# ResNet18-based Variational Autoencoder for Chest-xray Images Reconstruction and Generation

**Chenbin Yu**
A69030936
chy066@ucsd.edu

## Abstract

Variational Autoencoder (VAE) provides a principled probabilistic framework for generative modeling by learning a latent variable representation of data. In this work, we apply a convolutional VAE to the task of medical image generation, focusing on chest X-ray images. The model is trained to reconstruct input images while regularizing the latent space via a KL divergence term. We evaluate the quality of generated images using both qualitative visual inspection and quantitative metrics, including the Inception Score (IS) and Fréchet Inception Distance (FID). Experimental results demonstrate that the proposed model is able to capture meaningful structural patterns of chest X-ray images and generate visually plausible samples.

## 1 Introduction

Generative modeling of medical images has gained increasing attention due to its potential applications in data augmentation, privacy-preserving data sharing, and unsupervised representation learning. Chest X-ray images, in particular, are widely used in clinical practice and present unique challenges due to their high structural complexity and subtle visual patterns.

Variational Autoencoder (VAE) is a class of latent variable model that combine deep neural networks with variational inference. Unlike deterministic autoencoders, VAEs learn a probabilistic mapping from data to a latent space, enabling both reconstruction and novel sample generation. In this work, we employ a convolutional VAE to learn a compact latent representation of chest X-ray images and generate new images by sampling from the learned latent distribution.

Our method adopts a fully convolutional encoder–decoder framework with a Gaussian latent representation. The network is optimized in an end-to-end manner by jointly minimizing the reconstruction objective and the KL divergence regularization term. To assess the generative performance, we employ standard quantitative metrics, including Inception Score (IS) and Fréchet Inception Distance (FID), together with qualitative evaluation through visual inspection of the synthesized samples.

## 2 Methodology

### 2.1 Network Structure Design

**Design Motivation.** Chest X-ray images contain subtle anatomical structures that require both local detail modeling and global contextual understanding. Residual connections improve training stability, progressive downsampling enlarges the receptive field, and interpolation-based upsampling restores spatial details while reducing artifacts. Group normalization ensures stable optimization with small medical batch sizes. The overall architecture of the proposed VAE is illustrated in Figure 1, showing the encoder, reparameterization, and decoder components.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

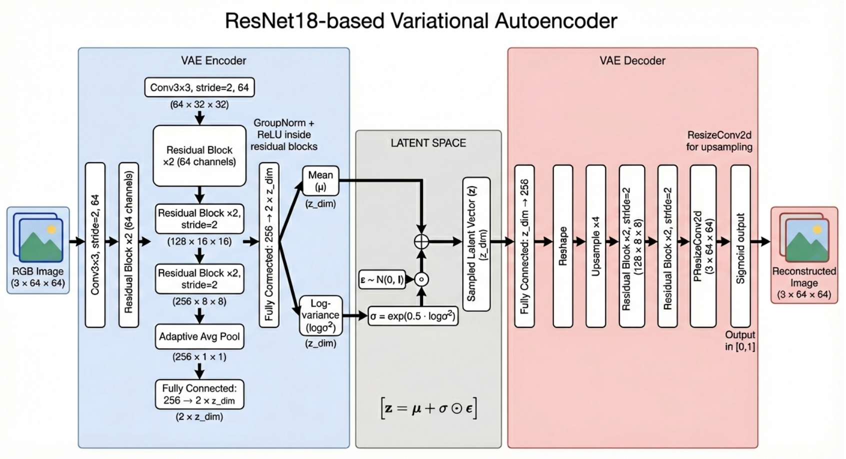
ResNet18-based Variational Autoencoder
VAE Encoder
RGB Image (3 × 64 × 64)
Conv3x3, stride=2, 64
Residual Block x2 (64 channels)
Conv3x3, stride=2, 64
(64 × 32 × 32)
Residual Block x2 (64 channels)
Residual Block ×2, stride=2
(128 × 16 × 16)
Residual Block ×2, stride=2
(256 × 8 × 8)
Adaptive Avg Pool
(256 × 1 × 1)
Fully Connected: 256 → 2 × z_dim
(2 × z_dim)
GroupNorm + ReLU inside residual blocks
Fully Connected: 256 → 2 × z_dim
Mean (μ)
(z_dim)
Log-variance (logσ²)
(z_dim)
LATENT SPACE
ε ~ N(0, I)
σ = exp(0.5 · logσ²)
Sampled Latent Vector (z) (z_dim)
$\left[\mathbf{z} = \mu + \sigma \odot \epsilon\right]$
VAE Decoder
ResizeConv2d for upsampling
Fully Connected: z_dim → 256
Reshape
Upsample ×4
Residual Block ×2, stride=2 (128 × 8 × 8)
Residual Block ×2, stride=2
PResizeConv2d (3 × 64 × 64)
Sigmoid output
Output in [0,1]
Reconstructed Image (3 × 64 × 64)

