
Contrastive Localized Language-Image Pre-Training

Hong-You Chen^{† 1} Zhengfeng Lai¹ Haotian Zhang¹ Xinze Wang¹ Marcin Eichner¹ Keen You¹
Meng Cao¹ Bowen Zhang¹ Yinfei Yang¹ Zhe Gan¹

Abstract

CLIP has been a celebrated method for training vision encoders to generate image/text representations facilitating various applications. Recently, it has been widely adopted as the vision backbone of multimodal large language models (MLLMs). The success of CLIP relies on aligning web-crawled noisy text annotations at *image levels*. However, such criteria may be insufficient for downstream tasks in need of fine-grained vision representations, especially when understanding *region-level* is demanding for MLLMs. We improve the localization capability of CLIP with several advances. Our proposed pre-training method, **Contrastive Localized Language-Image Pre-training (CLOC)**, complements CLIP with region-text contrastive loss and modules. We formulate a new concept, *promptable embeddings*, of which the encoder produces image embeddings easy to transform into region representations given spatial hints. To support large-scale pre-training, we design a *visually-enriched and spatially-localized captioning* framework to effectively generate region-text labels. By scaling up to billions of annotated images, CLOC enables high-quality regional embeddings for recognition and retrieval tasks, and can be a drop-in replacement of CLIP to enhance MLLMs, especially on referring and grounding tasks.

1. Introduction

Vision-language (VL) pre-training has been an important foundation for the recent tremendous growth of multimodal applications. Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021; Jia et al., 2021) is a successful VL representation learning method that connects im-

[†]Work done while at Apple. ¹Apple AI/ML. Correspondence to: Zhe Gan <zhe.gan@apple.com>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

ages and text by contrastive training on large-scale data collected from the Web. Strong transferability and generalizability have been proven in extensive downstream tasks such as zero-shot image classification and image-text retrieval. Even beyond, language-supervised methods like CLIP have become arguably the default choice of vision backbone for multimodal large language models (MLLMs) (Liu et al., 2023; McKinzie et al., 2024) due to their superior prior knowledge in aligning vision and language (Tong et al., 2024).

As VL research gets increasing attention, various advanced multimodal tasks are demanding stronger vision capabilities. For instance, recent MLLMs (Rasheed et al., 2024; Ren et al., 2024; Lai et al., 2023; Chen et al., 2023; Peng et al., 2023; Zhang et al., 2025) have been focusing on more fine-grained understanding tasks that require comprehension of the semantics at *region levels* such as visual question answering (VQA) with referring and grounding instructions. These MLLMs are fine-tuned on referring and grounding data with CLIP as the vision backbone, as seen in works like Kosmos-2 (Peng et al., 2023) and Ferret (You et al., 2023; Zhang et al., 2024). Due to the need for such region-level understanding, CLIP, which aligns entire images with text captions, seems insufficient, as its image-text contrastive loss primarily emphasizes global semantics.

To remedy such core localization capability for CLIP, we ask a challenging and fundamental question: *can we pre-train a stronger image encoder (1) with enhanced localization capability that can be inherently integrated into MLLMs, (2) and refine CLIP’s original image embeddings to the region level for zero-shot recognition/retrieval?*

Here, we explore a data-driven approach that complements the original CLIP image-text pre-training objective with explicit region-text supervision. Though conceptually simple, several challenges exist. *First*, it lacks public datasets with region-text annotations at scales large enough for CLIP training, which typically requires hundreds of millions even billions of images. Existing region-text corpus like Visual Genome (Krishna et al., 2017) contains about 108K images, and the largest noisy-labeled grounded dataset GRIT (Peng et al., 2023) features only around 20M images. Indeed, such a deficiency of labeled datasets has probably

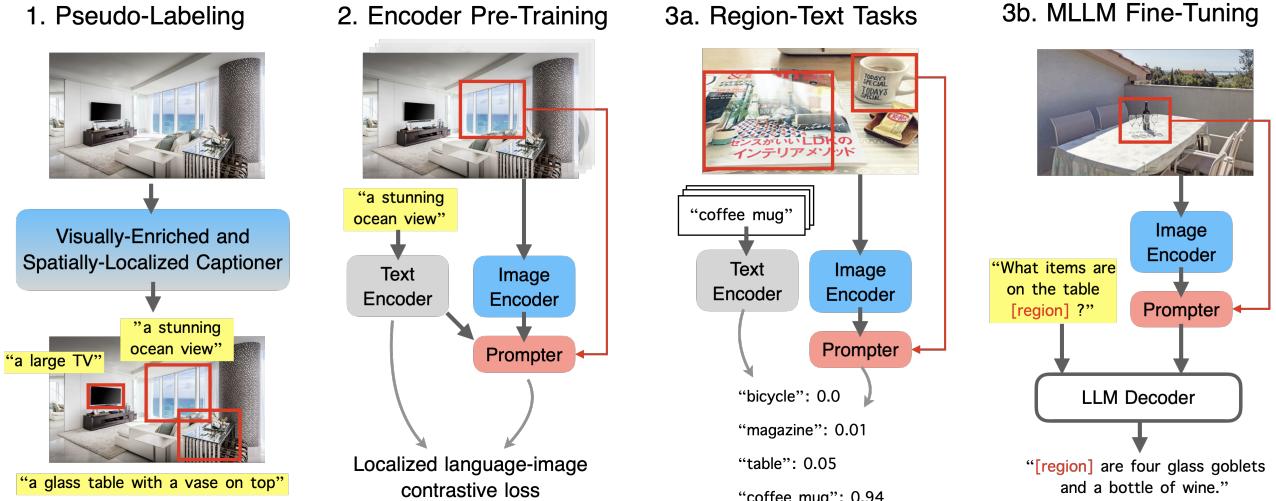


Figure 1. Our CLOC pre-training framework. (1) A visually-enriched and spatially-localized captioning pipeline pseudo-labels bounding boxes with detailed descriptions for key regions. (2) A lightweight Prompter attached on the CLIP image encoder can be prompted to transform the image embedding into region-focused features. All parameters are trained end-to-end from scratch with our contrastive localized language-image loss on the annotated region-text datasets. After pre-training, (3a) region features can be generated via the Prompter for region-text tasks like object classification in a training-free fashion. (3b) The image encoder, along with the optional Prompter, can also strengthen MLLMs fine-tuning by enhancing their fine-grained image understanding capabilities.

limited the literature to mainly consider semi-supervised or weakly-supervised approaches as somewhat a compromise (Naeem et al., 2023; Yao et al., 2022; 2023a).

Second, a plausible solution is to scale up training data in pursuit of image regions pseudo-labeled with text annotations via some open-vocabulary detectors (Minderer et al., 2024; Zhang et al., 2022). Though it seems feasible, we found it non-trivial to design such a pipeline as the annotations are noisy and will greatly affect the final model performance. *Third*, even if the region-text datasets are given, it is under-explored how to effectively train on them in terms of co-designs of training objectives, model architecture, and more design details.

To this end, we propose a new pre-training framework illustrated in Figure 1, named **Contrastive Localized Language-Image Pre-Training (CLOC)**, to improve CLIP with better localization capability, especially for MLLMs, by overcoming the above difficulties. Our main contributions are:

- We propose a new learning goal, **Promptable Embeddings**, that *a strong vision encoder should produce image embeddings that can be easily transformed into region representations, given some spatial hints (e.g., box referring or text prompts)*. This formulation not only facilitates the encoder as a prior of fine-grained VL alignment, but also enables new possibilities for the interactions between the image encoder and the language decoder.
- To optimize towards the goal, we design simple and minimal modifications on top of CLIP. We augment the orig-

inal CLIP loss with a region-text contrastive loss, where the region embeddings are extracted from the image embedding by a lightweight extractor module conditioned on the spatial hints (*i.e., prompts*).

- We design a large-scale pseudo-labeling engine to support CLOC training. We combine visual-enriched image captioners and open-vocabulary detectors for an effective recipe that improves previous practice of region annotations (Minderer et al., 2024; Peng et al., 2023). This approach yields a two-billion image-text dataset with fine-grained region-text annotations, which serves as the foundation for training our CLOC model.
- Through extensive experiments across 31 evaluation tasks, including standard image-text tasks, newly constructed region-text tasks, and downstream evaluations with MLLMs, we demonstrate that CLOC significantly and consistently outperforms the CLIP counterpart.
- We are working on releasing our pre-trained checkpoints and the constructed region-text annotations along with the final version to accelerate future research.

2. Related Work

Vision encoder pre-training. A popular approach to MLLMs such as LLaVA (Liu et al., 2023), typically connects a vision encoder (*e.g.*, ViT (Dosovitskiy et al., 2021)) to digest visual input and maps them to the input space of the LLM as tokens. Among various types of vision encoders (Oquab et al., 2023; He et al., 2022), language-

supervised methods like CLIP (Radford et al., 2021; Jia et al., 2021) becomes the popular choice, due to its superior performance on MLLMs reported (Tong et al., 2024).

Other training approaches like captioning loss (Tschannen et al., 2024) are also popular; Wan et al. (2024) further incorporates bounding box coordinates. However, they need to train a full encoder-decoder transformer with smaller batch sizes, which is less efficient than CLIP. Also, the vision embeddings are not directly aligned with languages, so more limited for search or retrieval tasks. *Therefore, our scope focuses on improving the CLIP approach.*

Improving localization of CLIP. Since CLIP was introduced, many follow-up works have been proposed to improve it from various aspects, for different target tasks, and with different approaches. From the aspect relevant to our work, improving the localization capability, most works specifically focus on the downstream dense vision tasks such as open-vocabulary detection (Minderer et al., 2024; Yao et al., 2022; Wu et al., 2023). Another, and arguably more challenging, direction is to maintain the generalizability of CLIP on image-level tasks while improving localization. Recent works (Naeem et al., 2023; Bica et al., 2024; Dong et al., 2023) combine localization-enhancing unsupervised objectives with the CLIP loss, but do not attempt supervision on large-scale explicit pseudo-labeled data like ours and have more computational overhead. Alpha-CLIP (Sun et al., 2024) shows that the SAM (Kirillov et al., 2023) can provide useful conditions for CLIP. Another relevant works, UMG-CLIP (Shi et al., 2025), fine-tunes a pre-trained CLIP model for dense vision tasks on a 41M image/region-text dataset. Although there is a technical similarity to ours, the goals and positioning of our work and theirs are quite different. UMG-CLIP is designed primarily for vision-centric tasks such as detection and segmentation, whereas our focus is on MLLMs. MLLM VQA tasks require more extensive language understanding and thus demand large-scale pre-training data. Our approach pre-trains from scratch on up to 2B images thanks to our scalable labeling pipeline. Last but not least, we believe that a fine-tuning approach like UMG-CLIP is complementary to CLOC pretraining rather than conflicting with it.

