Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training

Tony Bonnaire[†]
LPENS
Université PSL, Paris
tony.bonnaire@phys.ens.fr

Raphaël Urfin[†]
LPENS
Université PSL, Paris
raphael.urfin@phys.ens.fr

Giulio Biroli
LPENS
Université PSL, Paris
giulio.biroli@phys.ens.fr

Marc Mézard
Department of Computing Sciences
Bocconi University, Milano
marc.mezard@unibocconi.it

Abstract

Diffusion models have achieved remarkable success across a wide range of generative tasks. A key challenge is understanding the mechanisms that prevent their memorization of training data and allow generalization. In this work, we investigate the role of the training dynamics in the transition from generalization to memorization. Through extensive experiments and theoretical analysis, we identify two distinct timescales: an early time τ_{gen} at which models begin to generate high-quality samples, and a later time τ_{mem} beyond which memorization emerges. Crucially, we find that τ_{mem} increases linearly with the training set size n, while τ_{gen} remains constant. This creates a growing window of training times with n where models generalize effectively, despite showing strong memorization if training continues beyond it. It is only when n becomes larger than a model-dependent threshold that overfitting disappears at infinite training times. These findings reveal a form of implicit dynamical regularization in the training dynamics, which allow to avoid memorization even in highly overparameterized settings. Our results are supported by numerical experiments with standard U-Net architectures on realistic and synthetic datasets, and by a theoretical analysis using a tractable random features model studied in the high-dimensional limit.

1 Introduction

Diffusion Models [DMs, 44, 18, 49, 50] achieve state-of-the-art performance in a wide variety of AI tasks such as the generation of images [41], audios [58], videos [30], and scientific data [28, 36]. This class of generative models, inspired by out-of-equilibrium thermodynamics [44], corresponds to a two-stage process: the first one, called *forward*, gradually adds noise to a data, whereas the second one, called *backward*, generates new data by denoising Gaussian white noise samples. In DMs, the reverse process typically involves solving a stochastic differential equation (SDE) with a force field called *score*. However, it is also possible to define a deterministic transport through an ordinary differential equation (ODE), treating the score as a velocity field, an approach that is for instance followed in flow matching [29].

Understanding the generalization properties of score-based generative methods is a central issue in machine learning, and a particularly important question is how memorization of the training set

[†]Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

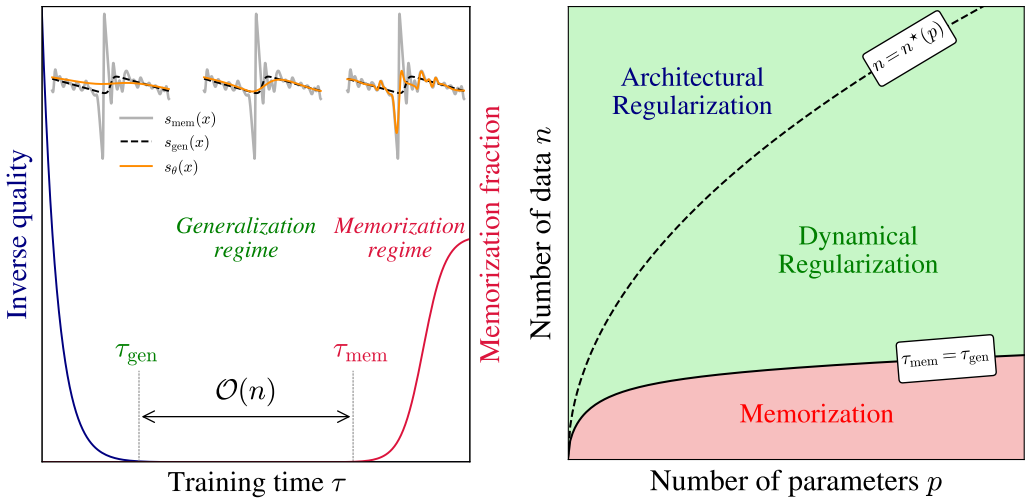


Figure 1: **Qualitative summary of our contributions.** *(Left)* Illustration of the training dynamics of a diffusion model. Depending on the training time τ, we identify three regimes measured by the inverse quality of the generated samples (blue curve) and their memorization fraction (red curve). The generalization regime extends over a large window of training times which increases with the training set size n. On top, we show a one dimensional example of the learned score function during training (orange). The gray line gives the exact empirical score, at a given noise level, while the black dashed line corresponds to the true (population) score. *(Right)* Phase diagram in the (n, p) plane illustrating three regimes of diffusion models: Memorization when n is sufficiently small at fixed p, Architectural Regularization for $n > n^\star(p)$ (which is model and dataset dependent, as discussed in [13, 23]), and Dynamical Regularization, corresponding to a large intermediate generalization regime obtained when the training dynamics is stopped early, i.e. $\tau \in [\tau_{\rm gen}, \tau_{\rm mem}]$.

is avoided in practice. A model without regularization achieving zero training loss only learns the empirical score, and is bound to reproduce samples of the training dataset at the end of the backward process. This memorization regime [27, 5, 4] is empirically observed when the training set is small and disappears when it increases beyond a model-dependent threshold [22]. Understanding the mechanisms controlling this change of regimes from memorization to generalization is a central challenge for both theory and applications. Model regularization and inductive biases imposed by the network architecture were shown to play a role [23, 43], as well as a dynamical regularization due to the finiteness of the learning rate [56]. However, the regime shift described above is consistently observed even in models where all these regularization mechanisms are present. This suggests that the core mechanism behind the transition from memorization to generalization lies elsewhere. In this work, we demonstrate – first through numerical experiments, and then via the theoretical analysis of a simplified model – that this transition is driven by an implicit dynamical bias towards generalizing solutions emerging in the training, which allows to avoid the memorization phase.

Contributions and theoretical picture. We investigate the dynamics of score learning using gradient descent, both numerically and analytically, and study the generation properties of the score depending on the time τ at which the training is stopped. The theoretical picture built from our results and combining several findings from the recent literature is illustrated in Fig. 1. The two main parameters are the size of the training set n and the expressivity of the class of score functions on which one trains the model, characterized by a number of parameters p; when both n and p are large one can identify three main regimes. Given p, if n is larger than $n^*(p)$ (which depends on the training set and on the class of scores), the score model is not expressive enough to represent the empirical score associated to n data, and instead provides a smooth interpolation, approximately independent of the training set. In this regime, even with a very large training time $\tau \to \infty$, memorization does not occur because the model is regularized by its architecture and the finite number of parameters. When $n < n^*(p)$ the model is expressive enough to memorize, and two timescales emerge during training: one, $\tau_{\rm gen}$, is the minimum training time required to achieve high-quality data generation; the second, $\tau_{\rm mem} > \tau_{\rm gen}$, signals when further training induces memorization, and causes the model to

increasingly reproduce the training samples (left panel). The first timescale, $\tau_{\rm gen}$, is found independent of n, whereas the second, $\tau_{\rm mem}$, grows approximately linearly with n, thus opening a large window of training times during which the model generalizes if early stopped when $\tau \in [\tau_{\rm gen}, \tau_{\rm mem}]$. Our results shows that implicit dynamical regularization in training plays a crucial role in score-based generative models, substantially enlarging the generalization regime (see right panel of Fig. 1), and hence allowing to avoid memorization even in highly overparameterized settings. We find that the key mechanism behind the widening gap between $\tau_{\rm gen}$ and $\tau_{\rm mem}$ is the irregularity of the empirical score at low noise level and large n. In this regime the models used to approximate the score provide a smooth interpolation that remains stable for a long period of training times and closely approximates the population score, a behavior likely rooted in the spectral bias of neural networks [38]. Only at very long training times do the dynamics converge to the low lying minimum corresponding to the empirical score, leading to memorization (as illustrated in the 1D examples in the left panel of Fig. 1).

