AN EXPLORATION OF INFORMATION PROCESSING IN DIFFUSION MODELS

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Paul Jason Mello

May 2024

© 2024

Paul Jason Mello

ALL RIGHTS RESERVED

The Designated Thesis Committee Approves the Thesis Titled

AN EXPLORATION OF INFORMATION PROCESSING IN DIFFUSION MODELS

by

Paul Jason Mello

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2024

Stas Tiomkin, Ph.D.	Department of Computer Engineering
Jorjeta Jetcheva, Ph.D.	Department of Computer Engineering
Carlos Rojas, Ph.D.	Department of Computer Engineering

ABSTRACT

AN EXPLORATION OF INFORMATION PROCESSING IN DIFFUSION MODELS

by Paul Jason Mello

Denoising diffusion probabilistic models have emerged as a powerful class of density modeling techniques. Characterized by their foundations in non-equilibrium thermodynamics, they are capable of modeling complex data distributions and generating novel samples. Their success in high quality sampling has resulted in significant research in sampling efficiency, improved estimation, and model control. Information theory provides tools for the exploration of diffusion models generative dynamics. Specifically, we explore two domains; Information-imbalanced data sets and score functions for recommendation systems. Through our exploration we observe an interesting phenomenon where certain classes of training data are more likely to be reconstructed than others. We propose information-theoretic reasoning as to why this phenomenon emerges across data sets and posit potential solutions to counteract this observation. We then apply denoising diffusion probabilistic models to recommender systems. We introduce a Score-based Diffusion Recommender Module (SDRM) to generate synthetic data for recommendation systems which accurately captures the sparse nature of this training data, while respecting user privacy. We show our generated samples are capable of fully replacing and or augmenting the initial training data, while boosting recommender model performance by an average improvement of 4.5% in both Recall@k and NDCG@k while retaining user privacy by achieving 99% dissimilarity.

ACKNOWLEDGMENTS

I can not express enough gratitude to my mentor, Dr. Stas Tiomkin, for his invaluable insights, continuous support, and extensive patience. Our discussions inspired me to work harder and achieve more everyday.

I would also like to thank Derek Lilienthal and Dr. Magdalini Eirinaki for the great privilege of collaborating with them and witnessing an idea transition from design, to research, to reality. Additionally, I am grateful to my peers, Tristan Shah and Volodymyr Makarenko, for our extensive discussions and assistance during this journey.

Finally, I would like to dedicate this work to my family. To my mother and father, Clara A. Mello and Paul A. Mello, for their unwavering support and extensive love, without which I would not have been able to complete this thesis. And to my brother, Isaac A. Mello, for inspiring and encouraging me to overcome each challenge I faced along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AWGN	Additive White Gaussian Noise
CFG	Classifier-Free Guidance
DDIM	Denoising Diffusion Implicit Models
DDPM	Denoising Diffusion Probabilistic Models
DIRAC	DIffusion-based Residual Augmentation Codec
DNN	Deep Neural Network
DPI	Data Processing Inequality
DPM	Diffusion Probabilistic Models
EBM	Energy-Based Models
ELBO	Evidence Lower Bound
FID	Fréchet Inception Distance
I-MMSE	I-Minimum Mean Square Error
IDDPM	Improved Denoising Diffusion Probabilistic Models
KDE	Kernel-Density Estimator
KL	Kullback-Leibler (Divergence)
LPIPS	Learned Perceptual Image Patch Similarity
MDP	Markov Decision Process
MI	Mutual Information
MINE	Mutual Information Neural Estimation
MMSE	Minimum Mean Square Error
MSE	Mean Squared Error
NLL	Negative Log-Likelihood
SDE	Stochastic Differential Equation
SNR	Signal-to-Noise Ratio
SOTA	State-of-the-Art
VAE	Variational Autoencoder
VLB	Variational Lower Bound

1 INTRODUCTION

In late 2015 DPMs were introduced in [1], ushering in a new class of EBMs. These EBMs take the form of deep, likelihood-based generative models parameterized with an arbitrary constant. They demonstrate the ability to synthesize high quality samples [2], but historically, probabilistic models have seen uncompromising trade-offs between tractability and flexibility. Tractable models allow for straightforward evaluation metrics and provide the ability to easily fit Gaussian data distributions. Simultaneously, tractable models tend to suffer from the inability to reconstruct deeply rich data. Whereas, flexible models are designed to fit the structure of arbitrary data distributions, but are often intractable. This has lead to approximations which attempt to minimize, but not eliminate, this trade-off [3].

GANs [4] arose as a popular solution to this trade-off. This technique pits two models, a generative model and a discriminative model, against each other. The generative model attempts to fit the training data distribution, while the discriminator model estimates the probability that a sample has come from the training data, as opposed to the generative model. This method has exhibited better sample quality than their likelihood-based counterparts such as VAEs [5]. VAEs are latent variable models which leverage a probabilistic encoder to feed a data distribution into a compressed latent Gaussian distribution, then are trained to decode the latent variable. Despite their success, the processes responsible for GANs have traditionally been unstable. This has ultimately required specific architectural designs to stabilize models during training [6].

A now viral class of deep generative models known as DDPMs have since dethroned GANs for their ability to synthesize high quality samples through non-equilibrium thermodynamics [7]. These generative models have captured the public's imagination for their capabilities of data transformations between any data modality and distribution. They have found use in large language models [8] and synthesizing data to improve final training

accuracy [9]. These results demonstrate only a small fraction of the extraordinary versatility of diffusion models that have resulted in disruptions across various industries and fields.

While diffusion models have demonstrated exceptional results in a wide range of tasks, the intricate mechanisms which drive these systems remain poorly understood. Denoising neural networks arose as a solution for tractability in thermodynamic problems, but are even less understood than diffusion. In this thesis, we explore the mechanisms driving diffusion from an information theoretic-perspective. We study the effects of different parameters on diffusion models and show that exact and class-based reconstructions of the input data are possible when information loss is kept low. We explore diffusion processes and DNNs through various connections including MMSE, MI, and SNR.

1.1 Information Theory

Information theory was established in 1948, following a seminal paper by Claude Shannon titled "A Mathematical Theory of Communication" [10]. Shannon's paper studies the minimally sufficient bits necessary for the lossless transmission of information over a noisy channel. A channel is considered noisy when random perturbations can find their way into the transmission medium and can corrupt the data being transmitted. It has since served as the basis for the quantification of information as a measure of entropy. Information theory defines a definite bound on the quantity of information that can be communicated through a channel between any two variables. Meaning that for any communication channel there is an upper limit on the channel capacity and traversing noisy channels can only reduce this theoretical limit.

Information theory has been used in numerous AI applications [11], [12], [13], and [14]. These approaches define a rising trend to study information as a means to understand a models internal dynamics. While a definitive structural framework to interpret neural networks has not yet been put forth, information theory is one avenue at our disposal to understand our model architectures. As it has offered a unique perspective into the information maximizing

processes inherent to neural architectures and the trendy forefront of interpretability research. As François Fleuret, head of the Machine Learning Group from the University of Geneva stated, "Information theory is maybe the only toolbox that sometimes gives you some certainties in this mess" [15].

1.1.1 Entropy

The entropy of a random variable $X \sim p(x)$ is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log\left(p(x)\right) \tag{1}$$

Entropy approximates the minimal number of bits necessary to describe X through the use of $\log(p(x))$. Bits may be defined as binary operators that trace the uncertainty of possible bit combinations of X.

We consider the case where the distributions are uniform between $[0,1]$. Here entropy follows a concave parabolic curve which reaches a peak at .5. This describes the certainty with which two random variables are equally likely to occur. This was shown through Gibbs' inequality [16] which demonstrated that the information entropy of a distribution P must be less than or equal to any other cross entropy distribution Q. Importantly, uniform distributions also contain a few interesting properties including their ability to maximize differential entropy and that their uniformity makes them robust to small additive noise perturbations. Maximizing differential entropy is thus equivalent to finding the maximum point of uncertainty within a probability density function. This will become a key component of the underlying idea in this thesis. We define the differential entropy of a continuous random variable X as:

$$h(X) = -\int_{-\infty}^{\infty} p(x) \log p(x) dx \tag{2}$$

These innate properties offer a means to measure the uncertainty of a probability density function. By measuring this uncertainty we can differentiate between desired and undesirable

signals propagated by the input data. In the next section we cover KL divergence, a tool to quantify this these differences.

1.1.2 Kullback-Leibler Divergence

KL divergence, also known as relative entropy, quantifies the information differences between any two probability distributions. It measures the expected log-likelihood between between two distributions defined by $p(x)$ and $q(x)$. This non-negative function has diverse theoretical and practical applications. The discrete case of KL divergence is denoted as:

$$D_{KL}(p|q) = \sum_{x \in X} p(x) \log \left(\frac{p(x)}{q(x)} \right) \tag{3}$$

When distributions $p(x)$ and $q(x)$ have low KL divergence, the values they define are considered similar. Conversely, when these values are large, these distributions are considered to be dissimilar. The relative entropy of these variables defines a quantified value of information divergence. For this reason it can be treated as a similarity measure between two arbitrary distributions.

$$D_{KL}(p|q) = \sum_{x \in X} p(x) \log \left(\frac{1}{q(x)} \right) - H(x) \tag{4}$$

KL divergence is non-symmetric $D_{KL}(p\|q) \neq D_{KL}(q\|p)$ and non-negative. D_{KL} is always positive because the difference between distributions are absolute. In continuous distributions, the summation may be replaced by an integral to operate over the range of a distribution.

$$D_{KL} = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) \tag{5}$$

Divergence of a distribution can be especially challenging to compute accurately. As a result, KL divergence is a rough estimator. Challenges arise in quantizing divergences because they require an arbitrarily fine number of measurements for precision. Additionally, similarity

in distributions $p(x)$ and $q(x)$ are defined by some sufficient statistic for measurement. These sufficient statistic are achieved when there is no loss in information between $p(x)$ and $q(x)$. This is a fundamental property of information theory known as MI. MI has opened up significant opportunities to explore expectation-maximization algorithms and serves as the fundamental component of this thesis.

1.1.3 Mutual Information

MI is defined as the shared entropy between random variables. It can be understood and used in many different forms including as a quantification of shared entropy, dependency measure, or saddle point for converging information. MI between two discrete variables X and Y are denoted as $I(X;Y)$. MI has a few notable properties such as its symmetry $I(X;Y) = I(Y;X)$, non-negativity $I(X;Y) \geq 0$, and that more data leads to more information $I(X_1;X_2;Z) \geq I(X_1;Z)$. This later property, DPI, is incredibly important and will be covered in the following section 1.1.4. For now, its worthy to mention that DPI is implied within MI under the following:

$$I(X;Z) = D_{KL}(P_{Z|X}||P_Z|P_X) \leq D(P_{Y|X}||P_Y|P_X) = I(X;Y) \tag{6}$$

The symmetric property of MI states that when X, Y are variables that are fully dependent, information known regarding X is information also known about Y. Conversely, when these random variables are fully independent, they share no information. This spectrum of MI follows a parabolic curve that spans between $[0, \infty]$. It is unbounded just as we see in entropy and is invariant to nonlinear transformations. MI may also be understood through the lens of uncertainty. As entropy of a random variable $H(X)$ becomes more certain, $H(Y)$ also becomes more certain. This can be seen through the conditional entropy defined by $H(X|Y)$.

$$I(X;Y) = H(X) - H(X|Y) \tag{7}$$

This uncertainty principle of MI provides a useful metric to analyze variables. Through various mathematical principles. Fig. 1 demonstrates the useful properties of MI in a geometric visualization.

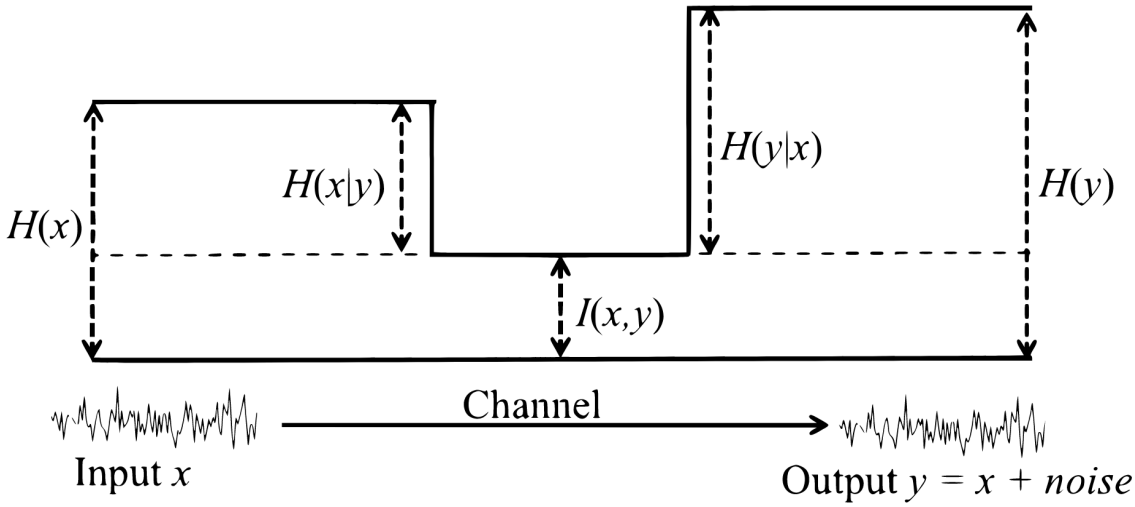

Fig. 1. This geometric depiction of MI, taken from [17], demonstrates the versatile properties of information between two random variables X and Y, as a measure of uncertainty defined by equation 7. From this visualization we can decompose information into a sum of its parts defined by individual and shared information. Figure from [17].

Traditionally, MI is seen to be constrained within some bound, however expanding to the continuous case provides significantly more refined information quantities. Consider $I(X;Y) \leq D(P_{Y|X}||Q_Y|P_X)$, here Q_Y and P_Y are uniquely equal as they are the minimal representations necessary to bound MI to its upper limits on Q_Y. When neither random variables are discrete it becomes necessary to use the respective marginal distributions. The marginals can be represented by the joint probability distribution of $p(x,y)$:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{8}$$

Interestingly, MI can also be defined through the KL divergence between two random variables [18] such that:

$$I(X;Y) = D_{KL}(p(x,y)||p(x)p(y)) = \mathbb{E}_{p(x,y)} \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{9}$$

Borrowing ideas from calculus, these discrete time equations can be thought of as subsections of a continuous time process. With this in mind we can define continuous functions to measure the transformation of information through diffusion processes:

$$I(X;Y) = \int_y \int_x p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy \tag{10}$$

So far, we have focused on instances containing two random variables. Expanding these notions to three or more random variables, such as (X,Y,Z), introduces additional complexity that can be adequately captured by Markov chains. Markov chains, also known as Markov kernels, have profound applications in statistical modeling and are the basis for stochastic simulation methods as they describe a sequence of events as states and probabilities.

1.1.4 Data Processing Inequality

DPI is a core principle to information theory and a reoccurring one in this work. When considering a Markov chain of three or more variables the joint distribution can be represented as $p(x,y,z) = p(x,y)p(z|y)$. Here the DPI becomes a fundamental property of MI. A Markov chain of three random variables $X \to Y \to Z$ describes the conditional distribution of Z to rely on Y, which also relies on X. Meaning that processing can not increase the information in a contained system, only maintain or reduce. The DPI can be seen in the equation below:

$$I(X;Y) \geq I(X;Z) \tag{11}$$

For all systems which process and transform information across a channel, the information extracted can only be less than or equal to the initial input information. An improved version

of the DPI exists known as Donsker-Varadhan which relates $D(P||Q)$ KL divergence to the supremum of a class of functions.

These fundamental challenges of information processing become more apparent as we consider signals sent through communication channels. These signals degrade over time due to the addition of unintended noise. Information sent over these noisy channels should thus be robust to noise perturbations, while remaining fully reconstructable at its endpoint. These channels often contain finite bandwidth resulting in the necessity to compress the information passing through the channel into some minimally sufficiency statistic capable of full reconstruction.

1.1.5 Minimally Sufficient Statistics

DPI can be seen as a special instance of minimally sufficient statistics where the inequality is determined to be equal. Consider the statistics of a data distribution $p(X)$ such that the sufficient information necessary to capture the exact data distribution is denoted as θ, or the maximum entropy. Here we can describe the minimal statistics through likelihood.

$$I(\theta; p(X)) = I(\theta; X) \tag{12}$$

Similar to Shannon's theory on the minimal amount of bits necessary for perfect reconstruction of a random variable sent through a noisy channel, [19] presented a methodology to find the most compressed representation of a random variable such that all information is necessary for a probabilistic reconstruction on p_θ. It becomes non-trivial to prove that there exists statistics θ which are minimally sufficient for an accurate reconstruction. Although, this process does require that the probability density can be factorized through Eq. 12. This is known as Fischer's factorization theorem and is used to guarantee the optimal statistic.

1.1.6 Channels: Capacity and Additive White Gaussian Noise

Shannon's work developed the foundations for information theory by defining the necessary bits for transmission over a noisy channel. Shannon's key finding was that a channel's capacity is linked to the maximum MI between the input and output. Informally, a channel is any medium capable of transporting information. Formally, a channel is more closely defined as a Markov kernel. Channel capacity refers theoretical transmission rate of information through a channel. Often channels are structured similarly to VAEs. Input is encoded, passed through a channel, then decoded. These encoders and decoders compress and decompress the information to transfer along a given channel. In instances of perfect information transfer, the decoder may be considered deterministic. Unfortunately, channels are often imperfect. Random noise can adversely affect the signal quality and potentially damage reconstruction, making the introduction of unintended noise a significant obstacle. Conversely, we can use this random noise to calculate the channel capacity by using conditional probability and the marginal distribution $p(x)$ to find the joint distribution of two random variables X and Y.

$$C = \sup_{p_x(x)} I(X;Y) \tag{13}$$

This equation defines a significant result for information theory, bounding channel capacity to an upper limit on information throughput defined by MI. Although these bounds are difficult to achieve due to noise, modern error correcting codes have pushed channel capacity just shy of this theoretical limit. To achieve this, researchers have used the statistical properties of noise and information to reduce the effects of unintended noise [20]. Rather than attempting to decode the input containing stochastic noise, researchers add white Gaussian noise to permute the stochastic noise into Gaussian noise. This AWGN channel is defined by the "Gaussian capacity":

$$I(X;Y) = \frac{1}{2}\log(1+\frac{\sigma_X^2}{\sigma_N^2}) \tag{14}$$

Here X and N are independent Gaussian's, Y is the total AWGN channel, and $\frac{\sigma_X^2}{\sigma_N^2}$ is the SNR. While more thorough descriptions of channel capacity exist, these descriptions are sufficient for this work.

A few interesting notions arise from these brief ideas. Primarily, additive noise contains saddle points that minimize SNR. In order to make predictions based on the subsequent information provided by a random variable X, the variance must remain normal and independent of other random variables. Another is that the Gaussian capacity is invariant under orthogonal transformations. This is because averaging over many rotations of variable X can only increase the captured MI [21].

1.2 Control Theory

In continuous dynamic settings it is often critical, and difficult, to manage the potential variation a system may encounter. Control theory attempts to stabilize these continuous linear and nonlinear systems. In nearly all sensor systems, there is an expectation of compounding uncertainty due to environmental noise or sensor inaccuracies. Control theory rectifies this uncertainty by approximating future state dynamics. This is only possible when all necessary environment variables are known. From this understanding control theory is the focus of reducing variance in non-trivial chaotic systems.

It is commonly understood that dynamic systems are determined by the state $x(t)$, its input $u(t)$, and its output $y(t)$ at any given time t. We define these dynamic systems by the following equations where A, B, C, D are matrices that describe the systems dynamic properties over time.

$$\begin{aligned} x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{15}$$

Control theory is a field which is defined by its effectiveness to correct observations to achieve a desired output state. Similar to empowerment, this notion depends on maximizing the MI between the input and output states to guide a dynamic system. Practically, this involves applying a feedback loop to adjust the noisy input data with error correcting predictions to adjust the input given the error between the desired output and the expected output. These error minimization techniques are shared with neural networks as they are defined by minimizing loss between output and target variables. A well known example of control theory, the Kalman filter, exists in the following section 1.2.1.

1.2.1 Kalman Filter

In this section, $A \in \mathbb{R}^{m \times n}$ is a stable state of sensor inputs C while W represents the covariance of the noise. $x_t \in \mathbb{R}^n$ represents the current state and Σ_t represents the covariance of the error estimate. $y \in \mathbb{R}^m$ is the measurement and V is the sensor noise.

The ability to stabilize systems depends entirely on the problem and action space. A linear system may have a trivial solution, but nonlinear systems often necessitate non-trivial solutions. Compounding noise introduced by external factors will eventually lead to large variations over time. The Kalman filter rectifies these non-trivial issues by applying local linearity through covariance estimation. The Kalman filter consists of a two step feedback loop. In step one, a prediction is made to correct the imperfect observation data of the prior state using the current state by applying updates to its covariance matrix. The prediction step is denoted below:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + \Sigma_{t|t-1} C^T (C\Sigma_{t|t-1}) C^T + V)^{-1} (y_t - C\hat{x}_{t|t-1}) \tag{16}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1} C^T (C\Sigma_{t|t-1}) C^T + V)^{-1} C\Sigma_{t|t-1} \tag{17}$$

Notice that the next state is entirely dependent on the previous state. This is known as the Markov property, which asserts that when all necessary state information is encapsulated into

the present state, a prediction can be made about future states to an arbitrary degree of accuracy. This notion is fundamental in Markovian systems and entails that future states $X(t_{n+1})$ are independent of past states $X(t_{n-1})$ given the present state Xt_n. This dictates an intriguing relationship between prior and future state dynamics where the immediate future is predictable.

Step two, known as the measurement update, incorporates the updated prediction with the current observation state and merges them to correct any noise introduced to the input. The update step can be seen below:

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t} \tag{18}$$

$$\Sigma_{t+1|t} = A\Sigma_{t|t}A^T + W \tag{19}$$

$A\Sigma_{t|t}A^T$ defines the signal's covariance. These steps can be further merged and condensed utilizing covariance through the process below:

$$\Sigma_{t+1|t} = A\Sigma_{t|t-1}A^T + W - A\Sigma_{t|t-1}C^T(C\Sigma_{t|t-1}C^T + V)^{-1}C\Sigma_{t|t-1}A^T \tag{20}$$

While counter intuitive, this recursive equation is able to estimate the covariance error before the next observed data is introduced. This improves prediction accuracy in stable and unstable environments. The Kalman filter is the best possible linear estimator due to the nature of independent Gaussian random process remaining fundamentally linear. Notably, Takens' Theorem [22] also demonstrates a similar capacity to predict future state dynamics, but does so through higher dimensions. These concepts have become foundational in the study of modern dynamical systems. This mixture of variance estimation and observation data provides a way to correct accumulating errors when the covariance is known.

For thoroughness, Taken's theorem states that even though higher dimensions may be "hidden", they remain vastly interconnected with "visible" dimensions. These hidden dimensions exert influence on its future trajectory. Through Takens' theorem, we understand that for any continuous d-dimensional state vector, the state dynamics can be considered deterministic. This deterministic relationships allows us to infer the hidden state dynamics from the visible.

