Contents lists available at ScienceDirect

Journal of the Mechanics and Physics of Solids

journal homepage: www.elsevier.com/locate/jmps

Statistical-Physics-Informed Neural Networks (Stat-PINNs): A machine learning strategy for coarse-graining dissipative dynamics

Shenglin Huang [a], Zequn He [a], Nicolas Dirr [b], Johannes Zimmer [c], Celia Reina [a],*

[a] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, USA
[b] School of Mathematics, Cardiff University, Cardiff CF24 4AG, UK
[c] School of Computation, Information and Technology, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

ARTICLE INFO

Dataset link: https://github.com/celiareina/Stat-PINNs

Keywords:
Physics-Informed Neural Networks
Fluctuation–dissipation relations
GENERIC
Non-equilibrium thermodynamics
Dissipation
Machine learning
Free energy
Arrhenius process

ABSTRACT

Machine learning, with its remarkable ability for retrieving information and identifying patterns from data, has emerged as a powerful tool for discovering governing equations. It has been increasingly informed by physics, and more recently by thermodynamics, to further uncover the thermodynamic structure underlying the evolution equations, *i.e.*, the thermodynamic potentials driving the system and the operators governing the kinetics. However, despite its great success, the inverse problem of thermodynamic model discovery from macroscopic data is in many cases non-unique, meaning that multiple pairs of potentials and operators can give rise to the same macroscopic dynamics, which significantly hinders the physical interpretability of the learned models. In this work, we consider the problem of deriving the macroscopic (continuum) equations from microscopic (particle) data, and encode knowledge from statistical mechanics to resolve this non-uniqueness for the first time. The proposed machine learning framework, named as Statistical-Physics-Informed Neural Networks (Stat-PINNs), is here developed for purely dissipative isothermal systems. Interestingly, it only uses data from short-time particle simulations to learn the thermodynamic structure, which can in turn be used to predict long-time macroscopic evolutions. We demonstrate the approach for particle systems with Arrhenius-type interactions, common to a wide range of phenomena, such as defect diffusion in solids, surface absorption, and chemical reactions. Our results from Stat-PINNs can successfully recover the known analytic solution for the case with long-range interactions and discover the hitherto unknown potential and operator governing the short-range interaction cases. We compare our results with direct particle simulations and an analogous approach that solely excludes statistical mechanics, and observe that, in addition to recovering the unique thermodynamic structure, statistical mechanics relations can increase the robustness and predictive capability of the learning strategy.

1. Introduction

Dissipative phenomena are pervasive across material systems, from diffusion in gases, to viscous flow in fluids, to plasticity in crystalline and granular media. Yet, our understanding of these phenomena is severely hindered by computational and theoretical challenges. Computationally, direct particle simulations remain elusive for macroscopic length- and time-scales, even with the latest supercomputers. And theoretically, only a small number of particle systems enjoy an analytical macroscopic description in terms of thermodynamic field equations (Kipnis and Landim, 1998; Presutti, 2009; Bodineau et al., 2016; Katsoulakis and Vlachos, 2003).

* Corresponding author.
E-mail address: creina@seas.upenn.edu (C. Reina).

https://doi.org/10.1016/j.jmps.2024.105908
Received 10 February 2024; Received in revised form 16 September 2024; Accepted 13 October 2024
Available online 24 October 2024

Therefore, macroscopic equations, while computationally efficient, are often based on phenomenological assumptions not grounded in the physics of the underlying particle dynamics. As a result, strategies for modeling of dissipative phenomena typically suffer from a trade-off between physical fidelity and computational efficiency. Notable approaches include sequential and concurrent multiscale techniques (Tadmor and Miller, 2011), mathematical coarse-graining strategies (Givon et al., 2004), such as the Mori–Zwanzig formalism (Zwanzig, 2001), and data-driven approaches (Brunton and Kutz, 2022), like dimensionality reduction and machine-learning techniques. The latter will be the focus of the present investigation.

Despite the numerous challenges in rigorously deriving macroscopic equations away from equilibrium, great progress has been made in the last 15 years in the understanding of their structure. Indeed, the General Equation for Non-Equilibrium Reversible-Irreversible Coupling (GENERIC) formalism (Grmela and Öttinger, 1997; Öttinger and Grmela, 1997; Öttinger, 2005) provides the structure of macroscopic evolution equations as they emerge from underlying microscopic Hamiltonian dynamics while ensuring compatibility with the laws of thermodynamics. According to this formalism, the state variables z describing a closed system evolve through a combination of a symplectic operator acting on the energy of the system and a dissipative operator acting on the entropy,

$$\frac{\partial z}{\partial t} = \mathcal{L}_z \frac{\delta E}{\delta z} + \mathcal{M}_z \frac{\delta S}{\delta z}, \tag{1}$$

where E and S denote the total energy and entropy functionals, and $\mathcal{L}_z$ and $\mathcal{M}_z$ are an (antisymmetric) Poisson operator and a symmetric and positive semi-definite operator, respectively, satisfying the degeneracy conditions $\mathcal{L}_z \frac{\delta S}{\delta z} = \mathcal{M}_z \frac{\delta E}{\delta z} = 0$. This structure guarantees the conservation of energy in closed systems, as well as a positive entropy production, in accordance with the second law of thermodynamics.

For isothermal systems, which are the focus of this article, the evolution is a special case of Eq. (1), namely (see Öttinger (2005, Sect. 1.2.2))

$$\frac{\partial z}{\partial t} = -\mathcal{K}_z \frac{\delta F[z]}{\delta z}, \tag{2}$$

where F is the free energy functional and $\mathcal{K}_z$ is the dissipative operator, which is a symmetric and positive semi-definite operator that linearly acts on the thermodynamic force $\frac{\delta F}{\delta z}$ and depends on the field z (generally in a nonlinear manner). Eq. (2) is often called a *gradient flow*, since the free energy functional F decreases monotonically along trajectories (*i.e.*, it is a Lyapunov functional):

$$\frac{\partial F}{\partial t} = \left\langle \frac{\delta F[z]}{\delta z}, \frac{\partial z}{\partial t} \right\rangle = -\left\langle \frac{\delta F[z]}{\delta z}, \mathcal{K}_z \frac{\delta F[z]}{\delta z} \right\rangle \leq 0.$$

The bracket $\langle \cdot, \cdot \rangle$ denotes an inner product in a Hilbert space, and the inequality holds since $\mathcal{K}_z$ is positive semidefinite, which is consistent with the second law of thermodynamics for isothermal systems.

The GENERIC formalism has proven to be very advantageous from a modeling standpoint (Mielke, 2011), and has been successfully used to model, for instance, finite-dimensional mechanical systems, phase field models, thermoviscoelastic solids, or diffusion phenomena. Its structure also reveals additional information. As just noted, the free energy acts as a Lyapunov functional for systems evolving according to Eq. (2). Furthermore, Eqs. (1) and (2) are also endowed with strong statistical mechanics foundations (Öttinger et al., 2021; Montefusco et al., 2021; Li et al., 2019; Kraaij et al., 2020; Leadbetter et al., 2023), leading directly, for instance, to the stochastic PDE describing the evolution of finitely many particles (the so-called equation of fluctuating hydrodynamics). In view of its success and microscopic foundation, we here assume that the macroscopic equations that we aim to discover are of GENERIC type.

With the advent of machine learning, several strategies have emerged to discover the thermodynamic potentials and operators of the GENERIC equation from the temporal evolution of macroscopic field variables. These strategies include, in the context of ODEs, Structure-Preserving Neural Networks (SPNNs) (Hernandez et al., 2021), GENERIC Neural Ordinary Differential Equations (GNODEs) (Lee et al., 2021), GENERIC formalism Informed Neural Networks (GFINNs) (Zhang et al., 2022), and thermodynamics-informed graph neural networks (Hernández et al., 2022). Within a variational perspective, Variational Onsager Neural Networks (VONNs) (Huang et al., 2022) learn the action density guiding the partial differential equations (of GENERIC type) in isothermal systems. The benefit of these strategies is that the learned structure automatically incorporates central physical features, such as the conservation of energy in closed systems, or the existence of a Lyapunov functional. However, all of the above methods can suffer from a lack of uniqueness, in the sense that correctly learning the dynamical equations in the form of Eqs. (1) or (2) does not always guarantee the discovery of the correct thermodynamics and kinetics. This lack of uniqueness, which commonly occurs, for example, in diffusive phenomena, can thus severely limit the *physics* learned by the machine learning algorithms. As an example, the evolution of the density field ρ for the simplest case of linear diffusion

$$\frac{\partial \rho}{\partial t} = \Delta \rho, \tag{3}$$

can be written in infinitely many different ways in the form of Eq. (2). That is, there are infinitely many pairs $\left(\mathcal{K}_\rho^{(j)}, F^{(j)}\right)$, with $j = 1, 2, \ldots$, which lead the linear diffusion equation (3). We describe here two possible representations. The first one is $\mathcal{K}_\rho^{(1)} \xi = \xi$, where ξ is an arbitrary field function, and $F^{(1)} = -\frac{1}{2} \int |\nabla \rho|^2 dx$. Indeed, $\frac{\delta F}{\delta \rho} = -\Delta \rho$. This is the so-called L^2 gradient flow of the negative Dirichlet integral. Second, the diffusion equation can be written as what is now called a *Wasserstein gradient flow* where $\mathcal{K}_\rho^{(2)} \xi = -\nabla \cdot (\rho \nabla \xi)$ and $F^{(2)} = \int \rho \ln \rho \, \mathrm{d}x$ is the Boltzmann entropy. In this case, $\frac{\delta F^{(2)}}{\delta \rho} = \ln \rho + 1$, hence $\mathcal{K}_\rho^{(2)} \frac{\delta F^{(2)}}{\delta \rho} = -\nabla \cdot (\rho \nabla (\ln \rho + 1)) = -\nabla \cdot \left(\rho \frac{\nabla \rho}{\rho}\right) = -\nabla \cdot \nabla \rho = -\Delta \rho$, and $\frac{\partial \rho}{\partial t} = \Delta \rho$. Thus, already in this very simple example, it is not possible to determine the operator $\mathcal{K}_\rho$ and the functional F from purely macroscopic (thermodynamic) considerations.

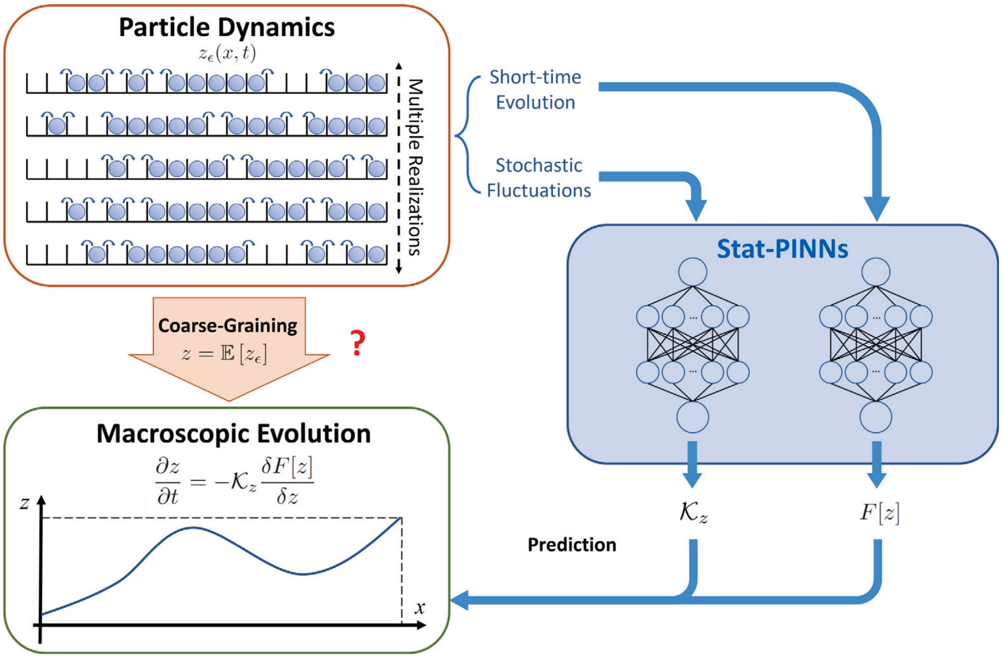

Fig. 1. Sketch of the Stat-PINNs (Statistical-Physics-Informed Neural Networks) framework for discovering the macroscopic evolution equation from short-time particle simulations.

The dissipative operator $\mathcal{K}$ and the free energy functional F are, however, unique for any given particle system. In the example above, it is a classic result going back to Einstein in 1905 that particle undergoing Brownian motion result macroscopically in linear diffusion. For these particles, the Boltzmann entropy $F^{(2)}$ is the natural functional, and this is twinned with the Wasserstein operator $\mathcal{K}_\rho^{(2)}$. In contrast, there is no natural particle foundation of the Dirichlet integral $F^{(1)}$.

The central idea of the present work is to incorporate microscopic information in the form of statistical physics to learn the true thermodynamics and kinetics of the system and resolve the non-uniqueness issue for the first time. While noise in the data is typically considered an undesirable feature, when such noise represents physical thermal fluctuations of particle data, it encodes a wealth of information that has proven instrumental in advancing our understanding of non-equilibrium phenomena (Kubo, 1966; Seifert, 2012; Sevick et al., 2008). Of particular interest to this work is the profound link between fluctuations of a given variable and the true dissipative operator governing its evolution, via an infinite-dimensional fluctuation–dissipation relation (Öttinger, 2005; Li et al., 2019). In particular, by measuring the fluctuations of z in addition to its expected value, it is possible to uniquely determine the dissipative operator guiding the evolution. The proposed machine learning strategy, which we denote as Statistical-Physics Informed Neural Networks (Stat-PINNs), is schematically shown in Fig. 1 for purely dissipative phenomena. It consists of two neural networks aimed at learning the unique thermodynamic dissipative operator $\mathcal{K}_z$ and thermodynamic potential (the free energy F for an isothermal process). Interestingly, both of these quantities, which govern the dynamics over arbitrarily long times, may be learned from short-time particle simulations.

Stat-PINNs are here developed in detail for dissipative phenomena with conservation laws, such as diffusion of conserved mass as their inverse problem is known to suffer from a lack of uniqueness. The proposed paradigm strongly encodes all the properties of the dissipative operator: acting linearly on the thermodynamic force $\frac{\delta F}{\delta z}$(while generally being nonlinear in the state variables z), being symmetric and positive semi-definite, as well as the restrictions associated with conservative fields z. We demonstrate the approach for particle systems with Arrhenius-type interactions, which are very common in thermally activated processes such as defect diffusion in solids, surface absorption, and chemical reactions. This specific particle process chosen has a known analytical solution for long-range interactions (Vlachos and Katsoulakis, 2000), while the continuum equation for short-range interactions is unknown, to the best of the authors' knowledge. Furthermore, it is one-dimensional, making it computationally feasible to perform long-time particle simulations as comparison data, to validate the proposed approach. Interestingly, Stat-PINNs are capable of recovering the analytical free energy and dissipative operator for the case of long-range interactions, as well as discovering those for the case of short-range interactions. The results are compared to an analogous approach that solely includes thermodynamic relations and conservation laws, *i.e.*, only excluding statistical mechanics. We observe that, in addition to recovering the unique thermodynamic potential and operator, statistical mechanics relations increase the robustness and predictive capability of the learning strategy.

We remark that Stat-PINNs purposely use a neural network to discover the free energy from data, in lieu of some alternative approaches rooted in statistical physics. One such alternative option would have been to use large deviation theory to determine the free energy (Touchette, 2009). While this approach is natural and elegant from a viewpoint of statistical mechanics, it requires resolution on an exponential scale. Conversely, the potential in this work is fitted directly by a neural network. There is also a rich body of literature on the computation of equilibrium free energies (see Lelièvre et al. (2010) for an in-depth presentation). These approaches allow the computation of free energy differences as required here, however only in a setting where a perturbative analysis of the system of interest is possible. The example of the Arrhenius process below demonstrates that the method presented here does not rely on perturbations.

The paper is organized as follows. In Section 2, we review the GENERIC formalism for isothermal dissipative systems and discuss the infinite-dimensional fluctuation–dissipation relation needed to uniquely discover the dissipative operator. Next, in Section 3, we introduce the Stat-PINNs architecture and the corresponding structure-preserving parameterization. This machine learning strategy is then applied in Section 4 over three Arrhenius-type interacting particle processes, of which only one of them has a known analytic solution. There, the results are compared to long-time particle simulations and Physics-Informed Neural Networks (PINNs) not informed by statistical mechanics. Finally, conclusions will be drawn in Section 5.

2. Isothermal dissipative dynamics

We consider an isothermal system whose macroscopic evolution may be written as a gradient flow of the non-equilibrium free energy as in Eq. (2).

There, $z = z(x,t)$ describes the field(s) of interest, and $F[z] = \beta\hat{F}[z]$ is the normalized total free energy functional (this will be referred to as free energy in the following, in the interest of simplicity), where $\hat{F}[z]$ is the original free energy functional in energy units, and $\beta = 1/k_B T$, with k_B being the Boltzmann constant and T the temperature of system. Further, $\delta F/\delta z$ is the functional derivative of the free energy, which acts as the thermodynamic driving force of the dissipative dynamics. Finally, $\mathcal{K}_z$ is a linear, symmetric and positive semi-definite operator, where the subscript z emphasizes its dependency on the field(s) z, which can be nonlinear in general. Dissipative dynamics of the form of Eq. (2) were proposed by Ginzburg–Landau and may be seen as a special case of the GENERIC formalism for a system at constant temperature (Öttinger, 2005; Mielke, 2011).

