cross-set signal. In contrast, changing the inverse-CCA kernel has a comparatively small effect across the linear, polynomial, and Gaussian variants in Table 1. Since the candidate pool and CLIP features are identical across all variants, these results support attributing the gain to the inverse selection and its back-projection to concepts rather than the pool, showing that the inverse-CCA mechanism is not interchangeable with the alternative linear latent-space analyses evaluated here.

Weighting Parameter Sensitivity We evaluate $\alpha \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$ on VisDiffBench, with Acc@1 and Acc@5 reported in Table 16. Setting $\alpha = 0.0$ uses only inverse correlation and performs worst across all reported metrics, with Easy Acc@1 decreasing from 0.96 at $\alpha = 0.3$ to 0.68 and Hard Acc@1 decreasing from 0.70 to 0.46. This result shows that the within-set similarity term acts as a relevance filter that suppresses rare or noisy concepts, keeping only concepts that are genuinely representative of the set and ensuring that a top-ranked concept is not only distinctive but also present in the set. Across the tested values in the range $\alpha \in [0.3, 1.0]$, performance forms a broad, flat plateau, showing that CCDiff is relatively insensitive to the exact value and that these settings are equally competitive. This flat plateau may arise because VisDiffBench is curated and the candidate concept pool has high coverage, allowing inverse-CCA discovery to produce a high-quality set of distinctive candidates for which within-set similarity is often sufficient for final ordering. We hypothesize that the inverse-correlation term may provide greater benefit in noisier or less clearly separated settings, although further evaluation is needed to test this hypothesis. The weighting parameter affects only the final ranking, while inverse-CCA performs candidate discovery for every value of α, so at $\alpha = 1.0$ the final ordering uses only within-set similarity but remains restricted to candidates discovered by inverse-CCA. We keep $\alpha = 0.3$ so the final ranking retains an explicit distinctiveness signal while the similarity term filters out noise, at negligible cost relative to the broad plateau. This empirical value is used unchanged across VisDiffBench, MetaShift, MIMIC-CXR, and BREEDS, with no per-dataset tuning and no different setting for general vision and medical images. We recommend $\alpha = 0.3$ or 0.5 to retain an explicit balance between within-set representativeness and cross-set distinctiveness while avoiding dataset-specific tuning. This insensitivity is practically useful because set difference captioning tasks often lack a validation split.

E.1 Controlled Synthetic Example

Empirical results on set difference captioning benchmarks do not fully quantify the need to balance distinctiveness and similarity in the final ranking because these benchmarks are curated to feature fully separable differences. For example, VisDiffBench includes a comparison between "people practicing yoga in a mountainous setting" (Set A) and "people meditating in a mountainous setting"

Table 16: Sensitivity of CCDiff to the weighting parameter α on VisDiffBench. Results are reported as Acc@1 and Acc@5 across difficulty levels.

α	Easy		Medium		Hard	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
0.0	0.68	0.98	0.50	0.77	0.46	0.71
0.3	0.96	**0.99**	0.76	**0.86**	0.70	**0.79**
0.5	0.96	**0.99**	**0.79**	**0.86**	0.71	**0.79**
0.7	**0.97**	**0.99**	0.78	**0.86**	**0.72**	**0.79**
1.0	**0.97**	**0.99**	0.78	**0.86**	**0.72**	**0.79**

Table 17: Composition of the controlled synthetic image sets. Foreground and background concepts are sampled independently.

		Foreground		Background	
Image Set	Images	Yoga	Meditation	Mountain	Grass
Set A	100	60%	40%	90%	10%
Set B	100	40%	60%	10%	90%

(Set B). Since the only fully separable difference for Set A is "yoga", most candidates in the pool are already related to yoga through inverse-CCA discovery, allowing CLIP similarity to readily rank a correct candidate at top-1.

In more complex settings, differences may arise from multiple and partially overlapping aspects. To demonstrate this setting, we use Stable Diffusion 3.5 Large to construct a controlled synthetic dataset in which each image combines a foreground concept, yoga or meditation, with a background concept, mountain or grass. Each set contains 100 images, with the foreground and background concepts sampled independently according to the distributions in Table 17. The background concept is strongly imbalanced across the two sets, while the foreground concept is only mildly imbalanced. Consequently, the background represents the stronger and more distinctive difference, whereas the foreground is a weaker discriminator.

To isolate the effect of the inverse-correlation metric in the final ranking, we use the controlled candidate vocabulary "practicing yoga", "meditation", "mountain", and "grass" instead of BLIP-2 captions, eliminating candidate-generation quality and coverage as confounding factors. We evaluate which concepts are more true for Set A than Set B, prefixing every candidate with "an image of" when computing its CLIP text embedding. As shown in Table 18, CLIP similarity ranks "practicing yoga" first and "mountain" second, even though mountain represents the stronger and more distinctive difference. This occurs because CLIP tends to assign higher similarity to salient foreground subjects than to background elements, irrespective of the actual difference between the sets. Furthermore, CLIP assigns relatively high similarity to both yoga and meditation images for the yoga concept because they share similar visual patterns.

In contrast, inverse correlation ranks "mountain" as the top-1 difference concept and "practicing yoga" second. When multiple candidates are valid, similarity alone can therefore be suboptimal because of its bias toward prominent foreground concepts and its difficulty in distinguishing fine-grained visual details. Inverse correlation addresses this limitation by explicitly accounting for the contrast between sets.

F Additional Details on MetaShift

MetaShift [Liang and Zou, 2022] is a large-scale and diverse collection of image subsets constructed from Visual Genome, where each subset represents a specific object class in a particular context (e.g., *birds(water)*, *banana(bowl)*). This dataset is significantly noisier [Dunlap et al., 2024] and less curated than VisDiffBench, making it a challenging testbed for generalization. The full dataset contains thousands of subsets across hundreds of object classes.

Table 18: Observed top-1 and top-2 rankings for concepts that are more true for Set A than Set B in the controlled synthetic example.

