Directly Denoising Diffusion Models

Dan Zhang [1 *] , **Jingjing Wang** [1 *] , **Feng Luo** [1]

Abstract

In this paper, we present the Directly Denoising Diffusion Model (DDDM): a simple and generic approach for generating realistic images with few-step sampling, while multistep sampling is still preserved for better performance. DDDMs require no delicately designed samplers nor distillation on pre-trained distillation models. DDDMs train the diffusion model conditioned on an estimated target that was generated from previous training iterations of its own. To generate images, samples generated from the previous time step are also taken into consideration, guiding the generation process iteratively. We further propose Pseudo-LPIPS, a novel metric loss that is more robust to various values of hyperparameter. Despite its simplicity, the proposed approach can achieve strong performance in benchmark datasets. Our model achieves FID scores of 2.57 and 2.33 on CIFAR-10 in one-step and two-step sampling respectively, surpassing those obtained from GANs and distillation-based models. By extending the sampling to 1000 steps, we further reduce FID score to 1.79, aligning with state-of-the-art methods in the literature. Our code is available at https://github.com/TheLuoFengLab/DDDM.

1. Introduction

Diffusion models (DM) recently have attracted significant attention for their exceptional ability to generate realistic samples in recent years, DMs achieved impressive performance in many fields, including image generation (Dhariwal & Nichol, 2021; Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022a; Rombach et al., 2022), video generation (Ho et al., 2022), inpainting (Rombach et al., 2022) and super-resolution (Saharia et al., 2022b; Rombach et al., 2022). However, a remarkable drawback is their relatively slow sampling speed, which poses challenges for practical applications. For instance, the vanilla method of DMs (DDPM Ho et al. (2020), Score based models (Song et al., 2020b)) takes hundreds to thousands of steps in sampling, which is very time-consuming compared with one-step generation such as GAN-style models (Brock et al., 2018; Karras et al., 2020b), normalizing flow models (Kingma & Dhariwal, 2018; Dinh et al., 2016), or consistency models (Song et al., 2023; Song & Dhariwal, 2023). When directly generating the image using DDPM, the accumulated distortion leads to poor performance.

Many efforts to accelerate the sampling process of DM have been proposed. Denoising Diffusion Implicit Models (DDIM, Song et al. (2020a)) modified the diffusion process into a non-Markovian format with a smaller number of function evaluations (NFE) to generate samples. Meanwhile, by viewing the sampling process through the lens of ordinary differential equation (ODE), Song et al. (2020b) developed faster numerical solvers to reduce the NFE required for generation rapidly, thus speeding up the process significantly (Zhang & Chen, 2022; Lu et al., 2022). Although these solvers can achieve comparable results as thousands-of-step samplers, the performance for single-step generation is still not good. Furthermore, knowledge distillation-based methods (Luhman & Luhman, 2021) compress the information learned by the thousands-step sampler into a one-step model, which enables the one-step generation of samples. However, the distillation-based models add computational overhead to the training process as they require another pretrained diffusion model (teacher model) and have potential architectural constraints (Song & Dhariwal, 2023).

In this paper, we propose Directly Denoising Diffusion Model (DDDM) that combines the efficiency of single-step generation with the benefits of iterative sampling for improved sample quality. DDDM employs a DDPM-style noise scheduler and denoises under the probability flow ODE framework. However, we solve the probability flow ODE using a neural network only without using any ODE solver. Our method enables the generation of data samples from random noise with high quality in just one step. Moreover, DDDM can still allow multi-step sampling to obtain better generation results. Furthermore, inspired by Pseudo-Huber losses, we proposed pseudo Learned Percep-

[*]Equal contribution [1]School of Computing, Clemson University, USA. Correspondence to: Feng Luo <luofeng@clemson.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

tual Image Patch Similarity (LPIPS) (Zhang et al., 2018), which shows more robustness in our study.

In our experiments, we demonstrate the effectiveness of DDDMs across various image datasets including CIFAR-10 (Krizhevsky et al., 2009), and ImageNet 64x64 (Deng et al., 2009), and observe comparable results to current state-of-the-art methods. Our model achieves FID scores of 2.57 and 2.33 on CIFAR-10 in one-step and two-step sampling respectively. By extending the sampling to 1000 steps, we further reduce FID score to 1.79.

Our contributions can be summarized as follows:

- We introduce the Directly Denoising Diffusion Models (DDDM) that can obtain the performance for image generation results that is comparable to current state-of-the-art methods and can obtain better generation results for multiple-step sampling.
- Our model provides a straightforward pass with much fewer constraints and does not need ODE solvers.
- We proposed the Pseudo-LPIPS metric with increased sensitivity when the loss gets smaller, which is more robust.

2. Preliminary

2.1. Diffusion models

Inspired by non-equilibrium thermodynamics, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020b), present a generative framework that models data from an unknown true distribution $p_{\text{data}}(\mathbf{x})$. These models consist of two processes: a forward diffusion process and a reverse denoising process.

The forward diffusion process is characterized by the gradual introduction of noise into the original data, denoted as $\mathbf{x}_0$, over a sequence of time steps from 0 to T. This process is mathematically structured as a Markov chain, where Gaussian noise is incrementally added to the data at each step. At time step t, the distribution of $\mathbf{x}_t$ condition on $\mathbf{x}_{t-1}$ can be expressed as:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

By the property of Markov chain, the distribution of $\mathbf{x}_t$ given $\mathbf{x}_0$ is described as (Ho et al., 2020):

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right)$$

where $\bar{\alpha}_t=\prod_{s=1}^{t}\left(1-\beta_s\right)$, $q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)$ is also known as diffusion kernel.

The reverse denoising process aims to learn the inverse of the forward diffusion. Starting from a random sample $\mathbf{x}_T$ with distribution $p\left(\mathbf{x}_T\right)=\mathcal{N}\left(\mathbf{x}_T; \mathbf{0}, \mathbf{I}\right)$, this sampled data is then progressively denoised through a neural network that parameterizes a conditional distribution $q\left(\mathbf{x}_s \mid \mathbf{x}_t\right)$, where $s<t$. This denoising process continues step by step, moving backward in time from step T towards step 0. The sequence of denoising steps gradually reconstructs the data, aiming to approximate the original data as closely as possible when the time step 0 is reached.

2.2. Stochastic Differential Equation Formulation

The discrete processes in DDPM can be linked to continuous-time diffusion processes (Song et al., 2020a). By obtaining a continuous approximation of the forward discrete process, we can align it with a Stochastic Differential Equation (SDE) and consequently derive a reverse continuous-time process that corresponds with the reverse discrete process defined in diffusion models.

For the diffusion kernels $\{p_{\beta_i}\left(\mathbf{x} \mid \mathbf{x}_0\right)\}_{i=1}^{N}$ used in DDPM, we have:

$$\mathbf{x}_i=\sqrt{1-\beta_i}\mathbf{x}_{t-1}+\sqrt{\beta_t}\mathbf{z}_{t-1}, \mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $t=1, \cdots, T$ and β_t can be approximated to an infinitesimal function $\beta(t)\Delta t$ as $T \rightarrow \infty$, and β_t sufficiently small.

Applying Taylor expansion, the following can be derived:

$$\mathbf{x}_t \approx \mathbf{x}_{t-1}-\frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1}+\sqrt{\beta(t)\Delta t}\mathbf{z}_{i-1}.$$

As the time increment $\Delta t \rightarrow 0$, the above discrete function transitions into the following Variance Preserving (VP) Stochastic Differential Equation (SDE) (Song et al., 2020a):

$$\mathrm{d}\mathbf{x}_t=-\frac{1}{2}\beta(t)\mathbf{x}_t\mathrm{d}t+\sqrt{\beta(t)}\mathrm{d}\mathbf{w},$$

where $\mathbf{w}$ is a Wiener process.

The reverse process for this VP SDE is defined as:

$$\mathrm{d}\mathbf{x}_t=\left(-\frac{1}{2}\beta(t)\mathbf{x}_t-\beta(t)\nabla_x \log q(\mathbf{x}_t)\right)\mathrm{d}t+\sqrt{\beta(t)}\mathrm{d}\mathbf{w}$$

The Generative probability flow Ordinary Differential Equation (ODE), which is deterministic, can be expressed as:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}=-\frac{1}{2}\beta(t)\left[\mathbf{x}_t-\nabla_{x_t}\log q_t\left(\mathbf{x}_t\right)\right]. \tag{1}$$

By replacing $\nabla_{\boldsymbol{x}} \log q_t\left(\boldsymbol{x}_t\right)$ with a neural network based estimation $\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)$, Song et al. (2020b) obtained the neural ODE:

$$\mathrm{d}\mathbf{x}_t=-\frac{1}{2}\beta(t)\left[\mathbf{x}_t+\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\right]\mathrm{d}t$$

Advanced ODE solvers can be applied to solve the above equations.

