Denoising Diffusion Probabilistic Models

拡散確率モデルによる高品質画像生成

Jonathan Ho, Ajay Jain, Pieter Abbeel
University of California, Berkeley
{jonathanho, ajayj, pabbeel}@berkeley.edu

要旨

私たちは、非平衡熱力学からの考察に触発された潜在変数モデルの一種である拡散確率モデルを使用した高品質な画像生成結果を提示します。最良の結果は、拡散確率モデルと Langevin ダイナミクスを用いたデノイジングスコアマッチングとの新たな関係に基づいて設計された重み付き変分境界を用いて学習することで得られました。また、私たちのモデルは、自己回帰デコーディングの一般化として解釈できる、段階的な損失圧縮スキームを自然に受け入れます。無条件の CIFAR10 データセットでは、Inception スコア 9.46 および最先端の FID スコア 3.17 を達成しました。256x256 の LSUN では、ProgressiveGAN に匹敵するサンプル品質を得ました。私たちの実装は https://github.com/hojonathanho/diffusion に記述しております。

1 Introduction

近年、あらゆる種類の深層生成モデルが、多種多様なデータモダリティにおいて高品質なサンプルを示しています。生成的敵対ネットワーク（GANs）、自己回帰モデル、フロー、変分オートエンコーダ（VAEs）は、印象的な画像や音声のサンプルを生成してきました [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]。さらに、エネルギーベースモデリングやスコアマッチングにおいても著しい進歩が見られ、GAN に匹敵する画像を生成することが可能となっています 本論文では、拡散確率モデル [?] における進展を紹介します。拡散確率モデル（以下、簡潔に「拡散モデル」と呼びます）は、有限時間後にデータと一致するサンプルを生成するために変分推論を用いて学習されたパラメータ化されたマルコフ連鎖です。この連鎖の遷移は、サンプリングの逆方向にデータに徐々にノイズを加えて信号を破壊する拡散プロセスを反転するように学習されます。拡散プロセスが少量のガウスノイズから構成される場合、サンプリング連鎖の遷移を条件付きガウス分布に設定するだけで十分であり、特に単純なニューラルネットワークによるパラメータ化が可能になります。

拡散モデルは定義が簡単であり、効率的に学習できますが、これまでのところ、高品質なサンプルを生成可能であることが実証されたことはありませんでした。本論文では、拡散モデルが実際に高品質なサンプルを生成可能であり、場合によっては他の種類の生成モデルの公開された結果よりも優れていることを示します（セクション 4）。さらに、特定のパラメータ化によって、拡散モデルがトレーニング中に複数のノイズレベルにわたるデノイジングスコアマッチングと同等であり、サンプリング時にはアニーリングされた Langevin ダイナミクスと同等であることを示します（セクション 3.2） [?, ?]。このパラメータ化を用いることで、最高のサンプル品質結果を得ることができたため（セクション 4.2）、これを本研究の主な貢献の一つと位置づけます。

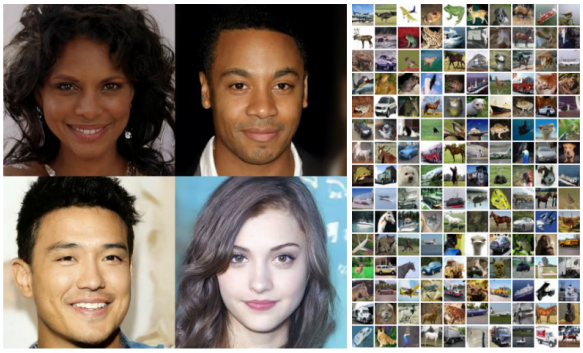
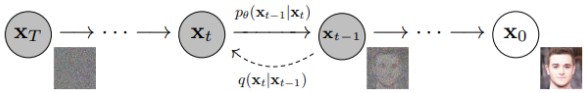


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)



サンプルの品質にもかかわらず、我々のモデルは、他の尤度ベースのモデルと比較して競争力のある対数尤度を持っていません（ただし、我々のモデルの対数尤度は、エネルギーベースモデルやスコアマッチングのアニーリング重要サンプリングで報告されている大まかな推定値よりも優れています [**?**, **?**]）。我々のモデルの無損失符号化長の大部分は、知覚できない画像の詳細を記述するために消費されていることが判明しました（セクション 4.3）。この現象を損失のある圧縮の観点でさらに洗練された分析を提示し、拡散モデルのサンプリング手法が通常の自己回帰モデルで可能なものを大幅に一般化したビット順序に沿った自己回帰デコーディングに類似した、段階的デコーディングの一種であることを示します。

2 Background

拡散モデル [**?**] は、以下の形式の潜在変数モデルです:

$$p_\theta(x_0) := \int p_\theta(x_{0:T})\, dx_{1:T},$$

ここで、$x_1, \dots, x_T$ はデータ $x_0 \sim q(x_0)$ と同じ次元を持つ潜在変数です。結合分布 $p_\theta(x_{0:T})$ は逆プロセス (reverse process) と呼ばれ、学習されたガウス遷移を持つマルコフ連鎖として定義されます。このプロセスは $p(x_T) = \mathcal{N}(x_T; 0, I)$ から始まります:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{1}$$

拡散モデルが他の潜在変数モデルと異なる点は、近似事後分布 $q(x_{1:T}|x_0)$ が固定されたマルコフ連鎖、つまり前向きプロセス (forward process) または拡散プロセス (diffusion process) と呼ばれることです。このプロセスは、分散スケジュール $\beta_1, \dots, \beta_T$ に従ってデータにガウスノイズを徐々に加えます:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{2}$$