Ranking Signal	Top-1	Top-2
CLIP similarity	*practicing yoga*	*mountain*
Inverse correlation	*mountain*	*practicing yoga*

F.1 Dataset Preparation

Dataset We construct 1,379 paired sets by pairing intra-class subsets with 100 images each (e.g., *birds(water)* vs. *birds(sky)*, *banana(bowl)* vs. *banana(table)*), representing a $7\times$–$8\times$ increase in scale over VisDiffBench. The pairs span differences from prominent to subtle, covering a wide range of visual contexts. We retain only subsets with at least 100 images to ensure sufficient visual diversity within each set. Subsets are grouped by object class and up to 5 random intra-class pairs are sampled per class. These paired sets span a wide range of object classes and visual contexts. For each set, exactly 100 images are sampled per subset.

F.2 Implementation Details

All experimental settings are identical to VisDiffBench (Section 4.1): BLIP-2 for candidate concept generation with the prompt *"Describe this image in detail,"* and GPT-4 for evaluation, using the same scoring rubric (perfect/partial/no match). The ground-truth difference for each pair is derived from their respective subset names.

F.3 Results

As shown in Table 3, CCDiff outperforms VisDiff on Acc@1 by approximately 33% relatively (0.32 vs. 0.24) while achieving equal Acc@5 (0.46), despite the increased scale and noise. The lower absolute performance of both methods reflects the weaker semantic separation and higher noise inherent to MetaShift [Dunlap et al., 2024], making the task inherently more challenging than VisDiffBench. These results demonstrate that CCDiff generalizes to larger and less curated settings.

Qualitative Examples of MetaShift Figure 4 presents qualitative examples from MetaShift, comparing differences discovered by VisDiff and CCDiff across representative intra-class paired sets, revealing two consistent observations. First, both CCDiff and VisDiff outputs occasionally retain shared context alongside true differences, a behavior inherent to both methods. Second, the shared context retained by CCDiff is distributional, reflecting visual patterns present across the full image set (e.g., the grassland context in *grass(elephant)* vs. *grass(sign)*). In contrast, the shared context in VisDiff outputs tends to reflect salient but non-discriminative intra-set attributes (e.g., "elephants and baby elephants together"), which may not be representative of the broader image distribution.

G Additional Details on MIMIC-CXR

MIMIC-CXR [Johnson et al., 2019a] is a publicly available dataset of 377,110 chest X-rays associated with 227,827 imaging studies, each accompanied by free-text radiology reports (findings and impressions). We construct an evaluation benchmark similar to VisDiffBench from the MIMIC-CXR-JPG dataset [Johnson et al., 2019b].

G.1 Dataset Preparation

Image Set Definitions From a publicly available HuggingFace subset of approximately 30,000 images, we define 26 clinically meaningful image sets corresponding to distinct radiological patterns (Table 19). These sets were derived from common chest X-ray findings described in radiology reports and validated for visual distinctness.

Image to Set Assignment Each image is assigned to exactly one set based on the semantic similarity between its radiology findings and the set definitions in Table 19. We use GPT-4 with a structured prompt that instructs the model to select the single best-matching set according to clinical descriptions.



Figure 4: Qualitative examples from MetaShift, comparing differences discovered by VisDiff and CCDiff for representative intra-class image-set pairs.

The prompt includes: (1) the radiology findings text, (2) the list of set definitions with example phrases, and (3) explicit rules to prioritize specificity and visual distinctness. The model outputs a set label along with supporting evidence; this approach ensures assignments are grounded in clinical semantics. The full prompt is shown in Figure 5. After assignment, we balance each set to 40 images by random sampling. Two sets (S and V) had fewer than 40 images and are discarded, leaving 24 balanced sets.

Set Pair Construction From the 24 balanced sets, we generate 30 paired-sets (10 easy, 10 medium, 10 hard) using a GPT-4 prompt that explicitly defines difficulty in terms of visual subtlety on chest X-rays:

- **Easy:** Large, obvious differences (e.g., normal vs. clearly abnormal, devices vs. none).
- **Medium:** Location- or shape-based differences.
- **Hard:** Subtle distinctions requiring careful inspection or radiological expertise.

The full prompt is shown in Figure 6, and the resulting 30 paired-sets are listed in Table 20.

Implementation Details We use CXR-CLIP [You et al., 2023] (SwinTiny backbone) for image and text embeddings, which is pretrained on MIMIC-CXR. Candidate concepts are generated by a BLIP model (Salesforce/blip-image-captioning-large) that has been finetuned on MIMIC-CXR to produce radiology-style captions. All other components (candidate concept selection, inverse-CCA, and scoring) use the same hyperparameters as in the VisDiffBench experiments (Section 4.1). Both CCDiff and VisDiff are evaluated using the same domain-adapted models. The generated difference descriptions are evaluated by GPT-4 following the same scoring rubric as VisDiffBench [Dunlap et al., 2024] (perfect/partial/no match). The ground-truth difference for each pair is the textual description of the two sets (e.g., "Clear lungs with no acute abnormality" vs. "Pleural fluid at the lung bases").

Results As shown in Table 4, CCDiff outperforms VisDiff across all difficulty splits, achieving Acc@1 scores of 73%, 68%, and 42% on Easy, Medium, and Hard, respectively.

Model Dependence The overall accuracy of both methods is influenced by the quality of the underlying models used for representation and caption generation. In particular, both CCDiff and VisDiff rely on a BLIP model fine-tuned on MIMIC-CXR and a pretrained CXR-CLIP model for

Table 19: The 26 clinical image-sets used in the MIMIC-CXR benchmark.

Set	Description
A	Clear lungs with no acute abnormality
B	Diffuse bilateral lung opacities
C	Localized lung opacity
D	Enlarged cardiac silhouette and Large heart
E	Pleural fluid at the lung bases
F	Air in the pleural space with absent lung markings
I	Medical tubes or implanted devices present
J	Post-surgical changes of the chest
K	Reduced lung volume due to atelectasis
L	Hyperinflated lungs with increased lung volume
M	Chest wall or rib abnormality
N	Abnormal widening or contour of the mediastinum
O	Elevation of hemidiaphragm
P	Upper-lobe-predominant lung abnormality
Q	Lower-lobe-predominant lung abnormality
R	Interstitial lung disease with reticular pattern
S	Ground-glass opacity pattern
T	Pulmonary nodules or lung mass
U	Unilateral lung abnormality
V	Asymmetric lung volumes
W	Apical pleural thickening or scarring
X	Costophrenic angle abnormality without large effusion
Y	Mediastinal or tracheal shift
Z	Focal lucency or cavitary lung lesion

embedding extraction, and limitations in these components directly propagate to the quality of the discovered differences and the resulting evaluation accuracy.

