GLoRIA: A Multimodal Global-Local Representation Learning Framework for Label-efficient Medical Image Recognition (Supplementary Material)

Shih-Cheng Huang* Liyue Shen* Matthew P. Lungren Serena Yeung

1. GLoRIA Implementation Details

1.1. Data preprocessing

We use the CheXpert [7] dataset to train our representation learning framework. Radiology reports are not included as part of the original dataset, so we request them from the corresponding authors. We use all frontal chest radiographs in the dataset, which contain 191,229 image-text pairs in total. For each image, we resize the larger side to 256 pixels and use zero-padding on the smaller side to get the final image size of 256×256. All images are normalized to range [0, 1] before feeding into the model. For textual data, we use the impression section of the radiology reports which include detailed description of medical conditions such as subtypes, locations and severity. We preprocess all reports by picking out sequences of alphanumeric characters and drop all other characters and symbols. We ignore any samples with report containing less than 3 words. Next, we use the WordPiece tokenizer provided by BioClinicalBERT [11] to tokenize the input sentences. We set the maximum token length for the tokenizer to 97, which is the 95th percentile reports lengths in the training dataset.

1.2. Model implementation

Text encoder. We use the BERT [3] model initialized with weights from BioClinicalBERT [1], which is pretrained using reports from the MIMIC III dataset [8]. The BERT model contains 12 layers of self-attention encoders and a final prediction linear layer. We use implementations by the HuggingFace python library [15] for our BERT model. We sum outputs from the last four layers of the BERT model to represent the semantic meaning of each token. The feature matrix for each encoded report is indicated by $e \in \mathbb{R}^{768 \times T}$, where 768 is the dimension of the word vector and T is the number of tokens in the text. The i^{th} column e_i of the feature matrix is the embedding vector for the i^{th} word. We take the average of the word embedding vectors over all the T tokens as the sentence embedding vector $\tilde{e} \in \mathbb{R}^{768}$.

Image encoder We adopt the ResNet-50 architecture provided by the torchvision python library as our image encoder. The ResNet architecture includes 4 bottleneck blocks of residual convolutional layers and a linear classification layer. Using the preprocessed 256 by 256 pixels image, we extract the feature maps from the 3^{rd} bottleneck building block $f \in \mathbb{R}^{1024 \times 19 \times 19}$ of the ResNet model as the localized image sub-region features, which are reshaped to feature matrix $f \in \mathbb{R}^{1024 \times 361}$. 1024 corresponds to the dimension of the feature vectors while 361 is the number of image regions. The global image representation $\tilde{f} \in \mathbb{R}^{1024}$ is extracted from the final adaptive average pooling layer of the ResNet-50 model.

1.3. Training details

We project the global and local features from both modalities to a multimodal representation space with dimension $d = 768$. All linear layers for representation learning are initiated with uniform weight between -0.1 and 0.1. We set the temperature parameters $\tau_1 = 4.0$, $\tau_2 = 5.0$ and $\tau_3 = 10.0$ (Equation 1, 2, 6, 7 & 8). During representation learning, we use the Adam optimizer [9] with an initial learning rate of 5e-5 and weight decay of 1e-6. We set the learning rate scheduler to monitor the validation loss and anneal the learning rate by a factor of 0.5 after 5 epochs of plateau. We use batch size of 48 and 16 bit precision to fit the GPU during training. We set the maximum number of epochs to 50, and save the model checkpoint that achieves the lowest validation loss as the final model.

2. Classification Implementation Details

2.1. Data preprocessing

We use CheXpert and the RSNA Pneumonia datasets [14] for supervised classification and the CheXpert 5x200 dataset for zero-shot classification. For all 3 datasets, we resize the larger side of the image to 256 pixels and use zero-padding on the smaller side to get an image size of 256×256. All images are normalized before feeding in the model. For both CheXpert and CheXpert 5x200 dataset, we conduct classification on 5 categories, which are the competition tasks in the CheXpert challenge selected based on clinical importance and prevalence: (a) Atelectasis, (b) Car-

diomegaly, (c) Consolidation, (d) Edema, and (e) Pleural Effusion.

