Discrete Stochastic Localization for Non-Autoregressive Generation

Yunshu Wu[1] , Jiayi Cheng[2], Rob Brekelmans[3], Evangelos E. Papalexakis[1], Greg Ver Steeg[1]
[1]University of California Riverside, [2]New York University, [3]Vector Institute
{ywu380,epapalex,gregoryv}@ucr.edu,
jiayicheng@nyu.edu, rob.brekelmans@vectorinstitute.ai

Abstract

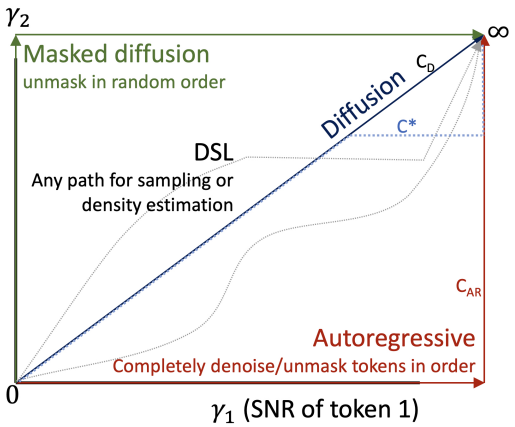
While autoregressive methods dominate language modeling, the promise of faster generation and more flexible control continues to spur interest in non-autoregressive approaches. Early efforts to adapt diffusion models with continuous Gaussian noise to discrete data have been superseded by discrete diffusion approaches based on adding and removing mask noise. The relative success of masked diffusion methods is somewhat puzzling because continuous noise appears to be more general as it can smoothly interpolate between complete masking and unmasking. We explore a new approach to non-autoregressive generative modeling based on *discrete stochastic localization* that leads to greater flexibility in noising and denoising. Masked discrete diffusion, continuous diffusion, and autoregressive models emerge as particular Signal-to-Noise Ratio (SNR) paths in our approach. We demonstrate that our approach is able to outperform existing continuous diffusion approaches on language modeling, and discuss the potential to close the gap with more established fully discrete approaches like masked diffusion models and autoregressive models. Code is anonymously available at https://anonymous.4open.science/r/DSL_anonymous-1833.

1 Introduction

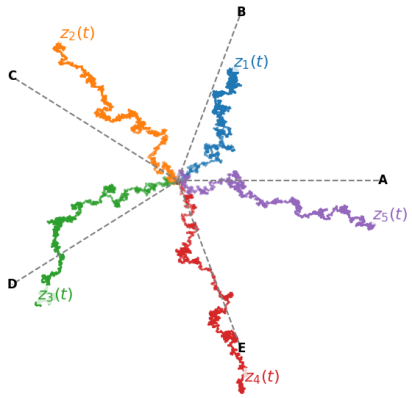
Non-autoregressive models like normalizing flows, diffusion, and VAEs start with a base distribution, usually a Gaussian, and transform it into the target distribution. The transformation is designed to facilitate easy sampling and likelihood estimation. The same intuition can be applied to discrete probability distributions, but a problem emerges. A smooth change in the probability distribution corresponds to state space dynamics that destroy information in an abrupt way. Reversing discrete noise leads to dynamics that unmask data in a fundamentally autoregressive way [34]. Early attempts to apply Gaussian noise in discrete embedding spaces have not performed as well as methods that apply discrete denoising [27, 36], but this is surprising because Gaussian noise is more flexible and can either partially or completely mask information, depending on the noise level.

"Stochastic Localization" (SL) has long been used as a proof technique [10], but we adopt the more general definition of Montanari [29] which defines SL as a noisy observation of some input that becomes "more informative" about the input over time. Generative modeling can then be achieved by simulating an unconditional version of this stochastic process ("conditional" on data and "unconditional" dynamics replace the notions of "forward" and "reverse" in classic diffusion). The output of the stochastic process "localizes" to regions that correspond to samples from the target distribution, which can be in a different domain from the dynamics, see Figure 1b. This approach provides a natural way to bridge continuous dynamics with modeling of discrete data like language.

We introduce an approach for Discrete Stochastic Localization (DSL) which we propose as an alternate paradigm for non-autoregressive generation of discrete data. By specializing stochastic



(a) We can specify a Signal-to-Noise Ratio (SNR) for each token and denoise along arbitrary paths. Autoregressive models take a single token from SNR=0 to SNR=∞ in one step. Diffusion models denoise/increase SNR in parallel across tokens.



(b) Example of DSL dynamics. Each component, $\boldsymbol{z}_i(t)$, localizes over time to a quadrant representing a symbol, A, B,... This dynamics uses the analytic/empirical denoiser for cyclic sequence data, ABCDE, BCDEA, etc. Details in Appendix C.

Figure 1: Discrete Stochastic Localization (DSL) dynamically "localizes" to sample discrete tokens.

localization to discrete embeddings on a hypersphere, we arrive at a formulation where each token can follow a different, arbitrary Signal-to-Noise Ratio(SNR) path. Remarkably, masked discrete diffusion, autoregressive models, and standard diffusion models emerge as particular "noise paths" in our formulation, see Figure 1a. Therefore, we propose that DSL is a natural framework for combining the performance benefits of autoregressive models with the fast and flexible sampling of non-autoregressive methods. **Contributions:**

- We propose Discrete Stochastic Localization (DSL), a non-autoregressive generation approach for discrete data that is noise path invariant. Autoregressive models, masked diffusion, and Gaussian diffusion models appear as particular paths.
- We provide novel and elegant expressions for probability estimation and sampling that can choose arbitrary SNR paths at inference time. This flexibility allows for hybrid approaches that combine benefits of diffusion and autoregressive models.
- Proof of concept experiments show that DSL outperforms state-of-the-art continuous diffusion language models in probability modeling, and that non-autoregressive generation is significantly improved using adaptive SNR paths that are a hybrid between diffusion and autoregressive models.

2 Discrete Stochastic Localization

Consider a discrete random variable or symbol, $s \in \mathcal{V}$. Let $\boldsymbol{x} = \text{enc}(s) \in \mathbb{R}^d, \|\boldsymbol{x}\| = 1$ represent an embedding of this discrete variable on the surface of the unit hyper-sphere. Define $\boldsymbol{z} \sim p_t(\boldsymbol{z}|\boldsymbol{x})$ as $\boldsymbol{z} = t\,\boldsymbol{x} + \sqrt{t}\,\boldsymbol{\epsilon}$, where $\boldsymbol{x} \sim P(\boldsymbol{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. Note that we use capital P for probability distributions on discrete spaces, and p for probability densities. For this noise model, the SNR is just t. We can give a dynamical realization of this process as an SDE.

$$d\boldsymbol{z} = \boldsymbol{x}\,dt + dW, \boldsymbol{x} \sim P(\boldsymbol{x}) \qquad \text{(Conditional SDE)}$$

As t grows, $\boldsymbol{z}$ becomes *more informative* about $\boldsymbol{x}$, and converges in distribution, $\boldsymbol{z}/t \xrightarrow{d} \boldsymbol{x}$. However, we can't use these dynamics to sample $P(\boldsymbol{x})$ because it requires conditioning on $\boldsymbol{x}$. Instead, we can find an *equivalent* SDE, in distribution, that does not require conditioning.

$$d\boldsymbol{z} = \hat{\boldsymbol{x}}(\boldsymbol{z}, t)\,dt + dW \qquad \text{(Unconditional SDE)}$$

The drift term needed for the marginal of the unconditional dynamics to match the conditional dynamics is just the Minimum Mean Square Error (MMSE) denoiser, which is also the (posterior) conditional mean, $\hat{\boldsymbol{x}}(\boldsymbol{z}, t) = \mathbb{E}_{P_t(\boldsymbol{x}|\boldsymbol{z})}[\boldsymbol{x}]$. All derivations appear in Appendix A for completeness.

SNR invariant denoising The first key new result in our proposed formulation is that by using our freedom to choose embeddings to restrict to the hyper-sphere, we get a simplified dynamics that are invariant to SNR, even with different SNR levels per token. Subsequent results then follow from this simplification. First, let's extend our setting to consider $\boldsymbol{x}_i, i = 1, \ldots, L$, for a sequence of length L. For readability, we assume $\boldsymbol{x} \equiv \boldsymbol{x}_{1:L}$. Then we can specify a separate SNR for each token,

$$\boldsymbol{z}_i = \gamma_i \, \boldsymbol{x}_i + \sqrt{\gamma_i} \, \boldsymbol{\epsilon}_i, \qquad \boldsymbol{x} \sim P(\boldsymbol{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(0, I).$$

We can now consider the SNR's to all be different functions of t, $\gamma_i(t)$. Surprisingly, the optimal denoiser is the same for any SNR combination or path, as shown in Appendix A.1.

$$\hat{\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{\gamma}) = \mathbb{E}_{P_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z})}[\boldsymbol{x}] = \mathbb{E}_{P(\boldsymbol{x})}[\boldsymbol{x} e^{\boldsymbol{x}\cdot\boldsymbol{z}}]/\mathbb{E}_{P(\boldsymbol{x})}[e^{\boldsymbol{x}\cdot\boldsymbol{z}}] = \hat{\boldsymbol{x}}(\boldsymbol{z}) \qquad \text{(SNR invariant denoiser)}$$

Sampling dynamics for arbitrary SNR paths For the arbitrary per-token SNR paths defined above, the conditional dynamics and the distributionally equivalent unconditional dynamics follow.

