Investigating Latent Space Representation of Text-to-Image VAE with PCA, and Kohonen Feature Maps

Advaith Ravishankar

April 2, 2025

Abstract

The field of generative models has undergone a paradigm shift with the advent of text-to-image generation, spearheaded by Diffusion Models, which set the state-of-the-art in translating between text and visual modalities Cao et al. [2024]. Earlier approaches, including Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), provided consistent results and established foundational principles Rombach et al. [2022]. VAEs, in particular, employ a latent space representation constrained by a Gaussian distribution, achieved through the reparameterization of the mean and log-variance in the encoder Kingma and Welling [2013]. Despite these advancements, the mechanisms through which these models encode and represent information within their latent spaces remain underexplored. This study investigates the latent space representations of VAEs in text-to-image generation using two techniques: Principal Component Analysis (PCA), and Kohonen Feature Maps (Self-Organizing Maps). By analyzing the visualization of latent spaces by the category of the image and text, this research seeks to uncover the representation of multi modalities in the same latent space.

source code: `https://github.com/AdvaithRavishankar/Text_To_Image_VAE`

1 Introduction

The rapid advancements in text-to-image generation have revolutionized the field of generative models, with Diffusion Models emerging as the state-of-the-art approach for translating between two information modalities Cao et al. [2024]. These models have demonstrated remarkable capabilities in generating high-quality images from textual descriptions, pushing the boundaries of multi-modal machine learning.

Earlier approaches to text-to-image generation, such as Variational Autoencoders

(VAEs) and Generative Adversarial Networks (GANs), also yielded consistent results Rombach et al. [2022]. In particular, VAEs are known for their latent space representation, which is constrained by a Gaussian distribution. This is achieved through the reparameterization of the mean (μ) and log-variance ($\log \sigma^2$) projection heads of the encoder Kingma and Welling [2013]. These foundational techniques laid the groundwork for understanding and leveraging latent space representations in generative models.

Despite the success of these models, a critical question arises: How do they represent knowledge in their latent space? While all these models utilize encoder-decoder structures to generate outputs, the precise mechanisms by which they map image and text encodings within the latent space are not fully understood. This study aims to demystify these latent space representations by analyzing the latent space of VAEs in text-to-image generation through two visualization methods: Principal Component Analysis (PCA), and Kohonen Feature Maps (Self-Organizing Maps).

The objective of this investigation is to gain deeper insights into how information is distributed within the latent space, thereby enhancing our understanding of the inner workings of generative AI models. Such insights could inform the development of more efficient and interpretable generative models in the future.

2 Dataset

MS COCO 2017 (Common Objects in Context) Lin et al. [2014] is used for the modeling of this task. This dataset provides a large-scale, richly annotated collection of images featuring everyday objects in diverse contexts. For this task, the study focuses on the images and their associated captions to obtain image-text-category triplets. The text is a couple sentences describing the image as seen in Figure 1 and the category uses the supercategory in the COCO's annotations.

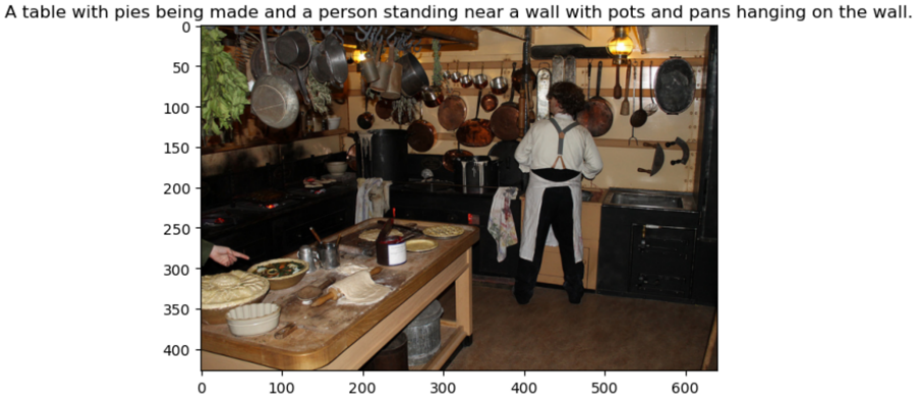
A table with pies being made and a person standing near a wall with pots and pans hanging on the wall.
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

Figure 1: Dataset Example

Due to limited storage constraints, the study also works with only 5k images (val 2017 set), their corresponding texts and categories. This will not be sufficient in creating strong VAE reconstructions but it will be sufficient for the model to converge during training, allowing us to analyze a well constructed latent space.

