Pro-DDPM: Progressive Growing of Variable Denoising Diffusion Probabilistic Models for Faster Convergence

Rohit Gandikota
gandikota.ro@northeastern.edu
Nik Bear Brown
ni.brown@northeastern.edu

Northeastern University
Boston, MA 02115, USA

Abstract

We describe a new training methodology to train Denoising Diffusion Probabilistic Models (DDPM) for faster convergence speeds. DDPMs have achieved high-quality image synthesis in non-adversarial training; however, their training is computationally intensive due to the Markov chain simulation and sampling for many time steps. The key idea of this work is to progressively increase the network depth of the diffusion model sinusoidally, starting from a low resolution and adding layers as they converge. This showed a 3 times improvement in training speeds while ensuring unprecedented image quality content generation. We compare our model's training performance with the original DDPM [12], the improved DDPM [22], diffusion implicit models [24], and generative models [2] in terms of FID and log-likelihood. We analyse the image quality and variance of the synthesised images on CelebA-HQ, Animal Faces-HQ, Imagenet and CIFAR-10 datasets. We also describe several implementation details of our training method that are important for quality preservation and faster convergence.

1 Introduction

Deep generative modelling has shown its ability to synthesise high-quality content in many domains (Karras *et al*. [19], Van den Oord *et al*. [27], Goodfellow *et al*. [10], Gandikota *et al*. [9]). In terms of image synthesis, Generative Adversarial Networks (GANs) and their variations have dominated the domain in terms of quality and variability. However, their major drawback is the convergence sensitivity owing to the adversarial training, specific architectural and optimisation choices for stable training (Arjovsky *et al*. [1]; Gulrajani *et al*. [12]; Karras *et al*. [18]; Brock *et al*. [3]). They could fail to show variability across the data distribution (Zhao *et al*. [30]).

GANs have previously outperformed likelihood-based models like variational autoencoders, autoregressive models and normalising flows. However, with the recent advent of Denoising Diffusion Probability Models (DDPM), the high-quality image synthesis has become stable due to eliminating adversarial settings in training. Essentially, DDPMs are trained to estimate the noise component of a given latent variable. It involves denoising the samples corrupted by various gaussian noise levels by progressively adding ϵ noise to

the training samples. During image synthesis after training, the model is sent through various denoising steps in a Markov chain. Starting from complete white noise, the model generates images that are sampled from the data distribution of the training sample space. DDPMs have proven to outperform the traditional GANs in image synthesis [8].

A critical drawback of DDPMs is the convergence speed being proportional to the Markov time steps and sample space size (both in resolution and sample size). For example, the convergence of a model to generate samples from the Animal Faces-HQ dataset containing 5000 cat image samples of size 128×1283 with 1000 Markov time steps has taken approximately 504 hours on Quadro GP100 GPU. This is because, in DDPMs, the reverse process approximates the gaussian noise added in the forward process; iterating over all the thousands of time steps is required to generate one single sample batch. Back-propagation at each timestep and each batch for learning the weights lead to a single training epoch, which is much slower in GPU compared to training an optimally designed GAN, that requires a single pass through the network for one training cycle. Secondly, the additive Markov noise model in high dimensions (>128) takes more epochs compared to the lower dimensions [14]. High-dimensional generative training requires large computational capacity and the complex loss landscape makes the training challenging.

This work addresses the high-dimensional training by progressively growing the diffusion model's layer depths: the Pro-DDPMS. Our critical insight is that, since lower resolutions can converge faster, adding layers to the lower resolution models and progressively increasing the resolution improves the convergence time compared to direct static high-resolution training. In section 3, we discuss the subtle implementation details we propose for improved training of DDPMs while augmenting the progressive growth training. Furthermore, we discuss the impact of the timestep choice in training the DDPMs compared to the Pro-DDPMs. In section 4, we evaluate Pro-DDPM on CIFAR10 [21], Imagenet [6], CelebA-HQ [17] and Animal Faces-HQ [5] datasets.

