Solving dissipative partial differential equations via Onsager variational formulations: Evolution Onsager Neural Networks (EONNs)

Anonymous Authors[1]

Abstract

Deep learning approaches to numerical solutions of dissipative partial differential equations (PDEs) stand at the forefront of research in artificial intelligence for scientific applications, encompassing domains such as physics, biology, and fluid mechanics. Existing approaches might lose the inheritance of the crucial physical properties from original systems. In this study, we present a novel time-dependent neural network framework, namely Evolution Onsager Neural Networks (EONNs), explicitly crafted for the direct simulation of dynamic processes in dissipative systems. EONN stands out by conceptualizing neural networks as time-dependent approximators seamlessly integrated into thermodynamic principles, thereby upholding the energy decreasing properties of the original dynamics in a thermodynamically consistent manner. Subsequently, we validate the effectiveness of EONN across diverse scientific domains, including biology population dynamics, physics, and fluid mechanics. Comparative experiments against the most representative Physics-Informed Neural Networks (PINNs) highlight EONN's comparable performance in linear PDEs and its superior capabilities in highly nonlinear scenarios. Notably, EONN's adaptability in neural network structure and its adeptness in handling irregular domains underscore its versatility. Furthermore, we introduce a hyperparameter tuning strategy rooted in Onsager-Maclup principles, showcasing its efficacy in optimizing model performance.

1. Introduction

Numerous challenges in science and engineering necessitate the iterative resolution of intricate systems of partial differential equations (PDEs) across varying parameter values. This scenario is particularly prevalent in scientific domains such as biology population dynamics, fluid mechanics, and the simulation on irregular domains. Frequently, these systems demand meticulous discretization to accurately capture the nuances inherent in the modeled phenomena. Moreover, those phenomena could be modeled by Onsager's principle.

Onsager's principle(Onsager, 1931a;b) is a notable variational principle introduced by Onsager in his influential paper on the reciprocal relation. This principle, rooted in the concept of the reciprocal relation, plays a fundamental role in numerous mechanical and thermodynamic theories, including the realm of physical and biological scenarios. By employing Onsager's principle, a systematic procedure emerges (Doi, 2022; 2015), facilitating the generation of governing equations in a structured manner. The application of this principle provides valuable insights into the behavior of complex systems, contributing to the comprehensive understanding of mechanical and thermodynamic phenomena.

Conventional methods for solving complex systems excel in providing robust error convergence analysis and enhancing the computational efficiency of nonlinear PDEs. However, they face challenges when dealing with irregular domains and high-dimensional cases, primarily due to the curse of dimensionality. In contrast, neural network approaches excel in tackling high-dimensional cases and exhibit flexibility in handling irregular domains. Nevertheless, many of these neural network methods lack a solid theoretical basis to guarantee convergence, posing difficulties in ensuring the consistency of approximated solutions with true solutions.

To overcome the shortcomings of both conventional methods and neural network methods, we propose a new framework—Evolution Onsager Neural Networks (EONNs) in this paper. We use neural networks with time-dependent parameters to represent the solutions of PDEs and exploit the Onsager variational principle (Onsager, 1931a;b) to derive the dynamics of parameters representing the evolution of PDEs. This is the first time a physical variational prin-

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ciple has been introduced to derive the dynamic evolution of neural networks. Therefore, it is natural to benefit from conserving the variational structure of the underlying law, maintaining the law of declining energy. Furthermore, the preserving-variation property makes the temporal variable t and spatial $\boldsymbol{x}$ separable through integration, reducing the original variational problems in infinite space to convex minimization problems with respect to the parameters of neural networks. The convexity makes it possible to obtain the optimal solution directly—an Ordinary Differential Equation (ODE) system of the parameters of neural networks. Finally, even if a hyper-parameter is introduced, a hyper-parameter selection strategy based on the Onsager-Machlup principle (Onsager & Machlup, 1953) is designed to guarantee the accuracy of derived dynamics.

In conclusion, EONNs present a pioneering framework for addressing challenges in scientific computing by integrating neural networks with time-dependent parameters and leveraging the Onsager variational principle. Notably, EONNs exhibit versatility in handling irregular domains, lower derivative orders, and deterministic solutions. Comparative experiments highlight their comparable performance in linear PDEs and superior capabilities in highly nonlinear scenarios. The proposed hyperparameter selection strategy further enhances their efficacy. In summary, EONNs offer a promising and theoretically grounded approach to diverse scientific problems.

The principal contributions of this paper can be succinctly outlined as follows:

1. We propose a novel method, Evolution Onsager Neural Networks (EONN). This framework effectively reduces complex PDEs to ODEs and exhibits adaptability to various neural network structures. The unique time-dependent weight parametrization structure enhances its versatility. Notably, the framework's simplicity is matched by its broad applicability across multiple scientific domains. Its distinctive feature lies in the theoretical guarantee of preserving fundamental physical properties inherent in the original system. EONN is particularly adept at handling irregular domains and has lower derivative orders, minimizing singularity requirements.

2. We conduct comprehensive experiments comparing EONN with PINNs(Raissi et al., 2019), a representative method in the field. The results highlight EONN's comparable performance in solving linear PDEs and its superior capabilities in addressing highly nonlinear scenarios. A noteworthy aspect is the deterministic nature of EONN's solutions, providing a more reliable modeling process compared to methods with stochastic properties.

3. We introduce a Hyper-Parameter selection strategy based on the Onsager-Maclup principle. This strategy serves as an effective method for selecting optimal regularization hyperparameters, showcasing its prowess in optimizing model performance.

4. Our framework is put to the test through numerical experiments spanning various scientific scenarios. Applications include fluid mechanics (Diffusion equations and Allen Cahn equations), computational biology (Fish-KPP equations), and others. These experiments illustrate the robustness of EONN across diverse problem landscapes. Importantly, the framework evolves to the stable state of the original system without requiring pre-defined terminal times, providing flexibility in capturing system behavior.

2. Preliminaries

2.1. Onsager Variational Principle

In this study, we seek to employ Onsager's principle to elucidate the dynamics of dissipative processes in biological systems and fluid flow fields. Leveraging the framework of non-equilibrium thermodynamics, Onsager's principle offers a powerful tool for modeling and analyzing. In fact, it can be regarded as an equivalent statement of Second Law of Thermodynamics, namely the entropy production rule, for an isothermal open system. Onsager proposed a variational principle to characterize the dynamics of non-equilibrium systems (Onsager, 1931a;b), which is formulated as follows. A non-equilibrium system is characterized by a set of state variables $\boldsymbol{u} = (u_1, u_2, \ldots, u_f)$. Our objective is to calculate the time dependence of $\boldsymbol{u}(t)$ for a given initial state $\boldsymbol{u}(t=0)$. The time evolution of $\boldsymbol{u}$ is obtained by solving Onsager's kinetic equations:

$$\dot{u}_i = -\sum_j \mu_{ij} \frac{\partial \mathcal{E}}{\partial u_j}, \tag{1}$$

where $\mathcal{E}(\boldsymbol{u})$ is the free energy of the system, $\dot{u}_i = \frac{du_i}{dt}$ and $\mu_{ij}(\boldsymbol{u})$ are kinetic coefficients, both are functions of $\boldsymbol{u}$. $\mathcal{E}(\boldsymbol{u})$ is an integral with respect to state variable $\boldsymbol{u}$ on the whole domain. Onsager showed that if the kinetic equation is written in the form of the equation above, $\mu_{ij}(\boldsymbol{u})$ must be positive definite and symmetric. Let ζ_{ij} be the inverse of matrix μ_{ij}. Then, the time evolution is given by

$$-\sum \zeta_{ij}\dot{u}_j - \frac{\partial \mathcal{E}}{\partial u_i} = 0. \tag{2}$$

This equation can be cast in the form of a minimum principle. Consider the following quadratic function of the rate of change of the state $\dot{\boldsymbol{u}} = (\dot{u}_1, \dot{u}_2, \ldots, \dot{u}_f)$: The time evolution is given by minimizing Rayleighian functional $\mathcal{R}(u, \dot{u})$ with respect to $\dot{u}_i$, which is defined as:

$$\mathcal{R}(\dot{\boldsymbol{u}};\boldsymbol{u})=\sum\frac{\partial\mathcal{E}}{\partial u_i}\dot{u}_i+\frac{1}{2}\sum\zeta_{ij}(\boldsymbol{u})\dot{u}_i\dot{u}_j. \tag{3}$$

The Rayleighian consists of two parts, namely the energy variation effect and the dissipation effect. The former is solely used to determine the equilibrium state with minimum energy, while the latter specifies how a system evolves towards this equilibrium state and thus gives a dynamic path. Onsager principle plays a similar role as the least action principle does in classical mechanics.

2.2. Onsager-Maclup principle

Consider the slightly modified definition for the Rayleighian (3):

$$\tilde{\mathcal{R}}(\dot{\boldsymbol{u}};\boldsymbol{u})=\mathcal{R}(\dot{\boldsymbol{u}};\boldsymbol{u})-\mathcal{R}_{\min}(\boldsymbol{u}), \tag{4}$$

where $\mathcal{R}_{\min}(\boldsymbol{u})$ is the minimum value of $\mathcal{R}(\dot{\boldsymbol{u}};\boldsymbol{u})$ in the space of $\dot{\boldsymbol{u}}$.

