Effective Dynamics of Generative Adversarial Networks

Steven Durr,[1] Youssef Mroueh,[2] Yuhai Tu,[2,*] and Shenshen Wang[1,†]
[1]Department of Physics and Astronomy, University of California Los Angeles, Los Angeles, California 90095, USA
[2]IBM T. J. Watson Research Center, Yorktown Heights, New York 10598

(Received 8 December 2022; revised 5 July 2023; accepted 25 August 2023; published 5 October 2023)

Generative adversarial networks (GANs) are a class of machine-learning models that use adversarial training to generate new samples with the same (potentially very complex) statistics as the training samples. One major form of training failure, known as mode collapse, involves the generator failing to reproduce the full diversity of modes in the target probability distribution. Here, we present an effective model of GAN training, which captures the learning dynamics by replacing the generator neural network with a collection of particles in the output space; particles are coupled by a universal kernel valid for certain wide neural networks and high-dimensional inputs. The generality of our simplified model allows us to study the conditions under which mode collapse occurs. Indeed, experiments which vary the effective kernel of the generator reveal a mode collapse transition, the shape of which can be related to the type of discriminator through the frequency principle. Further, we find that gradient regularizers of intermediate strengths can optimally yield convergence through critical damping of the generator dynamics. Our effective GAN model thus provides an interpretable physical framework for understanding and improving adversarial training.

DOI: 10.1103/PhysRevX.13.041004

Subject Areas: Complex Systems, Computational Physics, Nonlinear Dynamics

I. INTRODUCTION

In the past decade, deep generative models have proven to be an impressive tool for sampling from complex distributions. In particular, generative adversarial networks (GANs) have been used to produce realistic data and represent a powerful framework for training generative models [1–4]. Consequently, understanding and improving the training of GANs is of considerable interest.

GANs comprise two neural networks: one called the *generator* G_θ and the other called the *discriminator* D_ϕ (parametrized by θ and ϕ, respectively):

$$\text{generator } G_\theta\colon \mathbb{R}^n \to \mathbb{R}^d, \tag{1}$$

$$\text{discriminator } D_\phi\colon \mathbb{R}^d \to \mathbb{R}. \tag{2}$$

The generator is a function which maps randomly selected points in the latent space to points in data space. The discriminator assigns scores to these simulated data points, as well as to genuine samples from the dataset. During training, the discriminator's goal is to distinguish real data from simulated data (through high and low scores, respectively), while the generator's goal is to increase the score assigned to its outputs by the discriminator [5–8].

Although GANs are both powerful and popular, they are notoriously hard to train. The adversarial nature of the dynamics distinguishes a GAN's objective $\mathcal{L}(\phi, \theta)$ from a standard loss function—one that is bounded from below and which the training algorithm seeks to minimize. Rather than living at the minimum, the ideal parameter settings here are at the saddle points of the loss landscape [5]:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \max_{\phi} \mathcal{L}(\phi, \theta). \tag{3}$$

Convergence to such an equilibrium is difficult to attain, as it requires a careful balancing of the two competing networks during training.

One important form of nonconvergence commonly encountered during GAN training is known as *mode collapse* [9,10]. Mode collapse occurs when samples from the generator fail to capture the full diversity of modes present in the dataset. Instead, the generator's output "collapses" as it only produces samples from relatively few of the available modes in the data distribution.

*yuhai@us.ibm.com
†shenshen@physics.ucla.edu

When mode collapse occurs, during training the generator will focus its distribution on a small subset of the overall data distribution. Eventually, the discriminator learns to identify the concentrated output of the generator, at which point the generator will switch from its current specialization to another [9,10]. The generator's output switching from mode to mode, rather than converging on the distribution as a whole, is a key symptom of mode collapse.

Many practically useful training techniques for avoiding mode collapse have been proposed, often involving modified objective functions and novel regularizers [1,5,7,9]. Here, rather than constructing empirical methods for reducing mode collapse, we seek to understand this phenomenon from the perspective of dynamical systems, determine the physical meaning of competing factors, and derive principles to guide the training of GANs.

The dynamics of learning in neural networks have been studied in weight space [11,12]. Here, we map GANs to an effective model in which the output of the generator network is replaced by N particles in $\mathbb{R}^d$. The learning dynamics in GANs can then be studied in the output space by following the motion of the N "output" particles, which descend the loss landscape set by both the discriminator's score function and the collective state of N particles. We additionally incorporate a static neural tangent kernel (NTK)—a feature of realistic GANs using an infinite-width generator. Within our effective model, the NTK induces a dependence of the velocity of any particle on the discriminator gradient at the location of all particles. As a result of the sampling procedure of generators and the form of common NTKs, we show the presence of universality within a restricted set of neural network architectures; many different types of infinite-width generator neural networks may lead to the same particle dynamics. As a result of the sampling procedure of generators and the form of common NTKs, we show the presence of universality—many different types of infinite-width generator neural networks map to the same particle dynamics.

We argue that this effective model provides a simplified and interpretable framework in which to understand mode collapse. In particular, applying this model to a low-dimensional target distribution, we show a transition from convergence to mode collapse as a function of the NTK and the relative training time. We provide a physical interpretation which explains this transition in terms of learning characteristics of the discriminator.

Finally, we use this model to study GAN regularization—modification of the training objective in order to promote convergence. We find that when a gradient regularizer [13] is introduced, it results in a reduction of mode collapse in our model GAN. Additionally, by sweeping over regularization strengths, we are able to observe underregularized, overregularized, and critically regularized regimes. These regimes can be understood by analogy to the physics of a damped oscillator and its under, over, and critically damped cases. The regularizer, which incentivizes a "smoother" generator, here plays the role of a damping term.

II. TRAINING AND FAILURE

In GANs, the generator is a neural network G_θ, which is fed random inputs, $z \in \mathbb{R}^n$, selected from some noise distribution $q(z)$. The generator outputs, $G_\theta(z) \in \mathbb{R}^d$, therefore represent samples from its implicit probability distribution in data space $p_\theta(X)$. Conceptually, the generator and discriminator seek to minimize and maximize an objective expressing the expected difference between the dataset and the generator's outputs:

$$\mathcal{L}(\phi, \theta) = \langle D_\phi(x) \rangle_{x\sim p(x)} - \langle D_\phi(G_\theta(z)) \rangle_{z\sim q(z)}. \quad (4)$$

The function $p(x)$ is the probability distribution of samples in the dataset, while $q(z)$ is the distribution from which seeds in the latent space are sampled.

In practice, however, the discriminator's objective is often modified to include regularization, restricting the magnitude of the discriminator network and promoting stability [7,14]. Different GAN implementations exist, many with distinct objectives [14]. Here we consider objective functions of the following form [7,8,14,15] that characterize the discriminatory power under constraints:

$$\mathcal{L}_D \equiv \langle D_\phi(G_\theta(z)) \rangle_{z\sim q(z)} - \langle D_\phi(x) \rangle_{x\sim p(x)} + \lambda R(D_\phi, G_\theta), \quad (5)$$

$$\mathcal{L}_G \equiv \langle D_\phi(x) \rangle_{x\sim p(x)} - \langle D_\phi(G_\theta(z)) \rangle_{z\sim q(z)}. \quad (6)$$

$\mathcal{L}_\mathcal{D}$ and $\mathcal{L}_\mathcal{G}$ define the objectives for the discriminator and generator, respectively, where $R(D_\phi, G_\theta)$ represents a regularizer on the discriminator, limiting its magnitude under a norm of interest (here, we use an L_2-norm on the discriminator weights ϕ) with $\lambda \geq 0$ denoting the strength of the regularizer.

The discriminator parameters ϕ evolve to maximize the expected difference between the discriminator's value on the real data and the generated data [Eq. (5)], while the generator parameters θ evolve to minimize this difference:

$$\dot{\phi} = -\alpha_D \frac{d\mathcal{L}_D}{d\phi}, \qquad \dot{\theta} = -\alpha_G \frac{d\mathcal{L}_G}{d\theta}. \quad (7)$$

The discriminator and generator evolution occurs at individual learning rates α_D and α_G.

Practically, in neural networks, the loss function is defined using mini batches of N samples of real data and generated data, both of which are resampled at each training step:

$$\mathcal{L}_D^{(N)} \equiv \frac{1}{N}\sum_{i=1}^{N} D_\phi(G_\theta(z_i)) - \frac{1}{N}\sum_{i=1}^{N} D_\phi(x_i) + \lambda R(D_\phi, G_\theta), \tag{8}$$

$$\mathcal{L}_G^{(N)} \equiv \frac{1}{N}\sum_{i=1}^{N} D_\phi(x_i) - \frac{1}{N}\sum_{i=1}^{N} D_\phi(G_\theta(z_i)). \tag{9}$$

Training is performed in iterations. First, for n_{disc} steps, the discriminator is updated according to its stochastic gradient:

$$\phi \leftarrow \phi - \alpha_D \nabla_\phi \mathcal{L}_D^{(N)}. \tag{10}$$

Then, for a single step, the generator is updated analogously with stochastic gradient descent:

$$\theta \leftarrow \theta - \alpha_G \nabla_\theta \mathcal{L}_G^{(N)}. \tag{11}$$

Alternating updates are repeated until convergence, or until training is halted after a large number of iterations.

Mode collapse occurs when the generator's outputs focus on a few of the available modes, rather than replicating the full data distribution. During training, once the discriminator learns that the generator is focused at a particular mode, it assigns low scores to the data points coming from this mode. The response of the generator is then to shift its output distribution to another mode. Mode collapse is therefore characterized by the generator's distribution switching from mode to mode throughout training.

III. GENERATOR PARTICLES AND UNIVERSALITY

Rather than following the dynamics of generator parameters [Eq. (11)], we study instead the time evolution of generator outputs treated as particles in data space (Fig. 1), an approach applied in Ref. [16] and later in Ref. [17]. While each generator parameter follows its own local (stochastic) gradient, as we will show below, the dynamics of generator outputs are explicitly correlated.

With a time-dependent vector of parameters θ_t, a fixed seed z maps to a point in data space at time t according to

$$X_t = G_{\theta_t}(z).$$

This mapping relates updates in parameter space $d\theta_t$ to updates in data space dX_t by

$$dX_t = \frac{dG_\theta(z)^T}{d\theta}\bigg|_{\theta=\theta_t} \frac{d\theta_t}{dt}\, dt. \tag{12}$$

Under gradient dynamics, the generator parameters evolve according to

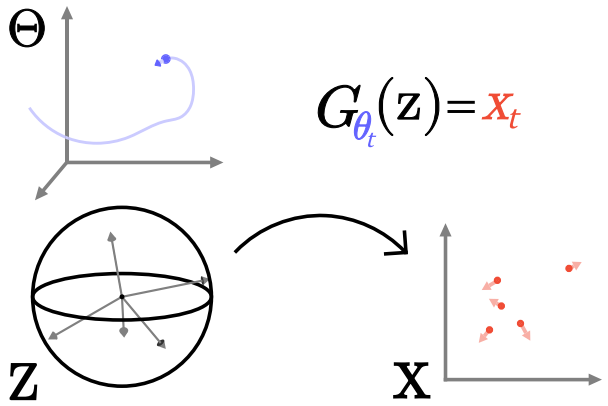

FIG. 1. Mapping to an effective GAN model. An illustration of how the input vectors z in the seed space $\mathbf{Z}$ (sampled from a high-dimensional sphere) map to particles in the data space $\mathbf{X}$. During GAN training, generator parameters θ_t evolve over time in the $\mathbf{\Theta}$ space (upper left, blue trajectory). As a result, given fixed inputs $\{z\}$ (lower left), the set of data space outputs $\{X_t\}$ also evolves in time (lower right). It is the dynamics of these points in data space that our effective model directly describes.

$$\dot{\theta}_t = -\alpha_G \frac{d\mathcal{L}_G}{d\theta_t} = \alpha_G \frac{d}{d\theta_t}\langle D_\phi(G_\theta(z))\rangle_{z\sim q(z)}, \tag{13}$$

and so

$$\frac{d\theta_t}{dt} = \alpha_G \frac{d}{d\theta}\left(\int dz' q(z') D(G_\theta(z'))\right)\bigg|_{\theta=\theta_t} \tag{14}$$

$$= \alpha_G \int dz' q(z') \nabla_j D(G_{\theta_t}(z')) \frac{\partial G_\theta^j(z')}{\partial \theta}\bigg|_{\theta=\theta_t}. \tag{15}$$

To see the corresponding data space dynamics, we plug this into Eq. (12) and write

$$dX_t^i = \alpha_G dt \int dz' \Gamma_{\theta_t}^{i,j}(z,z') \nabla_j D(G_{\theta_t}(z')) q(z'), \tag{16}$$

where i and j index the components of the data vector X_t and repeated indices are summed over.

Moreover, we have introduced the neural tangent kernel [18] $\Gamma_\theta^{i,j}(z,z')$, defined by

$$\Gamma_\theta \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{d\times d}, \tag{17}$$

$$\Gamma_\theta^{i,j}(z,z') = \sum_k \frac{\partial G^i(z)}{\partial \theta_k}\frac{\partial G^j(z')}{\partial \theta_k}, \tag{18}$$

where n and d are the dimensions of the inputs and the data space, respectively, and θ_k denotes the kth generator parameter. Importantly, Eq. (16) makes clear that the NTK Γ_θ couples the generator outputs; it specifies to what extent the dynamics of the generator particle at $X = G_\theta(z)$ is

influenced by the discriminator gradients at the position $X' = G_\theta(z')$ of all other particles.

