

Physics-informed neural networks for solving functional renormalization group on a latticeTakeru Yokota ^{*}*Interdisciplinary Theoretical and Mathematical Sciences Program (iTHEMS), RIKEN, Wako, Saitama 351-0198, Japan*

(Received 5 January 2024; revised 31 May 2024; accepted 4 June 2024; published 14 June 2024)

Addressing high-dimensional partial differential equations to derive effective actions within the functional renormalization group is formidable, especially when considering various field configurations, including inhomogeneous states, even on lattices. We leverage physics-informed neural networks (PINNs) as a state-of-the-art machine-learning method for solving high-dimensional partial differential equations to overcome this challenge. In a zero-dimensional $O(N)$ model, we numerically demonstrate the construction of an effective action on an N -dimensional configuration space, extending up to $N = 100$. Our results underscore the effectiveness of PINN approximation, even in scenarios lacking small parameters such as a small coupling.

DOI: [10.1103/PhysRevB.109.214205](https://doi.org/10.1103/PhysRevB.109.214205)**I. INTRODUCTION**

The utilization of the functional renormalization group (FRG) [1–4] has gained widespread popularity as a nonperturbative theoretical tool across diverse fields, encompassing high-energy physics, condensed-matter physics, and statistical physics (see Refs. [5–8] for comprehensive reviews). Central to the FRG is the utilization of functional differential equations (FDE), such as the Wetterich equation [4], which plays a pivotal role in describing the flow within the renormalization group (RG). Notably, the self-determination of the effective action (encompassing all the correlation information of a system) through the Wetterich equation enhances the precision and comprehensiveness of the FRG formalism. Despite these advantages, the absence of universal, efficient, and accurate algorithms for solving FDEs hampers an easily accessible and accurate determination of various properties. The quest for such algorithms or useful approximation schemes remains an open problem.

Power series expansions, such as the vertex expansion (the functional Taylor expansion) and the derivative expansion, are commonly employed in solving the FRG. In these approaches, the Wetterich equation transforms into an infinite hierarchy of differential equations for the expansion coefficients. These coefficients are subsequently truncated in a certain order to facilitate approximate solutions. However, the effective action $\Gamma[\varphi]$ is only valid for specific field configurations $\varphi(x)$, where x is a spatiotemporal coordinate. For instance, the effective action derived from the vertex expansion is valid in the vicinity of the expansion point $\varphi(x) \approx \varphi_{\text{exp}}(x)$, while that obtained from the derivative expansion is applicable when $\varphi(x) \approx \text{const}$. In these cases, prior knowledge of the field's ground state is a prerequisite for calculations, limiting the ability to capture complex structures such as inhomogeneous states. Moreover, enhancing the accuracy of results often

entails computationally demanding efforts in improving the truncation order.

The application of FRG extends beyond continuum models and is commonly employed in lattice models as well [7,8]. On a finite lattice, the Wetterich equation becomes an $(N_{\text{DOF}} + 1)$ -dimensional partial differential equation (PDE) involving N_{DOF} degrees of freedom for the field variables and the RG scale. While a finite-dimensional PDE might appear more amenable to numerical analysis than an FDE, the computational complexity of calculations with a large N_{DOF} grows exponentially when a computational grid is assigned to each field component. Therefore, even in lattice models, approximations based on power series expansions remain commonly employed.

This study aims to demonstrate that machine learning offers a framework for solving the FRG applied to lattice models with large N_{DOF} as an alternative to power series expansions. Among the array of recently developed machine-learning methods for handling high-dimensional PDEs [9–21], we leverage the physics-informed neural networks (PINNs) [9,11,19]. PINNs can be applied to various PDEs and involve optimizing a differentiable neural network (NN) to satisfy PDE and boundary conditions, providing a solution for a domain of input variables' space rather than a single point. Because of its grid-free characteristic, PINNs are particularly advantageous for handling high-dimensional inputs, as demonstrated in recent applications to high-dimensional PDEs [22–28], including 10^5 -dimensional cases [26,28]. In such scenarios, the limitations imposed by N_{DOF} are naturally expected to relax, implying the possibility of simultaneously constructing effective actions for various field configurations, including inhomogeneous states. Moreover, the universal approximation theorem [29–31] suggests that NNs can serve as accurate approximations for effective action.

In the subsequent sections, we present a PINN-based method for solving the FRG applied to a lattice (PINN-LFRG). We outline a methodology for representing the effective action using a differentiable NN, which is trained to satisfy the Wetterich equation. Furthermore, we provide

^{*}Contact author: takeru.yokota@riken.jp

numerical demonstrations of the scalability and accuracy of this approach in the zero-dimensional $O(N)$ model. Here, PDEs with $N_{\text{DOF}} + 1 = N + 1 \leq 101$ dimensions are solved within a few hours. The effective action and self-energy are computed across a domain of the field space simultaneously, exhibiting superior or comparable accuracies when contrasted with results obtained through perturbative and large- N expansions, spanning various choices of coupling strength and N . Additionally, the $O(N)$ symmetry for the effective action is successfully reproduced through training on the Wetterich equation. These findings underscore the feasibility of utilizing NNs to approximate the effective action even without a small parameter, such as a small coupling. We note that our purpose of solving FRG flow equations differs from that in a recent machine-learning-based FRG study [32], which focuses on the dimensionality reduction of the four-point vertex function as given by FRG.

The remaining part of this paper is organized as follows: In the following section (Sec. II), we briefly summarize the FRG formulation for bosons in a lattice and illustrate our idea for applying PINNs to solve the Wetterich equation. In Sec. III, we present a numerical demonstration of our approach in the zero-dimensional $O(N)$ model. Section IV is devoted to the conclusions.

II. GENERAL FORMULATION

We focus on the FRG applied to bosons in a d -dimensional space-time lattice. The action is represented by $S(\boldsymbol{\varphi})$, where $\boldsymbol{\varphi} = \{\varphi_{n,\alpha}\}_{n,\alpha}$ is a real bosonic field. Here, the d -dimensional vector n indicates a lattice site and α is the internal degrees of freedom index. The total degrees of freedom for this system are given by $N_{\text{DOF}} = VN_{\text{IDOF}}$, where V denotes the lattice volume, and N_{IDOF} is the internal degree of freedom. The imaginary-time formalism is employed, and all quantities are expressed in lattice units.

We adhere to the formalism outlined by Wetterich [4]. Following this formalism, a regulator term is introduced into the action to induce the RG flow,

$$S_k(\boldsymbol{\varphi}) = S(\boldsymbol{\varphi}) + \frac{1}{2} \sum_{n,\alpha,n,\alpha'} \varphi_{n,\alpha} R_{k,n-n'}^{\alpha\alpha'} \varphi_{n',\alpha'}. \quad (1)$$

The regulator $R_{k,n-n'}^{\alpha\alpha'}$ is a predefined function acting as an artificial mass, designed to dampen fluctuations with momenta smaller than the RG scale k . In the momentum space, the regulator must adhere to the following conditions:

$$\lim_{p^2/k^2 \rightarrow 0} \tilde{R}_k(p) > 0, \quad (2a)$$

$$\tilde{R}_{k_{\text{IR}} \rightarrow 0}(p) = 0, \quad (2b)$$

$$\tilde{R}_{k_{\text{UV}} \rightarrow \infty}(p) = \infty. \quad (2c)$$

For simplicity, we have omitted the indices for the internal degrees of freedom. The first condition signifies the suppression of infrared fluctuations, while the second condition ensures that all fluctuations are included at a small infrared scale k_{IR} . The final condition is crucial for determining the initial condition of the RG flow. It ensures that the system becomes classical, described by $S_{k_{\text{UV}}}(\boldsymbol{\varphi})$, at a large ultraviolet scale k_{UV} . In terms of the path integral introduced below, this condition

validates the saddle-point approximation at $k = k_{\text{UV}}$. Compared to the continuous one, the distinctions in the lattice setup lie in the dispersion relation and the restriction of the momentum to the Brillouin zone. An appropriate regulator choice that accommodates these differences is discussed in Ref. [33].

