Advanced artificial intelligence techniques
Assignment 3
Adrian Dinu Urse

1. VAE Architecture Design

The implemented VAE comprises two main components:

- **Encoder:** Transforms input images into a latent space representation.
- **Decoder:** Reconstructs images from latent vectors.

1.1 Core components

They are implemented as described in the architecure guide lines within the homework

ConvNormAct - This module integrates convolutional layers with Group Normalization and the SiLU (Sigmoid Linear Unit) activation function. The convolutional layers extract spatial features from the input images, while Group Normalization stabilizes training across varying batch sizes. The SiLU activation introduces non-linearity, enabling the network to model complex data distributions effectively.

Residual Blocks – Each Residual Block consists of three consecutive ConvNormAct modules with a skip connection that adds the input directly to the output. This design facilitates the training of deeper networks by mitigating issues like vanishing gradients

1.2. Encoder Design

The encoder is tasked with mapping input images to a compact latent space. Its architecture is structured as follows:

- **Initial Convolution**: Begins with a ConvNormAct module that increases the number of feature channels from 3 (RGB) to a base number – 64
- **Downsampling Stages:** Incorporates a series of downsampling blocks, each comprising a Residual Block followed by a strided convolutional layer. The strided convolution reduces the spatial dimensions by half while doubling the number of channels, effectively capturing hierarchical features and compressing the input data.
- **Latent Space Projection:** After several downsampling stages, the feature maps are flattened and passed through fully connected layers to produce the mean (mu) and log-variance (logvar) vectors of the latent distribution.

1.3. Decoder Design

The decoder reconstructs images from the latent vectors, reversing the encoder's operations:

1. **Latent Vector Projection:** The latent vector (z) is first projected back into a high-dimensional feature space through a fully connected layer, preparing it for upsampling.
2. **Upsampling Stages:** Features a series of upsampling blocks, each consisting of an upsampling layer (e.g., bilinear interpolation) followed by a Residual Block. These stages progressively restore the spatial dimensions while reducing the number of channels, effectively reconstructing the image from the compressed latent representation.
3. **Output Layer:** The final layer applies a convolutional layer with a sigmoid activation function to produce the reconstructed image with normalized pixel values between 0 and 1.

1.4. Addressing Posterior Collapse

During the initial phases of training, the VAE encountered the phenomenon of posterior collapse, where the model's encoder ignored the latent variables, resulting in reconstructions that relied solely on the decoder's capacity. This issue undermines the generative capabilities of the VAE, as the latent space fails to capture meaningful representations of the input data.

1.4.1. Mitigation Strategies

To address posterior collapse, several strategies were explored:

- **KL Weight Adjustment:** I observed that assigning a small weight to the Kullback-Leibler (KL) divergence term allows the model to prioritize reconstruction accuracy over latent space regularization. By reducing the influence of the KL term, the encoder remains incentivized to utilize the latent variables effectively.
- **Batch Size Reduction:** Training with a smaller batch size of **16** was employed to introduce more variability and prevent the encoder from bypassing the latent space. Smaller batches can lead to noisier gradient estimates, encouraging the model to maintain meaningful latent representations to achieve consistent reconstructions.

1.4.2. Final Architectural and Hyperparameter Choices

After iterative experimentation, the following architectural and hyperparameter configurations were adopted to ensure effective training and meaningful latent representations

Latent Dimension (latent_dim): Set to **256** to provide a rich and expressive latent space capable of capturing diverse facial attributes

Base Channels (base_channels): Fixed at **64** to balance model capacity and computational efficiency.

Number of Downsampling and Upsampling Layers (num_down, num_up): Both set to **3**, enabling the encoder to progressively reduce spatial dimensions and the decoder to incrementally restore them.

Optimizer: Adam Optimizer with a learning rate of **1e-4**.

Batch Size: Reduced to **16** to introduce greater variability and prevent the encoder from neglecting the latent variables.