3 Method

3.1 Overview

The method is broken down to two parts - the Modeling and the Latent Space Analysis. In the first section, the study goes over the model design and performance and then it is followed by the analysis of the latent space using PCA, and Kohonen Feature map

3.2 Model

3.2.1 Design

The Variational Autoencoder was implemented in pytorch. The design was guided by AntixK [2020]. The design was adjusted to work for text-to-image generation by having 3 different components:

Image Encoder: 5 convolution blocks of [Conv2d, Batch Norm, Leaky Relu] with hidden dimensions [8, 12, 16, 24, 32]. Kernel size was set to 3, stride to 2 and padding to 1. The end of the model was 2 linear layers (independent of each other) projecting onto the latent dimension of 64. (representing mu and log var).

Image Decoder: a linear layer projecting latent dim 64 to the same output size of the convolution blocks in the Image Encoder. The reverse of the Image Encoder conv block is used by replacing [Conv2d, Batch Norm, Leaky

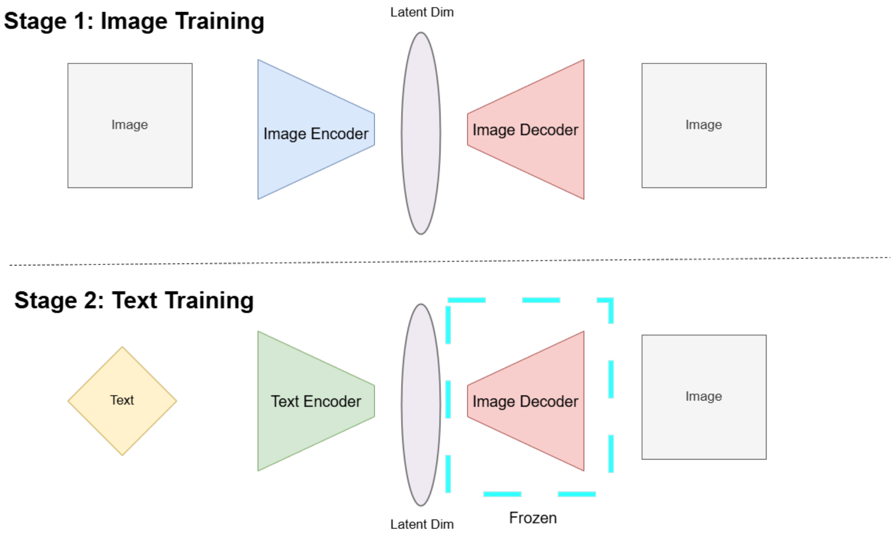

Figure 2: VAE Pipeline

Relu] with [ConTranpose2d, Batch Norm, leaky Relu]. The ouptput is the same dimensions of the image.

Text Encoder: The sentence was tokenized using BERT Devlin et al. [2018] followed by a multi attention head (num_heads=3) and MLP of linear, relu, linear to project onto the same latent dimension of 64.

Using these three components, the study follows the following training and inference pipeline.

3.3 Pipeline

Figure 2 has the visual breakdown of each training stage.

