Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models

Alessandro Favero [*1] **Antonio Sclocchi** [*2] **Matthieu Wyart** [1 3]

Abstract

Diffusion models have become a cornerstone of generative AI, yet the mechanisms underlying their generalization remain poorly understood. If these models were perfectly minimizing their training loss, they would just generate data belonging to their training set, as empirically found in the overparameterized regime. We revisit this view by showing that, in overparameterized diffusion models, generalization in natural data domains is progressively achieved during training *before* the onset of memorization. Our results, ranging from image to language diffusion models, systematically support the empirical law that memorization time is proportional to the dataset size. Generalization vs. memorization is then best understood as a competition between time scales. We show that this phenomenology is recovered in diffusion models learning a simple probabilistic context-free grammar, where generalization corresponds to the hierarchical acquisition of deeper grammar rules as training time grows, and the generalization cost of early stopping can be characterized. We summarize these results in a phase diagram. Overall, our results support that a principled early-stopping criterion – scaling with dataset size – can effectively optimize generalization while avoiding memorization.

1. Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have recently emerged as a transformative paradigm in AI, enabling the synthesis of high-quality data across a wide range of modalities – images, videos, text, and complex 3D structures such as molecules and proteins. At the heart of this process is the estimation of a *score function* (Song & Ermon, 2019; Song et al., 2020): a noise-dependent vector field that guides denoising by pointing in the direction of increasing likelihood. Since the score is learned from the empirical training distribution, minimizing the training loss leads the model to reproduce the training data itself, i.e., *memorization* (Carlini et al., 2023; Somepalli et al., 2022). This phenomenon is observed in practical settings and raises significant privacy and copyright concerns, as models trained on sensitive or proprietary data may inadvertently regenerate such content, exposing private information or violating intellectual property rights (Wu et al., 2022; Matsumoto et al., 2023; Hu & Pang, 2023). In contrast, *generalization* corresponds to the model producing novel samples that are consistent with, but not identical to, the training data, thereby approximating the broader target distribution.

Despite the empirical success of diffusion models, the mechanisms underlying their ability to generalize remain poorly understood. A prevailing view – rooted in classical learning theory – is that generalization depends on *underparameterization* (Yoon et al., 2023; Zhang et al., 2023; Kadkhodaie et al., 2023): only models that lack the capacity to memorize their training data are expected to generalize. In this work, we go beyond this view by demonstrating that even heavily overparameterized diffusion models exhibit generalization during training *before* they start memorizing the training data. We systematically investigate this phenomenon, showing that generalization and memorization are not mutually exclusive but unfold as distinct temporal phases of training. We empirically demonstrate the transition from generalization to memorization during training in a range of overparameterized diffusion models on images and text data. We measure memorization and generalization metrics and systematically vary the training set size, showing that generalization improves gradually, before the onset of memorization. We find the empirical law that the onset of memorization requires a number of training steps that is proportional to the training set size. We interpret these findings by studying a diffusion model trained to learn a simple formal grammar, where the number of training steps or samples required to generalize is known to be polynomial in the sequence length (Favero et al., 2025). We show that for moderate training set sizes, the diffusion model only

[*]Equal contribution [1]Institute of Physics, EPFL [2]Gatsby Unit, UCL [3]Department of Physics & Astronomy, Johns Hopkins University. Correspondence to: Alessandro Favero <alessandro.favero@epfl.ch>.

Published at ICML 2025 Workshop on the Impact of Memorization on Trustworthy Foundation Models, Vancouver, Canada. PMLR 267, 2025.

learns the lowest levels of the hierarchical grammar rules – corresponding to partial generalization – before starting to memorize. For larger training set sizes, the onset of memorization appears after perfect total generalization is achieved. These results lead to a phase diagram for memorization and generalization as a function of sample complexity and time.

On the theoretical level, these findings call for a revision of the view of generalization in diffusion models as being solely determined by model capacity, showing that generalization arises *dynamically during training* in overparameterized diffusion models. On the practical level, our results suggest that early stopping and dataset-size-aware training protocols may be optimal strategies for preserving generalization and avoiding memorization as the size of diffusion models is scaled up. In fact, meeting privacy and copyright requirements with principled procedures is of utmost importance for the deployment of generative AI, in contrast to heuristic procedures that lack quantitative grounding (Dockhorn et al., 2022; Vyas et al., 2023; Chen et al., 2024).

2. Diffusion Models and the Score Function

Diffusion models are generative models that sample from a data distribution $q(x_0)$ by reversing a noise addition process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2020). The *forward process* generates a sequence of increasingly noised data $\{x_t\}_{1\leq t\leq T}$, with distribution $q(x_1,\dots,x_T|x_0)=\prod_{t=1}^{T} q(x_t|x_{t-1})$, where t indicate the time step in $[0,\dots,T]$. At the final time T, x_T corresponds to pure noise. The *backward process* reverts the forward one by gradually removing noise and is obtained by learning the backward transition kernels $p_\theta(x_{t-1}|x_t)$ using a neural network. Learning these kernels is equivalent to learning the *score function*, which is proportional to the conditional expectation $\mathbb{E}_{q(x_0|x_t)}[x_0]$. To learn the score, training is performed by minimizing a bound on the negative log likelihood of the data $\mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)]$. The loss requires an integral over the target distribution $q(x_0)$. This is estimated with Monte Carlo from P examples $\{x_0^{(i)}\}_{i=1,P}$, associated with the empirical distribution $\hat{q}(x_0)=P^{-1}\sum_{i=1}^{P}\delta(x_0-x_0^{(i)})$. Thus, perfectly minimizing the empirical loss corresponds to learning the empirical score, which generates $\hat{q}(x_0)$. As a result, diffusion models would only generate data of the training set, corresponding to *memorization*. Their generalization abilities, therefore, derive from not perfectly minimizing the empirical loss.

3. Numerical Experiments

In this section, we analyze generalization and memorization in large vision and text diffusion models.

Vision diffusion models We assess the generalization and memorization behaviors of vision diffusion models by considering Improved Denoising Diffusion Probabilistic Models (iDDPMs) (Nichol & Dhariwal, 2021) with a U-Net architecture (Ronneberger et al., 2015; Salimans et al., 2017), including attention blocks (Vaswani et al., 2017). Each model, comprising approximately 0.5B parameters, is trained on four distinct subsets of the CIFAR-10 dataset (Krishnan et al., 2017), with training set sizes $P\in\{2048,\ 4096,\ 8192,\ 16384\}$. The models are trained for a total of 262,144 training steps. We track performance using the losses on the train set and a validation set of 1,024 images. At regular checkpoints, we generate 32,768 images using each model, and evaluate memorization by calculating the fraction of generated images that are near-exact replicas of training samples. Specifically, following Yoon et al. (2023), for a generated image x, we identify the two closest x' and x'' in Euclidean distance from the training set, and classify x as a copy if $\|x-x'\|_2/\|x-x''\|_2<1/3$.