Synthetic annotations for pre-training. The literature has been exploring scalable ways to generate high-quality synthetic annotations for CLIP. For instance, several works demonstrate that visually-enriched image captions improve CLIP (Lai et al., 2024). MOFI (Wu et al., 2024) augments CLIP with an extreme multi-classification task. However, these works only consider image-level annotations but not explicit region-level labels. In the context of dense vision tasks like open-vocabulary detection and segmentation, pseudo-labeling in a self-training paradigm has proven an effective approach (Kirillov et al., 2023; Minderer et al., 2024). We are inspired by these efforts and combine them

to enhance CLIP’s localization capabilities. Our approach is promising, since the advances of these labeling methods can further improve ours in the future.

3. CLOC

3.1. From Image-Text to Region-Text Alignment

CLIP (Radford et al., 2021) contrastively aligns the embedding from a pair of image and text encoders (f_I and f_T). Let a mini-batch of N image-text pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ be sampled from the large-scale training set during each training iteration. The contrastive loss is defined as follows:

$$\mathcal{L}_{\text{CLIP}} := (\mathcal{L}_{I \rightarrow T} + \mathcal{L}_{T \rightarrow I})/2. \quad (1)$$

$$\mathcal{L}_{I \rightarrow T} := -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(f_I(\mathbf{x}_i), f_T(\mathbf{y}_i))/\tau)}{\sum_{j=1}^N \exp(\text{sim}(f_I(\mathbf{x}_i), f_T(\mathbf{y}_j))/\tau)},$$

where $\text{sim}(\cdot, \cdot)$ is the similarity function and τ is a (learnable) temperature. The CLIP loss $\mathcal{L}_{\text{CLIP}}$ averages the symmetrical contrastive loss in which cross-entropy normalized along image-to-text and text-to-image axes, respectively.

Conceptually, $\mathcal{L}_{\text{CLIP}}$ aligns images with their associated text, but it overlooks subimage semantics. We propose augmenting this with *region-text* alignment on top of $\mathcal{L}_{\text{CLIP}}$. Specifically, assume an image-text pair (\mathbf{x}, \mathbf{y}) can be decomposed into image regions $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}$, and there exist regional captions $\mathbf{y}'^{(m)}$ that describe the corresponding regions $\mathbf{x}^{(m)}$. Thus, the original input (\mathbf{x}, \mathbf{y}) becomes region-text pairs $\{(\mathbf{x}^{(1)}, \mathbf{y}'^{(1)}), \dots, (\mathbf{x}^{(m)}, \mathbf{y}'^{(m)})\}$, and (\mathbf{x}, \mathbf{y}) is a special case when the “region” itself is the whole image. We identify several research questions and will answer them in the following sections:

1. Considering the goal is to train an image encoder f_I with enhanced localization capability, how should we formulate a region-text alignment goal that improves f_I ? We propose a novel learning task called *promptable embeddings* in Section 3.2.
2. How to properly extract region embedding from $f_I(\mathbf{x})$ as an effective joint design? We propose a lightweight promptable region extractor in Section 3.3.
3. How to generate meaningful image regions with high-quality captions? Furthermore, in many cases, the ideal region caption $\mathbf{y}'^{(m)}$ may not exist in the image-level caption, i.e., $\mathbf{y}'^{(m)}$ might not be a substring of the original \mathbf{y} . We design an effective and scalable data engine as a visually-enriched and spatially-localized labeler to generate high-quality region-text pairs in Section 4.
4. With the above considerations, we discuss how to train the model with minimal conflicts towards a drop-in replacement of the CLIP model in Section 3.4.

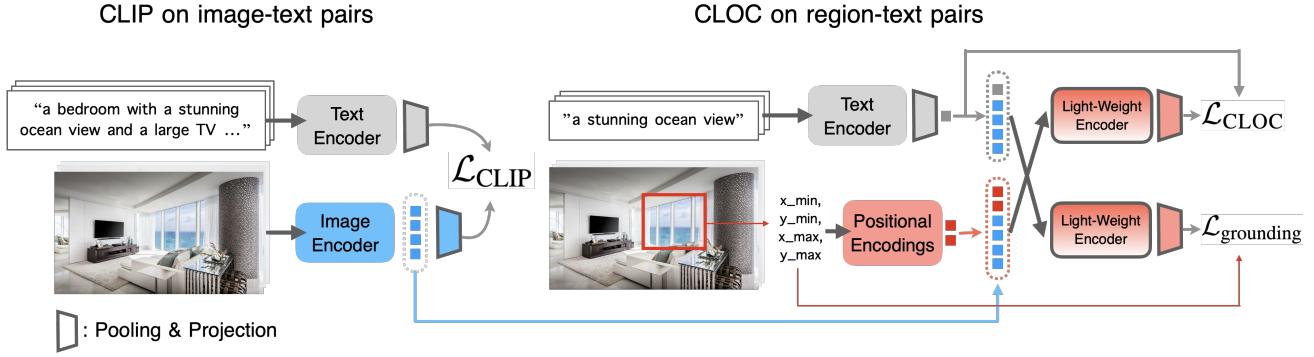


Figure 2. CLOC promptable embedding architecture. CLOC builds upon the image embedding from CLIP (before pooling and projection) and transforms it into a region-aware vision embedding given an encoded prompt; *e.g.*, positional encodings of box coordinates or regional caption encoded by the CLIP text encoder.

3.2. Promptable Embeddings

To optimize CLIP with better feature localization and eventually learn an enhanced CLIP vision encoder f_I for various VL downstream tasks, we argue that it will require at least two capabilities. (*i*) First, the encoder should recognize fine-grained small objects (*e.g.*, this image crop is an “airplane wheel”). (*ii*) Second, the *image embedding* produced by the encoder provides a holistic understanding such that an MLLM can reason more advanced spatial hierarchy relationships within the scene (*e.g.*, “The plane is lowering its front landing gear.”). As discussed in Section 2, many previous works improve CLIP toward object detection tasks thus mainly focusing on (*i*) only; *e.g.*, RegionCLIP (Zhong et al., 2022) that crops out image regions and uses them as additional input images to re-train the CLIP encoders for recognizing objects. However, to support comprehensive VL tasks, (*i*) is necessary but insufficient without (*ii*).

To achieve this, we introduce a new concept, *promptable embedding*. We consider a scenario similar to MLLM use cases, where answers are generated using CLIP image tokens alongside a question. We hypothesize that a strong encoder for MLLMs should produce an *image embedding* that can *easily be transformed into region representations, given location cues*.

We re-formulate the CLIP loss based on image-text pairs (\mathbf{x}, \mathbf{y}) into a **localized** language-image contrastive loss for region-text alignment based on triplets of $(\{\mathbf{l}\}, \mathbf{x}, \mathbf{y})$, where \mathbf{l} is a location representation such as a bounding box, and possibly there are several boxes as a set $\{\mathbf{l}\}$ per image. To make it compatible with CLIP training, we construct a promptable embedding transform module, or in short, *region prompter* $\mathbf{z} = \text{Prompter}(\mathbf{l}, f_I(\mathbf{x}))$, that extracts the region embedding specified by \mathbf{l} from the image embedding $f_I(\mathbf{x})$. This formulation is inspired by the success of the segmentation model SAM (Kirillov et al., 2023) which predicts *segmentation masks* conditioned on location prompt

(*e.g.*, a box), while CLOC predicts a *region embedding* conditioned on \mathbf{l} instead.

To this end, we decompose the location-image-text triplets as localized region-text pairs. Let $\mathbf{z}_i^{(m)} = \text{Prompter}(\mathbf{l}_i^{(m)}, f_I(\mathbf{x}_i))$ and $\mathbf{y}_i^{(m)}$ is the caption of the region specified by $\mathbf{l}_i^{(m)}$. $\mathbf{l}_i^{(m)} \in \mathbb{R}^4$ is the m -th box of image i represented as two coordinates (*i.e.*, top-left and bottom-right corners). We then formulate a symmetric region-text contrastive loss similar to Equation 1:

$$\begin{aligned} \mathcal{L}_{R \rightarrow T} := & -\frac{1}{MN} \sum_{i=1}^N \sum_{\mathbf{l}_i^{(m)} \sim \{\mathbf{l}_i\}} \log \mathbf{p} \\ \mathbf{p} = & \frac{\exp(\text{sim}(\mathbf{z}_i^{(m)}, f_T(\mathbf{y}_i^{(m)}))/\tau)}{\sum_{j=1}^N \sum_{\mathbf{l}_j^{(m')} \sim \{\mathbf{l}_j\}} \exp(\text{sim}(\mathbf{z}_i^{(m)}, f_T(\mathbf{y}_j^{(m')}))/\tau)}, \end{aligned} \quad (2)$$

where M is the number of regions $\mathbf{l}_i^{(m)}$ sampled randomly per image. We set $M = 4$ by default. We will discuss implementing the *Prompter* in Section 3.3, and generating $\mathbf{l}_i^{(m)}$ with $\mathbf{y}_i^{(m)}$ in Section 4. $\mathcal{L}_{T \rightarrow R}$ is the symmetric contrastive loss normalized along text-to-region axis, just like in Equation 1. Intuitively, for each region (m) of image i , it contrasts with all the other region captions \mathbf{y} of other regions (m') from the same (when $i = j$) or other images (when $i \neq j$). We define $\mathcal{L}_{\text{CLOC}} = (\mathcal{L}_{R \rightarrow T} + \mathcal{L}_{T \rightarrow R})/2$.

