Denoising Diffusion Probabilistic Model (NeurIPS’20)

Jonathan Ho, Ajay Jain, Pieter Abbeel

Sanghyeon Lee

03 May. 2021

Introduction

Contribution

- Generate high quality image synthesis using diffusion probabilistic models which is inspired by considerations from nonequilibrium thermodynamics (**State-of-the-art FID score for CIFAR10 Dataset)**
- Authors show connections to denoising score matching + Langevin dynamics

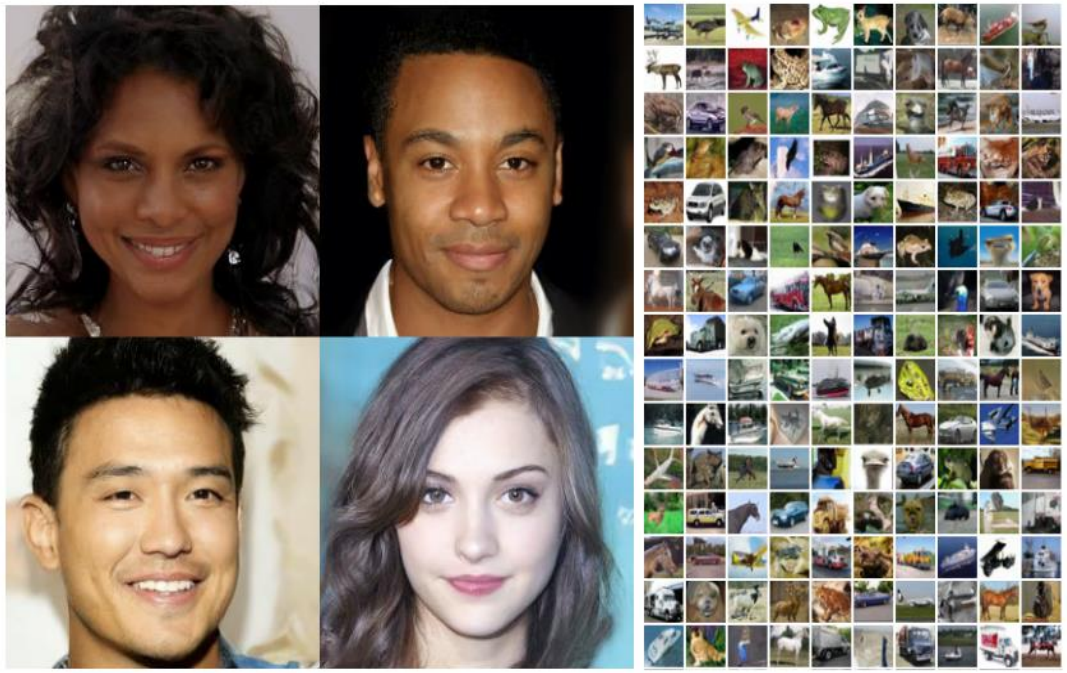
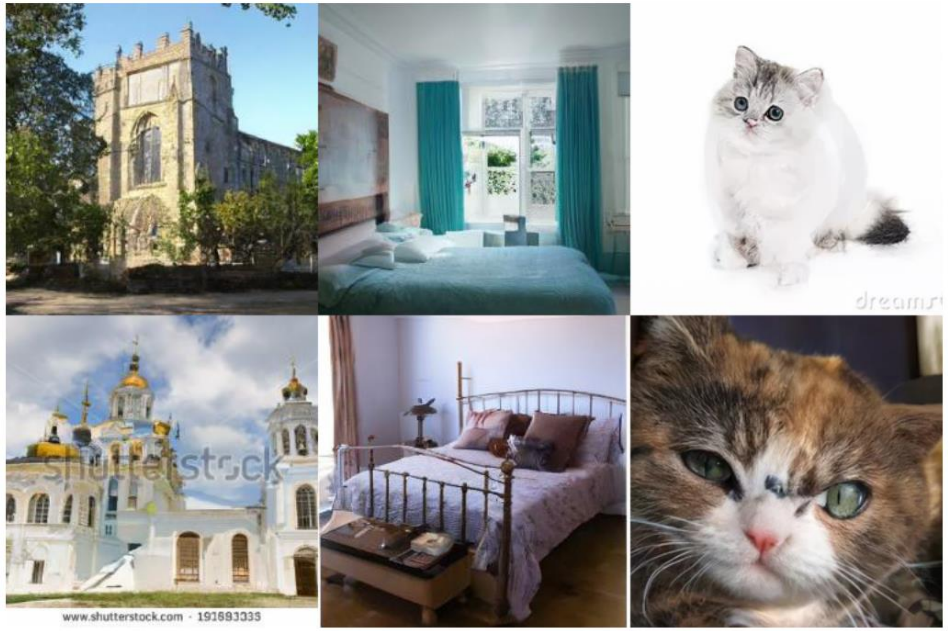


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)



LSUN 256x256 Church, Bedroom, and Cat samples. Notice that our models occasionally generate dataset watermarks.

Recap the VAE

Goal: Find $p_\theta(x)$ via MLE (Finding good representation for real world domain x)

Why using ELBO?: $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$ ➔ Intractable (z: hidden variable, $\in R^n$)

EM is general algorithm for find MLE or MAP but posterior term ($p_\theta(z|x) = \frac{p_\theta(X|Z)p_\theta(z)}{p_\theta(x)}$) of EM Algorithm is also intractable

& MCMC or Monte Carlo EM is too slow because of the parallelization

ELBO: Maximize $L(\theta, \phi; x^i) = -KL(q_\phi(z|x^i)||p_\theta(z)) + E_{q_\phi(z|x^i)}[log\ p_\theta(x^i|z)]$

SGVB, AEVB Algorithm ?