1.2.2 Minimum Mean Square Estimation

Similar to KL divergence in information theory, MMSE provides a similarity score between any two random variables. For any two random Gaussian vectors x and y, they have a conditional entropy of $\mathbb{E}_{x|y}$ and follow the form: $\mathcal{N}(\bar{x}, \Sigma_x)$. Conditional entropy emphasizes information $I(X;Y)$ is also a measurement of the uncertainty of X once we observe Y. The uncertainty of x in this instance describes a prior distribution which allows us to use the following vector representation:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_y \end{bmatrix} \right) \tag{21}$$

Through this understanding we can rewrite ϕ_{MMSE} as a similarity function of $\bar{x} - x$:

$$\text{MMSE} = \bar{x} - x \sim \mathcal{N}(0, \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T) \tag{22}$$

MMSE is a fundamental measure for many disciplines which utilize statistics. It provides an elegant way to iteratively reduce differences between the predicted and true data. This powerful comparison metric provides component to systematically find the best model estimator given a loss function.

1.3 Representation Learning

Artificial intelligence covers a broad selection of diverse fields including machine learning and reinforcement learning. The core of these fields are defined by learning abstract

representations of data in a latent space in order to compact and extract relevant information. These learned representations are imperative to the success of all models and their subsequent quality. "...[T]his is because different representations can entangle and hide more or less the different explanatory factors of variation behind the data" [23].

Representation learning serves to reduce computational costs and reliance on manual feature engineering for standard machine learning model tasks. These models learn to capture the underlying data distribution and structure of the data. This is particularly useful as the dimensionality of the data inputs increase. These representations are highly compressed input data, but exhibit feature rich qualities. Representation learning also utilizes disentanglement. Disentanglement is a method used to capture independent factors of data variation by separating them to promote generalization and enhance model interoperability. "The key idea behind the unsupervised learning of disentangled representations is that real-world data is generated by a few explanatory factors of variation which can be recovered by unsupervised learning algorithms" [24]. Models built with latent properties inherently learn these data representations as they represent the theoretical principal components which define the essence of the data distribution.

1.3.1 Latent Variables Models

Latent variable models are a class of statistical models that attempt to map the underlying distribution by introducing unobserved latent variables. These latent models can interpolate between discrete and continuous spaces due to the nature of latent sampling from prior distributions. Similar to likelihood models, they generate data from the conditional distribution $p(x|z)$. The goal is often to perform inference on the posterior distribution $p(z|x)$ to gain probabilistic insights into the latent variables. VAEs demonstrate the structure of latent variables models through the use of a compression and decompression schemes to map input data x to the latent variable z. In the context of multivariate discrete environments, one can

leverage the marginal and conditional distributions in their respective forms shown below and reinterpret these principles for continuous settings.

$$p(x_i|y_i) = \int p(x_i|z_i, y_i)p(z_i|y_i)dz_i \tag{23}$$

$$p(z_i|y_i, x_i) = \frac{p(x_i|z_i, y_i)p(z_i|y_i)}{p(x_i|y_i)} \tag{24}$$

From 24, we can infer the latent variable. From the conditional distribution, we can use the ELBO 1.3.2 and KL divergence to measure similarity through the use of the marginal in 23. These models serve as useful tools for understanding hidden patterns within data distributions. They efficiently capture the estimation of structural relationships by leveraging probability distributions to produce models which map features to latent representations. These are defined by expectation-maximizing algorithms that utilize Bayesian inferencing techniques for probabilistic modeling.

1.3.2 Evidence Lower Bound

In generative modeling, we can define latent variables and the generated data as being modeled by a joint distribution $p(x,z)$. Using likelihood-based methods, a model learns to maximize the likelihood $p(x)$ given x. Calculating this can be done in two ways, both of which are computationally intensive. One must either integrate z out, which may be intractable, or have access to the ground truth latent encoder on $p(z|x)$. For our purposes we focus on utilizing the chain rule of probability to derive an ELBO, or VLB, of the observed data through the log-likelihood. Here $q_\theta(z|x)$ defines a variational distribution.

$$\mathbb{E}_{q_\theta(z|x)}[\log \frac{p(x,z)}{q_\theta(z|x)}] \tag{25}$$

The ELBO objective can be optimized such that it becomes equivalent to the evidence $\log p(x)$. Through this $q_\theta(z|x)$, seeks to learn a real approximated posterior of $p(z|x)$. Using our learned distribution on $q_\theta(z|x)$ and a real distribution on $p(z|x)$ we can quantify their differences through KL Divergence from 1.1.2. Using this notion we understand ELBO as being bounded and a non-negative term that can never imply more or less information than truly exists. Unfortunately, it is often the case we do not have access to $p(z|x)$, making KL divergence intractable. Instead, the maximization of the ELBO can be used as an alternative for the minimization of KL Divergence. This ensures that our approximation remains close to the true distribution. Furthermore, the ELBO strikes a balance between the fidelity and complexity of the predicted distribution. This balance is crucial in preventing overfitting and ensuring the generalizability of the model. As such, ELBO serves as a computationally convenient and fundamental cornerstone in the framework of variational inferencing.

1.4 Diffusion Models

DPMs are a class of deep generative models that efficiently draw samples from a distribution $p(x)$. They consist of a class of latent variables models which merge probabilistic approaches with sampling techniques. Fundamentally, these models are structured to learn to denoise those latents through a DNN training process. These models have proven to be useful in a variety of tasks such as, super-resolution [25], compression [26], imitating human behavior [27], trajectory planning [28] and many more.

DPMs can be considered through the lens of SDEs [29]. Considering DPMs through the dynamics of SDEs can provide significant insight into the inner mechanisms of the diffusion process by formalizing their internal dynamics. Wielding the tools of SDE dynamics provides the ability to demonstrate the transformation of x through T timesteps. This SDE is also constrained by an upper and lower bound on information. We can consider this potential variation in state possibilities as a form of information density which can be tightened by

restricting the upper and lower bounds. By tightening these bounds we reduce the necessary complexity of the search space a DNN is subject to during training.

DPMs are the central point of this work and will be expanded on in 2.1

1.4.1 Variational Bounds

DPMs are data transformers with large variance in output dynamics. In this way, there is an upper and lower bound for each input-output pair, which is determined by the initial data space and parameters. Moreover, these bounds define all potential variational outputs that are possible in the trajectory space as we traverse our Markov chain of T steps. We may consider this variance as the information density defined by the VLB of the marginal likelihood.

$$-\log p(x) \leq -\text{VLB}(x) = D_{KL}\left(q(z_1|x) \parallel p(z_1)\right) + \mathbb{E}_{q(z_0|x)}\left[-\log p(x|z_0)\right] + \mathscr{L}_T(x) \tag{26}$$

In general, variation models are intractable without having prior knowledge about the distribution of x. However, one can approximate this through normalization and differential entropy. This can be seen as a case where an approximation of an otherwise intractable problem of predicting the upper and lower bounds can by simplified by normalization of the data structure and quantification of the information spread 1.1.1 without needing explicit knowledge of the prior distribution. The VLB is a universal bound for all diffusion models as it demonstrates information densities. Tightening this bound is akin to increasing the relevant information a diffusion has access too at any given time, which allows it to make more informed trajectory decisions.

The upper bound can be understood as a the theoretical information limit which is used to prevent unnecessary information from being used during classification. The lower bound ensures sufficient information is extracted from the input data. By bounding these information densities through variational objectives we can catalyze the training of tractable models.

1.5 Deep Neural Networks

DNNs are a subsection of machine learning where the model computationally reconstructs an approximation of the neural connections in a human brain. Their fundamental structure is a deep chain of neurons which individually take in inputs from the previous layers, multiply their outputs by the current neurons weight, then, depending on an activation function, will propagate the signal if it exceeds the activation function's threshold. Through this iterative process the data reaches the final layer then back-propagation updates all neuron weights according the accuracy determined by the objective function. Regardless of generalization, a DNN will learn to predict unseen data within the training distribution. This iterative process gradually converges to some near optimal representation.

These neuron chains can be thought of as a series of complex Markov chains which iteratively extract features in a manner that seeks to efficiently encapsulate all necessary information in a highly compressed format [30]. DNNs take high dimensional data x, such as images, and attempt to transcribe their sparse dimensional features into a dense lower dimensional output y. This sparse data transformation describe a data input of x likely has very low entropy shared with the output y. This ability to identify relationships from sparse instances has been attributed to their sequential data processing capabilities. As feature representations are fed into sequentially deeper hidden layers, each layer learns to extract, abstract, and represent the features in an organized and distributed manner.

These neuron chains are linked together to create a neural network. A DNN consists of an input layer, followed by a series of n hidden layers, and an output layer. The input and output layer often have neuron counts defined by the respective data, whereas the hidden layers have an arbitrary numbers of neurons that are dependant on a multitude of factors. As data flows through these layers, the neural network separates the data and geometrically clusters similar data points together. As we increase the number of hidden layers we find that DNNs perform better on classification tasks. The process underpinning the feature extraction capabilities of

DNNs is not well understood, but recent work [31] has shown that DNNs rely on affine splines which create decision boundaries on the input space. These decision boundaries become increasingly concentrated as training continues, leaving large regions of uninterrupted label classifications.

1.6 The Information Bottleneck Principal

The information bottleneck method was first proposed in [32] by Naftali Tishby as a framework for understanding the internal processes of deep neural networks. Utilizing information theory to contextualize the foundational aspects of deep learning has emerged as a prominent research direction for model interpretability. It focuses on an indirect calculation of model capabilities by focusing on deep learning as an information extractor through MI between input and output variables.

Neural networks are Markov chains which are improved through back propagation. Tishby found that as information is propagated through each Markov chain, the sum of information continues to decrease. This coincides directly with the decreasing information described by DPI 11 and demonstrates that signals propogated through these networks continually compress information. Moreover, the individual structure of each layer, including fully connected, convolutional, and any other layer type, has an effect on the networks capabilities for compression and information preservation. For example, convolutions exploit the spatial invariance between weights for better compression while recurrent layers capture temporal dependencies. Both of these layer types preserve information in different methods.

Tishby provides evidence of the transformation of information through layers and training epochs. They found that as neural networks are trained the information processing of each layer would first drift towards maximizing information, then in a second phase the model would learn to compress the extracted information. The drift phase is characterized by the early stages of training where a network rapidly learns to extract the relevant, information rich, features from the input data. These internal representations are gradually refined towards

maximizing the MI between the input and output variables. Refining these representations by maximizing MI asserts that there are MSS which can express these shared representations. The diffusion phase is significantly longer and starts immediately after the network extracts most, if not all, relevant interdependent information. This phase consists of learning dense compressions of the input-output pairs. Traditionally, this was considered time for the models to ”random walk” through the hyperplane to find better generalizations and global minimums. Tishby's work provides and answer to this intuition. His results point towards information becoming tightly compressed leading to improved generalization. Together these processes define the findings of Tishby's work demonstrating the internal dynamics of DNNs as a process of information maximizing $I(T;Y)$ measures while minimizing $I(T;X)$ subject to a Lagrange multiplier to balance these two competing processes. Fundamentally, the design of neural networks allow them to distinguish and separate the relevant information from the irrelevant noise.

The information bottleneck method demonstrates that neural networks, given enough time, will learn to extract significant relevant information from the input to identify the target output. It follows a natural information bound guided by DPI 11, where the output information can not exceed the input information. Fundamentally, these bounds describe the theoretical optimal limit for information extraction and, rather surprisingly, neural networks appear to follow this bound exactly. These findings have many applications in interpretability, by defining better model architectures and improving our understanding of model training.

1.7 Recommender Systems

Recommender systems have become an integral part of Web services. Recommender systems are algorithms which personalize content and serve it to users such that it filters content to maximize user engagement. This content can include things such as advertisements for products or for entertainment. Recommender systems serve many purposes including keeping users on a Web platform, but are primarily seeking to maximize generated profits by

serving content a user may not have otherwise found. As the Web ecosystem continues to expand relevant content becomes increasingly difficult for users to find. Moreover, large corporations rely on on these systems in order to optimize the generation of revenue, increase users engagement, and improve users experience. In some instances, the recommender algorithms used are directly tied to potential profits and thus improvements to these systems are of utmost importance.

Traditional approaches to recommender systems included searches, but have been replaced by machine learning algorithms which learn to serve content to users based on a variety of known factors regarding the user. These factors are aggregated in the form of data and can include information regarding a users past behavior, demographic data, and much more. Evidently, the collection and aggregation of user data has become a common occurrence throughout Web services to improve recommender model accuracy. These prior concepts are known as knowledge based systems. However, other variations exists known as collaborative filtering and content-based filtering. Collaborative filtering assumes that similar users share similar tastes while content-based filtering relies on the past behaviour of a user to infer their future behavior. These approaches all tend to suffer from data sparsity, meaning that there is a lack of user data. A lack of user data may be defined either by informational sparsity on a per-user basis or as an aggregate of users.

As user and item diversity increase, it becomes increasingly difficult to model user-item preferences due to the increasing aspects of data sparsity and potential biases which may arise. In this thesis we explore the utilization of generative modeling techniques in order to generate novel data to help augment or fully replace preexisting data sets used to train recommender models.

1.8 Score Functions

Selecting the proper objective function for machine learning algorithms is an important task. Objective functions fundamentally determine the desired optimization goal by allow the

model to minimize or maximize what is often the difference between generated predictions and observed data. Score functions are one form of objective functions. They evaluate and assign a variable with a numerical value which can be used as a metric for optimization. This numerical value is determined by the gradient of a log probability density function which defines a true gradient plane of the training data. The intuition is that by optimizing the score, or steepest ascent on the gradient plane, we can generate an unbiased value that guide a model towards desirable loss by minimizing the expected squared error. To iteratively improve a model score functions minimize loss by updating the model parameters given the models gradient plane leading to this unbiased representation. Importantly, this inevitably requires the ground truth data in order to train a score-based model properly.

Various challenges arise when utilizing a score function over an objective function. As previously mentioned, one must have the ground truth data. Often score functions are not differentiable or smooth. This makes gradient-based optimization techniques challenging to apply to score functions. Similarly, score functions may not produce desirable results since one can not necessarily control the gradient plane and the trajectory of optimization. Score functions also require significant hyperparameter tuning to control the complex space of the gradient plane. Despite this various score functions have been proposed including denoising score matching.

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p_{\text{data}}(x)}\left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log q_\sigma(\tilde{x}|x)\|^2\right], \tag{27}$$

In this instance of denoising score matching, the function described considers a joint density for $q_\sigma(\tilde{x}|x)$. The idea of desnoising score matching is to add various perturbations to the data itself so that a robust generalization of the non-differentiable and non-smooth gradient plane can be learned. Score matching can fail when data is sparse or ground truth data is unavailable. For now, it is only important to consider that score functions, like score matching,

utilize the probability density function of the training data and evaluate the likelihood of observing the data given probability distribution as shown below.

$$\mathbb{E}_{p(x)}[||s_\theta - \nabla \log p(x)||^2] \tag{28}$$

2 BACKGROUND

2.1 Deep Unsupervised Learning Using Non-equilibrium Thermodynamics

DPMs are a class of energy based models first introduced by [1]. Their approach was inspired by statistical physics through a forward process to destroy the structure of a data distribution, then utilizing a multi-layered perceptron a model learns the reverse process. While the approach to corrupt an arbitrary distribution towards another is not new [33], Sohl-Dickstein leverages Langevin dynamics to show that any forward process must have an approximate reverse process. This simple probabilistic model design enables tractability and flexibility with exact sampling processes.

2.1.1 Forward Process

The forward process is a Markov chain that algorithmically corrupts training data distribution $q(X)$ towards a Gaussian distribution, where each data point is defined by $q(x)$. This perturbation of x provides expressive decoding. Originally, the algorithmic corruption processes is a deterministic process characterized by a linearlly increasing distortion rate of β, which defines the schedule for added noise, over T timesteps. The longer the timesteps, the smaller the β value can be. As β becomes smaller, only a single sample from $q(x_t|x_0)$ is necessary to corrupt the data for training.

Here π refers to the final trajectory taken starting from the initial input and ending in Gaussian noise. Notably, although the forward process is algorithmic, the bit-flipping probabilities for data is determined for a binomial distribution.

$$q\left(x_t|x_{t-1}\right) = T_\pi\left(x_t|x_{t-1};\beta_t\right) \tag{29}$$

Since this algorithmic corruption is deterministic, many ideas from statistical mechanics can be applied to analyze and improve the process. The forward diffusion trajectory created through Eq. 29 is then defined as:

$$q\left(x_{0:T}\right) = q\left(x_0\right) \prod_{t=1}^{T} q\left(x_t | x_{t-1}\right) \tag{30}$$

The authors take some liberties with setting the initial conditions as an identity covariance matrix to define the data with a variance = 1. This form encapsulates the information of the data into a vector space that is useful for corruption and reconstruction.

2.1.2 Reverse Process

For every process, there must exist an approximate inverse process [34]. The reverse process, $p(x)$, for diffusion models employs a neural network to learn to denoise the corrupted data from the forward process. Effectively the neural network is tasked with learning to undo the noise added in the forward process through stochastic sampling.

$$p\left(x_{0:T}\right) = p\left(x_T\right) \prod_{t=1}^{T} p\left(x_{t-1} | x_t\right) \tag{31}$$

Eq. 31 is composed of the corrupted data from a forward pass of T timesteps. This data is approximately Gaussian making the process of learning to denoise simpler and encourages the generation of synthetic data which is perceptually similar to the initial training data distribution. During this learning, only the mean $\mu_\theta(x_t, t)$ and covariance $\sigma_\theta(x_t, t)$ is needed for an approximate estimation of the Gaussian distribution. The papers authors recognize that direct probability modeling fails to provide a tractable model. As a result a solution requires the use of derivative evaluation metrics in the form of location in μ_θ and information densities σ_θ. Using Bayes theorem, we can derive a model capable of reconstruction by relative probabilities averaged over T timesteps in the forward process.

$$p(x_0) = \int dx_{(1:T)} q\left(x_{(1:T)} | x_0\right) \cdot p(x_T) \prod_{t=1}^{T} \frac{p(x_{t-1} | x_t)}{q(x_t | x_{t-1})} \tag{32}$$

This alternative heuristic presents a novel and important breakthrough in modeling and generating complex distributions. Additionally, these algorithms only incur the computational cost associated with running the functions for T timesteps, giving them strong scaling capabilities.

2.1.3 Training

DPM can be trained by maximization of the log-likelihood. DPM relies on bounding the traversable search space of data trajectories by some objective in order to produce higher quality reconstructions. The authors demonstrate a theoretical upper and lower bound. The lower bound of the model log-likelihood is directly related to Jensen's inequality. Jensen's inequality refers to the fact that convex functions, like that of logarithms, the expectation of the function applied to a random variable is greater than or equal to the function applied to the expectation of the random variable. This allows for deriving the lower bounds L for the expected log-likelihood of the data. Through tightening these bounds, one can optimize these models with their diffusion trajectories; subsequently being defined by their MSE differences along the KL divergence over T timesteps:

$$\begin{aligned} & L \geq K \\ & K = -\sum_{t=2}^{T} \int dx_0 dx_t \, q(x_0, x_t) \cdot D_{\mathrm{KL}} \left(q(x_{t-1}|x_t, x_0) \,\middle\|\, p(x_{t-1}|x_t) \right) \\ & \qquad + H_q(X_T|X_0) - H_q(X_1|X_0) - H_p(X_T) \end{aligned} \tag{33}$$

As described in [1], "The derivation of this bound parallels the derivation of the log-likelihood bound in variational Bayesian methods". The lower bound approximates the entropic details of the data distribution as derived by the KL divergence. These bounds culminate in controlling the DNN's ability to learn the inverse forward process to an arbitrary accuracy as seen in the principle of $L \geq K$. Ultimately, these probability estimators are "...reduced to the task of performing regression on the functions..." [1].

2.1.4 Neural Network Architecture

In DPM, the neural architecture, as shown in Fig. 2, consists of a convolutional neural network which has data points defined by their temporally dependant prediction mean $\mu_i = (x_i - z_i^{\mu})(1 - \Sigma_{ii}) + z_i^{\mu}$ and covariance $\Sigma_{ii} = \sigma(z_i^{\Sigma} + \sigma^{-1}(\beta_t))$ for each pixel i. The act of generating a vector representation for each pixel is computationally intensive and difficult. The authors thus turn to multi-scale convolution down-sampling the data then up-sampling the data to retain compute resources, while also managing long term dependencies incurred by the system dynamics. Both the temporal and convolutional dependencies allow the model to learn denoising. Temporal information helps the model learn the series of diffusion denoising by providing noisy data with its associated timesteps. Similarly, when handling image data, convolutions can help model the spatial aspects of the data.

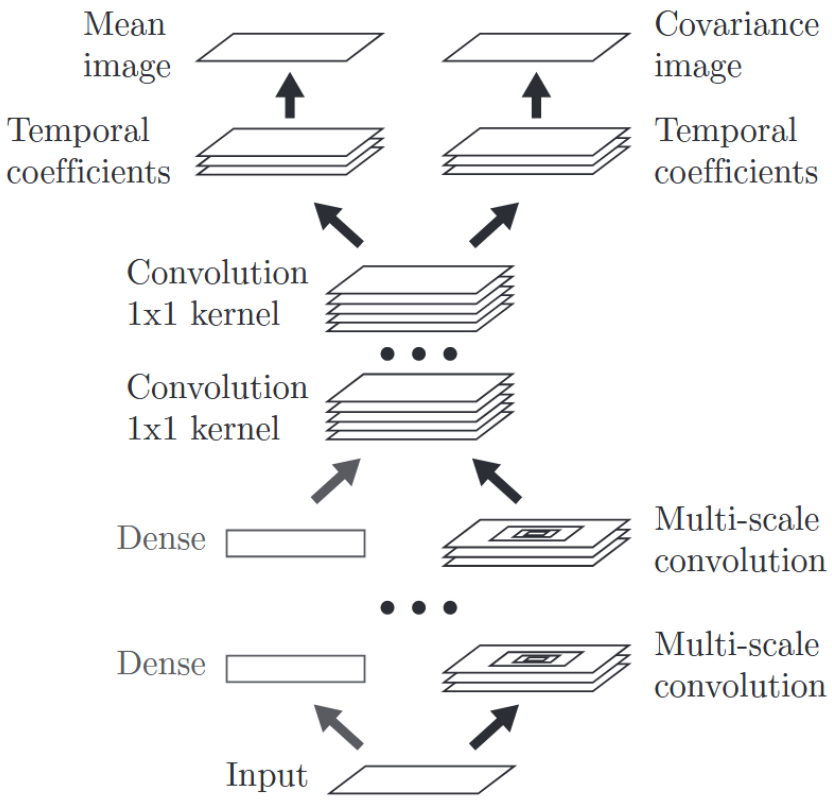

Fig. 2. This neural network architecture is defined in DPM [1].

2.2 Denoising Diffusion Probabilistic Models

Diffusion models have rapidly risen in popularity thanks to a series of papers culminating in DDPMs [7]. Diffusion modeling has been shown to surpass SOTA results from GANs on image synthesis [35], but has significant drawbacks including its sampling efficiency and computational costs. DDPMs are a tighter optimization of DPM. Like DPM, DDPMs consists of an iterative training and sampling process defined by its ability to algorithmically add and probabilistically remove noise.

