Exploring Effectiveness of Multimodal Variational Autoencoders on Text-Image Tasks

Joseph Taewoo Kim
Stanford University
josephtk@stanford.edu

1 Abstract

This paper explores the effectiveness of a *multimodal variational autoencoder* (VAE) architecture on text-image related tasks. In particular, we evaluate the model's performance on *consistency checking* (discriminative) and *cross-modal generation* (generative) tasks. The experiments showed that even with a simple multimodal VAE architecture that utilizes a shared latent space, meaningful cross-modal connections can be learned. Experiments were conducted on the *CelebAMask-HQ* dataset, achieving an F1-score of 0.89 on consistency checking and qualitatively coherent cross-modal generation results.

2 Introduction

VAEs and multimodal learning are two fields at the cutting edge of AI research. VAEs learn latent representations of data through *dimensionality reduction*, enabling them to create useful embeddings for various tasks. Multimodal systems establish connections between different *modalities*, which are types of data such as text, audio and images, allowing for richer and complex interactions. Recent breakthroughs including DALL-E and GPT-4 demonstrate the effectiveness of cross-modal generation and comprehension, which provided inspiration for this project, particularly the focus on text-image modalities.

Multimodal learning between text and image has broad applications in fields such as fraud detection, fact-checking, medical imaging, e-commerce, and news media. With the abundance of information in the world today, it is imperative to maintain alignment between modalities for accurate and reliable outcomes.

This project focused on a multimodal framework that learns how to perform both consistency checking and cross-modal generation tasks. For consistency checking, the model assesses whether a given text-image pair is coherent and consistent with each other. For the generation task, the model either generates an image given a text or generates text given an image. This approach explores how well a VAE can perform both generative and discriminative tasks, providing insight into the model's flexibility, potential, and ability to leverage the shared latent space across different modalities.

We built a multimodal architecture made of two VAEs, one for processing text data and another for processing image data. Each VAE consists of an encoder and a decoder. Both pairs of encoders and decoders learn to process data for the latents in both modalities, not just a single modality. Given both text and image as input, the architecture can perform consistency checking, while given either a text or image it can reconstruct the original input or generate the other modality coherent with the input.

We ran experiments on the CelebAMask-HQ dataset, which provides images of faces and attributes for characteristics of the image. Text captions are curated from the attributes, and the image and text pairs are used as inputs. We ran a phased training approach for it to learn objectives sequentially, which resulted in promising outcomes.

All code and scripts used for our experiments are publicly available at `https://github.com/josephtkim/Multimodal-VAE`.

3 Related Work

The original VAE paper (Kingma and Welling, 2013) introduced a probabilistic framework for encoding inputs into latent representations, facilitating reconstruction and generation. Extensions to VAEs have since explored disentangling latent dimensions, incorporating vector quantization, and expanding to multimodal settings.

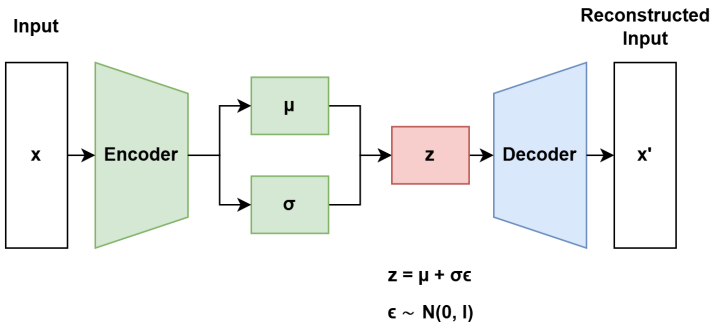

Figure 1: Diagram of a Variational Autoencoder (VAE).

The fields of multimodal and generative learning have seen significant advances over the years. Research has focused on improved representation learning, cross-modal alignment, and generative modeling strategies for integrated text-image tasks.

Cross-modal learning, in particular text-image alignment, gained traction with the advent of large-scale models like CLIP (Radford et al., 2021), which used contrastive learning on massive datasets of image-text pairs. It enabled zero-shot transfer capabilities and demonstrated the power of shared embedding spaces. Meanwhile, the Vision Transformer (ViT) (Dosovitskiy et al., 2020) adapted the transformer architecture to images, showcasing that attention-based models can rival or surpass convolutional approaches on vision tasks. Other attempts to do cross-modal tasks included AttnGAN (Xu et al., 2018), which worked on generating images from text descriptions. Although it did not reach the realism of some current models, it laid important groundwork for text-to-image synthesis.

Generative Adversarial Networks (GANS) (Goodfellow et al., 2014) established the generation of complex, high-fidelity images from random noise vectors. This early work laid a strong foundation for subsequent approaches that leverage learned latent spaces.

