Image Generation Using Diffusion Models

Shahin Majazi

August 24, 2023

Contents

Abstract

This report delves into the innovative domain of image generation through diffusion models. Traditional approaches to image generation often struggle with capturing intricate details and long-range dependencies present in complex visual content. Diffusion models have emerged as a promising alternative, simulating a continuous noise-spreading process that leads to the creation of coherent and highly realistic images. Inspired by concepts from statistical physics and information theory, diffusion models are equipped with stochastic differential equations, providing a solid mathematical foundation.

The integration of neural networks further augments the capabilities of diffusion models, enabling the generation of images that exhibit both local and long-range correlations. This amalgamation of probabilistic modeling and neural network architecture contributes to the production of high-quality and lifelike images.

The report presents a comprehensive analysis of the reverse and forward processes in diffusion models, shedding light on their distinct parameterizations and their equivalence to denoising score matching. The training techniques employed for diffusion models are discussed, focusing on variance reduction and the optimization of variational bounds.

Results and applications of diffusion models are demonstrated across various domains, including unconditional image generation, text-to-image conversion, super-resolution creation, and more. The provided figures offer a visual representation of the capabilities of diffusion models in generating diverse and high-quality images.

In conclusion, this report showcases the transformative potential of diffusion models in the realm of image generation, highlighting their ability to overcome the limitations of conventional methodologies and produce images that exhibit intricate details and authenticity.

Chapter 1

Introduction

1.1 Background

The field of image generation has seen a transformative shift with the emergence of diffusion models. As traditional pixel-wise sampling and autoregressive methods faced limitations in capturing complex visual patterns and global dependencies, diffusion models emerged as an innovative alternative.

At their core, diffusion models simulate a continuous process of spreading noise over time to create coherent images. This departure from discrete pixel generation allows for the generation of highly detailed and realistic images. Inspired by concepts from statistical physics and information theory, diffusion models employ stochastic differential equations to govern the evolution of noise vectors, providing a solid mathematical foundation.

Incorporating neural networks further enhances diffusion models. These networks, often designed as invertible functions, enable the generation of images that maintain both local and long-range correlations. This integration of neural networks and probabilistic modeling contributes to the high-quality output of diffusion models.

As the field evolves, training techniques for diffusion models continue to advance, addressing challenges posed by the continuous nature of diffusion processes. Techniques like contrastive divergence and maximum likelihood estimation play crucial roles in optimizing model parameters to align generated images with real data distribution.

1.2 Motivation

The impetus to explore image generation through diffusion models arises from the intrinsic shortcomings of conventional methodologies. While conventional pixel-wise sampling and autoregressive models have made strides in producing coherent images, they often struggle to encapsulate intricate nuances and long-range dependencies present in complex visual content. This inadequacy

becomes pronounced when attempting to faithfully replicate real-world scenes that demand a harmonious amalgamation of details.

In response, diffusion models offer a compelling alternative. By embracing a continuous diffusion process, these models inherently mitigate the limitations associated with discrete generation. This facet not only empowers the synthesis of images enriched with meticulous particulars but also facilitates the integration of broader contextual cues, ushering in a more authentic manifestation of the underlying data distribution.

Moreover, the fusion of stochastic differential equations and neural networks within diffusion models furnishes a robust framework adept at navigating the intricate nuances of image generation. This symbiotic relationship between mathematical finesse and neural network ingenuity equips diffusion models to yield images that not only meet but consistently transcend the benchmarks of realism and coherence established by traditional techniques.

The prospective applications of diffusion models span beyond image synthesis, encompassing domains like denoising, inpainting, and even dynamic video creation. This versatility underscores their relevance in tackling multifaceted challenges within the realm of artificial intelligence.

As we plunge into the depths of the impetus driving the incorporation of diffusion models in image generation, it becomes unmistakably apparent that their distinctive attributes and harmonized approach present an exhilarating prospect for surmounting the constraints of existing methodologies. Through this exploration, we aspire to unearth the transformative potency of diffusion models, poised to revolutionize our perception, conception, and interaction with generated visual artistry.

Chapter 2

Diffusion Models

2.1 Simplified Modeling for the Main Problem

Diffusion models are latent variable models of the form $p_\theta(x_0) := \int p_\theta(x_{0:T})dx_{1:T}$, where $p_\theta(x_{0:T})$ is called the *reverse process*, and it is defined as Markov chain with learned Gaussian transitions starting at $p(x_T) = \mathcal{N}(x_T; 0, I)$:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_t - 1; \mu_\theta, t, \Sigma_\theta(x_t, t)) \tag{2.1}$$

What distinguishes diffusion models from other types of latent variable models is that the approximate posterior $q(x_{1:T}|x_0)$, called the $forward\ process$ or diffusion process, is fixed to Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2.2}$$

Training is performed by optimizing the usual variational bound on negative log likelihood:

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] = \mathbb{E}_q[-\log p(x_T) - \sum_{t\geq 1} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] =: L \tag{2.3}$$

The forward process variances β_t can be learned by reparameterization or held constant as hyperparameters, and expressiveness of the reverse process is ensured in part by the choice of Gaussian conditionals in $p_\theta(x_{t-1}|x_t)$, because both

processes have the same functional form when β_t are small. A notable property of the forward process is that it admits sampling x_t at an arbitrary timestep t in closed form: using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, we have

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \tag{2.4}$$

Efficient training is therefore possible by optimizing random terms of L with stochastic gradient descent. Further improvements come from variance reduction by rewriting L as:

$$\mathbb{E}_q[\underbrace{D_{KL}(q(x_T|x_0) \parallel p(x_T))}_{L_T} + \sum_{t>1} \underbrace{D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \underbrace{- \log p_\theta(x_0|x_1)}_{L_0}] \tag{2.5}$$

Equation (2.5) uses KL divergence to directly compare $p_\theta(x_{t-1}|x_t)$ against forward process posteriors, which are tractable when conditioned on x_0:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I), \tag{2.6}$$

$$\text{where } \tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\tilde{\alpha}_{t-1}}\beta_t}{1 - \tilde{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1 - \tilde{\alpha}_{t-1})}{1 - \tilde{\alpha}_t}x_t \text{ and } \tilde{\beta}_t := \frac{1 - \tilde{\alpha}_{t-1}}{1 - \tilde{\alpha}_t}\beta_t \tag{2.7}$$

Consequently, all KL divergences in Eq. 2.5 are comparisons between Gaussians, so they can be calculated in a Rao-Blackwellized fashion with closed form expressions instead of high variance Monte Carlo estimates.