2.2. Model implementation

We use the ResNet-50 [6] architecture provided by the torchvision python library as our classification model. For **random** initialization, all the weights of ResNet model are randomly initialized. For **imagenet** initialization, the weights pretrained using ImageNet for classification are used to initialized the ResNet model. For all other methods, we use the pretrained weights from the representation learning step.

2.3. Training details

During training, we freeze all the weights from the ResNet-50 model except for the final classification layer. We use the Adam optimizer [9] with an initial learning rate of 1e-4 and weight decay of 1e-6. The classifier is trained using the Binary Cross Entropy loss function. We set our learning rate scheduler to monitor the validation loss and anneal the learning rate by a factor of 0.5 after 5 epochs of plateau. We use batch size of 64 and 16 bit precision to fit the GPU memory during training. We set the maximum number of epochs to 50, and save the model checkpoint that achieves the lowest validation loss. These hyper-parameters are tuned via a systematic search on 10% of the CheXpert dataset.

3. Segmentation Implementation Details

3.1. Data preprocessing

We use the SIIM Pneumothorax Dataset to train our segmentation model. Both the images and the segmentation masks are resized to 512x512 pixels. For augmentation on the training set, we apply ShiftScaleRotate provided by the albumentations python library, which includes random affine transforms of translation, scaling and rotation. We set the rotation limit to 10, scale limit to 0.1 and augmentation probability to 0.5. Images from both the training and validation set are normalized to range [0,1].

3.2. Model implementation

We use the UNet [13] architecture with ResNet-50 backbone implemented by the Segmentation-Models-PyTorch library. For **random** initialization, all weights are randomly initialized. For **imagenet** initialization, weights pretrained using ImageNet for classification are used to initialize the encoder portion of the UNet model. For all other methods, we use pretrained weights from the representation learning step to initialize the encoder.

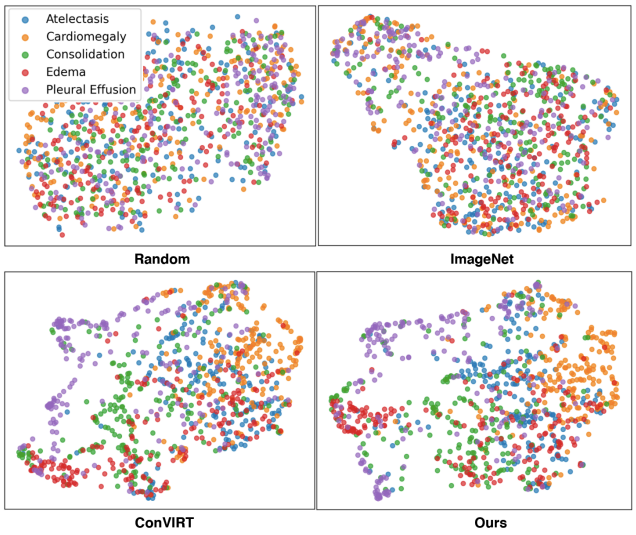

Figure 1. UMAP visualization of the global image representations for samples from the CheXpert 5x200 dataset using different pretrained weights.

3.3. Training details

During training, we use the Adam optimizer with an initial learning rate of 5e-4 and a weight decay of 1e-6. We set our learning rate scheduler to monitor the validation loss and anneal the learning rate by factor of 0.5 after 3 epochs of plateau. We use a combined loss of $\alpha * FocalLoss + DiceLoss$ and set $\alpha = 10$. Due to computation constraints, we train the segmentation model with a batch size of 4 and apply gradient accumulation for 8 batches. We set the maximum number of epochs to 100 and save the model checkpoints that achieve the highest validation dice score. The hyperparameters are set based on the model initialized with ImageNet pretrained weights.

4. Qualitative Results

4.1. Global embedding visualization

As a qualitative evaluation of the global representations learned via GLoRIA, we present the UMAP [12] plots of the global representations for all samples in the CheXpert 5x200 dataset in the Fig. 1. This allows us to understand how well our pretraining strategy help separate images from different classes in the representation space. The features are extracted from the vision encoder after the representation learning step is complete. More separated cluster for each category indicates that our method is able to learn better distinguishable semantic features. Compared to natural images, medical images have high visual similarities even for samples from different classes. This is due to the standardized protocols for medical image acquisition and the homogeneous nature of human anatomy. Furthermore, the

