Masked Diffusion Models are Secretly Learned-Order Autoregressive Models

Prateek Garg[*]
Aalto University

Bhavya Kohli[*]
NUS, Singapore

Sunita Sarawagi
IIT Bombay

Abstract

Masked Diffusion Models (MDMs) have emerged as one of the most promising paradigms for generative modeling over discrete domains. It is known that MDMs effectively train to decode tokens in a uniformly random order, and that this ordering has significant performance implications in practice. This observation raises a fundamental question: can we design a training framework that optimizes for a favorable decoding order? We answer this in the affirmative, showing that the continuous-time variational objective of MDMs, when equipped with multivariate noise schedules, can identify and optimize for a decoding order during training. We establish a direct correspondence between decoding order and the multivariate noise schedule and show that this setting breaks invariance of the MDM objective to the noise schedule. Furthermore, we prove that the MDM objective decomposes precisely into a weighted auto-regressive losses over these orders, which establishes them as auto-regressive models with learnable orders.

1 Introduction

Autoregressive models (ARMs) remain the dominant paradigm for sequential data generation, largely due to their natural alignment with next-token prediction tasks on domains such as text. While text data possesses an inherent left-to-right sequential structure, many other discrete modalities of interest lack such canonical orderings. For example, tabular data (Zhu et al., 2022), graphs (Bu et al., 2023), 1D tokenized images (Yu et al., 2024) present ambiguity in how dimensions should be ordered for ARMs. This limitation of ARMs has also motivated research in model classes based on diffusion (Sohl-Dickstein et al., 2015; Ho & Salimans, 2022; Dhariwal & Nichol, 2021; Ho et al., 2020; Song et al., 2021; Kingma et al., 2021; Karras et al., 2022). While diffusion models have demonstrated remarkable success in continuous domains–including image synthesis, video generation, and speech processing–their application to discrete structures remains an active research area (Austin et al., 2021; Shi et al., 2025a; Yu et al., 2025a; Gat et al., 2024; Lou et al., 2024; Rütte et al., 2025; Sahoo et al., 2025a). Extending diffusion-like dynamics to discrete data modalities such as natural language, molecular structures, and protein sequences is a compelling alternative to ARMs which can potentially enable parallel generation and provide trade-offs between quality and sampling speed. Recent research (Sahoo et al., 2024b; Shi et al., 2025a) has positioned one variant of discrete diffusion, Masked Diffusion Models (MDMs) as particularly promising, by extending their formulations to continuous time, combining the theoretical grounding of diffusion processes with the simplicity of masked token modeling.

Recent theoretical analyses (Zheng et al., 2025; Ou et al., 2025) have revealed a fundamental equivalence: the learning objective of MDMs corresponds exactly to masked language modeling and can be interpreted as any-order autoregressive models (AO-ARMs) (Hoogeboom et al., 2022). In other words, the MDM training objective optimizes equally over all possible orders. Kim et al.

[*]Work done while at IIT Bombay
Correspondence to: Prateek Garg <prateekg@iitb.ac.in>

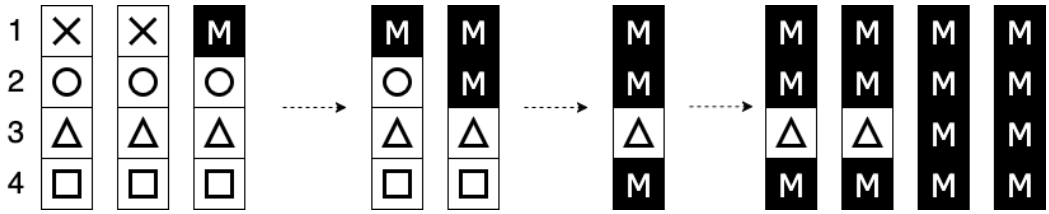
EurIPS 2025 Workshop on Principles of Generative Modeling (PriGM).

(2025) introduces inference-time heuristics to select a state-dependent ordering which achieves improvements on reasoning tasks such as sudoku. This observation suggests that MDMs possess the capacity to discover and exploit ordering structures during training, with certain orderings proving more advantageous than others even when the training objective weighs each ordering equally. This is an instance of train-test mismatch, while training weighs all the ordering equally, inference does not. This motivates us to extend the MDM objective, which can discover the order during training and thus optimize for that order. Concurrently, recent work (Wang et al., 2025) investigates Learned-Order Autoregressive Models (LO-ARMs) which attempts to learn a state-dependent order distribution. We instead pose a simpler question: *Can the masked diffusion training objective be extended in a principled manner to learn optimal state-independent orderings?*

Our Contributions We prove that when extended with multivariate noise schedules, the masked diffusion objective decomposes exactly as the expectation over possible orderings (Proposition 3.1), where the probability of sampling orders is defined by the noise schedule. We establish an exact correspondence of token ordering and multivariate noise schedule at inference time (Proposition 3.2). In Section 4, we validate our theoretical claims with experimental results on tabular generative modeling.

2 Background

Figure 1: Forward process of masked diffusion, masks variables in a order. While for univariate noise schedules, this order is uniformly random, multivariate noise schedule makes some order more likely than others.