➔ Generalized representation of reparameterization trick and batch-wise training

$z \sim q_\phi(z|x) \Rightarrow z = g_\phi(\epsilon, x)\ where\ \epsilon \sim p(\epsilon)$ then we can calculate the gradient

(Derivation for random variable z is not defined)

$$\mathrm{E}_{q_\phi(z|x)}[\mathrm{f}(z)] = \int q_\theta(z|x)f(z)dz = \int p(\epsilon)f(z)d\epsilon = \int p(\epsilon)f\left(g_\phi(\epsilon, x)\right)d\epsilon = \mathrm{E}_{p(\epsilon)}[\mathrm{f}(g(\epsilon, x^i)]\ ,$$

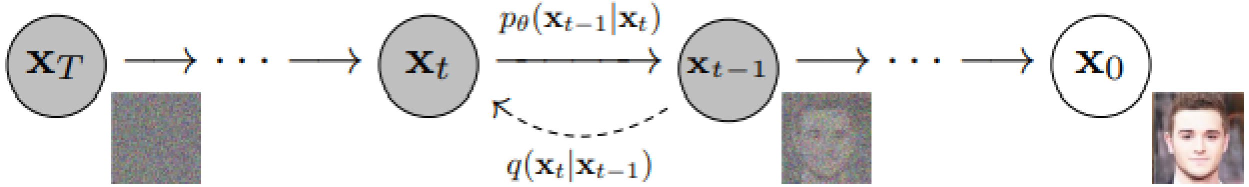
$where$ $q(z|x)dz = p(\epsilon)d\epsilon$ $(given\ diterminstic\ mapping)$!!

We can calculate ELBO term in closed form in Gaussian assumption !

In the inference time, VAE sample the latent variable z from standard multivariate gaussian: Training Encoder with KL regularization : $-KL(q_\phi(z|x^i)||p_\theta(z))$

Background

Diffusion model: Latent variable model of the from $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$
Where $\mathbf{X}_1, \ldots, \mathbf{X}_\mathbf{T}$: latents of the same dimensionality as the data $\mathbf{X}_0 \sim \boldsymbol{q}(\mathbf{X}_0)$


$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

Reverse Process $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

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

Forward Process(posterior):
$\boldsymbol{\beta}_t$: variance schedule(learnable parameter or hyperparameter)

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

$$\boldsymbol{q(x_t|x_{t-1}) \sim N\left(\mu_t + \Sigma_{t,t-1}\Sigma^{-1}_{t-1,t-1}(x_{t-1} - \mu_{t-1}), \Sigma_{t,t} - \Sigma_{t,t-1}\Sigma^{-1}_{t-1,t-1}\Sigma_{t-1,t}\right) \; where}$$

$$\boldsymbol{x_{t-1} \sim N\left(\sqrt{1-\beta_{t-1}}\mu_{t-1}, \left(\sqrt{1-\beta_{t-1}}\right)\Sigma\left(\sqrt{1-\beta_{t-1}}\right)^t\right) \Rightarrow let\; N(\mu_{t-1}, \Sigma_{t-1})}$$

$$\boldsymbol{q(x_t|x_0) = N(x_t; \sqrt{\Pi_i^t(1-\beta_i)}\; x_0, (1 - \Pi_i^t(1-\beta_i)I))}$$

➔ In the forward process, previous information(x_{t-1}) is decaying with $\sqrt{1-\beta}$

Background

ELBO of Diffusion model

Minimize –ELBO => L =
$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{L_T}+\sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

KL divergence term in ELBO are tractable when condition on $\boldsymbol{x_0}$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_{t-1};\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0),\beta_t\mathbf{I}),$$

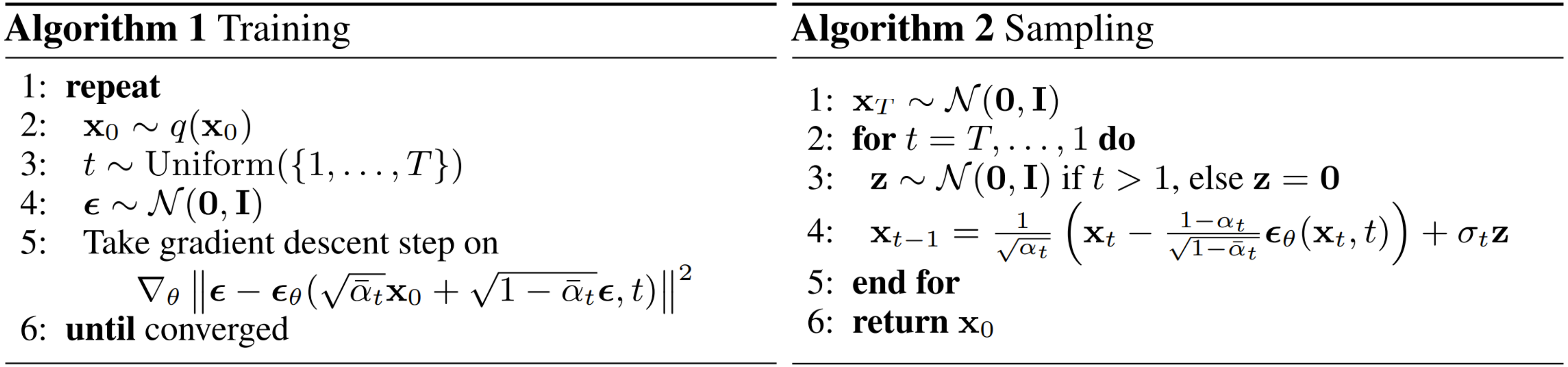
$$\text{where}\quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0):=\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0+\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t\quad\text{and}\quad\tilde{\beta}_t:=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

What is different?

