Supplementary Material for: Response Theory via Generative Score Modeling

Ludovico Theo Giorgini, Katherine Deck,Tobias Bischoff, Andre Souza

December 17, 2024

1 Generalized Fluctuation Dissipation Theorem

For a system in statistical equilibrium, the Generalized Fluctuation Dissipation Theorem (GFDT) relates perturbations to system dynamics and the invariant measure associated with the system. Although the perturbations are considered to act only on initial conditions, the resulting theory can also be applied when continuous forcing is used. As such, the GFDT can be used to calculate response functions. These are the appropriate objects to use when predicting expected (relatively) small changes in system dynamics. Applications of the theorem include statistical mechanics, causality relations, and climate change.

1.1 Variable Names and Assumptions

We assume a (finite-dimensional) dynamical system of the form

$$\dot{\vec{u}} = \vec{f}(\vec{u}) + \epsilon\vec{\xi} \tag{1}$$

where $\vec{f}$ is the deterministic part of the dynamics and $\vec{\xi}$ is delta-correlated in time (state-independent) white-noise. We further assume that the dynamical system admits a unique and differentiable steady-state probability distribution $\rho(\vec{u})$ that solves the steady-state Fokker-Plank equation

$$\nabla\cdot\left(\vec{f}\rho - \frac{\epsilon^2}{2}\nabla\rho\right) = 0. \tag{2}$$

In this section, we abuse notation and let subscripts on the vector $\vec{u}$ denote different random variables. For example, $\vec{u}_0$ is a random variable associated with the time $t = 0$, and $\vec{u}_{\mathcal{T}}$ is a random variable associated with $t = \mathcal{T}$.

The transition probability from a past state to a future state is denoted by $\rho(\vec{u}_{\mathcal{T}}|\vec{u}_0)$ and this probability is the solution to the forward evolution of the Fokker-Plank equation starting with initial density $\rho(\vec{u}, 0) = \delta(\vec{u} - \vec{u}_0)$, e.g.

$$\mathcal{L}\bullet \equiv \nabla\cdot\left(\vec{f}\,\bullet - \frac{\epsilon^2}{2}\nabla\bullet\right) \tag{3}$$

$$\rho(\vec{u}_{\mathcal{T}}|\vec{u}_0) \equiv \exp(\mathcal{T}\mathcal{L})\delta(\vec{u} - \vec{u}_0) \tag{4}$$

where $\mathcal{L}$ is the Fokker-Plank operator and $\exp(\mathcal{T}\mathcal{L})$ is the Perron-Frobenius operator at time $\mathcal{T}$, i.e., the operator exponential of $\mathcal{L}$ over timescale $\mathcal{T}$. Thus $\vec{u}_\mathcal{T}$ is correlated with $\vec{u}_0$ over short times. We also let

$$\rho(\vec{u}_\mathcal{T}, \vec{u}_0) \equiv \rho(\vec{u}_\mathcal{T}|\vec{u}_0)\rho(\vec{u}_0) \tag{5}$$

denote the joint probability distribution between $\vec{u}_0$ and $\vec{u}_\mathcal{T}$.

We have the relations

$$\rho(\vec{u}) = \lim_{\mathcal{T}\to\infty} \rho(\vec{u}_\mathcal{T}|\vec{u}_0) = \int \mathrm{d}\vec{u}_0 \rho(\vec{u}|\vec{u}_0)\rho(\vec{u}_0) \tag{6}$$

where the middle equality states that that the invariant measure $\rho(\vec{u})$ is unique and the last equality states that $\rho(\vec{u})$ is an eigenvalue of the Perron-Frobenius operator $\rho(\vec{u}|\vec{u}_0)$ with eigenvalue 1.

An observable g maps our state $\vec{u}$ to a scalar value. This observable can be any (linear/nonlinear) functional of the state $\vec{u}$. We are often concerned with the evolution of different observables of our system.

1.2 Definition of System Perturbations

An operational definition of "system perturbation" (in the text we denoted this as the "Dynamical response") is

1. Choose a fixed (small in magnitude) perturbation vector $\vec{\varepsilon}$
2. Choose an observable g
3. Draw an initial condition drawn from the system steady-state distribution
4. Evolve two trajectories, one with initial condition $\vec{u}_0$ and the other with initial condition $\vec{u}_0 + \vec{\varepsilon}$. Use the same sequence of noise realizations in each case if the system is stochastic.
5. Compute the difference between the observable g in the perturbed and unperturbed trajectory at each moment in time starting from $t = 0$.
6. Repeat 2-4 for many initial conditions drawn from the steady state distribution and average the result.

The above sequence of steps is considered the "left-hand side" of the GFDT.

If we treat the "left-hand side" of the GFDT as the "Langevin" version of GFDT, the "right-hand side" is the "Fokker-Plank" definition followed by approximations. We first consider the average value of an observable of the unperturbed trajectory. As is done in the Dynamical response, we first draw an initial condition from the distribution $\rho(\vec{u}_0)$, then evolve our trajectory forward in time so that the distribution in the end is $\rho(\vec{u}_\mathcal{T}|\vec{u}_0)$. We are interested in the average value of the observable at time $\mathcal{T}$; thus, the observable is evaluated at $g(\vec{u}_\mathcal{T})$. Together, this yields the unperturbed average

$$\int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} \rho(\vec{u}_0)\rho(\vec{u}_\mathcal{T}|\vec{u}_0) g(\vec{u}_\mathcal{T}), \tag{7}$$

which simplifies to

$$\int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} \rho(\vec{u}_0)\rho(\vec{u}_\mathcal{T}|\vec{u}_0)g(\vec{u}_\mathcal{T}) = \int \mathrm{d}\vec{u}_\mathcal{T} \rho(\vec{u}_\mathcal{T})g(\vec{u}_\mathcal{T}) = \langle g \rangle \tag{8}$$

from our assumptions that there is a unique steady-state distribution.

The perturbed version follows from the same logic. As is done in the Dynamical response, we first draw an initial condition from the distribution $\rho(\vec{u}_0)$, then evolve our trajectory forward in time from a perturbed version of our initial condition, $\vec{u}_0 + \vec{\varepsilon}$, so that the distribution in the end is $\rho(\vec{u}_\mathcal{T}|\vec{u}_0 + \vec{\varepsilon})$. Thus, our perturbed trajectory is

$$\langle g \rangle_p = \int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} \rho(\vec{u}_0)\rho(\vec{u}_\mathcal{T}|\vec{u}_0 + \vec{\varepsilon})g(\vec{u}_\mathcal{T}). \tag{9}$$

The difference between the two yields the Fokker-Plank version

$$\langle \delta g \rangle = \langle g \rangle_p - \langle g \rangle. \tag{10}$$

This equation is the starting point for deriving the Generalized Fluctuation Dissipation Theorem.

1.3 Derivation with State-Dependent Perturbations

We will derive the Generalized Fluctuation Dissipation Theorem, in line with Appendix A of (1), and further generalize to the case where the perturbation is allowed to be state-dependent. The average change in an observable g with respect to a state-dependent perturbation $\vec{\varepsilon}(\vec{u}_0)$ to an initial condition $\vec{u}_0$ is given by

$$\langle \delta g_\mathcal{T} \rangle \equiv \int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} g(\vec{u}_\mathcal{T})\rho(\vec{u}_0) \left[\rho(\vec{u}_\mathcal{T}|\vec{u}_0 + \vec{\varepsilon}(\vec{u}_0)) - \rho(\vec{u}_\mathcal{T}|\vec{u}_0)\right], \tag{11}$$

where $\vec{\varepsilon}$ is an infinitesimal of magnitude ε. If we assume that we have a steady state distribution, e.g., $\int \mathrm{d}\vec{u}_0 \rho(\vec{u}_\mathcal{T}|\vec{u}_0)\rho(\vec{u}_0) = \rho(\vec{u}_\mathcal{T})$, we have

$$\begin{aligned}
\int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} g(\vec{u}_\mathcal{T})\rho(\vec{u}_0)\rho(\vec{u}_\mathcal{T}|\vec{u}_0) &= \int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} g(\vec{u}_\mathcal{T})\rho(\vec{u}_0)\rho(\vec{u}_\mathcal{T}|\vec{u}_0) && (12)\\
&= \int \mathrm{d}\vec{u}_\mathcal{T} g(\vec{u}_\mathcal{T})\rho(\vec{u}_\mathcal{T}) && (13)\\
&= \langle g \rangle && (14)
\end{aligned}$$

where the last quantity is the ensemble average of g.