Masked diffusion assumes a forward process where a token $\mathbf{x}_0$ transitions to a special mask state $\mathbf{m}$ at a random time defined by a noise schedule α_t. For time $t \geq s$,

$$q(\mathbf{x}_t|\mathbf{x}_s) = \alpha_{t|s}\delta_{\mathbf{x}_s} + (1 - \alpha_{t|s})\delta_{\mathbf{m}} \tag{1}$$

where $\alpha_t \in [0, 1]$ is a strictly decreasing function in t, with $\alpha_0 \approx 1$, $\alpha_1 \approx 0$, and $\alpha_{t|s} = \alpha_t/\alpha_s$. The forward process can be interpreted as follows: Between timestep s and t, we transition to a masked state with probability $1 - \alpha_{t|s}$. Once masked, a token stays masked with probability 1. This forward process admits a closed form posterior. For $s < t$,

$$q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0) = \begin{cases} \delta_{\mathbf{x}_t} & \mathbf{x}_t \neq \mathbf{m}, \\ \frac{1-\alpha_s}{1-\alpha_t}\delta_{\mathbf{m}} + \frac{\alpha_s-\alpha_t}{1-\alpha_t}\delta_{\mathbf{x}_0} & \mathbf{x}_t = \mathbf{m}. \end{cases} \tag{2}$$

The reverse process conditioned on $\mathbf{x}_0$ also has a simple interpretation–if $\mathbf{x}_t$ is masked, it will jump to the state $\mathbf{x}_0$ at time s with probability $\frac{\alpha_s-\alpha_t}{1-\alpha_t}$, and stay masked otherwise. Once $\mathbf{x}_t$ is unmasked, it remains unchanged until $t = 0$. To learn a reverse model p_θ, we optimize a variational bound (ELBO) on likelihoods. For a given a number of discretization steps T, we define $s(i) := (i - 1)/T$ and $t(i) := i/T$, then we can write the discrete time ELBO as:

$$-\log p_\theta(\mathbf{x}_0) \leq \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{i \geq 1} \log \frac{p_\theta(\mathbf{x}_{s(i)} \mid \mathbf{x}_{t(i)})}{q(\mathbf{x}_{t(i)} \mid \mathbf{x}_{s(i)})} \right] = \mathcal{L}(\mathbf{x}_0) \tag{3}$$

Under mean-parameterization (Sahoo et al., 2024a; Shi et al., 2025a), the reverse model learns a distribution over $\mathbf{x}_0$ given $\mathbf{x}_t$; $\mathbf{x}_0 \sim \mu_\theta(\mathbf{x}_t, t; \theta)$ and plugs into the closed form expression (2). For a sequence $\mathbf{x}^{1:L}$ consisting of L tokens, such a reverse model can be optimized with the continuous

time loss:

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \left(\sum_{\ell} \mathbb{E}_{\mathbf{x}_t \sim q(\cdot|\mathbf{x}_0)} [\log \boldsymbol{\mu}_\ell^{\mathbf{x}_0^\ell}(\mathbf{x}_t, t; \theta)] \right) \mathrm{d}t \tag{4}$$

where $\mu_\theta^{\mathbf{x}_0}$ refers to the probability corresponding to the specific token $\mathbf{x}_0$ from a distribution defined over the space of all possible values of $\mathbf{x}_0$.

3 Equivalence of Masked Diffusion and Learned-Order Autoregression

In this section, we will explore the behavior of masked diffusion when each position has a different noise schedule $\alpha_{t,\ell}$, which we refer to as a multivariate noise schedule.

3.1 Reframing ELBO as Loss over Orders

Proposition 3.1 The diffusion loss (Equation 3) can be decomposed over the orders as follows:

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = -\mathbb{E}_\pi \left[\sum_i \mathbb{E}_{t^*_{\pi(i)}|\pi} \left[\log \boldsymbol{\mu}(\mathbf{x}_{\pi(i)} | \mathbf{x}_{\pi(<i)}, t^*_{\pi(i)}; \theta) \right] \right] \tag{5}$$

where $\boldsymbol{\mu}(\mathbf{x}_{\pi(i)} | \mathbf{x}_{\pi(<i)}, t^*_{\pi(i)}; \theta) = \boldsymbol{\mu}_{\pi(i)}^{\mathbf{x}_0^{\pi(i)}}(\mathbf{x}_{\pi(<i)}, t^*_{\pi(i)}; \theta)$, $\mathbf{x}_{\pi(<i)}$ is a sequence obtained by masking the $\pi(\geq i)$ indices of $\mathbf{x}_0$ and $t^*_{\pi(i)}$ is the transition time of position $\pi(i)$. Under a time-independent network parameterization,

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = -\mathbb{E}_\pi \left[\sum_i \log \boldsymbol{\mu}(\mathbf{x}_{\pi(i)} | \mathbf{x}_{\pi(<i)}; \theta) \right] = -\mathbb{E}_\pi \left[LL_\pi(\mathbf{x}_0, \theta) \right] \tag{6}$$

where LL_π is the auto-regressive log-likelihood computed under order π and expectation is taken over all orders π, where the corresponding probabilities are given as

$$P(\pi) = \int_{\Omega_\pi} \left(\prod_\ell -\alpha_{t,\ell}' \right) \tag{7}$$

where $\Omega_\pi = \{t : t_{\pi(1)} < t_{\pi(2)} < \cdots < t_{\pi(L)}\}$

This expression can be interpreted as Learning-Order Autoregressive Models (Wang et al., 2025), with the key difference that our distribution over orders is state-independent (Equation 25)

3.2 Relationship between Decoding Order and Noise Schedules

Proposition 3.2 Let t^* be the time left in the reverse process with schedule α_t when a token transitions to a non-mask state, then $P(t^* \leq t) = 1 - \alpha_t$

Corollary 3.1 *For the ℓ-th variable, the transition time t^*_ℓ is distributed according to the p.d.f $-\alpha_{t,\ell}'$*

This gives us a way to sample an order π given the schedule. We have outlined the sampling process in Algorithm 1.

Algorithm 1 Order Sampler

Require: Noise schedules $\{\alpha_{t,\ell}\}_{\ell=1}^L$
Ensure: Sampled order π
1: Initialize an empty list $\mathcal{T} \leftarrow \varnothing$.
2: **for** $\ell = 1$ to L **do**
3: Sample $u \sim \mathrm{Uniform}(0, 1)$
4: Compute $t_\ell \leftarrow \alpha_\ell^{-1}(u)$ ▷ sample t_i
5: Append t_ℓ to $\mathcal{T}$
6: **end for**
7: $\pi \leftarrow \textsc{Sorted}(\mathcal{T})$
8: **return** π

This algorithm when used with univariate schedule also simplifies FHS sampler proposed by Zheng et al. (2025) and "diffusion denoising schedule" used in Sahoo et al. (2025b) for univariate schedules.

