Towards Controllable High-Quality Image Generation

Rajiv Chitale CS21BTECH11051
Vedant Bhandare CS21BTECH11007
Umanshiva Ladva AI22BTECH11016
Nitya Bhamidipaty CS21BTECH11041

Abstract

Generative models have made significant advancements in producing high-quality images across various domains. But tasks such as inpainting, object removal and feature-based image generation requires better controllability and representation learning. However, achieving fine-grained control over the generation process while maintaining image fidelity remains a challenging problem. In this paper, we explore novel techniques to improve both the controllability in latent space and the quality of images generated

1. Introduction

Image generation has been extensively studied using models such as Variational Autoencoders [5], Generative Adversarial Networks (GANs) [1], and Denoising Diffusion Probabilistic Models [2]. VAEs offer a structured latent space but tend to produce blurry images due to pixel-based losses. GANs, with their adversarial training framework, generate high-quality images but lack explicit control over attributes. Denoising Diffusion Probabilistic Models (DDPMs) achieve state-of-the-art sample quality but are computationally expensive and lack structured latent representations for controllability. Despite their success, these models generally lack direct mechanisms for fine-grained control over the generated images. To address this limitation, various approaches have been proposed to introduce controllability, enabling users to manipulate specific features of generated images. This survey explores models such as Variational Autoencoders (VAEs), Denoising Diffusion Probabilistic Methods and their extensions such as DiffuseVAE [10], Latent Diffusion Models (LDMs) [12] and ControlNet [16]. Further, it examines Generative Adversarial Networks (GANs) including Conditional GANs [9], ContraGAN [3], VAE-GAN [6], and StyleGAN [4], discussing their methodologies, advantages, and trade-offs.

1.1. Problem Statement

While traditional generative models can generate high-quality images, they often fail to provide intuitive mechanisms for manipulating specific attributes such as structure, style, or spatial composition. This survey explores advancements over traditional generative models that improve control over generated image features while preserving high-quality generation.

2. Literature Survey

We surveyed the following literatures and their approaches towards achieving controllability in generating desired images.

2.1. Auto-Encoding Variational Bayes (VAE)

The paper assumes that a dataset is generated by a random process based on a continuous latent variable. A probabilistic decoder learns to generate a distribution of data points x given a latent variable z. Simultaneously, a probabilistic encoder is trained to produce a latent distribution for z from which the datapoint x could have been generated. The encoder and decoder can be combined into a variational autoencoder (VAE), which can be used for tasks such as image denoising or inpainting. The latent vector provides controllability, but the reconstructed images are often blurry.

This paper constructs an approximate lower bound for the likelihood of a datapoint. This allows for a loss formulation despite the intractability of the posterior term $p(z|x)$. This bound consists of two terms. First, a regularizer term brings the approximate posterior closer to a prior, such as a Gaussian. The second term is an expected negative reconstruction error. A reparameterization trick allows differentiation even with sampling.

2.2. Denoising Diffusion Probabilistic Methods

This work uses diffusion models to generate high-quality image samples. A forward process is defined using a Markov chain that gradually adds Gaussian noise to data. After a finite number of steps the signal is destroyed. The reverse process is learnt to predict and remove the noise at each timestep. This is used to obtain the original image from a pure noise sample.

The schedule for the variance of noise added in the forward process can be fixed or parameterized. This allows

more noise to be added at the final timesteps compared to initial timesteps. A U-Net[13] based model. Its output dimensions are the same as the input. DDPMs produce high-quality samples but lack a meaningful latent representation or method of control.

2.3. DiffuseVAE

It uses a generator-refiner framework, consisting of two stages. First, a VAE is used as a generator. This allows for controllable image synthesis using low-dimensional latent vectors. Second, a DDPM is used as a refiner, to generate high-quality images from the blurry images resulting from the first stage. It also takes fewer diffusion timesteps to achieve the same quality compared to a standard DDPM. Training is done in stages. The VAE is trained first, after which its weights are fixed. Then the DDPM is trained. The latent vector and decoder controls the overall structure and diversity of generated samples, whereas the DDPM controls high-frequency details.

2.4. Generative Adversarial Networks

Generative adversarial networks (GAN) are implicit generative models that use a generator and a discriminator to synthesize realistic images. While the discriminator (D) should distinguish whether the given images are synthesized or not, the generator (G) tries to fool the discriminator by generating realistic images from noise vectors.

