Improving Discrete Diffusion Models via Structured Preferential Generation

Severi Rissanen

Department of Computer Science Aalto University severi.rissanen@aalto.fi

Markus Heinonen

Department of Computer Science Aalto University

Arno Solin

Department of Computer Science Aalto University

Abstract

In the domains of image and audio, diffusion models have shown impressive performance. However, their application to discrete data types, such as language, has often been suboptimal compared to autoregressive generative models. This paper tackles the challenge of improving discrete diffusion models by introducing a structured forward process that leverages the inherent information hierarchy in discrete categories, such as words in text. Our approach biases the generative process to produce certain categories before others, resulting in a notable improvement in log-likelihood scores on the text8 dataset. This work paves the way for more advances in discrete diffusion models with potentially significant enhancements in performance.

1 Discrete Diffusion Models

Diffusion models can be described as hierarchical latent variable models that are trained with a variational lower bound objective on the marginal log-likelihood. The variational lower bound is formed using a predefined variational inference distribution $q(\mathbf{z}_{1:T} \mid \mathbf{z}_0) = \prod_{t=1}^T q(\mathbf{z}_t \mid \mathbf{z}_{t-1})$, where the latent trajectory $\mathbf{z}_{1:T}$ generates T successive variations of the original data \mathbf{z}_0 . The generative model is defined as a Markov chain going in the opposite direction: $p_{\theta}(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^T p_{\theta}(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$, where θ are neural network parameters and $p(\mathbf{z}_T)$ is some prior distribution that is tractable to sample from. The model is trained by maximizing the evidence lower bound:

$$\log p_{\theta}(\mathbf{z}_{0}) \geq \mathbb{E}_{q} \left[\log \frac{p_{\theta}(\mathbf{z}_{0:T})}{q(\mathbf{z}_{1:T} \mid \mathbf{z}_{0})} \right]$$

$$= -\mathbb{E}_{q} \left[\operatorname{KL}(q(\mathbf{z}_{T} \mid \mathbf{z}_{0}) \parallel p(\mathbf{z}_{T})) + \sum_{t=2}^{T-1} \operatorname{KL}(q(\mathbf{z}_{t-1} \mid \mathbf{z}_{t}, \mathbf{z}_{0}) \parallel p_{\theta}(\mathbf{z}_{t-1} \mid \mathbf{z}_{t})) - \log p_{\theta}(\mathbf{z}_{0} \mid \mathbf{z}_{1}) \right],$$
(1)

where the first term is zero if we set the inference distribution, or 'forward process', endpoint to $q(\mathbf{z}_T \mid \mathbf{z}_0) \approx p(\mathbf{z}_T)$. Usually all of these distributions are factorized with respect to the data dimensions (e.g., the pixels in an image). In the discrete diffusion framework first proposed by Sohl-Dickstein et al. [2015] and later extended by Hoogeboom et al. [2021] and Austin et al. [2021], all of these distributions are categorical distributions (e.g., categorical distributions over the 256 possibilities over a single byte representing a pixel, or all the tokens in a text data set). In practice, the transitions are parameterized as $p_{\theta}(\mathbf{z}_{t-1} \mid \mathbf{z}_t) \propto \sum_{\mathbf{x}_0} q(\mathbf{z}_{t-1}, \mathbf{z}_t \mid \mathbf{z}_0) p_{\theta}(\mathbf{z}_0 \mid \mathbf{z}_t)$.

Neural Information Processing Systems (NeurIPS) 2023 Workshop on Diffusion Models.

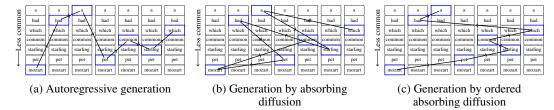


Figure 1: Three approaches to generating the expression: mozart had a pet starling which. The words are ordered top-down by how common they are, and the generation order is either left-to-right (a), random (b), or rare-words-first (c).

