Generative AI in Vision: A Survey on Models, Metrics and Applications

Gaurav Raut
University of Maryland
gauraut14@gmail.com

Apoorv Singh
Carnegie Mellon Univeristy
apoorv93singh@gmail.com

Abstract

Generative AI models have revolutionized various fields by enabling the creation of realistic and diverse data samples. Among these models, diffusion models have emerged as a powerful approach for generating high-quality images, text, and audio. This survey paper provides a comprehensive overview of generative AI diffusion and legacy models, focusing on their underlying techniques, applications across different domains, and their challenges. We delve into the theoretical foundations of diffusion models, including concepts such as denoising diffusion probabilistic models (DDPM) and score-based generative modeling. Furthermore, we explore the diverse applications of these models in text-to-image, image inpainting, and image super-resolution, along with others, showcasing their potential in creative tasks and data augmentation. By synthesizing existing research and highlighting critical advancements in this field, this survey aims to provide researchers and practitioners with a comprehensive understanding of generative AI diffusion and legacy models and inspire future innovations in this exciting area of artificial intelligence.

1. Introduction

Generative models have long been at the forefront of artificial intelligence, enabling the creation of synthetic data samples with remarkable realism and diversity. Initially introduced as a method for denoising images, diffusion models have evolved to become a versatile framework for generating high-quality images, text, and audio data. Over the years, they have garnered significant attention from researchers and practitioners alike for their ability to capture complex data distributions and produce realistic samples.

Generative models for computer vision started in 1950 through Hidden Markov models (HMMs) and Gaussian Mixture models (GMMs). These models used hand-designed features with limited complexities and diversity. With the advent of deep learning, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) enabled impressive image generation. However, in practice, GANs suffered several shortcomings in their architecture [17]. The simultaneous training of generator and discriminator models was inherently unstable; sometimes, the generator "collapsed" and outputted lots of similar-seeming samples. Then came diffusion models, which were in-

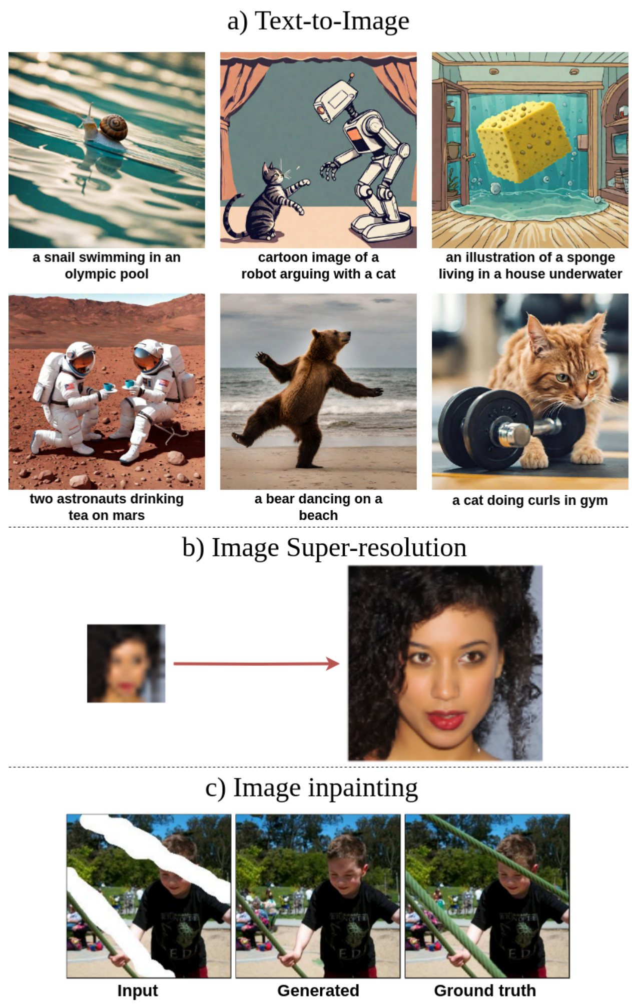

Figure 1. a) Images generated using stable diffusion[70]; b) Image super-resolution results from SR3[72]; c) Image inpainting results from Palette[73]

spired by physics. Diffusion systems borrow from diffusion in non-equilibrium thermodynamics, where the process increases the entropy or randomness of the system over time. The recent innovation of diffusion models from OpenAI made them more practical in everyday applications. This paper dives into a systematic review of techniques and methodologies involved in SOTA diffusion models.

The main contribution of this work can be summarized as follows:

- An overview of generative vision models to get readers up-to-speed with the theoretical prerequisites for going through the latest trends in diffusion models.
- In-depth survey on the SOTA approaches for diffusion models, including
- Highlight current research gaps and future research directions to provoke researchers to advance this generative vision modeling field further.

2. Generative Models in Vision

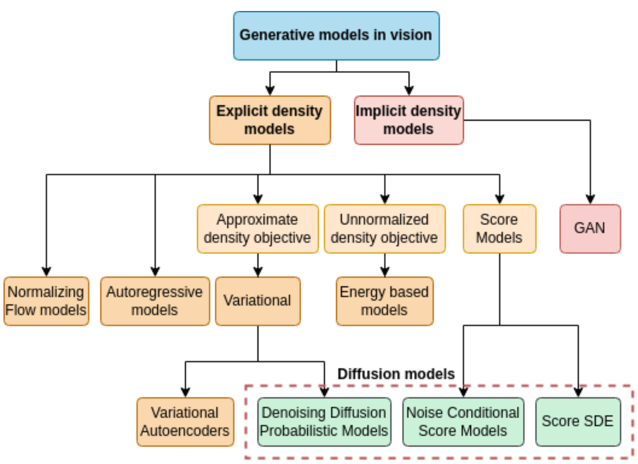

Figure 2. An extension of generative models classification based on[25]

Generative models in vision can be mainly classified into two categories Fig. 2: models that explicitly learn the probability density function by maximizing the likelihood or implicitly, where the model doesn't directly target to learn the density but does so using other strategies.