Figure 1: Overview of the VAE framework used in this work. The encoder maps input chest X-ray images to a latent Gaussian distribution, from which latent variables are sampled via the reparameterization trick and subsequently decoded to reconstruct or generate images.

**VAE Encoder.** The encoder adopts a ResNet18-based architecture to parameterize a diagonal Gaussian posterior

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2(\mathbf{x}))\big).$$

Given an input $\mathbf{x} \in \mathbb{R}^{3\times64\times64}$, we first apply a $3 \times 3$ convolution with stride 2, followed by group normalization (GN) and ReLU:

$$\mathbf{h}_0 = \mathrm{ReLU}\big(\mathrm{GN}(\mathrm{Conv}_{3\times3}^{s=2}(\mathbf{x}))\big), \quad \mathbf{h}_0 \in \mathbb{R}^{64\times32\times32}.$$

Three residual stages are then applied with progressive downsampling:

$$\mathbf{h}_1 \in \mathbb{R}^{64\times32\times32}, \tag{1}$$

$$\mathbf{h}_2 \in \mathbb{R}^{128\times16\times16}, \tag{2}$$

$$\mathbf{h}_3 \in \mathbb{R}^{256\times8\times8}. \tag{3}$$

Each stage contains two residual blocks. Downsampling is achieved by setting the stride of the first block in Stage 2 and Stage 3 to 2. A residual block is defined as

$$\mathrm{Block}(\mathbf{u}; s) = \mathrm{ReLU}\big(\mathcal{F}(\mathbf{u}; s) + \mathrm{SC}(\mathbf{u}; s)\big),$$

where $\mathcal{F}$ denotes two $3 \times 3$ convolutions with GN and ReLU, and SC is either identity ($s = 1$) or a $1 \times 1$ projection with stride 2.

Global average pooling produces $\mathbf{v} \in \mathbb{R}^{256}$, which is linearly projected to

$$[\boldsymbol{\mu}(\mathbf{x}),\ \log \boldsymbol{\sigma}^2(\mathbf{x})] \in \mathbb{R}^{2z_{\mathrm{dim}}}.$$

The latent variable is sampled via

$$\mathbf{z} = \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\sigma}(\mathbf{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

**VAE Decoder.** The decoder mirrors the encoder structure and reconstructs the image from a latent vector $\mathbf{z} \in \mathbb{R}^{z_{\mathrm{dim}}}$. First, the latent code is projected to a high-dimensional feature vector:

$$\mathbf{h}_0 = \mathrm{Linear}(\mathbf{z}) \in \mathbb{R}^{256},$$

which is reshaped to $\mathbf{h}_0 \in \mathbb{R}^{256\times1\times1}$ and upsampled to $\mathbb{R}^{256\times4\times4}$ using interpolation.