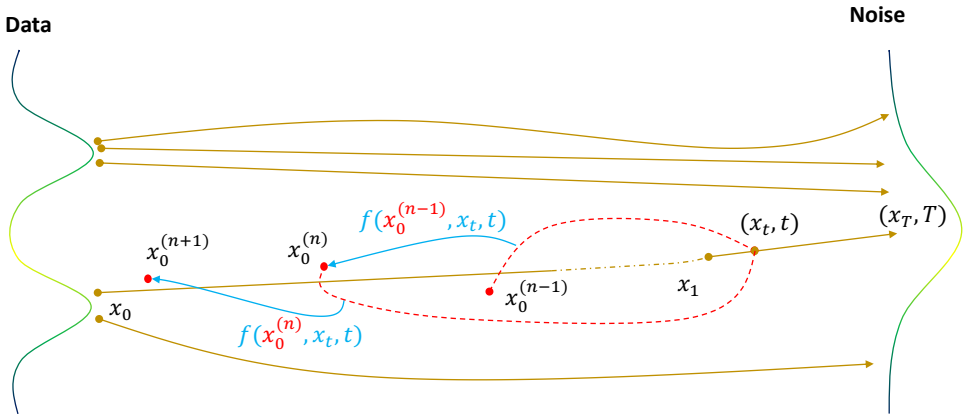


Figure 1. An illustration of DDDM. For current training epoch n, our model takes noisy data $\mathbf{x}_t$ and timestep t, as well as the estimated target from previous epoch $\mathbf{x}_0^{(n-1)}$ as inputs, predicts the new approximation $\mathbf{x}_0^{(n)}$, which will be utilized in the next training epoch. Through such an iterative process, our approximated result moves gradually towards real data $\mathbf{x}_0$.

3. Directly Denosing Diffusion Models

Solving the probability flow (PF) ODE is equivalent to computing the following integral:

$$\int_T^0 \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\,\mathrm{d}t \;=\; \int_T^0 -\tfrac{1}{2}\beta(t)\left[\mathbf{x}_t - \nabla_{x_t}\log q_t\left(\mathbf{x}_t\right)\right]\,\mathrm{d}t \iff \mathbf{x}_0 = \mathbf{x}_T + \int_T^0 -\tfrac{1}{2}\beta(t)\left[\mathbf{x}_t - \nabla_{x_t}\log q_t\left(\mathbf{x}_t\right)\right]\,\mathrm{d}t$$

where $\mathbf{x}_T$ is initialized from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

To generate samples from a DM, we propose Directly Denosing Diffusion Models, an iterative process designed to refine the estimation of $\mathbf{x}_0$. First, we define $\mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right)$ as the solution of the PF ODE from initial time t to final time 0 (Appendix A.1):

$$\mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) := \mathbf{x}_t + \int_t^0 -\frac{1}{2}\beta(s)\left[\mathbf{x}_s - \nabla_{x_s}\log q_s\left(\mathbf{x}_s\right)\right]\mathrm{d}s$$

where $\mathbf{x}_t$ is drawn from $\mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right)$. Subsequently, we introduce the function $\mathbf{F}\left(\mathbf{x}_0, \mathbf{x}_t, t\right)$ defined as:

$$\mathbf{F}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) := \int_t^0 \frac{1}{2}\beta(s)\left[\mathbf{x}_s - \nabla_{x_s}\log q_s\left(\mathbf{x}_s\right)\right]\mathrm{d}s$$

Thus, we have:

$$\mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) = \mathbf{x}_t - \mathbf{F}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) \tag{2}$$

By approximate $\mathbf{f}$, we can then recover the original image $\mathbf{x}_0$. We define a neural network-parameterized function $\mathbf{f}_{\boldsymbol{\theta}}$, which is employed to estimate the solution of the PF ODE and thereby recover the original image state at time 0. The predictive model is represented as:

$$\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) = \mathbf{x}_t - \mathbf{F}_{\boldsymbol{\theta}}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) \tag{3}$$

where $\mathbf{F}_{\boldsymbol{\theta}}$ is the neural network function parameterized by the weights $\boldsymbol{\theta}$. To achieve a good recovery of the initial state $\mathbf{x}_0$, $\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) \approx \mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right)$ need to be ensured.

3.1. Iterative solution

Eq. (3) shows that our neural network $\mathbf{F}_{\boldsymbol{\theta}}$ requires $\mathbf{x}_0$ as input, which is not applicable during sample generation. To unify the training and inference within the same framework, we propose an iterative update rule to estimate the initial state $\mathbf{x}_0$ in a dynamic system. This iterative process is formally defined by the following update equation:

$$\mathbf{x}_0^{(n+1)} = \mathbf{x}_t - \mathbf{F}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right) \tag{4}$$

where $\mathbf{x}_0^{(n)}$ denotes the estimated ground truth data $\mathbf{x}_0$ at the n-th training epoch or n-th sampling iteration. Each update refines this estimate in an attempt to converge to the true initial state. To effectively quantify the discrepancy between the n-th estimate $\mathbf{x}_0^{(n)}$ and the true initial state $\mathbf{x}_0$ in the DDDM, we employ the following loss function.

Definition 3.1. The loss function of the DDDM at the n-th iteration is defined as:

$$\begin{aligned}\mathcal{L}_{\text{DDDM}}^{(n)}(\boldsymbol{\theta}) :=& \mathbb{E}_{t\sim\mathcal{U}[1,T]}[\mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}}(\mathbf{x}_0)} \\ &[\mathbb{E}_{\mathbf{x}_t\sim\mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0,(1-\bar{\alpha}_t)\mathbf{I})} \\ &[d(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t), \mathbf{x}_0)]]]\end{aligned} \tag{5}$$

where $\mathcal{U}\left[1, T\right]$ denotes a uniform distribution over the integer set $[1, 2, \cdots, T]$. $d(\cdot,\cdot)$ is a metric function satisfies that for all vectors $\mathbf{x}$ and $\mathbf{y}$, $d(\mathbf{x}, \mathbf{y}) \geq 0$ and $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$. Therefore, commonly used metrics such as L_1, L_2 can be utilized. We will discuss our choice of $d(\cdot,\cdot)$ in Section 4.

This definition encapsulates the expected discrepancy between the estimated state $\mathbf{x}_0^{(n)}$ and the true initial state $\mathbf{x}_0$, integrated over a probabilistic model of the data and the time domain.

Algorithm 1 Training

Input: image dataset D, T, model parameter $\boldsymbol{\theta}$
initialize $\mathbf{x}_0^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, epoch $n \leftarrow 0$
repeat
  Sample $\mathbf{x}_0 \sim D$ and $t \sim \mathcal{U}[1, T]$.
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
  $\mathbf{x}_0^{(n+1)} \leftarrow \mathbf{x}_t - \mathbf{F}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right)$
  $\mathcal{L}_{\text{DDDM}}^{(n+1)}(\boldsymbol{\theta}) \leftarrow d\left(\boldsymbol{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right), \mathbf{x}_0\right)$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$
  $n \leftarrow n + 1$
until convergence

Algorithm 2 Sampling

Input: T, trained model parameter $\boldsymbol{\theta}$, sampling step s
$\mathbf{x}_0^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $n = 0$ **to** $s - 1$ **do**
  $\mathbf{x}_0^{(n+1)} \leftarrow \mathbf{x}_T - \mathbf{F}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_T, T\right)$
end for
Output: $\mathbf{x}_0^{(n+1)}$

Training. Each data sample $\mathbf{x}_0$ is chosen randomly from the dataset, following the probability distribution $p_{\text{data}}(\mathbf{x}_0)$. This initial data point forms the basis for generating a trajectory. Next, we randomly sample a timestep $t \sim \mathcal{U}[1, T]$, and obtain its noisy variant $\mathbf{x}_t$ from distribution $\mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$. we play the reparameterization trick to rewrite $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. For current training epoch n, our model takes noisy data $\mathbf{x}_t$ and timestep t, as well as the corresponding estimated target from previous epoch $\mathbf{x}_0^{(n-1)}$ as inputs, predicts a new approximation $\mathbf{x}_0^{(n)}$, which will be utilized in the next training epoch for the same target sample. DDDM is trained by minimizing the loss following Eq. 5. The full procedure of training DDDM is summarized in Algorithm 1.

Sampling. The generation of samples is facilitated through the use of a well-trained DDDM, denoted as $\boldsymbol{f}_{\boldsymbol{\theta}}(\cdot, \cdot)$. The process begins by drawing from the initial Gaussian distribution, where both $\mathbf{x}_0^{(0)}$ and $\mathbf{x}_T$ are sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Subsequently, these noise vectors and embedding of T are passed through the DDDM model to obtain $\mathbf{x}_0^{\text{est}} = \mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(0)}, \mathbf{x}_T, T\right)$. This approach is noteworthy for its efficiency, as it requires only a single forward pass through the model. Our model also supports a multistep sampling procedure for enhanced sample quality. Detail can be found in Algorithm 2.

Here, we provide theoretical justifications for the convergence of our method below.

Theorem 3.2. *If the loss function $\mathcal{L}_{DDDM}^{(n)}(\boldsymbol{\theta}) \to 0$ as $n \to \infty$, it can be shown that as $n \to \infty$,*

$$\sup_{\mathbf{x}_0} \left(\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right) - \mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right)\right) \to 0 \tag{6}$$

Proof. As $n \to \infty$, $\mathcal{L}_{\text{DDDM}}^{(n)}(\boldsymbol{\theta}) \to 0$, we have

$$\mathbb{E}\left[d\left(\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right), \mathbf{x}_0\right)\right] \to 0$$

According to the definition, we have

$p(\mathbf{x}_t) > 0$ for every $\mathbf{x}_t$ and $1 \leqslant t \leqslant T$. Therefore, we have:

$$d\left(\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right), \mathbf{x}_0\right) \to 0$$

Because $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$, this indicates that:

for any $\mathbf{x}_0$ sampled from $p_{\text{data}}(\mathbf{x}_0)$,

$$\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right) \to \mathbf{x}_0, \text{ as } n \to \infty,$$

which implies:

$$\sup_{\mathbf{x}_0} d\left(\mathbf{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_0^{(n)}, \mathbf{x}_t, t\right) - \mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right)\right) \to 0.$$

□

The following theorem draws inspiration from Proposition 3 in Kim et al. (2023), leading us to a similar conclusion.

