DISCRETE BRIDGES FOR MUTUAL INFORMATION ESTIMATION

Iryna Zabarianska[*]
MIRAI[†], Russia

Sergei Kholkin[*]
Applied AI Institute, Russia

Grigoriy Ksenofontov
Applied AI Institute, Russia,
MIRAI[†], Russia

Ivan Butakov
MIRAI[†], Russia,
Applied AI Institute, Russia,
Institute of Numerical Mathematics (RAS),
Russia

Alexander Korotin
Applied AI Institute, Russia,
AXXX, Russia

ABSTRACT

Diffusion bridge models in both continuous and discrete state spaces have recently become powerful tools in the field of generative modeling. In this work, we leverage the discrete state space formulation of bridge matching models to address another important problem in machine learning and information theory: the estimation of the mutual information (MI) between discrete random variables. By neatly framing MI estimation as a *domain transfer* problem, we construct a Discrete Bridge Mutual Information (**DBMI**) estimator suitable for discrete data, which poses difficulties for conventional MI estimators. We showcase the performance of our estimator on two MI estimation settings: low-dimensional and image-based.

1 INTRODUCTION

Mutual Information (MI) is a fundamental measure of nonlinear statistical dependence between two random vectors, defined as the Kullback-Leibler divergence between the joint distribution $\mathbb{P}_{X_0,X_1}$ and the product of marginals $\mathbb{P}_{X_0} \otimes \mathbb{P}_{X_1}$ Polyanskiy & Wu (2024):

$$\mathsf{I}(X_0; X_1) = \mathsf{KL}\left[\mathbb{P}_{X_0,X_1} \parallel \mathbb{P}_{X_0} \otimes \mathbb{P}_{X_1}\right]$$

Due to several outstanding properties, such as nullification only under statistical independence, invariance to invertible transformations, and ability to capture non-linear dependencies, MI is used extensively for theoretical analysis of overfitting Asadi et al. (2018), hypothesis testing Duong & Nguyen (2022), feature selection Peng et al. (2005) representation learning Hjelm et al. (2019); Butakov et al. (2025), and studying mechanisms behind generalization in deep neural nets Goldfeld et al. (2019); Butakov et al. (2024b). While less explored compared to the continuous case, MI estimation between **discrete random vectors** is often leveraged in fields such as bioinformatics Newcomb & Sayood (2021); Xia et al. (2024), neuroscience Chai et al. (2009), and natural language processing Darrin et al. (2024). Despite conventional non-parametric methods offering strong baselines in discrete setups, they typically degrade as data dimensionality and alphabet size increase Pinchas et al. (2024). This motivates the development of novel neural-based discrete MI estimators capable of capturing intricate interdependencies in complex high-dimensional settings.

Bridge Matching. Diffusion models are a powerful type of generative models that show an impressive quality of data generation on continuous state spaces Ho et al. (2020). However, they have some disadvantages, such as the inability to perform data-to-data translation via diffusion. To tackle this problem, a novel promising approach based on Reciprocal Processes Léonard et al. (2014) and Schrödinger Bridges theory De Bortoli et al. (2021) has emerged. This approach is known as *diffusion bridge matching* and is utilized for learning generative models as diffusion processes for *domain translation*. This type of model has shown itself to be a powerful approach for many applications

[*]Equal contribution.
† Moscow Independent Research Institute of Artificial Intelligence
Corresponding author: <kholkinsd@gmail.com>

in biology Bunne et al. (2023), chemistry Somnath et al. (2023), computer vision Liu et al. (2023), unpaired learning Gushchin et al. (2024a), and mutual information estimation Kholkin et al. (2025).

While the approach was introduced for the continuous state space $\mathbb{R}^D$ it has recently been generalized to discrete state spaces, i.e., *discrete bridge matching*, Ksenofontov & Korotin (2025). Such generalization proved itself useful in many applications: unpaired learning Ksenofontov & Korotin (2025); Kim et al. (2025), text generation Gat et al. (2024), and drug design Igashov et al. (2024).

Contributions. In this work, we study mutual information (MI) estimation between discrete random variables defined on the same state space. We introduce a novel MI estimator based on reciprocal processes and their Markov chain representation (§4.1). Building on this framework and Bridge Matching for discrete domain transfer, we develop a practical estimation algorithm, DBMI (§4.2). We evaluate the method on two benchmarks: a low-dimensional setting and a high-dimensional image-based benchmark (§5). The latter is introduced in this work to assess scalability in complex discrete spaces while retaining access to ground-truth MI (§5.2). To our knowledge, it is among the first benchmarks to combine these properties. We further show that our method outperforms existing MI estimators for discrete domains.

Notation. We consider discrete state spaces of the form $\mathcal{X} = \mathbb{S}^D$, where $\mathbb{S} = 1, 2, \dots, S$ denotes a categorical space and D is the number of dimensions. This formulation is general and widely used in practice. Consider a *time set* $\{t_n\}_{n=0}^{N+1}$, where $0 = t_0 < t_1 < \dots < t_N < t_{N+1} = 1$ are $N \geq 1$ time moments. The space $\mathcal{X}^{N+2}$ is referred to as the *path space* and represents all possible trajectories $(x_0, x_{\text{in}}, x_{t_{N+1}})$, where $x_{\text{in}} \stackrel{\text{def}}{=} (x_{t_1}, \dots, x_{t_N})$ corresponds to the intermediate states. Let $\mathcal{P}(\mathcal{X}^{N+2})$ be the space of probability distributions over paths. Each $r \in \mathcal{P}(\mathcal{X}^{N+2})$ can be interpreted as a discrete in time $\mathcal{X}$-valued stochastic process. We use $r(x_0, x_{\text{in}}, x_{t_{N+1}})$ to denote its probability mass function (PMF) at $(x_0, x_{\text{in}}, x_{t_{N+1}}) \in \mathcal{X}^{N+2}$ and use $r(\cdot|\cdot)$ to denote its conditional distributions, e.g., $r(x_1|x_0)$, $r(x_{\text{in}}|x_0, x_1)$. Finally, we introduce $\mathcal{M}(\mathcal{X}^{N+2}) \subset \mathcal{P}(\mathcal{X}^{N+2})$ as the set of all *Markov processes* r, i.e., those processes which satisfy the equality $r(x_0, x_{\text{in}}, x_{t_{N+1}}) = r(x_0) \prod_{n=1}^{N+1} r(x_{t_n}|x_{t_{n-1}})$. To denote the probability mass function of $r \in \mathcal{P}(\mathcal{X}^{N+2})$ at a point $x_{t_n} \in \mathcal{X}$ at time t_n, we use $r(x_{t_n})$. We write $\mathsf{KL}\left[\cdot \,\|\, \cdot\right]$ to denote the Kullback-Leibler divergence between two distributions or stochastic processes.

2 Background

In this section, we deliver background information on Mutual Information (§2.1) and explain concepts vital to our main result: Reciprocal processes (§2.2), their Markov chain representations (§2.3 and §2.4) and the concept of Bridge Matching for discrete state spaces (§2.5).

2.1 Mutual Information

For discrete $\mathcal{X}$-valued random variables X_0, X_1, with existent joint probability mass function $\pi(x_0, x_1)$, the mutual information (MI) quantifies the KL divergence between the joint distribution and the product of marginals:

$$\mathsf{I}(X_0; X_1) = \mathbb{E}_{\pi(x_0,x_1)} \log \frac{\pi(x_0, x_1)}{\pi(x_0)\pi(x_1)} = \mathsf{KL}\left[\pi(x_0, x_1) \,\|\, \pi(x_0)\pi(x_1)\right] \tag{1}$$

Mutual information is zero if and only if X_0 and X_1 are independent. It is also invariant under injective transforms and admits many other noticeable properties Polyanskiy & Wu (2024).

