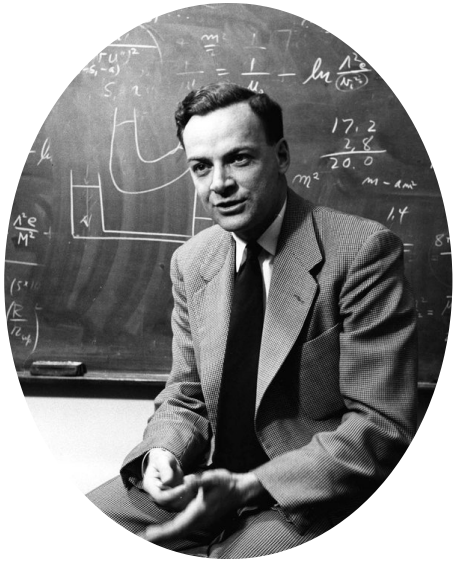
Denoising Diffusion Probabilistic Models

Jonathan Ho - Ajay Jain - Pieter Abbeel

Advanced Machine Learning Course

March 2021

Generative Modelling - Motivation



What I cannot create, I do not understand.

Richard Feynman.

———

Credits: Stefano Ermon, Aditya Grover, CS236 Deep Generative Models, Stanford.

Generative Modelling

Computer Graphics

- Computer graphics: "How to generate images with a computer?"
- High Level descriptions (e.g. shape, color, materials, physical properties.)
- A "Physics simulator" for rendering.

———

Credits: Stefano Ermon, Aditya Grover, CS236 Deep Generative Models, Stanford.

Generative Modelling

Data driven

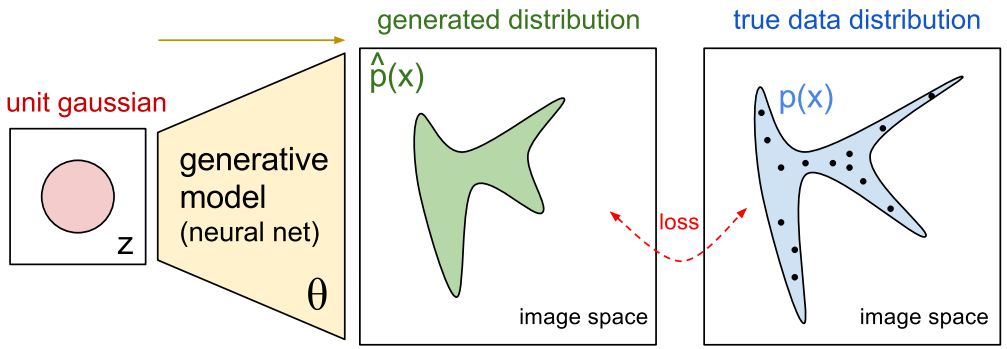
- Data Driven Generative Modelling: "*How to generate ~~images~~ data with a computer without strong human supervision?*"
- The descriptions of what we want to create is implicitely defined by a (very large) dataset.
- A statistical model for "rendering".

CelebA Dataset

Generated samples

Statistical Generative Modelling

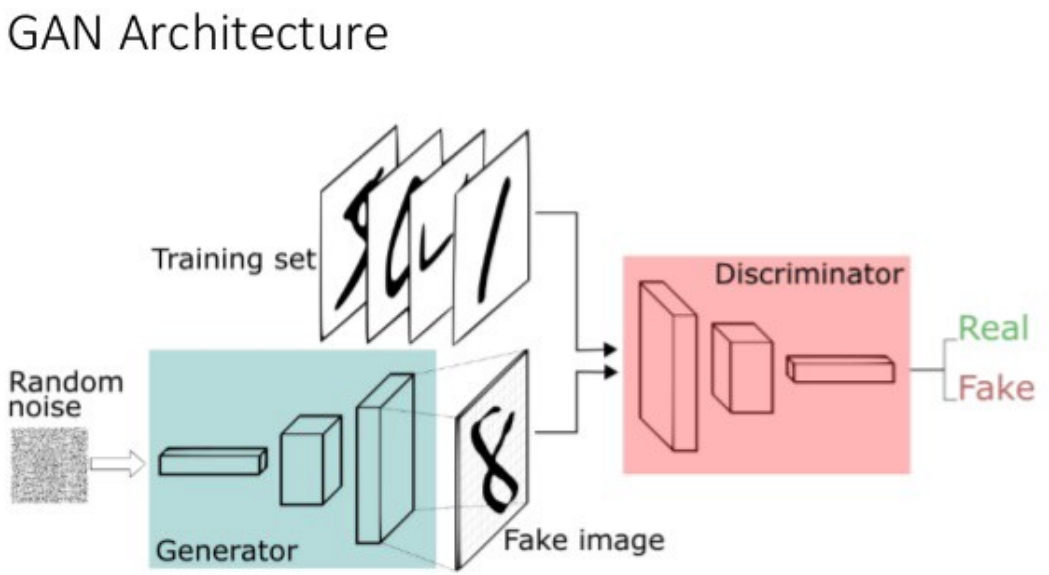
Formal Definition



- A statistical generative model is a probability distribution from which we can sample (e.g. $\mathcal{N}(\mu, \Sigma)$, GAN, VAE, Bayesian networks, ...).
- Ideally, it should sample objects that are similar to the one in the dataset; It models the joint distribution of the data $p(y, x)$ (or simply $p(x)$).
- In opposition, discriminative models represent the conditionnal distribution $p(y|x)$.

