Proceedings of Machine Learning Research 182:1–23, 2022 Machine Learning for Healthcare

EMIXER: End-to-end Multimodal X-ray Generation via Self-supervision

Siddharth Biswal SBISWAL7@GATECH.EDU
Georgia Institute of Technology

Peiye Zhuang PEIYE@ILLINOIS.EDU
UIUC

Ayis Pyrros AYIS@AYIS.ORG
Dupage Medical Group

Nasir Siddiqui NSIDDIQUI@GMAIL.COM
Dupage Medical Group

Sanmi Koyejo SANMI@ILLINOIS.EDU
UIUC

Jimeng Sun JIMENG.SUN@GMAIL.COM
UIUC

Abstract

Deep generative models have enabled the automated synthesis of high-quality data for diverse applications. However, the most effective generative models are specialized in data from a single domain (e.g., images or text). Real-world applications such as healthcare require multimodal data from multiple domains (e.g., both images and corresponding text), which are challenging to acquire due to limited availability and privacy concerns and are much harder to synthesize. To tackle this joint synthesis challenge, we propose an End-to-end MultImodal X-ray genERative model (EMIXER) for jointly synthesizing x-ray images and corresponding free-text reports, all conditional on diagnosis labels. EMIXER is a conditional generative adversarial model by 1) generating an image based on a label, 2) encoding the image to a hidden embedding, 3) producing the corresponding text via a hierarchical decoder from the image embedding, and 4) a joint discriminator for assessing both the image and the corresponding text. EMIXER also enables self-supervision to leverage a vast amount of unlabeled data. Extensive experiments with real X-ray reports data illustrate how data augmentation using synthesized multimodal samples can improve the performance of various supervised tasks, including COVID-19 X-ray classification with limited samples. Radiologists also confirm the quality of generated images and reports. We quantitatively show that EMIXER generated synthetic datasets can augment X-ray image classification, and report generation models to achieve 5.94% and 6.9% improvement on models trained only on real data samples. Overall, our results highlight the promise of generative models to overcome challenges in machine learning in healthcare.

1. Introduction

While clinical applications of supervised machine learning algorithms continue to advance, their impact is stifled by the limited amount of available labeled clinical data. This issue is only more dire by applications such as radiology report generation for medical images, which jointly combine paired data across images, clinical notes, and diagnosis labels. Data sharing across healthcare organizations and institutions remains problematic, often due to legal and privacy concerns McGuire et al. (2008); Filkins et al. (2016). On the other hand, generative modeling has improved dramatically in the past few years. While early Generative Adversarial Networks (GANs) could only synthesize low-resolution grayscale images Goodfellow et al. (2014), state-of-art generative models can now synthesize diverse high-quality and high-resolution images Brock et al. (2018); Karras et al. (2017, 2019a,b). GANs and related generative models have been applied to various domains such as computer vision Brock et al. (2018); Karras et al. (2017), natural language processing Dai et al. (2017b); Fedus et al. (2018), time-series synthesis Brophy et al. (2019), semantic segmentation Dong et al. (2017); Luc et al. (2016), among others. This manuscript explores using generative models to address the challenge of limited data in machine learning for clinical applications. We explore a variety of applications, with a focus on using synthetic data to augment real datasets – increasing the amount of the data and labels available Choi et al. (2017), thereby improving downstream model performance.

We focus on X-rays as they are a primary diagnostic tool in many clinical workflows, most importantly in radiology, and are used for detecting pneumonia, bone fracture, and cancerRajpurkar et al. (2017); Gulshan et al. (2016). Recent research efforts have shown promise for lung cancer detection in radiology, prostate cancer in pathology, and differential diagnoses in dermatology Ardila et al. (2019); Fujisawa et al. (2019); Arvaniti et al. (2018); Mohamed et al. (2018). Most recently, X-rays have been employed for the coronavirus diagnosis and prognosis Jacobi et al. (2020). Along with X-rays, associated reports written by clinicians are the primary communication between patients and doctors Schwartz et al. (2011); Kahn Jr et al. (2009). Several deep learning based X-ray image to report writing method have been proposed Jing et al. (2017, 2020); Li et al. (2018). Researchers have proposed generative models for clinical data Choi et al. (2017). However, existing methods are limited to a single modality – images or clinical reports only.

Thus, current generative models are not able to produce high-quality multimodal synthetic datasets, which is the focus of this paper. This manuscript investigates an end-to-end approach for generating multimodal X-ray images and text reports which are essential for radiology applications. To this end, our work addresses the following challenges.

- **Multimodal generation of images and corresponding reports**: Multimodal generative models are difficult to train compared to single-mode modal generative models Liu and Tuzel (2016); Isola et al. (2017); Zhu et al. (2017b,a); Choi et al. (2018, 2020). In the past few years, there have been multiple attempts at developing models that can generate multiple modalities at the same time Pu et al. (2018). In particular, text synthesis using generative models has proven to be extremely challenging – most likely because discrete text tokens are not differentiable – making it more difficult to train GANs. We show that using an end-to-end approach, combined with appropriate text embeddings, can overcome these issues.

- **Generative model training with limited labels**: Generative models typically require large quantities of high-quality labeled data for training. However, labels are scarcely available in real-world applications such as the medical domain. This renders training of high-quality generative models challenging. We present successful results with limited labeled X-ray data along with a large amount of unlabeled X-ray data and conjecture about the properties of X-rays, which make this feasible.

- **Difficultly of data augmentation with limited data:** The task of training a generative model for classifier augmentation Huang et al. (2018); Antoniou et al. (2017) is particularly challenging in the case of rare diseases or new phenotypes, as the limited amount of labels render training of generative models difficult. For example, in the case of the COVID-19 pandemic, the amount of available X-ray data and labels is extremely low. Given the limited labels, training high-quality generative models to augment the original dataset is a challenge. Pretraining models of large and diverse augmented data can potentially provide robust embeddings for new phenotypes.

We propose `EMIXER`, an end-to-end multimodal generative model that can generate paired chest X-ray images and corresponding reports simultaneously, conditioned on diagnosis labels. Our primary contributions are summarized in the following.

- **Multimodal X-ray image and report generation.** We show that `EMIXER` generates high-quality X-ray images and corresponding reports. Multiple radiologists scored average 7.340/10 for synthetic data and 7.825/10 for real data on their realisticness and quality. Furthermore, `EMIXER` generated synthetic datasets used to augment X-ray image classification models lead to up to 5.94% improvement in classification accuracy compared to models trained on real X-ray images only. Similarly, `EMIXER` augmented paired X-ray image and report datasets to improve X-ray report generation models up to 6.9% as measured by the CIDEr scores.
- **Learning high-quality generative models from limited samples.** `EMIXER` uses self-supervision to enable learning of high-quality generative models from limited labels. We show that even with 30% of the original labels, `EMIXER` can outperform baselines with the 100% labeled data in terms of image classification and report generation tasks.
- **Improved classification of COVID-19 chest X-rays via data augmentation.** We utilize the pre-trained model of `EMIXER` with augmenting classification models, applied to the automated diagnosis of COVID-19 from X-ray images. Our results show 11% improvement in predictive accuracy than the one without using pre-trained `EMIXER` model.

Generalizable Insights about Machine Learning in the Context of Healthcare

Our work shows that multimodal generative models can help overcome two major challenges in machine learning for healthcare. First, generative models enable generating synthetic data that can be shared across healthcare institutions to overcome data access issues. Synthetic multimodal data such X-ray image and reports can facilitate machine learning model development by many researchers who can not access private datasets. Second, generative models can augment real datasets to improve model performance. In machine learning for healthcare, there are many applications where dataset sizes are small to develop machine learning models. Generative models can augment real datasets to improve model performance leading more widespread adoption and deployment. Our method may be helpful in showing that these challenges in healthcare can be solved with multimodal generative models.

2. Related Work

Generative models. In the past few years, there has been significant progress in generative modeling of complex imaging data. Since the introduction of the Generative Adversarial Networks (GAN), there have been many variants proposed, such as DCGAN, Progressive GAN, Self-supervised GANs Goodfellow et al. (2014); Karras et al. (2017); Radford et al. (2015); Dai et al. (2017a), among others. In addition to GANs, other types of generative models are also quite widely used such as Flow Models, Autoregressive Models, and variational autoencoders Kingma and Welling (2013); Kingma and Dhariwal (2018); Dinh et al. (2014, 2016). Flow Models use a stack of invertible transformations to a sample from prior distributions, and thus can compute the exact log-likelihoods of observations. Autoregressive models factorize the distribution over observations into a sequence of conditional distributions (e.g. over pixels for images), then process each component in sequence Oord et al. (2016); Van den Oord et al. (2016). For image generation applications, GAN-based models produce among the photo-realistic images. However, the training of GAN models can be quite challenging with known issues such as mode collapse, and instability in convergence Salimans et al. (2016). There have been many works to improve upon these challenges, e.g., by changing the objective function Arjovsky et al. (2017). Some other research efforts have focused on constraining the discriminator through gradient penalties or normalization Miyato and Koyama (2018). BigGAN Zhang et al. (2018); Brock et al. (2018) adds the self-attention block, and ProGAN considers training a single model across a sequence of increasing resolutions Karras et al. (2017). While there is a lot of effort in modeling single modalities, especially images, there is a shortage of research on multimodal image and text generation. This work addresses the challenge of multimodal joint generation of image and text.