Beyond these representation-focused models, recent state of the art advances in latent diffusion (Rombach et al., 2022) have enabled the generation of high-resolution, photorealistic images from textual descriptions, moving towards more controllable and efficient generative frameworks. These innovations illustrate a trend toward combining textual and visual modalities in a single latent space, which is central to our approach.

While past research focused solely on generation or classification, our work aims to combine both consistency checking and multimodal generation within a shared latent representation. This project takes inspiration from the mentioned works to build a system that is able to understand and produce aligned text-image pairs.

4 Dataset and Features

4.1 Description and Selection

We use the CelebAMask-HQ dataset (Lee et al., 2020), introduced in the MaskGAN paper (Cheng-Han Lee, 2019), which is a high quality subset of 30,000 examples derived from the CelebA dataset (Liu et al., 2015) originally consisting of 200,000 examples. This dataset provides high-resolution images of celebrity faces, with 40 attribute annotations for facial characteristics, and values for pose (yaw, pitch, and roll). From the full CelebAMask-HQ dataset, we curate a subset of 5,000 examples, which is further split into a 90/10 split for training and validation. Images were sampled at random, with a 50/50 balance for male and female examples to ensure an equal representation of gender.

4.2 Preprocessing and Feature Extraction

We chose a focused subset of 9 attributes: young, male, smiling, eyeglasses, black hair, blond hair, bald, mustache, and wearing lipstick. We chose this subset of attributes because they are visually distinguishable, even at lower resolutions, have semantic importance for describing the faces, and for the computational efficiency of using fewer attributes. Furthermore, we add an inferred "female" attribute, when male is absent, bringing the total attributes to 10.

For preprocessing, the images are resized to 64x64 resolution to reduce computation requirements while still preserving details. Each image is normalized to a $[-1, 1]$ range to help with more stable training. Images are filtered by pose

Figure 2: CelebAMask-HQ dataset samples with natural language captions.

constraints (yaw, pitch, and roll) within some thresholds, to get more uniform and front-facing images. We also filter examples on a minimum present attribute count to enhance semantic richness.

We transformed the binary attribute annotations into a binary attribute vector, where each index corresponds to an attribute, with values of 1 or 0 indicating presence or absence of the attribute, respectively. The "female" attribute is inferred by the absence of the "male" attribute, and is added as a standalone attribute to help with learning, as the model will predict probabilities of attributes being present.

The dataset had good variability, so heavy data augmentations were not applied. Feature extraction was done on the normalized image pixel values and the attribute vectors directly, without additional feature extraction methods.

We curated natural language captions from the binary attribute annotations ("a young male who is smiling with blond hair.") with a few sample images and their corresponding images illustrated in Figure 2. When raw caption text is fed to the model, it is processed into the binary attribute vector, with each index corresponding to a particular "word-like" attribute token (e.g., "male", "smiling", "blond hair"). This attribute vector acts as a simplified textual modality, serving as a vocabulary of the semantic "words".

5 Methods

5.1 Background on VAE

A VAE (Figure 1) is a generative model that learns a latent representation z for some data x using two main components: an encoder and decoder. Instead of directly mapping inputs to a single latent vector, like a regular autoencoder, VAEs model a distribution over the latent space. The encoder outputs parameters μ and σ which define an approximate posterior distribution $q_\phi(z|x)$, typically a Gaussian. z is sampled using a *reparameterization trick* $z = \mu + \sigma\epsilon$ with $\epsilon \sim N(0, I)$. The decoder reconstructs x from the latent z. The reparameterization trick enables end-to-end training and allows gradients to flow through the random sampling, by effectively separating the randomness from the parameters.

Training a VAE involves optimizing the following objective:

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p(z))$$

$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ measures how well the decoder reconstructs the original input, and D_{KL} is the Kullback-Leibler divergence that regularizes $q_\phi(z|x)$ to stay close to a chosen prior $p(z)$, usually a standard normal, to ensure the latent space is well-structured and smooth. The result is a model that can learn a smooth latent space capturing underlying data variations, which allows VAEs to excel at tasks like image reconstruction and cross-modal generation when extended to multiple modalities.

While the equation for VAEs typically obtains the μ and σ to do the reparameterization trick to sample the latent vector z, in this project we use the log-variance $\log \sigma^2$ in our calculations, for benefits such as numerical stability and better gradient flow, while still preserving mathematical equivalence.

