# Neural Entropy

**Akhil Premkumar**
Department of Applied Physics
Yale University
New Haven, CT 06511, USA
akhil.premkumar@yale.edu
*Work done while at the University of Chicago*

## Abstract

We explore the connection between deep learning and information theory through the paradigm of diffusion models. A diffusion model converts noise into structured data by reinstating, imperfectly, information that is erased when data was diffused to noise. This information is stored in a neural network during training. We quantify this information by introducing a measure called *neural entropy*, which is related to the total entropy produced by diffusion. Neural entropy is a function of not just the data distribution, but also the diffusive process itself. Measurements of neural entropy on a few simple image diffusion models reveal that they are extremely efficient at compressing large ensembles of structured data.

## 1 Introduction

How much information is stored in a neural network? As a simple example, consider training a neural network to store an 8-bit grayscale image of dimension $H \times W$ pixels. The network learns a smooth map from pixel co-ordinates to grayscale intensity values from $H \times W$ bytes of raw data. This is not the total number of bytes of the parameters that constitute the network, and not every image of size $H \times W$ contains the same amount of information. But it is reasonable to expect that if we push images of higher and higher resolutions/detail onto the same network, at some point the network will not be able to reproduce the images faithfully

The question is even more pertinent in the context of generative models. These models are capable of producing seemingly endless variations of the original training data, say images, but that does not mean the neural network has stored an infinite number of images. Rather, generative models store a *distribution* of images, call it $p_{\mathrm{d}}$, and the generated samples are points that interpolate the training data in $p_{\mathrm{d}}$. This is similar to how the network from the prior example blends the grayscale intensities between neighboring pixels. So the analogous question to ask is this: how many bytes of data is $p_{\mathrm{d}}$ worth? The primary goal of this paper is to answer this question in the context of diffusion-based generative models (hint: it is not simply the Shannon entropy of $p_{\mathrm{d}}$, see App. C.2).

Diffusion models serve as a natural bridge between information theory and machine learning, having been inspired by ideas from non-equilibrium thermodynamics [1], which itself can be viewed as an application of information-theoretic principles to physical systems [2–4]. Very briefly, samples from a training dataset are incrementally noised till they are distributed as a generic Gaussian, call it $p_{\mathrm{eq}}$, while a neural network learns to reverse these noising steps. Once trained, the network can transform a random Gaussian vector into a highly structured output that resembles a typical member of the training data. In the continuum limit, the noising and denoising stages become diffusive processes [5, 6], the thermodynamic properties of which are well established [7–9].

Diffusion gradually wipes out information from $p_{\mathrm{d}}$ over time (cf. Fig. 6). The information loss is quantified by the total entropy produced during the process, $S_{\mathrm{tot}}$. Within this framework, we can

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

define the information content of $p_{\rm d}$ in relation to the diffusion process itself—it is the amount of information that must be reinstated to drive the process away from its equilibrium state $p_{\rm eq}$ back to $p_{\rm d}$. It is precisely $S_{\rm tot}$ (cf. App. C). A well-trained diffusion model retains nearly all of this information in its neural network. Therefore, we can characterize the information content of the network by a quantity we call the *neural entropy*, $S_{\rm NN} \approx S_{\rm tot}$, defined in Eq. (18).

Before we delve into the details, a few points must be clarified. First, it is important to stress that neural entropy quantifies the information stored in a perfectly trained network; it is *not* the entropy of the phase space density over the neural network's internal microstates. Second, no diffusion model can reconstruct $p_{\rm d}$ perfectly because we only have access to a finite number of training samples from it [10], and training is imperfect even with a large dataset. Third, the neural network encodes and interpolates the given information, drawing from its own inductive biases to fill in the gaps between the training data [11]. This is why diffusion models are able to estimate very high-dimensional distributions even from relatively small datasets [12]. Consequently, neural entropy is just one part of a slew of variables, like the choice of network architecture, optimization algorithm, etc. that ultimately affect the overall model performance.

Despite these caveats, empirical measurements of neural entropy reveal interesting insights into the behavior of neural networks. First, in a setting where the inductive biases are relatively weak and the data distribution is largely unstructured (e.g. Gaussian mixtures), diffusion models tend to struggle to reconstitute $p_{\rm d}$ accurately as more information is fed into the network (see Fig. 10). Second, in image diffusion models with U-nets trained on real images, the neural entropy shows a distinct *logarithmic* scaling with the number of training samples $N$ (see Figs. 1 and 16). That is, the marginal information gained per sample decreases approximately as $1/N$. The quality of generated images also reflects this trend (see Fig. 18). Provided that $N$ is sufficiently large for the model to approximate $p_{\rm d}$ well, diffusion models compress the images with great efficiency, since they encode the ensemble statistics of the training data; storing each image separately in old-fashioned memory would have incurred a linear cost in $N$.

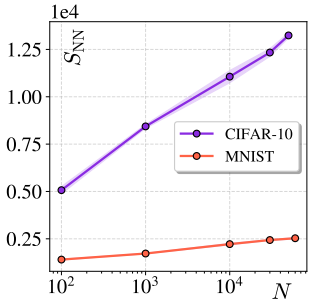



Figure 1: Neural entropy vs. number of samples for two image diffusion models.

## 2 Schrödinger's Gedankenexperiment

The link between diffusion and information theory can be traced back to a thought experiment introduced by Erwin Schrödinger in a seminal paper from 1931 [13]. Consider a diffusion process like the dissolution of an ink drop in water. Common experience suggests that the ink particles would homogenize over the available volume of water and remain in this diffused state indefinitely. However, there is a very small but non-zero probability that the ink particles collect together in some exotic configuration at a later time. Schrödinger asks: what is the probability that the particles diffuse back to their original state?

To answer this question in a simpler setting we study random walkers on a one-dimensional lattice. The lattice sites are spaced by $\ell$ and the walkers jump to one of their nearest neighbors at each time step. The density of walkers at $x$ updates as

$$p(x, t + \Delta t) = q_R(x - \ell)p(x - \ell, t) + q_L(x + \ell)p(x + \ell, t), \tag{1}$$

where $q_R(x)$ (or $q_L(x)$) is the probability that a walker at $x$ jumps to the right (or left) at time $\Delta t$ (see App. B.1 for notation). But that does not mean exactly $q_R(x)$ fraction of all walkers at $x$ always jump rightwards in $\Delta t$; over several trials, there will be small fluctuations in the actual number of walkers that make such a transition. Such fluctuations can accumulate to evolve $p$ in a manner different from Eq. (1), albeit with low probability. One may liken this to throwing a perfectly fair coin 1000 times: due to fluctuations, we do not always obtain the expected outcome of 500 heads and 500 tails, and in fact there is a minute probability of $2^{-1000}$ of obtaining all heads (or all tails).

For appropriate choices of $q_R(x)$ and $q_L(x)$, there exists an equilibrium distribution $p_{\rm eq}$ which satisfies the detailed balance conditions corresponding to Eq. (1). Let $T$ be a large enough time that

the walkers can equilibrate to very nearly $p_{\rm eq}$ from another state $p_{\rm d}$. Starting from $p_{\rm eq}$ at $t = 0$, the probability that the walkers would migrate back to the distribution $p_{\rm d}$ at time $t = T$ is [13, 14]

$$\mathcal{P}\left[p_{\rm d}\right] \propto \exp\left[-M \sum_{x_0, x_T} p_{\rm eq}(x_0) h(x_T|x_0) \log \frac{h(x_T|x_0)}{g(x_T|x_0)}\right] \equiv e^{-M \mathrm{D}_{KL}(h\|g)}. \tag{2}$$

Here, $M$ is the total number of walkers on the lattice and $g(x_T|x_0)$ is the probability that a walker at $x_0$ ends up at $x_T$ under Eq. (1). That is, $g$ is the transition kernel for Eq. (1). On the other hand $h(x_T|x_0)$ is a kernel that transports $p_{\rm eq}(x_0)$ to $p_{\rm d}(x_T)$. There are many kernels $h$ that accomplish this, but for sufficiently large $M$ the exponential in Eq. (2) picks out an optimal kernel $h_\star$ for which the Kullback-Leibler divergence $\mathrm{D}_{KL}(h_\star\|g)$ is minimum. It can be shown that the evolution of $p_{\rm eq} \to p_{\rm d}$ under $h_\star$ is a reversal (playback) of the transformation $p_{\rm d} \to p_{\rm eq}$ under Eq. (1) [14, 15].

With $p_{\rm eq}$ (or $g$) fixed, $\mathcal{P}$ can be understood as a *distribution of distributions*. A sample from $\mathcal{P}$ is a distribution that $p_{\rm eq}$ can fluctuate into at time $t = T$, under the dynamics in Eq. (1). If $M$ is large, $\mathcal{P}$ is sharply peaked at $p_{\rm eq}$; the probability that the walkers would deviate from this configuration is exponentially small. This is true even with $h_\star$—natural processes have a preferred direction of time, and they rarely evolve in reverse. Eq. (2) is intimately related to the Second Law of Thermodynamics. In fact,

$$S_{\rm tot} := \sum_{t=0}^{T-\Delta t} \sum_{x_{t+\Delta t}, x_t} p(x_t, t) h_\star(x_{t+\Delta t}|x_t) \log \frac{h_\star(x_{t+\Delta t}|x_t)}{g(x_{t+\Delta t}|x_t)} \geq \mathrm{D}_{KL}(h_\star\|g). \tag{3}$$

where $p(x_t, t)$ is the distribution of walkers as they evolve between $p_{\rm eq}$ and $p_{\rm d}$, and $S_{\rm tot}$ is the total entropy generated if $p_{\rm d}$ was subjected to Eq. (1) for time $T$ (cf. App. A.2). In simple terms, $S_{\rm tot}$ quantifies the time irreversibility of the process $p_{\rm d} \to p_{\rm eq}$ [9, 16, 17]. We discuss the meaning of $S_{\rm tot}$ in greater detail in the upcoming sections and App. C.

Combining Eqs. (2) and (3), we obtain a key relation between the Shannon information content of the outcome $p_{\rm d}$, per walker, and the total entropy [2, 18, 19]:

$$\frac{S_{\rm tot}}{\log 2} \geq -\frac{1}{M} \log_2 \mathcal{P}\left[p_{\rm d}\right]. \tag{4}$$

If we observe a set of random walkers that was initially at equilibrium and find that they are still distributed as $p_{\rm eq}$ we learn nothing new; that was the outcome we expected. However, in the unlikely event that we observe the random walkers distributed as $p_{\rm d}$, we would gain an amount of information commensurate with the total entropy generated in diffusing $p_{\rm d} \to p_{\rm eq}$.

## 3 Diffusion models and Maxwell's demon

We can enhance the probability of obtaining the outcome $p_{\rm d}$ by bringing $g$ closer to $h_\star$. That is, we adjust the jump probabilities in Eq. (1) such that the distribution of walkers evolves to $p_{\rm d}$ after time $T$. The modified dynamics reshapes the distribution $\mathcal{P}$ to be peaked around $p_{\rm d}$ rather than $p_{\rm eq}$

To see how this is implemented in a diffusion model we convert the discrete random walker setup from Eq. (1) to a continuous diffusion process by making the lattice spacing $\ell$ small. Taylor expanding in $\ell$ and keeping the leading terms, we obtain the Fokker-Planck equation (see App. A)

$$\partial_t p(x,t) = -\partial_x(b_+(x)p(x,t)) + \frac{\sigma^2}{2}\partial_x^2 p(x,t), \tag{5}$$

$$b_+(x) := \frac{\ell}{\Delta t}(q_R(x) - q_L(x)), \quad \sigma^2 := \frac{\ell^2}{\Delta t}. \tag{6}$$

We restrict ourselves to drift terms $b_+(x)$ that are confining so that $p_{\rm eq}(x) \propto \exp(\int^x 2b_+/\sigma^2)$ exists. By explicit calculation of Eq. (3), it can be shown that if a distribution $p_{\rm d}$ is subjected to Eq. (5) the total entropy produced after time $T$ is (cf. Eq. (39) and [9])

$$S_{\rm tot} = \int_0^T \mathrm{d}t \frac{\sigma^2}{2} \mathbb{E}_{p(\cdot,t)}\left[\|\partial_x \log p_{\rm eq} - \partial_x \log p\|^2\right], \tag{7}$$

where the expectation value is taken over the distribution $p$ that interpolates $p_{\rm d}$ and $p_{\rm eq}$. The r.h.s. in Eq. (7) is the KL divergence between path measures of two stochastic differential equations (SDEs),

$$\mathrm{d}X_t = -(b_+(X_t) - \sigma^2 \partial_x \log p(X_t, t))\mathrm{d}t + \sigma \mathrm{d}B_t, \tag{8a}$$

$$\mathrm{d}X_t = b_+(X_t)\mathrm{d}t + \sigma \mathrm{d}B_t, \tag{8b}$$

upto a boundary term that vanishes when $T$ is large [20]. Eqs. (8a) and (8b) that correspond to the transition kernels $h_\star$ and $g$ respectively (cf. App. A.1). That is, if we reset the clock to $t = 0$ and apply Eq. (8a) for a time $T$ we can drive $p_{\rm eq}$ back to $p_{\rm d}$ along $p$. Bringing Eq. (8b) closer to Eq. (8a) would concentrate $\mathcal{P}$ around $p_{\rm d}$. In a diffusion model this can be done by changing Eq. (8b) to

$$\mathrm{d}X_t = (b_+(X_t) + \sigma^2 \boldsymbol{e_\theta}(X_t, t))\mathrm{d}t + \sigma \mathrm{d}B_t, \tag{9}$$

where $\boldsymbol{e_\theta}(X_t, t)$ is the output of a neural network trained to minimize an equivalent of (cf. Eq. (90))

$$\mathcal{L}_{\rm EM} := \int_0^T \mathrm{d}t \frac{\sigma^2}{2} \mathbb{E}_p \left[ \| \partial_x \log p_{\rm eq} - \partial_x \log p + \boldsymbol{e_\theta} \|^2 \right]. \tag{10}$$

It follows from Eq. (4) that, a perfectly trained network stores *at least* the same amount of information as we would learn from observing $p_{\rm eq}$ fluctuate to $p_{\rm d}$ under Eq. (8b). In practice, training is not perfect, so the information absorbed by the network is not exactly $S_{\rm tot}$, as we discuss below. We define the *ideal* neural entropy as the information retained by the network under perfect training,

$$\hat{S}_{\rm NN} := S_{\rm tot}. \tag{11}$$

This discussion is reminiscent of Maxwell's demon, a famous thought experiment from physics [21, 22]. The crucial difference is that the demon does not perform work on the system; it measures the state of the system to make decisions about closing doors or adjusting potentials [23]. Diffusion models *do* expend work to reconstitute $p_{\rm d}$ from $p_{\rm eq}$, through the modified drift term in Eq. (9). The additional $\sigma^2 \boldsymbol{e_\theta}$ term reshapes the *free energy* landscape to make $p_{\rm d}$ the most probable outcome (cf. App. C.2). But these models also measure and store state information from simulations of $p_{\rm d} \to p_{\rm eq}$ during training.

A true Maxwell's demon would reverse diffusion by waiting for $p_{\rm eq}$ to fluctuate into $p_{\rm d}$, an event it learns about by measurement, and switch up the potential to lock $p_{\rm d}$ into place. This is an example of an 'information ratchet' [24, 25]. On the other hand, a diffusion model remembers $p_{\rm d}$ in a manner closer to how we store, say, an image in memory. A grayscale image of dimensions $H \times W$ is a sample from a uniform probability distribution over the hypercube $[0, 255]^{H \times W}$. The information gained from observing any sample is $\log_2(256)^{H \times W} = H \times W$ bytes. This is also the amount of information we need to specify to locate a specific sample/image in the hypercube. In the same way, $S_{\rm tot}$ is the information required to locate within the paths generated by Eq. (8b) a set of paths that transport $p_{\rm eq} \to p_{\rm d}$.

## 4 Entropy matching

Having introduced the total entropy $S_{\rm tot}$ in the context of random walkers on a lattice, we can generalize it to a $D$-dimensional continuous diffusion process with little effort. We will make the drift and diffusion diffusion coefficients time-dependent, but keep the latter isotropic. That is, $p_{\rm d}$ diffuses under

$$\mathrm{d}Y_s = b_+(Y_s, s)\mathrm{d}s + \sigma(s)\mathrm{d}\hat{B}_s, \tag{12}$$

where we have introduced a new time variable $s := T - t$ for the 'forward' evolution (see Fig. 6). Let $p_0$ be the result of evolving $p_{\rm d}$ for a time $T$ with Eq. (12). The SDEs from Eq. (8) are updated to

$$\mathrm{d}X_t = -(b_+(X_t, T-t) - \sigma(T-t)^2 \nabla \log p(X_t, t))\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t, \tag{13a}$$

$$\mathrm{d}X_t = b_+(X_t, T-t)\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t, \tag{13b}$$

There is no longer a static equilibrium state since Eq. (13b) changes over time; if we start with $p_0$ at time $t = 0$ and evolve under Eq. (13b) we will obtain a distribution different from $p_0$, which we denote as $p_{b_+}$, that depends on $p_0$ as well as Eq. (13b). But it is useful to define the *quasi-invariant* distribution, $p_{\rm eq}^{(t)}(x)$, which satisfies the homogeneous Fokker-Planck equation

$$0 = -\nabla \cdot (b_+(x, T-t)p_{\rm eq}^{(t)}) + \frac{1}{2}\sigma(T-t)^2 \nabla^2 p_{\rm eq}^{(t)} \implies p_{\rm eq}^{(t)}(x) = \frac{1}{Z_t} \exp\left[ \int^x \mathrm{d}\bar{x} \frac{2b_+}{\sigma^2} \right]. \tag{14}$$

Intuitively, $p_{\rm eq}^{(t)}$ can be understood as the 'least informative state' at time $t$. It is the distribution that would result if we froze $b_+$ and $\sigma$ at their values at $t$ and waited for the system to equilibrate. Therefore $p_{\rm eq}^{(t)}$ depends only on the drift and diffusion coefficients at $t$ and has no memory of the initial state. For example, if $b_+ = -(x-t)$ and $\sigma = 1$ the quasi-invariant state would be $p_{\rm eq}^{(t)} \propto \exp(-(x-t)^2)$ [7]. Thus, $p_{\rm eq}^{(t)}$ is the natural generalization of $p_{\rm eq}$ for time-dependent dynamics. In this paper, we restrict ourselves to forward processes for which the drift and diffusion coefficients have the same time-dependence, that is, $b_+/\sigma^2 = \text{const}$. In that case, it can be shown that

$$S_{\rm tot} \equiv \int_0^T {\rm d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \nabla \log p_{\rm eq}^{(t)} - \nabla \log p \right\|^2 \right] = {\rm D}_{KL}\left( p_{\rm d} \middle\| p_{\rm eq}^{(T)} \right) - {\rm D}_{KL}\left( p_0 \middle\| p_{\rm eq}^{(0)} \right). \tag{15}$$

This relation is derived in App. B.3. It bears a strong likeness to an important result in thermodynamics called the *Jarzynski equality* [26], specifically the form given in [27]. According to this relation, total entropy is the information gap between $p$ and the maximally ignorant state at a given $T$. Further discussion of the connection to thermodynamics is given in App. C.1. Our main goal is to understand the consequences of Eq. (15) to diffusion models.

