

Discrete Markov Bridge

Hengli Li^{1,2,3}

lihengli@stu.pku.edu.cn

Yuxuan Wang^{2,3}

wangyuxuan1@bigai.ai

Song-Chun Zhu^{1,2,3,4}

s.c.zhu@pku.edu.cn

Ying Nian Wu⁵✉

ywu@stat.ucla.edu

Zilong Zheng^{2,3}✉

zlzheng@bigai.ai

¹ Institute of Artificial Intelligence, Peking University

² NLCo Lab, Beijing Institute for General Artificial Intelligence

³ State Key Laboratory of General Artificial Intelligence

⁴ Department of Automation, Tsinghua University

⁵ University of California, Los Angeles

Abstract

Discrete diffusion has recently emerged as a promising paradigm in discrete data modeling. However, existing methods typically rely on a fixed-rate transition matrix during training, which not only limits the expressiveness of latent representations—a fundamental strength of variational methods—but also constrains the overall design space. To address these limitations, we propose **Discrete Markov Bridge**, a novel framework specifically designed for discrete representation learning. Our approach is built upon two key components: *Matrix*-learning and *Score*-learning. We conduct a rigorous theoretical analysis, establishing formal performance guarantees for *Matrix*-learning and proving the convergence of the overall framework. Furthermore, we analyze the space complexity of our method, addressing practical constraints identified in prior studies. Extensive empirical evaluations validate the effectiveness of the proposed **Discrete Markov Bridge**, which achieves an Evidence Lower Bound (ELBO) of **1.38** on the Text8 dataset, outperforming established baselines. Moreover, the proposed model demonstrates competitive performance on the CIFAR-10 dataset, achieving results comparable to those obtained by image-specific generation approaches.¹

1 Introduction

A fundamental question in generative modeling is estimating an underlying distribution, μ , from observed data and subsequently generating new samples from this distribution. Among the various generative models proposed, diffusion models have exhibited remarkable performance in both continuous [1, 2] and discrete domains [3, 4], demonstrating their versatility and effectiveness in diverse applications. These models effectively capture complex data distributions, enabling high-quality sample generation in various applications. However, despite their strong connection to variational models [5, 6], which are known for their impressive generative capabilities, diffusion models have yet to integrate the latent encoding ability inherent to variational approaches. Specifically, in the discrete domain, the noise rate transition matrices within discrete diffusion models are fixed and constrained, resulting in a limited design space and reduced expressive capacity. To the best of our knowledge, only the Absorb and Uniform Matrix [3, 4, 7] have been considered in computations due to their simplicity in handling exponential term calculations.

¹Implementation code is available at <https://github.com/Henry839/Discrete-Markov-Bridge>.

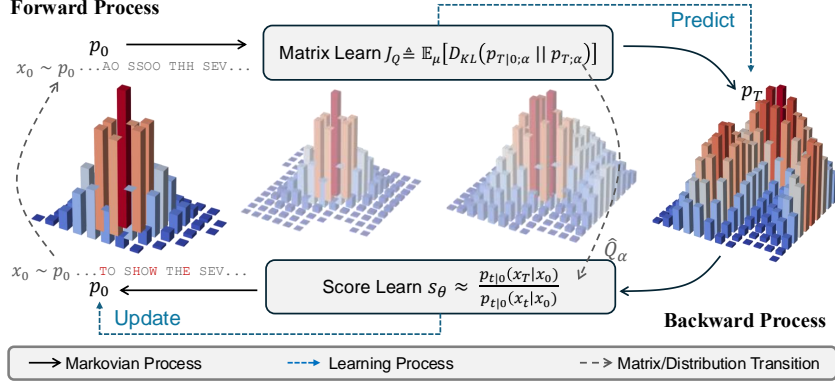


Figure 1: Overview of the **DMB** framework. **DMB** consists of two component: the *Matrix*-learning and the *Score*-learning. The *Matrix*-learning process is designed to learn an adaptive transition rate matrix, which facilitates the estimation of an adapted latent distribution. Concurrently, the *score*-learning process focuses on estimating the probability ratio necessary for constructing the inverse transition rate matrix, thereby enabling the reconstruction of the original data distribution.

In this study, we challenge the convention of using predefined static matrix in discrete modeling by introducing a novel approach, termed the **Discrete Markov Bridge (DMB)**, which aims to integrate the strengths of variational methods with discrete diffusion models, offering a more robust and efficient solution for complex discrete-state systems. This methodology seeks to enhance the modeling capabilities by leveraging the theoretical foundations of variational inference within the framework of discrete diffusion processes. Specifically, DMB is structured as a bidirectional two-stage learning algorithm. It comprises a forward variational process, i.e., *Matrix*-learning, that maps the data distribution to a learned distribution, followed by a backward decoding process, i.e., *Score*-learning, that reconstructs the data distribution from the learned representation.

In the *Matrix*-learning process, we propose a novel parameterized rate transition matrix that enhances the flexibility of the overall algorithm. This refinement allows for greater adaptability and improved performance in dynamic learning environments. The rate transition matrix is designed to be diagonalizable, ensuring high spatial efficiency while facilitating the rapid computation of matrix exponentials. On the other hand, in the *Score*-learning process, a neural network is employed to model the concrete score [4, 8]. This score serves a crucial role in the derivation of the backward rate transition matrix. As for the sampling procedure, the rate transition matrix derived from the *Matrix*-learning process and the neural network obtained from the *Score*-learning process are jointly employed to solve the backward differential equation.

Within this framework, a broad spectrum of tasks can be effectively addressed. For discrete data modalities such as text, the model supports non-autoregressive generation, following the approach outlined in [9]. In this work, we demonstrate that our proposed method surpasses the performance of the previously established SEDD model [4]. For image data, the model can be integrated with a VQ-VAE architecture [6], yielding performance on par with that of DDPM when evaluated on the CIFAR-10 dataset.

We summarize our contributions as follows:

- **Novel Framework for Discrete Data (Section 3):** We introduce the Discrete Markov Bridge, a new variational framework for learning discrete representations. By leveraging a variational formulation, this approach provides a novel method for modeling complex discrete data.
- **Theoretical Guarantee (Section 4):** We present a theoretical guarantee for the *Matrix*-learning process, covering both its validity and accessibility. Furthermore, we provide a comprehensive analysis of the entire framework, culminating in a formal convergence proof.
- **Addressing Practical Issues (Section 5):** Building on the theoretical insights established earlier, we propose a computationally efficient matrix to tackle the practical challenges discussed in Section 5. We then evaluate the model’s performance through experiments, demonstrating that it outperforms baseline methods in text modeling and provides competitive image modeling results.

