RayGen: A Vision-Language Masked Generative Transformer for Chest X-Ray Generation

Faraz Ali, Yasamin Nouri Jelyani, and Pooja Ravi

Department of Computer Science
University of Toronto
{farazali,yasamin,poojaravi}@cs.toronto.edu

Abstract

The lack of large-scale, accessible, and reliable labeled data in the medical imaging domain is a problem that plagues the development of intelligent diagnostic models. Medical image datasets tend to exhibit scarcity in data volume and quality in terms of class distribution. These challenges can be solved by augmenting model training data with synthetic images. To ensure a high-quality result is achieved with each synthetic image, it is useful to incorporate multimodal characteristics. We investigate the ability of a novel masked generative transformer model, RayGen, on the generation of chest x-rays (CXRs) using vision-language data. We use Anterior-Posterior (AP) views and the corresponding radiology reports from the MIMIC-CXR dataset for training. RayGen utilizes vision-language embeddings to learn the distribution of visual tokens of an image encoded by a VQ-GAN model via masked visual token modeling. RayGen achieves a high fidelity with a Frechet-Inception Distance (FID) of 7.9, comparable to state-of-the-art CXR generation models, but exhibits a low generation diversity with an average Multi-scale Structural Similarity Index Measure (MSSIM) of 0.79. The Area Under the Receiver Operating Characteristic (AUROC) of a downstream DenseNet-121 model for 10-class chest pathology classification decreases by 0.005 and 0.03 overall when 50% and 100% of the training data size is added as synthetic data, respectively. The AUROC for specific pathologies does however improve, indicating that RayGen can learn some semantic features specific to certain classes. Limitations of the study include a lack of compute for larger and more diverse training data inclusion, longer training schemes, and larger architectures. Future work involves developing the model with larger datasets and manipulating the model's attention to learn lower-level class-specific features.
Our code is available at: https://github.com/farazali7/multimodal-xray.

1 Introduction

The development of accurate and reliable diagnostic models in medical imaging, particularly in chest X-ray (CXR) analysis, is often hindered by a scarcity of high-quality, labeled datasets. This shortage is further compounded by challenges such as data privacy concerns, inconsistent data collection practices, and the inherent complexity of medical data labeling. Moreover, the computational demands of processing large-scale medical datasets pose additional challenges, limiting the scope of research and development in this field. Furthermore, there is usually a discrepancy in the quality of data present for the development of the diagnostic models. This is because most models aim to learn representations of a disease or positive event class. However, as these events tend to be rarer than their normal case counterparts, most medical image datasets exhibit an imbalance in the classes present. A potential way to mitigate both the data volume and quality scarcity is to employ synthetic

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

medical data during the development of these models. Doing so would require the generation of high-fidelity, highly usable medical images for downstream tasks. Recognizing the importance of addressing these challenges, our research proposes RayGen, a novel model that synthesizes chest X-ray images by integrating vision and language processing techniques. This combination allows us to harness the knowledge encapsulated in existing datasets, such as MIMIC-CXR, to generate new, synthetic images. The RayGen model is further distinguished by its use of a masked generative transformer, a technique that merges the distinct modalities of image and text data, offering a unique solution to the problem of data scarcity in medical imaging. This study is the first of its kind to employ a multi-modal masked transformer model trained on CXR images and radiology reports, to generate synthetic chest radiographs using text prompts. By generating diverse and representative images, RayGen can enhance the accuracy and generalizability of downstream diagnostic models. This advancement is particularly crucial in medical fields where diverse training datasets are vital for accurate and reliable diagnostics.

2 Related Work

2.1 Synthetic CXR Generation Models

2.1.1 RoentGen: Diffusion Model

The RoentGen model is a synthetic image generation model developed to generate high-fidelity chest X-ray images conditioned on text prompts. This model adapted a latent diffusion model, using the Stable Diffusion pipeline architecture. The model utilizes a conditional U-Net to denoise a matrix of random Gaussian noise, conditioned on embeddings created from short medical free-text prompts generated from the CLIP text encoder. The denoised latent vector is then mapped back to pixel space by the decoder of the variational autoencoder. RoentGen shows promise as a tool for data augmentation in medical imaging analysis, demonstrating a significant 5% improvement in the DenseNet-121 downstream classifier performance when trained on both synthetic and real images. The RoentGen model's experiments were conducted using 64 A100 GPUs (1). As RoentGen operates in the same domain as our model, it serves as an excellent model to compare results with given its vastly different generation approach.