The theoretical picture described above is based on our numerical and analytical results, and builds up on previous works, in particular numerical analysis characterizing the memorization–generalization transition [16, 57], analytical works on memorization of DMs [13, 23, 22], and studies on the spectral bias of deep neural networks [38]. Our numerical experiments[†] use a class of scores based on a realistic U-Net [42] trained on downscaled images of the CelebA dataset [31]. By varying n and p, we measure the evolution of the sample quality (through FID) and the fraction of memorization during learning, which support the theoretical scenario presented in Fig. 1. Additional experimental results on synthetic data are provided in Supplemental Material (SM, Sects. A and B). On the analytical side, we focus on a class of scores constructed from random features and simplified models of data, following [13]. In this setting, the timescales of training dynamics correspond directly to the inverse eigenvalues of the random feature correlation matrix. Leveraging tools from random matrix theory, we compute the spectrum in the limit of large datasets, high-dimensional data, and overparameterized models. This analysis reveals, in a fully tractable way, how the theoretical picture of Fig. 1 emerges within the random feature framework.

Related works.

- The memorization transition in DMs has been the subject of several recent empirical investigations [8, 45, 46] which have demonstrated that state-of-the-art image DMs – including Stable Diffusion and DALL·E – can reproduce a non-negligible portion of their training data, indicating a form of memorization. Several additional works [16, 57] examined how this phenomenon is influenced by factors such as data distribution, model configuration, and training procedure, and provide a strong basis for the numerical part of our work.
- A series of theoretical studies in the high-dimensional regime have analyzed the memorization–generalization transition during the generative dynamics under the empirical score assumption [5, 1, 52], showing how trajectories are attracted to the training samples. Within this high-dimensional framework, [9, 10, 55, 13] study the score learning for various model classes. In particular, [13] uses a Random Feature Neural Network [39]. The authors compute the asymptotic training and test losses for $\tau \to \infty$ and relate it to memorization. The theoretical part of our work generalizes this approach to study the role of training dynamics and early stopping in the memorization–generalization transition.
- Recent works have also uncovered complementary sources of implicit regularization explaining how DMs avoid memorization. Architectural biases and limited network capacity were for instance shown to constrain memorization in [23, 22, 4], and finiteness of the learning rate prevents the model from learning the empirical score in [56]. Also related to our analysis, [26, 4] show the beneficial role of early stopping the training dynamics to enhance the generalization.
- Finally, previous studies on supervised learning [38, 59], and more recently on DMs [54], have shown that deep neural networks display a frequency-dependent learning speed, and hence a learning bias towards low frequency functions. This fact plays an important role in the results we present since the empirical score contains a low frequency part that is close to the population score, and a high-frequency part that is dataset-dependent. To the best of our knowledge, the training time to learn the high-frequency part and hence memorize, that we find to scale with n, has not been studied from this perspective in the context of score-based generative methods.

[†]Code available at github.com/tbonnair/Why-Diffusion-Models-Don-t-Memorize.

Setting: generative diffusion and score learning. Standard DMs define a transport from a target distribution P_0 in $\mathbb{R}^d$ to a Gaussian white noise $\mathcal{N}(0, \boldsymbol{I}_d)$ through a *forward process* defined as an Ornstein-Uhlenbeck (OU) stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x} = -\mathbf{x}(t)\mathrm{d}t + \mathrm{d}\mathbf{B}(t), \tag{1}$$

where $\mathrm{d}\mathbf{B}(t)$ is square root of two times a Wiener process. Generation is performed by time-reversing the SDE (1) using the score function $\mathbf{s}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log P_t(\mathbf{x})$,

$$-\mathrm{d}\mathbf{x} = \left[\mathbf{x}(t) + 2\mathbf{s}(\mathbf{x}, t)\right] \mathrm{d}t + \mathrm{d}\mathbf{B}(t), \tag{2}$$

where $P_t(\mathbf{x})$ is the probability density at time t along the forward process, and the noise $\mathrm{d}\mathbf{B}(t)$ is also the square root of two times a Wiener process. As shown in the seminal works [21, 53], $\mathbf{s}(\mathbf{x}, t)$ can be obtained by minimizing the score matching loss

$$\hat{\mathbf{s}}(\mathbf{x}, t) = \arg\min_{\mathbf{s}} \mathbb{E}_{\mathbf{x}\sim P_0, \boldsymbol{\xi}\sim\mathcal{N}(0,\boldsymbol{I}_d)} \left[\| \sqrt{\Delta_t}\mathbf{s}(\mathbf{x}(t), t) + \boldsymbol{\xi} \|^2 \right], \tag{3}$$

where $\Delta_t = 1 - e^{-2t}$. In practice, the optimization problem is restricted to a parametrized class of functions $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$ defined, for example, by a neural network with parameters $\boldsymbol{\theta}$. The expectation over $\mathbf{x}$ is replaced by the empirical average over the training set (n iid samples $\mathbf{x}^\nu$ drawn from P_0),

$$\mathcal{L}_t(\boldsymbol{\theta}, \{\mathbf{x}^\nu\}_{\nu=1}^n) = \frac{1}{n}\sum_{\nu=1}^{n} \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\boldsymbol{I}_d)} \left[\| \sqrt{\Delta_t}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^\nu(t)) + \boldsymbol{\xi} \|^2 \right], \tag{4}$$

where $\mathbf{x}_t^\nu(\boldsymbol{\xi}) = e^{-t}\mathbf{x}^\nu + \sqrt{\Delta_t}\boldsymbol{\xi}$. The loss in (4) can be minimized with standard optimizers, such as stochastic gradient descent [SGD, 40] or Adam [25]. In practice, a single model conditioned on the diffusion time t is trained by integrating (4) over time [24]. The solution of the minimization of (4) is the so-called empirical score (e.g. [5, 27]), defined as $\mathbf{s}_{\mathrm{emp}}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log P_t^{\mathrm{emp}}(\mathbf{x})$, with

$$P_t^{\mathrm{emp}}(\mathbf{x}) = \frac{1}{n\,(2\pi\Delta_t)^{d/2}} \sum_{\nu=1}^{n} e^{-\frac{1}{2\Delta_t}\|\mathbf{x}-\mathbf{x}^\nu e^{-t}\|_2^2}. \tag{5}$$

This solution is known to inevitably recreate samples of the training set at the end of the generative process (i.e., it perfectly memorizes), unless n grows exponentially with the dimension d [5]. However, this is not the case in many practical applications where memorization is only observed for relatively small values of n, and disappears well before n becomes exponentially large in d. The empirical minimization performed in practice, within a given class of models and a given minimization procedure, does *not* drive the optimization to the global minimum of (4), but instead to a smoother estimate of the score that is independent of the training set with good generalization properties [22], as the global minimum of (3) would do. Understanding how it is possible, and in particular the role played by the training dynamics to avoid memorization, is the central aim of the present work.

