Ben-Gurion University

Image Synthesis via Shortcut Models

Final Project

Generative Models - 361.2.2370

Introduction

This project explores the use of shortcut models to improve Text-to-Image (T2I) tasks by reducing computational costs while maintaining high-quality outputs. Unlike traditional CycleGAN-based or Diffusion-based models, shortcut models enable efficient image generation with fewer inference steps.

Diffusion models

Diffusion models are a class of generative models that focus on learning the transformation from **noise to data** by simulating a forward process of **gradually adding noise** to a data point and **learning the reverse process of denoising**.

A diffusion model consists of three major components:

1. **Forward process** - progressive Gaussian noise addition
2. **Reverse process** - step-by-step noise removal
3. **Sampling procedure** - the final step where the model generates new data by reversing the noise added during the forward process

Flow-matching models

In contrast to diffusion models, flow-matching models approach the problem by **learning an ordinary differential equation** (ODE) that directly transform noise to data. A key concept is the notion of the velocity field v_t which determines the direction from the noise point x_0 to the data point x_1. The datapoint x_t is defined as the linear interpolation:

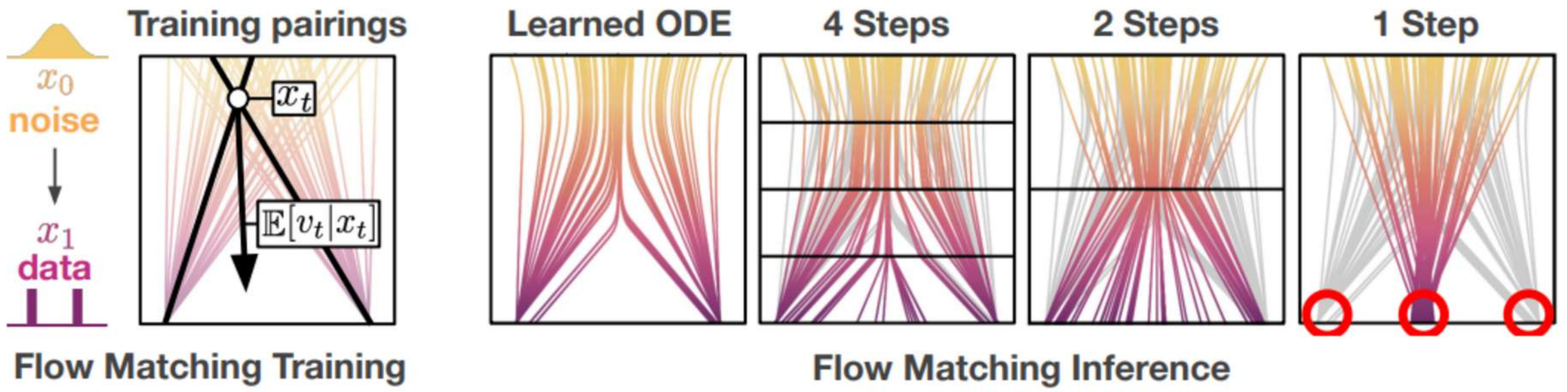
$$x_t = (1-t)x_0 + tx_1 \quad t \in [0,1]$$
$$v_t = x_1 - x_0$$

Given only x_t, multiple pairs of (x_0, x_1) are possible, introducing uncertainty in the velocity direction. Thus, the velocity is treated as a random variable and the model learns the expected velocity $\bar{v}_t = \mathbf{E}[v_t|x_t]$ by averaging over all plausible velocities. Therefore, the loss function:

$$L(\theta) = \mathbf{E}_{x_0,x_1\sim D}[||\bar{v}_\theta(x_t,t) - (x_1 - x_0)||^2]$$

Flow-matching models

- **Training** - Training paths are created by randomly pairing data points and noise. Uncertaintly of v_t direction given only x_t
- **Inference** - A perfectly trained ODE maps the noise distribution to the data distribution in continuous time. With finite steps, this guarantee is lost, and fewer steps lead to generations biased towards the dataset mean



K. Frans, D. Hafner, S. Levine, and P. Abbeel, “One step diffusion via shortcut models,” arXiv preprint arXiv:2410.12557, 2024.