The continuum equation (2) can be mathematically seen as the system's dynamics in the limit of *infinite* number of particles. For *finite*, yet a large number of particles, isothermal dissipative systems can often be described by a stochastic differential equation of the form (to be interpreted in a suitable way as discussed below)

$$\frac{\partial z_\epsilon}{\partial t} = -\mathcal{K}_{z_\epsilon}\frac{\delta F[z_\epsilon]}{\delta z_\epsilon} + \sqrt{2\epsilon\mathcal{K}_{z_\epsilon}} \diamond \dot{W}_{x,t}, \tag{4}$$

where ϵ represents the level of "zooming out" in the description, *i.e.* $\epsilon \approx \frac{1}{L^d}$, where d is the space dimension and $1/L$ the characteristic inter-particle length scale (in the continuum limit, $\epsilon \to 0$, the equation becomes deterministic, as expected). Here $\dot{W}_{x,t}$ is a space–time white noise satisfying $\mathbb{E}[\dot{W}_{x,t}] = 0$ and $\mathbb{E}[\dot{W}_{x,t}\dot{W}_{x',t'}] = \delta(x-x')\delta(t-t')$, with $\mathbb{E}[\cdot]$ denoting the expectation and $\delta(\cdot)$ being the Dirac delta function. The stochastic integral denoted with the symbol $\diamond$ is the Klimontovich integral[1]. For this notion of stochastic integral, as well as the related integrals of Itô and Stratonovich and their conversion formulae see Pavliotis (2014, Sect. 3.2); for a discussion in the context of (4) see Öttinger et al. (2021).

The reformulation of Eq. (4) as an equation with Itô noise results in (Öttinger, 2005, Section 1.2.5)

$$\frac{\partial z_\epsilon}{\partial t} = -\mathcal{K}_{z_\epsilon}\frac{\delta F[z_\epsilon]}{\delta z_\epsilon} + \epsilon\partial_{z_\epsilon}\mathcal{K}_{z_\epsilon} + \sqrt{2\epsilon\mathcal{K}_{z_\epsilon}}\dot{W}_{x,t}. \tag{5}$$

The second but last term is of higher order, $O(\epsilon)$, and can thus be neglected in the computations, where it is sufficient to resolve $O(\sqrt{\epsilon})$. We thus study the following equation in this work

$$\frac{\partial z_\epsilon}{\partial t} = -\mathcal{K}_{z_\epsilon}\frac{\delta F[z_\epsilon]}{\delta z_\epsilon} + \sqrt{2\epsilon\mathcal{K}_{z_\epsilon}}\dot{W}_{x,t}, \tag{6}$$

where the noise is of Itô form. We highlight that the fluctuation operator $\sigma_{z_\epsilon} = \sqrt{2\epsilon\mathcal{K}_{z_\epsilon}}$ acting on the white noise $\dot{W}_{x,t}$ is related to the dissipative operator $\mathcal{K}_{z_\epsilon}$ through an infinite-dimensional fluctuation–dissipation relation, $\sigma_{z_\epsilon}\sigma^*_{z_\epsilon} = 2\epsilon\mathcal{K}_{z_\epsilon}$. This relation will be crucial in upcoming sections to determine the dissipative operator from fluctuation data. Existence of a solution to Eq. (6) is typically very subtle; for atomistic initial data as data representing particles as in this article, martingale solutions can be shown to exist for simple cases (Konarovskyi et al., 2019). For regularization by correlated noise, as it is relevant for applications, existence results are available, see for example (Cornalba et al., 2021; Fehrman and Gess, 2024; Dirr et al., 2020) and the references therein.

We remark that Eq. (4) with the Klimontovich integral is natural from a physical viewpoint, as the Gibbs measure $\frac{1}{Z}\exp(-\frac{1}{\epsilon}F)$ is in suitable situations, at least formally, invariant. That is, the Gibbs measure is the equilibrium distribution, consistently with statistical mechanics theory. We sketch this for Eq. (5) interpreted as an ordinary differential equation,

$$\frac{\partial z_\epsilon}{\partial t} = -\mathcal{K}_{z_\epsilon}\nabla F + \epsilon\nabla\mathcal{K}_{z_\epsilon} + \sqrt{2\epsilon\mathcal{K}_{z_\epsilon}}\dot{W}, \tag{7}$$

[1] In short, the Klimontovich, Itô and Stratonovich integrals of a function can be defined as the limit of a Riemann sum, where the function is evaluated at the right, left or middle point, respectively, for each interval. Each will lead to different result, in contrast to what usually happens for deterministic integrals.

where W is a Wiener noise (Gaussian white noise) in time. Then the evolution of an observable $f(z_\epsilon)$ can be computed from (7) using the Itô formula. It is given by $\frac{\partial f}{\partial t} = Lf$, where L is the generator (see Oksendal (2013, Sect. 7.3)). The density ρ evolves according to the associated Fokker–Planck equation, which involves the formal adjoint operator L^*, that is, $\frac{\partial \rho}{\partial t} = L^*\rho$, with

$$L^*\rho = \nabla\cdot(\mathcal{K}_{z_\epsilon}\rho\nabla F) - \epsilon\nabla\cdot(\nabla\mathcal{K}_{z_\epsilon}\rho) + \epsilon\nabla\cdot\nabla(\mathcal{K}_{z_\epsilon}\rho) = \nabla\cdot\left[(\mathcal{K}_{z_\epsilon}\rho\nabla F) + \epsilon\mathcal{K}_{z_\epsilon}\nabla\rho\right].$$

This expression vanishes for $\rho = \frac{1}{Z}\exp(-\frac{1}{\epsilon}F)$, indicating it is the equilibrium distribution as expected. For background on this material see Pavliotis (2014).

2.1. Harnessing fluctuations to learn the dissipative operator

We here briefly describe the computational strategy of Li et al. (2019) to learn a discretized version of the dissipative operator $\mathcal{K}_{z_\epsilon}$ from fluctuation data in particle simulations at a fixed value of ϵ. To that end, the profile z and the thermodynamic driving force $Q := \delta F/\delta z$ are approximated as in finite element discretizations as $z(x,t) \approx \sum_i z_i(t)\gamma_i(x)$ and $Q(x,t) \approx \sum_i Q_i(t)\gamma_i(x)$, where $\{\gamma_i(x)\}$ is a set of suitable basis functions. Then the weak form of Eq. (2) can be approximated as

$$\sum_i \left\langle \gamma_j, \gamma_i \right\rangle \dot{z}_i = -\sum_i \left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle Q_i \quad \text{for all } j, \tag{8}$$

where the bracket $\langle\cdot,\cdot\rangle$ denotes the L^2 inner product. Here $\left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle$ represents the discretized dissipative operator (now a finite-dimensional matrix). In Li et al. (2019), it is shown that the entries may be computed by the covariation of the rescaled fluctuations as

$$\left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle = \lim_{h\searrow 0}\frac{1}{2h}\mathbb{E}\left[\left(Y_{\gamma_i}(t_0+h) - Y_{\gamma_i}(t_0)\right)\cdot\left(Y_{\gamma_j}(t_0+h) - Y_{\gamma_j}(t_0)\right)\right]. \tag{9}$$

Here $Y_\gamma = \lim_{\epsilon\to 0}\langle z_\epsilon - z, \gamma\rangle/\sqrt{\epsilon}$ are the fluctuations of the many-particle system around the limit $z = \mathbb{E}\left[z_\epsilon\right]$; the fluctuations are scaled by a factor of $\sqrt{\epsilon}$, as they are by the central limit theorem of that order. Above, t_0 is an initial time, arbitrary as long as the system has reached a local equilibrium for the profile $z(x,t)$, and h is a time step that is infinitesimally small from a macroscopic perspective, yet, sufficiently large for the system to exhibit some stochastic events at the microscale. In practice, multiple realizations of a particle simulation at a fixed value of ϵ and given profile $z(x)$ are performed, and the simulation domain is discretized with shape functions $\{\gamma_i(x)\}$. The values of $\langle z_\epsilon, \gamma\rangle$ are then recorded at times t_0 and $t_0 + h$ to compute the right-hand side of Eq. (9).

To fully tabulate the discretized operator $\left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle$ and enable in such a way standalone macroscopic simulations, its input, $z(x)$, shall also be discretized. The approach we will here use to that regard differs from that of Li et al. (2019), and this will be described in further detail in Section 3.1.

3. Statistical-Physics-Informed Neural Networks (Stat-PINNs)

In this section we are going to introduce a machine learning framework, which we refer to as Statistical-Physics-Informed Neural Networks (Stat-PINNs), to learn the full evolution equation described by Eq. (2) (including both the dissipative operator $\mathcal{K}_z$ and the free energy $F[z]$) from particle simulations spanning a macroscopically small time step. First, in Section 3.1, we will introduce a structure-preserving parameterization of the discretized operator entries $\left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle$, so that its input, $z(x)$, is also discretized. Next, in Section 3.2, we will discuss various common dependences of the free energy functional on $z(x)$ and the corresponding structure for the evolution equations. Finally, we will describe the architecture of Stat-PINNs in Section 3.3, which includes two neural networks for sequentially learning the dissipative operator $\mathcal{K}_z$ and the free energy $F[z]$.

3.1. Structure-preserving parameterization of the discretized dissipative operator

As previously noted in Section 2, the dissipative operator $\mathcal{K}_z$ is linear as an operator acting on the thermodynamic driving force, symmetric, and positive semi-definite. Furthermore, $\mathcal{K}_z$ may be subjected to additional constraints when the field(s) z represent conserved quantities such as mass or energy. In order to learn a thermodynamic-consistent and numerically stable structure for its discretized version from particle data, these properties should be retained during the learning strategy. We will here discuss how to preserve such properties in the context of a conserved filed z (linearity is automatically satisfied and hence omitted from the discussions).

For simplicity, we consider a one-dimensional problem and choose linear finite element shape functions $\{\gamma_i(x)\}$, satisfying $\gamma_i(x_j) = \delta_{ij}$, with $x_i = i\Delta x_\gamma$ and $i = 0, 1, 2, \ldots, N_\gamma$. This choice provides a piecewise linear approximation for both z and Q, and is sufficient to characterize dissipative operators containing up to second-order derivatives. Notably, the local support of the shape functions can result in a sparse matrix $\left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle$, highly simplifying the learning strategy, as will be discussed next.

In general, the dissipative operator $\mathcal{K}_z$ can have nonlocal dependencies on both z and the thermodynamic force Q being acted on. Here, we classify the nonlocality of $\mathcal{K}_z$ into three categories. First, $\mathcal{K}_z$ may be local on both z and Q, such as $\mathcal{K}_zQ = -\nabla\cdot(z(x,t)\nabla Q)$. In this case, $\mathcal{K}_z\gamma_i(x)$ has the same local support as $\gamma_i(x)$ on $x \in [x_{i-1}, x_{i+1}]$ and only depends on the profile of $z(x,t)$ within that support. Second, $\mathcal{K}_z$ may be local on the acting force Q but nonlocal on the profile z, such as $\mathcal{K}_zQ = -\nabla\cdot\left(\int_{-L}^{L} z(x-\xi,t)d\xi\nabla Q\right)$. Here, $\mathcal{K}_z\gamma_i(x)$ still has the same local support as $\gamma_i(x)$ but depends on a wider range of the profile z. Third, $\mathcal{K}_z$ may be nonlocal on Q, such as $\mathcal{K}_zQ = \int\kappa(|x-\xi|)[z(x-\xi,t)+z(\xi-x,t)]Q(\xi)d\xi$, where $\kappa(\cdot)$ is a kernel function. In this case, $\mathcal{K}_z\gamma_i(x)$ has a larger local support than $\gamma_i(x)$ and may even be non-zero on the full domain. In the following, we will focus on the first and second cases, and refer the reader to Appendix A for a detailed discussion on the fully nonlocal case.

When the operator $\mathcal{K}_z$ is local on the driving force Q (first and second cases listed above), $\mathcal{K}_z\gamma_i(x)$ has the same local support as $\gamma_i(x)$. As a result, the discretized dissipative operator becomes a tridiagonal matrix, *i.e.*, $\langle \gamma_j, \mathcal{K}_z\gamma_i \rangle$ non-zero only when $j \in \{i-1, i, i+1\}$. Under this assumption, which may be verified numerically using Eq. (9), Li et al. (2019) parameterized these three operator entries by the values of the local profile and its gradient at point x_i, *i.e.*, z_i and $\nabla z|_i$. However, this approximation breaks the symmetry of the operator and may induce, based on our observations, numerical instabilities when solving Eq. (8). To preserve the operator symmetry, we here regard the entries of the tridiagonal matrix as two functions of several discrete values of z. Specifically,

$$\begin{aligned} &K_0(\mathbf{Z}_i^0) = \langle \gamma_i, \mathcal{K}_z\gamma_i \rangle, \quad \text{and} \\ &K_1(\mathbf{Z}_i^1) = \left\langle \gamma_{i+1}, \mathcal{K}_z\gamma_i \right\rangle = \left\langle \gamma_i, \mathcal{K}_z\gamma_{i+1} \right\rangle, \end{aligned} \tag{10}$$

where $\mathbf{Z}_i^0 = (z_{i-1}, z_i, z_{i+1})$ and $\mathbf{Z}_i^1 = (z_i, z_{i+1})$ when the operator is local on z (or the nonlocal effect is negligible within the given numerical discretization). If $\mathcal{K}_z$ is nonlocal on z, then additional neighboring points may be needed, *e.g.*, $\mathbf{Z}_i^0 = (z_{i-2}, z_{i-1}, z_i, z_{i+1}, z_{i+2})$ and $\mathbf{Z}_i^1 = (z_{i-1}, z_i, z_{i+1}, z_{i+2})$. We remark that this multi-point expansion method on operator entry K_p with $p = 0, 1$ can be equivalently written as $K_p(\mathbf{Z}_i^p) = \tilde{K}_p(\tilde{\mathbf{Z}}^p_{i+\frac{p}{2}})$ to change the functional dependency from profiles at multiple local points $\mathbf{Z}_i^p$ to the value of the profile and its derivatives $\tilde{\mathbf{Z}}^p_{i+\frac{p}{2}}$ at the middle point $x_{i+\frac{p}{2}} = (x_i + x_{i+p})/2$ (see Appendix B for further details).

Next, we turn our attention to the constraints on the operator when the field z obeys a conservation law of the form $\frac{d}{dt}\int z(x)\,dx = 0$, such as the density in the diffusive system that will be discussed in Section 4. In this case, the operator entries should satisfy $\sum_j \left\langle \gamma_j, \mathcal{K}_z\gamma_i \right\rangle = 0$ (Li et al., 2019), and, thus

$$K_0(\mathbf{Z}_i^0) = -K_1(\mathbf{Z}_{i-1}^1) - K_1(\mathbf{Z}_i^1). \tag{11}$$

That is, the whole discretized operator can be written as a function of the off-diagonal entry K_1.

Finally, the discretized operator should be positive semi-definite for an arbitrary profile $z(x)$. Although the Cholesky decomposition is often used to guarantee that a given matrix is positive semi-definite, we here use an alternative approach, which will prove to be much simpler to implement. We first note that, by combining Eqs. (8) and (11), the evolution equation of the system can be re-written as

$$\sum_{i=j-1}^{j+1} \left\langle \gamma_j, \gamma_i \right\rangle \frac{\partial z_i}{\partial t} = -\Delta_b\left[K_1(\mathbf{Z}_j^1)\Delta_f Q_j\right], \tag{12}$$

where $\Delta_f B_i = B_{i+1} - B_i$ is the forward finite difference for an arbitrary field B and $\Delta_b B_i = B_i - B_{i-1}$ is the backward finite difference. Eq. (12) therefore suggests that $-\mathcal{K}_z$ is a second-order diffusion type operator with $-K_1$ acting as the mobility coefficient. As rigorously shown in Appendix A, the discretized operator can then be enforced to be positive semi-definite by simply requiring $K_1 \le 0$ for an arbitrary profile.

3.2. *Free energy parameterization and ensuing equation structure*

The thermodynamic driving force for systems governed by Eq. (2) is given by the functional derivative of the free energy, *i.e.*, $Q = \delta F/\delta z$. We will here assume, as it is common in continuum descriptions, that the free energy functional has a density f associated with it so that $F[z] = \int f\,dx$. Yet, we will allow for various dependencies of f on the profile z, which will result in different thermodynamic forces and, consequently, in different types of evolution equations.

The simplest case is that of $f = f(z)$, where the free energy density is only a function of the local profile, and the thermodynamic force is then given by

$$Q = f'(z). \tag{13}$$

For a discretized dissipative operator of the form of Eq. (12), the evolution equation will then be a standard second-order diffusion equation.

If the free energy density also depends on the local gradient, *i.e.*, $f = f(z, \nabla z)$, the thermodynamic force is then given by

$$Q = \frac{\partial f}{\partial z} - \nabla \cdot \frac{\partial f}{\partial \nabla z}. \tag{14}$$

We remark that if we use a first-order backward (or forward) finite difference scheme to discretize ∇z, the divergence in Eq. (14) should be expressed as a forward (or backward) scheme, respectively (see Appendix C for further details). In the discrete setting, f can be viewed as a function of two local points and Q as a function of three local points. Following the operator form given in Eq. (12), the evolution equation is now a fourth-order equation, such as a Cahn–Hilliard equation.

The free energy density may also have a dependency on higher-order derivatives. For instance, if it also depends on the Laplacian (denoted as $\nabla^2 z$ for simplicity for a one-dimensional system), *i.e.*, $f = f(z, \nabla z, \nabla^2 z)$, the thermodynamic force now includes an additional term,

$$Q = \frac{\partial f}{\partial z} - \nabla \cdot \frac{\partial f}{\partial \nabla z} + \nabla^2 \frac{\partial f}{\partial \nabla^2 z}. \tag{15}$$

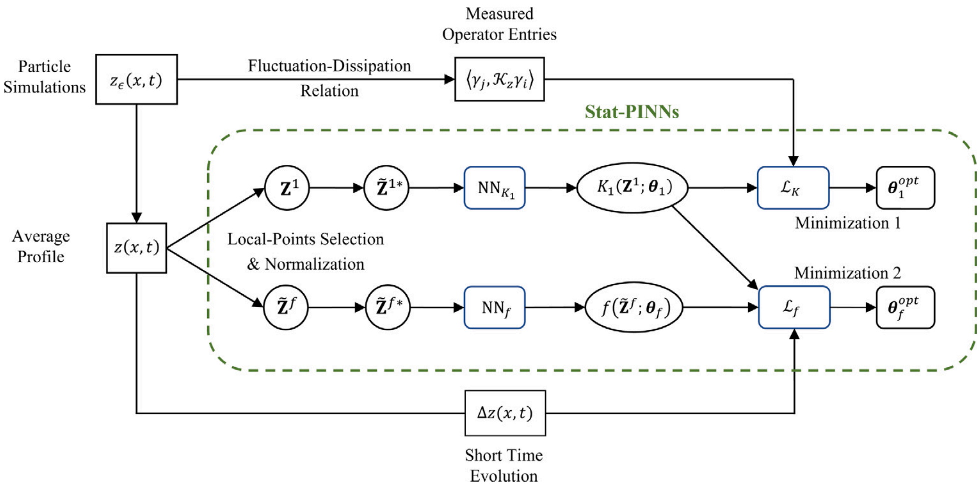

Fig. 2. Architecture for Stat-PINNs. Macroscopic dissipative evolution equations of the form $\frac{\partial z}{\partial t} = -\mathcal{K}_z \frac{\delta F[z]}{\delta z}$ are learned through the sequential training of NN_{K_1} and NN_f. These neural networks are aimed at learning the discretized operator $\mathcal{K}_z$ (in particular, the off-diagonal element of its tri-diagonal matrix representation) and the free energy density f ($F[z] = \int f dx$).