2 Generalization and memorization during training of diffusion models

Data & architecture. We conduct our experiments on the CelebA face dataset [31], which we convert to grayscale downsampled images of size $d = 32 \times 32$, and vary the training set size n from 128 up to 32768. Our score model has a U-Net architecture [42] with three resolution levels and a base channel width of W with multipliers 1, 2 and 3 respectively. All our networks are DDPMs [18] trained to predict the injected noise at diffusion time t using SGD with momentum at fixed batch size $\min(n, 512)$. The models are all conditioned on t, i.e. a single model approximates the score at all times, and make use of a standard sinusoidal position embedding [51] that is added to the features of each resolution. More details about the numerical setup can be found in SM (Sect. A).

Evaluation metrics. To study the transition from generalization to memorization during training, we monitor the loss (4) during training using a fixed diffusion time $t = 0.01$. At various numbers of SGD updates τ, we compute the loss on all n training examples (training loss) and on a held-out test set of 2048 images (test loss). To characterize the score obtained after a training time τ, we assess the originality and quality of samples by generating 10K samples using a DDIM accelerated sampling [47]. We compute (i) the Fréchet-Inception Distance [FID, 17] against 10K test samples which we use to identify the generalization time τ_{gen}; and (ii) the fraction of memorized generated

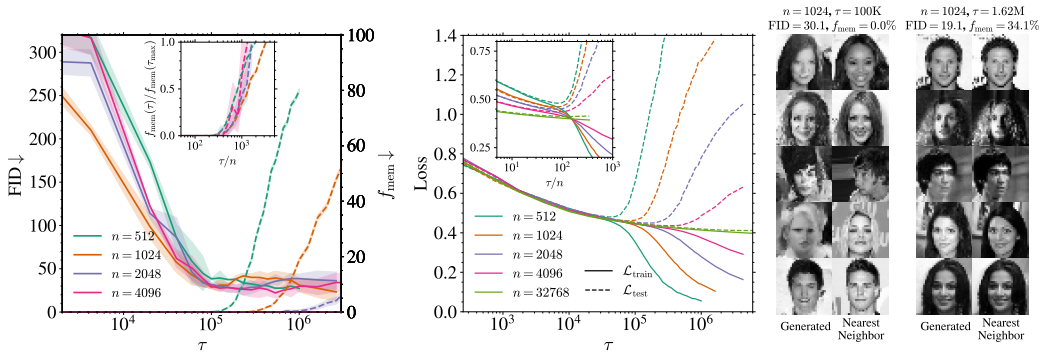


Figure 2: **Memorization transition as a function of the training set size n for U-Net score models on CelebA.** *(Left)* FID (solid lines, left axis) and memorization fraction f_{mem} (dashed lines, right axis) against training time τ for various n. Inset: normalized memorization fraction $f_{\text{mem}}(\tau)/f_{\text{mem}}(\tau_{\max})$ with the rescaled time τ/n. *(Middle)* Training (solid lines) and test (dashed lines) loss with τ for several n at fixed $t = 0.01$. Inset: both losses plotted against τ/n. Error bars on the losses are imperceptible. *(Right)* Generated samples from the model trained with $n = 1024$ for $\tau = 100$K or $\tau = 1.62$M steps, along with their nearest neighbors in the training set.

samples $f_{\text{mem}}(\tau)$ granting access to τ_{mem}, the memorization time. Following previous numerical studies [57, 16], a generated sample $\mathbf{x}_\tau$ is considered memorized if

$$\mathbb{E}_{\mathbf{x}_\tau}\left[\frac{\|\mathbf{x}_\tau - \mathbf{a}^{\mu_1}\|_2}{\|\mathbf{x}_\tau - \mathbf{a}^{\mu_2}\|_2}\right] < k, \tag{6}$$

where $\mathbf{a}^{\mu_1}$ and $\mathbf{a}^{\mu_2}$ are the nearest and second nearest neighbors of $\mathbf{x}_\tau$ in the training set in the L_2 sense. In what follows, we choose to work with $k = 1/3$ [57, 16], but we checked that varying k to $1/2$ or $1/4$ does not impact the claims about the scaling. Error bars in the figures correspond to twice the standard deviation over 5 different test sets for FIDs, and 5 noise realizations for $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{test}}$. For f_{mem}, we report the 95% CIs on the mean evaluated with 1,000 bootstrap samples.

Role of training set size on the learning dynamics. At fixed model capacity ($p = 4 \times 10^6$, base width $W = 32$), we investigate how the training set size n impacts the previous metrics. In the left panel of Fig. 2, we first report the FID (solid lines) and $f_{\text{mem}}(\tau)$ (dashed lines) for various n. All trainings dynamics exhibit two phases. First, the FID quickly decreases to reach a minimum value on a timescale τ_{gen} (≈ 100K) that does not depend on n. In the right panel, the generated samples at $\tau = 100$K clearly differ from their nearest neighbors in the training set, indicating that the model generalizes correctly. Beyond this time, the FID remains flat. $f_{\text{mem}}(\tau)$ is zero until a later time τ_{mem} after which it increases, clearly signaling the entrance into a memorization regime, as illustrated by the generated samples in the right-most panel of Fig. 2, very close to their nearest neighbors. Both the transition time τ_{mem} and the value of the final fraction $f_{\text{mem}}(\tau_{\max})$ (with $\tau_{\max}$ being one to four million SGD steps) vary with n. The inset plot shows the normalized memorization fraction $f_{\text{mem}}(\tau)/f_{\text{mem}}(\tau_{\max})$ against the rescaled time τ/n, making all curves collapse and increase at around $\tau/n \approx 300$, showing that $\tau_{\text{mem}} \propto n$, and demonstrating the existence of a generalization window for $\tau \in [\tau_{\text{gen}}, \tau_{\text{mem}}]$ that widens linearly with n, as illustrated in the left panel of Fig. 1.

As highlighted in the introduction, memorization in DMs is ultimately driven by the overfitting of the empirical score $\mathbf{s}_{\text{mem}}(\mathbf{x}, t)$. The evolution of $\mathcal{L}_{\text{train}}(\tau)$ and $\mathcal{L}_{\text{test}}(\tau)$ at fixed $t = 0.01$ are shown in the middle panel of Fig. 2 for n ranging from 512 to 32768. Initially, the two losses remain nearly indistinguishable, indicating that the learned score $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$ does not depend on the training set. Beyond a critical time, $\mathcal{L}_{\text{train}}$ continues to decrease while $\mathcal{L}_{\text{test}}$ increases, leading to a nonzero generalization loss whose magnitude depends on n. As n increases, this critical time also increases and, eventually, the training and test loss gap shrinks: for $n = 32768$, the test loss remains close to the training loss, even after 11 million SGD steps. The inset shows the evolution of both losses with τ/n, demonstrating that the overfitting time scales linearly with the training set size n, just like τ_{mem} identified in the left panel. Moreover, there is a consistent lag between the overfitting time and τ_{mem} at fixed n, reflecting the additional training required for the model to overfit the empirical score sufficiently to reproduce the training samples, and therefore to impact the memorization fraction.