- **Diffuse the variance via scheduling parameter $\boldsymbol{\beta}$ in the posterior (Enc), (Original diffusion model , but this paper design posterior with $\boldsymbol{\mu}$ & $\boldsymbol{\Sigma}$)**
- **VAE learn the $\boldsymbol{\mu}$ & $\boldsymbol{\Sigma}$ in the Enc**
- **Probabilistic model using Markov chain (It is different from the Progressive VAE models)**

What is same?

- **Learnable parameters of Enc & Dec : $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ & $\boldsymbol{\beta}$ (Enc: $\boldsymbol{q_\phi(x_0|x_1)}$ … $\boldsymbol{q_\phi(x_{T-1}|x_T)}$, Dec: $\boldsymbol{p_\theta(x_{0:T})} =$ $\boldsymbol{p_\theta(x_0|x_1)}$ …)**
- **ELBO**



Algorithm 1 Training

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

Algorithm 2 Sampling

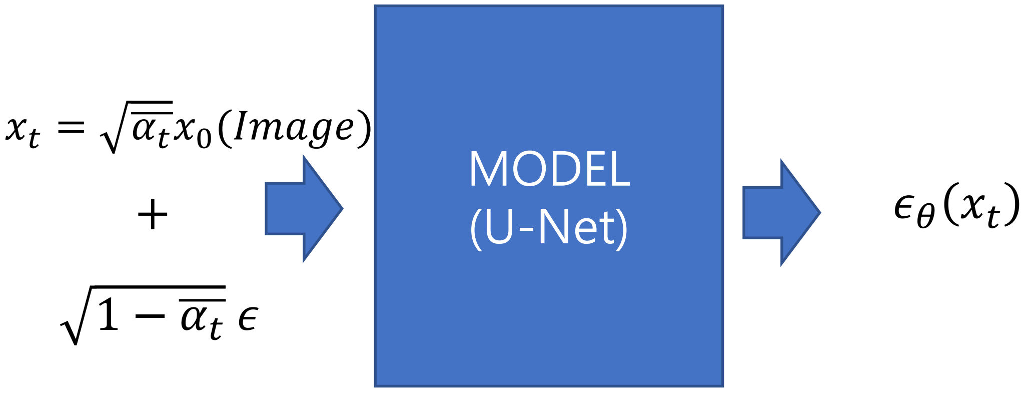
```
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ do
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,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

Simplified loss: $L_{\mathrm{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2 \right]$

Where ϵ_θ: function approximator intended to predict ϵ from x_t

Training:

$\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

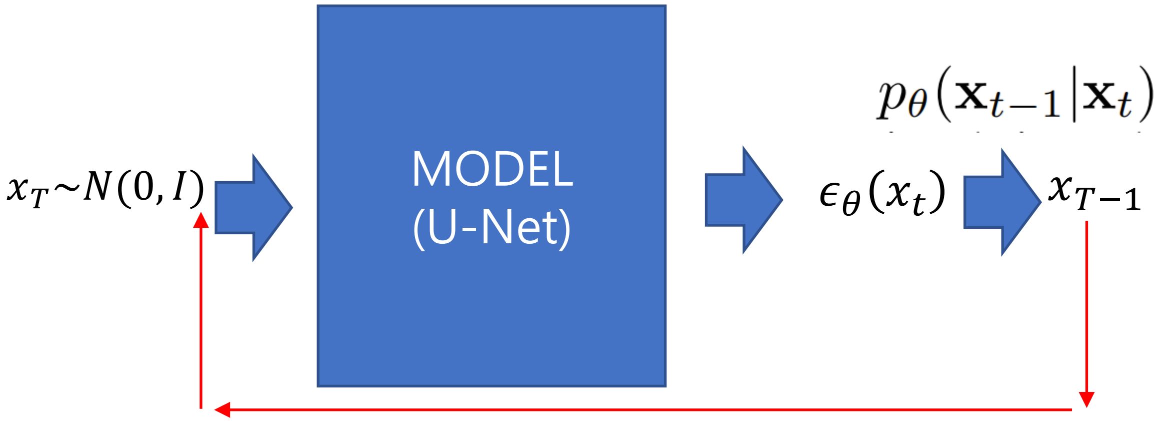


$x_t = \sqrt{\overline{\alpha_t}}x_0(Image)$
$+$
$\sqrt{1-\overline{\alpha_t}}\ \epsilon$
MODEL
(U-Net)
$\epsilon_\theta(x_t)$

Sampling t~ U[1,T]

Inference:

$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$



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

Method

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

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1};\boldsymbol{\mu}_\theta(\mathbf{x}_t,t),\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t))$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1};\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0),\beta_t\mathbf{I}),$$

➔ Parameters of Posterior are only depend on $\boldsymbol{\mu}$ & $\boldsymbol{\Sigma}$ $(\boldsymbol{\beta} \Rightarrow \boldsymbol{\Sigma})$ which is the parameters of the likelihood

$$\text{where}\quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \quad\text{and}\quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

1. **Forward process(L_1) : let $\boldsymbol{\beta}$ as hyperparameter ➔ Posterior (Enc in VAE) has no learnable parameters & L_T is a constant**

