Accelerated Image-Aware Diffusion Modeling

Tanmay Asthana
Dept. of Electrical and Computer Eng
North Carolina State University
Raleigh, NC, USA
tasthan@ncsu.edu

Yufang Bao
Mathematics and CS Dept.
Fayetteville State University
Fayetteville, NC, USA
ybao@uncfsu.edu

Amro Awad
Dept of Engineering Science
University of Oxford
Oxford, UK
amro.awad@eng.ox.ac.uk

Hamid Krim
Dept. of Electrical and Computer Eng
North Carolina State University
Raleigh, NC, USA
ahk@ncsu.edu

***Abstract*—We propose in this paper an analytically new construct of a Diffusion Model whose drift and diffusion parameters yield an accelerated time-decaying Signal-to-Noise Ratio (SNR) in the forward process. This consequently reduces the number of time steps required to converge to pure noise. It further allows us to depart from conventional models, which typically use time-consuming multiple runs, by introducing a parallel data-driven model to generate a reverse-time diffusion trajectory in a single run. Our construct cleverly carries out the learning of the diffusion coefficients on the structure of clean images using an autoencoder. Collectively, these advancements yield a generative model that is at least 4 times faster than conventional approaches, while maintaining high fidelity and diversity in generated images, hence promising widespread applicability in rapid image synthesis tasks.**

***Index Terms*—diffusion models, generative models, accelerated generation**

I. Introduction

Generative diffusion models have recently emerged as powerful tools for image modeling and numerous other applications [1]–[4], offering exceptional fidelity and generative diversity [5]. In contrast to existing generative models, like generative adversarial networks (GANs) and variational autoencoders (VAEs), Diffusion Models (DM) are more stable in training and less sensitive to hyper-parameter selection [6].

While effective, the performance of Conventional Diffusion Models (CDMs) [3], [4] entails a slow convergence, with a quality image generation requiring in turn, a large number of time steps, thus leading to an increased computational complexity. To this end, much effort ([7]–[13]) has been dedicated to reducing this lengthy process. However, current models have mostly focused on reducing the reverse trajectory by employing sub-sampling or fast ODE solver based strategies. In this paper, we propose an alternative approach and use insights from statistical physics of particles to account for local (i.e. pixel level) SNR in driving the microscopic dynamics of the diffusion. Intuitively, one may interpret the conventional macroscopic forward diffusion as a parallel (bundle) process of microscopic forward diffusion processes occurring on individual pixels in parallel with same amount of noise added to each pixel. In our model, the forward diffusion scheduling is dependent, as detailed later, on the initial clean pixel values while each pixel maintains its own diffusion independent from others. In so doing, our proposed DM leverages the structure of clean image data to learn the drift and diffusion parameters at a microscopic level. This is inspired by the well known information theoretic water-pouring paradigm [14] used in multi-channel communication systems which allocates power to a channel in accordance with the noise-level experienced in that channel. We demonstrate that we can achieve the target goal of reaching isotropic Gaussian distribution on all the pixels much faster than the conventional pixel agnostic diffusion scheduling.

With such an image-aware forward diffusion in hand, we proceed with a variational autoencoder (VAE) to learn the combined diffusion schedule across all the pixels of a noisy image. While conventional models generate the reverse trajectory one step at a time, we leverage the structural information learned in the scheduling strategy to generate the whole reverse-time diffusion path in one go. As a result of this strategy, we are able to accelerate the reverse-time diffusion process by nearly an order of magnitude.