Medical report generation. Deep learning based image classification has been successfully applied to many different types of medical image classification tasks such as diabetic retinopathy classification, X-ray classification, cancer detection from cell images, and X-ray based bone classification Wang et al. (2018); Gulshan et al. (2016); Milletari et al. (2016), among other applications. Similarly, different image segmentation algorithms have successfully applied to medical images to identify different organs and diseases. There have been progress in the task of automated report generation for medical images such as X-rays Liu et al. (2019). Generative models have been applied X-ray image generation but do not handle multimodal data generation Waheed et al. (2020). Existing applications of machine learning to clinical tasks must address a variety of challenges, such availability of large datasets.

3. Methods

3.1. Problem Definition

We begin by introducing notations. We denote real chest X-ray images as $\mathbf{I_n} \in \mathbb{R}^{l \times l}$ where $l \times l$ is the size of the image, text X-ray reports as $\mathbf{S_n}$ and labels as $\mathbf{y_n} \in \{0, 1\}^k$ for nth data sample.

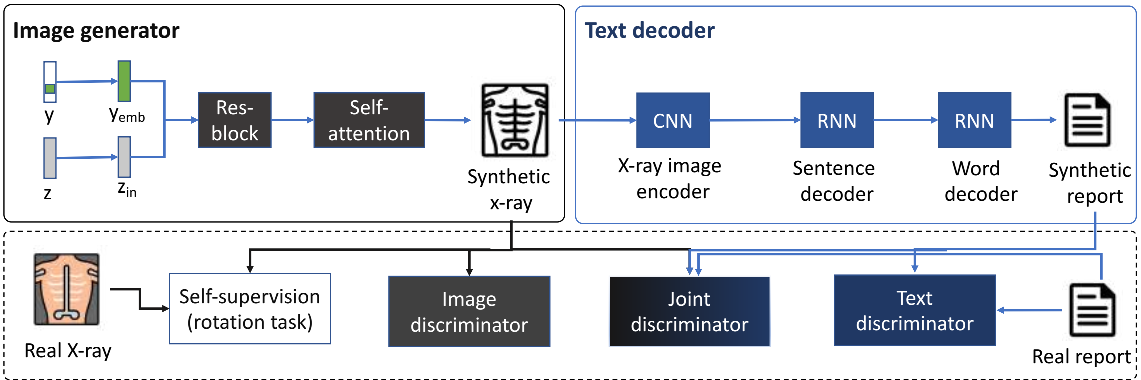

Figure 1: An overview of EMIXER generator framework

The X-ray report $\mathbf{S_n}$ contains a sequence of sentences $\mathbf{S_n} = [\boldsymbol{s_1^n} \dots \boldsymbol{s_T^n}]$, where the report length T may vary. Sentence $\boldsymbol{s_t^n}$ consists of sequence of words $\boldsymbol{s_t^n} = [\boldsymbol{a_{t,1}^n}, \boldsymbol{a_{t,2}^n}, \dots]$ where $\boldsymbol{a_{t,j}^n}$ j-th word represented as one-hot vectors in the sentence t of document n.

The dataset, denoted as $\mathcal{E}$ is a combination of images $\mathbf{I_n}$, reports $\mathbf{S_n}$ and labels $\mathbf{y_n}$ denoted as $\mathcal{E} = \{\mathbf{I_n}, \mathbf{S_n}, \mathbf{y_n}\}_{n=1}^N$. EMIXER generates synthetic dataset that consists of synthetic X-ray images $\hat{\mathbf{I}}_\mathbf{n}$, synthetic report $\hat{\mathbf{S}}_\mathbf{n}$ conditioned on class labels. We train an end-to-end generative model which consists of an X-ray image generator G, X-ray image discriminator $\mathrm{D_{image}}$, X-ray report discriminator $\mathrm{D_{report}}$, and an X-ray image to report decoder F. Each of these components is a neural network that are trained jointly to produce paired X-ray images and clinical reports conditioned on diagnosis labels.

3.2. The EMIXER Model

We describe primary components of EMIXER in this section. As illustrated in Fig. 1, EMIXER is composed of four different trainable networks: (a) **Image generator**: This image generator synthesizes X-ray images from a prior noise distribution conditioned on label information (b) **Image to report decoder**: An image to report decoder produces a text report from X-ray image (c) **Image Discriminator**: This discriminator is tasked with discriminating between real and synthetic X-ray images (d) **Text Discriminator**: This text discriminator distinguishes between real and synthetic X-ray reports (e) **Joint discriminator**: The joint discriminator combines the embedding of X-ray images and text to discriminate between real and synthetic embeddings.

3.2.1. X-ray Image Generator (G)

An X-ray image generator is a deep neural network that accepts two inputs; a noise vector $\boldsymbol{z} \in \mathbb{R}^{d_z}$ and class information $\mathbf{y}$ represented as one-hot vector. First, we split the noise vector $\mathbf{z}$ to obtain $\mathbf{z_{spl}} \in \mathbb{R}^{20}$ vectors.

The vectors $\mathbf{z_{spl}}$ is passed through a linear layer to obtain $\mathbf{z_{in}}$, $\mathbf{z_{in}} = \mathbf{W_l z_{spl}} + b_l$. We embed the class information $\mathbf{y}$ via a linear layer to obtain $\mathbf{y_{emb}} \in \mathbb{R}^{128}$. $\mathbf{z_{in}}$ concatenated with $\mathbf{y_{emb}}$ is passed through three layers of residual-block which applies batch-normalization with deconvolution operation, $\mathbf{res_{out}} = \text{res-block}(\mathbf{z_{in}}, \mathbf{y_{emb}})$ He et al. (2016). The output

$\mathbf{res_{out}}$ is passed through a self-attention block which applies applies a 1×1 convolution operation with softmax to obtain intermediate feature vectors which are combined with the original input to compute the att-block, $\mathbf{att_{out}}$ = self-att-block($\mathbf{res_{out}}$). Finally this output self-att-block is passed through another res-block to obtain the $\hat{\mathbf{I}}$ as the output of image generator. Taken together, the generator network can be abstracted as the following $\hat{\mathbf{I}} = G(\mathbf{z}, \mathbf{y})$. We provide implementation details of res-block, self-att-block blocks in the supplement.

3.2.2. X-ray Report Generator (F)

The image is fed through an image encoder convolutional neural network(CNN) to obtain a feature representation. These feature vectors are passed to a sentence decoder RNN to recurrently generate topic vectors for each sentence. These topic vectors are used by a word decoder to generate the words for each sentence as $\hat{\mathbf{S}} = \mathrm{F}(\hat{\mathbf{I}})$.

X-ray image encoder Specifically, given an image $\mathbf{I}$, we first extract its features $\overline{\mathbf{v}} \in \mathbb{R}^{512}$ from an intermediate layer of a CNN, $\overline{\mathbf{v}} = \mathrm{CNN}(\mathbf{I})$. We use a pretrained DenseNet-121 as the CNN model trained on a different chest X-ray dataset Huang et al. (2017). Note that this CNN is different from the CNN used in the image discriminator $\mathrm{D_{image}}$. The report generator module is composed of a sentence decoder and word decoder RNN which are described below.

Sentence decoder RNN: Given the X-ray image features $\overline{\mathbf{v}}$ extracted by the CNN, a sentence decoder is used generate topic vectors $\mathbf{t_i}$. We employ a Long-Short Term Memory network (LSTM) to compute the hidden state as $\mathbf{h}_i = \mathrm{LSTM}(\overline{\mathbf{v}}; \mathbf{h}_{\text{i-1}})$. We use the hidden states in two ways: First, we project the hidden state $\mathbf{h_i}$ through a linear layer and logistic layer to get probability distribution $\mathbf{u_i}$ over two states CONTINUE = 0, STOP = 1. Second, we also feed $\mathbf{h_i}$ through three-layered fully connected network to get a topic vector $\mathbf{t_i}$for ith sentence in the report, $\mathbf{t_i} = \mathbf{W}_{\text{to}}\mathbf{h_i} + b_{\text{to}}$.

Word decoder RNN: The words for each individual sentence are generated by a word decoder which is a trainable three-layer LSTM. The sentence topics $\mathbf{t_i}$ generated by the sentence decoder are combined with the <START> token as input for the first and second input to the word LSTM. In subsequent steps, we provide the hidden state of the last LSTM layer to predict a distribution over the words in the vocabulary. The hidden state $\mathbf{h}_{\text{word}} \in \mathbb{R}^H$ of the word LSTM is directly used to predict the distribution over words: $p\left(\text{word}|\mathbf{h}_{\text{word}}\right) = \mathrm{softmax}\left(\mathbf{W}_{\text{out}}\mathbf{h}_{\text{word}}\right)$ where $\mathbf{W}_{\text{out}}$ is the parameter matrix. Finally, after the word decoder generates the word sequences, we concatenate all the generated sequence to obtain the final report.

3.2.3. Discriminator (D)

EMIXER uses three discriminators, an image discriminator, a report discriminator and joint embedding discriminator to ensure image and report consistency of the synthetic data. The image discriminator measures whether the generated image $\hat{\mathbf{I}}$ matches the image distribution of real X-ray images, and the report discriminator $\hat{\mathbf{S}}$ discriminates between the real and synthetic X-ray reports.