4 Experiments

In this section, we apply a masked diffusion with feature wise schedules to the task of tabular data generation. The forward process is applied independently across the sequence but with different noise schedules $\alpha_{t,\ell}$ across the sequence. The loss we use is:

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = \int_{t=0}^{t=1} \sum_{\ell} \frac{\alpha'_{t,\ell}}{1-\alpha_{t,\ell}} \mathbb{E}_{\mathbf{x}_t \sim q(\cdot|\mathbf{x}_0)} [\log \boldsymbol{\mu}_\ell^{\mathbf{x}_0^\ell}(\mathbf{x}_t;\theta)] \mathrm{d}t \tag{8}$$

Note the difference with Equation 4, each dimension is weight defined by corresponding schedules. We parameterize $\alpha_{t,\ell} = 1 - t^{w_\ell}$. To differentiate through the masking process we use RLOO gradient estimation by Kool et al. (2019), denoted by MDM(LS), and compare it with a masked diffusion model trained with fixed linear schedule (denoted MDM).

4.1 Results

In this section, and in Appendix C.1, we compare the performance of well established tabular synthesis baselines on several data fidelity metrics. We refer readers to Shi et al. (2025b) for descriptions of the baselines and metrics used. For all the datasets, we implement MDM and MDM(LS) using the same hyperparameter set. All models trained by us have around 86K parameters. We also note that many of the baseline methods make use of gaussian diffusion which typically requires a much higher number of steps than MDMs, whereas in our case the max number of steps is simply the number of columns.

In Tables 1-3, Underlined numbers denote the best performing method for each dataset. In Table 1, specifically for the Trend metric, we see that while not the best, both MDM and MDM(LS) are very competitive with state of the art baselines while having a fraction of the parameters–for example, TabDiff utilises models with 10M parameters. We note similar performance comparisons on the other metrics. We also report additional metrics with standard deviations across runs in Appendix C.1.

Table 1: Performance comparison of our models on **Trend** against baselines across six datasets. The Trend metric measures the quality of pairwise correlations.

Method	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	0.8549	0.9404	0.9151	0.9339	0.9200	0.9693
CoDi	0.7751	0.3159	0.8222	0.9347	0.9293	0.8890
TabDDPM	0.9699	0.9511	0.9339	0.9830	0.9729	0.8684
TabSYN	0.9807	0.9719	0.9787	0.9912	0.9687	0.9848
TabDIFF	<u>0.9851</u>	<u>0.9745</u>	<u>0.9826</u>	<u>0.9924</u>	<u>0.9741</u>	<u>0.9872</u>
MDM(LS)	0.9778	0.9705	0.9743	0.9710	0.9628	0.9781
MDM	0.9747	0.9642	0.9773	0.9703	0.9724	0.9766

Table 2: Performance comparison of our models on **Shape** against baselines across six datasets. The Shape metric measures how well synthetic data captures each column's marginal density.

Method	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	0.8871	0.9423	0.9063	0.9371	0.9329	0.9311
CoDi	0.7862	0.8423	0.6816	0.8844	0.8306	0.6773
TabDDPM	0.9825	0.9843	0.9728	0.9899	0.9870	0.2125
TabSyn	0.9919	<u>0.9899</u>	0.9856	0.9897	0.9874	<u>0.9794</u>
TabDiff	<u>0.9937</u>	0.9876	<u>0.9872</u>	<u>0.9922</u>	<u>0.9897</u>	0.9765
MDM(LS)	0.9878	0.9830	0.9841	0.9832	0.9750	0.9787
MDM	0.9851	0.9787	0.9866	0.9829	0.9822	0.9785

Table 3: Performance comparison of our models on the data fidelity metrics α-Precision and β-Recall, across the six datasets.

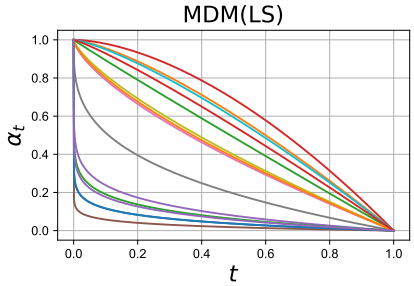
Method	α-Precision						β-Recall					
	Adult	Default	Beijing	Shoppers	Magic	News	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	0.8287	0.9048	0.8965	0.8656	0.8916	0.9476	0.2921	0.3931	0.3724	<u>0.5397</u>	0.5479	0.3942
CoDi	0.7758	0.8238	0.9495	0.8501	<u>0.9813</u>	0.8715	0.0920	0.1994	0.2082	0.5056	0.5219	0.3440
TabDDPM	0.9636	0.9759	0.8855	0.9859	0.9793	0.0000	0.4705	0.4783	0.4779	0.4846	0.5692	0.0000
TabSyn	<u>0.9939</u>	<u>0.9865</u>	0.9836	0.9942	0.9751	0.9505	0.4792	0.4645	0.4910	0.4803	0.5915	<u>0.4301</u>
TabDiff	0.9902	0.9849	<u>0.9911</u>	<u>0.9947</u>	0.9806	0.9736	<u>0.5164</u>	<u>0.5109</u>	0.4975	0.4801	<u>0.5963</u>	0.4210
MDM(LS)	0.9909	0.9839	0.9744	0.9818	0.9623	<u>0.9872</u>	0.4534	0.3781	0.5162	0.4435	0.3852	0.3473
MDM	0.9756	0.9717	0.9592	0.9827	0.9804	0.9779	0.4595	0.3810	<u>0.5309</u>	0.4583	0.3894	0.3405

Figure 2: Comparing the best validation losses for MDM(LS) and MDM and visualizing the noise schedules learned by MDM(LS) on the Adult dataset.

