Denoising Diffusion Implicit Models

Generative Modelling: Project report

Thomas Dujardin, Julien Gaubil, Thibault Richard

Contents

1 Denoising Diffusion Implicit Models

1.1 Background and motivations

Denoising Diffusion Implicit Models (DDIM) [10] are a development from their precursor, Denoising Diffusion Probabilistic Models (DDPM) [5]. In this subsection, we provide a short introduction to DDPMs.

DDPM - forward & backward At the core of DDPMs lies a Markovian forward diffusion process that gradually introduces noise to the data, represented as x_0:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1},\ (1-\alpha_{t-1})\mathbf{I}\right) \quad \text{then} \quad q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{1}$$

where $q(x_{1:T}|x_0)$ is the inference distribution over the latent variables $x_{1:T}$. This leads to:

$$q(x_t|x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t\mathbf{I})\right) \iff x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,\mathbf{I}) \tag{2}$$

where $\alpha_i = 1-\beta_i$ and $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$. A backward process $q(x_{t-1}|x_t)$ is established using Bayes' theorem, yielding a closed formula:

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

Since the original data x_0 is unknown during generative process, *i.e.* the transition from the latent space to the image distribution $q(x_0)$, an estimation of the backward process is needed. This is particularly challenging as the mean of the process depends on x_0.

DDPM objective To address this, a parametric process is implemented to estimate the backward process as accurately as possible. This parametric backward process estimation is written as $p_\theta(x_{0:T}) := p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$ with $p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t,t),\Sigma_\theta)$. As described in [5] [eq. 3], θ is learned using a variational bound on the negative log-likelihood. This variational bound is tractable and can be minimized with respect to θ:

$$\mathbb{E}(-log\ p_\theta(x_0)) \leq \underbrace{\mathbb{E}_{q(x_{0:T})}\left(-log\ \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right)}_{ELBO} \tag{3}$$

The ELBO can be rewritten in a more convenient way that involves KL divergences, that do not depend on the forward process used, but only of the conditionals $q(x_t|x_0)$:

$$\mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}}\underbrace{- \log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}] \tag{4}$$

The authors of DDPM chose to keep a fixed variance, leading to a formula for the K-L Divergence (which is the loss function at time t, L_{t-1}) that depends solely on the means of the Gaussians. It can further be reframed using a reparametrization trick based on 2's stochastic formulation. The goal becomes learning a noise function $\epsilon_\theta(x_t,t)$ that can best approximate ϵ from 2:

$$L_{t-1} = ||\epsilon - \epsilon_\theta(x_t,t)||_2^2$$

Once ϵ_θ is learned, the backward process is employed for inference. This process is Markovian:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right) + \sigma_t\mathbf{z}\ with\ \mathbf{z} \sim \mathcal{N}(0,\mathbf{I})$$

The stochastic term $\sigma_t\mathbf{z}$, where σ_t fixed hyperparameters, is linked to the fixed variance in $p_\theta(x_{t-1},x_t)$.

Motivations Despite their impressive photorealistic generated samples, DDPMs suffer from a major limitation: they require a large number of sampling steps T (typically 1000) to approximate a Gaussian reverse process. Due to the Markovian nature of DDPM's backward process, skipping too many steps during inference results in performance degradation. Generating one sample is therefore extremely long, (around 1 second for a 32×32 image), which scales with the resolution of the images. DDPM are in particular orders of magnitude slower than their GANs and VAEs counterparts.

1.2 Details

Overview Denoising Diffusion Implicit Models (DDIMs) address this issue by employing a different, non-Markovian diffusion process that enables skipping many steps during denoising. DDIMs share the same objective function as DDPMs during training, while providing faster sampling using the same training objective as DDPMs.

Non-Markovian forward process DDIMs forward process is non-Markovian since each latent variable x_t depends not only of x_{t-1} but also on x_0. The generative (backward) transitions are indeed defined as:

$$q_\sigma(x_{t-1}|x_t,x_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}.\frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}},\ \sigma_t^2 I_d\right) \tag{5}$$

where $\sigma \in (\mathbb{R}_+)^D$ is an hyperparameter. The inference distribution is then defined using a backward probabilistic decomposition of $q_\sigma(x_{1:T}|x_0)$:

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0)\prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0) \tag{6}$$

The forward process can then be derived using Bayes' rule and is indeed non-Markovian. Using a non-Markovian forward process enables to model a larger families of forward processes that include the family of Markovian forward processes. In particular, it also models diffusion process whose reverse process is a shorter chain. This process is designed so that the conditional $q_\sigma(x_t|x_0)$ matches the DDPM case, eq.2. Authors show that the choice of the mean function in eq.5 enables for the conditional $q_\sigma(x_t|x_0)$ to match with eq.2.

Backward process When sampling from the backward process, the initial data x_0 is unknown. Therefore, similar to DDPMs, the noise function ϵ at step t from eq.2) is approximated using a parameterized Neural Network $\epsilon_\theta(x_t, t)$. Using eq.2, the initial data x_0 can then be approximated by the function $f_\theta(x_t, t) = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}}$.

