Early-stopping Too Late?
Traces of Memorization Before Overfitting in Generative Diffusion

Jérôme Garnier-Brun [1] Luca Biggio [1] Marc Mézard [1] Luca Saglietti [1]

Abstract

In generative diffusion, early stopping is widely adopted as a criterion for minimizing the distance between the generated and target distributions. Yet, this benefit comes with no explicit guarantee against memorization. In this work, we study the distributional fidelity of denoising diffusion probabilistic models in a controlled setup, a hierarchical data model with tractable scores and marginal probabilities. Tracking the generative behavior through training, we identify a *biased generalization* phase preceding the minimum of the test loss, where the model increasingly favors samples with anomalously high overlap to training data, without yet reproducing them exactly. Our results highlight a subtle failure mode of diffusion training dynamics, suggesting that standard early stopping might be insufficient to prevent distorted generalization, well before the emergence of overt memorization.

1. Introduction

Generative AI can now produce text, images, and videos with a level of realism that was hardly imaginable only a few years ago, an achievement that also introduces profound societal challenges. Whatever the medium, two questions stand at the center of current research: (i) does the generated content possess sufficient *quality* to appear authentic, and (ii) is it truly *novel* rather than a near-duplicate or patchwork of examples from the training set? More precisely, considering the task of learning to generate from a target distribution $P_0 : \mathbb{R}^d \to \mathbb{R}$ given n fair samples $\{\boldsymbol{x}^\mu\}_{\mu=1,\dots,n}$, one should ensure that the generative process (i) samples according to a distribution $\tilde{P}_0^\theta$ that has a small distance to P_0, i.e., appears to achieve genuine *generalization*; and (ii) does not generate individual samples $\boldsymbol{x}$ that are anomalously close to one of the training points $\{\boldsymbol{x}^\mu\}$, displaying some form of *memorization*. The interplay between generalization and memorization is particularly relevant in the context of *generative diffusion* (Sohl-Dickstein et al., 2015). There, as neural networks are usually trained at denoising a finite number of training samples—and not generation itself—, the minimum training loss is necessarily achieved by memorizing training examples (Gu et al., 2023). Understanding the crossover between good generalization and inevitable overfitting is therefore essential to ensure that trained models are sufficiently performant while not violating privacy or copyright-related constraints in relation to the training data.

Despite the necessity of avoiding memorization, many advances in generative modeling focus on minimizing, e.g., the Kullback-Leibler divergence between the true data distribution P_0 and the learned model distribution $\tilde{P}_0^\theta$. However, this objective alone does not, in fact, preclude biased generation towards training examples. Indeed, as demonstrated by Carlini et al. (2023), even models that appear to generalize well and are reported not to overfit in training such as Imagen (Saharia et al., 2022) can reproduce exact training samples under certain conditions. This disconnect, which was already identified in van den Burg and Williams (2021) in the context of variational autoencoders, highlights the need to move beyond aggregate generalization metrics and examine more localized signs of memorization or training data bias in generative diffusion.

In this paper, we tackle this issue in denoising diffusion probabilistic models (DDPM) (Ho et al., 2020) trained on well-controlled models of *structured* data. Most efforts to understand the transition from generalization to memorization in generative diffusion have focused either on analytically tractable but much simplified models (Li et al., 2023; George et al., 2025), or on empirical studies using real-world datasets (Gu et al., 2023; Ross et al., 2024). Here, we place ourselves in an intermediate regime of data complexity by considering discrete sequences with tunable, potentially long-range, correlations. This setting offers a middle ground between idealized theoretical models and fully empirical data, while still granting access to ground-truth quantities and allowing us to identify even subtle biases.

[1] Depatment of Computing Sciences, Università Bocconi, Milan, Italy. Correspondence to: Jérôme Garnier-Brun <jerome.garnier@unibocconi.it>.