(a) Validation performance

Dataset	MDM(LS)	MDM
Adult	<u>13.1524</u>	13.1861
Default	<u>18.3272</u>	19.4247
Beijing	17.0060	<u>16.7610</u>
Shoppers	16.4604	<u>16.4246</u>
Magic	<u>12.4712</u>	12.6114
News	<u>48.9401</u>	49.9699

(b) Learned noise schedules on Adult Dataset



From Table 2a, we see that MDM(LS) leads to lower validation losses, although the performance gains are limited, as can be seen from Table 1. Among the baselines, TabDiff (Shi et al., 2025b) also utilises learned schedules. In Figure 2b, we visualize the schedules learned by MDM(LS) for the Adult dataset (Other datasets are provided in Figure 5). In contrast, TabDiff fails to learn any schedules as seen in Appendix C.2.

5 Discussion and Future Work

While we observe that masked diffusion models with learned schedules lead to lower validation losses, we do not observe significant gains in downstream data fidelity metrics. A possible reason for this observation is that learned schedulers increase the loss variance during training. For example, even in the univariate case where loss is invariant to the noise schedule, Shi et al. (2024) observe that the linear schedule leads to best likelihood performance and attribute it to low loss variance from linear schedule. Since our schedules are learned and different for each column, it might be possible that loss variance is high. Future work should focus on how these effects can be mitigated for multivariate schedules. An interesting open question is whether masked diffusion with learned schedules can discover independencies and structures in tabular data.

Acknowledgments and Disclosure of Funding

We acknowledge the support of the SBI Foundation Hub for Data Science & Analytics at the Indian Institute of Technology Bombay for providing financial support and infrastructure for conducting the research presented in this paper.

References

Alan N. Amin, Nate Gruver, and Andrew Gordon Wilson. Why Masking Diffusion Works: Condition on the Jump Schedule for Improved Discrete Diffusion, June 2025. URL http://arxiv.org/abs/2506.08316. arXiv:2506.08316 [cs].

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.

Yoann Boget. Simple and Critical Iterative Denoising: A Recasting of Discrete Diffusion in Graph Generation, June 2025. URL http://arxiv.org/abs/2503.21592. arXiv:2503.21592 [cs] Ordering and Detecting Mistakes.

Jie Bu, Kazi Sajeed Mehrab, and Anuj Karpatne. Let There Be Order: Rethinking Ordering in Autoregressive Graph Generation, May 2023. URL http://arxiv.org/abs/2305.15562. arXiv:2305.15562 [cs].

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design, June 2024. URL http://arxiv.org/abs/2402.04997. arXiv:2402.04997 [stat].

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete Flow Matching, November 2024. URL http://arxiv.org/abs/2407.15595. arXiv:2407.15595 [cs].

Jiaqi Han, Austin Wang, Minkai Xu, Wenda Chu, Meihua Dang, Yisong Yue, and Stefano Ermon. Discrete Diffusion Trajectory Alignment via Stepwise Decomposition, July 2025. URL http://arxiv.org/abs/2507.04832. arXiv:2507.04832 [cs].

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 12454–12465, 2021.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive Diffusion Models, February 2022. URL http://arxiv.org/abs/2110.02037. arXiv:2110.02037 [cs].

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 26565–26577, 2022.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the Worst, Plan for the Best: Understanding Token Ordering in Masked Diffusions, March 2025. URL http://arxiv.org/abs/2502.06768. arXiv:2502.06768 [cs].

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free!, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution, June 2024. URL http://arxiv.org/abs/2310.16834. arXiv:2310.16834 [stat].

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your Absorbing Discrete Diffusion Secretly Models the Conditional Distributions of Clean Data, February 2025. URL http://arxiv.org/abs/2406.03736. arXiv:2406.03736 [cs].

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path Planning for Masked Diffusion Model Sampling, May 2025. URL http://arxiv.org/abs/2502.03540. arXiv:2502.03540 [cs].

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized Interpolating Discrete Diffusion, June 2025. URL http://arxiv.org/abs/2503.04482. arXiv:2503.04482 [cs].

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024a.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models, November 2024b. URL http://arxiv.org/abs/2406.07524. arXiv:2406.07524 [cs].

Subham Sekhar Sahoo, Aaron Gokaslan, Chris De Sa, and Volodymyr Kuleshov. Diffusion Models With Learned Adaptive Noise, November 2024c. URL http://arxiv.org/abs/2312.13236. arXiv:2312.13236 [cs].

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The Diffusion Duality, June 2025a. URL http://arxiv.org/abs/2506.10892. arXiv:2506.10892 [cs].

Subham Sekhar Sahoo, Zhihan Yang, Yash Akhauri, Johnna Liu, Deepansha Singh, Zhoujun Cheng, Zhengzhong Liu, Eric Xing, John Thickstun, and Arash Vahdat. Esoteric Language Models, June 2025b. URL http://arxiv.org/abs/2506.01928. arXiv:2506.01928 [cs].

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow Matching with General Discrete Paths: A Kinetic-Optimal Perspective, December 2024. URL http://arxiv.org/abs/2412.03487. arXiv:2412.03487 [cs].

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and Generalized Masked Diffusion for Discrete Data, January 2025a. URL http://arxiv.org/abs/2406.04329. arXiv:2406.04329 [cs].

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation, February 2025b. URL http://arxiv.org/abs/2410.20626. arXiv:2410.20626 [cs].

Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Shuangyong Song, Yunhai Tong, Xiangtai Li, Xuelong Li, and Shuicheng Yan. Muddit: Liberating Generation Beyond Text-to-Image with a Unified Discrete Diffusion Model, May 2025c. URL http://arxiv.org/abs/2505.23606. arXiv:2505.23606 [cs].

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *The Ninth International Conference on Learning Representations*, 2021.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet Flow Matching with Applications to DNA Sequence Design, May 2024. URL http://arxiv.org/abs/2402.05841. arXiv:2402.05841 [q-bio].