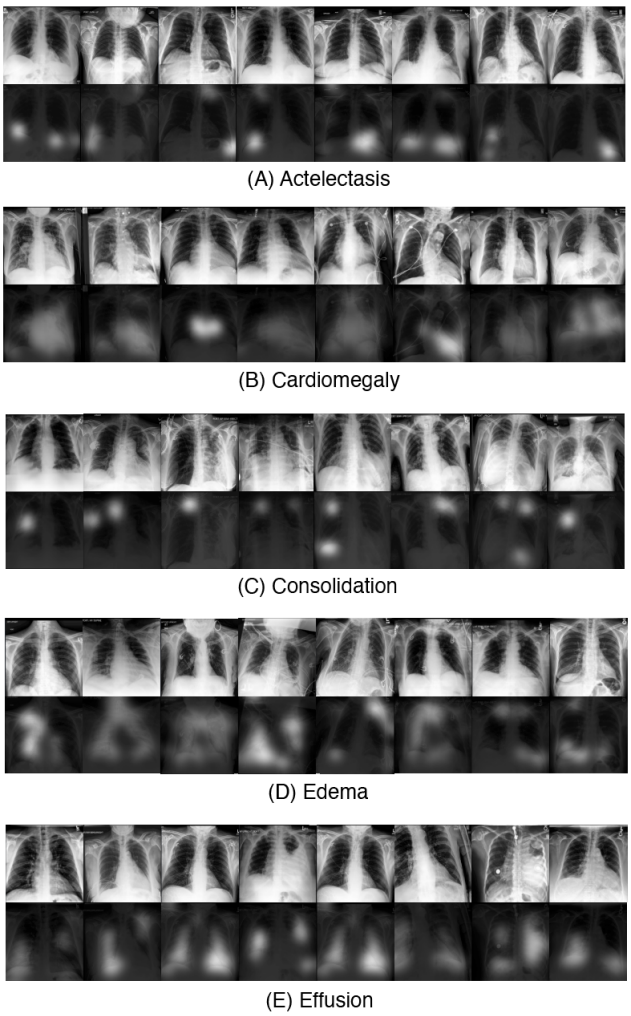

Figure 2. Examples of frontal chest radiographs of the chest (top) with corresponding attention weights for the given word (below).

CheXpert 5x200 is a subset of the CheXpert training set, which contains noisy labels resulting from the limitations of the CheXpert NLP labeler. Therefore, clustering for different categories using these images is a challenging task. From Fig. 1, it is clear that our pretrained model leads to more separable clusters with only global image representations, as compared to imagenet and randomly initiated models. Furthermore, since the global learning objective of our model uses the same loss function as ConVIRT, the global representations from our method and ConVIRT are expected to be comparable.

4.2. Visualization of attention weights

We visualize the attention weights (See Fig. 2) trained as part of our representation learning framework to qualitatively evaluate the localized features. Well-trained attention weights should correctly identify significant image regions corresponding to a given word. After reshaping and resizing the attention weights to match the input image size, we overlay the attention map on the original image for visualization. Fig. 2 demonstrates that our attention model is able correctly identify significant image-regions for key medical terms.

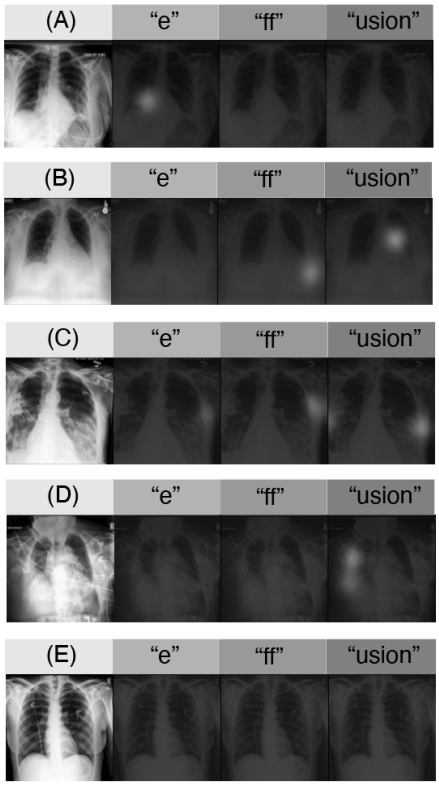

Figure 3. Examples of attention weights for each word piece ['e', 'ff', 'usion'] for the word "Effusion" on different frontal chest radiographs.

