Reproduction Project Final Report: Contrastive Learning of Medical Visual Representations from Paired Images and Text

Junyoung Lee (jl298@illinois.edu)

May 2025

1 Abstract

This report details the reproduction of Zhang et al.'s ConVIRT framework[ZJM+22], motivated by the need to validate its applicability on MIMIC-CXR-JPG[MCJ] dataset. Using PyHealth toolkit, I re-implemented the encoder pipelines, evaluating performance on RSNA[WPL+17], CheXpert[IRK+19], MURA[RIB+18], and COVIDx[WW20] classification tasks. I observed varying performance gaps compared to the original paper, with closest reproduction in fine-tuning scenarios using 10% training data-particularly for CheXpert (1.1% difference), COVIDx (1.5% difference) and MURA (5.7% difference). The implementation achieved 87.0% AUC on CheXpert fine-tuning with 10% data, 88.8% accuracy on COVIDx fine-tuning, and 81.0% AUC on CheXpert with just 1% labeled data, demonstrating the method's data efficiency despite training for only 40 epochs instead of the original 200. For Zero-shot retrieval, the reproduction achieved 26.7% Precision@5 for image-image retrieval and 29.0% Precision@5 for text-image retrieval, significantly outperforming random baselines but not matching the original paper's results. The 40-epoch training showed decent convergence with approximately 16% improvement in training loss. **GitHub repository**: https://github.com/jl298/repro-convirt, **Presentation video**: https://github.com/jl298/repro-convirt/blob/main/doc/presentation.mp4

2 Introduction

Medical image analysis remains challenging because expert annotations are limited and visual differences can be very subtle. Previous methods depend on costly expert labeling or rule-based report extraction, each with notable drawbacks. Zhang et al. introduced ConVIRT, employing a bidirectional InfoNCE contrastive objective on approximately 217k chest X-ray–report pairs from MIMIC-CXR-JPG to align visual and textual embeddings. This pretraining yields a 6.3 point AUC boost over ImageNet-pretrained ResNet-50 in CheXpert classification under the linear setting (87.3% vs. 81.0%). In this reproduction, I aim to:

1. Implement the ConVIRT framework using PyHealth[PyH].
2. Replicate the key CheXpert experiments in both linear and fine-tuning settings.
3. Verify the data-efficiency claims of the original work.

Dataset The implementation fairly reproduces the core dataset processing components from the original paper.

Scope	Item
In	• The code includes preprocessing pipelines for MIMIC-CXR dataset, handling both the image and text modalities • The image transformation pipeline is implemented with all key components mentioned in the paper: random cropping, horizontal flipping, affine transformation, brightness/contrast adjustments, and Gaussian blur • Text sampling from reports is implemented, with the ability to extract Findings and Impression sections • The preprocessing follows the original paper's specifications for resizing images to have 256 pixels on the longer edge
Out	• The bone image dataset referenced (Rhode Island Hospital dataset) was omitted; this work focuses solely on chest X-rays

Model The ConVIRT model architecture is well-reproduced.

Scope	Item
In	• The core components are implemented: ResNet50 image encoder, BERT text encoder (initialized with ClinicalBERT weights), non-linear projection heads for both modalities • The bidirectional contrastive learning objectives (image-to-text and text-to-image) using InfoNCE loss formulation are correctly implemented • The hyperparameters for training that match the paper's specifications, such as temperature parameter ($\tau = 0.1$) and loss weight ($\lambda = 0.75$) • The projection head architecture with a single hidden layer and ReLU activation follows the paper's description
Gap	• Dynamic learning rate scheduling was implemented using PyTorch's ReduceLROnPlateau scheduler, which may introduce slight deviations from the original approach • Minor deviations in batch construction and normalization methods were noted compared to the original implementation details.

Baselines The baseline implementations are not mostly reproduced, instead citing the numerical values reported in the paper.