G.2 Qualitative Examples of MIMIC-CXR

The qualitative examples in Figure 7 illustrate that, for a pair (post-surgical chest vs. clear lungs), CCDiff identifies precise and visually grounded distinctions such as *sternotomy wires* and *widened mediastinum*. In contrast, VisDiff produces more generic descriptions (e.g., *mediastinal silhouette changes*) that lack clinical specificity. Similarly, for another pair (abnormal vs. normal mediastinal contour), CCDiff correctly highlights findings such as *enlarged cardiac silhouette* and *tortuous aorta*, while VisDiff suggests unsupported surgical histories (e.g., *status post esophagectomy*) that are not evidenced in the images. This advantage stems from CCDiff incorporating both image and text distributions during difference discovery, ensuring that extracted concepts are grounded in the visual evidence. In contrast, VisDiff relies purely on captions for both proposal and ranking, making it sensitive to caption quality and prone to surfacing descriptions that reflect caption artifacts rather than true visual differences.

H ResNet Failure Mode Details

Model details We use the PyTorch pre-trained ResNet-50 and ResNet-101 models and the Huggingface `facebook/detr-resnet-50` object detector [Carion et al., 2020].

Recovered and novel failure signals CCDiff successfully recovers the *human presence* signal reported by VisDiff, with top-ranked differences such as "a group of people standing around a table" and "a person standing on a wooden floor". Beyond this, CCDiff highlights a previously unreported failure mode: *scene clutter*, captured by descriptions such as "a lot of junk in a room".

The following is a radiology FINDINGS text from a chest X-ray report:

{findings_text}

I am a doctor trying to build an findings classifier. I want to figure out which category does this findings belong to.
A list of categories (Set A/B/C/D/E/F/I/J/.../Z), each with typical phrases, provided below.

- Set A (Clear lungs with no acute abnormality): "lungs are clear", "no focal consolidation, pleural effusion, or pneumothorax", "no acute cardiopulmonary abnormality", "no acute intrathoracic process"
- Set B (Diffuse bilateral lung opacities): "diffuse bilateral opacities", "bilateral airspace disease", "diffuse interstitial and alveolar opacities", "pulmonary edema pattern", "widespread bilateral lung opacification"
- Set C (Localized lung opacity): "focal consolidation", "right/left lower lobe opacity", "localized airspace opacity", "patchy opacity in the [lobe]", "segmental consolidation"
- Set D (Enlarged cardiac silhouette and Large heart): "cardiomegaly", "enlarged cardiac silhouette", "heart size is enlarged", "markedly enlarged heart"
- Set E (Pleural fluid at the lung bases): "pleural effusion", "bilateral pleural effusions", "small/moderate/large effusion", "blunting of the costophrenic angles", "layering pleural fluid"
- Set F (Air in the pleural space with absent lung markings): "pneumothorax", "absent lung markings", "lucent region without vascular markings", "pleural line identified", "collapsed lung"
- Set I (Medical tubes or implanted devices present): "endotracheal tube", "nasogastric tube", "central venous catheter", "pacemaker", "lines and tubes in place"
- Set J (Post-surgical changes of the chest): "sternotomy wires", "status post CABG", "postoperative changes", "surgical clips", "median sternotomy"
- Set K (Reduced lung volume due to atelectasis): "atelectasis", "subsegmental atelectasis", "lobar collapse", "volume loss", "crowding of bronchovascular markings"
- Set L (Hyperinflated lungs with increased lung volume): "hyperinflated lungs", "flattened diaphragms", "increased lung volumes", "emphysematous changes", "COPD changes"
- Set M (Chest wall or rib abnormality): "rib fracture", "multiple rib fractures", "healing rib fracture", "osseous abnormality", "chest wall deformity"
- Set N (Abnormal widening or contour of the mediastinum): "widened mediastinum", "mediastinal enlargement", "abnormal mediastinal contour", "mediastinal mass", "tracheal deviation"
- Set O (Elevation of hemidiaphragm): "elevated hemidiaphragm", "eventration of diaphragm", "asymmetric diaphragms", "right/left hemidiaphragm elevation"
- Set P (Upper-lobe-predominant lung abnormality): "upper lobe opacities", "apical consolidation", "upper lung zone disease", "biapical scarring", "upper lobe fibrotic changes"
- Set Q (Lower-lobe-predominant lung abnormality): "lower lobe opacities", "basilar consolidation", "dependent atelectasis", "bibasilar airspace disease", "lower lung zone involvement"
- Set R (Interstitial lung disease with reticular pattern): "reticular opacities", "interstitial markings", "coarse interstitial pattern", "fibrotic lung changes", "honeycombing"
- Set S (Ground-glass opacity pattern): "ground-glass opacities", "hazy opacification", "partial airspace filling", "diffuse ground-glass pattern"
- Set T (Pulmonary nodules or lung mass): "pulmonary nodule", "rounded opacity", "lung mass", "solitary pulmonary nodule", "multiple pulmonary nodules"
- Set U (Unilateral lung abnormality):"unilateral opacity", "right lung involvement", "left lung consolidation", "asymmetric lung disease"
- Set V (Asymmetric lung volumes): "asymmetric lung volumes", "one lung smaller than the other", "post-surgical volume loss", "compensatory hyperinflation"
- Set W (Apical pleural thickening or scarring): "apical pleural thickening", "apical pleural scarring", "apical cap", "fibrothorax"
- Set X (Costophrenic angle abnormality without large effusion): "blunted costophrenic angle", "pleural thickening at lung bases", "trace pleural effusion", "minimal pleural fluid"
- Set Y (Mediastinal or tracheal shift): "mediastinal shift", "tracheal deviation", "shift of mediastinum toward lesion", "shift away from lesion"
- Set Z (Focal lucency or cavitary lung lesion): "cavitary lesion", "lung cavity", "focal lucency", "thin-walled cyst", "air-filled cavity in lung", "cavitation within consolidation"

Your job: assign the findings to exactly ONE category by choosing the category whose phrase list has the maximum semantic similarity to the findings text.

