Project Report
Image Synthesis via Shortcut Models

Generative Models - 361.2.2370

Abstract

Traditional approaches for Text-to-Image (T2I) tasks have largely been based on Generative Adversarial Networks (GANs). For T2I generation, GAN-based architectures such as StackGAN [1] have successfully produced images from text by progressively refining the output in stages. However, more recent innovations have introduced even more powerful techniques, including diffusion models and transformer-based methods. Models such as DALL · E [2], MidJourney, and others have excelled in various image synthesis tasks. Additionally, Stable Diffusion [3] is another notable model that employs a latent diffusion process to efficiently generate high-quality images from textual descriptions, offering impressive results with reduced computational overhead. Despite the remarkable success of these advanced methods in generating images, they often require substantial computational resources and long inference times, making them unsuitable for applications with limited time or budget constraints.

This project explores the use of shortcut models to enhance generative image synthesis for T2I tasks, focusing on reducing computational costs while preserving output quality. By integrating flow-matching and self-consistency objectives, shortcut models enable high-quality image generation with fewer inference steps compared to traditional diffusion models. The architecture incorporates CLIP encoders for semantic alignment, variational autoencoders for latent space representation, and diffusion-transformer backbones for capturing long-range dependencies. Evaluation using metrics such as Frechet Inception Distance (FID), along with visual comparison, will potentially demonstrate thelity and efficiency of the model in various image synthesis applications.

1 Problem Description

Text-to-Image (T2I) generation is a fundamental task in generative modeling, aimed at synthesizing images from textual descriptions. This process requires models to accurately interpret the meaning of the text and generate visually coherent representations. The main challenge lies in capturing fine details while ensuring that the generated image aligns precisely with the given textual input.

1.1 Text-to-Image Generation

The process of generating an image from a textual description involves converting a given text $T \in \mathcal{T}$ into a corresponding image $I \in \mathcal{I}$, such that $G : \mathcal{T} \to \mathcal{I}$, where $G(T) = I$. The goal is for the image I to accurately reflect both the high-level content and the fine details outlined in the description T. This requires the image generation model G to capture the relationships, structure, and context described in the text, ensuring that the resulting image is visually coherent, realistic, and faithful to the input description.

1.2 Key Challenges

- **Textual Guidance**: Effectively incorporating textual descriptions to guide the generative process of the output image.
- **Domain Gap**: Bridging the differences between source and target domains.
- **Content Consistency**: Ensuring that the generated image faithfully reflects the high-level content and fine details from the textual description.

1.3 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 de-noising. These models work by defining a fixed forward Markov chain that starts with a data point x_0 drawn from a real-world data distribution $p(x_0)$ and progressively adds noise, resulting in a noisy sample x_T from a simple noise distribution, typically a standard normal $\mathcal{N}(0, I)$. The goal of the model is to learn the reverse process, which transforms x_T back into x_0, recovering data samples from the noise distribution. The reverse process is parameterized as a sequence of denoising steps, each modeled by a neural network $\epsilon_\theta(x_t, t)$, which predicts the noise added at each time step t. The training objective is to minimize the discrepancy between the predicted noise $\epsilon_\theta(x_t, t)$ and the true noise ϵ that was added during the forward process. This is captured by the loss function:

$$L_{\text{DM}} = \mathbb{E}_{x,\epsilon\sim N(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right],$$

where x_t represents the noisy version of the data at time t, and ϵ is the noise injected at each step. By minimizing this objective, the model learns to reverse the noise process, allowing it to generate data samples from noise by following the learned reverse path.

