DeClip: Hard Negative Mining through Image Decomposition and Inpainting

Zhanhao Liu (zhanhaol@umich.edu), Huanchen Jia (jhuanch@umich.edu),
Qiulin Fan (rynnefan@umich.edu), Lingyu Meng (jmly@umich.edu)

05/01/2025

Abstract

Contrastive Language–Image Pretraining(CLIP) excels in multimodal learning, enabling zero-shot classification, cross-modal retrieval, and transfer learning. However, its reliance on global image-text alignment limits its ability to capture localized features, weakening performance in fine-grained visual tasks.[1] To address this, we propose DeCLIP, which introduces image-based hard negatives by modifying key image-caption pairs, enhancing model robustness and discrimination. This improves CLIP's ability to distinguish fine-grained details, strengthening its effectiveness in contrastive learning and multimodal tasks.

1 Introduction

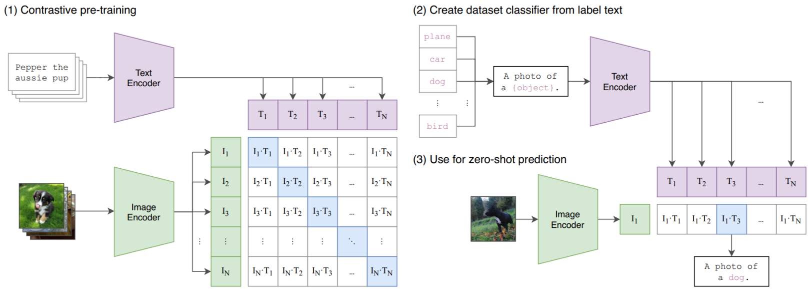
(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T1 T2 T3 ... TN
I1 I2 I3 ... IN
I1·T1 I1·T2 I1·T3 ... I1·TN
I2·T1 I2·T2 I2·T3 ... I2·TN
I3·T1 I3·T2 I3·T3 ... I3·TN
IN·T1 IN·T2 IN·T3 ... IN·TN
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I1
I1·T1 I1·T2 I1·T3 ... I1·TN
A photo of a dog.

Figure 1: Original CLIP Model

CLIP[2] (Radford et al., 2021), trained on full images, struggles to learn localized features due to its reliance on global image-text alignment, limiting its effectiveness in fine-grained visual-text tasks like object classification and caption-based retrieval. This limitation hinders its ability to capture subtle visual differences, which is especially problematic in safety-critical domains such as medical imaging and robotics, where precise distinctions—like identifying diseased tissue or ensuring accurate tool positioning—are crucial. Solving this problem would improve CLIP's ability to learn localized visual details, enhancing its adaptability to tasks requiring fine-grained reasoning and ultimately making it more effective in real-world multimodal applications.

