# ENTROPIC TIME SCHEDULERS FOR GENERATIVE DIFFUSION MODELS

**Dejan Stančević**
Donders Institute
Radboud University
Nijmegen, 6525 XZ, NL
dejan.stancevic@donders.ru.nl

**Luca Ambrogioni**
Donders Institute
Radboud University
Nijmegen, 6525 XZ, NL
luca.ambrogioni@donders.ru.nl

## ABSTRACT

The practical performance of generative diffusion models depends on the appropriate choice of the noise scheduling function, which can also be equivalently expressed as a time reparameterization. In this paper, we present a time scheduler that selects sampling points based on entropy rather than uniform time spacing, ensuring that each point contributes an equal amount of information to the final generation. We prove that this time reparameterization does not depend on the initial choice of time. Furthermore, we provide a tractable exact formula to estimate this *entropic time* for a trained model using the training loss without substantial overhead. Alongside the entropic time, inspired by the optimality results, we introduce a rescaled entropic time. In our experiments with mixtures of Gaussian distributions and ImageNet, we show that using the (rescaled) entropic times greatly improves the inference performance of trained models. In particular, we found that the image quality in pretrained EDM2 models, as evaluated by FID and FD-DINO scores, can be substantially increased by the rescaled entropic time reparameterization without increasing the number of function evaluations, with greater improvements in the few NFEs regime.

## 1 INTRODUCTION

Generative diffusion models (Sohl-Dickstein et al., 2015), and especially score-based diffusion models, have achieved state-of-the-art performance in image (Dhariwal & Nichol, 2021; Rombach et al., 2022; Song et al., 2021) and video generation (Ho et al., 2022; Singer et al., 2022). Generative diffusion models are obtained by reverting a forward diffusion process, which injects noise into the distribution of the data until all information has been lost. In practice, the performance of these models is highly dependent on the choice of a noise scheduling function that regulates the rate of noise-injection (Song et al., 2022). In most commonly used models, a change of noise scheduling is mathematically equivalent to a change of time parameterization. From a theoretical perspective, the choice of time parametrization, or equivalently of noise scheduling, is not constrained by theory since any change of time in the forward process is automatically corrected in reverse dynamics (Song et al., 2021). However, as explained above, the choice of time is very important practically since it affects both the temporal weighting during training and the discretization scheme during inference. Consequently, an 'incorrect' choice of time variable can lead to severe inefficiencies due to the under-sampling of some temporal windows and the redundant over-sampling of others. This is particularly problematic since recent theoretical and experimental work suggested that 'generative decisions' tend to be clustered in critical time windows (Raya & Ambrogioni, 2023; Li & Chen, 2024), which have been connected to symmetry-breaking phase transitions in physics (Raya & Ambrogioni, 2023; Ambrogioni, 2024; Biroli et al., 2024; Sclocchi et al., 2024). The "triviality" of the first phase of diffusion prior to the initial phase transitions has led to the idea that this early phase can be skipped in one 'jump' using a pre-trained initialization Lyu et al. (2022); Raya & Ambrogioni (2023). These late initialization schemes can be seen as a special case of time re-scheduling that compresses the high-noise part of the original schedule.

The idea of changing the diffusion time in a data dependent way, also known as time-warping, was first introduced in Dieleman et al. (2022) in the context of a class of diffusion models for sequences

of discrete tokens. However, their implementation required the used of special architectures trained with cross-entropy loss instead of the standard denoising score matching. In this paper, we show that a natural data-dependent time parametrization can be tractably obtained for any continuous generative diffusion model as the (rescaled) conditional entropy of $\mathbf{x}_0$ given $\mathbf{x}_t$. This choice of time leads to a constant entropy rate, meaning that each time point contributes to the final generation in an equal amount. Furthermore, we show that this *entropic time* is invariant, meaning that it does not depend on the original choice of time parameterization. Examples of the same SDE in the entropic time and standard time are given in figure 1. Furthermore, inspired by the optimality results, we introduce a *rescaled entropic time*. We provide an exact tractable formula that relates these quantities to the empirical EDM (Karras et al., 2022) and DDPM (Song et al., 2022) loss, which can be used to easily define the entropic time for any given trained network.

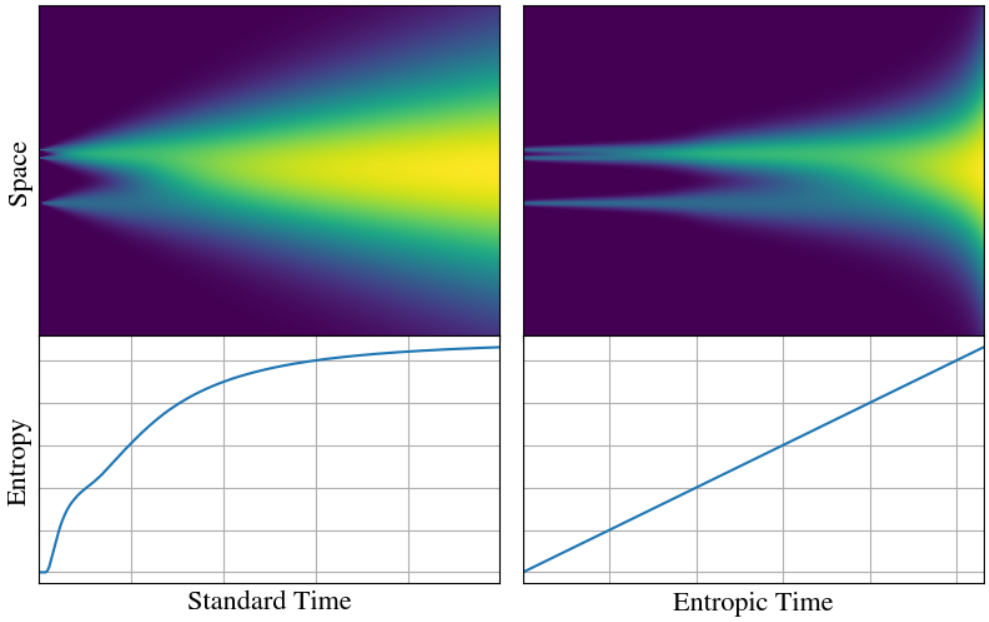



Figure 1: An example of the same SDE and its conditional entropy in the standard and entropic time.

## 2 Related work

**Accelerated Sampling Procedures** One of the most significant challenges in current diffusion models is the slow generative process. Since the introduction of the connection between the diffusion models and SDEs (Song et al., 2021), a wide array of research has aimed to address this issue by designing better numerical integrators. Some of the research in that direction includes the works of Liu et al. (2022) and Lu et al. (2022). An alternative line of research focuses on optimizing the sampling time schedule itself. Sabour et al. (2024) presents a principled approach to optimizing sampling schedules in diffusion models by aligning them with stochastic solvers, enabling higher efficiency. Wang et al. (2024) splits the generation process into three categories (acceleration, deceleration, and convergence steps), identifies imbalances in time step allocation, and introduces methods to address them, leading to faster training and sampling. Lee et al. (2024) uses spectral analysis of images to design a sampling strategy that prioritizes critical time steps, improving quality while reducing the number of steps. Li et al. (2023) explores joint optimization of time steps and architectures for more efficient generation without additional training. While much of this research focuses on learning or empirically determining optimal sampling schedules, our work provides a more theoretical perspective based on ideas from information theory. The closest work to ours is Dieleman et al. (2022), in which they use a cross-entropy loss to deduce a time-warping function for diffusion language models. However, our work differs since we analyze the standard diffusion models, where cross-entropy is not available. Furthermore, their expression for the entropy is not exact as it implies an assumption of conditional independence of the tokens given the noisy state. On the other hand,

here we provide exact formulas that can be applied to any generative diffusion model trained with denoising score matching, both in the continuous and in the discrete regime.

