The Family of Denoising Diffusion Probabilistic Models

Jack Naish [*1] **RuiKang OuYang** [*1] **Akshay Choudhry** [*1]

Abstract

Denoising Diffusion Probabilistic Models (DDPM) are a class of generative models, inspired by non-equilibrium thermodynamics, that feature stable training regimes and are capable of producing high quality, diverse data. However, DDPMs suffer from expensive inference times compared with GANs and VAEs. Furthermore, they can't achieve competitive log-likelihoods. To solve these issues and further improve DDPMs' performance, *Denoising Diffusion Implicit Model* (DDIM) and *Improved DDPM* (IDDPM) have been proposed. In this paper, we systematically summarise, replicate, and compare this family of models. We show that DDPMs can generate high quality images; DDIM speed up images generation but lower sample quality; while IDDPM can speed up sampling with negligible quality reduction. Additionally, we further explore an application of DDPMs on image in-painting via *RePaint*, illustrating power of this family. We release our code here.

1. Introduction

Deep Generative Models are capable of producing multi-modal, high quality data, such as images (Kingma et al., 2014), videos (Unterthiner et al., 2018; Brooks et al., 2024) and audio (Pascual et al., 2017). The generative paradigm underpinning learning is reminiscent of Richard Feynman's view of *What I cannot create, I don't Understand* (ope, 2024). Ilya Sutskever would a agree, critiquing *compression* rather than pure *creation*, of data would lead to an unsupervised learning breakthrough (Nvidia, 2023). The intuition behind these intellectual titans is reflected throughout the GM landscape. The notion of compression manifests in neural networks with far fewer parameters than the amount of data they are trained on, forcing efficient internalization of a dataset's natural features (ope, 2024). Similarly, accurately creating new data demands approximating underlying latent concepts that describe it.

In previous research, Generative Adversarial Networks (GANs, Goodfellow et al. 2014) have achieved impressive image generation results, outperforming log-likelihood-based models such as Variational Anto-Encoders (VAEs, Kingma & Welling 2013). However, GANs can suffer from *mode collapse* (Zhang et al., 2018), where the generator produces limited sample diversity, and *training instability* (Becker et al., 2022), making it difficult to achieve consistent results (Mescheder et al., 2018) (Gulrajani et al., 2017).

Variational Auto-encoders (VAEs) are another popular method, and instead optimise the likelihood lower bound to learn a latent space representation (Kingma & Welling, 2013). Although capable of stable training and easy sampling/interpolation, VAEs tend to produce *blurrier less-detailed* results compared to GANs (Wan et al., 2017).

Diffusion Models are a class of generative model first introduced by Sohl-Dickstein et al. (2015) inspired by non-equilibrium thermodynamics and characterised by supervised noising-denoising Markov processes. Ho et al. (2020) iterated on the idea with Denoising Diffusion Probabilistic Models (DDPMs), and proved the technique capable of generating samples of comparable quality to that of GANs, without adversarial training and the associated convergence instabilities.

Despite impressive performance, DDPMs suffers from two primary issues: 1) expensive data generation; 2) non-competitive log-likelihoods. Since DDPMs employ an iterative denoising process, inference is far more expensive than GANs which only require one forward pass through a neural network. To close the efficiency gap, *Denoising Diffusion Implicit Models* (DDIMs) (Song et al., 2020) and *Improved DDPMs* (IDDPM) (Nichol & Dhariwal, 2021) have been proposed, which both capitalize on step-skips to accelerate denoising processes. Furthermore, to achieve competitive log-likelihoods, IDDPM addtionally learns the reverse variance of a hybrid learning objective.

Our selection of these papers is due to their foundational relevance amongst a series of world-changing industry products, such as Stable-Diffusion, DALLE-2 and Sora(Brooks et al., 2024). Diffusion Models also share strong ties to concepts in statistical physics, thus providing us a duality of perspective in the analysis of this fascinating class of model.

[*]Equal contribution [1]Department of Engineering, University of Cambridge, Cambridge, United Kingdom. Correspondence to: Jack Naish <jrhn2@cam.ac.uk>, RuiKang OuYang <ro352@cam.ac.uk>, Akshay Choudhry <ac2591@cam.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

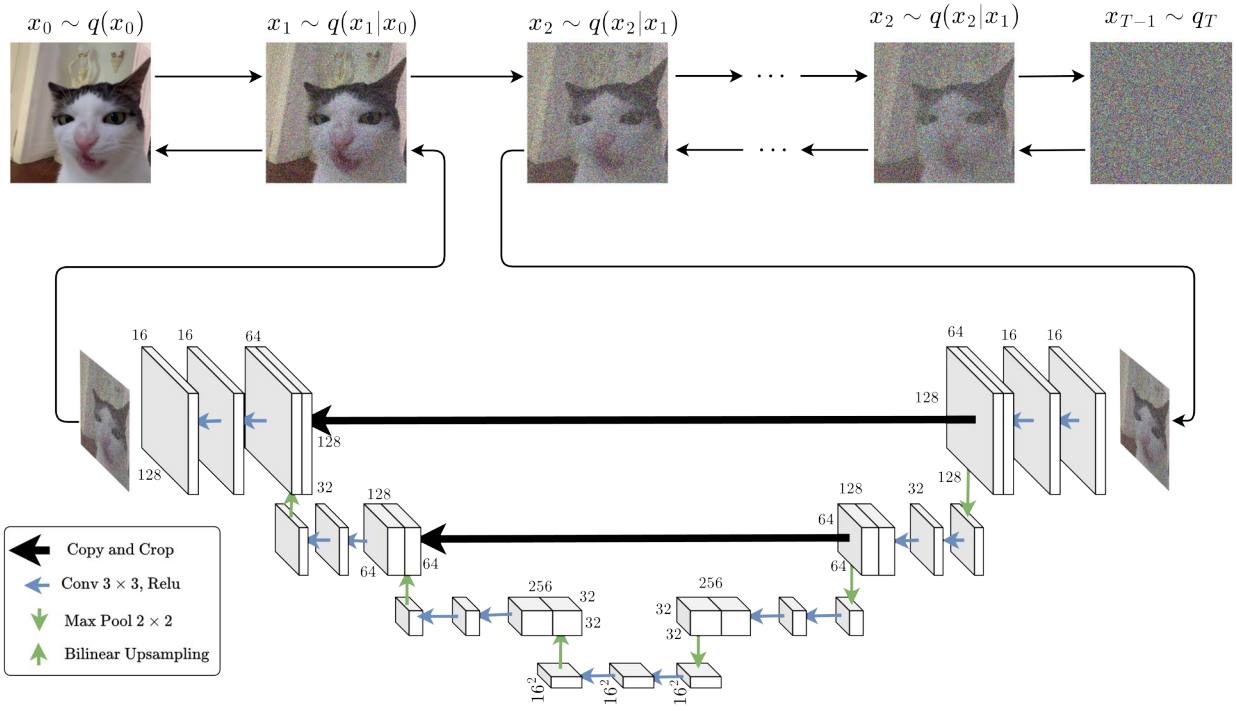

Figure 1. Noising and diffusion processes of the diffusion model over T steps.

Contributions: In Section 2, we introduce the family of denoising diffusion models, including DDPMs (Section 2.1), DDIMs (Section 2.2) and IDDPM (Section 2.3), covering their intuition, theory and limitations. An application to image inpainting (Section 2.4) is also explored.

In Section 3, the main experiments of DDPMs, DDIMs and IDDPM are reproduced. We first train the DDPM and IDDPM on CIFAR-10 (Krizhevsky et al.) and find that IDDPM achieves better training loss than DDPM in Section 3.2. The log marginal likelihood of these two models are compared in Section 3.3, showing that IDDPM achieves much better performance, which aligns our hypothesis. To compare data generation quality between DDPMs, DDIMs and IDDPM, we evaluate three quality metrics: FID, Inception Score and Distortion (Section 3.4). Finally, a noising schedule ablation study in Section 3.5 quantifies the effect on IDDPM training. We conclude with critical reflection of our work and future directions.