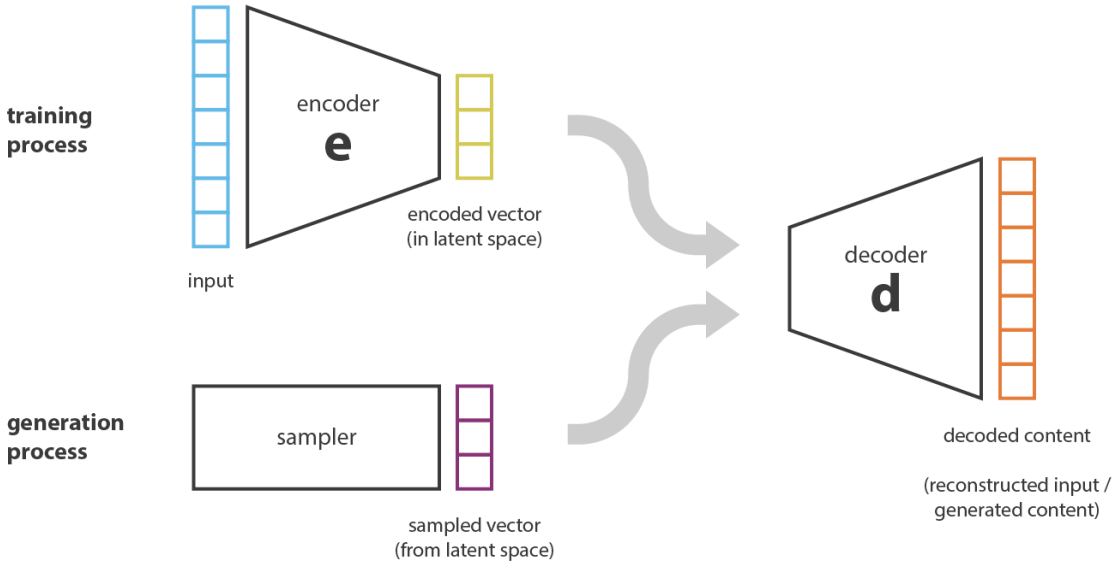
Deep Generative Modelling

State of the art - GANs



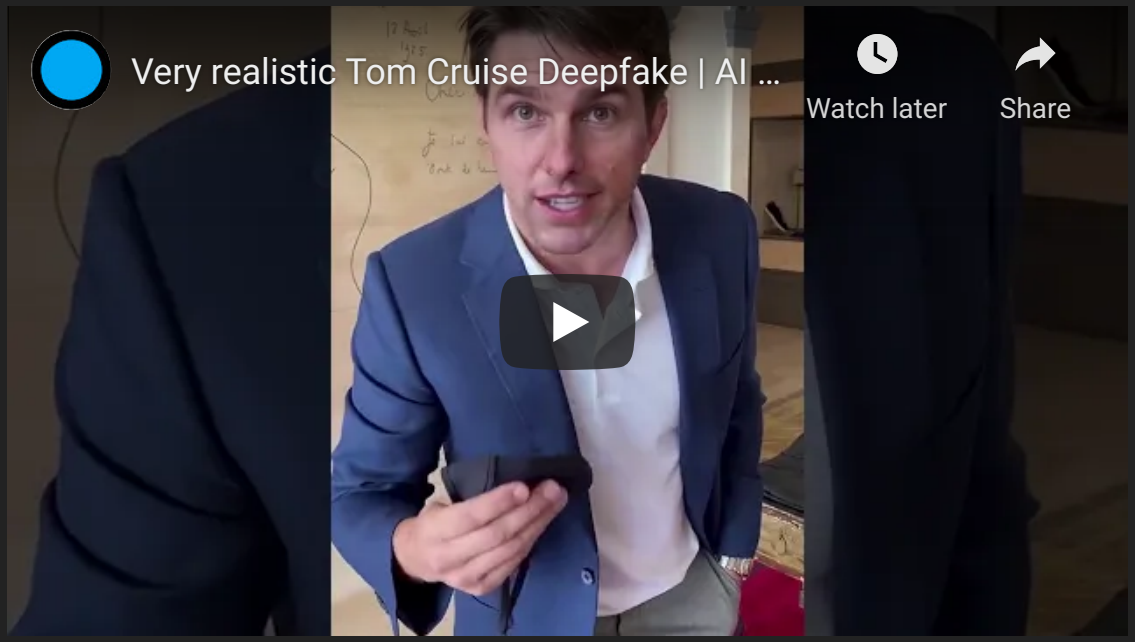
Deep Generative Modelling

State of the art - VAEs



Deep Generative Modelling

Deep Fake



Deep Generative Modelling

Image Generation

Deep Generative Modelling

Conditionnal Image Generation - DALL-E (OpenAI)

TEXT PROMPT an armchair in the shape of an avocado. an armchair imitating an avocado.