The architecture of diffusion models follow a simple recipe defined by the interconnected nature of statistical mechanisms. The six main components responsible for their success are the denoising neural network ε_θ, the forward process $q(x_{1:T}|x_0)$, the reverse process $p(x_T)$, the noise scheduler β, the timesteps T, and the objective function. The denoising neural network seeks to minimize the difference between the noise and expected reconstruction of the data. The forward process gradually corrupts the input data towards a Gaussian normal distribution through an algorithmic scheduling process defined by the noise scheduler. The reverse process then utilizes a denoising neural network to learn how to denoise the latent variable x_t from the forward process into a reconstruction. Finally, the objective function effectively minimizes the difference between the latents in the reverse processes and the expected noise at a given timestep. DPMs abstractly learn to synthesize a reconstruction of the input data from a model tasked with learning the inverse trajectory of the forward process, by focusing on denoising the corrupted latent, rather than generating a novel one.

2.2.1 Forward Process

Fig. 3 defines the forward process where noise is incrementally added to corrupt input data. Starting from input data x_0, we gradually add noise such that x_t has a quantifiable change in noise from x_{t-1}. Once reaching x_T, the distribution of pixels across the image can be thought of as approximately Gaussian defined by $x_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$.

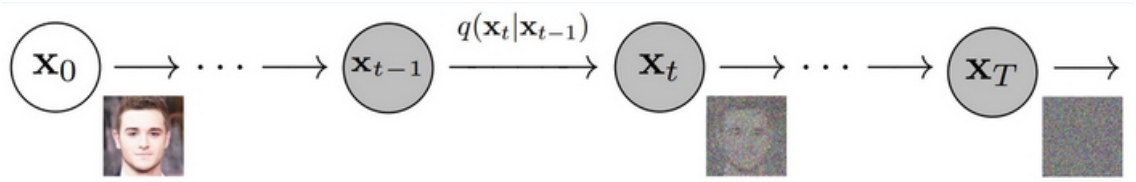

Fig. 3. The forward process for denoising diffusion probabilistic. Over a series of discrete timesteps, a predetermined noise quantity will be added to the distribution. This will teach a denoising neural network to reduce the errors incurred in modeling complex distributions. Figure from [7].

The forward process is mathematically defined through a closed form nearly identical to the scheme shown in Eq. 30, where we can define the posterior probability from input data x_0 to noise x_T in a tractable manner.

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{34}$$

Subsequently, we can define the diffusion process by the following terms. The forward process determines the next state of the corrupted image by T and β. Following a normal distribution $\mathcal{N}$ the data point x_t is defined by the location of noise at a timestep μ_t and the information density of Σ_t.

$$q(x_t|x_{t-1}) := \mathcal{N}\left(x_t; \mu_t = \sqrt{1-\beta_t}x_{t-1}, \Sigma_t = \beta_t \mathbf{I}\right) \tag{35}$$

Originally, DDPMs assumed a Gaussian distribution in order to apply methods of variance control through the scheduling process. High dimensional data, such as training inputs, can be understood as complex probability distribution. As we traverse the forward process, we transform the complex probability distributions which define the initial input towards a significantly more trivial Gaussian distribution. This approximate Gaussian contains imperceptible differences from a standard Gaussian distribution which contains information regarding the initial input. These differences are essentially a trace of compressed information

relating the final Gaussian distribution to the original input. In other words, as the latent is transformed the information is also transforming as a function of SNR.

2.2.2 Reverse Process

Fig. 4 demonstrates the regenerative theory of the reverse process. Information regarding the initial distribution is hidden within the imperceptible bits of the corrupted data. The reverse process learns to extract those hidden variables and propagate their signal in a tractable way towards a novel reconstruction.

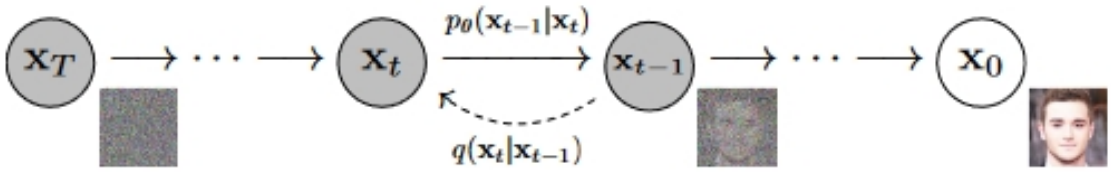

Fig. 4. The reverse process for DDPMs. The goal of the reverse process is to learn to reverse the forward diffusion process. This effectively allows a model to generate new synthetic data which resembles the training data. Figure from [7].

The Markov chain of the reverse process also parameterizes a neural network with the ability to learn to synthesize accurate image reconstructions such that $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$.

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{36}$$

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t)) \tag{37}$$

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon(x_t,t) \right) \tag{38}$$

DDPMs utilizes the models predicted mean μ_θ and covariance Σ_θ of the data distribution to predict the denoising process. In 38, μ_θ must predict $\frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon \right)$ which is a parameterization defined on the same principles of the forward process. β_t (also known as

$1-\alpha_t$) and $\Sigma_\theta(x_t,t)$ are subsequently constrained to being time dependent constants. This decision leads the authors to focus on μ_θ which, through reparameterization, simplifies their implementation to predict a "learned gradient of the data density" [7].

In subsequent studies, researchers have found μ_θ to be overwhelmingly influential in the reconstruction capabilities of the model. The exceptional influence of μ_θ asserts that knowing the mean is far more valuable than knowing Σ_θ.

2.2.3 Training

DDPM follows a series of steps beginning with input data x_0 and defining $0 \leq t \leq T$ timesteps. The uniform selection of t timesteps provides the reverse process with a robust training space to learn from. However, during training it is only necessary to generate a single data point containing the expected noise of a random t timestep. Despite this random selection, by randomly selecting $t \in T$ we are still able to denoise over the entire T. Fig. 5 demonstrates this algorithmic training process.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1,\ldots,T\})$
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

Fig. 5. DDPM training algorithm. Figure from [7].

2.2.4 Sampling

The sampling process, as shown in Fig. 6, is key to extract relevant information from the corrupted input by teaching a DNN to learn the process of denoising through iterative sampling. The sampling process amounts to the iterative denoising of x_{t-1} towards a

reconstruction of the data given the timestep. $x_T \sim N(\mathbf{0}, \mathbf{I})$ defines the corrupted Gaussian input. z defines the Gaussian latent variables from $x_{T-1}, ..., x_1$.

Algorithm 2 Sampling

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

Fig. 6. DDPM sampling algorithm. Figure from [7].

Here x_{t-1} is defined by a few key components, comprised of $\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}$ which is the scaled noise adjustments for the reverse process. Similarly, $1/\sqrt{a_t}$ defines the scale of the noise. Both of these help to control the SNR during sampling. Given the current timestep x_t and the models predicted noise $\varepsilon_\theta(x_t, t)$, we can iteratively denoise towards a reconstruction on $\hat{x}_0$.

2.2.5 Negative Log-Likelihood

Recently, the consensus has shifted to redefine diffusion as a class of latent variables models where the latent variables are exactly the same dimensions as the input data [7]. Diffusion models can capture temporal aspects of the data as they are designed to apply a series of small perturbations to the input without a reduction in representation. As a form of latent variables models, similar to VAEs, researchers impose techniques traditionally used in VAEs in DPMs. Particularly, they introduce the NLL to assess the differences between the noise distribution and the target data distribution 39.

$$L := \mathbb{E}[-\log p_\theta(x_0)] + \mathbb{E}_q \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] = \mathbb{E}_q \left[-\log p(x_T) - \sum_{t \geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] \tag{39}$$

This equation converts KL divergence into an objective to be minimized. Minimizing the KL divergence maintains the models ability to generate highly varied samples based on the training distribution.

2.2.6 Lower Bounds

A variational perspective on DDPM [36] focuses on measuring the VLB of the information curve. DDPM presents an approximation of this using a loss function defined as L and defines of L_{t-1} and L_T respectively, as the distributions between two Gaussian's. The stochastic gradient process can be coupled with sampling from Langevin dynamics, due to the noise incurred by β_t over a sequence of timesteps. The scheduled addition of variational noise is a critical component of DDPMs as it ensures a model can learn the theoretical reverse process defined by a natural denoising function. This iterative introduction of noise presents an opportunity to measure information through the following loss terms.

$$\begin{aligned} L_{VLB} &:= L_0 + L_1 + ... + L_{T-1} + L_T \\ L_0 &:= -\log p_\theta(x_0|x_1) \\ L_{t-1} &:= D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) \\ L_T &:= D_{KL}(q(x_T|x_0) \parallel p(x_T)) \end{aligned} \tag{40}$$

In the final iteration, DDPM aims to maximize the probability the model generates a strong sample $\hat{x}_0$ from $-\log p_\theta(x_0|x_1)$. This is done because minimizing NLL is equivalent to maximizing accuracy. These steps ensure DDPM has a strong Langevin sampling from x_t to x_{t+1} or x_{t-1} respectively.

$$L_{simple} = \mathbb{E}_{t,x_0,\varepsilon}[||\varepsilon - \varepsilon_\theta(x_{t,t})||^2] \tag{41}$$

The authors focus on Σ_θ as the driving influence for sampling quality. They utilize L_{simple} which results in $\Sigma_\theta(x_t, t)$ not having much influence on the denoising process. The choice to

train on L_{simple} and optimize for Σ_θ ultimately hindered the performance of DDPM. The researchers optimize the objective loss function to predict ϵ and optimize on 41 rather than L_{VLB} for simplicity. However, more recent research has pointed to $\mu(x_t, t)$ having more influence on controlling generation.

2.2.7 Neural Network Architecture

The neural architecture employed by DDPM follows an upgraded DPM architecture. The key is the introduction of a powerful neural network designed for bio-medical image segmentation called U-Net [37], see Fig. 7.

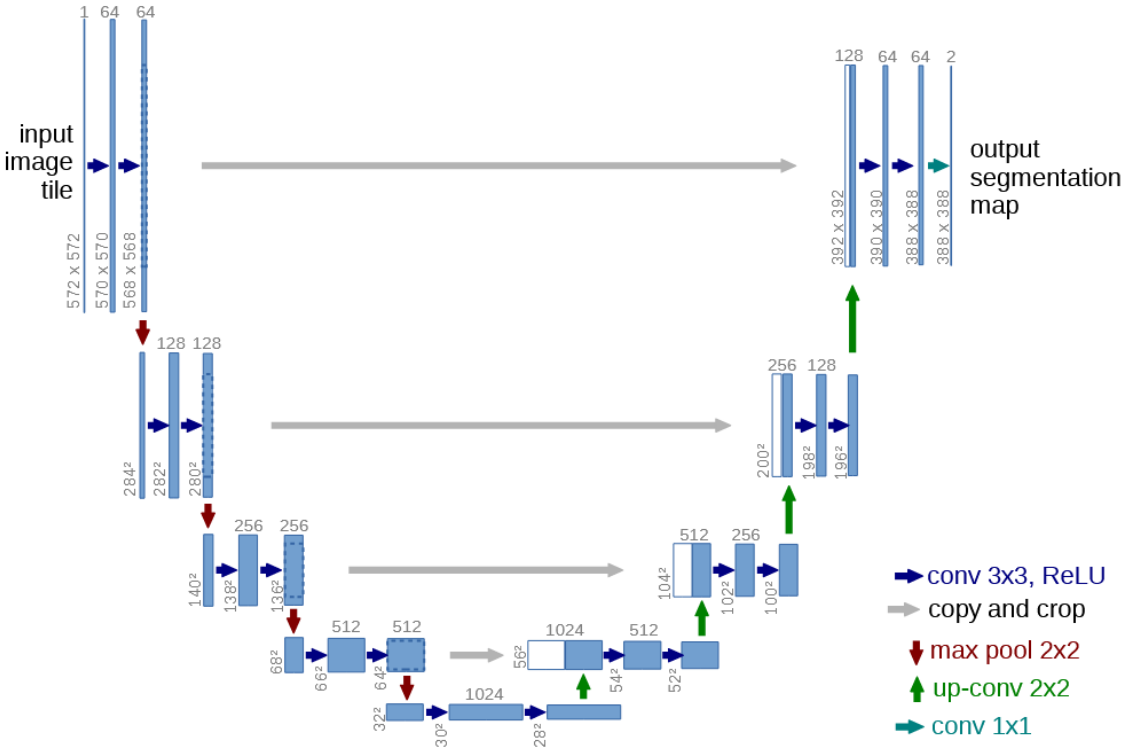

Fig. 7. This is the original U-Net architecture for bio-medical image segmentation. The key design choices to be aware of lie in the continual down-sampling, then up-sampling with the introduction of skip connections. In DDPM's version of a U-Net, they utilize these ideas coupled with the addition of self-attention. Ablations are also made to the convolutional count, weight normalization is substituted with group normalization, and the introduction of a Transformer sinusoidal position embedding for each residual block in attention they use. Figure from [37].

By utilizing self-attention in the residual block, the model learns what spatial sections of the data input have relevance in the final output. The sinusoidal position embedding provides the network with temporal understanding of how noise levels are defined at some t timestep and helps extrapolate to sequence lengths longer than the ones encountered in training. Recently, the academic community has viewed positional embedding as hindrance more so than a benefit.

2.2.8 Discussion

Equation 35 illustrates the proposed scheduled variance, β_t, which can be refined to ensure the expressiveness of the reverse process. This guarantee generates high fidelity and highly accurate reconstructions. This scheduled variance is essentially additive noise defined the underlying covariance matrix. This fixed inferencing process, coupled with powerful latent variable models, can handle arbitrarily many dimensions as an input. The approximately isotropic Gaussian distribution is defined by $\Sigma = \sigma^2\mathbf{I}$ where the covariance matrix is a scalar multiple of the identity matrix. This entails that a distribution has the same variance or standard deviation β_t across all trajectories without maintaining correlations between any pair of dimensions as implied by the diagonal matrix.

The importance of the noise scheduling process can not be overstated [38]. It is responsible for the success of the reverse process in extracting relevant information from the algorithmic scheduled noise β_t. β controls the corruption process which becomes equivalent to controlling variance. The noise scheduler linearly scales between $\beta_1 = 0.0001$ and $\beta_T = 0.02$. β_t and is purposely kept small to control the SNR of inputs x_t, but has the nice property of scaling well with images of larger dimensions. DDPM arbitrarily sets $T = 1000$ diffusion steps because it was enough to synthesize good image reconstructions. The excessive length of T allows the model time to learn how to reconstruct the input. Setting $T = 1000$ timesteps is addressed later in 3 to be an arbitrary bottleneck decision which only requires increased computation to reach arbitrary degrees of perceptible loss. However, T is also very important

as a measure of information loss when coupled with β. Diffusion models offer a level of control in data transformations while maintaining accurate reconstructions which were previously unseen in any other denoising model.

DDPMs represent a compelling and controllable approach to generative modeling, leveraging the intricate connections between noise scheduling, the forward process, and reverse process. Noise scheduling, a pivotal aspect of DDPMs, determines the gradual infusion of Gaussian noise into the input data towards a corrupted state. The neural network learns to approximate the reverse process defined by the noise scheduler and subsequently to denoise the data at each timestep through a Markov chain. This attempts to find an inverse function to accurately reverse the forward process. Ultimately, the success of DDPMs rely on their ability to synthesize realistic reconstructions of the input data such they are indistinguishable from the ground truth.

Since I began writing this body of work, significant breakthroughs have been made to improve this generation process including, single shot generation, controlability of generated images, and more. Diffusion models offer the unique ability to control data transformations and has resulted in a wide variety of models capable of SOTA results across many domains. The results of DDPM have inspired further research into expanding diffusion models to be capable of generating various data forms. To include a small fraction of their capabilities, diffusion models can generate images [39], [40], [41], audio [42], [43], [44], video [45], [46], and various other data types [47], [48]. Diffusion models have also sparked a new class of multi-modal models which allow for switching between increasingly impressive modalities such as text-to-image [49], [50], [51], and image-to-video [52], and text-to-video [53], [54] among others.

These works exhibit only a small fraction of the recent advances in diffusion based data generation. The unprecedented abilities of diffusion models have resulted in immense strides

across various domains presenting some of the most influential and promising work in AI to date.

2.3 Mutual Information Neural Estimation

MINE is a technique which leverages neural networks to approximate MI. Measuring MI has traditionally been a intractable problem. Through the universal approximation theorem [55], MINE seeks to abstract this approximate calculation of MI to become the responsibility of a neural network to measure. Unlike *critic* model architectures, MINE does not suffer from intractability because it uses the gradient plane to determine MI rather than the data distribution itself. This design has allowed MINE to retain linear scaling properties despite arbitrary dimensionality and sample size while simultaneously guaranteeing consistency. This process ensures an expressive approximation of MI while maintaining a controllable degree of accuracy. Using KL divergence we can define MI as the joint probability distribution and the product of the marginals.

$$I(X;Z) = D_{KL}(\mathbb{P}_{XZ} \| \mathbb{P}_X \otimes \mathbb{P}_Z) \tag{42}$$

This derivation is intrinsically expressive and a fundamental feature of all generative modeling. Importantly, KL divergence has many different useful mathematical expressions which ultimately allow for tightening the lower bound of information variation. The key ingredient in MINE is the estimation of the lower bound as defined by:

$$\widehat{I(X;Z)}_n = \sup_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}^{(n)}_{XZ}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}^{(n)}_X \times \hat{\mathbb{P}}^{(n)}_Z}[e^{T_\theta}]) \tag{43}$$

Here, $I(X;Z)$ denotes tracking the information change between timesteps. This equation derives the expected values from $\mathbb{P}_{XZ}$ and $\mathbb{P}_X \otimes \mathbb{P}_Z$. The pair of variables in $\mathbb{P}_{XZ}$ is sampled simultaneously with $\mathbb{P}_Z$ as a marginal probability distribution. The marginal $\mathbb{P}_Z$ also protects the representation and maximizes MI for complex feature extraction. This maximization

becomes increasingly useful as dimensionality of the input is scaled. Synchronized sampling guides the gradient descent process to a minimum and maintains consistency across all features and dimensions.

Through their experiments, researchers discovered that maximizing MI also has the unintended effect of minimizing the expected reconstruction error leading to drastic model improvements. They suspect this is possible due to the marginal of $\mathbb{P}_Z$ describing a latent representation of the input data. These representations capture the intricate details of the data distribution and aids in tightening the lower bound by minimizing the conditional entropy shared between latent variables. As highlighted in previous sections, building a tractable estimator for a VLB requires additional information that is often unknown to the model. MINE's reliance on the gradient plane avoids tractability becoming an issue, as seen in other models, making it a strong contender to measure MI.

The authors propose a competitive model capable of achieving strong results when compared to other bi-directional adversarial models. When comparing MINE results to GAN, MINE performs consistently better in a wider range of tasks and measurements solely by maximizing MI. This conclusion establishes MI as an important metric for complex feature extraction, consistency, and can help improve classification tasks in machine learning applications. MINE is also both strongly consistent and has great sample complexity making it a great ingredient for improving any deep learning task. Furthermore, the researchers apply MINE to the information bottleneck principle in continuous settings, leading to superior results when compared to other bi-directional Markovian methods confirming MINE's capabilities and MI's general importance as a information theoretic tool.

2.3.1 Information Bottleneck and Diffusion

As discussed in 1.6, the layers of a neural network naturally follow the DPI 11 curve. It gradually learns to maximize relevant MI as the information passes through successively deeper layers. The DNN learns to extract information from the signals. At this step we inject a

measurement tool to quantify the transformation of information from input representations to an output representation. As these MI representations become stronger, the model begins to seek a better compressed representations and the retention of relevant information.

Tishby's information bottleneck framework presents a methodology to understand DNNs as relevant information maximizing processes. Recall that mutually shared information can also be defined as entropy, or uncertainty. DDPMs utilize a denoising neural network that learns to predict denoising information from the Guassian input. This lay's bear a crucial component of the theory of information processing through neural networks and diffusion processes. Fundamentally, diffusion and DNNs both rely on the extraction of relevant features by learning to remove noise. Despite the similarity, denoising networks elevate the internal process of DNNs to be the explicit objective. In many ways, we may consider DDPM an explicit representation of the internal dynamics of DNNs. This makes DDPM an enticing avenue to explore the information processing capabilities of DNNs through MI significantly easier and defines the problem space in terms of DPMs.

Similar to neural network layers, diffusion seeks to incrementally transform input data towards an output, while maintaining the same principles from the information bottleneck. Diffusion models target this specific balancing process of learning to retain only relevant information for the reconstruction of the initial inputs. This perfect match of denoising an input lends itself naturally to the information bottleneck principle and the changing dynamics of MI in DNNs. The denoising network effectively takes the processes seen internally in each DNN layer and makes it the objective of the whole network. Diffusion models offer a new tool that parallels the changes of MI through DNNs making it an appropriate mechanism to understand the inner workings of DNNs.

3 PRIOR WORK

For decades diffusion models have seen many use cases and studies across various scientific disciplines. Recently, these use cases have expanded to probabilistic modeling in computational sciences. Compared to other generative model types, diffusion leverages thermodynamics to model complex data distributions with ease. Unfortunately, all generative modeling techniques, including diffusion models, seems to suffer from trade-offs incurred by sample quality, fast sampling, and diversity of generation. In the following section we discuss a small fraction of the works which have been shown to improve, apply, and redefine DDPMs towards optimality without compromising on these trade-offs.

3.1 Ablation Studies to Diffusion Models

3.1.1 Improved Denoising Diffusion Probabilistic Models

IDDPM [56] is one of the first attempted ablation studies conducted on DDPM. The researchers ablate various DDPM techniques to improve sample efficiency and demonstrate the potential for industrial scaling. Even minor architecture changes achieve efficient sampling while improving log-likelihoods. Furthermore, they provide significant insight into DDPMs establishing them as an enticing class of generative models.

Diffusion models are notorious for having poor sample efficiency when compared to other log-likelihood models. The log-likelihood significantly contributes to learning desirable feature representations [57] making its optimization an important factor. One method to advance sampling has promoted reducing the VLB, otherwise known as the ELBO. The researchers report that tightening the variance effectively reduces the search space required during sampling leading to better approximations. Through this, IDDPM significantly lowers the required timesteps from 1000 to 50 while maintaining equivalent sampling quality [56]. Through this, they present a novel approach to track the variance of the noise schedule over an arbitrary sequence S within T diffusion steps. However, as a consequence, the uniform sampling of data causes an infusion of unwanted noise which is addressed through importance

sampling. We defining the sequence of importance sampling here, utilizing $\bar{\alpha}_t$ to represent the noise schedule and $\bar{\alpha}_{S_t}$ to denote the equivalent sequence.

$$\beta_{S_t} = 1 - \frac{\bar{\alpha}_{S_T}}{\bar{\alpha}_{S_{t-1}}}, \quad \tilde{\beta}_{S_t} = \frac{1 - \bar{\alpha}_{S_{t-1}}}{1 - \bar{\alpha}_{S_t}} \beta_{S_t} \tag{44}$$

Sampling variance can be used to scale the data automatically to lessen the necessary steps for T in the forward pass. This is possible because $\Sigma_\theta(x_{S_t}, S_t)$ is bounded between the ranges β_{S_t} and $\tilde{\beta}_{S_t}$. "We can thus compute $p(x_{S_{t-1}}|x_{S_t})$ as $\mathcal{N}(\mu_\theta(x_{S_t}, S_t), \Sigma_\theta(x_{S_t}, S_t))$" [56]. In contrast to the fixed constants β_t featured in DDPM, IDDPMs learned variance of noise produces robust, high quality models while reducing computational inefficiencies.