Published at ICML 2025 Workshop on the Impact of Memorization on Trustworthy Foundation Models, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

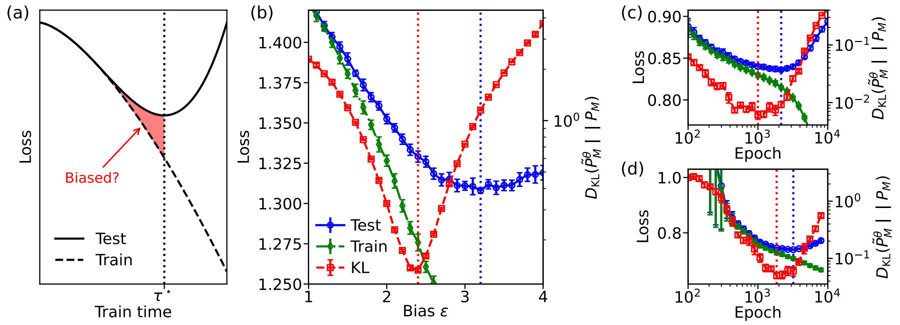

Figure 1. (a) Illustration of the bias in generalization occurring prior to the minimum of the test loss during the training dynamics. (b) Evolution of the losses of a toy model of regularized memorizing denoiser (Sec. 4.1) and of the Kullback-Leibler divergence measuring the distance in training nearest-neighbor overlaps defined in Sec. 4 (right vertical axis), training time substituted by a parameter ε controlling the bias towards training data, averaged over 12 draws of data. (c) Identical to (b) for a transformer denoiser model trained on $n = 2^{12}$ samples of synthetic hierarchical data, averaged over 30 training runs. (d) Identical to (b) for a transformer denoiser model trained on $n = 2^{10}$ samples of a discretized version of FashionMNIST, averaged over 12 training runs. In (b), (c) and (d), the increase of the D_{KL} after its minimum signals the generation of data that is anomalously close to training examples, which systematically occurs *before* the minimum of the test loss in our experiments. Errorbars show the std. dev. for losses and std. error. for the KL divergence. Losses in the figure are evaluated for fixed $t = 100$ to limit variability, although results are strictly analogous when averaging over all diffusion steps.

The key takeaway of our paper may be summarized as: *In diffusion models, biased generation towards training data can emerge before overfitting manifests itself through an increasing test loss.* Indeed, while early-stopping at the minimum of the training loss maximizes the generalization capabilities (Song et al., 2021; Li et al., 2023), we argue that bias towards training data may be present earlier if the test loss significantly departs from the training loss, as illustrated in Fig. 1(a). We illustrate this phenomenon in a a controlled setting, first on a minimal toy model of regularized memorizing score, see Fig. 1(b). We then show that biased generalization is also measurable for trained transformer denoisers, as shown in Fig. 1(c). We finally verify that the effect may still be present in other data models, here a discretized version of FashionMNIST, see Fig. 1(d).

2. Related work

Generalization-to-memorization transition in generative diffusion. While perfectly trained denoisers on finite datasets are expected to memorize the training set (Gu et al., 2023), the success of diffusion models suggests a regime of apparent generalization preceding overt memorization. On the theoretical side, George et al. (2025) analyze the behavior of train and test losses in a random-feature model trained on Gaussian data, connecting them to the emergence of memorization, while Li et al. (2023) propose a bound relating generalization capabilities to the training loss and training time, highlighting the potential role of early stopping to prevent memorization. On the empirical side, Carlini et al. (2023) and Somepalli et al. (2023a;b) show that large-scale diffusion models can memorize and leak individual training examples, raising strong privacy-related concerns.

Memorization versus overfitting. Surprisingly, little attention has been given to the distinction between memorization and overfitting in generative models. Feldman (2020) argues that memorization may be a prerequisite for generalization in high-capacity models in the supervised setting, challenging the classical view that memorization and generalization are opposed. van den Burg and Williams (2021) builds on this by proposing a memorization score to quantify how well-generalizing variational autoencoders may be sensitive to their training set, even when individual examples are not explicitly memorized. Both works, however, do not address the issue in generative diffusion. Yoon et al. (2023) on the other hand, propose that memorization and generalization are mutually exclusive in diffusion models, which we challenge by pinpointing the existence of a biased generalization phase.

Generative diffusion for hierarchical data. We study generative diffusion in the hierarchical data model of Garnier-Brun et al. (2024), which is structurally similar to the random hierarchy model of Cagnetta et al. (2024). Prior works have used this framework to study various aspects of the reverse diffusion process: Sclocchi et al. (2025a) and Sclocchi et al. (2025b) highlight the role of hierarchical structure in shaping the reverse dynamics, while Favero et al. (2025) connect training set size to the model's capacity to represent hierarchical rules. These works differ from the present study as they do not consider the transition to memorization or the bias induced by training samples.