2.1.2 UniXGen: Bidirectional Transformer

The UniXGen model (2) is a transformer-based auto-regressive synthetic image generation model designed for bidirectional chest X-ray and report generation. This model generates radiology reports from chest x-ray images and can generate new views of the chest x-ray images given radiology reports as well as chest x-ray images from a different view. It employs VQ-GAN (Vector Quantized Generative Adversarial Network) for image tokenization, converting input MIMIC-CXR images into 1024 discrete visual tokens using a codebook. For text embeddings, the model uses a byte-level BPE (Byte Pair Encoding) tokenizer to process radiology reports. The UniXGen model also adapts an efficient transformer model called a Performer. The VQ-GAN has a better performance on tokenizing images than other approaches such as Discrete VAE's (3), VQ-VAE (4) and even ViT-VQGAN (5) and has been a preferred method for creating image embeddings in the Muse model as well (6).

2.2 Masked Generative Transformer Models

2.2.1 MaskGIT: Masked Generative Image Transformers

The MaskGIT model utilizes a masked modeling approach similar to BERT in Natural Language Processing (NLP) for image generation. The authors employ a 2-stage approach that consists of tokenization and Masked Visual Token Modeling. They ideate a mask design algorithm (a mask scheduling function) that controls the ratio of masking in the tokens. Upon comparing a few methods they adopt the cosine function. In the second stage, the model features a bidirectional transformer decoder that significantly accelerates the image generation process(7). The authors achieve this by encoding an input image via a VQ-GAN model into discrete tokens. This ensures that conditional dependency is bidirectional thereby better facilitating context capturing for all the visual tokens. A transformer then learns to unmask these tokens to produce the original latent representation. The VQ-GAN decoder is used during inference time to decode transformer predictions to a new image.

It leverages parallel decoding, allowing it to significantly boost the speed and efficiency of the generation process. With this, they achieve a significant decrease in FID and an increase in Inception score for both 256x256 and 512x512 image resolution as compared to the existing state-of-the-art.

2.2.2 Muse: Text-to-Image Masked Generative Transformer

The Muse model is a state-of-the-art text-to-image Transformer model notable for its efficiency, which has outperformed diffusion and autoregressive models in natural image generation. It builds upon the work in (7), by adding a two-staged approach where the first transformer performs unmasking of a lower-resolution 224x224 image, while a larger transformer then performs super-resolution to generate a higher quality 512x512 unmasked image. The authors also introduce the inclusion of textual prompts to guide the generation process. Muse is trained on masked modeling tasks using pre-trained large language model (LLM) embeddings, enabling fine-grained language understanding and high-fidelity image generation (6).

3 Methods

3.1 Model

RayGen consists of several parts, some trained and others adapted from pre-trained models. Figure 1 provides a visual representation of the architecture. Our RayGen model is heavily inspired from the work done in the Muse model. Namely, we incorporate a very similar setup as the base Muse model, with a VQ-GAN used for encoding an decoding an image to and from the discrete visual tokens and a transformer decoder trained to learn the probability distribution over the token. We build upon the work of (6) by applying the architecture in a single-stage, higher resolution (i.e., 512x512), changing the setup of layers within the transformer model, utilizing a VQ-GAN pretrained on a CXR dataset, and employing a CXR BERT for text tokenization.

At a high level, a given CXR in the VQ-GAN encoder model, was already pre-trained on the MIMIC-CXR dataset by the authors of the UniXGen model (2). The image embeddings are randomly masked according to a cosine schedule, flattened into a sequence, and then passed through an embedding layer before being added with learnable positional embeddings. The impressions section of a corresponding radiology report is also encoded via a pre-trained CXR BERT model (8) and projected to the hidden dimensionality of the transformer decoder. The transformer decoder takes the masked image embeddings sequence as queries and the report embeddings as the context through several stacked layers of attention to eventually output a reconstruction of the latent representation of the image. The different parts of the architecture are described in detail below.

3.1.1 Text Encoder

The model CXR-BERT from Microsoft (8) is a specialized pretrained text encoder that excels at identifying the underlying semantics in radiology reports. The vocabulary for this model has been obtained from PubMed abstracts, MIMIC-III notes, and MIMIC-CXR reports. A randomly initialized BERT model is trained using Masked Language Modeling or MLM on the aforementioned datasets. For the CXR-BERT specialized model, they further pretrain exclusively on the MIMIC-CXR data for domain expertise. Overall, the pipeline involves three stages of pretraining - vocabulary-based, MLM, and radiology section matching (8). They employ a contrastive learning procedure in radiology section matching to penalize IMPRESSIONS-FINDINGS pairs from different samples while favoring the pairs from the same sample. This helps the model learn the matching representations well. In RayGen, the encoded text from CXR BERT is further projected to match the hidden dimensionality of the subsequent transformer decoder.