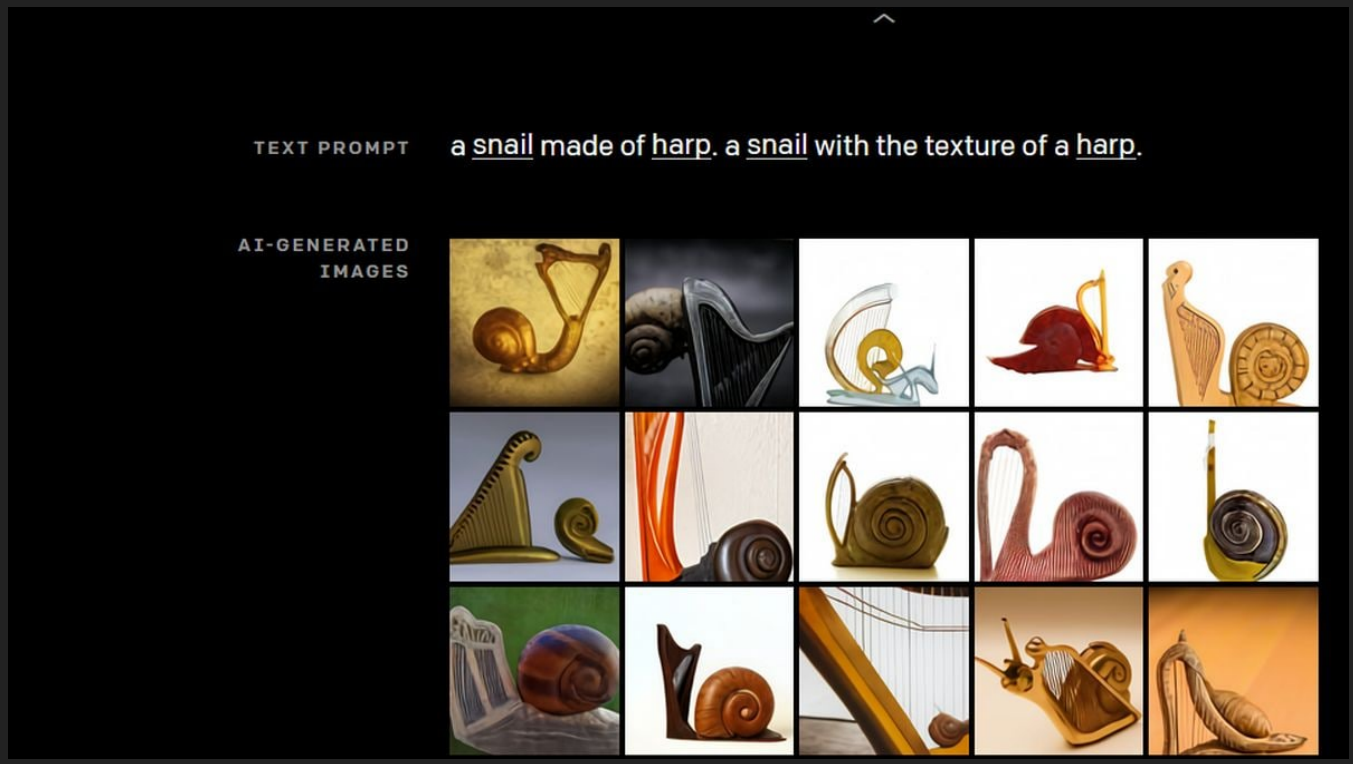
AI-GENERATED IMAGES

In the preceding visual, we explored DALL·E's ability to generate fantastical objects by combining two unrelated ideas. Here, we explore its ability to take inspiration from an unrelated idea while respecting the form of the thing being designed, ideally producing an object that appears to be practically functional. We found that prompting DALL·E with the phrases "in the shape of," "in the form of," and "in the style of" gives it the ability to do this.

When generating some of these objects, such as "an armchair in the shape of an avocado", DALL·E appears to relate the shape of a half avocado to the back of the chair, and the pit of the avocado to the cushion. We find that DALL·E is susceptible to the same kinds of mistakes mentioned in the previous visual.

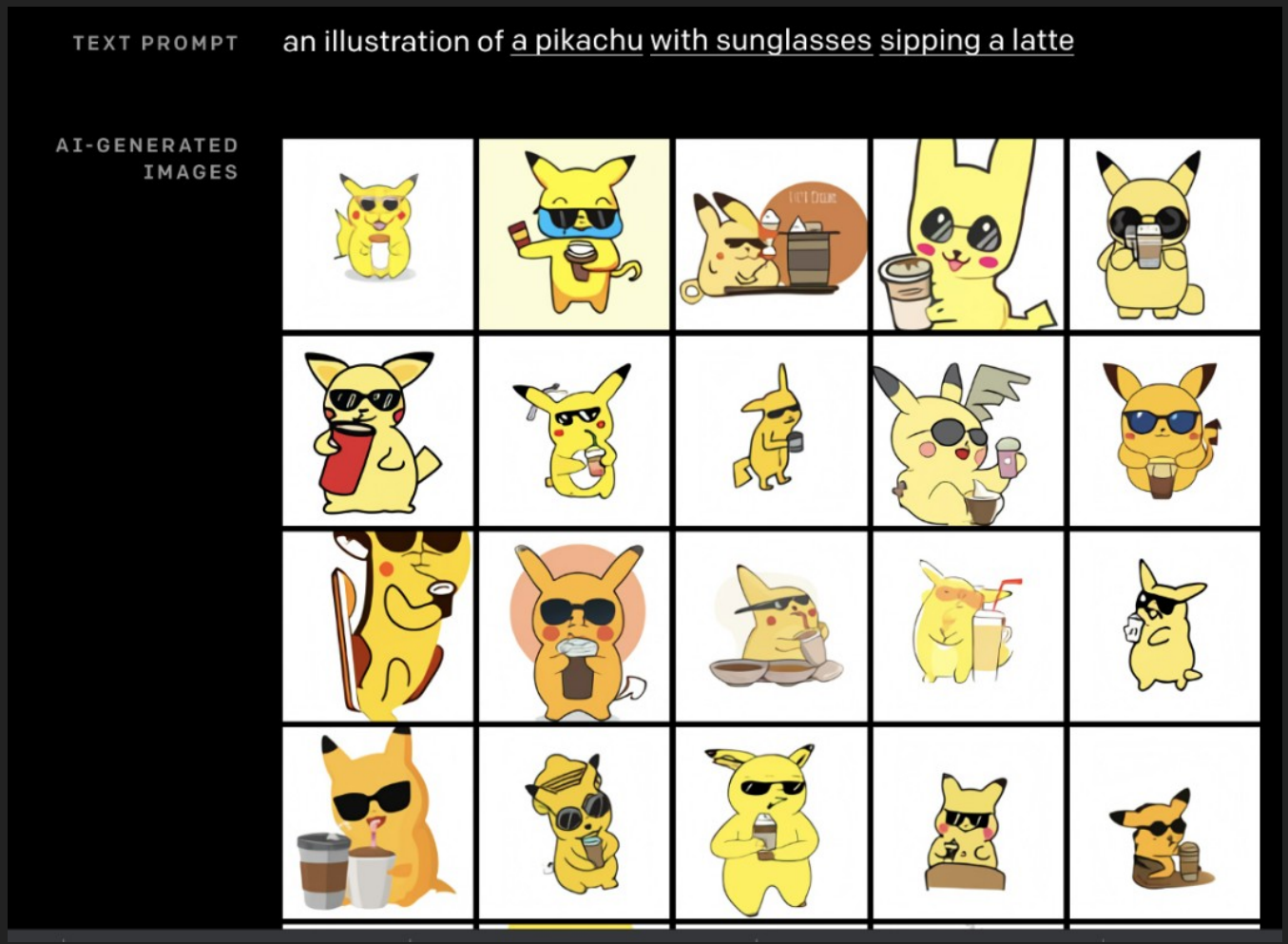
Deep Generative Modelling

Conditionnal Image Generation - DALL-E (OpenAI)



Deep Generative Modelling

Conditionnal Image Generation - DALL-E (OpenAI)



Back to the paper

A new class of deep generative models

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

A new class of deep generative models

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

A new class of deep generative models

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

A new class of deep generative models

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

Diffusion - Physical process

- Wikipedia: *Diffusion is the net movement of anything (for example, atoms, ions, molecules, energy) from a region of higher concentration to a region of lower concentration. The word diffusion derives from the Latin word, diffundere, which means "to spread out."*
- Diffusion continuously increases the entropy of the complete system.
- Idea: Learn a model that reverses the diffusion process to go from a chaotic to a structure configuration.

