Discrete Diffusion for Image Generation

Anthony Zhan
Stanford University
azhan9@stanford.edu

Aaron Lou*
Stanford University
aaronlou@stanford.edu

Abstract

Diffusion models have risen to the top in recent years for image generation tasks with their ability to generate diverse, high fidelity samples. Traditional diffusion processes work over continuous spaces, which makes sense for modalities like image but fail to generalize to discrete modalities like language. However, there has been a growing effort to extend diffusion to discrete spaces, yielding results comparable to autoregressive models in language (at GPT-2 scale), for example; in this work, we investigate the efficacy of these discrete approaches on generating images via tokenization. We show that using score entropy loss — the discrete analogue of score matching — and a transformer-based architecture with 186M parameters, discrete diffusion is capable of generating samples on ImageNet 256×256 *that are competitive with continuous diffusion models and autoregressive models at the same scale (although far from state-of-the-art results). Our work demonstrates that "tokenize and diffuse" is a viable paradigm for image generation and opens the door for future work on scaling these discrete diffusion models.*

1. Introduction

The field of generative modeling, specifically image generation, has seen rapid advancement over the past several years, with many models and architectures vying for dominance, including GANs, variational autoencoders, flow-based models, and autoregressive models. Recently, the most promising models to emerge have been diffusion models, which can generate high quality, diverse samples — unlike GANs, for example, which suffer from unstable training and mode collapse — while avoiding the slow inference speeds of autoregressive models [1, 2, 6, 7].

In image generation, we would like to train a model that generates images from a data distribution $q(x_0)$. Here we consider class-conditioned generation, where the input is a noised image (which can mean Gaussian/uniform noise or a blank image) along with a class label, and the output is an image drawn from the data distribution.

Diffusion models rely on a forward Markov process, where noise is gradually added to the data distribution according to a fixed schedule to obtain a sequence of latent variables $q(x_{1:T}|x_0)$, and a reverse process $p_\theta(x_{0:T})$, in which a parameterized model (usually a neural network) learns to start from pure noise (plus optionally a class label) and arrive at the data distribution. In the continuous setting, the noise is usually Gaussian, and the training objective is the ELBO on the negative log likelihood [4] (since diffusion model don't allow for easy access to actual log likelihoods):

$$\begin{aligned}\mathbb{E}[-\log p_\theta(x_0)] &\leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] \\ &= \mathbb{E}_q\left[\log p_\theta(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]\end{aligned}$$

which can be further rewritten in terms of KL divergences. Notably, in the continuous setting, the objective can be recast as *score matching*, *i.e*. learning the gradient of the log-likelihood at time t: $\nabla_x \log q_t$ [1, 6].

Despite the success of continuous diffusion models, however, many distributions of interest (*e.g*. language) are inherently discrete, which requires a different set of tools than ones developed for continuous diffusion. While these domains have historically been dominated by autoregressive models [1], there has been some promising recent work in adapting diffusion models to these discrete spaces. These models come with several benefits, like being able to trade off compute and sample quality at inference time by changing the number of sampling steps.

In this work, we investigate the efficacy of this discrete diffusion paradigm on the task of generating images by treating images as a sequence of discrete tokens, an approach that is becoming increasingly favored [11] because it unlocks the dominance of autoregressive language models.

VQ-VAEs offer an amenable approach to tokenizing images: an encoder learns to encode images into latent space; latent embeddings go through a nearest-neighbor lookup to one of K discrete embeddings; finally, a decoder recon-

*Not enrolled in CS 231N.

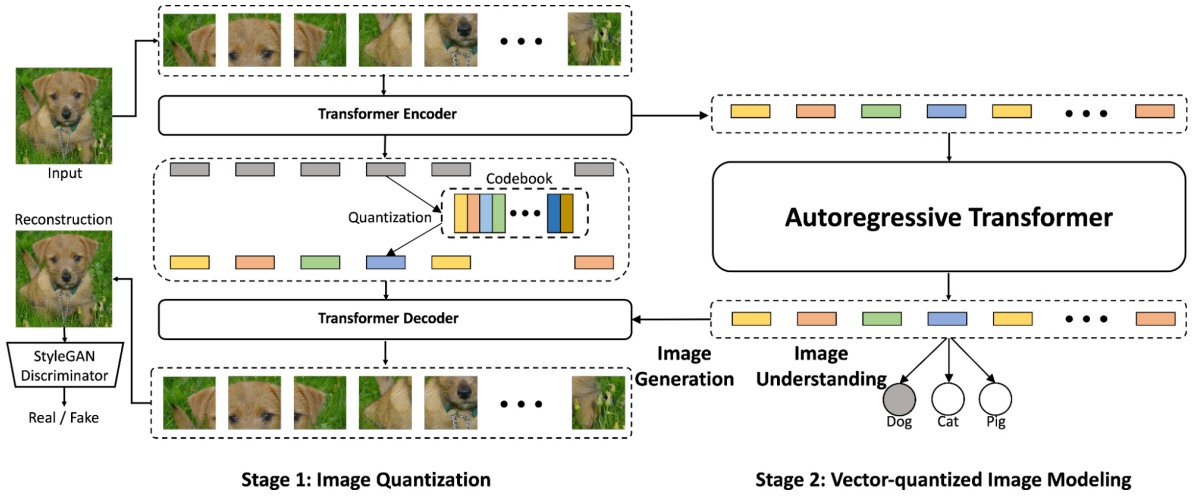

Figure 1. VQGAN architecture. In our approach, we replace the autoregressive transformer with a discrete diffusion model.