**Connection Between Entropy, Information Theory, and Diffusion Models** Diffusion models are inherently tied to concepts from information theory, particularly in the context of denoising Gaussian noise, which is a fundamental operation in information-theoretic frameworks. This connection has inspired a growing body of work exploring the interplay between diffusion models and information theory. Premkumar (2024) investigates entropy-based objectives for learning more robust generative models. Kong et al. (2023a) and Kong et al. (2023b) aim to provide a clearer understanding of diffusion models through an information-theoretic lens. Although these works explore the connection between information theory and diffusion models, our focus diverges slightly. We use information theory as a guide to design better sampling algorithms. Work exploring a similar direction to ours is Li et al. (2025). However, they explore the conditional entropy between two consecutive time steps given a fixed discretization grid, while we look at the conditional entropy between the current time step and time zero in a way that is invariant under the change of time and discretization.

## 3 BACKGROUND ON SCORE-MATCHING GENERATIVE DIFFUSION

The mathematics of generative diffusion models can be elegantly formalized in term of stochastic differential equations (SDE). Consider a target distribution $p(\mathbf{x}_0)$ defined by a data source such as a distribution of, for example, natural images, sound waves, or linguistic strings. We interpret this data source as the initial distribution of a diffusion process governed by the SDE:

$$dX_t = \boldsymbol{f}(X_t, t)dt + g(t)dW \ , \tag{1}$$

where $dW$ is a standard Wiener process, $\boldsymbol{\mu}(X_t, t)$ is a vector-valued drift function, and $g(t)$ is a scalar volatility function, which regulates the standard deviation of the input noise. The marginal densities of the process can be obtained from the Fokker-Planck equation:

$$\partial_t p_t(\mathbf{x}_t) = \sum_{j=1}^{d} \partial_{x_j} \left(-f_j(\mathbf{x}_t, t) + g(t)\partial_{x_j}\right) p_t(\mathbf{x}_t) \ , \tag{2}$$

where $\partial_t$ is the partial derivative with the spect to time and $\partial_{x_j}$ is the partial derivative with respect to the $j$-th component of $\mathbf{x}_t$. We denote the forward "solution kernel" of the diffusion process as $p(\mathbf{x}_t \mid \mathbf{y})$, which is the solution of the Fokker-Plank equation for $p_0(\mathbf{y}) = \delta(\mathbf{y} - \mathbf{x}_0)$. The core idea of generative diffusion is to sample from $\mathbf{x}_0$ by initializing an asymptotic noise state $\mathbf{x}_T$ (where $T$ is large enough for the SDE to reach its stationary distribution) and by "inverting" the temporal dynamic. This can be done using the reverse SDE:

$$dX_t = \left(f(X_t, t) - g(t)^2 \nabla \log p(X_t)\right) dt + g(t)d\tilde{W} \ , \tag{3}$$

which can be proven to give the same marginal densities of Eq. 3 when initialized with the appropriate stationary distribution, which is usually Gaussian white noise. We denote the reverse solution kernel of the reverse dynamics as $q(\mathbf{x}_0 \mid \mathbf{x}_t)$, which can be interpreted as the optimal denoising distribution. The data-dependent key component of the reverse dynamics is the so-called *score function*, which can be written as an expectation over the optimal denoising distribution:

$$\nabla \log p_t(\mathbf{x}_t) = \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)} \left[\nabla \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right] \ . \tag{4}$$

In most practical forms of generative diffusion, the score function is approximated using a deep network $\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$, where the parameters $\boldsymbol{\theta}$ are optimized by minimizing an upper bound on the quadratic score-matching loss:

$$\begin{aligned} \mathcal{L}_{\text{SM}}(\boldsymbol{\theta}) &\equiv \mathbb{E}_{p_0(x_0), t\sim\lambda(t)} \left[\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla \log p_t(\mathbf{x}_t)\right\|^2\right] \\ &\leq \mathbb{E}_{p_0(x_0), p(x_t|x_0), t\sim\lambda(t)} \left[\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right\|^2\right] \equiv \mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) \end{aligned} \tag{5}$$

where $\lambda(t)$ is a density defined on the time axis. Note that $\mathcal{L}_{\text{sm}}(\boldsymbol{\theta})$ and $\mathcal{L}(\boldsymbol{\theta})$ only differ by a constant and therefore have the same gradients and optima. However, $\mathcal{L}(\boldsymbol{\theta})$ is substantially more tractable as it does not require to sample from the unknown optimal denoiser $q(\mathbf{x}_0 \mid \mathbf{x}_t)$.

# 4 OPTIMAL SAMPLING SCHEDULE AS A CHANGE OF TIME

In this section, we revisit a result from Sabour et al. 2024 and notice some interesting features. Inspired by it, we formalize what we mean by the change of time. Obtaining the analytical expression for the optimal sampling schedule is difficult and, in most practical cases, impossible. However, Sabour et al. 2024 shows that for the EDM noise schedule (Karras et al., 2022), the optimal sampling schedule for the ODE flow when data comes from a normal distribution with variance $c^2$ has an analytical expression. More precisely, the sampling schedule, $[t_{min}, t_1, ...t_N max$, that minimize the KL divergence is given by

$$\arctan\left(\frac{t_i}{c}\right) = \alpha_{min} + \frac{i}{N}(\alpha_{max} - \alpha_{min}) \tag{6}$$

where $\alpha_{min/max} = \arctan\left(\frac{t_{min/max}}{c}\right)$ (see theorem 3.1 in Sabour et al. 2024). From here, we can see that even in the simple case, the optimal schedule depends on the data distribution. In addition, this result frames the optimization of a sampling schedule as a problem of time change. Rather than selecting timesteps differently for different numbers of sampling steps (e.g. EDM scheduler), Theorem 3.1 shows that one should think of the sampling schedule as a transformation of time such that the sampling schedule becomes linear in the new time. Furthermore, in section 5.3, we will connect equation 6 with the conditional entropy production.

## 4.1 CHANGE OF TIME

The change of time in SDEs is a powerful technique used to simplify their analysis and solutions. By altering the time variable, the dynamics of the SDE can be transformed into a more manageable form. More information can be found in section 8.3 in Lawler (2010).

**Definition 4.1.** *We say a function $\phi$ is a proper time change if it is continuous and strictly increasing.*

It can be shown that given a proper time change, $f$, and a random process, $X_t$, that solves the SDE $dX_t = f_t(X_t, t)dt + g_t(t)dW_t$ , then $Y_t = X_{\phi(t)}$ solves $dY_t = \dot{\phi}(t)f_t(Y_t, \phi(t))dt + \sqrt{\dot{\phi}(t)}g_t(\phi(t))dW_t$ . Guided by the theory of time change, we define an equivalence between SDEs.