2.5. Conditional Generative Adversarial Networks

One of the most widely used strategies to synthesize realistic images is to utilize the information from the class label.Early approaches in this category are conditional variational auto-encoder (CVAE)[14] and conditional generative adversarial networks[9].These approaches concatenate a latent vector with the label to manipulate the semantic characteristics of the generated image.The most common approach of conditional GANs is to inject label information into the generator and discriminator.The Generator(G) and the discriminator(D) in the value function of GAN described in[1] are both conditioned on some condition **y** which can be any form of modality.

2.6. ContraGAN

The idea is to add a self-supervised learning or metric learning objective[3] in the discriminator (D) and generator (G) to explicitly control the distances between the embedded image features depending on the labels. It learns the data-to-data relations along with data-to-class relations by contrastive approach. The loss function is analogous to InfoNCE used in CLIP[11], but also tries to capture or learn and exploit the intra-class characteristics and higher order image representations of the real images, to generate more realistic images. The discriminator updates itself by minimizing the distances between real image embeddings from the same class while maximizing it otherwise. The generator then exploits the knowledge of discriminator to generate more realistic images.

2.7. VAE-GAN

By integrating adversarial training into VAEs, this hybrid approach generates sharper images while retaining meaningful latent representations.

Variational Autoencoders (VAEs) provide explicit latent representations of images, making them useful for structured image manipulation. However, they tend to produce blurry reconstructions due to their pixel-based loss function. On the other hand, Generative Adversarial Networks (GANs) generate high-quality images but lack a straightforward way to infer latent representations of a given image. VAE-GAN [6] integrates VAEs and GANs to leverage their strengths.

In VAE-GAN the generator and decoder are combined, and a discriminator is trained to differentiate between real images, reconstructed images, and generated samples (where z is sampled from a standard normal distribution and passed through the decoder). To mitigate the blurriness typical of VAEs, the traditional pixel-based reconstruction loss is replaced with a feature-based reconstruction loss, extracting features from an intermediate layer of the discriminator. This results in more perceptually coherent reconstructions, as small spatial shifts in an image do not disproportionately affect the loss.

The learned latent representations capture semantic attributes, allowing attribute-based image modifications via vector arithmetic. Given an image's latent code, adding or subtracting attribute vectors enables controlled edits, such as changing facial expressions or styles.

2.8. Style-GAN

Style transfer models aim to apply the styles of a reference image to the generated image. This paper [4] introduces an innovative generator architecture that enables the blending of two generated images by merging their styles at different layers.

Instead of directly utilizing the latent vector, StyleGAN maps it to an intermediate latent space for a better separation of semantic features. This leads to overall better latent space disentanglement.

Styles (i.e different attributes) are extracted from the latent vector at different layers and applied hierarchically through AdaIN [4] Normalization layers. The models learns to extract different attributes at different layers. This structured approach grants distinct levels of control: early layers extract global attributes like face shape and position, middle layers influence hair color and style, and final layers refine

intricate details such as skin texture and background color.

There are many aspects in human portraits that can be regarded as stochastic, such as the exact placement of hairs, freckles, etc. For this reason, noise is injected at different layers to add this stochasticity needed for generation. Similar to styles, noise at different layers affected different parts of the generated images. The absence of noise generated a painting-like image, and adding noise closer to the output gave finer hair and textures.

After training the model, different generated images can be fused by mixing their styles at different layers. Lower layers control fundamental facial features and positioning, middle layers shape attributes like hair, and the final layers influence fine details such as hair texture and color.

2.9. Latent Diffusion Models

Diffusion Probabilistic Models (DPMs) have two limitations - low inference speed and high training costs. LDMs[12] aim to address both. LDMs first train an autoencoder to learn a lower-dimensional latent space where the data is represented. Then, a diffusion model is trained on this latent space rather than the original image space.

LDMs scale more gently to higher-dimensional latent spaces, i.e. as we increase the size of the latent space, the computational burden doesn't grow as sharply as it would in pixel-space diffusion. We can choose the level of compression, latent space should neither be too large (might as well train in pixel space) nor too compressed (might lose image details). This model also provides controllability - text-to-image tasks use a cross-attention mechanism during the denoising process, whereas image-to-image tasks supported (super-resolution, inpainting, object removal) High quality image generation is supported

2.10. ControlNet

Text-to-Image models are limited to textual conditions, do not capture spatial conditions like edge maps, human skeleton pose, segmentation maps. ControlNet runs a copy of pre-trained Stable Diffusion Model in parallel. The original pre-trained model is locked and only the copy is trainable. Input to SD is text prompt and input to trainable copy is a control vector c. The trainable copy is linked to the frozen SD model using ZeroConv layers. This ensures gradual learning, i.e. in the beginning, SD output is not altered by the trainable copy as ZeroConv would be all 0s.