$$d\boldsymbol{z}_i = \boldsymbol{x}_i \, \dot{\gamma}_i dt + \sqrt{\dot{\gamma}_i} dW_i, \qquad \boldsymbol{x} \sim P(\boldsymbol{x}) \tag{1}$$

$$d\boldsymbol{z}_i = \hat{\boldsymbol{x}}_i(\boldsymbol{z}) \dot{\gamma}_i \, dt + \sqrt{\dot{\gamma}_i} dW_i \tag{2}$$

The rate of change in SNR, $\dot{\gamma}_i \equiv d\gamma_i/dt$, sets the effective per-token step size for the dynamics. To define admissible SNR contours or paths, let $\gamma_i(t) \colon [0, \infty) \to \mathbb{R}$ be continuous, non-decreasing functions with $\gamma_i(0) = 0$ and $\gamma_i(t) \to \infty$ as $t \to \infty$. If the marginals of simulating Equation (1) are $q_{\boldsymbol{\gamma}}(\boldsymbol{z})$ and for the unconditional dynamics, Equation (1), are $p_{\boldsymbol{\gamma}}(\boldsymbol{z})$, then we show in Appendix A that $q_{\boldsymbol{\gamma}}(\boldsymbol{z}) = p_{\boldsymbol{\gamma}}(\boldsymbol{z})$. Therefore by simulating Equation (2) using an arbitrary path, we can generate samples from $P(\boldsymbol{x})$. Because the bias doesn't depend on SNR, we can use the same denoiser to generate any path. An example of sampling using the unconditional SDE dynamics with $\gamma_i(t) = t$ is shown in Figure 1b.

Distribution modeling with DSL Besides sampling, DSL also allows estimation of probability. We can use arbitrary SNR paths as summarized by the line integral below.

$$-\log P(\boldsymbol{x}) = \frac{1}{2} \int_C \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) \cdot d\boldsymbol{\gamma} \qquad E_i(\boldsymbol{x}, \boldsymbol{\gamma}) \equiv \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}[\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i(\boldsymbol{z})\|^2] \tag{3}$$

We allow C to represent any SNR contour defined above, $\hat{\boldsymbol{x}}$ represents the MMSE denoiser, and $\mathbf{E}$ is the denoising "Error field". Note that we can also represent conditional distributions, $P(\boldsymbol{x}|y)$ by using the optimal conditional denoiser, $\hat{\boldsymbol{x}}(\boldsymbol{z}, y) \equiv \mathbb{E}_{P_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z},y)}[\boldsymbol{x}]$, which is also SNR invariant.

Equivalence on autoregressive paths We can define autoregressive denoising paths, C_{AR}, as in Figure 1a that denoise one token at a time (in order or in random order as masked diffusion models implicitly do [40]). This contour can be decomposed, $C_{AR} = \sum_{i=1}^{L} C_i$, where C_i is the contour that takes $\gamma_i = 0$ to ∞, while $\gamma_{<i} = \infty$ and $\gamma_{>i} = 0$. In that case, the path integral turns into the autoregressive probability decomposition.

$$\begin{aligned} -\log P(\boldsymbol{x}) &= \frac{1}{2} \int_{C_{AR}} \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) \cdot d\boldsymbol{\gamma} = \frac{1}{2} \sum_{i=1}^{L} \int_{C_i} \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) \cdot d\boldsymbol{\gamma} \\ &= \sum_{i=1}^{L} \frac{1}{2} \int_0^\infty d\gamma_i E_i(\boldsymbol{x}, (\infty, \ldots, \gamma_i, 0, \ldots)) = -\sum_{i=1}^{L} \log P(\boldsymbol{x}_i|\boldsymbol{x}_{<i}) \end{aligned} \tag{4}$$

The last equality is shown in Appendix A.1, and follows from applying Equation (3) to the conditional probability. The chosen contours have infinite SNR for $\boldsymbol{x}_{<i}$ and zero SNR for $\boldsymbol{x}_{>i}$, and this can be shown to be equivalent to the autoregressive decomposition.

Hybrid paths for practical negative log likelihood bound estimation The connection to autoregressive bounds has tangible benefits, as we now have the option of designing hybrid paths to make probability bounds easier and tighter. For instance, if we chose the typical diffusion path, C_D, parametrized by $\gamma_i(t) = t$ we'd get $-\log P(\boldsymbol{x}) = \frac{1}{2} \int_0^\infty \sum_i E_i(\boldsymbol{x}, t \, \mathbf{1}) dt = \frac{1}{2} \int_0^\infty \mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})} \|\boldsymbol{x} - \hat{\boldsymbol{x}}(\boldsymbol{z})\|^2 dt$. It is inconvenient to try to integrate to infinity, so instead we choose the hybrid path, C^*, in Figure 1a, that uses a diffusion path to t^*, then switches to autoregressive decoding to estimate the small contribution from high SNRs. The following is derived in Appendix A.

$$-\log P(\boldsymbol{x}) = \frac{1}{2} \int_{C^*} \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) \cdot d\boldsymbol{\gamma} = \frac{1}{2} \int_0^{t^*} \mathbb{E}_{p_t(\boldsymbol{z}|\boldsymbol{x})} \|\boldsymbol{x} - \hat{\boldsymbol{x}}(\boldsymbol{z})\|^2 dt - \mathbb{E}_{p_{t^*}(\boldsymbol{z}|\boldsymbol{x})}[\log P(\boldsymbol{x}|\boldsymbol{z})] \tag{5}$$

3 Implementation

Training and input parameterization Building on continuous-diffusion language models[11, 17], we decouple the learnable token-embedding space from the transformer's hidden dimensionality (unlike standard AR LMs where the embedding is simply the first layer). We constrain token embeddings to a unit hypersphere as mentioned in Section 2, which renders the MMSE denoiser SNR-invariant and lets us inject/remove noise directly in embedding space. Prior continuous-diffusion LMs map these noisy embeddings to the transformer space with a single linear projection; we find this ill-conditions attention under high-variance Gaussian noise, hampering likelihood and stability.

To address this inductive bias, we replace the naive projection with a *softmax converter* that first maps each noisy vector to a vocabulary distribution and then re-embeds it for the transformer. This produces inputs that are naturally compatible with attention and substantially stabilizes training:

$$\mathrm{SoftmaxConvert}(\boldsymbol{z}) = W_{\mathrm{tr}}\,\mathrm{softmax}\big(\beta\,\boldsymbol{z}\,W_{\mathrm{tok}}^{\top}\big), \quad (6)$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is a noisy embedding, $W_{\mathrm{tok}} \in \mathbb{R}^{V\times d}$ are token weights, β is a learnable temperature (initialized at $1/\sqrt{d}$), and $W_{\mathrm{tr}} \in \mathbb{R}^{d_{\mathrm{tr}}\times V}$ produces transformer-ready embeddings (not constrained to the sphere).

Table 1: Bits Per Character (BPC) on Text8 test set.

Method	BPC (↓)
Continuous Diffusion	
Plaid (MD4 impl.)	≤ 1.48
DSL (Ours)	$\leq$ **1.45**
Discrete Diffusion	
Mult. Diffusion	≤ 1.72
D3PM Uniform	≤ 1.61
D3PM Absorb	≤ 1.45
SEDD Absorb	≤ 1.41
MDLM	≤ 1.40
MD4	≤ 1.37
Autoregressive	
Transformer AR	**1.23**
IAF/SCF	1.88
AR Argmax Flow	1.39
AR Discrete Flow	**1.23**
Any-order Autoregressive	
ARDM	≤ 1.43
MAC	≤ 1.40

Cross-entropy loss as simplified log-likelihood objective Minimizing denoiser MSE under broad SNR coverage leads to a trivial joint-collapse of learned embeddings, a phenomenon noted in prior work[17, 8, 25, 11]. Instead, we predict logits over the vocabulary and minimize cross-entropy $\mathrm{CE}[P(\boldsymbol{x}_i \mid \boldsymbol{z}), P_\theta(\boldsymbol{x}_i \mid \boldsymbol{z})]$. Because the optimal denoiser satisfies $\hat{\boldsymbol{x}}_i(\boldsymbol{z}) = \mathbb{E}[\boldsymbol{x}_i \mid \boldsymbol{z}]$ under the true posterior, posterior matching yields the same optimum for the denoiser (and thus the NLL estimator), while avoiding collapse and training more efficiently on transformers. The implementation details (SNR sampling, optimizer, regularization) are deferred to Appendix B.

4 Experiments

Training setup Text8 (char-level, length 256). All models use a 12-layer transformer (12 heads, hidden 784). Baselines include Plaid (continuous diffusion) and masked/discrete diffusion variants. We report Bits-Per-Character (BPC, lower is better). All models are trained for 1 million steps.

Probability Estimate DSL attains BPC ≤ 1.45 on Text8, improving over the strongest scalable continuous-diffusion baseline Plaid ($\leq$ 1.48; MD4 reimpl. [36]) and surpassing earlier discrete-diffusion variants (D3PM-Uniform ≤ 1.61, Multinomial Diffusion ≤ 1.72). Although MD4 [36] (≤ 1.37) still leads, DSL closes over half of the continuous–discrete gap. Results can be found in Table. 1.