In order to combat poor sample quality, it was generally understood that one must increase T to an arbitrarily large number. The gradual addition of noise through the forward process, and its subsequent removal in the reverse process, was fundamental to ensure the neural network learns to properly denoise the input data. It stands then, that by increasing the total timesteps used in each pass provides the denoising DNN more time to learn the proper denoising function as each discrete timestep is responsible for a smaller portion of the total additive noise. The researchers study this effect, by setting $T = 4000$, instead of the proposed $T = 1000$. This change marginally improves FID scores and significantly increasing the computational costs. As T becomes larger the model learns to approximate imperceptibly finer denoising details, making T a desirable hyperparamter to improve image quality. They go on to prove their sampling and training ablations successful. They present that minimizing T diffusion steps to 100 produces similar quality images across various image sizes such as 64×64 and 32×32.

The lengthened diffusion process utilizes a longer β_t to reconstruct a finer $\tilde{\beta}_t$. This change improves the learned variance of the reverse process and provides a theoretical foundation for the role of the changing information through additive noise. The essence of the statement is that with a decrease in the timestep size, there's an increase in the granularity of additive noise

encountered at each step, which in turn elevates the variance of noise and diminishes the structural information accessible to the model at every timestep. Fig. 8 demonstrates the perceptibly loss in terms of bits per timestep through the VLB objective.

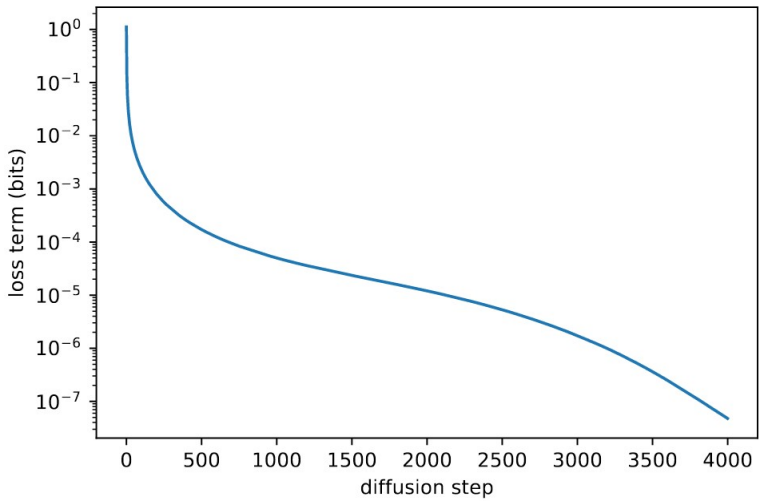

Fig. 8. Bit capture as a function of diffusion steps. Figure from [56].

Consequently, this perceptibility gain demonstrates that the model mean $\mu_\theta(x_t,t)$ has much more influence in determining the final output distribution than $\Sigma_\theta(x_t,t)$. [7] sets $\Sigma_\theta(x_t,t) = \sigma_t^2\mathbf{I}$ where σ_t is not learned by the model. Since the goal is to minimize the log-likelihood we consider the VLB, importantly within the context of T. IDDPM illustrates the decision to fix σ_t^2 as a detriment, since most of the perceptible bits are denoised within the first few hundred steps of DDPM. A few opportunities exist to overcome this inefficiency. Namely, in directly predicting $\Sigma_\theta(x_t,t)$, or the information density, to guide the denoising model. Unfortunately, this approach would be computationally intensive leading to the alternative of parameterizing variance between β_t and $\tilde{\beta}_t$ such that $\Sigma_\theta(x_t,t) = \exp(v\log\beta_t + (1-v\log\tilde{\beta}_t))$. They subsequently indicate that this approach allows for a simplified objective function such that $\Sigma_\theta(x_t,t)$ can guide the objective while allowing granting $\mu_\theta(x_t,t)$ to remain overwhelmingly influential. This solves many of the prior difficulties encountered in such as the influence of Σ_θ and diffusion's sampling efficiency 2.2.

In the following subsections 1.4.1, 3.3.2, we will discuss the theoretical foundations of the proposed improvements from this section, derived chiefly from diffusion experiments and cross-pollination from other various domains.

3.1.2 Non-Gaussian Denoising Diffusion Models

Non-Gaussian diffusion offers another ablation point of diffusion models. DPMs, such as DDPM, have traditionally used Gaussian distributions as their main distribution fitting goal with the desire to obtain $\mathcal{N}(0,\mathbf{I})$. DDPMs noise scheduler follows a set of parameters defined by the variance of noise added at each timestep denoted by 35. As a result, the authors investigate alternative distributions and study their effects on model capabilities. They study Gaussian, Gamma, and a mixture of these distributions.

Gamma distributions differ from Gaussian distributions through a few key points. While both are continuous distributions, Gamma distributions are comprised of only positive real numbers from $0 \leq x \leq \infty$, which can be contrasted with Gaussian's bound between $-\infty \leq x \leq \infty$. The Gamma distribution is also considerably more flexible in its distributional shape. Gamma distributions are highly expressive which make them strong contenders for the flexible and expressive modeling in DDPMs. With simple changes to DDPMs, one can substitute Gaussian's with Gamma distribution such that:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + (g_t - \mathbb{E}(g_t)) \tag{45}$$

Where $g_t \sim \Gamma(k_t, \theta_t)$, $\theta_t = \sqrt{\bar{\alpha}_t}\theta_0$, and $k_t = \beta_t/\alpha_t\theta_0^2$. θ_0 and β_t are hyperparameters determined by Grid search [58]. Through a close formed solution of Gamma distributions, one can derive an equation to define x_t:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + (\bar{g}_t - \bar{k}_t\theta_t) \tag{46}$$

Expanding from the closed form formula, where $\bar{g}_t \sim \Gamma(\bar{k}_t, \theta_t)$ and $\bar{k}_t = \sum_{i=1}^{t} k_i$ one can rewrite the subsequent inferencing process given by the systems Langevin dynamics to incorporate Gamma distributions:

$$x_{t-1} = \frac{x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\varepsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} + \sigma_t \frac{\bar{g}_t - \mathbb{E}(\bar{g}_t)}{\sqrt{V(\bar{g}_t)}} \tag{47}$$

Minor adjustments to the training and sampling algorithms are necessary to encourage stronger model performance. The training algorithm includes changes to derive the appropriate Gamma parameters, updating x_t through eq 46, and updating the gradient descent equation to account for a new set of parameters. The sampling algorithm is changed so that x_t starts the sampling process, with additional updates to the introduction of noise and the update equation.

The mixture of distributions approach is very similar to the introduction of Gamma distributions described above. The replacement of the Gaussian with the Gamma distribution is relatively straightforward, but in mixture distributions the challenge becomes defining the proper mixture of the two distributions. To do this, the authors ablate 35 to normalize the probability distribution at a scheduled pace. This process can then be transformed by introducing C to represent a discrete number of Gaussian variables. This allows for a mixture of distributions at any timestep.

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\left(\sum_{i=0}^{C} z_i \varepsilon_t^i\right) \tag{48}$$

Here, we define C as the number of Gaussian variables used while p_i is the probability that $z_i = 1$ for the i^{th} Gaussian variable for generation.

The authors experiment using DDPM and DDIM testing for single Gaussian, mixture Gaussian, and Gamma distributions respectively. They demonstrate that mixture distributions overwhelmingly provides better FID scores on the CelebA and LSUN datasets. They explain they were limited in testing other distribution types which may offer improvements or other

beneficial results. Notably, there are some distributions which are not adequate for diffusion modeling such as Poisson distributions which are discrete and require independence with regard to time since the last event. Overall, [59] offers another point of improvements for diffusion models and importantly substantiates the versatility of parameters which can be ablated. The authors provide extensive theoretical proofs and have empirical evidence to corroborate their claims.

3.2 Advancements in Sample Efficient Diffusion

3.2.1 Denoising Diffusion Implicit Models

DDIMs are a generalization of DDPMs to a class of non-Markovian processes that are deterministic and subsequently sample efficient with the trade-off of sample quality. They target the DDPM objective through a core property of diffusion where the objective "depends on marginals of $q(x_t|x_0)$, but not directly on the joint [distribution] $q(x_{1:T}|x_0)$" [60]. In this sense utilizing a deterministic or implicit model can result in sampling improvements anywhere from $10\times$ to $100\times$. Importantly, these improvements add only a negligible deterioration in sample quality compared to setting $T = 1000$, while utilizing the same objective function found in DDPM.

When given the same high dimensional data, DDIMs provide consistent latent variable representations resulting in stable high-level feature generation, as opposed to their non-Markovian counterparts. This asserts direct controllability of image synthesis through the latent space.

Fig. 9 illustrates the forward process where each x_t may depend on either X_0 or x_{t-1}. Accounting for the marginals to be conditionally guided by x_0 for regeneration. This forward process is defined for any real vector in $\sigma \in \mathbb{R}^T_{\geq 0}$.

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0)\prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0) \tag{49}$$

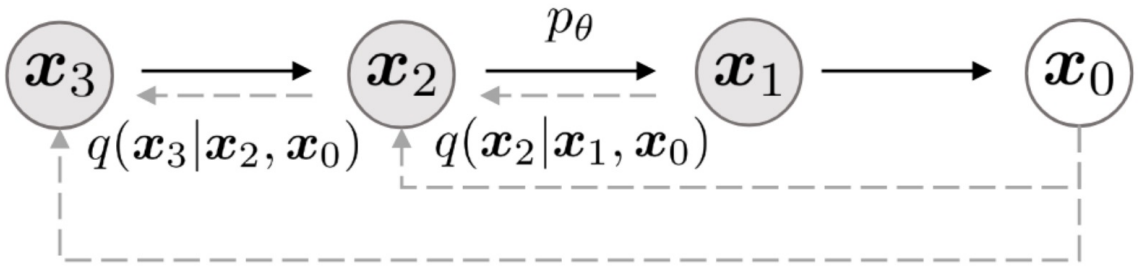

Fig. 9. Non-Markovian inference model proposed in DDIMs. Figure from [60].

This method abstracts DDPM granting dependency on both x_0 and x_{t-1} while σ controls the stochasticity of the forward process, leading to a fixed x_{t-1}. As a consequence of removing the Markovian nature of diffusion, it is necessary to update the reverse process. Through θ, the variational inference objective produces results which appear to require different models for every σ, more on this in 3.3.2. Notably, these proposed ablations illuminate the objective function in DDPM to be equivalent to DDIM's derived objective. Furthermore, this notion extends the information transformation properties of diffusion to non-Markovian spaces and demonstrates a general principle outside of Markovian inferencing and generative processes presented in DDPM.

In the subsequent reverse process, they define $p_\theta(x_{0:T})$ such that each $p_\theta^{(t)}(x_{t-1}|x_t)$ uses the knowledge of $q_\sigma(x_{t-1}|x_t,x_0)$ intrinsically. With this in mind and a fixed prior denoted by $\varepsilon_t \sim \mathcal{N}(0,\mathbf{I})$ the reverse process can be written as the function:

$$f_\theta^{(t)}(x_t) := \frac{(x_t - \sqrt{1-\alpha_t}\cdot\varepsilon_\theta^{(t)}(x_t))}{\sqrt{\alpha_t}} \tag{50}$$

This necessitates a sampling process over $p_\theta(x_{1:T})$ such that x_{t-1} is dependent on x_t and conditional to x_0. The sampling process below demonstrates how one can define diffusion's reverse process, in a non-Markovian case, as the sum of:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{x_t - \sqrt{1-\alpha_{t-1}}\varepsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\varepsilon_\theta^{(t)} + \sigma_t\varepsilon_t \tag{51}$$

They optimize θ over a variational inference objective, seen below, which allows them to reuse DDPMs model training objective.

$$\begin{aligned} J_\sigma(\varepsilon_\theta) :&= \mathbb{E}_{x_{0:T}\sim q_\sigma(x_{0:T})}\left[\log q_\sigma(x_{1:T}|x_0) - \log p_\theta(x_{0:T})\right] \\ &= \mathbb{E}_{x_{0:T}\sim q_\sigma(x_{0:T})}\Bigg[\log q_\sigma(x_T|x_0) + \sum_{t=2}^{T}\log q_\sigma(x_{t-1}|x_t,x_0) \\ &\quad - \sum_{t=1}^{T}\log p_\theta^{(t)}(x_{t-1}|x_t) - \log p_\theta(x_T)\Bigg] \end{aligned} \tag{52}$$

Since the reverse process is equivalent to learning the inverse of the forward process, they define the latent variables $x_{1:T}$ as a subset which matches the marginals. For further efficiency, rather than applying incremental denoising at each timestep as DDPM does, DDIM attempts to reduce total noise at every step towards x_0. The experiments described outperform DDPM even when using the same hyperparameters and models. However, the authors note that differences in results reside in how samples are gathered. DDIM produces faster and more consistent results when comparing high-level feature generation and sampling. Moreover, the overwhelming information changes between x_0 and x_t can be captured in fewer steps with only a minor loss to quality.

3.2.2 Consistency Models

Consistency models were inspired by the continuous time SDE diffusion models seen in 3.3.2) [29]. They follow the same abstract forward, reverse, and denoising techniques from DDPM, but they extend the frontier significantly, by solving many of the fundamental issues associated with generative modeling including sample efficiency. The authors present how diffusion contains SDE dynamics lead by a drift and diffusion coefficient. These coefficients relate DPMs to a specific probability flow ODE. Given this specific probability flow ODE, one can integrate a consistency function on $f:(x_t,t) = x_\varepsilon$ such that the outputs of DDPM are consistent across the same probability flow ODEs.

To parameterize this, consistency models utilize skip connections defined by the following function:

$$f_\theta(x,t) = c_{skip}(t)x + c_{out}(t)F_\theta(x,t), \tag{53}$$

where "c_{skip} and c_{out} are differentiable functions such that $c_{skip}(\theta) = 1$ and $c_{out}(\theta) = 0$ ... are differentiable at $t = \varepsilon$ if $F_\theta(x,t), c_{skip}(t), c_{out}(t)$ are all differentiable" [40]. Through this differentiability, consistency models only require a single pass to generate samples. This comes at the cost of a necessary constraint where the consistency function $f(x_\varepsilon, \varepsilon) = x_\varepsilon, i.e., f(\cdot, \varepsilon)$ is an identity function.

Models which utilize these probability flow dynamics attempt to regenerate something similar to the input data by mapping trajectories to outputs. One possible way to follow this, is to develop a model for each timestep of noise addition, but this approach is infeasible let along computationally intensive; Unless you can one-step generation. Thanks to the Probability Flow ODE, generated data can be mapped to its input data. This creates parings and highly efficient sampling procedures which sample from the transition density of the SDE $\mathcal{N}(x, t_{n+1}^2\mathbf{I})$. In sampling, the goal is to attempt to iteratively denoise. Unlike DDPM, one cares about the trajectory taken. In this regard, the consistency model typically introduces a consistency loss 54 which discourages the model for predictions other than what would be consistent over given timesteps.

$$L_{CT}(\theta, \theta') = \mathbb{E}_{\lambda(t_n)}[d(f_\theta(x + t_{n+1}z, t_{n+1}), f_{\theta'}(x + t_n z, t_n))] \tag{54}$$

Through minimizing the differences associated with each pair of series data points x_{t_n} and $x_{t_{n+1}}$ consistency models learn to quickly solve probability flow ODE. This, in essence, becomes a goal to enforce the self-consistency property of DDPMs. In this regard, each

input-output pair should be very similar, but generation in DDPM is not necessarily exact or even class consistent.

Through consistency models, additional properties of DPMs arise. One example, which is supremely relevant to the body of this work, is that of the ability to utilize transfer learning in diffusion. Ultimately, these training decisions ensure consistency models are capable of one-step or few-step generation.

3.3 Unifying Diffusion Models: Variational Bounds, I-MMSE, and SDEs

3.3.1 Variational Diffusion Models

In a paper titled Variational Diffusion Models [36], researchers achieve competitive log-likelihood scores while also enhancing the VLB through a tractable estimator, demonstrating that this bound can be significantly tightened by formulating an objective in terms of its SNR. Through this notion, they show "...continuous-time VLBs are invariant to the noise schedule, except for the SNR at its endpoints" [36]. This alone defines a pivotal moment in understanding diffusion processes as it is unlike any other previously proposed objective ablation. They go on to demonstrate this change leads to efficient optimization of the noise schedule by directly minimizing the variance of the VLB estimator and show lossless compression rates which achieve a theoretical optimum. They optimize parameters using the traditional VLB of the marginal likelihood given by (26).

The forward process is primarily defined by their SNR function. Here SNR is described to be strictly monotonically decreasing through time, a property consistent with the DPI defined in 11. They define this function to be $\text{SNR}(t) = \alpha_t^2/\sigma_t^2$. When SNR is sufficiently small, x_0 is approximately Gaussian, while a high SNR denotes a close approximation to $q(x|z_0)$. The noise schedule is derived from a corresponding function that employs a fixed schedule, formulated as $\sigma_t^2 = \text{sigmoid}(\gamma_\eta(t))$. Here $\gamma_\eta(t)$ defines a monotonic neural network. Due to assumptions that α and σ are differentiable, it becomes trivial to convert DDPM's reverse process to continuous time.

$$p(x) = \int_z p(z_1)p(x|z_0)\prod_{i=1}^{T} p(z_{(i-1)/T}|z_{(i/T)}) \tag{55}$$

Continuous time necessitates that $T \to \infty$ which effectively amounts to diffusion loss coinciding with the integral of the MSE over SNR. Through a series of simplifications and replacements T can be defined to follow $\text{SNR}^{-1}(v)$. In this context, the loss function is constrained between SNR limits, such that as T trends towards infinity, only the maximum and minimum values of SNR stay relevant. In doing so, the diffusion loss becomes invariant of the SNR function between SNR_{min} and SNR_{max} except w.r.t the VLB and its endpoints.

$$\mathscr{L}_{\infty}(x) = \frac{1}{2}\mathbb{E}_{\varepsilon\sim\mathscr{N}(0,\mathbf{I})}\int_{\text{SNR}_{min}}^{\text{SNR}_{max}} ||x - \tilde{x}_{\theta}(z_v, v)||_2^2 dv \tag{56}$$

The shift to a continuous time framework, paired with an SNR noise methodology, provides SOTA results across all evaluations, both in metrics and datasets. The researchers highlight key characteristics of diffusion involving the denoising sequence as a process dependent on the SNR. This objective changed and proposed methodology provides a variance minimization technique to tighten the VLB.

Variational diffusion models can also be defined by MMSE which are related to the lower bound on log-likelihood. MMSE is related to the VLB by the reparameterization trick, which allows gradients to be computed with parameters. In diffusion the variational inferencing allows for MMSE to minimize MSE, which effectively minimizes KL divergence between ground truth and variational distributions.

3.3.2 Score-Based Modeling through Stochastic Differential Equations

In an important extension to Variational Diffusion Models 3.3.1, a score-based generative modeling approach has lead to the relation of diffusion processes to SDE through:

$$dx_t = \mu(x_t, t)dt + \sigma(t)dw_t \tag{57}$$

Researchers utilize this SDE approach to smoothly transform complex data from a known prior and the learn time reversal. The approach abstracts DDPM even further than [60] by using a continuous distribution to interpolate over timesteps. Through this abstraction, researchers were able to flexibly control the generation of data by conditioning on information that was unknown during inferencing. Remarkably, the reverse-time SDE can be denoted by the following equation:

$$dx = \big(f(x,t) - g(t)^2 \nabla_x \log p_t(x)\big)\, dt + g(t) dW \tag{58}$$

In this context W denotes a standard Wiener process, under Brownian motion, that starts from T and ends in 0, while dt stands for a small finite timestep. The score of a distribution may be estimated through a score-based modeling approach during training where the continuous time generalization necessitates the following function:

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{x(t)|x(0)} \Big[\| \mathbf{s}_\theta(x(t),t) - \nabla_{x(t)} \log p_{0t}(x(t)|x(0)) \|_2^2 \Big] \right\} \tag{59}$$

The researchers note that with a sufficient amount of data and sufficient model capacity, this score matching approach ensures that the optimal solution is reached, denoted by the function $\mathbf{s}_\theta(x,t)$. Where each point accurately captures the data density $\nabla_x \log p_t(x)$ for nearly all instances of x and t. Typically, the transition kernel $p_{0t}(x(t)|x(0))$ is necessary for efficient computation, however they bypass this requirement by replacing the denoising score matching with sliced score matching such that Eq.59 becomes:

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{x(t)} \mathbb{E}_{v \sim p_v} \Big[\frac{1}{2} \| \mathbf{s}_\theta(x(t),t) \|_2^2 + \mathbf{v}^T \mathbf{s}_\theta(x(t),t) \mathbf{v} \Big] \right\} \tag{60}$$

Additionally, utilizing a score-based modeling approach to solve a reverse-time SDE the researchers reinterpret the diffusion process to a deterministic one defined by marginal probabilities for an ODE:

$$dx = \left[f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x) \right] dt \tag{61}$$

In many instances, ODE's may be considered a deterministic limit of an SDE where the average dynamics of the SDE can be described as a trajectory of an ODE. Within probability density estimation for $p(x,t)$ the ordinary differential equation corresponding to an averaged SDE becomes a deterministic sampler which gives exact solutions to the likelihood of any input data. This exactness allows for the reverse process to be uniquely associated with an input such that the trajectory is always equal from noise to data. This also introduces both privacy concerns, as training data may be reproduced through sampling. Significant improvements to sampling quality can also be realized through larger error tolerance. Furthermore, through the encoding process on x_0 and decoding process through the proper ordinary differential equation channel, these latent representations become flexible abstractions which may be molded for image manipulation techniques.

3.3.3 Information-Theoretic Diffusion

Recently, [11] demonstrated core relationships to the foundations of this thesis relating MI with MMSE which they call I-MMSE. "We [the authors] generalize the I-MMSE relations to exactly relate the data distribution to an optimal denoising regression problem, leading to an elegant refinement of existing diffusion bounds" [11]. This relationship also exposes fundamental ideas related to probability distribution estimation. Previously, [61], proposed a fundamental connection between input-output pairs of MI and MMSE which states that SNR is equal to half the MMSE for any optimal estimation.

As the authors point out, variational diffusion models 3.3.1 demonstrate that diffusion models are defined by MMSE on the lower bound on the log-likelihood because MMSE minimizes KL divergence.

From these intuitions the authors present a fundamental relationship of denoising by MMSE and KL Divergence:

$$\frac{d}{d\text{SNR}} D_{KL}[p(z_y|x)\|p(z_y)] = \frac{1}{2}\text{MMSE}(x,\text{SNR}). \tag{62}$$

The marginal is defined by $p(z_y) = \int p(z_y|x)p(x)dx$ and MMSE is a pointwise estimation:

$$\text{MMSE}(x,\text{SNR}) \equiv \mathbb{E}_{p(x|z_y)}[\|x-\hat{x}^*(z_y,\text{SNR})\|^2]. \tag{63}$$

Through thermodynamic integration [62], they demonstrate an expansion to Eq. 62:

$$\begin{aligned} -\log p(x) \leq\ & D_{KL}[p(z_{\text{SNR}_1}|x)\|p(z_{\text{SNR}_1})] \\ & + \mathbb{E}_{p(z_{\text{SNR}_0}|x)}[-\log p(x|z_{\text{SNR}_0})] \\ & - \frac{1}{2}\int_{\text{SNR}_1}^{\text{SNR}_0} \text{MMSE}(x,\text{SNR})d\text{SNR} \end{aligned} \tag{64}$$

"... where evaluation of the endpoints corresponds to a difference in the free energy or log partition function, and the derivatives of these quantities may be more amenable to Monte Carlo simulation" [11]. Through strong applications of thermodynamic integration, the authors demonstrate an expansion to continuous time integration where $\text{SNR} \in [0,\infty)$. Here the variational bounds in diffusion are determined by SNR samples from $\alpha \sim q(\alpha)$. The log-likelihood can be exactly defined by a simplified Gaussian density function which is the global optimum of denoising MSE, which can be rewritten as a regression problem.

$$h(p) \equiv \mathbb{E}_{p(x)}[-\log p(x)] = \frac{d}{2}log(2\pi e) - \frac{1}{2}\int_0^\infty d\text{SNR}\left(\frac{d}{1+\text{SNR}} - \text{MMSE}(x,\text{SNR})\right) \tag{65}$$

Through these generalizations, an exact relationship between data probability and data distribution is related to the optimal denoiser Eq. 63 and show regression is a fundamental estimator for probability densities such as DPMs.