We now focus on simplifying the expression

$$\int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} g(\vec{u}_\mathcal{T})\rho(\vec{u}_0)\rho(\vec{u}_\mathcal{T}|\vec{u}_0 + \vec{\varepsilon}(\vec{u}_0)). \tag{15}$$

First, make the change of variable

$$\vec{v}_0 = \vec{u}_0 + \vec{\varepsilon}(\vec{u}_0) \tag{16}$$

which implies

$$\vec{u}_0 = \vec{v}_0 - \vec{\varepsilon}(\vec{u}_0) \Rightarrow \vec{u}_0 = \vec{v}_0 - \vec{\varepsilon}(\vec{v}_0 - \vec{\varepsilon}(\vec{u}_0)). \tag{17}$$

To first order, we then have

$$\vec{u}_0 = \vec{v}_0 - \vec{\varepsilon}(\vec{v}_0) + \mathcal{O}(\varepsilon^2) \tag{18}$$

with ε denoting the perturbation strength of $\vec{\varepsilon}$. The resulting differential is

$$\mathrm{d}\vec{u}_0 = \det\left(\mathbb{I} - \varepsilon\vec{J}\right)\mathrm{d}\vec{v}_0 \tag{19}$$

where $\varepsilon\vec{J}$ is the Jacobian of $\vec{\varepsilon}$, $\nabla\vec{\varepsilon} = \varepsilon\vec{J}$, and $\mathbb{I}$ is the identity matrix. The expression can be simplified using Jacobi's formula for the derivative of the determinant to yield

$$\det\left(\mathbb{I} - \varepsilon\vec{J}\right) \approx 1 - \ \mathrm{trace}\left(\varepsilon\vec{J}\right) = 1 - \nabla\cdot\vec{\varepsilon} \tag{20}$$

The change of coordinates does not shift the limits of integration, which is over all of space. Thus we have

$$\int \mathrm{d}\vec{u}_0\mathrm{d}\vec{u}_{\mathcal{T}}\, g(\vec{u}_{\mathcal{T}})\rho(\vec{u}_0)\rho(\vec{u}_{\mathcal{T}}|\vec{u}_0 + \vec{\varepsilon}(\vec{u}_0)) \tag{21}$$

$$= \int \mathrm{d}\vec{v}_0\mathrm{d}\vec{u}_{\mathcal{T}}(1 - \nabla\cdot\vec{\varepsilon})g(\vec{u}_{\mathcal{T}})\rho(\vec{v}_0 - \vec{\varepsilon})\rho(\vec{u}_{\mathcal{T}}|\vec{v}_0)) \tag{22}$$

$$\approx \langle g\rangle - \int \mathrm{d}\vec{v}_0\mathrm{d}\vec{u}_{\mathcal{T}}\, g(\vec{u}_{\mathcal{T}})\left(\vec{\varepsilon}\cdot\nabla\ln\rho(\vec{v}_0) + \nabla\cdot\vec{\varepsilon}(\vec{v}_0)\right)\rho(\vec{v}_0, \vec{u}_{\mathcal{T}}) + \mathcal{O}(\varepsilon^2) \tag{23}$$

Our final expression then simplifies to

$$\langle \delta g_{\mathcal{T}}\rangle = -\int \mathrm{d}\vec{u}_0\mathrm{d}\vec{u}_{\mathcal{T}}\, g(\vec{u}_{\mathcal{T}})\left(\vec{\varepsilon}(\vec{u}_0)\cdot\nabla\ln\rho(\vec{u}_0) + \nabla\cdot\vec{\varepsilon}(\vec{u}_0)\right)\rho(\vec{u}_0, \vec{u}_{\mathcal{T}}) \tag{24}$$

where we took the liberty to replace the dummy variable $\vec{v}_0$ with $\vec{u}_0$. In total, we need to calculate the correlation of the function $\vec{\varepsilon}\cdot\nabla\ln\rho + \nabla\cdot\vec{\varepsilon}$ at time 0, with the observable g at time $0 + \mathcal{T}$, for each $\mathcal{T}$ of interest. Under the assumption of a statistically steady state, we observe that each time t of our ensemble of simulations comes from the same steady-state distribution ρ. Thus we can correlate the observable g at time $t + \mathcal{T}$ with $\vec{\varepsilon}\cdot\nabla\ln\rho + \nabla\cdot\vec{\varepsilon}$ at time t.

1.4 Simplifications

In the text, the perturbation $\vec{\varepsilon}$ is not state-dependent, in which case the change in the observable simplifies to

$$\langle \delta g_{\mathcal{T}}\rangle = -\vec{\varepsilon}\cdot\int \mathrm{d}\vec{u}_0\mathrm{d}\vec{u}_{\mathcal{T}}\, g(\vec{u}_{\mathcal{T}})\nabla\ln\rho(\vec{u}_0)\rho(\vec{u}_0, \vec{u}_{\mathcal{T}}). \tag{25}$$

We observe that the vector quantity

$$\vec{R}_g(\mathcal{T}) = \int \mathrm{d}\vec{u}_0 \mathrm{d}\vec{u}_\mathcal{T} g(\vec{u}_\mathcal{T}) \nabla \ln \rho(\vec{u}_0) \rho(\vec{u}_0, \vec{u}_\mathcal{T}) \tag{26}$$

is independent of the perturbation choice $\vec{\varepsilon}$. The typical response function makes several choices for the observable g by choosing a set of observables g_i where $g_i(\vec{u}_0) = \langle \hat{e}_i, \vec{u}_0 \rangle$, so that the set of vectors $\vec{R}_{g_i}(\mathcal{T})$ are combined together into a single response matrix $\mathcal{R}_{ij}$. Once one has access to $\mathcal{R}_{ij}$, the response of any linear functional g to a small perturbation can be computed. The continuous analog of the Response matrix is the Response kernel, where the i, j indices represent different locations in the domain.

1.5 Further Uses of Response Functions

Once one has the Response matrix $\mathcal{R}_{ij}$, it is possible to understand how a system behaves with respect to a continuous forcing by convolving with the response matrix. For example, if

$$\dot{\vec{u}} = \vec{f}(\vec{u}) + \vec{h}(t) + \vec{\xi} \tag{27}$$

$$\dot{\vec{v}} = \vec{f}(\vec{v}) + \vec{\xi} \tag{28}$$

then $\langle u_i - v_i \rangle(t) = \int_0^t d\mathcal{T} \sum_j \mathcal{R}_{ij}(t - \mathcal{T}) h_j(\mathcal{T})$ where $\vec{u}$ and $\vec{v}$ have the same initial condition drawn from the invariant distribution of the $\vec{v}$ system, u_i, v_i, h_i denotes the i'th component of $\vec{u}, \vec{v}, \vec{h}$, respectively, and the same sequence of noise ξ is used in both system. In other words, the ensemble average of the difference between the two dynamical systems is given by convolution with respect to the response function.