3.1.2 Image Tokenization

In our model, we leverage a VQ-GAN (5) model pre-trained by (2) on MIMIC-CXR for image tokenization. The VQ-GAN model first learns a fixed-size (Z) codebook which is created by quantizing the encoded representations of an input image via the generator of a GAN model. The decoder is used to improve the learned quantized vectors in the codebook. A transformer model is subsequently trained on the codebook to learn the probability distribution of tokens in the codebook. The model from (2) takes an input CXR, $x \in \mathbb{R}^{3 \times H \times W}$, and tokenizes it into a latent encoding,

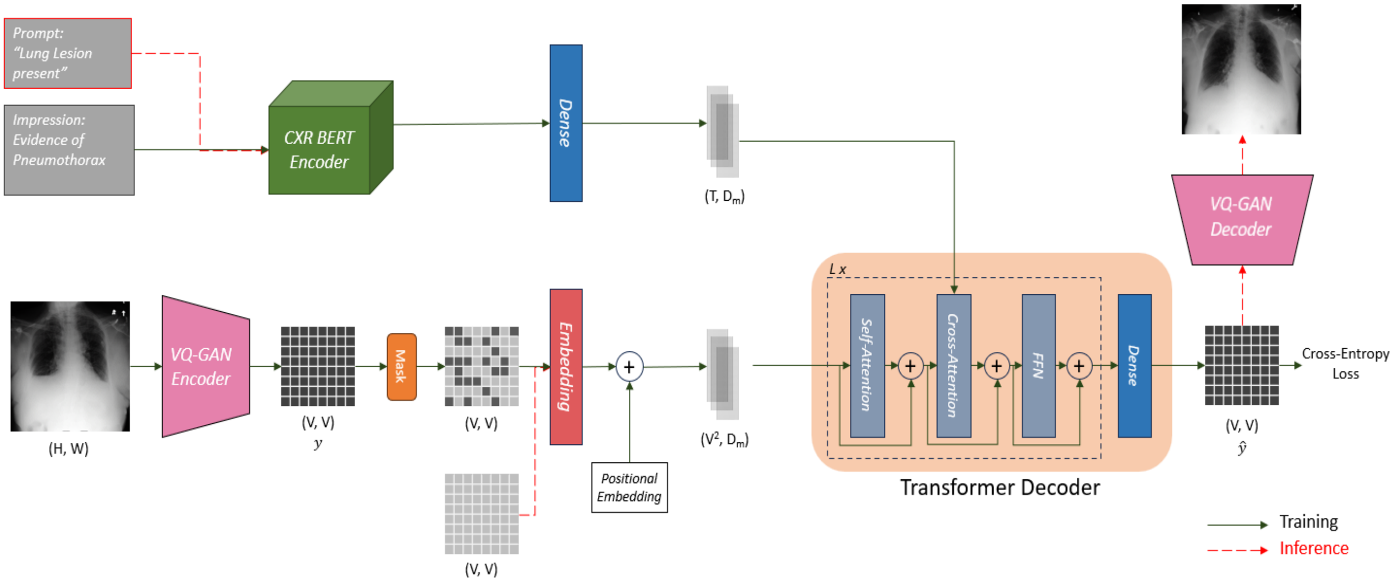

Figure 1: RayGen Model Architecture: The model seeks to recover a masked token in an encoded latent representation of an input CXR using the masked CXR embedding and context from the textual clinical impressions. Dark green arrows represent the flow of data during model training which begins with tokenizing and masking a CXR and encoding the clinical impressions section. Dashed red arrows represent the flow of data during inference, which begins with a fully masked latent representation of the image and a prompt, and ends with decoding the unmasked output via the VQ-GAN decoder.

$y \in \mathbb{R}^{V \times V}$, where $V = 32$, with each pixel having a value from the discrete visual tokens from a codebook of size $Z = 1024$.

3.1.3 Masking

The latent encodings are masked according to a cosine schedule as done similarly in (6). This is done by uniformly sampling a rate, $r \in [0, 1]$, which is then used to mask the tokens with a probability of $p(r) = \cos(\frac{\pi}{2} r)$. The $p(r)$ cosine function is used since it has an average masking probability of 0.64, which makes it biased towards masking higher fractions of the total input sequence, and thus, creates a more difficult unmasking task for the transformer decoder.

Let $\mathbf{Y} = [y_i]_{i=1}^{V^2}$ represent the set of latent encodings after being flattened to sequence. The corresponding mask, $\mathbf{M} = [m_i]_{i=1}^{V^2}$ is generated by randomly selecting $\lceil V^2 p(r) \rceil$ indices to set to 1, while all other indices are set to 0. Then, the masked latent encodings, $\mathbf{Y}_M$, are generated by assigning y_i to index i if $m_i = 0$, otherwise, a special mask token, $[\text{MASK}]$, that lies outside the original codebook size is assigned. Since each token in $\mathbf{Y}_M$ is a discrete scalar, a learnable embedding layer expands the dimensionality to the hidden dimensionality of the transformer decoder before learnable positional embeddings are added.