Diffusion - As a Markov Chain

Markov assumption

- The current state of the world depends only on its immediate previous state(s), i.e., $\mathbf{x}_t$ depends on only a bounded subset of $\mathbf{x}_{0:t-1}$.
- Random processes that satisfy this assumption are called **Markov processes**.

———

Slide courtesy of Gilles Louppe INFO8006

Diffusion - As a Markov Chain

Markov assumption

- The current state of the world depends only on its immediate previous state(s), i.e., $\mathbf{x}_t$ depends on only a bounded subset of $\mathbf{x}_{0:t-1}$.
- Random processes that satisfy this assumption are called **Markov processes**.

First-order Markov processes

- Markov processes such that

$$\mathbf{p}(\mathbf{x}_t|\mathbf{x}_{0:t-1}) = \mathbf{p}(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

- i.e., $\mathbf{x}_t$ and $\mathbf{x}_{0:t-2}$ are conditionally independent given $\mathbf{x}_{t-1}$.

———

Slide courtesy of Gilles Louppe INFO8006

Diffusion - As a Markov Chain

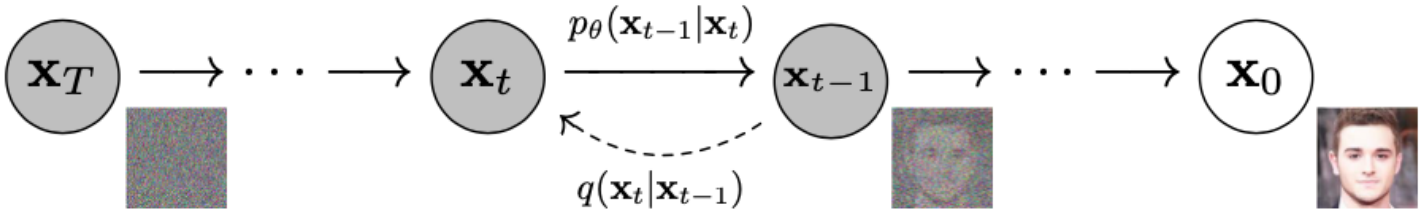


Figure 2: The directed graphical model considered in this work.

Diffusion - As a Markov Chain

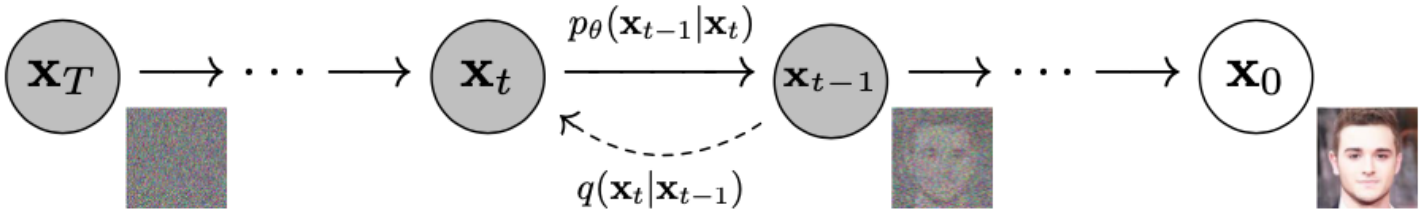


Figure 2: The directed graphical model considered in this work.

Diffusion models [53] are latent variable models of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latents of the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The joint distribution $p_\theta(\mathbf{x}_{0:T})$ is called the *reverse process*, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{1}$$

Diffusion - As a Markov Chain

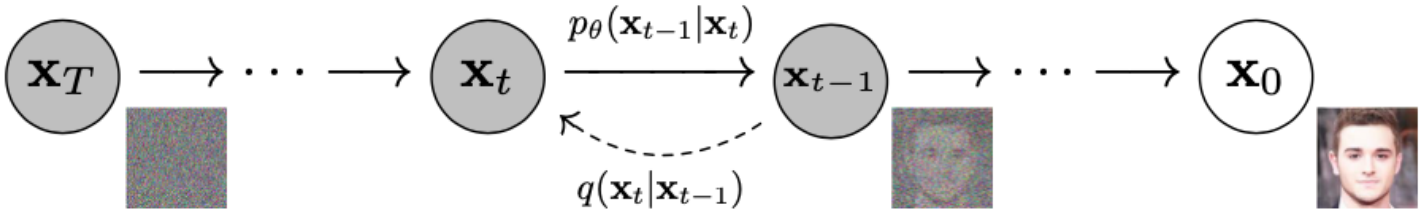


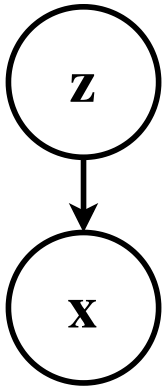
Figure 2: The directed graphical model considered in this work.

What distinguishes diffusion models from other types of latent variable models is that the approximate posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, called the *forward process* or *diffusion process*, is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2}$$

Variational inference

Latent variable model



Consider for now a **prescribed latent variable model** that relates a set of observable variables $\mathbf{x} \in \mathcal{X}$ to a set of unobserved variables $\mathbf{z} \in \mathcal{Z}$.

- $p(z)$ is the prior distribution, defined by domain knowledge.
- $p(x|z)$ is defined by your model.

———

Slide courtesy of Gilles Louppe INFO8010

The probabilistic model defines a joint probability distribution $p(\mathbf{x}, \mathbf{z})$, which decomposes as

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}).$$

If we interpret $\mathbf{z}$ as causal factors for the high-dimension representations $\mathbf{x}$, then sampling from $p(\mathbf{x}|\mathbf{z})$ can be interpreted as **a stochastic generating process** from $\mathcal{Z}$ to $\mathcal{X}$.

