Conditional Handwritten Digit Generation

1 Introduction

Handwritten digit recognition, a classic challenge in machine learning and computer vision [1, 2], continues to offer avenues for exploration, particularly in generative modeling. This report investigates and compares two advanced autoencoder architectures - Conditional Autoencoders (CAEs) [3] and Conditional Variational Autoencoders (CVAEs) [4] - in their application to simultaneously recognize and generate handwritten digits using the MNIST dataset, a standard benchmark in the field. This project focuses on the models' ability to operate under various conditions: reconstructing images and validating labels when both are provided, predicting labels from images alone, and generating plausible images from labels. This multi-faceted approach allows us to explore the strengths and limitations of each model in tasks ranging from image reconstruction to conditional generation.

The remainder of this report is structured as follows: First, we define the problem formulation, detailing our objectives and the specific tasks we aim to address. Then, we describe our model designs, loss functions, and training procedures for both CAE and CVAE architectures. We also explore the impact of different latent space dimensions on model performance. Our methods section concludes with an overview of the evaluation metrics used to assess and compare the models. Finally, we present and discuss our results, offering insights into the relative performance of CAEs and CVAEs in handwritten digit recognition and generation tasks.

2 Problem Formulation

This project focuses on the conditional generation and recognition of handwritten digits using two advanced autoencoder architectures: the Conditional Autoencoder (CAE) and the Conditional Variational Autoencoder (CVAE). The primary objective is to develop and compare models that can interpret, generate, and classify handwritten digits with conditional inputs. While the core tasks of digit recognition and conditional generation are fundamentally supervised learning problems, our approach incorporates unsupervised learning elements through the autoencoder architecture. This hybrid methodology allows us to leverage both labeled data for prediction tasks and the generative capabilities of autoencoders for image reconstruction and generation.

The dataset for this project is the MNIST collection [5], originally created by Yann LeCun, Corinna Cortes, and Christopher J.C. Burges for handwritten digit recognition tasks. It consists of 60,000 training images and 10,000 test images of handwritten digits, derived from a larger set of documents available from the U.S. National Institute of Standards and Technology (NIST). The digits were size-normalized and centered in fixed-size (28x28 pixel) grayscale images. Each pixel has a value between 0 and 1, and each image is paired with its corresponding label (a categorical number from 0 to 9). For our models, these labels are converted into one-hot encoded vectors. Figure A.1.1 shows some example images from the dataset. For feature selection, we use all 784 raw pixel values from the 28x28 MNIST images. This is standard practice for MNIST due to its small input size and the proven effectiveness of using all features in deep learning models [1].

The challenge lies in creating a model with the flexibility to operate under various conditions. When both an image and its one-hot encoded label are available, the model's task is to reconstruct the original image and validate the label. If only an image is provided, the model should be capable of reconstructing the image and predicting the most likely label. Additionally, when given only a label, the model must generate a plausible handwritten image that corresponds to that digit. This problem involves integrating several key aspects of machine learning: (1) generative modeling to create new and realistic digit images, (2) reconstruction to accurately reproduce input images, and (3) discriminative modeling to distinguish between digit classes and predict labels. The challenge is to design a single model that effectively balances these tasks, learning to represent handwritten digits in a way that captures both their visual features and their semantic meanings. The project compares the performance of deterministic CAE and probabilistic CVAE architectures, exploring how these models balance the tasks of generation, reconstruction, and classification, and how different latent space dimensions (8, 16, 32, and 64) impact model performance.

3 Methods

Autoencoders [6] are a class of neural networks designed for unsupervised learning of efficient data encodings. The fundamental architecture of an autoencoder comprises two main components: an encoder, which maps the input data to a compressed latent representation, and a decoder, which attempts to reconstruct the original input from this latent representation. In this work, we explore more advanced variants of autoencoders that incorporate additional input conditions, enabling the model to learn relationships between the input data and specific attributes or labels.

All models used in this work follow an encoder-decoder architecture and are designed to process both image and digit inputs simultaneously. The encoder takes two inputs: a 784-dimensional image vector (representing a 28x28 pixel image) and a 10-dimensional one-hot encoded digit vector. Both inputs are processed through separate pathways, each comprising two fully connected layers with ReLU activations, outputting 512-dimensional vectors.

The outputs from these image and digit pathways are then combined through element-wise addition. This fused representation is projected into a latent space, the nature of which differs between the models. The decoder takes the latent representation and reconstructs both the image and the digit. It starts with two shared fully connected layers with ReLU activations, each with 512 neurons. For image reconstruction, a final fully connected layer projects the shared representation back to the image space with a sigmoid activation. For digit reconstruction, another fully connected layer projects to the digit space with a log-softmax activation.

Conditional Autoencoder (CAE)

The CAE projects the fused representation directly into a low-dimensional latent space using a fully connected layer. The CAE loss function is defined as:

$$\mathcal{L}_{\text{CAE}} = -\sum_{i=1}^{784} [x_i \log(\hat{x}_i) + (1 - x_i) \log(1 - \hat{x}_i)] - \sum_{j=1}^{10} y_j \log(\hat{y}_j)$$

where x and $\hat{x}$ are the input and reconstructed images respectively, and y and $\hat{y}$ are the true and predicted digit labels. This loss function encourages the CAE to accurately reconstruct the input images while correctly predicting the associated digit labels, without imposing any explicit constraints on the structure of the latent space.

Conditional Variational Autoencoder (CVAE)

The CVAE extends the CAE by introducing a probabilistic approach to the latent space. The fused representation is projected into the latent space using two parallel fully connected layers: one producing the mean (μ) and the other the log-variance ($\log \sigma^2$) of the latent distribution. The latent vector z is then sampled using the reparameterization trick: $z = \mu + \sigma\epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. See Figure A.2.2 for more details on the CVAE architecture.

