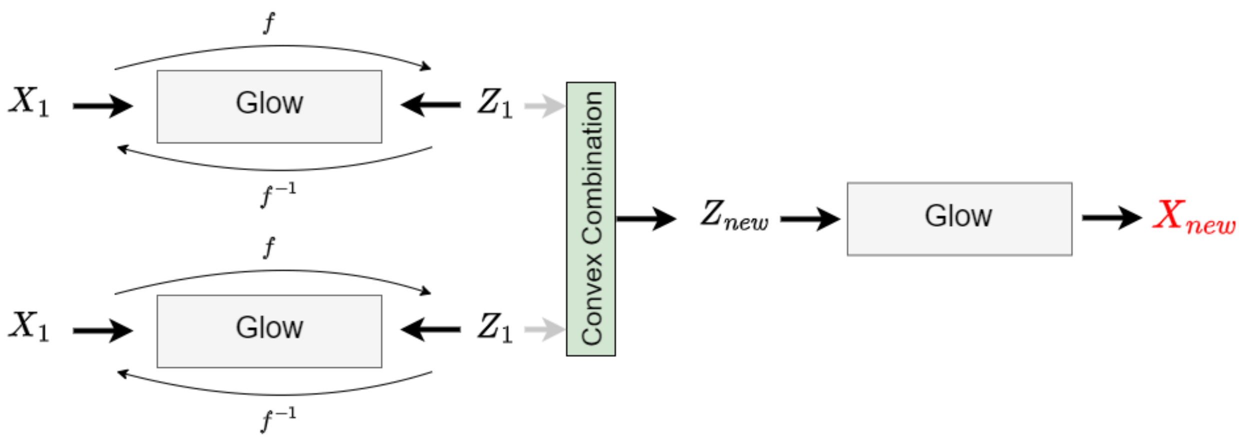
Homework 4

Coding Practice 1: Generating Images using a Pretrained GLOW Model



Use a pretrained GLOW model (or a similar normalizing flow-based model) to generate embeddings. Fuse the embeddings of different images, then apply the reverse process to generate new images with intermediate attributes.

1.1. Data Preparation

- **You can use any arbitrary face images (try with several images); However, for one of your test cases, include your own image as one of the inputs!**

1.2. Inference

First, read the article carefully to fully understand the concept and how it works. You can either train the model yourself or use an available pretrained flow-based model. Follow the procedure in the provided scheme.

1.3. Face State Combination

Take a photo of yourself, add noise with varying levels of amplification to its latent representation, then apply the reverse process to reconstruct the image. Compare the resulting image with the original to observe the effects of the added noise.

1.4. Embeddings Alteration

Modify certain values in the latent space Z (e.g., by setting one or more values to zero), reconstruct the new image, and compare it with the original to observe the effects of the changes.

1.5. Face State Transformation

Take a serious photo of yourself. Then, find two images of the same person—one where they are smiling and another where they look serious. Compute the difference between their latent embeddings. Add this difference to your own serious image's embedding, then apply the reverse process to reconstruct the image. Observe whether your expression changes as a result.

$$Z_{\Delta} = Z_{smiling} - Z_{serious} \quad ,$$

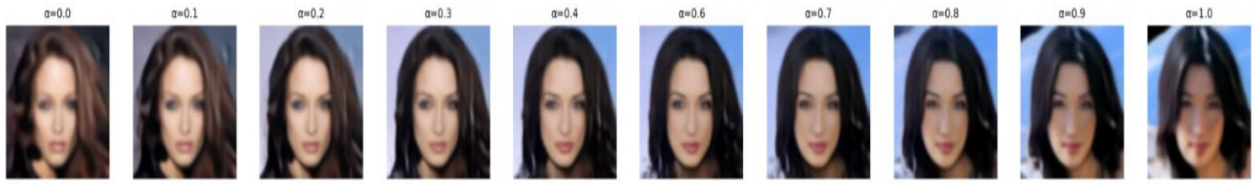
$$Z_{new} = Z_{your\ image!} + Z_{\Delta}$$

1.6. Embeddings Fusion

To create the new image, you must use a convex combination of the two generated embeddings before applying the reverse pass. The new embedding is created by following this procedure.