structs the image based on the tokens [2, 10]. This reduces the problem to predicting the tokens in latent space, which is now a discrete distribution. Here one could use an autoregressive model, which leads to ViTs (Vision Transformers — see Figure 1, taken from VQGAN [10]); we instead propose using a discrete diffusion model.

2. Related work

The idea of tokenizing images and training a model to predict tokens is not new, stretching back to the original VQ-VAE from 2017 [9]. However, while many attempts have been made to create suitable models to generate image tokens, few can compete with continuous diffusion models, which have reigned supreme in image generation for the past few years.

An early influential work is MaskGIT, introduced by Chang *et al*. in 2022 [2]. They observe that existing autoregressive approaches still naively view images as flattened sequences of tokens, generating in a left-to-right, top-to-bottom manner. In their new proposed paradigm, a transformer is instead trained to predict masked tokens using bidirectional attention, much like the masked language modeling objective in BERT. During inference time, they use a novel parallel decoding method where the model starts with a blank (masked) image and samples all tokens simultaneously, keeping the most confident ones and masking the rest, until no more masked tokens remain. The training procedure — where the model learns to "denoise" masked tokens — and sampling procedures — where the image starts off fully masked/noised and is gradually coaxed into a realistic image — are very similar in nature to diffusion, and both MaskGIT and our proposed model share the same tokenization step (essentially using a pretrained VQ-VAE), making it a good baseline.

Recent work by Yu *et al*. [11] has showed that even under the autoregressive paradigm, the power of LLMs is enough to beat diffusion with a good enough tokenizer. Notably, using some newer quantization techniques, they train a tokenizer for both videos and images (which outperforms the best existing tokenizers in compression) using a vocabulary size of $2^{18} \approx 262,000$, a huge increase over MaskGIT (vocab size 1024). Their model, MAGVIT-v2, achieves SoTA results in several standard video and image benchmarks (ImageNet and Kinetics).

Turning to discrete diffusion, Austin *et al*. [1] were among the first to demonstrate the potential of discrete diffusion processes. They generalize earlier results in discrete diffusion by introducing structured ways to corrupt the data in the forward process, which essentially correspond to different Markov transition matrices (Gaussian, uniform, absorbing). Additionally, they introduce better noise schedules and a new hybrid loss that combines ELBO with cross-entropy that helps stabilize training. Empirically, however, their methods still lag considerably behind transformers on text generation (even on the relatively small text8 dataset) and continuous diffusion models on image generation on CIFAR-10.

More recently, Lou *et al*. [6] bridges the gap between discrete diffusion models and autoregressive models by introducing a novel "score entropy" loss (see **Methods** section for more details) that allows the score matching framework from continuous diffusion models to be extended to the discrete setting. Specifically, instead of parameterizing gradients, they parameterize the ratios $\frac{q_t(y)}{q_t(x)}$ (which can be thought of as the discrete analogue of the gradient). After proving some theoretical results, they demonstrate that this new model (called SEDD — Score Entropy Discrete Dif-

fusion) is competitive with autoregressive language models (GPT-2) in terms of perplexity, even beating GPT-2 on several datasets.

3. Methods

Building on [6], the goal for our model will be to learn the discrete analogue of the score (gradient of the log likelihood), which is the ratio of likelihoods between all pairs

$$s_\theta(x,t) \approx \left[\frac{p_t(y)}{p_t(x)}\right]_{y\neq x}.$$

(Intuitively, in the continuous setting, having access to the score allows us to simulate the reverse process via a random walk). The *score entropy* loss of a distribution p with parameterized concrete score $s_\theta(x)$ is (naively)

$$L_{SE} = \mathbb{E}_{x\sim p}\left[\sum_{y\neq x}\left(s_\theta(x) - \frac{p(y)}{p(x)}\log s_\theta(x)\right)\right].$$

In practice, we need to modify the loss slightly to make it tractable. Specifically, we need to add a constant to each term to make it nonnegative, and replace $p(x)$ by $p(x|x_0)p(x_0)$ (because p_t is intractable for arbitrary t), which is reasonable in a diffusion setting where we are gradually adding noise. We also consider ratios between pairs which differ at only one token, rather than all pairs $y \neq x$, in order to avoid exponential blowup in sequence length.

Finally, skipping over some other minor technical details for brevity, we can integrate over t to get an ELBO that we can then optimize over during training.

The upshot is that we have an objective function tailored for discrete diffusion that is computationally tractable as well as scalable. Furthermore, it can be proven that matching the score exactly recovers the true data distribution.