Normalizing flow and autoregressive models are dense probability estimators that model the exact likelihood of a distribution. Although, they are limited by the complexity of data distribution as converging the objective to achieve exact density representation of high dimensional complex data, such as images, can yield to computationally heavy and impractical models.

Variational autoencoders (VAE) alleviate the computation issue by allowing approximation of the intractable density distribution. This allows for a more efficient generative model with a trade-off of struggling against capturing complex data distributions.

Energy-based models offer a flexible modeling objective without any restrictions. They use an unnormalized representation of the probability distribution, making them excellent density estimators. However, the intractable objective makes them computationally inefficient for both training and sampling.

GANs[26], on the other hand, don't model the density objective directly. They rely on using an adversarial approach, which uses a minimax game between a discriminator and a generator to learn the density estimation explicitly. Although they have been largely successful on a wide set of applications, training them is difficult and suffer problems like vanishing gradient and mode collapse.

Diffusion models use variational or score-based approaches to model the probability density function. They work by perturbing data with continuous or discrete noise injection on either the data directly or a latent representation such as the latent diffusion model[70] and learning the reverse denoising process.

2.1. Diffusion models

2.1.1 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models, or DDPMs for short, are a class of diffusion models that are based on slowly introducing Gaussian noise to a training data sample x_0 over a large set of time steps ($1 \rightarrow T$), thereby obtaining a set of noise perturbed latent intermediates of the original sample (x_1, x_2, ..., x_T). This forward process is governed by the forward diffusion kernel (*FDK*). At the end of T time steps, we obtain the resulting x_T sample, which can be approximated to represent isotropic Gaussian noise. From here, a deep neural network is tasked to learn the *FDK*, and the parameters of this network are called the Reverse Diffusion Kernel (*RDK*). The aim for *RDK* is to predict the noise introduced by the *FDK* at each time step starting from x_T and slowly remove the noise to generate new samples that belong to the same probability distribution as our training dataset [33, 82].

The forward process: Let $q(\mathbf{x_0})$ represent the given sample's probability distribution before the noise perturbation. We define a Markov chain with our *FDK* defined as a Gaussian distribution,

$$q(\mathbf{x}_{1,\ldots,T}) := \prod_{i=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{1}$$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathrm{I}) \tag{2}$$

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon \tag{3}$$

where, Eq. (2) defines the *FDK*; $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The term β_t is a hyper-parameter in the diffusion process controlled by the variance scheduler. By applying this kernel to our $q(\mathbf{x}_0)$ repeatedly over $T-1$ time steps, we obtain $q(\mathbf{x}_T)$ which approximates to an isotropic Gaussian distribution given that our covariance matrix in the *FDK* is isotropic. Since the process is a Markov chain, we can seamlessly obtain any latent representation and probability distribution from $\mathbf{x}_0$ by simply substituting $\alpha_t := 1 - \beta_t$ and $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$.

$$q(\mathbf{x}_t \mid \mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\overline{\alpha}_t}\mathbf{x}_{t-1}, (1 - \overline{\alpha}_t\mathbf{I}) \tag{4}$$

$$\mathbf{x}_t := \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon \tag{5}$$

The reverse process: Starting from approximately an isotropic Gaussian distribution obtained at the time step T, the aim is to learn the *RDK* $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ which will predict the noise injected at each time step and will generate our original sample $q(\mathbf{x}_0)$ back within the finite T time steps. The fact that the reverse process starts from a random isotropic Gaussian distribution serves as a cue that a trained network can produce new samples in the probability distribution of the training dataset while sampling.

The probability distribution in the reverse process to obtain our original sample can be illustrated as all the possible paths are taken from $p_\theta(\mathbf{x}_T)$ to obtain $p_\theta(\mathbf{x}_0) \sim q(\mathbf{x}_0)$ at each time step in $T \to 1$.

$$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0\ldots T} d\mathbf{x}_{0\ldots T}) \tag{6}$$

This integral is intractable since it integrates over a complex high-dimensional space. To solve this, the authors [33, 82] introduced a variational lower bound (or Evidence lower bound (**ELBO**)) of the negative log-likelihood similar to VAEs[42] to minimize this. Using this, we obtain the variational lower bound loss, which is [15],

$$\begin{aligned} L_{VLB} = &\underbrace{-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{\mathbf{L}_0} + \underbrace{\mathbf{D}_{KL}(p(\mathbf{x}_T \mid \mathbf{x}_0) \| p_{\mathbf{x}_T})}_{\mathbf{L}_T} \\ &+ \underbrace{\sum_{t>1} \mathbf{D}_{KL}(p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{\mathbf{L}_{t-1}}, \end{aligned} \tag{7}$$

Here, we observe that the $\mathbf{L}_T$ doesn't depend on any learnable parameters from the network and hence can be omitted from the loss. The term $\mathbf{D}_{KL}$ is called the Kullback-Leibler divergence, a non-symmetric measure of the statistical distances between two probability distributions. This loss function trains the network and estimates the forward process posterior. We then define that the *RDK* learned by the network is a Gaussian distribution given as $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t))$ where the network, minimizing the variation lower bound loss, will learn to estimate the mean μ_θ and the covariance σ_θ of the forward process posterior distribution. The authors further minimize the loss function in [33] where they set the variance β from Eq. (3) as a constant. The authors then re-parameterize the mean μ_θ in terms of noise, and we obtain the simplified loss function given as:

$$L_{simple} := \mathbb{E}_{\mathbf{x}_0, \epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2], \tag{8}$$

which is equivalent to the loss introduced in the score-based models[83].

