MA279 Group 7 - Diffusion Models

ptomalak, ndullam, wang5028

November 2022

Contents

1 Introduction

1.1 What are Diffusion Models?

For a human, to be good at some craft, it takes a lot of time and effort to build up this particular skill. Until recently, computers were very good at following a specific list of pre-programmed steps to achieve a specific goal. Now, thanks to diffusion models, computers can attempt to 'learn' what it takes to reach a certain outcome, starting from scratch. This enables greater flexibility of inputs, and given enough references from which it can learn, a far better outcome than any previous attempts to teach computers how to learn. Diffusion models are most commonly used in image generation, where the model is given a vast amount of reference images from which it learns the meaning behind each prompt and can later use this knowledge to come up with a coherent image based on a new prompt combination.

Diffusion models come from the realms of machine learning and thermodynamics. They take inspiration from thermodynamic diffusion processes and combine them with neural networks. However, they are not (at least primarily) used in physics. As the researchers found out they are incredibly useful in the field of image manipulation. Diffusion models, more explicitly, are a form of generative model, making them great for this kind of workload – they work best by taking sample data and finding the best approach to recreate it starting from scratch, recording all the steps along the way. Where the power comes in is the ability to assess and combine steps, using neural networks to evaluate how steps should be weighed against each other to create something unique. In our analysis, we decided to focus on diffusion models in context of the image generation, where anyone can create a unique image simply based on an input sequence of words. We talk more about it in the Preamble of the Ethics section and we describe how this tool is used and what are its benefits or downsides in the next Ethics subsections.

1.2 The Process

To create the diffusion model, first, we need to create a degraded data point from which the model can get information on how to reverse it to the original form. A data point, in this case, can be considered a series of steps to get from pure Gaussian noise, to a reference image, attached to a specific key/prompt – something we'll touch more on as we continue. However, we do not teach the model how to faithfully recreate the trained images. We have to 'feed' it hundreds of images of certain objects so that it can learn the idea of an item, not just how one particular item looks. This training process creates a database of transformations - we can think of these steps as vector fields, that can be applied to completely random starting data and move points within the random data according to these pre-learned patterns. We attach those learning patterns with a key – a prompt describing the original data point, like 'dog' or 'tree' – so when the model is given a new prompt it can find learning patterns with similar keys. Then, the model applies the related learning models accordingly, based upon the weighted similarity of the prompts, to generate a final result.

1.2.1 Example

Let's assume a scenario where we train a diffusion model on 1,000 images of different puppies. It degrades the images with noise and saves combined transformations of steps that best reverse the degradation and saves it with the keywords (some might have also different keys, like 'red' or 'sleepy'). Now, when we ask it to create a 'very happy puppy sleeping on the bed', it merges all the transformations related to the input prompt, so it merges the idea it has of 'very,' 'happy,' etc... and includes how important certain ideas are to the generation (objects can be more important than adjectives). In other words, it merges all the transformations for all the keys available in the model that the prompt ask for. Starting with an image of pure Guasian noise, we can then apply the appropriate transformations and weights – allowing us to reverse the noisy pattern into a new image, being the result of the 'idea' the diffusion model has from processing the input images. In the end, creating a unique image of a new puppy that is not present in any of the 1000 input images.

1.2.2 Forward Diffusion - Breaking the Data Apart

The process of forward diffusion is based on injecting an increasing amount of Gaussian noise into a given image. We start with the original image, and each step makes it more and more degraded. In the end, we are left with data nearly representative of pure Gaussian noise. We save each of those data points for further processing by the next step.

1.2.3 Reverse Diffusion - Creating Learning Patterns

In reverse diffusion we get the data points from the previous process, placing all of them in pairs after n and $n+1$ steps of degradation. Then we see how the image changed after adding the noise and see what transformation would best reverse the image from $n+1$ to the n step. We do it for all data point pairs and save the resultant transformations. We can think of those transformations as some sort of vector or gravitational force maps that move groups of 'particles' closer and closer to the state before applying any noise. Then we scale the transformations, including different weights for increasing n steps. This means that the final transformation would have the sub-transformations that are from the furthest time-step with a very small weight to gently guide the noisy 'particles' into a better spot, and the final sub-transformation will have a higher weight to lock the particles in the correct place.

