Project Report: Single-View 2D Image to 3D Point Cloud Reconstruction

Executive Summary

This project implements a Deep Learning system capable of reconstructing 3D point clouds from single 2D RGB images. The architecture utilizes a two-stage approach: first training a Variational Autoencoder (VAE) to learn a compact latent representation of 3D shapes, and subsequently training an Image Encoder to map 2D images into that learned latent space. The system is deployed via an interactive Streamlit web application for real-time inference and visualization.

System Architecture

The project consists of two distinct neural network components that share a common latent space.

2.1 3D Point Cloud VAE

The Variational Autoencoder learns to compress and reconstruct 3D geometric data.

- **Encoder (PointNet-style):** The encoder processes an input point cloud of shape (B, N, 3) using 1D convolutions with batch normalization. It increases channel dimensions (3 to 64 to 128 to 256) and applies a global max pooling operation to extract global features, resulting in a 256-dimensional latent space.
- **Decoder (MLP):** The decoder takes the latent vector and passes it through fully connected layers (256 to 512 to 1024). The final layer outputs 2048 **x** 3 values, reshaped into a standard point cloud format.
- **Activation:** The decoder uses `tanh` activation to constrain output points within a normalized range.

2.2 Image Encoder

The image encoder is responsible for understanding 2D visual data and translating it to 3D features.

- **Backbone:** Uses a ResNet18 architecture pretrained on ImageNet to extract rich visual features.

- **Feature Projection:** The final fully connected layer of ResNet is removed and replaced with a linear layer that projects the features (size 512) into the shared 256-dimensional latent space defined by the VAE.

Training Pipeline

The training is split into two sequential phases to ensure stability.

Phase 1: VAE Training

- **Objective:** Learn to reconstruct 3D point clouds.
- **Configuration:**
  - **Optimizer:** Adam with Learning Rate 0.001.
  - **Scheduler:** StepLR (gamma 0.5 every 20 epochs).
  - **Duration:** Configured for 30 epochs (optimized for speed) or 50 epochs (standard).
  - **Checkpoints:** Saves the model with the lowest validation loss to `checkpoints/vae_best.pth`.

Phase 2: Image Encoder Training

- **Objective:** Map images to the frozen VAE latent space.
- **Method:** The VAE weights are loaded and frozen (non-trainable). The image encoder is trained to minimize the distance between its output latent vector and the VAE's encoded latent vector.
- **Configuration:**
  - **Optimizer:** Adam with Learning Rate 0.0001.
  - **Duration:** Configured for 30 epochs.

Dataset and Preprocessing

The project utilizes the **ShapeNet Core v2** dataset.

- **Data Selection:** The loader scans specific ShapeNet directories (defaulting to category `02691156`, Airplanes).
- **Point Cloud Generation:** 2048 points are sampled uniformly from the surface of the mesh.
- **Normalization:** Points are centered and normalized to the unit sphere.
- **2D Rendering:** For every 3D model, the system renders 3 different 2D views (front, side, top-front) at a resolution of 128 **x** 128 using orthographic projection.

Performance Summary Table

Metric	VAE (3D Input)	Image Encoder (2D Input)	Improvement / Gap
Chamfer Distance (CD)	**0.0046**	0.0087	+0.0041 (Expected Gap)
F-Score	**0.064**	0.048	-0.016
Hausdorff Distance	**0.134**	0.231	+0.097 (Outlier Sensitivity)
Point Cloud Accuracy	**6.72%**	4.83%	-1.89%