II. Methodology: Image Aware Diffusion

A. Motivation

In the forward direction of a conventional diffusion process, a clean image with d pixels, represented as $\mathbf{x}_0 = [x_0^1, \ldots, x_0^d] \in \mathbb{R}^d$ is diffused iteratively in T steps as

$$\mathbf{x}_{i+1} = \sqrt{\alpha_i}\mathbf{x}_i + \sqrt{1-\alpha_i}\epsilon_i, \;\; \epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{1}$$

where $\alpha_i \in (0,1), \; \forall \; i \in \{1, ..., T\}$, is a decreasing scalar schedule, $\mathbf{I}_d$ is an identity matrix. Consequently with large enough T, $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, which indicates that the diffusion of each pixel leads to approximately 0 SNR. The SNR degradation in forward diffusion direction across pixels can be parallely viewed as a dual of the water pouring algorithm employed in multi-channel communication systems [14]. The algorithm similarly addresses assignment of signal power distribution across frequency channels with different ambient noise powers to maximize SNR. Faced with our objective of all pixels simultaneously achieving approximately 0 SNR over a certain time interval (the total number of steps, T), it makes sense to diffuse higher-valued pixels at a faster rate than lower-valued pixels. As our first innovation herein, we propose a novel pixel-value-driven diffusion that is a departure from the existing SOTA DMs. The advantage of this pixel-aware diffusion over conventional pixel agnostic diffusion is a comparatively shorter trajectory in the forward direction of diffusion. Consequently, it shortens the corresponding reverse trajectory based on the same diffusion schedule.

Furthermore, the CDMs generate new samples by a reverse diffusion process which involves sequential sampling (over i ranging from T to 1) from the learned conditional posterior distributions, $p(\mathbf{x}_{i-1}|\mathbf{x}_i)$, with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, or with a uniformly distributed subsampling sequence, as discussed in Denoising Diffusion Implicit Model (DDIM) [7]. The sequential sampling entails multiple forward passes through a trained model, significantly increasing the over-all generation time. Theoretically, based on the Universal Approximation Theorem, the CDMs can also try to generate all the steps parallelly by relying on a prohibitively large neural network, however, it is much too difficult to train in its original format.

This highlights our second innovative contribution. We propose a parallel generation of the reverse diffusion that rely on a more informed prior, a rough estimate of the clean $\mathbf{x}_0$. This provides some early-scale feature information of the clean image such as image boundaries. It acts as a regularizer to our model. This, together with our fast pixel-wise diffusion, leads to a parameter complexity reduction thus affording a parallel generation of reverse diffusion steps.

B. Redefining forward diffusion

For a d-dimensional vector, $\mathbf{x}_0 = [x_0^1, \ldots, x_0^d] \in \mathbb{R}^d$ with $\mathbf{x}_0^j \in (0, 1]$, $j \in \{1, \cdots, d\}$ representing d pixels of a clean normalized image, we define image scale $\mathbf{x}_\delta \in \mathbb{R}^d$ as

$$\mathbf{x}_\delta \triangleq e^{-\gamma \mathbf{x}_0}, \tag{2}$$

where γ is a scalar hyperparameter such that $x_0^j << \gamma < T$, and exponentiation carried out element-wise.

The diffusion schedule parameters are now vectors with varying elements, redefined as

$$\boldsymbol{\alpha}_i = \mathbf{1} - \boldsymbol{\beta}_i = \mathbf{x}_\delta^{1/T}, \tag{3}$$

where $\mathbf{1} = [1, 1, \cdots, 1]$ is a d-dimensional vector.

This is in contrast to conventional Denoising Diffusion Probabilistic Model (DDPM) [3], in which the schedule parameters can be regarded as vectors with the same repeated element: $\boldsymbol{\alpha}_{C,i} = \mathbf{1} - \boldsymbol{\beta}_{C,i} = \alpha_i \mathbf{1}$, where α_i is a scalar independent of $\mathbf{x}_0$.

Our algorithm, scheduled per Eqn.-3, thus allows the diffusion rate to vary across all the pixels. The resulting reparametrized forward step dependent on $\mathbf{x}_0$ is written as

$$\mathbf{x}_i = \sqrt{\overline{\boldsymbol{\alpha}}_i} \odot \mathbf{x}_0 + \sqrt{1 - \overline{\boldsymbol{\alpha}}_i} \odot \tilde{\epsilon}_i, \tag{4}$$

where $\odot$ is an element-wise multiplication. $\overline{\boldsymbol{\alpha}}_i = e^{-\gamma\, i\, \mathbf{x}_0/T}$, $i \in \{0, \ldots, T\}$. $\tilde{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$.