Stage 1 - Image Training: First, the VAE is set the Image encoder and decoder. The entire image dataset is trained on a reconstruction task using KL Divergence and MSE on the reconstructed image in pixel space. KL is comparing the distribution of what is generated by the encoder with a Gaussian to obtain a metric to constrain the latent space to be sampleable.

$$\mathcal{L}(x, \hat{x}, \mu, \sigma^2) = \mathrm{MSE}(x, \hat{x}) + \mathrm{KL}(\mu, \sigma^2) \tag{1}$$

$$\mathrm{KL}(\mu, \sigma^2) = \frac{1}{N} \sum_{i=1}^{N} -\frac{1}{2} \sum_{j=1}^{d} \left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right), \tag{2}$$

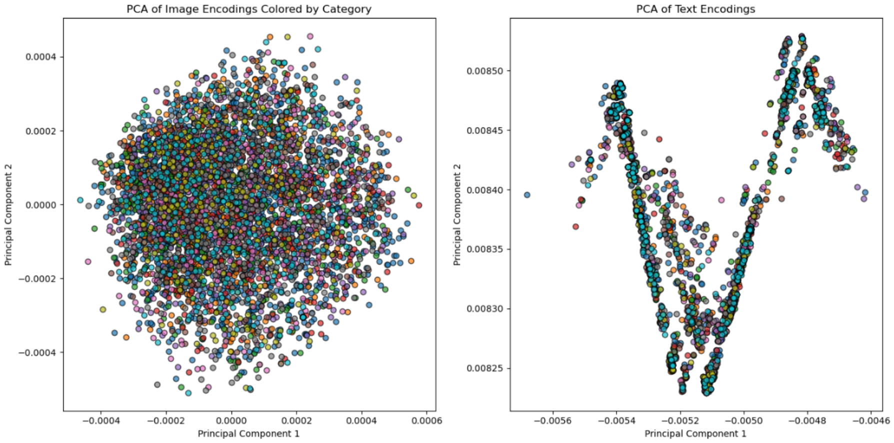

Figure 3: PCA Visualization

$$\mathrm{MSE}(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{d} \sum_{j=1}^{d} \left(x_{i,j} - \hat{x}_{i,j}\right)^2, \tag{3}$$

Stage 2 - Text Training: the study then replaces Image encoder with the Text encoder and freezes the Image decoder. This is done to map the text representation onto the trained image latent space.

Inference: Once training converges, the inference pipeline will be text –> Text Encoder –> latent space –> Image Decoder –> Generated Image.

3.4 Latent Space Analysis

Note: Stage 1 training was trained for 10 epochs and plateaued with reconstruction loss of 0.073. Stage 2 failed to train for more than 1 epoch due to consistent crashing/kernel restarts on data hub.

3.4.1 PCA

Overview: For the PCA visualization, the study takes samples of the image-text-category triplet to ensure the first 10 supercategories are taken (for clear visualizations) and computes the image and text latent space representations independently. Next, PCA is computed separately on both image latents and text latents. The top 2 principal components are then visualized and are colored based on the category. Figure 3 showcases the plot for both image and text PCAs.

Analysis: The Image PCA has no clustering of categories. This is because

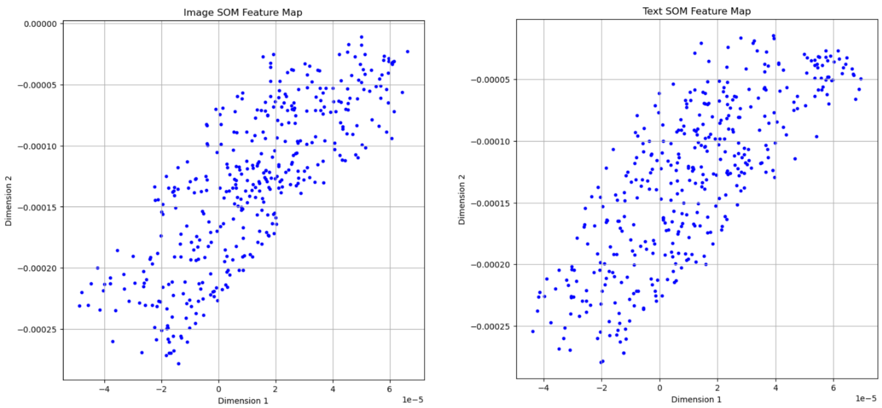

Figure 4: Kohonen Map (Self Organizing Map) Visualization

the number of samples in the data (5k) is insufficient in capturing structural similarity. A strength however is that the values are projected like a circle with dense points at the 0 and the distribution peters off as we move away from the mean. This showcases the VAE's ability to project data to a Gaussian Distribution (data is distributed in 2D Gaussian Space with PCA), showcasing that event with low samples, the KL Divergence impacts learning.

