Learning high-order geometric flow based on the level set method

Chun Li · Yunyun Yang · Hui Liang · Boying Wu

Received: 3 June 2021 / Accepted: 1 November 2021 / Published online: 23 January 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2021

Abstract Recently, the development of deep learning has accomplished unbelievable success in many fields, especially in scientific computational fields. And almost all computational problems and physical phenomena can be described by partial differential equations. In this work, we proposed two potential high-order geometric flows. Motivation by the physical-information neural networks and the traditional level set method (LSM), we have integrated deep neural networks and the traditional level set method to make the proposed method more robust and efficient. Also, to test the sensitivity of the system to different input data, we set up three sets of initial conditions to test the model. Furthermore, numerical experiments on different input data are implemented to demonstrate the effectiveness and superiority of the proposed models compared to the state-of-the-art approach.

Keywords High-order geometric flow · Physics-constrained learning · Deep learning · Level set method

1 Introduction

Recently, the development of artificial intelligence (AI) has received comprehensive attention and assistance through the breakthrough of deep learning (DL) technology. DL has accomplished unbelievable success in many fields, especially in scientific computational fields, including computer vision, medical imaging, and control problem [1–6].

Fortunately, almost all computational problems and physical phenomena can be described by differential equations, especially geometric partial differential equations (GPDEs). Such multi-phase flows play an increasing role in several scientific and engineering applications [7–9]. Also, GPDEs has essential application in computer vision, such as Liang et al. [10] used PDE for facial image analysis. Moreover, Chen et al. [11] developed geodesic paths for image segmentation. In addition, PDEs are widely used in medical image analysis. For example, Chen and Amini [12] developed the geometric deformable models for MRA images. Also, Salinas and Fernández [13] used PDE
approaches for medical image enhancement. Furthermore, Karasew et al. [14] proposed an interactive PDE control of active contours for medical image segmentation. Besides, PDE has important applications in the field of control. Such as Birn et al. [15] developed a PDE approach for real-time detection and isolation. Moreover, a robust cooperative output regulation for a network of parabolic PDE systems was proposed by Deutscher [16]. Furthermore, Song et al. [17] used the PDE systems for fuzzy event-triggered control. More references about the applications of PDE in control are shown [18,19].

The PDEs can be solved analytically and numerically. However, so far, the analytical solutions of many PDEs have not been solved. Therefore, in terms of PDEs that have not been analytically solved, we can only understand the physical phenomenon described by them from the perspective of the numerical solution. Usually, the numerical approaches are used to discretized the solution domain and construct algebraic equations, after that, they are solved analytically or iterative. The traditional numerical methods include the finite volume method (FVM), finite difference method (FDM), finite element method (FEM), and so on. The computational cost of the equations becomes extremely expensive when the number of equations increases. Moreover, the GPDEs are generally defined on the surface (manifold), which can fall into the curse of dimensionality. Furthermore, GPDEs solutions can be distinctly different, and there is no general approach that applies to all kinds of GPDEs.

Happily, since the universal approximation theorem (UAT), which is the fundamental theoretical basis of DL, and it opens another door for the numerical solution of GPDEs. According to the UAT, the complex or even dynamical GPDEs can be approximated via a DNN. To find the solutions, the DNN is trained on the solution domain of the GPDEs. Moreover, a surface (manifold) can be implicitly represented by a level set method (LSM), which was first introduced by Osher and Sethian [20], and has significant impact on the computational field. Currently, DNN has applied successfully for solving PDEs, such as hidden physics models [21] and physics-informed neural networks (PINNs) [22].

In this work, motivated by hidden physics models, PINNs, and LSM, a robust deep LSM learning of the data-driven high-order GPDEs method is proposed. Our contributions are summarized as follows:

- We focus on the data-driven and learning methods to solve two GPDEs: (1) Quasi Xuguo flow [23], (2) High-order surface diffusion flow of Cahn–Hilliard model.
- Theoretically, our framework is flexible to adapt to different high-order GPDEs, which enables effective integration of traditional LSM and DNN to improve computational efficiency. Our experiments confirm this property.

The rest of the paper is organized as follows, in Sect. 2, we first present some related works about learning high-order geometric flow. Furthermore, we introduce the proposed algorithm about learning high-order geometric flow based on the level set method in Sect. 3. After that, in Sect. 4, we describe the experimental settings and experimental results. Finally, we summarize this paper and present the limitations of the current study, and present several future research directions in Sect. 5.