To analyze the time trajectory of our diffusion model, we substitute the discrete ratio i/T with a continuous variable $t \in [0, 1]$, by letting $T \to \infty$. As a result, the discrete time-step Eqn.-4 becomes a continuous time diffusion:

$$\mathbf{x}_t = \sqrt{\overline{\boldsymbol{\alpha}}_t} \odot \mathbf{x}_0 + \sqrt{1 - \overline{\boldsymbol{\alpha}}_t} \odot \tilde{\epsilon}_t, \;\; \overline{\boldsymbol{\alpha}}_t = e^{-\gamma\, t\, \mathbf{x}_0/T}. \tag{5}$$

Proposition 1. *[15] The SNR of the j^{th} pixel at time t is:*

$$SNR(j, t) = \frac{(x_0^j)^2}{e^{\gamma\, t\, x_0^j} - 1}. \tag{6}$$

Eqn.-6 can be calculated using Eqn.-5. It clearly shows that the SNR of any pixel decreases exponentially with time. The SNR of higher value clean pixel is reduced at a faster rate than its lower value counterparts.

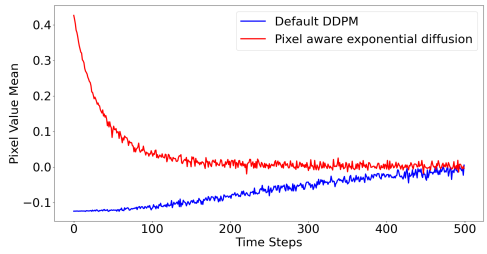
For a conventional DDPM with linear schedule of $\beta_t = at$, $t \in [0, 1]$, the expected trajectory can be shown [16] to be $\chi_C(t) = \mathbb{E}[\mathbf{x}_t] = \mathbf{x}_0 e^{-\frac{at^2}{2}}$. For our new diffusion model defined in Eqn.-5, the expected trajectory is $\chi_N(t) = \mathbb{E}[\mathbf{x}_t] = \mathbf{x}_0 \odot e^{-\frac{\gamma \mathbf{x}_0 t}{2}}$. This demonstrates that our diffusion has the advantage of a globally tunable decay rate (by setting $\gamma > at \;\; \forall\, t \in [0, 1]$), locally varying with pixel values.

Proposition 2. *[15] The selection $\gamma x_0^j > at$, $\forall\, j \in \{1, \cdots, d\}$, $t \in (0, 1]$ results in*

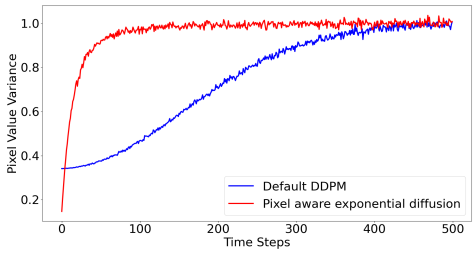
$$\left|\frac{\chi_C(t)}{dt}\right| < \left|\frac{\chi_N(t)}{dt}\right|. \tag{7}$$

From Eqn.-6, it is clear that the SNR of any pixel decreases exponentially with time and a higher valued clean pixel experiencing this reduction at a faster rate.

The CDMs inject equal noise power to all pixels. Their diffusion reaches 0 SNR state objective at a cost of an increased number of diffusion steps. Per Proposition-2, the proposed diffusion rate of convergence can be optimized beyond that of conventional processes by carefully choosing the hyperparameter γ. Fig.-1 shows a comparative progression of pixel mean and variance across the forward trajectory to those of other conventional diffusions.



(a) Pixel mean progression



(b) Pixel variance progression

Fig. 1: Comparison of mean (left) and variance (right) progression in forward diffusion trajectory of pixels using a single color channel (red color) over time of the conventional DDPM (blue) vs. our model (red).

