A theory of learning data statistics in diffusion models, from easy to hard

Lorenzo Bardone, Claudia Merger and Sebastian Goldt

International School of Advanced Studies (SISSA)
Trieste, Italy
{lbardone, cmerger, sgoldt}@sissa.it

Abstract

While diffusion models have emerged as a powerful class of generative models, their learning dynamics remain poorly understood. We address this issue first by empirically showing that standard diffusion models trained on natural images exhibit a simplicity bias, learning simple, pair-wise input statistics first before specializing to higher-order correlations. We reproduce this behaviour in simple denoisers trained on a minimal data model, the mixed cumulant model, where we precisely control both pair-wise and higher-order correlations of the inputs. We identify a scalar invariant of the model that governs the sample complexity of learning pair-wise and higher-order correlations that we call the *diffusion information exponent*, in analogy to related invariants in different learning paradigms. Using this invariant, we prove that the denoiser learns simple, pair-wise statistics of the inputs at linear sample complexity, while more complex higher-order statistics, such as the fourth cumulant, require at least cubic sample complexity. We also prove that the sample complexity of learning the fourth cumulant is linear if pair-wise and higher-order statistics share a correlated latent structure. Our work describes a key mechanism for how diffusion models can learn distributions of increasing complexity and suggests that correlated latent structures may be at the core of how diffusion models are able to learn at low sample complexity.

1 Introduction

Introduced only ten years ago, diffusion models [15, 31, 32] quickly reached state-of-the-art performance on many tasks. Despite this empirical success, our theoretical understanding of why these models learn so efficiently remains limited compared to the standard, supervised learning setting, where some relevant questions have already been answered in genuine feature learning regimes. In particular, one universal property of neural networks that emerges from classical and recent studies is that neural networks exhibit a simplicity bias – they learn functions and distributions of increasing complexity sequentially (with respect to the number of steps and amount of data used), from easy to hard [3, 4, 12, 13, 16, 21, 22, 24, 25, 27–29].

Whether similar principles govern the *distributional* learning dynamics of diffusion models remains an open question. Indeed, while several works have examined the dynamics of score denoising in simplified models [7, 14], these analyses rely on random features or Gaussian data assumptions, that do not allow to investigate the presence of simplicity bias. A first study of the simplicity bias was done by Li et al. [18] for linear denoiser. A considerable step toward more complex settings was done by Cui et al. [9], who analysed the performance of a two-layer autoencoder. Building on previous work that described the dynamics of auto-encoders for (noisy) reconstruction by Refinetti and Goldt [23] and Cui and Zdeborová [8], Cui et al. [9] analyse the learning dynamics of an auto-encoder

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Workshop on Principles of Generative Modeling.

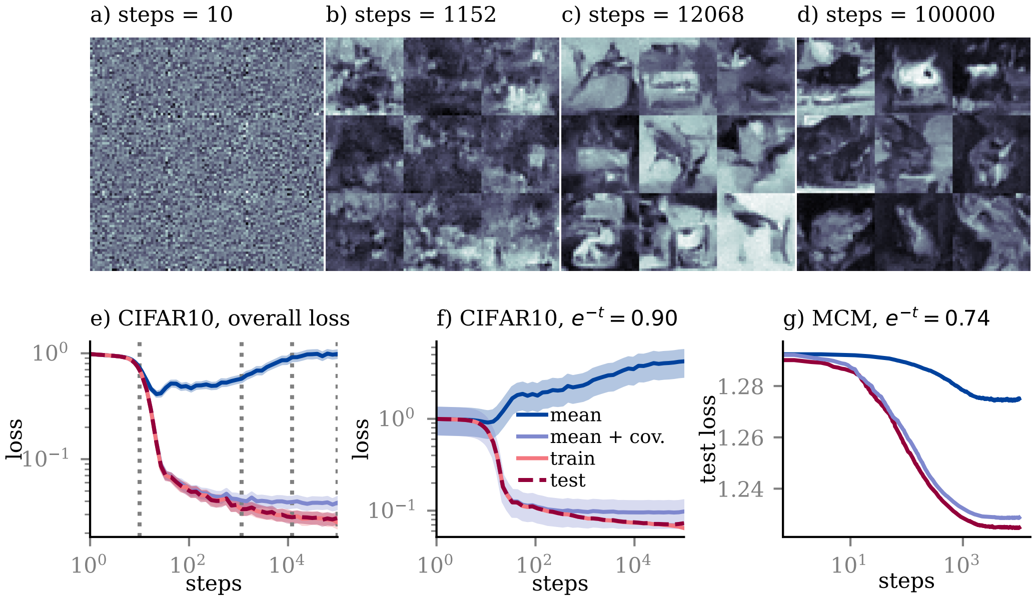

Figure 1: Sequential learning in diffusion models. a)-d) Samples generated from U-nets [26] on CIFAR-10 at various training stages. e) Test loss of diffusion model and loss on CIFAR-10 clones during training. Vertical dotted lines mark training stages of images generated from the model shown in panels a)-d). All curves are averages over 3 initializations of the network models and $5 \cdot 10^3$ test data. Shaded areas report standard deviation over random initialization. Panel f) reports the same as e), but for denoising samples with fixed level of noise $x = e^{-t}x_0 + \sqrt{1 - e^{-2t}}z$ where x_0 is a data point and noise $z \sim \mathcal{N}(0, \mathrm{Id})$. g) Test loss of neural network trained on the mixed cumulant model with dimension $d = 10^2$ an fixed level of noise, evaluated on clones of the data set. All curves in panel g) are averages over 5 initializations of the network model and 10^4 test data.

trained in a diffusion setting in the regime of linear sample complexity, where number of training samples is linear in the input dimension, $n \asymp d$. Here, we characterize the learning dynamics in a feature-learning regime with non-Gaussian data distributions *without the constraint of linear sample complexity*.

In this manuscript, we first demonstrate that diffusion models also learn simpler, pair-wise statistics of the inputs first before specializing to more complex, higher-orders statistics. We then introduce a solvable model of the learning dynamics of score diffusion that allows us to prove nearly sharp thresholds for the number of samples required by a simple denoiser to learn from pair-wise and higher-order correlations, respectively, thereby establishing a simplicity bias in a rigorous way.