トレーニングは、負の対数尤度に対する通常の変分境界を最適化することで行われます:

$$\mathbb{E}[-\log p_\theta(x_0)] \le \mathbb{E}_q \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] = \mathbb{E}_q \left[-\log p(x_T) - \sum_{t \ge 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] =: \mathcal{L}. \tag{3}$$

前向きプロセスの分散 β_t は、再パラメータ化 [**?**] によって学習するか、ハイパーパラメータとして固定することができます。また、逆プロセスの表現力は、$p_\theta(x_{t-1}|x_t)$ におけるガウス条件付き分布の選択によって部分的に保証されます。これは、β_t が小さい場合、前向きプロセスと逆プロセスが同じ関数形式を持つためです [**?**]。

前向きプロセスの顕著な特性として、任意の時刻 t における x_t を閉形式でサンプリングできることがあります。$\alpha_t := 1 - \beta_t$ と $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ の記法を用いると、以下が成り立ちます:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I). \tag{4}$$

効率的なトレーニングは、確率的勾配降下法 (stochastic gradient descent) を用いて $\mathcal{L}$ のランダムな項を最適化することで可能です。さらに、以下のように $\mathcal{L}$ (式 (3)) を書き換えることで、分散削減による改善が得られます:

$$\mathbb{E}_q \left[\underbrace{D_{\mathrm{KL}}(q(x_T|x_0) \parallel p(x_T))}_{\mathcal{L}_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))}_{\mathcal{L}_{t-1}} - \underbrace{\log p_\theta(x_0|x_1)}_{\mathcal{L}_0} \right] \tag{5}$$

(詳細については付録 A を参照してください。この式中の項に付けられたラベルは、セクション 3 で使用されます。)

式 (5) は、$p_\theta(x_{t-1}|x_t)$ と前向きプロセスの事後分布を直接比較するために、KL ダイバージェンスを使用します。この事後分布は、x_0 を条件とした場合に計算可能です:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I), \tag{6}$$

ここで、

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{7}$$

その結果、式 (5) におけるすべての KL ダイバージェンスはガウス分布間の比較となり、高分散なモンテカルロ推定を使用せずに、閉形式の表現を用いた Rao-Blackwell 化による計算が可能となります。

3 Diffusion Models and Denoising Autoencoders

拡散モデルは潜在変数モデルの制限されたクラスのように見えるかもしれませんが、実装には多数の自由度が存在します。前向きプロセスの分散 β_t、モデルのアーキテクチャ、逆プロセスのガウス分布のパラメータ化を選択する必要があります。これらの選択を導くために、拡散モデルとデノイジングスコアマッチングとの間に新しい明示的な関係を確立しました（セクション 3.2）。これにより、拡散モデルのための簡略化された重み付き変分境界の目的関数が導かれます（セクション 3.4）。最終的に、我々のモデル設計はシンプルさと経験的結果によって正当化されます（セクション 4）。我々の議論は式 (5) の項に基づいて分類されます。

3.1 Forward Process and $\mathcal{L}_T$

前向きプロセスの分散 β_t は再パラメータ化によって学習可能であるという事実を無視し、定数に固定します（詳細はセクション 4 を参照）。したがって、我々の実装では、近似事後分布 q に学習可能なパラメータは含まれません。このため、$\mathcal{L}_T$ はトレーニング中は定数であり、無視することができます。

3.2 Reverse Process and $\mathcal{L}_{1:T-1}$

次に、逆プロセス $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$ （$1 < t \leq T$）における選択について議論します。

まず、$\Sigma_\theta(x_t, t)$ を学習されない時間依存の定数 $\sigma_t^2 I$ に設定します。実験的には、$\sigma_t^2 = \beta_t$ と $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ の両方が類似した結果を示しました。最初の選択は $x_0 \sim \mathcal{N}(0, I)$ に最適であり、2 番目の選択は x_0 が 1 点に決定論的に設定される場合に最適です。これらは、座標ごとの単位分散を持つデータにおける逆プロセスエントロピーの上限と下限に対応する 2 つの極端な選択です [?]。

次に、平均 $\mu_\theta(x_t, t)$ を表現するために、$\mathcal{L}_{t-1}$ の次の解析に基づいた特定のパラメータ化を提案します。$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$ の場合、以下のように記述できます:

$$\mathcal{L}_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \right] + C. \tag{8}$$

ここで、C は θ に依存しない定数です。したがって、μ_θ の最も単純なパラメータ化は、前向きプロセスの事後平均 $\tilde{\mu}_t$ を予測するモデルであることがわかります。しかし、式 (8) をさらに展開するために、式 (4) を以下のように再パラメータ化します:

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

さらに、前向きプロセスの事後公式 (式 (7)) を適用します:

$$\mathcal{L}_{t-1} - C = \mathbb{E}_{x_0,\epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t \left(x_t(x_0, \epsilon), \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t(x_0, \epsilon) - \sqrt{1-\bar{\alpha}_t}\epsilon \right) \right) - \mu_\theta \left(x_t(x_0, \epsilon), t \right) \right\|^2 \right]. \tag{9}$$

さらに、これを以下のように書き換えることができます:

$$\mathcal{L}_{t-1} - C = \mathbb{E}_{x_0,\epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(x_t(x_0, \epsilon) - \beta_t\sqrt{1-\bar{\alpha}_t}\epsilon \right) - \mu_\theta \left(x_t(x_0, \epsilon), t \right) \right\|^2 \right]. \tag{10}$$

