ViLLA: Fine-Grained Vision-Language Representation Learning from Real-World Data

Appendix

Maya Varma Jean-Benoit Delbrouck Sarah Hooper Akshay Chaudhari Curtis Langlotz
Stanford University
{mvarma2, jbdel, smhooper, akshaysc, langlotz}@stanford.edu

Contents

A. Extended Related Work

In this section, we provide an extended analysis of related work. Our work builds on several directions explored in prior studies.

One-to-One Vision-Language Models: Early works in computer vision performed image representation learning by training on large labeled datasets, where annotations were primarily obtained from crowdworkers [18, 27]. However, the recent development of contrastive self-supervised methods have greatly reduced the need for large-scale, annotated datasets. Driven by methods like SimCLR and MoCo, contrastive self-supervised methods learn representations by pulling similar images together and pushing dissimilar images apart in the latent space [5, 17]. Recent works (*e.g.* CLIP, ALIGN, BASIC, ConVirt) extended these approaches to multimodal datasets, where each image is pulled towards an associated caption and pushed away from dissimilar captions in the latent space [36, 24, 35, 49, 13]. When trained over large-scale datasets consisting of millions of image-text pairs collected from the web, these vision-language models (VLMs) were shown to be highly effective across a variety of classification, retrieval, and robustness tasks [10, 11, 36, 24, 35]. We refer to this class of VLMs as one-to-one models, since a single embedding of the entire image is aligned with a single embedding of the textual caption.

Fine-Grained Representation Learning: Models that possess knowledge of fine-grained region-level information have been shown in prior work to exhibit a number of advantages. For example, given the image of the cow shown in Figure 1, a model that can effectively capture region-level details would learn the features (*e.g.* cow, trees, clouds, etc.) appearing in each bounding box. Computer vision models that can understand details at the region-level are particularly useful for fine-grained reasoning tasks, such as region-level retrieval and object detection [19, 50, 37, 31]. Additionally, [39] demonstrated that medical image models provided with pathology bounding boxes during training are more resistant to spurious correlations, suggesting that knowledge of region-level information can also improve model robustness. These works demonstrate the importance of developing models that can capture region-level details.

However, recent studies have demonstrated that standard one-to-one VLMs often struggle to capture fine-grained region-level information [34]. In particular, [50] applied CLIP, a widely-used one-to-one VLM, to both an image classification task (ImageNet [8]) and a region classification task (LVIS [16]) with a similar number of classes. Results showed that classification accuracy dropped significantly from 60% on ImageNet to 19% on LVIS, which the authors hypothesized was likely due to the fact that CLIP was trained in a one-to-one fashion on image-text pairs and did not learn to capture fine-grained relationships between image regions and textual attributes. Additionally, [28] presents a distinct but related study demonstrating that one-to-one VLMs like CLIP often fail to understand subtle differences between images. In particular, given a contextual description, VLMs are evaluated on their ability to retrieve the correct image from a set of ten candidate images that vary only in fine-grained details. Here, CLIP performs almost 60 points worse than human accuracy. Our work extends these lines of research by systematically evaluating the effect of image-text pairwise complexity in the training dataset on the fine-grained reasoning ability of one-to-one VLMs.

Several prior approaches have been proposed for learning fine-grained region-level information from image-text datasets. One line of recent work (*e.g.* GLIP, X-VLM) leverages large quantities of human-labeled region-text pairs during model training [30, 46]. A similar line of research has used object detectors pretrained on labeled region-text pairs to generate pseudo-labels for image regions in a semi-supervised fashion [40, 43]; the generated region-label pairs can then be used during model training. Although both approaches have led to significant improvements on vision-language tasks, obtaining human-annotated region-text pairs is expensive, time-consuming, and difficult to extend to other domains, such as medical images. In order to mitigate the need for human-annotated region-text pairs, [50] proposed RegionCLIP, which uses the pretrained CLIP model [36] in a zero-shot fashion to match candidate image regions with plausible textual attributes; these mappings are then utilized during training. This approach does not require ground-truth region-text pairs and was shown to work well across open vocabulary and zero-shot object detection tasks; however, this approach relies heavily on the CLIP model, which (a) has been shown to work poorly on localizing regions to text [50] and (b) cannot be accurately applied in a zero-shot fashion to out-of-domain data (such as medical images) [36]. Our approach ViLLA aims to address these issues by introducing a specific training phase to learn region-text mappings, rather than directly using an off-the-shelf pretrained VLM model. Our work is also inspired by other recent studies in open vocabulary object detection [45, 31, 15, 37].

Another related line of work aims to learn fine-grained patterns by aligning individual image patches with textual tokens. [44] proposed FiLIP, which leverages self-supervised contrastive learning to learn token-wise similarities between individual image patches (generated from a vision transformer) and text tokens (generated from a text transformer). LOUPE follows a similar set-up, although a game theoretic algorithm is utilized to learn fine-grained interactions [29]. In the medical domain, [20] introduced GLoRIA, which adds a local contrastive loss term to the optimization objective in order to match image patches with textual tokens. Although these methods demonstrate performance improvements over standard one-to-one VLMs on several tasks, learning relationships between all image patches and all text tokens is extremely computationally expensive (*e.g.* FILIP leverages a number of computational tricks in order to make training feasible) and is likely to model patches and tokens that are not semantically meaningful (*e.g.* solid-color image patches with no discernible features). Furthermore, these approaches will be particularly expensive when applied to image-text pairs with high pairwise complexity, which are likely to have a large number of textual tokens.

Learning from Real-World Multimodal Data: Our work relates closely to prior studies that have developed VLMs for real-world datasets.

In the medical domain, early works learned medical image representations by training classification models with either (a) labels generated by experts, which are often time-consuming and expensive to obtain, or (b) labels obtained from applying labeling rules to radiologist reports, which are often noisy and limited to a few pre-defined categories [41, 1, 21, 42]. In order to address these issues, [49] proposed ConVIRT, the first contrastive, self-supervised VLM for learning representations directly from chest X-rays and radiologist reports. ConVIRT, which leverages a one-to-one training approach, was shown to significantly outperform image-only baselines across a range of medical tasks. Other one-to-one VLMs (*e.g.* BiomedCLIP, BioViL) have also been introduced for biomedical tasks [47, 3]. In order to improve the fine-grained reasoning ability of ConVIRT, [20] proposed GLoRIA, which uses a local contrastive loss function as discussed in the previous section. Adding this extra loss computation is expensive, and the model is trained using only a small portion of the radiology report ("Impression" section). In the product domain, several recent studies have focused on training VLMs on fashion data, IKEA catalogs, and car manuals [2, 6]. In order to capture finer-grained signal from multimodal fashion datasets, [48] introduced MaskCLIP, which aligns image patches obtained from a garment segmentation mask with textual tokens extracted from simple phrase-based captions; MaskCLIP is introduced as part of a pipeline for synthesizing fashion designs and is not directly evaluated on fine-grained reasoning tasks.

Our work extends upon these lines of research by introducing a lightweight training mechanism for capturing fine-grained relationships even when training datasets possess high image-text pairwise complexity. Of particular note, we demonstrate that our approach ViLLA can effectively learn representations across multiple real-world domains.

B. Extended Dataset Complexity Evaluations with DocMNIST

In this section, we provide additional details on our dataset complexity evaluations as described in Section 4.