In general, NTKs evolve during training. However, for larger-width networks, the weights θ_t will asymptotically remain in the vicinity of their initial values θ_0. The network's NTK, which involves a sum over the network's weights, changes even less—in the infinite-width limit becoming fixed at initialization [18,19] (see Appendix G for an example of such large-width training dynamics). In this work, we will assume that the generator is in this infinite-width regime, and enforce that the generator NTK remains fixed during training: $\Gamma_{\theta_t} = \Gamma_{\theta_0}$.

The infinite-width regime is of particular interest, as the performance of neural networks has been observed to improve as their width is increased. Additionally, in this limit, it becomes possible to derive analytical results, as certain theoretical aspects of neural networks simplify [18–22]. The exact form of the infinite-width NTK can be found for particular network architectures, such as those with a ReLU or Erf activation [22].

A. Mapping to model GANs

In generative adversarial networks, random seeds are provided to the generator by sampling from a so-called noise distribution $q(z)$ at each iteration. Usually, this is taken to be a high-dimensional Gaussian. Noting that points from $\mathcal{N}^n(0,1)$ are approximately on a sphere of radius $\sqrt{n}$ in n dimensions [23], we take the noise distribution as a uniform selection from a $(n-1)$ sphere.

For certain activations (most prominently, ReLU), if input seeds have a fixed magnitude, then the infinite-width NTK will be a function only of the angle between inputs, $\varphi_{z,z'}$: $\Gamma(z,z') = \Gamma(\varphi_{z,z'})$ [22,24]. Additionally, these samples selected uniformly from a high-dimensional sphere will, with high probability, be nearly orthogonal [25]. Therefore, given such an NTK and high-dimensional inputs, it becomes possible to estimate the distribution of NTK values within a mini batch.

Using these observations, we propose a simplification of the GAN training protocol. Within our simplified model, we take the generator to be of large width with a static NTK. The noise distribution $q(z)$ is taken to be a uniform distribution over a high-dimensional sphere. Finally (as in wide ReLU networks with inputs of fixed magnitude), we take the NTK to be a function of the dot product of inputs only.

Our first assumption fixes the NTK at initialization [18,22]. The latter two concentrate the pairwise NTK values, obtained using one sample of inputs $\{z\}$, to two characteristic numbers g_1 and g_2. The first number, $g_1 \equiv \Gamma(\varphi_{z,z} = 0)$, corresponds to evaluations involving the same point, and the second, $g_2 \equiv \Gamma(\varphi_{z,z'} = \pi/2)$, corresponds to pairs of distinct points chosen from the high-dimensional latent space. The values of g_1 and g_2 are determined by the architecture of the network, but can also be modified by, for instance, the use of batch normalization [26,27].

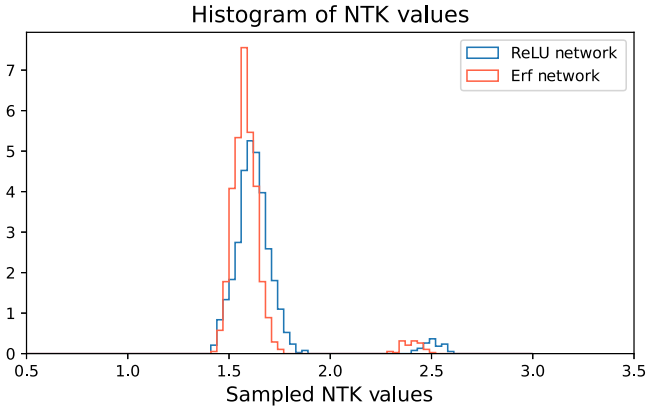

FIG. 2. Universality of NTK values. Two distinct network architectures result in similar distributions of sampled NTK values. Fifty inputs $\{z_i\}$ are sampled from a unit sphere in 100 dimensions. Using two untrained networks, the NTK values for all pairs of inputs are computed. The red histogram is obtained using a single hidden-layer network (width 2048) with an Erf activation, while for the blue histogram a ReLU activation is used. Zero-mean normal distributions with distinct variances are used to initialize two networks' weights and biases. Despite their differences, the two networks' NTK values are approximately characterized by the same two numbers, $\Gamma(\varphi_{z,z'} \approx \pi/2)$ for distinct inputs and $\Gamma(\varphi_{z,z} = 0)$ for pairs of the same input.

The fact that an entire generator neural network, with its activation functions and individual weight and bias distributions, can be to an extent characterized by just two numbers, suggests a sort of universality within this particular set of neural network architectures. Many different generator neural network architectures may be mapped onto the same system, parametrized only by (g_1, g_2). This universality can be observed in Fig. 2, in which two distinct networks (one with ReLU activation, the other with Erf, both using a single hidden layer with 2048 units [28]) are observed to have very similar pairwise NTK values across a sample from a unit sphere in 100 dimensions.

Additionally, we consider a restricted version of GAN training. Rather than resampling from the noise distribution [i.e., taking a new mini batch from $q(z)$] at each training iteration, we instead train by effectively using one fixed set of N generator inputs $\{z\}$.

Although our approximations reduce the model's exact resemblance to practical GAN frameworks, our simplified model still tackles the same min-max game that governs the dynamics of a full GAN. Moreover, while in practice both the dimension of latent space and the width of neural networks are finite, our assumptions yield reasonable and efficient approximations that allow systematic experimentation. Most importantly, the resulting simplifications enable physical interpretation and control of learning dynamics, which provide broader insights beyond the limiting regime.

Based on these simplifications, we propose a coarse-grained NTK of the form

$$\Gamma^{i,j}(z,z') = \delta_{i,j}(g_1\delta_{z,z'} + g_2(1-\delta_{z,z'})). \tag{19}$$

This NTK is static throughout training, and its two constants, g_1 and g_2, characterize the NTK values for pairs of identical and distinct points, respectively.

This NTK allows us to further simplify the effective model—ignoring the latent space entirely, and instead explicitly correlating particles in data space:

$$\Gamma^{i,j}_{a,b} = \delta_{i,j}(g_1\delta_{a,b} + g_2(1-\delta_{a,b})). \tag{20}$$

Here, a and b index particles, while i and j index components in data space. Out front, $\delta_{i,j}$ can be understood as implying a lack of correlation between the gradients of output degrees of freedom of a wide neural network—this is exact in the infinite-width limit [18] and further described in Appendix H. g_1 and g_2 set the degree to which the discriminator gradient at identical and distinct points, respectively, contribute to a generator particle's velocity [29],

Using this effective NTK, we can model the dynamics of data points in output space by dynamics of coupled generator particles:

$$\frac{dX^i_a}{dt} = \frac{\alpha_G}{N}\sum_{b,j}^{N,d}\Gamma^{i,j}_{a,b}\nabla_j D(X_b). \tag{21}$$

These dynamics are reminiscent of flocking behavior, in which local velocities are found through a spatial average [30]. Here, however, the average is not over velocities, but over discriminator gradients. In addition, the average is taken over all particles, rather than over a local region.

B. Multimodal target

We now proceed with our simplified GAN training protocol, replacing the generator network with a collection of N particles in data space and using the generator update rule of Eq. (21) rather than that of Eq. (16).

As a case study, we consider a canonical two-dimensional problem of training a GAN on a distribution of eight Gaussians arranged in a circle of radius 2, each having a standard deviation 0.02 [31]. Since each Gaussian can naturally represent a distinct mode, this data distribution is used throughout GAN literature as a toy dataset for observing mode collapse [9,10,13,32]. Mode collapse in this context would correspond to a generator whose outputs are focused on one, or a subset, of the eight Gaussians. During training, mode collapse would cause the outputs to oscillate between distinct modes, without splitting to cover all eight.

The generator particles are taken to be 2000 parametrized points in the plane, initialized as a Gaussian distribution with $\sigma = 0.5$, while the discriminator is a ReLU network with four hidden layers of width 512 [33]. The discriminator parameters ϕ and the generator points X_a are both updated during training according to their objective functions, following training routine described in Algorithm 1.

In Figs. 3 and 4, we show time slices of the training progress. The generator particles are shown in white on a heat map of the discriminator values. We begin by running an experiment using a diagonal NTK ($g_2/g_1 = 0$ [35]). In this case, the generator particles independently ascend the local gradient of the discriminator: $\dot{X}^i_a \propto \nabla_i D(X_a)$. Visually (Fig. 3), this corresponds to each particle (in white) drifting up the color gradient (taking steps toward lighter regions). Meanwhile, the discriminator modifies its parameters to increase the difference between the expectation on the real data and the generator particles—assigning higher values (brighter colors) to the eight data points in black, and lower values (darker colors) to the particles in white.

We observe the result of this dynamic in Fig. 3. Initially, the discriminator assigns low values to the cluster of particles. However, the initial cloud of particles rapidly splits apart, and the adversarial dynamic results in informative gradients being passed to the generator particles, which quickly converge to the full multimodal distribution.

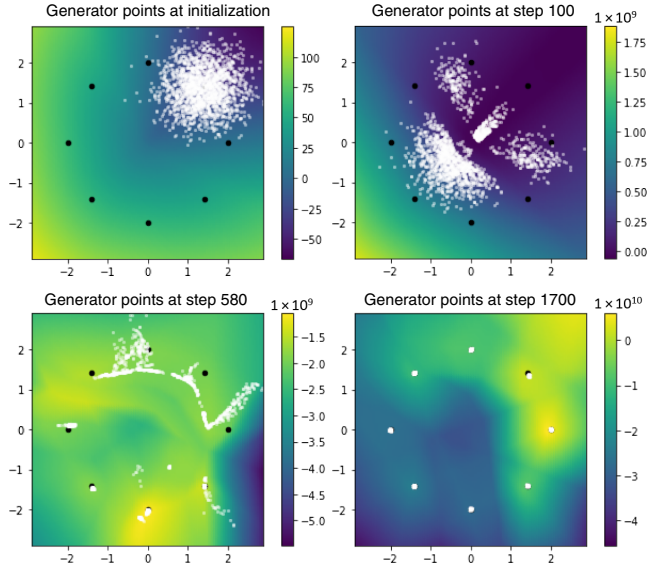

FIG. 3. Convergence of the model GAN dynamics under a diagonal NTK ($g_2 = 0$). The generator particles are shown in white, while the discriminator values are color coded, with higher values shown in brighter colors. Because the NTK is diagonal, the particles' velocities are not explicitly correlated, and they ascend their local gradients (from darker to brighter colors), $\dot{X} \propto \nabla D(X)$. Meanwhile, the discriminator attempts to maximize the difference between its expectation on the data (black points) and generator particles (in white). Initially, the discriminator assigns a low value to the cluster's location. Consequently, the cluster rapidly splits apart (step 100), with each point following the local discriminator gradient. The combined adversarial dynamics are seen to result in convergence to the eight modes.

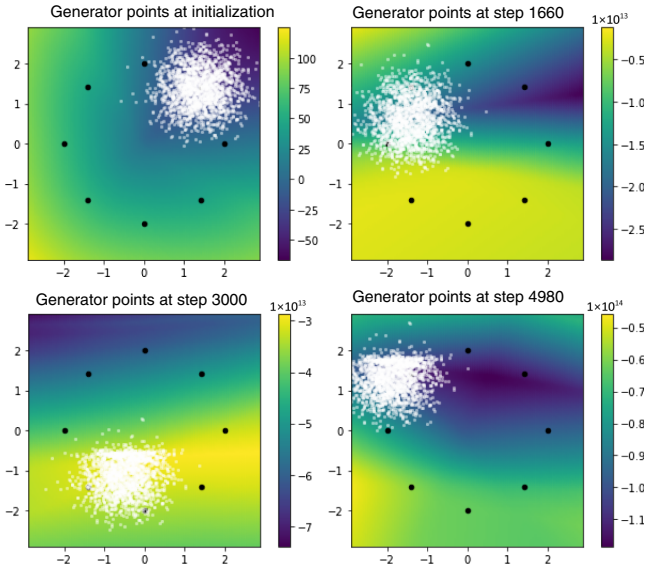

FIG. 4. Model GAN dynamics exhibit mode collapse under all-to-all coupling by a nondiagonal NTK ($g_2/g_1 = 1/5$). As in Fig. 3, dynamics are depicted over time. During training, the discriminator seeks to assign higher values (brighter colors) to real data points (the eight points in black), and lower values (darker colors) to generated points (white dots). As had occurred in Fig. 3, the discriminator initially places its minimum at the position of the generator particle cluster. Now, however, due to correlations in particle velocity, the cluster no longer splits apart. Instead, it shifts away from discriminator minima before splitting can happen. As a result, the cluster of generator particles switches from mode to mode, as the discriminator attempts to "catch up"—a behavior indicative of mode collapse.

In a second experiment, we begin with the same initialization, but instead use an NTK satisfying $g_2/g_1 = 1/5$. Because of the nontrivial off-diagonal terms of the NTK, the velocities of the particles are correlated. As is shown in Fig. 4, generator particles (in white) no longer split apart. Instead, they stay together as the entire cluster shifts from mode to mode indefinitely. The discriminator repeatedly attempts to assign low values (darker colors) to the generator particle cluster's spatial region.

This behavior is a key signature of mode collapse, and suggests an understanding of this phenomenon through the lens of our model. For the remainder of this work, we will identify the observed failure of convergence and switching between modes with mode collapse. By varying g_2/g_1, we will probe the onset of this failure mode and investigate what training algorithms and discriminator characteristics would lead to improved performance.