As the *Prompter* is a simple transformer encoder, it allows flexible types of prompts besides bounding boxes we have used, such as points, free-form referring, text, and etc. We further consider the case where the prompt is free-form text, and leave others for future study. We add a grounding loss that extracts a region feature from the image (*e.g.*, a picture of the bedroom) given its regional caption (*e.g.*, “a large TV”), and predicts the bounding box with an MLP regression head, *i.e.*,

$$\mathcal{L}_{\text{grounding}} := \frac{1}{4MN} \sum_{i=1}^N \sum_{\mathbf{l}_i^{(m)} \sim \{\mathbf{l}_i\}} \|\mathbf{l}_i^{(m)} - \text{Head}(\mathbf{z}(\mathbf{y}_i^{(m)}))\|_2, \quad (3)$$

where $\mathbf{z}(\mathbf{y}) := \text{Prompter}(f_T(\mathbf{y}), f_I(\mathbf{x}))$ is the grounded embedding conditioned on the text (encoded by the CLIP text encoder). All the learnable parameters are trained end-to-end. With a scalar λ , the overall loss is

$$\mathcal{L} := \mathcal{L}_{\text{CLIP}} + \lambda(\mathcal{L}_{\text{CLOC}} + \mathcal{L}_{\text{grounding}}). \quad (4)$$

3.3. CLOC Model Architecture

We implement the promptable embedding introduced in Section 3.2 with minimal extra modules on top of the original CLIP. As illustrated in Figure 2, the CLIP model remains the same for computing $\mathcal{L}_{\text{CLIP}}$. For $\mathcal{L}_{\text{CLOC}}/\mathcal{L}_{\text{grounding}}$, the image embedding is re-used from the CLIP ViT but before the pooled projection and normalization f'_I . To extract the region embedding $\mathbf{z} = \text{Prompter}(\mathbf{l}, f'_I(\mathbf{x}))$ from the image, we consider the location representation \mathbf{l} as two coordinates (top-left and bottom-right corners of a box), each vectorized by positional encoding. The `Prompter` is a lightweight one-layer transformer encoder. It takes the positional encodings prepended with the image tokens from ViT together as the input, and outputs the region embedding with a pooled projection layer. For the grounding loss, we re-use the same CLIP text encoder for encoding the region captions $\mathbf{z} = \text{Prompter}(f_T(\mathbf{y}), f'_I(\mathbf{x}))$ to predict the bounding boxes with a two-layer MLP head. Overall, CLOC only adds lightweight additional parameters of the heads. Note that, the main overheads during forward are from ViT image encoding – CLOC reuses it for multiple prompts.

3.4. Discussions on Design Choices and Extensions

We provide discussions here on the rationale behind our design choices and some minor extensions.

Extracting region embedding with visual prompts. Training with $\mathcal{L}_{\text{CLOC}}$ in Equation 4 requires extracting region embeddings from the image features given the bounding boxes. Another alternative could be Region-of-Interest (RoI) pooling/alignment (He et al., 2017) from the spatial image feature of ViT before pooling. RoI operations are popular, especially in the object detection literature. However, as will be evidenced by worse performance in Section 5, we found spatial pooling an over-strong premise for CLOC pre-training here for several reasons.

First, unlike object detection datasets that typically contain golden labels, here the pseudo-labels are much noisier on the large-scale web-crawled images. The resulting RoI features may be inaccurate due to the imprecise bounding boxes, making model training less effective. Second, unlike dense vision tasks that directly rely on the spatial features, MLLM has a transformer decoder that consists of several attention layers such that the constraint of semantics in the spatial feature space becomes somewhat indirect.

Table 1. Region-text dataset statistics. We summarize the text token length for both images and regions. Partial statistics of the proprietary datasets revealed by their papers. *The 20M subset of GRIT is released; we removed the invalid URLs.

| Dataset | # of imgs | regions per img | img len. | region text len. |
|--|-----------|-----------------|----------|------------------|
| FlickrEntities (Plummer et al., 2015) | 32K | 8.7 | – | – |
| RefCOCO (Yu et al., 2016) | 20K | 2.5 | – | 3.6 |
| RefCOCO+ (Yu et al., 2016) | 20K | 2.5 | – | 3.5 |
| RefCOCOg (Mao et al., 2016) | 27K | 2.1 | – | 8.4 |
| Visual Genome (Krishna et al., 2017) | 108K | 38.0 | – | – |
| GRIT (prop.) (Peng et al., 2023) | 91M | 1.5 | – | 4.7 |
| GRIT (released) (Peng et al., 2023)* | 17M | 1.8 | 17.2 | 4.6 |
| Florence-2 (prop.) (Xiao et al., 2024) | 126M | 5.4 | 70.5 | 2.6 |
| OWLv2 (prop.) (Minderer et al., 2024) | 2B | – | – | – |
| Minderer et al. (2024) on WiT | 300M | 5.1 | 17.1 | 3.9 |
| VESL WiT (Ours) | 300M | 11.6 | 44.9 | 2.1 |
| VESL WiT+DFN (Ours) | 2B | 11.5 | 35.9 | 2.1 |

Table 2. Object hallucination in captions. We evaluate different captioners and the captions generated by our pipeline using the CHAIR score (Rohrbach et al., 2018); the lower the better

| Model | CHAIR _i | CHAIR _s |
|---------------------------------|--------------------|--------------------|
| InstructBLIP (Dai et al., 2023) | 14.5 | 30.0 |
| MiniGPT-4 (Zhu et al., 2023) | 8.2 | 24.2 |
| Shikra (Chen et al., 2023) | 7.0 | 22.0 |
| LLaVA-1.5 (Liu et al., 2023) | 6.2 | 20.6 |
| VESL (Ours) | 5.9 | 19.6 |

Our `Prompter` mimics such inductive bias in pre-training via a single-attention-layer encoder that may leverage better global context reasoning compared to RoIs.

Avoiding region-text conflicts. While region annotations introduce location information, a concern of contrastive learning may be similar objects within an image (*e.g.*, “boats” in the harbor) or a mini-batch. To mitigate it, we apply two tricks. First, fortunately, we found it sufficient to sample a few regions per image for each update, *e.g.*, we set $M = 4$ in Equation 2 in experiments. Second, we filter similar texts when computing the negatives in the contrastive loss. More specifically, these pairs of $(\mathbf{z}_i^{(m)}, f_T(\mathbf{y}_j^{(m')}))$ are “masked” and ignored in the denominators of both $\mathcal{L}_{R \rightarrow T/T \rightarrow R}$, if $\text{sim}(f_T(\mathbf{y}_i^{(m)}), f_T(\mathbf{y}_j^{(m')})) > 0.9$ (no gradients on f_T).

4. VESL Captioning Pipeline

As discussed in Section 1 and 3.1, a key bottleneck of CLOC is the region-text datasets in terms of both the data size and the label quality since there are no public datasets with region-text annotations at scales large enough for contrastive pre-training. Inspired by recent works that en-

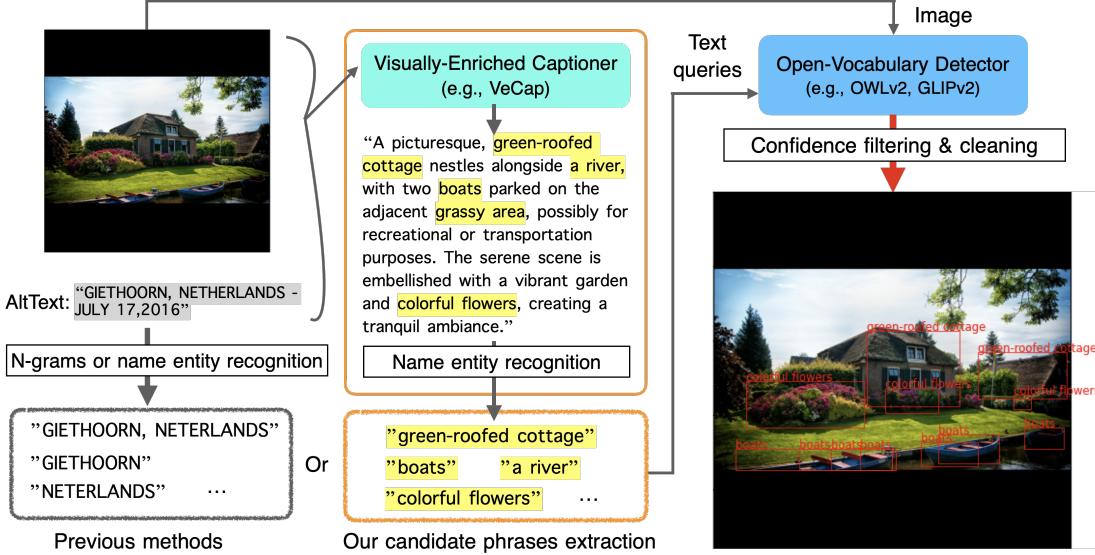


Figure 3. Our Visually-Enriched and Spatially-Localized (VESL) captioning pipeline. We leverage an existing open-vocabulary detector (e.g., OWLv2) that predicts bounding boxes on the images and assigns the labels from the given text phrase candidates. Previous methods often use the alt-text attached to the images, which is prone to insufficient region descriptions. We found it crucial to re-caption images with the visually-enriched captioner VeCap (Lai et al., 2024) for better visual concept exploitation for the detector.

rich image captions for better CLIP training, we make a step further for **Visually-Enriched and Spatially-Localized (VESL)** captioning that generates fine-grained captions at *region level*. The goal of VESL is, given an image with the original web-crawled alt-text, to annotate it with the grounded bounding boxes, each associated with a caption in natural language for optimizing Equation 2 in Section 3.2.