B.1. DocMNIST Implementation Details

We introduce DocMNIST (adapted from the popular MNIST benchmark [9]), a customizable multimodal training dataset consisting of synthetically-generated image-text pairs.

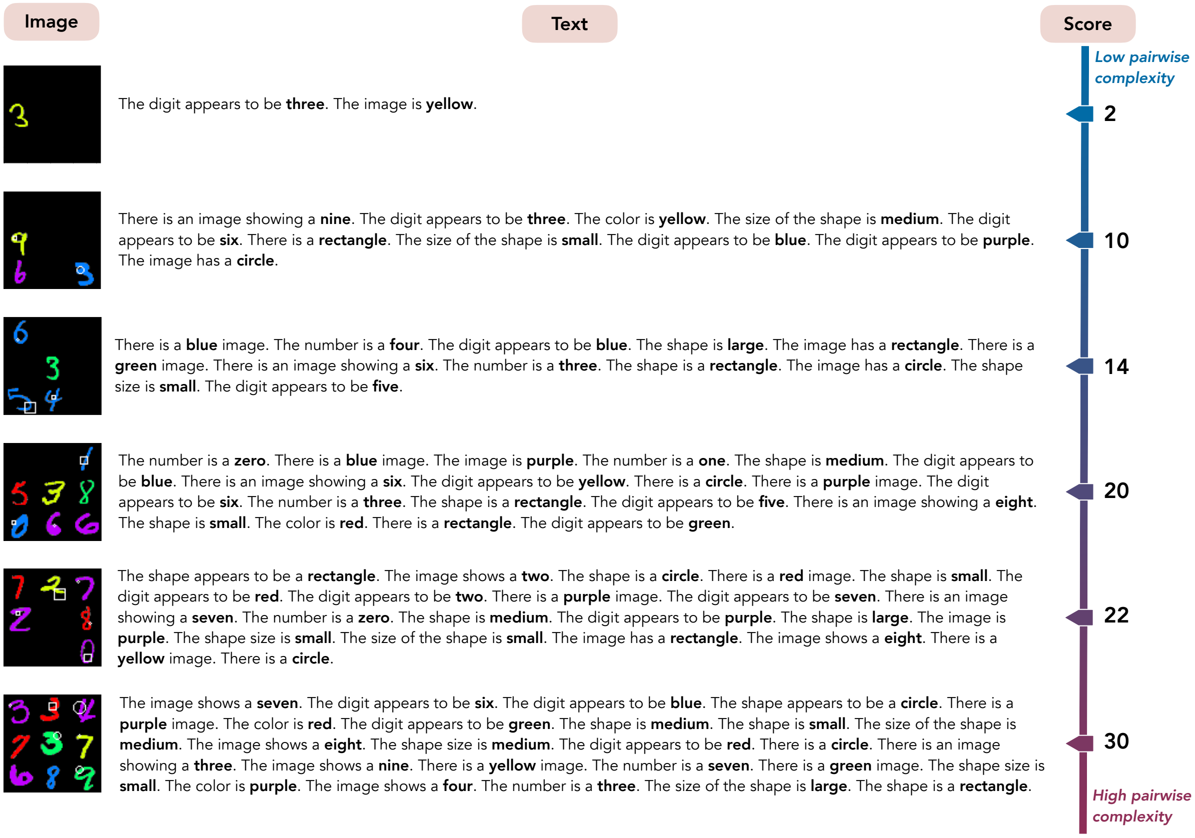
We first discuss our methodology for generating images. Each image in DocMNIST is set to a size of $3 \times 84 \times 84$; the image is then subdivided into a square grid with 9 regions of size $3 \times 28 \times 28$. We establish an attribute set A consisting of 20 attributes divided across 4 categories as follows: 10 digits (zero, one, two, three, four, five, six, seven, eight, nine), 5 digit colors (purple, blue, green, yellow, red), 2 shapes (rectangle, circle), and 3 shape sizes (small, medium, and large). We selected these attributes in order to emulate properties of real-world multimodal training datasets; for instance, size-based attributes are often used in medical reports when describing anatomic features (*e.g.* "the heart appears enlarged"). We use the following procedure to assign attributes to regions:

- First, we randomly choose a region from the set of nine possible regions.
- We randomly select a digit from the set of ten possible digits $\{$zero, one, two, three, four, five, six, seven, eight, nine$\}$. We then sample a digit image from the MNIST dataset with the selected digit label and resize the image to $3 \times 28 \times 28$. The digit image is pasted within the chosen region.
- We randomly select a color from the set of five possible colors $\{$purple, blue, green, yellow, red$\}$. The selected color is applied to the digit.
- Next, we randomly sample a shape from the set $\{$no shape, rectangle, circle$\}$. If *no shape* is selected, then no further action is taken and the region will consist of only a digit and a color. If a rectangle or circle is selected, we then randomly sample a size from the set $\{$small, medium, large$\}$. The shape is drawn in a random location within the region. For circles, *small* corresponds to a radius of 1 pixel, *medium* corresponds to a radius of 3 pixels, and *large* corresponds to a radius of 5 pixels. For rectangles, *small* corresponds to a length and width of 1 pixel, *medium* corresponds to a length and width of 4 pixels, and *large* corresponds to a length and width of 7 pixels.

For each image, we repeat the above procedure until the number of region-attribute pairs reaches the user-specified value for c, which defines the average pairwise complexity. The final size of the training dataset is constrained by a pre-defined attribute budget b, which represents the total number of attributes across all images. We continue generating image-text pairs until the budget b is reached.

Next, we discuss our methodology for generating text. For each image, we automatically generate a textual description by filling the selected attributes into pre-defined templates. Templates are defined for each attribute category as follows, where components in square brackets are replaced with the corresponding attribute:

- Digit: $\{$"The image shows a [digit]", "The digit appears to be [digit]", "There is an image showing a [digit]", "The number is a [digit]"$\}$
- Digit Color: $\{$"The color is [color]", "The digit appears to be [color]", "There is a [color] image", "The image is [color]"$\}$



The digit appears to be **three**. The image is **yellow**.

There is an image showing a **nine**. The digit appears to be **three**. The color is **yellow**. The size of the shape is **medium**. The digit appears to be **six**. There is a **rectangle**. The size of the shape is **small**. The digit appears to be **blue**. The digit appears to be **purple**. The image has a **circle**.

There is a **blue** image. The number is a **four**. The digit appears to be **blue**. The shape is **large**. The image has a **rectangle**. There is a **green** image. There is an image showing a **six**. The number is a **three**. The shape is a **rectangle**. The image has a **circle**. The shape size is **small**. The digit appears to be **five**.

The number is a **zero**. There is a **blue** image. The image is **purple**. The number is a **one**. The shape is **medium**. The digit appears to be **blue**. There is an image showing a **six**. The digit appears to be **yellow**. There is a **circle**. There is a **purple** image. The digit appears to be **six**. The number is a **three**. The shape is a **rectangle**. The digit appears to be **five**. There is an image showing a **eight**. The shape is **small**. The color is **red**. There is a **rectangle**. The digit appears to be **green**.

The shape appears to be a **rectangle**. The image shows a **two**. The shape is a **circle**. There is a **red** image. The shape is **small**. The digit appears to be **red**. The digit appears to be **two**. There is a **purple** image. The digit appears to be **seven**. There is an image showing a **seven**. The number is a **zero**. The shape is **medium**. The digit appears to be **purple**. The shape is **large**. The image is **purple**. The shape size is **small**. The size of the shape is **small**. The image has a **rectangle**. The image shows a **eight**. There is a **yellow** image. There is a **circle**.