2 A distributional simplicity bias in denoising diffusion models

We trained a U-net architecture [26] using denoising score matching [15, 32] on grayscale CIFAR-10 images [17]. We show the samples obtained from these models at various training stages and the loss on the denoising score objective in fig. 1 a)-f). During training, we also evaluate the test loss of the models on several “clones” of the CIFAR10 data set [24]. The cloned data sets are designed to only capture part of the statistics of the original CIFAR10 images. Each clone is a data set of inputs sampled from a Gaussian, whose mean (“mean”) or mean and covariance (“mean + cov”) have been fitted to the images in CIFAR-10. We detail their generation and show examples of the clone data sets in section B.1.

We test the performance of the diffusion model trained on CIFAR10 on "clones" of the data set to the performance on test data. Figure 1 shows that as training progresses, the diffusion model specializes more and more: at first, the performance on all test sets is equal. As training progresses, the models performance on the more specialized clones improves, whereas its performance on less specialized clones stagnates, meaning that the model has learned to exploit the statistics of the data that go beyond the one of the clones it outperforms. We repeat the same experiment with the CelebA data set and find the same sequential learning behaviour, see section B. We report details on the training procedure in B.2. This experiment substantiates the claim that lower order statistics are learnt in the initial phases of learning, whereas higher-order statistics are learnt later.

3 Theoretical analysis of score denoising

In the following, we will introduce a simplified setup where we can analyse the distributional simplicity bias of diffusion models rigorously. We will analyse the dynamics of projected stochastic gradient descent (pSGD) for a simple, non-linear denoising model trained on inputs sampled from a non-Gaussian distribution and non-linear neural networks. We will find a scalar invariant that describes the initial stages of the learning dynamics, which we call the **diffusion information exponent** k^* by analogy to similar invariants found for the single/multi-index problems like the information exponent [5]), generative exponent ([10]), or the leap index ([11]). The main difference compared to their setting are that they assumed an Gaussian distribution over inputs with identity covariance; here instead we will analyse a non-Gaussian input distribution, using the methodology of Bardone and Goldt [3]. The diffusion information exponent allows us to establish nearly sharp thresholds for the number of samples required by the denoiser to weakly-recover directions encoded by different cumulants of the inputs using online SGD, thereby allowing us to establish a distributional simplicity bias in a rigorous way.

3.1 Setup

Denoising diffusion We model the diffusion process following a standard approach, see for instance [6]; all the details are in section A.2. We consider a diffusion on a time interval $[0,T]$, with P_0 the unknown distribution that we want to learn to sample from, $P_T \approx \mathcal{N}(0,\mathbb{1}_d)$. The goal in diffusion models is learning the score of the density at intermediate times, which is defined as

$$\mathcal{F}_i(x,t) = \frac{\partial \log P_t(x)}{\partial (x_t)_i} = -\frac{(x_t)_i - \mathbb{E}\left[(x_0)_i | x(t)=x\right]e^{-t}}{\Delta_t}, \tag{1}$$

where the last equality, called *Tweedie's formula*, is at the core of the feasibility of diffusion models. It gives a recipe on how to approximate the score via empirical averages of the noised process.

The objective then becomes learning $\mathcal{F}$. To do this, one can build a mean-square objective for a collection of fixed time intervals. Let us denote $\mathcal{S}_t^w(x)$ the approximated score that depends on weight w. The loss function for diffusion time t can be rewritten (for the derivation, see section A.2)

$$\mathcal{L}(w) = \frac{1}{2}\mathop{\mathbb{E}}_{x_0\sim\mathbb{P}_0}\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\mathbb{1}_d)}\left[\left\|S_t^w(x_0e^{-t}+\sqrt{\Delta_t}z)+\frac{z}{\sqrt{\Delta_t}}\right\|^2\right]+C. \tag{2}$$

This quantity can be well approximated having just samples from P_0, which allows to estimate the integral over x_0. The additional Gaussian integral over z in eq. (2) is a term that is peculiar to diffusion models. We use *Stein's lemma* (lemma A.4 in the appendix), which was first used in a similar way in [30], to rewrite the loss in a way so that it is possible to apply the Hermite decomposition.

Denoiser and learning algorithm We consider the simplest feature learning setting, with a denoiser $S_t^w(x) = -x - \sigma(w\cdot x)w$ and optimize by updating an initial weight $w_0 \sim$ Unif($\mathbb{S}^{d-1}$) via online projected SGD (pSGD):

$$\tilde{w}_{\tau+1} = w_\tau - \eta_d \nabla_{\text{sph}}\mathscr{L}(w_\tau, x_\tau), \qquad w_{\tau+1} = \frac{\tilde{w}_{\tau+1}}{\|\tilde{w}_{\tau+1}\|} \tag{3}$$

where ∇_{sph} is the spherical gradient: $\nabla_{\text{sph}}f(w) = (\mathbb{1} - ww^\top)\nabla f(w)$. We simplify $\nabla_{\text{sph}}\mathscr{L}$, by expanding the square in eq. (2), differentiating with respect to w and then applying Stein lemma A.4 to remove the integral over z. At each training time τ, $\mathscr{L}$ is computed on a new independent sample $x_\tau \sim \mathbb{P}_t$. We obtain a formula that depends only on samples of $x \sim \mathbb{P}_t$, not z:

$$\nabla_{\text{sph}}\mathscr{L}_t(w,x) = (\mathbb{1}_d - ww^\top)F_\sigma(x\cdot w) \tag{4}$$

with

$$F_\sigma(x\cdot w) := \sigma''(x\cdot w) - \sigma'(x\cdot w)\sigma(x\cdot w) - \sigma(x\cdot w) - \sigma'(x\cdot w)x\cdot w, \tag{5}$$

The detailed derivation is given in appendix A.3.