Concretely, VESL is a pseudo-labeling pipeline with the following steps, with pseudo codes in Appendix C:

- 1. Re-captioning with visual concept exploitation:** We follow the VeCap2 (Lai et al., 2024; 2025) to generate long, diverse, and detailed image captions.
- 2. Region phrase candidates extraction:** We apply name entity recognition (NER) to extract phrases from the visually-enriched captions as candidates describing a region inside the image, inspired by Zhang et al. (2022).
- 3. Open-vocabulary detection with extracted phrases:** the final region-text annotations are generated by a pre-trained open-vocabulary detector. It matches the phrases extracted from Step 2 to the bounding boxes proposed by the detector. We adopted OWLv2 detector (Minderer et al., 2024) which combines the CLIP image/text encoders with detection heads. The boxes with confidence larger than 0.1 are kept as the region location and the most confident phrases are considered as their captions.

Remarks. We highlight our insights behind the proposed recipe. The most relevant work was proposed in (Minderer et al., 2024) that scales up open-vocabulary (OV) detection

via self-training. We are inspired by its success and extend it to CLOC contrastive learning with important modifications. Different from (Minderer et al., 2024) that generates candidate phrases from the n -grams of the web-crawled alt-text of the images for OV detection, we found the alt-text might not have enough details describing the image region content, thus limiting the diversity and quality of the annotations predicted by the OV detector. We thus caption each image augmented with more visual details. However, the long captions make the n -grams candidates verbose and grow exponentially, thus we generate high-quality candidates via name entity recognition instead. We found that such a pipeline produces training data more suitable for CLOC, as will be validated in Section 5. We believe our approach is promising, as it could enjoy future improvements from better building blocks we rely on such as re-captioning (Li et al., 2024; Fan et al., 2024) and open-vocabulary detection (Minderer et al., 2024) models.

Our pre-training datasets. Our pre-training data consists of two parts: (*i*) image-text pairs, and (*ii*) region-text pairs. For image-text pairs, we reproduce the image re-captioning pipeline from VeCap (Lai et al., 2024), and generate synthetic captions for WiT-300M (Wu et al., 2024) and DFN-5B (Fang et al., 2023) images. For region-text pairs, we pseudo-label WiT-300M and a 2B-image subset of DFN-5B using our VESL pipeline. In VESL, we adopted the official OWLv2 L/14 model (Minderer et al., 2024) as the open-vocabulary detector. All images are pseudo-labeled with 448×448 resolution, where a maximum number of 20 phrase queries are sampled for moderate computation budget. Table 1 summarizes the statistics of existing region-

text datasets and ours. Notably, we also ablate annotating ViT-300M following (Minderer et al., 2024) and found it detects *less* objects with longer region text, likely due to the verbose n -grams of alt-text are in lower quality than our approach, as discussed in the remarks.

Although synthetic captions are generated as detailed descriptions, one may wonder if hallucinations would be introduced into the captions. As VESL grounded the objects into the captions, we expect that hallucinations are mitigated. In Table 2, we evaluate the hallucination metric, the CHAIR score (Rohrbach et al., 2018), where lower is better, to assess the quality of synthetic captions. As shown in the table, our captioner demonstrates high quality compared to other models.

5. Experiments

5.1. Setup (more details in Appendix B)

Pre-training. We follow OpenAI-CLIP (Radford et al., 2021) to train both our CLIP baseline and CLOC model using a similar budget of around 14B images seen. Many previous models are trained on quite different data, labels, training cost, architectures, and etc. This makes it hard to draw a fair comparison directly, and some of them are not open-sourced. For a faithful comparison of our proposed methodology, we use the same hyper-parameters and image datasets for both the CLIP baseline and CLOC. We experimented with the ViT B/16 and L/14 architectures, pre-trained with 224×224 and 336×336 image resolutions, respectively. All parameters are trained end-to-end from scratch. We implement the above in JAX (Bradbury et al., 2018).

Evaluation tasks. The image encoders are evaluated across a wide range of downstream tasks. First, we assess performance on ImageNet image classification (Deng et al., 2009; Shankar et al., 2020) and COCO retrieval (Lin et al., 2014). Second, we construct region-level tasks, including COCO object recognition and region-text retrieval using the GRIT dataset (Peng et al., 2023). Furthermore, we show CLOC is particularly useful for MLLMs, validated by the Ferret model (You et al., 2023) which requires fine-grained image understanding for referring and grounding tasks. We also evaluate on general multimodal benchmarks using LLaVA-1.5 (Liu et al., 2023) and Open-LLaVA-NeXT (Liu et al., 2024; Chen & Xing, 2024), which both use the 7B Vicuna LLM. For all evaluation tasks, we use the same official hyper-parameters, fine-tuning datasets, and codebase for all the image encoders we experimented with, without specific tuning.

5.2. Image and Region Classification and Retrieval

CLOC encoder produces not only image embedding but also region embeddings. It can be used directly for region-level tasks without further training, in analogy to the zero-shot capability of CLIP on images. This emergent capability enables us to construct region-level zero-shot tasks for fast development and ablation studies.

In addition to *image*-level evaluation like ImageNet classification and COCO image-text retrieval, we additionally construct *region*-level tasks, including region object recognition and text retrieval. More specifically, the region-level tasks leverage the labeled bounding boxes in the evaluation set for CLOC to extract region embedding. For region retrieval, we use a validation set of the GRIT dataset (Peng et al., 2023) and encode both the image regions and the region captions. For region classification, the class names are encoded as text embedding (80 / 1203 classes for COCO / LVIS, respectively), and the highest cosine similarity for each region embedding is predicted as its class. We highlight important variables for the performance in Table 3 with the following observations:

- CLOC performs decently on region-level tasks¹ with strong image-level performance (② vs. ⑧ ⑯).
- The `Prompter` is an important ingredient for CLOC’s success to go beyond CLIP (③ ⑧ ⑯ vs. ④ ⑨ ⑭). We replace the `Prompter` with RoI alignment to extract region features and train with $\mathcal{L}_{\text{CLOC}}$ (similar to (Shi et al., 2025)). We found it performs much worse on region-level tasks, possibly due to difficulties of strong RoI constraints and the noisy labels as discussed in Section 3.4.
- VESL helps, as the visually-enriched captions improve image retrieval tasks (as expected (Lai et al., 2024)) and the versatile visual concepts candidates facilitate the OV detector, supporting Section 4 (③ ⑯ vs. ⑤ ⑮).
- OV detector OWLv2 > GLIPv2 in VESL (③ vs. ⑥).
- Tricks in Section 3.4 offer slight performance gains, but $\mathcal{L}_{\text{CLOC}}$ is already highly effective on its own (⑩ ⑪).
- Practically, sampling 2/4 boxes works (⑫) well already.
- Scaling up images saturated on region tasks but further improved on MLLM tasks (③ ⑧); Table 4.
- Scaling up the ViT model sizes can further improve both image and region tasks (⑯ ⑯ ⑰).

Overall, CLOC not only achieves strong performance on image-level tasks, but unlocks zero-shot region-level capability. Below, with our design choices validated, ⑯ will be used by default if not specified.

¹For reference, in a different setup, Wu et al. (2023) reports 46.5% mAcc on the COCO region classification task, trained with 320×320 COCO images directly. In contrast, our approach achieves over 70% mAcc, pre-trained on a 224×224 web-crawled dataset with our object labels (thus not a fair comparisons).

Table 3. Zero-shot evaluation on image-level tasks (recall@1 of COCO retrieval and accuracy of ImageNet (IN) classification) and region-level tasks (recall@10 of GRIT region retrieval and mAcc of region object recognition on COCO and LVIS), using ViT-B/16 as the default encoder backbone. The indentation with different symbols denotes removing (–) or changing a component (○).

| Models | Training Labels | | Image tasks | | | | Region tasks | | | | Avg. | |
|--------------------------------------|-----------------|---------|-------------|----------|------|------|--------------|----------|---------|---------|-------|--------|
| | Image | Region | COCO-i2t | COCO-t2i | INV1 | INV2 | GRIT-r2t | GRIT-t2r | RecCOCO | RecLVIS | Image | Region |
| ① OpenAI-CLIP | proprietary | - | 52.4 | 33.1 | 68.3 | 62.3 | - | - | - | - | 54.0 | - |
| ② CLIP | WiT+DFN | - | 66.3 | 45.1 | 76.2 | 69.6 | - | - | - | - | 64.3 | - |
| ③ CLOC | WiT | WiT | 68.8 | 50.1 | 66.7 | 59.7 | 65.1 | 67.2 | 70.6 | 26.7 | 61.3 | 57.4 |
| ④ – Prompt \circ | WiT | WiT | 67.0 | 49.7 | 65.6 | 58.6 | 44.8 | 4.4 | 55.3 | 13.2 | 60.2 | 29.4 |
| ⑤ – VESL | WiT | WiT | 53.9 | 36.3 | 66.6 | 59.5 | 71.5 | 63.8 | 62.2 | 22.2 | 54.1 | 54.9 |
| ⑥ ○ w/ GLIPv2 | WiT | WiT | 68.8 | 50.0 | 65.8 | 59.2 | 67.9 | 71.1 | 64.9 | 23.1 | 61.0 | 56.8 |
| ⑧ CLOC | WiT+DFN | WiT | 66.1 | 46.5 | 75.5 | 68.6 | 65.8 | 67.4 | 70.1 | 27.2 | 64.2 | 57.6 |
| ⑨ – Prompt \circ | WiT+DFN | WiT | 65.8 | 46.5 | 75.7 | 68.0 | 55.5 | 18.4 | 67.1 | 24.6 | 64.0 | 41.4 |
| ⑩ – text filtering | WiT+DFN | WiT | 65.4 | 46.0 | 75.7 | 68.4 | 66.3 | 66.5 | 68.7 | 24.8 | 63.9 | 56.6 |
| ⑪ – $\mathcal{L}_{\text{grounding}}$ | WiT+DFN | WiT | 66.0 | 46.3 | 75.7 | 67.9 | 66.0 | 66.8 | 70.0 | 25.8 | 64.0 | 57.2 |
| ⑫ ○ $M = 2$ | WiT+DFN | WiT | 66.6 | 46.2 | 75.5 | 67.9 | 66.5 | 67.0 | 69.8 | 25.8 | 64.1 | 57.3 |
| ⑯ CLOC | WiT+DFN | WiT+DFN | 69.2 | 49.3 | 74.9 | 67.0 | 63.9 | 65.9 | 71.1 | 28.5 | 65.1 | 57.3 |
| ⑰ – Prompt \circ | WiT+DFN | WiT+DFN | 70.2 | 49.7 | 74.7 | 67.6 | 65.7 | 23.0 | 67.1 | 25.4 | 65.6 | 45.3 |
| ⑯ – VESL | WiT+DFN | WiT+DFN | 65.3 | 46.6 | 75.5 | 67.7 | 55.7 | 22.3 | 66.3 | 25.3 | 63.8 | 42.4 |
| ⑯ ○ ViT L/14 | WiT+DFN | WiT+DFN | 74.8 | 54.4 | 80.1 | 73.2 | 66.9 | 68.3 | 72.9 | 32.6 | 70.6 | 60.2 |
| ⑰ ○ ViT H/14 | WiT+DFN | WiT+DFN | 75.7 | 55.1 | 81.3 | 74.7 | 67.4 | 69.4 | 73.0 | 35.6 | 71.7 | 61.3 |

