Diffusing Gaussian Mixtures for Generating Categorical Data

Florence Regol, Mark Coates
Dept. Electrical and Computer Engineering, McGill University
Montréal, QC, Canada
florence.robert-regol@mail.mcgill.ca, mark.coates@mcgill.ca

Abstract

Learning a categorical distribution comes with its own set of challenges. A successful approach taken by state-of-the-art works is to cast the problem in a continuous domain to take advantage of the impressive performance of the generative models for continuous data. Amongst them are the recently emerging diffusion probabilistic models, which have the observed advantage of generating high-quality samples. Recent advances for categorical generative models have focused on log likelihood improvements. In this work, we propose a generative model for categorical data based on diffusion models with a focus on high-quality sample generation, and propose sampled-based evaluation methods. The efficacy of our method stems from performing diffusion in the continuous domain while having its parameterization informed by the structure of the categorical nature of the target distribution. Our method of evaluation highlights the capabilities and limitations of different generative models for generating categorical data, and includes experiments on synthetic and real-world protein datasets.

Introduction

There are numerous applications for generative models of categorical random sequences; text generation, speech and music synthesis, drug design and protein synthesis are all important tasks that require modeling of high-dimensional nominal data. Learning the structure and substructure underlying those complex high-dimensional distributions can be useful for downstream tasks. For example, in drug synthesis, studies have confirmed the important role that mutational covariation plays in determining protein function, and this has found practical applications in drug design and drug resistance prediction (McGee et al. 2021; Tubiana, Cocco, and Monasson 2019; Socolich et al. 2005). As a result, recent works have employed generative models to learn from existing proteins and generate new ones (Trinquier et al. 2021; McGee et al. 2021; Jain et al. 2022). For this type of problem, the ability to generate quality samples is essential.

While the research on generative models for continuous data has been flourishing (see Bond-Taylor et al. 2022) for a review), the literature on modeling nominal categorical data is not as developed (Hoogeboom et al. 2021a).

Diffusing Gaussian Mixtures for Generating Categorical Data

Florence Regol, Mark Coates
Dept. Electrical and Computer Engineering, McGill University
Montréal, QC, Canada
florence.robert-regol@mail.mcgill.ca, mark.coates@mcgill.ca

Abstract

Learning a categorical distribution comes with its own set of challenges. A successful approach taken by state-of-the-art works is to cast the problem in a continuous domain to take advantage of the impressive performance of the generative models for continuous data. Amongst them are the recently emerging diffusion probabilistic models, which have the observed advantage of generating high-quality samples. Recent advances for categorical generative models have focused on log likelihood improvements. In this work, we propose a generative model for categorical data based on diffusion models with a focus on high-quality sample generation, and propose sampled-based evaluation methods. The efficacy of our method stems from performing diffusion in the continuous domain while having its parameterization informed by the structure of the categorical nature of the target distribution. Our method of evaluation highlights the capabilities and limitations of different generative models for generating categorical data, and includes experiments on synthetic and real-world protein datasets.

Introduction

There are numerous applications for generative models of categorical random sequences; text generation, speech and music synthesis, drug design and protein synthesis are all important tasks that require modeling of high-dimensional nominal data. Learning the structure and substructure underlying those complex high-dimensional distributions can be useful for downstream tasks. For example, in drug synthesis, studies have confirmed the important role that mutational covariation plays in determining protein function, and this has found practical applications in drug design and drug resistance prediction (McGee et al. 2021; Tubiana, Cocco, and Monasson 2019; Socolich et al. 2005). As a result, recent works have employed generative models to learn from existing proteins and generate new ones (Trinquier et al. 2021; McGee et al. 2021; Jain et al. 2022). For this type of problem, the ability to generate quality samples is essential.

While the research on generative models for continuous data has been flourishing (see Bond-Taylor et al. 2022) for a review), the literature on modeling nominal categorical data is not as developed (Hoogeboom et al. 2021a).

Autoregressive (AR) methods are well suited for modeling categorical data (Cooijmans et al. 2017). A notable class of AR models that give impressive performance for this problem are Transformers (Dai et al. 2019; Child et al. 2019; Hua et al. 2022; Jun et al. 2020). Transformers are powerful, but generally suffer from the weaknesses associated with autoregressive models; they are generally slow to train and slow to sample from (Bond-Taylor et al. 2022). They also suffer from quadratic complexity (w.r.t. sequence length), and because of their impressive flexibility in modeling capability, are harder to apply to smaller datasets (Lin et al. 2021). As a result, many works have attempted to linearize the time/memory complexity (Hua et al. 2022; Katharopou-
Discretization of continuous methods has been explored (Dinh, Sohl-Dickstein, and Bengio 2017; Ho et al. 2019; Theis, van den Oord, and Bethge 2016; Uria, Murray, and Larochelle 2013). However the modeling assumptions of these works are not suited to data that has no natural ordering of the categories. For the specific problem of nominal data generation, current state-of-the-art works are based on extending generative models that were initially developed for continuous data: normalizing flow (Ziegler and Rush 2019; Lippe and Gavves 2021; Hoogeboom et al. 2021b) and diffusion models (Hoogeboom et al. 2021a). Hoogeboom et al. report results indicating that the diffusion models can outperform Transformers. Diffusion probabilistic models (Sohl-Dickstein et al. 2015) are attractive for their generative capability. Compared to their competitors, such models have the characteristic of generating high quality samples and are relatively fast to train. The general trade-off is that they achieve lower likelihood and slower sampling (Bond-Taylor et al. 2022; Ho, Jain, and Abbeel 2020). As a result, substantial effort has been devoted to address these limitations (Nichol and Dhariwal 2021; Kingma et al. 2021; Xiao, Kreis, and Vahdat 2022; Salimans and Ho 2022).

