Homework 1: Conditional β-VAE on MNIST

CSC 8851 – Deep Learning - Fall 2025

Learning Objectives

- Implement and train a **Conditional Variational Autoencoder (CVAE)** on MNIST with a 2-D latent space.
- Extend the CVAE to a β**-VAE** and analyze the effect of β on disentanglement.
- Visualize latent structure via scatter plots, latent traversals, and 2-D manifolds.

Assignment Tasks (100 pts)

Part 1: CVAE + β-VAE Implementation & Training (40 pts)

- Implement a Conditional VAE with 2-D latent space.
- Train with $\beta = 1$ (baseline VAE) and $\beta = 4$ (stronger disentanglement).
- Plot training and validation curves for both settings (loss, reconstruction, KL) for $\beta = 1$ and $\beta = 4$ (see Fig. 1 and Fig. 2).

Deliverables: Code and figures for the training and validation (test data is used for validation) curves.

Suggested Architecture

- **Label embedding:** $\text{Emb} : \{0, \dots, 9\} \to \mathbb{R}^{16}$.
- **Encoder (CNN $\to$ MLP heads):**
  - Conv(1→32, k=4, s=2, p=1) → ReLU (28→14)
  - Conv(32→64, k=4, s=2, p=1) → ReLU (14→7)
  - Conv(64→128, k=3, s=1, p=1) → ReLU (7×7)
  - Flatten to $128 \cdot 7 \cdot 7$; *concat* label embedding (16) ⇒ Linear($128 \cdot 7 \cdot 7 + 16 \to 256$)→ReLU
  - Heads: Linear(256→2) for μ and Linear(256→2) for $\log \sigma^2$. We prefer to get $\log \sigma^2$ as output since the regularization loss uses $\log \sigma^2$. We get $\sigma = \exp(0.5. \log \sigma^2)$ for reparameterization.
- **Reparameterization:** $z = \mu + \sigma \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$; clamp $\log \sigma^2 \in [-8, 8]$.
- **Decoder (MLP + Deconv):**
  - *concat* z (2) with label embedding (16) ⇒ Linear(18→256)→ReLU
  - Linear(256→128·7·7)→ReLU → reshape to $(128, 7, 7)$
  - ConvT(128→64, k=4, s=2, p=1)→ReLU (7→14)
  - ConvT(64→32, k=4, s=2, p=1)→ReLU (14→28)
  - Conv(32→1, k=3, s=1, p=1) *(logits; no sigmoid in the module)*

Loss (with logits)

$$\mathcal{L} = \underbrace{\text{BCE\_with\_logits}(\hat{x}, x) \times 784}_{\text{recon}} + \beta \cdot \underbrace{\mathbb{E}\big[\tfrac{1}{2} \sum_{i=1}^{2} \big(\mu_i^2 + e^{\log \sigma_i^2} - 1 - \log \sigma_i^2\big)\big]}_{\text{KL}(q(z|x,y)\,\|\,\mathcal{N}(0,I))}$$

- Train two runs: $\beta = 1$ and $\beta = 4$; It is a good idea to linearly increase β to the final value within warmup_epochs. For e.g., $\beta_{\text{eff}} = \min(\beta, \beta * \text{epoch}/\max(1, \text{warmup\_epochs}))$, with warmup_epochs=8.
- Optimizer: AdamW (lr 3×10^{-4}, betas $(0.9, 0.95)$, wd 10^{-4}), batch size ≈ 128, grad clip 1.0.
- Init: small std (10^{-2}) for Conv/Linear; zero-init final decoder conv (weights & bias) for stable start.

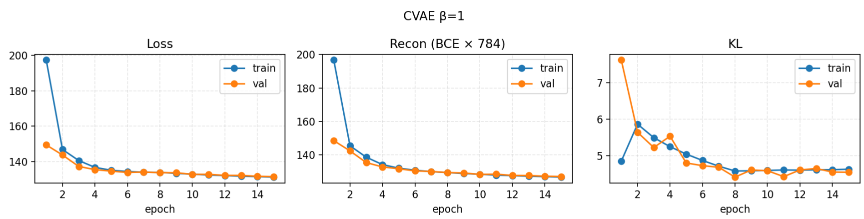


Figure 1: Training/validation curves for $\beta = 1$.