https://github.com/SaddySamoyed/545-project

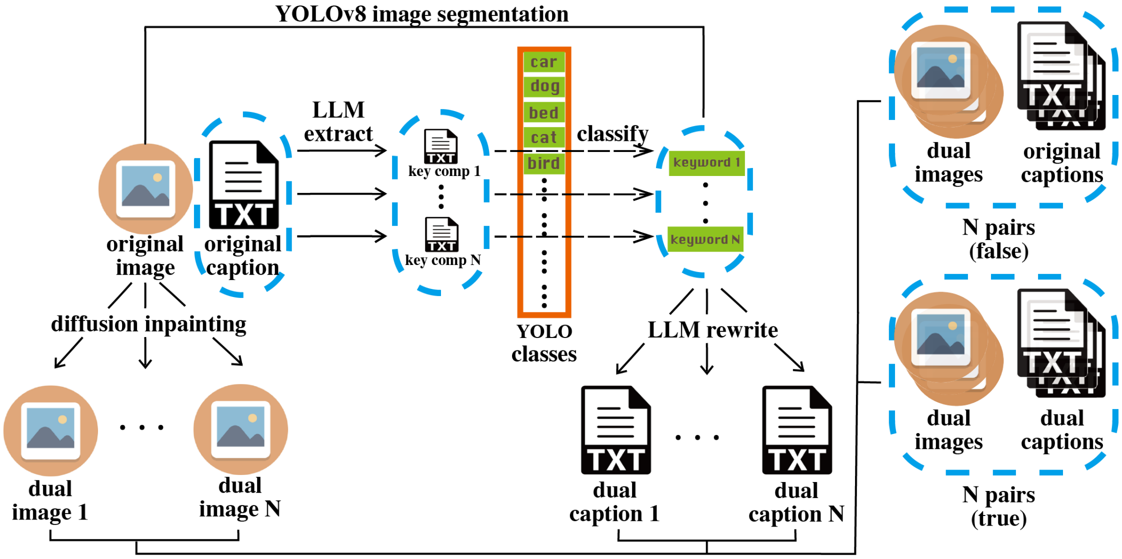

Figure 2: Overall approach

2 Proposed method

2.1 Overall Approach

We propose DeCLIP, an image-side hard negative mining architecture as a way to enhance CLIP's attention to fine-grained features and improve its robustness in real-world multimodal tasks.

Prior research suggests that, for each pair of correct (image, caption) example, by rewriting the caption, e.g. reordering and changing its constituents, we can generate into a hard negative. (Fan, Krishnan, et. al.[3]). Our idea has similarities and extensions.

For each ground-truth pair (I, C), we remove certain key visual components (e.g. the main object or salient regions) from the image I, yielding I' that remains highly similar in low-level appearance yet no longer matches its original caption C, thus forming a hard negative for contrastive learning. Simultaneously, we excise the corresponding keywords from the caption C to create a modified caption C' as a new positive example, reinforcing the model's sensitivity to missing information. We prioritize the image side because the visual embedding manifold is both higher-dimensional and exhibits superior topological smoothness, allowing fine-grained decomposition and recomposition of localized features that text-only perturbations cannot capture. Although constructing image-level hard negatives incurs greater computational and annotation cost compared to text-level augmentations, it significantly amplifies sample diversity and robustness—especially on small datasets—yielding more discriminative vision–language alignment than text-only methods.

Given an image-text pair:

- Removing a key object from the image creates a new image we call **dual image**, together with the caption unchanged → a semantically mismatched hard negative.
- Removing corresponding keywords from the caption creates a new caption we call **dual caption**, together with the altered image → a new positive pair.

Below is our algorithm in concrete:

Algorithm 1 Generating Dual Image-Caption Pairs via Component Removal

```
Input: Batch size $K$, data $\{(I_i, C_i)\}_{i=1}^{K}$, pretrained LLM, segmentation model (e.g., YOLOv8), diffusion inpainting model
Hyperparameter: Number of key components to extract $N$ (e.g., $N = 3$)
for $i = 1$ to $K$ do
    Step 1: Use LLM to extract up to $N$ key components from caption $C_i$
    Classify extracted keywords into predefined object classes, forming a keyword-class map
    for $n = 1$ to $N$ do
        Step 2: Try to detect the $n^{\text{th}}$ component in image $I_i$ using a segmentation model
        if detection succeeds then
            Save mask $M_{i,n}$ for component $n$
        else
            Remove the $n^{\text{th}}$ keyword-class mapping from the keyword map, continue
        end if
    end for
    for $n = 1$ to $N$ do
        Step 3: Use diffusion model to remove component $n$ by applying mask $M_{i,n}$ to $I_i$, producing $I'_{i,n}$
    end for
    for $n = 1$ to $N$ do
        Step 4: Use LLM to remove keyword $n$ from caption $C_i$, producing dual caption $C'_{i,n}$
    end for
end for
Output: $\{(I'_{i,n}, C'_{i,n})\}_{i=1,\dots,K}^{n=1,\dots,N}$ — Dual image-caption pairs with one component removed
```

2.2 Reasonability of the Idea

Our idea invokes the manifold hypothesis, aims at improving CLIP's composition-reasoning ability. Prior work has shown that the embedding manifold of CLIP is characterized as double-ellipsoids[4] (Levi, Gilboa, 2024), and we hypothesize: dual images with critical components removed characterize the local structure of the original image, as a valid interpolation; and the generation of dual texts can let the model learn the mapping between the sample image and the open neighborhood near the caption. We conjecture that our approach increases the combinatorial inference ability and robustness of the model on training with small datasets.