2 Preliminaries and Related Works

2.1 Continuous-Time Discrete Markov Chain

Let $\mathbb{X} = \{1, 2, \dots, n\}$ denote a finite state space, where $n \in \mathbb{R}$. A continuous time discrete Markov chain (CTDMC) defined on \mathbb{X} is represented as $\{X(t) \mid t \in \mathbb{R}, X(t) \in \mathbb{X}\}$. For convenience, we use the notation $X_t \triangleq X(t)$. The probability of transitioning from state $x \in \mathbb{X}$ at time t to state $y \in \mathbb{X}$ at time $t + s$ is denoted as $p_{t+s|t}(y|x) \triangleq P(X_{t+s} = y \mid X_t = x)$. Similarly, the probability that X_t takes state x at time t is expressed as $p_t(x) \triangleq P(X_t = x)$. The probability distribution over the state space at time t is then given by the vector $p_t \triangleq (p_t(1), p_t(2), \dots, p_t(n))$. The core component to describe a continuous time discrete Markov chain is the rate transition matrix. We defined the rate transition probability as follows:

$$q_t(x, y) \triangleq \frac{dp_{t+s|t}(y|x)}{ds} = \lim_{\Delta s \rightarrow 0} \frac{p_{t+s|t}(y|x) - p_{t|t}(y|x)}{\Delta s} = \lim_{\Delta s \rightarrow 0} \frac{p_{t+s|t}(y|x) - \delta_x(y)}{\Delta s},$$

where $\delta_x(y)$ is the Dirac delta function. The Forward Kolmogorov Equation can be written as $\frac{dp_t}{dt} = p_t Q^{(t)}$. The notation $Q_{x,y}^{(t)} \triangleq q_t(x, y)$, for all $x, y \in \mathbb{X}$, denotes the rate transition matrix at time t . The subscripts x and y indicate the row and column indices, respectively. Each rate transition matrix satisfies the conditions: the sum of each row must be zero, and all off-diagonal entries must be non-negative. Formally, this is expressed as $\sum_y Q_{x,y} = 0$ for all x and $Q_{x,y} \geq 0$ for all $y \neq x$.

2.2 Related Works

Prior Learning Leveraging a prior is a longstanding paradigm in machine learning. In the field of natural language processing, for example, training typically begins with pretrained language models [10–16]. Likewise, pretrained models are highly valued in computer vision [17]. In our approach, the concept of a prior is equally fundamental: the forward process adaptively refines this prior based on the evolving training dynamics of the backward process.

Discrete Diffusion Models Diffusion models [2, 1, 18, 19] add noise to data and use a denoiser for reconstruction, achieving success in image tasks and gaining traction in discrete domains like natural language [20, 4, 3, 21–24]. Some methods map discrete data to continuous space [20, 21], introducing rounding errors, while others operate directly in discrete space but impose rigid, non-learnable noise structures [3, 4]. In the continuous domain, trainable Gaussian parameters improve flexibility [25], but no such method exists for discrete diffusion, where Gaussian distributions also remain restrictive. Moreover, masked discrete diffusion models struggle to learn temporal dependencies [26].

Flow Models Flow-based models [27–32] constitute a prominent class of machine learning models characterized by their ability to perform reversible transformations on data representations. In contrast to conventional flow models, which rely on transformation paths predefined by human designers [31, 29], our approach autonomously learns these paths, enhancing adaptability and expressiveness in data modeling.

3 Discrete Markov Bridge

The target distribution, denoted as $\mu \in \mathbb{R}^n$, is a probability vector, meaning that its elements are non-negative and collectively sum to one. As shown in Figure 1, our objective is to estimate the distribution at one endpoint of the Markov chain, denoted as p_0 , such that $p_0 \approx \mu$. The other endpoint, denoted as p_T , serves as the distribution for the latent variables or prior. To achieve the specified objectives, the proposed **DMB** framework is structured into two distinct components: *Matrix Learning* and *Score Learning*.

The *Matrix-learning* serves as a forward bridge, facilitating the transition from μ to the latent distribution. Conversely, the *Score-learning* function delineates a reverse pathway from the latent distribution back to μ , leveraging the groundwork established by the *Matrix-learning* process. This dual-function framework ensures a comprehensive bidirectional understanding of the data structure, enhancing the robustness of the analytical model.

The structure of the **DMB** is demonstrated in Algorithm 2. This pseudocode illustrates two nested while loops that operate within the overarching while loop governing the training epochs. Each of these nested loops corresponds to a distinct learning stage within the framework, effectively organizing the training process into two phases. We list the following theorem to ensure the reversibility of the forward and backward Markovian processes.

Theorem 3.1 (Reversibility [3, 4]). *Given the Forward Kolmogorov Equation of a CTDMC:*

$$\frac{dp_t}{dt} = p_t Q^{(t)} \quad (1)$$

There exists a reverse CTDMC with Forward Kolmogorov Equation:

$$\frac{dp_{T-t}}{dt} = p_{T-t} \hat{Q}^{(T-t)}, \text{ where } \hat{Q}_{x,y}^{(t)} = \frac{p_t(y)}{p_t(x)} Q_{y,x}^{(t)} \quad (2)$$

This theorem elucidates the reverse form of a CTDMC, proposing that knowledge of the probability ratio enables the derivation of a reversal of the original Markov chain that is almost everywhere equivalent. This assertion underscores the theoretical framework necessary to comprehend the conditions under which the reverse process mirrors the dynamics of the forward stochastic process.

We structure the learning process of the framework by employing the continuous-time Evidence Lower Bound (ELBO) as an alternative optimization objective to Maximum Likelihood Estimation (MLE). In the **DMB** framework, both *Matrix*-learning and *Score*-learning collaboratively optimize distinct segments of the full bound through their respective subprocesses.

3.1 Matrix-Learning

In the *Matrix*-learning process, our primary objective is to estimate the rate transition matrix Q_α , where α denotes the set of model parameters. For simplicity, we assume that the forward rate transition matrix at time t , denoted $Q_\alpha^{(t)}$, is given by $\sigma(t)Q_\alpha$. Furthermore, we employ the following Q_α :

$$Q_\alpha = A \begin{bmatrix} -\sum_{i=1}^{n-1} a_i & a_1 & \dots & a_{n-2} & a_{n-1} \\ 0 & -\sum_{i=2}^{n-1} a_i & \dots & a_{n-2} & a_{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -a_{n-1} & a_{n-1} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} A^{-1} := H A^{-1} \quad (3)$$

, where $\{a_1, a_2, \dots, a_{n-1}\} = \alpha$ are parameters for learning, A, A^{-1} are fixed predefined permutation matrices and H is introduced to stand for the upper-triangle matrix. The derivation and underlying rationale for utilizing this matrix are detailed in Section 4 and further explored in Section 5.1. Another essential component of this process is μ , which is approximated using the currently predicted p_0 obtained through *Score*-learning as a prior (see Section 3.2). By integrating Equation (1) from time 0 to time t , the following equation can be derived:

$$p_t = p_0 \exp\left\{\int_0^t \sigma(s) ds Q_\alpha\right\} \quad (4)$$

Note that the exponential in the formula is a matrix exponential. The training procedure aims to minimize a portion of the variational bound, leading to the following objective function J_Q :

$$J_Q \triangleq \mathbb{E}_\mu[D_{KL}(p_{T|0;\alpha} || p_{T;\alpha})], \quad (5)$$

where the conditional probability distribution $p_{T|0;\alpha}$ is given by the rows of $\exp\{\int_0^T \sigma(s) ds Q_\alpha\}$:

$$p_{T|0;\alpha}(x_T | x_0) = \exp\left\{\int_0^T \sigma(s) ds Q_\alpha\right\}_{x_0, x_T} \quad (6)$$

The final distribution $p_{T;\alpha}$ is obtained by multiplying the initial distribution p_0 with the conditional distribution, as presented in Equation (4), evaluated at time $t = T$.