The image shows a **seven**. The digit appears to be **six**. The digit appears to be **blue**. The shape appears to be a **circle**. There is a **purple** image. The color is **red**. The digit appears to be **green**. The shape is **medium**. The shape is **small**. The size of the shape is **medium**. The image shows a **eight**. The shape size is **medium**. The digit appears to be **red**. There is a **circle**. There is an image showing a **three**. The image shows a **nine**. There is a **yellow** image. The number is a **seven**. There is a **green** image. The shape size is **small**. The color is **purple**. The image shows a **four**. The number is a **three**. The size of the shape is **large**. The shape is a **rectangle**.

Figure 5. Examples of randomly-selected image-text pairs and the associated pairwise complexity scores from each of the six DocMNIST training datasets. Each textual attribute (bolded) appears in at least one region of the image. The image-text pairwise complexity score measures the number of distinct region-attribute pairs.

- Shape: {"The shape is a [shape]", "The shape appears to be a [shape]", "There is a [shape]", "The image has a [shape]"}
- Shape Size: {"The shape size is [size]", "The size of the shape is [size]", "The shape is [size]"}

In order to construct the final description, the generated sentences are shuffled and any duplicate sentences are pruned.

B.2. Extended Evaluations

In order to systematically evaluate the role of training dataset complexity on the fine-grained reasoning ability of a standard one-to-one VLM, we generate a set of six DocMNIST training datasets that vary in average image-text pairwise complexity. In Figure 5, we provide examples of randomly-selected image-text pairs from each of the six training datasets.

For each generated DocMNIST dataset, we train a standard one-to-one VLM to contrastively learn alignments between images and the associated text. The image encoder consists of a ResNet-50 model initialized with pretrained CLIP RN50 weights [36, 18]. The output of the encoder is an L_2-normalized embedding of dimension 1024. The text encoder consists of a pretrained CLIP text encoder with frozen weights. Since textual descriptions may be longer than the maximum token limit of the text encoder, we split each description into sentences, compute the embedding of each sentence, and then average the embeddings together; this yields a single L_2-normalized text embedding of dimension 1024. The VLM is optimized using a standard bidirectional contrastive loss function with a temperature of 0.07. We train on a single NVIDIA V100 GPU for 100 epochs with early stopping if the loss fails to decrease for 5 consecutive epochs. We use an initial learning rate of 5e-5 and a batch size of 256. We repeat this procedure with three different random seeds for each generated DocMNIST dataset.

At inference time, we construct a fixed DocMNIST test set with 1196 images; the images consist of a total of 5982 regions. We evaluate our trained VLMs on both a fine-grained text → region retrieval task and a region → text retrieval task. We

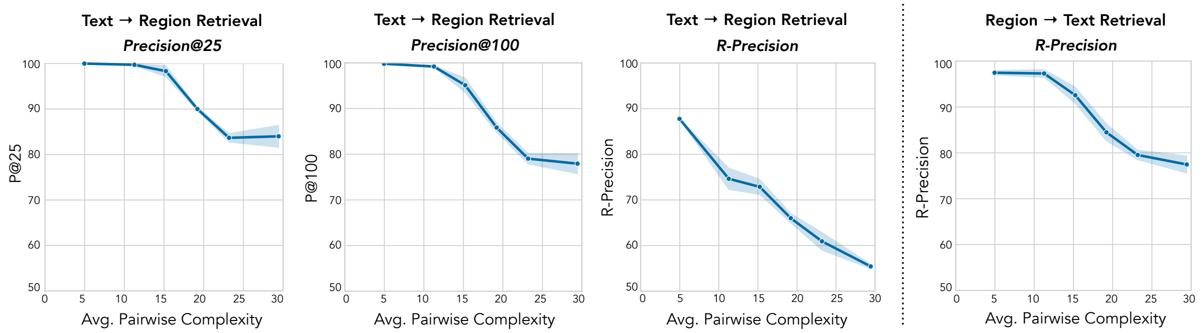


Figure 6. We expand on the results shown in Figure 3 by providing Precision@25 and Precision@100 metrics for the text → region retrieval task. Across all metrics, representations generated using one-to-one VLMs exhibit significant performance degradations as the average pairwise complexity of the dataset increases.

emphasize here that although the VLMs were trained at the *image-level*, we are specifically evaluating the ability of the models to understand *region-level* features.

Given a textual query, the text $\rightarrow$ region retrieval task involves retrieving image regions that capture the content of the query. To this end, we use the trained VLM to generate an embedding for each individual region in the DocMNIST test set. Then, for each of the twenty attributes in set A, we generate query sentences using the templates defined in the previous section; this yields a total of twenty textual query embeddings. We use dot-product similarity to identify the top-k regions from the DocMNIST test set that are most similar to each query. In Figure 3, we report the R-Precision metric, which defines k as the total number of ground-truth regions in the DocMNIST test set that possess the queried attribute.

We follow a similar procedure for the region $\rightarrow$ text retrieval task. For each of the 5.9k regions in the DocMNIST test set, the region $\rightarrow$ text retrieval task determines if we can identify the textual attributes depicted in the region. We compute the dot-product similarity between the region embedding and the 20 attributes in set A. We note that a given region includes at most one attribute from each of the four categories (as defined in Section B.1); as a result, we identify the top-scoring attribute from each of the four categories (digit, digit color, shape, and shape size). In Figure 3, we report the R-precision metric, which determines the percentage of the top-k identified attributes that are accurate; here, k is defined as the total number of ground-truth attributes in the region, which will range between 2 and 4.

In Figure 6, we expand on the results shown in Figure 3 by providing Precision@25 and Precision@100 metrics for the text $\rightarrow$ region retrieval task. Additional details on the fine-grained text $\rightarrow$ region and region $\rightarrow$ text retrieval tasks are provided in Appendix Section D.

C. ViLLA Implementation Details

In this section, we provide additional implementation details for ViLLA.

C.1. Datasets

In this work, we use four training datasets across various real-world domains: DocMNIST (synthetic images), DeepFashion (product data), MIMIC-CXR (medical images), and COCO (natural images). Additional details on each dataset are provided below:

DocMNIST: We create the synthetic DocMNIST dataset using the procedure described in Section 4. In Section 4, we generated six versions of the DocMNIST training set with varying complexities; for the remainder of this work, we use the version of the DocMNIST training set with the highest complexity. Specifically, the training set has an average pairwise complexity of $c = 29.4$ with a total of $b = 300K$ attributes. We also generate a validation dataset with $b = 10k$ attributes.