Both $\tilde{\mathcal{R}}$ and $\mathcal{R}$ delineate the dynamics of a physical system at a specific moment t. However, to characterize the system's dynamics across the entire kinetic path, it becomes necessary to study the accumulated effect over a time period. This leads to the introduction of the Onsager-Maclup functional:

$$\mathcal{O}[\boldsymbol{u}(t)]=\int_0^t ds\tilde{\mathcal{R}}\left[\dot{\boldsymbol{u}}(s);\boldsymbol{u}(s)\right]. \tag{5}$$

It represents the accumulated excess of the Rayleighian over its minimum for an irreversible system along a kinetic path $\boldsymbol{u}(s)$, s in the interval $[0,t]$. Widely employed in the analysis of systems far from equilibrium, the Onsager-Machlup functional offers a mathematical framework for assessing fluctuations and probability distributions of stochastic processes. This functional enables the quantification of fluctuations and their correlation with the underlying dynamics of the system. The minimization of the Onsager-Machlup functional leads to the most probable paths that the system follows over time, and thus offers a statistical ground for selecting dynamic paths.

3. Proposed Method

In this section, we explain how we derived the EONNs using Onsager variational principle and how to design an associated hyperparameter selection strategy.

3.1. EONN

Firstly, the equation (6) is always accompanied by some boundary conditions, such as Dirichlet boundary conditions, Neumann boundary conditions, and natural boundary conditions.

$$\partial_t u-\frac{\epsilon}{\xi}\Delta_{\boldsymbol{x}}u+\frac{1}{\xi}f(u)=0, \tag{6}$$

where $f(u)=\partial_u F(u)$, $F(u)$ is a potential function (refer to the discussion in Appendix B). The Eq. (6), satisfying the Onsager variational principle, is derived from minimizing Rayleighian functional $\mathcal{R}(u,\dot{u})=\Phi(\dot{u})+\dot{\mathcal{E}}$:

$$\mathcal{R}(u,\dot{u})=\frac{\xi}{2}\int_\Omega\dot{u}^2d\boldsymbol{x}+\int_\Omega\epsilon\nabla_{\boldsymbol{x}}u\cdot\nabla_{\boldsymbol{x}}\dot{u}+\partial_u F(u)\dot{u}d\boldsymbol{x}, \tag{7}$$

where $\Phi(\dot{u})=\frac{\xi}{2}\int_\Omega\dot{u}^2d\boldsymbol{x}$ is usually called the dissipation of system and $\mathcal{E}=\int_\Omega\frac{\epsilon}{2}|\nabla u|^2+F(u)d\boldsymbol{x}$ is usually called the free energy of system. The evolutionary dynamics of the original system in $\mathbb{R}^d$ is parametrized by a neural network as follows:

$$u(\boldsymbol{x},t)\approx u^N(\boldsymbol{x},t)=\boldsymbol{G}(\boldsymbol{x},\boldsymbol{\theta}(t)),\quad \boldsymbol{x}\in\Omega, \tag{8}$$

where $\boldsymbol{G}(\boldsymbol{x},\boldsymbol{\theta}(t))$ is a neural network, $\boldsymbol{\theta}(t)\in\boldsymbol{\Theta}^K$ is the weight parameters of neural networks, K is the number of neurons.

According to equation (7), the Rayleighian functional is approximated by:

$$\begin{aligned}&\mathcal{R}(u,\dot{u})\approx R^N(u^N(\boldsymbol{x},\boldsymbol{\theta}(t)),\dot{u}^N(\boldsymbol{x},\boldsymbol{\theta}(t)))=\\&\frac{\xi}{2}\int_\Omega(u^N_{\boldsymbol{\theta}}\cdot\dot{\boldsymbol{\theta}}(t))^2+\int_\Omega\epsilon\nabla_{\boldsymbol{x}}u\cdot\nabla_{\boldsymbol{x}}(u^N_{\boldsymbol{\theta}}\cdot\dot{\boldsymbol{\theta}}(t))\\&+\partial_u F(u)u^N_{\boldsymbol{\theta}}\cdot\dot{\boldsymbol{\theta}}(t)d\boldsymbol{x},\end{aligned} \tag{9}$$

where $u^N_{\boldsymbol{\theta}}:=\nabla_{\boldsymbol{\theta}}u^N$, $\dot{\boldsymbol{\theta}}(t)$ is the time derivative of $\boldsymbol{\theta}$. The subtle structure in equation (9) allows for the separation of spatial $\boldsymbol{x}$ and time t. This separation induced a convex minimization problem (10) with respect to $\dot{\boldsymbol{\theta}}(t)$ while spatial $\boldsymbol{x}$ is integrated on $\Omega\in\mathbb{R}^d$.

$$R^N(\dot{u}^N(\boldsymbol{x},\boldsymbol{\theta}(t)))=\frac{1}{2}\dot{\boldsymbol{\theta}}(t)^T\boldsymbol{C}(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t)+(-\boldsymbol{b}(\boldsymbol{\theta}(t)))\dot{\boldsymbol{\theta}}(t), \tag{10}$$

where

$$\boldsymbol{C}(\boldsymbol{\theta}(t))=\xi\int_\Omega u^N_{\boldsymbol{\theta}}(u^N_{\boldsymbol{\theta}})^Td\boldsymbol{x}, \tag{11}$$

$$\boldsymbol{b}(\boldsymbol{\theta}(t))=-\int_\Omega\epsilon\nabla_x u\cdot\nabla_x u^N_{\boldsymbol{\theta}}+\partial_u F(\boldsymbol{x},u)u^N_{\boldsymbol{\theta}}d\boldsymbol{x}. \tag{12}$$

Those integrals are estimated by Monte Carlo methods, such as importance sampling and control variate method. Furthermore, the first-order optimality condition for this convex minimization problem is given by:

$$\boldsymbol{C}(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t)-\boldsymbol{b}(\boldsymbol{\theta}(t))=0. \tag{13}$$

However, numerical experiments (refer to Appendix E) indicate that most eigenvalues of the coefficient matrix $\boldsymbol{C}$ approach zeros. Consequently, this leads to numerical instability when solving ODE system (13). To address this problem, we introduce a regularization term λI for the original ODE system (13).

$$(\boldsymbol{C}(\boldsymbol{\theta}(t))+\lambda I)\dot{\boldsymbol{\theta}}(t)=\boldsymbol{b}(\boldsymbol{\theta}(t)), \tag{14}$$

which is derived from:

$$\mathcal{L}(\dot{\boldsymbol{\theta}}(t), \lambda) = \frac{1}{2}\dot{\boldsymbol{\theta}}(t)^T(\boldsymbol{C}(\boldsymbol{\theta}(t)) + \lambda I)\dot{\boldsymbol{\theta}}(t) + \boldsymbol{b}(\boldsymbol{\theta}(t))^T\dot{\boldsymbol{\theta}}(t). \tag{15}$$

The coefficient matrix is non-singular; thus, the unique ODE system is given by:

$$\dot{\boldsymbol{\theta}}(t) = (\boldsymbol{C}(\boldsymbol{\theta}(t)) + \lambda I)^{-1}\boldsymbol{b}(\boldsymbol{\theta}(t)). \tag{16}$$

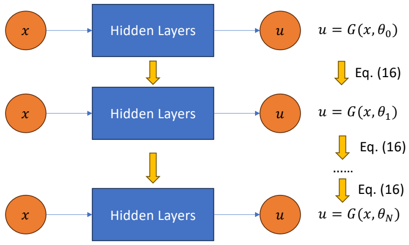


Figure 1. Structure of the EONN. The hidden layers can take any neural network structure. The θ_0 is trained by approximating the initial condition, followed by updating θ_t using Eq. (16).

This ODE system (16) represents a critical point in our approach, as it signifies the transition from a PDE problem to an ODE problem, simplifying the computational complexity. Notably, the resulting ODE (16) exhibits the essential property of energy stability, with the energy consistently decreasing over time (refer to discussion in Appendix B.2). To validate the efficiency of our method, we employ explicit Runge-Kutta methods to solve the ODE system (16). The numerical solution provides the approximate dynamics of the system:

$$u_\lambda^N(\boldsymbol{x}, t) = \boldsymbol{G}_\lambda(\boldsymbol{x}, \boldsymbol{\theta}(t)). \tag{17}$$

Here $\boldsymbol{G}_\lambda(\boldsymbol{x}, \boldsymbol{\theta}(t))$ is a neural network having $\boldsymbol{x}$ as its input, $\boldsymbol{\theta}(t)$ as weight parameter and λ as hyperparameters.This numerical approach ensures the accurate representation of the system's evolution, offering a practical and computationally efficient solution to complex partial differential equation problems.

The original dynamic processes are discretized through the application of EONN, employing Algorithm 1. For simplicity, the corresponding ODE systems are solved using the Forward Euler scheme, with variations in the chosen time step. Monte Carlo simulations are employed at each time step to estimate the coefficient matrix C (11) and vector $\boldsymbol{b}$ (12).

The introduced EONN framework emerges as an innovative neural network method distinguished by its unwavering fidelity in preserving the fundamental physical properties ingrained in the original system. In contrast to PINNs, EONNs exhibit exceptional proficiency in addressing highly nonlinear PDEs while delivering deterministic solutions. Moreover, the framework's utilization of lower derivative orders further mitigates singularity requirements, contributing to its versatility and robust performance. This characteristic significantly augments the reliability of the modeling process. A noteworthy capability of EONNs is its adeptness in handling irregular domains, setting it apart from conventional methods.