3.4 Diffusion Model Applications

3.4.1 Image Super-Resolution via Iterative Refinement

In one of the earliest applications of DDPM, Image Super-Resolution via Iterative Refinement [25] up-scales images through an unconditional diffusion model. The paper demonstrates the potential to upscale images by $8\times$, from 16×16 - $> 128 \times 128$. Crucially, this diffusion model is trained to reproduce high resolution images through a stochastic iterative refinement defined as $p_\theta(y_{t-1}|y_t, x)$, where x and y are pairs of images representing an input of lower dimensionality and output of higher dimensionality respectively.

The forward process is defined as:

$$\begin{aligned} p_\theta(y_{0:T}|x) &= p(y_T)\prod_{t=1}^{T} p_\theta(y_{t-1}|y_t, x) \\ p(y_T) &= \mathcal{N}(y_T|0, \mathbf{I}) \\ p_\theta(y_{t-1}|y_t, x) &= \mathcal{N}(y_{t-1}|\mu_\theta(x, y_t, \gamma_t, \sigma_t^2\mathbf{I}) \end{aligned} \tag{66}$$

The inference process $p_\theta(y_{t-1}|y_t, x)$ is learned and its reverse process becomes an approximate Gaussian. When given an image from an arbitrary timestep, y_0 can be approximated through:

$$\hat{y}_0 = \frac{1}{\sqrt{\gamma_t}}(y_t - \sqrt{1-\gamma_t}\, t\ (x, y_t, \gamma_t)) \tag{67}$$

$\hat{y}_0$ defines the variance of the forward process and can be repurposed to approximate the reverse process into the posterior distribution. Similarly to DDPM [7], the authors use an equivalent inference training process, but replace the posterior distribution with a new

description to find y_{t-1} as seen in Eq. 68. The models trajectory is effectively guided to reconstruct the image while retaining high-level features.

$$\hat{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(x, y_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\varepsilon_t \tag{68}$$

Previous attempts at increasing image fidelity have utilized bicubic and regression systems with various degrees of accuracy. Diffusion models provide an unprecedented level of image reconstruction on imperceptible details that, when zoomed in, demonstrate significant fidelity improvements. Despite this, Super-Resolution via iterative refinement suffers from multiple issues including biases such as dropping finer facial details. Super-Resolution also provides near equivalent results on PSNR and the structural similarity index measure, but significantly improves on previous FID scores.

The importance and non-trivial nature of ensuring consistency to represent high level features is crucial to upscaling images. As a result, utilizing consistency models, as proposed in [40], may yield better reconstructions. Super-Resolution via iterative refinement utilizes the strengths of diffusion modeling by harnessing its ability to augment and reconstruct data towards a the desired distribution.

3.4.2 Residual Diffusion Based Compression

Using the generative capabilities of diffusion models, [63] developed a lossy neural codec used for rate-distortion interpolation during test time called DIRAC. Typical codecs often suffer from a trade-off between perceptual quality and fidelity, however neural codecs suffer from the triple trade-off of rate-distortion-perception. DIRAC solves this through a dynamic neural codec, capable of dynamic rate control at inference time and has the added bonus of being integratable alongside other codecs. They go on to demonstrate that with a reduced representation of data, they can sample with significantly reduced step counts.

Consider a noisy channel 1.1.6 where data is compressed before entering the channel and decoded after leaving the channel. It is computationally efficient to reduce the data sizes

before transmitting data through a channel. Conversely more compressed the representation is, the more perceptual loss accrues. DIRAC solves this by utilizing a dynamic rate-controller.

DIRAC utilizes a DDIM [60] and a novel residual method proposed by [64] where the conditional probability of $p(x_0|\tilde{X})$ is replaced by $p(r_0|\tilde{X})$ where $r_0 = x - \tilde{X}$. Notably DIRAC avoids using JPEG as the default to restore reconstructions because the JPEG reconstructions are lossy. When mixed with less desirable forms of reconstruction such as PSNR. This can result in worse reconstructions. Instead, DIRAC uses LPIPS [65] and FID [66] to evaluate any perceptible distortions. These notions effectively allow the development of a new loss term where r'_0 defines the model output from the residual at a given timestep. W_t defines a weighting factor to help small or large t.

$$\mathscr{L}(x,\tilde{x}) = \mathbb{E}_{t,r_t}[w_t||(r_0 - r'_0)||^2 + \lambda_{\text{LPIPS}} d_{\text{LPIPS}}(x, \tilde{x} + r'_0)] \tag{69}$$

Through their experiments researchers demonstrate DIRAC as a competitive model when compared on PSNR, despite being trained on LPIPS. This approach allows for users to control rate, distortion, and perception at test time. Many other approaches to improve compression have been developed from GAN based architectures [67], to diffusion ones [68], [69], but DIRAC is the first to leverage dynamic reconstruction between fidelity and perception quality.

3.4.3 MINDE: Mutual Information Neural Diffusion Estimation

In MINDE an explicit, tractable estimator for MI is presented. Utilizing score-based diffusion models [29] to estimate KL divergence as the difference between two score functions. Finding the exact score function is not necessarily tractable, but through by parameterization θ they can derive a score network based on minimizing the loss 70 found in prior work [29], [70], [36].

$$L(\theta) = \mathbb{E}_{\mathbb{P}^\mu}\left[\int_0^T \frac{g_t^2}{2}\left\|\tilde{s}_t^\mu(X_t) - \nabla \log\left(\bar{v}_t^{\delta_{x_0}}(X_t)\right)\right\|^2 dt\right], \tag{70}$$

Following this, the researchers demonstrate that encoder and decoder functions, as seen in VAEs and diffusion, the KL divergence can be computed in the latent space. This approach derives neither upper nor lower bounds of the true KL divergence, but does allow a tractable estimator for the forward process. In the reverse process however, the authors utilize Monte Carlo integration because "... analytic computation ... is in general, out of reach" [71]. Contrarily, in this work, we will demonstrate an accurate estimator of MI capable of performing in the forward and reverse process.

To calculate the MI between two random variables A, B the researchers must define the marginals μ^A, μ^B respectively, joint $\mu^C = (C = [A, B])$, and conditional $(A|B) = \mu^{A_y}$ and $(B|A) = \mu^{B_x}$ measures. μ^C and μ^{A_y} values are then introduced into the score-based diffusion process. This sets the foundation of a novel diffusion model which can model the joint and conditional measures according to the following SDE:

$$\begin{cases} d\left[X_t, Y_t\right]^\top = f_t\left[aX_t, \beta Y_t\right]^\top dt + g_t\left[adW_t, \beta dW_t'\right]^\top, \\ \left[X_0, Y_0\right]^\top \sim \mu^C, \end{cases} \tag{71}$$

The key intuition for this SDE is that only a single score network is necessary as it blends the formulation in 71. To calculate MI in their joint diffusion model they utilize the equation $H(C) - H(A|B) - H(B|A)$ and derive a new loss function to predict $I(A, B)$.

$$I(A,B) \simeq \mathbb{E}_{\mathbb{P}^{\mu^C}}\left[\int_0^T \frac{g_t^2}{2}\left[\left\|\tilde{s}_t^{\mu^C}([X_t, Y_t]) - [\tilde{s}_t^{\mu^{A_{Y_0}}}(X_t), \tilde{s}_t^{\mu^{B_{X_0}}}(Y_t)]\right\|^2\right] dt\right] \tag{72}$$

3.4.4 Imitating Human Behaviour with Diffusion Models

The authors of this work propose using diffusion models to imitate "...human behaviour, since they learn an expressive distribution over the joint action space" [27]. They incorporate reinforcement learning into their diffusion model to manage diverse trajectories which may not be captureable using standard diffusion practices. This is possible because diffusion models

are capable of learning complex state-action relationships. This lead to improved accuracy on common RL control tasks from prior SOTA results of 44% to 89%.

Using DDPM as a foundation, Pearce et al explores the use different neural network architectures including the transformer and CFG. CFG is a neural network trained for both conditional and unconditional modeling, but fundamentally one which balances a trade-off between data consistency and data diversity. To allow DDPM based models to learn diverse state actions spaces, like that in imitating human behavior, they ablate the predicted noise and variance scheduler to account for these spaces. Using human training data as a foundation, they define the distribution $\mathscr{D}$ as $\mathscr{D} \sim o, a$, and set $\tau \sim \text{Uniform}[1, T]$. Their predicted noise follows the following equations which seeks to minimize a variation of MSE loss given an observation/state o and an action a.

$$\mathscr{L}_{\text{DDPM}} = \mathbb{E}_{o,a,\tau,z}[||\varepsilon(o, a_\tau, \tau) - z||_2^2] \tag{73}$$

At sampling time, they redefine the variance scheduler to account for states and actions:

$$a_{\tau-1} = \frac{1}{\sqrt{\alpha_\tau}} \left(a_\tau - \frac{1-\alpha_\tau}{\sqrt{1-\bar{a}_\tau}} \varepsilon(o, a_\tau, \tau) \right) + \sigma_\tau z \tag{74}$$

After "baking-in" these ingredients, the CFG becomes defined by its weighted guidance metric w which gives more weight to the conditional variable prediction "cond." when $w > 0$ and results in higher values of $p(o|a)$. This approach also gives a negative weight on the unconditional prediction "uncond." . as defined by:

$$\hat{z}_\tau = (1+w)\varepsilon_{\text{cond.}}(a_{\tau-1}, o, \tau) - w\varepsilon_{\text{uncond.}}(a_{\tau-1}, \tau) \tag{75}$$

From these ideas, they test different diffusion methods they define as Diffusion-X and Diffusion-KDE. These diffusion methods differ only in the sampling methodology where the researchers suspect that encouraging higher-likelihood actions during sampling will lead to

faster modeling. Diffusion-X is defined by continuing additional denoising over M timesteps after the traditional sampling is complete. This likely leads to higher-likelihoods. [27] also proposes Diffusion-KDE which utilizes a KDE to score and extract the highest likelihood over all samples. These ablations result in improvements to out-of-distribution actions and result in better performance in complex environments.

The researchers mix transformer and MLP architectures with their ablated diffusion methods to produce SOTA results. Interestingly, they analyze the CFG and its bias w.r.t distributions. They find that CFG creates a bias towards following unconventional trajectories, but almost always solves the human evaluation task. Conversely, in the absence of CFG they find that their approach is only able to complete tasks 63% of the time.

Notably, they train their model to play videos games. Using a dataset of human actions from the popular game ”CS:GO”, they are able to produce a SOTA model achieving 24 points out of the human baseline of 36. Prior competitive models achieved a score of 18 points. They test performance by placing an agent in a fixed position and giving it the goal defending itself against all approaching enemies.

In conclusion, the researchers demonstrate the capability of their model in imitating human behaviors successfully. They demonstrate diffusion to be an ideal match for learning observation-to-action distributions. Using a mix of diffusion ablations and avoiding coarse gradients by increasing likelihood-estimations they achieve SOTA results in a range of complex tasks.

3.5 Insights on the Transformation of Information in DDPMs

3.5.1 Deep Neural Networks

DNN’s have been widely accepted to be ”black boxes” due to the youth of the field. Fundamentally, DNN’s represent a simplified and digitized version of the neural circuitry in a human brain. For humans, data is first captured by our senses then propagated by signals and path ways to the brain. In the brain, these sense representations are passed through a series of

neuron chains where the action potential determines if the signal will continue to propagate and to which neurons this signal will propagate to. Over time, neurons which are used often are reinforced while neurons that are unused deteriorate. This neural plasticity helps our brains learn to represent and act on the world around us.

In a computational neuron, data x is compiled and ingested by the network through input neurons. These neurons go one to propagate their signal to the next neuron layer, known as the hidden layer. Neurons contain a weight w associated with the connections between neurons which signifies its influence. This is multiplied by the sum of observation data signals on x and a bias b to adjust. Once this weighted sum is calculated, an activation function will be applied. The activation function will convert the input signal to an output signal. These signals are compared with a threshold set by an activation function then propagated. In the final output layer, the representations of signals will collapse the prior layer's output signal to a predicted representation denoted by $\hat{y}$. A cost function then evaluates $\hat{y}$ based on a reward metric to the expected data y. This final reward determines the models predictive accuracy. To induce learning and improve prediction accuracy, backpropagation is employed to send update signals through the DNN in reverse. These signals update the weights and biases of each neuron in the DNN.

While the complexity of DNN types are vast, fundamentally they take an input and propagate signals of that input to produce an output representation. The neuron count per layer and even the number of hidden layers are conceptually arbitrary since only three hidden layers are necessary to approximate any function [55]. Although it must be noted that networks with more layers and tailored neuron counts tend to learn better representations and improved accuracy's when compared to smaller networks.

We can study DNNs by considering them as layer-wise Markov chains defined by MDPs. Each neuron layer acts as a decision making process. MDPs are defined by states, actions, probabilities, and rewards. In this context, the state defines the output signal from a prior layer,

the action is the transformation applied by the layer over all its neurons, the probability is the stochastic signal propagation, and the reward is the correctness of the final output $\hat{y}$. MDPs provide a way to study DNNs through the lens of information theory, ultimately providing mechanistic interpretability of DNN model dynamics.

Various attempts have been made to explore the information processing capabilities of DNNs. [72] demonstrated that DNNs can be quantified by the MI between layers. They show with training, feature representations become increasingly compressed towards the bounds of minimally sufficient statistics 1.1.5, effectively mapping the input variables into a representation which preserves the maximal amount of information for exact reconstruction on a lower-dimensional plane for the output y. MDPs follow DPI such that deeper DNN layers will have access to less information than earlier layers. One way to imagine this is that as x passes through each hidden DNN layer, the relevant feature representations from x are decoupled from unintended observation noise. We can define this relevant feature information as a function of SNR, or by the MI through the assumption of statistical dependence between x and y. MI can effectively be bound by the prediction error. Maximizing MI for DNNs is a natural optimization goal as the more information that is shared between the input data x_0 and output prediction $\hat{y}$, the stronger the prediction will be.

[30] extends the work proposed in [72] to further study the effects of training on DNNs. They show two phases of SGD known as drift and diffusion. In the first phases known as drift, MI between the input x and output y is rapidly increased over a few hundred training epochs. Here the average gradient fluctuation purports high SNR. This phase controls the rapid reduction in the prediction error from the cost function. Once sufficiently trained to maximize MI extraction, the SGD will naturally attempt to compress the representations in the second, far longer diffusion phase, which leads to low SNR. The diffusion phase here represents the distribution of x as being maximized by its entropy weight distribution based on the training error, or as a minimization of MI. The diffusion phase can also be represented by the

Fokker-Planck equation, which describes the temporal evolution of a representative probability density function as the ”speed” of data transformations while bound by random forces. This illuminates a deeper connection between diffusion processes and DNN mechanics, as DPMs are guided by the Fokker-Planck for score-matching functions [73].

During training, DNNs learn to extract relevant features which contribute to reducing the prediction error. This amounts to optimizing SNR. DNNs can then be interpreted as a denoising process. Interestingly, data augmentation techniques to introduce noise to the input data during training results in robust models due to learning better signal representations to circumvent noise. Ablating various other points including layer count, neuron count, training times, and objective functions can also lead to improved model prediction accuracy.

3.5.2 DDPMs, DNNs, and Information

DDPMs and DNNs share many conceptual commonalities. Both systems are defined by MDPs, where DNNs pertain to layer count and DDPMs pertain to timesteps. Where DNNs intrinsically maximize MI between input-output pairs through maximizing SNR at each layer, denoising networks explicitly learn to extract signals from a noisy latent in the reverse process. DDPMs and DNNs also share the same types of objective functions. Functions like L2 loss consist of the squared error $L2 = (Y - \hat{Y})^2$ and MSE $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ are commonly used as objectives in both systems. However, these objectives often fail to capture granular data intricacies. While DDPMs utilize a denoising U-Net, the similarities go beyond a simplistic look at DNNs. Many of the same concepts from the prior section 3.5.1 are applicable to DDPMs including maximization of SNR and MI, and MDP chains. DNNs have been shown to be equivalent to Gaussian processes when network width is infinitely wide [74]. The authors express that this allows neural networks to be trained for regression objectives. DDPMs, also consist of Gaussian processes. [11] has shown that DDPMs can be characterized and defined by regression objectives with continuous and discrete probabilities [11], This

connection illuminates a broader correlation between DNNs and DDPMs which escape the scope of this work, but set contextual understanding for the following sections.

DDPMS and DNNs also have distinctive properties. DDPMs are AWGN channels 14 which explicitly and incrementally add noise according to a variance scheduler β. This deterministically transforms the latent variables true and expected SNR through time. Learning these SNR expectations in the reverse process amounts to learning denoising. DNNs do not add noise, but instead only focus on this reverse process to separate signals from noise. Data augmentation techniques to introduce noise of varying levels to data effectively does the same process, but learns a single representation instead of the chained representation present in DDPMs. However, DDPMs also utilize a similar technique where data can be algorithmically corrupted to any $t \in T$ then used to train the reverse process, making denoising agnostic to timesteps. While DNNs are intrinsic SNR maximizers, DDPMs utilize hyperparameters to achieve the same goal. We can thus study the effects of these hyperparameters on information processing. This provides an opportunity to explore various information-theoretic properties entropy, MI, information bounds, and opens various directions of research to optimize and apply diffusion.

For the remainder of this work, and particularly this section, we explore diffusion through the lens of incremental channels as shown in [61]. Guo et al. lays much of the foundational theory behind diffusion models through AWGN channels. Their key contribution is the fundamental relations between MI, optimal MMSE, and SNR through incremental channels. The relation between DNNs and DDPMs are thus characterized by these connections. We define these connections explicitly in the following paragraphs.

Developing these information-theoretic principles, we can explore the forward process of diffusion in a deterministic setting and simultaneously ground the reverse process in the same principles. The fundamental difference between the forward and reverse are the algorithmic corruption and the stochastic denoising done by a neural network. Guo et al. [61] eloquently

defines the following equations where we assume X to be input data, Y is a single channel, $\mathscr{N}$ is an independent Gaussian random variable, σ is the addition of noise, δ defines the total timesteps, and are represented by the i-th timestep in the sequence. Each channel can be defined by its input data and added noise.

$$\begin{aligned} Y_1 &= X + \sigma_1 \mathscr{N}_1, \\ Y_2 &= Y_1 + \sigma_2 \mathscr{N}_2, \end{aligned} \tag{76}$$

Since this is clearly a Markov chain where $X \to Y_1 \to Y_2$, we can apply the chain rule of MI which states that successive chains of independent processes can be reduced to the sum of conditional MI terms for $I(X;Y_1|Y_2)$ in the following:

$$I(X;Y_1) = \sum_{i=0}^{T} I(X;Y_i|Y_{i+1}) \tag{77}$$

As the SNR increases through each independent incremental channel, Eq. 77 redefines Eq. 76 as $(\text{SNR}+\delta)Y_1 = \text{SNR}Y_2 + \delta X + \sqrt{\delta}\mathscr{N}$. Where the SNR is the primary difference over time. Through this process [61] shows that conditional MI can be expressed as half the expected value of the squared different between X and its conditional state given the incremental Gaussian channel of Y_2 with the added noise $o(\delta)$.

$$I(X;Y_1|Y_2) = \frac{\delta}{2}\mathbb{E}\big\{(X - \mathbb{E}\big\{X|Y_2\big\})^2\big\} + o(\delta) \tag{78}$$

As the number of successive channels increase, Eq. 78 and Eq. 77 show that as $\delta \to \infty$ SNR channels with low information signals correspond to MMSE multiplied by additive SNR leading to:

$$I(\text{SNR}) = \frac{1}{2}\int_0^{\text{SNR}} \text{MMSE}(\gamma) d\gamma \tag{79}$$

This result applies to successive Gaussian channels and can be studied from the score-based SDE interpretation of DDPMs 3.3.2. However, the seminal results of Guo's paper abstract this notion by demonstrating a fundamental relationship 80 of MI being the accumulation of MMSE as a function of SNR. Where "...the derivative of the mutual information (nats) with respect to the signal-to-noise ratio (SNR) is equal to half the MMSE, regardless of input statistics" [61].

$$\frac{d}{d\text{SNR}} I(\text{SNR}) = \frac{1}{2}\text{MMSE}(\text{SNR}) \tag{80}$$

This demonstrates MI and optimal estimation of MMSE in Gaussian channels are connected. Through increasing MI, the SNR must increase according to half of MMSE. [11] introduced the application and connection of DDPMs to Guo's theoretical work and arrived at the results we illustrate above. The researchers demonstrated the same intimate connection between DDPMs, MI, and MMSE in Gaussian channels. In summary, MI, MMSE, and SNR are deeply related to SDEs like DDPMs, and increasing the MI should also increase the MMSE leading to improved model capabilities and providing an information-theoretic approach to studying these systems.

3.5.3 Information in DDPMs

The incremental MDP nature of DDPMs provides a natural point to sample from. In the reverse process the denoising procedure is determined by a few key DDPM components; Namely the number of T timesteps, the noise scheduler β, and the objective function. Recall that [7] uses a linear scheduler β from 0.0001 to 0.02, sets timesteps to 1000. The noise perturbations are added according to β_t. Through this incremental noise, a gradual decay in information takes place within the input data. The minuscule and uniform additions of noise provides many benefits. Recall 1.1.1, uniform distributions maximize differential entropy and makes them robust to small noise perturbations like the ones set by the linear scheduler.

Maximizing differential entropy amounts to finding the maximum uncertainty in a probability density function. Information is destroyed uniformly, resulting better estimation of the probability density function like DDPM.

Consider input data in the form of a two-dimensional image. One can describe the input data as a complex probability density function. As noise is added, the rich information of the probability density function is gradually replaced by Gaussian noise. Over T steps, these additive noise perturbations sum to an approximately Gaussian distribution. The approximate nature of x_T contains information in the random perturbations when compared to a perfectly Gaussian distribution. A well trained neural network can take these small differences in distributions through KL divergence and extrapolate this information back into the input data. In practice however, many variables can influence this exact regeneration process. The most important hyperparameter is T, followed by β, then the objective function. The larger T is, the finer the information addition. While the smaller T is, the more noise is added. Simultaneously, β and T are intimately connected where β bounds the added information between two points and T divides the information increments uniformly. With sufficient added noise, the data can be rapidly corrupted leading to a loss of the initial information signal. The denoising U-Net in DDPM is tasked with predicting noise at a given timestep, but this is subject to the chosen objective function which may also be inefficient at capturing small differences in the distributions. The neural network may also fall victim to hallucinating information from other classes, and cause regeneration of another class or a degenerate output. However, in generative modeling, it is often the case that desired reconstructions are novel samples, not reconstructions from the training data.