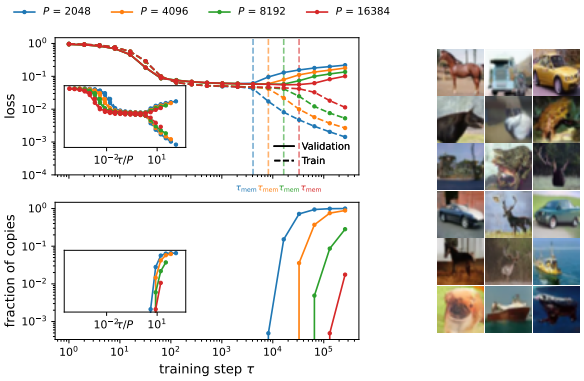


Figure 1. **Memorization in vision models.** *Left:* Train, validation loss, and fraction of copied images as a function of training steps τ for iDDPM models trained on CIFAR10 with training set sizes P. *Right:* Samples generated with early stopping at $\tau_{\rm mem}$ with a model trained on 16,384 images.

Results and analysis Figure 1 (left panel) presents the results of this experiment. Our key findings are as follows:

1. **Generalization before memorization:** Initially, both train and validation loss decrease, indicating that the model is generalizing, i.e., approaching the population score. Yet, at some time $\tau_{\rm mem}$, the two losses bifurcate, signaling the onset of memorization. After this point, the number of copies among generated images steadily increases. By the end of training, all models exhibit some degree of memorization, with copy rates ranging from 1% for the largest train set to 100% for the smaller ones.
2. **Memorization is delayed by larger training sets:** The onset of memorization $\tau_{\rm mem}$ scales approximatively linearly with the training set size P (insets of Figure 1).

These observations suggest that early stopping can effectively prevent the model from entering the memorization phase. As a concrete example, the right panel of Figure 1

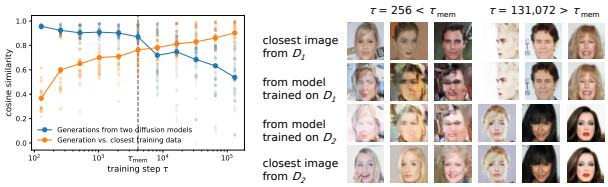


Figure 2. **Progressive generalization in vision models.** Cos. sim. between images generated by two diffusion models trained on disjoint subsets of CelebA of size $P = 2,048$, as a function of τ.

displays images generated by a diffusion model trained on $16,384$ images, with early stopping applied. The quality and diversity of these images are quantified using the Fréchet Inception Distance (FID), calculated using Inception v3. The model achieves an FID score of 5.4, indicating – despite being strongly overparameterized – robust generalization.

Progressive generalization We extend our analysis by conducting a second experiment inspired by Kadkhodaie et al. (2023). Specifically, we train two models on two non-overlapping subsets $\mathcal{D}_1$ and $\mathcal{D}_2$ of $2{,}048$ images of CelebA (Liu et al., 2018), a dataset with faces of celebrities, each using an iDDPM. Our setup goes beyond prior work by dynamically tracking the evolution of the generated images throughout training, rather than statically only at convergence. This approach provides a detailed view of how models first approach the population score and then diverge after entering the memorization phase. We generate samples from both models at multiple checkpoints during training, initializing the generations from the same Gaussian random noise and fixing the stochastic part of the backward trajectories. Remarkably, initially, the images generated by the two models are nearly identical, reflecting that the two models are learning the same score function, even though they are trained on disjoint data subsets. However, at some time τ_{mem}, the models begin to diverge. This divergence coincides with the onset of memorization, where the models start generating images increasingly similar to the ones contained in their respective training sets. We quantitatively assess this phenomenon using cosine similarity between whitened images generated by the two models and their nearest training images. As shown in Figure 2, **before memorization** ($\tau < \tau_{\text{mem}}$), the two models generate nearly identical images, indicating that they are dynamically learning the same underlying distribution. **During memorization** ($\tau > \tau_{\text{mem}}$), the similarity between the models' generated images decreases monotonically, while the similarity between each model's generated images and their training set increases. This reflects the transition from generalization to memorization.

Our findings extend those of Kadkhodaie et al. by revealing that the transition from generalization to memorization is not only a matter of model capacity and final convergence but is dynamically observable throughout training. In practice, this further supports the view that early stopping can prevent the memorization phase and maintain generalization.

Language diffusion models We extend our analysis of generalization and memorization to language data, using MD4, a masked diffusion model designed for text (Shi et al., 2024). Our experiments are conducted on the text8 dataset, a standard benchmark for language modeling based on Wikipedia, with character-level tokenization. To the best of our knowledge, this is the first demonstration of memorization in the language diffusion setting. As shown in App. E, MD4 initially generalizes, improving the log-likelihood on the validation corpus. Yet, at τ_{mem} the model begins to produce exact or near-exact copies of training text. Notably, τ_{mem} scales linearly with P.

Summary We have shown empirically that as they train, diffusion models generate higher and higher quality data, which are novel. This is true up to an early stopping time τ_{mem} where memorization starts, which we found to follow a remarkably universal empirical law: $\tau_{\text{mem}} \sim P$. Having more data thus allows training longer. We will now study a controlled model of synthetic data that captures this law. Most importantly, it will allow us to quantify in detail the inaccuracy of generations of diffusion models with limited training, responsible for the inconsistent images in Figure 2.

4. Toy Setting

We consider models trained to generate strings respecting the rules of a simple grammar, which gives a theoretical framework to interpret the generalization-memorization dynamics. *Probabilistic Context-Free Grammars* (PCFGs) consist of a vocabulary of latent (*nonterminal*) symbols and a vocabulary of visible (*terminal*) symbols, together with probabilistic *production rules* establishing how one latent symbol generates tuples of latent or visible symbols. The **Random Hierarchy Model** (RHM) (Cagnetta et al., 2024) is a simple PCFG introduced as a theoretical toy model describing hierarchy and compositionality in data. With respect to generic PCFGs, it is built with some simplifying assumptions. Symbols are organized in a regular-tree topology of depth L and branching factor s. Symbols are taken from a vocabulary of size v. The production rules transform one symbol in a node at level $\ell + 1$ into a string of s symbols in its children nodes at level ℓ. For each non-terminal symbol, there are m rules with equal probability, which are *unambiguous*. Rules are sampled randomly without replacement. The fixed tree topology ensures that visible data at the leaves are strings of fixed length $d = s^L$, corresponding to the data dimension d. In analogy with language modeling, we call visible symbols *tokens*.