1.4 Flow-matching Models

Flow-matching models approach generative modeling through continuous-time transformations, learning an ordinary differential equation (ODE) that directly maps noise to data. A key concept is the notion of a velocity field, v_t, which determines the direction from the noise point x_0 to the data point x_1. The data point x_t is defined as a linear interpolation between x_0 and x_1:

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

with the corresponding velocity field:

$$v_t = x_1 - x_0.$$

Given only x_t, multiple pairs of (x_0, x_1) are possible, introducing uncertainty in the velocity direction. Thus, the velocity v_t is treated as a random variable, and the model learns the expected velocity, $\bar{v}_t = \mathbb{E}[v_t|x_t]$, by averaging over all plausible velocities. The optimization objective for flow-matching models is to minimize the difference between the predicted expected velocity $\bar{v}_\theta(x_t, t)$ and the true velocity $(x_1 - x_0)$, leading to the loss function:

$$L_F(\theta) = \mathbb{E}_{x_0,x_1\sim D}\left[\|\bar{v}_\theta(x_t, t) - (x_1 - x_0)\|^2\right].$$

For sampling, a noise point x_0 is first drawn from a normal distribution $\mathcal{N}(0, I)$, and then the model iteratively updates x_0 to x_1 following the learned velocity field $\bar{v}_\theta(x_t, t)$, using methods like Euler sampling to approximate the continuous time dynamics in discrete steps. By learning to predict the noise in diffusion models or the velocity in flow-matching models, these approaches both enable the generation of high-quality data samples starting from simple noise. In line with the authors of [4], we will consider flow-matching as a particular case of diffusion modeling and use the terms interchangeably.

1.5 Shortcut Models

Shortcut models are a family of de-noising generative models that address the high sampling step requirements of diffusion and flow-matching models. By conditioning the model on both the timestep t and step size d, shortcut models can handle different sampling budgets. Flow-matching learns an ODE that maps noise to data along curved paths, but large steps can lead to discretization errors. Shortcut models overcome this by allowing the model to "jump" to the correct next point, accounting for future curvature. This is formalized by the shortcut $s(x_t, t, d)$, where:

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

The goal is to train a shortcut model $s_\theta(x_t, t, d)$ for all combinations of x_t, t, and d. Shortcut models generalize flow-matching models, enabling larger jumps. As $d \to 0$, the shortcut reduces to the flow.

To efficiently train $s_\theta(x_t, t, d)$, shortcut models use a self-consistency property, where one shortcut step equals two smaller steps:

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

This enables the model to be trained with self-consistency targets for $d > 0$ and the flow-matching loss for $d = 0$. As a result, it has the potential to greatly reduce inference time, facilitating efficient sampling across various budgets, from short to long inference steps, while preserving high performance with just a few steps or even a single step. The concept of the shortcut models is illustrated in Fig. 1.

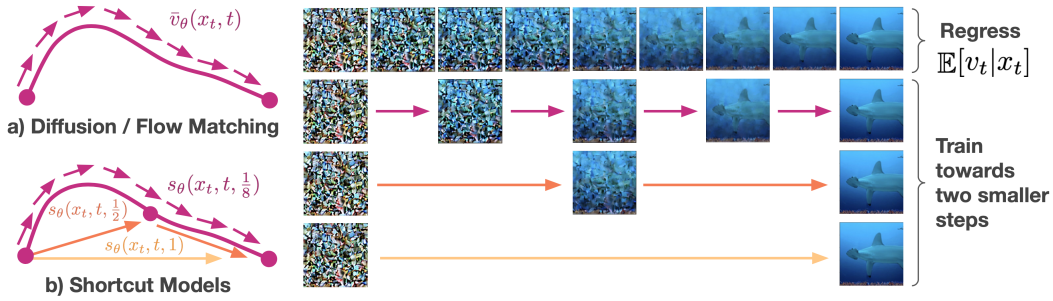

Figure 1: Overview of shortcut model training

2 Chosen Method for Solving

Our solution aims to leverage shortcut models to produce high-quality outputs efficiently, eliminating the need for iterative refinement or long de-noising processes. It operates in just a few steps or even a single step while supporting an adaptive budget.

2.1 Pre-processing

For pre-processing, we resized all images to 256×256 pixels to ensure computational efficiency while preserving visual detail. The pixel values were normalized between -1 and 1, ensuring stable training and improved convergence. Additionally, we replaced binary class labels representing "man" and "woman" with detailed text descriptions, allowing the model to learn finer-grained relationships between textual attributes and visual features. This transformation enabled a more nuanced understanding of the images in the dataset.