2 Motivation

For text data, Austin et al. [2021] suggested a structured forward process that induced a high probability of moving between semantically similar states. This seems intuitively sensible, since diffusion models for image data also transition to nearby states more probably than to far-away states. Their results were worse than with forward processes that had no particular structure [Austin et al., 2021]. This raises the question: What is the right way to add inductive biases and data structure to discrete diffusion models without hurting performance? In this paper, we explore *generating some categories of the data distribution before others*, with the idea that different categories, e.g., token types in text, encode different types of information. This is also analogous to the fact that Gaussian diffusion models for images generate low frequencies before the higher ones.

3 Related Work

In the context of language modelling, there has been some research on non-diffusion based models that generate tokens in structured orders. Gu et al. [2019] propose an approach that treats the orders of generated tokens as latent variables in a variational inference framework, where both a searched adaptive order and different predefined strategies are applied. One of the approaches, also explored in Ford et al. [2018], is to separate generation into two stages where the most common tokens are generated first and the less common after that, and vice versa. The difference to our diffusion-based framework is that they use causal masking in the self-attention and generate the positions of the added tokens as well as the tokens themselves one at a time, whereas we have the possibility of updating all of the tokens in the sequence in parallel, the number of generation steps is not bound to the dimension of the data and the neural network architecture is not restricted in any way.

4 Method

We begin with the 'absorbing' version of the discrete diffusion model from Austin et al. [2021], but modify it to generate some tokens before others. The typical 'masking' diffusion model introduces an additional 'absorbing' state in the discrete state space, moving original tokens to this state in a specific order until all are absorbed by step T. The generation then reveals the masked tokens.

If we instead mask various categories at different steps in masking diffusion models, some categories are also generated earlier than others. This means we can introduce inductive biases, such as ordering tokens by their frequency in the original data. The forward distribution is defined with

$$q(\mathbf{z}_t = a \mid \mathbf{z}_0) = \delta_{a,M} m_t(\mathbf{z}_0) + \delta_{a,\mathbf{z}_0} (1 - m_t(\mathbf{z}_0)). \tag{2}$$

Here, M is the masking state and m_t is the probability of having moved to the masking state by time t from the initial category z_0 . We recover the regular masking diffusion with the special case $m_t(\mathbf{z}_0) = m_t$, i.e., no dependence on the initial category itself. The m_t functions leave us a lot of room for design. We propose a way to cut down this design space to a single schedule that allows for different tokens to be generated at different stages and where, on average, an equal amount of information is generated at each step.

The mutual information schedule Austin et al. [2021] proposed to decrease the mutual information between the original data x_0 and x_t at an equal amount at each step in the forward process as a

heuristic. If T is the total amount of forward steps, then at step t the following equation should hold:

$$\frac{t}{T} = 1 - \frac{I(\mathbf{z}_0, \mathbf{z}_t)}{H(\mathbf{z}_0)} = \frac{H(\mathbf{z}_0, \mathbf{z}_t) - H(\mathbf{z}_t)}{H(\mathbf{z}_0)}.$$
(3)

For simplicity, the mutual informations and entropies are taken to be w.r.t. to the marginal distributions of all \mathbf{z}_0 , \mathbf{z}_t tokens. For the case of the standard mask diffusion and uniform diffusion, the schedule does not depend on the marginal distribution of \mathbf{z}_0 , but for more complex cases such as ours, it does. The resulting formula for the forward process is:

$$\frac{t}{T} = \frac{\sum_{\mathbf{z}_0} p(\mathbf{z}_0) m_t(\mathbf{z}_0) \log \frac{p(\mathbf{z}_0) m_t(\mathbf{z}_0)}{\sum_{\mathbf{z}_0'} p(\mathbf{z}_0') m_t(\mathbf{z}_0')}}{\sum_{\mathbf{z}_0} p(\mathbf{z}_0) \log p(\mathbf{z}_0)}.$$
(4)

Narrowing down The special case $m_t(\mathbf{z}_0) = m_t$ results in the regular mask diffusion with $\frac{t}{T} = m_t$. We choose $m_t(\mathbf{z}_0)$ to be such that in the limit of infinite steps $T \to \infty$, $m_t(\mathbf{z}_0)$ changes for a single \mathbf{z}_0 at a time. That is, for all but one \mathbf{z}_0 , $m_t(\mathbf{z}_0)$ is either 0 or 1, and for a single \mathbf{z}_0 , it is a monotonically increasing function between 0 and 1. Only one token is being destroyed at the same time. This, alongside the mutual information requirement, specifies the functions $m_t(\mathbf{z}_0)$ entirely. In practice, with a finite amount of timesteps T, we simply take snapshots of this idealized continuous process $\{m_t(\mathbf{z}_0)\}_{\mathbf{z}_0}$.