2. **Reverse process($L_{1:T-1}$): Reparameterize trick**

Posterior is not learnable parameter

Posterior $q(x_t|x_0) = N\big(x_t; \sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})I\big) \Rightarrow$ $\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right] + C \qquad \mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1};\boldsymbol{\mu}_\theta(\mathbf{x}_t,t),\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)) \text{ for } 1 < t \leq T \qquad \mathbf{x}_{t-1} = \tfrac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \tfrac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\right) + \sigma_t\mathbf{z}, \text{ where } \mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I}$$

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

➔ Estimate ϵ_θ instead of μ_θ

Experiments

1. Image Generation

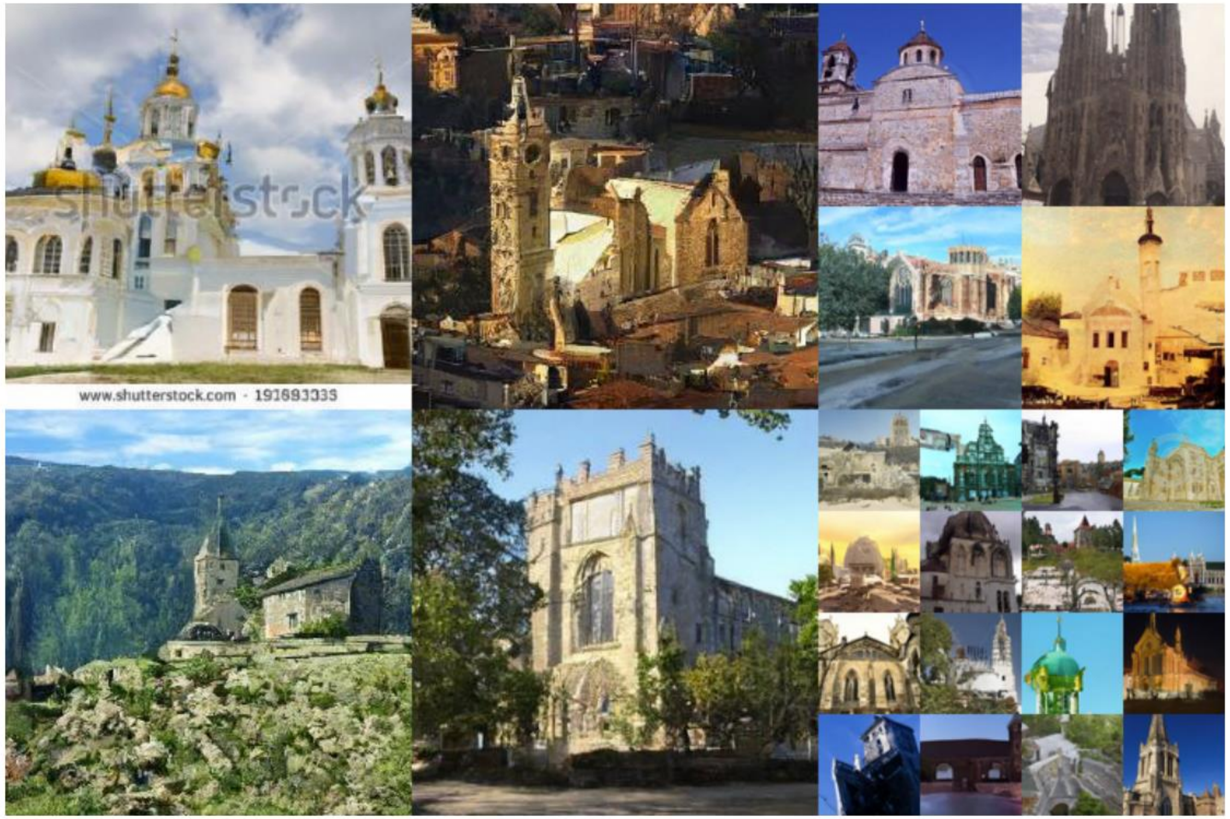
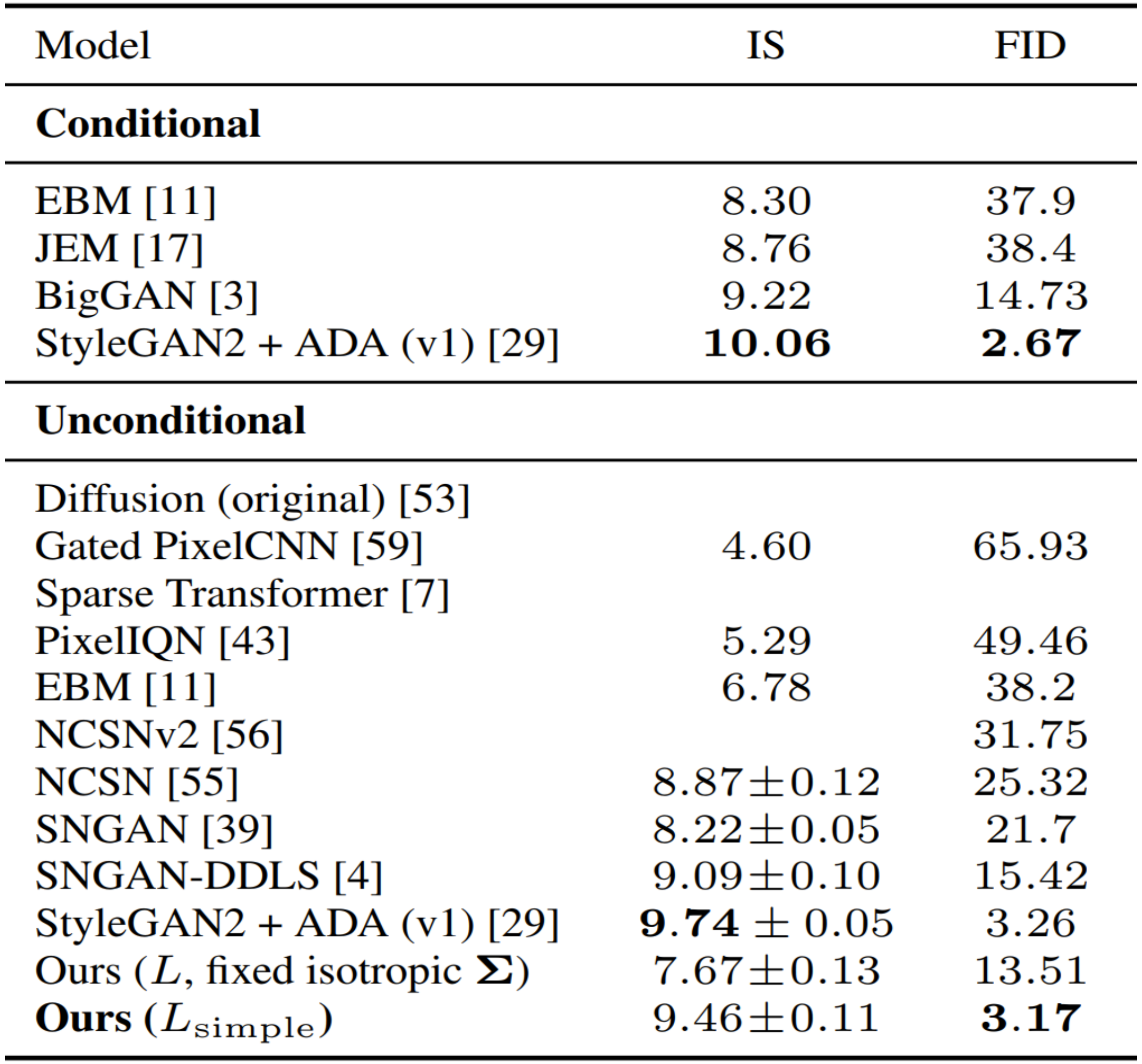


Figure 3: LSUN Church samples. FID=7.89

Dataset: Cifar 10

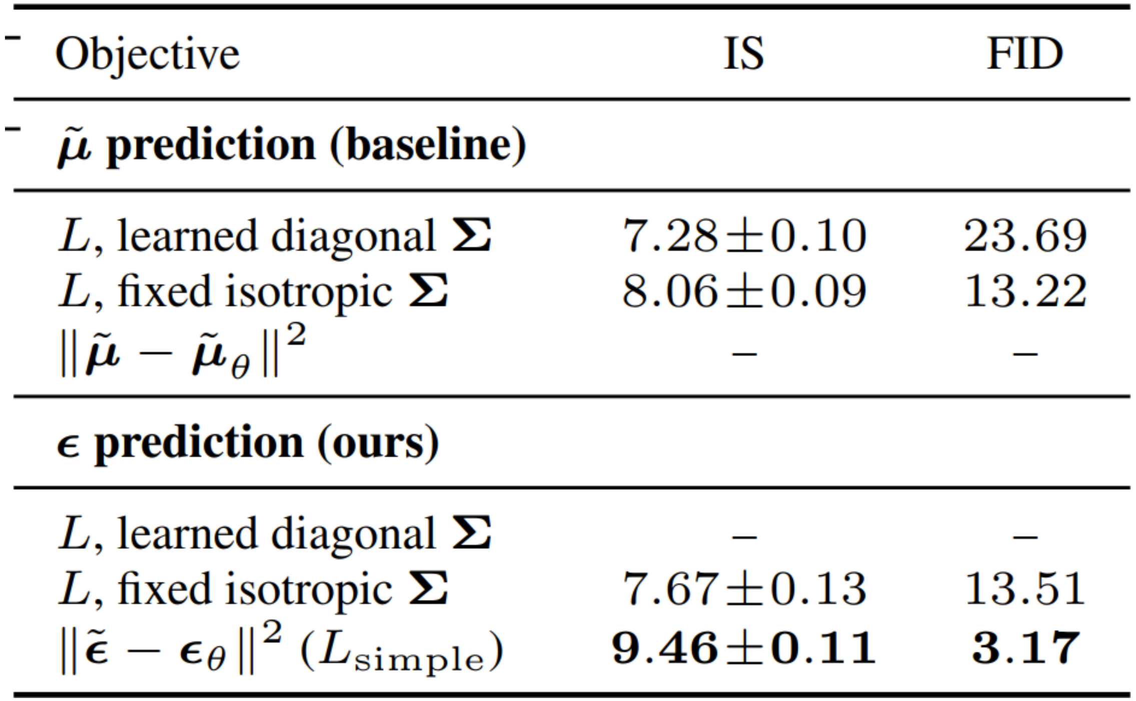


Model	IS	FID
Conditional		
EBM [11]	8.30	37.9
JEM [17]	8.76	38.4
BigGAN [3]	9.22	14.73