Contrasting VAEs with DDPMs provides theoretical insights into generative models and information processing. Where VAEs are hierarchical, DDPMs are global information estimators because the latent dimensions are equivalent throughout training. Diffusion focuses on learning the information representations from the features sampled at every x_t. This

information is dependent on that tight coupling between T and β_t. When T is sufficiently small DDPM enables exact reconstructions of x_0, subject to the minimally sufficient statistics 1.1.5 for reconstruction being present in the latent. As T is increased, the reconstructions begin to degrade such that they generate alternative classes from the training distribution. This discrepancy shows there are training dynamics responsible for class dependency and reconstruction consistency. There must then exist a T and β_t that produces the minimally sufficient statistics to accurately reconstruct x_0 from x_T. Consider the optimal diffusion time to be a product of $T \in 0 \rightarrow \infty$. Assuming a sufficiently large T, there must exist a time which optimally maximizes the difference between the KL terms of the ELBO. Training for an infinite diffusion time, as assumed in score-based diffusion models, does not maximize the ELBO [75].

As described in prior the sections 1.3.2 and 2.2.6. The ELBO quantifies the lower bound of the quantified log-likelihood of the input data. By optimizing this lower bound one can effectively utilize the ELBO as a proxy optimization objective for rich feature representation and better generative modeling. The ELBO is strictly non-negative following KL-divergence. This is because we want to match the variational posterior of $q(z|x)$ where z is the latent variable, and the true posterior given by $p(z|x)$ through minimization of the KL divergence. Since we do not have access to the true posterior distribution, we instead seek to maximizing the ELBO as it becomes an equivalent minimization of KL divergence. This fundamentally relates the ELBO to the optimal objective function.

Significant research has been conducted on continuous time diffusion models and optimization of the ELBO [36] and [76]. They demonstrated better reconstructions capabilities by maximizing the lower bounds. In particular, one work has explored an interesting relationship between the ELBO and the SNR of a given timestep. The authors of [77] posit that the optimal ELBO objective should evolve with each timestep given the changes to the SNR of the latent variables. With each x_t there is an expressive ELBO for that SNR. The

optimal DDPM objective must then be the sum of weighted SNR integrals over the ELBO. Meaning that, from this perspective, prior objective functions were not expressive enough to capture the intricacies of the data input. Without additive noise perturbations DDPM approaches the ELBO depending on the objective function being used. Improving on this, DPPMs can directly match the ELBO when subject to monotonic weighting. Interestingly, in the foundational DDPM paper [7] they derive MSE as the optimal objective for diffusion, but utilize mean absolute error in training their implementation.

By chaining the sum of weighted integrals we can get an exact ELBO calculation for DPMs. This ELBO, a product of the interplay between likelihood estimation, AWGN, and MDP, provides a way to quantify the transformation of information in diffusion at any given timestep. The lower bound signifies the minimal information of the system. Utilizing the notion of minimally sufficient statistics for exact reconstructions and likelihood training for the ELBO, one can study the effects of ablations to T and β through the MI and MMSE connection presented by Guo et al. [61] and derived by [11]. Reconstructions are still subject to the objective function. The objective function may or may not the capture expressive distributions diffusion is trained on, but the better the function the better the informational capture.

Various attempts have been made to explore the information processing capabilities of DNNs. [72] demonstrated that DNNs can be quantified by the MI between layers. They show with training, feature representations become increasingly compressed towards the bounds of minimally sufficient statistics 1.1.5, effectively mapping the input variables into a representation which preserves the maximal amount of information for exact reconstruction on a lower-dimensional plane for the output y. MDPs follow DPI such that deeper DNN layers will have access to less information than earlier layers. One way to imagine this is that as x passes through each hidden DNN layer, the relevant feature representations from x are decoupled from unintended observation noise. We can define this relevant feature information

as a function of SNR, or by the MI through the assumption of statistical dependence between x and y. MI can effectively be bound by the prediction error. Maximizing MI for DNNs is a natural optimization goal as the more information that is shared between the input data x_0 and output prediction $\hat{y}$, the stronger the prediction will be.

4 CASE STUDY I: INFORMATION-IMBALANCED DATA SETS

Given our prior discussion of DDPMs as a form of AWGN channels 1.1.6 characterized by the ELBO 1.3.2, we seek to explore the effects of ablating various parameters inherent to the introduction and removal of noise, particularly T and β. We study the information theoretic quantities responsible for the reconstruction capabilities of DDPM through MI and raise questions regarding the optimal diffusion training procedures. Specifically, we observe an interesting phenomenon where DDPMs are disproportionately likely to reconstruct particular classes across training epochs, noise schedulers, timesteps, and datasets. The transformation of latent variables to noise, over the course of the DDPM process, should result in uniform class reconstruction when sufficient information is destroyed. Instead, we observe particular classes to be consistently more likely to be generated despite the fact these latent variables become noise.

4.1 Experimental Settings and Design

4.1.1 Neural Networks Settings

Our denoising neural network is a U-Net similar to hat proposed in [7]. We define our model parameters by the following fixed points. Model channels = 32, Residual Blocks = 1, Dropout = 0, Convolutional Resampling = True, Attention Heads = 2, and FreeU = False. The remaining model parameters are dynamic and dependant on the complexity of the dataset. They are comprised on Input/Output Channel, Attention Resolution, Channel Multiplier.

Model channels define the feature maps in the convolution layers and is set to a power of 2 in order to enable smooth down sand up sampling while also maintaining efficiency. Residual blocks [78] amount to skip connections to transfer high level feature representations to deeper network levels. Dropout [79] is a regularization technique which focuses on setting individual random neurons to 0 to stop propagation of signals to deeper layers and encourage robust feature representations. Here we set it to 0 so the model has full capacity. Convolutional resampling is a technique which uses strided convolutions for downsampling and transposed

convolutions for upsampling. Rather than relying pooling techniques convolutional upsampling introduces dynamic model adaptability for learning the downsampled and upsampled representations. Attention is part of the transformer architecture [80] and is utilized to ensure the model can learn spatial relationships well by learning where to focus a models ”attention” in an image. FreeU [81], although not utilized, was added to the model to improve its semantic capabilities. FreeU improves model generation because skip connections tend to focus on passing high-level feature representations through the residual blocks, by reweighing the contributions by the U-Net skip connections we can develop better semantic representations.

The dynamic model parameters include the input and output channels which signify the number of channels in the dataset. For example MNIST is gray-scaled and single channel, while CIFAR10 is RGB with three channels. The attention resolution defines the scale for the attention heads to operate. We dynamically adjust this depending on the dimensions of the dataset we train on, where high dimensional data like CELEBA is set to $(8,16,32)$ resolutions while MNIST and CIFAR10 is set to $(8,16)$ resolutions which define the feature maps of the attention mechanism. Similarly, channel multiplier is also scaled according the image size of the data set where MNIST is set to $(1,2)$, CIFAR10 is set to $(1,2,4)$ and CELEBA is set to $(1,2,3,4)$.

The model parameter choices were determined by hand tuning the model for efficiency and speed. As previously mentioned, deeper models perform better due to the successive chaining of layers resulting in better feature extraction. The trade-off is the computational requirements to train models these models become longer with each additional operation. Each model is trained on a single NVIDIA 4090 and utilizes a batch size of 512 and the standard Adam optimizer [82] with a learning rate of 2×10^{-4} and random perturbations of 1×10^{-8}. Unfortunately, computational complexity, limited resources, and limited time prevented further exploration. Despite this, with small changes, our code base can be adapted to test various other points of interest.

To estimate MI we utilize MINE. MI, for MNIST data, is calculated to be close to the information entropy of $log_2 10 = 3.32$ where 10 defines the number of class labels. We test MINE against these bounds and find that with additional data augmentation techniques we can more closely reach the theoretical upper bound of their entropy. This was necessary to be able to calculate the incremental differences in MI through time. To calculate MI during the DDPM trajectories, we ablate MINE and stack MNIST training data along the channel dimension. This effectively multiplies the expected channel count by 2 of our batched input tensor. This provides a single forward pass through which MINE can calculate the MI of two images at once.

For RGB and Grey-scaled image data MINE uses a simple CNN from a batch size of 512, a learning rate of 0.0001, and train for $150,000$ epochs. Similarly, we train a Gaussian MINE implementation through a simple feed forward MLP which dynamically changes dimensions to fit the Gaussian data dimensions. During training the class label is inserted with the respective training data and processed in the forward pass. An expected moving average is used to ensure the model is predicting MI accurately. Despite rapid training, epochs are kept to $150,000$ as training longer demonstrated divergent and degenerate model behavior.

4.1.2 DDPM Settings

To explore the information processing capabilities of DDPMs we design and test our own implementation. Unless otherwise stated, our control parameters follows those set in [7] which defines timesteps $= 1000$, linear scheduling $= [10^{-4}, 0.02]$, MSE as the objective function, and use models trained to 1000 epochs. Our implementation leaves significant room for testing across various datasets, parameters, and model architectures.

We train various DDPMs with different noise schedulers β including sigmoidal, cosine, and linear. The first two introduce gradual noise at a pace similar to that of the linear scheduler, but in the final steps of T begin introducing significant noise rapidly effectively transforming the latent into a Gaussian distribution. The linear scheduler does not introduce significantly

more noise in the final steps. Given these DDPMs are trained on various noise schedulers we also train each of them across a range of timesteps consisting of $[100, ..., 1000]$ in increments of 100 steps and train for timesteps beyond 1000 at $[1500, 2000, 3000, 4000, 5000]$. Each of these models consisting of variations on noise schedulers and timesteps are trained to 1000 epochs with periodic checkpoints at epochs $[1, 5, 10, 25, 50, 100, 250, 500, 750, 1000]$. In this way, we can observe the effects of T, β, and epochs on the regenerative capabilities of diffusion and particularly the evolution of MI through time.

4.1.3 Experimental Design

Through the various models across T and β we generate samples across epochs which illustrate the input data, the evolution MI over the forward process, the final corrupted data, the evolution of MI over the reverse process, and the final reconstruction generated. Through this process, we can explore the effects of these parameter changes through time and training.

Similarly, for each of the potential combinations of T, β, and epochs we generate input and output pairs of data to compare diffusion reconstructions with their training data inputs. In other words, for every data input we pass through these various DDPM models, we save the input and the generated novel sample. We save 4000 input-output pairs for each class label within the training data, consisting of MNIST and Fashion-MNIST data sets. We utilize our robust classifier models for each data set and explore the reconstruction accuracy of DDPM in generating same input class as its output.

4.2 MI Recovery Results

Through our exploration in studying the information processing capabilities of diffusion processes across a range of ablated parameters, we apply MINE at each timestep in DDPM. In order to ensure MINE predictions are strong, we average the predicted MI over 5 training runs. We explored predicting MI for single data points and batched data points. MI predictions for our work are aggregated in batches of 512 to better model the MI over time.

In all instances of the following figures: Fig. 10, Fig. 11 we define the axis as such: The x-axis in the following individual graphs represents the number of timesteps utilized. The y-axis of the individual plots defines the MI. The plots are also defined by a blue and red lines. The blue bounds signify the variation in MI prediction of the same batched data that is processed 5 times. The red line signifies the average of these variations. The figure is then split into a two by two grid defined by the y-axis which shows the timesteps of the figures within the rows and the x-axis which shows the epochs by the column.

The bounds of MI remain consistently tight across the forward process. Particularly, as the timesteps are increased the MI that is removed due to noise at a given timestep is reduced. We can see this by the differences from the top row, which set timesteps to 100, and the bottom row, which set timesteps to 1000. In the first moments of these graphs, the information decay is significantly less for 1000 timesteps due to the spread of removing information increasing. However, this is to be expected as the forward process is an algorithmic corruption.

In the subsequent reverse process, the predicted MI is significantly different. After sufficiently training a DDPM model, one would expect to find that the MI removed from the forward process would be approximately reconstructed in the reverse process. This is exactly what we find when timesteps are considerable low as the MSS for reconstructions are recoverable since the latent variable at x_T from the forward process is not sufficiently corrupted to remove information present from the input. We can see that during the first epoch of training DDPM fails to extract MI. However, through training and when timesteps are low, the MI extracted increases over the course of the reverse process. When timesteps are increased the MI extraction is significantly reduced. As we see in the row where timesteps are 1000, the MI is significantly more difficult to recover.

During this training process, by testing various ablations to the DDPM parameter settings we find that increasing the training epochs increases the MI extracted similar to Tishby's findings [30]. MI extraction in this instance is tied both to the noise scheduling process, the

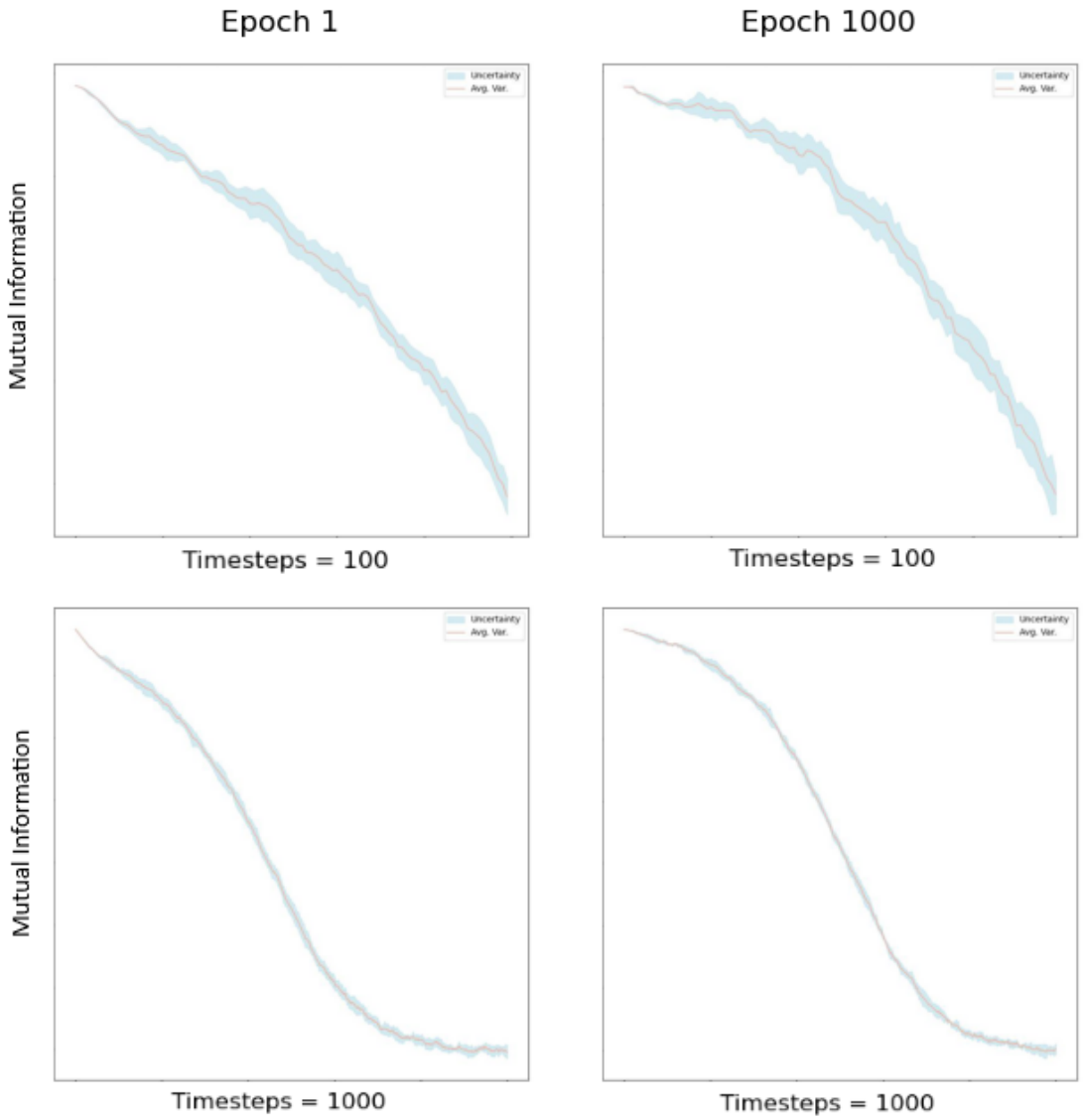

Fig. 10. As we pass through the forward process noise perturbations are gradually added to the input data. Within each chart, the y-axis defines the MI, while the x-axis defines the timestep count. The rows define the timesteps utilized while the columns define the epochs. The forward process is consistent even across any potential variation in predicted MI and are consistent across timesteps according to the scheduler.

number of timesteps utilized, and the number of epochs a model is trained for. We illustrate these findings across a range of timesteps below to better illustrate the effects of timesteps on MI recovery when the noise scheduler and epochs are consistent.

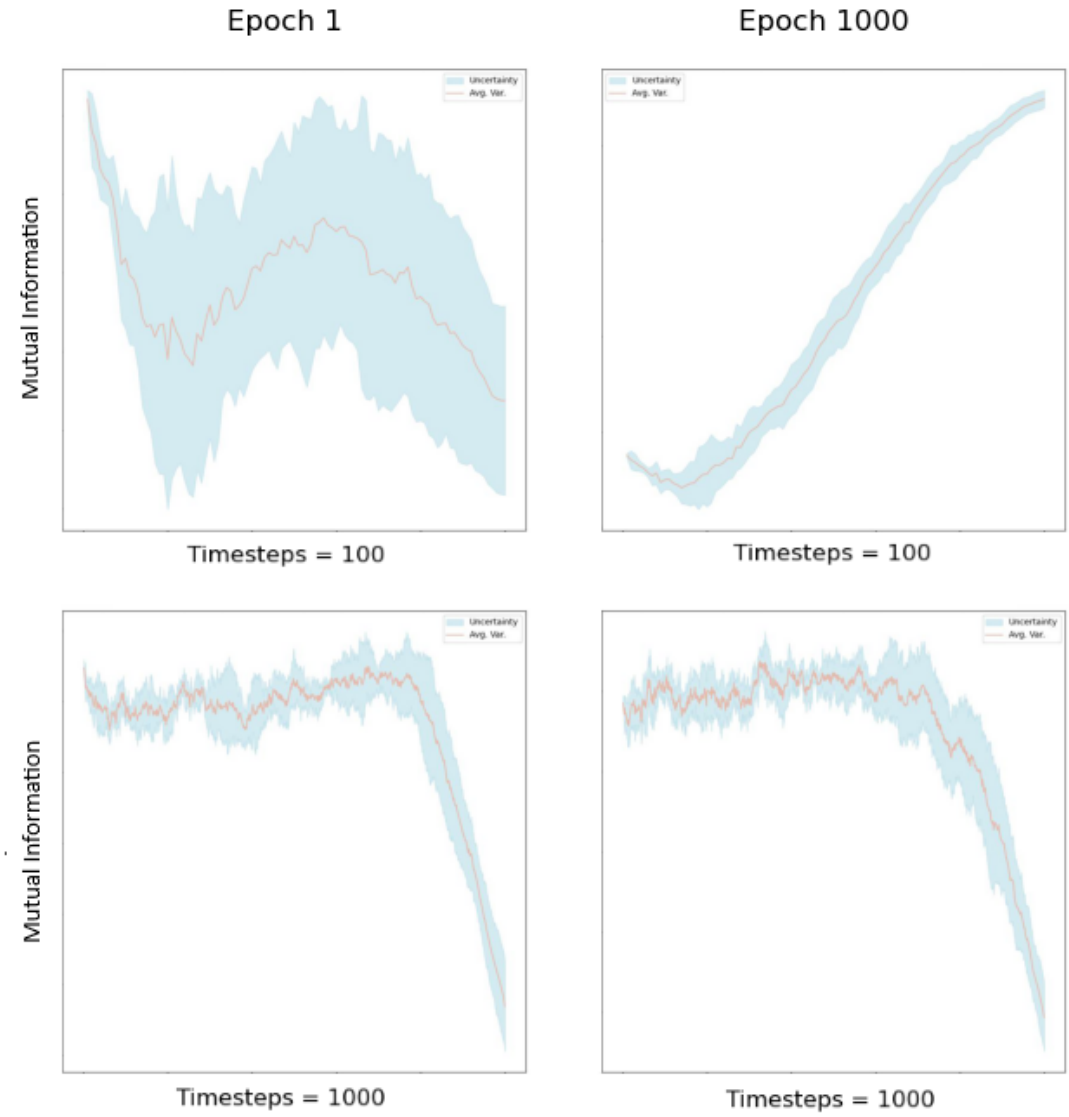

Fig. 11. Similar to 10, we utilize the same grid structure and axis, but demonstrate the MI transformation over the course of the reverse process. We see that as timesteps are increased the extraction of MI becomes more challenging to recover, even as the model is trained for 1000 epochs.

In the following figures, Fig. 12 and Fig. 13, the sequential images define the input data x_0, MI in the forward process, the corrupted latent x_T from the forward process which is then

passed through the reverse process, the MI in the reverse process, and the generated reconstruction $\hat{x}_0$.

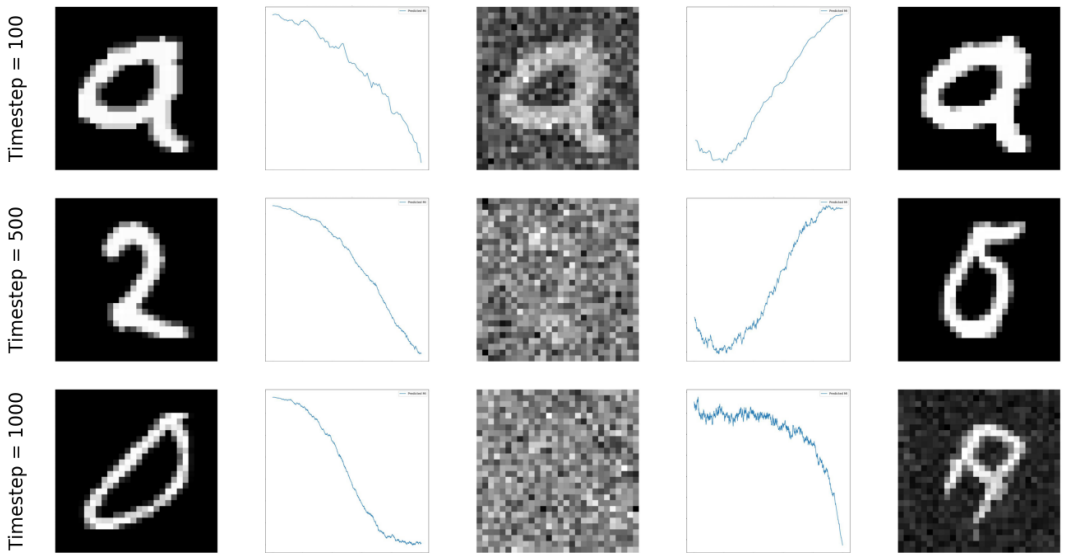

Fig. 12. As data traverses the DDPM process MI gradually decreases with each timestep in the forward process. In the reverse process MI is extracted. When timesteps are low the MI recovery of the latent variable x_t increases. As previously demonstrated in 11, as timesteps are increased MI recovery becomes more difficult.

We observe between 500 and 1000 timesteps, MI recovery fails to become possible. However, we notice through training various models that when model training is consistently kept to 1000 epochs, at approximately 600 timesteps we can witness and obvious shift in MI recovery over the course of training 13. Through training, the MI recovery becomes increasingly better.