Algorithm 1 EONN Solver

```
Require: The PDE (6), initial condition and regularity term $\lambda$.
Set $t_k, k = 0, \dots, N, t_0 = 0, t_N = T, \boldsymbol{\theta}_k = \boldsymbol{\theta}(t_k)$.
Train $\boldsymbol{\theta}_0$ by approximating initial condition with $\boldsymbol{G}_\lambda(\boldsymbol{x}, \boldsymbol{\theta}_0)$
while $k \leq N$ do
    Calculate the coefficient $\mathbf{C}(\boldsymbol{\theta}_k)$ and $\mathbf{b}(\boldsymbol{\theta}_k)$.
    Update $\boldsymbol{\theta}_{k+1}$ explicitly by equation
        $(\mathbf{C}(\boldsymbol{\theta}_k) + \lambda I)\frac{\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k}{t_{k+1} - t_k} = \mathbf{b}(\boldsymbol{\theta}_k).$
end while
return Discretized solution $\boldsymbol{G}_\lambda(\boldsymbol{x}, \boldsymbol{\theta}_k), k = 0, ..., N$.
```

The validity of these claims is substantiated through numerical experiments conducted across various scientific computing scenarios. The results provide robust evidence in support of the superior capabilities of EONNs in faithfully simulating complex physical systems.

3.2. Onsager Machlup Hyper-parameter Selection

The introduction of the hyper-parameter λ in the modified ODE system (16) plays a crucial role in shaping the dynamics of the original system. Tuning hyperparameters is a fundamental aspect in artificial intelligence (AI) applications, particularly in the context of solving PDEs. Notably, the convergence behavior of PINNs (Raissi et al., 2019) is affected by the weighting factor assigned to different loss terms (Xiang et al., 2022; Riganti & Negro, 2023). Similarly, the DeepRitz method (Han et al., 2018) is sensitive to the predefined hyper-parameter β, although these approaches do not provide specific tuning strategies and instead directly assign values to the hyper-parameters. The Neural Galerkin method (Bruna et al., 2022) also introduces a regularizing term λ, but it does not provide guidance on selecting an appropriate λ.

To optimize the choice of λ and ensure the accuracy of the system's representation, we have devised a parameter selection strategy based on the Onsager-Machlup function. The Onsager-Machlup integral (5) is positive definite and attains zero only when $\boldsymbol{u}(t)$ matches the actual kinetic path. Consequently, selecting the optimal λ is akin to choosing the path with the lowest path energy $\mathcal{O}[\boldsymbol{u}_{\boldsymbol{\lambda}}(t)]$, where $u_\lambda^N(\boldsymbol{x}, t) = \boldsymbol{G}_\lambda(\boldsymbol{x}, \boldsymbol{\theta}(t))$ is calculated using Algorithm 1.

4. Related Work

In the last few years, there has been considerable research on numerically solving PDEs with neural network parametrization(Han et al., 2018; L. Wight & Zhao, 2021; E & Yu, 2018; Raissi et al., 2019; Bihlo & Popovych, 2022; Kharazmi et al., 2021; Ramabathiran & Ramachandran, 2021; Shukla et al., 2021; Jagtap & Karniadakis, 2020; Mao et al., 2020; Sirignano & Spiliopoulos, 2018; Bruna et al., 2022; Du & Zaki, 2021; Li et al., 2021; Rotman et al., 2023; Belbute-Peres et al., 2021; Schotthöfer et al., 2022). In the first category, the idea behind this is to train a neural network to well-approximate the solution of PDEs by minimizing the well-designed loss functions, such as DeepRitz method (DRM)(E & Yu, 2018),physics-informed neural network (PINNs)(Raissi et al., 2019; L. Wight & Zhao, 2021; Bihlo & Popovych, 2022; Kharazmi et al., 2021; Ramabathiran & Ramachandran, 2021; Shukla et al., 2021; Jagtap & Karniadakis, 2020; Mao et al., 2020), deep Galerkin method (DGM)(Sirignano & Spiliopoulos, 2018) and weak adversarial network (WAN) (Zang et al., 2020). PINNs stand as a prominent approach in scientific computing, with diverse applications across various domains. Yazdani et al. utilized PINNs for parameter inference and uncovering hidden dynamics in biological systems, while Lütjens et al. applied PINNs to climate modeling, assessing uncertainty propagation in ocean modeling. In fluid mechanics, Cai et al. provided an insightful review of PINNs applications. However, these methods encounter challenges related to stochastic training and hyperparameter sensitivities in loss functions, leading to potential inaccuracies and loss of physical meaning in the generated solutions. Moreover, a holistic approximation of solutions across the entire temporal-spatial domain may compromise the preservation of underlying variational structures in the original evolution PDEs. The equitable treatment of time and space in training proves inefficient for long-term PDE solution predictions. Additionally, the non-convex nature of the optimization problem in these methods raises concerns about convergence guarantee.

In the second category, researchers used neural networks to solve PDEs from a function approximation theory perspective rather than train neural networks to solve PDEs(Bruna et al., 2022; Du & Zaki, 2021; Li et al., 2023). They proposed a new parametrization framework that represents the spatial variable only by neural networks and the time evolution is realized by evolving in the neural network parameter space. However, they used the residual of the PDEs to derive the dynamics of parameters of neural networks. These dynamics make it hard to maintain the structure inherent in the physical problem. A regular term is artificially introduced to avoid singularity, which causes the resulting evolutionary process to deviate from the true process.

5. Implementation and Evaluation

In this section, we conducted numerical experiments to substantiate the theoretical arguments presented earlier and evaluated the performance of the EONN across various scientific domains. These domains include physics, fluid mechanics, computational biology, as well as scenarios involving irregular and high-dimensional cases (refer to the discussion in Appendix D).

5.1. Benchmark Comparison

Given that PINNs are a prominent method for solving PDEs with neural networks, we use PINNs as the benchmark for evaluating the performance of EONNs. This section aims to validate the superior performance of EONNs compared to PINNs, examining both linear and highly nonlinear PDEs.

5.1.1. Linear Case: Diffusion Equation

The diffusion equation characterizes the macroscopic behavior of micro-particles undergoing Brownian motion, a consequence of their random movements and collisions. We specifically consider the diffusion equation with zero boundary conditions on the one-dimensional spatial domain $x \in [-1, 1]$:

$$u_t = u_{xx}, \quad u(\pm 1, t) = 0, \tag{18}$$

as an example. Its analytic solution is given by: $u(x,t) = \exp(-\frac{\pi^2 t}{4})\cos(\frac{\pi}{2}x)$. Its corresponding free energy functional and dissipative functional are given by:

$$\mathcal{E} = \int \frac{1}{2}|u_x|^2 dx, \Phi(\dot{u}) = \frac{1}{2}\int_\Omega \dot{u}^2 dx. \tag{19}$$

The EONN with hard constrains(Lu et al., 2021b) is parametrized as follows:

$$u^N(x,t) = (x^2 - 1)\boldsymbol{G}(x, \boldsymbol{\theta}(t)). \tag{20}$$

The alignment between the EONN solution and the analytical solution is evident in Figure 2. The solution of PINN (implemented with DeepXDE(Lu et al., 2021a)) showcases consistency with the analytical solution. Overall, both EONN and PINN exhibit comparable error levels in addressing linear parabolic PDEs.

5.1.2. Nonlinear Case: Allen Cahn Equation

The Allen–Cahn (AC) equation, initially conceived as a phenomenological model for antiphase domain coarsening in a binary alloy (Allen & Cahn, 1979), has found extensive applications across diverse scientific domains. Modified versions of the AC equation have been employed in addressing a broad spectrum of problems, including but not limited to phase transitions (Allen & Cahn, 1979), image analysis (Beneš et al., 2004; Dobrosotskaya & Bertozzi, 2008),

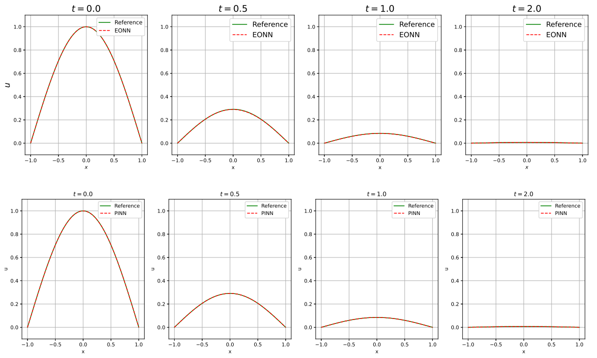


Figure 2. Diffusion equation (1D): Compare the analytical solution and EONN solution with $dt = 0.001, \lambda = 0.001$ and $u_0(x) = \cos(\frac{\pi}{2}x)$. Up: EONN Down: PINN

motion by mean curvature (Evans et al., 1992), two-phase fluid flows (Yang et al., 2006), crystal growth (Lee et al., 2012) and grain growth (Kobayashi et al., 2000). This versatility underscores the significance and applicability of the AC equation in capturing and elucidating various complex phenomena in different scientific disciplines.

We consider the Allen Cahn equation x in $[-1, 1]$ as follow:

$$u_t = \Delta u - \frac{1}{\epsilon^2}\partial_u F(u), \quad u(\pm 1, t) = -1. \tag{21}$$

Its corresponding free energy functional and dissipative functional are given by:

$$\Phi(\dot{u}) = \frac{1}{2}\int_\Omega \dot{u}^2\, dx, \dot{\mathcal{E}}(\dot{u}) = \int_\Omega \nabla u \cdot \nabla \dot{u} + \partial_u F(u)\dot{u} dx. \tag{22}$$

Set:$F(u) = \frac{(u^2-1)^2}{4\epsilon^2}$.Then:

$$\mathcal{R}(\dot{u}, u) = \frac{1}{2}\int_\Omega \dot{u}^2\, dx + \int_\Omega \nabla u \cdot \nabla \dot{u} + \frac{u^3 - u}{\epsilon^2}\dot{u}\, dx.$$

Minimizing $\mathcal{R}(\dot{u}, u)$ is equivalent to solving equation (21). The EONN with hard constrains is parametrized as follows:

$$u^N(x, t) = (x^2 - 1)^2 \boldsymbol{G}(x, \boldsymbol{\theta}(t)) - 1.$$

The EONN solution and its absolute error are shown in Figure 3.