5 Conclusion

We proposed Discrete Stochastic Localization (DSL) as a way to bridge the gap between empirically successful autoregressive language models, including masked discrete diffusion, and the flexibility of non-autoregressive Gaussian diffusion models. Since Gaussian noise can, in some limit, completely mask tokens in an autoregressive way, a flexible Gaussian diffusion model should be able to replicate autoregressive models. We were able to analytically show this in our framework, with sampling and probability estimates that replicate autoregressive models on particular noise paths. We used these insights to improve over Gaussian diffusion approaches on language modeling tasks. While we did not bridge the gap with autoregressive models in terms of performance, we suggest that this is mostly due to a head start in architecture and training optimization and that it should be possible to incorporate the best of both worlds in future work using our unified framework.

Acknowledgements

This research was supported in part by the National Science Foundation under CAREER grant no. IIS 2046086. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[2] Milla Anttila, Keith Ball, and Irini Perissinaki. The central limit problem for convex bodies. *Transactions of the American Mathematical Society*, 355(12):4723–4735, 2003.

[3] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

[4] 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.

[5] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*, 2023.

[6] Sergey G Bobkov and Alexander Koldobsky. On the central limit property of convex bodies. In *Geometric Aspects of Functional Analysis: Israel Seminar 2001-2002*, pages 44–52. Springer, 2003.

[7] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024.

[8] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[9] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[10] Yuansi Chen and Ronen Eldan. Localization schemes: A framework for proving mixing bounds for markov chains. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 110–122. IEEE, 2022.

[11] 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.

[12] Ahmed El Alaoui and Andrea Montanari. An information-theoretic view of stochastic localization. *IEEE Transactions on Information Theory*, 68(11):7423–7426, 2022.

[13] Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Sampling from the sherrington-kirkpatrick gibbs measure via algorithmic stochastic localization. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 323–334. IEEE, 2022.

[14] Ronen Eldan. Thin shell implies spectral gap up to polylog via a stochastic localization scheme. *Geometric and Functional Analysis*, 23(2):532–569, 2013.

[15] Ronen Eldan. Taming correlations through entropy-efficient measure decompositions with applications to mean-field approximation. *Probability Theory and Related Fields*, 176(3):737–755, 2020.

[16] Davide Ghio, Yatin Dandi, Florent Krzakala, and Lenka Zdeborová. Sampling with flows, diffusion, and autoregressive neural networks from a spin-glass perspective. *Proceedings of the National Academy of Sciences*, 121(27):e2311810121, 2024.

[17] Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

[18] Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in gaussian channels. *IEEE transactions on information theory*, 51(4):1261–1282, 2005.

[19] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*, 2022.

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

[21] Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13:541–559, 1995.

[22] 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. *arXiv preprint arXiv:2502.06768*, 2025.

[23] Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *International Conference on Learning Representations*, 2023.

[24] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[25] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

[26] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[27] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024.

[28] Justin Lovelace, Varsha Kishore, Yiwei Chen, and Kilian Q Weinberger. Diffusion guided language modeling. *arXiv preprint arXiv:2408.04220*, 2024.

[29] Andrea Montanari. Sampling, diffusions, and stochastic localization. *arXiv preprint arXiv:2305.10690*, 2023.

[30] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[31] Daniel P Palomar and Sergio Verdú. Gradient of mutual information in linear vector gaussian channels. *IEEE Transactions on Information Theory*, 52(1):141–154, 2005.

[32] Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

[33] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

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

[35] Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

[36] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

[37] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

[38] Qiao Sun, Zhicheng Jiang, Hanhong Zhao, and Kaiming He. Is noise conditioning necessary for denoising generative models? *arXiv preprint arXiv:2502.13129*, 2025.

[39] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024.

[40] 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. *arXiv preprint arXiv:2409.02908*, 2024.

Appendix (for OpenReview)

A Derivations

Symbol	Description
$i = 1, \ldots, L$	Index for sequences of length L
$t \in [0, \infty)$	Index for continuous time dynamics, equal to SNR
$\gamma_i \in [0, \infty)$	Per token SNR, defined on a contour or as function of t
$s_i \in \mathcal{V}$	i-th symbol/token from the vocabulary $\mathcal{V}$
$\boldsymbol{x}_i = \text{enc}(s_i) \in \mathbb{R}^d$	Embed token as vector on unit hyper-sphere surface ($\|\boldsymbol{x}\| = 1$)
$\boldsymbol{z}_i \in \mathbb{R}^d$	Noisy embedding of the i-th token at noise level t
$d\boldsymbol{z} = \boldsymbol{x}\, dt + dW$	Conditional SDE for dynamics
$\boldsymbol{z} = t\, \boldsymbol{x} + \sqrt{t}\, \boldsymbol{\epsilon}$	Sample conditional marginal
$\boldsymbol{z} \sim \mathcal{N}(t\, \boldsymbol{x}, t)$	Alternate form to sample conditional marginal
$d\boldsymbol{z} = \hat{\boldsymbol{x}}(\boldsymbol{z})\, dt + dW$	Unconditional SDE with equivalent dynamics to conditional SDE
$\hat{\boldsymbol{x}}(\boldsymbol{z}, t) = \mathbb{E}[\boldsymbol{x}\|\boldsymbol{z}(t)]$	Optimal denoiser would generally depend on time or noise level
$\hat{\boldsymbol{x}}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x} e^{\boldsymbol{x}\cdot\boldsymbol{z}}]/\mathbb{E}_{\boldsymbol{x}}[e^{\boldsymbol{x}\cdot\boldsymbol{z}}]$	Key result: optimal denoiser doesn't depend on time for spherical embeddings
$\hat{\boldsymbol{x}}(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \text{LSE}_{\boldsymbol{x}\in\mathcal{D}}(\boldsymbol{z}\cdot\boldsymbol{x})$	For empirical distribution, relates to gradient of cumulant generating function, or modern Hopfield energy[33]
$-\log P(\boldsymbol{s}) = -\log P(\boldsymbol{x})$ $= {}^1\!/_2 \int_0^\infty dt\, \mathbb{E}_{\boldsymbol{z}(t)\|\boldsymbol{x}}[\|\boldsymbol{x} - \hat{\boldsymbol{x}}(\boldsymbol{z})\|^2]$	Probability relates to MMSE, for any one-to-one embedding [18, 23]

Table 2: Summary of notation and key relations.

A.1 Optimal Denoiser is SNR invariant

We now derive the optimal denoiser for the noise channel with per token SNR described in the main text. The denoiser is as follows, where we first re-write with Bayes rule, then expand the Gaussian noise channel.

$$\begin{aligned}
\hat{\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{\gamma}) \equiv \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z})}[\boldsymbol{x}] &= \frac{\sum_{\boldsymbol{x}} p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) P(\boldsymbol{x}) \boldsymbol{x}}{p_{\boldsymbol{\gamma}}(\boldsymbol{z})} = \frac{\sum_{\boldsymbol{x}} p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) P(\boldsymbol{x}) \boldsymbol{x}}{\sum_{\bar{\boldsymbol{x}}} p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\bar{\boldsymbol{x}}) P(\bar{\boldsymbol{x}})} \\
&= \frac{\sum_{\boldsymbol{x}} \exp(-{}^1\!/_2 \|\boldsymbol{z} - \boldsymbol{\gamma}\boldsymbol{x}\|^2 / \boldsymbol{\gamma}) / \mathcal{Z}(\boldsymbol{\gamma}) P(\boldsymbol{x}) \boldsymbol{x}}{\sum_{\bar{\boldsymbol{x}}} \exp(-{}^1\!/_2 \|\boldsymbol{z} - \boldsymbol{\gamma}\bar{\boldsymbol{x}}\|^2 / \boldsymbol{\gamma}) / \mathcal{Z}(\boldsymbol{\gamma}) P(\bar{\boldsymbol{x}})} \\
&= \frac{\sum_{\boldsymbol{x}} \exp(\boldsymbol{z}\cdot\boldsymbol{x}) P(\boldsymbol{x}) \boldsymbol{x}}{\sum_{\bar{\boldsymbol{x}}} \exp(\boldsymbol{z}\cdot\bar{\boldsymbol{x}}) P(\bar{\boldsymbol{x}})} \\
\hat{\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{\gamma}) = \hat{\boldsymbol{x}}(\boldsymbol{z}) &= \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x} e^{\boldsymbol{x}\cdot\boldsymbol{z}}] / \mathbb{E}_{\boldsymbol{x}}[e^{\boldsymbol{x}\cdot\boldsymbol{z}}]
\end{aligned}$$

The division and multiplication between $\boldsymbol{\gamma}$ and $\boldsymbol{z}$ should be understood to be applied component-wise, per token. We canceled out the $\|\boldsymbol{z}\|$ term, and due to the normed embedding, $\exp(-{}^1\!/_2 \boldsymbol{\gamma} \|\boldsymbol{x}\|^2)$ terms become constant and cancel out, leaving us with no dependence on $\boldsymbol{\gamma}$ at all. The distribution $P(\boldsymbol{x}) e^{\boldsymbol{z}\cdot\boldsymbol{x}} / \mathcal{Z}(\boldsymbol{z})$ is sometimes referred to as an exponentially tilted distribution of $P(\boldsymbol{x})$. This form also can be written as $\nabla_z \log \mathbb{E}_{\boldsymbol{x}}[e^{\boldsymbol{x}\cdot\boldsymbol{z}}]$, which is a gradient of the cumulant generating function.

The results above also hold for the simpler case where $\gamma_i(t) = t$, leading to a denoiser which is invariant to t. Recent work has also empirically noted that even for traditional diffusion models, noise conditioning is not always necessary or useful[38].