With this regulator, one can define the effective average action $\Gamma_k(\boldsymbol{\varphi})$, which interpolates between the bare action $S(\boldsymbol{\varphi})$ and the effective action $\Gamma(\boldsymbol{\varphi})$. The definition is

$$\Gamma_k(\boldsymbol{\varphi}) = \sup_{\mathbf{J}} \left(\sum_{n,\alpha} J_{n,\alpha} \varphi_{n,\alpha} - \ln Z_k(\mathbf{J}) \right) - \frac{1}{2} \sum_{n,\alpha,n,\alpha'} \varphi_{n,\alpha} R_{k,n-n'}^{\alpha\alpha'} \varphi_{n',\alpha'}, \quad (3)$$

with the path-integral form of the partition function

$$Z_k(\mathbf{J}) = \int d\boldsymbol{\varphi} e^{-S_k(\boldsymbol{\varphi}) + \alpha^d \sum_{n,\alpha} J_{n,\alpha} \varphi_{n,\alpha}}. \quad (4)$$

The condition $\lim_{k \rightarrow 0} \Gamma_k(\boldsymbol{\varphi}) = \Gamma(\boldsymbol{\varphi})$ immediately follows from Eq. (2b). From the saddle-point approximation validated by Eq. (2c), we have $\Gamma_{k_{\text{UV}}}(\boldsymbol{\varphi}) = S(\boldsymbol{\varphi}) + \text{const}$. The RG flow equation is derived as an $(N_{\text{DOF}} + 1)$ -dimensional PDE by the derivative of Eq. (3) with respect to k ,

$$\partial_k \Gamma_k(\boldsymbol{\varphi}) = \frac{1}{2} \text{tr} \left[\partial_k R_k \left(\frac{\partial^2 \Gamma_k(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_k \right)^{-1} \right], \quad (5)$$

which is known as the Wetterich equation. Here, the inverse is defined by

$$\sum_{n',\alpha'} \left(\frac{\partial^2 \Gamma_k(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_k \right)^{-1}_{n\alpha,n'\alpha'} \left(\frac{\partial^2 \Gamma_k[\boldsymbol{\varphi}]}{\partial \varphi_{n',\alpha'} \partial \varphi_{n'',\alpha''}} + R_{k,n'-n''}^{\alpha'\alpha''} \right) = \delta_{n,n''} \delta_{\alpha\alpha''}. \quad (6)$$

In principle, Eq. (5) determines $\Gamma(\boldsymbol{\varphi})$, encompassing all the thermodynamic properties and correlations. Typically, Taylor series expansions, including vertex and derivative expansions, are employed. These expansions yield an approximate calculation of $\Gamma(\boldsymbol{\varphi})$ for a specific configuration of $\varphi_{n,\alpha}$. However, there is currently no established method to accurately and efficiently obtain $\Gamma(\boldsymbol{\varphi})$ for a broad domain of the $\varphi_{n,\alpha}$ space. Our goal is to propose a promising candidate for such a method.

A. PINNs for the Wetterich equation

Initially, calculations involving large N_{DOF} may appear computationally challenging, given their complexity, which grows exponentially when a grid is associated with each component $\varphi_{n,\alpha}$. However, our approach is rooted in the resilience of PINNs to this issue, given its grid-free nature and applicability to high-dimensional PDEs [22–28]. In PINNs, the solution is represented by a differentiable NN, eliminating the need for discretization in numerical differentiation. The NN is optimized to satisfy the PDE and the boundary conditions (BCs) using backpropagation. The optimization function may take the form $L = L_{\text{PDE}} + \lambda L_{\text{BC}}$ [19], where L_{PDE} (L_{BC}) reaches its minimum if, and only if, the NN satisfies the PDE

(BC) for any input, and λ is a positive hyperparameter to adjust the relative scale of the two terms.

The presence of both terms L_{PDE} and L_{BC} can pose challenges. For example, tuning λ for efficient optimization convergence can be required. However, in our case of the initial value problem, L_{BC} can be omitted with an appropriate choice of the ansatz on $\Gamma_k(\boldsymbol{\varphi})$ similar to Ref. [11]. We make such an ansatz based on the decomposition,

$$\Gamma_k(\boldsymbol{\varphi}) = S(\boldsymbol{\varphi}) + \Gamma_{\text{RG}}(l, \boldsymbol{\varphi}), \quad (7)$$

where $l = \ln(k_{\text{UV}}/k)$. Since the initial condition is $\Gamma_{k_{\text{UV}}}(\boldsymbol{\varphi}) = S(\boldsymbol{\varphi})$, the RG-induced part $\Gamma_{\text{RG}}(l, \boldsymbol{\varphi})$ satisfies $\Gamma_{\text{RG}}(0, \boldsymbol{\varphi}) = 0$. We further decompose $\Gamma_{\text{RG}}(l, \boldsymbol{\varphi})$ as

$$\Gamma_{\text{RG}}(l, \boldsymbol{\varphi}) = \gamma_{\text{free}}(l) + \gamma(l, \boldsymbol{\varphi}). \quad (8)$$

Here, $\gamma_{\text{free}}(l)$ represents the constant term originating from the free quadratic term $S_{\text{free}}(\boldsymbol{\varphi})$ of $S(\boldsymbol{\varphi})$. In other words, it is the solution when $\Gamma_k(\boldsymbol{\varphi})$ on the right-hand side of Eq. (5) is substituted by $S_{\text{free}}(\boldsymbol{\varphi})$. The remaining term $\gamma(l, \boldsymbol{\varphi})$ constitutes the nontrivial interaction-induced part, corresponding to the shift in the free energy. By imposing $\gamma(0, \boldsymbol{\varphi}) = 0$, we replace $\gamma(l, \boldsymbol{\varphi})$ with an NN. A conceivable choice is

$$\gamma(l, \boldsymbol{\varphi}) \approx \gamma(l, \boldsymbol{\varphi}; \boldsymbol{\theta}) = \text{NN}_{\boldsymbol{\theta}}(l, \boldsymbol{\varphi}) - \text{NN}_{\boldsymbol{\theta}}(0, \boldsymbol{\varphi}), \quad (9)$$

where $\text{NN}_{\boldsymbol{\theta}}(l, \boldsymbol{\varphi})$ is a differentiable NN with parameters $\boldsymbol{\theta}$.

A possible choice of $L = L_{\text{PDE}}$ to train $\gamma(l, \boldsymbol{\varphi}; \boldsymbol{\theta})$ is

$$L_{\boldsymbol{\theta}} = \mathbb{E}_{\substack{\boldsymbol{\varphi} \sim \mathcal{P}_{\boldsymbol{\varphi}} \\ l \sim \mathcal{P}_l}} \left[\left(\partial_l \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) - \frac{1}{2} \text{tr} \partial_l R_k \left(\frac{\partial^2 \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_k \right)^{-1} \right)^2 \right], \quad (10)$$

with $\Gamma_l^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) = S(\boldsymbol{\varphi}) + \gamma_{\text{free}}(l) + \gamma(l, \boldsymbol{\varphi}; \boldsymbol{\theta})$, we introduce probability distributions $\mathcal{P}_{\boldsymbol{\varphi}}$ and \mathcal{P}_l , defined for the $\boldsymbol{\varphi}$ space and $l \in [0, l_{\text{end}}]$, with $l_{\text{end}} = \ln(k_{\text{UV}}/k_{\text{IR}})$, respectively. In practice, the expectation value is approximately evaluated using a finite number of collocation points $\{(l^{(i)}, \boldsymbol{\varphi}^{(i)})\}_{i=1}^{N_{\text{col}}}$ sampled according to $\mathcal{P}_{\boldsymbol{\varphi}}$ and \mathcal{P}_l . Naively, if one is interested in a specific configuration $\boldsymbol{\varphi} = \boldsymbol{\varphi}_{\text{target}}$, then $\mathcal{P}_{\boldsymbol{\varphi}}$ should be chosen as to sample the neighborhoods of $\boldsymbol{\varphi}_{\text{target}}$ at high rates. A caveat is that, even in such a case, $\mathcal{P}_{\boldsymbol{\varphi}}$ should be sufficiently broad for learning $\boldsymbol{\varphi}$ derivatives, i.e., the $\boldsymbol{\varphi}$ dependence of $\gamma(l, \boldsymbol{\varphi}; \boldsymbol{\theta})$. We surmise that the breadth should have the scale of the fluctuation $\sqrt{\langle (\varphi_{n,\alpha} - \varphi_{\text{target},n,\alpha})^2 \rangle}$ for each direction $\varphi_{n,\alpha}$ to describe correlations.