IV. MODEL GAN EXPERIMENTS—THE MODE COLLAPSE TRANSITION

We have observed that the ratio g_2/g_1 may be increased to induce mode collapse. Apart from the architecture of the discriminator network, the remaining adjustable parameters in Algorithm 1 concern the relative training dynamics of the discriminator and generator. The parameters α_D and α_G control the step size of the discriminator and generator, respectively, while n_{disc} tunes the number of discriminator steps taken for each generator step.

We will therefore vary these parameters to examine the relationship between g_2/g_1 and the discriminator's dynamics. The latter can be varied in two ways: by modifying the learning rate α_D or by modifying the value of n_{disc} used in the algorithm. Here, we show the result of modifying n_{disc}, leaving α_D experiments (which produce similar results) to Appendix A.

To characterize whether, at a given time, generator particles have converged or collapsed to a single mode, we define a metric based on the entropy of the distribution. Letting P_i be the fraction of particles for which the ith mode is the nearest, we define the following:

$$\text{mode collapse metric} = \log(8) + \sum_i P_i \log(P_i). \tag{22}$$

Note that $P_i = 1/8$, $i = 1, 2, \ldots, 8$, would give a complete mode coverage with an even split, and have a value of 0. On the other hand, $P_1 = 1$, $P_{i>1} = 0$ would correspond to all generator points being nearest to a single mode, giving a mode collapse metric value of log 8.

To further characterize the quality of convergence of the generator particles, we can compute the average log-likelihood, $(1/N)\sum_a \log[p(X_a)]$, where $p(X)$ is the probability density of the multimodal Gaussian distribution.

Algorithm 1. The coarse-grained, (g_1, g_2) GAN training algorithm.

for iteration number **do**
  for n_{disc} **do**
    i Sample N data points,$\{x_i\}$ from the eight-Gaussian distribution.
    ii Compute

$$\mathcal{L}_D^{(N)} = \frac{1}{N}\sum_{a=1}^{N} D_\phi(X_a) - \frac{1}{N}\sum_{l=1}^{N} D_\phi(x_l) + \frac{\lambda}{2}\sum_k \phi_k^2$$

    and update discriminator parameters by descending its stochastic gradient

$$\phi \leftarrow \phi - \alpha_D \nabla_\phi \mathcal{L}_D^{(N)}$$

  end for
  iii update X_a according to Eq. (21)

$$X_a^i \leftarrow X_a^i + \alpha_G \frac{1}{N}\sum_{b,j}^{N,d} \Gamma_{a,b}^{i,j} \nabla_j D(X_b)$$

end for

The combination of these two metrics (mode collapse and log-likelihood) indicates whether the generator points have both avoided mode collapse and successfully converged to the modes of the distribution.

A. GAN setup

To maximize the interpretability of our results, we employ a simpler discriminator with a single wide hidden layer (2048 units):

$$D(x) = \sqrt{\frac{2}{\text{width}}} a_i \sigma(w_i^j x^j + b_i). \tag{23}$$

Details of the initialization can be found in Appendix A. The activation function is set to ReLU, $\sigma(x) = \max(0, x)$. Experiments employing a Tanh activation were also performed and the results can be found in Appendix C. The target data distribution is again taken to be the eight Gaussians. A total of 200 generator particles are initialized at $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ with a standard deviation of 0.1.

The discriminator loss function is defined as

$$\mathcal{L}_D = \langle D(X)\rangle_{\text{gen}} - \langle D(x)\rangle_{\text{target}} + \frac{1}{\text{width}}\left(\sum_{i,j}(w_i^j)^2 + \sum_i a_i^2 + \sum_i b_i^2\right), \tag{24}$$

where $\langle D(X)\rangle_{\text{gen}}$ is the expectation of the discriminator on the generator distribution, $(1/N)\sum_a D(X_a)$, while $\langle D(x)\rangle_{\text{target}}$ is its expectation on the data distribution (the eight Gaussians). The remaining terms represent an L_2 regularizer on the weights, placing an overall restriction on the discriminator.

Following Eq. (21), particle velocities are given by

$$\frac{dX_a^i}{dt} = \frac{\alpha_G}{N}\sum_{b,j}^{N,d}\Gamma_{a,b}^{i,j}\nabla_j D(X_b) \tag{25}$$

$$= \alpha_G\left(\frac{g_1 - g_2}{N}\nabla_i D(X_a) + g_2\langle\nabla_i D(X)\rangle\right). \tag{26}$$

Here the angular bracket indicates an average over all generator particles. Hence, each particle, at position X_a, experiences a competition between the mean discriminator gradient over the ensemble $g_2\langle\nabla D(X)\rangle$ and the contribution from its local gradient $((g_1 - g_2)/N)\nabla D(X_a)$.

The entire system is trained using Algorithm 1.

B. Results and interpretation

We run the model GAN training algorithm for each $(g_2/g_1,\ n_{\text{disc}})$ pair considered. After training for a fixed number of iterations, and computing the mode collapse metric [Eq. (22)] for all pairs, we can observe a clear transition from convergence (blue data points) to mode collapse (yellow data points), as shown in Fig. 5. A complementary metric representing the quality of convergence across the transition is plotted in Fig. 6. In the following, we provide a heuristic argument to explain the observed characteristics of the mode-collapse transition.

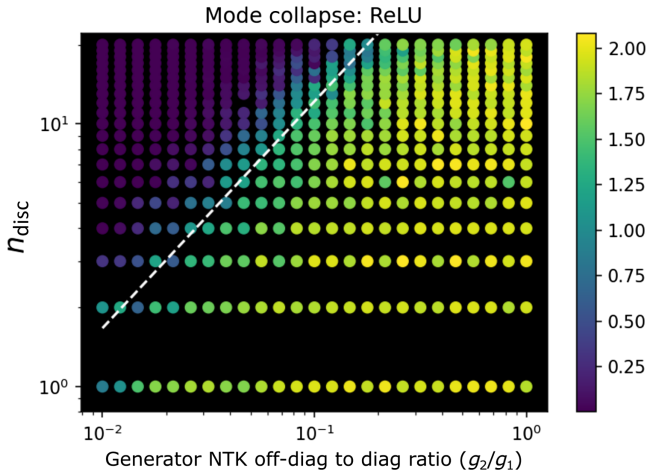

FIG. 5. Phase diagram of the transition to mode collapse using a ReLU discriminator. Mode collapse metric values are shown as a function of generator NTK and discriminator training rate. Brighter points indicate mode collapse while darker points correspond to an even distribution of generator particles over the target data. The x axis gives the value of g_2/g_1 for a given experiment, while the y axis indicates the number of training steps the discriminator takes at each iteration. As g_2/g_1 is increased, the discriminator requires more "time" (a greater n_{disc} value) in order to shift the training from mode collapse (bright points) to convergence (darker points). Experimental results are taken after 5000 training iterations, and data are time averaged, representing the mean of results taken at $5000 \pm n \times 20$, with $n = 0, 1, 2, 3$. A line is fit to $(1/2)\log 8$, indicating a power-law boundary.

During training, the discriminator seeks to maximize the difference between its expectation on the training data and on the generator distribution. Suppose, to this end, the discriminator has formed its minimum within a region (cluster) of generator particles. According to the equation of motion for the generator particles [Eq. (26)], if the term involving local gradients dominates over $g_2\langle\nabla D(X)\rangle$, then the particles in this cluster "split apart"; each particle follows its own local gradient, regardless of the location of the discriminator minimum within the region. As a result, the generator particle cluster, which corresponds to mode collapse, can be split and the full targeted distribution can be recovered.

However, for sufficiently large g_2, the term $g_2\langle\nabla D(X)\rangle$ may dominate. Now the location of the discriminator's minimum within the region becomes important, as it may determine both the magnitude and direction of $\langle\nabla D(X)\rangle$. As is depicted in Fig. 7, if the discriminator obtains a minimum far from the center of a cluster, $|\langle\nabla D(X)\rangle|$ would become non-negligible, leading to an onset of instability,

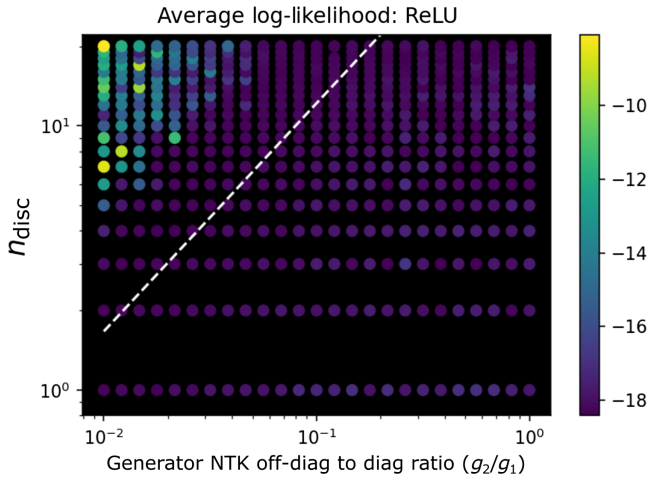

FIG. 6. Precise convergence occurs well above the mode-collapse transition. The average log-likelihood of generator particles is depicted following the same procedure as in Fig. 5. The power-law transition of Fig. 5 is shown as a reference; only sufficiently far above this boundary would particles converge precisely to the eight modes.

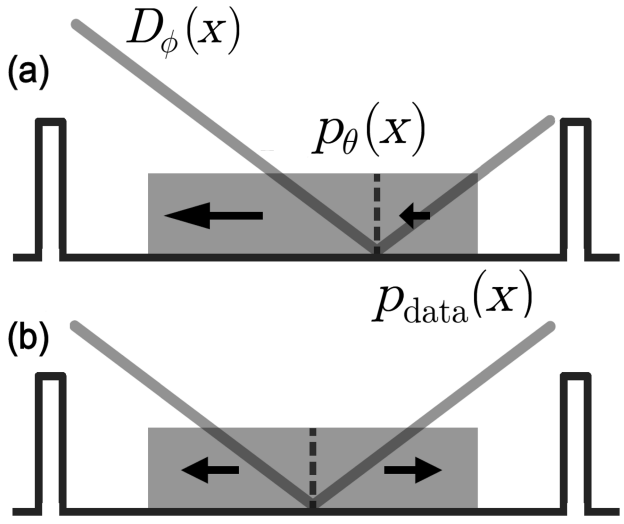

FIG. 7. Increased discriminator precision may result in splitting of a cluster of generator particles. A cluster of generator particles is depicted as a uniform shaded region (with distribution p_θ) between two modes of the target distribution (p_{data}). The value of the discriminator is shown as a simple function of the form $D_\phi(x) \equiv a|x-b|$. Arrows indicate local velocities of the particles on either side of the discriminator minimum. (a) Imprecision of the discriminator leads to a subminimal expectation $\langle D(X)\rangle$ and a significant $\langle \nabla D(X)\rangle$ value, causing all particles to slip to the left due to all-to-all coupling through the NTK. (b) The discriminator has found the precise minimum at the center of the distribution, which kills off average gradients and allows particles to ascend their local slopes; the cluster is thus splitting apart.

and causing the entire group of generator particles to "slip away"—including those on the opposing slope [Fig. 7(a)]. In contrast, for a minimum closer to the cluster's center, the mean gradient experienced by the particle cloud becomes sufficiently small to allow the cluster to "split apart" [Fig. 7(b)].

In this way, the discriminator's precision in minimizing its value over a cluster of generator particles influences its ability to split apart the cluster. More spatially precise discriminators may yield smaller values of $|\langle \nabla D(X)\rangle|$, allowing local gradients to dominate particle dynamics.

The mode-collapse data resulting from using a ReLU discriminator is shown in Fig. 5 on a log-log scale. A dashed white line emphasizes a visible power-law boundary separating mode collapse from convergence. Examples of generator particle distributions sampled across this transition can be found in Appendix D. Figure 6 plots the log-likelihood data and shows that only sufficiently far above the transition boundary would particles converge precisely to the target modes.

This power-law behavior matches another feature of wide ReLU networks, referred to as the frequency principle [36–38]. As networks learn, they tend to first learn lower-frequency functions, before including higher-frequency contributions. This behavior thus sets a rate $\gamma(k)$ at which a network can learn a feature of spatial frequency k. For example, within wide ReLU networks, $\gamma(k)$ is expected to be power law, whereas for wide Tanh networks, an exponential $\gamma(k)$ is predicted [38].

If we identify spatial features of size $1/k$ as having a dominant spatial frequency of k, then simple arguments suggest (Appendix B) that in order to split a cluster, the maximum allowable spatial imprecision falls with increasing g_2 as $(g_1-g_2)/g_2$. This indicates that to break apart such a distribution, we require the discriminator to learn a feature with spatial *frequency* proportional to $g_2/(g_1-g_2)$; for $g_2 \ll g_1$, roughly, $k \sim g_2/g_1$.

Assuming that the discriminator has a frequency-dependent learning rate $\gamma(k)$, then the time required to learn such a feature scales as $T \sim 1/\gamma(k)$. The necessary discriminator steps n_{disc} (and learning rate α_D) to overcome mode collapse would then scale as $1/\gamma(k)$.

Experiments involving wide Tanh networks, among other discriminators, are described in Appendix C. Interestingly, Tanh discriminators display a roughly exponential transition boundary as predicted, which lends further support to the above-proposed explanation. As such, the frequency principle suggests a connection between the relevant length scales and timescales of the discriminator's learning objective. That is, it takes a longer training time to learn a finer spatial feature.

C. Relation to catastrophic forgetting

By analyzing effective properties of the generator network, and how these might interplay with those of the discriminator, our analysis suggests an explanation for the onset of mode collapse in terms of insufficient discriminator precision and hence overwhelmed local gradients

of the discriminator's loss landscape. However, the mode collapse dynamics we observe (shown in Fig. 4) can also be understood through the lens of catastrophic forgetting [32].