2.2 Forward Process and L_T

We ignore the fact that the forward process variances β_t are learnable by reparameterization and instead fix them to constants. Thus, in our implementation, the approximate posterior q has no learnable parameters, so L_T is a constant during training and can be ignored.

2.3 Reverse Process and $L_{1:T-1}$

Now we discuss our choices in $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$ for $1 < t \leq T$. First, we set $\Sigma_\theta(x_t, t) = \sigma^2 I$ to untrained time dependent constants. Experimentally, both $\sigma_t^2 = \beta_t$ and $\sigma^2 t = \tilde{\beta}_t = \frac{1-\tilde{\alpha}_{t-1}}{1-\tilde{\alpha}_t}\beta_t$ had similar results.

The first choice is optimal for $x_0 \sim \mathcal{N}(0, I)$, and the second is optimal for x_0 deterministically set to one point. These are the two extreme choices corresponding to upper and lower bounds on reverse process entropy for data with coordinate-wise unit variance.
Second, to represent the mean $\mu_\theta(x_t, t)$, we propose a specific parameterization motivated by the following analysis of L_t. With $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$, we can write:

$$L_{t-1} = \mathbb{E}_q[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2] + C \tag{2.8}$$

where C is a constant that does not depend on θ. So, we see that the most strightforward parameterization of μ_θ is a model that predicts $\tilde{\mu}_t$, the forward process posterior mean. However, we can expand Eq. 2.8 further by reparameterizing Eq. **??** as $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ for $\epsilon \sim \mathcal{N}(0, I)$ and applying the forward process posterior formula 2.7:

$$L_{t-1} - C = \mathbb{E}_{x_0,\epsilon}[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t(x_0,\epsilon), \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t(x_0,\epsilon) - \sqrt{1-\bar{\alpha}_t}\epsilon)) - \mu_\theta(x_t(x_0,\epsilon), t)\|^2] \tag{2.9}$$

$$= \mathbb{E}_{x_0,\epsilon}[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\alpha_t}}(x_t(x_0,\epsilon) - \sqrt{1-\bar{\alpha}_t}\epsilon) - \mu_\theta(x_t(x_0,\epsilon), t)\|^2] \tag{2.10}$$

Equation 2.10 reveals that μ_θ must predict $\frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$ given x_t. Since x_t is available as input to the model, we may choose the parameterization

$$\mu_\theta(x_t, t) = \tilde{\mu}_t(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t))) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) \tag{2.11}$$

where ϵ_θ is a function approximator intended to predict ϵ from x_t. To sample $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$ is to compute $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(x_t, t)) + \sigma_t \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The complete sampling procedure, Algorithm 2.2, resembles Langevin dynamics with $\boldsymbol{\epsilon}_\theta$ as a learned gradient of the data density. Furthermore, with the parameterization 2.11, Eq. 2.10 simplifies to:

$$\mathbb{E}_{\mathbf{x}_0,\epsilon}[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x_0} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2] \tag{2.12}$$

: Algorithm 1

repeat
  $x_0 \sim q(x_0)$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\epsilon \sim \mathcal{N}(0, I)$
  Take gradient descent step on $\Delta_\theta \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\|^2$
until converged

(b) Training Algorithm

: Algorithm 2

$x_T \sim \mathcal{N}(0, I)$
for t = T, . . . , 1 **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) + \sigma_t \mathbf{z}$
end for
return x_0

(d) Sampling Algorithm

Figure 2.1: Two Algorithms

which resembles denosing score matching over multiple noise scales indexed by t. As Eq. 2.12 is equal to (one term of) the variational bound for the Langevin-like reverse process 2.11 we see that optimizing an objective resembling denoising score matching is equivalent to using variational inference to fit the finite-time marginal of a sampling chain resembling Langevin dynamics.

To summarize, we can train the reverse process mean function approximator $\boldsymbol{\mu}_\theta$ to predict $\tilde{\boldsymbol{\mu}}_t$, or by modifying its parameterization, we can train it to predict $\boldsymbol{\epsilon}$. (There is also the possibility of predicting $\mathbf{x_0}$ but we found this to lead to worse sample quality early in our experiments.) We have shown that the $\boldsymbol{\epsilon}-$prediction parameterization both resembles Langevin dynamics and simplifies the diffusion model's variational bound to an objective that resembles denoising score matching. Nonetheless, it is just another parameterization of $p_\theta(x_{t-1}|x_t)$, so we verify its effectiveness where we compare predicting $\boldsymbol{\epsilon}$ against predicting $\tilde{\boldsymbol{\mu}}_t$.

2.4 Reverse Process Deconder

We assume that image data consists of integers in $\{0, 1, \dots, 255\}$ scaled linearly to $[-1, 1]$. This ensures that the neural network reverse process operates on consistently scaled inputs starting from the standard normal prior $p(x_T)$. To obtain discrete log likelihoods, we set the last term of the reverse process to an

independent discrete decoder derived from the Gaussian $\mathcal{N}(x_0; \boldsymbol{\mu}_\theta(x_1, 1), \sigma^2 \mathbf{I})$:

$$p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x_1}, 1))\, dx,$$
$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases}, \tag{2.13}$$
$$\delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

where D is the data dimensionality and the i superscript indicates extraction of one coordinate. (It would be straightforward to instead incorporate a more powerful decoder like a conditional autoregressive model, but we leave that to future work.) Similar to the discretized continuous distributions used in VAE decoders and autoregressive models [34, 52], our choice here ensures that the variational bound is a lossless codelength of discrete data, without need of adding noise to the data or incorporating the Jacobian of the scaling operation into the log likelihood. At the end of sampling, we display $\boldsymbol{\mu}_\theta(\mathbf{x}_1, 1)$ noiselessly.