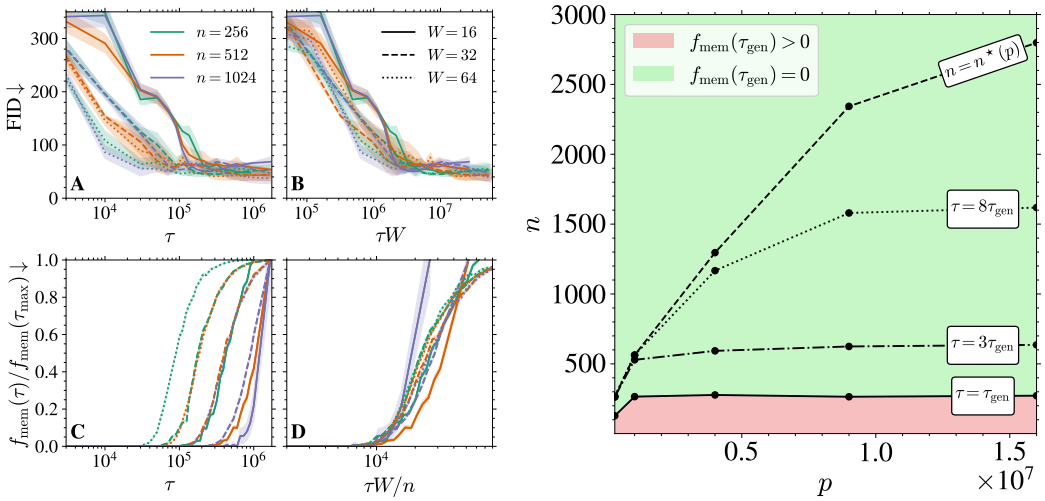


Figure 3: **Effect of the number of parameters in the U-Net architecture on the timescales of the training dynamics.** *(Left)* FID (panels **A**, **B**) and normalized memorization fraction $f_{\text{mem}}(\tau)/f_{\text{mem}}(\tau_{\text{max}})$ (panels **C**, **D**) for various n and W during training. In panels **B** and **D**, time is rescaled such that all curves collapse. *(Right)* (n, p) phase diagram of generalization vs memorization for U-Nets trained on CelebA. Curves show, for $\tau \in \{\tau_{\text{gen}}, 3\tau_{\text{gen}}, 8\tau_{\text{gen}}\}$, the minimal dataset size $n(p)$ satisfying $f_{\text{mem}}(\tau) = 0$. The shaded background indicates the memorization–generalization boundary for $\tau = \tau_{\text{gen}}$.

Memorization is *not* due to data repetition. We must stress that this delayed memorization with n is *not* due to the mere repetition of training samples, as a first intuition could suggest. In SM Sects. A and B, we show that full-batch updates still yield $\tau_{\text{mem}} \propto n$. In other words, even if at fixed τ all models have processed each sample equally often, larger n consistently postpone memorization. This confirms that memorization in DMs is driven by a fundamental n-dependent change in the loss landscape – not by a sample repetition during training.

Effect of the model capacity. To study more precisely the role of the model capacity on the memorization–generalization transition, we vary the number of parameters p by changing the U-Nets base width $W \in \{8, 16, 32, 48, 64\}$, resulting in a total of $p \in \{0.26, 1, 4, 9, 16\} \times 10^6$ parameters. In the left panel of Fig. 3, we plot both the FID (top row) and the normalized memorization fraction (bottom row) as functions of τ for several width W and training set sizes n. Panels **A** and **C** demonstrate that higher-capacity networks (larger W) achieve high-quality generation and begin to memorize *earlier* than smaller ones. Panels **B** and **D** show that the two characteristic timescales simply scale as $\tau_{\text{gen}} \propto W^{-1}$ and $\tau_{\text{mem}} \propto nW^{-1}$. In particular, this implies that, for $W > 8$, the critical training set size $n_{\text{gm}}(p)$ at which $\tau_{\text{mem}} = \tau_{\text{gen}}$ is approximately independent of p (at least on the limited values of p we focused on).When $n > n_{\text{gm}}(p)$, the interval $[\tau_{\text{gen}}, \tau_{\text{mem}}]$ opens up, so that early stopping within this window yields high quality samples without memorization. In the right panel of Fig. 3, we display this boundary (solid line) in the (n, p) plane by fixing the training time to $\tau = \tau_{\text{gen}}$, that we identify numerically using the collapse of all FIDs at around $W\tau_{\text{gen}} \approx 3 \times 10^6$ (see panel **B**), and computing the smallest n such that $f_{\text{mem}}(\tau) = 0$. The resulting solid curve delineates two regimes: below the curve, memorization already starts at τ_{gen}; above the curve, the models generalize perfectly under early stopping. We repeat this experiment for $\tau = 3\tau_{\text{gen}}$ and $\tau = 8\tau_{\text{gen}}$, showing saturation to larger and larger p as τ increases. Eventually, for $\tau \to \infty$, we expect these successive boundaries to converge to the architectural regularization threshold $n^\star(p)$, i.e. the point beyond which the network avoids memorization because it is not expressive enough, as found in [13] and highlighted in the right panel of Fig. 1. In order to estimate $n^\star(p)$, we measure for a given τ the largest $n(\tau)$ yielding $f_{\text{mem}} \approx 0$. The curve $n(\tau)$ approaches $n^\star(p)$ for large τ. We therefore estimate $n^\star(p)$ by measuring the asymptotic values of $n(\tau)$, which in practice is reached already at $\tau = \tau_{\text{max}} = 2$M updates for the values of W we focus on.

3 Training dynamics of a Random Features Network

Notations. We use bold symbols for vectors and matrices. The L^2 norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\| = (\sum_i \mathbf{x}_i^2)^{1/2}$. We write $f = \mathcal{O}(g)$ to mean that in the limit $n, p \to \infty$, there exists a constant C such that $|f| \leq C|g|$.

Setting. We study analytically a model introduced in [13], where the data lie in d dimensions. We parametrize the score with a Random Features Neural Network [RFNN, 39]

$$\mathbf{s}_{\mathbf{A}}(\mathbf{x}) = \frac{\mathbf{A}}{\sqrt{p}} \sigma\left(\frac{\mathbf{W}\mathbf{x}}{\sqrt{d}}\right). \tag{7}$$

