Language Diffusion Models

A Survey

Tyler Crosse, May 2025

0 - Introduction

1 - Preliminaries - Generative Models
2 - Preliminaries - Generative Model Taxonomy
3 - Diffusion Model Foundations
4 - Language Diffusion Models

What is a diffusion model?

Traditional autoregressive language models generate text one word – or token – at a time. This sequential process can be slow, and limit the quality and coherence of the output.

Diffusion models work differently. Instead of predicting text directly, they learn to generate outputs by refining noise, step-by-step. This means they can iterate on a solution very quickly and error correct during the generation process. This helps them excel at tasks like editing, including in the context of math and code.

Capabilities

Rapid response

Generates content significantly faster than even our fastest model so far.

More coherent text

Generates entire blocks of tokens at once, meaning it responds more coherently to a user's prompt than autoregressive models.

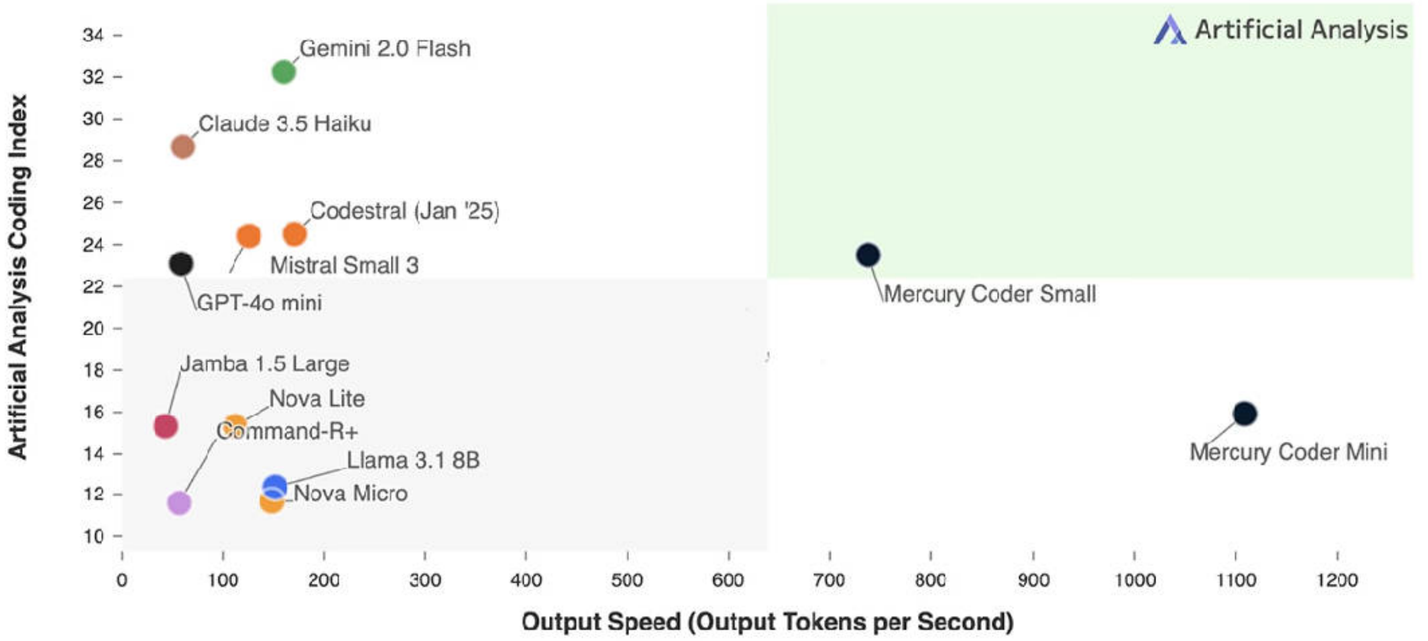
Iterative refinement

Corrects errors during generation for more consistent outputs.

Benchmarks

Gemini Diffusion's external benchmark performance is comparable to much larger models, whilst also being faster.

Benchmark	GEMINI DIFFUSION	GEMINI 2.0 FLASH-LITE
Code LiveCodeBench (v6)	30.9%	28.5%
Code BigCodeBench	45.4%	45.8%
Code LBPP (v2)	56.8%	56.0%
Code SWE-Bench Verified*	22.9%	28.5%



	Mercury Coder Mini	Mercury Coder Small	Gemini 2.0 Flash-Lite	Claude 3.5 Haiku	GPT-4o Mini	Qwen 2.5 Coder 7B	DeepSeek Coder V2 Lite
HumanEval	88.0	90.0	90.0	86.0	88.0	90.0	92.1
MBPP	77.1	76.6	75.0	78.0	74.6	80.0	81.0
EvalPlus	78.6	80.4	77.3	75.1	78.5	79.3	82.1
MultiPL-E	74.1	76.2	79.5	72.3	72.0	75.3	79.1
LiveCodeBench	17.0	25.0	18.0	31.0	23.0	9.0	37.8
BigCodeBench	42.0	45.5	44.4	45.4	46.8	41.4	50.0
Fill-in-the-Middle	82.2	84.8	60.1	45.5	60.9	56.1	46.9

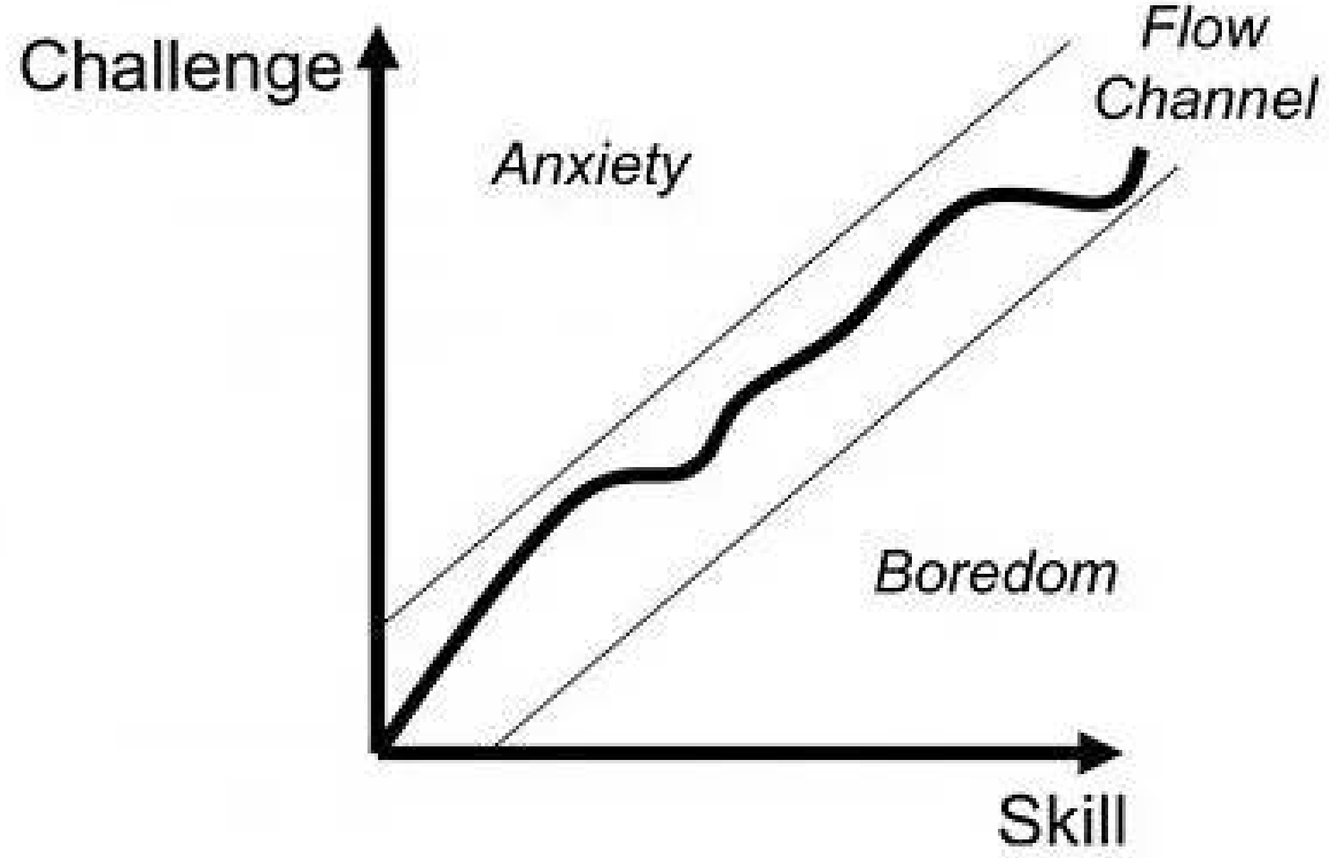
https://www.inceptionlabs.ai/introducing-mercury

Tyler Crosse

About me

- MSCS at Georgia Tech focus on ML + Systems (Expected May 2026)
- 5 Years Knowable, Legal Contract Analytics
  - Data Ingestion & Processing
  - Information Retrieval & Presentation
- 2 Years Rosetta Stone
  - Experiment tracking and iteration
- BS Biomedical Engineering

https://www.linkedin.com/in/tylercrosse

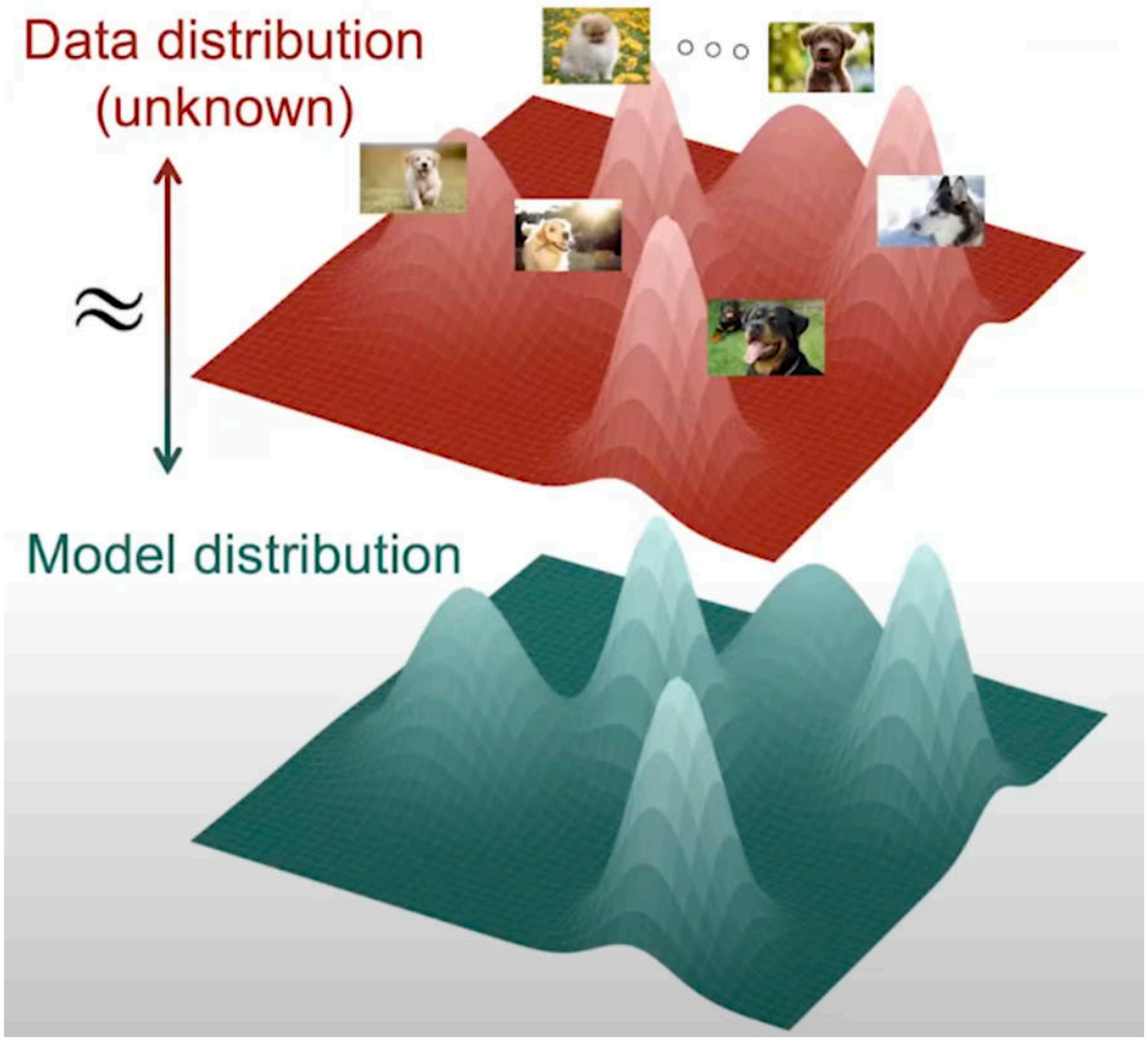


"Flow" concept by Mihaly Csikszentmihalyi. Drawn by Senia Maymin.

0 - Introduction

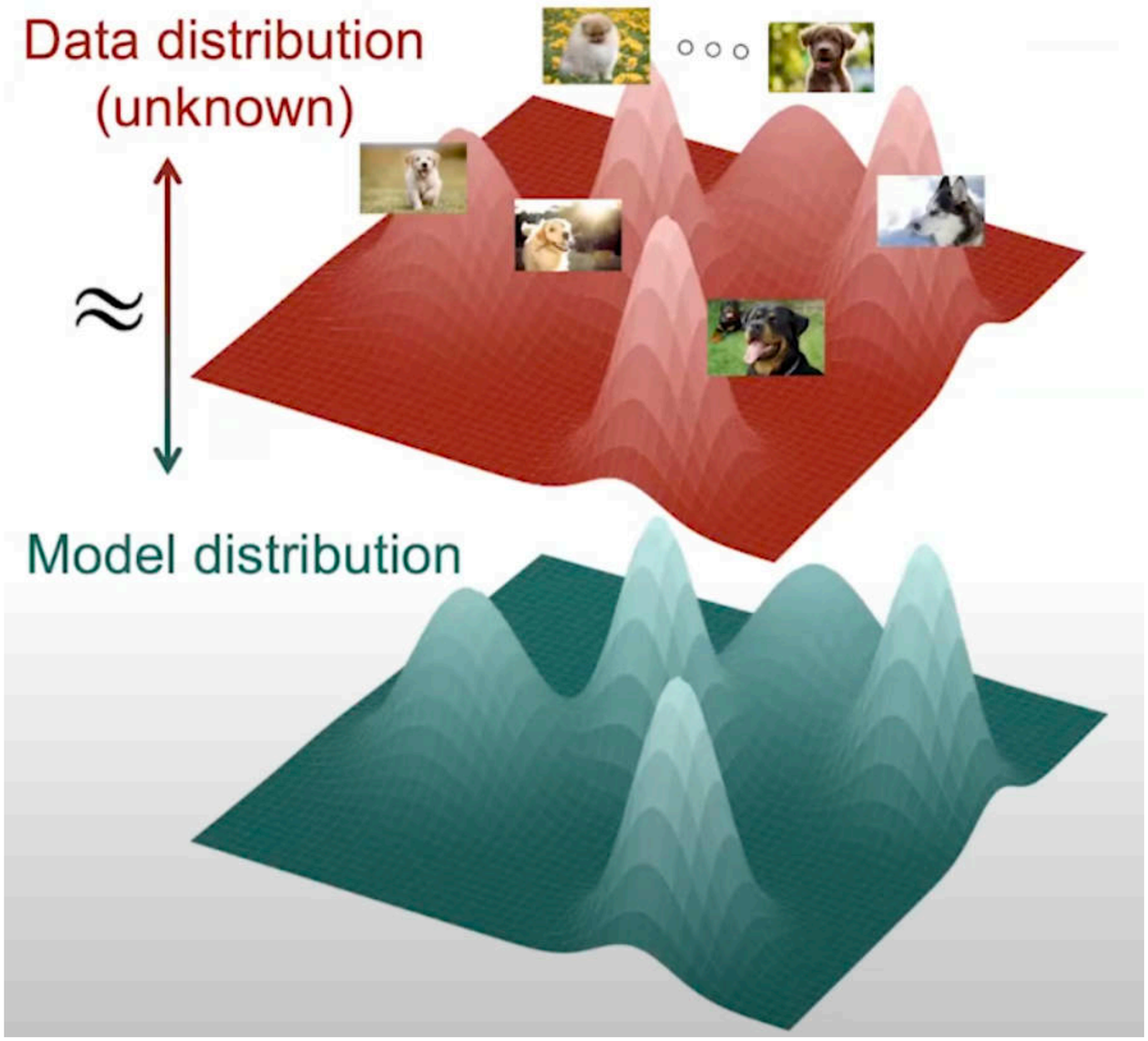
1 - Preliminaries - Generative Models

2 - Preliminaries - Generative Model Taxonomy
3 - Diffusion Model Foundations
4 - Language Diffusion Models



A generative model aims to learn the underlying probability distribution $P(data)$ of a given dataset. Once learned, it can be used to generate new data samples that resemble the original dataset.

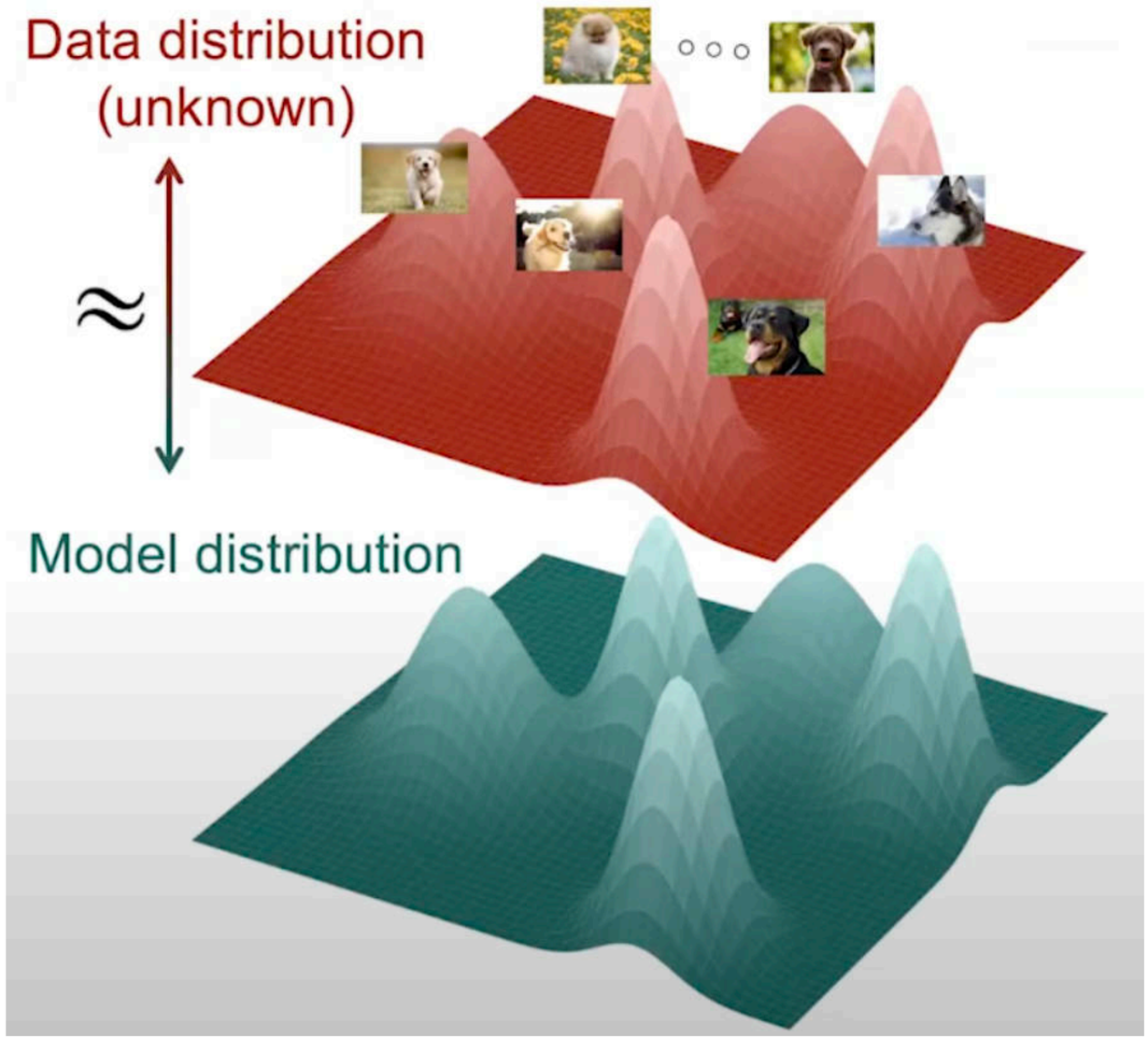
Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q



$p_{\text{data}}(x)$ is the true probability of x, based on the actual (but unknown) data distribution.

$p_\theta(x)$ describes the model's learned probability distribution over x, parameterized by θ. During training, $p_\theta(x)$ is treated as a **likelihood** function, measuring how likely the observed data x is, under the current parameters θ.

Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q

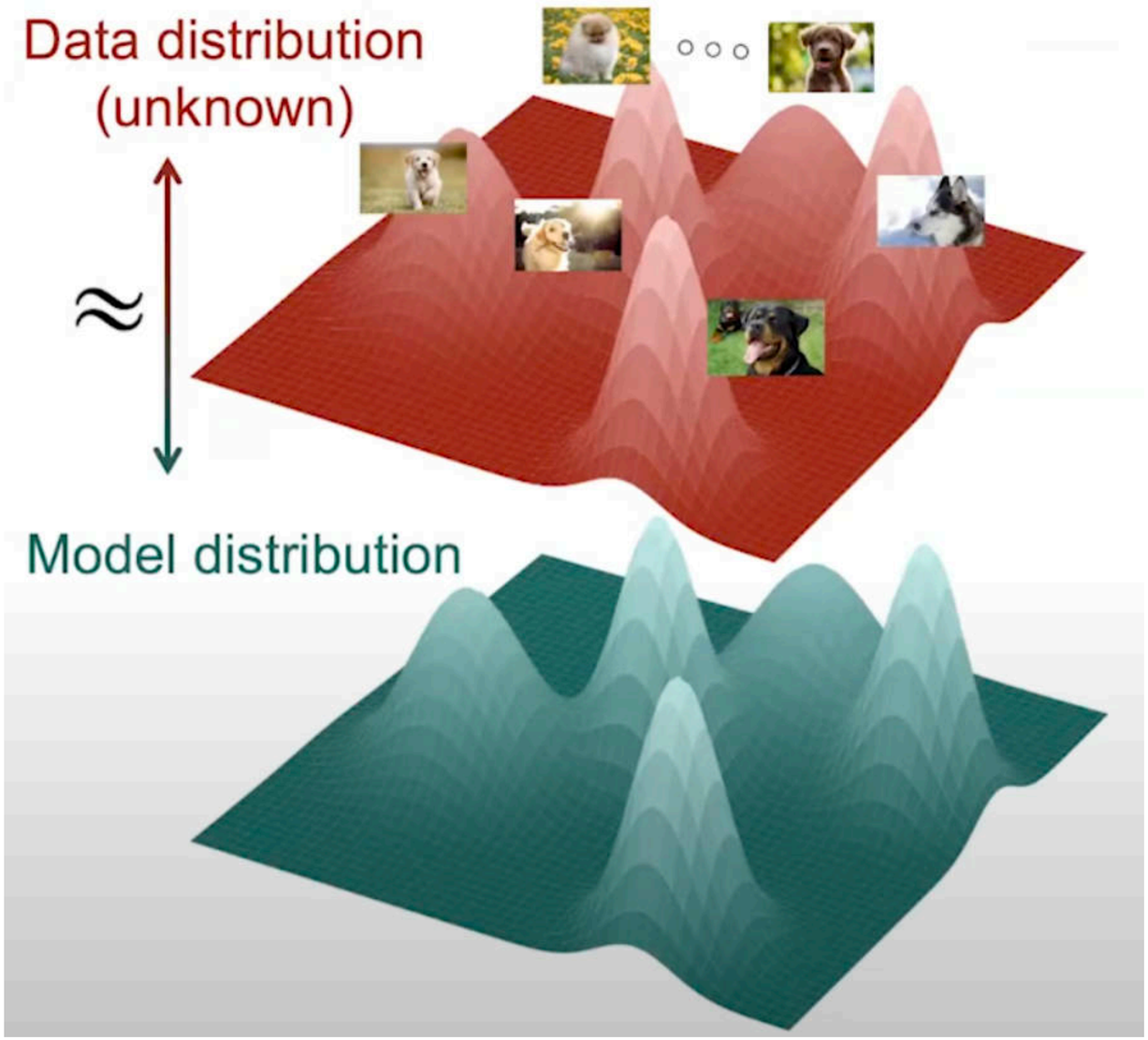


Optimization objective:

Find model parameters θ to minimize divergence between data distribution and model distribution

$$\min_{\theta} \mathrm{KL}\left(p_{\text{data}}(x) \| p_{\theta}(x)\right)$$

Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q



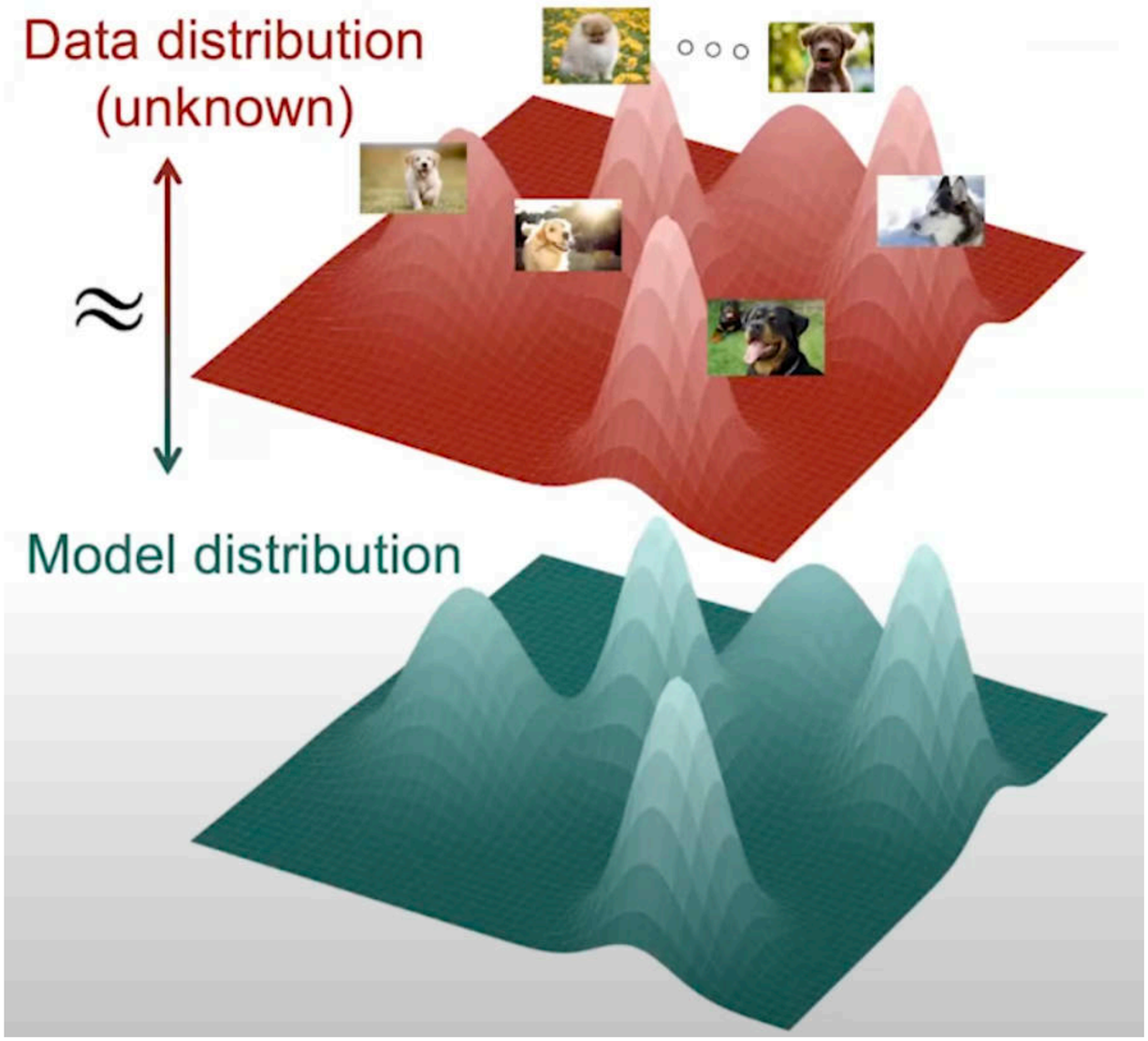
Optimization objective:

Find model parameters θ that maximize the likelihood of estimating the (expected) data

Maximum Likelihood Estimation (MLE)

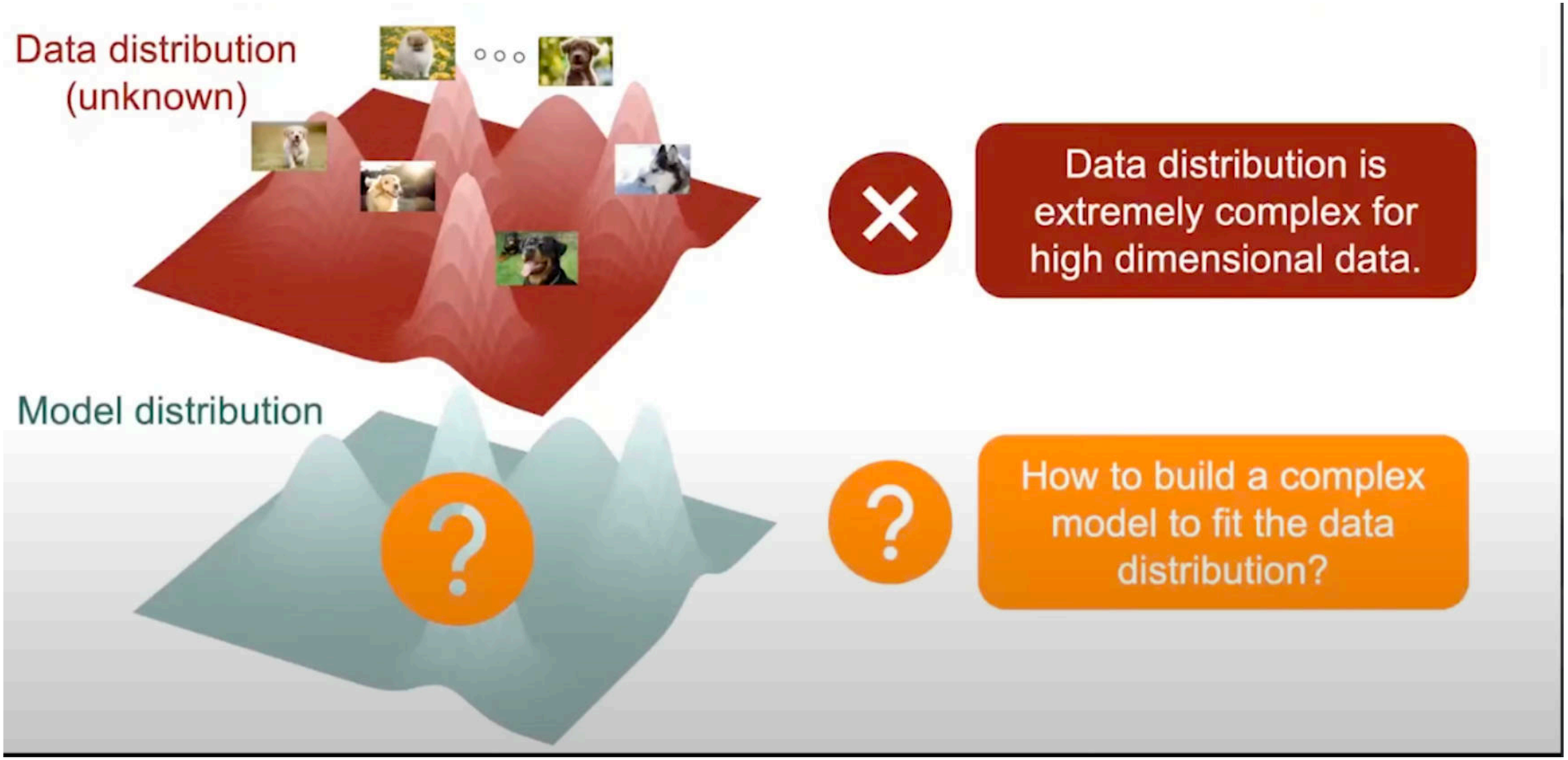
$$\max_{\theta} \mathbb{E}_{p_{\text{data}}(x)} \left[\log p_\theta(x) \right]$$

Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q



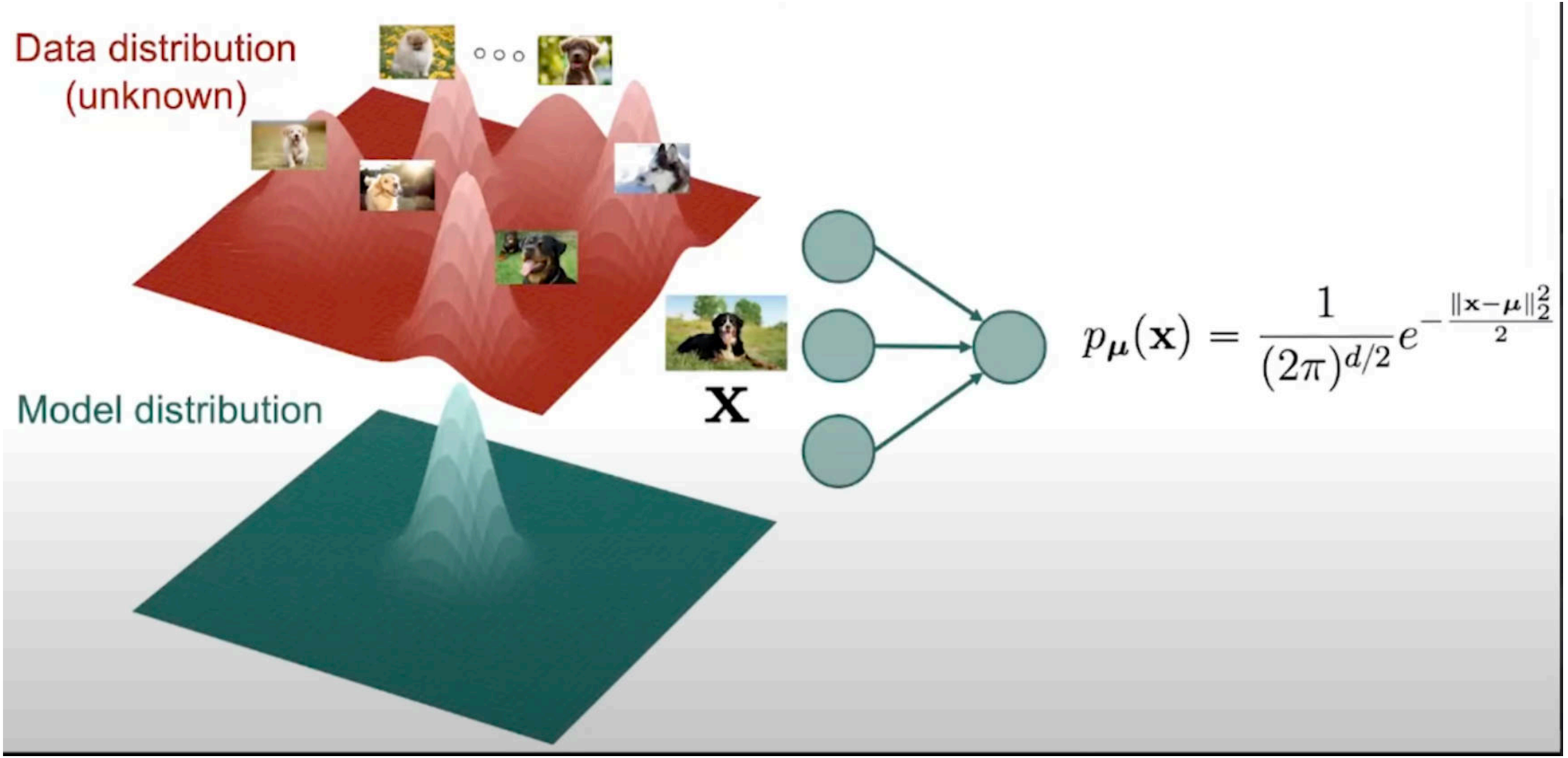
$$\underbrace{\max_{\theta} \mathbb{E}_{p_{\text{data}}(x)}\left[\log p_\theta(x)\right]}_{MLE} \Leftrightarrow \underbrace{\min_{\theta} \mathrm{KL}\left(p_{\text{data}}(x) \| p_\theta(x)\right)}_{\text{Data / Model Divergence}}$$

Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q



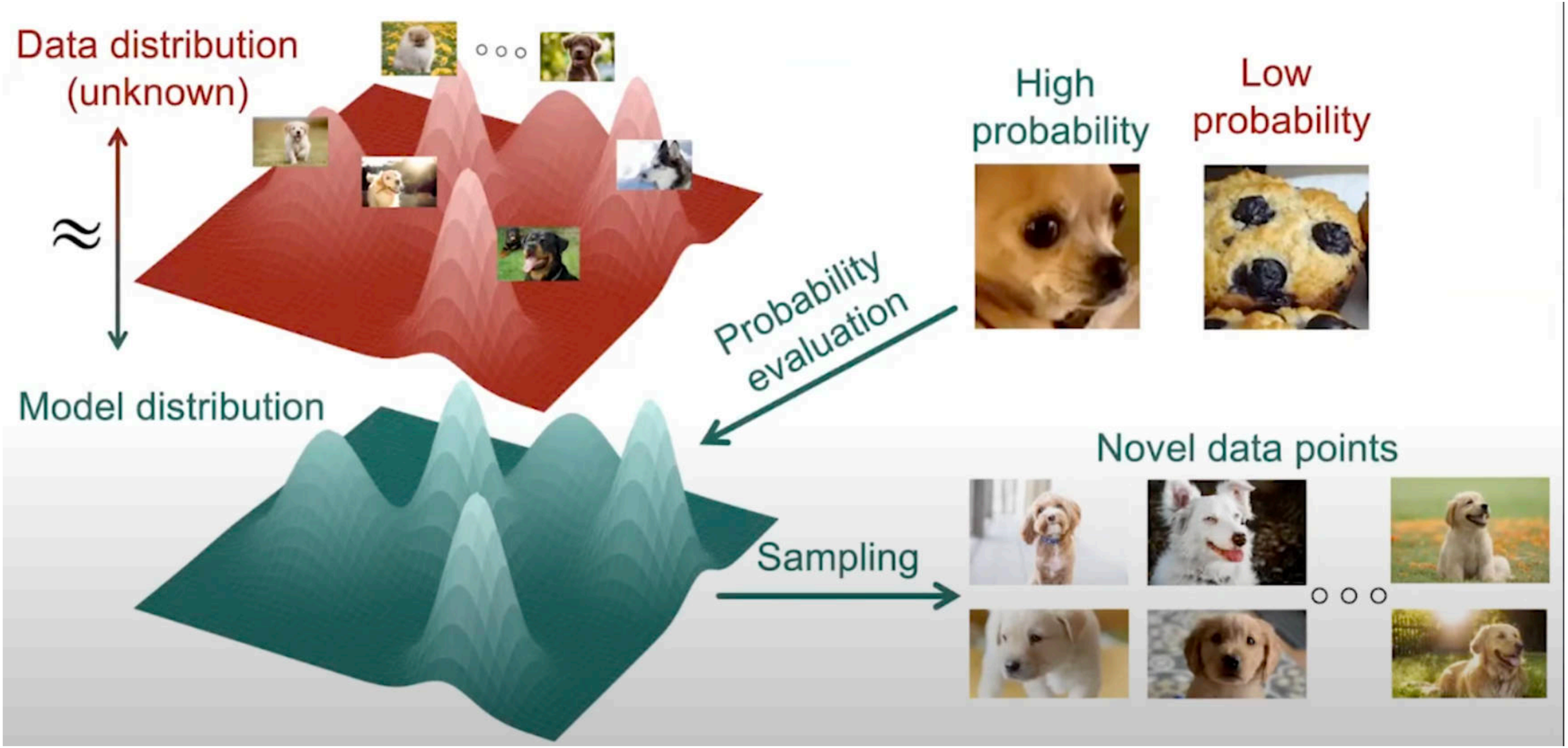
Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q

Gaussian Model - Not sufficiently expressive

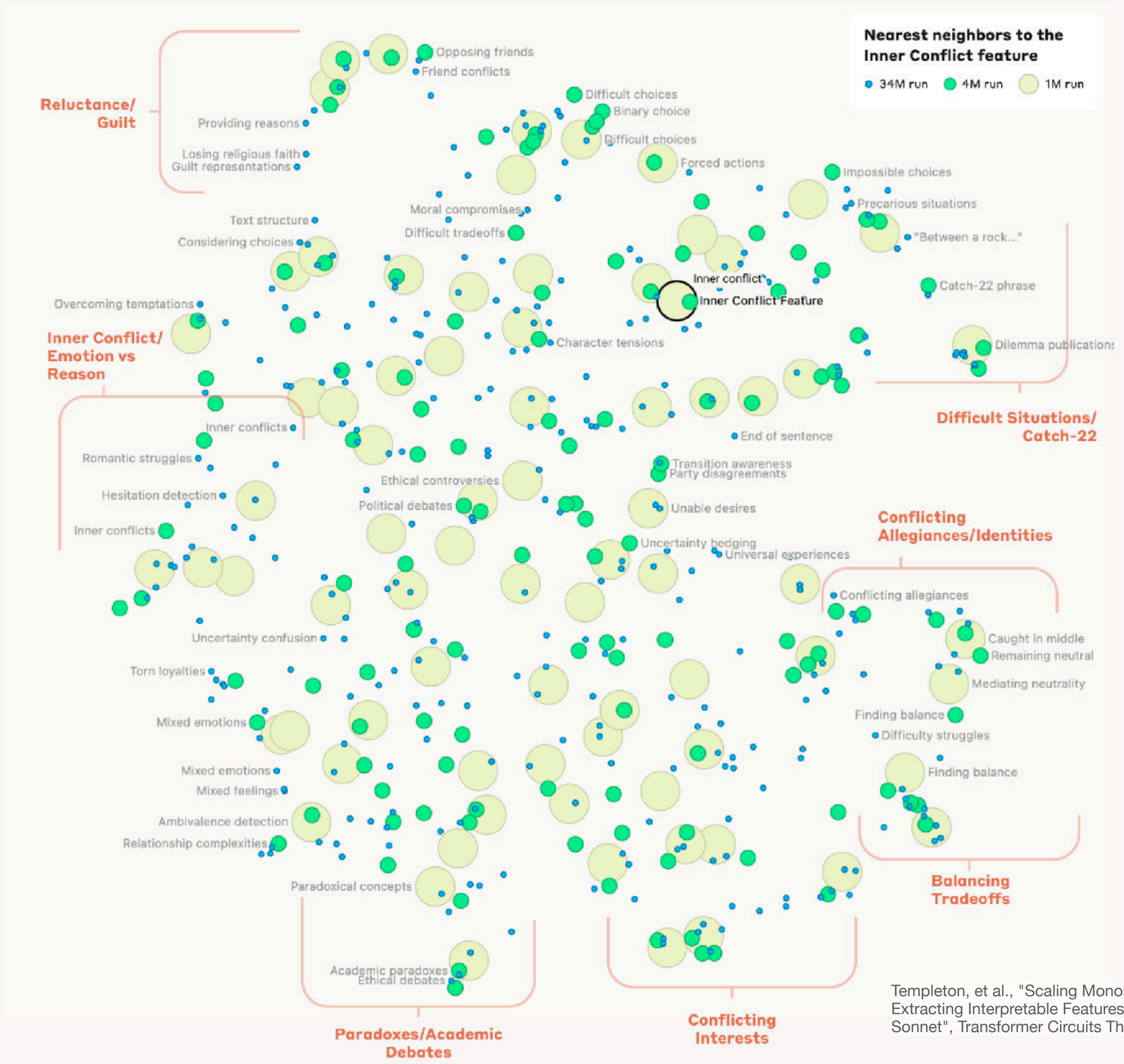


Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q

Deep Neural Network - Universal Approximation Theorem



Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q



Templeton, et al., "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet", Transformer Circuits Thread, 2024.

0 - Introduction
1 - Preliminaries - Generative Models

2 - Preliminaries - Generative Model Taxonomy

3 - Diffusion Model Foundations
4 - Language Diffusion Models

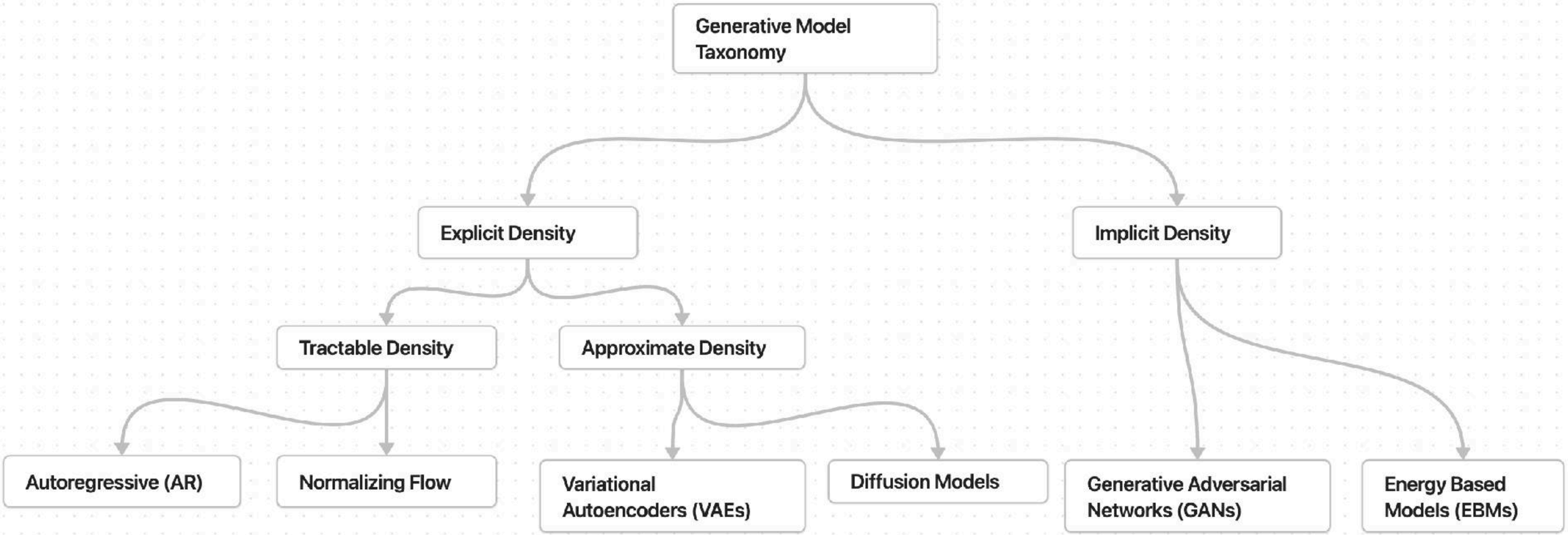


Figure inspired by Ian Goodfellow's, Tutorial on Generative Adversarial /networks, 2017

Generative Model Taxonmy

A generative model aims to learn the underlying probability distribution $P(data)$ of a given dataset. Once learned, it can be used to generate new data samples that resemble the original dataset.