The network then applies a sequence of residual decoding stages with progressive upsampling:

$$\mathbf{h}_1 \in \mathbb{R}^{128\times8\times8}, \tag{4}$$

$$\mathbf{h}_2 \in \mathbb{R}^{64\times16\times16}, \tag{5}$$

$$\mathbf{h}_3 \in \mathbb{R}^{32\times32\times32}, \tag{6}$$

$$\mathbf{h}_4 \in \mathbb{R}^{32\times32\times32}. \tag{7}$$

Each stage consists of residual blocks. Upsampling is achieved by replacing strided convolutions with interpolation followed by a 3 × 3 convolution. A decoding residual block is defined as

$$\mathrm{Block}_{\text{dec}}(\mathbf{u}; s) = \mathrm{ReLU}\big(\mathcal{G}(\mathbf{u}; s) + \mathrm{SC}_{\text{dec}}(\mathbf{u}; s)\big),$$

where $\mathcal{G}$ denotes convolutional layers with GN and ReLU, and the shortcut path applies interpolation when $s = 2$.

Finally, a 3 × 3 convolution followed by a sigmoid activation produces the reconstructed image:

$$\hat{\mathbf{x}} = \sigma\big(\mathrm{Conv}_{3\times 3}(\mathbf{h}_4)\big), \quad \hat{\mathbf{x}} \in \mathbb{R}^{3\times 64\times 64}.$$

### 2.2 Loss Function Design

**KL Divergence Loss.** The KL divergence term regularizes the approximate posterior toward a standard normal prior, promoting a structured latent space that enables meaningful sampling and interpolation. Assuming a diagonal Gaussian posterior $q(z|x) = \mathcal{N}(\mu, \mathrm{diag}(\sigma^2))$ and prior $p(z) = \mathcal{N}(0, I)$, the KL divergence is given by

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2}\,\mathbb{E}\left[1 + \log\sigma^2 - \mu^2 - \sigma^2\right]. \tag{8}$$

Here, $\mu$ and $\log\sigma^2$ are the mean and log-variance vectors output by the encoder, respectively. The expectation is taken over the latent dimensions. In practice, this loss is scaled by a weighting factor $\lambda_{\text{KL}}$ to balance latent regularization and reconstruction accuracy.

**Reconstruction Loss.** The reconstruction loss measures the discrepancy between the input image $x$ and its reconstruction $\hat{x}$, encouraging pixel-level fidelity. In our implementation, we consider two forms of reconstruction loss: mean squared error (MSE) and $\ell_1$ loss.

The MSE loss is defined as

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\|\hat{x} - x\|_2^2\right], \tag{9}$$

while the $\ell_1$ loss is given by

$$\mathcal{L}_{\ell_1} = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\|\hat{x} - x\|_1\right]. \tag{10}$$

Here, $x \in \mathbb{R}^{C\times H\times W}$ denotes the input image and $\hat{x}$ is the reconstructed output. The MSE loss emphasizes overall pixel-wise consistency, whereas the $\ell_1$ loss tends to produce sharper reconstructions and is more robust to outliers.

## 3 Experiments

### 3.1 Datasets

The experiments were conducted on the *Chest X-Ray Pneumonia* dataset from KaggleHub (`paultimothymooney/chest-xray-pneumonia`). The dataset contains chest X-ray images collected for pneumonia classification tasks. Since chest X-ray images are grayscale, each image was converted to a 3-channel format to match the input requirement of the ResNet18-based encoder. After preprocessing and filtering, a total of 3883 images were used in our experiments. All images were resized to 64 × 64 resolution before being fed into the model. Using PyTorch DataLoader, the dataset was randomly split into training and testing subsets with an 8:2 ratio. Specifically, 3106 images were used for training and 777 images were used for testing.

### 3.2 Hyperparameter Settings

**Learning Rate.** The learning rate controls the step size of parameter updates during optimization. A higher learning rate accelerates convergence but may cause instability, while a lower value leads to slower yet more stable training. In our experiments, we carefully tuned the learning rate to balance convergence speed and stability.