In this work, we propose a generative model based on a diffusion process that can remain in the continuous space without sacrificing our knowledge that the data is nominal. To do so, we introduce a novel approach to encode nominal data in the continuous space via a sphere packing algorithm that places each category in the encoding space. We then incorporate the structural knowledge that follows from this construction into the denoising step of the diffusion using a Gaussian mixture conditioned on the current state of the diffusion. The advantages of such a design are threefold: 1) Unlike previous work (Hoogeboom et al. 2021a; Lippe and Gavves 2021; Hoogeboom et al. 2021b), this fixed encoding allows flexibility of the dimensionality of the representations without added complexity; 2) the structured denoising step requires significantly fewer diffusion steps, which greatly improves sampling time (which is identified as one of the main limitations of the diffusion model) while keeping the benefit of the diffusion model; 3) the generated samples are of higher quality.

Currently, the main method of evaluating a categorical generative model is via the log likelihood of held-out data. Although useful, this metric has some known drawbacks. (Theis, van den Oord, and Bethge 2016) use a simple example to show exactly how a good likelihood does not guarantee good sample generation. Proper evaluation of generative models is an ongoing research topic in many fields, including text, image, and graph generation (Garbacea et al. 2019; Celikyilmaz, Clark, and Gao 2020; Zhou et al. 2019; Borji 2019; Thompson et al. 2022; Theis, van den Oord, and Bethge 2016; Wu et al. 2017).

The general consensus has been to push towards a more comprehensive and task-oriented approach for assessing performance. Candidate metrics do not necessarily correlate with each other (Theis, van den Oord, and Bethge 2016; Zhou et al. 2019), so it can be important to measure performance in multiple ways. Indeed, failure to follow a comprehensive evaluation methodology has been linked to difficulties in assessing which models are actually better and to unexpected results (Caccia et al. 2020; Lucic et al. 2018; Rabanser, Günemann, and Lipton 2019). A notable example is the finding by (Nagarajan, Andreesen, and Neyshabur 2021) that high likelihood on a dataset and good sample generation does not guarantee good out-of-distribution detection capability, one of the candidate uses of a good generative model.

With these observations in mind, in this work, we expand on standard evaluation metrics to include distribution distance metrics. We propose a synthetic experiment with a known ground truth distribution to aid performance evaluation, with the goal of providing a more complete account of the generative capability of the models considered. To summarize, the major contributions of this paper are:

1. We introduce a novel procedure to represent nominal data in the continuous space based on sphere packing.
2. This allows us to design a novel denoising function tailored to model nominal data in the continuous space.
3. Our presented model offers state-of-the-art sample generation quality and is efficient in both sampling time and training time, as demonstrated by our experiments on both synthetic datasets and on protein datasets.

**Related Work**

Early approaches to handle the related problem of discrete data generation were based on dequantization and thresholding. The overall idea is to add noise to the discrete point and treat it as a continuous generative modeling problem, and then use thresholding to generate samples (Ho et al. 2019; Theis, van den Oord, and Bethge 2016; Dinh, Sohl-Dickstein, and Bengio 2017). Current state-of-the-art methods avoid injecting an arbitrary ordering to the categories by either adapting the methodology to stay in the categorical domain, or modelling the data using a latent representation in the continuous space that can be later mapped to the categorical space. In (Ziegler and Rush 2019) and (Lippe and Gavves 2021), normalizing flows (NF) are used to model such a latent representation. An encoder-decoder framework is used to map from the categorical to the continuous space and vice versa. The overall model is learned through variational inference. (Hoogeboom et al. 2021b) build on the same idea as (Lippe and Gavves 2021), but rather than learning the encoder/decoder, they fix the decoder with an argmax function. This induces a constraint on the functional space of the encoder that is maintained throughout training. Both of these state-of-the-art works keep the mapping from the continuous to the categorical space simple. In (Hoogeboom et al. 2021a) this is done by using a fixed deterministic argmax function, and (Lippe and Gavves 2021) experimented with learning the encoder/decoder of varying complexity and found that a simple parameterization of the mean and variance gave the optimal result. Unlike our approach, once this mapping is done, nothing informs the NF that it is treating a latent representation of a categorical variable.
Moving away from the normalizing flow methods, (Hoogeboom et al. 2021a) also presented a diffusion-based model that operates directly on the categorical space. Instead of diffusing the signal with Gaussian distributions and learning means and variance of parameterized Gaussian as denoising process, they diffuse a one-hot encoded sequence with a multinomial categorical distribution. As is the case for the argmax, the dimension of the sequence representation scales linearly with the number of categories. Other related work that takes a similar approach to us by mapping to an alternative space to perform diffusion includes (Valdat, Kreis, and Kautz 2021) and (Sinha et al. 2021). These works tackle the tangential problems of generating ordinal data and conditional generative modeling.

Lastly, related works that target a similar task connected to generating quality includes (Jain et al. 2022; Brookes, Park, and Listgarten 2019; Kumar and Levine 2020; Hoffman et al. 2022). This literature focuses on generating high score protein sequences, which are evaluated by an oracle. Even though these models are generative in nature, the end task is still somewhat supervised. The models explicitly aim to maximize a quantity, whereas for our purposes we remain in the traditional generative modeling problem formulation of learning a distribution.