Explicit Density

These models explicitly define a probability density function $p_\theta(x)$ (or $p_{\text{model}}(x)$) and aim to maximize the likelihood of the training data under this model.

Implicit Density

These models do not explicitly define the density function $p_\theta(x)$. Instead, they provide a mechanism to sample from the distribution $P(data)$ directly, often through a learned transformation.

Explicit Density

These models explicitly define a probability density function $p_\theta(x)$ (or $p_{\text{model}}(x)$) and aim to maximize the likelihood of the training data under this model.

Tractable Density Models

The model $p_\theta(x)$ can be directly computed and optimized (e.g., via Maximum Likelihood Estimation - MLE).

Approximate Explicit Density Models

The model $p_\theta(x)$ is defined, but it's intractable to compute or optimize directly. Instead, approximations or bounds are used.

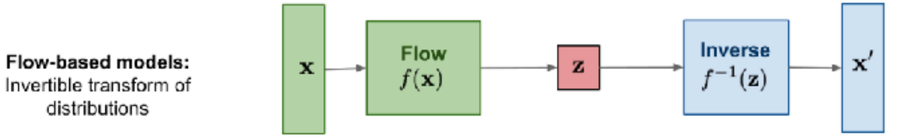
Tractable Density Models

The model $p_\theta(x)$ can be directly computed and optimized (e.g., via Maximum Likelihood Estimation - MLE).

Autoregressive Models

- **Core Idea:** Decompose the joint probability distribution $P(x)$ over a high-dimensional vector x into a product of conditional probabilities using the chain rule: $P(x) = \prod P(x_i|x_1, \ldots, x_{i-1})$.
- **How it works:** Each dimension (or token, pixel, etc.) is generated sequentially, conditioned on previously generated dimensions. $\mathcal{L} = -\sum_i \log p_\theta(x_i \mid x_{<i})$
- **Examples:**
  - *Text:* GPT (Generative Pre-trained Transformer), LLaMA, PaLM, Transformer-XL, RNNs/LSTMs for language modeling.
  - *Images:* PixelRNN, PixelCNN, ImageGPT.
  - *Audio:* WaveNet, SampleRNN.

Normalizing Flow Models



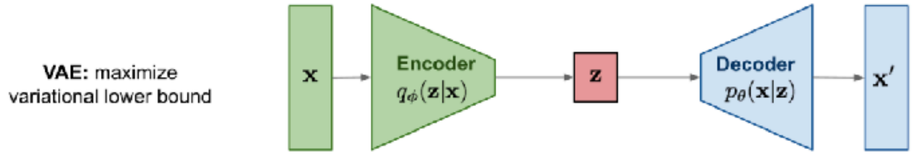
- **Core Idea:** Transform a simple base probability distribution (e.g., Gaussian) into a complex target distribution using a series of invertible and differentiable transformations.
- **How it Works:** $P(x)$ is computed using the change of variables formula, involving the Jacobian determinant of the transformations.
- **Examples:** NICE, RealNVP, Glow, Masked Autoregressive Flow (MAF).

Figure from What are Diffusion Models?, Lilian Wang, 2021

Approximate Explicit Density Models

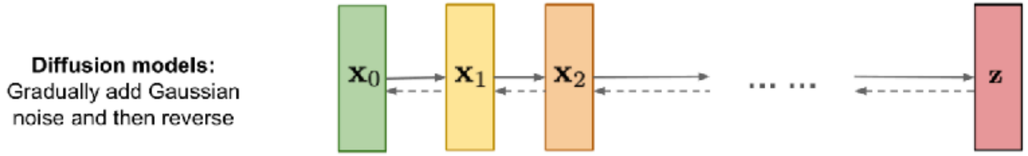
The model $p_\theta(x)$ is defined, but it's intractable to compute or optimize directly. Instead, approximations or bounds are used.

Variational Autoencoders (VAEs)



- **Core Idea:** Learn a latent variable model $p(x, z) = p(x|z)p(z)$. Since the true posterior $p(z|x)$ is intractable, an approximate posterior $q(z|x)$ (the encoder) is learned alongside the generative model $p(x|z)$ (the decoder).
- **Examples:** Standard VAE, β-VAE, VQ-VAE (uses discrete latent codes).

Diffusion Models



(Denoising Diffusion Probabilistic Models - DDPMs, Score-Based Generative Models)

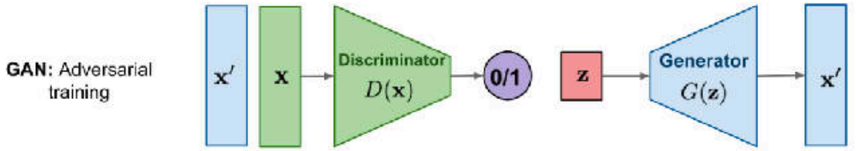
- **Core Idea:** Define a forward "diffusion" process that gradually adds noise to data until it becomes pure noise. Then, learn a reverse "denoising" process that can transform noise back into a data sample.
- **Examples:** DDPM, DDIM, Improved DDPM, Latent Diffusion Models (e.g., Stable Diffusion), Imagen, DALL-E 2 (uses diffusion for prior and decoder).

Figure from What are Diffusion Models?, Lilian Wang, 2021

Implicit Density

These models do not explicitly define the density function $p_\theta(x)$. Instead, they provide a mechanism to sample from the distribution $P(data)$ directly, often through a learned transformation.

Generative Adversarial Networks (GANs)



- **Core Idea:** A minimax game between two neural networks: a Generator (G) and a Discriminator (D).
- **How it works:**
  - Generator (G): Takes random noise z as input and tries to generate samples $G(z)$ that look real.
  - Discriminator (D): Tries to distinguish between real data samples and fake samples generated by G.
  - G is trained to fool D, while D is trained to get better at discriminating. They are trained adversarially.
- **Examples:** Original GAN, DCGAN, WGAN, StyleGAN, BigGAN,

Energy Based Models (EBMs)

- Loss: Contrastive divergence or score-based

$\mathcal{L} = \mathbb{E}x \sim p\text{data}[E_\theta(x)] - \mathbb{E}x \sim p\theta[E_\theta(x)]$

- Often approximated using Monte Carlo Markov Chains (MCMC), Langevin dynamics

Figure from What are Diffusion Models?, Lilian Wang, 2021

Aa Milestone	# Year	Model family
Shannon-style Markov text generators	1959	Markov n-gram
EM-trained Gaussian Mixture OPEN	1995	Mixture model
Latent Dirichlet Allocation (LDA)	2003	Probabilistic topic model
Deep Belief Networks & RBMs	2006	Boltzmann Energy-based
Variational Autoencoders (VAE)	2014	VAE
Generative Adversarial Networks (GAN)	2014	GAN
DCGAN	2015	GAN
Pix2Pix / cGAN	2016	GAN
PixelRNN	2016	AR
Transformer (“Attention is All You Need”)	2017	AR
BERT / GPT-1	2018	AR
GLOW	2018	Normalizing Flow
GPT-2	2019	AR
StyleGAN	2019	GAN
DDPM (Denoising Diffusion)	2020	Diffusion
CLIP + DALL·E 1	2021	VQ-VAE
VQ-GAN / VQ-VAE-2	2021	VQ-VAE
Stable Diffusion / Imagen / DALL·E 2	2022	VQ-VAE Diffusion
GPT-4 / PaLM-2 / Claude	2023	AR
Diffusion Transformer (DiT)	2023	Diffusion
Consistency Models	2023	Diffusion
Sora	2024	Diffusion
Runway Gen-2	2024	Diffusion
LLaDA / Diffusion Language Models	2025	Diffusion
MMaDA	2025	Diffusion

0 - Introduction
1 - Preliminaries - Generative Models
2 - Preliminaries - Generative Model Taxonomy

3 - Diffusion Model Foundations

4 - Language Diffusion Models

Extracting and Composing Robust Features with Denoising Autoencoders

Pascal Vincent VINCENTP@IRO.UMONTREAL.CA
Hugo Larochelle LAROCHEH@IRO.UMONTREAL.CA
Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Pierre-Antoine Manzagol MANZAGOP@IRO.UMONTREAL.CA
Université de Montréal, Dept. IRO, CP 6128, Succ. Centre-Ville, Montral, Qubec, H3C 3J7, Canada

Abstract

Previous work has shown that the difficulties in learning deep generative or discriminative models can be overcome by an initial unsupervised learning step that maps inputs to useful intermediate representations. We introduce and motivate a new training principle for unsupervised learning of a representation based on the idea of making the learned representations robust to partial corruption of the input pattern. This approach can be used to train autoencoders, and these denoising autoencoders can be stacked to initialize deep architectures. The algorithm can be motivated from a manifold learning and information theoretic perspective or from a generative model perspective. Comparative experiments clearly show the surprising advantage of corrupting the input of autoencoders on a pattern classification benchmark suite.

1. Introduction

Recent theoretical studies indicate that deep architectures (Bengio & Le Cun, 2007; Bengio, 2007) may be needed to *efficiently* model complex distributions and achieve better generalization performance on challenging recognition tasks. The belief that additional levels of functional composition will yield increased representational and modeling power is not new (McClelland et al., 1986; Hinton, 1989; Utgoff & Stracuzzi, 2002). However, in practice, learning in deep architectures has proven to be difficult. One needs only to ponder the difficult problem of inference in deep directed graphical models, due to "explaining away". Also looking back at the history of multi-layer neural networks, their difficult optimization (Bengio et al., 2007; Bengio, 2007) has long prevented reaping the expected benefits of going beyond one or two hidden layers. However this situation has recently changed with the successful approach of (Hinton et al., 2006; Hinton & Salakhutdinov, 2006; Bengio et al., 2007; Ranzato et al., 2007; Lee et al., 2008) for training Deep Belief Networks and stacked autoencoders.

One key ingredient to this success appears to be the use of an unsupervised training criterion to perform a layer-by-layer initialization: each layer is at first trained to produce a higher level (hidden) representation of the observed patterns, based on the representation it receives as input from the layer below, by optimizing a local unsupervised criterion. Each level produces a representation of the input pattern that is more abstract than the previous level's, because it is obtained by composing more operations. This initialization yields a starting point, from which a global fine-tuning of the model's parameters is then performed using another training criterion appropriate for the task at hand. This technique has been shown empirically to avoid getting stuck in the kind of poor solutions one typically reaches with random initializations. While unsupervised learning of a mapping that produces "good" intermediate representations of the input pattern seems to be key, little is understood regarding what constitutes "good" representations for initializing deep architectures, or what explicit criteria may guide learning such representations. We know of only a few algorithms that seem to work well for this purpose: Restricted Boltzmann Machines (RBMs) trained with contrastive divergence on one hand, and various types of autoencoders on the other.

The present research begins with the question of what

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

Extracting and Composing Robust Features with Denoising Autoencoders, Vincent, Larochelle, Bengio, Manzagol, 2008

"destruction": for each input x, a fixed number vd of components are chosen at random, and their value is forced to 0, while the others are left untouched. All information about the chosen components is thus removed from that particuler input pattern, and the autoencoder will be trained to "fill-in" these artificially introduced "blanks".

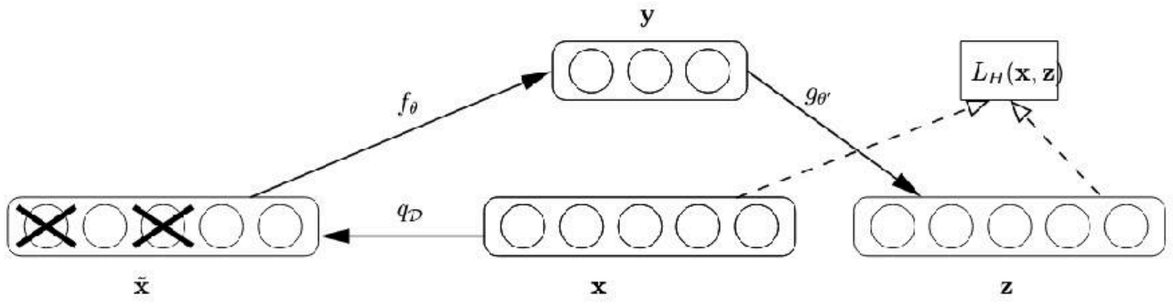


Figure 1. An example $\mathbf{x}$ is corrupted to $\tilde{\mathbf{x}}$. The autoencoder then maps it to $\mathbf{y}$ and attempts to reconstruct $\mathbf{x}$.

Generalized Denoising Auto-Encoders as Generative Models

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent
Département d'informatique et recherche opérationnelle
Université de Montréal

Abstract

Recent work has shown how denoising and contractive autoencoders implicitly capture the structure of the data-generating density, in the case where the corruption noise is Gaussian, the reconstruction error is the squared error, and the data is continuous-valued. This has led to various proposals for sampling from this implicitly learned density function, using Langevin and Metropolis-Hastings MCMC. However, it remained unclear how to connect the training procedure of regularized auto-encoders to the implicit estimation of the underlying data-generating distribution when the data are discrete, or using other forms of corruption process and reconstruction errors. Another issue is the mathematical justification which is only valid in the limit of small corruption noise. We propose here a different attack on the problem, which deals with all these issues: arbitrary (but noisy enough) corruption, arbitrary reconstruction loss (seen as a log-likelihood), handling both discrete and continuous-valued variables, and removing the bias due to non-infinitesimal corruption noise (or non-infinitesimal contractive penalty).

1 Introduction

Auto-encoders learn an encoder function from input to representation and a decoder function back from representation to input space, such that the reconstruction (composition of encoder and decoder) is good for training examples. Regularized auto-encoders also involve some form of regularization that prevents the auto-encoder from simply learning the identity function, so that reconstruction error will be low at training examples (and hopefully at test examples) but high in general. Different variants of auto-encoders and sparse coding have been, along with RBMs, among the most successful building blocks in recent research in deep learning (Bengio *et al.*, 2013b). Whereas the usefulness of auto-encoder variants as feature learners for supervised learning can directly be assessed by performing supervised learning experiments with unsupervised pre-training, what has remained until recently rather unclear is the interpretation of these algorithms in the context of pure unsupervised learning, as devices to capture the salient structure of the input data distribution. Whereas the answer is clear for RBMs, it is less obvious for regularized auto-encoders. Do they completely characterize the input distribution or only some aspect of it? For example, clustering algorithms such as k-means only capture the modes of the distribution, while manifold learning algorithms characterize the low-dimensional regions where the density concentrates.

Some of the first ideas about the probabilistic interpretation of auto-encoders were proposed by Ranzato *et al.* (2008): they were viewed as approximating an energy function through the reconstruction error, i.e., being trained to have low reconstruction error at the training examples and high reconstruction error elsewhere (through the regularizer, e.g., sparsity or otherwise, which prevents the auto-encoder from learning the identity function). An important breakthrough then came, yielding a first formal probabilistic interpretation of regularized auto-encoders as models of the input distribution, with the work of Vincent (2011). This work showed that some denoising auto-encoders correspond to a Gaussian RBM and that minimizing the denoising reconstruction error (as a squared error) estimates the energy function through a regularized form of score matching, with the regularization disappearing as the amount of corruption noise goes to 0, and then converging to the same

Generalized Denoising Auto-Encoders as Generative Models, Bengio, Alain, Vincent, 2013-06-02

"We have proven that training a model to denoise is a way to implicitly estimate the underlying datagenerating process, and that a simple Markov chain that alternates sampling from the denoising model and from the corruption process converges to that estimator. This provides a means for generating data from any denoising auto-encoder"

Figure 4: *Successive samples generated by Markov chain associated with the trained denoising auto-encoders according to the plain sampling scheme (left) and walkback sampling scheme (right). There are less "spurious" samples with the walkback algorithm.*

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However, these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $\phi(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff–for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective[1].

1.1. Diffusion probabilistic models

We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

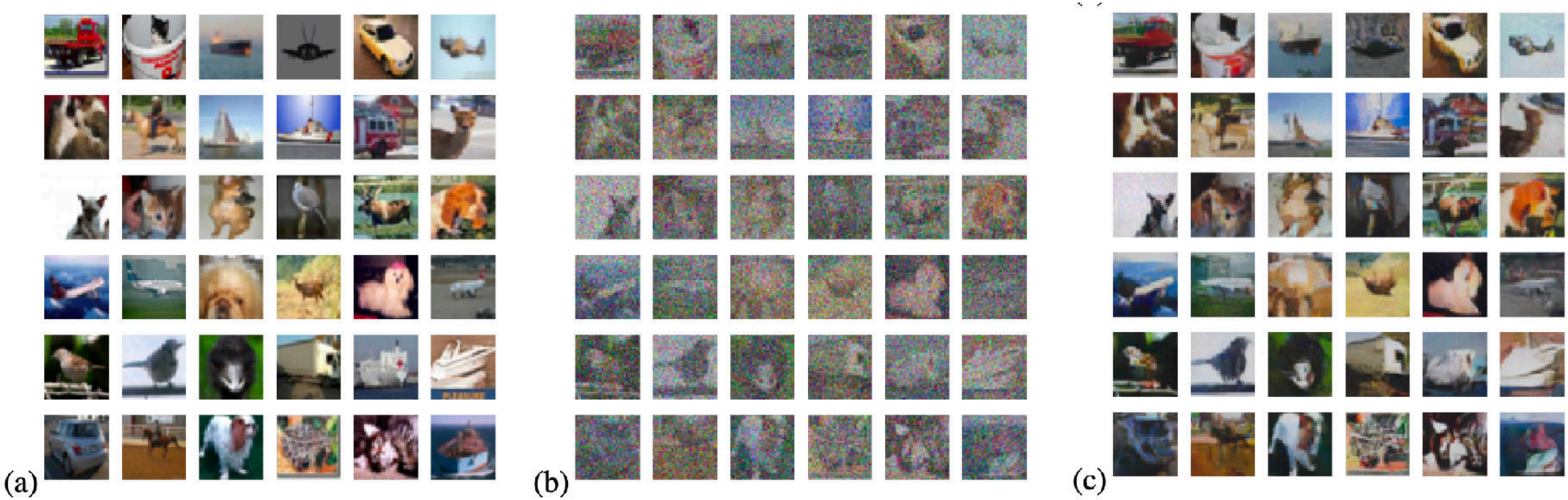
Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

[1] Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Sohl-Dickstein Et al. 2015-03-12

- **Thermodynamic Inspiration**: Generative modeling framed through nonequilibrium physics.
- **Reversed Diffusion**: Generate data by reversing a noise-adding diffusion process.
- **SDE Foundation**: Introduced stochastic differential equations to model forward/reverse dynamics.
- **Nonparametric Data Modeling**: Learn dynamics directly from noisy samples, not density estimates.

Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Sohl-Dickstein Et al. 2015-11-18

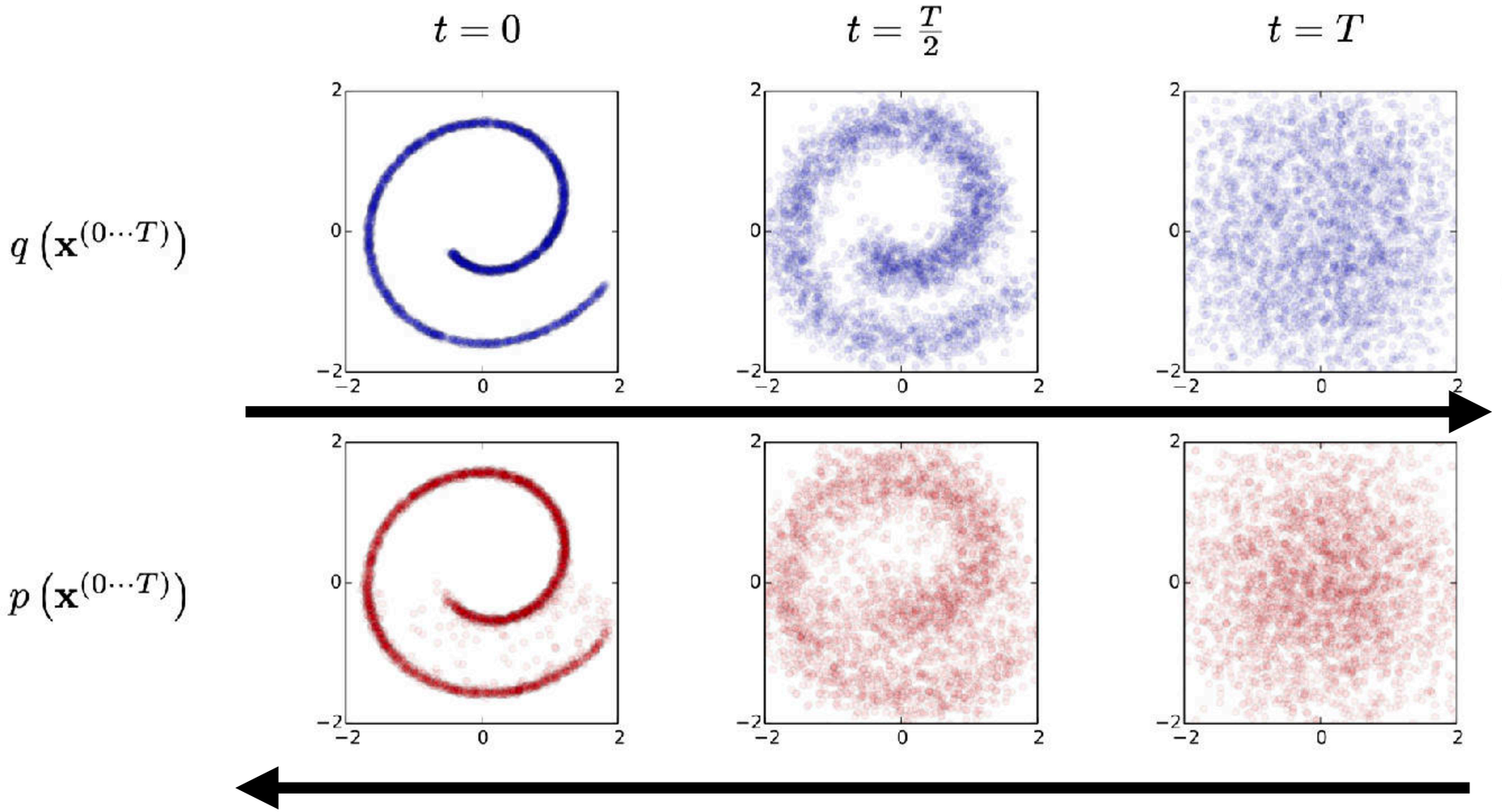


CIFAR-10 holdout images

Corrupted with Gaussian noise

Denoised images, generated by sampling from the posterior distribution over denoised images conditioned on the images in (b)

Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Sohl-Dickstein Et al. 2015-11-18



forward diffusion
gradually add noise

reverse diffusion
recover the data distribution

2015 - 2019

- BigGAN and StyleGAN2 greatly advanced image generation but leaned toward realism over diversity due to their mode-seeking behavior.

- GANs struggled to generalize impressive domain-specific results (e.g., realistic faces) to more diverse datasets.

- VQ-VAE 2 and VQGAN promoted a two-stage generative modelling approach:
  - Compress images into discrete one-dimensional sequences.
  - Use autoregressive models to predict these sequences step-by-step.

- Although the concept originated with the original VQ-VAE, VQ-VAE 2 and VQGAN reinforced its value for large-scale, diverse data generation.