The conditional image inputs are also taken reduced to latent space of the same dimension as that of the input image to SD. This ensures similar computational complexity to that of LDM. Multiple Image controls can be combined (depth + pose + canny edges). ZeroConv ensures that training does not degrade SD's generation quality, thus generating high quality images as before.

3. Results Replicated

3.1. VAE [5]

To introduce controllability in image generation, we began by training a Variational Autoencoder (VAE) on a subset of the CelebA dataset, comprising approximately 80,000 facial images. The VAE learns a compressed latent representation of the images while preserving the ability to reconstruct them. Once trained, we encoded the entire dataset through the VAE encoder to obtain the corresponding latent vectors, effectively capturing the learned latent space.

To explore meaningful directions in this space, we applied Principal Component Analysis (PCA) on the collected latent vectors. The objective here was to identify principal axes in the latent space — each axis representing a dominant direction of variation in the data. These directions, or eigenvectors, correspond to semantically interpretable modifications in the image space.

By linearly perturbing a latent vector z along one of the top k eigenvectors (i.e., generating $z' = z + \alpha \cdot e_i$, where e_i is the i^{th} eigenvector and α is a scalar), we can systematically observe how the generated image changes. This method allows us to control specific aspects of the image, such as background blur, hair color, or skin tone, by simply modifying the latent code.

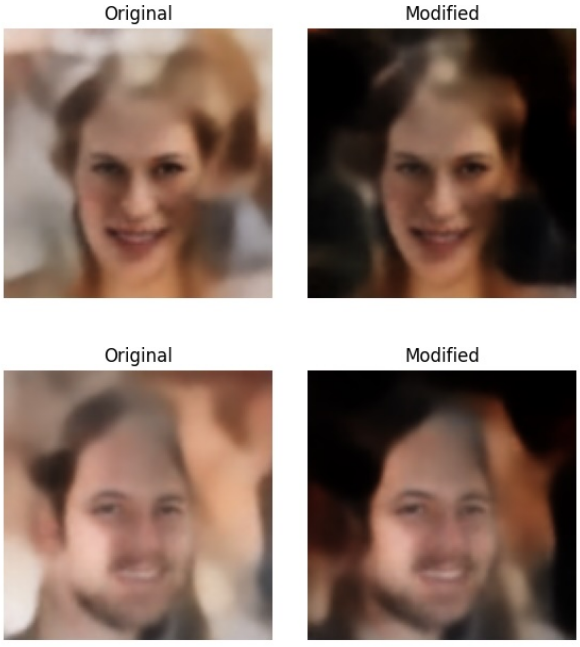


Figure 1. Effect of perturbing the latent vector along a principal direction - background removal

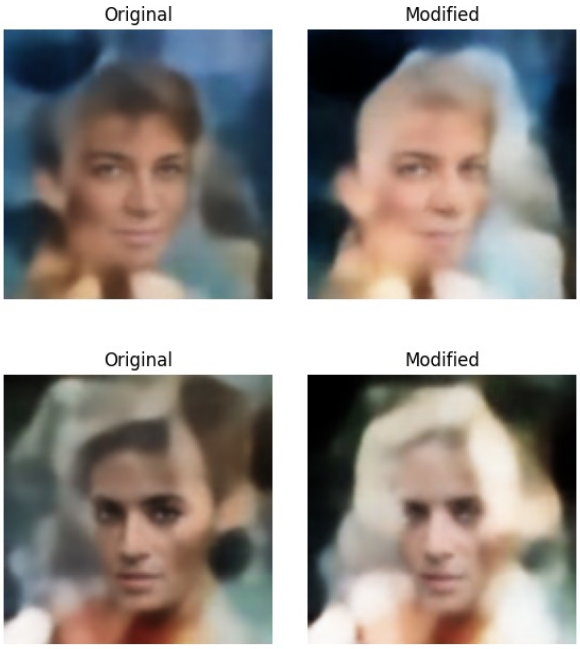


Figure 2. Effect of perturbing the latent vector along a principal direction - hair color