Methodology

Problem Setting. Consider a categorical multivariate random variable \(X = [X_1, \ldots, X_{|S|}]\) where each element belongs to one of \(K\) categories: \(X_{(j)} \in C, C = \{C_1, \ldots, C_K\}\) with associated pmf \(p(X)\). Given a dataset of realizations \(D = \{x_i\}_{i=1}^N, x_i \sim X\), the task is to learn \(p(X)\).

Encoding the categorical sequences and sphere packing. We lift the problem to the continuous space by introducing a latent continuous random variable \(Z^0\) that is mapped from and to the categorical sequence \(X \in C^{|S|}\) with an encoder, \(q(Z^0|X)\), and decoder, \(p(X|Z^0)\), respectively. The log likelihood and its variational lower bound are given by:

\[
\log p(X) = \log \int p(X, Z^0) q(Z^0|X) dZ^0, \\
\log p(X) \geq \mathbb{E}_{q(Z^0|X)} \left[ \log \left( \frac{p(X|Z^0)}{q(Z^0|X)} \right) \right] + \log \left( \frac{p(X)}{\mathbb{E}_{q(Z^0|X)} \left( \frac{p(X|Z^0)}{q(Z^0|X)} \right)} \right). \tag{1}
\]

It is desirable to focus complexity into learning \(p(Z^0)\), so we make the mappings from \(Z^0\) to \(X\) simple and tractable. Consequently, we use a fixed, factorized encoding distribution to associate each categorical element \(X_{(j)}\) of the sequence with a random vector in a \(d\)-dimensional continuous space \(Z^0_{(j)} \in \mathbb{R}^d\). The mapping depends on the category; each category \(C_k\) is assigned a distribution \(f(\cdot; \mu_{C_k}, \sigma)\) that is clearly distinguishable from others by its mean \(\mu_{C_k} \in \mathbb{R}^d\) and variance \(\sigma^2 \in \mathbb{R}\). We use a Gaussian \(f(\cdot)\) for simplicity, and similarly to (Lippe and Gavves 2021) we obtain the decoder \(p(X|Z^0)\) through Bayes’ rule, so we have:

\[
q(Z^0|X) = \prod_{s=1}^S \mathcal{N}(Z^0_{(s)}; \mu_{X_{(s)}}, \sigma^2) \quad \text{as the encoder and}
\]

\[
p(X|Z^0) = \prod_{s=1}^S \mathcal{N}(Z^0_{(s)}; \mu_{X_{(s)}}, \sigma^2) \quad \text{as the decoder.}
\]

(The prior on \(X\) does not appear as we assume uniformity). The advantages are twofold: 1) it imposes a structure on the target distribution \(p(Z^0)\) that can be used in modeling the learnable \(p_0(Z^0)\), as we will show shortly; and 2) it simplifies the learning objective since only \(p_0(Z^0)\) is learnable.

Our aim is to make it as easy as possible for the decoder to distinguish between categories. This implies that we should strive to identify maximally separated means. This leads to a sphere packing problem — finding the placement of \(K\) points on the surface of a \(d\)-dimensional sphere \(S^d(1)\) that maximizes the minimum distance between any two points:

\[
\mu_1, \ldots, \mu_K = \arg \max_{\mu_1, \ldots, \mu_K \in \mathbb{B}(1)} \min_{i \neq j} ||\mu_i - \mu_j||_2^2
\]

Hence we can use solutions of this problem, e.g., (Gamal et al. 1987), to 1) set the means of the encoding distributions \{\(\mu_{C_{k}}\)\}_{k=1}^K; and 2) determine, based on the minimum distance \(d_{\mu^*} = \min_{i \neq j} ||\mu_i - \mu_j||_2^2\), a value for the variance \(\sigma^2\) such that the Gaussian distributions \(\mathcal{N}(\mu_{C_k}, \sigma^2)\); \(k \in [K]\) have limited overlap but are not too concentrated. Denoting \(d_{\mu^*} = \min_{i \neq j} ||\mu_i - \mu_j||_2^2\), we have:

\[
\mu_{C_k} = \mu^*_k; \quad k \in [K] \quad \text{and} \quad \sigma = \frac{d_{\mu^*}}{2K} \sqrt{3} \tag{2}
\]

Almost all (99.7%) of the mass of a \(d\)-dimensional m.v. Gaussian R.V. is within \(\sqrt{3}\) standard deviations, so we set \(\sigma\) to half that radius, and divide by the number of categories.

Learning the latent distributions \(p_0(Z^0)\). The complex correlation structure of the categorical distribution must be captured in \(p_0(Z^0)\). We propose to use a diffusion probabilistic model (DPM) (Sohl-Dickstein et al. 2015) with a novel denoising component, tailored to our encoding scheme and categorial data, based on Gaussian Mixtures. The DPM introduces \(T\) latent random variables \(Z^1, \ldots, Z^T\). Commencing with the targeted encoded sequence \(Z^0\), the variables are derived by gradually adding known Gaussian noise of increasing variance to the variable from the previous timestep: \(n(Z^t|Z^{t-1}) = \mathcal{N}(Z^t; \sqrt{1-\beta_t}Z^{t-1}, \beta_t I)\); \(\beta_t < \beta_{t+1} \in (0, 1)\). At the end of the chain, only noise should remain \(Z^T \sim \mathcal{N}(Z^T; 0, 1)\). The task of the DPM is to learn the denoising process \(d_0(Z^{t+1}|Z^t); t \in [T]\).