5.2 Architecture

Our architecture (Figure 3) integrates two VAEs—a Text VAE and an Image VAE—into a shared latent space. By enforcing shared latent representations, we enable discriminative and generative tasks:

- Given text and image:

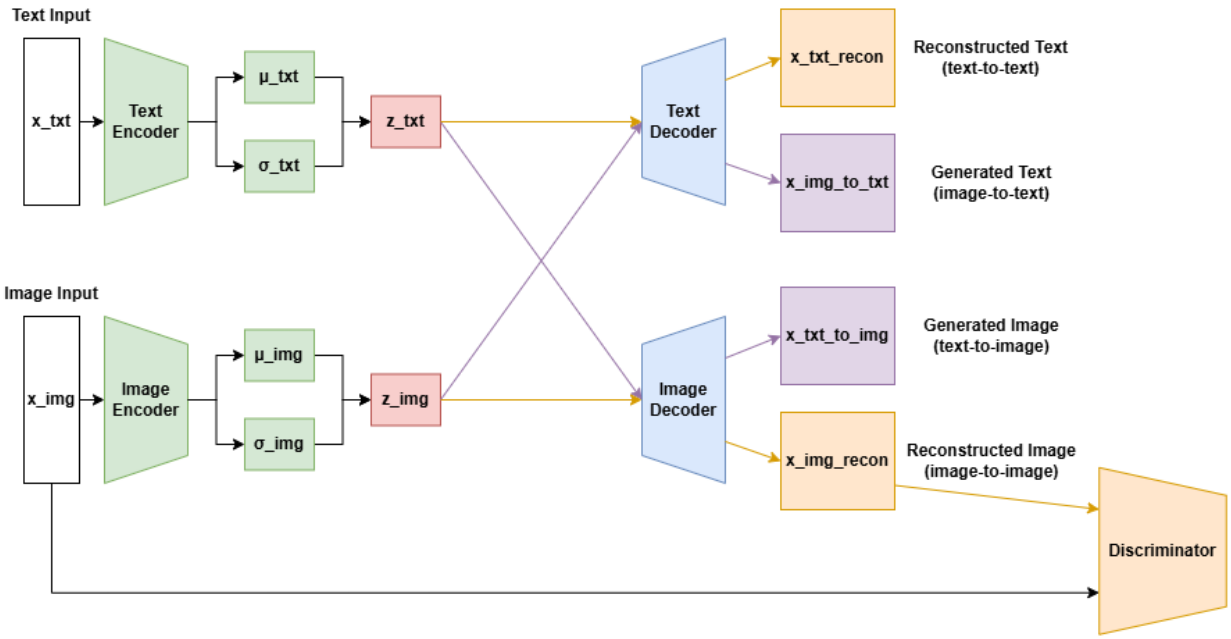

Figure 3: Multimodal VAE architecture with text-image pathways and cross-modal connections.

  - Compute a consistency score used to determine whether they match.

- Given text:
  - Reconstruct the original text.
  - Generate a corresponding image.
- Given image:
  - Reconstruct the original image.
  - Generate a corresponding text.

5.2.1 Text VAE Component

The Text VAE consists of an encoder and decoder which consist of fully connected layers. The encoder takes in a binary attribute vector, representing the vocabulary of semantic attribute "words". The output of this network is passed into two heads, one that produces the mean and one that produces the log variance of the latent distribution for the text embedding. The decoder has a final sigmoid activation which produces the probabilities for each facial attribute. Using an index-to-attribute mapping, this vector of probabilities is converted into a natural language caption.

5.2.2 Image VAE Component

The Image VAE consists of convolutional encoder and decoder networks to compress and reconstruct images. The image encoder downsamples the image with a flattening layer at the end, which is finally passed into mean and variance heads to produce the parameters for the latent distribution, similar to the text encoder. The decoder takes the latent vector and progressively upsamples it into the proper image shape.

During adversarial training, the Image VAE applies a patch discriminator, as used in PatchGAN (Isola et al., 2017), solely for the image-to-image reconstruction path. The discriminator takes the reconstructed image and classifies patches of the image as real or fake, helping maintain local image details. By improving the image decoder's ability to generate more realistic images in image-to-image reconstruction, it could also improve the cross-modal text-to-image generation performance, encouraging the text encoder to make its encodings closer to what the image decoder prefers.