2.2 Architecture

In our design, we build upon the stable diffusion architecture [3], incorporating a key modification: replacing the diffusion module with a shortcut-based model. This alteration enables us to avoid the computationally expensive inference steps instead of using a streamlined process that requires only a few or a single step. The stable diffusion framework is highly versatile, supporting a variety of tasks such as text-to-image synthesis and text-guided image in-painting. The image encoder-decoder and the CLIP encoder are pre-trained and ready for integration. The architecture is illustrated as follows:

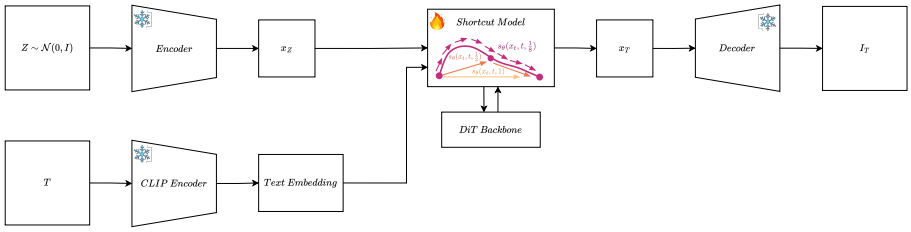

Figure 2: Text-to-Image Generation

Input: Text embeddings (from 512 to 768)
Output: RGB images $256 \times 256 \times 3$

2.2.1 CLIP Encoder

Stands for Contrastive Language–Image Pre-training. The CLIP encoder [5] plays a pivotal role in our proposed diffusion model by bridging the gap between the the textual and the visual modalities. It is designed to encode text into a shared latent embedding space, enabling seamless cross-modal understanding. In our model, it guides the generative process by providing robust semantic representations of the input prompts. This allows the model to synthesize outputs that are not only visually coherent but also accurately aligned with the intended semantics, enhancing the overall quality and applicability of the generated results.

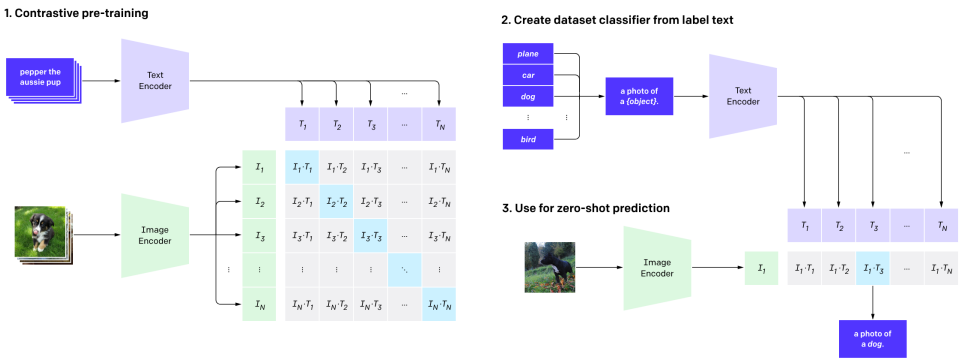

Figure 3: CLIP Overview

2.2.2 Variational Autoencoder

The de-noising process demands going through the network many times. To make this process faster and more efficient we tend to use a Variational Autoencoder. Unlike traditional autoencoders, which map inputs directly to deterministic encodings, VAEs model the latent space probabilistically, learning a distribution rather than fixed points. This is achieved by introducing an encoder network that maps inputs to a mean and variance, defining a latent Gaussian distribution, and a decoder network that reconstructs the input from sampled latent variables. The VAE is trained using a loss function that combines a reconstruction loss to preserve data fidelity and a Kullback-Leibler (KL) divergence term to regularize the latent space, ensuring it closely follows a predefined prior (usually a standard normal distribution).

- **Image Encoder**:
  The Image Encoder maps input images to a probabilistic latent representation, capturing their underlying structure and allowing efficient generation by sampling from this distribution.

- **Image Decoder**:
  The Image Decoder reconstructs images from the latent variables, ensuring high-quality and varied outputs by minimizing reconstruction loss.