Chapter 3

Results and Applications

3.1 Image Generation, Results, and Applications

This section provides some results of the diffusion models for image generation, text-to-image conversion, super-resolution creation, and other applications of these models.

Figure 3.1: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to righ).

Figure 3.2: When conditioned on the same latent, CelebA-HQ 256×256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x_0}|\mathbf{x_t})$.

Figure 3.3: Interpolations of CelebA-HQ 256×256 images with 500 timesteps of diffusion.

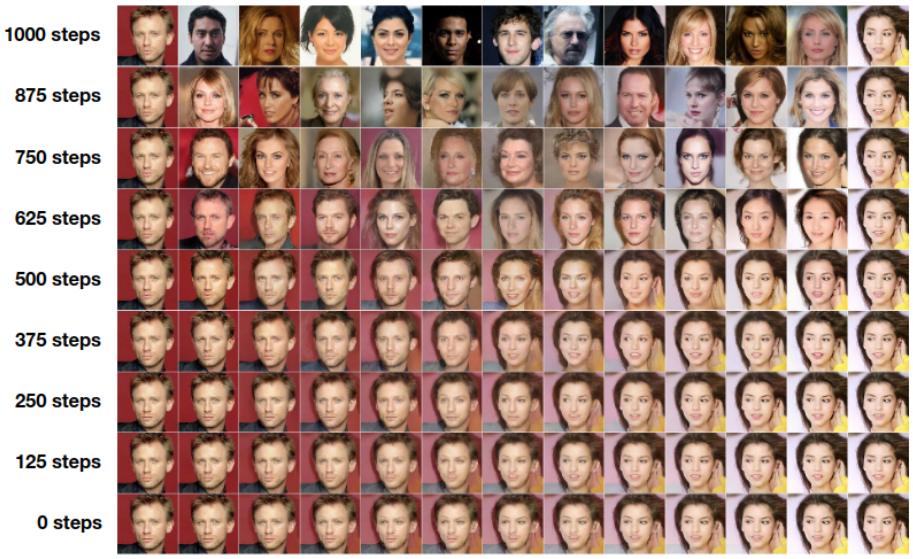


Figure 3.4: Coarse-to-fine interpolations that vary the number of diffusion steps prior to latent mixing.

Figure 3.5: CelebA-HQ 256×256 generated samples

(a) Pixel space nearest neighbors

Figure 3.6: CelebA-HQ 256×256 nearest neighbors, computed on a 100×100 crop surrounding the faces.