Shortcut Models

- A family of denoising generative models
- It conditions the model both on the timestep **t** and the step-size **d**, allowing to handle different sampling budgets
- Flow-matching models learns an ODE to map noise to data along curved paths but **large steps can lead to errors**. The shortcut models overcome this by allowing the model to "jump" to the correct next point, accounting for future curvature, and by that greatly reduce inference time while preserving high performance:

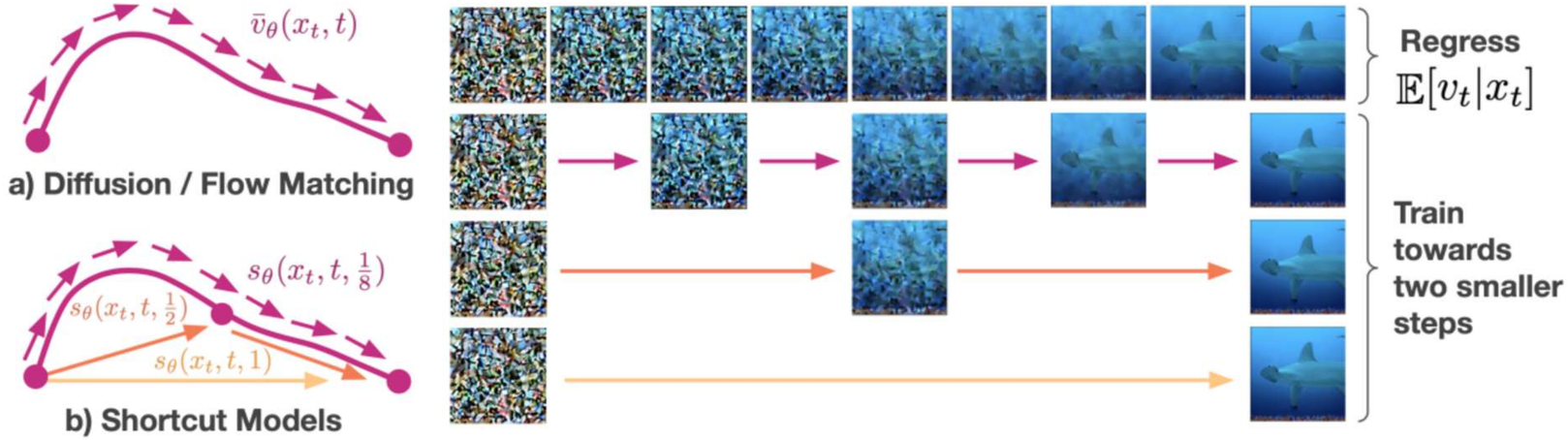
$$x'_{t+d} = x_t + s(x_t, t, d)d$$

- At $d \to 0$, the shortcut is equivalent to the flow

Shortcut Models

- To efficiently train the shortcut, instead of fully simulate ODE forward with small enough step size, it uses a **self-consistency** property, where one-step can be broken into two consecutive smaller steps:

$$s(x_t, t, 2d) = \frac{s(x_t, t, d)}{2} + \frac{s(x'_{t+d}, t, d)}{2}$$



K. Frans, D. Hafner, S. Levine, and P. Abbeel, "One step diffusion via shortcut models," arXiv preprint arXiv:2410.12557, 2024.

Algorithm

Algorithm 1 Shortcut Model Training

while not converged **do**
  $x_0 \sim \mathcal{N}(0, I),\ x_1 \sim D,\ (d, t) \sim p(d, t)$
  $x_t \leftarrow (1 - t)\, x_0 + t\, x_1$ — Noise data point
  for first k batch elements **do**
    $s_{\text{target}} \leftarrow x_1 - x_0$ — Flow-matching target
    $d \leftarrow 0$
  for other batch elements **do**
    $s_t \leftarrow s_\theta(x_t, t, d)$ — First small step
    $x_{t+d} \leftarrow x_t + s_t\, d$ — Follow ODE
    $s_{t+d} \leftarrow s_\theta(x_{t+d}, t + d, d)$ — Second small step
    $s_{\text{target}} \leftarrow \text{stopgrad}(s_t + s_{t+d})/2$ — Self-consistency target
  $\theta \leftarrow \nabla_\theta ||s_\theta(x_t, t, 2d) - s_{\text{target}}||^2$