**Definition 4.2.** *Given two SDEs $dX_t = f_t(X_t, t)dt + g_t(t)dW_t$ and $dX_s = \tilde{f}_s(X_s, s)ds + \tilde{g}_s(s)dW_s$ , we say that they are equivalent up to a time change if there exists a proper time change, $\phi : t \mapsto s$, such that*

1. $\dot{\phi}(t)\tilde{f}_s(x, \phi(t)) = f_t(x, t)$
2. $\sqrt{\dot{\phi}(t)}\tilde{g}_s(\phi(t)) = g_t(t)$

*where subscripts $s$ and $t$ of drift and noise terms are to distinguish their connection to a specific time parameterization.*

Furthermore, we can require $f(0) = 0$ without affecting anything (since it is equivalent to subtracting a constant from the original function). By requiring that, we get that a time change between two SDEs is unique if it exists. Under a time change, the forward kernels stay the same, in the sense that $p_t(x|x_0) = q_{\phi(t)}(x|x_0)$ holds (this follows from $Y_t = X_{\phi(t)}$). Essentially, time change squeezes and stretches the time axis but does not fundamentally change the diffusion process. Hence, the natural question is whether there is a natural time parameterization that should be used. We argue that a conditional entropy, $\mathbf{H}[x_0|x_t]$, and quantities derived from it are good candidates. However, for $\mathbf{H}[x_0|x_t]$ to make sense, we assume that we are given an initial distribution, $p_0(x)$ (that is, a data set). Therefore, besides an SDE, we require a dataset for the entropic time. In the further text, we will always assume that the dataset is given and is the same for different time parameterizations of SDE.

# 5 ENTROPIC TIME SCHEDULES

In this section, we introduce the concepts of entropic time and rescaled entropic time. First, we provide some reasons for using the conditional entropy as a new time parameterization. Then, we show

how to obtain the conditional entropy in practice and show its connection with commonly used quantities in diffusion literature. Furthermore, we demonstrate that the entropic time parameterizations are well-defined and invariant under the initial time parameterization of the SDE.

There are serval possible choices for the entropy function, which highlights different aspects of information transfer. The most straightforward choice is the information transfer $T_t$. Consider an initial source $\mathbf{x}_0 \sim p_0$ is transmitted through a noisy channel $p(\mathbf{x}_t \mid \mathbf{x}_0)$, which is determined by the solution of the SDE given in Eq. 3. The noise corrupted signal is received and decoded using $q(\mathbf{x}_0 \mid \mathbf{x}_t)$. The amount of information transferred at time $t$ can be quantified as the difference between the prior and posterior entropy:

$$\mathbf{T}_t = \mathbf{H}[\mathbf{x}_0] - \mathbf{H}[\mathbf{x}_0|\mathbf{x}_t] = \mathbf{I}[\mathbf{x}_0; \mathbf{x}_t] \tag{7}$$

where $\mathbf{H}[\mathbf{x}_0] = \mathbb{E}_{p_0(x_0)}[\log p(\mathbf{x}_0)]$ is the entropy of the source, $\mathbf{H}[\mathbf{x}_0|\mathbf{x}_t] = \mathbb{E}_{p(x_0,x_t)}[\log p(\mathbf{x}_0|\mathbf{x}_t)]$ is the conditional entropy under the optimal denoising distribution, and $\mathbf{I}[\mathbf{x}_0; \mathbf{x}_t]$ is a mutual information. Therefore, it is natural to interpret this quantity as the amount of information available at time $t$ concerning the identity of the source data. Up to a constant shift, this is equivalent to using the time variable $\phi(t) = \mathbf{H}[\mathbf{x}_0|\mathbf{x}_t]$ in the forward process. This time axis is defined by having a constant conditional entropy rate between the final generated image and the noisy state at time $t$.

### 5.1 Characterizing the conditional entropy

Having established that a conditional entropy makes sense as a new time parameterization, a question arises: How do we calculate it in practice? In general, conditional entropy can be written as $\mathbf{H}[\mathbf{x}_0|\mathbf{x}_t] = \mathbf{H}[\mathbf{x}_0] - \mathbf{I}[\mathbf{x}_0; \mathbf{x}_t] = \mathbf{H}[\mathbf{x}_0] + \mathbf{H}[\mathbf{x}_t|\mathbf{x}_0] - \mathbf{H}[\mathbf{x}_t]$ .

In practice, $\mathbf{H}[\mathbf{x}_t|\mathbf{x}_0]$ is easy to get once the forward kernel is known, but it is difficult to obtain a numerical value of $\mathbf{H}[\mathbf{x}_t]$. However, by looking at a time derivative of the conditional entropy, we get a method for obtaining a numerical value. The time derivative is given by

$$\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t] = \dot{\mathbf{H}}[\mathbf{x}_t|\mathbf{x}_0] - \dot{\mathbf{H}}[\mathbf{x}_t]. \tag{8}$$

Hence, to know the time derivative, we need to calculate the time derivative of $\mathbf{H}[\mathbf{x}_t]$. In case when an SDE is given by 4.1, the entropy production is given by

$$\dot{\mathbf{H}}[\mathbf{x}_t] = \mathbb{E}_{p_t(x_t)}[\nabla(f_t)] + \frac{g_t^2}{2}\mathbb{E}_{p_t(x_t)}[||\nabla \log p(\mathbf{x}_t)||^2]. \tag{9}$$

The equation is a well-known expression in nonequilibrium thermodynamics for entropy production (Premkumar, 2024). The derivation of the expression can be found in the appendix A. Similarly, we can obtain the similar expression for $\mathbf{H}[\mathbf{x}_t|\mathbf{x}_0]$. Combining these two expressions, we obtain

$$\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t] = \frac{g_t^2}{2}\left(\mathbb{E}_{p(x_t,x_0)}[||\nabla \log p(\mathbf{x}_t|\mathbf{x}_0)||^2] - \mathbb{E}_{p_t(x_t)}[||\nabla \log p(\mathbf{x}_t)||^2]\right). \tag{10}$$

Note that this expression depends on the data distribution only through the Euclidean norm of the score function, which is approximated by a neural network in diffusion models.

### 5.2 Estimating the entropy rate from the training loss

In this section, we present a connection between the conditional entropy rate and training loss. For more details on the derivation of these results, see the Appendix D. In practice, most diffusion models can be written using the framework introduced in Karras et al. (2022). In this framework, the SDE is written as $dX_t = \frac{\dot{s}(t)}{s(t)}X_t dt + s(t)\sqrt{2\dot{\sigma}(t)\sigma(t)}dW$ , with $p(x_t|x_0) = \mathcal{N}(x_t; s(t)x_0, s(t)^2\sigma(t)^2 I)$ as a forward kernel. This leads to the following conditional entropy production

$$\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t] = \frac{D\dot{\sigma}(t)}{\sigma(t)} - s(t)^2\dot{\sigma}(t)\sigma(t)\mathbb{E}_{p_t(x_t)}[||\nabla \log p_t(\mathbf{x}_t)||^2] \tag{11}$$

where $D$ is a dimension of the space (e.g. for the MNIST dataset, it would be $28^2$). In the rest of this paper, we will be using this framework.

The squared error, $\epsilon_t^2$, encapsulates our uncertainty at time $t$ about the final sample $x_0$ and is given by

$$\epsilon_t^2 = \mathbb{E}_{p_t(x_t)}[\mathbb{E}_{p(x_0|x_t)}[||\mathbf{x}_0 - \mathbb{E}_{p(y_0|x_t)}[\mathbf{y}_0]||^2]] = \mathbb{E}_{p_t(x_t)}[tr(\sigma^2_{\mathbf{x}_0|\mathbf{x}_t})]. \tag{12}$$

Using the fact that we can write $\sigma^2_{\mathbf{x}_0|\mathbf{x}_t}$ as $\sigma(t)^2(I + s(t)^2\sigma(t)^2 H[\log p_t(x_t)])$ (see Appendix E), we get

$$\epsilon_t^2 = \frac{\sigma(t)^3}{\dot{\sigma}(t)}\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t]. \tag{13}$$

Recognizing that $S\dot{N}R = -\frac{\dot{\sigma}}{\sigma^3}$ and $\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t] = -\dot{\mathbf{I}}[\mathbf{x}_t;\mathbf{x}_0]$, equation 13 becomes a well-known $I - mmse$ relation in information theory (Guo et al., 2005).