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**
Unconditional		
Diffusion (original) [53]		
Gated PixelCNN [59]	4.60	65.93
Sparse Transformer [7]		
PixelIQN [43]	5.29	49.46
EBM [11]	6.78	38.2
NCSNv2 [56]		31.75
NCSN [55]	8.87±0.12	25.32
SNGAN [39]	8.22±0.05	21.7
SNGAN-DDLS [4]	9.09±0.10	15.42
StyleGAN2 + ADA (v1) [29]	**9.74** ± 0.05	3.26
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67±0.13	13.51
Ours (L_{simple})	9.46±0.11	**3.17**

Experiments

2. Ablation study

- Training reverse process (Σ_θ)

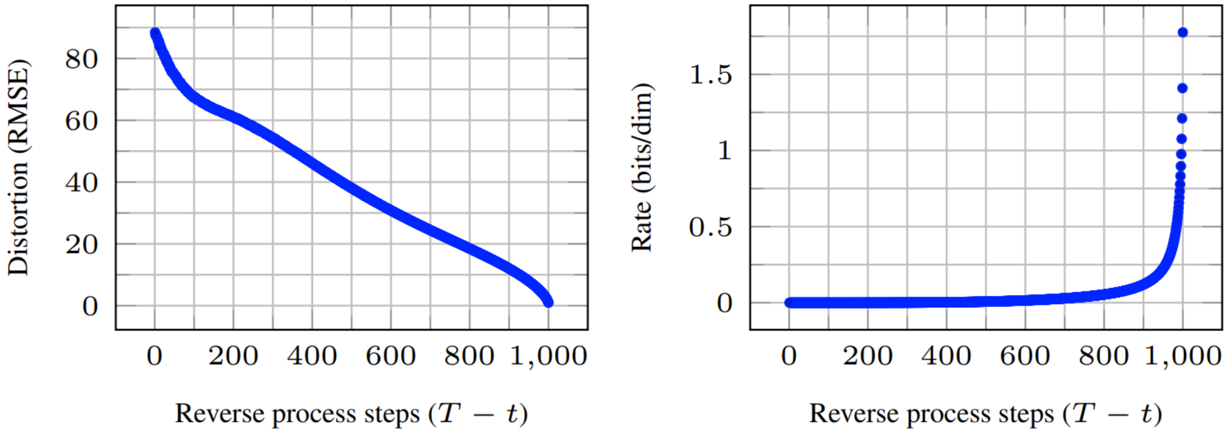


Objective	IS	FID
$\tilde{\boldsymbol{\mu}}$ prediction (baseline)		
L, learned diagonal $\boldsymbol{\Sigma}$	7.28 ± 0.10	23.69
L, fixed isotropic $\boldsymbol{\Sigma}$	8.06 ± 0.09	13.22
$\lVert \tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{\mu}}_\theta \rVert^2$	–	–
$\boldsymbol{\epsilon}$ prediction (ours)		
L, learned diagonal $\boldsymbol{\Sigma}$	–	–
L, fixed isotropic $\boldsymbol{\Sigma}$	7.67 ± 0.13	13.51
$\lVert \tilde{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}_\theta \rVert^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	$\mathbf{3.17}$