Scope	Item
Partially	• The evaluation on the Zero-shot retrieval tasks (image-image and text-image retrieval) is implemented for the main model but not comprehensively for all baselines
Out	• The ImageNet initialization baseline is not implemented • Random initialization baseline is not implemented • The captioning-based baselines (Caption-LSTM and Caption-Transformer) are not implemented with all the details from the original paper • The image-only contrastive learning baselines (SimCLR and MoCo v2) mentioned in the analysis section of the paper are not implemented • The contrastive-binary-loss baseline is not implemented, including the MLP classification head for binary prediction

3 Methodology

3.1 Environment

OS	Ubuntu 22.04.5 on WSL2
Language	Python 3.8.20
Packages	PyHealth 1.1.14, PyTorch 1.8, pandas, matplotlib, numpy, scikit-learn, transformers, etc.

3.2 Data

The datasets used in the reproduction study can be accessed through the URLs listed below. Each dataset can be downloaded directly from its respective source repository by following the provided links.

Dataset	Download URL
MIMIC-CXR-JPG	https://physionet.org/content/mimic-cxr-jpg/2.1.0/
RSNA	https://kaggle.com/competitions/rsna-pneumonia-detection-challenge/
CheXpert	https://stanfordmlgroup.github.io/competitions/chexpert/
COVIDx	https://kaggle.com/datasets/andyczhao/covidx-cxr2
MURA	https://stanfordmlgroup.github.io/competitions/mura/

I utilized five standard medical imaging datasets: MIMIC-CXR-JPG for chest radiographs with paired reports, RSNA for pneumonia detection, CheXpert for multi-label classification, COVIDx for COVID-19 classification, and MURA for musculoskeletal abnormality detection. The detailed key characteristics of each dataset are listed in Table 1.

Characteristic	MIMIC-CXR-JPG	RSNA	CheXpert	COVIDx	MURA
Image Type	Chest X-rays	Chest X-rays	Chest X-rays	Chest X-rays	Musculoskeletal X-rays
Images / Pairs	∼217k pairs	∼26k images	∼224k images	∼14k images	∼40k images
Purpose	Pretraining	Evaluation	Evaluation	Evaluation	Evaluation
Task Type	Contrastive Learning	Binary Classification	Multi-label Binary Classification	Multi-class Classification	Binary Classification
Classes / Labels	N/A (Paired text)	Normal vs Pneumonia	Atelectasis, Cardiomegaly, Consolidation, Edema, Pleural Effusion	COVID-19, non-COVID pneumonia, normal	Normal vs Abnormal
Training Split	∼212k pairs	25,184 images	218,414 images	13,598 images	33,078 images
Validation Split	∼5k pairs	1,500 images	5,000 images	300 images	3,730 images
Test Split	N/A	3,000 images	234 images (expert-labeled)	300 images	3,197 images
File Format	JPG	DICOM	JPG	PNG/JPG	PNG

Table 1: Statistics of Medical Imaging Datasets

3.3 Model

Figure 1 illustrates the ConVIRT[ZJM+22] pretraining pipelines.

Image pipeline In the image pipeline, each chest X-ray undergoes augmentations-resizing/cropping to 224×224, flips, slight rotations/translations, brightness/contrast tweaks, and Gaussian blur-to create varied views $\tilde{x}_v$ for invariance learning. That augmented image is then passed through a ResNet-50 backbone f_v, which applies convolutional layers and global average pooling to produce a 2048-dimensional feature vector $h_v = f_v(\tilde{x}_v)$. This feature passes through a two-layer MLP projection head g_v, linear → ReLU → linear, to produce a 512-dimensional vector $v' = g_v(h_v)$. Finally, it normalizes v' by its norm

$$v = \frac{v'}{\|v'\|_2}$$

, producing the unit-norm 512-dimensional image embedding v.