From the transition $q_\sigma(x_{t-1}|x_t, f_\theta(x_t, t))$, the DDIM backward process can then be derived as:

$$x_{t-1} = \sqrt{\alpha_{t-1}} f_\theta(x_t, t) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2}\,\epsilon_\theta(x_t, t) + \sigma_t \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}) \tag{7}$$

This expression clearly shows that σ controls the stochasticity in the generative process: $\sigma = \mathbf{0}$ leads to a deterministic forward process. The latent variable x_{t-1} is indeed entirely determined by the knowledge of x_{t-1}, and then by induction $\forall 1 \leq t \leq T,\ x_t$ is entirely determined by the knowledge of x_T, that is sampled according to a Gaussian prior $\mathcal{N}(0, \mathbf{I})$.

Objective function The marginals $p_\theta^{(t)}(x_{t-1}|x_t)$ and the generative process $p_\theta(x_{0:T})$ are defined by:

$$p_\theta^{(t)}(x_{t-1}|x_t) = \begin{cases} \mathcal{N}\left(f_\theta(x_1, 1),\ \sigma_1^2\mathbf{I}\right) & \text{if } t = 1 \\ q_\sigma\left(x_{t-1}|x_t, f_\theta(x_t, t)\right) & \text{else} \end{cases} \quad \text{then} \quad p_\theta(x_{0:T}) = p_\theta(x_T)\prod_{t=1}^{T} p_\theta^{(t)}(x_{t-1}|x_t) \tag{8}$$

The latent variables $q_{\sigma,\tau}(x_{0:T})$ and the DDIM generative inference process $p_\theta(x_{0:T})$ being fixed, similar to DDPMs, the DDIM objective $J_\sigma(\epsilon_\theta)$ is defined according to eq.3. Since conditionals $q_\sigma(x_{t-1}|x_t, x_0)$ and $q_\sigma(x_{t-1}|x_t, f_\theta(x_t, t))$ are Gaussians, this leads to a closed formula. Authors show that it is equivalent to the DDPM objective. This important property implies that a DDIM generative process can be used over an *already trained* DDPM model. Since the backward process is not sampled during training, one can define different DDIM models without having to re-train the weights θ by changing σ.

Accelerated generative process The main advantage of DDIM lies in the fact that it is possible to *accelerate* the generative sampling by only performing a subsequence τ of the steps. Indeed, DDIMs are trained with an objective that is equivalent to the DDPM objective. As discuted in eq.4, this objective does not depend on the forward transitions $q(x_t|x_{t-1})$, but only on the conditionals $q_\sigma(x_t|x_0)$. Since the forward process is not Markovian anymore, it is possible to consider forward processes defined on a subsequence τ of the steps whose length is typically smaller than T without suffering from the degradation that would occur in DDPMs, while still having a T latent variables model $x_{1:T}$. This in turn accelerates the generative process, whose length would be $S = |\tau|$.

Formally, the forward process is defined by $\{x_{\tau_1}, ..., x_{\tau_S}\}$, with the conditionals $q_\sigma(x_{\tau_s}|x_0)$ that match eq.2. The generative process then becomes $x_{\tau_S}, ..., x_{\tau_1}$. It is therefore possible to consider for the DDIM accelerated inference distribution over the latent variables $q_{\sigma,\tau}(x_{0:T})$ and the DDIM generative inference process $p_\theta(x_{0:T})$:

$$\begin{cases} q_{\sigma,\tau}(x_{0:T}) = q_{\sigma,\tau}(x_{\tau_S}) \prod_{s=1}^{S} q_{\sigma,\tau}(x_{\tau_s}|x_{\tau_{s-1}}, x_0) \ \prod_{t\in\bar{\tau}} q_{\sigma,\tau}(x_0|x_t) \\ p_\theta(x_{0:T}) = \underbrace{p_\theta(x_T) \prod_{s=1}^{S} p_\theta(x_{\tau_{s-1}}|x_{\tau_s})}_{\text{sampled during generative process}} \ \underbrace{\prod_{t\in\bar{\tau}} p_\theta(x_0|x_t)}_{\text{not sampled}} \end{cases} \tag{9}$$

where $\bar{\tau} = \{1, ..., T\} \setminus \tau$. The variational objective can then be derived using eq.3, which is the same formula as DDPM/non-accelerated DDIM. The latent variables $x_t,\ t \in \bar{\tau}$ are not sampled during the accelerated generative process, but are however still used in the objective.

Similar to the non-accelerated DDIMs, authors show that plugging the DDIM accelerated inference distribution over the latent variables $q_{\sigma,\tau}(x_{0:T})$ and the DDIM accelerated generative process $p_\theta(x_{0:T})$ in the objective eq.3 is equivalent to DDPM objective. This enables to use the accelerated DDIM procedure to sample from an *already trained* DDPM model. Same as non-accelerated DDIMs, different accelerated DDIMs can be defined by modifying σ or τ without re-training the model.

1.3 Results

Authors evaluate their methods on datasets CelebA, CIFAR10 and Bedrooms, using FID metric. To perform their experiments, they use a DDPM model trained with $T = 1000$ steps.

Subsampling influence They first evaluate the influence of the subsampling used over the quality of the generated samples. They show that the quality increases with the length of the subsequence used τ, but so does the time to generate a sample. Specifically, using 20 to 100 steps in the subsequence yields comparable sample quality to original DDPMs while enabling a $10 - 50\times$ generation speedup over DDPMs. They mention that even though DDPM could achieve satisfying results with fewer steps (typically 100), DDIMs show similar quality using less steps (20 steps). Meanwhile, DDPM achieve by far the worst quality in this low regime number of steps (between 10-50 steps) compared to its ”less stochastic” counterparts (*i.e.* when decreasing the magnitude of σ).