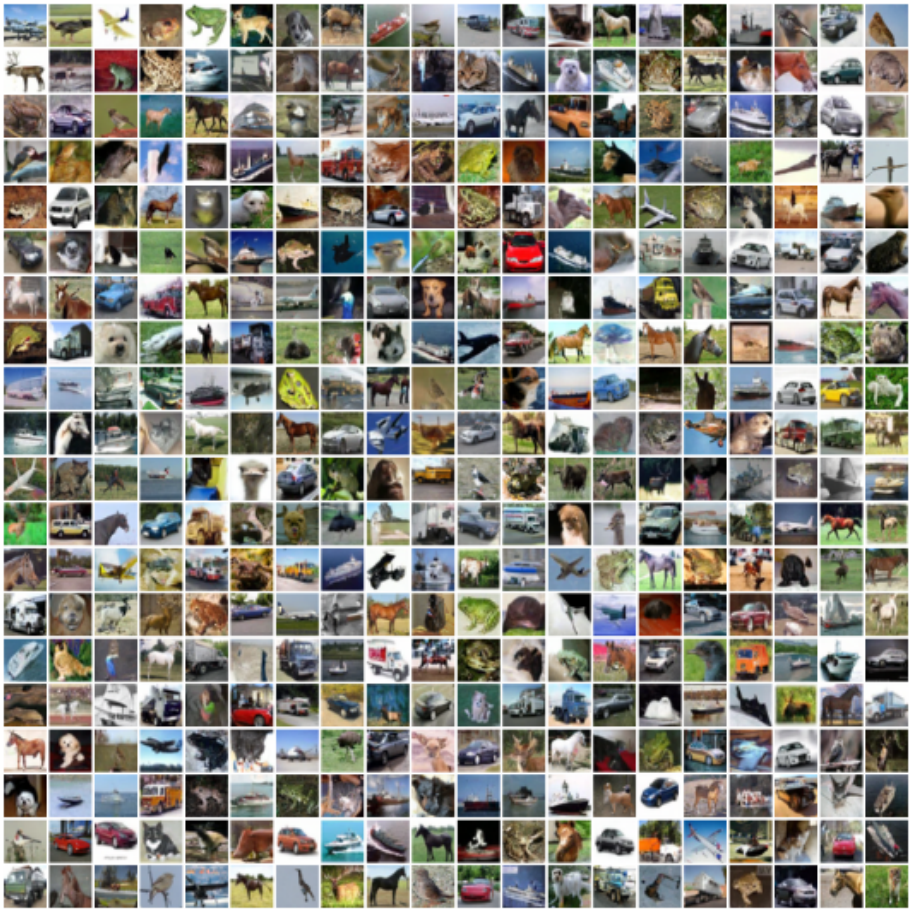


Figure 3.7: Unconditional CIFAR10 generated samples

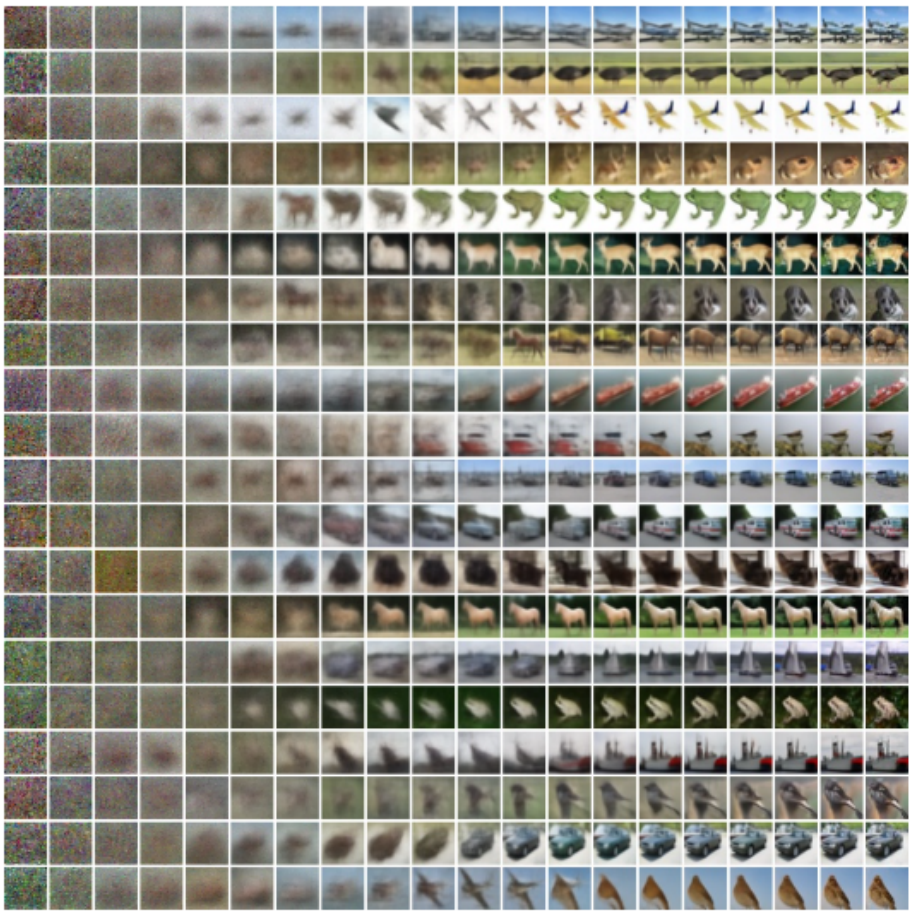


Figure 3.8: Unconditional CIFAR10 progressive generation

Figure 3.9: LSUN Church generated samples.

Figure 3.10: LSUN Bedroom generated samples.

Figure 3.11: LSUN Cat generated samples.

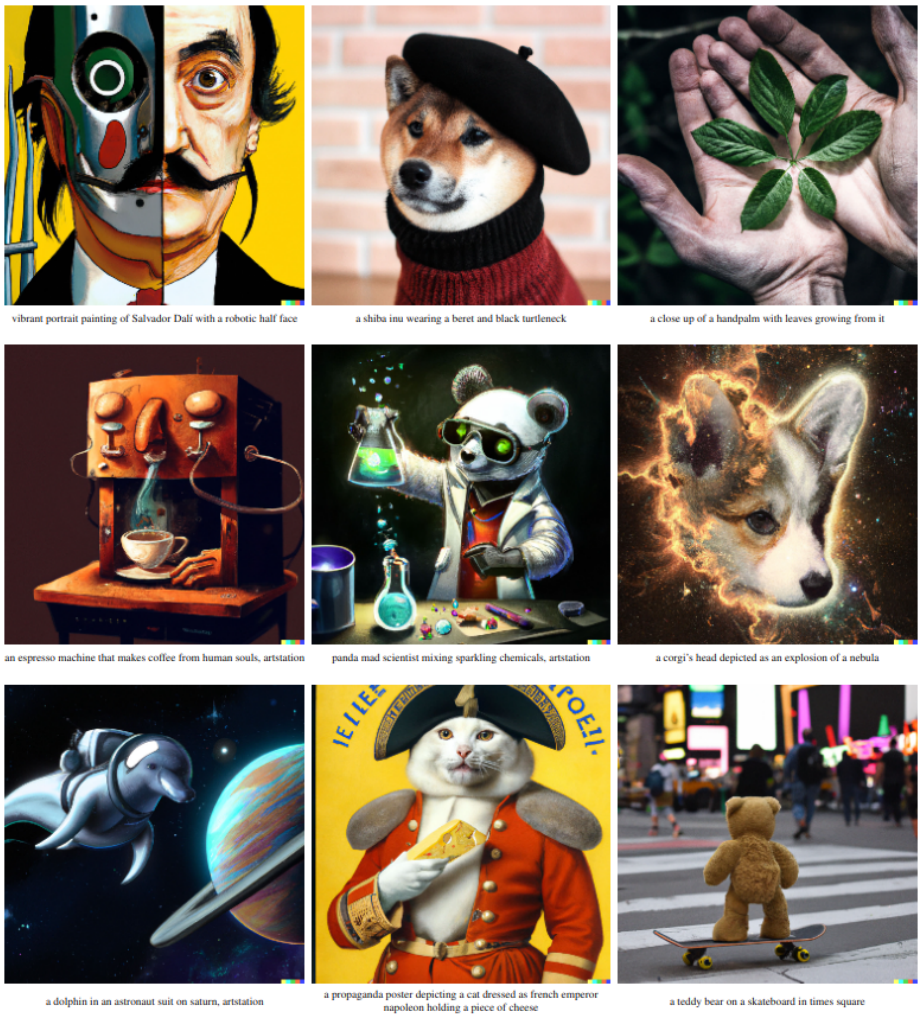


Figure 3.12: Selected 1024×1024 samples from a production version of Open.AI model.

