Latent Trajectory Optimization for Smooth Semantic Edits

Bhagyesh Kumar
Manipal Institute of Technology, Manipal, India
bhagyesh.mitmpl2024@learner.manipal.edu

Abstract

We explore a framework for semantic image editing that learns smooth, interpretable transformations in the latent space of a generative adversarial network (GAN). Unlike prior approaches such as StyleCLIP or InterFaceGAN, our method does not depend on any pretrained encoders (e.g., CLIP). Instead, it relies solely on the generator and discriminator of a GAN to optimize trajectories in latent space. A dedicated trajectory network learns to map an initial latent vector to a sequence of latent codes encoding a smooth semantic change, enabling natural and identity-preserving edits without additional supervision. Code can be accessed at https://github.com/invi-bhagyesh/lateGAN/tree/main

I. Introduction

Latent space manipulation has become a key tool for image editing using GANs. Traditional methods like InterFaceGAN or StyleCLIP typically apply fixed directional shifts in latent space to achieve edits such as age progression or expression change. However, these fixed directions often lead to abrupt or unrealistic transitions, and heavily depend on large pretrained encoders.

In this work, we explore an alternative: representing semantic transformations as smooth trajectories in the latent space. By optimizing a neural trajectory generator, we can produce interpretable paths from an initial latent vector that gradually transform the output image while preserving identity. This approach bypasses the need for external supervision or pretraining, enabling a self-contained and plug-and-play editing system.

II. Related Work

StyleGAN has enabled high-quality image generation, and latent space editing techniques such as InterFaceGAN, GANSpace, and StyleCLIP have attempted to control semantics by shifting in latent directions. However, most of these rely on external embeddings or linear approximations.

Trajectory-based editing offers a nonlinear alternative, better suited for gradual transformations. Our approach builds on this idea, enhancing edit quality and generalization by training a latent trajectory network end-to-end.

III. Methodology

A. Backbone GAN

We use a GAN architecture based on Deep Convolutional GAN (DCGAN), a well-established framework for stable image synthesis. The generator $G(z)$ maps a latent vector $z \in R^d$ to an image in pixel space, while the discriminator $D(x)$ attempts to distinguish between real and generated images, providing adversarial feedback to guide the generator's learning.

The generator is composed of transposed convolutional layers (also known as deconvolutions), batch normalization, and ReLU activations, progressively upsampling the latent vector into a realistic image. Conversely, the discriminator uses a series of convolutional layers with LeakyReLU activations and batch normalization to classify images as real or fake.

To train this architecture, we use the CelebA dataset—a large-scale face attributes dataset with over 200,000 celebrity face images. CelebA provides rich diversity in pose, background, and facial attributes, making it ideal for training generative models that aim to capture complex facial semantics. Images are cropped and resized to 64×64, which is a standard resolution used in DCGAN experiments to balance visual detail and training stability.

The resulting DCGAN learns to generate high-quality face images that capture prominent features such as gender, hairstyle, and expression. This trained generator serves as the foundation for evaluating the latent trajectories learned by TrajectoryNet.

B. Latent Trajectory Generator

To generate a sequence of semantically meaningful latent vectors, we introduce a simple yet effective neural network called **TrajectoryNet**. Instead of relying on recurrent or attention-based mechanisms, TrajectoryNet is implemented as a fully connected feedforward network that maps an initial latent code $z_0 \in R^d$ to a sequence $\{z_0, z_1, \ldots, z_n\} \in R^{n \times d}$, where each z_i represents a point in the generator's latent space.

The network consists of a multi-layer perceptron (MLP) that projects the input vector into a larger vector of size $n \times d$, which is then reshaped into a trajectory of latent codes. This design

allows the model to learn fixed-length semantic transitions, such as facial transformations, identity morphing, or style drift, without relying on complex sequential dependencies.

Compared to standard interpolation techniques, this learned latent trajectory is more expressive and data-driven. When passed through a fixed generator (e.g., DCGAN), the trajectory produces a sequence of images showing smooth, coherent changes across various attributes. Notably, this method captures non-linear changes in the latent space while maintaining computational simplicity and interpretability.

C. Loss Functions

- **Adversarial loss:** Each $G(z_i)$ must be realistic, enforced by D.
- **Smoothness loss:** Encourage consistent transitions by minimizing $\sum ||z_{i+1} - z_i||^2$.
- **Optional attribute loss:** When attribute labels are available (e.g., "smile"), use contrastive or classifier-based loss to guide trajectories.

IV. Experiments and Results

A. Generated Output after 50 epochs

Fig. 1: Base image generated from initial latent vector using trained generator.

B. Trajectory-based Latent Output

Fig. 2: Sequence of images generated from latent trajectory showing smooth semantic change.

The output depicts a sequence of images generated by mapping a continuous trajectory in latent space through a trained generator. Starting from a randomly sampled latent vector, the trajectory is defined using TrajectoryNet, which ensures a smooth and structured interpolation in the latent manifold. The generator then transforms these intermediate latent representations into corresponding face images.

A notable observation is the semantic smoothness in transitions—from abstract or stylized appearances at the ends to more realistic and coherent faces in the middle of the sequence. This suggests that the trajectory passes through high-density regions of the latent space where the generator performs reliably. The symmetry of the progression, with the starting and ending faces resembling each other, hints at a cyclic or looping behavior in the latent dynamics modeled by TrajectoryNet.