Table 4. Ferret-Bench for referring and grounding VQA, based on Ferret (You et al., 2023) equipped with different image encoders. Models are evaluated with OpenAI gpt-4○ API. *replace Ferret visual sampler with Prompt \circ ; Details in Section 5.3.

| Method | ViT | Region Align | #images w/ region labels | Ref. Descript. | Ref. Reason. | Ground Conv. | Avg. Δ_{CLIP} |
|----------|------|----------------|--------------------------|----------------|--------------|--------------|-----------------------------|
| CLIP | B/16 | None | None | 47.5 | 50.3 | 45.3 | 47.7 |
| CLOC | B/16 | RoI-Align | 300M | 48.0 | 48.4 | 40.0 | 45.5 |
| CLOC | B/16 | Prompt \circ | 300M | 50.2 | 55.5 | 41.5 | 49.1 |
| CLOC | B/16 | Prompt \circ | 2B | 53.6 | 53.7 | 42.2 | 49.8 (4.2) |
| CLOC * | B/16 | Prompt \circ | 2B | 54.8 | 54.9 | 44.7 | 51.5 (3.7) |
| OAI-CLIP | L/14 | None | None | 50.8 | 55.4 | 45.7 | 50.6 |
| CLIP | L/14 | None | None | 54.2 | 54.6 | 43.3 | 50.7 |
| CLOC | L/14 | Prompt \circ | 300M | 51.0 | 65.7 | 44.9 | 53.9 |
| CLOC | L/14 | Prompt \circ | 2B | 55.9 | 63.3 | 46.0 | 55.1 (4.4) |
| CLOC * | L/14 | Prompt \circ | 2B | 56.3 | 67.4 | 47.1 | 56.9 (6.2) |

5.3. Referring and Grounding with Ferret

As discussed in Section 1, a key motivation is to provide an enhanced image encoder for training MLLMs, particularly for tasks requiring fine-grained image understanding. A notable example is Ferret (You et al., 2023), a recently proposed MLLM that builds on LLaVA and aims to handle more advanced spatial interactions, such as referring and grounding in VQA tasks. Ferret can take region prompts such as a box, a point, or a free-form location referring to the input image as input, and answer a question specific to the region such as “Do you know when the object[region] was invented?” Ferret thus requires fine-grained image features from the vision encoder for spatial reasoning.

We evaluate CLOC by replacing the CLIP ViT encoder

with our CLOC ViT as a drop-in replacement. We use the official codebase for training the Ferret model. We further consider a variant based on Ferret: the Ferret model implements a spatial-aware visual sampler that samples image features from the region specified in the question. We replace the sophisticated visual sampler with our simple Prompt \circ introduced in Section 3.3 to extract region embedding with $z = \text{Prompt}(\mathbf{l}, f'_I(\mathbf{x}))$ instead, as illustrated in Figure 1(right).

In Table 4, we evaluate different pre-trained image encoders on the Ferret-Bench benchmark (You et al., 2023). Ferret-Bench includes challenging multimodal dialogue-style VQA of three tasks constructed with GPT-4. Results show that our Prompt \circ is essential to improve upon the CLIP baseline – RoI-Align may even slightly degrade performance. Scaling region labels from 300M to 2B further improves performance. Interestingly, our Prompt \circ (denoted as *) can be a replacement of the FERRET visual sampler in fine-tuning, which is simpler and performs even better up to 6% against both the OpenAI-CLIP and our in-house CLIP. We also evaluate CLOC (2B labeled) on other referring and grounding tasks ranging from referring object classification, referring expression comprehension, and phrase grounding across multiple datasets. As summarized in Table 5, CLOC is also superior evidenced by 1 ~ 3% improvements in average of 13 evaluation sets.

5.4. General VQA with LLaVA-1.5 and LLaVA-NeXT

We further show that the CLOC encoder is also competitive against CLIP on general VQA tasks without regression

Table 5. Results on referred-LVIS object classification, referring expression comprehension (0.5 IoU on RefCOCO, RefCOCO+, RefCOCOg), and phrase grounding (0.5 IoU on Flickr30k Entities). FERRET *: replace visual sampler in FERRET with CLOC prompter.

| Model | Encoder | RefLVIS box point free- form | RefCOCO | | | RefCOCO+ | | | RefCOCOg | | | Flickr | | | Avg. (Δ to CLIP) |
|-----------------------------|------------------|---------------------------------------|----------------|----------------|---------------------|-----------------|-------|-------|----------|------|-----|--------|------|--|-----------------------------|
| | | | val | testA | testB | val | testA | testB | val | test | val | val | test | | |
| FERRET | CLIP B/16 | 72.5 56.9 57.2 | 80.7 84.2 77.1 | 71.9 76.1 63.7 | 75.9 76.2 76.2 78.3 | 72.8 | | | | | | | | | |
| FERRET | CLOC B/16 | 74.3 56.7 60.2 | 84.2 87.0 80.0 | 74.7 80.0 67.0 | 78.8 79.5 80.0 81.5 | 75.7 ($+2.9$) | | | | | | | | | |
| FERRET * | CLOC B/16 | 78.9 58.2 61.4 | 84.4 86.8 78.9 | 74.0 78.7 65.5 | 78.0 78.7 80.1 81.4 | 75.8 ($+3.0$) | | | | | | | | | |
| Shikra (Chen et al., 2023) | OpenAI-CLIP L/14 | 57.8 67.7 n/a | 87.0 90.6 80.2 | 81.6 87.4 72.1 | 82.3 82.2 75.8 76.5 | - | | | | | | | | | |
| FERRET (Zhang et al., 2024) | OpenAI-CLIP L/14 | 79.4 67.9 69.8 | 87.5 91.4 82.5 | 80.8 87.4 73.1 | 83.9 84.8 80.4 82.2 | 80.8 | | | | | | | | | |
| FERRET | CLIP L/14 | 78.7 66.9 70.2 | 88.0 90.4 83.5 | 80.1 85.8 73.3 | 82.8 83.4 79.0 80.1 | 80.2 | | | | | | | | | |
| FERRET | CLOC L/14 | 81.6 67.9 69.9 | 89.0 91.0 84.7 | 81.4 86.8 74.7 | 84.0 85.2 82.3 83.3 | 81.7 ($+1.5$) | | | | | | | | | |
| FERRET * | CLOC L/14 | 79.8 67.9 69.1 | 88.2 91.1 84.5 | 80.6 86.7 73.9 | 84.8 85.1 82.4 83.5 | 81.4 ($+1.2$) | | | | | | | | | |

Table 6. Results on multimodal benchmarks using LLaVA-1.5 and Open-LLaVA-NeXT with ViT-L/14 and Vicuna-7B.

| Method | LLaVA ^W | TextVQA | GQA | MMVet | POPE | MMEp | MMEc |
|-----------------|--------------------|---------|------|-------|------|--------|-------|
| LLaVA-1.5 | | | | | | | |
| CLIP | 59.3 | 53.3 | 62.2 | 30.0 | 86.7 | 1451.4 | 254.3 |
| CLOC | 64.3 | 54.9 | 62.7 | 31.5 | 87.3 | 1482.0 | 288.9 |
| Open-LLaVA-NeXT | | | | | | | |
| CLIP | 67.3 | 61.4 | 63.5 | 38.5 | 87.9 | 1486.1 | 279.6 |
| CLOC | 69.5 | 61.9 | 64.2 | 40.2 | 88.3 | 1451.1 | 312.5 |

and can even provide performance improvements. We use the Vicuna 7B LLM decoder for two experiments based on LLaVA-1.5 (frozen encoder) and Open-LLaVA-NeXT (unfrozen encoder with AnyRes (Liu et al., 2024) inputs). Since general VQA does not provide spatial referring inputs, we simply replace the ViT in LLaVA. Table 6 summarizes the results. Encouragingly, with our CLOC designs, the improved region-level alignment is also beneficial to some general multimodal benchmarks, as they may also require fine-grained image understanding.

5.5. Open-Vocabulary Detection

In Table 3, our models demonstrate that on the region classification task (predicting class names given a bounding box), our approach achieves over 70% mAcc in COCO, significantly outperforming 47% reported in previous work (Wu et al., 2023).

As introduced in subsection 3.2, CLOC consists of a region regression head and a classification head. Here we further benchmark CLOC as an open-vocabulary detection model. We provide zero-shot evaluation results in Table 7 on COCO Detection (minival), ODinW (test-dev), and LVIS-Det (minival). When comparing GLIP (Li et al., 2022) and CLOC, we observe that CLOC consistently achieves better results than GLIP across all datasets across

Table 7. Results on open-vocabulary detection.

| Method | ViT | COCO-Det (minival) | ODinW (test) | LVIS-Det (minival) |
|--------|------|-----------------------|-----------------|-----------------------|
| GLIP-T | T/16 | 46.6 | 46.5 | 26.0 |
| GLIP-L | L/14 | 49.8 | 52.1 | 37.3 |
| CLOC-B | B/16 | 47.3 | 48.4 | 29.6 |
| CLOC-L | L/14 | 50.8 | 53.6 | 38.1 |

different ViT backbones, suggesting that CLOC offers advantages in localization and object detection performance. Notably, GLIP employs DyHead (Dai et al., 2021), a strong detector head module, on top of the encoder, whereas our ablation study uses only two simple heads for classification and regression. This further supports that the encoder representation in CLOC is indeed superior.