In the discrete setting, if we use a backward, forward, or central scheme to calculate ∇z and use a second-order central scheme for $\nabla^2 z$, the discretized f can be viewed as a function of three local points and Q as a function of five local points (see Appendix C for more details). Eq. (12) is now a sixth-order partial differential equation.

The above discussion highlights how higher-order terms in the dependence of the free energy density increase the order of the resulting PDE. Yet, it is interesting to note that, regardless of the order of the PDE (it being fourth- or even sixth- order), one can still use linear shape functions to discretize the dissipative operator, as this only contains second-order derivatives under the assumptions made in Section 3.1. Testing examples can be found in Appendix D.

3.3. Stat-PINNs architecture

We will now introduce a machine learning architecture, which we denote as Statistical-Physics-Informed Neural Networks (Stat-PINNs), to learn the coarse-grained dissipative evolution equation. This architecture is depicted in Fig. 2 and consists of two neural networks that will be trained sequentially. These neural networks are aimed at learning the discretized operator $\mathcal{K}_z$ and the free energy density f, based on the parameterizations discussed in Sections 3.1 and 3.2, respectively.

The first neural network is constructed to learn the off-diagonal entry K_1, as this is sufficient to fully characterize the discretized operator (see Section 3.1). Such an entry will be learned from multiple realizations of short-time particle simulations using Eqs. (9) and (11) so that both the diagonal and off-diagonal entries are well-approximated. We denote the measured data of the diagonal operator entries, off-diagonal operator entries and corresponding field values as $\left\{ K_0^{(s)}, K_1^{(s)}, \mathbf{Z}_{i^s}^{0(s)} \right\}_{s=1}^{N_K}$, where N_K is the number of training data and i^s denotes the spatial index for the sth sample. We remark that $\mathbf{Z}_{i^s}^{0(s)}$ includes both $\mathbf{Z}_{i^s-1}^{1(s)}$ and $\mathbf{Z}_{i^s}^{1(s)}$, which are the field values needed in the loss function, as will be seen next. The off-diagonal entry $K_1(\mathbf{Z}^1;\boldsymbol{\theta}_1)$ is represented through a neural network $\mathrm{NN}_{K_1}(\tilde{\mathbf{Z}}^{1*};\boldsymbol{\theta}_1)$, applied onto a non-positive function $g(\cdot)$, *i.e.*, $g(\mathrm{NN}_{K_1})$, to ensure the positive semi-definiteness of the operator. Here, $\tilde{\mathbf{Z}}^{1*}$ is the normalized version of $\mathbf{Z}^1 = \left\{ \mathbf{Z}_{i^s}^{1(s)} \right\}_{s=1}^{N_K}$ and θ_1 represents the trainable parameters. Further details on the relation between K_1 and NN_{K_1} can be found in Appendix E. Then, the training is performed by minimizing the following loss function

$$
\begin{aligned}
\mathcal{L}_K = & \frac{\lambda_0}{2N_0} \sum_{s=1}^{N_0} \left\| -K_1\left(\mathbf{Z}_{i^s-1}^{1(s)};\boldsymbol{\theta}_1\right) - K_1\left(\mathbf{Z}_{i^s}^{1(s)};\boldsymbol{\theta}_1\right) - K_0^{(s)} \right\|^2 \\
& + \frac{\lambda_1}{2N_1} \sum_{s=1}^{N_1} \left\| K_1\left(\mathbf{Z}_{i^s}^{1(s)};\boldsymbol{\theta}_1\right) - K_1^{(s)} \right\|^2 ,
\end{aligned}
\tag{16}
$$

where λ_0 and λ_1 are two adaptive loss weights determined automatically based on the neural tangent kernel method (Wang et al., 2022).

Once the dissipative operator $\mathcal{K}_z$ is learned from the first neural network, we can learn the free energy density $f(\tilde{\mathbf{Z}}^f)$ from macroscopic evolutions over a small time step Δt. We approximate f through a second neural network $\mathrm{NN}_f(\tilde{\mathbf{Z}}^{f*};\theta_f)$, where $\boldsymbol{\theta}_f$ denotes the trainable parameters for NN_f, $\tilde{\mathbf{Z}}^f$ denotes the value of the local field and potentially its spatial derivative, and $\tilde{\mathbf{Z}}^{f*}$ is the corresponding normalized version (see Appendix E for details). The training data is denoted as $\left\{ \mathbf{Z}_{i^s}^{Q(s)}, \Delta\mathbf{Z}_{i^s}^{Q(s)} \right\}_{s=1}^{N_f}$, where $\mathbf{Z}_{i^s}^{Q(s)}$ represents the local field values needed for computing the thermodynamic force $Q(\mathbf{Z}_{i^s}^{Q(s)};\theta_f)$ at point x_{i^s} through NN_f, and $\Delta\mathbf{Z}_{i^s}^{Q(s)}$

represents the time increments of these field values during the time interval Δt. We remark that both the local field values and their time increments are averaged from R realizations of macroscopically equivalent particle simulation. Here, the time interval Δt should be macroscopically small, but still much longer than the infinitesimal timestep h used for calculating the operator entries. Since the deterministic increment in Eq. (6) is proportional to Δt and the stochastic noise is proportional to $\sqrt{\Delta t}$, Δz will in general be noisy. We should therefore construct the loss function not only based on the deterministic Eq. (8) as Physics-Informed Neural Networks (PINNs) but also taking into account the effect of the stochastic noise term as (Dietrich et al., 2023)

$$\mathcal{L}_f = \frac{1}{2N_f}\sum_{s=1}^{N_f} \frac{\left\| \sum_i \langle \gamma_{j^s}, \gamma_i\rangle \Delta z_i^{(s)}/\Delta t + \sum_i \langle \gamma_{j^s}, \mathcal{K}_{z^{(s)}}\gamma_i\rangle Q(\mathbf{Z}_i^{Q(s)};\boldsymbol{\theta}_f)\right\|^2}{\left(\sigma^{(s)}_{Eq,j^s}\right)^2}. \tag{17}$$

Here, $\langle \gamma_{j^s}, \mathcal{K}_{z^{(s)}}\gamma_i\rangle$ is calculated from the trained neural network NN_{K_1} and the local field values, the thermodynamic forces Q_i are parameterized by the neural network NN_f according to the discussion in Section 3.2, and $\sigma^2_{Eq,j}$ represents the variance of the stochastic noise at point x_j, which is equal to $2\epsilon\langle \gamma_{j^s}, \mathcal{K}_{z^{(s)}}\gamma_{j^s}\rangle/(R\Delta t)$ (see Appendix F for further details). This denominator can normalize the noise at different points to a standard Gaussian distribution and is crucial for the training process. This is particularly the case when the noise is comparable to or dominant over the deterministic term and when the variance of noise varies among different data points.

In summary, the coarse-grained dissipative evolution equation can be learned by the sequential training processes of two neural networks NN_{K_1} and NN_f. First, the dissipative operator is learned from NN_{K_1} by minimizing the loss function $\mathcal{L}_\mathcal{K}$ (Eq. (16)). Then, the free energy is learned from NN_f by minimizing the loss function $\mathcal{L}_f$ (Eq. (17)) based on the knowledge of learned operator. Notice that we here apply fully-connected feed-forward neural networks for both NN_{K_1} and NN_f.

4. Example: Arrhenius interacting particles processes

4.1. Particle process and existing macroscopic model for Arrhenius diffusion

Arrhenius-type dynamics are characterized by an exponential dependence on an activation energy $\hat{E}_d$ and inverse temperature $\beta = 1/(k_B T)$, *i.e.*, $e^{-\beta \hat{E}_d}$, and arise in a wide range of phenomena, such as surface absorption, chemical reactions, or vacancies and intersticials in solids (Gibbs, 1972; Linderoth et al., 1997; Gilmer and Bennema, 1972; Laidler, 1984; Zhdanov, 1991; Dewey, 1994).

As an illustrative example of such type of dynamics, we here consider a stochastic jumping process for interacting particles on a one-dimensional lattice (Vlachos and Katsoulakis, 2000; Katsoulakis and Vlachos, 2003), where each lattice site can only be occupied by at most one particle, *i.e.*, the occupation number is $\eta(x) = 1$ if site x is occupied and $\eta(x) = 0$ if it is empty. The probability of a particle jumping event from site x to its nearest-neighbor y (left or right in 1D) is of Arrhenius type and given by $p(x \to y) = d\,\eta(x)\,(1-\eta(y))\,e^{-\beta \hat{U}(x)}$. Here, d is the jumping frequency, and $\hat{U}(x) = \hat{U}_0 + \sum_{\xi \neq x} \hat{J}(x-\xi)\eta(\xi)$, where $\hat{U}_0$ is the binding energy, and $\hat{J}(\cdot)$ describes the interaction energy between particles, which is attractive for $\hat{J} > 0$ and repulsive for $\hat{J} < 0$. Despite the apparent simplicity of this Arrhenius diffusion model, its analytic macroscopic description only exists for limited cases. Particularly, when the interaction between the particles is long range (compared to the lattice size), the macroscopic evolution for the particle density ρ is given by Vlachos and Katsoulakis (2000)

$$\frac{\partial \rho}{\partial t} = \nabla \cdot \left(m[\rho]\,\nabla \frac{\delta F[\rho]}{\delta \rho} \right) \quad \text{and} \quad m[\rho] = D\rho(1-\rho)e^{-J*\rho}, \tag{18}$$

where $D = d\,e^{-\beta \hat{U}_0}$, $J(\cdot) = \beta \hat{J}(\cdot)$ is the dimensionless interaction energy, and $J * \rho = \int J(x-\xi)\rho(\xi)\mathrm{d}\xi$ denotes the convolution between J and ρ. Furthermore, F is the (dimensionless) Helmholtz free energy given by

$$F[\rho] = -\int \frac{1}{2}\rho\,(J*\rho)\,\mathrm{d}x + \int [\rho \ln \rho + (1-\rho)\ln(1-\rho)]\,\mathrm{d}x, \tag{19}$$

which consists of a nonlocal energy term and a cross entropy term. When the interaction potential is symmetric, *i.e.*, $J(x) = J(-x)$, the thermodynamic driving force can be written as,

$$Q = \frac{\delta F}{\delta \rho} = -J * \rho - \ln\left(\frac{1}{\rho} - 1\right). \tag{20}$$

Moreover, the fluctuations of the particle jumping process can be accounted for by adding the stochastic noise term $\nabla\cdot\left(\sqrt{2\epsilon m[\rho]}\dot{W}_{x,t}\right)$ to Eq. (18) (Vlachos and Katsoulakis, 2000). That is, the macroscopic description and its stochastic counterpart are of the form of Eq. (2) and (6), respectively, with the following dissipative operator

$$\mathcal{K}_\rho = -\nabla \cdot (m[\rho]\,\nabla) = -\nabla \cdot \left[D\rho(1-\rho)e^{-J*\rho}\nabla\right]. \tag{21}$$

Further details for the weak form of this long-range model can be found in Appendix G.

A specific case of the above dynamics is that of no interaction between the particles, *i.e.*, $\hat{U}(x) = 0$. Such particle dynamics is known as the symmetric simple exclusion process (SSEP) (Embacher et al., 2018; Adams et al., 2013; Huang et al., 2020). Even though this "interaction" is not long-range, its coarse-grained evolution (both the PDE and its stochastic version) can also be given by the equations above with $J = 0$ and $\tilde{U}_0 = 0$.

Beyond this long-range model described by Eqs. (18)–(19), our understanding of the macroscopic evolution equations for the case with short-range interactions is still limited. The coarse-grained free energy expression has only been obtained from statistical mechanics for the case of nearest-neighbor interactions (Katsoulakis and Vlachos, 2003), whilst the analytical expression for the dissipative operator, or the full dynamics, for such nearest-neighbor interaction case is still unknown. We will thus aim to capture the full dynamics of the Arrhenius process with different interactions by the proposed Stat-PINNs framework in the following section.

4.2. Results

We consider the Arrhenius particle process just described with $d = 1$, $U_0 = 0$ and a symmetric step function interaction defined as $J(x) = J_0$ for $0 < |x| \leq L$ and $J(x) = 0$ otherwise. Here, we will test our Stat-PINNs framework for three different cases of the Arrhenius diffusion process. The first one is chosen to have long-range interactions between the particles, so as to enable a direct comparison with the known analytic solution. The other two cases will correspond to weak and strong short-range interactions, and are aimed at testing the capability of the method to discover their continuum thermodynamic model, previously unknown, to the best of the authors' knowledge. In all cases, the results will be directly compared to the results from traditional PINNs, where no recourse is made to statistical mechanics. To perform a fair comparison and isolate the contribution from statistical mechanics, the same thermodynamic structure and parameterization will be used. Specifically, two neural networks for $\mathcal{K}_1$ and f will be jointly trained using the loss function given by Eq. (17), but without normalizing the residuals with the equation variance (*i.e.*, setting the denominator $\sigma^2_{Eq} = 1$). This is analogous to previous integrations of PINNs with GENERIC, in the sense that the operator and thermodynamic potential are jointly trained.

For each case, the data is collected from a one-dimensional particle system with $N_b = 2000$ bins and 25 shape functions using KMC particle simulations starting from 28 different cosine initial profiles. This particle system is then rescaled to the macroscopic range $x \in [0, 0.5]$, indicating a lattice size $\epsilon = 2.5 \times 10^{-4}$ and a spatial discretization $\Delta x_\gamma = 0.02$. The restriction to the domain $[0, 0.5]$ is motivated by symmetry considerations to facilitate the particle simulations used for comparison, see Appendix H. Each data point for the operator entries is calculated from 10 sequential h intervals in $R = 10^4$ realizations, *i.e.*, expectations are approximated by performing averages over $R_K = 10^5$ realizations for each initial profile, where the small time interval h is taken as $h = 0.01\epsilon^2\tau$, with $\tau = \exp\left(2J_0 L\rho_{\max}\right)$ a factor that depends on the maximum density $\rho_{\max}$ of each simulated profile. Each data point for learning the free energy is generated from these same $R = 10^4$ realizations for each initial profile, within a short time interval $\Delta t \gg h$, chosen to be about $20\epsilon^2\tau \sim 40\epsilon^2\tau$ depending on each case. More details about the particle simulations and corresponding parameter settings can be found in Appendix H. In all three cases below, we set the off-diagonal entry K_1 as a function of two local points, or equivalently as $\tilde{K}_1(\rho, \nabla\rho)$, with ρ and $\nabla\rho$ evaluated at the middle point of the support, and the free energy density as $f(\rho)$, indicating that the input of the first neural network NN_{K_1} has a dimension of two and that of the second neural network NN_f has a dimension of one. Furthermore, all neural networks are chosen to have two hidden layers with 20 neurons per layer. More training details are included in Appendix J. Using the learned operator and free energy, Eq. (8) is then solved using a forward Euler scheme and the results are compared to those stemming from the long-range analytical model (Eqs. (18) and (19)) together with the average of 600 realizations from direct KMC simulations using the Bortz–Kalos–Lebowitz (BKL) algorithm (Bortz and Lebowitz, 1975). The macroscopic predictions are solved on $x \in [0, 1]$ with spatial discretization $\Delta x_\gamma = 0.02$ and periodic boundary conditions. The initial condition chosen to test the predictive capability of Stat-PINNs is a triangular profile, and hence distinct from the training profiles. We remark that while the training profiles are simulated only for a Δt time interval, the macroscopic simulations can be performed over an arbitrary large time interval, as well as arbitrary initial conditions, without needing to perform extrapolations, as long as the macroscopic evolution lies within the trained phase space, *e.g.*, range of ρ and $\nabla\rho$ explored within the profiles used in the training process.

All KMC particle simulations are coded in C++ and run on Intel(R) Xeon(R) CPUs E5-2683 v4 @ 2.10 GHz. Stat-PINNs and the corresponding continuum predictions are coded in Python. JAX is the main Python library for implementing neural networks. Other standard libraries such as Numpy, JAX Numpy, Scipy, and Matplotlib are used for data pre- and post-processing.