The previous results provide a simple way of estimating the conditional entropy rate from the standard loss function of a trained diffusion model due to a close connection between the squared error and the loss. This provides a tractable way to estimate the conditional entropy from the training error. Note that, using the error of the model entails an approximation since the entropy is defined with respect to the true score function and, therefore, does not take into account the discrepancy between the learned and true score.

To analyze this deviation, we start from a striking result: the conditional entropy production is, up to a multiplicative factor, the gap between the explicit and denoising score matching loss in 5! In fact, following the steps from Vincent (2011) and keeping track of the terms that are constant in $\boldsymbol{\theta}$, we have

$$\mathcal{L}_{\text{SM}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) - \mathbb{E}_{p(x_0,x_t),\lambda(t)}[\|\nabla \log p(\mathbf{x}_t|\mathbf{x}_0)\|^2 - \|\nabla \log p(\mathbf{x}_t)\|^2]. \tag{14}$$

Using expression 10, we can rewrite the above equality as

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{SM}}(\boldsymbol{\theta}) + \mathbb{E}_{\lambda(t)}\left[\frac{2}{g_t^2}\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t]\right]. \tag{15}$$

This expression can also be given for a single time point $t$:

$$\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t] + \frac{g_t^2}{2}\delta_t^2(\boldsymbol{\theta}) = \frac{g_t^2}{2}\mathbb{E}_{\mathbf{x}_t,\mathbf{x}_0}\left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t) - \nabla \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\|^2\right] \tag{16}$$

where $\delta_t^2(\boldsymbol{\theta}) = \mathbb{E}_{p_t(x_t)}\left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t) - \nabla \log p_t(\mathbf{x}_t)\|^2\right]$ is the squared error between the true score and the neural estimate and the left hand side is our estimate of the conditional entropy production. This implies that the estimated entropy production is always larger than the true value. In this sense, we can interpret $\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t]$ as the 'unavoidable' component of the loss, which is caused by intrinsic undecidability in the optimal denoising.

### 5.3 The entropic and rescaled entropic times

Here, we introduce a rescaled entropy and show that both rescaled entropy and conditional entropy are proper changes of times and are invariant under different time parameterizations of SDE. Proofs can be found in the Appendix B.

First, we notice that in the case of continuous data, the conditional entropy goes to negative infinity. In practice, this is not observed since diffusion models always start from a non-zero initial time. However, it adds arbitrariness to the overall curve of the conditional entropy. To combat this problem, guided by the observation that the change of time for the optimal sampling schedule for normally distributed data, eq. 6, is equal to the rescaled entropy (see Appendix C), we introduce a rescaled entropy as $\int_0^t \sigma(\tau)\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_\tau]d\tau$.

**Theorem 5.1.** *Given an SDE and initial data distribution $p_0(x)$, $\phi(t) = \mathbf{H}[x_0|x_t]$ and $\phi(t) = \int_0^t \sigma(\tau)\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_\tau]d\tau$ are proper time changes.*

We call these time parameterizations an *entropic time* and *rescaled entropic time*, respectively.

Naturally, an important question emerges: How does the time parameterization of an initial SDE influence its reparameterized form? We show that an SDE written in entropic time is unique and does not rely on its initial parameterization. More precisely, given two SDEs equivalent up to a time change, the SDEs expressed in their respective entropic times are equivalent up to a time change, with the time change being the identity function (i.e. drift and noise terms of SDEs in entropic times are related by conditions 1. and 2. from definition 4.2, and are the same since the time derivative of the time change is one).

**Theorem 5.2.** *Given two SDEs as given in definition 4.2, and following time changes*

1. $\phi : t \mapsto s = f(t)$
2. $\Phi_t : t \mapsto \mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t]$
3. $\Phi_s : s \mapsto \mathbf{H}_s[\mathbf{x}_0|\mathbf{x}_s]$,

*it follows that*

$$F := \Phi_s \circ \phi \circ \Phi_t^{-1} : \mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t] \mapsto \mathbf{H}_s[\mathbf{x}_0|\mathbf{x}_s]$$

*is a proper time change implementing the equivalence and is equal to the identity map, $F = id$.*

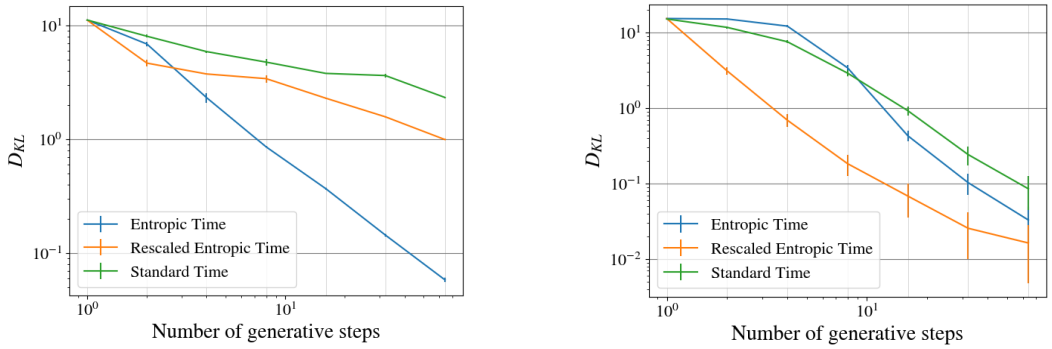
A similar result holds for the rescaled entropic time as well. Therefore, once reparameterized in entropic time (or rescaled entropic time), no matter the starting SDE time parameterization, drift and noise are always the same.

## 6 Experiments

We compare the performance of a few-step generation in the standard, entropic, and rescaled entropic times for several low-dimensional examples where an analytic expression for a score is easy to calculate. Next, we compare the performance of a trained EDM2 models (Karras et al., 2024) on ImageNet-64 using the FID (Heusel et al., 2017) and FD-DINOv2 (Oquab et al., 2023) scores. More details can be found in appendix F.

### 6.1 One-dimensional experiments

We used an analytic expression of a score function to compare the performance of a few-step generation process in different time parameterizations in one dimension. We used equidistant steps in the standard, entropic, and rescaled entropic times. We used the stochastic DDIM solver (Song et al., 2022). We compared those schedules for discrete data and a mixture of Gaussians. We used the Kullback-Leibler divergence to compare results for different schedules. An example of KL divergence behavior against the number of generative steps is given in figure 2. In general, we can see that in the discrete case, the entropic time outperforms other schedules by a large margin, while the standard schedule gives the worst results. Furthermore, we noticed that when variances of Gaussians are much smaller than the distance between them (i.e. there is no significant overlap between Gaussians), the entropic schedule gives better results. However, when the variances are not negligible in the mixture of Gaussians case, we can see that the rescaled entropic schedule gives the best results, while the entropic schedule underperforms. This suggests that the entropic time might significantly improve certain discrete diffusion models.




(a) Discrete

(b) Continuous

Figure 2: Kullback–Leibler divergence against the number of generative steps for different time parameterizations for mixture of 15 data points (discrete) and 15 Gaussians (continuous).

