Progress Report: TransGAN for Point Cloud Reconstruction

Project Title: TransGAN for Point Cloud Reconstruction
Project Mentor: Seema Kumari
Team Members: Aditya Kumar, Vatsalraj Rathod, and Vedanshi Raiyani

---

Overview

This project aims to implement TransGAN for tasks related to point cloud reconstruction, including completion, upsampling, and generation. TransGAN, a GAN architecture based on transformers, is being adapted to process 3D point cloud data, which is essential in areas like 3D object reconstruction and modeling. The project builds on recent advancements in GANs, such as PointStyleGAN, with a particular focus on 3D point cloud data.

Initial Weeks:

1. Learning Phase

In the early phase of the project, the team worked on gaining a solid understanding of point cloud processing, classification, segmentation, and GAN architectures. The following papers were critical to the theoretical foundation of the project:

- **“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”**
  This paper introduced a groundbreaking method for point cloud classification and segmentation, laying the theoretical foundation for the project.
- **“PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”**
  This extended the PointNet architecture by introducing hierarchical feature learning, which is crucial for understanding more advanced techniques in 3D data processing.
- **TransGAN and PointStyleGAN Research Papers**
  - **"TransGAN: Two Transformers Can Make One Strong GAN"**
    The paper provides insights into how transformer-based approaches in GANs can be adapted for tasks such as image super-resolution and point cloud upsampling.
  - **"PointStyleGAN: Generative Point Cloud Completion and Stylization"**
    Demonstrates the application of GANs in 3D point cloud tasks such as completion and stylization, which aligns with the project's objectives.

In addition, the team explored relevant GitHub repositories like **PointStyleGAN** and **TransGAN**, which provided essential codebases and resources.

2. Troubleshooting and Error Resolution Attempts

In the initial weeks of the project, we explored two different approaches. Our first task involved running a point cloud classification model on several large datasets to gather and analyze the results. However, we encountered datatype mismatch errors that impeded the execution of the code. Parallelly, we also worked on addressing dimensional errors in a Point Style GAN code, which originated from a different project group. Although we dedicated considerable time to troubleshooting both sets of errors, we were unable to fully resolve them. After troubleshooting with no resolution, the team shifted to working on implementing **TransGAN** for point cloud reconstruction.

Current Progress

1. TransformerBlock3D and PointCloudGenerator Definitions (past weeks)

- **TransformerBlock3D**: This neural network block uses transformer architecture components like self-attention and feed-forward layers, along with normalization and dropout. These transformers are used in both the generator and discriminator of the TransGAN model.
- **PointCloudGenerator**: The generator is responsible for producing 3D point clouds from latent vectors, leveraging multiple transformer layers to transform latent space into 3D points.

2. Visualization of Point Clouds (recent development)

- The visualization component uses **Open3D** to display generated point clouds in real-time. This is used to monitor and visualize the output of the generator and discriminator models as they process 3D data.

3. Random Point Cloud Generator (`pointcloud_random.py`)

- **Purpose**: This script defines the method for generating randomly distributed 3D point clouds.
- **Processes**:
  - Latent space vectors sampled from a normal distribution are transformed into 3D coordinates.
  - A training loop alternates between training the discriminator (classifying real vs. fake point clouds) and the generator (producing realistic point clouds).

4. Spherical Point Cloud Generation and Discriminator Training (`pointcloud_sphere.py`)

- **Purpose**: To simulate real-world objects using point clouds distributed on a sphere's surface.
- **Processes**:
  - A function `generate_sphere_points()` generates real spherical point clouds.
  - The discriminator is trained to differentiate between spherical point clouds (real) and randomly generated point clouds (fake).

5. Training Loop and Losses (recent phase)

- The training loop generates batches of both real (spherical) and fake (random) point clouds.
  - **Discriminator Training**: The discriminator learns to classify real versus fake point clouds.
  - **Generator Training**: The generator is trained to fool the discriminator by generating realistic 3D point clouds.

Timeline

- **Two weeks ago**:
  - The code for the transformer block and the basic setup for the point cloud generator was likely implemented, creating the architecture for generating random point clouds.
- **One week ago**:
  - The spherical point cloud generation code and corresponding discriminator training were introduced. This focused on generating real-world-like point clouds with spherical symmetry, training the discriminator to classify them.
- **Recent days**:
  - The training loop and visualization tools (using Open3D) were refined, and adversarial losses were incorporated to enhance model training and obtain visualized point cloud outputs.

Next Steps

- Complete the **TransGAN** implementation for point cloud completion.
- Explore point cloud **upsampling techniques** using the capabilities of TransGAN.
- Review more literature on point cloud generation to optimize the current approach and deepen the team's understanding.

Challenges

- The team faced initial challenges with dimensional errors in the classification code.
- Transitioning theoretical knowledge into practical applications, particularly using GAN architectures for 3D point cloud data, has been a continuous learning process.

Conclusion

The project has progressed well, with the team now focusing on applying TransGAN to point cloud completion and preparing to extend the approach to upsampling and generation tasks. The literature and resources provided by the mentor have been invaluable, and the team anticipates achieving significant results in the next phase.