Fig. 3 shows the coarse-grained results from a particle system with long-range interactions with $J_0 L = 0.9$ and $L = 40\epsilon$, *i.e.*, each particle interacts with other particles within 40 neighboring lattice sites on each side. Given that the interaction is weak, we regard the dynamics as local within numerical accuracy even though the interaction range $L = \Delta x_\gamma/2$ is long from a particle perspective. Figs. 3a-3c show the comparison between the Stat-PINNs result (blue) and long-range analytic model (yellow) for the off-diagonal entry $\tilde{K}_1$, shown for two different ranges, and the free energy density. The latter is calibrated by setting its value and that of its derivative to zero at $\rho = 0.5$ in order to cancel the non-uniqueness due to the structure of the operator and the thermodynamic force Q (see Appendix K for details), and it is denoted as $\tilde{f}(\rho)$ after calibration. We see that the prediction of Stat-PINNs (blue) has an excellent agreement with the long-range analytic model (yellow), exhibiting a relative L^2 error of 0.59% for $\tilde{K}_1$ and 2.56% for $\tilde{f}$. In contrast, PINNs (orange) cannot uniquely identify the true operator and free energy. Furthermore, the operator entry identified by PINNs differs in magnitude for low and high densities, exhibits an opposite trend, and is non-smooth. The learned free energy from PINNs even lacks symmetry. The learned operator and free energy from both Stat-PINNs and PINNs are used to solve Eq. (8) for a triangular wave initial profile with average density $\rho_{ave} = 0.5$, amplitude $A = 0.45$ and frequency $f = 2$, and the results are compared to that of the analytical long-range evolution equation as well as direct particle simulations. The total time simulated is $t_{tot} = 0.08$, and the timestep used in the discretization is $\Delta t_{sim} = \Delta x^2_\gamma/5 = 8 \times 10^{-5}$. Figs. 3d-3f show the results for the macroscopic evolution. Both the prediction from Stat-PINNs (blue solid line) and the solution from the long-range analytic model (yellow dotted line) agree with the KMC particle simulation (purple dots). In contrast, there exists a noticeable error in the prediction from PINNs

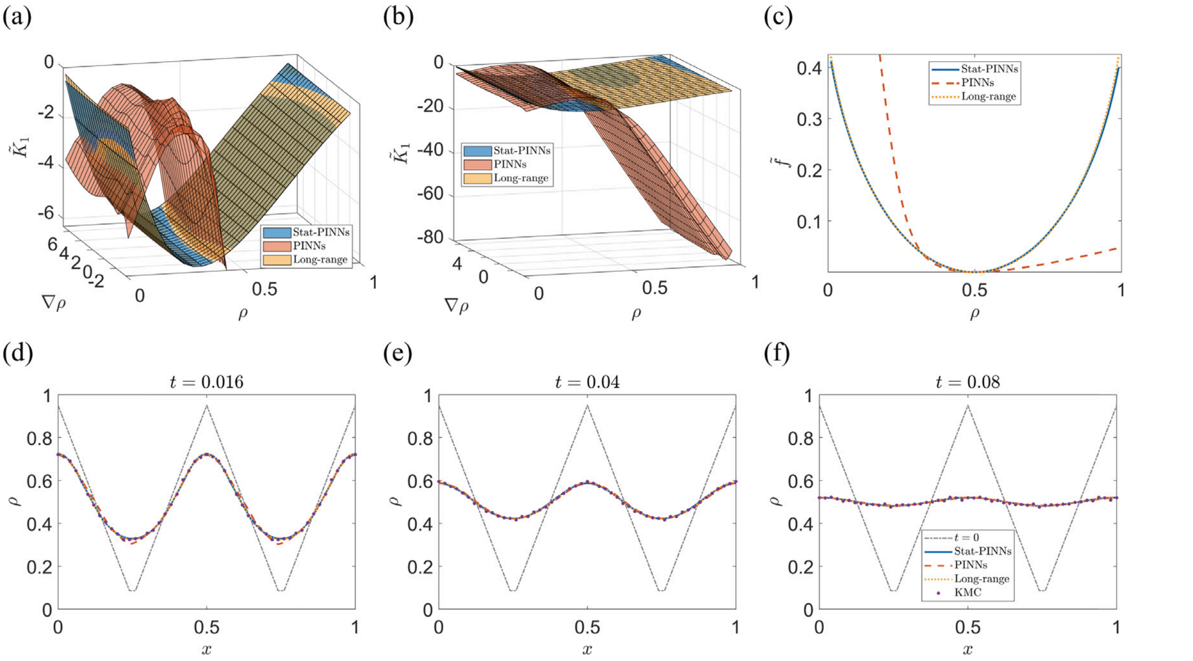

Fig. 3. Results for the Arrhenius process with $JL = 0.9, L = 40\epsilon$, including (a, b) the dissipative operator entry $\tilde{K}_1$ plotted in two different ranges, (c) the calibrated free energy density $\tilde{f}$, and (d-f) snapshots of the macroscopic evolution starting from a triangular wave initial profile (black dash-dotted line). Predictions from Stat-PINNs (blue surfaces or blue solid lines), PINNs (orange surfaces or orange dashed lines), and a long-range analytic model (yellow surfaces or yellow dotted lines) are shown. Results from KMC particle simulations (purple dots) are used as the true macroscopic evolution for comparison.

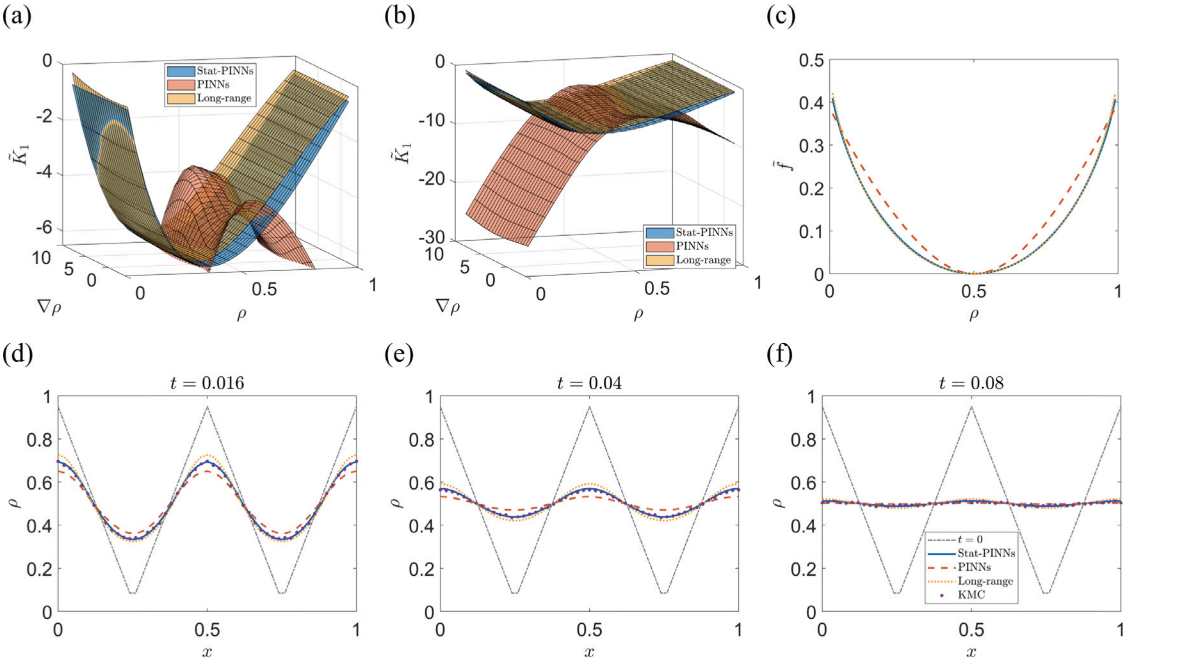

Fig. 4. Results for Arrhenius process with $JL = 0.9, L = 2\epsilon$, including (a, b) dissipative operator entry $\tilde{K}_1$ plotted in two different ranges, (c) calibrated free energy density $\tilde{f}$, and (d-f) snapshots of macroscopic evolution starting from a triangular wave initial profile (black dash-dotted line). Predicting methods include Stat-PINNs (blue surfaces or blue solid lines), PINNs (orange surfaces or orange dashed lines), and a long-range analytic model (yellow surfaces or yellow dotted lines). Results from KMC particle simulation (purple dots) are used as true macroscopic evolution for comparison.

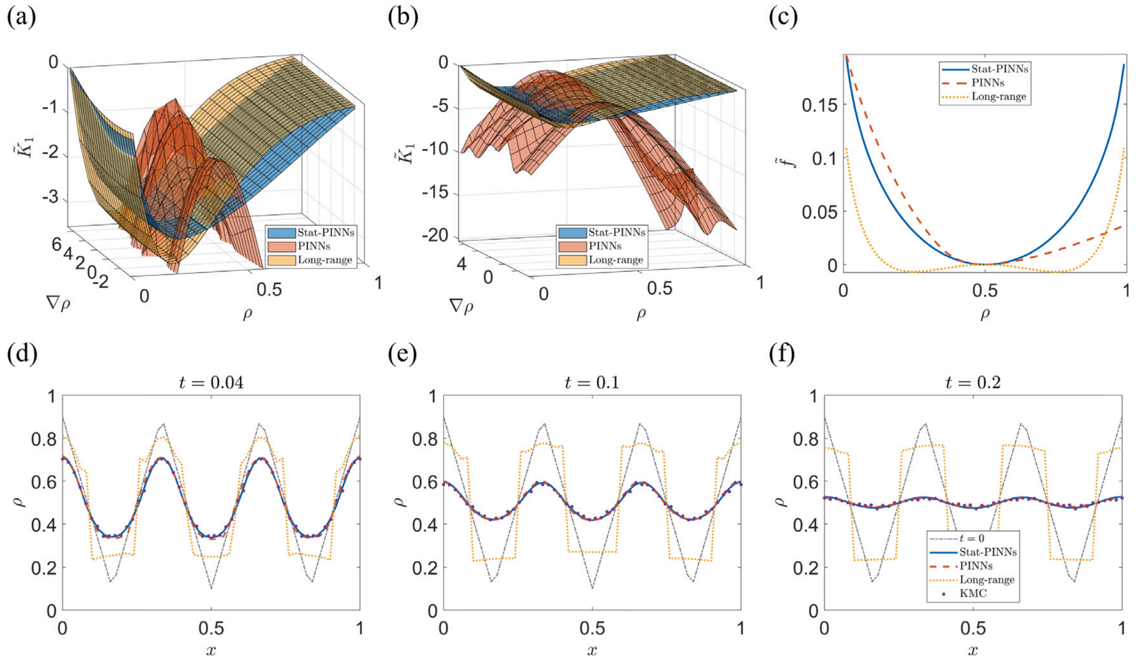

Fig. 5. Results for Arrhenius process with $JL = 2.2, L = 2\epsilon$, including (a, b) dissipative operator entry $\tilde{K}_1$ plotted in two different ranges, (c) calibrated free energy density $\tilde{f}$, and (d-f) snapshots of macroscopic evolution starting from a triangular wave initial profile (black dash-dotted line). Predicting methods include Stat-PINNs (blue surfaces or blue solid lines), PINNs (orange surfaces or orange dashed lines), and a long-range analytic model (yellow surfaces or yellow dotted lines). Results from KMC particle simulation (purple dots) are used as true macroscopic evolution for comparison.

(orange dashed line) at the beginning of the evolution. These results therefore show that Stat-PINNs can not only predict the correct coarse-grained macroscopic evolution but also learn the true thermodynamics and kinetics of the system.

Fig. 4 shows the results from a short-range weakly interacting Arrhenius process with $J_0 L = 0.9$ and $L = 2\epsilon$, where the long-range analytic model is not expected to work and no analytic solution is known, to the best of the authors' knowledge. Figs. 4a-4c show the comparison between the results from PINNs (orange), Stat-PINNs (blue) and the long-range analytical model (yellow) for the off-diagonal entry $\tilde{K}_1$, shown for two different ranges, and the calibrated free energy density $\tilde{f}(\rho)$, respectively. While the predicted free energy from Stat-PINNs agrees with Eq. (19) with a 1.85% relative L^2 error, the predicted operator is different from the long-range model as could be expected (there is a 13.83% L^2 relative difference). Predictions for the macroscopic evolution of a triangular wave initial profile with average density $\rho_{ave} = 0.5$, amplitude $A = 0.45$ and frequency $f = 2$ are shown in Figs. 4d-4f. These are solved for a total time $t_{tot} = 0.08$ using a timestep $\Delta t_{sim} = \Delta x_\gamma^2/5 = 8 \times 10^{-5}$. While the Stat-PINNs' prediction perfectly agrees with the true evolution from KMC simulations, the results from PINNs exhibit a noticeable error and so does as well the long-range analytical model (particularly in the higher density range). In particular, the long-range analytical prediction appears to lag behind the true evolution (which coincides with that of Stat-PINNs). This difference can be explained from the predicted operator entry shown in Fig. 4a, which can be physically interpreted as a mobility coefficient. The value of K_1 predicted with Stat-PINNs has a larger absolute value compared to the long-range model, and this difference is more significant in the higher density range.

Fig. 5 shows the results from a short-range strongly interacting Arrhenius particle process with $J_0 L = 2.2$ and $L = 2\epsilon$. As shown in Figs. 5a-5c, the predictions for both the off-diagonal operator entry $\tilde{K}_1$ and the free energy $\tilde{f}$ are very different from the long-range model, as could be anticipated. Particularly, while the long-range model has a double-well free energy, which suggests that the system may exhibit phase separation, the predicted free energy is clearly single-welled and is not suggestive of phase separation. This observation is in agreement with the macroscopic evolution shown in Figs. 5d-5f. Here the initial profile is chosen as a triangular wave with average density $\rho_{ave} = 0.5$, amplitude $A = 0.4$ and frequency $f = 3$ and the simulation is run for a total time of $t_{tot} = 0.2$ and a timestep of $\Delta t_{sim} = \Delta x_\gamma^2/2 = 2{\times}10^{-4}$. While the long-range model reached a steady-state with a bistable profile, the Stat-PINNs' prediction decays to the average value, in agreement with the particle simulations. Although the results from PINNs capture the overall trend of the macroscopic evolution, noticeable errors exist at the beginning of the simulations.

We remark that in the three particle processes discussed, the errors observed in the macroscopic dynamics predicted by PINNs can be eliminated by increasing the number of time intervals Δt used for training the networks (see Appendix L), although this will naturally entail an increased computational cost. This therefore indicates that statistical mechanics relations can increase the robustness of Stat-PINNs, particularly when the amount of data is limited and the noise is not negligible. In any case, macroscopic data, regardless of how extensive this is, is insufficient to characterize the physical free energy and dissipative operator in the examples considered.

Table 1
Computational costs for Stat-PINNs compared with KMC particle simulations. Here the numbers with units of days represent the total cost for all realizations as if they were performed in the absence of any parallelization and the number in the following parenthesis denotes the maximum factor for increasing computational efficiency by parallelizing all realizations.

Case	Stat-PINNs Data Collection	Stat-PINNs Training	Stat-PINNs Prediction	KMC for Full Model
$J_0 L = 0.9$ $L = 40\epsilon$	55 days ($\div 28 \div 10^4$)	~ 20 min	< 5 min	3500 days ($\div 600$)
$J_0 L = 0.9$ $L = 2\epsilon$	19 days ($\div 28 \div 10^4$)	~ 20 min	< 5 min	875 days ($\div 600$)
$J_0 L = 2.2$ $L = 2\epsilon$	39 days ($\div 28 \div 10^4$)	~ 20 min	< 5 min	625 days ($\div 600$)

Finally, we would like to comment on the computational cost for Stat-PINNs and the KMC particle simulations, highlighted in Table 1. As is to be expected, direct particle simulations of long-time macroscopic evolutions are extremely costly, even for the one-dimensional example chosen. Here, parallelization is only done for independent realizations (600 in this case), as causality severely hinders parallelizations in time. For the specific particle systems chosen, the long-time KMC simulations performed took of the order of 1 to 6 days at full parallelization (or 1 to 10 years without any parallelization). With the Stat-PINNs strategy, the training and macroscopic prediction can all be done within 30 min. Here, the computational cost is dominated by the data collection process, which consists of a large number (28 profiles times $R = 10^4$ realizations) of independent short-time trajectories. For the most ideal case, where independent trajectories are fully parallelized, data collection can be done within a minute. And even in the most undesired case, where no parallelization is available, the computational cost of Stat-PINNs is still at least one to two orders of magnitude faster than particle simulations. Furthermore, Stat-PINNs can be applied to predict the evolution of different initial profiles and different boundary conditions, at minimal cost. Finally, we want to comment that although all particle simulations mentioned above are performed using the BKL algorithm, other algorithms with more efficient searching (Schulze, 2008) or parallelization strategies (Arampatzis et al., 2011) could be applied to further reduce the computational cost for the data collection in Stat-PINNs.

5. Conclusions and discussions

In this work, we proposed a machine learning architecture called Statistical-Physics-Informed Neural Networks (Stat-PINNs) for learning purely dissipative evolution equations of GENERIC type from short-time particle simulations. It consists of two neural networks that are sequentially trained to learn the unique discretized dissipative operator and free energy density of the system, through a carefully designed structure-preserving parameterization method. More specifically, the proposed strategy strongly ensures that the dissipative operator is symmetric (as required by Onsager's reciprocity relations), positive semi-definite (in accordance with the second law of thermodynamics), and that the ensuing dynamics are conserved, when applicable, as is the case of diffusive phenomena. The key idea behind the proposed framework is that field fluctuations, in addition to the average particle dynamics, contain crucial information for tackling the inverse problem of thermodynamic model discovery from data. Such fluctuations are leveraged in two ways. First, the fluctuation–dissipation relation is used to uniquely determine the dissipative operator. Second, the evolution equation is treated as a stochastic differential equation when defining the loss function used for learning the free energy. That is, in contrast to traditional PINNs, the loss function contains a denominator that is related to the stochastic noise estimated from statistical physics to normalize the equation's residue. The Stat-PINNs architecture is demonstrated over a particle process with three different Arrhenius-type interactions, of which only one has an analytically known macroscopic evolution equation. The predicted continuum evolution from Stat-PINNs perfectly agrees with the true particle dynamics in all cases, while PINNs exhibit noticeable deviations from the true evolution. Although such deviations can be reduced by increasing the length of the particle simulations and gathering more training data along the particle trajectories (see Appendix L), the unique free energy and operator of the system cannot be learned by PINNs. Stat-PINNs may thus be regarded as more physically interpretable and also more robust when temporal data from particle simulations is limited.

CRediT authorship contribution statement

Shenglin Huang: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Zequn He:** Writing – original draft, Software, Investigation. **Nicolas Dirr:** Writing – review & editing, Investigation, Formal analysis. **Johannes Zimmer:** Writing – review & editing, Investigation, Funding acquisition, Formal analysis. **Celia Reina:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

S. H., Z. H. and C. R. gratefully acknowledge support from National Science Foundation CMMI-2047506, seed grant from (MRSEC) DMR-1720530, and the US Department of the Army W911NF2310230. J. Z. also acknowledges support from W911NF2310230. The authors further thank Mr. Travis Leadbetter for his valuable insight to encode the positive semi-definiteness in the proposed architecture, and Prof. Markos Katsoulakis for suggesting the Arrhenius process as an important case study and sharing his knowledge on the coarse-graining of this particle system.