1.6 The Response Function Using the Gaussian Approximation

In this section, we derive the response function using the Gaussian approximation presented in Equation (5) of the manuscript. The Gaussian approximation consists of approximating the unperturbed steady-state distribution ρ with a multivariate Gaussian distribution

$$\rho_G(\vec{u}) = \frac{1}{(2\pi)^{N^2/2} (\det \vec{\Sigma})^{1/2}} \exp\left(-\frac{1}{2} \vec{u}^{\mathrm{T}} \vec{\Sigma}^{-1} \vec{u} \right), \tag{29}$$

with $\vec{\Sigma}$ the correlation matrix $\vec{C}(0)$ calculated for $t = 0$. Using Equation (4) of the manuscript, we can write the score function as

$$\vec{s}_G(\vec{u}) = -\vec{\Sigma}^{-1} \vec{u}. \tag{30}$$

The response function then becomes

$$\vec{R}_G(\mathcal{T}) = \langle \vec{u}(\mathcal{T}) \otimes \vec{u}(0) \rangle_\mathcal{J} \vec{\Sigma}^{-1} = \vec{C}(t) \vec{C}^{-1}(0). \tag{31}$$

2 Derivation of the Score Function for Potential Systems

For a stochastic dynamical system with correlated noise,

$$\dot{\vec{u}} = \vec{f}(\vec{u}) + \Sigma^{1/2}\vec{\xi} \tag{32}$$

the corresponding Fokker-Planck equation is

$$\partial_t \rho + \nabla \cdot \left(\vec{f}\rho - \frac{1}{2}\Sigma \nabla \rho \right) = 0. \tag{33}$$

We observe that if $\vec{f} = -\frac{1}{2}\Sigma\nabla V$ for some potential function $V(\vec{u})$ then the solution steady-state Fokker-Planck equation is

$$\rho \propto e^{-V}, \tag{34}$$

from whence the score function is $\vec{s}(\vec{u}) = -\nabla V$. A generalization of this observation is if $\vec{f} = -\frac{1}{2}\Sigma\nabla V + \vec{g}$ where $\vec{g}$ satisfies the following property

$$\nabla \cdot \left(\vec{g} e^{-V} \right) = 0. \tag{35}$$

As a comment, we can always decompose the deterministic component $\vec{f}$ in this manner if we take $V \equiv \ln \rho$ with ρ as the steady state distribution and *define* $\vec{g} \equiv \vec{f} + \frac{1}{2}\Sigma\nabla V$. The gradient of the potential term, ∇V, corresponds to the time-reversible part of the dynamics, and $\vec{g}$ corresponds to the time-irreversible component.

A sufficient condition that can often be satisfied is

$$\nabla \cdot \vec{g} = 0 \text{ and } \nabla V \cdot \vec{g} = 0. \tag{36}$$

The first restriction states that the function $\vec{g}$ can't concentrate probability, and the latter restriction states that $\vec{g}$ can only move the probability on iso-surfaces of the steady state probability density. This equation must be satisfied by all linear systems since $\nabla \cdot \vec{g}$ would be independent of $\vec{u}$ while $\nabla V \cdot \vec{g}$ would be quadratic in $\vec{u}$. The advection terms in the two potential systems of the manuscript satisfied Equation 36.

For example, consider the linear stochastic differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \tag{37}$$

$$= \underbrace{\begin{bmatrix} -0.8 & 0.4 \\ 0.4 & -1.2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{-\frac{1}{2}\Sigma\nabla V} + \underbrace{\begin{bmatrix} -0.2 & 0.6 \\ -0.4 & 0.2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\vec{g}} + \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \tag{38}$$

where the function $\vec{g}$ satisfies Equation 36,

$$V = \begin{bmatrix} u_1 & u_2 \end{bmatrix}\begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 1.2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \text{ and } \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{39}$$

See (2) for examples corresponding to Langevin dynamics.

Similar considerations apply to Stochastic PDEs. We can connect them to the finite-dimensional case through a spatial numerical discretization of the underlying PDE.

2.1 The Modified Allen-Cahn Equation

The potential function for the modified Allen-Cahn system is given by the following integral

$$\mathcal{V}[u] = \frac{2}{\epsilon^2} \int \left[\frac{\alpha}{4}(1-u^2)^2 + \frac{\kappa}{2} \left(|\nabla u|^2 + |\Delta u|^2 \right) \right] \mathrm{d}\vec{x}, \tag{40}$$

where the parameters are defined in the main text. The functions $u = \pm 1$ are global minima of the functional. In steady-state, the probability density (or Gibbs measure) is given by (3)

$$\rho(u) \propto \exp\left(-\mathcal{V}\right), \tag{41}$$

which yields the score function

$$s(u) = -\frac{\delta \mathcal{V}}{\delta u} = \frac{2}{\epsilon^2} \left[\alpha u(1-u^2) + \kappa(\Delta - \Delta^2)u \right]. \tag{42}$$

We comment that the fluctuation-dissipation relation at time $t = 0$ is

$$-\int \mathcal{D}[u] u s(u) \rho(u) = \delta(\vec{x} - \vec{y}), \tag{43}$$

which differs from the discrete version by δ_{ij} to $\delta(\vec{x} - \vec{y})$. Here, we used a path integral notation for the previous equation. The presence of the correlated noise yields dynamics of the form

$$\partial_t u = -\frac{\epsilon^2}{2} \Sigma \frac{\delta \mathcal{V}}{\delta u} + \epsilon \Sigma^{1/2} \xi. \tag{44}$$

Adding in an advection term, as was done in the manuscript,

$$\partial_t u = -\frac{\epsilon^2}{2} \Sigma \frac{\delta \mathcal{V}}{\delta u} - A \partial_x u + \epsilon \Sigma^{1/2} \xi. \tag{45}$$

does not affect the steady-state distribution.

To see that the advection term does not affect the steady-state distribution only requires changing reference frames to one with a velocity A and using the translational invariance of the probability density. Alternatively, one can directly show via the Fokker-Plank equation that the advection term satisfies Equation 36. The reason the advection term vanishes is two-fold: the first is that the trace of the advection operator is zero, the second observation is that the inner product vanishes,

$$\int (A \partial_x u) \frac{\delta \mathcal{V}}{\delta u} \, \mathrm{d}\vec{x} = 0, \tag{46}$$

in a periodic domain for a constant velocity A. For more details on the topic, see (3). The same procedure can be applied to the Ornstein-Uhlenbeck system by replacing the nonlinear term with a linear one. With the Ornstein-Uhlenbeck system, it is possible to write down the score function for all diffusion times analytically.

3 Practical Considerations

Bridging the gap between theories and data requires several practical considerations. The discrete identity

$$\langle u_i s_j(\vec{u}) \rangle = -\delta_{ij} \tag{47}$$

will often not be satisfied due to approximation errors. This identity follows from the calculation

$$\langle u_i s_j(\vec{u}) \rangle = \int_\Omega \mathrm{d}\vec{u} \rho(\vec{u}) u_i s_j(\vec{u}) = \int_\Omega \mathrm{d}\vec{u} \rho(\vec{u}) u_i \partial_{u_j} \ln \rho(\vec{u}) \tag{48}$$

$$= \int_\Omega \mathrm{d}\vec{u} \rho(\vec{u}) u_i \frac{\partial_{u_j} \rho(\vec{u})}{\rho(\vec{u})} = -\int_\Omega \mathrm{d}\vec{u} u_i \partial_{u_j} \rho(\vec{u}) \tag{49}$$

$$= -\int_\Omega \mathrm{d}\vec{u} \left(\partial_{u_j} u_i\right) \rho(\vec{u}) = -\int_\Omega \mathrm{d}\vec{u} \delta_{ij} \rho(\vec{u}) \tag{50}$$

$$= -\delta_{ij} \int_\Omega \mathrm{d}\vec{u} \rho(\vec{u}) = -\delta_{ij} 1 \tag{51}$$

There are three primary sources of approximation error in satisfying this discrete identity:

1. Ensemble averages are often approximated using empirical averages.
2. The score function is approximated using a parameterized model with finite data and training time.
3. The generative score function is the score function of the data convolved with a multivariate Gaussian of small width when using a denoising score loss.

With sufficient data and a sufficiently flexible parametric model, the first two errors are mitigated but not eliminated. The latter error is fundamental to estimating a score function using the denoising score loss but is often negligible compared to the first two errors. Thus, we will focus on the first two sources of approximation error and will return to the third afterward.

3.1 Correction Procedure

In practice, with sufficient data, Equation (47) becomes

$$-\langle u_i s_j(\vec{u}) \rangle \approx -\sum_{\omega=1}^{N} u_i^\omega s_j(\vec{u}^\omega) = \delta_{ij} - \Xi_{ij} \tag{52}$$

where we have approximated the ensemble average with an empirical average with the superscript ω denoting different ensemble members, and the matrix Ξ is a small "error" matrix. The score function at diffusion time $\tau = 0$, used in the manuscript, corrected this issue by using a modified score to satisfy the discrete identity, similar to how the Gaussian approximation satisfies the discrete identity.