If production rules are known, thanks to the tree structure, the score function can be computed exactly using the Belief Propagation (BP) algorithm (Mezard & Montanari, 2009). Favero et al. (2025) studied the sample complexity for diffusion models based on deep neural networks trained on finite RHM data. The sample complexity to learn to gen-

erate valid data depends on the parameters of the model as $P^* \sim v\, m^{L+1}$, which is polynomial in the dimension, i.e., $P^* \sim vmd^{\log m/\log s}$. For $P < P^*$, there are regimes of partial generalization where the generated data are consistent with the rules up to layer ℓ. The sample complexity to learn the rules at layer ℓ scales as $P^*_\ell \sim v\, m^{\ell+1}$. When $P > P^*_\ell$, the number of training steps τ^*_ℓ required to learn the rules at layer ℓ is proportional to P^*_ℓ. Complete generalization is therefore achieved with $\tau^* \propto P^* = P^*_L$.

We generate P training strings from an RHM. We train a discrete diffusion model (Austin et al., 2021) with a convolutional net with $2L$ layers and 8,192 channels. Figure 3 shows the training evolution of a model for $v = 16$, $m = 4$, $L = 3$, $s = 2$ ($P^* \approx 4{,}096$). Varying P, the validation and training losses start decreasing at the same time and follow the same behavior until separating later in training, at a time depending on P. Comparing these losses with the fraction of copies between the generated data and the training ones, we observe that the increase of the validation loss corresponds to the onset of memorization. As observed for real data, we find empirically that memorization requires a number of training steps $\tau_{\rm mem} \propto P$. We observe that for $P < 4{,}096$, the fraction of errors, i.e., how many generated data are not compatible with the rules, decreases only in correspondence with memorization: generated data are valid according to the grammar rules, but are copies of the training set. For $P > 4{,}096$, instead, the fraction of errors decreases *before* the onset of memorization: the model generates valid data which are not copies, and it is thus generalizing. The generalizing models ($P = 4{,}096$ and $P = 16{,}384$) present a dynamical phase $\tau^* < \tau < \tau_{\rm mem}$ where they achieve nearly perfect generalization before starting to memorize, which becomes longer as P increases.

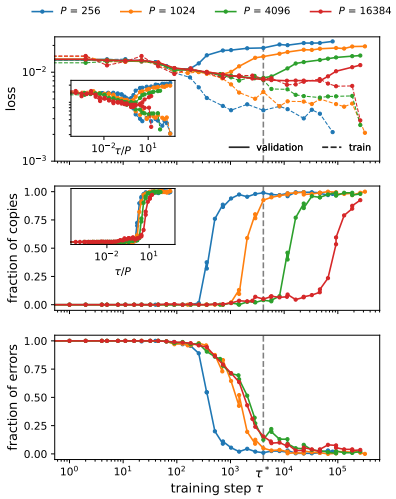
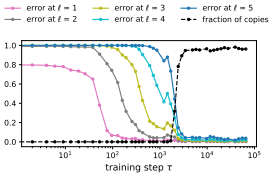
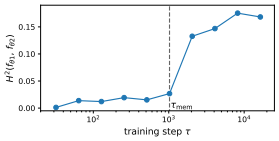


Figure 3. **RHM.** For $P = 256$, the diffusion model generates valid data only when it is memorizing the training data. For $P = 16384$, instead, the model generalizes, approximately at τ^*, before starting to memorize. The memorization time scales linearly in P (insets).



(a) Layer-wise learning in the RHM before memorization.



(b) Distance between the outputs of two-diffusion models trained on disjoint training sets.

Figure 4. **Partial generalization in RHM.** (a) The model learns progressively deeper rules during training. (b) Two models trained on disjoint training sets learn the same score before memorization.

Partial generalization For $P < P^*$, the diffusion model does not have enough data to learn the deeper rules. Yet, it can still learn the lower levels up to $\tilde{\ell}$, with $P > P^*_{\tilde{\ell}}$. In this case, the model achieves *partial generalization*, corresponding to learning to generate data with local coherence but lacking a global one, consistent with observations of Figure 2. In Fig. 4(a), a diffusion model is trained with $P = 1{,}024$ training points of an RHM with $L = 5$ ($P^* = P_L \simeq 10^4$). During training, we generate data and measure if they are compatible with the rules at level ℓ. The errors at $\ell \leq 3$ decrease at training times depending on ℓ, in accordance with $\tau_\ell \propto P^*_\ell$. However, for $\ell > 3$, the fractions of errors reach small values only at the onset of memorization $\tau_{\rm mem}$, when the fraction of copies goes up. This implies that the model never learns the rules at the deeper levels $\ell = 4, 5$, since the number of training data is smaller than the sample complexity, and generates data with global consistency only when it starts memorizing. Even without achieving perfect generalization, diffusion models gradually improve generalization during training – before memorizing – by capturing some structure of the data distribution. For the RHM, it corresponds to the lowest grammar levels. Hence, the score function learned *before memorization* is the same *independently* of the sampling of the training set. In Fig. 4(b), we train two models on disjoint training sets. We measure the difference in their outputs during training via the Hellinger distance, which remains low until it jumps to higher values when the models start memorizing. The two diffusion models learn the same score when generalization is improving, before overfitting their respective empirical scores.

5. Conclusion

We argued that the learning dynamics in diffusion models is best understood as a competition between time scales, as summarized in Figure 12. A larger training set implies a larger memorization time, thus opening a larger time window to generate more coherent data. These results open new avenues for fine control of copyright issues, using early stopping to avoid memorization.

References

Achilli, B., Ventura, E., Silvestri, G., Pham, B., Raya, G., Krotov, D., Lucibello, C., and Ambrogioni, L. Losing dimensions: Geometric memorization in generative diffusion. *arXiv preprint arXiv:2410.08727*, 2024.

Achilli, B., Ambrogioni, L., Lucibello, C., Mézard, M., and Ventura, E. Memorization and generalization in generative diffusion under the manifold hypothesis. *arXiv preprint arXiv:2502.09578*, 2025.

Ambrogioni, L. The statistical thermodynamics of generative diffusion models. *arXiv preprint arXiv:2310.17467*, 2023.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Biroli, G. and Mézard, M. Generative diffusion in very large dimensions. *arXiv preprint arXiv:2306.03518*, 2023.