Replacing $p_{\rm eq}^{(t)}$ in Eq. (15) with $p_{b_+}$ turns it into an inequality,

$$S_{\rm tot} \equiv \int_0^T {\rm d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \nabla \log p_{\rm eq}^{(t)} - \nabla \log p \right\|^2 \right] \geq {\rm D}_{KL}\left( p_{\rm d} \middle\| p_{b_+} \right) \tag{16}$$

This is a slight variation of Theorem 1 from [20]. A detailed proof is given in App. B.2. If we modify the drift term in Eq. (13b) to $b_+ + \sigma^2 \boldsymbol{e_\theta}$ as we did in Eq. (9), Eq. (16) changes to (cf. Eq. (47))

$$\int_0^T {\rm d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \nabla \log p_{\rm eq}^{(t)} - \nabla \log p + \boldsymbol{e_\theta} \right\|^2 \right] \geq {\rm D}_{KL}\left( p_{\rm d}(\cdot) \| p_{\boldsymbol{\theta}}(\cdot, T) \right). \tag{17}$$

The l.h.s. is the training objective, $\mathcal{L}_{\rm EM}$, the minimization of which can now be seen as tightening the KL divergence between true $p_{\rm d}$ and the reconstructed distribution $p_{\boldsymbol{\theta}}$. We call this the *entropy-matching* objective. It is nearly the same as the flow-matching objective from [28], except for the factors multiplying the expectation value.

The neural entropy defined in Eq. (11) is not always the true measure of the information stored in the network. This is often beneficial; if the neural network stored $S_{\rm tot}$ perfectly for a relatively sparse dataset, such as images, the diffusion model would learn to reconstruct a series of Dirac delta functions in pixel space. We propose that the actual value of neural entropy is estimated by

$$\boxed{S_{\rm NN} := \int_0^T {\rm d}s\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \boldsymbol{e_\theta} \right\|^2 \right].} \tag{18}$$

The time integral has been expressed in terms of $s$ here because entropy is produced in the $s$-direction. Practically Eq. (18) is computed by simulating the forward process Eq. (12) and taking the Monte Carlo average (cf. Eq. (23)). So the expectation is still taken with respect to the ideal reverse evolution.

A relation analogous to Eq. (17) can be derived for score-matching diffusion models by switching the drift term in Eq. (13b) to $-b_+ - \sigma^2 \boldsymbol{s_\theta}$ (cf. Eq. (46)),

$$\int_0^T {\rm d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \boldsymbol{s_\theta} - \nabla \log p \right\|^2 \right] \geq {\rm D}_{KL}\left( p_{\rm d}(\cdot) \| p_{\boldsymbol{\theta}}(\cdot, T) \right). \tag{19}$$

However, setting $\boldsymbol{s_\theta} = 0$ on the l.h.s. does *not* give us a term that can be interpreted sensibly as an entropy. For example, if we consider the special case where $b_+$, $\sigma$ are time-independent and choose $p_{\rm d} = p_{\rm eq}$, we see that $S_{\rm tot}$ vanishes and no information would be stored in the neural network in an entropy-matching model. However, $\mathbb{E}[\|\nabla \log p\|^2] \neq 0$ since the score function is non-zero over the support of $p_{\rm eq}$, so the network ends up having to store 'information' to convert $p_{\rm eq}$ to itself! Comparing Eqs. (17) and (19) we see that setting $\boldsymbol{s_\theta} = \nabla \log p_{\rm eq}^{(t)} + \boldsymbol{e_\theta}$ makes both approaches equivalent, in principle. However, the score-matching network must put additional effort into learning the quasi-invariant distribution, which complicates the interpretation of score-matching loss as an entropy. See App. D for further discussion.

## 5 Thermodynamic uncertainty

Returning for a moment to the random walkers on a discrete lattice, it is apparent that the walkers are less likely to fluctuate into a $p_{\rm d}$ that is far different from $p_{\rm eq}$, compared to one that is more similar to $p_{\rm eq}$. This is manifest from Eqs. (2) and (4): a larger KL between $p_{\rm d}$ and $p_{\rm eq}$, which is $S_{\rm tot}$, suppresses $\mathcal{P}[p_{\rm d}]$ further. In practice $p_{\rm d}$ is often fixed by the training data and $p_{\rm eq}^{(t)}$ changes as we adjust the drift and diffusion coefficients in the forward process, Eq. (12), to speed up the generative process. There is great interest in straightening the trajectories from the Probability Flow (PF) ODE [6] by clever choices of $b_+$ and $\sigma$, to enable few-shot sampling during the generative stage [29–31]. However, such forward processes often produce more entropy, which means these models may inadvertently be placing a higher information load on the neural network.

As an illustrative example, consider the Straight Line Diffusion Model (SL) introduced in [31]. The forward process is

$$\mathrm{d}Y_s = -\frac{1}{1-s}Y_s + \sqrt{\frac{2}{1-s}}\sigma_0 \mathrm{d}B_s. \tag{20}$$

At an intermediate time $s \in (0,T)$ (with $T=1$), a sample $y_{\rm d} \sim p_{\rm d}$ is propagated to

$$y_s = (1-s)y_{\rm d} + \sigma_0\sqrt{1-(1-s)^2}\epsilon, \tag{21}$$

where $\epsilon \sim \mathcal{N}(0, \mathbb{1}_D)$. For small $\sigma_0$ and a fairly 'wide' $p_{\rm d}$, the trajectories in Eq. (21) are nearly straight lines that land at $y_T \sim \mathcal{N}(0, \sigma_0^2\mathbb{1}_D)$. This is a result of allowing the drift term to dominate the noise in Eq. (20). But that also makes $p_{\rm eq}^{(t)} \propto \exp(-y^2/\sigma_0^2)$ a very narrow Gaussian, which increases the KL to $p_{\rm d}$ and thereby $S_{\rm tot}$. Or, using the intuition from Sec. 2, decreasing the randomness in the diffusion process diminishes the chances of an automatic fluctuation into $p_{\rm d}$. Using a forward process with more noise would lower the entropy, but only to a certain extent. If the $\sigma$ is too large $p_{\rm eq}^{(t)}$ becomes too wide compared to $p_{\rm d}$ and $S_{\rm tot}$ rises again.

The above discussion is meant to highlight that $S_{\rm tot}$ depends on the forward diffusion process and $p_{\rm d}$ in a non-trivial way, and that there might be an optimal process that produces the least entropy for a given $p_{\rm d}$. This intuition is made more precise by the *thermodynamic uncertainty relation*, which relates the total entropy produced to the $L^2$-Wasserstein distance between $p_{\rm d}$ and $p_0$ [32, 33],

$$S_{\rm tot} \times \sigma^2 T \geq \frac{1}{2}\mathcal{W}_2(p_{\rm d}, p_0)^2. \tag{22}$$

Here $\sigma$ is assumed to be a constant for simplicity and $\sigma^2 T$ is the time it takes for $p_{\rm d}$ to reach $p_0$, measured in units of $\sigma^{-2}$. The $\mathcal{W}_2$ depends only on the initial and final distributions. If two processes take the same time to equilibrate (reach $p_0 \approx p_{\rm eq}$), the one whose equilibrium state is farther from $p_{\rm d}$ will generate more entropy. If two process transform $p_{\rm d}$ to the same $p_0$, the $\mathcal{W}_2$ is the same in both cases, but the faster transformation will produce more entropy to satisfy the bound. Therefore a diffusion model must store more information to reverse a faster diffusion process. This is the thermodynamic speed limit: given $p_{\rm d}$ and $p_0$, there is an upper limit to how fast we can diffuse one to the other without exceeding a specific entropy production budget. Equivalently, a faster transformation requires a greater amount of information to reverse, which has been found to affect accuracy [34]. These observations are also confirmed in our experiments.

## 6 Experiments

Neural entropy, as defined in Eq. (11), quantifies the information presented to the network in an idealized setting. In practice, the finiteness of the data, imperfections in training, and strong inductive biases of the network all affect the amount of information stored in the neural network. To address these points we will perform two broad classes of experiments, first to probe the transport properties of diffusion discussed in Eq. (22), and second, to study the storage efficiency of diffusion models.

**Transport experiments** We work with synthetic datasets sampled from simple multivariate distributions for which we have closed-form expressions for both $p_{\rm d}$ and $\nabla \log p$ (e.g. Gaussian mixtures). This allows us to produce as many samples as we require with high fidelity, compute their exact log densities, and work in arbitrary dimensions. Recall from Eq. (17) that the loss function upper bounds the KL divergence between the data distribution and the generated distribution, $\mathrm{D}_{KL}(p_{\rm d}(\cdot)\|p_{\boldsymbol{\theta}}(\cdot, T))$,

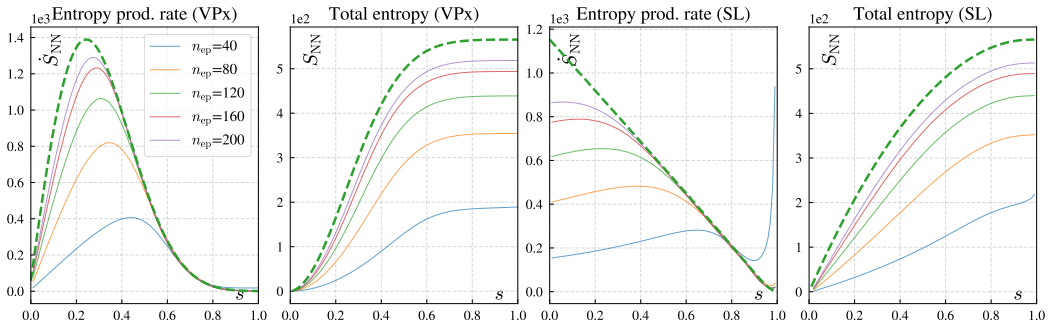



Figure 2: Entropy production rate and total entropy as $p_{\rm d}$ is diffused to $p_0$ by the VPx and SL processes from Eq. (24) and Eq. (20) respectively. The dashed lines are the ideal curves for $\dot{S}_{\rm tot}$ and $S_{\rm tot}$, while the solid lines are $\dot{S}_{\rm NN}$ and $S_{\rm NN}$ at the end of the $n_{\rm ep}$-th training epoch.

which means this KL can be used to assess the performance of the diffusion model—a smaller KL implies the model reproduces $p_{\rm d}$ more faithfully. For any sample $x$ we know $p_{\rm d}(x)$, so all we need to compute the KL is $\log p_{\boldsymbol{\theta}}(x, T)$. The latter is approximated by the method discussed in App. E.2. Finally, the transport experiments will be carried out with diffusion models with a multi-layer perceptron (MLP) core. Since the inductive biases in such fully connected networks are weak [35], these models enable us to isolate the effects of varying levels of neural entropy on the model performance.

If $p_{\rm d}$ is a mixture of Gaussians it is possible to compute Eq. (11) explicitly, which gives the ideal value of neural entropy. However, due to the imperfections mentioned above the actual neural entropy is given by Eq. (18). Both these expressions are computed by Monte Carlo averages, and it is useful to examine how they change over time $s \in (0, T]$. For example, with some new samples $\tilde{y}_{\rm d} \sim p_{\rm d}$,

$$S_{\rm NN}(s) = \int_0^s {\rm d}\bar{s}\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \|\boldsymbol{e}_{\boldsymbol{\theta}}\|^2 \right] \approx s\, \mathbb{E}_{\tilde{y}_{\rm d} \sim p_{\rm d}} \mathbb{E}_{\bar{s} \sim \mathcal{U}(0,s)} \left[ \frac{\sigma(\bar{s})^2}{2} \mathbb{E}_{y_{\bar{s}} \sim p(y_{\bar{s}}|\tilde{y}_{\rm d})} \left[ \|\boldsymbol{e}_{\boldsymbol{\theta}}(y_{\bar{s}}, \bar{s})\|^2 \right] \right], \tag{23}$$

To explore the considerations raised in Sec. 5 more thoroughly, we experiment with a few different diffusion processes. We introduce a minor generalization of the Variance Preserving (VP) process, given by the SDE

$${\rm d}Y_s = -\frac{\beta(s)}{2} Y_s\, {\rm d}s + \kappa \sqrt{\beta(s)} {\rm d}B_s, \tag{24}$$

which we shall henceforth refer to as VPx. For $\kappa = 1$ this is the same as the VP process from [1, 6, 29]. If we set $\kappa = \sigma_0$ we obtain a process that has the same quasi-invariant distribution as Eq. (20). However, the trajectories generated by Eq. (24) are

$$y_s = e^{-\frac{1}{2}\int_0^s \beta(\bar{s}){\rm d}\bar{s}} y_{\rm d} + \kappa \sqrt{1 - e^{-\int_0^s \beta(\bar{s}){\rm d}\bar{s}}}\, \epsilon, \tag{25}$$

which 'forgets' $y_{\rm d}$ at an exponential rate as opposed to the linear evolution in Eq. (21). This difference is borne out in their respective entropy profiles—plots of the entropy production rate $\dot{S}_{\rm tot} \equiv {\rm d}S_{\rm tot}/{\rm d}s$, and the total entropy $S_{\rm tot}$, over time. These are the dashed curves in Fig. 2. The final value of $S_{\rm tot}$ is very nearly the same for both processes when $\kappa = \sigma_0$ since they have a common $p^{(t)}_{\rm eq}$ (cf. Eq. (16)), with small discrepancies arising from differences in their respective $p_0$. The solid lines in Fig. 2 are $\dot{S}_{\rm NN}$ and $S_{\rm NN}$, evaluated at various stages of training. As we train over more epochs, $n_{\rm ep}$, the network absorbs more information, bringing $S_{\rm NN}$ closer to its ideal value $S_{\rm tot}$.

The experiments in Fig. 2 were carried out on a model trained on $N = 8192$ samples from a mixture of five Gaussians in $D = 6$ dimensions, which is $p_{\rm d}$. The Gaussian components were $\mathcal{N}(\bar{x}_r, \mathbb{1}_D)$, where the means $\bar{x}_r$ were randomly chosen from a $D$-hypercube of side length 4 centered at the origin. We used $\kappa = \sigma_0 = 0.1$, so both processes transform $p_{\rm d}$ to $p_0 \approx \mathcal{N}(0, 10^{-2}\mathbb{1}_D)$ at $T = 1$ (cf. Eq. (40)). The SL process has a singularity at $T = 1$ exactly (cf. Eq. (20)); this divergence manifests in the $\dot{S}_{\rm NN}$ curve for SL at the early stages of training.

In both VPx and SL dynamics, the combination of weak noise with a relatively strong drift subdues the randomness in the diffusion process, leading to larger $S_{\rm tot}$ (cf. Sec. 5). Equivalently, the $p_{\rm d}$ used above

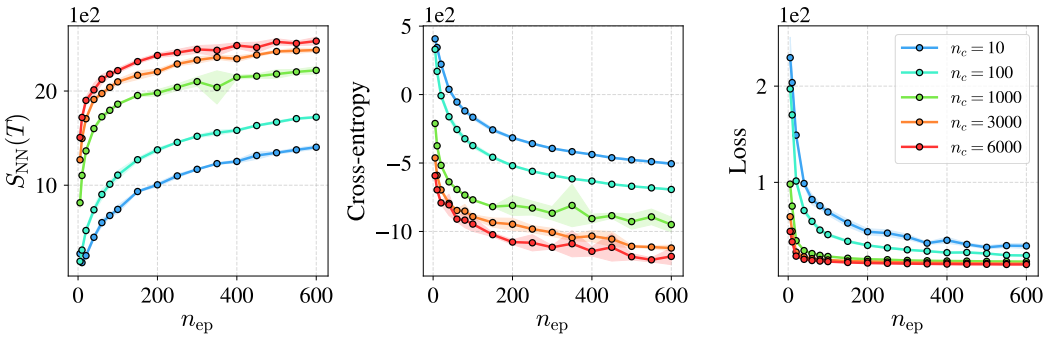



Figure 3: The evolution of neural entropy, cross-entropy, and loss over training epochs for an unconditional image diffusion model (VP) trained on the MNIST dataset. The different colors correspond to models trained on $n_c$ number of samples per class; $n_c = 6000$ means the model was trained on the entire dataset. The growth in neural entropy with the number of samples is nearly logarithmic. The values of $S_{\rm NN}(T)$ at the end of training are shown in Fig. 16.

is too far from their equilibrium state so a larger amount of information must be supplied to transform $p_{\rm eq} \approx p_0$ to $p_{\rm d}$. On the other hand, if we use a regular VP process, for which $p_{\rm eq} = \mathcal{N}(0, \mathbb{1}_{\rm D})$, the 'distance' to $p_{\rm d}$ is smaller and so is the total entropy. As a result, the VP model trains much faster and produces a more accurate reconstruction of $p_{\rm d}$ than the VPx or SL model. This is shown in Fig. 10.

**Storage experiments** We carry out similar experiments on a simple image diffusion model with a U-net core, trained on the MNIST dataset without class conditioning [36]. In this instance, the training dataset is small relative to the dimensionality of pixel space. Therefore the model relies on the inductive biases of the network to generalize from the given data points rather than memorize them [12]. Entropy curves for this dataset also show a sharp peak in entropy production near $s = 0$ (see Fig. 12). This is because the images live on a manifold $\mathcal{M}_{\rm d}$ of much lower dimensionality compared to the ambient 784-dimensional space. Seen from the $t$-direction, the sharp rise in entropy as $t \to T$ tells us that the diffusion model needs to inject a lot of information in the final few time steps to locate the sample precisely on $\mathcal{M}_{\rm d}$. Similar plots for the SL model are given in App. E.

In Fig. 3 we train the image model on the first $n_c$ samples from each class, $N = 10n_c$ samples in total, and measure $S_{\rm NN}$ and the cross entropy $\mathbb{E}_{p_{\rm d}}[-\log p_{\boldsymbol{\theta}}]$ as training progresses. We cannot compute the true $S_{\rm tot}$ or KL since the exact log densities are not known. Nonetheless, we see a similar result as before: the neural entropy and cross-entropy saturates after the model trains for a while. Importantly, the model absorbs more information if it is presented with a larger number of samples, but the growth in $S_{\rm NN}(T)$ with $N$ is not linear, it appears to be *logarithmic* (see Figs. 1 and 16).