DeepFashion [33, 25]: The DeepFashion-Multimodal dataset consists of 44,096 high-resolution human images obtained from clothing retail websites. We filter the dataset to include 42,537 images that have associated textual descriptions, and we resize all images to 3 × 335 × 228. We randomly assign 70% of the dataset (corresponding to 29,694 images) to the training set, 10% of the dataset (corresponding to 3985 images) to the validation set, and the remaining 20% of the dataset

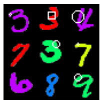
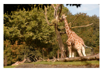
Dataset	Image	Text
DocMNIST		The image shows a seven. The digit appears to be six. The digit appears to be blue. The shape appears to be a circle. There is a purple image. The color is red. The digit appears to be green. The shape is medium. The shape is small. The size of the shape is medium. The image shows a eight. The shape size is medium. The digit appears to be red. There is a circle. There is an image showing a three. The image shows a nine. There is a yellow image. The number is a seven. There is a green image. The shape size is small. The color is purple. The image shows a four. The number is a three. The size of the shape is large. The shape is a rectangle.
DeepFashion		Her sweater has long sleeves, cotton fabric and plaid patterns. The neckline of it is lapel. The pants the lady wears is of long length. The pants are with denim fabric and solid color patterns. The female also wears an outer clothing, with knitting fabric and solid color patterns. This woman is wearing a pair of socks.
MIMIC-CXR		Findings and Impression sections from radiologist reports (excluded in this figure due to data use restrictions)
COCO		A giraffe eating food from the top of the tree. A giraffe standing up nearby a tree. A giraffe mother with its baby in the forest. Two giraffes standing in a tree filled area. A giraffe standing next to a forest filled with trees.

Figure 7. Examples of randomly-selected image-text pairs from each of the four training datasets: DocMNIST, DeepFashion, MIMIC-CXR, and COCO. Note that due to data use restrictions for the MIMIC-CXR dataset, we provide a representative example of a chext X-ray from the web and do not provide a text sample.

(corresponding to 8858 images) to the test set. Each image is annotated with structured labels, which we filter to include 58 labels divided across 16 categories as follows: glasses (eyeglasses, sunglasses, glasses in hand or on clothes), hat (yes), lower clothing length (long, medium short, three-quarter, three-point), lower color (lattice, pure color, floral, color block, graphic, striped), lower fabric (cotton, chiffon, leather, knitted, denim), neckline (square, lapel, round, standing, v-neck, suspenders), neckwear (yes), outer color (pure color, graphic, color block, floral, lattice, striped), outer fabric (cotton, chiffon, knitted, denim, leather), ring (yes), sleeve length (sleeveless, medium-sleeve, short-sleeve, long-sleeve), socks (leggings, socks), upper color (floral, pure color, graphic, lattice, color block, striped), upper fabric (leather, chiffon, knitted, cotton, denim), waist accessory (belt, some accessory), wrist accessory (yes).

MIMIC-CXR [26, 14]: The MIMIC-CXR dataset consists of 377,110 chest X-ray images and associated physician reports obtained from the Beth Israel Deaconess Medical Center. Following prior work [20], we filter the dataset to include only frontal views (AP or PA). We also remove any images from the dataset that do not have a valid report; here, we define a valid report as one that has a non-empty Findings or Impression section. The final dataset includes a total of 231,564 images, and we resize all images to 3 × 256 × 256. We randomly assign 70% of the dataset (162,417 images) to the training set, 10% of the dataset (23,592 images) to the validation set, and 20% of the dataset (45,555 images) to the test set. Since medical reports often include a large sentences with negative findings (*e.g. the patient does not have pneumonia*), we use RadGraph to remove any sentences discussing absent entities [23]. The final textual description includes all remaining sentences from the Findings and Impression sections.

COCO [32]: The Microsoft COCO training dataset consists of 114,648 natural images, each associated with five captions. We use the train2017 data split for training and the val2017 data split for inference.

In order to estimate the average image-text pairwise complexity of each dataset (as shown in Table 1), we compute the number of ground-truth region-attribute pairs in each image as follows.

- *DocMNIST*: As detailed in Section 4, the average image-text pairwise complexity of the DOCMNIST dataset is 29.4.
- *DeepFashion*: The DeepFashion-Multimodal dataset does not provide ground-truth region-attribute annotations. As a result, we provide a rough estimate of image-text pairwise complexity; it is important to note that this value may fluctuate depending on the selection of attributes and regions. We use the provided structured labels in the dataset as the attribute set, and we determine appropriate locations for each label from the following set of regions: head, upper body, hands, lower body, and feet. We count the number of attribute-region paris in each image and determine the average pairwise complexity to be 7.9.

- *MIMIC-CXR*: Similar to DeepFashion, the MIMIC-CXR dataset does not provide ground-truth region-attribute annotations. As a result, we provide a rough estimate of pairwise complexity; again, it is important to note that this value may fluctuate depending on the selection of attributes and regions. In order to select attributes, we use RadGraph [23] to extract all entities from the reports (filtered to only include nouns). We then use a series of hand-crafted rules to parse sentences for location-related information, and we determine associations between identified entities and nine regions: left upper lung, left lower lung, left lung, right upper lung, right lower lung, right lung, heart, osseous structures, and stomach. We count the number of attribute-region pairs in each image and determine the average image-text pairwise complexity to be 5.0.

- *COCO*: We estimate pairwise complexity by using the ground-truth region annotations provided in the COCO training set. We count the number of ground truth region-attribute pairs in each image and determine the average image-text pairwise complexity of the COCO dataset to be 6.8.

C.2. Extended Description of the Mapping Model (Stage 1)

Decomposing Images and Text: We decompose each image x_i into r_i regions, expressed as $x_i = \{x_i^0, x_i^1, \ldots, x_i^{r_i}\}$. There are a variety of ways in which an image can be decomposed into regions, such as dividing images into equal-sized segments (*e.g.* quadrants) or using region proposal networks (RPNs). Note that we place no restrictions on the size or coverage of regions; regions can be of any size and can overlap. Ideally, regions should capture the key features in the image. We explore a variety of region selection methods across our four datasets, which we discuss in further detail below:

- *DocMNIST:* As described in Section 4, each image is composed of 9 equally-sized candidate regions of size $3 \times 28 \times 28$. We only consider regions with at least one assigned attribute.

- *DeepFashion:* Since the DeepFashion dataset consists of human images, we divide each image lengthwise into 4 equally-sized regions. Roughly, these regions correspond to the head, torso, legs, and feet; however, some variation is expected due to poses.

- *MIMIC-CXR:* We train a custom region proposal network to divide each image into 3 anatomic regions: right lung, left lung, and heart. To train the heart segmentation network, we used the JSRT dataset (`http://db.jsrt.or.jp/eng.php`) which contains 247 chest x-rays with heart segmentation masks. To train the lung segmentation network, we used the JSRT dataset (`http://db.jsrt.or.jp/eng.php`), which contains 247 chest x-rays with lung segmentation masks, and two additional datasets published by the U.S. National Library of Medicine [22, 4], which contain a total of 566 chest x-rays with lung segmentation masks. Each image in these datasets was preprocessed with the following operations, performed in sequence: normalization to the range [0,1]; resized to size (224, 224); histogram equalization. Both the lung and heart segmentation networks were trained with a batch size of 16 for 150 epochs using a UNet [38] architecture and a learning rate of 1e-4. We used random brightness contrast, gaussian blur, and affine transforms as augmentations during training. The predicted segmentation masks were postprocessed with the following operations, performed in sequence: binary opening operation with a disk (radius=5) structuring element; keeping only the largest contiguous predicted segment; binary fill holes; binary dilation with a disk (radius=5) structuring element.

- *COCO:* We use a pretrained RPN with identical settings to prior work [50]. The RPN generates 300 candidate regions for each image. For training the mapping model (stage 1), we select 20 regions from the set of 300 such that selected regions share minimal overlap. For training the VLM (stage 2), we randomly sample 100 regions from the set of 300.