2 Related work

In this section, we briefly review the most relevant studies from the following three aspects: (1) The manifold learning and some definitions, (2) The level set method, and (3) The deep learning-based approach for solving PDEs.

2.1 Manifold learning and some definitions

Manifold learning is a method for nonlinear dimension reduction. Algorithms for this task think that the dimension of several data sets is only artificially high. Set \( S = \{ u(x, y) \in R^3 : (x, y) \in D \in R^2 \} \) be a sufficiently smooth, regular and the parametric surface. And let \( g = \langle u_x, u_y \rangle \) be the coefficients of the first and second fundamental forms of surfaces with \( u_x = \frac{\partial u}{\partial x}, u_y = \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x \partial y} = u_{xy} \), where \( \langle \cdot, \cdot \rangle \) refers to the usual inner product in Euclidean space \( R^3 \).

**Definition 1 (Tangential gradient operator).** Suppose \( |\nabla \phi| \neq 0 \) on some open neighborhood \( \Omega \) of the level set \( \Gamma = \{ (x, y) : \phi(x, y, t) = 0 \} \), \( f \) is a differentiable function on \( \Omega \), then the tangential gradient operator \( \nabla_\tau \) acting on \( f \) is given by \( \nabla_\tau f = P \nabla f \), where \( P = I - n \otimes n = I - nn^T \) is the projection operator to the tangential plane of the surface \( \Gamma \), \( n = \frac{\nabla \phi}{|\nabla \phi|} \), and \( I \) refers to the identity mapping.
| Authors and Year          | Objective                                                                 | Advantage                                                                 | Disadvantage                                                                 | Application                                                                 |
|--------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Lustig et al. [24] 2020  | Identifying topological phase transitions in experiments using manifold learning | A nonlocal unsupervised machine learning method to the identification of topological phase transitions | The diffusion maps are used to identify phase transitions in quantum many-body systems are very challenging | Identification of topological phase transitions                             |
| Eo et al. [25] 2020      | Accelerating cartesian MRI by domain-transform manifold learning in phase-encoding direction | A domain-transform framework comprising domain-transform manifold learning was proposed to reconstruct image | The properties of the k-space that make the algorithm is still a challenge for reconstruction results | Accelerate reconstruct cartesian magnetic resonance imaging                 |
| Ma et al. [26] 2020      | Manifold learning based on straight-like geodesics and local coordinates   | A manifold learning algorithm is developed                               | The computational cost is relatively high                                      | Data classification                                                         |
| Pournemat et al. [27] 2021 | Semisupervised charting for spectral multimodal manifold learning and alignment | A semi-supervised learning model was proposed to different modalities data. | Lacking applications such as multimodal image registration, cross-modal image retrieval | Multimodal data                                                            |
| Mehrdad and Kahaei [28] 2021 | Deep Learning approach for matrix completion using manifold learning | The new algorithm address both linear and nonlinear relations among entries of the data matrix | Super parameters need to be adjusted automatically, it would be better if the model can learn the super parameters automatically | Matrix completion using manifold learning                                  |
| Rodrigues et al. [29] 2021 | Manifold learning for real-world event understanding                      | A learning-from-data method for dynamically learning the contribution of different components for a more effective event representation | Lacking the analysis the context of the rankings                              | Real-world event understanding                                              |
| Chen et al. [30] 2021    | Semisupervised feature selection via structured manifold learning         | Solving the multimodality problem and its superior performance compared with the state-of-the-art methods | The new method is not scalable to large-scale data                           | Feature selection                                                          |
| Authors and Year          | Objective                                                                 | Advantage                                                                                                                                  | Disadvantage                                                                                                                                      | Application                  |
|--------------------------|----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Yan et al. [31] 2020     | Convexity shape prior for level set-based image segmentation method         | The convexity shape prior based on the signed distance function was proposed. A new algorithm was proposed to solve the constraint optimization problem | It belongs to the traditional optimization method and is not combined with the new machine learning methods such as deep learning                  | Image segmentation           |
| Falcone et al. [32] 2020 | A high-order scheme for image segmentation via a modified level set method | A modified level set approach for the image segmentation problem was proposed. The method can solve the segmentation problem with noise | Authors only consider the classical first-order equation, not consider the second-order problems                                                      | Image segmentation           |
| Liu et al. [33]          | B spline level set method for shape reconstruction in electrical impedance tomography | The shapes reconstruction was implicitly represented by a level set function                                                                 | The reconstruction problem needs to be solved iteratively. The selection of control points net was not optimal                                     | Shape reconstruction in electrical impedance tomography |
| Howard and Tartakovsky [34] 2021 | A conservative level set method for N phase flows with a free-energy-based 11 surface tension model. | The authors developed a method to solve the multiphase flow                                                                                   | High computational cost and long time consuming                                                                                              | Multiphase flow              |
| Luis and Gibou [35] 2021 | A deep learning approach for the computation of curvature in the level set method | The mean curvature is estimated by the deep learning strategy and the level set method.                                                      | Robustness is poorer than traditional methods                                                                                                 | Curvature computation        |
| He et al. [36] 2021      | A level set method for inhomogeneous SAR image segmentation                 | A novel level set method that integrates the local intensity and the global feature information to solve image segmentation tasks             | Few evaluation indexes, relative lack of testing                                                                                               | Image segmentation           |
### Table 3  Some recent references about the deep PDEs solver