As a result, a carefully chosen γ, achieves a faster convergence of our model over the conventional DDPM model. In our

experiments for CIFAR10 dataset images of 32×32 resolution, we fixed $T = 200$ and $\gamma = 20$. For CelebA dataset images of 128×128 resolution, we fixed these values to 500 and 50 respectively. These values were chosen by manually applying forward diffusion using Eqn.-5 over training set images, and verifying pixel mean and variance approach 0 and 1 over enough steps.

C. Modified Reverse Diffusion

In conventional DDPM based models [3], the goal is to generate a sample which has the same marginal probability as that for $\mathbf{x}_0$. This is achieved by a reverse diffusion process which includes sequential sampling (over i ranging from T to 1) from the learned conditional posterior distributions:

$$\begin{aligned} p_\phi(\mathbf{x}_{i-1}|\mathbf{x}_i) &= \mathcal{N}(\tilde{\mu}_\phi(\mathbf{x}_i, i)), \tilde{\boldsymbol{\beta}}_i \mathbf{I}), \\ \tilde{\mu}_\phi(\mathbf{x}_i, i) &= \frac{\mathbf{1}}{\sqrt{\boldsymbol{\alpha}_j}} \odot \left(\mathbf{x}_i - \frac{\mathbf{1} - \boldsymbol{\alpha}_j}{\sqrt{\mathbf{1} - \overline{\boldsymbol{\alpha}}_\mathbf{j}}} \odot \epsilon_\phi(\mathbf{x}_i, i) \right), \\ \tilde{\boldsymbol{\beta}}_{C,i} &= \frac{1 - \overline{\boldsymbol{\alpha}}_{i-1}}{1 - \overline{\boldsymbol{\alpha}}_i} \odot \boldsymbol{\beta}_i, \end{aligned} \tag{8}$$

where $\epsilon_\phi(\mathbf{x}_i, i)$ is the denoising output produced by a neural network with learned parameters, ϕ.

The sampling posterior for our generative algorithm would additionally require an estimate of image scale $\mathbf{x}_\delta = e^{-\gamma \mathbf{x}_0}$ as the diffusion scheduling parameters. At first glance, this appears to be a counter-intuitive task as acquiring $\mathbf{x}_0$ through a stochastic trajectory seems to require knowledge of $\mathbf{x}_0$ itself. To avoid this dilemma, we take advantage of the observation that as a result of exponentiation and the large value of γ, $\mathbf{x}_\delta$ or an approximation thereof only provides coarse structural information. With fewer fine details, it provides some prior information of the image structure. Consequently, we exploit a VAE, a less complex denoiser to estimate $G_\theta(\mathbf{x}_i, i) = \hat{\mathbf{x}}_\delta \approx \mathbf{x}_\delta$ from a noisy image $\mathbf{x}_i$. The requirement on parameter complexity of $G_\theta(\mathbf{x}_i, i)$ can be kept low because $\mathbf{x}_\delta$ lacks finer details (unlike $\mathbf{x}_0$). The image scale, $\mathbf{x}_\delta$ and γ are independent of time-steps. Once $\mathbf{x}_\delta$ is estimated, the approximations of the factors $\boldsymbol{\alpha_i}^j$ and $\overline{\boldsymbol{\alpha}}_i^j$ in Eqn.-8 can also be readily calculated to further recover $\mathbf{x}_0$.

Fig.-2 illustrates the comparison of the real vs generated $\mathbf{x}_\delta$ via the VAE decoder applied on pure noise samples for CelebA dataset. The structural similarity (SSIM) index between real and estimated images wasin the range $[0.86, 0.99]$ for both $\mathbf{x}_\delta$ and $\overline{\boldsymbol{\alpha}}_i$ (the higher the better, with 1 signifying perfect similarity) across different values of i.

Fig. 2: Real and generated $\mathbf{x}_\delta$ examples for CelebA (A database of human faces) dataset.