For a given model $p(\mathbf{x}, \mathbf{z})$, inference consists in computing the posterior

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}.$$

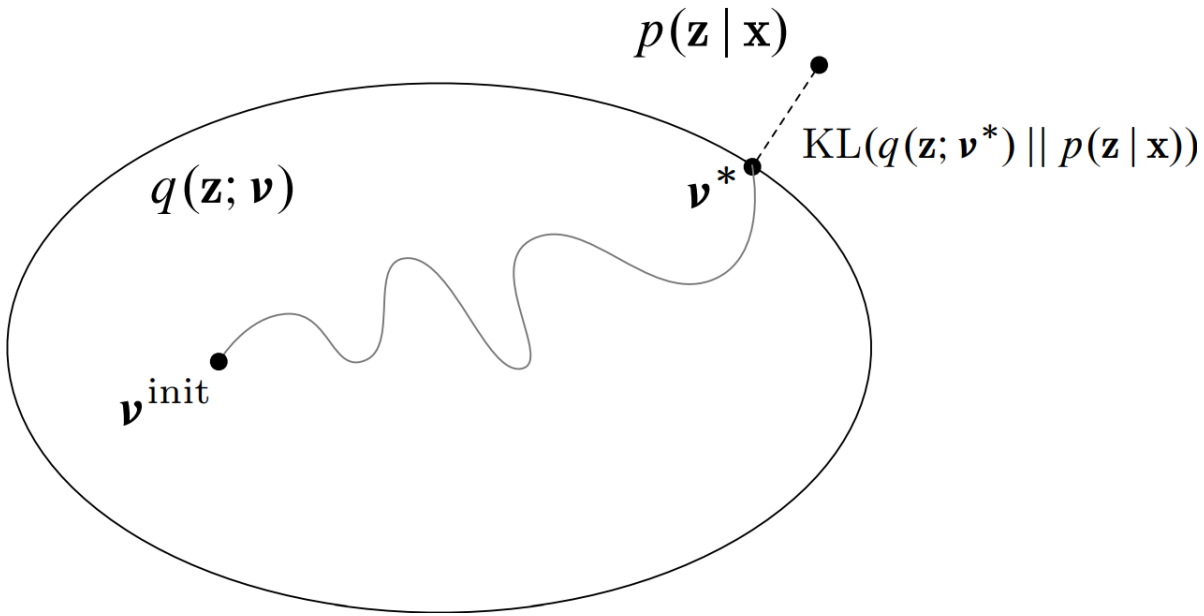
For most interesting cases, this is usually intractable since it requires evaluating the evidence

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}.$$

———

Slide courtesy of Gilles Louppe INFO8010

Variational inference



Variational inference turns posterior inference into an optimization problem.

- Consider a family of distributions $q(\mathbf{z}|\mathbf{x}; \nu)$ that approximate the posterior $p(\mathbf{z}|\mathbf{x})$, where the variational parameters ν index the family of distributions.
- The parameters ν are fit to minimize the KL divergence between $p(\mathbf{z}|\mathbf{x})$ and the approximation $q(\mathbf{z}|\mathbf{x}; \nu)$.

———

Slide courtesy of Gilles Louppe INFO8010

Formally, we want to minimize

$$\begin{aligned} KL(q(\mathbf{z}|\mathbf{x};\nu)||p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\nu)}\left[\log \frac{q(\mathbf{z}|\mathbf{x};\nu)}{p(\mathbf{z}|\mathbf{x})}\right] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\nu)}\left[\log q(\mathbf{z}|\mathbf{x};\nu) - \log p(\mathbf{x},\mathbf{z})\right] + \log p(\mathbf{x}). \end{aligned}$$

For the same reason as before, the KL divergence cannot be directly minimized because of the $\log p(\mathbf{x})$ term.

———

Slide courtesy of Gilles Louppe INFO8010

However, we can write

$$\begin{aligned}\arg\min_{\nu} KL(q(\mathbf{z}|\mathbf{x};\nu)||p(\mathbf{z}|\mathbf{x})) &= \arg\min_{\nu} \log p(\mathbf{x}) - \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\nu)}\left[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x};\nu)\right] \\ &= \arg\max_{\nu} \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\nu)}\left[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x};\nu)\right]}_{\mathrm{ELBO}(\mathbf{x};\nu)}\end{aligned}$$

where $\mathrm{ELBO}(\mathbf{x};\nu)$ is called the **evidence lower bound objective**.

- Since $\log p(\mathbf{x})$ does not depend on ν, it can be considered as a constant, and minimizing the KL divergence is equivalent to maximizing the evidence lower bound, while being computationally tractable.
- Given a dataset $\mathbf{d} = \{\mathbf{x}_i | i = 1, ..., N\}$, the final objective is the sum $\sum_{\{\mathbf{x}_i \in \mathbf{d}\}} \mathrm{ELBO}(\mathbf{x}_i;\nu)$.

———

Slide courtesy of Gilles Louppe INFO8010

Remark that

$$\begin{aligned}\mathbf{ELBO}(\mathbf{x};\nu) &= \mathbb{E}_{q(\mathbf{z};|\mathbf{x}\nu)}\left[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x};\nu)\right] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\nu)}\left[\log p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x};\nu)\right] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\nu)}\left[\log p(\mathbf{x}|\mathbf{z})\right] - KL(q(\mathbf{z}|\mathbf{x};\nu)||p(\mathbf{z}))\end{aligned}$$

Therefore, maximizing the ELBO:

- encourages distributions to place their mass on configurations of latent variables that explain the observed data (first term);
- encourages distributions close to the prior (second term).

———

Slide courtesy of Gilles Louppe INFO8010

Parametric generative model

Suppose we have:

- The "likelihood" $p(\mathbf{x}|\mathbf{z};\theta)$ is parameterized by θ and you would like to fit the best theta for your observation.
- The approximate posterior $q(\mathbf{z}|\mathbf{x};\nu)$ is parameterized by ν and you would like to fit well the posterior of the latent provided the generative model.
- You know/assume the prior distribution over latents $p(z)$.

Parametric generative model

Suppose we have:

- The "likelihood" $p(\mathbf{x}|\mathbf{z};\theta)$ is parameterized by θ and you would like to fit the best theta for your observation.
- The approximate posterior $q(\mathbf{z}|\mathbf{x};\nu)$ is parameterized by ν and you would like to fit well the posterior of the latent provided the generative model.
- You know/assume the prior distribution over latents $p(z)$.
- The "likelihood" + the prior defines a generative model of the observations.