1.2.4 Breaking Apart the Prompt and Using Score Networks

As noted above, we've since applied forward and reverse diffusion to some data points; but we must also note, that for diffusion models to work it is also important to store a key, or prompt, for every data point. In this case, the key is an English word or sentence. For example, we can process data with keys like: 'orange cube', 'tasty apple', and 'enormous tree'.

Now, let's assume we've learned some transformations based on each of the aforementioned keys. This trains the 'idea' behind individual words that are included in the model. Next, we get an input from the user, which is also an English word or sentence; in other words, a prompt. For example, if we trained the diffusion model to generate some transformations for concepts like 'big', 'red', and 'mug', we can input the prompt 'big red mug', break apart the given prompt to retrieve their respective transformations and combine those transformations with weights from the prompt - essentially, how important we think each word, or transformation, is to get the best result. In this case, we could add more weight to the word 'mug', then 'red', and elicit the least importance for 'big'.

The process of choosing the appropriate weights for each word in the prompt is the role of score networks (a form of neural network) that are trained to produce the best resultant image and require training separate from the forward and reverse diffusion processes.

1.2.5 What do we mean by transformations? How does the model learn?

Whenever we mentioned transformations we simplified their function to some sort of vector field that moves a group of 'particles' as if it was affected by some force. In truth, they are much more complex and we used this simplification as an approximation of the real process to help understand the process in a bigger picture before delving into the details.

These transformations, in reality, are neural networks used to approximate the conditioned probability distributions in the reverse diffusion process so we can know in which direction a group of 'particles' in a certain area should move at a certain step of the process. So, after forward and reverse diffusion we end up with a massive database of statistical information and probabilities, from which we can extract the information on where we should place a certain object and what combination of transformations has the greatest probability to achieve the best result. The idea behind it is that we want to minimize the loss that the reverse diffusion introduces by reversing the forward diffusion, compared to the original step. We introduce neural networks – a means of score modeling because of the fact that it bypasses the need for training explicit log-likelihoods which are difficult to compute. We include the paper that breaks down this process in the sources, as the detailed explanation of this process would require a separate article and we break down only the forward and reverse diffusion in the math section. A summary of this step was selected from the paper "Maximum Likelihood Training of Score-Based Diffusion Models" by Song, Durkan, Murray and Ermon.

"This gradient field can be efficiently estimated by training a neural network (called a score-based model [44, 45]) with a weighted combination of score-matching losses [23, 56, 46] as the objective. A key advantage of score-based diffusion models is that they can be transformed into continuous normalizing flows (CNFs) [6, 15], thus allowing tractable likelihood computation with numerical ODE solvers."

This can be summarized as using score networks to exchange time-consuming computation into one that is easier and still provides a relatively good approximation. We encourage everyone to delve into the sources and take a look

at the explanation of this process there.

Summing up, neural networks are used to learn the relationship between the extracted features and the content of the images and to create weights for words in the prompt for the model to follow. The neural network is also used to make the generated images more realistic and to add additional details and textures to the images at the end of the process (tweaks to improve the final result). In fact, the learning that results from forward and reverse diffusion is a massive database of statistical information, from which we can extract the information on where we should place a certain object and what combination of transformations has the greatest probability to achieve the best result. In other words, this creates a massive library of instructions (probability of result given we do x) on how to manipulate and create new data. To get and use this information we use log-likelihoods, which for a dataset this size is extremely computationally expensive. Here we also use machine learning to come up with a good approximation of this process that is completed in a feasible timeframe. There is no singular, colossal neural network that takes care of everything, rather the learning is saved as a database that can be later used by the neural network, handling the prompts. The process of 2 colossal neural networks updated on each iteration was implemented by GANs, the predecessor of diffusion models, and it yielded far inferior results.

1.3 Ethical Implications of Diffusion Models

Diffusion models have a few niche applications in the realm of machine learning and generative models, primarily revolving around the principles of image generation. This niche, alongside the general behavior of the model, has led to a few overarching questions regarding the ethical implications of these applications, and how some have or look to address them. One of the most direct applications would be that of 'Deep Fakes', or generating artificial images based upon a real-life person or figure. However, paired with the generation of other people or figures, the intellectual property of the resulting images must be considered. Since diffusion models, and most generative models, can be based on crowd-sourced images, code, or broadened datasets, the question arises of who owns the resulting product. Is the resulting image considered uniquely creative, and if so, who can claim it as intellectual property? Finally, as with most machine learning and AI models, we must note the ethical implications surrounding the environmental impact of the processes; both in the realm of the hardware requirements to run such intensive operations, and the energy requirements backing such through extensive training processes.