2.1.2 Noise Conditional Score Models

Score-based models define the dataset's probability distribution as $q(\mathbf{x})$. The score function for the probability function can then be defined as the gradient of the log of this probability distribution, $\nabla_{\mathbf{x}} \log q(\mathbf{x})$ [83]. The objective is to train a deep neural network, parameterized by θ to approximate over the score function of our dataset's probability function, $\mathbf{s}_\theta(\mathbf{x}) \sim \nabla_{\mathbf{x}} \log q(\mathbf{x})$, also known as score matching. Using the score function instead of the probability distribution allows the network to work with a tractable objective by eliminating the normalizing constant [35]. The loss function is the Fischer divergence between the actual score and the learned score, which is derived to form,

$$\mathbb{E}_{q(\mathbf{x})}[\mathrm{tr}(\nabla_{\mathbf{x}}\mathbf{s}_\theta(\mathbf{x})) + \frac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|^2] \tag{9}$$

With a trained network, Langevin dynamics allows us to generate new samples using only the score function[83].

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}\nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\mathcal{N}(0, \mathbf{I}) \tag{10}$$

Under the ideal conditions of $t \to \infty$ and $\epsilon \to 0$, we can generate exact samples coming from our dataset's distribution $q(\mathbf{x})$ [97]. The generation of new samples can then happen by simply substituting the learned score function $s_\theta(\mathbf{x})$ in Eq. (10).

However, the authors [83] observed that this approach did not do well in practice because the scores generated were often inaccurate in lower-density regions. To address this, two solutions are employed to enhance the score matching using either the denoising score matching [83] or sliced scored matching [86] where the loss in Eq. (9) is either bypassed or approximated using random projections.

The idea of denoising score matching is to train the model by perturbing the dataset by inserting an isotropic Gaussian noise $\mathcal{N}(0, \sigma_{(1,\ldots,L)}\mathbf{I})$ and $\sigma_1 < \ldots < \sigma_L$ where the prior data distribution is approximately equal to the noise perturbed distribution as the noise is inserted gradually over a large number of steps, $\nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}) \sim \nabla_{\mathbf{x}} \log q(\mathbf{x})$. This results in the modified loss function of,

$$\sum_{i=1}^{L} \lambda(i)\mathbb{E}_{q_{\sigma_i}(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log q_{\sigma_i}(\mathbf{x}) - s_\theta(\mathbf{x}, i)\|^2] \tag{11}$$

where, $\lambda(i) \sim \sigma_i^2$ is a positive weighting function. For sampling, the Langevin dynamics are updated such that the trained network will produce samples similar to Eq. (10) for each $i = L, \ldots, 1$, and the prior output will be used as the input for the next run. This process is called the annealed Langevin dynamics, producing a less noisy sample after every run from $i = L, \ldots, 1$.

2.1.3 Stochastic Differential Equations Generative Models

So far, we have seen that the diffusion models perturb data over a range of time steps or iterations $1 \to L$ [33, 83, 84]. However, in [87], the authors generalized the previous approaches by defining a stochastic differential equation (SDE) to perturb data with noise in a continuum. The aim is to govern the diffusion process in both forward and reverse direction as a representation of an SDE.

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w} \tag{12}$$

where, $\mathbf{w}$ signifies the Brownian motion, the function $\mathbf{f}(\mathbf{x}, t)$ is called as the drift coefficient and $g(t)$ is the diffusion coefficient [87]. This is the SDE defining the forward process. Intuitively, it can be considered an SDE that forces a random sample to converge in areas of high probability densities. To sample, we need to solve this SDE backward in time. The reverse process is also a SDE[1, 87] which is given as,

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_x \log q_t(\mathbf{x})]dt + g(t)d\overline{\mathbf{w}} \tag{13}$$

This SDE has a negative time step since the solution must be reversed ($t = T \to t = 0$). To sample from this SDE, we need to learn the score as defined in [83] as a function of time. Analogous to the noise conditional models, where the score function depends on the noise scales $\sigma_{L,\ldots,i,\ldots,1}$, the score function here will depend on time $s_\theta(\mathbf{x}, t)$, formulating the loss function as,

$$\mathbb{E}_t \mathbb{E}_{q_t(\mathbf{x})}[\lambda(t) \|\nabla_{\mathbf{x}} \log q_t(\mathbf{x}) - s_\theta(\mathbf{x}, t)\|^2] \tag{14}$$

After training our network $s_\theta(\mathbf{x}, t) \sim \nabla_x \log q_t(\mathbf{x})$, we can generate a new sample by solving the reverse SDE starting from pure noise $q_T(\mathbf{x})$. Many SDE solvers exist, the simplest of which is the Euler-Maruyama method. Similar to [83], we choose an infinitesimally small $\Delta(t)$ to solve the generalization of this SDE.

$$\begin{aligned}
&\Delta(\mathbf{x}) = [\mathbf{f}(\mathbf{x}, t) - g^2(t) s_\theta(\mathbf{x}, t)]\Delta(t) + g(t)\sqrt{|\Delta t|} z_t \\
&\mathbf{x} = \mathbf{x} + \Delta\mathbf{x} \\
&t = t + \Delta t
\end{aligned} \tag{15}$$

Following this, the authors improvise the sampling by introducing a predictor-corrector sampler where the idea is that for every step, the SDE solver will predict $\mathbf{x}(t + \Delta t)$ and then, the corrector will use this as an initial sample to refine the sample using $s_\theta(\mathbf{x}, t + \Delta t)$ by running it through the corrector network.