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhe Wang, Jiaxin Shi, Nicolas Heess, Arthur Gretton, and Michalis K. Titsias. Learning-Order Autoregressive Models with Application to Molecular Graph Generation, July 2025. URL http://arxiv.org/abs/2503.05979. arXiv:2503.05979 [cs].

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An Image is Worth 32 Tokens for Reconstruction and Generation, June 2024. URL http://arxiv.org/abs/2406.07550. arXiv:2406.07550 [cs].

Runpeng Yu, Qi Li, and Xinchao Wang. Discrete Diffusion in Large Language and Multimodal Models: A Survey, July 2025a. URL http://arxiv.org/abs/2506.13759. arXiv:2506.13759 [cs].

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete Diffusion Multimodal Large Language Model with Parallel Decoding, May 2025b. URL http://arxiv.org/abs/2505.16990. arXiv:2505.16990 [cs].

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed Correctors for Discrete Diffusion Models, June 2025. URL http://arxiv.org/abs/2407.21243. arXiv:2407.21243 [cs].

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, April 2025. URL http://arxiv.org/abs/2409.02908. arXiv:2409.02908 [cs].

Yujin Zhu, Zilong Zhao, Robert Birke, and Lydia Y. Chen. Permutation-Invariant Tabular Data Synthesis, November 2022. URL http://arxiv.org/abs/2211.09286. arXiv:2211.09286 [cs].

Masked Diffusion Models are Secretly Learned-Order Autoregressive Models (Appendix)

A Related Work

Diffusion models have transformed generative modeling (Sohl-Dickstein et al., 2015; Ho & Salimans, 2022; Dhariwal & Nichol, 2021; Ho et al., 2020; Song et al., 2021; Kingma et al., 2021; Karras et al., 2022), and recent work has adapted them to discrete domains such as language, graphs, and proteins (Austin et al., 2021; Vignac et al., 2023; Gat et al., 2024; Stark et al., 2024; Shaul et al., 2024; Boget, 2025; Han et al., 2025; Zhao et al., 2025; Yu et al., 2025a). Discrete diffusion has been applied to diverse structured data types, including categorical modeling (Hoogeboom et al., 2021; Dieleman et al., 2022), text (Lou et al., 2024; Sahoo et al., 2025a; Amin et al., 2025), and multimodal setups (Yu et al., 2025b; Shi et al., 2025c; Campbell et al., 2024). Recent work by Sahoo et al. (2024c) applies multivariate in continuous diffusion setting. We instead focus on discrete diffusion processes elucidating a direct connections to decoding orders. Shi et al. (2024) also introduces GenMD4, where instead of token position, token values dictate the noise schedules which results in a significantly complicated objective. We instead show that a simple extension to the MDM objective works for token position dependent schedules.

The role of ordering and masking schedules in discrete diffusion has only recently received attention. Recent studies investigate inference heuristic or learned sampler for improved sample quality (Peng et al., 2025; Kim et al., 2025). Findings suggest that ordering choices significantly influence model inference quality. Wang et al. (2025) do not proceed from a diffusion setup, rather models decoding orders as a latent variable. Our work instead approaches this from discrete point of view, connecting them to learned-order autoregressive model.

B Proofs

Proposition 3.2 Let t^* be the time left in the reverse process with schedule α_t when a token transitions to a non-mask state, then $P(t^* \leq t) = 1 - \alpha_t$

Proof of Proposition 3.2: Consider a discretization of time interval $[0, 1]$ into T steps. We define $s(i) = \frac{i-1}{T}$ and $t(i) = \frac{i}{T}$, Let E_i be the event that the token transition to a non-mask state for the *first time* when transitioning from $t(i)$ to $s(i)$ We denote the event that it transition in the i-th with E'_i. We know that from the definition of reverse process, $P(\neg E'_i) = 1 - \frac{\alpha_{s(i)} - \alpha_{t(i)}}{1-\alpha_{t(i)}} = \frac{1-\alpha_{s(i)}}{1-\alpha_{t(i)}}$

$$
\begin{aligned}
P(E_i) &= P(E'_i) \prod_{i<j} P(\neg E'_j) && (9)\\
&= (1 - P(\neg E'_i)) \prod_{i<j} P(\neg E'_j) && (10)\\
&= \prod_{i<j} P(\neg E'_j) - \prod_{i\leq j} P(\neg E'_j) && (11)\\
&= \prod_{i<j} \frac{1-\alpha_{s(j)}}{1-\alpha_{t(j)}} - \prod_{i\leq j} \frac{1-\alpha_{s(i)}}{1-\alpha_{t(i)}} && (12)\\
&= \frac{1-\alpha_{s(i+1)}}{1-\alpha_{t(T)}} - \frac{1-\alpha_{s(i)}}{1-\alpha_{t(T)}} && (13)\\
&= \frac{1-\alpha_{t(i)}}{1-0} - \frac{1-\alpha_{s(i)}}{1-0} && (14)\\
&= \alpha_{s(i)} - \alpha_{t(i)} && (15)
\end{aligned}
$$

$$
P(\tau \leq t) = P\left(\bigcup_{i=0}^{i^*(t)} E_i\right) \tag{16}
$$

$i^*(t) = \arg\max_i s(i) \leq t < t(i)$. Since E_i's are disjoint events, we have,

$$P(\tau \leq t) = \sum_{i=0}^{i^*(t)} P(E_i) \tag{17}$$

$$= \sum_{i=1}^{i^*(t)} \alpha_{s(i)} - \alpha_{t(i)} \tag{18}$$

$$= 1 - \alpha_{t(i^*)} \tag{19}$$

As $T \to \infty$, $t(i^*) \to t$, so we get,

$$P(\tau \leq t) = 1 - \alpha_t \tag{20}$$

Proposition 3.1 The diffusion loss (Equation 3) can be decomposed over the orders as follows:

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = -\mathbb{E}_\pi \left[\sum_i \mathbb{E}_{t^*_{\pi(i)}|\pi} \left[\log \boldsymbol{\mu}(\mathbf{x}_{\pi(i)} | \mathbf{x}_{\pi(<i)}, t^*_{\pi(i)}; \theta) \right] \right] \tag{5}$$

where $\boldsymbol{\mu}(\mathbf{x}_{\pi(i)} | \mathbf{x}_{\pi(<i)}, t^*_{\pi(i)}; \theta) = \boldsymbol{\mu}^{\mathbf{x}_0^{\pi(i)}}_{\pi(i)} (\mathbf{x}_{\pi(<i)}, t^*_{\pi(i)}; \theta)$, $\mathbf{x}_{\pi(<i)}$ is a sequence obtained by masking the $\pi(\geq i)$ indices of $\mathbf{x}_0$ and $t^*_{\pi(i)}$ is the transition time of position $\pi(i)$. Under a time-independent network parameterization,

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = -\mathbb{E}_\pi \left[\sum_i \log \boldsymbol{\mu}(\mathbf{x}_{\pi(i)} | \mathbf{x}_{\pi(<i)}; \theta) \right] = -\mathbb{E}_\pi \left[L L_\pi(\mathbf{x}_0, \theta) \right] \tag{6}$$

Proof of Proposition 3.1 We will analyse the expression in 3

$$\mathcal{L}(\mathbf{x}_0) = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} (\log p_\theta(\mathbf{x}_{s(i)} \mid \mathbf{x}_{t(i)}) - \log q(\mathbf{x}_{t(i)} \mid \mathbf{x}_{s(i)})) \right] \tag{21}$$

Crucial thing to note is that this expectation is taken over all possible trajectories as shown in Figure 1 With $\beta_{t|s} = \frac{1-\alpha_s}{1-\alpha_t}$, we define the $p_\theta(\mathbf{x}_{s(i)} \mid \mathbf{x}_{t(i)})$ and $q(\mathbf{x}_{t(i)} \mid \mathbf{x}_{s(i)})$ in the table below

s	t	p	q
$\mathbf{m}$	$\mathbf{m}$	1	$\beta_{t\|s}$
$\mathbf{m}$	$\mathbf{x}$	0	0
$\mathbf{x}$	$\mathbf{x}$	$\alpha_{t\|s}$	1
$\mathbf{x}$	$\mathbf{m}$	$(1 - \alpha_{t\|s}) \cdot \mu^x(\mathbf{m}, t^*)$	$1 - \beta_{t\|s}$

where t^* is the time token transitions in the reverse process. We can safely ignore the case $\mathbf{m}, \mathbf{x}$ because it happens with probability 0. Consider a discretization of time interval $[0, 1]$ into T steps. We define $s(i) = \frac{i-1}{T}$ and $t(i) = \frac{i}{T}$,

$$\log p_\theta(\mathbf{x}_{s(i)} \mid \mathbf{x}_{t(i)}) - \log q(\mathbf{x}_{t(i)} \mid \mathbf{x}_{s(i)}) = \begin{cases} -\log(\beta_{t|s}(i)) & i < i^* \\ \log(1 - \alpha_{t|s}(i)) + \log \mu^x(\mathbf{m}, t(i^*)) - \log(1 - \beta_{t|s}(i)) & i = i^* \\ \log \alpha_{t|s}(i) & i > i^* \end{cases} \tag{22}$$

i^* refers to the discrete step where the transition happens. Summing it up, it reduces to telescopic sums and the only term which remains is $\log \mu^{\mathbf{x}}(\mathbf{m}, t(i^*))$. So, rewriting Equation 21, we have,

$$\mathcal{L}(\mathbf{x}_0) = \mathbb{E}_q[\log \mu^x(\mathbf{m}, t(i^*))] \tag{23}$$

For a sequence of token $\mathbf{x}^{1:L}$, it will have corresponding transition times $\{t(i)_\ell^*\}_{\ell=1}^L$. In continuous time limit, two tokens do not transition at the same time ($p(t(i)_i^* = t(i)_j^*.i \neq j) = 0$). For given transition times we define a map π such that $t^*_{\pi(1)} < t^*_{\pi(2)} < \cdots t^*_{\pi(L)}$, it is easy to see that π is a permutation. Corresponding $\mathbf{x}_t$'s at these transition time will have the corresponding masks (specifically at $t^*_{\pi(i)}$, all the indices corresponding to $\pi(\leq i)$, would be masked, and rest will be same as $\mathbf{x}_0$)

$$\mathcal{L}(\mathbf{x}_0) = \mathbb{E}_q\left[\sum_\ell \log \mu^{\mathbf{x}_{\pi(\ell)}}(\mathbf{x}_{t_\pi(\ell)^*}, t^*_{\pi(\ell)}))\right] \tag{24}$$

Given an order π, we can define joint distribution over $\{t^*_\ell\}$, using Proposition 3.2.

$$P(\pi) = \int_{\Omega_\pi}\left(\prod_\ell -\alpha'_{t,\ell}\right) \tag{25}$$

where $\Omega_\pi = \{t : t_{\pi(1)} < t_{\pi(2)} < \cdots < t_{\pi(L)}\}$

$$p(t^*_{1:L} \mid \pi) = \left(\frac{\prod_\ell -\alpha'_{t^*,\ell}}{P(\pi)}\right) I[t^* \in \Omega_\pi] \tag{26}$$

We can collect all the trajectories which result in permutation π and thus we arrive at the result,

$$\mathcal{L}(\mathbf{x}_0^{1:L}) = -\mathbb{E}_\pi\left[\sum_i \mathbb{E}_{t_{\pi(i)}|\pi}\left[\log \boldsymbol{\mu}(\mathbf{x}_{\pi(i)}|\mathbf{x}_{\pi(<i)}, t_{\pi(i)}; \theta)\right]\right] \tag{27}$$

C Experiments Details and Additional Results

C.1 Performance Metrics and Baseline Comparisons

For the tables below, results for MDM(LS) and MDM are averaged over 10 runs. The results of other baselines are taken from Shi et al. (2025b).