Beta Parameter (beta): Assigned a value of **0.5**, striking a balance between reconstruction fidelity and latent space regularization.

Epochs: The model was trained for **10** epochs, providing sufficient iterations for the model to learn meaningful representations without overfitting.

Loss Function: Employed a combination of **Mean Squared Error (MSE)** for reconstruction loss and **Kullback-Leibler Divergence** for latent space regularization.

The total loss is computed as:

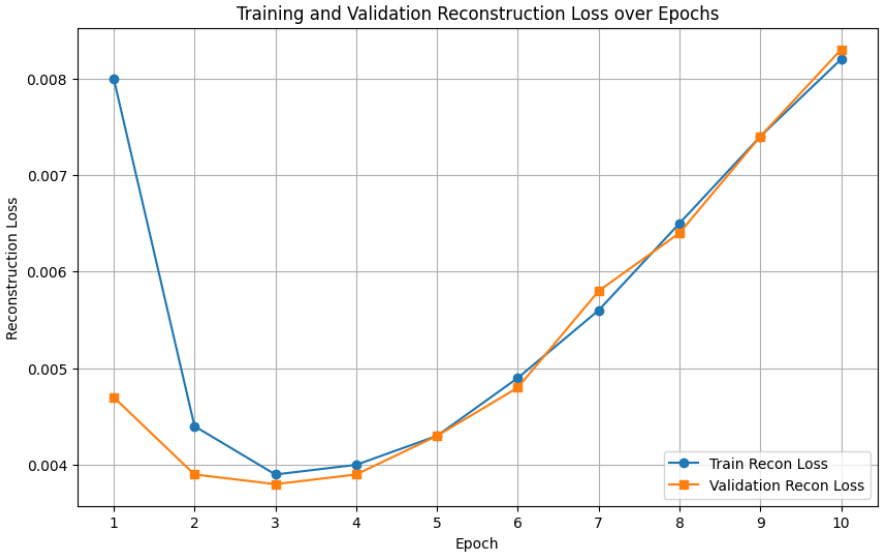
$$\text{Total Loss} = \text{Reconstruction Loss} + \beta \times \text{KL Weight} \times \text{KL Divergence}$$

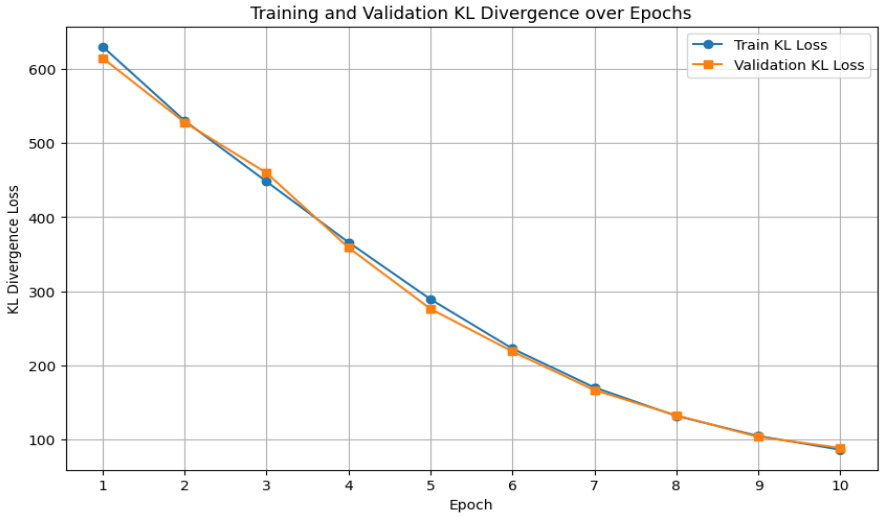
Where:

- $\beta = 0.5$ balances the influence of the KL divergence.
- **KL Weight** is dynamically adjusted using a sigmoid annealing schedule to gradually introduce the regularization term, preventing posterior collapse.