This leads to construction of the generative model for \(Z^0\):

\[
p_0(Z^0) \equiv \mathbb{E}_{d_0(Z^T)} = \int d(Z^T) \prod_{t=1}^T d_0(Z^{t+1}|Z^t) dZ^{1:T}. \tag{3}
\]

See (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) for more detailed discussion of the diffusion process.

Exploiting the structure. In most denoising approaches, the distributions \(d_0(Z^{t+1}|Z^t)\) are modelled as normal distributions with learnable means and (usually fixed) variances.
In our case, we take advantage of the known structure. By our construction, the target distribution is a mixture of Gaussians; conditioned on knowledge of the target sequence, the distribution \( p(Z^{t-1} | Z^t, X) \) is Gaussian, and the mean and variance can be evaluated analytically.

If we are at a point in the chain \( Z^t \), then if we are given an element of the sequence \( x_{(s)}, Z^t_{(s)} \) is conditionally independent of other \( Z^t_{(s-1)} \), and we can derive the conditional of the next denoising step in closed-form:

\[
p(Z^{t-1}_{(s)} | Z^t_{(s)}, x_{(s)}) = \int p(Z^{t-1}_{(s)} | Z^t_{(s)}, z^0_{(s)}) p(Z^t_{(s)} | x_{(s)}) dz^0_{(s)} = \mathcal{N}(Z^{t-1}_{(s)}; \mu_{Z^t_{(s)}}, \sigma^2_{Z^t_{(s)}})\]

where \( \mu_{Z^t_{(s)}} = \frac{\alpha_{t-1} \beta_t}{1 - \alpha_t} \mu_{x_{(s)}} + \frac{\sqrt{\alpha_t (1 - \alpha_{t-1})}}{1 - \alpha_t} Z^t_{(s)}, \sigma^2 = \frac{1 - \alpha_{t-1}}{1 - \alpha_t} \beta_t + \left( \frac{\sqrt{\alpha_t (1 - \alpha_{t-1})}}{1 - \alpha_t} \right)^2. \)

(See the supplementary for the detailed derivation.) Hence if we have a predictor \( p_d(Z^{t-1}, t) \) of the distribution of the sequence \( X \) based on the current state \( Z^t \) and the diffusion step \( t \), we can model the denoising step as:

\[
d_d(Z^{t-1}, t) = \sum_{X \in \mathcal{S}} \left( \prod_{s=1}^S p(Z^{t-1}_{(s)} | Z^t_{(s)}, X_{(s)}) \right) p_d(X | Z^t, t). \]

(5)

If \( p_d(X | Z^t, t) \) is structured to assume independence among the elements of \( X \), we can factorize:

\[
p_d(X | Z^t, t) = \prod_{s=1}^S p_d(X_{(s)} | Z^t_{(s)}, t) \quad \text{and write:}
\]

\[
d_d(Z^{t-1}, t) = \sum_{X \in \mathcal{S}} \left( \prod_{s=1}^S \prod_{k=1}^K p(Z^{t-1}_{(s)} | Z^t_{(s)}, C^t_k) p_d(X_{(s)} | Z^t_{(s)}, t) \right) \\
= \prod_{s=1}^S d(s, \theta) p_d(Z^{t-1}_{(s)} | Z^t_{(s)}). \]

(6)

Replacing the Gaussian denoising term used in (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) with this more complex denoising model results in a more involved loss expression, but the denoising process can be successful with far fewer diffusion steps (10-40 versus thousands). This effect was also observed in (Xiao, Kreis, and Vahdat 2022).

**Loss objective.** Since the encoder and decoder are fixed, optimization of the loss function (Eqn. (1)) simplifies to:

\[
\theta^* = \arg \max_{\theta \in \Theta} \mathbb{E}_q(z^0 | x) \left[ \log p_d(z^0) \right],
\]

i.e., the log likelihood of the diffusion model under the expectation of the decoder. Since the DPM is a latent variable model, its log likelihood is also optimized via a lower bound:

\[
\log \left( p_d(z^0) \right) \geq \mathbb{E}_n \left[ - \log \frac{n(Z^T | z^0)}{d(Z^T)} \right] \\
= \sum_{t=2}^T \log \frac{n(Z^{t-1} | Z^t, z^0)}{d_d(Z^{t-1} | Z^t)} + \log d_d(z^0 | z^0).
\]

Employing this bound in Eqn. (7), substituting with (6), and removing terms that are independent of \( \theta \), we can identify the final optimization task:

\[
\theta^* = \arg \max_{\theta \in \Theta} \mathbb{E}_q(z^0, n) \left[ - \sum_{t=2}^T \mathcal{L}_{t-1} + \mathcal{L}_0 \right]
\]

(8)

where \( \mathcal{L}_{t-1} = KL \left( n \left( Z^{t-1} | Z^t, z^0 \right) | | d_d(Z^{t-1} | Z^t) \right) \)

and \( \mathcal{L}_0 = \log d_d (z^0 | z^0) \).