Improved Techniques for Training Score-Based Generative Models

Yang Song
Computer Science Department
Stanford University
yangsong@cs.stanford.edu

Stefano Ermon
Computer Science Department
Stanford University
ermon@cs.stanford.edu

Abstract

Score-based generative models can produce high quality image samples comparable to GANs, without requiring adversarial optimization. However, existing training procedures are limited to images of low resolution (typically below 32×32), and can be unstable under some settings. We provide a new theoretical analysis of learning and sampling from score-based models in high dimensional spaces, explaining existing failure modes and motivating new solutions that generalize across datasets. To enhance stability, we also propose to maintain an exponential moving average of model weights. With these improvements, we can scale score-based generative models to various image datasets, with diverse resolutions ranging from 64×64 to 256×256. Our score-based models can generate high-fidelity samples that rival best-in-class GANs on various image datasets, including CelebA, FFHQ, and several LSUN categories.

1 Introduction

Score-based generative models [1] represent probability distributions through score—a vector field pointing in the direction where the likelihood of data increases most rapidly. Remarkably, these score functions can be learned from data without requiring adversarial optimization, and can produce realistic image samples that rival GANs on simple datasets such as CIFAR-10 [2].

Despite this success, existing score-based generative models only work on low resolution images (32×32) due to several limiting factors. First, the score function is learned via denoising score matching [3, 4, 5]. Intuitively, this means a neural network (named the *score network*) is trained to denoise images blurred with Gaussian noise. A key insight from [1] is to perturb the data using *multiple* noise scales so that the score network captures both coarse and fine-grained image features. However, it is an open question how these noise scales should be chosen. The recommended settings in [1] work well for 32×32 images, but perform poorly when the resolution gets higher. Second, samples are generated by running Langevin dynamics [6, 7]. This method starts from white noise and progressively denoises it into an image using the score network. This procedure, however, might fail or take an extremely long time to converge when used in high-dimensions and with a necessarily imperfect (learned) score network.

We propose a set of techniques to scale score-based generative models to high resolution images. Based on a new theoretical analysis on a simplified mixture model, we provide a method to analytically compute an effective set of Gaussian noise scales from training data. Additionally, we propose an efficient architecture to amortize the score estimation task across a large (possibly infinite) number of noise scales with a single neural network. Based on a simplified analysis of the convergence properties of the underlying Langevin dynamics sampling procedure, we also derive a technique to approximately optimize its performance as a function of the noise scales. Combining these techniques with an exponential moving average (EMA) of model parameters, we are able to significantly improve

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Improved Techniques for Training Score-Based Generative Models, Song & Ermon 2020-06-16

- **Theoretical Insights**: Provided a new understanding of score function learning in high-dimensional spaces.
- **Score Matching**: Train a network to estimate $\nabla_x \log p(x)$ at each noise level.
- **Adaptive Noise Scaling**: Introduced analytical methods to determine effective noise scales for training.
- **Stable Training**: Utilized Exponential Moving Average (EMA) of model parameters to improve training stability and sample quality.
- **Optimized Sampling**: Enhanced Langevin dynamics (type of Monte Carlo) for faster and more stable image generation from 64×64 to 256×256.

Improved Techniques for Training Score-Bas
2020-10-23

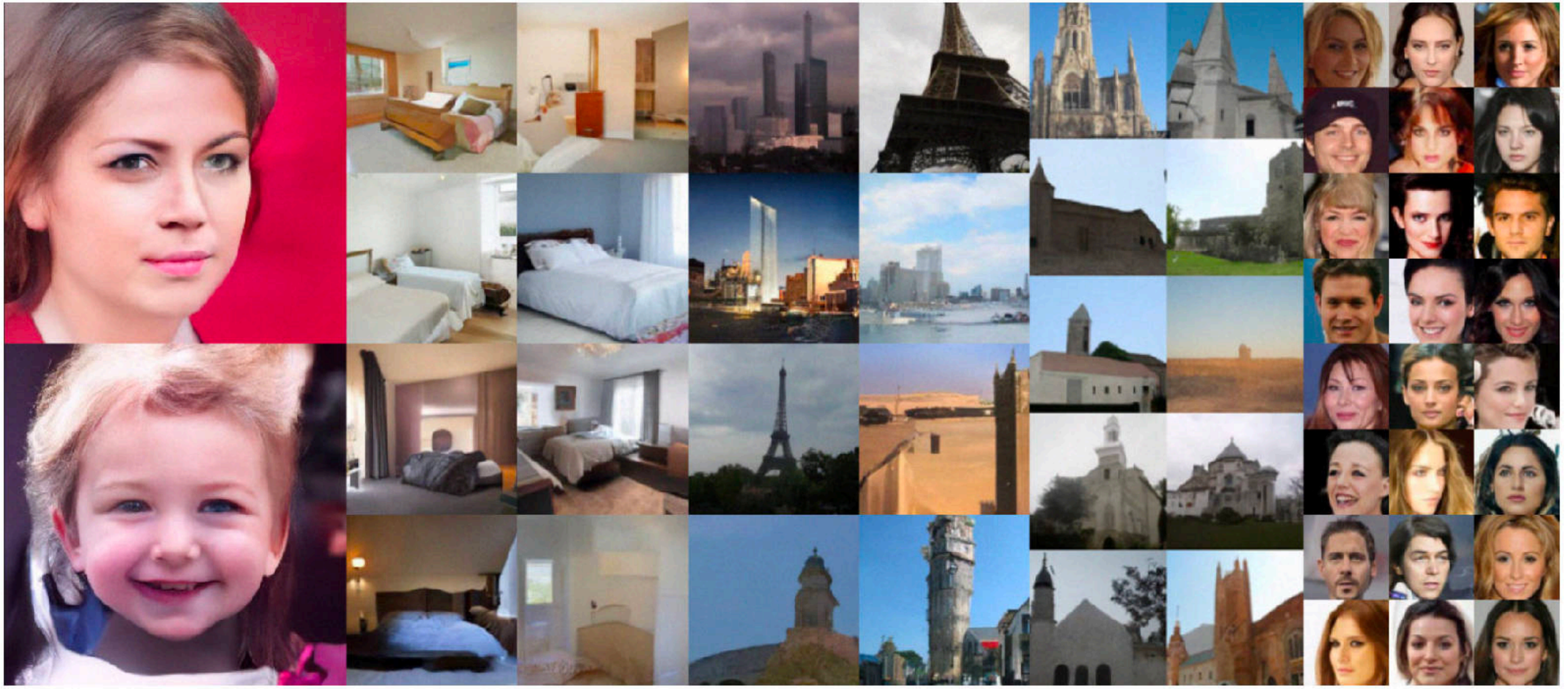


Figure 1: Generated samples on datasets of decreasing resolutions. From left to right: FFHQ 256×256, LSUN bedroom 128×128, LSUN tower 128×128, LSUN church_outdoor 96×96, and CelebA 64×64.

Improved Techniques for Training Score-Based Generative Models, Song & Ermon 2020-10-23

Table 1: Inception and FID scores.

Model	Inception ↑	FID ↓
CIFAR-10 Unconditional		
PixelCNN [17]	4.60	65.93
IGEBM [18]	6.02	40.58
WGAN-GP [19]	$7.86 \pm .07$	36.4
SNGAN [20]	$8.22 \pm .05$	21.7
NCSN [1]	$\mathbf{8.87 \pm .12}$	25.32
NCSN (w/ denoising)	$7.32 \pm .12$	29.8
NCSNv2 (w/o denoising)	$8.73 \pm .13$	31.75
NCSNv2 (w/ denoising)	$8.40 \pm .07$	**10.87**
CelebA 64×64		
NCSN (w/o denoising)	-	26.89
NCSN (w/ denoising)	-	25.30
NCSNv2 (w/o denoising)	-	28.86
NCSNv2 (w/ denoising)	-	**10.23**

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanhc@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at https://github.com/hojonathanho/diffusion.

1 Introduction

Deep generative models of all kinds have recently exhibited high quality samples in a wide variety of data modalities. Generative adversarial networks (GANs), autoregressive models, flows, and variational autoencoders (VAEs) have synthesized striking image and audio samples [14, 27, 3, 58, 38, 25, 10, 32, 44, 57, 26, 33, 45], and there have been remarkable advances in energy-based modeling and score matching that have produced images comparable to those of GANs [11, 55].

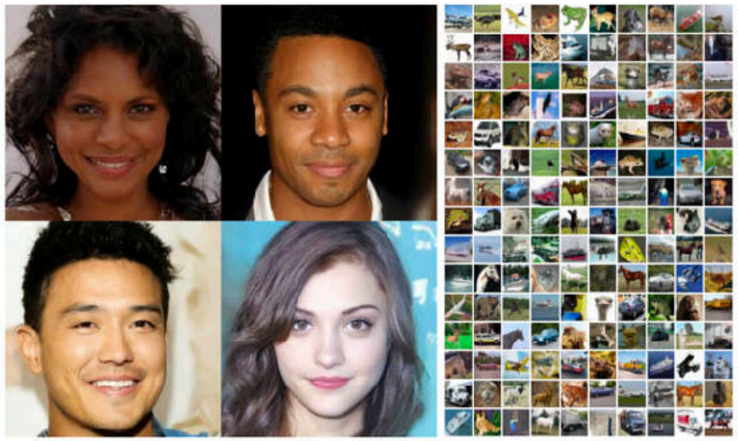


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-06-19

- **Continuous Diffusion:** Forward & reverse diffusion processes simulate gradual noise addition and denoising.
- **Noise prediction:** objective simplifies training and aligns with score matching.
- **Fixed variance:** in reverse process stabilizes training and reduces parameterization.
- **High-Fidelity Outputs**: Generates images with quality rivaling GANs, without adversarial training.
- **U-Net with Time Conditioning**: Utilizes a U-Net architecture with time-step embeddings for effective denoising across timesteps.

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-06-19

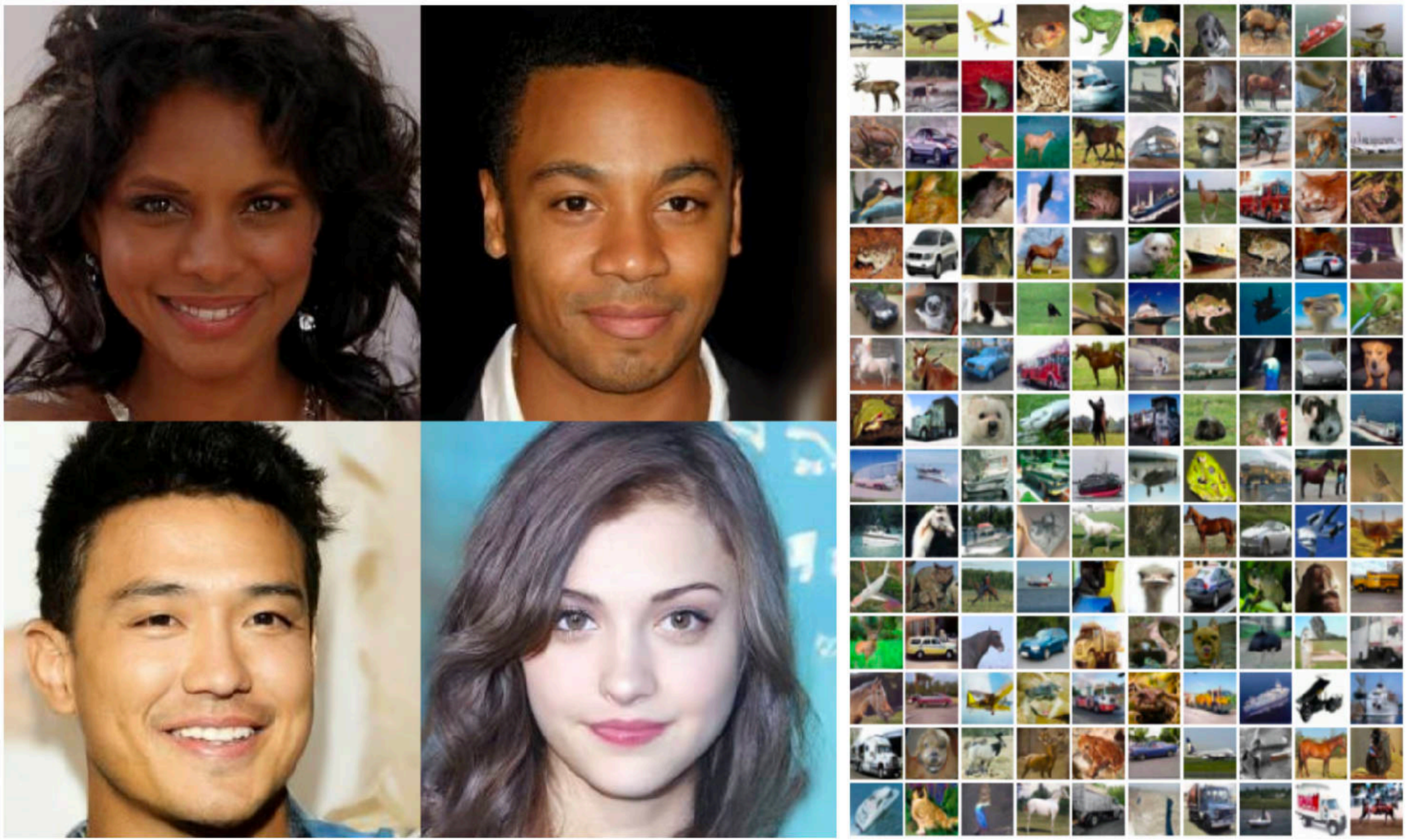


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-06-19

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]			$\leq$ 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$\pm$0.12	25.32	
SNGAN [39]	8.22$\pm$0.05	21.7	
SNGAN-DDLS [4]	9.09$\pm$0.10	15.42	
StyleGAN2 + ADA (v1) [29]	**9.74** $\pm$ 0.05	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67$\pm$0.13	13.51	$\leq$ 3.70 (3.69)
Ours (L_{simple})	9.46$\pm$0.11	**3.17**	$\leq$ 3.75 (3.72)

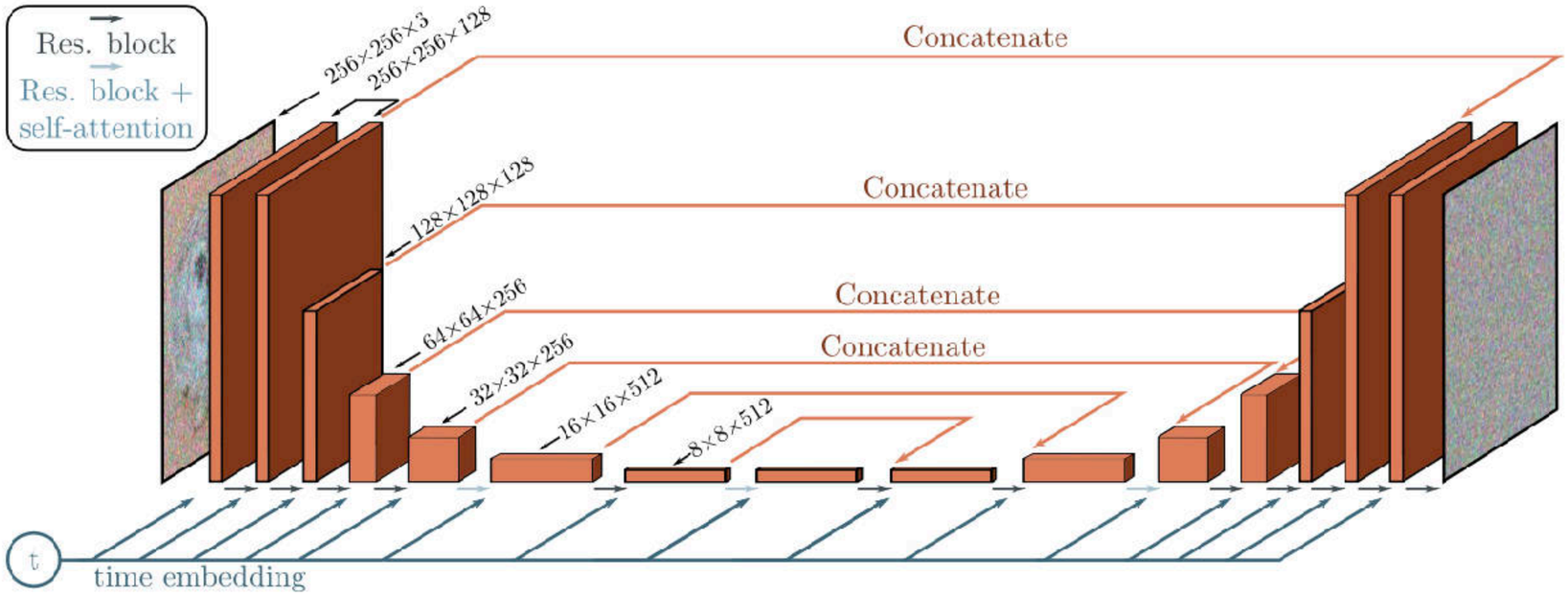


Figure 18.9 U-Net as used in diffusion models for images. The network aims to predict the noise that was added to the image. It consists of an encoder which reduces the scale and increases the number of channels and a decoder which increases the scale and reduces the number of channels. The encoder representations are concatenated to their partner in the decoder. Connections between adjacent representations consist of residual blocks, and periodic global self-attention in which every spatial position interacts with every other spatial position. A single network is used for all time steps, by passing a sinusoidal time embedding (figure 12.5) through a shallow neural network and adding the result to the channels at every spatial position at every stage of the U-Net.

Image from Understanding Deep Learning, Prince, 2023

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-12-16

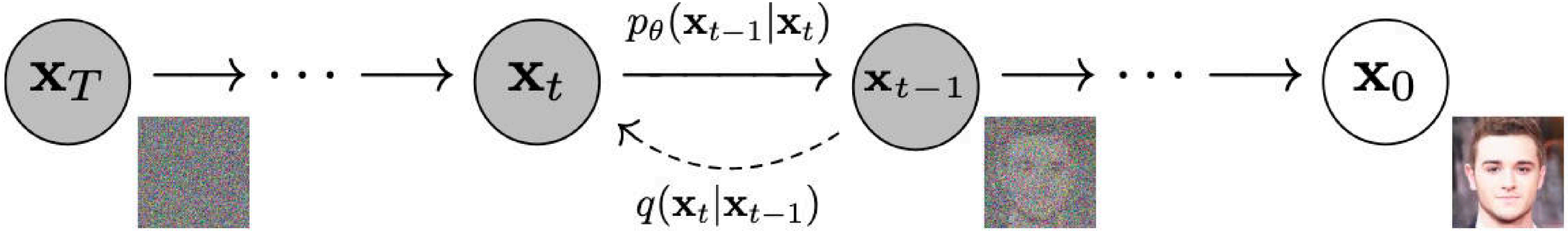


Figure 2: The directed graphical model considered in this work.

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-12-16

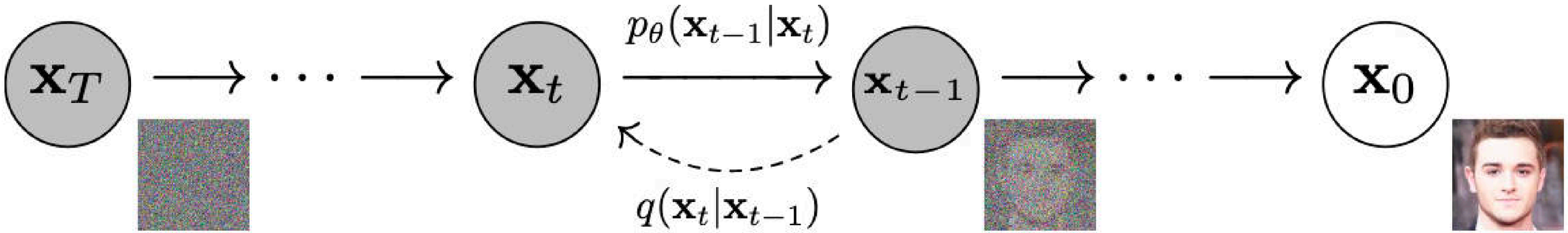


Figure 2: The directed graphical model considered in this work.

- $\mathbf{x}_0$: Original clean image (sampled from the data distribution).
- $\boldsymbol{\epsilon}$: Gaussian noise sampled from $\mathcal{N}(0,\mathbf{I})$.
- t: A randomly chosen time step (from 1 to T).
- $\bar{\alpha}_t$: A precomputed noise schedule value at time t, determining how much noise has been added.
- $\boldsymbol{\epsilon}_\theta(\cdot, t)$: The neural network (with parameters θ) that tries to predict the noise $\boldsymbol{\epsilon}$.

At each training step:

1. A clean image $\mathbf{x}_0$ is selected.
2. Noise $\boldsymbol{\epsilon}$ is added to create a noisy image: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
3. The model is given $\mathbf{x}_t$ and t, and tries to predict the noise $\boldsymbol{\epsilon}$.
4. The loss penalizes the difference between the true noise and the predicted noise.

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-12-16

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

- $\mathbf{x}_0$: Original clean image (sampled from the data distribution).
- $\boldsymbol{\epsilon}$: Gaussian noise sampled from $\mathcal{N}(0,\mathbf{I})$.
- t: A randomly chosen time step (from 1 to T).
- $\bar{\alpha}_t$: A precomputed noise schedule value at time t, determining how much noise has been added.
- $\boldsymbol{\epsilon}_\theta(\cdot, t)$: The neural network (with parameters θ) that tries to predict the noise $\boldsymbol{\epsilon}$.

At each training step:

1. A clean image $\mathbf{x}_0$ is selected.
2. Noise $\boldsymbol{\epsilon}$ is added to create a noisy image: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
3. The model is given $\mathbf{x}_t$ and t, and tries to predict the noise $\boldsymbol{\epsilon}$.
4. The loss penalizes the difference between the true noise and the predicted noise.

Denoising Diffusion Probabilistic Models, Ho Et al. 2020-12-16

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** $\mathbf{x}_0$

- $\boldsymbol{\epsilon}$: Gaussian noise sampled from $\mathcal{N}(0,\mathbf{I})$.
- t: A randomly chosen time step (from 1 to T).
- $\bar{\alpha}_t$: A precomputed noise schedule value at time t, determining how much noise has been added.
- $\boldsymbol{\epsilon}_\theta(\cdot, t)$: The neural network (with parameters θ) that tries to predict the noise $\boldsymbol{\epsilon}$.

At each training step:

1. A clean image $\mathbf{x}_0$ is selected.
2. Noise $\boldsymbol{\epsilon}$ is added to create a noisy image: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
3. The model is given $\mathbf{x}_t$ and t, and tries to predict the noise $\boldsymbol{\epsilon}$.
4. The loss penalizes the difference between the true noise and the predicted noise.

Impact: These 2019–2020 advances proved that diffusion-based generation can match or exceed traditional generative models. They established core techniques (iterative denoising, score matching connections) that subsequent works would build upon, and hinted that diffusion might be a general-purpose paradigm applicable beyond images.

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

ABSTRACT

Creating noise from data is easy; creating data from noise is generative modeling. We present a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise. Crucially, the reverse-time SDE depends only on the time-dependent gradient field (a.k.a., score) of the perturbed data distribution. By leveraging advances in score-based generative modeling, we can accurately estimate these scores with neural networks, and use numerical SDE solvers to generate samples. We show that this framework encapsulates previous approaches in score-based generative modeling and diffusion probabilistic modeling, allowing for new sampling procedures and new modeling capabilities. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE. We also derive an equivalent neural ODE that samples from the same distribution as the SDE, but additionally enables exact likelihood computation, and improved sampling efficiency. In addition, we provide a new way to solve inverse problems with score-based models, as demonstrated with experiments on class-conditional generation, image inpainting, and colorization. Combined with multiple architectural improvements, we achieve record-breaking performance for unconditional image generation on CIFAR-10 with an Inception score of 9.89 and FID of 2.20, a competitive likelihood of 2.99 bits/dim, and demonstrate high fidelity generation of 1024 × 1024 images for the first time from a score-based generative model.