式 (10) から、μ_θ は x_t を入力として与えられたときに $\sqrt{\frac{1}{\alpha_t}} \left(x_t - \sqrt{\frac{\beta_t}{1-\bar{\alpha}_t}}\epsilon \right)$ を予測する必要があることが

Algorithm 1 Training

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

Algorithm 2 Sampling

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

わかります。x_t はモデルへの入力として利用可能であるため、以下のようなパラメータ化を選択できます:

$$\mu_\theta(x_t, t) = \tilde{\mu}_t \left(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t} \epsilon_\theta(x_t) \right) \right) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \beta_t \sqrt{1-\bar{\alpha}_t} \epsilon_\theta(x_t, t) \right), \tag{11}$$

ここで、ϵ_θ は x_t から ϵ を予測する関数近似器です。サンプリング $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$ を行うには、以下を計算します:

$$x_{t-1} = \sqrt{\frac{1}{\alpha_t}} \left(x_t - \sqrt{\frac{\beta_t}{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z,$$

ここで、$z \sim \mathcal{N}(0, I)$ です。完全なサンプリング手順（アルゴリズム 2）は、ϵ_θ をデータ密度の学習された勾配として用いる Langevin ダイナミクスに類似しています。

さらに、パラメータ化 (11) を用いると、式 (10) は次のように簡略化されます:

$$\mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_t}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, t \right) \right\|^2 \right]. \tag{12}$$

この式は、複数のノイズスケール t にまたがるデノイジングスコアマッチングに類似しています [**?**]。また、式 (12) が Langevin ライクな逆プロセス (11) の変分境界（の一項）に等しいことから、デノイジングスコアマッチングに類似した目的関数を最適化することは、Langevin ダイナミクスに類似したサンプリング連鎖の有限時間周辺分布を適合させるための変分推論を用いることと同等であるとわかります。

要約すると、逆プロセスの平均関数近似器 μ_θ を $\tilde{\mu}_t$ を予測するように訓練することもできますし、そのパラメータ化を変更して ϵ を予測するように訓練することも可能です（また、x_0 を予測することもできますが、初期の実験ではサンプル品質が悪化しました）。ϵ を予測するパラメータ化は、Langevin ダイナミクスに類似しており、拡散モデルの変分境界をデノイジングスコアマッチングに類似した目的関数に簡略化します。ただし、この方法は $p_\theta(x_{t-1}|x_t)$ の別のパラメータ化に過ぎないため、セクション 4 で ϵ を予測する方法と $\tilde{\mu}_t$ を予測する方法を比較したアブレーションでその有効性を検証します。

3.3 Data Scaling, Reverse Process Decoder, and $\mathcal{L}_0$

画像データは、$\{0, 1, \dots, 255\}$ の整数から $[-1, 1]$ に線形にスケーリングされていると仮定します。このスケーリングにより、ニューラルネットワークの逆プロセスが、標準正規事前分布 $p(x_T)$ から始まる一貫性のあるスケールの入力で動作することを保証します。離散的な対数尤度を得るために、逆プロセスの最後の項を、ガウス分布 $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$ から派生した独立した離散デコーダとして設定します:

$$p_\theta(x_0|x_1) = \prod_{i=1}^{D} \int_{\delta^+(x_0^i)}^{\delta^-(x_0^i)} \mathcal{N}(x; \mu_\theta^i(x_1, 1), \sigma_1^2) dx,$$

ここで、

$$\delta^+(x) = \begin{cases} \infty & \text{if } x = 1, \\ x + \frac{1}{255} & \text{if } x < 1, \end{cases} \quad \delta^-(x) = \begin{cases} -\infty & \text{if } x = -1, \\ x - \frac{1}{255} & \text{if } x > -1, \end{cases} \tag{13}$$

D はデータの次元であり、上付き添字 i は 1 つの座標を抽出することを示します。

（条件付き自己回帰モデルのようなより強力なデコーダを組み込むのは簡単ですが、それは将来の作業に委ねます。）VAE デコーダや自己回帰モデルで使用される離散化された連続分布 [**?**, **?**] と同様に、ここでの選択により、変分境界が離散データの無損失符号長となることが保証されます。この際、データにノイズを追加したり、スケーリング操作のヤコビアンを対数尤度に取り込む必要はありません。サンプリングの最後に、$\mu_\theta(x_1, 1)$ をノイズレスで表示します。

3.4 Simplified Training Objective

上述の逆プロセスとデコーダを定義したことで、式 (12) と (13) から導出された項を含む変分境界は、θ に関して明確に微分可能であり、トレーニングに利用する準備が整いました。

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
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]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
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	≤ 3.70 (3.69)
Ours (L_{simple})	9.46±0.11	**3.17**	≤ 3.75 (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

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

ただし、以下の変分境界の変種を用いてトレーニングを行うことが、サンプル品質の向上に有益であり、実装が簡単であることが判明しました:

$$\mathcal{L}_{\text{simple}}(\theta) := \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\|^2\right], \tag{14}$$

ここで、t は 1 から T の間で一様分布します。$t = 1$ の場合は、離散デコーダの定義 (式 (13)) における積分が、ガウス確率密度関数にビン幅を掛けた近似で置き換えられ、σ_1^2 と端効果を無視した $\mathcal{L}_0$ に対応します。$t > 1$ の場合は、式 (12) の非重み付きバージョンに対応し、NCSN デノイジングスコアマッチングモデル [**?**] で使用される損失重み付けと類似しています。(前向きプロセスの分散 β_t が固定されているため、$\mathcal{L}_T$ は登場しません。) アルゴリズム 1 に、この簡略化された目的を用いた完全なトレーニング手順を示します。