2 Background

This section provides a brief overview of the denoising diffusion probabilistic models. On a higher level, the diffusion model attempts to generate a sample from a distribution space by reversing a gradual noising process in a Markov fashion. To elaborate, the denoising starts from a complete white noise sample x_T and the model gradually attempts to generate less noisy samples $x_{T-1}, x_{T-2}, x_{T-3}, \ldots$ until it generates the final realistic sample x_0. Ho *et al.* [14] have modeled this process as a function $\varepsilon_\theta(x_t, t)$ where the model estimates the noise in the sample x_t at every time step t. To train this function, they sample a datapoint x_0 from the training dataset and gradually add noise ε to the resulting noisy samples. This builds a series of noisy sample $x_0, x_1, x_2, \ldots x_t$. This process is then reversed and the model is fed the input x_t and the estimated noise $\varepsilon_\theta(x_t, t)$ is compared with the true noise ε using the standard mean-squared error $|\varepsilon_\theta(x_t, t) - \varepsilon|^2$.

Ho *et al.* [14] show that in the reverse process (progressive Markovian denoising process), the model generates realistic sample x_0 from complete gaussian noise $\mathcal{N}(x_T, 0, \mathcal{T})$ by learning the joint distribution $p_\theta(x_{0:T})$. They model the denoising distribution at a lower level as a Markovian chain, where the reverse transition distribution depends on the previous timestep: $p_\theta(x_{t-1}/x_t)$. It is modelled as a diagonal gaussian function $\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t); \sum_\theta(x_t, t))$. The mean $\mu_\theta(x_t, t)$ can be modelled as a function of $\varepsilon_\theta(x_t, t)$. In theory, training the DDPMs requires minimising the variational upper bound of the negative log-likelihood as they are

Figure 1: Cat images generated with 1000 sampling steps using Pro-DDPMs conditioned over Animal-HQ dataset [6]. We observe a good diversity in sample poses, breeds and backgrounds, suggesting a good target distribution coverage.

likelihood-based probabilistic models, maximising the training data's likelihood. However, according to Ho *et al*. [14], minimising the simpler mean squared error $\mathcal{L}_{mse}$ works better in practice than actual variational lower bound $\mathcal{L}_{vlb}$. They also indicate that using the $\mathcal{L}_{mse}$ and the Markovian sampling is equivalent to the denoising score matching with the Langevin dynamics proposed by Song and Ermon [25]. Ho *et al*. [14] also suggest that their models naturally admit a progressively lossy decompression scheme that can be seen as a generalisation of autoregressive decoding.

With the proposal of the breakthrough work by Ho *et al*. [14], several recent works have proposed improvements to the diffusion models. Nichol and Dhariwal [22] have proposed a neural network to model the variance $\sum_\theta(x_t,t)$ parameter that was proposed as a constant by Ho *et al*. [14]. They argue that the choice of variance as a constant is sub-optimal when the sampling is done for fewer time steps. Nichol and Dhariwal [22] has also proposed an upgraded objective function that optimises the weighted sum between $\mathcal{L}_{mse}$ and $\mathcal{L}_{vlb}$. This objective function was used to reduce the number of sampling steps and improve the convergence speed throughout our work.

Song *et al*. [24] have proposed an implicit model, DDIM, that introduced an alternative process to the Markovian sampling process. They argue that the reverse process, being non-markovian, can produce different reverse samples by changing the noise variance. They provided a way to turn any model $\varepsilon_\theta(x_t,t)$ into a deterministic mapping by setting the noise to 0. They argue that images can be generated with fewer sampling steps than the DDPMs. Currently, we focus on models trained for >50 sampling steps; however, since Nichol and Dhariwal [22] have found the DDPMs with hybrid objective function and tuned parameterisation to be more efficient than DDIM, we only focus on the DDPMs with improvements proposed by Nichol and Dhariwal [22] in this work.