An RFNN, illustrated in Fig. 4 (left), is a two-layer neural-network whose first layer weights ($\mathbf{W} \in \mathbb{R}^{p\times d}$) are drawn from a Gaussian distribution and remain frozen while the second layer weights ($\mathbf{A} \in \mathbb{R}^{d\times p}$) are learned during training. This model has already served as theoretical framework for studying several behaviors of deep neural network such as the double descent phenomenon [32, 11]. σ is an element-wise non-linear activation function. We consider a training set of n iid samples $\mathbf{x}^\nu \sim P_{\mathbf{x}}$ for $\nu = 1, \dots, n$ and we focus on the high-dimensional limit $d, p, n \to \infty$ with the ratios $\psi_p = p/d, \psi_n = n/d$ kept fixed. We study the training dynamics associated to the minimization of the empirical score matching loss defined in (4) at a fixed diffusion time t. This is a simplification compared to practical methods, which use a single model for all t. It has been already studied in previous theoretical works [9, 13]. The loss (4) is rescaled by a factor $1/d$ in order to ensure a finite limit at large d. We also study the evolution of the test loss evaluated on test points and the distance to the exact score $\mathbf{s}(\mathbf{x}) = \nabla \log P_{\mathbf{x}}$,

$$\mathcal{L}_{\text{test}} = \frac{1}{d}\mathbb{E}_{\mathbf{x},\boldsymbol{\xi}}\left[\|\sqrt{\Delta_t}\mathbf{s}_{\mathbf{A}}(\mathbf{x}_t(\boldsymbol{\xi})) + \boldsymbol{\xi}\|^2\right], \quad \mathcal{E}_{\text{score}} = \frac{1}{d}\mathbb{E}_{\mathbf{x}}\left[\|\mathbf{s}_{\mathbf{A}}(\mathbf{x}) - \nabla \log P_{\mathbf{x}}\|^2\right], \tag{8}$$

where the expectations $\mathbb{E}_{\mathbf{x},\boldsymbol{\xi}}$ are computed over $\mathbf{x} \sim P_{\mathbf{x}}$ and $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{I}_d)$. The generalization loss, defined as $\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{test}} - \mathcal{L}_{\text{train}}$, indicates the degree of overfitting in the model while the distance to the exact score $\mathcal{E}_{\text{score}}$ measures the quality of the generation as it is an upper bound on the Kullback–Leibler divergence between the target and generated distributions [48, 7]. The weights $\mathbf{A}$ are updated via gradient descent

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} - \eta \nabla_{\mathbf{A}} \mathcal{L}_{\text{train}}(\mathbf{A}^{(k)}), \tag{9}$$

where η is the learning rate. In the high-dimensional limit, as the learning rate $\eta \to 0$, and after rescaling time as $\tau = k\eta/d^2$, the discrete-time dynamics converges to the following continuous-time gradient flow:

$$\dot{\mathbf{A}}(\tau) = -d^2 \nabla_{\mathbf{A}} \mathcal{L}_{\text{train}}(\mathbf{A}(\tau)) = -2\Delta_t \frac{d}{p}\mathbf{A}\mathbf{U} - \frac{2d\sqrt{\Delta_t}}{\sqrt{p}}\mathbf{V}^T, \tag{10}$$

with

$$\mathbf{U} = \frac{1}{n}\sum_{\nu=1}^{n} \mathbb{E}_{\boldsymbol{\xi}}\left[\sigma\left(\frac{\mathbf{W}\mathbf{x}_t^\nu(\boldsymbol{\xi})}{\sqrt{d}}\right)\sigma\left(\frac{\mathbf{W}\mathbf{x}_t^\nu(\boldsymbol{\xi})}{\sqrt{d}}\right)^T\right], \quad \mathbf{V} = \frac{1}{n}\sum_{\nu=1}^{n} \mathbb{E}_{\boldsymbol{\xi}}\left[\sigma\left(\frac{\mathbf{W}\mathbf{x}_t^\nu(\boldsymbol{\xi})}{\sqrt{d}}\right)\boldsymbol{\xi}^T\right]. \tag{11}$$

Assumptions. For our analytical results to hold, we make the following mathematical assumptions which are standard when studying Random Features [37, 15, 20] namely (i) the activation function σ admits a Hermite polynomial expansion $\sigma(x) = \sum_{s=0}^{\infty} \frac{\alpha_s}{s!} He_s(x)$; and (ii) the data distribution $P_{\mathbf{x}}$ has zero mean, sub-Gaussian tails and a covariance $\boldsymbol{\Sigma} = \mathbb{E}_{P_{\mathbf{x}}}[\mathbf{x}\mathbf{x}^T]$ with bounded spectrum. We assume that the empirical distribution of eigenvalues of $\boldsymbol{\Sigma}$ converges weakly in the high dimensional limit to a deterministic density $\rho_{\boldsymbol{\Sigma}}(\lambda)$ and that $\text{Tr}(\boldsymbol{\Sigma})/d$ converges to a finite limit (for a more precise mathematical statement see SM Sect. C.3). Moreover, we make additional assumptions that are not essential to the proofs but which simplify the analysis: (iii) the activation function σ verifies $\mu_0 = \mathbb{E}_z[\sigma(z)] = 0$ for z standard Gaussian; and (iv) the second layer $\mathbf{A}$ is initialized with zero weights $\mathbf{A}(\tau = 0) = 0$. In numerical applications, unless specified, we use $\sigma(z) = \tanh(z)$ and $P_{\mathbf{x}} = \mathcal{N}(0, \boldsymbol{I}_d)$.

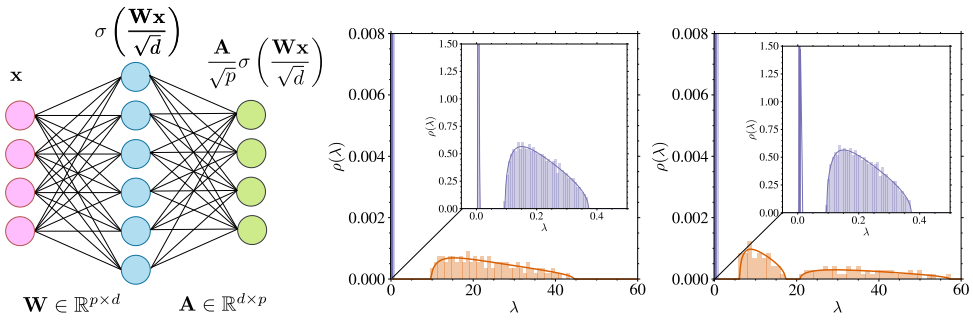
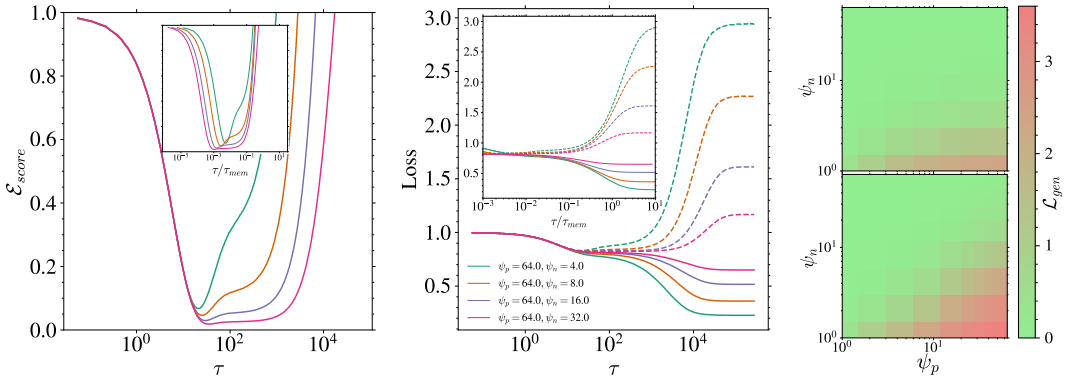


Figure 4: (*Left*) **Illustration of an RFNN.** (*Middle/Right*) **Spectrum of U.** Density $\rho(\lambda)$ from Theorem 3.1 in the overparameterized Regime I described in Theorem 3.2, with $\psi_p = 64$, $\psi_n = 8$, $t = 0.01$, and $\rho_{\mathbf{\Sigma}}(\lambda) = \delta(\lambda - 1)$. The bulk of the spectrum (orange) is between $\lambda \approx 10$ and $\lambda \approx 45$. The histogram shows the eigenvalues from a single realization of $\mathbf{U}$ at $d = 100$. Inset: zoom near $\lambda = 0$ (in blue) showing the first bulk ρ_1 and the delta peak at $\lambda = s_t^2$. (*Right*) Same as (*Middle*), but with $\rho_{\mathbf{\Sigma}}(\lambda) = \frac{1}{2}\delta(\lambda - 0.5) + \frac{1}{2}\delta(\lambda - 1.5)$. The first bulk in blue remains unchanged, as it depends only on $\sigma_{\mathbf{x}}^2 = \mathrm{Tr}(\mathbf{\Sigma})/d = 1$ in both cases, while the second bulk varies with $\mathbf{\Sigma}$.