We also visualize the attention weights trained using our representation framework without using token aggregation aggregation strategy. Figure 3 show attention maps for Chest X-rays with visual signals for Effusion. Without token aggregation, we can see that the attention weights are scattered across the different word pieces, resulting in imprecise and incorrect attentions. Furthermore, from Figure 3A-E, we can see that the attention weights for effusion are arbitrarily attributed to different word pieces [*"e", "ff", "usion"*]. By applying token aggregation, we obtain reasonably localized attention weights corresponding to each word as shown in Fig. 2. This strategy also makes the model easier to learn and leads to better performance on downstream tasks as shown in the next section.

5. Ablation Study

We conduct ablation studies using two tasks: (1) image-text retrieval (Table. 1) and (2) zero-shot classification (Table. 2). We use the CheXpert 5x200 dataset for both tasks. In addition to comparing with different methods from prior works [4, 5, 16], we perform ablation experiments to inves-

Methods	Prec@5	Prec@10	Prec@100
DSVE [4]	40.64	32.77	24.74
VSE++ [5]	44.28	36.81	26.89
ConVIRT [16]	66.98	63.06	49.03
Ablation			
RNN Text Encoder	53.88	46.33	34.57
No Token Aggregation	54.48	49.63	38.31
Local Loss Only	51.66	43.40	30.99
Global Loss Only	66.98	63.06	49.03
Freeze Encoders	66.30	61.87	46.83
Similarity Type			
Global Similarity Only	67.02	64.68	49.55
Local Similarity Only	68.22	64.58	48.17
GLoRIA	**69.24**	**67.22**	**53.78**

Table 1. Ablation study on image-text retrieval

tigate and understand the proposed approach from multiple aspects. This includes comparisons for: i) text encoder type (Sec. 5.4), ii) the use of token aggregation (Sec. 5.6), and iii) training objectives (Sec. 5.5). For image-text retrieval, we also discuss the influence from different similarity metrics. For zero-shot classification, we compare different strategies for class prompts generation (Sec. 5.2).

5.1. Baselines

We compare GLoRIA to other representation learning methods in different domains, including medical imaging recognition [16] as well as natural image-caption retrieval [4, 5]. Recent state-of-the-art methods are based on localized representation learning, which often rely on pretrained object detection models (Bottom-up-attention) to extract localized images features. We use the same Bottom-up-attention model [2] pretrained with VisualGenome [10] on the CheXpert dataset to extract image sub-region features. In Figure 4, we show that bounding boxes proposed by the pretrained object detection model. We can see that the placement of bounding boxes are fairly similar between different images. This is not surprising since medical images have high visual similarity due to standardized protocols and the homogeneous nature of human anatomy. Furthermore, the majority of the bounding boxes are aggregated at the heart or beneath the lung region, which are unlikely locations for lung abnormalities. This suggests that using object detection models pretrained on natural images is not suitable for extracting informative regions of medical images. Thus, we do not include methods that require pretrained object detectors for comparison. Table 1 & 2 show that GLoRIA outperforms all other representation learning methods we compare with for both zero-shot classification and image-text retrieval.

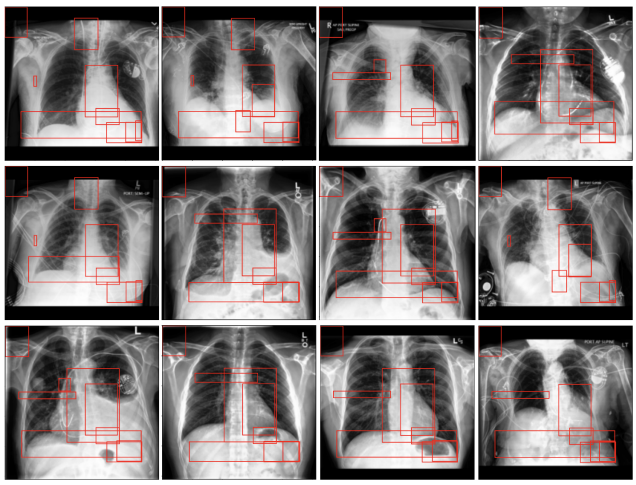

Figure 4. Examples of proposed regions of interested as bounding boxes using Bottom-up-attention pretrained with the VisualGenome dataset.