2. Results

Loss Graphs for Reconstruction and KL Divergence





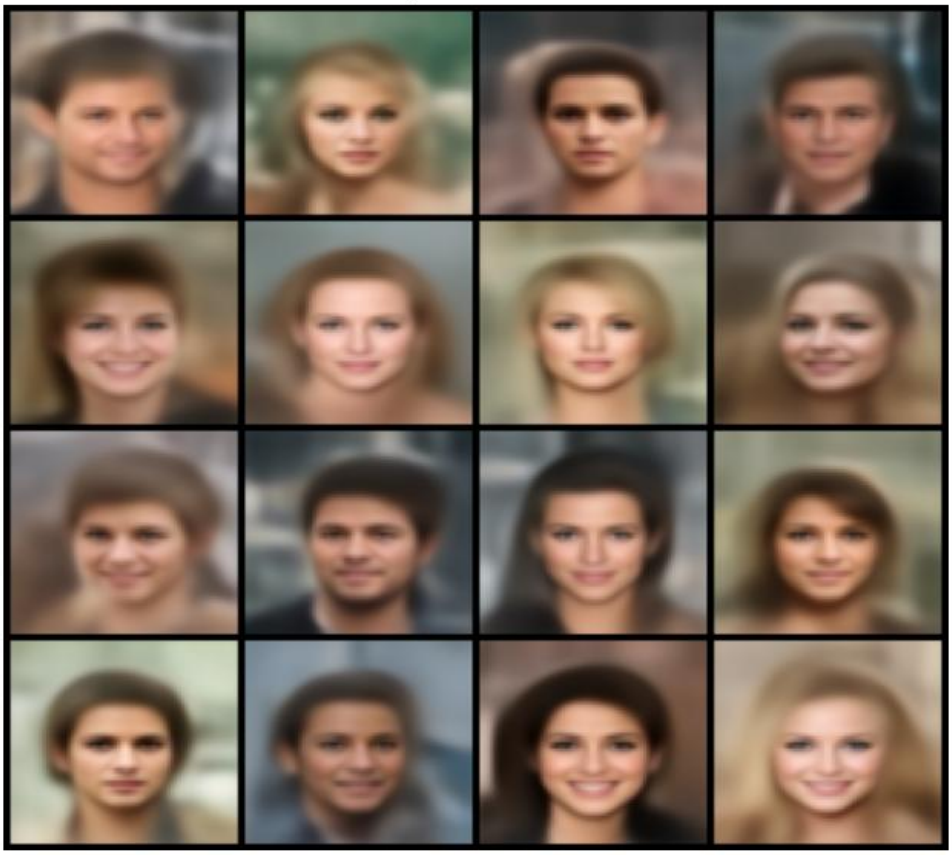
During the first three epochs, both the training and validation reconstruction losses exhibited a consistent decline. This trend indicates that the VAE was effectively learning to capture and reproduce the salient features of the input images. Following the initial improvement, the reconstruction loss for both training and validation datasets began to rise from the fourth epoch onwards. suggesting potential overfitting scenario, where the model starts to memorize the training data rather than generalizing from it.

Throughout the 10 training epochs, the KL divergence steadily decreased. This trend indicates that the model was increasingly aligning the learned latent distribution with the standard normal prior.

