INFO-SEDD: CONTINUOUS TIME MARKOV CHAINS AS SCALABLE INFORMATION METRICS ESTIMATORS

Alberto Foresti, Giulio Franzese, Pietro Michiardi
EURECOM
{alberto.foresti,giulio.franzese,pietro.michiardi}@eurecom.fr

ABSTRACT

Information-theoretic quantities play a crucial role in understanding non-linear relationships between random variables and are widely used across scientific disciplines. However, estimating these quantities remains an open problem, particularly in the case of high-dimensional discrete distributions. Current approaches typically rely on embedding discrete data into a continuous space and applying neural estimators originally designed for continuous distributions, a process that may not fully capture the discrete nature of the underlying data.

We consider Continuous-Time Markov Chains (CTMCs), stochastic processes on discrete state-spaces which have gained popularity due to their generative modeling applications. In this work, we introduce INFO-SEDD, a novel method for estimating information-theoretic quantities of discrete data, including mutual information and entropy. Our approach requires the training of a single parametric model, offering significant computational and memory advantages. Additionally, it seamlessly integrates with pretrained networks, allowing for efficient reuse of pretrained generative models.

To evaluate our approach, we construct a challenging synthetic benchmark. Our experiments demonstrate that INFO-SEDD is robust and outperforms neural competitors that rely on embedding techniques. Moreover, we validate our method on a real-world task: estimating the entropy of an Ising model. Overall, INFO-SEDD outperforms competing methods and shows scalability to high-dimensional scenarios, paving the way for new applications where estimating MI between discrete distribution is the focus.

1 INTRODUCTION

Information theoretic quantities represent a powerful tool to understand non-linear relationships between random variables (Shannon, 1948; MacKay, 2003) and find wide range of applications in scientific fields (Karbowski, 2024; Eckford et al., 2016). Mutual information, in particular, has become an established metric in machine learning (Bell & Sejnowski, 1995; Stratos, 2019; Belghazi et al., 2018; Oord et al., 2018; Hjelm et al., 2019), both for training models (Alemi et al., 2016; Chen et al., 2016; Zhao et al., 2018) and at inference time (Alemi & Fischer, 2018; Huang et al., 2020).

Estimating information theoretic quantities remains an open problem, and different paradigms for their estimation emerged. Classical parametric and non-parametric methods (Pizer et al., 1987; Moon et al., 1995; Kraskov et al., 2004; Gao et al., 2015) have been recently superseded by variational approaches (Barber & Agakov, 2004; Nguyen et al., 2007; Nowozin et al., 2016; Poole et al., 2019; Wunder et al., 2021; Letizia et al., 2023; Federici et al., 2023) and neural estimators (Papamakarios et al., 2017; Belghazi et al., 2018; Oord et al., 2018; Song & Ermon, 2019b; Rhodes et al., 2020; Letizia & Tonello, 2022; Brekelmans et al., 2022; Franzese et al., 2023; Butakov et al., 2024). Despite its practical importance, few estimators for high-dimensional **discrete** distributions have been proposed in the literature. While classical estimators for discrete random variables exist (Pinchas et al., 2024), their accuracy rapidly decrease with increasing dimensionality of the considered problem. Applications that would benefit from scalable estimators of mutual information quantities include, among others, DNA or peptide sequencing (Newcomb & Sayood, 2021; Xia et al.), text summarization (Darrin et al., 2024) and neuroscience (Chai et al., 2009). Conse-

quently, the development of new estimation techniques is of paramount importance for the broader scientific community.

The current approach to solve the problem of lack of a viable estimator for discrete, high dimensional scenarios, is to embed in a real space the discrete quantities and then adopt neural estimators conceived for continuous distributions. One recent example is (Lee & Rhee, 2024), where it is showed that the embeddings of pretrained language models can provide meaningful representations to estimate information theoretic quantities in unstructured data. However such a process may not fully capture the discrete nature of the underlying data and might suffer from several limitations, such as the necessity to consider embeddings which are application specific.

One extremely promising class of estimators which has been recently been considered in the continuous state space settings has its roots in generative diffusion models Song et al. (2021); Ho et al. (2020). While these generative models have been successfully considered in continuous settings for estimating information metrics (Franzese et al., 2023; Kong et al., 2022; Bounoua et al., 2024), a discrete state space adaptation is currently not available. In this work we fill this important gap and present INFO-SEDD, a novel method for estimating information theoretic quantities of discrete data using Continuous Time Markov Chains (CTMCs) (Lou et al., 2024). These stochastic processes have recently saw a surge in popularity due to the associated generative modelling applications as direct counterpart of the continuous state space models Song et al. (2021); Ho et al. (2020). Their fundamental working principle is the reversal of a perturbation process which starts from clean data from a given distribution and converge to uninformative noise. The workhorse of these approaches is the *score function*, which contains information about the probability distributions associated to the CTMCs at different time instants. Our proposed method, INFO-SEDD, builds upon the same fundamental mathematical framework, extending it via Dynkin's lemma (Hanson, 2007), and leverages score functions to compute key information-theoretic metrics, such as mutual information between two random variables and the entropy of a given distribution. By carefully selecting perturbation processes, our approach requires training only a single parametric model to compute mutual information across arbitrary subsets of variables. Furthermore, INFO-SEDD seamlessly integrates with pretrained networks, enabling the reuse of computational resources already expended in training generative models.

To rigorously evaluate our method, we design a benchmark that presents challenges across three critical dimensions: (1) the support size of the random variables, which defines the range of possible values each variable can take; (2) the dimensionality of the variable representations, referring to the number of components each variable contains; and (3) the mutual information value, which captures the complexity of dependencies between variables. Our results demonstrate that INFO-SEDD is both robust and consistently outperforms neural estimation methods that rely on embedding techniques. Beyond synthetic benchmarks, we further assess the practical utility of our approach by applying it to the real-world problem of estimating the entropy of the Ising model (Onsager, 1944). The Ising model is a paradigmatic example of a complex system with broad applications in statistical physics, neuroscience, and machine learning. Crucially, it provides a well-characterized ground truth for entropy, making it an ideal test case for evaluating the accuracy of information-theoretic estimators in high-dimensional discrete distributions.Our method achieves precise entropy estimates in this challenging setting, reinforcing INFO-SEDD as a promising and reliable estimator for complex discrete data distributions.

2 METHODOLOGY

In this Section, we explore the relationship between CTMCs over discrete state spaces (Anderson, 2012) and the computation of Kullback-Leibler (KL) divergences. First, in Section 2.1, we provide a brief introduction to the fundamentals of CTMCs, emphasizing their time-reversal properties and parametric approximations (Lou et al., 2024). Then, in Section 2.2, we demonstrate how these processes can be adapted for divergence estimation, specifically by analyzing two processes that share the same generator but differ in their initial conditions.

2.1 PRELIMINARIES

Consider a CTMC $\overrightarrow{X}_t, t \in [0,T]$, defined over a finite state space $\chi = \{1,\dots,N\}$ and specified by the infinitesimal generators $\overrightarrow{Q}_t : [0,T] \to \mathbb{R}^{N\times N}$, where the diagonal entries statisfy $\overrightarrow{Q}_t(x,x) = -\sum_{x\neq y}\overrightarrow{Q}_t(x,y)$, with $\overrightarrow{Q}_t(x,y) \geq 0, x \neq y$. As established in (Anderson, 2012), the time evolution of the probability distribution $Pr(X_t = i) \overset{\text{def}}{=} \overrightarrow{p}_t(i)$, satisfies the following Ordinary Differential Equation (ODE)

$$\overrightarrow{p}_t = \overrightarrow{p}_0 + \int_0^t \overrightarrow{p}_s \overrightarrow{Q}_s ds, \tag{1}$$

where the initial conditions of the process $\overrightarrow{p}_0$ determine the distribution $\overrightarrow{p}_t$ at any time t.

