A Simple Diffusion Transformer for Text-to-Image Generation

Rohith Garapati
GitHub: INFINITYone22

July 2025

Abstract

This paper proposes a straightforward transformer-based diffusion model for text-to-image generation, trained from scratch without pre-trained encoders. By concatenating simple text and image embeddings and processing them through a diffusion transformer, the model generates high-resolution images (1024×1024) from arbitrary-length prompts. We detail the architecture, diffusion process, and three scalable configurations targeting 8B, 12B, and 16B parameters. This design emphasizes simplicity, efficiency in FP8 precision, and dynamic handling of long prompts.

1 Introduction

Text-to-image generation has advanced rapidly with diffusion models, but many rely on complex pre-trained components. This work introduces a simple alternative: a diffusion transformer (DiT) that uses basic embeddings and direct concatenation for cross-modal conditioning. Trained from scratch on image-text pairs, it leverages self-attention to align prompts with generated images.

Key innovations include: - Dynamic text handling with rotary positional embeddings (RoPE) for unlimited prompt lengths. - Efficient FP8 precision and flash attention for scalability. - Configurations balanced for parameter efficiency.

This concept enables accessible, customizable text-to-image synthesis, suitable for research and prototyping.

2 Model Architecture

The model consists of simple embedding layers for text and images, concatenated into a single sequence fed to a stack of transformer blocks. Diffusion is used for iterative denoising.

2.1 Embeddings and Concatenation

- **Text Embedding**: A learnable embedding matrix maps tokenized prompts to vectors of dimension d. RoPE enables dynamic lengths. - **Image Embedding**: 1024×1024 images are patched (16×16) into 4096 tokens, projected to dimension d. - **Concatenation**: Text tokens prefix image tokens, forming a sequence of length $L_{text} + 4096$.

2.2 Transformer Backbone

A stack of N layers, each with multi-head self-attention (H heads) and MLPs. Adaptive layer normalization (adaLN) injects timestep conditioning. Attention is global, allowing text to influence image tokens.

2.3 Diffusion Process

Forward diffusion adds noise to image embeddings over T steps:

$$q(z_t|z_0) = \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t}z_0, (1 - \bar{\alpha}_t)I)$$

Reverse diffusion predicts noise $\epsilon_\theta(z_t, t, c)$, where c is the concatenated sequence. Fresh text embeddings are used each step.

3 Model Configurations

We propose three configurations for different scales, optimized for FP8 efficiency and balanced depth/width. Parameters are estimated using $\approx N \times (12d^2 + 13d)$.

Config	**Dimension (d)**	**Layers (N)**	**Parameters**
8B Model	4096	32	~8.1B
12B Model	5120	32	~12.2B
16B Model	6144	36	~16.3B

Table 1: Optimal configurations for the diffusion transformer.

These prioritize layers for detailed prompt handling, with widths divisible by 128 for hardware efficiency.

4 How It Works

During training, noisy image embeddings are concatenated with fresh text embeddings and processed to predict noise. Self-attention enables a single text token (e.g., "cat") to influence all 4096 image tokens, refining features over layers and diffusion steps.

For generation: 1. Embed prompt and start with pure noise. 2. Iterate denoising: Predict and subtract noise, using fresh text each step. 3. Output: Reconstruct 1024×1024 image from denoised patches.

This yields coherent, prompt-aligned images, with depth aiding complex scenes.

5 Conclusion

This simple diffusion transformer offers an efficient path to text-to-image generation. Future work could explore larger datasets or multimodal extensions. Code available on GitHub.