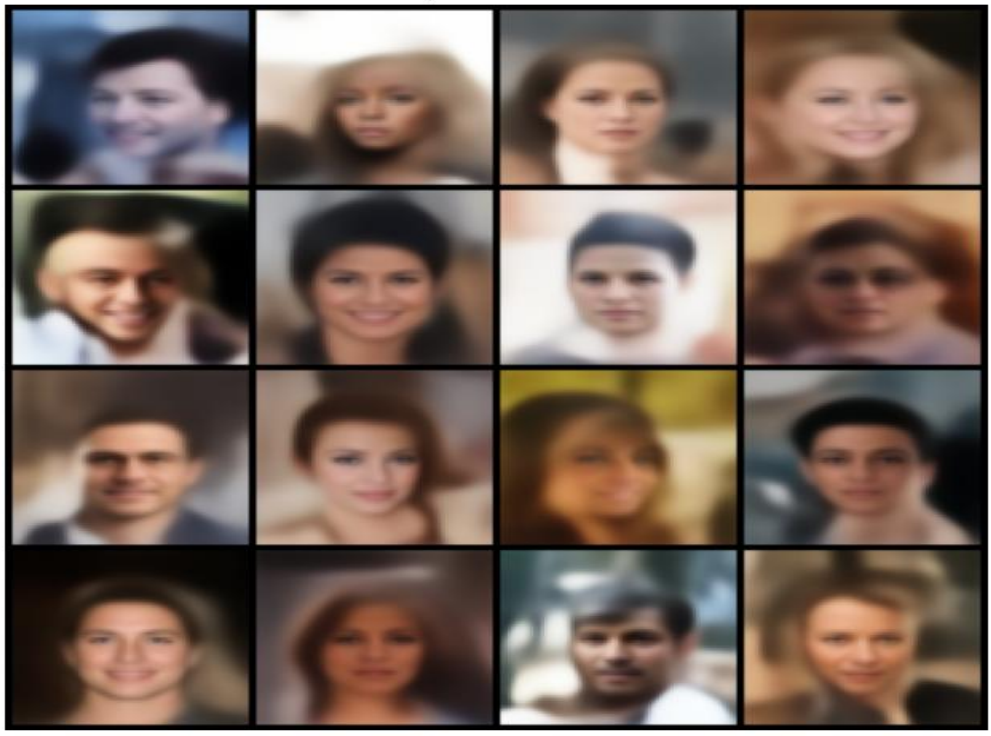
Random samples

Temperature = 0.2

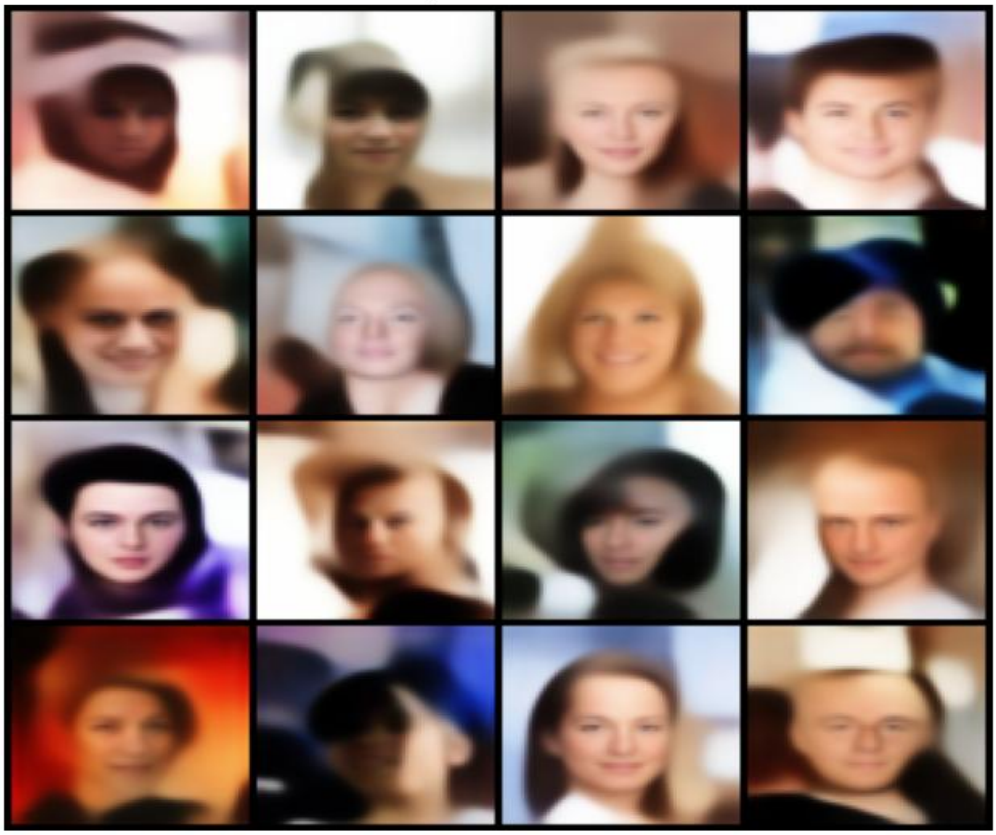
Temperature = 0.5



Temperature = 1.0



Temperature = 2.0