The loss function used for training the CVAE balances three key objectives: accurate image reconstruction, regularization of the latent space, and correct digit classification. This composite loss function consists of three components: (1) the reconstruction loss, which measures how well the decoder can reconstruct the input image from the latent representation, (2) the Kullback-Leibler divergence (KLD), which acts as a regularizer by encouraging the learned latent distribution to approximate a standard normal distribution, and (3) the classification loss, which quantifies the accuracy of the digit predictions. The reconstruction loss is formulated as the binary cross-entropy (BCE) between the input image and its reconstruction, which is equivalent to the negative log-likelihood of the input image given its latent representation for binary data. The KLD term ensures a well-structured and continuous latent space. The classification loss is implemented as the negative log-likelihood (NLL) of the correct digit label. These components are combined in the following loss function:

$$\mathcal{L}_{\text{CVAE}} = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x)||p(z)) - \sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where the first term represents the reconstruction loss (implemented as BCE, with x and $\hat{x}$ being the input and reconstructed images respectively), the second term is the KL divergence between the approximate posterior $q_\phi(z|x)$ and the prior $p(z)$, and the third term is the classification loss (with y and $\hat{y}$ being the true and predicted digit labels). By minimizing this combined loss, the CVAE learns to generate realistic digit images while also capturing the underlying digit classes in a structured latent space.

Training Procedure

The training procedure for our model utilizes a modified split of the MNIST dataset. From the original 60,000 training samples, we allocate 48,000 (80%) for actual training and reserve 12,000 (20%) for validation. The original 10,000 test samples are kept separate for final model evaluation. We chose this 80-20 configuration to balance the needs of model training and validation. The training set of 48,000 samples provides a substantial amount of data for effective model training, allowing the models to learn from a diverse range of handwritten digits. The validation set of 12,000 samples offers reliable estimates of model performance during training, helping to prevent overfitting and guide hyperparameter tuning. This split is particularly beneficial for our models, as it ensures a robust evaluation of both reconstruction quality and generative capabilities across a wide range of digits. Keeping the original test set of 10,000 samples intact allows for fair comparison with other models in the literature and provides a final, unbiased evaluation of model performance. All available data is used without performing any feature selection, allowing the model to learn from the full range of information present in the dataset. For images, we use raw pixel values (0-1) of the 28x28 images as features, preserving crucial spatial information. Digit labels are one-hot encoded to ensure equal weight for each class.

All models undergo training for a total of 50 epochs utilizing the Adam optimizer [7] with a learning rate of 10^{-3}. A key aspect of the training procedure is the dynamic input selection implemented for each batch. Specifically, for every batch during training, it is randomly decided what type of input to pass to the model. There are three possible scenarios: (1) both the image and the corresponding digit are provided as input, (2) only the image is provided as input, or (3) only the digit is provided as input. This randomized input strategy serves a crucial purpose: it trains the model to reconstruct one type of information from the other. By alternating between these input types, we encourage the model to learn robust representations that can bridge the gap between visual and numerical representations of digits.

Evaluation Metrics

To assess and compare the performance of our CAE and CVAE models, we employ several evaluation metrics. These include Generated Images Accuracy, Fréchet Inception Distance (FID) [8], Inception Score [9], Classification Accuracy and Reconstruction Error. Each metric provides unique insights into the models' performance in terms of image generation quality, distribution matching, and classification capability. Detailed explanations and mathematical formulations of these metrics are provided in Appendix A.3.

4 Results

Our evaluation compares the performance of CAEs and CVAEs across four latent space dimensions: 8, 16, 32, and 64. Table 1 summarizes the quantitative results for each model configuration. The highest Generated Images Accuracy was achieved by CAE (32) at 0.94 ± 0.05. The lowest FID score was obtained by CVAE (64) at 414.87 ± 2.56. CAE (8) showed the highest Inception Score of 2.25 ± 0.14. The best Classification Accuracy was shared by CAE (16), CAE (32), and CAE (64), all at 0.98 with slight variations in standard deviation. The lowest Reconstruction Error was seen in CAE (64) at 61.06 ± 1.64. Across all metrics, CAE models generally outperformed CVAE

	Generated Images Accuracy	FID	Inception Score	Classification Accuracy	Reconstruction Error
CAE (8)	0.90 ± 0.06	424.12 ± 3.19	**2.25 ± 0.14**	0.96 ± 0.0009	83.40 ± 0.85
CVAE (8)	0.84 ± 0.02	419.51 ± 3.83	2.15 ± 0.14	0.94 ± 0.0013	92.99 ± 1.17
CAE (16)	0.90 ± 0.09	426.09 ± 3.56	2.17 ± 0.11	0.98 ± 0.0009	68.00 ± 0.75
CVAE (16)	0.82 ± 0.03	417.84 ± 6.23	2.16 ± 0.17	0.94 ± 0.0028	89.52 ± 1.63
CAE (32)	**0.94 ± 0.05**	427.64 ± 2.71	2.15 ± 0.13	**0.98 ± 0.0006**	62.30 ± 1.20
CVAE (32)	0.84 ± 0.02	417.79 ± 1.57	2.13 ± 0.13	0.94 ± 0.0027	88.26 ± 0.38
CAE (64)	0.88 ± 0.10	425.50 ± 2.02	2.21 ± 0.08	0.98 ± 0.0011	**61.06 ± 1.64**
CVAE (64)	0.84 ± 0.02	**414.87 ± 2.56**	2.24 ± 0.10	0.94 ± 0.0022	89.13 ± 0.83

Table 1: Results comparing CAE and CVAE. For the *Generated Images Accuracy* 1000 images have been generated and tested if a classifier can correctly classify them. *FID* is the Frechét Inception Distance between the generated images and the test set and the *Classification Accuracy* tells how good the joint model can classify the images if only the image is given as an input on the test set. The *Reconstruction Error* is a measure on how well the model is able to reconstruct the images of the test set. The table shows the results for models with different latent dimensions.

models of the same latent dimension, except for FID scores where CVAE models consistently achieved lower values.