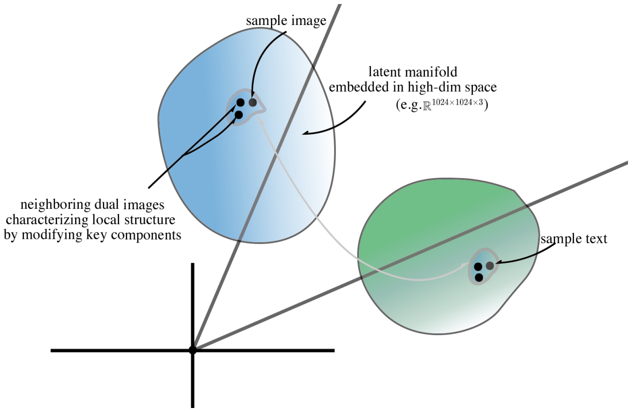
sample image
latent manifold
embedded in high-dim space
(e.g.$\mathbb{R}^{1024\times1024\times3}$)
neighboring dual images
characterizing local structure
by modifying key components
sample text

Figure 3: embedding data manifold of CLIP space

3 Related work

3.1 Existing Methods:

CLIP's performance is shown recently that it could
be enhanced by leverging the difficulty in training.
Currently the exisiting methods mostly improve in the following perspectives.

- **Hard Negative Mining**: Hard negatives closely resemble positive samples yet belong to different classes, making them difficult to distinguish. In contrastive learning, two main strategies address this: generation-based and regularization-based. Generation-based methods (e.g., NegCLIP[5] ((Yuksekgonul et al., 2022)) modify positive samples to create semantically similar but incorrect pairs, forcing models to learn subtle distinctions. Regularization-based methods use techniques like intra-modal contrastive loss and cross-modal ranking of hard negatives to refine learning.
- **Fine-Grained Cross-Modal Alignment Fine-Tuning**: Some previous works such as FILIP[6] (Yao et al., 2021) enhances multimodal alignment by matching local image patches with specific text tokens via a similarity matrix. Unlike CLIP's global image-text matching, it computes token-level contrastive losses, forcing visual and textual tokens to mutually select their most relevant counterparts. The regional information and details could be better learned this way.

DeCLIP combines the idea of **Hard Negative Mining** and **Fine-Grained Cross-Modal Alignment Fine-Tuning** such that the model can deepen its understanding of the entire image by paying attention to the regional changes of the image. DeCLIP is different from existed **Hard Negative Mining** in applying the changes to **images** instead of **texts**, while it is improved based on **Fine-Grained Cross-Modal Alignment Fine-Tuning** by considering contexts of texts and layout of images

3.2 Core Model and Tool References:

Our framework integrates state-of-the-art multimodal tools. For text extraction, we query GPT-4o to identify key noun words from the caption. We then map these nouns to 80 specified catogories of `yolov8n-seg.pt`. This whole process is also conducted by GPT-4o. For image segmentation, `yolov8n-seg.pt` [7] (Ultralytics, 2025)(a lightweight YOLOv8 variant pretrained on COCO) is used.
Inpainting leverages `runwayml/stable-diffusion-inpainting` [8](Stable-diffusion-v1-5/Stable-diffusion-inpainting · Hugging Face, n.d.) and `CLIP`, ensuring robust image–text understanding.

4 Pipeline Instance Demonstration

We have designed and implemented a pipeline that generates hard negative samples by removing key visual objects from an image and rewriting the associated caption accordingly. The resulting (image, caption) pairs enhance the CLIP model's ability to distinguish subtle, localized visual-text misalignments. This section presents an example of our four-step procedure.