我々の簡略化された目的関数 (式 (14)) は、式 (12) における重み付けを破棄しているため、標準的な変分境界 [**?**, **?**] と比較して、再構成の異なる側面を強調する重み付き変分境界となっています。特に、セクション 4 で説明するように、我々の拡散プロセスの設定では、簡略化された目的関数が小さい t に対応する損失項の重みを減少させます。これらの項は、非常に少量のノイズを持つデータをデノイズするようネットワークを訓練するものであるため、重みを減少させることで、ネットワークがより困難なデノイズタスク (大きな t に対応する項) に集中できるようになります。我々の実験では、この再重み付けがより良いサンプル品質につながることが確認されます。

4 Experiments

全ての実験で $T = 1000$ と設定し、サンプリング中に必要なニューラルネットワーク評価の回数が以前の研究 [**?**, **?**] に一致するようにしました。前向きプロセスの分散は、$\beta_1 = 10^{-4}$ から $\beta_T = 0.02$ まで線形に増加する定数として設定しました。これらの定数は、データが $[-1, 1]$ にスケーリングされていることを考慮して小さく設定されており、逆プロセスと前向きプロセスがほぼ同じ関数形式を持ちながら、x_T における信号対雑音比を可能な限り小さく保つことを保証します (実験では、$\mathcal{L}_T = D_{\text{KL}}(q(x_T|x_0) \,\|\, \mathcal{N}(0, I)) \approx 10^{-5}$ ビット/次元です)。

逆プロセスを表現するために、グループ正規化 [**?**] を全体に適用した、マスクされていない PixelCNN++ [**?**, **?**] に類似した U-Net バックボーンを使用しました。パラメータは時間軸で共有され、Transformer のサイン波位置埋め込み [**?**] を使用してネットワークに時間を指定します。自己注意は 16×16 の特徴マップ解像度で使用しました [**?**, **?**]。詳細は付録 B を参照してください。

4.1 Sample Quality

表 1 は、CIFAR10 における Inception スコア、FID スコア、および負の対数尤度 (無損失符号長) を示しています。我々の FID スコアは 3.17 であり、無条件モデルとしては文献中のほとんどのモデル (クラス条件付きモデルを含む) よりも優れたサンプル品質を達成しました。FID スコアは標準的な慣例に従い、トレーニングセットに対して計算しました。テストセットに対して計算するとスコアは 5.24 となり、これは文献中の多くのトレーニングセット FID スコアよりも依然として優れています。

モデルを真の変分境界でトレーニングすると、簡略化された目的関数でトレーニングする場合と比べてより良い符号長が得られることが期待されますが、後者の方法は最良のサンプル品質をもたらします。CIFAR10 と CelebA-HQ (256×256) のサンプルは図 1 を、LSUN (256×256) のサンプルは図 3 および図 4 を参照してください [**?**]。さらに詳細は付録 D を参照してください。

4.2 Reverse Process Parameterization and Training Objective Ablation

表 2 は、逆プロセスのパラメータ化およびトレーニング目標 (セクション 3.2) のサンプル品質への影響を示しています。基準として、$\tilde{\mu}$ を予測する方法は、式 (14) に類似した非重み付き平均二乗誤差のような簡

Figure 3: LSUN Church samples. FID=7.89

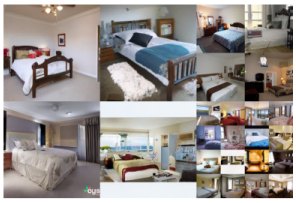


Figure 4: LSUN Bedroom samples. FID=4.90

Algorithm 3 Sending $\mathbf{x}_0$

1: Send $\mathbf{x}_T \sim q(\mathbf{x}_T|\mathbf{x}_0)$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 2, 1$ **do**
3: Send $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: Send $\mathbf{x}_0$ using $p_\theta(\mathbf{x}_0|\mathbf{x}_1)$

Algorithm 4 Receiving

1: Receive $\mathbf{x}_T$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 1, 0$ **do**
3: Receive $\mathbf{x}_t$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: **return** $\mathbf{x}_0$

略化された目的関数ではなく、真の変分境界でトレーニングされた場合にのみ良好に機能することがわかりました。また、変分境界にパラメータ化された対角行列 $\Sigma_\theta(x_t)$ を組み込むことで逆プロセスの分散を学習すると、不安定なトレーニングとサンプル品質の低下を招くことがわかりました。

提案したように ϵ を予測する方法は、固定された分散を用いた変分境界でトレーニングされた場合、$\tilde{\mu}$ を予測する方法とほぼ同等の性能を示しましたが、簡略化された目的関数でトレーニングされた場合はるかに良好な結果を示しました。

4.3 Progressive Coding

表 1 は、我々の CIFAR10 モデルの符号長も示しています。トレーニングセットとテストセット間のギャップは 1 次元あたり最大で 0.03 ビットであり、他の尤度ベースのモデルで報告されているギャップと比較可能であり、拡散モデルが過学習していないことを示しています（最近傍可視化については付録 D を参照してください）。それでも、我々の無損失符号長は、アニーリング重要サンプリングを用いたエネルギーベースモデルやスコアマッチングで報告された大まかな推定値よりも優れていますが、他の種類の尤度ベース生成モデルとは競争できません [**?**]。