2. Background

Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020), achieve high quality data generation, stable training, and multi-modal dataset coverage. Drawbacks include expensive inference associated with data generation. In this section, we first introduce general DDPMs, before moving into two variants: Denoising Diffusion Implicit Models (DDIM)(Song et al., 2020) and Improved DDPMs (IDDPM)(Nichol & Dhariwal, 2021), which ameliorate the limitations DDPM in terms of data generation speeds and improved quality.

2.1. Denoising Diffusion Probabilistic Models

DDPMs operate via two distinct passes over a first order Markov chain (Figure 1). The *noising* (aka *diffusion*, *forward*) process start from a real data point (such as an image, protein, or waveform) and iteratively imposes noise over a fixed number of steps, T, until an uncorrelated sample is obtained (Biroli et al., 2024). In the *denoising* (aka *backward*, *reverse*) process, the uncorrelated data point is gradually reconstructed using a *diffusion* in an effective force field. This is learnt by leveraging variational inference to produce samples matching the data after finite time (Ho et al., 2020).

Noising Process: Given a training input x_0 from true data distribution $q(x)$, the *Noising Process* additively imposes isotropic noise over each diffusion step $\forall t \in \{1, ..., T\}$, producing a sequence of increasingly corrupted latents $x_{0:T}$. From the first order Markov assumption, the distribution of each latent $q(x_t|x_{t-1})$ is a Gaussian centered around its predecessor x_{t-1} with mean $\mu_t = \sqrt{1-\beta_t}x_{t-1}$ and variance $\Sigma_t = \beta_t \mathbf{I}$:

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}) \quad (1)$$

where $\beta_{0:T}$ denote the variance schedule applied across T noising steps (Croitoru et al., 2023a). By the end of the noising pass, the latent x_T is almost an uncorrelated isotropic Gaussian distribution . Unlike Variational Auto

Encoders, this noising process (aka, *encoder*) is not learned; instead, it's an autoregressive Gaussian model. Given T is often thousands of steps long, it is inefficient to recursively apply Equation (1) T times. Instead, the reparameterization trick is used to enable closed form constant time direct-sampling of arbitrary x_t:

$$p(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{2}$$

where $\bar{\alpha}_t = \Pi_{s=1}^{t}\alpha_s$ Equation (2) (Croitoru et al., 2023b). See Appendix A for derivation.

Denoising Process: The goal of the denoising process is to take an uncorrelated sample $x_T \sim q(x_T)$ and gradually reconstruct x_T back into an non-distorted sample x_0 that could plausibly have originated from our dataset. This process involves following the reverse steps $p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu(x_t, t), \Sigma(x_t, t))$. In practice, we train a neural network with weights θ to approximate these steps,

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{3}$$

This network takes as input the current noised latent x_t and a time embedding t (Croitoru et al., 2023b). The exact modality predicted by the neural network comes in two flavors: predicting the *additive noise* ϵ_t, or predicting the *mean* $\mu_\theta(x_t, t)$ and *covariance* $\Sigma_\theta(x_t, t)$, at each denoising step.

Crucially, since the diffusion timestep T is often large ($\geq$ 1000), using unique neural networks for each step would be prohibitively expensive. Thus, a parameter-sharing scheme is introduced allowing a single network to learn all latent transients over all steps. The time-embedding serves this purpose by allowing the network to observe where it is in the current denoising process.

Training: As discussed in (Croitoru et al., 2023a), ideally our network $p_\theta(x_t, t)$ would be trained via maximum likelihood learning to ensure the probability assigned by our model to a given training example is maximal. However, as this would require marginalisation over all possible reverse trajectories (of which there are infinitely many) such a process would be intractable. Instead, similar to VAEs, we minimise a variational lower bound on the negative log likelihood. This is defined by (see Appendix B for derivation):

$$L_0 = \log p_\theta(x_0|x_1) \tag{4}$$

$$L_{t-1} = \mathbb{D}_{\text{KL}}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t)), \quad t = 2, ..., T \tag{5}$$

Such that,

$$L_{\text{vlb}} = -L_0 + \sum_{t=2}^{T} L_{t-1} \tag{6}$$

where $q(x_{t-1}|x_t, x_0)$ is the "true" denoising process given the original input x_0 (see Appendix C for derivation):

$$q(x_{t-1}|x_t, x_0) = N(x_{t-1}|\tilde{\mu}(x_t, x_0), \tilde{\beta}_t I) \tag{7}$$

$$\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \tag{8}$$

where $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$.

In practice, $p_\theta(x_t, t)$ is often made a U-Net (Ronneberger et al., 2015) to take advantage of efficient multi-scale processing associated with this class of network. The U-Net architecture, visualised in Figure 1, incorporates copy-crop skip connections to facilitate the integration of contextual information across layers, ensuring the preservation of fine details while supporting high-level abstraction.

Ho et al. (2020) found that using a fixed isotropic denoising variance $\sigma_t^2 I$ would result in better model performance and either fixing this variance as $\beta_t I$ or $\tilde{\beta}_t I$ would be almost the same. In this case, the learning objective becomes a simple regression problem of the noise prediction (see Appendix C for derivation):

$$L_{t-1} = \mathbb{E}_q[\frac{1}{2\sigma_t^2}\|\mu_\theta(x_t, t) - \tilde{\mu}(x_t, x_0)\|^2] + C \tag{9}$$

The prediction for $\mu_\theta(x_t, t)$ can be further parameterized as a linear transformation of predicted noise $\epsilon_\theta(x_t, t)$. For detailed derivation, see Appendix E:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) \tag{10}$$

And thus Equation (9) is rewritten as:

$$L_{t-1} = \mathbb{E}_{x_0,\epsilon}[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(x_t, t)\|^2] + C \tag{11}$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim N(0, I) \tag{12}$$

Instead of directly optimizing the ELBO here (Equation (6)), Ho et al. (2020) found optimizing the following *simpler* mean-squared error objective, L_{simple}, is more stable and results to better performance:

$$L_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2] \tag{13}$$

In the rest of paper, we define other models from the DDPMs family under the prediction of noise $\epsilon_\theta(x_t, t)$ without specification.

Sampling: At inference time, we sample standard Gaussian noise $x_T \sim N(0, I)$ and reverse the noising process via a stochastic differential equation defined by Equation (3), recursively feeding forward the denoised result after each step.

$$x_{t-1} \sim N(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{14}$$

2.2. Denoising Diffusion Implicit Models

Generating samples with DDPMs is computationally intensive due to the necessity of iterating through thousands of denoising steps. DDIMs attempt to accelerate this process through a non-Markovian generalization, enabling "leap-frogging" large regions of the diffusion process while maintaining the same training objective and similar sample quality. The key enabling insight follows from this more general non-Markovian inference process having the same marginal distribution $q(x_t|x_0)$ at every t as in DDPM, but not necessarily the same joint distribution over all latents $q(x_{1:T}|x_0)$.

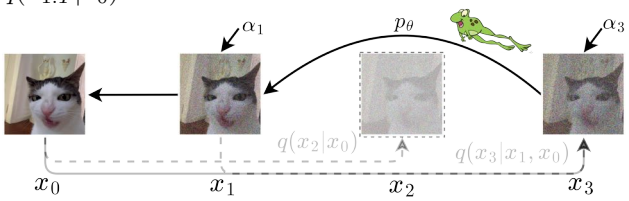

Figure 2. Graphical Model for DDIM's accelerated sampling, $\tau = [1, 3]$. Adapted (Szegedy et al., 2016)

Firstly, DDIM defines a general non-Markovian forward process by setting a conditional reverse process as:

$$q_\sigma(x_{t-1}; x_t, x_0) = N(x_{t-1}; \tilde{\mu}_\sigma(x_t, x_0), \sigma_t^2 \mathbf{I}) \tag{15}$$

$$\tilde{\mu}_\sigma(x_t, x_0) = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1 - \bar{\alpha}_t}} \tag{16}$$

Under this setting, the noising process satisfies: $p_\sigma(x_t; x_0) = N(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)\mathbf{I})$, meaning this non-Markovian setting is leading to a marginal distribution equivalent to DDPMs' diffusion process, which can verified by Bayes's rule (see Appendix 6).