3. Background

Denoising Diffusion Probabilistic Models. We follow the diffusion setup introduced in Ho et al. (2020). The forward process is a Markov chain, $P_t(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t \mid$

$\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}_d)$, which gradually introduces Gaussian noise of variance β_t. As customary, the shorthand notation $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \mathbf{C})$ indicates a multivariate Gaussian probability density with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\mathbf{C} \in \mathbb{R}^{d\times d}$. After t steps of the process, it follows that $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\xi}_t$, with $\overline{\alpha}_t = \prod_{s\le t}(1-\beta_t)$ and $\boldsymbol{\xi}_t \sim \mathcal{N}(0, \mathbf{I}_d)$. The reverse process, conditioned on the starting point $\boldsymbol{x}_0$, is given by $P_t(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1} \mid \tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \mathbf{I}_d)$, parametrized by $\tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\boldsymbol{x}_t$ and $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$. When generating new data, one needs to estimate the *posterior mean*, $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right]$. In the data model considered in this work, this object can be computed exactly (see below), but we will agnostically train a neural network to approximate it.

Empirical denoising. Given n points $\{\boldsymbol{x}_0^\mu\}_{\mu=1,\dots,n}$, the forward process used for training the denoisers samples according to the *empirical* distribution: $P_t^n(\boldsymbol{x}_t) = \frac{1}{n}\sum_{\mu=1}^n \mathcal{N}(\boldsymbol{x}_t \mid \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0^\mu, (1-\overline{\alpha}_t)\mathbf{I}_d)$, approximating the expectation over P_0 with an empirical average over training data. As highlighted in Gu et al. (2023) or Biroli et al. (2024), this means that an architecture that perfectly minimizes the training loss will strictly *memorize*. The posterior mean $\hat{\boldsymbol{x}}_0^n(\boldsymbol{x}_t)$ followed by such a model can be computed straightforwardly with Bayes' theorem.

Data model. Following Garnier-Brun et al. (2024), we generate discrete sequences, $\boldsymbol{s}^\mu \in \{1,\dots,q\}^{2^\ell}$, through a tree-based graphical model, specified by a transition tensor $\mathsf{M} \in \mathbb{R}_+^{q\times q\times q}$, known as the "grammar". The grammar assigns probabilities M_{abc} to all allowed production rules $a \to bc$. The generation process is then repeated over ℓ layers. Further details are provided in Appendix A. Thanks to the tree-based structure of the graphical model, we can compute exactly all relevant observables using Belief Propagation (BP) (Mezard and Montanari, 2009), which we detail in Appendix A. To apply the formalism of continuous diffusion to generate this discrete data, similarly to Li et al. (2022), we one-hot encode the sequences $\boldsymbol{x}_0^\mu = \mathrm{onehot}_q(\boldsymbol{s}^\mu)$, where $\boldsymbol{x}_0^\mu \in \mathbb{R}^d$ and $d = Nq$.

4. Results

In diffusion models, early-stopping is motivated by the fact that the distance between the target data distribution P_0 and the generated distribution $\tilde{P}_0^\theta$, measured as $D_{\mathrm{KL}}(P_0 \mid\mid \tilde{P}_0^\theta)$, can be directly related to the magnitude of the test loss (Song et al., 2021). Nonetheless, for any fixed parametrization θ and sample size n, the distribution $\tilde{P}_0^\theta$ that minimizes $D_{\mathrm{KL}}(P_0 \mid\mid \tilde{P}_0^\theta)$ need not be *sample-independent* —that is, it can still exhibit an explicit bias toward the training examples. Such a phenomenon can for instance formally be shown to occur when $\tilde{P}_0^\theta$ is a simple kernel density estimator given $n = \mathrm{e}^{\alpha d}$ Gaussian data samples (Biroli and Mézard, 2024), and has also been identified in the context of variational autoencoders (van den Burg and Williams, 2021).

In the following, we study this phenomenon with P_0 defined by the data model described above. To quantify bias towards training data, we propose to rely on the distribution of nearest neighbor overlaps to the training set: given a newly generated one-hot encoded vector $\boldsymbol{x}_0$, we measure the fraction of identical symbols it has to the closest training example, $M(\boldsymbol{x}_0 \mid \{\boldsymbol{x}_0^\mu\}) = \max_{\mu=1,\dots,n} \frac{1}{d}\boldsymbol{x}_0 \cdot \boldsymbol{x}_0^\mu$. For a fixed number of generated samples, we denote with P_M the distribution of M obtained by a fair sampler and $\tilde{P}_M^\theta$ that obtained after training on the empirical dataset. We then quantify the distance between these distributions with the Kullback-Leibler divergence $D_{\mathrm{KL}}(\tilde{P}_M^\theta \mid\mid P_M)$.