Biroli, G., Bonnaire, T., de Bortoli, V., and Mézard, M. Dynamical regimes of diffusion models, 2024.

Block, A., Mroueh, Y., and Rakhlin, A. Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

Cagnetta, F. and Wyart, M. Towards a theory of how the structure of language is acquired by deep neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Cagnetta, F., Petrini, L., Tomasini, U. M., Favero, A., and Wyart, M. How deep neural networks learn compositional data: The random hierarchy model. *Phys. Rev. X*, 14: 031001, Jul 2024. doi: 10.1103/PhysRevX.14.031001.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Chen, C., Liu, D., and Xu, C. Towards memorization-free diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8425–8434, 2024.

Chomsky, N. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

Cui, H., Krzakala, F., Vanden-Eijnden, E., and Zdeborová, L. Analysis of learning a flow-based generative model from limited sample complexity. *arXiv preprint arXiv:2310.03575*, 2023.

Dockhorn, T., Cao, T., Vahdat, A., and Kreis, K. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022.

Favero, A., Sclocchi, A., Cagnetta, F., Frossard, P., and Wyart, M. How compositional generalization and creativity improve as diffusion models are trained. *arXiv preprint arXiv:2502.12089*, 2025.

Garnier-Brun, J., Mézard, M., Moscato, E., and Saglietti, L. How transformers learn structured data: insights from hierarchical filtering. *arXiv preprint arXiv:2408.15138*, 2024.

George, A. J., Veiga, R., and Macris, N. Denoising score matching with random features: Insights on diffusion models from precise learning curves. *arXiv preprint arXiv:2502.00336*, 2025.

Grenander, U. *Elements of pattern theory*. JHU Press, 1996.

Gu, X., Du, C., Pang, T., Li, C., Lin, M., and Wang, Y. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=D3DBqvSDbj.

Han, Y., Razaviyayn, M., and Xu, R. Neural network-based score estimation in diffusion models: Optimization and generalization. *arXiv preprint arXiv:2401.15604*, 2024.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Hu, H. and Pang, J. Membership inference of diffusion models. *arXiv preprint arXiv:2301.09956*, 2023.

Jin, Y. and Geman, S. Context and hierarchy in a probabilistic image model. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 2145–2152. IEEE, 2006.

Joshi, A. K. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Dowty, D. R., Karttunen, L., and Zwicky, A. M. (eds.), *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pp. 206–250. Cambridge Univ. Press, Cambridge, UK, 1985.

Kadkhodaie, Z., Guth, F., Simoncelli, E. P., and Mallat, S. Generalization in diffusion models arises from geometry-adaptive harmonic representations. *arXiv preprint arXiv:2310.02557*, 2023.

Kamb, M. and Ganguli, S. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.

Krishnan, S., Xiao, Y., and Saurous, R. A. Neumann optimizer: A practical optimization algorithm for deep neural networks. *arXiv preprint arXiv:1712.03298*, 2017.

Li, M. and Chen, S. Critical windows: non-asymptotic theory for feature emergence in diffusion models. In *International Conference on Machine Learning*, pp. 27474–27498. PMLR, 2024.

Li, P., Li, Z., Zhang, H., and Bian, J. On the generalization properties of diffusion models. *Advances in Neural Information Processing Systems*, 36:2097–2127, 2023.

Liu, Z., Luo, P., Wang, X., and Tang, X. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15 (2018):11, 2018.

Manning, C. D. and Schütze, H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999. doi: 10.7551/mitpress/10438.001.0001.

Matsumoto, T., Miura, T., and Yanai, N. Membership inference attacks against diffusion models. In *2023 IEEE Security and Privacy Workshops (SPW)*, pp. 77–83. IEEE, 2023.

Mei, S. U-nets as belief propagation: Efficient classification, denoising, and diffusion in generative hierarchical models. *arXiv preprint arXiv:2404.18444*, 2024.

Mezard, M. and Montanari, A. *Information, physics, and computation*. Oxford University Press, 2009.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Oko, K., Akiyama, S., and Suzuki, T. Diffusion models are minimax optimal distribution estimators. *arXiv preprint arXiv:2303.01861*, 2023.

Pizzi, E., Roy, S. D., Ravindra, S. N., Goyal, P., and Douze, M. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14532–14542, 2022.

Pullum, G. K. and Gazdar, G. Natural languages and context-free languages. *Linguist. Philos.*, 4(4):471–504, 1982. doi: 10.1007/BF00360802.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Rozenberg, G. and Salomaa, A. *Handbook of Formal Languages*. Springer, January 1997. doi: 10.1007/978-3-642-59126-6.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35: 25278–25294, 2022.

Sclocchi, A., Favero, A., Levi, N. I., and Wyart, M. Probing the latent hierarchical structure of data via diffusion models. *arXiv preprint arXiv:2410.13770*, 2024a.

Sclocchi, A., Favero, A., and Wyart, M. A phase transition in diffusion models reveals the hierarchical nature of data. *arXiv preprint arXiv:2402.16991*, 2024b.

Shah, K., Chen, S., and Klivans, A. Learning mixtures of gaussians using the ddpm objective. *Advances in Neural Information Processing Systems*, 36:19636–19649, 2023.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. K. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

Siskind, J. M., Sherman, J., Pollak, I., Harper, M. P., and Bouman, C. A. Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1504–1519, 2007.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Diffusion art or digital forgery. *Investigating Data Replication in Diffusion Models*, 2022.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Vyas, N., Kakade, S. M., and Barak, B. On provable copyright protection for generative models. In *International conference on machine learning*, pp. 35277–35299. PMLR, 2023.

Wang, W., Sun, Y., Yang, Z., Hu, Z., Tan, Z., and Yang, Y. Replication in visual diffusion models: A survey and outlook. *CoRR*, 2024.

Wu, Y., Yu, N., Li, Z., Backes, M., and Zhang, Y. Membership inference attacks against text-to-image generation models. *arXiv preprint arXiv:2210.00968*, 2022.

Yang, G. and Hu, E. J. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

Yoon, T., Choi, J. Y., Kwon, S., and Ryu, E. K. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

Zhang, H., Zhou, J., Lu, Y., Guo, M., Wang, P., Shen, L., and Qu, Q. The emergence of reproducibility and generalizability in diffusion models. *arXiv preprint arXiv:2310.05264*, 2023.