Given x_0, we obtain ϵ_t, x_t from Equation (2), while the model $\epsilon_\theta(x_t, t)$ tries to recover the noise added to x_t. This allows rewriting Equation (2) and aligning the predicted noise to derive an estimation of x_0 given x_t and the noise predictor ϵ_θ:

$$x_0 \approx f_\theta(x_t, t) := \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\bar{\alpha}_t} \tag{17}$$

Plugging the denoised observation Equation (17) into Equation (15), yields the following data generation process:

$$x_{t-1} = \underbrace{\sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{Predicted } "x_0"} + \underbrace{\sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \epsilon_\theta(x_t, t)}_{\text{Direction pointing to } "x_t"} + \underbrace{\sigma_t \epsilon_t}_{\text{Noise}} \tag{18}$$

where $\bar{\alpha}_0 := 1$ and $\epsilon_t \sim N(0, \mathbf{I})$. Note that when $\sigma_t = \sqrt{(1 - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_t)} \sqrt{1 - \bar{\alpha}_t / \bar{\alpha}_{t-1}}$, the data generation process is equivalent to DDPMs. Furthermore, this new data generation process is stochastic. To control said stochasticity, a hyper parameter $\eta \in [0, 1]$ is added:

$$\sigma_t(\eta) = \eta \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}} \sqrt{1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}} \tag{19}$$

When $\eta = 0$, the data generation process become deterministic, which is simply an ODE.

This process eliminates the need to retrain the foundational DDPM since ϵ_θ remains unchanged. Moreover, the non-Markovian forward process in DDIM could be defined in a sub-sequence $[\tau_1, ..., \tau_S] \in [1, ..., T]$ of length S, which leads to a data generation process over this sub-sequence that reduces sampling time. Given the sub-sequence, we could theoretically speed up the data generation pass of $\frac{T}{S}$ times faster by:

$$x_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}} \left(\frac{x_{\tau_i} - \sqrt{1 - \bar{\alpha}_{\tau_i}} \epsilon_\theta(x_{\tau_i}, \tau_i)}{\sqrt{\bar{\alpha}_{\tau_i}}} \right) + \sqrt{1 - \bar{\alpha}_{\tau_{i-1}} - \sigma_{\tau_i}^2(\eta)} \epsilon_\theta(x_{\tau_i}, \tau_i) + \sigma_{\tau_i}(\eta) \epsilon_{\tau_i} \tag{20}$$

2.3. Improved DDPMs

Though DDPMs generate high quality data, they can't achieve competitive log-likelihoods. This is an important metric, both for comparison with other generative models, and given the consensus that optimizing log-likelihood forces generative models to capture all of the modes of the data distribution (Nichol & Dhariwal, 2021). The key insight of IDDPMs is, rather than using a fixed variance in the reverse process, (Nichol & Dhariwal, 2021) we instead *learn* this variance to produce a tighter variational lower bound estimate that leads to better log-likelihood. Additionally, the learnable variance schedule of the reverse process enables *accelerated* data sampling at inference time with only small differences in sample quality.

Learnable Reverse Variance: (Nichol & Dhariwal, 2021) learn the reverse process variance by predicting an *interpolation* between β_t and $\tilde{\beta}_t$:

$$\Sigma_\theta(x_t, t) = \exp(v \log \beta_t + (1 - v) \log \tilde{\beta}_t) \tag{21}$$

$$v = v_\theta(x_t, t) \tag{22}$$

enabling the network to produce $(\epsilon_\theta(x_t, t), v_\theta(x_t, t))$ rather than a single point estimate noise prediction. This interpolation idea is inspired by the narrow range of possible values for $\Sigma_\theta(x_t, t)$. To avoid the instability associated with directly predicting Σ_θ, we instead parameterize the model using β_t and $\tilde{\beta}_t$ as soft extremes.

Training: The L_{simple} refeq:l-simple could only guide learning of ϵ_θ. To guide the learnable variance v_θ, Nichol &

Dhariwal (2021) further use the ELBO 6 and propose the following hybrid-objective:

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}} \tag{23}$$

It is also valid to directly optimize the log-likelihood by optimizing the ELBO. However, Nichol & Dhariwal (2021) find in practice that L_{vlb} (Equation (6)) has extremely noisy gradients, thereby leading to unstable training regimes, but a simple importance sampling technique could reduce the variance and allows us to achieve better log-likelihood than the hybrid-objective (Equation (23)).

Improved Noise scheduling: The linear noise scheduler introduced by Ho et al. (2020) destroys information too quickly, thus the FID of resulting images does not noticeably increase until $\approx 30\%$ of steps in the reverse process. As such, Nichol & Dhariwal (2021) propose a cosine scheduler.

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos^2\left(\frac{\pi}{2} \cdot \frac{t/T + s}{1+s}\right) \tag{24}$$

Performance schedulers are compared and evaluated in Section 3.5.

Accelerated Sampling: To speed up data generation process, Nichol & Dhariwal (2021) process a similar "*leapfrogging*" idea as DDIM 2.2 but without the non-Markovian reverse process assumptions. In details, given a subsequence $[\tau_1, ..., \tau_S] \in [1, ..., T]$ of length S,

$$p_\theta(x_{\tau_{i-1}}|x_{\tau_i}) = N(x_{\tau_{i-1}}; \mu_\theta(x_{\tau_i}, \tau_i), \Sigma_\theta(x_{\tau_i}, \tau_i)) \tag{25}$$

where $\beta_{\tau_i} = 1 - \frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}}$ and $\tilde{\beta}_{\tau_i} = \frac{1-\bar{\alpha}_{\tau_{i-1}}}{1-\bar{\alpha}_{\tau_i}}\beta_{\tau_i}$ and thus $\Sigma_\theta(x_{\tau_i}, \tau_i)$ would be automatically rescaled for shorter diffusion process due to its interpolated nature. Denoising over a subsequence of length S, sampling time in IDDPM would be T/S times faster, similar to DDIMs. However, it is slightly slower than original DDIM since the neural network has more parameters for learning the variance.

2.4. Inpainting with RePaint

Inpainting is a popular task in Generative Modelling that involves predicting a masked, unseen portion of an image based on the surrounding content. This problem tests generative capabilities of tasked models because it requires extrapolation of often complex visual scenes. RePaint (Lugmayr et al., 2022) is one such in-painting technique that builds upon DDPMs, and which is capable of handling arbitrary masks without training on specific masked distributions. This is a significant capability since prior techniqes, such as *Large Mask Inpainting* (LaMa), require training on specific masked distributions and often perform poorly when encountering off-distribution masks (Suvorov et al., 2021).

RePaint works by receiving as input two regions of an image: the known area x^{known} containing the base image, and an unknown area x^{unknown} representing a masked region to be produced. At inference time, to produce each $x_{T:0}$ during the reverse process, the model predicts x_t^{unknown} whilst conditioned on the known region x_t^{known}. This process is realised by an element-wise multiplication between x_t^{known} and x_t^{unknown} regions, over with the binary mask, m, as depicted in Equation (26).

$$x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1-m) \odot x_{t-1}^{\text{unknown}} \tag{26}$$

$$x_{t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{27}$$

$$x_{t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{28}$$

The known noise region is obtained by passing the input image through an unmodified DDPM forward process using Equation (27), which corresponds to the top row of Figure 3. The unknown region is exclusively obtained via the denoising process using Equation (28) and Figure 3 (bottom row). Crucially, since the forward process is unmodified and RePaint exclusively modifies the denoising procedure, the foundation DDPM model does not need to be retrained, an off-the-shelf pretrained model can be used.