Within GANs, the task of the discriminator is to distinguish between the target distribution and that of the generator. As the generator's distribution evolves over time, so does this task. When GANs do converge, the sequence of discriminator tasks does so as well. Catastrophic forgetting occurs when this sequence of tasks fails to converge, and consequently the lessons learned by the discriminator from previous tasks become irrelevant for the current task.

From this perspective, the dominant source of mode collapse is indeed catastrophic forgetting. This is apparent from our experiments in which the discriminator tries and fails to split apart the cluster of generator particles. When the generator's tightly focused distribution travels from mode to mode, the task of the discriminator changes significantly over time. Indeed, the nature of mode collapse in our empirical studies is highly similar to that observed in Ref. [32].

When convergence does occur (for instance, when $g_2/g_1 = 0$ as depicted in Fig. 3), we can see that the discriminator's task also converges. Consequently, the lessons learned at previous training steps remain relevant to the current task.

V. CRITICAL REGULARIZATION

Various regularization techniques have been applied to the problem of mode-collapse avoidance [9,10,13]. Here, we demonstrate that even in our simplified model GAN, the effect of regularization on reducing mode collapse can be observed.

Following Ref. [13], we introduce a gradient regularizer,

$$\beta||\nabla_\theta\langle D(G_\theta(z))\rangle||^2/2, \tag{27}$$

into the discriminator's loss function during training. Since the velocities of generator parameters are driven by local gradients of the discriminator, this term is analogous to the kinetic energy of these parameters. This regularization term penalizes sharp gradients and encourages generators to take smoother paths to the target distribution. The effect of the regularizer on the GAN system can be viewed in analogy to a damping term in physics (a connection made explicit in Appendix E), with oscillations from mode to mode corresponding to an underdamped regime, slow convergence to all available modes corresponding to overdamping, and a most efficient convergence corresponding to critical damping. Using this analogy as a conceptual starting point, we can sweep over β to identify a regime of "critical regularization."

Despite the fact that our model GAN setup does not have any reference to generator parameters (or a generator network), we may still incorporate such a term into training via the effective NTK:

$$\begin{aligned}&||\nabla_\theta\langle D(G_\theta(z))\rangle||^2\\&=\frac{1}{N^2}\sum_{z,z'\sim q(z)}\nabla_i D(G_\theta(z))\nabla_\theta G^i_\theta(z)\nabla_\theta G^j_\theta(z')\nabla_j D(G_\theta(z'))\\&=\frac{1}{N^2}\sum_{z,z'\sim q(z)}\nabla_i D(G(z))\Gamma^{i,j}_\theta(z,z')\nabla_j D(G(z')).\end{aligned}$$

This effective form immediately allows us to apply the regularizer within the model GAN by including the following term in Eq. (5):

$$\frac{\beta}{N^2}\sum_{a,b}\nabla_i D(X_a)\Gamma^{i,j}_{a,b}\nabla_j D(X_b). \tag{28}$$

We can understand the effect of such a regularizer by expressing Eq. (28) in the form

$$\beta\left(g_2|\langle\nabla D(X)\rangle|^2+\frac{(g_1-g_2)}{N}\langle|\nabla D(X)|^2\rangle\right). \tag{29}$$

The second term discourages sharp gradients from being provided to generator particles, leading to smoother paths to convergence. The first term, directly proportional to g_2, can be seen to discourage the presence of large mean gradients over the ensemble of generator particles. Incorporating this into the same setup used to produce mode collapse in Fig. 4 and repeating the procedure [39], we now observe convergence instead (Fig. 8).

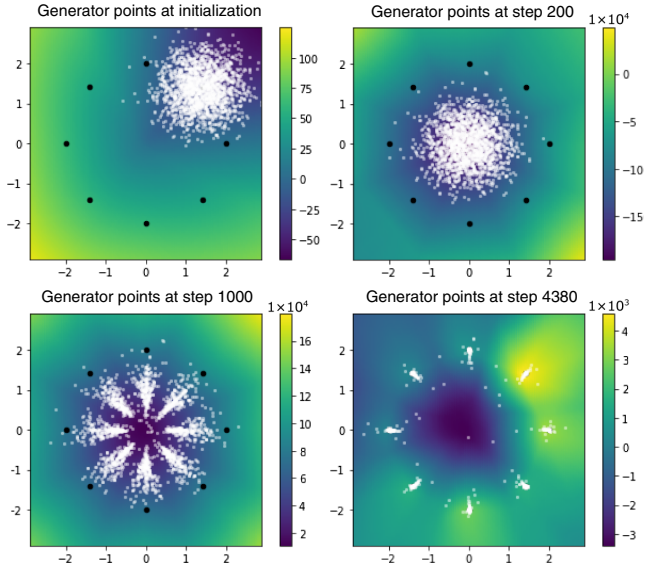

FIG. 8. Model GAN dynamics with significant off-diagonal NTK values ($g_2/g_1 = 1/5$) converge under regularization. Model GAN dynamics is shown over time, taking $g_2/g_1 = 1/5$ and including a regularizer [Eq. (28), with $\beta = 100$]. Despite that without the regularizer the system oscillates from mode to mode (Fig. 4), now particles converge evenly and steadily to the eight modes.

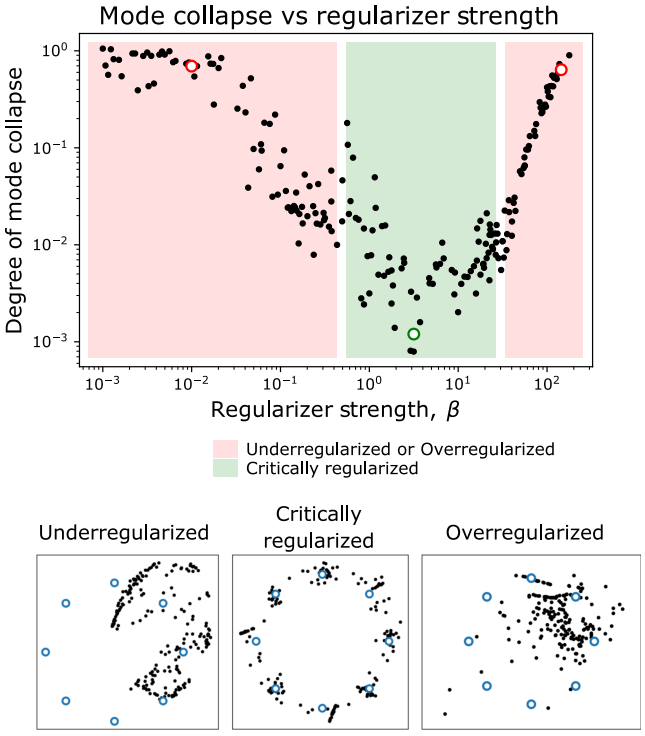

FIG. 9. Critical regularization mitigates mode collapse within a model GAN. Upper: each point represents an experiment using a single hidden-layer ReLU discriminator, halted after 1600 iterations (similar to the experiments of Sec. IV). 200 generator points are used, with $g_2/g_1 = 0.275$ and $n_{\text{disc}} = 5$, placing the system within the regime of mode collapse. The regularizer, which plays a role similar to that of a damping term, results in a reduction of mode collapse. We observe three regimes analogous to underdamped, critically damped, and overdamped dynamics. Lower: snapshots of sampled generator particle configurations for the three circled points in the upper diagram. Note that only the critically regularized example has converged.

Within our setup, we can experiment with the regularization parameter β. Running model GAN experiments using different β values, we note regions corresponding to underregularization and overregularization, and an intermediate regime of critical regularization. In this regime, convergence is most efficiently achieved (Fig. 9).

VI. DISCUSSION

In this paper, we consider a model GAN system constructed by incorporating limiting features present within real GANs. The generator inputs are taken to be sampled uniformly from a sphere of high dimension. Additionally, the generator is assumed to be of infinite width and to have a static NTK such that given inputs of fixed magnitude, the NTK is a function only of their dot product [$\Gamma(z, z') = \Gamma(z \cdot z')$, as is the case in infinite-width ReLU networks]. We also modify the training procedure, using a single fixed mini batch of generator seeds throughout training. Under these approximations, the outputs of an infinite-width generator may be represented as a cloud of particles, whose velocities are coupled through the generator network's NTK. Further, due to the nature of the assumed NTK and the high-dimensional inputs, we argue that this coarse-grained NTK may be characterized using just two values.

Despite the simplicity of our model GAN system, we observe that it is able to exhibit the defining symptoms of mode collapse—generator outputs fail to become diverse. Indeed, the simplified particle-based setting allows for physical interpretation of the phenomenon through competition between the local gradient experienced by each particle and the average discriminator gradient experienced by the cloud as a whole. When the latter dominates over the former, the particle cloud fails to split and hence cannot cover a target diversity of modes. From this physically motivated effective description, we are able to connect the ratio of the two effective NTK values to the occurrence or avoidance of mode collapse.

Because the generator NTK values within the model GAN setup can be easily modified, our framework makes it possible to study learning dynamics over a broad range of generators. Using simple discriminators with a single hidden layer, we investigated the onset of mode collapse as a function of the NTK parameters and the relative training rates of the discriminator and generator. We were able to identify power-law and exponential relationships, and explain their presence by drawing a connection to the frequency principle; frequency-dependent learning rates alone suffice to explain the shape of the transition boundary. To our knowledge, this is the first time that the principle has been observed in the context of GANs. As a consequence, for a given NTK matrix, mode collapse can be avoided by allowing sufficient time for the discriminator to learn finer features characterizing a multimodal target distribution.

Would it be possible to reduce the training time while avoiding mode collapse? We have experimented with a regularizer designed to reduce mode collapse in real GANs, the effect of which is to dampen the velocity of the generator parameters during training. Despite the fact that our model contains no generator parameters, and focuses instead on the dynamics of the outputs, we show how it is possible to adapt such a regularizer to our particle-based setting. We demonstrate that, in our effective model too, such a regularizer can encourage smooth paths to convergence. Importantly, an analogy to a damped oscillator, made clear through examples, enables us to identify regimes analogous to overdamping, underdamping, and critical damping. Intermediate regularization strengths would allow most efficient convergence, suggesting critical regularization as a potential means to cure mode collapse and shorten training time.

In our experiments, we used a two-dimensional dataset of eight Gaussians. This simplified the model GAN dynamics, and maximized the clarity of our theoretical

insights. To confirm these insights and our model's validity in a more realistic, much higher-dimensional setting, we perform further experiments analogous to those of Sec. IV using a target distribution of MNIST data (784 dimensional images of hand-written digits). As described in Appendix I, the central findings of our study using the simple dataset indeed replicate when using the high-dimensional MNIST data. In particular, this is true of those results regarding the effect of the relative discriminator learning rate and the off-diagonal to diagonal ratio of NTK values (g_2/g_1) on mode collapse.

The problem of understanding GAN convergence is complex. By essentializing key features of real GANs, we have probed GAN failure in a more physically interpretable setting, which allows for extensive experimentation. However, the model's assumptions also suggest directions of future work in refining the model by incorporating deviations from these limiting approximations, and in mapping the lessons learned to more realistic GAN settings.

We have, for instance, assumed a time-independent NTK with uniform values throughout data space. In reality, however, for networks of finite widths, the NTK evolves throughout training. Indeed, such dynamic corrections may be studied order by order (in $1/$network width) [19] and incorporated into a more complete analysis. An NTK function which develops spatial features during training [that is, an NTK defined in data space, $\Gamma(X, X')$] might yield dynamics showing closer parallels to flocking, in which individual birds look at spatial neighbors to update their velocities [30,40].

We have also replaced the distribution of NTK values [comprising the evaluations of $\Gamma(z, z)$ and $\Gamma(z \neq z')$ within a mini batch] with just two numbers, g_1 and g_2. By including some variance in the diagonal and off-diagonal NTK values, as is present in realistic settings of finite latent-space dimensions (see, for example, Fig. 2), future work might broaden the scope of the noted universality.

Finally, we have obtained our results using a modified training algorithm in which only one mini batch of seeds is used throughout training. In order to extrapolate the lessons learned to a more realistic setting, we would like to better understand the implications of our results for contexts in which mini batches of seeds $\{z\}$ are continually resampled.

The data depicted in Appendix I, as well as a notebook to aid analysis, can be found at Ref. [41].

ACKNOWLEDGMENTS

During the preparation of this work, S. D. was supported by funds from the Bhaumik Institute for Theoretical Physics at UCLA. This work used computational and storage services associated with the Hoffman2 Cluster hosted by the UCLA Institute for Digital Research and Education. S. W. is grateful for support from an NSF CAREER Award (Grant No. PHY-2146581).

APPENDIX A: NTK SWEEP EXPERIMENTAL DETAILS

The experiments of Sec. IV are performed using the following protocol.

(i) Generator: A collection of 200 two-dimensional points initialized as a Gaussian centered at $(\sqrt{2}, \sqrt{2})$, with standard deviation 0.1.

(ii) Discriminator: A single hidden-layer neural network of width 2048:

$$D(x) = \sqrt{\frac{2}{\text{width}}} a_i \sigma(w_i^j x_j + b_i). \tag{A1}$$

Experiments were run using both ReLU and Tanh activation functions. At initialization, we take $a_i, w_i^j \sim \mathcal{N}(0, \sigma^2 = 1)$ and $b_i \sim \mathcal{N}(0, \sigma^2 = 9)$.