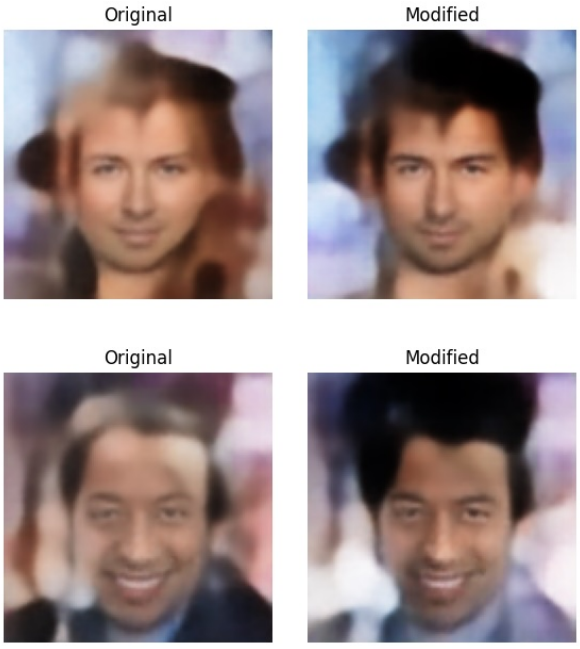


Figure 3. Effect of perturbing the latent vector along a principal direction - hair style

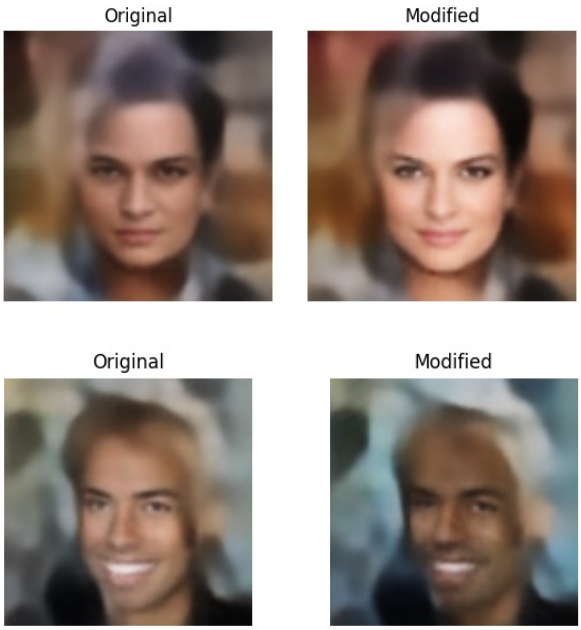


Figure 4. Effect of perturbing the latent vector along a principal direction - skin tone

3.2. DiffuseVAE

In this experiment, we trained a DiffuseVAE model on the CelebA dataset, utilizing approximately 80,000 images for training the VAE component. Due to computational constraints, the DDPM component was trained on a subset of 20,000 images. The models were trained using formulation-

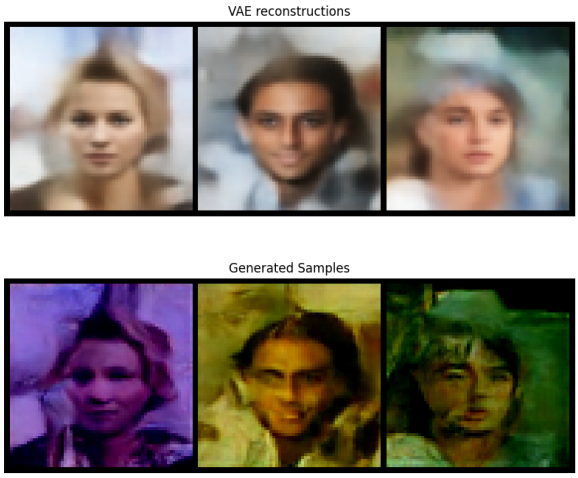


Figure 5. Generated Images using DiffuseVAE

1 [10].
The model is trained in two stages:

1. Train a VAE and freeze its weights
2. Train a DDPM model where input is the reconstructed image concatenated with random noise along the channels (i.e total 6 channels).

Preliminary results indicate that the DDPM model, when conditioned on VAE reconstructions, struggled to learn the denoising process effectively. Notably, the generated samples exhibit color shifts, suggesting that the model is not accurately estimating noise across different RGB channels. However, note that the blurriness around the images is not there in DiffuseVAE output.

4. Preliminary Approach/Idea

4.1. Datasets

We have used MNIST and Celeba dataset[7] for our experiments. We used a smaller subset of the Celeba Dataset(about 80K images) for efficient training.

4.2. Motivation

VAEs are able to reproduce the low-frequency structures in images. But they struggle with representing high-frequency details in their latent space, resulting in blurry outputs. The following approaches leverage the denoising of DDPMs to sharpen the images produced by VAE.