Appendix A. Discussion on the nonlocal operator beyond the tri-diagonal form

When the dissipative operator $\mathcal{K}_z$ is nonlocal in the thermodynamic force Q, the discretized operator is no longer a tri-diagonal matrix and more non-zero off-diagonal entries need to be considered. In this case, the parameterization method introduced in Section 3.1 may still be utilized. In general, we can assume that the operator entry $\left\langle \gamma_j, \mathcal{K}_z \gamma_i \right\rangle \neq 0$ only when $j \in \{i, i \pm 1, \dots, i \pm n_K\}$ with n_K being a positive integer. Then, all non-zeros operator entries can be written as

$$K_p(\mathbf{Z}_i^p) = \left\langle \gamma_{i+p}, \mathcal{K}_z \gamma_i \right\rangle = \left\langle \gamma_i, \mathcal{K}_z \gamma_{i+p} \right\rangle, \tag{A.1}$$

where $p = 0, 1, \dots, n_K$ and $\mathbf{Z}_i^p$ represents the profile z at multiple local points within the non-local dependence on z for calculating the integral.

Furthermore, when z satisfies a conservation law, K_0 can be expressed as

$$K_0(\mathbf{Z}_i^0) = -\sum_{p=1}^{n_K} \left[K_p(\mathbf{Z}_{i-p}^p) + K_p(\mathbf{Z}_i^p) \right], \tag{A.2}$$

analogously to Eq. (11) for the fully local case. Then, n_K neural networks, representing the n_K off-diagonal entries would be required to learn the operator.

As mentioned in Section 3.1, when z obeys a conservation law and $\mathcal{K}_z$ is local on Q (indicating a tri-diagonal form), the weak form of the evolution equation can be written as Eq. (12), indicating a diffusive equation. In comparison, when $\mathcal{K}_z$ is nonlocal on Q, the evolution equation (Eq. (8)) can be re-written as

$$\begin{aligned} \sum_a \langle \gamma_b, \gamma_a \rangle \frac{\partial z_a}{\partial t} &= -\sum_{p=1}^{n_K} \left\langle \gamma_{b+p}, \mathcal{K}_z \gamma_b \right\rangle \left(Q_{b+p} - Q_b \right) + \sum_{p=1}^{n_K} \left\langle \gamma_b, \mathcal{K}_z \gamma_{b-p} \right\rangle \left(Q_b - Q_{b-p} \right) \\ &= -\sum_{p=1}^{n_K} \Delta_b^{(p)} \left[\left\langle \gamma_{b+p}, \mathcal{K}_z \gamma_b \right\rangle \Delta_f^{(p)} Q_b \right], \end{aligned} \tag{A.3}$$

where $\Delta_f^{(p)} B_i = B_{i+p} - B_i$ is the p-step forward finite difference for an arbitrary variable B and $\Delta_b^{(p)} B_i = B_i - B_{i-p}$ is the p-step backward finite difference. Eq. (A.3) can be viewed as a generalized version of Eq. (12) with nonlocal diffusion behavior, where different off-diagonal entries serve as mobility functions corresponding to different nonlocal spatial derivatives. Eq. (A.3) can also be written in matrix form as

$$\mathbf{M}\dot{\mathbf{z}} = -\sum_{p=1}^{n_K} \left[\mathbf{D}^{(p)} \mathbf{\Lambda}^{(p)} \left(-\mathbf{D}^{(p)\mathbf{T}} \right) \right] \mathbf{Q}, \tag{A.4}$$

where

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{N_\gamma} \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{N_\gamma} \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \langle \gamma_1, \gamma_1 \rangle & \langle \gamma_1, \gamma_2 \rangle & \cdots & \langle \gamma_1, \gamma_{N_\gamma} \rangle \\ \langle \gamma_2, \gamma_1 \rangle & \langle \gamma_2, \gamma_2 \rangle & \cdots & \langle \gamma_2, \gamma_{N_\gamma} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \gamma_{N_\gamma}, \gamma_1 \rangle & \langle \gamma_{N_\gamma}, \gamma_2 \rangle & \cdots & \langle \gamma_{N_\gamma}, \gamma_{N_\gamma} \rangle \end{pmatrix},$$

$$\mathbf{\Lambda}^{(p)} = \begin{pmatrix} K_{1+p,1} & 0 & \cdots & 0 \\ 0 & K_{2+p,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & K_{pN_\gamma} \end{pmatrix} \text{ with } K_{ji} = \left\langle \gamma_j, \mathcal{K}_z \gamma_i \right\rangle,$$

$$\text{and} \quad \mathbf{D}^{(p)} = \left(D_{ij}^{(p)} \right)_{N_\gamma \times N_\gamma} \text{ with } D_{ij}^{(p)} = \begin{cases} 1, & i = j \\ -1, & i = j + p \text{ or } i = j + p - N_\gamma \\ 0 & \text{otherwise} \end{cases}$$

Here $\mathbf{D}^{(p)}$ and $-\mathbf{D}^{(p)\mathbf{T}}$ represent the p-step backward and forward finite difference matrices, respectively. We note that we have applied periodic boundary conditions here for simplicity.

Finally, the discretized operator should be positive semi-definite. If the operator is local on Q, (being a tridiagonal form), a positive semi-definite $\mathcal{K}_z$ requires

$$\mathbf{u}^{\mathbf{T}} \mathbf{D}^{(1)} \mathbf{\Lambda}^{(1)} \left(-\mathbf{D}^{(1)\mathbf{T}} \right) \mathbf{u} = -\left(\mathbf{D}^{(1)\mathbf{T}} \mathbf{u} \right)^{\mathbf{T}} \mathbf{\Lambda}^{(1)} \left(\mathbf{D}^{(1)\mathbf{T}} \mathbf{u} \right) \geq 0, \tag{A.5}$$

for an arbitrary vector $\mathbf{u}$. This can be simplified to $K_1(\mathbf{Z}_i^1) \le 0$ when $\mathcal{K}_z$ is local on the thermodynamic force Q. Such a practical strategy to encode the positive semi-definiteness when $\mathcal{K}_z$ is nonlocal on Q has not yet been discovered.

Appendix B. Multi-point expansion and middle-point expansion for the discretized operator

In Section 3.1, we introduced the multi-point expansion method to parameterize the discrete operator entry as $K_p(\mathbf{Z}_i^p) = \left\langle \gamma_{i+p}, \mathcal{K}_z \gamma_i \right\rangle = \left\langle \gamma_i, \mathcal{K}_z \gamma_{i+p} \right\rangle$. This method is also mathematically equivalent to expanding the same entry at the middle point $x_{i+\frac{p}{2}} = (x_i + x_{i+p})/2$,

$$K_p(\mathbf{Z}_i^p) = \tilde{K}_p(\tilde{\mathbf{Z}}^p_{i+\frac{p}{2}}), \tag{B.1}$$

where $\tilde{K}_p$ represents the middle-point expansion form of K_p, and $\tilde{\mathbf{Z}}^p_{i+\frac{p}{2}}$ represents the corresponding values of the local profile and its derivatives. For example, when $\mathcal{K}_z$ is local on both z and Q, the two operator entries K_0 and K_1 can be equivalently written as

$$\begin{aligned} K_0(z_{i-1}, z_i, z_{i+1}) &= \tilde{K}_0(z_i, \nabla z|_i\,, \nabla^2 z\big|_i), \\ K_1(z_i, z_{i+1}) &= \tilde{K}_1(z_{i+\frac{1}{2}}, \nabla z|_{i+\frac{1}{2}}), \\ \text{with} \quad \nabla z|_i = \frac{z_{i+1} - z_{i-1}}{2\Delta x_\gamma}, &\qquad \nabla^2 z\big|_i = \frac{z_{i+1} - 2z_i + z_{i-1}}{\Delta x_\gamma^2}, \\ \text{and} \quad z_{i+\frac{1}{2}} = \frac{z_i + z_{i+1}}{2}, &\qquad \nabla z|_{i+\frac{1}{2}} = \frac{z_{i+1} - z_i}{\Delta x_\gamma}. \end{aligned} \tag{B.2}$$

Appendix C. Numerical schemes for the free energy and thermodynamic force

In Section 3.2, we introduced the expression of the thermodynamic force Q when the free energy has different functional dependencies. In this section, we will discuss the expression of f and Q in the discrete setting.

Here we assume that the free energy density can be written as $f = f(z, \nabla z, \nabla^2 z)$. If we use a backward scheme for the gradient, $\nabla z|_i = (z_i - z_{i-1})/\Delta x_\gamma$, and a central scheme for the Laplacian, $\nabla^2 z\big|_i = (z_{i-1} - 2z_i + z_{i+1})/\Delta x_\gamma^2$, and denote $f_i = f(z_i, \nabla z|_i\,, \nabla^2 z\big|_i)$, the variation of the total free energy functional can be given by

$$\begin{aligned} \delta F &= \sum_{i=1}^{N_\gamma} \left(f_{i,z}\delta z_i + f_{i,\nabla z}\frac{\delta z_i - \delta z_{i-1}}{\Delta x_\gamma} + f_{i,\nabla^2 z}\frac{\delta z_{i+1} - 2\delta z_i + \delta z_{i-1}}{\Delta x_\gamma^2} \right) \Delta x_\gamma \\ &= \sum_{i=1}^{N_\gamma} \left[f_{i,z}\delta z_i + \left(f_{i,\nabla z} - \frac{f_{i,\nabla^2 z}}{\Delta x_\gamma} \right) \frac{\delta z_i - \delta z_{i-1}}{\Delta x_\gamma} + \frac{f_{i,\nabla^2 z}}{\Delta x_\gamma}\frac{\delta z_{i+1} - \delta z_i}{\Delta x_\gamma} \right] \Delta x_\gamma, \end{aligned} \tag{C.1}$$

where $f_{i,z} = \partial f_i / \partial z$ and similarly for $f_{i,\nabla z}$ and $f_{i,\nabla^2 z}$. For simplicity, we denote $r_i = f_{i,\nabla z} - f_{i,\nabla^2 z}/\Delta x_\gamma$. Then, the summation of the second term in the square bracket above can be given by

$$\begin{aligned} \sum_{i=1}^{N_\gamma} \left(r_i \frac{\delta z_i - \delta z_{i-1}}{\Delta x_\gamma} \right) \Delta x_\gamma &= \sum_{i=1}^{N_\gamma} \left(\frac{r_i \delta z_i - r_{i-1}\delta z_{i-1}}{\Delta x_\gamma} + \frac{r_{i-1}\delta z_{i-1} - r_i \delta z_{i-1}}{\Delta x_\gamma} \right) \Delta x_\gamma \\ &= \frac{r_{N_\gamma}\delta z_{N_\gamma} - r_0 \delta z_0}{\Delta x_\gamma} + \sum_{i=1}^{N_\gamma} \left(\frac{r_{i-1}\delta z_{i-1} - r_i \delta z_{i-1}}{\Delta x_\gamma} \right) \Delta x_\gamma \\ &= -\sum_{i=1}^{N_\gamma} \left(\frac{r_{i+1} - r_i}{\Delta x_\gamma} \right) \delta z_i \Delta x_\gamma, \end{aligned} \tag{C.2}$$

where the first term in the second line is cancelled out when applying periodic boundary conditions for z. Similarly, the third term in the square bracket of Eq. (C.1) can be expressed as

$$\sum_{i=1}^{N_\gamma} \left(\frac{f_{i,\nabla^2 z}}{\Delta x_\gamma} \frac{\delta z_{i+1} - \delta z_i}{\Delta x_\gamma} \right) \Delta x_\gamma = -\sum_{i=1}^{N_\gamma} \left(\frac{f_{i,\nabla^2 z} - f_{i-1,\nabla^2 z}}{\Delta x_\gamma^2} \right) \delta z_i \Delta x_\gamma. \tag{C.3}$$

Combining the above results, Eq. (C.1) can be simplified as

$$\begin{aligned} \delta F &= \sum_{i=1}^{N_\gamma} \left(f_{i,z} - \frac{r_{i+1} - r_i}{\Delta x_\gamma} - \frac{f_{i,\nabla^2 z} - f_{i-1,\nabla^2 z}}{\Delta x_\gamma^2} \right) \delta z_i \Delta x_\gamma \\ &= \sum_{i=1}^{N_\gamma} \left(f_{i,z} - \frac{f_{i+1,\nabla z} - f_{i,\nabla z}}{\Delta x_\gamma} + \frac{f_{i+1,\nabla^2 z} - f_{i,\nabla^2 z}}{\Delta x_\gamma^2} - \frac{f_{i,\nabla^2 z} - f_{i-1,\nabla^2 z}}{\Delta x_\gamma^2} \right) \delta z_i \Delta x_\gamma \\ &= \sum_{i=1}^{N_\gamma} \left(f_{i,z} - \frac{f_{i+1,\nabla z} - f_{i,\nabla z}}{\Delta x_\gamma} + \frac{f_{i+1,\nabla^2 z} - 2f_{i,\nabla^2 z} + f_{i-1,\nabla^2 z}}{\Delta x_\gamma^2} \right) \delta z_i \Delta x_\gamma. \end{aligned} \tag{C.4}$$

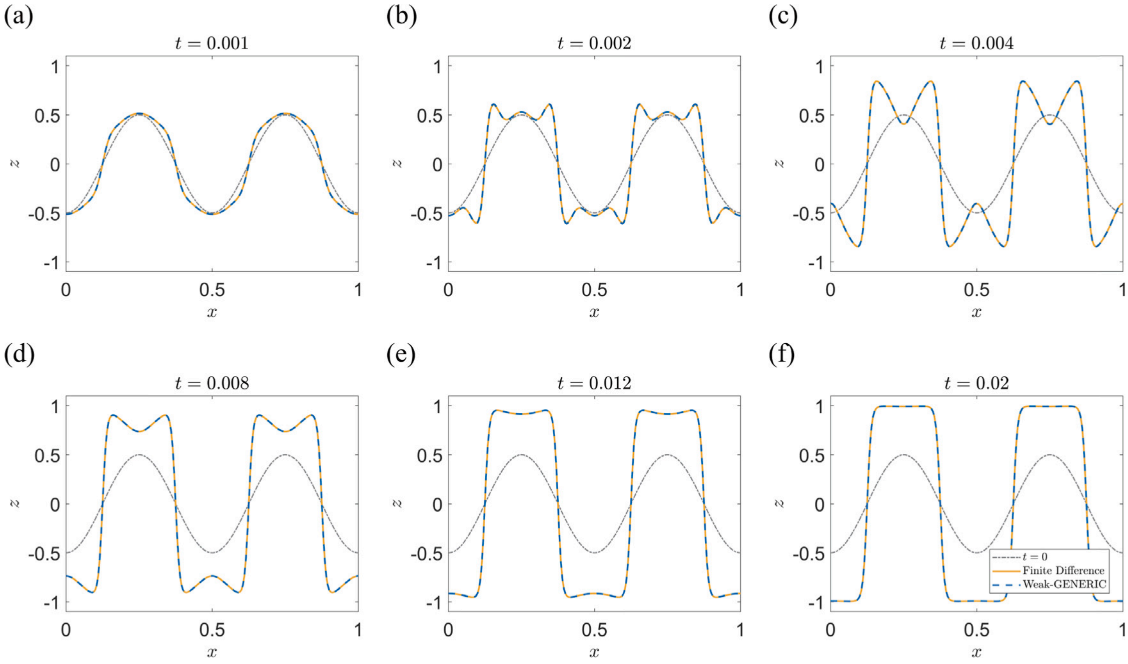

Fig. D.6. Solution of a 4th-order Cahn-Hilliard equation using a finite difference scheme (yellow solid) and the proposed weak-GENERIC method with linear shape functions (blue dashed). (a-f) Different snapshots for the evolution of $z(x,t)$ at $t = 0.001, 0.002, 0.004, 0.008, 0.012$ and 0.02. The black dash dotted lines indicate the sinusoidal initial profile at $t = 0$.

Therefore, the discrete thermodynamic force at x_i is given by

$$Q_i = f_{i,z} - \frac{f_{i+1,\nabla z} - f_{i,\nabla z}}{\Delta x_\gamma} + \frac{f_{i+1,\nabla^2 z} - 2f_{i,\nabla^2 z} + f_{i-1,\nabla^2 z}}{\Delta x_\gamma^2} = f_{i,z} - \frac{\Delta_f f_{i,\nabla z}}{\Delta x_\gamma} + \frac{\Delta_f \Delta_b f_{i,\nabla^2 z}}{\Delta x_\gamma^2}. \tag{C.5}$$

Here, we want to emphasize that assuming that the free energy has a functional dependence on a backward scheme gradient will lead to a forward scheme spatial derivative of $f_{i,\nabla z}$ in the expression of Q. Similarly, a forward scheme for ∇z will lead to a backward scheme for $\nabla f_{,\nabla z}$, and a central scheme for ∇z will lead to a central scheme for $\nabla f_{,\nabla z}$.

Appendix D. Solving higher-order PDEs with linear shape functions

In Section 3.2, we have theoretically shown the possibility of solving PDEs with order higher than two only using linear shape functions (as long as the operator is second order). In this appendix, we will numerically verify such claim.

Consider a one-dimensional diffusion problem for field $z(x,t)$ described by a Laplacian dissipative operator $\mathcal{K} = -\nabla^2$ and the following free energy density,

$$f(z, \nabla z, \nabla^2 z) = \frac{1}{4}\left(z^2 - 1\right)^2 + \frac{\alpha_1}{2}(\nabla z)^2 + \frac{\alpha_2}{2}\left(\nabla^2 z\right)^2, \tag{D.1}$$

which includes a quartic double-well term, a quadratic gradient term, and a quadratic term for the Laplacian of z, where α_1 and α_2 are two coefficients. The evolution equation, Eq. (2), for z can be written as the following 6th-order equation

$$\frac{\partial z}{\partial t} = \nabla^2\left(z^3 - z - \alpha_1 \nabla^2 z + \alpha_2 \nabla^4 z\right). \tag{D.2}$$

Here we solve Eq. (D.2) by two methods. The first is the standard finite difference method, which we will here take as the ground truth solution. The second corresponds to the weak form of the GENERIC formalism using linear shape functions as in Eq. (8), which we denote as weak-GENERIC. In both cases, a simple forward Euler scheme is used for the time derivative and a 2nd-order central scheme is used for the Laplacian. The 4th-order gradient ∇^4 is obtained by applying the Laplacian twice.