To understand the relationship between the generator NTK and the training rate, we vary both the discriminator learning rate α_D and the discriminator updates per iteration n_{disc} (Algorithm 1). We then examine the degree of mode collapse after some fixed number of training iterations.

The results of experiments which sweep over n_{disc} are described in the main text (Sec. IV), and the results of those varying α_D are given in Sec. A 1. Both reflect the same pattern: generally, for larger g_2/g_1, the discriminator requires more "time" (larger α_D or greater n_{disc}) in order for the adversarial dynamics to overcome mode collapse. In addition, the mode-collapse transition boundaries for ReLU and Tanh discriminators are, respectively, power law and exponential.

Our conclusions concerning the influence of relative learning rates and training iterations on the convergence of GANs do have some grounding in empirical and theoretical works. Notably, the paper which introduced Wasserstein GANs [7] also used the strategy of applying multiple discriminator iterations for each generator iteration ($n_{\text{disc}} = 5$) and demonstrated its efficacy. In a separate paper [42] on theoretical guarantees of the convergence of GANs, the authors proposed a so-called two-timescale update rule, in which the discriminator and generator were given individual learning rates. Under mild assumptions, the authors were able to prove convergence when the discriminator's learning rate was much higher than that of the generator.

In our experiments, as we vary g_2/g_1, we take care to control for the overall effect that the NTK has on total velocity. For example, if all points were initialized at the same location, X, then their velocities would obey

$$\begin{aligned}\frac{dX_a}{dt} &= \alpha_G \frac{1}{N} \sum_j \Gamma_{a,b} \nabla D(X_b) \\ &= \alpha_G \frac{\nabla D(X)}{N} [g_1 + g_2(N-1)].\end{aligned}$$

To control for this effect, as we vary g_2/g_1 we maintain $g_1 + g_2(N-1) = \text{const}$. We take $g_1 + g_2(N-1) = N$, so that $g_2/g_1 = 0 \Rightarrow g_2 = 0, g_1 = N$, and $g_2/g_1 = 1 \Rightarrow g_2 = g_1 = 1$. Since in our experiments, points are initialized in a tight distribution (with $\sigma = 0.1$), we believe this allows us to meaningfully compare the effect of different generator NTKs.

The NTK values written in Eq. (21) can be thought of as elements of an NTK Gram matrix [26]. Our choice of (g_1, g_2) normalization is then equivalent to fixing the eigenvalue of the constant mode to $\lambda_{\text{const}} = N$ for all g_2/g_1. All other eigenvalues are then equal to

$$\frac{N}{N-1}\left(\frac{N}{(N-1)g_2/g_1 + 1} - 1\right).$$

From this perspective, as g_2/g_1 grows, the constant mode of the NTK Gram matrix dominates. The effect of a nonzero constant mode (which, using our language, is equivalent to a nonzero g_2/g_1) was examined in Ref. [26]. In their paper, the authors proposed creative interventions (including specific normalizations) which allowed them to remove the constant NTK contribution for orthogonal inputs. Effectively, these interventions set g_2/g_1 to 0. Their results are in agreement with our interpretation of the role of g_2/g_1. Indeed it was found that by removing the constant component, mode collapse could be successfully avoided. Conversely, if this constant term of the NTK did dominate, the resulting networks were prone to mode collapse [26].

1. Discriminator learning rate experiments

Rather than varying n_{disc}, we run experiments which vary the training rate of the discriminator learning rate α_D.

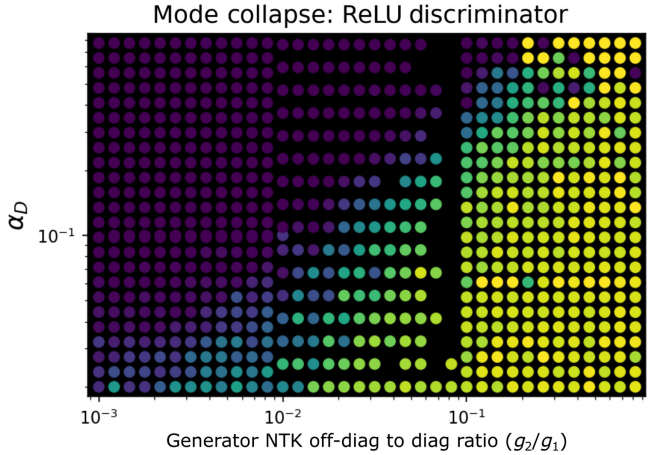

FIG. 10. A power-law mode collapse threshold is found for ReLU discriminators. Mode collapse is depicted as a function of α_D and the ratio g_2/g_1 using a ReLU discriminator. Mode collapse data are averaged over the iterations, $4000 \pm n \times 20$, with $n = 0, 1, 2, 3$. On the log-log plot a linear threshold is observed.

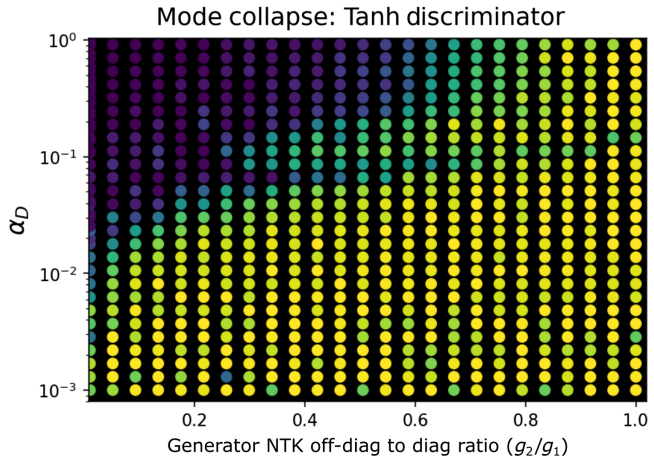

FIG. 11. An exponential mode collapse threshold is found when using Tanh discriminators. Mode collapse as shown as a function of discriminator learning rate α_D and g_2/g_1. As in Fig. 10, mode collapse is averaged about iteration 4000. On a log-linear plot, a broadly linear threshold is observed, indicating an exponential transition.

Apart from this difference, the experiments performed are identical to those of Sec. IV.

Using a ReLU discriminator, a roughly power-law transition boundary is found (shown in Fig. 10). A Tanh discriminator shows an exponential boundary (shown in Fig. 11). As in the case of sweeps over n_{disc}, this matches the expected frequency-dependent learning rate $\gamma(k)$ of the respective networks [36–38].

APPENDIX B: PRECISION SCALING ARGUMENTS

The correspondence between ReLU networks and power-law boundaries (Fig. 5), and between Tanh networks and exponential boundaries (Fig. 12), can be interpreted by considering the spatial precision required for a discriminator to split apart a collection of particles uniformly distributed within a one-dimension region. In Fig. 13, we depict a hypothetical discriminator [defined by $D(x) = c|x|$] and a distribution of generator points uniformly distributed within $[-pl, (1-p)l]$ (depicted as a shaded region). The placement of discriminator's minimum with respect to the center of the generator distribution is determined by p, with $p = 1/2$ placing its minimum directly at the center, and $p = 0$ completely shifting the distribution to the right side of $D(x)$.

Under this setup, the velocities of the points to the left and right of the minimum of $D(x)$ will be

$$v_L = c\left(-\frac{(g_1 - g_2)}{N} + g_2(1-2p)\right)$$

and

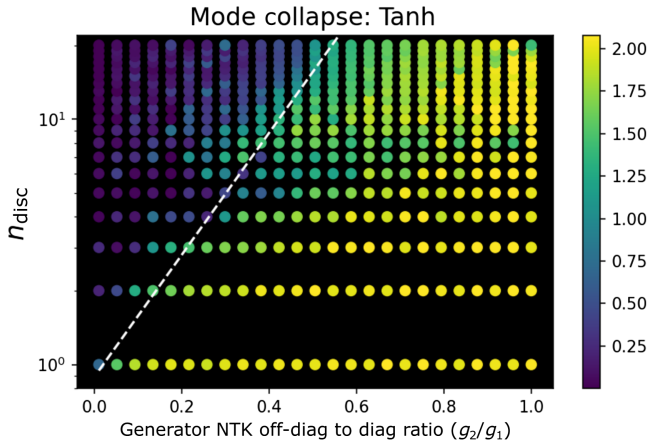

FIG. 12. Phase diagram using a Tanh discriminator. Results are depicted as in Fig. 5, although experimental results are taken after 2500 training iterations. A dashed line is fit to the transition, highlighting a roughly exponential phase boundary, and differing from the power-law boundary observed in Fig. 5.

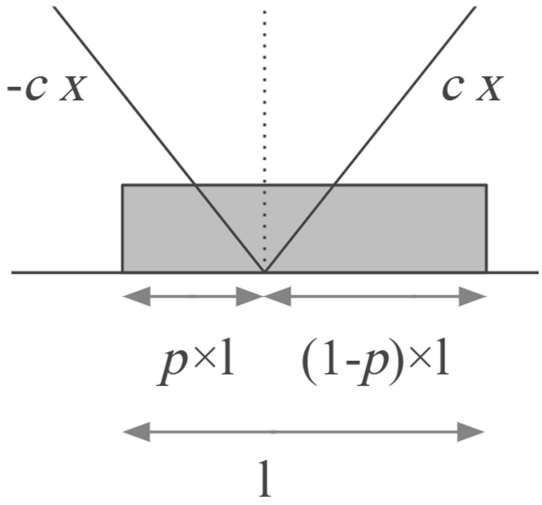

FIG. 13. A schematic depicts a hypothetical discriminator and generator distribution. The generator's uniform distribution of length l, depicted shifted a distance $pl/2$, $p \in [0, 1]$, to the right of the origin. The discriminator of the form $c|x|$ is superimposed.

$$v_R = c\left(\frac{(g_1 - g_2)}{N} + g_2(1 - 2p)\right).$$

These velocities satisfy

$$v_L < 0 \Rightarrow p > \frac{1}{2} - \frac{g_1 - g_2}{2g_2 N},$$

$$v_R > 0 \Rightarrow p < \frac{1}{2} + \frac{g_1 - g_2}{2g_2 N}.$$

In order to "split" the points, and ensure that the particles on each side have opposing velocities, we require

$$\left(\frac{1}{2} - \frac{g_1}{2Ng_2} + \frac{1}{2N}\right) < p < \left(\frac{1}{2} + \frac{g_1}{2Ng_2} - \frac{1}{2N}\right).$$

This range of p indicates that for the discriminator to be able to split apart the distribution, we require the discriminator's minimum to be near the center of the generator distribution, with a *spatial precision* of order

$$l\frac{g_1 - g_2}{Ng_2}.$$

Here if we take g_1/g_2 to be large, then the relevant frequency corresponding to this spacial precision is roughly

$$k \sim \frac{Ng_2}{lg_1}. \tag{B1}$$

APPENDIX C: SUPPORTING THE F-PRINCIPLE MECHANISM

Analogous to the mode collapse experiments using a ReLU discriminator (Sec. IV, and in Fig. 14), experiments were also performed employing Tanh discriminators (Figs. 12 and 15). Here, a roughly exponential phase boundary was found, appearing to match the predicted exponential frequency-dependent learning rate $\gamma(k)$ [38].

Our physically motivated mechanism for the transition (described in Sec. IV B) makes use of the so-called frequency principle within neural networks to explain the shape of the phase boundary. In light of the stark contrast between the shapes of the ReLU and Tanh boundaries (shown in Figs. 5 and 13), which match the differences in their respective frequency learning rates [38], this connection appears very plausible.

We would like, however, to ensure that such a frequency relationship is *sufficient* on its own to create such power-law and exponential phase boundaries, since it is conceivable that some other property of the networks is responsible.

We note that our discriminators are very wide networks with a single hidden layer. In this large-width limit, it is expected to be approximately linear in parameters during training [22]. Additionally, the networks in question are known to obey a given frequency principle. We therefore define a new discriminator which has these precise properties alone and rerun the same experiment to observe the resulting phase boundary. If the same power-law and exponential phase boundaries are found, we can be much more confident in this connection.

We define

$$D(x) = \sum_k D_k(x), \tag{C1}$$

$$D_k(x) = w_k^{(1)} \sin(k \cdot x) + w_k^{(2)} \cos(k \cdot x), \tag{C2}$$

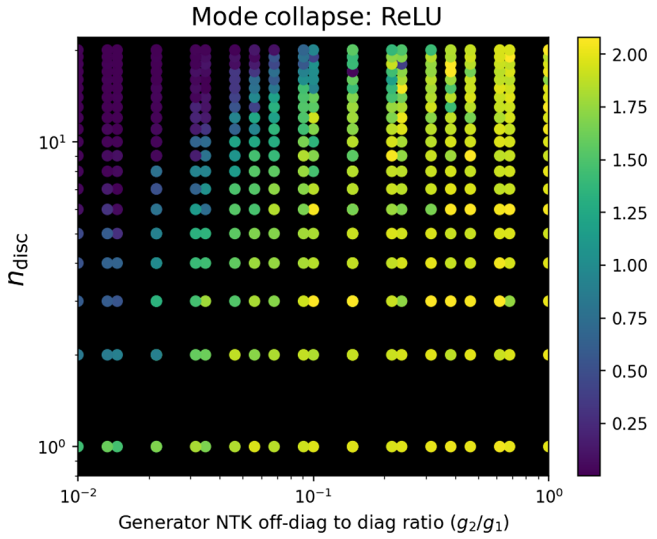

FIG. 14. Modifying the target distribution produces a similar diagram using a wide ReLU discriminator. We replicate the experiment of Fig. 5, but with a modified mixture of Gaussians (here we take $\sigma = 0.3$, rather than the original $\sigma = 0.02$). Despite the modified target distribution, we see the same broadly power-law behavior of the transition.