Important rules:

- Use semantic meaning, not only exact keyword match (synonyms count).
- If multiple categories match, choose the strongest / most specific visual cue.
- If the findings explicitly mention devices or surgery, prefer Set I or Set J over others (they're very distinctive).
- If the findings say "no effusion, no pneumothorax, no consolidation / clear lungs", choose Set A.
- Output must be strict format.

Output format (strict):
Category: Set <letter>
Evidence: <the key phrase(s) or semantic match from the findings>

Again, I want to figure out which category does this findings belong to. Your response:

Figure 5: Prompt used to assign each chest X-ray image to one of the 26 clinical sets.

Quantitative validation For *human presence*, we select classes with a roughly balanced ratio of images with and without detected persons and compare classification accuracy on each subset. For *scene clutter*, we use the number of detected objects per image as a proxy for visual complexity and split images at the median object count. As shown in Table 21, both factors correlate with consistent accuracy drops across ResNet-50, ResNet-101, and their ensemble. Notably, the clutter effect is larger than the human-presence effect (7.4% vs. 2.9% for ResNet-50), suggesting that scene complexity is a stronger, and previously overlooked, predictor of misclassification.

I Stable Diffusion Experiment Details

We use the Huggingface models `CompVis/stable-diffusion-v1-4` (SDv1) and `stabilityai/stable-diffusion-2-1` (SDv2) with a guidance scale of 7.5. And apply the following negative prompt to both models: "bad anatomy, bad proportions, blurry, cloned face, cropped, deformed, dehydrated, disfigured, duplicate, error, extra arms, extra fingers, extra legs, extra limbs, fused fingers, gross proportions, jpeg artifacts, long neck, low quality, lowres, malformed limbs, missing arms, missing legs, morbid, mutated hands, mutation, mutilated, out of frame, poorly drawn face, poorly drawn hands, signature, text, too many fingers, ugly, username, watermark, worst quality".

Pairing Since there exists a one-to-one mapping between $\mathcal{D}_A$ and $\mathcal{D}_B$ through the generation prompt, we construct paired sets by grouping images generated from the same prompts. Differences are proposed on 1,634 PartiPrompts [Yu et al., 2022] images and validated on 10,000 unseen DiffusionDB [Wang et al., 2023] images to confirm generalization beyond the proposal prompts.

Cross-dataset transfer As shown in Table 22, the top CCDiff hypotheses: "strong color contrast", "vibrant colors", "diagram-like images", "framed artworks", and "single centered subjects", achieve comparable AUROC scores on both PartiPrompts and unseen DiffusionDB images. This consistency indicates that these differences reflect systematic model-level properties of SDv2 rather than artifacts of specific generation prompts, providing strong evidence for the generalizability of the discovered differences.

Saturation analysis To quantitatively verify the color difference hypothesis, we compute the mean saturation of generated images. SDv2 consistently produces higher saturation than SDv1 across both

```
You are given a set of chest X-ray visual categories, each representing a distinct and clinically meaningful visual pattern.

Category set:
{
  "Set A": "Clear lungs with no acute abnormality",
  "Set B": "Diffuse bilateral lung opacities",
  "Set C": "Localized lung opacity",
  "Set D": "Enlarged cardiac silhouette and Large heart",
  "Set E": "Pleural fluid at the lung bases",
  "Set F": "Air in the pleural space with absent lung markings",
  "Set I": "Medical tubes or implanted devices present",
  "Set J": "Post-surgical changes of the chest",
  "Set K": "Reduced lung volume due to atelectasis",
  "Set L": "Hyperinflated lungs with increased lung volume",
  "Set M": "Chest wall or rib abnormality",
  "Set N": "Abnormal widening or contour of the mediastinum",
  "Set O": "Elevation of hemidiaphragm",
  "Set P": "Upper-lobe-predominant lung abnormality",
  "Set Q": "Lower-lobe-predominant lung abnormality",
  "Set R": "Interstitial lung disease with reticular pattern",
  "Set T": "Pulmonary nodules or lung mass",
  "Set U": "Unilateral lung abnormality",
  "Set W": "Apical pleural thickening or scarring",
  "Set X": "Costophrenic angle abnormality without large effusion",
  "Set Y": "Mediastinal or tracheal shift",
  "Set Z": "Focal lucency or cavitary lung lesion"
}

Your task:
- Construct 30 comparison pairs from these categories.
- Divide them into three difficulty levels: Easy, Medium, and Hard (10 pairs each).
- Difficulty is defined by how visually subtle the difference would be in a chest X-ray PNG image.

Difficulty guidelines:
- Easy: Large, obvious visual differences (normal vs clearly abnormal, air vs fluid, devices vs none).
- Medium: Differences are visible but require some attention (location-based, shape-based, or size-based).
- Hard: Subtle visual differences that require careful inspection or radiological expertise, but are still visually distinguishable.

Additional rules:
- Do NOT fix any category to always appear as set1 or set2.
- For each pair, randomly decide which category appears as `set1` and which appears as `set2`.
- The left/right ordering must NOT imply which category is more distinctive.
- Each pair must be visually distinguishable in the image alone.
- Do NOT invent new categories, only use the provided set.
- Avoid duplicate pairs across difficulty levels.
- Keep descriptions exactly as written in the category set.

Output format (STRICT — no extra text):

Easy:
{"set1": "...", "set2": "..."}
(repeat for 10 pairs)

Medium:
{"set1": "...", "set2": "..."}
(repeat for 10 pairs)

Hard:
{"set1": "...", "set2": "..."}
(repeat for 10 pairs)
```

Figure 6: Prompt used to generate 30 paired-sets (10 easy, 10 medium, 10 hard), such that each pair is visually distinguishable.