**Architecture and training.** In practice, it has been shown beneficial for this type of loss to randomly optimize one of the terms \( \mathcal{L}_t \) at a time (Ho, Jain, and Abbeel 2020) (Nichol and Dhariwal 2021). The objective then becomes to either maximize the log likelihood of the final step for \( t = 0 \), or to minimize the KL divergence between a Gaussian mixture with learnable mixture weights for time step \( t > 0 \):

\[
\mathcal{L}_{t-1} = KL \left( n \left( Z^{t-1} | Z^t, z^0 \right) | | d_d(Z^{t-1} | Z^t) \right) \\
= \sum_{s=1}^S KL \left( n \left( Z^{t-1}_{(s)} | Z^t_{(s)}, z^0_{(s)} \right) | | d_d(Z^{t-1}_{(s)} | Z^t_{(s)}) \right)
\]

Using the variational approximation of the KL divergence between Gaussian mixtures from (Hershey and Olsen 2007), we can approximate the individual step loss as follow:

\[
\mathcal{L}_{t-1} \approx - \sum_{s=1}^S \log \sum_{k=1}^K p_d(X_{(s)} = C^t_k | z^0_{(s)}, t) w^t_{z^0, o}(C_k)
\]

where \( w^t_{z^0, o}(C_k) = \exp \left( -KL \left( n(|z^0_{(s)}, z^0_{(s)} | | N(\mu_{C^t_k}^{(s)}, \sigma_{C^t_k}^{(s)}) \right) \right). \)

Details of the derivation are provided in the supplementary.

At this point, we can see that the optimization of this term is reached when \( p_d(X_{(s)} | Z^t_{(s)}, t) \) gives maximum weight to the highest term of the sum \( w^t_{z^0, o}(C_k) \), which is the initial sequence \( C^t_k = x_{(s)} \). As a result, we approximate this optimization by maximizing the log likelihood of \( p_d(X_{(s)} = x_{(s)} | Z^t_{(s)}, t) \), as both isolated optimization problems have the same solution:

\[
\arg \max_{\theta \in \Theta} \mathcal{L}_{t-1} \approx \arg \max_{\theta \in \Theta} \log p_d(X = x | Z^t, t)
\]

(9)

As a result, learning hinges on the modeling capability of \( p_d(X | Z^t, t) \). We employ a transformer-based architecture. The vector \( z^t \) and an embedding of time \( t \) serve as inputs. We adopt a sampling approach for the training. For each sequence in the training data, we sample \( z^0 \sim q(|z^0 | X = x) \), and then draw a time \( t \), we sample \( z^t \sim n(Z^t | z^0) \) to evaluate the loss. It is important to emphasize that this transformer does not have an autoregressive structure — all elements of a sequence are generated in parallel. The correlations are induced by the denoising diffusion process.

**Data Augmentation.** In practice, we observe that \( p_d(X | Z^t, t) \) learns to be increasingly certain of its prediction as we approach the end of the chain \( Z^t \). This behavior can be seen in Figure 2 where we show an example of the entropy at every time step \( H(p_d(X | Z^t, t), \ldots, H(p_d(X | Z^t, 1)) \).

We can imagine that alongside the gradually noisy \( Z^t \), there is also a corresponding noisy categorical sequence \( X^t \).
that \( p_\theta(X|Z^t,t) \) aims to predict. As a result, instead of training on the ground truth sequence at the beginning of the diffusion \( T; T−1,\ldots \), we inject some noise by training on a “diffused” version of \( x \), denoted by \( \tilde{x}^t \), and thus modify Eqn. (9) to:

\[
\arg\max_{\theta \in \Theta} \mathcal{L}_{t−1} \approx \arg\max_{\theta \in \Theta} \log p_\theta(X = \tilde{x}^t|Z^t, t) \tag{10}
\]

\( \tilde{x}^t \sim p^{x^t:t}(\tilde{X}) \), where \( p^{x^t:t}(\tilde{X}) = \frac{(w^{Z^{t:0}}(x(s)))^2}{\sum_{k=1}^{K}(w^{Z^{t:0}}(C_k))^2} \)

Algorithms detailing the training and sampling procedures are provided in the supplementary.

**Experiments**

In this section, we first present the evaluation metrics, the datasets and the experimental set-up. We then report the performance of our proposed GMCD model and conduct ablation studies to validate the effectiveness of its modules.

**Evaluation metrics.** Many of the difficulties and limitations associated with evaluating generative models stem from the fact that we do not have access to the ground truth distribution. With access to ground truth, the problem formulation changes and the previously mentioned problems associated with log likelihood (LL) and sampled-based metrics disappear. Instead of:

• Maximizing the LL of unseen samples \( \rightarrow \) we aim to assign the correct probability mass to unseen samples,

• Generating “good” samples \( \rightarrow \) we aim to generate samples that are distributed according to the ground truth,

• Maximizing a heuristic for sample quality (novelty, diversity, etc.) \( \rightarrow \) we aim to generate samples with the same heuristic value as the expected value from ground truth.

In this work, we are interested in evaluating how close a generative model is to the true probability measure based on its samples in the discrete domain.

**With known ground truth distribution.** The distance between two distributions \( p, q \) on a discrete sample space \( \Omega \) can be measured by the total variation and Hellinger distances:

\[
d_{TV}(p, q) \triangleq \frac{1}{2} ||p - q||_1 = \frac{1}{2} \sum_{x \in \Omega} |p_x - q_x|, \\
Hel(p, q) \triangleq \frac{1}{\sqrt{2}} ||\sqrt{p} - \sqrt{q}||_2 = \frac{1}{\sqrt{2}} \sqrt{\sum_{x \in \Omega}(\sqrt{p_x} - \sqrt{q_x})^2}.
\]

(with \( p_x \) used as a shorthand for \( p(x) \)). These are principled metrics but they can rapidly become impractical as \( \Omega \) grows, especially as we must usually rely on samples to estimate \( p_x \).