X-ray Image discriminator ($\mathrm{D_{image}}$): We use a convolutional neural network discriminator for X-ray images which are fed real and synthetic X-ray images for classification. The discriminators use a ResNet architecture where the input image is passed through multiple layers of ResBlocks, where ResBlocks are composed of 3×3 convolution with ReLU layers He et al. (2016). This image discriminator can be represented as $D(\mathbf{I}, y) = c_{\mathrm{rf}}(\tilde{D}(\mathbf{I})) + P(\tilde{D}(\mathbf{I}), y)$ where $P(\tilde{v}, y) = \tilde{x}^{\top} W y$ is a linear layer with weight matrix W applied to to image feature v and one-hot encoded label y. c_{rf} is a linear classifier tasked with detecting if the provided sample is real or fake.

X-ray Report Discriminator ($\mathrm{D}_{\mathrm{report}}$): We use a X-ray report discriminator which classifies a given X-ray report as real or fake. X-ray reports generated from the decoder and real X-ray reports are passed as input discriminator. We employ a LSTM to to extract text embeddings from given X-ray report $\mathbf{S}$, $\mathbf{p} = \mathrm{LSTM}(\mathbf{S})$ Cho et al. (2014). These report embeddings $\mathbf{p}$ are passed through multi-layer linear layers with softmax layer to obtain $\mathbf{y}_{\mathrm{r/f}}$. The report discriminator can be abstracted as to discriminate between real or fake report embedding as $\mathbf{y}_{\mathrm{r/f}} = \mathrm{D}_{\mathrm{report}}(\hat{\mathbf{e}})$. We provide further details of the implementation in the supplementary section.

Joint Discriminator for X-ray images and Reports ($\mathrm{D}_{\mathrm{joint}}$): Along with the image discriminator and report discriminator, we also use a joint embedding discriminator. We hypothesize that as the X-ray images and reports are dependent upon each other, a joint multimodal embedding discriminator provides further guidance to the generator network for generating higher quality images and reports. This joint embedding discriminator is designed to discriminate real joint embeddings from fake joint embeddings. The joint embedding discriminator first obtains image features $\mathbf{I_{emb}}$ from the X-ray images using a CNN before the pooling layer. The text-reports are provided as input to an LSTM. The last hidden vector of the LSTM is passed through a linear layer to obtain report embedding $\mathbf{S_{emb}}$. The image feature vector $\mathbf{I_{emb}}$ and report embedding $\mathbf{S_{emb}}$ are concatenated together to form the joint embedding $\mathbf{C_{joint}}$. This joint embedding is passed through linear layers to obtain probability of real or fake embedding. This discriminator can be abstracted as $\mathbf{y}_{\mathrm{r/f}} = \mathrm{D}_{\mathrm{joint}}(\hat{\mathbf{C}}_{\mathbf{joint}})$.

Learning: Previous works have shown that self-supervision guide the classifier to learn useful data representation by detecting auxiliary information such as rotation angles. When applied to image classification, typically images are rotated and the angle of rotation is provided as the artificial label. In this rotation task, the self-supervised task is to predict the angle of rotation of an image. We use $\mathcal{R} = \{0, 90, 180, 270\}$ rotation angles. Image $\mathbf{I}$ is rotated by r degrees is denoted as $\mathbf{I}^r$ and $Q_{D_{\mathrm{image}}}(R = r|\mathbf{I}^r)$ is probability distribution over the rotation angles. The `EMIXER` framework corresponds to a constrained minimax game given by where the value function V is given by

$$
\begin{aligned}
V(G, D_*) = {} & \mathbb{E}_{\mathbf{x}_1 \sim p_{\mathbf{X}_1}}\left[-\log D_{\mathrm{image}}\left(\hat{\mathbf{I}}\right)\right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}}\left[-\log\left(1 - D_{\mathrm{image}}(G(\mathbf{z}))\right)\right] \\
& + \mathbb{E}_{\mathbf{x}_1 \sim p_{\mathbf{X}_1}}\left[-\log D_{\mathrm{report}}\left(\hat{\mathbf{S}}\right)\right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}}\left[-\log\left(1 - D_{\mathrm{report}}(F(G(\mathbf{z})))\right)\right] \\
& + \mathbb{E}_{\mathbf{x}_1 \sim p_{\mathbf{X}_1}}\left[-\log D_{\mathrm{joint}}\left(\hat{\mathbf{I}}, \hat{\mathbf{S}}\right)\right] \\
& + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}}\left[-\log\left(1 - D_{\mathrm{joint}}(F(G(\mathbf{z})), G(\mathbf{z})\right)\right] \\
& + \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}}\left[\log Q_{D_{\mathrm{image}}}(R = r|\mathbf{I}^r)\right]
\end{aligned}
$$

where G, D_{image}, D_{report}, D_{joint}, F are the image generator, image discriminator, report discriminator and image to report decoder, respectively. `EMIXER` can be trained by back propagation with the alternating gradient update steps. The details of the learning algorithm are given in the supplementary materials.

4. Experiments

In this section, we perform extensive evaluations to measure the effectiveness of `EMIXER` for paired chest X-ray images and report generation. We empirically show that (1) our proposed model can generate high-quality X-ray images and reports (2) `EMIXER` with self-supervised loss can match the generated sample quality of the conditional models using an only fraction of labels (3) `EMIXER` can be used to augment datasets in limited label settings such as COVID-19 chest X-ray detection.

4.1. Datasets

We perform experiments on MIMIC-CXR dataset, one of the largest X-ray datasets containing 377,110 X-ray images and corresponding reports Johnson et al. (2019). MIMIC dataset contains 377,110 chest X-rays associated with 227,827 imaging studies sourced from the Beth Israel Deaconess Medical Center between 2011-2016. The labels extracted from the reports include 14 different unique classes. We resize the images to $128 \times 128 \times 3$ as done in previous work Miyato and Koyama (2018).

4.2. Evaluation Metrics

We perform quantitative and qualitative experiments: (a) We used accuracy and AUC as classification metrics for classification experiments. We use CIDEr, BLEU scores for image captioning experiments Vedantam et al. (2015); Papineni et al. (2002). (b) To evaluate X-ray image quality, we use the Fréchet Inception Distance (FID) scores. We use a special pre-trained Inception network on chest X-ray images. We have provided further details in the supplement. Lower FID scores indicate that model can generate different images per class so it's indicative of whether the model is able to generate diverse images. If the model generates images that are not diverse, the distance of the data distribution will be higher from the real samples. (c) We qualitatively evaluate the generated X-ray images and reports. We present randomized pairs of real or synthetic X-ray images and reports to clinical experts for evaluation (they do not know if the presented sample is real or synthetic). The clinical experts were asked to provide a numerical quality score between 1-10 (10 being the best) for each sample.

4.3. Models

JointGAN: JointGAN trains multiple generators and a single softmax-based critic, all jointly trained via adversarial learningPu et al. (2018) to generate joint data distributions. **CoGAN**: CoGAN learns separate generators for two different domains with tied weights on the first few layers for shared latent representations Liu and Tuzel (2016). **Single Modal Image GAN with text decoder(SM-GAN)** In this setup, we use a GAN model to generate X-ray images. These X-ray images are passed to a text decoder which produces text reports corresponding to the synthetic chest X-rays. `EMIXER`: We compare these baselines against which is a self-supervised generative model with multiple discriminators for each modality. The final loss for our discriminator is the combination of adversarial loss of both the generators and joint embedding.

4.4. Experimental Results and Discussion

Our experiments aim to answer the following questions.

- Can EMIXER generate high-quality X-ray images?
- Can EMIXER generate high-quality pairs of X-ray images and reports?
- Can EMIXER learn a high-quality generative model from limited samples?
- Can EMIXER be used to improve COVID X-ray classification?

4.4.1. Image quality evaluation: Is EMIXER capable of generating high-quality X-ray images?

One of the primary applications of generative models is data augmentation to increase sample size and improve downstream model performance. We use the baselines and EMIXER to augment the real X-ray images and evaluate the improved quality of the datasets by using these augmented datasets for X-ray image classification.

X-ray image classification setup: We trained two separate X-ray image classification models on real X-ray images and synthetic X-ray images. We hypothesize that good generative models can generate images that resemble real data and can be used to train a classification model. These classification models are evaluated on held out real X-ray images. This setup evaluates the performance of the classification model for five different classes of diseases related to the X-ray images. In this experiment, we report accuracy and AUC for classification scores in Table 1, where we increase the dataset size by augmenting the real data with generated X-ray images. We use 100k real X-ray images and gradually increase the augmented dataset size by adding synthetic X-ray images up to 600k. We notice improved performance of these image classification model by up to 5.94% compared to real X-ray images, and 3.6% improvement compared to the best baseline. This highlights that EMIXER can generate synthetic X-ray images that can augment the real dataset to improve the classification performance.

4.5. Joint Image and Text Evaluation: Can EMIXER generate high quality pairs of image and reports?

One of the primary advantages of EMIXER is the ability to jointly generate paired X-ray images and reports. We performed two different experiments to understand the effectiveness of EMIXER towards generating paired images and reports.