4.3. NoisyVAE + Partial Denoising

In this approach, the forward process and most of the reverse process of a DDPM is replaced by a VAE that generates noisy images for a time T' in the diffusion process. Only partial denoising is performed to reverse the T' time steps.

As the aim is to sharpen the image quality rather than change the structure of the image, it may not be necessary to run the entire forward and reverse diffusion process, as is the case with DiffuseVAE. This approach is as follows:

1. Train a DDPM on the original data.
2. To the original images, add the total noise up to time T' in the forward diffusion process.
3. Train a VAE with reconstruction loss between original data and data with noise at time T'
4. For inference, sample from the decoder of VAE. Then denoise for the final T' steps using DDPM.

The VAE learns to reconstruct low-frequency features. The partial denoising allows for sharpening of the high-frequency details. Further, this can be used for style transfer from the DDPM to VAE's reconstructions.

In this example the VAE has learnt the structure of images and the DDPM has learnt to produce digits with a bold style. Combining the two with this approach provides controllability of the structure using latent space of VAE. It also enables us to transfer style from another dataset by using a DDPM trained on it.

Figure 6. Images generated by DDPM, having bold digits

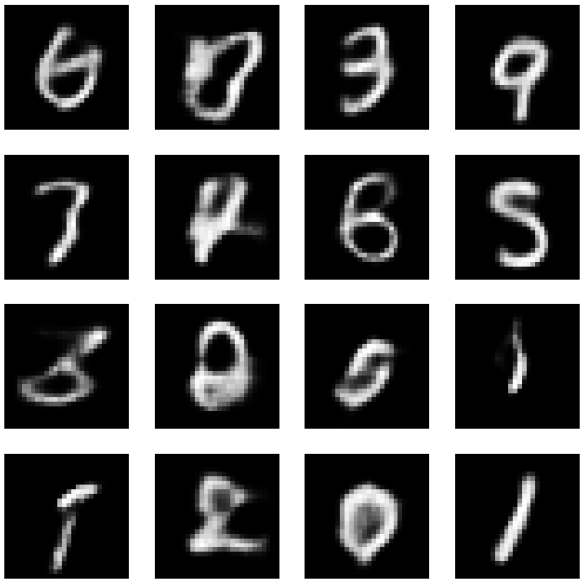


Figure 7. Images generated by Noisy-VAE

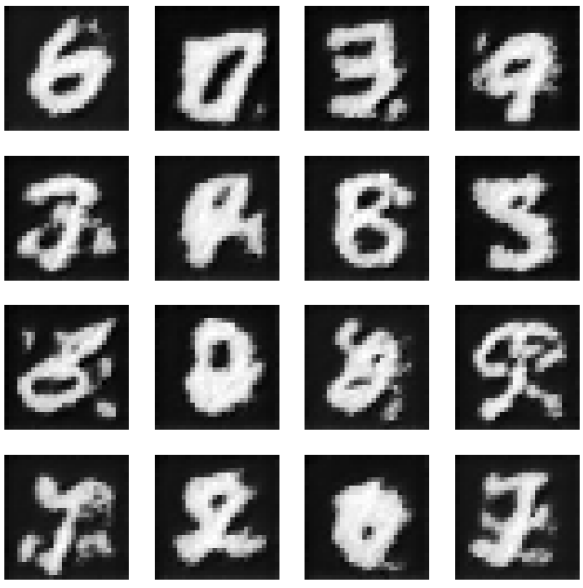


Figure 8. Noisy-VAE outputs after partial denoising

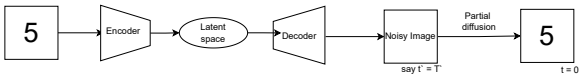


Figure 9. Noisy VAE + Partial Denoising-inference

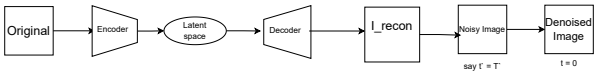


Figure 10. VAE + Noise + Partial Denoising-inference

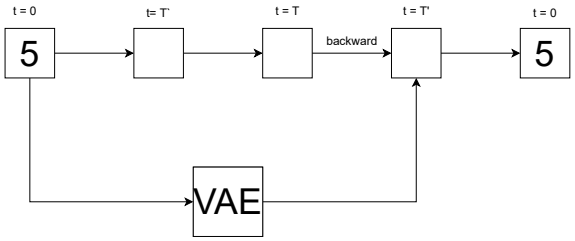


Figure 11. Noisy VAE + Partial Denoising-training

4.4. VAE + Noise + Partial Denoising

This approach breaks down the previous approach so that the VAE is independent of the diffusion process.

