COMPARISON OF LATENT SPACE INTERPRETABILITY

Alex Iliarski, Isaac Berlin, Joshua Krueger, Tianzhe Han
Department of Computer Science & Engineering
University of Minnesota
Minneapolis, MN 55414, USA
{iliar004, berli113, krueg709, han00510}@umn.edu

ABSTRACT

We investigate the semantic interpretability of latent spaces for several generative models including StyleGAN2, Variational Autoencoders (VAEs), β-VAEs, and β-TCVAEs. Using the CelebA and CelebA-HQ datasets, we explore semantic manipulation in latent spaces by training linear SVM classifiers and analyzing their effect on generated outputs. We compare results both quantitatively and qualitatively, and discuss which models yield more interpretable latent representations.

1 PROBLEM STATEMENT

This project investigates the interpretability of latent spaces in generative deep learning models. Specifically, we study and compare different model architectures, such as StyleGAN (Karras et al., 2019), StyleGAN2 (Karras et al., 2020b), Variational Autoencoders (VAEs) (Kingma & Welling, 2022), β-VAEs (Higgins et al., 2017; Burgess et al., 2018), and β-TCVAEs (Chen et al., 2018). In particular, we examine how these different generative models structure their latent representations and the degree to which these latent representations demonstrate interpretable or semantically meaningful information.

In this context, interpretability is defined as the extent to which individual dimensions or directions in the latent space correspond to an understandable or disentangled feature of the generated output, such as facial expressions, accessories, object orientation, color, etc. Understanding the structure of latent spaces is a critical problem for generative models due to its implications for AI explainability and controllability.

We use the CelebA dataset (Liu et al., 2015), which contains a large collection of face images and has been widely used in previous work in the area of generative models and latent space analysis. Thus, by training different model architectures to generate faces and manipulating their latent spaces, we aim to investigate how the structure of these spaces relates to interpretable features. Additionally, we seek to understand which model architectures provide the most interpretability of their latent space, and possibly provide explanations for these differences.

2 MOTIVATION

Acting as a compressed summary of the dataset, latent space can reveal the important features and patterns considered by a model, helping us dive into the model's thought process. Interpretable latent spaces allow us to modify specific attributes of generated outputs. Adjusting the latent vector in a certain direction leads to predictable and meaningful modifications in the output, like we can control smile intensity, head pose, hair style or color by manipulating specific latent dimensions. And different from conditioning on a label or a class, we can gradually vary features along continuous axes, realizing fine-grained control. Also, if the latent dimensions can be mapped to real-world concepts, like "smiling" in a face generator or "tense" in language, these factors can be manipulated in intuitive ways. We can deal with dimensions tied to concepts we understand to guide generation outcomes instead of handling raw vectors or opaque model behavior. This benefits human users to reason about and interact with deep generative models in a way that aligns with our understanding of the world, bridging the gap between black-box models and human reasoning. Besides, specifying the latent space can help align different data types, like images and texts, which usually involve

embedding each modality into a shared latent space. Making the latent space interpretable can help us understand why and how different data types are matched, facilitating applications like image captioning, text-to-image generation, translation across modalities, etc. Further, interpretable latent spaces can help identify and modify biased features. If a dimension in latent space strongly correlates with some sensitive attributes, we can adjust it to mitigate bias.

3 PREVIOUS WORK

Understanding and interpreting latent spaces is an active area of research in generative deep learning (Asperti & Tonelli, 2023). Models like the β-VAE (Higgins et al., 2017) and Factor-VAE (Kim & Mnih, 2018) adjust their architectures to make the structure of their latent spaces more interpretable. Alongside architectural changes, metrics such as the Mutual Information Gap (MIG) (Chen et al., 2018) have been introduced to help quantify the degree of disentanglement between different latent factors.

Beyond interpretability, a growing body of work has explored how latent vectors can be manipulated to produce targeted changes in generated images. Parihar et al. (2022) and Shen et al. (2020) demonstrated that transformations within the latent space can effectively encode high-level attributes such as age, hairstyle, and facial accessories. Both proposed frameworks, FLAME (*Few-shot Latent-based Attribute Manipulation and Editing*) and InterFaceGAN (*Interpret the Face representation learned by GANs*), are evaluated using StyleGANs with impressive results. Similarly, Goetschalckx et al. (2019) used transformations within the latent space of a GAN to scale semantic attributes like memorability or aesthetics.

