JULY 15, 2026 / #AI

AI Paper Review: Deep Unsupervised Learning using Nonequilibrium Thermodynamics



Mohammed Fahd Abrah

Today, diffusion models power some of the most impressive AI systems ever built. They generate photorealistic images, create videos, synthesize speech, design proteins, and increasingly influence fields far beyond computer vision.

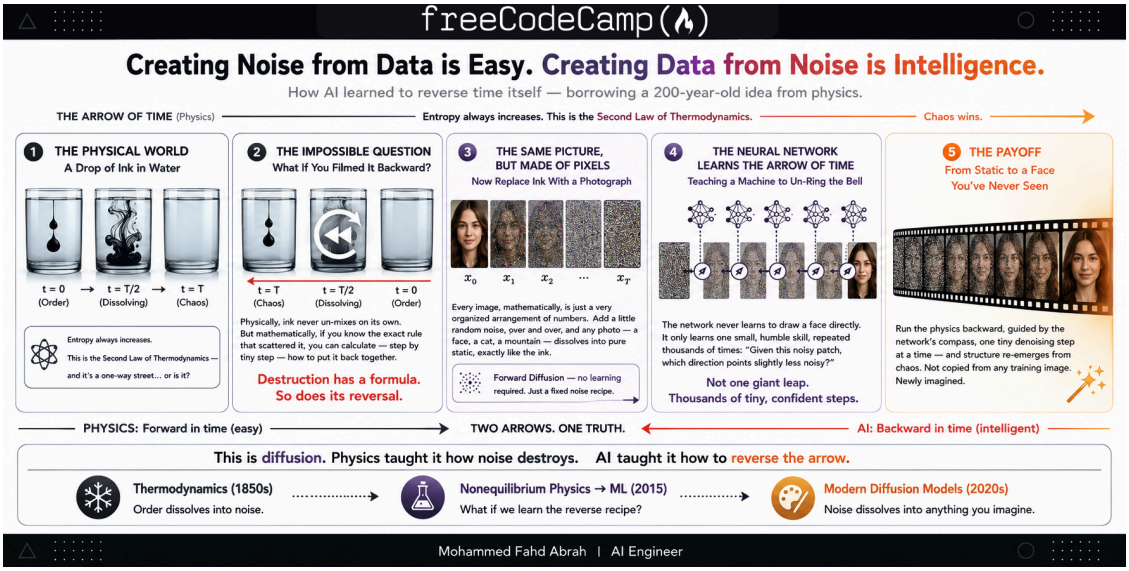
Models such as Stable Diffusion, DALL·E, and many modern generative systems all trace their roots back to a single question: How can a model learn an incredibly complex data distribution without becoming mathematically intractable?

For decades, this question remained one of the biggest obstacles in generative modeling. Researchers faced an uncomfortable trade-off. Models that were easy to train and evaluate were often too simple to capture the richness of real-world data. More expressive models could represent complex distributions, but they were notoriously difficult to optimize, sample from, or even evaluate. Countless methods attempted to narrow this gap, yet none fully escaped it.

In 2015, Jascha Sohl-Dickstein and his collaborators proposed a remarkably different way of thinking about the problem. Instead of trying to learn a complex data distribution directly, they asked a surprisingly simple question: What if we first destroyed the data by gradually adding noise, then learned how to reverse that process?

That single idea transformed a seemingly impossible learning problem into a sequence of small, manageable prediction tasks.

The infographic below provides an intuitive overview of the core idea behind diffusion models, showing how a concept borrowed from thermodynamics became the foundation of modern generative AI.



At the time, the paper attracted relatively little attention compared with other breakthroughs in deep learning. But looking back, it represents one of the most important turning points in the history of generative AI. It introduced the first practical formulation of diffusion probabilistic models (DPM), laying the mathematical foundation for the diffusion revolution that would reshape AI several years later.

In this review, we'll explore the paper step by step, from the motivation behind the idea, through the diffusion algorithm itself, to the experiments that demonstrated its potential. We'll see why this overlooked work ultimately became the starting point of one of the most influential families of generative models in modern AI.

Paper Overview

Deep Unsupervised Learning using Nonequilibrium Thermodynamics (2015) introduced the first practical formulation of diffusion probabilistic models, establishing the foundation for the diffusion models that dominate generative AI today.

Rather than learning a complex data distribution directly, the paper proposes a two-stage process: a forward diffusion process that gradually transforms data into simple noise, and a reverse diffusion process that learns to reconstruct the original data from that noise.

Drawing inspiration from nonequilibrium statistical physics, the authors show that this formulation overcomes the long-standing trade-off between expressive generative models and computational tractability. The framework enables efficient training, exact sampling, likelihood evaluation, and posterior inference within a single probabilistic model.

To validate the approach, the paper demonstrates strong results on synthetic datasets, MNIST, CIFAR-10, and natural image benchmarks, while also showcasing practical applications such as image denoising and inpainting.

Although initially overlooked, this pioneering work laid the mathematical groundwork for later breakthroughs such as DDPMs and score-based diffusion models, making it one of the most influential papers in the history of modern generative AI.

And here's a quick infographic of what we'll cover throughout this review, highlighting the paper's main ideas, methodology, key findings, and lasting impact.

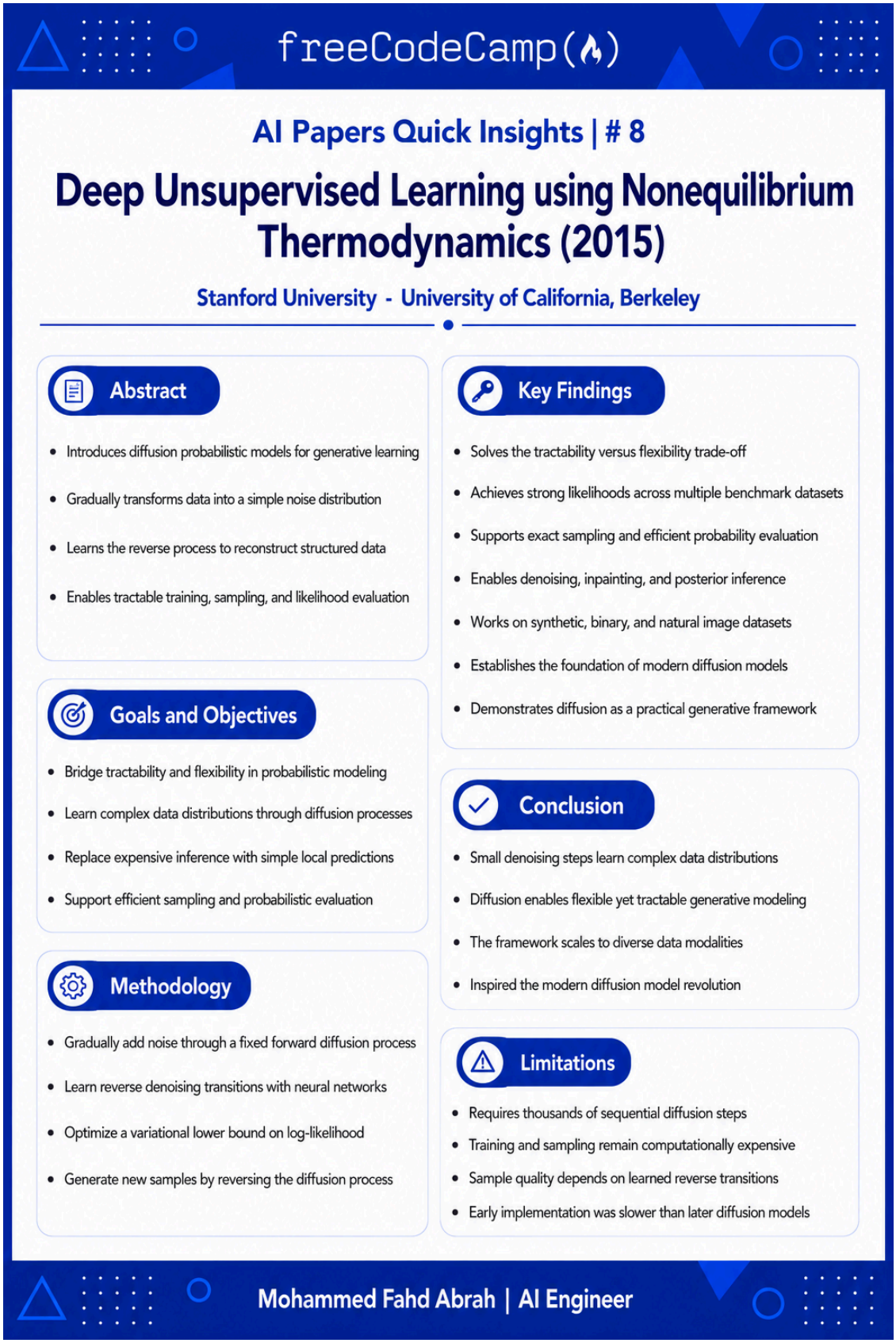


Table of Contents:

- Abstract
- Introduction
    - 1.1 Diffusion Probabilistic Models

    - 1.2 Relationship to Other Work
- 2. Algorithm
    - 2.1 Forward Trajectory
    - 2.2 Reverse Trajectory
    - 2.3 Model Probability
    - 2.4 Training
    - 2.4.1 Setting the Diffusion Rate
    - 2.5 Multiplying Distributions and Computing Posteriors
    - 2.5.1 Modified Marginal Distributions
    - 2.5.2 Modified Diffusion Steps
    - 2.5.3 Applying the Conditioning Function
    - 2.5.4 Choosing the Conditioning Function
    - 2.6 Entropy of the Reverse Process
- 3. Experiments
    - 3.1 Toy Problems
    - 3.1.2 Binary Heartbeat Distribution
    - 3.2 Images
    - 3.2.1 Datasets
- 4. Conclusion
- Resources

Prerequisites

This review assumes a basic understanding of probability, linear algebra, and deep learning fundamentals. Familiarity with Markov chains, Gaussian distributions, and generative models such as VAEs

or GANs will help you appreciate the paper's contributions, but no prior knowledge of diffusion models is required.

Throughout the review, we'll build the key ideas step by step.

Abstract

By 2015, generative modeling faced a familiar trade-off: models were usually either expressive enough to capture complex real-world data or simple enough to train and evaluate efficiently, but rarely both. This paper set out to bridge that gap by introducing a new kind of generative model inspired by nonequilibrium thermodynamics.

The core idea is surprisingly intuitive. Instead of learning the data distribution directly, the model first gradually corrupts the data through a forward diffusion process, adding small amounts of noise until only a simple, well-understood distribution remains. It then learns the reverse process, step by step, transforming pure noise back into realistic data.

This formulation gives the model an unusual combination of strengths. It remains flexible enough to model complex datasets while keeping learning, sampling, likelihood evaluation, and posterior inference computationally tractable.

The framework scales naturally to thousands of diffusion steps, laying the foundation for what would later evolve into modern diffusion models. To encourage further research, they released an open-source reference implementation alongside the paper.

Introduction

The paper begins by highlighting one of the oldest challenges in probabilistic machine learning: the trade-off between tractability and flexibility. In practice, researchers had to choose between models that were easy to work with and models that were powerful enough to represent complex data, but rarely both.

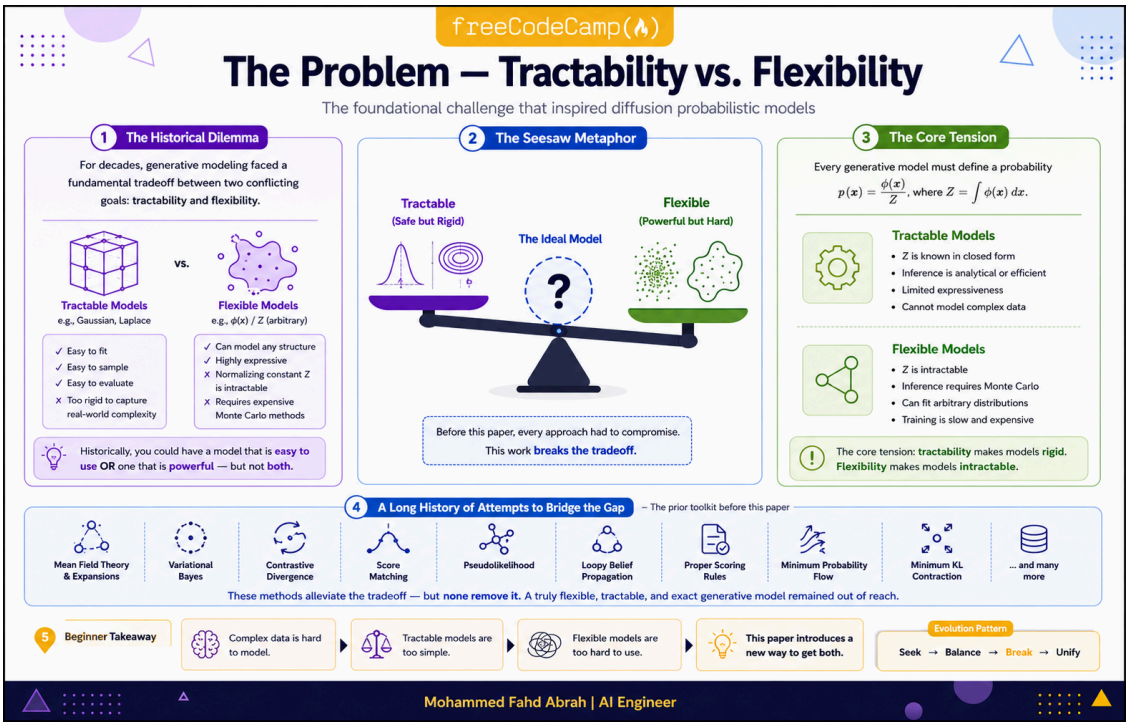
Simple probability distributions, such as Gaussian or Laplace distributions, are mathematically convenient. They're easy to train, evaluate, and sample from, making them attractive for practical applications. The downside is that they struggle to capture the rich, highly structured patterns found in real-world data like images, audio, and text.

At the opposite extreme are highly flexible models, which can represent almost any data distribution. Their limitation is computational rather than expressive: evaluating probabilities requires computing a normalization constant that's often intractable. As a result, training and sampling typically rely on expensive Monte Carlo methods, making these models difficult to use at scale.

The authors acknowledge that many techniques had already been proposed to narrow this gap, including variational inference, contrastive divergence, score matching, pseudolikelihood, belief propagation, and several other approximation methods.

While these approaches improved the situation, they didn't fundamentally eliminate the trade-off. This unresolved problem was the motivation for the diffusion framework introduced in the rest of the paper.

The infographic below summarizes the central dilemma that motivated this paper, along with the major approaches researchers explored before diffusion probabilistic models emerged.



1.1 Diffusion Probabilistic Models

After introducing the long-standing trade-off between flexibility and tractability, the paper presents its solution: diffusion probabilistic models. Its goal is ambitious: to build a generative model that is expressive, supports exact sampling, allows efficient posterior computation, and still makes likelihood evaluation practical.