1. Train a VAE on the original data
2. Train a DDPM on the original data
3. For inference, sample from the VAE.
4. Add the total noise up to time T' in the forward diffusion process.
5. Denoise for the final T' steps using DDPM.

This approach can also be viewed as simplifying the complete diffusion process in DiffuseVAE to partial diffusion for T' timesteps.

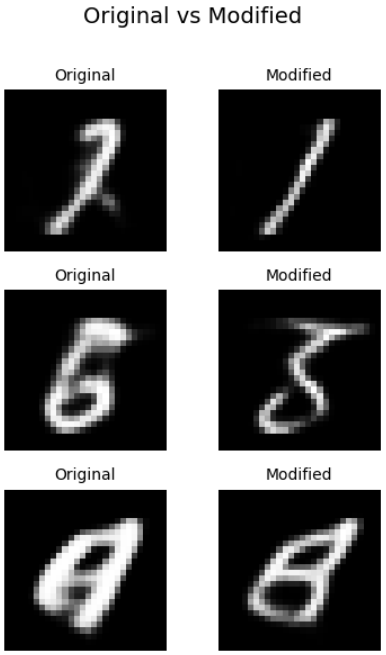
Note that adding noise to the output of a VAE may seem contradictory to the aim of high-quality image generation. An intuitive explanation is that we undo the high-frequency information in the images generated by VAE before letting the DDPM redo it with higher quality or a different style.

5. Contrastive Learning for Latent Vectors

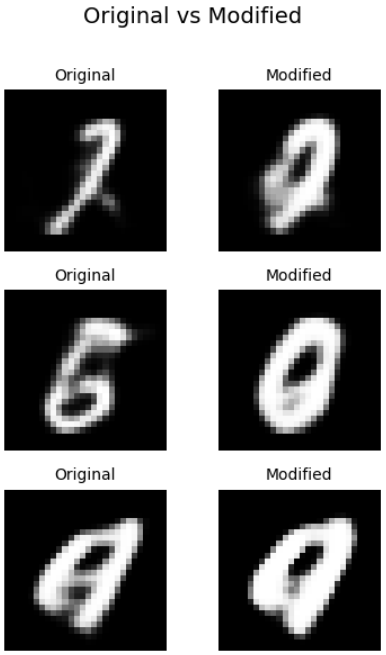
Latent Space Manipulation via PCA Directions

To investigate the structure of the learned latent space and its semantic alignment, we performed controlled manipulations on latent vectors. Specifically, we sampled a latent vector z from the training set and applied small perturbations along one of the principal directions obtained by fitting a PCA model on the entire latent space.

This technique aims to interpret how variations along these directions impact the reconstructed images. In our experiment, we attempted two modifications: one in the direction expected to cause font thinning, and another aimed at font bolding.



(a) Font Thinning Attempt



(b) Font Bolding Attempt

Figure 12. Effect of perturbations in latent space along principal PCA directions.

While the modifications sometimes induced the desired visual change (thinner or bolder strokes), we observed an unwanted side effect — the overall digit shape often

changed significantly, leading to distortion or transformation into a different digit. This suggests that variations along principal components of the latent space do not always correspond to clean, disentangled factors of variation such as stroke width.

This limitation arises due to the fact that standard VAEs do not enforce disentanglement of features in the latent space. Therefore, principal components may combine multiple entangled factors (e.g., thickness and shape), leading to unpredictable changes upon manipulation.

In contrast, contrastive learning-based approaches are known to induce more semantically meaningful and disentangled latent spaces, making such targeted modifications more effective and interpretable. Nonetheless, this experiment demonstrates the expressive potential of the latent space and highlights the need for stronger structure in unsupervised representations.

5.1. cGAN

A Conditional Generative Adversarial Network (cGAN) [8] is an extension of the standard Generative Adversarial Network (GAN), which introduces conditional variables to both the generator and the discriminator. While traditional GANs generate data solely from a noise vector, cGANs incorporate auxiliary information, such as class labels, allowing for controlled data generation. This conditional input enables the model to generate data that conforms to specific categories, making cGANs especially useful for tasks that require fine-grained control over the generated outputs.

A cGAN was trained on the MNIST dataset. Both the generator and discriminator were conditioned on digit labels to guide the generation of class-specific images.