Although the majority of work in this area has focused on Generative Adversarial Networks, there are still resources and studies exploring latent space manipulation in other architectures. Additionally, Li et al. (2018) proposed a attribute-disentangled model as a combination of VAEs and GANs to accomplish our same goal. Ślot et al. (2021) used a Convolutional Autoencoder to transform semantic attributes on datasets of both human faces and handwritten digits.

4 OBJECTIVES

The goal of this project is to evaluate and compare the interpretability of latent spaces across several generative models, including StyleGAN2, VAE, β-VAE, and β-TCVAE. We aim to identify interpretable directions using linear SVMs trained on labeled latent representations, allowing us to assess the structure and disentanglement of each model's latent space.

We compare the models using both quantitative metrics (accuracy, precision, recall, F1-score) and qualitative evaluation of edited outputs. These edits involve traversing latent space along semantic directions and observing resulting image changes. Additionally, we investigate how architectural features—such as KL divergence weighting or StyleGAN's layered latent codes—affect the smoothness and usability of latent representations.

5 METHODOLOGY & RESULTS

5.1 STYLEGAN2

We have found and experimented with a StyleGAN2 (Karras et al., 2020b) pretrained on the CelebA-HQ dataset (Karras et al., 2017). This model was pretrained by NVIDIA (Karras et al., 2021), and is provided from their ngc catalog. We are using the stylegan2-ada-pytorch (Karras et al., 2020a) package from GitHub to facilitate interfacing with the model and its latent space. To extract the latent vector from a StyleGAN2 model, we project an input image back onto the model's latent space. This process begins by sampling a random latent vector and generating an image from it. We then use gradient descent, guided by a perceptual loss function (Johnson et al., 2016), to iteratively optimize the latent vector until the generated image closely matches the target input.

In order to explore the controllable edits we could make using the latent space of StyleGAN2, we focused on the "Eyeglasses" attribute in the CelebA dataset. We selected 1500 images labeled with

eyeglasses and 1500 without. Using the pretrained model at 256x256 resolution, we projected each image into the model's $\mathcal{W}^+$ latent space using gradient descent. The optimization minimizes perceptual loss (LPIPS, or Learned Perceptual Image Patch Similarity) (Johnson et al., 2016), making small adjustments to the latent vector until the generated image closely resembles the original. This process gave us a set of 3000 latent vectors of shape [14, 512], corresponding to the 14-layer latent space of the StyleGAN2 model.

After achieving poor results using the difference of class means to find an "eyeglasses" direction in the latent space, we instead turned to a discriminative method. We averaged the latent vectors across the 14 layers to get a 512-dimensional representation in $\mathcal{W}$ space, and then trained a linear SVM (Support Vector Machine) classifier model to distinguish between the "Eyeglasses" and "No Eyeglasses" classes. This SVM model performed reasonably well, with 80% accuracy on a held-out test set, achieving strong precision and recall. This indicates that the two classes are reasonably separable in the StyleGan2 model's latent space, meaning this approach is solidly motivated. Thus, the normal vector of the SVM's separating hyperplane was extracted, normalized, and used as our semantic direction vector, encoding the most discriminative direction between the two classes.

To evaluate how well this semantic direction works as an editing tool, we sampled random latent vectors from the standard Gaussian prior and ran them through the generator's mapping network to produce latent vectors in $\mathcal{W}^+$. For each sampled latent vector, we added scaled multiples of the SVM direction to the first three layers, controlling the edit intensity with a scalar factor α ranging from 1 to 250. We only applied the change to the first three layers based on experimentation that it provided the best and most consistent results. This aligns with the idea that important localized semantic information in the picture, such as eyeglasses, hats, etc, are controlled primarily in the early layers of the latent space.

We applied this process to 20 randomly generated face images, saving a horizontal image strip showing the original face followed by a progression of movement along the eyeglass edit direction. The results showed that the SVM-based latent direction was effective at inducing eyeglasses in many of the samples. For 5 of the 20 images, the edit had very little desired effect, likely due to the latent being far from the SVM decision boundary or due to entangled features in the generator's latent space. However, in 8 of the 20 samples, the results were very strong, adding glasses onto the image with very little distortion or decoherence. In the remaining 7 images, glasses were eventually produced in the image, but the decoherence of the image is too strong to be considered a successful edit. Examples of these results are demonstrated in Figure 1.