Figure 3.13: Variations of an input image by encoding with CLIP and then decoding with a diffusion model. The variations preserve both semantic information like presence of a clock in the painting and the overlapping strokes in the logo, as well as stylistic elements like the surrealism in the painting and the color gradients in the logo, while varying the non-essential details.

Figure 3.14: Random samples from unCLIP for prompt “Vibrant portrait painting of Salvador Dali with a robotic half face”

Figure 3.15: Random samples from unCLIP for prompt “A teddybear on a skateboard in Times Square.”

Figure 3.16: Select 1024×1024 Imagen samples for various text inputs.

Figure 3.17: Select 1024×1024 Imagen samples for various text inputs.

Figure 3.18: Select 1024 × 1024 Imagen samples for various text inputs.

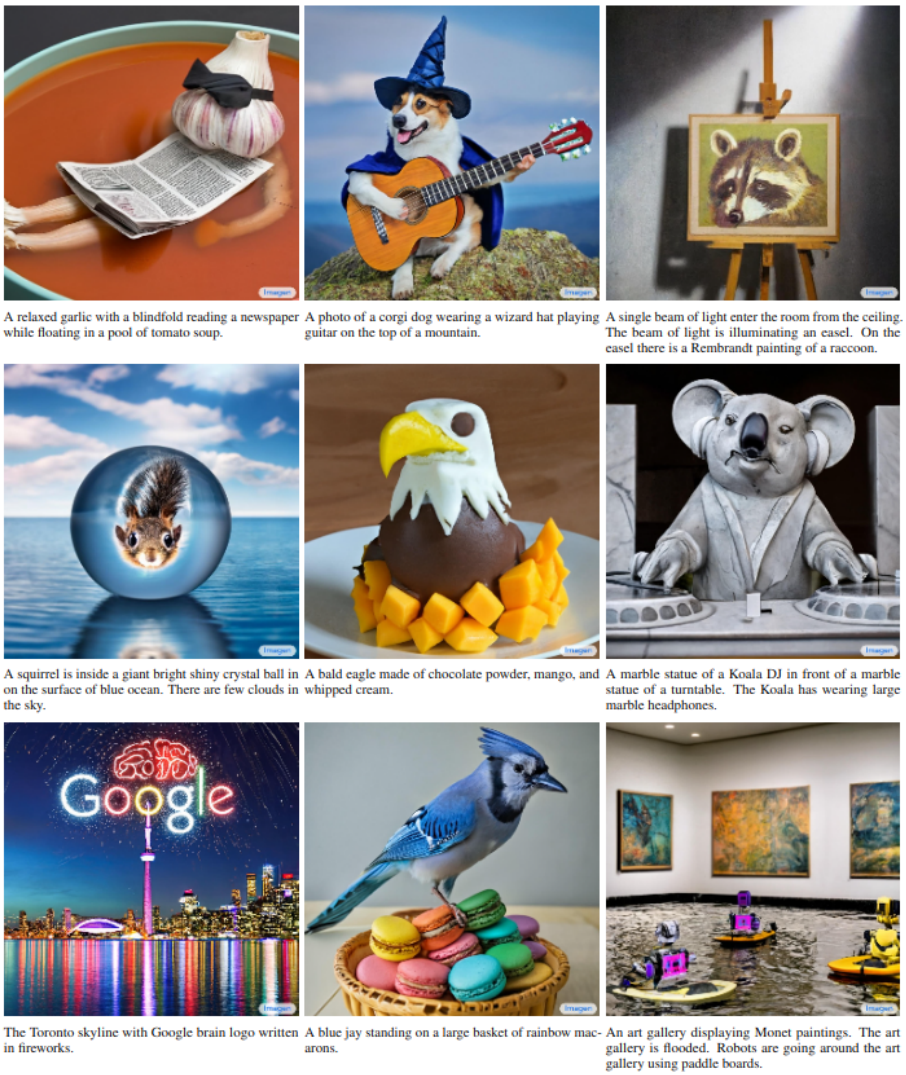


Figure 3.19: Select 1024×1024 Imagen samples for various text inputs.

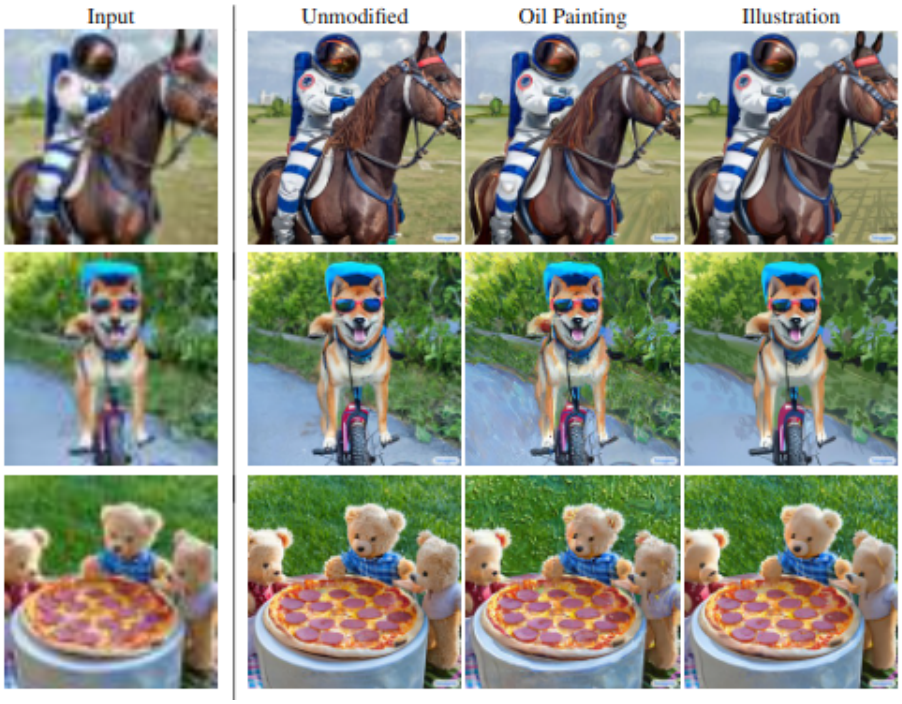


Figure 3.20: Super-resolution variations for some $64 \times 64r$ generated images.