2.2. Generative Adversarial Networks

Generative Adversarial Networks (*a.k.a.* GANs)[26] are a class of generative models that implicitly learns the probability distribution $q(\mathbf{x})$ of the dataset using an adversarial approach where two networks, the discriminator D and the Generator G play a two-player min-max game. The discriminator's objective is to maximize the binary classification of distinguishing real and generated images, whereas the generator's objective is to fool the discriminator into misclassifying the generated images.

$$\begin{aligned}
&\min_G \max_D V(D, G) = \\
&\mathbb{E}_{\mathbf{x}\sim q_{\mathbf{x}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))],
\end{aligned} \tag{16}$$

GANs are notoriously difficult to train [2, 26, 62, 76]. The objective is to find the Nash equilibrium (also referred to as a saddle point in other literature) of a two-player game (D & G) where both networks try to minimize their cost function simultaneously Eq. (16). This results in a highly unstable training process as optimizing D can lead to the deterioration of G and vice-versa. Mode collapse is another problem where the generator's objective function converges to a specific data distribution instead of the whole training set, thus only generating images belonging to this small subset. Also, when the discriminator is trained to optimality, the gradients of D approach zero. This causes the problem of vanishing gradients for the generator, where it has no guidance into which direction to follow for achieving optimality.

To prune these problems, various modifications on the vanilla GAN were introduced, which either suggested architectural optimizations or loss function optimizations [36]. As studied by [14, 101], the loss function optimizations were divided into optimizing the discriminator's D or generator's G loss objective. These include minimizing the f-divergences along with the Jensen-Shannon divergence [59], weight normalization for stabilizing D's training [55], WGAN and WGAN-GP[3, 30] which changed the objective of D from binary classification to a probability output by applying the Earth Mover (EM) or Wasserstein distance, EBGAN [113] introduces an energy-based formulation of D's objective function where the architecture of D is modified to be an auto-encoder, BEGAN [5] which uses the same auto-encoder architecture for D from EBGAN but modifies the objective to use the Wasserstein distance instead, SAGAN[111] introduces self-attention modules on both D and G to enhance feature maps and uses spectral normalization[55].

2.3. Variational Autoencoders

Variational Autoencoders (VAEs)[41, 42] are generative models based on learning the latent space representation by projecting a prior $\mathbf{z}$ on the latent vector before generating a distribution. They have an encoder-decoder architecture similar to auto-encoders, while mathematically, they differ a lot [21]. Instead of learning the latent vector as a discrete representation of the dataset, VAEs learn the probability distribution of this space. More intuitively, $q(\mathbf{x})$ is mapped to learn a multi-variate Gaussian distribution represented by the mean $\mu_{\mathbf{z}}$ and co-variance $\sigma_{\mathbf{z}}$ where $\mathbf{z}$ is in the latent space. While generating new samples, we want to start with the latent representation of $\mathbf{z}$ as an isotropic Gaussian distribution $\mathcal{N}(\mathbf{z}|\mu, \sigma * I)$. The regularization term and the reconstruction loss are introduced to achieve this. This regularization term is the KL divergence between the encoder's estimation of the latent variables and the standard Gaussian distribution. The loss function is then formulated as,

$$L(\theta, \phi; \mathbf{x}) = \underbrace{-\mathbf{D}_{KL}(p_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))}_{L_{KL}} + \underbrace{\mathbb{E}_{p_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{L_{reconstruction}} \tag{17}$$

where ϕ and θ represent the encoder and decoder parameters, respectively. The only problem here is that $p_\theta(\mathbf{z})$ is intractable. To make the latent variable a learnable parameter, the authors [42] introduced a reparameterization trick to allow backpropagation. This is done by separating the stochastic part ϵ and reconstructing the latent vector as $\mathbf{z} = \mu + \sigma * \epsilon$. When the network has been trained to optimality, $p_\theta(\mathbf{x}|\mathbf{z}) \sim q(\mathbf{x})$.

2.4. Autoregressive models

Autoregressive models are generative models that use sequential data to calculate the likelihood of the next value in series[46, 90]. The joint distribution of a dataset can be given as,

$$q(\mathbf{x}) = \prod_{i=1}^{N} q(\mathbf{x}_i|\mathbf{x}_1, \ldots, \mathbf{x}_{n<i}) \tag{18}$$

In vision, this translates to generating images by sequencing the generation of each pixel given the prior pixels[77, 92, 93]. More formally, the autoregressive network consists of either recurrent or convolutional layers that jointly learn the dataset's density distribution in a tractable manner and, during inference, will run $N = n^2$ times for an image of size $n * n$ to generate a sample. Images are not unlike audio[91] or text[27, 95] where the data is structured and sequenced. [92] introduces a sequential approach for image synthesis by masking the pixels on the right and below and only considering the pixels above and on the left of the pixel we want to predict.