Theorem 3.3. *Suppose the following conditions are met:*

(i) For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$, time $t \in \mathcal{U}[1, T]$, $\mathbf{x}_t \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$, $\mathbf{y}_t \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{y}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$, the function $\mathbf{f}_{\boldsymbol{\theta}}$ satisfies the uniform Lipschitz condition:

$$\sup_{\boldsymbol{\theta}} \|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_t, t) - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{y}_t, t)\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2$$

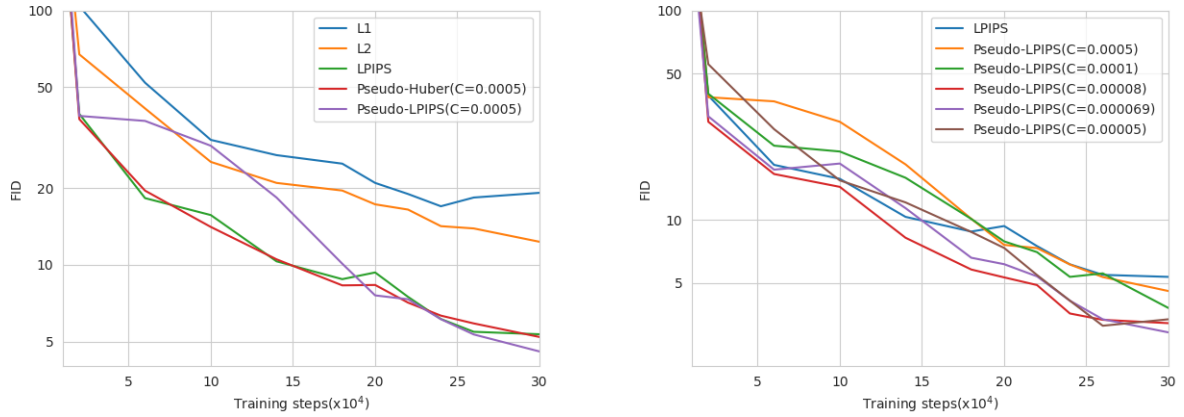
where L is a Lipschitz constant.

(ii) There exists a non-negative function $L(\mathbf{x})$ such that for all $\mathbf{x} \in \mathbb{R}^D$, time $t \in \mathcal{U}[1, T]$ and $\mathbf{x}_t \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}, (1-\bar{\alpha}_t)\mathbf{I}\right)$, the function $\mathbf{f}_{\boldsymbol{\theta}}$ is uniformly bounded in $\boldsymbol{\theta}$:

$$\sup_{\boldsymbol{\theta}} \|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_t, t)\|_2 \leq L(\mathbf{x}) < \infty.$$

If there exists a $\tilde{\boldsymbol{\theta}}$ such that the loss function $\mathcal{L}_{DDDM}\left(\tilde{\boldsymbol{\theta}}\right) \to 0$ as the iteration number n becomes sufficiently large. Let $p_{\tilde{\boldsymbol{\theta}}}(\cdot)$ denote the pushforward distribution of p_T induced by $\mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\cdot, \mathbf{x}_T, T)$.

Then, under these conditions, the discrepancy between pushforward distribution $p_{\tilde{\boldsymbol{\theta}}}(\cdot)$ and the data distribution $p_{data}(\cdot)$ measured in the uniform norm, converges to 0 as the number of iterations n approaches infinity: $\left\|p_{\tilde{\boldsymbol{\theta}}}(\cdot) - p_{data}(\cdot)\right\|_\infty \to 0$ as iteration $n \to \infty$.



(a) (b)

Figure 2. Ablation analysis for our proposed Pseudo-LPIPS metric. (a) While LPIPS and Pseudo-Huber perform closely, Pseudo-LPIPS further reduces FID to under 5. (b) Pseudo-LPIPS outperforms LPIPS with various values of hyperparameter c, where $c = 0.000069$ is the best. The y-axis for both figures is scaled logarithmically for better visualization.

Proof. Based on Theorem 3.2, let $t = T$ we have that for sufficiently large n,

$$\sup_{\mathbf{x}_0} \left\| \mathbf{f}_{\tilde{\theta}} \left(\mathbf{x}_0^{(n)}, \mathbf{x}_T, T \right) - \mathbf{f} \left(\mathbf{x}_0, \mathbf{x}_T, T \right) \right\|_2 \to 0$$

which implies that

$$\mathbf{f}_{\tilde{\theta}} \left(\mathbf{x}_0^{(n)}, \mathbf{x}_T, T \right) \to \mathbf{f} \left(\mathbf{x}_0, \mathbf{x}_T, T \right)$$

when n is large enough. Then we conclude that the push-forward distribution of $\mathbf{x}_T$, say $p_{\tilde{\boldsymbol{\theta}}}(\cdot)$, converges in distribution to the data distribution $p_{\text{data}(\cdot)}$. Since for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D, t \sim \mathcal{U}[1, T]$, and $\boldsymbol{\theta}$,$\{\mathbf{f}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ is uniform Lipschitz $\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_t, t) - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{y}_t, t)\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2$, $\{\mathbf{f}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ is asymptotically uniformly equicontinuous (a.u.e.c.). Additionally, $\{\mathbf{f}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ is uniform bounded in $\tilde{\boldsymbol{\theta}}$. Thus, by the converse of Scheffé's theorem (Boos, 1985; Sweeting, 1986), it suggests $\left\| p_{\tilde{\boldsymbol{\theta}}}(\cdot) - p_{\text{data}}(\cdot) \right\|_\infty \to 0$ as n sufficiently large. □

4. Psuedo LPIPS Metric

Assessment of image quality becomes increasingly crucial in image generation. The Learned Perceptual Image Patch Similarity (LPIPS, Zhang et al. (2018)) metric has been a significant tool to help improve the quality of generated images. However, the LPIPS is still not robust enough to outliers in practice. Inspired by Song & Dhariwal (2023) recent using the pseudo-Huber loss to significantly improve the robustness of the training consistency model, we propose a modified version of the LPIPS metric, defined as Pseudo-LPIPS:

$$\text{Pseudo-LPIPS} = \sqrt{\text{LPIPS} + c^2} - c \tag{7}$$

where c is an adjustable hyperparamter. Similar to the Pseudo-Huber metric, the inclusion of the term c^2 and the subsequent square root transformation in the Pseudo-LPIPS metric aim to provide a more balanced and perceptually consistent measure. This approach mitigates the overemphasis on larger errors and increases the sensitivity and accuracy of the metric in discerning perceptual differences in images.

The merits of the Pseudo-LPIPS Metric are as follows:

- Enhanced Sensitivity to Perceptual Differences: The modified metric is finely attuned to subtle perceptual variances, often missed by traditional metrics. This sensitivity is especially valuable in fields requiring high-precision image quality, like medical imaging or high-fidelity rendering.
- Balanced Error Emphasis: It provides a more equitable emphasis across various error magnitudes, contrasting the L_2 norm's tendency to disproportionately penalize larger errors.
- Adaptability: The incorporation of the constant c allows for flexibility, making the metric versatile for different scenarios and datasets.
- Improved Robustness: The metric is more resilient against outliers and anomalies due to the square root transformation, addressing a common flaw in the Pseudo-Huber loss.

When comparing this modified metric to the $L2$ norm and pseudo-Huber loss, the Modified LPIPS aligns more closely with human perceptual judgments. The $L2$ norm, while simple in its mathematical form, often fails to accurately

Table 1. Comparing the quality of unconditional samples on CIFAR-10

Method	NFE(↓)	FID(↓)	IS(↑)
Fast samplers & distillation for diffusion models			
DDIM (Song et al., 2020a)	10	13.36	
DPM-solver-fast (Lu et al., 2022)	10	4.70	
3-DEIS (Zhang & Chen, 2022)	10	4.17	
UniPC (Zhao et al., 2023)	10	3.87	
DFNO (LPIPS) (Zheng et al., 2023)	1	3.78	
2-Rectified Flow (Liu et al., 2022)	1	4.85	9.01
Knowledge Distillation (Luhman & Luhman, 2021)	1	9.36	
TRACT (Berthelot et al., 2023)	1	3.78	
	2	3.32	
Diff-Instruct (Luo et al., 2023)	1	4.53	9.89
CD (LPIPS) (Song et al., 2023)	1	3.55	9.48
	2	2.93	9.75
Direct Generation			
Score SDE (Song et al., 2020b)	2000	2.38	9.83
Score SDE (deep) (Song et al., 2020b)	2000	2.20	9.89
DDPM (Ho et al., 2020)	1000	3.17	9.46
LSGM (Vahdat et al., 2021)	147	2.10	
PFGM (Xu et al., 2022)	110	2.35	9.68
EDM (Karras et al., 2022)	35	2.04	9.84
EDM-G++ (Kim et al., 2022)	35	1.77	
NVAE (Vahdat & Kautz, 2020)	1	23.5	7.18
BigGAN (Brock et al., 2018)	1	14.7	9.22
StyleGAN2 (Karras et al., 2020a)	1	8.32	9.21
StyleGAN2-ADA (Karras et al., 2020b)	1	2.92	9.83
CT (LPIPS) (Song et al., 2023)	1	8.70	8.49
	2	5.83	8.85
iCT (Song & Dhariwal, 2023)	1	2.83	9.54
	2	2.46	9.80
iCT-deep (Song & Dhariwal, 2023)	1	2.51	9.76
	2	2.24	9.89
DDDM(T=1000)	1	2.90	9.81
	2	2.79	9.89
	1000	1.87	9.94
DDDM(T=8000)	1	2.82	9.83
	2	2.53	9.84
	1000	1.74	9.93
DDDM-deep(T=1000)	1	2.57	9.91
	2	2.33	9.91
	1000	1.79	9.95