The PINN-LFRG method described above is expected to offer advantages for complex structures, such as inhomogeneous states, over conventional FRG approximations. Specifically, the PINN-LFRG demonstrates improved scaling of computational complexity with respect to N_{DOF} compared to the vertex expansion, as illustrated in Appendix A.

III. NUMERICAL DEMONSTRATION IN THE ZERO-DIMENSIONAL $O(N)$ MODEL

To illustrate how PINN-LFRG works, we apply it to the zero-dimensional $O(N)$ model, which possesses an exact

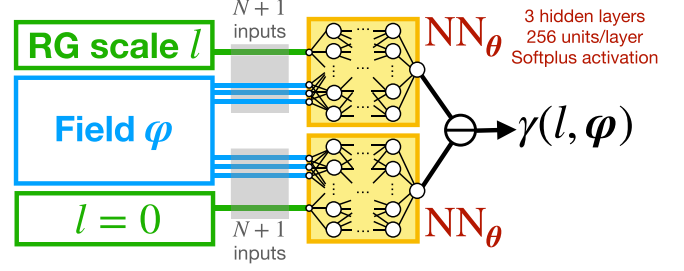


FIG. 1. The schematic picture of our NN architecture for the interaction-induced effective action $\gamma(l, \boldsymbol{\varphi})$.

solution. The action is given by

$$S(\boldsymbol{\varphi}) = \frac{1}{2} m^2 \boldsymbol{\varphi}^2 + \frac{g}{4!} (\boldsymbol{\varphi}^2)^2, \quad (11)$$

where $\boldsymbol{\varphi} = (\varphi_1, \dots, \varphi_N)$ represents an N -component scalar field, and m and g are the mass and coupling, respectively.

This model gives the total degree of freedom by $N_{\text{DOF}} = N_{\text{IDOF}} = N$ due to $V = 1$. We investigate the scalability with respect to N_{DOF} by increasing N [34]. We also assess accuracy by comparing the results of the interaction-induced effective action $\gamma(l, \boldsymbol{\varphi})$ and the RG-induced self-energy $\sigma_{\alpha}(l, \boldsymbol{\varphi}) = \partial^2 \gamma(l, \boldsymbol{\varphi}) / \partial \varphi_{\alpha}^2$ to those from the exact calculation, perturbative expansion up to the leading order, and large- N expansion up to $O(1)$, which are summarized in Appendix B. Note that $\tilde{g} = Ng/m^4$ is the dimensionless control parameter determining the perturbative region as $\tilde{g} \ll 1$ due to $\langle \boldsymbol{\varphi}^2 \rangle \sim N/m^2$ [35]. With the regulator $R_k^{\alpha\alpha'} = k_{\text{UV}}^2 e^{-2l} \delta_{\alpha\alpha'}$, our parameters satisfy $m^2/k_{\text{UV}}^2 = 0.01$ and $\tilde{g} \ll 100$, which validate the stationary point approximation at $l = 0$ and realize $\Gamma_{k_{\text{UV}}}(\boldsymbol{\varphi}) \approx S(\boldsymbol{\varphi})$. We set $l_{\text{end}} = 5$. Our ansatz on $\gamma(l, \boldsymbol{\varphi})$ is based on Eq. (9). Our $\text{NN}_{\boldsymbol{\theta}}(l, \boldsymbol{\varphi})$ is a fully connected NN composed of three hidden layers with 256 units per layer and the differentiable softplus activation function. We find that this choice of NN shows successful convergence in the pretraining described below. Figure 1 depicts a schematic of our proposed NN architecture for $\gamma(l, \boldsymbol{\varphi})$.

In our experience, the regularity of the matrix $\partial_{\boldsymbol{\varphi}}^2 \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) + R_k$, which is needed for the matrix inverse in Eq. (10), is frequently broken for randomly chosen $\boldsymbol{\theta}$. We find that pretraining with some approximate analytical results remedies this problem. Specifically, we use the result of the first-order perturbation $\gamma^{1\text{pt}}(l, \boldsymbol{\varphi})$, employing the following optimization function:

$$L_{\boldsymbol{\theta}}^{\text{pre}} = \mathbb{E}_{\substack{\boldsymbol{\varphi} \sim \mathcal{P}_{\boldsymbol{\varphi}} \\ l \sim \mathcal{P}_l}} [(\gamma(l, \boldsymbol{\varphi}; \boldsymbol{\theta}) - \gamma^{1\text{pt}}(l, \boldsymbol{\varphi}))^2]. \quad (12)$$

It should be noted that for stabilizing training on the Wetterich equation, the approximate solution does not always need high accuracy. In fact, when we use $\gamma^{1\text{pt}}(l, \boldsymbol{\varphi})$ during the pretraining phase, it significantly aids in successful training, even in nonperturbative cases, such as $\tilde{g} = 10$.

The Adam optimizer [36] is utilized to train the NN with Eqs. (10) and (12). All computations are executed on an NVIDIA A100 GPU with 40 GB of memory. Additional details about our training procedures are in Appendix C. The code for our numerical experiment is available at Ref. [37].

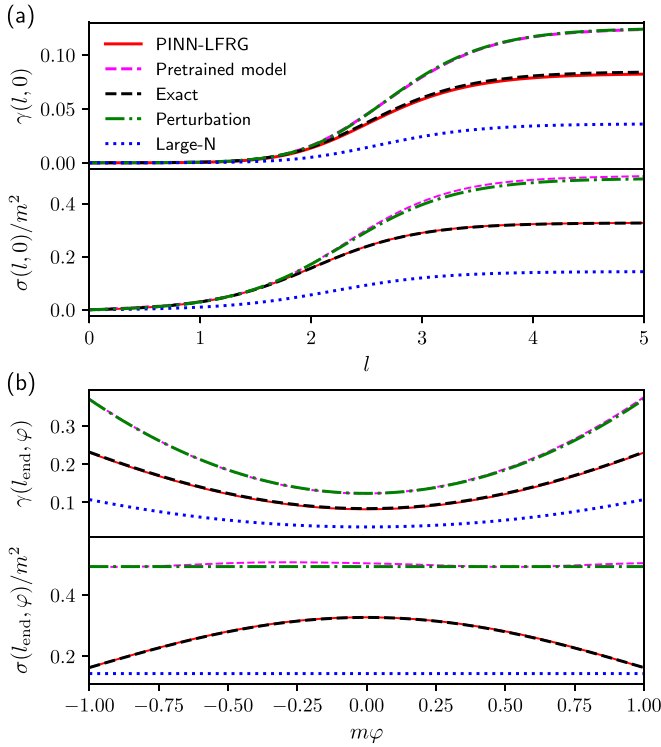


FIG. 2. $\gamma(l, \varphi)$ and $\sigma(l, \varphi)$ in the case of $N = 1$ and $\tilde{g} = 1$. (a) l dependence at $\varphi = 0$ and (b) φ dependence at $l = l_{\text{end}}$. Results of the PINN-LFRG (red-solid line), the pretrained model (magenta-dashed line), exact calculation (black-dashed line), perturbative expansion (green-dotted-dashed line), and large- N expansion (blue-dotted line) are represented.