6. Conclusion

Please see Appendix D for more discussions where we comment on the limitations, future directions, computation cost, design rationales, etc.

We tackle a deficiency of CLIP, to make the semantics aligned in the vision space for both image and region level. We propose a pre-training framework that enables a strong encoder to adapt seamlessly for downstream use in MLLMs, supporting feature interaction via input prompts. Our encoder creates a new possibility for adapting the features with input prompts of interaction together with MLLMs. To resolve the need for large-scale region-text training data, we carefully design a pseudo-labeling pipeline for visually-enriched and spatially-localized captions. Our pre-trained encoder is essentially a drop-in replacement of CLIP with competitive image-text performance and extra capability demonstrated in region-text tasks and VQA tasks with MLLMs.

Impact Statement

The goal of our work is to improve the performance of the well established CLIP pre-training. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgment

We thank Yanghao Li, Felix Bai, and many others for their invaluable help and feedback.

References

- Bica, I., Ilić, A., Bauer, M., Erdogan, G., Bošnjak, M., Kaplanis, C., Gritsenko, A. A., Minderer, M., Blundell, C., Pascanu, R., et al. Improving fine-grained understanding in image-text pre-training. *arXiv preprint arXiv:2401.09865*, 2024.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/jax-ml/jax>.
- Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., and Zhao, R. Shikra: Unleashing multimodal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.
- Chen, L. and Xing, L. Open-llava-next: An open-source implementation of llava-next series for facilitating the large multi-modal model community. <https://github.com/xiaoachen98/Open-LLAVA-NeXT>, 2024.
- Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructclip: Towards general-purpose vision-language models with instruction tuning, 2023. URL <https://arxiv.org/abs/2305.06500>.
- Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., and Zhang, L. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7373–7382, 2021.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Dong, X., Bao, J., Zheng, Y., Zhang, T., Chen, D., Yang, H., Zeng, M., Zhang, W., Yuan, L., Chen, D., et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10995–11005, 2023.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- Fan, L., Krishnan, D., Isola, P., Katabi, D., and Tian, Y. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36, 2024.
- Fang, A., Jose, A. M., Jain, A., Schmidt, L., Toshev, A., and Shankar, V. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.
- Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.
- Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2017.
- Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., and Jia, J. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2023.
- Lai, Z., Zhang, H., Wu, W., Bai, H., Timofeev, A., Du, X., Gan, Z., Shan, J., Chuah, C.-N., Yang, Y., et al. Veclip: Improving clip training via visual-enriched captions. In *ECCV*, 2024.

- Lai, Z., Saveris, V., Chen, C., Chen, H.-Y., Zhang, H., Zhang, B., Tebar, J. L., Hu, W., Gan, Z., Grasch, P., et al. Revisit large-scale image-caption data in pre-training multimodal foundation models. In *ICLR*, 2025.
- Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10975, 2022.
- Li, X., Tu, H., Hui, M., Wang, Z., Zhao, B., Xiao, J., Ren, S., Mei, J., Liu, Q., Zheng, H., et al. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.08478*, 2024.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.
- Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *NeurIPS*, 2023.
- Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL <https://llava-vl.github.io/blog/2024-01-30-llava-next/>.
- Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.
- McKinzie, B., Gan, Z., Fauconnier, J.-P., Dodge, S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng, F., Weers, F., et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. In *ECCV*, 2024.
- Minderer, M., Gritsenko, A., and Houlsby, N. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36, 2024.
- Naeem, M. F., Xian, Y., Zhai, X., Hoyer, L., Van Gool, L., and Tombari, F. Silc: Improving vision language pretraining with self-distillation. *arXiv preprint arXiv:2310.13355*, 2023.
- Oquab, M., Dariseti, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., and Wei, F. Kosmos-2: Grounding multimodal large language models to the world. *ArXiv*, abs/2306.14824, 2023.
- Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R. M., Xing, E., Yang, M.-H., and Khan, F. S. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13009–13018, June 2024.
- Ren, Z., Huang, Z., Wei, Y., Zhao, Y., Fu, D., Feng, J., and Jin, X. Pixellm: Pixel reasoning with large multimodal model. In *CVPR*, 2024.
- Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., and Saenko, K. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018.
- Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., and Schmidt, L. Evaluating machine accuracy on ImageNet. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8634–8644. PMLR, 13–18 Jul 2020.
- Shi, B., Zhao, P., Wang, Z., Zhang, Y., Wang, Y., Li, J., Dai, W., Zou, J., Xiong, H., Tian, Q., et al. Umg-clip: A unified multi-granularity vision generalist for open-world understanding. In *European Conference on Computer Vision*, pp. 259–277. Springer, 2025.
- Sun, Z., Fang, Y., Wu, T., Zhang, P., Zang, Y., Kong, S., Xiong, Y., Lin, D., and Wang, J. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13019–13029, June 2024.

- Tong, S., Brown, E., Wu, P., Woo, S., Middepogu, M., Akula, S. C., Yang, J., Yang, S., Iyer, A., Pan, X., et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.
- Tschannen, M., Kumar, M., Steiner, A., Zhai, X., Houlsby, N., and Beyer, L. Image captioners are scalable vision learners too. *Advances in Neural Information Processing Systems*, 36, 2024.
- Wan, B., Tschannen, M., Xian, Y., Pavetic, F., Alabdulmohsin, I., Wang, X., Pinto, A. S., Steiner, A., Beyer, L., and Zhai, X. Locca: Visual pretraining with location-aware captioners. *arXiv preprint arXiv:2403.19596*, 2024.
- Wu, S., Zhang, W., Xu, L., Jin, S., Li, X., Liu, W., and Loy, C. C. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. *arXiv preprint arXiv:2310.01403*, 2023.
- Wu, W., Timofeev, A., Chen, C., Zhang, B., Duan, K., Liu, S., Zheng, Y., Shlens, J., Du, X., Gan, Z., et al. Mofi: Learning image representations from noisy entity annotated images. In *ICLR*, 2024.
- Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., and Yuan, L. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4818–4829, 2024.
- Yao, L., Han, J., Wen, Y., Liang, X., Xu, D., Zhang, W., Li, Z., and Xu, H. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022.
- Yao, L., Han, J., Liang, X., Xu, D., Zhang, W., Li, Z., and Xu, H. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23497–23506, June 2023a.
- Yao, L., Han, J., Liang, X., Xu, D., Zhang, W., Li, Z., and Xu, H. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23497–23506, 2023b.
- You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.-F., and Yang, Y. Ferret: Refer and ground anything anywhere at any granularity. In *ICLR*, 2023.
- Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pp. 69–85. Springer, 2016.
- Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.
- Zhang, H., Zhang, P., Hu, X., Chen, Y.-C., Li, L., Dai, X., Wang, L., Yuan, L., Hwang, J.-N., and Gao, J. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022.
- Zhang, H., You, H., Dufter, P., Zhang, B., Chen, C., Chen, H.-Y., Fu, T.-J., Wang, W. Y., Chang, S.-F., Gan, Z., et al. Ferret-v2: An improved baseline for referring and grounding with large language models. In *Conference on Language Modeling*, 2024.
- Zhang, H., Gao, M., Gan, Z., Dufter, P., Wenzel, N., Huang, F., Shah, D., Du, X., Zhang, B., Li, Y., et al. Mm1.5: Methods, analysis & insights from multimodal llm fine-tuning. In *ICLR*, 2025.
- Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L. H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16793–16803, 2022.
- Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Appendix

A. Reproducibility Statement

We made our best efforts to exhaustively state the implementation details. Training hyper-parameters and model architectures are discussed in Section 3.2, 3.3, and 5.1, with a summary in Appendix B and Table A. For evaluation, as mentioned in Section 5.1, we strictly follow the official setup with the codebase released by the original authors if applicable, with details provided in Section B.2. For our datasets, we provide data processing details in Section 4 and example codes in Appendix C. We are working hard on releasing the annotations with internal approvals.

B. Experiment Details

We provide the omitted experiment details for pre-training and the downstream evaluation tasks.

B.1. Pre-training Hyper-parameters

For pre-training both the in-house CLIP baseline and CLOC, we mainly follow the hyper-parameters in (Radford et al., 2021) to train on our in-house datasets. The training images are identical for CLIP and CLOC, while CLOC is trained on the extra region-text annotations of the same images via the proposed VESL pipeline (details in Section C). Table A summarizes the general training hyper-parameters used for all experiments and the setup for components specific to CLOC.

In terms of the CLOC architecture, as illustrated in Figure 2, the image and text encoders including the attention pooling and projection layers follow the same as OpenAI-CLIP (Radford et al., 2021). Our `Prompter` consists of a positional encoding matrix for bounding boxes, and a single-layer single-head transformer encoder with another set of the global average pooler and a projection layer to map the region embeddings into the same dimension as the CLIP text/image embeddings.

B.2. Evaluation Tasks

We provide more details about the tasks constructed for evaluating the encoders in Section 5.

Zero-shot region tasks. Our CLOC training augments a new capability for CLIP to generate region-level embeddings. This enables us to perform zero-shot region-text tasks, in analogy to the image-text zero-shot tasks like ImageNet classification and COCO text-image retrieval that CLIP has been evaluated on.

In a similar rationale of image-level evaluation, we further construct region-level tasks including region object recognition and region-text retrieval. For region object recogni-

tion, the class names are encoded by the text encoder into class embedding. We do not add the text prompts (e.g., “a photo of ...”) to object classes used when CLIP (Radford et al., 2021) evaluated on image classification. The CLOC model takes all the labeled bounding boxes in the images to generate a region embedding $\mathbf{z} = \text{Prompter}(\mathbf{l}, f_I(\mathbf{x}))$. The class embedding with the highest similarity is predicted as the class of the region (i.e., out of 80 / 1203 classes for COCO / LVIS).