Algorithm 2 Training Algorithm of the DMB

Input: Target discrete data $X \sim \mu$

```
1: Initialize  $p_0, p_T \leftarrow \text{random\_init}()$ 
2: while not converge do
3:   Sample a batch of discrete instance  $X_0 \sim \mu$ . /* Data for the two learning processes. */
   /* Matrix Learning */
4:    $\text{step} \leftarrow 0$ 
5:   while  $\text{step} \leq \text{max\_step}$  &  $\mathcal{L}_Q \geq \epsilon_Q$  do
6:     Update  $Q_\alpha, \mathcal{J}_Q$  according to Eqn. (5) and predict  $p_T$  using Eqn. (4) at  $t = T$ .
7:      $\text{step} \leftarrow \text{step} + 1$ 
8:   end while
   /* Score Learning */
9:    $\text{step} \leftarrow 0$ 
10:  while  $\text{step} \leq \text{max\_step}$  &  $\mathcal{J}_{\text{score}} \geq \epsilon_{\text{score}}$  do
11:    Update  $s_\theta, \mathcal{J}_{\text{score}}$  w.r.t. current  $Q_\alpha$  using Eqn. (8).
12:     $\text{step} \leftarrow \text{step} + 1$ 
13:  end while
14:  Predict updated  $p_0$  that estimates  $\mu$  using Eqn. (10). /* Used for Matrix Learning */
15:  if  $\mathcal{J}_Q + \mathcal{J}_{\text{score}} < \epsilon$  then
16:    converge  $\leftarrow \text{TRUE}$ 
17:  end if
18: end while
```

3.2 Score-learning

Score-learning constitutes a reverse process of *Matrix*-learning. It is noted that in Theorem 3.1, the reverse rate transition matrix adheres to the following relationship:

$$\hat{Q}_{x,y}^{(t)} = \frac{p_t(y)}{p_t(x)} Q_{x,y} \sigma(t) \quad (7)$$

Consequently, while *Matrix*-learning handles the forward rate transition matrix Q , *Score*-learning focuses on managing the remaining part, i.e. $\frac{p_t(y)}{p_t(x)}$. A learnable model $s_\theta(x_t, t)_y$ is designed to model the ratio, and the main part of the continuous time Evidence Lower Bound (ELBO) [3–5] is leveraged as the training objective, denoted as $\mathcal{J}_{\text{score}}$:

$$\int_0^T \mathbb{E}_{x_0 \sim \mu, x_t \sim p_{t|0}} \left[\sum_{y \neq x_t} Q_{y,x_t}^{(t)} \left(s_\theta(x_t, t)_y - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} + \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \left(\log \left(\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right) - \log s_\theta(x_t, t)_y \right) \right) \right] dt \quad (8)$$

To provide a comprehensive understanding, we present the complete ELBO as follows, demonstrating how *Matrix*-learning and *Score*-learning collaboratively contribute to minimizing the ELBO bound.

$$\mathbb{E}_{x_0 \sim \mu} [-\log p_{0;\theta}(x_0)] \leq \mathcal{J}_{\text{score}} + \mathcal{J}_Q. \quad (9)$$

Estimating μ The estimation of μ is expressed as Equation (10). The equation below is derived under the Euler method and can be generalized to other ODE-solving methods. Suppose the inference time process is partitioned as: $[0, t_1], [t_1, t_2], \dots, [t_n, T]$. By Bayesian rules:

$$\mu(x_0) \approx p_0(x_0) = \mathbb{E}_{X_T, X_n, \dots, X_1} [p_{0|1}(x_0|x_1)]. \quad (10)$$

Under the guidance of Equation (10), the sampling process begins with drawing x_T , followed by obtaining x_n through the conditional distribution $p_{t_n|T}(x_n|X_T = x_T)$. This procedure continues iteratively, generating x_{n-1} , and proceeding sequentially until the complete sequence $\{X_T, X_n, \dots, X_1\}$ is sampled. Subsequently, the conditional probability $p_{0|1}(x_0|x_1)$ is determined. By repeating this process multiple times and averaging the sampled probabilities, an estimation can be obtained by approximating the expectation with the empirical mean.

3.3 Sampling

The sampling process is done under the cooperation of *Matrix*-learning and *Score*-learning in a similar way as estimating μ . The reverse rate transition matrix is calculated as Equation (7), and an

ode-solving method such as the Euler method can be further applied to solve Equation (2). Noticed that, as shown in line 15 of Algorithm 2, the sampling process is performed every time after the *Score*-learning process to gain the estimation of μ and samples for evaluation.

4 Theoretical Foundations

4.1 Validity and Accessibility of *Matrix*-learning

The validity and accessibility of the backward process are established by Theorem 3.1. In this subsection, we extend our analysis to the same aspects of the *Matrix*-learning process. Specifically, **validity** concerns the ultimate state of the forward process and whether it remains confined within a well-defined domain, i.e., whether a probability distribution transforms into another valid probability distribution. **Accessibility**, on the other hand, pertains to the ability of the process to transition between any two arbitrary discrete distributions, thereby characterizing the reachability and adaptability of the *Matrix*-learning process.

Validity Proposition 4.1, presented below, establishes that any transformation originating from a probability distribution must result in another probability distribution. This theorem guarantees that, despite the presence of errors in the learning process, the outcome remains a valid probability distribution. For a detailed proof, refer to Section A.

Proposition 4.1 (Conservation of the Sum). *For two arbitrary vectors $\phi, \mu \in \mathbb{R}^n$, rate transition matrix $Q \in \mathbb{R}^{n \times n}$, if $\phi = \mu \exp\{Q\}$, then*

$$\sum_{i=1}^n \phi[i] = \sum_{i=1}^n \mu[i]$$

Accessibility Theorem 4.2 ensures that any two probability distributions are accessible in the forward process. Consequently, this implies that the optimality of *Matrix*-learning can be achieved, provided the presence of a strong optimizer.

Theorem 4.2 (Accessibility). *For two arbitrary discrete distributions $p, q \in \mathbb{R}^n$, there exists a rate transition matrix $Q \in \mathbb{R}^{n \times n}$ such that:*

$$p = qe^Q \quad (11)$$

The central idea of the proof is to construct a specialized matrix that possesses strong representational capacity while remaining computationally manageable within the framework of matrix exponentiation. The designed matrix, which is depicted in Lemma 4.3, is an upper triangle matrix with the vanished sum of rows. A remarkable characteristic of this matrix is its elegant eigendecomposition form, which presents a well-structured and analytically convenient representation. Its eigenmatrix is an all-one upper triangular matrix, as shown in Lemma 4.3.