2.2 Reciprocal Processes

Reciprocal processes form a class of stochastic processes that have recently attracted increasing interest in various domains, including stochastic optimal control Léonard et al. (2014), Schrödinger bridge problems De Bortoli et al. (2021), and diffusion-based generative modeling Gushchin et al. (2024a). Although traditionally defined in continuous time, recent works have adopted discrete-time formulations Gushchin et al. (2024b); Ksenofontov & Korotin (2025), favoring their simplicity and modeling flexibility. In this work, we consider a discrete-time reciprocal process, defined via a reference Markov process $q^{\text{ref}} \in \mathcal{M}(\mathcal{X}^{N+2})$.

Consider a distribution $\pi(x_0, x_1) \in \mathcal{P}(\mathcal{X}^2)$, reference process q^{ref} and define the discrete in time stochastic process r_π:

$$r_\pi(x_0, x_{\text{in}}, x_1) \stackrel{\text{def}}{=} q^{\text{ref}}(x_{\text{in}}|x_0, x_1)\pi(x_0, x_1). \tag{2}$$

The process r_π can be interpreted as the reference process q^{ref} constrained to have marginal distribution $\pi(x_0, x_1)$ at its endpoints, while the inner part, i.e., conditioned on start x_0 and end x_1, reference process $q^{\text{ref}}(x_{\text{in}}|x_0, x_1)$, is also known as a *bridge*. In continuous time, this construction aligns with the classical Doob h-transform framework Palmowski & Rolski (2002). Due to the non-causal nature of trajectory formation, process r_π is, in general, not Markov. The set of all such r_π for a particular q^{ref} can be described as:

$$\mathcal{R}(q^{\text{ref}}) = \{r \in \mathcal{P}(\mathcal{X}^{N+2}) \; s.t. \; \exists \pi \in \mathcal{P}(\mathcal{X}^2) : r = r_\pi\}.$$

and we call it a set of **reciprocal processes** for q^{ref}.

2.3 Reciprocal Processes Conditioned on a Point

Consider a reciprocal process r_π and its corresponding conditional process $r_\pi(x_{\text{in}}, x_1|x_0) = r_{\pi|x_0}(x_{\text{in}}, x_1)$. One can be observed, process $r_{\pi|x_0}(x_{\text{in}}, x_1)$ is Markov:

Proposition 1. *Consider the reciprocal process conditioned on point x_0, $r_{\pi|x_0}(x_{\text{in}}, x_1)$. Then $r_{\pi|x_0}(x_{\text{in}}, x_1)$ is Markov:*

$$r_{\pi|x_0}(x_{\text{in}}, x_1) = \prod_{n=1}^{N+1} r_{\pi|x_0}(x_{t_n}|x_{t_{n-1}}) \tag{3}$$

See Appendix A for proof. Conditioning of reciprocal process on the start point can be generalized to reciprocal process starting from $\delta(x_0)$, i.e., Dirac delta distribution, and is a rare case of a stochastic process when both Markov and reciprocal properties are present. This also holds in continuous time, where the processes are known as Schrödinger–Föllmer processes Vargas et al. (2023).

2.4 Reciprocal Processes as Conditioned Markov Chains

Let us note that one can naturally represent the whole r_π as the mixture of $r_{\pi|x_0}$ over initial states x_0 and decompose $r_{\pi|x_0}$ as the Markov chain, see equation 3:

$$r_\pi(x_0, x_{\text{in}}, x_1) = \pi(x_0) \prod_{n=1}^{N+1} r_{\pi|x_0}(x_{t_{n+1}}|x_{t_n}) = \pi(x_0) \prod_{n=1}^{N+1} r_\pi(x_{t_{n+1}}|x_{t_n}, x_0) \tag{4}$$

This represents r_π as a *Markov process family* conditioned on x_0, i.e., $r_{\pi|x_0}$. In that light, if one does know all the transition probabilities $r_\pi(x_{t_{n+1}}|x_{t_n}, x_0)$ and marginal $\pi(x_0)$, then one can sample from r_π by performing the inference of Markov chain.

2.5 Bridge Matching for Discrete State Spaces

The $r_{\pi|x_0}$ admits a Markov chain formulation as given in equation 3 and it has analytic form:

$$r_{\pi|x_0}(x_{t_n}|x_{t_{n-1}}, x_0) = \mathbb{E}_{\pi(x_1|x_0)}\left[q^{\text{ref}}(x_{t_n}|x_{t_{n-1}}, x_1)\right], \tag{5}$$

where $q^{\text{ref}}(x_{t_n}|x_{t_{n-1}}, x_1)$ is also known as *posterior* and is usually analytically known and easily computable in exact form Austin et al. (2021). However, the expectation in equation 5 is taken with respect to $\pi(x_1|x_0)$, which in practice is typically available only through empirical data samples; consequently, an exact evaluation of equation 5 is rarely feasible.

Fortunately, they can be recovered as the solution to the following problem, (Ksenofontov & Korotin, 2025, Prop. 3.3):

$$r_{\pi|x_0} = \arg\min_s \mathbb{E}_{\pi(x_1|x_0)}\Big[\sum_{n=1}^{N} \mathbb{E}_{q^{\text{ref}}(x_{t_{n-1}}|x_0,x_1)} \mathsf{KL}\left[q^{\text{ref}}(x_{t_n}|x_{t_{n-1}}, x_1) \,\|\, s(x_{t_n}|x_{t_{n-1}})\right] - \mathbb{E}_{q^{\text{ref}}(x_{t_N}|x_0,x_1)}[\log s(x_1|x_{t_N})]\Big], \tag{6}$$

where $r_{\pi|x_0} \in \mathcal{M}(\mathcal{X}^{N+1})$ and each markov transition $r_{\pi|x_0}(x_{t_n}|x_{t_{n-1}}) \in \mathcal{P}(\mathcal{X})$, see equation 3.

In addition, one can create a family of independent subproblems like equation 6 indexed by *condition* x_0 to recover r_π as the family of $r_{\pi|x_0}$. Such an approach is known as *conditional bridge matching* Zhou et al. (2024); Igashov et al. (2024):

$$r_\pi = \arg\min_s \mathbb{E}_{\pi(x_1,x_0)}\Big[\sum_{n=1}^{N} \mathbb{E}_{q^{\text{ref}}(x_{t_{n-1}}|x_0,x_1)} \mathsf{KL}\left[q^{\text{ref}}(x_{t_n}|x_{t_{n-1}}, x_1) \,\|\, s(x_{t_n}|x_{t_{n-1}}, x_0)\right] - \mathbb{E}_{q^{\text{ref}}(x_{t_N}|x_0,x_1)}[\log s(x_1|x_{t_N}, x_0)]\Big], \tag{7}$$

where $r_\pi \in \mathcal{P}(\mathcal{X}^{N+2})$ at the optimum is the mixture of $r_{\pi|x_0}$ processes equation 6. The optimization objective equation 7 closely resembles the variational bound used in diffusion models Ho et al. (2020); Austin et al. (2021) and is usually solved with standard deep learning techniques Austin et al. (2021); Ksenofontov & Korotin (2025). Common practice is to parametrize a conditional Markov chain r_π using a neural network. Specifically, each transition is modeled as $r_\theta(x_{t_n}|x_{t_{n-1}}, x_0)$, where $x_{t_{n-1}}$ and x_0 serve as inputs and the network outputs a probability distribution over $\mathcal{X}$. That is $r_\theta : \mathcal{X} \times \mathcal{X} \to \Delta^{|\mathcal{X}|}$, where $\Delta^{|\mathcal{X}|}$ denotes the probability simplex over $\mathcal{X}$.