Alternatively, we can consider a partitioning of the sample space: \( \mathcal{P} = \{A_1; A_i \subset \Omega, A_i \cap A_j = \emptyset\} \) and estimate the probability mass of these events \( p(A_i) = \sum_{x \in A_i} p_x \). It is less precise but can be more informative if \( \Omega \) is large and/or if the partitioning has a particular meaning. One obvious partitioning of interest would be to divide the sample space into positive-support elements (in distribution - ID) and the zero support elements (out-of-distribution - OOD); the partitioning is then \( \mathcal{P}^{id} = \{A_o, A_o\} \) where \( \{x \in A_o; p_x > 0, x \in \Omega\}, \{x \in A_o; p_x = 0, x \in \Omega\} \).

As our focus is on sample quality, we compare the ground truth distribution \( p \) to the empirical distribution \( \hat{p}_0 \) constructed from the samples of a generative model. For the synthetic experiments where we have access to \( p \), we report:

• \( Hel(p, \hat{p}_0) \) and \( d_{TV}(p, \hat{p}_0) \),

• \( d_{TV+} \triangleq \frac{1}{2} \sum_{x \in A_o} |p_x - \hat{p}_0 x|, d_{TV−} \triangleq \frac{1}{2} \sum_{x \in A_o} |p_x - \hat{p}_0 x| \),

• \( \hat{p}_0(A_i) = \sum_{x \in A_i} \hat{p}_0 x ; \) prob. estimates of valid sequences,

• \( \hat{p}_0(A_i) = \sum_{x \in A_i} \hat{p}_0 x ; \) prob. estimates of specified \( A_i \).

**Without ground truth distribution.** In practice, \( p \) is not available. We still focus on generating samples that are representative of the distribution by comparing statistics of the ground truth distribution with those derived from generated samples. A major capability of interest of a generative model is its ability to properly capture patterns in the data; as such we can compare the higher order covariation of patterns of a generated set of samples to that of a test set. Such evaluation metrics are commonly used in the generative protein sequence modeling literature (Trinquart et al. 2021; McGee et al. 2021). Given a pattern of size \( p \), described by positions and corresponding categories \( \{s_1, \ldots, s_p\}, \{k_1, \ldots, k_p\} \), and a set of \( M \) sequences \( X^M \), the higher order pattern covariation \( C_{k_1, \ldots, k_p}(X^M) \) is the frequency of the appearance of the pattern in \( X^M \) minus the product of the frequencies of each individual element of the pattern:

\[
\hat{f}^{s_1, \ldots, s_p}_{k_1, \ldots, k_p}(X^M) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}[x_{s_1} = k_1, \ldots, x_{s_p} = k_p], \\
C^{s_1, \ldots, s_p}_{k_1, \ldots, k_p}(X^M) = \hat{f}^{s_1, \ldots, s_p}_{k_1, \ldots, k_p}(X^M) - \prod_{j=1}^{p} \hat{f}^{s_j}_{k_j}(X^M). \tag{11}
\]

For a given pattern length \( p \), we select a random subset of all possible patterns \( \{pattern_1, \ldots, \} \) by following the procedure described in (McGee et al. 2021), which focuses on the most likely patterns (the detailed selection procedure is described in the supplementary).
report the Pearson correlation $\rho^P$ between the pattern higher order covariations computed on the test set $C_p = \{C_{pattern}^p(x), \ldots \}$, $x \sim D$ and the set of generated samples $C_{p_0} = \{C_{pattern}^p(x), \ldots \}$, $x \sim \tilde{p}_0$.

**Datasets**

We design a ground truth distribution to generate a synthetic dataset of sequences of length $S = K$. We define the sample space $\Omega^K = \mathbb{C}^K$ and only assign probability mass on permutations of $C$, i.e., $A_+ = \{x; x(i) \neq x(j) \forall i \neq j \}$. Finally, we separate the positive sets in two and assign $3$ times more mass to sequences with a “smaller” category at the start of the sequence than at the end, i.e.: $p(x) = \begin{cases} \frac{3}{2K} & \text{if } x \in A_{\text{likely}} = \{x; x \in A_+ \land x(i) < x(s) \} \\ \frac{2}{2K} & \text{if } x \in A_{\text{rare}} = \{x; x \in A_+ \land x(i) > x(s) \} \\ 0 & \text{otherwise} \end{cases}$.

This synthetic dataset is designed to emulate characteristics of a real world dataset. In practice, the distributions that we wish to model are likely to have positive support on a very small fraction of the probability mass. Whether we are trying to generate text, images or proteins, the likelihood of stumbling across a “valid” sample when drawing from a uniform distribution is extremely small.