Input distribution We draw samples from the *mixed cumulant model* of Bardone and Goldt [3]. This input model has a spike u that spikes the covariance, and another direction v that spikes the higher order cumulants. We construct samples $x^\mu,\ \mu \in [n]$ of the mixed cumulant model thus:

$$x^\mu = \sqrt{\beta_u}\lambda^\mu u + \sqrt{\beta_v}\nu^\mu v + z^\mu \tag{6}$$

where $\beta_u \in \mathbb{R}, \beta_v \in [0,1]$ are constant *signal to noise ratios* that modulate the intensity of the signal, $\lambda^\mu \sim \mathcal{N}(0,1)$ and $\nu^\mu \sim$ Rademacher$(1/2)$ are latent variables, and $z^\mu \sim \mathcal{N}(0, \mathbb{1} - \beta_v v v^\top)$ is high-dimensional noise. Hence samples from the mixed cumulant model appear Gaussian in all directions except v. The covariance of z^μ is chosen match the covariance of the mixed cumulant model to one with no cumulant spike, such that v must be discovered using only the higher order statistics of the data, making it harder to detect.

3.2 Main result

Intuition behind diffusion information exponent Let us focus for simplicity in the case with only the cumulant spike ($\beta_u = 0$), so the only relevant order parameter is the overlap $\alpha = w \cdot v$ between the weight vector of the denoiser and the cumulant spike. This overlap is small at initialization, of order $\alpha = \Theta\left(\frac{1}{\sqrt{d}}\right)$. We can expand both F and the likelihood ratio $L_t := \frac{\mathrm{d}\mathbb{P}_t}{\mathcal{N}(0,\mathbb{1}_d)}$ in Hermite polynomials (see definition A.2) with coefficients $(c_i^F)_{i\in\mathbb{N}}, (c_j^L)_{j\in\mathbb{N}}$. If the learning rate is small enough so that the noisy online process can be well approximated by the gradient flow of the population loss, we have that the dynamics are well approximated by the leading contribution after the Hermite expansion

$$\alpha_{\tau+1} = \alpha_\tau + \eta_d c^L_{k^*} c^F_{k^*-1} \alpha_\tau^{k^*-1} + O(\alpha^{k*}) \tag{7}$$

We call k^* as the *diffusion information exponent* and it is defined as the **lowest k such that the k-th coefficient of the likelihood ratio, and the $(k-1)$-th coefficient of F_σ are both non-zero**. From this, it can be shown that *it takes order of d^{k^*-1} steps*, hence samples since we are in online regime, to reach recovery of the spike v starting from a random initialization, as detailed in proposition A.8 in the appendix.

Simplicity bias in denoising diffusion We now add back the second spike β_u which is carried by the covariance. The dynamics in this case are more complex, and depend on whether correlations among latent variable are present or not, as illustrated by the next proposition.

Proposition 3.1. *Under assumption A.10, consider pSGD eq.* (3) *dynamics trained on $\mathscr{L}$ defined as eq.* (4)*, with data distributed as a* mixed cumulant model *eq.* (6)*. Then:*

1. *with* independent latent variables λ^μ, ν^μ *and learning rate $\eta_d \to 0$ as $d \to \infty$, as long as $n = o_d\left(\min\left(\frac{d}{\eta_d^2}, d\right)\right)$, we have that* $\lim_{d\to\infty}\sup_{\tau\le n}|w_\tau \cdot v| = 0$ *in L^p for every $p \ge 1$.*

2. *with a number of samples $n = \theta_d d$, with $\theta_d = \Omega(\log^2 d)$ and growing at most polynomially in d; step size η_d chosen so that $\frac{1}{\theta_d} \ll \eta_d \ll \frac{1}{\sqrt{\theta_d}}$ pSGD reaches* weak recovery *in a time $\tau_u \le n$ i.e. there exists η independent of d such that for $\tau \ge \tau_u$, with high probability $w_\tau \cdot u \ge \eta$. Moreover, in case of positive correlation of latent variables $\mathbb{E}[\lambda^\mu \nu^\mu] > 0$, conditioning on having matching sign at initialization: $v \cdot w_0 u \cdot w_0 > 0$,* weak recovery is achieved also for the cumulant spike v in a time $\tau_v \le n$.

The first part of proposition 3.1 is a negative result: the cumulant spike cannot be recovered at linear sample complexity. The second of this statement is a positive result: pSGD weakly recovers the covariance spike u, and hence learns about the pair-wise statistics, in quasi-linear sample complexity. For the cumulant spike v instead, we find that if the latent variables λ^μ and ν^μ are uncorrelated, pSGD will need at least d^3 samples to weakly recover v (as for single spike models, see proposition A.8), and hence learn about higher-order correlations. This clear separation of timescales rigorously establishes the distributional simplicity bias: the model learns pair-wise statistics (long) before higher-order correlations. However, if latent variables have a positive correlation, pSGD will recover the spike v with $\Theta(d\text{polylog}(d))$ samples. A similar speed-up due to correlated latent variables was found by Bardone and Goldt [3] for supervised classification. We provide the proof in section A.4. We show an example of a neural network trained on the mixed cumulant model in fig. 1 g); details on the training procedure are provided in section C.

4 Discussion

We have demonstrated experimentally that diffusion models learn distributions of increasing complexity as training progresses, where we defined the notion of complexity via the order of the cumulants of the data model which the neural network learns to exploit. We rigorously analyze training in the mixed cumulant model, which exhibits the same sequential learning property governed by the diffusion information exponent k^*, and highlighted how correlated latent variables facilitate learning. We have focused on a simple architecture for the denoiser that is capable of learning the exact score function, but we note that the diffusion information exponent k^* could also be computed for a mismatched denoiser. One limitation of the approach is that it applies to a model that features higher order statistics only in a single spatial direction. A fruitful direction for future research will to incorporate more complex models of data.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *Applied Mathematics Series*. National Bureau of Standards, 1964.

[2] A. S. Bandeira, D. Kunisky, and A. S. Wein. Computational Hardness of Certifying Bounds on Constrained PCA Problems. In T. Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 78:1–78:29. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020.

[3] L. Bardone and S. Goldt. Sliding down the stairs: How correlated latent variables accelerate learning with neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 3024–3045. PMLR, 2024.

[4] N. Belrose, Q. Pope, L. Quirke, A. Mallen, and X. Fern. Neural networks learn statistics of increasing complexity. *arXiv preprint arXiv:2402.04362*, 2024.