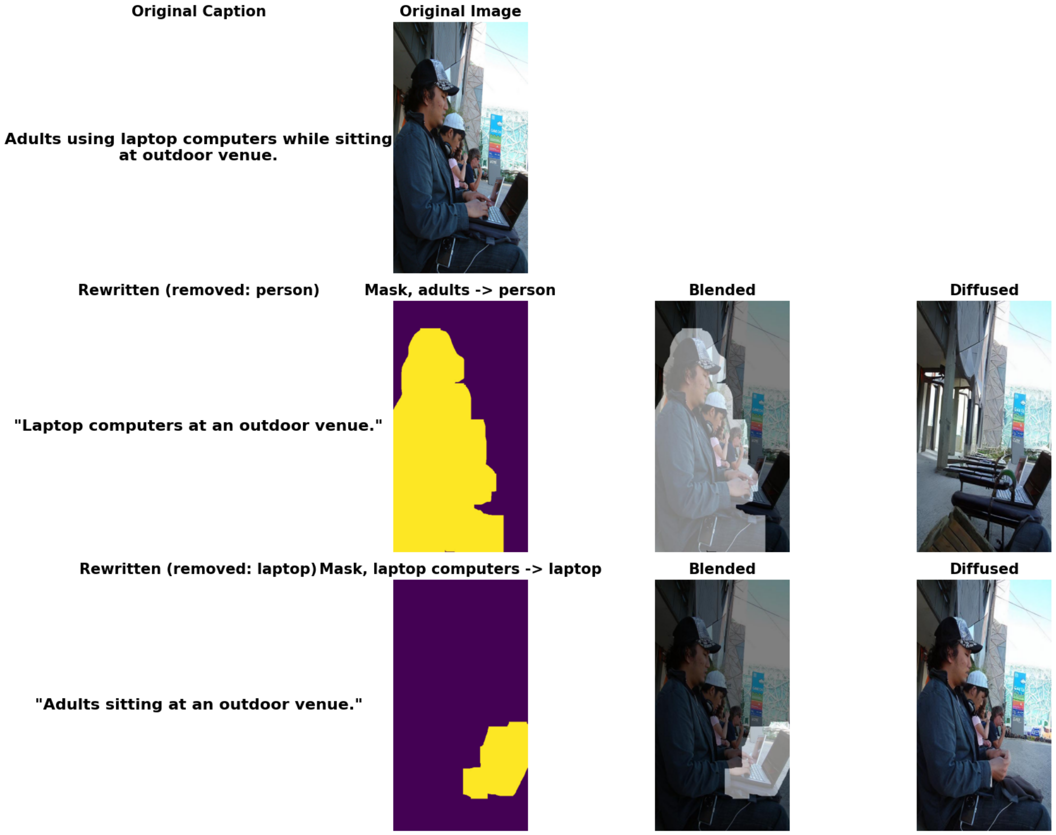
Original Caption
Original Image
Adults using laptop computers while sitting at outdoor venue.
Rewritten (removed: person)
Mask, adults -> person
Blended
Diffused
"Laptop computers at an outdoor venue."
Rewritten (removed: laptop)
Mask, laptop computers -> laptop
Blended
Diffused
"Adults sitting at an outdoor venue."

Figure 4: A random sample of dual (image, caption) pairs generated by DeCLIP. The left column shows the original image and caption. The middle columns show object-level masks (e.g., person, laptop) and corresponding rewritten captions. The right columns show inpainted images after object removal.

Step 1: Caption Parsing and Key Term Extraction **Method:** We use GPT-4o to extract noun phrases from the caption. Multi-word expressions like "laptop computers" are treated as single semantic units. **Example:**

- Extracted Terms: "adults", "laptop computers", "outdoor venue"

Step 2: Mapping Phrases to Object Categories **Method:** Extracted phrases are matched to YOLOv8[7] object categories using LLM-rewrite. Phrases that cannot be reliably mapped or detected are discarded. **Example:**

- `adults` → `person`, `laptop computers` → `laptop`

Step 3: Object Detection and Mask Generation **Method:** YOLOv8 segmentation detects the target objects and produces binary masks. Each mask isolates a specific object (e.g., person or laptop) for subsequent inpainting.