Table 4: Trend

Method	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	$0.8549_{\pm 0.0025}$	$0.9404_{\pm 0.0026}$	$0.9151_{\pm 0.0015}$	$0.9339_{\pm 0.0053}$	$0.9200_{\pm 0.0010}$	$0.9693_{\pm 0.0004}$
CoDi	$0.7751_{\pm 0.0008}$	$0.3159_{\pm 0.0005}$	$0.8222_{\pm 0.0011}$	$0.9347_{\pm 0.0025}$	$0.9293_{\pm 0.0015}$	$0.8890_{\pm 0.0001}$
TabDDPM	$0.9699_{\pm 0.0025}$	$0.9511_{\pm 0.0010}$	$0.9339_{\pm 0.0016}$	$0.9830_{\pm 0.0022}$	$0.9729_{\pm 0.0009}$	$0.8684_{\pm 0.0011}$
TabSyn	$0.9807_{\pm 0.0007}$	$0.9719_{\pm 0.0048}$	$0.9787_{\pm 0.0010}$	$0.9912_{\pm 0.0018}$	$0.9687_{\pm 0.0034}$	$0.9848_{\pm 0.0003}$
TabDiff	$\underline{0.9851_{\pm 0.0016}}$	$\underline{0.9745_{\pm 0.0075}}$	$\underline{0.9826_{\pm 0.0008}}$	$\underline{0.9924_{\pm 0.0012}}$	$\underline{0.9741_{\pm 0.0015}}$	$\underline{0.9872_{\pm 0.0004}}$
MDM(LS)	$0.9778_{\pm 0.0023}$	$0.9705_{\pm 0.0028}$	$0.9743_{\pm 0.0038}$	$0.9710_{\pm 0.0038}$	$0.9628_{\pm 0.0070}$	$0.9781_{\pm 0.0017}$
MDM	$0.9747_{\pm 0.0020}$	$0.9642_{\pm 0.0016}$	$0.9773_{\pm 0.0034}$	$0.9703_{\pm 0.0027}$	$0.9724_{\pm 0.0026}$	$0.9766_{\pm 0.0016}$

Table 5: Shape

Method	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	$0.8871_{\pm 0.0006}$	$0.9423_{\pm 0.0006}$	$0.9063_{\pm 0.0009}$	$0.9371_{\pm 0.0013}$	$0.9329_{\pm 0.0003}$	$0.9311_{\pm 0.0003}$
CoDi	$0.7862_{\pm 0.0006}$	$0.8423_{\pm 0.0007}$	$0.6816_{\pm 0.0005}$	$0.8844_{\pm 0.0026}$	$0.8306_{\pm 0.0002}$	$0.6773_{\pm 0.0004}$
TabDDPM	$0.9825_{\pm 0.0003}$	$0.9843_{\pm 0.0008}$	$0.9728_{\pm 0.0013}$	$0.9899_{\pm 0.0009}$	$0.9870_{\pm 0.0003}$	$0.2125_{\pm 0.0001}$
TabSyn	$0.9919_{\pm 0.0005}$	$\underline{0.9899_{\pm 0.0008}}$	$0.9856_{\pm 0.0007}$	$0.9897_{\pm 0.0014}$	$0.9874_{\pm 0.0005}$	$\underline{0.9794_{\pm 0.0004}}$
TabDiff	$\underline{0.9937_{\pm 0.0005}}$	$0.9876_{\pm 0.0007}$	$\underline{0.9872_{\pm 0.0009}}$	$\underline{0.9922_{\pm 0.0008}}$	$\underline{0.9897_{\pm 0.0005}}$	$0.9765_{\pm 0.0003}$
MDM(LS)	$0.9878_{\pm 0.0016}$	$0.9830_{\pm 0.0007}$	$0.9841_{\pm 0.0009}$	$0.9832_{\pm 0.0018}$	$0.9750_{\pm 0.0044}$	$0.9787_{\pm 0.0004}$
MDM	$0.9851_{\pm 0.0015}$	$0.9787_{\pm 0.0010}$	$0.9866_{\pm 0.0006}$	$0.9829_{\pm 0.0011}$	$0.9822_{\pm 0.0018}$	$0.9785_{\pm 0.0004}$

Table 6: α-Precision

Method	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	$0.8287_{\pm 0.0026}$	$0.9048_{\pm 0.0011}$	$0.8965_{\pm 0.0025}$	$0.8656_{\pm 0.0019}$	$0.8916_{\pm 0.0012}$	$0.9476_{\pm 0.0033}$
CoDi	$0.7758_{\pm 0.0045}$	$0.8238_{\pm 0.0015}$	$0.9495_{\pm 0.0035}$	$0.8501_{\pm 0.0036}$	$\underline{0.9813_{\pm 0.0038}}$	$0.8715_{\pm 0.0012}$
TabDDPM	$0.9636_{\pm 0.0020}$	$0.9759_{\pm 0.0036}$	$0.8855_{\pm 0.0068}$	$0.9859_{\pm 0.0017}$	$0.9793_{\pm 0.0030}$	$0.0000_{\pm 0.0000}$
TabSyn	$\underline{0.9939_{\pm 0.0018}}$	$\underline{0.9865_{\pm 0.0023}}$	$0.9836_{\pm 0.0052}$	$0.9942_{\pm 0.0028}$	$0.9751_{\pm 0.0024}$	$0.9505_{\pm 0.0030}$
TabDiff	$0.9902_{\pm 0.0020}$	$0.9849_{\pm 0.0028}$	$\underline{0.9911_{\pm 0.0034}}$	$\underline{0.9947_{\pm 0.0021}}$	$0.9806_{\pm 0.0024}$	$0.9736_{\pm 0.0017}$
MDM(LS)	$0.9909_{\pm 0.0028}$	$0.9839_{\pm 0.0019}$	$0.9744_{\pm 0.0034}$	$0.9818_{\pm 0.0048}$	$0.9623_{\pm 0.0034}$	$\underline{0.9872_{\pm 0.0012}}$
MDM	$0.9756_{\pm 0.0032}$	$0.9717_{\pm 0.0030}$	$0.9592_{\pm 0.0022}$	$0.9827_{\pm 0.0033}$	$0.9804_{\pm 0.0023}$	$0.9779_{\pm 0.0016}$