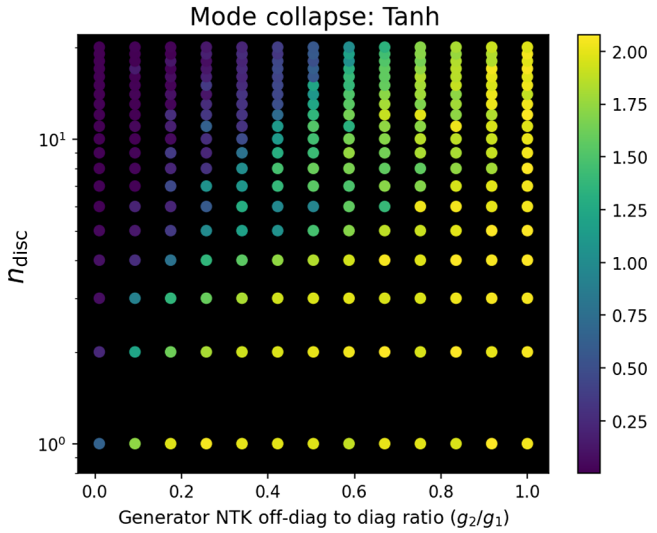

FIG. 15. A modified target distribution yields a similar transition boundary. A replication of the experiment of Fig. 12, still using a mixture of eight Gaussians as the target distribution, but with standard deviations of more than 10 times their original values ($\sigma = 0.3$, as opposed to 0.02). Using the same wide Tanh discriminator, we see the same exponential-like behavior of the transition, indicating robustness with respect to details of the target distribution.

where $k = (k_1, k_2)$ and k_i are taken from 25 values of equal logarithmic spacing from [0.01, 20], as well as the negatives of these values. $w_k^{(i)}$ are the weights of the model.

During training, we follow the routine of Algorithm 2, a modification of Algorithm 1, in which each $w_k^{(i)}$ is updated with a rate proportional to the value of a function $\gamma(k)$. We then plug in power-law and exponential $\gamma(k)$ functions by hand and run the same experiments performed in Sec. IV. The power-law and exponential $\gamma(k)$ functions are defined below [43]:

$$\gamma_{\text{pow}}(k) = \min(10^3, |k|^{-3}), \tag{C3}$$

$$\gamma_{\text{exp}}(k) = 668.8 \exp(-2.05 \cdot |k|). \tag{C4}$$

Our new routine essentializes the properties of the wide ReLU and Tanh discriminators by being linear in the parameters and explicitly learning frequency k features with a rate $\gamma(k)$.

The results of Figs. 16 and 17 show a very clear phase boundary having precisely the power-law and exponential behavior, respectively. This indicates that a frequency-dependent learning rate is sufficient to produce the type of phase boundary we previously observed and lends credence to the connection drawn between the onset of mode collapse and the frequency principle of the discriminator network.

We do, however, emphasize that our simple explanation of the threshold shape (Sec. IV B) is likely incomplete. In particular, the assumption of $g_1/g_2 \gg 1$ breaks down within Fig. 17, and yet the depicted transition remains

Algorithm 2. The model GAN training algorithm, with a Fourier discriminator and a frequency-dependent learning rate.

for iteration number **do**
  for n_{disc} **do**
    i Sample N data points,$\{x_i\}$ from the eight-Gaussian distribution.
    ii Compute

$$\mathcal{L}^{(N)} = \frac{1}{N}\sum_{a=1}^{N} D(X_a) - \frac{1}{N}\sum_{i=1}^{N} D(x_i) + \frac{\lambda}{2}\sum_k [(w_k^{(1)})^2 + (w_k^{(2)})^2]$$

    and update discriminator parameters by descending its stochastic gradient

$$w_k^{(i)} \leftarrow w_k^{(i)} + \alpha_D \gamma(k) \nabla_{w_k^{(i)}} \mathcal{L}_D^{(N)}$$

  end for
  iii update X_a according to Eq. (16)

$$X_a \leftarrow X_a + \alpha_G \frac{1}{N}\sum_b^N \Gamma_{a,b} \nabla_x D(X_b)$$

end for

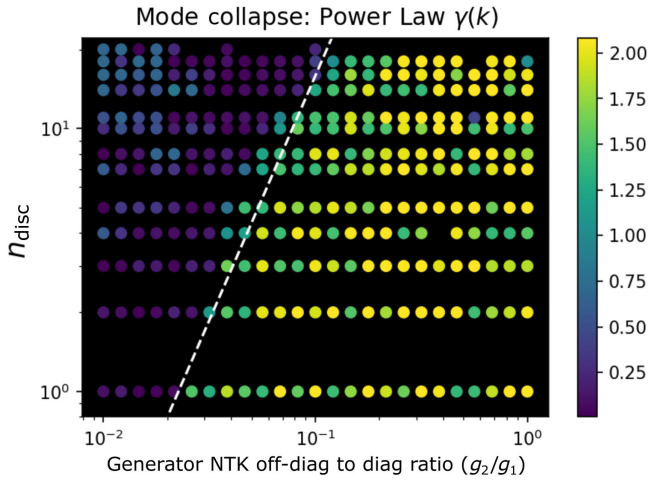

FIG. 16. Fourier discriminators [Eq. (C1)] with power law $\gamma(k)$ have a power-law mode-collapse transition. A scatter plot depicts the transition for a power law $\gamma(k)$ after 3000 steps. Brighter points indicate mode collapse, and darker points indicate convergence. An extremely clear power-law boundary is found here, with a slope of ≈4.90.

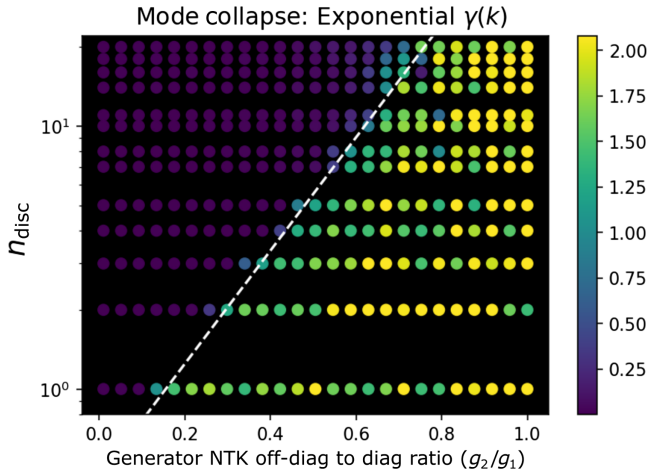

FIG. 17. Fourier discriminators [Eq. (C1)] with exponential $\gamma(k)$ show an exponential mode-collapse transition. A scatter plot shows the transition for an exponential $\gamma(k)$ after 2000 steps. Brighter points indicate mode collapse, and darker points indicate convergence. Again, a clear exponential boundary is found here, having a slope of ≈1.86.

essentially linear (exponential) even to $g_2/g_1 \approx 0.7$. Rather, our description outlines a plausible causal connection, from which a more general explanation might be obtained.

APPENDIX D: GENERATOR DISTRIBUTION ACROSS THE TRANSITION

To visualize the behavior of the generator points through the transition, here we plot the generator distributions for different g_2/g_1 values given a fixed n_{disc}. This uses a ReLU discriminator and the outputs of the experiment performed in Sec. IV.

Taking $n_{\text{disc}} = 6$, the transition here occurs roughly at $g_2/g_1 = 0.06$ (see the transition depicted in Fig. 5). We therefore show plots from below and above this value of g_2/g_1 (Fig. 18).

The distribution of generator particles across the mode collapse phase boundary can also be seen through the average (Euclidean) distance to the nearest mode, shown in Fig. 19. Below the transition, points are tightly focused, oscillate from mode to mode, and are therefore relatively close to the modes. Far above the transition, the distance to the nearest mode is very small; however, this is now due to convergence. Between these two phases, the particles have spread apart. They have begun the process

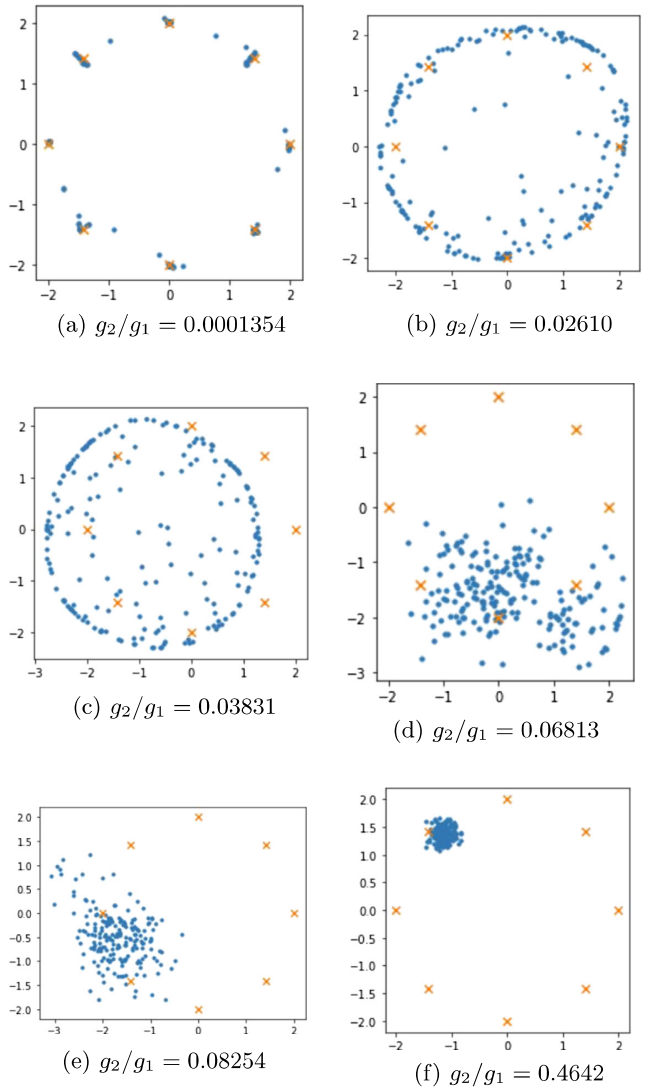

FIG. 18. Generator particles overcome mode collapse and converge below the g_2/g_1 boundary. Generator outputs are shown after 3000 steps. Note the full convergence for small g_2/g_1, while for $g_2/g_1 > 0.06$ the generator fails to converge.

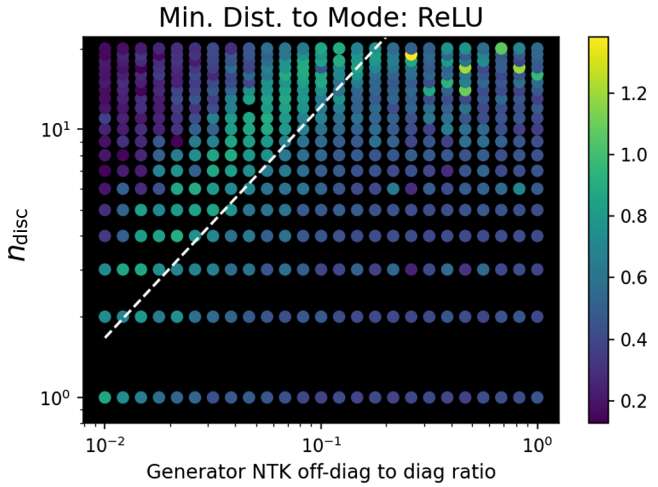

FIG. 19. Euclidean distance to nearest mode drops above (convergence) and below (mode collapse) the mode-collapse transition. Above the phase boundary, points are close to modes due to convergence. Below the transition, points are close to modes due to mode collapse. We observe an increase near the phase transition, indicating that the initially tight clusters of particles have broken apart and the process of convergence has begun. This behavior is somewhat apparent in the log-likelihood plotted in Fig. 6.

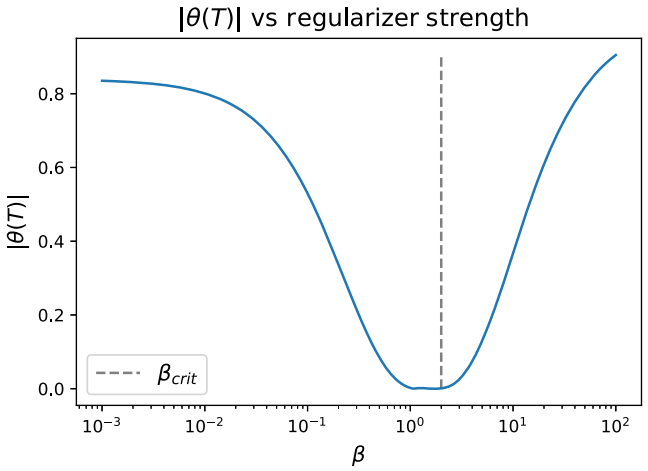

FIG. 20. Critical regularization is demonstrated to result in convergence [$\theta(T) = 0$] within a Dirac GAN. A plot showing the distance after a fixed time, between the Dirac GAN's generator output θ and its equilibrium point 0. The system is initialized at $\theta = 1$, $\phi = 0$, and halted at $T = 10$. The critical regularization value $\beta = 2$ is indicated by a vertical dashed line. Note the overdamped, underdamped, and critically damped regions bear a striking resemblance to the three regimes identified in Fig. 9.

of convergence, and therefore have a larger distance to the nearest mode.