Step 4: Object Removal and Caption Rewriting **Method:** Detected objects are removed from the image using a Stable Diffusion inpainting model[8]. Simultaneously, the caption is modified by removing the corresponding phrase, forming the "dual caption".
Resulting Pairs:

- Original: "Adults using laptop computers while sitting at outdoor venue."
- Dual 1: "Laptop computers at an outdoor venue." (person removed)
- Dual 2: "Adults sitting at an outdoor venue." (laptop removed)

5 Fine-tune

We explored several strategies for adapting CLIP to our specialized, localization-focused dataset. First, we attempted full-parameter fine-tuning but found that, given our relatively small training set, this led to overfitting and degraded generalization. To address this, we froze all of CLIP's weights except the top three transformer layers of the text encoder and the top six layers of the visual encoder. This partial-freeze scheme preserves CLIP's broad, pretrained representations while providing capacity for learning finer, localized patterns. Because our "dual" images share high-level semantics with the originals but differ only in localized details, we found it counterproductive to push their embeddings arbitrarily far apart. Instead, we introduce a dual-caption hinge loss that penalizes precisely two error modes:

1. when a dual image is more similar to the original caption than the true image is, and
2. when a dual caption is more similar to the original image than the true caption is.

This targeted penalty encourages CLIP to focus on the subtle, localized differences encoded by our dual examples without overriding its general alignment. Finally, we combine this hinge term with the standard CLIP cross-entropy loss (InfoNCE) so that the model retains its pretrained zero-shot retrieval performance even as it acquires new, location-sensitive discriminative ability. Notation setup for sample i:

1. v_i: normalized embedding of the true image.
2. t_i: normalized embedding of the true caption.
3. $D_i = \{d_{i,1}, \dots, d_{i,M_i}\}$: normalized embeddings of the dual captions.
4. $N_i = \{n_{i,1}, \dots, n_{i,M_i}\}$: normalized embeddings of the negative images.

$$\mathcal{L}_{\text{dual},i} = \sum_{k=1}^{M_i} \max\left(0, sim(v_i, d_{i,k}) - sim(v_i, t_i)\right). \tag{1}$$

$$\mathcal{L}_{\text{neg},i} = \sum_{j=1}^{M_i} \max\left(0, sim(n_{i,j}, d_{i,j}) - sim(n_{i,j}, t_i)\right). \tag{2}$$

Table 1: Zero-shot Performance to cifar10 and ARO

Method	**Zero-shot on cifar10**	**ARO**
base clip	0.1845	-
Tuned clip	0.1876	0.49129553
D-dip	0.1964	0.49881730

6 Result

To set up our experiments, we first trained CLIP from scratch for a limited number of epochs to establish a base model. We then fine-tuned this base model on the original dataset - this variant is referred to as **Tuned CLIP**. In parallel, we fine-tuned the same base model on our generated dual-image/dual-caption dataset, yielding the **D-CLIP** model for comparison. Notably, because we filtered out unhelpful objects from the captions, roughly 30 % of the images in the original dataset could not be paired with a corresponding dual image and caption. As a result, the dataset used to train D-CLIP is significantly smaller than the one used to train Tuned CLIP.

The results show better performance for D-CLIP compared with the Tuned clip in Cifar 10, indicating that our dataset has helped CLIP enhance its general ability while learning localized features. In the ARO benchmark, D-CLIP shows a slight increase in accuracy over the Tuned clip. Although the improvement is modest, D-CLIP is trained on a smaller dataset than the Tuned clip - only 70% of the original examples were used for dual generation. This demonstrates that, with our dataset, we can achieve similar or better results with reduced storage and computational costs. Given the size of our training data, we believe the results provide a convincing signal that our pipeline successfully enhances the CLIP model's localized feature learning and general performance.

7 Limitations

7.1 Time and Computational Cost Analysis

Generating dual examples scales linearly with dataset size but remains dominated by diffusion sampling. On an A100 GPU we process 1 000 COCO images at a time and observe ≈59 min per batch (≈3.54 s per image–caption pair, of which ≈2.9 s is diffusion). Projected to all 82 783 COCO training examples, dual data generation takes ≈81.4 hours. For larger collections, lighter generators (e.g. VAE-based) will be needed to trade off image quality against run time.