A. Related work

Memorization in diffusion models Several works have documented the tendency of diffusion models to memorize the training data (Carlini et al., 2023; Somepalli et al., 2022; 2023; Wang et al., 2024). Dockhorn et al. (2022) propose a mitigation strategy based on differentially private stochastic gradient descent, while Chen et al. (2024) introduce an anti-memorization guidance. Yoon et al. (2023); Kadkhodaie et al. (2023); Gu et al. (2025) interpret memorization as an overfitting phenomenon driven by the large capacity of overparameterized neural networks. In particular, Kadkhodaie et al. (2023) show that underparametrized models trained on disjoint training sets learn the same score function, therefore generalizing by sampling the same target distribution; in contrast, overparametrized models memorize their respective training data.

Theory of diffusion Under mild assumptions on the data distribution, diffusion models achieve a sample complexity scaling exponentially with data dimension (Block et al., 2020; Oko et al., 2023). The sampling and memorization process has been studied for Gaussian mixtures and linear manifolds using the empirical score function (Biroli et al., 2024; Ambrogioni, 2023; Achilli et al., 2024; 2025; Li & Chen, 2024). Learning the empirical score function was studied in (Cui et al., 2023; Shah et al., 2023; Han et al., 2024). The memorization-generalization trade-off in terms of model capacity with random features was studied in (George et al., 2025). Generalization bounds for early-stopped random features learning simple score functions were derived in (Li et al., 2023). Biroli & Mézard (2023); Ambrogioni (2023); Biroli et al. (2024) show for Gaussian mixtures the existence of a characteristic noise level during the diffusion process where the single modes merge into one. In (Biroli et al., 2024), another noise scale is identified, corresponding to short diffusion times, where the backward process collapses into the single training data points, associated with memorization. Kamb & Ganguli (2024) study generalization in vision diffusion models through the inductive bias of translational equivariance and locality.

Diffusion models for hierarchical data For hierarchically structured data, Sclocchi et al. (2024b;a) show that the reconstruction of high-level features undergoes a phase transition in the diffusion process, while low-level features vary smoothly around the same noise scale. For the same data model, Favero et al. (2025) show that UNet diffusion models learn to generate these data by sequentially learning different levels of the grammatical rules, with a sample complexity polynomial in data dimension. Sclocchi et al. (2024b) show that Bayes-optimal denoising algorithm for hierarchical data corresponds to belief propagation, Mei (2024) shows that U-Net architectures are able to efficiently approximate this algorithm. Moreover, Garnier-Brun et al. (2024) show that transformers can implement the same algorithm.

B. Experimental Details

B.1. Vision diffusion models

iDDPM In our experiments, we utilize Improved Denoising Diffusion Probabilistic Models (iDDPMs) for image generation on the CIFAR-10 and CelebA datasets, following the codebase of Improved DDPMs (Nichol & Dhariwal, 2021): `https://github.com/openai/improved-diffusion`. Specifically, we train iDDPMs with 256 and 128 channels for CIFAR-10 and CelebA, respectively. Our models are implemented using a U-Net architecture with attention layers and 3 resolution blocks. We use $4,000$ diffusion steps, a cosine noise schedule, a learning rate of 10^{-4}, and a batch size of 128. Training is performed for $262{,}144$ steps using a *hybrid objective* (Nichol & Dhariwal, 2021) and the Adam optimizer with dropout of 0.3.

Stable Diffusion We fine-tune Stable Diffusion v2.1[1] using the codebase `https://github.com/somepago/DCR` from (Somepalli et al., 2022; 2023). The model is pre-trained on LAION-2B (Schuhmann et al., 2022) and consists of a latent diffusion U-Net architecture with frozen text and autoencoder components. We fine-tune the U-Net for $262{,}144$ steps on $8{,}192$ images from the LAION-10k dataset at resolution 256×256, using a batch size of 16. We employ a constant learning rate of 5×10^{-6} with $5{,}000$ warm-up steps and use a single image-caption pair per datapoint.

B.2. Language diffusion models

MD4 Our experiments leverage the codebase of MD4 (Shi et al., 2024), available at `https://github.com/google-deepmind/md4`. MD4 is a masked diffusion model that progressively transforms tokens into a special [MASK] token as training proceeds. Specifically, at each timestep t, each non-masked token has a probability β_t of being replaced by

[1] `https://huggingface.co/stabilityai/stable-diffusion-2-1`

[MASK]. The forward transition process for this model can be formally described using a one-hot encoding of the $|\mathcal{V}|+1$ states, where the transition matrix is defined as:

$$Q_t = (1-\beta_t)\mathbb{I} + \beta_t \mathbf{1}\mathbf{e}_M^\top. \tag{1}$$

Here $\mathbb{I}$ the identity matrix, $\mathbf{1}$ a vector of ones and $\mathbf{e}_M$ the one-hot-encoding vector corresponding to the [MASK] symbol. The entries $[Q_t]_{ij}$ of Q_t indicate the probability of the token x_k transitioning from state i to state j, i.e., $[Q_t]_{ij} = q(x_{k,t} = j | x_{k,t-1} = i)$. At the final timestep T, all tokens are fully masked, i.e., $x_{k,T} = [\mathsf{MASK}]$ for every $k \in [\dim(x)]$. For our experiments, we train MD4 using a batch size of 64 and a context size of 256. All other hyperparameters are kept consistent with the original MD4 implementation.

B.3. Random Hierarchy Model

D3PM For our experiments on the Random Hierarchy Model, we employ convolutional U-Net-based Discrete Denoising Diffusion Probabilistic Models (D3PMs) (Austin et al., 2021). These models are tasked to predict the conditional expectation $\mathbb{E}(x_0|x_t)$, which parameterizes the reverse diffusion process. In particular, we consider a uniform diffusion process (Hoogeboom et al., 2021; Austin et al., 2021), where, at each timestep t, tokens can either stay unchanged or, with probability β_t, can transition to some other symbol in the vocabulary. One-hot encoding the $|\mathcal{V}|$ states, the forward transition matrix formally reads:

$$Q_t = (1-\beta_t)\mathbb{I} + \frac{\beta_t}{|\mathcal{V}|}\mathbf{1}\mathbf{1}^\top. \tag{2}$$

Here $\mathbb{I}$ is the identity and $\mathbf{1}$ is a vector of all ones. At the final time T, the stationary distribution is uniform over the vocabulary. The convolutional U-Net has L resolution blocks in both the encoder and decoder parts. Each block features the following specification: filter size s, stride s, 8,192 channels per layer, GeLU non-linearity, skip connections linking encoder and decoder blocks of matching resolution to preserve multi-scale feature information. We include embedding and unembedding layers implemented as convolutional layers with a filter size of 1. This architecture is specifically aligned with the RHM's hierarchical structure, where the filter size and stride of s in the convolutional layers mirror the branching factor of the RHM tree. While this design provides practical benefits in terms of training efficiency, it should not alter the fundamental sample complexity of the problem, as long as the network is sufficiently deep and expressive (Cagnetta et al., 2024). The networks are initialized with the maximal-update (μP) parameterization (Yang & Hu, 2020), ensuring stable feature learning even in the large-width regime. We train with Adam with a learning rate of 0.1 and a batch size of 32. For the diffusion process, we adopt a linear schedule with 1,000 noise levels.