| Authors and Year | Objective | Advantage | Disadvantage | Application |
|------------------|-----------|-----------|--------------|-------------|
| Huang et al. [37] 2020 | Nonlinear problems are solved with deep learning initialized iterative method | A deep initialized iterative method was developed. This method can handle variational inequalities and eigenvalue problems. | The design of the solution process is relatively cumbersome | Variational inequalities and eigenvalue problems |
| Belbute-Peres [38] 2020 | Combining differentiable PDE solvers and graph neural networks for fluid flow prediction | The higher fidelity results can be provided by the hybrid network. And this approach improved computing performance. | Some specific practical applications are missing | Fluid flow prediction |
| Liang et al. [10] 2020 | PDE learning of filtering and propagation for task-aware facial intrinsic image analysis | A PDE learning framework that unifies both filtering and propagation operations was developed. The mathematical relations of different filtering and propagation models were derived. | If this model can complete multi-task learning, it will be more perfect | Facial intrinsic image analysis |
| So et al. [39] 2021 | Differential spectral normalization (DSN) for PDE discovery | A novel regularization method tailored for moment-constrained filters was developed. | Lack of specific applications in various disciplines | PDE discovery problem and the limitations of existing data-driven approaches |
| Bar and Sochen [40] 2021 | Strong solutions for PDE-based tomography by unsupervised learning | A novel neural PDE solver for forward and inverse problems was proposed | Lack of research on two types of problems such as higher dimensional problems and dynamic nonlinear equations | Electrical Impedance Tomography (EIT) |
| Lu et al. [41] 2021 | Deep operator networks | A deep operator network was proposed to learn operators accurately and efficiently from a relatively small dataset | There have not been any theoretical results of network size for operator approximation. | Identifying two types of operators |
| Thanasutives et al. [42] 2021 | Multi-task learning | A novel approach was developed to solve multitask learning | This method lack handles chaotic systems, parameterized PDEs, and a system of PDEs | Multitask learning |
and Table 2. Other related works about manifold learning include
tors, compared with the traditional mesh-based numerical schemes include FVM, FDM, and FEM, the DNNs
are inherently mesh-free function-approximators. Such that they not only can avoid the curse of dimensionality
but also approximate the solutions of PDEs on complex geometries effectively. One remarkable application
of DNNs is the physics-informed neural networks (PINNs) [22], which can solve both forward and inverse
problems with the desired accuracy. Also, Sirignano et al. [62] developed a method called DGM for solving
PDEs. Moreover, Long et al. [2] proposed PDE-Net for learning PDEs from data. For more examples on
solving differential equations with DL, please refer to [21,41,44,63–68] and Table 3.