Figure 3 serves as a pivotal illustration, showcasing the consistent approximation of the reference solution by the EONN throughout the evolution process. Notably, EONN demonstrates a robust convergence to the stable solution of the Allen Cahn equations. A direct comparison with the vanilla PINN method, implemented by DeepXDE, reveals the superior performance of EONN, as depicted in Figure 3. PINN fails to approximate the solution of Allen Cahn equation while there exists sharp transition layers(Wight & Zhao, 2020). Moreover, Figure 4 provides valuable insights into the solution behavior of EONN, highlighting its ability to preserve the energy decay property inherent in the system.

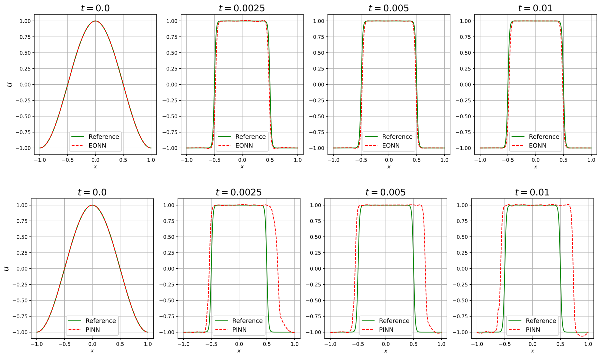
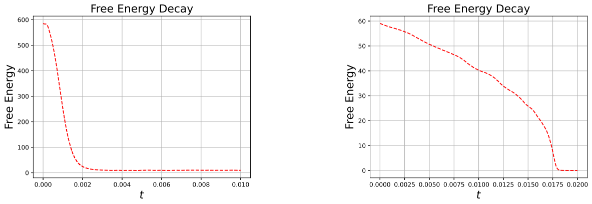


Figure 3. Allen Cahn equation (1D): Compare the reference solution (given by Finite difference method) and EONN solution with $\epsilon = 0.02$, $dt = 10^{-5}$. Using Onsager Machlup functional to select the best $\lambda = 0.01$ Up: EONN Down: PINN



(a) 1D Case

(b) 2D Case

Figure 4. Energy Decay of the Allen Cahn equation: The free energy of the system approximated by EONN. Left: Allen Cahn in 1D. Right: Allen Cahn in 2D.

5.2. Approximate Mean Curvature Flow and Hyperparameter Selection

The Allen Cahn equation (21) is the most natural diffuse-interface approximation for two-phase mean curvature flow. The limiting solutions of Allen Cahn, as $\epsilon \to 0$, converge to characteristic functions, whose motions are driven by mean curvature flow (Refer to Appendix B.2 for details).

We applied the EONN method to solve the Allen Cahn equations (2D) with boundary condition $u(\boldsymbol{x}, t) = -1, \boldsymbol{x} \in \partial\Omega, \Omega = [0, 1]^2$ and compared the numerical and analytic solutions in Figure 5. The EONN is parametrized as follows:

$$u^N(x, y, t) = g(x, y)\boldsymbol{G}(x, y; \boldsymbol{\theta}(t)) - 1,$$

where $g(x, y) = 1000(x^2 - x)^2(y^2 - y)^2$.

In Figure 6, the EONN solution, optimized with the hyperparameter $\lambda = 0.03$ as further detailed in the upcoming section, is visually presented. This meticulously selected

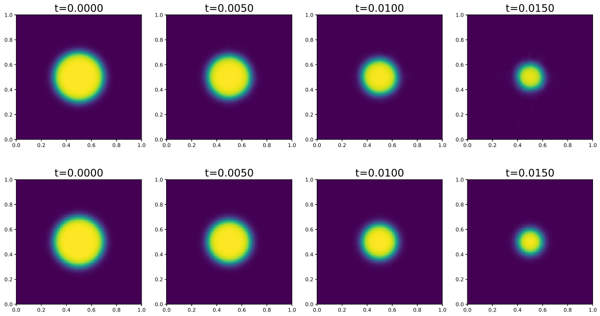


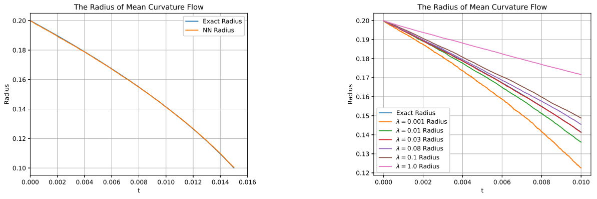
Figure 5. Allen Cahn equation (2D): Compare the reference so lution (given by Finite difference method (FDM)) and EONN solution with $\epsilon = 0.02$, $dt = 10^{-6}$, $\lambda = 0.03$. $\lambda = 0.03$ is the best hyper-parameter, which will be discussed in Section 5.2.1. Initial condition is given by $u_0(x, y) = \tanh(\frac{0.2-\sqrt{(x-0.5)^2+(y-0.5)^2}}{\sqrt{2}\epsilon})$, $x \in [0, 1]^2$. Up: EONN solutions Down: FDM solutions

hyper-parameter manifests in a solution that impeccably approximates the dynamics of the original mean curvature flow. The noteworthy performance showcased in this figure underscores the effectiveness of the hyper-parameter tuning process detailed in the subsequent section.

5.2.1. Hyperparameter Sensitivity Analysis

As regularization was introduced for the ODE system 16, the selection of the hyperparameter λ led to different dynamics.

Given different λ values, we simulated various dynamics and compared their Onsager Machlup functional values 5. The case with the lowest Onsager Machlup value provides the best approximation for the original Mean Curvature Flow, as shown in Figure 6.



(a) The best λ

(b) Different λ

Figure 6. Comparison of Mean Curvature Flow Solutions for various λ values. The circular interface is characterized as contour $u_\lambda^N = 0.R_0 = 0.2$. Figure (a) demonstrates that the EONN solution is consistent with the practical evolving dynamics, while Figure (b) shows the influence of different λ values (0.001, 0.01, 0.03, 0.08, 0.1).

In Figure 7, the optimal hyperparameter $\lambda = 0.03$, selected via the Onsager Machlup functional, is prominently displayed. This specific choice aligns seamlessly with achieving the most effective approximation of the original mean curvature flow. The consistency observed in this illustration not only underscores the accuracy of the selected hyperparameter but also affirms the exceptional performance of the Onsager Machlup functional as a strategy for hyperparameter selection.

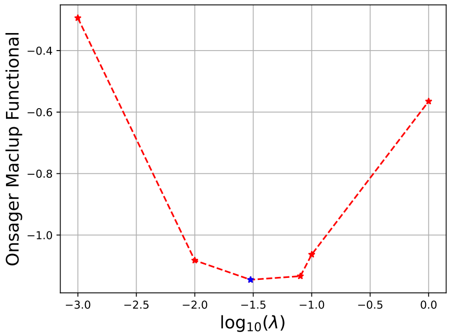


Figure 7. Onsager-Machlup Hyperparameter Tuning Strategy: Employing this strategy to select the optimal $\lambda = 0.03$, which minimizes the Onsager-Machlup functional (5). This choice of $\lambda = 0.03$ yields the best approximation for the mean curvature flow.

5.3. Biological Application: Fisher KPP equation

The Fisher-KPP equation, named after Ronald A. Fisher, Kolmogorov, Petrovsky, and Piscounov, describes the spatiotemporal dynamics of a population in one spatial dimension and is commonly used in mathematical biology and ecology(Du & Ni, 2021; Angstmann & Henry, 2020; Berestycki et al., 2009; El-Hachem et al., 2019; Hasnain & Saqib, 2017). The 1D Fisher-KPP equation can be written as:

$$u_t = du_{xx} + ru(1-u). \tag{23}$$

It has become one of the most important nonlinear equations and occurs in many biological and chemical processes. Eq. (23) is commonly associated with natural boundary conditions, represented as $\nabla u \cdot \boldsymbol{n} = 0$. These conditions can be derived from the variational structure (refer to Appendix B). Consequently, there is no need to impose hard constraints for PDEs with natural boundary conditions. Its corresponding free energy functional and dissipative functional are given by:

$$\Phi(\dot{u}) = \frac{1}{2}\int_\Omega \dot{u}^2 \, dx, \dot{\mathcal{E}}(\dot{u}) = \int_\Omega d\nabla u \cdot \nabla \dot{u} + \partial_u F(u)\dot{u} \, dx.$$

Set:$F(u) = -\frac{r}{3}u^3 + \frac{r}{2}u^2 + r$.The EONN is parametrized as follows:

$$u^N(x, t) = \boldsymbol{G}(x, \boldsymbol{\theta}(t)).$$

The EONN solution is shown in Figure 8. Figure 8 illus-

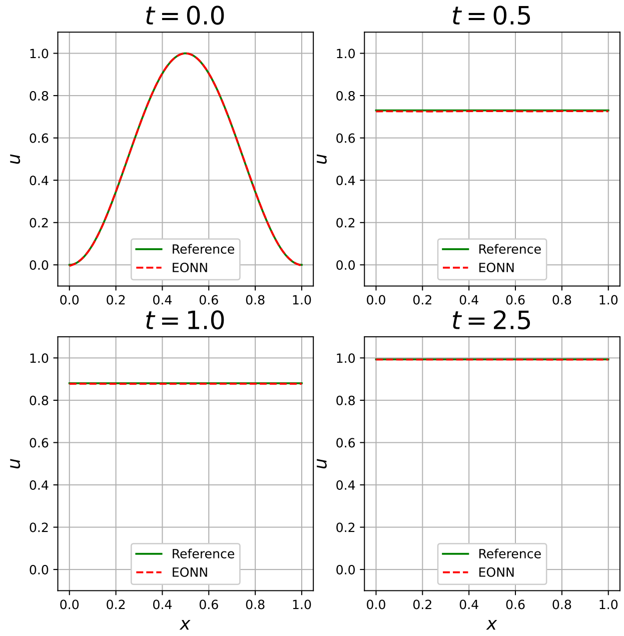


Figure 8. Fisher KPP equation (1D): Compare the reference solution (given by FDM) and EONN solution with $d = 0.5$, $r = 2.0$, $dt = 10^{-5}$. Using Onsager Machlup functional to select the best $\lambda = 0.01$.

trates that EONN is proficient in simulating biological population dynamics and aligns well with the reference solution obtained through the finite difference method.