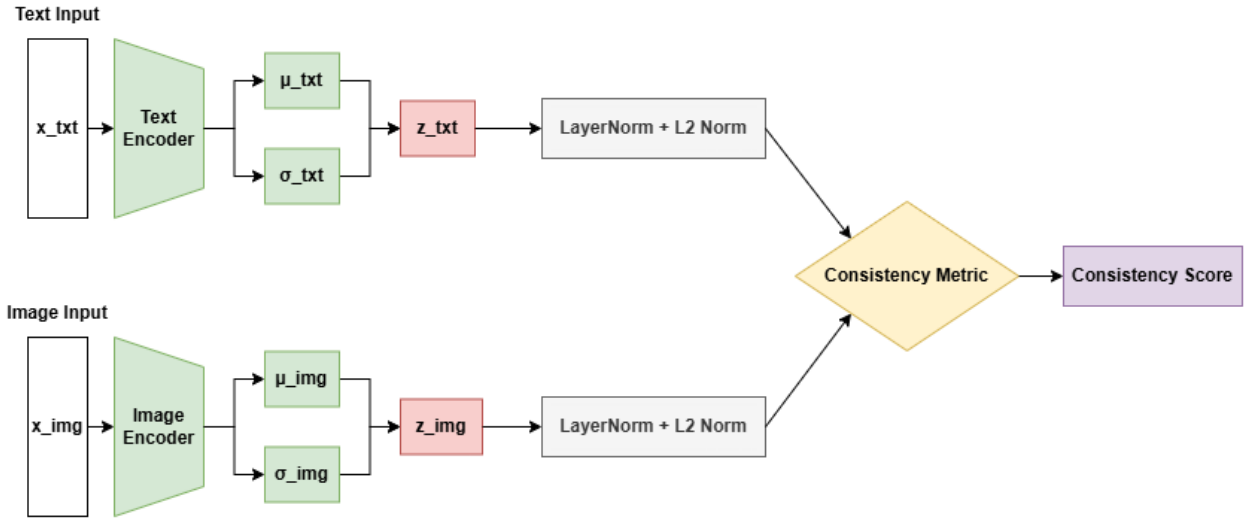

Figure 4: Consistency checking mechanism in shared latent space.

5.2.3 Network Components

VGG for Perceptual Loss. The model utilizes VGG16 pretrained on ImageNet for computing the perceptual loss. We only use the first 16 layers of VGG's feature extractor, keeping it in evaluation mode with frozen weights. During training, this helps ensure perceptual similarity between the generated and original images by comparing the feature representations instead of just pixel-level differences.

Residual Components. The architecture employs two types of residual components in the Image VAE's decoder:

- **ResidualBlock**: Convolutional residual blocks are used to process features through convolutional layers with batch normalization and LeakyReLU activations, adding the original input to maintain gradient flow and preventing degradation in deeper networks.
- **ResidualLinear**: Fully connected residual blocks are applied in the decoder's input processing, through linear layers with LeakyReLU activation, employing skip connections to help with gradient flow in dense layers.

These residual components help train deeper networks effectively by allowing direct gradient flow through skip connections while maintaining the ability to learn residual functions when needed.

5.2.4 Consistency Checking Mechanism

When provided both modalities, the model can perform consistency checking by comparing the latent encodings for image and text and assessing their alignment, depicted in Figure 4. Given an image-text pair, we encode both modalities to obtain their respective latent distributions:

$$q_{\phi_{\text{img}}}(z|x_{img}) = \mathcal{N}(\mu_{\text{img}}(x_{img}), \sigma^2_{\text{img}}(x_{img}))$$
$$q_{\phi_{\text{txt}}}(z|x_{txt}) = \mathcal{N}(\mu_{\text{txt}}(x_{txt}), \sigma^2_{\text{txt}}(x_{txt}))$$

where x_{img} is the input image and x_{txt} is the text. The consistency score is computed using the KL divergence between these distributions:

$$D_{\text{KL}}(q_{\phi_{\text{txt}}}(z|x_{txt}) \| q_{\phi_{\text{img}}}(z|x_{img})) = \frac{1}{2}\left(\log \frac{\sigma^2_{\text{txt}}}{\sigma^2_{\text{img}}} + \frac{\sigma^2_{\text{img}}}{\sigma^2_{\text{txt}}} + \frac{(\mu_{\text{img}} - \mu_{\text{txt}})^2}{\sigma^2_{\text{txt}}} - 1\right)$$

To note, While this mathematical formulation uses variances directly, the code implementation works with log variances for numerical stability, computing the same quantity using $\log \sigma^2$ terms.

We convert this divergence to a similarity score using:

$$s(x_{img}, x_{txt}) = \frac{1}{1 + D_{\text{KL}}(q_{\phi_{\text{txt}}}(z|x_{txt}))\|q_{\phi_{\text{img}}}(z|x_{img}))}$$

This transformation bounds the score between 0 and 1, where higher values indicate better alignment between the modalities. To classify pairs as matching or mismatched, we use a threshold τ determined empirically on a subset of the data:

$$\text{match}(x_{img}, x_{txt}) = \begin{cases} 1 & \text{if } s(x_{img}, x_{txt}) > \tau \\ 0 & \text{otherwise} \end{cases}$$

It considers the distribution of both matching and mismatching pairs. The optimal threshold τ is computed as a weighted average: $\tau = w_m * \mu_{match} + (1 - w_m) * \mu_{mismatch}$. The means are the mean similarity scores for matching and mismatched pairs from the subset. w_m is a weighting factor to favor matching pairs slightly, and is set to 0.6.