Conditional denoisers Replacing $P(\boldsymbol{x})$ above with $P(\boldsymbol{x}|\boldsymbol{y})$ gives the conditional denoiser, and we still get a result invariant to SNR.

$$\hat{\boldsymbol{x}}(\boldsymbol{z}, y, \boldsymbol{\gamma}) = \hat{\boldsymbol{x}}(\boldsymbol{z}, y) = \mathbb{E}_{P(\boldsymbol{x}|y)}[\boldsymbol{x} e^{\boldsymbol{x}\cdot\boldsymbol{z}}]/\mathbb{E}_{P(\boldsymbol{x}|y)}[e^{\boldsymbol{x}\cdot\boldsymbol{z}}]$$

In Equation (4), we used the following result to show the connection between probability estimates using certain contours and autoregressive models.

$$\begin{aligned} -\log P(\boldsymbol{x}) &= \frac{1}{2}\int_{C_{AR}} \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) \cdot d\boldsymbol{\gamma} = \frac{1}{2}\sum_{i=1}^{L}\int_{C_i} \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) \cdot d\boldsymbol{\gamma} \\ &= \sum_{i=1}^{L}\frac{1}{2}\int_0^\infty d\gamma_i E_i(\boldsymbol{x}, (\infty, \dots, \gamma_i, 0, \dots)) = -\sum_{i=1}^{L}\log P(\boldsymbol{x}_i|\boldsymbol{x}_{<i}) \end{aligned}$$

To show this, we need to demonstrate the last equality explicitly.

$$-\log P(\boldsymbol{x}_i \mid \boldsymbol{x}_{<i}) = \frac{1}{2}\int_0^\infty d\gamma_i \; E_i\left(\boldsymbol{x}, \boldsymbol{\gamma} = (\infty, \dots, \gamma_i, 0, \dots)\right) \tag{7}$$

First, we rewrite the left hand side.

$$\begin{aligned} \frac{1}{2}\int_0^\infty d\gamma_i \; E_i\left(\boldsymbol{x}, \boldsymbol{\gamma} = (\infty, \dots, \gamma_i, 0, \dots)\right) &= \frac{1}{2}\int_0^\infty d\gamma_i \; \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}\left[\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i(\boldsymbol{z})\|^2\right] \\ &= \frac{1}{2}\int_0^\infty d\gamma_i \; \mathbb{E}_{p_{\gamma_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}\left[\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i(\boldsymbol{x}_{<i}, \boldsymbol{z}_i, 0, \dots)\|^2\right] \end{aligned}$$

In the last line, we note that for tokens at SNR 0, there is no information, so the optimal denoiser will be conditioned on a constant, zero. For tokens with infinite SNR, $\boldsymbol{z}_{<i}$ is equivalent in distribution to $\boldsymbol{x}_{<i}$, which we replace in the argument, with a slight abuse of notation.

Next, we see what happens when we try to write $P(\boldsymbol{x}_i|\boldsymbol{x}_{<i})$, in terms of the optimal *conditional denoiser* for denoising only $\boldsymbol{z}_i$, using Equation (3), where the contours simplify to 1-d integrals as we only have one SNR, γ_i. In that case we know we can write the probability in terms of the conditional denoiser, which we will call $\tilde{\boldsymbol{x}}(\boldsymbol{z}_i, \boldsymbol{x}_{<i})$ to distinguish from $\hat{\boldsymbol{x}}$.

$$-\log P(\boldsymbol{x}_i|\boldsymbol{x}_{<i}) = \frac{1}{2}\int_0^\infty d\gamma_i \; \mathbb{E}_{p_{\gamma_i}(\boldsymbol{z}_i|\boldsymbol{x}_i)}[\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i(\boldsymbol{z}_i, \boldsymbol{x}_{<i})\|^2]$$

We can see that this matches the expression above, except for the denoisers. These denoisers are conditioned on the same information, and have to be MMSE denoisers for the same denoising problem, and therefore they must be equivalent, and equality of Equation (7) must hold.

A.2 Probability with Arbitrary SNR Paths

We make use of the following theorem, re-stated with our setting and notation, from [31], to prove Equation (5) and Equation (3).

Theorem A.1 ([31, Thm. 5]) *Consider the signal model*

$$\boldsymbol{z}_i = \gamma_i \boldsymbol{x}_i + \sqrt{\gamma_i}\boldsymbol{\epsilon}_i,$$

where $\boldsymbol{x}$ is arbitrarily distributed with finite second-order moments, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \mathbf{I})$ is independent Gaussian noise.

Then the gradient of the Kullback–Leibler divergence between the conditional output distribution $p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})$ and the unconditional output distribution $p_{\boldsymbol{\gamma}}(\boldsymbol{z})$ is

$$\nabla_{\boldsymbol{\gamma}} D\big(p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p_{\boldsymbol{\gamma}}(\boldsymbol{z})\big) = {}^1\!/_2 \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}),$$

where $E_i(\boldsymbol{x}, \boldsymbol{\gamma}) \equiv \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}[\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i(\boldsymbol{z})\|^2]$, and $\hat{\boldsymbol{x}}(\boldsymbol{z}) = \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z})}[\boldsymbol{x}]$ is the MMSE estimator.

We use this theorem in conjunction with the fundamental theorem of calculus for line integrals to obtain the following.

$$\int_C d\boldsymbol{\gamma} \cdot \nabla_{\boldsymbol{\gamma}} D\big(p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p_{\boldsymbol{\gamma}}(\boldsymbol{z})\big) = 1/2 \int_C d\boldsymbol{\gamma} \cdot \mathbf{E}(\boldsymbol{x}, \boldsymbol{\gamma}) = D\big(p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p_{\boldsymbol{\gamma}}(\boldsymbol{z})\big)\Big|_{\boldsymbol{\gamma}_0}^{\boldsymbol{\gamma}_1} \tag{8}$$

Here, $\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1$ define the endpoints of the contour, which must be piecewise smooth. Next, let's consider some of the limits.

$$\lim_{\boldsymbol{\gamma} \to 0} D\big(p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p_{\boldsymbol{\gamma}}(\boldsymbol{z})\big) = 0 \tag{9}$$

$$\lim_{\boldsymbol{\gamma} \to \infty} D\big(p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p_{\boldsymbol{\gamma}}(\boldsymbol{z})\big) = -\log P(\boldsymbol{x}) \tag{10}$$

The second limit can be observed as follows.

$$\begin{aligned} D\big(p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p_{\boldsymbol{\gamma}}(\boldsymbol{z})\big) &= \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\right] \\ &= \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{P(\boldsymbol{x}|\boldsymbol{z})}{P(\boldsymbol{x})}\right] \\ &= \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log P(\boldsymbol{x}|\boldsymbol{z})\right] - \log P(\boldsymbol{x}) \end{aligned} \tag{11}$$

Because $P(\boldsymbol{x})$ is discrete, $\lim_{\boldsymbol{\gamma} \to \infty} P(\mathbf{x} \mid \mathbf{z}) = 1$ almost surely.

Combining Equation (11) and Equation (8), we obtain the following.

$$-\log P(\boldsymbol{x}) = 1/2 \int_{C_{\boldsymbol{\gamma}}} d\bar{\boldsymbol{\gamma}} \cdot \mathbf{E}(\boldsymbol{x}, \bar{\boldsymbol{\gamma}}) - \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log P(\boldsymbol{x}|\boldsymbol{z})\right] \tag{12}$$

Where the curve $C_{\boldsymbol{\gamma}}$ is piecewise smooth, begins at $\boldsymbol{\gamma} = 0$ and ends at $\boldsymbol{\gamma}$. We can use this result to obtain Equation (5) for a path that starts at zero and ends at t, and Equation (3) when we let the end of the contour go to infinity. ■

A.3 Log-likelihood Upper Bound

In practice, the model learns a distribution $Q_\theta(\boldsymbol{x}|\boldsymbol{z})$, then we derive an upper bound on the Equation (12).

$$-\log P(\boldsymbol{x}) \leq 1/2 \int_{C_{\boldsymbol{\gamma}}} d\bar{\boldsymbol{\gamma}} \cdot \mathbf{E}(\boldsymbol{x}, \bar{\boldsymbol{\gamma}}) - \mathbb{E}_{p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log Q_\theta(\boldsymbol{x}|\boldsymbol{z})\right] \tag{13}$$

To upper bound the expected negative log likelihood on the data, in Equation (13) we can use any denoiser in the first term whose error will, by definition, be an upper bound on the MMSE. Likewise, for the second term, we can also use any estimate, since the cross entropy will upper bound the true entropy. The simplest choice would be to factorize, $\hat{P}(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{L} Q_\theta(\boldsymbol{x}_i|\boldsymbol{z})$, since our model already outputs an estimate of $Q_\theta(\boldsymbol{x}_i|\boldsymbol{z})$. If t^* represents a high enough SNR, then the second term is small and close to the factorized approximation.