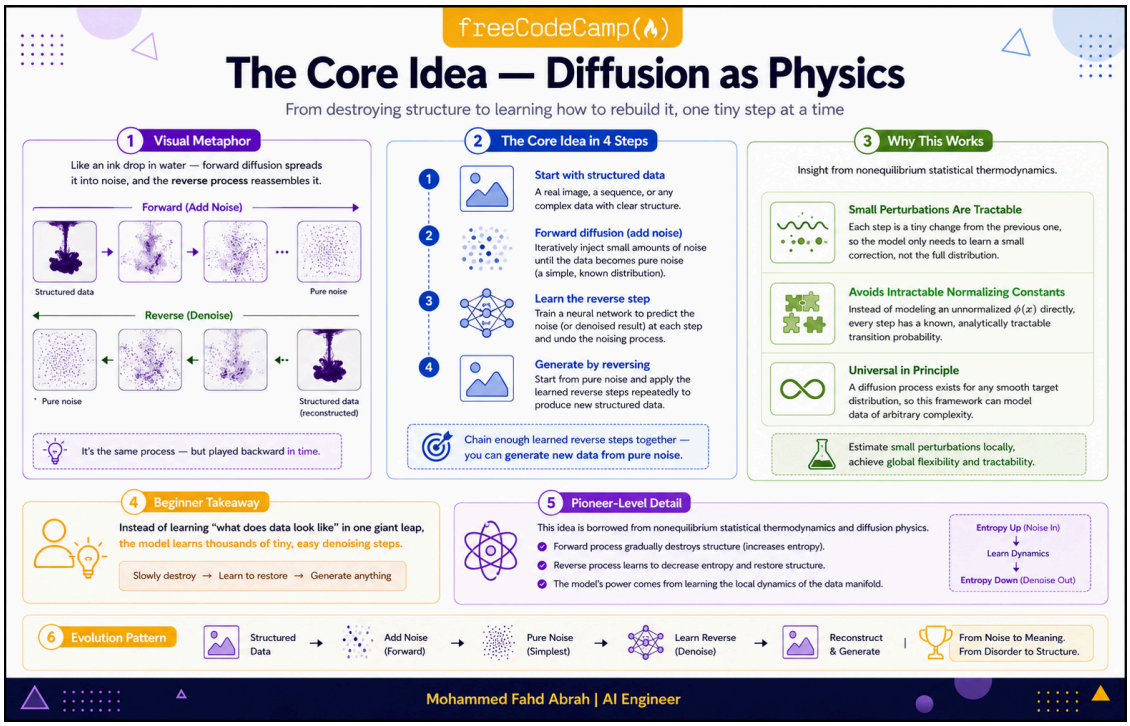
The key insight is to model data as a Markov diffusion process. Instead of learning a complex data distribution directly, the model starts from a simple distribution, such as a Gaussian, and gradually transforms it into the target data distribution through many small diffusion steps. Rather than treating this Markov chain as merely a computational tool, the chain itself becomes the probabilistic model. Because every transition has a tractable probability, the entire process remains analytically manageable.

Training also becomes simpler. Instead of learning one highly complicated distribution all at once, the model only needs to learn the small changes that occur between consecutive diffusion steps. These local transformations are much easier to estimate while

remaining expressive enough to approximate virtually any smooth data distribution.

To demonstrate the versatility of the framework, they evaluate it on a diverse collection of datasets, ranging from simple synthetic data and binary sequences to handwritten digits (MNIST) and natural images, including CIFAR-10, bark textures, and dead leaves. This broad evaluation shows that the same diffusion framework can be applied across very different data domains.

The infographic below illustrates the core intuition behind diffusion probabilistic models, showing how data is gradually transformed into noise before the learned reverse process reconstructs it.



1.2 Relationship to Other Work

Before introducing the algorithm in detail, the paper places its contribution within the broader landscape of generative modeling. It acknowledges that many recent methods, particularly Variational Autoencoders (VAEs) had already made significant progress by jointly learning generative models and inference networks.

While its training objective shares similarities with the variational lower bound used in those methods, the underlying philosophy is fundamentally different.

Instead of building on variational Bayesian inference, the proposed framework is rooted in statistical physics. It draws inspiration from nonequilibrium thermodynamics, quasi-static processes, and annealed importance sampling.

This perspective leads to several practical advantages: it naturally supports posterior computation by combining distributions, simplifies the design of the forward and reverse processes by giving them the same functional form, scales to thousands of diffusion steps, and provides theoretical bounds on entropy throughout the diffusion process.

The authors also place diffusion models alongside many of the leading generative approaches of the time, including Wake-Sleep, Generative Stochastic Networks (GSNs), Neural Autoregressive Distribution Estimators (NADEs), Generative Adversarial Networks (GANs), invertible generative models, Bayesian network inversion methods, and Gaussian scale mixture models.

Rather than presenting diffusion as a replacement for these methods, they argue that it offers a different path toward the same goal: learning expressive probability distributions while preserving tractable inference and sampling. The experimental section later compares diffusion models directly with several of these approaches, particularly GANs and MCGSMs.

Finally, the paper highlights the physics concepts that inspired the framework. Ideas such as Annealed Importance Sampling (AIS), Langevin dynamics, the Fokker–Planck equation, and the Kolmogorov forward and backward equations provide the

theoretical foundation for viewing generation as the reversal of a diffusion process.

This connection between machine learning and statistical physics would later become one of the defining characteristics of diffusion models.

2. Algorithm

With the motivation established, the paper turns to the core of the method: the diffusion algorithm itself. The central idea is to frame generative modeling as learning the reversal of a carefully designed diffusion process.

Instead of modeling the data distribution directly, the method first defines a forward diffusion process that gradually transforms complex data into a simple, tractable distribution, typically Gaussian noise. It then learns to reverse this process, reconstructing realistic data from noise one small step at a time.

The remainder of this section develops the framework piece by piece. It begins by defining the forward diffusion process, then explains how to learn the reverse process that serves as the generative model.

The authors next show how this formulation enables efficient likelihood evaluation, derive entropy bounds for the reverse trajectory, and demonstrate that the learned model can be combined with other probability distributions to perform tasks such as posterior inference, image denoising, and inpainting.

Together, these components establish the complete mathematical foundation of diffusion probabilistic models.

2.1 Forward Trajectory

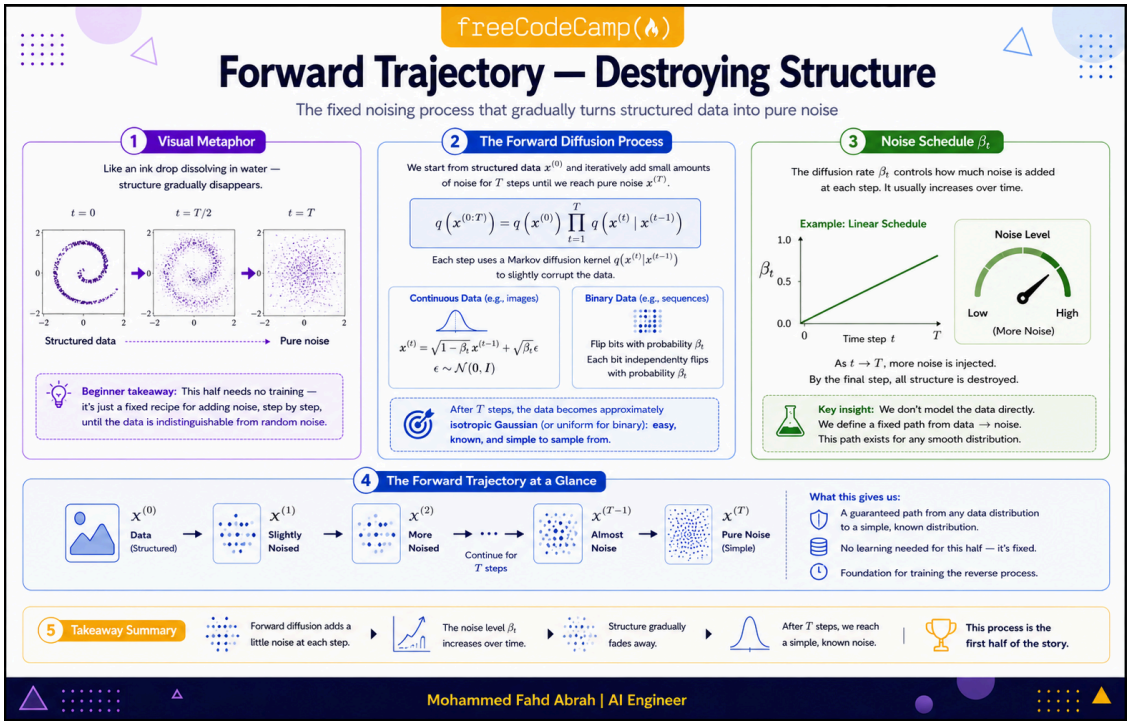
The algorithm begins with an unusual idea: instead of learning how to generate data, it first learns how to destroy it. Starting from the original data distribution, a small amount of noise is added at every step until the rich structure of the data gradually disappears, eventually becoming a simple distribution such as standard Gaussian noise. Because each noising step is small and follows a Markov process, the entire transformation remains easy to analyze while preparing the model for the much harder task of learning how to reverse it.