それにもかかわらず、我々のサンプルが高品質であることを考えると、拡散モデルには損失圧縮に優れた誘導バイアスがあると結論づけます。変分境界項 $\mathcal{L}_1 + \cdots + \mathcal{L}_T$ をレート、$\mathcal{L}_0$ を歪みと見なすと、最高品質のサンプルを持つ CIFAR10 モデルは、1 次元あたり 1.78 ビットのレートと 1.97 ビットの歪みを持ち、0 から 255 のスケールで 0.95 の二乗平均平方誤差に相当します。無損失符号長の半分以上が知覚できない歪みを記述しています。

Progressive Lossy Compression

本モデルのレート-歪み挙動をさらに詳しく調べるために、式 (5) の形式を反映した段階的損失圧縮コードを導入します。アルゴリズム 3 およびアルゴリズム 4 を参照してください。これらは、最小ランダムコーディング [**?**, **?**] のような手法にアクセス可能であることを前提としています。この手法は、受信者が事前に分布 p のみを利用可能である場合、任意の分布 p および q に対して、$x \sim q(x)$ を平均的に $D_{\mathrm{KL}}(q(x) \parallel p(x))$ ビットで送信することが可能です。

$x_0 \sim q(x_0)$ に適用すると、アルゴリズム 3 および 4 は、$x_T, \ldots, x_0$ を順次送信し、合計の期待符号長は式 (5) に等しくなります。受信者は任意の時刻 t で、部分的な情報 x_t を完全に利用可能であり、次のように x_0 を段階的に推定できます:

$$x_0 \approx \hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t)}{\sqrt{\bar{\alpha}_t}}, \tag{15}$$

これは式 (4) に基づいています。（確率的再構成 $x_0 \sim p_\theta(x_0|x_t)$ も有効ですが、歪みの評価がより困難になるため、ここでは考慮しません。）

図 5 は、CIFAR10 のテストセットにおけるレート-歪みプロットを示しています。各時刻 t で、歪みは次のようにして計算されます:

$$\text{歪み} = \sqrt{\frac{\|x_0 - \hat{x}_0\|^2}{D}},$$

また、レートは時刻 t までに受信されたビット数の累積として計算されます。このプロットの低レート領域では歪みが急激に減少しており、大部分のビットが知覚できない歪みに割り当てられていることを示しています。

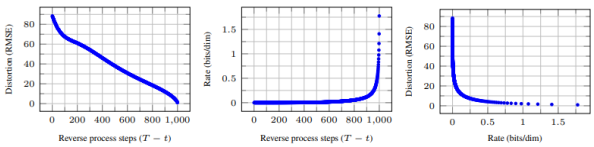


Figure 5: Unconditional CIFAR10 test set rate-distortion vs. time. Distortion is measured in root mean squared error on a $[0, 255]$ scale. See Table 4 for details.

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).

Figure 7: When conditioned on the same latent, CelebA-HQ 256×256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Progressive Generation

ランダムビットからの段階的な伸張（decompression）によって与えられる段階的な無条件生成プロセスも実行しました。言い換えると、アルゴリズム 2 を使用して逆プロセスをサンプリングする際に、逆プロセスの結果 $\hat{x}_0$ を予測します。図 6 および図 10 は、逆プロセスの進行に伴う $\hat{x}_0$ のサンプル品質を示しています。大規模な画像特徴は最初に現れ、細部は最後に現れます。

図 7 は、さまざまな t に対して x_t を固定したままの確率的予測 $x_0 \sim p_\theta(x_0|x_t)$ を示しています。t が小さい場合、細部を除くすべての特徴が保持されます。一方で、t が大きい場合、大規模な特徴のみが保持されます。これらは、おそらく概念的な圧縮（conceptual compression）[?] を示唆しているのかもしれません。

Connection to Autoregressive Decoding

変分境界 (式 (5)) は次のように書き換えることができます:

$$\mathcal{L} = D_{\mathrm{KL}}(q(x_T) \parallel p(x_T)) + \mathbb{E}_q \left[\sum_{t \geq 1} D_{\mathrm{KL}}(q(x_{t-1}|x_t) \parallel p_\theta(x_{t-1}|x_t)) \right] + H(x_0), \tag{16}$$

(導出については付録 A を参照してください。)

次に、拡散プロセスの長さ T をデータの次元に設定し、以下のようにプロセスを構成することを考えます: - 前向きプロセスを、$q(x_t|x_0)$ が x_0 の最初の t 座標をマスクしたものに全ての確率質量を集中させるように定義する（すなわち、$q(x_t|x_{t-1})$ が t 番目の座標をマスクする）。- $p(x_T)$ を空白画像に全ての確率質量を集中させるものとして設定する。- 議論のために、$p_\theta(x_{t-1}|x_t)$ を完全に表現力のある条件付き分布と仮定する。

これらの選択を行うと、$D_{\mathrm{KL}}(q(x_T) \parallel p(x_T)) = 0$ となり、$D_{\mathrm{KL}}(q(x_{t-1}|x_t) \parallel p_\theta(x_{t-1}|x_t))$ を最小化することは、p_θ を訓練して座標 $t+1, \ldots, T$ をそのままコピーし、座標 t を座標 $t+1, \ldots, T$ に基づいて予測することを意味します。この特定の拡散プロセスを用いて p_θ を訓練することは、自己回帰モデルを訓練することと同等です。

したがって、ガウス拡散モデル（式 (2)）を、データ座標の並べ替えでは表現できない一般化されたビット順序を持つ自己回帰モデルの一種として解釈できます。先行研究では、このような並べ替えがサンプル品質に影響を与える誘導バイアスを導入することが示されています [?]。したがって、ガウス拡散も同様の目的を果たしていると推測されます。特に、ガウスノイズはマスクノイズと比較して画像に追加するのがより自然である可能性があります。