We modify the score function by composing it with an additional linear operator to satisfy Equation (47),

$$\vec{\varsigma} \equiv \vec{s}(\mathbb{I} - \Xi)^{-1} \tag{53}$$

This modified score satisfies the discrete identity since

$$-\sum_{\omega=1}^{N} u_i^\omega \varsigma_j(\vec{u}^\omega) = -\left(\sum_{\omega=1}^{N} u_i^\omega s_j(\vec{u}^\omega)\right)(\mathbb{I} - \Xi)^{-1} = (\mathbb{I} - \Xi)(\mathbb{I} - \Xi)^{-1} = \mathbb{I} \tag{54}$$

We could have also used

$$\vec{\varsigma}_2 \equiv (\mathbb{I} - \Xi)^{-1}\vec{s} \text{ or } \vec{\varsigma}_3 \equiv \frac{1}{2}\left(\vec{s}(\mathbb{I} - \Xi)^{-1} + (\mathbb{I} - \Xi)^{-1}\vec{s}\right) \tag{55}$$

as the definition of the modified score, but experimentation yielded little difference between the different choices.

Although we used the corrected score function ς from Equation (53) in all computations in the manuscript, including those involving the analytic score function, it was not strictly necessary for the fidelity of any computations; however, using the correction procedure here, greatly decreased the number of epochs necessary to obtain an improved result over the Gaussian approximation. See Section 5.3.

3.2 Uncorrected Score Results

We justify the statement from the previous section by showing results with the uncorrected score function values. It is worth keeping in mind that the Gaussian approximation that is often used in the literature satisfies the identity Equation (47); thus, the baseline comparison will be more favorable towards the Gaussian model at short times even when compared to the "exact" score function; however, the generative model's performance remains better than the Gaussian counterpart for the nonlinear cases.

We see that in Figure 1 the results are relatively unchanged with respect to the results in the main manuscript with respect to the strongest responses in the system. In the Allen-Cahn system we see that the analytic score function underperforms with respect to the generative model, which we justify as a combination of two effects: the generative score is more tailored to the data-distribution and the ensemble averaging yields errors in computing the response function.

We quantify the error in Figure 2. We see that the generative model has a higher error than the Analytic error in the L^2 sense for the Ornstein-Uhlenbeck

and Allen-Cahn systems. The generative model has enhanced performance with respect to the Gaussian approximation in the Allen-Cahn system but, due to the accumulation of errors in many of the grid-points, the L^2 error is comparable to the Navier-Stokes Gaussian approximation error. For the L^∞ error we see that the generative model outperforms the Gaussian approximation in the nonlinear regime and is comparable (but less performant) in the linear regime.

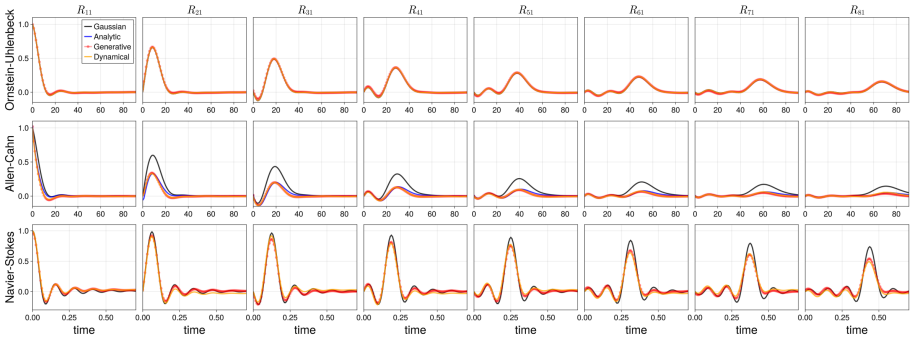

Figure 1: **Response functions for Three Systems**. The response to a perturbation at the pixel component $i = 1$ is evaluated at various pixel components with coordinates indicated at each panel. All evaluated components align with the advection direction. Responses are computed by integrating the dynamical system (orange lines, "Dynamical") via Gaussian approximation (black lines, "Gaussian"), using the score function derived from generative modeling (red lines and dots, "Generative"), and using the analytic score function (blue lines, "Analytic"). The "Dynamical" response is taken as the "ground truth".

3.3 A Comment on Generative Score function

The generative score function does not find the score function of the data distribution but instead finds the score function convolved with an independent multivariant normal whose width is $\sigma_{\min}$ due to the use of the denoising score loss. In principle, we can instead calculate covariances of the "noised" data. Thus, we can add to our multivariate random noise $\vec{Z}$. This modification, in principle, allows us to satisfy the identity, Equation (47), to a higher fidelity; however, this procedure is useful only if the true system response is robust in the $\sigma_{\min} \to 0$ limit.

Adding small noise to the data has the additional benefit of regularizing the distribution. Thus, if the distribution lies in a lower dimensional subspace, convolving it with a multivariate Gaussian makes it occupy the full space. When performing an estimate in the Gaussian case, this has the consequence that the covariance matrix becomes

$$C(\vec{u} + \vec{Z}, \vec{u} + \vec{Z}) = C(\vec{u}, \vec{u}) + \sigma_{\min}^2 \mathbb{I} \tag{56}$$

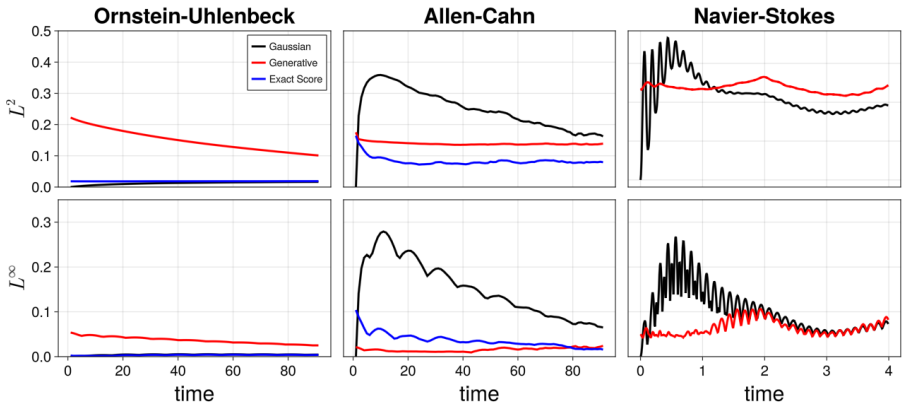

Figure 2: **Response Function Errors for the Three Systems.** Here, we show both the RMS error and the Infinity Norm error of the Response function as a function of time. Here, we use the Dynamic response function as the "ground truth". We see that the Generative response (red) generally outperforms the Gaussian response (blue) for the nonlinear cases (Allen-Cahn and Navier-Stokes), especially in the Infinity Norm, and performs similarly for the linear system (Ornstein-Uhlenbeck).

where C is the spatial covariance operator and $\mathbb{I}$ is the identity matrix.

An alternative is to train a Neural network for $\sigma_{\min} = 0$ separately using a different loss function.

3.4 The Effect of Shifts and Normalizations on Response Functions

It is common to shift and normalize the data so that

$$\tilde{u} = au + b \tag{57}$$

where a and b are constants. Regarding response functions, we note that the response function for u and $\tilde{u}$ is exactly the same. To see this denote the distributions for u and $\tilde{u}$ by $\rho_1(u)$ and $\rho_2(\tilde{u})$. The response matrix is

$$\langle u_i \partial_{u_j} \ln \rho_1 \rangle = \langle (a^{-1}\tilde{u}_i - a^{-1}b)\partial_{a^{-1}\tilde{u}_j} \ln \rho_2 \rangle = \langle \tilde{u}_i \partial_{\tilde{u}_j} \ln \rho_2 \rangle. \tag{58}$$

It is easier to see that the above relation holds from the empirical estimate of the expected value. Thus, one can correlate the score function of the shifted and scaled data distribution with the shifted and scaled data to get the response function for the unshifted and scaled data distribution. More complex data transformations require further analysis.

4 Denoising Score Matching

In this section, we focus on estimating the score function of the attractor of a dynamical system, as defined in Equation (3), through a denoising score matching training procedure often used in generative modeling (4; 5, and follow-on studies). This process approximates the score with a trainable function and subsequently optimizes its parameters. In principle, this could be achieved by minimizing an explicit score-matching loss function in which a trainable function is compared to the exact score. However, the true score function is generally unknown.