2 Math

2.1 Forward Diffusion [q] (Encoding)

$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \mu_t = \sqrt{1-\beta_t}x_{t-1}, \sum_t = \beta_t I)$

- β_t: Variance at time step t; $\beta_t << 1$
- $(x_1, x_2, ..., x_t)$: Sequence of noise samples; latent variable to $(x_0, x_1, ..., x_{t-1})$
- μ_t: Accumulation of variance
- I: Identity matrix
- $\mathcal{N}$ - normal distribution

Forward diffusion is a stochastic process (well described by a random probability distribution) that describes how a variable changes over time. In this process, the probability of a variable x_t at time t is given by a normal distribution with mean μ_t and variance $\sum_t$. The mean μ_t is determined by the previous value of the variable, x_{t-1}, and a parameter β_t that represents the amount of change in the variable.

The forward diffusion process is related to Markov chains, which are stochastic processes that are memoryless, meaning that the current state of the process only depends on the previous state and not on any previous states. In the forward diffusion process, the mean of the normal distribution is determined by the previous value of the variable, which makes the process a Markov chain.

Generally, A Markov chain is a mathematical system that undergoes transitions from one state to another according to certain probabilistic rules. The transition probabilities are represented by a transition matrix, and the behavior of the system over time can be modeled using a state transition diagram. A simple example of a Markov chain is a two-state system where a person is either healthy or sick, with transition probabilities of 0.9 for staying healthy and 0.1 for getting sick, and 0.5 for staying sick and 0.5 for recovering.

$$P = \begin{pmatrix} 0.9 & 0.5 \\ 0.1 & 0.5 \end{pmatrix}$$

The long-term behavior of the system is that the person will mostly remain healthy, with occasional periods of sickness followed by recovery.

The variance of the normal distribution in the forward diffusion process is determined by the parameter β_t. This parameter represents the amount of change in the variable, and a higher value of β_t indicates a higher amount of change, resulting in a higher variance in the distribution.

A variance schedule is a method for calculating the variance of a statistical sample. Variance is a measure of the spread or dispersion of a set of data, and is calculated as the average squared deviation of each data point from the mean. To create a variance schedule, the first step is to calculate the mean of the sample. Then, for each data point, the deviation from the mean is calculated and squared. These squared deviations are added to the variance schedule, along with the total squared deviation and the variance.

Example:

Data point:	3	5	7	9
Deviation from mean:	-3	-1	1	3
Squared deviation:	9	1	1	9
Total squared deviation:	20			

Variance: $20/4 = 5$

$\mathcal{N}$ represents the normal distribution. The normal distribution is a continuous probability distribution that is defined by its mean and variance. In this equation, the normal distribution is used to model the probability of x_t given x_{t-1}. The mean of the distribution is $\mu_t = \sqrt{1-\beta_t}x_{t-1}$, and the variance is $\sum_t = \beta_t I$, where β_t is a parameter and I is the identity matrix, so the probability of x_t given x_{t-1} is modeled by a normal distribution with a mean that is a function of x_{t-1} and a variance that is determined by the parameter β_t.

Summing up, forward diffusion uses a Markov chain process in which the time step t only depends on the previous step $t-1$. The variance β_t strictly increase from $0 \to 1$. This process makes the μ_t approach 0 since $\sqrt{1-\beta_t}$ approach 0 as β_t approaches 1. β_t, which is the accumulation of variance, approaches 1 as t increases until T, which the original image is turned into complete noise. Therefore x_t approaches a $\mathcal{N}(0,1)$ distribution (Gaussian) at the end of the process.