5.4. Complex Geometries and Irregular Domains

Beyond solving PDEs on regular domains, the ability to capture complex geometry and rigorously account for nontrivial boundary conditions on irregular domains has been a driving force in the development of mesh generation and PDE solver technology(Hwang et al., 2023; Sethian & Shan, 2008). Our EONNs technology extends its applicability to solving PDEs on complex and irregular domains.

In this section, we illustrate the versatility of EONNs by solving the Allen-Cahn equation with natural boundary conditions on an irregular domain. The chosen domain is defined as:

$$\Omega_1 = \{(x, y) | (x, y) \in [0, 1]^2, (x, y) \notin [0.5, 1]^2\},$$
$$\Omega_2 = \{(x, y) = r(\theta)(\cos\theta, \sin\theta) \,|\, r(\theta) \leq 2 + 0.5\cos(3\theta)\}.$$

This example demonstrates the capability of EONNs to handle irregular geometries, showcasing its versatility across a wide range of problem domains. The EONN is parametrized as follows:

$$u^N(x, t) = \boldsymbol{G}(x, \boldsymbol{\theta}(t)).$$

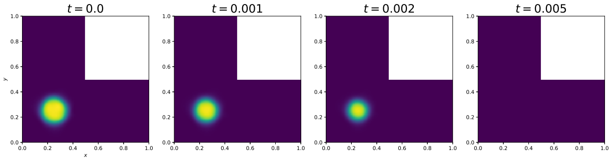


Figure 9. Allen Cahn equation (2D) with the irregular domain: EONN obtains an accurate approximation on Ω_1 with $\epsilon = 0.02$, $dt = 10^{-6}$, $\lambda = 0.03$. $\lambda = 0.03$ is the best hyper-parameter, which is quoted from the normal 2D case. Initial condition is given by $u_0(x, y) = \tanh(\frac{0.1 - \sqrt{(x-0.25)^2 + (y-0.25)^2}}{\sqrt{2}\epsilon})$, $x \in \Omega_1$.

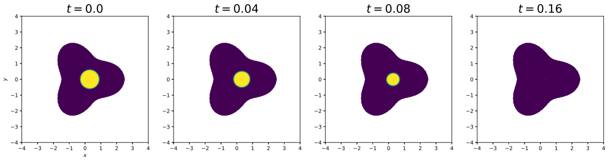


Figure 10. Allen Cahn equation (2D with the irregular domain): EONN obtains an accurate approximation on Ω_2 with $\epsilon = 0.02$, $dt = 10^{-6}$, $\lambda = 0.03$. $\lambda = 0.03$ is the best hyper-parameter, which is quoted from the normal 2D case. Initial condition is given by $u_0(x, y) = \tanh(\frac{0.6 - \sqrt{(x-0.3)^2 + y^2}}{\sqrt{2}\epsilon})$, $x \in \Omega_2$.

Traditional approaches encounter challenges when handling PDEs with irregular domains. Nevertheless, as depicted in Figure 10 and 9, the gradual evolution observed in the inner region of the initial condition mirrors that of a scenario with a regular domain. The outer region tends to maintain a nearly constant value of -1, making minimal contributions to the evolving dynamics of the inner portion. Consequently, the evolving dynamics of the inner part aligns with the behavior observed in regular cases.

6. Conclusion

In conclusion, this paper introduces EONNs, a novel framework that effectively addresses the challenges in applying neural networks in scientific computing. By combining neural networks with time-dependent parameters and leveraging the Onsager variational principle, EONNs offer a simple yet powerful approach. The framework's ability to preserve the variational structure of the underlying law, coupled with its versatility in handling irregular domains and lower derivative orders, marks a significant advancement. Comparative experiments demonstrate EONNs' comparable performance in linear PDEs and superior capabilities in highly nonlinear scenarios. The deterministic nature of EONN's solutions enhances reliability, and the proposed hyperparameter selection strategy proves effective. Overall, EONNs provide a promising avenue for tackling diverse scientific problems with a unified and theoretically grounded approach.

References

Allen, S. M. and Cahn, J. W. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979. ISSN 0001-6160. doi: https://doi.org/10.1016/0001-6160(79)90196-2. URL https://www.sciencedirect.com/science/article/pii/0001616079901962.

Angstmann, C. N. and Henry, B. I. Time fractional fisher–KPP and fitzhugh–nagumo equations. *Entropy*, 22(9):1035, 2020. ISSN 1099-4300. doi: 10.3390/e22091035. URL https://www.mdpi.com/1099-4300/22/9/1035.

Belbute-Peres, F. d. A., Chen, Y.-f., and Sha, F. Hyper-PINN: Learning parameterized differential equations with physics-informed hypernetworks, 2021. URL http://arxiv.org/abs/2111.01008.

Beneš, M., Chalupecký, V., and Mikula, K. Geometrical image segmentation by the Allen–Cahn equation. *Applied Numerical Mathematics*, 51(2):187–205, 2004. ISSN 0168-9274. doi: https://doi.org/10.1016/j.apnum.2004.05.001. URL https://www.sciencedirect.com/science/article/pii/S0168927404000819.

Berestycki, H., Nadin, G., Perthame, B., and Ryzhik, L. The non-local fisher–KPP equation: travelling waves and steady states. *Nonlinearity*, 22 (12):2813–2844, 2009. ISSN 0951-7715, 1361-6544. doi: 10.1088/0951-7715/22/12/002. URL https://iopscience.iop.org/article/10.1088/0951-7715/22/12/002.

Bihlo, A. and Popovych, R. O. Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics*, 456:111024, 2022. ISSN 00219991. doi: 10.1016/j.jcp.2022.111024. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999122000869.

Bruna, J., Peherstorfer, B., and Vanden-Eijnden, E. Neural galerkin scheme with active learning for high-dimensional evolution equations, 2022. URL http://arxiv.org/abs/2203.01360.

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.

Dobrosotskaya, J. A. and Bertozzi, A. L. A wavelet-laplace variational technique for image deconvolution and inpainting. *Trans. Img. Proc.*, 17(5):657–663, may 2008. ISSN 1057-7149. doi: 10.1109/TIP.2008.919367. URL https://doi.org/10.1109/TIP.2008.919367.

Doi, M. Onsager principle as a tool for approximation. *Chinese Physics B*, 24(2), 2015. ISSN 1674-1056. doi: 10.1088/1674-1056/24/2/020505. URL https://iopscience.iop.org/article/10.1088/1674-1056/24/2/020505.

Doi, M. Onsager's variational principle in soft matter. *Journal of Physics: Condensed Matter*, 23 (28):284118, 2022. ISSN 0953-8984, 1361-648X. doi: 10.1088/0953-8984/23/28/284118. URL https://iopscience.iop.org/article/10.1088/0953-8984/23/28/284118.

Du, Y. and Ni, W. The high dimensional fisher-KPP nonlocal diffusion equation with free boundary and radial symmetry, 2021. URL http://arxiv.org/abs/2102.05286.

Du, Y. and Zaki, T. A. Evolutional deep neural network. *Physical Review E*, 104(4):045303, 2021. ISSN 2470-0045, 2470-0053. doi: 10.1103/PhysRevE.104.045303. URL https://link.aps.org/doi/10.1103/PhysRevE.104.045303.

E, W. and Yu, B. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018. ISSN 2194-6701, 2194-671X. doi: 10.1007/s40304-018-0127-z. URL http://link.springer.com/10.1007/s40304-018-0127-z.

El-Hachem, M., McCue, S. W., Jin, W., Du, Y., and Simpson, M. J. Revisiting the fisher–kolmogorov–petrovsky–piskunov equation to interpret the spreading–extinction dichotomy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2229):20190378, 2019. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2019.0378. URL https://royalsocietypublishing.org/doi/10.1098/rspa.2019.0378.

Evans, L. C., Soner, H. M., and Souganidis, P. E. Phase transitions and generalized motion by mean curvature. *Communications on Pure and Applied Mathematics*, 45(9):1097–1123, 1992. doi: https://doi.org/10.1002/cpa.3160450903. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160450903.

Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115 (34):8505–8510, 2018. ISSN 0027-8424, 1091-6490.

doi: 10.1073/pnas.1718942115. URL https://pnas.org/doi/full/10.1073/pnas.1718942115.

Hasnain, S. and Saqib, M. Numerical study of one dimensional fishers KPP equation with finite difference schemes. *American Journal of Computational Mathematics*, 07(1):70–83, 2017. ISSN 2161-1203, 2161-1211. doi: 10.4236/ajcm.2017.71006. URL http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ajcm.2017.71006.

Hwang, Y., Ham, S., Lee, C., Lee, G., Kang, S., and Kim, J. A simple and efficient numerical method for the allen–cahn equation on effective symmetric triangular meshes. *Electronic Research Archive*, 31(8): 4557–4578, 2023. ISSN 2688-1594. doi: 10.3934/era.2023233. URL http://www.aimspress.com/article/doi/10.3934/era.2023233.

