PERFORMANCE LIMITS OF SCORE-BASED GENERATIVE MODELS VIA STOCHASTIC THERMODYNAMICS

Nathan X. Kodama
Case Western Reserve University
Cleveland, OH 44106
nxk281@case.edu

Michael Hinczewski
Case Western Reserve University
Cleveland, OH 44106
mxh605@case.edu

ABSTRACT

We connect score-based generative models and stochastic thermodynamics by deriving performance limits expressed in terms of entropy rates. Our main theoretical result is a lower bound on the negative log-likelihood that relates model accuracy to the entropy rates induced by the learned score and the entropies of the data and noise distributions. We numerically validate the lower bound and show that improved model accuracy is accompanied by increased entropy production, revealing an explicit accuracy–dissipation tradeoff. Finally, we use the lower bound to estimate the differential entropy of data directly from the trained score network. Together, these results provide physically interpretable limits, practical empirical probes, and a unified theoretical framework linking generative modeling to fundamental principles of stochastic thermodynamics, with implications for controllable generative modeling and emerging computing hardware.

1 INTRODUCTION

Score-based diffusion models have achieved remarkable success in generative modeling by learning to reverse a stochastic diffusion process (Song et al., 2021). Recent advances have exploited physical connections to optimal transport (Kwon et al., 2022; Lipman et al., 2022), critical damping (Dockhorn et al., 2022), and heat dissipation (Rissanen et al., 2023) to achieve significant performance gains, while others have connected generative processes to Maxwell's demon (Premkumar, 2025) and thermodynamic hardware (Coles et al., 2023).

Expanding on pioneering work connecting deep learning with non-equilibrium thermodynamics (Sohl-Dickstein et al., 2015), recent work has highlighted fundamental connections between these models and stochastic thermodynamics, including speed–accuracy tradeoffs derived from entropy production (Ikeda et al., 2025) and related energy–speed–accuracy limits in driven nonequilibrium systems Klinger & Rotskoff (2025). Our contribution is complementary: we focus on formalizing the analogy to Maxwell's demon and deriving a thermodynamically motivated lower bound on the negative log-likelihood (NLL).

Prior variational treatments provide an evidence lower bound (ELBO) on the log-likelihood (Huang et al., 2021), which is equivalently an upper bound on NLL, and some analyses give upper bounds on $\mathrm{KL}(p_{\mathrm{data}} \| p_{\mathrm{model}})$ (Premkumar, 2025), again implying upper bounds on NLL. In contrast, under a consistent plug-in convention where system entropy rates $\dot{S}_{\boldsymbol{\theta}}(t)$ are computed from the learned score, we derive a thermodynamic lower bound on NLL

$$\mathrm{NLL} \;\geq\; \frac{S_{\text{data}} + S_{\text{noise}}}{2} - \frac{1}{2}\int_0^1 \dot{S}_{\boldsymbol{\theta}}(t)\, dt,$$

where S_{data} is the entropy of the data and S_{noise} that of the equilibrium distribution. Note, a trivial bound $\mathrm{NLL} \geq S_{\text{data}}$ follows from $\mathrm{KL} \geq 0$: our result strengthens it via S_{noise} and entropy rate corrections. Because NLL is a widely reported performance metric for diffusion models, this inequality gives a clear limit on achievable performance: no training or sampling procedure can reduce NLL below this thermodynamically motivated floor, distinguishing our bound from the ELBO- and KL-based bounds.

Our main contributions are as follows:

- We derive a lower bound on the negative log-likelihood (NLL) that links achievable generative model performance directly to entropy rates of the diffusion process, providing a thermodynamically motivated floor that complements existing ELBO-based upper bounds.
- Using this decomposition, we introduce a practical method for estimating dataset entropy from the trained score network, and demonstrate empirically that improved score accuracy yields both lower NLL and tighter thermodynamic performance bounds.
- We demonstrate that improving generative accuracy is accompanied by increased integrated entropy production, revealing an explicit accuracy–dissipation tradeoff in score-based diffusion models and providing a thermodynamic interpretation of performance scaling with model capacity.

Together, these results establish a unified framework that connects generative model accuracy, entropy rates from stochastic thermodynamics, and energy dissipation in score-based diffusion models.

2 BACKGROUND

2.1 SCORE-BASED GENERATIVE MODELS

Score-based diffusion models learn to reverse a forward diffusion process. We consider the forward stochastic process $\mathbf{x}_t \in \mathbb{R}^d$ governed by the Itô SDE:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + \mathbf{G}(\mathbf{x}_t, t)d\mathbf{w}_t, \quad t \in [0, T],$$

where $\mathbf{f}(\mathbf{x}_t, t) : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the deterministic drift vector, $\mathbf{G}(\mathbf{x}_t, t) : \mathbb{R}^d \times [0, T] \to \mathbb{R}^{d\times m}$ is the stochastic diffusion matrix, and $\mathbf{w}_t$ is an m-dimensional standard Wiener process. The reverse-time diffusion process $\overline{\mathbf{x}}_t := \mathbf{x}_\tau$ with $\tau = T - t$ can be derived (Anderson, 1982; Haussmann & Pardoux, 1986; Song et al., 2021):

$$d\overline{\mathbf{x}}_\tau = \left[-\mathbf{f}\left(\overline{\mathbf{x}}_\tau, T-\tau\right) + 2\mathbf{D}(\overline{\mathbf{x}}_\tau, T-\tau)\nabla_{\overline{\mathbf{x}}_\tau} \log p_\tau\left(\overline{\mathbf{x}}_\tau\right)\right] d\tau + \mathbf{G}\left(\overline{\mathbf{x}}_\tau, \tau\right) d\mathbf{w}_\tau$$

where $\mathbf{D}(\overline{\mathbf{x}}_\tau, T-\tau) = \frac{1}{2}\mathbf{G}\left(\overline{\mathbf{x}}_\tau, T-\tau\right)\mathbf{G}\left(\overline{\mathbf{x}}_\tau, T-\tau\right)^\top$ and $\nabla_{\overline{\mathbf{x}}_\tau} \log p_\tau\left(\overline{\mathbf{x}}_\tau\right)$ is called the score function of the marginal distribution over $\overline{\mathbf{x}}_\tau$. Score-based diffusion models use a deep neural network to approximate the score function: $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, \tau) \approx \nabla_{\overline{\mathbf{x}}_\tau} \log p_\tau\left(\overline{\mathbf{x}}_\tau\right)$.

The reverse-time process can be used as a generative model. In particular, (Song et al., 2021) model data $\mathbf{x}$, setting $p\left(\mathbf{x}_0\right) = p_{\text{data}}(\mathbf{x})$. Currently, diffusion models (Song et al., 2021) have drift and diffusion coefficients of the simple form $\mathbf{f}\left(\mathbf{x}_t, t\right) = f(t)\mathbf{x}_t$ and $\mathbf{G}\left(\mathbf{x}_t, t\right) = g(t)\mathbf{I}_d$. Generally, $\mathbf{f}$ and $\mathbf{G}$ are chosen such that the marginal, equilibrium density is approximately normal at time T, i.e., $p\left(\mathbf{x}_T\right) \approx \mathcal{N}\left(\mathbf{0}, \mathbf{I}_d\right)$. We can then initialize $\mathbf{x}_0$ based on a sample drawn from a complex data distribution, corresponding to a far-from-equilibrium state. While the state $\mathbf{x}_0$ relaxes towards equilibrium via the forward diffusion, we can learn a model $\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)$ for the score $\nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t\right)$, which can be used for generation via the reverse process. If $\mathbf{f}$ and $\mathbf{G}$ take the simple form from above, the unweighted denoising score matching (Vincent, 2011) objective for this task is:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}[0,T]}\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0)}\mathbb{E}_{\mathbf{x}_t\sim p_t(\mathbf{x}_t|\mathbf{x}_0)}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) - \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right\|_2^2\right]$$

2.2 ENTROPY RATES IN STOCHASTIC THERMODYNAMICS

In stochastic thermodynamics, irreversibility is quantified by entropy production along a stochastic process. Recent work has applied these principles to generative models, showing that entropy production constrains achievable speed and accuracy (Ikeda et al., 2025). Let p_t denote the marginal density of $\mathbf{x}_t$ and define the differential entropy

$$S(t) \;:=\; -\mathbb{E}_{p_t}\big[\log p_t(\mathbf{x}_t)\big], \qquad \dot{S}(t) \;=\; \frac{d}{dt}S(t). \tag{1}$$

For a stochastic process, the system entropy rate $\dot{S}(t)$ decomposes as (Seifert, 2012)

$$\dot{S}(t) \;=\; \dot{S}^{\,i}(t) \;+\; \dot{S}^{\,e}(t), \tag{2}$$

where $\dot{S}^i(t) \geq 0$ is the intrinsic entropy production rate (dissipation or irreversibility), and $\dot{S}^e(t)$ is an exchange entropy rate (entropy flow between the system and its surroundings).

For an overdamped SDE with homogeneous scalar noise, $d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\, dt + g(t) d\mathbf{w}_t$, the probability current is defined as $\mathbf{J}(\mathbf{x}, t) = p_t(\mathbf{x})\big(\mathbf{f}(\mathbf{x}, t) - D(t)\mathbf{s}(\mathbf{x}, t)\big)$ and the intrinsic entropy production rate has the form

$$\dot{S}^i(t) = \int \frac{\|\mathbf{J}(\mathbf{x}, t)\|^2}{D(t)\, p_t(\mathbf{x})}\, d\mathbf{x} = \frac{1}{D(t)}\, \mathbb{E}_{p_t}\Big[\big\|\mathbf{f}(\mathbf{x}_t, t) - D(t)\, \mathbf{s}(\mathbf{x}_t, t)\big\|^2\Big], \tag{3}$$

which grows when the dynamics drive large probability currents.