Methods	Acc.	Sens.	Spec.	PPV	NPV	F1
DSVE [4]	0.27	0.11	0.86	0.16	0.79	0.13
VSE++ [5]	0.31	0.20	**0.92**	0.38	0.82	0.26
ConVIRT [16]	0.56	0.43	0.90	0.50	0.86	0.46
Ablation						
RNN Text Encoder	0.52	0.62	0.85	0.51	0.90	0.56
No Token Aggregation	0.56	0.51	0.91	0.58	0.88	0.54
Local Loss Only	0.48	0.39	0.88	0.45	0.85	0.42
Global Loss Only	0.56	0.43	0.90	0.50	0.86	0.46
Prompt Type						
Class Name	0.51	0.68	0.86	0.54	0.91	0.60
Random Sample	0.56	0.54	0.90	0.57	0.89	0.56
GLoRIA	**0.61**	**0.70**	0.91	**0.65**	**0.92**	**0.67**

Table 2. Ablation study on zero-shot classification

5.2. Prompt generation

We experiment with different approaches for class prompts generation (See Fig. 5). The use of text prompts for each class not only enables zero-shot image classification by framing it as an image-text similarity task, but also helps provide domain knowledge as context for each class label. This is especially important for medical imaging since different sub-types or severity of an abnormality can have drastically different visual appearance yet labeled as the same medical condition. We experiment with the following prompt engineering approaches.

- **Class Name**: This baseline approach simply represents each class by its class name as string. For example, the category *Cardiomegaly* is represented as a string *"Cardiomegaly"*.
- **Random Sample**: For each class we are predicting, we randomly sample sentences from the reports in the

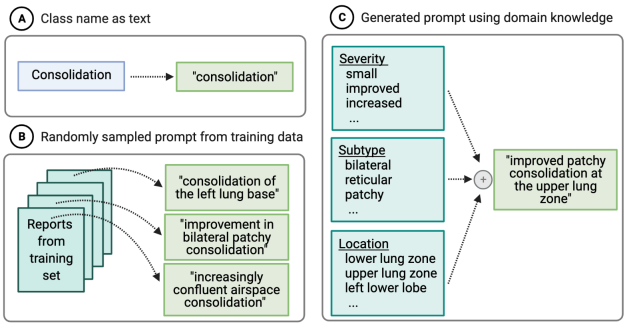

Figure 5. Methods for generating class prompt. (A) Class names are simply represented as text. (B) Sentences from the reports with the same class are sampled from the training dataset. (C) For each classification category, prompts are generated by combining possible severity, sub-types and locations for that medical condition using domain knowledge.

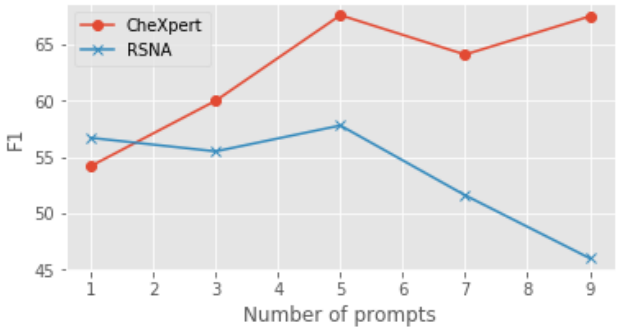

Figure 6. Zero-shot classification using different number of generated prompts.

training dataset with the same class label. Predictions are ensembled from each of the sampled text.

- **Generated Prompts**: In this setting, we generate prompts for each class label using medical domain knowledge. For each abnormality we are predicting, we consider all possible combinations of severity, abnormality sub-types and locations (see Fig. 5). Predictions are based on an ensemble of N randomly generated prompts.

As shown in Table 1 & 2, generating prompts using domain knowledge achieved significantly better predictions as compared to using class name as text or random sampling.