Report Generation Task: X-ray report generation is one of the key tasks in radiology clinical workflow Schwartz et al. (2011). We validate the effectiveness of augmented paired image and report datasets for report generation task. In this setup, we train report generation models on real data and a combination of real and synthetic data. These trained models are evaluated on held-out real paired datasets. We present the results of these experiments in Table 1. In this setup, we vary the amount of synthetic data added to the real dataset. We present the performance of real and augmented datasets for report generation task in terms of natural language processing metrics such as CIDEr, BLEU-1 Vedantam et al. (2015);

Table 1: Comparison of X-ray report generation model performance with real and augmented image dataset; In this table R indicates real data samples, S indicates synthetic data samples

		Image Classification		Report Generation				
Dataset	Method	AUC	ACC	CIDEr	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Only real data	R 100k	$.824 \pm 0.0034$	$.846 \pm 0.0041$	$.712 \pm 0.0014$	$.253 \pm 0.0024$	$.198 \pm 0.0034$	$.095 \pm 0.0041$	$.074 \pm 0.0012$
JointGAN	R100k + S50k	$.798 \pm 0.0012$	$.814 \pm 0.0023$	$.719 \pm 0.0023$	$.259 \pm 0.0019$	$.201 \pm 0.0024$	$.098 \pm 0.0025$	$.079 \pm 0.0031$
	R100k + S100k	$.812 \pm 0.0013$	$.835 \pm 0.0021$	$.725 \pm 0.0019$	$.261 \pm 0.0014$	$.207 \pm 0.0029$	$.114 \pm 0.0022$	$.081 \pm 0.0034$
	R100k + S300k	$.826 \pm 0.0018$	$.839 \pm 0.0011$	$.748 \pm 0.0017$	$.272 \pm 0.0015$	$.213 \pm 0.0021$	$.129 \pm 0.0029$	$.085 \pm 0.0051$
	R100k + S600k	$.831 \pm 0.0009$	$.846 \pm 0.0019$	$.773 \pm 0.0022$	$.313 \pm 0.0011$	$.224 \pm 0.0041$	$.134 \pm 0.0021$	$.093 \pm 0.0021$
CoGAN	R100k + S50k	$.827 \pm 0.0034$	$.843 \pm 0.0019$	$.703 \pm 0.0019$	$.231 \pm 0.0022$	$.192 \pm 0.0012$	$.082 \pm 0.0019$	$.073 \pm 0.0031$
	R100k + S100k	$.829 \pm 0.0011$	$.854 \pm 0.0021$	$.692 \pm 0.0014$	$.214 \pm 0.0023$	$.187 \pm 0.0012$	$.073 \pm 0.0022$	$.067 \pm 0.0033$
	R100k + S300k	$.831 \pm 0.0013$	$.857 \pm 0.0024$	$.724 \pm 0.0015$	$.241 \pm 0.0024$	$.211 \pm 0.0012$	$.091 \pm 0.0041$	$.076 \pm 0.0025$
	R100k + S600k	$.837 \pm 0.0011$	$.849 \pm 0.0023$	$.734 \pm 0.0022$	$.251 \pm 0.0021$	$.236 \pm 0.0012$	$.114 \pm 0.0032$	0.081 ± 0.0019
SMGAN	R100k + S50k	$.818 \pm 0.0013$	$.832 \pm 0.0013$	$.713 \pm 0.0031$	$.251 \pm 0.0019$	$.203 \pm 0.0034$	$.093 \pm 0.0021$	$.077 \pm 0.0014$
	R100k + S100k	$.823 \pm 0.0035$	$.831 \pm 0.0014$	$.723 \pm 0.0032$	$.258 \pm 0.0031$	$.207 \pm 0.0034$	$.098 \pm 0.0022$	$.079 \pm 0.0019$
	R100k + S300k	$.821 \pm 0.0021$	$.847 \pm 0.0019$	$.731 \pm 0.0033$	$.263 \pm 0.0018$	$.212 \pm 0.0034$	$.106 \pm 0.0032$	$.089 \pm 0.0036$
	R100k + S600k	$.842 \pm 0.0029$	$.859 \pm 0.0018$	$.752 \pm 0.0039$	$.275 \pm 0.0011$	$.236 \pm 0.0034$	$.125 \pm 0.0031$	$.096 \pm 0.0025$
EMIXER	R100k + S50k	$.835 \pm 0.0015$	$.857 \pm 0.0024$	$.731 \pm 0.0031$	$.276 \pm 0.0034$	$.204 \pm 0.0027$	$.112 \pm 0.0032$	$.078 \pm 0.0021$
	R100k + S100k	$.842 \pm 0.0021$	$.864 \pm 0.0024$	$.752 \pm 0.0024$	$.297 \pm 0.0041$	$.216 \pm 0.005$	$.132 \pm 0.0019$	$.083 \pm 0.0024$
	R100k + S300k	$.853 \pm 0.0019$	$.869 \pm 0.0028$	$.763 \pm 0.0035$	$.324 \pm 0.0042$	$.229 \pm 0.0014$	$.145 \pm 0.0014$	$.097 \pm 0.0031$
	R100k + S600k	$.873 \pm 0.0025$	$.884 \pm 0.0026$	$.783 \pm 0.0043$	$.346 \pm 0.0022$	$.247 \pm 0.0019$	$.169 \pm 0.0018$	$.132 \pm 0.0052$

Table 2: Comparative evaluation of phenotype classification via joint embedding with real and augmented data

Method	Dataset	AUC	Acc
Only Real	Real [100k]	$.849 \pm 0.0025$	$.868 \pm 0.0021$
JointGAN	R100k + S300k	$.869 \pm 0.0023$	$.905 \pm 0.0023$
CoGAN	R100k + S300k	$.871 \pm 0.0014$	$.896 \pm 0.0019$
SMGAN	R100k + S300k	$.883 \pm 0.0016$	$.902 \pm 0.0015$
EMIXER	R100k + S300k	$.905 \pm 0.0019$	**.924** ± 0.0012

Papineni et al. (2002). We show that EMIXER improves up to 6.9% compared to models trained only on real datasets. This highlights the fact that EMIXER can be used to augment and improve report generation models.

Multimodal joint embeddings of X-ray images and reports: The multimodal embeddings learned can be used for classification tasks. We perform an experiment to evaluate the joint quality of images and generated text. In table 2, we compare the result of varying combinations of real and synthetic data on the joint modeling task. In this joint modeling task, we combine features from X-ray images and text reports together for downstream classification. We classify different disease phenotypes using these joint embeddings. We find that adding a synthetic dataset to the real dataset for this joint embedding significantly improves the performance of the classification model.

4.5.1. Limited label setup: Can we learn a high quality generative model from limited data?

Machine Learning applications in clinical domains are often limited by the amount of available data and labels. Since generative models require large amounts of data and labels to train, it is a challenge in clinical tasks to learn a high-quality generative models. We show in the

Table 3: Comparison of generative models with limited labels

Method	Acc	BLEU-1	FID
CoGAN(full)	0.827 ± 0.0011	0.247 ± 0.0023	15.23 ± 0.0015
JointGAN(full)	0.813 ± 0.0014	0.221 ± 0.0031	16.58 ± 0.0019
SM-GAN(full)	0.813 ± 0.0021	0.221 ± 0.0028	16.58 ± 0.0021
EMIXER (30%)	0.838 ± 0.0013	0.258 ± 0.0021	12.84 ± 0.0021
EMIXER (50%)	0.842 ± 0.0008	0.269 ± 0.0024	11.73 ± 0.0023
EMIXER (100%)	0.845 ± 0.0024	0.271 ± 0.0014	11.31 ± 0.0028

following experiments that we can employ self-supervision to overcome the label limitations. We explore the limits of usage of labels by varying the percentage of labels used in the models. In this experiment, we use limited labels ranging from 30%, 50% to compare with 100% label usage. We show that even with limited labels EMIXER can perform competitively. We compare existing baselines to our model which uses self-supervision to able to generate images from limited labels. Table 3 shows that EMIXER outperforms the baselines in terms of image generation diversity as measured by FID.

4.5.2. Case Study: COVID-19 X-ray data augmentation experiment

In this task, we use EMIXER to augment chest X-ray images to improve COVID-19 detection. We use COVID-19 Radiography Database Chowdhury et al. (2020) which contains 10912 normal, 3616 COVID-19 positive cases, 6012 lung opacity and 1345 viral pneumonia cases. Currently, COVID X-rays classification includes four classes: normal, bacterial pneumonia, viral pneumonia and COVID-19. In this experiment, we evaluate if EMIXER generated synthetic data can augment chest X-ray image samples for the COVID-19 classification task. Specifically, we compare two different models: model trained on COVID-19 dataset, model trained on combined data of COVID-19 data and EMIXER generated synthetic data. We show that augmenting real datasets with EMIXER generated samples improves the overall performance in Table 4.