**Latent Space Dimension.** The latent dimension determines the capacity of the bottleneck representation. A larger latent space allows the model to encode more detailed information but increases the risk of overfitting or posterior collapse. A smaller latent dimension enforces stronger compression and may limit reconstruction fidelity.

**Batch Size.** Batch size affects gradient estimation stability and training efficiency. Larger batches provide smoother gradient updates but require more memory, while smaller batches introduce more stochasticity and may improve generalization.

**Number of Epochs.** The number of epochs specifies how many full passes over the training dataset are performed. More epochs allow the model to converge more fully, though excessive training may lead to diminishing returns.

**KL Loss Weight.** The KL weight balances the regularization term in the VAE objective. Increasing this value strengthens latent distribution regularization and improves generative consistency, while decreasing it emphasizes reconstruction quality.

**Reconstruction Loss Weight.** The reconstruction weight scales the reconstruction loss term. A larger value prioritizes pixel-level fidelity, while a smaller value shifts the focus toward latent regularization and generative diversity.

### 3.3 Training

**Training Procedure.** The VAE model is trained in an end-to-end manner using the Adam optimizer. For each input image, the encoder first produces the mean and log-variance of the latent distribution. A latent vector is then sampled using the reparameterization trick and passed to the decoder to reconstruct the image.

During each training iteration, the reconstruction loss and KL divergence are computed. These two terms are weighted by predefined coefficients and summed to form the total loss. Backpropagation is applied to update the model parameters. For every epoch, the average reconstruction loss, KL loss, and total loss are recorded. The model checkpoint with the lowest total loss is saved as the best model. Periodic checkpoints are also stored during training.

**Training Metrics.** Two main metrics are monitored during training:

1. **Reconstruction Loss** This measures the difference between the input image and its reconstruction. In our implementation, L1 loss is used by default (MSE can also be applied). It reflects how well the model preserves image details.
2. **KL Divergence Loss** This regularization term encourages the learned latent distribution to be close to a standard normal distribution. It ensures that the latent space remains continuous and suitable for image generation. The final objective balances reconstruction quality and latent space regularization.

### 3.4 Inference

**Reconstruction.** During evaluation, the trained model is loaded from the best checkpoint and set to evaluation mode. For each input image, the encoder outputs the latent mean and log-variance. Instead of sampling, the latent mean is directly used as the deterministic representation. This latent vector is then fed into the decoder to obtain the reconstructed image. The reconstructed samples are saved for visualization and further evaluation.

**Generation.** For image generation, latent vectors are randomly sampled from a standard normal distribution. These sampled latent codes are passed through the decoder to synthesize new images. The generated samples are then used to compute quantitative metrics and are also saved for visualization.

**Evaluation Metrics:** Two quantitative metrics are used to evaluate generative performance:

1. **FID** This measures the distance between real and generated image feature distributions. Let $(\mu_r, \Sigma_r)$ and $(\mu_g, \Sigma_g)$ denote the mean and covariance of real and generated features extracted from a pretrained Inception network. FID is defined as:

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right) \tag{11}$$

   Lower FID indicates better alignment between real and generated distributions.

2. **Inception Score** This evaluates both image quality and diversity. Given the conditional label distribution $p(y|x)$ predicted by the Inception model, IS is defined as:

$$\text{IS} = \exp\left(\mathbb{E}_x\left[\text{KL}\left(p(y|x)\|p(y)\right)\right]\right) \tag{12}$$

   Higher IS indicates that generated images are both sharp (low entropy $p(y|x)$) and diverse (high entropy marginal $p(y)$).

# 4 Results and Evaluation

During preliminary experiments, it was observed that the scales of L1 loss and KL loss were significantly different. In early training, the KL term showed signs of posterior collapse (although it gradually recovered after around 20 epochs). To reduce its destabilizing effect, a very small KL weight was used.