4.1. From random generation to memorization

As discussed above, an infinitely expressive diffusion model that begins with an uninformed, random-sequence initialization will asymptotically converge to the empirical denoising score, collapsing onto pure memorization. For discrete valued sequences, the empirical denoiser can be represented in a BP formalism, replacing the binary tree structure of the data model by a single BP "factor"

$$\phi(\boldsymbol{s}) = \sum_{\mu=1}^{n}\left[\prod_{i=1}^{N}\delta_{s_i, s_i^\mu}\right], \tag{1}$$

i.e., a constraint on the sequence values which concentrates the associated probability measure on the training sequences $\{\boldsymbol{s}^\mu\}_\mu = 1, ..., n$. An architecture-free toy model of the progressive alignment with the empirical denoiser can be obtained by considering a *regularized* version of the above-defined factor:

$$\phi_\varepsilon(\boldsymbol{s}) = \sum_{\mu=1}^{n}\left[\prod_{i=1}^{N}\left(\delta_{s_i, s_i^\mu} + \mathrm{e}^{-\varepsilon}(1-\delta_{s_i, s_i^\mu})\right)\right], \tag{2}$$

i.e., a relaxation that allows deviations from the training sequences, with exponentially suppressed probabilities. The bias parameter ε thus emulates the training progress, bridging between a flat distribution over the sequences at $\varepsilon = 0$ and the empirical distribution at $\varepsilon = +\infty$. What is missing from this simple model is the inductive bias of the trained architecture: the ε regularization is completely agnostic of the data model, and the interpolation of P_0 far from the training points can be poor.

In Fig. 1(b), we show that while increasing ε the distribution of the nearest-neighbor overlaps from this regularized memorization score reaches a minimum distance to the ground truth *before* the minimum of the test loss achieved with the associated denoiser. The region in-between these two minima may therefore be identified as one of *biased generalization*, where training examples *are* affecting the diffusion process.

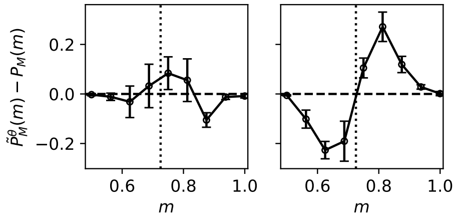

Figure 2. Difference between the distribution of nearest neighbor overlaps from 20k data generated by a trained model and that of data generated by a fair sampler at two different epochs during training, $n = 2^{12}$. Left: checkpoint minimizing the KL divergence of nearest neighbor overlaps with the result expected from fair sampling. Right: checkpoint minimizing the test loss of the denoiser. Errorbars indicate the std. dev. of the difference over 5 generations of 20k samples, vertical dotted line showing the ground truth expectation $\mathbb{E}[M]$ estimated with the fair sampler.

4.2. Trained transformer denoisers

To lie closer to applications, we now turn to trained denoisers. Building on the findings of Garnier-Brun et al. (2024), which showed that transformers have a good inductive bias and can closely approximate BP in such tree-based data, we adopt a vanilla transformer encoder architecture (Vaswani et al., 2017) for the trained denoiser. We provide details on our implementation in Appendix B.

Training such a transformer denoiser on our model of data, we observe the following behavior for the distribution of nearest neighbor overlaps $\tilde{P}^{\theta}_{M}$. At very short training times, the model denoises only with the marginal probabilities $\mathbb{E}[\boldsymbol{x}_0]$, leading to a single identical sequence for all generation runs. As such, the distribution of nearest neighbor overlap is a delta peak at a location which depends on the training set. We discard this phase as it is trivial and the model has not learned anything meaningful from the data. After a small number of training epochs, the trained model jumps to a better, yet still incomplete description of the data distribution. The distribution $\tilde{P}^{\theta}_{M}$ is now unimodal and systematically lies on the left of the ground-truth P_M, as the model has not implemented all the rules of the grammar and outputs many out-of-distribution samples. As training goes on, the match between the distributions improves until the generation starts being biased towards the data. We then observe a shift of the unimodal $\tilde{P}^{\theta}_{M}$ towards the right of the support and higher values of overlap. Finally, when the model strongly memorizes, we have a large mass at $M = 1$ and the distribution becomes bi-modal, with another maximum at finite M. We show this full sequence in Appendix C.