2.2.3 Diffusion Transformer Backbone

We adopt the original implementation of the shortcut model, which incorporates a DiT [6] (Diffusion Transformer) model as its backbone. This approach leverages the strengths of transformers to enhance the de-noising process in diffusion models. By capturing long-range dependencies through self-attention and enabling cross-attention between text embeddings and the latent data, the DiT model significantly improves noise reduction and feature extraction. An illustration of the DiT architecture is shown below:

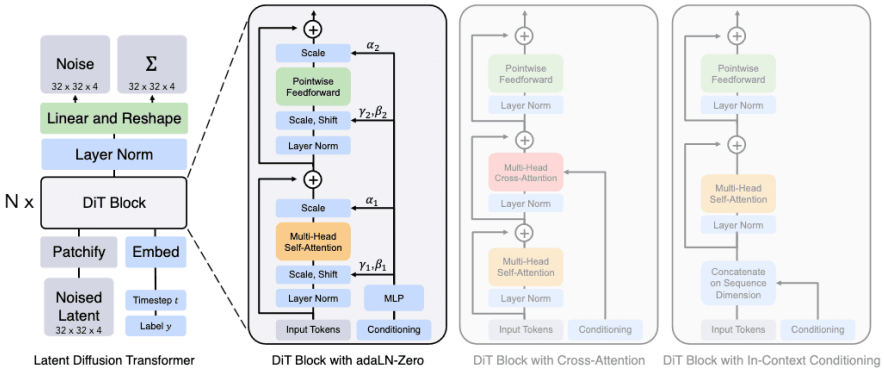

Figure 4: Diffusion-Transformer architecture

2.2.4 Objective Function

- **Flow-Matching Loss:**
  For $d = 0$, train the model to match the empirical target:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\|s_\theta(x_t, t, 0) - (x_1 - x_0)\|^2\right] \tag{1}$$

  where x_1 represents the clean image in the target domain $\mathcal{D}$, and x_0 is the noisy initialization.

- **Self-Consistency Loss:**
  For $d > 0$, train the model to predict large steps using smaller predicted steps:

$$s_{\text{target}} = \frac{s_\theta(x_t, t, d)}{2} + \frac{s_\theta(x'_{t+d}, t, d)}{2} \tag{2}$$

$$\mathcal{L}_{\text{self}} = \mathbb{E}\left[\|s_\theta(x_t, t, 2d) - s_{\text{target}}\|^2\right] \tag{3}$$

Combining both losses resulting in the global objective function:

$$L_S(\theta) = \mathbb{E}_{x_0\sim\mathcal{N}, x_1\sim\mathcal{D}, (t,d)\sim p(t,d)}\Big[\underbrace{\|s_\theta(x_t, t, 0) - (x_1 - x_0)\|^2}_{\text{Flow-Matching}} + \underbrace{\|s_\theta(x_t, t, 2d) - s_{\text{target}}\|^2}_{\text{Self-Consistency}}\Big] \tag{4}$$

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

2.2.5 Inference

During inference, shortcut models are employed to enhance the efficiency of text-to-image (T2I) synthesis. The user enters a **text prompt** and the **number of steps** (1,2,4,8,16,32,64,or 128), and the model generates the image accordingly. Unlike traditional iterative diffusion processes, the proposed architecture utilizes learned mappings to generate high-quality images in as few as 1 to 128 steps, as specified by the user. In this framework, a text prompt serves as the primary conditioning signal, guiding the generation of an output image while ensuring structural coherence and the precise application of target styles. This approach is particularly well-suited for resource-constrained environments, as it significantly reduces computational overhead without compromising output fidelity. For Example:

Figure 5: Inference Example

2.3 Training

The modified shortcut model was trained from scratch, with key architectural components carefully optimized to enhance efficiency and performance. In particular, the CLIP model was fine-tuned to ensure better alignment with the modified architecture, improving the semantic consistency between textual inputs and generated images. To further enhance latent space representation and reconstruction fidelity, a pre-trained Variational Autoencoder (VAE) was incorporated into the framework, leveraging its ability to compress high-dimensional data into a more structured latent space.

Due to computational constraints and time limitations, the total number of training epochs was reduced from 300, as was suggested in the article, to 100. Future work may explore strategies to further optimize training efficiency, such as leveraging distributed training techniques or more computationally efficient backbone architectures.