Lemma 4.3. *Let matrix $Q \in \mathbb{R}^{n \times n}$ and hold the following form:*

$$Q = H$$

, where H is defined in Equation (3), then Q can be diagonalized in the following form:

$$Q = U \Lambda U^{-1}$$

$$\text{, where } U = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \Lambda = \text{diag}(\{-\sum_{i=1}^{n-1} a_i, -\sum_{i=2}^{n-1} a_i, \dots, -a_{n-1}, 0\}).$$

There are two key observations regarding the Q matrix. First, it contains only $n - 1$ parameters, which constitute the minimal set necessary to solve Equation (11). This sufficiency implies that the solution derived for the Q matrix is unique. Second, the matrix retains nonzero elements exclusively in its upper triangular portion, implying that each element can transition only to those with a larger index. This observation raises an additional consideration: for effective state transitions, the matrix must allocate sufficient “mass” or probability. Consequently, a matrix is required to appropriately adjust the indices of elements within the finite set \mathbb{X} , as shown in Lemma 4.4. Lemma 4.4 establishes

that, after a permutation, the cumulative probability at each element of the initial distribution in the transition process is greater than or equal to that of the target distribution. This guarantees that elements with surplus probability can redistribute their excess, while those with a deficiency can receive the necessary adjustments, ensuring a balanced transformation.

Lemma 4.4. *For arbitrary distribution $p, q \in \mathbb{R}^n$, there exists a permutation matrix A such that:*

$$\frac{p'_1}{q'_1} \leq \frac{p'_1 + p'_2}{q'_1 + q'_2} \leq \dots \leq \frac{\sum_{i=1}^k p'_i}{\sum_{i=1}^k q'_i} \leq \dots \leq \frac{\sum_{i=1}^n p'_i}{\sum_{i=1}^n q'_i} = 1 \quad (12)$$

where $p' = pA$, $q' = qA$, p'_i is the i -th entry of p' , and q'_i is the i -th entry of q' .

Lemma 4.5. *Let $Q \in \mathbb{R}^{n \times n}$ be a rate transition matrix, $A \in \mathbb{R}^{n \times n}$ be a permutation matrix, then AQA^{-1} is a rate transition matrix.*

By integrating the lemmas above, we aim to establish the proof of Theorem 4.2. A comprehensive derivation of these lemmas and the theorem is provided in Section B.

4.2 Convergence

As discussed earlier, the **DMB** framework operates as a two-step learning algorithm, necessitating a thorough examination of its convergence properties. In this section, we present a formal theorem that establishes the convergence guarantee for the entire algorithm. The convergence problem is nontrivial, as the *Score*-learning process does not merely constitute a direct inversion of the *Matrix*-learning process. The discrepancy arises because the score model s_θ is trained under the supervision of the distribution μ , rather than $p_0^{(k)}$, where k denotes the epoch number. To be specific, we have

Proposition 4.6 (Supervision of *Score*-learning). *Suppose Q_t 's elements are non-zeros, the training objective is depicted as in Equation (8), then the optimality of the score model $s_{\theta^*}(x_t, t)_y$ satisfies:*

$$s_{\theta^*}(x_t, t)_y = \mathbb{E}_{x_0 \sim \mu_{0|t}(\cdot|x_t)} \left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right] = \frac{\sum_{x_0} \mu(x_0) p_{t|0}(y|x_0)}{\sum_{x_0} \mu(x_0) p_{t|0}(x_t|x_0)}$$

The proposition presented above illustrates the influence of μ on the training process and underscores the challenge of convergence arising from the absence of $p_0^{(k)}$. A detailed proof of this proposition can be found in Section C.

Under the assumption that each process achieves optimality, the following theorem establishes the convergence of **DMB** from the perspective of KL divergence, thereby demonstrating the validity of the overall **DMB** framework. Moreover, given our primary focus on the algorithmic aspects, this assumption is justified, consistent with prior work that introduces new frameworks, such as Goodfellow et al. [33]. Notably, although the training objective of the *Score*-learning process is the continuous ELBO bound, the theorem presented below can be generalized to encompass a broader class of objectives. This generalization suggests the potential for designing improved training objectives within our framework.

Theorem 4.7 (Convergence of the algorithm). *If we assume optimality is achieved in every epoch of the *Matrix*-learning process and the *Score*-learning process, and we denote the k -th epoch estimation of μ as $p_0^{(k)}$, then $\lim_{k \rightarrow \infty} D_{KL}(\mu || p_0^{(k)})$ converges.*

Please refer to Section D for the proof.

5 Practical Issues and Experiments

5.1 High Dimensional Data

In this section, we discuss the practical issues of **DMB** by assuming our data coming from a high dimensional space, i.e. $\mu \in \mathbb{R}^{d \times n}$, where n is the size of the finite set and d is the number of dimensions. For instance, for textual data, n is the size of the vocabulary and d is the sequence length.

Assumptions. When dealing with high-dimensional data, such as textual sequences, the combinatorial explosion in the number of possible sequences imposes prohibitive constraints on both storage and computational efficiency. To address this challenge, certain assumptions are introduced [4, 3, 34]:

- Independent Evolution: $p_{T|0;\alpha}(x_T|x_0) = \prod_{i=1}^d p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})$
- Independent Terminal: $p_T(x_T) = \prod_{i=1}^d p(x_T^{(i)})$

The first assumption posits that, during the forward process, each dimension evolves independently. The second assumption asserts that the latent space consists of independent dimensions.

5.2 Addressing Practical Issues

Both the **DMB** model and discrete diffusion models [4, 3, 35] face significant challenges related to the Q matrix. In particular, during the *Score*-learning process, the computational efficiency of matrix exponential operations becomes a critical constraint. Furthermore, the *Matrix*-learning process often requires storing the entire Q matrix, posing substantial concerns regarding space efficiency. These limitations have been the primary reasons restricting previous studies to utilizing only the Uniform and Absorb matrices.

As Jean le Rond d’Alembert once remarked, *Algebra is generous; she often gives more than is asked of her*. In the context of proving Theorem 4.2, we identify a distinct class of matrices, as mentioned in Section 3.1 and further rigorously discussed in Lemma 4.3. This structured approach not only underscores the theoretical underpinnings but also highlights the practical implications of matrix manipulation in these models.

Efficient Computation of the permutation matrix. Before proceeding with the analysis of the Q_α matrix, we first outline the computation of the predefined permutation matrix A . As illustrated in the assumptions, the evolution of each dimension occurs independently. Consequently, for each dimension, the permutation matrix is computed separately. In accordance with Lemma 4.4, we assume the denominator to be constant. Therefore, the permutation matrix for the i -th dimension satisfies the following inequality:

$$\mu(X_0^{(i)} = j) \leq \mu(X_0^{(i)} \leq j + 1), \forall j \in 1, 2, \dots, n - 1$$

The marginal distribution $\mu(X_0^{(i)})$ can be efficiently estimated in the form of a histogram by extracting a subbatch from the dataset. Subsequently, the permutation matrix is computed using a fast sorting algorithm with a time complexity of $O(n \log n)$.

Efficient Computation of Matrix Exponential. Matrix exponential is difficult to calculate as it’s defined through Tylor expansion, however, a property exists:

Proposition 5.1. For a matrix $Q \in \mathbb{R}^{n \times n}$ and a non-degenerate matrix $D \in \mathbb{R}^{n \times n}$, we have:

$$\exp\{DQD^{-1}\} = D \exp\{Q\}D^{-1}$$

Please refer to Section E for the derivation of Proposition 5.1. By Proposition 5.1,

$$\exp\{Q_\alpha\} = \exp\{(AU)\Lambda_\alpha(AU)^{-1}\} = (AU) \exp\{\Lambda_\alpha\}(AU)^{-1} \quad (13)$$

, where U is the all-one upper triangle matrix, Λ_α is a diagonal matrix parameterized by α . Therefore, the computation of the matrix exponential is reduced to evaluating the exponential of a diagonal matrix, which is significantly more efficient.