As before, we can use variational inference, but to jointly optimize the "likelihood" and the approximate posterior parameters θ and φ.

We want

$$\begin{aligned}\theta^*, \varphi^* &= \arg\max_{\theta,\varphi} \mathrm{ELBO}(\mathbf{x}; \theta, \varphi) \\ &= \arg\max_{\theta,\varphi} \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\varphi)} \left[\log p(\mathbf{x}, \mathbf{z}; \theta) - \log q(\mathbf{z}|\mathbf{x}; \varphi)\right] \\ &= \arg\max_{\theta,\varphi} \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\varphi)} \left[\log p(\mathbf{x}|\mathbf{z}; \theta)\right] - KL(q(\mathbf{z}|\mathbf{x}; \varphi)||p(\mathbf{z})).\end{aligned}$$

- Given some generative model with parameters θ, we want to put the mass of the latent variables, by adjusting φ, such that they explain the observed data, while remaining close to the prior.
- Given some approximate posterior φ, we want to put the mass of the observed variables, by adjusting θ, such that they are well explained by the latent variables.

———

Modified from Gilles Louppe INFO8010

Variational inference on diffusion models

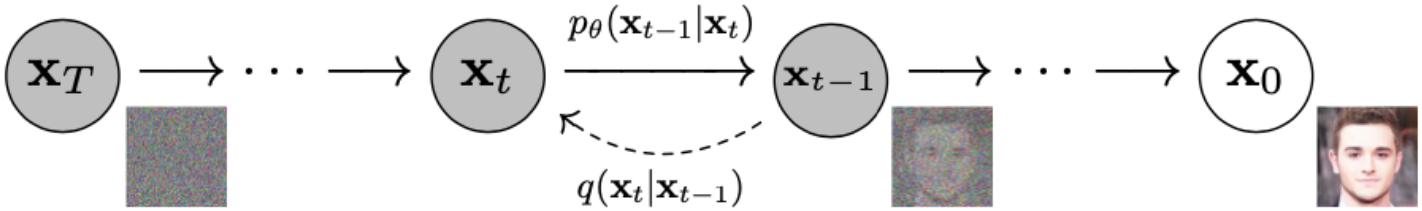


Figure 2: The directed graphical model considered in this work.

- $x_1, ..., x_T$ are the latent variables and x_0 the observation.
- $p_\theta(x_0, x_1, ..., x_T)$ parameterized as $x_T \sim \mathcal{N}(0, I)$ and $x_{t-1}|x_t \sim \mathcal{N}(\mu_{\theta,t}(x_t), \Sigma_{\theta,t}(x_t))$.
- Use variational inference to learn θ with diffusion as an approximate posterior $q(x_0|x_1, ..., x_T)$ fixed in advance as $x_t|x_{t-1} \sim \mathcal{N}(x_{t-1}, \beta_t I)$.

Variational inference on diffusion models

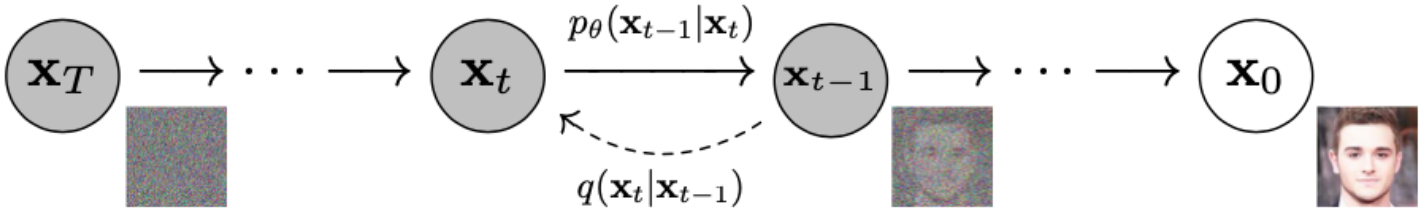


Figure 2: The directed graphical model considered in this work.

- $x_1, ..., x_T$ are the latent variables and x_0 the observation.
- $p_\theta(x_0, x_1, ..., x_T)$ parameterized as $x_T \sim \mathcal{N}(0, I)$ and $x_{t-1}|x_t \sim \mathcal{N}(\mu_{\theta,t}(x_t), \Sigma_{\theta,t}(x_t))$.
- Use variational inference to learn θ with diffusion as an approximate posterior $q(x_0|x_1, ..., x_T)$ fixed in advance as $x_t|x_{t-1} \sim \mathcal{N}(x_{t-1}, \beta_t I)$.
- If β_t are small compared to x_{t-1} then the assumption of Normal transition for $p_\theta(x_{t-1}|x_t)$ holds.

Variational inference on diffusion models

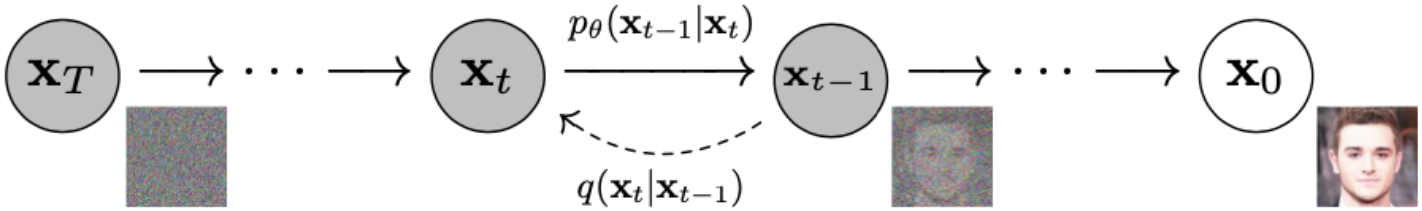


Figure 2: The directed graphical model considered in this work.