B.4. Hardware

All experiments are run on a single NVIDIA H100 SXM5 GPU with 94GB of RAM.

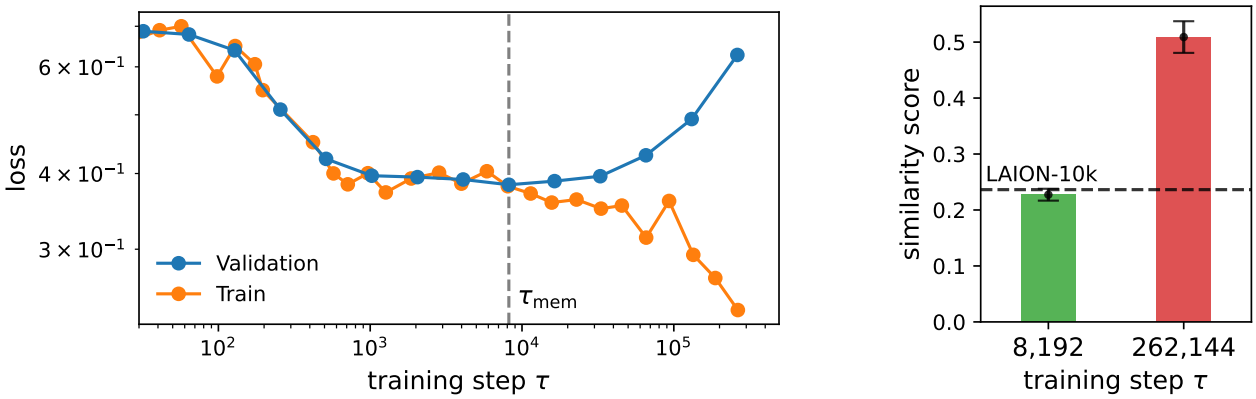
C. Experiments on Stable Diffusion

We consider Stable Diffusion v2.1 (Ronneberger et al., 2015), a text-to-image latent diffusion model pre-trained on the LAION-2B dataset(Schuhmann et al., 2022). We fine-tune this model for 262,144 steps on $8,192$ samples from the LAION-10k dataset (Somepalli et al., 2023), using a resolution of 256×256. During fine-tuning, the text encoder and encoder-decoder components are kept frozen. We use a held-out validation set of 1,024 image-text pairs to monitor the validation loss. Full training details are provided in Appendix B.

To quantify memorization, we follow the protocol of Somepalli et al. (2022) and compute a similarity score for each generated image based on the cosine similarity of SSCD (Self-Supervised Descriptor for Image Copy Detection) (Pizzi et al., 2022) features, extracted from a ResNet-50 model. Each score is defined as the similarity between a generated image and its nearest neighbor in the training set.

5(a) plots the training and validation losses as a function of the training step τ. As observed in the main text, initially, both losses decrease, indicating generalization: the model output aligns increasingly with the population score. At a critical time τ_{mem}, the validation loss diverges from the training loss, marking the onset of memorization. Early stopping at this point can prevent the model from entering the memorization phase.

In 5(b), we report the similarity scores for 200 generated images at two checkpoints: early stopping ($\tau = 8{,}192$) and the final training step ($\tau = 262{,}144$). For reference, we also show the similarity score for real images from the full LAION-10k



(a) Generalization-memorization dynamics.

(b) Similarity scores.

Figure 5. **Memorization dynamics in Stable Diffusion.** (a) Training and validation losses as a function of training step τ for Stable Diffusion fine-tuned on LAION-10k. Both losses initially decrease, indicating generalization, and diverge at the memorization onset time τ_{mem}. (b) Cosine similarity scores between SSDC ResNet embedding for generated images and their nearest training neighbor at early stopping ($\tau = 8{,}192$) and final training ($\tau = 262{,}144$). The dashed line indicates the mean similarity score between the closest LAION-10k samples. The sharp increase at late training signals memorization.

Figure 6. **Replicates generated by Stable Diffusion.** Example generations (left) from the final training checkpoint ($\tau = 262{,}144$) with similarity score > 0.5 to their nearest neighbor in the training set (right), confirming memorization.

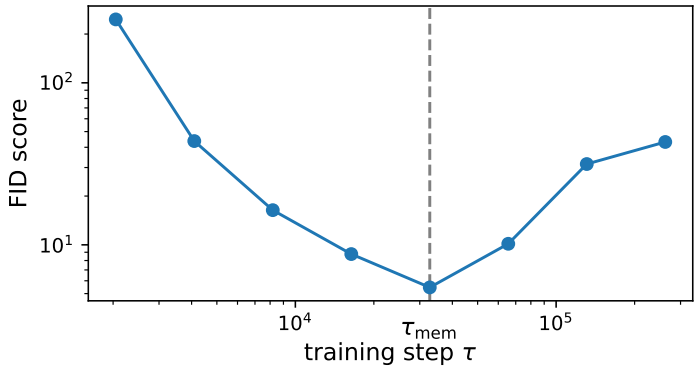


Figure 7. **FID dynamics.** Fréchet Inception Distance (FID) as a function of training step τ for a DDPM trained on 16,384 CIFAR-10 images. The FID initially decreases, reflecting improved generation quality and diversity, but begins to rise past τ_{mem} as the model starts copying training examples.

dataset (black dashed line). At the early stopping time, the generated images exhibit diversity similar to that of the dataset. In contrast, by the end of training, the similarity score increases by a factor of two, indicating memorization.

Finally, in Figure 6, we show representative examples of replicated samples (similarity score > 0.5) from the final checkpoint, confirming that Stable Diffusion memorized part of its training set.

D. Further Results on iDDPMs

FID dynamics Figure 7 reports the Fréchet Inception Distance (FID) as a function of the training step τ for a DDPM trained on 16,384 CIFAR-10 images, consistent with the setup in Figure 1. At each checkpoint, we generate 32,768 samples and compute the FID against the union of CIFAR-10 standard train and test splits. The FID captures both the quality and diversity of the generated images. As training progresses, the FID decreases monotonically until the memorization onset time τ_{mem}, after which it gradually increases – reflecting a loss in sample diversity as the model begins replicating its training data.