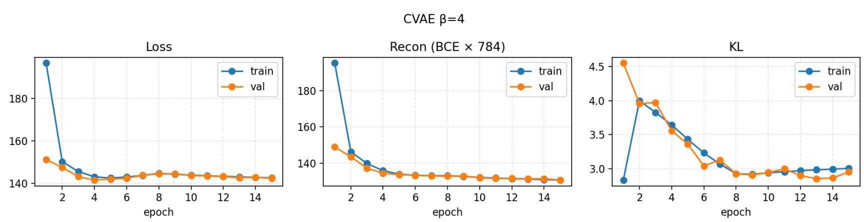


Figure 2: Training/validation curves for $\beta = 4$.

Part 2: Reconstructions & Conditional Generation (20 pts)

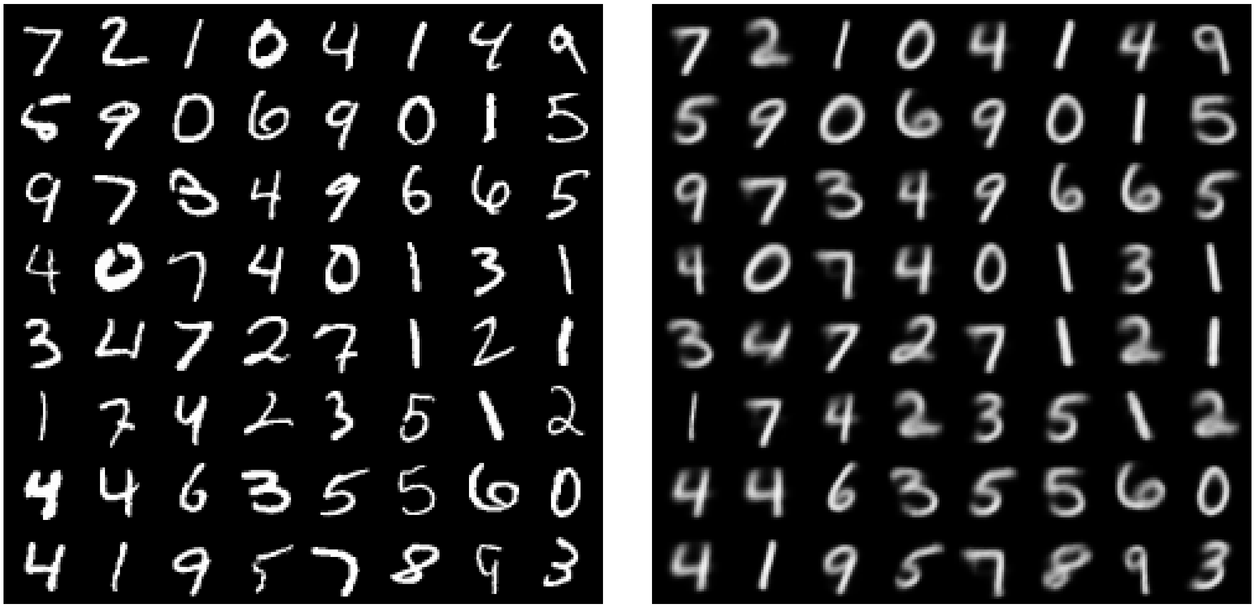
- Reconstruct test images and compare with inputs (see Fig. 3).
- Generate digits by sampling from the prior $p(z)$ and conditioning on labels (see Fig. 4).

Deliverables: Reconstruction grid, conditional generation grid.

Part 3: Latent Space Visualization (10 pts)

- Encode some of the test set into 2-D latent vectors.
- Plot scatter plots for $\beta = 1$ and $\beta = 4$, color-coded by digit label (see Fig. 5).
- Briefly comment on cluster separation.

Deliverables: Two scatter plots + observations.



(a) Input test images (subset). (b) CVAE reconstructions of the inputs.

Figure 3: Reconstructions example with $\beta = 1$.

Part 4: Latent Traversals & Manifold (30 pts)

- **Traversals:** For every label (digit), start with $z = [0, 0]$ and vary $z[0]$ and $z[1]$ independently. Show smooth changes for both the models (see Fig. 6 and Fig. 7).
- **Manifold:** Sweep a 2-D latent grid for a fixed label (digit) to produce an atlas of variations (see Fig. 8 and Fig. 9).