2.5. Normalizing flow models

Normalizing flow is a way of mapping a data's complex probability distribution $q(\mathbf{x})$ to a simple latent distribution $p(\mathbf{z})$ using a set of invertible, bijective and continuous functions $\mathbf{z} = f(\mathbf{x})$ such that both f and f^{-1} are differentiable[19, 20, 69]. When the function f is a deep neural network, the model is called a normalizing flow model. Using the rule of change of variables, the probability density can be explicitly given as,

$$p_{\mathbf{x}}(\mathbf{x}, \theta) = p_{\mathbf{z}}(f_\theta(\mathbf{x}))\left|\det \frac{\delta f_\theta(\mathbf{x})}{\delta \mathbf{x}}\right| \tag{19}$$

Since normalizing flow models estimate the exact likelihood of the distribution, the training is done by minimizing the negative log-likelihood given as,

$$L_{NF} = -\log p_{\mathbf{x}}(\mathbf{x}, \theta) = -\log p_{\mathbf{z}}(f_\theta(\mathbf{x})) - \log\left|\det \frac{\delta f_\theta(\mathbf{x})}{\delta \mathbf{x}}\right| \tag{20}$$

When the model is trained, the latent representation $\mathbf{z}$, often chosen as a multivariate Gaussian distribution[43, 60], can generate a sample from the dataset's probability distribution by simply applying the inverse function f_θ^{-1} to it. Although flow models are based on modeling the exact data distribution, they are often computationally expensive since they depend on the calculation of jacobians, have scalability and expressiveness issues on large and complex data distributions, and because of the invertibility constraint in calculations, require the input and output dimensions to be the same[8].

2.6. Energy Based models

In simplicity, Energy-based models (EBMs) [47, 56, 85] are generative models that assign an energy function $E_\theta(\mathbf{x})$ to a dataset's probability distribution $q(\mathbf{x})$ and minimize the energy function for samples from the dataset while assigning high energy to samples that don't belong to it.

$$p_\theta(\mathbf{x}) = \frac{e^{-E_\theta(\mathbf{x})}}{Z_\theta} \tag{21}$$

EBMs are incredibly flexible in the domain of the type of data[23, 56, 85] and, in vision, are an excellent choice for tasks of anomaly detection as an optimally trained model can distinguish between the anomalies and ideal sample[107].

Because the energy function is unnormalized, $Z_\theta = \int_{\mathbf{x}} e^{-E_\theta(\mathbf{x})} d\mathbf{x}$ is used in Eq. (21) for normalizing the likelihood which is often intractable. This makes the training and sampling of EBMs difficult, and one has to rely on computationally heavy methods such as MCMC, contrastive divergence (CD), and score matching, which make them an impractical choice for fast inference use cases.

3. Evaluation metrics for vision generative models

Evaluating generative models in vision is an active research topic where different tasks involve specialized metrics such as DrawBench[72], PartiPrompts[106], CLIPScore[31] for text-to-image tasks or PSNR for image reconstruction tasks. Here, we mainly introduce metrics that measure image fidelity and model diversity.

3.1. Inception Score (IS)

Inception Score (or IS) was first introduced to assess the quality of the images generated by GAN as an automated alternative process against human annotators[76]. Using a feature extracting network (often the Inception model[88]), which is trained on the same dataset as the generative model, the score measures two components of the generated samples: entropy of a single sample over the class labels and secondly, entropy of class distribution over a large number of samples (suggested close to 50K samples) to measure the diversity. For a well-trained model, the entropy of a class over a single sample should be low, and the entropy of class distribution over all generated samples should be high. This indicates that the network can generate both meaningful and diverse sets of images. The score is calculated as follows:

$$\mathbf{IS} = \exp(\mathbb{E}_{\mathbf{x}}[\mathbf{D}_{KL}(p_\theta(\mathbf{y}|\mathbf{x})\|p_\theta(\mathbf{y})))] \tag{22}$$

where $\mathbf{D}_{KL}$ is the KL divergence, $p_\theta(\mathbf{y}|\mathbf{x})$ is the predicted class probability and $p_\theta(\mathbf{y})$ is the probability of classes over all generated images. This implies that the higher the IS score is, the better the model's generative capabilities.

3.2. Fréchet Inception Distance

The drawback of the Inception score is that it only considers the generated samples for evaluation and disregards comparing them with the actual dataset. Also, the IS will only compare the class probabilities as opposed to the image feature distribution, which causes it to miss the more relevant image features and requires a labeled dataset. The Fréchet Inception Distance (FID)[22, 32] allows us to overcome this by comparing the extracted features from a certain layer of the feature extractor. More specifically, it assumes that the extracted features belong to a Gaussian distribution with a certain mean μ and co-variance Σ and calculates the Fréchet distance (measures the similarity between two probability distributions) between samples generated by the model (μ_g,Σ_g) and the samples from the training (μ_t,Σ_t) dataset($\sim$ 50K samples). It is given as,

$$\mathbf{D}_{FID} = \|\mu_t - \mu_g\|^2 + \mathrm{tr}(\Sigma_t + \Sigma_g - 2\sqrt{\Sigma_t\Sigma_g}) \tag{23}$$

Since FID is a similarity distance, the lower the FID is, the better. For zero-shot applications [57, 65, 105], a modified version called the zero-shot FID is used where μ_t, Σ_t signify the target distribution based on textual cues as opposed to the training dataset. Kernel Inception Distance (KID) [7] is another flavor of the FID, where the distance is measured between a polynomial representation of the inception layer's distribution.