Table 2. Comparing the quality of class-conditional samples on ImageNet 64x64

Method	NFE(↓)	FID(↓)	Prec.(↑)	Rec.(↑)
Fast samplers & distillation for diffusion models				
DDIM (Song et al., 2020a)	50	13.7	0.65	0.56
	10	18.3	0.60	0.49
DPM solver (Lu et al., 2022)	10	7.93		
	20	3.42		
DEIS (Zhang & Chen, 2022)	10	6.65		
	20	3.10		
DFNO (LPIPS) (Zheng et al., 2023)	1	7.83	0.61	
TRACT (Berthelot et al., 2023)	1	7.43		
	2	4.97		
BOOT (Gu et al., 2023)	1	16.3	0.68	0.36
Diff-Instruct (Luo et al., 2023)	1	5.57		
PD (LPIPS) (Song et al., 2023)	1	7.88	0.66	0.63
	2	5.74	0.67	0.65
	4	4.92	0.68	0.65
CD (LPIPS) (Song et al., 2023)	1	6.20	0.68	0.63
	2	4.70	0.69	0.64
	3	4.32	0.70	0.64
Direct Generation				
RIN (Jabri et al., 2022)	1000	1.23		
DDPM (Ho et al., 2020)	250	11.0	0.67	0.58
iDDPM (Nichol & Dhariwal, 2021)	250	2.92	0.74	0.62
ADM (Dhariwal & Nichol, 2021)	250	2.07	0.74	0.63
EDM (Karras et al., 2022)	511	1.36		
BigGAN-deep (Brock et al., 2018)	1	4.06	0.79	0.48
CT (LPIPS) (Song et al., 2023)	1	13.0	0.71	0.47
	2	11.1	0.69	0.56
iCT (Song & Dhariwal, 2023)	1	4.02	0.70	0.63
	2	3.20	0.73	0.63
iCT-deep (Song & Dhariwal, 2023)	1	3.25	0.72	0.63
	2	2.77	0.74	0.62
DDDM(T=1000)	1	4.21	0.71	0.64
	2	3.53	0.73	0.64
	1000	2.76	0.75	0.65
DDDM-deep(T=1000)	1	3.47	0.71	0.63
	2	3.08	0.74	0.66
	1000	2.11	0.73	0.67

represent human vision. The Pseudo-Huber loss, although attempting to merge the benefits of $L1$ and $L2$ norms, sometimes falls short in providing a balanced representation of perceptual quality. The Pseudo-LPIPS, through its nuanced formulation, effectively bridges these gaps, presenting a metric that is both perceptually meaningful and mathematically sound.

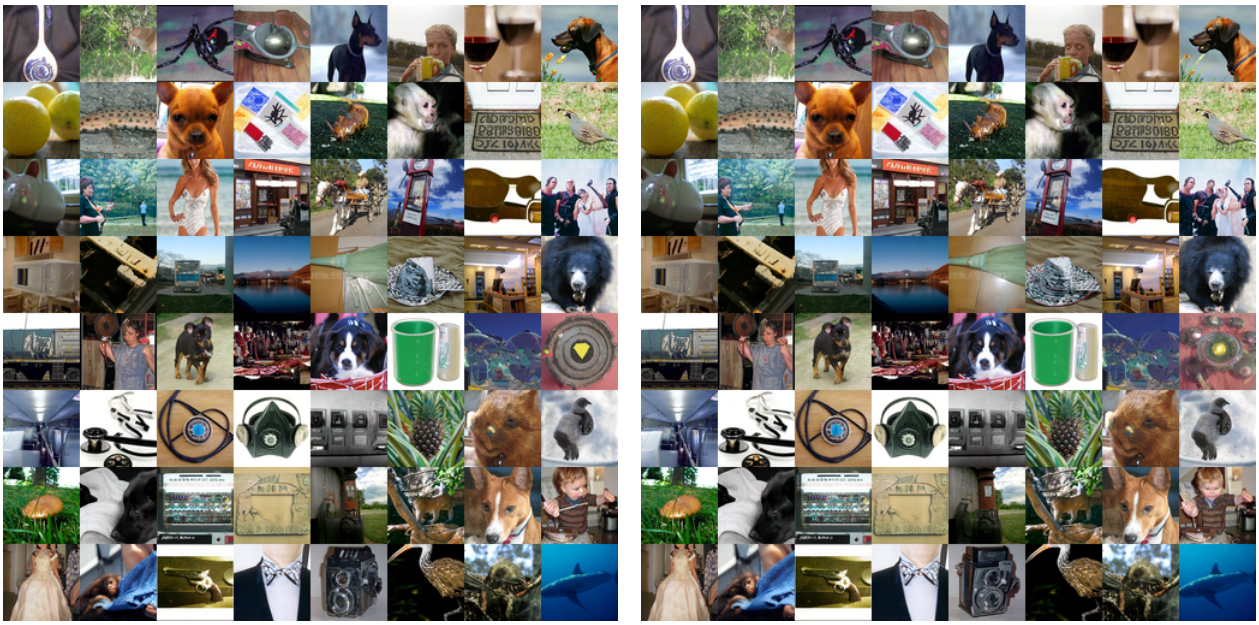
5. Experiments

To evaluate our method for image generation, we train several DDDMs on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet 64x64 (Deng et al., 2009) and benchmark their performance with competing methods in the literature. Results are compared according to Frechet Inception Distance (FID, Heusel et al. (2017)), which is computed between 50K generated samples and the whole training set. We also employ Inception Score (IS, Salimans et al. (2016)) and Precision/Recall (Kynkäänniemi et al., 2019) to measure sample quality.

5.1. Implementation Details

Architecture. We use the U-Net architecture from ADM (Dhariwal & Nichol, 2021) for both datasets. For CIFAR-10, we use a base channel dimension of 128, multiplied by 1,2,2,2 in 4 stages and 3 residual blocks per stage. Dropout (Srivastava et al., 2014) of 0.3 is utilized for this task. For ImageNet 64x64, we use a base channel dimension of 192, multiplied by 1,2,3,4 in 4 stages and 3 residual blocks per stage, which account for a total of 270M parameters. Following ADM, we employ cross-attention modules not only at the 16x16 resolution but also at the 8x8 resolution, through which we incorporate the conditioning image $\mathbf{x}_0^{(n)}$ into the network. We also explore deeper variants of these architectures by doubling the number of blocks at each resolution, which we name DDDM-deep. All models on CIFAR-10 are unconditional, and all models on ImageNet 64x64 are conditioned on class labels.

Other settings. We use Adam for all of our experiments.



(a) one-step (b) two-step

Figure 3. One-step and two-step samples from DDDM-deep model trained on ImageNet 64x64

For CIFAR-10, we set $T = 1000$ for baseline model and train the model for 1000 epochs with a constant learning rate of 0.0002 and batch size of 1024. We also explore models with larger T values and longer training epochs. Details can be found in Table 3. For ImageNet 64×64, we only investigate $T = 1000$ due to time constraints and train the model for 520 epochs with a constant learning rate of 0.0001 and batch size of 1024. We use an exponential moving average (EMA) of the weights during training with a decay factor of 0.9999 for all the experiments. All models are trained on 8 Nvidia A100 GPUs.

5.2. Ablations

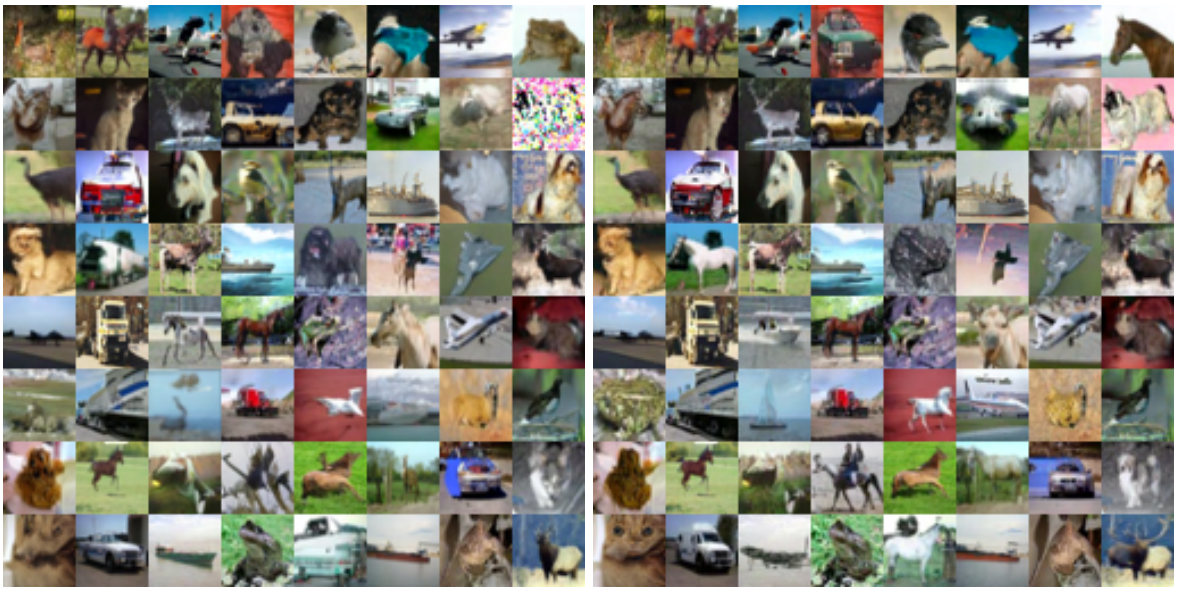
In this section, we ablate various metrics employed in the loss function. We evaluate the effectiveness of the proposed Pseudo-LPIPS metric by training several models with varying c values and comparing the sample qualities with models trained with L_1, squared L_2, LPIPS, and Pseudo-Huber on CIFAR-10. As depicted by Figure 2a, Pseudo-LPIPS outperforms L_1 and squared L_2 by a substantial margin. Given the same value of c, Pseudo-LPIPS exhibits notably superior results compared to Pseudo-Huber metrics. Figure 2b shows hyperparameter c in our proposed metric plays a significant role in sample quality. When $c = 0$, Pseudo-LPIPS degrades to LPIPS and it is clear that Pseudo-LPIPS consistently outperforms LPIPS even when the value of c varies in a relatively wide range. These findings collectively validate the effectiveness and robustness of our proposed metrics.