[5] G. Ben Arous, R. Gheissari, and A. Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *J. Mach. Learn. Res.*, 22(1), 2021.

[6] G. Biroli and M. Mézard. Generative diffusion in very large dimensions. *Journal of Statistical Mechanics: Theory and Experiment*, 2023(9):093402, Sept. 2023. ISSN 1742-5468. doi: 10.1088/1742-5468/acf8ba. URL `https://iopscience.iop.org/article/10.1088/1742-5468/acf8ba`.

[7] T. Bonnaire, R. Urfin, G. Biroli, and M. Mézard. Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training, May 2025. URL `http://arxiv.org/abs/2505.17638`. arXiv:2505.17638 [cs].

[8] H. Cui and L. Zdeborová. High-dimensional asymptotics of denoising autoencoders. *Advances in Neural Information Processing Systems*, 36:11850–11890, 2023.

[9] H. Cui, C. Pehlevan, and Y. M. Lu. A solvable model of learning generative diffusion: theory and insights. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=5b5wZg6Zeo`.

[10] A. Damian, L. Pillaud-Vivien, J. Lee, and J. Bruna. Computational-statistical gaps in gaussian single-index models (extended abstract). In S. Agrawal and A. Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1262–1262. PMLR, 30 Jun–03 Jul 2024. URL `https://proceedings.mlr.press/v247/damian24a.html`.

[11] Y. Dandi, F. Krzakala, B. Loureiro, L. Pesce, and L. Stephan. How two-layer neural networks learn, one (giant) step at a time. *Journal of Machine Learning Research*, 25(349):1–65, 2024.

[12] F. Farnia, J. Zhang, and D. Tse. A spectral approach to generalization and optimization in neural networks. In *ICLR*, 2018.

[13] A. Favero, A. Sclocchi, F. Cagnetta, P. Frossard, and M. Wyart. How compositional generalization and creativity improve as diffusion models are trained, Mar. 2025. URL `http://arxiv.org/abs/2502.12089`. arXiv:2502.12089 [stat].

[14] A. J. George, R. Veiga, and N. Macris. Analysis of Diffusion Models for Manifold Data, Feb. 2025. URL `http://arxiv.org/abs/2502.04339`. arXiv:2502.04339 [math].

[15] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models, Dec. 2020. URL `http://arxiv.org/abs/2006.11239`. arXiv:2006.11239 [cs, stat].

[16] D. Kalimeris, G. Kaplun, P. Nakkiran, B. L. Edelman, T. Yang, B. Barak, and H. Zhang. SGD on neural networks learns functions of increasing complexity. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3491–3501, 2019.

[17] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

[18] X. Li, Y. Dai, and Q. Qu. Understanding generalizability of diffusion models requires rethinking the hidden gaussian structure, 2024. URL `https://arxiv.org/abs/2410.24060`.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738. IEEE Computer Society, Dec. 2015. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.425. URL `https://www.computer.org/csdl/proceedings-article/iccv/2015/8391d730/12OmNzGlRCR`. ISSN: 2380-7504.

[20] P. McCullagh. *Tensor methods in statistics*. Courier Dover Publications, 2018.

[21] C. Merger, A. René, K. Fischer, P. Bouss, S. Nestler, D. Dahmen, C. Honerkamp, and M. Helias. Learning Interacting Theories from Data. *Physical Review X*, 13(4):041033, Nov. 2023. doi: 10.1103/PhysRevX.13.041033. URL `https://link.aps.org/doi/10.1103/PhysRevX.13.041033`. Publisher: American Physical Society.

[22] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. C. Courville. On the spectral bias of neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proc. of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 2019.

[23] M. Refinetti and S. Goldt. The dynamics of representation learning in shallow, non-linear autoencoders. In *International Conference on Machine Learning*, pages 18499–18519. PMLR, 2022.

[24] M. Refinetti, A. Ingrosso, and S. Goldt. Neural networks trained with sgd learn distributions of increasing complexity. In *International Conference on Machine Learning*, pages 28843–28863. PMLR, 2023.

[25] R. Rende, F. Gerace, A. Laio, and S. Goldt. A distributional simplicity bias in the learning dynamics of transformers. *Advances in Neural Information Processing Systems*, 37:96207–96228, 2024.

[26] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. URL `http://arxiv.org/abs/1505.04597`. arXiv:1505.04597 [cs].

[27] D. Saad and S. Solla. Exact Solution for On-Line Learning in Multilayer Neural Networks. *Phys. Rev. Lett.*, 74(21):4337–4340, 1995.

[28] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR*, 2014.

[29] A. M. Saxe, J. L. McClelland, and S. Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[30] K. Shah, S. Chen, and A. Klivans. Learning mixtures of gaussians using the ddpm objective, 2023. URL https://arxiv.org/abs/2307.01178.

[31] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[32] Y. Song and S. Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html.

[33] G. Szegő. *Orthogonal Polynomials*. American Mathematical Society colloquium publications. American mathematical society, 1939.

[34] E. Székely, L. Bardone, F. Gerace, and S. Goldt. Learning from higher-order correlations, efficiently: hypothesis tests, random features, and neural networks. *Advances in Neural Information Processing Systems*, 37:78479–78522, Dec. 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8f8af4eebc4e50994e0490898d891c96-Abstract-Conference.html.

A Details on the mathematical analysis

A.1 Hermite polynomials

We recall the definition and a few properties of the Hermite polynomials.