Figs. D.6 and D.7 depict the numerical results from two examples using different coefficients α_1 and α_2. In Fig. D.6, we choose $\alpha_1 = 10^{-4}$ and $\alpha_2 = 0$, indicating a 4th-order Cahn-Hilliard equation. The equation is discretized using 200 spatial nodes and solved with a timestep $\Delta t = 2.5 \times 10^{-8}$. In Fig. D.7, we choose $\alpha_1 = 10^{-4}$ and $\alpha_2 = 10^{-5}$, indicating a 6th-order diffusion equation. The equation is discretized using 200 spatial nodes and solved with a timestep $\Delta t = 10^{-9}$. In both examples, the equations are solved from the same sinusoidal initial condition $z = 0.5\sin(2\pi x)$, plotted as a black dash dotted line. In both examples, the solution obtained using the weak-GENERIC method (blue dashed lines) agree very well with that obtained using a finite difference scheme (yellow solid lines). This indicates that the proposed approach utilizing linear shape functions can be used to solve PDEs with order higher than two, as long as the dissipative operator is at most of second order.

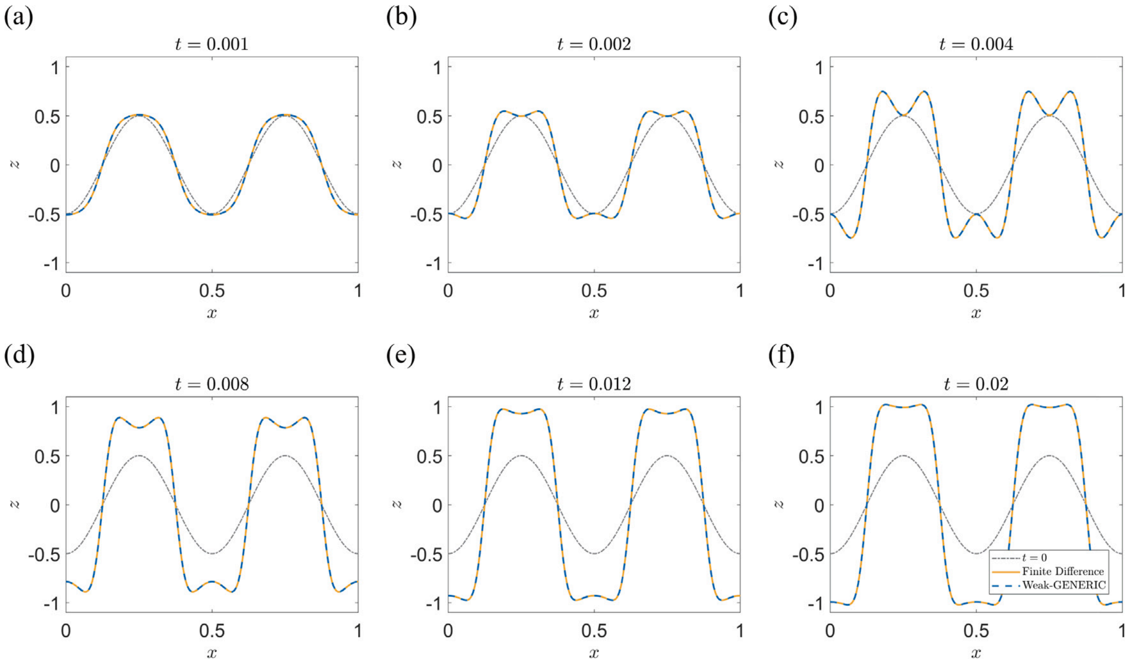

Fig. D.7. Solution of a 6th-order evolution equation using a finite difference scheme (yellow solid) and the proposed weak-GENERIC method with linear shape functions (blue dashed). (a-f) Different snapshots for the evolution of $z(x,t)$ at $t = 0.001, 0.002, 0.004, 0.008, 0.012$ and 0.02. The black dash dotted lines indicate the sinusoidal initial profile at $t = 0$.

Appendix E. Structure details for Stat-PINNs

When learning the off-diagonal entry K_1 from neural network NN_{K_1}, a normalization of the input data is needed. First, the local field values $\mathbf{Z}^1$ at multiple points are transformed into $\tilde{\mathbf{Z}}^1$, which consists of local field values and its derivatives at the middle point as discussed in Appendix B. Then, each component of $\tilde{\mathbf{Z}}^1$ is subtracted by the mean and then divided by the standard deviation of that component within all training data to obtain the normalized input $\tilde{\mathbf{Z}}^{1*}$ for NN_{K_1}. This normalization process can decouple the dependence that naturally exists between sequential points (*e.g.*, between z_i and z_{i+1}), increasing the training robustness, and making it easier to understand and manipulate practically. The off-diagonal entry K_1 can then by parameterized by

$$K_1\left(\mathbf{Z}^1;\boldsymbol{\theta}_1\right) = \left|\mu_{K_1}\right| g\left(\mathrm{NN}_{K_1}\left(\tilde{\mathbf{Z}}^{1*};\boldsymbol{\theta}_1\right)\right), \tag{E.1}$$

where μ_{K_1} is the mean of all measured off-diagonal entry data for ensuring that the output of NN_{K_1} is of scale one, and $g(\cdot)$ is a non-positive function to encode the positive semi-definiteness of the operator $\mathcal{K}_z$. Here, we choose $g(y) = -y - e^{-5}$ if $y \geq 0$ and $g(y) = -e^{y-5}$ if $y < 0$ (Sivaprasad et al., 2021; Huang et al., 2022).

A similar normalization process is applied to the second neural network NN_f. Each component of $\tilde{\mathbf{Z}}^f$, the local profile and its derivatives for expressing the free energy, is subtracted by the mean and then divided by the standard deviation of that component within all training data to obtain the normalized input $\tilde{\mathbf{Z}}^{f*}$. The free energy density can then by represented by

$$f\left(\tilde{\mathbf{Z}}^f;\boldsymbol{\theta}_f\right) = f^{\dagger}\mathrm{NN}_f\left(\tilde{\mathbf{Z}}^{f*};\boldsymbol{\theta}_f\right), \tag{E.2}$$

where $f^{\dagger}$ is an estimated scale for the free energy density based on the evolution equation in order to control that the output of NN_f is of scale one. For the case of $f = f(z)$, we use $f^{\dagger} = \mu_{\mathbf{M}}\mu_{|\dot{z}|}\mu_{|z|}/\mu_{K_1}$, where $\mu_{\mathbf{M}} = \left(\langle\gamma_{i-1},\gamma_i\rangle + \langle\gamma_i,\gamma_i\rangle + \langle\gamma_{i+1},\gamma_i\rangle\right)/3$, $\mu_{|\dot{z}|}$ is the mean of the absolute value of the profile time rate, $|\Delta z/\Delta t|$, within all the training data, and $\mu_{|z|}$ is the mean of the absolute value of all the profile data. For the case of $f = f(z, \nabla z)$, we use $f^{\dagger} = \mu_{\mathbf{M}}\mu_{|\dot{z}|}\left(\mu_{|z|} + \mu_{|\nabla z|}\Delta x_{\gamma}\right)/\mu_{K_1}$, where $\mu_{|\nabla z|}$ is the mean of the absolute value of all ∇z. For the case of $f = f(z, \nabla^2 z)$, we use $f^{\dagger} = \mu_{\mathbf{M}}\mu_{|\dot{z}|}\left(\mu_{|z|} + \mu_{|\nabla^2 z|}\Delta x_{\gamma}^2\right)/\mu_{K_1}$, where $\mu_{|\nabla^2 z|}$ is the mean of the absolute value of all $\nabla^2 z$. Finally, we want to emphasize that the training data for NN_f are $\mathbf{Z}^Q$ and $\Delta\mathbf{Z}^Q$ with superscript Q. This is due to the fact that Q is directly involved in the loss function Eq. (17) instead of f and calculating Q may require information from more local points. Therefore, a transformation from $\mathbf{Z}^Q$ to $\tilde{\mathbf{Z}}^f$ is also required at the beginning of training procedure.

Appendix F. Variance estimation for the weak form of the evolution equation

As mentioned in Section 2.1, the added stochastic noise term for Eq. (8) is $\left\langle\gamma_j, \sqrt{2\epsilon\mathcal{K}_z}\dot{W}_{x,t}\right\rangle$. We note that we are here omitting the subscript ϵ that emphasizes the dependence on the lattice size, for simplicity. Given that $\mathcal{K}_z$ is a linear, symmetric, positive semi-definite operator and $\dot{W}_{x,t}$ is a Gaussian process in space and time, the random variable $\left\langle \gamma_j, \sqrt{2\epsilon\mathcal{K}_z} dW_{x,t} \right\rangle$ during a time interval dt has a (multivariate) Gaussian distribution. The corresponding mean and variance, conditioned on the profile z at time t, are given by

$$\mathbb{E}\left[\left\langle \gamma_j, \sqrt{2\epsilon\mathcal{K}_z} dW_{x,t} \right\rangle \middle| z\right] = \left\langle \gamma_j, \sqrt{2\epsilon\mathcal{K}_z}\mathbb{E}\left[dW_{x,t}\right] \right\rangle = 0, \tag{F.1}$$

$$\begin{aligned} \mathrm{Var}\left[\left\langle \gamma_j, \sqrt{2\epsilon\mathcal{K}_z} dW_{x,t} \middle| z \right\rangle\right] &= \mathbb{E}\left[\left\langle \gamma_j, \sqrt{2\epsilon\mathcal{K}_z} dW_{x,t} \right\rangle^2 \middle| z\right] \\ &= 2\epsilon \int_x \int_y \sqrt{\mathcal{K}_z}\gamma_j(x)\sqrt{\mathcal{K}_z}\gamma_j(y)\mathbb{E}\left[dW_{x,t}dW_{y,t}\right] dx\, dy \\ &= \int_x \int_y 2\epsilon\sqrt{\mathcal{K}_z}\gamma_j(x)\sqrt{\mathcal{K}_z}\gamma_j(y)\delta(x-y)dt\, dx\, dy \\ &= 2\epsilon \left\langle \sqrt{\mathcal{K}_z}\gamma_j, \sqrt{\mathcal{K}_z}\gamma_b \right\rangle dt \\ &= 2\epsilon \left\langle \gamma_j, \mathcal{K}_z\gamma_j \right\rangle dt, \end{aligned} \tag{F.2}$$

where we have used $\mathbb{E}\left[dW_{x,t}dW_{y,t}\right] = \delta(x-y)dt$. Therefore, considering the average evolution of R realizations of particle dynamics during a short time interval Δt, the variance for the residue of Eq. (8) can be given by $\sigma^2_{Eq,j} = 2\epsilon\langle\gamma_j, \mathcal{K}_z\gamma_j\rangle / (R\Delta t)$.

Appendix G. Long-range analytic model for Arrhenius process

The long-range analytic model is introduced in Section 4.1 with Eqs. (18)–(21). Given that Eq. (18) is a diffusion equation, the following numerical scheme is used for solving the continuum evolution

$$\rho_i^{n+1} = \rho_i^n + \frac{m^n_{i+\frac{1}{2}}(Q^n_{i+1} - Q^n_i) - m^n_{i-\frac{1}{2}}(Q^n_i - Q^n_{i-1})}{\Delta x_\gamma^2}\Delta t, \tag{G.1}$$

where $\rho_i^n = \rho(x_i, t^n)$ with $t^n = n\Delta t$ and similarly for Q. We note that the mobility is evaluated at the mid-point to preserve the conservation of ρ and that the density profiles at mid-points are defined as $\rho^n_{i+\frac{1}{2}} = (\rho^n_i + \rho^n_{i+1})/2$.

The discrete operator entries for the long-range Arrhenius analytic expression can be evaluated by the following calculation

$$\begin{aligned} \left\langle \gamma_i, \mathcal{K}_\rho\gamma_i \right\rangle &= \langle \nabla\gamma_i, \cdot m[\rho]\nabla\gamma_i \rangle \\ &= \int_{x_{i-1}}^{x_{i+1}} m[\rho(x)] \left(\nabla\gamma_i(x)\right)^2 dx \\ &= \frac{1}{\Delta x_\gamma^2}\int_{x_{i-1}}^{x_{i+1}} \left[m_i + \nabla m|_i \left(x - x_i\right) + \frac{1}{2}\nabla^2 m\Big|_i \left(x - x_i\right)^2 + O\left(\Delta x_\gamma^3\right)\right] dx \\ &= \frac{2m_i}{\Delta x_\gamma} + \frac{1}{3}\nabla^2 m\Big|_i \Delta x_\gamma + O\left(\Delta x_\gamma^3\right), \end{aligned} \tag{G.2}$$

and

$$\begin{aligned} \left\langle \gamma_{i+1}, \mathcal{K}_\rho\gamma_i \right\rangle &= \left\langle \nabla\gamma_{i+1}, \cdot m[\rho]\nabla\gamma_i \right\rangle \\ &= -\frac{1}{\Delta x_\gamma^2}\int_{x_i}^{x_{i+1}} \left[m_{i+\frac{1}{2}} + \nabla m|_{i+\frac{1}{2}} \left(x - x_{i+\frac{1}{2}}\right) + \frac{1}{2}\nabla^2 m\Big|_{i+\frac{1}{2}} \left(x - x_{i+\frac{1}{2}}\right)^2 + O\left(\Delta x_\gamma^3\right)\right] dx \\ &= -\frac{m_{i+\frac{1}{2}}}{\Delta x_\gamma} - \frac{1}{6}\nabla^2 m\Big|_{i+\frac{1}{2}} \Delta x_\gamma + O\left(\Delta x_\gamma^3\right). \end{aligned} \tag{G.3}$$

We note that all diagonal entries are expanded at the middle point as described in Appendix B. Furthermore, if we denote $V = J * \rho$, the nonlocal mobility functional can be simplified into a local mobility function, *i.e.*, $m[\rho] = \tilde{m}(\rho, V) = D\rho(1-\rho)e^{-V}$. Given that $\tilde{m}(\rho, V)$ and its higher order derivatives are now local, the first two spatial gradients of m can be given by

$$\nabla m = \tilde{m}_{,\rho}\nabla\rho + \tilde{m}_{,V}\nabla V = \tilde{m}_{,\rho}\nabla\rho - \tilde{m}\nabla V, \tag{G.4}$$

$$\begin{aligned} \nabla^2 m &= \tilde{m}_{,\rho\rho}(\nabla\rho)^2 + 2\tilde{m}_{,\rho V}\nabla\rho\nabla V + \tilde{m}_{,\rho}\nabla^2\rho + \tilde{m}_{,VV}(\nabla V)^2 + \tilde{m}_{,V}\nabla^2 V \\ &= \tilde{m}_{,\rho\rho}(\nabla\rho)^2 - 2\tilde{m}_{,\rho}\nabla\rho\nabla V + \tilde{m}_{,\rho}\nabla^2\rho + \tilde{m}\left[(\nabla V)^2 - \nabla^2 V\right], \end{aligned} \tag{G.5}$$

where $\tilde{m}_{,\rho} = D(1-2\rho)e^{-V}$ and $\tilde{m}_{,\rho\rho} = -2De^{-V}$.

In practice, we choose a step function interaction J as described in Section 4.2 and approximate $V = J * \rho$ numerically based on the nonlocality of the system. In all three cases considered in Section 4.2, the nonlocal effect is insignificant, which is verified by the macroscopic evolution. Therefore we approximate the convolution by a zeroth-order expansion as $V = J * \rho \approx 2J_0L\rho$.

Table H.2
Parameters used for the 28 initial cosine profiles, $\rho(x) = \rho_{ave} - A\cos(4\pi f x)$, used in the KMC particle simulation.

Label	f	A	ρ_{ave}
1–7	1	$0.05 + 0.03i$ with $i = 1, \ldots, 7$	$0.5 - (-1)^i \times 0.6\,(0.5 - A)$ with $i = 1, \ldots, 7$
8–14	1	0.31	$0.272 + 0.057i$ with $i = 1, \ldots, 7$
15–21	2	$0.04 + 0.03i$ with $i = 1, \ldots, 7$	$0.5 + (-1)^i \times 0.84\,(0.5 - A)$ with $i = 1, \ldots, 7$
21–28	2	0.29	$0.272 + 0.057i$ with $i = 1, \ldots, 7$

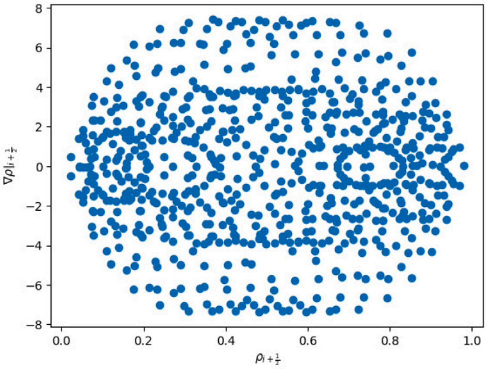

Fig. H.8. Distribution of the training data in the space of $\left(\rho_{i+\frac{1}{2}}, \nabla\rho|_{i+\frac{1}{2}}\right)$ generated from the 28 initial profiles.

Appendix H. Settings for the particle simulation

The particle simulations performed for the Arrhenius process are run using the Kinetic Monte Carlo (KMC) method. Such simulations were used to (1) generate the training data for Stat-PINNs and (2) validate the long-time continuum evolution equations. All particle simulations are coded in C++.