A key property of CTMCs is that their time-reversed counterpart, recently utilized in generative modeling (Lou et al., 2024), also follows a CTMC but with a different set of transition matrices. More precisely, defining the time-reversed process as $\overleftarrow{p}_t \overset{\text{def}}{=} \overrightarrow{p}_{T-t}$, it evolves according to (Lou et al., 2024; Sun et al., 2023):

$$\overleftarrow{p}_t = \overrightarrow{p}_T + \int_0^t \overleftarrow{Q}_s \overleftarrow{p}_s ds, \tag{2}$$

where the reverse-time transition matrices $\overleftarrow{Q}_t$ relate to the forward transition matrices as follows:

$$\overleftarrow{Q}_t(y,x) = \left(\frac{\overrightarrow{p}_{T-t}(y)}{\overrightarrow{p}_{T-t}(x)}\overrightarrow{Q}_{T-t}(x,y)\right)(1-\delta(x,y)) + \left(-\sum_{y\neq x}\overleftarrow{Q}_t(y,x)\right)\delta(x,x) \tag{3}$$

Under appropriate technical conditions on $\overrightarrow{Q}_t$ (Lou et al., 2024) the terminal distribution $\overrightarrow{p}_T$ converges to a known reference distribution π, which is independent of the initial distribution $\overrightarrow{p}_0$. This property enables sampling from $\overrightarrow{p}_0$ by simulating a CTMC with appropriately chosen generators (Sun et al., 2023; Kelly, 1981). However, other than simple and uninteresting scenarios, exact knowledge of the quantities $\frac{\overrightarrow{p}_{T-t}(y)}{\overrightarrow{p}_{T-t}(x)}$ is out of reach. A practical solution is to substitute in this numerical integration a parametric function $s^p_\theta(x,t)_y$, whose parameters are optimized according to Lou et al. (2024):

$$\mathcal{L}(\theta) = \mathbb{E}\left[\int_0^T \sum_{y\neq \overrightarrow{X}_t} \overrightarrow{Q}_t(\overrightarrow{X}_t, y)\left(s^p_\theta(\overrightarrow{X}_t,t)_y - \frac{\overrightarrow{p}_t(y|\overrightarrow{X}_0)}{\overrightarrow{p}_t(\overrightarrow{X}_t|\overrightarrow{X}_0)}\log s^p_\theta(\overrightarrow{X}_t,t)_y + K\left(\frac{\overrightarrow{p}_t(y|\overrightarrow{X}_0)}{\overrightarrow{p}_t(\overrightarrow{X}_t|\overrightarrow{X}_0)}\right)\right)dt\right] \tag{4}$$

Where $\overrightarrow{p}_t(\cdot|x_0)$ is a known perturbation kernel, obtained from Equation (1) with the deterministic initial distribution centered in x_0, and $K(a) = a(\log a - 1)$. Whenever the context is clear, we simplify the notation for the parametric score in the remainder of the paper and denote $s^p_\theta(\overrightarrow{X}_t)_y$ instead of $s^p_\theta(\overrightarrow{X}_t,t)_y$.

2.2 KL DIVERGENCES VIA CTMCS

In this work, we leverage the CTMC framework to compute the KL divergence between two probability distributions $\overrightarrow{p}_0$ and $\overrightarrow{q}_0$ defined over the same support χ, expressed as $\text{KL}\left[\overrightarrow{p}_0 \parallel \overrightarrow{q}_0\right]$. To achieve this, we construct two Markov chains that differ only in their initial conditions: one initialized from $\overrightarrow{p}_0$ whose time evolution follows Equation (2), and the other initialized from $\overrightarrow{q}_0$, which

evolves analogously by substituting $\overrightarrow{q}_t$ for $\overrightarrow{p}_t$ in Equation (2). Since the KL divergence satisfies $\text{KL}\left[\overrightarrow{p}_0 \parallel \overrightarrow{q}_0\right] = \mathbb{E}\left[\log \frac{\overrightarrow{p}_0}{\overrightarrow{q}_0}(\overrightarrow{X}_0)\right]$, we can equivalently express it as

$$\mathbb{E}\left[\log \frac{\overrightarrow{p}_0}{\overrightarrow{q}_0}(\overleftarrow{X}_T)\right] = \mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T}(\overleftarrow{X}_T)\right] = \mathbb{E}\left[\mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T}(\overleftarrow{X}_T)\middle|\overleftarrow{X}_0\right]\right]. \tag{5}$$

The last term in equation 5 can be rewritten using Dynkin's formula Hanson (2007), which states that for a generic function $f : \chi \times [0, T] \to \mathbb{R}$, we have:

$$\mathbb{E}\left[f(\overleftarrow{X}_T, T)\middle|\overleftarrow{X}_0\right] - f(\overleftarrow{X}_0, 0) = \mathbb{E}\left[\int_0^T \frac{\partial f}{\partial t}(\overleftarrow{X}_t, t) + \mathcal{B}[f](\overleftarrow{X}_t, t)dt\middle|\overleftarrow{X}_0\right] \tag{6}$$

where $\mathcal{B}$ is the *backward operator*, defined in our setting as:

$$\mathcal{B}[f](x, t) = \sum_{y \neq x} \overleftarrow{Q}_t(y, x)(f(y) - f(x)). \tag{7}$$

By combining the result from Equation (6) with Equation (5), we obtain the following expression for the KL divergence between discrete distributions $\text{KL}\left[\overrightarrow{p}_0 \parallel \overrightarrow{q}_0\right]$:

$$\mathbb{E}\left[\int_0^T \sum_{y \neq \overrightarrow{X}_t} \overrightarrow{Q}_t(\overrightarrow{X}_t, y)\left(K\left(\frac{\overrightarrow{p}_t(y)}{\overrightarrow{p}_t(\overrightarrow{X}_t)}\right) + \frac{\overrightarrow{q}_t(y)}{\overrightarrow{q}_t(\overrightarrow{X}_t)} - \frac{\overrightarrow{p}_t(y)}{\overrightarrow{p}_t(\overrightarrow{X}_t)} \log \frac{\overrightarrow{q}_t(y)}{\overrightarrow{q}_t(\overrightarrow{X}_t)}\right) dt\right] \tag{8}$$

where $K(a) = a(\log(a) - 1)$. The missing term $\mathbb{E}\left[\log \frac{\overleftarrow{p}_0}{\overleftarrow{q}_0}(\overleftarrow{X}_0)\right]$ is omitted, as both $\overleftarrow{p}_0$ and $\overleftarrow{q}_0$ converge to π (Lou et al., 2024).