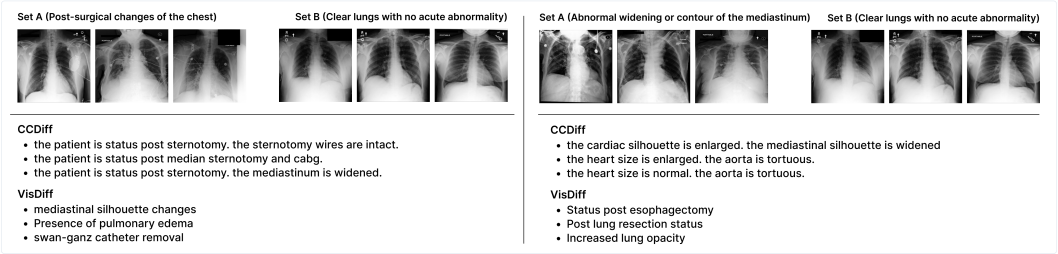

Figure 7: Qualitative comparison between CCDiff and VisDiff on two MIMIC-CXR set pairs (left: easy, right: medium). CCDiff discovers clinically relevant distinctions (e.g., "post sternotomy", "silhouette enlargement"), while VisDiff generates less precise descriptions.

datasets: 100.53 vs. 97.20 on PartiPrompts and 98.03 vs. 93.47 on DiffusionDB. These results are consistent with VisDiff [Dunlap et al., 2024] (112.61 vs. 110.45 on PartiPrompts and 97.96 vs. 93.49 on DiffusionDB), confirming that CCDiff reliably captures this systematic color difference.

Edge analysis Consistent with VisDiff [Dunlap et al., 2024], CCDiff surfaces the finding that SDv2 generated images tend to contain more image frames than SDv1, characterized by white borders consisting of thick, straight lines spanning much of the image. To quantify this, we employ a Canny edge detector and search for straight white lines in the generated images, with a thickness ranging from 5 to 20 pixels and a length exceeding 300 pixels (given the image size of 512×512 pixels). Applying this analysis to DiffusionDB images reveals that 13.2% of SDv2 images exhibit such borders, compared to only 5.1% of SDv1 images, consistent with the VisDiff finding of 13.6% vs. 5.4%.

Table 20: The 30 paired-sets used for evaluation, grouped by difficulty level.

Difficulty	Set-1	Set-2
Easy	Enlarged cardiac silhouette and Large heart	Clear lungs with no acute abnormality
	Clear lungs with no acute abnormality	Diffuse bilateral lung opacities
	Chest wall or rib abnormality	Clear lungs with no acute abnormality
	Clear lungs with no acute abnormality	Air in the pleural space with absent lung markings
	Localized lung opacity	Diffuse bilateral lung opacities
	Clear lungs with no acute abnormality	Pleural fluid at the lung bases
	Clear lungs with no acute abnormality	Medical tubes or implanted devices present
	Post-surgical changes of the chest	Clear lungs with no acute abnormality
	Clear lungs with no acute abnormality	Hyperinflated lungs with increased lung volume
	Pleural fluid at the lung bases	Air in the pleural space with absent lung markings
Medium	Clear lungs with no acute abnormality	Localized lung opacity
	Diffuse bilateral lung opacities	Unilateral lung abnormality
	Lower-lobe-predominant lung abnormality	Clear lungs with no acute abnormality
	Upper-lobe-predominant lung abnormality	Clear lungs with no acute abnormality
	Clear lungs with no acute abnormality	Reduced lung volume due to atelectasis
	Mediastinal or tracheal shift	Clear lungs with no acute abnormality
	Clear lungs with no acute abnormality	Elevation of hemidiaphragm
	Pleural fluid at the lung bases	Costophrenic angle abnormality without large effusion
	Pulmonary nodules or lung mass	Clear lungs with no acute abnormality
	Abnormal widening or contour of the mediastinum	Clear lungs with no acute abnormality
Hard	Lower-lobe-predominant lung abnormality	Localized lung opacity
	Lower-lobe-predominant lung abnormality	Upper-lobe-predominant lung abnormality
	Localized lung opacity	Reduced lung volume due to atelectasis
	Reduced lung volume due to atelectasis	Elevation of hemidiaphragm
	Diffuse bilateral lung opacities	Lower-lobe-predominant lung abnormality
	Interstitial lung disease with reticular pattern	Diffuse bilateral lung opacities
	Upper-lobe-predominant lung abnormality	Apical pleural thickening or scarring
	Clear lungs with no acute abnormality	Costophrenic angle abnormality without large effusion
	Localized lung opacity	Pulmonary nodules or lung mass
	Localized lung opacity	Focal lucency or cavitary lung lesion

Table 21: Validation of CCDiff-derived ResNet failure hypotheses. Both human presence and scene clutter correlate with lower classification accuracy. The clutter signal is a novel finding not reported by VisDiff.

	Human Presence		Scene Clutter	
Model	w/ Person	w/o Person	Cluttered	Simple
ResNet-50	66.60%	69.49%	67.39%	74.82%
ResNet-101	68.16%	71.39%	68.87%	76.45%
Ensemble	73.94%	76.63%	74.43%	81.12%

Qualitative comparison Figure 8 shows randomly sampled image pairs generated from SDv2 and SDv1 using the same DiffusionDB prompts. In each panel, SDv2 images (top row) and SDv1 images (bottom row) are generated from identical prompts, allowing direct visual comparison. The examples clearly confirm our quantitative findings: SDv2-generated images display noticeably more vivid, high-contrast colors and frequently exhibit frame-like white borders, both of which are largely absent in SDv1 outputs.