Jagtap, A. D. and Karniadakis, G. E. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2020.

Kharazmi, E., Zhang, Z., and Karniadakis, G. E. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021. ISSN 00457825. doi: 10.1016/j.cma.2020.113547. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782520307325.

Kobayashi, R., Warren, J. A., and Carter, W. C. A continuum model of grain boundaries. *Physica D: Nonlinear Phenomena*, 140(1):141–150, 2000. ISSN 0167-2789. doi: https://doi.org/10.1016/S0167-2789(00)00023-3. URL https://www.sciencedirect.com/science/article/pii/S0167278900000233.

L. Wight, C. and Zhao, J. Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. *Communications in Computational Physics*, 29(3):930–954, 2021. ISSN 1991-7120. doi: https://doi.org/10.4208/cicp.OA-2020-0086. URL http://global-sci.org/intro/article_detail/cicp/18571.html.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lee, D.-S., Lee, H. G., JEONG, D., Lee, C., Yang, D.-G., and Kim, J. A ROBUST AND ACCURATE PHASE-FIELD SIMULATION OF SNOW CRYSTAL GROWTH. *Journal of the Korean Society for Industrial and Applied Mathematics*, 16(1):15–29, 2012. doi: 10.12941/JKSIAM.2012.16.1.015. URL https://doi.org/10.12941/JKSIAM.2012.16.1.015.

Li, Y., Wu, Z., and Ye, F. Evolutionary neural networks for option pricing: Multi-assets option and exotic option. In *Proceedings of the AAAI Symposium Series*, volume 1, pp. 46–51, 2023.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations, 2021. URL http://arxiv.org/abs/2010.08895.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021a. doi: 10.1137/19M1274067.

Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., and Johnson, S. G. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021b.

Lütjens, B., Crawford, C. H., Veillette, M., and Newman, D. Pce-pinns: Physics-informed neural networks for uncertainty propagation in ocean modeling. *arXiv preprint arXiv:2105.02939*, 2021.

Mao, Z., Jagtap, A. D., and Karniadakis, G. E. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020. ISSN 00457825. doi: 10.1016/j.cma.2019.112789. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782519306814.

Onsager, L. Reciprocal relations in irreversible processes. i. *Physical Review*, 37(4):405–426, 1931a. ISSN 0031-899X. doi: 10.1103/PhysRev.37.405. URL https://link.aps.org/doi/10.1103/PhysRev.37.405.

Onsager, L. Reciprocal relations in irreversible processes. II. *Physical Review*, 38(12):2265–2279, 1931b. ISSN 0031-899X. doi: 10.1103/PhysRev.38.2265. URL https://link.aps.org/doi/10.1103/PhysRev.38.2265.

Onsager, L. and Machlup, S. Fluctuations and irreversible processes. *Phys. Rev.*, 91:1505–1512, Sep 1953. doi: 10.1103/PhysRev.91.1505. URL https://link.aps.org/doi/10.1103/PhysRev.91.1505.

Raissi, M., Perdikaris, P., and Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 00219991. doi: 10.1016/j.jcp.2018.10.045. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125.

Ramabathiran, A. A. and Ramachandran, P. SPINN: Sparse, physics-based, and partially interpretable neural networks for PDEs. *Journal of Computational Physics*, 445: 110600, 2021. ISSN 00219991. doi: 10.1016/j.jcp.2021.110600. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999121004952.

Riganti, R. and Negro, L. D. Auxiliary physics-informed neural networks for forward, inverse, and coupled radiative transfer problems. *Applied Physics Letters*, 123 (17):171104, 10 2023. ISSN 0003-6951. doi: 10.1063/5.0167155. URL https://doi.org/10.1063/5.0167155.

Rotman, M., Dekel, A., Ilan Ber, R., Wolf, L., and Oz, Y. Semi-supervised learning of partial differential operators and dynamical flows. In Evans, R. J. and Shpitser, I. (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 1785–1794. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/rotman23a.html.

Schotthöfer, S., Xiao, T., Frank, M., and Hauck, C. Structure preserving neural networks: A case study in the entropy closure of the boltzmann equation. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19406–19433. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/schotthofer22a.html.

Sethian, J. and Shan, Y. Solving partial differential equations on irregular domains with moving interfaces, with applications to superconformal electrodeposition in semiconductor manufacturing. *Journal of Computational Physics*, 227(13):6411–6447, 2008. ISSN 00219991. doi: 10.1016/j.jcp.2008.03.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999108001381.

Shukla, K., Jagtap, A. D., and Karniadakis, G. E. Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447: 110683, 2021. ISSN 00219991. doi: 10.1016/j.jcp.2021.110683. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999121005787.

Sirignano, J. and Spiliopoulos, K. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018. ISSN 00219991. doi: 10.1016/j.jcp.2018.08.029. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999118305527.

Thomas, P. D. and Lombard, C. K. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17(10):1030–1037, 1979. doi: 10.2514/3.61273. URL https://doi.org/10.2514/3.61273.

Wight, C. L. and Zhao, J. Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. *arXiv preprint arXiv:2007.04542*, 2020.

Xiang, Z., Peng, W., Liu, X., and Yao, W. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.05.015. URL https://www.sciencedirect.com/science/article/pii/S092523122200546X.

Yang, X., Feng, J., Liu, C., and Shen, J. Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method. *Journal of Computational Physics*, 218(1):417–428, October 2006. ISSN 0021-9991. doi: 10.1016/j.jcp.2006.02.021.

Yazdani, A., Lu, L., Raissi, M., and Karniadakis, G. E. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS computational biology*, 16(11):e1007575, 2020.

Zang, Y., Bao, G., Ye, X., and Zhou, H. Weak adversarial networks for high-dimensional partial differential equations. *Journal of Computational Physics*, 411: 109409, 2020. ISSN 00219991. doi: 10.1016/j.jcp.2020.109409. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999120301832.

A. Implementation details of numerical experienments

A.1. PINNs

The PINNs algorithms for the diffusion and Allen Cahn equations share a common structure, featuring 2 input neurons, 4 hidden layers with 25 neurons each, and tanh activation functions. Despite efforts to enhance the sample size in the Allen Cahn case up to 200,000, PINNs struggled to effectively address the highly nonlinear nature of the PDE (with $\epsilon = 0.02$).

A.2. EONN for Diffusion equation and Allen Cahn equations

As for EONN applied to the Diffusion and Allen Cahn equations, the neural network structures are outlined as follows:

- 1D case: Neural network has 1 input neurons, 4 hidden layers, 25 neurons for each layer, tanh activation functions.
- 2D case: Neural network has 2 input neurons, 4 hidden layers, 30 neurons for each layer, tanh activation functions.

A.3. EONN for Fisher KPP equation

To validate the versatility of EONN, we employ a more intricate structure to solve the Fisher-KPP equation. The network architecture is depicted in the accompanying Figure 11.

B. Variation structure with Natural Boundary Condition

Let u represent the state of a material, and let $F(u)$ denote the local energy per unit volume of the material. Assuming that the material's motion is influenced by the gradient ∇u with a coefficient ϵ, a suitable measure for the material's action is given by

$$\mathcal{E}(u) := \int_\Omega \left(\frac{1}{2}\epsilon^2 |\nabla u|^2 + F(u) \right) d\boldsymbol{x}$$

Here, Ω represents the fixed domain in which the material is confined. Employing integration by parts and the natural boundary condition $u \cdot \boldsymbol{n} = 0$, where $\boldsymbol{n}$ is the normal vector.

The gradient of $E(u)$ is determined using the standard method, yielding

$$\frac{\delta \mathcal{E}(u)}{\delta u} = -\epsilon^2 \Delta u + F'(u).$$

B.1. Derive evolution PDEs with Onsager variational principle

Particularly, we consider the special case that we denote by Ω a domain in $\mathbb{R}^n$. Suppose the energy functional is given by

$$\mathcal{E}(u, \nabla u) = \int_\Omega \frac{\epsilon}{2} |\nabla u|^2 + F(u) d\boldsymbol{x}, \tag{24}$$

where $\epsilon > 0$ is the diffusion coefficient, and $F(\boldsymbol{x}, u)$ is a function with respect to $\boldsymbol{x}$ and u. We assume that $u \in H_0^1(\Omega)$ and $\dot{u} \in L^2(\Omega)$. The dissipation function is given by

$$\Phi(\dot{u}) = \frac{\xi}{2} \int_\Omega \dot{u}^2 d\boldsymbol{x}, \tag{25}$$

with a positive friction coefficient ξ. Then the Rayleighian is calculated as $\mathcal{R}(u, \dot{u}) = \Phi(\dot{u}) + \dot{\mathcal{E}}$:

$$\mathcal{R}(u, \dot{u}) = \frac{\xi}{2} \int_\Omega \dot{u}^2 d\boldsymbol{x} + \int_\Omega \epsilon \nabla u \cdot \nabla \dot{u} + \partial_u F(u) \dot{u} d\boldsymbol{x}. \tag{26}$$

In order to derive the corresponding PDE, we minimize $\mathcal{R}$ with respect to $\dot{u}$. The corresponding Euler-Lagrange equation is

$$\partial_t u - \frac{\epsilon}{\xi} \Delta u + \frac{1}{\xi} f(u) = 0. \tag{27}$$