Figure 3: Images generated with the deterministic DDIM sampler using the rescaled entropic schedule over 64 steps, with the EDM2-L model.

### 6.2 IMAGENET

We compared the performance of the EDM2-S and EDM2-L ImageNet-64 models for different numbers of generative steps using standard, entropic, and rescaled entropic times. To sample, we used the deterministic and stochastic DDIM solver. An example of generated images is given in figure 3. The results are given in table 1. We observed that the entropic time produced unrecognizable images (see Appendix F), therefore, we have not included it in the table with the results. From the table, we can see that the schedule based on rescaled entropy consistently produces better results. More examples of generated images are given in Appendix F.

Table 1: FID and FD-DINOv2 scores for different sampling schedules for ImageNet-64

| Solver | Schedule | Network Size | FID ↓ | | | FD-DINOv2 ↓ | | |
|---|---|---|---|---|---|---|---|---|
| | | | NFE=16 | NFE=32 | NFE=64 | NFE=16 | NFE=32 | NFE=64 |
| Stochastic DDIM | EDM2-S | EDM | 20.03 | 8.18 | 3.81 | 263.60 | 135.67 | 86.32 |
| | | Rescaled Entropy | **11.69** | **4.95** | **2.75** | **194.02** | **109.55** | **81.25** |
| | EDM2-L | EDM | 22.60 | 9.46 | 4.44 | 284.74 | 141.70 | 79.86 |
| | | Rescaled Entropy | **13.56** | **5.59** | **3.06** | **208.27** | **108.31** | **72.06** |
| Deterministic DDIM | EDM2-S | EDM | 5.00 | 2.49 | 1.90 | 128.25 | 99.64 | **92.88** |
| | | Rescaled Entropy | **3.46** | **2.15** | **1.77** | **117.26** | **98.28** | 93.34 |
| | EDM2-L | EDM | 5.49 | 2.55 | 1.82 | 120.35 | 84.57 | **74.87** |
| | | Rescaled Entropy | **3.63** | **2.09** | **1.65** | **104.88** | **81.98** | 75.87 |

## 7 CONCLUSIONS

In this paper, we introduced the concept of entropic time as a natural time reparameterization for generative diffusion models. By ensuring a constant entropy rate across sampling points, entropic time equalizes the contribution of each time step to the generative process. Following the observation that the entropic time can be connected to the optimal sampling schedule for Gaussian data, we introduced the rescaled entropic time. Theoretical results demonstrated the invariance of these times, making them invariant to the initial parameterization of the stochastic differential equations.

Empirical results in the toy dataset show the promise of entropic time, especially when the data is discrete. Furthermore, experiments on ImageNet-64 show that the rescaled entropic time can achieve substantial improvements in FID and FD-DINOv2 scores when compared with the standard EDM schedule. These findings highlight the potential for incorporating information-theoretic principles into the design of sampling schedules for diffusion models. In future work, it would be interesting to combine the idea of entropic time and discrete diffusion, something akin to time warping in Dieleman et al. (2022).

## REFERENCES


Luca Ambrogioni. The statistical thermodynamics of generative diffusion models: Phase transitions, symmetry breaking and critical instability, 2024. URL https://arxiv.org/abs/2310.17467.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1), November 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54281-3. URL http://dx.doi.org/10.1038/s41467-024-54281-3.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Alex Dytso, H Vincent Poor, and Shlomo Shamai Shitz. Conditional mean estimation in gaussian noise: A meta derivative identity with applications. *IEEE Transactions on Information Theory*, 69(3):1883–1898, 2022.

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in gaussian channels. *IEEE transactions on information theory*, 51(4):1261–1282, 2005.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. *arXiv preprint arXiv:2302.03792*, 2023a.

Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. Interpretable diffusion via information decomposition. *arXiv preprint arXiv:2310.07972*, 2023b.

Gregory F Lawler. Stochastic calculus: An introduction with applications. *American Mathematical Society*, 2010.

Haeil Lee, Hansang Lee, Seoyeon Gye, and Junmo Kim. Beta sampling is all you need: Efficient image generation strategy for diffusion models using stepwise spectral analysis. *arXiv preprint arXiv:2407.12173*, 2024.

Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7105–7114, 2023.

Marvin Li and Sitan Chen. Critical windows: non-asymptotic theory for feature emergence in diffusion models, 2024. URL https://arxiv.org/abs/2403.01633.

Shigui Li, Wei Chen, and Delu Zeng. Improving denoising diffusion with efficient conditional entropy reduction, 2025. URL https://openreview.net/forum?id=OT2NFdNrny.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

C Lu, Y Zhou, F Bao, J Chen, and C Li. A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Proc. Adv. Neural Inf. Process. Syst., New Orleans, United States*, pp. 1–31, 2022.

Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Akhil Premkumar. Neural entropy, 2024. URL https://arxiv.org/abs/2409.03817.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models, 2023. URL https://arxiv.org/abs/2305.19693.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. *arXiv preprint arXiv:2404.14507*, 2024.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data, 2024. URL https://arxiv.org/abs/2402.16991.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. URL https://arxiv.org/abs/2209.14792.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

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

Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training. *arXiv preprint arXiv:2405.17403*, 2024.

## A ENTROPY PRODUCTION

Here, we show

$$\dot{\mathbf{H}}[\mathbf{x}_t] = \mathbb{E}_{p_t(x_t)}[\nabla(f)] + \frac{g_t^2}{2}\mathbb{E}_{p_t(x_t)}[||\nabla \log p(\mathbf{x}_t)||^2]. \tag{17}$$

By looking inside the integral of $\dot{\mathbf{H}}[\mathbf{x}_t]$, we get

$$\begin{aligned}\dot{\mathbf{H}}[\mathbf{x}_t] &= -\int \left(\dot{p}(x_t)\log p(x_t) + p(x_t)\frac{\dot{p}(x_t)}{p(x_t)}\right) dx_t \\ &= -\int \dot{p}(x_t)\log p(x_t)dx_t - \frac{d}{dt}\int p(x_t)dx_t \\ &= -\int \dot{p}(x_t)\log p(x_t)dx_t.\end{aligned} \tag{18}$$

Assuming our dynamic is determined by the SDE 4.1, we can use the Fokker-Planck equation to simplify the derivative as follows

$$\begin{aligned}\dot{\mathbf{H}}[\mathbf{x}_t] &= -\int \left(-\nabla\left(\left(f_t - \frac{g_t^2}{2}\nabla \log p(x_t)\right)p(x_t)\right)\right)\log p(x_t)dx_t \\ &= -\int \left(f_t - \frac{g_t^2}{2}\nabla \log p(x_t)\right)\nabla \log p(x_t)p(x_t)dx_t \\ &= -\int f_t \nabla \log p(x_t)p(x_t)dx_t + \int \frac{g_t^2}{2}\nabla \log p(x_t)\nabla \log p(x_t)p(x_t)dx_t \\ &= -\int f_t \nabla p(x_t)dx_t + \int \frac{g_t^2}{2}||\nabla \log p(x_t)||^2 p(x_t)dx_t \\ &= \int \nabla(f_t)p(x_t)dx_t + \frac{g_t^2}{2}\mathbb{E}_{p_t(x_t)}[||\nabla \log p(\mathbf{x}_t)||^2] \\ &= \mathbb{E}_{p_t(x_t)}[\nabla(f_t)] + \frac{g_t^2}{2}\mathbb{E}_{p_t(x_t)}[||\nabla \log p(\mathbf{x}_t)||^2]\end{aligned} \tag{19}$$

which is exactly what we wanted to show. We used integration by parts in going from the first line to the second and from the fourth to the fifth.