Algorithm 2 Sampling

$x \sim \mathcal{N}(0, I)$
$d \leftarrow 1/M$
$t \leftarrow 0$
for $n \in [0, \ldots, M - 1]$ **do**
  $x \leftarrow x + s_\theta(x, t, d)\, d$
  $t \leftarrow t + d$
return x

- The first k=1/4 batch elements were treated as a flow-matching targets and the rest as self-consistency targets
- 128 steps were chosen, creating $log_2\,(128) + 1 = 8$ possible shortcut lengths and step-sizes $d \in \left(\frac{1}{128}, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\right)$ when $d = \frac{1}{128}$ considered as 0

Loss Function

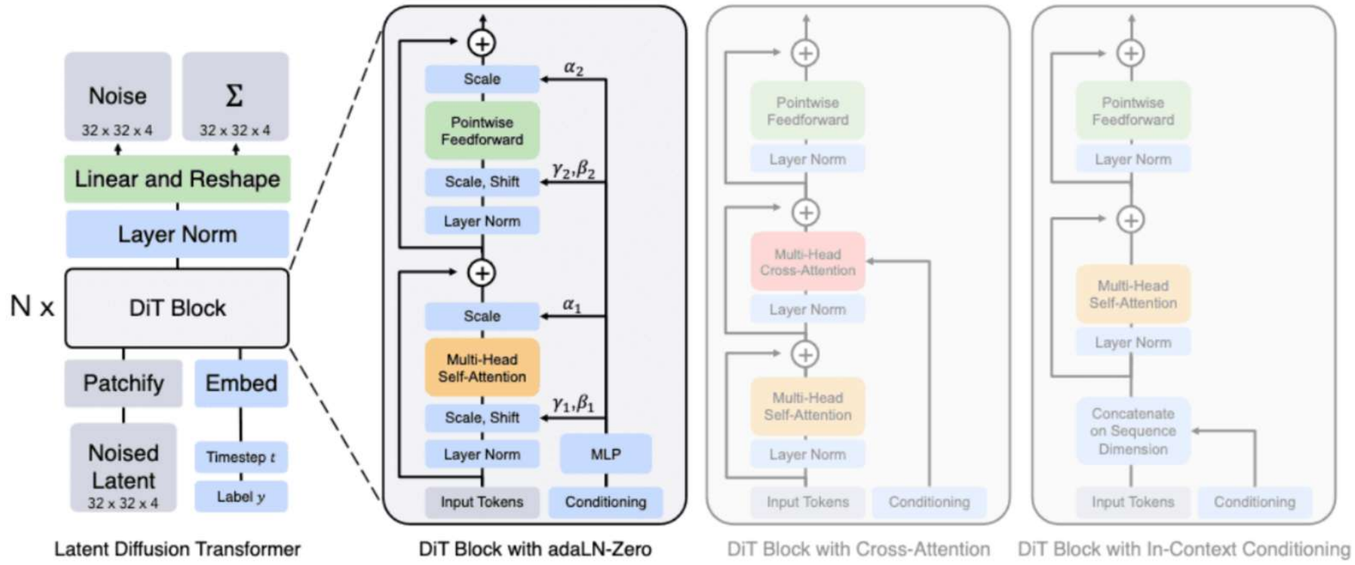
The objective function combines the flow-matching objective for smaller steps with self-consistency targets for larger steps, ensuring the model learns a consistent mapping from noise to data across a range of step sizes:

$$L_S(\theta) = \mathbb{E}_{x_0 \sim \mathcal{N}, x_1 \sim \mathcal{D}, (t,d) \sim p(t,d)} \Big[\underbrace{\|s_\theta(x_t, t, 0) - (x_1 - x_0)\|^2}_{\text{Flow-Matching}} + \underbrace{\|s_\theta(x_t, t, 2d) - s_{\text{target}}\|^2}_{\text{Self-Consistency}} \Big]$$

Diffusion-Transformer based Backbone

The DiT backbone enhances the denoising process by leveraging the transformers' ability to model **complex** dependencies in **high-dimensional** data.