3 Progressive Growing of Diffusion Model

Our significant contribution is the improved training process of the DDPMs by implementing the progressive growth of the network layers. Our work inspires this idea by Karras *et al*. [17], where they progressively grow the depth of GANs. In contrast to their linear fading, we propose a more efficient sinusoidal fading growth in diffusion models. This method significantly impacts diffusion model training by allowing for a faster convergence as the objective function's manifold is less complex in lower dimensions owing to the smaller dimensional sample space. The training is progressive and takes fewer steps to converge as we ask a much simpler question at each progression than directly learning on the complex high-dimensional space. The sinusoidal growth improves learning during transitions between architectural growths through a non-linear fading mechanism.

We add new layers to the network as the training progress while keeping all the layers trainable throughout the training process. The addition of layers is done both in a faded and abrupt fashion; it is observed that the faded addition of new layers has an advantage as they do not abruptly disturb the previously trained network's progress. This allows the network to adjust the weights as we add new layers progressively.

Before introducing progressive growth in GANs by Karras *et al*. [17], similar ideas were introduced in a few other works. Wang *et al*. [28] proposed a multi-discriminator architecture that operates on various spatial zoom levels. Wang *et al*. [28] have taken inspiration from Durugkar *et al*. [16] who proposed the usage of multiple discriminators for a single generator in the exact spatial resolution, and Ghosh *et al*. [10] who, in contrast, use multiple generators. Hierarchical adversarial training (Denton *et al*. [7]; Huang *et al*. [15]; Zhang *et al*. [29]) that introduced GAN architectures for different image pyramids was one of the first works to propose this idea. In contrast to these works, our approach progressively adds layers to a single network in a predefined architecture format and an optimal sinusoidal fading mechanism.

3.1 Forward Process

The forward process involves progressively noising the samples. We briefly discuss the forward noising process from Ho *et al*. [14] and the improvements suggested by Nichol and Dhariwal [22]. Given a sample x_0 from data distribution, the forward process q that produces the latent noise samples $x_1, x_2, ...x_T$ by adding the gaussian noise with variance $\beta_t \in (0,1)$ at time step t is:

$$q(x_t/x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathcal{I}) \tag{1}$$

$$q(x_1, x_2, ...x_T/x_0) = \prod_{t=1}^{T} q(x_t/x_{t-1}) \tag{2}$$

To implement the progressive growth training, we require a dataset in all the dimensions, starting from the lowest resolution of 2×2 to the highest resolution. We implement the forward process only once at the highest possible resolution to enable this. During the training, for each resolution starting from 2×2, we reduce the stored samples from the forward process to the dimension under training; this will ensure consistency in training the network and avoid using the Markov chain multiple times. The sub-sampling of noisy forward samples to lower resolutions transforms the process into a variable noising as the discrete noise levels in higher resolutions will collapse into fewer noise mode levels in lower dimensions.

3.2 Reverse Process

This section describes the reverse process (denoising process) followed to implement progressive growth. If we know the exact reverse process $q(x_{t-1}/x_t)$, we can can sample a gaussian white noise sample $x_T \sim \mathcal{N}(0,\mathcal{I})$ and run the reverse process till we arrive at $q(x_0/x_1)$. However, it requires the knowledge of the entire data distribution to calculate $q(x_{t-1}/x_t$, we model the function using a deep network with parameter θ; a progressively growing network in our case:

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

To maximise the likelihood, considering the q and p processes like Kingma *et al.* [20], we can rewrite the objective function as KL divergence and thereby optimise the variation lower bound:

$$\mathcal{L}_{vlb} = \mathcal{L}_0 + \mathcal{L}_1 + ...\mathcal{L}_T \tag{4}$$
$$\mathcal{L}_0 = -\log p_\theta(x_0/x_1) \tag{5}$$
$$\mathcal{L}_{t-1} = \mathcal{D}_{KL}(q(x_{t-1}/x_t,x_0)||p_\theta(x_{t-1}/x_t)) \tag{6}$$
$$\mathcal{L}_T = \mathcal{D}_{KL}(q(x_T/x_0)||p(x_T)) \tag{7}$$