3 The proposed method

3.1 High-order quasi Xuguo flow

In this section, we describe more details about the algorithm of learning high-order geometric flow based
on the LSM. In [23], Xu and Zhang constructed the Quasi Xuguo flow from the perspective of computational
geometry. However, the higher-order geometric flow is not solved analytically and numerically in that paper, according to [23] and LSM, the high-order GPDE is obtained as follows,

$$\phi_t = \Delta_3 \left( di v \left( \frac{\nabla \phi}{\| \nabla \phi \|_\epsilon} \right) \right) \| \nabla \phi \|_\epsilon,$$

where $\Delta_3$ refers to LBO, $\| \nabla \phi \|_\epsilon = \sqrt{\phi_x^2 + \phi_y^2 + \epsilon}, \kappa = di v \left( \frac{\nabla \phi}{\| \nabla \phi \|_\epsilon} \right)$ refers to the mean curvature (MC), and $\epsilon > 0$. According to [69], take $\kappa$ as an example, the LBO can be explicitly formulated as following,

$$\Delta_3 \kappa = \frac{\kappa_{xx} \left( 1 + \kappa_{y}^2 \right) + \kappa_{yy} \left( 1 + \kappa_{x}^2 \right) - 2 \kappa_{x} \kappa_{y} \kappa_{xx}}{(1 + \kappa_{x}^2 + \kappa_{y}^2)^2}. \quad (2)$$

Definition 2 (Tangential divergence operator). Suppose $\| \nabla \phi \| \neq 0$ on some open neighborhood $\Omega$ of the
level set $\Gamma = \{ (x, y) : \phi(x, y, t) = 0 \}$, $\mathbf{v}$ is a smooth vector field defined on $\Omega$, the divergence operator $\text{div}_S$ acting on $\mathbf{v}$ is given by $\text{div}_S(\mathbf{v}) = \text{div}(\mathbf{v}) - \mathbf{n}^T (\nabla \mathbf{v}) \mathbf{n}$, where $\text{div}$ is the usual divergence operator.

Definition 3 (Laplace-Beltrami operator (LBO)). Suppose $\| \nabla \phi \| \neq 0$ on some open neighborhood $\Omega$
of the level set $\Gamma = \{ (x, y) : \phi(x, y, t) = 0 \}$, $f$ is twice differentiable function on $\Omega$, then the Laplace-
Beltrami operator $\Delta_3$ acting on $f$ is given by $\Delta_3 f = \text{div}_S (\nabla f)$.

Recently, manifold learning has made great progress. For example, Bachmann et al. [1] developed constant
curvature graph convolutional networks (GNN), bridging the gap that popular GNNs consider the data only
via Euclidean geometry and associated vector space operations. Also, Sanchez-Gonzalez et al. [3] used
the GNN to simulate complex physical phenomenon. Moreover, Chen et al. [43] utilized the convolutional
kernel networks for learning the graph-structured data. Other related works about manifold learning, include
and not only include the following references [44–48] and Table 1.

2.2 Level set method

The LSM was first developed by Osher and Sethian [20], which made a huge impact on computational
methods for interface motion. Since the surface manifold can be implicitly represented by the LSM, therefore,
which is widely used in various fields including computational geometry, fluid mechanics, computer
vision, and materials science [49]. Moreover, Fedkiw et al. [50] used the LSM for representing the dynamic
implicit surfaces. More recently, Lin et al. [51] developed the LSM for solving constrained convex optimization.
More about the LSM please refer to [31,52–61] and Table 2.

2.3 Deep learning-based approach for solving PDEs

Recently, with the rapid development of DL, the numerical methods of PDEs have made significant
progress. Advantageously, as mesh-free approximators, compared with the traditional mesh-based numerical
schemes include FVM, FDM, and FEM, the DNNs

$\odot$ Springer
Based on Eq. (2), such that Eq. (1) can be rewritten as follows,

\[
\begin{align*}
\kappa &= \frac{\phi_{xx}(1+\phi_y^2)+\phi_{yy}(1+\phi_x^2)-2\phi_x\phi_y\phi_{xy}}{(1+\phi_x^2+\phi_y^2)^2}, \\
\phi_t &= \Delta_s^3(\kappa) \|\nabla \phi\|_e, \\
\Delta_s \kappa &= \frac{\kappa_{xx}(1+\kappa_y^2)+\kappa_{yy}(1+\kappa_x^2)-2\kappa_x\kappa_y\kappa_{xy}}{(1+\kappa_x^2+\kappa_y^2)^2}.
\end{align*}
\] (3)

To obtain the order reduction of GPDEs, the new variables \( q = \Delta_s \kappa \), \( w = \Delta_s q \) are introduced. Hence, \( \Delta_s q \) and \( \Delta_s w \) can be explicitly formulated via the same approach as \( \Delta_s \kappa \). Therefore, Eq. (1) can be reformulated as follows,

\[
\begin{align*}
q &= \Delta_s \kappa, \\
w &= \Delta_s q, \\
\phi_t &= \Delta_s w \|\nabla \phi\|_e.
\end{align*}
\] (4)