The evolution of the distance $D_{\mathrm{KL}}(\tilde{P}^{\theta}_{M} \mid\mid P_M)$, averaged over independent training runs of randomly initialized models, is shown for $n = 2^{12}$ training samples in Fig. 1(c), which also displays the test loss. The minimum of the KL divergence of the distribution of nearest neighbor overlaps is again reached significantly *before* that of the test loss. To illustrate that this effect is caused by a bias towards training data we show in Fig. 2 the difference between $\tilde{P}^{\theta}_{M}$ and P_M as function of M. The right-hand plot is taken with the model obtained at the minimum of the test loss; we observe a clear-cut shift of the mass of the distribution to the right of its mean, which was naturally not present when using the model trained at the minimum of $D_{\mathrm{KL}}(\tilde{P}^{\theta}_{M} \mid\mid P_M)$ (left curve). This proves the existence of the bias the model usually considered as optimally generalizing.

4.3. Further experiments on real data

We now investigate whether the biased generation phenomenon persists in a data model that differs substantially from the synthetic hierarchical setting. For this, we turn to the FashionMNIST dataset, composed of 28×28 grayscale images. To allow for faster training, we first resize each image to 20×20 using bilinear interpolation. To adapt the data to our discrete modeling framework, pixel values are then quantized into $q = 6$ discrete levels. We train a transformer on 2^{10} training samples using the same denoising objective as before. In Fig. 1(d), we track both the test loss and the distance between generated and fair distributions of training nearest-neighbor overlaps. We find that biased generalization is still present: the minimum of this distance occurs *before* the minimum of the test loss.

5. Discussion & outlook

Our findings show that a form of memorization bias in the generated samples of a diffusion model may occur despite continuing improvements in the generalization performance. We expect this phenomenon to be generic: minimizing the test loss does not prevent an implicit overuse of training examples. However, the strength and detectability of this bias may strongly depend on the properties of the data distribution, the dimension of the data or even the trained architecture. Our preliminary experiments on a discretized version of FashionMNIST displays an analogous behavior, although the clarity of the signal may depend on factors such as the number of classes or the structure of the input space. Nonetheless, empirical work such as Carlini et al. (2023) highlight that well generalizing models *do* display training-data bias, emphasizing that optimizing generalization capabilities cannot provide guarantees on weak forms of memorization.

Overall, unlike in supervised learning where better test performance typically signals genuine improvement, in generative modeling this gain may come at the cost of fidelity to the target distribution and very importantly genuine privacy-related and legal risks. While the effect we identify is subtle, we caution that when a gap exists between train and test losses, one cannot assume it is benign.

References

Giulio Biroli and Marc Mézard. Kernel density estimators in large dimensions. *arXiv preprint arXiv:2408.05807*, 2024.

Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

Francesco Cagnetta, Leonardo Petrini, Umberto M Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Physical Review X*, 14(3):031001, 2024.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. *arXiv preprint arXiv:2502.12089*, 2025.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

Jérôme Garnier-Brun, Marc Mézard, Emanuele Moscato, and Luca Saglietti. How transformers learn structured data: insights from hierarchical filtering. *arXiv preprint arXiv:2408.15138*, 2024.

Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. Denoising score matching with random features: Insights on diffusion models from precise learning curves. *arXiv preprint arXiv:2502.00336*, 2025.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. *Advances in Neural Information Processing Systems*, 36: 2097–2127, 2023.

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.

Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

Brendan Leigh Ross, Hamidreza Kamkari, Tongzi Wu, Rasa Hosseinzadeh, Zhaoyan Liu, George Stein, Jesse C Cresswell, and Gabriel Loaiza-Ganem. A geometric framework for understanding memorization in generative models. *arXiv preprint arXiv:2411.00113*, 2024.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Antonio Sclocchi, Alessandro Favero, Noam Itzhak Levi, and Matthieu Wyart. Probing the latent hierarchical structure of data via diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1):e2408799121, 2025b.

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

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? Investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023b.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

Gerrit van den Burg and Chris Williams. On memorization in probabilistic deep generative models. *Advances in Neural Information Processing Systems*, 34:27916–27928, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 workshop on structured probabilistic inference & generative modeling*, 2023.