APPENDIX E: REGULARIZATION AND CRITICAL DAMPING

To understand the physical meaning of the regularizer implemented in Sec. V, we can consider its effect on a so-called Dirac GAN [44]. Here, our generator's implicit distribution is simply a Dirac δ focused at θ, with an output given by $G_\theta(z) = \theta$, a data distribution focused at 0, $\delta(x)$, and a discriminator defined by $D_\phi(X) = \phi \cdot X$. In this system, equilibrium would correspond to the point $\phi = \theta = 0$.

The regularizer in this setup then takes the form

$$\beta|\nabla_\theta \phi \cdot \theta|^2/2 = \beta\phi^2/2.$$

In a simultaneous descent-ascent setup, we find that

$$\dot{\theta} = \nabla_\theta D_\phi(\theta) = \phi, \tag{E1}$$

$$\dot{\phi} = -\nabla_\phi[D_\phi(\theta) + \beta\phi^2/2] = -\theta - \beta\phi. \tag{E2}$$

Diagonalizing, we obtain the eigenvalues of the dynamical matrix: $(\beta \pm \sqrt{\beta^2 - 4})/2$, giving us critical damping at $\beta = 2$.

Indeed, if we initialize such a system from $(\theta, \phi) = (1, 0)$ for different β values, and observe the value of $|\theta|$ after a set time T (here we used $T = 10$), we obtain Fig. 20, showing a similar behavior to that found in Fig. 9.

APPENDIX F: MODE COLLAPSE AND REGULARIZATION FOR TANH DISCRIMINATORS

Using a Tanh discriminator, we compute the degree of mode collapse as regularizer strength β is varied. As in Fig. 9, regimes of over, under, and critical regularization are found (Fig. 21).

APPENDIX G: NTK EVOLUTION DURING TRAINING

In the infinite-width limit, the NTK remains fixed during training [18]. An example is shown in Fig. 22, where despite the convergence of a large-width generator's outputs to the target distribution, its NTK values remain nearly constant. This reflects the assumption we have made in using constant values for g_1 and g_2 throughout training.

In the upper plot of Fig. 22 is shown the evolution of generator outputs during training. At each time slice, we compute the NTK for each pair of inputs and find that they are very nearly proportional to a $d \times d = 2 \times 2$ identity matrix [reflecting the $\delta_{i,j}$ in Eq. (20)]. Below, three histograms show the distributions of NTK magnitudes at each time slice. The two peaks hardly vary and correspond to the values of g_2 (at $\approx$1.1) and g_1 ($\approx$4.5) used in Eq. (20).

During training, we use a discriminator and a generator both with a single hidden layer and both using ReLU activations. The generator, expressed,

$$G^i(x) = \sqrt{\frac{2}{\text{width}}} a_i^j \sigma(w_j^k x_k + b_j),$$

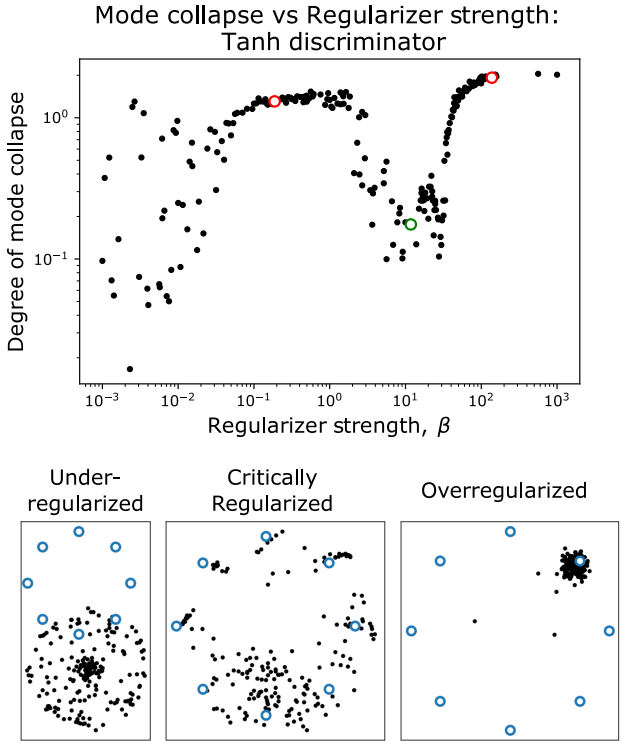

FIG. 21. Regularization regimes are observed using a Tanh discriminator. A plot of the mode collapse present within a model GAN when trained using a regularizer of strength β and a Tanh discriminator. Similar to Fig. 9, in the upper plot each point represents an experiment halted after 2000 iterations, using a single hidden-layer Tanh discriminator. Here, g_2/g_1 is set to $1/5$ and 200 points are used. A clear dip in mode collapse is visible about $\beta \sim 10$. For vanishing β values, we observe volatility in mode collapse. Samples from this region are oscillatory, and may oscillate into and out of a more symmetric distribution with respect to the modes. The drop in volatility as β is increased reflects the regularizer's influence in encouraging smooth paths to convergence. Sampled generator particle configurations are shown for each of the three circled points, corresponding to under-regularized, critically regularized, and overregularized regions.

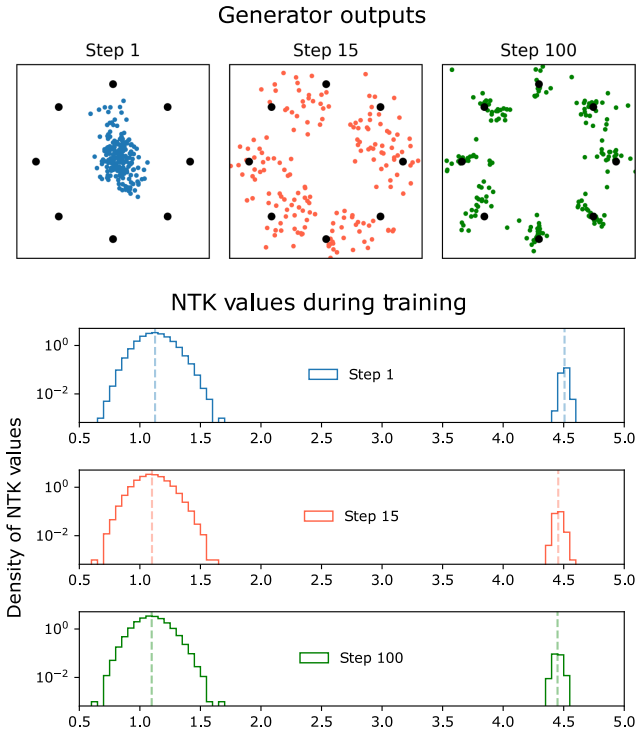

FIG. 22. The NTK values of large-width generators remain approximately fixed during training, despite the convergence of generator outputs. A single set of seeds is used to train a large-width generator (hidden-layer width $= 2^{16}$). Generator outputs are shown at three time slices (above), and the corresponding NTK magnitudes are shown in histograms (below). The medians of the histogram's two peaks are indicated by vertical dashed lines, and roughly correspond to the values of g_1 and g_2 used in the effective NTK of Eq. (20).

has a hidden-layer width of 2^{16} and parameters initialized according to $w_i^j \sim \mathcal{N}(0, \sigma^2 \approx 0.046)$, $b_i \sim \mathcal{N}(0, \sigma^2 = 0)$, $a_i^j \sim \mathcal{N}(0, \sigma^2 \approx 2.25)$. Mirroring the training described in the text, throughout training we use only a single set of 200 seeds sampled from a unit sphere in 256 dimensions. The generator is then trained using RMSProp with a learning rate of 10^{-3}.

The location of the two peaks within each histogram (which determine g_1 and g_2) are a function of the network's architecture and the initialization of its parameters. For instance, replacing the ReLU activation function, which gives $(g_1, g_2) \approx (4.5, 1.1)$, with an Erf activation yields $(g_1, g_2) \approx (10.6, 5.2)$. In general, even when an analytical form of the NTK is available, its value is computed recursively through the layers of the network. Typically no simple closed form is available.

Within certain deep ReLU networks, however, the magnitude of the NTK for orthogonal inputs (corresponding to g_2) can be related to the presence of order or chaos within the network [26]. Using the notation of Refs. [18,26,45], taking inputs from a sphere in n_0 dimensions of radius $\sqrt{n_0}$, taking the lth layer to have width n_l, and $\sigma(x) = \max(0, x)$, we may write

$$\alpha^0(z) = z,$$

$$\tilde{\alpha}^{l>0}(z) \equiv \mu b^{(l-1)} + \sqrt{\frac{1-\mu^2}{n_{l-1}}} W^{(l-1)} \alpha^{l-1}(z),$$

$$\alpha^{l>0}(z) \equiv \sigma(\tilde{\alpha}^l(z)).$$

The output of the neural network function itself is then $f_\theta(z) = \tilde{\alpha}^l(z)$, where the parameters $\theta = \{(W^{(l)})_i^j, b_i^{(l)}\}$ are initialized according to $W_i^j, b_i \sim \mathcal{N}(0, 1)$.

Using six hidden layers of width 2^{12} and $n_0 = 256$, Fig. 23 demonstrates the effect of varying the parameter, $\mu \in [0, 1]$, tuning between networks which are more chaotic

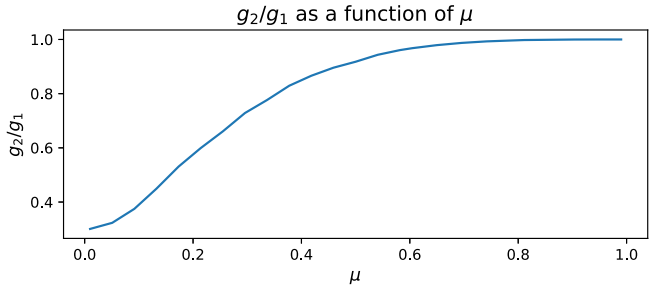

FIG. 23. Tuning between order and chaos changes the value of g_2/g_1. More chaotic networks (small μ) have lower values of g_2/g_1 compared to those in the ordered phase ($\mu \approx 1$).

and those which are ordered [26]. For values of μ near 1, the network is expected to be in an ordered phase, and g_2/g_1 approach unity. Smaller μ values correspond to networks that are more chaotic, and g_2/g_1 is much lower.

APPENDIX H: INDEPENDENCE OF COORDINATES

In the infinite-width limit, the matrix-valued outputs of the NTK (Γ_θ: $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{d\times d}$) become asymptotically diagonal [18]. This phenomenon is shown in Fig. 24, in which off-diagonal to diagonal ratios are plotted as a function of network width. Intuitively, as network widths are expanded, layer outputs are scaled by a factor of $1/\sqrt{N_l}$ (where N_l is the width of the lth hidden layer). Diagonal components of the NTK outputs compound while off-diagonal values drop to zero. The resulting effective NTK

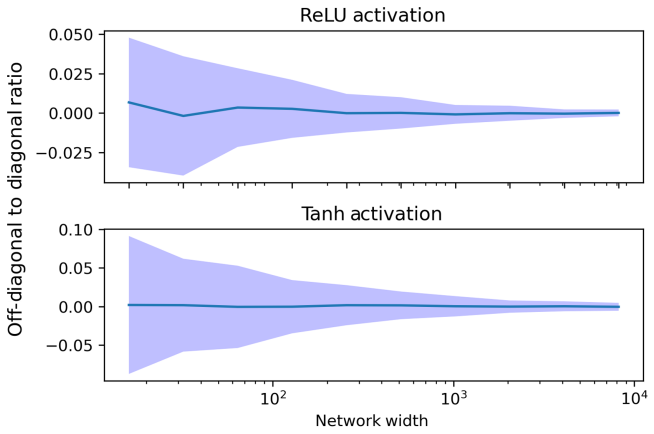

FIG. 24. NTK values are asymptotically diagonal. As the layer width is increased, the off-diagonal matrix elements of NTK outputs die off. Using neural networks with one hidden layer and two outputs, the NTK is computed using two inputs sampled from a unit Gaussian in 1024 dimensions. 100 random initializations of these two-layer networks are done for each width, using either a ReLU (upper panel) or Tanh (lower panel) activation. The mean ratio of off-diagonal NTK values to diagonal NTK values is plotted as a solid line, with the shading above and below indicating one standard deviation.

of Eq. (19) therefore gains an overall $\delta_{i,j}$ for each pair of inputs, (z, z'), decoupling output coordinates in data space.

APPENDIX I: EXPERIMENTS IN HIGH DIMENSIONS

Within our model GAN setup, and using a low-dimensional Gaussian mixture as the target distribution, we have observed the phenomena of convergence and mode collapse. In order to confirm that our framework is capable of modeling this phenomena in more realistic, much higher-dimensional settings, we perform experiments analogous to those of Sec. IV using a target distribution of MNIST data (784 dimensional images of handwritten digits).

In our experiments, we use 100 particles in 784 dimensions, with dynamics coupled through an effective NTK parametrized by the pair (g_1, g_2) following Eq. (20). Our data distribution is taken to be a set of 100 MNIST images. All particles are initialized near a single data point (resembling the initialization of particles in Sec. IV), with Gaussian noise ($\sigma = 1/4$) added to create a high-dimensional particle cloud focused about a single image.

As a discriminator, we use a ReLU neural network having three hidden layers of respective widths 64, 1024, and 64 before outputting a single float. Here, the discriminator is regularized through weight decay as well as a gradient regularizer. Particles are updated using gradient ascent [following precisely Eq. (21)], while the discriminator is updated using the ADAM optimizer [46].