2.2 Reverse Diffusion [p] (Decoding)

$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \wedge p_\theta(x_{t-1}, x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t))$

- p_θ: Learned model to describe conditional distribution since each reverse step is mapped using the model of the entire reverse diffusion process.
- $p(x_t)$: Pure noise distribution $p(x_t) = \mathcal{N}(x_t; 0, I)$
- $\sigma_\theta(x_t, t))$is set to be a time dependent constant
- $\mu_\theta(x_t, t)$ Predicted by the learned model

Summary: In reverse diffusion, we reverse the forward diffusion process and try to recreate the sample from Gaussian noise. This will also approach $\mathcal{N}(0, I)$ isotropic normal distribution if n is small enough. We write this as joint steps of conditionals. We start from the complete Gaussian noise and sample through each learned reverse process steps until producing x_0. It is important to use very small sizes for reverse diffusion as the model will determine the steps tracing back to the original image much more accurately. The second equation is a learned neural network to describe the diffusion process.

Both formulas calculate the probability of observing a sequence of random variables over time by multiplying the probability of observing the final variable by the product of the probabilities of observing each variable given the previous variable. The parameter θ represents the parameters of the probability distributions used to model the sequence of random variables. This formula is used to estimate the distribution of a random variable over time and make predictions about the future values of the variable.

The variable μ represents the mean of the Gaussian distribution, which is the average or expected value of the data points in the distribution. The variables θ and t are parameters that define the Gaussian distribution and its mean, respectively.

The expression $\mu_\theta(x_t, t)$ indicates that the mean of the Gaussian distribution is a function of both the data point x_t and the parameter t. In other words, the mean of the Gaussian distribution is dependent on both the data point and the parameter, which allows for a more flexible and nuanced model of the data.

The formula for a Gaussian distribution is:
$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
where x is a real-valued data point, μ is the mean of the distribution, σ is the standard deviation of the distribution. The formula describes the probability of a data point x occurring in a Gaussian distribution with a particular mean and standard deviation. The probability is represented by the height of the bell-shaped curve at the point x.
The formula has a number of key components. The first part, $\frac{1}{\sqrt{2\pi\sigma^2}}$, is a normalization factor that ensures that the probabilities of all the data points in the distribution sum to 1. The second part, $e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, is the core of the formula, which describes the shape of the bell-shaped curve and determines the probability of a data point x occurring in the distribution.

3 Ethics

3.1 Ethics Preamble

Diffusion models are relatively recent phenomena, having their roots in non-equilibrium thermodynamics - the process of thermodynamic diffusion. Surprisingly, diffusion models work similarly but do not apply to describing physical phenomena. Researchers found out that they are incredibly useful when used with images. By combining the physical process of diffusion with machine learning (score networks), they created many useful tools that can be applied to images and used for image generation.

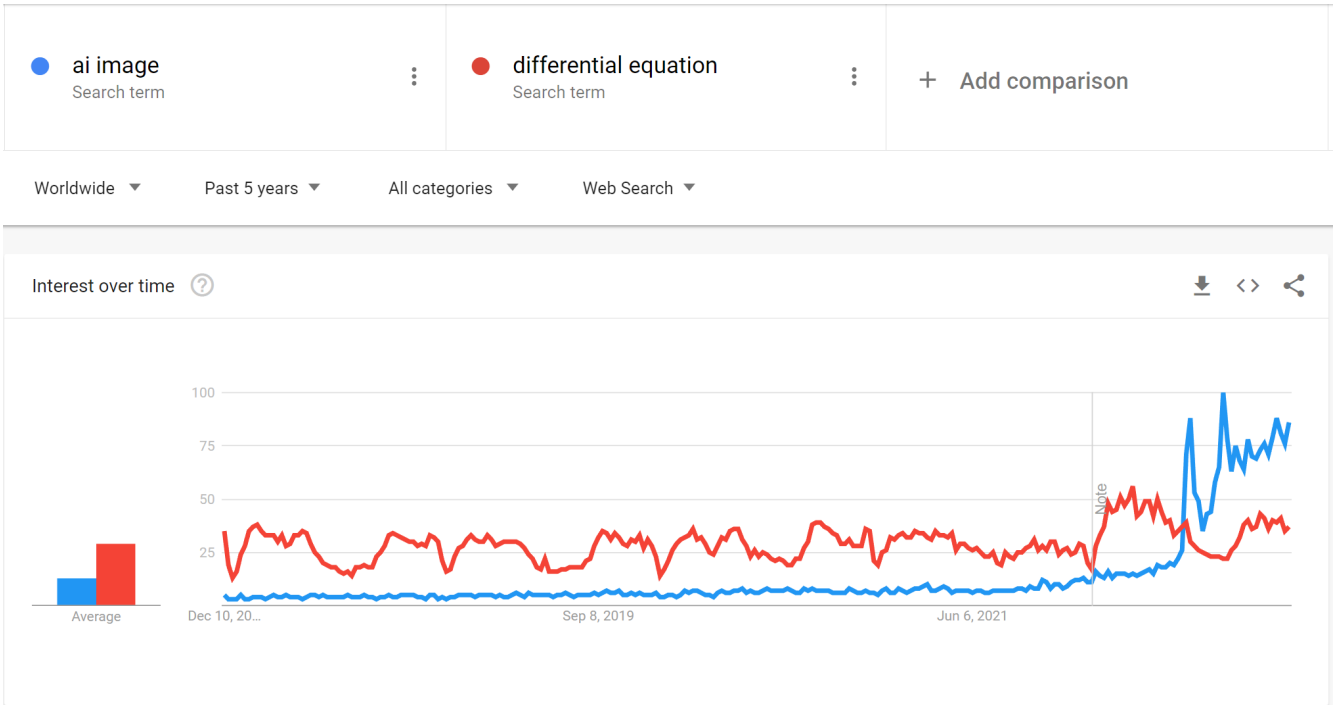
The main tools we are able to currently distinguish:

1. Image denoising - the model takes the dataset of noisy and reference images and learns how to best reverse the noise. It is useful in rendering 3D images using a ray-tracing process, we can finish the render faster when the noise is still noticeable and approximate the denoised final image using a diffusion model, speeding up the job exponentially.

2. Super-resolution imaging - the model takes the dataset of shrank down and reference images and learns how to best upscale it, losing the least amount of details. It is useful to obtain a decent upscaling approximation of the images for which better resolution does not exist. It's worth noting that this process can be applied to videos as well, with recent developments in this technology being very promising.

3. Image generation and inpainting - the process that is described in the 'Introduction' section. Image inpainting is analogous to generation, with the only difference that the process takes place in a specified region of chosen image, making sure the generated image blends is with the remaining content.

That said we would like to point out that our consideration will focus only on diffusion models that implement image generation. We made this choice for two reasons. First, the other tools that use Diffusion Models are currently used in ways that do not raise many (if any) ethical concerns, they result in a positive outcome that we briefly mentioned above. Second, the image generation encounters a significant and constantly increasing boom in popularity - breaking into the mainstream media and being the topic of vivid debate. We consider this phenomenon to be very rare, for immediate adoption of a scientific breakthrough, hence our focus on this particular tool. To support our claim, according to the popularity in Google trends it became more popular than differential equations over the course of just one year.



Source: [trends.google.com] as for Dec 5th

3.2 Targeted Content

Although, art and prompted generation is the most common use cases of this technology, diffusion models have also proved useful in image-image generation (versus the aforementioned text-image generation). This technology generally implies the use of images as a relative 'prompt' for the generation of another image. In principle, this method of diffusion modeling can be used for developing 'Deep fakes,' or artificial images and videos to falsely represent a person or figure. While this methodology can be used in beneficial means, let's say by remastering older images or movies in higher resolutions or bringing back the legacy of actors who passed away back to the movie screens, it can also be used more explicitly for malicious and exploitative means – leading towards misinformation and potential damage for one's reputation. Not only does it introduce skepticism into the rights to develop such a form of media, but towards whether you own your own representation in artificial media. It's important to point out, that at the current stage, experts from the fields of deep fakes and generative imaging, make effort to create tools that allow for identifying artificial media; enabling disclosures to prevent extensive misinformation. But despite such, one's artificial representation by third parties is an ethical dilemma in and of itself, beyond its identification. Moreover, in the near future, we might face a competition between rapidly evolving diffusion models and tools to identify them with an unclear end result, and countless cases of abuse of this technology along the way.