3.1.4 Transformer Decoder

The transformer decoder follows mostly the same architecture found in (7). It consists of a series of L stacked layers of multi-headed self-attention, multi-headed cross-attention, and a feed-forward network (FFN). In our work, $L = 6$ and the number of heads in each attention mechanism is 8. The hidden dimensionality of the transformer decoder in our work is set to 1024. There are skip connections between each sub-layer to help extracted information signals propagate better. The FFN is composed of two dense layers with pre-layer normalizations before each, and a Gaussian error linear unit (GELU) nonlinearity in between. The FFN projects the input from the transformer decoder's hidden dimensionality to a higher dimensionality of 4096. At the end of the stacked layers, a final dense layer projects the extracted embeddings to a latent space equivalent to the codebook size.

The transformer decoder learns to predict the probability distribution over the ground-truth tokens as $\hat{y}_i = p(y_i|\mathbf{Y}_M)$. Note that the decoder has access to information from all visual token positions when predicting probabilities, making it bidirectional.

3.2 Loss Function

The decoder is trained via the cross-entropy loss function, J, between one-hot encodings of the original latent tokens from the VQ-GAN encoder and the predicted token probabilities from the final dense layer:

$$J = -\sum_{\forall i \in [1, V^2], m_i = 1} y_i \log(\hat{y}_i) \quad (1)$$

The model computes the loss and backpropagates gradients only for masked tokens, hence, the decoder ends up learning the probability distribution over the masked tokens. By training the model to minimize the cross entropy between the masked tokens and their true values, the model should learn to effectively incorporate textual and any unmasked visual information to generate masked visual content. This makes the model particularly well-suited for generation during inference.

3.3 Inference

During inference, a fully masked latent representation, as well as a user-specified prompt, are input into the same encoding pipeline as described above. Given a certain number of steps K, an initial sampling temperature, T, and a selection threshold, α, the model iteratively unmasks a certain number of tokens. At each step, the model predicts the probabilities for all tokens, but keeps only a subset of the tokens with the highest probability scores. Furthermore, for each token, it retains only the top $1 - \alpha$ fraction of all the probabilities. The rest of the tokens are re-masked at the start of the next iteration. The masking rate, r, is set to a linear relationship with the current timestep, k, so that the number of remasked tokens at each step decreases. Thus, the model generates an unmasked result in T timesteps. A visual representation is shown in Figure 8.4 in the Appendix.

4 Experiments & Results

4.1 Dataset

Our primary dataset was the MIMIC Chest X-ray (MIMIC-CXR) Database v2.0.0, a comprehensive collection of chest radiographs paired with free-text radiology reports. This dataset, sourced from the Beth Israel Deaconess Medical Center in Boston, MA, comprises 377,110 images corresponding to 227,835 radiographic studies (9). These were sourced from the MIMIC CXR JPG database (10), which provides images in JPG format - a lower resolution alternative to the original DICOM format. This selection was influenced by the practical constraints of storage and processing capabilities of our study. We utilized the most common view available in the MIMIC-CXR database (AP views), and selected only one image when multiple were available for the same view, to train our generative model. In addition to image data, we extracted and utilized the interpretation section of the radiology reports associated with each X-ray. To ensure the data was manageable for our model and within the memory limitations of our GPU, we limited these text sections to 256 tokens from the Interpretation section of the reports. For this study, our generative model was trained using a single RTX 4090 GPU with 24 GB of memory.

4.2 Downstream Classification

To assess the efficacy of synthetic images in enhancing the performance of a downstream classifier model, we utilized a DenseNet-121 classifier, initially pre-trained on the ChexNet model (11) (12), for classifying anterior-posterior (AP) and posterior-anterior (PA) chest X-ray images into 14 distinct classes. Our focus, however, was on adapting this model for a slightly different task: classifying images into 10 specific classes. Consequently, we did not utilize the pre-existing classification layer parameters, as they were tailored for a different classification scheme.

Our primary dataset for fine-tuning the model comprised AP and PA images from the p10 subgroup of the MIMIC CXR dataset. This subset, drawn from the larger MIMIC-CXR JPG database, included images from 17,024 studies. The task involved multi-label classification into 10 categories of findings, namely: Atelectasis, Cardiomegaly, Consolidation, Edema, Fracture, Lung Lesion, Lung Opacity, Pleural Effusion, Pneumonia, and Pneumothorax. Initially, the model was fine-tuned using 4,684 real images from this dataset.

To evaluate the impact of synthetic images, we conducted two parallel fine-tuning experiments. In the first round, the model was trained on a dataset augmented with an equal number of synthetic and real images (4,673 synthetic and 4,684 real images). In the second round, the proportion of synthetic images was reduced to 50% of the real images (2,337 synthetic and 4,684 real images). This approach allowed us to compare the classifier's performance across different ratios of real to synthetic images.