## B PROOFS FROM SECTION 5.3

**Theorem B.1.** *Given an SDE and initial data distribution $p_0(x)$, $\phi(t) = \mathbf{H}[x_0|x_t]$ and $\phi(t) = \int_0^t ds\sigma_s\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_s]$ are proper time changes.*

*Proof.* As already mentioned, a proper time change must be a strictly increasing continuous function. Since $\mathbf{H}[x_0|x_t]$ has a derivative (see section 5.1), we need to show that it is positive. However, our claim follows from equation 13 (the squared error is equal to zero only when an initial distribution consists of one data point). □

**Theorem B.2.** *Given two SDEs as given in definition 4.2, and following time changes*

1. $\phi : t \mapsto s = f(t)$
2. $\Phi_t : t \mapsto \mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t]$
3. $\Phi_s : s \mapsto \mathbf{H}_s[\mathbf{x}_0|\mathbf{x}_s]$,

*it follows that*

$$F := \Phi_s \circ \phi \circ \Phi_t^{-1} : \mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t] \mapsto \mathbf{H}_s[\mathbf{x}_0|\mathbf{x}_s]$$

*is a proper time change implementing the equivalence and is equal to the identity map, $F = id$.*

*Proof.* Immediately, we can see that $g$ is a proper time change since it is composed of other time changes. Similarly, using a chain rule, it is observed that $g$ implements the equivalence. Furthermore,

$$F(\mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t]) = (\Phi_s \circ \phi)(\Phi_t^{-1}(\mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t])) = \Phi_s(\phi(t)) = \mathbf{H}_s[\mathbf{x}_0|\mathbf{x}_{\phi(t)}]. \tag{20}$$

However, since $p_t(x) = q_{\phi(t)}(x)$ and $p(x_t|x_0) = q(x_{\phi(t)}|x_0)$, it follows

$$\begin{aligned} \mathbf{H}_t[\mathbf{x}_0|\mathbf{x}_t] &= -\iint p(x_t, x_0) \ln\left(p(x_0|x_t)\right) dx_0 dx_t \\ &= -\iint q(x_{\phi(t)}, x_0) \ln\left(q(x_0|x_{\phi(t)})\right) dx_0 dx_{\phi(t)} = \mathbf{H}_s[\mathbf{x}_0|\mathbf{x}_{\phi(t)}], \end{aligned} \tag{21}$$

where $x_t = x_{\phi(t)}$ (i.e. are the same spatial point) and time subscripts represent at which point in time probability distribution is evaluated. This proves that $F = id$. □

Similarly, we can prove the same claim for the rescaled entropic time since $\sigma(t) = \sigma(\phi(t))$ for any proper change of time $\phi$.

## C Rescaled entropy for Gaussian data

Here, we show that, in the case of the EDM noise schedule, the rescaled entropic time is the optimal sampling schedule for the ODE flow when data comes from a normal distribution with variance $c^2$ (equation 6).

Recall the expression for the rescaled entropy, $\int_0^t \sigma(\tau)\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_\tau]d\tau$. From equation 11, we have

$$\int_0^t \sigma(\tau)\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_\tau]d\tau = \int_0^t \left(D\dot{\sigma}(\tau) - s(\tau)^2\dot{\sigma}(\tau)\sigma(\tau)^2\mathbb{E}_{p_\tau(x_\tau)}[||\nabla \log p_\tau(\mathbf{x}_\tau)||^2]\right) d\tau. \tag{22}$$

Using the facts that $\sigma(\tau) = s$, $s(\tau) = 1$ and $\nabla \log p_\tau(x_\tau) = \frac{-x_\tau}{s(\tau)^2\sigma(\tau)^2 + s(\tau)^2c^2}$, we get

$$\begin{aligned} \int_0^t \sigma(\tau)\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_\tau]d\tau &= \int_0^t \left(D - \tau^2\frac{D}{\tau^2 + c^2}\right) d\tau \\ &= \int_0^t \frac{Dc^2}{\tau^2 + c^2} d\tau = Dc\arctan\left(\frac{t}{c}\right). \end{aligned} \tag{23}$$

Therefore, a linear sampling schedule, $[t_{min}, t_1, ...t_N max$, in the rescaled entropic time is given by

$$Dc\arctan\left(\frac{t_i}{c}\right) = Dc\left(\alpha_{min} + \frac{i}{N}(\alpha_{max} - \alpha_{min})\right) \tag{24}$$

where $\alpha_{min/max} = \arctan\left(\frac{t_{min/max}}{c}\right)$. Exactly the same as equation 6.

## D Connection with a squared error and loss

In this Appendix, we show connections between conditional entropy production and some commonly used expressions in the diffusion literature. Firstly, we show how the conditional entropy production is related to the squared error at time $t$, $\epsilon_t^2$.

$$\begin{aligned} \epsilon_t^2 = \mathbb{E}_{p(x_0,x_t)}[||\mathbf{x}_0 - \hat{\mathbf{x}}_0(\mathbf{x}_t)||^2] &= \iint ||x_0 - \hat{x}_0(x_t)||^2 p(x_t|x_0)p(x_0)dx_t dx_0 \\ &= \iint \left|\left|x_0 - \frac{(x_t + s(t)^2\sigma(t)^2\nabla(\log p(x_t))}{s(t)}\right|\right|^2 p(x_t|x_0)p(x_0)dx_t dx_0 \end{aligned} \tag{25}$$

The squared error encapsulates our uncertainty at time $t$ about the final sample $x_0$. The following simplification of the above equation gives a more precise meaning.

$$\begin{aligned}\epsilon_t^2 &= \mathbb{E}_{p_t(x_t)}[\mathbb{E}_{p(x_0|x_t)}[||\mathbf{x}_0 - \mathbb{E}_{p(y_0|x_t)}[\mathbf{y}_0]||^2]] \\ &= \mathbb{E}_{p_t(x_t)}[tr(\sigma^2_{\mathbf{x}_0|\mathbf{x}_t})].\end{aligned} \tag{26}$$

From Appendix E, we know

$$Var_{p(x_0|x_t)}[\mathbf{x}_0] = \sigma(t)^2(I + s(t)^2\sigma(t)^2 H[\log p_t(x_t)]). \tag{27}$$

Hence,

$$\begin{aligned}\epsilon_t^2 = \mathbb{E}_{p_t(x_t)}[tr(\sigma^2_{\mathbf{x}_0|\mathbf{x}_t})] &= \sigma(t)^2\mathbb{E}_{p_t(x_t)}[tr(I + s(t)^2\sigma(t)^2 H[\log p(\mathbf{x}_t)])] \\ &= \sigma(t)^2(D - s(t)^2\sigma(t)^2\mathbb{E}_{p_t(x_t)}[||\nabla \log(p_t(\mathbf{x}_t))||^2]) \\ &= \frac{\sigma(t)^3}{\dot{\sigma}(t)}\left(\frac{D\dot{\sigma}(t)}{\sigma(t)} - s(t)^2\dot{\sigma}(t)\sigma(t)\mathbb{E}_{p_t(x_t)}[||\nabla \log(p_t(\mathbf{x}_t))||^2]\right) \\ &= \frac{\sigma(t)^3}{\dot{\sigma}(t)}\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t].\end{aligned} \tag{28}$$