For region retrieval, similarly, the CLOC model encodes both the image regions and the region captions from the public region-text GRIT dataset that the regions are annotated by the Kosmos-2 pipeline (Peng et al., 2023). We randomly sampled a 2K image validation set for fast evaluation. We have verified it is statistically stable compared to the whole set that contains about 20M in total. Unlike image-text retrieval the image captions are likely unique, the objects in regions of many images might be duplicated. Therefore, we opt to report recall@10 rather than recall@1 for GRIT region retrieval in Table 3.

MLLM tasks. To demonstrate our CLOC can benefit MLLM end tasks as a better image backbone, we consider two sets of MLLM experiments.

First, we experiment with the FERRET MLLM that is capable of taking spatial referring inputs for grounding and referring VQA tasks². FERRET can consume a point, a bounding box, or a free-form referring. It designs a quite complicated visual sampler module that involves point sampling and k NN grouping. We suggest the readers refer to Figure 3 and Section 3.2 in (You et al., 2023) for more details. Here we consider two variants of use cases of CLOC compatible with FERRET: (1) we only take the ViT encoder in CLOC to replace the CLIP ViT and still use the original FERRET visual sampler or (2) we further replace the visual sampler with our simple `Prompter` (essentially a lightweight transformer encoder with box positional encodings) in Section 3.3 as illustrated in Figure 1(3b). More specifically, we simply convert all types of spatial referring as boxes. As evidenced by Table 4 and Table 5, our `Prompter` can indeed be a much simpler alternative and may perform even better as it is more consistent with CLOC pre-training.

Second, we evaluate on general VQA tasks that do not consider extra spatial referring inputs. The pre-trained ViT of CLOC is a drop-in-replacement of CLIP ViT in two sets of experiments of LLaVA-1.5 (Liu et al., 2023) and LLaVA-NeXT (Liu et al., 2024). The main difference includes different supervised fine-tuning (SFT) sets. Also, LLaVA-NeXT uses the AnyRes technique that decomposes an image into several subimages that are encoded inde-

²We use the official Ferret codebase: <https://github.com/apple/ml-ferret>.

Table A. Pre-training hyper-parameters and settings for the in-house CLIP baseline and CLOC.

| General | |
|-------------------------|--|
| Batch size | 32768 |
| Image size | 224×224 (ViT B/16) or 336×336 (ViT L/14, H/14) |
| Image pre-processing | long-side resizing with padding (i.e., <code>tf.image.resize_with_pad</code>) |
| Text tokenizer | T5 (Raffel et al., 2020), lowercase |
| Text maximum length | 77 tokens |
| Steps | 439087 (i.e., $\sim 14B$ examples seen) |
| Optimizer | AdamW ($\beta_1 = 0.9, \beta_2 = 0.98$) |
| Peak learning rate (LR) | 0.0005 |
| LR schedule | cosine decays with linear warm-up (first 2k steps) |
| Weight decay | 0.2 |
| Dropout rate | 0.0 |
| CLOC | |
| # of sampled regions | maximum $M = 4$ per image |
| CLOC loss weight | $\lambda = 1.0 \times \frac{\# \text{of images contain region text in the mini-batch}}{\text{batch size}}$ (in Equation 4) |
| Encoding box prompts | sinusoidal positional encoding of coordinates (top-left and bottom right of a box) |
| Encoding text prompts | encoded by re-using the text encoder (w/ pooling & projection) |
| Prompter architecture | a single-layer single-head transformer encoder (same feature dimension as the ViT) |
| BoxHead architecture | 2-layer MLP with GELUs activations (Hendrycks & Gimpel, 2016) |

pendently with the ViT and concatenated together as the input for the decoder. LLaVA-1.5 by default freezes the ViT while LLaVA-NeXT fine-tunes all parameters during SFT. Since the official LLaVA-NeXT is trained on some proprietary datasets that are not reproducible, we use the Open-LLaVA-NeXT repository³. Our experiments in Table 6 demonstrate CLOC not only slightly improves such general VQA besides FERRET tasks but also generalizes well for both LLaVA-1.5 and LLaVA-NeXT settings.

C. VESL Data Engine

We provide more information about our pseudo-labeling data pipeline proposed in Section 4.

C.1. Implementation Details

As already mentioned in Section 4, there are three steps for VESL: image re-captioning, region phrase candidates extraction from the captions, and open-vocabulary (OV) detection given the region candidates as queries.

For the re-captioning, the goal is to replace AltText with long, diverse, and detailed captions that can be used to generate more visual concepts as the region candidate phrases for the OV detector. Technically, any strong image captioner can be an option. In our paper, we adopt the VeCap pipeline (Lai et al., 2024) and leverage their images with enriched captions.

To extract region phrase candidates from the long captions, we adopt name entity recognition (NER) to extract leaf entities from the captions, inspired by (Zhang et al., 2022).

³<https://github.com/xiaoachen98/Open-LLaVA-NeXT>

The code listing below shows the Python example implementation, where stop-words and common generic words are filtered, following (Minderer et al., 2024).

Generating bounding boxes and assigning region captions can be done by querying an OV objection detector. We adopted the OWLv2 detector (Minderer et al., 2024) with their pre-trained L/14 checkpoint⁴ to annotate inputs with 448×448 image resolutions.

```

1 from typing import Iterable, List
2 import nltk
3
4 # STOPWORDS_EN and COMMON_GENERIC_WORDS are following:
5 # Section A.2 (Minderer et al., 2024)
6
7 # Stopwords from nltk.corpus.stopwords.words("english")
8 STOPWORDS_EN = frozenset({
9     "a", "about", "above", "after", "again", "against",
10    "all", "am", "an",
11    "and", "any", "are", "as", "at", "be", "because",
12    "been", "before", "being",
13    "below", "between", "both", "but", "by", "can",
14    "did", "do", "does",
15    "doing", "don", "down", "during", "each", "few",
16    "for", "from", "further",
17    "had", "has", "have", "having", "he", "her", "here",
18    "hers", "herself",
19    "him", "himself", "his", "how", "i", "if", "in",
20    "into", "is", "it", "its",
21    "itself", "just", "me", "more", "most", "my",
22    "myself", "no", "nor", "not",
23    "now", "of", "off", "on", "once", "only", "or",
24    "other", "our", "ours",
25    "ourselves", "out", "over", "own", "s", "same",
26    "she", "should", "so",
27    "some", "such", "t", "than", "that", "the", "their",
28    "theirs", "them",
29    "themselves", "then", "there", "these", "they",
30    "this", "those", "through",
31    "to", "too", "under", "until", "up", "very", "was",
32    "wasn"
33})

```

⁴OWLv2 CLIP L/14 ST+FT in: https://github.com/google-research/scenic/tree/main/scenic/projects/owl_vit

```

21     "we", "were", "what",
22     "when", "where", "which", "while", "who", "whom",
23     "why", "will", "with",
24     "you", "your", "yours", "yourself", "yourselves"
25 )
26
27 # These words were found by manually going through the
28 # most common 1000 words
29 # in a sample of alt-texts and selecting generic words
30 # without specific meaning:
31 COMMON_GENERIC_WORDS = frozenset({
32     "alibaba", "aliexpress", "amazon", "available", "
33     background", "blog", "buy",
34     "co", "com", "description", "diy", "download", "
35     facebook", "free", "gif",
36     "hd", "ideas", "illustration", "illustrations", "
37     image", "images", "img",
38     "instagram", "jpg", "online", "org", "original", "
39     page", "pdf", "photo",
40     "photography", "photos", "picclick", "picture", "
41     pictures", "png", "porn",
42     "premium", "resolution", "royalty", "sale", "sex",
43     "shutterstock", "stock",
44     "svg", "thumbnail", "tumblr", "tumgir", "twitter",
45     "uk", "uploaded", "vector",
46     "vectors", "video", "videos", "wallpaper",
47     "wallpapers", "wholesale", "www",
48     "xxx", "youtube"
49 })
50
51
52 def _is_all_stopwords(query_words: Iterable[str]) ->
53     bool:
54     return set(query_words).issubset(STOPWORDS_EN)
55
56
57 def _get_name_entities(words: List[str]) -> List[str]:
58     """
59     Returns name entities of image caption as queries,
60     similar to GLIP.
61     """
62     pos_tags = nltk.pos_tag(words)
63     grammar = "NP: {<DT>?<JJ.*>*<NN.*>+}"
64     cp = nltk.RegexpParser(grammar)
65     result = cp.parse(pos_tags)
66
67     queries = []
68     for subtree in result.subtrees():
69         if subtree.label() == "NP":
70             query_words = [t[0] for t in subtree.leaves
71 ()]
72             # Don't use it if it only consists of stop
73             words.
74             if _is_all_stopwords(query_words):
75                 continue
76             queries.append(" ".join(query_words))
77     return queries
78
79
80 def find_noun_phrases(
81     caption: str, max_num_queries: int = 20,
82 ) -> List[str]:
83     caption = caption.lower()
84     tokens = nltk.word_tokenize(caption)
85     # Remove common generic words.
86     words = [w for w in tokens if w not in
87     COMMON_GENERIC_WORDS]
88     queries = _get_name_entities(words)[:, :
89     max_num_queries]
90     return queries
91
92 candidate_queries = find_noun_phrases(caption)

```

Listing 1. Python example codes for Step 2 of VESL in Section 4 for extracting text candidate queries from a caption.

C.2. More Visualizations

As mentioned in the remarks of Section 4, we found the AltText sourced from the original web-crawled images might not have enough details describing the subimage content, thus limiting the diversity and quality of the text candidate queries for the OV detector to detect more meaningful objects. In Figure A we show some cherry-picked examples (since the web-crawled images are quite noisy) just to demonstrate the reasons why high-quality captions can help our region-text annotation pipeline. In (Minderer et al., 2024), the queries are generated by the n -grams of the AltText, while ours are by NER as described in Section C.1 on top of the visually-enriched re-captions. Note that, in both methods we use the same pre-trained OV detector but with different approaches to generate the queries.