A key design choice is that each diffusion step makes only a small perturbation to the data. Individually, these changes are almost imperceptible, but after many steps, the accumulated effect completely removes the original structure. Because every transition depends only on the previous state, a defining property of a Markov chain, the entire forward process is mathematically simple and tractable.

The authors experiment with two types of diffusion processes. For continuous data, such as images, they use Gaussian diffusion, which progressively transforms data into standard Gaussian noise. For discrete data, such as binary sequences, they use binomial diffusion, where bits are randomly flipped until the sequence becomes independent random noise.

This demonstrates that the same diffusion framework can naturally handle both continuous and discrete data distributions.

The infographic below illustrates how the forward diffusion process gradually transforms structured data into pure noise through a sequence of small, fixed noising steps.



2.2 Reverse Trajectory

If the forward trajectory gradually destroys structure by turning data into noise, the reverse trajectory does exactly the opposite: it learns how to reconstruct meaningful data from that noise.

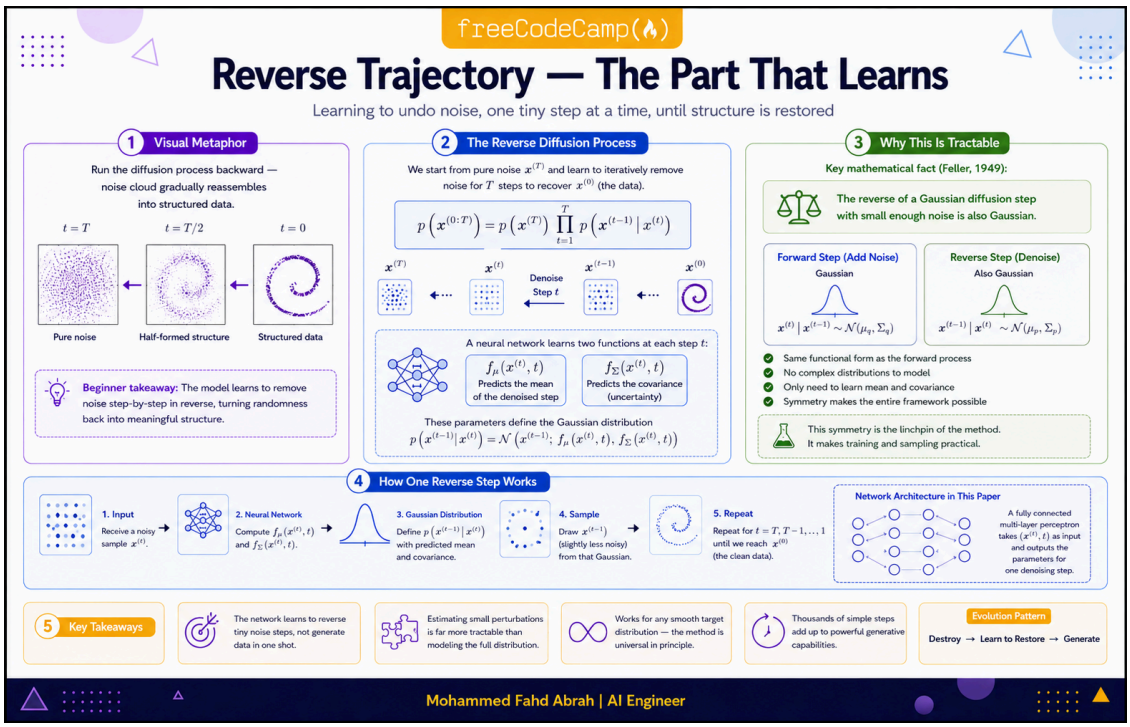
This reverse process is the actual generative model. Starting from a simple noise distribution, the model repeatedly removes a small amount of noise at each step until it recovers a realistic sample from the target data distribution.

A crucial observation is that when the forward diffusion steps are sufficiently small, the reverse process has the same mathematical form as the forward process. This symmetry greatly simplifies learning. Rather than solving a completely different problem, the model only needs to estimate the parameters that describe each reverse transition. For continuous data, these are the mean and covariance of the Gaussian distribution, while for discrete data they correspond to the bit-flip probabilities of the binomial distribution.

The authors implement these reverse transition functions using multi-layer perceptron layers (MLPs), although they note that any suitable regression or function approximation method could be

used. As a result, the computational cost scales with the complexity of these prediction functions and the number of diffusion steps, making the reverse trajectory both practical to train and flexible enough to model complex data distributions.

The infographic below illustrates how the reverse diffusion process learns to remove noise step by step, gradually reconstructing structured data from pure noise.



2.3 Model Probability

Learning to generate realistic data is only half the challenge. A generative model should also be able to tell you how likely that data is.

For diffusion models, this turns out to be surprisingly difficult because computing the exact probability means accounting for every possible diffusion path.

Rather than tackling this enormous calculation head-on, the paper borrows an elegant idea from statistical physics, drawing on Annealed Importance Sampling (AIS) and the Jarzynski equality to

turn an intractable problem into one that can be estimated efficiently.

The central idea is to compare the forward diffusion trajectory, which gradually corrupts the data, with the reverse trajectory, which reconstructs it. By averaging over trajectories produced by the forward process, the model can efficiently estimate the probability assigned to the original data without explicitly evaluating every possible path. This transforms an otherwise intractable computation into one that is computationally manageable.

The framework becomes even more elegant when the diffusion steps are very small. In this limit, the forward and reverse trajectories become nearly identical. Under this quasi-static condition, a single sample from the forward diffusion process is sufficient to compute the probability exactly, providing both a practical estimation method and a direct connection between diffusion models and principles from statistical physics.

2.4 Training

Once the forward and reverse diffusion processes are defined, the next challenge is learning the parameters of the reverse process. The objective is straightforward: maximize the likelihood of the training data under the generative model. But computing the exact likelihood remains difficult because it depends on every possible diffusion trajectory.

To overcome this, the authors derived a tractable lower bound on the log-likelihood using Jensen's inequality, much like the variational objectives used in VAEs. Instead of optimizing the true likelihood directly, the model optimizes this lower bound, which can be computed analytically through a combination of KL divergence

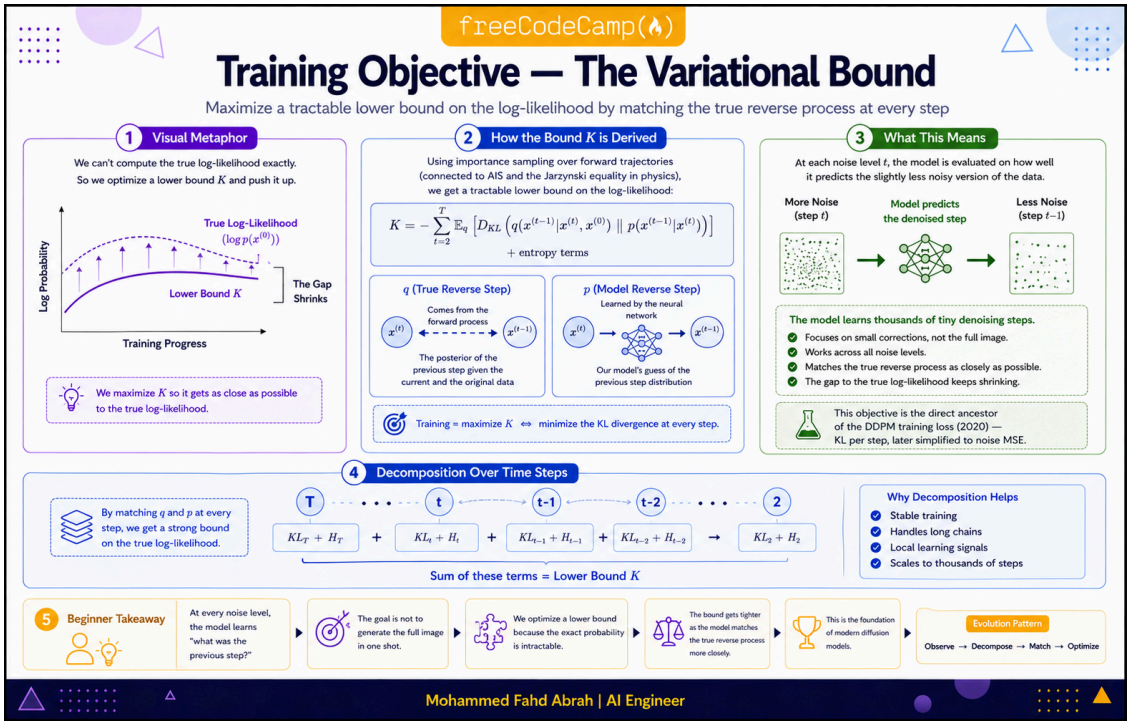
terms and entropy terms. As the forward and reverse diffusion processes become increasingly similar, this lower bound becomes progressively tighter, eventually matching the true likelihood in the ideal quasi-static limit.