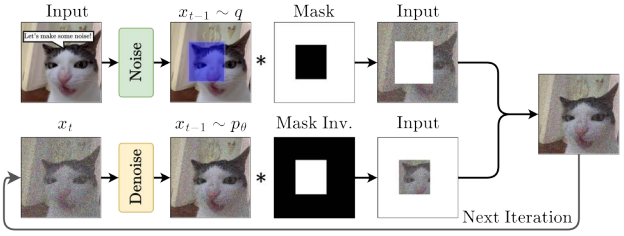

Figure 3. Overview of RePaint approach. Adapted from (Lugmayr et al., 2022)

It is this *resampling* approach that gives RePaint its name, and which serves to harmonize the image during the reverse process. This is done by feeding forward diffusion outputs x_{t-1} back into x_t by sampling Equation (27) as $x_t \sim \mathcal{N}(\sqrt{1-\beta_t}, \beta * \mathbf{I})$. Despite this operation scaling the output and adding noise, it allows contextual information extrapolated from the previous diffusion step to be combined into the next step, aiding in temporal coherence. This innovation is cited as crucial to ensuring the semantic meaning of RePaint predictions are consistent with the remainder of the image.

3. Experiments

In this section, we first introduce the experimental setups, including model architecture, diffusion settings and training settings. We then present the simulations we ran, including: 1) Training, sampling and evaluation for DDPM, DDIM and IDDPM; 2) Ablation study on noise scheduler; 3) Images inpainting using Diffusion models.

At the end of this section, we compare our results with those in the original paper, as well as present our conclusions.

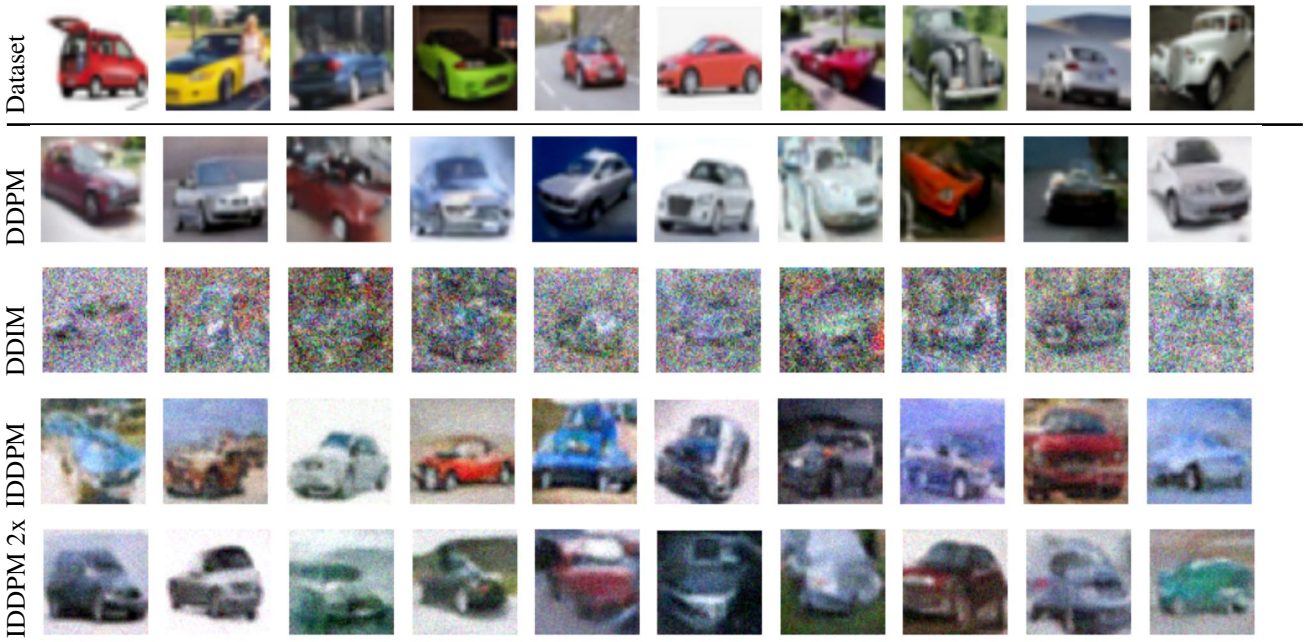

Figure 4. Image samples generated by our DDPM, DDIM, IDDPM models.

Table 1. Model Performance Comparison

Diffusion Model	Sampling Time(s)	FID	Inception Score	Log Likelihood	Train Time	Train MSE
DDPM	12s	**47.4**	**2.03**	-2.98	5hr 12min	0.007841
DDIM	7s	362.1	1.08	NA	Uses DDPM	Uses DDPM
IDDPM	14s	56.3	1.98	**-1.52**	3hr 38min	**0.002376**

3.1. Experimental Setup:

Model architecture and dataset. In our experiments, we use an UNet (Ronneberger et al., 2015) with 4 down-sampling layers, 4 up-sampling layers and self-attention blocks containing $23,332,739$ parameters. Due to compute limitations, we train our models on one class (car) in CIFAR-10 (Krizhevsky et al.).

Diffusion setting: For all diffusion models we trained, fixed-length diffusion time-step of $T = 1000$ is selected with initial variance $\beta_0 = 0.0001$ and final $\beta_T = 0.02$, sized commensurate with available computation resources. The original IDDPM paper (Nichol & Dhariwal, 2021) uses $T = 4000$ steps. Per the original paper, a linear scheduler for DDPM, and cosine scheduler 24 for IDDPM are used. We also conduct an ablation study among linear, cosine and tanh schedulers for IDDPM to quantify differences.

Training settings: We train all models with batch size 32, image size 64×64 for $40,000$ iterations (260 epochs, around 6 hrs). Learning rate is set to 0.0003 and $\lambda = 0.001$ Equation (23) for IDDPM training.

3.2. Training on CIFAR10

The DDPM and IDDPM models are trained on a singe class (car) of the CIFAR-10 dataset. Figure 4 shows the samples generated, with quantitative metrics in Table 1. DDPMs generate high quality (similar to the original dataset) images but suffer from an expensive denoising process. Conversely DDIMs, which use the same foundation model as DDPM, achieve much faster sampling time ($\approx$ 2x speedup in our experiments) yet with a sizable tradeoff in image quality. IDDPM ameliorate this detriment by generating and maintaining high-quality images by skipping sampling steps (see IDDPM 2x).

Figure 5 details the Log Mean Square Error (MSE) throughout a 6hr, 260 epoch training process. For both DDPM (blue) and IDDPM (red), image quality subjectively increases, initially producing low-detail, incoherent images before gradually learning more detailed features such as wheels, doors, and windows. Of note is the large MSE asymptotic difference between our models. A lower MSE for IDDPM suggests the model is more accurately predicting and removing corrupting noise at each timestep compared to DDPM. Although not obvious from the plot (due to log scale), IDDPM also demonstrated increased training stability with a smaller MSE variance.

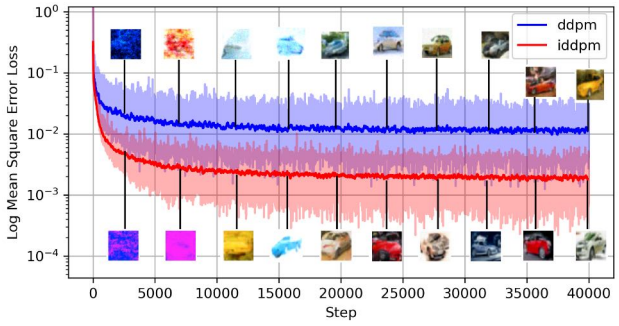

Figure 5. MSE Loss curves with sampled images during training for DDPM (blue) and IDDPM (red).