Further examples of generations Figure 8 presents further images sampled from the early stopped iDDPM trained on 16,384 CIFAR-10 images.

Examples of copies Figure 9 shows examples of generated samples (top row) and their nearest neighbors in the training set (bottom row) for the iDDPM trained on 8,192 CIFAR-10 images. These examples are taken from the end of training, within the memorization phase, where the model begins to replicate its training data.

E. Experiments on Diffusion LLMs

We extend our analysis of generalization and memorization to language data, using MD4, a masked diffusion model specifically designed for text (Shi et al., 2024). Our experiments are conducted on the text8 dataset, a standard benchmark for language modeling based on Wikipedia, with character-level tokenization. To the best of our knowledge, this is the first demonstration of memorization in the language diffusion setting.

We train MD4 from scratch using a standard GPT-like transformer architecture with approximately 165M parameters. Following the masked diffusion approach, the model is trained to predict masked tokens in noisy text sequences, effectively learning a score function over text data. Full details are presented in Appendix B. We use training set sizes $P \in \{64, 128, 256, 512, 1024\}$ ranging from 16,384 to 262,144 tokens. We track model performance using the validation loss on 19,531 sentences, which provide a lower bound to the negative log likelihood, and monitor memorization by generating 1,024 text samples at regular training checkpoints.

Figure 8. **CIFAR-10 samples generated with early-stopped model.** Additional samples from the iDDPM trained on 16,384 CIFAR-10 images, generated at the early stopping point before memorization. The model produces diverse and high-quality images without replicating the training data.

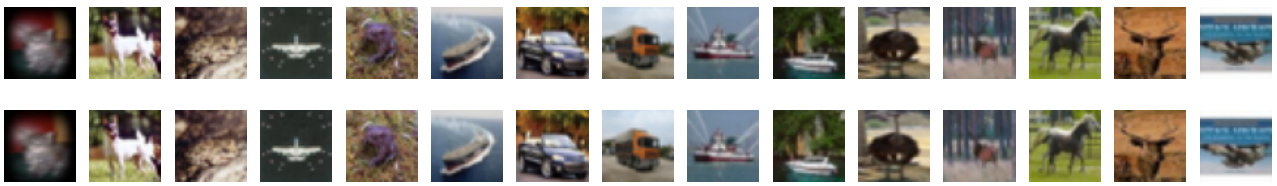


Figure 9. **Examples of copies on CIFAR-10.** Top: samples generated by the iDDPM trained on 8,192 CIFAR-10 images at the end of training. Bottom: nearest neighbors from the training set. The model reproduces specific training examples, indicating memorization.

Memorization is quantified by calculating the Hamming distance between each generated text sample and the closest training set text, averaged over the generations and divided by the sequence length. This metric captures the fraction of exact token matches between the generated and training text.

Results and analysis Figure 10 presents the results of this experiment. As with the vision diffusion models, MD4 initially generalizes, improving the log-likelihood on the validation corpus. However, after τ_{mem} the model begins to produce exact or near-exact copies of training text, signaling the onset of memorization. Notably, τ_{mem} scales linearly with the training set size P, consistent with our previous findings. The transition to memorization is also marked by a sudden increase in the validation loss, indicating that early stopping can effectively prevent memorization also in this setting.

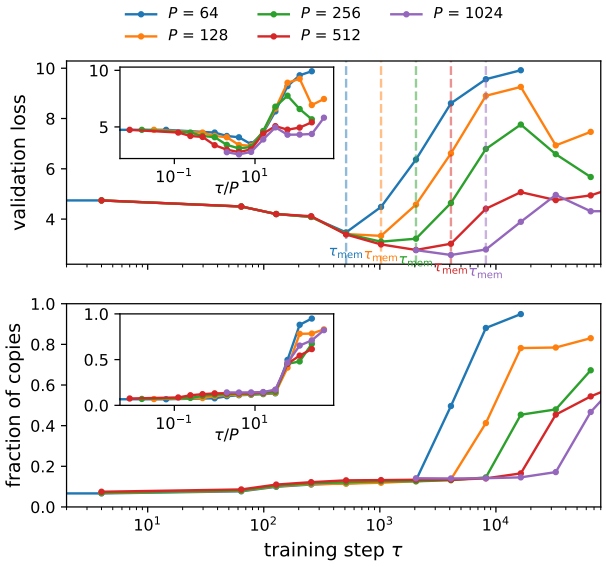


Figure 10. **Memorization dynamics in language diffusion models.** Train loss, validation loss, and fraction of copied text as a function of training steps for GPT-based MD4 models trained on text8 with character-level tokenization and varying training set sizes P. Both losses decrease initially, indicating generalization, but diverge at the onset of memorization (τ_{mem}), where the models start copying training text. τ_{mem} grows linearly with P (insets).

F. The Random Hierarchy Model

F.1. Probabilistic graphical models

In theoretical linguistics, *Probabilistic Context-Free Grammars* (PCFG) have been proposed as a framework to describe the hierarchical structure of the syntax of several languages (Chomsky, 1956; Rozenberg & Salomaa, 1997; Pullum & Gazdar, 1982; Joshi, 1985; Manning & Schütze, 1999). Moreover, they have been proposed for describing semantic aspects of images under the name of *Pattern Theory* (Grenander, 1996; Jin & Geman, 2006; Siskind et al., 2007). PCFGs consist of a vocabulary of latent (*nonterminal*) symbols and a vocabulary of visible (*terminal*) symbols, together with probabilistic *production rules* establishing how one latent symbol generates tuples of latent or visible symbols.

The Random Hierarchy Model (RHM) The RHM (Cagnetta et al., 2024) is a simple PCFG introduced as a theoretical toy model describing hierarchy and compositionality in data. With respect to generic PCFGs, it is built with some simplifying assumptions:

- Symbols are organized in a regular-tree topology of depth L and branching factor s. The bottom layer, indexed as $\ell = 0$, corresponds to the leaves of the tree, which are the visible (terminal) symbols. The upper part of the tree, with layers $\ell = 1, \dots, L$, corresponds to latent (nonterminal) symbols in the data structure.
- Nonterminal symbols are taken from L finite vocabularies $(\mathcal{V}_\ell)_{\ell=1,\dots,L}$ of size v for each layer $\ell = 1, \dots, L$. Terminal symbols belong to the vocabulary $\mathcal{V} \equiv \mathcal{V}_0$ of size v.
- The production rules transform one symbol in a node at level $\ell + 1$ into a string of s symbols in its children nodes at level ℓ. For each non-terminal symbol, there are m rules with equal probability, which are *unambiguous*, i.e., two distinct symbols cannot generate the same s-string. Rules are sampled randomly without replacement and frozen for a given instance of the RHM. The m strings generated by the same latent symbol are referred to as *synonyms*.