3.3. Precision and Recall

Precision and recall [45, 75] are two metrics that follow the same motivation as IS and FID of measuring the quality and diversity of the generated samples while overcoming issues of mode dropping. It provides a two-dimensional score where precision measures the quality of images produced while recall measures the diversity coverage of the generative model.

4. Applications in Vision

Generative models find applications in various tasks such as image denoising, inpainting, super-resolution, text-to-video synthesis, image-to-image translation, image search, and reverse image search. These applications can be divided into two broad categories:

4.1. Unconditional generation

As the name suggests, unconditional generative models are trained to learn a target distribution and synthesize new samples without getting conditioned by any other input. All models described in Sec. 2 can be considered a base unconditioned model whose only focus is on learning the target distribution[15, 100]. Unconditional image generation models usually start with a seed that generates a random noise vector. The model will then use this vector to create output images that resemble training data distribution.

4.2. Conditional generation

On the contrary, conditional diffusion models take a prompt and some random initial noise and iteratively remove the noise to construct an image. The prompt guides the denoising process, and once the denoising process ends after a predetermined number of time steps, the image representation is decoded into an image. There are several forms of conditional models:

4.2.1 Text-to-Image Generation

Text-to-image has been the most naturally prominent use case for generative models. Providing conditioned textual information for image generation has improved the model's generative capabilities[63, 108]. Over the years, architectures reigning the task have used recurrent layers [53], GANs [39, 49, 67, 78, 110], autoregressive models [18, 64, 106] and the diffusion-based models [10, 28, 66, 70, 74].

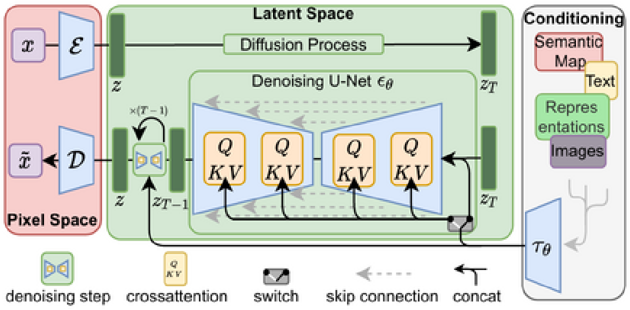

Figure 3. An example of a conditional generative model: Latent diffusion model[70]

Below we present an overview of their functioning and also present their metrics in Tab. 1.

Table 1. Comparison of different text-to-image models

Models	Architecture	Best Reported			
		Zero-shot FID↓	IS↑	Params	Dataset
StackGAN[110]	GAN	-	8.45	-	COCO
GigaGAN[39]	GAN	9.09	-	1.0B	COCO
DALL-E[64]	Autoregressive	27.50	18	12B	COCO
GLIDE[58]	Diffusion	12.24	23.7	5B	COCO
Stable Diffusion[70]	Latent Diffusion	12.63	30.29	1.45B	COCO
DALL-E2[66]	Diffusion	10.39	-	5.5B	COCO
Imagen[74]	Diffusion	7.27	-	3B	COCO
Parti-20B[105]	Autoregressive	**3.22**	-	20B	COCO
Re-Imagen[10]	Diffusion	6.88	-	3.6B	COCO

Text conditioned GANs: Followed by cGANs [54], [67] introduced embedding textual information to achieve text-to-image generation. The model is jointly trained using images and text captions. During sampling, the text prompts are converted to text encoding using an encoder and compressed using a 128-dimensional fully connected layer. This compressed encoding is concatenated with the latent vector $\mathbf{z}$ and passed to the generator to generate images. StackGAN [110] introduced a series of two GAN networks where the text encodings from the encoder are mapped to a Gaussian distribution with random noise and are then concatenated with the latent vector of the stage-I GAN. The stage-I GAN generates a low-dimensional image, which is further embedded in the latent vector of the stage-II GAN to generate a high-resolution image. The image compression is done using the trained discriminator from stage-I.

Text conditioned autoregressive models: DALL-E [64] is a two-stage autoregressive transformer model where the first stage has a discrete VAE to tokenize images in a 32×32 grid and the second stage concatenates the text encodings from a BP-Encoder to create text tokens. These concatenated image-text tokens are jointly trained to maximize the ELBO [42] to obtain an image-text distribution. Similarly, CogView [18] employs a VQ-VAE[94] for image tokenization and SentencePiece[44] for generating text tokens and Parti[106] uses the ViT-VQGAN[104] as the image tokenizer and a pre-trained BERT[16] for text encodings with the best fine-tuned model achieving SOTA FID score of 3.22 on MS-COCO dataset.

Text conditioned Diffusion models: GLIDE[58] uses an ADM model [17] and a transformer-based text encoder to generate prompts in place of class labels for image synthesis. DALL-E2 (*a.k.a.* unCLIP)[66] is a two-stage model where the first stage uses the CLIP[63] model to generate image embedding from text captions and the second stage uses these image embedding as a prior to generating samples via a diffusion decoder. The authors also experimented with using an autoregressive decoder instead of the diffusion decoder, but the latter yielded better results. Stable Diffusion[70] is a latent diffusion model (LDM). Instead of directly dealing with the complex pixel representation, LDMs apply the DDPM model's forward kernel on the latent representation generated by the encoder Fig. 3. A series of trained denoising U-Net are applied to denoise this corrupted latent space. The text prompts are embedded in the denoising steps using cross-attention layers. The retrieved latent space, after denoising, is passed through a decoder to generate the sample. Because LDMs use diffusion on the latent representation, training, and sampling have proven to be computationally inexpensive as compared to other models. Imagen[74] encodes its textual prompts using a T5-XXL LLM model similar to CLIP. It then generates low-resolution images (64×64) using a series of denoising U-Net and then up-samples these images using a series of two super-resolution U-Net diffusion models to generate images of size 256×256 and 1024×1024 respectively. Re-Imagen[10] focuses on retrieving k-nearest neighboring images from a dataset based on the text prompts provided and uses these images as a reference to generate new samples. DALL-E3[6] attends to improving image quality by re-captioning text prompts into a more descriptive prompt, which has proven to generate higher-quality images.