4.3 Information-Imbalanced Data Set Results

The results of the prior section 4.2 demonstrate a propagation of information through DDPM in the form of improving MI. Given DDPMs information propagation across its various MDPs and its ability to approximately generate exact reconstructions of input data when timesteps are low, we explore DDPMs class-based reconstruction accuracy across these various models with different parameters.

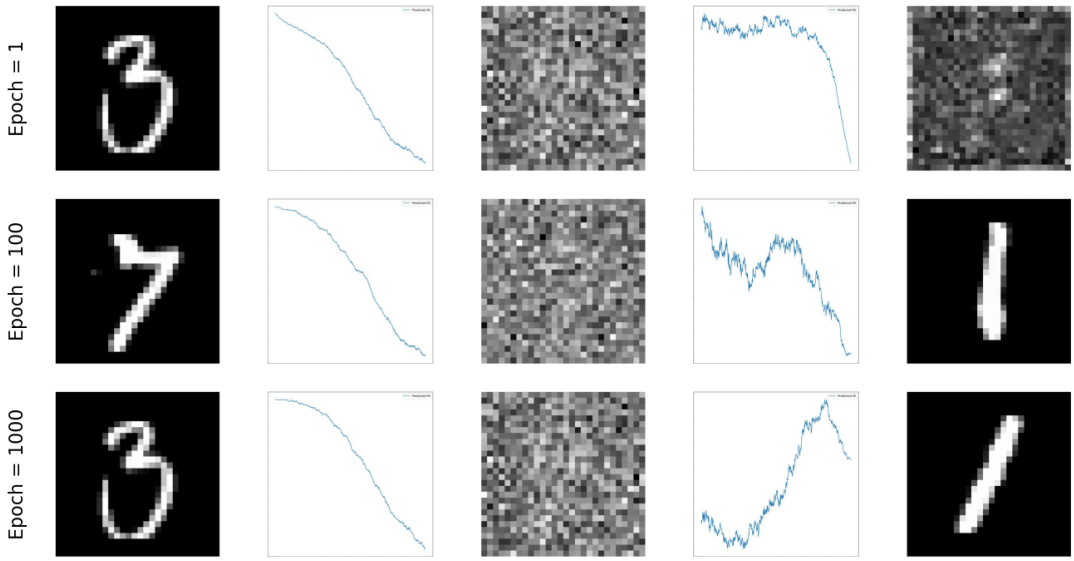

Fig. 13. We study the effects of training DDPM by sampling the MI at different epochs. We observe during training between epochs 100 and 1000 the MI becomes recoverable. As we train the model, the MI extracted from the DDPM latent variables increase.

Recall that information is gradually destroyed as the latent variable is turned into a corrupted Gaussian data sample. As x_0 approaches x_T it becomes increasingly Gaussian and replaces the signals of information with noise. This asserts that recovering the information which ascribe the information of a particular class should become increasingly difficult to extract. During reconstruction we would then expect to see that classes are uniformly generated according to the number of classes in the data set. Particularly, in MNIST and Fashion-MNIST there are 10 distinct classes. This would lead to the expectation that each class has a 10% probability of being reconstructed. Instead, we observe that some classes like label 8 in MNIST are significantly more likely to be reconstructed at a rate of $20\% - 40\%$ across timesteps, schedulers, and epochs. We attribute this finding to an information-imbalance since these data sets contain equal quantities of labelled data for each class to the tune of 6000 samples per label. We demonstrate these observations below.

For context, a robust classification model was trained on each individual data set. This classifier then predicts the input data class and the reconstruction data class. From these predicted classifications, we compare the model predictions with the true labels of the input and novel samples. We graph the prediction accuracy compared to the true accuracy in the graphs, where the input data accuracy is displayed in blue and the novel sample data accuracy is displayed in red. Our results can be seen in Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18, and Fig. 19.

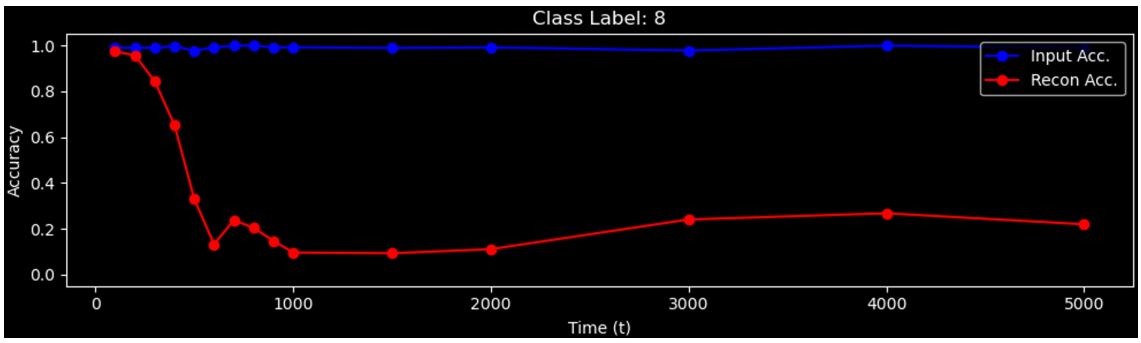

Fig. 14. The reconstruction accuracy for the MNIST data set on class label 8 with a linear DDPM noise scheduler and trained for 1000 epochs. After the initial degradation of accuracy from the increasing noise perturbations added by increasing the timesteps, we find that DDPM disproportionately generates novel samples for timesteps past 2000.

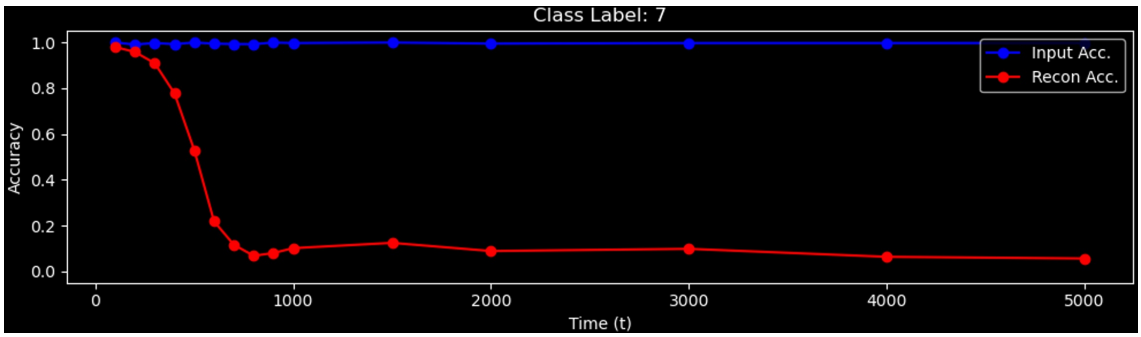

Fig. 15. The reconstruction accuracy for the MNIST data set on class label 7 with a linear DDPM noise scheduler and trained for 1000 epochs.

Across epochs, timesteps, and noise schedulers some classes like more likely to be reconstructed. Fig. 14, Fig. 16, and Fig. 18 all demonstrate imbalanced data reconstructions

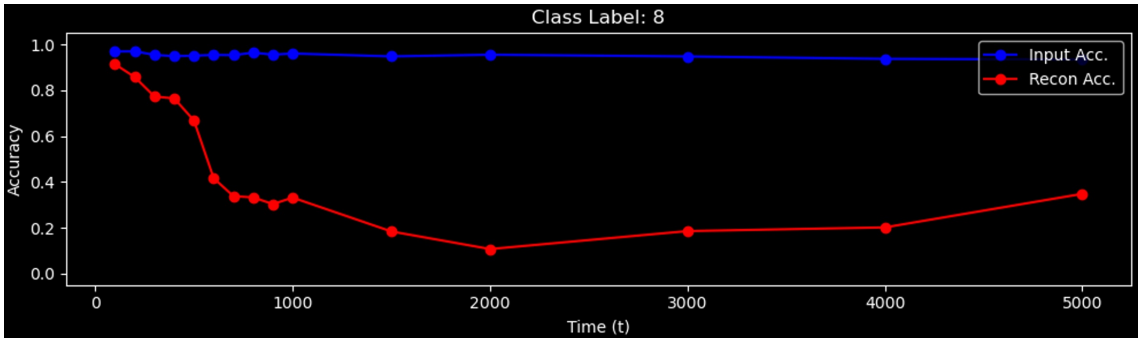

Fig. 16. The reconstruction accuracy for the Fashion-MNIST data set on class number 8 with a linear DDPM noise scheduler and trained for 1000 epochs.

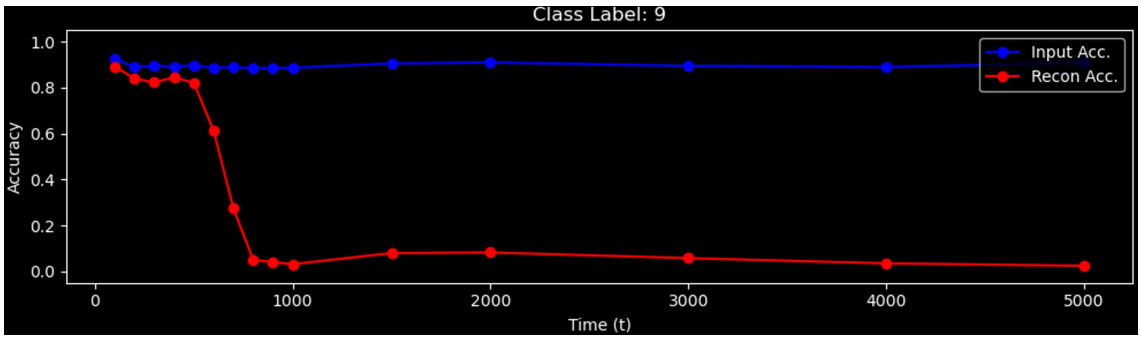

Fig. 17. The reconstruction accuracy for the Fashion-MNIST data set on class number 9 with a linear DDPM noise scheduler and trained for 1000 epochs.

which are above the expected accuracy of 10%. Conversely, Fig. 15, Fig. 17, and Fig. 19 all constitute the expected accuracy. Fig. 16 also exaggerates the notion that different timesteps

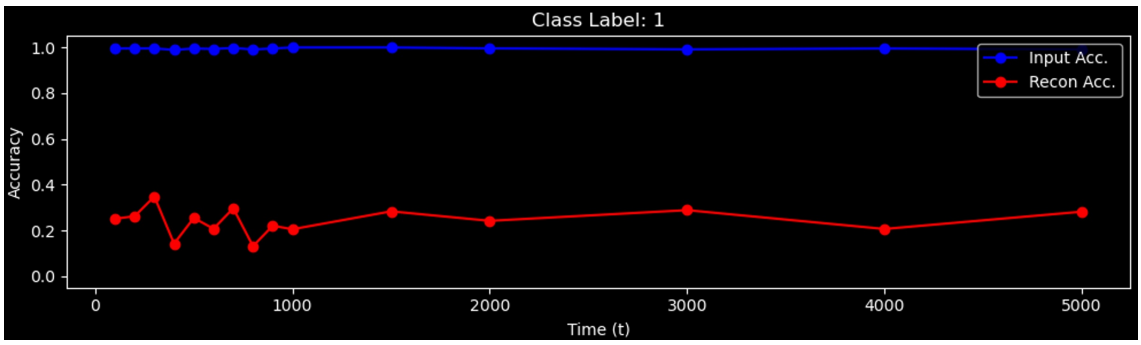

Fig. 18. The reconstruction accuracy for the MNIST data set on class label 1 with a linear DDPM noise scheduler and trained for 10 epochs.

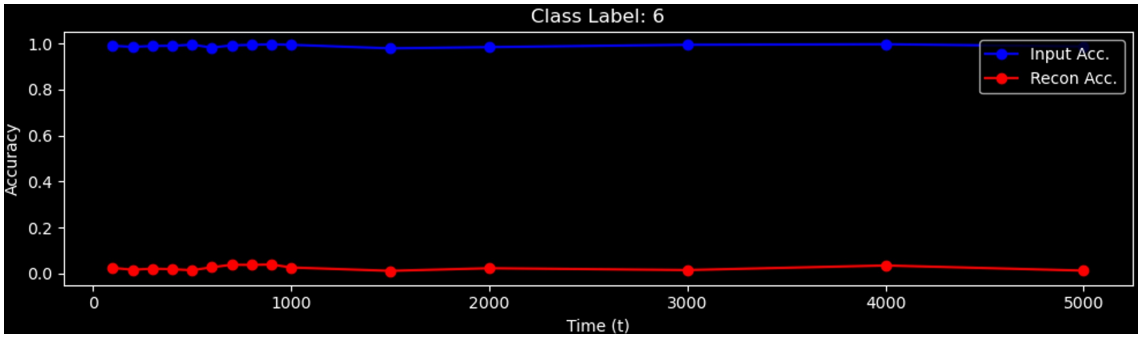

Fig. 19. The reconstruction accuracy for the MNIST data set on class label 6 with a linear DDPM noise scheduler and trained for 10 epochs.

effect the generative capabilities of each class differently. Notice that as our prediction accuracy reduces to a minimum at timestep 2000 the accuracy of predicting class label 8 increases as we utilize more timesteps. This demonstrates that selecting timesteps may have an effect on what classes are likely to be reconstructed. Additionally, consider that when utilizing the cosine scheduler a significant amount of noise is added in the final timesteps virtually replacing the input image with pure noise. Despite this, DDPM continues to reconstruct particular classes even though the corrupted latent should contain no information about the input data as shown in Fig. 18 and Fig. 19. Moreover, we observe these phenomenon across data sets as well since we can illustrate this accuracy imbalance in the Fashion-MNIST data set as seen in Fig. 16 and Fig. 17. These results are consistent across epochs, timesteps, and noise schedulers.

4.4 Discussion

We postulate the imbalance in regenerated accuracy as a product of the information from the input data. Classes with higher information content would likely be harder to destroy, particularly in regard to the MSS. For this reason we suspect that there are a few ways to combat this phenomenon. The first we consider is the maximize the entropy of each data point, or image, in the data set such that all the training data is maximally equal. This would ensure that the information contained in any one data point would be uniform and thus likely to have

equal reconstruction potential. Another potential route is to utilize our prior findings regarding training diffusion models for specific timesteps based on the individual classes. We believe that some classes will take more or less time to diffuse properly and by training to this points we can control the class based reconstruction capabilities. The final method we propose would be to ablate the objective function in such a way that one could guide DDPM towards a desired class reconstruction. This could potentially be done by utilizing an intermediate classifier model which penalizes the objective function if the information in the latent data is not predicted to be the desired class. These potential solutions offer approaches to help guide the diffusion process towards uniform class reconstructions, but are left as open research problems to explore.

5 CASE STUDY II: A NOVEL RECOMMENDER MODULE SCORE FUNCTION

Recommendation systems have become a crucial component of Web systems. In a paper I helped publish to IEEE Access [83], we introduce a recommender module which utilizes a score-based DDPM approach to generate novel user-item interactions which can be used to partially augment or fully synthesize data sets. We call this approach a Score-based Diffusion Recommender Module (SDRM). SDRM achieves an average boost of 4.3% on partial augmentation of data sets and 4.6% for fully synthetic data. These boosts consist of training recommender systems utilizing the data we generated through SDRM. Simultaneously, our generated data is 99% dissimilar to the training data. This is important because training recommender models on user data raises many privacy concerns.

In SDRM, my main contribution was a novel score function inspired by denoising score matching which helps SDRM achieve its state-of-the-art results, surpassing previous attempts at generating novel user data. In the following section we provide context for the different components of SDRM and explore the novel score function.

5.1 Background

5.1.1 Variational Autoencoders

VAEs are class of generative models where the input data dimensions changes through time. A probabilistic encoder compresses the input data x according to the mean and standard deviation of x into a lower dimensional latent vector. This latent vector is then passed through a probabilistic decoder which samples from the latent and produces a reconstruction of x'. The goal of the VAE is to train the encoder and decoder networks in such a way that x' is a similar reconstruction of x. VAEs optimize the ELBO which is generally intractable. This normally requires a variational posterior, but the probabilistic encoder and decoder jointly optimize the data samples. In effect, the encoder maps to the latent space and the decoder maps from the latent space. The variation in sampling leads to novel reconstructions or ones which are similar to the input data. Optimization of VAEs amounts to reducing the reconstruction error

according the KL divergence. [5] In SDRM, we utilize VAEs as a means of tightly compressing sparse data entries, thus utilizing their primary components in mapping data to a lower dimensional latent space.

5.1.2 Recommender Systems

As previously described 1.7, recommender systems are algorithms which filter content to provide a more personalized experience to users. Machine learning approaches are an ideal setting for recommender systems since they can search for hidden patterns within user data to better serve content. Methods like collaborative filtering and content-based filtering have found significant success. Simpler approaches like user-item interactions can also yield strong penalization, but have suffered from data sparsity. User-item interactions are large matrices which consist of data for a single user. The user will interact with particular items which is transcribed as a data entry within these large item matrices. These matrices are often incredibly sparse leading to challenges in modeling data. In SDRM, we utilize user-item interaction data sets and address the challenges posed by data sparsity through the utilization of generative modeling and training dynamics.

5.1.3 Score Functions

As we previously described in 1.8, score functions are a class of objective functions. Score matching is a way to approximate the score at a given point on the gradient plane determined by its steepness, and minimizes the expected squared error of the data. Generative models are capable of utilizing score-based objectives as an alternative to maximizing the ELBO, and are particularly useful as an unbiased objective. [29]. However, when data is sparse or ground truth data is unavailable, score functions like a score matching objective can fail to model data. In SDRM, we train recommender systems on ground truth data and are able to apply score-based approaches to training SDRMs architecture due to the nature of recommender system user-item data.

5.2 Score-based Diffusion Recommender Module

SDRM is "... a module designed to generate artificial user-item preferences to augment or replace the data set it is trained on" [83]. SDRM is a strong application of diffusion models for the task of modeling complex training data. SDRMs architecture is illustrated in Fig. 20 Often, user-item preferences are sparse matrices which are difficult to model. Utilizing a VAE architecture SDRM first compresses these user-item preferences into a lower-dimensional Gaussian distribution. The encoder network is pretrained to map training data x into a compressed latent vector z_0. This latent is then processed by a score-based diffusion model utilizing a multi-layer perceptron denoising model. This diffusion reconstruction gradually corrupts the compressed latent VAE output z_0 into a noisy Gaussian sample z_T. This sample is then reconstructed towards a Gaussian distribution modeled by the VAE encoder output $\hat{z}_0$. Finally, this novel Gaussian reconstruction is processed by the decoder network back into a novel reconstruction of the same dimensions as the input to the VAE encoder $\hat{x}$. Training diffusion to corrupt a Gaussian sample and reconstruct it into a Gaussian sample allows the diffusion model to learn finer details of its input. Below we illustrate this module.

Unlike prior approaches [84], [85] which utilize at most 10 timesteps for the diffusion process, SDRM utilizes up to 200 timesteps to allow diffusion to better learn the underlying data distribution. We propose two approaches called full-resolution sampling (F-SDRM) and multi-resolution sampling (M-SDRM). F-SDRM trains for the full 200 steps while M-SDRM trains for a random number of timesteps $t \in T$. This provides a more robust sampling space for SDRM to learn the details of the data distribution.

5.3 A Novel Score Function for SDRM

My primary contribution to this work consists of a novel score function to train the score-based diffusion model with the VAE encoder and was inspired by denoising score matching. Score-based training objectives are ideal for the setting of recommender systems.

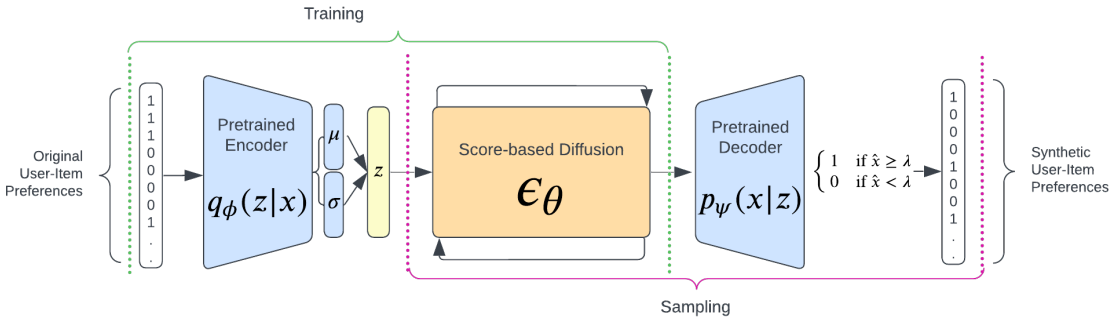

Fig. 20. The SDRM architecture including training and sampling methodologies. Importantly, this module pretrains the encoder and decoder networks together. Then, we train the VAE encoder with the score-based diffusion model. We apply a novel heuristic score function to this module and achieve strong performance on generating data and using this data to train recommender systems. Figure from [83].

Not only is ground truth training data available for these models, but score functions are naturally unbiased and can be a replacement to optimizing the ELBO.

Denoising score matching seeks to estimate the gradient of the log probability density function of the data distribution. In denoising score matching, the model is trained to predict the added noise to the input data. Through this training techniques, the model learns to estimate the score function of the data distribution and particularly optimizes movement on the gradient of the log probability density function towards what could be considered a ”cleaner” sample of the input data which does not contain the added noise.

Our proposed objective function for SDRM utilizes various aspects of the latent variable encoded by the VAE and the diffusion model latent to our advantage. The proposed score-based objective function can be seen below.

$$\ell(z_t) = \frac{||\big(s_\theta(\hat{z}_t) - s_\theta(z_t)\big) - \Delta_\theta(z_t)||_2^2 + ||s_\theta(z_t) - \Delta_\theta(z_t)||_2^2}{||\Delta_\theta(z_t)||_2^2} \tag{81}$$

z_t is the encoder networks latent representation output. $\hat{z}_t$ represents a perturbed version of the latent variable where noise is added according to $\mathcal{N}(0, \sigma_v \mathbf{I})$. s_θ is the score function of the diffusion model according to the parameters on θ and $\Delta_\theta(z_t)$ is the prediction noise at a given timestep t.

This objective function comprises three distinct components. The first component, represented by $\big((s_\theta(\hat{z}_t) - s_\theta(z_t)\big) - \Delta_\theta(z_t)\big)$, quantifies and penalizes discrepancies which arise from the random perturbations to the noisy data on the regenerated diffusion latent $\hat{z}_t$ when compared to $\Delta_\theta(z_t)$. The second component, $\big(s_\theta(z_t) - \Delta_\theta(z_t)\big)$, measures changes between the ground truth score function and the approximated noise deviations generated by the VAEs latent encoding process effectively measuring the accuracy and effectiveness of the encoder network. The final component, $\Delta_\theta(z_t)$, is a denominator which acts as a normalization term to stabilize the function. This ensures the loss function does not incur numerical instabilities during the training process. We show this score-based objective is empirically more effective than other objective functions which have been applied to recommender diffusion models, including [85] and [84].

5.4 Experimental Settings and Design

In our experiments we explore SDRMs effectiveness utilizing four publicly available data sets. MovieLens 1M (ML-1M) [86], MovieLens 100k (ML-100k) [86], Amazon Digital Music (ADM) [87], and Amazon Luxury Beauty (ALB) [87]. Each data sets comprises a wide range of diverse user-item ratings and contain varying levels of data sparsity. We also remove users who have not rated 5 items and items which do not have 5 ratings. Table 1 illustrates our data settings succinctly.