We obtain the same behavior from a diffusion model trained on the CIFAR-10 dataset [37]. This time we use a U-net with self-attention layers [5, 38] and apply class conditioning. The image quality improves substantially with $N$ but neural entropy scales as nearly $\log N$ (see Fig. 18). Notably, such a trend is absent from the Gaussian mixture experiments performed on an MLP-based diffusion model (see Figs. 4 and 15). At low $N$ these models concentrate the probability mass around the sparse data points and learn a very different distribution from the true $p_{\rm d}$. This is why $S_{\rm NN}$ is larger at small $N$; it could also be larger than $S_{\rm tot}$ produced by diffusion of the actual $p_{\rm d}$. On the other hand, the combination of structured data and well-matched inductive biases in the image models saves us from overfitting even when data density is low, while also compressing the details from additional samples efficiently.

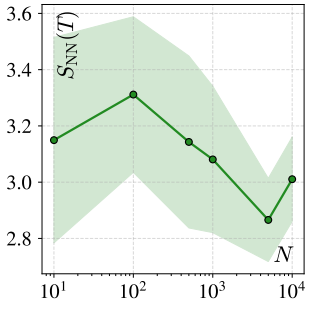



Figure 4: Neural entropy vs. number of samples for a diffusion model with an MLP trained on Gaussian mixtures.

## 7 Conclusion

In Sec. 1 we used the example of storing images to motivate neural entropy. We presented two scenarios: the storage of individual images versus the retention of a distribution of images. In a conventional memory the number of bytes required to store $N$ images will be $N$ times the bytes per image, even with efficient compression. Classical compression algorithms make implicit assumptions about the statistical properties of natural images and apply them universally to all inputs. However a diffusion model sees a large ensemble of images and can *learn* non-local correlations between the pixels of similar groups of images; they are infinitely deep autoencoders [39–41]. In other words, these models can tailor a compression scheme specific to the properties of the data distribution. This is a lossy form of compression of course, since we rarely recover the training images perfectly from these probabilistic models.

How about other generative models? At a conceptual level, Schrödinger's argument can be extended beyond just diffusion. For instance, a very simple-minded approach to creating a sentence of length $L$ would be to choose $L$ words at *random* from a corpus. The vast majority of sentences produced by this process would be utterly meaningless, but once in a while, we get a rare fluctuation that counts as a sensible statement. The collection of all such sentences constitutes an island of meaning $p_{\rm d}$, in a sea of non-sense $p_{\rm eq}$. Here $p_{\rm eq}$ can be a uniform distribution over all $L$-word combinations, say. $\mathcal{P}$ is the distribution of all distributions on these combinations. The information needed to transform $p_{\rm eq} \to p_{\rm d}$ is proportional to the negative logarithm of the probability of fluctuating into $p_{\rm d}$ automatically. A mechanism that enhances this probability to near certainty would need to store that much information to effect the transformation. If we start with a $p_{\rm eq}$ that assigns a greater probability to more frequently used words it would be easier, albeit still very improbable, to auto-fluctuate to $p_{\rm d}$. This is a manifestation of thermodynamic uncertainty, for words.

Diffusion-based LLMs [42, 43] offer the most direct path to defining neural entropy in the context of language. They are based on a discrete diffusion process very similar to the lattice random walks we considered in Secs. 2 and 3. The ideal neural entropy in that case should have a form similar to Eq. (37), with transitions that extend beyond nearest neighbor jumps. The discrete analog of the thermodynamic uncertainty relation, Eq. (22), is discussed in [33]. It would be interesting to investigate how the choice of the forward process in diffusion LLMs affects the training efficiency and model performance. More work will be needed to extend the entropic picture to transformer models, but investigations into the compressive abilities of such models are underway [44].

**Limitations** Our definition of neural entropy is limited to continuous diffusion models at present. In App. D we also point out the difficulties in defining neural entropy with score-matching models. With respect to the experimental results, the logarithmic growth of neural entropy with $N$ is stated as an empirical observation with little explanation. A deeper investigation of this phenomenon is relegated to future work. In particular, it would be interesting to check whether this trend is repeated in more sophisticated network architectures like diffusion transformers [45], or if there is a connection to the scaling laws [46]. As noted in Figs. 12 and 17 the calculation of neural entropy in the image models requires some care due to divergent entropy production near $s = 0$. This behooves us to investigate the neural entropy in diffusion processes with momentum components which soften such singular behavior [47, 48].

**Related work** The original work that introduced diffusion models took inspiration from the Jarzynski equality and fluctuation theorem from non-equilibrium thermodynamics [1, 49, 50]. The relation between these ideas becomes apparent through [40, 7], both of which use the Feynman-Kac formula and Girsanov's theorem to develop similar results separately for diffusion models and thermodynamics. These results can also be understood in the language of stochastic optimal control [15, 51]. We illustrate both approaches in App. B.4. Several authors have also developed the connection between these models and the Schrödinger's bridge problem [52, 53]. The consequences of the thermodynamic speed limit on diffusion model accuracy are studied in greater detail in [34]. Information-theoretic aspects of diffusion models are also explored in [54], and a deeper discussion of their ability to capture mutual information is given in [55]. It has also been noted that diffusion models are a form of energy-based memory [56], which concurs with the discussion in Sec. 4 and App. C.2, since the generative process in these models is a descent along a learned free energy landscape.

## Acknowledgments and Disclosure of Funding

We thank Lorenzo Orecchia, William Cottrell, Austin Joyce, Maurice Weiler, and Ryan Robinett for valuable discussions, and Yuji Hirono for bringing [34] to our attention. We are especially grateful to William Cottrell and Lorenzo Orecchia for their comments on an earlier version of this paper. The author was supported in part by the Kavli Institute for Cosmological Physics at the University of Chicago through an endowment from the Kavli Foundation. Computational resources for this project were made available through the AI+Science research funding from the Data Science Institute at the University of Chicago. Code is available at https://github.com/akhilprem1/NeuralEntropy

## References


[1] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html. 1, 7, 9, 28

[2] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x. 1, 3

[3] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957. doi: 10.1103/PhysRev.106.620. URL https://link.aps.org/doi/10.1103/PhysRev.106.620. 26, 27

[4] E. T. Jaynes. Gibbs vs Boltzmann Entropies. *American Journal of Physics*, 33(5):391–398, 05 1965. ISSN 0002-9505. doi: 10.1119/1.1971557. URL https://doi.org/10.1119/1.1971557. 1, 26

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html. 1, 8, 22, 31

[6] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS. 1, 6, 7, 22, 28, 32, 35

[7] Hao Ge and Da-Quan Jiang. Generalized jarzynski's equality of inhomogeneous multidimensional diffusion processes. *Journal of Statistical Physics*, 131(4):675–689, 5 2008. ISSN 1572-9613. doi: 10.1007/s10955-008-9520-4. URL https://doi.org/10.1007/s10955-008-9520-4. 1, 5, 9, 22

[8] Raphaël Chetrite and Krzysztof Gawędzki. Fluctuation relations for diffusion processes. *Communications in Mathematical Physics*, 282(2):469–518, 2008. ISSN 1432-0916. doi: 10.1007/s00220-008-0502-9. URL https://doi.org/10.1007/s00220-008-0502-9.

[9] Udo Seifert. Entropy production along a stochastic trajectory and an integral fluctuation theorem. *Phys. Rev. Lett.*, 95:040602, Jul 2005. doi: 10.1103/PhysRevLett.95.040602. URL https://link.aps.org/doi/10.1103/PhysRevLett.95.040602. 1, 3, 23, 26

[10] William Bialek, Curtis G. Callan, and Steven P. Strong. Field theories for learning probability distributions. *Phys. Rev. Lett.*, 77:4693–4697, Dec 1996. doi: 10.1103/PhysRevLett.77.4693. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.4693. 2

[11] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/322f62469c5e3c7dc3e58f5a4d1ea399-Paper.pdf. 2

[12] Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *CoRR*, abs/2310.02557, 2023. doi: 10.48550/ARXIV.2310.02557. URL https://doi.org/10.48550/arXiv.2310.02557. 2, 8

[13] Raphaël Chetrite, Paolo Muratore-Ginanneschi, and Kay Schwieger. E. Schrödinger's 1931 paper "on the reversal of the laws of nature" ["über die umkehrung der naturgesetze", sitzungsberichte der preussischen akademie der wissenschaften, physikalisch-mathematische klasse, 8 n9 144–153]. *The European Physical Journal H*, 46(1):28, Nov 2021. ISSN 2102-6467. doi: 10.1140/epjh/s13129-021-00032-7. URL https://doi.org/10.1140/epjh/s13129-021-00032-7. 2, 3

[14] Akhil Premkumar. Generative diffusion from an action principle, 2023. URL https://arxiv.org/abs/2310.04490. 3, 24

[15] Michele Pavon. Stochastic control and nonequilibrium thermodynamical systems. *Applied Mathematics and Optimization*, 19(1):187–202, 1989. doi: 10.1007/BF01448198. URL https://doi.org/10.1007/BF01448198. 3, 9, 22, 23, 24, 25, 26

[16] Udo Seifert. Stochastic thermodynamics, fluctuation theorems and molecular machines. *Reports on Progress in Physics*, 75(12):126001, nov 2012. doi: 10.1088/0034-4885/75/12/126001. URL https://dx.doi.org/10.1088/0034-4885/75/12/126001. 3, 20

[17] Édgar Roldán and Juan M. R. Parrondo. Estimating dissipation from single stationary trajectories. *Phys. Rev. Lett.*, 105:150607, Oct 2010. doi: 10.1103/PhysRevLett.105.150607. URL https://link.aps.org/doi/10.1103/PhysRevLett.105.150607. 3

[18] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, USA, 2002. ISBN 0521642981. 3

[19] Kim Sharp and Franz Matschinsky. Translation of Ludwig Boltzmann's Paper "On the Relationship between the Second Fundamental Theorem of the Mechanical Theory of Heat and Probability Calculations Regarding the Conditions for Thermal Equilibrium". *Entropy*, 17(4):1971–2009, 2015. ISSN 1099-4300. doi: 10.3390/e17041971. URL https://www.mdpi.com/1099-4300/17/4/1971. Originally published in *Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften. Mathematisch-Naturwissen Classe. Abt. II*, vol. 76, pp. 373–435 (Wien, 1877); reprinted in *Wiss. Abhandlungen*, Vol. II, reprint 42, pp. 164–223, Barth, Leipzig, 1909. 3

[20] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1415–1428. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0a9fdbb17feb6ccb7ec405cfb85222c4-Paper.pdf. 4, 5, 21, 22, 31, 37

[21] J. C. Maxwell. Ii. illustrations of the dynamical theory of gases. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 20(130):21–37, 1860. doi: 10.1080/14786446008642902. URL https://doi.org/10.1080/14786446008642902. 4

[22] Koji Maruyama, Franco Nori, and Vlatko Vedral. Colloquium: The physics of maxwell's demon and information. *Rev. Mod. Phys.*, 81:1–23, Jan 2009. doi: 10.1103/RevModPhys.81.1. URL https://link.aps.org/doi/10.1103/RevModPhys.81.1. 4

[23] J M R Parrondo, C Van den Broeck, and R Kawai. Entropy production and the arrow of time. *New Journal of Physics*, 11(7):073008, jul 2009. doi: 10.1088/1367-2630/11/7/073008. URL https://dx.doi.org/10.1088/1367-2630/11/7/073008. 4

[24] Takahiro Sagawa and Masahito Ueda. Generalized jarzynski equality under nonequilibrium feedback control. *Phys. Rev. Lett.*, 104:090602, Mar 2010. doi: 10.1103/PhysRevLett.104.090602. URL https://link.aps.org/doi/10.1103/PhysRevLett.104.090602. 4

[25] Shoichi Toyabe, Takahiro Sagawa, Masahito Ueda, Eiro Muneyuki, and Masaki Sano. Experimental demonstration of information-to-energy conversion and validation of the generalized jarzynski equality. *Nature Physics*, 6(12):988–992, 2010. ISSN 1745-2481. doi: 10.1038/nphys1821. URL https://doi.org/10.1038/nphys1821. 4

[26] Christopher Jarzynski. Rare events and the convergence of exponentially averaged work values. *Phys. Rev. E*, 73:046105, Apr 2006. doi: 10.1103/PhysRevE.73.046105. URL https://link.aps.org/doi/10.1103/PhysRevE.73.046105. 5

[27] S. Vaikuntanathan and C. Jarzynski. Dissipation and lag in irreversible processes. *Europhysics Letters*, 87(6):60005, oct 2009. doi: 10.1209/0295-5075/87/60005. URL https://dx.doi.org/10.1209/0295-5075/87/60005. 5, 26

[28] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *CoRR*, abs/2210.02747, 2022. doi: 10.48550/ARXIV.2210.02747. URL https://doi.org/10.48550/arXiv.2210.02747. 5

[29] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/a98846e9d9cc01cfb87eb694d946ce6b-Abstract-Conference.html. 6, 7, 30

[30] Nikita Kornilov, Alexander V. Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. *CoRR*, abs/2403.13117, 2024. doi: 10.48550/ARXIV.2403.13117. URL https://doi.org/10.48550/arXiv.2403.13117.

[31] Yuyan Ni, Shikun Feng, Haohan Chi, Bowen Zheng, Huan ang Gao, Wei-Ying Ma, Zhi-Ming Ma, and Yanyan Lan. Straight-line diffusion model for efficient 3d molecular generation, 2025. URL https://arxiv.org/abs/2503.02918. 6, 35

[32] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, Jan 2000. ISSN 0945-3245. doi: 10.1007/s002110050002. URL https://doi.org/10.1007/s002110050002. 6

[33] Tan Van Vu and Keiji Saito. Thermodynamic unification of optimal transport: Thermodynamic uncertainty relation, minimum dissipation, and thermodynamic speed limits. *Phys. Rev. X*, 13:011013, Feb 2023. doi: 10.1103/PhysRevX.13.011013. URL https://link.aps.org/doi/10.1103/PhysRevX.13.011013. 6, 9

[34] Kotaro Ikeda, Tomoya Uda, Daisuke Okanohara, and Sosuke Ito. Speed-accuracy trade-off for the diffusion models: Wisdom from nonequilibrium thermodynamics and optimal transport, 2024. URL https://arxiv.org/abs/2407.04495. 6, 9, 10

[35] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261. 7

[36] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. 8

[37] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. 8

[38] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *CoRR*, abs/2202.00512, 2022. URL https://arxiv.org/abs/2202.00512. 8

[39] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7794–7803. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00813. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Wang_Non-Local_Neural_Networks_CVPR_2018_paper.html. 9

[40] Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22863–22876. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c11abfd29e4d9b4d4b566b01114d8486-Paper.pdf. 9, 21, 22, 24, 31

[41] David McAllester. On the mathematics of diffusion models, 2023. URL https://arxiv.org/abs/2301.11108. 9

[42] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=CNicRIVIPA. 9

[43] Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *CoRR*, abs/2410.21357, 2024. doi: 10.48550/ARXIV.2410.21357. URL https://doi.org/10.48550/arXiv.2410.21357. 9

[44] Ziguang Li, Chao Huang, Xuliang Wang, Haibo Hu, Cole Wyeth, Dongbo Bu, Quan Yu, Wen Gao, Xingwu Liu, and Ming Li. Lossless data compression by large models. *Nature Machine Intelligence*, May 2025. ISSN 2522-5839. doi: 10.1038/s42256-025-01033-7. URL https://doi.org/10.1038/s42256-025-01033-7. 9

[45] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, October 2023. 9

[46] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361. 9

[47] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations (ICLR)*, 2022. 9

[48] Alexandra Lamtyugina, Agnish Kumar Behera, Aditya Nandy, Carlos Floyd, and Suriyanarayanan Vaikuntanathan. Score-based generative diffusion with "active" correlated noise sources. *CoRR*, abs/2411.07233, 2024. doi: 10.48550/ARXIV.2411.07233. URL https://doi.org/10.48550/arXiv.2411.07233. 9

[49] C. Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Phys. Rev. E*, 56:5018–5035, Nov 1997. doi: 10.1103/PhysRevE.56.5018. URL https://link.aps.org/doi/10.1103/PhysRevE.56.5018. 9, 26

[50] Gavin E. Crooks. Nonequilibrium measurements of free energy differences for microscopically reversible markovian systems. *Journal of Statistical Physics*, 90(5):1481–1487, 1998. ISSN 1572-9613. doi: 10.1023/A:1023208217925. URL https://doi.org/10.1023/A:1023208217925. 9

[51] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=oYIjw37pTP. 9

[52] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 62183–62223. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c428adf74782c2092d254329b6b02482-Paper-Conference.pdf. 9

[53] Ludwig Winkler, Cesar Ojeda, and Manfred Opper. A score-based approach for training schrödinger bridges for data modelling. *Entropy*, 25(2):316, 2023. ISSN 1099-4300. doi: 10.3390/e25020316. URL https://www.mdpi.com/1099-4300/25/2/316. 9

[54] Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023. URL https://openreview.net/forum?id=UvmDCdSPDOW. Preprint at https://arxiv.org/abs/2302.03792. 9

[55] Giulio Franzese, Mustapha Bounoua, and Pietro Michiardi. MINDE: Mutual information neural diffusion estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 16685–16716, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/47f75e809409709c6d226ab5ca0c9703-Paper-Conference.pdf. 9

[56] Benjamin Hoover, Hendrik Strobelt, Dmitry Krotov, Judy Hoffman, Zsolt Kira, and Duen Horng Chau. Memory in plain sight: A survey of the uncanny resemblances between diffusion models and associative memories. *CoRR*, abs/2309.16750, 2023. doi: 10.48550/ARXIV.2309.16750. URL https://doi.org/10.48550/arXiv.2309.16750. 9

[57] Kunio Yasue. A simple derivation of the onsager–machlup formula for one-dimensional nonlinear diffusion process. *Journal of Mathematical Physics*, 19(8):1671–1673, 08 1978. ISSN 0022-2488. doi: 10.1063/1.523888. URL https://doi.org/10.1063/1.523888. 19

[58] Edward Nelson. Derivation of the schrödinger equation from newtonian mechanics. *Phys. Rev.*, 150:1079–1085, Oct 1966. doi: 10.1103/PhysRev.150.1079. URL https://link.aps.org/doi/10.1103/PhysRev.150.1079. 21

[59] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: https://doi.org/10.1016/0304-4149(82)90051-5. URL https://www.sciencedirect.com/science/article/pii/0304414982900515.

[60] U. G. Haussmann and E. Pardoux. Time Reversal of Diffusions. *The Annals of Probability*, 14 (4):1188 – 1205, 1986. doi: 10.1214/aop/1176992362. URL https://doi.org/10.1214/aop/1176992362.