The original DDPM paper obtains a final MSE of 10^{-5} and thus us not shown. This large difference can be at-

tributed to our compute restraints. Given more HPC time, we posit MSE of a similar order of magnitude would be obtained.

3.3. Log Marginal Likelihoods

Prior to IDDPM in 2021, it was yet to be shown DDPMs could achieve log-likelihoods competitive with other likelihood-based models such as VAEs. This raised important questions regarding DDPM's ability to capture all modes of a distribution (Nichol & Dhariwal, 2021). Furthermore, small improvements in log-likelihood have been shown to yield dramatically better sample quality and learned feature representations (Henighan et al., 2020)

A key result of the IDDPM paper was that the proposed hybrid objective enabled the model to obtain larger log-likelihoods than those obtained by optimizing the log-likelihood directly. This experiment quantifies that difference in Figure 6.

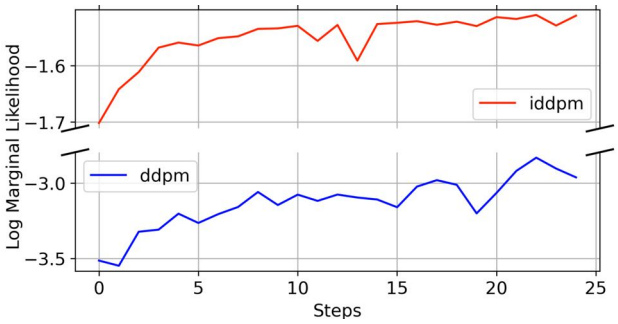

Figure 6. Comparison of Log Marginal Likelihoods, estimated by ELBO, for DDPM (blue) and IDDPM (red)

Our results suggest IDDPM learns a better *representation* of the underlying latent concepts encapsulated in our dataset compared to DDPM. Evidently, even the small increase in LML ($-1.52 - -2.98 = 1.46$) correlates with a drastic subjective improvement in sample quality (Figure 4). However, this comparison might be biased and unfair, since the ELBO is not tight for the log-likelihood. Instead, we should compute the exact log-likelihood for fair model comparison.

$$\begin{aligned}\log p_\theta(x_0) &= \log \int p_\theta(x_0, x_1, ..., x_T) dx_1 ... dx_T \\ &\approx \log \int p_\theta(x_0|x_1) ... p_\theta(x_{T-1}|x_T) p_\theta(x_T) dx_{1:T} \\ &\approx \log \frac{1}{M} \sum_{m=1}^{M} p_\theta(x_0^m|x_1^m) ... p_\theta(x_{T-1}^m|x_T^m) p_\theta(x_T^m)\end{aligned} \tag{29}$$

Equation (29) envisions this could be accomplished by approximating log-likelihood via a Monte Carlo method. Due to scope constraints we leave this for future work.

3.4. Quality Metrics: FID, Inception Score, Distortion

Our three models are evaluated using Fréchet Inception Distance, Inception Score, and Distortion.

Fréchet Inception Distance measures the similarity between two probability distributions over the means and covariances of activation layer outputs from an InceptionV3 network trained on Image Net (Heusel et al., 2017) (Szegedy et al., 2016). These two distributions are composed of our own *generated* images and CIFAR-10's *ground-truth* images. Lower FID scores indicate these distributions are "*closer*", implicating higher generated image quality.

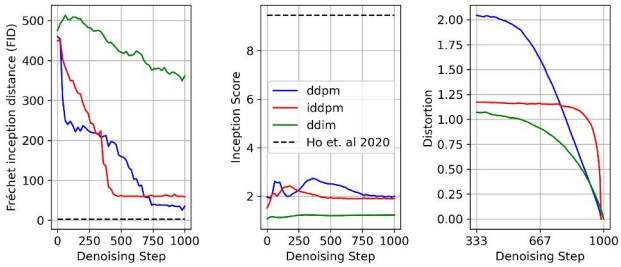

Figure 7. DDPM, DDIM, and IDDPM FID, inception, and distortion scores for images sampled every 20 timesteps.

Our FID results indicate DDPM (blue, FID$= 47.4$) performs slightly poorer than IDDPM (red, FID$= 56.3$) which aligns with visual intuition since its sampled images in Figure 4 are comparatively more noisy. Our FIDs are an order of magnitude worse than the 3.17 reported by (Ho et al., 2020) on the full CIFAR-10 dataset. We attribute this large difference to the original authors 1) training their model on significantly more data (entire CIFAR10 vs our 1 class) and evaluation over a much larger validation set ($10,000$ vs our 100 images). Our smaller numbers here are a consequence of HPC computation constraints.

Interestingly, the assumption of multivariate normal distributions in the activation layer of the InceptionV3 score can lead to biased FID scores Chong & Forsyth (2019). This means the expected value of the score computed for a finite sample set is not the score's true value. Given this bias decreases as sample size increase, (Ho et al., 2020) large sample size of $50,000$ images to compute their FID scores is a suitable choice.

Inception scores also utilize the InceptionV3 network, but compute the exponentiated expected KL divergence between the conditional distribution of InceptionV3 classification labels, given a generated sample $\mathbf{x}$ and the marginal distribution over all samples (Salimans et al., 2016). Thus, the inception score theoretically rewards samples of *high quality* and *diversity*.

Our inception score results also indicate similar performance between DDPM (2.03) and IDDPM (1.98). DDIM images attain a score of 1.08, suggesting that they are of significantly lower quality. Once again, our inception scores are worse than Ho et al. (2020), which obtain 9.46. As Bar-

ratt, et. al. Barratt & Sharma (2018) note, this is an apples to oranges comparison since we only trained on one class of images (cars), and thus would expect the KL divergence between the conditional and marginal class distributions to be significantly lower than when performed on 10 classes. Moreover, the InceptionV3 network is capable of predicting many different nuanced cars/trucks classes whereas CIFAR-10 only contains one *car* class. Evaluating probability distributions for these mismatched classes was an interesting choice made by the original authors.

Distortion measures how well a given image at each denoising timestep can approximate a fully reconstructed image $\mathbf{x}_0$, according to the following equation (Ho et al., 2020):

$$\bar{\mathbf{x}}_0 = \mathbf{x}_t - \frac{\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}} \tag{30}$$

Using the RMSE between the reconstruction and the approximation at timestep t (normalized by the dimensionality of the image), we can calculate an estimate of the information conveyed at each timestep. Our distortion results show a quadratic drop in information conveyed through denoising steps for DDPM, which contrasts the linear shape that Ho et al. (2020) find. We also find that DDPM and DDIM show less sharp convergences in reconstruction than IDDPM.

3.5. Custom Noise Schedules

Noising schedules describe how a dataset sample is systematically corrupted during the forward process. How much noise is added, and when, is strongly correlated with diffusion model performance. The original DDPM formulation assumed a *linear* Gaussian noising schedule over timesteps (Figure 8 red line). A flaw identified with this approach is that the original image is lost too quickly such that the last quarter of the sequence is pure noise. At train time, this last quarter presents minimal learning utility due to low correlation between adjacent x_t, and as a result wastes a lot of compute.

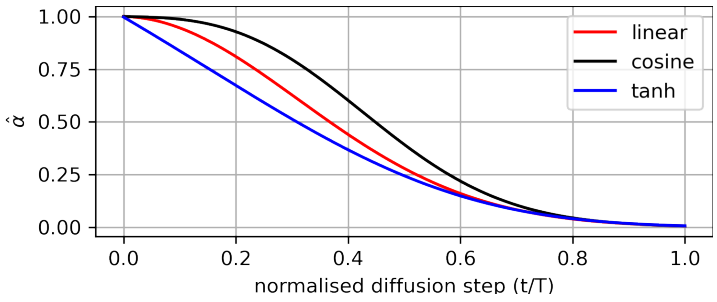

Figure 8. Linear, cosine, and tanh noising schedulers as a function of diffusion step

IDDPM addresses this through an alternative *cosine schedule* (Figure 8 black line) designed to have a linear drop-off in the middle of the process while changing very little near the extremes of $t = 0$ and $t = T$. We also propose our own schedule, theorizing the addition of too much noise at the end of the process (rather than the start) causes a deterioration in sample quality. To test this, we introduce a *hyperbolic tangent* scheduler that adds noise very fast at the beginning of the forward process before drastically decelerating noise imposition towards the end (Figure 8 blue line).