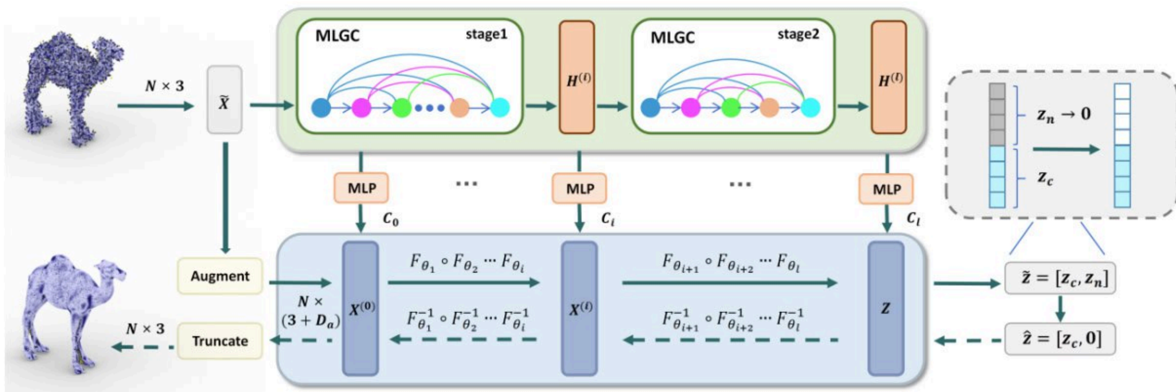
$$Z_{new} = (1 - \alpha) Z_1 + \alpha Z_2 \ , \ \alpha \in \{0,\ 0.05,\ 0.10,\ \ ...,\ 1.00\}$$

The output will be as follows (two images are from the CelebFaces Attributes dataset):



For the implementation, you are free to use the [official codebase](), but you must be able to interpret and explain every line of your code. Although the official codebase uses the TensorFlow framework, you have to use PyTorch for inference.

Coding Practice 2: Denoising Point Clouds in Latent Space



In this section, investigate how manipulating the embedding space of a model (trained on 3D point clouds)—for example, by zeroing out part of the embedding or applying other techniques—affects noise reduction and overall model performance.

1.1. Data Preparation

- **Use three arbitrary samples from the corrupted version of the ModelNet40 dataset—called [ModelNet-40C]—which includes 15 corruption categories; you have to use the background noise category. The dataset is free to download.**

1.2. Inference

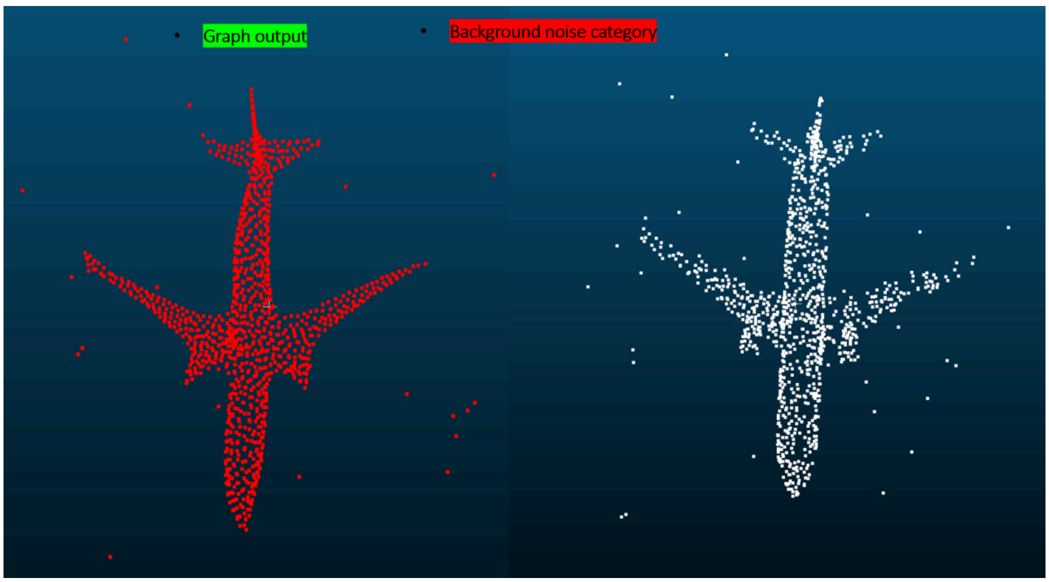
- **First, read the [article] carefully to fully understand the concept and how it works. You can simply use its pretrained model.**
- **In the first scenario, follow the procedure in the article to set only the corresponding part of the embedding to zero in order to remove noise.**
- **In the second scenario, observe the effect of not setting that part of the embedding to zero and using the original embedding instead.**

- **In the third scenario, replace this part of the embedding with various random noise vectors and observe the effect.**
- **In the fourth scenario, zero out a smaller or larger number of values (preserving the original dimension).**

1.2. Visualization

- Visualize the 3D point cloud for each scenario and compare it with its original corrupted version, as shown in the following comparison.

Following the procedure in the article, the following result is obtained.



For the implementation and pretrained model, feel free to use the official codebase.

references

- Kingma, Durk P., and Prafulla Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions." *Advances in neural information processing systems* 31 (2018).
-
- Sun, J., Zhang, Q., Kailkhura, B., Yu, Z., Xiao, C., & Mao, Z. M. (2022). Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*.

- https://openaccess.thecvf.com/content/CVPR2024/papers/Mao_Denoising_Point_Clouds_in_Latent_Space_via_Graph_Convolution_and_CVPR_2024_paper.pdf