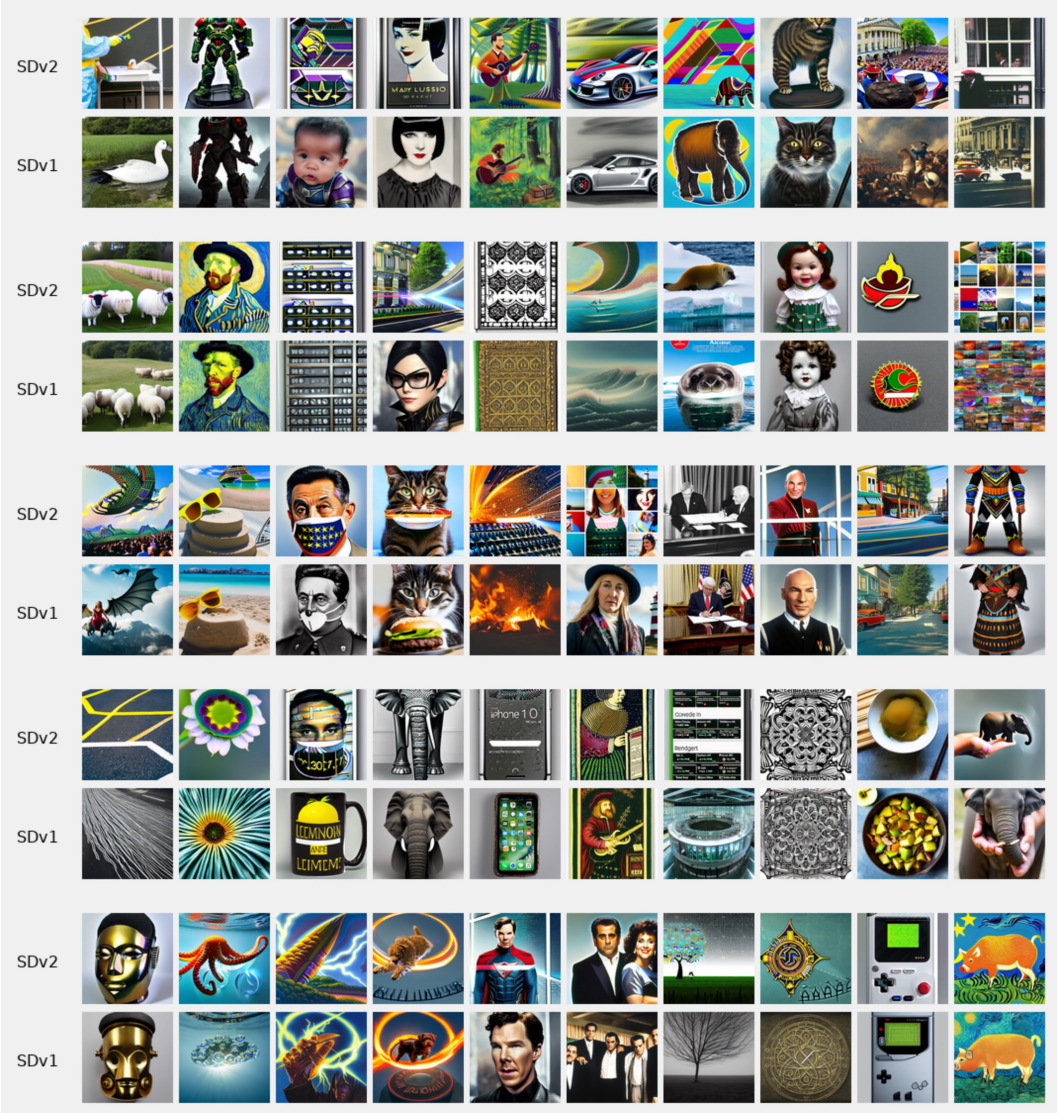

Figure 8: Randomly sampled image pairs generated from SDv2 (top row) and SDv1 (bottom row) using the same DiffusionDB prompts. SDv2 consistently produces images with more vibrant, high-contrast colors and occasional frame-like white borders that are absent in SDv1 outputs.

J Memorability Experiment Details

We use the LaMem dataset [Khosla et al., 2015], where each image is assigned a memorability score by humans. We divide images into two groups: $\mathcal{D}_A$ (most memorable 25th percentile) and $\mathcal{D}_B$ (least memorable 25th percentile).

Qualitative findings CCDiff recovers the broad categories reported by VisDiff: memorable images contain more "humans", "close-up views", and "humorous settings", while providing finer-grained characterization: memorable images are associated with *close-up, attention-grabbing foregrounds*, while forgettable images are dominated by *low-saliency, background-dominant scenes*.

Classification analysis We use CLIP to classify each LaMem test image against the top-25 CCDiff-derived difference descriptions (Table 23) and assign a "memorable" or "forgettable" label based on the highest-scoring description. CCDiff achieves an accuracy of 87.06%, compared to 89.8% reported by VisDiff [Dunlap et al., 2024].

Table 22: Cross-dataset validation of CCDiff hypotheses for SDv2 vs. SDv1. Hypotheses proposed on PartiPrompts generalize to unseen DiffusionDB prompts with comparable AUROC.

Hypothesis (more true for SDv2)	Parti	DiffusionDB
strong color contrast	0.6861	0.7083
vibrant colors	0.7069	0.6994
diagram-like images	0.6894	0.6809
framed artworks	0.6350	0.6261
single centered subjects	0.5891	0.5998

Table 23: Top-25 CCDiff-derived difference descriptions for memorable ($\mathcal{D}_A$) and forgettable ($\mathcal{D}_B$) images on LaMem.

#	Memorable ($\mathcal{D}_A$)	Forgettable ($\mathcal{D}_B$)
1	close-up human body details	sunsets and dusky outdoor scenes
2	close-up facial expressions	cloudy skies over landscapes
3	humorous or exaggerated facial expressions	landscapes with water bodies
4	tattoos or writing on human skin	rivers and water under cloudy skies
5	human-object interactions in close-up	urban cityscapes
6	attention-grabbing foreground subjects	city skylines at dusk
7	disturbing or unusual close-up subjects	buildings under dramatic skies
8	uncanny dolls or altered human-like figures	open scenery with distant subjects
9	focus on specific body parts	expansive outdoor landscapes
10	provocative or unusual body-focused imagery	scenic outdoor environments
11	expressive portraits of individuals	desert or valley landscapes
12	quirky or visually striking personal presentation	natural scenery with trees and water
13	unconventional use of everyday items	afar views of buildings or architectural structures
14	close-up of objects or body parts	environment-focused compositions
15	images with a playful or unusual element	large outdoor background scenes
16	fashion or personal style details	serene and tranquil environments
17	artistic or unnaturally altered human features	stormy or overcast skies
18	collectible or toy-like human figures	picturesque outdoor settings
19	visually intense foreground compositions	background-dominant landscape imagery
20	strong subject-centered imagery	low-subject-saliency outdoor scenes
21	people in casual or humorous situations	agricultural fields and rural scenery
22	detailed description of attire	mountain views from a distance
23	pop culture references and themed costumes	calm waterfront scenes
24	animated or cartoonish characters	clear daytime outdoor settings
25	light-hearted or comedic elements	scenes with a serene and peaceful atmosphere

K Asset Licenses

The existing assets used in this paper have the following licenses:

- CLIP (ViT-bigG-14): MIT License
- BLIP-2: MIT License
- BLIP (image-captioning-large): BSD-3-Clause License
- Kosmos: MIT License
- GIT: MIT License
- MetaShift: MIT License

Table 24: Evaluation on the BREEDS-derived benchmark across 50 paired image sets with 100 images per set.