We conducted computations for all the combinations of $N = 1, 10, 100$ and $\tilde{g} = 0.1, 1, 10$. In each case, the computational time for training is kept within 11 hours, ensuring the convergence of L_θ and physical quantities; see Appendix C for details. Figure 2 illustrates the results of $\gamma(l, \varphi)$ and $\sigma(l, \varphi)$ for $N = 1$ and $\tilde{g} = 1$. Specifically, the plot depicts the l dependence at the vacuum expectation value $\varphi = 0$ and the φ dependence at $l = l_{\text{end}}$. The results from exact calculations, perturbative and large- N expansions, and the model after the

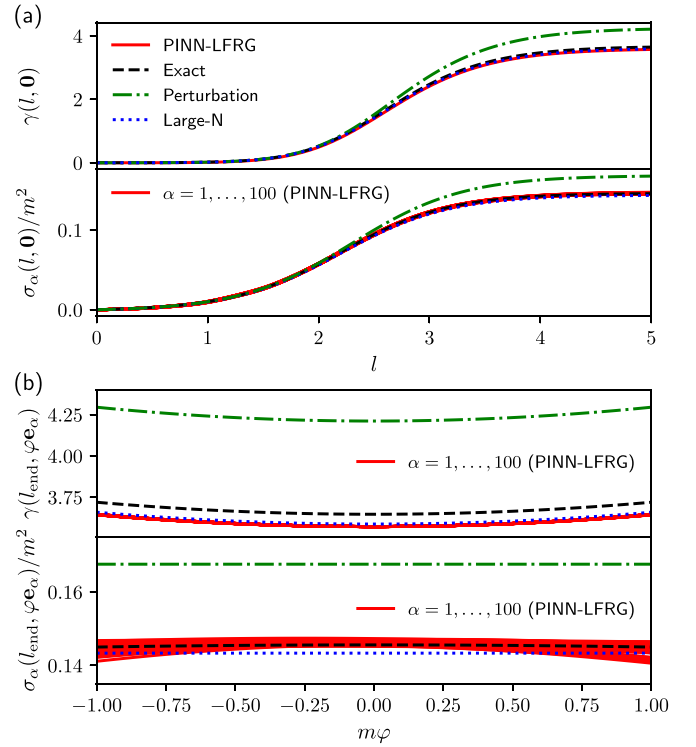


FIG. 3. $\gamma(l, \varphi_{\mathbf{e}_\alpha})$ and $\sigma_\alpha(l, \varphi_{\mathbf{e}_\alpha})$ in the case of $N = 100$ and $\tilde{g} = 1$. (a) l dependence at $\varphi = 0$ and (b) φ dependence at $l = l_{\text{end}}$. The outcomes of PINN-LFRG are presented as the $N = 100$ lines, except for $\gamma(l, \mathbf{0})$, where each line corresponds to a different choice of α .

pretraining are also presented. The perturbative and large- N expansion results show considerable deviations from the exact ones since both $\tilde{g} = 1$ and $1/N = 1$ are not small. Notably, the training of the Wetterich equation successfully shifts values from those obtained by the perturbation approach toward the exact results. In all instances, our PINN-LFRG approach exhibits higher accuracy than the perturbative method and large- N expansions. It is crucial to highlight that our approach provides solutions over a broad domain of φ , in contrast to the limitations of the vertex expansion method. Figure 3 illustrates the result for $N = 100$ and $\tilde{g} = 1$. With the exception of

TABLE I. Relative errors of $\gamma = \gamma(l_{\text{end}}, \mathbf{0})$ and $\sigma = \sigma(l_{\text{end}}, \mathbf{0})$ compared to the exact values in percentage. The results of the perturbation, large- N expansion, and PINN-LFRG are displayed. For PINN-LFRG, we present the ambiguity of the relative error of σ estimated from the standard deviation (denoted by $\Delta\sigma$). The minus sign indicates underestimation. The best values in each column are highlighted with bold font.

		1			10			100		
		0.1	1	10	0.1	1	10	0.1	1	10
Perturb. (%)	γ	6.2	47	275	2.1	19	129	1.7	15	110
	σ	7.6	51	228	2.3	19	109	1.7	15	92
Large- N (%)	γ	-65	-57	-40	-16	-14	-8.4	-1.9	-1.6	-0.95
	σ	-65	-56	-42	-16	-13	-8.2	-1.9	-1.5	-0.89
PINN-LFRG (%)	γ	-2.0	-2.2	-2.8	-1.9	-2.1	-2.3	-1.9	-2.0	-2.3
	σ	-0.17	0.12	0.76	0.16	0.46	0.42	-0.011	0.44	0.50
	$\Delta\sigma$	0	0	0	0.27	0.18	0.24	0.38	0.29	0.26

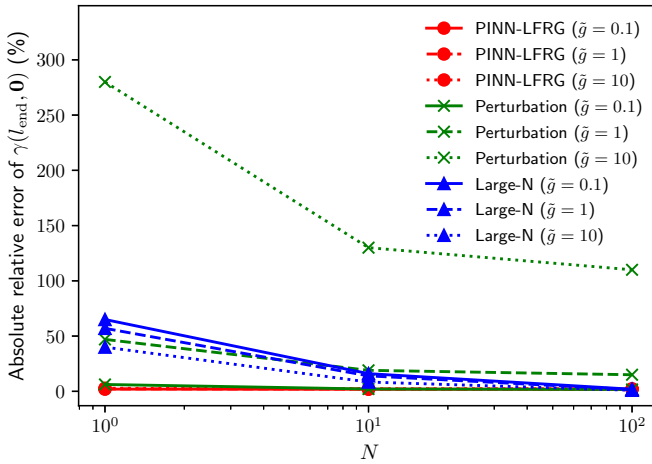


FIG. 4. Absolute values of the relative errors of $\gamma = \gamma(l_{\text{end}}, \mathbf{0})$ compared to the exact values as a function of N . All the cases of PINN-LFRG, perturbation, and large- N expansion with $\tilde{g} = 0.1, 1, 10$ are depicted.

$\gamma(l, \mathbf{0})$, our results are presented as the $N = 100$ lines corresponding to the N directions in the φ space. This includes all the $\alpha = 1, \dots, 100$ cases of $\gamma(l, \varphi \mathbf{e}_\alpha)$ and $\sigma_\alpha(l, \varphi \mathbf{e}_\alpha)$, where \mathbf{e}_α denotes the unit vector in the φ_α direction. The PINN-LFRG results for different α closely match, with differences being imperceptible in $\gamma(l_{\text{end}}, \varphi \mathbf{e}_\alpha)$ and $\sigma_\alpha(l, \mathbf{0})/m^2$. Even in $\sigma_\alpha(l_{\text{end}}, \varphi \mathbf{e}_\alpha)$, all the results from our approach are as close to the exact result as those of the large- N expansion, which is expected to be accurate for $N = 100$. These findings indicate that the NN automatically captures the $O(N)$ symmetry, enabling a simultaneously accurate solution for a domain of the high-dimensional configuration space.

Table I summarizes the relative errors of $\gamma(l_{\text{end}}, \mathbf{0})$ and $\sigma(l_{\text{end}}, \mathbf{0})$ compared to the exact values for all values of N and \tilde{g} . In the case of PINN-LFRG for $N > 1$, we determine $\sigma(l_{\text{end}}, \mathbf{0})$ by averaging $\sigma_\alpha(l_{\text{end}}, \mathbf{0})$ with respect to $\alpha = 1, \dots, N$, and we derive the standard deviation. To show the tendency, we plot the absolute values of the relative errors of $\gamma(l_{\text{end}}, \mathbf{0})$ compared to the exact values as a function of N in Fig. 4; almost the same tendency is seen for $\sigma(l_{\text{end}}, \mathbf{0})$. For all N and \tilde{g} values, the errors of PINN-LFRG are within 3% for $\gamma(l_{\text{end}}, \mathbf{0})$ and 1% for $\sigma(l_{\text{end}}, \mathbf{0})$ even if the standard deviations are taken into account. Particularly, PINN-LFRG is accurate even for the nonperturbative and small- N regions, where both the perturbative and large- N expansions break down. This suggests that the NN is a promising tool for providing accurate approximation independently of the existence of a small parameter.

IV. CONCLUSIONS

This study introduces PINN-LFRG as a framework for solving the Wetterich equation on a finite lattice. The approach demonstrates the ability to simultaneously derive an effective action for various field configurations. The proposed procedure involves representing the effective action through an NN

and optimizing it. The demonstration in the zero-dimensional $O(N)$ model indicates the feasibility of calculations involving a substantial number of degrees of freedom, around 10^2 or more, with NNs effectively approximating the effective action without the reliance on a small parameter.