A.4 Equivalence of conditional and unconditional dynamics

Consider the marginals generated conditioned on the data with the following noise channel.

$$p_{\boldsymbol{\gamma}}(\boldsymbol{z}) \equiv \sum_{\boldsymbol{x}} p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) P(\boldsymbol{x})$$

Where $p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x})$ is a Gaussian noise channel, which could be defined as follows,

$$\boldsymbol{z}_i = \gamma_i \, \boldsymbol{x}_i + \sqrt{\gamma_i} \, \boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I).$$

This is equivalent to the conditional SDE dynamics,

$$d\boldsymbol{z}_i = \boldsymbol{x}_i \dot{\gamma}_i \, dt + \sqrt{\dot{\gamma}_i} dW_i, \;\; \boldsymbol{x} \sim P(\boldsymbol{x}), \tag{14}$$

as can be seen from direct integration. We can use the Fokker-Planck equation to understand the marginal dynamics of this process. Consider a particular contour, $\boldsymbol{\gamma}(t)$, as defined in the text, which goes from zero to infinity. At $\boldsymbol{\gamma}(0) = 0$, we have

$$p_{\boldsymbol{\gamma}(0)}(\boldsymbol{z}) = \delta(\boldsymbol{z}).$$

The Fokker-Planck equation gives the following for the conditional dynamics.

$$\frac{\partial}{\partial t} p_{\boldsymbol{\gamma}(t)}(\boldsymbol{z} \mid \boldsymbol{x}) = -\sum_i \nabla_{\boldsymbol{z}_i} \cdot (\dot{\gamma}_i \, \boldsymbol{x} \, p_{\boldsymbol{\gamma}}(\boldsymbol{z} \mid \boldsymbol{x})) + \sum_i \dot{\gamma}_i \Delta_{\boldsymbol{z}_i} \, p_{\boldsymbol{\gamma}}(\boldsymbol{z} \mid \boldsymbol{x})$$

By marginalizing over $P(\boldsymbol{x})$, get the following.

$$\frac{\partial}{\partial t} p_{\boldsymbol{\gamma}(t)}(\boldsymbol{z}) = \sum_{\boldsymbol{x}} P(\boldsymbol{x}) \frac{\partial}{\partial t} p_{\boldsymbol{\gamma}(t)}(\boldsymbol{z} \mid \boldsymbol{x}) \tag{15}$$

$$= -\sum_i \nabla_{\boldsymbol{z}_i} \cdot \left(\dot{\gamma}_i \sum_{\boldsymbol{x}} P(\boldsymbol{x}) \boldsymbol{x}_i \, p_{\boldsymbol{\gamma}}(\boldsymbol{z} \mid \boldsymbol{x}) \right) + \sum_i \dot{\gamma}_i \Delta_{\boldsymbol{z}_i} \, p_{\boldsymbol{\gamma}}(\boldsymbol{z}) \tag{16}$$

$$= -\sum_i \nabla_{\boldsymbol{z}_i} \cdot (\dot{\gamma}_i \, \hat{\boldsymbol{x}}_i(\boldsymbol{z}) \, p_{\boldsymbol{\gamma}}(\boldsymbol{z})) + \sum_i \dot{\gamma}_i \Delta_{\boldsymbol{z}_i} \, p_{\boldsymbol{\gamma}}(\boldsymbol{z}) \tag{17}$$

In the last line, we used the definition of the optimal denoiser in terms of the posterior mean.

The goal is to design an unconditional dynamics that obeys the same evolution of the marginal. The unconditional dynamics are defined as follows.

$$d\boldsymbol{z}_i = \hat{\boldsymbol{x}}_i(\boldsymbol{z}) \dot{\gamma}_i \, dt + \sqrt{\dot{\gamma}_i} dW_i, \tag{18}$$

Applying the Fokker-Planck equation to this expression directly matches Equation (17). ■

A.5 Zero Prior Mismatch Error in DSL

In diffusion models, the forward process during training pushes data x toward an approximately Gaussian terminal distribution q_T, while the generation phase forcibly begins from a standard Gaussian $\mathcal{N}(0, I)$, resulting in an unavoidable "prior mismatch" between distribution q_T and $\mathcal{N}(0, I)$. In contrast, DSL uses the same distribution $\delta_0(z)$ (we don't use q_T here to avoid notation confusion) for both conditional (forward) and unconditional (reverse) phase, ensuring no initialization error between endpoint forward distribution and the starting point generation distribution, thereby eliminating this prior mismatch term. Rigorously, for any generative SDE (diffusion models and stochastic localization models), the distributional divergence between the generated distribution $\hat{q}$ and the data distribution p_{data} can be decomposed as[1]

$$\mathcal{E}\big(\hat{q} \,\|\, p_{\text{data}}\big) \lesssim \underbrace{\mathcal{E}\big(q_{\text{minSNR}} \,\|\, p_{\text{prior}}\big)}_{\textit{prior mismatch}} + \mathcal{E}_{\text{net}} + \mathcal{E}_{\text{disc}} + \mathcal{E}_{\text{early-stop}}, \tag{19}$$

where q_{minSNR} is the endpoint distribution of the *forward* noising process where the signal to noise ratio is minimized. In diffusion models, $q_{\text{minSNR}} = q_T$ approximates $\mathcal{N}(0, I)$; in stochastic localization models, $q_{\text{minSNR}} = \delta_0(z)$. p_{prior} is the distribution from which the *reverse* SDE is initialized. $\mathcal{E}_{\text{net}}$ represents the network approximation error for the noise, score or data in diffusion and stochastic localization models. $\mathcal{E}_{\text{disc}}$ stands for the discretization error of the reverse SDE. $\mathcal{E}_{\text{early-stop}}$ is the error for early stopping in implementation. In stochastic localization, $q_{\text{minSNR}} = p_{\text{prior}} = \delta_0(z)$, thus term $\mathcal{E}(q_{\text{minSNR}} \,\|\, p_{\text{prior}})$ vanishes, making the distributional error bound tighter in stochastic localization models. Note though Montanari [29] proves diffusion models and stochastic localization are equivalent under a time change, it is in the limit setting where $T \to \infty$. However, the distributional error bound is derived from a practical perspective when the limit can never be achieved. Therefore, our analysis does not conflict with the result in [29].

[1] See, e.g., [5] for a derivation with KL divergence.

A.6 Prior Mismatch Scaling vs. Drift-Driven Identification

In high-dimensional settings ($d \gg 1$), diffusion models suffer from a prior mismatch [5], yet this term can be driven till very small by choosing a sufficiently large terminal time T, rendering its impact negligible. Stochastic Localization (SL), by contrast, is immune to prior mismatch, but the network must infer x instantaneously in the vanishing-noise limit $s \to 0$; since all information must be injected via the drift (also known as denoiser), the resulting training problem is considerably more challenging. In Section 3 we describe how to design our training schedule to overcome this challenge.

A.7 Lipschitz Continuity of the Drift in Unconditional SDE

Lemma A.1 *Let the drift (MMSE denoiser) of the unconditional SDE be defined by*

$$\hat{\boldsymbol{x}}(\boldsymbol{z}) = \mathbb{E}_{P(\boldsymbol{x})}[\boldsymbol{x} e^{\boldsymbol{x}\cdot\boldsymbol{z}}]/\mathbb{E}_{P(\boldsymbol{x})}[e^{\boldsymbol{x}\cdot\boldsymbol{z}}] \qquad z \in \mathbb{R}^d,$$

where the data distribution $p(\boldsymbol{x})$ is supported on the unit sphere $\|\boldsymbol{x}\| = 1$. Then $\hat{\boldsymbol{x}}(\cdot)$ is 1-Lipschitz; that is,

$$\forall \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^d : \quad \|\hat{\boldsymbol{x}}(\boldsymbol{z}_1) - \hat{\boldsymbol{x}}(\boldsymbol{z}_2)\| \;\le\; \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|.$$

Define the log–partition function

$$\phi(\boldsymbol{z}) \;=\; \log \mathbb{E}_{p(\boldsymbol{x})}\big[e^{\boldsymbol{x}\cdot\boldsymbol{z}}\big],$$

so that $\hat{\boldsymbol{x}}(\boldsymbol{z}) = \nabla_{\boldsymbol{z}}\phi(\boldsymbol{z})$.

Because $e^{\boldsymbol{x}\cdot\boldsymbol{z}}$ is convex in $\boldsymbol{z}$ and the expectation preserves convexity, ϕ is convex. Its Hessian equals the covariance of $\boldsymbol{x}$ under the Gibbs posterior $p(\boldsymbol{x} \mid \boldsymbol{z}) \propto p(\boldsymbol{x}) e^{\boldsymbol{x}\cdot\boldsymbol{z}}$:

$$H(\boldsymbol{z}) \;=\; \nabla^2_{\boldsymbol{z}}\phi(\boldsymbol{z}) \;=\; \mathrm{Cov}_{p(\boldsymbol{x}|\boldsymbol{z})}\big(\boldsymbol{x}\big).$$

For any unit vector $v \in \mathbb{R}^d$,

$$v^\top H(\boldsymbol{z})\, v \;=\; \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{z})}\big[(v\cdot\boldsymbol{x})^2\big] - \Big(\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{z})}[v\cdot\boldsymbol{x}]\Big)^2 \;\le\; \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{z})}\big[(v\cdot\boldsymbol{x})^2\big] \;\le\; \|v\|^2 \;=\; 1,$$

where the last inequality uses $\|\boldsymbol{x}\| = 1$, therefore, spectral norm satisfies $\|H(\boldsymbol{z})\| \le 1$.