While Equation (8) provides a complete formulation for estimating the divergence of interest, in practical applications—similar to those in generative modeling (Lou et al., 2024; Ren et al., 2024; Holderrieth et al., 2024)—the key quantities $\frac{\overrightarrow{p}_t(y)}{\overrightarrow{p}_t(x)}$ and $\frac{\overrightarrow{q}_t(y)}{\overrightarrow{q}_t(x)}$ are not directly accessible. To address this, we adopt the approach of Lou et al. (2024), replacing these unknown ratios with parametric approximations optimized via Equation (4). This leads to the construction of the estimator:

$$\int_0^T \mathbb{E}\left[\sum_{y \neq \overrightarrow{X}_t} \overrightarrow{Q}_t(\overrightarrow{X}_t, y)\left(K\left(s_\theta^p(\overrightarrow{X}_t)_y\right) + s_\theta^q(\overrightarrow{X}_t)_y - s_\theta^p(\overrightarrow{X}_t)_y \log s_\theta^q(\overrightarrow{X}_t)_y\right)\right] dt \tag{9}$$

where $s_\theta^p(\overleftarrow{X}_t)_y \approx \frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}$ and $s_\theta^q(\overleftarrow{X}_t)_y \approx \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)}$ serve as parametric approximations of the true ratios. Estimating Equation (9) using Monte Carlo techniques is conceptually straightforward: we sample time instants t uniformly in $[0, T]$, simulate the forward process X_t, and compute the required quantities using the parametric scores. However, as we will discuss in the next Section, the estimator in its general form is not scalable for computing information metrics, which is the primary focus of this work.

3 INFO-SEDD

In this section, we introduce our mutual information estimator, INFO-SEDD, which is based on Equation (9). Given two random variables, $\overrightarrow{X}_0$ and $\overrightarrow{Y}_0$, mutual information can be expressed in

terms of the KL-divergence between the joint distribution $\overrightarrow{p}_0$ and the product of the marginals, defined as $\overrightarrow{m}_0 = \overrightarrow{p}_0^X \otimes \overrightarrow{p}_0^Y$, where $\otimes$ denotes the Kronecker product. This leads to the standard formulation: $I(\overrightarrow{X}_0, \overrightarrow{Y}_0) = \text{KL}\left[\overrightarrow{p}_0 \parallel \overrightarrow{m}_0\right]$. However, this approach presents two key limitations, which we address in the following.

First, in high dimensional applications, a naive implementation of Equation (9) quickly becomes unfeasible. Indeed, the size of the number of entries of the matrix $\overrightarrow{Q}_t$ scales with $|\chi|^2$, becoming quickly untractable. Fortunately, in many cases of interest, the random variables can be naturally decomposed into a structured sequence of M subcomponents, each taking values from a discrete set of size N (Lou et al., 2024; Austin et al., 2021; Campbell et al., 2022), i.e. $\overrightarrow{X}_0 = [\overrightarrow{X}_0^1, \dots, \overrightarrow{X}_0^M]$, leading to a total state space of size $|\chi| = N^M$.This structured decomposition enables the use of sparse rate matrices, which constrain the CTMC to modify only one subcomponent at a time, significantly reducing computational complexity:

$$\overrightarrow{Q}_t(x, y) = \delta(d_{\text{hamming}}(x, y), 1) \left(\sum_i (1 - \delta(x^i, y^i)) \overrightarrow{Q}_t^{\text{tok}}(x_i, y_i) \right), \quad x \neq y \tag{10}$$

Since the summation in the inner expectation of Equation (9) ($\sum_{y \neq \overrightarrow{X}_t} \overrightarrow{Q}_t(\overrightarrow{X}_t, y) \dots$) depends only on non-zero entries of the matrix $\overrightarrow{Q}_t(\overrightarrow{X}_t, y)$, the formulation in Equation (10) greatly reduces the number of transitions that need to be considered.

Second, formulating mutual information as the KL divergence between the joint distribution and the product of marginals, $\text{KL}\left[\overrightarrow{p}_0 \parallel \overrightarrow{m}_0\right]$, typically requires training two separate score models, each tailored to a specific distribution. However, a carefully chosen transition matrix can circumvent this requirement, allowing for a single unified model to be trained instead. In particular, selecting $\overrightarrow{Q}_t^{\text{tok}} = \sigma(t)\overrightarrow{Q}_{\text{absorb}}^{\text{tok}}$, with $\sigma(t)$ a fixed scalar function, and the absorbing matrix [1] (Lou et al., 2024; Campbell et al., 2022; Austin et al., 2021),

$$\overrightarrow{Q}_{\text{absorb}}^{\text{tok}} = \begin{bmatrix} -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 0 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \tag{11}$$

ensures that the subcomponents can only transition into an absorbing state $\emptyset$. This choice is crucial because it enables the computation of marginal scores using a model trained solely on the joint distribution. Specifically, as demonstrated in Appendix A.3,

$$\frac{\overrightarrow{p}_t(\overrightarrow{X}_t = x, \overrightarrow{Y}_t = \emptyset)}{\overrightarrow{p}_t(\overrightarrow{X}_t = x', \overrightarrow{Y}_t = \emptyset)} = \frac{\overrightarrow{p}_t^X(x)}{\overrightarrow{p}_t^X(x')}, \quad \frac{\overrightarrow{p}_t(\overrightarrow{X}_t = \emptyset, \overrightarrow{Y}_t = y)}{\overrightarrow{p}_t(\overrightarrow{X}_t = \emptyset, \overrightarrow{Y}_t = y')} = \frac{\overrightarrow{p}_t^Y(y)}{\overrightarrow{p}_t^Y(y')}. \tag{12}$$

This result implies that a single score model trained on the joint distribution is sufficient for computing the marginal scores as well. By integrating these design choices, we now present the full formulation of INFO-SEDD, whose pseudocode is detailed in Algorithm 1.

[1] This configuration adds an absorbing state, increasing the dimension of the support

Algorithm 1 INFO-SEDD: Estimate $I(\mathbf{x}, \mathbf{y})$

Require: Initial sample $[\vec{X}_0, \vec{Y}_0] \sim \vec{p}_0$, score network s_θ
$t \sim u(0, T)$ {Sample time uniformly}
$[\vec{X}_t, \vec{Y}_t] \sim \vec{p}_t(\cdot|[\vec{X}_0, \vec{Y}_0])$ {Perturb data}
$\hat{I} = 0$
for $i : \vec{X}_t^i = \emptyset$ **do**
  for $n \in [1 : N]$ **do**
    $\tilde{X} = [\vec{X}_t^1, \ldots, \vec{X}_t^{i-1}, n, \vec{X}_t^{i+1}, \ldots, \vec{X}_t^M]$
    $\hat{I} += T\sigma(t)\left(K\left(s_\theta([\vec{X}_t, \vec{Y}_t])_{[\tilde{X}, \vec{Y}_t]}\right) + s_\theta([\vec{X}_t, \emptyset])_{[\tilde{X}, \emptyset]} - s_\theta([\vec{X}_t, \vec{Y}_t])_{[\tilde{X}, \vec{Y}_t]} \log\left(s_\theta([\vec{X}_t, \emptyset])_{[\tilde{X}, \emptyset]}\right)\right)$
  end for
end for
for $i : \vec{Y}_t^i = \emptyset$ **do**
  for $n \in [1 : N]$ **do**
    $\tilde{Y} = [\vec{Y}_t^1, \ldots, \vec{Y}_t^{i-1}, n, \vec{Y}_t^{i+1}, \ldots, \vec{Y}_t^M]$
    $\hat{I} += T\sigma(t)\left(K\left(s_\theta([\vec{X}_t, \vec{Y}_t])_{[\vec{X}_t, \tilde{Y}]}\right) + s_\theta([\emptyset, \vec{Y}_t])_{[\emptyset, \tilde{Y}]} - s_\theta([\vec{X}_t, \vec{Y}_t])_{[\vec{X}_t, \tilde{Y}]} \log\left(s_\theta([\emptyset, \vec{Y}_t])_{[\emptyset, \tilde{Y}]}\right)\right)$
  end for
end for
return $\hat{I}$