A key identity we use later is a simplified expression for the *system* entropy rate:

$$\dot{S}(t) = \mathbb{E}_{p_t}\big[\nabla_{\mathbf{x}} \cdot \mathbf{f}(\mathbf{x}_t, t)\big] + D(t)\, \mathbb{E}_{p_t}\big[\|\mathbf{s}(\mathbf{x}_t, t)\|^2\big]. \tag{4}$$

In generative models, the true score $\mathbf{s}(\mathbf{x}, t)$ is approximated by a neural network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$. We therefore define the model-based entropy rates by substituting $\mathbf{s}_{\boldsymbol{\theta}}$ into the above expressions, e.g.

$$\dot{S}_{\boldsymbol{\theta}}(t) := \mathbb{E}_{p_t}\big[\nabla_{\mathbf{x}} \cdot \mathbf{f}(\mathbf{x}_t, t)\big] + D(t)\, \mathbb{E}_{p_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\|^2\big], \tag{5}$$

and similarly for $\dot{S}^i_{\boldsymbol{\theta}}(t)$ and $\dot{S}^e_{\boldsymbol{\theta}}(t)$. These quantities are directly computable from the trained score network and will appear in our NLL bound and experiments.

3 Main Results

3.1 Lower Bound on Negative Log-Likelihood

For an approximate score function $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, T - \tau)$, the negative log-likelihood (NLL) satisfies

$$\boxed{\mathrm{NLL} - S_{\text{data}} \geq \frac{1}{2}\left[S_{\text{noise}} - S_{\text{data}} - \int_0^1 \dot{S}_{\boldsymbol{\theta}}(T - \tau)\, d\tau\right],} \tag{6}$$

where S_{data} is the entropy of the data distribution, S_{noise} that of the equilibrium (prior), and $\dot{S}_{\boldsymbol{\theta}}$ the entropy rate defined by the learned score function. The trivial bound $\mathrm{NLL} \geq S_{\text{data}}$ follows directly from $\mathrm{NLL} = S(p_{\text{data}}, p_{\boldsymbol{\theta}}) \geq S(p_{\text{data}}) = S_{\text{data}}$, i.e. from the non-negativity of $\mathrm{KL}(p_{\text{data}} \| p_{\boldsymbol{\theta}})$. Equality holds only if $p_{\boldsymbol{\theta}} = p_{\text{data}}$; our result strengthens it by incorporating S_{noise} and entropy rate corrections. Briefly, the bound comes from the definition of NLL via the probability flow ODE, followed by applying polarization and Stein's identities combined with the score-based definition of entropy rates.

Derivation sketch. Let $\tilde{\mathbf{f}}_{\boldsymbol{\theta}}(\mathbf{x}, t) := \mathbf{f}(\mathbf{x}, t) - D(t)\, \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$ denote the probability-flow ODE vector field associated with the forward SDE Song et al. (2021). By the instantaneous change-of-variables formula Chen et al. (2018),

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \log p_1(\mathbf{x}_1) + \int_0^1 \Big(\nabla \cdot \mathbf{f}(\mathbf{x}_t, t) - D(t)\, \nabla \cdot \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\Big)\, dt. \tag{7}$$

Taking expectation over $\mathbf{x}_0 \sim p_{\text{data}}$ yields the data-average log-likelihood, and we define $\mathrm{NLL} := -\mathbb{E}_{p_{\text{data}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}_0)]$.

To connect to thermodynamic quantities, we rewrite the score-divergence term using Stein's identity $\mathbb{E}_{p_t}[\nabla \cdot \mathbf{s}_{\boldsymbol{\theta}}] = -\mathbb{E}_{p_t}[\mathbf{s}_{\boldsymbol{\theta}} \cdot \mathbf{s}_{\text{true}}]$ (where $\mathbf{s}_{\text{true}} = \nabla_{\mathbf{x}} \log p_t$), and then apply the polarization identity $\mathbf{s}_{\boldsymbol{\theta}} \cdot \mathbf{s}_{\text{true}} = \frac{1}{2}(\|\mathbf{s}_{\boldsymbol{\theta}}\|^2 + \|\mathbf{s}_{\text{true}}\|^2 - \|\mathbf{s}_{\boldsymbol{\theta}} - \mathbf{s}_{\text{true}}\|^2)$. This yields the decomposition

$$\mathrm{NLL} \;=\; \frac{S_{\text{data}} + S_{\text{noise}}}{2} \;-\; \frac{1}{2}\int_0^1 \dot{S}_{\boldsymbol{\theta}}(t)\, dt \;+\; \frac{1}{2}\int_0^1 D(t)\, \mathbb{E}_{p_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \mathbf{s}_{\text{true}}(\mathbf{x}_t, t)\|^2\big]\, dt. \tag{8}$$

The last term in Eq. 8 is nonnegative: we immediately obtain the lower bound in Eq. 6. Details of the derivation appear in Appendix B.

3.2 Connection to Maxwell's Demon and Entropy Rates

The Maxwell's Demon thought experiment involves an external controller that selectively manipulates systems to lower their entropy. Score-based models operate analogously to Maxwell's Demon: the neural network measures the system state during training (forward process) and uses this information to decrease entropy during generation (reverse process).

We consider the special case of drift-less diffusion, $d\mathbf{x}_t = g(t)d\mathbf{w}_t$. For a score network that reverses drift-less diffusion, the intrinsic entropy production rate is

$$\dot{S}^i_{\boldsymbol{\theta}}(T-\tau) = \frac{g(T-\tau)^2}{2}\mathbb{E}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}\left(\overline{\mathbf{x}}_\tau, T-\tau\right)\right\|^2\right]. \tag{9}$$

While the drift-less forward process has no exchange entropy, the reverse process has exchange-entropy rate that is

$$\dot{S}^{\mathrm{e}}_{\boldsymbol{\theta}}(T-\tau) = \mathbb{E}\left[\nabla_{\overline{\mathbf{x}}_\tau}\cdot\tilde{\mathbf{f}}_{\boldsymbol{\theta}}\left(\overline{\mathbf{x}}_\tau, T-\tau\right)\right].$$

For the score network controlled-forward process (see Appendix C.2), the drift is $\tilde{\mathbf{f}}_{\boldsymbol{\theta}}\left(\overline{\mathbf{x}}_\tau, T-\tau\right) = g(T-\tau)^2\mathbf{s}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_\tau, T-\tau)$, so

$$\begin{aligned}\dot{S}^e_{\boldsymbol{\theta}}(T-\tau) &= g(T-\tau)^2\mathbb{E}\left[\nabla_{\overline{\mathbf{x}}_\tau}\cdot\mathbf{s}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_\tau, T-\tau)\right]\\ &= -g(T-\tau)^2\mathbb{E}\left[||\mathbf{s}_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_\tau, T-\tau)||^2\right] = -2\dot{S}^i_{\boldsymbol{\theta}}(T-\tau),\end{aligned}$$

where we have used Stein's identity (see Appendix B.3). Thus, the system entropy rate is

$$\begin{aligned}\dot{S}_{\boldsymbol{\theta}}(T-\tau) &= \dot{S}^i_{\boldsymbol{\theta}}(T-\tau) + \dot{S}^e_{\boldsymbol{\theta}}(T-\tau)\\ &= \dot{S}^i_{\boldsymbol{\theta}}(T-\tau) - 2\dot{S}^i_{\boldsymbol{\theta}}(T-\tau) = -\dot{S}^i_{\boldsymbol{\theta}}(T-\tau),\end{aligned}$$

which means that the score network must reverse the forward process. This connects the score model directly to thermodynamic entropy rates and the neural network's outputs to Maxwell's Demon.

4 Numerical Results

We validate our theoretical predictions using the noisy 2D Swiss roll dataset, Gaussian data, and synthetic 8-bit grayscale images with uniformly distributed pixel values between 0 and 1. We compute exact NLL values via the probability ODE framework and measure entropy rates directly from the trained score network, enabling direct comparison with our theoretical predictions. In Figure 1, the left panel exposes the relationship between the NLL and lower bound across 5 noise parameters, $\sigma \in \{10, 15, 20, 25, 30\}$, and 10 runs per noise parameter (Gaussian and Uniform data) and different model sizes (Swiss roll). The theoretical bound consistently holds across all parameters and runs, with tighter bounds correlating with better model performance. The performance gaps correspond to the squared difference term $||\mathbf{s}_{\boldsymbol{\theta}} - \mathbf{s}_{\mathrm{true}}||^2$ in the exact decomposition of the NLL. We observe strong positive correlations between the NLL and the performance gap, quantified by the Pearson coefficient ($r = 0.803$, $p < 0.001$) and Spearman coefficient ($r_s = 0.909$, $p < 0.001$).

Entropy rate estimates (intrinsic $\dot{S}^i_{\boldsymbol{\theta}}$, exchange $\dot{S}^e_{\boldsymbol{\theta}}$, and system $\dot{S}_{\boldsymbol{\theta}}$) computed from the score neural network yielding the best performance are presented in the right panel of Figure 1. These empirical measurement validate our theoretical predictions for the drift-less diffusion process: the intrinsic entropy production rate $\dot{S}^i_{\boldsymbol{\theta}}(T-\tau)$ remains positive throughout the controlled process, the exchange entropy rate maintains the predicted 2:1 ratio, $\dot{S}^e_{\boldsymbol{\theta}}(T-\tau) = -2\dot{S}^i_{\boldsymbol{\theta}}(T-\tau)$ and the system entropy rate $\dot{S}_{\boldsymbol{\theta}}(T-\tau) = -\dot{S}^i_{\boldsymbol{\theta}}(T-\tau)$ confirms that the score network successfully reverses the forward diffusion process by maintaining negative system entropy production.