Table 4: Comparison of performance for COVID-19 classification

Type	Phenotype	Precision	Recall	F1-score
COVID samples	Normal	0.904 ± 0.0021	0.895 ± 0.0013	0.899 ± 0.0024
	Lung Opacity	0.866 ± 0.0026	0.853 ± 0.0014	0.859 ± 0.0018
	Viral Pneumonia	0.898 ± 0.0016	0.887 ± 0.0024	0.892 ± 0.0018
	COVID-19	0.865 ± 0.0019	0.884 ± 0.0023	0.852 ± 0.0019
COVID samples+ EMIXER (50k Samples)	Normal	$\mathbf{0.928 \pm 0.0021}$	$\mathbf{0.916 \pm 0.0019}$	$\mathbf{0.921 \pm 0.0018}$
	Lung Opacity	$\mathbf{0.912 \pm 0.0020}$	$\mathbf{0.878 \pm 0.0023}$	$\mathbf{0.894 \pm 0.0021}$
	Viral Pneumonia	$\mathbf{0.927 \pm 0.0018}$	$\mathbf{0.903 \pm 0.0022}$	$\mathbf{0.914 \pm 0.0021}$
	COVID-19	$\mathbf{0.892 \pm 0.0019}$	$\mathbf{0.901 \pm 0.0014}$	$\mathbf{0.896 \pm 0.0019}$

4.5.3. Evaluation by Radiologists

We perform a qualitative evaluation of the generated X-ray images and reports. In this task, we present randomized X-ray images and reports to expert doctors. Two radiologists provide a rating between 1-10 for each pair of images and reports. We have shown the results of this evaluation task in figure 2. The scores for real and synthetic X-rays samples were 7.825 ± 1.17 and 7.34 ± 1.321. The inter-rater agreement was 0.832 measured using cohen's kappa. The comments provided by the doctors indicate that synthetic samples were similar to real examples, with some language incoherence in X-ray reports.

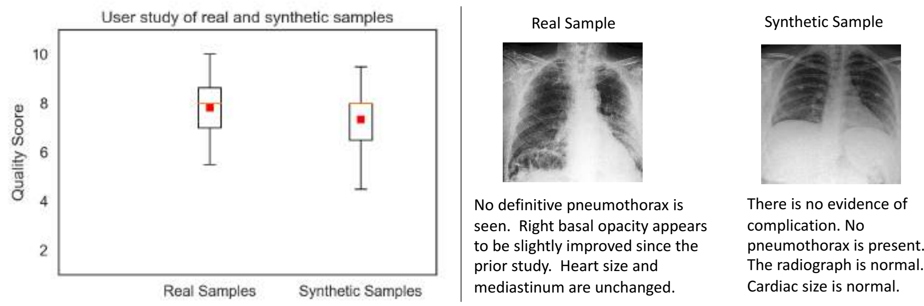

Figure 2: Qualitative evaluation. (a) User study Results (b) Comparative real and synthetic samples

5. Discussion

This paper addresses the challenging multimodal paired x-ray image and report generation task by proposing a novel self-supervised multimodal generative model called `EMIXER`. `EMIXER` successfully uses a multimodal generative model to learn to generate paired x-ray images and reports. We use self-supervision to guide `EMIXER` to learn from limited samples which are very applicable in the medical domain as the number of labels is often limited. We also use multiple discriminators to guide the process of image generation, and report decoding. We show via extensive experiments that `EMIXER` can augment real x-ray image datasets to improve downstream classification tasks. Finally, in a timely case-study, we show that `EMIXER` can also improve COVID-19 x-ray classification.

Limitations There are a few limitations of our proposed method. Our proposed method uses disease labels for conditioning the generation process. In real world, there are many more controls which affect the x-ray images such fluid, heart shape and size etc. We do not account for those control parameters for X-ray image generation. Another shortcoming of our proposed method is that sometimes there are few blurry or floating images among the generated X-ray images. These images do not affect final classification performance in image augmentation settings.

References

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Diego Ardila, Atilla P Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Eirini Arvaniti, Kim S Fricker, Michael Moret, Niels Rupp, Thomas Hermanns, Christian Fankhauser, Norbert Wey, Peter J Wild, Jan H Rueschoff, and Manfred Claassen. Automated gleason grading of prostate cancer tissue microarrays via deep learning. *Scientific reports*, 8(1):1–11, 2018.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Eoin Brophy, Zhengwei Wang, and Tomas E Ward. Quick and easy time series generation with established image-based gans. *arXiv preprint arXiv:1902.05624*, 2019.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

Yunjey Choi, Minje Choi, Munyoung Kim, et al. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, et al. Stargan v2: Diverse image synthesis for multiple domains. *CVPR*, 2020.

Muhammad EH Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al Emadi, et al. Can ai help in screening viral and covid-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020.

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2970–2979, 2017a.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in neural information processing systems*, pages 6510–6520, 2017b.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5706–5714, 2017.

William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*, 2018.

Barbara L Filkins, Ju Young Kim, Bruce Roberts, Winston Armstrong, Mark A Miller, Michael L Hultner, Anthony P Castillo, Jean-Christophe Ducom, Eric J Topol, and Steven R Steinhubl. Privacy and security in the era of digital health: what should translational researchers know and do about it? *American journal of translational research*, 8(3):1560, 2016.

Y Fujisawa, Y Otomo, Y Ogata, Y Nakamura, R Fujita, Y Ishitsuka, R Watanabe, N Okiyama, K Ohara, and M Fujimoto. Deep-learning-based, computer-aided classifier developed with a small dataset of clinical images surpasses board-certified dermatologists in skin tumour diagnosis. *British Journal of Dermatology*, 180(2):373–381, 2019.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 718–731, 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, et al. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

Adam Jacobi, Michael Chung, Adam Bernheim, and Corey Eber. Portable chest x-ray in coronavirus disease-19 (covid-19): A pictorial review. *Clinical Imaging*, 2020.

Baoyu Jing, Pengtao Xie, and Eric Xing. On the automatic generation of medical imaging reports. *arXiv preprint arXiv:1711.08195*, 2017.

Baoyu Jing, Zeya Wang, and Eric Xing. Show, describe and conclude: On exploiting the structure information of chest x-ray reports. *arXiv preprint arXiv:2004.12274*, 2020.

Alistair EW 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.

Charles E Kahn Jr, Curtis P Langlotz, Elizabeth S Burnside, John A Carrino, David S Channin, David M Hovsepian, and Daniel L Rubin. Toward best practices in radiology reporting. *Radiology*, 252(3):852–856, 2009.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019a.

Tero Karras, Samuli Laine, Miika Aittala, et al. Analyzing and improving the image quality of stylegan. *arXiv preprint*, 2019b.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

Yuan Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. Hybrid retrieval-generation reinforced agent for medical image report generation. In *Advances in neural information processing systems*, pages 1530–1540, 2018.

Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, Willie Boag, Wei-Hung Weng, Peter Szolovits, and Marzyeh Ghassemi. Clinically accurate chest x-ray report generation. *arXiv preprint arXiv:1904.02633*, 2019.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

Amy L McGuire, Rebecca Fisher, Paul Cusenza, Kathy Hudson, Mark A Rothstein, Deven McGraw, Stephen Matteson, John Glaser, and Douglas E Henley. Confidentiality, privacy, and security of genetic and genomic test information in electronic health records: points to consider. *Genetics in Medicine*, 10(7):495–499, 2008.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

Aly A Mohamed, Wendie A Berg, Hong Peng, Yahong Luo, Rachel C Jankowitz, and Shandong Wu. A deep learning method for classifying mammographic breast density categories. *Medical physics*, 45(1):314–321, 2018.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Yunchen Pu, Shuyang Dai, Zhe Gan, Weiyao Wang, Guoyin Wang, Yizhe Zhang, Ricardo Henao, and Lawrence Carin. Jointgan: Multi-domain joint distribution learning with generative adversarial nets. *arXiv preprint arXiv:1806.02978*, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2226–2234, 2016.

Lawrence H Schwartz, David M Panicek, Alexandra R Berk, Yuelin Li, and Hedvig Hricak. Improving communication of diagnostic radiology findings through structured reporting. *Radiology*, 260(1):174–181, 2011.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Fadi Al-Turjman, and Plácido Rogerio Pinheiro. Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection. *Ieee Access*, 8:91916–91923, 2020.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, and Ronald M Summers. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9049–9058, 2018.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Jun-Yan Zhu, Taesung Park, Phillip Isola, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, et al. Toward multimodal image-to-image translation. In *Advances in neural information processing systems (NeurIPS)*, 2017b.

Appendix A. Supplementary

A.1. Preliminaries: Generative Adversarial Networks

The Generative Adversarial Network (GAN) involves a Generator (G) and a Discriminator (D) network. The purpose of Generator (G) is to map random noise to samples, while the Discriminator (D) classifies real and generated samples. The generator builds a mapping function from a prior noise distribution $p_z(z)$ to data space as G($\mathbf{z}$) to learn a generator distribution p_g, while the discriminator D($\mathbf{x}$) outputs a single scalar representing the probability that $\mathbf{x}$ came form training data rather than p_g where p_{data} is the real data distribution. The basic GAN objective function seeks a Nash equilibrium to the following two player min-max problem where value function is defined as $\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$ Goodfellow et al. (2014) where $z \in \mathbb{R}^{d_z}$ is a latent variable drawn from distribution $p(\boldsymbol{z})$ such as the unit Gaussian $\mathcal{N}(0, I)$ or the unit uniform $\mathcal{U}[-1, 1]$. Generative adversarial networks can be extended to conditional versions if the generators and discriminators are conditioned on label information $\mathbf{y}$ Mirza and Osindero (2014). The condition information $\mathbf{y}$ and $p(\boldsymbol{z})$ are combined in the joint representation of the generator. The discriminator is provided with generated samples and labels $\mathbf{y}$ as inputs. The objective function can be modified as $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$

A.2. EMIXER: Architecture Details

A.2.1. Notations Table

We used these notations to describe different modules. The notations are described in table 4.