3. Progressive coding $D_{\mathrm{KL}}(q(\mathbf{x}) \parallel p(\mathbf{x}))$

Cal KL between forward and backward path

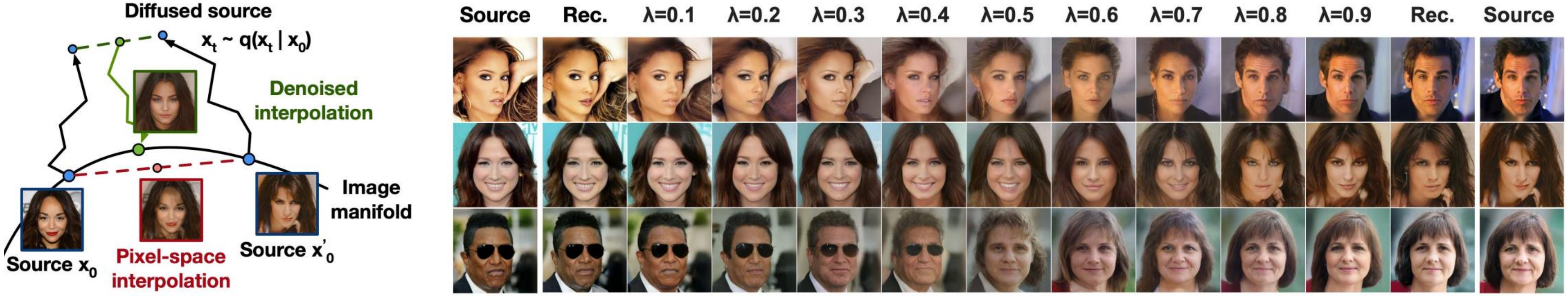
$$\sqrt{\lVert \mathbf{x}_0 - \hat{\mathbf{x}}_0 \rVert^2 / D}$$



Experiments

4 Interpolation

Encode two image $\mathbf{x}_t, \mathbf{x}'_t \sim q(x_t|x_0)$ & $\bar{x}_t = (1-\lambda)x_t + \lambda x'_t$



5. Progressive generation

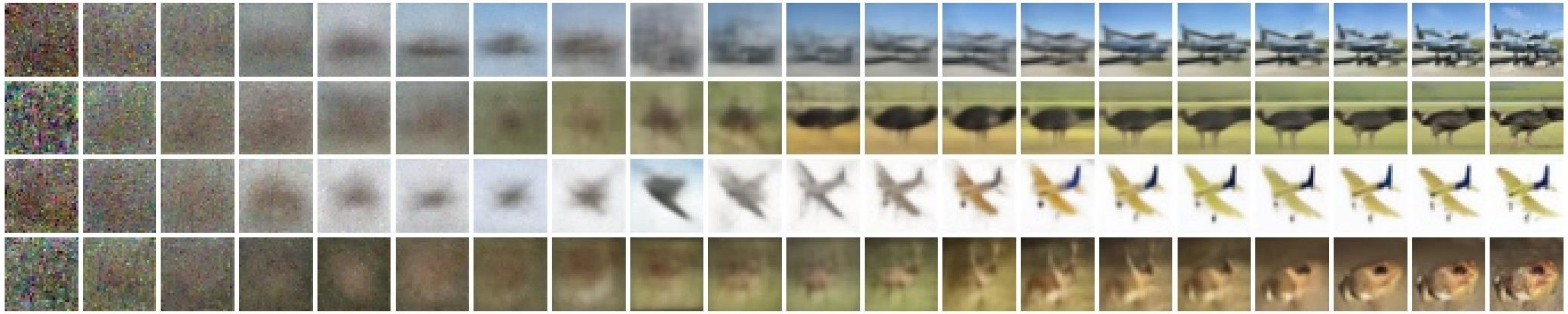


Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

Experiments

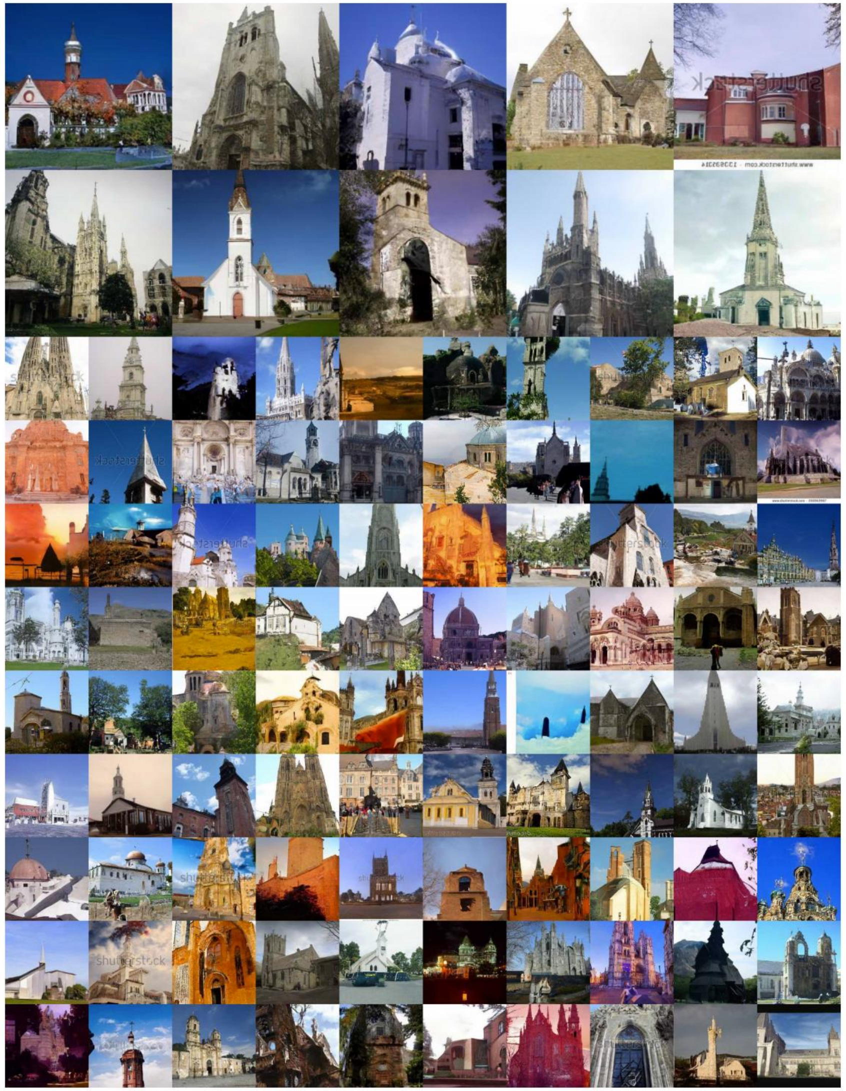


Figure 16: LSUN Church generated samples. FID=7.89

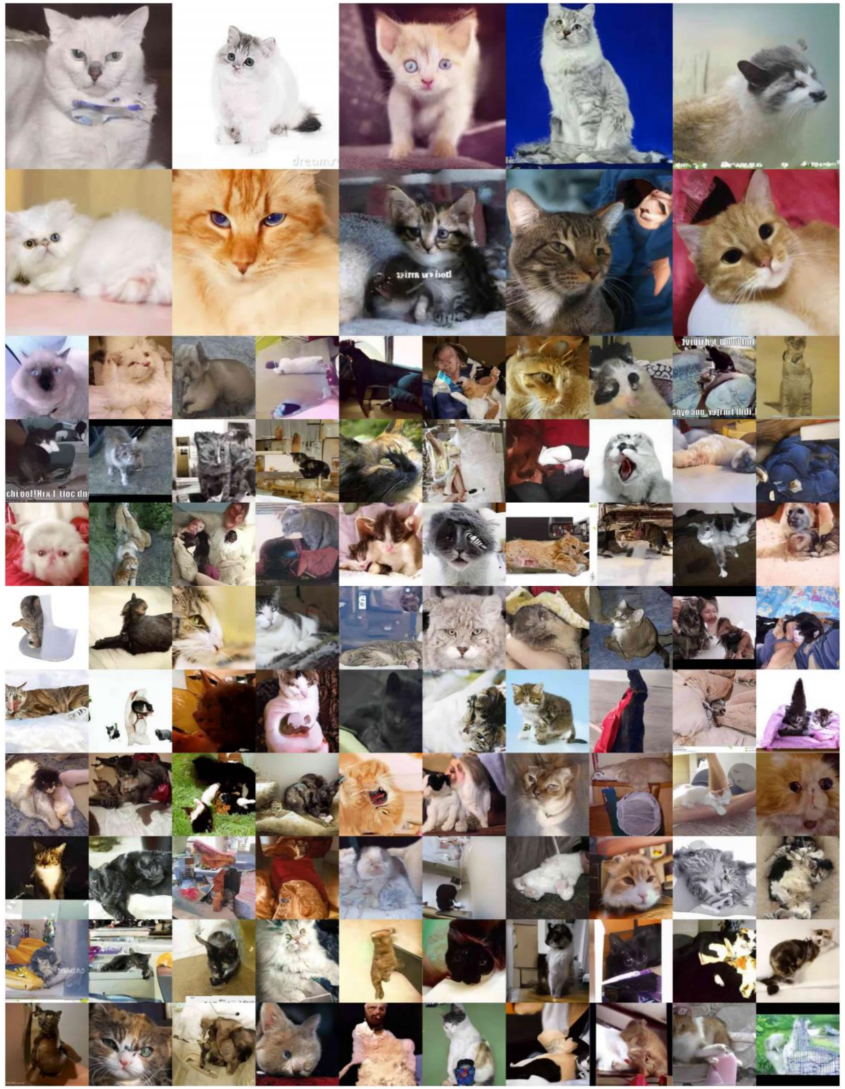


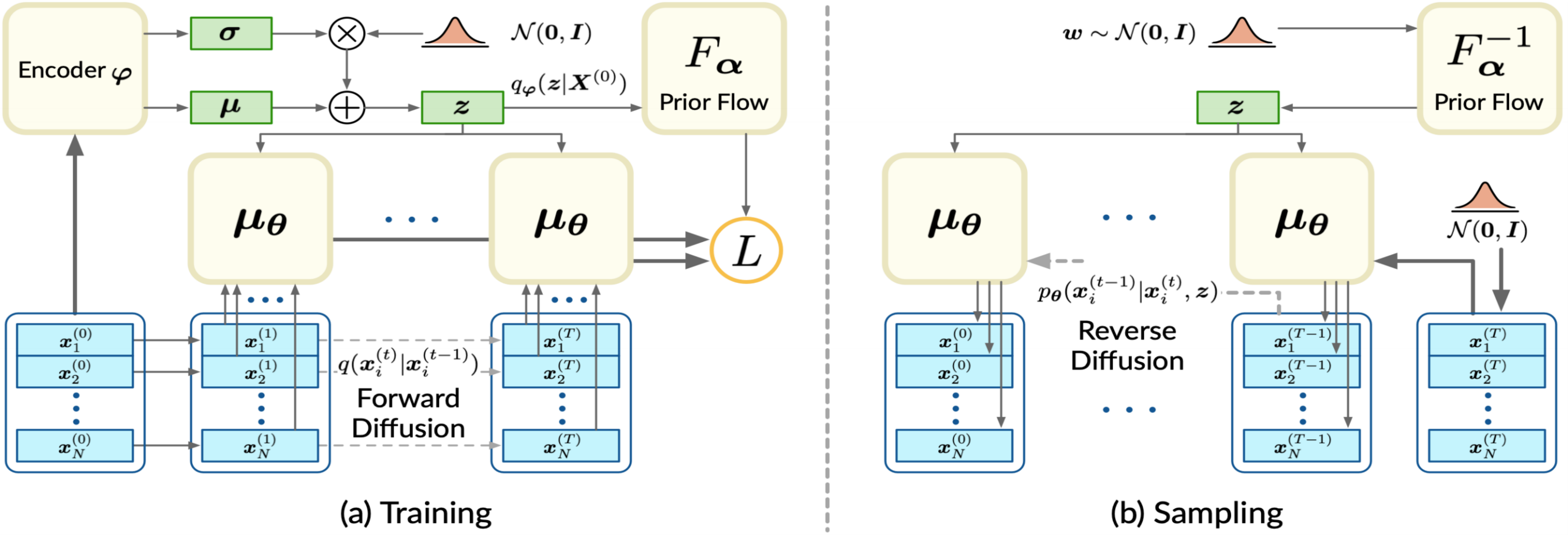
Figure 19: LSUN Cat generated samples. FID=19.75

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

Other works

There can be another model like here (Neural network estimate the $\boldsymbol{\mu_\theta}$)

DENOISING DIFFUSION IMPLICIT MODELS (ICLR'21)
DIFFWAVE: A VERSATILE DIFFUSION MODEL FOR AUDIO SYNTHESIS (ICLR'21 Oral)



Encoder φ
$\boldsymbol{\sigma}$
$\boldsymbol{\mu}$
$\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\boldsymbol{z}$