3.1 ESTIMATING ENTROPY

Notably, the proposed class of estimators can be readily adapted for entropy estimation. The entropy of a given distribution can be expressed in terms of the KL divergence from the uniform distribution $\vec{u}_0$, as follows: $H(\vec{p}_0) = \log N - \text{KL}\left[\vec{p}_0 \parallel \vec{u}_0\right]$. Since the ratio $\frac{\vec{u}_t(x)}{\vec{u}_t(\emptyset)} = \frac{1}{N(e^{\overline{\sigma}(t)}-1)}$, with $\overline{\sigma}(t) = \int_0^t \sigma(s)ds$ (see appendix A.2), we can extend the formulation of Equation (9) to derive INFO-SEDD-H, an entropy estimator. The working mechanism of this estimator is straightforward and is detailed in Algorithm 2.

Algorithm 2 INFO-SEDD-H: Estimate $H(\vec{X}_0)$

Require: Initial sample $\vec{X}_0 \sim \vec{p}_0$, score network s_θ
$t \sim u(0, T)$ {Sample time uniformly}
$\vec{X}_t \sim \vec{p}_t(\cdot|\vec{X}_0)$ {Perturb data}
$\hat{H} = 0$
for $i : \vec{X}_t^i = \emptyset$ **do**
  for $n \in [1 : N]$ **do**
    $\tilde{X} = [\vec{X}_t^1, \ldots, \vec{X}_t^{i-1}, n, \vec{X}_t^{i+1}, \ldots, \vec{X}_t^M]$
    $\hat{H} += T\sigma(t)\left(K\left(s_\theta(\vec{X}_t)_{\tilde{X}}\right) + \frac{1}{N(e^{\overline{\sigma}(t)}-1)} - s_\theta(\vec{X}_t)_{\tilde{X}} \log\left(\frac{1}{N(e^{\overline{\sigma}(t)}-1)}\right)\right)$
  end for
end for
return $\hat{H}$

4 EXPERIMENTAL VALIDATION

In this section, we numerically validate the performance of INFO-SEDD on both synthetic and real-world datasets. Specifically, we evaluate the mutual information and entropy estimators presented in Algorithm 1 and Algorithm 2, respectively, through the following experiments: (i) benchmarking on high-dimensional distributions where ground truth values are known by construction, and (ii) assessing the accuracy of our method in a *real-world* application by estimating the entropy of spin glass configurations in the Ising model (Onsager, 1944).

4.1 SYNTHETIC BENCHMARK

Real-world data distributions are rapidly increasing in complexity. Modern language models (Radford et al., 2019; Dubey et al., 2024; Lou et al., 2024) process sequences consisting of thousands of tokens, drawn from vocabularies of tens of thousands of tokens. These correspond to the structured discrete variables introduced earlier, where tokens map to sequence length M, and vocabulary size corresponds to the number of states per subcomponent N. As mutual information quantifies dependencies, its value increases rapidly with N^M, making estimation challenging.

To evaluate the feasibility of mutual information estimation in high-dimensional discrete settings, we design synthetic experiments that highlight the limitations of estimators originally developed for continuous variables. We control three key factors: support size (N), representation dimension (M), and mutual information value. By varying one factor at a time, we systematically assess whether continuous-variable estimators fail while INFO-SEDD scales effectively.

We generate joint distributions for random variables X, Y with user-defined mutual information and support sizes χ_X, χ_Y. Using an evolutionary strategy, we encode the joint distribution in a vector $g_a \in \mathbb{R}^{|\chi_X||\chi_Y|}$ and transform it into a valid probability distribution via normalization and reshaping: $\frac{g_a - \min(g_a)\mathbf{1} + \epsilon\mathbf{1}}{\mathbf{1}(g_a - \min(g_a)\mathbf{1} + \epsilon\mathbf{1})}$ where ϵ ensures full support. The mutual information, computable in closed form, serves as the selection criterion in the evolutionary process. For large $|\chi_X|$ and $|\chi_Y|$, the evolutionary strategy struggles. Instead, we generate high mutual information distributions by concatenating independent distributions, leveraging the additive property of mutual information. Additionally, isomorphisms like Cantor's pairing function, $\pi(x, y) = \frac{1}{2}(x+y)(x+y+1)+y$, enables support expansion without altering mutual information, aiding consistency across experiments. Full details are in Appendix B.1.

Experimental Setup To evaluate the performance of our mutual information estimator, INFO-SEDD, we design three sets of controlled synthetic experiments, each isolating a key aspect of data complexity. In the first experiment, we fix the mutual information value at 0.5 and the length of the random vector at 2, meaning each of the two random variables has one dimension. We then increase the support size using Cantor's mapping (Appendix B.1). The second experiment maintains a mutual information value of 0.5 and binary support for each element of the random vector, but instead increases the length of the representation vector. Finally, in the third experiment, we generate distributions with varying mutual information values while keeping the support binary for each variable. The length of the random vectors is fixed at 10, and mutual information values are linearly spaced from 0 to 5. We compare the results of our proposed methodology against MINE (Belghazi et al., 2018), a variational neural estimator, MINDE (Franzese et al., 2023), a generative neural estimator and KSG (Kraskov et al., 2004), a *classical* statistical estimator. For a fair comparison, we use Multi-Layer-Perceptrons (MLPs) based architectures for the neural estimators and scale parameters appropriately when needed (see Appendix B.2). When an estimate is not available, we fill the entry in the tables with $-$. We report all results in nats.

Results and Analysis The results from these experiments, shown in Tables 1 to 3, demonstrate that INFO-SEDD consistently outperforms competing methods. Notably, our model excels in settings where an inherent property of discrete systems—such as support size, representation length, or mutual information—introduces increased complexity.

Although our competitors perform relatively well in some benchmarks, they fail in at least one experiment. KSG performs well with large support size (Table 1), but it fails when considering higher dimensions (Table 2). MINE and MINDE, on the contrary, excels with higher dimensions but struggle with large support size. MINDE also struggles more with higher mutual information values (Table 3). Overall, these experiments motivate the usage of adequate neural estimators when dealing with discrete distributions.

4.2 SPIN GLASSES EXPERIMENTS

Entropy computation in Ising models enables insights on the thermodynamics properties of the system (Cincio et al., 2007), which can be used for scientific discovery in the domain where the Ising model is applied (Macy et al., 2024; Schneidman et al., 2006; Sherrington & Kirkpatrick, 1975).