Qualitative analysis of the models' performance reveals several key trends. Examination of the loss curves (Figures A.4.3 to A.4.6) shows that CAE consistently achieves lower training and validation losses compared to CVAE across all latent dimensions. A significant decrease in loss is observed when increasing from 8 to 16 latent dimensions, with more modest improvements at higher dimensions. Reconstruction quality, as illustrated in Figure A.5.7, improves with increasing latent dimensions for both models. This improvement is most noticeable when moving from 8 to 16 and then to 32 dimensions, with less pronounced enhancements between 32 and 64 dimensions. Generated image quality, shown in Figures A.6.8 to A.6.11, also varies with latent dimension. At lower dimensions (8 and 16), CVAE tends to produce more clearly defined digits compared to CAE. As latent dimensions increase to 32 and 64, both models demonstrate improved generation capabilities, producing mostly recognizable digits with occasional instances of less defined or ambiguous outputs.

5 Discussion and Conclusion

This work compares Conditional Autoencoders (CAEs) and Conditional Variational Autoencoders (CVAEs) for handwritten digit recognition and generation using the MNIST dataset. CAE models generally outperform CVAE models in most quantitative metrics, including Generated Images Accuracy, Inception Score, Classification Accuracy, and Reconstruction Error. However, CVAEs consistently achieve lower FID scores, indicating their potential superiority in generating images that match the overall distribution of the training data. The impact of latent space dimensionality is evident across both model types, with substantial improvements when increasing from 8 to 16 dimensions, and more modest gains at higher dimensions. Qualitative analysis reveales that while CAEs show better quantitative performance, CVAEs demonstrate advantages in image generation at lower latent dimensions. This discrepancy highlights the importance of considering both quantitative metrics and qualitative assessments when evaluating generative models. The choice between CAE and CVAE may depend on the specific requirements of the task at hand, with CAEs potentially preferable for applications prioritizing reconstruction accuracy and classification, while CVAEs could be more suitable for tasks requiring diverse image generation.

In conclusion, while CAEs demonstrate superior performance in most quantitative metrics, the strengths of CVAEs in certain areas suggest that both models have valuable roles to play in conditional image generation and recognition. Future work could explore more complex datasets, investigate hybrid architectures, or delve deeper into the factors influencing the performance differences between these models.

References

[1] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[2] ABM Ashikur Rahman et al. "Two decades of bengali handwritten digit recognition: A survey". In: *IEEE Access* 10 (2022), pp. 92597–92632.

[3] Savvas Karatsiolis and Christos N Schizas. "Conditional generative denoising autoencoder". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.10 (2019), pp. 4117–4129.

[4] Ashish Mishra et al. "A generative model for zero shot learning using conditional variational autoencoders". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 2188–2196.

[5] Li Deng. "The mnist database of handwritten digit images for machine learning research [best of the web]". In: *IEEE signal processing magazine* 29.6 (2012), pp. 141–142.

[6] Pengzhi Li, Yan Pei, and Jianqiang Li. "A comprehensive survey on design and application of autoencoder in deep learning". In: *Applied Soft Computing* 138 (2023), p. 110176.

[7] Diederik P Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[8] Martin Heusel et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (2017).

[9] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016).

A.1 MNIST Examples

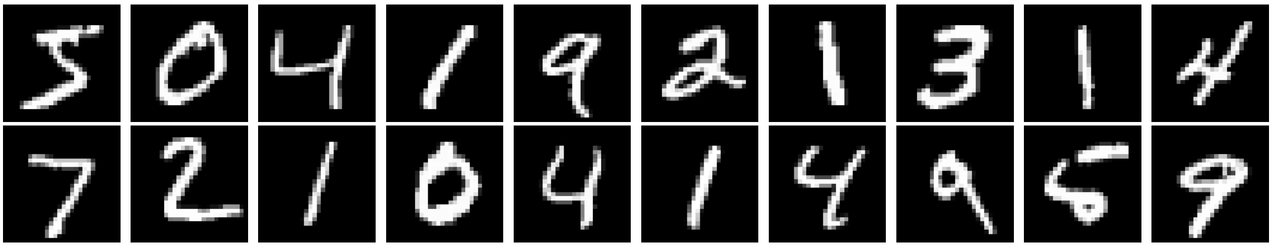


Figure A.1.1: A set of randomly sampled images from the MNIST dataset.