1 INTRODUCTION

Two successful classes of probabilistic generative models involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. *Score matching with Langevin dynamics* (SMLD) (Song & Ermon, 2019) estimates the *score* (*i.e.*, the gradient of the log probability density with respect to data) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decreasing noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a sequence of probabilistic models to reverse each step of the noise corruption, using knowledge of the functional form of the reverse distributions to make training tractable. For continuous state spaces, the DDPM training objective implicitly computes scores at each noise scale. We therefore refer to these two model classes together as *score-based generative models*.

Score-based generative models, and related techniques (Bordes et al., 2017; Goyal et al., 2017; Du & Mordatch, 2019), have proven effective at generation of images (Song & Ermon, 2019; 2020; Ho et al., 2020), audio (Chen et al., 2020; Kong et al., 2020), graphs (Niu et al., 2020), and shapes (Cai

*Work partially done during an internship at Google Brain.

Score-Based Generative Modeling Through Stochastic Differential Equations, Song Et al. 2020-11-26

- **Unified Framework**: Bridges diffusion models and score-based methods under stochastic differential equations (SDEs).
- **Flexible Sampling**: Introduces predictor-corrector and probability flow ordinary differential equations (ODE) for efficient sampling.
- **Controllable Generation**: Enables conditional tasks without retraining.

Score-Based Generative Modeling Through Stochastic Differential Equations, Song Et al. 2021-02-10

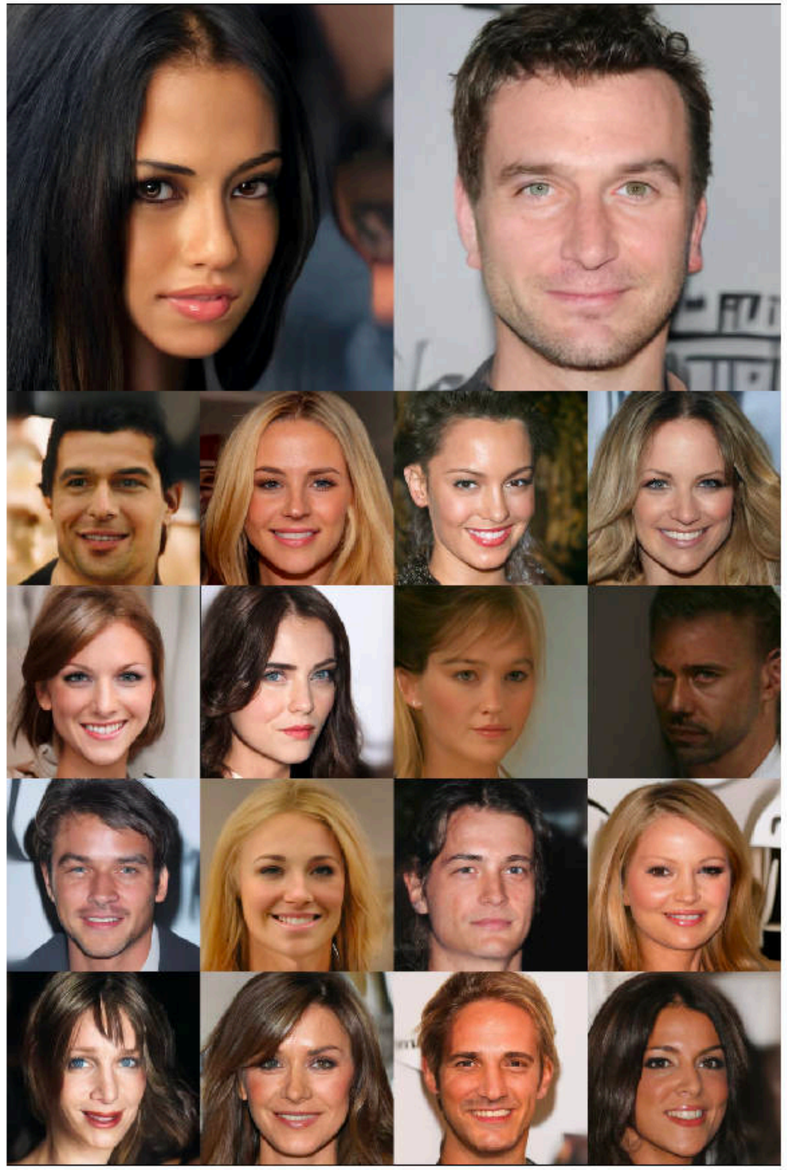


Figure 12: Samples on 1024 × 1024 CelebA-HQ from a modified NCSN++ model trained with the VE SDE.

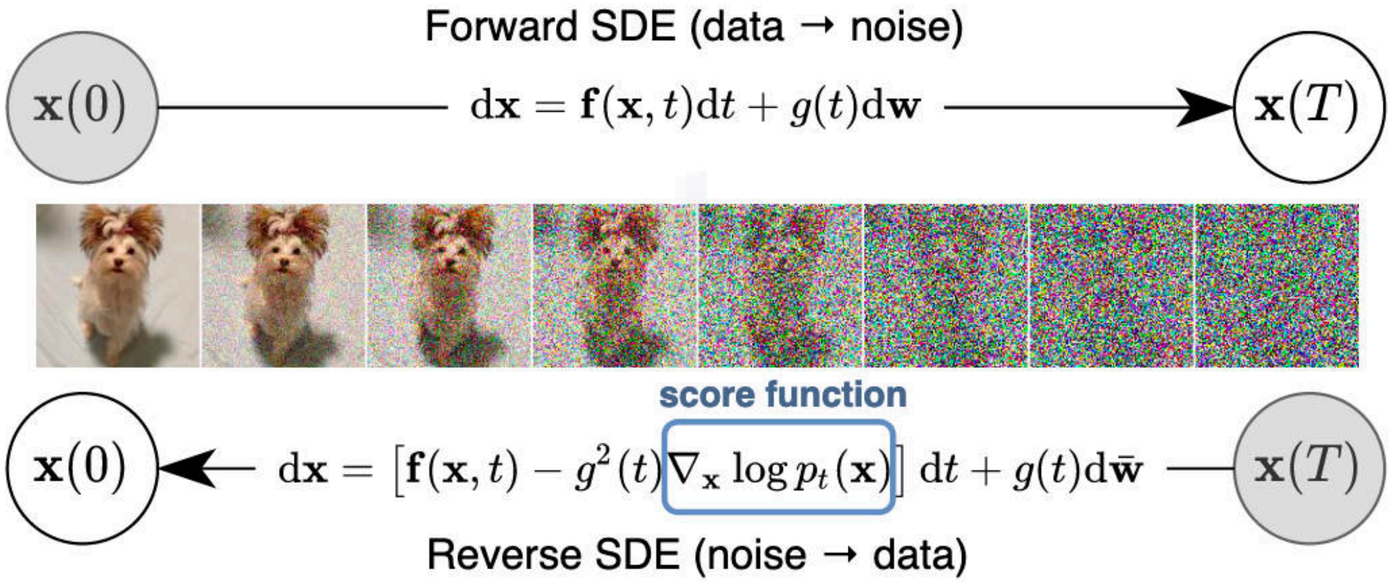


Figure 1: **Solving a reverse-time SDE yields a score-based generative model.** Transforming data to a simple noise distribution can be accomplished with a continuous-time SDE. This SDE can be reversed if we know the score of the distribution at each intermediate time step, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

Table 3: CIFAR-10 sample quality.

Model	FID↓	IS↑
Conditional		
BigGAN (Brock et al., 2018)	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	**2.42**	**10.14**
Unconditional		
StyleGAN2-ADA (Karras et al., 2020a)	2.92	9.83
NCSN (Song & Ermon, 2019)	25.32	8.87 ± .12
NCSNv2 (Song & Ermon, 2020)	10.87	8.40 ± .07
DDPM (Ho et al., 2020)	3.17	9.46 ± .11
DDPM++	2.78	9.64
DDPM++ cont. (VP)	2.55	9.58
DDPM++ cont. (sub-VP)	2.61	9.56
DDPM++ cont. (deep, VP)	2.41	9.68
DDPM++ cont. (deep, sub-VP)	2.41	9.57
NCSN++	2.45	9.73
NCSN++ cont. (VE)	2.38	9.83
NCSN++ cont. (deep, VE)	**2.20**	**9.89**

Algorithm 2 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}'_i \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 3 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}'_i \leftarrow (2 - \sqrt{1-\beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\beta_{i+1}}\mathbf{z}$ — Predictor
  for $j = 1$ **to** M **do** — Corrector
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 4 Corrector algorithm (VE SDE).

Require: $\{\sigma_i\}_{i=1}^N, r, N, M$.
$\mathbf{x}_N^0 \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
for $i \leftarrow N$ to 1 **do**
  for $j \leftarrow 1$ to M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{g} \leftarrow \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i^{j-1}, \sigma_i)$
    $\epsilon \leftarrow 2(r \|\mathbf{z}\|_2 / \|\mathbf{g}\|_2)^2$
    $\mathbf{x}_i^j \leftarrow \mathbf{x}_i^{j-1} + \epsilon\, \mathbf{g} + \sqrt{2\epsilon}\, \mathbf{z}$
  $\mathbf{x}_{i-1}^0 \leftarrow \mathbf{x}_i^M$
return $\mathbf{x}_0^0$

Algorithm 5 Corrector algorithm (VP SDE).

Require: $\{\beta_i\}_{i=1}^N, \{\alpha_i\}_{i=1}^N, r, N, M$.
$\mathbf{x}_N^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i \leftarrow N$ to 1 **do**
  for $j \leftarrow 1$ to M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{g} \leftarrow \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i^{j-1}, i)$
    $\epsilon \leftarrow 2\alpha_i(r \|\mathbf{z}\|_2 / \|\mathbf{g}\|_2)^2$
    $\mathbf{x}_i^j \leftarrow \mathbf{x}_i^{j-1} + \epsilon\, \mathbf{g} + \sqrt{2\epsilon}\, \mathbf{z}$
  $\mathbf{x}_{i-1}^0 \leftarrow \mathbf{x}_i^M$
return $\mathbf{x}_0^0$

0 - Introduction
1 - Preliminaries - Generative Models
2 - Preliminaries - Generative Model Taxonomy
3 - Diffusion Model Foundations

4 - Langauge Diffusion Models

Structured Denoising Diffusion Models in Discrete State-Spaces

Jacob Austin,* Daniel D. Johnson,* Jonathan Ho, Daniel Tarlow & Rianne van den Berg
Google Research, Brain Team
{jaaustin,ddjohnson,jonathanho,dtarlow,riannevdberg}@google.com

Abstract

Denoising diffusion probabilistic models (DDPMs) [19] have shown impressive results on image and waveform generation in continuous state spaces. Here, we introduce Discrete Denoising Diffusion Probabilistic Models (D3PMs), diffusion-like generative models for discrete data that generalize the multinomial diffusion model of Hoogeboom et al. [20], by going beyond corruption processes with uniform transition probabilities. This includes corruption with transition matrices that mimic Gaussian kernels in continuous space, matrices based on nearest neighbors in embedding space, and matrices that introduce absorbing states. The third allows us to draw a connection between diffusion models and autoregressive and mask-based generative models. We show that the choice of transition matrix is an important design decision that leads to improved results in image and text domains. We also introduce a new loss function that combines the variational lower bound with an auxiliary cross entropy loss. For text, this model class achieves strong results on character-level text generation while scaling to large vocabularies on LM1B. On the image dataset CIFAR-10, our models approach the sample quality and exceed the log-likelihood of the continuous-space DDPM model.

1 Introduction

Generative modeling is a core problem in machine learning, useful both for benchmarking our ability to capture statistics of natural datasets and for downstream applications that require generating high-dimensional data like images, text, and speech waveforms. There has been a great deal of progress with the development of methods like GANs [15, 4], VAEs [25, 35], large autoregressive neural network models [51, 50, 52], normalizing flows [34, 12, 24, 32], and others, each with their own tradeoffs in terms of sample quality, sampling speed, log-likelihoods, and training stability.

Recently, diffusion models [43] have emerged as a compelling alternative for image [19, 46] and audio [7, 26] generation, achieving comparable sample quality to GANs and log-likelihoods comparable to autoregressive models with fewer inference steps. A diffusion model is a parameterized Markov chain trained to reverse a predefined forward process, which is a stochastic process constructed to gradually corrupt training data into pure noise. Diffusion models are trained using a stable objective closely related to both maximum likelihood and score matching [21, 53], and they admit faster sampling than autoregressive models by using parallel iterative refinement [30, 45, 47, 44].

Although diffusion models have been proposed in both discrete and continuous state spaces [43], most recent work has focused on Gaussian diffusion processes that operate in continuous state spaces (e.g. for real-valued image and waveform data). Diffusion models with discrete state spaces have been explored for text and image segmentation domains [20], but they have not yet been demonstrated as a competitive model class for large scale text or image generation.

Preprint. Under review.

*Equal contributions

Structured Denoising Diffusion Models in Discrete State-Spaces, Austin et al. 2021-07-13

- **Structured Transitions**: Utilize Gaussian-like, nearest-neighbor, and absorbing state transitions for nuanced discrete data modeling.
- **Hybrid Loss**: Combine variational and cross-entropy losses to stabilize and enhance training.
- **Scalable Text Modeling**: Effectively handle large vocabularies in language tasks like LM1B.

Structured Denoising Diffusion Models in Discrete State-Spaces, Austin et al. 2021-07-13

“We develop structured corruption processes appropriate for text data, using similarity between tokens to enable gradual corruption and denoising. Expanding further, we also explore corruption processes that insert [MASK] tokens, which let us draw parallels to autoregressive and mask-based generative models."

Structured Denoising Diffusion Models in Discrete State-Spaces, Austin et al. 2021-07-13

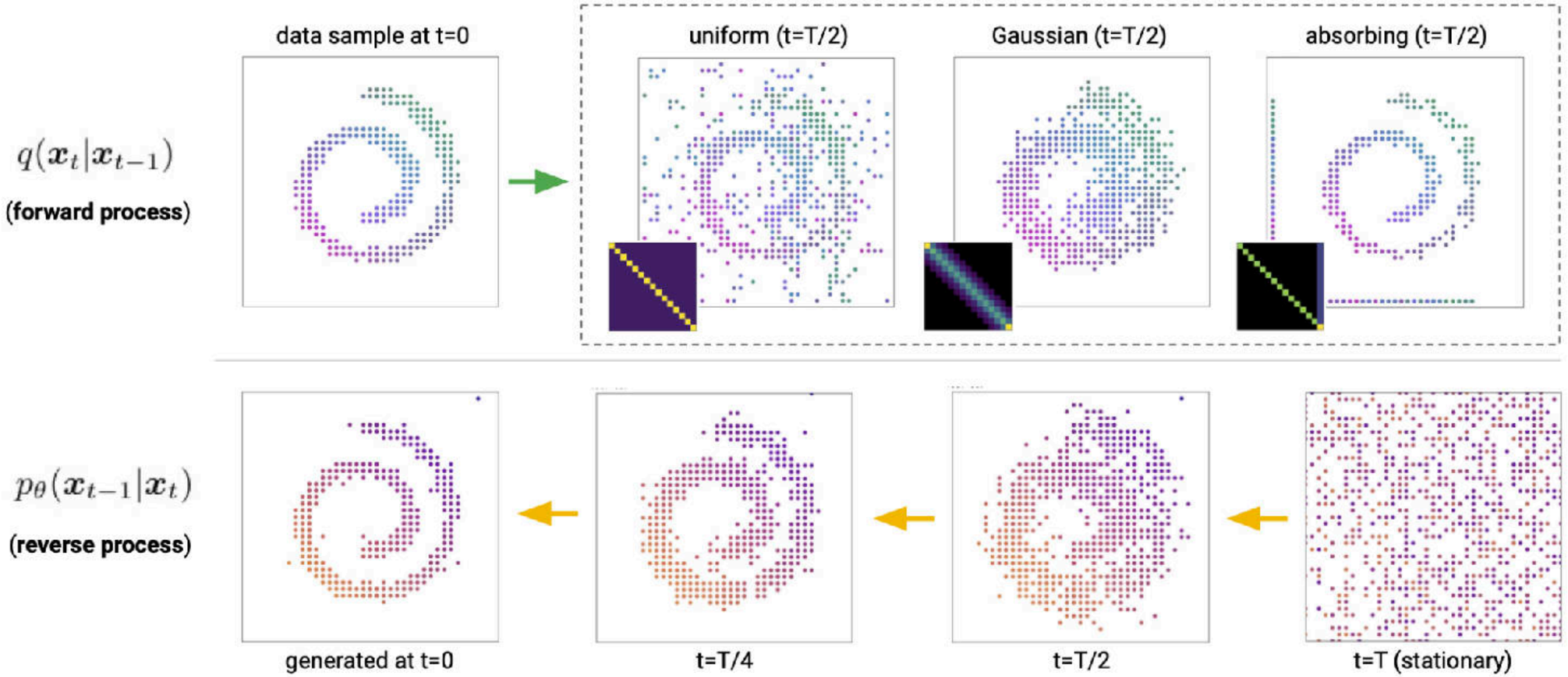


Figure 1: D3PM forward and (learned) reverse process applied to a quantized swiss roll. Each dot represents a 2D categorical variable. Top: samples from the uniform, discretized Gaussian, and absorbing state D3PM model forward processes, along with corresponding transition matrices $\boldsymbol{Q}$. Bottom: samples from a learned discretized Gaussian reverse process.

Masked Diffusion

Also known as absorbing diffusion, first proposed in Austin et al. (2021)

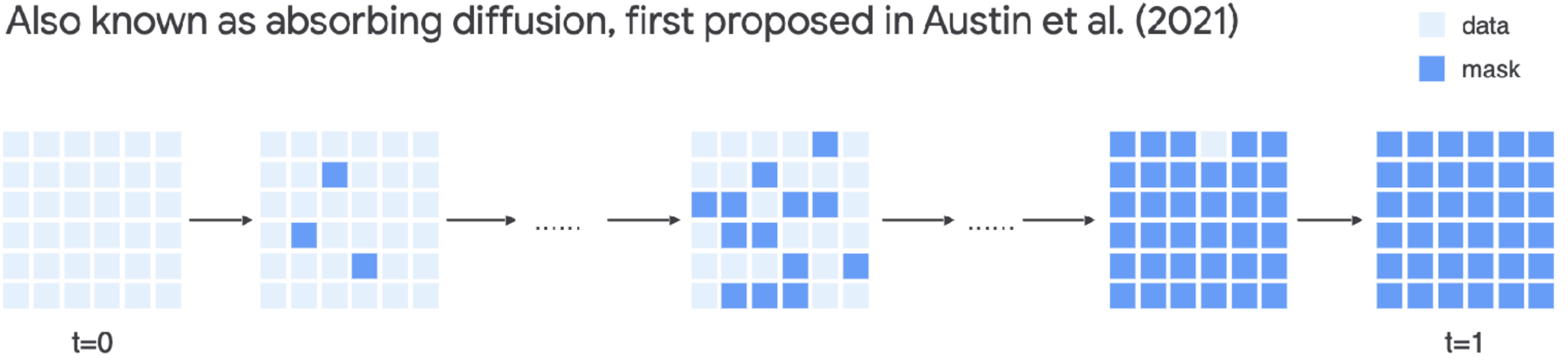
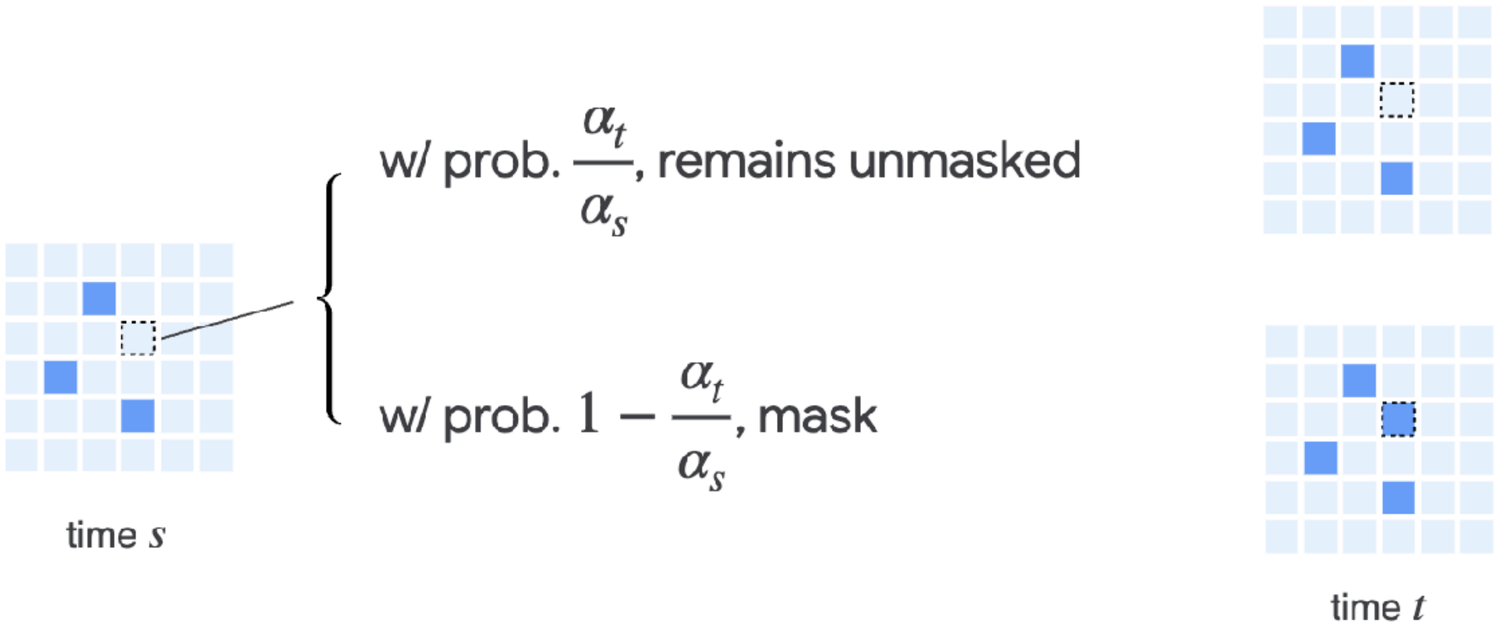


Figure from Discrete Generative Modeling with Masked Diffusions Jiaxin Shi Google DeepMind 2024/9/18 @GenU 2024 https://jiaxins.io/talks/jiaxins_md4_genu.pdf