The threshold τ is selected to optimize classification accuracy on a held-out validation set, balancing the true positive rate for matching pairs against the true negative rate for mismatched pairs. In our experiments, using the model with 256 latent dimension, we found an optimal threshold of $\tau = 0.621$.

5.3 Loss Functions

Reconstruction Losses. The reconstruction objectives ensure accurate generation in both modalities:

- Image reconstruction combines Mean Squared Error (MSE) and perceptual loss using VGG features. The VGG perceptual loss compares feature representations of real and generated images through a pretrained VGG16 network, ensuring both pixel-level accuracy and semantic-level similarity in the reconstructed images
- Text reconstruction employs Binary Cross-Entropy (BCE) loss with a penalization factor for larger errors to ensure accurate attribute prediction
- Identity consistency loss helps maintain facial identity across reconstructions

Variational Losses. The variational components enforce proper latent space structure:

- KL divergence losses for both text and image VAEs ensure the learned latent distributions match the prior
- Distribution matching loss aligns the text and image latent spaces

Cross-Modal Losses. Cross-modal objectives ensure consistency between modalities:

- Attribute consistency loss ensures consistent predictions between image-to-text and text-to-image paths
- Cross-modal reconstruction losses measure how well each modality can be generated from the other

Adversarial Losses. The adversarial components improve image generation quality:

- Generator loss encourages the model to produce realistic images
- Discriminator loss helps distinguish between real and generated images

The total loss is a weighted combination of these components:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \lambda_r(\mathcal{L}_{\text{mse}} + \alpha\mathcal{L}_{\text{percep}} + \beta\mathcal{L}_{\text{identity}}) + \\ & \lambda_t\mathcal{L}_{\text{text}} + \lambda_c\mathcal{L}_{\text{consist}} + \\ & \lambda_k(\mathcal{L}_{\text{kl\_img}} + \mathcal{L}_{\text{kl\_txt}}) + \\ & \lambda_x\mathcal{L}_{\text{cross}} + \lambda_a\mathcal{L}_{\text{adv}} \end{aligned}$$

where λ terms are phase-specific weights controlling the contribution of each loss component.

5.4 Training Strategy

The training process is divided into three phases, each emphasizing different loss components.

5.4.1 Phase 1: Unimodal Training

Focuses on unimodal reconstruction to build strong modality-specific encoders and decoders. The weights are set as:

- Image reconstruction ($\lambda_r = 2.0$)
- Text reconstruction ($\lambda_t = 1.0$)
- Other components are zero or near-zero

5.4.2 Phase 2: Cross-Modal Alignment

This phase introduces consistency and distribution matching losses to align latent spaces. The weights are set as:

- Reconstruction weights ($\lambda_r = 1.0$, $\lambda_t = 1.0$)
- Adds consistency loss ($\lambda_c = 0.05$)
- Introduces KL divergence ($\lambda_k = 0.005$)
- Adds cross-modal weight ($\lambda_x = 0.1$)

5.4.3 Phase 3: Adversarial Refinement

The final phase adds adversarial training to improve image realism and finalize cross-modal generation performance. The weights are set as:

- Reconstruction weights ($\lambda_r = 1.0$, $\lambda_t = 1.0$)
- Increases consistency weight ($\lambda_c = 0.2$)
- Strengthens KL divergence ($\lambda_k = 0.01$)
- Increases cross-modal weight ($\lambda_x = 0.3$)
- Introduces adversarial component ($\lambda_a = 0.001$)

This phased approach allows the model to first learn good reconstructions, then align the modalities, and finally refine the generation quality.

6 Experiments

6.1 Setup

For the experiments, we utilized several libraries including PyTorch (Paszke et al., 2017), Pillow (Umesh, 2012), panda (McKinney et al., 2010), matplotlib (Hunter, 2007), seaborn (Waskom et al., 2017), sklearn (Pedregosa et al., 2011), and numpy (Harris et al., 2020).

6.2 Training Details

Hyperparameters and phase-specific configurations were put into a centralized configuration file. A random seed of 42 was used to allow reproducibility of the results.

For training, we used hyperparameters that provided a good balance between computational costs and performance. We ran experiments on models with latent dimensions of 128, 256, and 512. For most of the results in this paper, we use the outputs from the 256 latent dim model as it provided a sufficient capacity to encode the latents while avoiding too much overhead and also possible overfitting. We used a batch size of 16 to ensure good diversity during each training step, while also being computationally faster than larger batch sizes, and being more stable than smaller batch sizes. A learning rate of 10^{-4} was used for the generator (the multimodal VAE model itself). For adversarial training, specifically for image reconstruction, a learning rate of 10^{-5} was used for the discriminator, to prevent it from overpowering the generator.