Definition A.1. The Hermite polynomial of degree m is

$$h_m(x) := (-1)^m e^{\frac{x^2}{2}} \frac{\mathrm{d}^m}{\mathrm{d}x^m} \left(e^{-\frac{x^2}{2}} \right) \tag{8}$$

There is also a general formula:

$$h_m(x) = m! \sum_{j=0}^{\lfloor m/2 \rfloor} \frac{(-1)^j}{2^j j!(n-2j)!} x^{m-2j} \tag{9}$$

The Hermite polynomials enjoy the following properties (for details see McCullagh [20], Szegő [33] and Abramowitz and Stegun [1]):

- they are an orthogonal system with respect to the L^2 product weighted with the density of the Normal distribution:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h_n(x) h_m(x) e^{-\frac{x^2}{2}} \mathrm{d}x = n! \delta_{m,n}; \tag{10}$$

- h_m is a monic polynomial of degree m, hence $(h_m)_{m \in \{1,\dots,N\}}$ generates the space of polynomials of degree $\leq N$;
- the previous two properties imply that the family of Hermite polynomials is an orthogonal basis for the Hilbert space $L^2(\mathbb{R}, \mathbb{Q})$ where $\mathbb{Q}$ is the normal distribution;
- they enjoy the following recurring relationship

$$h_{m+1}(x) = x h_m(x) - h_m'(x) = x h_m(x) - m h_{m-1}(x), \tag{11}$$

Multivariate case In the multivariate m-dimensional case we can generalize to Hermite tensors $(H_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^m}$ defined as:

$$H_\alpha(x_1,\ldots,x_m)=\prod_{i=1}^{m}h_{\alpha_i}(x_i) \tag{12}$$

most of the properties of the one-dimensional Hermite polynomials extend to this case: they form an orthogonal basis of $L^2(\mathbb{R}^m,\mathcal{N}(0,\mathbb{1})$. We have that:

$$\mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1})}[H_{\boldsymbol{\alpha}}(x)H_{\boldsymbol{\beta}}(x)]=\boldsymbol{\alpha}!\delta_{\boldsymbol{\alpha},\boldsymbol{\beta}} \tag{13}$$

Definition A.2 (Hermite expansion)**.** Consider a function $f:\mathbb{R}\to\mathbb{R}$ that is square integrable with weight the standard normal distribution $p(x)=(1/\sqrt{2\pi})\,e^{-x^2/2}$. Then, there exists a unique sequence of real numbers $\{c_k\}_{k\in\mathbb{N}}$ called Hermite coefficients, such that:

$$f(x)=\sum_{k=0}^{\infty}\frac{c_k}{k!}h_k(x)\quad\text{ and }\quad c_k(x):=\mathbb{E}_{x\sim\mathcal{N}(0,1)}[f(x)h_k(x)],$$

where h_i is the i-th probabilist's Hermite polynomial.

Definition A.3 (Information exponent)**.** Consider a function f that can be expanded with Hermite polynomials with coefficients $(c_i)_i\in\mathbb{N}$. Its information exponent $k^*=k^*(f)$ is the smallest index $k\geq 1$ such that $c_k\neq 0$.

The following lemma from [2] provides a version of the integration by parts technique that is tailored for Hermite polynomials.

Lemma A.4 (Stein lemma)**.** *Let $f:\mathbb{R}^d\to\mathbb{R}^d$ be a continuously differentiable k times function. Suppose that f and all of its partial derivatives up to the k-th are bounded by $O(\exp(|y|^\lambda))$ for a $\lambda\in(0,2)$, then for any $\boldsymbol{\alpha}\in\mathbb{N}^d$ such that $|\boldsymbol{\alpha}|\leq k$*

$$\langle f,H_{\boldsymbol{\alpha}}\rangle=\mathbb{E}_{y\sim\mathcal{N}(0,\mathbb{1})}[H_{\boldsymbol{\alpha}}(y)f(y)]=\mathbb{E}_{y\sim\mathcal{N}(0,\mathbb{1})}[\partial_{\boldsymbol{\alpha}}f(y)] \tag{14}$$

Proof. 14 can be proved by doing induction on k using 11, see [2] for details. □

Corollary A.5. *Let $u_1,u_2\in S^{d-1}$, then the following formula holds:*

$$\mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[h_i(u_1\cdot x)h_j(u_2\cdot x)]=(u_1\cdot u_2)^i i!\delta_{i,j} \tag{15}$$

Proof. It follows from the application of lemma A.4 □

A.2 More on the diffusion model

We model the diffusion process for $t\to x(t)\in\mathbb{R}^d$, with $x(0)=a\sim P_0$, the target distribution, that we want to learn how to sample from. The diffusion process has the following form:

$$\frac{\mathrm{d}x}{\mathrm{d}t}=-x\mathrm{d}t+\mathrm{d}\mathcal{W}_t, \tag{16}$$

in which $\mathcal{W}_t\in\mathbb{R}^d$ is a d dimensional Wiener process. The solution at time t can be written in distribution as:

$$x(t)=x(0)e^{-t}+\sqrt{1-e^{-2t}}z \tag{17}$$

where $z\sim\mathcal{N}(0,\mathbb{1}_d)$. So, defining $\Delta_t=1-e^{-2t}$ The density at time t is given by:

$$P_t(x)=\int\mathrm{d}a\,P_0(a)\frac{1}{(2\pi\Delta_t)^{d/2}}\exp\left(-\frac{1}{2}\frac{(x-ae^{-t})^2}{\Delta_t}\right). \tag{18}$$

To provide intuition on the forward and backward processes described in section 3, we provide here the explicit formulas in the case in which P_0 is the spiked cumulant model from [34]. We can choose the case $x(0)=hv+z$ with $h\sim$ Rademacher$(1/2)$, v is the norm 1 spike and z is a $d-1$-standard

Gaussian, in the space orthogonal to v, i.e. $z \sim \mathcal{N}(0, \mathbb{1}_d - vv^\top)$. Then from eq. (18), P_t has the following expression:

$$\begin{aligned}
P_t(x) &= \frac{1}{(2\pi)^{(d-1)/2}} \exp\left(-\frac{1}{2} x_{\perp v}^\top x_{\perp v}\right) \frac{1}{2\,(2\pi\Delta_t)^{1/2}} \times \\
&\qquad \left[\exp\left(-\frac{1}{2}\frac{(x_v - e^{-t})^2}{\Delta_t}\right) + \exp\left(-\frac{1}{2}\frac{(x_v + e^{-t})^2}{\Delta_t}\right)\right] \\
&= \frac{1}{(2\pi)^{(d-1)/2}} \exp\left(-\frac{1}{2} x_{\perp v}^\top x_{\perp v}\right) \frac{\exp\left(-\frac{1}{2}\frac{x_v^2 + e^{-2t}}{\Delta_t}\right)}{(2\pi\Delta_t)^{1/2}} \cosh\left(\frac{x_v e^{-t}}{\Delta_t}\right),
\end{aligned}$$

were $x = x \cdot vv + (x - x \cdot vv) = x_v v + x_{\perp v}$.