T	Epochs	FID(one-step)
1000	1000	2.90
2000	2000	2.86
4000	4000	2.83
8000	8000	2.82

Table 3. DDDM with different training configurations on CIFAR10.

5.3. Comparison to SOTA

We compare our model against state-of-the-art generative models on CIFAR-10 and ImageNet 64x64. Quantitative results are summarized in Table 1 and Table 2. Our findings reveal that DDDMs exceed previous distillation diffusion models and methods that require advanced sampling procedures in both one-step and two-step generation on CIFAR-10 and ImageNet 64x64, which breaks the reliance on the well-pretrained diffusion models and simplifies the generation workflow. Moreover, our model demonstrates performance comparable to numerous leading generative models on both datasets. Specifically, baseline DDDM obtains FIDs of 2.90 and 2.79 for one-step and two-step generation on CIFAR-10, both results exceed that of StyleGAN2-ADA (Karras et al., 2020b). With deeper architecture and 1000-step sampling, DDDM-deep further reduces FID to 1.79, aligning with state-of-the-art method (Kim et al., 2022). It is worth noting that the leading few-step generation model iCT/iCT-deep (Song & Dhariwal, 2023) is trained for 400k iterations, while our approach delivers competitive FIDs and higher



(a) one-step (b) two-step

Figure 4. One-step and two-step samples from DDDM-deep model trained on CIFAR-10

IS scores under fewer training iterations. With $T = 8000$ and trained for competitive epochs, DDDM achieves 2.82 and 1.74 on one-step and 1000-step generation respectively, both setting state-of-the-art performance.

On the ImageNet 64x64 dataset, DDDM attains FID scores of 4.21 and 3.53 for one-step and two-step generation, respectively. We have observed that iCT/iCT-deeper achieves superior results, benefitting from a 4x larger batch size and 1.6x more training iterations compared to our model. We hypothesize that the observed performance gap may be attributed to such computational resource disparities and suboptimal hyperparameters in our loss function. Despite these limitations, DDDM showcases improved precision and recall compared to iCT, demonstrating enhanced diversity and mode coverage while maintaining a similar model size.

The effectiveness of our iterative solution can also be clearly demonstrated by Figure 5. Overall, the FID consistently demonstrates a downward trend among different datasets and architectures, though it is not strictly monotonically decreasing with respect to the sampling steps.

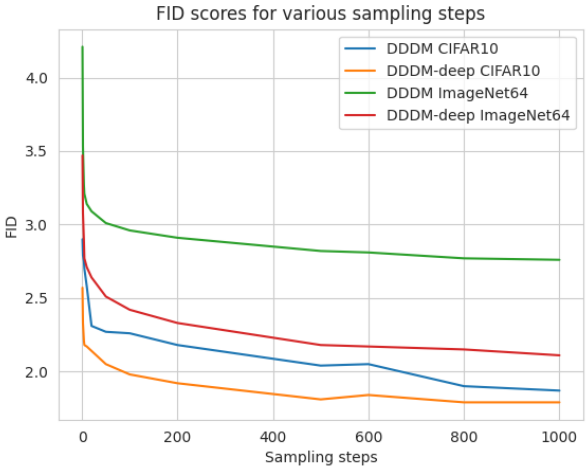


Figure 5. FID w.r.t inference iterations.

6. Related Work

The foundational work in diffusion probabilistic models (DPM) was initially conceptualized by Sohl-Dickstein et al. in 2015, where a generative Markov chain is developed to transfer the Gaussian distribution into the data distribution. Then Ho et al. (2020) developed denoising diffusion probabilistic models (DDPM) and demonstrated their exceptional capabilities in image generation. By improving noise schedule and variance taking into consideration, Nichol & Dhariwal further enhanced these models in 2021, achieving better log-likelihood scores and better FID scores. Song et al. focused on optimizing the score-matching objective and developed Noise Conditional Score Network (NCSN) (Song & Ermon, 2020). Despite their different motivations, DDPMs and NCSNs are closely related. Both DDPM and NCSN require many steps to achieve good sample quality and therefore have trouble generating high-quality samples in a few iterations.

Many studies tried to reduce the sampling steps. DDIM (Song et al., 2020a) has demonstrated effectiveness in

few-step sampling, similar to the probability flow sampler. Jolicoeur-Martineau et al. (2021) examine fast stochastic differential equation integrators for reverse diffusion processes, and Tzen & Raginsky (2019) explore unbiased samplers conducive to fast, high-quality sampling. (Nichol & Dhariwal, 2021) and Kong & Ping (2021) describe methodologies for adapting discrete-time diffusion models. Watson et al. (2021) have proposed a dynamic programming algorithm aimed at minimizing the number of timesteps required for a diffusion model, optimizing for log-likelihood.

Several studies have also shown the effectiveness of training diffusion models across continuous noise levels and subsequently tuning samplers post-training. Kingma et al. (2021) involves adjusting the noise levels of a few-step discrete time reverse diffusion process. San-Roman et al. (2021) train a new network to estimate the noise level in noisy data, demonstrating how this estimate can expedite sampling.

Modifications to the specifications of the diffusion model itself can also facilitate faster sampling. This includes altered forward and reverse processes, as studied by Nachmani et al. (2021) and Lam et al. (2021). Moreover, Consistency models (Song et al., 2023) boost the sampling speed into a single step, in both consistency distillation (CD) and consistency training (CT). Later, by introducing several techniques, the one-step generation reaches state-of-the-art FID scores (Song & Dhariwal, 2023).

7. Discussion and Limitations

Since DDDM keeps track of $\mathbf{x}_0^{(n)}$ for each sample in the dataset, there will be additional memory consumption during training. Specifically, it requires extra 614MB for CIFAR10 and 29.5GB for ImageNet 64x64. Although it can be halved by using FP16 data type, such memory requirement might still be a challenge for larger dataset or dataset with high-resolution images.

We also notice that there could be bias in evaluation since the ImageNet is utilized in both LPIPS and Inception network for FID. The accidental leakage of ImageNet features from LPIPS may potentially lead to inflated FID scores. Other evaluation metrics such as human evaluations are needed to further validate our model. Furthermore, investigating unbiased loss for DDDM presents an interesting avenue for future research.

8. Conclusion

In conclusion, our presented DDDMs offer a straightforward and versatile approach to generating realistic images with minimal sampling steps and also support the iterative sampling process for better performance, eliminating the need for intricately designed samplers or distillation on pre-trained models. The core concept of our method involves conditioning the diffusion model on an estimated target generated from the previous training iteration, and during image generation, incorporating samples from previous timestep to guide the iterative process. Additionally, the incorporation of the proposed Pseudo-LPIPS enhances the robustness of our model, showcasing its potential for broader applications in other generative models. To further enhance the capabilities of DDDM and unleash its potential, we aim to leverage its strengths in a continuous-time setting.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement

Research primarily supported as part of the AIM for Composites, an Energy Frontier Research Center, an Energy Frontier Research Center funded by the U.S. Department of Energy (DOE), Office of Science, Basic Energy Sciences (BES), under Award # DE-SC0023389 (Development of DDDM method) and by the US National Institute of Food and Agriculture (NIFA; Grant Number 2023-70029-41309) and the US National Science Foundation (NSF; Grant Number ABI-1759856, MTM2-2025541, OIA-2242812) (DDDM implementation and theoretical proof). In addition, the authors acknowledge research support from Clemson University with a generous allotment of computation time on the Palmetto cluster.

References

Berthelot, D., Autef, A., Lin, J., Yap, D. A., Zhai, S., Hu, S., Zheng, D., Talbot, W., and Gu, E. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Boos, D. D. A converse to scheffe's theorem. *The Annals of Statistics*, pp. 423–427, 1985.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Gu, J., Zhai, S., Zhang, Y., Liu, L., and Susskind, J. Boot: Data-free distillation of denoising diffusion models with bootstrapping. *arXiv preprint arXiv:2306.05544*, 2023.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

Ho, J., Salimans, T., Gritsenko, A. A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. URL https://openreview.net/forum?id=BBelR2NdDZ5.