To prevent overfitting, we used regularization techniques like gradient clipping and weight decay. A fixed 90/10 train-validation split was used for the 5,000 sample dataset, and validation performance is checked over each training step. Loss weights for each phase were set to the values mentioned in the Training Strategy section, and we ran phase 1 for 40 epochs, phase 2 for 30 epochs, and phase 3 for 70 epochs.

7 Results

7.1 Quantitative Results

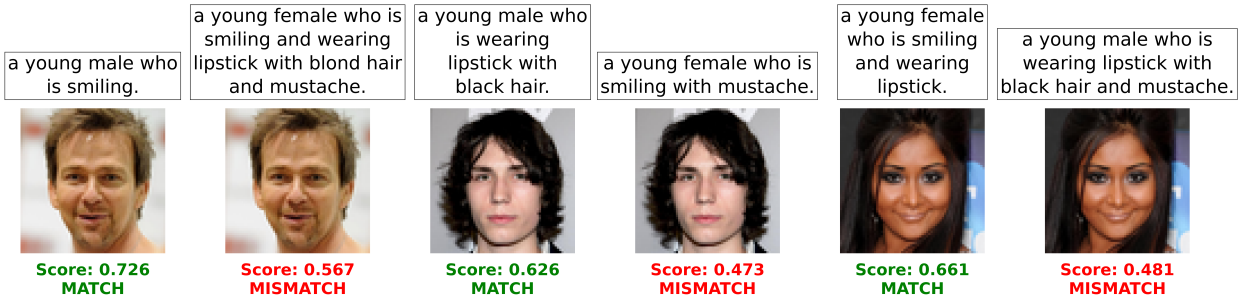

Figure 5: Consistency checking results for matched and mismatched image-text pairs.

7.1.1 Consistency Checking Framework

For the consistency checking task, we evaluated each sample with both its original text caption and a curated mismatching text caption, resulting in $2N$ total predictions (where N is the number of examples). The decision boundary threshold of 0.621 was determined using a random subset of 1,000 examples from the 5,000 example subset. Scores above this threshold indicate matching pairs, while lower scores indicate mismatches.

7.1.2 Mismatched Text Curation Strategy

To ensure semantic distinctiveness of mismatched pairs, we enforce a minimum of four attribute changes between matched and mismatched text captions. These changes were applied over the following attribute groups, with "young" excluded due to its subjectivity:

- Gender (male/female)
- Facial features (smiling, eyeglasses, wearing lipstick)
- Hair characteristics (black hair, blond hair, bald)
- Other features (mustache)

To maintain diversity, we apply changes over each of these attribute groups rather than concentrating changes within a single group. Additionally, we always assign the opposite gender in mismatched captions to ensure strong semantic differentiation.

7.1.3 Performance Analysis

We evaluated the model on the validation set of 500 examples (1000 total predictions). Figure 5 illustrates several matched and mismatched examples with their corresponding similarity scores and classifications. The confusion matrix results are presented in Table 1. For the different latent dimensions (128, 256, and 512), the models achieved accuracies between 0.88 and 0.92, with consistent F1 scores (Table 2). Interestingly, lower latent dimensions yielded better accuracy, precision, recall, and F1 scores. We hypothesize this is due to feature dilution in higher-dimensional spaces, which may weaken the alignment between image and text representations.

Table 1: Confusion Matrix for Latent Dimension 256.

	Predicted Mismatch	**Predicted Match**
Actual Mismatch	427	73
Actual Match	40	460

Table 2: Performance metrics for different latent dimensions.

Latent Dim	Accuracy	Precision	Recall	F1 Score
128	0.92	0.90	0.94	0.92
256	0.89	0.86	0.92	0.89
512	0.88	0.86	0.91	0.88

7.2 Qualitative Analysis

7.2.1 Cross-Modal Generation Performance

Our model demonstrated strong performance in cross-modal generation tasks. The image-to-text generation (Figure 6) achieved near-perfect accuracy, most likely due to the simplified textual representation scheme.

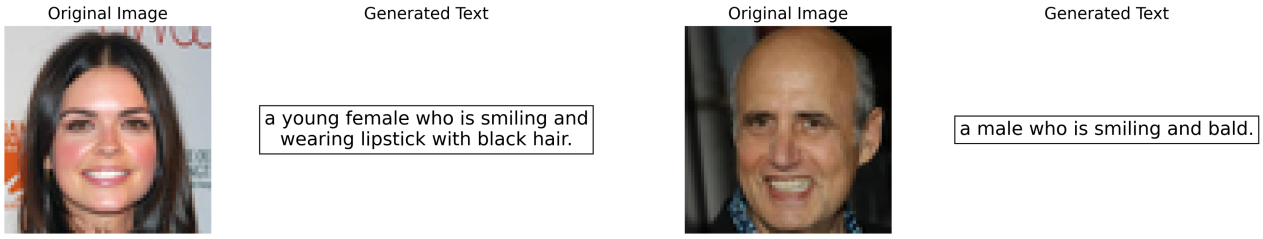

Figure 6: Image-to-text generation results.