Latent representation, interpolation Starting from the same initial state x_T and applying deterministic generative process with different subsequences τ leads to images having the same high-level features. This indicates that x_T would capture a latent representation of the images, and that the generative process would affect the details that impact the sample quality. Finally, when using a deterministic generative process, it becomes possible to obtain a meaningful interpolation in the image space from an interpolation in the latent space, which authors demonstrate qualitatively.

1.4 Limitations and following work

As previously mentioned, the sequential nature of both DDPM and DDIM inhibits their potential for parallelization. Observing this limitation, Pokle et al. [8] proposed an innovative solution known as Deep Equilibrium DDIMs (DEQ-DDIM). They achieved parallelization of DDIM sampling by reformatting the deterministic backward process of DDIM into a fixed-point problem across the entire chain ($x_{0:T}$). They then applied black-box fixed point solvers to this restructured problem. As a result, a significant increase in inference speed was realized.

DEQ-DDIM is designed to address two challenges in diffusion: standard sampling for FID evaluation and model inversion. For the purpose of demonstrating DEQ-DDIM's effectiveness and for clarity, we will focus on the latter, model inversion.

Model Inversion Given an image $x_0 \in \mathcal{D}$ and a denoising diffusion model $\epsilon_\theta(x_t, t)$ trained on $\mathcal{D}$, the aim is to find a $\hat{x}_T^\star \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ such that $\hat{x}_0$ minimizes $\mathcal{L}(x_0, \hat{x}_0^\star) = ||x_0 - \hat{x}_0||_F^2$. In other words, we are looking for the latent representation $\hat{x}_T^\star$ that is the closest to the original image when passed through the backwards process. We will see that an appropriate reformulation of DDIM allows to solve this problem much faster than with the simple DDIM-subsampling algorithm.

Reformulating DDIM using DEQs DEQ-DDIM [8] proposes to redefine the generative (backwards) process of DDIM as a problem of finding a fixed point. More precisely, the authors start from the original generative process of DDIMs:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t-1}}\epsilon_\theta^{(t)}(x_t),\ t \in [1, ..., T] \tag{10}$$

Rewriting 10 yields:

$$x_{T-k} = \sqrt{\frac{\alpha_{T-k}}{\alpha_T}} x_T + \sum_{T-k}^{T-1} \sqrt{\frac{\alpha_{T-k}}{\alpha_t}} c_1^{(t+1)} \epsilon_\theta^{(t+1)}(x_{t+1}) = h(x_{T-(k-1):T}),\ k \in [0, ..., T] \tag{11}$$

Where $c_1^{(t)}$ is a constant depending on α_t. x_t now depends on $x_{t+1:T}$ instead of only x_{t+1}. With $\tilde{h}(.)$ denoting the operation $h(.)$ applied to all timesteps simultaneously, we have:

$$x_{0:T-1} = \tilde{h}(x_{0:T-1}; x_T) \tag{12}$$

Which is a DEQ [2] with $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as an input. The entire chain $x_{0:T}^\star$ corresponding to this input is found by computing $x_{0:T}^\star = Solver(\tilde{h}(x_{0:T-1}; x_T) - x_{0:T-1})$, where Solver is any efficient solver. DEQ's paper suggests using Anderson's solver [1] which is a quasi-Newton method.

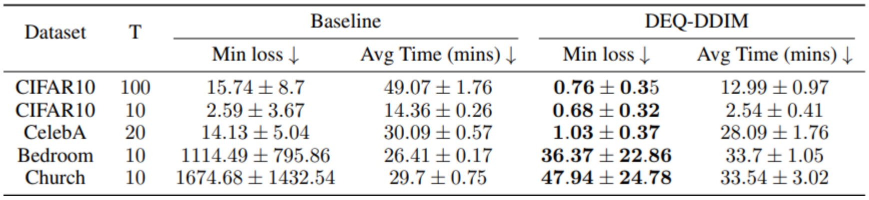
Finding $x_T^\star$ by gradient descent The previous DEQs formulation allows to define an efficient gradient descent method on the loss function $\mathcal{L}$. As described in [2], one can use the implicit function theorem on the computed fixed point $x_{0:T}^\star$ to derive the gradients of the loss w.r.t. any latent state $x_1, ..., x_T$:

$$\forall k \in [1, ..., T],\ \frac{\partial \mathcal{L}}{\partial x_k} = -\frac{\partial \mathcal{L}}{\partial x_{0:T}^\star} \tilde{J} \frac{\partial \tilde{h}(x_{0:T}^\star; x_T)}{\partial x_k} \tag{13}$$

Where $\tilde{J}$ is either a well-conditioned approximation of the inverse Jacobian of $\tilde{h}(x_{0:T-1}; x_T) - x_{0:T-1}$ at $x_{0:T}^\star$, or simply the identity matrix $\mathbf{I}$ (1-step gradient). In the end, we obtain an optimal $x_T^\star$.