Jabri, A., Fleet, D., and Chen, T. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., and Mitliagkas, I. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020a.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020b.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

Kim, D., Kim, Y., Kwon, S. J., Kang, W., and Moon, I.-C. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Kong, Z. and Ping, W. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Lam, M. W., Wang, J., Huang, R., Su, D., and Yu, D. Bilateral denoising diffusion models. *arXiv preprint arXiv:2108.11514*, 2021.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Luhman, E. and Luhman, T. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Luo, W., Hu, T., Zhang, S., Sun, J., Li, Z., and Zhang, Z. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *arXiv preprint arXiv:2305.18455*, 2023.

Nachmani, E., Roman, R. S., and Wolf, L. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

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

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022. *URL https://arxiv. org/abs/2204.06125*, 7, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022a.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022b.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

San-Roman, R., Nachmani, E., and Wolf, L. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021.

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

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Song, Y. and Dhariwal, P. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Sweeting, T. On a converse to scheffé's theorem. *The Annals of Statistics*, 14(3):1252–1256, 1986.

Tzen, B. and Raginsky, M. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Conference on Learning Theory*, pp. 3084–3114. PMLR, 2019.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Watson, D., Ho, J., Norouzi, M., and Chan, W. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021.

Xu, Y., Liu, Z., Tegmark, M., and Jaakkola, T. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022.

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhao, W., Bai, L., Rao, Y., Zhou, J., and Lu, J. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023.

Zheng, H., Nie, W., Vahdat, A., Azizzadenesheli, K., and Anandkumar, A. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, pp. 42390–42402. PMLR, 2023.

A. Appendix

A.1. Derivation of the definition of $\mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right)$

Starting with Eq. (1) we find that the integration of $\mathbf{x}_t$ from time t to 0 is given by:

$$\int_t^0 \frac{\mathrm{d}\mathbf{x}_s}{\mathrm{d}s}\,\mathrm{d}s = \int_t^0 -\frac{1}{2}\beta(s)\left[\mathbf{x}_s - \nabla_{x_s}\log q_s\left(\mathbf{x}_s\right)\right]\mathrm{d}s$$

Thus,

$$\mathbf{x}_0 - \mathbf{x}_t = -\int_t^0 \frac{1}{2}\beta(s)\left[\mathbf{x}_s - \nabla_{x_s}\log q_s\left(\mathbf{x}_s\right)\right]\mathrm{d}s$$

Identifying the right-hand side of this equation as a function of $(\mathbf{x}_0, \mathbf{x}_t, t)$ allows us to introduce the $\mathbf{F}(\mathbf{x}_0, \mathbf{x}_t, t)$. Consequently, it can be reformulated as:

$$\mathbf{x}_0 - \mathbf{x}_t = -\mathbf{F}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) \Longrightarrow \mathbf{x}_0 = \mathbf{x}_t - \mathbf{F}\left(\mathbf{x}_0, \mathbf{x}_t, t\right),$$

leading to the definition:

$$\mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) = \mathbf{x}_t - \mathbf{F}\left(\mathbf{x}_0, \mathbf{x}_t, t\right).$$

In our case, $\mathbf{f}\left(\mathbf{x}_0, \mathbf{x}_t, t\right) = \mathbf{x}_0\left(\mathbf{x}_t, t\right)$ suggests that $\mathbf{x}_0$ is a function of $\mathbf{x}_t$ and t, but it is embedded within a larger function f that equates to $\mathbf{x}_0$. This setup implies an implicit relationship between $\mathbf{x}_t, t$, and $\mathbf{x}_0$. In our implicit case, no direct expression is present, and we often cannot isolate one of the variables on one side of the equation without involving the others. Thus, it lets the neural network function estimate all the unstable parts.

In contrast, in DDPM, $\mathbf{x}_0$ is approximated as $\hat{\mathbf{x}}_0 = \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta\left(\mathbf{x}_t, t\right)\right)/\sqrt{\bar{\alpha}_t}$, presenting a partially explicit framework for relating $(\mathbf{x}_t, t)$ to $\mathbf{x}_0$. This equation, though it provides a method to estimate $\mathbf{x}_0$, highlights the potential for numerical instability. The division by $\sqrt{\bar{\alpha}_t}$ can amplify errors in estimating the noise $\epsilon_\theta\left(\mathbf{x}_t, t\right)$, especially as $\bar{\alpha}_t$ becomes small, which is typical in the latter stages of the reverse process where the data is more significantly noised.

B. Additional Samples

In this section, we provide additional samples from our models.