Method	Acc@1	Acc@5
VisDiff	**0.76**	0.91
CCDiff	0.73	**0.92**

- VisDiffBench: Apache-2.0 license
- MIMIC-CXR-JPG: PhysioNet Credentialed Health Data License 1.5.0
- CXR-CLIP: CC BY-NC 4.0
- DETR: Apache-2.0 License
- StableDiffusion V1/V2: CreativeML Open RAIL-M License
- LaMem: MIT License
- PartiPrompts: Apache-2.0 License
- DiffusionDB: CC0 1.0 License.
- LLaVA-1.5: Llama 2 Community License
- Vicuna-1.5: Llama 2 Community License
- ImageNet: Custom Research License (non-commercial research and educational use only)
- DTD: Available for research purposes

L Broader Impact

Positive Societal Impacts CCDiff enables efficient, interpretable discovery of dataset-level visual differences without relying on large-scale LLM inference, making it accessible in resource-constrained and privacy-sensitive environments. Its applicability to medical imaging (MIMIC-CXR) demonstrates potential for supporting clinical workflows, such as identifying systematic differences between patient cohorts, in settings where proprietary API access is infeasible. More broadly, CCDiff can assist in dataset auditing, model debugging, and fairness analysis by surfacing distributional differences that may otherwise go undetected.

Negative Societal Impacts CCDiff's outputs are bounded by the candidate concept pool and may inherit biases from CLIP and BLIP pretrained representations, which could inadvertently surface or amplify stereotypical associations present in training data. In clinical or other high-stakes settings, automatically discovered differences should be validated by domain experts rather than used as standalone evidence, as incorrect outputs could lead to misleading conclusions. Additionally, since CCDiff surfaces differences based on the provided candidate concept pool, a malicious actor could curate this pool to selectively highlight spurious correlations while suppressing others, potentially framing dataset differences in misleading ways to support a predetermined narrative.

M Additional Details on BREEDS

Dataset Setup We construct a set difference captioning benchmark from BREEDS [Santurkar et al., 2021], consisting of 50 paired image sets from sibling ImageNet superclasses, including *dog vs. wolf*, *car vs. truck*, and *ape vs. monkey*. Each set contains 100 images, and the pairs span fine-grained to broad semantic differences. The two superclass names provide a clean, name-based ground-truth difference for each pair.

Implementation Details We use the same pipeline and GPT-4-based evaluation protocol as VisDiffBench, described in Section 4.1.

Results As shown in Table 24, CCDiff achieves 0.73 Acc@1 and 0.92 Acc@5, compared to 0.76 Acc@1 and 0.91 Acc@5 for VisDiff. These results demonstrate that CCDiff performs on par with VisDiff across the two metrics while avoiding LLM inference during core difference discovery.

Table 25: Performance of CCDiff on the DTD-derived texture benchmark across 50 paired category sets with 100 images per set.

Method	Acc@1	Acc@5
CCDiff	0.89	0.99

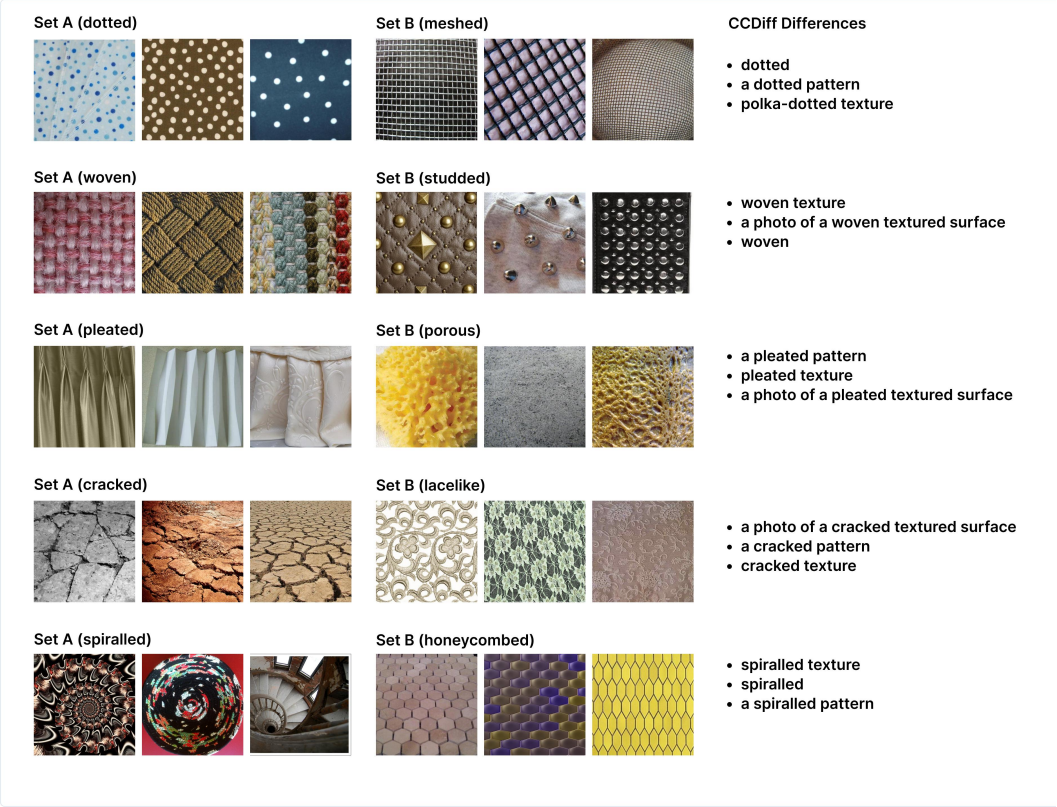

Figure 9: Qualitative examples on DTD. Each row shows Set A, Set B, and the differences discovered by CCDiff for Set A relative to Set B.