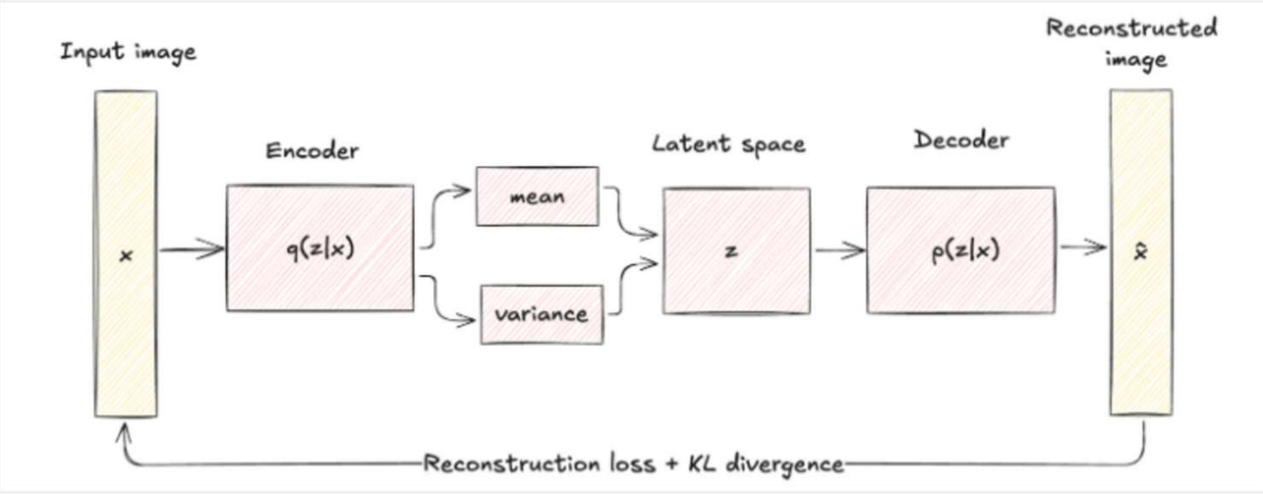
- Patchifying the Input
- Transformer Blocks
- DiT-B



W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4195–4205.

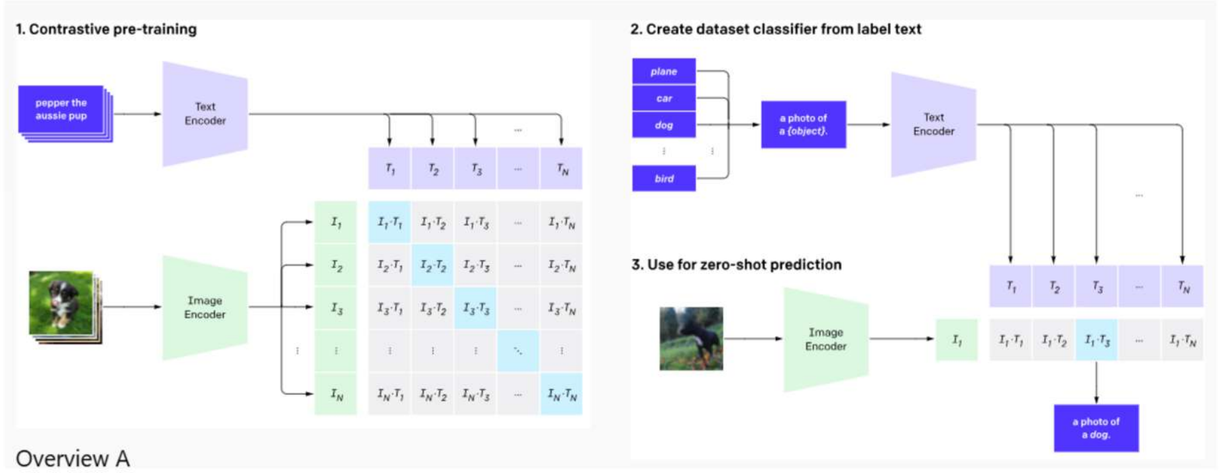
Variational Autoencoder

- To enhance the speed and efficiency of the de-noising process, we employed a variational autoencoder, reducing the need for multiple passes through the network.
- The **Image Encoder** maps input images to a **probabilistic latent representation**, capturing their underlying structure and allowing efficient generation by sampling from this distribution.
- The **Image Decoder** reconstructs images from the latent variables, ensuring **high-quality** and **varied outputs** by **minimizing reconstruction loss.**