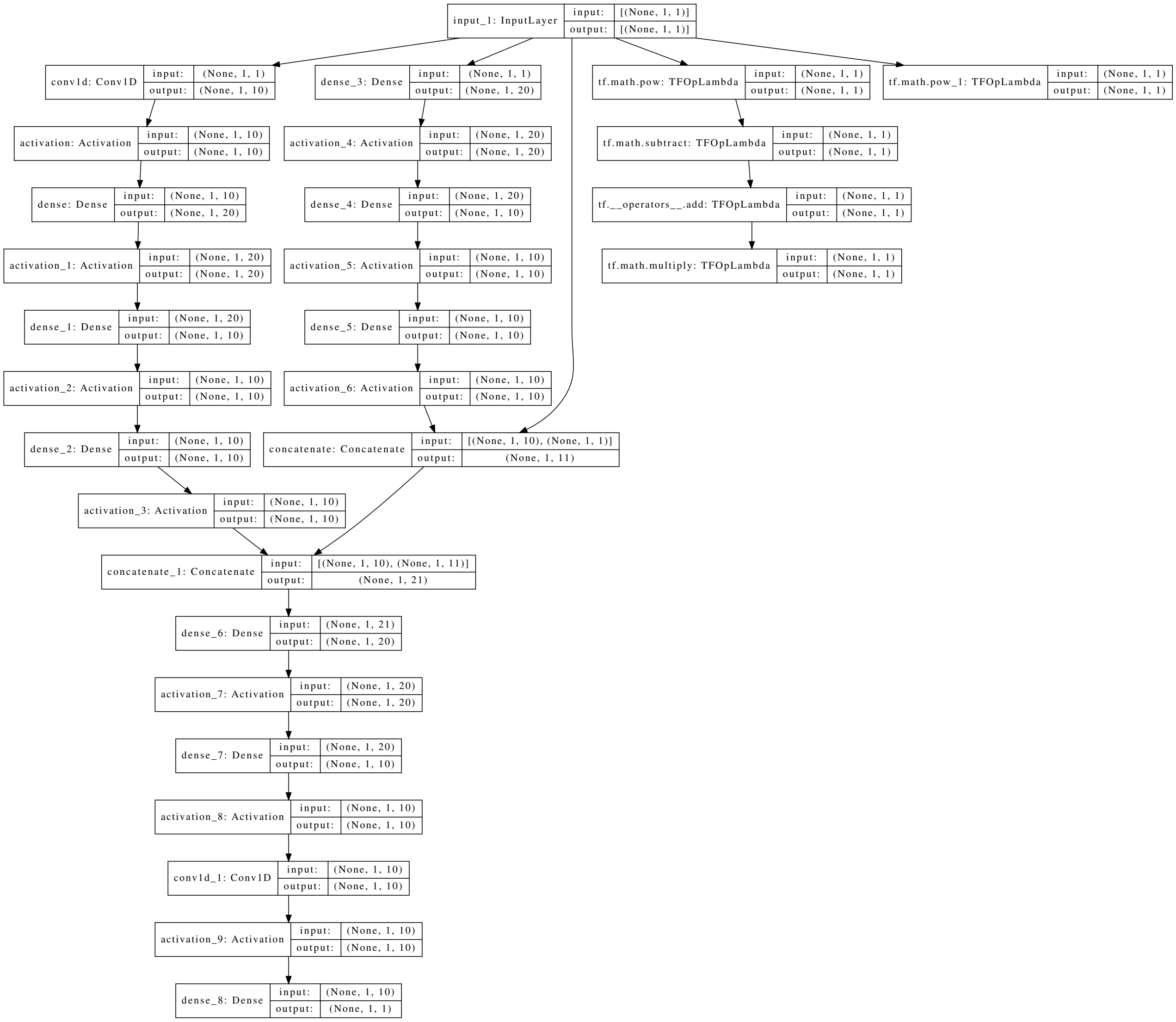

input_1: InputLayer | input: [(None, 1, 1)] | output: [(None, 1, 1)]
conv1d: Conv1D | input: (None, 1, 1) | output: (None, 1, 10)
dense_3: Dense | input: (None, 1, 1) | output: (None, 1, 20)
tf.math.pow: TFOpLambda | input: (None, 1, 1) | output: (None, 1, 1)
tf.math.pow_1: TFOpLambda | input: (None, 1, 1) | output: (None, 1, 1)
activation: Activation | input: (None, 1, 10) | output: (None, 1, 10)
activation_4: Activation | input: (None, 1, 20) | output: (None, 1, 20)
tf.math.subtract: TFOpLambda | input: (None, 1, 1) | output: (None, 1, 1)
dense: Dense | input: (None, 1, 10) | output: (None, 1, 20)
dense_4: Dense | input: (None, 1, 20) | output: (None, 1, 10)
tf.__operators__.add: TFOpLambda | input: (None, 1, 1) | output: (None, 1, 1)
activation_1: Activation | input: (None, 1, 20) | output: (None, 1, 20)
activation_5: Activation | input: (None, 1, 10) | output: (None, 1, 10)
tf.math.multiply: TFOpLambda | input: (None, 1, 1) | output: (None, 1, 1)
dense_1: Dense | input: (None, 1, 20) | output: (None, 1, 10)
dense_5: Dense | input: (None, 1, 10) | output: (None, 1, 10)
activation_2: Activation | input: (None, 1, 10) | output: (None, 1, 10)
activation_6: Activation | input: (None, 1, 10) | output: (None, 1, 10)
dense_2: Dense | input: (None, 1, 10) | output: (None, 1, 10)
concatenate: Concatenate | input: [(None, 1, 10), (None, 1, 1)] | output: (None, 1, 11)
activation_3: Activation | input: (None, 1, 10) | output: (None, 1, 10)
concatenate_1: Concatenate | input: [(None, 1, 10), (None, 1, 11)] | output: (None, 1, 21)
dense_6: Dense | input: (None, 1, 21) | output: (None, 1, 20)
activation_7: Activation | input: (None, 1, 20) | output: (None, 1, 20)
dense_7: Dense | input: (None, 1, 20) | output: (None, 1, 10)
activation_8: Activation | input: (None, 1, 10) | output: (None, 1, 10)
conv1d_1: Conv1D | input: (None, 1, 10) | output: (None, 1, 10)
activation_9: Activation | input: (None, 1, 10) | output: (None, 1, 10)
dense_8: Dense | input: (None, 1, 10) | output: (None, 1, 1)

Figure 11. Neural network's structure for solving Fish KPP equation.

Given $F(u) = 0$, we get the diffusion equation:

$$\partial_t u - \frac{\epsilon}{\xi}\Delta u = 0. \tag{28}$$

Given $F(u) = \frac{(u^2-1)^2}{4\epsilon}$ and $\xi = \epsilon$, we get the Allen Cahn equation:

$$\partial_t u - \Delta u + \frac{u^3 - u}{\epsilon^2} = 0. \tag{29}$$

which approximates the mean curvature flow when $\epsilon \to 0$.

B.2. Mean Curvature Flow

In two-dimensional space, the normal velocity of the circular interface satisfies the following geometric law (Thomas & Lombard, 1979): $V = -\kappa = -\frac{1}{R}$, where V is the velocity, κ is the curvature, and R is the radius. Given the initial condition

$$u_0(x, y) = \tanh\left(\frac{0.2 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right).$$

Let R_0 be the initial radius; then, the analytic solution of circular interface can be expressed as:

$$R(t) = \sqrt{R_0^2 - 2t}. \tag{30}$$

B.3. Energy decay of variational structure

An important feature of equation (6) is that the problem can be characterized by the gradient flow of Liapunov energy functional $\mathcal{E}(u) = \int_\Omega \left(\frac{\epsilon}{2}|\nabla u|^2 + F(u)\right) d\boldsymbol{x}$. in L^2. We can obtain the energy law as

$$\frac{\partial \mathcal{E}(u(t))}{\partial t} = -\int_\Omega |-\epsilon\Delta u + f(u)|^2 d\boldsymbol{x} \le 0, \tag{31}$$

which yields the following energy-decay property:

$$\mathcal{E}\left(u\left(t_n\right)\right) \le \mathcal{E}\left(u\left(t_m\right)\right), \quad t_n > t_m. \tag{32}$$

C. EONN in H^{-1} Space

In this section, we also designed specific neural networks with exact boundary conditions for PDEs. The EONN are also capable of Ongager variational principle in H^{-1} space. The diffusion equation with Neumann boundary condition:

$$u_t = u_{xx}, \quad u_0(x) = \cos(\pi x), \quad x \in [-1, 1]. \tag{33}$$

Its analytic solution is given by:

$$u(x, t) = \exp(-\pi^2 t)\cos(\pi x).$$

Its corresponding free energy functional and dissipative functional are given by:

$$\begin{aligned} \mathcal{E} &= \frac{1}{2}\int u^2 dx, \\ \Phi(\dot{u}) &= \frac{1}{2} < \dot{u}, \dot{u} >_{H_{-1}} . \end{aligned} \tag{34}$$

We can get:

$$\triangle^{-1}\dot{u} = u.$$

$$u_t = \triangle u,$$

Suppose any $u \in H^{-1}$

$$\eta_x(x, t) \approx \boldsymbol{G}(x, \boldsymbol{\theta}(t)) = f(x, \boldsymbol{\theta}(t).) \tag{35}$$

Then the Rayleighian functional is approximated by:

$$\begin{aligned} \mathcal{R}^N(\dot{\eta}_x(x, \boldsymbol{\theta}(t))) &= \Phi(\dot{\eta}_x) + \dot{\mathcal{E}}(\dot{\eta}_x, u) \\ \mathcal{R}^N(\dot{\boldsymbol{\theta}}(t)) &= \frac{1}{2}\int (\mathbf{f}_{\boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}}(t))^2 + \int (-\triangle u + \partial_u F(x, u))\mathbf{u}_{\boldsymbol{\theta}} \cdot \dot{\boldsymbol{\theta}}(t) dx, \end{aligned} \tag{36}$$

where

$$\mathbf{u}_{\boldsymbol{\theta}} := \nabla_{\boldsymbol{\theta}} \frac{\partial f}{\partial x}.$$