Natural partitionings of interest for this type of dataset are: 1) $\mathcal{P}^{od}$ as previously described where we can see a model’s ability to grasp the positive support of the sample space; and 2) $\mathcal{P} = \{A_{\text{likely}}, A_{\text{rare}}, A_+\}$ where we can see a model’s ability to assign the right amount of probability mass to the different sets.

| $K = 6$ | Hcl | dRV | (dRV) | dRV$|$ | $p(A_{\text{likely}})$ | $p(A_{\text{rare}})$ | $p(A_+)$ |
|---------|-----|-----|-------|-------|----------------|----------------|--------|
| CNF+    | 37.02 | 34.63 | 25.59 | 9.04  | 40.91          | 41.01          | 81.92   |
| argmaxAR+ | 24.22 | 17.90 | 13.55 | 4.35  | 66.32          | 24.97          | 91.29   |
| CDM     | 19.27 | 16.19 | 14.34 | 1.84  | 66.33          | 29.98          | 96.31   |
| GMCD    | 16.62 | 16.07 | 15.56 | 0.51  | 75.71*         | 23.27*         | 98.98*  |

| $K = 8$ | Hcl | dRV | (dRV) | dRV$|$ | $p(A_{\text{likely}})$ | $p(A_{\text{rare}})$ | $p(A_+)$ |
|---------|-----|-----|-------|-------|----------------|----------------|--------|
| CNF+    | 81.86 | 83.63 | 69.79 | 13.84 | 35.98          | 36.34          | 72.32   |
| argmaxAR+ | 73.48 | 76.06 | 73.19 | 2.87  | 66.94          | 27.32          | 94.27   |
| CDM     | 73.65 | 76.36 | 73.81 | 2.55  | 62.70          | 32.20          | 94.90   |
| GMCD    | 72.30 | 74.98* | 73.34 | 1.64  | 71.77*         | 24.94*         | 96.71*  |

| $K = 10$ | Hcl | dRV | (dRV) | dRV$|$ | $p(A_{\text{likely}})$ | $p(A_{\text{rare}})$ | $p(A_+)$ |
|---------|-----|-----|-------|-------|----------------|----------------|--------|
| CNF+    | 98.28 | 99.81 | 83.43 | 16.38 | 33.64          | 33.59          | 67.23   |
| argmaxAR+ | 98.43 | 99.81 | 76.95 | 22.86 | 40.38          | 13.89          | 54.27   |
| CDM     | 97.32 | 99.69 | 97.39 | 2.30  | 64.62          | 30.77          | 95.40   |
| GMCD    | 97.27* | 99.68* | 97.71 | 1.98  | 66.99*         | 29.05*         | 96.05*  |

Table 1: Distances metrics and probability estimates for partitionings $\mathcal{P}, \mathcal{P}^{od}$ for the synthetic datasets. * indicates significance w.r.t. to the Wilcoxon signed-rank test at the 5% level. + indicates that more epochs were required to reach competitive results.

We consider a small scale experiment $K = 6$ where the models are exposed to the entire ID set $A_+$ multiple times, a medium scale experiment $K = 8$ where the models are exposed to a sizeable fraction of the ID set, and a larger scale experiment $K = 10$ where the models are exposed to less than 1% of $A_+$ (see Table ?? for additional details).

As a real world application, we measure the performance of the models on two protein datasets from the Pfam protein family: PF00014, which contains $N = 137,605$ proteins of length $S = 70$ and PF00016, which contains $N = 13,600$ proteins of length $S = 53$. The number of categories for both datasets corresponds to the list of amino acids $K = 21$.

**Experiment Details**

**Baselines.** We compare our GMCD approach to three state-of-the-art baselines: 1) CNF (Lippe and Gavves 2021), a normalizing flow method that learns a mapping to/from the categorical space; 2) CDM (Hoogeboom et al. 2021a), a diffusion-based model; and 3) argmaxAR (Hoogeboom et al. 2021b), a normalizing flow method that uses an argmax operation to map to the discrete space. We select the autoregressive version because it was reported as the best alternative.

**Experimental set-up.** We train all models using the RAdam optimizer (Liu et al. 2020) and early stopping and keep the best model evaluated on the validation set. For the proteins dataset and for the large scale synthetic experiment $K = 10$, in order to avoid overfitting, we monitor to ensure that the model is not reproducing more than 1% of the training dataset in its generated samples. Performance metrics are averaged over 10 trials of $M = 10,000$ generated samples. A split of 70/20/10 is used for the protein datasets. The $p_0(X|Z^t, t)$ function is modeled using a non-autoregressive transformer similar to that used in (Hoogeboom et al. 2021a). Following (Ho, Jain, and Abbeel 2020), we use sinusoidal position embedding to process the time step $t$ and concatenate it to $Z^t$ to form the input to the transformer. The means $\mu_1, \ldots$ are computed using the procedure from (Gamal et al. 1987), which employs simulated
Table 3: Proteins experiment results. – indicates that the Pearson coefficient was not significant at the 5% level.