For simple reference distributions (e.g., Uniform or Gaussian), S_{data} is available in closed form; it is generally unknown for structured datasets such as the Swiss roll. Here we use our lower bound framework as a practical model-based approach for estimating the differential entropy of data distributions. Rearranging the lower bound for NLL yields the model-based upper bound for S_{data}

$$S_{\mathrm{data}} \leq 2\mathrm{NLL}(\theta) + \int_0^1 \dot{S}_{\boldsymbol{\theta}}(t)dt - S_{\mathrm{noise}}\,.$$

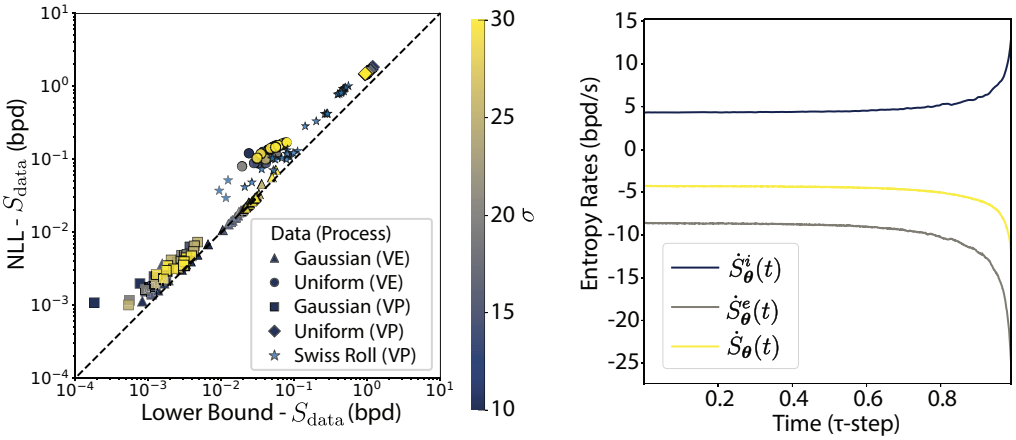

Figure 1: Comparison between the NLL and theoretical lower bound across diffusion model configurations. (Left) NLL values versus the lower bound in Eq. 6 (dashed) for Gaussian, Uniform, and Swiss roll data under different noising processes. Marker shape denotes data distribution and process, while color indicates the noise parameter $\sigma \in [10, 30]$. (Right) Entropy rates (intrinsic $\dot{S}^i_{\boldsymbol{\theta}}$, exchange $\dot{S}^e_{\boldsymbol{\theta}}$, and system $\dot{S}_{\boldsymbol{\theta}}$) estimated from the score network yielding the best NLL in the Uniform (VE) case, confirming the predicted 2:1 ratio $\dot{S}^e_{\boldsymbol{\theta}} = -2\dot{S}^i_{\boldsymbol{\theta}}$ expected for the drift-less (VE) setting.

This bound is tight in the limit of an exact score model. In this ideal limit the upper bound becomes exact and recovers the true differential entropy of the dataset. We compare the model-based upper bound on S_{data} against non-parametric entropy estimators, including the Kozachenko–Leonenko estimator using k-nearest neighbors (k-NN) Kozachenko & Leonenko (1987) and a 2D discretized histogram estimator. Details of all three approaches are provided in Appendix H.2.2.

Figure 2 visualizes the model-based upper bound on S_{data} across trained score networks, sorted high to low. Each point corresponds to a single trained model, with the color indicating the number of hidden units in the neural network. As the capacity of the model increases, the upper bound decreases. Importantly, the model-based estimates approach the independent non-parametric estimates obtained from the Kozachenko-Leonenko k-nearest-neighbors method and 2D discretized histogram method. In the remainder of the paper, we adopt the smallest observed upper bound across models as our working estimate of S_{data}, which provides a conservative and internally consistent reference for evaluating KL divergence and performance limits.

Table 1: **Comparison of data entropy estimates for the noisy 2D Swiss roll.** Estimated data entropy S_{data} (bits per dimension) for the 2D Swiss roll. We report the model-based upper bound on S_{data} minimized across all trained models, which is attained by a three-layer neural network with 256 hidden units in each layer, alongside two non-parametric estimates from the Kozachenko-Leonenko method and discretized 2D histogram. For reference, we also report the differential entropy of maximum entropy distributions under matched constraints: a Gaussian with the same empirical mean and covariance, and a Uniform distribution over the same effective support.

Estimator / reference	$\widehat{S}_{\text{data}}$ **(bpd)**
Model-Based Upper Bound	1.441
k-NN (Kozachenko–Leonenko)	1.432
2D discretized histogram	1.415
Uniform (reference)	2.472
Gaussian (reference)	2.487

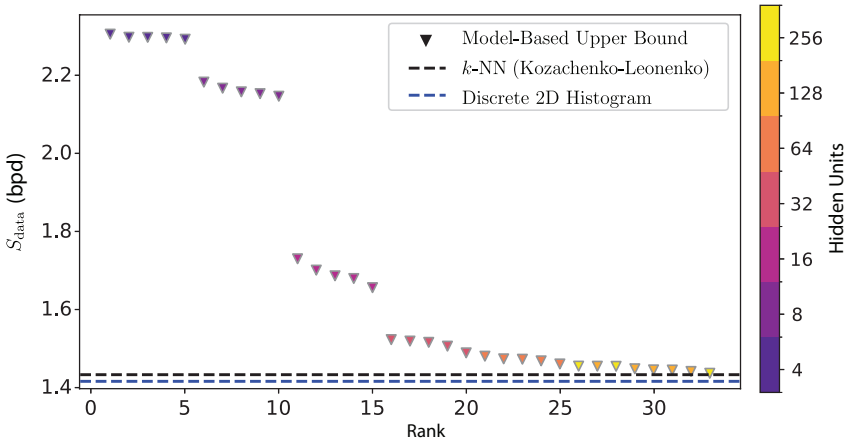

Figure 2: **Model-based upper bound on the data entropy for the noisy 2D Swiss roll.** Each point corresponds to a trained score network, with color indicating model capacity (number of hidden units). The model-based upper bound on S_{data} are obtained from the exact NLL and integrated entropy rates from the probability flow ODE. Horizontal dashed lines denote non-parameteric entropy estimates using the Kozachenko-Leonenko k-nearest-neighbors estimator and a discretized 2D histogram. Across model sizes and random seeds, the upper bound decreases towards the non-parametric estimates, indicating that improved score accuracy yields tighter entropy estimates. The minimum upper bound across models is taken as our estimate of S_{data} for the Swiss roll.

Table 1 summarizes several complementary estimates of the differential entropy S_{data} for the noisy Swiss roll distribution. Notably, both non-parametric estimators—the Kozachenko–Leonenko k-NN method and the 2D discrete histogram method—are close to the upper bound given by the trained score network, providing independent validation of the model–based upper bound. The reported upper bound corresponds to the smallest value obtained across all model sizes and seeds, and is achieved by the highest-capacity score network (256 hidden units), consistent with the expectation that improved score accuracy yields tighter bounds.

For context, the Uniform and Gaussian reference values correspond to maximum entropy distributions under matched constraints: a Uniform distribution over the same effective support, and a Gaussian distribution with the same empirical mean and covariance as the data. As expected, both reference entropies substantially exceed the Swiss roll estimates. Throughout this work, we adopt the upper model-based bound as our operational estimate of S_{data} for the Swiss roll. This choice is conservative, ensures consistency with the exact likelihood-based decomposition, and avoids optimistic bias when evaluating entropy gaps, KL divergences, and thermodynamic bounds derived from S_{data}.

Figure 3 shows how model capacity controls the achievable accuracy–dissipation tradeoff in score-based generative models. We vary the number of hidden units of a 3-layer multilayer perceptron, and within each set of 5 random initializations, we plot three quantities evaluated from the trained model—the accuracy of the generated data $D_{\text{KL}}(p_0 \| p_{\boldsymbol{\theta}}) = \text{NLL} - S_{\text{data}}$, an ELBO-derived upper bound on D_{KL} Huang et al. (2021), and our entropic lower bound—against the dissipation, $\int_0^1 \dot{S}^{ij}_{\boldsymbol{\theta}}(\tau), d\tau$. Across all model sizes and seeds, the KL divergence consistently falls between these two bounds, confirming that standard variational guarantees remain valid while our theory supplies a complementary, physically-interpretable floor. Moreover, increasing model size systematically shifts models toward lower KL and higher integrated entropy production, indicating that improved likelihood requires paying additional dissipation. The gap between the observed KL and the entropic lower bound shrinks for the best-performing models: as the score approximation improves with capacity, the lower bound tightens.