Text pipeline In the text pipeline, each radiology report is tokenized into WordPiece subword IDs $\{w_1, \ldots, w_T\}$ and passed through a ClinicalBERT encoder f_u. That transformer produces per-token

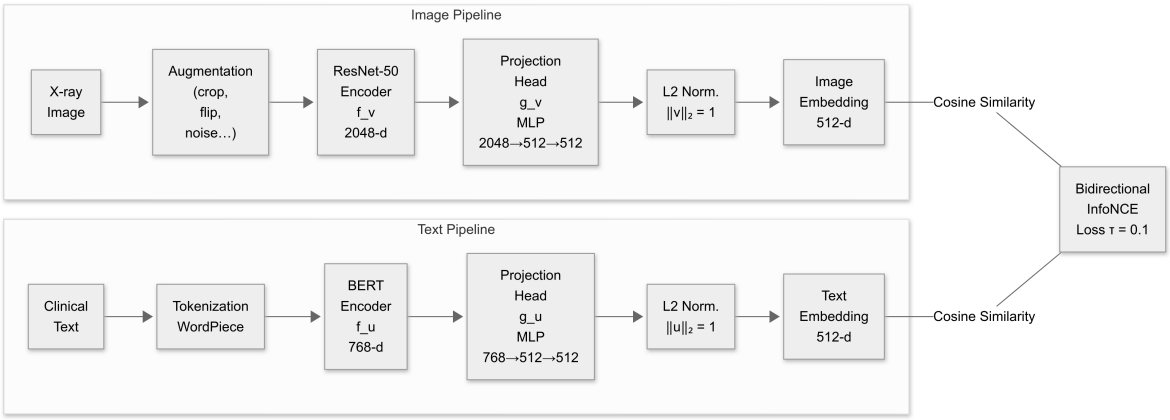

Figure 1: Image and text encoder pipelines utilized during pretraining

embeddings $H_i \in \mathbb{R}^{768}$, which are aggregated by taking the element-wise maximum across all positions

$$h_u = \max_{i=1\ldots T} H_i.$$

This 768-dimensional pooled feature is then projected into the same 512-dimensional space by another two-layer MLP g_u, analogously applying a linear layer, ReLU, and a second linear layer to obtain $u' = g_u(h_u)$. A final L2 normalization

$$u = \frac{u'}{\|u'\|_2}$$

ensures the text embedding u also has unit norm, making it directly comparable to the image embedding.

Contrastive objective Once both modalities produce unit-norm vectors v and u, their alignment is learned via the bidirectional InfoNCE loss. For a batch of N paired examples, the cosine similarity $s(v_i, u_j) = v_i^\top u_j$ is computed and scaled by a temperature $\tau = 0.1$. The loss for each pair (v_i, u_i) simultaneously attracts the true pair and repels mismatches

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\Big[-\log\frac{e^{s(v_i,u_i)/\tau}}{\sum_k e^{s(v_i,u_k)/\tau}} \;-\; \log\frac{e^{s(u_i,v_i)/\tau}}{\sum_k e^{s(u_i,v_k)/\tau}}\Big].$$

Minimizing this loss clusters true image–text pairs on the hypersphere and separates all other pairings.

The final pretrained checkpoint contains the learned weights for f_v (ResNet-50 backbone), g_v (vision projection head), f_u (BERT encoder), and g_u (language projection head).

3.4 Training

Hyperparameter The paper provides reasonably detailed descriptions of the hyperparameters used for training, so most of them were used exactly as specified in the paper for this reproduction work. Table 2 lists the values of the hyperparameters used for pretraining or evaluations.

Computational requirements The original ConVIRT pretraining leveraged a single NVIDIA Titan RTX GPU with mixed-precision training and a fixed batch size of 32 for efficient memory use and stable gradients. The full MIMIC-CXR pretraining run spanned roughly three days, about 72 GPU-hours, across 200 epochs, averaging close to 21.6 minutes per epoch.
In the reproduction environment, a single NVIDIA GeForce RTX 4060 Ti was used, with each epoch taking approximately 97.43 minutes. Since the full 200 epochs would require over 13 days, I limited the training to 40 epochs to shorten the development and experiment cycle.