Space Efficiency. For the permutation matrices $A, A^{-1} \in \mathbb{R}^{d \times n \times n}$, a total of $d \times 2n$ parameters are required. Apart from A, A^{-1} , the upper triangle matrix can be decomposed into a non-parameterized all-one upper triangle matrix, a parameterized diagonal matrix, and a constant matrix. Consequently, the storage requirement is of the order $O(nd)$ parameters.

5.3 ELBO Bound Calculation

As shown in Equation (9), the computation of the full bound necessitates the evaluation of both the J_{score} and the expected Kullback–Leibler (KL) divergence between the evolved distribution and the target distribution, expressed as $\mathbb{E}_\mu D_{KL}(P_{T|0}||P_T)$. Under the assumptions outlined within Section 5.1, we can derive a closed-form expression for computing the KL term:

Proposition 5.2. *The KL term can be calculated as:*

$$D_{KL}(p_{T|0;\alpha}(x_T|x_0)||p_T(x_T)) = \sum_{i=1}^d D_{KL}(p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})||p_T(x_T^{(i)})) \quad (14)$$

Table 1: The results were tested 1000 times on the Text8 dataset. We adopt the baseline results reported in [4] for comparison. AR: Autoregressive. NAR: Non-autoregressive.

Type	Model	BPC (\downarrow)
AR	IAF/SCF [36]	1.88
	AR Argmax Flow [34]	1.39
	Discrete Flow [37]	1.23
NAR	SEDD Uniform [4]	≤ 1.47
	SEDD Absorb [4]	≤ 1.39
	D3PM Uniform [7]	≤ 1.61
	D3PM Absorb [7]	≤ 1.45
	Mult. Diffusion [34]	≤ 1.72
	MAC [38]	≤ 1.40
	BFN [39]	≤ 1.41
	DMB (Ours)	$\leq \mathbf{1.38}$

Table 2: CIFAR-10 Results. We report inception score (IS), and Fréchet Inception Distance (FID) score. Results are adopted from Ho et al. [40].

Model	IS (\uparrow)	FID (\downarrow)
Conditional		
EBM [41]	8.30	37.9
JEM [42]	8.76	38.4
BigGAN [43]	9.22	14.73
StyleGAN2 + ADA (v1) [44]	10.06	2.67
Unconditional		
Gated PixelCNN [45]	4.60	65.93
PixelQIN [46]	5.29	49.46
EBM [41]	6.78	38.2
NCSN [47]	8.87 ± 0.12	25.32
SNGAN [48]	8.22 ± 0.05	21.7
SNGAN-DDLS [49]	9.09 ± 0.10	15.42
StyleGAN2 + ADA (v1) [44]	9.74 ± 0.05	3.26
DDPM (fixed isotropic) [40]	7.67 ± 0.13	13.51
DDPM (simple) [40]	9.46 ± 0.11	3.17
Ours	8.64	11.63

5.4 Experiment

In this section, the performances of **DMB** on Text8 and CIFAR-10 are reported.

Best Performance on Text8 We conduct our experiments using the Text8 dataset to evaluate the proposed framework. The experimental results are summarized in Table 1. To ensure statistical reliability, the model was evaluated across 1,000 independent trials. The primary performance metric, the Evidence Lower Bound (ELBO), was computed following the methodology outlined in Section 5.3. Our proposed approach, **DMB**, achieves a Bits Per Character (BPC) bound of 1.38, surpassing baseline models such as SEDD [4], a representative discrete diffusion model. Notably, our approach does not modify the vocabulary; in particular, no mask token is introduced. Consequently, when compared to similar methods that also do not incorporate a mask token—such as SEDD Uniform and D3PM Uniform—our approach demonstrates an improvement of approximately 0.1 points.

Competitive Performance on CIFAR-10 Although our approach is not specifically tailored for image modeling tasks, we evaluate its performance on the CIFAR-10 dataset using a VQ-VAE framework [6]. The quantitative results are presented in Table 2. Our method, **DMB**, achieves an Inception Score (IS) of 8.64 and a Fréchet Inception Distance (FID) of 11.63. Notably, these results surpass those of several models explicitly designed for image generation, including DDPM (fixed isotropic) and SNGAN [48], in both IS and FID metrics. This demonstrates the effectiveness and generalization capability of our model beyond its primary design scope.

6 Conclusion

In this study, we propose a novel paradigm, the **Discrete Markov Bridge (DMB)**, which combines the strengths of variational methods with the capabilities of discrete diffusion models. We provide theoretical guarantees to substantiate the feasibility and effectiveness of the proposed *Matrix*-learning process and prove the convergence of the DMB algorithm. In addition to our theoretical contributions, we conduct extensive empirical evaluations on the Text8 and CIFAR-10 datasets. The experimental results indicate that **DMB** not only surpasses existing baselines such as SEDD [4] in text modeling tasks, but also achieves competitive performance in image modeling on CIFAR-10, thereby demonstrating its potential as a unified framework for discrete representation learning.

Acknowledgement

We thank Junqi Wang from BIGAI for inspiration of discovering the matrix and Jianwen Xie from Lambda for discussion.

References

- [1] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=PXTIG12RRHS>.
- [2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [3] Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models, 2022.
- [4] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *CoRR*, 2023.
- [5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.
- [6] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL <https://arxiv.org/abs/1711.00937>.
- [7] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 17981–17993, 2021. URL <https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html>.
- [8] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data, 2023. URL <https://arxiv.org/abs/2211.00802>.
- [9] Jiatao Gu and Xu Tan. Non-autoregressive sequence generation. In Luciana Benotti, Naoaki Okazaki, Yves Scherrer, and Marcos Zampieri, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 21–27, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-tutorials.4. URL <https://aclanthology.org/2022.acl-tutorials.4>.
- [10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *CoRR*, 2019.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, 2019.
- [12] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [13] 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.
- [14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020.
- [15] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

- [18] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.
- [19] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.
- [20] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.
- [21] Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-based diffusion language models, 2023.
- [22] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models, 2023.
- [23] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data, 2022.
- [24] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL <https://arxiv.org/abs/2502.09992>.
- [25] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2023. URL <https://arxiv.org/abs/2107.00630>.
- [26] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling, 2024. URL <https://arxiv.org/abs/2409.02908>.
- [27] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.
- [28] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions, 2018.
- [29] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.
- [30] Victor Garcia Satorras, Emiel Hooeboom, Fabian B. Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows, 2022.
- [31] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023.
- [32] Asher Trockman and J. Zico Kolter. Orthogonalizing convolutional layers with the cayley transform, 2021.
- [33] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL <https://arxiv.org/abs/1406.2661>.
- [34] Emiel Hooeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021. URL <https://arxiv.org/abs/2102.05379>.
- [35] Kun Sun, Mingli Jing, Yuliag Hu, and Yao Jiao. Image style transfer based on improved convolutional neural. In *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pages 575–579, 2021. doi: 10.1109/ICAICE54393.2021.00114.
- [36] Zachary M. Ziegler and Alexander M. Rush. Latent normalizing flows for discrete sequences, 2019. URL <https://arxiv.org/abs/1901.10548>.
- [37] Dustin Tran, Keyon Vafa, Kumar Agrawal, Laurent Dinh, and Ben Poole. Discrete flows: Invertible generative models of discrete data. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/e046ede63264b10130007afca077877f-Paper.pdf.
- [38] Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way, 2022. URL <https://arxiv.org/abs/2205.13554>.
- [39] Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks, 2024. URL <https://arxiv.org/abs/2308.07037>.