Perhaps the most elegant aspect of the training procedure is how it simplifies the learning problem. Rather than learning an entire high-dimensional probability distribution at once, the model only learns the small reverse transition at each diffusion step.

In practice, this reduces generative modeling to a sequence of ordinary regression problems: predicting the mean and covariance of Gaussian transitions for continuous data or the bit-flip probabilities for binary data.

This decomposition turns an otherwise intractable density estimation problem into a collection of much simpler prediction tasks, making diffusion models both scalable and practical to train.

The infographic below illustrates the intuition behind the training objective, showing how the model maximizes a tractable lower bound on the log-likelihood by learning the reverse diffusion process one step at a time.



2.4.1 Setting the Diffusion Rate

The paper emphasize that how quickly noise is added during the forward diffusion process is just as important as the diffusion process itself.

If too much noise is introduced too early, valuable information is lost before the model can learn meaningful reverse transitions. If too little noise is added, the diffusion process becomes unnecessarily long and inefficient.

Finding the right diffusion schedule is therefore critical to the model's performance.

For Gaussian diffusion, the diffusion rates are treated as learnable parameters. The model optimizes the noise schedule during training, while keeping the first diffusion step intentionally small to reduce overfitting.

To make this optimization stable, the authors adopted the "frozen noise" technique introduced in variational autoencoders, where the sampled noise is held fixed while computing gradients. This allows

the diffusion schedule to be learned efficiently through gradient-based optimization.

Things become a little different for binomial diffusion. Because the data is discrete rather than continuous, the optimization strategy used for Gaussian diffusion no longer applies. Instead of learning the diffusion schedule, the paper adopts a simple rule: each step removes the same fraction of the remaining signal. The result is a smooth, gradual loss of information, allowing the data to fade into noise without abrupt changes along the way

2.5 Multiplying Distributions and Computing Posteriors

One of the most practical advantages of diffusion probabilistic models is that they make posterior inference remarkably straightforward. Many real-world tasks, such as image denoising, inpainting, or inferring missing values, require combining the learned data distribution with additional information or constraints.

In many existing generative models, performing this multiplication is difficult or computationally expensive. The proposed work shows that diffusion models avoid this limitation.

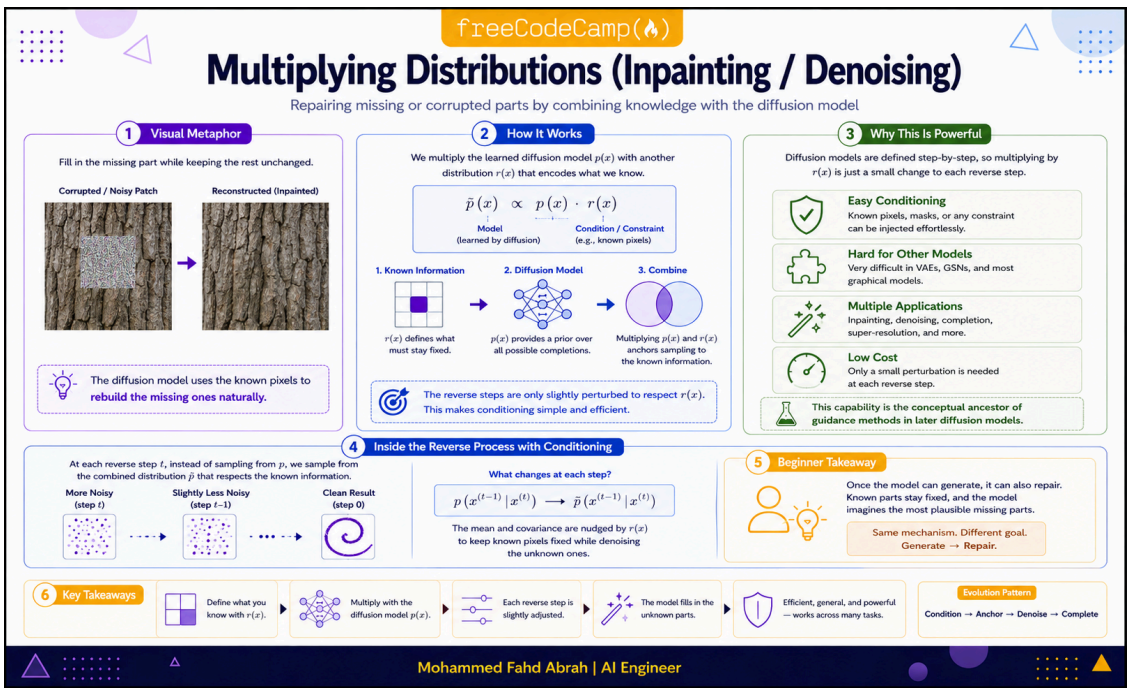
The key idea is simple. Instead of combining distributions only after generation is complete, the additional information is incorporated throughout the reverse diffusion process. At every diffusion step, the model slightly adjusts its estimate using a second distribution or constraint.

Because each reverse transition is already a small update, these modifications naturally integrate into the generation process without fundamentally changing the algorithm. This makes

conditioning the model on known information much more efficient than in many earlier generative frameworks.

This capability is more than a theoretical convenience. It allows the same diffusion model to solve practical inference tasks, such as reconstructing corrupted images or filling in missing regions, by simply guiding the reverse diffusion process with the available observations. As later experiments demonstrate, the framework performs both denoising and image inpainting using this mechanism.

The infographic below illustrates how diffusion models incorporate known information during the reverse process, enabling tasks such as image denoising and inpainting through conditional generation.



2.5.1 Modified Marginal Distributions

To incorporate external information, the authors introduce a modified version of the reverse diffusion trajectory. Instead of relying solely on the learned model distribution, every intermediate distribution is multiplied by a corresponding conditioning function. This produces a new sequence of distributions that remains

consistent throughout the reverse diffusion process while incorporating the desired constraints.

Conceptually, this means the model doesn't wait until the final generation step to enforce known information. Instead, it guides the sample continuously as it evolves from noise into data, allowing conditioning information to influence every stage of generation.

This elegant formulation is one of the reasons diffusion models later became so effective for guided image generation and other conditional generation tasks.

2.5.2 Modified Diffusion Steps

Updating the intermediate distributions isn't enough. The reverse diffusion process has to follow those changes as well. Otherwise, the model would be generating samples from a different distribution than the one it's trying to learn.

To keep everything aligned, each reverse Markov transition is adjusted so it remains consistent with the newly conditioned distributions. As generation unfolds, the model naturally balances what it has learned from the data with the additional constraints introduced during conditioning.

The resulting formulation preserves the structure of the original diffusion algorithm. Rather than redesigning the reverse process, each diffusion step is simply reweighted and normalized to reflect the conditioning information. This means the model can incorporate external knowledge while maintaining a valid probability distribution throughout the reverse trajectory.

For Gaussian diffusion, the update becomes even simpler. Because each reverse step has only a very small variance, the conditioning term acts as a small perturbation rather than a major change. In

practice, this mainly shifts the predicted mean while leaving the normalization essentially unchanged. As a result, conditioning can be incorporated with very little additional computational cost, making guided generation both efficient and mathematically elegant.

2.5.3 Applying the Conditioning Function

The next question is how this conditioning information is actually incorporated into the reverse diffusion process.

When the conditioning signal changes smoothly, it only nudges each reverse diffusion step rather than altering the algorithm itself. The overall procedure stays exactly the same – the model simply adjusts the parameters of each transition.

For Gaussian diffusion, this means slightly shifting the predicted mean, while for binomial diffusion it comes down to making small adjustments to the bit-flip probabilities.