Figure 6. One-step samples from DDDM model trained on CIFAR-10

Figure 7. Two-step samples from DDDM model trained on CIFAR-10

Figure 8. 1000-step samples from DDDM model trained on CIFAR-10

Figure 9. One-step samples from DDDM-deep model trained on CIFAR-10

Figure 10. Two-step samples from DDDM-deep model trained on CIFAR-10

Figure 11. 1000-step samples from DDDM-deep model trained on CIFAR-10

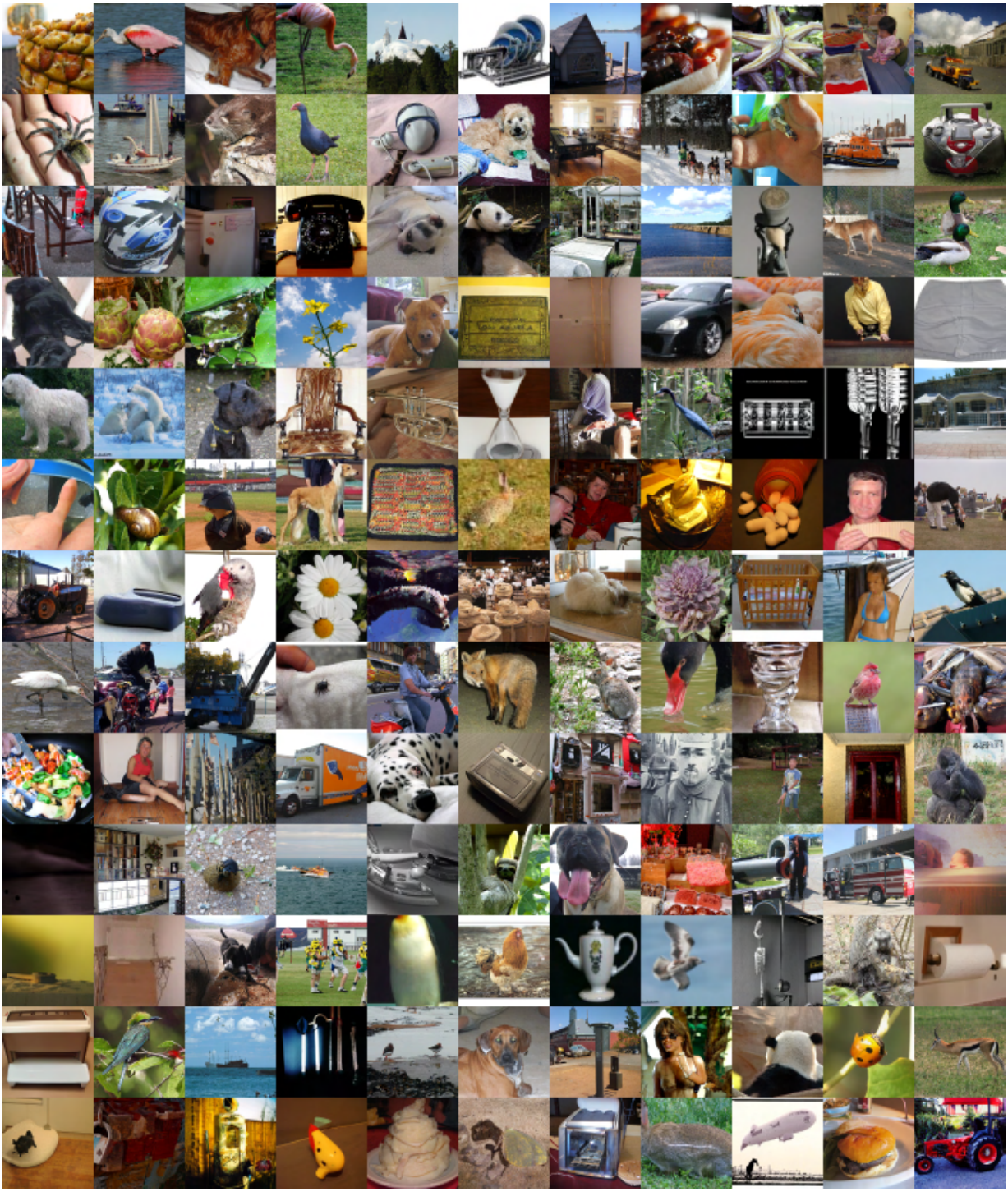


Figure 12. one-step samples from DDDM model trained on ImageNet 64x64

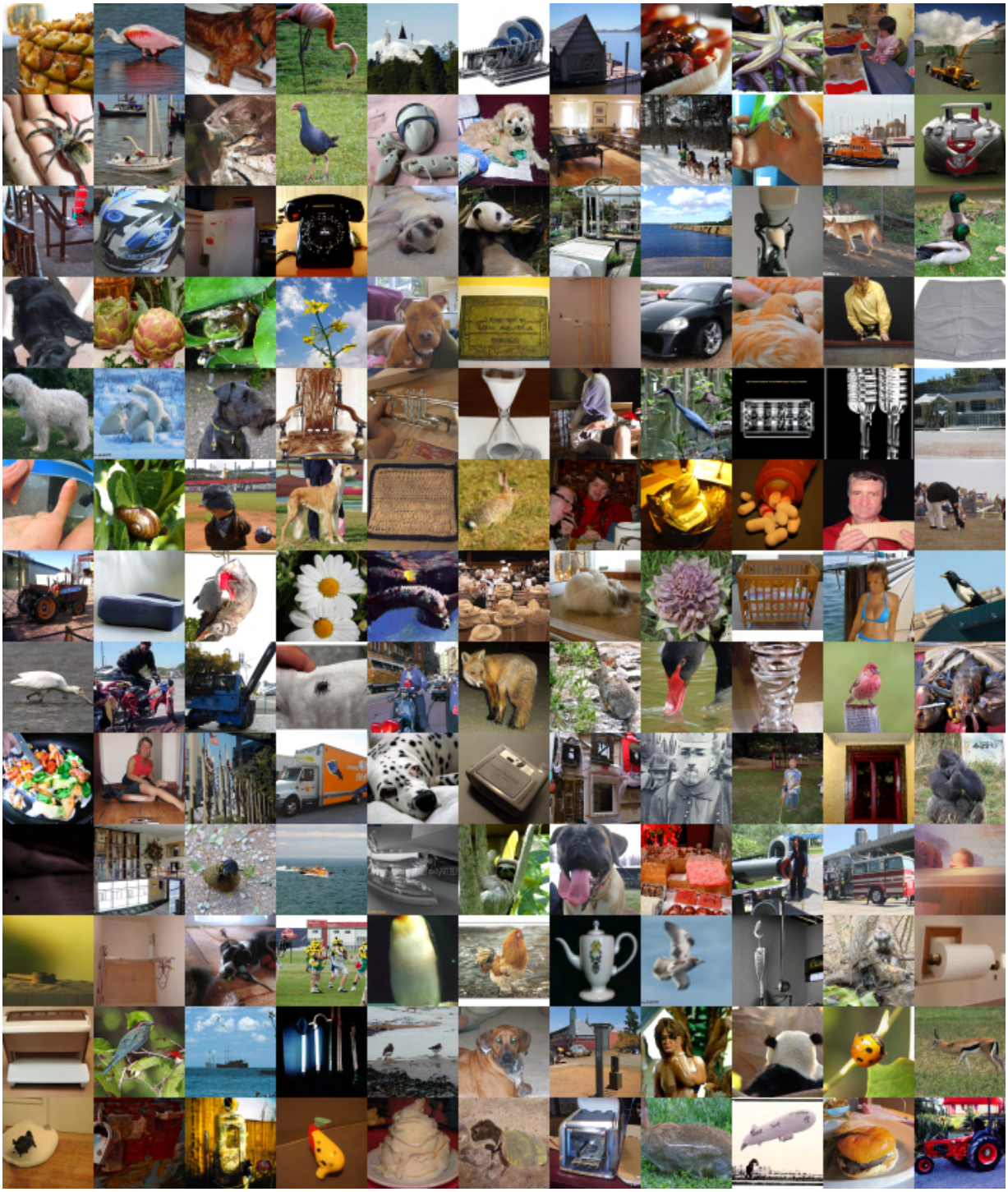


