Megapixel Image Generation with Step-unrolled Denoising Autoencoders

Alex F. McKinney
Durham University
Durham, UK
alexander.f.mckinney@durham.ac.uk

Chris G. Willcocks
Durham University
Durham, UK
christopher.g.willcocks@durham.ac.uk

Abstract

Advancements in deep generative modelling has pushed sample resolution higher whilst reducing computational requirements and sampling speeds. One approach works in two stages: training a powerful vector-quantization image model and then training a second discrete prior to predict discrete tokens corresponding to image patches. Early work produced high fidelity and diverse samples, but were prohibitively slow to sample from as they were autoregressive in nature. Later work exploited discrete diffusion models in order to allow for parallel token prediction, dramatically speeding up the sampling process. In this work, we push the sampling speed and computational requirements further by replacing discrete diffusion models with denoising autoencoders, as well as modifications to the Transformer backbone including axial embeddings, an hourglass structure, and resampling layers more suited to image tasks. Furthermore, the non-autoregressive nature of the model allows for arbitrary inpainting patterns. Finally, we train new vector-quantization models to allow for the sampling of upwards of a megapixel images in seconds, and without relying on sliding window mechanisms.

1. Introduction

foobar

2. Related Work

This work builds upon much prior research into powerful deep generative models [3], self-supervised methods, and efficient transformer architectures. We briefly cover relevant prior work into deep generative models in §2.1-2.3 and a recent and highly effective development into a efficient transformer architecture in §2.4.

2.1. Autoregressive Generative Models

One major deep generative model family is autoregressive models, characterised by a training and inference process based on the probabilistic chain rule. During training, they directly aim to maximise the likelihood of the data they are trained on. Prior work using these methods resulted in impressive results in terms of both sample quality and diversity, but are ultimately unwieldy for use in real world applications due to their slow sampling speed.

The slow sampling speed is due to their sequential nature, defined by the chain rule of probability. Given an input $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, an autoregressive model $p_\theta(\cdot)$ can generates new samples sequentially:

$$p_\theta(\mathbf{x}) = p_\theta(x_1, \ldots, x_n) = \prod_{i=1}^{n} p_\theta(x_i | x_1, \ldots, x_{i-1}) \quad (1)$$

meaning that the number of sampling steps is equal to the size of the decomposition of $\mathbf{x}$, making this slow for large inputs.

For certain tasks, the ordering of the decomposition of $\mathbf{x}$ is obvious, for example on text or speech. For images this is less obvious, however typically a raster scan ordering is used. Certain autoregressive models are order-agnostic, allow for arbitrary ordering to be used during training and inference.

2.2. Non-autoregressive Generative Models

foobar

2.3. Step-unrolled Denoising Autoencoder

foobar

2.4. Hourglass Transformers

foobar

3. Methodology

3.1. Latent Dataset Generation

We use the standard two-stage scheme for vector-quantized image modelling [2, 4, 7, 11] using VQ-GAN [4] as our feature extractor. Where such models are available, we use pretrained VQ-GANs for our experiments.

For higher resolution experiments (for example, FFHQ-1024 [5]), pretrained models are not available and so training our own VQ-GAN was necessary (see §3.6).

The second stage is to learn a discrete prior model over these latent variables. To enable this, we must first build a latent dataset using our trained VQ-GAN. Formally, given a dataset of images $\mathcal{X}$, a VQ-GAN encoder E with downsample factor f, and vector-quantization codebook $\mathcal{C}$ trained on $\mathcal{X}$, we define our latent dataset $\mathcal{L}$ as:

$$\mathcal{L} = \{\mathcal{C}(E(\mathbf{x})) \mid \mathbf{x} \in \mathcal{X}\} \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$ is a single element of the image dataset and $\mathbf{z} = \mathcal{C}(E(\mathbf{x})) \in \{1, \ldots, |\mathcal{C}|\}^{h\times w}$ is the corresponding discrete latent representation. In other words, each $f \times f$ pixels in $\mathbf{x}$ is mapped to a single discrete value from 1 to $|\mathcal{C}|$ (which in turn, corresponds to a vector $\mathbf{e} \in \mathcal{C}$), resulting in a latent representation of shape $\frac{H}{f} \times \frac{W}{f} = h \times w$.

We then use $\mathcal{L}$ to train a discrete prior over the latents. Coupled with the VQ-GAN decoder G, we obtain a powerful generative model.

3.2. 2D-Aware Hourglass Transformer

Inspired by successes in hierarchical transformers for generative language modelling [6], we modify their architecture for use with discrete latent representations of image data. We will later use this architecture as the discrete prior over the VQ-GAN latents.

Hourglass transformers have been seen to efficiently handle long-sequences, outperform existing models using the same computational budget, and meet the same performance as existing models more efficiently by using an explicit hierarchical structure [6]. The same benefits should also apply to vector-quantized image modelling.

Our modifications are 2D-aware downsampling, axial rotary embeddings, and removal of causal modelling constraints.

3.2.1 2D-Aware Downsampling

The original formulation of hourglass transformers [6] introduced both upsampling and downsampling layers, allowing the use of hierarchical transformers in tasks that have output sequence length equal to the input sequence length. However, applying their proposed resampling strategies directly on the vector-quantized image may not be the best strategy. Resampling is applied to flattened token sequence, meaning that the corresponding two-dimensional vector-quantized image is actually resampled more in one axis compared to the other. In their work they did not address this, except for experiments on ImageNet32 [8] where they resampled with a rate of $k = 3$, corresponding to three colour channels.

In our formulation, we instead reshape the flattened sequence back into a two-dimensional form and then apply resampling equally in the last two axes. With a resampling rate of k we apply $\sqrt{k}$ in each axis. We found this to significantly improve the performance of the discrete prior model, and suspect a similar approach could improve performance if applied to pixels directly, which we leave for future work.

3.2.2 Axial Rotary Embeddings

Rotary positional embeddings [10] are a good default choice for injecting positional information into transformer models, requiring no additional parameters. Additionally, they can be easily extended to the multi-dimensional case [1] which we do here. Though transformers are clearly capable of learning that elements far apart in a flattened sequence may be close in a multi-dimensional final output, we find that explicitly extending positional embeddings to the multi-dimensional case to provide a modest boost in performance.

3.2.3 Removal of Causal Constraints

In the original autoregressive formulation of hourglass transformers, great care was taken to avoid information leaking during resampling, and hence making the model non-causal [6]. We use a non-autoregressive method which is therefore not causal. Hence, in our approach we do not make any special considerations to avoid information leaking into the future.

3.3. Non-Autoregressive Generator Training

We follow the same process for training the discrete prior model step-unrolled denoising autoencoders (SUNDAE) [9].

3.4. Generating High-Resolution Images

3.5. Arbitrary Pattern Inpainting

3.6. Training a megapixel VQ-GAN

4. Evaluation

BEPIS

5. Conclusion

FUBAR

References

[1] Stella Biderman, Sid Black, Charles Foster, Leo Gao, Eric Hallahan, Horace He, Ben Wang, and Phil Wang. Rotary embeddings: A relative revolution. blog.eleuther.ai/, 2021. [Online; accessed]. 2

[2] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P. Breckon, and Chris G. Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes, 2021. 1

[3] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021. 1

[4] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. 1

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 2

[6] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models, 2021. 2

[7] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. 1

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 2

[9] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation, 2022. 2

[10] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021. 2

[11] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 1