Figure 1: Examples of SVM-based eyeglasses editing. The top row shows successful edits where eyeglasses are added with minimal distortion. The second row shows examples where the edit failed to induce any glasses in the image. The bottom row shows cases where eyeglasses appeared, but the edits introduced visual artifacts or significant decoherence.

5.2 Variational Auto-Encoder

Using a basic variational autoencoder trained on the CelebA dataset, we were able to successfully add sunglasses to an image. The original and sunglasses images are provided below. This approach first involved training the model and then choosing 2000 images with sunglasses and 2000 images without. These images were then used separated using a linear support vector machine classifier. The vector parameterizing the decision boundary of the SVM was then extracted, added to the latent space vector for the image and the result was decoded. As of right now the results, while clearly wearing sunglasses, are very low fidelity. This is likely due to the fact that only a small fraction of the CelebA dataset is trained with sunglasses and we used a relatively small model architecture $\sim$6.3M parameters. To experiment on classes with more representation, we repeated the process for the male and smiling classes. The SVM for these classes was trained on the entire CelebA dataset. The results of editing these vectors for a single face is shown below. Despite the larger representation within the datset and the use of the full amount of data in the support vector classifier, the accuracy on classifying smiling and male was still 75% for both.

Figure 2: Image Editing Gradient for Single Face

(a) Original CelebA images

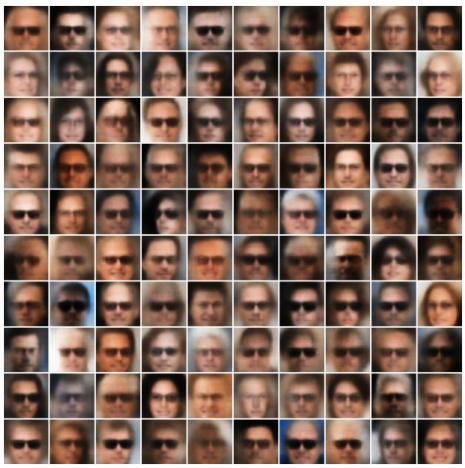
(b) Images with added sunglasses

Figure 3: Images edited by the VAE

5.3 Beta Variational Auto Encoder

We also trained a β-VAE, as introduced by Higgins et al. (2017), on the CelebA dataset. This model builds on the standard VAE by scaling the KL divergence term in the loss function, encouraging more disentangled and discrete latent representations. This structure is particularly useful for latent space editing, as it can produce clearer and more interpretable directions. We trained the β-VAE with around 1.3M parameters for 20 epochs on the full CelebA dataset. After training, we encoded 2,000 images of faces with glasses and 2,000 without to obtain their latent representations. We then trained a linear SVM on these vectors and extracted the direction corresponding to the presence of glasses by taking the normal vector to the SVM decision boundary. Finally, we generated a 10×10 grid of baseline faces and edited them by adding the "glasses" direction to their latent vectors. The results are shown in figure 4.

(a) Original randomly generated faces



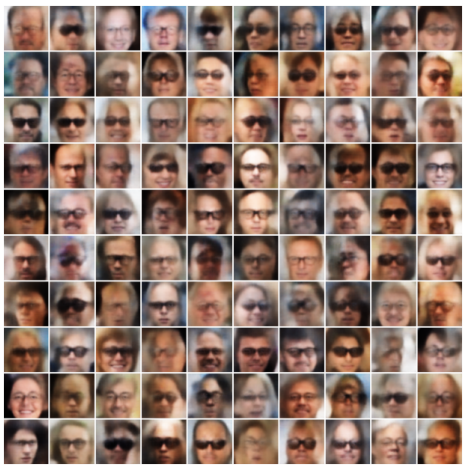
(b) Faces edited by addition of eyeglasses vector

Figure 4: Images generated and edited by the β-VAE.

5.4 Beta Total Correlation Variational Autoencoder

We trained a β-TCVAE model (Chen et al., 2018), which extends the β-VAE by breaking down the KL divergence into three parts: the mutual information between the input and the latent variables, the total correlation (which measures how dependent the latent dimensions are on each other), and the dimension-wise KL divergence. Instead of increasing the weight on the entire KL term like the β-VAE, the β-TCVAE specifically targets the total correlation, encouraging the latent variables to be more independent. We followed a similar training and evaluation strategy to the β-VAE using the same optimizer, number of epochs, and $\sim$1.3M parameters. Our results are available below in figure 5.