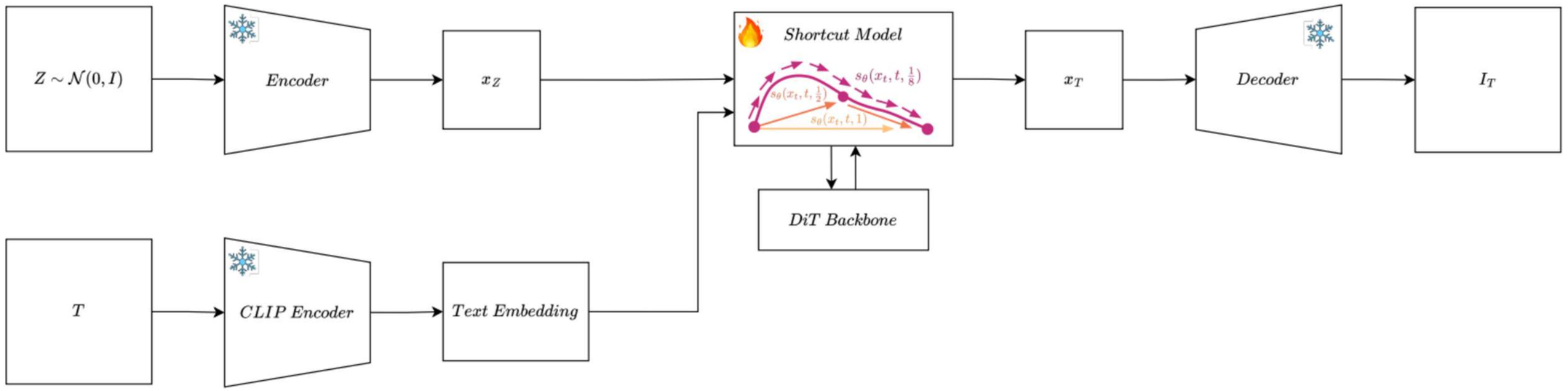
CLIP (Contrastive Language Image Pre-training) Encoder

- A powerful AI model developed by OpenAI that bridges the gap between images and text by **learning their associations**
- Trained on publicly available text–image pairs, enabling it to learn associations between **visual content** and **natural language descriptions**
- Uses a dual-encoder architecture - one encoder processes images, while the other processes text, mapping both into a shared feature space
- **Zero-shot learning capability** - allows a model to generalize using prior knowledge



A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language super vision," in International conference on machine learning. PMLR, 2021, pp. 8748–8763.

Proposed Method - Text to Image



Dataset and Pre-Processing

- We trained the model on **CelebA-HQ** dataset containing 30,000 high-quality celebrity faces. We utilized the dataset by replacing binary class labels of man and woman with **detailed text descriptions**, allowing our model to learn nuanced relationships between text and visual features

- Images were **resized** to a resolution of 256×256 pixels to ensure computational efficiency while preserving visual detail and **normalized** to values between -1 and 1

Model’s Adjustments

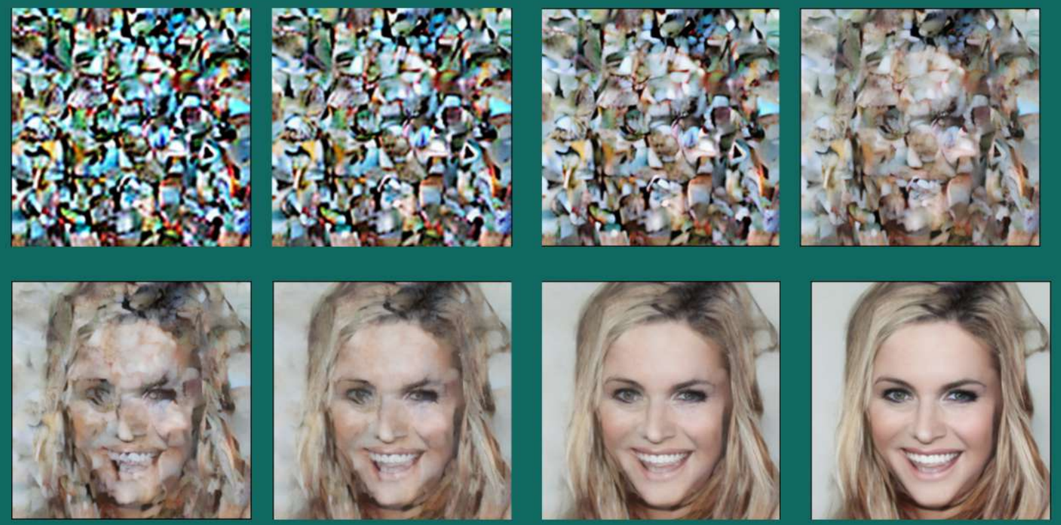
- The original **JAX** implementation for TPUs was adapted to **PyTorch** due to resource limits and model changes
- Incorporated **CLIP** for aligning text embeddings
- Replacing image labels with detailed **captions** to align with CLIP's input requirements
- **Train** the shortcut model from scratch and **fine-tune** CLIP to align with architectural changes
- Reducing the number of **epochs** due to limited computational power