Figure 13. two-step samples from DDDM model trained on ImageNet 64x64

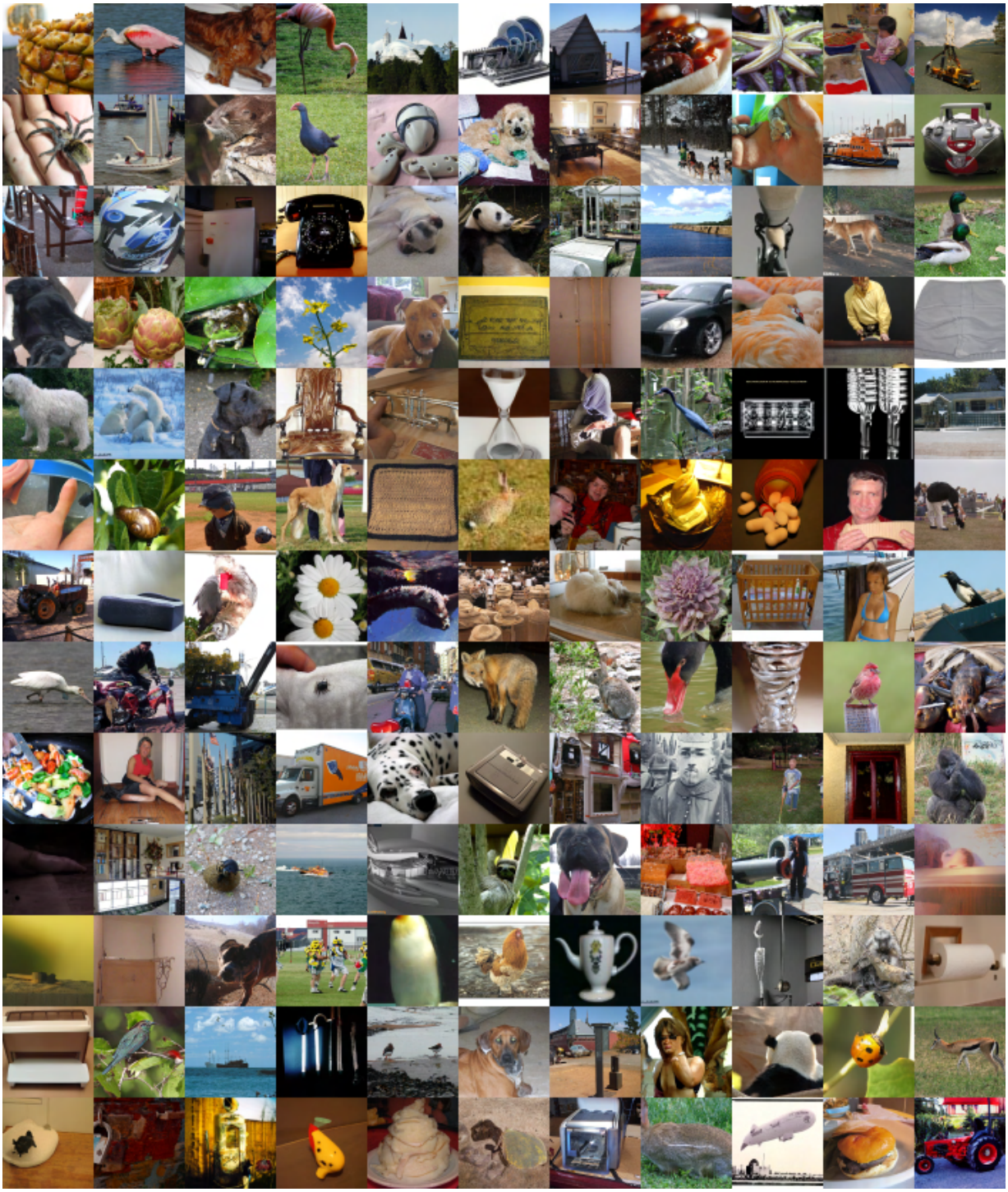


Figure 14. 1000-step samples from DDDM model trained on ImageNet 64x64

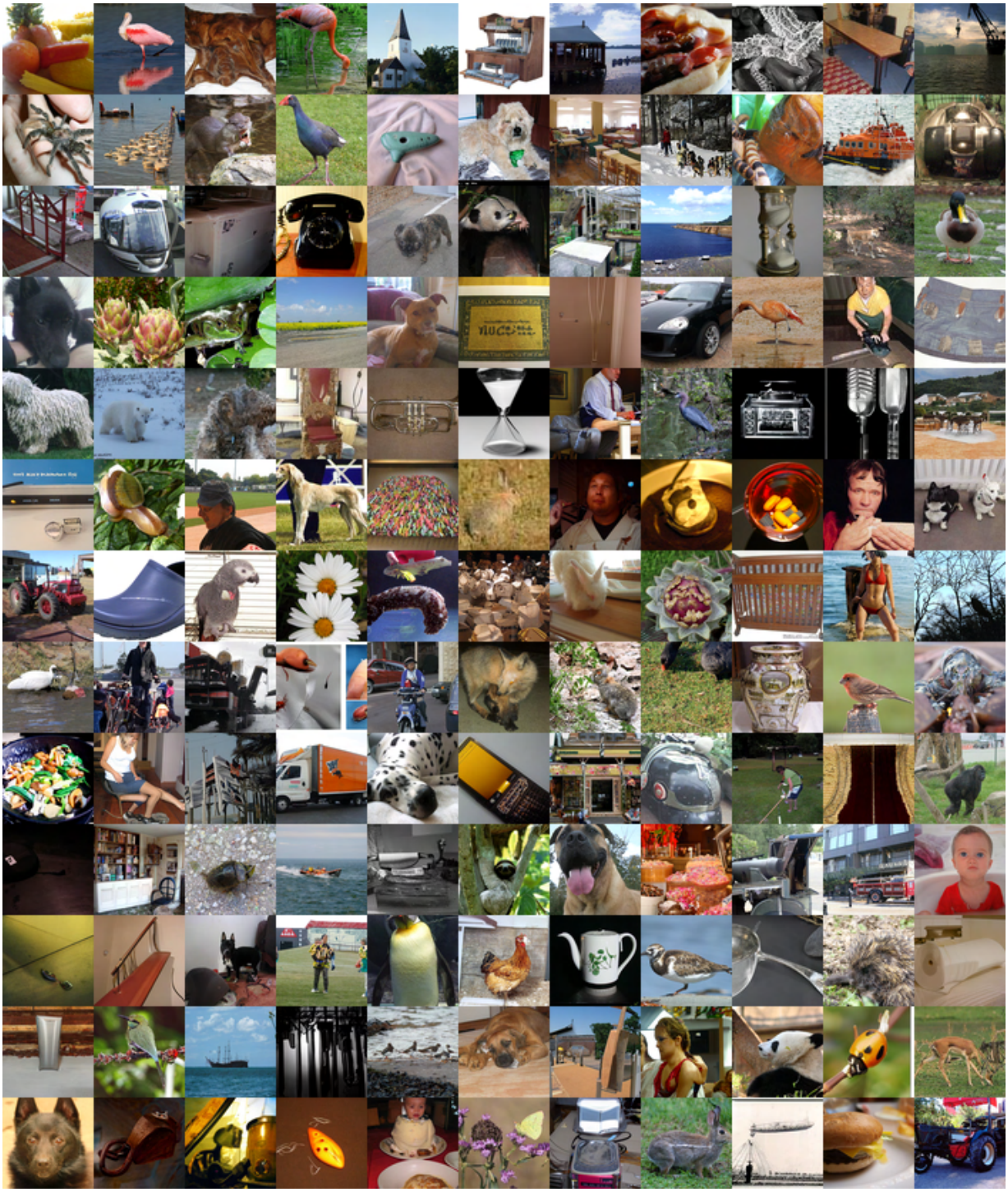


Figure 15. one-step samples from DDDM-deep model trained on ImageNet 64x64

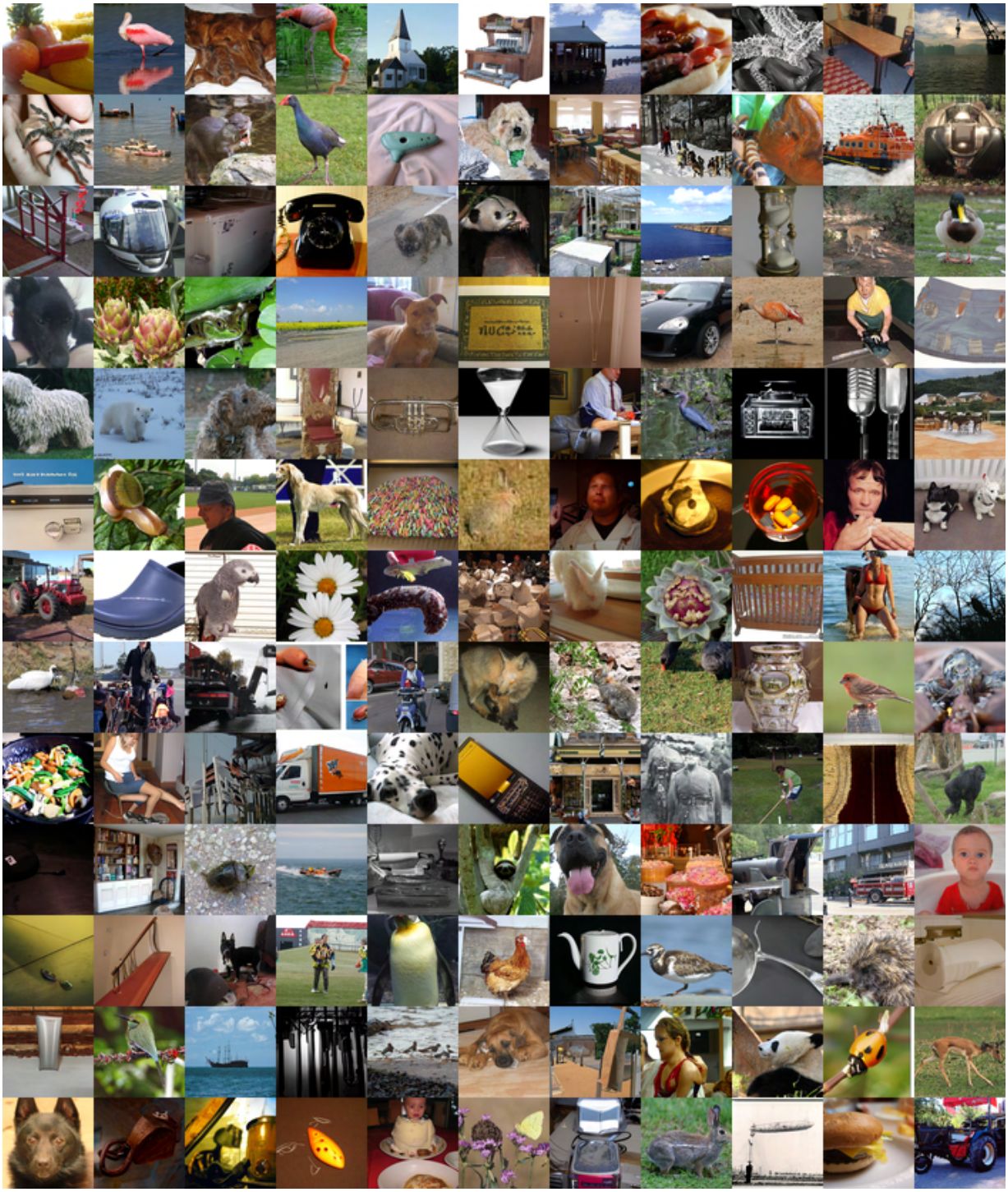


Figure 16. two-step samples from DDDM-deep model trained on ImageNet 64x64

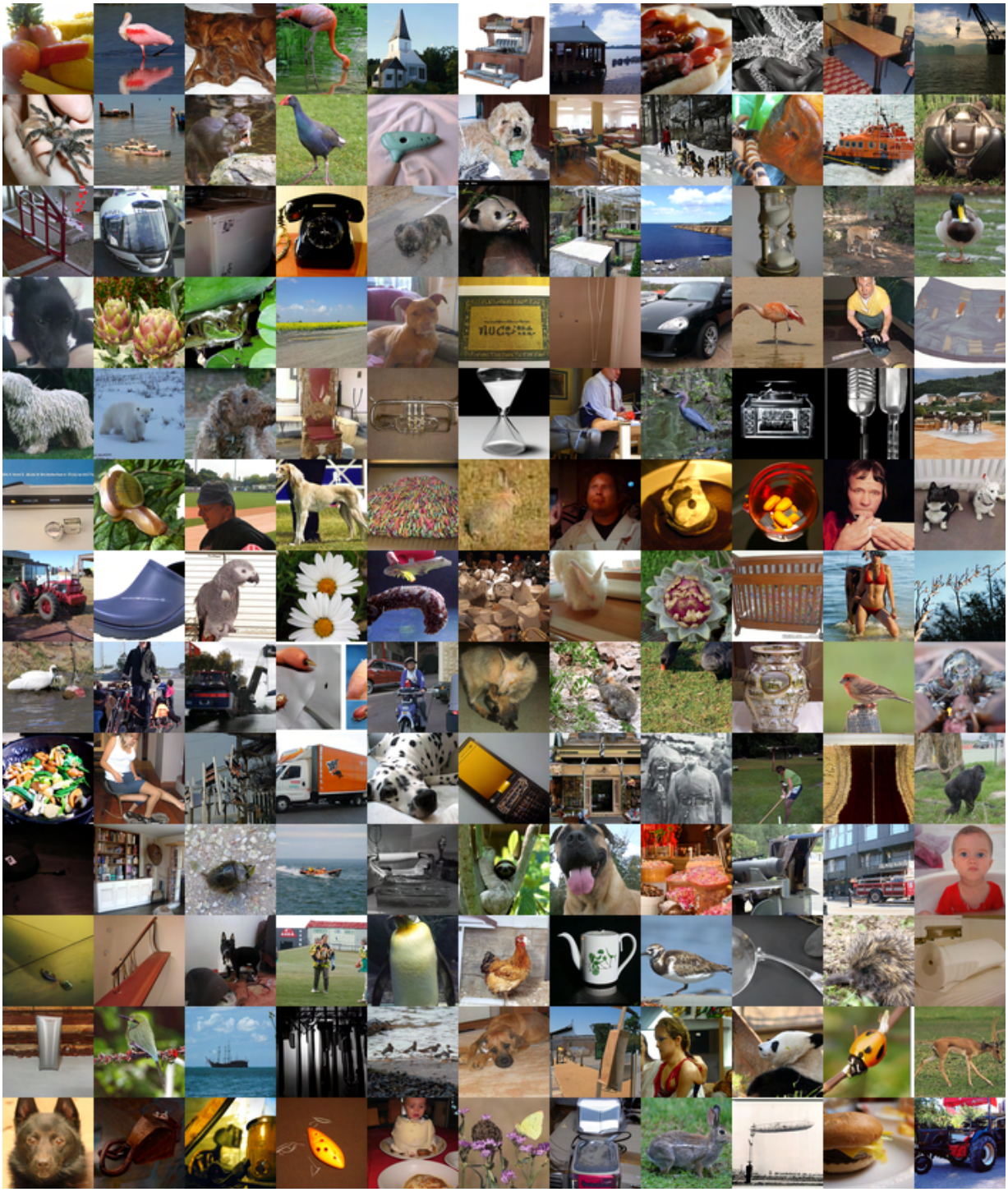


Figure 17. 1000-step samples from DDDM-deep model trained on ImageNet 64x64