D. Reverse-time diffusion modelling

To proceed with a reverse-time diffusion we estimate $\hat{\mathbf{x}}_\delta$, $\boldsymbol{\beta}_i$, $\boldsymbol{\alpha}_i$ and $\overline{\boldsymbol{\alpha}}_i$ as discussed in the previous subsection. In conventional designs the same trained network is used to generate the reverse trajectory samples by using previously generated sample and next time-step positional encoding as input. This provides evidence that the architecture has a sufficient capacity to process the semantic information hidden in the noisy image at any time-step. The overall architecture of our parallelized reverse diffusion model using a U-net architecture, is shown in Fig.-3. In adapting it to our proposed methodology, the following modifications are in order:

1) We also fuse (by addition to feature maps) $\hat{\mathbf{x}}_\delta$ predicted from the image scale autoencoder $G_\theta(x_i, i)$.
2) We modify the structure of the last layer of the model to predict the additive noise, $\mathbf{Z}^k$ for all the preceding time-steps ($k \in \{i-1, \cdots, 1\}$) in different channels of the last layer. While the one time complexity of our model is higher than existing competing models, unlike the CDMs, only a single execution is required of the trained model to obtain a clean image $\mathbf{x}_0$. This thus results in an overall reduction in sample generation time.

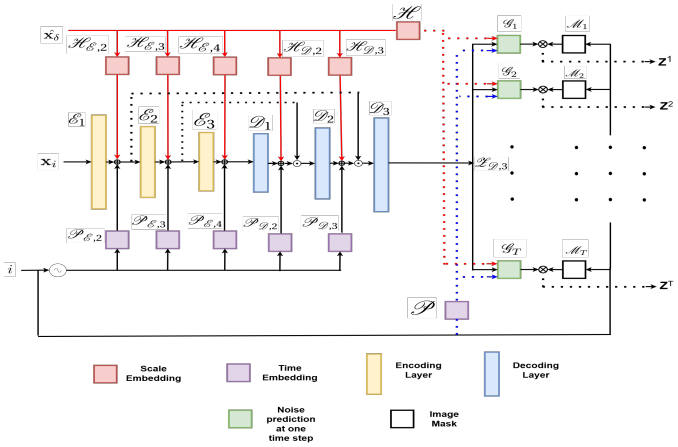


Fig. 3: Reverse diffusion model architecture

The model $R_\phi(\mathbf{x}_i, \hat{\mathbf{x}_\delta}, i)$ (with trainable parameters, ϕ) receives a noisy image $\mathbf{x}_i$, its predicted scale $\hat{\mathbf{x}_\delta}$ and time step information i as input and predicts additive noise for all the steps of the forward diffusion in parallel. These predictions can then be used to generate the reverse trajectory using Eqn.-8 with $\epsilon_\phi(x_j, j)$ replaced by predictions, $\mathbf{Z}^j$ of the model.

The last layer of the model has T feature maps. The j^{th} feature map, $\mathbf{Z}^j$, $j \in J = \{1, \cdots, T\}$, is:

$$\mathbf{Z}^j = \mathscr{M}_j \odot \mathscr{G}_j(\mathscr{Z}_{\mathscr{D},K}, \mathscr{P}(i), \mathscr{H}(\hat{\mathbf{x}}_\delta); \phi_j), \tag{9}$$

where $\phi_j \subset \phi$. $\mathscr{M}_j$ is a channel mask with $1's$ if $j < i$, and $0's$ otherwise. This ensures that only predictions for time-steps preceding i are made. $\mathscr{G}_j(.)$ is a feature map implemented using a small neural network. The fusion of $\hat{\mathbf{x}}_\delta$ allows us to reduce the parameter complexity of $\mathscr{G}_j(.)$. $\mathscr{P}(i)$ is a non-linear mapping of the input time-step i with the same dimensions as

a single channel of the decoder output, $Z_{\mathscr{D},K}$ and $\mathscr{H}(\hat{\mathbf{x}}_\delta)$ is a non-linear mapping of $\hat{\mathbf{x}}_\delta$. This mapping is fused with $\mathscr{Z}_{\mathscr{D},K}$, the decoder output, before being fed to $\mathscr{G}_j(.)$.