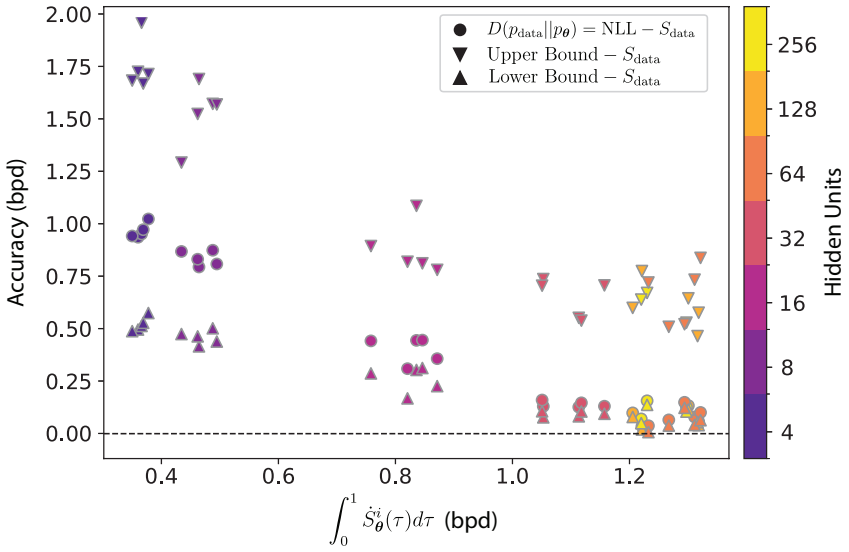

Figure 3: **Accuracy-dissipation tradeoffs across model capacity.** We plot the accuracy of the generated data, $D_{\text{KL}}(p_0 \| p_{\boldsymbol{\theta}}) = \text{NLL} - S_{\text{data}}$ (circles), against the dissipation, i.e. the intrinsic entropy production $\int_0^1 \dot{S}^i_{\boldsymbol{\theta}}(\tau)\, d\tau$ (x-axis, bpd), for a sweep of MLP score networks with different hidden layer widths (color; each point is one trained model/seed). For each run, the empirical KL divergence falls between an ELBO-derived upper bound (down-triangles) and our entropy-based lower bound (up-triangles). Overall, increased model capacity tends to improve fit (lower KL) while incurring larger entropy production, illustrating an explicit accuracy–dissipation tradeoff.

5 Conclusion

Our work establishes fundamental connections between generative modeling and entropy rates in stochastic thermodynamics, elaborating on pioneering insights connecting deep learning with non-equilibrium thermodynamics Sohl-Dickstein et al. (2015) and complementing recent analyses of speed–accuracy tradeoffs in diffusion models (Ikeda et al., 2025) and related thermodynamic energy–speed–accuracy trade-offs in driven non-equilibrium systems Klinger & Rotskoff (2025). Our theoretical framework extends on existing variational bounds (Huang et al., 2021) by deriving a fundamental limit that relates model performance directly to entropy rates in diffusion processes. Our main contributions are recapitulated as follows:

- We derive a lower bound on the NLL that links generative model performance to entropy rates of the diffusion process, complementing existing ELBO-based upper bounds.
- Using this decomposition, we introduce a model-based method for estimating the data entropy, and show empirically that improved score accuracy yields both lower NLL and tighter performance bounds.
- We find that improvements in generated data accuracy are achieved with higher entropy production, revealing an accuracy–dissipation tradeoff in score-based generative models.

Together, these results provide interpretable performance limits, physics-guided diagnostic tools, and a principled framework for analyzing generative models. We offer several application areas below.

Emerging Computing Hardware. Our results suggest fundamental limits that may be exploited in emerging computing hardware. In the current formulation, entropy rates are defined via mathematical analogy to thermodynamics. However, when realized on thermodynamic hardware (Coles et al., 2023), entropy rates become physical quantities and the bound becomes a target, extending connections to Maxwell's demon (Premkumar, 2025) into practical hardware design principles.

Accuracy-Energy-Speed Tradeoffs. Entropy rates provide new diagnostics for model behavior, complementing existing metrics with physically motivated quantities that reveal fundamental tradeoffs. In particular, the mathematical connection to Maxwell's Demon in terms of entropy rates not only

provides a conceptual framework for understanding the operation of score-based diffusion models, but also enables us to estimate the amount of entropy the score network removes from the system during the reverse process. This perspective clarifies the role of the score network and highlights entropy reduction as a measurable quantity that links model performance to physical limits.

Controlling Generation. Minimizing entropy production while maintaining model quality could lead to faster sampling and training. Connections to optimal transport theory (Kwon et al., 2022; Lipman et al., 2022) and thermodynamic uncertainty principles suggest design principles for designing more controllable and efficient diffusion models.

REFERENCES

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

Ricky T Q Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Patrick J Coles, Collin Szczepanski, Denis Melanson, Kaelan Donatella, Antonio J Martinez, and Faris Sbahi. Thermodynamic ai and the fluctuation frontier, 2023. URL https://arxiv.org/abs/2302.06584.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. 2005. ISBN 9780471241959. doi: 10.1002/047174882X.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations (ICLR)*, 4 2022.

U G Haussmann and E Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14:1188 – 1205, 1986. doi: 10.1214/aop/1176992362. URL https://doi.org/10.1214/aop/1176992362.

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 (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22863–22876. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c11abfd29e4d9b4d4b566b01114d8486-Paper.pdf.

Kotaro Ikeda, Tomoya Uda, Daisuke Okanohara, and Sosuke Ito. Speed-accuracy relations for diffusion models: Wisdom from nonequilibrium thermodynamics and optimal transport. *Physical Review X*, 15:31031, 7 2025. doi: 10.1103/x5vj-8jq9. URL https://link.aps.org/doi/10.1103/x5vj-8jq9.

Jérémie Klinger and Grant M Rotskoff. Universal energy-speed-accuracy trade-offs in driven nonequilibrium systems. *Physical Review E*, 111:14114, 1 2025. doi: 10.1103/PhysRevE.111.014114. URL https://link.aps.org/doi/10.1103/PhysRevE.111.014114.

L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Problems of Information Transmission*, 23:95–101, 6 1987.

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the wasserstein distance. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2022. ISBN 9781713871088.

Yaron Lipman, Ricky T Q Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Qiang Liu and Dilin Wang. Stein variational gradient descent: a general purpose bayesian inference algorithm. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2378–2386. Curran Associates Inc., 2016. ISBN 9781510838819.

Akhil Premkumar. Neural entropy, 2025. URL https://arxiv.org/abs/2409.03817.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations*, pp. 1–54, 2 2023. URL https://iclr.cc/. Publisher Copyright: © 2023 11th International Conference on Learning Representations, ICLR 2023. All rights reserved.; International Conference on Learning Representations, ICLR ; Conference date: 01-05-2023 Through 05-05-2023.

Udo Seifert. Stochastic thermodynamics, fluctuation theorems and molecular machines. *Reports on Progress in Physics*, 75:126001, 11 2012. doi: 10.1088/0034-4885/75/12/126001. URL https://dx.doi.org/10.1088/0034-4885/75/12/126001.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 2256–2265. PMLR, 2 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. doi: 10.1162/NECO_a_00142.

A ENTROPY RATES FOR SCORE-BASED DIFFUSION MODELS

Seifert's original formulation (Seifert, 2012) and subsequent applications to diffusion models (Ikeda et al., 2025) motivate the mathematical framework we adopt here.

A.1 CURRENT-SCORE-DRIFT IDENTITY

For the general overdamped SDE $d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\, dt + g(t) d\mathbf{w}_t$, the probability current is

$$\mathbf{J}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x}) - D(t) \nabla_{\mathbf{x}} p_t(\mathbf{x}) = p_t(\mathbf{x})[\mathbf{f}(\mathbf{x}, t) - D(t)\mathbf{s}(\mathbf{x}, t)],$$

where we have used $g(t)^2 = 2D(t)$, $\mathbf{s}(\mathbf{x}) = \nabla \log p_t(\mathbf{x})$ and $\nabla p_t(\mathbf{x}) = p_t(\mathbf{x}) \nabla \log p_t(\mathbf{x}) = p_t(\mathbf{x})\mathbf{s}(\mathbf{x})$. The local velocity field is defined as

$$\mathbf{v}(\mathbf{x}, t) := \mathbf{f}(\mathbf{x}, t) - D(t)\mathbf{s}(\mathbf{x}, t) = \mathbf{J}(\mathbf{x}, t)/p_t(\mathbf{x})$$

A.2 INTRINSIC ENTROPY-PRODUCTION RATE

Seifert (2012)'s original expression for the intrinsic entropy production rate is given by

$$\dot{S}^i(t) = \int \frac{\|\mathbf{J}(\mathbf{x}, t)\|^2}{D(t) p_t(\mathbf{x})} \mathrm{d}\mathbf{x} = \frac{1}{D(t)} \int p_t(\mathbf{x}) \, \|\mathbf{v}(\mathbf{x}, t)\|^2 \, d\mathbf{x}.$$

In expectation notation,

$$\begin{aligned} \dot{S}^i(t) &= \frac{1}{D(t)} \mathbb{E}\left[\|\mathbf{v}(\mathbf{x}, t)\|^2\right] = \frac{2}{g(t)^2} \mathbb{E}\left[\|\mathbf{f}(\mathbf{x}, t) - \frac{g(t)^2}{2}\mathbf{s}(\mathbf{x}, t)\|^2\right] \\ &= \frac{1}{2g(t)^2} \mathbb{E}\left[\|2\mathbf{f}(\mathbf{x}, t) - g(t)^2 \mathbf{s}(\mathbf{x}, t)\|^2\right] \end{aligned}$$

A.3 EXCHANGE (MEDIUM) ENTROPY-FLOW RATE

Seifert defines the entropy component of the medium surrounding a system (related to the heat dissipated into that medium) through the work done by the force $F(\mathbf{x}, t)$ on the system at some time-dependent temperature $T(t)$,

$$\dot{S}^m(t) = \frac{1}{T(t)} \int \mathbf{F}(\mathbf{x}, t) \cdot \mathbf{J}(\mathbf{x}, t) d\mathbf{x} = \frac{1}{D(t)} \int \mathbf{f}(\mathbf{x}, t) \cdot \mathbf{J}(\mathbf{x}, t) d\mathbf{x}$$

In order to make the analogy between the diffusion algorithm and a physical system, we imagine a mobility (inverse friction) constant μ and corresponding Einstein relation $D(t) = \mu T(t)$, allowing us to write the drift term $\mathbf{f} = \mu\mathbf{F}$ in the SDE $d\mathbf{x}_t = \mu\mathbf{F}(\mathbf{x}, t) + g(t) d\mathbf{w}_t$.