Hyperparameter	Original	Modified
batch size	32	-
epochs	200	40
learning rate(Pretraining)	1e-4(half-on-plateau)	1e-4(with PyTorch scheduler)
learning rate(Classification:COVIDx)	1e-3(half-on-plateau)	1e-3(with PyTorch scheduler)
learning rate(Classification:Others)	1e-4(half-on-plateau)	1e-4(with PyTorch scheduler)
weight decay	1e-6	-
temperature	0.1	-
lambda weight	0.75	-
projection dimension	512	-
image size	224	-

Table 2: Key hyperparameters for pretraining and evaluations

Data augmentation The *transforms.py* file implements the data augmentation strategies for the ConVIRT approach, which learns joint representations of medical images and text through self-supervised contrastive learning. In ConVIRT, robust image augmentation is the core part for creating diverse views of the same image to form positive pairs for contrastive training. The training transforms include RandomResizedCrop (scale 0.6-1.0), RandomHorizontalFlip (p=0.5), RandomAffine (rotation $\pm 20°$, translation 0.1, scale 0.95-1.05), ColorJitter (brightness/contrast 0.4), and GaussianBlur (σ=0.1-3.0).

For downstream classification tasks, the implementation differs from pretraining by introducing zero-padding to make images square before resizing to 224×224, as specified in the paper. This preprocessing ensures consistent aspect ratios across different medical image datasets. The classification training transforms maintain the same augmentation pipeline as pretraining but apply them after zero-padding. The validation transforms for classification only apply zero-padding and resize operations without augmentation, following ConVIRT's evaluation protocol.

The retrieval evaluation tasks use the same transform pipeline as classification validation, applying zero-padding to square followed by resize to 224×224. This consistency ensures that the pretrained encoder sees images processed in the same way across different downstream tasks, maintaining the learned feature representations effectively.

Dynamic learning rate The implementation uses PyTorch's ReduceLROnPlateau scheduler to adjust the learning rate dynamically based on validation loss. The learning rate strategy starts with an initial value(Table 2) and automatically adjusts based on validation performance. The scheduler provides adaptive learning rate decay, reducing the rate by half whenever validation loss plateaus for 5 epochs(3 epochs for classification).

3.5 Evaluation

Baseline comparison strategy In the ConVIRT study, two standard initialization baselines are used to evaluate performance. The first approach is random initialization. All weights of the ResNet-50 encoder are set to default random values before training. The second is ImageNet initialization. The ResNet-50 encoder is initialized with weights pretrained on the ImageNet dataset, leveraging features learned from over a million natural images. In the reproduction evaluation, I directly cite the Random Init. and ImageNet Init. metrics as reported in Table 1 of Zhang et al. (2022)[ZJM+22], ensuring exact alignment with the original study's baseline figures for both linear classification and fine-tuning settings.

Downstream classification The classification evaluation, depicted as a diagram in Figure 2, follows a systematic process to assess the pretrained ConVIRT encoder's performance on medical image classification tasks. First, the model removes projection heads, which returns only the base ResNet50 backbone without the projection layers used during pretraining. This encoder is then evaluated in two distinct modes: linear evaluation (frozen encoder) and fine-tuning (full network). In linear mode, the encoder weights remain frozen, and only a newly attached linear classification head is trained. Fine-tuning mode begins with a 200-step warm-up phase where only the classifier head trains, followed