Figure 13. cGAN generated images

5.2. ContraVAE

ContraGAN [3] enhances conditional GANs by adding contrastive loss, improving the quality of generated samples. However, GANs lack the ability to encode a given image into a latent representation, which limits their use in representation learning tasks. In contrast, this work focuses on training a Variational Autoencoder (VAE) enhanced with contrastive loss, allowing the model not only to generate but also to encode inputs into a meaningful latent space. This approach aims to improve representation learning by producing more separated and informative latent vectors than those learned by a standard VAE.

We propose a Variational Autoencoder (VAE) combined with contrastive learning. The VAE learns to encode inputs into a compressed latent space and then reconstruct them. During training, it uses two views of the same input and passes both through the VAE. The loss has two parts: the regular VAE loss (reconstruction + KL divergence) and a contrastive loss that brings the two latent representations closer together. The contrastive term is weighted and added to the total loss to guide training.

The reason for adding contrastive loss is to help the model learn a more disentangled latent space. In a standard VAE, the latent representations can become entangled, meaning different factors of variation in the data get mixed together. This can make the learned features less useful or harder to interpret. By using contrastive learning, the model is encouraged to separate out these factors by pulling together representations of similar inputs and pushing apart dissimilar ones. This results in a cleaner, more structured latent space where different dimensions can capture distinct, interpretable aspects of the data.

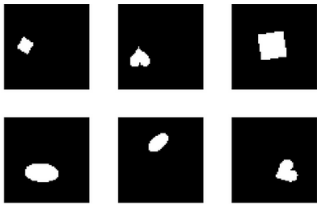


Figure 14. Sample images from dsprites dataset

The dataset used for training is dsprites [15]. dSprites is a dataset of 2D shapes procedurally generated from 6 ground truth independent latent factors. These factors are color, shape, scale, rotation, x and y positions of a sprite. This dataset is chosen since it contains labelled features which helps us experiment and manipulate the latent space.

The images below show that the latent representations learned in ContraVAE are disentangled incomparison to a VAE.

Figure 15. Varying along a dimension in ContraVAE changes the position of the shape

Figure 16. Varying along a dimension in ContraVAE changes the size of the shape

Figure 17. Varying along a dimension in a normal VAE changes the shape (oval to heart) and position showing entangled representation

5.3. Other Experiments: Feat-GAN

Feat-GAN introduces contrastive learning into the GAN training loop to enhance diversity and structure in the generated outputs, specifically in an unconditional setting. After generating images, the model passes them through a fixed encoder to extract feature embeddings. A contrastive loss is computed on these embeddings to encourage the generator to produce outputs that are diverse and well-separated in feature space. This loss is added to the adversarial generator loss, weighted by a factor lambda, allowing fine control over the influence of the contrastive component.

A key difference from ContraGAN is that Feat-GAN does not rely on class labels. While ContraGAN applies class-conditional contrastive learning, Feat-GAN uses contrastive loss purely at the feature level, making it label-free during generator training. However, if a small portion of the dataset is labeled, it can be used to train the encoder (such as an MNIST classifier) in a supervised manner. Once trained, this encoder can be fixed and reused during Feat-GAN training on a larger, unlabeled dataset. This approach allows Feat-GAN to benefit from feature-aware learning even when labeled data is limited, making it especially useful in low-label or semi-supervised settings.

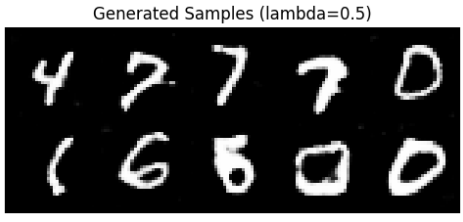


Figure 18. Images generated in feature-based unconditional ContraGAN

6. Modifications to VAE for Higher Quality Images

6.1. Investigating Frequency Domain based VAE

VAE can capture structural details but often leads to blurry output. Isolating the structural information and sharp details could potentially allow a model to train on both in a balanced manner.

Discrete Fourier Transform produces the spectrum of an image. It provides the amplitudes of sinusoidal components of different frequencies. Low frequencies correspond to structural and smooth components. They take up less area on the spectrum. High frequencies encode sharp edges and textures. They take up a greater area on the spectrum.

It is observed that blurring image erases a large portion of the spectrum. High frequency details take up a majority of the information. A model would need higher capacity for these compared to low frequency regions.

Further, small operations such as a 3x3 blur or a shift by 1 element led to great distortions after applying the inverse transform. As a generative model like VAE will produce some errors, this domain would not be feasible for operation. This was confirmed by training and testing a VAE.