As shown in Figure A, for easier images like the first row, both methods are doing reasonably well to detect “message card”. However, when the scene becomes complicated (e.g., the second row), our methods can detect more objects since more visual concepts can be extracted from our rich caption as queries for the detector. Similarly, it can be seen that our method captures more items that the AltText missed, e.g., “banana”, “eggs”, “butter”, etc in the third row; “drawstring” in the fourth row; “apples” and “vases” in the last row. Also, it is more likely to extract a more detailed description of the region rather than a class name, such as “green-roofed cottage nestles” in Figure 3 and “decorative metal tree sculptures” in the image in the last row of Figure A. We believe such high-quality region labels essentially contribute to better supervision for CLOC pre-training.

D. More Discussions

Limitations. One limitation for CLOC is the labeling efforts in preparing the training data. As we discussed in Section 1, there are no public large-scale region-text datasets since it is expensive to infer such labels up to the scales we consider here. Unlike previous work (Zhong et al., 2022) that cropping boxes from images for annotating, our VESL inference in *image-level* thus the cost does not scale with the number of detected regions. With that being said, such inference still requires hundreds of GPUs running in parallel for days to scale up to billions of images. We are working on releasing the annotations to accelerate future research for the community.

For CLOC, we focus on the training objective and framework formulation, while making minimal efforts on hyper-parameter tuning, architecture search, dataset cleaning, and etc., thus better performance could be achieved. Besides, although we have included extensive standard evaluation tasks, the fine-grained region knowledge could also be used.

ful on more other under-explored tasks.

Future directions. We suggest promising future directions. In Section 3.2, our `Prompter` formulation can take flexible prompts to guide the embeddings for specific tasks. In this work, we consider a prompt as a single bounding box or a text caption, but it has the potential to expand to various types such as points, a mask, users’ free-form referring, or multiple prompts in multiple types together. We think a more versatile `Prompter` with co-designs for different objectives can have a big potential. Similarly, our VESL labeling pipeline limits to detection box format. Annotators supported for more formats may further boost it. We believe our approach is promising, as more attention has been drawn recently for better re-captions (Li et al., 2024; Fan et al., 2024) that VESL relies on. In addition, CLOC model provides a new capability to extract region features without further training, and thus can be used as a foundation model for exploring new VL applications.

Training cost. We comment on the computation cost of our framework. Our large models (ViT L/14) were trained on 1024 v5p TPUs for about 6 days. To optimize Equation 2, CLOC needs extra computation. The main overheads come from the contrastive matrix but not the lightweight `Prompter`. Fortunately, we found it feasible since (1) only a few boxes in each image need to be sampled per update; (2) the loss computation becomes a smaller proportion when the ViT scales up. Overall, we found the computation acceptable compared to CLIP. More memory-efficient optimization like SigLIP (Zhai et al., 2023) can be implemented with JAX `shard_map`⁵ ops.

Discussions on design rationals. Besides the main discussions we have stressed in the main text, here we provide more thoughts behind our design rationales that a reader may be wondering.

(1) *Why not use a local-enhanced encoder?* We would like to note that many encoders with great localization like DINOv2 (Oquab et al., 2023), OWLv2 (Minderer et al., 2024), CLIP-Self (Wu et al., 2023), etc. are developed specifically for dense vision tasks that cannot perform image zero-shot tasks like CLIP and CLOC. We would like to emphasize that our goal is to build a drop-in-replacement of CLIP encoder with better localization, without sacrificing CLIP’s original capabilities such as image zero-shot tasks and its important backbone position for MLLMs. Furthermore, perhaps well-known within the MLLMs community, these encoders have been shown in recent reports that they are not comparable enough to compete with CLIP as the vision backbone for MLLM tasks (Tong et al., 2024) due to CLIP’s superiority in vision-language alignment. We thus believe enhancing CLIP itself is more demanding as this

paper focuses on.

(2) *Why not just train a CLIP with object detection?* One may wonder why we do not just train an encoder with joint optimization of the CLIP contrastive loss with some object detection loss instead of the CLOC design of Equation 4.

Although it sounds like a plausible approach, we would like to point out that contrastive pre-training and object detection are fundamentally quite different in their technical rationales. CLIP pre-training is often on large batches of low-resolution and noisy images, while object detection is trained on small batches of high-resolution images. CLIP is by default trained from scratch and object detection is typically initialized from pre-trained encoders and focuses on the detection head. Furthermore, detection requires heavy computation on box proposals to detect all boxes appearing in an image, while our region-text contrastive design allows us to flexibly sample fewer regions per image as motivated in Equation 3. Overall, their data pipeline and distributed training setup are not on the same scale thus such joint training may not be very reasonable.

With that being said, some previous works do have attempts that are the exceptions but only for some but not all of the mentioned aspects, and mainly for the purpose of detection. For instance, DetCLIP-v2 (Yao et al., 2023b) adds image-text contrastive loss into detection loss to improve open-vocabulary capability for detection. OWLv2 pre-trains the detector with rather small resolutions but still with a batch size of a maximum 256 since each image will need to predict up to 100 boxes during training. Both DetCLIP-v2 and OWLv2 fine-tune from a pre-trained encoder.

On the contrary, we study pre-training the encoder from scratch, which may be complementary to the previous efforts. CLOC maximizes the similarity in co-design with CLIP, thus making it much easier to develop within the same codebase.

(3) *Do we really need to train CLOC from scratch? What if we fine-tune from CLIP?* As CLIP pre-training is expensive, one may wonder if it is necessary to train from scratch on the proposed region-text datasets, or if we can initialize from a standard CLIP trained on image-text pairs only and fine-tunes with CLOC for a shorter stage. Our early investigation, even with extensive hyper-parameter tuning, suggests it is likely to be suboptimal compared to training from scratch directly. For instance, we initialize from the CLIP model ② in Table 3 and fine-tunes it for another extra 100K steps with the CLOC training loss Equation 4. The model reaches 64.1%/19.1% mAcc on COCO/LVIS region recognition, which is much worse than 70.1%/27.2% of the trained-from-scratch model ⑧, even with more overall training steps.

⁵<https://jax.readthedocs.io/en/latest/jep/14273-shard-map.html>

Minderer et al., 2024
(AltText + ngrams)

Ours (VESL)

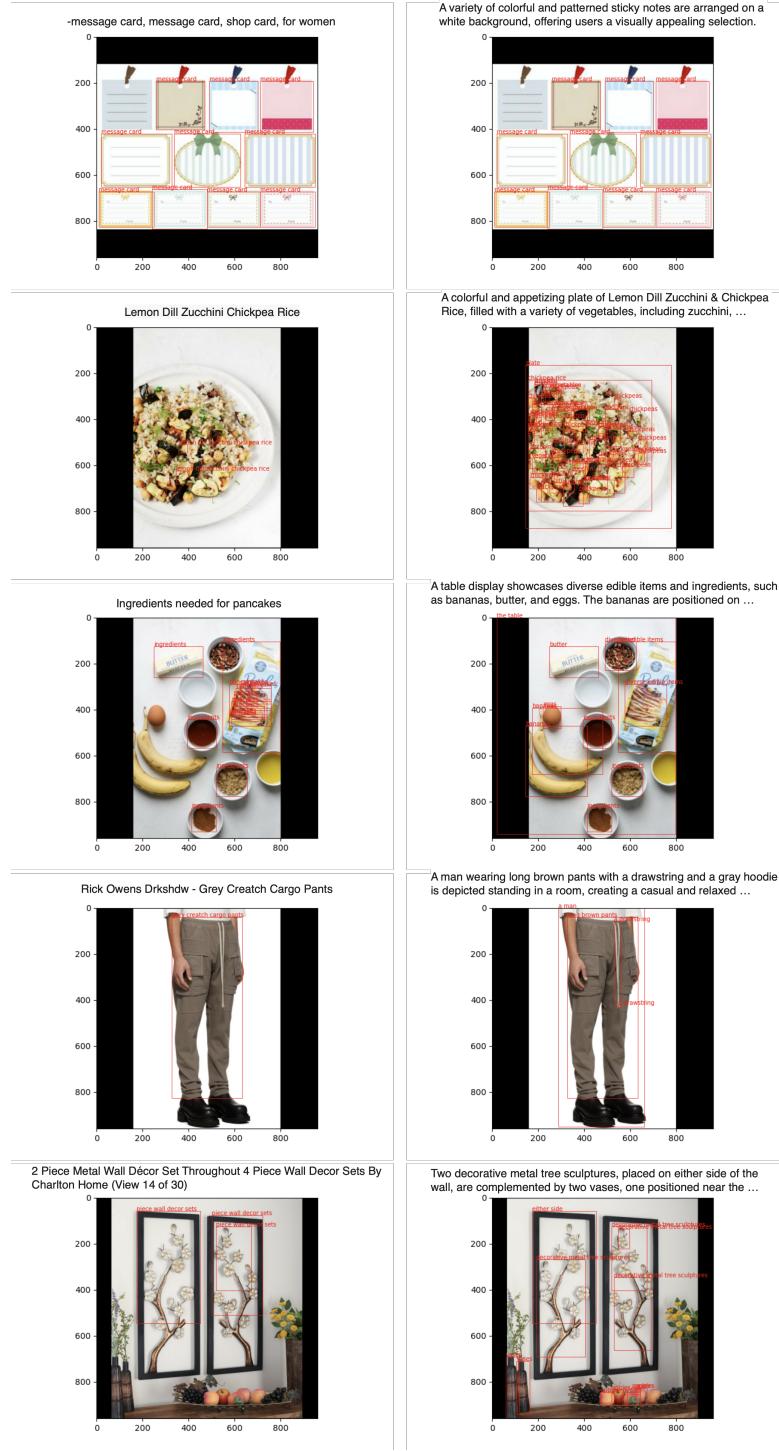


Figure A. Examples comparing our VESL and the labeling approach in (Minderer et al., 2024) that directly uses the n -grams of the crawled AltText. For VESL, each image is annotated with the visual-enriched caption to replace the AltText, which is used to generate region text candidates that capture the image content better.