Our analysis can be readily extended to models incorporating temporal and spatial degrees of freedom. An intriguing avenue for further exploration is the investigation of inhomogeneous states in scalar models, such as solitons, within our framework, building upon existing work on this topic [38–40]. Extending the approach to fermionic systems poses a substantial challenge since there is currently no efficient method for constructing NNs for Grassmann variables. However, one could apply our approach to fermionic systems by introducing bosonic auxiliary fields, for example. An exciting application in this direction is the adaptation of our method to density functional theory [41–43], a standard tool for analyzing many-body systems. This has been extended to apply to lattice models, such as the Hubbard model [44]. We anticipate that our approach holds promise for the FRG-based formalism of density functional theory, a framework that has seen recent developments [45–57].

ACKNOWLEDGMENTS

The author thanks Taiki Miyagawa for carefully reading the manuscript and making valuable comments. The author also thanks Gergely Fejős, Tetsuo Hatsuda, Tomoya Naito, Osamu Sugino, and Junji Yamamoto for valuable discussions. The RIKEN Special Postdoctoral Researchers Program supported the author.

APPENDIX A: COMPARISON OF COMPLEXITY

As delineated in the main text, the PINN-LFRG approach is anticipated to offer advantages for analyzing complex structures, such as inhomogeneous states, when compared to conventional methods, including the vertex expansion. We discuss this advantage from the perspective of computational complexity. In the context of the lattice setup described in the main text, we concentrate on the scaling relative to N_{DOF} .

The derivative expansion is ineffective for systems with large field gradients, which may be described by the vertex expansion, albeit at a significantly greater computational cost than in homogeneous cases. The computational complexity for the i th-order vertex expansion scales as $O(N_{\text{DOF}}^{i+2})$ or $O(N_{\text{DOF}}^{i+3})$, derived as follows: The i th-order vertex expansion around an inhomogeneous field profile $\varphi = \varphi_{\text{inhom}}$ produces the flow equations for

$$\Gamma_{k;n_1, \dots, n_j}^{(j)} = \frac{\partial^j \Gamma_k(\varphi_{\text{inhom}})}{\partial \varphi_{n_1} \dots \partial \varphi_{n_j}} \quad (j = 0, \dots, i), \quad (\text{A1})$$

where n_1, \dots, n_j are site indices. For simplicity, we have omitted the internal degrees of freedom introduced in the main text. The computational bottleneck is the calculation of

$\Gamma_{k;m_1,\dots,m_i}^{(i)}$, governed by the following flow equation:

$$\partial_k \Gamma_{k;m_1,\dots,m_i}^{(i)} = \frac{1}{2} \text{tr} \left[\text{matrix products of } \partial_k R_k, [\Gamma_k^{(2)} + R_k]^{-1}, \text{ and } [\Gamma_{k;m_1,\dots,m_{j-2}}^{(j)}] (j = 3, \dots, i+2) \right], \quad (\text{A2})$$

where $[\Gamma_{k;m_1,\dots,m_{j-2}}^{(j)}]$ is an $N_{\text{DOF}} \times N_{\text{DOF}}$ matrix defined by

$$[\Gamma_{k;m_1,\dots,m_{j-2}}^{(j)}]_{n,n'} = \Gamma_{k;m_1,\dots,m_{j-2},n,n'}^{(j)} \quad (\text{A3})$$

and each m_l ($l = 1, \dots, j-2$) represents any of n_1, n_2, \dots, n_i . The computational complexity of evaluating the trace is $O(N_{\text{DOF}}^2)$ or $O(N_{\text{DOF}}^3)$ depending on the algorithm [58]. Since the flow equation is calculated for all combinations of (n_1, \dots, n_i) , the complexity escalates to $O(N_{\text{DOF}}^{i+2})$ or $O(N_{\text{DOF}}^{i+3})$. In contrast, in homogeneous cases, the complexity is only $O(N_{\text{DOF}}^i)$, as the number of independent components of $\Gamma_{k;m_1,\dots,m_i}^{(i)}$ is $O(N_{\text{DOF}}^{i-1})$ and the trace evaluation requires merely $O(N_{\text{DOF}})$ due to translational symmetry.

The complexity of the PINN-LFRG model is assessed as either $O(N_{\text{DOF}}^2)$ or $O(N_{\text{DOF}}^3)$, depending on the specific algorithm employed. This complexity arises primarily due to the evaluation of the trace in Eq. (10). These findings indicate that for large values of N_{DOF} , the computational demands of the PINN-LFRG are potentially less than those of the vertex expansion, even at the second order, where the complexity reaches $O(N_{\text{DOF}}^4)$ or $O(N_{\text{DOF}}^5)$. For a more comprehensive comparison, further discussion of the influence of additional factors, such as network size, on the total numerical effort would be desirable.

APPENDIX B: EXACT CALCULATION, PERTURBATIVE EXPANSION, AND LARGE- N EXPANSION IN THE ZERO-DIMENSIONAL $O(N)$ MODEL

We summarize the numerical procedure for the exact calculation and the results of the perturbative and large- N expansions for the interaction-induced effective action $\gamma(l, \boldsymbol{\varphi}) = \gamma(l, \varphi)$ and the RG-induced self-energy $\sigma(l, \varphi) = \partial_\varphi^2 \gamma(l, \varphi)$ in the zero-dimensional $O(N)$ model. We use the form of the regulator $R_k^{\alpha\alpha'} = r_k \delta_{\alpha\alpha'}$ as in the main text.

The exact results are obtained by directly evaluating the path integral of the partition function,

$$Z_l(\mathbf{J}) = \int d\boldsymbol{\varphi} e^{-\frac{1}{2} m_l^2 \varphi^2 - \frac{g}{4!} (\varphi^2)^2 + \mathbf{J} \cdot \boldsymbol{\varphi}}, \quad (\text{B1})$$

where $m_l^2 = m^2 + r_k$ represents the regulated mass squared. Due to the presence of an $O(N-1)$ symmetry in the $\boldsymbol{\varphi}$ space perpendicular to \mathbf{J} , the integral can be simplified as follows:

$$\begin{aligned} Z_l(J) &= \Omega_{N-1} \int_{-\infty}^{\infty} d\varphi e^{-\frac{1}{2} m_l^2 \varphi^2 - \frac{g}{4!} \varphi^4 + J\varphi} Q_{N-2,l}(\varphi^2), \quad (\text{B2}) \\ Q_{N-2,l}(\varphi^2) &= \int_0^{\infty} dx x^{N-2} e^{-\frac{1}{2} (m_l^2 + \frac{g}{6} \varphi^2) x^2 - \frac{g}{4!} x^4} \\ &= \frac{1}{2} \left(\frac{6}{g} \right)^{\frac{N-1}{4}} \Gamma\left(\frac{N-1}{2} \right) \\ &\quad \times U\left(\frac{N-1}{4}, \frac{1}{2}, \frac{3}{2g} \left(m_l^2 + \frac{g}{6} \varphi^2 \right)^2 \right), \quad (\text{B3}) \end{aligned}$$

where we have introduced $J = \|\mathbf{J}\|$, the surface area of the unit $(N-1)$ -sphere $\Omega_N = 2\pi^{N/2}/\Gamma(N/2)$, the gamma function $\Gamma(x)$, and the Tricomi's confluent hypergeometric function $U(a, b, z)$. Let $J = J_{\text{sup},l}(\varphi)$ be an external field realizing $\langle \varphi \rangle = \varphi$, i.e., the solution of

$$\begin{aligned} \varphi &= \frac{\partial \ln Z_l}{\partial J} (J_{\text{sup},l}(\varphi)) \\ &= \frac{\int_{-\infty}^{\infty} dx x e^{-\frac{1}{2} m_l^2 x^2 - \frac{g}{4!} x^4 + J_{\text{sup},l}(\varphi)x} Q_{N-2,l}(x^2)}{\int_{-\infty}^{\infty} dx e^{-\frac{1}{2} m_l^2 x^2 - \frac{g}{4!} x^4 + J_{\text{sup},l}(\varphi)x} Q_{N-2,l}(x^2)}. \quad (\text{B4}) \end{aligned}$$