Deliverables: Figures for Traversal grid + manifold grid along with code

Submission Instructions

1. Submit a Jupyter Notebook named CSC8851_F2025_HW1_<YourName>.ipynb. Include all code, plots, and generated images. The notebook must run top-to-bottom.
2. Submit a PDF export of your notebook named CSC8851_F2025_HW1_<YourName>.pdf. DO NOT paste images of your code in the PDF. It will fail the plagiarism check. :
   - Using **nbconvert**:

     ```
     jupyter nbconvert --to pdf CSC8851_F2025_HW1_<YourName>.ipynb
     ```

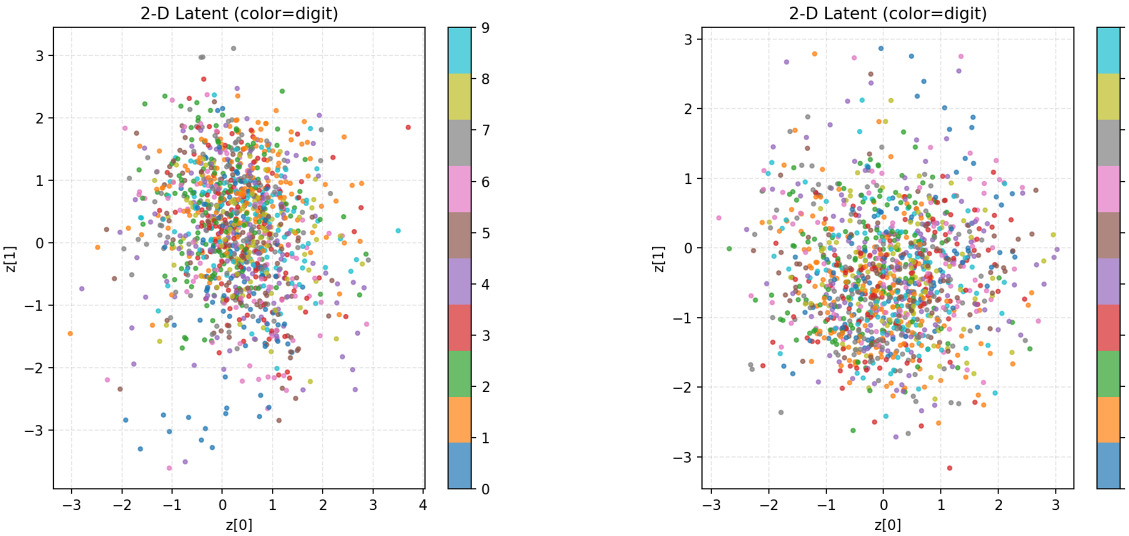
   - Or convert to HTML then print to PDF:

     ```
     jupyter nbconvert --to html CSC8851_F2025_HW1_<YourName>.ipynb
     ```

     Open the HTML in a browser and print to PDF.
   - Or use the Jupyter/Colab menu: File → Download as → PDF.
3. Upload both files (.ipynb and .pdf) to the assignment on iCollege.

Figure 4: Conditional generation: rows = labels 0–9; columns = samples ($z \sim \mathcal{N}(0, I)$) with $\beta = 1$.

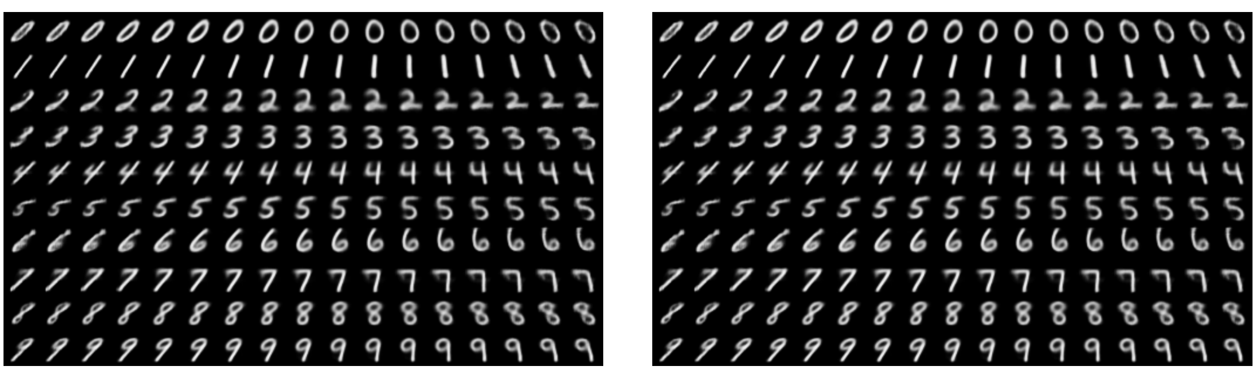
You may include additional qualitative results. Cite any code sources you consulted. Keep your code and discussion concise and clear.