The exchange/flow rate of entropy into the system is just the negative of the one into the medium,

$$\begin{aligned} \dot{S}^e(t) = -\dot{S}^m(t) &= -\frac{1}{D(t)} \mathbb{E}\left[\mathbf{f}(\mathbf{x}, t) \cdot (\mathbf{f}(\mathbf{x}, t) - D(t)\mathbf{s}(\mathbf{x}, t))\right] \\ &= -\frac{1}{D(t)} \mathbb{E}\left[\|\mathbf{f}(\mathbf{x}, t)\|^2\right] + \mathbb{E}[\mathbf{f}(\mathbf{x}, t) \cdot \mathbf{s}(\mathbf{x}, t)] \\ &= -\frac{2}{g(t)^2} \mathbb{E}\left[\|\mathbf{f}(\mathbf{x}, t)\|^2\right] + \mathbb{E}[\mathbf{f}(\mathbf{x}, t) \cdot \mathbf{s}(\mathbf{x}, t)], \end{aligned}$$

where in the first line we have use the fact that $\mathbf{J}(\mathbf{x}, t) = \mathbf{v}(\mathbf{x}, t) p_t(\mathbf{x})$.

A.4 SYSTEM ENTROPY RATE

Combining the expressions for $\dot{S}^i$ and $\dot{S}^e$, expanding the square and canceling terms gives the simplified equation for the (total) system entropy rate:

$$\begin{aligned}\dot{S}(t) &= \dot{S}^i(t) + \dot{S}^e(t) \\ &= \frac{1}{D(t)}\mathbb{E}\left[\|\mathbf{f}(\mathbf{x},t) - D(t)\mathbf{s}(\mathbf{x},t)\|^2\right] - \frac{1}{D(t)}\mathbb{E}\left[\|\mathbf{f}(\mathbf{x},t)\|^2\right] + \mathbb{E}[\mathbf{f}(\mathbf{x},t)\cdot\mathbf{s}(\mathbf{x},t)] \\ &= -\mathbb{E}[\mathbf{f}(\mathbf{x},t)\cdot\mathbf{s}(\mathbf{x},t)] + D(t)\mathbb{E}\left[\|\mathbf{s}(\mathbf{x},t)\|^2\right] \\ &= \mathbb{E}[\nabla_{\mathbf{x}}\cdot\mathbf{f}(\mathbf{x},t)] + \frac{g(t)^2}{2}\mathbb{E}\left[\|\mathbf{s}(\mathbf{x},t)\|^2\right].\end{aligned}$$

where we have used Stein's identity for $\mathbb{E}[\nabla_{\mathbf{x}}\cdot\mathbf{f}(\mathbf{x},t)] = -\mathbb{E}[\mathbf{f}(\mathbf{x},t)\cdot\mathbf{s}(\mathbf{x},t)]$ (see Sec. B.3).

B LOWER BOUND FOR NEGATIVE LOG-LIKELIHOOD

B.1 LOG-LIKELIHOOD FROM PROBABILITY FLOW ODE

For all diffusion processes, there exists a corresponding deterministic process called the probability flow ODE whose trajectories share the same marginal probability densities $\{p_t(\mathbf{x})\}_{t=0}^T$ as the SDE Song et al. (2021). For the case of $d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t,t)dt + g(t)\mathrm{d}\mathbf{w}_t$, where $g(t) = \sigma^t$, the probability flow ODE is

$$d\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t,t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right]dt.$$

The probability flow ODE has the following form when we approximate the score with the score neural network model $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t) \approx \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$:

$$d\mathbf{x}_t = \underbrace{\left[\mathbf{f}(\mathbf{x}_t,t) - \frac{1}{2}g(t)^2\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)\right]}_{=:\tilde{\mathbf{f}}_\theta(\mathbf{x},t)}dt.$$

With the instantaneous change of variables formula (Chen et al., 2018), we can compute the log-likelihood of $p_0(\mathbf{x})$ using

$$\log p_0(\mathbf{x}_0) = \log p_T(\mathbf{x}_T) + \int_0^T \nabla\cdot\tilde{\mathbf{f}}_\theta(\mathbf{x}_t,t)\mathrm{d}t$$

where $\mathbf{x}(t)$ as a function of t can be obtained by solving the probability flow ODE. Using $T=1$ and the definition of $\tilde{\mathbf{f}}_{\boldsymbol{\theta}}$ above, the log-likelihood is

$$\log p_0(\mathbf{x}_0) = \log p_1(\mathbf{x}_1) + \int_0^1\left[\nabla\cdot\mathbf{f}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}\nabla\cdot\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)\right]dt.$$

B.2 NLL LOWER BOUND

The data-average log-likelihood at $t=0$ is

$$\mathbb{E}_{p_{\text{data}}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_0)\right] = \mathbb{E}_{p_1}\left[\log p_1(\mathbf{x}_1)\right] + \int_0^1\mathbb{E}_{p_t}\left[\nabla\cdot\mathbf{f}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}\nabla\cdot\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)\right]dt. \quad (10)$$

The first term is the entropy at $t=1$, $S_{\text{noise}} = -\mathbb{E}_{p_1}[\log p_1]$. We re-express the divergence of the score term using $\mathbb{E}_{p_t}[\nabla\cdot\mathbf{s}_{\boldsymbol{\theta}}] = -\mathbb{E}_{p_t}[\mathbf{s}_{\boldsymbol{\theta}}\cdot\mathbf{s}_{\text{true}}]$ (Stein's identity (Liu & Wang, 2016)), which gives:

$$\text{NLL} := -\mathbb{E}_{p_{\text{data}}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_0)\right] = S_{\text{noise}} + \int_0^1\left[-\mathbb{E}_{p_t}\left[\nabla\cdot\mathbf{f}(\mathbf{x}_t,t)\right] - \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[\mathbf{s}_{\boldsymbol{\theta}}\cdot\mathbf{s}_{\text{true}}\right]\right]dt$$

Using one of the polarization identities, $\mathbf{s}_{\boldsymbol{\theta}} \cdot \mathbf{s}^{\text{true}} = \frac{1}{2}\left(\|\mathbf{s}_{\boldsymbol{\theta}}\|^2 + \|\mathbf{s}_{\text{true}}\|^2 - \|\mathbf{s}_{\boldsymbol{\theta}} - \mathbf{s}_{\text{true}}\|^2\right)$, gives

$$\begin{aligned}\text{NLL} = S_{\text{noise}} - \int_0^1 \mathbb{E}_{p_t}\left[\nabla \cdot \mathbf{f}(\mathbf{x}_t, t)\right] dt - \frac{1}{2}\int_0^1 \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[||\mathbf{s}_{\boldsymbol{\theta}}||^2\right] dt \\ - \frac{1}{2}\int_0^1 \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[||\mathbf{s}_{\text{true}}||^2\right] dt + \frac{1}{2}\int_0^1 \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[||\mathbf{s}_{\boldsymbol{\theta}} - \mathbf{s}_{\text{true}}||^2\right] dt.\end{aligned}$$

We use $\int \frac{g(t)^2}{2}\mathbb{E}\,\|\mathbf{s}_{\text{true}}\|^2 = (S_{\text{noise}} - S_{\text{data}}) - \int_0^1 \mathbb{E}[\nabla \cdot \mathbf{f}(\mathbf{x}_t, t)]dt$, giving

$$\begin{aligned}\text{NLL} = \frac{S_{\text{data}} + S_{\text{noise}}}{2} - \frac{1}{2}\int_0^1 \mathbb{E}_{p_t}\left[\nabla \cdot \mathbf{f}(\mathbf{x}_t, t)\right] dt - \frac{1}{2}\int_0^1 \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[||\mathbf{s}_{\boldsymbol{\theta}}||^2\right] dt \\ + \frac{1}{2}\int_0^1 \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[||\mathbf{s}_{\boldsymbol{\theta}} - \mathbf{s}_{\text{true}}||^2\right] dt.\end{aligned}$$

Using $\dot{S}_{\boldsymbol{\theta}}(t) = \mathbb{E}[\nabla_{\mathbf{x}} \cdot \mathbf{f}(\mathbf{x}, t)] + \frac{g(t)^2}{2}\mathbb{E}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)\|^2\right]$ and the non-negativivity of the squared-difference term, we find that the negative log-likelihood obeys the lower bound

$$\boxed{\text{NLL} \geq \frac{S_{\text{data}} + S_{\text{noise}}}{2} - \frac{1}{2}\int_0^1 \dot{S}_{\boldsymbol{\theta}}(t)dt} \tag{11}$$

and the bound is tight when $s_{\boldsymbol{\theta}} = s_{\text{true}}$ and $\text{NLL} = S_{\text{data}}$.