The effectiveness of the model was assessed based on the macro and class-wise Area Under the Receiver Operating Characteristic (AUROC) values obtained from our model and compared with those reported in related works, as shown in Table 1. This metric is essential for gauging the classifier's proficiency in distinguishing between different classes. The AUROC reflects the two-dimensional area underneath the curve plotted with the False Positive Rate (FPR) on the x-axis against the True Positive Rate (TPR), or recall, on the y-axis. As seen from the results, the macro AUROC decreased by 3% when an amount of synthetic data equal to the training set was added and decreased only 0.5% when half that amount was added.

Table 1: Comparing Average AUROC values of the downstream classifier. The P10 subset of the MIMIC CXR data was used to train the downstream classifier for RayGen. The results were tested with 605 real chest x-ray images. The P19 subset of the MIMIC CXR data was used to train the RoentGen classifier (1). The model was trained to classify the images into 10 labels in the multilabel configuration.

	RayGen		
Average AUROC	Real Images	Real + Synthetic Images	RoentGen Real + Synthetic images
4673 Synthetic, 4684 Real images	0.71	0.68 (0.03 ↓)	0.77 (1.1k Synthetic, 1.1k Real images)
2337 Synthetic, 4684 Real images	0.71	0.705 (0.005 ↓)	0.77 (1.1k Synthetic, 1.1k Real images)

4.3 Fidelity & Diversity

The quality of the generated CXRs must ideally be evaluated on various dimensions to ensure they are capable of representing real CXRs in any extenuating scenarios. To that end, we employed the Frechet-Inception Distance (FID) measure to assess the fidelity of the synthetic CXRs and the Multi-scale Structural Similarity Index Measure (MSSIM) for assessing diversity. The FID metric is commonly used for image generation fidelity evaluation (1)(2) and requires comparing the image statistics that are derived from a pre-trained deep neural network between an original data distribution and a synthetic one. A lower score indicates higher similarity which means the synthetic distribution matches well with the original one. The MSSIM is used for comparing the visual similarity between multiple images generated via the same prompt from a generative model. A lower score also indicated a higher diversity of outputs which is generally attractive for generative models.

We assess FID by reusing a subset of 2,568 images from the p10 subset of MIMIC-CXR as the original dataset and generating synthetic versions of each with its corresponding Impressions section as the prompt. The pre-trained embedding extractor model was the CheXpert model from (13). We also computed MSSIM over pairwise comparisons of two sets of synthetic images, both generated from the 2,568 subset from p10 used for FID. Results are shown in Table 2. From the results, it is seen that RayGen can achieve a decently low FID score comparable to other methods, but lacks diversity in outputs from a higher MSSIM of 0.79.

4.4 Qualitative Results

Figure 4.4 showcases synthetic examples for each of the 10 classes used in downstream classification generated by the model with decoding parameters: $T = 1.3, K = 3, \alpha = 0.5$.

Table 2: Average AUROC, MSSIM and FID scores. The AUROC is from experiments with some mixture of original and synthetic data used for training, and the RoentGen and UniXGen metrics are from 14-class classification whereas RayGen is reported for only 10 classes.

Model	FID	MSSIM	AUROC
RoentGen	3.6	0.32	0.77
UniXGen	18.9	-	0.73
RayGen (Ours)	7.9	0.79	0.71