- [40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.
- [41] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, pages 3603–3613, 2019.
- [42] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.
- [43] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.
- [44] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676v1*, 2020.
- [45] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.
- [46] Georg Ostrovski, Will Dabney, and Remi Munos. Autoregressive quantile networks for generative modeling. In *International Conference on Machine Learning*, pages 3936–3945, 2018.
- [47] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.
- [48] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.
- [49] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.
- [50] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL <https://arxiv.org/abs/1711.05101>.

Appendices

Contents

A	Proof of Conservation of the Sum	14
B	Proof of Accessibility	14
B.1	Proof of Lemmas	14
B.2	Proof of the theorem	16
C	Proof of Supervision of <i>Score</i>-learning	17
D	Proof of Convergence	18
D.1	Proof of Lemmas	18
D.2	Proof of the theorem	18
E	Derivation of Matrix Exponential Calculation	19
F	Derivation of KL term calculation proposition	19
G	Additional Experimental details	20
G.1	Model Details	20
G.2	Training Details	20
H	Limitations and Societal Impact	20

A Proof of Conservation of the Sum

Proposition A.1 (Conservation of the Sum). *For two arbitrary vectors $\phi, \mu \in \mathbb{R}^d$, rate transition matrix $Q \in \mathbb{R}^{d \times d}$, if $\phi = \mu \exp Q$, then*

$$\sum_{i=1}^d \phi[i] = \sum_{i=1}^d \mu[i]$$

Proof. As $\phi = \mu \exp Q$,

$$\phi(i) = \sum_j \mu(j) (\exp\{Q\})_{j,i}$$

Therefore,

$$\sum_i \phi(i) = \sum_i \sum_j \mu(j) (\exp\{Q\})_{j,i}$$

As we have

$$\sum_j (\exp\{Q\})_{i,j} = 1$$

Thus,

$$\sum_i \phi(i) = \sum_j \mu(j) \sum_i (\exp\{Q\})_{j,i} = \sum_j \mu(j)$$

■

B Proof of Accessibility

B.1 Proof of Lemmas

Lemma B.1. *Let matrix $Q \in \mathbb{R}^{d \times d}$ and hold the following form:*

$$Q = \begin{bmatrix} -\sum_{i=1}^{n-1} a_i & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} \\ 0 & -\sum_{i=2}^{n-1} a_i & a_2 & \dots & a_{n-2} & a_{n-1} \\ 0 & 0 & -\sum_{i=3}^{n-1} a_i & \dots & a_{n-2} & a_{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -a_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

then Q can be diagonalized in the following form:

$$Q = U \Lambda U^{-1}$$

$$, \text{where } U = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \Lambda = \text{diag}(\{-\sum_{i=1}^{n-1} a_i, -\sum_{i=2}^{n-1} a_i, \dots, -a_{n-1}, 0\})$$

$$\text{Proof. } Q = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{diag}(\{-\sum_{i=1}^{n-1} a_i, -\sum_{i=2}^{n-1} a_i, \dots, -a_{n-1}, 0\}) \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

■

Lemma B.2. For arbitrary distribution $p, q \in \mathbb{R}^{1 \times d}$, there exists an permutation matrix A such that:

$$\frac{p'_1}{q'_1} \leq \frac{p'_1 + p'_2}{q'_1 + q'_2} \leq \dots \leq \frac{\sum_{i=1}^k p'_i}{\sum_{i=1}^k q'_i} \leq \dots \leq \frac{\sum_{i=1}^n p'_i}{\sum_{i=1}^n q'_i} = 1 \quad (15)$$

where $p' = pA$, $q' = qA$, p'_i is the i -th entry of p'

Proof. It's obvious that there exists a permutation matrix A which can sort $\frac{p_i}{q_i}$ ascendly, i.e.:

$$\frac{p'_i}{q'_i} \leq \frac{p'_{i+1}}{q'_{i+1}}$$

, where $p' := pA$, $q' := qA$, and the corner mark i refer to the i -th entry.

Also, we can demonstrate that:

$$\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \Rightarrow \frac{a_1}{b_1} \leq \frac{a_1 + a_2}{b_1 + b_2} \leq \frac{a_2}{b_2} \quad (\triangle)$$

The inequality we need to prove is:

$$\frac{\sum_{i=1}^k p'_i}{\sum_{i=1}^k q'_i} \leq \frac{\sum_{i=1}^{k+1} p'_i}{\sum_{i=1}^{k+1} q'_i}$$

and it's sufficient to proving the following inequality:

$$\frac{\sum_{i=1}^k p'_i}{\sum_{i=1}^k q'_i} \leq \frac{p'_{k+1}}{q'_{k+1}}$$

We then start to prove the inequality by induction.

$k = 1$: Let $a_1 = p'_1, a_2 = p'_2, b_1 = q'_1, b_2 = q'_2$, and by using inequality \triangle , the statement is proved.

$k + 1$: By induction:

$$\frac{\sum_{i=1}^k p'_i}{\sum_{i=1}^k q'_i} \leq \frac{p'_{k+1}}{q'_{k+1}}$$

By leveraging inequality \triangle :

$$\frac{\sum_{i=1}^{k+1} p'_i}{\sum_{i=1}^{k+1} q'_i} \leq \frac{p'_{k+1}}{q'_{k+1}}$$

As $\frac{p'_{k+1}}{q'_{k+1}} \leq \frac{p'_{k+2}}{q'_{k+2}}$:

$$\frac{\sum_{i=1}^{k+1} p'_i}{\sum_{i=1}^{k+1} q'_i} \leq \frac{p'_{k+2}}{q'_{k+2}}$$

Thus the lemma is proved. ■

Lemma B.3. Let $Q \in \mathbb{R}^{d \times d}$ be a rate transition matrix, $A \in \mathbb{R}^{d \times d}$ be a permutation matrix, then AQA^{-1} is a rate transition matrix.