Advantages of the DDIM-DEQ formulation, results By integrating DEQ with DDIM, we can calculate all equilibrium points concurrently (i.e., solving all $x_{T-(k-1)} = h(x_{T-k}), k \in [1, ..., T]$ simultaneously). This improves the estimation of intermediate latent states x_t and accelerates the convergence towards x_0. As a result, this approach eliminates the need for sequential sampling and introduces a parallel method that can utilize multiple GPUs to perform mini-batches computations, significantly increasing computational speed.

Comparing DDIMs to DEQ-DDIMs in terms of speed and precision on multiple datasets (such as CIFAR, CelebA, ...) with varying amounts of sampling steps shows the superiority of DEQ-DDIMs over DDIMs on the given task of model inversion, as shown in figure 1 :



Dataset	T	Baseline		DEQ-DDIM	
		Min loss ↓	Avg Time (mins) ↓	Min loss ↓	Avg Time (mins) ↓
CIFAR10	100	15.74 ± 8.7	49.07 ± 1.76	**0.76 ± 0.35**	12.99 ± 0.97
CIFAR10	10	2.59 ± 3.67	14.36 ± 0.26	**0.68 ± 0.32**	2.54 ± 0.41
CelebA	20	14.13 ± 5.04	30.09 ± 0.57	**1.03 ± 0.37**	28.09 ± 1.76
Bedroom	10	1114.49 ± 795.86	26.41 ± 0.17	**36.37 ± 22.86**	33.7 ± 1.05
Church	10	1674.68 ± 1432.54	29.7 ± 0.75	**47.94 ± 24.78**	33.54 ± 3.02

Figure 1: Results obtained by DDIM (Baseline) and DDIM-DEQ. Source: [2]

2 Theoretical study

2.1 Derivation and analysis of the forward Process

In this section we compute the forward process to establish the non markovian property of this process. From the conditional $q(x_t|x_0)$ and the marginals $q_\sigma(x_{t-1}|x_t, x_0)$ defined eq.5 that verify the conditional $q_\sigma(x_t|x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t\mathbf{I})\right)$ (eq.2), we want to deduce $q_\sigma(x_t|x_{t-1}, x_0)$.

For that we will use the property (2.115) of [4]. Denoting $\Gamma^{-1} = (1-\alpha_t)I,\ \mu_t = \sqrt{\alpha_t}x_0,\ A = \sqrt{\frac{1-\alpha_t-\sigma_t^2}{1-\alpha_t}}I,\ b = \mu_{t-1} - \frac{\mu_t}{\sqrt{1-\alpha_t}},\ L^{-1} = \sigma_t^2 I,\ \Sigma = \sigma_t^2\frac{1-\alpha_t}{1-\alpha_{t-1}}I$ yields:

$$q_\sigma(x_t|x_{t-1}, x_0) = \mathcal{N}\left(\Sigma[A^T L(x_{t-1} - b) + \Gamma\mu_t], \Sigma\right).$$

After some computations, we show that

$$\begin{aligned}\Sigma[A^T L(x_{t-1} - b) + \Gamma\mu_t] &= \frac{\sqrt{1-\alpha_t-\sigma_t^2}}{1-\alpha_{t-1}}\left[\sqrt{1-\alpha_t}(x_{t-1} - \mu_{t-1}) + \left(1 + \frac{\sigma_t^2}{\sqrt{1-\alpha_t-\sigma_t^2}}\right)\mu_t\right] \\ &= \frac{\sqrt{1-\alpha_t-\sigma_t^2}}{1-\alpha_{t-1}}\left[\sqrt{1-\alpha_t}x_{t-1} + \sqrt{\alpha_t}\left(1 + \frac{\sigma_t^2}{\sqrt{1-\alpha_t-\sigma_t^2}} - \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}}\right)x_0\right]\end{aligned}$$

So we can conclude that

$$x_t = \frac{\sqrt{1-\alpha_t-\sigma_t^2}}{1-\alpha_{t-1}}\left[\sqrt{1-\alpha_t}x_{t-1} + \sqrt{\alpha_t}\left(1 + \frac{\sigma_t^2}{\sqrt{1-\alpha_t-\sigma_t^2}} - \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}}\right)x_0\right] + \sqrt{\sigma_t^2\frac{1-\alpha_t}{1-\alpha_{t-1}}}\epsilon \quad (14)$$

with ϵ following a standard gaussian distribution. When $\sigma_t = 0$, (14) becomes:

$$x_t = \frac{\sqrt{1-\alpha_t}}{1-\alpha_{t-1}}\left[\sqrt{1-\alpha_t}x_{t-1} + \left(\sqrt{\alpha_t} - \sqrt{\alpha_{t-1}}\right)x_0\right]$$

It is clear that whenever $\sqrt{\alpha_t}\left(1 + \frac{\sigma_t^2}{\sqrt{1-\alpha_t-\sigma_t^2}} - \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}}\right) \neq 0$, the law of x_t depends on x_0 and so (x_t) is non markovian.

2.2 Deterministic sampling vs Stochastic sampling

Knowledge distillation The deterministic DDIM sampling procedure has beeen used in several applications. Indeed, a deterministic procedure enables to use the DDIM as a Teacher model for knowledge distillation in a student model. This knowledge distillation procedures aims at building a function whose output distribution would approximate the output distribution of the teacher model. Using a stochastic sampler in such a setting would make this task a lot more difficult. This idea has been leveraged in [6] that aims at accelerating the generative process, DDIM still requiring between 20 and 100 steps to generate a sample.