さらに、ガウス拡散の長さはデータ次元と等しい必要はありません。例えば、我々の実験では $T = 1000$ を使用していますが、これは $32 \times 32 \times 3$ や $256 \times 256 \times 3$ の画像の次元よりも小さいです。ガウス拡散は、高速サンプリングのために短縮したり、モデルの表現力を高めるために長くすることができます。

Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

4.4 Interpolation

潜在空間で、ソース画像 $x_0, x_0' \sim q(x_0)$ を補間することができます。具体的には、q を確率的エンコーダとして使用し、$x_t, x_t' \sim q(x_t|x_0)$ を取得し、線形補間された潜在ベクトル $\bar{x}_t = (1-\lambda)x_t + \lambda x_t'$ を逆プロセスによって画像空間にデコードします: $\bar{x}_0 \sim p(x_0|\bar{x}_t)$。これは、線形補間された汚染画像のアーティファクトを逆プロセスで除去する効果があり、図 8（左）に示されています。

異なる値の λ に対してノイズを固定することで、x_t と x_t' を同じ状態に保ちました。図 8（右）は、CelebA-HQ（256×256）画像（$t = 500$）の補間と再構成を示しています。逆プロセスは、高品質な再構成と、姿勢、肌の色、髪型、表情、背景などの属性が滑らかに変化する妥当な補間を生成しますが、眼鏡の有無などは補間されません。t が大きい場合、補間は粗くなり、より多様な結果となります。特に、$t = 1000$ では新しいサンプルが生成されます（付録図 9 を参照）。

5 Related Work

拡散モデルは、フロー [?, ?, ?, ?, ?, ?, ?] や VAE [?, ?, ?] に似ているように見えるかもしれませんが、q がパラメータを持たず、最上位の潜在変数 x_T がデータ x_0 とほぼゼロの相互情報を持つように設計されています。我々の ϵ 予測型逆プロセスのパラメータ化は、拡散モデルと、複数のノイズレベルにわたるデノイジングスコアマッチングおよびサンプリングのためのアニーリングされた Langevin ダイナミクスとの関連を確立します [?, ?]。

一方で、拡散モデルは簡単な対数尤度評価を許容し、トレーニング手順では変分推論を用いて Langevin ダイナミクスサンプラーを明示的に訓練します（詳細は付録 C を参照）。この関連は逆にも成り立ち、特定の重み付け形式のデノイジングスコアマッチングが、Langevin ライクなサンプラーを訓練するための変分推論と同じであることを示しています。

マルコフ連鎖の遷移演算子を学習する他の方法として、infusion training [?]、variational walkback [?]、生成確率ネットワーク [?] などがあります。また、その他の手法についても研究が進められています [?, ?, ?, ?, ?, ?]。

スコアマッチングとエネルギーベースモデリングの既知の関連を通じて、本研究はエネルギーベースモデルに関する他の最近の研究に示唆を与える可能性があります [?, ?, ?, ?, ?, ?, ?, ?, ?, ?]。我々のレート-歪み曲線は、変分境界の 1 回の評価で時間にわたって計算され、アニーリングされた重要サンプリングで歪みペナルティに基づいて曲線を計算する方法を思い起こさせます [?]。

また、我々の段階的デコーディングの議論は、畳み込み DRAW モデルおよび関連するモデル [?, ?] にも見られ、自己回帰モデルのサブスケール順序やサンプリング戦略のより一般的な設計にもつながる可能性があります [?, ?]。

6 Conclusion

我々は、拡散モデルを用いて高品質な画像サンプルを提示し、拡散モデルとマルコフ連鎖のトレーニングにおける変分推論、デノイジングスコアマッチング、アニーリングされた Langevin ダイナミクス（および拡張としてエネルギーベースモデル）、自己回帰モデル、段階的損失圧縮との関連性を見出しました。拡散モデルは画像データに対して優れた誘導バイアスを持つように見えるため、他のデータモダリティや他の種類の生成モデルおよび機械学習システムのコンポーネントとしての有用性を調査することを楽しみにしています。

Broader Impact

本研究における拡散モデルの研究範囲は、GAN、フロー、自己回帰モデルなどのサンプル品質を向上させる試みといった、他の種類の深層生成モデルに関する既存の研究と類似しています。我々の論文は、この手法群の中で拡散モデルを一般的に有用なツールにする進展を表しており、生成モデルがこれまで（そして今後）世界に与える影響を増幅させる可能性があります。

残念ながら、生成モデルの悪意のある使用については、多くのよく知られた問題があります。サンプル生成技術は、高名な人物の偽の画像や動画を政治目的で作成するために利用されることがあります。ソフ

トウェアツールが利用可能になる以前から、偽画像は手動で作成されていましたが、本研究のような生成モデルはそのプロセスを容易にします。幸いにも、現在の CNN 生成画像には微妙な欠陥があり、これにより検出が可能ですが [?]、生成モデルの改善により検出が困難になる可能性があります。また、生成モデルはトレーニングに使用されたデータセットのバイアスを反映します。多くの大規模データセットはインターネットから自動収集されるため、特に画像がラベル付けされていない場合、これらのバイアスを除去することは困難です。これらのデータセットでトレーニングされた生成モデルのサンプルがインターネット上に拡散すれば、これらのバイアスがさらに強化される可能性があります。

一方で、拡散モデルはデータ圧縮に役立つ可能性があります。データが高解像度化し、世界のインターネットトラフィックが増加する中で、インターネットの幅広い利用を確保するためにはデータ圧縮が重要になるかもしれません。我々の研究は、画像分類から強化学習に至る幅広い下流タスクのために、ラベル付けされていない生データの表現学習に貢献する可能性があります。また、拡散モデルは芸術、写真、音楽などの創造的な用途においても実用的になるかもしれません。

Acknowledgments and Disclosure of Funding

本研究は、ONR PECASE および NSF Graduate Research Fellowship（助成金番号 DGE-1752814）の支援を受けています。また、Google の TensorFlow Research Cloud（TFRC）より Cloud TPU を提供していただきました。

References

1. Guillaume Alain, Yoshua Bengio, Li Yao, Jason Yosinski, Eric Thibodeau-Laufer, Saizheng Zhang, and Pascal Vincent. GSNs: generative stochastic networks. *Information and Inference: A Journal of the IMA*, 5(2):210–249, 2016.