$q_\varphi(\boldsymbol{z}|\boldsymbol{X}^{(0)})$
$F_{\boldsymbol{\alpha}}$
Prior Flow
$\boldsymbol{\mu_\theta}$
$\boldsymbol{\mu_\theta}$
L
$\boldsymbol{x}_1^{(0)}$
$\boldsymbol{x}_2^{(0)}$
$\boldsymbol{x}_N^{(0)}$
$\boldsymbol{x}_1^{(1)}$
$\boldsymbol{x}_2^{(1)}$
$\boldsymbol{x}_N^{(1)}$
$q(\boldsymbol{x}_i^{(t)}|\boldsymbol{x}_i^{(t-1)})$
Forward Diffusion
$\boldsymbol{x}_1^{(T)}$
$\boldsymbol{x}_2^{(T)}$
$\boldsymbol{x}_N^{(T)}$
$\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$F_{\boldsymbol{\alpha}}^{-1}$
Prior Flow
$\boldsymbol{z}$
$\boldsymbol{\mu_\theta}$
$\boldsymbol{\mu_\theta}$
$\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$p_\theta(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})$
Reverse Diffusion
$\boldsymbol{x}_1^{(0)}$
$\boldsymbol{x}_2^{(0)}$
$\boldsymbol{x}_N^{(0)}$
$\boldsymbol{x}_1^{(T-1)}$
$\boldsymbol{x}_2^{(T-1)}$
$\boldsymbol{x}_N^{(T-1)}$
$\boldsymbol{x}_1^{(T)}$
$\boldsymbol{x}_2^{(T)}$
$\boldsymbol{x}_N^{(T)}$

(a) Training

(b) Sampling

Figure 3. The illustration of the proposed model. (a) illustrates how the objective is computed during the training process. (b) illustrates the generation process.

Thank you