With this external field, the effective action and the self-energy are given by

$$\Gamma(l, \varphi) = J_{\text{sup},k}(\varphi) \varphi - \ln Z_k(J_{\text{sup},k}(\varphi)) - \frac{1}{2} r_k \varphi^2, \quad (\text{B5})$$

$$\Sigma(l, \varphi) = \partial_\varphi^2 \Gamma(l, \varphi) - m^2 = \frac{1}{\langle \varphi^2 \rangle - \varphi^2} - m_l^2, \quad (\text{B6})$$

where correlation function $\langle \varphi^2 \rangle - \varphi^2$ is evaluated by

$$\begin{aligned} \langle \varphi^2 \rangle - \varphi^2 &= \frac{\partial^2 \ln Z_l}{\partial J^2} (J_{\text{sup},l}(\varphi)) \\ &= \frac{\int_{-\infty}^{\infty} dx e^{-\frac{1}{2} m_l^2 x^2 - \frac{g}{4!} x^4 + J_{\text{sup},l}(\varphi)x} (x - \varphi)^2 Q_{N-2,l}(x^2)}{\int_{-\infty}^{\infty} dx e^{-\frac{1}{2} m_l^2 x^2 - \frac{g}{4!} x^4 + J_{\text{sup},l}(\varphi)x} Q_{N-2,l}(x^2)}. \quad (\text{B7}) \end{aligned}$$

With these $\Gamma(l, \varphi)$ and $\Sigma(l, \varphi)$, we obtain

$$\gamma(l, \varphi) = \Gamma(l, \varphi) - \Gamma(0, \varphi) - \gamma_{\text{free}}(l), \quad (\text{B8})$$

$$\sigma(l, \varphi) = \Sigma(l, \varphi) - \Sigma(0, \varphi), \quad (\text{B9})$$

where $\gamma_{\text{free}}(l) = (N/2) \ln(m_l^2/m_0^2)$ is the solution of

$$\partial_l \gamma_{\text{free}}(l) = \frac{1}{2} \sum_{\alpha=1}^N \partial_l r_k \left(\frac{\partial^2 S_{\text{free}}(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + r_k \right)_{\alpha\alpha}^{-1}, \quad \gamma_{\text{free}}(0) = 0, \quad (\text{B10})$$

with $S_{\text{free}}(\boldsymbol{\varphi}) = m^2 \boldsymbol{\varphi}^2 / 2$. We numerically solve Eq. (B4) for $J_{\text{sup},l}(\varphi)$ by use of `scipy.optimize.fsolve` in SciPy. With this $J_{\text{sup},l}(\varphi)$, we numerically evaluate Eqs. (B2) and (B7) to obtain $\gamma(l, \varphi)$ and $\sigma(l, \varphi)$. The integrals in Eqs. (B2), (B4), and (B7) are evaluated using the Gauss quadrature method implemented as `scipy.integrate.quad` in SciPy.

The perturbative and large- N expansion results are obtained from Ref. [35]. By substituting the regulated mass squared m_l^2 into these expressions, the results at the scale l for the effective action and self-energy up to the leading order are given by

$$\begin{aligned} \Gamma(l, \varphi) &= S(\boldsymbol{\varphi}) + N \frac{1 + 2N^{-1}}{24} \tilde{g}_l + \frac{1 + 2N^{-1}}{12} \tilde{g}_l m_l^2 \varphi^2 \\ &\quad + \gamma_{\text{free}}(l) + O(\tilde{g}_l^2), \quad (\text{B11}) \end{aligned}$$

$$\Sigma(l, \varphi) = \partial_\varphi^2 S(\boldsymbol{\varphi}) - m^2 + \frac{1 + 2N^{-1}}{6} \tilde{g}_l m_l^2 + O(\tilde{g}_l^2). \quad (\text{B12})$$

TABLE II. Computational times for the pretraining and the training of the Wetterich equation (labeled by Wetterich) on an NVIDIA A100 GPU. The results are consistent for $\tilde{g} = 0.1, 1, 10$.

N	1	10	100
Pretraining	4 m	4 m	6 m
Wetterich	6 h	7 h	11 h

Here, the dimensionless quantity $\tilde{g}_l = Ng/m_l^4$ is employed as the expansion parameter instead of g . The result of the large- N expansion up to $O(1)$ is expressed as follows:

$$\Gamma(l, \varphi) = S(\varphi) + N \left(\frac{z_l - 1}{4} - \frac{1}{2} \ln z_l \right) + \frac{1}{2} \ln(2 - z_l) + \frac{1}{2} (z_l^{-1} - 1) m_l^2 \varphi^2 + \gamma_{\text{free}}(l) + O(1/N), \quad (\text{B13})$$

$$\Sigma(l, \varphi) = \partial_\varphi^2 S(\varphi) - m^2 + (z_l^{-1} - 1) m_l^2 + O(1/N), \quad (\text{B14})$$

$$L_\theta \approx \frac{1}{N_{\text{col}}} \sum_{n=1}^{N_{\text{col}}} \left[\left(\partial_l \gamma(l^{(n)}, \varphi^{(n)}; \theta) + \partial_l \gamma_{\text{free}}(l^{(n)}) - \frac{1}{2} \sum_{\alpha, \alpha'} \partial_l R_{k^{(n)}}^{\alpha\alpha'} \left(\frac{\partial^2 S(\varphi^{(n)})}{\partial \varphi \partial \varphi} + \frac{\partial^2 \gamma(l^{(n)}, \varphi^{(n)}; \theta)}{\partial \varphi \partial \varphi} + R_{k^{(n)}} \right)^{-1} \right)_{\alpha', \alpha} \right]^2, \quad (\text{C1})$$

$$L_\theta^{\text{pre}} \approx \frac{1}{N_{\text{col}}} \sum_{n=1}^{N_{\text{col}}} [(\gamma(l^{(n)}, \varphi^{(n)}; \theta) - \gamma^{\text{1pt}}(l^{(n)}, \varphi^{(n)}))^2], \quad (\text{C2})$$

where $l^{(n)} (= \ln(k_{\text{UV}}/k^{(n)}))$ and $\varphi^{(n)}$ are randomly sampled following the probability distributions \mathcal{P}_φ and \mathcal{P}_l . For \mathcal{P}_l , we adopt a uniform distribution within the interval $[0, l_{\text{end}}]$. To sample the neighborhoods of $\varphi = \mathbf{0}$, representing the vacuum expectation value, we define \mathcal{P}_φ such that the direction $\hat{n} = \varphi/\|\varphi\|$ is uniformly sampled. The norm $\|\varphi\|$ is sampled following a normal distribution $\mathcal{N}(0, N/m^2)$ without the sign, where the variance N/m^2 corresponds to the order of $\langle \varphi^2 \rangle$. It is noteworthy that the efficiently sampling neighborhoods

with

$$z_l = \frac{2}{1 + \sqrt{1 + \frac{2}{3} \tilde{g}_l}}. \quad (\text{B15})$$

With these $\Gamma(l, \varphi)$ and $\Sigma(l, \varphi)$, we obtain $\gamma(l, \varphi)$ and $\sigma(l, \varphi)$ from Eqs. (B8) and (B9).

APPENDIX C: DETAILS ABOUT TRAINING

We provide some details about the training and information about the convergence of our results. We optimize our NN to minimize L_θ (L_θ^{pre}) in the main text for the training of the Wetterich equation (the pretraining). The expectations of these equations are approximately evaluated on a finite number of collocation points,

of $\varphi = \mathbf{0}$ for large N is challenging if \mathcal{P}_φ is set to an N -dimensional normal distribution $\mathcal{N}(\mathbf{0}, m^{-2} \mathbf{1})$ or a uniform distribution in an N -dimensional box due to the curse of dimensionality. Specifically, we choose $N_{\text{col}} = 500$ collocation points, which are refreshed each time the optimization functions are assessed.