The fixed tree topology ensures that visible data at the leaves are strings of fixed length $d = s^L$, corresponding to the data dimension d. In analogy with language modeling, we call visible symbols *tokens*.

The number of possible data generated by this model is $vm^{\frac{d-1}{s-1}}$, therefore exponential in the data dimension. Because of the random production rules, the tokens of the RHM data have non-trivial correlations reflecting the latent hierarchical structure (Cagnetta & Wyart, 2024).

F.2. Diffusion on the Random Hierarchy Model

The exact score function of the RHM Because of its correlations, the probability distribution of the RHM data and its corresponding score function are highly non-trivial. Nevertheless, if the production rules are known, thanks to the latent tree structure, the score function for any noise level can be computed exactly using the Belief Propagation (BP) algorithm (Mezard & Montanari, 2009). Given a noisy RHM datum, Belief Propagation computes the marginal probabilities, conditioned on this noisy observation, of the symbols in any node of the tree. Computing the expectations from these conditional probabilities at the leaf nodes corresponds to computing the score function, which can be used to reverse a diffusion process. Moreover, BP also provides a way to sample directly from these posterior probabilities, corresponding to a perfect integration of the backward diffusion process. The exact sampling of diffusion processes with the RHM data was studied in (Sclocchi et al., 2024b;a).

Sample complexity Favero et al. (2025) studied the sample complexity for diffusion models based on deep neural networks trained on finite RHM data. Their main findings are the following.

- The sample complexity to learn to generate valid data depends on the parameters of the model as $P^* \sim vm^{L+1}$, which is polynomial in the dimension, i.e., $P^* \sim vmd^{\log m/\log s}$. This scale can be theoretically predicted by comparing the size of the correlations between tokens and latent features, used in deep architectures for denoising, with their sampling noise.
- For $P < P^*$, there are regimes of partial generalization where the generated data are consistent with the rules up to layer ℓ. The sample complexity to learn the rules at layer ℓ scales as $P^*_\ell \sim vm^{\ell+1}$.
- When $P > P^*_\ell$, the number of training steps τ^*_ℓ required to learn the rules at layer ℓ is proportional to P^*_ℓ, therefore having the same polynomial scaling with the dimension. Complete generalization is therefore achieved with $\tau^* \propto P^* = P^*_L$ number of training steps.

Notice that the sample complexity depends on the underlying distribution, e.g., the parameters of the grammar, and not on the specific number of available training samples.

G. Further Results on the RHM

Production rules sampling Figure 11 shows the mean occurrence and centered covariance of the production rules sampled by a diffusion model trained on $P = 16{,}384$ strings ($v = 16$, $m = 4$, $L = 3$, $s = 2$). The model, trained with early stopping ($\tau = 32{,}768$), samples all RHM rules with a mean occurrence that is approximately uniform (up to sampling noise); likewise, the correlations between the cooccurrence of sampled rules show that they are sampled approximately independently. Therefore, the generated data reproduce the correct data distribution of the RHM, corresponding to generalization.

Phase diagram Figure 12 summarizes the different regimes of generalization and memorization in a phase diagram.

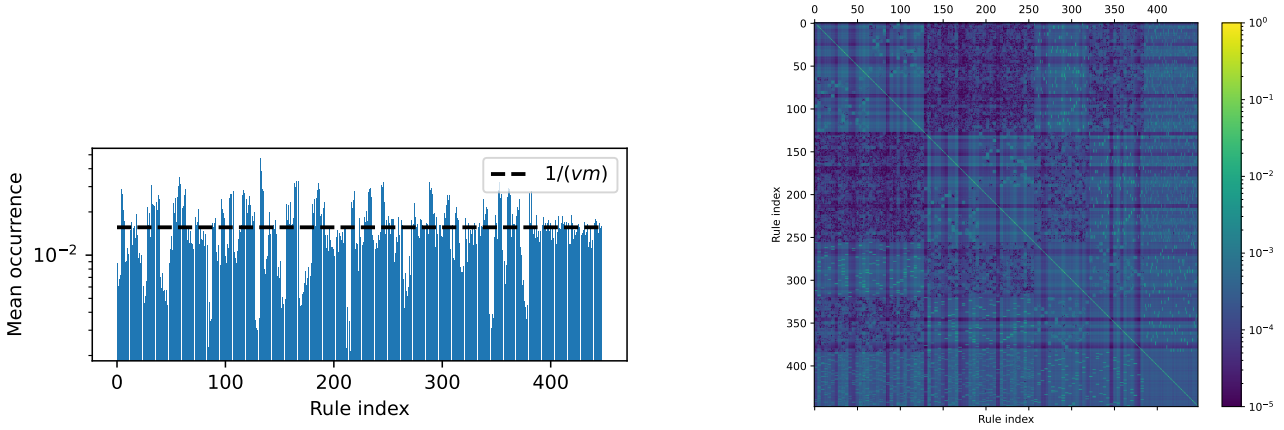


Figure 11. **Sampling of RHM production rules.** Mean occurrence (*left*) and centered covariance (*right*) of the production rules sampled by a diffusion model trained on $P = 16{,}384$ strings ($v = 16$, $m = 4$, $L = 3$, $s = 2$). The model, trained with early stopping ($\tau = 32{,}768$), samples all RHM rules with a mean occurrence that is approximately uniform (up to sampling noise). Likewise, the correlations between the cooccurrence of sampled rules show that they are sampled approximately independently.

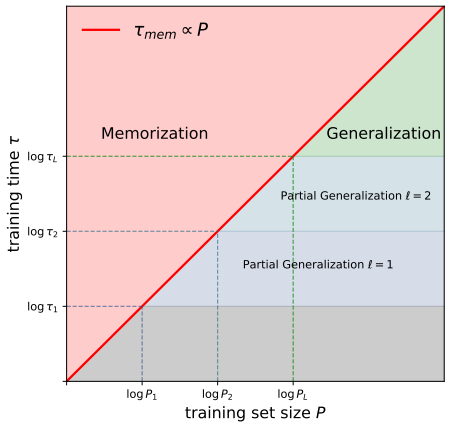


Figure 12. **Diagram of generalization vs. memorization** indicating different regimes as a function of training time τ and sample complexity P. In the simplest version of the RHM, learning proceeds by well-distinct steps, while it is smoother for natural data.