Some conditioning functions are even more convenient. When they can be combined analytically with Gaussian or binomial distributions, they can be incorporated exactly into each reverse diffusion step rather than approximately.

The authors highlight image inpainting as an example: known pixels are treated as fixed constraints, while the missing regions continue to evolve through the reverse diffusion process. This allows the model to reconstruct only the unknown parts of the image while preserving the observed content.

This section illustrates an important property of diffusion models: conditioning isn't an additional module layered on top of the model. Instead, it becomes an integral part of the reverse diffusion process,

enabling tasks such as inpainting and guided generation with minimal changes to the underlying algorithm.

2.5.4 Choosing the Conditioning Function

The final step is deciding how the conditioning information should evolve during the reverse diffusion process. They recommend that the conditioning function change gradually across the trajectory rather than introducing abrupt constraints. This keeps the reverse diffusion process stable and allows the generated samples to adapt smoothly as noise is removed.

For most of the experiments in the paper, the conditioning function is kept constant throughout the entire reverse process. The paper also describes an alternative schedule in which the conditioning influence gradually decreases as the diffusion process moves toward the initial noise distribution.

An advantage of this second approach is that it leaves the starting noise distribution unchanged, making it just as easy to sample the initial noisy state before guided generation begins.

Although this design choice may seem like a minor implementation detail, it reflects a broader principle of diffusion models: conditioning should guide generation progressively rather than forcing it abruptly.

By introducing constraints smoothly over the reverse trajectory, the model preserves both stability and sample quality while remaining easy to initialize and sample from.

2.6 Entropy of the Reverse Process

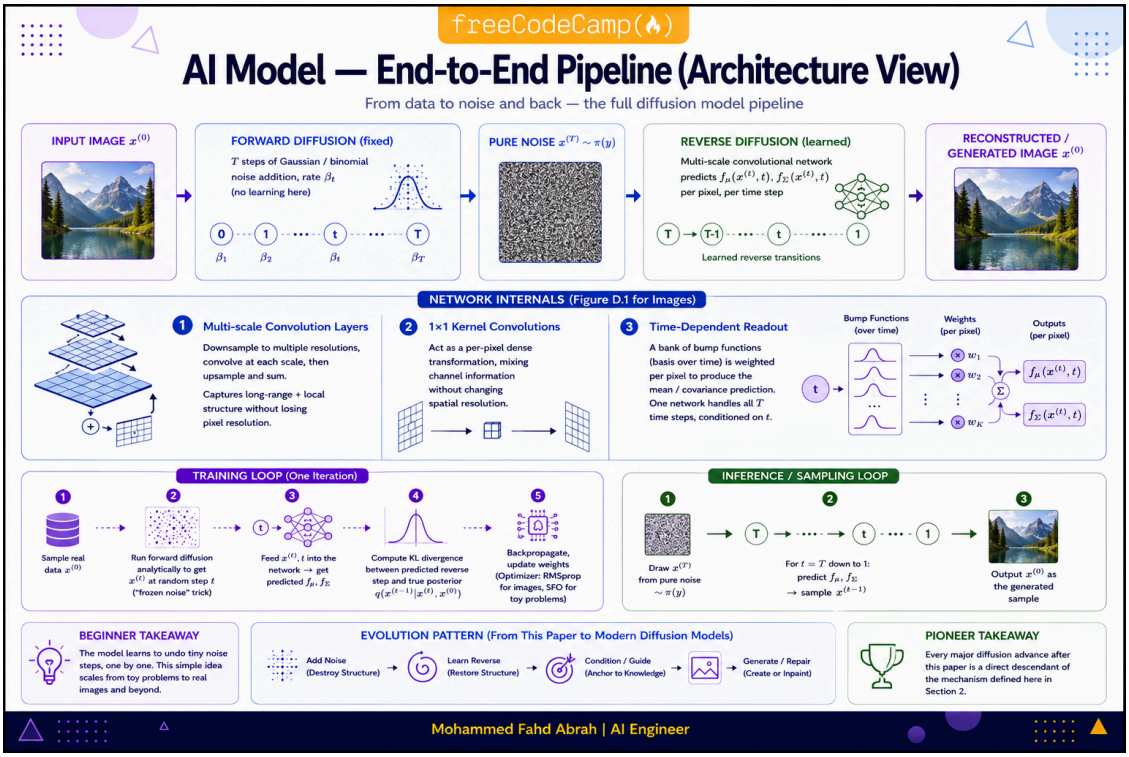
One question still remains: how much uncertainty is left as the model gradually reconstructs data from noise?

Because the forward diffusion process is completely known, it's possible to place both upper and lower bounds on the entropy of every reverse diffusion step. Those same bounds also extend to the model's log-likelihood, providing a theoretical lens for understanding how uncertainty shrinks as the reverse process unfolds.

An important advantage of this result is that these entropy bounds depend only on the forward diffusion process, which is fully specified by design. This means they can be computed analytically without requiring additional approximations or expensive estimation procedures.

While this section is primarily theoretical, it strengthens the foundation of diffusion probabilistic models. Beyond providing an effective generative algorithm, the framework also offers mathematical guarantees about the behavior of the reverse process, reinforcing its connection to both probabilistic modeling and statistical physics.

The infographic below summarizes the complete diffusion model pipeline, bringing together the forward process, reverse process, network architecture, training loop, and sampling procedure described throughout Section 2.



3. Experiments

After building the theoretical foundation, the paper shifts from theory to practice. The real question now is whether diffusion probabilistic models can learn meaningful distributions beyond mathematical examples.

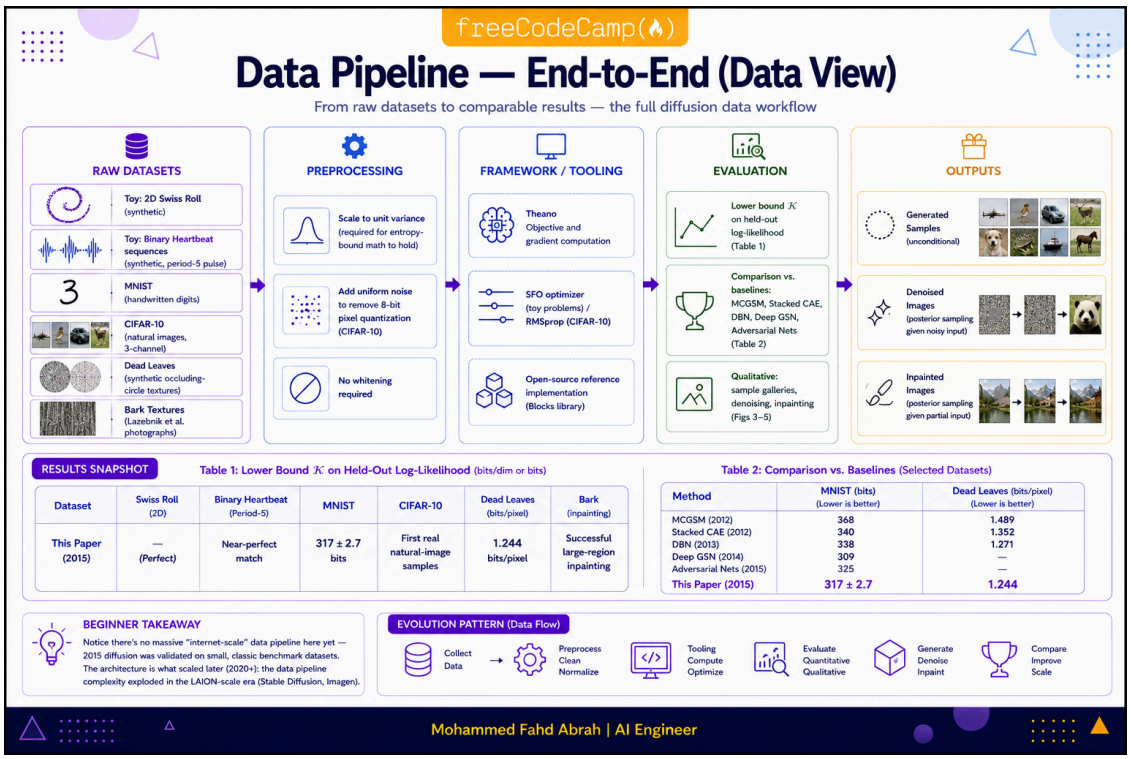
To answer that, the framework is evaluated on a diverse collection of datasets, ranging from binary sequences to natural images. The experiments go beyond generating new samples, demonstrating that the same diffusion process can also tackle practical tasks such as image inpainting, highlighting its ability to handle both unconditional generation and conditional inference within a single framework.