We then evaluate our generated data using a few well known metrics in recommender settings, Recall@k and NDCG@k, which utilize the top-k metric. These effectively measure the top@k ranking and its similarity to the ground truth data. We also evaluate our model both in the augmented data set and fully synthetic data set setting. Notably, the augmented and fully

Table 1
The statistics for each data set used to train SDRM. Table from [83].

Dataset	#Users	#Items	#Ratings	Sparsity
Amazon Luxury Beauty	1,344	729	15,359	98.43%
Amazon Digital Music	10,621	8,582	108,509	99.88%
MovieLens 100k	938	1,008	95,215	89.93%
MovieLens 1M	3,125	6,034	994,338	94.73%

synthetic data set comprise at least 20% of real user data. "This approach is necessary to show how synthetic data can enhance the predictive performance on real users, as it is impractical to measure the effectiveness of synthetic data in a recommendation model without incorporating some actual user data" [83]. We then apply these generated data sets to three recommendation algorithms, namely SVD [88], MLP [89], and NeuMF [90], to test the effectiveness of our generated data. We then compare our F-SDRM and M-SDRM models against other competitive models which aim to generate user-item data, known as CTGAN [91], TVAE [92], CODIGEM [85], DiffRec [84], MultiVAE [93], and MultiVAE++ which we use as our pre-trained multi-nomial VAE for the encoder and decoder.

We also conduct a strong hyperparameter sweep across many variables for the various components of SDRM. This includes the VAE architecture and its parameters as well as the score-based diffusion models parameters like timesteps, learning rate and batch size to name a few.

5.5 SDRM Results

We compare SDRM against other generative recommender models and select various top-k according to $k \in (1,3,6,10,20,50)$. We present Recall@k and NDCG@k over five runs with its average score and standard deviation. In Table 2 and Table 3 we present our results.

We demonstrate an average improvement of $\sim 4.5\%$ in both Recall@k and NDCG@k, with some instances reaching a maximum of 24.28% and 21.87% respectively. A significant improvement over prior approaches. Additionally, through the use of the Jaccard similarity

Table 2
Overall performance comparison between baselines by training with synthetic and the original dataset. The best results are in bold and the second best are underlined. Average overall improvement: Recall@10 4.48%, NDCG@10 5.07%. Prior caption and figure below from [83].

Model	SVD							
Dataset	ALB		ML-100k		ML-1M		ADM	
Baseline	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
Original	0.3113	0.287	0.3716	0.3973	**0.3769**	**0.4035**	0.0624	0.0427
CTGAN	0.2642	0.2407	0.3533	0.3813	0.3652	0.3917	0.0405	0.0267
TVAE	0.3113	0.2873	0.3668	0.3918	0.3638	0.3869	0.0624	0.0427
CODIGEM	0.3025	0.2819	0.3492	0.3745	0.3113	0.3336	0.049	0.0301
DiffRec	0.2936	0.2748	0.37	0.4021	0.3598	0.383	0.0306	0.0228
MultiVAE	0.3226	**0.2998**	0.3822	0.4103	0.3391	0.3658	0.0612	0.0389
MultiVAE++	0.3163	0.2912	0.3878	0.4126	0.3717	0.3946	0.0613	0.0406
F-SDRM	**0.325**	0.2988	0.3924	**0.4181**	0.3722	0.3977	0.0623	0.041
M-SDRM	0.3249	0.2957	**0.3971**	0.417	0.372	0.3962	**0.0641**	**0.0442**
Improvement	**0.74 %**	**-0.33 %**	**2.39 %**	**1.33 %**	**-1.26 %**	**-1.44 %**	**2.72 %**	**3.51 %**
	MLP							
Original	0.3183	0.3007	0.3569	0.3788	0.3319	0.3558	0.0194	0.0128
CTGAN	0.3188	0.2987	0.3567	0.3781	0.3228	0.3463	0.0208	0.0135
TVAE	0.2438	0.2431	0.3437	0.3622	0.2949	0.3186	0.0184	0.012
CODIGEM	**0.3365**	**0.3071**	0.3479	0.3686	0.3132	0.3377	0.0596	0.0385
DiffRec	0.3283	0.3013	0.355	0.3769	0.3246	0.3482	0.0213	0.0139
MultiVAE	0.3328	0.3048	0.053	0.0615	0.3588	0.372	0.0199	0.0137
MultiVAE++	0.3316	0.3	0.3901	0.4101	0.3528	0.3763	0.0756	0.0489
F-SDRM	0.3246	0.3004	0.3839	0.4055	**0.3595**	**0.3845**	0.0146	0.0095
M-SDRM	0.3343	0.303	**0.3947**	**0.419**	0.3591	0.384	**0.0798**	**0.0523**
Improvement	**-3.66 %**	**-1.35 %**	**1.18 %**	**2.17 %**	**0.19 %**	**2.18 %**	**5.55 %**	**6.95 %**
	NeuMF							
Original	0.2399	0.0818	0.1006	0.096	0.0113	0.0094	0.0057	0.0011
CTGAN	0.2444	0.0877	0.0889	0.086	0.0219	0.0192	0.007	0.0014
TVAE	0.2439	0.0874	0.1258	0.1282	0.0126	0.0105	0.0053	0.0013
CODIGEM	0.1709	0.0477	0.1015	0.0839	0.0642	0.0601	0.007	0.0014
DiffRec	0.2621	0.0923	0.1147	0.1093	0.0203	0.0178	0.0138	0.0035
MultiVAE	0.2475	0.0826	0.1253	0.118	0.0487	0.0483	0.0154	0.0036
MultiVAE++	0.2687	0.0877	0.2357	0.1858	0.1046	0.0972	0.0224	0.005
F-SDRM	**0.3225**	**0.109**	0.232	**0.1891**	0.1026	0.0935	0.0234	0.0054
M-SDRM	0.2953	0.1026	**0.2425**	0.186	**0.1059**	**0.099**	**0.0273**	**0.006**
Improvement	**20.02 %**	**24.28 %**	**2.88 %**	**1.77 %**	**1.24 %**	**1.85 %**	**21.87 %**	**20 %**

metric we demonstrate that our synthetic data in both M-SDRM and F-SDRM achieves over 99% dissimilarity which effectively protects against user privacy concerns.

5.6 Discussion

We believe our approach generates better recommendation data sets because of the score-based diffusion model which models the VAE's prior distribution on $p_\theta(z)$. This process can improve the resolution of the noise processing before utilizing the decoder network to decompress the latent variable. Utilizing all timesteps *t* according to a uniform selection process also allows for a better model since M-SDRM is better at generating data that F-SDRM. Our approach directly utilizes many strengths and turns weakness into advantages

Table 3
Overall performance comparison between baselines by training with synthetic data. The best results are in bold and the second best are underlined. Average overall improvement: Recall@10 2.08%, NDCG@10 0.88%. Prior caption and figure below from [83].

Model	SVD							
Dataset	ALB		ML-100k		ML-1M		ADM	
Baseline	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
Original	0.3113	0.287	0.3716	0.3973	**0.3769**	**0.4035**	0.0624	0.0427
CTGAN	0.0654	0.056	0.2215 1	0.225	0.2623	0.2769	0.0152	0.0095
TVAE	0.2521	0.2503	0.2641	0.2766	0.3082	0.325	0.0214	0.0143
CODIGEM	0.3044	0.2813	0.2799	0.286	0.2624	0.2774	0.0481	0.0295
DiffRec	0.2702	0.2541	0.3476	0.3624	0.3354	0.3595	0.0258	0.0197
MultiVAE	0.3299	0.3063	0.3724	0.3937	0.3134	0.3359	0.0393	0.0246
MultiVAE++	0.3406	0.3188	0.3889	0.414	0.3704	0.3935	0.0617	0.0412
F-SDRM	**0.3471**	**0.3229**	0.3931	**0.4176**	0.3707	0.3944	**0.0628**	**0.0428**
M-SDRM	0.3384	0.3174	**0.3946**	0.4176	0.3703	0.3946	0.0622	0.0423
Improvement	**1.87%**	**1.27%**	**1.45%**	**0.86%**	**-1.65%**	**-2.21%**	**0.64%**	**0.23%**
Model	MLP							
Original	0.3183	0.3007	0.3569	**0.3788**	0.3319	0.3558	0.0194	0.0128
CTGAN	0.2651	0.2189	0.2017	0.2197	0.2081	0.2216	0.0011	0.0007
TVAE	0.2438	0.2431	0.1334	0.1586	0.1569	0.1674	0.0153	0.0095
CODIGEM	0.1568	0.1012	0.1445	0.1403	0.1882	0.187	0.0249	0.0173
DiffRec	0.3065	0.2947	0.2997	0.3107	0.2349	0.2526	0.0181	0.0115
MultiVAE	0.311	0.2925	0.3359	0.3465	0.3385	0.3633	0.0172	0.012
MultiVAE++	0.3356	0.3079	0.3488	0.3616	0.3409	0.3636	**0.071**	**0.0492**
F-SDRM	0.3375	0.3103	**0.3601**	0.3754	**0.3451**	0.3693	0.013	0.0085
M-SDRM	**0.3474**	**0.3142**	0.3544	0.3755	0.3438	**0.3702**	0.0668	0.0451
Improvement	**3.51 %**	**2.04 %**	**0.89%**	**-0.87 %**	**1.23%**	**1.81 %**	**-6.28 %**	**-9.09%**
Model	NeuMF							
Original	0.2399	0.0818	0.1006	0.096	0.0113	0.0094	0.0057	0.0011
CTGAN	0.1786 3	0.0405	0.0532	0.05	0.0053	0.0055	0.0073	0.0014
TVAE	0.0957	0.0231	0.1289	0.1326	0.0237	0.023	0.009	0.0013
CODIGEM	0.1154	0.0294	0.0782	0.0552	0.0606	0.0574	0.0022	0.0004
DiffRec	0.2613	0.0881	0.0726	0.0629	0.0143	0.0134	0.0168	0.0031
MultiVAE	0.0834	0.0196	0.0951	0.0894	0.0198	0.0195	0.0174	0.0042
MultiVAE++	0.331	**0.1143**	0.2309	0.1853	0.126	0.1154	0.0259	0.0059
F-SDRM	**0.3339**	0.1137	**0.2421**	**0.1913**	0.1265	0.1146	0.028	0.0058
M-SDRM	0.332	0.1141	0.2303	0.1814	**0.1296**	**0.1212**	**0.0297**	**0.0064**
Improvement	**0.87 %**	**-0.17 %**	**4.85 %**	**3.23 %**	**2.85 %**	**5.02 %**	**14.67 %**	**8.47 %**

for the VAE scheme and diffusion model scheme. The high quality and highly flexible nature of diffusion models coupled with a score-based objective allows SDRM to surpass all existing data generation techniques in the recommender settings.

5.7 Limitations and Future Work

There are obvious points of improvement including replacing the multi-layered perceptron with a stronger model. Even clustering data and strong data preprocessing may provide significant improvements to the quality of SDRM. Additionally, our method can only generate a single data point at a time. Batching data would be significantly faster and improve the scaling potential of our model. Additionally, the data sets we tested were small. Exploration

into testing on larger data sets may yield interesting results or demonstrate weaknesses in our approach. We leave further exploration to future work.

5.8 Conclusion

In this work, we introduced SDRM as an approach to generate synthetic data for training recommendation systems. Utilizing a VAE and diffusion framework we successfully captured strong latent representations and were able to apply a double Gaussian filtering through the VAE and diffusion to generated novel samples which were significantly dissimilar to the training data sets. Through our ablation studies we demonstrated SDRM achieved a 4.5% average improvements in Recall@k and NDCG@k with the best improvement being 24.5%. We successfully overcame the challenges of data sparsity, while also helping to develop an approach to improve data set quality through novel sampling. Our studies also bridge a gap in that is often overlooked in prior literature in benchmarking the generated data results. SDRM proves to be a strong module capable of generating synthetic data to partially augment or fully replace recommendation data sets.

6 CONCLUSION

In this thesis we explored the information processing capabilities of diffusion models. Utilizing the tools of information theory, we explore diffusion models. We studied two domains consisting of information-imbalanced data sets and proposed a novel score function for training SDRM. We also illustrated various relations between DDPMs and DNNs through an information theoretic approach. We observed class balanced data sets generating classes of a particular label significantly more than other labels across multiple clean data sets. We proposed an information-theoretic reasoning in that the entropy of these data points are imbalanced and proposed potential solutions to alleviate this problem. Particularly through utilizing the training dynamics of diffusion to balance label reconstruction, maximizing the entropy of each data point, and or utilizing a classification model with ablations to the objective function to guide the diffusion generation processes towards a desired class. We then introduce a novel score function for SDRM with the purpose of generating novel samples which can augment or replace existing data sets to improve training other recommender models. We show our SDRM model to reach significant improvements over prior works at an average improvement of 4.5%.

Literature Cited

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," *ArXiv*, vol. 37, pp. 2256–2265, 07–09 Jul 2015. [Online]. Available: https://proceedings.mlr.press/v37/sohl-dickstein15.html

[2] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *ArXiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[3] S. Lyu, "Unifying non-maximum likelihood learning objectives with minimum kl contraction," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. Red Hook, NY, USA: Curran Associates Inc., 2011, pp. 64–72.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *ACM*, vol. 63, no. 11, pp. 139–144, Oct 2020. [Online]. Available: https://doi.org/10.1145/3422622

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:216078090

[6] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:54482423

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[8] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[9] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves imagenet classification," *ArXiv*, vol. abs/2304.08466, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258179174

[10] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. [Online]. Available: https://ieeexplore.ieee.org/document/6773024

[11] X. Kong, R. Brekelmans, and G. Ver Steeg, "Information-theoretic diffusion," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://arxiv.org/abs/2302.03792

[12] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 531–540. [Online]. Available: https://proceedings.mlr.press/v80/belghazi18a.html

[13] S. Zhao, J. Song, and S. Ermon, "Infovae: Balancing learning and inference in variational autoencoders," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5885–5892, Jul 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4538

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Paper.pdf

[15] F. Fleuret, "Information theory is maybe the only toolbox that sometimes gives you some certainties in this mess," Twitter. Accessed Sept. 19, 2023. [Online.] Available: https://x.com/francoisfleuret/status/1679780036432232448, Jul 2023.

[16] P. Brémaud, *An Introduction to Probabilistic Modeling*, ser. Undergraduate Texts in Mathematics. New York, NY: Springer New York, 1988.

[17] J. V. Stone, *Information Theory: A Tutorial Introduction*. Sebtel Press, 2013.

[18] X. Lu, K. Lee, P. Abbeel, and S. Tiomkin, "Dynamics generalization via information bottleneck in deep reinforcement learning," *ArXiv*, vol. abs/2008.00614, 2020. [Online]. Available: https://arxiv.org/abs/2008.00614

[19] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 222, pp. 309–368, 1922. [Online]. Available: http://www.jstor.org/stable/91208

[20] J. A. T. Thomas M. Cover, "Channel capacity," in *Elements of Information Theory*. John Wiley and Sons, Ltd, 2005, pp. 183–223. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471200611.ch8

[21] Y. Polyanskiy and Y. Wu, *Information Theory: From Coding to Learning*. Cambridge University Press, 2024.

[22] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, D. Rand and L.-S. Young, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 366–381.

[23] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[24] F. Locatello, S. Bauer, M. Lučić, G. Rätsch, S. Gelly, B. Schölkopf, and O. F. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*, 2019. [Online]. Available: http://proceedings.mlr.press/v97/locatello19a.html

[25] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4713 – 4726, 2023.

[26] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 64 971–64 995. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/ccf6d8b4a1fe9d9c8192f00c713872ea-Paper-Conference.pdf

[27] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, and S. Devlin, "Imitating human behaviour with diffusion models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Pv1GPQzRrC8

[28] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 9902–9915. [Online]. Available: https://proceedings.mlr.press/v162/janner22a.html

[29] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=PxTIG12RRHS

[30] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *ArXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1703.00810

[31] A. I. Humayun, R. Balestriero, and R. G. Baraniuk, "Deep networks always grok and here is why," *ArXiv*, vol. abs/2402.15555, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267938432

[32] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *ArXiv*, vol. physics/0004057, 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:8936496

[33] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, vol. 11, pp. 125–139, 1998.

[34] W. Feller, "On the theory of stochastic processes, with particular reference to applications," in *Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1949. [Online]. Available: https://api.semanticscholar.org/CorpusID:121027442

[35] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf

[36] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 21 696–21 707. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/b578f2a52a0229873fefc2a4b06377fa-Paper.pdf

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[38] T. Chen, "On the importance of noise scheduling for diffusion models," *ArXiv*, vol. abs/2301.10972, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256274607

[39] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021, arXiv:2112.10752.

[40] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 32 211–32 252. [Online]. Available: https://proceedings.mlr.press/v202/song23a.html

[41] Z. Liu, R. Feng, K. Zhu, Y. Zhang, K. Zheng, Y. Liu, D. Zhao, J. Zhou, and Y. Cao, "Cones: Concept neurons in diffusion models for customized generation," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 21 548–21 566. [Online]. Available: https://proceedings.mlr.press/v202/liu23j.html

[42] G. Mariani, I. Tallini, E. Postolache, M. Mancusi, L. Cosmo, and E. Rodolà, "Multi-source diffusion models for simultaneous music generation and separation," 2024.

[43] F. Schneider, "ArchiSound: Audio Generation with Diffusion," Jan 2023, arXiv:2301.13267.

[44] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J

[45] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," 2022, arXiv:2204.03458.

[46] S. Yu, K. Sohn, S. Kim, and J. Shin, "Video probabilistic diffusion models in projected latent space," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18 456–18 466.

[47] A. Gupta and A. Gupta, "3dgen: Triplane latent diffusion for textured mesh generation," *ArXiv*, vol. abs/2303.05371, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257427345

[48] P. Zhuang, S. Abnar, J. Gu, A. Schwing, J. M. Susskind, and M. Á. Bautista, "Diffusion probabilistic fields," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=ik91mY-2GN

[49] Y. Ma, H. Yang, W. Wang, J. Fu, and J. Liu, "Unified multi-modal latent diffusion for joint subject and text conditional image generation," 2023, arXiv:2303.09319.

[50] Y. Wang, J. Wang, G. Lu, H. Xu, Z. Li, W. Zhang, and Y. Fu, "Entity-level text-guided image manipulation," *ArXiv*, vol. abs/2302.11383, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257078872

[51] T.-H. Liao, G. Songwei, X. Yiran, Y.-C. Lee, A. Badour, and J.-B. Huang, "Text-driven visual synthesis with latent diffusion prior," 2023, arXiv:2302.08510.

[52] I. Skorokhodov, A. Siarohin, Y. Xu, J. Ren, H.-Y. Lee, P. Wonka, and S. Tulyakov, "3d generation on imagenet," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=U2WjB9xxZ9q

[53] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, and T. Salimans, "Imagen video: High definition video generation with diffusion models," 2022, arXiv:2210.02303.

[54] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=FjNys5c7VyY

[55] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208

[56] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8162–8171. [Online]. Available: https://proceedings.mlr.press/v139/nichol21a.html

[57] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, C. Hallacy, B. Mann, A. Radford, A. Ramesh, N. Ryder, D. M. Ziegler, J. Schulman, D. Amodei, and S. McCandlish, "Scaling laws for autoregressive generative modeling," *ArXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2010.14701

[58] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: Association for Computing Machinery, 1992, pp. 144–152. [Online]. Available: https://doi.org/10.1145/130385.130401

[59] E. Nachmani, R. San Roman, and L. Wolf, "Non Gaussian Denoising Diffusion Models," Jun 2021, arXiv:2106.07582.

[60] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[61] D. Guo, S. Shamai, and S. Verdu, "Mutual information and minimum mean-square error in gaussian channels," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1261–1282, 2005.

[62] A. Gelman and X.-L. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Statistical Science*, vol. 13, pp. 163–185, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:9883683

[63] N. Fathima Ghouse, J. Petersen, A. Wiggers, T. Xu, and G. Sautière, "A Residual Diffusion Model for High Perceptual Quality Codec Augmentation," Jan 2023, arXiv:2301.05489.

[64] J. Whang, M. Delbracio, H. Talebi, C. Saharia, A. G. Dimakis, and P. Milanfar, "Deblurring via stochastic refinement," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16 272–16 282.

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[66] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6629–6640.

[67] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 11 913–11 924. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/8a50bae297807da9e97722a0b3fd8f27-Paper.pdf

[68] B. Kawar, J. Song, S. Ermon, and M. Elad, "Jpeg artifact correction using denoising diffusion restoration models," in *Neural Information Processing Systems (NeurIPS) Workshop on Score-Based Methods*, 2022.

[69] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 64 971–64 995. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/ccf6d8b4a1fe9d9c8192f00c713872ea-Paper-Conference.pdf

[70] C.-W. Huang, J. H. Lim, and A. Courville, "A variational perspective on diffusion-based generative models and score matching," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=bXehDYUjjXi

[71] G. Franzese, M. BOUNOUA, and P. Michiardi, "MINDE: Mutual information neural diffusion estimation," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=0kWd8SJq8d

[72] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," *ArXiv*, 2015. [Online]. Available: https://arxiv.org/abs/1503.02406

[73] C.-H. Lai, Y. Takida, N. Murata, T. Uesaka, Y. Mitsufuji, and S. Ermon, "FP-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho,

B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 18 365–18 398. [Online]. Available: https://proceedings.mlr.press/v202/lai23d.html

[74] J. Lee, J. Sohl-dickstein, J. Pennington, R. Novak, S. Schoenholz, and Y. Bahri, "Deep neural networks as gaussian processes," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1EA-M-0Z

[75] G. Franzese, S. Rossi, L. Yang, A. Finamore, D. Rossi, M. Filippone, and P. Michiardi, "How much is enough? a study on diffusion times in score-based generative models," *Entropy*, vol. 25, no. 4, Apr 2023. [Online]. Available: http://dx.doi.org/10.3390/e25040633

[76] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 1415–1428. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/0a9fdbb17feb6ccb7ec405cfb85222c4-Paper.pdf

[77] D. Kingma and R. Gao, "Understanding diffusion objectives as the elbo with simple data augmentation," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 65 484–65 516. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/ce79fbf9baef726645bc2337abb0ade2-Paper-Conference.pdf

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[79] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan 2014.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000—6010.

[81] C. Si, Z. Huang, Y. Jiang, and Z. Liu, "FreeU: Free Lunch in Diffusion U-Net," September 2023, arXiv:2309.11497.

[82] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec 2014, arXiv:1412.6980.

[83] D. Lilienthal, P. Mello, M. Eirinaki, and S. Tiomkin, "Multi-resolution diffusion for privacy-sensitive recommender systems," *IEEE Access*, 2024.

[84] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, "Diffusion recommender model," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, vol. 46, 2023, pp. 832–841. [Online]. Available: https://doi.org/10.1145/3539618.3591663

[85] J. Walker, T. Zhong, F. Zhang, Q. Gao, and F. Zhou, "Recommendation via collaborative diffusion generative model," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2022, pp. 593–605.

[86] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, Dec 2015. [Online]. Available: https://doi.org/10.1145/2827872

[87] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 188–197. [Online]. Available: https://aclanthology.org/D19-1018

[88] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, 2007.

[89] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[90] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[91] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds.,

vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf

[92] H. Ishfaq, A. Hoogi, and D. Rubin, "Tvae: Triplet-based variational autoencoder using metric learning," 2018. [Online]. Available: https://openreview.net/forum?id=Sym_tDJwM

[93] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 689–698.