[61] Hans Föllmer. Random fields and diffusion processes. In Paul-Louis Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, pages 101–203, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg. ISBN 978-3-540-46042-8. 21

[62] Sosuke Ito. Geometric thermodynamics for the fokker–planck equation: stochastic thermodynamic links between information geometry and optimal transport. *Information Geometry*, 7(1):441–483, 2024. ISSN 2511-249X. doi: 10.1007/s41884-023-00102-3. URL https://doi.org/10.1007/s41884-023-00102-3. 22

[63] M. Kac. On distributions of certain wiener functionals. *Transactions of the American Mathematical Society*, 65(1):1–13, 1949. ISSN 00029947, 10886850. URL http://www.jstor.org/stable/1990512. 24

[64] Wendell Fleming and Raymond Rishel. *Deterministic and Stochastic Optimal Control*. Stochastic Modelling and Applied Probability. Springer New York, NY, 1 edition, 1975. ISBN 978-0-387-90155-8. doi: 10.1007/978-1-4612-6380-7. URL https://doi.org/10.1007/978-1-4612-6380-7. 25

[65] M.I. Kamien and N.L. Schwartz. *Dynamic Optimization, Second Edition: The Calculus of Variations and Optimal Control in Economics and Management*. Dover Books on Mathematics. Dover Publications, 2013. ISBN 9780486310282. URL https://books.google.com/books?id=liLCAgAAQBAJ. 25

[66] Ludwig Boltzmann. *Further Studies on the Thermal Equilibrium of Gas Molecules*, pages 262–349. Imperial College Press, 2003. doi: 10.1142/9781848161337_0015. 26

[67] Gavin E. Crooks and Christopher Jarzynski. Work distribution for the adiabatic compression of a dilute and interacting classical gas. *Phys. Rev. E*, 75:021116, Feb 2007. doi: 10.1103/PhysRevE.75.021116. URL https://link.aps.org/doi/10.1103/PhysRevE.75.021116. 26

[68] Christopher Jarzynski. *Equalities and Inequalities: Irreversibility and the Second Law of Thermodynamics at the Nanoscale*, pages 145–172. Springer Basel, Basel, 2013. ISBN 978-3-0348-0359-5. doi: 10.1007/978-3-0348-0359-5_4. URL https://doi.org/10.1007/978-3-0348-0359-5_4. 26

[69] E. T. Jaynes. Information theory and statistical mechanics. ii. *Phys. Rev.*, 108:171–190, Oct 1957. doi: 10.1103/PhysRev.108.171. URL https://link.aps.org/doi/10.1103/PhysRev.108.171. 26

[70] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954. 26

[71] Maxwell Aifer, Samuel Duffield, Kaelan Donatella, Denis Melanson, Phoebe Klett, Zach Belateche, Gavin E. Crooks, Antonio J. Martinez, and Patrick J. Coles. Thermodynamic bayesian inference. *CoRR*, abs/2410.01793, 2024. doi: 10.48550/ARXIV.2410.01793. URL https://doi.org/10.48550/arXiv.2410.01793. 27

[72] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. doi: 10.1137/S0036141096303359. URL https://doi.org/10.1137/S0036141096303359. 27

[73] Mehran Kardar. *Statistical Physics of Particles*. Cambridge University Press, 2007. 28

[74] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf. 30

[75] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11201–11228. PMLR, 2022. URL https://proceedings.mlr.press/v162/kim22i.html. 32

[76] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. URL https://api.semanticscholar.org/CorpusID:5560643. 31

[77] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21696–21707. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b578f2a52a0229873fefc2a4b06377fa-Paper.pdf. 35, 37

[78] Akhil Premkumar. Diffusion density estimators, 2024. URL https://arxiv.org/abs/2410.06986. 35, 37

[79] Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8), 2020. ISSN 1099-4300. doi: 10.3390/e22080802. URL https://www.mdpi.com/1099-4300/22/8/802. 35

[80] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=zyLVMgsZ0U_. 35

## Contents

# A Random walk on a lattice

Random walks on a discrete one-dimensional lattice serve as a simple toy model to understand many of the results in this paper. Consider $M$ random walkers on a lattice of spacing $\ell$, each of whom can occupy any of the sites $x = n\ell, n \in \mathbb{Z}$. We shall use the time variable $s$ for the forward evolution of the walkers. In every time step $\Delta s$ every walker at $x$ jumps either left or right with probability $q_L(x,s)$ and $q_R(x,s)$ respectively, so $q_L + q_R = 1$. Then, the probability of finding a walker at $x$ updates as

$$p(x, s+\Delta s) = q_R(x-\ell, s)p(x-\ell, s) + q_L(x+\ell, s)p(x+\ell, s), \tag{26}$$

since all walkers that were at $x$ at time $s$ cleared out and have been replaced by incoming walkers from either the left or right. Taylor expanding Eq. (26) in small $\Delta s$ and $\ell$,

$$\partial_s p(x,s) = -\frac{\ell}{\Delta s}\partial_x((q_R(x,s) - q_L(x,s))p(x,s)) + \frac{\ell^2}{2\Delta s}\partial_x^2 p(x,s), \tag{27}$$

which is the Fokker-Planck equation with drift and diffusion coefficients

$$b_+(x,s) = \frac{\ell}{\Delta s}(q_R(x,s) - q_L(x,s)), \quad \sigma^2 = \frac{\ell^2}{\Delta s}. \tag{28}$$

Conversely, given a Fokker-Planck equation with a generic drift $b_+$ we can think of it as the small $\ell$ limit of a lattice model with jump probabilities

$$q_R(x,s) = \frac{1}{2} + \frac{\Delta s}{2\ell}b_+(x,s), \quad q_L(x,s) = \frac{1}{2} - \frac{\Delta s}{2\ell}b_+(x,s). \tag{29}$$

If the diffusion coefficient is time and/or position dependent we can map it to Eq. (28) by rescaling time and/or by a change of variables [57].

## A.1 Reversal

Eq. (26) can be reversed by returning to $x-\ell$ and $x+\ell$ respectively a fraction

$$\frac{q_R(x-\ell,s)p(x-\ell,s)}{p(x,s+\Delta s)} =: \tilde{q}_L(x, s+\Delta s), \tag{30a}$$

$$\frac{q_L(x+\ell,s)p(x+\ell,s)}{p(x,s+\Delta s)} =: \tilde{q}_R(x, s+\Delta s) \tag{30b}$$

of $p(x, s+\Delta s)$. It is useful to introduce a new time variable $t$ for the reverse direction, as shown in Fig. 5. Then, the site $x$ receives $\tilde{q}_R$ fraction of the contents of $x-\ell$, and $\tilde{q}_L$ of the walkers in $x+\ell$ from time $t$, and the distribution of walkers evolve as

$$\tilde{p}(x, t+\Delta t) = \tilde{q}_R(x-\ell, t)\tilde{p}(x-\ell, t) + \tilde{q}_L(x+\ell, t)\tilde{p}(x+\ell, t), \tag{31}$$

In the small $\ell$ limit, we can compute the new drift (cf. Eq. (41)),

$$\begin{aligned} b_-(x,t) &:= \frac{\ell}{\Delta t}(\tilde{q}_L(x,t) - \tilde{q}_R(x,t)) \\ &= \frac{\ell}{\Delta t}(q_R - q_L) - \frac{\ell^2}{\Delta t}\partial_x \log p + \frac{1}{p}\frac{\ell^3}{2\Delta t}\partial_x^2((q_R - q_L)p) - \ell(q_R - q_L)\partial_s p + \dots \Big|_{x,s} \\ &= b_+(x,s) - \sigma(x,s)^2\partial_x \log p(x,s) + \mathcal{O}(\ell^3). \end{aligned} \tag{32}$$

The diffusion coefficient remains unchanged at the same order,

$$\frac{\ell^2}{\Delta t}(\tilde{q}_R(x,t) + \tilde{q}_L(x,t)) = \frac{\ell^2}{\Delta t}(q_R(x,t) + q_L(x,t)) + \mathcal{O}(\ell^3). \tag{33}$$

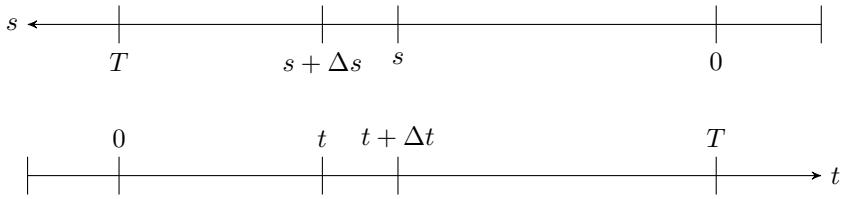



Figure 5: Time variables in the forward (top) and reverse (bottom) directions.

### A.2 Entropy production

Eq. (31) is not the only way to return to the original distribution of walkers, but it is the strategy that corresponds to the optimal kernel $h_\star$ in Eq. (3). In that expression, $g$ is the transition kernel for the forward dynamics Eq. (26), now applied in the $t$-direction. Explicitly,

$$h_\star(x_{t+\Delta t}|x_t) = \tilde{q}_R(x_t,t)\delta_{x_{t+\Delta t},x_t+\ell} + \tilde{q}_L(x_t,t)\delta_{x_{t+\Delta t},x_t-\ell}, \tag{34a}$$

$$g(x_{t+\Delta t}|x_t) = q_R(x_t,t)\delta_{x_{t+\Delta t},x_t+\ell} + q_L(x_t,t)\delta_{x_{t+\Delta t},x_t-\ell}. \tag{34b}$$

Here $\delta_{x_{t+\Delta t},x_t+\ell}$ means $x_{t+\Delta t}$ is to the right of $x_t$ etc. We can compute Eq. (3) explicitly for the random walker on a lattice.

$$\mathrm{D}_{KL}(h_\star\|g) = \sum_{x_0,x_T} p_{\mathrm{eq}}(x_0)h_\star(x_T|x_0)\log\frac{h_\star(x_T|x_0)}{g(x_T|x_0)}. \tag{35}$$

Using the log sum inequality,

$$\mathrm{D}_{KL}(h_\star\|g) \le \sum_{t=0}^{T-\Delta t}\sum_{x_{t+\Delta t},x_t} p(x_t,t)h_\star(x_{t+\Delta t}|x_t)\log\frac{h_\star(x_{t+\Delta t}|x_t)}{g(x_{t+\Delta t}|x_t)} =: S_{\mathrm{tot}}. \tag{36}$$

Using Eq. (34),

$$S_{\mathrm{tot}} = \sum_{t=0}^{T-\Delta t}\sum_{x_t} p(x_t,t)\left(\tilde{q}_R\log\frac{\tilde{q}_R}{q_R} + \tilde{q}_L\log\frac{\tilde{q}_L}{q_L}\right)\Big|_{x_t,t}. \tag{37}$$

Substituting Eq. (30) and Taylor expanding,

$$\begin{aligned} S_{\mathrm{tot}} = \sum_{t=0}^{n-1}\sum_{x_t} & p(q_L-q_R)\log\frac{q_L}{q_R} + \ell\partial_x p\log\frac{q_L}{q_R} + \frac{\ell^2}{2}p\left(q_L(\partial_x\log(q_Lp))^2 + q_R(\partial_x\log(q_Rp))^2\right) \\ & + \ell\partial_x((q_L-q_R)p) + \frac{\ell^2}{2}\partial_x^2 p - \Delta s\left(\frac{q_L}{q_R}\partial_s(q_Rp) + \frac{q_R}{q_L}\partial_s(q_Lp)\right) + \mathcal{O}(\ell^3). \end{aligned} \tag{38}$$

Expressing the jump probabilities in terms of $b_+$ and $\sigma^2$ (cf. Eq. (29)) and using the Fokker-Planck equation to eliminate some terms, we obtain the final expression for total entropy [16]:

$$S_{\mathrm{tot}} = \int_0^T \mathrm{d}t\frac{1}{2\sigma^2}\mathbb{E}_p\left[\left\|2b_+ - \sigma^2\partial_x\log p\right\|^2\right]. \tag{39}$$

## B Stochastic control

### B.1 Notation

We use the time variable $s$ for the forward diffusion process, which runs from right ($s=0$) to left ($s=T$) in Fig. 6. Sometimes we indicate functions of $s$ as $\overleftarrow{f}$ to remove ambiguity when the same function is also expressed in terms of time variable $t = T-s$. That is, $\overleftarrow{f}(s) = \overleftarrow{f}(T-t) = f(t)$. For instance, $p(x,t) \equiv \overleftarrow{p}(x,s)$. $\hat{B}_s$ and $B_t$ denote the Brownian motions associated with the forward and reverse/controlled SDEs, respectively. $\nabla$ is the gradient with respect the spatial coordinates, and $\partial_t, \partial_s$ are partial time derivatives. $S_{\mathrm{tot}}$ is the total entropy produced during forward diffusion,

$S_{\rm G}$ is the non-equilibrium Gibbs entropy of the distribution, and $S_{\rm NN}$ is the neural entropy. We make the time-dependence of the entropies explicit later in the paper after we have introduced the $s$ variable; $S_{\rm tot}$ and $S_{\rm NN}$ without the time argument should be understood as $S_{\rm tot}(s=T) \equiv S_{\rm tot}(T)$. Throughout the paper, we set Boltzmann's constant to unity, $k_{\rm B} = 1$. log is the natural logarithm. $p_{\rm d}$ and $p_0$ denote the initial ($s=0$) and final ($s=T$) densities for the forward process, and $p_{\rm eq}$ is its equilibrium state. Diffusion takes an infinite time to equilibrate but we always take $T$ to be large compared to the intrinsic time scale of the diffusion process, which is why we ignore the difference between $p_0$ and $p_{\rm eq}$ in Secs. 2 and 3. $p_u(\cdot,0)$ and $p_u(\cdot,T)$ are the initial ($t=0$) and final ($t=T$) densities of the controlled process. There is a slight abuse of notation here because $p_u(\cdot,0)$ is a distribution that does not depend on the control $u$, it is just the initial state to which the control is applied.

**Assumptions** We make the same assumptions given in App. A of [20], with the following additions for entropy-matching models:

1. $\exists C > 0 \, \forall x \in \mathbb{R}^D, t \in [0,T] : \|\boldsymbol{e_\theta}(x,t)\|_2 < C(1 + \|x\|_2)$ ,
2. $\exists C > 0 \, \forall x, y \in \mathbb{R}^D, t \in [0,T] : \|\boldsymbol{e_\theta}(x,t) - \boldsymbol{e_\theta}(y,t)\|_2 < C\|x-y\|_2$ ,
3. Novikov's condition: $\mathbb{E}_p\left[\exp\left(\int_0^T \mathrm{d}t \, \frac{1}{2}\left\|\nabla \log p_{\rm eq}^{(t)} - \nabla \log p + \boldsymbol{e_\theta}\right\|^2\right)\right] < \infty$ .

### B.2 A fluctuation relation for diffusion models

Given a set of data vectors, probabilistic models attempt to learn the underlying data distribution from which these vectors could have been sampled. One way to do this is to minimize the KL divergence between the data and the model distributions. Score-based diffusion models are trained by optimizing an objective that upper bounds this KL [20, 40]. In this section, we extend this bound to a more general parameterization of the generative process.

Consider a $D$-dimensional probability density function $p_{\rm d}$ subjected to a diffusive process

$$\mathrm{d}Y_s = b_+(Y_s, s)\mathrm{d}s + \sigma(s)\mathrm{d}\hat{B}_s. \tag{40}$$

The noise is isotropic and position-independent. Under Eq. (40), the distribution $p_{\rm d}(y) \equiv \overleftarrow{p}(y,0)$ evolves to some another distribution $p_0(y) \equiv \overleftarrow{p}(y,T)$ (see Fig. 6). This process can be reversed by an SDE [58–61]

$$\mathrm{d}X_t = -b_-(X_t, T-t)\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t, \tag{41}$$

where $t = T - s$, and the drift term is

$$b_-(X,s) := b_+(X,s) - \sigma^2(s)\nabla \log \overleftarrow{p}(X,s). \tag{42}$$

Starting from $p_0(x) \equiv p(x,0)$, the reverse evolution back to $p_{\rm d}(x) \equiv p(x,T)$ appears as a playback of the forward process in Eq. (40), so that $p(x,t) = \overleftarrow{p}(x, T-t)$ at an intermediate time $t$. Crucially, we need information about the forward process, specifically the *score function* $\nabla \log \overleftarrow{p}$, to construct the reverse drift term in Eq. (42). This makes sense: the final distribution $p_0$ has little to no memory of the initial state $p_{\rm d}$, meaning that many different $p_{\rm d}$ could diffuse to roughly the same $p_0$, rendering the problem non-invertible without explicit knowledge of the forward process.

If we replace $b_-$ in Eq. (41) with a different drift term $u$, called the *control*, and evolve $p_0(x)$ by the stochastic process

$$\mathrm{d}X_t = -u(X_t, t)\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t, \tag{43}$$

the density $p_u(x,t)$ of $X_t$ will differ from $p(x,t)$, and land on a terminal distribution $p_u(x,T) \neq p(x,T)$. The KL divergence between these distributions is bounded as

$$\int_0^T \mathrm{d}t \, \frac{1}{2\sigma^2}\mathbb{E}_{p(\cdot,t)}\left[\|b_- - u\|^2\right] \geq \mathrm{D}_{KL}\left(p(\cdot,T)\|p_u(\cdot,T)\right). \tag{44}$$

More generally, if we start at some $p_u(x,0) \neq p_0(x)$,

$$\int_0^T \mathrm{d}t \, \frac{1}{2\sigma^2}\mathbb{E}_p\left[\|b_- - u\|^2\right] + \mathrm{D}_{KL}\left(p_0(\cdot)\|p_u(\cdot,0)\right) \geq \mathrm{D}_{KL}\left(p(\cdot,T)\|p_u(\cdot,T)\right). \tag{45}$$

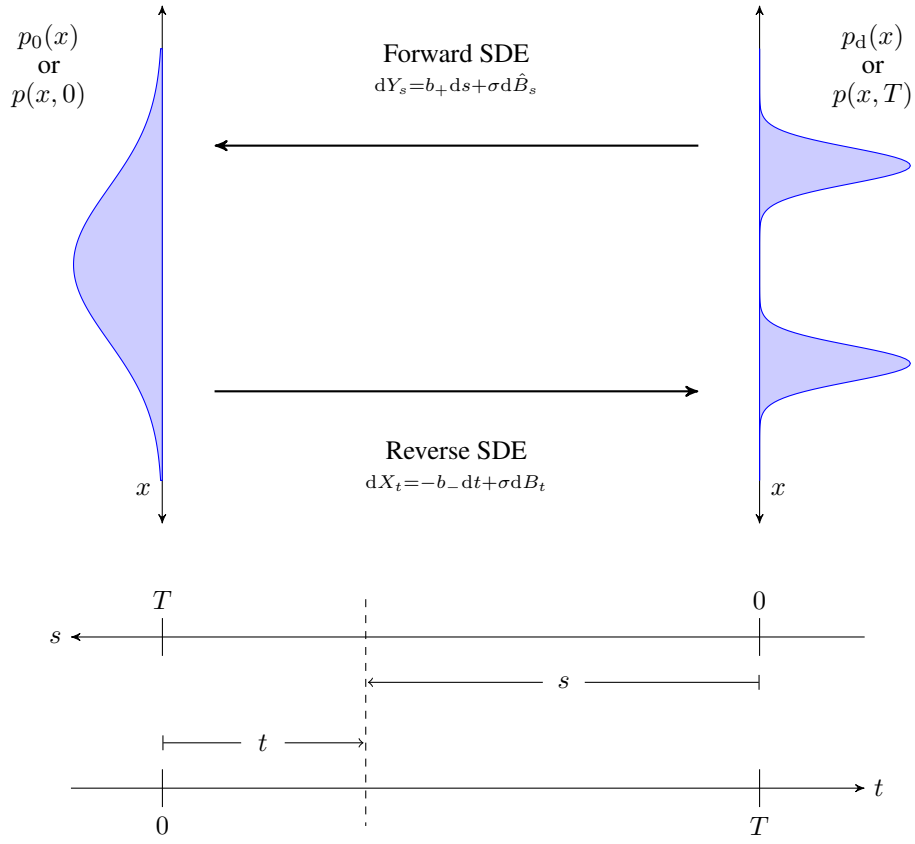



Figure 6: A schematic of the forward and reverse diffusion processes.