Figure 3.21: Control Stable Diffusion with Canny edge map.

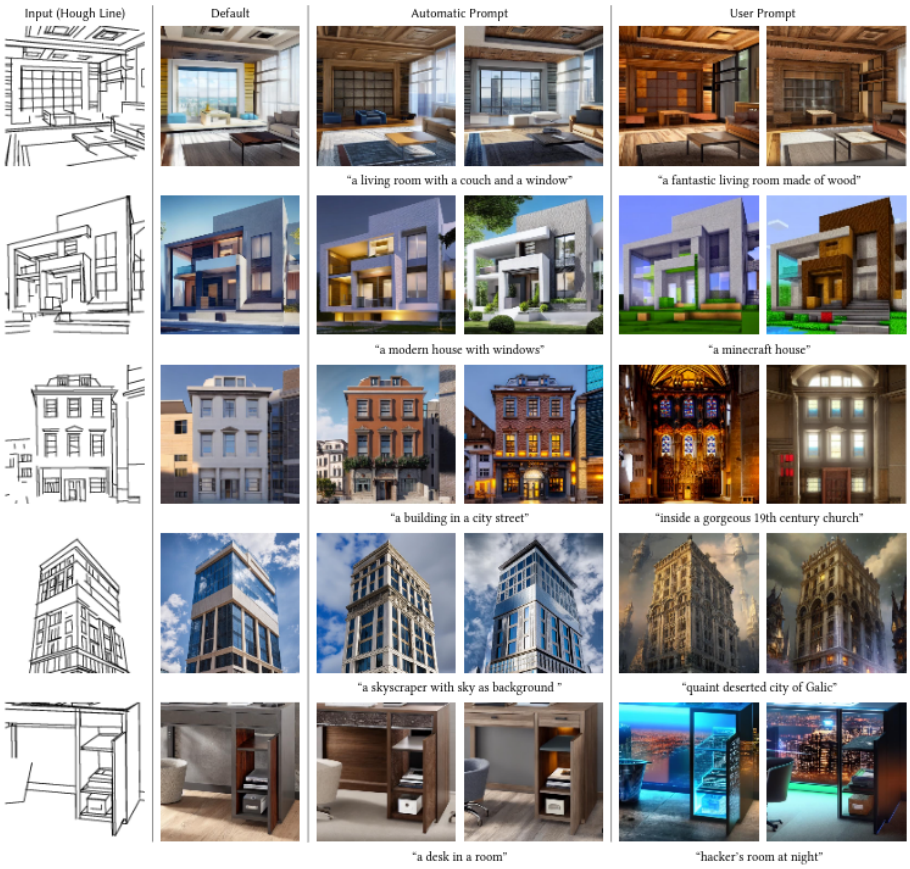


Figure 3.22: Controlling Stable Diffusion with Hough lines (M-LSD).

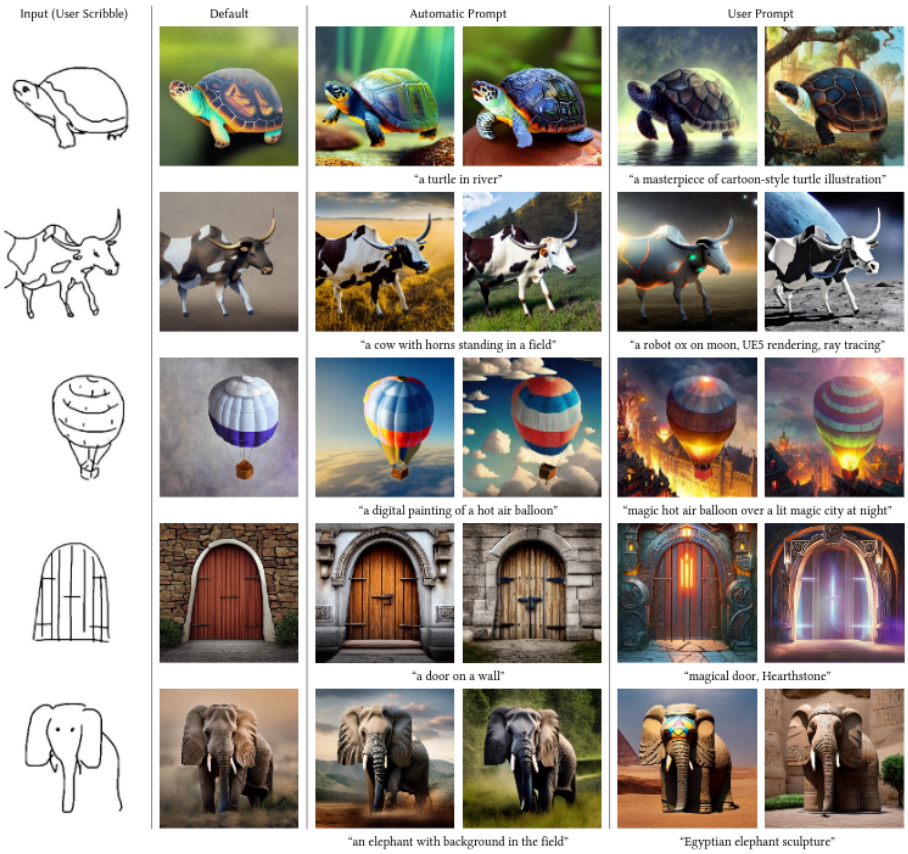


Figure 3.23: Controlling Stable Diffusion with Human scribbles.