Since $\nabla^2_{\boldsymbol{z}}\phi$ is bounded by 1 everywhere, the gradient $\nabla_{\boldsymbol{z}}\phi = \hat{\boldsymbol{x}}$ is 1-Lipschitz:

$$\|\hat{\boldsymbol{x}}(\boldsymbol{z}_1) - \hat{\boldsymbol{x}}(\boldsymbol{z}_2)\| \;\le\; \int_0^1 \big\|H\big(\boldsymbol{z}_2 + t(\boldsymbol{z}_1 - \boldsymbol{z}_2)\big)\big\|\, \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|\, dt \;\le\; \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|.$$

Thus the empirical observation $\|\hat{\boldsymbol{x}}(\boldsymbol{z}_1) - \hat{\boldsymbol{x}}(\boldsymbol{z}_2)\| \le K\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|$ with $K < 1$ is justified. ■.

A.8 Curvature of SDE

For completeness we derive the directional curvature term κ_i that drives the adaptive step sizes in Algorithm 3 (main text, Eq. 25).

Fix a point $\boldsymbol{z}$ and a small scalar step size $h > 0$, perform a first-order Taylor expansion of the drift along its own direction,

$$\hat{\boldsymbol{x}}\big(\boldsymbol{z} + h\,\hat{\boldsymbol{x}}(\boldsymbol{z})\big) = \hat{\boldsymbol{x}}(\boldsymbol{z}) \;+\; h\, \underbrace{\big(\hat{\boldsymbol{x}}(\boldsymbol{z})^\top \nabla_{\boldsymbol{z}}\big)\hat{\boldsymbol{x}}(\boldsymbol{z})}_{\text{directional derivative}} + O(h^2). \tag{20}$$

Re-arranging and dividing by h gives the finite-difference approximation,

$$\frac{\hat{\boldsymbol{x}}\big(\boldsymbol{z} + h\,\hat{\boldsymbol{x}}(\boldsymbol{z})\big) - \hat{\boldsymbol{x}}(\boldsymbol{z})}{h} \;\approx\; \big(\hat{\boldsymbol{x}}(\boldsymbol{z})^\top \nabla_{\boldsymbol{z}}\big)\hat{\boldsymbol{x}}(\boldsymbol{z}). \tag{21}$$

Taking the limit $h \to 0$ identifies the right-hand side with the material (total) time derivative of the drift evaluated along the trajectory of Unconditional SDE,

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{x}}\big(\boldsymbol{z}(t)\big) \;=\; \big(\hat{\boldsymbol{x}}(\boldsymbol{z})^\top \nabla_{\boldsymbol{z}}\big)\hat{\boldsymbol{x}}(\boldsymbol{z}).$$

Algorithm 1 Curvature Estimation

Require: Embedding tensor $z \in \mathbb{R}^{B\times T\times D}$, differentiable model $\mathrm{Denoiser}(\cdot)$
Ensure: Curvature tensor $\kappa \in \mathbb{R}^{B\times T\times 1}$
$(B, T, D) \leftarrow \mathrm{shape}(z)$
$z_{\mathrm{flat}} \leftarrow \mathrm{reshape}(z,\ (B,\ T \cdot D))$
$z_{\mathrm{flat}} \leftarrow z_{\mathrm{flat}}.\mathrm{detach}();\quad z_{\mathrm{flat}}.\mathrm{requires\_grad\_}(True)$
$f_{\mathrm{flat}} \leftarrow \mathrm{reshape}\big(\mathrm{Denoiser}(\mathrm{reshape}(z_{\mathrm{flat}}, (B, T, D))),\ (B,\ T \cdot D)\big)$
$(\_, v) \leftarrow \mathrm{jvp}\big(\ \lambda x : \mathrm{Denoiser}(\mathrm{reshape}(x, (B, T, D))),\ (z_{\mathrm{flat}},),\ (f_{\mathrm{flat}},)\big)$
$v \leftarrow \mathrm{reshape}(v,\ (B,\ T,\ D))$
$\kappa \leftarrow \| v \|_{\mathrm{dim}=D} \in \mathbb{R}^{B\times T\times 1}$
return κ

We define the token-wise curvature,

$$\kappa_i(\boldsymbol{z}) \;=\; \Big|\tfrac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{x}}_i\big(\boldsymbol{z}(t)\big)\Big| \;=\; \big|\hat{\boldsymbol{x}}(\boldsymbol{z})\cdot\nabla_{\boldsymbol{z}}\hat{\boldsymbol{x}}_i(\boldsymbol{z})\big|, \qquad i=1,\dots,L, \tag{22}$$

which measures how fast the prediction for the i-th token changes as one moves along the instantaneous velocity field $\hat{\boldsymbol{x}}(\boldsymbol{z})$. Eq. 22 is exactly Eq. 25 in the main text.

In Algorithm 3 we approximate $\kappa_i(\boldsymbol{z})$ via a single reverse-mode Jacobian–vector product and choose the per-token step size $\Delta t_i \propto 1/(\kappa_i + \varepsilon)$ so that directions with large curvature receive smaller forward SNR increments. Because the extra Jacobian–vector product shares activations with the denoiser call, the overhead is negligible in practice.

Discretize Unconditional SDE with Euler–Maruyama scheme on an adaptive grid yields, for each token,

$$\boldsymbol{z}_{n+1,i} \;=\; \boldsymbol{z}_{n,i} + \hat{\boldsymbol{x}}_i(\boldsymbol{z}_n)\,\Delta t_i + \sqrt{\Delta t_i}\,\xi_{n,i}, \qquad \xi_{n,i}\sim\mathcal{N}(0,1),$$

ensuring a uniform local truncation error over the sequence whenever Δt_i scales like κ_i^{-1}.

Details in Algorithm 1.

B More Implementation Details

In this section, we present a series of implementation design choices for the DSL framework Equation (Conditional SDE) and Equation (Unconditional SDE), and also the corresponding sampling algorithms.

B.1 Training

Learned embeddings and softmax converter Building on recent advances in continuous-diffusion language models [11, 17], we decouple the learnable token-embedding space from the hidden dimensionality of the backbone transformer, which is quite different from autoregressive language models where the embedding operation is simply the first layer of the nerual network.

In practice, we constrain our token embeddings to lie on a unit hypersphere, as mentioned in Section 2, which guarantees that our denoiser remains invariant to SNR and allows us to learn the discrete token distribution by injecting and removing noise directly in embedding space. Since these token embeddings live in a space whose dimensionality differs from the transformer's latent space, the previous continuous-diffusion language models apply a simple linear projection to perform this mapping. We found that this mapping falls short.

We trace the difficulty in training previous continuous diffusion language models to an inductive bias in the attention mechanism: the attention is ill-conditioned when faced with high-variance Gaussian noise in the noisy embeddings. To overcome this inductive bias, we replace the naive linear projection with a softmax converter block that first maps each noisy vector to a vocabulary-level probability distribution and then re-embeds it into the transformer's latent space. This yields input representations that are far more compatible with standard attention mechanisms, improving training stability and likelihood.

$$\mathrm{SoftmaxConvert}(\boldsymbol{z}) \;=\; W_{\mathrm{transformer}}\,\mathrm{softmax}\big(\beta\,\boldsymbol{z}\,W_{\mathrm{token}}^{\top}\big) \tag{23}$$

Algorithm 2 Per-Sentence SNR Sampler

Require: Batch size B, sequence length L, uniform time grid $\{t_0, \ldots, t_N\}$
Ensure: Final state z_N
1: Sample $z_0 = 0 \in \mathbb{R}^{B \times L \times D}$
2: **for** $n = 0$ **to** $N - 1$ **do**
3: $\quad \Delta t \leftarrow t_{n+1} - t_n$
4: $\quad \hat{x}_n \leftarrow \text{Denoiser}(z_n)$
5: $\quad z_{n+1} \leftarrow z_n + \hat{x}_n \, \Delta t + \sqrt{\Delta t} \, \xi_n$, where $\xi_n \sim \mathcal{N}(0, I)$ and $\Delta W = \sqrt{\Delta t} \, \xi_n$
6: **end for**
7: **return** z_N

where V is the vocabulary size, $\boldsymbol{z} \in \mathbb{R}^d$ is noisy embedding, $W_{\text{token}} \in \mathbb{R}^{V \times d}$ is token embedding weight matrix, β is a learnable temperature parameter which controls the distribution shape of the softmax (with initial value to be $\frac{1}{\sqrt{d}}$), $W_{\text{transformer}} \in \mathbb{R}^{d_{\text{transformer}} \times V}$ transforms to an embedding with dimensions for the transformer, which need not be constrained to a hypersphere.

Cross-entropy loss and categorical reparameterization Recall that minimizing the mean-square error (MMSE) also minimizes the negative log-likelihood in Eq. 3, which might suggest using an MSE loss under a uniform SNR schedule. However, this induces a trivial "shortcut" solution whereby all embeddings collapse to the same point, since then the denoiser can perfectly predict their common mean.[2] This joint-collapse issue has also been observed in prior continuous-diffusion language models [17, 8, 25, 11].

Instead, we follow [17] and train our model to output a probability distribution over discrete tokens for each location (via unnormalized logits, as is typical for language models). However, we then exploit the observation that the optimal denoiser, $\hat{\boldsymbol{x}}_i(\boldsymbol{z}) = \mathbb{E}_{P(\boldsymbol{x}_i|\boldsymbol{z})}[\boldsymbol{x}_i]$, depends on the true posterior, $P(\boldsymbol{x}_i|\boldsymbol{z})$. So if we train our model to minimize $CE[P(\boldsymbol{x}_i|\boldsymbol{z}), P_\theta(\boldsymbol{x}_i|\boldsymbol{z})]$, then the optimal distribution will also give us the optimal denoiser (and therefore optimal NLL estimate). Note that the true posterior, like the optimal denoiser, is SNR invariant. Optimizing the cross entropy, rather than the denoiser MSE, avoids the trivial "shortcut" solution, and is more efficient for training transformers.