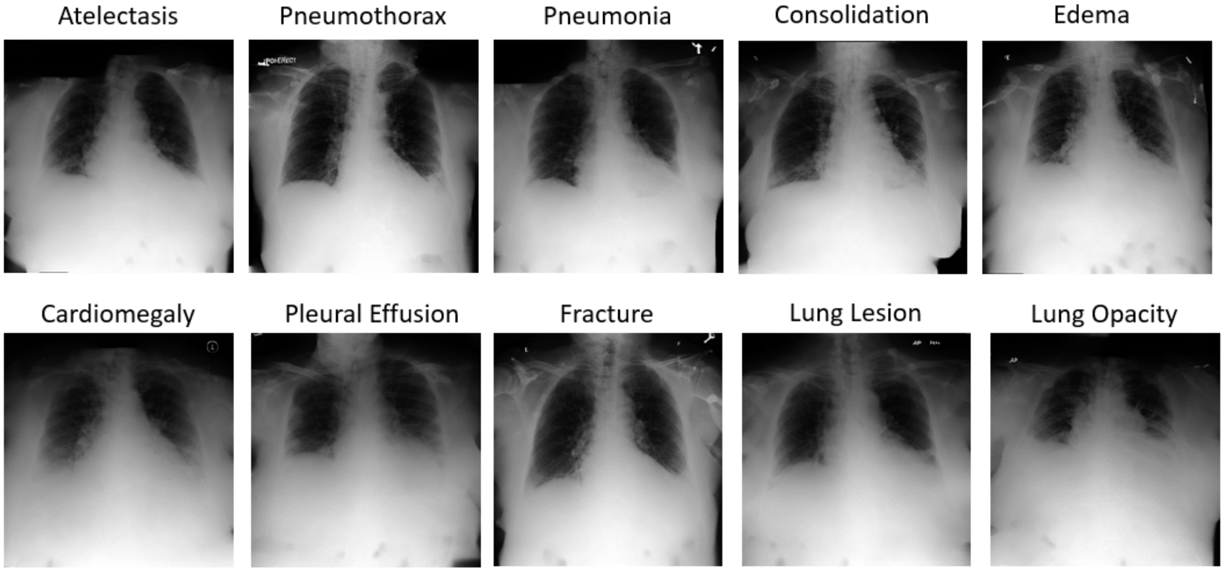

Figure 2: Synthetic CXRs Corresponding to Various Pathologies. This image represents a series of CXRs generated by RayGen given the text prompts above each respective image.

5 Discussion

5.1 Downstream Classifier AUROC

Based on the visual quality and diversity of the synthetic CXRs from Figure 4.4, it is reasonable to see the apparent decrease in macro-AUC of the classifier from the incorporation of synthetic images. This is likely because while the generated images have high visual similarity to real CXRs from the dataset, the model has not yet captured the semantic features between each of the classes. As a result, the inclusion of this level of synthetic data likely introduces more noise in the classification process for the downstream model. For AUROC improvement, it is required that the generative model be trained to better delineate between specific visual features of the different pathological classes. For some classes, the AUROC did however marginally improve, perhaps indicating that the model could capture some features of those classes in its generation process. These are shown in Table 3 in the Appendix. This finding indicates that synthetic images can potentially enhance the classifier's ability to correctly identify certain conditions. Moreover, our results did not reach the performance levels reported in the RoentGen study average AUROC performance, which used stable diffusion-generated images. This difference is likely due to several differences in our study, primarily the constrained computational resources that restricted our ability to train larger and potentially more effective models. The disparity could also be attributed to differences in computational capacity and the datasets used; the RoentGen study used less images (1.1k) from the p19 dataset for classification, whereas our study utilized more images (4.6k) from the p10 dataset.

5.2 Quality of Synthetic Images

As shown in Table 2, RayGen was able to sample and generate synthetic CXRs from a distribution fairly close to the original data distribution, as made evident by a low FID score. However, when accounting for the high MSSIM score and the visual similarity in outputs in Figure 4.4, it is clear that the model failed to learn some of the important visual differences that exist in each pathological class. This result indicates that the model training was useful for learning global level features that were indicative of a general CXR, but did not encourage learning of lower-level semantics. More nuanced model architectural choices must be made in the future to encourage learning of these more specific features per class. One way could be to train the model in a two-staged approach, that emphases learning features at different scales, as done in the paper introducing the Muse model (6).

5.3 Limitations & Future Work

GPU RAM significantly influences the quality of the training procedure, but our limited capacity GPU posed a constraint to our study. We trained on a single GPU with 24GB of RAM, leading to compromises in image resolution and dataset size, affecting our training capacity and hindering our results. This limitation also prevented us from being able to make equivalent comparisons with other baselines. Another limitation of the project was utilizing a single image from all AP views in each study for training RayGen. As a result, the model had less diverse information when learning the features of CXRs for a generation. A consequence of this can be seen through the low diversity of the model. This may be mitigated by including training data composed of multiple views, which has been shown to increase MSSIM (1). Another avenue for future exploration could be to try fine-tuning the VQ-GAN decoder after the main RayGen model is trained. This is because although the VQ-GAN model's decoder was pretrained to reconstruct input CXRs from the tokenized sequence, it was not necessarily trained to handle new unmasked sequences. Thus, fine-tuning the decoder might help improve the fidelity and diversity of the outputs. A third limitation related to using only a single view (AP) is that the model was trained on less data than was available. While the quality of data mattered, the quantity of data also significantly impacted the learning process for the model. A solution to this for our project could be to instead focus on fine-tuning the initially trained RayGen model on a smaller subset of the data belonging to a specific downstream pathology class. This may allow the model to better learn features particular to that class. Furthermore, the model was trained to directly generate a high-resolution 512 x 512 image. This resulted in the model favoring global features of CXRs over small-scale semantics. This is made evident by the high fidelity but low diversity of the model. Future work can attempt to remedy this situation through various methods. One might be following a paired lower resolution and super-resolution training approach as done in (6). Another way could be to limit the receptive fields of the attention mechanisms in the decoder to a certain window size such that the transformer can learn more specific semantics for a given CXR. Finally, the current model is specific to a single imaging modality. This limits its usefulness in other domains, such as image diagnostic settings for MRI, CT scans, or ultrasounds. To utilize the model for other modalities, it would have to be retrained on another dataset of that modality, which can be an expensive task.