At Temperature T=0.2:

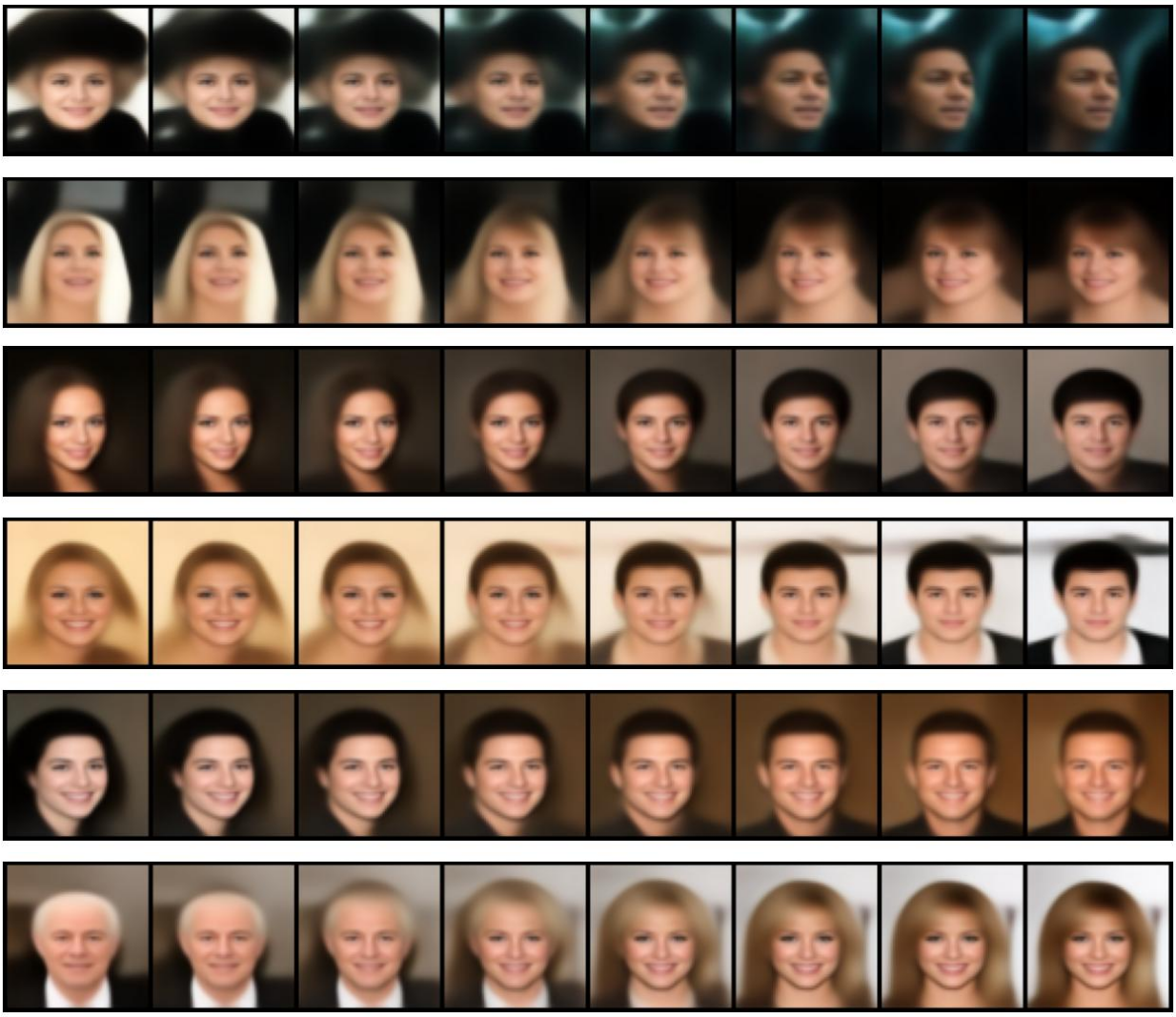
- The generated images exhibit high similarity.
- Facial features are relatively consistent across samples, with minimal variation.