Following notation from Karras et al. (2022) for the loss at time $t$, we have

$$\mathcal{L}(t) = \mathbb{E}_{p_0(x_0),\mathcal{N}(\epsilon;0,I)}[\lambda(t)\,\|c_{out}(t)F_\theta - s(t)x_0 + c_{skip}(t)(s(t)x_0 + s(t)\sigma(t)\epsilon)\|^2]. \tag{29}$$

The formula for a prediction $\hat{x}_0(x_t)$ is given by

$$\hat{x}_0(x_t) = \frac{c_{out}(t)F_\theta(x_t) + c_{skip}(t)x_t}{s(t)}. \tag{30}$$

We can express the loss at time $t$ using the squared error as

$$\mathcal{L}(t) = \lambda(t)\mathbb{E}[\|s(t)\mathbf{x}_0 - s(t)\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2] = \lambda(t)s(t)^2\epsilon_t^2. \tag{31}$$

Furthermore, using the connection between a squared error and a conditional entropy production, we get

$$\dot{\mathbf{H}}[\mathbf{x}_0|\mathbf{x}_t] = \frac{\dot{\sigma}(t)}{\lambda(t)s(t)^2\sigma(t)^3}\mathcal{L}(t). \tag{32}$$

## E TWEEDIE'S SECOND ORDER FORMULA

Assume we are given a distribution $p(y)$ that is obtained by adding a Gaussian noise to a distribution $q(x)$, i.e. $q(y|x) = \mathcal{N}(y; sx, s^2\sigma^2)$.

Now given some $y \sim p(y)$, if we are interested in which $x \sim q(x)$ generated it, the best we can do is guess $\hat{x}(y) = E_{q(x|y)}[x]$. Tweedie's formula gives us

$$\mathbb{E}_{q(x|y)}[x] = \frac{y + s^2\sigma^2\nabla_y \log p(y)}{s} \tag{33}$$

Now, we might ask how sure we are in our guess. To answer that question, we need to look at the variance, $Var_{q(x|y)}[x] = \mathbb{E}_{q(x|y)}[x^2] - \mathbb{E}_{q(x|y)}[x]^2$. In this section, we derive the following result

$$Var_{q(x|y)}[x] = s\sigma^2\nabla_y E_{q(x|y)}[x] = \sigma^2(I + s^2\sigma^2 H[\log p(y)]). \tag{34}$$

However, a more general result regarding the cumulants of $q(x|y)$ holds (Dytso et al., 2022). That is, all the cumulants can be calculated using the score function and its derivatives.

Since we already have $\mathbb{E}_{q(x|y)}[x]$, we need to find an expression for $\mathbb{E}_{q(x|y)}[x^2]$.

$$\begin{aligned}\mathbb{E}_{q(x|y)}[x^2] &= \int dx \frac{q(y|x)q(x)}{p(y)} x^2 = \int dx \frac{q(x)x}{p(y)} q(y|x)x \\ &= \int dx \frac{xq(x)}{p(y)} \frac{yq(y|x) + s^2\sigma^2 \nabla_y q(y|x)}{s} \\ &= \frac{y\mathbb{E}_{q(x|y)}[x]}{s} + \frac{s^2\sigma^2}{sp(y)} \nabla_y \int dx q(y|x)q(x)x\end{aligned} \tag{35}$$

Where in going from the first line to the second, we used $\nabla_y q(y|x) = \frac{sx-y}{s^2\sigma^2} q(y|x)$. However, we seem to have encountered a problem with the second term in our expression. However, by using $q(x, y) = q(y|x)q(x) = q(x|y)p(y)$, for the second term we get

$$\begin{aligned}\nabla_y \int dx q(y|x)q(x)x &= \nabla_y \int dx q(x|y)p(y)x \\ &= \nabla_y \left( p(y) \int dx q(x|y)x \right) = \nabla_y (p(y)\mathbb{E}_{q(x|y)}[x]) \\ &= \mathbb{E}_{q(x|y)}[x]\nabla_y p(y) + p(y)\nabla_y \mathbb{E}_{q(x|y)}[x].\end{aligned} \tag{36}$$

Hence,

$$\begin{aligned}\mathbb{E}_{q(x|y)}[x^2] &= \frac{y\mathbb{E}_{q(x|y)}[x]}{s} + \frac{s^2\sigma^2}{sp(y)} (\mathbb{E}_{q(x|y)}[x]\nabla_y p(y) + p(y)\nabla_y \mathbb{E}_{q(x|y)}[x]) \\ &= \frac{y\mathbb{E}_{q(x|y)}[x]}{s} + \frac{s^2\sigma^2(\mathbb{E}_{q(x|y)}[x]\nabla_y \log p(y) + \nabla_y \mathbb{E}_{q(x|y)}[x])}{s} \\ &= \mathbb{E}_{q(x|y)}[x] \frac{y + s^2\sigma^2\nabla_y \log p(y)}{s} + \frac{s^2\sigma^2}{s} \nabla_y \mathbb{E}_{q(x|y)}[x] \\ &= E_{q(x|y)}[x]^2 + \frac{s^2\sigma^2}{s} \nabla_y \mathbb{E}_{q(x|y)}[x].\end{aligned} \tag{37}$$

Now, we get an elegant expression for the variance

$$Var_{q(x|y)}[x] = s\sigma^2 \nabla_y \mathbb{E}_{q(x|y)}[x] = \sigma^2(1 + s^2\sigma^2 \partial_{yy} \log p(y)). \tag{38}$$

So far, we have been dealing with one-dimensional random variables, but it is easy to generalize all the steps to arbitrary dimensions, which gives us the general formula

$$Var_{q(x|y)}[x] = s\sigma^2 \nabla_y E_{q(x|y)}[x] = \sigma^2(I + s^2\sigma^2 H[\log p(y)]). \tag{39}$$

# F Experimental details and Additional results

## F.1 One-dimensional experiments

We used an analytic expression of a score function to compare the performance of a few-step generation process in different time parameterizations in one dimension. We used equidistant steps in the standard time, entropic time, and rescaled entropic time. All entropic quantities were obtained from the squared error using equation 13. The squared error was estimated at $10^4$ equidistant timesteps with $10^3$ samples at each timestep. We used a mixture of data points (discrete case) and a mixture of Gaussians (continuous case). In both cases, data had a mean of zero and a standard deviation of one. For sample generation we used the stochastic DDIM (Song et al., 2022). Results are given in figure 4.

For the discrete case, the performance was measured by creating nonoverlapping bins around data points, $[a_i - \epsilon, a_i + \epsilon]$, and calculating the Kullback-Leibler divergence between the initial distribution