(a) Original randomly generated faces



(b) Faces edited by addition of eyeglasses vector

Figure 5: Images generated and edited by the β-TCVAE.

5.5 Overall Comparison

Across all models tested, the β-TCVAE achieved the highest latent space interpretability, as reflected in both quantitative classification metrics and qualitative editing outcomes. It consistently produced the clearest edits when traversing along semantic directions, likely due to its explicit penalization of total correlation, which encourages disentangled and independent latent dimensions.

While the StyleGAN2 model was capable of producing high-fidelity images, its latent space proved more challenging to manipulate. Although linear SVMs could identify meaningful directions, editing in $\mathcal{W}^+$ space often led to visual artifacts or incoherent results, especially when moving far from the decision boundary. This suggests that StyleGAN2's latent space may contain irregular or poorly populated regions that hinder smooth interpolation.

Overall, VAEs—particularly β-TCVAE—demonstrated smoother, more structured latent spaces that better support semantic control. Meanwhile, StyleGAN2's higher visual realism comes at the cost of reduced editability and latent consistency. Additionally, use of more training data for developing the SVM models could potentially produce more robust results, especially in the sparse latent space of the StyleGAN2.

Table 1: Model latent space SVM comparison

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
StyleGAN2	0.8000	0.7895	0.8108	0.8000
VAE	0.7611	0.7680	0.7484	0.7581
β-VAE	0.8387	0.7984	0.8543	0.8254
β-TCVAE	**0.8750**	**0.8516**	**0.8997**	**0.8750**

6 CONCLUSION

Our technique of determining feature vectors in the latent space without positive / negative feature pairings was very successful. The SVM approach allowed us to find feature vectors using the labeled data already present in the data set and did not require manual use of tools such as Photoshop. We found that the β-TCVAE contains the most interpretable latent space, which makes sense due to the design of the model architecture.

Another interesting finding is that the decoherence present in the StyleGAN2 strongly suggests that the latent space for this model is much less smooth than that of the variational autoencoders. Editing in certain directions usually caused the model to generate images exhibiting strong decoherence, whereas with the autoencoders, while images may not have been high-fidelity, they were generally all recognizable as relatively normal faces. The editing was also much more likely to have an effect compared to StyleGAN. From a mathematical perspective, this makes sense. Because the variational autoencoder encodes images as a distribution, and then samples from the distribution during decoding, the result is a latent space far more likely to be continuous.

What was surprising is the ability of the SVM to find a strong linear boundary within the StyleGAN latent space despite the decoherence indicating that sunglasses cannot be trivially added by a simple linear change in the latent space. The problem is less likely linearity, then, and more likely that the latent space is full of 'holes' i.e. regions where the latent space does not correspond to any image that would be reasonable to expect from the data. Variational Autoencoders avoid this problem due to the random sampling of the latent space, smoothing over any holes present. A possible solution to this issue, which could be explored in future work, would be to add a small amount of noise to the W^+ latent space vector during training. Additionally, more images within a specific region may also contribute to smoothing the desired latent region.

A CODE

Our code for all the models and latent space exploration are available on our GitHub.

REFERENCES

Andrea Asperti and Valerio Tonelli. Comparing the latent space of generative models. *Neural Computing and Applications*, 35(4):3155–3172, 2023.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018. URL https://arxiv.org/abs/1804.03599.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties, 2019. URL https://arxiv.org/abs/1906.10112.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 694–711. Springer, 2016.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. URL https://arxiv.org/abs/1812.04948.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020b. URL https://arxiv.org/abs/1912.04958.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pp. 2649–2658. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Defang Li, Min Zhang, Weifu Chen, and Guocan Feng. Facial attribute editing by latent space adversarial variational autoencoders. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1337–1342, 2018. doi: 10.1109/ICPR.2018.8545633.

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)*, December 2015.

Rishubh Parihar, Ankit Dhiman, Tejan Karmali, and Venkatesh R. Everything is there in latent space: Attribute editing and attribute style manipulation by stylegan latent space exploration. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 1828–1836. ACM, October 2022. doi: 10.1145/3503161.3547972. URL http://dx.doi.org/10.1145/3503161.3547972.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans, 2020. URL https://arxiv.org/abs/2005.09635.

Krzysztof Ślot, Paweł Kapusta, and Jacek Kucharski. Autoencoder-based image processing framework for object appearance modifications. *Neural Computing and Applications*, 33:1079–1090, 2021.