Their idea is to use knowledge distillation with a DDIM teacher to enable generation in a single step while preserving sample quality by compressing the knowledge of the teacher DDIM. Specifically, they aim at matching the distribution $p_{student}(x_0|x_T)$ with $p_{teacher}(x_0|x_T)$, which is enforced by a Kullback-Leibler divergence loss. The two models being gaussian, this results in maximizing the closed form:

$$\mathcal{L}_{student} = \frac{1}{2}\mathbb{E}[\|\mu_{student}(x_T) - \mu_{teacher}(x_T)\|_2^2] + C,$$

where μ denotes the mean of the Gaussian. This ultimately lead to a 1000× speed-up over the original DDPM, 100× speedup over the 100-step DDIM teacher, and is slightly faster than state-of-the art VAE NVAE [11]. This yields relatively good sample quality while still lower than the DDIM teacher used. The student model also incorporates its teacher interpolation properties.

This idea has further been expanded in [9] that is motivated by the fact that the training process described in [6] requires sampling the entire teacher model process to generate training samples, which can be particularly long. To alleviate this, they propose a progressive distillation scheme, still using a DDIM teacher. At a given iteration, a $N-$steps DDIM student will be trained to approximate a $2N-step$ DDIM teacher while maintaining sample quality. By iterating this process K times, they ultimately decrease the number of steps by a factor 2^K, leading to a generative process that maintains sample quality while generating samples in as few as 4 steps. Instead of the original data x_0, the target $\tilde{x}$ of the student model enhances a step of the student DDIM to match two steps of the teacher DDIM. This is achieved for:

$$\tilde{x} = \frac{z_{t-1/N} - (\sigma_{t-1/N}/\sigma_t)z_t}{\alpha_{t-1/N} - (\sigma_{t-1/N}/\sigma_t)\alpha_t}$$

where $z_{t-1/N}$ is sampled by the teacher model from z_t in two steps $t - 0.5/N$ then $t - 1/N$. Performing this indeed does not require to sample the whole backward chain of the teacher model during training, performing a constant number of sampling steps that does not depend on the number of steps of used by the teacher. After convergence, the student becomes the teacher of a $N/2-$step DDIM.

Latent representation and interpolation Unlike DDPMs, DDIMs can represent images in a latent space by the forward process. This deterministic sampling procedure also enables interpolation in the latent space between latent codes that results in meaningful interpolation in the image space, which again is impossible with stochastic sampling.

Sample diversity The main advantage of stochastic sampling however lies in the fact that it increases sample diversity, which is a desirable property of a generative model. Particularly, the fact that DDPM do not suffer from classical mode collapse issues probably favored their adoption over GANs, that are known to suffer from such mode collapse, which highly reduces the sample diversity. This has for instance been recently studied in [3].

2.3 Gamma-DDIM

In this subsection, we attempt to derive Gamma-DDIMs using [7] as a starting point. This is, to the best of our knowledge, the first attempt to derive Gamma-DDIMs. What follows is therefore an original derivation.

Explications In the original DDIM paper, the backward transitions $q_\sigma(x_{t-1}|x_t, x_0)$ (eq.5) are derived so that the conditionals $q_\sigma(x_t|x_0)$ verify eq.2, as in DDPM. Using properties of the multivariate Gaussian distributions, the update rule for the backward process is derived in eq.7. Here, we do a similar reasoning as we look for an update rule for the backward that is of the desired form (eq.24) whose conditionals $q_\sigma(x_t|x_0)$ verify eq.2. Using properties of the Gamma laws, we will then derive the backward marginals (eq.29).

Deriving backward We are aiming to find $(C_t)_{t=0,\dots,T}, (D_t)_{t=0,\dots,T}$ such that

$$x_{t-1} = C_{t-1}x_t + D_{t-1}x_0 + \sigma_t(g_t - \mathbb{E}(g_t)) \tag{15}$$

Where $g_t \sim \Gamma(k_t, \theta_t)$, $\theta_t = \sqrt{\bar{\alpha}_t}\theta_0$, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^T \alpha_i$, and $k_t = \frac{\beta_t}{\alpha_t\theta_0}$, so as to impose a backward process whose marginals can be written as [7]'s marginals for a Gamma-DDPM, i. e.

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + (\bar{g}_t - \mathbb{E}(\bar{g}_t)) \tag{16}$$

Where $\bar{k}_t = \sum_{i=1}^{t} k_i$ and $\bar{g}_t \sim \Gamma(\bar{k}_t, \theta_t)$. We present the detailed derivation in Annex 4.1 in the appendix. Using a constraint over k_t that is $k_{t+1} = \sum_{i=1}^{t} k_i$, we find, for C_t and D_t:

$$C_t = \frac{\sqrt{\bar{\alpha}_t} - \sigma_{t+1}\sqrt{\bar{\alpha}_{t+1}}}{\sqrt{\bar{\alpha}_{t+1}}(1+\sqrt{\bar{\alpha}_t})} \quad \text{and} \quad D_t = \frac{\bar{\alpha}_t + \sigma_{t+1}\sqrt{\bar{\alpha}_{t+1}}}{1+\sqrt{\bar{\alpha}_t}} \tag{17}$$

Hence the backward process for Gamma-DDIM:

$$x_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}} - \sigma_t\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}(1+\sqrt{\bar{\alpha}_{t-1}})} x_t + \frac{\bar{\alpha}_{t-1} + \sigma_t\sqrt{\bar{\alpha}_t}}{1+\sqrt{\bar{\alpha}_{t-1}}} \hat{x}_{0,t} + \sigma_t(g_t - \mathbb{E}(g_t)) \tag{18}$$

Where $\hat{x}_{0,t} = \frac{x_t - (\bar{g}_t - \bar{k}_t\theta_t)}{\sqrt{\bar{\alpha}_t}}$ is an estimation of x_0 at time t that has to be learned.