During data generation for Stat-PINNs, the initial profiles should be carefully chosen according to two criteria. First, we would like to avoid an initial profile with a too-wide density range. This is because the jumping probability in the Arrhenius process has an exponential factor to local interaction. Then the corresponding time scale is approximately proportional to $e^{J_0 L\rho}$, which indicates that the time scale at different points in the same profile can be exponentially different which may lead to practical issues. Second, the generated data should be relatively uniform and dense in the input space for training purposes and avoid extrapolation during the prediction stage. Particularly, here we targeted the space of $\left(\rho_{i+\frac{1}{2}}, \nabla\rho|_{i+\frac{1}{2}}\right)$ for learning the off-diagonal entry $\tilde{K}_1\left(\rho_{i+\frac{1}{2}}, \nabla\rho|_{i+\frac{1}{2}}\right) = K(\rho_i, \rho_{i+1})$ of a local operator.

For the three cases shown in Section 4.2, all particle simulations used for data generation are run on a one dimensional system $x \in [0, 0.5]$ with 2000 lattice sites, 25 shape functions and periodic boundary conditions, indicating a lattice size of $\epsilon = 2.5 \times 10^{-4}$ and a spatial discretization of $\Delta x_\gamma = 0.02$. We use 28 different cosine functions $\rho(x) = \rho_{ave} - A\cos(4\pi f x)$ as initial profiles (see Table H.2 for the choice of parameters ρ_{ave}, A and f). The corresponding data distribution can found in Fig. H.8 , which consists of $N_K = 28 \times 25 = 700$ data points. A schematic for the KMC simulation can be found in Fig. H.9. During the simulation, each profile is run for $R = 10^4$ realizations. Note that in each realization, since each lattice site can have at most one particle, the initial profile cannot match the targeted profile. The initial profile for each realization is generated according to the sample probability distribution such that the corresponding ensemble average over all realizations has the targeted cosine profile. Each realization is first run for time $t_{eq} = 50\epsilon^2\tau$ to achieve local equilibration, then following by $N_{\Delta t}$ number of time intervals Δt. Here we choose $N_{\Delta t} = 1$ for all three cases in Section 4.2 and set Δt as $20\epsilon^2\tau$, $40\epsilon^2\tau$ and $20\epsilon^2\tau$ for each case, respectively. The factor τ is used for compensating the timescale difference among different profiles. It is profile-related and is defined as $\tau = e^{2J_0 L\rho_{\max}}$ with $\rho_{\max} = \rho_{ave} + A$ the maximum density in each profile. For each initial profile, the weak-form of the density for all R realizations, $\langle\rho_\epsilon, \gamma_i\rangle$, are outputted at time $t = t_{eq} + nh$ with $n = 0, 1, \ldots, N_h$. Here we choose $N_h = 10$, $h = 0.01\epsilon^2\tau$. Then, profiles $\rho_{\epsilon,i}$ for all spatial indices i are calculated by solving equations $\langle\rho_\epsilon, \gamma_i(x)\rangle = \sum_j \rho_{\epsilon,j}\langle\gamma_j, \gamma_i\rangle$. Given that $N_h h$ is extremely small from a macroscopic perspective such that the average profile barely changes during this interval, we regard these $R_K = N_h R = 10^5$ tiny time intervals h as independent realizations (Embacher et al., 2018). Next, Eq. (9) is applied to calculate the discrete operator entries $\langle\gamma_i, \mathcal{K}_z\gamma_i\rangle$ and $\left\langle\gamma_{i+1}, \mathcal{K}_z\gamma_i\right\rangle$ at all spatial points over R_K realizations. This forms $N_K = 700$ number of training data points for both K_0 and K_1. Furthermore, the weak-form average profiles over R realizations, $\langle\rho, \gamma_i\rangle$, at time $t = t_{eq} + n\Delta t$ with $n = 0, 1, \ldots, N_{\Delta t}$ are outputted, where $i = 0, 1, \ldots, N_\gamma - 1$, and the average profiles ρ_i for all i are obtained. This forms $N_f = N_K N_{\Delta t} = 700$ number of training data for learning the free energy according to the coarse-grained evolution equation during the short time interval Δt.

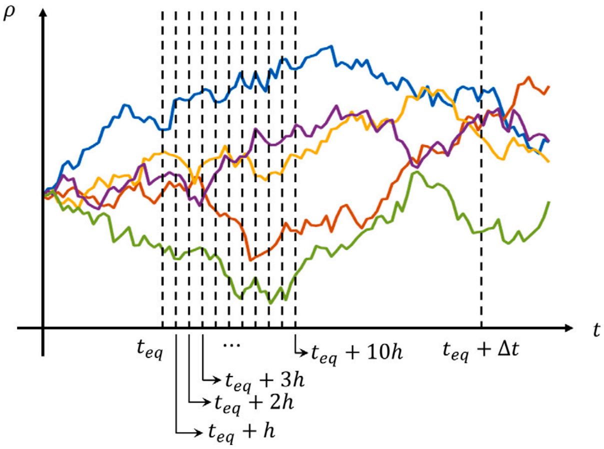

Fig. H.9. Schematic for the data collection using Kinetic Monte-Carlo (KMC) simulations. For all cases in Section 4.2, we run $R = 10^4$ realizations (five are shown in figure) for a given profile. In each realization, we first run a time interval t_{eq} for achieving local equilibrium. The following 10 tiny timesteps h are used for calculating the operator entries, and the density values at $t = t_{eq}$ and $t = t_{eq} + \Delta t$ are used for learning the free energy.

The particle simulations used for validation purposes are necessarily long in macroscopic time scales and thus extremely time-consuming even after calculating each trajectory in parallel. In order to save computational cost, we choose to run the continuum evolution with initial profiles that include multiple periods under a periodic boundary condition but only run the simulation on a single period instead of the whole system. Here we use the triangular wave $\rho(x) = \rho_{ave} - A(|fx - \lfloor fx + 0.5\rfloor| - 1)$ as initial profile, where ρ_{ave}, A and f represent the average density, amplitude and frequency, respectively, and $\lfloor a\rfloor$ is the floor function that outputs the greatest integer less or equal to a. The evolution within one period is duplicated to the full system ranging in $x \in [0, 1]$. For example, for the two cases with $J_0L = 0.9$ in Section 4.2, initial profiles with $f = 2$ are used. Here particle simulations are run on $x \in [0, 0.5]$ with 2000 lattice sites and 25 shape functions, which have the same spatial discretization at both the macroscopic scale $\Delta x_\gamma = 0.02$ and the microscopic scale $\epsilon = 2.5 \times 10^{-4}$ as the predictions from Stat-PINNs and PINNs. For the two cases with $J_0L = 2.2$, initial profiles with $f = 3$ are used. The corresponding particle simulations are run on $x \in [0, 1/3]$ with 1333 lattice sites and 17 shape functions, indicating $\Delta x_\gamma \approx 0.0196$ and $\epsilon \approx 2.5006\times 10^{-4}$, which is a good approximation of the spatial discretization of Stat-PINNs and PINNs. Notice that another rescaling trick is applied here for obtaining particle simulations within a spacial range different from $x \in [0, 1]$ in order to reduce the complexity of the code. See Appendix I for more details.

Appendix I. Rescaling for the particle simulations

When bridging scales, a rescaling of length and time scales is usually required between the microscopic particle scale and the macroscopic continuum scale (Embacher et al., 2018). In this section, we will introduce a rescaling trick for learning and predicting the dissipative evolution for systems with the same physics at the particle level but different spatial descriptions at the macroscale.

First, we consider a target particle process on $x \in [0, \alpha]$ that has N_b bins and N_γ shape functions $\gamma_i(x)$. The Arrhenius process has an interaction strength of J and an interaction range within L_b neighbors on each side, indicating a length of interaction range of $L = \alpha L_b/N_b$. In the target process, the macroscopic times include t_{eq}, h and Δt. The corresponding microscopic times are calculated by multiplying the macroscopic times by a factor N_b^2/α^2 (Embacher et al., 2018). During postprocessing, the operator entries $K_{ij} = \langle \gamma_j, \mathcal{K}_\rho \gamma_i\rangle$ are calculated according to Eq. (9), where Y_{γ_i} is calculated as $Y_{\gamma_i} = \langle \rho - \bar{\rho}, \gamma_i\rangle/\sqrt{\Delta x_\gamma}$ with $\Delta x_\gamma = \alpha/N_\gamma$ and $\bar{\rho}$ as the average of ρ over all realizations. Furthermore, using Eq. (8), we can obtain the free energy $f(\rho)$ or $f(\rho, \nabla\rho)$.

Next, we consider a different system, which we will call the simulated system. This system also has N_b bins and N_γ shape functions but ranges between $x \in [0, 1]$. We denote such shape functions as $\gamma_i^s(x)$ with a superscript s to indicate that they refer to the simulated system. Although $\gamma_i^s(x)$ is squeezed by a factor α with respect to $\gamma_i(x)$, it covers the same amount of bins as $\gamma_i(x)$ in the target system. Furthermore, the interaction strength J and interaction range L_b (in unit of bins) are the same as the ones in the target system, indicating the same Arrhenius process at the particle level. We note that the microscopic times for this simulated system should be calculated by multiplying the macroscopic times by a factor of N_b^2. Therefore, in order to keep the particle dynamics the same as that of the target system, all macroscopic times should be rescaled by multiplying by a factor $1/\alpha^2$ from the target system, *i.e.*, $h^s = h/\alpha^2$ and $\Delta t^s = \Delta t/\alpha^2$. Given the same microscopic local dynamics, ρ^s remains the same as ρ since both are defined within the range $[0, 1]$. Then, based on the rescaling of length and time scales, the following rescalings can be found, $\dot{\rho} = \dot{\rho}^s/\alpha^2$, $\nabla\rho = (\nabla\rho)^s/\alpha$, $M = \alpha M^s$ and $Y_{\gamma_i} = \sqrt{\alpha} Y^s_{\gamma_i^s}$. Therefore, the dissipative operator entries and free energy density for target system can be rescaled from the simulated system by $K_{ij} = K_{ij}^s/\alpha$, $f(\rho) = f^s(\rho)$ and $f(\rho, \nabla\rho) = f^s(\rho, \alpha\nabla\rho)$.

Appendix J. Settings for Stat-PINNs and PINNs training

Stat-PINNs involve two sequential training processes for NN_{K_1} and NN_f. All neural networks are fully-connected feed-forward networks with SoftPlus activation functions and are implemented in Python using JAX as the main library. Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-2} is used for all cases in Stat-PINNs. For all cases in Section 4.2, NN_{K_1} has 2 hidden layers with 20 neurons per layer and is trained for 6000 epochs. NN_f has 2 hidden layers with 20 neurons per layer and is trained for 2000 epochs.

All cases in Section 4.2 are also trained by PINNs as comparisons, which also consist of NN_{K_1} and NN_f but are trained jointly using the following loss function

$$\mathcal{L}_{\text{PINNs}} = \frac{1}{2N_f}\sum_{s=1}^{N_f}\left\|\sum_i \langle\gamma_{j^s},\gamma_i\rangle \Delta z_i^{(s)}/\Delta t + \sum_i \langle \gamma_{j^s}, \mathcal{K}_{z^{(s)}}\gamma_i\rangle Q(\mathbf{Z}_i^{Q(s)};\boldsymbol{\theta}_f)\right\|^2. \tag{J.1}$$

This is almost the same loss function as the one used for NN_f in Stat-PINNs (Eq. (17)) but setting the variance on denominator to one. Notice that now the operator entries $\langle\gamma_{j^s},\mathcal{K}_{z^{(s)}}\gamma_i\rangle$ are represented by NN_{K_1} to be trained. Structures of NN_{K_1} and NN_f in PINNs are identical to those in Stat-PINNs, but parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_f$ are trained simultaneously. All cases are trained for 10000 epochs using Adams optimizer with a learning rate of 10^{-3}. We remark that for both Stat-PINNs and PINNs, the loss functions have reached stable plateaus after the epochs previously mentioned, indicating the convergence of the training process, though the results from PINNs may not be robust enough.

Appendix K. Non-uniqueness problem for the free energy

As discussed in Section 3.2, different functional dependencies of the free energy density f lead to different forms of the driving force and hence the evolution equations. Due to the mathematical structure of the evolution equations, f cannot be uniquely determined from the macroscopic evolution equation, even when the operator is uniquely identified. In this section, we will discuss the non-uniqueness of the free energy based for the three different cases, assuming that the kernel of $\mathcal{K}_z$ is one-dimensional and consists of constants, which covers the cases we are interested in. For simplicity, we will use ρ as the field variable in this section to keep it the same as in the Arrhenius example.

First, we assume that the free energy density can be written as $f = f(\rho)$. This is the simplest case and also the one used in Section 4.2. In this case, the thermodynamic force is given by Eq. (13) as $Q = f'(\rho)$ and the governing equation Eq. (12) is a diffusion equation. By assumption, the difference of two driving forces leading to the same evolution must be constant, and hence the difference of two free energies leading to the same evolution must be an affine function of ρ. We therefore define the calibrated free energy density as

$$\tilde{f}(\rho) = f(\rho) - f(\rho_{r0}) - f'(\rho_{r1})\left(\rho - \rho_{r0}\right). \tag{K.1}$$

This can be understood as forcing $\tilde{f}(\rho_{r0}) = 0$ and $\tilde{f}'(\rho_{r1}) = 0$ at two reference points ρ_{r0} and ρ_{r1}. For simplicity, we choose $\rho_{r0} = \rho_{r1} = 0.5$ as reference points. We acknowledge that defining $f(0) = 0$ would be more natural from a physical standpoint as there is no energy when no particle exists. $\rho_{r0} = 0.5$ is chosen for practical reasons, to facilitate plotting and comparing free energy profiles from different models in a easier and more intuitive way.

Second, when assuming $f = f(\rho, \nabla\rho)$, the thermodynamic force is given by Eq. (14) as $Q = f_{,\rho} - \nabla\cdot f_{,\nabla\rho}$ and the governing equation is a fourth order equation, such as the Cahn–Hilliard equation. Then the difference $\hat{f}(\rho,\nabla\rho) := \tilde{f}(\rho,\nabla\rho) - f(\rho,\nabla\rho)$ of two free energies leading to the same evolution must satisfy the condition

$$\hat{f}_{,\rho}(\rho,\nabla\rho) - \left(\hat{f}_{,\nabla\rho}\right)_{,\rho}\nabla\rho - \left(\hat{f}_{,\nabla\rho}\right)_{,\nabla\rho}\nabla^2\rho = \text{const.}$$

As only the third term depends on the second derivative of the density, this term must vanish, as otherwise we can chose ρ such that its variations lead to a nonconstant term. Thus $\hat{f}(\rho,\nabla\rho)$ has to be affine in $\nabla\rho$, i,e, $\hat{f}(\rho,\nabla\rho) = g(\rho) + \hat{f}_{,\nabla\rho}(\rho,0)\nabla\rho$ for some $g(\rho)$. Using this structure in the equation above we see that $g'(\rho) = \text{const}$, and therefore $g(\rho)$ is affine in ρ. We use this structure to define the calibrated free energy density as

$$\tilde{f}(\rho,\nabla\rho) = f(\rho,\nabla\rho) - f(\rho_{r0},0) - f_{,\rho}(\rho_{r1},0)\left(\rho-\rho_{r0}\right) - f_{,\nabla\rho}(\rho,0)\nabla\rho, \tag{K.2}$$

which satisfies $\tilde{f}(\rho_{r0},0) = 0$, $\tilde{f}_{,\rho}(\rho_{r1},0) = 0$ and $\tilde{f}_{,\nabla\rho}(\rho,0) = 0$. Similar to the case above, we choose $\rho_{r0} = \rho_{r1} = 0.5$ as reference points. By construction, the calibrated free energy density is unique.

The final case to discuss is $f = f(\rho,\nabla^2\rho)$, which corresponds to the long-range analytic model for Arrhenius process when the convolution is approximated to second order as $J * \rho \approx 2J_0L\left(\rho + L^2\nabla^2\rho/6\right)$. In this case, the thermodynamic force is given by $Q = f_{,\rho} + \nabla^2 f_{,\nabla^2\rho}$ according to Eq. (15). We define the calibrated free energy density as

$$\tilde{f}(\rho,\nabla^2\rho) = f(\rho,\nabla^2\rho) - f(\rho_{r0},0) - f_{,\rho}(\rho_{r1},0)\left(\rho-\rho_{r0}\right) - f_{,\nabla^2\rho}(0,0)\nabla^2\rho, \tag{K.3}$$

which satisfies $f(\rho_{r0},0) = 0$, $f_{,\rho}(\rho_{r1},0) = 0$ and $f_{,\nabla^2\rho}(0,0) = 0$. Here we choose $\rho_{r0} = \rho_{r1} = 0.5$ as reference points.

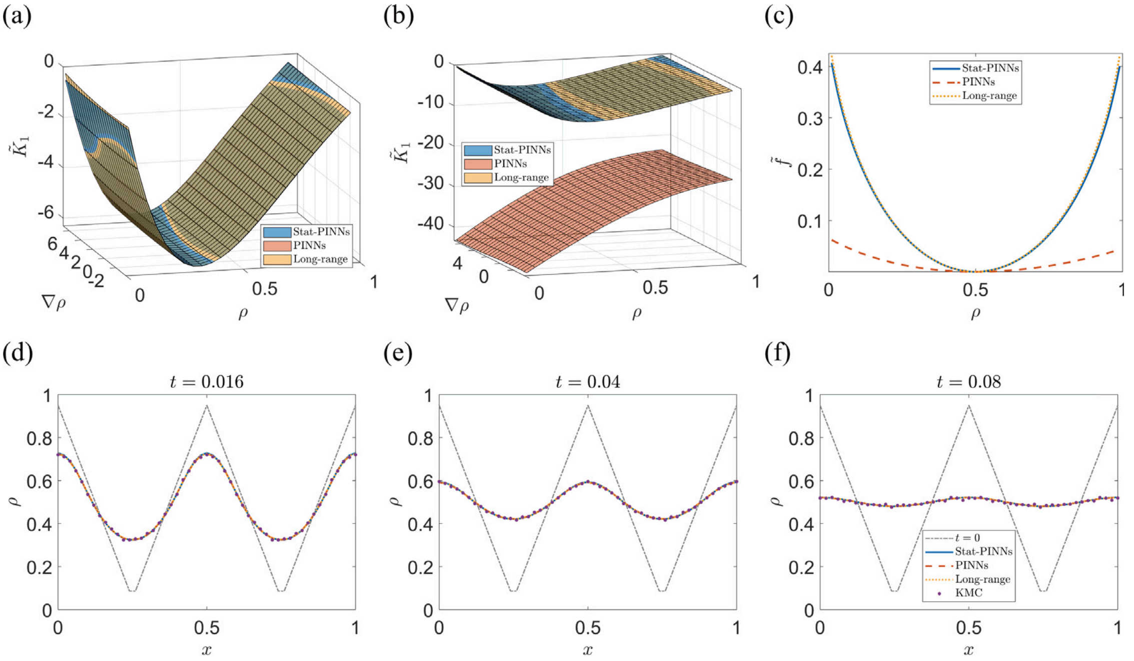

Fig. L.10. Results for Arrhenius process with $JL = 0.9, L = 40\epsilon$ using additional training data, including (a, b) dissipative operator entry $\tilde{K}_1$ plotted in two different ranges, (c) calibrated free energy density $\tilde{f}$, and (d-f) snapshots of macroscopic evolution starting from a triangular wave initial profile (black dash-dotted line). Predicting methods include Stat-PINNs (blue surfaces or blue solid lines), PINNs (orange surfaces or orange dashed lines), and a long-range analytic model (yellow surfaces or yellow dotted lines). Results from KMC particle simulations (purple dots) are used as true macroscopic evolution for comparison.