Choosing the order We experiment with different token orderings. 1) Most frequent categories first 2) Least frequent categories first 3) A random order 4) An order based on information gain, as a proxy for finding conditional independence structures in data. For the last one, we first estimate the marginal distribution of token types in sequences of fixed length from the data, and then calculate the information gain on this marginal distribution given that we observed some token type in the sequence. Formally:

$$\mathbb{E}_a \mathrm{IG}(X, a) = \mathbb{E}_a [H(X) - H(X \mid a)], \tag{5}$$

where X is a random variable that gives the probability of observing different token types when sampling a single element from the sequence, and a is the observation that a single example of a given token type exists in the sequence. The expectation \mathbb{E}_a means that we consider both realizations of observing a token and not observing it in the sequence. The sequence length that we consider depends on the data set: For TEXT8, even though we generate sequences of length 200, we choose the information gain sequence length to be 10 since the amount of token types is low and it is likely that all token types are observed in longer sequences.

5 Results

We focus on text data for these experiments. Experiments done with simple Transformers without causal masking.

Toy data To illustrate why adding bias on the token order could improve the results, we experiment with the following toy data set: First, we sample every other token of a token sequence randomly with 50% probability as a's or b's, and leave the other tokens blank, e.g., a?b?b?a?b?a?.... Second, the remaining blank tokens are filled with c:s,d:s,e:s and f:s deterministically depending on the combination of the surrounding two tokens with the

	Valid ELBO	Train ELBO
Absorbing	0.181	0.4626
Ordered	0.129	0.3244

Table 1: Generating the tokens 'a' and 'b' before 'c', 'd', 'e' and 'f' improves results on a toy data set where the latter are conditionally independent of the former.

rules a?a \to aca, a?b \to acb, b?a \to bda and b?b \to beb. Since there are correlations between the tokens, a standard absorbing-state model with a small enough amount of diffusion steps will produce mistakes due to the factorized nature of the reverse process $p_{\theta}(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$. In contrast, a model that generates a:s and b:s first can, in principle, generalise perfectly in only two steps. We confirm this in a simple experiment where we train a mdoel to generate a:s and b:s before the others, and list the achieved losses in Table 1.

The toy data set illustrates a mechanism through which a token-biased model can perform better: (Approximate) conditional independence of some token types given others. The question is then how to find such approximate conditional independences in data.

TEXT8 We next experiment on the TEXT8 character-level data set. Here, we compare 5 different token orders, including on an information gain based ordering, but the MI schedule is skewed such that slightly more time is spent on the high information gain tokens. A visualization of the forward process with the commonfirst generation is shown in Fig. 7. The models are 4-layer transformers trained on a single GPU with T=1000 diffusion steps.

Fig. 2 shows the test perplexities with the different orderings across multiple runs for each ordering. The perplexities are estimated as 2^{-ELBO}, where the ELBO is normalized to per-character and expressed in log-base 2. The standard absorbing state model is also included for comparison. The common-first generation outperforms the other methods in text perplexity by a clear margin. Perhaps surprisingly, the information gain based ordering performs noticeably worse, although the skewed version seems to be slightly better on average. Generating the low-frequency tokens first performs about as badly as the random ordering. Another surprising observation is that all of the ordered methods perform slightly

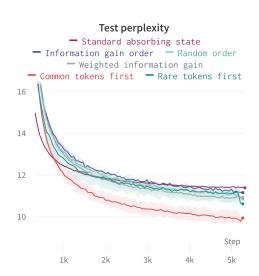


Figure 2: Test perplexities on TEXT8 calculated with the ELBO as a proxy for the marginal likelihood.

better than the standard absorbing state diffusion, on average, although the variance across runs is also clearly higher. Thus, especially the common-first method is able to fit better to the data, but training may be more challenging in practice.