- $x_1, ..., x_T$ are the latent variables and x_0 the observation.
- $p_\theta(x_0, x_1, ..., x_T)$ parameterized as $x_T \sim \mathcal{N}(0, I)$ and $x_{t-1}|x_t \sim \mathcal{N}(\mu_{\theta,t}(x_t), \Sigma_{\theta,t}(x_t))$.
- Use variational inference to learn θ with diffusion as an approximate posterior $q(x_0|x_1, ..., x_T)$ fixed in advance as $x_t|x_{t-1} \sim \mathcal{N}(x_{t-1}, \beta_t I)$.
- If β_t are small compared to x_{t-1} then the assumption of Normal transition for $p_\theta(x_{t-1}|x_t)$ holds.
- This means T is large (1000 in the paper).

ELBO for diffusion models - Loss

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

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L \quad (3)$$

Indeed we have:

$$\begin{aligned}
\mathbb{E}_q[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] &= \mathbb{E}_q[\log \frac{p_\theta(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} + \log p_\theta(x_0)] \\
&= \log p_\theta(x_0) - \underbrace{KL(p_\theta(x_{1:T}|x_0)||q(x_{1:T}|x_0))}_{\geq 0} \\
&\leq \log p_\theta(x_0)
\end{aligned}$$

- Minimizing the loss is equivalent to maximizing the ELBO and so to maximize the likelihood of the model.

ELBO for diffusion models - Loss

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

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L \quad (3)$$

Indeed we have:

$$\begin{aligned}
\mathbb{E}_q[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] &= \mathbb{E}_q[\log \frac{p_\theta(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} + \log p_\theta(x_0)] \\
&= \log p_\theta(x_0) - \underbrace{KL(p_\theta(x_{1:T}|x_0)||q(x_{1:T}|x_0))}_{\geq 0} \\
&\leq \log p_\theta(x_0)
\end{aligned}$$

- Minimizing the loss is equivalent to maximizing the ELBO and so to maximize the likelihood of the model.

Can you see potential drawbacks of this loss function?

ELBO for diffusion models - Loss

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

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L \quad (3)$$

Indeed we have:

$$\begin{aligned}
\mathbb{E}_q[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] &= \mathbb{E}_q[\log \frac{p_\theta(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} + \log p_\theta(x_0)] \\
&= \log p_\theta(x_0) - \underbrace{KL(p_\theta(x_{1:T}|x_0)||q(x_{1:T}|x_0))}_{\geq 0} \\
&\leq \log p_\theta(x_0)
\end{aligned}$$

- Minimizing the loss is equivalent to maximizing the ELBO and so to maximize the likelihood of the model.

Can you see potential drawbacks of this loss function?

The expectation is over the posterior $q(x_{1:T}|x_0)$ which may be expensive to sample and produces samples with high variance!

Reduced variance ELBO

- But we can sample from (and evaluate) $q(x_{t-1}|x_t, x_0) = (x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I)$.
- $\tilde{\mu}$ and $\tilde{\beta}_t$ can be computed in closed form from the variance schedule β_t.

Reduced variance ELBO

- But we can sample from (and evaluate) $q(x_{t-1}|x_t, x_0) = (x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I)$.
- $\tilde{\mu}$ and $\tilde{\beta}_t$ can be computed in closed form from the variance schedule β_t.
- Thus authors suggest to rewrite the loss function as follows:

$$\mathbb{E}_q \Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Bigg] \quad (5)$$

Reduced variance ELBO

- But we can sample from (and evaluate) $q(x_{t-1}|x_t, x_0) = (x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I)$.
- $\tilde{\mu}$ and $\tilde{\beta}_t$ can be computed in closed form from the variance schedule β_t.
- Thus authors suggest to rewrite the loss function as follows:

$$\mathbb{E}_q \Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Bigg] \quad (5)$$

- This estimator has a reduced variance compared to the "classical" ELBO.
- All KL are comparison between Gaussians $\rightarrow$ can be computed in closed form!

Reduced variance ELBO

- But we can sample from (and evaluate) $q(x_{t-1}|x_t, x_0) = (x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I)$.
- $\tilde{\mu}$ and $\tilde{\beta}_t$ can be computed in closed form from the variance schedule β_t.
- Thus authors suggest to rewrite the loss function as follows:

$$\mathbb{E}_q \Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Bigg] \quad (5)$$

- This estimator has a reduced variance compared to the "classical" ELBO.
- All KL are comparison between Gaussians $\rightarrow$ can be computed in closed form!
- We can approximate the sum via Monte Carlo (uniform distribution on t) and avoid evaluating the full Markov Chain for each training loop.

Simplified loss function

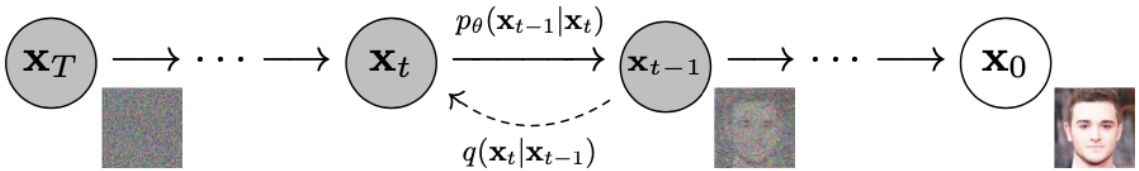


Figure 2: The directed graphical model considered in this work.

Second, to represent the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$, we propose a specific parameterization motivated by the following analysis of L_t. With $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$, we can write:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{8}$$

- We just have to predict the $\mu_{x_{t-1}}$ from x_t (here with a neural network).

Simplified loss function

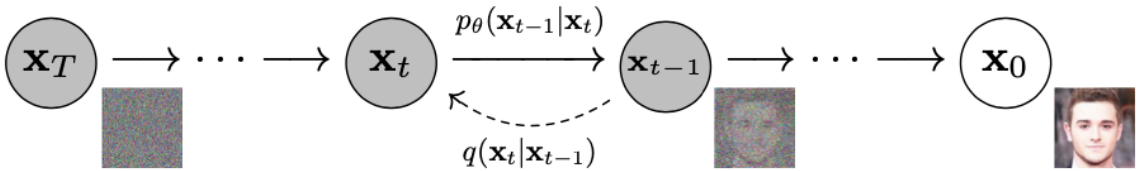


Figure 2: The directed graphical model considered in this work.