Hence the score is:

$$\begin{aligned}
\mathcal{F}(x,t) &= -x_{\perp v} - \frac{x_v}{\Delta_t} v + \frac{e^{-t}}{\Delta_t} v \tanh\left(\frac{x_v e^{-t}}{\Delta_t}\right) \\
&= -x - \frac{e^{-t}}{\Delta_t} v \left(e^{-t} x_v - \tanh\left(\frac{x_v e^{-t}}{\Delta_t}\right)\right). \qquad (19)
\end{aligned}$$

A.3 Projected SGD dynamics

We will now focus on the dynamics of projected SGD, so we can assume $||w|| = 1$. Denoting for brevity by $x_w := x \cdot w$:

$$\begin{aligned}
-\nabla_{sph} \mathcal{L}_t = -(\mathbb{1} - ww^\top)\mathbb{E}_{x,z} \Bigg[&\sigma'(x_w)\sigma(x_w)x + \sigma^2(x_w)w + \sigma(x_w)x + \\
&+ \sigma'(x_w)x_w x - \frac{1}{\sqrt{\Delta_t}}\left(\sigma(x_w)z + \sigma'(x_w)z_w x\right)\Bigg] \\
= (\mathbb{1} - ww^\top)\mathbb{E}_x &\left[x\left(\sigma''(x_w) - \sigma'(x_w)\sigma(x_w) - \sigma(x_w) - \sigma'(x_w)x_w\right)\right]
\end{aligned}$$

where the second equality all the terms proportional to w have been canceled by the factor $\mathbb{1} - ww^\top$ and the terms depending on z can be reduced to terms involving just x through Stein lemma (as detailed in lemma F.1 in [30]):

$$\begin{aligned}
\mathbb{E}_{z\sim\mathcal{N}(0,\mathbb{1}),x}\left[\frac{1}{\sqrt{\Delta_t}}\sigma(x_w)z\right] &= \mathbb{E}_x\left[\sigma'(x_w)w\right] \\
\mathbb{E}_{z\sim\mathcal{N}(0,\mathbb{1}),x}\left[\frac{1}{\sqrt{\Delta_t}}\sigma'(x_w)z_w x\right] &= \mathbb{E}_x\left[\sigma''(x_w)||w||^2 x + \sigma'(x_w)w\right]
\end{aligned}$$

Then introducing $L(v \cdot x)$ and defining

$$F_\sigma(x_w) := \sigma''(x_w) - \sigma'(x_w)\sigma(x_w) - \sigma(x_w) - \sigma'(x_w)x_w$$

we can expand in Hermite orthonormal basis:

$$\begin{aligned}
L(v \cdot x) &= \sum_{i=0}^{\infty} c_i^L h_i(x_v) \\
F_\sigma(x_w) &= \sum_{j=0}^{\infty} c_j^F h_j(x_w)
\end{aligned}$$

and get:

$$
\begin{aligned}
-\nabla_{sph}\mathcal{L}_t &= (\mathbb{1} - ww^\top) \mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,\mathbb{1})} \left[x \left(\sum_{i=0}^{\infty} c_i^L h_i(x_v) \right) \left(\sum_{j=0}^{\infty} c_j^F h_j(x_w) \right) \right] \\
&= (\mathbb{1} - ww^\top) \left[\left(\sum_{i=1}^{\infty} c_i^L c_{i-1}^F (v\cdot w)^{i-1} \right) v + \left(\sum_{j=1}^{\infty} c_j^F c_{j-1}^L (v\cdot w)^{j-1} \right) w \right] \\
&= \left(\sum_{i=1}^{\infty} c_i^L c_{i-1}^F (v\cdot w)^{i-1} \right) (1 - v\cdot w) v
\end{aligned}
$$

So, let $\alpha := v \cdot w$ in the early stages of learning the dynamics to reach weak recovery are described by:

$$
-\nabla_{sph}\mathcal{L}_t = c_{k^*}^L c_{k*-1}^F \alpha^{k^*-1} v + O(\alpha^{k^*}) \tag{20}
$$

where k^* is the first non zero term of the series. So it is possible to apply the results from [5] to get the following results.

Assumption A.6 (Essential). σ and $\mathbb{P}$ are such that $\nabla_{sph}\mathcal{L}(\alpha)$ is strictly negative for all $\alpha \in (0,1)$

Assumption A.7 (Technical). Let the empirical loss be

$$
\mathscr{L}(\mathcal{D}, w) = \sum_{x\in\mathcal{D}} \mathbb{E}_z \left[\frac{1}{2} || \frac{z}{\sqrt{\Delta_t}} - x - w\sigma(w\cdot x) ||^2 \right] + C
$$

define the martingale term

$$
H_d := \mathscr{L} - \mathcal{L}
$$

We assume that the following estimates hold for some $C_1 > 0$:

$$
\sup_{w\in\mathbb{S}^{d-1}} \mathbb{E}\left[(\nabla_{sph} H_d(w) \cdot v_d)^2 \right] \le C_1 \tag{21}
$$

$$
\sup_{w\in\mathbb{S}^{d-1}} \mathbb{E}\left[\|\nabla_{sph} H_d(w)\|^{4+\varepsilon} \right] \le C_1 d^{(4+\varepsilon)/2} \tag{22}
$$

Proposition A.8 (Positive result). *Assume that $L(x\cdot v)$ is the likelihood ratio of a sub-Gaussian random variable, and σ an activation function such that F_σ satisfies assumption A.6 and assumption A.7.Denote with k^* the information exponent of the loss $\mathcal{L}$ and let $\hat{n}(d,k^*)$ be a sample complexity threshold defined as:*

$$
\begin{cases}
\hat{n}(d,1) = \omega(d) \\
\hat{n}(d,2) = \omega(d\log^2 d) \\
\hat{n}(d,k) = \omega(d^{k-1}\log^2 d) \quad k \ge 3
\end{cases}
$$

then the application of $\hat{n}(d,k^*)$ steps of projected gradient descent with step size η_d satisfying

$$
\frac{1}{\hat{n}} \ll \eta_d \ll \frac{1}{\sqrt{\hat{n}d}} \tag{23}
$$

starting from isotropic initialization $w \sim$ Unif$(\mathbb{S}^{d-1})$ leads to:

$$
\lim_{d\to\infty} |v \cdot w(\hat{n}(d,k^*))| = 1. \tag{24}
$$

Where the limit holds in probability and in L^p for all $p \ge 1$.