Additionally, the original magnitude of L1 loss was relatively small. Since increasing the learning rate led to unstable training and divergence, a reconstruction weight was introduced to amplify its contribution. This helped accelerate the decrease of L1 loss and made parameter updates more effective.

## 4.1 Trial 1

Batch size was set to 16, number of epochs to 100, learning rate to 0.0005, latent dimension to 64, KL weight to 0.002, and reconstruction weight to 5.

**Quantitative Evaluation.**

FID = 0.6297     IS = 1.3233

Although the FID value is relatively low, it should be interpreted cautiously. The dataset distribution is relatively simple and reconstruction-dominant training can reduce distribution distance without improving true generative diversity. The Inception Score remains low, indicating limited sample diversity and weak semantic confidence. This further confirms that the small KL weight restricts latent space expressiveness.

**Loss Analysis.** The KL term shows increase due to the very small weighting, but its overall impact on total loss is minimal. The L1 loss steadily decreases throughout training. Considering computational cost and diminishing returns, the number of epochs was not further increased.

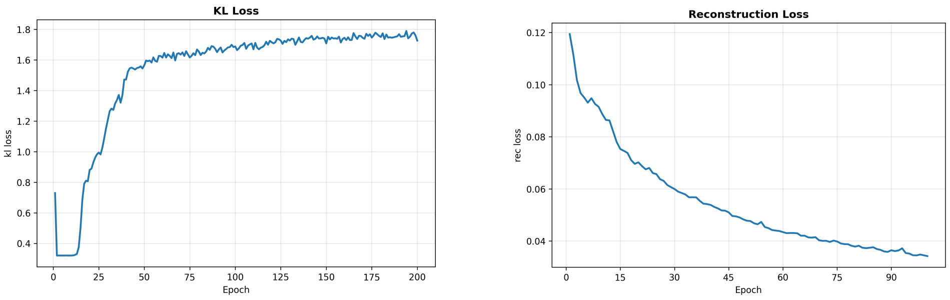

Figure 2: Loss curves for Trial 1: (1) KL Loss, (2) L1 Loss

**Reconstruction Results.** The reconstructed images preserve the main structural contours of the original inputs. However, most fine-grained features and detailed textures are missing. The model mainly captures coarse global structure rather than discriminative visual patterns.

**Generation Results.** The generation quality is relatively poor. Due to the extremely small KL weight, the latent space is weakly regularized, resulting in limited diversity and poor generative capability. The generated samples lack semantic consistency and visual clarity.

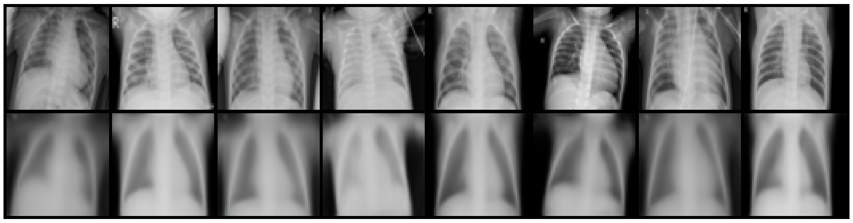

Figure 3: Reconstruction results for Trial 1.

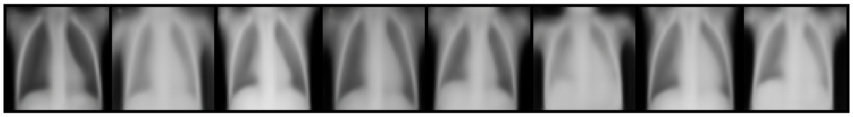

Figure 4: Generation results for Trial 1.

### 4.2 Trial 2

Batch size was set to 32, number of epochs to 200, learning rate to 0.001, latent dimension to 64, KL weight to 0.0001, and reconstruction weight to 5.