Symbol	Definition and description
$\mathbf{I_n}$	Notation for X-ray Images
$\mathbf{S_n}$	Notation for sentences in the X-ray report
$\mathbf{\hat{I}_n}$	Generated X-ray images
$\mathbf{\hat{S}_n}$	Generated X-ray reports
$\mathcal{E}$	Dataset consisting of images, reports and labels
$\mathbf{y_n}$.	Notation for labels associated with images
$\mathbf{w_{N_s}}$	Words in the sentences of X-ray report
$\mathbf{z}$	Noise vector for the generator
G(.)	Generator Neural Network
D_{image}(.)	X-ray image discriminator Neural Network
D_{report}(.)	Discriminator Neural Network
D_{joint}(.)	Discriminator Neural Network
F(.)	Report Generator Network

Table 5: Notations used in EMIXER

A.2.2. EMIXER Model

In this section, we provide further description of the different neural networks within in EMIXER.

X-ray Image Generator(G): Figure 3 shows the architecture of the image generator. X-ray image generator accepts two inputs: (a) noise vector $\boldsymbol{z} \in \mathbb{R}^{120}$ (b) class information $\mathbf{y}$ represented as one-hot vector. We embed the class information $\mathbf{y}$ via a linear layer to obtain vector $\mathbf{y_{emb}} \in \mathbb{R}^{128}$. It has been shown generators can use the latent space to influence features at different resolutions by providing direct connections from noise vector to different layers of the generator. We split the noise vector $\mathbf{z}$ to obtain different smaller vectors $\mathbf{z_{spl}} \in \mathbb{R}^{20}$ (https://pytorch.org/docs/master/generated/torch.split.html). The vectors $\mathbf{z_{spl}}$ is passed through a linear layer to obtain $\mathbf{z_{in}}$, $\mathbf{z_{in}} = \mathrm{fc}(\mathbf{z_{spl}})$. We concatenate $\mathbf{z_{in}}$ with $\mathbf{y_{emb}}$ which is passed through three layers of Res-block-up He et al. (2016). We have provided the details of this convolution block in table 6. We use h, w to denote input height and width and c_i, c_o are input and output channels for the Res-block-up. The output from the previous layer and the concatenated vector from $\mathbf{z_{in}}$, $\mathbf{y_{emb}}$ is provided as input to each of the residual block. The final residual output $\mathbf{res_{out}}$ is passed through a self-attention block which applies applies a 1×1 convolution operation with softmax to obtain intermediate feature vectors which are combined with the original input to compute the att-block, $\mathbf{att_{out}} = \text{self-att-block}(\mathbf{res_{out}})$. Finally this output self-att-block is passed through another res-block to obtain the $\hat{\mathbf{I}}$ as the output of generator.

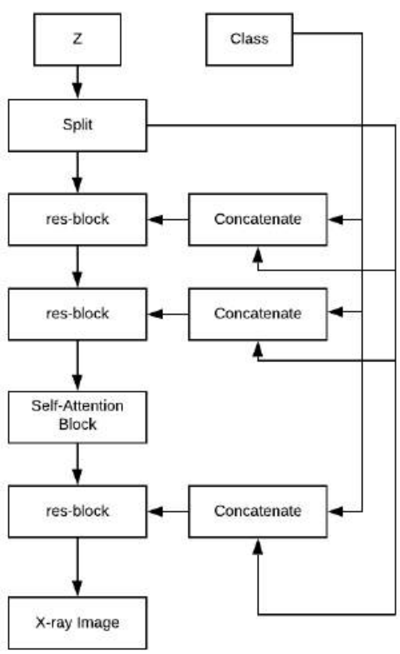

Figure 3: Architectural layout of EMIXER image generator G

Table 6: Details of Res-block-up for generator

Layer	Kernel	Output
Shortcut	[1,1,1]	$2h \times 2w \times c_o$
condBN, ReLU	–	$h \times w \times c_i$
Conv	[3,3,1]	$2h \times 2w \times c_o$
condBN, ReLU	–	$2h \times 2w \times c_o$
Conv	[3,3,1]	$2h \times 2w \times c_o$
Addition	–	$2h \times 2w \times c_o$

Table 7: Details of Res-block-Down for discriminator.

Layer	kernel	Output
Shortcut	[1,1,1]	$h/2 \times w/2 \times c_o$
ReLU	–	$h \times w \times c_i$
Conv	[3,3,1]	$h \times w \times c_o$
ReLU	–	$h \times w \times c_o$
Conv	[3,3,1]	$h/2 \times w/2 \times c_o$
Addition	–	$h/2 \times w/2 \times c_o$

Generator component dimensions: Residual block

- Conv 2D operation 1: kernel size = (3, 3), stride= (1, 1), padding= (1, 1), in channel = 1024, out channel = 1024
- Conv 2D operation 2: kernel size = (3, 3), stride= (1, 1), padding= (1, 1), in channel = 1024, out channel = 1024
- Conv 2D operation 3: kernel size= (1, 1), stride= (1, 1), in channel = 1024, out channel = 1024
- Batch Normalization: in channels: 1024, out=2014

Components of neural network

- Split function: This Split function splits input vector into multiple smaller chunks. As described in figure 3, noise vector z is split into smaller chunks and combined with class embeddings to be passed to residual blocks. We use the split operation in PyTorch library which splits the input vector along the specified dimension(https://pytorch.org/docs/stable/generated/torch.split.html).
- Residual block up (res-block-up): A residual block used to up sample the provided input. This is a combination of convolution blocks with conditional batch normalization layers.

X-ray Image Discriminator ($\mathrm{D}_{\text{image}}$): Figure 4 shows the architecture of the X-ray image discriminator. X-ray image discriminator is used to distinguish between real and fake X-ray images. The discriminator takes an X-ray image $\mathbf{I} \in \mathbb{R}^{128\times128\times3}$ as an input. Image

I is passed through multiple layers of residual convolutional blocks Res-block-Down. We have provided the details of the convolution block in table 7. We use h, w to denote input height and width and c_i, c_o are input and output channels for the Res-block-Down. In each residual convolutional block the number of channels is doubled to process the previous layers input. The intermediate feature vector obtained from the residual blocks is passed through a pooling layer and ReLU activation layer. Finally we combine it with the projected condition vector and pass there through a linear layer to obtain the final output.

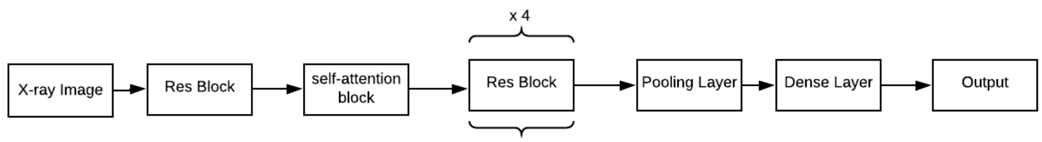

Figure 4: Architectural layout of EMIXER image generator $\mathrm{D_{Image}}$

X-ray Report Generator (F): We describe the architecture of the X-ray report generation module in Figure 5. The report generation component contains three different sub-components: (a) Image encoder CNN (b) Sentence LSTM (c) Word LSTM. The image encoder CNN takes an X-ray image as input and produces feature vectors. This CNN model is pre-trained on X-ray images **I** using a DenseNet model. The sentence LSTM produces topic vectors $\mathbf{t_i}$ which are used as input for word LSTMs to produce the words. After the word LSTM produces all the words, the words are combined to create the final report **S**.

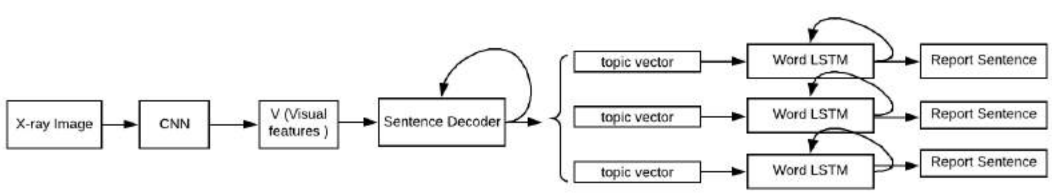

Figure 5: Architectural layout of EMIXER report generator F

X-ray Report Discriminator ($\mathrm{D_{report}}$): As we show in the figure 6, the X-ray report **S** is passed as input to the LSTM. LSTMs have been used to represent paragraphs and sentences to produce context vectors. We use the final representation obtained from the LSTM and pass that to a linear layer. This is finally passed through a softmax layer to obtain the probability of real or fake.

Joint Discriminator ($\mathrm{D_{joint}}$) As shown in figure 7, the X-ray report **S** and image **I** are used to create a joint embedding. X-ray images **I** is passed through CNN to obtain an X-ray image feature vector $\mathbf{f_v}$. X-ray report **S** is passed through a LSTM to obtain the final representation of the report $\mathbf{f_s}$. The feature vectors are concatenated together to obtain a joint embedding $\mathbf{C}_{\text{joint}}$. This is finally passed through a linear layer and softmax layer to obtain the probability of embedding being real or fake.