Proof. As every permutation matrix can be expressed as the products of elementary matrices, we denote:

$$A = \prod_{k=N_A}^1 T_{ij}^{(k)} = T_{ij}^{(N_A)} T_{ij}^{(N_A-1)} \dots T_{ij}^{(1)}$$

, where T_{ij} is the elementary matrix obtained by swapping row i and row j of the identity matrix, $N_A \in \mathbb{R}$

Therefore:

$$AQA^{-1} = \left(\prod_{k=N_A}^1 T_{ij}^{(k)} \right) Q \left(\prod_{k=1}^{N_A} T_{ij}^{(k)} \right)$$

For a single pair of transformation, i.e. $T_{ij}^{(k)} Q T_{ij}^{(k)}$, the row sums remain unchanged, and the diagonal elements is still the diagonal elements after transformation, thus AQA^{-1} is a rate transition matrix. \blacksquare

B.2 Proof of the theorem

Theorem B.4 (Accessibility). For two arbitrary discrete distributions $p, q \in \mathbb{R}^d$, there exists a rate transition matrix $Q \in \mathbb{R}^{d \times d}$ such that:

$$p = qe^Q$$

Proof. By Lemma 4.4, there exists permutation matrix A which satisfies inequality 15, and we denote:

$$p' := pA$$

$$q' := qA$$

Suppose:

$$Q := A Q' A^{-1}$$

$$, \text{ where } Q' = \begin{bmatrix} -\sum_{i=1}^{n-1} a_i & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} \\ 0 & -\sum_{i=2}^{n-1} a_i & a_2 & \dots & a_{n-2} & a_{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -a_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = U \Lambda U^{-1}, U \text{ is all one upper}$$

triangle matrix, and $\Lambda = \text{diag}(\{-\sum_{i=1}^{n-1} a_i, -\sum_{i=2}^{n-1} a_i, \dots, -a_{n-1}, 0\})$

Denote:

$$p'' := p'U = [p'_1, p'_1 + p'_2, \dots, \sum_{i=1}^{n-1} p'_i, 1]$$

$$q'' := q'U = [q'_1, q'_1 + q'_2, \dots, \sum_{i=1}^{n-1} q'_i, 1]$$

Thus the solution of $p = qe^Q$ can be obtained by solving:

$$p'' = q''e^\Lambda$$

, where $e^\Lambda = \text{diag}(\{e^{-\sum_{i=1}^{n-1} a_i}, e^{-\sum_{i=2}^{n-1} a_i}, \dots, e^{-a_{n-1}}, 1\})$ Solving the equation:

$$a_k = \ln \frac{\sum_{i=1}^{k+1} p'_i}{\sum_{i=1}^{k+1} q'_i} - \ln \frac{\sum_{i=1}^k p'_i}{\sum_{i=1}^k q'_i}$$

and specifically,

$$a_{n-1} = -\ln \frac{\sum_{i=1}^{n-1} p'_i}{\sum_{i=1}^{n-1} q'_i}$$

By the inequality 15 which p', q' satisfies and the monotonicity of the $\ln(\cdot)$ function, $a_k \geq 0, \forall k$, and thus Q' is a rate transition matrix

Transferring the solution of $p'' = q'' e^\Lambda$ back, we obtain:

$$Q = AU\Lambda U^{-1}A^{-1} = AQ'A^{-1}$$

and by Lemma 4.5, Q is a rate transition matrix. ■

C Proof of Supervision of Score-learning

Proposition C.1 (Supervision of Score-learning). *Suppose $Q^{(t)}$'s elements are non-zeros, the training objective is depicted as in Equation (8), then the optimality of the score model $s_{\theta^*}(x_t, t)_b$ satisfies:*

$$s_{\theta^*}(x_t, t)_y = \mathbb{E}_{x_0 \sim \mu_{0|t}(\cdot|x_t)} \left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right] = \frac{\sum_{x_0} \mu(x_0) p_{t|0}(y|x_0)}{\sum_{x_0} \mu(x_0) p_{t|0}(x_t|x_0)}$$

Proof.

$$\begin{aligned} J_{score} &= \int_0^T \mathbb{E}_{x_0 \sim \mu, x_t \sim p_{t|0}(x_t|x_0)} \left[\sum_{y \neq x_t} Q_{y,x_t}^{(t)} \left(s_{\theta}(x_t, t)_y - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right) \right. \\ &\quad \left. + \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \left(\log s_{\theta}(x_t, t)_y - \log \left(\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right) \right) \right] dt \end{aligned}$$

Therefore, with a little abuse of notation, we have

$$\begin{aligned} \arg \min_{\theta} J_{score} &= \arg \min_{\theta} \int_0^T \mathbb{E}_{x_0 \sim \mu, x_t \sim p_{t|0}(x_t|x_0)} \left[\sum_{b \neq x_t} Q_{y,x_t}^{(t)} \left(s_{\theta} - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \log s_{\theta} \right) \right] dt \\ &= \arg \min_{\theta} \underbrace{\int_0^T \mathbb{E}_{x_t \sim \mu_t} \left[\sum_{y \neq x_t} Q_{y,x_t}^{(t)} \left(s_{\theta} - \mathbb{E}_{x_0 \sim \mu_{0|t}} \left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right] \log s_{\theta} \right) \right] dt}_{\mathcal{L}} \\ \frac{\partial \mathcal{L}}{\partial s_{\theta}} &= \int_0^T \mathbb{E}_{x_t \sim \mu_t} \left[\sum_{y \neq x_t} Q_{y,x_t}^{(t)} \left(1 - \mathbb{E}_{x_0 \sim \mu_{0|t}} \left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right] \frac{1}{s_{\theta}} \right) \right] dt \end{aligned}$$

As $Q^{(t)}$'s elements are non zeros, therefore

$$\begin{aligned} Q_{y,x_t}^{(t)} &> 0, \forall y \neq x_t \\ \frac{\partial \mathcal{L}}{\partial s_{\theta}} &= 0 \iff 1 - \mathbb{E}_{x_0 \sim \mu_{0|t}} \left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right] \frac{1}{s_{\theta}} = 0 \end{aligned}$$

Therefore, the optimality of s_θ satisfies:

$$s_{\theta^*}(x_t, t)_y = \mathbb{E}_{x_0 \sim \mu_{0|t}(\cdot|x_t)} \left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \right]$$

Furthermore, as $\mu_{0|t}(x_0|x_t) = \frac{\mu(x_0)p_{t|0}(x_t|x_0)}{\sum_{x_0} \mu(x_0)p_{t|0}(x_t|x_0)}$, we have

$$s_{\theta^*}(x_t, t)_y = \frac{\sum_{x_0} \mu(x_0)p_{t|0}(y|x_0)}{\sum_{x_0} \mu(x_0)p_{t|0}(x_t|x_0)}$$

■

D Proof of Convergence

D.1 Proof of Lemmas

Lemma D.1. *For a random variable $X_0 \in \mathbb{R}^n$ with arbitrary two distributions p_0, p'_0 , the transition kernel is $p_{t|0}(x_t|x_0)$. We denote*

$$p_t(x_t) := \sum_{x_0} p_0(x_0)p_{t|0}(x_t|x_0)$$

$$p'_t(x_t) := \sum_{x_0} p'_0(x_0)p_{t|0}(x_t|x_0)$$

Then we have:

$$D_{KL}(p_t||p'_t) \leq D_{KL}(p_0||p'_0)$$

Proof.

$$\begin{aligned} D_{KL}(p_{0,t}(\cdot, \cdot)||p'_{0,t}(\cdot, \cdot)) &= \sum_{x_0, x_t} p_{0,t}(x_0, x_t) \log \frac{p_{0,t}(x_0, x_t)}{p'_{0,t}(x_0, x_t)} \\ &= \sum_{x_0, x_t} p_{0,t}(x_0, x_t) \log \frac{p_{t|0}(x_t|x_0)p_0(x_0)}{p_{t|0}(x_t|x_0)p'_0(x_0)} \\ &= D_{KL}(p_0||p'_0) \end{aligned}$$