4.2.2 Image super resolution

ViT-based models have been shown to achieve SOTA results for the task of image super-resolution[11–13, 109, 112]. Even so, the generalization capabilities of generative models can soon catch up the leaderboard[24, 99]. SRGAN[48] first introduced a generative framework for this task using an adversarial objective. ESRGAN[96] and RFB-ESRGAN[81] further improvise the SRGAN implementation by employing architectural modifications such as relativistic discriminator, dense residual block, and upgrading the perceptual loss. GLEAN [9] introduced a novel encoder-bank-decoder approach where the encoder's latent vectors and multi-layer convolutional features are passed to a StyleGAN[40] based latent bank. This generative bank combines features from the encoder at various scales and generates new latent feature representations passed to the decoder to generate a super-resolution image. SR3[72] and SRDiff[50] were the first diffusion (DDPM) based SR models. SR3 uses a conditional DDPM U-Net architecture with some adaptations in the residual layers and conditions the denoising process using an LR image directly at each iteration. SRDiff uses an encoder-generated embedding of the LR image and conditions the denoising step by concatenating the embeddings at each iteration. IDM[24] uses an implicit neural representation in addition to the conditioned DDPM to achieve a continuous restoration over multiple resolutions using the current iteration's features. EDiffSR[99] uses the SDE diffusion process where isotropic noise is conditioned with the LR image during sampling to generate the high-resolution image.

4.2.3 Image anomaly detection

AnoGAN[79] and f-AnoGAN[80] are unsupervised adversarial anomaly detection networks that were trained on healthy data and using the proposed anomaly score along with the residual score; the model predicts anomalies on unseen data depending on the variation in the learned latent space. DifferNet[71] uses the normalizing flow model to map the density of healthy image features extracted from a feature extraction network. By training on healthy data, anomalies will have a lower likelihood and will be out of distribution in the density space. FastFlow[103] uses a similar approach but extends the normalizing flow into a 2D space, which allows to directly output location results of anomalies. CFLOW-AD[29] uses an encoder feature extractor where features from every scale are pooled to form multi-scale feature vectors, which are passed to the specific normalizing flow decoder along with the positional encoding for localization of anomalies. The outputs from each decoder are aggregated to generate an anomaly map. AnoDDPM[98] approaches the problem using a DDPM model with simplex noise for data perturbation. The diffusion model is trained to generate healthy images. During inference, the simplex noise perturbs the test sample for a certain number of pre-set steps, and the denoising diffusion model then generates an anomaly-free image using the perturbed sample as the prior. Comparing the generated and input samples using reconstruction error, an anomaly segmentation map is generated.

4.2.4 Image inpainting

Image inpainting tasks include restoration, textural synthesis, and mask filling[4]. Context encoders [61] applied an adversarial loss along with the reconstruction loss to achieve sharp and coherent mask filling. [37] advances the simple discriminator by introducing a mixture of local and global context discriminators. The local discriminator has the filled mask as input, and the global discriminator takes the whole image as input to collectively create the discriminator's objective. Following it, [102] uses a two-step generative approach where the first generator (trained on reconstruction loss) generates a coarse prediction and the second generator with a WGAN-GP[30] based objective is the refinement generator trained on local and global adversarial loss along with the reconstruction loss. StructurFlow[68] bifurcates the GAN generation in two steps: the structure generator and the texture generator. The structure generator creates a smoothened edge-preserved image, and the texture generator fills the texture in the smooth reconstructed image. The texture generator uses an additional input of appearance flow in the latent space, which predicts the texture of the masked regions based on the texture from source regions. CoModGAN[114] further generalizes the inpainting task with both input-masked image and stochastic noise-conditioned latent vector input to the generator, which enabled them for large region image inpainting. Pallete[73] is a DDPM[33] based image-to-image translation model capable of image inpainting. It fills the masked region with standard Gaussian noise and performs the denoising training only on the masked region. RePaint[51] salvages the pre-trained unconditional DDPM model instead of training a model for the inpainting task. Since DDPM follows a Markov chain for data perturbation, the masked input image's noise-perturbed data is known for every iteration in reverse. Using this knowledge, the denoising process is conditioned to add the noise-perturbed image at each reverse step to predict the masked region. Given the stochastic nature, this process can generate multiple candidates for the inpainting task.

4.2.5 Other tasks

In addition to the above tasks, these models can be used for various other generative tasks like image-to-image translation[38, 89], image colorization[73], video generation[34], point cloud generation[52], restoration, etc.

5. Future directions

Some of the unsolved but highly sought-after directions that researchers can take are:

- Exploring Time-Series Forecasting Applications: Future research could delve deeper into leveraging diffusion models for improved forecasting accuracy and efficiency.
- Physics-Inspired Generative Models: Future research could focus on advancing physics-inspired generative models to achieve unprecedented speed and quality in content creation.
- Ethical Considerations: Future research could involve addressing issues related to bias, fairness, and the societal

impact of generative diffusion models.