Metrics

Frechet Inception Distance (FID)

$$FID = \|\mu_r - \mu_g\|^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right)$$

- A metric used to evaluate the quality of generated images by comparing the distribution of features extracted from real and generated images
- Calculates the Fréchet distance (a measure of similarity)
- Lower FID scores indicate higher similarity and better quality of generated images



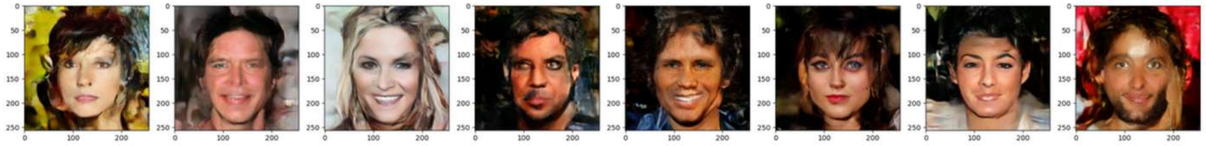
“You can't manage what you don't measure.”

—Peter Drucker

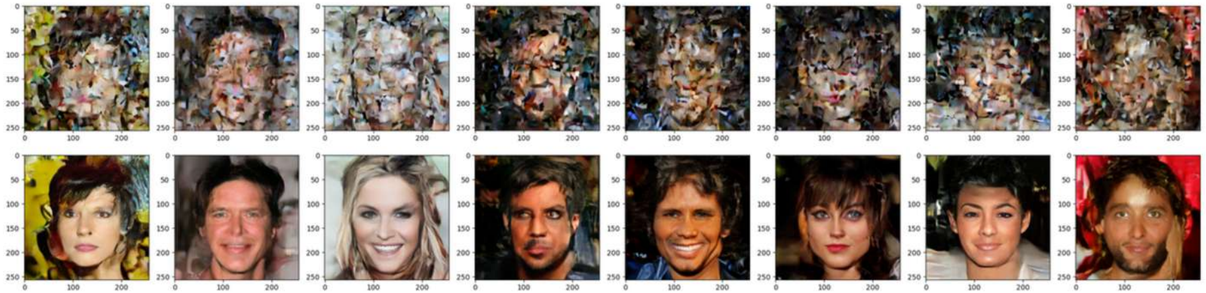
Results

Diffusion Process

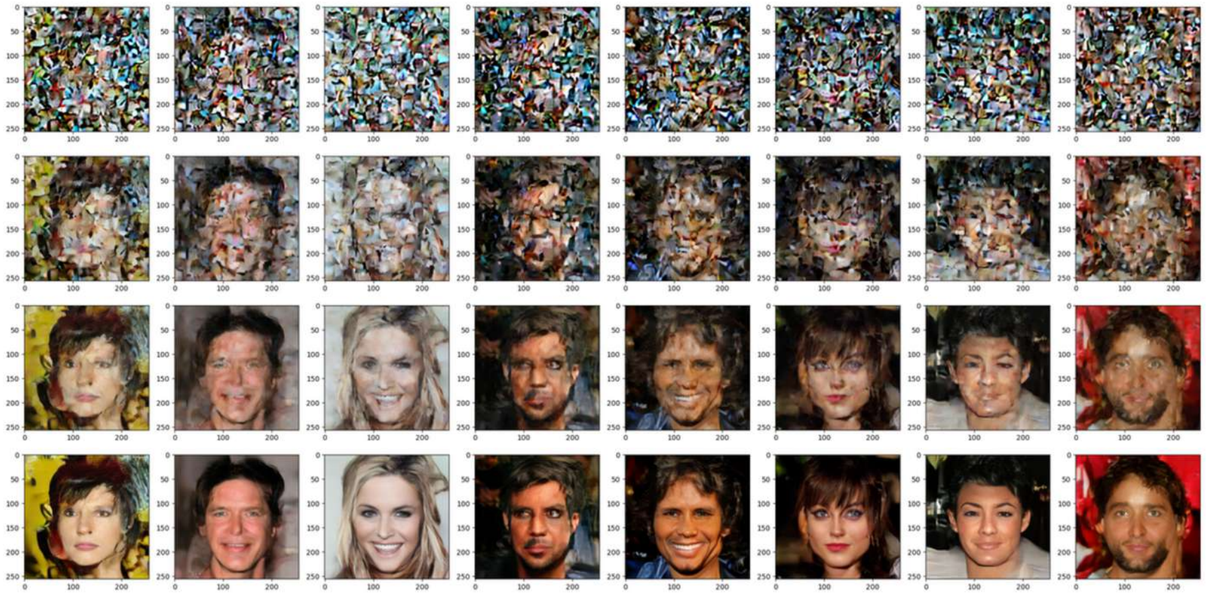
1 - step