Using the chain rule for KL divergence:

$$D_{KL}(p_t||p'_t) = D_{KL}(p_{0,t}(x_0, x_t)||p'_{0,t}(x_0, x_t)) - \mathbb{E}_{p_t}[D_{KL}(p_{0|t}(x_0|x_t)||p'_{0|t}(x_0|x_t))]$$

As KL divergence is greater than zero, we have:

$$D_{KL}(p_t||p'_t) \leq D_{KL}(p_{0,t}(x_0, x_t)||p'_{0,t}(x_0, x_t)) = D_{KL}(p_0||p'_0)$$

■

D.2 Proof of the theorem

Theorem D.2 (Convergence of the algorithm). *If we assume optimality is achieved in every epoch of the forward process and the reverse process, and we denote the k -th epoch estimation of μ as p_0 , then $\lim_{k \rightarrow \infty} D_{KL}(\mu||p_0^{(k)})$ converges.*

Proof. According to the assumption that each subprocess reaches its optimum,

$$\begin{aligned} \mu &= \mu p_{T|0}^{(k)} p_{0|T}^{(k); \leftarrow} \\ p_0^{(k+1)} &= p_0^{(k)} p_{T|0}^{(k)} p_{0|T}^{(k); \leftarrow} \end{aligned}$$

Therefore, by using Lemma D.1 twice:

$$D_{KL}(\mu||p_0^{(k)}) \geq D_{KL}(\mu p_{T|0}^{(k)}||p_0^{(k)} p_{T|0}^{(k)}) \geq D_{KL}(\mu p_{T|0}^{(k)} p_{0|T}^{(k); \leftarrow}||p_0^{(k)} p_{T|0}^{(k)} p_{0|T}^{(k); \leftarrow})$$

Therefore,

$$D_{KL}(\mu||p_0^{(k)}) \geq D_{KL}(\mu||p_0^{(k+1)})$$

As KL divergence is greater than zero, then

$$\lim_{k \rightarrow \infty} D_{KL}(\mu||p_0^{(k)})$$

converges. ■

E Derivation of Matrix Exponential Calculation

Proposition E.1. For a matrix $Q \in \mathbb{R}^{n \times n}$ and a non-degenerate matrix $D \in \mathbb{R}^{n \times n}$, we have:

$$\exp\{DQD^{-1}\} = D \exp\{Q\} D^{-1}$$

Proof. According to the definition of matrix exponential,

$$\exp\{DQD^{-1}\} = I + \sum_{i=1}^{\infty} (DQD^{-1})^i$$

As $(DQD^{-1})^i = DQ^i D^{-1}$,

$$\exp\{DQD^{-1}\} = I + \sum_{i=1}^{\infty} DQ^i D^{-1} = D(I + \sum_{i=1}^{\infty} Q^i) D^{-1} = D \exp\{Q\} D^{-1}$$
■

F Derivation of KL term calculation proposition

The full bound [8, 3] is as follows:

$$\mathbb{E}_{x_0 \sim \mu}[-\log p_{0;\theta}(x_0)] \leq J_{score} + \mathbb{E}_{x_0 \sim \mu}[D_{KL}(p_{T|0;\alpha}(x_T|x_0)||\phi)]$$

, where

$$J_{score} \triangleq \int_0^T \mathbb{E}_{x_0 \sim \mu, x_t \sim p_{t|0}(x_t|x_0)} \left[\sum_{b \neq x_t} Q_{b,x_t}^{(t)} \left(s_{\theta}(x_t, t)_b - \frac{p_{t|0}(b|x_0)}{p_{t|0}(x_t|x_0)} \right) + \frac{p_{t|0}(b|x_0)}{p_{t|0}(x_t|x_0)} (\log s_{\theta}(x_t, t)_b - \log(\frac{p_{t|0}(b|x_0)}{p_{t|0}(x_t|x_0)})) \right] dt$$

However, unlike previous works, the second term, which is the KL term should be considered, and it seems impossible to compute. Fortunately, certain characteristics of the *Matrix*-learning process can be used to justify a computable form for the second term. Suppose the text sequence holds d dimensions, *i.e.* $x \in \mathbb{R}^d$, then the characteristics can be described as follows:

- Independent Evolution:

$$p_{T|0;\alpha}(x_T|x_0) = \prod_{i=1}^d p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})$$

- Independent Terminal:

$$\phi(x_T) = \prod_{i=1}^d p_T(x_T^{(i)})$$

As a result, we provide a computable form for the KL term.

Proposition F.1. $D_{KL}(p_{T|0;\alpha}(x_T|x_0)||p_T(x_T)) = \sum_{i=1}^d D_{KL}(p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})||p_T(x_T^{(i)}))$

Proof. By independent evaluation and independent terminal, we have

$$\begin{aligned}
D_{KL}(p_{T|0;\alpha}(x_T|x_0)||p_T(x_T)) &= \sum_{x_T} p_{T|0;\alpha}(x_T|x_0) \log \frac{p_{T|0;\alpha}(x_T|x_0)}{\phi} \\
&= \sum_{x_T^{(1)}, x_T^{(2)}, \dots, x_T^{(d)}} p_{T|0;\alpha}(x_T|x_0) \sum_{i=1}^d \log \frac{p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})}{p_T(x_T^{(i)})} \\
&= \sum_{i=1}^d \sum_{x_T^{(1)}, x_T^{(2)}, \dots, x_T^{(d)}} p_{T|0;\alpha}(x_T|x_0) \log \frac{p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})}{p_T(x_T^{(i)})} \\
&= \sum_{i=1}^d \sum_{x_T^{(i)}} p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)}) \log \frac{p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})}{p_T(x_T^{(i)})} \\
&= \sum_{i=1}^d D_{KL}(p_{T|0;\alpha}(x_T^{(i)}|x_0^{(i)})||p_T(x_T^{(i)}))
\end{aligned}$$

■

G Additional Experimental details

G.1 Model Details

In terms of text modeling, for *Matrix*-learning, the Q_α matrix is initialized as follows:

$$\begin{aligned}
a_i &= 0, \forall i = 1, 2, 3, \dots, n-2 \\
a_{n-1} &= 1
\end{aligned}$$

The model is kept the same as SEDD [4].

As for image modeling, for *Matrix*-learning, the Q_α matrix is initialized as follows:

$$a_i = 1e-5, \forall i = 1, 2, 3, \dots, n-2$$

The model is kept the same as SEDD [4].

G.2 Training Details

The model is trained with a batch size of 512 and trained with a learning rate of 3×10^{-4} (Adam optimizer) on 8 4090 24GB GPUs. Both the *Matrix*-learning as well as the *Score*-learning are trained with the AdamW [50]. Training start with a weight decay factor 0.01, which then turn to 0 in the 7,900,000 step for text8.

H Limitations and Societal Impact

In this work, the **DMB** framework primarily relies on the evidence lower bound (ELBO) for both training and evaluation. However, given that Theorem 4.7 is not dependent on the specific form of the loss function, it is theoretically possible to derive other bounds for training. This flexibility opens new avenues for optimizing **DMB** under different theoretical and practical settings. Furthermore, we haven't provided a theorem focusing on optimality, which may be done for future work. As for societal impact, our work focus on foundation learning algorithms, which doesn't hold direct societal impact.