Table 1: Different support dimension $|\chi|$ and MI=0.5

$\lvert\chi\rvert$	MINDE	MINE	I-SEDD	KSG
2	0.51	0.50	0.50	0.49
4	0.04	0.46	0.53	0.49
16	0.02	0.00	0.51	0.49
64	0.00	-12644	0.54	0.48
256	0.00	0.00	0.58	0.44
1024	0.00	-	0.61	0.24

Table 2: Different random vector lengths and MI=0.5

Length	MINDE	MINE	I-SEDD	KSG
2	0.55	0.50	0.50	0.51
8	0.39	0.51	0.55	0.51
32	0.27	0.73	0.52	0.07
128	0.35	0.43	0.48	0.00
512	0.38	1.22	1.60	0.00
2048	5.56	0.00	5.19	0.01

Table 3: Different MI values

MI	MINDE	MINE	I-SEDD	KSG
0	0.40	0.02	0.00	0.01
1	0.23	1.26	1.01	0.49
2	1.44	2.29	2.03	1.14
3	2.17	3.08	3.02	2.15
4	3.51	3.84	4.01	3.39
5	17.92	4.93	5.07	4.72

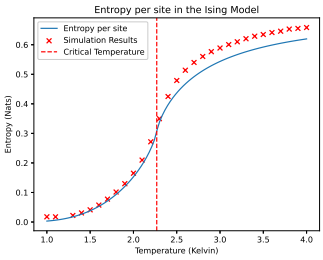

Figure 1: Entropy of the Ising model at different temperatures

Experimental setup We consider a simplified Ising model applied to spin glasses (Sherrington & Kirkpatrick, 1975). We do not include an external field, we set a unitary interaction strength for all the sites interactions, unitary Boltzmann's constant and a 20×20 square lattice. The entropy per site of this configuration can be computed in closed form Onsager (1944). We test our model by estimating the entropy per particle at linearly spaced temperatures from $1.0K$ to $4.0K$. We generate our dataset using the Metropolis-Hastings algorithm, with 10000 samples for each temperature (see Appendix B.2.2). We post-process the output of INFO-SEDD by dividing the entropy estimate by 400 to report the entropy per site.

Results and Analysis Variational estimators cannot estimate large KLs divergences reliably with limited samples sizes (McAllester & Stratos, 2020; Song & Ermon, 2019a). In this scenario, instead, INFO-SEDD accurately estimates large KLs divergences (Figure 1), performing particularly well at low temperatures where we need to estimate large KLs divergences.

5 Conclusion

Our work introduces INFO-SEDD, a novel method for estimating information-theoretic quantities in discrete data using CTMCs. By leveraging a single parametric model, our approach offers computational and memory efficiency while maintaining accuracy in high-dimensional settings. Our experiments highlight INFO-SEDD's robustness against existing neural estimators and demonstrate its effectiveness on both synthetic benchmarks and real-world applications, such as entropy estimation in the Ising model. These results underscore the importance of using specialized techniques for discrete distributions and open new avenues for scalable and accurate information-theoretic analysis in complex systems.

6 Acknowledgments

Alberto Foresti, Giulio Franzese and Pietro Michiardi were partially funded by project MUSECOM2- AI-enabled MUltimodal SEmantic COMmunications and COMputing, in the Machine Learning based Communication Systems, towards Wireless AI (WAI), Call 2022, ChistERA.

REFERENCES

Alexander A Alemi and Ian Fischer. Gilbo: One metric to measure them all. *Advances in Neural Information Processing Systems*, 31, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2016.

William J Anderson. *Continuous-time Markov chains: An applications-oriented approach*. Springer Science & Business Media, 2012.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Gyan Bhanot. The metropolis algorithm. *Reports on Progress in Physics*, 51(3):429, 1988.

Mustapha Bounoua, Giulio Franzese, and Pietro Michiardi. S $omega$ i: Score-based o-information estimation. In *Forty-first International Conference on Machine Learning*, 2024.

Rob Brekelmans, Sicong Huang, Marzyeh Ghassemi, Greg Ver Steeg, Roger Baker Grosse, and Alireza Makhzani. Improving mutual information estimation with annealed and energy-based bounds. In *International Conference on Learning Representations*, 2022.

Ivan Butakov, Alexander Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, and Alexey Frolov. Mutual information estimation via normalizing flows. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=JiQXsLvDls`.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Barry Chai, Dirk Walther, Diane Beck, and Li Fei-Fei. Exploring functional connectivities of the human brain using multivariate information analysis. *Advances in neural information processing systems*, 22, 2009.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Lukasz Cincio, Jacek Dziarmaga, Marek M Rams, and Wojciech H Zurek. Entropy of entanglement and correlations induced by a quench: Dynamics of a quantum phase transition in the quantum ising model. *Physical Review A—Atomic, Molecular, and Optical Physics*, 75(5):052321, 2007.

Maxime Darrin, Philippe Formont, Jackie Chi Kit Cheung, and Pablo Piantanida. `COSMIC`: Mutual information for task-agnostic summarization evaluation, 2024. URL `https://arxiv.org/abs/2402.19457`.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Andrew W. Eckford, Kenneth A. Loparo, and Peter J. Thomas. Finite-state channel models for signal transduction in neural systems. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6300–6304. IEEE, March 2016. doi: 10.1109/icassp.2016.7472889. URL http://dx.doi.org/10.1109/ICASSP.2016.7472889.

Marco Federici, David Ruhe, and Patrick Forré. On the effectiveness of hybrid mutual information estimation. *arXiv preprint arXiv:2306.00608*, 2023.

Giulio Franzese, Mustapha Bounoua, and Pietro Michiardi. Minde: Mutual information neural diffusion estimation. *arXiv preprint arXiv:2310.09031*, 2023.

Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial intelligence and statistics*, pp. 277–286. PMLR, 2015.

Floyd B Hanson. *Applied stochastic processes and control for jump-diffusions: modeling, analysis and computation*. SIAM, 2007.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. *arXiv preprint arXiv:2410.20587*, 2024.

Sicong Huang, Alireza Makhzani, Yanshuai Cao, and Roger Grosse. Evaluating lossy compression rates of deep generative models. In *International Conference on Machine Learning*. PMLR, 2020.

Jan Karbowski. Information thermodynamics: From physics to neuroscience. *Entropy*, 26(9):779, September 2024. ISSN 1099-4300. doi: 10.3390/e26090779. URL http://dx.doi.org/10.3390/e26090779.

Frank P Kelly. Reversibility and stochastic networks. *PF Kelly—New York: Willy*, 1981.

Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *International Conference on Learning Representations*, 2022.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

Kyungeun Lee and Wonjong Rhee. A benchmark suite for evaluating neural mutual information estimators on unstructured datasets, 2024. URL https://arxiv.org/abs/2410.10924.

Nunzio A Letizia and Andrea M Tonello. Copula density neural estimation. *arXiv preprint arXiv:2211.15353*, 2022.

Nunzio A Letizia, Nicola Novello, and Andrea M Tonello. Variational f-divergence and derangements for discriminative mutual information estimation. *arXiv preprint arXiv:2305.20025*, 2023.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *stat*, 1050:21, 2024.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Michael W Macy, Boleslaw K Szymanski, and Janusz A Hołyst. The ising model celebrates a century of interdisciplinary contributions. *npj Complexity*, 1(1):10, 2024.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318, 1995.

Garin Newcomb and Khalid Sayood. Use of average mutual information and derived measures to find coding regions. *Entropy*, 23(10):1324, 2021.

XuanLong Nguyen, Martin J Wainwright, and Michael Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Advances in Neural Information Processing Systems*, 2007.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. F-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 271–279, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117, 1944.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Advances in neural information processing systems*, 2018.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Assaf Pinchas, Irad Ben-Gal, and Amichai Painsky. A comparative analysis of discrete entropy estimators for large-alphabet problems. *Entropy*, 26(5):369, 2024.

Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Yinuo Ren, Haoxuan Chen, Grant M Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. *arXiv preprint arXiv:2410.03601*, 2024.

Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 2020.

Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical review letters*, 35(26):1792, 1975.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2019a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.

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.

Karl Stratos. Mutual information maximization for simple and accurate part-of-speech induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models, 2023. URL `https://arxiv.org/abs/2211.16750`.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Gerhard Wunder, Benedikt Groß, Rick Fritschek, and Rafael F Schaefer. A reverse jensen inequality result with application to mutual information estimation. In *2021 IEEE Information Theory Workshop (ITW)*, 2021.

Jun Xia, Shaorong Chen, Jingbo Zhou, Xiaojun Shan, Wenjie Du, Zhangyang Gao, Cheng Tan, Bozhen Hu, Jiangbin Zheng, and Stan Z Li. Adanovo: Towards robust\emph {De Novo} peptide sequencing in proteomics against data biases. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. A lagrangian perspective on latent variable generative models. In *Proc. 34th Conference on Uncertainty in Artificial Intelligence*, 2018.

A Appendix

A.1 Proof of equation 8

We first break $\mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T}\middle|\overleftarrow{X}_0\right]$ in $\mathbb{E}\left[\log \overleftarrow{p}_T\middle|\overleftarrow{X}_0\right]-\mathbb{E}\left[\log \overleftarrow{q}_T\middle|\overleftarrow{X}_0\right]$ and apply the Dynkin's formula separately. We start with $\mathbb{E}\left[\log \overleftarrow{p}_T\middle|\overleftarrow{X}_0\right]$:

$$\begin{aligned}
\mathbb{E}\left[\log \overleftarrow{p}_T\middle|\overleftarrow{X}_0\right] &= \log \overleftarrow{p}_0(\overleftarrow{X}_0) + \mathbb{E}\left[\int_0^T \frac{\partial \log \overleftarrow{p}_t}{\partial t}(\overleftarrow{X}_t, t) + \mathcal{B}[\log \overleftarrow{p}_t](\overleftarrow{X}_t, t)dt\middle|\overleftarrow{X}_0\right] \\
&= \log \overleftarrow{p}_0(\overleftarrow{X}_0) + \mathbb{E}\left[\int_0^T \frac{\partial \log \overleftarrow{p}_t}{\partial t}(\overleftarrow{X}_t, t) + \sum_{y\neq \overleftarrow{X}_t} \overleftarrow{Q}_t(y, \overleftarrow{X}_t)(\log \overleftarrow{p}_t(y) - \log \overleftarrow{p}_t(\overleftarrow{X}_t))dt\middle|\overleftarrow{X}_0\right] \\
&= \log \overleftarrow{p}_0(\overleftarrow{X}_0) + \mathbb{E}\left[\int_0^T \frac{\partial \log \overleftarrow{p}_t}{\partial t}(\overleftarrow{X}_t, t) + \sum_{y\neq \overleftarrow{X}_t} \overleftarrow{Q}_t(y, \overleftarrow{X}_t) \log \frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}dt\middle|\overleftarrow{X}_0\right]
\end{aligned}$$

We now focus on the term $\frac{\partial \log \overleftarrow{p}_t}{\partial t}$, which we can rewrite using the definition given in equation 2:

$$\frac{\partial \log \overleftarrow{p}_t}{\partial t} = \frac{\partial \overleftarrow{p}_t}{\partial t}\Big/\overleftarrow{p}_t = \frac{\overleftarrow{Q}_t \overleftarrow{p}_t}{\overleftarrow{p}_t} \tag{13}$$

Where the division between numerator and denominator in the last two terms denotes element-wise division. We focus now on simplifying the numerator, using the definition in equation 14. Recalling that

$$\overleftarrow{Q}_t(y,x) = \left(\frac{\overrightarrow{p}_{T-t}(y)}{\overrightarrow{p}_{T-t}(x)}\overrightarrow{Q}_{T-t}(x,y)\right)(1-\delta(x,y)) + \left(-\sum_{y\neq x}\overleftarrow{Q}_t(y,x)\right)\delta(x,x) \tag{14}$$

$$= \left(\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(x)}\overrightarrow{Q}_{T-t}(x,y)\right)(1-\delta(x,y)) + \left(-\sum_{y\neq x}\overleftarrow{Q}_t(y,x)\right)\delta(x,x) \tag{15}$$

we compute the x-th element of $\overleftarrow{Q}_t\overleftarrow{p}_t$:

$$\begin{aligned}
[\overleftarrow{Q}_t\overleftarrow{p}_t](x) &= \sum_y \overleftarrow{Q}_t(x,y)\overleftarrow{p}_t(y) \\
&= \sum_{y\neq x}\overleftarrow{Q}_t(x,y)\overleftarrow{p}_t(y) + \overleftarrow{Q}_t(x,x)\overleftarrow{p}_t(x) \\
&= \sum_{y\neq x}\overrightarrow{Q}_{T-t}(y,x)\frac{\overleftarrow{p}_t(x)}{\overleftarrow{p}_t(y)}\overleftarrow{p}_t(y) - \sum_{y\neq x}\overleftarrow{Q}_t(y,x)\overleftarrow{p}_t(x) \\
&= \sum_{y\neq x}\overrightarrow{Q}_{T-t}(y,x)\overleftarrow{p}_t(x) - \sum_{y\neq x}\overrightarrow{Q}_{T-t}(x,y)\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(x)}\overleftarrow{p}_t(x) \\
&= \sum_{y\neq x}\overrightarrow{Q}_{T-t}(y,x)\overleftarrow{p}_t(x) - \sum_{y\neq x}\overrightarrow{Q}_{T-t}(x,y)\overleftarrow{p}_t(y) \\
&= \sum_{y\neq x}\overrightarrow{Q}_{T-t}(y,x)\overleftarrow{p}_t(x) - \overrightarrow{Q}_{T-t}(x,y)\overleftarrow{p}_t(y)
\end{aligned} \tag{16}$$

Finally, if we divide by the denominator we obtain:

$$\begin{aligned}
\left[\frac{\overleftarrow{Q}_t\overleftarrow{p}_t}{\overleftarrow{p}_t}\right](x) &= \frac{\sum_{y\neq x}\overrightarrow{Q}_{T-t}(y,x)\overleftarrow{p}_t(x) - \overrightarrow{Q}_{T-t}(x,y)\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(x)} \\
&= \sum_{y\neq x}\overrightarrow{Q}_{T-t}(y,x) - \overrightarrow{Q}_{T-t}(x,y)\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(x)}
\end{aligned} \tag{17}$$

Moreover, if we notice that $\overleftarrow{Q}_t(y,\overleftarrow{X}_t)\log\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)} = \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t,y)\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}\log\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}$, we can write the Dynkin's formula as:

$$\begin{aligned}
&\mathbb{E}\left[\log\overleftarrow{p}_T|\overleftarrow{X}_0\right] = \\
&\log\overleftarrow{p}_0(\overleftarrow{X}_0)+ \\
&\mathbb{E}\left[\int_0^T\sum_{y\neq\overleftarrow{X}_t}\overrightarrow{Q}_{T-t}(y,\overleftarrow{X}_t) - \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t,y)\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)} + \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t,y)\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}\log\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}dt\middle|\overleftarrow{X}_0\right]
\end{aligned} \tag{18}$$