Proposition A.9 (Negative result). *In the setting of the previous propositions, if $n(d) = o(\hat{n}(d,k^*)$ and*

$$
\eta_d = \begin{cases}
O\left(\frac{1}{d}\right) & k^* = 1,2 \\
\eta_d = O\left(\frac{1}{\sqrt{n(d)d}}\right) & k^* \ge 3
\end{cases}
$$

the online SGD with learning rate η_d will fail reach weak recovery:

$$
\lim_{d\to\infty} \sup_{\tau\le n(d)} |v\cdot w(t)| = 0 \qquad \textit{in probability and in } L^p \textit{ for any } p \ge 1 \tag{25}
$$

Proof of proposition A.8 and proposition A.9. These propositions can be seen as corollaries of theorems 1.3 and 1.4 from [5]. Assumptions A.6 and A.7 verify the core requirements. The only catch is that all the assumptions are not verified on the loss function, but directly on the spherical gradient. However tha whole reasoning detailed in [5] never relies on computations of the loss function, but only of its gradient, hence we can apply the proof to our setting.

□

In the following we consider some examples of applications of proposition A.8.

A.3.1 Spiked Wishart

In the case of spiked Wishart model, $k^* = 2$, so we can take $\sigma = -id$:

$$F(x_w) = x_w$$

and we get that the projected SGD reaches weak recovery in $d \log d$ sample complexity

A.3.2 Spiked cumulant

We choose σ to match equation 19:

$$\sigma(x_v) = \frac{e^{-t}}{\Delta_t}\left(e^{-t}x_v - \tanh\left(\frac{e^{-}t}{\Delta_t}x_v\right)\right)$$

and simulations confirm that projected SGD works in this regime reaching weak recovery in d^3 samples. Note that the coefficients depend exponentially on diffusion time $c_4^L = e^{-4t} - 3e^{-2t}$

A.4 Mixed cumulant model

Assumption A.10. The link function σ is a thrice differentiable function, with bounded first, second and third order derivatives. Hence $F(z) = \sigma''(z) - \sigma'(z)\sigma - \sigma(z) - \sigma'(z)z$ is a globally Lipschitz function, hence F belongs to the space of square integrable function with respect to the density of $\mathcal{N}(0, \mathbb{1})$ for all t (being all sub-Gaussian distributions). Assume moreover that σ is so that F satisfies the following conditions

$$c_1^F = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[F(z)h_1(z)] > 0 \tag{26}$$

$$c_3^F = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[F(z)h_3(z)] < 0 \tag{27}$$

Proof of proposition 3.1. The proof relies on verifying the precise hypothesis of propositions 3 and 4 in [3], so that all the argument can be replicated in the exact same way. The starting point is that the population loss expansion

$$-\nabla_{sph}\mathcal{L}(\alpha_u, \alpha_v) = \sum_{k=1}^{\infty}\sum_{i=0}^{k} c_{k-1}^F c_{i,k-i}^L \left(\alpha_u^{i-1}\alpha_v^{k-i}u + \alpha_u^i\alpha_v^{k-i-1}v\right) \tag{28}$$

coincides with the population loss from [3], eq. (25), with a different naming of the coefficients: in their notation kc_k^σ corresponds to c_{k-1}^F in our notation. Hence assumption A.10 verifies the requirements of Assumption 1 in [3]. The only term that in principle could behave differently is the directional noise martingale $H_d(x, w) := \mathscr{L} - \mathcal{L}$. However L_t is the likelihood ratio of a sub-Gaussian random variable, and H is a Lipshitz transformation, so $H(x, w)$, with $||w|| = 1$ and $x \sim \mathbb{P}_t$ is sub-Gaussian. Hence requirements in assumption A.7, which were the same as the ones needed in [3], are satisfied. Hence, we can apply propositions 3-4 from [3] and conclude the proof. □

B Sequential learning in CelebA data

We here report the outcome of the sequential learning experiment on $1.2 \cdot 10^5$ CelebA data [19], which we downscale to 80×80 greyscale pixels. We observe the same sequential learning behavior as for the CIFAR-10 data. However, we also observe large jumps in the learning trajectories, which persist over different random initializations and optimizers (Adam vs. AdamW). The second large jump at around 10^3 steps occurs at the start of the second epoch of training.