Such visualizations are instrumental in understanding how neural networks learn structured transformations in high-dimensional spaces and validate the continuity and coherence of learned latent representations.

C. Trajectory-based Latent Output (Smoothed)

Fig. 3: TrajectoryNet output with smoothed latent interpolation. Faces evolve gradually, exhibiting continuous semantic shifts.

In this iteration, we introduce a smoothed latent trajectory using a 1D average pooling operation to enhance temporal coherence in the latent transitions. Compared to raw trajectories, the smoothed path reduces abrupt artifacts and stabilizes intermediate representations. The figure shows a 10-step evolution between two latent points, where facial features such as gender, hairstyle, and lighting conditions change gradually.

Interestingly, the generated images toward the center of the trajectory exhibit clearer facial geometry and feature blending, suggesting that the trajectory passes through high-likelihood regions in the latent space. On the other hand, the start and end points tend to be noisier or more

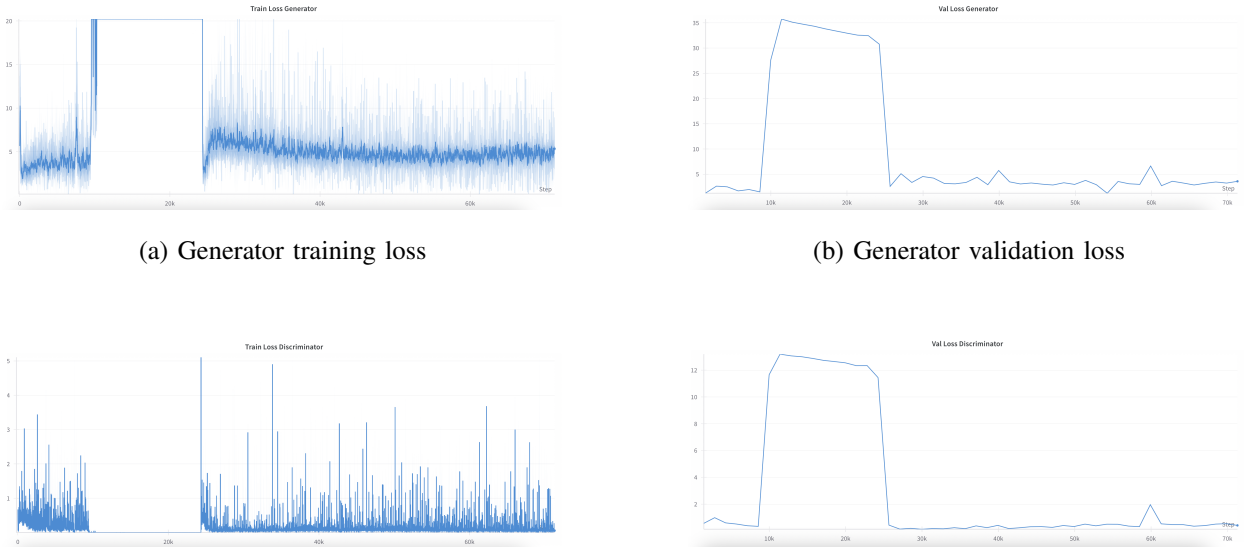
abstract, possibly indicating regions where the generator has lower confidence. This observation aligns with the idea that well-trained latent dynamics traverse semantically meaningful manifolds, guided here by TrajectoryNet's learned flow.

Such smooth interpolations serve as a qualitative diagnostic tool to assess the continuity, expressiveness, and structure of the latent space and the generator's capacity to synthesize plausible interpolants.

D. Frechet Inception Distance (FID)

A Frechet Inception Distance (FID) score of 32 was observed on the CelebA dataset, indicating reasonable image quality and diversity for a DCGAN-based model. This metric quantifies the similarity between the distributions of real and generated images by comparing feature representations extracted from a pretrained Inception network. Lower FID scores imply greater fidelity and diversity in the generated images

E. Loss Curve



(a) Generator training loss

(b) Generator validation loss

(c) Discriminator training loss

(d) Discriminator validation loss

Fig. 4: Training and validation loss curves for generator and discriminator.

V. Conclusion

This paper describes a latent trajectory-based framework for semantic image editing without relying on pretrained models. By treating semantic changes as smooth paths in latent space, we

achieve interpretable, continuous transformations with improved realism and identity preservation. This approach is architecture-agnostic, making it compatible with a wide range of GANs and use cases. Future work may involve integrating user-guided editing and applying this method to high-resolution real-world datasets.

REFERENCES

[1] I. Goodfellow et al., ”Generative Adversarial Networks,” in *Advances in Neural Information Processing Systems*, 2014.

[2] A. Radford, L. Metz, and S. Chintala, ”Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” arXiv preprint arXiv:1511.06434, 2015.

[3] Y. Shen, C. Yang, X. Tang, and B. Zhou, ”Interpreting the Latent Space of GANs for Semantic Face Editing,” in *CVPR*, 2020.

[4] R. Patashnik, O. Metzer, Y. Shechtman, and D. Cohen-Or, ”StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery,” in *ICCV*, 2021.