2.3.1 Joint Optimization

By optimizing the losses Eq. 1 and Eq. 3 together, the model can simultaneously handle both fine-grained (small-step) and coarse-grained (large-step) denoising tasks. This approach simplifies the training pipeline by eliminating the need for multiple training phases or distillation procedures. It also provides flexibility, allowing the number of inference steps ranging from single-step to many-step generation to be chosen freely after training, without requiring retraining or additional tuning. Furthermore, it seamlessly propagates the model's generation capability from many-step to few-step settings, ensuring high-quality results across different inference budgets.

2.3.2 Classifier-Free Guidance

Classifier-Free Guidance (CFG) is a technique that eliminates the need for a separate classifier by leveraging the generative model itself, trained in both conditioned and unconditioned modes. During training, the model alternates between receiving conditioning inputs (e.g., text prompts) and generating without any conditions, enabling it to learn both conditioned and unconditioned generation effectively.

At inference time, CFG combines the outputs of the conditioned and unconditioned modes to guide the model more strongly towards the conditioned output. This is achieved by adjusting the output as a linear combination, expressed mathematically as:

$$\epsilon_{guided} = \epsilon_{uncond} + w \cdot (\epsilon_{cond} - \epsilon_{uncond})$$

where ϵ_{cond} and ϵ_{uncond} are the conditioned and unconditioned outputs, respectively, and w is the guidance scale. The benefits of CFG include improved fidelity of the generated samples by strongly aligning them with the desired condition and the flexibility to control the trade-off between sample fidelity and diversity through the guidance scale w. In this paper, CFG is applied during evaluation at small step sizes (e.g., $d = 0$), but the authors note that it can be error-prone at larger step sizes due to limitations in the linear approximation.

3 Novelty of method with respect to the literature

We propose an end-to-end text-to-image synthesis framework that introduces novel shortcut model into an existing stable diffusion architecture to improve the efficiency of image generation. In contrast to the standard implementations of diffusion with several inference steps, this strategy has the potential to reduce inference time while maintaining high-quality output images, with only very few sampling steps. In our method, the textual input is first encoded into a latent representation using a CLIP text encoder, which then serves as the conditioning input for the diffusion process. This novelty stems from conditioning the model by timestep t and step size d which enables the model to bypass the conventional progressive denoising process typically required for image generation.

4 Datasets

The training process was conducted using the CelebA-HQ dataset, which consists of 30,000 high-resolution images of celebrity faces. This dataset is widely used in generative image synthesis tasks because of its diverse range of facial attributes, high-quality annotations, and balanced distribution of features. Using CelebA-HQ, the model was trained to generate visually realistic and semantically coherent images while preserving fine-grained facial details.
To evaluate the quality of the generated images, we employed the Frechet Inception Distance (FID) score, a widely accepted metric for assessing the realism and diversity of synthetic images. Lower FID scores indicate higher similarity between generated and real images, making it a crucial benchmark for performance comparison. The FID-based validation was conducted following the methodology outlined in the article *"One-Step Diffusion via Shortcut Models"*, ensuring consistency with established evaluation protocols.

5 Definition of Success and Training Goals

5.1 Inference Time

A key objective of this work is to **minimize inference time** while maintaining high visual fidelity in generated images. Efficient image synthesis is particularly critical for real-time or resource-constrained applications, where computational efficiency directly impacts practical usability. The proposed shortcut model is designed to significantly accelerate the generative process by reducing the number of required inference steps compared to conventional diffusion-based methods.

5.2 Frechet Inception Distance

The Frechet Inception Distance (FID) is a common metric used to evaluate the quality of images generated by generative models. It measures the distance between the feature distributions of real and generated images using a pre-trained Inception v3 model. The FID score is calculated as:

$$FID = \|\mu_r - \mu_g\|^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}) \tag{5}$$

where μ_r and Σ_r are the mean and covariance of the real image features, and μ_g and Σ_g are the mean and covariance of the generated image features. Lower FID scores indicate better quality and closer resemblance to real images.
The article on which our shortcut model is based reported FID scores ranging from 20 to 40 for 1-step inference, 14 to 30 for 4-step inference, and 7 to 16 for 128-step inference, depending on the dataset. These values were our target performance, which we aimed to achieve if sufficient computational resources had been available for extended training.