$\mathcal{L}_T$ does not depend on θ and will be very close to zero if the forward noise can destroy the data distribution purely such that $q(x_T/x_0) \sim \mathcal{N}(0,I)$. $\mathcal{L}_0$ can be computed by tracing the CDF of the gaussian distribution of the probability of each image falling in a bin of 256 color components. Ho *et al.* [14] pointed out that the forward process as formulated in Equation. 1, allows for random sampling from any noisy level conditioned on x_0. By defining $\alpha_t = 1-\beta_t$ and $\overline{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, we can formulate marginal distribution as:

$$q(x_t/x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathcal{I}) \tag{8}$$

Since $1-\overline{\alpha}_t$ talks about the variance at any arbitrary time step and hence can replace β_t. Assuming $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$ and $\tilde{\mu}_t(x_t,x_0) = \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}x_0 + \frac{\sqrt{\overline{\alpha}_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}x_t$, we can formulate the posterior as :

$$q(x_{t-1}/x_t,x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t,x_0), \tilde{\beta}_t\mathcal{I}) \tag{9}$$

To parametrize $\mu_\theta(x_t,t)$ in Equation. 3, we can directly model a neural network to directly predict $\mu_\theta(x_t,t)$. Else, we could model a neural network to predict x_0 that can be fed to definition of $\tilde{\mu}_t$ to produce $\mu_\theta(x_t,t)$. Ho *et al.* [14] proposed a neural network to predict the noise ε added to x_0, and this noise can predict x_0 as follows:

$$x_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon) \tag{10}$$

Now instead of fixing the $\sum_\theta(x_t,t)$ as a constant as proposed by Ho *et al.* [14], inspired by Nichol and Dhariwal [22], we chose to parametrise a neural network to predict this parameter indirectly. We use a model to output a parameter z that in turn is used to parametrise $\sum_\theta(x_t,t)$ as a log interpolation between β_t and $\tilde{\beta}_t$:

$$\sum_\theta(x_t,t) = e^{(z\log\beta_t + (1-z)\log\tilde{\beta}_t)} \tag{11}$$

This will enable us to solve for the variational lower bound along with the standard $\mathcal{L}_{mse}$ as suggested by Ho *et al.* [14]. Nichol and Dhariwal [22] suggested that unconstrained network

output on z has produced bounded $\sum_\theta(x_t,t)$ in practice. Therefore, the objective function we solve for is $\mathcal{L}_{final}$:

$$\mathcal{L}_{final} = \mathcal{L}_{mse} + \lambda \mathcal{L}_{vlb} \tag{12}$$

As suggested by Nichol and Dhariwal[22], we chose to apply stop gradient on $\mu_\theta(x_t,t)$ output of $\lambda\mathcal{L}_{vlb}$. This way $\mathcal{L}_{vlb}$ will guide $\sum_\theta(x_t,t)$ and $\mathcal{L}_{mse}$ will be majorly guiding $\mu_\theta(x_t,t)$.

3.3 Sinusoidal Faded Growth

After formulating the forward and reverse processes inspired by Ho *et al*. [14] and Nichol *et al*. [22], we establish various methods to incorporate progressive growth in diffusion models. To enable an automatic checkpoint for layer addition, we have implemented an early stop checkpoint based on the FID metric on generated samples. We have selected an early stop buffer of 15 epochs; this way, the model softly monitors the FID degradation point for 15 epochs and stops the training to add the growing layer based on the predefined architecture. The architecture details are discussed in the supplementary document. This ensures a faster and more intuitive way than the method followed by Karras *et al*. [17] where they choose an arbitrary number of epochs for each resolution. We find that to be time-consuming in training and analysis on design choice.

As proposed by Karras *et al*. [17], the growth can be done in 2 ways; faded and abrupt. We analyse the effects of both growing methods on DDPMs. As shown in Figure 2.b, the generated image quality across the training epochs for a simple DDPM [14] is improving gradually. In contrast, the FID of Pro-DDPM-generated images can proliferate in lower resolutions and converge faster than DDPMs. The effect of abrupt growth in Figure 3 can be seen as valleys in the FID curve due to the sudden addition of a new layer to a well-trained architecture. However, the faded addition does not have a similar effect on FID growth; the transitions were kept smooth, making the training faster than abrupt growth as the network was not pushed to a sudden valley. This made us choose the faded growth of layers over the abrupt growth.