Then:

$$u(x,t) = \eta_{xx}(x,t) \approx f_x(x, \boldsymbol{\theta}(t)).$$

The separation of spatial x and time t induces a convex minimization problem with respect to $\dot{\boldsymbol{\theta}}(t)$ while spatial x is integrated on $\Omega \in R^d$. Rewrite:

$$R^N(\dot{\boldsymbol{\theta}}(t)) = \frac{1}{2}\dot{\boldsymbol{\theta}}(t)^T C(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) + (-b(\boldsymbol{\theta}(t)))\dot{\boldsymbol{\theta}}(t), \tag{37}$$

where

$$\begin{aligned} C(\boldsymbol{\theta}(t)) &= \int \mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{f}_{\boldsymbol{\theta}}{}^T dx, \\ b(\boldsymbol{\theta}(t)) &= \int (\triangle u - \partial_u F(x,u))\mathbf{f}_{\boldsymbol{\theta}} dx. \end{aligned} \tag{38}$$

Minimize $\mathcal{R}^N$ in $\dot{\boldsymbol{\theta}}(t)$:

$$\frac{\partial R^N}{\partial \dot{\boldsymbol{\theta}}(t)} = 0.$$

We get:

$$C(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) = b(\boldsymbol{\theta}(t)). \tag{39}$$

Unfortunately, C is a semi-positive definite matrix; this linear ODE system is not uniquely determined. Therefore, we introduce a regularization term:

$$(C(\boldsymbol{\theta}(t)) + \lambda I)\dot{\boldsymbol{\theta}}(t) = b(\boldsymbol{\theta}(t)). \tag{40}$$

$$\dot{\boldsymbol{\theta}}(t) = A(\boldsymbol{\theta}(t))b(\boldsymbol{\theta}(t)).$$

where $A(\boldsymbol{\theta}(t)) = (C(\boldsymbol{\theta}(t)) + \lambda I)^{-1}$ Solving this ODE system with a numerical solver gives the evolution of the original system.

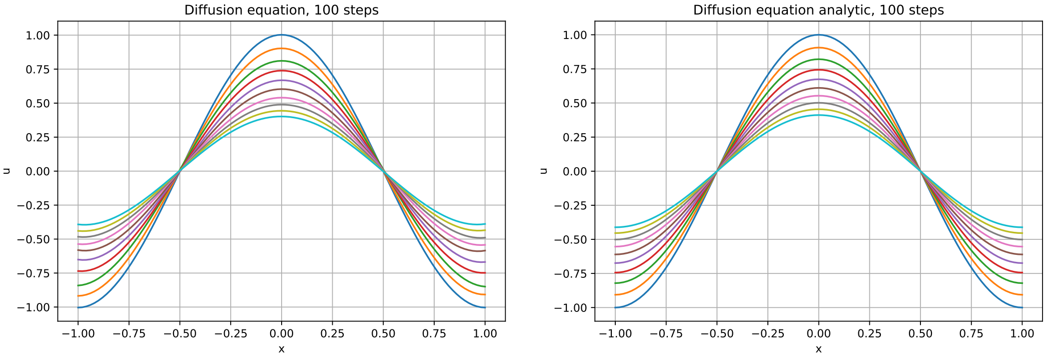


Figure 12. Diffusion equation(1D): EONN obtains an accurate approximation with $dt = 0.001, \lambda = 0.001$ in H^{-1} space Left: Analytic solutions Right: EONN solutions

D. High Dimension Diffusion Equation

We consider the diffusion equation with zero boundary condition on the ten-dimensional spatial domain $x \in [-1,1]^{10}$

$$u_t = \Delta u, \quad x \in [-1,1]^{10}, \tag{41}$$

$$\begin{aligned} u(\boldsymbol{x}, 0) = \sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3)\sin(\pi x_4)\sin(\pi x_5) \\ \sin(\pi x_6)\sin(\pi x_7)\sin(\pi x_8)\sin(\pi x_9)\sin(\pi x_{10}). \end{aligned} \tag{42}$$

Its analytical solution is given by:

$$u(\boldsymbol{x},t) = \exp(-10\pi^2 t)\sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3)\sin(\pi x_4)$$
$$\sin(\pi x_5)\sin(\pi x_6)\sin(\pi x_7)\sin(\pi x_8)\sin(\pi x_9)\sin(\pi x_{10}). \tag{43}$$

Its corresponding free energy functional and dissipative functional are given by:

$$\begin{aligned}\mathcal{E} &= \int \frac{1}{2}|\nabla u|^2,\\ \Phi(\dot{u}) &= \frac{1}{2}\int_{\Omega} \dot{u}^2 d\boldsymbol{x}.\end{aligned} \tag{44}$$

The EONN is parametrized as follows:

$$u^N(\boldsymbol{x},t) = 10000(x_1^2-1)(x_2^2-1)(x_3^2-1)(x_4^2-1)(x_5^2-1)$$
$$(x_6^2-1)(x_7^2-1)(x_8^2-1)(x_9^2-1)(x_{10}^2-1)\boldsymbol{G}(\boldsymbol{x},\boldsymbol{\theta}(t)) \tag{45}$$

where $\boldsymbol{G}(\boldsymbol{x},\boldsymbol{\theta}(t))$ is a fully connected neural network with 5 hidden layers and 40 neurons in each hidden layer. The corresponding ODE system is solved by the Forward Euler scheme with timestep $dt = 0.0001$. To estimate the coefficient matrix C (11) and vector $\boldsymbol{b}$ (12), we applied Monte Carlo simulation with 10000 samples at each time step. The EONN solution and its absolute error are shown in Figure 13.

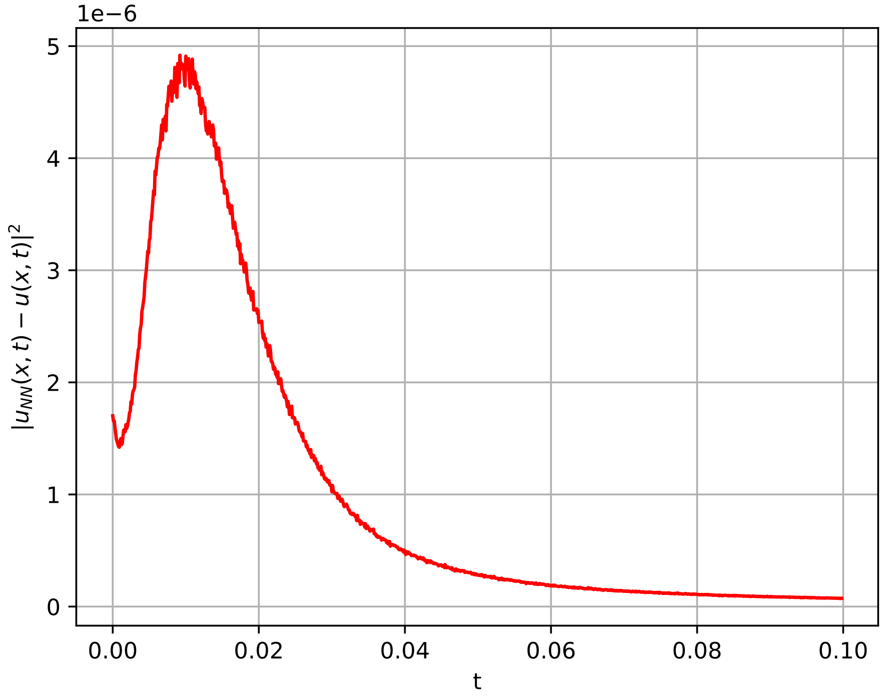
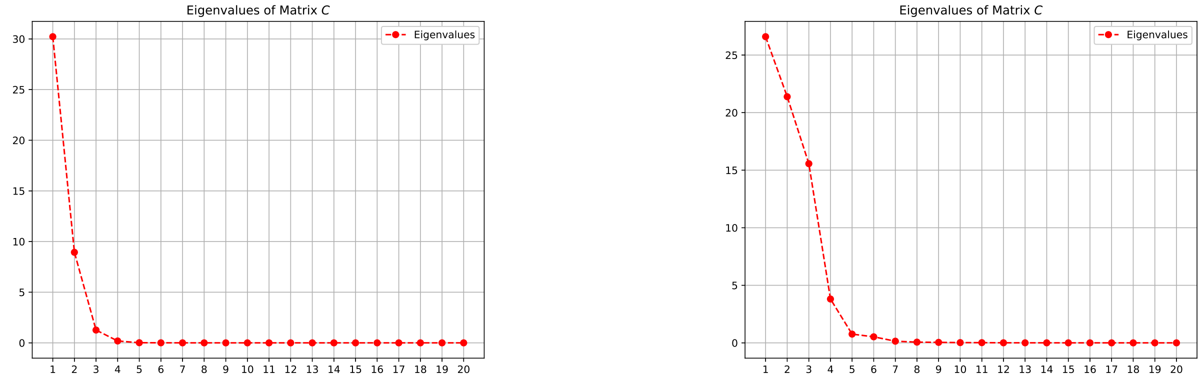


Figure 13. Diffusion equation (10D): EONN(5 hidden layers, 40 neurons for each layer, tanh activation functions) obtains an accurate approximation with $dt = 0.0001$ and $\lambda = 0.00001$. L_2 error of EONN solutions, estimated by Monte Carlo simulation with 10000 samples at each time step.

E. Eigenvalues of Matrix C



(a)

(b)

Figure 14. Eigenvalues of the matrix C at different time steps are illustrated in (a) and (b). The plots demonstrate that the eigenvalues of the matrix C are predominantly influenced by the first few components, while the remaining eigenvalues converge to nearly zero.