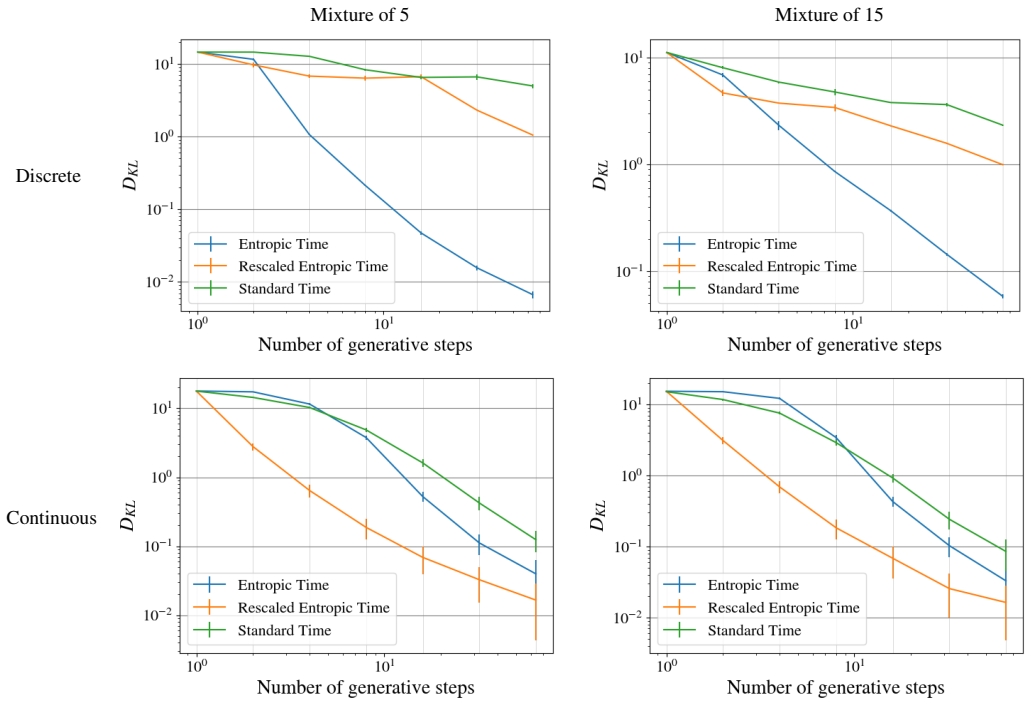



Figure 4: Kullback–Leibler divergence against the number of generative steps for different time parameterizations for mixture of data points (discrete) and Gaussians (continuous).

and the binned distribution ($p_{bin}(a_i)$ = probability of a generated sample ending up in the $i$-th bin). A variance-preserving SDE and EDM SDE were used for our experiments. Datapoints were randomly initialized and Kullback-Leibler divergence was estimated $10^2$ times using $10^4$ different paths, so mean and variance of KL estimate could be obtained.

For the continuous case, the performance was measured by estimating the SDE-generated distribution using Gaussian kernal density estimation (with a standard deviation of $10^{-2}$) and then evaluating the KL divergence using Monte-Carlo methods with $10^3$ samples. Similarly to the discrete case, the KL divergence was estimated $10^2$ times using $10^4$ different paths to estimate the SDE-generated distribution.

### F.2 IMAGENET

We used EDM2-S and EDM2-L models for ImageNet-64 (Karras et al. (2024), `https://github.com/NVlabs/edm2`). For generating samples, we used the stochastic and deterministic DDIM (Song et al., 2022). To compare performance between different runs we used the FID (Heusel et al., 2017) and FD-DINOv2 (Oquab et al., 2023) scores provided by Karras et al. (2024) implementation. We generated $50,000$ images and compared them with pre-computed reference statistics. Class labels were drawn from a uniform distribution.

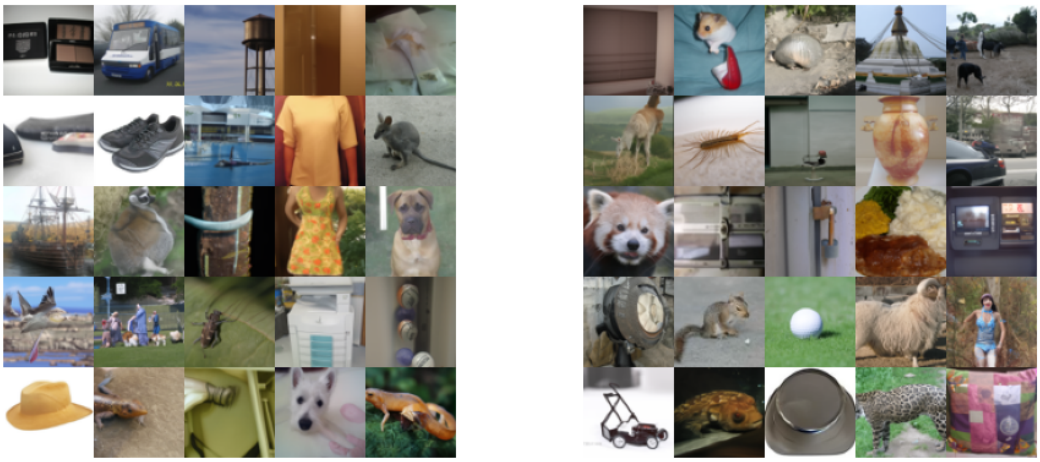
Entropy and rescaled entropy were calculated using an estimation of squared error using equation 13. The squared error was estimated at 128 time points according to the EDM schedule ($\rho = 7$, $\sigma_{min} = 0.002$, $\sigma_{max} = 80$) using the Monte-Carlo method with 1024 samples at each timestep. Entropy and rescaled entropy were calculated with both network sizes, $S$ and $L$, and there was no significant difference between them, as expected.

As already stated in the main text, the entropic time generated blurry images and was not used for the comparison in table 1. An example of images generated with the deterministic DDIM sampler using the entropic schedule over 64 steps, with the EDM2-L model, is given in figure 5. Examples of generated images using the EDM and rescaled entropy schedules are given in figures 6, 7, and 8.

Figure 5: Images generated with the deterministic DDIM sampler using the entropic schedule over 64 steps, with the EDM2-L model.

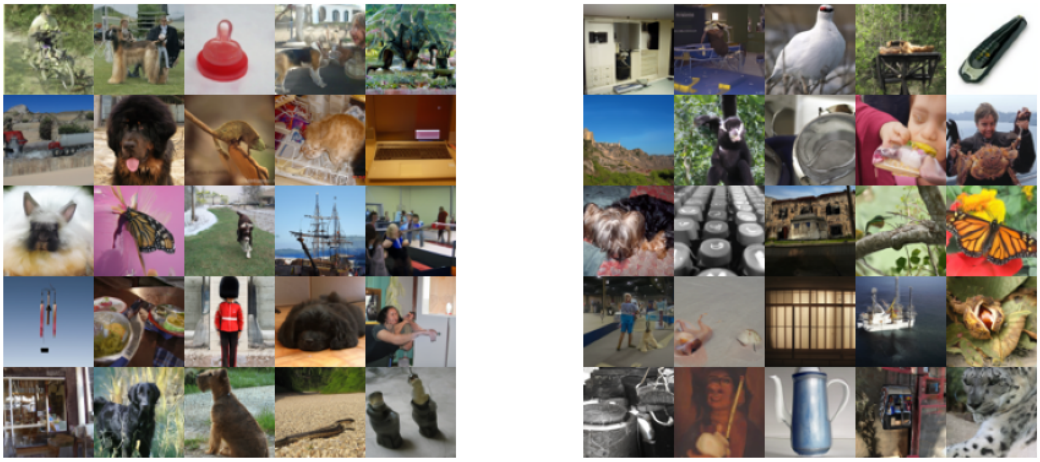
(a) EDM (b) Rescaled Entropy

Figure 6: Images generated with the stochastic DDIM sampler using the EDM schedule (left) and rescaled entropic schedule (right) over 64 steps, with the EDM2-S model.



(a) EDM (b) Rescaled Entropy

Figure 7: Images generated with the stochastic DDIM sampler using the EDM schedule (left) and rescaled entropic schedule (right) over 64 steps, with the EDM2-L model.



(a) EDM

(b) Rescaled Entropy

Figure 8: Images generated with the deterministic DDIM sampler using the EDM schedule (left) and rescaled entropic schedule (right) over 64 steps, with the EDM2-L model.