Masked Diffusion

Forward process $q(x_t \mid x_s)$



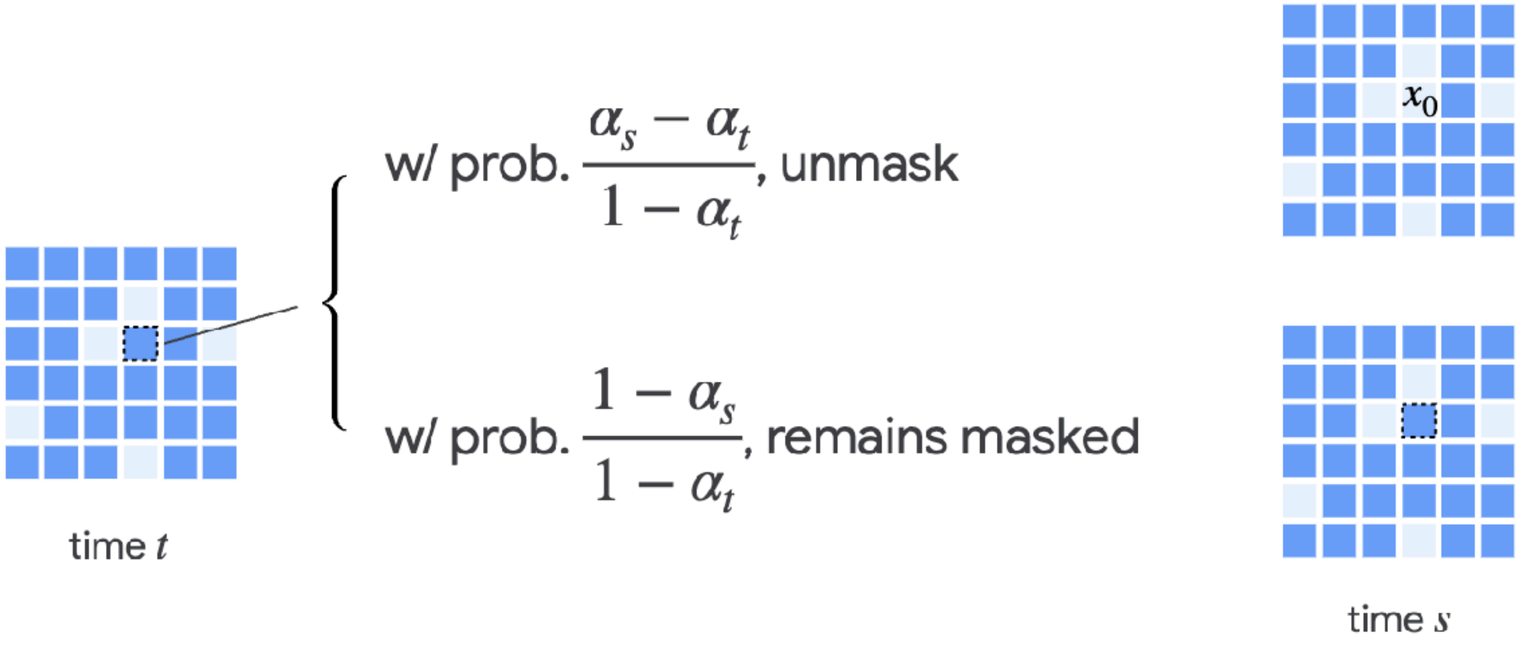
Transition matrix: $\bar{Q}(s,t)_{ij} \triangleq q(x_t = j \mid x_s = i)$

$$\bar{Q}(s,t) = \frac{\alpha_t}{\alpha_s} I + \left(1 - \frac{\alpha_t}{\alpha_s}\right) \mathbf{1} e_m^\top$$

Figure from Discrete Generative Modeling with Masked Diffusions Jiaxin Shi Google DeepMind 2024/9/18 @GenU 2024 https://jiaxins.io/talks/jiaxins_md4_genu.pdf

Masked Diffusion

Reverse process $q(x_s \mid x_t, x_0)$ (knowing x_0)

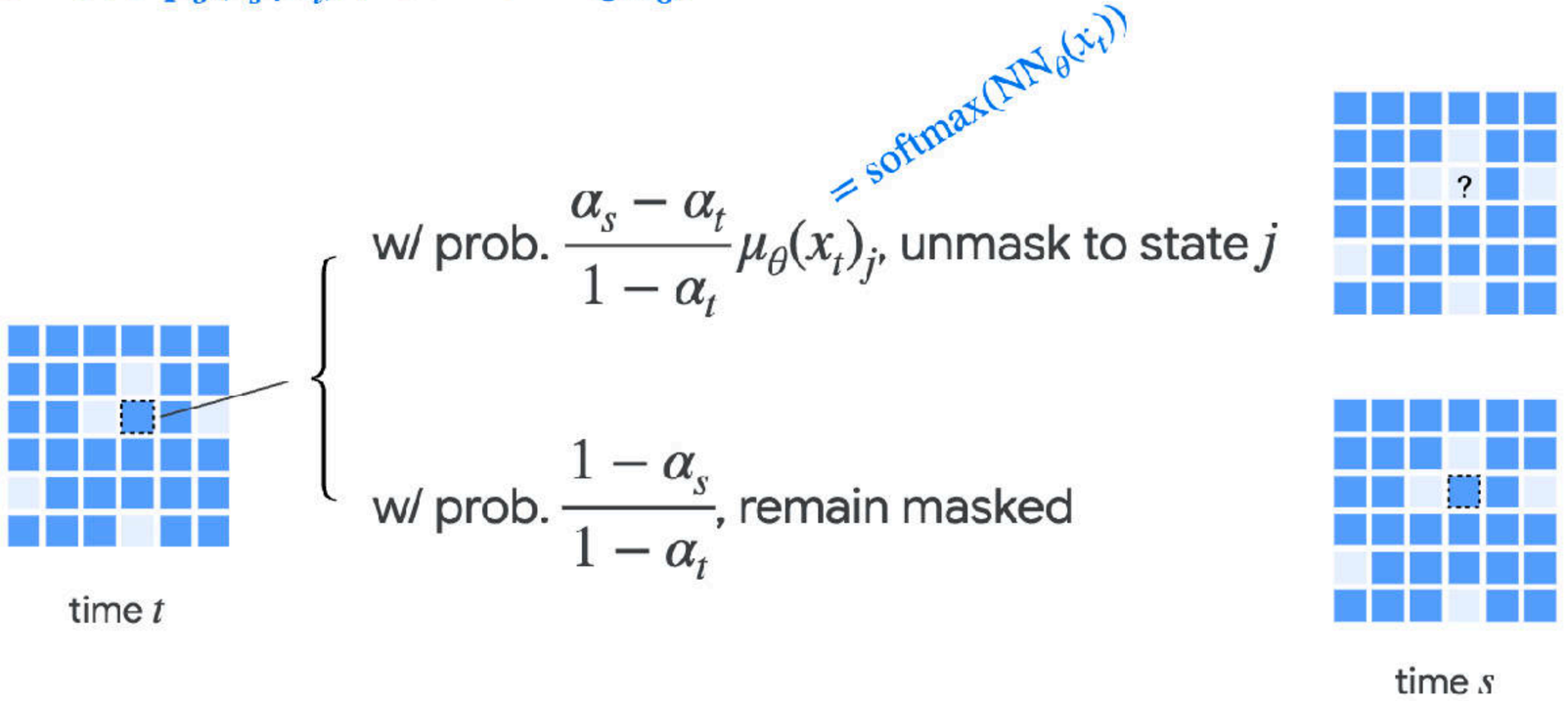


Transition matrix $\bar{R}^{x_0}(t, s) = I + \frac{\alpha_s - \alpha_t}{1 - \alpha_t} e_m (x_0 - e_m)^\top$

Figure from Discrete Generative Modeling with Masked Diffusions Jiaxin Shi Google DeepMind 2024/9/18 @GenU 2024 https://jiaxins.io/talks/jiaxins_md4_genu.pdf

Masked Diffusion Models

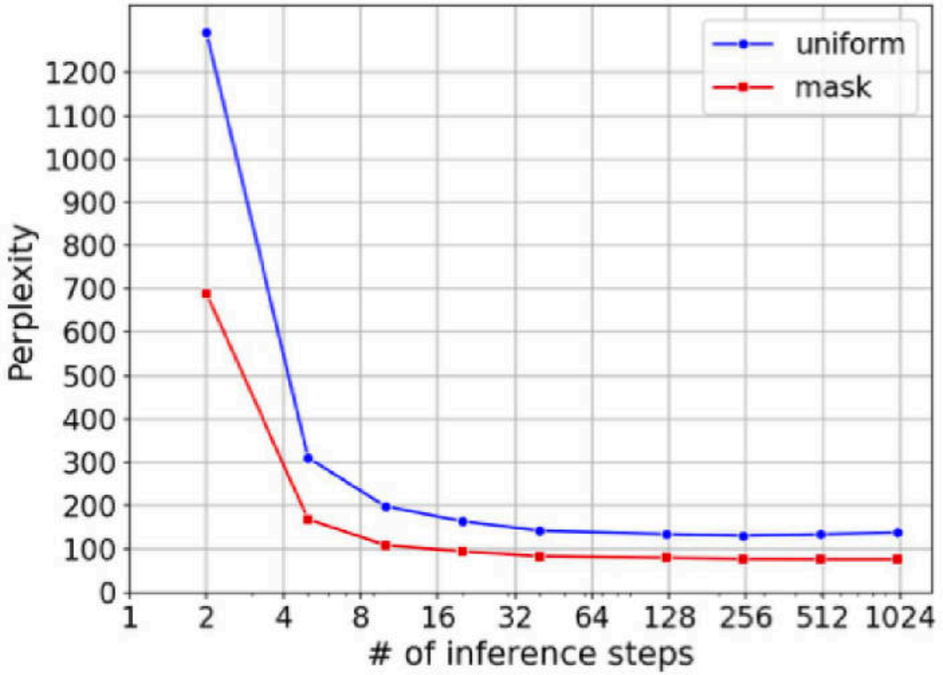
Generative model $p_\theta(x_s \mid x_t)$ (not knowing x_0)



Mean Parameterization: $\mu_\theta(x_t)$ as a prediction model for $\mathbb{E}[x_0 \mid x_t]$

Figure from Discrete Generative Modeling with Masked Diffusions Jiaxin Shi Google DeepMind 2024/9/18 @GenU 2024 https://jiaxins.io/talks/jiaxins_md4_genu.pdf

Structured Denoising Diffusion Models in Discrete State-Spaces, Austin et al. 2021-07-13



t = 128 **[MASK] [MASK] [MASK] [MASK] [MASK] [MASK]...**
t = 25 In response **[MASK]** the demands , **[MASK] [MASK]**y Workers union said **[MASK]** backflow fund **[MASK]**s would face further investigation and a fine.
t = 0 In response to the demands , the Community Workers union said the backflow fund managers would face further investigation and a fine .

Original: Caterpillar is eager to expand in Asia , where it trails local competitors such as Komatsu Ltd
Corrupted: Caterpillar is eager to expand in**[MASK]** , **[MASK]** it **[MASK]** s local competitors such as Komatsu Ltd
Reconstructed: Caterpillar is eager to expand in China , where it faces local competitors such as Komatsu Ltd

Figure 2: Left: perplexity v.s. sampling iterations for LM1B. Right: Using a trained D3PM absorbing model for LM1B to (top) generate new sentences and (bottom) reconstruct corrupted examples.

Structured Denoising Diffusion Models in Discrete State-Spaces, Austin et al. 2021-07-13

BERT is a one-step diffusion model: One possible D3PM transition matrix is a combination of a uniform transition matrix and an absorbing state at the [MASK] token (i.e. $\boldsymbol{Q} = \alpha \mathbb{1} e_m^T + \beta \mathbb{1}\mathbb{1}^T/K + (1-\alpha-\beta)I$, where e_m is a one-hot vector on the [MASK] token). For a one-step diffusion process in which $q(\boldsymbol{x}_1|\boldsymbol{x}_0)$ replaces 10% of tokens with [MASK] and 5% uniformly at random, this leads precisely to the BERT denoising objective, i.e. $L_{vb} - L_T = -\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)] = L_{BERT}$, since L_T is a constant independent of θ (assuming a fixed prior).

Autoregressive models are (discrete) diffusion models: Consider a diffusion process that deterministically masks tokens one-by-one in a sequence of length $N = T$: $q([\boldsymbol{x}_t]_i \mid \boldsymbol{x}_0) = [\boldsymbol{x}_0]_i$ if $i <$

$N - t$ else [MASK] . This is a deterministic forward process, so $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ is a delta distribution on the $\boldsymbol{x}_t$ sequence with one fewer mask: $q([\boldsymbol{x}_{t-1}]_i|\boldsymbol{x}_t, \boldsymbol{x}_0) = \delta_{[\boldsymbol{x}_t]_i}$ if $i \neq T - t$ else $\delta_{[\boldsymbol{x}_0]_i}$. While this process is not applied independently to each token, it can be recast as an independently-applied diffusion process on the product space $[0...N] \times \mathcal{V}$, where each token is tagged with its position in the sequence, $\mathcal{V}$ is the vocabulary, and $\boldsymbol{Q}$ is an $N \times |\mathcal{V}| \times N \times |\mathcal{V}|$ sparse matrix.

Because all tokens except the one at position $i = T - t$ have deterministic posteriors, the KL divergence $D_{KL}(q([\boldsymbol{x}_{t-1}]_j|\boldsymbol{x}_t, \boldsymbol{x}_0) \;||\; p_\theta([\boldsymbol{x}_{t-1}]_j|\boldsymbol{x}_t))$ is zero for all other positions. The only token for which this is not true is the token at position i, for which $D_{KL}(q([\boldsymbol{x}_{t-1}]_i|\boldsymbol{x}_t, \boldsymbol{x}_0) \;||\; p_\theta([\boldsymbol{x}_{t-1}]_i|\boldsymbol{x}_t)) = -\log p_\theta([\boldsymbol{x}_0]_i|\boldsymbol{x}_t)$, the standard cross entropy loss for an autoregressive model.

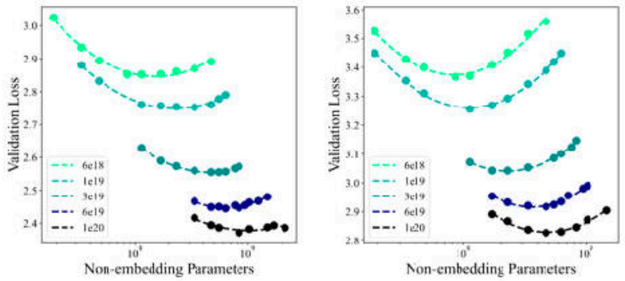
(Generative) Masked Language-Models (MLMs) are diffusion models: Generative Masked Language Models ([14], [54]) are generative models that generate text from a sequence of [MASK] tokens. They are usually trained by sampling a sequence $\boldsymbol{x}_0$, masking k tokens according to some schedule, and learning to predict the masked tokens given context. It turns out that a D3PM absorbing ([MASK]) model trained on the usual ELBO objective with the $\boldsymbol{x}_0$-parameterization from 3.3 reduces to a reweighted version of this MLM objective (see Appendix A.3 for a detailed derivation).

SCALING UP MASKED DIFFUSION MODELS ON TEXT

Shen Nie[1,2*], **Fengqi Zhu**[1,2], **Chao Du**[3‡], **Tianyu Pang**[3], **Qian Liu**[3], **Guangtao Zeng**[4]
Min Lin[3], **Chongxuan Li**[1,2‡†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]Sea AI Lab, Singapore [4]Singapore University of Technology and Design
{nieshen, fengqizhu}@ruc.edu.cn; {duchao, tianyupang, liuqian}@sea.com;
zengguangtao98@gmail.com; linmin@sea.com; chongxuanli@ruc.edu.cn

ABSTRACT

Masked diffusion models (MDMs) have shown promise in language modeling, yet their scalability and effectiveness in core language tasks, such as text generation and language understanding, remain underexplored. This paper establishes the first scaling law for MDMs, demonstrating a scaling rate comparable to autoregressive models (ARMs) and a relatively small compute gap. Motivated by their scalability, we train a family of MDMs with up to 1.1 billion (B) parameters to systematically evaluate their performance against ARMs of comparable or larger sizes. Fully leveraging the probabilistic formulation of MDMs, we propose a simple yet effective *unsupervised classifier-free guidance* that effectively exploits large-scale unpaired data, boosting performance for conditional inference. In language understanding, the 1.1B MDM outperforms the 1.1B TinyLlama model trained on the same data across four of eight zero-shot benchmarks. Notably, it achieves competitive math reasoning ability with the 7B Llama-2 model on the GSM8K dataset. In text generation, MDMs with 16 times more pre-training time offer a flexible trade-off against ARMs with the accelerated sampling technique KV-Cache: MDMs match ARMs in performance while being 1.4 times faster during sampling. Moreover, MDMs address challenging tasks for ARMs by effectively handling bidirectional reasoning and adapting to temporal shifts in data. Notably, a 1.1B MDM breaks the *reverse curse* encountered by much larger ARMs with significantly more data and computation, such as 13B Llama-2 and 175B GPT-3. Our code is available at https://github.com/ML-GSAI/SMDM.



(a) ARMs.

(b) MDMs.

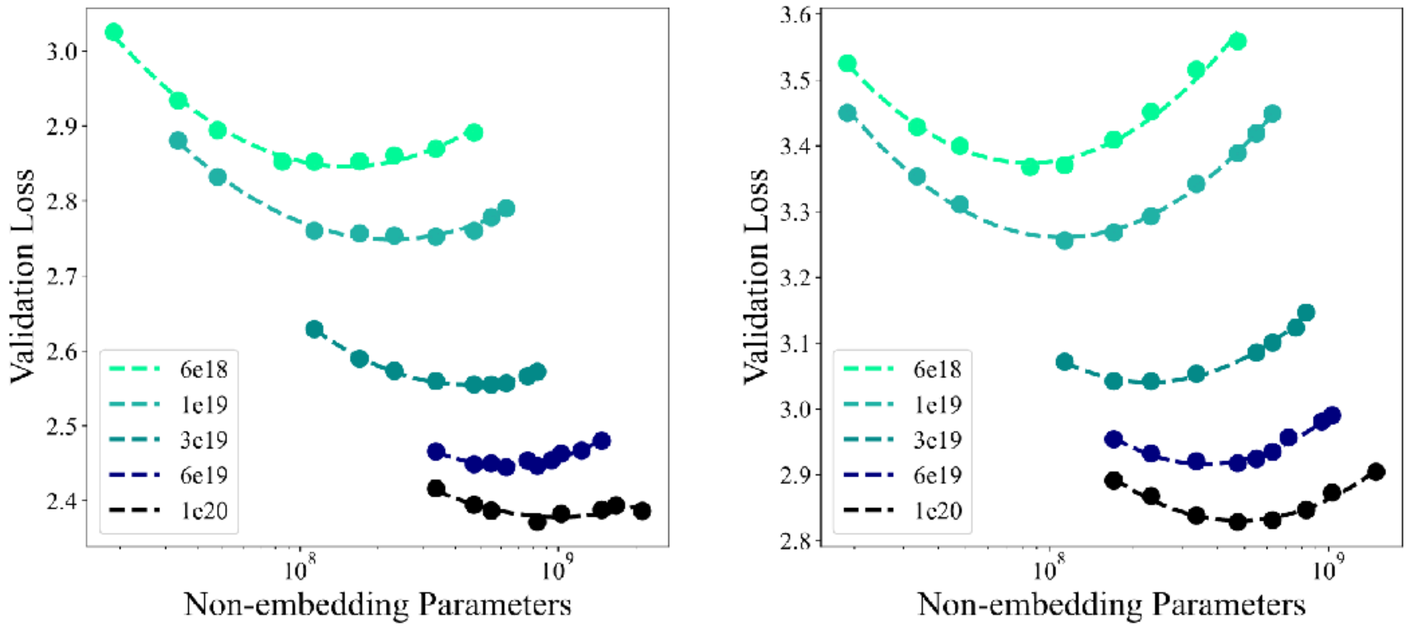
Figure 1: **IsoFLOP curves** plot optimal model sizes under fixed computation budgets. The optimal MDMs validation loss exhibits power-law scaling, decreasing at a rate comparable to that of ARMs.

*Work done during Shen Nie's internship at Sea AI Lab.
‡Project leaders. †Correspondence to Chongxuan Li.

Scaling up Masked Diffusion Models on Text, Nie et al. 2024-10-24

- **Scalable Performance**: MDMs scale effectively, matching the performance improvements seen in ARMs as model size and compute increase.
- **Competitive with Larger ARMs**: A 1.1B MDM outperforms or matches larger ARMs like GPT-2 (1.5B) and Llama-2 (7B) in various tasks.
- **Robust Bidirectional Reasoning**: MDMs handle bidirectional context and temporal data shifts better than traditional ARMs.
- **Efficient Generation**: Offers faster sampling speeds and flexible quality-computation trade-offs compared to ARMs.
- **Unsupervised Conditional Generation**: Introduces a classifier-free guidance method that enhances conditional generation without labeled data.

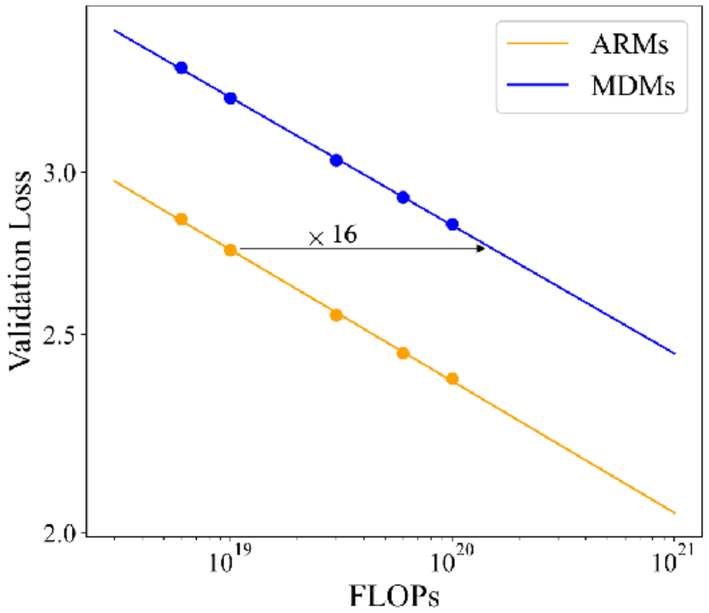
Scaling up Masked Diffusion Models on Text, Nie et al. 2024-10-24



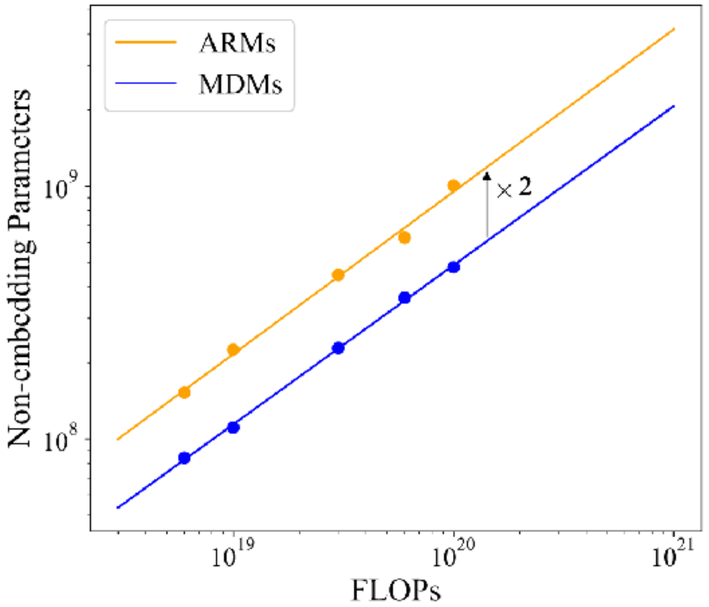
(a) ARMs. (b) MDMs.

Figure 1: **IsoFLOP curves** plot optimal model sizes under fixed computation budgets. The optimal MDMs validation loss exhibits power-law scaling, decreasing at a rate comparable to that of ARMs.

Scaling up Masked Diffusion Models on Text, Nie et al. 2024-10-24



(a) Loss-Flops curve.



(b) Parameters-Flops curve.

Figure 2: **Scaling laws for MDMs.** Compared to ARMs, MDMs demonstrate competitive scalability with comparable scaling rates and similar scaling behavior on utilizing the parameter capacity.

Scaling up Masked Diffusion Models on Text, Nie et al. 2024-10-24

Table 2: **Evaluation of our 1.1B MDM.** Both the MDM and Llama-2 models are fine-tuned for GSM8K, with all other benchmarks assessed in zero-shot settings. Result marked * is from Gong et al. (2024). The pre-training datasets consist of approximately 540B tokens for TinyLlama and MDM, compared to 2T tokens for Llama-2. Our 1.1B MDM outperforms the same-size TinyLlama on four out of eight tasks and surpasses the larger GPT-2 on six tasks. Notably, MDM achieves GSM8K accuracy comparable to that of Llama-2, requiring less than 5% of its pre-training FLOPs.