5.3. Number of class prompts

We generate multiple text prompts for each class during zero-shot classification as an ensemble strategy. The final classification results are obtained by taking the average similarities between the input image and all prompts within the same class. Figure 7 shows the performance for both CheXpert and RSNA based on different number of class prompts. We use the CheXpert 5x200 dataset to determine the optimal number of prompts to use. We see that the F1 score increases with the number of prompt and plateaus after 5 prompts, which indicates that there is a ceiling for performance with additional prompts. Therefore, we use 5 text prompts for our zero-shot classification tasks.

5.4. Text encoder

Many state-of-the-art image-text representation learning methods use Recurrent Neural Networks (RNNs) as their text encoder. While RNNs are successful for natural image-caption datasets with short and precise textual descriptions, medical reports typically consist of long paragraphs and require reasoning across multiple sentences. Therefore, we propose to use a self-attention based model (BERT) as our text encoder. We find that using self-attention based model as text encoder outperforms RNN for both tasks (Table 1 & 2).

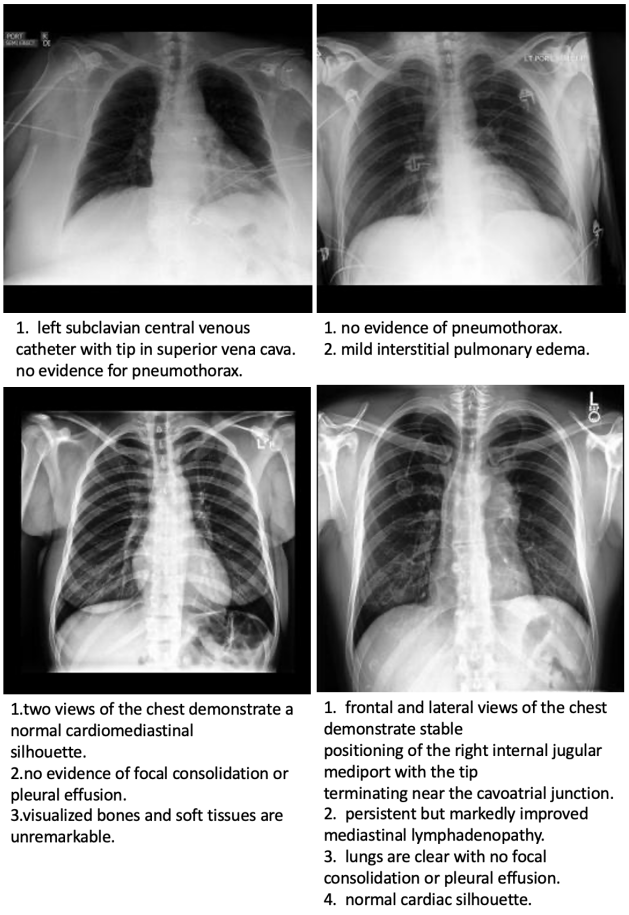

Figure 7. Examples of X-ray and report pairs from the dataset used in this work.

5.5. Training objective

Our representation learning training objective consists of two parts, a global contrastive loss $L_g^{(t|v)} + L_g^{(v|t)}$ and a local contrastive loss $L_l^{(t|v)} + L_l^{(v|t)}$. When we only use global contrastive loss to train our representation learning framework, our training objective is exactly the same as ConVIRT [16]. Table 1 and 2 show that using both losses leads to better performance as compared to using either global or local loss alone.

5.6. Token aggregation

To account for typographical errors and abbreviations common in medical reports, we use the WordPiece tokenizer to tokenize the input text reports. However, as shown in Fig. 3, this leads to attention weights that scatter across each word piece when learning localized representations. Therefore, we use a token aggregation strategy to combine the features from each word piece for each word. We find that aggregating tokens lead to a big increase in performance, 0.06 increase in F1 score for Zero-shot classification (Table 2) and 14.76 Prec@5 improvement for retrieval (Table 1).

References

[1] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3984–3993, 2018.

[5] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[8] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[12] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[14] George Shih, Carol C Wu, Safwan S Halabi, Marc D Kohli, Luciano M Prevedello, Tessa S Cook, Arjun Sharma, Judith K Amorosa, Veronica Arteaga, Maya Galperin-Aizenberg, et al. Augmenting the national institutes of health chest radiograph dataset with expert annotations of possible pneumonia. *Radiology: Artificial Intelligence*, 1(1):e180041, 2019.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[16] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.