For the numerical implementation, we employ Pytorch. The learning rate for the Adam optimizer is initially set to 10^{-4} and exponentially decays with a factor of 0.99999.

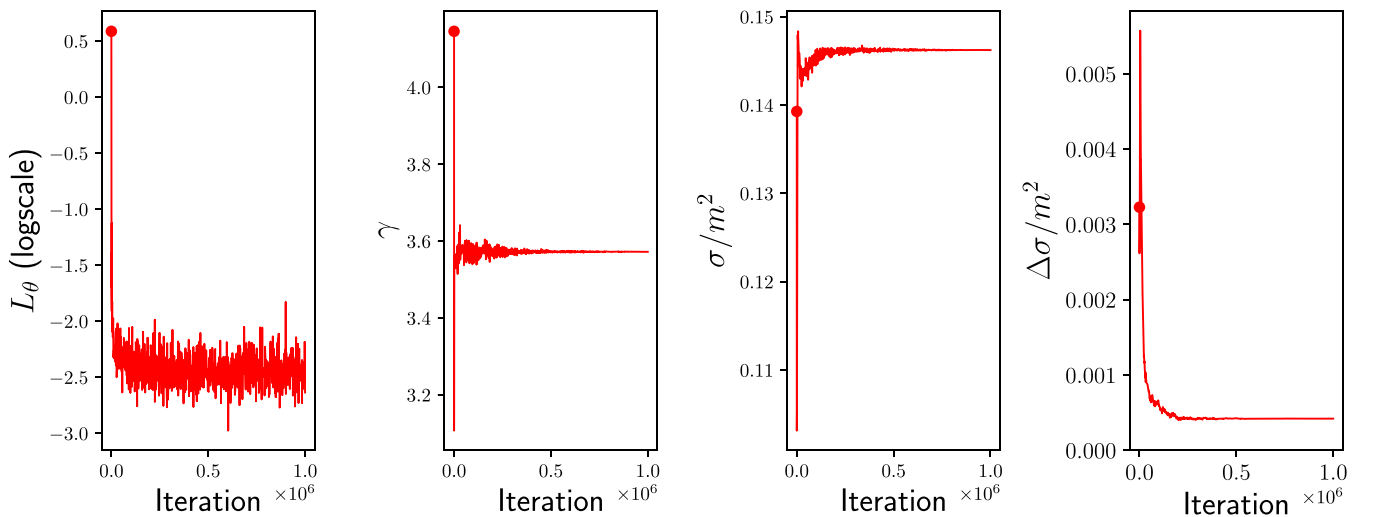


FIG. 5. Learning curve and the histories of $\gamma = \gamma(l_{\text{end}}, \mathbf{0})$, the average σ/m^2 , and the standard deviation $\Delta\sigma/m^2$ of $\sigma_\alpha(l, \mathbf{0})/m^2$ with respect to α in the case of $N = 100$ and $\tilde{g} = 1$. The red dots indicate the initial values.

TABLE III. Relative errors of $\gamma(l_{\text{end}}, 0)$ compared to the exact solutions for different numbers of the hidden layers and the units per layer with $N = 1$ and $\tilde{g} = 1$. The minus sign indicates underestimation.

Relative error of γ (%)		Number of units per layer		
		64	128	256
	1	-3.2	-0.23	-3.7
Number of hidden layers	2	-2.2	-2.2	-2.2
	3	-2.2	-2.2	-2.2

The learning rate is fixed at 10^{-3} in the pretraining phase. The Xavier initialization [60] is used for this pretraining. It is worth noting that the computational cost of evaluating the matrix inverse in L_θ is substantial. To facilitate implementation, we directly compute the inversion using the `torch.linalg.inv` function in Pytorch. Efficiency enhancement, potentially utilizing alternative algorithms such as the Hutchinson trace estimator [59], is reserved for future study.

The training process involves 10^6 iterations for the Wetterich equation and 10^5 for the pretraining. Table II provides an overview of the computational time required. With this iteration count, we observe the convergence of L_θ and physical quantities. As illustrated in Fig. 5, we present a learning curve along with the histories of $\gamma = \gamma(l_{\text{end}}, \mathbf{0})$ and the average σ/m^2 , as well as the standard deviation $\Delta\sigma/m^2$ of $\sigma_\alpha(l, \mathbf{0})/m^2$ with respect to α for the case of $N = 100$ and $\tilde{g} = 1$. In

TABLE IV. Relative errors of $\sigma = \sigma(l_{\text{end}}, 0)$ compared to the exact solutions for different numbers of the hidden layers and the units per layer with $N = 1$ and $\tilde{g} = 1$. The minus sign indicates underestimation.

Relative error of σ (%)		Number of units per layer		
		64	128	256
	1	4.1	-2.1	4.2
Number of hidden layers	2	0.68	0.36	0.62
	3	0.16	0.26	0.12

the initial iterations, L_θ rapidly decreases, and the physical quantities approach convergence quickly. Subsequently, as the learning rate decays, physical quantities gradually converge. The diminishing $\Delta\sigma/m^2$ over iterations indicates successfully reproducing the $O(N)$ symmetry during training. Finally, we explore how the NN's size influences our results. Tables III and IV detail the errors in measuring $\gamma(l_{\text{end}}, 0)$ and $\sigma(l_{\text{end}}, 0)$, respectively, each considering different numbers of hidden layers and units per layer, with settings of $N = 1$ and $\tilde{g} = 1$. From these, it is clear that more hidden layers lead to better accuracy. However, while this improvement tends to level off for $\gamma(l_{\text{end}}, 0)$, the accuracy for $\sigma(l_{\text{end}}, 0)$ continues to benefit from additional layers. This difference might be unique to the architecture we've used. We observed similar trends under other settings for N and \tilde{g} .

-
- [1] F. J. Wegner and A. Houghton, Renormalization group equation for critical phenomena, *Phys. Rev. A* **8**, 401 (1973).
- [2] K. G. Wilson and J. Kogut, The renormalization group and the ϵ expansion, *Phys. Rep.* **12**, 75 (1974).
- [3] J. Polchinski, Renormalization and effective Lagrangians, *Nucl. Phys. B* **231**, 269 (1984).
- [4] C. Wetterich, Exact evolution equation for the effective potential, *Phys. Lett. B* **301**, 90 (1993).
- [5] J. Berges, N. Tetradis, and C. Wetterich, Non-perturbative renormalization flow in quantum field theory and statistical physics, *Phys. Rep.* **363**, 223 (2002)
- [6] H. Gies, Introduction to the functional RG and applications to gauge theories, *Lect. Notes Phys.* **852**, 287 (2012).
- [7] W. Metzner, M. Salmhofer, C. Honerkamp, V. Meden, and K. Schönhammer, Functional renormalization group approach to correlated fermion systems, *Rev. Mod. Phys.* **84**, 299 (2012).
- [8] N. Dupuis, L. Canet, A. Eichhorn, W. Metzner, J. M. Pawłowski, M. Tissier, and N. Wschebor, The nonperturbative functional renormalization group and its applications, *Phys. Rep.* **910**, 1 (2021).
- [9] D. C. Psychogios and L. H. Ungar, A hybrid neural network-first principles approach to process modeling, *AIChE J.* **38**, 1499 (1992).
- [10] M. W. M. G. Dissanayake and N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, *Commun. Numer. Meth. Eng.* **10**, 195 (1994).
- [11] I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Networks* **9**, 987 (1998).
- [12] W. E, J. Han, and A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, *Commun. Math. Stat.* **5**, 349 (2017).
- [13] J. Han, A. Jentzen, and W. E, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci. USA* **115**, 8505 (2018).
- [14] M. Raissi, Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations, in *Peter Carr Gedenkschrift* (World Scientific, Singapore, 2023), Chap. 18, pp. 637–655.
- [15] W. E and B. Yu, The Deep Ritz method: A Deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.* **6**, 1 (2018).
- [16] Y. Khoo, J. Lu, and L. Ying, Solving for high-dimensional committor functions using artificial neural networks, *Res. Math. Sci.* **6**, 1 (2018).
- [17] J. Sirignano and K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.* **375**, 1339 (2018).
- [18] C. Beck, W. E, and A. Jentzen, Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations, *J. Nonlinear Sci.* **29**, 1563 (2019).