A.2 CVAE Architecture

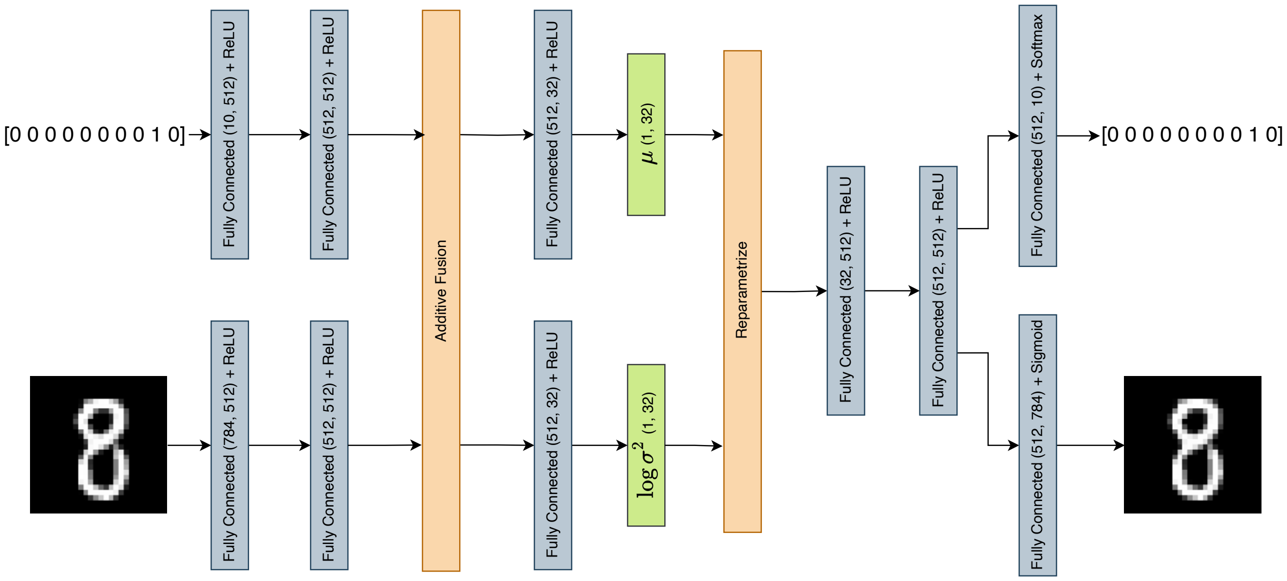

[0 0 0 0 0 0 0 0 1 0]
Fully Connected (10, 512) + ReLU
Fully Connected (512, 512) + ReLU
Fully Connected (784, 512) + ReLU
Fully Connected (512, 512) + ReLU
Additive Fusion
Fully Connected (512, 32) + ReLU
Fully Connected (512, 32) + ReLU
μ (1, 32)
$\log \sigma^2$ (1, 32)
Reparametrize
Fully Connected (32, 512) + ReLU
Fully Connected (512, 512) + ReLU
Fully Connected (512, 10) + Softmax
Fully Connected (512, 784) + Sigmoid
[0 0 0 0 0 0 0 0 1 0]

Figure A.2.2: CVAE architecture for simultaneous digit recognition and generation. The encoder processes image (784D) and digit (10D) inputs through separate pathways, combines them, and projects to a latent space. The decoder reconstructs both image and digit from the sampled latent vector.

A.3 Evaluation Metrics

A.3.1 Generated Images Accuracy

This metric evaluates the quality of generated images by measuring a separate classifier's ability to correctly identify the digits in these images. It is calculated as:

$$\text{Accuracy} = \frac{\text{Number of correctly classified generated images}}{\text{Total number of generated images}} \tag{1}$$

A.3.2 Fréchet Inception Distance (FID)

FID measures the similarity between the distribution of generated images and that of real MNIST images. Lower FID scores indicate better quality and diversity in the generated samples. The FID is computed as:

$$\text{FID} = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \tag{2}$$

where μ_r and μ_g are the mean feature representations of real and generated images respectively, and Σ_r and Σ_g are their covariance matrices.

A.3.3 Inception Score

This score assesses both the quality and diversity of generated images. Higher inception scores suggest better model performance. The Inception Score is defined as:

$$\text{IS} = \exp(\mathbb{E}_x[\text{KL}(p(y|x)||p(y))]) \tag{3}$$

where x is a generated image, $p(y|x)$ is the conditional class distribution, and $p(y)$ is the marginal class distribution.

A.3.4 Classification Accuracy

This metric evaluates the models' ability to correctly classify MNIST digits when given only the image as input. It is calculated as:

$$\text{Classification Accuracy} = \frac{\text{Number of correctly classified test images}}{\text{Total number of test images}} \tag{4}$$

A.3.5 Reconstruction Error

This metric measures how accurately the model can reconstruct the original input images. For binary images like MNIST, we use Binary Cross-Entropy (BCE) as the reconstruction error:

$$\text{BCE} = -\frac{1}{n}\sum_{i=1}^{n}[x_i \log(\hat{x}_i) + (1 - x_i)\log(1 - \hat{x}_i)] \tag{5}$$

where n is the total number of pixels, x_i is the true value of the i-th pixel, and $\hat{x}_i$ is the reconstructed value. Lower values indicate better reconstruction quality.
Each of these metrics provides a different perspective on model performance:

- **Generated Images Accuracy** directly assesses the quality and recognizability of the generated images.
- **FID** provides a measure of how close the distribution of generated images is to the real data distribution, capturing both quality and diversity.
- **Inception Score** evaluates both the quality of individual generated images and the diversity of the generated set as a whole.
- **Classification Accuracy** measures how well the model has learned to extract relevant features for digit recognition.
- **Reconstruction Error** evaluates how well the model preserves information through the encoding and decoding process.

Together, these metrics allow for a comprehensive evaluation of the CAE and CVAE architectures across various latent space dimensions, providing insights into their strengths and weaknesses in different aspects of image generation and classification tasks.

A.4 Training and Validation Losses

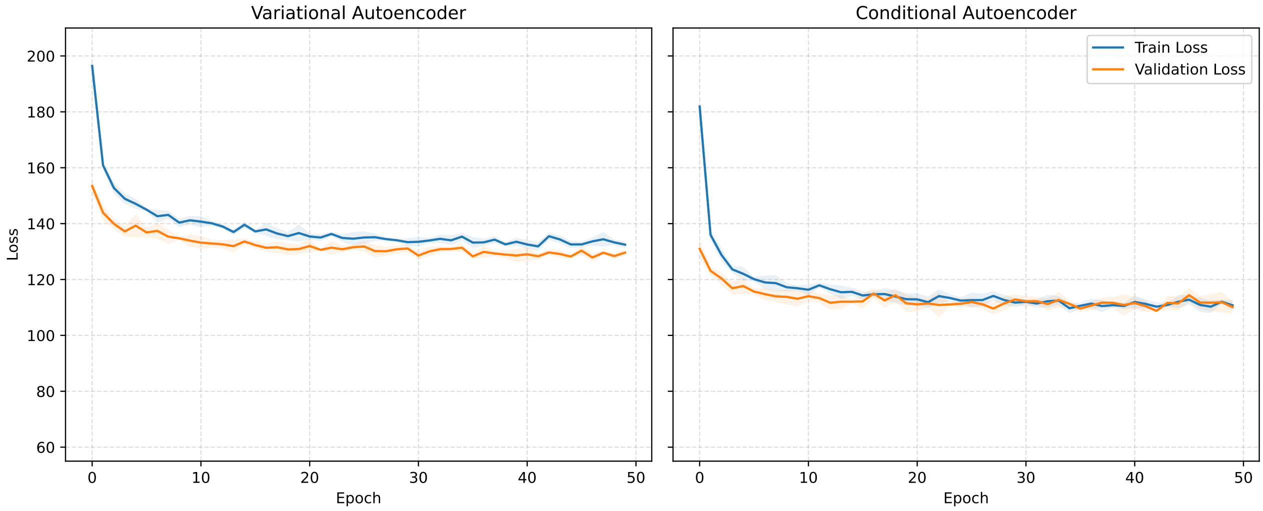


Figure A.4.3: Training and Validation Losses for CVAE (left) and CAE (right) with a latent dimension of 8.