For text PCA, the study expects it to be mapped to its corresponding image encoding. However, it is not projected onto the same circular representation of the Image PCA. This is due to the training of Stage 2 crashing with kernel restarts, making the model unable to learn the mapping.

3.4.2 Kohonen Feature Map

Overview: For the Kohonen map visualization, the study uses the same dataset as PCA and initializes a self-organizing map (SOM) with a two-dimensional grid structure and trains it on high-dimensional input data. The implementation was influenced by Sarkar [2018]. Figure 4 showcases the plot for both image and text PCAs.

Analysis: The Image SOM map shows a uniform spread of feature points. This lack of distinct clusters indicates that the SOM is capturing broad structural relationships but does not sufficiently differentiate between specific categories within the data. Hence, this is unable to capture any notion of categories.

Similar to the Image SOM, the Text SOM map does not exhibit clear clustering. The lack of correspondence between the text and image mappings (as was the expectation for PCA) suggests that the text embeddings and image embeddings have diverged significantly during training.

4 Discussion

4.1 Overall

One of the strengths of the PCA visualization is its ability to highlight the VAE's success in projecting data to a Gaussian distribution, as evidenced by the circular distribution of the image PCA plot. This circular pattern, with dense points near the center and a gradual tapering outward, reflects the impact of KL Divergence in shaping the latent space even with a relatively small sample size (5k). This suggests that the VAE effectively captures global structural properties despite data limitations.

However, the PCA analysis also reveals significant limitations. The image PCA lacks clustering of categories, indicating insufficient structural similarity in the latent space due to the low number of samples. Additionally, the text PCA fails to align with the image PCA's Gaussian-like structure. This misalignment is attributed to the training instability in Stage 2, where kernel restarts disrupted the model's ability to learn coherent mappings between text and image representations.

For the Kohonen Feature Maps, the SOM successfully preserves broad structural relationships within the data, as evidenced by the uniform spread of feature points in both the image and text maps. However, the lack of distinct clustering indicates that the SOM fails to capture category-level differentiation.

4.2 Limitations

One of the largest constraints was runtime. If the model took more than 5-6 hours on datahub, the kernel would restart automatically making it difficult to process all data points. This was specially true for the text encoder training which crashed after only 1 epoch of training.

Another issue was data storage. Due to Datahub's limits, the study was unable to use a large dataset of COCO and could only achieve minimal results on training. Due to this limitation, the PCA and Kohenan feature maps failed to show any clustering.

5 Conclusion

In this study, the latent space representations of Variational Autoencoders (VAEs) in text-to-image generation were analyzed using Principal Component Analysis (PCA) and Kohonen Feature Maps (Self-Organizing Maps). PCA effectively highlighted the VAE's ability to project data into a Gaussian distribution, with the circular pattern of the image PCA plot reflecting the impact of KL Divergence in shaping the latent space, even with a limited dataset. However, both the image and text PCA visualizations failed to show clustering of categories, largely due to the insufficient number of samples and training instabilities caused by kernel restarts. Similarly, Kohonen Feature Maps preserved broad structural relationships in the data but lacked category-level differentiation, as evidenced by the uniform spread of feature points without distinct clusters. Runtime and data storage constraints further limited the scope of this analysis, as the inability to process larger datasets, such as COCO, restricted the study to minimal results. Despite these challenges, the visualizations provided meaningful insights into the latent space structure and highlighted areas for further improvement in aligning text and image representations.

References

AntixK. Pytorch-vae.
urlhttps://github.com/AntixK/PyTorch-VAE, 2020.

Pu Cao, Feng Zhou, Qing Song, and Lu Yang. Controllable generation with text-to-image diffusion models: A survey. *arXiv preprint arXiv:2403.04279*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll
'ar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *European conference on computer vision*, pages 740–755, 2014.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2207.03332*, 2022.

Eklavya Sarkar. Artificial neural networks: Kohonen self-organising maps, 2018. URL `https://eklavyafcb.github.io/som.html`.