To overcome this problem, we perturb the data across a continuum of noise scales as is done in (6; 4; 7) by using a forward diffusion process governed by the stochastic differential equation (SDE; (5)).

$$\mathrm{d}_\tau \vec{u} = \vec{\xi}_\tau, \tag{59}$$

where τ is a diffusion time not to be confused with the physical time of the dynamical system from the previous section. Here, we employ a notation common in the physical sciences, rather than using the Wiener process differential $\mathrm{d}\vec{W}$. Note that we consider a discretized version of the continuous scalar field u on an $N \times N$ grid, such that the dimensionality of the discretized u (denoted with the vector arrow) is N^2. Furthermore, $\vec{\xi}_\tau$ is a N^2-dimensional zero-mean random normal vector with diagonal covariance matrix and variance g_τ^2 for all components of $\vec{\xi}_\tau$. The initial condition for Equation (59), at time $\tau = 0$, is drawn from the steady-state distribution (e.g., the attractor):

$$\vec{u}_0 \sim \rho(u_1, \ldots, u_{N^2}), \tag{60}$$

where ρ corresponds here to the distribution of the discretized system in steady state. The solution of Equation (59) at any diffusion time τ, $\vec{u}_\tau$, follows a multivariate Gaussian distribution

$$\vec{u}_\tau \sim \mathcal{N}(\vec{u}_0, \sigma_\tau^2 \mathbb{I}), \tag{61}$$

with diagonal covariance matrix $\sigma_\tau^2 \mathbb{I}$ governed by the variance

$$\sigma_\tau^2 = \int_0^\tau g_{\tau'}^2 \, \mathrm{d}\tau'. \tag{62}$$

In particular, at $\tau = 0$ we have that $\sigma_0 = 0$ and therefore $\vec{u}_0$ is a sample from the data distribution. For generative modeling and this form of forward diffusion, the function g_τ is often chosen to grow exponentially with diffusion time τ so that at a final time $\tau = 1$ the variance is much larger than the original scale of the data points (e.g., 5). As a result, the distribution of $\vec{u}_1$ can be approximated as

$$\vec{u}_1 \approx \mathcal{N}(\vec{0}, \sigma_1^2 \mathbb{I}). \tag{63}$$

This is referred to as the "variance-exploding" form of the forward diffusion process (5). As noted in the main text, there is a corresponding equation for

the density which approaches a Gaussian distribution as $\tau \to \infty$. This choice ensures that the memory of the initial condition in the diffusion process is lost, mapping points in the observed space $\vec{u}_0$ to points in the latent space $\vec{u}_1$ drawn from a known distribution independent of the source data; the independence of the latent distribution from the data is what allows for easy generation of new samples by solving the reverse process.

Since the score function $\vec{s}_\tau$ of the density of the discrete system changes smoothly with τ, it is easier to learn using the denoising score matching loss (5). We choose a suitable function approximator (e.g., a neural network) and train it with the denoising score matching loss given by

$$\mathcal{L}(\theta) = \mathbb{E}\left[\lambda(\tau)^2 \left\| \vec{s}_\theta(\vec{u}_0 + \sigma_\tau \vec{\epsilon}, \tau) + \frac{\vec{\epsilon}}{\sigma_\tau} \right\|^2\right], \tag{64}$$

where $\lambda(\tau)$ is an optional weighting factor, θ indicate a set of parameters, and the expectation $\mathbb{E}$ is a shorthand for $\mathbb{E}_{\tau\sim\mathcal{U}_{[\tau_{\min},1]},\vec{u}_0\sim\rho,\vec{\epsilon}\sim\mathcal{N}(\vec{0},\mathbb{I})}$ (5). Previous work in (5) has found that limiting the expectation over time to a minimum value of $\tau_{\min} \ll 1$ is necessary as the loss function diverges at $\tau = 0$, where $\sigma_0 = 0$. Furthermore, the loss function in Equation (64) is slightly modified in training: we rewrite the score network as

$$\vec{s}_\theta(\vec{u}, \tau) = \frac{\vec{f}_\theta(\vec{u}, \tau)}{\sigma_\tau}, \tag{65}$$

where $\vec{f}_\theta(\vec{u}, \tau)$ is a trainable function and σ_τ is the conditional standard deviation of the forward diffusion process. This allows the function $\vec{f}_\theta$ to target a quantity of order unity (e.g., 4; 7; 8).

Having trained the score function $\vec{s}_\theta$ we can use it to estimate the score of the steady-state (e.g., the attractor) ρ to use it as a drop-in score estimator within the context of Equation (3) of the manuscript. This is accomplished by setting $\tau = \tau_{\min} \ll 1$ as input to the trained score model so that we have

$$\vec{s}(u) \approx \vec{s}_\theta(\vec{u}, 0) \approx \vec{s}_\theta(\vec{u}, \tau_{\min}), \tag{66}$$

where the first approximation is due to representation error in s_θ or imperfect training, and the second is due to evaluating that function at $\tau = \tau_{\min}$ and not at $\tau = 0$.

4.1 Choosing a weighting function $\lambda(\tau)$

As noted, the score function $\vec{s}_\theta(\vec{u}, \tau)$ is most often used in the literature to generate synthetic data samples using the reverse process of Equation (59) (9), and hence $\vec{s}_\theta(\vec{u}, \tau)$ is required for all τ. However, in this work, we are primarily interested in the score network evaluated at $\tau = 0$. This can be controlled with the $\lambda(\tau)$ weighting function.

For the Ornstein-Uhlenbeck and Allen-Cahn systems, we adopt the standard parameterization of $\lambda(\tau) = \sigma(\tau)$(4; 7; 8), which has been found to produce

high-quality samples. However, for the Navier-Stokes case, we found it important to emphasize the diffusion times closest to zero in the loss function; in this case, we chose $\lambda(\tau) = \delta(\tau - \tau_{\min})\sigma(\tau)$. Note that different weighting schedules can strongly affect the quality of generated samples (e.g., 10). This is important because the samples can be used to validate the model when no other source of truth is available.

4.2 The noising schedule

The prescribed noising schedule $\sigma(\tau)$ remains to be specified. Using the Variance Exploding (VE) schedule (5), we have the noising process

$$g_\tau \equiv \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^\tau \sqrt{2 \log\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)} \tag{67a}$$

$$\sigma_\tau^2 \equiv \sigma_{\min}^2 \left[\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2\tau} - 1\right], \tag{67b}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are scalar parameters determining the shape of the variance with time. Suitable values of the $\sigma_{\max}$ can be computed from the data (8), e.g., $\sigma_{\max}$ is given by the largest pairwise L_2-distance in the dataset. From these expressions, it is clear that $\sigma_0 = 0$, i.e. that $\vec{u}_0$ coincides with the data. As noted above, $\sigma_0 = 0$ leads to divergences in the loss function and instability in training (see e.g. (5)), so we restrict the diffusion time to $1 \geq \tau \geq \tau_{\min}$.

For the Navier-Stokes case, we use a weighting function that only includes $\tau = \tau_{\min}$ as that is all we really need to generate good estimates for the score function at small τ. We used a value of $\sigma_{\min} = 10^{-3}$ and $\tau_{\min}$ as

$$\tau_{\min} = \log\left(\sqrt{2}\right) / \log\left(\sigma_{\max}/\sigma_{\min}\right), \tag{68}$$

which implies that the score function learned at this time is the score function of the data with a small amount of Gaussian noise added. Note that, $\sigma_{\tau_{\min}} = \sigma_{\min}$ thus the parameter $\sigma_{\max}$ cancels out at $\tau = \tau_{\min}$ and therefore does not affect the results for this case. For the Ornstein-Uhlenbeck and Allen-Cahn systems, we set $\sigma_{\min} = 0.01$, and $\sigma_{\max} = 14$.

5 Score Model Training Details

In this section, we describe the training procedure for the score models. Dataset sizes and time to train for the architecture we used were provided in the main text. All of what is described below is a summary of standard methods for the field. We found that no hyperparameter tuning was required for training the models using the Ornstein-Uhlenbeck and Allen-Cahn data sets, but varying the kernel size and noising schedule improved results for the Navier-Stokes data. Prior experience with training these models was invaluable.