Training and architecture The goal is to learn the optimal denoiser, $\hat{\boldsymbol{x}}(\boldsymbol{z}) = \mathbb{E}_{P(\boldsymbol{x}|\boldsymbol{z})}[\boldsymbol{x}]$. Because our proposed approach has no time conditioning, it simplifies the architecture, allowing the use of standard transformer encoder models. However, we can also adapt the standard diffusion transformers for easier comparison to existing discrete diffusion results shown in [27].

At first glance, it might seem that since the denoiser is time invariant, we can train to denoise using a $\boldsymbol{z}(t) \sim \mathcal{N}(t \, \boldsymbol{x}, t)$ for noisy samples at a fixed t. While that is true in principle, in practice we would find that our denoiser is trained well in some parts of the space, but poorly in others. Looking at Figure 1b, you can see that different SNR values explore different parts of space, and the magnitude of $|\boldsymbol{z}|$ plays a similar role to SNR. Therefore, we should train our denoiser using multiple SNR values, and sample $\boldsymbol{z}$'s with different magnitudes.

DSL Training Schedule Choice In DSL, the network must infer x in the low-SNR limit, t small, which is considerably more challenging than inference for moderate to large SNR. To mitigate this challenge, we adopt a skewed training schedule by sampling SNR values from a log-normal distribution. More details on training can be found in the Appendix C.

B.2 Sampling Algorithms

In this section, we first talk about a per-sentence SNR sampler which directly discretizes Unconditional SDE, showing that in a similar setting to Plaid [17], we are able to improve the NLL. Then, we move on to a per-token SNR sampler, which enables a brand-new design which covering AR, discrete, and continuous diffusion language models.

[2] Equivalently, joint optimization can drive all learned embeddings to coincide, and the MSE-optimal prediction is simply their shared centroid.

Algorithm 3 Per-Token SNR Sampler with Curvature-Based Adaptive Step Size

Require: Batch size B, sequence length L, number of steps N, maximum SNR $\gamma_{\max}$
Ensure: Final embeddings z_N

Initialize:
  $\gamma_{\min} \leftarrow 0.01, \quad \delta \leftarrow 0.01, \quad \delta_{\max} \leftarrow \gamma_{\max}/N$
Sample $z \sim \mathcal{N}(0, 0) \in \mathbb{R}^{B\times L\times D}$
$\gamma \leftarrow \gamma_{\min}\, \mathbf{1}_{B\times L\times 1}$
for $i = 1$ **to** N **do**
  $\kappa \leftarrow \mathrm{Curvature}(z)$
  $h \leftarrow \min\big(\delta/(\kappa + 10^{-6}),\, \delta_{\max}\big)$
  Sample $\xi \sim \mathcal{N}(0, I)$
  $z \leftarrow z + h \odot \mathrm{Denoiser}(z) + \sqrt{h} \odot \xi$
  $\gamma \leftarrow \gamma + h$
end for
return z

Per-sentence SNR sampler Because Unconditional SDE has unit diffusion coefficient and a drift (also called bias and denoiser) that is *Lipschitz* [3], the Euler–Maruyama scheme offers a stable and unbiased first–order discretisation. The Lipschitz continuity of the drift is formally proven in Appendix A.

Fix a time grid $0 = t_0 < t_1 < \cdots < t_N = T$ and let $\Delta t_n = t_{n+1} - t_n$, starting from the initial state $z_0 = 0$, the Euler–Maruyama discretisation is,

$$z_{n+1} \;=\; z_n \;+\; \hat{x}(z_n)\,\Delta t_n \;+\; \sqrt{\Delta t_n}\,\xi_n, \qquad \xi_n \sim \mathcal{N}(0, I), \quad n = 0, \ldots, N-1. \tag{24}$$

The final iterate z_N is returned as the sentence representation at the terminal time step T. We have this algorithm described in Algorithm. 2.

However, different positions often show different levels of uncertainty: some tokens are almost determined by their context, while others remain ambiguous. Using a uniform step size therefore wastes computation on "easy" tokens or, conversely, under-resolves "hard" ones. To address this, we assign each token its *own* SNR schedule $\gamma_i(t)$ and adapt the step size according to the local vector feild of that token, which is supported by the time-invariant denoiser and arbitrary SNR paths in Section 2.

Per-token SNR sampler with a curvature-based adaptive step size We assign an individual step size Δt to each token and let it adapt automatically to the local geometry of the velocity field. To keep the discretisation error under control, we track a per-token *curvature*,

$$\kappa_i \equiv \left\| \tfrac{d}{dt}\, \hat{\boldsymbol{x}}_i(z) \right\| \approx \left\| \hat{\boldsymbol{x}}(\boldsymbol{z}) \cdot \nabla_{\boldsymbol{z}} \hat{\boldsymbol{x}}_i(\boldsymbol{z}) \right\| \tag{25}$$

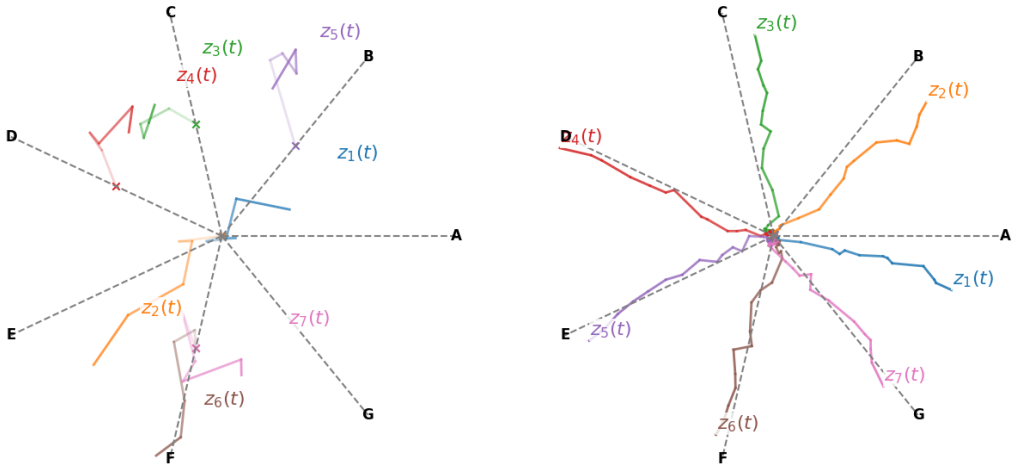
which measures how rapidly the prediction for i-th token changes when we take an infinitesimal step along the current velocity direction. The approximation on the right comes from a first-order Talyor expansion, and a detailed derivation can be found in Appendix A. Tokens that sit in highly curved regions (large κ_i) receive a *smaller* SNR increases, while those in flatter regions have larger increment. Note that computing κ adds negligible overhead, because the Jacobian vector product (JvP) can be obtained in one reverse mode call. The exact numerical scheme is presented in Algorithm 3 [4].

C More Experimental Details

C.1 Cyclic Data Experiments, Inpainting, and Error Correction

One of the benefits of our approach is that we can handle both masked data and "garbled data" or error correction in a single framework. To illustrate this point, we describe a simple experiment involving cyclic data. We imagine that we have K symbols, and sequences of length K that consist of

[3] Empirically we observe $|\hat{x}(z_1) - \hat{x}(z_2)| \leqslant K|z_1 - z_2|$ with $K \approx 1$ after training.
[4] Details of Algorithm 1 can be found in Appendix C



(a) Masked + garbled input

(b) Reconstruction

Figure 2: DSL dynamics using a masked and garbled input: __CDBFF, correctly recovers the original sequence ABCDEF. (Left) shows the initialization and first few steps. Masked values, z_1, z_2 start at zero. Note that z_5, z_7 are initially set to incorrect values. (Right) Shows the correct sequence is recovered after 50 steps of adaptive dynamics.

all cyclic permutations. For K=5, e.g., we'd have "ABCDE, BCDEA, CDEAB, DEABC, EABCD". Then, we simply embed each token into a two-dimensional space, with embeddings equally spaced. Figure 1b shows an example. In this case, we can use the analytic denoiser, $\hat{\boldsymbol{x}}(\boldsymbol{z}) = \mathbb{E}_{P(\boldsymbol{x}|\boldsymbol{z})}[\boldsymbol{x}]$, where we explicitly take the expectation over all data samples. Then we sample by simulating the unconditional SDE dynamics as described in Section 3.

Suppose that the true sequence is "ABCDEFG", but we receive a garbled version, where some of the characters are masked, and some (but we don't know which ones) are garbled.

Original:	A	B	C	D	E	F	G
Input:	$\underbrace{\_}_{\text{masked}}$	$\underbrace{\_}_{\text{masked}}$	$\underbrace{C}_{\text{correct}}$	$\underbrace{D}_{\text{correct}}$	$\underbrace{B}_{\text{garbled}}$	$\underbrace{F}_{\text{correct}}$	$\underbrace{F}_{\text{garbled}}$

Masked diffusion models (MDMs) could handle the masked values, but would have a hard time correcting the garbled tokens, as MDMs never change an unmasked token. YW: MDMs have carry-over the generated tokens. Of course we could mask all tokens before sending it to the MDM, but then they wouldn't take advantage of the signal that remains in the garbled string. Conversely, diffusion models based on uniform character noise could handle garbling, but they would have to fill in the masks with random characters before running. This would make it impossible to leverage the fact that some characters have information and some don't.