WIKITEXT In this example, our model is evaluated on the WIKITEXT-2 data set, which has a larger vocabulary (8300) compared to TEXT8 using a simple word-based tokenizer, leading to a challenge: most tokens will only have a chance of being generated in a single time step, leading to slow convergence in early experiments. Tokens were grouped into 20 blocks based on frequency on the training set, and the ordering was defined between those 20 blocks. We experimented with skewing the token frequencies for the blocking procedure with a parameter freq $^{\alpha}$, where $\alpha \in \{0.9, 1, 1.1\}$, to adjust the low-frequency blocks to be larger or smaller. We trained a transformer using common-first generation and a standard absorbing state model. As shown in Fig. 3, test perplexities were generally close but did not match the absorbing state model, suggesting a need for more careful design for larger data sets and complex token sets.

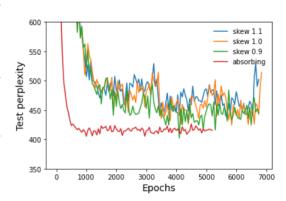


Figure 3: Test perplexities with different variants of the common-first model where tokens are grouped together with different strategies on WIKITEXT-2, as well as the standard absorbing state model.

6 Conclusion

We have proposed and experimented on a new type of discrete diffusion model that generates some categories strictly before others, and discovered that it can clearly improve results with a simple text data set. Next steps include trying out more complex, non-character-level, text data sets as well as

other types of data, such as graphs or segmentation maps. We believe that our model lays groundwork for a more systematic exploration of structured diffusion processes in discrete state spaces.

References

- Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.
- Nicolas Ford, Daniel Duckworth, Mohammad Norouzi, and George Dahl. The importance of generation order in language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2942–2946, 2018.
- Jiatao Gu, Qi Liu, and Kyunghyun Cho. Insertion-based Decoding with Automatically Inferred Generation Order. Transactions of the Association for Computational Linguistics, 7:661–676, 11 2019.
- Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

```
t=0 ployees it produced nothing but created jobs that would never have
\hookrightarrow existed if one was only concerned with developing a real mine the
\,\hookrightarrow\, world says the owner is an exploiter and the workers do all the real
\hookrightarrow work he left the enterprise entirely in the hands of t
t=125 ploy??s?i??produc?d?no?hing?bu??cr?a??d?jobs??ha??would?n?v?r?hav???
xis??d?if?on??was?only?conc?rn?d?wi?h?d?v?loping?a?r?al?min???h??world |
  ?says??h??own?r?is?an??xploi??r?and??h??work?rs?do?all??h??r?al?work?h
→ ??1?f???h???n??rpris???n?ir?ly?in??h??hands?of??
t=250 ploy??s?i??produc?d?no?hi?g?bu??cr????d?jobs??h???would???v?r?h?v???
xis??d?if?o???w?s?o?ly?co?c?r??d?wi?h?d?v?lopi?g???r??l?mi????h??world |
 ?s?ys??h??ow??r?is?????xploi??r???d??h??work?rs?do??ll??h??r??l?work?h
→ ??1?f???h?????rpris????ir?ly?i???h??h??ds?of??
t=375 pl?y??s????produc?d????h??g?bu??cr????d?j?bs??h???w?uld???v?r?h?v???
x?s??d??f?o???w?s???ly?c??c?r??d?w??h?d?v?l?p??g???r??l?m?????h??w?rld |
 ?s?ys??h??ow??r??s?????xpl????r???d??h??w?rk?rs?d???ll??h??r??l?work?h |
→ ??1?f???h??????rpr?s?????r?ly?????h??h??ds??f??
t=500 pl?y??????p??duc?d????h??g?bu??c?????d?j?b???h???w?uld???v???h?v??? |
x????d??f?????w?s???ly?c??c????d?w??h?d?v?l?p??g??????l?m?????h??w??ld 
  ???ys??h???w?????????xpl???????d??h??w??k???d???ll??h?????l?w??k?h
\  \, \rightarrow \  \, ??1?f???h??????p??????????1y?????h??h??ds??f??
t=625 p??y??????p??duc?d???????g?bu??c?????d?j?b???????w?u?d???v????v??? |
x????d??f?????w?????y?c??c????d?w????d?v???p?g???????m???????w???d
  ????y??????w???????????xp????????d?????w??k???d???????????????k?? |
→ ????f?????????p?????????y??????????d???f??
t=750 p??y??????p????????????g?b?????????j?b??????w??????v????v???
x???????f????w??????y???????w?????v???p??g???????m??????w????
```