All T optimization objective functions are similar to those used in conventional DDPM [3]:

$$L(\phi, j) = \mathbb{E}_{i,x_i}[||\mathbf{Z}^j - \tilde{\epsilon_j}||_2^2], \;\; j \in \{1, \cdots, T\} \tag{10}$$

For $j > i$, $L(\phi, j)$ is fixed to 0 as a consequence of the same argument of using the mask $\mathscr{M}_j$. The parameters of the common network backbone, $\phi_b = \{\phi_l | \phi_l \in \phi, \phi_l \notin \phi_j, \forall j \in J\}$, are trained by all the $L(\phi, j \in J = \{1, \cdots, T\})$, whereas each parameter ϕ_j of a particular $\mathscr{G}_j(.)$ is trained by only optimizing its particular loss function $L(\phi, j)$, and the opimatization is obtained in parallel.

The procedure for generating the final clean image is shown in Algorithm 1. It is similar to the one used by [3], the difference being that the scheduling parameters are calculated from $\hat{\mathbf{x}}_\delta$ and only a forward pass through the model $R_\phi(.)$ is required to predict all denoising terms $\epsilon_\phi(., i)$ in Eqn.-8, as they are available in parallel as $\mathbf{Z}^j$.

Algorithm 1 Sampling algorithm

Require: Pre-trained scale autoencoder $G_\theta(.)$ and reverse diffusion model, $R_\phi(.)$.

Input: Noisy image $\mathbf{x}_i$ and time-step i of the forward diffusion

Output: Clean image $\hat{\mathbf{x}_0}$

Scale estimate: $\hat{\mathbf{x}_\delta} = G_\theta(\mathbf{x}_i, i)$

$$\boldsymbol{\alpha}_j = \exp\left\{(\frac{1}{T}\log(\hat{\mathbf{x}_\delta}))\right\}, \;\; \forall j \in 1, \cdots, T$$

$$\overline{\boldsymbol{\alpha}}_j = \exp\left\{(\frac{j}{T}\log(\hat{\mathbf{x}_\delta}))\right\}, \;\; \forall j \in 1, \cdots, T$$

$$\tilde{\boldsymbol{\beta}}_j = \frac{\mathbf{1} - \overline{\boldsymbol{\alpha}}_{\mathbf{j-1}}}{\mathbf{1} - \overline{\boldsymbol{\alpha}}_{\mathbf{j}}}(\mathbf{1} - \boldsymbol{\alpha_j}), \;\; \forall \mathbf{j} \in \mathbf{1}, \cdots, \mathbf{T}$$

Reverse diffusion noise predictions: $\mathbf{Z} = \{\mathbf{Z}^1, \cdots, \mathbf{Z}^T\} = r_\phi(\mathbf{x}_i, \hat{\mathbf{x}_\delta}, i)$

Initialization $j = i, \;\; \hat{\mathbf{x}}_j = \mathbf{x}_i$

while $j > 0$ **do**

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\hat{\mathbf{x}}_{j-1} = \frac{\mathbf{1}}{\sqrt{\boldsymbol{\alpha}_j}} \odot \left(\hat{\mathbf{x}}_j - \frac{\mathbf{1} - \boldsymbol{\alpha_j}}{\sqrt{\mathbf{1} - \overline{\boldsymbol{\alpha}}_{\mathbf{j}}}} \odot \mathbf{Z}^j\right) + \sqrt{\tilde{\boldsymbol{\beta}}_j} \odot \mathbf{z}$$

$j = j - 1$

end while

return $\hat{\mathbf{x}}_0$

III. Experiments and Results

The models were trained on Cifar10 and CelebA datasets for fair comparison with other models. The images were first normalized to the range $(\epsilon, 1]$. Note that a small $\epsilon = 4 \times 10^{-3}$ is added to all x_0^j to ensure their values are greater than 0 so that $\overline{\boldsymbol{\alpha}}_i$ vary with i. Time-step inputs to the modified U-net for the reverse diffusion model were encoded using sinusoidal positional embedding [17].