In DSL, it is trivial to handle situations such as these. We can assign an $SNR_i = 0$ to the masked tokens. This is equivalent to setting $\boldsymbol{z}_i = 0$. For the other tokens, we can "soft mask" by setting some small SNR, like SNR=0.5, $\boldsymbol{z}_i = 0.5\,\mathrm{enc}(s_i)$. That way, the model has some noisy information about the true value, but it is still possible for the dynamics to fix an incorrect value.

For this example: 3 correct, 2 garbled (in unknown locations), 2 masked, if we run the adaptive sampler many times we get 100 percent of the answers to be correctly sampled cyclic sequences, and about 50 percent are exactly equal to the original input.

C.2 Training Schedule Choice

In Section 3, we talk about the implementation details about DSL. The generation process of Stochastic localization (SL) starts from low-SNR limit, where $t = 0$, therefore, for training we should focus more on the low-SNR region. We show an example in Figure. **??**, which satisfy the above requirements. Notice that this is very different from the traditional diffusion model training schedule

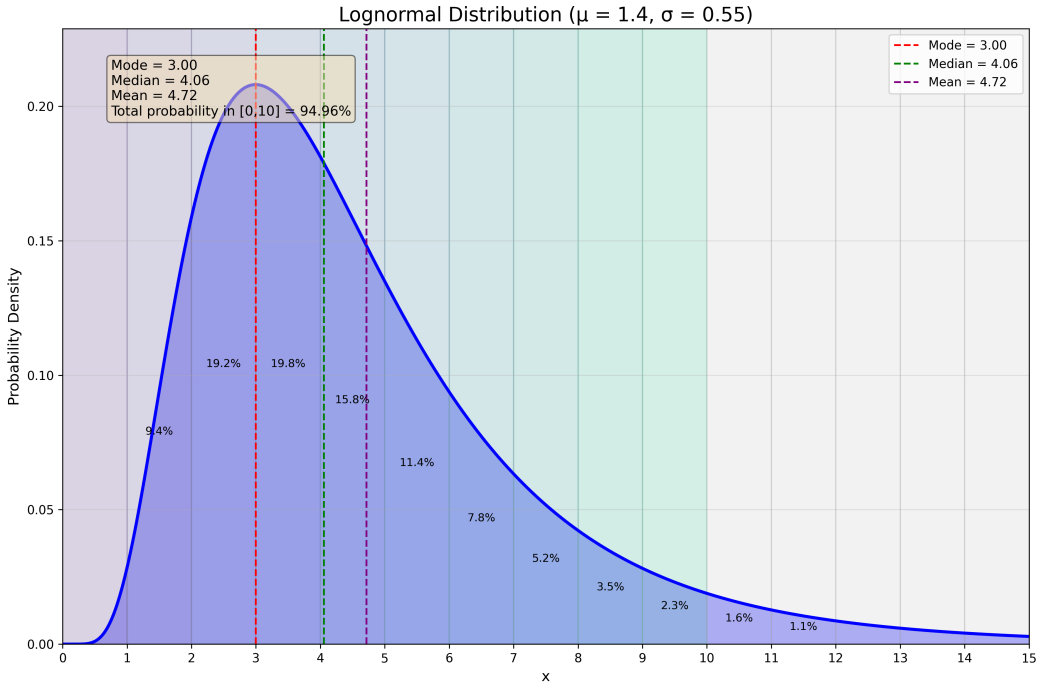


Figure 3: Log-normal Distribution Choice.

design, since diffusion usually starts from a standard Gaussian which carries more information compared to the $t = 0$ in SL.

C.3 Additional Experiment Deatils

We follow all the common practices in [4, 27, 36] to conduct Text8 experiments, which has a dictionary size of 35. We adhere to the standard dataset split and train MDLM [34] using a standard 12-layer transformer architecture. The transformer also has the same number of heads (12) and hidden dimension (784) as in [4]. The model is trained on text chunks of length 256 for 0.5 million steps with batch size 512. Following previous work, we use the cosine learning rate schedule with a linear warm-up of 2000 steps. We conduct optimization with AdamW and a learning rate of 3e-4, and adopt a weight decay factor of 0.03.

C.4 Analysis and Discussion

Language models can be categorized into three approaches with distinct objectives and mechanisms: Autoregressive (AR) models employ Transformers for multiclass classification, predicting $P(x_t|x_{<t})$ through a clean probability chain that enables intuitive but sequential generation. Discrete diffusion models learn at each time-step t how to unmask α_t percentage tokens given a certain context [4, 27, 34, 36]] (where the context is more random and harder to learn compare to AR's context). Continuous diffusion models utilize Transformers as implicit regression models to estimate noise components in continuous embedding space ($\epsilon_\theta(\mathbf{z}_t, t) \approx \epsilon$). The difficulties here can be summarized into the following categories: 1) empirically people find it is harder for Transformers to do regression tasks comparing to classification tasks; 2) previous continuous diffusion language models have to condition on timestep t, whereas discrete diffusion models, especially masked diffusion models inherently know their "timestep" which is given by the number of unmasked token at t; In our work, the softmax converter is designed to deal with the difficulty of predicting from a nearly Gaussian noise to a specific token, making the implicit regression task easier for Transformers to learn. Additionally, we have a time-invariant denoiser, which enables easier training and flexible sampling designs. Thus, our DSL improves the previous continuous diffusion language models.

D Related Work

Non-AutoRegressive(NAR) generation of discrete data via diffusion Diffusion models have become the dominant approach for NAR generation, and discrete data was considered even in the original paper[37]. After [20] used diffusion models to revolutionize image generation, a push to use these methods to speed up discrete tasks like language modeling naturally followed [17, 11, 9, 25].

Discrete Diffusion Models Applying Gaussian diffusion on discrete data has proven difficult, so a parallel research track explored the idea that for discrete data a fundamentally discrete noise model should be considered [4]. In recent years, discrete noise diffusion has dominated in empirical performance, though still lagging behind autoregressive approaches, thanks to a shift in focus to *masking* noise that can efficiently leverage BERT-like[19] transformer architectures [27, 34, 36, 30]. These approaches are fundamentally random order autoregressive models which are sensitive to the order [22, 32] and parallelizing generation has been shown to reduce diversity in generation [40].

Bridging the gap between AutoRegressive(AR) and diffusion Attempts to combine the strengths of diffusion and AR models are proliferating[7, 3, 24, 39, 28]. Most relevant to DSL are recent works that recognize that discrete noise is fundamentally a limiting behavior of Gaussian noise [35, 16]. We make this more explicit, by showing that we can sample or estimate likelihood using arbitrary SNR paths that include traditional Gaussian diffusion and AR paths as special cases. While Flow Matching (FM)[26] and Stochastic Interpolants[1] also generalize the paths in distribution space, they can only consider a single, fixed path for training and inference. In DSL, we show that a universal, SNR-invariant denoiser can be used for sampling or likelihood estimation on any path. Essentially, we are matching marginals on an entire manifold of probability space, rather than a single path.

Stochastic Localization Stochastic localization was first developed in [14] with the idea to use Brownian motion to gradually tilt the probability measure. It is an extension of deterministic localization proposed in [21] for isoperimetric inequalities, and connects such inequalities with open problems like Kannan–Lovasz–Simonovits (KLS) conjecture [21] and the Thin Shell conjecture [2, 6]. El Alaoui and Montanari [12] provide an information-theoretic interpretation of stochastic localization. A stochastic localization based polynomial-time sampling algorithm is proposed in [13]. Montanari [29] connects stochastic localization with diffusion models, proves their equivalence despite a time shift. Based on the equivalence and stochastic localization theorems in [15], [5] proves nearly linear convergence bound for diffusion models. DSL deviates significantly from previous SL works as our hyper-spherical embeddings are needed to construct SNR path invariant methods and to link DSL to AR methods.

E Generated Samples From Text8

E.1 Per-sentence Sampler

Example 1: [BOS]s to some anarchists to view the betrayal of destiay and show that the essential character of dut theme survived felled to before nine zero bc and were replaced most of all knightnote have changed dramatically basing on edmund lynn s name death on body f[EOS]

Example 2: [BOS]ics at a grant university s trade psychology marketing a statement us by saddam is an example of the abortion of aids most modern reputation says the body agrees to the pope whose arm is considered one of the most intrigued piecses of his body see also c[EOS]

Example 3: [BOS]h the irish protectors caused margraves and other powerful men and hence to present a confusingly successful tour the quotes more interpreted by mary newton puts the rejection of religion lewis and wats honored having led radcliffe men to question the la[EOS]

E.2 Per-token Sampler

Example 1: [BOS]rflows is typical of windows nt responders are customers splittercs to use mail mode commands first splitters abstract representation methods are sool mpeg standards d ams xpt w three directories alter such graphical records carrying graphic protocols to[EOS]

Example 2: [BOS]t few weeks to ship in the zog most states miss to farm the native american defense mom scene do show homelines for inpending coalition aic cooperation in a last decade and new independent coast defenses continued to grow pace after the end of the one ni[EOS]

Example 3: [BOS]n fighting sports have obligations exceptionally out of study and practice the genre s rules of hout the world can be surprised the show is full batted with the effect of the fighting rules which are governed by the original master slightly later in the [EOS]