(A) (B) (C)

Figure 5: **Evolution of the training and test losses for the RFNN.** (A) Distance to the true score $\mathcal{E}_{\text{score}}$ against training time τ for $\psi_n = 4, 8, 16, 32$,$\psi_p = 64$, $t = 0.1$ and $d = 100$. In the inset, the training time is rescaled by $\tau_{\text{mem}} = \psi_p/\Delta_t\lambda_{\min}$. (B) Training (solid) and test (dashed) losses for various ψ_n. The inset shows both losses rescaled by τ_{mem}. (C) Heatmaps of $\mathcal{L}_{\text{gen}}$ for $\tau = 10^3$ (top) and $\tau = 10^4$ (bottom) as a function of ψ_n and ψ_p. All the curves use Pytorch [35] gradient descent. More numerical details can be found in SM Sect. D.

Emergence of the two timescales during training. We first show in Fig. 5 that the behavior of training and test losses in the RF model mirrors the one found in realistic cases in Sect. 2, with a separation of timescales τ_{gen} and τ_{mem} which increases with n. Equation (10) is linear in $\mathbf{A}$ and hence it can be solved exactly (see SM). The timescales of the training dynamics are given by the inverse eigenvalues of the $p \times p$ matrix $\Delta_t\mathbf{U}/\psi_p$. Building on the Gaussian Equivalence Principle [GEP, 14, 15, 33] and the theory of linear pencils [6], George et al. (2025) derive a coupled system of equations characterizing the Stieltjes transform of the eigenvalue density $\rho(\lambda)$ of $\mathbf{U}$ for isotropic Gaussian data that lie in a D-dimensional subspace with $D \leq d$ and $D = \mathcal{O}(d)$. We offer an alternative derivation presented in SM for general variance using the replica method [34] – a heuristic method from the statistical physics of disordered systems – yielding the more compact formulation for obtaining the spectrum stated in Theorem 3.1. Before stating the theorem, we introduce

$$b_t = \mathbb{E}_{u,v}[v\sigma(e^{-t}\sigma_{\mathbf{x}}u + \sqrt{\Delta_t}v)], \quad a_t = \mathbb{E}_{u,v}[\sigma(e^{-t}\sigma_{\mathbf{x}}u + \sqrt{\Delta_t}v)\frac{u}{e^{-t}\sigma_{\mathbf{x}}}], \tag{12}$$

$$v_t^2 = \mathbb{E}_{u,v,w}[\sigma(e^{-t}\sigma_{\mathbf{x}}u + \sqrt{\Delta_t}v)\sigma(e^{-t}\sigma_{\mathbf{x}}u + \sqrt{\Delta_t}w)] - a_t^2 e^{-2t}\sigma_{\mathbf{x}}^2, \tag{13}$$

$$s_t^2 = \mathbb{E}_u[\sigma(\Gamma_t u)^2] - a_t^2 e^{-2t}\sigma_{\mathbf{x}}^2 - v_t^2 - b_t^2, \tag{14}$$

where $\sigma_{\mathbf{x}}^2 = \frac{\mathrm{Tr}(\mathbf{\Sigma})}{d}$, $\Gamma_t = e^{-2t}\sigma_{\mathbf{x}}^2 + \Delta_t = 1 + e^{-2t}(\sigma_{\mathbf{x}}^2 - 1)$ and the expectation is over the u, v, w random variables which are independent standard Gaussian $\mathcal{N}(0,1)$.

Theorem 3.1. *Let* $q(z) = \frac{1}{p}\,\mathrm{Tr}(\mathbf{U} - z\boldsymbol{I}_p)^{-1}$, $r(z) = \frac{1}{p}\,\mathrm{Tr}(\mathbf{\Sigma}^{1/2}\mathbf{W}^T(\mathbf{U} - z\boldsymbol{I}_p)^{-1}\mathbf{W}\mathbf{\Sigma}^{1/2})$ *and* $s(z) = \frac{1}{p}\,\mathrm{Tr}(\mathbf{W}^T(\mathbf{U} - z\boldsymbol{I}_p)^{-1}\mathbf{W})$, *with* $z \in \mathbb{C}$. *Let*

$$\hat{s}(q) = b_t^2\psi_p + \frac{1}{q}, \tag{15}$$

$$\hat{r}(r,q) = \frac{\psi_p a_t^2 e^{-2t}}{1 + \frac{a_t^2 e^{-2t}\psi_p}{\psi_n} r + \frac{\psi_p v_t^2}{\psi_n} q}. \tag{16}$$

Then $q(z), r(z)$ *and* $s(z)$ *satisfy the following set of three equations:*

$$s = \int \mathrm{d}\rho_{\mathbf{\Sigma}}(\lambda) \frac{1}{\hat{s}(q) + \lambda\hat{r}(r,q)}, \tag{17}$$

$$r = \int \mathrm{d}\rho_{\mathbf{\Sigma}}(\lambda) \frac{\lambda}{\hat{s}(q) + \lambda\hat{r}(r,q)}, \tag{18}$$

$$\psi_p(s_t^2 - z) + \frac{\psi_p v_t^2}{1 + \frac{a_t^2 e^{-2t}\psi_p}{\psi_n} r + \frac{\psi_p v_t^2}{\psi_n} q} + \frac{1 - \psi_p}{q} - \frac{s}{q^2} = 0, \tag{19}$$

The eigenvalue distribution of $\mathbf{U}$, $\rho(\lambda)$, *can then be obtained using the Sokhotski–Plemelj inversion formula* $\rho(\lambda) = \lim_{\varepsilon \to 0^+} \frac{1}{\pi}\,\mathrm{Im}\, q(\lambda + i\varepsilon)$.

We now focus on the asymptotic regime $\psi_p, \psi_n \gg 1$, typical for strongly over-parameterized models trained on large data sets. In this limit, the spectrum of $\mathbf{U}$ can be described analytically by the following Theorem 3.2.

Theorem 3.2 (Informal)**.** *Let* ρ *denote the spectral density of* $\mathbf{U}$.