2. Florian Bordes, Sina Honari, and Pascal Vincent. Learning to generate samples from noise through infusion training. In *International Conference on Learning Representations*, 2017.

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.

4. Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

5. Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.

6. Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 863–871, 2018.

7. Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

8. Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc ' Aurelio Ranzato. Residual energy-based models for text generation. *arXiv preprint arXiv:2004.11714*, 2020.

9. Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

10. Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

11. Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, pages 3603–3613, 2019.

12. Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative ConvNets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9155–9164, 2018.

13. Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7518–7528, 2020.

14. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

15. Anirudh Goyal, Nan Rosemary Ke, Surya Ganguli, and Yoshua Bengio. Variational walkback: Learning a transition operator as a stochastic recurrent net. In *Advances in Neural Information Processing Systems*, pages 4392–4402, 2017.

16. Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

17. Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

18. Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pages 3549–3557, 2016.

19. Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC' 07)*, pages 10–23. IEEE, 2007.

20. Marton Havasi, Robert Peharz, and José Miguel Hernández-Lobato. Minimal random code learning: Getting bits back from compressed model parameters. In *International Conference on Learning Representations*, 2019.

21. 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.

22. Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676v1*, 2020.

23. Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

24. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

25. Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

26. Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

27. Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

28. John Lawson, George Tucker, Bo Dai, and Rajesh Ranganath. Energy-inspired models: Learning with sampler-induced distributions. In *Advances in Neural Information Processing Systems*, pages 8501–8513, 2019.

29. Daniel Levy, Matt D. Hoffman, and Jascha Sohl-Dickstein. Generalizing Hamiltonian Monte Carlo with neural networks. In *International Conference on Learning Representations*, 2018.

30. Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *Advances in Neural Information Processing Systems*, pages 6548–6558, 2019.

31. Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *International Conference on Learning Representations*, 2019.

32. Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

33. Alex Nichol. VQ-DRAW: A sequential discrete VAE. *arXiv preprint arXiv:2003.01599*, 2020.

34. Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370*, 2019.

35. Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems*, pages 5233–5243, 2019.

36. Georg Ostrovski, Will Dabney, and Remi Munos. Autoregressive quantile networks for generative modeling. In *International Conference on Machine Learning*, pages 3936–3945, 2018.

37. Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

38. Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.

39. Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

40. Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

41. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

42. Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

43. Tim Salimans, Diederik Kingma, and Max Welling. Markov Chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

44. Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

45. Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

46. Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

47. Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-NICE-MC: Adversarial training for MCMC. In *Advances in Neural Information Processing Systems*, pages 5140–5150, 2017.

48. Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

49. Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

50. Aaron 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. *arXiv preprint arXiv:1609.03499*, 2016.

51. Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016.

52. Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

53. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

54. Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

55. Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot...for now. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

56. Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

57. Auke J Wiggers and Emiel Hoogeboom. Predictive sampling with forecasting autoregressive models. *arXiv preprint arXiv:2002.09928*, 2020.

58. Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *arXiv preprint arXiv:2002.06707*, 2020.

59. Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

60. Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644, 2016.

61. Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7093–7101, 2017.

62. Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor networks for 3D shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8629–8638, 2018.

63. Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning energy-based spatial-temporal generative convnets for dynamic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

64. Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

65. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Table 3: FID scores for LSUN 256 × 256 datasets

Model	LSUN Bedroom	LSUN Church	LSUN Cat
ProgressiveGAN [27]	8.34	6.42	37.52
StyleGAN [28]	**2.65**	4.21*	8.53*
StyleGAN2 [30]	-	**3.86**	**6.93**
Ours (L_{simple})	6.36	7.89	19.75
Ours (L_{simple}, large)	4.90	-	-

Table 4: Unconditional CIFAR10 test set rate-distortion values (accompanies Fig. 5)

Reverse process time ($T-t+1$)	Rate (bits/dim)	Distortion (RMSE $[0, 255]$)
1000	1.77581	0.95136
900	0.11994	12.02277
800	0.05415	18.47482
700	0.02866	24.43656
600	0.01507	30.80948
500	0.00716	38.03236
400	0.00282	46.12765
300	0.00081	54.18826
200	0.00013	60.97170
100	0.00000	67.60125

Extra Information

LSUN データセットの FID スコアは表 3 に示されています。StyleGAN2 がベースラインとして報告したスコアには * が付されています。他のスコアは、それぞれの著者によって報告されたものです。

Progressive Compression

セクション 4.3 における損失圧縮の議論は、概念実証に過ぎません。なぜなら、アルゴリズム 3 および 4 は、高次元データに対しては実行が困難な最小ランダムコーディング [?] のような手法に依存しているからです。これらのアルゴリズムは、圧縮に対する解釈を提供するものとして機能します。