5.3 Visual Comparison

In addition to quantitative evaluation metrics, a visual inspection of the generated images provides qualitative insights into the model's strengths and limitations in terms of image quality and task performance. While visual assessment is inherently subjective and lacks the precision of formal metrics such as FID, it serves as an essential complementary evaluation method. By analyzing aspects such as texture fidelity, structural coherence, and semantic alignment with input prompts, we can gain a deeper understanding of how effectively the model captures intricate details and stylistic consistency. Although visual comparison does not offer a strictly numerical measure of performance, it remains a valuable tool for identifying patterns of failure, assessing perceptual realism, and guiding future improvements in model design. Coupled with quantitative metrics, this approach enables a more holistic evaluation of the model's generative capabilities.

6 Results

6.1 Inference Time

As mentioned, a key measure of success for our proposed model is its ability to substantially reduce the number of inference steps up to a single step, leading to significant time savings and lower computational requirements. Crucially, this efficiency is achieved while preserving the ability to generate high-quality images that maintain realism, structural integrity, and alignment with input conditions. By validating the effectiveness of the shortcut model, we demonstrate that it produces results almost immediately, within just a few seconds for a batch of generated images, and considerably faster than traditional diffusion models, making it a viable solution for computationally constrained environments.

6.2 Loss

To assess the training progress of our model, we analyze the loss as a function of the number of epochs:

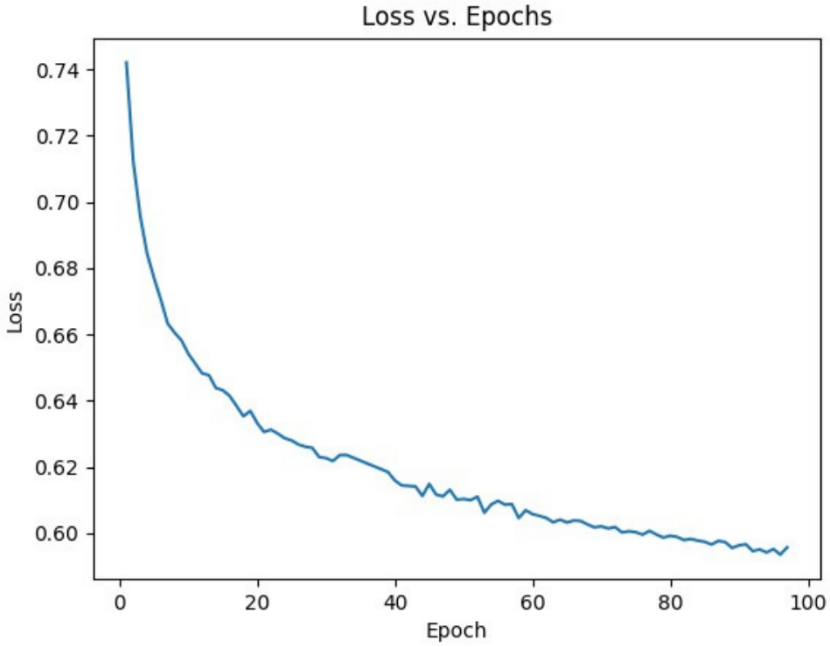

Figure 6: Loss chart as a function of number of epochs

From the chart, it is evident that the loss value decreases steadily, although it has not yet reached saturation, indicating that additional training steps are needed to achieve optimal performance.

6.3 Visual comparison

Below are examples of generated images for different number of steps. As can be observed, there is no significant visual difference for images generated with one step, two steps, or four.