Similarly, we decompose each textual description t_i into a_i attributes, expressed as $t_i = \{t_i^0, t_i^1, \ldots, t_i^{a_i}\}$. Dataset-specific implementation details are provided below:

- *DocMNIST*: As described in Section 4, the DocMNIST dataset includes a total of 20 attributes: zero, one, two, three, four, five, six, seven, eight, nine, purple, blue, green, yellow, red, rectangle, circle, small, medium, and large. We extract attributes in this set from the textual description.

- *DeepFashion:* We use the 58 provided structured labels (described in Section C.1) as our set of relevant attributes.

- *MIMIC-CXR:* We use RadGraph, an off-the-shelf entity extractor, to identify entities in each textual description [23]. RadGraph classifies each entity with one of three labels: Definitely Present, Uncertain, and Definitely Absent; we filter the set of identified entities to only include those that are Definitely Present. We then use the spaCy library to filter the set of entities to those that are nouns. Finally, we identify the 50 entities that occur most frequently in the training dataset as our final set of attributes.

- *COCO:* In line with with prior work, we extract 4.7k textual attributes (e.g. giraffe, man, bicycle, etc.) from the captions [50].

Representing Regions and Attributes: We generate embeddings for image regions using the procedure described in Section 5.1. As described in Section 5.1, the output of the image encoder is passed through a series of p projection heads, where each projection head consists of a linear layer, a ReLU function, and a second linear layer. For our experiments with DocMNIST, DeepFashion, and MIMIC-CXR, we set p to be equal to the total number of attributes in the dataset; as a result p equals 20, 58, and 50 respectively. We find that using a distinct projection heads for each attribute allows the model to learn fine-grained differences between similar attributes (*e.g.* distinguishing cotton and chiffon fabric in DeepFashion). This is particularly useful for real-world datasets, where images often exhibit high inter-class similarity [49]. However, using distinct projection heads for each attribute may be ineffective if (a) there are a large number of attributes in the dataset or (b) there is high diversity across images and attributes; in these cases, we find that models can effectively learn patterns with a fewer number of projection heads. Consequently, for our experiments with COCO, we use a single projection head ($p = 1$).

Additionally, since the COCO dataset is comprised of natural images and likely exhibits a similar distribution to the CLIP pretraining dataset, we find that an off-the-shelf CLIP model (with no additional tuning) can map regions to attributes with relatively high accuracy (as shown in Table 5). As a result, in order to capture the knowledge of the CLIP model, we compute the final embedding as a weighted average of (a) the input embedding provided to the projection head and (b) the output embedding generated by the projection head; this approach is similar to prior work on feature adapters [12]. Our approach generates more accurate region-attribute mappings than the original off-the-shelf CLIP model (by 4.9 points as shown in Table 5).

We generate embeddings for attributes using the procedure described in Section 5.1. We insert each attribute into pre-defined prompt templates and extract representations from a pretrained text encoder. Below, we provide dataset-specific implementation details:

- *DocMNIST:* We insert each attribute into the prompt templates defined in Section B.1. We represent each attribute as the average of its associated prompt embeddings.

- *DeepFashion:* We insert each attribute into the following pre-defined prompt templates, where components in square brackets are replaced with the attribute: {"The person wears a [attribute]", "There is a [attribute]", "The person is wearing [attribute]", "The man is wearing a [attribute]", "The woman is wearing a [attribute]", "The clothing is [attribute]"}. We represent each attribute as the average of its associated prompt embeddings.

- *MIMIC-CXR:* For each attribute, we identify all sentences in the training dataset that mention the attribute and then filter the list to include the 200 most frequently-occurring sentences. We represent each attribute as the average of its associated sentence embeddings.

- *COCO:* We insert each attribute into 81 prompt templates defined in a prior study [50]. Sample prompt templates include: {"A photo of a [attribute]", "A good photo of the [attribute]", "A photo of the small [attribute]"}. We represent each attribute as the average of its associated prompt embeddings.

Training Procedure: The mapping model is optimized with the following batch-wise loss function (where $L(x_i, t_i)$ is defined in Section 5.1).

$$L_B = \frac{1}{\sum_{j=1}^{|B|} |t_j|} \sum_{i=1}^{|B|} L(x_i, t_i)$$

We train the mapping model on a single NVIDIA A100 GPU with an initial learning rate of 1e-4. Dataset-specific training details are provided below, with hyperparameters selected based on mapping accuracy on a validation set as well as GPU memory constraints:

- *DocMNIST*: We use a batch size of 48 and train for 30 epochs. We set the loss temperature as $\tau = 0.07$.

- *DeepFashion*: We use a batch size of 48 and train for 20 epochs. We set the loss temperature as $\tau = 0.07$.
- *MIMIC-CXR*: We use a batch size of 48 and train for 5 epochs. We set the loss temperature as $\tau = 1$.
- *COCO*: We use a batch size of 24 and train for 5 epochs. We set the loss temperature as $\tau = 0.07$.

C.3. Extended Description of the VLM Model (Stage 2)

We use the mapping model from Section 5.1 to assign attributes to regions as follows. For DocMNIST, DeepFashion, and MIMIC-CXR, our approaches for selecting regions yield regions with zero, one, or multiple attributes. In order to account for a variable number of attributes that may be associated with each region, we use the following procedure to map attributes to regions. For a sample (x_i, t_i) and textual attribute $k \in t_i$, we compute the pairwise dot product between $P_k(e_i)$ and h_i^k, resulting in a score vector $v \in \mathbb{R}^{r_i \times 1}$. We then assign k to all regions with a score greater than $max(v) - \epsilon$, where ϵ is a pre-defined threshold. As desired, this procedure assigns zero, one, or multiple attributes to a region. We assign the value of ϵ by evaluating the quality of the region-attribute mappings on a validation set (details on this evaluation are provided in Appendix Section D.2). We select $\epsilon = 0.2$ for DocMNIST, $\epsilon = 0.2$ for DeepFashion, and $\epsilon = 0.1$ for MIMIC-CXR.

For COCO, we select candidate regions using an RPN, which generally yields tight bounding boxes that capture a single attribute. In this case, we invert the above procedure and instead assign regions to attributes. For a sample (x_i, t_i) and textual attribute $k \in t_i$, we compute the pairwise dot product between $P_k(e_i)$ and h_i^k. We repeat this computation for all textual attributes in t_i, resulting in a final score vector $v \in \mathbb{R}^{r_i \times a_i}$. We then assign each region to its highest-scoring attribute. As desired, this procedure assigns each region to a single attribute. Additionally, using this procedure ensures that our work is consistent with prior object detection models trained on the COCO dataset [50].

For each generated region-attribute pair, we replace the attribute with the segment of the original textual description containing the attribute; for example, the attribute "red" may be replaced with the sentence "The digit appears to be red" from the original description. We perform this step in order to allow for better reasoning over textual cues when training the VLM.

We then augment the training dataset to include generated region-attribute pairs in addition to the original image-text samples. We use the augmented dataset to train a one-to-one VLM. Below, we provide implementation details for each dataset.