Backward marginals We then obtain the backward marginals using Gamma law properties:

$$q_\sigma(x_{t-1}|x_t, x_0) = \frac{(x_{t-1} - C_{t-1}x_0 - D_{t-1}x_t + \sigma_t\bar{k}_t\theta_t)^{\bar{k}_t - 1} \; e^{\left(-(x_{t-1} - C_{t-1}x_0 - D_{t-1}x_t + \sigma_t\bar{k}_t\theta_t)/(\sigma_t\theta_t)\right)}}{\Gamma(\bar{k}_t)(\sigma_t\theta_t)^{\bar{k}_t}} \tag{19}$$

From this equation, it is possible to see that the conditional of $f_{t-1,x_0,x_t}(x_{t-1})$ w.r.t. x_0, x_t follows a Gamma law $\Gamma(\bar{k}_t, \sigma_t\theta_t)$ where $f_{t,y,z}(x) = x - C_t y - D_t z + \sigma_{t+1}\bar{k}_{t+1}\theta_{t+1}$. Putting all together leads to a Gamma-DDIM generative process $p_\theta(x_{0:T})$ defined with eq.8, apart from $p_\theta^{(1)}(x_0|x_1)$ that verifies:

$$p_\theta^{(1)}(x_0|x_1) = \frac{(x_0 - \frac{1}{\sqrt{\bar{\alpha}_1}}x_1 + \frac{\bar{k}_1\theta_1}{\sqrt{\bar{\alpha}_1}})^{\bar{k}_1 - 1} \; e^{\left(-(x_0 - \frac{1}{\sqrt{\bar{\alpha}_1}}x_1 + \frac{\bar{k}_1\theta_1}{\sqrt{\bar{\alpha}_1}}))/(\frac{\theta_1}{\sqrt{\bar{\alpha}_1}})\right)}}{\Gamma(\bar{k}_1)(\frac{\theta_1}{\sqrt{\bar{\alpha}_1}})^{\bar{k}_1}}$$

Attemps to derive the objective The objective function to be minimized in θ can be written exactly like in the Gaussian DDIM' case *i.e.*:

$$\mathbb{E}_{\boldsymbol{x}_{0:T} \sim q(\boldsymbol{x}_{0:T})}\left[\sum_{t=2}^{T} D_{\mathrm{KL}}\left(q_\sigma(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,\|p_\theta^{(t)}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)\right) - \log p_\theta^{(1)}\left(\boldsymbol{x}_0 \mid \boldsymbol{x}_1\right)\right] \tag{20}$$

When dealing with Gaussian marginals $q_\sigma(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0$ and $p_\theta^{(t)}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$, the KL divergences have a closed formula and the objective can therefore be straightforwardly reformulated. However, in our scenario, as previously described, the marginals are not Gamma distributed. It's a transformation of these marginals that follow a Gamma distribution. The transformation applied is not identical for p_θ and q_σ, which complicates the process. As a result, we cannot simply apply a variable substitution in the integral defining the Kullback-Leibler (KL) divergence to compute it by bringing ourselves back to the known Gamma case.

Limitations A limitation of our derivation is that it requires a constraint on the shape parameter k of the Gamma noise employed, which would quickly explode with our formulation. A derivation that do not exploit such a constraint would be an interesting development of this project. The lack of closed formula also limits the application of our work. A direct compute of the integral defined by the KL divergence with the marginals obtained might enable such a closed formula. It would then be interesting to see if it is equivalent to Gamma-DDPM case.

3 Experiments

3.1 Experimental setting

Implementation details We conducted all our experiments using the CIFAR10 training dataset, which comprises 50,000 images, each of size 64x64 pixels, spanning across 10 distinct classes.

Our computations were executed on the "Ruche" cluster provided by Université Paris-Saclay. Utilizing the capabilities of an NVIDIA Tesla A100 GPU, we were able to generate 50,000 samples from the CIFAR10 dataset in a time-efficient manner, with the whole process taking roughly 30 minutes.

To evaluate the quality of our data distribution approximation, we employed the Fréchet Inception Distance (FID). This metric is commonly adopted in image generation tasks for its effectiveness in measuring the resemblance between generated images and original data.

Methodology In our experiments, we tried to answer the following question: which properties should a "good" subsampling approach verify?

We tested three different subsampling approaches, that we compared against the quadratic subsampling done in the DDIM paper:

- First, we explored an 'exponential' subsampling method, which extends the idea underlying quadratic subsampling—increasing sampling density as we approach the data distribution. In this approach, we used powers of 2 as the steps within the 0 to 1000 range (including 0 and 1000), resulting in a total of 12 steps.
- Additionally, we implemented a more 'aggressive' adaptation of this principle. We experimented with sampling just the first 100 and last 100 steps of the backward process, and in another setup, solely the last 100 steps. The purpose of these experiments was to examine the effects of bypassing a substantial number of samples at once.