For text-to-image generation (Figure 7), the output images clearly exhibit the attributes from the input descriptions, validating the model's ability to capture semantic relationships between modalities. The quality of the generated images was assessed through visual inspection. These qualitative results, particularly the cross-modal generations, confirmed effective interaction between the text and image latent spaces and demonstrate the robustness of our phased training approach.

Figure 7: Text-to-image generation results.

7.2.2 Training Dynamics Analysis

Figure 8 illustrates the evolution of training and validation losses across the three-phase training strategy for models with different latent dimensions (128, 256, and 512). The phase boundaries are marked by vertical dashed lines and show clear transitions where the losses get activated in later phases. While the loss patterns appeared similar across different latent dimensions, we observed that larger latent spaces consistently yielded lower losses. This suggested that increased dimensionality provided more capacity for storing and learning detailed representations.

7.2.3 Latent Space Analysis

We visualize the shared latent space using t-SNE projections (Figure 9). The visualization shows distinct clusters for male and female attributes in both text and image modalities, despite differences in scale between the two spaces. Connecting lines between corresponding image-text pairs demonstrate the model's ability to learn aligned representations across modalities. This clustering behavior confirms that the model successfully captured gender attributes at the latent level. Additionally, the right plot showing the text embeddings appears sparser than the left plot of the image embeddings, due to the simplicity of the binary attribute vectors, which can result in many overlapping points.