We believe that the fading mechanism proposed by Karras *et al*. [17] is sub-optimal with the fading procedure. We propose a sinusoidally fading mechanism where the fading parameter is increased sinusoidally as training progresses. In Figure 3, we show the FID curves for the faded mechanism proposed by Karras *et al*. [17] and the sinusoidal faded growth. As can be seen, when we apply linear fading, the learning required at the beginning of change is much stronger than later. To address this, we introduce sinusoidal fading:

$$\lambda = \sin\left(\frac{\frac{t}{T}+s}{1+s}\cdot\frac{\Pi}{2}\right) \tag{13}$$

Where λ is the fading factor that allows the gradual introduction of a new layer, this is achieved by using a residual connection between the upsampled old layer output and the new layer output, as shown in the Figure 3. t is the time step beginning from the addition of the layer, and T is the time factor used to adjust the fading period. We choose $T = 100$, and an offset s is used to avoid extremely small values of λ at $t = 0$ since we found that having λ smaller than 0.1 at the beginning of fading made it harder to avoid more profound valleys; it gave the choice of $s = 0.7$. We increase the fading parameter λ sinusoidally from 0.1 to 1 as the training progresses.

4 Experiments

In this section, we discuss experiments that we conducted to evaluate the Pro-DDPMs improvement in training and to validate the quality of the generated images. The supplementary document provides a detailed description of the network and training settings. In this section, we concentrate on the training speeds and improvements that come with sinusoidal growth. We validate the quality of the generated images using Fréchet inception distance (FID), proposed by Heusel *et al*. [13]. We also compare Pro-DDPM's generated images with BigGAN proposed by Brock *et al*. [2] to compare the performance of the adversarial generative models to diffusion-based models.

4.1 Training Convergence Speed

The DDPM models (Pro-DDPM and improved DDPM by Nichol and Dhariwal [22]) are trained for 1000 diffusion steps and 700k epochs wherever applicable. The training progress is visualised in Figure 2.a where training computational time for every 25k epochs is plotted. The vertical dashed lines in the plot represent instances where we doubled the resolution by adding layers in the Pro-DDPM training. The lower resolutions converged faster due to simplicity in features and the lack of complex structures in the data. As the training progresses, the learning gets easier for the higher resolution indicating that the network is asked to learn progressively complex tasks. The convergence of Pro-DDPMs for a resolution of 128 is almost 3.5 times faster than the DDPMs [22]. However, it is essential to note that the inference time is still the same as the improved DDPMs [22] because ultimately, at the time of inference, both the models are identical; the only difference is the 3x faster training process.