This result can be derived using the theory of stochastic optimal control [15] or by an application of the Feynman-Kac formula and Girsanov's theorem [40, 7]. The details are given in App. B.4. See also [62].

As a particular example, we can choose $u = b_+ - \sigma^2 \boldsymbol{s_\theta}$, where $\boldsymbol{s_\theta}$ is the output of a neural network, which converts the l.h.s. in Eq. (44) into the score-matching objective from [5, 6]. This leads to Theorem 1 from [20] (cf. Eq. (19)),

$$\int_0^T \mathrm{d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \|\boldsymbol{s_\theta} - \nabla \log p\|^2 \right] + \mathrm{D}_{KL}\left(p_0(\cdot) \| p_{\boldsymbol{\theta}}(\cdot, 0)\right) \geq \mathrm{D}_{KL}\left(p(\cdot, T) \| p_{\boldsymbol{\theta}}(\cdot, T)\right). \tag{46}$$

If we pick the initial distribution $p_{\boldsymbol{\theta}}(x, 0)$ to be close to $p_0(x)$, and train a neural network to minimize the score-matching term, we can tighten the KL divergence between the data distribution, $p(x, T) \equiv p_\mathrm{d}(x)$, and the generated one, $p_{\boldsymbol{\theta}}(x, T)$. This is how score-matching diffusion models work. Similarly, the parameterization $u = -b_+ - \sigma^2 \boldsymbol{e_\theta}$ gives the entropy-matching objective (cf. Eq. (17)),

$$\int_0^T \mathrm{d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \frac{2b_+}{\sigma^2} - \nabla \log p + \boldsymbol{e_\theta} \right\|^2 \right] + \mathrm{D}_{KL}\left(p_0(\cdot) \| p_{\boldsymbol{\theta}}(\cdot, 0)\right) \geq \mathrm{D}_{KL}\left(p(\cdot, T) \| p_{\boldsymbol{\theta}}(\cdot, T)\right). \tag{47}$$

Lastly, if we set $u = -b_+$ and choose $p_u(x, 0) = p_0(x)$ for simplicity, we obtain a lower bound on the total entropy (cf. Eq. (16))

$$S_\mathrm{tot}(T) \equiv \int_0^T \mathrm{d}t\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \left\| \frac{2b_+}{\sigma^2} - \nabla \log p \right\|^2 \right] \geq \mathrm{D}_{KL}\left(p(\cdot, T) \| p_{b_+}(\cdot, T)\right). \tag{48}$$

Next, we look at the conditions for which this bound is saturated.

### B.3 The $H$-theorem

We derive Eq. (15) here, following [15]. It states that $p$ relaxes toward $p^{\rm eq}$, with the rate of approach slowing down as it nears that state. To prove it we start with the time derivative

$$\begin{aligned}
&-\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{D}_{KL}\left(p(x,t)||p_t^{\rm eq}(x)\right)\\
&\qquad=\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_p\left[-\log\frac{p(x,t)}{p_t^{\rm eq}(x)}\right]\\
&\qquad=\frac{\mathrm{d}}{\mathrm{d}t}\left(-\int\mathrm{d}x\,p(x,t)\log p(x,t)\right)-\frac{\mathrm{d}}{\mathrm{d}t}\left(-\int\mathrm{d}x\,p(x,t)\log p_t^{\rm eq}(x)\right).
\end{aligned}\tag{49}$$

The first term on the r.h.s. is the Gibbs entropy production rate $\dot{S}(t)$ [9],

$$\begin{aligned}
\dot{S}(t)&=-\int\mathrm{d}x\,\partial_t p\log p-\int\mathrm{d}x\,\partial_t p\\
&=\int\mathrm{d}x\left(-\nabla\cdot(b_-p)-\frac{\sigma^2}{2}\nabla^2 p\right)\log p+\int\mathrm{d}x\nabla(\cdots)\\
&\overset{\rm IBP}{=}\int\mathrm{d}x\,p(x,t)\left(\frac{\sigma^2}{2}\left\|\nabla\log p\right\|^2+b_-\cdot\nabla\log p\right).
\end{aligned}\tag{50}$$

The second term in Eq. (49) can be simplified by using the Fokker-Planck equations for $p$ and $p^{\rm eq}$,

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\left(\int\mathrm{d}x\,p(x,t)\log p_t^{\rm eq}(x)\right)\\
&=\int\mathrm{d}x\,\left(\partial_t p(x,t)\log p_t^{\rm eq}(x)+p(x,t)\frac{\partial_t p_t^{\rm eq}(x)}{p_t^{\rm eq}(x)}\right)\\
&=\int\mathrm{d}x\,\left(\nabla\cdot(b_-p)+\frac{\sigma^2}{2}\nabla^2 p\right)\log p_t^{\rm eq}(x)+\int\mathrm{d}x\,\frac{p(x,t)}{p_t^{\rm eq}(x)}\left(-\nabla\cdot(b_+p_t^{\rm eq})+\frac{\sigma^2}{2}\nabla^2 p_t^{\rm eq}\right).
\end{aligned}$$

Here, we used the fact that $\partial_t p_t^{\rm eq}=0$ when $b_+/\sigma^2=$ const., and replaced that zero with Eq. (14). We will consider each new term separately for clarity. Integrating by parts,

$$\begin{aligned}
&\int\mathrm{d}x\,\left(\nabla\cdot(b_-p)+\frac{\sigma^2}{2}\nabla^2 p\right)\log p_t^{\rm eq}(x)\\
&\qquad=\int\mathrm{d}x\,p(x,t)\left(-b_-\cdot\nabla\log p_t^{\rm eq}-\frac{\sigma^2}{2}\nabla\log p\cdot\nabla\log p_t^{\rm eq}\right),
\end{aligned}$$

and

$$\begin{aligned}
&\int\mathrm{d}x\,\frac{p(x,t)}{p_t^{\rm eq}(x)}\left(-\nabla\cdot(b_+p_t^{\rm eq})+\frac{\sigma^2}{2}\nabla^2 p_t^{\rm eq}\right)\\
&=\int\mathrm{d}x\,\left(b_+\cdot\nabla p-b_+p\cdot\frac{\nabla p_t^{\rm eq}}{p_t^{\rm eq}}-\frac{\sigma^2}{2}\nabla\log p\cdot\nabla\log p_t^{\rm eq}+\frac{\sigma^2}{2}p\left\|\nabla\log p_t^{\rm eq}\right\|^2\right),\\
&=\int\mathrm{d}x\,p(x,t)\left(b_+\cdot(\nabla\log p-\nabla\log p_t^{\rm eq})-\frac{\sigma^2}{2}\nabla\log p\cdot\nabla\log p_t^{\rm eq}+\frac{\sigma^2}{2}\left\|\nabla\log p_t^{\rm eq}\right\|^2\right).
\end{aligned}$$

Adding up everything, we obtain

$$\begin{aligned}
&-\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{D}_{KL}\left(p(x,t)||p_t^{\rm eq}(x)\right)\\
&=\mathbb{E}_p\left[\frac{\sigma^2}{2}\left\|\nabla\log p(x,t)-\nabla\log p_t^{\rm eq}(x)\right\|^2+(b_-+b_+)\cdot(\nabla\log p(x,t)-\nabla\log p_t^{\rm eq}(x))\right]\\
&=-\mathbb{E}_p\left[\frac{1}{2\sigma^2}\left\|2b_+-\sigma^2\nabla\log p(x,t)\right\|^2\right],
\end{aligned}\tag{51}$$

where in the last step we used Eq. (14) to replace $\nabla\log p_t^{\rm eq}(x)$. Integrating Eq. (51) yields Eq. (15).

### B.4 Derivation of the bound

We present a derivation of the bound in Eq. (44), drawing from the proofs in [40, 15]. We only outline the steps here, and refer the reader to those papers for more formal details. Consider the process specified by the SDE

$$\mathrm{d}X_t = v(X_t, t)\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t \tag{52}$$

with the initial condition $X_0 \sim p_v(\cdot, 0)$. The evolution of $p_v(x,t)$ under Eq. (52) is given by the Fokker-Planck equation

$$\partial_t p_v + \nabla \cdot (v p_v) - \frac{\sigma^2}{2}\nabla^2 p_v = 0. \tag{53}$$

Switching the time variable to $s = T - t$ (see Fig. 6) converts this into a backward Kolmogorov equation for $\overleftarrow{p}_v(\cdot, s) := p_v(\cdot, t)$,

$$\partial_s \overleftarrow{p}_v - (\nabla \cdot v)\overleftarrow{p}_v - v \cdot \nabla \overleftarrow{p}_v + \frac{\sigma^2}{2}\nabla^2 \overleftarrow{p}_v = 0, \tag{54}$$

with the terminal condition $\overleftarrow{p}_v(\cdot, T) = p_v(\cdot, 0)$. The solution for Eq. (54) is given by the *Feynman-Kac formula* [63],

$$\overleftarrow{p}_v(x,s) = \mathbb{E}\left[\overleftarrow{p}_v(Y_T, T)\exp(-\int_s^T \mathrm{d}\bar{s}\,\nabla \cdot v(Y_{\bar{s}}, T-\bar{s}))\middle| Y_s = x\right], \tag{55}$$

where $Y_{\bar{s}}$ is a diffusion process that solves

$$\mathrm{d}Y_s = -v(Y_s, T-s)\mathrm{d}s + \sigma(s)\mathrm{d}B'_s. \tag{56}$$

That is, Eq. (55) is a *path integral* over all paths that start from $x$ at time $s$ and evolve under Eq. (56). Setting $s = 0$ in Eq. (55) gives us the likelihood $p_v(\cdot, T) \equiv \overleftarrow{p}_v(\cdot, 0)$. Next, [40] bounds the log likelihood by a change of measure and Jensen's inequality,

$$\log p_v(x,T) \geq \mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}} + \log p_v(Y_T, 0) - \int_0^T \mathrm{d}\bar{s}\,\nabla \cdot v \middle| Y_0 = x\right]. \tag{57}$$

Here $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}$ is the Radon-Nikodym derivative. For our purposes it is enough to understand the expectation value of this object as

$$\mathbb{E}_{\mathbb{Q}}\left[\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}\middle| Y_0 = x\right] = -\int \mathrm{d}y_T\, Q(y_T|x)\log\frac{Q(y_T|x)}{P(y_T|x)}, \tag{58}$$

where $P$ is the transition probability corresponding to Eq. (56) and $Q$ is the transition probability for a new process[1]

$$\mathrm{d}Y_s = w(Y_s, s)\mathrm{d}s + \sigma(s)\mathrm{d}\hat{B}_s. \tag{59}$$

Then, Eq. (58) can be simplified [14], allowing the r.h.s. in Eq. (57) to be written as the negative of a cost functional (at $s = 0$)

$$\overleftarrow{J}(x, s; v, w) := \mathbb{E}_w\left[\int_s^T \mathrm{d}\bar{s}\left(\frac{1}{2\sigma^2}\|v + w\|^2 + \nabla \cdot v\right) - \log p_v(Y_T, 0)\middle| Y_s = x\right]. \tag{60}$$

We have tinkered with the notation a little, using $\mathbb{E}_w$ to indicate that the averages are taken over Eq. (59). Eqs. (57) and (60) will be used in two different ways below. We will set $p_v(\cdot, 0) = p_0(\cdot)$ in both cases for simplicity.

**Case 1:** $v = -u, w = b_+$. Then, Eq. (52) becomes the controlled process Eq. (43) and Eq. (59) is the forward diffusion from Eq. (40), and Eq. (60) is

$$\log p_u(x,T) \geq -\overleftarrow{J}(x, 0; -u, b_+). \tag{61}$$

[1] $B'_s$ is a reparameterization of $\hat{B}_s$. See Sec. 4 of [40]

**Case 2:** $v = -b_-$. Under this choice Eq. (52) is the reverse diffusion process from Eq. (41), which takes $p_0 \to p_{\rm d}$ via $p$. Then,

$$\log p(x,T) \geq -\overleftarrow{J}\,(x,0;-b_-,w)\,. \tag{62}$$

This inequality is saturated if we set $w = b_+$ [15]. To see this, we define the *value function* $\overleftarrow{W}(x,s) := \min_w \overleftarrow{J}(x,s;-b_-,w)$, which is the minimum cost over all admissible values of $w$. It satisfies the *Dynamic Programming equation* [64, 65],

$$\partial_s \overleftarrow{W} + \frac{\sigma^2}{2}\nabla^2 \overleftarrow{W} - \nabla\cdot b_- = \min_w \left( -\frac{1}{2\sigma^2}\|b_- - w\|^2 - w\cdot\nabla\overleftarrow{W}\right). \tag{63}$$

Pointwise minimization of the r.h.s. gives $w_\star = b_- - \sigma^2\nabla\overleftarrow{W}$. Substituting this back into Eq. (63) we find that $\overleftarrow{W}$ solves

$$\partial_s \overleftarrow{W} + b_+\cdot\nabla\overleftarrow{W} + \frac{\sigma^2}{2}\nabla^2\overleftarrow{W} = \frac{\sigma^2}{2}\|\nabla\overleftarrow{W}\|^2 + \sigma^2\nabla\log\overleftarrow{p}\cdot\nabla\overleftarrow{W} + \nabla\cdot b_-, \tag{64}$$

with terminal value $\overleftarrow{W}(x,T) = -\log p_0(x)$. But notice that, for $v = -b_-$, Eq. (54) can be written as following equation for $\overleftarrow{S} := -\log\overleftarrow{p}$,

$$\partial_s \overleftarrow{S} + b_+\cdot\nabla\overleftarrow{S} + \frac{\sigma^2}{2}\nabla^2\overleftarrow{S} = -\frac{\sigma^2}{2}\|\nabla\overleftarrow{S}\|^2 + \nabla\cdot b_-, \tag{65}$$

also with a terminal value $\overleftarrow{S}(x,T) = -\log p_0(x)$. Comparing Eqs. (64) and (65), we see that $\overleftarrow{W}(x,s) = -\log\overleftarrow{p}\,(x,s)$, and Eq. (62) becomes

$$\begin{aligned}\log p(x,T) &= -\overleftarrow{W}(x,0)\\ &= -\mathbb{E}_{b_+}\left[\int_s^T \mathrm{d}\bar{s}\left(\frac{1}{2\sigma^2}\|\nabla\log\overleftarrow{p}\,\|^2 - \nabla\cdot b_-\right) - \log p_0(Y_T)\middle| Y_0 = x\right].\end{aligned} \tag{66}$$

The bound on the KL divergence between $p$ and $p_u$ can be obtained by integrating Eqs. (61) and (66) over $p_{\rm d}$,

$$-\int_0^T \mathrm{d}\bar{s}\,\mathbb{E}_{\overleftarrow{p}}\left[\frac{\|b_+ - b_-\|^2 - \|b_+ - u\|^2}{2\sigma^2} - \nabla\cdot(b_- - u)\right] \geq \mathrm{D}_{KL}\left(p(\cdot,T)||p_u(\cdot,T)\right). \tag{67}$$

The last term in the average can be integrated by parts,

$$-\mathbb{E}_{\overleftarrow{p}}\left[\nabla\cdot(b_- - u)\right] = \mathbb{E}_{\overleftarrow{p}}\left[(b_- - u)\cdot\nabla\log\overleftarrow{p}\right] \overset{(42)}{=} \frac{1}{\sigma^2}\,\mathbb{E}_{\overleftarrow{p}}\left[(b_- - u)\cdot(b_- - b_+)\right], \tag{68}$$

to rewrite Eq. (67) in its final form

$$\int_0^T \mathrm{d}\bar{s}\,\frac{1}{2\sigma^2}\mathbb{E}_{\overleftarrow{p}}\left[\|b_- - u\|^2\right] \geq \mathrm{D}_{KL}\left(p(\cdot,T)\|p_u(\cdot,T)\right). \tag{69}$$

Note that (a) since $p(\cdot,t) = \overleftarrow{p}\,(\cdot,s)$ we can replace the average $\mathbb{E}_{\overleftarrow{p}} \to \mathbb{E}_p$ and change the time integral to run over $t$, which gives Eq. (44), and (b) we would have an additional KL term if we had $p_0(\cdot) \neq p_u(\cdot,0)$ in Eq. (67), the one from Eq. (45).

# C Non-equilibrium thermodynamics

## C.1 Dissipation, lag, and the information gap

Eq. (15) is a Jarzynski equality, applied to Langevin dynamics. Assuming $p_0 \approx p_{\rm eq}^{(0)}$ so that we can ignore the KL between them, Eq. (15) is

$$S_{\rm tot} = \mathrm{D}_{KL}\left(p_{\rm d}\middle\|p_{\rm eq}^{(T)}\right). \tag{70}$$

The l.h.s. is the total entropy produced as the distribution $p_{\rm d}$ is diffused to $p_0$ by Eq. (40) [9]. As diffusion progresses in the $s$-direction, our knowledge of the system diminishes over time. $S_{\rm tot}$ is a measure of this information loss. Seen from the $t$-direction, Eq. (41) restores the information worn away in the forward process. This is the perspective put forward in the Vaikuntanathan-Jarzynski (VJ) relation for irreversible processes [27],

$$W_{\rm diss}(t) = \beta^{-1} {\rm D}_{KL}(\rho_t \| \rho_t^{\rm eq}). \tag{71}$$

This equation can be understood by considering a system, initially at a temperature $\beta^{-1}$, driven away from equilibrium by varying an external parameter $\lambda$ from $A$ to $B$, over a time interval $t \in [0, T]$. Let $\langle W \rangle$ be the average mechanical work needed to effect this transformation which, according to the Second Law, is at least equal to the free energy difference $\Delta F$ between $A$ and $B$. Then, $W_{\rm diss} = \langle W \rangle - \Delta F$ is the average work dissipated over the whole process. $\rho_t$ is the phase space density as the system evolves from $A$ to $B$, and $\rho_t^{\rm eq}$ is the equilibrium density corresponding to the value of the parameter at that instant, $\lambda_t$.[2] That is, if we adjust the parameter to $\lambda_t$ and wait a long time, the system will evolve to $\rho_t^{\rm eq}$, its entropy increasing monotonically during the process. This is Boltzmann's $H$-theorem [66, 15]. In other words, $\rho_t^{\rm eq}$ is the maximum entropy (or minimum information) distribution consistent with $\lambda_t$ [3, 4].