6 Conclusion

RayGen demonstrates a great advancement and potential for efficient chest x-ray image generation. By synthesizing high quality and diverse chest x-ray images, we can address the persistent challenge of data scarcity in training downstream classifiers. As we continue to refine the performance of the RayGen model, we anticipate that its contributions to enhancing the performance of predictive tools in medicine can impact patients worldwide and significantly contribute to better health outcomes. While RayGen demonstrated capability to generate high-fidelity CXRs, it lacked diversity in its generation process. This negatively affected the usability of its generated images for downstream tasks. However, future work can improve upon this method by utilizing larger, more diverse data, and experimenting with spatial inductive biases which can help the model learn class-specific features better in its attention mechanism.

7 Team Contributions

Faraz worked on the image encoder for the initial approach, as well as the second approach (RayGen)'s VQ-GAN image encoder, transformer decoder layers, training and inference (generation) code. He also worked on implementing the masking and objective functions, as well as the sampling algorithms for decoding. He also worked on composing all of the different parts of the models (for both approach 1 and RayGen) into a unified model. He also worked on adding cloud-based logging capability for experiments and setting up the general project file codebase. Faraz also contributed in writing of the proposal as well as final report & maintaining the GitHub repo.

Yasamin worked on generating text embedding and utilizing the CXR BERT model. She also worked on MIMIC CXR Data processing, retrieval and cleaning. She also Generated JSON for different representations of the data used for training the model. Also, she worked on training the densenet121 mode as well as performing inference on this downstream classifier using real and synthetic images. She also performed AUROC evaluations. Yasamin also contributed in Writing of the proposal as well as final report and maintaining the Github repo.

Pooja contributed to the initial approach by ideating a custom U-Net-based image decoder for image synthesis and performing the initial training runs. She was also involved in experimenting with the VQ-GAN architecture for creating image embeddings in the main approach. She further helped with obtaining, processing, and storing MIMIC-CXR data. Moreover, she aided in creating a baseline classifier using torchxrayvision for the downstream classification task, and writing parts of the project proposal & final report as well as maintaining the GitHub repository.

8 Appendix

8.1 Code

Our code is available at: https://github.com/farazali7/multimodal-xray

8.2 Additional Quantitative Results

Table 3: Comparing class-wise AUROC values of a downstream classifier. Ate.: Atelectasis, Car.: Cardiomegaly, Con: Consolidation, Ede.: Edema, Fra: Fracture, Les: Lung Lesion, Opa: Lung Opacity, Eff: Pleural Effusion, monia: Pneumonia, orax: Pneumothorax

Data	Ate.	Car.	Con.	Ede.	Fra	Les.	Opa	Eff.	monia	orax.
4684 Real images	0.7149	0.7605	0.6761	0.8943	0.6871	0.5857	0.5981	0.8530	0.7139	0.6574
4673 Synthetic, 4684 Real images	0.7062	0.7080	0.5933	0.8547	0.6138	0.5380	0.6147	0.8723	0.6849	0.6264
2337 Synthetic, 4684 Real images	**0.7151**	0.7371	0.6362	0.8704	0.6510	**0.6015**	**0.6200**	**0.8592**	0.7103	0.6496

8.3 Metrics Used

8.3.1 Frechet Inception Distance (FID)

The Frechet Inception Distance is widely used in image generation tasks to measure realism and variety among the synthesized images. It uses the pretrained Inception model to extract features from the images and uses the formula mentioned in Equation 1. Note that a lower FID score is desirable as it corresponds to more realistic images.

$$FID = \|\mu_1 - \mu_2\|^2 + T_r(C_1 + C_2 - 2\sqrt{C_1 \times C_2}) \quad (1)$$

Here, $\mu_1 and \mu_2$ are the feature-wise mean values of the real and generated images. T_r is the trace matrix of $C_1 and C_2$ matrices which are covariance matrices.

8.3.2 True and False Positive Rate

To compute the TPR and FPR, we used the counts of True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN)(14). The formulas for calculating TPR and FPR are:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

8.3.3 Structural Similarity Index Measure (SSIM)

The structural similarity index takes into account an image's luminance and contrast while comparing it with the reference image. This is done using the average, standard deviation of pixel values in the image. Usually, SSIM is applied locally over smaller regions of the image or patches rather than globally (over the entire image) all at once.

$$SSIM(x, y) = \frac{(2\mu_y\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

8.3.4 F1 Score

The F1 score metric is a balanced measure that combines both the precision and recall values to obtain a single score to evaluate an algorithm's performance holistically. It is the harmonic mean of Precision and Recall as can be seen in Equation 3.