6. Conclusion

This survey paper has provided a detailed examination of generative AI diffusion models, shedding light on their techniques, applications, and challenges. Despite their promising capabilities, generative AI diffusion models still face significant challenges such as training stability, scalability issues, and interpretability concerns. Addressing these challenges will be crucial for advancing the field and unlocking the full potential of diffusion models in generating realistic and diverse data samples. By synthesizing current research findings and identifying key areas for future research, this survey aims to guide researchers and practitioners toward further advancements in generative AI diffusion models, paving the way for innovative applications and breakthroughs in artificial intelligence.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3): 313–326, 1982. 4

[2] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017. 4

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. 4

[4] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, page 417–424, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 8

[5] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks, 2017. 4

[6] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 7

[7] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021. 6

[8] 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*, 44(11):7327–7347, 2022. 5

[9] Kelvin C.K. Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14245–14254, 2021. 7

[10] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W. Cohen. Re-imagen: Retrieval-augmented text-to-image generator, 2022. 6, 7

[11] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer, 2023. 7

[12] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, and Xiaokang Yang. Recursive generalization transformer for image super-resolution, 2023.

[13] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution, 2023. 7

[14] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018. 4

[15] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869, 2023. 3, 6

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 7

[17] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 1, 7

[18] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers, 2021. 6, 7

[19] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015. 5

[20] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017. 5

[21] Carl Doersch. Tutorial on variational autoencoders, 2021. 5

[22] D.C Dowson and B.V Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. 6

[23] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 5

[24] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10021–10030, 2023. 7, 8

[25] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2017. 2

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2, 4

[27] Alex Graves. Generating sequences with recurrent neural networks, 2014. 5

[28] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis, 2022. 6

[29] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 98–107, 2022. 8

[30] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017. 4, 8

[31] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. 6

[32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 6

[33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 2, 3, 4, 8

[34] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 8

[35] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 3

[36] Guillermo Iglesias, Edgar Talavera, and Alberto Díaz-Álvarez. A survey on gans for computer vision: Recent research, analysis and taxonomy. *Computer Science Review*, 48:100553, 2023. 4

[37] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), 2017. 8

[38] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 8

[39] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis, 2023. 6, 7

[40] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 7

[41] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. 5

[42] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 3, 5, 7

[43] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2021. 5

[44] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018. 7

[45] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. 6

[46] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, Fort Lauderdale, FL, USA, 2011. PMLR. 5

[47] Yann Lecun, Sumit Chopra, and Raia Hadsell. *A tutorial on energy-based learning*. 2006. 5

[48] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017. 7

[49] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr. Controllable text-to-image generation, 2019. 6

[50] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 8

[51] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, 2022. 8

[52] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 8

[53] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention, 2016. 6

[54] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 7

[55] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks, 2018. 4

[56] Jiquan Ngiam, Zhenghao Chen, Pang Koh, and Andrew Ng. Learning deep energy models. pages 1105–1112, 2011. 5

[57] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022. 6

[58] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022. 7

[59] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization, 2016. 4

[60] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021. 5

[61] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature

learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 8

[62] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016. 4

[63] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 6, 7

[64] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. 6, 7

[65] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 6

[66] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 6, 7

[67] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis, 2016. 6, 7

[68] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 181–190, 2019. 8

[69] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016. 5

[70] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2, 6, 7

[71] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows, 2020. 8

[72] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021. 1, 6, 8

[73] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022. 1, 8

[74] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 6, 7

[75] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall, 2018. 6

[76] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. 4, 6

[77] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications, 2017. 5

[78] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis, 2023. 6

[79] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017. 8

[80] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019. 8

[81] Taizhang Shang, Qiuju Dai, Shengchen Zhu, Tong Yang, and Yandong Guo. Perceptual extreme super resolution network with receptive field block, 2020. 7

[82] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 2, 3

[83] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. 3, 4

[84] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models, 2020. 4

[85] Yang Song and Diederik P. Kingma. How to train your energy-based models, 2021. 5

[86] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation, 2019. 3

[87] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. 4

[88] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 6

[89] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 8

[90] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation, 2016. 5

[91] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016. 5

[92] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks, 2016. 5

[93] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016. 5

[94] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 7

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 5

[96] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks, 2018. 7

[97] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011. 3

[98] Julian Wyatt, Adam Leach, Sebastian M. Schmon, and Chris G. Willcocks. Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 649–655, 2022. 8

[99] Yi Xiao, Qiangqiang Yuan, Kui Jiang, Jiang He, Xianyu Jin, and Liangpei Zhang. Ediffsr: An efficient diffusion probabilistic model for remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–14, 2024. 7, 8

[100] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2023. 6

[101] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58:101552, 2019. 4

[102] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8

[103] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows, 2021. 8

[104] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan, 2022. 7

[105] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022. 6, 7

[106] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022. 6, 7

[107] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, pages 1100–1109. PMLR, 2016. 5

[108] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion models in generative ai: A survey, 2023. 6

[109] Dafeng Zhang, Feiyu Huang, Shizhuo Liu, Xiaobing Wang, and Zhezhu Jin. Swinfir: Revisiting the swinir with fast fourier convolution and improved training for image super-resolution, 2023. 7

[110] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2017. 6, 7

[111] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019. 4

[112] Leheng Zhang, Yawei Li, Xingyu Zhou, Xiaorui Zhao, and Shuhang Gu. Transcending the limit of local window: Advanced super-resolution transformer with adaptive token dictionary, 2024. 7

[113] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network, 2017. 4

[114] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks, 2021. 8