- [19] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [20] M. Fujii, A. Takahashi, and M. Takahashi, Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs, *Asia-Pacific Financ. Markets* **26**, 391 (2019).
- [21] M. A. Nabian and H. Meidani, A deep learning solution approach for high-dimensional random differential equations, *Probabilistic Eng. Mech.* **57**, 14 (2019).
- [22] L. Guo, H. Wu, X. Yu, and T. Zhou, Monte Carlo fPINNs: Deep learning method for forward and inverse problems involving high dimensional fractional partial differential equations, *Comput. Methods Appl. Mech. Eng.* **400**, 115523 (2022).
- [23] D. He, S. Li, W. Shi, X. Gao, J. Zhang, J. Bian, L. Wang, and T.-Y. Liu, Learning physics-informed neural networks without stacked back-propagation, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2023), pp. 3034–3047.
- [24] J. Cen and Q. Zou, Deep finite volume method for high-dimensional partial differential equations, [arXiv:2305.06863](https://arxiv.org/abs/2305.06863) (2023).
- [25] K. Tang, X. Wan, and C. Yang, DAS-PINNs: A deep adaptive sampling method for solving high-dimensional partial differential equations, *J. Comput. Phys.* **476**, 111868 (2023).
- [26] Z. Hu, K. Shukla, G. E. Karniadakis, and K. Kawaguchi, Tackling the curse of dimensionality with physics-informed neural networks, *Neural Networks* **176**, 106369 (2024).
- [27] Z. Hu, Z. Yang, Y. Wang, G. E. Karniadakis, and K. Kawaguchi, Bias-variance trade-off in physics-informed neural networks with randomized smoothing for high-dimensional PDEs, [arXiv:2311.15283](https://arxiv.org/abs/2311.15283) (2023).
- [28] Z. Hu, Z. Shi, G. E. Karniadakis, and K. Kawaguchi, Hutchinson trace estimation for high-dimensional and high-order physics-informed neural networks, *Comp. Meth. Appl. Mech. Eng.* **424**, 116883 (2024).
- [29] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.* **2**, 303 (1989).
- [30] K. Hornik, Maxwell Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* **2**, 359 (1989).
- [31] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inf. Theory* **39**, 930 (1993).
- [32] D. Di Sante, M. Medvidović, A. Toschi, G. Sangiovanni, C. Franchini, A. M. Sengupta, and A. J. Millis, Deep learning the functional renormalization group, *Phys. Rev. Lett.* **129**, 136402 (2022).
- [33] N. Dupuis and K. Sengupta, Non-perturbative renormalization-group approach to lattice models, *Eur. Phys. J. B* **66**, 271 (2008).
- [34] The Wetterich equation for this model can be reduced to a two-dimensional PDE with variables l and $\rho = \varphi^2/2$, but we do not use this reduction to investigate scalability.
- [35] J. Keitel and L. Bartosch, The zero-dimensional $O(N)$ vector model as a benchmark for perturbation theory, the large- N expansion and the functional renormalization group, *J. Phys. A: Math. Theor.* **45**, 105401 (2012).
- [36] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2015).
- [37] T. Yokota, <https://github.com/TakeruYokota/PINNLFRG.git>.
- [38] S. Bolognesi, K. Konishi, and K. Ohashi, Large- N $\mathbb{C}P^{N-1}$ sigma model on a finite interval, *J. High Energy Phys.* **10** (2016) 073.
- [39] M. Nitta and R. Yoshii, Self-consistent large- N analytical solutions of inhomogeneous condensates in quantum $\mathbb{C}P^{N-1}$ model, *J. High Energy Phys.* **12** (2017) 145.
- [40] A. Gorsky, A. Pikalov, and A. Vainshtein, On instability of ground states in 2D $\mathbb{C}P^{N-1}$ and \mathbb{O}^N models at large N , [arXiv:1811.05449](https://arxiv.org/abs/1811.05449) (2018).
- [41] P. Hohenberg and W. Kohn, Inhomogeneous electron gas, *Phys. Rev.* **136**, B864 (1964).
- [42] W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev.* **140**, A1133 (1965).
- [43] W. Kohn, Nobel lecture: Electronic structure of matter—Wave functions and density functionals, *Rev. Mod. Phys.* **71**, 1253 (1999).
- [44] D. J. Carrascal, J. Ferrer, J. C. Smith, and K. Burke, The Hubbard dimer: A density functional case study of a many-body problem, *J. Phys.: Condens. Matter* **27**, 393001 (2015).
- [45] J. Polonyi and K. Sailer, Effective actions and the density functional theory, *Phys. Rev. B* **66**, 155113 (2002).
- [46] A. Schwenk and J. Polonyi, Towards density functional calculations from nuclear forces, in *Proceedings of 32nd International Workshop on Gross Properties of Nuclei and Nuclear Excitation: Probing Nuclei and Nucleons with Electrons and Photons* (Hirschegg, Austria, 2004), pp. 273–282.
- [47] S. Kemler and J. Braun, Towards a renormalization group approach to density functional theory—general formalism and case studies, *J. Phys. G* **40**, 085105 (2013).
- [48] J. F. Rentrop, S. G. Jakobs, and V. Meden, Two-particle irreducible functional renormalization group schemes—A comparative study, *J. Phys. A* **48**, 145002 (2015).
- [49] S. Kemler, M. Pospiech, and J. Braun, Formation of selfbound states in a one-dimensional nuclear model—A renormalization group based density functional study, *J. Phys. G* **44**, 015101 (2017).
- [50] H. Liang, Y. Niu, and T. Hatsuda, Functional renormalization group and Kohn-Sham scheme in density functional theory, *Phys. Lett. B* **779**, 436 (2018).
- [51] T. Yokota, K. Yoshida, and T. Kunihiro, Functional renormalization-group calculation of the equation of state of one-dimensional uniform matter inspired by the Hohenberg-Kohn theorem, *Phys. Rev. C* **99**, 024302 (2019).
- [52] T. Yokota, K. Yoshida, and T. Kunihiro, *Ab initio* description of excited states of 1D uniform matter with the Hohenberg-Kohn-theorem-inspired functional-renormalization-group method, *Prog. Theor. Exp. Phys.* **2019**, 011D01 (2019).
- [53] T. Yokota and T. Naito, Functional-renormalization-group aided density functional analysis for the correlation energy of the two-dimensional homogeneous electron gas, *Phys. Rev. B* **99**, 115106 (2019).
- [54] T. Yokota, H. Kasuya, K. Yoshida, and T. Kunihiro, Microscopic derivation of density functional theory for superfluid systems based on effective action formalism, *Prog. Theor. Exp. Phys.* **2021**, 013A03 (2021).

- [55] T. Yokota and T. Naito, *Ab initio* construction of the energy density functional for electron systems with the functional-renormalization-group-aided density functional theory, *Phys. Rev. Res.* **3**, L012015 (2021).
- [56] T. Yokota, J. Haruyama, and O. Sugino, Functional-renormalization-group approach to classical liquids with short-range repulsion: A scheme without repulsive reference system, *Phys. Rev. E* **104**, 014124 (2021).
- [57] T. Yokota and T. Naito, Construction of energy density functional for arbitrary spin polarization using functional renormalization group, *Phys. Rev. B* **105**, 035105 (2022).
- [58] The complexity scales as $O(N_{\text{DOF}}^3)$ for direct evaluation of the matrix product and can be reduced to $O(N_{\text{DOF}}^2)$ using methods such as the Hutchinson trace estimator [59].
- [59] M. F. Hutchinson, A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines, *Commun. Stat. - Simul. Comput.* **19**, 433 (1990).
- [60] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, edited by Y.W. Teh and M. Titterton, Vol. 9 of Proceedings of Machine Learning Research (Chia Laguna Resort, Sardinia, Italy, 2010), pp. 249–256.