$$F1\,Score = \frac{2 \times P \times R}{P + R} \tag{3}$$

8.4 Figures

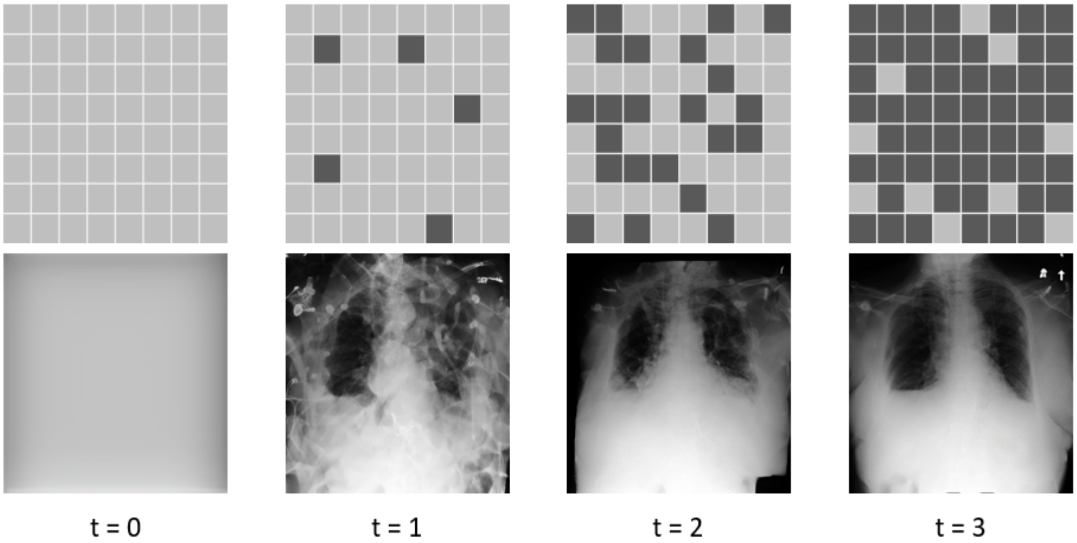

Figure 3: This figure is an illustration of the progressive unmasking the fully masked figure using the prompt: "Pneumothorax" at different time steps from t=0 having the fully masked images and t=3 depicted with a fully unmasked chest xray image. The top row displays a representation of the tokenized image grid. Light grey squared represent masked regions and dark grey squares represent unmasked tokens. We start with a completely masked images and iteratively unmask tokens non-autoregressively at each step to reveal the CXR.

References

[1] P. Chambon, C. Bluethgen, J.-B. Delbrouck, R. Van der Sluijs, M. Połacin, J. M. Z. Chaves, T. M. Abraham, S. Purohit, C. P. Langlotz, and A. Chaudhari, "Roentgen: Vision-language foundation model for chest x-ray generation," 2022, preprint from arXiv. [Online]. Available: https://doi.org/10.48550/arxiv.2211.12737

[2] H. Lee, D. Y. Lee, W. Kim, J.-H. Kim, T. Kim, J. Kim, L. Sunwoo, and E. Choi, "Unixgen: A unified vision-language model for multi-view chest x-ray generation and report generation," *arXiv preprint arXiv:2302.12172*, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2302.12172

[3] J. T. Rolfe, "Discrete variational autoencoders," *arXiv preprint arXiv:1609.02200*, 2016.

[4] A. van den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Gulati, Y. Song, J. Baldridge, and Y. Wu, "Vector-quantized image modeling with improved vqgan," *arXiv preprint arXiv:2110.04627*, 2021.

[6] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, "Muse: Text-to-image generation via masked generative transformers," *arXiv preprint arXiv:2301.00704*, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2301.00704

[7] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 315–11 325.

[8] B. Boecking, N. Usuyama, S. Bannur, D. C. Castro, A. Schwaighofer, S. Hyland, M. Wetscherek, T. Naumann, A. Nori, J. Alvarez-Valle, H. Poon, and O. Oktay, "Making the

most of text semantics to improve biomedical vision–language processing," *arXiv preprint arXiv:2204.09817*, 2022. [Online]. Available: https://doi.org/10.48550/arxiv.2204.09817

[9] A. Johnson, T. Pollard, R. Mark, S. Berkowitz, and S. Horng, "Mimic-cxr database," *PhysioNet*, 2019. [Online]. Available: https://doi.org/10.13026/C2JT1Q

[10] A. E. W. Johnson, T. J. Pollard, N. R. Greenbaum, M. P. Lungren, C. ying Deng, Y. Peng, Z. Lu, R. G. Mark, S. J. Berkowitz, and S. Horng, "Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs," 2019.

[11] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *ArXiv*, vol. abs/1711.05225, 2017. [Online]. Available: https://doi.org/10.48550/arxiv.1711.05225

[12] A. Weng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," https://github.com/arnoweng/CheXNet, 2017.

[13] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," 2019.

[14] Google Developers, "Classification: Roc curve and auc," https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc, Accessed 2023, accessed: [2023-12-08].