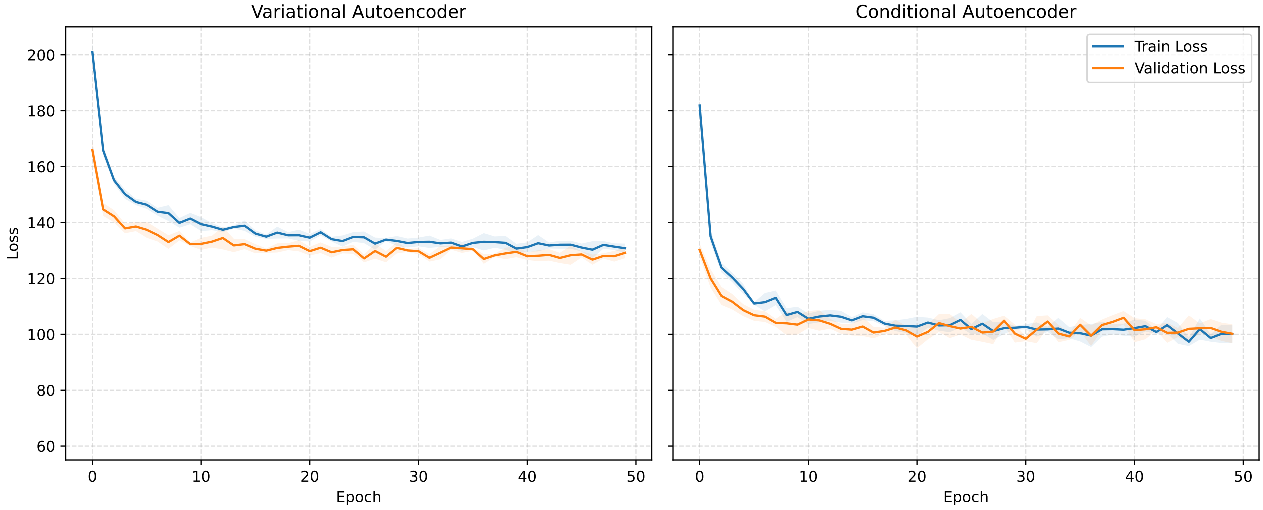


Figure A.4.4: Training and Validation Losses for CVAE (left) and CAE (right) with a latent dimension of 16.

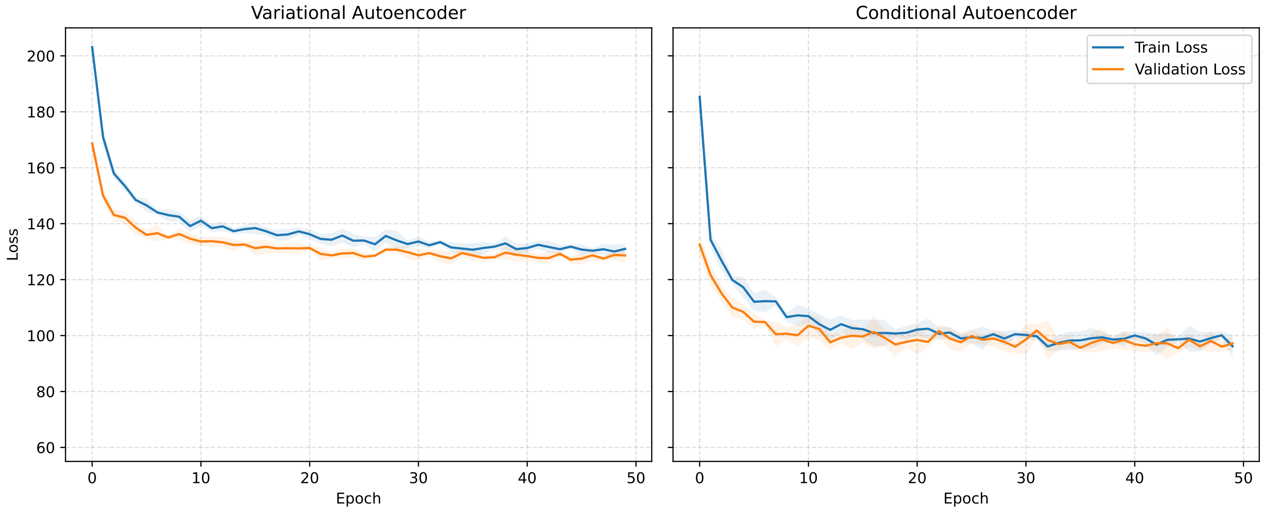


Figure A.4.5: Training and Validation Losses for CVAE (left) and CAE (right) with a latent dimension of 32.