- *DocMNIST*: The image encoder is initialized with weights from a pretrained CLIP ResNet-50. The text encoder takes the form of a pretrained CLIP text encoder with frozen weights. We optimize the VLM using a standard bidirectional contrastive loss function with the temperature parameter set as $\tau = 0.07$. We train the VLM on a single NVIDIA V100 GPU with an initial learning rate of 5e-5 and a batch size of 256.
- *DeepFashion*: The image encoder is initialized with weights from a pretrained CLIP ResNet-50. The text encoder takes the form of a pretrained CLIP text encoder with frozen weights. We optimize the VLM using a standard bidirectional contrastive loss function with the temperature parameter set as $\tau = 0.07$. We train the VLM on a single NVIDIA V100 GPU with an initial learning rate of 5e-5 and a batch size of 96.
- *MIMIC-CXR*: The image encoder is initialized with weights from a pretrained ConVIRT ResNet-50 using the implementation provided in ViLMedic [7]. The text encoder takes the form of a pretrained SBERT text encoder with frozen weights. We optimize the VLM using a standard bidirectional contrastive loss function with the temperature parameter set as $\tau = 1.0$. We train the VLM on a single NVIDIA V100 GPU with an initial learning rate of 5e-5 and a batch size of 196.
- *COCO*: We use the training framework and pretrained configurations provided by [50]. The image encoder is initialized with weights from a pretrained CLIP ResNet-50. The text encoder takes the form of a pretrained CLIP text encoder with frozen weights. The VLM is optimized using both a contrastive loss function and a distillation loss function, as implemented by [50]. We modify the contrastive loss function provided by [50] as follows: given a region-attribute pair, we expand the negative set to include *all* attributes, rather than just the other attributes in the batch. We train the VLM on 8 NVIDIA A100 GPUs with an initial learning rate of 0.002 and a batch size of 96.

D. Experimental Details

In this section, we provide additional details on our experiments.

D.1. Evaluation Tasks

Here, we provide further details on our three evaluation tasks: zero-shot object detection, text → region retrieval, and region → text retrieval.

D.1.1 Zero-Shot Object Detection

We compare ViLLA with three prior zero-shot object detection methods; we provide additional details on these methods below.

- *OVR-CNN* [45]: OVR-CNN is an object detector trained with both annotated region-category pairs as well as image-caption pairs. OVR-CNN is trained using 114k image-text pairs from the COCO dataset.
- *CLIP* [36]: CLIP is a VLM that leverages contrastive self-supervised learning to learn relationships between paired image-text samples. We use a CLIP model with a ResNet-50 backbone, which was trained on 400 million image-text pairs.
- *RegionCLIP* [50]: RegionCLIP was recently introduced for improving the fine-grained reasoning ability of CLIP. RegionCLIP uses the pretrained CLIP model in a zero-shot fashion to match candidate image regions with plausible textual attributes; these mappings are then used during training. We compare against two versions of RegionCLIP: (a) a version trained on 3 million image-text pairs from the Conceptual Captions (CC3M) dataset and (b) a version trained on 114k image-text pairs from the COCO dataset. Our approach is most comparable to version (b); however, this model is not made publicly available, resulting in some missing values in the corresponding row in Table 2.

D.1.2 Text → Region Retrieval

The text→region retrieval task evaluates the ability of a VLM to reason over fine-grained relationships between image regions and textual attributes. Given a textual query (*e.g.* "The person is wearing a hat"), the text→region retrieval task determines if we can retrieve image regions that capture the content of the query. Below, we provide implementation details for the text→region retrieval task on the DocMNIST and DeepFashion datasets:

- *DocMNIST*: We consider 20 textual queries for DocMNIST, with each query representing a distinct attribute from the following set: {zero, one, two, three, four, five, six, seven, eight, nine, purple, blue, green, yellow, red, rectangle, circle, small, medium, large}. We generate textual queries by inserting each attribute into the prompt templates provided in Appendix Section B.1.
- *DeepFashion*: We construct the text→region retrieval task by extracting 7.9k regions from the DeepFashion test set. The DeepFashion dataset includes human-parsing labels for a subset of the images, where various features of each image (*e.g.* glasses, hat, etc.) are labeled with pixel-level annotations in the image; we use the human-parsing labels to identify the ground-truth attributes in each region. We then construct 46 textual queries for DeepFashion, with each query representing a distinct attribute from the set listed in Section C.1. Although the original attribute set includes 58 attributes across 16 categories, we include only (a) attributes that occur at least once in the retrieval set of 7.9k regions and (b) attributes that are included in the provided human-parsing labels; this results in a total of 46 attributes across 14 categories. For the Precision@25 and Precision@100 metrics reported in Table 3, we further filter this list to only include attributes that occur at least 25 and 100 times in the retrieval set of 7.9k regions; this yields 40 and 25 attributes respectively. For each attribute, we construct a textual query by inserting the attribute into a custom prompt template. Prompt templates were designed to emulate linguistic patterns in the training set and are constructed based on attribute categories, as shown below (categories are italicized and attributes are in parentheses).
  - *glasses* (sunglasses): "The person wears sunglasses"
  - *hat* (yes): "The person wears a hat"
  - *lower color* (floral, pure color, graphic, lattice, color block, striped): "The lower clothing has [attribute] patterns"
  - *lower fabric* (leather, chiffon, knitted, cotton, denim): "The lower clothing is [attribute]"
  - *neckline* (square, lapel, round, standing, v-neck, suspenders): "The neckline is [attribute]"
  - *neckwear* (yes): "The person has neckwear"

  - *outer color* (floral, pure color, graphic, lattice, color block, striped): “The sweater has [attribute] patterns”
  - *outer fabric* (leather, chiffon, knitted, cotton, denim): “The outer clothing is [attribute]”
  - *ring* (yes): “The person is wearing a ring”
  - *socks* (leggings, socks): “The person wears [attribute]”
  - *upper color* (floral, pure color, graphic, lattice, color block, striped): “The upper clothing has [attribute] patterns”
  - *upper fabric* (chiffon, knitted, cotton, denim): “The upper clothing is [attribute]”
  - *waist accessory* (belt): “The person is wearing a belt”
  - *wrist accessory* (yes): “There is an accessory on the wrist”

We compare ViLLA with four baselines; we provide additional details on these methods below.

- *CLIP-ZS* [36]: The CLIP-ZeroShot (CLIP-ZS) baseline applies an off-the-shelf CLIP model to the text→region retrieval task in a zero-shot manner. We use a CLIP model with a ResNet-50 visual backbone. We encode each textual query using the CLIP text encoder; then, the regions with the highest dot-product similarities are identified.

- *CLIP-FT-Img*: The CLIP-FineTuned-Image (CLIP-FT-Img) baseline applies a fine-tuned CLIP model to the text→region retrieval task. We fine-tune a CLIP model with a ResNet-50 visual backbone on image-text pairs from the DocMNIST and DeepFashion training datasets. The model is optimized using a standard bidirectional contrastive loss function with a temperature of 0.07. We train on a single NVIDIA V100 GPU for 100 epochs with early stopping if the loss fails to decrease for 5 consecutive epochs. We use an initial learning rate of 5e-5 and a batch size of 256 for DocMNIST and 96 for DeepFashion. In order to perform text→region retrieval, we encode each textual query using the CLIP text encoder and identify regions with the highest dot-product similarities.