After 10 000 time steps, we analyze the outputs using a mode-collapse metric analogous to that employed in Sec. IV to produce a plot indicating the degree of mode collapse present. In particular, we extend Eq. (22) by defining

$$\text{mode collapse metric} = \log(100) + \sum_i P_i \log P_i. \tag{I1}$$

Here P_i expresses the fraction of particles for which the ith MNIST data point is the nearest using cosine distance. The larger this metric, the less diverse and more distinct outputs are as compared to the target distribution.

The resulting diagram is shown in Fig. 25, and plots of outputs from throughout the diagram are depicted in Figs. 26 and 27. Crucially, even in this high-dimensional setting, we observe similar features noted previously; that is, as g_2/g_1 is decreased, and as the discriminator learning rate is increased, convergence is improved.

It is encouraging that the broad trends produced by experiments in high dimensions match those performed in low dimensions. This signals a broad applicability of our model, demonstrates a general validity of the principles learned, and highlights the utility of interpretable effective models for understanding complex learning dynamics [41].

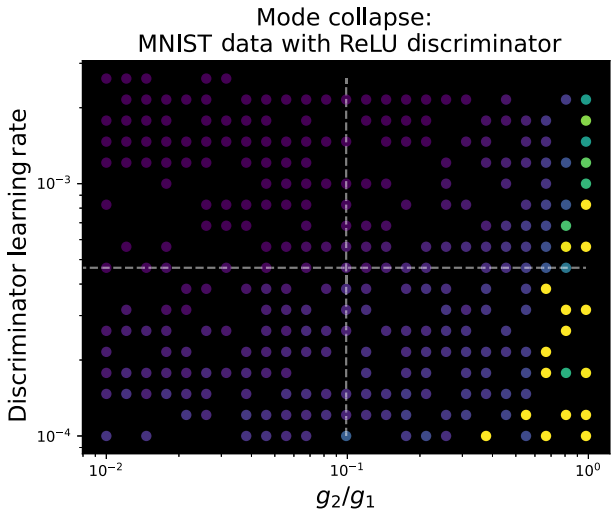

FIG. 25. Mode collapse decreases for lower g_2/g_1 and greater discriminator learning rate in high-dimensional experiments, matching phenomena in low dimensions. The mode-collapse metric [Eq. (I1)] is evaluated on 100 particles in 784-dimensional data space using an MNIST target distribution, with different pairings of (g_2/g_1, discriminator learning rate). Matching the lessons obtained in the two-dimensional cases, greater g_2/g_1 increases mode collapse (corresponding to brighter points), while for larger discriminator learning rates, particle convergence is improved. We later visualize particle distributions using samples at a fixed g_2/g_1 (corresponding to the vertical dashed line) and at a fixed discriminator learning rate (along the horizontal dashed line).

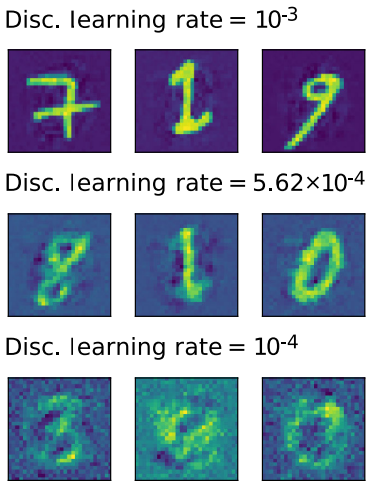

FIG. 26. Increased discriminator learning rate yields better convergence. Within our model GAN trained on MNIST target data, for a fixed g_2/g_1, increasing discriminator learning rate improves convergence. This matches the results obtained for a Gaussian mixture in two dimensions, demonstrating the generality of our previous observations. Samples of three particles in 784 dimensions after 10 000 time steps are shown at three different discriminator learning rates. The value of g_2/g_1 corresponds to that of the vertical dashed line of Fig. 25. As the discriminator learning rate decreases, samples match MNIST data less clearly.

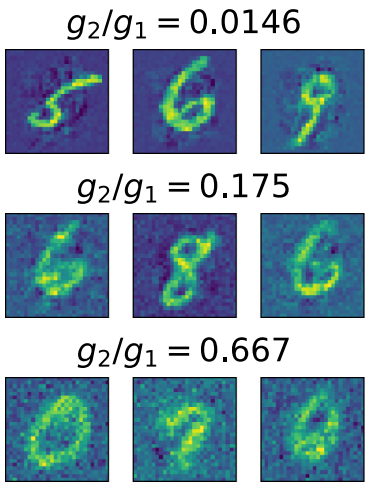

FIG. 27. Increased g_2/g_1 reduces convergence. For a fixed discriminator learning rate (corresponding to the horizontal dashed line of Fig. 25), increasing g_2/g_1 decreases convergence, agreeing with the results obtained for a low-dimensional Gaussian mixture. Samples of three particles in 784 dimensions after 10 000 time steps are shown at three different values of g_2/g_1. As g_2/g_1 is increased, samples less neatly match distinct MNIST data points.

[1] A. Aggarwal, M. Mittal, and G. Battineni, *Generative Adversarial Network: An Overview of Theory and Applications*, Int. J. Inf. Manag. Data Insights **1**, 100004 (2021).

[2] T. Karras, S. Laine, and T. Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks*, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 4217 (2021).

[3] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, *Local Class-Specific and Global Image-Level Generative Adversarial Networks for Semantic-Guided Scene Generation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA*, 2020.

[4] M. Kowalski, S. J. Garbin, V. Estellers, T. Baltrušaitis, M. Johnson, and J. Shotton, *CONFIG: Controllable Neural Face Image Generation*, in *Proceedings of the 16th European Conference Computer Vision—ECCV 2020*, edited by A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm (Springer International Publishing, Cham, Switzerland, 2020), pp. 299–315.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Nets*, in *Advances in Neural Information Processing Systems*, Vol. 27, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (Curran Associates, Inc., Red Hook, NY, 2014).

[6] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016), http://www.deeplearningbook.org.

[7] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein Generative Adversarial Networks*, in *Proceedings of the*

34th International Conference on Machine Learning, ICML'17, Sydney, Australia, 2017, Vol. 70, pp. 214–223.

[8] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, *MMD GAN: Towards Deeper Understanding of Moment Matching Network*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)* (Curran Associates Inc., Red Hook, NY, 2017), pp. 2200–2210.

[9] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, *VEEGAN: Reducing Mode Collapse in GANs Using Implicit Variational Learning*, in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., Red Hook, Long Beach, CA, 2017).

[10] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, *Mode Regularized Generative Adversarial Networks*, in *Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 2017*.

[11] Y. Feng and Y. Tu, *The Inverse Variance-Flatness Relation in Stochastic Gradient Descent Is Critical for Finding Flat Minima*, Proc. Natl. Acad. Sci. U.S.A. **118**, e2015617118 (2021).

[12] Y. Feng and Y. Tu, *Phases of Learning Dynamics in Artificial Neural Networks in the Absence or Presence of Mislabeled Data*, Mach. Learn. Sci. Technol. **2**, 043001 (2021).

[13] Y. Mroueh and T. V. Nguyen, *On the Convergence of Gradient Descent in GANs: MMD GAN as a Gradient Flow*, in *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA* (PMLR, 2021), Vol. 130.

[14] S. Nowozin, B. Cseke, and R. Tomioka, *F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization*, in *Advances in Neural Information Processing Systems*, Vol. 29, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., Red Hook, NY, 2016).

[15] Y. Li, K. Swersky, and R. Zemel, *Generative Moment Matching Networks*, in *Proceedings of the 32nd International Conference on Machine Learning (ICML'15), Lille, France, 2015*, Vol. 37, pp. 1718–1727.

[16] Y. Mroueh, T. Sercu, and A. Raj, *Sobolev descent*, in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan* (PMLR, 2019), Vol. 89.

[17] J.-Y. Franceschi, E. De Bézenac, I. Ayed, M. Chen, S. Lamprier, and P. Gallinari, *A Neural Tangent Kernel Perspective of GANs*, in *Proceedings of the 39th International Conference on Machine Learning (PMLR 2022), Baltimore, Maryland*, edited by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (PMLR, 2022), Vol. 162 pp. 6660–6704.

[18] A. Jacot, F. Gabriel, and C. Hongler, *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)* (Curran Associates Inc., Red Hook, NY, 2018), pp. 8580–8589.

[19] D. A. Roberts, S. Yaida, and B. Hanin, *The Principles of Deep Learning Theory* (Cambridge University Press, Cambridge, England, 2022), https://deeplearningtheory.com, arXiv:2106.10165 [cs.LG].

[20] B. Hanin and M. Nica, *Finite Depth and Width Corrections to the Neural Tangent Kernel*, in *Proceedings of the International Conference on Learning Representations*, 2020.

[21] J. Halverson, A. Maiti, and K. Stoner, *Neural Networks and Quantum Field Theory*, Mach. Learn. Sci. Technol. **2**, 035002 (2021).

[22] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, *Wide Neural Networks of Any Depth Evolve as Linear Models under Gradient Descent*, J. Stat. Mech. (2020) 124002.

[23] Note that a vector selected from $\mathcal{N}^n(0,\sigma^2)$ will have an average squared length of $n\sigma^2$, and the relative standard deviation of this estimate will drop as $\sqrt{2/n}$.

[24] Y. Cho and L. Saul, *Kernel Methods for Deep Learning*, in *Advances in Neural Information Processing Systems*, Vol. 22, edited by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, Inc., Red Hook, NY, 2009).

[25] Two elements, z and z' uniformly selected from an $(n-1)$ sphere of radius $\sqrt{n}$ will have a dot product obeying

$$\cos(\varphi_{z,z'}) \sim \mathcal{N}\left(0, \sigma^2 = \frac{1}{n}\right).$$

[26] A. Jacot, F. Gabriel, F. Ged, and C. Hongler, *Freeze and Chaos: NTK Views on DNN Normalization, Checkerboard and Boundary Artifacts*, in *Proceedings of Mathematical and Scientific Machine Learning (PMLR 2022), Baltimore, Maryland*, edited by B. Dong, Q. Li, L. Wang, and Z.-Q. J. Xu, Vol. 190, pp. 257–270.

[27] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, in *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)* (Ref. [15]), pp. 448–456.

[28] The ReLU and Erf networks have respective weights sampled from $\mathcal{N}(\mu = 0, \sigma^2 \approx 0.42 \text{ and } 1.18)$, and respective biases sampled from $\mathcal{N}(\mu = 0, \sigma^2 \approx 1.17 \text{ and } 11.67)$.

[29] Intuitively considering the NTK values as a matrix defined over particle indices (to which $\delta_{i,j}$ is then applied), we will refer to g_1 as contributing to the NTK's diagonal values, and g_2 as giving its off-diagonal values.

[30] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, *Novel Type of Phase Transition in a System of Self-Driven Particles*, Phys. Rev. Lett. **75**, 1226 (1995).

[31] Our results remain qualitatively the same as long as the standard deviation (size of each Gaussian) is much smaller than the separation between modes. See Figs. 14 and 15 for results with a larger standard deviation ($\sigma = 0.3$).

[32] H. Thanh-Tung and T. Tran, *Catastrophic Forgetting and Mode Collapse in GANs*, in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN 2020), Glasgow, United Kingdom, 2020*, pp. 1–10.

[33] Weights are initialized using a Glorot uniform distribution [34], and biases are initialized at zero.

[34] X. Glorot and Y. Bengio, *Understanding the Difficulty of Training Deep Feedforward Neural Networks*, in *Proceedings of the Thirteenth International Conference on Artificial*

Intelligence and Statistics, Sardinia, Italy, 2010, edited by Y. W. Teh and M. Titterington, Vol. 9, pp. 249–256.

[35] Noting the dynamics described in Eq. (16), we normalize the generator's dynamics by the particle number, setting $g_1 = 2000$, $g_2 = 0$ so that $(1/N)\sum_{a,b}^{N}\Gamma(X_a, X_b) \times \nabla D(X_a) = \nabla D(X_b)$.

[36] Z.-Q. J. Xu, Y. Zhang, and Y. Xiao, *Training Behavior of Deep Neural Network in Frequency Domain*, in *Neural Information Processing*, edited by T. Gedeon, K. W. Wong, and M. Lee (Springer International Publishing, Cham, Switzerland, 2019), pp. 264–274.

[37] R. Basri, D. Jacobs, Y. Kasten, and S. Kritchman, *The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies*, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY, 2019).

[38] Y. Zhang, T. Luo, Z. Ma, and Z.-Q. J. Xu, *A Linear Frequency Principle Model to Understand the Absence of Overfitting in Neural etworks*, Chin. Phys. Lett. **38**, 038701 (2021).

[39] Here, we set $\beta = 100$.

[40] J. Toner and Y. Tu, *Long-Range Order in a Two-Dimensional Dynamical XY Model: How Birds Fly Together*, Phys. Rev. Lett. **75**, 4326 (1995).

[41] https://github.com/durrcommasteven/effective-gan-mnist-data.

[42] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)* (Ref. [8]), pp. 6629–6640.

[43] These functions were obtained by experimenting with the $\gamma(k)$ functions corresponding to real neural networks and finding approximate matches for our sum-of-Fourier-mode discriminators.

[44] L. M. Mescheder, A. Geiger, and S. Nowozin, *Which Training Methods for GANs Do Actually Converge?*, in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA* (Curran Associates Inc., New York, 2017).

[45] In other works, the letter β is used to scale weights and biases. Here, to avoid conflating this variable with the β which scales the gradient regularizer, we instead use μ.

[46] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 2015*, arXiv:1412.6980.