On the other hand, under finite time non-equilibrium evolution the system is rushed along to the state $\rho_t$ and is not afforded the time to relax to the maximum entropy configuration. As a result, a *lag* develops between $\rho_t$ and $\rho_t^{\rm eq}$, as measured by the KL divergence in Eq. (71). Lag indicates the extent to which the system is out of equilibrium. The VJ relation, Eq. (71), says that the dissipated work dictates the maximum extent to which the equilibrium can be broken at a given instant.

We can also interpret the lag as the *information gap* between $\rho_t$ and $\rho_t^{\rm eq}$. The entropy of a system is a measure of missing information, with larger entropy associated with a greater degree of ignorance about the system's true state. $\rho_t^{\rm eq}$ has a higher entropy than $\rho_t$ since much of the information in the latter is lost when $\rho_t$ equilibrates to $\rho_t^{\rm eq}$. Intuitively, it is clear that the gap is precisely the amount of information we need to exhume $\rho_t$ from $\rho_t^{\rm eq}$.

In the context of Eq. (70), the non-equilibrium process is the reverse diffusion from Eq. (41). The VJ relation forces a shift in perspective, nudging us to think of Eq. (13b) as the 'native' dynamics of the system, with the process in Eq. (13a) driving it away from its 'preferred state' $p_{\rm eq}^{(t)}$. The gap measures the information deficit that keeps the native dynamics from reaching $p_{\rm d}$ on its own. This is also the lesson from Schrödinger's thought experiment (cf. Sec. 2). We can close the gap by modifying the native dynamics to enhance the probability of the outcome $p_{\rm d}$ (cf. Eq. (9)). The *maximum* additional information needed to do this is $S_{\rm tot}$.

These arguments also serve to illustrate a specific point about the Second Law: it is a statement about the irreversibility of a non-equilibrium transformation, *even if* that process is simulated on a computer. In the real world irreversibility is an observed property of almost all physical processes[3] [68, 69], but we take this for granted since we evolve with SDEs that are not time-symmetric by construction (cf. Eq. (40)).

### C.2 Entropy and free energy

Shannon entropy can be understood as a measure of ignorance. The analogous quantity we use for diffusive processes is the non-equilibrium Gibbs entropy [9]

$$S_{\rm G}[\overleftarrow{p}] := -\int {\rm d}x\, \overleftarrow{p}(x,s) \log \overleftarrow{p}(x,s). \tag{72}$$

$S_{\rm G}$ can be understood as a continuous version of the Shannon entropy,[4] up to a multiplicative factor $k_{\rm B}$ which we set to 1 (see footnote 15 in [3]). Gibbs entropy is a measure of our uncertainty in the state of the system, which in this case are the locations of the diffusing particles. But we can choose the drift and diffusion coefficients such that the final distribution is narrower than the initial one

[2] $\rho_t^{\rm eq} \equiv \rho^{\rm eq}(\cdot, \lambda_t)$ depends on time only through the parameter $\lambda_t$.

[3] We are specifically referring to systems with a large number of degrees of freedom, $N$; if $N$ is small enough it is possible to obtain 'second law violating' behavior, see for instance [49, 67].

[4] Eq. (72) is also called the *differential entropy* of a continuous random variable, which has some important differences from the discrete version. Refer to chapter 8 of [70] for details.

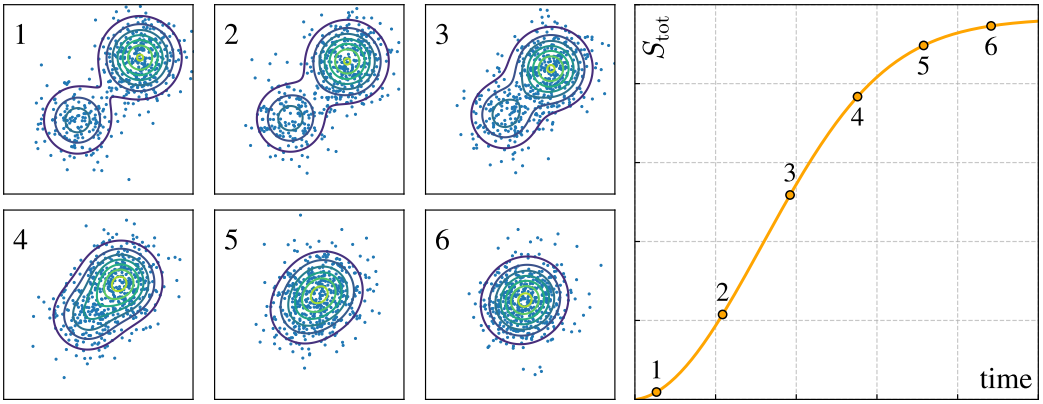



Figure 7: Diffusion is a non-equilibrium process that generates entropy over time. On the left, we see snapshots of a diffusive process (Ornstein-Uhlenbeck). In the forward direction (increasing $s$) the distribution evolves from $1 \to 6$ (i.e. $p_{\mathrm{d}} \to p_0$), and entropy produced till that point in time is indicated on the right. Note that $S_{\mathrm{tot}}$ is the total entropy produced, which is different from the change in Gibbs entropy of the distribution, which is negative in this experiment.

(see Fig. 7). Then, our ignorance of the particle positions would be reduced, so the change in Gibbs entropy is *negative*. However, the total entropy increases, as expected; $S_{\mathrm{tot}}$ is not just the change in Gibbs entropy.

We can see this explicitly by looking at the expressions for both. Combining Eqs. (66) and (72) and integrating by parts one obtains

$$S_{\mathrm{G}}[p_0] - S_{\mathrm{G}}[p_{\mathrm{d}}] = \int_0^T \mathrm{d}s\, \mathbb{E}_{\overleftarrow{p}} \left[ \frac{\sigma^2}{2} \left\| \nabla \log \overleftarrow{p} \right\|^2 + \nabla \cdot b_+ \right], \tag{73}$$

which is different from the total entropy,

$$S_{\mathrm{tot}} = \int_0^T \mathrm{d}s\, \frac{\sigma^2}{2} \mathbb{E}_{\overleftarrow{p}} \left[ \left\| \frac{2b_+}{\sigma^2} - \nabla \log \overleftarrow{p} \right\|^2 \right] \overset{(70)}{=} \mathrm{D}_{KL}\left(p_{\mathrm{d}} \| p_{\mathrm{eq}}^{(T)}\right) - \mathrm{D}_{KL}\left(p_0 \| p_{\mathrm{eq}}^{(0)}\right). \tag{74}$$

These expressions differ when $b_+ \neq 0$. To understand how they are related we look at the free energy

$$F[p] = E[p] - \beta^{-1} S_{\mathrm{G}}[p], \tag{75}$$

where the temperature $\beta^{-1} := \sigma^2/2$ and the energy is

$$E[p] := \mathbb{E}_p\left[U(x)\right], \qquad U(x) = -\int^x \mathrm{d}\bar{x}\, b_+(\bar{x}). \tag{76}$$

For simplicity we will assume that $b_+$ and $\sigma$ are time-independent (cf. Eq. (5)), so we have a static equilibrium state $p_{\mathrm{eq}} = Z^{-1}\exp(-\beta U(x))$, and that $p_0 = p_{\mathrm{eq}}$. The generalization to the time-dependent case is straightforward. Then,

$$S_{\mathrm{tot}} = \mathrm{D}_{KL}\left(p_{\mathrm{d}} \| p_{\mathrm{eq}}\right) = -S_{\mathrm{G}}[p_{\mathrm{d}}] + \beta E[p_{\mathrm{d}}] + \log Z = \beta(F[p_{\mathrm{d}}] - F[p_{\mathrm{eq}}]), \tag{77}$$

where in the last step we have used $\beta F[p_{\mathrm{eq}}] = -\log Z$, which follows from evaluating Eq. (75) on $p_{\mathrm{eq}}$. Since $S_{\mathrm{tot}}$ is positive, $F[p_{\mathrm{d}}] > F[p_{\mathrm{eq}}]$ irrespective of whether $S_{\mathrm{G}}[p_{\mathrm{d}}]$ is larger or smaller than $S_{\mathrm{G}}[p_{\mathrm{eq}}]$. In Fig. 7, frame 1 has a higher Gibbs entropy but also a higher energy, so the particles coalesce into the distribution in frame 6, giving up some of their Gibbs entropy to move to a lower energy configuration. Thus, Langevin dynamics moves $p_{\mathrm{d}}$ towards the lower free energy state $p_{\mathrm{eq}}$ [71]. The minimization of free energy is also closely related to the interpretation of the Fokker-Planck equation as a Wasserstein gradient flow [72].

Relating $S_{\mathrm{tot}}$ to free energy also helps us connect the discussion in Sec. 2 to statistical mechanics. In systems with a large number of particles, the equilibrium distribution of microstates is sharply peaked, with the most probable microstates piled up around the free energy minima (see [3] or Sec. 4.6 in

[73]). The distributions $p$ are the microstates in the Schrödinger picture, and $\mathcal{P}[p] \propto \exp(-\beta N F[p])$ is peaked at $p_{\rm eq}$. Then,

$$e^{-NS_{\rm tot}} = \frac{e^{-\beta N F[p_{\rm d}]}}{e^{-\beta N F[p_{\rm eq}]}} \approx \mathcal{P}[p_{\rm d}]. \tag{78}$$

Modifying the drift term in Eq. (9) changes the free energy landscape such that $p_{\rm d}$ becomes its new minima. In other words, the diffusion model applies an external force to do work on the system, and the range of possible outcomes of the combined arrangement constitutes a non-equilibrium analog of the *Gibbs ensemble*.

# D Score matching

In Sec. 4 we touched on the difficulty in defining neural entropy for the score-matching model. This point warrants further elaboration. We start with

$$\mathrm{d}X_t = -(b_+(X_t, T-t) - \sigma(t)^2 \nabla \log p(X_t, t))\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t, \tag{79a}$$

$$\mathrm{d}X_t = -b_+(X_t, T-t)\mathrm{d}t + \sigma(T-t)\mathrm{d}B_t. \tag{79b}$$

Notice that the drift term in Eq. (79b) has the opposite sign to the one in Eq. (13b). This seems like a natural choice, since modifying $-b_+ \to -b_+ + \sigma^2 \boldsymbol{s_\theta}$ sets up the model to learn the score $\nabla \log p$ Eq. (19). But note that $b_+$ is a confining drift term which means $-b_+$ is not. Therefore, Eq. (79b) does not have a quasi-invariant distribution, and the intuition from Sec. 4 no longer holds. We can identify the inconsistencies arising from this conceptual breakdown through a straightforward calculation.

Starting with Eq. (11), let us tentatively define the ideal score matching neural entropy

$$\hat{S}^{\rm sm}_{\rm NN}(T) := \int_0^T \mathrm{d}s\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \|\nabla \log p\|^2 \right]. \tag{80}$$

The quantity on the right can be related to the non-equilibrium Gibbs entropy of the diffusing distribution, which we will write in terms of the time variable $s$ (cf. Eq. (72)). The change in Gibbs entropy under the forward process is given in Eq. (73). We may therefore rewrite Eq. (80) as

$$\hat{S}^{\rm sm}_{\rm NN}(T) = S_{\rm G}[p_0] - S_{\rm G}[p_{\rm d}] - \int_0^T \mathrm{d}s\, \mathbb{E}_p \left[ \nabla \cdot b_+ \right]. \tag{81}$$

To simplify further we need to choose the drift term $b_+$. Let us consider the Variance Preserving (VP) process [1, 6], for which $b_+(y, s) = -\beta(s)y/2$ and $\sigma(s) = \sqrt{\beta(s)}$ in Eq. (40), where $\beta(s)$ is positive. Noting that $\nabla \cdot b_+ = -\frac{1}{2}\beta(s) \nabla \cdot x = -\frac{D}{2}\beta(s)$, Eq. (81) reduces to

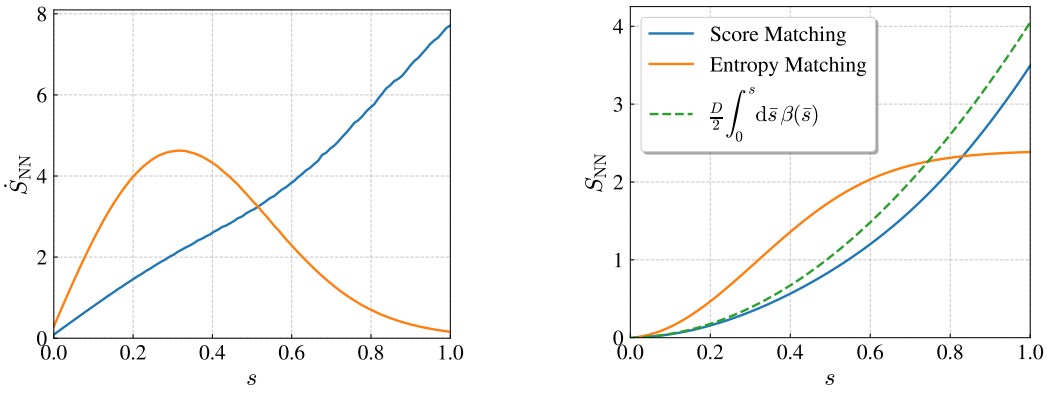
$$\hat{S}^{\rm VP}_{\rm NN}(T) = S_{\rm G}[p_0] - S_{\rm G}[p_{\rm d}] + \frac{D}{2} \int_0^T \mathrm{d}s\, \beta(s). \tag{82}$$

Upon closer inspection, Eq. (82) reveals a problem with the score matching neural entropy. Consider the trivial case where $p_0$ and $p_{\rm d}$ are identical. The Gibbs entropies at the initial and final times are equal, therefore the first two terms cancel. But we are still left with a positive integral on the r.h.s. (see Fig. 8b). This is also apparent from Eq. (80)—the score function is static but it is not zero, so the neural entropy is some positive number, and 'information' is delivered to the network. Next, imagine changing $p_{\rm d}$ to some other distribution $p_{\rm d}{}'$, with $p_0$ and the $\beta$-schedule kept fixed. The only change in Eq. (11) is the $-S_{\rm G}[p_{\rm d}]$ term. As a result, if $p_{\rm d}{}'$ has a larger entropy than $p_{\rm d}$ the neural entropy decreases. It would seem that the network needs to remember *less* information to transform $p_0 \to p_{\rm d}{}'$ than it does to convert $p_{\rm d}$ to itself!

The issue arises from the $-b_+$ term in Eq. (79). For the VP process that SDE is

$$\mathrm{d}X = \frac{\beta}{2} X_t \mathrm{d}t + \sqrt{\beta}\mathrm{d}B_t. \tag{83}$$

But this process has a repulsive drift term which, given enough time, dilutes the distribution away to infinity. Therefore, the network $\boldsymbol{s_\theta}$ has to *work against* $-b_+$ to keep the distribution $p_{\boldsymbol{\theta}}$ intact. In the above examples the growth in neural entropy due to the $\int \mathrm{d}s \beta(s)$ term in Eq. (82) is indicative of the effort needed to hold the diffusing particles in place as the drift $\beta X/2$ tries to drive them apart (see




(a) Entropy production rate.

(b) Total entropy.

Figure 8: Ideal neural entropy curves for score matching and entropy matching models with a VP process. For entropy-matching models, the neural entropy is the same as $S_{\mathrm{tot}}$ (cf. Eq. (11)), which is why entropy production trails off as forward diffusion approaches its final state.

Fig. 8). For this reason, the score matching neural entropy from Eq. (80) is not an accurate gauge of the non-trivial information the network must store to reverse diffusion.

In practice, Eq. (80) can be approximated by

$$S^{\mathrm{sm}}_{\mathrm{NN}}(T) := \int_0^T \mathrm{d}s\, \frac{\sigma^2}{2} \mathbb{E}_p \left[ \|\boldsymbol{s}_{\boldsymbol{\theta}}\|^2 \right], \tag{84}$$

just like we did in Eq. (18). Experimental comparison of the neural entropy in entropy-matching models and the $S^{\mathrm{sm}}_{\mathrm{NN}}$ for score-matching models are shown in Fig. 9.

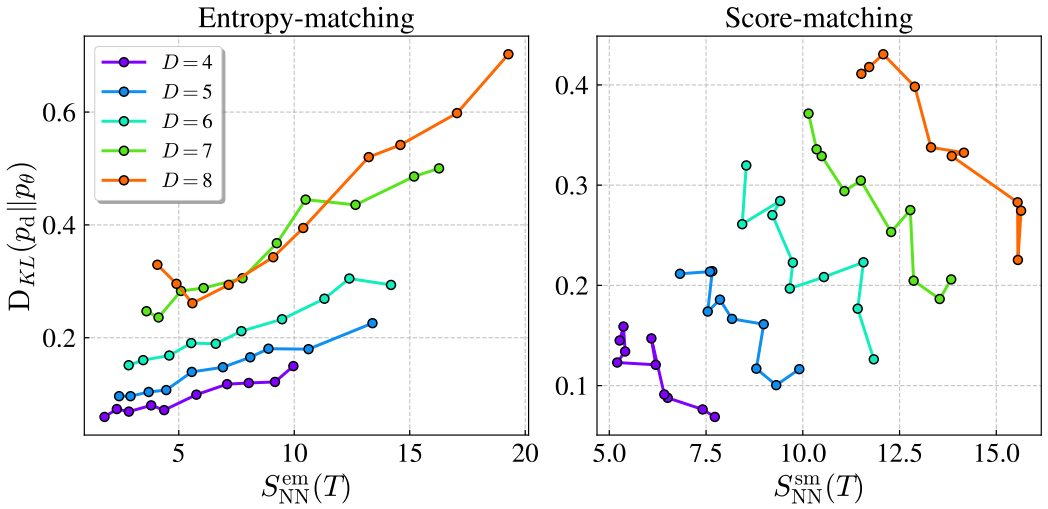



Figure 9: KL versus neural entropy for the entropy-matching and score-matching models. Both models are trained on a series of progressively larger Gaussian mixtures, just like the ones used for Fig. 11. The experiments are repeated in different dimensions, $D$. $S^{\mathrm{em}}_{\mathrm{NN}}$ is the neural entropy defined in Eq. (11). $S^{\mathrm{sm}}_{\mathrm{NN}}$ is an analogous, but different, quantity defined in App. D in an attempt to define neural entropy for a score-matching model. Note how network performance *decreases* at lower values of $S^{\mathrm{sm}}_{\mathrm{NN}}$, in contrast to the trend in entropy-matching. This behavior is explained in App. D. The VP process was used for both sets of experiments, with the same $\beta$-schedule.