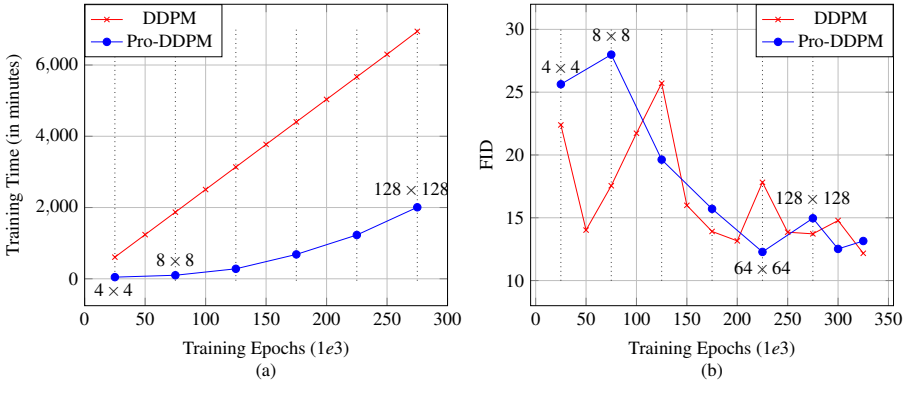

Figure 2: Effect of progressive growth of diffusion models on convergence speeds and image quality on Animal-HQ[5]. (a) The timings were registered for every 25k epochs on a Quadro GP100 GPU. (b) FID is calculated at every 25k epochs in the same setup. The blue curve represents Pro-DDPM training, and the vertical dashed lines in the graph are the checkpoints where we added layers to double the resolution. The red curve represents the DDPM training proposed by Ho *et al*. [14] and with improvements suggested by Nichol *et al*. [22] where the resolution is kept constant at 128×128. We observe more than 3 times faster convergence without compromising image quality using the progressive growth for DDPMs.

As shown in Figure 2.b, we also compare the performance differences over the training epochs using FID, a popular generative metric proposed by Heusel *et al*. [13]. FID calculation requires latent representations, and as many recent works [22] and [24], we also use the InceptionV3 model by Szegedy *et al*. [26] for extracting the latent features for comparison. For consistency in analysis across all the works and model compatibility, all the images are rescaled to the 128×128 resolution before passing through the InceptionV3 network. As the plot suggests, the full resolution FID of Pro-DDPM is not degraded noticeably compared to improved DDPMs [22]. We observe underperformance in terms of FIDs at lower resolutions for Pro-DDPM than the DDPM. This is due to the FID calculations at lower resolutions, i.e. the generated images are upsampled before passing through the InceptionV3 network. This makes it tricky to compare the models in lower resolutions as DDPM generations are compared at full resolution. The comparison is more apt at higher resolutions and indicates the Pro-DDPM's conservation of quality.

4.2 Effects of Sinusoidal Growth on Transitional Training Speed

As briefly discussed earlier, we propose a sinusoidal growing of network layers allowing for a smoother and faster transition of network training compared to the linear fading proposed by Karras *et al*. [17]. In Figure 3, we analyse the effect of sinusoidal growth at the transition instance (the instance when a new layer is added). As can be seen, the linear growth [17] converges slower compared to the sinusoidal growth. It is observed that the final steps are sub-optimal for linear growth as the network learning was faster at lower λ values and slower at higher values. However the sinusoidal growth converged at lower loss for all the values of λ as the schedule mimics linear growth at lower values and fades slower at higher values.

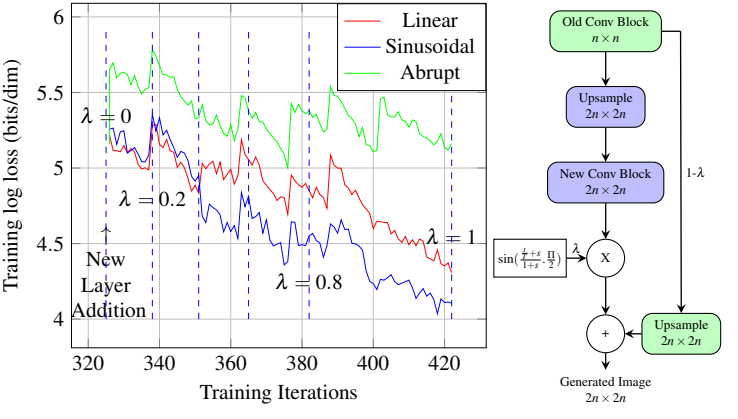

Figure 3: Learning curves comparing log-likelihoods achieved by sinusoidal growth and linear growth (at full resolution) during layer addition. The blue plot represents sinusoidal growth, the red plot represents linear growth [17] and the green plot represents abrupt growth. The blue dashed vertical lines represent the points where the sinusoidal λ is a multiple of 0.2. We observe an improvement in recovery time with sinusoidal growth compared to linear growth. When doubling the resolution, we fade in the layers using λ, as a sinusoidal function of time for the weighted growth of layers. Upsampling of the previous convolution layer is used as a residual addition to preserving the information from the previous iteration.