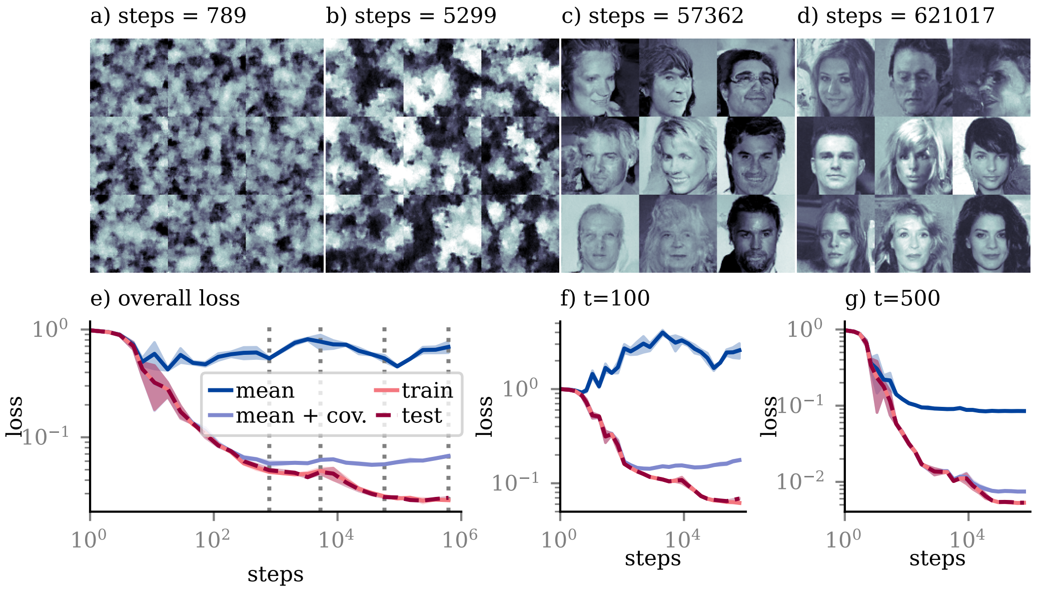

Figure 2: Training Unets on CelebA data. a)-d) samples generated from the model at various training stages. e) Test loss of the model on clones of the dataset during training. Vertical dotted lines mark training stages of images generated from the model shown in panels a)-d). f)-g) same as e), but for fixed level of noise. All curves are averaged over 3 initializations of the network models and $5 \cdot 10^3$ test data. Shaded areas report standard deviation over random initialization.

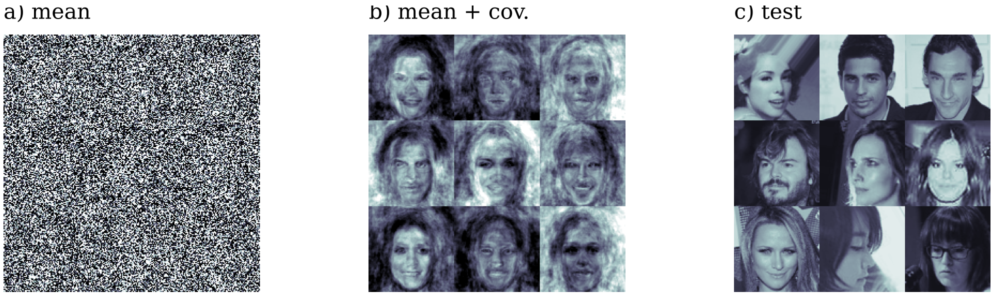

Figure 3: Samples from the different "clones" as well as the test data set. a) shows images drawn from the mean clone which follows a Gaussian distribution with matching mean to the CelebA dataset and identity covariance. In b), we additionally match the covariance matrix of the Gaussian distribution to the CelebA dataset. c) shows 9 images from the CelebA dataset.

B.1 The clones

To generate the clone datasets, we first determine the mean μ and the covariance Σ of the test datasets for both CelebA and CIFAR-10. We then sample the "mean" clone from a Gaussian distribuution with mean μ and identity covariance. We then sample the "mean + cov." clone from a Gaussian distribution with matching mean and covariance. We show examples of these datasets in fig. 3 and fig. 4.

B.2 Training hyperparameters

We use diffusion models with a Unet architecture [26], $T = 10^3$ levels of noise and sinusoidal embedding for t. For the CIFAR-10 data reported in 1, we use the Adam optimizer with learning rate 10^-3. For the CelebA data reported in 2, we use the AdamW optimizer with the same baseline learning rate and an additional cosine learning rate schedule. For both data sets we use a batch-size of 10^2 samples.

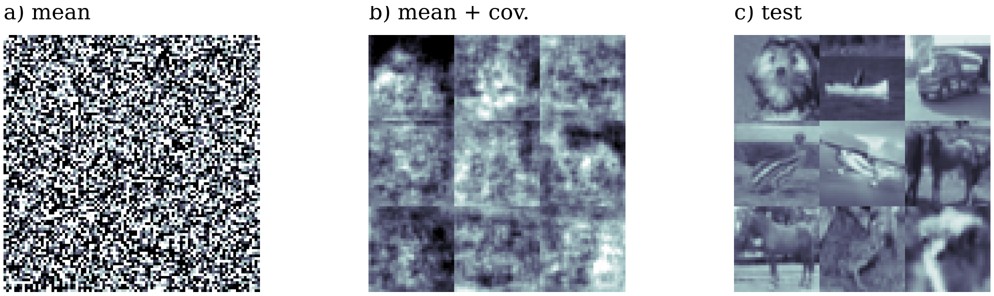

Figure 4: Samples from the different "clones" as well as the test data set. a) shows images drawn from the mean clone which follows a Gaussian distribution with matching mean to the CIFAR-10 dataset and identity covariance. In b), we additionally match the covariance matrix of the Gaussian distribution to the CIFAR-10 dataset. c) shows 9 images from the CIFAR-10 dataset.

C Learning the mixed cumulant model

In fig. 1, we show learning curves of a neural network trained on a multi-spiked data model,

$$x^\mu = \sqrt{\beta_m} m + \sqrt{\beta_u}\lambda^\mu u + \sqrt{\beta_v}\nu^\mu v + z^\mu \, ,$$

where $z^\mu, \nu^\mu, \lambda^\mu$ follow the same distribution as in eq. (6) but we also add a "mean spike" $m \in \mathbb{R}^d$ of length 1. We choose and $d = 100, \beta_u = 10, \beta_v = 1$ and all the spikes m, u, v to have length one and be orthogonal to each other. Our network model has a matched architecture

$$S_t^W = x + w_m + w_u \odot x + \frac{e^{-t}}{\Delta_t} w_v \left(e^{-t} w_v^{\mathrm{T}} x - \tanh\left(\frac{w_v^{\mathrm{T}} x e^{-t}}{\Delta_t} \right) \right) \, ,$$

where $\odot$ is the Hadamard product. We initialize the all the weights w_u, w_m, w_v uniform on the unit sphere in $\mathbb{R}^d$, but set $w_v^{\mathrm{T}} v = 0$ to ensure that the cumulant spike is found only through the training dynamics. We then train the model using stochastic gradient descent, using 10 noised versions of one sample x^μ per training step and a learning rate $\eta = 10^{-3}$. We additionally fix the norm of w_v to one during training, hence we train this weight (but not w_u, w_m) with projected gradient descent.