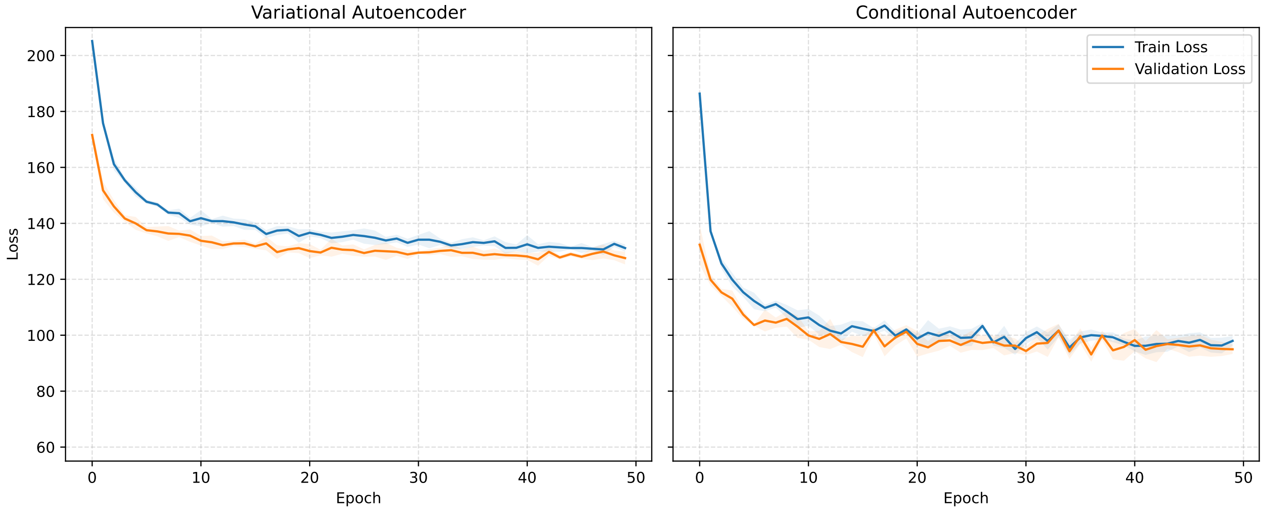
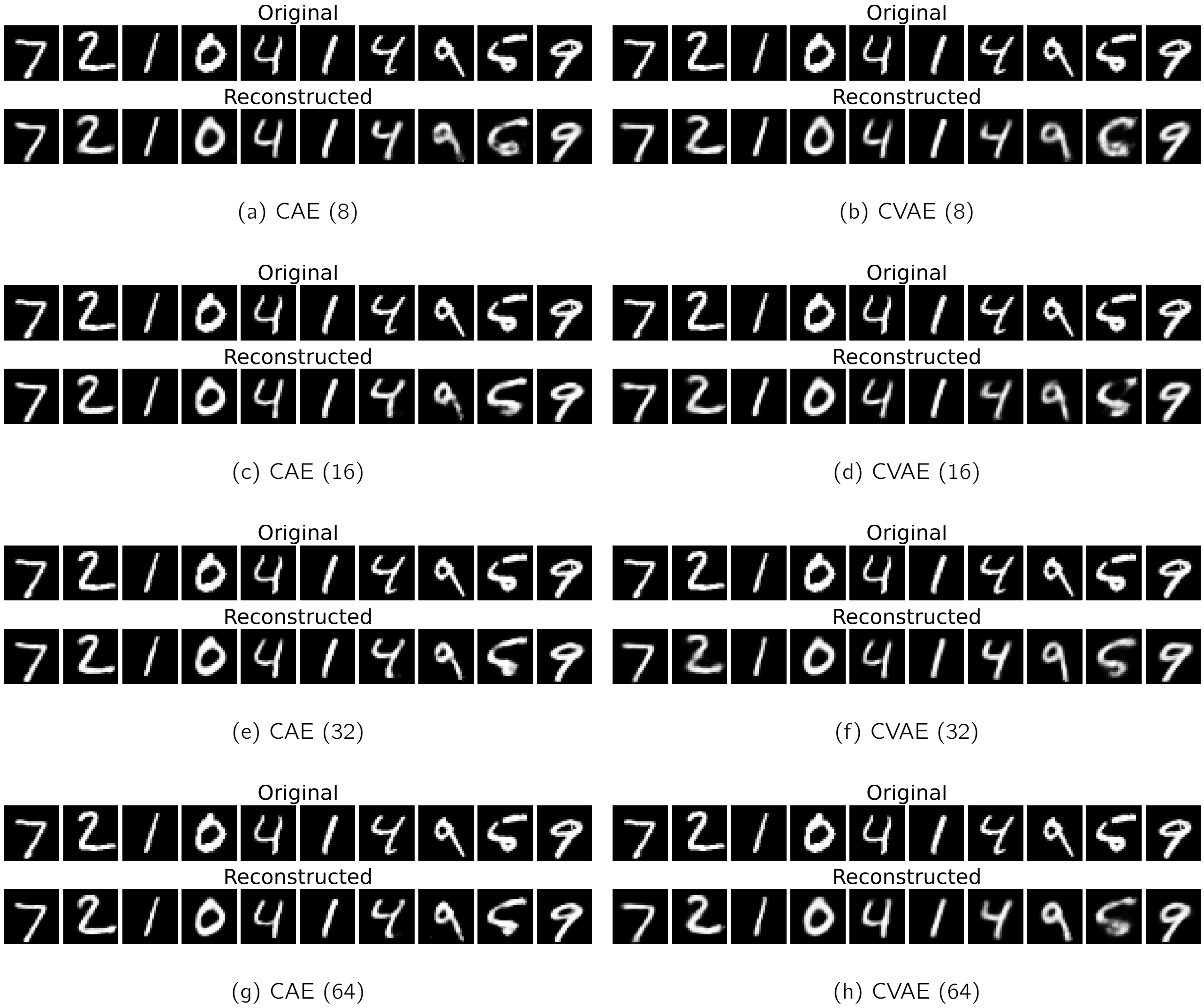


Figure A.4.6: Training and Validation Losses for CVAE (left) and CAE (right) with a latent dimension of 64.