3 Related Work

We briefly review existing approaches to mutual information estimation, covering non-parametric, variational, and diffusion-based methods, and highlight the limitations of current techniques when applied to discrete state spaces.

Non-parametric Estimators. Typical non-parametric entropy and mutual information estimators rely on empirical PMF estimates, offering strong baselines for low-dimensional problems with small state spaces Pinchas et al. (2024). However, as task complexity increases, these approaches struggle to yield accurate estimates because many states occur infrequently or remain unobserved. For example, MI calculated through naïve PMF estimation is always upper-bounded by $\log(\text{sample size})$ Cover & Thomas (2012), a severe limitation in large state spaces.

Variational Estimators. Neural parametric approaches address this challenge by capturing the intrinsic structure of the data, filling the missing states implicitly. Currently, the most prominent class of neural MI estimators capable of handling discrete cases leverages variational bounds. These include MINE Belghazi et al. (2018), NWJ Nguyen et al. (2010), InfoNCE van den Oord et al. (2019), f-DIME Letizia et al. (2024), and similar methods. Such estimators, however, are prone to other theoretical limitations McAllester & Stratos (2020) and currently lag behind more elaborate approaches based on diffusion models and normalizing flows Franzese et al. (2024); Butakov et al. (2024a). On the other hand, these advanced methods target continuous distributions and do not apply to the discrete case — a gap we fill in our work.

Diffusion Based Estimators. A recent class of MI estimators is based on diffusion processes, including MINDE Franzese et al. (2024), which treats MI estimation as a *generative modeling* problem, and InfoBridge Kholkin et al. (2025), which approaches it as *domain transfer*. These methods leverage tools from stochastic calculus, such as the disintegration and Girsanov theorems, to express $\mathsf{KL}\left[\pi(x_0, x_1) \,\|\, \pi(x_0)\pi(x_1)\right]$ as a KL divergence between diffusion processes.

The MI estimation is the result of MSE integration over the stochastic process trajectory. Such methods are known to be particularly suitable for complex data and estimation of high mutual information. While a discrete-state variant of MINDE, Info-SEDD Foresti et al. (2025), has recently been proposed, diffusion-based estimators have so far been developed primarily for continuous state spaces, limiting their applicability in discrete settings.

4 DBMI. Mutual Information Estimator

In §4.1, we propose our novel DBMI method to estimate MI between discrete state space random variables, which is based on computing the KL divergence between reciprocal processes. Next, in §4.2 we explain the practical implementation of our proposed MI estimator.

4.1 Computing the MI though Reciprocal Processes

Consider the MI estimation for discrete random variables $X_0, X_1 \in \mathcal{P}(\mathcal{X})$ with corresponding joint distribution $\pi(x_0, x_1)$. To tackle this problem the key idea we choose the reference process q^{ref} and employ corresponding reciprocal processes:

$$r_\pi^{\text{joint}}(x_0, x_{\text{in}}, x_1) \stackrel{\text{def}}{=} q^{\text{ref}}(x_{\text{in}}|x_0, x_1)\pi(x_0, x_1), \tag{8}$$

$$r_\pi^{\text{ind}}(x_0, x_{\text{in}}, x_1) \stackrel{\text{def}}{=} q^{\text{ref}}(x_{\text{in}}|x_0, x_1)\pi(x_0)\pi(x_1). \tag{9}$$

We show that the Mutual Information, as the KL between $\pi(x_0)\pi(x_1)$ and $\pi(x_0, x_1)$ equation 1, is equal to the KL between reciprocal processes r_π^{joint} and r_π^{ind}, i.e., discrete bridge matching models solving domain transfer problem. The latter can be decomposed into the KL between Markov chains.

Algorithm 1: DBMI. Learning procedure.

Input: Distribution $\pi(x_0, x_1)$ accessible by samples, neural network parametrization r_θ of conditional Markov transitions r_π^{joint} and r_π^{ind}, batch size K, number of inner samples M
Output: Learned neural network r_θ approximating processes r_π^{joint} and r_π^{ind}
repeat
  Sample batch of pairs $\{x_0^k, x_{1,v=1}^k\}_{k=0}^K \sim \pi(x_0, x_1)$;
  Make a shuffle $\{x_{1,v=0}^k\}_{k=0}^K = \text{Permute}(\{x_{1,v=1}^k\}_{k=0}^K)$; // v=0 marks r_π^{ind}, v=1 marks r_π^{joint}
  Sample batch $\{n_k\}_{k=0}^K \sim U[1, N]$;
  Sample batch $\{x_{t_{n_k},v}^m\}_{m=0}^M \sim q^{\text{ref}}(x_{t_{n_k}} | x_0^k, x_{1,v}^k)$;
  $\mathcal{L}_\theta = \frac{1}{KM} \sum_{v\in\{0,1\}} \sum_{k=1}^K \sum_{m=1}^M \left[\mathbb{I}_{[n_k \neq N]} \cdot \mathsf{KL}\left[q^{\text{ref}}(x_{t_{n_k+1}} | x_{t_{n_k},v}^m, x_{1,v}^k) \,\|\, r_\theta(x_{t_{n_k+1}} | x_{t_{n_k},v}^m, x_0^k, v) \right] - \mathbb{I}_{[n_k = N]} \cdot \log r_\theta(x_1 | x_{t_N,v}^m, x_0^k, v) \right]$;
  Update θ using $\frac{\partial \mathcal{L}_\theta}{\partial \theta}$
until *converged*;

Algorithm 2: DBMI. Estimation procedure.

Input: Distribution $\pi(x_0, x_1)$ accessible by samples, neural network parametrization r_θ of conditional Markov transitions r_π^{joint} and r_π^{ind}, number of samples K, number of inner samples M
Output: Mutual information estimation $\widehat{\text{MI}}$
Sample batch of pairs $\{x_0^k, x_1^k\}_{k=1}^K \sim \pi(x_0, x_1)$;
Sample batch $\{n_k\}_{k=0}^K \sim U[1, N]$;
Sample $\{x_{t_{n_k}}^m\}_{m=0}^M \sim q^{\text{ref}}(x_{t_{n_k}} | x_0^k, x_1^k)$;
$\widehat{\text{MI}} \leftarrow \frac{1}{KM} \sum_{k=1}^K \sum_{m=1}^M \left[\mathsf{KL}\left[r_\theta(x_{t_{n_k+1}} | x_{t_{n_k}}^m, x_0^k, 1) \,\|\, r_\theta(x_{t_{n_k+1}} | x_{t_{n_k}}^m, x_0^k, 0) \right] \right]$;
return $\widehat{\text{MI}}$

Proposition 2 (Mutual Information Decomposition)**.** *Consider random variables $X_0, X_1 \in \mathcal{P}(\mathcal{X})$ and their joint distribution $\pi(x_0, x_1)$. Consider reciprocal processes r_π^{joint} and r_π^{ind} induced by distributions $\pi(x_0, x_1)$ and $\pi(x_1)\pi(x_0)$ respectively, as in equation 8 and equation 9. The the MI between the random variables X_0 and X_1 can be expressed as:*

$$I(X_0; X_1) = \sum_{n=1}^N \mathbb{E}_{r_\pi(x_0, x_{t_n})} \left[\mathsf{KL}\left[r_\pi^{\text{joint}}(x_{t_{n+1}} | x_{t_n}, x_0) \,\|\, r_\pi^{\text{ind}}(x_{t_{n+1}} | x_{t_n}, x_0) \right] \right], \tag{10}$$

where

$$r_\pi^{\text{joint}}(x_0, x_{\text{in}}, x_1) = \pi(x_0) \prod_{n=1}^N r_\pi^{\text{joint}}(x_{t_{n+1}} | x_{t_n}, x_0), \tag{11}$$

$$r_\pi^{\text{ind}}(x_0, x_{\text{in}}, x_1) = \pi(x_0) \prod_{n=1}^N r_\pi^{\text{ind}}(x_{t_{n+1}} | x_{t_n}, x_0). \tag{12}$$

The r_π^{joint} and r_π^{ind} are the conditional Markov transitions from the corresponding representations equation 4 of the reciprocal processes r_π^{joint} and r_π^{ind}.