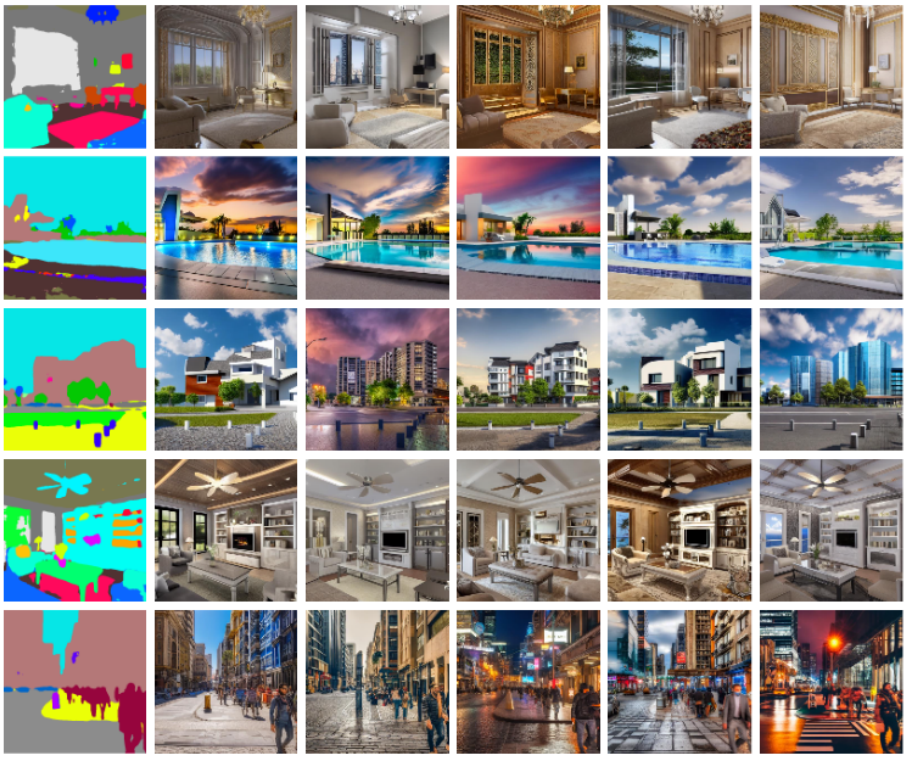
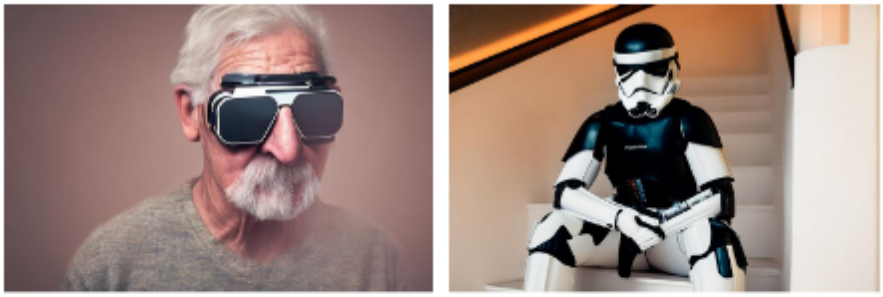
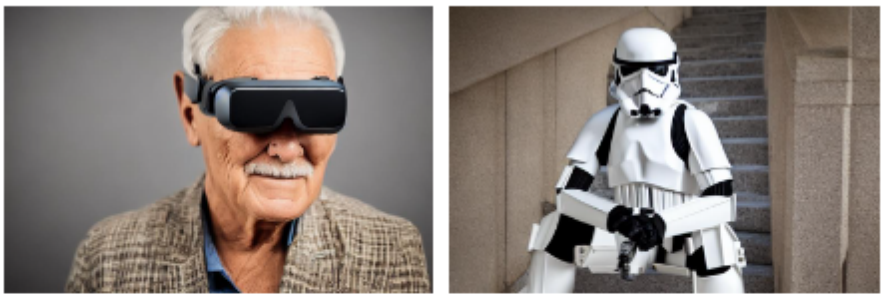


Figure 3.24: Controlling Stable Diffusion with segmentation map

Images and Midas Depth



Stable Diffusion V2 Depth-to-Image
resumed from **SD 2.0**, continued training on **Large-scale Nvidia A100 Clusters**,
more than **12M** training data, **more than 2000 GPU-hours** (estimation)



Stable Diffusion with **Depth-based ControlNet**
controlling **SD 1.5**, trained on **one single Nvidia RTX 3090TI**,
with **200K** training data, trained **less than one week**

Figure 3.25: Comparison of Depth-based ControlNet and Stable Diffusion V2 Depth-to-Image.

Figure 3.26: Controlling Stable Diffusion (anime weights) with cartoon line drawings.

Figure 3.27: Ablative study. We compare the ControlNet structure with a standard method that Stable Diffusion uses as default way to add conditions to diffusion models.

Figure 3.28: (Continued) Comparison of six detection types and the corresponding results.

Chapter 4

Implementation of the Denoising Diffusion Probabilistic Model (DDPM) on a Simple Dataset (EMNIST)

Let's Begin by Introducing the EMNIST Dataset

The Extended MNIST (EMNIST) dataset is an extension of the widely used MNIST dataset, designed to offer a more comprehensive collection of handwritten characters. While the original MNIST dataset contains only digits, EMNIST provides a diverse array of characters, including both uppercase and lowercase letters, as well as digits. This expansion enhances the dataset's applicability to a broader range of tasks, such as character recognition, language modeling, and even symbol classification. EMNIST's rich diversity of handwriting styles, combined with its extensive character set, makes it a valuable resource for training and evaluating various machine-learning models. Its availability has contributed significantly to advancing research and applications in fields such as optical character recognition and natural language processing.

After doing the learning for DDPM we will following results for image generation approach.