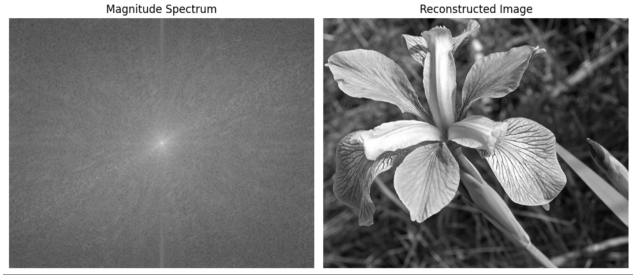


Figure 19. Discrete Fourier Transform

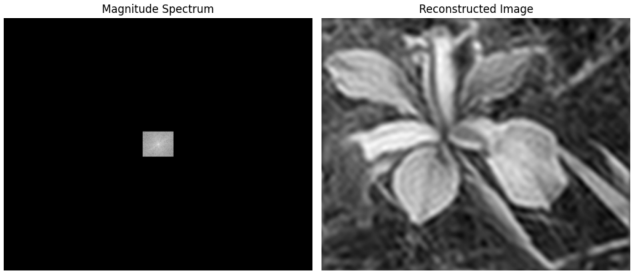


Figure 20. Discrete Fourier Transform - Low Frequencies

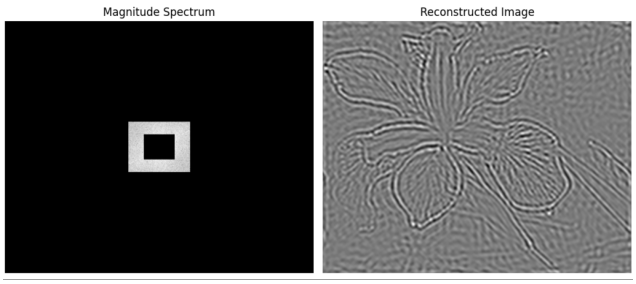


Figure 21. Discrete Fourier Transform - Middle Frequencies

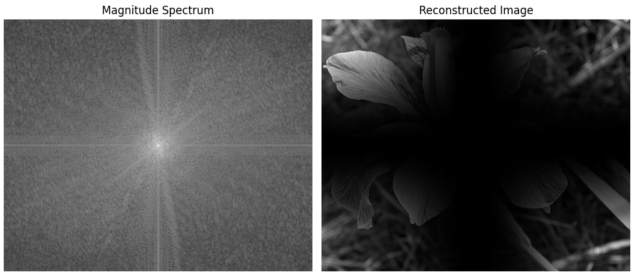


Figure 22. Inverse DFT after 3x3 blur

6.2. Multi Level VAE

Due to the blurry nature of VAE outputs, there is a difference between original image and generated image. This architecture aims to model this difference or residual using another level of latent vectors.

The model for the second layer encodes the residual into a latent vector. The decoder operates on patches from the output of first VAE along with the latent vector of the second.

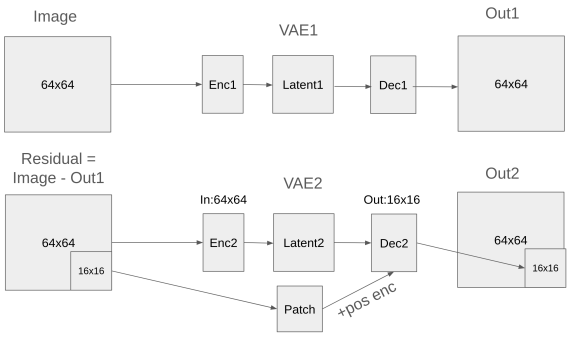


Figure 23. Multi Level VAE Architecture

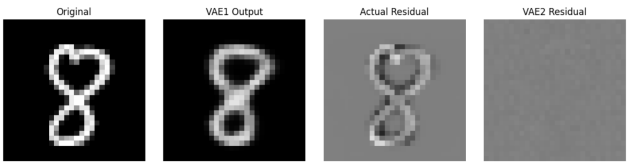


Figure 24. Multi Level VAE - Vanishing Values in Training

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc.

[3] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation, 2021.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[5] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[6] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2016.

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

[8] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[9] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[10] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents, 2022.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[14] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, page 3483–3491, Cambridge, MA, USA, 2015. MIT Press.

[15] TensorFlow Datasets. dsprites dataset. `https://www.tensorflow.org/datasets/catalog/dsprites`, 2024. Accessed: 2025-05-01.

[16] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.