See Appendix A.2 for proof. Once the Markov chain transitions r_π^{joint} and r_π^{ind} are known, our Proposition 2 provides a straightforward way to estimate the mutual information between random variables X_0 and X_1 by evaluating the KL divergence between r_π^{joint} and r_π^{ind} at trajectory points $x_{t_n}, x_0 \sim r_\pi^{\text{joint}}(x_0, x_{t_n})$.

4.2 Practical Algorithm

We developed a practical algorithm to estimate the mutual information based on Proposition 2 and call it **DBMI**. Our approach consists of two distinct stages: first, we learn the Markov transitions via bridge matching, (see §2.5 and Algorithm 1); second, we use these learned transitions to compute the mutual information (see Algorithm 2).

The reciprocal processes r_π^{joint} and r_π^{ind} can be recovered by the Bridge Matching for discrete state spaces procedure, see §2.5. We have to solve optimization problem equation 7 by parametrizing r_π^{joint} and r_π^{ind} with neural networks $r_\pi^{\text{joint},\phi}$ and $r_\pi^{\text{ind},\psi}$, respectively, and applying Gradient Descent on Monte Carlo approximation of equation 7.

The MI estimate in equation 10 requires sampling from the reciprocal process $r_\pi^{\text{joint}}(x_0, x_{t_n})$ at times 0 and t_n, which is straightforward in a *simulation-free* manner since:

$$r_\pi^{\text{joint}}(x_0, x_{t_n}) = \mathbb{E}_{\pi(x_1)}[r_\pi^{\text{joint}}(x_0, x_{t_n}|x_1)] = \mathbb{E}_{\pi(x_1)}[q^{\text{ref}}(x_{t_n}|x_0, x_1)\pi(x_0|x_1)]. \tag{13}$$

Therefore, a sample from it can be obtained by sampling from $\pi(x_0, x_1)$ and $q^{\text{ref}}(x_{t_n}|x_0, x_1)$.

Amortization of Learning Procedure. In practice, to ammortize the optimization procedure, we replace two separate neural networks that approximate the conditional Markov transitions $r_\pi^{\text{joint}}(x_{t_{n+1}}|x_{t_n}, x_0)$ and $r_\pi^{\text{ind}}(x_{t_{n+1}}|x_{t_n}, x_0)$ with a single neural network that incorporates an additional binary input. Specifically, we introduce a binary input $v \in \{0, 1\}$ to unify the Markov chain transitions approximations in the following way: $r_\theta(\cdot, 1) \approx r_\pi^{\text{joint}}(\cdot)$ and $r_\theta(\cdot, 0) \approx r_\pi^{\text{ind}}(\cdot)$. The introduction of such additional input is a common technique (see, for e.g., Franzese et al. (2024)).

Markov Transitions Parametrization. Note that the size of the state space grows exponentially as $|\mathcal{X}| = |\mathbb{S}|^D = S^D$, where S is the number of categories per variable and D is the total number of variables. Consequently, a neural network used to parametrize the transition kernel $r_\theta(x_{t_n}|x_{t_{n-1}})$ would, in principle, need to output S^D probabilities, which quickly becomes computationally prohibitive. Because directly modeling such transition distributions is impractical, we adopt a common strategy from discrete generative modeling Austin et al. (2021); Sahoo et al. (2024) and impose a factorized structure over dimensions on the probabilities:

$$r_\theta(x_{t_n}|x_{t_{n-1}}, x_0) \approx \prod_{d=1}^{D} r_\theta(x_{t_n}^d|x_{t_{n-1}}, x_0). \tag{14}$$

In this case, for each $x_{t_{n-1}}$, we predict a row-stochastic $D \times S$ matrix of probabilities $r_\theta(x_{t_n}^d \mid x_{t_{n-1}}, x_0)$. Rather than using separate models for each of the $N + 1$ transitions, we train a single neural network with an additional timestep input. The network takes inputs of the form $S^D \times S^D \times [0, 1]$ and outputs $\mathbb{R}^{S \times D}$. We also apply posterior sampling (see Appendix B.1 for details).

Reference Process. Reciprocal processes that we work with depend on the reference process q^{ref}, see §2.2. We consider the case where we do not have any prior information on data so we choose the simplest reference process with *uniform* categorical noise, i.e., random change of category, Austin et al. (2021). We start from one-dimensional case. Define a process where the state stays in the current category $x_{t_{n-1}}$ with high probability, while the remaining probability is distributed uniformly among all other categories. We call this process the *uniform*. Formally, as a homogeneous Markov process, it is defined by the following probabilities:

$$q^{\text{unif}}(x_{t_n}|x_{t_{n-1}}) = \begin{cases} 1 - \alpha, & x_{t_n} = x_{t_{n-1}}, \\ \dfrac{\alpha}{S-1}, & x_{t_n} \neq x_{t_{n-1}}, \end{cases} \tag{15}$$

where $\alpha \in [0, 1]$ is the *stochasticity parameter* that controls the probability of transitioning into another state. Then, to construct such a process for $D > 1$ one has to combine several such independent one-dimensional processes.

For such case bridge $q^{\text{ref}}(x_{t_n}|x_0, x_1)$ and posterior $q^{\text{ref}}(x_{t_n}|x_{t_{n-1}}, x_1)$ is known analytically Austin et al. (2021) and can be sampled very easily for all t_n. While alternative reference processes, such as those incorporating intercategory dependencies Austin et al. (2021), are possible, we adopt q^{unif} due to its simplicity and robustness, as it does not rely on any additional assumptions.

5 Experiments

We evaluate our method on two benchmarks, where known ground truth MI values allow us to assess the performance of our method. To cover low-dimensional cases, we construct joint probability distributions with a manageable number of categories and then stack these distributions upon each other (see §5.1). Next, to evaluate the capability of our method to estimate mutual information between complex, high-dimensional random variables, we propose a procedure for embedding discrete state

Dim	MI_{true}	DBMI (**ours**)	f-DIME-KL	f-DIME-H	f-DIME-G	MINE	InfoNCE
$D = 2$	3.38	**3.40**	2.97	3.10	3.17	3.23	3.31
$D = 5$	8.50	**8.67**	3.43	6.89	7.08	6.69	5.99
$D = 10$	16.80	**16.90**	5.04	7.01	9.23	3.28	6.22
$D = 25$	42.36	**42.10**	4.35	4.69	9.33	9.83	6.22
$D = 50$	85.47	**79.30**	3.82	1.85	9.52	8.75	6.19

Table 1: Results on the low-dimensional benchmark for varying dimension D (see Dim column) and fixed number of categories $S = 10$. MI_{true} denotes ground-truth MI; other columns show estimates from each method. The closer to MI_{true} the better. The closest to GT estimates are marked in **bold**.