(a) $\beta = 1$.

(b) $\beta = 4$.

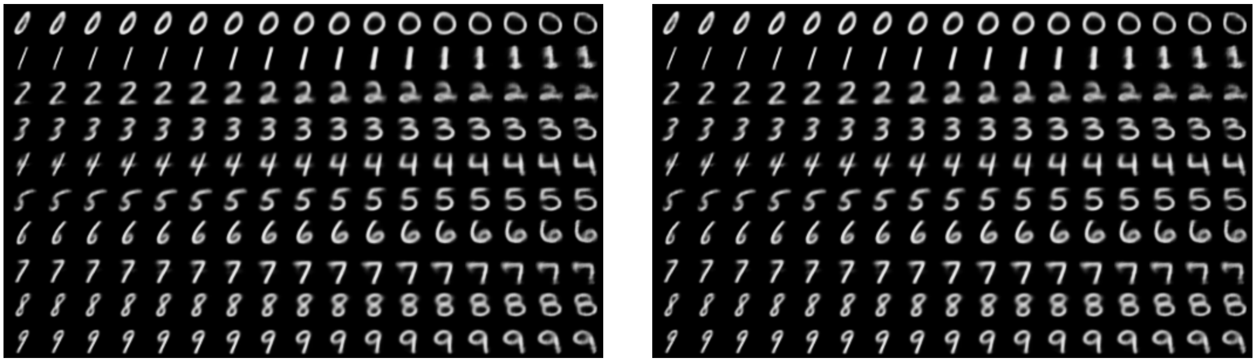
Figure 5: Scatter plot of latents for some test data for two models.



(a) Traversal along $z[0]$ (fixed $z[1]$), label 0.

(b) Traversal along $z[1]$ (fixed $z[0]$), label 0.

Figure 6: Latent traversals around the origin ($\beta = 1$) for all digits.



(a) Traversal along $z[0]$ (fixed $z[1]$), label 0.

(b) Traversal along $z[1]$ (fixed $z[0]$), label 0.

Figure 7: Latent traversals around the origin ($\beta = 4$) for all digits.