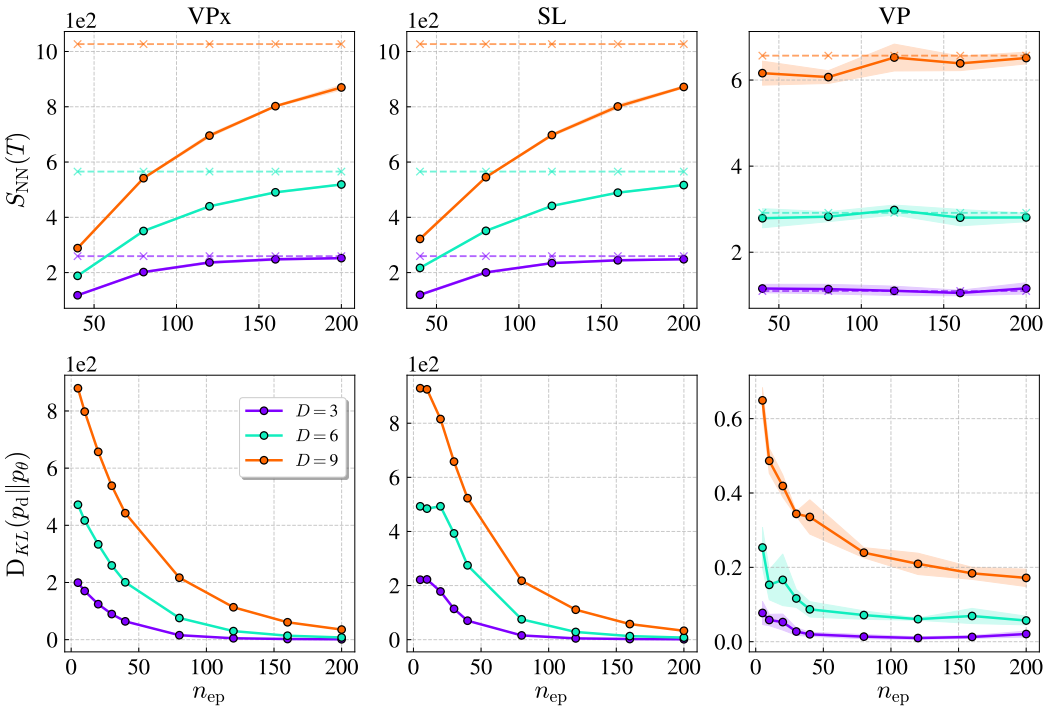



Figure 10: The evolution of neural entropy (top) and the KL between $p_{\rm d}$ and the reconstructed distribution $p_{\boldsymbol{\theta}}(\cdot, T)$ (bottom) over training epochs, $n_{\rm ep}$. The entropies and KL are measured at $s = T = 1$. The $p_{\rm d}$'s are Gaussian mixtures in $D = 3, 6, 9$ dimensions. The dashed lines are the actual value of $S_{\rm tot}$ generated by the respective diffusion processes. The VPx and SL processes ($\kappa = \sigma_0 = 0.1$) produce two orders of magnitude larger $S_{\rm tot}$ than VP, which is why $S_{\rm NN}$ is slow to catch up in these models—the network takes longer to absorb more information. This increase in retention is tracked by a decrease in the KL. Here again, the VP model outshines VPx and SL: the diffusion model can reconstitute $p_{\rm d}$ more faithfully when it has to remember less information to do so.

# E Details of experiments

All models in this paper, except for the ones in Fig. 9, were trained with 4 random seeds varying both weights initialization and order of training data, and the results were averaged over. The faint bands in the plots are within one standard deviation from the mean measurements. All computations were done on A100 GPUs with 80 GB of memory. The CIFAR-10 models were trained on 4 GPUs in parallel while the Gaussian mixture and MLP experiments were trained on just one. Training on CIFAR-10 with the full dataset ($n_c = 5000$ in Fig. 16) takes 4.5 hours. Experiments for MNIST that stop between training epochs to compute log densities (see Figs. 3 and 13), and repeat for different numbers of training samples, take around 4 hours for each training seed. For the low-$D$ models in the transport experiments we used Fourier features on the $x$ variable to help the the MLPs learn better [74]. These were inserted before the input stage of an MLP with architecture $(512, 256, D)$. We use $T = 1$ in all experiments. More details about specific experiments are given in the respective figure captions.

## E.1 Diffusion models

We use three kinds of diffusion processes to experiment with different entropy production profiles in diffusion models. These are the VP, VPx, and Straight Line (SL) SDEs from Eq. (24) and Eq. (20) respectively. These are Ornstein-Uhlenbeck processes, which have the general form [29]

$$\mathrm{d}Y_s = f(s)Y_s\mathrm{d}s + \sigma(s)\mathrm{d}B_s. \tag{85}$$

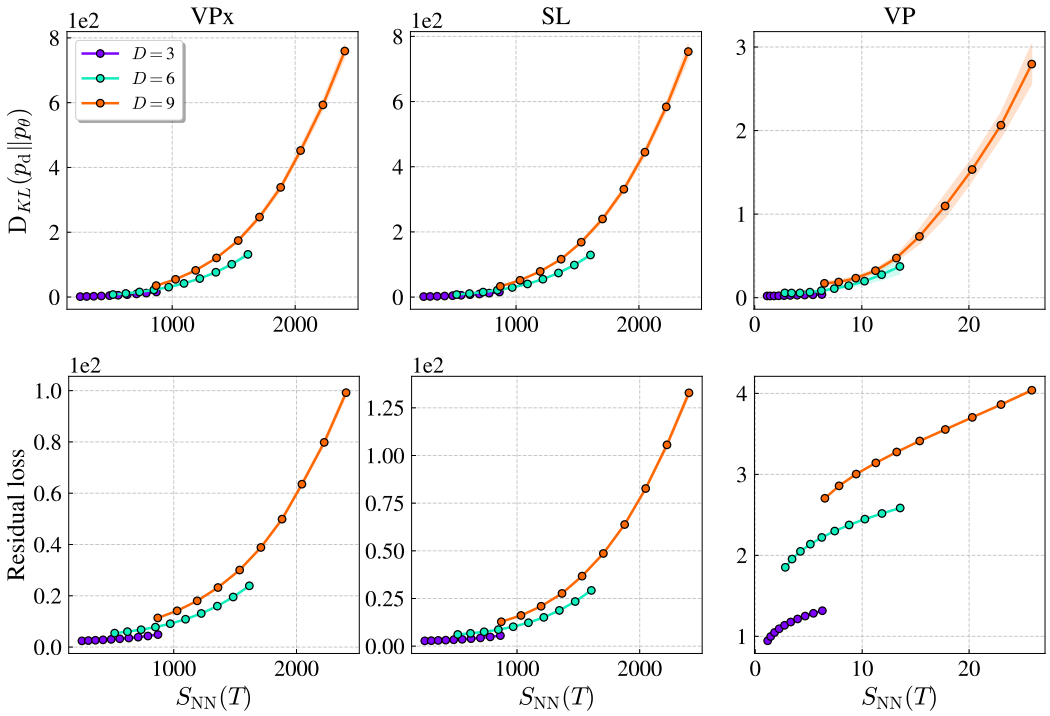



Figure 11: KL and residual loss at $n_\mathrm{ep} = 200$ epochs vs. the neural entropy in the network. The plots are generated with the same experimental setup as Fig. 10, but we vary the total entropy produced by using increasingly broader $p_\mathrm{d}$'s.

The perturbation kernel of this SDE is

$$p(y_s|y_0) = \mathcal{N}\left(y_s; \mu(s)y_0, \Sigma(s)^2 \mathbb{1}_D\right), \tag{86}$$

where

$$\mu(s) = \exp\left(\int_0^s \mathrm{d}\bar{s} f(\bar{s})\right), \tag{87a}$$

$$\Sigma(s)^2 = \mu(s)^2 \int_0^s \mathrm{d}\bar{s} \frac{\sigma(\bar{s})^2}{\mu(\bar{s})^2}. \tag{87b}$$

Starting at $s = 0$, a sample $y_\mathrm{d} \sim p_\mathrm{d}$ is propagated to

$$y_s = \mu(s)y_\mathrm{d} + \Sigma(s)\epsilon \tag{88}$$

at an intermediate time $s \in (0, T]$, where $\epsilon \sim \mathcal{N}(0, \mathbb{1}_\mathrm{D})$. The object

$$\nabla \log p(y_s|y_\mathrm{d}) = -\frac{\epsilon}{\Sigma(s)} \tag{89}$$

is used in the *denoising* entropy-matching objective

$$\mathcal{L}_\mathrm{DEM} := T\, \mathbb{E}_{y_\mathrm{d}\sim p_\mathrm{d}} \mathbb{E}_{s\sim\mathcal{U}(0,T)} \left[\Lambda(s) \frac{\sigma(s)^2}{2} \mathbb{E}_{y_s\sim p(y_s|y_\mathrm{d})} \left\| \nabla \log p_\mathrm{eq}^{(s)} + \boldsymbol{e}_{\boldsymbol{\theta}} - \nabla \log p(y_s|y_\mathrm{d}) \right\|^2\right]. \tag{90}$$

It is straightforward to show that $\mathcal{L}_\mathrm{DEM}$ is equivalent to the upper bound in Eq. (17) when $\Lambda(s) = 1$ [76]. For instance, plugging $\boldsymbol{s}_{\boldsymbol{\theta}} = \nabla \log p_\mathrm{eq}^{(t)} + \boldsymbol{e}_{\boldsymbol{\theta}}$ into the derivation in App. A of [40] would suffice. Empirically, it has been found that the choice $\Lambda(s) = 2\Sigma(s)^2/\sigma(s)^2$ produces better results in image models [5], although [20] reports that $\Lambda(s) = 1$ gives better log densities when used in conjunction with importance sampling. We have used the prescription from [5] with $s$ sampled from the uniform distribution $\mathcal{U}(0, T)$ with no importance sampling. In the experiments shown in Figs. 2 and 10 each $y_\mathrm{d}$ is evolved to 10 random values of $s$ from this interval, which improved KL estimates whilst also

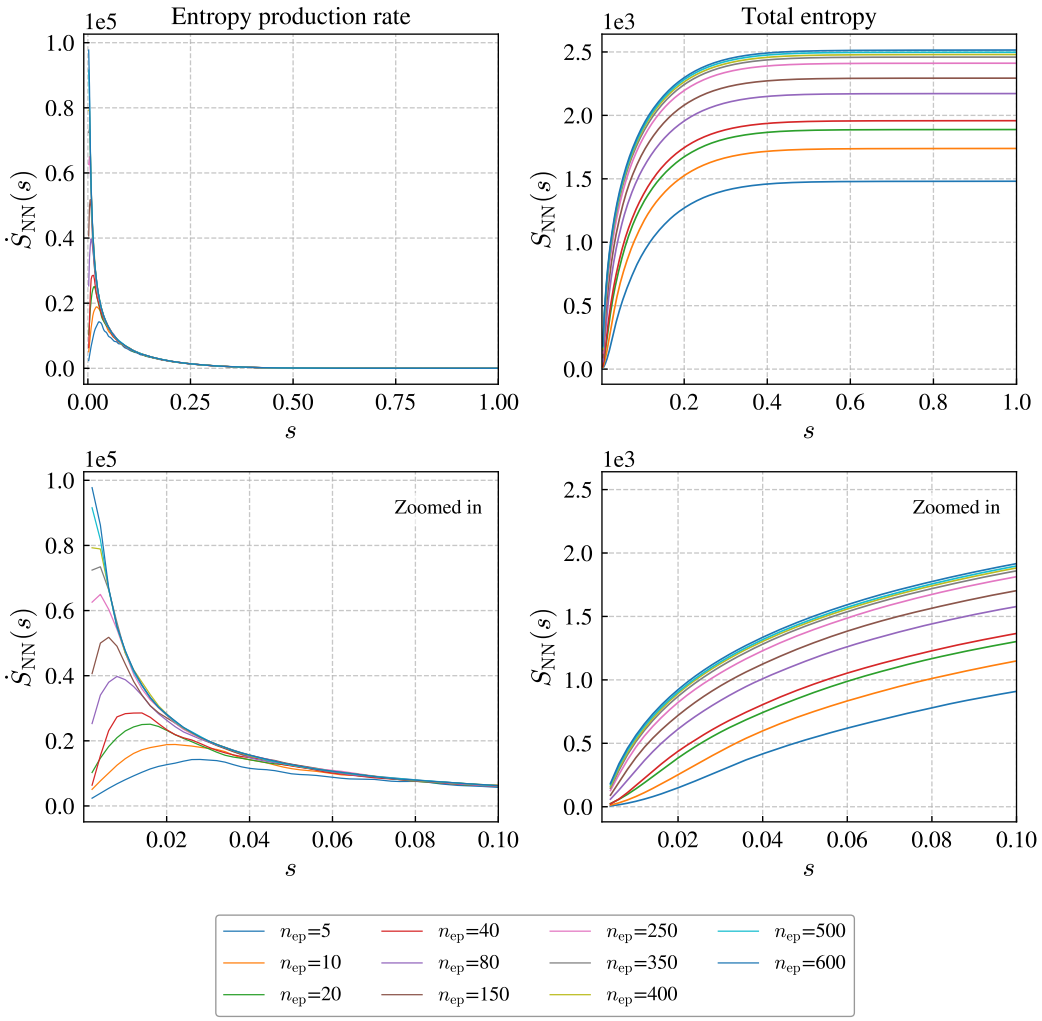



Figure 12: Entropy production curves for an image diffusion model trained on the entire MNIST dataset ($n_c = 6000$ from Fig. 3) with the VP process. The different lines correspond to entropy measurements at different epochs of training (see Fig. 2). The lower panels zoom in on a time interval close to the start of the forward diffusion process. Notice how the entropy production rate is sharply peaked near $s = 0$. This is due to the dimensionality of the data manifold $\mathcal{M}_{\mathrm{d}}$. Since $\mathcal{M}_{\mathrm{d}}$ is much lower dimensional than the ambient pixel space the model must supply a large amount of information as $t \to T$ (or $s \to 0$) to place the samples on $\mathcal{M}_{\mathrm{d}}$. The same behavior also appears in score-matching models, but it is often conflated with a numerical divergence at $s = 0$ due to the vanishing of $\Sigma(s)$ in Eq. (89) [75]. The latter is addressed by truncating the diffusion process at $s = 10^{-5}$ [6]. These entropy production rates are computed by splitting the interval $(10^{-5}, T]$ into 500 time steps and computing the average in Eq. (23) over 1000 samples from the test dataset propagated to each step.

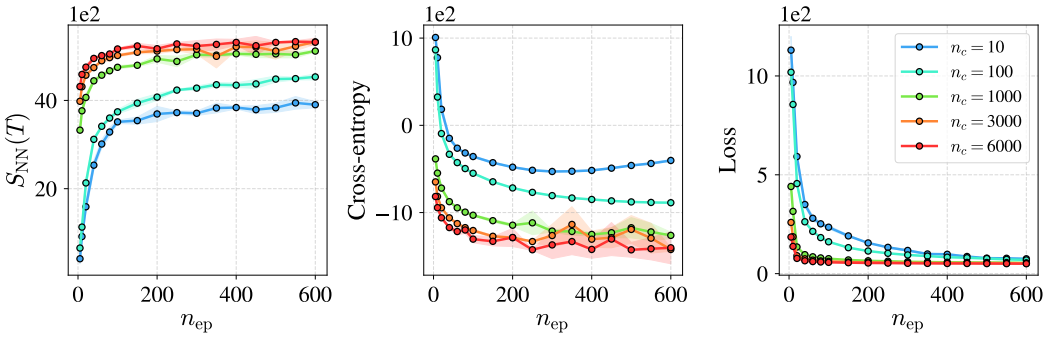



Figure 13: The evolution of neural entropy, cross-entropy, and loss over training epochs for an unconditional image diffusion model (SL) trained on the MNIST dataset. This is the analog of Fig. 3 for the Straight Line diffusion model (cf. Eq. (20)). Notice that a far greater amount of neural entropy is produced here compared to the VP process, for reasons explained in Sec. 5. However, the cross entropy settles to similar values for both VP and SL. The scaling of $S_{\mathrm{NN}}(T)$ with the number of samples, at $n_{\mathrm{ep}} = 600$, still exhibits a nearly logarithmic trend, just like with the VP process (see Fig. 16).

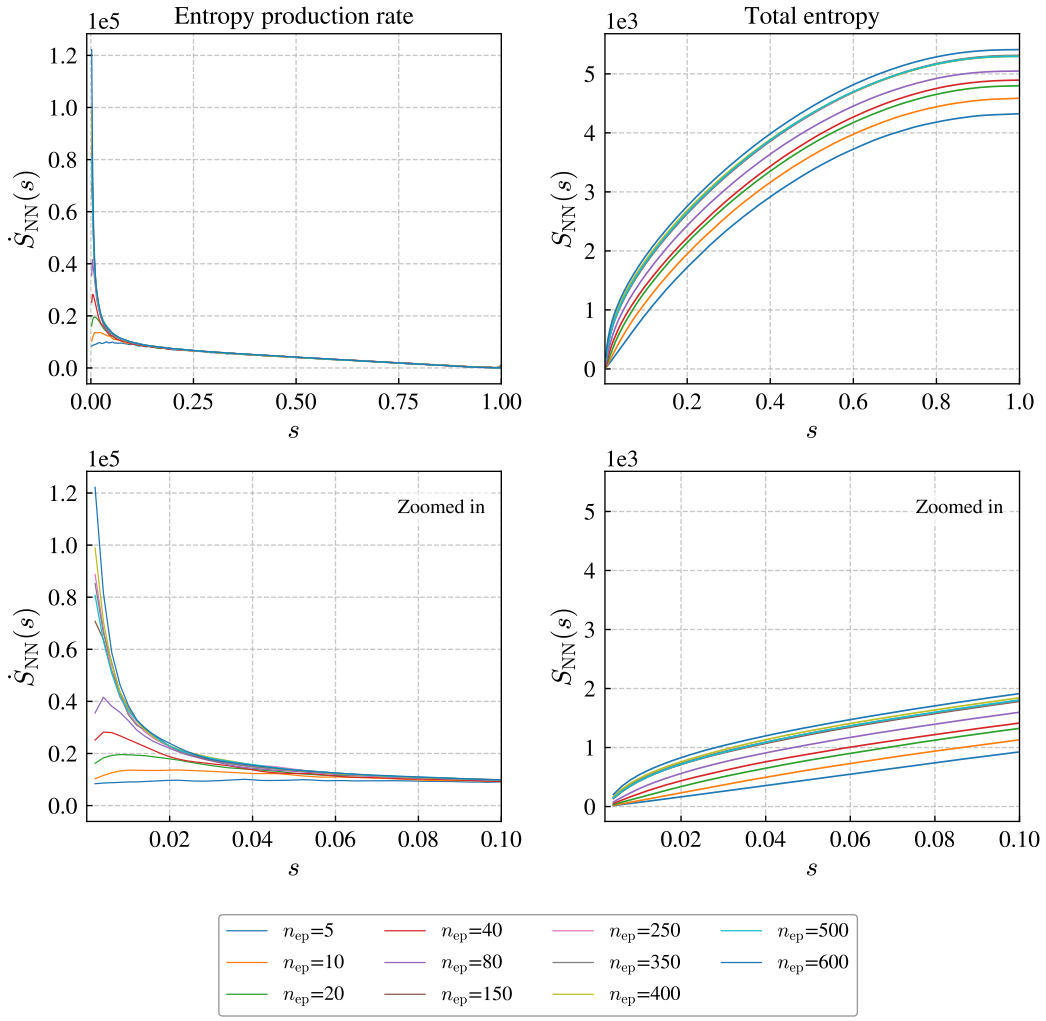



Figure 14: Entropy production curves for an image diffusion model trained on the entire MNIST dataset ($n_c = 6000$ from Fig. 13) with the SL process.