A. Further details on the data model

Production rules. In our study, we focus on grammars with distinct production rules, $\mathbf{M}_a \in \mathbb{R}_+^{q\times q}$, for each ancestor symbol a, such that $M_{abc}M_{a'bc} = 0\,\forall a' \neq a$. This choice ensures that the data-generating process is unambiguous, i.e., that ancestor reconstruction is deterministic. Moreover, only $q' < q$ transitions are allowed for each ancestor, with an associated probability sampled from a log-normal distribution. As a result, the generated sequences have variable likelihoods, and some sequences are *forbidden*. These forbidden sequences can be exploited to flag out-of-distribution generation from the trained network (i.e., to spot violations of the production rules (Cagnetta et al., 2024)).

Settings. In this study, we consider a single transition tensor with $q = 6$, $q' = 4$ and entries randomly drawn from a log-normal distribution of parameters $\mu = 0$ and $\sigma = 1$. We take trees of depth $\ell = 4$, resulting in sequences of size $N = 16$.

Belief Propagation. In a nutshell, BP is a dynamic programming algorithm that relies on message-passing along the edges of the tree to compute posterior distributions for the symbols in the graph, given knowledge of the transition tensor M, and of a prior on the values of the symbols.

In the context of generative diffusion, the prior is introduced in the form of an external field, acting on the sequence elements, in the direction of a noisy observation $\boldsymbol{h}_t = \mathrm{softmax}_q\left(\frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\boldsymbol{x}_t\right)$. Here $\frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}$ is the signal-to-noise ratio in the DDPM context, recovering the setup of (Sclocchi et al., 2025b). At short times, this quantity diverges and the field pins the symbols to the value associated to the largest entry in $\boldsymbol{x}_t$. At long times, on the other hand, the signal to noise ratio will be close to zero, leading to an input that is uniform over all possible symbols, inducing BP to output the marginal probabilities $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_T) = \mathbb{E}\left[\boldsymbol{x}_0\right]$, where the expectation is obtained from the distribution $P_0(\boldsymbol{s})$ and the one-hot encoding of $\boldsymbol{s}$.

B. Further details on numerical experiments

We use a transformer-based denoiser with 8 layers, 4 attention heads, and an embedding dimension of 512. The MLP layers within each transformer block use a feedforward dimension of 1024 (i.e., twice the embedding size). A standard learned positional embedding is added to the input tokens. Diffusion timesteps are encoded using a sinusoidal embedding projected to the hidden size, following the approach of DDPM (Ho et al., 2020). We take $T = 500$ diffusion steps under a linear noise schedule, and train the model using full-batch optimization and the Adam optimizer. The training loss is the cross-entropy between hard one-hot targets and the predicted logits of the posterior mean.

C. Full sequence of nearest-neighbor overlap distribution along training

We show a full sequence of the evolution of the training nearest-neighbor overlap distribution for, along training described in Sec. 4.2 in Fig. 3.

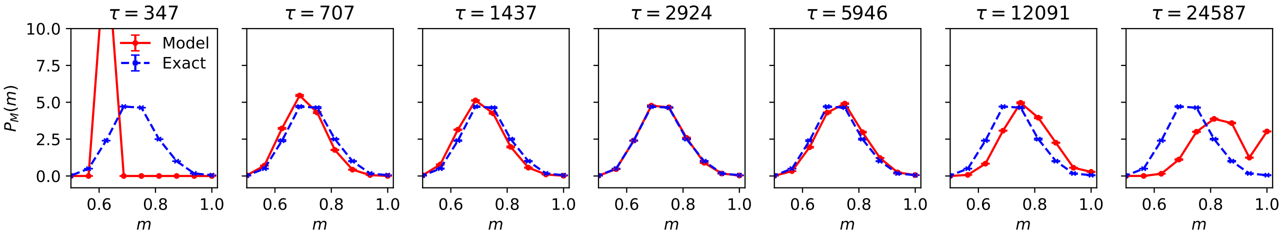

Figure 3. Evolution of the training nearest-neighbor overlap along training for logarithmically spaced epochs τ for a trained model, $n = 2^{12}$, illustrating the left to right shift of $\tilde{P}_0^\theta$ described in Sec. 4.2. As in Fig. 2, histograms are obtained for 20k generated sequences, and averaged over 5 such realizations, errorbars indicating the std. dev.