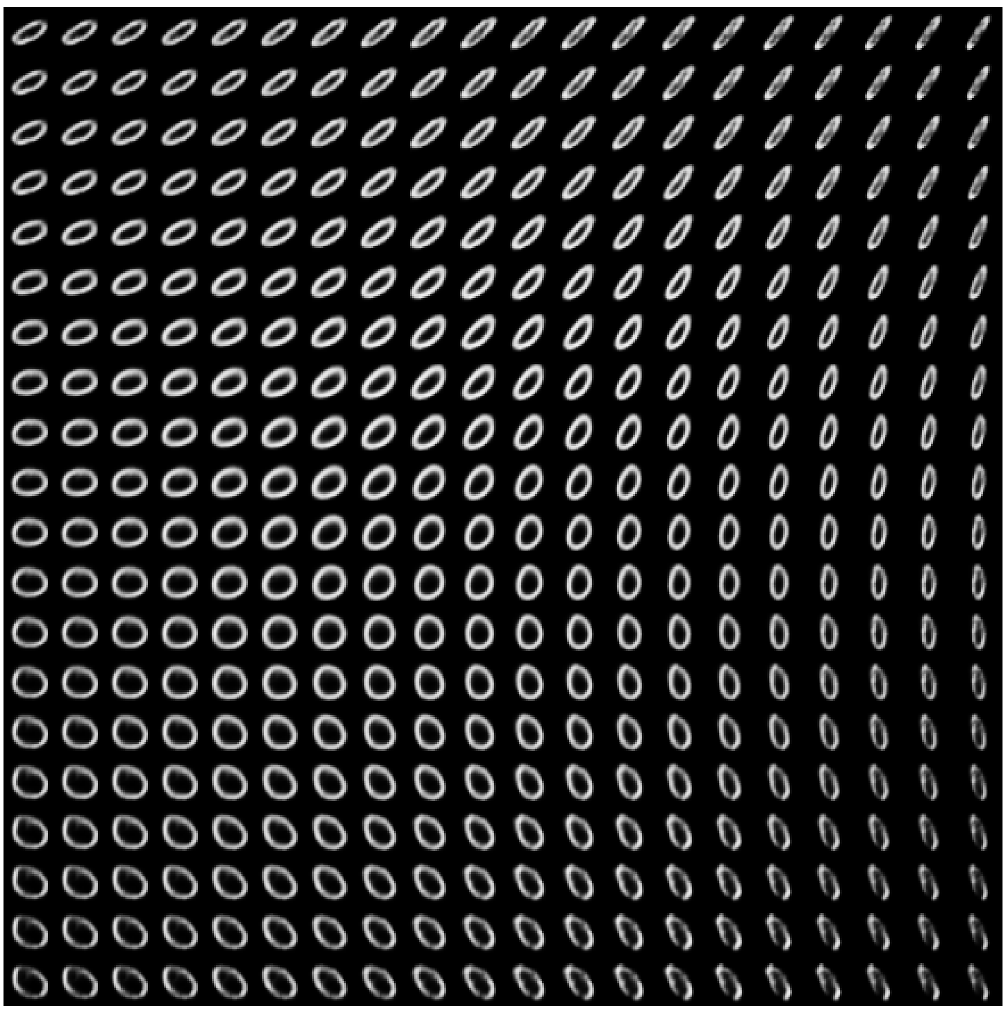


Figure 8: Latent manifold for label "0" over $(z_0, z_1) \in [-3, 3]^2$ with $\beta = 1$.

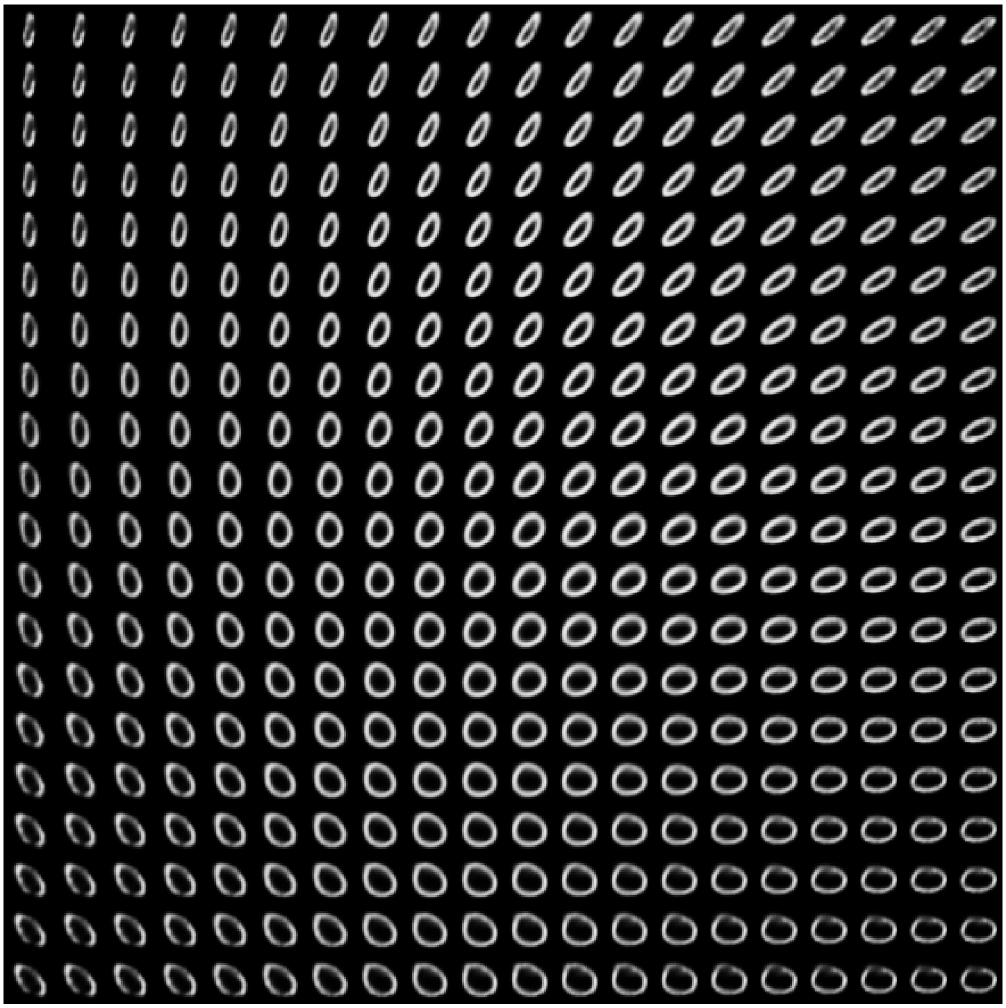


Figure 9: Latent manifold for label “0” over $(z_0, z_1) \in [-3, 3]^2$ with $\beta = 4$.