2 - step



4 - step



Results

Diffusion Process

128 - step

Results

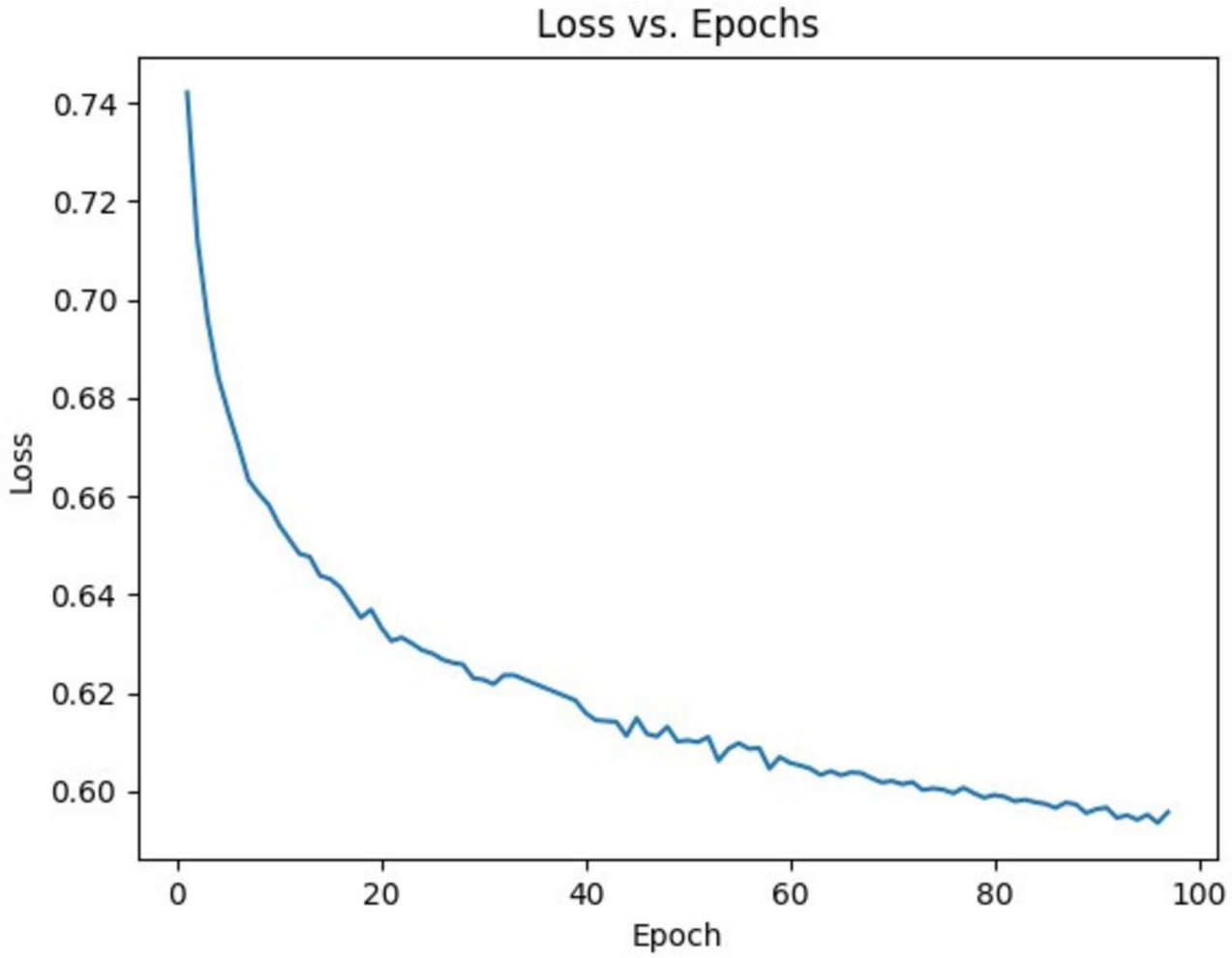
Inference

- The trained model is conditioned to "jump" across steps in the denoising process, guided by the step size parameter
- The user inputs a text prompt and the number of steps and the model generates image accordingly
- **Input** - Text embeddings [from 512 to 768], **Output** - RGB images [256x256x3]

Text Prompt: **“A Man With Glasses”**
(For steps = 128)

Results

Loss Function

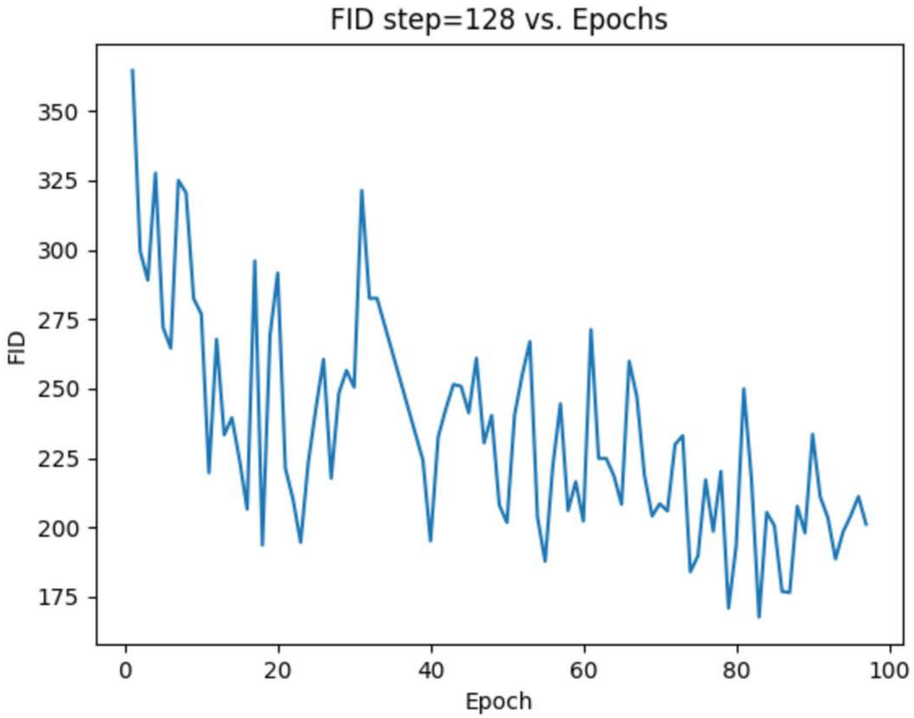


Results

FID Values

- The graph shows a downward trend in values, and with continued training, we expect them to decrease further significantly
- The table shows FID values for the 100th epoch for different number of steps

Steps	FID ↓
1	185.5
2	187.7
4	193.4
8	191.8
16	186.1
32	178.5
128	175.8



Conclusions and Future Work

- Although the training process requires about 16% more compute that of a base diffusion model, the inference stage is indeed fast and results high quality images

- Due to our limited computational power the current model must be farther trained with more epochs to get better results or alternatively use a bigger DiT architecture

- Other datasets can be chosen for this tasks for comparison

Thank You!