Fig.-4 shows some generated examples for CIFAR10 and CelebA datasets. While recent improvements are mostly focusing on speeding up the reverse diffusion by implementing faster solvers, improvements on forward diffusion are limited. Fast solvers-driven methods like [18] have a fast computational goal which is different from our diffusion approach. SlimFlow [19] is a framework that trains compact, one-step generative models by distilling knowledge from large diffusion models using a modified Rectified Flow framework [20]. DeepCache [21] accelerates the generation process by caching high-level features from previous steps and updating only low-level features.

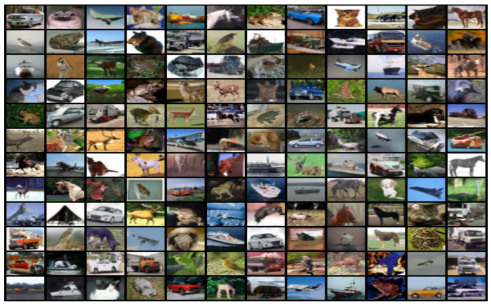
Table I illustrates generative performance of these models on the CIFAR10 dataset using trainable parameter complexity, FID scores and execution time. We compared our algorithm to a similarly discrete time model DDPM [3] based on a Stochastic Differential Equation(SDE) based model introduced by Song et al. [4], and its accelerated versions, namely DDIM [7], Fast DPM Solvers [18], Slimflow [19] and DeepCache [21]. We used the best DeepCache model (generation quality) which caches the high level features on every other step (N=2). Table II shows the comparisons for CIFAR10 dataset. Training and inference were performed on a single NVIDIA Tesla V100 SXM2 32 GB GPU. Most research efforts, such as [18]–[23], focus on fast sampling using fast ODE solvers applied to the backward diffusion of SDE based model. A continuous time version of our model will appear in a future paper as it is out of scope of the present paper due to limited space. These fast solvers will be compatible and can be applied to our continuous time model as well, providing further acceleration.

While the image quality of our model is competitive, its execution time is at least 4 times less than that of DDPM. They are slightly less than DDIM and Fast DPM solvers even when more time steps are used, without any compromise on the generation quality. Unlike the SDE-based model that needs orders of magnitude more steps due to MCMC subsampling corrections, our model achieves comparable performance with just 200 time-steps and a 500-step trajectory. Our model has more trainable parameters due to the usage of image scale estimation and the multiple parallel channels. However, this burden is compensated by just a single forward pass required by our model. Even simplified and distilled models like [19] and [21] with reduced execution time, produce inferior quality results, with our model exhibiting a better generation quality vs time trade-off.

Increased data complexity in the case of higher resolution images will undoubtedly increase the model complexity for all algorithms. However, as is evident from the parameters count in Tables I and II, moving from 32 x 32 to 128 x 128 resolution only resulted in approximately 2x increase in the number of trainable parameters, which is minor in comparison to conventional diffusion models.

IV. Conclusion

We have introduced in this paper, a novel forward diffusion model and backward recovery which significantly improve the convergence speed and computational efficiency limitations of conventional models. With an overall accuracy and execution time advantage over conventional models, a systematic selec-



(a) CIFAR10 examples

(b) CelebA examples

Fig. 4: Image generation examples

Cifar10 generative performance				
Model	#Param(M)	#Steps	FID	Time(in sec)
DDPM	35.7	1000	3.28	1.26
SDE based	31.4	1000	2.99	47.67
DDIM	35.7	10	13.36	0.03
DDIM	35.7	100	4.16	0.33
DDIM	35.7	1000	4.04	3.22
DPM Discrete Solver	35.7	10	5.37	0.02
DPM Discrete Solver	35.7	100	3.94	0.15
DPM Discrete Solver	35.7	500	3.41	0.76
DeepCache	35.7 (N=2)	100	4.56	0.1
SlimFlow	15.7	1	5.02	0.004
Our Model	71.5	200	**3.15**	**0.3**

TABLE I

CelebA generative performance				
Model	#Param(M)	#Steps	FID	Time(in sec)
DDPM	78.7	1000	3.51	10.19
SDE based	65.6	1000	3.20	246.69
DDIM	78.7	10	17.33	0.53
DDIM	78.7	100	6.53	5.55
DDIM	78.7	1000	3.51	48.44
DPM Discrete Solver	78.7	10	4.85	0.04
DPM Discrete Solver	78.7	100	4.52	0.33
DPM Discrete Solver	78.7	500	3.79	1.48
Our Model	145.5	500	**3.25**	**1.3**

TABLE II

tion of a hyperparameter γ figures in our future plan [15]. This entails considering the joint the data/pixels distribution. Uncovering the control of a diffusion process for a system of interacting particles [24], [25] in lieu of independent forward diffusions lies, we believe, at the center of this challenge.