5.1 Architecture

The network architecture used is based on a U-Net (11). An initial "lifting" layer preserves the size of the image, while increasing the number of channels. Three downsampling layers follow, which reduce the size of the image each by a factor of 2, while increasing the number of channels. Eight residual blocks follow, which preserve image size and channel number. Three upsampling layers then increase image size, each by a factor of two, and decrease the number of channels. A final projection layer projects down to a single channel while preserving image size. Two-dimensional spatial convolutions are used in each layer. For both the Ornstein-Uhlenbeck and Allen-Cahn data sets, convolutional kernels of size 3 x 3 pixels were used throughout. For the Navier-Stokes data set, a higher capacity network was required; we found that kernels of size 31x31, 31x31, 15x15, and 7x7 worked well for the lifting and three downsampling layers; we found also that dropout was required, and used a value of 0.5. In all cases, the channel increases are $1 \to 32 \to 64 \to 128 \to 256$. All convolutions used respect the periodicity of the system using periodic padding. As is standard in score-based diffusion modeling, the diffusion time τ is embedded using a Gaussian Fourier projection (12); we used an embedding dimension of 256. We made use of the publicly available Julia code `https://github.com/CliMA/CliMAgen.jl/` for the score-model training and sampling. All other architecture parameters were left at their defaults.

We found that doubling the number of channels in the U-net architecture to $1 \to 64 \to 128 \to 256 \to 512$ did not substantially improve results. Furthermore, increasing the kernel sizes in the Allen-Cahn and Ornstein-Uhlenbeck process did not improve the results. Using dropout decreased training time when using larger kernel sizes, but was unnecessary for the smaller kernel sizes.

5.2 Training

5.2.1 Data processing

All snapshots were normalized as described in Section 3.4. We used batch sizes of 128 for the Allen-Cahn and Ornstein-Uhlenbeck data, and 1024 for the Navier-Stokes data. We trained for 200 passes over the data for the OU and Allen-Cahn datasets, and 1600 for the Navier-Stokes dataset.

The Ornstein-Uhlenbeck system used $12,800$ snapshots, the Allen-Cahn system used $25,600$, and the Navier-Stokes data used $131,072$ snapshots. Approximately 20% of this data was witheld during training to create a "test" set. We did not thoroughly explore whether or not we could use less data and achieve the same fidelity of results.

5.2.2 Optimization

We used the default optimization configuration of CliMAgen.jl. This is an Adam optimizer with $\epsilon = 10^{-8}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ (13). We normalize the

gradient at each step, and use a warmup period of 5000 gradient updates to linearly increase the learning rate to 2×10^{-4} from 0.

5.3 Hyperparameter Sensitivity

Here, we aim to provide a condensed parameter sensitivity study by showcasing errors in computing response functions for two extremes of training a neural network architecture for the score of the Navier-Stokes system. The first architecture has kernel sizes of 3x3, 3x3, 3x3, and 3x3 for the lifting and three downsampling layers and channel increases of $1 \to 32 \to 64 \to 128 \to 256$. Furthermore, it is trained without dropout on the default loss function for all diffusion times $[\tau_{\min}, 1]$ with $\sigma_{\min} = 10^{-3}$ and $\sigma_{\max} = 10$. The second architecture has kernel sizes of 31x31, 31x31, 15x15, and 7x7 for the lifting and three downsampling layers and channel increases of $1 \to 64 \to 128 \to 256 \to 512$. Furthermore, it is trained with a dropout of 0.5 on a modified loss function for diffusion times $[\tau_{\min}, \tau_{\min}]$ with $\sigma_{\min} = 10^{-3}$. This architecture is twice the size used in the main manuscript due to the channel sizes and produces lower errors than that reported in the main manuscript. The first architecture took 30 seconds per epoch, and the second architecture took 5 minutes per epoch on an Nvidia A100 GPU.

We see the response function errors of using the raw and corrected score functions as a function of simulated time and number of epochs used for training the network over the same dataset in Figures 3 and 4. We overlay the Gaussian error in both figures for comparison. In all cases, we see that using the corrected generative score improves the error in computing the response function compared to the generative score without correction (raw). Furthermore, we see that in both architectures, once sufficiently trained, the generative model outperforms the Gaussian response, especially with regard to the most extreme responses. From Figure 3, we see that the raw generative response does not match the performance of the Gaussian approximation for the default network and loss functions for any number of training epochs, which we attribute to a neural network architecture with insufficient complexity to capture the score function of the Navier-Stokes system over all diffusion times. On the other hand, we see in Figure 4, with a larger architecture and different loss function, the raw generative model outperforms the Gaussian approximation and nearly matches that of the corrected generative model for the latter epochs. Early in training, we see that the corrected generative response matches that of the Gaussian response.

5.4 Model Validation

For both the Ornstein-Uhlenbeck and Allen-Cahn data sets, we validated the learned score function by numerically solved the reverse SDE using an Euler-Maruyama timestepper with initial condition $\vec{u}_1$ drawn from the approximate latent distribution at $\tau = 1$, e.g., $\vec{u}_1 \sim \mathcal{N}(\vec{0}, \sigma_{\max}^2 \mathbb{I})$ (5). The diffusion time interval $\tau \in [1, \tau_{\min}]$ was split into 250 equally spaced steps.

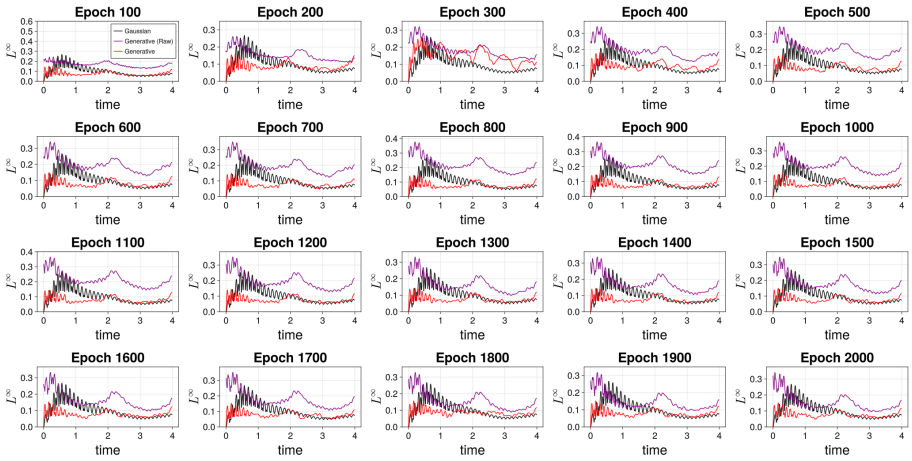

Figure 3: **Response Function Errors for the Navier-Stokes System as the Training Proceeds for a Small Network.** We show the Gaussian Response (black) as a baseline for comparison with the Raw Generative Model (purple) and the corrected Generative model (red) for several epochs of training.

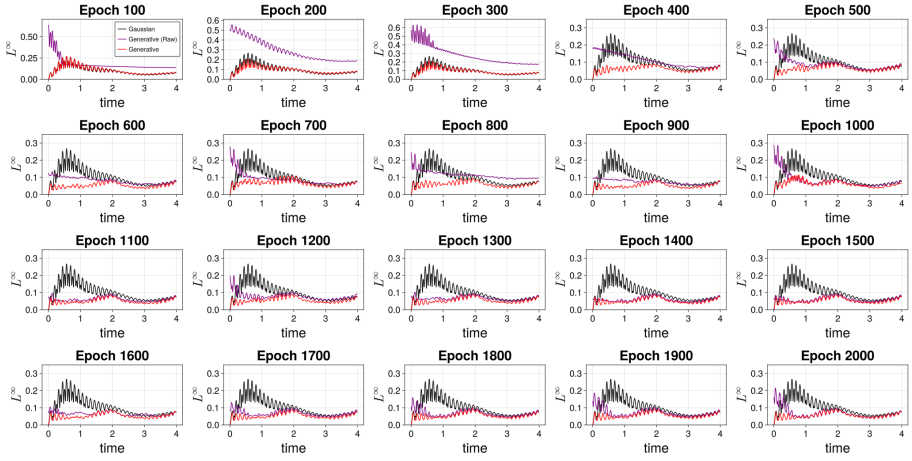

Figure 4: **Response Function Errors for the Navier-Stokes System as Training Proceeds for a Larger Network.** We show the Gaussian Response (black) as a baseline for comparison with the Raw Generative Model (purple) and the corrected Generative model (red) for several epochs of training.

We generated samples using the reverse diffusion sampling strategy. Various statistical metrics (power spectra, the first four moments of the pixel value distribution) were computed using these synthetic samples and using samples from the real training data in order to assess the quality of the model. We ascertained that the power spectra and moments computed with both sample sets agreed within the uncertainties due to finite sample size, and we also visually inspected the loss curves to assess whether the training was converged.