To measure performance, the paper reports the lower bound on the log-likelihood achieved by each trained model. The evaluation includes a wide range of datasets: Swiss Roll, Binary Heartbeat, Bark textures, Dead Leaves, CIFAR-10, and MNIST. Across all of them, the diffusion models consistently improve over simple baseline distributions, demonstrating that the framework

successfully learns meaningful data distributions in both synthetic and real-world settings.

The paper also measures diffusion models against the leading generative approaches of the time and provides an open-source implementation so others can reproduce the results. More importantly, the experiments show that the framework is far more than a theoretical idea. It learns high-quality data distributions across a wide range of domains while naturally supporting tasks such as sampling, likelihood evaluation, denoising, and image inpainting.

The infographic below summarizes the complete experimental pipeline used in the paper, from dataset preparation and preprocessing to training, evaluation, and generated outputs.



3.1 Toy Problems

Before moving to natural images, the paper starts with two simple toy problems. These small-scale experiments serve as an initial proof of concept, showing that the diffusion framework can

accurately learn well-understood probability distributions before taking on more challenging real-world data.

They also provide an intuitive look at how the forward and reverse diffusion processes work together, confirming that the training procedure behaves as expected.

3.1.1 Swiss Roll

The first experiment uses the classic Swiss Roll dataset, a two-dimensional synthetic distribution widely used to evaluate generative models. The reverse diffusion process is parameterized with a radial basis function (RBF) network, which predicts the mean and covariance of each reverse diffusion step.

The learned reverse process successfully reconstructs the characteristic spiral structure of the Swiss Roll from noise, demonstrating that the model can accurately recover a complex nonlinear distribution.

While the Swiss Roll is a simple benchmark, it plays an important role in the paper. Before tackling handwritten digits and natural images, the framework first proves that it can learn and reproduce a structured low-dimensional distribution.

That early success builds confidence that the diffusion process behaves as intended, setting the stage for the more challenging experiments that follow.

3.1.2 Binary Heartbeat Distribution

The second toy experiment shifts from continuous data to a discrete binary sequence, demonstrating that the diffusion framework isn't limited to Gaussian data.

The dataset consists of sequences of length 20, where a value of 1 appears every fifth position and all remaining positions are 0. To learn the reverse diffusion process, the authors use a multi-layer perceptron (MLP) to predict the probability of each bit flipping at every diffusion step.

This experiment highlights the flexibility of the framework. Instead of predicting Gaussian means and variances, as in continuous diffusion, the model learns Bernoulli transition probabilities, showing that the same diffusion principle naturally extends to discrete data.

The results are nearly perfect. The learned model achieves a log-likelihood that closely matches the true underlying distribution, indicating that it successfully captures the simple periodic structure of the binary sequences.

Although this is a synthetic benchmark, it demonstrates that diffusion probabilistic models can accurately model both continuous and discrete probability distributions using the same underlying framework.

3.2 Images

With the toy problems behind it, the paper turns to a much more demanding challenge: natural images. Here, Gaussian diffusion probabilistic models are trained on several image datasets, learning to reverse the gradual noising process until meaningful images emerge from pure Gaussian noise.

This marks the first real test of whether the framework can handle the complexity and rich structure of real-world visual data.

To handle the complexity of image data, all image experiments share the same multi-scale convolutional architecture. Rather than

designing a different model for each dataset, the authors use a unified architecture capable of capturing image features at multiple spatial scales. This consistent design demonstrates that the diffusion framework is general enough to work across diverse image domains without requiring major architectural changes.

With this common architecture in place, the paper evaluates the model on several image benchmarks, including MNIST, CIFAR-10, Dead Leaves, and Bark textures, to assess its ability to generate, restore, and model increasingly complex visual data.

3.2.1 Datasets

To evaluate the proposed framework on real-world image generation tasks, the authors conduct experiments on four datasets that vary in complexity, ranging from handwritten digits to natural textures. Together, these datasets demonstrate both the versatility of diffusion probabilistic models and their ability to generalize across different types of visual data.

1. MNIST

The authors first train their model on the MNIST handwritten digit dataset to enable direct comparisons with previous generative models.

Since earlier studies commonly estimated log-likelihood using Parzen-window evaluation, they adopt the same evaluation procedure to ensure a fair comparison.

The results show that diffusion models achieve performance comparable to the strongest methods available at the time while providing a principled likelihood-based training objective.

2. CIFAR-10

The framework is then evaluated on the much more challenging CIFAR-10 dataset. Despite the increased complexity of natural images, the trained diffusion model is able to generate realistic samples, demonstrating that the approach extends well beyond simple handwritten digits.

3. Dead Leaves

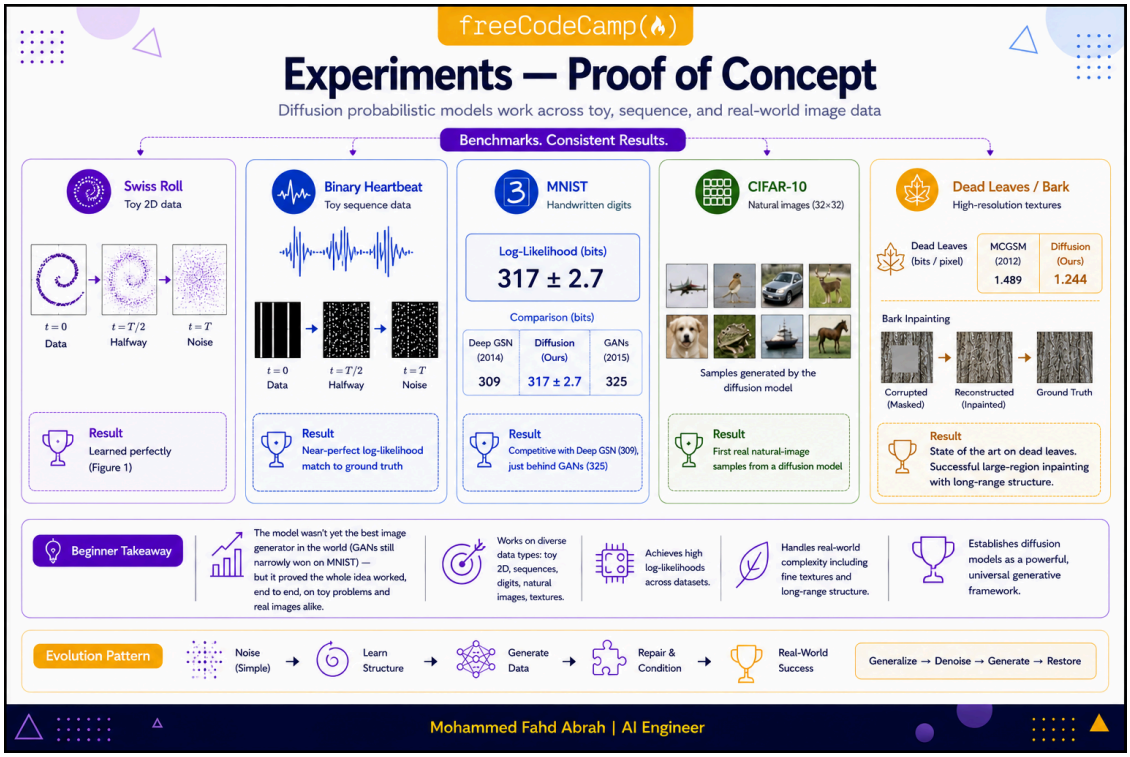
The Dead Leaves dataset provides an interesting intermediate benchmark. Although it's synthetically generated, it captures many of the statistical properties of natural images, such as occlusion and objects appearing at multiple scales.

On this benchmark, the diffusion model achieves state-of-the-art log-likelihood performance, outperforming previous approaches while accurately modeling the dataset's complex spatial structure.