Second, to represent the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$, we propose a specific parameterization motivated by the following analysis of L_t. With $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$, we can write:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{8}$$

- We just have to predict the $\mu_{x_{t-1}}$ from x_t (here with a neural network).
- And we can go one step further and explictly parameterized $\mu_{x_{t-1}}$ as:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t))\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \tag{11}$$

Simplified loss function

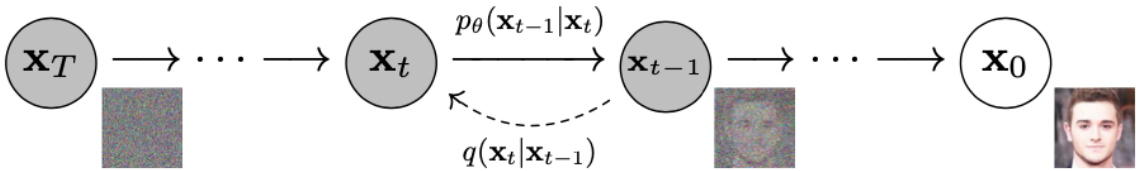


Figure 2: The directed graphical model considered in this work.

Second, to represent the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$, we propose a specific parameterization motivated by the following analysis of L_t. With $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$, we can write:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{8}$$

- We just have to predict the $\mu_{x_{t-1}}$ from x_t (here with a neural network).
- And we can go one step further and explictly parameterized $\mu_{x_{t-1}}$ as:

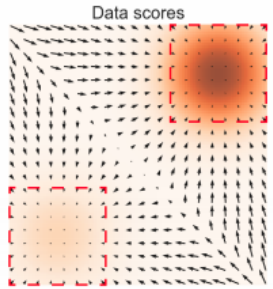
$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t))\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \tag{11}$$

- We could also parameterize it as:

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{12}$$

Denoising score matching

- Another way to parameterize a data distribution is via the gradient of the distribution $s_\theta(x) = \nabla_x p(x)$.



- We can then sample from these models with MCMC or Langevin dynamics. *In physics, Langevin dynamics is an approach to the mathematical modeling of the dynamics of molecular systems. It was originally developed by French physicist Paul Langevin. The approach is characterized by the use of simplified models while accounting for omitted degrees of freedom by the use of stochastic differential equations.*

———

Generative Modeling by Estimating Gradients of the Data Distribution, Yang Song and Stefano Ermon.

Denoising score matching

- Another way to parameterize a data distribution is via the gradient of the distribution.
- The score can be learned with the following loss function:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\|_2^2].$$

- Which is similar to the loss function

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \tag{12}$$

Traning & Sampling with Langevin dynamics

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

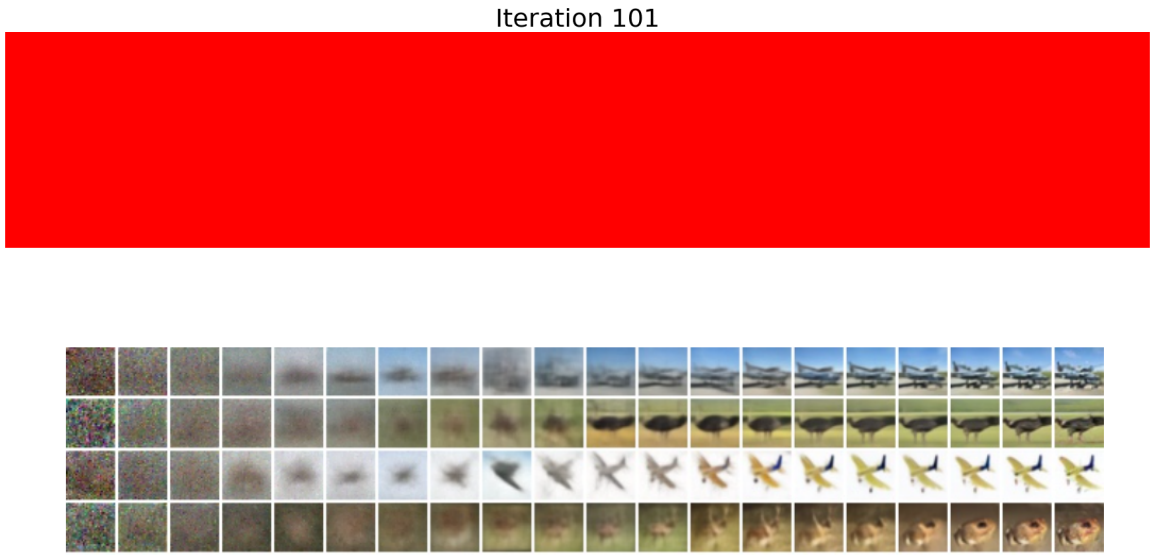


Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

Results

Figure 3: LSUN Church samples. FID=7.89

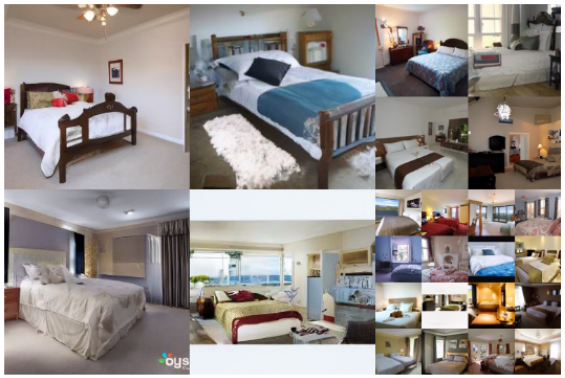


Figure 4: LSUN Bedroom samples. FID=4.90

Results

Figure 7: 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 $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Results

Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

Discussion

- The paper is strongly based on Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." International Conference on Machine Learning. PMLR, 2015.
- Impressive engineering to make it work with (very large) deep neural networks.
- Compared to GANs and VAEs we have only one neural network (shared for all time t).
- Sampling is slow compared to GANs and VAEs.

To go further

- Score-Based Generative Modeling through Stochastic Differential Equations. Song, Yang, et al. arXiv preprint arXiv:2011.13456 (2020)..
- Improved Denoising Diffusion Probabilistic Models. Nichol, Alex, and Prafulla Dhariwal. arXiv preprint arXiv:2102.09672 (2021).