REFERENCES

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," In International Conference on Machine Learning, Francis R. Bach and David M. Blei (Eds.), 2015.

[2] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," In Advances in Neural Information Processing Systems, 2019.

[3] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in Neural Information Processing Systems, 2020.

[4] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=PxTIG12RRHS

[5] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," ACM Computing Survey, 2023.

[6] M. Welling and D. Kingma, "An introduction to variational autoencoders," arXiv:1906.02691v3, 2019.

[7] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," ArXiv, vol. abs/2102.09672, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231979499

[8] A. Jolicoeur-Martineau, R. Piche-Taillefer, R. T. des Combes, and I. Mitliagkas, "Fast sampling of diffusion models with exponential integrator," ArXiv abs/2009.05475, 2021.

[9] T. Salimans and J. Ho, "Adversarial score matching and improved sampling for image generationprogressive distillation for fast sampling of diffusion models," ICLR 2021, 2021.

[10] Q. Zhang and Y. Chen, "Adversarial score matching and improved sampling for image generation," ICLR 2023 arXiv preprint arXiv:2204.13902 (2022), 2023.

[11] T. Dockhorn, A. Vahdat, , and K. Kreis, "Genie: Higher-order denoising diffusion solvers," Advances in Neural Information Processing Systems (2022), 2022.

[12] Z. Lyu, X. Xu, C. Yang, D. Lin, , and B. Dai, "Accelerating diffusion models via early stop of the diffusion process," arXiv preprint arXiv:2205.12524 (2022), 2022.

[13] H. Zheng, P. He, W. Chen, , and M. Zhou, "Truncated diffusion probabilistic models," arXiv preprint arXiv:2202.09671 (2022, 2022.

[14] R. G. Gallager, Information Theory and Reliable Communication. USA: John Wiley & Sons, Inc., 1968.

[15] T. Asthana, Y. Bao, and H. Krim, "Accelerated image-aware generative diffusion modeling," 2024. [Online]. Available: https://arxiv.org/abs/2408.08306

[16] S. Särkkä and A. Solin, Applied stochastic differential equations. Cambridge University Press, 2019, vol. 10, pp. 69–72.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[18] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," in Advances in Neural Information Processing Systems, 2022.

[19] Y. Zhu, X. Liu, and Q. Liu, "Slimflow: Training smaller one-step diffusion models with rectified flow," in European Conference on Computer Vision. Springer, 2024, pp. 342–359.

[20] Q. Liu, "Rectified flow: A marginal preserving approach to optimal transport," arXiv preprint arXiv:2209.14577, 2022.

[21] X. Ma, G. Fang, and X. Wang, "Deepcache: Accelerating diffusion models for free," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 15 762–15 772.

[22] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in Proceedings of the 40th International Conference on Machine Learning, ser. ICML'23. JMLR.org, 2023.

[23] T. S. . J. Ho, "On drift, diffusion and geometry," Journal of Geometry and Physics, vol. 56, pp. 1215–1234, 2006.

[24] Y. Bao and H. Krim, "Smart nonlinear diffusion: a probabilistic approach," IEEE Trans Pattern Anal Mach Intell. doi: 10.1109/tpami.2004.1261079, vol. 26(1), pp. 63–72, 2004. [Online]. Available: https://ieeexplore.ieee.org/document/1261079

[25] H. Krim and Y. Bao, "Nonlinear diffusion: A probabilistic view," Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), vol. 2, pp. 21–25 vol.2, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:7963466