Table 7: β-Recall

Method	Adult	Default	Beijing	Shoppers	Magic	News
STaSy	$0.2921_{\pm 0.0034}$	$0.3931_{\pm 0.0039}$	$0.3724_{\pm 0.0045}$	$\underline{0.5397_{\pm 0.0057}}$	$0.5479_{\pm 0.0018}$	$0.3942_{\pm 0.0032}$
CoDi	$0.0920_{\pm 0.0015}$	$0.1994_{\pm 0.0022}$	$0.2082_{\pm 0.0023}$	$0.5056_{\pm 0.0031}$	$0.5219_{\pm 0.0012}$	$0.3440_{\pm 0.0031}$
TabDDPM	$0.4705_{\pm 0.0025}$	$0.4783_{\pm 0.0035}$	$0.4779_{\pm 0.0025}$	$0.4846_{\pm 0.0042}$	$0.5692_{\pm 0.0013}$	$0.0000_{\pm 0.0000}$
TabSyn	$0.4792_{\pm 0.0023}$	$0.4645_{\pm 0.0035}$	$0.4910_{\pm 0.0060}$	$0.4803_{\pm 0.0050}$	$0.5915_{\pm 0.0022}$	$\underline{0.4301_{\pm 0.0028}}$
TabDiff	$\underline{0.5164_{\pm 0.0020}}$	$\underline{0.5109_{\pm 0.0025}}$	$0.4975_{\pm 0.0064}$	$0.4801_{\pm 0.0031}$	$\underline{0.5963_{\pm 0.0023}}$	$0.4210_{\pm 0.0032}$
MDM(LS)	$0.4534_{\pm 0.0020}$	$0.3781_{\pm 0.0012}$	$0.5162_{\pm 0.0027}$	$0.4435_{\pm 0.0060}$	$0.3852_{\pm 0.0057}$	$0.3473_{\pm 0.0033}$
MDM	$0.4595_{\pm 0.0022}$	$0.3810_{\pm 0.0030}$	$\underline{0.5309_{\pm 0.0031}}$	$0.4583_{\pm 0.0067}$	$0.3894_{\pm 0.0042}$	$0.3405_{\pm 0.0022}$

C.2 Visualisation of Learned Schedules

Figure 3: Schedules learned by TabDiff, reproduced from original implementation

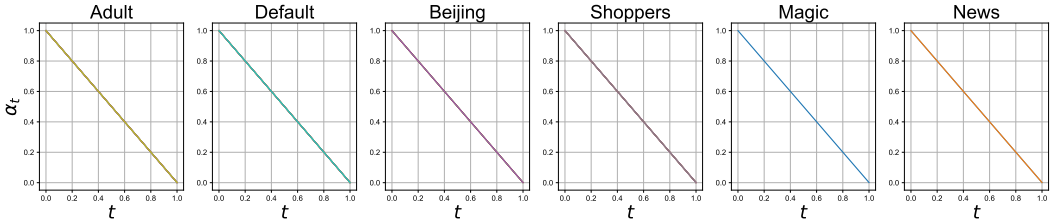


Figure 4: Tabdiff fails to learn schedules even with random initialization

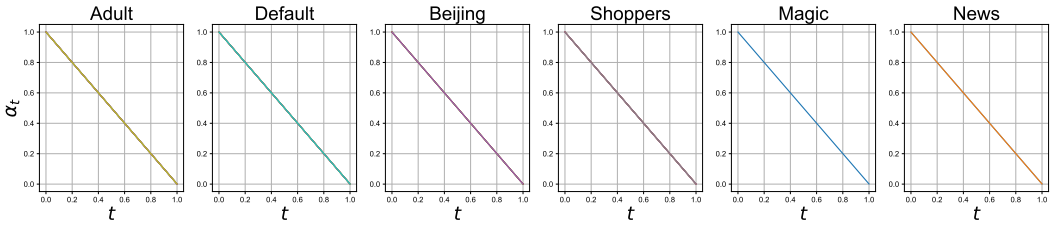
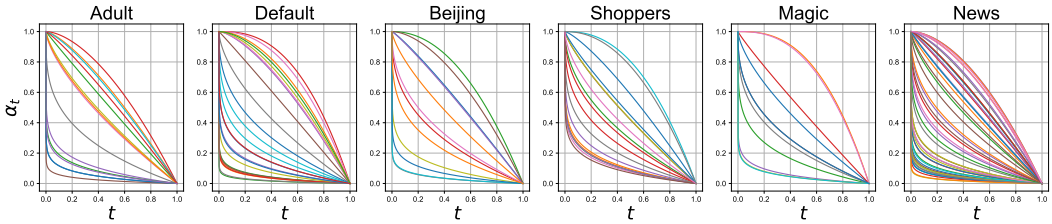


Figure 5: Schedules learned by our implementation of MDM(LS). Note that our implementation uses masked diffusion for all feature columns and not just for categorical features.



TabDiff makes use of a hybrid setup with masked diffusion for categorical columns with column wise learnable schedules. From our reproduction, we see that it fails to learn the schedules In Figure 3, we train the models from the implementation provided by the authors[2] and plot the schedules learned at the end of training. In Figure 4, we attempt to break symmetry by randomizing the initialization of the noise schedule, which unfortunately leads to the same result. For comparison, we include the schedules learned by our MDM(LS) models in Figure 5.

[2]https://github.com/MinkaiXu/TabDiff