NumCat	MI_{true}	DBMI (**ours**)	f-DIME-KL	f-DIME-H	f-DIME-G	MINE	InfoNCE
$S = 2$	3.64	3.91	3.08	3.37	**3.84**	3.28	2.909
$S = 5$	10.35	**10.21**	4.65	7.44	6.97	6.82	6.13
$S = 10$	16.80	**16.90**	5.04	7.01	9.23	3.28	6.21
$S = 25$	25.86	**25.33**	4.42	-9.65	9.97	9.8	6.21
$S = 50$	32.73	**31.03**	3.57	7.11	8.65	9.71	6.18

Table 2: Results on the low-dimensional benchmark for varying number of categories S (see Dim column) and fixed dimension $D = 10$. MI_{true} denotes ground-truth MI; other columns show estimates from each method. The closer to MI_{true} the better. The closest to GT estimates are marked in **bold**.

space variables into structured image data and estimating MI between them (see §5.2). We compare ours DBMI to other popular neural MI estimators that are suitable for working with discrete random variables: MINE Belghazi et al. (2018), InfoNCE van den Oord et al. (2019), NWJ Nguyen et al. (2010) and f-DIME Letizia et al. (2024) with KL, Hellinger (H), and GAN (G) divergences, using the same deranged architecture across all divergences.

5.1 Low Dimensional Benchmark

We validation our method on distributions with known mutual information, constructed in the following way. Consider a random vector $(X_0, X_1) = (X_0^1, X_1^1, \ldots, X_0^D, X_1^D)$ taking values in $\mathbb{S}^{2D}$. We factorize this random vector pdf $\pi(x_0, x_1)$ across dimensions:

$$\pi(x_0, x_1) = \prod_{d=1}^{D} \pi_d(x_0^d, x_1^d),$$

where each bivariate factor $\pi_d(x_0^d, x_1^d)$ is constructed by first drawing x^d uniformly over $\mathbb{S}$ and then defining the conditional distribution via a randomly generated stochastic transition matrix $\Pi(x_1^d|x_0^d)$ with controlled level of dependence between X_0^d and X_1^d. Owing to the factorization across dimensions, the total mutual information decomposes as $I(X_0; X_1) = \sum_{d=1}^{D} I(X_0^d; X_1^d)$.

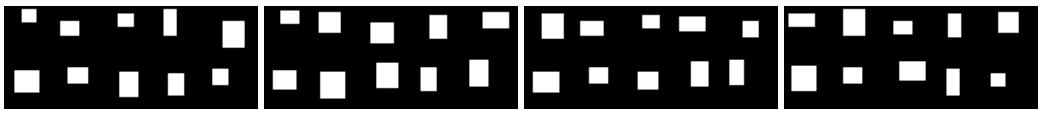
We compare our DBMI and other mentioned MI estimators on low-dimensional benchmark, by increasing both $S = |\mathbb{S}|$ and D as shown in Table 1 and Table 2. All the methods are trained using 10^4 train samples and 10^4 test samples. Our method employs a Diffusion Transformer (DiT)-based architecture trained for 150 epochs, $\alpha = 10^{-4}$ for q^{ref} equation 15, $M = 1$ for training, Algoritm 1, and $M = 10$ for MI estimation, Algoritm 1. For further details, see Appendix B.3.

The results demonstrate that DBMI consistently outperforms competing methods. DBMI holds precise MI estimation with the growth of dimension, see Table 1, and the growth of number of categories, see Table 2, while other neural estimators start to fail very early.

5.2 Image Benchmark

To evaluate scalability, we propose the first image-based benchmark for discrete data. Building on the idea that MI is invariant w.r.t. bijective mappings Butakov et al. (2024b). We construct low-dimensional latent random variables and corresponding bijective mapping to the *image* space. This allows for precise MI control within a high-dimensional, complex image state space.

Specifically, we have four *latent variables* $(X_0^1, X_0^2, X_0^3, X_0^4)$ that specify the pixel border coordinates of the single rectangle. We sample each coordinate as $X_0^i \sim U(0, \ldots, V)$ and enforce a minimum side length $V^{\min}$ to avoid degenerate rectangles. The target X_1 is obtained by passing



(a) $x_0 \sim \pi(x_0)$ (b) $x_1 \sim \pi(x_1)$ (c) $x_1 \sim r_\theta(\cdot, 0)$ (d) $x_1 \sim r_\theta(\cdot, 1)$

Figure 1: Qualitative samples, presented in 5×2 image grids, generated using image benchmark and DBMI, i.e., $r_\theta(\cdot)$, at resolution 32×32.

MI_{true}	DBMI (**ours**)	f-DIME-KL	f-DIME-H	f-DIME-G	MINE	NWJ
1	0.504	0.668	0.687	**0.701**	0.0	0.0
2	**1.426**	1.158	1.264	1.330	0.0	0.0
4	**3.923**	2.211	2.676	2.719	1.84	0.685
6	**6.301**	2.957	3.678	3.385	2.784	1.789
8	**8.447**	4.239	4.919	4.886	3.776	2.554

Table 3: Results for the high-dimensional image benchmark with 32×32 images across all estimators. MI_{true} denotes the ground-truth mutual information. All the other columns correspond to MI estimation results for a particular method. The closer to MI_{true} the better. The closest to GT estimates are highlighted in **bold**.

each latent coordinate X_0^i through a symmetric noisy channel (Lee & Rhee, 2024, §4.5) independently, which yields following mutual information between images $\mathsf{I}(X_0; X_1) = \sum_{i=1}^{4} \mathsf{I}(X_0^i; X_1^i)$.

In our experiments, we set $V = 10$, $V^{\min} = 10$ and render 32×32 binary images ($S = 2$), where 1 indicates the rectangle and 0 the background. We train all methods on 10^5 samples for mutual information values in $\{1, 2, 4, 6, 8\}$ and evaluate on 10^4 validation samples. Specifically, we train our method for 20 epochs and use $\alpha = 10^{-2}$ for q^{ref} in equation 15, $M = 1$ for training, Algoritm 1, and $M = 10$ for MI estimation, Algoritm 1. Additional experiments with 16×16 images are presented in Appendix C.

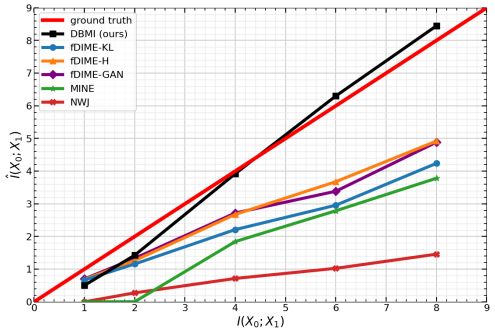


Figure 2: Comparison of estimated mutual information $\hat{\mathsf{I}}(X_0; X_1)$ across methods against the ground-truth $\mathsf{I}(X_0; X_1)$ (red) on a high-dimensional image benchmark with size 32×32.

Results in Figure 2 and Table 3 show that DBMI scales to high-dimensional images and accurately estimates mutual information in regimes where competing methods fail. In addition to accurate MI estimation, our method learns to generate perfect samples from marginal x_1, see Figure 1.

6 Discussion

Potential Impact. Our contributions include the development of a novel Mutual Information estimator for the discrete state space random variables grounded in the reciprocal processes and the bridge matching theory. The proposed algorithm, DBMI, achieves superior performance over widely used mutual information estimators applicable for discrete state spaces, including MINE Belghazi et al. (2018), InfoNCE van den Oord et al. (2019), and f-DIME Letizia et al. (2024), across both low-dimensional benchmark and challenging high-dimensional image-based benchmark.

Limitations. The factorization of transition probabilities described in §4.2 introduces some bias towards the MI estimation. However, generally such bias is considered negligible, see (Campbell et al., 2022, §4.5), and reduces to zero with $N \to \infty$. Moreover, our training and estimation are simulation-free, so we can choose N arbitrarily large.

Future Work. This exploration of mutual information estimation for discrete state space variables opens new directions in information theory, particularly for **mixed state spaces**—combinations of discrete and continuous variables. Notably, bridge matching has already shown promise in mixed-type data generation, such as tabular data Guzmán-Cordero et al. (2025).