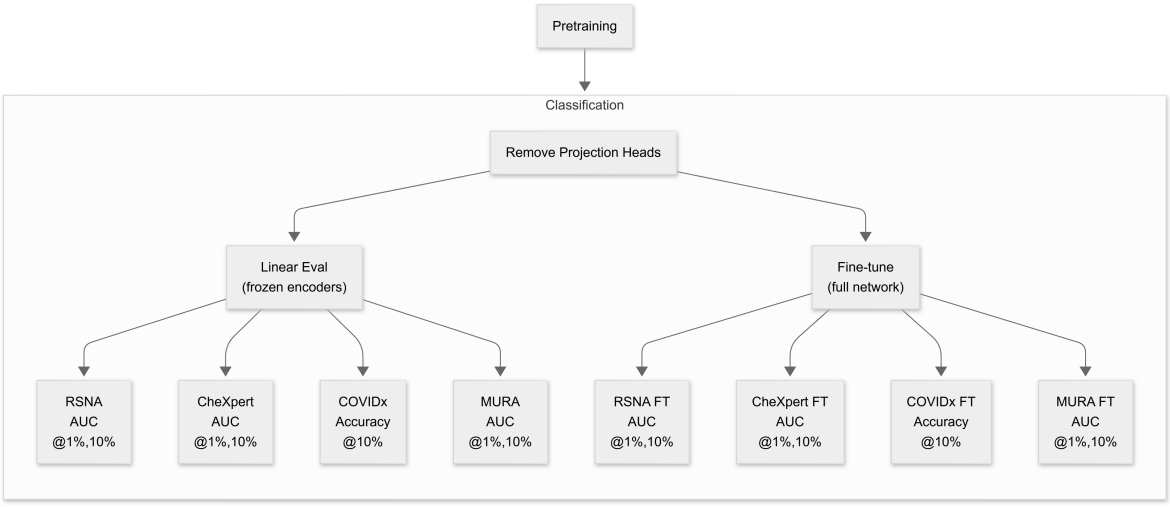

Figure 2: Pipeline for downstream classification evaluation

by unfreezing and jointly optimizing both encoder and classifier. Evaluation is conducted on four medical imaging datasets (RSNA Pneumonia, CheXpert, COVIDx, and MURA) using varying data percentages (1%, 10%). Performance metrics are task-specific: AUC for binary tasks (RSNA, MURA) and multi-label tasks (CheXpert), and accuracy for multi-class tasks (COVIDx). Training uses Adam optimizer with early stopping based on validation performance (patience of 10 epochs), saving the best model checkpoint for final test evaluation.

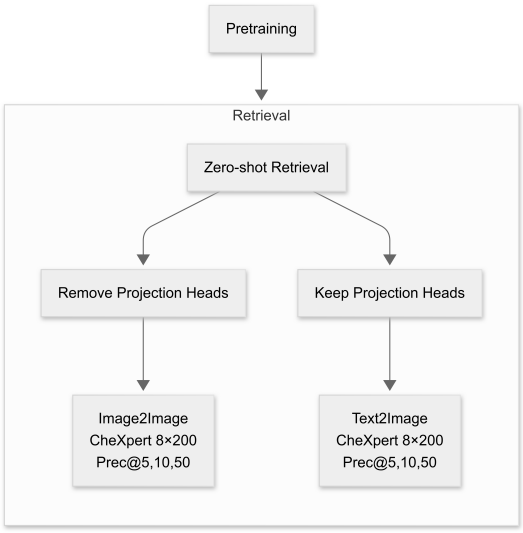

Figure 3: Pipeline for downstream zero-shot retrieval evaluation

Downstream Zero-shot retrieval The retrieval evaluation, depicted as a diagram in Figure 3, measures the pretrained ConVIRT model's ability to match queries to candidates without any task-specific training. The process follows the paper's specification by removing projection heads from the image encoder for image-to-image retrieval. For text-to-image retrieval, the paper does not explicitly mention removing projection heads, so the text encoder retains its projection head to maintain dimensional alignment between text and image representations. For image-to-image retrieval, both query and candidate images are processed through the base ConVIRT image encoder (ResNet50 backbone) without projection heads. The encoder outputs are L2-normalized to create feature embeddings. For text-to-image retrieval, text queries are processed through the full text pipeline (BERT encoder +

projection head) while candidate images use only the base image encoder. This asymmetric architecture allows matching between the projected text space and the base image feature space. The model computes cosine similarity between query and candidate embeddings, ranks candidates by similarity scores, and evaluates using Precision@k metrics (k=5, 10, 50). Testing uses the CheXpert 8x200 dataset containing 8 abnormality categories, each with 10 query images/5 text queries and 200 candidate images. All evaluation is performed Zero-shot - the model directly leverages its pretrained representations without any retrieval-specific fine-tuning.