**Quantitative Evaluation.**

$$\text{FID} = 0.2746 \qquad \text{IS} = 1.8402$$

The FID score improves significantly compared to Trial 1, indicating that the generated distribution is much closer to the real data distribution in feature space. This improvement is mainly driven by the strong reconstruction dominance, which forces outputs to remain highly aligned with training samples.

The Inception Score also increases, suggesting clearer visual structures and improved confidence of predicted semantics. However, this does not necessarily imply true diversity improvement, as the extremely small KL weight still weakens latent regularization. The model behaves more like a deterministic autoencoder, prioritizing reconstruction fidelity over generative variability.

**Loss Analysis.** Since the KL weight is extremely small, its numerical value increases during training but contributes very little to the total loss. The L1 loss decreases steadily and converges clearly, showing strong reconstruction dominance. Increasing the number of epochs to 200 further stabilizes convergence.

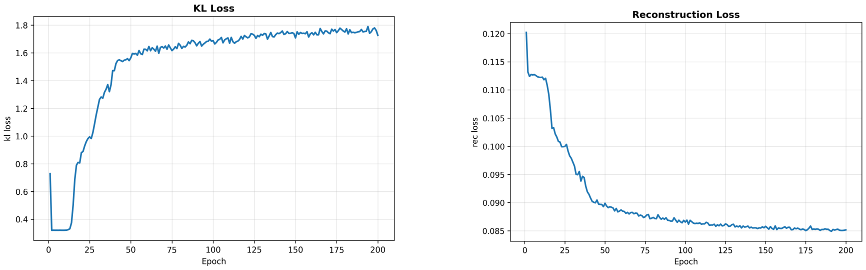

Figure 5: Loss curves for Trial 2: (1) KL Loss, (2) L1 Loss

**Reconstruction Results.** The reconstruction quality is significantly improved. Major structures and detailed features are clearly preserved. Although resizing the input images to $64 \times 64$ inevitably removes some fine-grained information, the overall reconstruction performance is satisfactory and visually sharp.

**Generation Results.** The generation results remain nearly identical across samples. Due to the negligible KL constraint, the latent space lacks meaningful regularization, resulting in very limited variability. The generated images show low diversity and weak semantic variation.

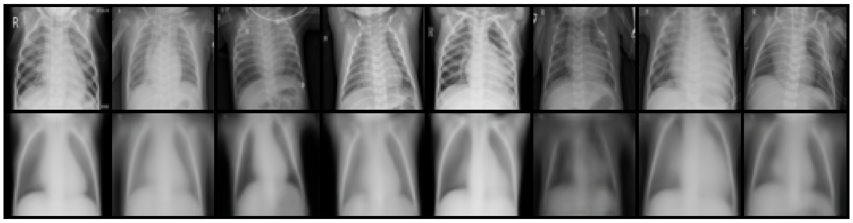

Figure 6: Reconstruction results for Trial 2.

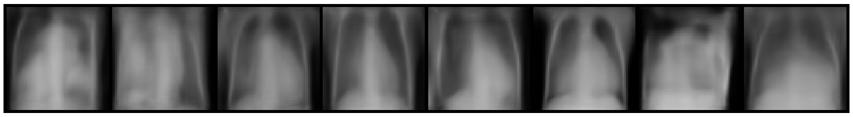

Figure 7: Generation results for Trial 2.

### 4.3 Trial 3

Batch size was set to 32, number of epochs to 100, learning rate to 0.001, latent dimension to 128, KL weight to 2, and reconstruction weight to 1.

**Quantitative Evaluation.**

$$\text{FID} = 0.4209 \qquad \text{IS} = 1.9845$$

Both FID and Inception Score improve, indicating a better trade-off between distribution matching and sample quality/diversity. Increasing the latent dimension allows the posterior to encode richer information, while a larger KL weight strengthens latent regularization and makes random sampling more meaningful. Meanwhile, reducing the reconstruction weight prevents the model from behaving like a purely deterministic autoencoder, which helps improve generative behavior.