Regime I (overparametrized): $\psi_p > \psi_n \gg 1$**.**

$$\rho(\lambda) = \Big(1 - \frac{1 + \psi_n}{\psi_p}\Big)\delta(\lambda - s_t^2) + \frac{\psi_n}{\psi_p}\,\rho_1(\lambda) + \frac{1}{\psi_p}\,\rho_2(\lambda).$$

Regime II (underparametrized): $\psi_n > \psi_p \gg 1$**.**

$$\rho(\lambda) = \Big(1 - \frac{1}{\psi_p}\Big)\rho_1(\lambda) + \frac{1}{\psi_p}\,\rho_2(\lambda).$$

where ρ_1 *is an atomless measure with support*

$$\left[s_t^2 + v_t^2\left(1 - \sqrt{\psi_p/\psi_n}\right)^2,\ s_t^2 + v_t^2\left(1 + \sqrt{\psi_p/\psi_n}\right)^2\right],$$

and ρ_2 *coincides with the asymptotic eigenvalue bulk density of the population covariance* $\tilde{\mathbf{U}} = \mathbb{E}_{\mathbf{X}}[\mathbf{U}]$*;* ρ_2 *is independent of* ψ_n *and its support is on the scale* ψ_p*. The eigenvectors associated with* $\delta(\lambda - s_t^2)$ *leave both training and test losses unchanged and are therefore irrelevant. In the limit* $\psi_p \gg \psi_n$*, the supports of* ρ_1 *and* ρ_2 *are respectively on the scales* ψ_p/ψ_n *and* ψ_p*, i.e. they are well separated.*

The proofs of both theorems are shown in SM (Sect. C). We recall that training timescales are directly related to eigenvalues λ via the relation $\tau^{-1} = \psi_p/\Delta_t\lambda_{\min}$. Theorem 3.2 therefore demonstrates the emergence of the two training timescales τ_{mem} and τ_{gen} in the overparametrized regime of the RFNN model. They are respectively associated to the measures ρ_1 and ρ_2, which are well separated in regime I, for $\psi_p \gg \psi_n \gg 1$, as shown in Fig. 4.

Generalization: The timescale τ_{gen} on which the first relaxation takes place is associated to the formation of the generalization regime. It is related to the bulk ρ_2 and is or order $1/\Delta_t$. This regime only depends on the population covariance $\mathbf{\Sigma}$ of the data and is independent of the specific realization

of the dataset. On this timescale, which is of order one, both the training $\mathcal{L}_{\text{train}}$ and test $\mathcal{L}_{\text{test}}$ losses decrease. The generalization loss $\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{test}} - \mathcal{L}_{\text{train}}$ is zero, and $\mathcal{E}_{\text{score}}$ tends to a value that we find to scale as $\mathcal{O}(\psi_n^{-\eta})$ with $\eta \simeq 0.59$ numerically (see Fig. 5).

Memorization: The timescale τ_{mem}, on which the second stage of the dynamics takes place, is associated to overfitting and memorization. It is related to the bulk ρ_1, and scales as $\psi_p/\Delta_t\lambda_{\text{min}}$, where λ_{min} is the left edge of ρ_1. In the overparameterized regime $p \gg n$, τ_{mem} becomes large and of order ψ_n/Δ_t, thus implying a scaling of τ_{mem} with n. On this timescale, the training loss decreases while the test loss increases, converging to their respective asymptotic values as computed in [13]. Fig. 5 indeed shows that all training and test curves separate, correspondingly the generalization loss $\mathcal{L}_{\text{gen}}$ increases, at a time that scales with $\psi_p/\Delta_t\lambda_{\text{min}}$, as shown in the inset.

As n increases, the asymptotic ($\tau \to \infty$) generalization loss $\mathcal{L}_{\text{gen}}$ decreases, indicating a reduced overfitting. For $n > n^*(p) = p$, although some overfitting remains (i.e., $\mathcal{L}_{\text{gen}} > 0$), the value of $\mathcal{L}_{\text{gen}}$ is sensibly reduced, and the model is no longer expressive enough to memorize the training data, as shown in [13]. This regime corresponds to the *Architectural Regularization* phase in Fig. 1. We show in Fig. 5 (panel C) how the generalization loss $\mathcal{L}_{\text{gen}}$ varies in the (n, p) plane depending on the time τ at which training is stopped. In agreement with the above results, we find that the generalization–memorization transition line depends on τ and moves upward for larger values of τ, similarly to the numerical results exposed in Fig. 3 and the illustration in Fig. 1.

4 Conclusions

We have shown that the training dynamics of neural network-based score functions display a form of implicit regularization that prevents memorization even in highly overparameterized diffusion models. Specifically, we have identified two well-separated timescales in the learning: τ_{gen}, at which models begins to generate high-quality, novel samples, and τ_{mem}, beyond which they start to memorize the training data. The gap between these timescales grows with the size of the training set, leading to a broad window where early stopped models generate novel samples of high-quality. We have demonstrated that this phenomenon happens in realistic settings, for controlled synthetic data, and in analytically tractable models. Although our analysis focuses on DMs, the underlying score-learning mechanism we uncover is common to all score-based generative models such as stochastic interpolants [3] or flow matching [29]; we therefore expect our results to generalize to this broader class.

Limitations and future works.

- While we derived our results under SGD optimization, most DMs are trained in practice with Adam [25]. In SM Sects. A.3 and D, we show that the two key timescales still arise using Adam, although with much fewer optimization steps. Studying how different optimizers shift these timescales would be valuable for practical usage.
- All experiments in Sect. 2 are conducted with unconditional DMs. We additionally verify in SM Sect. B, using a toy Gaussian mixture dataset and classifier-free guidance [19], that the same scaling of τ_{mem} with n holds in the conditional settings. Understanding precisely how the absolute timescales τ_{mem} and τ_{gen} depend on the conditioning remains an open question.
- Our numerical experiments cover a range of p between 1M and 16M. Exploring a wider range is essential to map the full (n, p) phase diagram sketched in Fig. 1 and understand the precise effect of expressivity on dynamical regularization.
- Finally, our theoretical analysis rely on well-controlled data and score models that reproduce the core effects. Extending these analytical frameworks to richer data distributions (such as Gaussian mixtures or data from the hidden manifold model) and to structured architectures would be valuable to further characterize the implicit dynamical regularization of training score-functions. In particular investigating how heavy-tailed data distribution [2] affect the picture described here could be valuable.
- Although DMs trained on large and diverse datasets likely avoid the memorization regime we study here, some industrial models were shown to exhibit partial memorization [8, 45]. Our results provide practical guidelines (early-stopping, control the network capacity) to train DMs robustly and hence avoid memorization, which can be especially helpful in data-scarce domains (e.g., physical sciences).

Acknowledgments and Disclosure of Funding

The authors thank Valentin De Bortoli for initial motivating discussions on memorization–generalization transitions. RU thanks Beatrice Achilli, Jérome Garnier-Brun, Carlo Lucibello and Enrico Ventura for insightful discussions. RU is grateful to Bocconi University for its hospitality during his stay, during which part of this work was conducted. This work was performed using HPC resources from GENCI-IDRIS (Grant 2025-A0181016159). GB acknowledges support from the French government under the management of the Agence Nationale PR[AI]RIE-PSAI (ANR-23-IACL-0008). MM acknowledges the support of the PNRR-PE-AI FAIR project funded by the NextGeneration EU program. After completing this work, we became aware that A. Favero, A. Sclocchi, and M. Wyart [12] had also been investigating the memorization–generalization transition from a similar perspective.