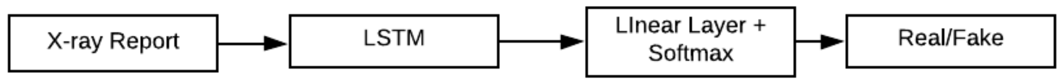

Figure 6: Architectural layout of EMIXER text discriminator D_{Report}

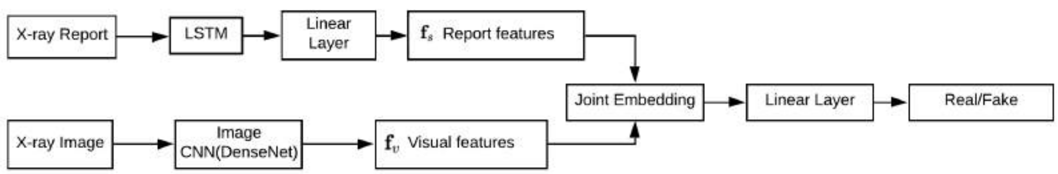

Figure 7: Architectural layout of EMIXER image generator D_{Joint}

A.3. Appendix B Experimental Details

A.3.1. Dataset Details

We used MIMIC-CXR dataset consisting of X-ray images and reports Johnson et al. (2019). This data set was collected from Beth Israel Deaconess Hospital. We apply pre-processing to remove duplicated samples from this dataset. The Radiology reports typically contain an impression and findings section. We extracted the finding section from the report for training our models. We apply tokenization and only keep tokens with at least 6 occurrences in the corpus for training purposes.

A.3.2. Architecture and hyperparameters

We use Adam optimizer with a learning rate of $5 \cdot 10^{-5}$ for the generative model and $2 \cdot 10^{4}$ for the discriminators for training EMIXER Kingma and Ba (2014). We staggered discriminator steps and generator steps in 2:1 ratio which led to 400k (800k) generator (discriminator) steps. This helps the discriminator improve it's parameter update process faster compared to a generator. We fix our batch size at 512 while training. We use a noise vector of 120 dimensions as input for the generator. We also use spectral normalization for the layers in the generator and discriminator in the training process. All the models generate $128 \times 128 \times 3$ X-ray images. We obtain partially labeled data sets for the self-supervised experiments by randomly selecting 30% of the samples from each class. We rotate the images and use the rotation angles as labels for self-supervision Gidaris et al. (2018).

A.3.3. Evaluation Metrics

Fréchet Inception Distance (FID score): We first pass real data and generated samples embedded in a specific layer of special pre-trained Inception network on chest X-ray images instead of ImageNet Heusel et al. (2017). Then, a multivariate Gaussian is fit to the data and the distance computed as $\mathrm{FID}(x, g) = \|\mu_x - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_x + \Sigma_g - 2\left(\Sigma_x \Sigma_g\right)^{\frac{1}{2}}\right)$ where μ and Σ denote the empirical mean, covariance and subscripts x and g denote the real and generated data respectively.

A.4. Results

A.4.1. Phenotype Classification from X-ray Images with augmented data

We report the performance of EMIXER and the baseline models for different phenotype detection from chest X-ray images. The setup for this experiment is similar where we train two models on real X-ray images and generated X-ray images. These trained models are evaluated on held-out X-ray images. The performance of the test X-ray images are reported in Table 8

Table 8: Performance of X-ray image classification using synthetic X-ray

Dataset	Method	Cardiomegaly	Consolidation	Pleural Effusion	Pneumothorax	Pulmonary Edema
MIMIC	Real data [100k images]	0.812	0.847	0.753	0.735	0.732
	CoGAN [100k images]	0.741	0.817	0.708	0.713	0.682
	JointGAN [100k images]	0.732	0.785	0.724	0.681	0.713
	EMIXER [100k images]	0.784	0.734	0.728	0.715	0.718

A.4.2. Performance comparison of augmented data to real data

We performed an experiment to evaluate augmented datasets in comparison to real datasets of similar size. In this setup, we keep the total size of the dataset constant at 100k and change the ratio of real and synthetic images. We present the results of this experiment in Table 9. This experiment evaluates the performance of augmented datasets where the total dataset size is low. We show that even when we use 80% fewer real images, augmented datasets only show 6% decrease in performance. This shows that even with low-data availability, synthetic data augmentation can perform competitively compared to models trained only on real X-ray images.

A.4.3. Additional generated data samples

In figures 8 and 9, we show additional generated X-ray image,report pairs in comparison to real X-ray image and report pairs. Figure 10 and 11 show comparison of real X-ray images to synthetic X-ray images. Finally, figure 12 shows more synthetic X-ray images.

Table 9: X-ray image classification performance comparison with EMIXER augmented data. Dataset size at 100k while reducing the amount of real images in the augmented dataset. In this table, R indicates Real data and S indicate Synthetic data.

Method	Data	AUC	Acc
Only Real	R100k	.824	.846
JointGAN	R90k + S10k	.796	.813
	R80k + S20k	.778	.801
	R60k + S40k	.745	.764
	R20k + S80k	.717	.732
CoGAN	R90k + S10k	.784	.808
	R80k + S20k	.771	.796
	R60k + S40k	.736	.757
	R20k + S80k	.712	.746
SMGAN	R90k + S10k	.794	.812
	R80k + S20k	.764	.783
	R60k + S40k	.742	.763
	R20k + S80k	.723	.742
EMIXER	R90k + S10k	.808	.828
	R80k + S20k	.792	.821
	R60k+ S40k	.773	.796
	R20k + S80k	.756	.774

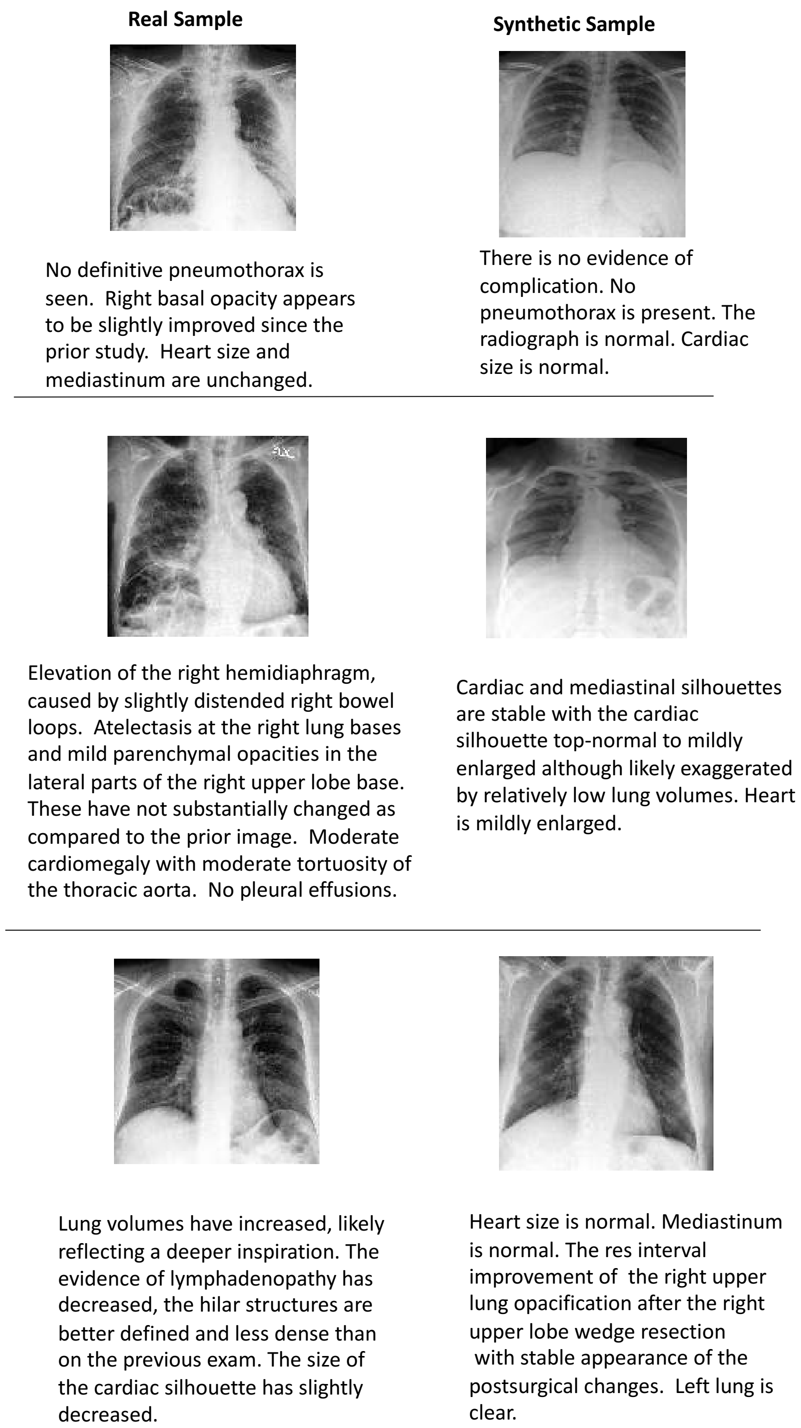

Figure 8: Comparison of Real X-ray image and report pairs with generated X-ray images, reports pairs