For the transition matrix Q_t in the forward Markov process, we test both a uniform matrix (where tokens can "jump" to any other token with equal probability) and an absorbing matrix (where tokens either stay or jump to an absorbing state with some probability), both with a logsin noise schedule. Different transition matrices, along with noise schedules (*i.e.* dependence on t), could be one avenue of further exploration.

3.1. Reverse process

Once we have the ratios $s_\theta(x,t)$, we can solve the reverse SDE

$$\frac{dp_{T-t}}{dt} = \overline{Q}_{T-t}p_{T-T}$$

where $\overline{Q}_{T-t}$ is some matrix depending on s_θ and Q_t. (We omit some technical details regarding solving this SDE for brevity; see [6] for a complete explanation.)

There are also a number of tricks we implement to boost sample quality: Classifier-free guidance [5] learns both conditional scores $\frac{p(y|c)}{p(x|c)}$ as well as unconditional scores $\frac{p(y)}{p(x)}$ by randomly dropping the class label with some probability during training, and then combines them at sampling time using

$$s_\theta(x,t) = \frac{p(y|c)}{p(x|c)}^{1+w}\frac{p(y)}{p(x)}^{-w}.$$

w is a hyperparameter that controls the tradeoff between fidelity and diversity.

Temperature sampling is a technique that has been widely adopted by autoregressive models which can also be adapted to discrete diffusion models which essentially scales the distribution to push the model to sample from higher-probability regions.

4. Experimental setup

Code for SEDD [6], which applies discrete diffusion to language modeling, was provided by Aaron Lou as a starting point for much of the experiments.

We test three different variants of SEDD:

- Absorb: tokens are randomly set to a special mask token (in similar fashion to MaskGIT).
- Uniform: tokens randomly jump to other tokens with uniform probability.
- Absorb-2D: same as Absorb with 2D rotary position embeddings (RoPE); see [3].

For baseline, we use MaskGIT [2], which essentially uses a masked language model (rather than autoregressive) to predict tokens, as it offers the most high-level similarity to our approach on a similar scale while using a fundamentally different mechanism. We also include results from VQGAN and LDM [8] (a continuous diffusion model) for posterity. We compare on the Fréchet Inception Distance (FID) metric (similar to perplexity for natural language generation).

4.1. Dataset

We use the ImageNet dataset, which consists of 1.3M training examples, 50K validation examples, and 100K test examples. Images are randomly cropped to 256×256 and randomly flipped horizontally.

4.2. Tokenizer

For the tokenizer, we use a pretrained release of VQGAN [10], which holds SoTA results among open-source VQ-VAEs (and is also the same tokenizer used by MaskGIT [2]).

4.3. Model architecture

We base our model on the diffusion transformer [7], which is more or less a standard encoder-only transformer that also takes in time as input and predicts the score $s_\theta(x, t)$. Specifically, our model consists of

- An embedding layer which processes the class label as well as time, and additionally computes rotary position embeddings,
- 24 encoder layers with layer norm and dropout, and
- A final fully-connected layer.

We use the Adam optimizer, with model and optimizer hyperparameters chosen based on similar work in existing literature.

For each proposed variant, we train a 186M parameter model for $\sim$1.3M iterations.

5. Experiments

Due to time and compute constraints, we present results for a checkpoint of the model trained for $\sim$400K iterations using an absorbing transition matrix and $\log \sin$ noise schedule (see [2] for comparison of noise schedules). Further work could also involve measuring Inception Score (IS) and precision/recall. See Figure 2 for a random snapshot of Absorb generated samples at $\sim$400K iterations.

Model	FID (↓)	# params
VQGAN [10]	15.8	1.4B
MaskGIT [2]	6.2	227M
LDM-8 [8]	15.5	395M
Absorb	16.2	186M
Uniform	20.7	186M
Absorb-2D	14.4	186M

Table 1. Evaluation of image generation samples using Imagenet 256×256 at $\sim$200K iterations.

6. Conclusion and future work

While our FID scores are worse than baselines MaskGIT and VQGAN, we expect this is due to unfinished training (we expect quality to increase $\sim$1.3M iterations), as well as the relatively small parameter count (which was chosen due to compute constraints). However, our results demonstrate that discrete diffusion is a promising approach, and may very well be competitive when scaled up appropriately.

References

[1] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg. Structured denoising diffusion models in discrete state-spaces. *CoRR*, abs/2107.03006, 2021.

[2] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[3] B. Heo, S. Park, D. Han, and S. Yun. Rotary position embedding for vision transformer, 2024.

[4] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[5] J. Ho and T. Salimans. Classifier-free diffusion guidance, 2022.

[6] A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2024.

[7] W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021.

[9] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. *CoRR*, abs/1711.00937, 2017.

[10] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. Vector-quantized image modeling with improved VQGAN. *CoRR*, abs/2110.04627, 2021.

[11] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, V. Birodkar, A. Gupta, X. Gu, A. G. Hauptmann, B. Gong, M.-H. Yang, I. Essa, D. A. Ross, and L. Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024.

Figure 2. ImageNet 256×256 samples generated by Absorb.