This validation was not possible for the Navier-Stokes trained model, due to our choice of weighting function of $\lambda(\tau)$ - see Section 4.2.

In all cases, we validate the score at $\tau = \tau_{\min}$ via response functions, as noted in the main text, and against analytic expectation if available.

6 Further Studies with the Allen-Cahn Equation

Figure 5 shows an example trajectory of the modified Allen-Cahn equation. As we proceed further in time, we see that the pattern shifts to the right (as given by the advection term) and changes according to the stochastic dynamics.

Example training data samples and the steady-state pixel distribution of the numerically simulated system are shown in the upper row of Figure (6). The machine-learned method generates new samples in the bottom row of Figure (6) and the corresponding steady-state pixel distribution. The machine-learned score function can generate samples similar to the original distribution. We compare the steady-state pixel distributions of both the generated fields and data samples to the Gaussian distribution defined by the data's mean and variance (as used in the Gaussian approximation) in the rightmost panels of Figure (6).

Since we have access to the analytic score function, we have an alternative method of assessing the generative model's fidelity. We evaluate Equation (42) (scaled by $1/32^2$) and the generative score for several choices of fields u in Figure (7). We test both on data similar to what the neural network has seen before (but not in the training set) and fields far from the original attractor distribution. When applying the score function to a constant field (left panel) for a choice of amplitude (such as $u = 0, u \pm 1, u = 0.5$, etc.), we obtain a cubic polynomial for the analytic score and a similar shape for the generative score, albeit with a significantly smaller amplitude. When evaluated on a smooth field (top three panels on the right), the generative score predicts the "rings". Still, it cannot obtain the center and mistakenly assigns different values to that region. Lastly, we see excellent agreement between the analytic and predictive score functions when evaluated on data similar to the steady-state distribution of the Allen-Cahn equation (bottom three panels on the right). We reemphasize that the neural network was only exposed to images of the steady-state distribution. In summary, the neural network reproduces the score function when evaluated on data similar to what it has been exposed to. However, the neural network produces quantitatively incorrect answers with similar qualitative features for significantly out-of-sample functions.

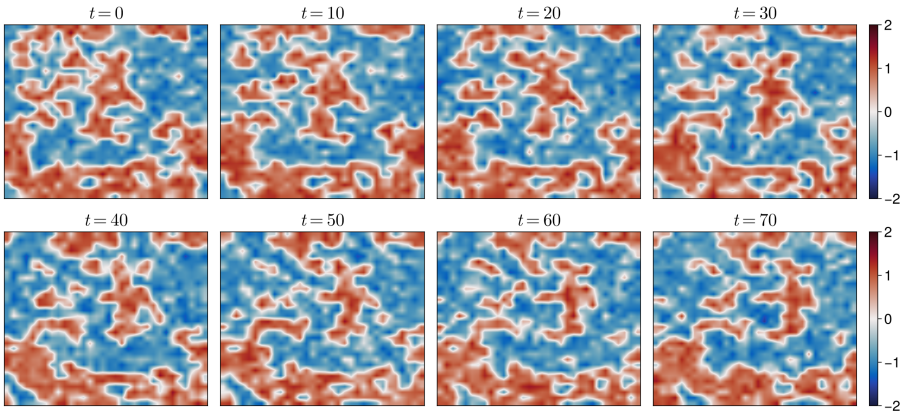

Figure 5: **Example trajectory of the Allen-Cahn equation.** Time progresses from 0 to 70 time units from top left to bottom right. We used approximately uncorrelated snapshots - sampled every 100 units of time - in training. Both the periodicity of the solution at the boundaries and the advection (to the right) are also visible in the trajectory.

7 Colormaps of the System Responses

In this section, we present the colormaps of $R_{ij}(t)$, defined in Equation (3) of the manuscript, for different values of t and for each of the three systems studied: Ornstein-Uhlenbeck (Figure 8), Allen-Cahn (Figure 9), and Navier-Stokes (Figure 10). These figures illustrate how each system responds to a small perturbation applied at the index $(1, 1)$, showing how the perturbation propagates over time.

The use of periodic boundary conditions on all sides allowed us to apply a circular shift to the image pixels. This method repositions the perturbed pixel to the center. For visual clarity, a shorter color range compared to Figure 2 in the manuscript is used.

8 Numerical Solution of the PDEs in the Manuscript

We use a Fourier spectral method for the spatial discretization for all three systems in the manuscript (14). The temporal discretization was implemented via a Runge-Kutta 4 scheme for the deterministic dynamics accounting for the noise term after the final Runge-Kutta stage. The timestep size was $\Delta t = 1/32$ for the Ornstein-Uhlenbeck and Allen-Cahn equations and $\Delta t = 1/128$ for the Navier-Stokes system. No dealiasing was used for any of the nonlinear terms of the simulation. All systems were evolved in a domain of size $[0, 2\pi)^2$ and took less than an hour to generate the dataset. The Navier-Stokes system was

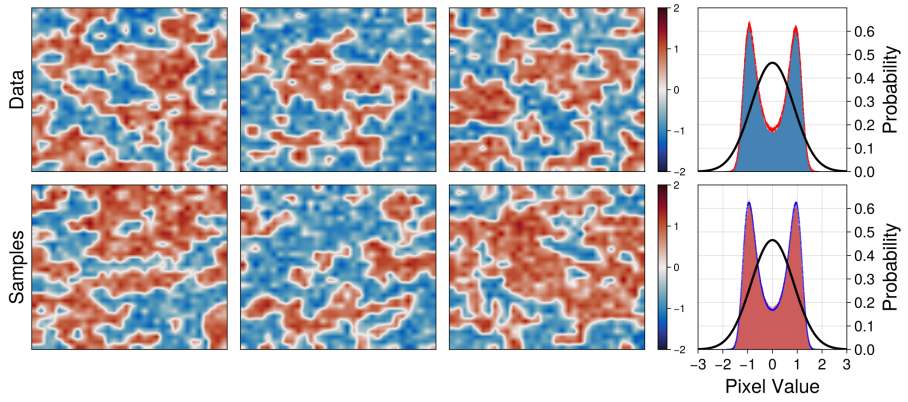

Figure 6: **Example snapshots and steady-state pixel distribution of the solution to the Allen-Cahn equation**. In the **top row**, we show the results of a numerical simulation of the Allen-Cahn equation) on a domain of size $2\pi \times 2\pi$ using 32×32 pixels; in the **bottom row**, we show samples generated via the learned score function using reverse diffusion (5). The rightmost column shows the corresponding pixel distributions, where the blue and red lines indicate the equivalent distributions from the top and bottom, respectively. The black line shows the Gaussian approximation fitted to the bimodal pixel distribution.

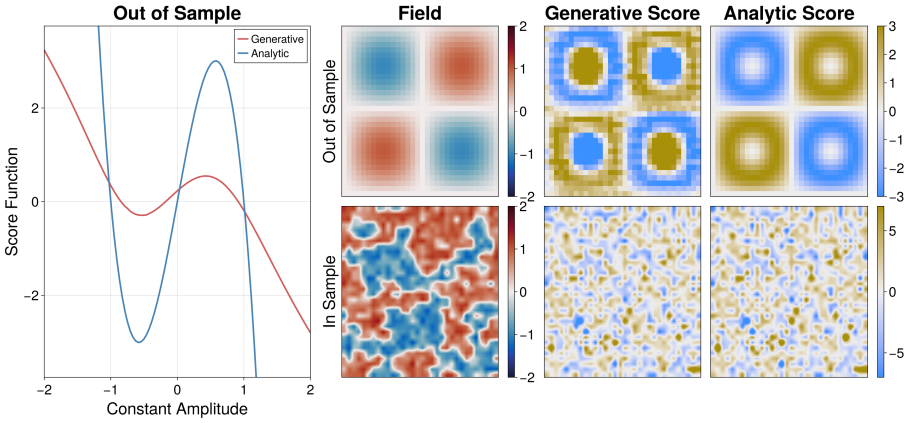

Figure 7: **Score function evaluation.** Here, we evaluate the analytic score function and the generative score function for several cases. In the leftmost plot, we show the evaluation of a constant function with amplitude given by the horizontal axis. On the right, we show the evaluation of the score function in two separate cases: The first is on the function $\sin(x)\sin(y)$, and the second is a data sample in the test set. Color ranges are the same for fields in which comparisons can be made.