A Extended Derivations

以下は、拡散モデルにおける式 (5) の導出であり、分散を削減した変分境界を示しています。この内容は Sohl-Dickstein ら [?] の研究から引用されたものであり、完全性を期すためにここに記載します。

$$\mathcal{L} = \mathbb{E}_q \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] \tag{17}$$

$$= \mathbb{E}_q \left[-\log p(x_T) - \sum_{t \geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] \tag{18}$$

$$= \mathbb{E}_q \left[-\log p(x_T) - \sum_{t > 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)} \right] \tag{19}$$

$$= \mathbb{E}_q \left[-\log p(x_T) - \sum_{t > 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \cdot \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} - \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)} \right] \tag{20}$$

$$= \mathbb{E}_q \left[\mathrm{DKL}(q(x_T|x_0) \parallel p(x_T)) + \sum_{t > 1} \mathrm{DKL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1) \right] \tag{22}$$

代替的な形式　以下に、$\mathcal{L}$ の代替形式を示します。この形式は直接的に推定することは困難ですが、セクション 4.3 での議論に役立ちます。

$$\mathcal{L} = \mathbb{E}_q \left[-\log p(x_T) - \sum_{t \geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] \tag{23}$$

$$= \mathbb{E}_q \left[-\log p(x_T) - \sum_{t \geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t)} \cdot \frac{q(x_{t-1})}{q(x_t)} \right] \tag{24}$$

$$= \mathrm{DKL}(q(x_T) \parallel p(x_T)) + \mathbb{E}_q \left[\sum_{t \geq 1} \mathrm{DKL}(q(x_{t-1}|x_t) \parallel p_\theta(x_{t-1}|x_t)) \right] + H(x_0) \tag{26}$$

B Experimental Details

我々のニューラルネットワークアーキテクチャは、PixelCNN++ [**?**] のバックボーンに従っており、Wide ResNet [**?**] に基づく U-Net [**?**] です。実装を簡略化するために、重み正規化 [**?**] をグループ正規化 [**?**] に置き換えました。32 × 32 モデルは 4 つの特徴マップ解像度（32 × 32 から 4 × 4）を使用し、256 × 256 モデルは 6 つの解像度を使用します。全てのモデルには、解像度レベルごとに 2 つの畳み込み残差ブロックがあり、畳み込みブロック間の 16 × 16 解像度に自己注意ブロックを配置しています [**?**]。拡散時間 t は、各残差ブロックに Transformer のサイン波位置埋め込み [**?**] を加えることで指定します。我々の CIFAR10 モデルには 3570 万パラメータがあり、LSUN および CelebA-HQ モデルには 1 億 1400 万パラメータがあります。また、LSUN Bedroom モデルの大規模なバリアントは、フィルタ数を増やして約 2 億 5600 万パラメータに設定しました。

すべての実験において、TPU v3-8（8 V100 GPU に相当）を使用しました。CIFAR モデルは、バッチサイズ 128 で 1 秒あたり 21 ステップの速度でトレーニングされ、800k ステップのトレーニング完了には 10.6 時間を要しました。256 枚の画像バッチをサンプリングするには 17 秒かかります。CelebA-HQ および LSUN（256 × 256）モデルは、バッチサイズ 64 で 1 秒あたり 2.2 ステップの速度でトレーニングされ、128 枚の画像バッチをサンプリングするには 300 秒かかります。我々は CelebA-HQ を 50 万ステップ、LSUN Bedroom を 240 万ステップ、LSUN Cat を 180 万ステップ、LSUN Church を 120 万ステップでトレーニングしました。大規模な LSUN Bedroom モデルは 115 万ステップでトレーニングされました。

初期段階でネットワークサイズをメモリ制約に収めるためにハイパーパラメータを選定した以外は、大部分のハイパーパラメータ検索を CIFAR10 のサンプル品質の最適化に費やし、その設定を他のデータセットに転送しました。

- β_t スケジュールは、定数、線形、および二次スケジュールのセットから選択しましたが、すべて $L_T \approx 0$ となるように制約しました。$T = 1000$ と設定し、線形スケジュールを選択しました（$\beta_1 = 10^{-4}$ から $\beta_T = 0.02$ まで）。

- CIFAR10 のドロップアウト率を 0.1 に設定しました（0.1, 0.2, 0.3, 0.4 の値を調査）。CIFAR10 でドロップアウトを使用しない場合、正則化されていない PixelCNN++ [**?**] における過学習アーティファクトに似た品質低下が見られました。他のデータセットではドロップアウト率を 0 に設定しましたが、調査は行いませんでした。

- CIFAR10 のトレーニング時にはランダム水平反転を使用しました。反転を使用する場合と使用しない場合の両方を試しましたが、サンプル品質がわずかに向上しました。他のデータセットでもランダム水平反転を使用しましたが、LSUN Bedroom は例外です。

- 初期の実験では Adam [**?**] と RMSProp の両方を試し、前者を選択しました。ハイパーパラメータは標準値のままとし、学習率を 2×10^{-4} に設定しました（調査なし）。256 × 256 画像に対しては学習率を 2×10^{-5} に下げました。これは、より高い学習率ではトレーニングが不安定であることがわかったためです。

- CIFAR10 にはバッチサイズ 128 を、より大きな画像にはバッチサイズ 64 を設定しました。この値については調査を行いませんでした。

- モデルパラメータには指数移動平均（EMA）を用い、減衰係数を 0.9999 に設定しました。この値についても調査を行いませんでした。

最終実験は一度だけトレーニングを行い、その過程でサンプル品質を評価しました。サンプル品質スコアと対数尤度は、トレーニングの過程で得られた最小 FID 値に基づいて報告されています。CIFAR10 で