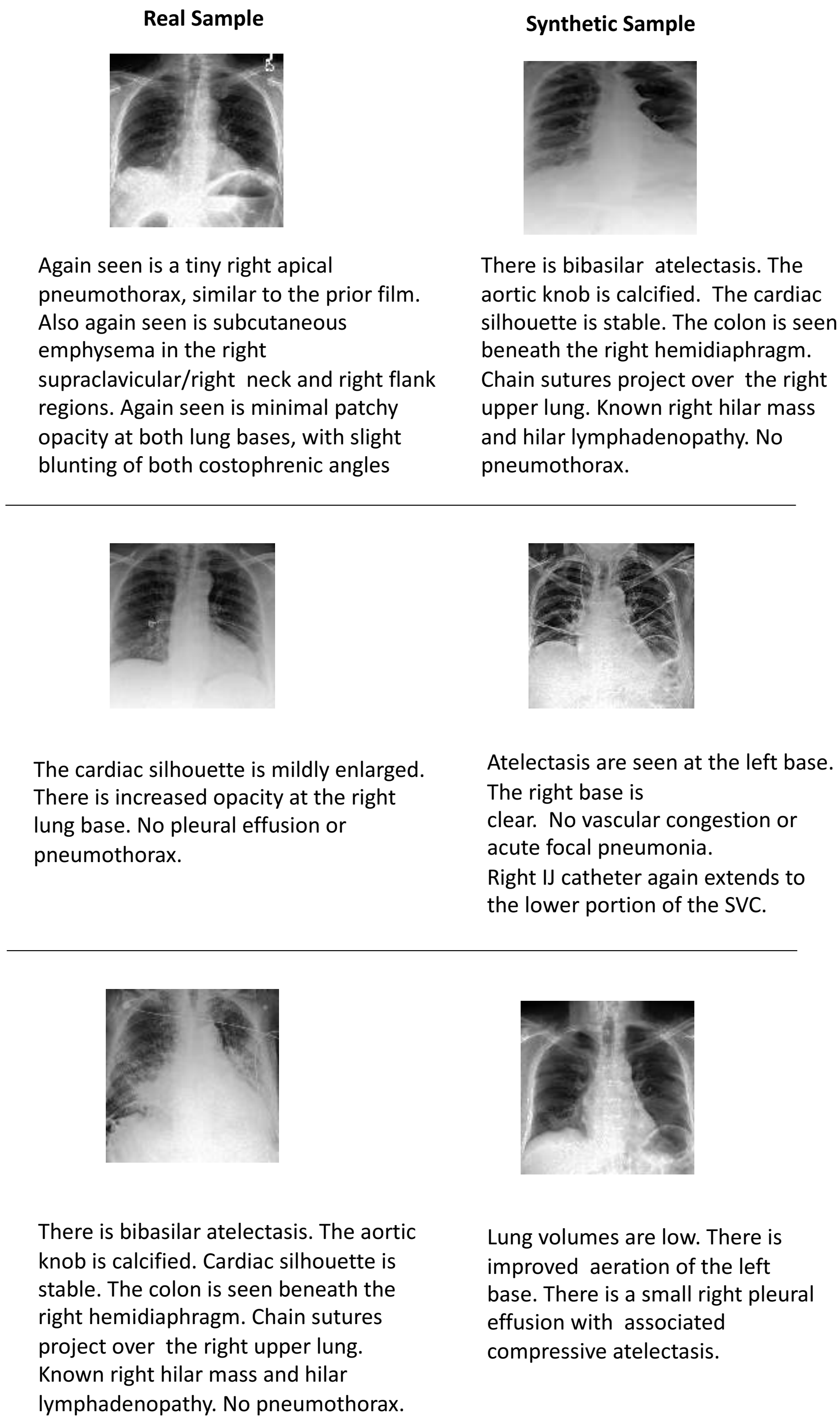

Figure 9: Comparison of Real X-ray image and report pairs with generated X-ray images, reports pairs

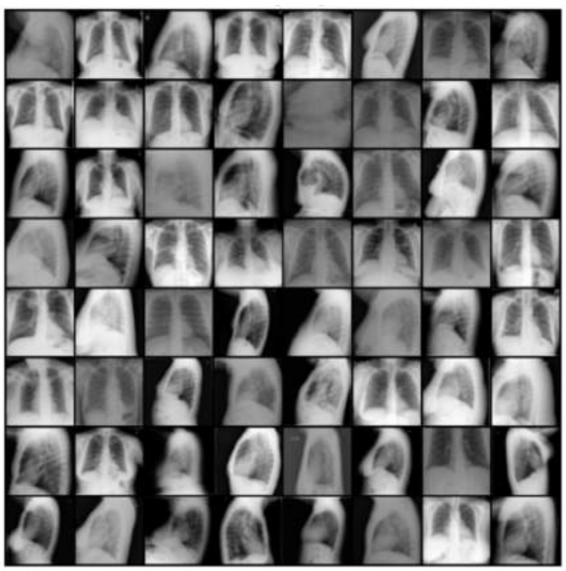

Figure 10: figure
Real X-ray images

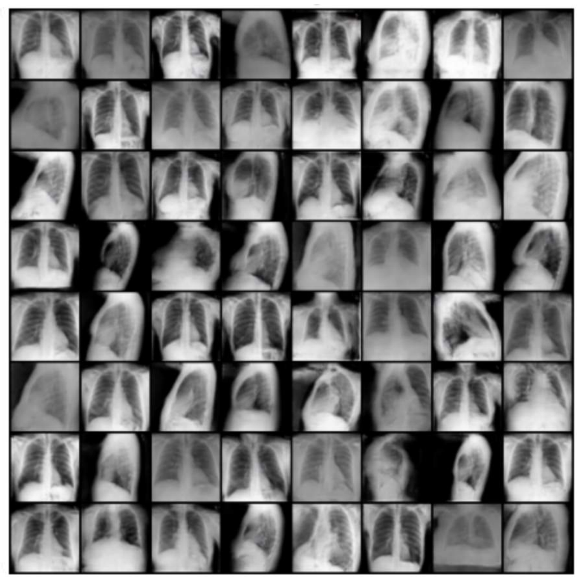

Figure 11: figure
Synthetic X-ray images

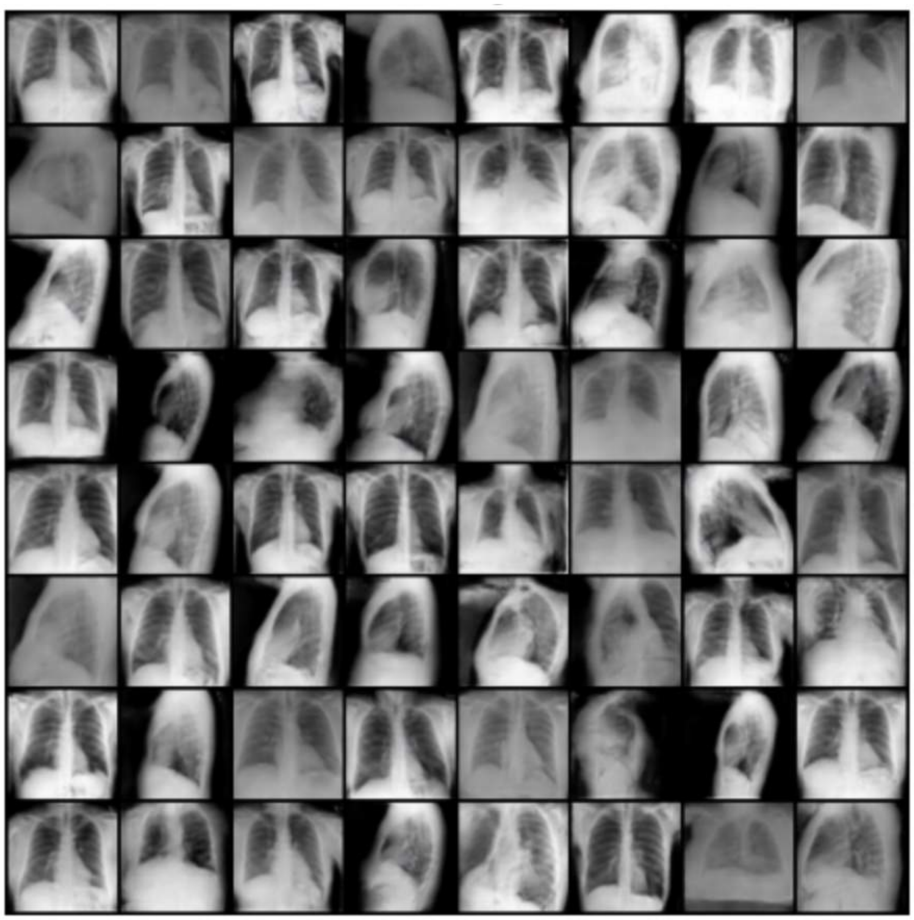

Figure 12: Samples of Synthetic X-ray images