A.5 Reconstruction Examples



(a) CAE (8)

(b) CVAE (8)

(c) CAE (16)

(d) CVAE (16)

(e) CAE (32)

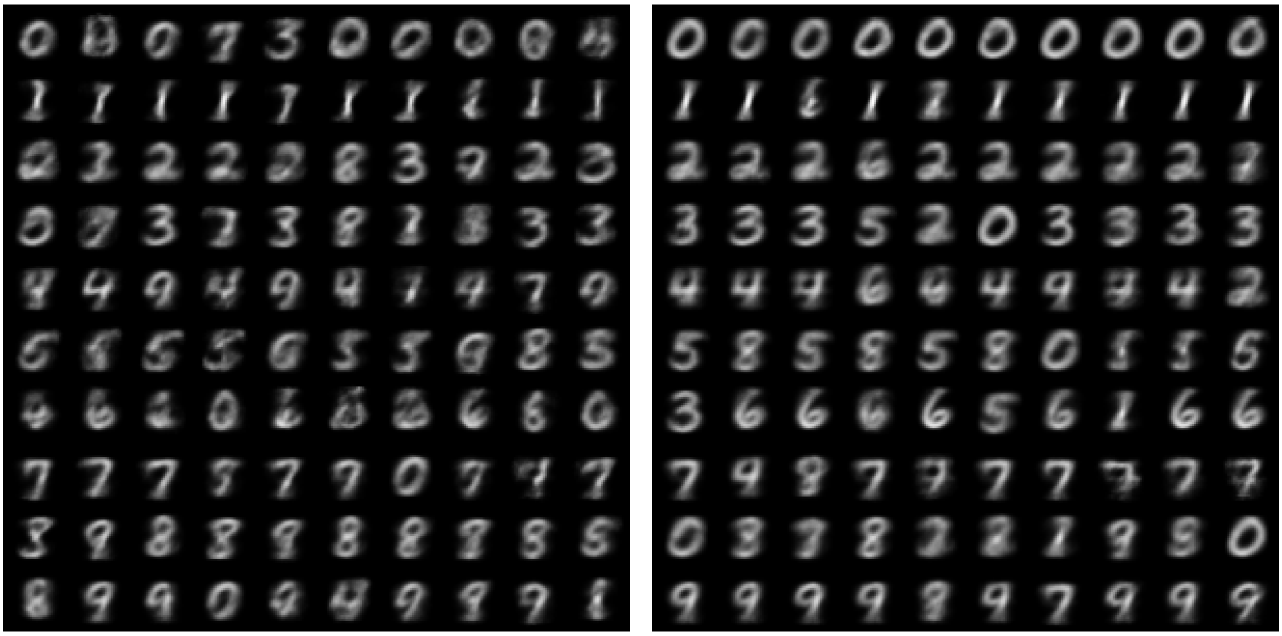
(f) CVAE (32)

(g) CAE (64)

(h) CVAE (64)

Figure A.5.7: Comparison of original and reconstructed MNIST digits using CAE and CVAE models with different latent space dimensions (8, 16, 32, and 64). Each pair of rows shows the original digits (top) and their reconstructions (bottom) for both model types across increasing latent space sizes, demonstrating the impact of latent dimension on reconstruction quality.

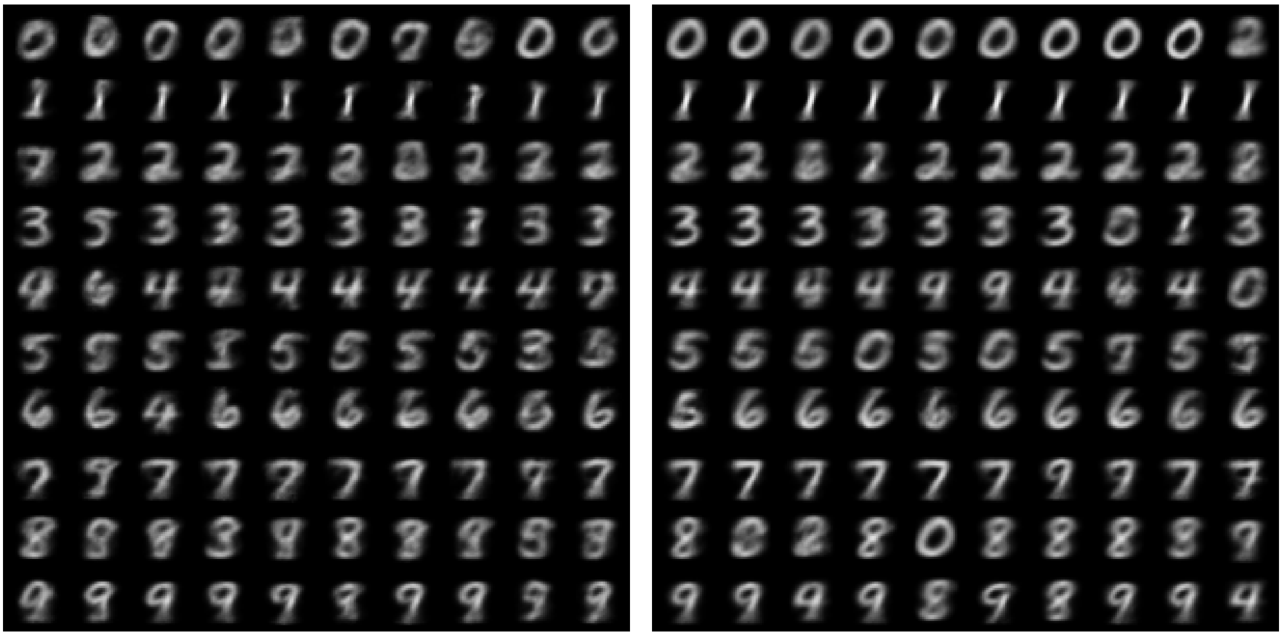
A.6 Generation Examples



(a) CAE (8)

(b) CVAE (8)