Motivated by PINNs, the GPDEs are encoded into the loss function, and the partial derivatives can be computational via automatic differentiation (AD). For convenience, the following symbols are introduced as,

\[
\begin{align*}
e_2 &= \kappa - \frac{\phi_{xx}(1+\phi_y^2)+\phi_{yy}(1+\phi_x^2)-2\phi_x\phi_y\phi_{xy}}{(1+\phi_x^2+\phi_y^2)^2}, \\
e_3 &= \Delta_s \kappa - \frac{\kappa_{xx}(1+\kappa_y^2)+\kappa_{yy}(1+\kappa_x^2)-2\kappa_x\kappa_y\kappa_{xy}}{(1+\kappa_x^2+\kappa_y^2)^2}, \\
e_4 &= \Delta_s q - \frac{q_{xx}(1+q_y^2)+q_{yy}(1+q_x^2)-2q_xq_yq_{xy}}{(1+q_x^2+q_y^2)^2}, \\
e_5 &= \Delta_s w - \frac{w_{xx}(1+w_y^2)+w_{yy}(1+w_x^2)-2w_xw_yw_{xy}}{(1+w_x^2+w_y^2)^2}, \\
e_1 &= \phi_t - \Delta_s w \|\nabla \phi\|_e, \\
e_6 &= q - \Delta_s \kappa, \\
e_7 &= w - \Delta_s q.
\end{align*}
\] (5)

Based on the analysis above, then the total loss function is obtained as follows,

\[
\text{Loss} = \text{Loss}_{IC} + \text{Loss}_{BC} + \sum_{i=1}^{7} \|e_i\|_2^2,
\] (6)

where \( IC \) and \( BC \) refer to the initial and bound conditions, respectively. In this work, the initial conditions are discussed in more detail later, and the boundary condition is the periodic boundary one.

Usually, the systems of PDEs are viewed as the PDE-constrained optimization problem,

\[
\min_{\Theta} L(\Theta) \text{ s.t. } F(\Theta, u) = 0,
\] (7)
where $\Theta$ is the set of model parameters, and $L$ is the loss function. When we replace $\Theta$ with $NN_{\Theta}$, the following physics-constraint optimization learning one is obtained,

$$\min_{\Theta} L(\Theta) \text{ s.t. } F(NN_{\Theta}, u) = 0. \tag{8}$$

The gradient descent approach can be used for solving it, and the parameter update equation is,

$$\Theta_{n+1} = \Theta_n - \alpha_n \nabla_{\Theta} L(\Theta_n), \tag{9}$$

where $L$ refers to Eq. (6). The algorithm is summarized as in Algorithm 1, and the schematic illustration of our LSM-physics constrained learning (PCL) framework for learning GPDE is shown in Fig. 1.

### 3.2 High-order surface diffusion flow of Cahn–Hilliard model

Such as [7], we want to approximate the surface diffusion flow using LSM and high-order LBO of $\kappa$, to smoothing the system of PDEs, the Heaviside function is introduced as,

$$\begin{align*}
H_{\varepsilon}(u) &= 0.5 + \frac{1}{\pi \varepsilon} \arctan \left( \frac{u}{\varepsilon} \right), \\
\delta_{\varepsilon}(u) &= H'_{\varepsilon}(u) = \frac{\varepsilon}{\pi (\varepsilon^2 + u^2)},
\end{align*} \tag{10}$$

where $\varepsilon$ is a small positive constant. Then, the proposed new Cahn-Hilliard model reads as,

$$\begin{align*}
\phi_t &= N(\phi) \text{div}_{\varepsilon}(M(\phi) \nabla_{\varepsilon}(N(\phi) \mu)) \parallel \nabla \phi \parallel_{\varepsilon} \delta_{\varepsilon}(\phi), \\
\mu &= \frac{1}{\varepsilon} W'(\phi) - \Delta_{\varepsilon} \kappa \parallel \nabla \phi \parallel_{\varepsilon} \delta_{\varepsilon}(\phi), \\
M(\phi) &= W(\phi) + \gamma \varepsilon^2, \quad \gamma > 0, \\
N(\phi) &= \sqrt{M(\phi)}; \quad W(\phi) = 0.5\phi^2(1-\phi)^2.
\end{align*} \tag{11}$$