Then, if we define $K(a) = a(\log a - 1)$ and group some terms we obtain:

$$\mathbb{E}\left[\log \overleftarrow{p}_T | \overleftarrow{X}_0\right] = \tag{19}$$
$$\log \overleftarrow{p}_0(\overleftarrow{X}_0) + \mathbb{E}\left[\int_0^T \sum_{y \neq \overleftarrow{X}_t} \overrightarrow{Q}_{T-t}(y, \overleftarrow{X}_t) + \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t, y) K\left(\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}\right) dt \middle| \overleftarrow{X}_0\right]$$

We now repeat similar calculations for $\mathbb{E}\left[\log \overleftarrow{q}_T \middle| \overleftarrow{X}_0\right]$. Firstly, the term $\frac{\partial \log \overleftarrow{q}_t}{\partial t}$ becomes:

$$\left[\frac{\partial \log \overleftarrow{q}_t}{\partial t}\right](x) = \sum_{y \neq x} \overrightarrow{Q}_{T-t}(y, x) - \overrightarrow{Q}_{T-t}(x, y) \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(x)} \tag{20}$$

Whereas the backward operator term $\mathcal{B}[\log \overleftarrow{q}_t](\overleftarrow{X}_t, t)$ becomes:

$$\overleftarrow{Q}_t(y, \overleftarrow{X}_t) \log \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)} = \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t, y) \frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)} \log \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)} \tag{21}$$

Putting everything together gives:

$$\mathbb{E}\left[\log \overleftarrow{q}_T \middle| \overleftarrow{X}_0\right] = \tag{22}$$
$$\log \overleftarrow{q}_0(\overleftarrow{X}_0) +$$
$$\mathbb{E}\left[\int_0^T \sum_{y \neq \overleftarrow{X}_t} \overrightarrow{Q}_{T-t}(y, \overleftarrow{X}_t) - \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t, y) \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)} + \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t, y) \frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)} \log \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)} dt \middle| \overleftarrow{X}_0\right]$$

Finally, we can estimate $\mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T} \middle| \overleftarrow{X}_0\right]$ by subtracting equation 22 from equation 18:

$$\mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T} \middle| \overleftarrow{X}_0\right] \approx \mathbb{E}\left[\int_0^T \sum_{y \neq \overleftarrow{X}_t} \overrightarrow{Q}_{T-t}(\overleftarrow{X}_t, y) \left(K\left(\frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)}\right) + \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)} - \frac{\overleftarrow{p}_t(y)}{\overleftarrow{p}_t(\overleftarrow{X}_t)} \log \frac{\overleftarrow{q}_t(y)}{\overleftarrow{q}_t(\overleftarrow{X}_t)}\right) dt \middle| \overleftarrow{X}_0\right]$$

By using the fact that $\mathbb{E}\left[\mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T} \middle| \overleftarrow{X}_0\right]\right] = \mathbb{E}\left[\log \frac{\overleftarrow{p}_T}{\overleftarrow{q}_T}\right]$, $\overleftarrow{p}_t = \overrightarrow{p}_{T-t}$, $\overleftarrow{q}_t = \overrightarrow{q}_{T-t}$, $\overleftarrow{X}_t = \overrightarrow{X}_{T-t}$ and by setting $\tau = T - t$, we get:

$$\mathbb{E}\left[\int_0^T \sum_{y \neq \overrightarrow{X}_\tau} \overrightarrow{Q}_\tau(\overrightarrow{X}_\tau, y) \left(K\left(\frac{\overrightarrow{p}_\tau(y)}{\overrightarrow{p}_\tau(\overrightarrow{X}_\tau)}\right) + \frac{\overrightarrow{q}_\tau(y)}{\overrightarrow{q}_\tau(\overrightarrow{X}_\tau)} - \frac{\overrightarrow{p}_\tau(y)}{\overrightarrow{p}_\tau(\overrightarrow{X}_\tau)} \log \frac{\overrightarrow{q}_\tau(y)}{\overrightarrow{q}_\tau(\overrightarrow{X}_\tau)}\right) d\tau\right] \tag{23}$$

recovering Equation (8).

A.2 Proof $\frac{\overrightarrow{u}_t(x)}{\overrightarrow{u}_t(\emptyset)} = \frac{1}{N(e^{\overline{\sigma}(t)}-1)}$ for $x \neq \emptyset$

$$\frac{\overrightarrow{u}_t(x)}{\overrightarrow{u}_t(\emptyset)} = \frac{\sum_{x_0 \in \mathcal{X}} \overrightarrow{u}_t(x|x_0) \overrightarrow{u}_0(x_0)}{\sum_{x_0 \in \mathcal{X}} \overrightarrow{u}_t(\emptyset|x_0) \overrightarrow{u}_0(x_0)} = \frac{\sum_{x_0 \in \mathcal{X}} \delta(x, x_0) e^{-\overline{\sigma}(t)} \frac{1}{N}}{\sum_{x_0 \in \mathcal{X}} (1 - e^{-\overline{\sigma}(t)}) \frac{1}{N}} = \frac{e^{-\overline{\sigma}(t)}}{N(1 - e^{-\overline{\sigma}(t)})} = \frac{1}{N(e^{\overline{\sigma}(t)} - 1)}$$

A.3 PROOF OF EQUATION (12)

Consider $\vec{p}_t(\vec{X}_t = \bar{x}, \vec{Y}_t = \emptyset)$:

$$\begin{aligned}
&\vec{p}_t(\vec{X}_t = \bar{x}, \vec{Y}_t = \emptyset) = \sum_{x,y} \text{Pr}(\vec{X}_t = \bar{x}, \vec{Y}_t = \emptyset, \vec{X}_0 = x, \vec{Y}_0 = y) \qquad (24)\\
&= \sum_{x,y} \underbrace{\text{Pr}(\vec{Y}_s = \emptyset \mid \vec{X}_t = \bar{x}, \vec{X}_0 = x, \vec{Y}_0 = y)}_{\text{Pr}(\vec{Y}_s \text{ jumps to } \emptyset \text{ in } [0,t])} \text{Pr}(\vec{X}_t = \bar{x} \mid \vec{X}_0 = x, \vec{Y}_0 = y) \text{Pr}(\vec{X}_0 = x, \vec{Y}_0 = y)\\
&= \sum_{x,y} \text{Pr}(\vec{Y}_s \text{ jumps to } \emptyset \text{ in } [0,t]) \text{Pr}(\vec{X}_t = \bar{x} \mid \vec{X}_0 = x) \text{Pr}(\vec{X}_0 = x, \vec{Y}_0 = y)\\
&= \text{Pr}(\vec{Y}_s \text{ jumps to } \emptyset \text{ in } [0,t]) \sum_x \text{Pr}(\vec{X}_t = \bar{x} \mid \vec{X}_0 = x) \underbrace{\sum_y \text{Pr}(\vec{X}_0 = x, \vec{Y}_0 = y)}_{\text{Pr}(\vec{X}_0 = x)}\\
&= \text{Pr}(\vec{Y}_s \text{ jumps to } \emptyset \text{ in } [0,t]) \text{Pr}(\vec{X}_t = \bar{x})
\end{aligned}$$

Equation (24) implies that $\frac{\vec{p}_t(\vec{X}_t = x, \vec{Y}_t = \emptyset)}{\vec{p}_t(\vec{X}_t = \bar{x}, \vec{Y}_t = \emptyset)} = \frac{\text{Pr}(\vec{X}_t = x)}{\text{Pr}(\vec{X}_t = \bar{x})}$. This important property enables the estimation of mutual information without modifying the score network.