4 Results

The results from this reproduction study show that ConVIRT effectively learns meaningful medical visual representations. Table 3 presents the performance comparison between our implementation and the original paper's reported results across four classification tasks.

Dataset	Mode	Training	Metric	Paper	Repro	Diff
RSNA	Linear	1%	AUC	90.7	77.8	-12.9
RSNA	Linear	10%	AUC	91.7	82.5	-9.2
RSNA	Finetune	1%	AUC	88.8	77.8	-11.0
RSNA	Finetune	10%	AUC	91.5	82.7	-8.8
CheXpert	Linear	1%	AUC	85.9	77.5	-8.4
CheXpert	Linear	10%	AUC	86.8	79.8	-7.0
CheXpert	Finetune	1%	AUC	87.0	81.0	-6.0
CheXpert	**Finetune**	**10%**	**AUC**	**88.1**	**87.0**	**-1.1**
COVIDx	Linear	10%	Accuracy	85.9	66.3	-19.6
COVIDx	**Finetune**	**10%**	**Accuracy**	**90.3**	**88.8**	**-1.5**
MURA	Linear	1%	AUC	81.2	46.4	-34.8
MURA	Linear	10%	AUC	85.1	60.0	-25.1
MURA	Finetune	1%	AUC	81.3	52.4	-28.9
MURA	**Finetune**	**10%**	**AUC**	**86.5**	**80.8**	**-5.7**

Table 3: Classification performance comparison between original paper and reproduction

The repro 40-epoch trained model achieves reasonable but lower performance than the original 200-epoch model. The gap is smallest in fine-tuning scenarios with 10% training data, particularly for CheXpert (1.1% difference), COVIDx (1.5% difference) and MURA (5.7% difference). The largest discrepancies occur with MURA linear evaluation at 1% data (34.8% gap), likely due to the fact that the implementation did not use the Rhode Island Hospital dataset for bone image pretraining as mentioned in the original paper.

	Image-Image Retrieval			Text-Image Retrieval		
Method	Prec@5	Prec@10	Prec@50	Prec@5	Prec@10	Prec@50
Random (Paper)	12.5	12.5	12.5	12.5	12.5	12.5
ImageNet (Paper)	14.8	14.4	15.0	-	-	-
ConVIRT (Paper)	45.0	42.9	35.7	60.0	57.5	48.8
ConVIRT (Repro)	26.7	26.0	24.2	29.0	24.5	23.4

Table 4: Zero-shot retrieval performance on CheXpert 8×200 dataset

Repro implementation demonstrates effective Zero-shot retrieval capabilities, significantly outperforming random and ImageNet baselines. However, a substantial gap remains compared to the original paper's results. The text-image retrieval shows a larger performance differential than image-image retrieval, suggesting the cross-modal alignment requires more extensive training than our 40 epochs provided.

The loss graph in Figure 4 shows a gradual improvement over 40 epochs with both training and validation losses decreasing from 3.45 to 2.9. A significant drop occurs around epochs 9-12, which appears to be directly influenced by the learning rate change from 0.0001 to 5e-05. The train-validation

gap widens over time (from 0.02 to 0.10), presumably indicating mild overfitting. After the initial improvement, loss changes stabilize in later epochs with smaller incremental gains. The distribution of loss changes is concentrated near zero in later epochs, confirming convergence slowdown. Overall, the model shows healthy learning with approximately 16% improvement in training loss and 13.5% improvement in validation loss.