Figure 4: Visualization of the forward process for text8 with the diffusion where common tokens are moved to the absorbing state first. The process starts out by diffusing 'e':s and spaces.

```
t=667 e e???t??? t??? e??e??? ??? ?t? ?e?e??? ??te? ??e? ?? t?e ??????? ??
→ ?e?? ?? ??????? ?????? ?? t?e e????e? ?e?e ?e???? ?????? ?? ?e????t?

→ ??? ?e?? ?e?? t? t??? ??? ?? ???te ???????t??? ??

→ ?e?? a? ?????a? ?????? a? t?e e????e? ?e?e ?e??an ???a?? an ?e????t?
→ ????a??e? ?? ?t?e? ??a?a?te?? t?e ???????? ?? ???????n? a?? ?e??t??

→ t?? ?e?? ?e?? t? t??? ?a? ?n ???te t?a???at??? a?

t=444 e e???ti?n t?i? e??e??? an? it? ?e?e?a? a?te? ??e? ?n t?e ?a?i??? a?
→ ?e?? a? ?????a? ?i?i?? a? t?e en???e? ?e?e ?e??an ???ain an ?e???it?
→ ????a??e? in ?t?e? ??a?a?te?? t?e ?in??i?? ?? ??????in? an? ?en?t??

→ t?? ?e?? ?e?? t? t?i? ?a? in ??ite t?an??ati?n a?

t=333 e e???tion t?i? e??eror an? it? ?e?e?a? a?ter ??e? on t?e ?a?i??? a?
→ ?e?? a? ?o???ar ?i?i?? a? t?e en?o?e? ?e?e ?er?an ???ain an ?e??rit?
  ????a??e? in ot?er ??a?a?te?? t?e ?in??i?? o? ?o??o?in? an? ?en?t??
→ t?o ?ero ?ero to t?i? ?a? in ??ite tran??ation a?
t=222 e e???tion this e??eror an? its se?era? a?ter ?se? on the ?a?i??? as
→ ?e?? as ?o???ar si?i?? as the en?o?e? ?ere ?er?an h?rain an se??rit?
 s???a??e? in other ?hara?ters the ?in??is? o? ?o??o?in? an? ?en?ths
→ t?o ?ero ?ero to this ?as in ??ite trans?ation as
t=111 e e???tion this e??eror and its se?eral a?ter used on the ?a?i?u? as
→ ?ell as ?o?ular sicil? as the en?o?ed ?ere ?er?an hurain an securit?
→ s?lla?led in other characters the lin?uis? o? ?ollo?in? and len?ths
\hookrightarrow t?o ?ero ?ero to this ?as in ?uite translation as
t=0 e egyption this emperor and its several after used on the maximum as
\hookrightarrow well as popular sicily as the enjoyed were german hurain an security

→ syllabled in other characters the linguism of following and lengths

  two zero zero to this was in quite translation as
```