	FLOPs	ARC-e	BoolQ	Hellaswag	Obqa	PIQA	RACE	SIQA	LAMBADA	GSM8K
GPT-2 (1.5B)	-	51.05	61.77	50.89	32.00	**70.51**	33.11	40.28	44.61	-
TinyLlama (1.1B)	3.3×10^{21}	**52.19**	59.39	**54.07**	33.20	70.29	**35.60**	39.41	43.22	-
MDM (1.1B)	3.3×10^{21}	48.74	**62.17**	51.83	**33.40**	69.53	**35.60**	**41.04**	**52.73**	58.5
Llama-2 (7B)	7.8×10^{22}	74.49	77.68	75.98	44.20	79.00	39.52	46.11	68.00	58.6*

Large Language Diffusion Models,
Nie et al. 2025-02-14

- **Scalable**: Demonstrates strong scalability with model size and data up to 7B, offering efficient inference capabilities.
- **ARM-Level Performance**: Matches or exceeds leading autoregressive models (LLaMa 2 7B & LLaMa 3 8B) in various language tasks.
- **Bidirectional Reasoning**: Effectively handles tasks requiring reverse reasoning, overcoming limitations of traditional ARMs. Attention doesn't use a causal mask.

Large Language Diffusion Models

Shen Nie[1*†] **Fengqi Zhu**[1*†] **Zebin You**[1†] **Xiaolu Zhang**[2‡] **Jingyang Ou**[1] **Jun Hu**[2‡] **Jun Zhou**[2]
Yankai Lin[1‡] **Ji-Rong Wen**[1] **Chongxuan Li**[1‡¶]

Abstract

Autoregressive models (ARMs) are widely regarded as the cornerstone of large language models (LLMs). We challenge this notion by introducing **LLaDA**, a diffusion model trained from scratch under the pre-training and supervised fine-tuning (SFT) paradigm. LLaDA models distributions through a forward data masking process and a reverse process, parameterized by a vanilla Transformer to predict masked tokens. By optimizing a likelihood bound, it provides a principled generative approach for probabilistic inference. Across extensive benchmarks, LLaDA demonstrates strong *scalability*, outperforming our self-constructed ARM baselines. Remarkably, LLaDA 8B is competitive with strong LLMs like LLaMA3 8B in *in-context learning* and, after SFT, exhibits impressive *instruction-following* abilities in case studies such as multi-turn dialogue. Moreover, LLaDA addresses the reversal curse, surpassing GPT-4o in a reversal poem completion task. Our findings establish diffusion models as a viable and promising alternative to ARMs, challenging the assumption that key LLM capabilities discussed above are inherently tied to ARMs. Project page and codes: https://ml-gsai.github.io/LLaDA-demo/.

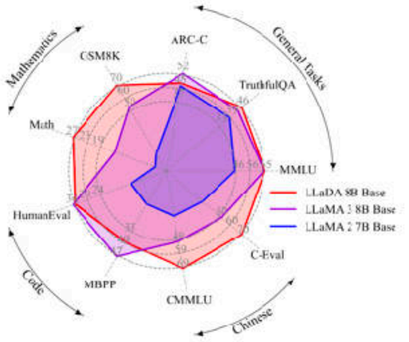


Figure 1. **Zero/Few-Shot Benchmarks.** We scale LLaDA to an unprecedented size of 8B parameters from scratch, achieving competitive performance with strong LLMs (Dubey et al., 2024).

1. Introduction

What is now proved was once only imagined.
—William Blake

Large language models (LLMs) (Zhao et al., 2023) fall entirely within the framework of *generative modeling*. Specifically, LLMs aim to capture the true but unknown language distribution $p_{\text{data}}(\cdot)$ by optimizing a model distribution $p_\theta(\cdot)$ through maximum likelihood estimation, or equivalently KL divergence minimization between the two distributions:

$$\underbrace{\max_\theta \mathbb{E}_{p_{\text{data}}(x)} \log p_\theta(x) \Leftrightarrow \min_\theta \mathrm{KL}(p_{\text{data}}(x)||p_\theta(x))}_{\text{Generative modeling principles}}. \quad (1)$$

The predominant approach relies on the *autoregressive* modeling (ARM)—commonly referred to as the *next-token prediction* paradigm—to define the model distribution:

$$\underbrace{p_\theta(x) = p_\theta(x^1)\prod_{i=2}^{L} p_\theta(x^i \mid x^1, \ldots, x^{i-1}),}_{\text{Autoregressive formulation}} \quad (2)$$

where x is a sequence of length L, and x^i is the i-th token.

This paradigm has proven remarkably effective (Radford, 2018; Radford et al., 2019; Brown, 2020; OpenAI, 2022) and has become the foundation of current LLMs. Despite its widespread adoption, a fundamental question remains unanswered: *Is the autoregressive paradigm the only viable path to achieving the intelligence exhibited by LLMs?*

*Equal contribution †Work done during an internship at Ant Group ‡Project leaders [1]Gaoling School of Artificial Intelligence, Renmin University of China; Beijing Key Laboratory of Big Data Management and Analysis Methods [2]Ant Group ¶Correspondence to: Chongxuan Li <chongxuanli@ruc.edu.cn>.

Preprint.

Large Language Diffusion Models, Nie et al. 2025-02-14

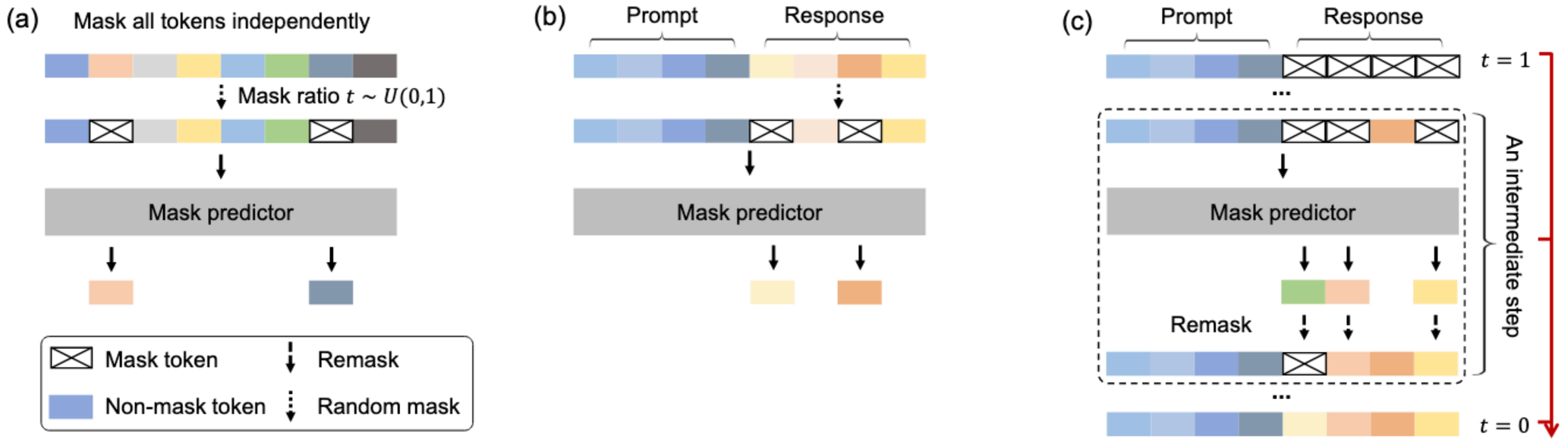


Figure 2. **A Conceptual Overview of LLaDA.** (a) Pre-training. LLaDA is trained on text with random masks applied independently to all tokens at the same ratio $t \sim U[0, 1]$. (b) SFT. Only response tokens are possibly masked. (c) Sampling. LLaDA simulates a diffusion process from $t = 1$ (fully masked) to $t = 0$ (unmasked), predicting all masks simultaneously at each step with flexible remask strategies.

Large Language Diffusion Models, Nie et al. 2025-02-14

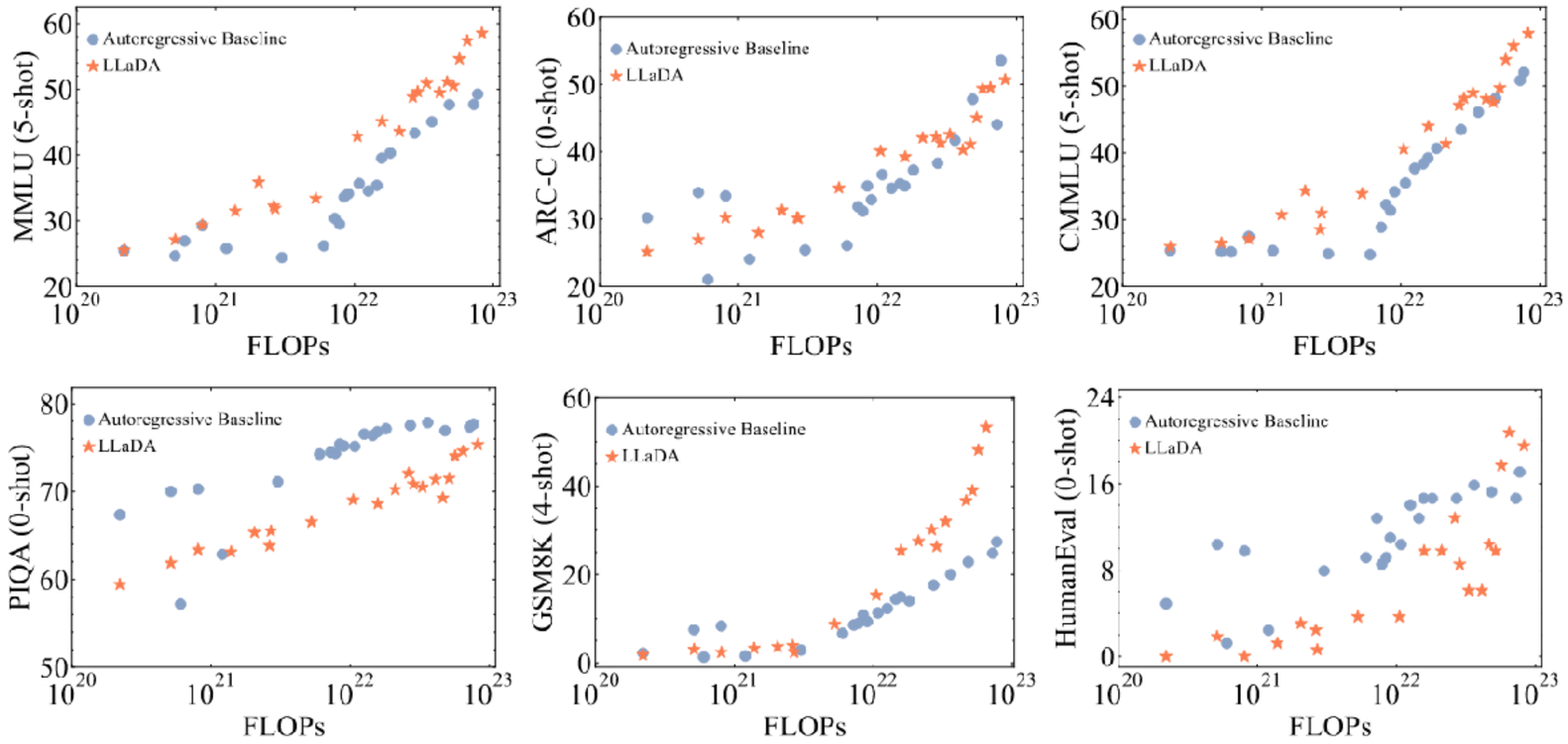


Figure 3. **Scalability of LLaDA.** We evaluate the performance of LLaDA and our ARM baselines trained on the same data across increasing computational FLOPs. LLaDA exhibits strong scalability, matching the overall performance of ARMs on six tasks.

Large Language Diffusion Models, Nie et al. 2025-02-14

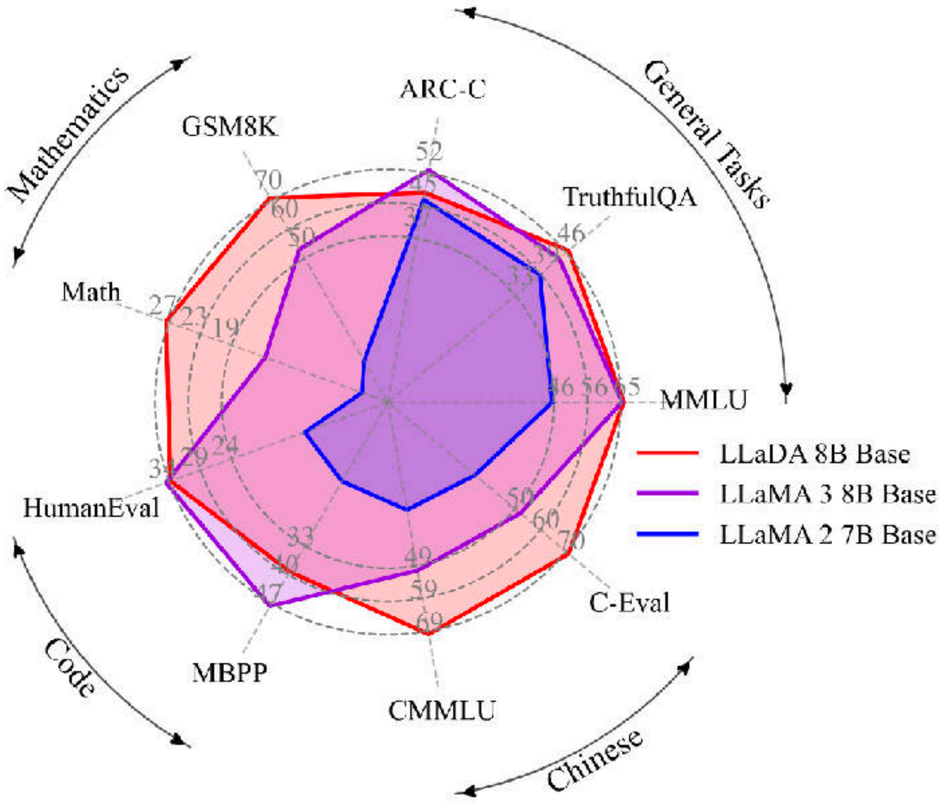


Figure 1. **Zero/Few-Shot Benchmarks.** We scale LLaDA to an unprecedented size of 8B parameters from scratch, achieving competitive performance with strong LLMs (Dubey et al., 2024).

BLOCK DIFFUSION: INTERPOLATING BETWEEN AUTOREGRESSIVE AND DIFFUSION LANGUAGE MODELS

Marianne Arriola[†*] **Aaron Kerem Gokaslan**[†] **Justin T. Chiu**[‡] **Zhihan Yang**[†]

Zhixuan Qi[†] **Jiaqi Han**[¶] **Subham Sekhar Sahoo**[†] **Volodymyr Kuleshov**[†]

ABSTRACT

Diffusion language models offer unique benefits over autoregressive models due to their potential for parallelized generation and controllability, yet they lag in likelihood modeling and are limited to fixed-length generation. In this work, we introduce a class of block diffusion language models that interpolate between discrete denoising diffusion and autoregressive models. Block diffusion overcomes key limitations of both approaches by supporting flexible-length generation and improving inference efficiency with KV caching and parallel token sampling. We propose a recipe for building effective block diffusion models that includes an efficient training algorithm, estimators of gradient variance, and data-driven noise schedules to minimize the variance. Block diffusion sets a new state-of-the-art performance among diffusion models on language modeling benchmarks and enables generation of arbitrary-length sequences. We provide the code[‡], along with the model weights and blog post on the project page:

https://m-arriola.com/bd3lms

1 INTRODUCTION

Diffusion models are widely used to generate images (Ho et al., 2020; Dhariwal & Nichol, 2021; Sahoo et al., 2024b) and videos (Ho et al., 2022; Gupta et al., 2023), and are becoming increasingly effective at generating discrete data such as text (Lou et al., 2024; Sahoo et al., 2024a) or biological sequences (Avdeyev et al., 2023; Goel et al., 2024). Compared to autoregressive models, diffusion models have the potential to accelerate generation and improve the controllability of model outputs (Schiff et al., 2024; Nisonoff et al., 2024; Li et al., 2024; Sahoo et al., 2024c).

Discrete diffusion models currently face at least three limitations. First, in applications such as chat systems, models must generate output sequences of arbitrary length (e.g., a response to a user's question). However, most recent diffusion architectures only generate fixed-length vectors (Austin et al., 2021; Lou et al., 2024). Second, discrete diffusion uses bidirectional context during generation and therefore cannot reuse previous computations with KV caching, which makes inference less efficient (Israel et al., 2025). Third, the quality of discrete diffusion models, as measured by standard metrics such as perplexity, lags behind autoregressive approaches and further limits their applicability (Gulrajani & Hashimoto, 2024; Sahoo et al., 2024a).

This paper makes progress towards addressing these limitations by introducing Block Discrete Denoising Diffusion Language Models (BD3-LMs), which interpolate between discrete diffusion and autoregressive models. Specifically, block diffusion models (also known as semi-autoregressive models) define an autoregressive probability distribution over blocks of discrete random variables (Si et al., 2022; 2023); the conditional probability of a block given previous blocks is specified by a discrete denoising diffusion model (Austin et al., 2021; Sahoo et al., 2024a).

Developing effective BD3-LMs involves two challenges. First, efficiently computing the training objective for a block diffusion model is not possible using one standard forward pass of a neural

[*]Correspondence to Marianne Arriola: marriola@cs.cornell.edu
[†]Cornell Tech, NY, USA. [¶]Stanford University, CA, USA. [‡] Cohere, NY, USA.
[‡]Code: https://github.com/kuleshov-group/bd3lms

Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models, Arriola et al. 2025-03-12

- **Hybrid Architecture**: Combines autoregressive modeling across blocks with diffusion within blocks for balanced performance.
- **Flexible Sequence Lengths**: Capable of generating sequences of arbitrary lengths, surpassing fixed-length limitations.
- **Efficient Inference**: Utilizes KV caching and parallel sampling to enhance inference speed and efficiency.
- **Stable Training**: Employs data-driven noise schedules to reduce gradient variance during training.

Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models, Arriola et al. 2025-03-12

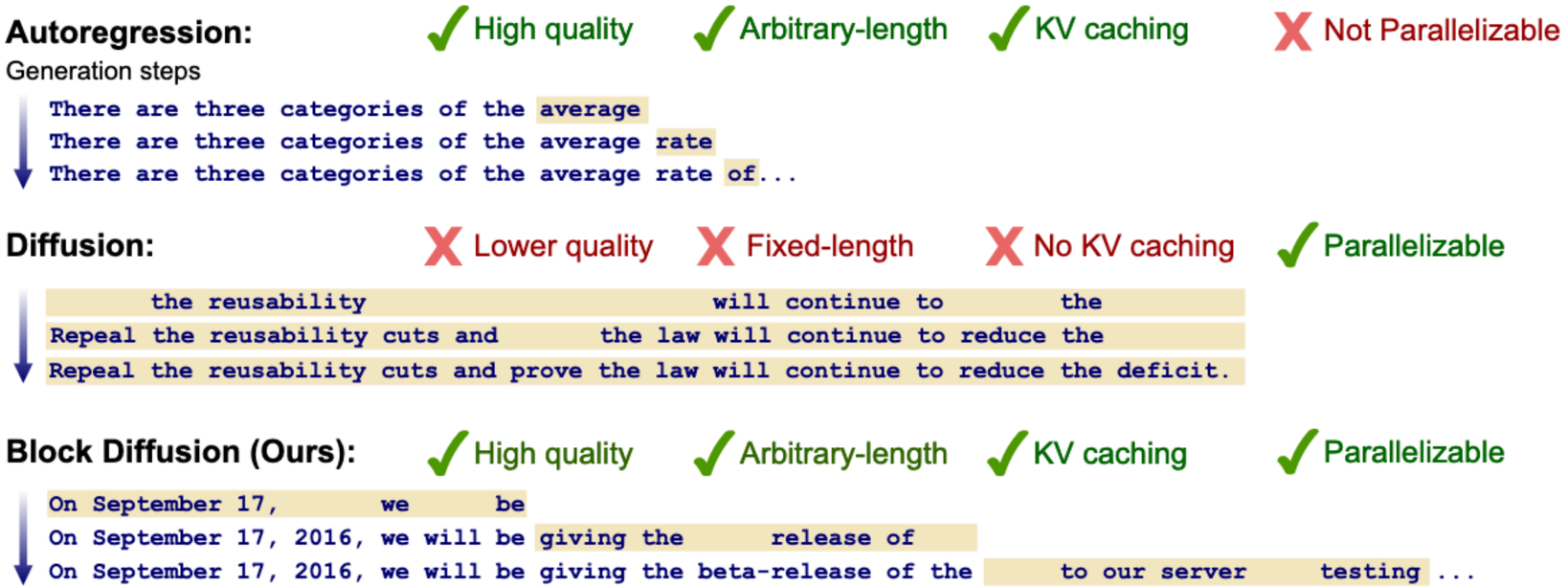


Figure 1: Block diffusion sequentially generates blocks of tokens by performing diffusion within each block and conditioning on previous blocks. By combining strength from autoregressive and diffusion models, block diffusion overcomes the limitations of both approaches by supporting variable-length, higher-quality generation and improving inference efficiency with KV caching and parallel sampling.

d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning

Siyan Zhao* UCLA
Devaansh Gupta* UCLA
Qinqing Zheng Meta AI
Aditya Grover UCLA

Abstract

Recent large language models (LLMs) have demonstrated strong reasoning capabilities that benefits from online reinforcement learning (RL). These capabilities have primarily been demonstrated within the left-to-right autoregressive (AR) generation paradigm. In contrast, non-autoregressive paradigms based on diffusion generate text in a coarse-to-fine manner. Although recent diffusion-based large language models (dLLMs) have achieved competitive language modeling performance compared to their AR counterparts, it remains unclear if dLLMs can also leverage recent advances in LLM reasoning. To this end, we propose *d1*, a framework to adapt pre-trained masked dLLMs into reasoning models via a combination of supervised finetuning (SFT) and RL. Specifically, we develop and extend techniques to improve reasoning in pretrained dLLMs: (a) we utilize a masked SFT technique to distill knowledge and instill self-improvement behavior directly from existing datasets, and (b) we introduce a novel critic-free, policy-gradient based RL algorithm called *diffu*-GRPO. Through empirical studies, we investigate the performance of different post-training recipes on multiple mathematical and logical reasoning benchmarks. We find that d1 yields the best performance and significantly improves performance of a state-of-the-art dLLM. Project page: https://dllm-reasoning.github.io/.

1 Introduction

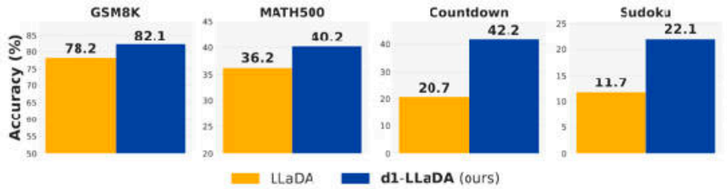


Figure 1: Across four math and logical reasoning tasks, **d1-LLaDA**, which undergoes SFT followed by our proposed ***diffu*-GRPO**, consistently outperforms the base LLaDA-8B-Instruct model. We report results using the best performing generation sequence length for each task and model, with complete sequence length results shown in Table 1.

Recent advances in large language models (LLMs) have demonstrated remarkable capabilities across diverse applications spanning chatbots, coding, summarization, and translation (Achiam et al., 2023; Dubey et al., 2024). While these models typically scale through next-token prediction on vast corpora via computationally intensive pretraining, the finite availability of high-quality training data poses a fundamental scaling challenge. Reinforcement learning (RL) methods have emerged as a promising post-training method, enabling

*Equal Contribution

d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning, Zhao et al. 2025-04-16

- **Two-Stage Training**: Combines supervised fine-tuning with reinforcement learning to enhance reasoning in diffusion-based language models.
- **diffu-GRPO Algorithm**: Introduces a novel, critic-free RL algorithm tailored for the non-sequential nature of diffusion models.
- **Improved Reasoning Performance**: Demonstrates significant gains on mathematical and logical reasoning benchmarks.

d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning, Zhao et al. 2025-04-16

For SFT, we train on s1k (Muennighoff et al., 2025) for 20 epochs, with a sequence length of 4096. For RL, we train a separate model for each task. More specifically, for GSM8K, MATH500, we train on the training split; for Countdown and Sudoku, we train on synthetic generated datasets. We use a composed reward function that combines both formatting and correctness rewards.