3.2 Using alternative subsampling methods

As mentioned previously, DDIM enables significant subsampling during the backward process. Since the backward process transforms Gaussian samples into a more intricate distribution, it is logical to require more samples as it approaches the data distribution. Consequently, the linear subsampling method described in the DDIM paper may not be the optimal choice in terms of speed or accuracy for a given number of steps. Subsampling, in this context, is intimately related to the discretization step of the underlying Ornstein-Uhlenbeck (OU) process that simulates temporal evolution of noise. The step size crucially influences the level of detail and accuracy captured by the model, thereby determining the necessary number of steps. Using smaller step sizes in the backward process closer to the distribution of the original image x_0 while using bigger steps at the beginning of the backward process is a commonly used strategy.

In the same way, an adaptive subsampling strategy could potentially provide benefits by bypassing inconsequential variations in noise while focusing more on substantial changes. In this subsection, we examine alternative subsampling approaches that utilize a total of T timesteps (typically, $T = 1000$) and S subsampled timesteps (commonly, $S = 100$).

Quadratic Subsampling This subsampling technique, which is in-built in the DDIM framework, involves the following sequence of steps:

$$x_i \quad \text{where} \quad x_i = \left(\frac{\sqrt{S \cdot 0.8}}{T-1} \cdot i \right)^2 \quad \text{for} \quad i = 0, 1, 2, \ldots, T-1 \tag{21}$$

Each term in the sequence is rounded to the nearest integer. Owing to the convexity of the function $x \to x^2$, this results in a higher number of steps closer to the data distribution and fewer during the "noisy" phase.

Last 100 Steps Subsampling This sampling strategy, $\mathbf{x}_{last100}$, involves drawing a sample from the Gaussian noise (at step $T-1$) and directly progressing to the final 100 samples:

$$\mathbf{x} := [0, 1, ..., 98, 99, 1000] \tag{22}$$

This technique effectively bypasses the majority of the noisy segment of the process.

Exponential Subsampling Analogous to the quadratic method, the exponential subsampling strategy uses the function $S \in \mathbb{N} \rightarrow 2^S$, while retaining the first and last steps:

$$\mathbf{x}_{exp} := [0, 1, 2, 4, 8, ..., 512, 1000] \tag{23}$$

Results The results of this investigation are presented in Table 1.

Method	FID Score on CIFAR10	Steps
Last 100	33.8	102
Exp	49.4	12
First100Last100	161.9	200
Quadratic DDIM (10 steps)	13.36	10
Quadratic DDIM (100 steps)	4.16	100

Table 1: Comparison of FID scores for various methods on CIFAR10

It appears that quadratic subsampling provides an optimal balance between maintaining regularity in subsampling and reducing the number of steps at the beginning of the generative process, when the distribution is close to a Gaussian noise. The method whose number of steps is concentrated both at the beginning and the end of the generative process (First100Last100) yields the worst result, which is expected by the analogy with the step size previously mentioned. Sampling many steps at the beginning of the generative process is indeed not really helpful.

We also notice that methods whose subsampling is the least regularly spaced provide the worst results. These methods typically sample too many steps at beginning and the end of the backward process, but have a weak approximation of the transition between the extremities, that is too important to be correctly approximated in a single step.

From our experiments, it therefore seems that an ideal subsampling strategy would result in a tradeoff between regularly-spaced subsampling and concentration of the steps at the end of the generative process, close to the image distribution x_0. The strategy experimented that gathers these characteristics is the quadratic one, and it indeed yields the best results.

References

[1] Donald G. Anderson. Iterative procedures for nonlinear integral equations. *J. ACM*, 12(4):547–560, oct 1965.

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Reza Bayat. A study on sample diversity in generative models: GANs vs. diffusion models. In *International Conference on Learning Representations TinyPapers*, 2023.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[6] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

[7] Eliya Nachmani, Robin San Roman, and Lior Wolf. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021.

[8] Ashwini Pokle, Zhengyang Geng, and J Zico Kolter. Deep equilibrium approaches to diffusion models. *Advances in Neural Information Processing Systems*, 35:37975–37990, 2022.

[9] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[10] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[11] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020.

4 Appendix

4.1 Gamma-DDIM backward derivation

We are aiming to find $(C_t)_{t=0,...,T}$, $(D_t)_{t=0,...,T}$ such that

$$x_{t-1} = C_{t-1}x_t + D_{t-1}x_0 + \sigma_t(g_t - \mathbb{E}(g_t)) \tag{24}$$

Where $g_t \sim \Gamma(k_t, \theta_t)$, $\theta_t = \sqrt{\bar{\alpha}_t}\theta_0$, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$, and $k_t = \frac{\beta_t}{\alpha_t \theta_0}$, so as to impose a backward process whose marginals can be written as [7]'s marginals for a Gamma-DDPM, i. e.