Figure 9 qualitatively compares samples from IDDPM for each of the three schedules (linear, cosine, tanh (ours)). As was highlighted in the original paper, the first quarter of the linear schedule is incoherent.

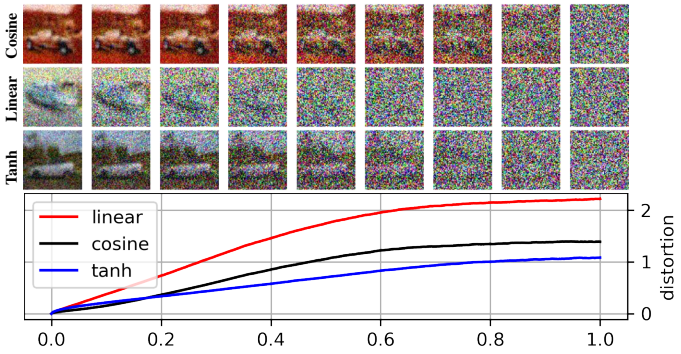

Figure 9. Denoising results for IDDPM trained on linear, cosine, and tanh (ours) schedules

As indicated by Table 2, the Tanh schedule performs better than the linear alternative in terms of FID, and surpasses the cosine scheduler in terms of inception score. Tanh distortion is also better than cosine for all but the final 10% of the denoising process, where it is surpassed marginally by cosine.

Table 2. Performance metrics of noising schedules

Schedule	FID	Inception
Linear	124.17	**3.08**
Cosine	**56.34**	1.98
Tanh (ours)	96.26	2.14

3.6. In-painting with RePaint

A successful in-painting example is shown in Figure 10 (top row), where the intersecting region between a car windshield and bonnet is masked.

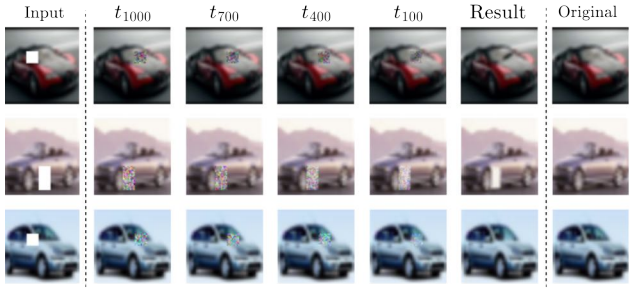

Figure 10. Image Inpainting using the technique proposed by RePaint(Lugmayr et al., 2022)

Our RePaint model can reconstruct this region by convincingly extrapolating the presence of a windshield wiper,

despite no such wiper being present in the original image. This kind of extrapolation ability alludes to diffusion model's ability to reason about latent concepts in the world.

As evident in Figure 10, the masked region can be of any arbitrary size. This is a desirable attribute since some other techniques such as LaMa (Suvorov et al., 2021) or DeepFillv2 (Yu et al., 2018) require expensive retraining on mask-specific datasets.

However, a common failure mode of RePaint was generating inpainted predictions that match high-level semantic image features, yet lack subject-specific details. For example, in Figure 10 (bottom row), the in-painted area is merely a solid grey rectangle which, despite matching the general color of surrounding car metal, lacks convincing details such as specular highlights. This occasional inability to harmonize a masked region with remainder of the image was identified as a shortcoming in the original RePaint paper (Lugmayr et al., 2022).

4. Discussion

In this paper, we systematically investigated the family of denoising diffusion based models, a popular class of deep generative models. We traced the thread of innovation across DDPMs and its two variants: DDIM and IDDPM. Combining the family of DDPMs, we provide a foundational understanding of this state-of-the-art model class, providing readers with a deep understanding of underlying diffusion theory, advantages and limitations. Most importantly, we highlight how to improve DDPMs to generate higher quality data, faster. Additionally, we demonstrated application of DDPMs to image inpainting via RePaint, highlighting the interoperable applications of using DDPMs as foundation models by applying them to a broad range of general purpose computer vision tasks.

We released our code here, which contains all models we used for images generation.

Limited Compute: A combined total of 60 hours GPU-HPC time, although sufficient to train DDPM and IDDPM models for $40,000$ iterations each ($\approx$ 260 epochs), was significantly less than the original papers. For example, (Nichol & Dhariwal, 2021) train over $1,500,000$ iterations per experiment (Nichol & Dhariwal, 2021). This compute discrepancy is directly attributed to reduced aesthetic quality in our sampled images, and limited us to exclusive use of the "*cars*" label in CIFAR-10. Despite this, we were still able to successfully replicate many key results: our IDDPM exhibits better log-marginal likelihoods, our DDIM model accelerates sampling times, DDPM obtains a poorer MSE loss during training, and repaint is able to infill unknown regions convincingly.

Conditional Models: Our denoising-diffusion-based models are unconditionally trained with respect to image labels. At present, the practitioner has no control over the class of image produced during denoising. A *conditional* generative model could yield improvements by conditioning image generation on class labels, enabling a degree of control over denoised results. Future work would investigate this valuable research direction.

Hyperbolic Tangent: Though our proposed hyperbolic tangent noising scheduler shows promising results, it remains an open question whether it can perform as well on more generalized datasets. With more time (and, again, more compute), we would have more thoroughly tested the results achieved with this new scheduler.

Physics: Inspired by physics, DDPMs have elegant mathematical explanations. Also, we show that an appropriate choice of diffusion (forward process) variance schedule could strongly benefit the learning process, while the choice of this schedule is yet based on experiments and is thus empirical. For example, Nichol & Dhariwal (2021) found that a slow noising rate at the beginning of diffusion would provide more training signal and help DDPMs train better, but there's still not general setting for it. Bridging physics and the variance schedule might help us to mathematically derive a generally appropriate variance schedule that provides most meaningful training signals for DDPMs. For example, the diffusion variance schedule might have multi-phases, rather than a sigmoid shape we introduced in Figure 8.

5. Acknowledgements

We heavily adapted a publicly available Jupyter notebook from this YouTube tutorial[1] which was used as a starting point for our DDPM model. However, DDIM, IDDPM, retraining, ablation, and RePaint implementations are our own work.

6. Word Count

Our word count is 4716.

References

Generative models. *Open AI Blog Post*, February 2024. URL https://openai.com/research/generative-models. [Online; accessed 20. Feb. 2024].

Barratt, S. and Sharma, R. A note on the inception score. *Proc. ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018. doi: 10.48550/arXiv.1801.01973. URL https://arxiv.org/abs/1801.01973.

Becker, E., Pandit, P., Rangan, S., and Fletcher, A. K. Instability and Local Minima in GAN Training with Kernel Discriminators. *Advances in Neural Information Processing Systems*, 35:20300–20312, December 2022. URL https://proceedings.

[1] https://www.youtube.com/watch?v=TBCRlnwJtZU

neurips.cc/paper_files/paper/2022/hash/7f9a44cb707ede42a659ad85d940dd55.

Biroli, G., Bonnaire, T., de Bortoli, V., and Mézard, M. Dynamical Regimes of Diffusion Models. *arXiv*, February 2024. doi: 10.48550/arXiv.2402.18491.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024.

Chong, M. J. and Forsyth, D. Effectively Unbiased FID and Inception Score and where to find them. *arXiv*, November 2019. doi: 10.48550/arXiv.1911.07023.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. Diffusion Models in Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(9):10850–10869, March 2023a. doi: 10.1109/TPAMI.2023.3261988.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023b.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Networks. *arXiv*, June 2014. doi: 10.48550/arXiv.1406.2661.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, 30, 2017.