<table>
<thead>
<tr>
<th>Method</th>
<th>ρ^2</th>
<th>ρ^4</th>
<th>ρ^4</th>
<th>ρ^6</th>
<th>ρ^7</th>
<th>ρ^8</th>
<th>ρ^9</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PF00076</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>argmaxAR</td>
<td>73.13</td>
<td>73.00</td>
<td>69.85</td>
<td>63.88</td>
<td>58.74</td>
<td>49.08</td>
<td>49.74</td>
</tr>
<tr>
<td>CDM</td>
<td>82.30</td>
<td>82.44</td>
<td>80.48</td>
<td>78.08</td>
<td>74.95</td>
<td>73.10</td>
<td>75.56</td>
</tr>
<tr>
<td>GMCD</td>
<td><strong>84.10</strong></td>
<td><strong>82.85</strong></td>
<td><strong>82.36</strong></td>
<td><strong>81.04</strong></td>
<td><strong>77.67</strong></td>
<td><strong>78.09</strong></td>
<td><strong>78.95</strong></td>
</tr>
<tr>
<td><strong>PF00014</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>argmaxAR</td>
<td>78.06</td>
<td>79.05</td>
<td>80.89</td>
<td>83.57</td>
<td>84.97</td>
<td><strong>88.51</strong></td>
<td>91.26</td>
</tr>
<tr>
<td>CDM</td>
<td><strong>81.28</strong></td>
<td><strong>80.38</strong></td>
<td><strong>78.98</strong></td>
<td><strong>78.76</strong></td>
<td><strong>77.35</strong></td>
<td>80.40</td>
<td>85.31</td>
</tr>
<tr>
<td>GMCD</td>
<td>80.41</td>
<td>80.81</td>
<td><strong>82.01</strong></td>
<td><strong>84.09</strong></td>
<td><strong>85.83</strong></td>
<td>88.39</td>
<td><strong>91.50</strong></td>
</tr>
<tr>
<td>abl.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMCD random</td>
<td>69.21</td>
<td>69.04</td>
<td>71.69</td>
<td>78.81</td>
<td>80.26</td>
<td>87.65</td>
<td>90.54</td>
</tr>
<tr>
<td>GMCD sharp</td>
<td>79.86</td>
<td>79.13</td>
<td>81.07</td>
<td>82.96</td>
<td>83.85</td>
<td>85.26</td>
<td>88.78</td>
</tr>
</tbody>
</table>

Table 4: Pattern covariance metrics.

In the covariance pattern metrics (Table 4). The decomposi-
set for a total size datasets. We generate 10K sequences for the train/valid/test
<table>
<thead>
<tr>
<th>K</th>
<th>ρ^2</th>
<th>ρ^4</th>
<th>ρ^4</th>
<th>ρ^6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>K = 6</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>argmaxAR+</td>
<td>63.16</td>
<td>58.38</td>
<td>59.22</td>
<td>63.66</td>
</tr>
<tr>
<td>CNF+</td>
<td>-12.97</td>
<td>14.61</td>
<td>-5.05</td>
<td>-21.10</td>
</tr>
<tr>
<td>CDM</td>
<td>54.00</td>
<td>58.26</td>
<td>59.07</td>
<td>63.52</td>
</tr>
<tr>
<td>GMCD</td>
<td><strong>64.03</strong></td>
<td><strong>63.88</strong></td>
<td><strong>66.22</strong></td>
<td><strong>67.64</strong></td>
</tr>
<tr>
<td><strong>K = 8</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNF+</td>
<td>-10.13</td>
<td>3.20</td>
<td>-1.31</td>
<td></td>
</tr>
<tr>
<td>CDM</td>
<td>20.19</td>
<td>16.61</td>
<td>12.03</td>
<td>6.98</td>
</tr>
<tr>
<td>GMCD</td>
<td><strong>32.70</strong></td>
<td><strong>26.83</strong></td>
<td><strong>16.31</strong></td>
<td><strong>10.71</strong></td>
</tr>
<tr>
<td><strong>K = 10</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>argmaxAR+</td>
<td>11.85</td>
<td>6.54</td>
<td>4.57</td>
<td>1.55</td>
</tr>
<tr>
<td>CNF+</td>
<td>-</td>
<td>-4.68</td>
<td>-1.62</td>
<td></td>
</tr>
<tr>
<td>CDM</td>
<td>22.46</td>
<td>13.40</td>
<td>7.80</td>
<td>4.55</td>
</tr>
<tr>
<td>GMCD</td>
<td><strong>25.19</strong></td>
<td><strong>18.67</strong></td>
<td><strong>6.60</strong></td>
<td><strong>4.79</strong></td>
</tr>
</tbody>
</table>

Table 5: Size of the sample space |Ω|, of the positive support set |A_{+}|, and the fraction of valid sequences contained in the training set of the synthetic
<table>
<thead>
<tr>
<th>S, K</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6^6 = 46,566</td>
<td>6^6 = 720</td>
<td>100%</td>
</tr>
<tr>
<td>8</td>
<td>8^8 = 16,777,216</td>
<td>8^8 = 40,320</td>
<td>21.34%</td>
</tr>
<tr>
<td>10</td>
<td>10^{10} = 3,628,800</td>
<td>0.28%</td>
<td></td>
</tr>
</tbody>
</table>

almost all valid (p(A_{+}) is close to 100). Its deficiency is in
assigning a probability mass ratio of approximately 2:1 to the
two sets A_{likely} and A_{rare}. This results in higher sta-
tistical distance metrics d_{TV} and H_{el}. argmaxAR struggles
to identify A_{+} and requires additional training to reach a

Ablation study and Time Analysis. We report abla-
tion studies to verify the relative contribution of two model
components. We compare with a GMCD version with no

Conclusion

In conclusion, we introduced the GMCD model; a continu-
ous diffusion-based model for nominal data. We introduced
a novel novel fixed encoding procedure to map categorical
data to the continuous space and gain representation flexi-
bility. This also leads to a novel continuous denoising pro-
cess that is cognizant of the categorical nature of the targeted
distribution. The GMCD is fast to train, fast to sample from
and generates representative samples of the ground truth dis-
tribution as demonstrated on synthetic and on a real world
datasets.
References