Figure 5: Visualization of the reverse process of the diffusion process where the common tokens are generated first. The trained model is a 12-layer transformer with about 10 million parameters.

```
?? ??e ??? ???????? ???? ??e? ????? ???? ? e?? ???e? ??? ?e????e
→ ???e??e? ?e?e? ????? ?ee??e????e?? ?e???e? ?e
t=625 ea?e? ???e? ??a?? ?? t?e ????a?e ??? t?e ?e?t?? ?e??te ?? ? ????t
→ ???e ? ???? ??? t?e ???e?te?e? ?? ???? ??t?? ???ee?at???e? ???t ????e?
→ at t?e ??? ???????e? ???? ??e? ????? ?e? ???? ? e?? t??ee ??? ?e????e
 ??te??e? ?e?e? ????? ?ee??e????e?t ?e???e? ?e ??e
t=500 ea?e? ???e? ??a?? ?? t?e ?a??a?e ??n t?e ?eat?? ?e??te ?? a ????t
→ ?a?e a ???? an? t?e ?n?e?te?e? ?? ??? ??t?? ?n?ee?at??ne? ???t ????e?
→ at t?e ??? ??????ne? ?an? ??e? ???n? ne? ???? ? e?? t??ee ??? ?e????e

→ ?nte??e? ?e?en ?a??? ?een?e????ent ?e???e? ?e ?ne

t=375 ea?e? ?i?e? ??a?? o? t?e ?a??a?e ?on t?e ?eat?? ?e?ote o? a ?i??t
→ ?a?e a ??o? an? t?e in?e?te?e? ?? ?i? ??t?? in?ee?atione? ?o?t o???e?
\hookrightarrow at t?e ?i? ?o???ine? ?an? o?e? ?o?n? ne? ??o? ? e?o t??ee ?o? ?e??i?e
  inte??e? ?e?en ?a?o? ?een?e??i?ent ?e?o?e? ?e one
t=250 ea?er ?i?es s?a?s o? the ?assa?e ?on t?e ?eat?? se?ote o? a ?irst
→ ?a?e a ?ro? an? the in?e?te?e? ?? ?is ?ot?? in?ee?atione? ?ost o???e?
\hookrightarrow at the ?is ?o???ine? ?an? o?er ?o?n? ne? ?ro? s ero three ?o? ser?i?e
→ inte??es se?en ?a?or ?een?ersi?ent ?e?o?e? ?e one
t=125 ea?er ?i?es s?als o? the ?assa?e ?on the death? secote o? a ?irst
-> ?a?e a ?ro? and the indecteded ?? his ?otld indee?ationed ?ost occ?ed
→ at the his co??lined ?an? o?er ?ound ne? ?ro? s ero three co? ser?ice
→ intelles se?en la?or ?eendersi?ent ?eco?ed ?e one
t=0 eaver gives smals of the kassage won the deathy secote of a first wave
\,\hookrightarrow\, a from and the indecteded by his botld indegationed most occued at
intelles seven labor peendersiment becomed be one
```

Figure 6: Visualization of the reverse process of the diffusion process where the common tokens are generated first. The trained model is a 4-layer transformer with about 3 million parameters (also used for the experiments in Fig.2)

```
t=1000 <mask> <m
   \hookrightarrow <mask> <mas
  \rightarrow <mask>
t=889 < mask > , < mask > < 
  t=778 < mask> , < mask> < mask> a < mask> to < mask> < mask> < mask> < mask> < mask>
  _{\hookrightarrow} <mask> <
t=667 < mask> , < mask> < mask> a < mask> to < mask> < mask> < mask> < mask> < mask>
  _{\hookrightarrow} <mask> as <mask> <mask> . <mask> <mask> <mask> '
t=556 <mask> , they <mask> a <mask> to their <mask> over new world as they
 \hookrightarrow <mask> . <mask> their <mask> '
t=444 \le mask > , they <mask > a <mask > to their way over new world as they
 \hookrightarrow did . despite their men '
t=333 \le mask > , they sent a <mask > to their way over new world as they did
  \hookrightarrow . despite their men '
t=222 arrived , they sent a <mask> to their way over new world as they did
  \hookrightarrow . despite their men '
t=111 arrived , they sent a <mask> to their way over new world as they did
 \hookrightarrow\, . despite their men '
t=0 arrived , they sent a platoon to their way over new world as they did
  \,\hookrightarrow\, . despite their men '
```

Figure 7: Visualization on the Wikitext data set of the reverse process of the diffusion process where the common tokens are generated first. The trained model is a 12-layer transformer with about 10 million parameters.