Acknowledgements. The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4F0002 and the agreement №139-10-2025-033.

REFERENCES

Amir Asadi, Emmanuel Abbe, and Sergio Verdu. Chaining mutual information and tightening generalization bounds. 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/8d7628dd7a710c8638dbd22d4421ee46-Paper.pdf.

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.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 531–540. PMLR, 07 2018. URL https://proceedings.mlr.press/v80/belghazi18a.html.

Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. The schrödinger bridge between gaussian measures has a closed form. In *International Conference on Artificial Intelligence and Statistics*, pp. 5802–5833. PMLR, 2023.

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*, 2024a. URL https://openreview.net/forum?id=JiQXsLvDls.

Ivan Butakov, Alexander Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, Alexey Frolov, and Kirill Andreev. Information bottleneck analysis of deep neural networks via lossy compression. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=huGECz8dPp.

Ivan Butakov, Alexander Semenenko, Alexander Tolmachev, Andrey Gladkov, Marina Munkhoeva, and Alexey Frolov. Efficient distribution matching of representations via noise-injected deep infomax. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mAmCdASmJ5.

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. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/8248a99e81e752cb9b41da3fc43fbe7f-Paper.pdf.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 2012. ISBN 9781118585771.

Maxime Darrin, Philippe Formont, Jackie Cheung, and Pablo Piantanida. COSMIC: Mutual information for task-agnostic summarization evaluation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12696–12717, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.686. URL https://aclanthology.org/2024.acl-long.686/.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

B. Duong and T. Nguyen. Conditional independence testing via latent representation learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 121–130, Los Alamitos, CA, USA, dec 2022. IEEE Computer Society. doi: 10.1109/ICDM54844.2022.00022. URL https://doi.ieeecomputersociety.org/10.1109/ICDM54844.2022.00022.

Alberto Foresti, Giulio Franzese, and Pietro Michiardi. Info-sedd: Continuous time markov chains as scalable information metrics estimators. *arXiv preprint arXiv:2502.19183*, 2025.

Giulio Franzese, Mustapha BOUNOUA, and Pietro Michiardi. MINDE: Mutual information neural diffusion estimation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0kWd8SJq8d.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Ziv Goldfeld, Ewout Van Den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2299–2308. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/goldfeld19a.html.

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In *Forty-first International Conference on Machine Learning*, 2024a.

Nikita Gushchin, Daniil Selikhanovych, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Adversarial schr\" odinger bridge matching. *arXiv preprint arXiv:2405.14449*, 2024b.

Andrés Guzmán-Cordero, Floor Eijkelboom, and Jan-Willem van de Meent. Exponential family variational flow matching for tabular data generation. In *Forty-second International Conference on Machine Learning*, 2025.

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. URL https://openreview.net/forum?id=Bklr3j0cKX.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ilia Igashov, Arne Schneuing, Marwin Segler, Michael M Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with markov bridges. In *The Twelfth International Conference on Learning Representations*, 2024.

Sergei Kholkin, Ivan Butakov, Evgeny Burnaev, Nikita Gushchin, and Alexander Korotin. Infobridge: Mutual information estimation via bridge matching. *arXiv preprint arXiv:2502.01383*, 2025.

Jun Hyeong Kim, Seonghwan Kim, Seokhyun Moon, Hyeongwoo Kim, Jeheon Woo, and Woo Youn Kim. Discrete diffusion schrödinger bridge matching for graph transformation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Grigoriy Ksenofontov and Alexander Korotin. Categorical schrödinger bridge matching. In *Forty-second International Conference on Machine Learning*, 2025.

Kyungeun Lee and Wonjong Rhee. A benchmark suite for evaluating neural mutual information estimators on unstructured datasets. *Advances in Neural Information Processing Systems*, 37: 46319–46338, 2024.

Christian Léonard, Sylvie Rœlly, and Jean-Claude Zambrini. Reciprocal processes. a measure-theoretical point of view. *Probability Surveys*, 11:237–269, 2014.

Nunzio Alexandro Letizia, Nicola Novello, and Andrea M Tonello. Mutual information estimation via f-divergence and data derangements. *Advances in Neural Information Processing Systems*, 37:105114–105150, 2024.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schr\" odinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 875–884. PMLR, 08 2020. URL https://proceedings.mlr.press/v108/mcallester20a.html.

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

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Zbigniew Palmowski and Tomasz Rolski. A technique for exponential change of measure for markov processes. *Bernoulli*, pp. 767–785, 2002.

Hanchuan Peng, Fuhui Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. doi: 10.1109/TPAMI.2005.159.

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

Y. Polyanskiy and Y. Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2024. ISBN 9781108832908. URL https://books.google.ru/books?id=CySo0AEACAAJ.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. In *Uncertainty in Artificial Intelligence*, pp. 1985–1995. PMLR, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Francisco Vargas, Andrius Ovsianas, David Fernandes, Mark Girolami, Neil D Lawrence, and Nikolas Nüsken. Bayesian learning via neural schrödinger–föllmer flows. *Statistics and Computing*, 33(1):3, 2023.

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 de novo peptide sequencing in proteomics against data biases. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 1811–1828. Curran Associates, Inc., 2024. doi: 10.52202/079017-0057. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/034b4a2860d4d170ce663584bc78cb32-Paper-Conference.pdf.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024.

A PROOFS

A.1 CONDITIONAL RECIPROCAL PROCESS AS MARKOV CHAIN

Proof. Consider conditional reciprocal process r_π:

$$r_\pi(x_{\text{in}}, x_1|x_0) = r_\pi(x_{t_1:t_{N+1}}|x_0) \tag{16}$$

Let us consider part of the reciprocal process $r(x_{t_1:t_k}|x_0)$, marginalize it by future timesteps $t_{k+1} : t_{N+1}$ and apply the reciprocal process definition equation 2:

$$r(x_{t_1:t_k}|x_0) = \sum_{x_{t_{k+1}:x_{t_{N+1}}}} r(x_{t_1:t_{N+1}}|x_0) = \sum_{x_{t_{k+1}:x_{t_{N+1}}}} q^{\text{ref}}(x_{t_1:t_N}|x_0)\pi(x_1|x_0) =$$
$$\sum_{x_{t_{k+1}:x_{t_{N+1}}}} q^{\text{ref}}(x_{t_1:t_N}|x_0)\frac{q^{\text{ref}}(x_1|x_0)}{q^{\text{ref}}(x_1|x_0)}\pi(x_1|x_0) = \sum_{x_{t_{k+1}:x_{t_{N+1}}}} q^{\text{ref}}(x_{t_1:t_{N+1}}|x_0)\frac{\pi(x_1|x_0)}{q^{\text{ref}}(x_1|x_0)} \tag{17}$$

Then recap that reference process q^{ref} is Markov:

$$q^{\text{ref}}(x_{t_1:t_{N+1}}|x_0) = \left[\prod_{i=1}^{k} q^{\text{ref}}(x_{t_i}|x_{t_{i-1}}, x_0)\right] \cdot q^{\text{ref}}(x_{t_{k+1}:t_{N+1}}|x_k, x_0) \tag{18}$$

Apply Markov chain form equation 18 for reciprocal process equation 18:

$$r(x_{t_1:t_k}|x_0) = \left[\prod_{i=1}^{k} q^{\text{ref}}(x_{t_i}|x_{t_{i-1}}, x_0)\right] \cdot \sum_{x_{t_{k+1}:x_{t_{N+1}}}} \frac{\pi(x_1|x_0)}{q^{\text{ref}}(x_1|x_0)} q^{\text{ref}}(x_{t_{k+1}:t_{N+1}}|x_k, x_0) \tag{19}$$