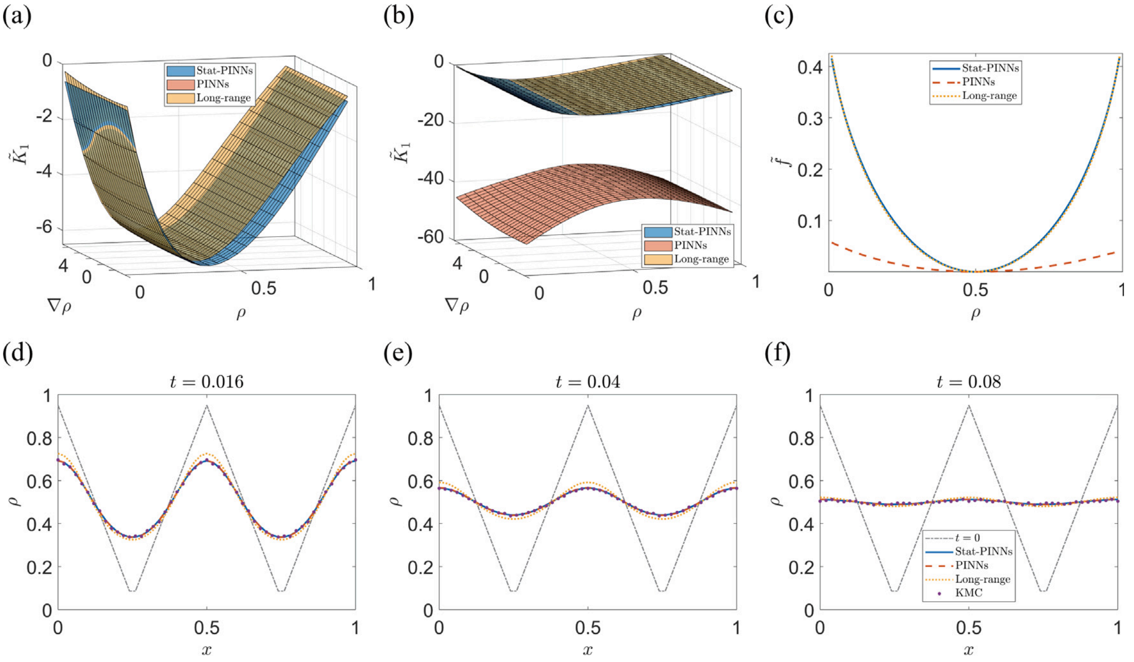

Fig. L.11. Results for Arrhenius process with $JL = 0.9, L = 2\epsilon$ using additional training data, including (a, b) dissipative operator entry $\tilde{K}_1$ plotted in two different ranges, (c) calibrated free energy density $\tilde{f}$, and (d-f) snapshots of macroscopic evolution starting from a triangular wave initial profile (black dash-dotted line). Predicting methods include Stat-PINNs (blue surfaces or blue solid lines), PINNs (orange surfaces or orange dashed lines), and a long-range analytic model (yellow surfaces or yellow dotted lines). Results from KMC particle simulations (purple dots) are used as true macroscopic evolution for comparison.

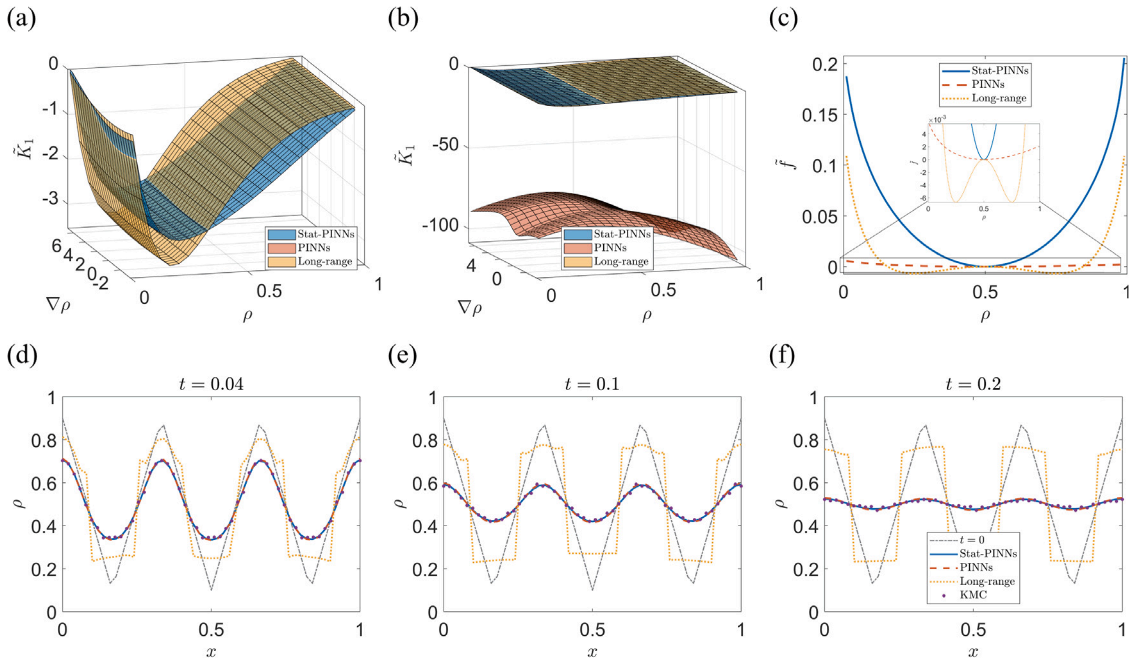

Fig. L.12. Results for Arrhenius process with $JL = 2.2, L = 2\epsilon$ using sufficient training data, including (a, b) dissipative operator entry $\tilde{K}_1$ plotted in two different ranges, (c) calibrated free energy density $\tilde{f}$, and (d-f) snapshots of macroscopic evolution starting from a triangular wave initial profile (black dash-dotted line). Predicting methods include Stat-PINNs (blue surfaces or blue solid lines), PINNs (orange surfaces or orange dashed lines), and a long-range analytic model (yellow surfaces or yellow dotted lines). Results from KMC particle simulation (purple dots) are used as true macroscopic evolution for comparison.

Appendix L. Results with sufficient training data

While Stat-PINNs successfully predict the continuum evolution for the first three cases in Section 4.2, results from PINNs are not as good with the given limited data. In this section, we use almost the same parameter setting as in Appendix H, but increase the training data for NN_f by 10 times, *i.e.*, $N_{\Delta t} = 10$ and $N_f = 7000$. Results of the off-diagonal operator entry K_1, the calibrated free energy density $\tilde{f}$ and the continuum evolution for all three cases in Section 4.2 are shown in Figs. L.10–L.12. As we can see, although PINNs cannot predict the true operator or free energy, the predicted continuum evolution from PINNs now matches the true results from particle simulations. This indicates that PINNs can predict the continuum evolution as long as there exists sufficient training data such that the noise from stochastic particle dynamics is negligible, as could be expected. However, as noted continuum data alone, in the absence of statistical physics, is insufficient to uniquely characterize the free energy and operator.

During the preparation of this work the authors used ChatGPT in order to check the grammatical construction of some sentences. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Code availability

The data and code used in this work is available at https://github.com/celiareina/Stat-PINNs.

References

Adams, Stefan, Dirr, Nicolas, Peletier, Mark, Zimmer, Johannes, 2013. Large deviations and gradient flows. Phil. Trans. R. Soc. A 371 (2005), 20120341.

Arampatzis, Giorgos, Katsoulakis, Markos A., Plecháč, Petr, Taufer, Michela, Xu, Lifan, 2011. Hierarchical fractional-step approximations and parallel kinetic Monte Carlo algorithms. J. Comput. Phys. 231 (23), 7795–7814.

Bodineau, Thierry, Gallagher, Isabelle, Saint-Raymond, Laure, 2016. The Brownian motion as the limit of a deterministic system of hard-spheres. Invent. Math. 203, 493–553.

Bortz, Malvin H., Lebowitz, Joel L., 1975. A new algorithm for Monte Carlo simulation of ising spin systems. J. Comput. Phys. 17 (1), 10–18.

Brunton, Steven L., Kutz, J. Nathan, 2022. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge University Press.

Cornalba, Federico, Shardlow, Tony, Zimmer, Johannes, 2021. Well-posedness for a regularised inertial dean-kawasaki model for slender particles in several space dimensions. J. Differential Equations 284, 253–283.

Dewey, William C., 1994. Arrhenius relationships from the molecule and cell to the clinic. Int. J. Hyperth. 10 (4), 457–483.

Dietrich, Felix, Makeev, Alexei, Kevrekidis, George, Evangelou, Nikolaos, Bertalan, Tom, Reich, Sebastian, Kevrekidis, Ioannis G., 2023. Learning effective stochastic differential equations from microscopic simulations: Linking stochastic numerics to deep learning. Chaos 33 (2).

Dirr, Nicolas, Fehrman, Benjamin, Gess, Benjamin, 2020. Conservative stochastic PDE and fluctuations of the symmetric simple exclusion process.

Embacher, Peter, Dirr, Nicolas, Zimmer, Johannes, Reina, Celia, 2018. Computing diffusivities from particle models out of equilibrium. Proc. R. Soc. 474 (2212), 20170694.

Fehrman, Benjamin, Gess, Benjamin, 2024. Well-posedness of the Dean-Kawasaki and the nonlinear Dawson-Watanabe equation with correlated noise. Arch. Ration. Mech. Anal. 248 (2), 20,60.

Gibbs, Julian H., 1972. Sufficient conditions for the arrhenius rate law. J. Chem. Phys. 57 (10), 4473–4478.

Gilmer, G.H., Bennema, P., 1972. Simulation of crystal growth with surface diffusion. J. Appl. Phys. 43 (4), 1347–1360.

Givon, Dror, Kupferman, Raz, Stuart, Andrew, 2004. Extracting macroscopic dynamics: model problems and algorithms. Nonlinearity 17 (6), R55.

Grmela, Miroslav, Öttinger, Hans Christian, 1997. Dynamics and thermodynamics of complex fluids. I. Development of a general formalism. Phys. Rev. E 56 (6), 6620.

Hernández, Quercus, Badías, Alberto, Chinesta, Francisco, Cueto, Elías, 2022. Thermodynamics-informed graph neural networks. IEEE Trans. Artif. Intell. 5 (3), 967–976.

Hernandez, Quercus, Badias, Alberto, Gonzalez, David, Chinesta, Francisco, Cueto, Elias, 2021. Deep learning of thermodynamics-aware reduced-order models from data. Comput. Methods Appl. Mech. Engrg. 379, 113763.

Huang, Shenglin, Aguado-Montero, Santiago, Dirr, Nicolas, Zimmer, Johannes, Reina, Celia, 2020. From particle fluctuations to macroscopic evolution equations: The case of exclusion dynamics.. CEUR Workshop Proc. 2783, 140–152.

Huang, Shenglin, He, Zequn, Reina, Celia, 2022. Variational onsager neural networks (VONNs): A thermodynamics-based variational learning strategy for non-equilibrium PDEs. J. Mech. Phys. Solids 163, 104856.

Katsoulakis, Markos A., Vlachos, Dionisios G., 2003. Coarse-grained stochastic processes and kinetic Monte Carlo simulators for the diffusion of interacting particles. J. Chem. Phys. 119 (18), 9412–9427.

Kingma, Diederik P., Ba, Jimmy, 2014. Adam: A method for stochastic optimization.

Kipnis, Claude, Landim, Claudio, 1998. Scaling Limits of Interacting Particle Systems. Vol. 320, Springer Science & Business Media.

Konarovskyi, Vitalii, Lehmann, Tobias, von Renesse, Max-K., 2019. Dean-kawasaki dynamics: ill-posedness vs. triviality. Electron. Commun. Probab. 24, 8–9.

Kraaij, Richard C., Lazarescu, Alexandre, Maes, Christian, Peletier, Mark, 2020. Fluctuation symmetry leads to GENERIC equations with non-quadratic dissipation. Stochastic Process. Appl. 130 (1), 139–170.

Kubo, Rep, 1966. The fluctuation-dissipation theorem. Rep. Progr. Phys. 29 (1), 255.

Laidler, Keith J., 1984. The development of the arrhenius equation. J. Chem. Educ. 61 (6), 494.

Leadbetter, Travis, Purohit, Prashant K., Reina, Celia, 2023. A statistical mechanics framework for constructing nonequilibrium thermodynamic models. PNAS Nexus 2 (12), pgad417.

Lee, Kookjin, Trask, Nathaniel, Stinis, Panos, 2021. Machine learning structure preserving brackets for forecasting irreversible processes. Adv. Neural Inf. Process. Syst. 34, 5696–5707.

Lelièvre, Tony, Rousset, Mathias, Stoltz, Gabriel, 2010. Free energy computations. Imperial College Press, London, p. xiv+458, 2681239(2012c:82068), A mathematical perspective.

Li, Xiaoguai, Dirr, Nicolas, Embacher, Peter, Zimmer, Johannes, Reina, Celia, 2019. Harnessing fluctuations to discover dissipative evolution equations. J. Mech. Phys. Solids 131, 240–251.

Linderoth, Trolle René, Horch, Sebastian, Læ gsgaard, Erik, Stensgaard, Ivan, Besenbacher, Flemming, 1997. Surface diffusion of pt on pt (110): Arrhenius behavior of long jumps. Phys. Rev. Lett. 78 (26), 4978.

Mielke, Alexander, 2011. Formulation of thermoelastic dissipative material behavior using GENERIC. Contin. Mech. Thermodyn. 23 (3), 233–256.

Montefusco, Alberto, Peletier, Mark A, Öttinger, Hans Christian, 2021. A framework of nonequilibrium statistical mechanics. II. coarse-graining. J. Non-Equilib. Thermodyn. 46 (1), 15–33.

Oksendal, Bernt, 2013. Stochastic Differential Equations: An Introduction with Applications. Springer Science & Business Media.

Öttinger, Hans Christian, 2005. Beyond Equilibrium Thermodynamics. John Wiley & Sons.

Öttinger, Hans Christian, Grmela, Miroslav, 1997. Dynamics and thermodynamics of complex fluids. II. Illustrations of a general formalism. Phys. Rev. E 56 (6), 6633.

Öttinger, Hans Christian, Peletier, Mark A., Montefusco, Alberto, 2021. A framework of nonequilibrium statistical mechanics. I. Role and types of fluctuations. J. Non-Equilib. Thermodyn. 46 (1), 1–13.

Pavliotis, Grigorios A., 2014. Stochastic processes and applications. Texts in Applied Mathematics, Vol. 60, Springer, p. xiv+339.

Presutti, Errico, 2009. Scaling limits in statistical mechanics and microstructures in continuum mechanics. Theoretical and Mathematical Physics, Springer, Berlin, p. xvi+467.

Schulze, Tim P., 2008. Efficient kinetic Monte Carlo simulation. J. Comput. Phys. 227 (4), 2455–2462.

Seifert, Udo, 2012. Stochastic thermodynamics, fluctuation theorems and molecular machines. Reports on progress in physics 75 (12), 126001.

Sevick, Edith M., Prabhakar, Ranganathan, Williams, Stephen R., Searles, Debra J., 2008. Fluctuation theorems. Annu. Rev. Phys. Chem. 59, 603–633.

Sivaprasad, Sarath, Singh, Ankur, Manwani, Naresh, Gandhi, Vineet, 2021. The curious case of convex neural networks. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21. Springer, pp. 738–754.

Tadmor, Ellad B., Miller, Ronald E., 2011. Modeling Materials: Continuum, Atomistic and Multiscale Techniques. Cambridge University Press.

Touchette, Hugo, 2009. The large deviation approach to statistical mechanics. Phys. Rep. 478 (1–3), 1–69.

Vlachos, D.G., Katsoulakis, M.A., 2000. Derivation and validation of mesoscopic theories for diffusion of interacting molecules. Phys. Rev. Lett. 85 (18), 3898.

Wang, Sifan, Yu, Xinling, Perdikaris, Paris, 2022. When and why PINNs fail to train: A neural tangent kernel perspective. J. Comput. Phys. 449, 110768.

Zhang, Zhen, Shin, Yeonjong, Em Karniadakis, George, 2022. GFINNs: GENERIC formalism informed neural networks for deterministic and stochastic dynamical systems. Phil. Trans. R. Soc. A 380 (2229), 20210207.

Zhdanov, Vladimir Petrovich, 1991. Arrhenius parameters for rate processes on solid surfaces. Surf. Sci. Rep. 12 (5), 185–242.

Zwanzig, Robert, 2001. Nonequilibrium Statistical Mechanics. Oxford University Press, New York, p. viii+222, 2012558.