References

[1] Achilli, B., Ventura, E., Silvestri, G., Pham, B., Raya, G., Krotov, D., Lucibello, C., and Ambrogioni, L. (2024). Losing dimensions: Geometric memorization in generative diffusion.

[2] Adomaityte, U., Defilippis, L., Loureiro, B., and Sicuro, G. (2024). High-dimensional robust regression under heavy-tailed data: asymptotics and universality. *Journal of Statistical Mechanics: Theory and Experiment*, 2024(11):114002.

[3] Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. (2023). Stochastic interpolants: A unifying framework for flows and diffusions.

[4] Baptista, R., Dasgupta, A., Kovachki, N. B., Oberai, A., and Stuart, A. M. (2025). Memorization and regularization in generative diffusion models.

[5] Biroli, G., Bonnaire, T., de Bortoli, V., and Mézard, M. (2024). Dynamical regimes of diffusion models. *Nature Communications*, 15(9957). Open access.

[6] Bodin, A. P. M. (2024). *Random Matrix Methods for High-Dimensional Machine Learning Models*. Phd thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

[7] Bortoli, V. D. (2022). Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*. Expert Certification.

[8] Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. (2023). Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA. USENIX Association.

[9] Cui, H., Krzakala, F., Vanden-Eijnden, E., and Zdeborova, L. (2024). Analysis of learning a flow-based generative model from limited sample complexity. In *The Twelfth International Conference on Learning Representations*.

[10] Cui, H., Pehlevan, C., and Lu, Y. M. (2025). A precise asymptotic analysis of learning diffusion models: theory and insights.

[11] D'Ascoli, S., Refinetti, M., Biroli, G., and Krzakala, F. (2020). Double trouble in double descent: Bias and variance(s) in the lazy regime. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2280–2290. PMLR.

[12] Favero, A., Sclocchi, A., and Wyart, M. (2025). Bigger isn't always memorizing: Early stopping overparameterized diffusion models.

[13] George, A. J., Veiga, R., and Macris, N. (2025). Denoising score matching with random features: Insights on diffusion models from precise learning curves.

[14] Gerace, F., Loureiro, B., Krzakala, F., Mezard, M., and Zdeborova, L. (2020). Generalisation error in learning with random features and the hidden manifold model. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3452–3462. PMLR.

[15] Goldt, S., Loureiro, B., Reeves, G., Krzakala, F., Mézard, M., and Zdeborová, L. (2021). The gaussian equivalence of generative models for learning with shallow neural networks.

[16] Gu, X., Du, C., Pang, T., Li, C., Lin, M., and Wang, Y. (2023). On memorization in diffusion models.

[17] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium.

[18] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models.

[19] Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance.

[20] Hu, H. and Lu, Y. M. (2023). Universality laws for high-dimensional learning with random features. *IEEE Transactions on Information Theory*, 69(3):1932–1964.

[21] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

[22] Kadkhodaie, Z., Guth, F., Simoncelli, E. P., and Mallat, S. (2024). Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*.

[23] Kamb, M. and Ganguli, S. (2024). An analytic theory of creativity in convolutional diffusion models.

[24] Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models.

[25] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *ICLR (Poster)*.

[26] Li, P., Li, Z., Zhang, H., and Bian, J. (2025). On the generalization properties of diffusion models.

[27] Li, S., Chen, S., and Li, Q. (2024a). A good score does not lead to a good generative model.

[28] Li, T., Biferale, L., Bonaccorso, F., and et al. (2024b). Synthetic lagrangian turbulence by generative diffusion models. *Nat Mach Intell*, 6:393–403.

[29] Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

[30] Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., He, L., and Sun, L. (2024). Sora: A review on background, technology, limitations, and opportunities of large vision models.

[31] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[32] Mei, S., Misiakiewicz, T., and Montanari, A. (2019). Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2388–2464. PMLR.

[33] Mei, S. and Montanari, A. (2020). The generalization error of random features regression: Precise asymptotics and double descent curve.

[34] Mézard, M., Parisi, G., and Virasoro, M. A. (1987). *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, volume 9 of *Lecture Notes in Physics*. World Scientific Publishing Company, Singapore.

[35] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035. Curran Associates, Inc.

[36] Price, I., Sanchez-Gonzalez, A., Alet, F., and et al. (2025). Probabilistic weather forecasting with machine learning. *Nature*, 637:84–90.

[37] Péché, S. (2019). A note on the pennington-worah distribution. *Electronic Communications in Probability*, 24:1–7.

[38] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR.

[39] Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

[40] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

[41] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.

[42] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

[43] Shah, K., Kalavasis, A., Klivans, A. R., and Daras, G. (2025). Does generation require memorization? creative diffusion models using ambient diffusion.

[44] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

[45] Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. (2023a). Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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

[47] Song, J., Meng, C., and Ermon, S. (2022). Denoising diffusion implicit models.

[48] Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021a). Maximum likelihood training of score-based diffusion models.

[49] Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[50] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations.

[51] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[52] Ventura, E., Achilli, B., Silvestri, G., Lucibello, C., and Ambrogioni, L. (2025). Manifolds, random matrices and spectral gaps: The geometric phases of generative diffusion.

[53] Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

[54] Wang, B. (2025). An analytical theory of power law spectral bias in the learning dynamics of diffusion models.

[55] Wang, P., Zhang, H., Zhang, Z., Chen, S., Ma, Y., and Qu, Q. (2024). Diffusion models learn low-dimensional distributions via subspace clustering.

[56] Wu, Y.-H., Marion, P., Biau, G., and Boyer, C. (2025). Taking a big step: Large learning rates in denoising score matching prevent memorization.

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

[58] Zhang, C., Zhang, C., Zheng, S., Zhang, M., Qamar, M., Bae, S.-H., and Kweon, I. S. (2023). A survey on audio diffusion models: Text to speech synthesis and enhancement in generative ai.

[59] Zhi-Qin John Xu, Z.-Q. J. X., Yaoyu Zhang, Y. Z., Tao Luo, T. L., Yanyang Xiao, Y. X., and Zheng Ma, Z. M. (2020). Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are supported by the thorough numerical experiments from Sect. 2 and the deep analytical study of random features in Sect. 3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the presented work are discussed in a dedicated paragraph in Sect. 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All the results we establish analytically in Sect. 3 come with a clear description of the assumptions that are summarized in the main text through theorems and complete proofs can be found in the extensive supplemental materials.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All the main practical information used to run the experiments are clearly stated in the main text, and further details are given in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code used to train the models, solve the equations and make the plots is publicly available at the following address: https://github.com/tbonnair/Why-Diffusion-Models-Don-t-Memorize.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details about the architecture and the parameters we used to perform the experiments are given in the main text, and more details can be found in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All plots of the numerical parts include shaded areas corresponding to confidence intervals on the mean or standard deviations over multiple runs when relevant. Sect. 2 explains how they are computed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the information about the resources used to run the experiments can be found in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research presented in this paper is compliant with the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Due to the theoretical nature of the paper, there is no direct broader impact to declare.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not pose such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the openly public dataset CelebA used to carry parts of the numerical experiments.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The present research does not involve LLMs as an important component.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.