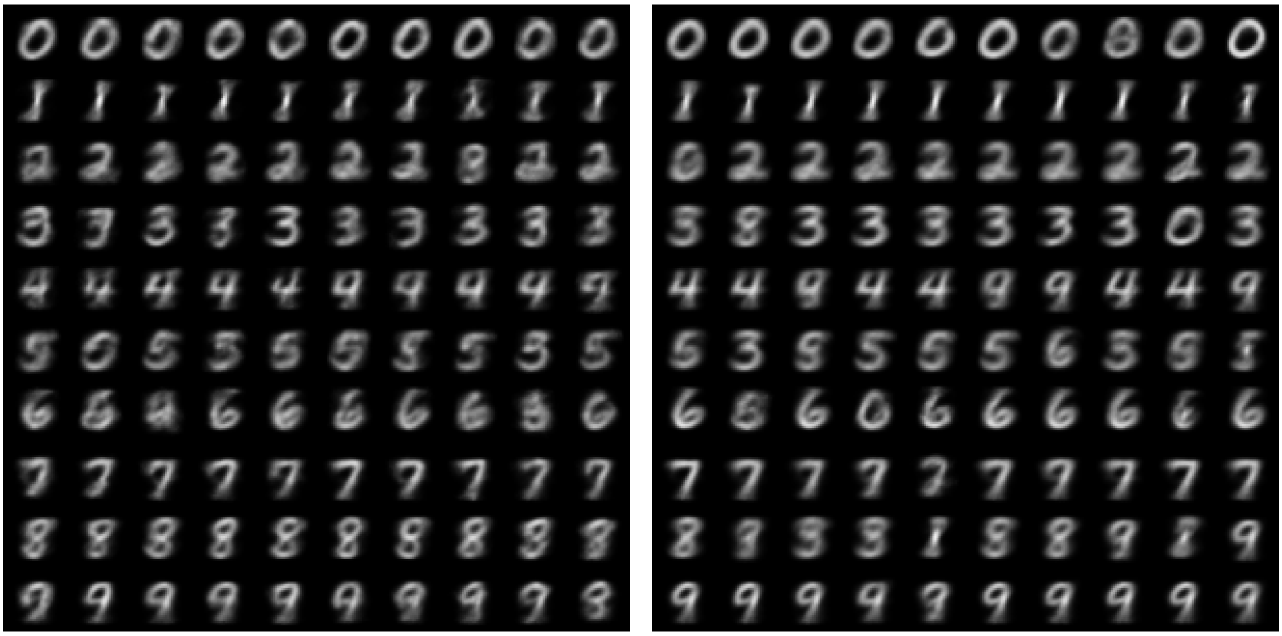
Figure A.6.8: Comparison of generated MNIST digits using (a) CAE and (b) CVAE models, both with a latent space dimension of 8. Each image shows a 10x10 grid of generated digits, demonstrating the generative capabilities.



(a) CAE (16)

(b) CVAE (16)

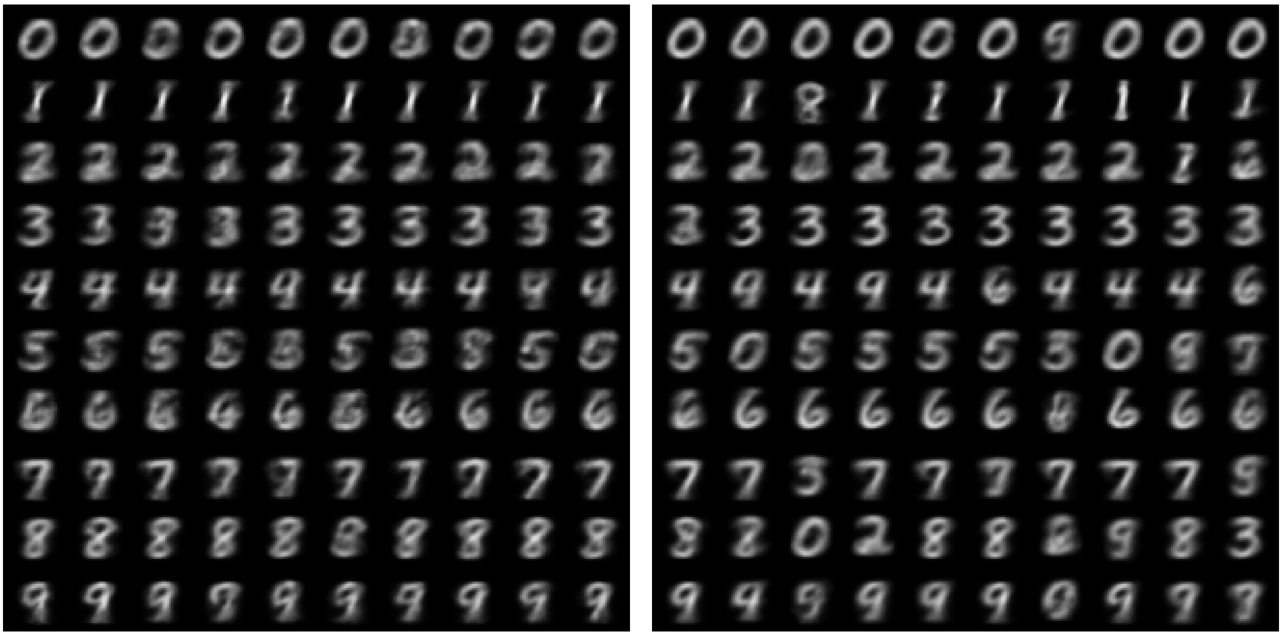
Figure A.6.9: Comparison of generated MNIST digits using (a) CAE and (b) CVAE models, both with a latent space dimension of 16. Each image shows a 10x10 grid of generated digits, demonstrating the generative capabilities.



(a) CAE (32)

(b) CVAE (32)

Figure A.6.10: Comparison of generated MNIST digits using (a) CAE and (b) CVAE models, both with a latent space dimension of 32. Each image shows a 10x10 grid of generated digits, demonstrating the generative capabilities.



(a) CAE (64)

(b) CVAE (64)

Figure A.6.11: Comparison of generated MNIST digits using (a) CAE and (b) CVAE models, both with a latent space dimension of 64. Each image shows a 10x10 grid of generated digits, demonstrating the generative capabilities.

AI Disclaimer

This work was conceptualized and executed by a human. The textual formulation and presentation of ideas in this report have been enhanced with the assistance of a large language model. All core scientific content, analyses, and conclusions remain the original work of the human author.