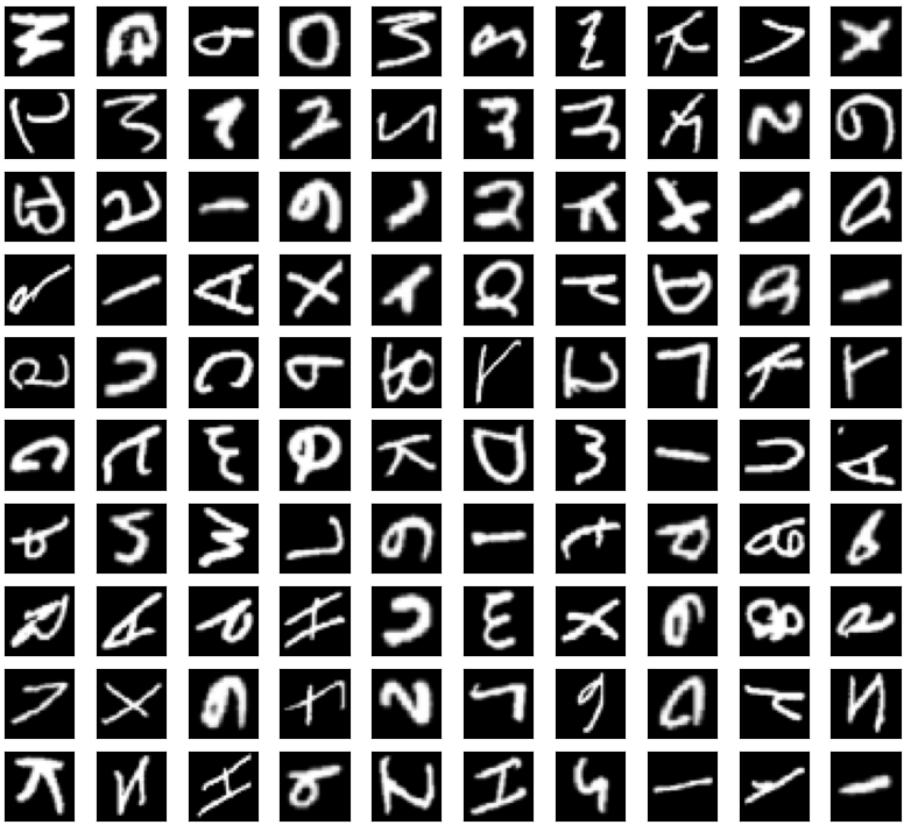


Figure 4.1: Some images of EMNIST Dataset.

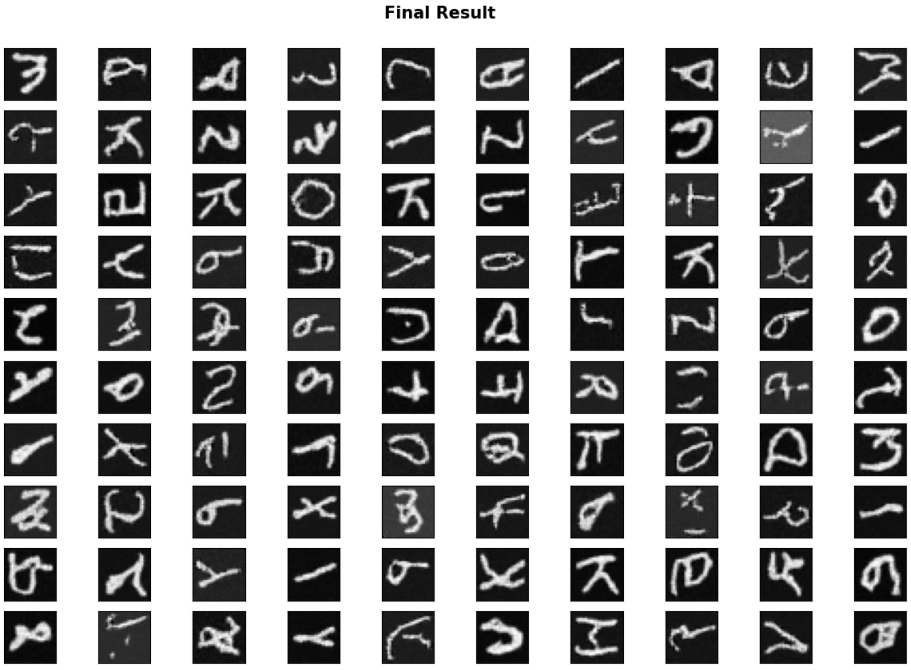


Figure 4.2: Generated images using DDPM.

Chapter 5

Conclusion

Diffusion deep models have emerged as a powerful and versatile class of algorithms in the realm of deep learning. Their ability to capture complex temporal dependencies and patterns in data through diffusion processes presents a promising avenue for addressing a wide array of real-world challenges. By integrating ideas from various domains such as computer vision, natural language processing, and generative modeling, these models have showcased remarkable performance in tasks ranging from image synthesis and denoising to time series analysis and recommendation systems. As researchers continue to refine and extend the capabilities of diffusion deep models, their potential impact on diverse fields remains substantial. As we navigate the evolving landscape of artificial intelligence, diffusion deep models are poised to make significant contributions, driving innovation and advancing our understanding of complex data distributions.

References

Bibliography

[1] arXiv 2020: Denoising Diffusion Probabilistic Models

- GitHub: Source Code

[2] arXiv 2021: Diffusion Models Beat GANs on Image Synthesis

- GitHub: Source Code

[3] arXiv 2022: High-Resolution Image Synthesis with Latent Diffusion Models

- GitHub: Source Code

[4] arXiv 2022: Hierarchical Text-Conditional Image Generation with CLIP Latents

- DALL-E 2: Blog Post

[5] arXiv 2022: Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

- GitHub: Source Code

[6] arXiv 2023: Adding Conditional Control to Text-to-Image Diffusion Models

- GitHub: Source Code