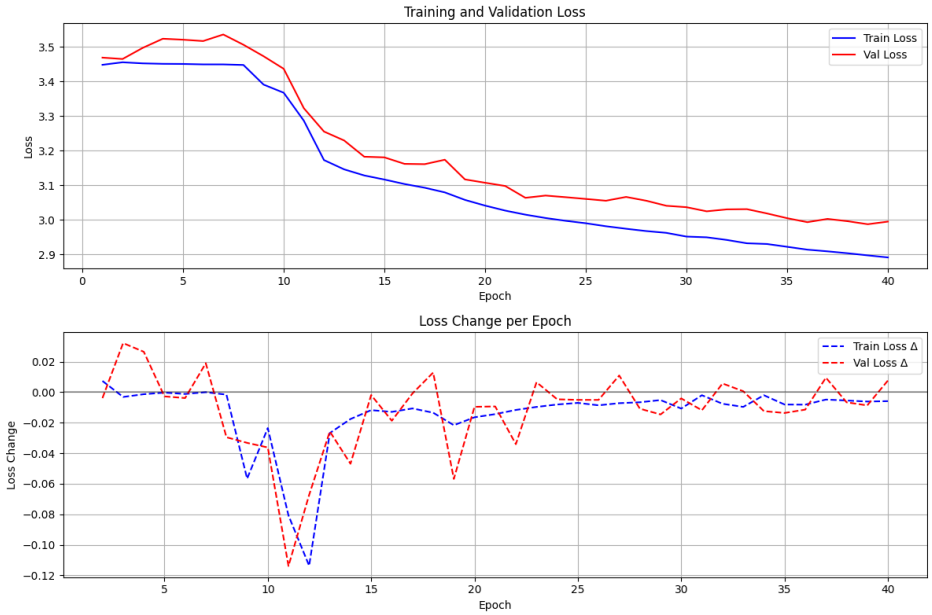

Figure 4: Training and Validation Loss Progression During Model Training

5 Discussion

The ConVIRT reproduction shows the original paper is largely reproducible. While the simplified repro implementation achieved better performance than Random, ImageNet baseline, some performance gaps persist compared to the original results. With more computational resources and further clarification from the original authors regarding undocumented implementation details, achieving the original performance is expected to be feasible. Despite these challenges, it is worth noting that the paper provided sufficient details for core components - hyperparameters, loss functions, and model architecture were well-documented. The bidirectional contrastive loss, temperature, and projection dimension specifications were particularly helpful. However, computational limitations due to GPU resources, coupled with limited time to explore implementation nuances, restricted the scope of the training experiments.

Moving forward from this reproduction study, several enhancements based on recent developments in the field could be proposed. Building upon ConVIRT's foundation, we can advance this replication work by incorporating key insights from one of its successors, CLIP[RKH+21], which demonstrated significant improvements in visual-language representation learning: 1) Scaling Up: Expand from ConVIRT's 217k to CLIP's 400M image-text pairs using multi-institutional data and various imaging modalities 2) Advanced Text Encoders: Replace ClinicalBERT with state-of-the-art models like BioBERT, PubMedBERT, or medical GPT variants 3) Prompt Engineering: Use structured templates ("Chest X-ray showing finding") instead of random sentence sampling to improve image-text alignment and Zero-shot performance 4) Training Optimization: Increase batch size from 32 to thousands using mixed precision training and gradient checkpointing for better efficiency.

Through this reproduction effort, the viability of ConVIRT's contrastive learning framework for medical vision-language tasks is confirmed, while underscoring how seemingly minor implementation choices and resource constraints can significantly impact the replication of deep learning benchmarks.

References

[IRK+19] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, and et al. A large chest radiograph dataset with uncertainty labels and expert comparison. *In Proceedings of the AAAI Conference on Articial Intelligence*, 2019.

[MCJ] https://www.physionet.org/content/mimic-cxr-jpg/2.1.0/.

[PyH] https://github.com/sunlabuiuc/PyHealth.

[RIB+18] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L Ball, and et al. Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *In 1st Conference on Medical Imaging with Deep Learning (MIDL)*, 2018.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and et al. A large chest radiograph dataset with uncertainty labels and expert comparison. *Computer Vision and Pattern Recognition (cs.CV)*, 2021.

[WPL+17] 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 weaklysupervised classication and localization of common thorax diseases. *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[WW20] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *arXiv preprint arXiv:2003.09871*, 2020.

[ZJM+22] 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*, 182:1–24, 2022.