7.2 Limitation of Performance by Error Labelling

Since our fine-tuning relies on automatically generated "dual" examples, a small fraction of labels can be incorrect. Two main error modes arise:

- Diffusion hallucinations: the model sometimes inserts an unrelated object into the background, breaking the intended mismatch.
- Over-reduction in captions: when the original caption contains only nouns, removing it leaves almost no content, so the LLM resorts to generic rewrites (e.g. "a man" → "somebody") that fail to change semantics.

7.3 Forgetting Previously Learned Information

Since our model is intentionally trained to distinguish images and captions based on only subtle feature changes, there is a risk of disrupting CLIP's original capabilities.

8 Conclusion

In this work, we developed a caption-guided object removal pipeline that leverages YOLOv8-based object detection, stable diffusion inpainting, and large language models to extract key objects from image-caption pairs, classify them into predefined categories, and generate precise masks for targeted object removal. By modifying images, while preserving structural relevance, we created hard negative images that misalign with their original captions, enhancing contrastive learning in CLIP-based models and improving their ability to distinguish fine-grained differences between text and images. Additionally, the altered images were incorporated into the training dataset as false samples to fine tune the existing model. Our model was tested on CIFAR-10 and the ARO benchmark, and it demonstrated convincing results. While our approach demonstrates the feasibility, future work should focus on refining object classification, supporting multi-object handling, and scaling data generation to improve DeClip's effectiveness in multimodal learning.

9 Future Plan

- We will need to automate the procedure for relatively large dataset (e.g. 3 million images), requiring the balance between efficiency and quality of inpainting, otherwise we might introduce noisy generated images. So we need to tweak the tech stack and architecture slightly after practice.
- We will further optimize the structure: if a class of examples are already well-learned, then we do not need to improve the accuracy of shotting concerning images. We will not generate $2N$ new examples for all existing examples, but reselect a subset of all images in a dataset for computation cost.

Contribution

- **Zhanhao Liu**
  - Implemented the *word matching* and *Stable Diffusion* code.
  - Develop multiple fine-tuning methods and train CLIP model
  - Conduct experiments and benchmarks
- **Huanchen Jia**
  - Researched state-of-the-art methods and reviewed the literatures for the group.
  - Implemented the *noun extraction* and *compound word emerging* and *LLM-rewrite* code.
  - Data generation and organization for the model to process.
- **Qiulin Fan**

  - Conceived the idea and drawed the demonstration images
  - Implemented the *object detection, mask generation and blend demonstration* code.
  - Stringed together the code in the pipeline, and written debugs

- **Lingyu Meng**
  - Researched *datasets and benchmarks* for future work.
  - Organize the overall project content and design the poster layout

References

[1] S. Liu, Y. Lyu, H. Lee, and T. C. Hollon, "An empirical study of CLIP fine-tuning with similarity clusters," in *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024. [Online]. Available: `https://openreview.net/forum?id=NmNmlAEBAl`.

[2] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: `2103.00020 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2103.00020`.

[3] L. Fan, D. Krishnan, P. Isola, D. Katabi, and Y. Tian, *Improving clip training with language rewrites*, 2023. arXiv: `2305.20088 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2305.20088`.

[4] M. Y. Levi and G. Gilboa, *The double-ellipsoid geometry of clip*, 2024. arXiv: `2411.14517 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2411.14517`.

[5] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, *When and why vision-language models behave like bags-of-words, and what to do about it?* 2023. arXiv: `2210.01936 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2210.01936`.

[6] L. Yao, R. Huang, L. Hou, *et al.*, *Filip: Fine-grained interactive language-image pre-training*, 2021. arXiv: `2111.07783 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2111.07783`.

[7] Ultralytics, *Segment - ultralytics yolov8 docs*, Accessed: 2025-03-13, 2023. [Online]. Available: `https://docs.ultralytics.com/tasks/segment/`.

[8] S. AI, *Stable diffusion v1-5 inpainting*, Hugging Face Model Hub, Accessed: 2025-03-13, 2023. [Online]. Available: `https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-inpainting`.