N Additional Details on Abstract Visual Differences

DTD Setup We further evaluate CCDiff on the Describable Textures Dataset (DTD) [Cimpoi et al., 2014]. We construct 50 paired image-set comparisons from distinct texture categories, including *striped vs. dotted* and *cracked vs. lacelike*, with 100 images per set. We use a texture vocabulary as the candidate concept pool and follow the same GPT-4-based evaluation protocol described in Section 4.1.

Results As shown in Table 25, CCDiff identifies the distinctive texture with high accuracy, reaching up to 0.89 Acc@1 and 0.99 Acc@5. These results demonstrate that CCDiff extends naturally to fine-grained, abstract texture differences.

Qualitative Results Figure 9 presents five representative comparisons, including dotted vs. meshed, woven vs. studded, pleated vs. porous, cracked vs. lacelike, and spiralled vs. honeycombed. For each pair, CCDiff outputs descriptions that capture the distinctive texture of Set A relative to Set B.

Limitations on Image Distortions We have not explicitly benchmarked image distortions such as blur, noise, or compression. CCDiff surfaces a difference only when the candidate concept and the corresponding visual pattern are aligned in the embedding space, and this alignment comes from the underlying vision-language model, CLIP, which was not trained explicitly to align such low-level patterns with text. Therefore, even with a distortion vocabulary, CCDiff may not recover these axes

reliably. Inverse-CCA itself is a general algorithm for discovering distinctive features between image sets, but its reach is bounded by the image-text alignment of the foundation model on which it is built. This limitation therefore stems from the underlying vision-language foundation model.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The contributions stated in the abstract and Section 1 (lines 65–72), a training-free framework with significantly reduced LLM dependence, for set difference captioning, the novel inverse use of CCA, and the superior or competitive accuracy on benchmarks, are supported by the method (Section 3) and experiments on VisDiffBench, Metashift and MIMIC-CXR (Sections 4.2–4.5).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5 ("Limitations") explicitly discusses that CCDiff's discoverable differences are bounded by the coverage and quality of its candidate concept pool, articulates the trade-off between expressive flexibility and efficiency, and acknowledges that output differences may retain shared context alongside true distinctions — a challenge inherent to other methods as well, in set-level difference discovery.

   Guidelines:

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

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not introduce new theorems or formal theoretical results. Section 3 derives a procedural algorithm rather than proving theoretical guarantees.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Algorithm 1 specifies the full CCDiff procedure, Sections 3.2–3.6 describe each stage in detail, Section 4.1 lists the backbones, captioner, top-k, weighting parameter α, and evaluation protocol for VisDiffBench, similarly 4.4 and 4.5 respectively for MetaShift and MIMIC-CXR dataset. Appendices F and G documents the MetaShift and MIMIC-CXR paired-sets dataset construction details.

   Guidelines:

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

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

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

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

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

   Answer: [Yes]

   Justification: Code is provided as part of the supplementary material, along with instructions to reproduce the main experimental results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All training and test details are specified in Section 4.1 (backbone, captioner, hyperparameters, evaluation protocol) and Sections 4.4–4.5 and Appendices F–G for MetaShift and MIMIC-CXR. CCDiff is training-free, so optimizer details are not applicable.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Appendix B.3 reports 95% bootstrap confidence intervals over image-set pairs and paired-bootstrap significance tests for the VisDiffBench results. It identifies the image-set pairs as the resampling unit, reports confidence intervals and p-values, and discusses the limitation of having 50 pairs per split. The Pairing Stability analysis in Appendix E also reports mean $\pm$ standard deviation across 5 runs with different random modulo pairings.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments were conducted on a single NVIDIA GeForce RTX 3090 (24 GB). Runtime and memory details are provided in Section 4.3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses only publicly available datasets: VisDiffBench [Dunlap et al., 2024], MetaShift [Liang and Zou, 2022], and MIMIC-CXR-JPG [Johnson et al., 2019b], which is a credentialed-access medical dataset available through PhysioNet and properly cited. No new human-subjects data was collected, no personally identifiable information is used or disclosed, and the research does not involve any form of human annotation or crowdsourcing. The method is designed for dataset auditing and model analysis, with no

apparent potential for direct harm. The paper presents no deviation from the NeurIPS Code of Ethics.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Both positive and negative societal impacts are discussed in Appendix L (Broader Impact). Positive impacts include enabling efficient dataset auditing, model debugging, and clinical workflow support in resource-constrained settings. Negative impacts include potential bias amplification from pretrained representations and possible misuse to selectively highlight spurious correlations.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper releases no high-risk artifacts: CCDiff is a training-free analysis pipeline built on top of already-public foundation models (CLIP, BLIP-2, CXR-CLIP), and no pretrained generator, dataset, or new high-risk model is introduced.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All third-party assets are credited via citation throughout the paper. License details for all assets are explicitly listed in Appendix K.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: Code is provided in the supplementary material with instructions to reproduce the main experimental results.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: The work does not involve any crowdsourcing experiments or new human-subjects data collection.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper conducts no new human-subjects research; it relies on the previously released, de-identified MIMIC-CXR-JPG dataset, whose original collection and IRB oversight were handled by its creators.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LLM usage is fully disclosed throughout the paper. GPT-4 is used as an automated evaluator for VisDiffBench, MetaShift, and MIMIC-CXR (Sections 4.1, 4.4, and 4.5), and for MIMIC-CXR paired-set construction (Appendix G), with exact prompts shown in the appendix. BLIP-2 is used as a lightweight captioning model for candidate concept generation (Section 3.3). The core CCDiff method (inverse-CCA filtering and ranking) does not rely on any LLM.

Guidelines:

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