At Temperature T=0.5:

- The generated images show moderate diversity while retaining coherence.
- Some variation in specific facial features (e.g., hair style, expression) begins to emerge.

At higher values of temperature the images are highly diverse, with a wide range of facial feature and attributes, but also start to be unrealistic or incoherent.

Interpolations



The model successfully generates a series of intermediate images that transition smoothly between the two endpoints.
Facial features such as **skin tone**, **hair style**, and **facial structure** change gradually without introducing abrupt or unrealistic artifacts.
The intermediate images blend features from both starting and ending images in a balanced manner, showcasing the model's ability to understand and represent subtle attribute changes.

3. Editing Pictures

To facilitate the editing tasks, I fine-tuned a pretrained ResNet18 model for multi-label classification of the 40 facial attributes available in the CelebA dataset. The classifier achieved a high accuracy of **0.91**, indicating its effectiveness in predicting attributes.

3.1 Feature Amplification

Objective: Feature amplification involves discovering latent dimensions in the VAE that are strongly correlated with specific attributes and amplifying those dimensions to observe their effects on reconstructed images.
To identify meaningful latent dimensions, I used the following methodology:
Correlation Analysis:

For each attribute, I computed the **Pearson correlation coefficient** between the latent dimensions (z) of the VAE and the predicted probabilities from the fine-tuned ResNet18 classifier.

Process: Once the relevant latent dimensions were identified, I performed amplification by modifying the latent vectors

$$Zi = Zi + \alpha,$$

α represents the amplification factor, varied across a range of values.

Selected the following attributes in this order:
Attribute: 'Attractive' --> Latent Dimensions: [48]
Attribute: 'Bangs' --> Latent Dimensions: [50]
Attribute: 'Big_Nose' --> Latent Dimensions: [150]
Attribute: 'Black_Hair' --> Latent Dimensions: [254]
Attribute: 'Blond_Hair' --> Latent Dimensions: [116]
Attribute: 'Gray_Hair' --> Latent Dimensions: [254]
Attribute: 'Heavy_Makeup' --> Latent Dimensions: [134]
Attribute: 'High_Cheekbones' --> Latent Dimensions: [204]
Attribute: 'Male' --> Latent Dimensions: [134]
Attribute: 'Mouth_Slightly_Open' --> Latent Dimensions: [204]
Attribute: 'Oval_Face' --> Latent Dimensions: [48]
Attribute: 'Pale_Skin' --> Latent Dimensions: [100]
Attribute: 'Smiling' --> Latent Dimensions: [204]
Attribute: 'Wearing_Lipstick' --> Latent Dimensions: [134]
Attribute: 'Wearing_Necklace' --> Latent Dimensions: [150]
Attribute: 'Wearing_Necktie' --> Latent Dimensions: [150]
Attribute: 'Young' --> Latent Dimensions: [254]

For each attribute, I plotted the reconstructed images across different values of α\alphaα, demonstrating a smooth and realistic transition.

Feature Amplification on a Single Image

Amplification resulted in **observable changes** in facial attributes:

- Increasing alpha for the latent dimension correlated with Smiling enhanced the smile intensity.
- Increasing alpha for the latent dimension correlated with Attractive enhanced the attractiveness of the subject.