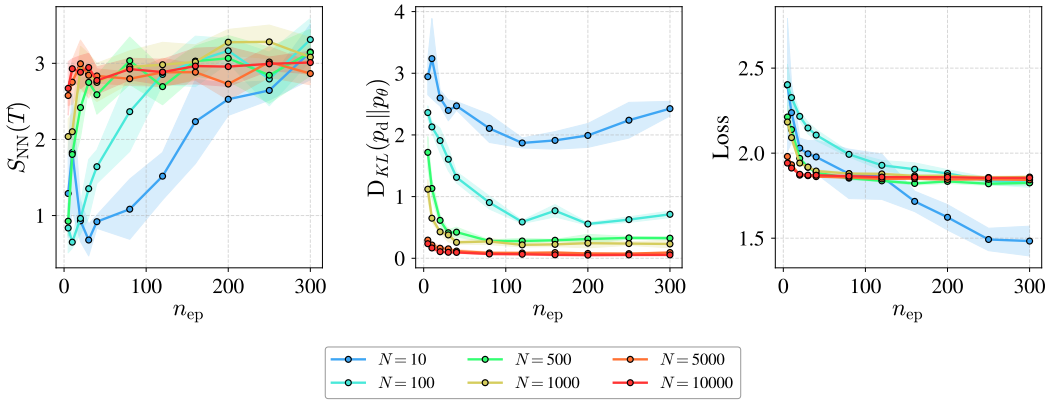



Figure 15: The evolution of neural entropy, cross-entropy, and loss over training epochs for diffusion model (VP) with an MLP core trained on a mixture of five Gaussians in $D = 6$ dimensions. $N$ is the number of samples used for training. The scaling of $S_{\mathrm{NN}}(T)$ with the number of samples does *not* show a neat trend like the ones for the image models (see Fig. 16). Due to the unstructured nature of the data and the relatively weak prior constraints imposed by the MLP, the model learns different distributions at different $N$. This is most emphatic for $N = 10$ where the data is so sparse that the model increasingly concentrates probability mass around the given samples as training progresses. This is why the KL rises and loss continues to drop for $N = 10$.

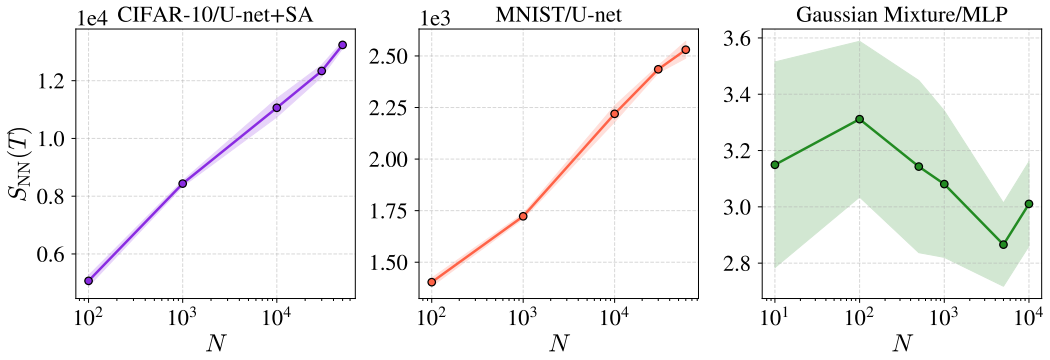



Figure 16: Neural entropy versus number of samples for CIFAR-10 trained on a U-net with self-attention layers (left), MNIST trained on a simple U-net (center) (cf. Fig. 3), and mixture of Gaussians in $D = 6$ trained on an MLP-based diffusion model (right) (cf. Fig. 15). These are the values of $S_{\mathrm{NN}}(T)$ at the end of training. The first two plots are the same ones from Fig. 1. Note the absence of the logarithmic trend in the Gaussian mixture/MLP case. All models shown here use the VP process.

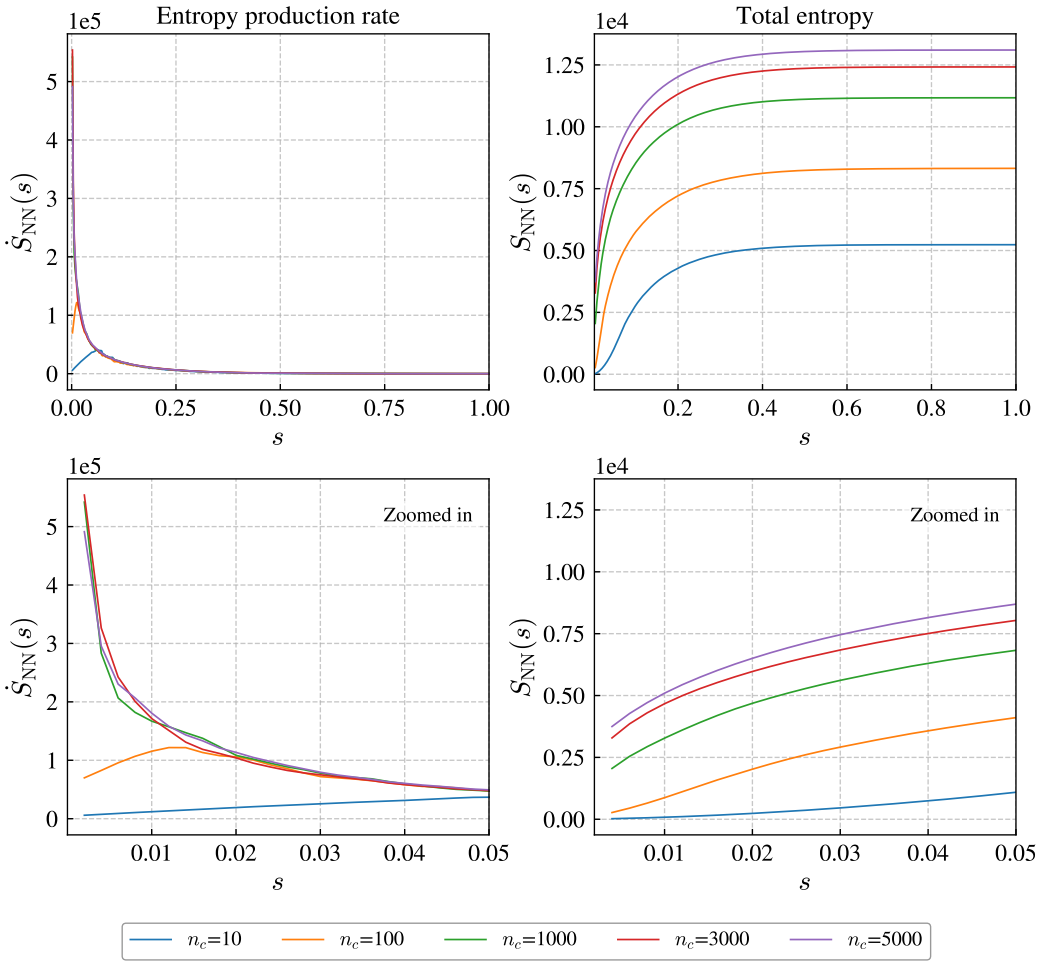



Figure 17: Entropy production curves for CIFAR-10. Note that the different colors correspond to the different number of samples per class used for training, $n_c$, rather than the number of epochs. All CIFAR-10 experiments were trained to 200 epochs. Computing the neural entropy in this case requires special care since the peak near $s = 0$ is even sharper than the ones for MNIST (see Fig. 12); the lower-dimensional data manifold with the CIFAR-10 images lives in a much higher-dimensional pixel space compared to MNIST.

reducing loss fluctuations [77, 78]. But in the image models we sampled at just one random $s$ per $y_{\mathrm{d}}$ per epoch.

The functions $\mu(s)$ and $\Sigma(s)$ for the VPx and SL processes can be read off from Eq. (25) and Eq. (21) respectively. In both cases $\Sigma(s)$ vanishes at $s = 0$, so Eq. (89) diverges at that instant. Therefore we do not venture below $s = 10^{-5}$ when training with Eq. (90). The SL SDE has an additional singularity at $s = T = 1$, so we also cut off samples at $s = 1 - 10^{-5}$ in that case. Note that [31] approximates $\Sigma(s)_{\mathrm{SL}} \approx \sigma_0$ since $\sigma_0$ is small, but we retain the full time-dependence in our experiments.

New samples from a diffusion model can be generated efficiently using the Probability Flow ODE [6, 79, 80],

$$\mathrm{d}x(t) = -\left(b_+(x, T-t) - \frac{\sigma^2}{2}\nabla \log p(x,t)\right)\mathrm{d}t, \tag{91}$$

where the true score is approximated by $\nabla \log p \approx \nabla \log p^{(t)}_{\mathrm{eq}} + \boldsymbol{e}_{\boldsymbol{\theta}}$ in entropy-matching models.

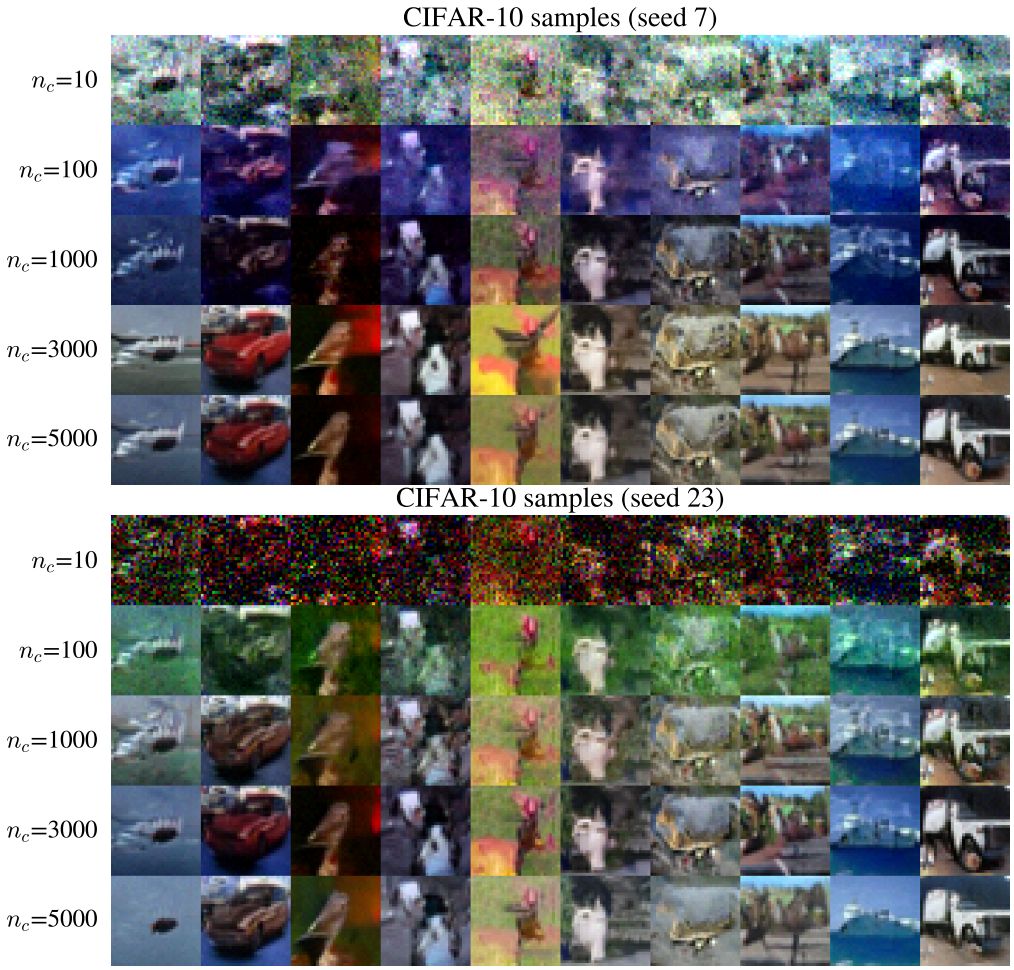



Figure 18: A few samples generated from the conditional diffusion model trained on CIFAR-10 images. Each row in a grid contains one image from each class, and the rows correspond to models trained with different numbers of samples $n_c$ per class. We used the probability flow ODE to produce these images (cf. Eq. (91)), with the *same* ten initial noise vectors for each $n_c$. These samples affirm the key takeaway from the neural entropy vs. $N$ trends Figs. 1 and 16: the rate of additional information absorbed by the model decreases with each new sample. The two grids differ by the seed value used to initialize model weights and fix the order in which the training samples are applied.

### E.2 Density estimation

In our experiments with Gaussian mixtures and MNIST, we compute the KL divergence to the true distribution and the cross-entropy respectively to gauge mode performance (cf. Figs. 3, 13 and 15). But diffusion models do *not* give exact log densities on the learned distribution, despite claims in the literature. However, a lower bound on the log density $\log p_{\boldsymbol{\theta}}(x, T)$ can be established from Eq. (61). That latter is, explicitly,

$$\log p_u(x, T) \geq -\mathbb{E}\left[\int_0^T \mathrm{d}s \left(\frac{1}{2\sigma^2}\left\|b_+ - u\right\|^2 - \nabla \cdot u\right) - \log p_u(Y_T, 0) \middle| Y_0 = x\right]. \tag{92}$$

The expectation is computed over trajectories generated by Eq. (40) that start at $x$ at $s = 0$. The bound is saturated if $u = b_-$, in which case $p_u(x, T) = p_{\mathrm{d}}(x)$ (cf. App. B.4), but $u$ is approximated by a neural network in diffusion models. We can use integration by parts to avoid taking the gradient

of the neural network in the $\nabla \cdot u$ term. For a vector-valued function $h(y_s, s)$,

$$\begin{aligned}\mathbb{E}_{Y_s}\left[\nabla \cdot h(Y_s, s)|Y_0 = x\right] &= -\int \mathrm{d}y_s\, h(y_s, s) \cdot \nabla p(y_s, s|x, 0) \\ &= -\mathbb{E}_{Y_s}\left[h(Y_s, s) \cdot \nabla \log p(Y_s, s|Y_0, 0)|Y_0 = x\right],\end{aligned} \tag{93}$$

where $p(y_s, s|x, 0)$ is the same kernel from Eq. (89) with the time-dependence indicated explicitly, and we have assumed that the product $hp$ vanishes at the $x$-boundaries. Using Eq. (93) in Eq. (92) we obtain

$$\begin{aligned}&\log p_u(x, T) \geq \\ &\quad -\mathbb{E}\left[\int_0^T \mathrm{d}s \left(\frac{1}{2\sigma^2}\left\|b_+ - u\right\|^2 + u \cdot \nabla \log p(Y_s, s|Y_0, 0)\right) - \log p_u(Y_T, 0)\middle| Y_0 = x\right].\end{aligned} \tag{94}$$

By transferring the gradient operator from $u$ we avoid the need to take derivatives of the neural network; since the transition probability is a Gaussian the gradients of their log are easy to calculate. The r.h.s. is a path integral, which can estimate efficiently as a Monte Carlo average

$$\log p_u(x, T) \geq -T\, \mathbb{E}_{s\sim\mathcal{U}(0,T)}\mathbb{E}_{y_s\sim p(y_s,s|x,0)}\left[\frac{1}{2\sigma^2}\left\|b_+ - u\right\|^2 + u \cdot \nabla \log p(y_s, s|x, 0)\right] - S_{\mathrm{G}}[p_0]. \tag{95}$$

Here, we have replaced $\mathbb{E}_{y_T\sim p(y_T,T|x,0)}\left[\log p_u(y_T)\right]$ with the negative Gibbs entropy $-S_{\mathrm{G}}[p_0]$ since $y_T$ would be distributed as $p_0$ irrespective of the $x$ at which it started, to a very good approximation (cf. App. C.2). Finally, for entropy-matching models, $u = -b_+ - \sigma^2 \boldsymbol{e}_{\boldsymbol{\theta}}$, and therefore

$$\begin{aligned}\log\, p^{\mathrm{em}}_{\boldsymbol{\theta}}(x, T) \geq &- S_{\mathrm{G}}[p_0] \\ &-T\, \mathbb{E}_{s\sim\mathcal{U}(0,T)}\mathbb{E}_{y_s\sim p(y_s,s|x,0)}\left[\frac{\sigma^2}{2}\left\|\nabla \log p^{(t)}_{\mathrm{eq}} + \boldsymbol{e}_{\boldsymbol{\theta}}\right\|^2 - (b_+ + \sigma^2 \boldsymbol{e}_{\boldsymbol{\theta}}) \cdot \nabla \log p(y_s, s|x, 0)\right].\end{aligned} \tag{96}$$

We use this lower bound in lieu of the true neural log densities in our calculations. The MC average must be computed with a fairly large number of samples [77, 78]. For the Gaussian mixture experiments, this is an excellent substitution. The results are noisy but still insightful in the image experiments. A version of Eq. (96) for score-matching is derived in [20], which is obtained by setting $\boldsymbol{e}_{\boldsymbol{\theta}} = -\nabla \log p^{(t)}_{\mathrm{eq}} + \boldsymbol{s}_{\boldsymbol{\theta}}$ in this one.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the goal of the paper and the main contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our work are discussed inline in Sec. 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions and proofs are given in detail in App. B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details of the experiments are discussed in Sec. 6 and App. E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code for all experiments is available here.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training details are given in Sec. 6, and under the figures in App. E. More detail will be provided with the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experiments in this paper, except for the ones in Fig. 9 are computed over four different initializations of the respective model weights. The plots indicate the region of one standard deviation from the mean results, when relevant.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details are given in App. E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no direct societal impact that we can see.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines: Not applicable.

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the creators of MNIST and CIFAR-10.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.