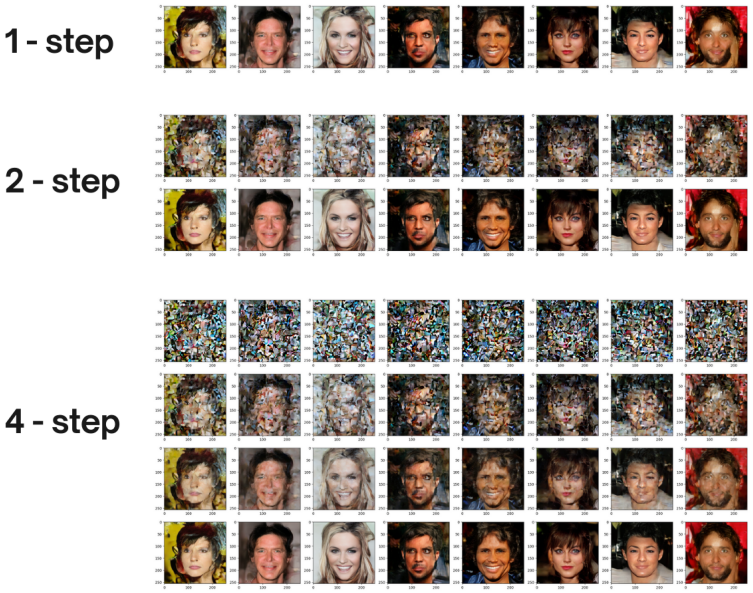

Figure 7: Generated images for 1, 2 and 4 steps

Below is an example of the same images generated with 128. Again, no significant improvement in the image qualities can be seen, showing the benefit of using the shortcut model and the high-quality results for one-step generation.

Figure 8: Generated images for 128 steps

6.4 FID

The possible inference steps in our model are 1, 2, 4, 8, 16, 32, and 128. The FID scores do not show a significant improvement as the number of inference steps increases, indicating that the quality gains from additional steps are minimal. This highlights a key advantage of the one-step model, as it achieves comparable image quality while significantly reducing computational cost. The FID scores at 4 and 8 steps are higher than at 1 and 2 steps, suggesting that additional inference steps do not always guarantee better performance. The best FID score achieved was 175.5 for 128 inference steps, aligning with the general trend that higher inference steps lead to better image quality. However, the difference between the one-step FID score (185.5) and the best FID score (175.5) is relatively small, suggesting that the improvement may not be substantial enough to justify the significantly higher computational cost. Achieving an FID score of 20 is a significant accomplishment, especially for a single inference step. If we had continued the training over 100 epochs, for example for 300 epochs as done in the reference article, we would have been able to reach their target FID values.

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

Table 1: Fréchet Inception Distance (FID) values on the 100th epoch

7 Summary

This project introduces a shortcut model for Text-to-Image generation, designed to improve the efficiency of traditional diffusion-based approaches. Unlike standard diffusion models, which require multiple iterative denoising steps, our method leverages flow-matching and self-consistency objectives to generate high-quality images with fewer inference steps, significantly reducing computational costs. The model integrates CLIP for semantic alignment, a Variational Autoencoder for latent space representation, and a Diffusion Transformer backbone to enhance feature extraction and noise reduction. We trained the model on the CelebA-HQ dataset and evaluated its performance using the visual comparison and FID metrics, demonstrating that despite the short training for only 100 epochs, the model generated high-quality images that remained visually consistent and aligned with the input text.
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. Overall, our results indicate that the shortcut model effectively balances efficiency and quality, making it a promising alternative to traditional diffusion models.

8 Additional remarks

8.1 Model's Adjustments

The original implementation of the model was developed using the JAX library to leverage TPU resources. However, due to resource constraints, we adapted the script to PyTorch, allowing for GPU utilization instead of TPUs, along with necessary adjustments to the model architecture to optimize performance within these limitations.

8.2 Future Work

Given the current limitations in computational power, the model would benefit from additional training over more epochs to improve results. Alternatively, exploring a larger DiT architecture could enhance performance. Furthermore, expanding the model's capabilities by using alternative datasets for comparative analysis could help generate images with greater detail and additional objects. This would improve the model's ability to interpret and synthesize more complex textual descriptions. Future research will focus on refining the model's generalizability across different image distributions, enabling it to produce higher-quality and more diverse outputs.

References

[1] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.

[2] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

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

[5] 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 supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[6] 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.