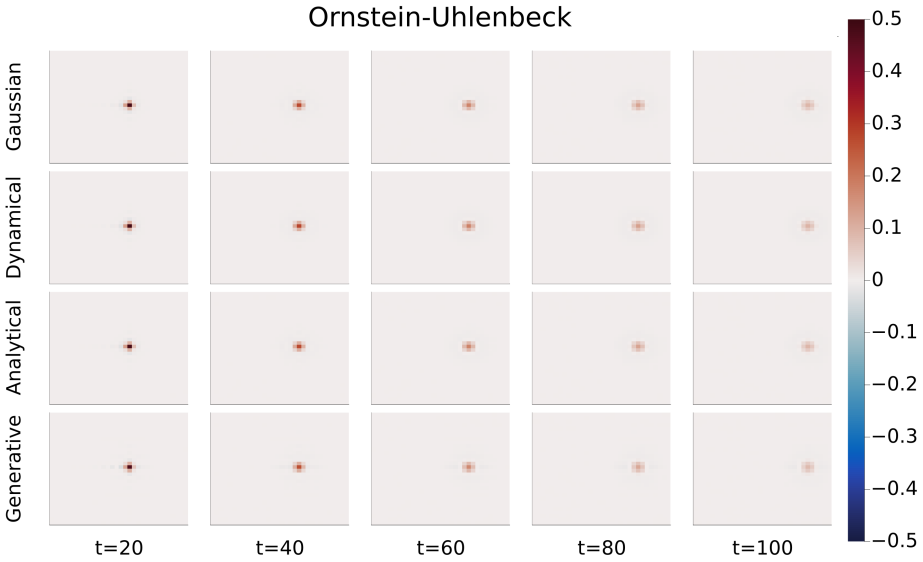

Figure 8: Colormap of the Ornstein-Uhlenbeck system response for different values of the time parameter t.

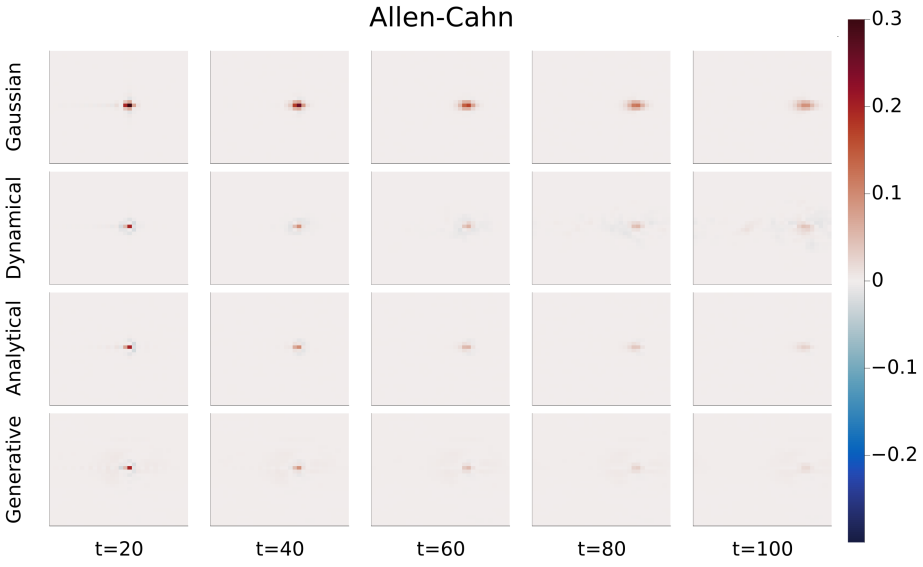

Figure 9: Colormap of the Allen-Cahn system response for different values of the time parameter t.

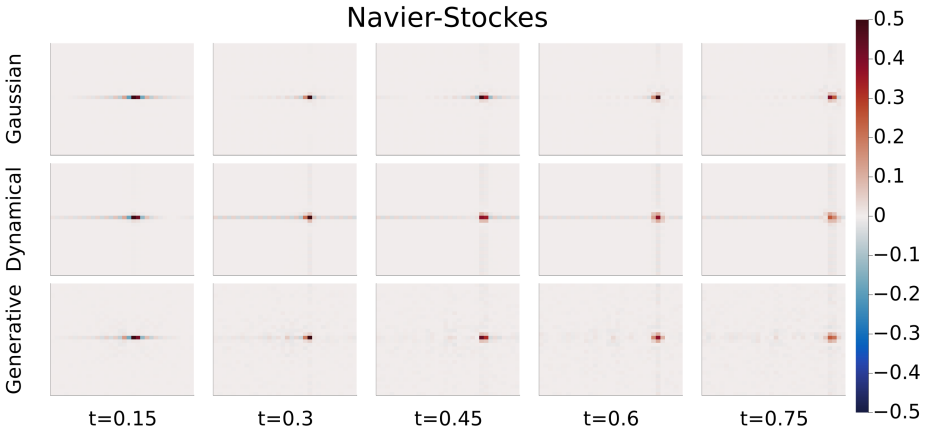

Figure 10: Colormap of Navier-Stockes system response for different values of the time parameter t.

implemented on an A100 GPU and the Ornstein-Uhlenbeck and Allen-Cahn equations were run on a single core of a CPU with an Apple M1 chip, both with the Julia programming language.

The Navier-Stokes system was discretized by averaging the conservative and advective forms of the discrete system. The random-wave forcing in (15) co-evolves a random variable $\vec{\varphi}$ alongside the Navier-Stokes equations as,

$$\dot{\vec{\varphi}} = \xi \tag{69}$$

and the forcing on the right-hand side of the Navier-Stokes equation was

$$\tilde{\varsigma} = \mathcal{F}^{-1}\left[F\sum_{i=1}^{N} a_i \exp(\iota\varphi_i)\right] \text{ and } \varsigma = \text{real}\left(\tilde{\varsigma} - \nu_f \int \tilde{\varsigma}\,\mathrm{d}x\,\mathrm{d}y\right) \tag{70}$$

where $F = 18.75, \nu_f = 0.9/(2\pi)^2$, $\mathcal{F}^{-1}$ is the default discrete inverse Fourier-Transform from FFTW, $\iota = \sqrt{-1}$, and a_i is a filter with the subscript looping over all wavenumbers. Specifically, $a_i = 1$ if $|\vec{k}_i| \leq \sqrt{112.5}$ and $a_i = 0$ otherwise. In our case, the dimensionality of the vector $\vec{\varphi}$ is the same as the Navier-Stokes equation, 32^2.

References

[1] M. Baldovin, F. Cecconi, A. Vulpiani, Understanding causation via correlations and linear response theory, Physical Review Research 2 (4) (2020) 043436.

[2] A. N. Souza, M. Tao, Metastable transitions in inertial langevin systems: What can be different from the overdamped case?, European Journal of Applied Mathematics 30 (5) (2019) 830–852. doi:10.1017/S0956792518000414.

[3] I. Corwin, H. Shen, Some recent progress in singular stochastic partial differential equations, Bulletin of the American Mathematical Sociefty 57 (3) (2020) 409–454.

[4] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, Advances in neural information processing systems 32 (2019).

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

[6] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International conference on machine learning, PMLR, 2015, pp. 2256–2265.

[7] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

[8] Y. Song, S. Ermon, Improved techniques for training score-based generative models, Advances in neural information processing systems 33 (2020) 12438–12448.

[9] B. D. Anderson, Reverse-time diffusion equation models, Stochastic Processes and their Applications 12 (3) (1982) 313–326.

[10] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, S. Yoon, Perception prioritized training of diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11472–11481.

[11] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[12] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, Advances in Neural Information Processing Systems 33 (2020) 7537–7547.

[13] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[14] J. P. Boyd, Chebyshev and Fourier Spectral Methods, 2nd Edition, Dover Books on Mathematics, Dover Publications, Mineola, NY, 2001.

[15] G. R. Flierl, A. N. Souza, On the non-local nature of turbulent fluxes of passive scalars, Journal of Fluid Mechanics 986 (2024) A8. doi:10.1017/jfm.2024.302.