4.3 Results and Comparisons

This section briefly discusses the results of Pro-DDPM and compares its performance and quality with other latest models like DDPM [14], improved DDPM [22] and BigGAN [2]. We also present some visual results of the generated data of animal faces and CIFAR10 images. In Table 1, we show the FID comparison between BigGAN, Improved DDPM and Pro-DDPM over the generation of Imagenet 64×64 images [6]. Pro-DDPM generates images with quality that is similar to the Improved DDPM with a much faster training time. Pro-DDPMs outperform BigGAN in terms of FID over the Imagenet 64×64 dataset. In Table 2, we compare the log-likelihoods against DDPM and Improved DDPMs, showing that our training method has achieved relative likelihoods with less training time. However, the fully-transformer-based models outperform convolution-based models in terms of log-likelihood. Figure 1 shows our samples of cat images, and we provide many other results in the supplementary material.

Model	FID
BigGAN-deep [2]	4.06
Improved DDPM (large) [22]	2.92
Pro-DDPM (Ours)	2.96

Table 1: Sample quality comparison on ImageNet 64×64. FID [13] is measured using InceptionV3 [26] top network. BigGAN-deep was trained for 125k epochs without truncation as used by Nichol and Dhariwal [22] for their analysis.

Model	ImageNet	CIFAR-10
Sparse Transformer [4]	3.44	**2.80**
Routing Transformer [23]	**3.43**	-
DDPM [14]	3.77	3.70
DDIM [24]	-	3.10
Improved DDPM [22]	3.53	2.94
Pro-DDPM (Ours)	**3.61**	**3.09**

Table 2: Log-likelihood comparison of Pro-DDPM with other diffusion models and transformer models on ImageNet 64×64 [6] and CIFAR-10 [21]. Our model has a competitive likelihood with the existing DDPM models.

5 Conclusion

We have shown that with a few modifications and progressive growth, DDPMs can train much faster and achieve competitive log-likelihood and generative scores with a minor effect on image quality. We have found that sinusoidal fading is relatively more efficient than linear fading during the transitions. We have also shown that Pro-DDPMs can outperform the state-of-the-art adversarial generative models while achieving >3 times faster training than the previous DDPM models. This makes Pro-DDPMs, and diffusion models in general, an attractive choice for generative modelling. With good likelihoods and close to hyper-realistic images, there is still room for improvement regarding background focusing and microstructure preservations in the generated images.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

[4] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *URL https://openai.com/blog/sparse-transformers*, 2019.

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[7] Emily L Denton, Soumith Chintala, arthur szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/aa169b49b583a2b5af89203c2b78c67c-Paper.pdf.

[8] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

[9] Rohit Gandikota, Radha Krishna K, Anupama Sharma, ManjuSarma M, and Vinod M Bothale. Rtc-gan: Real-time classification of satellite imagery using deep generative adversarial networks with infused spectral information. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 6993–6996, 2020. doi: 10.1109/IGARSS39084.2020.9323363.

[10] Arnab Ghosh, Viveka Kulharia, Vinay P. Namboodiri, Philip H.S. Torr, and Puneet K. Dokania. Multi-agent diverse generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets.

In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[15] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1866–1875, 2017. doi: 10.1109/CVPR.2017.202.

[16] Sridhar Mahadevan Ishan Durugkar, Ian Gemp. Generative multi-adversarial networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byk-VI9eg.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020. doi: 10.1109/CVPR42600.2020.00813.

[20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL https://openreview.net/forum?id=33X9fd2-9FyZd.

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[22] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://openreview.net/forum?id=-NEXDKk8gZ.

[23] Aurko Roy*, Mohammad Taghi Saffar*, David Grangier, and Ashish Vaswani. Efficient content-based sparse attention with routing transformers, 2020. URL https://openreview.net/forum?id=B1gjs6EtDr.

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[25] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12438–12448. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

[27] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. In *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, page 125, 2016.

[28] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. doi: 10.1109/CVPR.2018.00917.

[29] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5908–5916, 2017. doi: 10.1109/ICCV.2017.629.

[30] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/5317b6799188715d5e00a638a4278901-Paper.pdf.