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + (\bar{g}_t - \mathbb{E}(\bar{g}_t)) \tag{25}$$

Where $\bar{k}_t = \sum_{i=1}^{t} k_i$ and $\bar{g}_t \sim \Gamma(\bar{k}_t, \theta_t)$. Replacing x_t in 24 by its expression in 25 yields:

$$x_{t-1} = (C_{t-1}\sqrt{\bar{\alpha}_t} + D_{t-1})x_0 + C_{t-1}(\bar{g}_t - \mathbb{E}(\bar{g}_t)) + \sigma_t(g_t - \mathbb{E}(g_t))$$

Which gives a system to solve for C_t and D_t $\forall t \in [0, ..., T]$:

$$\begin{cases} \sqrt{\bar{\alpha}_{t+1}}C_t + D_t = \sqrt{\bar{\alpha}_t} \\ (\bar{g}_t - \mathbb{E}(\bar{g}_t)) = C_t(\bar{g}_{t+1} - \mathbb{E}(\bar{g}_{t+1})) + \sigma_{t+1}(g_{t+1} - \mathbb{E}(g_{t+1})) \end{cases}$$

But, $\bar{g}_{t+1} - \mathbb{E}(\bar{g}_{t+1}) = \sqrt{\bar{\alpha}_{t+1}}(\bar{g}_t - \mathbb{E}(\bar{g}_t)) + g_{t+1} - \mathbb{E}(g_{t+1})$. By taking (k_t) such that $\sum_{i=1}^{t} k_i = k_{t+1}$ (i. e. $k_t = 2^{max(0,t-2)}k_1$):

$$\bar{g}_t - \mathbb{E}(\bar{g}_t) = C_t(\sqrt{\bar{\alpha}_{t+1}}(\bar{g}_t - \mathbb{E}(\bar{g}_t)) + g_{t+1} - \mathbb{E}(g_{t+1})) + \sigma_{t+1}(g_{t+1} - \mathbb{E}(g_{t+1}))$$

$\Leftrightarrow$ $(a.s.)$

$$\sqrt{\bar{\alpha}_t}\theta_0 Z_{t+1} = C_t(\sqrt{\bar{\alpha}_{t+1}}\sqrt{\bar{\alpha}_t}\theta_0 Z_{t+1} + \sqrt{\bar{\alpha}_{t+1}}\theta_0 Z_{t+1}) + \sigma_{t+1}\sqrt{\bar{\alpha}_{t+1}}\theta_0 Z_{t+1}$$

Where $Z_{t+1} + k_{t+1}$ follows a $\Gamma(k_{t+1}, 1)$. Therefore, we have for C_t:

$$C_t = \frac{\sqrt{\bar{\alpha}_t} - \sigma_{t+1}\sqrt{\bar{\alpha}_{t+1}}}{\sqrt{\bar{\alpha}_{t+1}}(1 + \sqrt{\bar{\alpha}_t})} \tag{26}$$

And for D_t:

$$D_t = \frac{\bar{\alpha}_t + \sigma_{t+1}\sqrt{\bar{\alpha}_{t+1}}}{1 + \sqrt{\bar{\alpha}_t}} \tag{27}$$

Hence the backward process for Gamma-DDIM:

$$x_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}} - \sigma_t\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}(1 + \sqrt{\bar{\alpha}_{t-1}})} x_t + \frac{\bar{\alpha}_{t-1} + \sigma_t\sqrt{\bar{\alpha}_t}}{1 + \sqrt{\bar{\alpha}_{t-1}}} \hat{x}_{0,t} + \sigma_t(g_t - \mathbb{E}(g_t)) \tag{28}$$

Where $\hat{x}_{0,t} = \frac{x_t - (\bar{g}_t - \bar{k}_t\theta_t)}{\sqrt{\bar{\alpha}_t}}$ is an estimation of x_0 at time t that has to be learned.

Backward marginals We can therefore obtain the backward marginals:

$$q_\sigma(x_{t-1}|x_t, x_0) = \frac{(x_{t-1} - C_{t-1}x_0 - D_{t-1}x_t + \sigma_t\bar{k}_t\theta_t)^{\bar{k}_t - 1}\ e^{\left(-(x_{t-1} - C_{t-1}x_0 - D_{t-1}x_t + \sigma_t\bar{k}_t\theta_t)/(\sigma_t\theta_t)\right)}}{\Gamma(\bar{k}_t)(\sigma_t\theta_t)^{\bar{k}_t}} \tag{29}$$

From this equation, it is possible to see that the conditional of $f_{t-1,x_0,x_t}(x_{t-1})$ w.r.t. x_0, x_t follows a Gamma law $\Gamma(\bar{k}_t, \sigma_t\theta_t)$ where $f_{t,y,z}(x) = x - C_t y - D_t z + \sigma_{t+1}\bar{k}_{t+1}\theta_{t+1}$. Putting all together leads to a Gamma-DDIM generative process $p_\theta(x_{0:T})$ defined with eq.8, apart from $p_\theta^{(1)}(x_0|x_1)$ that verifies:

$$p_\theta^{(1)}(x_0|x_1) = \frac{(x_0 - \frac{1}{\sqrt{\bar{\alpha}_1}}x_1 + \frac{\bar{k}_1\theta_1}{\sqrt{\bar{\alpha}_1}})^{\bar{k}_1 - 1}\ e^{\left(-(x_0 - \frac{1}{\sqrt{\bar{\alpha}_1}}x_1 + \frac{\bar{k}_1\theta_1}{\sqrt{\bar{\alpha}_1}}))/(\frac{\theta_1}{\sqrt{\bar{\alpha}_1}})\right)}}{\Gamma(\bar{k}_1)(\frac{\theta_1}{\sqrt{\bar{\alpha}_1}})^{\bar{k}_1}}$$