- *CLIP-FT-Reg*: The CLIP-FineTuned-Region (CLIP-FT-Reg) baseline applies a fine-tuned CLIP model to the text→region retrieval task. We fine-tune a CLIP model with a ResNet-50 visual backbone on both image-text and region-text pairs, where each region is aligned with the entire textual description. The model is optimized using a modified bidirectional contrastive loss function with a temperature of 0.07. Since each batch now includes multiple regions that share the same textual description, we modify the contrastive loss function such that for a given region or textual description, the negative set will not include any other regions or textual descriptions from the same image. We train on a single NVIDIA V100 GPU for 100 epochs with early stopping if the loss fails to decrease for 5 consecutive epochs. We use an initial learning rate of 5e-5 and a batch size of 256 for DocMNIST and 96 for DeepFashion. In order to perform text→region retrieval, we encode each textual query using the CLIP text encoder and identify regions with the highest dot-product similarities.

- *CLIP-ZS-Map* [36, 50]: The CLIP-ZeroShot-Mapping (CLIP-ZS-Map) baseline first applies CLIP in a zero-shot manner to generate region-attribute pairs (following the procedure in Section 5.2). We set $\epsilon = 0.1$ for DocMNIST and $\epsilon = 0.01$ for DeepFashion (details on the selection ϵ are provided in Appendix Sections C.3 and D.2). Then, a CLIP model with a ResNet-50 visual backbone is fine-tuned with image-text pairs as well as the generated region-attribute pairs. This baseline is comparable to RegionCLIP [50]. The model is optimized using a standard bidirectional contrastive loss function with a temperature of 0.07. We train on a single NVIDIA V100 GPU for 100 epochs with early stopping if the loss fails to decrease for 5 consecutive epochs. We use an initial learning rate of 5e-5 and a batch size of 256 for DocMNIST and 96 for DeepFashion. In order to perform text→region retrieval, we encode each textual query using the CLIP text encoder and identify regions with the highest dot-product similarities.

D.1.3 Region → Text Retrieval

Given an image region, the region→text retrieval task determines if we can retrieve the textual attributes depicted in the region. We follow a similar set-up to the text→region retrieval task detailed in Appendix Section D.1.2, and we compare ViLLA with the same baselines: CLIP-ZS, CLIP-FT-Img, CLIP-FT-Reg, and CLIP-ZS-Map. We note that for the DocMNIST dataset, a given region includes at most one attribute from each of the four categories (as defined in Section B.1); as a result, we retrieve at most one attribute from each of the four categories (digit, digit color, shape, and shape size). In Figure 3, we report the R-precision metric, which determines the percentage of the top-k retrieved attributes that are accurate; here, k is defined as the total number of ground-truth attributes in the region.

We additionally evaluate retrieval performance on the CheXpert 5x200 benchmark, which consists of 1000 chest X-rays across five disease categories (atelectasis, cardiomegaly, consolidation, pulmonary edema, and pleural effusion) [20]. The dataset is class-balanced, with 200 chest X-rays corresponding to each disease; additionally, the chest X-rays were selected such that no X-ray depicts more than one disease. Disease labels are converted into textual phrases using pre-defined prompts. Then, given a chest X-ray, the goal is to retrieve the textual phrase corresponding to the correct disease. In Table 1, we report retrieval accuracy (which is equivalent to a Precision@1 score) in line with prior work [20].

In order to convert disease labels into text, we use the same process that we use to generate attribute embeddings for MIMIC-CXR. For each disease label, we use the filtered set of RadGraph entities created in Section C.2 to identify all sentences in the training dataset that mention the disease; we then filter the list to include the 200 most frequently-occurring sentences. We represent each disease as the average of its associated sentence embeddings. If a disease label does not occur in the filtered RadGraph entity set, we instead represent the disease by encoding the following simple prompt: “patient with [disease]”.

Although the CheXpert 5x200 task has been previously considered as an image→text retrieval task, we instead formulate it as a region→text retrieval task when evaluating ViLLA. We do so by considering each X-ray as a set of four regions - right lung, left lung, heart, full image - and using ViLLA to generate embeddings for each region; then, we perform retrieval by computing the maximum pairwise similarity with the text phrases. Note that for our baselines, we follow the standard image→text retrieval approach in order to remain consistent with prior work [20].

We compare ViLLA with three baselines; we provide additional details on these methods below. Of these baselines, we note that ConVIRT and BioViL can be classified as one-to-one VLMs; however, GLoRIA is not a one-to-one VLM due to the use of a fine-grained local contrastive loss function.

- *ConVIRT* [49]: ConVIRT is a self-supervised, one-to-one VLM trained on chest X-rays and associated radiology reports. We use the implementation of ConVIRT provided in ViLMedic [7], which is trained on MIMIC-CXR.
- *BioViL* [3]: BioViL is a one-to-one VLM trained on chest X-rays and associated radiology reports. BioViL is trained using two self-supervised loss functions: (1) a standard image-text contrastive loss as well as (2) a masked language modeling loss to improve the quality of the text encoder. BioViL is trained on MIMIC-CXR.
- *GLoRIA* [20]: GLoRIA is a VLM trained on chest X-rays and radiology reports. GLoRIA leverages two self-supervised loss functions: (1) a standard image-text contrastive loss for global alignment between images and text as well as (2) a local contrastive loss for finer-grained alignment between image patches and words. GLoRIA is not a one-to-one model. We evaluate three variants of GLoRIA: (1) GLoRIA-Global Only, which represents an image with a single image-level embedding, (2) GLoRIA-Local Only, which uses all image patch embeddings, and (3) GLoRIA-Global+Local, which uses both the image-level embedding as well as all patch embeddings. We use the implementation of GLoRIA provided in ViLMedic [7], which is trained on MIMIC-CXR.

D.2. Extended Details on Evaluating Region-Attribute Mappings

In Section 6.3, we evaluated the quality of the region-attribute pairs generated using the mapping model. Here, we provide additional implementation details related to these evaluations.

As described in Section 5.2, we use our trained mapping model (Stage 1 of ViLLA) to generate region-attribute pairs. Our goal here is to quantitatively measure the number of generated region-attribute pairs that capture correct associations. However, measuring region-attribute quality requires ground-truth pairings, which are not always provided in real-world datasets. As a result, we generate ground-truth region-attribute pairings for each of our four pretraining datasets as follows:

- *DocMNIST*: Since DocMNIST is a synthetically-generated dataset, we have access to the ground-truth attributes associated with each region.
- *DeepFashion*: DeepFashion provides human parsing labels for a subset of the images, where various features of each image (*e.g*. glasses, hat, etc.) are labeled with pixel-level annotations in the image. We use our computed regions (four per image) and the human parsing labels to assign attributes to their ground-truth regions.
- *MIMIC-CXR*: MIMIC-CXR does not provide any region-level annotations. As a result, we estimate region-attribute quality by focusing only on two entities - cardiomegaly and pacemaker; these entities were selected due to their consistent association with a single region: the heart. We evaluate mapping accuracy using only this subset of two ground-truth region-attribute pairs. In the future, we aim to conduct user studies to better evaluate the quality of region-attribute mappings on datasets like MIMIC-CXR that lack region-level annotations.

- *COCO*: On the COCO dataset, we train our mapping model using candidate regions generated by a RPN; however, we do not have have access to ground-truth attribute pairings for these generated regions. As a result, we estimate region-attribute mapping quality by using the annotated, ground-truth regions in the COCO dataset. We evaluate the number of ground-truth regions that can be accurately mapped to their corresponding attributes.