B EXPERIMENTAL DETAILS

B.1 DATASET GENERATION

In our experiments, we exploit the additivity of mutual information with independent random variables to generate complex datasets. By appending discrete noise random variables $Z_{\mathbf{x}}$, $Z_{\mathbf{y}}$ to the original random variables $\vec{X}_0$, $\vec{Y}_0$, we have $I(\vec{X}_0, \vec{Y}_0) = I([\vec{X}_0, Z_{\mathbf{x}}], [\vec{Y}_0, Z_{\mathbf{y}}])$. Pairing functions are isomorphisms that map $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. They allow preserving mutual information through the Markov Chain:

$$[\vec{X}_0, Z_{\mathbf{x}}] \to \hat{X}_0 \to \hat{Y}_0 \to [\vec{Y}_0, Z_{\mathbf{y}}] \to \hat{Y}_0 \to \hat{X}_0 \to [\vec{X}_0, Z_{\mathbf{x}}] \qquad (25)$$

Where, by the *data processing inequality*, we have $I([\vec{X}_0, Z_{\mathbf{x}}], [\vec{Y}_0, Z_{\mathbf{y}}]) \geq I(\hat{X}_0, \hat{Y}_0) \geq I([\vec{X}_0, Z_{\mathbf{x}}], [\vec{Y}_0, Z_{\mathbf{y}}]) \implies I([\vec{X}_0, Z_{\mathbf{x}}], [\vec{Y}_0, Z_{\mathbf{y}}]) = I(\hat{X}_0, \hat{Y}_0) \implies I(\hat{X}_0, \hat{Y}_0) = I(\vec{X}_0, \vec{Y}_0)$.

In Section 4.1 we keep the same support dimension for both random variables, increasing it using the following procedure:

1. We sample two binomial random variables $Z_{\mathbf{x}}, Z_{\mathbf{y}}$ with parameters (n, p), We vary n for increasing the complexity of the experiment and we keep p fixed to 0.5.
2. We concatenate $Z_{\mathbf{x}}$ and $Z_{\mathbf{y}}$ respectively to $\vec{X}_0$ and $\vec{Y}_0$, to form higher support versions $\hat{X}_0, \hat{Y}_0$.
3. We map the noisy $\hat{X}_0, \hat{Y}_0$ versions to univariate random variables by applying Cantor's mapping.

B.2 MODEL AND TRAINING SETUP

For our method, we use a Multi Layer Perceptron (MLP) with skip connection, based on the architecture used in Franzese et al. (2023), reworking the initial layer to include absolute positional

embeddings. For training, we match the methodology used by Lou et al. (2024), using the absorb configuration. Similarly, we follow prior work to train other models in the synthetic benchmark. At inference time, we always take the last valid validation step estimate of each model to avoid not-a-number values in our tables.

B.2.1 SYNTHETIC BENCHMARK

In this section, we describe how we scaled the architectures with increasing complexity in the synthetic benchmark. We benchmark our competitors using architectures from prior work (Franzese et al., 2023; Belghazi et al., 2018). Depending on the task, some architectures are forced to increase the number of parameters. For example, INFO-SEDD is forced to increase the number of parameters with the support size. In order to maintain a fair comparison between different models, we make the other architectures match the number of parameters of the architecture which is forced to include more parameters by either increasing the number of layers or by increasing the layer width.

Big support experiments For supports smaller than 256, we use architectures with a comparable number of parameters, in the order of $20k$. Instead, after this support dimension we increase the number of parameters, with an order of $70k$ parameters for support dimension 256 and an order of $300K$ for support dimension 1024.

Representation length experiments For vectors of length shorter than 512, we use architectures with a comparable number of parameters, remaining in the order of $20k$. Instead, after this length we increase the number of parameters, with an order of $50k$ parameters for length 512 and an order of $150k$ for length 2048.

High mutual information experiments In this case we keep the number of parameters fixed on the order of $20k$, as fixed representation length and fixed support dimension allow MINE and MINDE to keep the number of parameters constant.

B.2.2 ISING MODEL EXPERIMENTS

The Ising model is a system consisting of particles arranged in a lattice. In our experiments, we consider a $L \times L$ square. A particle i of the lattice is associated with a discrete value $\sigma_i \in \{-1, +1\}$ called spin and each pair of particles ij is characterized by an interaction strength J_{ij}. With no external fields, these quantities determine the energy $E(\sigma)$ of the configuration σ:

$$E(\sigma) = \sum_{i,j} J_{ij} \sigma_i \sigma_j \tag{26}$$

In turns, the energy of a configuration determines its likelihood. In particular, the configurations of the Ising model follow a probability distribution $\overrightarrow{p}_0$ parametrised by the temperature T, the Boltzmann constant k_b and the interaction strengths:

$$\overrightarrow{p}_0(\sigma) = \frac{e^{-\beta E(\sigma)}}{Z(T)} \tag{27}$$

Where $Z(T) = \sum_i e^{-\beta E(\sigma_i)}$ and $\beta = (k_b T)^{-1}$. In order to generate our dataset from $\overrightarrow{p}_0$, we follow the Metropolis algorithm (Bhanot, 1988):

Algorithm 3 Metropolis Algorithm for 2D Ising Spin Glass

Input: Lattice size N, interaction strengths J_{ij}, temperature T, number of iterations iter_max
Initialize: Spin lattice σ with $\sigma_{i,j} \in \{-1, +1\}$ randomly assigned
for iteration = 1 to iter_max **do**
  Randomly select a lattice site i
  Compute the change in energy ΔE if $\sigma_{i,j}$ is flipped:

$$\Delta E = 2\,\sigma_i \sum_j J_{i,j}\,\sigma_j$$

  Generate a random number r uniformly distributed in $[0, 1]$
  if $r < \exp(-\beta \Delta E)$ **then**
    Flip the spin: $\sigma_i \leftarrow -\sigma_i$
  end if
end for
Output: Final spin configuration σ

We compute the entropy of $\vec{p}_0$ analytically, starting from the free energy F per site of the lattice:

$$F(T) = -k_b T \log \lambda_T \tag{28}$$

Where λ_T is the partition function, which depends on the interaction horizontal and vertical interaction strength. For simplicity, we consider the same interaction strength $J = 1$ for all neighboring particles, while we set it to zero for non neighboring particles. Under these assumptions, we can calculate $\log \lambda$ with a double integral(Onsager, 1944):

$$\log \lambda_T = \log 2 + \frac{1}{2\pi^2} \int_0^{\pi} \int_0^{\pi} \log(\cosh(2\beta J)\cosh(2\beta J) - \sinh(2\beta J)\cos(\theta_1) - \sinh(2\beta J)\cos(\theta_2)) d\theta_1 d\theta_2 \tag{29}$$

For simplicity, we also set $k_b = 1$. From F, we can calculate the entropy H using the thermodynamic relation $H = -\frac{\partial F}{\partial T}$. We compute the integral numerically using the *SciPy* Python package (Virtanen et al., 2020) and we approximate H as $H \approx \frac{F(T+\Delta T) - F(T-\Delta T)}{2\Delta T}$, with $\Delta T = 10^{-4}$.

For what concerns the INFO-SEDD architecture, we keep a single model configuration for all temperatures, both for the model, which contains around $90k$ parameters, and for the diffusion. To get the entropy per site, we divide the estimates of the model by the number of particles in the configurations (400).