Table 1: **Model performance on GSM8K, MATH500, Countdown, and Sudoku Benchmarks:** All models are evaluated with 0-shot prompting, where the generation sequence length varies from 128 to 512. **Green values** indicate best performance and blue values indicate second-best performance in each column. The results demonstrate that **d1-LLaDA** consistently outperforms all other models, applying *diffu*-GRPO consistently improves the starting checkpoint, and *diffu*-GRPO alone shows better performance than SFT.

	GSM8K (0-shot)			MATH500 (0-shot)			Countdown (0-shot)			Sudoku (0-shot)		
Model / Seq Len	**128**	**256**	**512**	**128**	**256**	**512**	**128**	**256**	**512**	**128**	**256**	**512**
LLaDA-8B-Instruct	68.7	76.7	78.2	26.0	32.4	36.2	20.7	19.5	16.0	11.7	6.7	5.5
+ SFT	66.5	78.8	81.1	26.2	32.6	34.8	20.3	14.5	23.8	16.5	8.5	4.6
+ diffu-GRPO	72.6	79.8	81.9	33.2	37.2	39.2	33.2	31.3	37.1	18.4	12.9	**11.0**
+ SFT + diffu-GRPO (**d1-LLaDA**)	**73.2**	**81.1**	**82.1**	**33.8**	**38.6**	**40.2**	**34.8**	**32.0**	**42.2**	**22.1**	**16.7**	9.5

Key Ideas

- BigGAN and StyleGAN2 greatly advanced image generation but leaned toward realism over diversity due to their mode-seeking behavior.

- GANs struggled to generalize impressive domain-specific results (e.g., realistic faces) to more diverse datasets.

- VQ-VAE 2 and VQGAN promoted a two-stage generative modelling approach:
  - Compress images into discrete one-dimensional sequences.
  - Use autoregressive models to predict these sequences step-by-step.

- Although the concept originated with the original VQ-VAE, VQ-VAE 2 and VQGAN reinforced its value for large-scale, diverse data generation.

0 - Introduction
1 - Preliminaries - Generative Models
2 - Preliminaries - Generative Model Taxonomy
3 - Diffusion Model Foundations
4 - Language Diffusion Models

5 - End

Thank you!

References

Background on Diffusion

- Extracting and Composing Robust Features with Denoising Autoencoders, Vincent, Larochelle, Bengio 2008-01 https://dl.acm.org/doi/10.1145/1390156.1390294
- Generalized Denoising Auto-Encoders as Generative Models, Bengio et al. 2013-06-02 https://arxiv.org/abs/1305.6663
- Deep Unsupervised Learning using Nonequilibrium Thermodynamics,
  Sohl-Dickstein Et al. 2015-03-12 https://arxiv.org/abs/1503.03585
- Improved Techniques for Training Score-Based Generative Models,
  Song & Ermon 2020-06-16 https://arxiv.org/abs/2006.09011
- Denoising Diffusion Probabilistic Models,
  Ho Et al. 2020-06-19 https://arxiv.org/abs/2006.11239
- Score-Based Generative Modeling Through Stochastic Differential Equations,
  Song Et al. 2020-11-26 https://arxiv.org/abs/2011.13456

Language Diffusion Models

- Structured Denoising Diffusion Models in Discrete State-Spaces,
  Austin et al. 2021-07-13 https://arxiv.org/abs/2107.03006
- DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models,
  Gong et al. 2022-10-17 https://arxiv.org/abs/2210.08933
- Scaling up Masked Diffusion Models on Text, Nie et al. 2024-10-24 https://arxiv.org/abs/2410.18514
- Large Language Diffusion Models,
  Nie et al. 2025-02-14 https://arxiv.org/abs/2502.09992
- Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models,
  Arriola et al. 2025-03-12 https://arxiv.org/abs/2503.09573
- d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning,
  Zhao et al. 2025-04-16 https://arxiv.org/abs/2504.12216

Misc

- Gemini Diffusion https://deepmind.google/models/gemini-diffusion/
- Inception Labs Mercury https://www.inceptionlabs.ai/introducing-mercury
- Diffusion and Score-Based Generative Models, Yang Song, 2022-12-12 https://www.youtube.com/watch?v=wMmqCMwuM2Q
- Figure from What are Diffusion Models?, Lilian Wang, 2021-07-11 https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- Discrete Generative Modeling with Masked Diffusions, Jiaxin Shi, Google DeepMind 2024-09-18 https://jiaxins.io/talks/jiaxins_md4_genu.pdf

Appendix

Diffusion-LM Improves Controllable Text Generation

Xiang Lisa Li
Stanford University
xlisali@stanford.edu

John Thickstun
Stanford University
jthickst@stanford.edu

Ishaan Gulrajani
Stanford Univeristy
igul@stanford.edu

Percy Liang
Stanford Univeristy
pliang@cs.stanford.edu

Tatsunori B. Hashimoto
Stanford Univeristy
thashim@stanford.edu

Abstract

Controlling the behavior of language models (LMs) without re-training is a major open problem in natural language generation. While recent works have demonstrated successes on controlling simple sentence attributes (e.g., sentiment), there has been little progress on complex, fine-grained controls (e.g., syntactic structure). To address this challenge, we develop a new *non-autoregressive* language model based on *continuous* diffusions that we call Diffusion-LM. Building upon the recent successes of diffusion models in continuous domains, Diffusion-LM iteratively denoises a sequence of Gaussian vectors into word vectors, yielding a sequence of intermediate latent variables. The continuous, hierarchical nature of these intermediate variables enables a simple gradient-based algorithm to perform complex, controllable generation tasks. We demonstrate successful control of Diffusion-LM for six challenging fine-grained control tasks, significantly outperforming prior work.[1]

1 Introduction

Large autoregressive language models (LMs) are capable of generating high quality text [33, 3, 5, 45], but in order to reliably deploy these LMs in real world applications, the text generation process needs to be *controllable*: we need to generate text that satisfies desired requirements (e.g. topic, syntactic structure). A natural approach for controlling a LM would be to fine-tune the LM using supervised data of the form (control, text) [16]. However, updating the LM parameters for each control task can be expensive and does not allow for compositions of multiple controls (e.g. generate text that is both positive sentiment *and* non-toxic). This motivates light-weight and modular plug-and-play approaches [6] that keep the LM frozen and steer the generation process using an external classifier that measures how well the generated text satisfies the control. But steering a frozen autoregressive LM has been shown to be difficult, and existing successes have been limited to simple, attribute-level controls (e.g., sentiment or topic) [6, 22, 44].

In order to tackle more complex controls, we propose *Diffusion-LM*, a new language model based on *continuous* diffusions. Diffusion-LM starts with a sequence of Gaussian noise vectors and incrementally denoises them into vectors corresponding to words, as shown in Figure 1. These gradual denoising steps produce a hierarchy of continuous latent representations. We find that this hierarchical and continuous latent variable enables simple, gradient-based methods to perform complex control tasks such as constraining the parse tree of a generated sequence.

Continuous diffusion models have been extremely successful in vision and audio domains [12, 21, 34, 8, 4], but they have not been applied to text because of the inherently discrete nature of text

[1] Code is available at https://github.com/XiangLi1999/Diffusion-LM.git

Preprint. Under review.

Diffusion-LM Improves Controllable Text Generation
Li et al. 2022-05-27

Diffusion for Text in Continuous Space

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

Abstract

We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

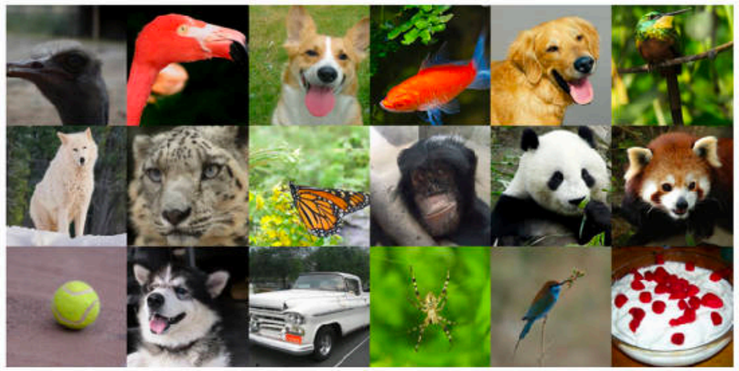


Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 28, 51] and highly diverse human speech and music [64, 13]. These models can be used in a variety of ways, such as generating images from text prompts [72, 50] or learning useful feature representations [14, 7]. While these models are already

*Equal contribution

Diffusion Beats GANs

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[R] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [R]Runway ML
https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [66,67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

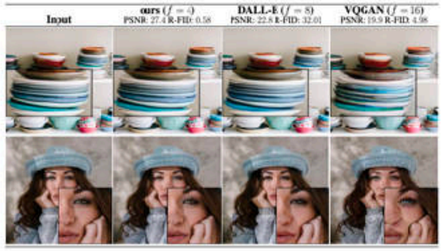


Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [30, 85] and beyond [7, 45, 48, 57], and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

*The first two authors contributed equally to this work.

Latent Diffusion Models (LDMs)

DIFFUSEQ: SEQUENCE TO SEQUENCE TEXT GENERATION WITH DIFFUSION MODELS

Shansan Gong[1], **Mukai Li**[1], **Jiangtao Feng**[1], **Zhiyong Wu**[1], **Lingpeng Kong**[1,2]
[1]Shanghai AI Lab [2]The University of Hong Kong
{gongshansan,limukai,fengjiangtao,wuzhiyong}@pjlab.org.cn
lpk@cs.hku.hk

ABSTRACT

Recently, diffusion models have emerged as a new paradigm for generative models. Despite the success in domains using continuous signals such as vision and audio, adapting diffusion models to natural language is difficult due to the discrete nature of text. We tackle this challenge by proposing DIFFUSEQ: a diffusion model designed for sequence-to-sequence (SEQ2SEQ) text generation tasks. Upon extensive evaluation over a wide range of SEQ2SEQ tasks, we find DIFFUSEQ achieving comparable or even better performance than six established baselines, including a state-of-the-art model that is based on pre-trained language models. Apart from quality, an intriguing property of DIFFUSEQ is its high diversity during generation, which is desired in many SEQ2SEQ tasks. We further include a theoretical analysis revealing the connection between DIFFUSEQ and autoregressive/non-autoregressive models. Bringing together theoretical analysis and empirical evidence, we demonstrate the great potential of diffusion models in complex conditional language generation tasks. [1]

1 INTRODUCTION

Existing generative models such as GAN (Goodfellow et al., 2014), VAE (Kingma & Welling, 2014) and Flow-based models (Dinh et al., 2017) suffer from the instability issue (Salimans et al., 2016), subjecting to mode collapse (Metz et al., 2017), and have to rely on surrogate objectives to approximate maximum likelihood training. Diffusion models (Ho et al., 2020; Nichol & Dhariwal, 2021) have circumvented several of these limitations and emerged as a new paradigm for generative models, theoretically underpinned by non-equilibrium thermodynamics (Sohl-Dickstein et al., 2015) and score-matching network (Song & Ermon, 2019). To date, the major breakthroughs are in domains using continuous signals, such as vision (Saharia et al., 2022a;b; Ramesh et al., 2022) and audio (Kong et al., 2020). However, extending continuous diffusion models to natural language remains an open challenge due to the inherently discrete nature of texts.

On the basis of unconditional generation in continuous space (illustrated in Figure 1(a)), existing efforts (Hoogeboom et al., 2021; Austin et al., 2021) start customizing diffusion models to text in discrete space on unconditional language modeling (i.e., free text generation). Diffusion-LM (Li et al., 2022) (as in Figure 1(b)) models texts in continuous space and proposes to use a extra-trained classifier as a guidance (i.e., the condition signal $\mathbf{x}$) to impose subtle changes (usually complex, fine-grained constraints) on generated sentences. Nonetheless, these models do not naturally generalized to conditional language modeling (i.e., the model assigns probabilities $p(\mathbf{w}|\mathbf{x})$ to sequences of words $\mathbf{w}$ given $\mathbf{x}$). In the more general sequence-to-sequence (SEQ2SEQ) setting where the condition $\mathbf{x}$ is also a sequence of words, applying Diffusion-LM can be difficult. The reason is that classifiers are attributes-oriented, and we can not train hundreds-of-thousands classifiers to model the semantic meaning between conditions and generated sentences.

SEQ2SEQ is an essential setting in NLP that covers a wide range of important tasks such as open-ended sentence generation, dialogue, paraphrasing, and text style transfer. In this paper, we propose DIFFUSEQ (in Figure 1 (c)), a classifier-free diffusion model that supports SEQ2SEQ text generation

[1]Code is available at https://github.com/Shark-NLP/DiffuSeq

DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models, Gong et al. 2022-10-17

- **Partial Noising**: Only the target sequence is noised, preserving the source for better conditional generation.
- **Classifier-Free Training**: Simplifies the model by eliminating the need for external classifiers.
- **Bridges AR and NAR**: Theoretically connects diffusion models with existing sequence modeling approaches.
- **High Diversity Outputs**: Generates varied and creative text, enhancing the richness of generated content.

DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models, Gong et al. 2022-10-17

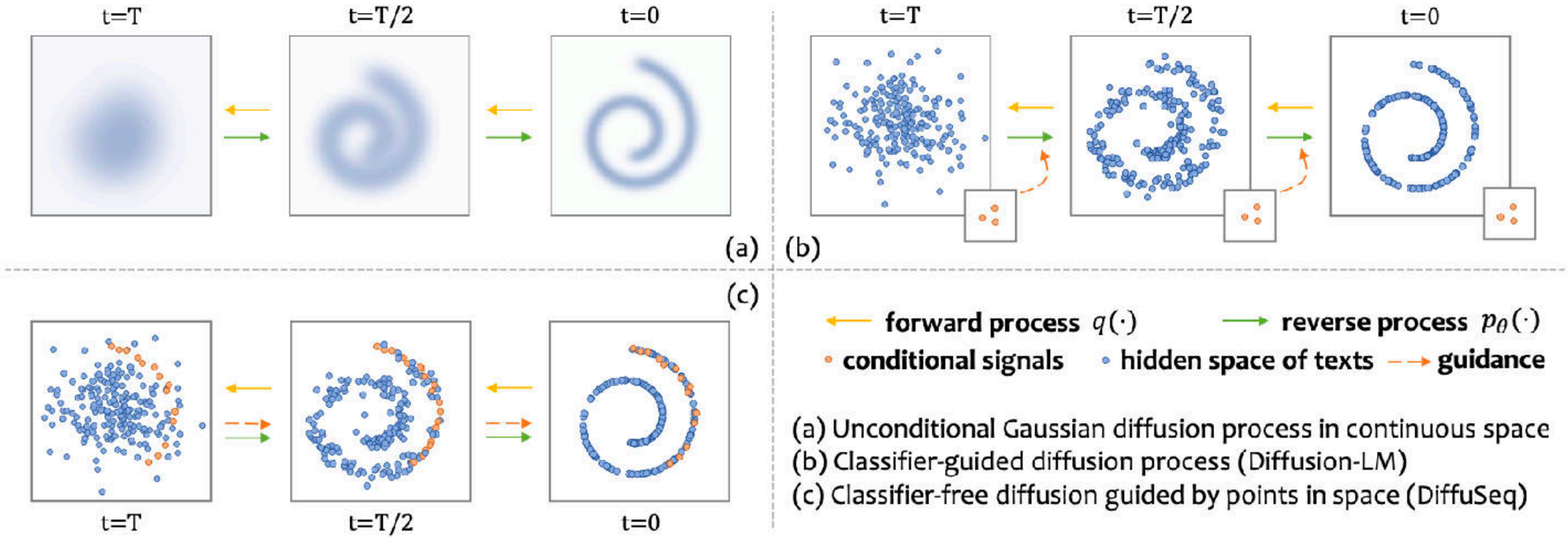


Figure 1: The demonstration of unconditional, classifier-guided and classifier-free diffusion models.

DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models, Gong et al. 2022-10-17

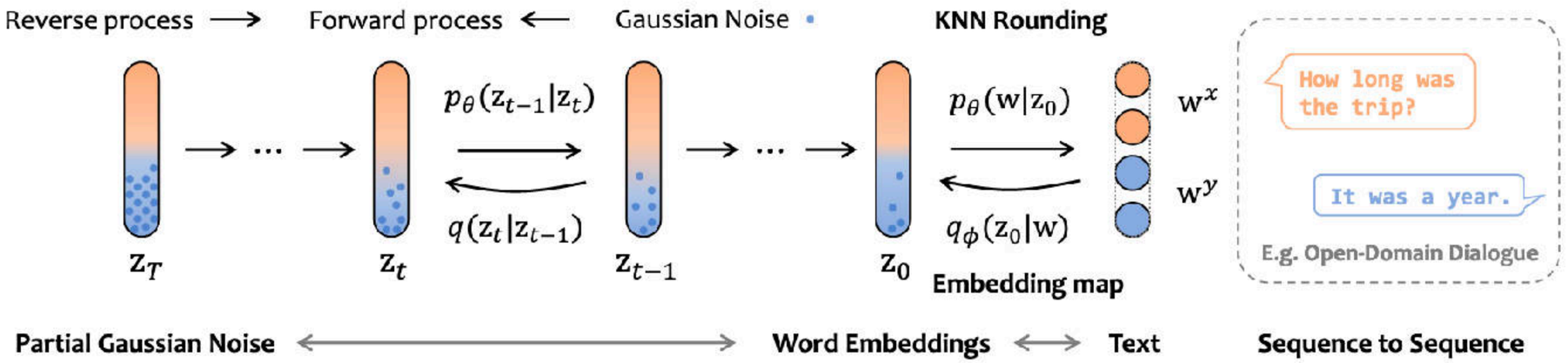


Figure 2: The diffusion process of our conditional diffusion language model DIFFUSEQ. Given the source $\mathbf{w}^x$ and the target $\mathbf{w}^y$, we pair-wisely transform them into continuous space $\mathbf{z}_0$. The partial Gaussian noise is iteratively added on the target space of $\mathbf{z}_t$.

DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models, Gong et al. 2022-10-17

Table 2: Sample outputs in QQP test set, conditioned on the same **x**.

Original sentence: *How do I make friends.*		***Paraphrase reference***: *How to make friends ?*
GPT2-large finetune	**GPVAE-T5**	**DIFFUSEQ**
How can I make friends?	How can I make friends?	How can I make friends better?
How can I make friends?	How do I make friends?	How can I make friends?
How can I make friends?	How can I make friends?	How do you make friends?
How can I make friends?	How can I make friends?	What is the best way to make friends?
How do I make friends and keep them?	What's the best way to make friends and make make friends?	How can I make friends and more something?

DiffuSeq-v2: Bridging Discrete and Continuous Text Spaces for Accelerated Seq2Seq Diffusion Models

Shansan Gong[1] **Mukai Li**[1] **Jiangtao Feng**[2] **Zhiyong Wu**[3] **Lingpeng Kong**[1]
[1]The University of Hong Kong [2]Independent Researcher [3]Shanghai AI Lab
sansa933@connect.hku.hk jiangtaofeng0906@gmail.com lpk@cs.hku.hk

Abstract

Diffusion models have gained prominence in generating high-quality sequences of text. Nevertheless, current approaches predominantly represent discrete text within a continuous diffusion space, which incurs substantial computational overhead during training and results in slower sampling speeds. In this paper, we introduce a soft absorbing state that facilitates the diffusion model in learning to reconstruct discrete mutations based on the underlying Gaussian space, thereby enhancing its capacity to recover conditional signals. During the sampling phase, we employ state-of-the-art ODE solvers within the continuous space to expedite the sampling process. Comprehensive experimental evaluations reveal that our proposed method effectively accelerates the training convergence by 4x and generates samples of similar quality 800x faster, rendering it significantly closer to practical application. [1]

1 Introduction

After diffusion models gained significant attention in the vision domain (Ho et al., 2020), progress has been made in applying them to text generation tasks, including constrained text generation in Diffusion-LM (Li et al., 2022) and sequence-to-sequence (Seq2Seq) text generation in DiffuSeq (Gong et al., 2023). Numerous subsequent studies demonstrate that diffusion models have achieved results comparable to traditional autoregressive models and non-autoregressive models in tasks such as machine translation (Yuan et al., 2022; Gao et al., 2022; Zheng et al., 2023) and summarization (Lin et al., 2022; Zhang et al., 2023; Mahabadi et al., 2023; Zhou et al., 2023). However, most of these works suffer from slow convergence during training and slow generation speed, particularly considering that these approaches require the Minimum Bayes Risk (MBR) decoding strategy (Koehn, 2004) to enhance generation quality, resulting in a doubling of time consumption.

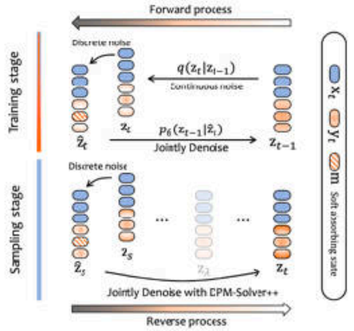


Figure 1: Training and sampling stages with discrete noise, which helps the two stages align better.

In order to further narrow the gap between diffusion models and the prevailing autoregressive models, we aim to propose an accelerated version of DiffuSeq for both the training and sampling stages. For training, in addition to GPU acceleration techniques such as FP16, an improved training scheme can enable the model to better represent knowledge and learn data distribution more quickly (Hang et al., 2023). For sampling, a well-trained diffusion model can achieve similar quality within a single sampling and without the need for MBR decoding, thus saving generation time. Furthermore, we can borrow the state-of-the-art ODE sampler DPM-solver++ (Lu et al., 2022a,b) which is already applied in fast image generation. The progress in discrete text diffusion models (Zheng et al., 2023) exhibits its superiority in using fewer sampling steps, thus intuitively inspiring us to bridge the gap between continuous and discrete spaces.

Based on the BERT (Zhang et al., 2019) and

[1]The code is released at https://github.com/Shark-NLP/DiffuSeq/tree/diffuseq-v2.

Scalable Diffusion Models with Transformers

William Peebles*
UC Berkeley

Saining Xie
New York University

Figure 1. **Diffusion models with transformer backbones achieve state-of-the-art image quality.** We show selected samples from two of our class-conditional DiT-XL/2 models trained on ImageNet at 512×512 and 256×256 resolution, respectively.

Abstract

We explore a new class of diffusion models based on the transformer architecture. We train latent diffusion models of images, replacing the commonly-used U-Net backbone with a transformer that operates on latent patches. We analyze the scalability of our Diffusion Transformers (DiTs) through the lens of forward pass complexity as measured by Gflops. We find that DiTs with higher Gflops—through increased transformer depth/width or increased number of input tokens—consistently have lower FID. In addition to possessing good scalability properties, our largest DiT-XL/2 models outperform all prior diffusion models on the class-conditional ImageNet 512×512 and 256×256 benchmarks, achieving a state-of-the-art FID of 2.27 on the latter.

1. Introduction

Machine learning is experiencing a renaissance powered by transformers. Over the past five years, neural architectures for natural language processing [8, 42], vision [10] and several other domains have largely been subsumed by transformers [60]. Many classes of image-level generative models remain holdouts to the trend, though—while transformers see widespread use in autoregressive models [3, 6, 43, 47], they have seen less adoption in other generative modeling frameworks. For example, diffusion models have been at the forefront of recent advances in image-level generative models [9, 46]; yet, they all adopt a convolutional U-Net architecture as the de-facto choice of backbone.

* Work done during an internship at Meta AI, FAIR Team. Code and project page available here.

Diffusion Transformers (DiT)