3.3 Intellectual Property

The principles of generative models are on the basis of the training data with which they're built on. For the model to understand the human world, we first need to 'feed it' with a significant portion of manually generated data. With this property comes the concerns of ownership, and the drives of intellectual property. While some companies, like the team behind the GitHub Copilot, an AI-assisted programming extension that can help in regenerating simple programming tasks, have focused on the use of licensed sources under their own platform, the general consensus behind the model is still speculative with accusations of stealing people's code, and relevant lawsuits underway. As we've noted in the behaviors of diffusion models more specifically, the model is trained on the application and reversal of noise to achieve a similar representation of the reference image – weighing the actions taken to be used in generating a given prompt. This can lead to producing vastly similar results compared to training images. This is particularly visible in cases of over-fitting the model, so giving it directions to be very truthful to the prompt. While a similar result to a specific piece of training data can pose a more obvious ethical concern, given the ambiguous ownership of the model's result, there exists a broadened concern over the remainder of the training data. And that is, the potential influence the remainder of the training data may have had on the resulting model, and how ownership could have been distributed accordingly. While this issue can be further resolved with the use of copyright-free materials and reference images, it still drives the question of whether the model's results are redeemable as unique pieces of work; and if so, who would be the owner of said intellectual property? Would the result of directing the model to generate a combination of "The Great Wave off Kanagawa" and "The Starry Night" could be attributed to Van Gogh, Hokusai or the person coming up with the prompt? Sometimes the answer is not that obvious, like Jason Allen's A.I.-generated work "Théâtre D'opéra Spatial," where coming up with the right prompt could be debated to be an art in itself.

3.4 Environmental Implications

As for the environmental implications of diffusion models and generative modeling as a whole, we must acknowledge both the hardware requirements, paired with the extensive energy requirements for most modern machine learning techniques. First, let's acknowledge the hardware requirements, revolving mostly around GPUs; processing units geared towards optimizing floating point operations, with a more modern focus on machine learning or neural cores. We talk not only about the power to run them, but also the carbon footprint it takes to produce them. While GPU cores are highly optimized for their scope, when it comes to training machine learning models, the work required to run them could be achieved with significantly less electricity given a dedicated hardware solution. Training models, while the performance of the machine of course has an impact on its effective speed, draws lots of power; and can take extended periods of time to complete. Here, not only the training of the model takes a long time and countless processing units, but also generating images from the prompts results in full utilization of even the state-of-the-art hardware. Alongside that note, data processing and training is only one part of the story for effectively creating accurate diffusion models or models in a general sense. Data centers, especially for data as dense as images, are a near requirement, which have their own extensive hardware and energy costs to maintain. And with such, it must be noted that AI and machine learning, and the ethics attached to their environmental implications, must not be taken with a grain of salt when considering the generation of models.

3.5 Art Industry Implications

There are plenty of jobs whose sole purpose is to bring imagination into reality. For example, concept artists, people working in the film industry, or just creating any form of art. Alongside AI-generated models grabbing more public attention, there appeared more artists concerned about the future of their jobs and livelihood. They sacrificed a significant portion of their lives to learn the skill that allows them to translate their thoughts into tangible representation, and now computers can do this work for them, with incrementally improving results. In other words, a sizable portion of the industry feels threatened and uncertain due to the development and use of diffusion modeling. Still, there are some people who oppose this sentiment, claiming that AI art can be a great tool to improve and speed up their work. For example to generate some form of a 'sketch' from which the artist can take over, or come up with inspirational reference images that can help overcome a creative roadblock. Summing up, it's very likely, that from now on artists will need to learn how to use and integrate diffusion models in their workflow in order to stay competitive. The result of this paradigm shift might yield a long-term benefit of increased efficiency, but also impact the creative process and livelihood of people amid the shift.

4 Sources

4.1 Summary

The subsection below contains a list of any sources we used to learn the topic of diffusion models. Some parts of our work may contain elements and reference to such.

4.2 Links

[1] https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

[2] https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/

[3] https://ayandas.me/blog-tut/2021/12/04/diffusion-prob-models.html

[4] https://arxiv.org/pdf/2209.00796.pdf

[5] https://github.com/CompVis/stable-diffusion/blob/main/scripts/sample_diffusion.py

[6] https://github.com/openai/guided-diffusion

[7] https://yang-song.net/blog/2021/score/

[8] https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da

[9] https://theaisummer.com/diffusion-models/

[10] Maximum Likelihood Training of Score-Based Diffusion Models

4.3 Utility Links

[1] Presentation

[2] Notes and Ideas