**Loss Analysis.** Both reconstruction loss and KL loss converge well, suggesting stable optimization under the adjusted learning rate and rebalanced loss weights. Compared with previous trials, the KL term plays a more dominant role and the training objective becomes less reconstruction-dominant, leading to better latent space structure for generation.

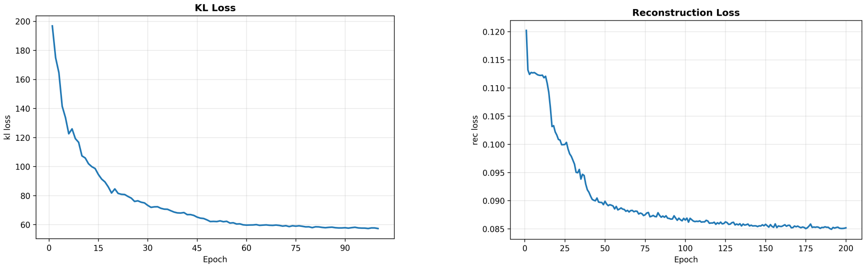

Figure 8: Loss curves for Trial 3: (1) KL Loss, (2) L1 Loss

**Reconstruction Results.** Reconstructed images show clear global contours and preserve the main anatomy structure. However, due to the limited input resolution, fine-grained details are still missing. For example, subtle rib patterns in chest X-rays are not fully reconstructed, indicating that higher-frequency features are difficult to recover under the current image size constraint.

**Generation Results.** Generation quality improves compared to earlier trials, but remains only moderate. The increased KL weight and larger latent space enhance variability and produce more coherent samples, yet the restricted image size limits the realism and fine-detail synthesis. The generated images tend to capture coarse anatomical structure but lack high-resolution texture.

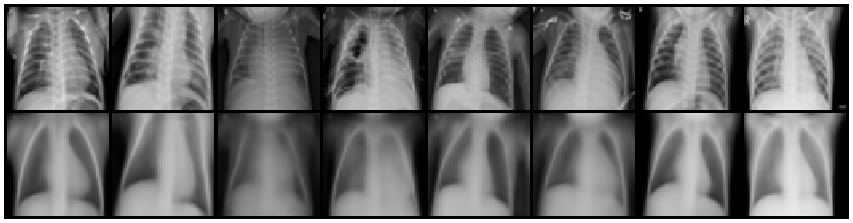

Figure 9: Reconstruction results for Trial 3.

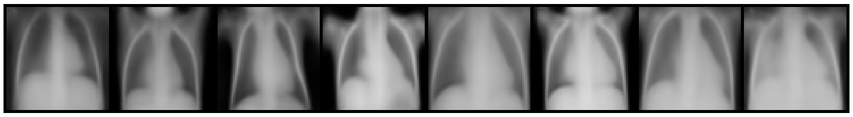

Figure 10: Generation results for Trial 3.

### 4.4 Loss Comparison

We compared MSE and L1 losses for the chest X-ray VAE task. Models trained with MSE produce smoother reconstructions but tend to blur fine details such as rib structures and subtle textures. This is due to the averaging effect of squared error.

In contrast, L1 loss generates sharper images with clearer anatomical boundaries. Although both losses converge stably, L1 preserves more meaningful structural details. Therefore, L1 loss was adopted as the main reconstruction objective in our experiments.

## 5 Conclusion

In this work, we implemented a ResNet18-based Variational Autoencoder and explored the trade-off between reconstruction quality and generative performance under different hyperparameter settings. We found that small KL weights favor reconstruction but weaken generation, while stronger KL regularization improves latent structure and diversity at the cost of fine reconstruction details. The limited input resolution also restricts high-frequency feature recovery.

To improve generation quality, future work may increase image resolution, apply KL annealing to stabilize training, and adopt more expressive decoders or perceptual losses. These strategies could enhance both latent representation quality and visual realism.