For each image-text sample, we are given a set of generated region-attribute mappings as well as a set of ground-truth region-attribute pairs. A region-attribute pair is correct if it occurs in both sets. We compute *mapping precision* and *mapping recall* with the following formulas:

$$\text{mapping precision} = \frac{\text{number of correct region-attribute mappings}}{\text{number of generated region-attribute mappings}}$$

$$\text{mapping recall} = \frac{\text{number of correct region-attribute mappings}}{\text{number of ground-truth region-attribute pairs}}$$

As our final quality metric, we report the *mapping F1* score in Table 5, which is computed using the standard formula for F1 scores:

$$\text{mapping F1} = \frac{2 * \text{mapping precision} * \text{mapping recall}}{\text{mapping precision} + \text{mapping recall}}$$

We note here that there exists a tradeoff between mapping precision and mapping recall that is related to the choice of ϵ in Section 5.2. If ϵ is high, then ViLLA will generate a large number of region-attribute pairs, resulting in a low mapping precision and a high mapping recall. If ϵ is low, then ViLLA will generate a small number of region-attribute pairs, resulting in a high mapping precision and a low mapping recall. Empirically, as stated in Appendix Section C.3, we select the value of ϵ by computing mapping F1 scores on a validation set (if sufficient ground-truth region-attribute pairings are available). We note here that when ground-truth region-attribute pairings are limited (such as in the MIMIC-CXR dataset), we set ϵ to a default value of 0.1.

References

[1] Michael David Abramoff, Yiyue Lou, Ali Erginay, Warren Clarida, Ryan Amelon, James C. Folk, and Meindert Niemeijer. Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. *Investigative Opthalmology and Visual Science*, 57(13):5200, Oct. 2016. 2

[2] Amit Alfassy, Assaf Arbelle, Oshri Halimi, Sivan Harary, Roei Herzig, Eli Schwartz, Rameswar Panda, Michele Dolfi, Christoph Auer, Kate Saenko, PeterW. J. Staar, Rogerio Feris, and Leonid Karlinsky. Feta: Towards specializing foundation models for expert task applications, 2022. 3

[3] Benedikt Boecking, Naoto Usuyama, Shruthi Bannur, Daniel C. Castro, Anton Schwaighofer, Stephanie Hyland, Maria Wetscherek, Tristan Naumann, Aditya Nori, Javier Alvarez-Valle, Hoifung Poon, and Ozan Oktay. Making the most of text semantics to improve biomedical vision–language processing. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 1–21, Cham, 2022. Springer Nature Switzerland. 2, 12

[4] Sema Candemir, Stefan Jaeger, Kannappan Palaniappan, Jonathan P. Musco, Rahul K. Singh, Zhiyun Xue, Alexandros Karargyris, Sameer Antani, George Thoma, and Clement J. McDonald. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE Transactions on Medical Imaging*, 33(2):577–590, Feb. 2014. 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning (ICML)*, 2020. 1

[6] Patrick John Chia, Giuseppe Attanasio, Federico Bianchi, Silvia Terragni, Ana Rita Magalhaes, Diogo Goncalves, Ciro Greco, and Jacopo Tagliabue. Contrastive language and vision learning of general fashion concepts. *Scientific Reports*, 12(1), Nov. 2022. 3

[7] Jean-Benoit Delbrouck, Khaled Saab, Maya Varma, Sabri Eyuboglu, Jared A. Dunnmon, Pierre Chambon, Juan Manuel Zambrano, Akshay Chaudhari, and Curtis P. Langlotz. Vilmedic: a framework for research at the intersection of vision and language in medical ai. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, May 2022. 9, 12

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2

[9] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 3

[10] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. International Conference on Learning Representations (ICLR), 2022. 1

[11] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip), 2022. 1

[12] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters, 2021. 8

[13] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. Neural Information Processing Systems (NeurIps), 2022. 1

[14] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):E215–220, Jun 2000. 6

[15] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. 2022. 2

[16] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*, abs/1911.05722, 2019. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1, 4

[19] Danyang Hou, Liang Pang, Yanyan Lan, Huawei Shen, and Xueqi Cheng. Region-based cross-modal retrieval. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. 2

[20] Shih-Cheng Huang, Liyue Shen, Matthew P Lungren, and Serena Yeung. Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3942–3951, 2021. 2, 3, 6, 12

[21] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019. 2

[22] Stefan Jaeger, Alexandros Karargyris, Sema Candemir, Les Folio, Jenifer Siegelman, Fiona Callaghan, Zhiyun Xue, Kannappan Palaniappan, Rahul K. Singh, Sameer Antani, George Thoma, Yi-Xiang Wang, Pu-Xuan Lu, and Clement J. McDonald. Automatic tuberculosis screening using chest radiographs. *IEEE Transactions on Medical Imaging*, 33(2):233–245, 2014. 7

[23] Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven QH Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P. Lungren, Andrew Y. Ng, Curtis P. Langlotz, and Pranav Rajpurkar. Radgraph: Extracting clinical entities and relations from radiology reports, 2021. 6, 7

[24] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021. 1

[25] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2human: Text-driven controllable human image generation. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022. 5

[26] Alistair E W Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):1–8, 2019. 6

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, 2012. 1

[28] Benno Krojer, Vaibhav Adlakha, Vibhav Vineet, Yash Goyal, Edoardo Ponti, and Siva Reddy. Image retrieval from contextual descriptions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022. 2

[29] Juncheng Li, Xin He, Longhui Wei, Long Qian, Linchao Zhu, Lingxi Xie, Yueting Zhuang, Qi Tian, and Siliang Tang. Fine-grained semantically aligned vision-language pre-training, 2022. 2

[30] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, abs/2112.03857, 2022. 2

[31] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. *International Conference on Learning Representations (ICLR)*, 2023. 2

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 6

[33] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5

[34] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally?, 2023. 2

[35] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V. Le. Combined scaling for zero-shot transfer learning. *European Conference on Computer Vision*, abs/2111.10050, 2022. 1

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 1, 2, 4, 10, 11

[37] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection, 2022. 2

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 7

[39] Khaled Saab, Sarah Hooper, Mayee Chen, Michael Zhang, Daniel Rubin, and Christopher Re. Reducing reliance on spurious features in medical image classification with spatial specificity. *Machine Learning for Healthcare*, 2022. 2

[40] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection, 2020. 2

[41] Linda Wang, Zhong Qiu Lin, and Alexander Wong. COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. *Scientific Reports*, 10(1), Nov. 2020. 2

[42] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017. 2

[43] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher, 2021. 2

[44] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training, 2022. 2

[45] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14388–14397, 2021. 2, 10

[46] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *International Conference on Machine Learning*, 2022. 2

[47] Sheng Zhang, Yanbo Xu, Naoto Usuyama, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, Matthew P. Lungren, Tristan Naumann, and Hoifung Poon. Large-scale domain-specific pretraining for biomedical vision-language processing, 2023. 2

[48] Xujie Zhang, Yu Sha, Michael C. Kampffmeyer, Zhenyu Xie, Zequn Jie, Chengwen Huang, Jianqing Peng, and Xiaodan Liang. ARMANI: Part-level garment-text alignment for unified cross-modal fashion design. In *Proceedings of the 30th ACM International Conference on Multimedia*. ACM Conference on Multimedia, oct 2022. 3

[49] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text. *Machine Learning for Healthcare*, abs/2010.00747, 2022. 1, 2, 8, 12

[50] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16793–16803, June 2022. 2, 7, 8, 9, 10, 11