For the drift-less diffusion process, $\mathbf{f}(\mathbf{x}_t, t) = 0$ and the system entropy rate is

$$\dot{S}_{\boldsymbol{\theta}}(t) = \dot{S}^i_{\boldsymbol{\theta}}(t) = \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)\|^2\right]$$

so the lower bound is given in terms of entropies is

$$\boxed{\text{NLL} \geq \frac{S_{\text{data}} + S_{\text{noise}}}{2} - \frac{1}{2}\int_0^1 \dot{S}^i_{\boldsymbol{\theta}}(t)dt.} \tag{12}$$

B.3 Stein's Identity

In (Liu & Wang, 2016), Stein's identity states that for sufficiently regular ϕ, we have

$$\mathbb{E}_{x\sim p}\left[\mathcal{A}_p\boldsymbol{\phi}(x)\right] = 0, \quad \text{where} \quad \mathcal{A}_p\boldsymbol{\phi}(x) = \boldsymbol{\phi}(x)\nabla_x \log p(x)^\top + \nabla_x\boldsymbol{\phi}(x), \tag{13}$$

where $\mathcal{A}_p$ is called the Stein operator, which acts on function ϕ and yields a zero mean function $\mathcal{A}_p\phi(x)$ under $x \sim p$. Expanding this identity coordinate-wise, it is exactly the statement:

$$\mathbb{E}_p[\nabla \cdot \phi] = -\mathbb{E}_p[\phi \cdot s].$$

With the true score $\mathbf{s}_{\text{true}} = \nabla \log p(\mathbf{x})$, we have:

$$\mathbb{E}_p[\nabla \cdot \mathbf{s}_{\text{true}}] = -\mathbb{E}_p\left[\|\mathbf{s}_{\text{true}}\|^2\right].$$

For an approximate score $\mathbf{s}_{\boldsymbol{\theta}}$:

$$\mathbb{E}_p\left[\nabla \cdot \mathbf{s}_{\boldsymbol{\theta}}\right] = -\mathbb{E}_p\left[\mathbf{s}_{\boldsymbol{\theta}} \cdot \mathbf{s}_{\text{true}}\right]$$

which equals $-\mathbb{E}_p\left[\|\mathbf{s}_{\boldsymbol{\theta}}\|^2\right]$ only if $\mathbf{s}_{\boldsymbol{\theta}} = \mathbf{s}_{\text{true}}$.

C Maxwell's Demon in Controlled-Forward Process

Song et al. (2021) use the notation of Haussman-Pardoux / Anderson (Haussmann & Pardoux, 1986; Anderson, 1982)

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t \quad \text{(Forward)} \tag{14}$$

$$\mathrm{d}\mathbf{x}_t = [\mathbf{f}(\mathbf{x}, t)\mathrm{d}t - g(t)^2\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}_t \quad \text{(Reverse)} \tag{15}$$

where $\overline{\mathbf{w}}_t$ is the standard Wiener process when time is run backwards. **Note**, Eq. (8) is usually integrated from T down to 0, making dt negative.

C.1 Reverse Process

We have chosen the forward SDE to be

$$d\mathbf{x}_t = \sigma^t d\mathbf{w}_t, \quad t \in [0, 1].$$

To sample from our time-dependent score-based model $s_{\boldsymbol{\theta}}(\mathbf{x}, t)$, we first draw a sample from the prior distribution $p_1 \approx \mathbf{N}\left(\mathbf{x}; \mathbf{0}, \frac{1}{2}\left(\sigma^2 - 1\right)\mathbf{I}\right)$, and then solve the reverse-time SDE with numerical methods. In particular, using our time-dependent score-based model, the reverse-time SDE can be approximated by

$$d\mathbf{x}_t = -\sigma^{2t} s_{\boldsymbol{\theta}}(\mathbf{x}, t)dt + \sigma^t d\overline{\mathbf{w}}_t$$

Next, one can use numerical methods to solve for the reverse-time SDE, such as the Euler-Maruyama approach. It is based on a simple discretization to the SDE, replacing dt with $\Delta t > 0$ and $d\mathbf{w}$ with $\mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, g^2(t)\Delta t\mathbf{I}\right)$. When applied to our reverse-time SDE, we can obtain the following iteration rule

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t + \sigma^{2t} s_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\Delta t + \sigma^t\sqrt{\Delta t}\mathbf{z}_t$$

where $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

C.2 Controlled-Forward Process

Time always runs forward in the real world: one can achieve a physical realization of the generative process by defining a clock

$$\tau := T - t, \quad 0 \le \tau \le T,$$

such that integrating forward in τ is the same as integrating backward in t. We plug in $t = T - \tau$, $dt = -d\tau$, and $d\bar{\mathbf{w}}_t = d\mathbf{w}_\tau$ to re-parameterize reverse Eq. (8) as a controlled-forward process

$$\mathrm{d}\bar{\mathbf{x}}_\tau = [-\mathbf{f}(\bar{\mathbf{x}}_\tau, T - \tau) + g(T - \tau)^2\mathbf{s}_{\boldsymbol{\theta}}(\bar{\mathbf{x}}_\tau, T - \tau)]\mathrm{d}\tau + g(T - \tau)\mathrm{d}\mathbf{w}_\tau \tag{16}$$

where $\bar{\mathbf{x}}_\tau := \mathbf{x}_t$.

C.3 Entropy Rates of the Controlled-Forward Process

For the controlled-forward formulation, the drift becomes $\tilde{\mathbf{f}}(\bar{\mathbf{x}}_\tau, \tau) = g(\tau)^2\mathbf{s}_{\boldsymbol{\theta}}(\bar{\mathbf{x}}_\tau, \tau)$. Substituting this into the general entropy rate expressions derived in Appendix A yields the simplified relations:

$$\dot{S}^i_{\boldsymbol{\theta}}(\tau) = \frac{g(\tau)^2}{2}\,\mathbb{E}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\bar{\mathbf{x}}_\tau, \tau)\|^2\right], \quad \dot{S}^e_{\boldsymbol{\theta}}(\tau) = -2\dot{S}^i_{\boldsymbol{\theta}}(\tau), \quad \dot{S}_{\boldsymbol{\theta}}(\tau) = -\dot{S}^i_{\boldsymbol{\theta}}(\tau).$$

Thus, in the controlled-forward process the system entropy rate is exactly the negative of the intrinsic entropy production rate.

D General Continuous-Time Diffusion Processes

Song et al. (2021) showed that score-based generative models can be formulated in terms of a general Itô SDE of the form

$$d\mathbf{x}_t = \mathbf{f}\left(\mathbf{x}_t, t\right)dt + g(t)d\mathbf{w}_t$$

where $\mathbf{f}\left(\mathbf{x}_t, t\right)$ is the drift, $g(t)$ the diffusion coefficient, and $\mathbf{w}_t$ a standard Wiener process. Two canonical instantiations of this framework correspond to the variance exploding (VE) and variance preserving (VP) processes.

D.1 Variance Exploding (VE) SDE

The VE process is defined by

$$d\mathbf{x}_t = \sqrt{\frac{d}{dt}\sigma^2(t)}d\mathbf{w}_t$$

with $\sigma^2(t)$ a non-decreasing variance schedule. Here the drift vanishes, $\mathbf{f}\left(\mathbf{x}_t, t\right) = 0$, while the diffusion coefficient is chosen so that the marginal variance of $\mathbf{x}_t$ increases monotonically in t. As $t \to T$, the variance diverges (hence "exploding"), and the distribution approaches a Gaussian prior. This setting is natural when starting from bounded data distributions, since the forward process progressively washes out structure by injecting unbounded noise

D.2 VARIANCE PRESERVING (VP) SDE

In contrast, the VP process includes both drift and diffusion terms:

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t dt + \sqrt{\beta(t)}d\mathbf{w}_t$$

where $\beta(t)$ is a positive noise-rate schedule. The drift pulls $\mathbf{x}_t$ toward the origin at a rate proportional to $\beta(t)$, while the diffusion injects noise of matching strength. This balance ensures that the overall variance of the process remains bounded (and can be normalized to unity) for all t. Thus, the forward diffusion maps data smoothly into an isotropic Gaussian prior without variance blow-up.

Both SDEs fit seamlessly into the score-based generative modeling framework. In each case, the reverse-time dynamics introduce an additional score-dependent drift term,

$$d\mathbf{x}_\tau = \left[-\mathbf{f}\left(\mathbf{x}_\tau, T-\tau\right) + g^2(T-\tau)\nabla_{\mathbf{x}_\tau}\log p_\tau\left(\mathbf{x}_\tau\right)\right]d\tau + g(T-\tau)d\mathbf{w}_\tau$$

where the score $\nabla_{\mathbf{x}_\tau}\log p_\tau\left(\mathbf{x}_\tau\right)$ is approximated by a neural network. The VE and VP choices thus represent two distinct, yet complementary, continuous-time noise injection schemes, both of which reduce to the driftless case when $f = 0$ and variance is allowed to grow freely. They provide the practical foundation for most modern diffusion models, differing primarily in how variance is managed over time and, correspondingly, in their tradeoffs between sample quality and likelihood.

In addition to the variance exploding (VE) process considered in the main text, we evaluate the variance preserving (VP) process used in denoising diffusion probabilistic models (DDPMs). The VP process is governed by the forward SDE

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\,\mathbf{x}_t\,dt + \sqrt{\beta(t)}\,d\mathbf{w}_t,$$

where $\beta(t)$ is the variance schedule and w_t is standard Brownian motion. This process interpolates between the data distribution at $t = 0$ and an isotropic Gaussian prior at $t = 1$ while preserving variance at each time step. The reverse process is parameterized by the learned score network, and the associated entropy-production integrals are estimated analogously to the VE case.