Henighan, T., Kaplan, J., Katz, M., Chen, M., Hesse, C., Jackson, J., Jun, H., Brown, T. B., Dhariwal, P., Gray, S., Hallacy, C., Mann, B., Radford, A., Ramesh, A., Ryder, N., Ziegler, D. M., Schulman, J., Amodei, D., and McCandlish, S. Scaling Laws for Autoregressive Generative Modeling. *arXiv*, October 2020. doi: 10.48550/arXiv.2010.14701.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30:6626–6637, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Kingma, D. P. *Variational Inference & Deep Learning: A New Synthesis*. Thesis, fully internal, Universiteit van Amsterdam, 2017. [Thesis, fully internal, Universiteit van Amsterdam].

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. *arXiv*, December 2013. doi: 10.48550/arXiv.1312.6114.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. *arXiv*, January 2022. doi: 10.48550/arXiv.2201.09865.

Mescheder, L., Geiger, A., and Nowozin, S. Which Training Methods for GANs do actually Converge? In *International Conference on Machine Learning*, pp. 3481–3490. PMLR, July 2018. URL https://proceedings.mlr.press/v80/mescheder18a.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Nvidia. Fireside Chat with Ilya Sutskever and Jensen Huang: AI Today and Vision of the Future, May 2023. URL https://www.youtube.com/watch?v=-yquJiNKlAE. [Online; accessed 19. Feb. 2024].

Pascual, S., Bonafonte, A., and Serra, J. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv*, May 2015. doi: 10.48550/arXiv.1505.04597.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising Diffusion Implicit Models. *arXiv*, October 2020. doi: 10.48550/arXiv.2010.02502.

Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park,

K., and Lempitsky, V. Resolution-robust Large Mask Inpainting with Fourier Convolutions. *arXiv*, September 2021. doi: 10.48550/arXiv.2109.07161.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. pp. 2818–2826, 2016.

Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Wan, C., Probst, T., Van Gool, L., and Yao, A. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 680–689, 2017.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. Free-Form Image Inpainting with Gated Convolution. *arXiv*, June 2018. doi: 10.48550/arXiv.1806.03589.

Zhang, Z., Li, M., and Yu, J. On the convergence and mode collapse of GAN. In *SA '18: SIGGRAPH Asia 2018 Technical Briefs*, pp. 1–4. Association for Computing Machinery, New York, NY, USA, December 2018. ISBN 978-1-45036062-3. doi: 10.1145/3283254.3283282.

A. Foward diffusion sampling through one-step computation

Simply applying the Gaussian property under the Markov assumption, any sample x_t during the forward process can be sampled in closed form through one-step computation, instead of iteratively running the SDE:

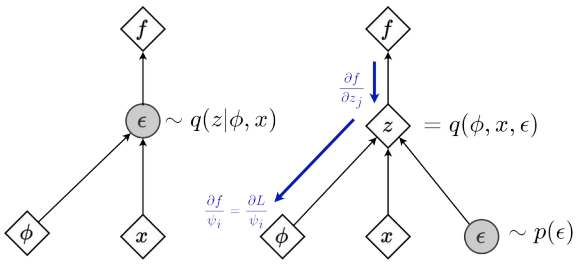

Figure 11. Graphical explanaition of forward diffusion sampling, which is adapted from (Kingma, 2017)

$$
\begin{aligned}
x_t &= \sqrt{a_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t}(x_{t-1} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-2}}\epsilon_{t-2}) + \sqrt{1-\alpha_{t-1}}\epsilon_{t-1} \\
&= \dots \\
&= \sqrt{\prod_{i=1}^{t} \alpha_i} x_0 + \sqrt{1 - \prod_{i=1}^{t} \alpha_i}\epsilon_0 \\
&= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0 \\
&\sim \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I)
\end{aligned} \tag{31}
$$

B. ELBO of DDPMs

In DDPMs (or IDDPM), we would like to model the denoising process, which is assumed Markovian: $p_\theta(x_{t-1}|x_t)$. While the diffusion (noising) process is designed to be Markovian towards a standard Gaussian noise. The orginal data is denoted as x_0. We thus optimize the parameter θ by variational inference:

$$
\begin{aligned}
\log p_\theta(x) := \mathbb{E}_{x_0} \log p_\theta(x_0) &= \mathbb{E}_{x_0} \log \int p_\theta(x_{0:T})\, dx_{1:T} && \text{(32a)} \\
&= \mathbb{E}_{x_0} \log \int \frac{p_\theta(x_{0:T}) q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)}\, dx_{1:T} && \text{(32b)} \\
&= \log \mathbb{E}_{x_0, q(x_{1:T}|x_0)} \left[\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] && \text{(32c)} \\
&\geq \mathbb{E}_{x_0, q(x_{1:T}|x_0)} \left[\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] \qquad \text{Evidence Lower Bound (ELBO)} && \text{(32d)}
\end{aligned}
$$

We then optimize the ELBO instead. Following the Markov assumption of diffusion and reverse process, we have: 1. $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}, x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$; 2. $p_\theta(x_{0:T}) = \Pi_{t=1}^{T} p_\theta(x_{t-1}|x_T) p_\theta(x_T)$; 3. $p_\theta(x_T) = N(x_T; 0, \mathbf{I})$. Thus we optimize:

$$
\theta^* = \arg\max_\theta \log p_\theta(x) = \arg\max_\theta \mathbb{E}_{x_0, q(x_{1:T}|x_0)} \left[\log \frac{p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})} \right] \tag{32e}
$$

In theory, directly optimising Equation (32e) is fine. However, x_{t-1}, x_t, x_{t-1} will each have large variance as the backward process is attempting to operate without knowledge of x_0. Ho et al. (2020) noted this variance could be drastically reduced, and performance improved, by conditioning on x_0. Notice that, by Bayes' rule:

$$
q(x_t|x_{t-1}) = q(x_t|x_{t-1}, x_0) = \frac{q(x_t, x_{t-1}, x_0)}{q(x_{t-1}|x_t)} = \frac{q(x_{t-1}|x_t, x_0) q(x_t|x_0)}{q(x_{t-1}|x_0)} \tag{32f}
$$

Plugging Equation (32f) into Equation (32e), we have:

$$\theta^* = \arg\max_\theta \mathbb{E}_{x_0, q(x_{1:T}|x_0)} \left[\log \frac{p_\theta(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^T \frac{q(x_{t-1}|x_t, x_0) q(x_t|x_0)}{q(x_{t-1}|x_0)}} \right] \tag{32g}$$

$$= \arg\max_\theta \mathbb{E}_{x_0, q(x_{1:T}|x_0)} \left[\log p_\theta(x_T) + \log p_\theta(x_0|x_1) + \sum_{t=2}^T \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} + \sum_{t=1}^T \log \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \right] \tag{32h}$$

$$= \arg\max_\theta \mathbb{E}_{x_0, q(x_T|x_0)} \left[\log N(x_T; 0, \mathbf{I}) \right] + \mathbb{E}_{q(x_1|x_0)} \left[\log p_\theta(x_0|x_1) \right] - \sum_{t=2}^T \mathbb{E}_{q(x_{t-1}|x_t, x_0)} \left[\log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} \right] \tag{32i}$$

$$= \arg\max_\theta \mathbb{E}_{x_0, q(x_1|x_0)} \left[\log p_\theta(x_0|x_1) \right] + \sum_{t=2}^T \mathbb{D}_{\text{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) \tag{32j}$$

$$= \arg\max_\theta L_{\text{vlb}} := L_0 + \sum_{t=2}^T L_{t-1} \tag{32k}$$

C. The "true" denoising process conditioned on x_0

Recap Equation (32f), we could derive the "true" denoising process by reversing that Baye's rule:

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0) q(x_{t-1}|x_0)}{q(x_t|x_0)} \tag{33a}$$

$$= \frac{\mathcal{N}(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)\mathbf{I}) \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}} x_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t)\mathbf{I})} \tag{33b}$$

$$\propto \exp\left(-\frac{1}{2} \left[\frac{(x_t - \sqrt{\alpha_t} x_{t-1})^2}{1-\alpha_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}} x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t} x_0)^2}{1-\bar{\alpha}_t} \right] \right) \tag{33c}$$

$$\tag{33d}$$

$$\propto \exp\left(-\frac{1}{2} \left[\frac{\alpha_t x_{t-1}^2 - 2\sqrt{\alpha_t} x_t x_{t-1}}{1-\alpha_t} + \frac{x_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}} x_0 x_{t-1}}{1-\bar{\alpha}_{t-1}} \right] \right) \tag{33e}$$

$$\propto \exp\left(-\frac{1}{2} \left[\left(\frac{1}{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}} \right) \left[x_{t-1}^2 - 2 x_{t-1} \left(\frac{\sqrt{\alpha_t} x_t (1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}} x_0 (1-\alpha_t)}{1-\bar{\alpha}_t} \right) \right] \right] \right) \tag{33f}$$

Thus,

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\alpha_t} x_t (1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}} x_0 (1-\alpha_t)}{1-\bar{\alpha}_t}, \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \right) \tag{33g}$$

$$= \mathcal{N}\left(x_{t-1}; \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t}_{\tilde{\mu}(x_t, x_0)}, \underbrace{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t}_{\tilde{\beta}_t} \right) \tag{33h}$$

D. Learning with fixed reverse variance

In this section, we show that when fixing the reverse variance $\sigma_t^2 = \beta_t$ (or $\tilde{\beta}_t$), the learning objective 32k could be rewritten as a simple MSE over the mean function μ_θ, where we model $p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2)$. According to the

Gaussian nature of p_θ and q (see Equation (33h)):

$$L_{t-1} = -\mathbb{E}_{x_0}\mathbb{E}_{\mathcal{N}(x_{t-1};\tilde{\mu}(x_t,x_0),\tilde{\beta}_t))}\left[\log\frac{\mathcal{N}(x_{t-1};\tilde{\mu}(x_t,x_0),\tilde{\beta}_t))}{\mathcal{N}(x_{t-1};\mu_\theta(x_t,t),\sigma_t^2))}\right] \tag{34a}$$

$$= \frac{1}{2}\mathbb{E}_{x_0}\mathbb{E}_{\mathcal{N}(x_{t-1};\tilde{\mu}(x_t,x_0),\tilde{\beta}_t))}\left[\frac{1}{\sigma_t^2}\left(\|x_{t-1}-\tilde{\mu}(x_t,x_0)\|^2 - \|x_{t-1}-\mu_\theta(x_t,t)\|^2\right)\right] + C \tag{34b}$$

$$= \frac{1}{2}\mathbb{E}_{x_0}\mathbb{E}_{\mathcal{N}(x_{t-1};\tilde{\mu}(x_t,x_0),\tilde{\beta}_t))}\left[\frac{1}{\sigma_t^2}\left((2x_{t-1}-\tilde{\mu}(x_t,x_0)-\mu_\theta(x_t,t))^T(\tilde{\mu}(x_t,x_0)-\mu_\theta(x_t,t))\right)\right] + C \tag{34c}$$

$$= \frac{1}{2}\mathbb{E}_{x_0}\mathbb{E}_{\mathcal{N}(x_{t-1};\tilde{\mu}(x_t,x_0),\tilde{\beta}_t))}\left[\frac{1}{\sigma_t^2}\left(2x_{t-1}^T(\tilde{\mu}(x_t,x_0)-\mu_\theta(x_t,t)) - \|\tilde{\mu}(x_t,x_0)\|^2 + \|\mu_\theta(x_t,t)\|^2\right)\right] + C \tag{34d}$$

$$= \frac{1}{2\sigma_t^2}\mathbb{E}_{x_0}\left[\|\tilde{\mu}(x_t,x_0)\|^2 - 2\tilde{\mu}(x_t,x_0)^T\mu_\theta(x_t,t) + \|\mu_\theta(x_t,t)\|^2\right] + C \tag{34e}$$

$$= \frac{1}{2\sigma_t^2}\mathbb{E}_{x_0}\|\tilde{\mu}(x_t,x_0)-\mu_\theta(x_t,t)\|^2 + C \tag{34f}$$

E. Reparameterization Trick

According to the diffusion process $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ where $\epsilon \sim \mathcal{N}(0,\mathbf{I})$, we have $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon)$. Then we could rewrite $\tilde{\mu}(x_t,x_0)$ as:

$$\tilde{\mu}(x_t,x_0) = \tilde{\mu}(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon)) \tag{35a}$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \times \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \tag{35b}$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)/\sqrt{\bar{\alpha}_t} + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\epsilon \tag{35c}$$

$$= \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) \tag{35d}$$

Since $\mu_\theta(x_t,t)$ is trained to approximate $\tilde{\mu}(x_t,x_0)$, alternatively, we could parameterise ϵ_θ and approximate the true noise ϵ instead:

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)) \tag{35e}$$

Thus, the learning objective becomes:

$$L_{t-1} = \frac{1}{2\sigma_t^2}\mathbb{E}_{x_0}\left[\|\frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t))\|^2\right] + C \tag{36a}$$

$$= \frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\mathbb{E}_{x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t,t)\|^2\right] + C \tag{36b}$$

Ignoring the scalars and estimating the sum of L_{t-1} via sampling, we could obtain the L_{simple} objective as follows:

$$L_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t,t)\|^2\right] \tag{37}$$

F. Marginal equivalence of the non-Markovian reverse process in DDIM

In this section, we're going to show that the non-Markovian reverse process $q_\sigma(x_{t-1}|x_t,x_0)$ leads to the same forward diffusion process as DDPMs, i.e. $q_\sigma(x_t|x_0) \equiv p(x_t|x_0) = N(x_t;\sqrt{\bar{\alpha}_t}x_0,(1-\bar{\alpha}_t)\mathbf{I})$.

$$q_\sigma(x_{t-1}|x_0) = \int q_\sigma(x_t,x_{t-1}|x_0)dx_t = \int q_\sigma(x_{t-1}|x_t,x_0)q_\sigma(x_t|x_0)dx_t \tag{38a}$$

$$= \int N(x_{t-1};\tilde{\mu}_\sigma(x_t,x_0),\sigma_t^2\mathbf{I})q_\sigma(x_t|x_0)dx_t \tag{38b}$$

Thus, we could prove this property by induction. Assume that $q_\sigma(x_{t-1}|x_0) = N(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0, (1-\bar{\alpha}_t)\mathbf{I})$, according to the Gaussian property, $q_\sigma(x_t|x_0)$ must be Gaussian as well, say $q_\sigma(x_t|x_0) = N(x_t|Ax_0, B)$. Then we have:

$$\sqrt{\bar{\alpha}_{t-1}}x_0 = \tilde{\mu}_\sigma(Ax_0, x_0) \tag{38c}$$

$$= \sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{Ax_0 - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}} \tag{38d}$$

$$\Rightarrow A = \sqrt{\bar{\alpha}_t} \tag{38e}$$

$$(1-\bar{\alpha}_{t-1})\mathbf{I} = C^T BC + \sigma_t^2\mathbf{I}, \quad \text{where } C = \frac{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}}{\sqrt{1-\bar{\alpha}_t}} \tag{38f}$$

$$= \frac{1-\bar{\alpha}_{t-1}-\sigma_t^2}{1-\bar{\alpha}_t}B + \sigma_t^2\mathbf{I} \tag{38g}$$

$$\Rightarrow B = (1-\bar{\alpha}_t)\mathbf{I} \tag{38h}$$

Thus, we prove that:

$$q_\sigma(x_t|x_0) = N(x_t|\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{38i}$$