Then it is easy to see that last line term depends only on x_t:

$$h(x_k) = \sum_{x_{t_{k+1}:x_{t_{N+1}}}} \frac{\pi(x_1|x_0)}{q^{\text{ref}}(x_1|x_0)} q^{\text{ref}}(x_{t_{k+1}:t_{N+1}}|x_k, x_0) \tag{20}$$

Finally let us take a look at the ratio between the parts of reciprocal process r_π and apply equation 19:

$$r_\pi(x_{t_k}|x_{t_1:t_{k-1}}, x_0) = \frac{r_\pi(x_{t_1:t_k}|x_0)}{r_\pi(x_{t_1:t_{k-1}}|x_0)} = \frac{\left[\prod_{i=1}^{k} q^{\text{ref}}(x_{t_i}|x_{t_{i-1}}, x_0)\right]}{\left[\prod_{i=1}^{k-1} q^{\text{ref}}(x_{t_i}|x_{t_{i-1}}, x_0)\right]} \frac{h(x_k)}{h(x_{k-1})} =$$
$$q^{\text{ref}}(x_{t_k}|x_{t_{k-1}}, x_0)\frac{h(x_k)}{h(x_{k-1})} = r_\pi(x_{t_k}|x_{t_{k-1}}, x_0) \tag{21}$$

Since $r_\pi(x_{t_k}|x_{t_1:t_{k-1}}, x_0) = r_\pi(x_{t_k}|x_{t_{k-1}}, x_0)$ it is evident that $r_\pi(x_{\text{in}}, x_1|x_0)$ is a Markov chain. □

A.2 MUTUAL INFORMATION DECOMPOSITION PROOF.

Proof. Using the chain rule for KL divergence between stochastic processes r_π^{joint} and r_π^{ind}:

$$\mathsf{KL}\left[r_\pi^{\text{joint}}(\cdot) \,\|\, r_\pi^{\text{ind}}(\cdot)\right] = \mathsf{KL}\left[\pi(x_0) \,\|\, \pi(x_0)\right] + \mathbb{E}_{\pi(x_0)}\left[\mathsf{KL}\left[r_{\pi|x_0}^{\text{joint}}(\cdot) \,\|\, r_{\pi|x_0}^{\text{ind}}(\cdot)\right]\right] \tag{22}$$

Note that r_π^{joint} and r_π^{ind} share the same $\pi(x_0)$ marginal by construction, see equation 9 and equation 8, so the first KL term is zero. Applying the chain rule for KL divergence between stochastic

MI_{true}	DBMI (**ours**)	f-DIME-KL	f-DIME-H	f-DIME-G
1	**0.845**	0.729	0.804	0.784
2	1.932	1.852	**1.999**	2.104
3	**2.928**	2.439	2.796	2.74
4	4.124	3.646	3.829	**3.902**
5	**5.103**	4.284	4.881	4.834
6	6.41	5.442	5.655	**6.01**

Table 4: Results for the high-dimensional image benchmark with 16×16 images across all estimators. MI_{true} denotes the ground-truth mutual information. All the other columns correspond to MI estimation results for a particular method. The closer to MI_{true} the better. Best estimates are highlighted in **bold**.

processes once more, and decomposing at the initial and terminal times of the reciprocal process, we obtain:

$$\mathsf{KL}\left[r_\pi^{\text{joint}}(\cdot)\,\|\,r_\pi^{\text{ind}}(\cdot)\right] = \mathsf{KL}\left[\pi(x_0,x_1)\,\|\,\pi(x_0)\pi(x_1)\right] + \mathbb{E}_{\pi(x_0,x_1)}[\mathsf{KL}\left[r_\pi^{\text{joint}}(\cdot|x_1,x_0)\,\|\,r_\pi^{\text{ind}}(\cdot|x_1,x_0)\right]] \quad (23)$$

Recap that both $r_\pi^{\text{joint}}(\cdot|x_1,x_0)$ and $r_\pi^{\text{ind}}(\cdot|x_1,x_0)$ are $q^{\text{ref}}(\cdot|x_0,x_1)$, see equation 2. Therefore, $\mathsf{KL}\left[r_\pi^{\text{joint}}(\cdot|x_1,x_0)\,\|\,r_\pi^{\text{ind}}(\cdot|x_1,x_0)\right] = 0$. Then by combining the equation 22 and equation 23 the following holds:

$$\mathsf{KL}\left[r_\pi^{\text{joint}}(\cdot)\,\|\,r_\pi^{\text{ind}}(\cdot)\right] = \mathsf{KL}\left[\pi(x_0,x_1)\,\|\,\pi(x_0)\pi(x_1)\right] = \mathbb{E}_{\pi(x_0)}\left[\mathsf{KL}\left[r_{\pi|x_0}^{\text{joint}}(\cdot)\,\|\,r_{\pi|x_0}^{\text{ind}}(\cdot)\right]\right] = \mathsf{I}(X_0;X_1) \quad (24)$$

Finally by presenting the conditional reciprocal processes $r_{\pi|x_0}^{\text{joint}}$ and $r_{\pi|x_0}^{\text{ind}}$ as the Markov chains:

$$I(X_0;X_1) = \mathbb{E}_{\pi(x_0)}\left[\mathsf{KL}\left[r_{\pi|x_0}^{\text{joint}}(\cdot)\,\|\,r_{\pi|x_0}^{\text{ind}}(\cdot)\right]\right] = \mathbb{E}_{\pi(x_0)}\Big[\sum_{n=1}^{N}\mathbb{E}_{x_{t_n}}\,\mathsf{KL}\left[r_\pi^{\text{joint}}(x_{t_{n+1}}|x_{t_n},x_0)\,\|\,r_\pi^{\text{ind}}(x_{t_{n+1}}|x_{t_n},x_0)\right]\Big]$$

□

B Experimental Details

B.1 Posterior sampling

In line with the standard practice in diffusion models (see, e.g. Ho et al. (2020)), we parameterize the transition probabilities $r_\theta(x_{t_n}|x_{t_{n-1}},x_0)$ using a posterior sampling scheme:

$$r_\theta(x_{t_n}\mid x_{t_{n-1}},x_0) = \mathbb{E}_{\tilde{r}_\theta(\tilde{x}_1|x_{t_{n-1}},x_0)}\big[q^{\text{ref}}(x_{t_n}\mid x_{t_{n-1}},\tilde{x}_1)\big],$$

where $\tilde{r}_\theta(\tilde{x}_1\mid x_{t_{n-1}},x_0)$ is a learnable distribution. This parameterization assumes that sampling x_{t_n} given $x_{t_{n-1}}$ and x_0 proceeds in two stages:

1. Obtain probabilities of an endpoint $\tilde{r}_\theta(\tilde{x}_1\mid x_{t_{n-1}},x_0)$;
2. Compute expectation of the reference process $\mathbb{E}_{\tilde{x}_1}[q^{\text{ref}}(x_{t_n}\mid x_{t_{n-1}},\tilde{x}_1)]$ distribution and sample the next state x_{t_n}.

B.2 General Other Methods Details.

MINE, NWJ, InfoNCE. We use a nearly identical experimental framework to assess each approach within this category. To approximate the critic function in experiments with synthetic data, we adopt the NN architecture from Butakov et al. (2024a), which we also report in Table 5 ("Critic NN").

Table 5: The NN architectures for variational methods (MINE, NWJ, InfoNCE, f-DIME) used to conduct the tests in §5.2.