D.3 CONSTRUCTING THE VP SCHEDULE FROM σ

To specify $\beta(t)$, we set a desired terminal noise scale σ, which encodes how much Gaussian noise is injected by the end of the forward process.

D.3.1 INTEGRATED NOISE BUDGET

The mean-scaling factor of the VP process is

$$\alpha(t) = \exp\left(-\frac{1}{2}\int_0^t \beta(u)du\right)$$

so at terminal time $t = 1$,

$$\alpha(1)^2 = \exp(-B), \quad B := \int_0^1 \beta(u)du$$

The variance contributed by the noise term is

$$\sigma(1)^2 = 1 - \alpha(1)^2.$$

Requiring $\sigma(1)^2 = \sigma^2/1 + \sigma^2$ yields the condition

$$B = \log\left(1 + \sigma^2\right)$$

Thus the entire VP schedule is determined by the integrated noise budget B.

D.3.2 LINEAR SCHEDULE CONSTRUCTION

A common choice is to make $\beta(t)$ linear in t:

$$\beta(t) = \beta_{\min} + t\,(\beta_{\max} - \beta_{\min})$$

The constants β_{min} and β_{max} are set so that the integral matches the budget:

$$\int_0^1 \beta(t)dt = \frac{1}{2}\,(\beta_{\min} + \beta_{\max}) = B$$

Introducing a ratio parameter $r \in (0, 1)$, we define

$$\beta_{\min} = rB, \quad \beta_{\max} = (2 - r)B,$$

which ensures the correct average while allowing flexibility in the temporal profile of noise injection. Smaller r front-loads noise near $t = 1$, while larger r distributes noise more evenly across time.

In the VP formulation, B is the logarithmic noise budget: it quantifies the total exponential damping of the signal. In the VE formulation, the corresponding budget is the variance scale σ^2. The two are linked by $\sigma^2 = e^B - 1$. Hence, the VP schedule can be constructed from a single intuitive parameter σ, which specifies the effective strength of the forward noise process, while B serves as its natural exponential coordinate.

E STANDARD GAUSSIAN DATA

For validation we include experiments where the data distribution p_{data} is a standard Gaussian. In this case, the score function is exactly linear:

$$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) = -\mathbf{x},$$

which can be fit by a single-layer neural network with linear weights. This setting provides a ground-truth baseline where the score is known analytically, allowing us to verify the tightness of the lower bound and the accuracy of our numerical estimators.

Consider the case in which the data is normally distributed $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and a drift-less forward process with variance increment $v(t) = \int_0^t g(u)^2 du$, with $g^2(t) = \sigma^{2t}$ and $v(t) = \frac{\sigma^{2t}-1}{2\ln\sigma}$. Then $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma} + v(t)\boldsymbol{I})$ and

$$\mathbf{s}_{\text{true}}\,(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = -(\boldsymbol{\Sigma} + v(t)\mathbf{I})^{-1}(\mathbf{x} - \boldsymbol{\mu}),$$

which is exactly linear in $\mathbf{x}$ for every t. A tiny network (even a single linear layer conditioned on t) can represent this perfectly, so training can drive $\mathbf{s}_{\boldsymbol{\theta}} \to \mathbf{s}_{\text{true}}$.

Proof. Let

$$p_t(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C}(t)), \quad \mathbf{C}(t) = \boldsymbol{\Sigma} + v(t)\mathbf{I}_d,$$

so $\mathbf{C}(t)$ is symmetric positive-definite. The multivariate Gaussian pdf is

$$p_t(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\det(\mathbf{C})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

Take logs:

$$\log p_t(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}) - \frac{1}{2}\log\det(2\pi\mathbf{C}).$$

Only the quadratic term depends on $\mathbf{x}$. With $h = (x - \mu)_j A_{jk}(x - \mu)_k$,

$$\frac{\partial h}{\partial x_i} = A_{ij}(x - \mu)_j + A_{ji}(x - \mu)_j = \left[\left(A + A^\top\right)(x - \mu)\right]_i$$

and we have

$$\nabla_{\mathbf{x}}\left[(\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{A}(\mathbf{x} - \boldsymbol{\mu})\right] = \left(\mathbf{A} + \mathbf{A}^\top\right)(\mathbf{x} - \boldsymbol{\mu}) = 2\mathbf{A}(\mathbf{x} - \boldsymbol{\mu}) \quad \left(\mathbf{A} = \mathbf{A}^\top\right),$$

with $\mathbf{A} = \mathbf{C}^{-1}$, we get

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = -\frac{1}{2} \cdot 2\mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = -\mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}).$$

Therefore the true score is

$$\mathbf{s}_{\text{true}}\,(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = -(\boldsymbol{\Sigma} + v(t)\mathbf{I})^{-1}(\mathbf{x} - \boldsymbol{\mu})$$

exactly as claimed.

F Exact Score of the Uniform and Normal Distributions

Pixels are independent under Uniform[0,1] and the Gaussian noise factorizes across coordinates, so we consider the 1D case and adopt scalar notation throughout. For the drift-less diffusion,

$$dx_t = g(t)dW_t, \quad v(t) = \int_0^t g(u)^2 du, \quad s = \sqrt{v(t)}$$

Conditioned on x_0, we have

$$x_t \mid x_0 \sim \mathcal{N}\left(x_0, s^2\right)$$

If the data is Uniform on $[0, 1]$ (density $p_0(u) = \mathbf{1}_{[0,1]}(u)$), the marginal at time t is the convolution

$$p_t(x) = \int_0^1 \phi_s(x - u)du$$

where

$$\phi_s(z) = \frac{1}{\sqrt{2\pi}s} \exp\left(-\frac{z^2}{2s^2}\right)$$

is the $\mathcal{N}\left(0, s^2\right)$ pdf. With a change of variable $z = (x - u)/s$ and $du = -sdz$, we have

$$p_t(x) = \int_{(x-1)/s}^{x/s} \phi(z)dz = \Phi\left(\frac{x}{s}\right) - \Phi\left(\frac{x-1}{s}\right)$$

with ϕ and Φ the standard normal pdf/cdf. Differentiate w.r.t. x :

$$\partial_x p_t(x) = \frac{1}{s}\left[\phi\left(\frac{x}{s}\right) - \phi\left(\frac{x-1}{s}\right)\right].$$

The score is the gradient of the log-density,

$$s(x, t) = \partial_x \log p_t(x) = \frac{\partial_x p_t(x)}{p_t(x)} = \frac{\frac{1}{s}\left[\phi\left(\frac{x}{s}\right) - \phi\left(\frac{x-1}{s}\right)\right]}{\Phi\left(\frac{x}{s}\right) - \Phi\left(\frac{x-1}{s}\right)}.$$

G Swiss Roll Experimental Details

We adapt our implementation from the open-source codebase `sdeflow-light` (`https://github.com/CW-Huang/sdeflow-light`).

G.1 Swiss roll dataset specifications

We generate a 2D Swiss roll dataset using `sklearn.datasets.make_swiss_roll`. For each draw, we sample n points from the standard 3D Swiss roll with additive isotropic Gaussian noise (default `noise`$= 0.5$), then retain the (x, z) coordinates and rescale by a factor of $1/5$, yielding $\mathbf{x}_0 \in \mathbb{R}^2$. Concretely, if the raw sample is $\tilde{\mathbf{x}}_0 \in \mathbb{R}^3$, we form

$$\mathbf{x}_0 \;=\; \frac{1}{5}\begin{bmatrix}\tilde{x}_0 \\ \tilde{z}_0\end{bmatrix} \in \mathbb{R}^2. \tag{17}$$

During training, minibatches are generated on-the-fly by repeatedly sampling fresh i.i.d. points from this procedure (i.e., we do not materialize a fixed finite training set). For visualization, we plot 2D histograms over a fixed window approximately $[-5, 5]^2$.

G.2 VP schedule specifications

We use the variance-preserving (VP) forward diffusion in the continuous-time form

$$d\mathbf{x}_t \;=\; -\frac{1}{2}\beta(t)\,\mathbf{x}_t\,dt \;+\; \sqrt{\beta(t)}\,d\mathbf{w}_t, \qquad t \in [0, T], \tag{18}$$

where $\mathbf{w}_t$ is standard Brownian motion in $\mathbb{R}^2$. We take a linear noise schedule

$$\beta(t) \;=\; \beta_{\min} + t\big(\beta_{\max} - \beta_{\min}\big), \tag{19}$$

and set $T = 1$ in all experiments.

Closed-form marginal sampling. The conditional distribution $\mathbf{x}_t \mid \mathbf{x}_0$ is Gaussian:

$$\mathbf{x}_t \mid \mathbf{x}_0 \sim \mathcal{N}\big(m(t)\mathbf{x}_0,\ \mathrm{var}(t)\,\mathbf{I}\big), \tag{20}$$

with

$$m(t) \ = \ \exp\Big(-\tfrac{1}{4}t^2(\beta_{\max}-\beta_{\min})-\tfrac{1}{2}t\beta_{\min}\Big), \tag{21}$$

$$\mathrm{var}(t) \ = \ 1-\exp\Big(-\tfrac{1}{2}t^2(\beta_{\max}-\beta_{\min})-t\beta_{\min}\Big), \tag{22}$$

and diffusion amplitude

$$g(t) \ = \ \sqrt{\beta(t)}. \tag{23}$$

In the reference implementation, we use $(\beta_{\min},\beta_{\max})=(0.1,20.0)$.