The transitions were smooth and free from noticeable artifacts, preserving the realism of the images.

3.1 Label guidance

Objective:
The aim of label guidance is to modify the latent representation of a given image such that the classifier assigns specific desired attribute labels to the reconstructed image.
Latent Vector Optimization:

- Starting from the original latent vector z, I used **gradient-based optimization** to iteratively update z towards a target latent representation z'.
- The optimization goal was to minimize the **Binary Cross-Entropy (BCE) loss** between the classifier's predicted labels and the target labels.

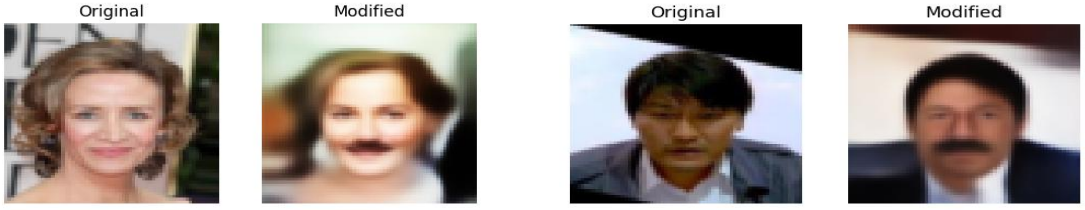
Gradient Descent Process:

- **Step 1:** Decode z using the VAE's decoder to reconstruct the image.
- **Step 2:** Pass the reconstructed image through the classifier to obtain predicted labels.
- **Step 3:** Compute the BCE loss between the predictions and the desired labels.
- **Step 4:** Backpropagate the loss and update z using an Adam optimizer.

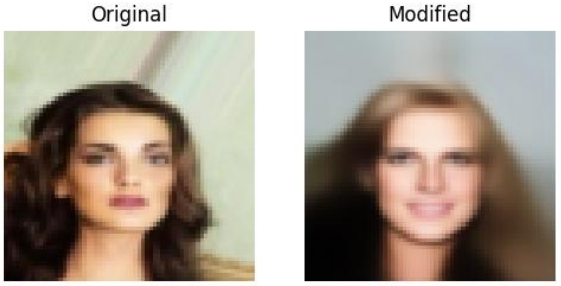
Examples:

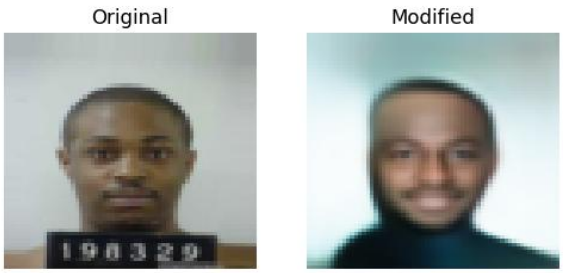
Applying label guidance: {'Blond_Hair': 1}

Applying label guidance: {'Mustache': 1}



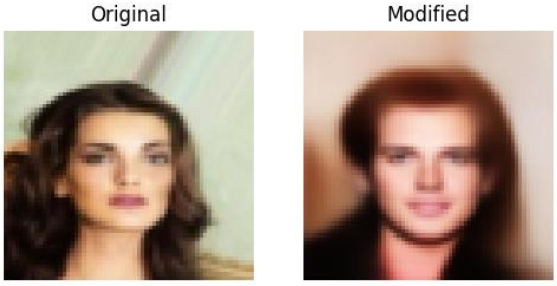
Applying label guidance: {'Smiling': 1}





Applying label guidance: {'Bangs': 1}

Applying label guidance: {'Male': 1}



The modified latent vectors produced **coherent and targeted edits** to the reconstructed images, successfully aligning them with the desired attribute labels.
The gradient descent optimization process converged smoothly, with the BCE loss decreasing steadily over the iterations.