NN		Architecture
Critic NN, low-dimensional data	$\times 1$:	Dense($2 \cdot$ dim, 256), LeakyReLU(0.01)
	$\times 1$:	Dense(256, 256), LeakyReLU(0.01)
	$\times 1$:	Dense(128, 1)
Critic NN, 16×16 (32×32) images	$\times 1$:	[Conv2d(1, 16, ks=3), MaxPool2d(2), LeakyReLU(0.01)]$^{\times 2 \text{ in parallel}}$
	$\times 1(2)$:	[Conv2d(16, 16, ks=3), MaxPool2d(2), LeakyReLU(0.01)]$^{\times 2 \text{ in parallel}}$
	$\times 1$:	Dense(256, 128), LeakyReLU(0.01)
	$\times 1$:	Dense(128, 128), LeakyReLU(0.01)
	$\times 1$:	Dense(128, 1)

f-DIME For all the f-DIME-H, f-DIME-KL, f-DIME-G we take the official implementation:

`https://github.com/nicolaNovello/fDIME`

Info-SEDD. We attempted to reproduce the Info-SEDD method Foresti et al. (2025) following the publicly available description, but were unable to obtain competitive performance under our experimental settings.

B.3 Low Dimensional Benchmark Details

Conditional Probability Matrix Generation The conditional probability matrix $\Pi(x_1^d|x_0^d)$ for each dimension is generated as follows:

1. Create exponential kernel matrix using a Gaussian similarity measure:

$$K_{ij} := \exp\left(-\frac{(i-j)^2}{2\sigma^2}\right), \quad i,j \in \{0,\dots,S-1\}$$

   where σ controls the bandwidth of the kernel (we use $\sigma = 0.5$).

2. Multiply K element-wise by a random matrix:

$$P_0 := K \odot \text{Uniform}(0,1) + \epsilon,$$

   with $\epsilon = 10^{-12}$ for numerical stability.

3. Apply normalization to obtain a stochastic matrix: $\Pi(x_1^d|x_0^d) := Normalize(P_0)$.

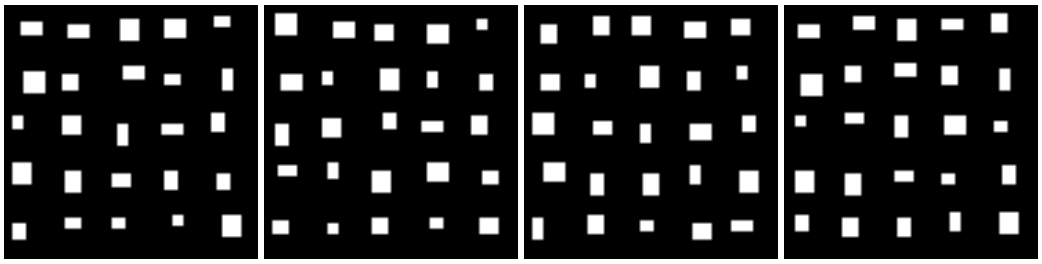
DBMI We employ a DiT-based architecture with $\approx 300k$ parameters and following specifications:

Architecture: Embedding layer $\to$ 4 Transformer blocks with rotary positional embeddings $\to$ linear projection. Each block has 4 attention heads, hidden dimension 32, conditioning dimension 32. The approximate number of parameters in a neural network is $\approx 312k$.

Hyperparameters: Batch size 512, learning rate 3×10^{-4}, $\alpha = 10^{-4}$, $M = 10$ (number of inner samples in Algorithm 2), training for 150 epochs via KL loss. Nvidia A100 was used to train the models. In any setup, each run (one seed) took no longer than one GPU-hour to be completed.

MINE, InfoNCE. The networks were trained via Adam optimizer with a learning rate 10^{-3}, a batch size 512. For averaging, we used 5 different seeds. Nvidia A100 was used to train the models. In any setup, each run (one seed) took no longer than one GPU-hour to be completed.

f-DIME The neural network is MLP with Embedding layer to handle discrete data with $\approx 600k$ parameters. We follow the default parameters provided by the authors as much as possible: learning rate $2 \cdot 10^{-4}$, batch size 128, number of gradient steps 50k. Nvidia A100 was used to train the models. In any setup, each run (one seed) took no longer than one GPU-hour to be completed.



(a) $x_0 \sim \pi(x_0)$ (b) $x_1 \sim \pi(x_1)$ (c) $x_1 \sim r_\theta(\cdot, 0)$ (d) $x_1 \sim r_\theta(\cdot, 1)$

Figure 3: Qualitative samples, presented in 5×2 image grids, generated using image benchmark and DBMI, i.e., $r_\theta(\cdot)$, at resolution 16×16.

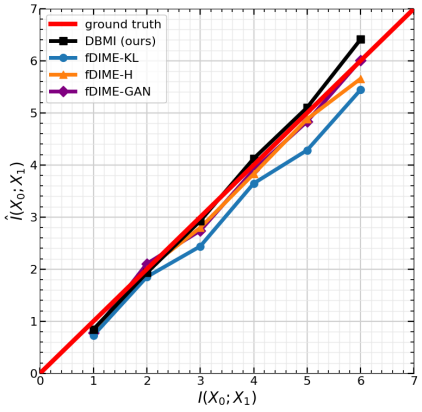


Figure 4: Comparison of estimated mutual information $\hat{\mathsf{I}}(X_0; X_1)$ across methods against the ground-truth $\mathsf{I}(X_0; X_1)$ (red) on a high-dimensional image benchmark with size 16×16.

B.4 Image Based Benchmark

DBMI We employ a UNet architecture with $\approx 6.8M$ parameters from `diffusers` library:

`https://huggingface.co/docs/diffusers/v0.21.0/en/api/models/unet2d`

Architecture: 2-channel input $\rightarrow$ Downsampling block (2 ResNet layers, 128 channels) $\rightarrow$ Downsampling block with spatial self-attention (2 ResNet layers, 128 channels) $\rightarrow$ Upsampling block with spatial self-attention (2 ResNet layers, 128 channels) $\rightarrow$ Upsampling block (2 ResNet layers, 128 channels) $\rightarrow$ 2-channel output The conditioning is done via concatenation of $x_{t_{n-1}}$ and x_1 by channel dimension.

Hyperparameters: batch size 128, learning rate 3×10^{-4}, $\alpha = 10^{-2}$, and $M = 10$ inner samples in Algorithm 2. We train for 30 epochs using the KL objective, with an additional cross-entropy term weighted by 10^{-3} (see (Austin et al., 2021, §3.4)).

Nvidia A100 was used to train the models. In any setup, each run (one seed) took no longer than one GPU-hour to be completed.

MINE, NWJ. The networks were trained via Adam optimizer with a learning rate 10^{-3}, a batch size 512. For averaging, we used 5 different seeds. Nvidia A100 was used to train the models. In any setup, each run (one seed) took no longer than two GPU-hours to be completed.

f-DIME The neural network architectures for 32×32 and 16×16 image resolution setup are presented in Table 5, . We follow the default parameters provided by the authors as much as possible: learning rate $2 \cdot 10^{-4}$, batch size 128, number of gradient steps 50k. Nvidia A100 was used to train the models. In any setup, each run (one seed) took no longer than one GPU-hour to be completed.

C Additional Experiments

Image Benchmark 16×16**.** We test our DBMI and f-DIME on image benchmark §5.2 but with images being 16×16. In that case latent variables $\{X_0^i, X_1^i\}$ are constructed in the same way, but alphabet of discrete random variable is smaller, i.e., $V = 5$. As well as minimum side length for rectangle $V^{\min} = 5$. We vary ground truth mutual information: $MI_{\text{true}} = \{1, 2, 3, 4, 5, 6\}$ and one can see the experiment results in Table 4, Figure 4 and Figure 3.

It is evident that both our DBMI and f-DIME do solve the problem.