G.3 Model specifications

Time-conditioned MLP. We parameterize the learned score/drift network as a time-conditioned multilayer perceptron (MLP). The input is formed by concatenating time $t\in\mathbb{R}$ to the state $\mathbf{x}\in\mathbb{R}^2$:

$$[\mathbf{x};t]\in\mathbb{R}^3, \tag{24}$$

and the network outputs a 2D vector:

$$\mathrm{MLP}_\theta:\mathbb{R}^3\to\mathbb{R}^2, \qquad (\mathbf{x},t)\mapsto \mathrm{MLP}_\theta(\mathbf{x},t). \tag{25}$$

We use Swish activations and three hidden layers.

We sweep model capacity by varying the hidden width $H\in\{4,8,16,32,64,128,256\}$ (three hidden layers fixed), training separate models from scratch for each H and we run 5 independent replicates per configuration with different random seeds.

G.4 Training specifications

Denoising score matching (DSM) objective. At each iteration, we sample $t\sim\mathrm{Unif}[0,T]$, draw $\mathbf{x}_t\mid\mathbf{x}_0$ using the closed form in equation 20, and obtain the associated injected noise $\boldsymbol{\epsilon}$ such that

$$\mathbf{x}_t \ = \ m(t)\mathbf{x}_0+\mathrm{std}(t)\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}), \quad \mathrm{std}(t)=\sqrt{\mathrm{var}(t)}. \tag{26}$$

We minimize the DSM loss (as implemented) given by

$$\mathcal{L}_{\mathrm{DSM}}(\theta) \ = \ \frac{1}{2}\,\mathbb{E}\left[\left\|\mathrm{MLP}_\theta(\mathbf{x}_t,t)\cdot\frac{\mathrm{std}(t)}{g(t)}+\boldsymbol{\epsilon}\right\|_2^2\right], \tag{27}$$

where the expectation is approximated by a minibatch average.

Optimization and runtime settings. We train using Adam with learning rate 10^{-3}, batch size 256, for 100,000 iterations. All experiments are run with GPU acceleration on an NVIDIA A100 GPU.

H Numerical Estimates of Exact NLL Terms

H.1 Equilibrium entropy S_{NOISE}

At $t=1$, the forward drift-less diffusion process has covariance $v(1)\mathbf{I}$ with

$$v(1)=\frac{\sigma^2-1}{2\ln\sigma}.$$

Hence the equilibrium entropy in nats for a d-dimensional Gaussian is

$$S_{1,\mathrm{nats}}=\frac{d}{2}\ln(2\pi e v(1))=\frac{d}{2}\ln\left(2\pi e\frac{\sigma^2-1}{2\ln\sigma}\right)$$

and in bits-per-dimension (bpd),

$$S_1=\frac{S_{1,\mathrm{nats}}}{d\ln 2}$$

This term is computed analytically. We compute the exact closed form and convert it to bpd.

H.2 DATASET ENTROPY S_{DATA}

H.2.1 UNIFORM AND GAUSSIAN DATASETS

These constants are independent of the diffusion schedule (VE vs VP) and enter directly into the NLL lower bound formulas. For the standard Gaussian datasets considered, the data entropy is fixed by the closed-form expression

$$S_{\text{data}} = \frac{1}{2} d \log(2\pi e)$$

corresponding to the entropy of a d-dimensional standard normal.

For the Uniform $[0,1]^d$ datasets, the entropy vanishes, $S_{\text{data}} = 0$, since the density is constant on its support. Let $X \sim \text{Unif}([0,1]^d)$, so $p(x) = 1$ for $x \in [0,1]^d$ and $p(x) = 0$ otherwise. The differential entropy is

$$S_{\text{data}} \;=\; -\int_{\mathbb{R}^d} p(x)\, \log p(x)\, dx \;=\; -\int_{[0,1]^d} 1 \cdot \log 1 \; dx \;=\; 0.$$

By factorization across coordinates, $S_{\text{data}} = \sum_{i=1}^{d} h(X_i)$ with $X_i \sim \text{Unif}([0,1])$ and $h(X_i) = -\int_0^1 1 \cdot \log 1 \, dx_i = 0$, hence $S_{\text{data}} = 0$.

H.2.2 ESTIMATING THE ENTROPY OF THE SWISS ROLL DATASET

This section describes three complementary ways we obtain the differential entropy of the 2D Swiss roll distribution.

(1) Model-based upper bound on S_{data}. Rearranging the lower bound on negative log-likelihood gives the upper bound

$$S_{\text{data}} \;\le\; 2\,\text{NLL}(\theta) + \int_0^1 \dot{S}_{\boldsymbol{\theta}}(t)\, dt - S_{\text{noise}}.$$

We sweep model capacity by varying the hidden width $H \in \{4, 8, \ldots, 256\}$ (three hidden layers fixed), training separate models from scratch for each H and running 5 independent replicates per configuration with different random seeds. We report the smallest upper bound observed across all configurations, attained by the model with $H = 256$.

(2) Differential entropy via the Kozachenko–Leonenko (KL) k-NN estimator. As an independent estimator of S_{data}, we use the Kozachenko–Leonenko k-nearest-neighbors estimator Kozachenko & Leonenko (1987). Given i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^d$, let ε_i be the Euclidean distance from $\mathbf{x}_i$ to its k-th nearest neighbor. The estimator of the differential entropy (in nats) is

$$\widehat{S}_{\text{data}}^{\text{kNN}}(k) = \psi(N) - \psi(k) + \log V_d + d\,\frac{1}{N}\sum_{i=1}^{N} \log(\varepsilon_i),$$

where $\psi(\cdot)$ is the digamma function and $V_d = \pi^{d/2}/\Gamma(d/2+1)$ is the volume of the unit ball in $\mathbb{R}^d$. We compute ε_i efficiently using a KD-tree query and report $\widehat{S}_{\text{data}}^{\text{kNN}}(k)$ in bits per dimension.

(3) Differential entropy via a 2D histogram discretization. We also estimate S_{data} using a discretization approach. Fix a rectangular range $[x_{\min}, x_{\max}] \times [z_{\min}, z_{\max}]$ and a uniform $B \times B$ grid (bins). Let p_{ij} be the empirical mass in bin (i,j). The discrete entropy of the quantized variable X^Δ is

$$H(X^\Delta) = -\sum_{i,j} p_{ij} \log p_{ij}.$$

To convert this to a differential-entropy estimate, we add the log cell area:

$$\widehat{S}_{\text{data}}^{\text{hist}}(B) = H(X^\Delta) + \log(\Delta x\, \Delta z), \qquad \Delta x = \frac{x_{\max} - x_{\min}}{B}, \quad \Delta z = \frac{z_{\max} - z_{\min}}{B}.$$

This is the standard Riemann-sum correction for moving from probabilities (masses) to densities Cover & Thomas (2005).

Choosing the histogram range and resolution is crucial. We set the range using a pilot sample with a small padding fraction to reduce boundary artifacts, then sweep over candidate B and find the B whose $\widehat{S}^{\mathrm{hist}}_{\mathrm{data}}(B)$ is consistent with $\widehat{S}^{\mathrm{kNN}}_{\mathrm{data}}(k)$ (e.g. $k = 10$).

H.3 Squared-norm of the model score

We estimate

$$I_{\boldsymbol{\theta}} = \frac{1}{2}\int_0^1 g(t)^2 \mathbb{E}_{\mathbf{x}_t \sim p_t}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\right\|^2\right] dt$$

Estimator (per time grid t_k, batch size B):

1. Draw $\mathbf{x}_0 \sim p_{\mathrm{data}}, \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$.
2. Form $\mathbf{x}_t = \mathbf{x}_0 + \sqrt{v\left(t_k\right)}\mathbf{z}$.
3. Evaluate the model score $\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t_k\right)$.
4. Compute the batch mean $\widehat{E}_k = \frac{1}{B}\sum_{i=1}^{B}\left\|\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t^{(i)}, t_k\right)\right\|^2$.

Finally, we integrate over t with the trapezoid rule:

$$\hat{I}_{\boldsymbol{\theta}} = \frac{1}{2}\sum_k w_k \widehat{E}_k, \quad w_k = g\left(t_k\right)^2 \Delta t_k$$

and convert to bpd by dividing by $d \ln 2$.

H.4 Squared-difference term, I_{diff}

$$I_{\mathrm{diff}} = \frac{1}{2}\int_0^1 g(t)^2 \mathbb{E}_{\mathbf{x}_t \sim p_t}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) - \mathbf{s}_{\mathrm{true}}\left(\mathbf{x}_t, t\right)\right\|^2\right] dt \geq 0$$

We estimate it in two ways: directly by computing $\left\|\mathbf{s}_{\boldsymbol{\theta}} - \mathbf{s}_{\mathrm{true}}\right\|^2$ per sample and average, and using the polarization identity as a sanity check:

$$\|a - b\|^2 = \|a\|^2 + \|b\|^2 - 2\langle a, b\rangle$$

to obtain the the squared difference term from separate estimates of $\left\|\mathbf{s}_{\boldsymbol{\theta}}\right\|^2$, $\left\|\mathbf{s}_{\mathrm{true}}\right\|^2$, and $\langle \mathbf{s}_{\boldsymbol{\theta}}, \mathbf{s}_{\mathrm{true}}\rangle$. We use the agreement between the two as a useful consistency diagnostic.