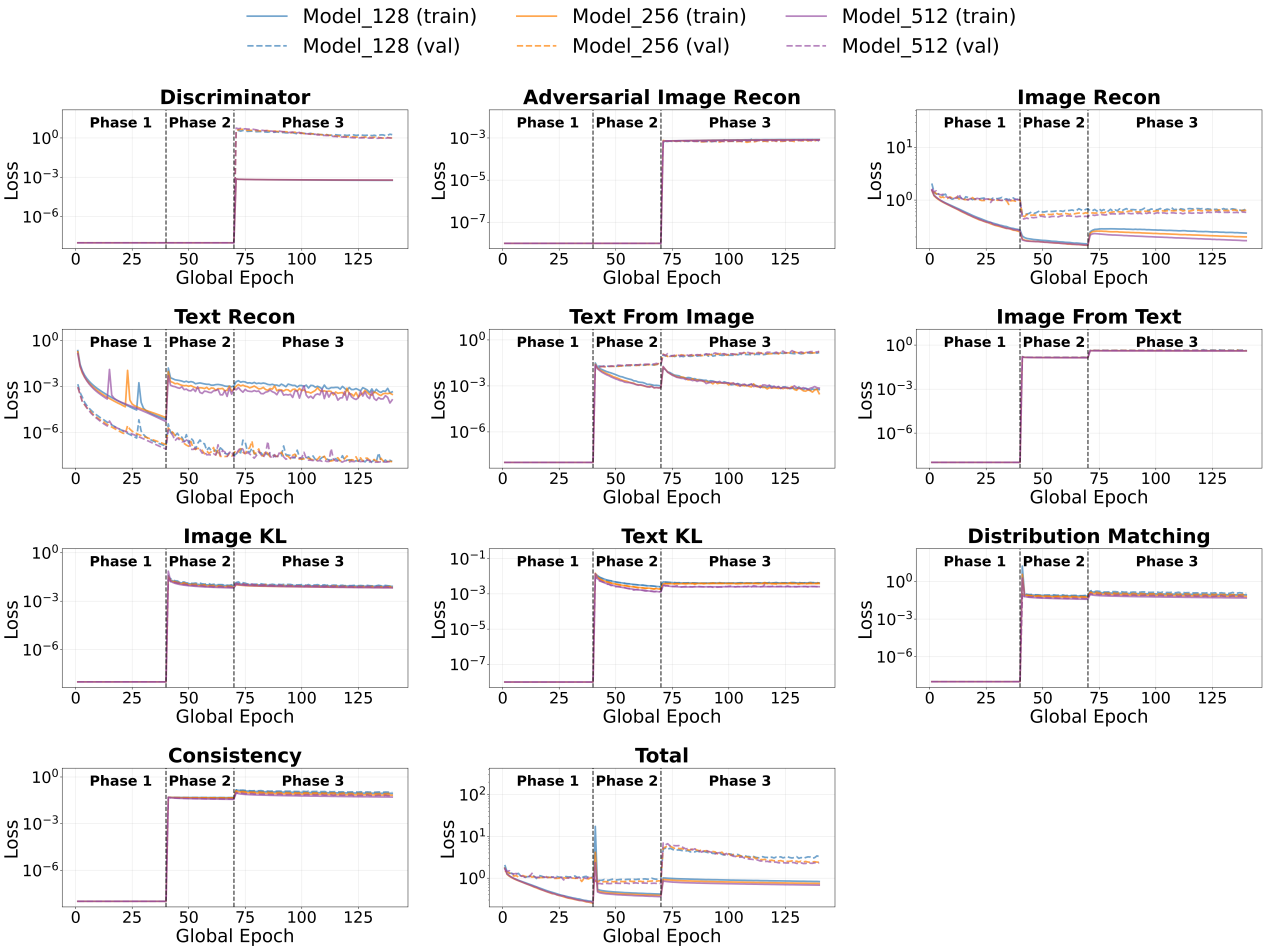

Figure 8: Training and validation losses across phases for different latent dimensions.

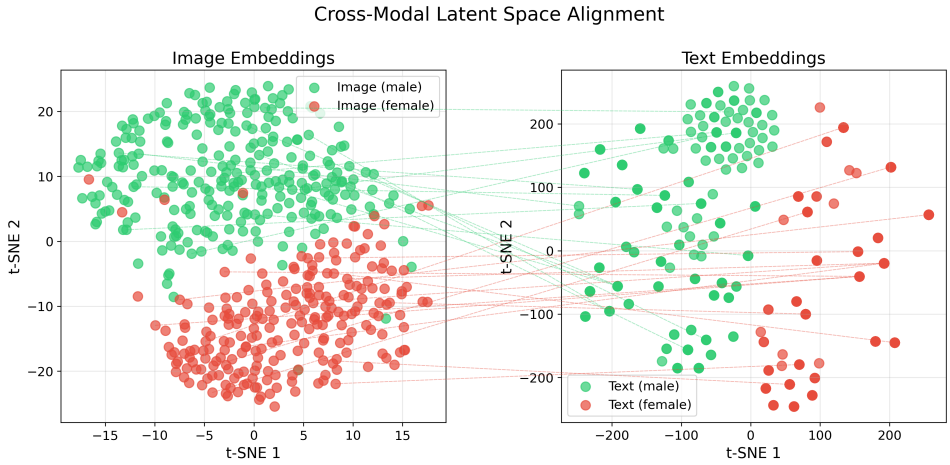

Figure 9: t-SNE visualization of image-text alignment in shared latent space.

8 Conclusion

8.1 Summary of Findings

Our work demonstrates the effectiveness of multimodal VAEs in learning meaningful cross-modal relationships, even with a relatively simple architecture. The model achieved strong performance across multiple tasks:

- Consistency checking with an F1-score of 0.89
- Qualitatively successful cross-modal generation between text and image modalities
- Effective shared latent space representation learning

The VAE framework proved effective for this task, offering two advantages over traditional autoencoders: probabilistic modeling of inherent variability in the data and smoother latent space representations. Additionally, the phased training strategy demonstrated consistent improvements over phases, culminating in successful multimodal integration.

8.2 Limitations

Our implementation has several constraints:

- Data Representation:
  - Binary attribute annotations create an information bottleneck, with limited expressiveness compared to continuous-valued attributes
  - Generated images tend toward attribute-specific averages rather than distinct individuals
- Dataset Constraints:
  - Limited to 5,000 examples (out of 30,000)
  - Only 9 attributes were used out of the original 40
  - Low resolution (64x64) images restrict output quality

8.3 Future Work

There are several promising directions for future expansion and improvement of this project. A few immediate refinements would be using a higher resolution for the images than 64x64, incorporating more binary attributes, including more samples from the dataset, and scaling up the architecture complexity. Exploring richer image-text datasets with more diverse and expressive text such as MSCOCO, Flickr30k, and LAION can be effective at addressing the information bottleneck that the binary attributes had. Furthermore, it can be promising to investigate techniques to improve the quality of the generated results by incorporating ideas like attention layers. Finally, more applications can be explored, such as modality fusion to combine image and text into fused outputs.

References

Lingyun Wu Ping Luo Cheng-Han Lee, Ziwei Liu. 2019. Maskgan: Towards diverse and interactive facial image manipulation. `https://arxiv.org/abs/1907.11922`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. `https://arxiv.org/abs/2010.11929`.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585:357–362.

John D Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134. Introduces PatchGAN discriminator architecture.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamala Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. `https://arxiv.org/abs/2103.00020`.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. `https://arxiv.org/abs/2112.10752`.

P Umesh. 2012. Image processing in python. *CSI Communications*, 23.

Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, Chris Fonnesbeck, Antony Lee, and Adel Qalieh. 2017. mwaskom/seaborn: v0.8.1 (september 2017).

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1316–1324.