4. Bark Texture Images

Finally, the authors train the model on bark texture images to showcase one of the framework's most practical capabilities: posterior inference. By conditioning the reverse diffusion process on the known pixels of an image, the model successfully fills in large missing regions through image inpainting, illustrating how diffusion models naturally support conditional generation in addition to unconditional sampling.

The infographic below summarizes the paper's experimental results, highlighting how diffusion probabilistic models performed across synthetic datasets, handwritten digits, and natural images.



4. Conclusion

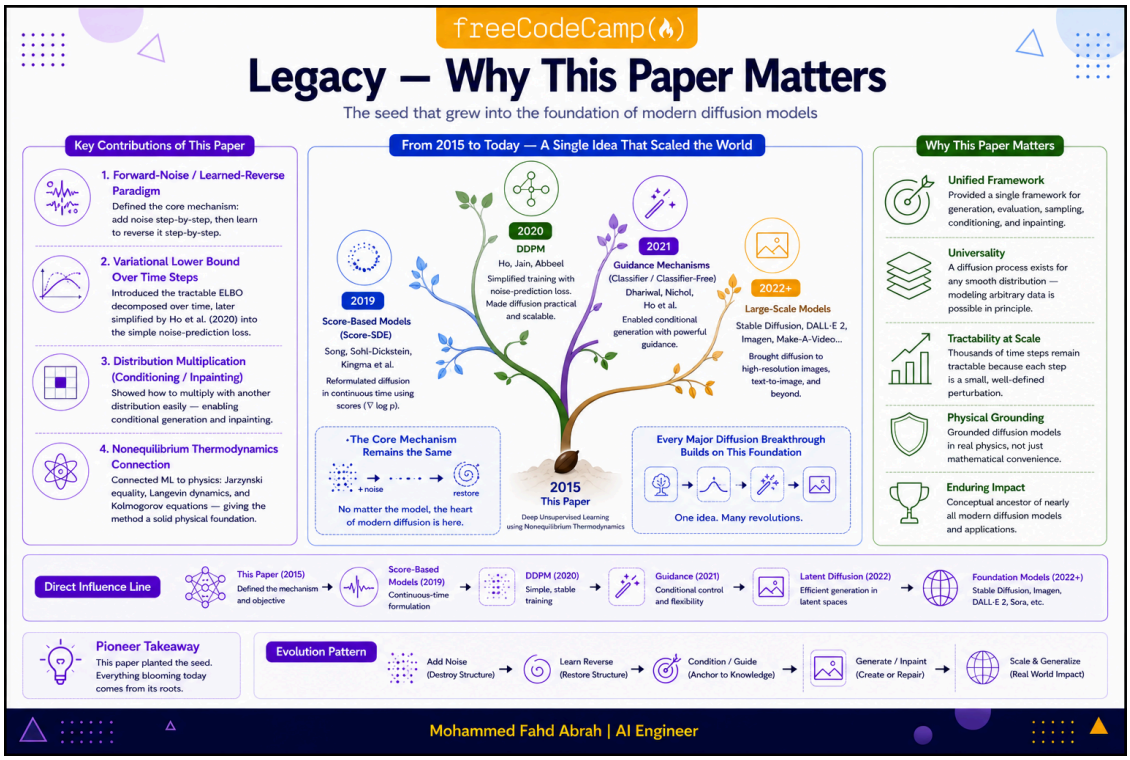
The paper concludes by showing that diffusion probabilistic models successfully overcome one of the central challenges in generative modeling: achieving both expressive modeling power and computational tractability.

Instead of relying on a single complex probability distribution, the framework learns to reverse a Markov diffusion process that gradually transforms data into noise. As the diffusion process is broken into many small steps, each reverse transition becomes simple enough to estimate accurately, making the overall learning problem far more manageable.

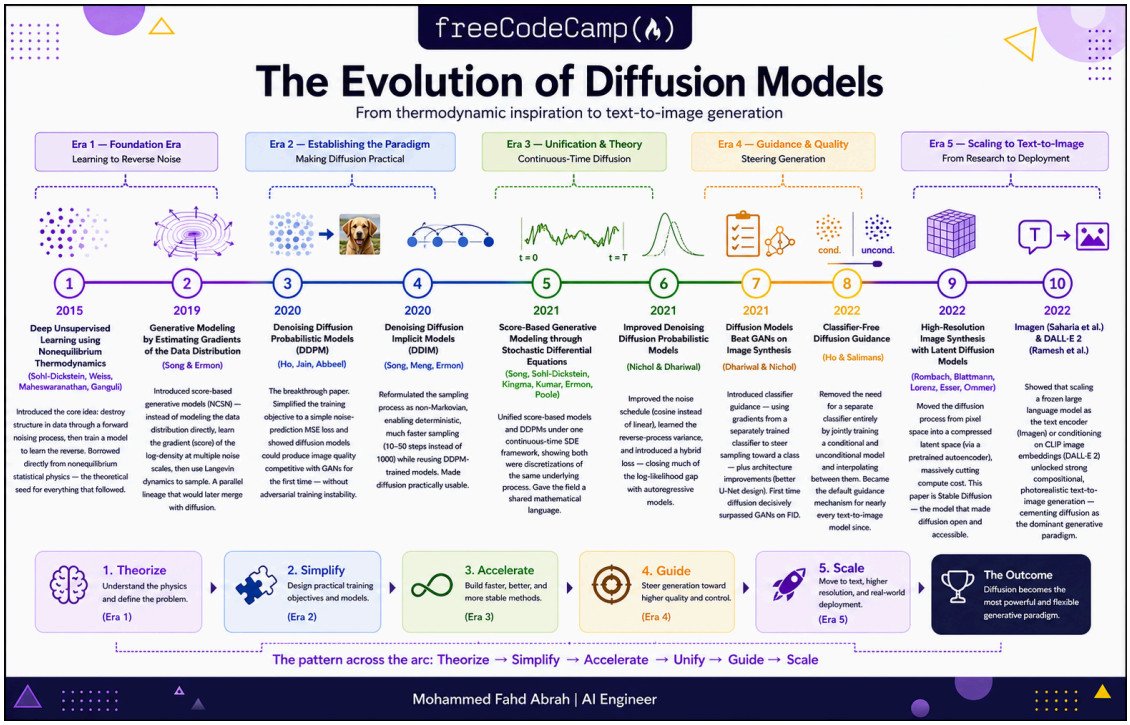
Across both synthetic and real-world datasets, the same core algorithm demonstrates strong performance without requiring major changes to the underlying framework. This consistency suggests that diffusion probabilistic models provide a general approach to density estimation rather than a solution tailored to a specific domain.

Perhaps the paper's most enduring contribution is its demonstration that a generative model can be tractable to train, capable of exact sampling, efficient to evaluate, and naturally suited for conditional inference. Although the generated images in this early work were modest compared to today's diffusion models, the paper established the mathematical and conceptual foundation that later advances, including DDPMs and score-based diffusion models, would build upon.

The infographic below summarizes the paper's lasting contributions and shows how its core ideas became the foundation for modern diffusion models.



Finally, to place this paper in context, the timeline below traces the key milestones that transformed its original diffusion framework into the powerful generative models driving modern AI today.



Resources:

- PyTorch Diffusion Implementations (GitHub)
- Auto-Encoding Variational Bayes (Kingma & Welling, 2013)
- Generative Adversarial Networks (Goodfellow et al., 2014)
- Annealed Importance Sampling (Neal, 2001)
- Estimation of Non-Normalized Statistical Models using Score Matching (Hyvärinen, 2005)
- Deep Generative Stochastic Networks (Bengio & Thibodeau-Laufer, 2013)
- NICE: Non-linear Independent Components Estimation (Dinh et al., 2014)
- The Wake-Sleep Algorithm (Hinton et al., 1995)
- Equilibrium Free-Energy Differences from Nonequilibrium Measurements (Jarzynski, 1997)
- Stochastic Backpropagation and Approximate Inference in Deep Generative Models (Rezende et al., 2014)

- Training Products of Experts by Minimizing Contrastive Divergence (Hinton, 2002)



Mohammed Fahd Abrah

Data Scientist- AI Engineer - Machine Learning and Deep Learning Engineer

---

ADVERTISEMENT