Since the Definition 1–2, we can get a smooth vector field $v$ defined on $\Omega$, and the divergence operator $\text{div}_{S}$ acting on $v$, which has the following explicit representation,

$$\begin{align*}
\text{div}_{S}(v) &= \text{div}(v) - \nabla^T(\nabla v) \cdot n \\
&= v_x + v_y - \frac{\phi_{x}^2 v_x + \phi_{y}^2 (v_x + v_y) + \phi_{y}^2 v_y}{\phi_{x}^2 + \phi_{y}^2 + 1}.
\end{align*} \tag{12}$$

### Algorithm 1: The framework of learning high-order geometric flow based on the LSM.

**initialization:**
1. Initialize the parameters $\Theta$, and the learning rate $\alpha_n$;
2. Initialize $u^0$;

for $n = 1, 2, 3, \ldots, N$ do
1. Read current $L(\Theta)$ via Eq. (6);
2. Update the parameters $\Theta$ via gradient descent

$$\Theta_{n+1} = \Theta_n - \alpha_n \nabla_{\Theta} L(\Theta_n) - \alpha_n \sum_{i=1}^{N} \nabla_{\Theta} L_i(\Theta_n);$$

end

**output:**
1. Output the learned solution;
2. Output the parameters $\Theta_{n+1}$;

### Table 4: Comparison results of SSIM, PSNR, and NRMSE for high-order Quasi Xuguo flow under different initial conditions (FNN)

| Noise level | SSIM     | PSNR     | NRMSE    |
|-------------|----------|----------|----------|
| Condition-1 |          |          |          |
| Level = 0.3 | 0.8496   | 15.0381  | 0.1987   |
| Level = 0.5 | 0.8542   | 15.5021  | 0.1884   |
| Level = 0.9 | 0.8124   | 12.4637  | 0.2673   |
| Condition-2 |          |          |          |
| Level = 0.3 | 0.8568   | 13.8815  | 0.2243   |
| Level = 0.5 | 0.8569   | 13.7903  | 0.2267   |
| Level = 0.9 | 0.8174   | 11.9501  | 0.2802   |
| Condition-3 |          |          |          |
| Level = 0.3 | 0.9362   | 19.5200  | 0.1157   |
| Level = 0.5 | 0.9459   | 20.9853  | 0.0978   |
| Level = 0.9 | 0.8880   | 15.6270  | 0.1812   |

Since Eq. (12) and Definition 1, so Eq. (11) is explicitly represented, thereafter, which can be solved by Algorithm 1 with order reduction (Fig. 3).

### 4 Experiments

To test the robustness of the flows, we add different degrees of Gaussian noise to the initial conditions, and the noise levels are 0.3, 0.5, and 0.9, respectively. Furthermore, the metrics are shown as follows,

1. **Peak signal-to-noise ratio (PSNR)** [75] is an engineering term for the ratio between the max-
Fig. 2 Examples of the visualization process of evolutionary for high-order Quasi Xuguo flow with initial condition 1 under different NNs.
Fig. 3  Examples of the visualization process of evolutionary for high-order Quasi Xuguo flow under different initial conditions (noise level is 0.9, FNN)
Fig. 4 Examples of the visualization process of evolutionary for high-order surface diffusion flow of Cahn–Hilliard under different initial conditions and noise levels (FNN)
Table 6  The comparison of the training loss values and times for high-order Quasi Xuguo flow under different initial conditions

| Initial condition | Model     | Training loss | Training time (s) |
|-------------------|-----------|---------------|-------------------|
| Initial condition-1 | CBDNet [70] | 0.0080        | 313.986           |
|                   | RDNet [71] | 17.6982       | 585.564           |
|                   | VDNNet [72] | 2.5858       | 41.836            |
|                   | DnCNN [73] | 58.3773       | 32.224            |
|                   | FNN [74]   | 0.7245        | 44.1791           |
| Initial condition-2 | CBDNet [70] | 0.0164        | 316.049           |
|                   | RDNet [71] | 10.0092       | 590.711           |
|                   | VDNNet [72] | 0.0160       | 41.845            |
|                   | DnCNN [73] | 20.1100       | 31.986            |
|                   | FNN [74]   | 0.1599        | 46.657            |
| Initial condition-3 | CBDNet [70] | 0.0052        | 313.050           |
|                   | RDNet [71] | 0.0004        | 579.060           |
|                   | VDNNet [72] | 0.000001     | 41.789            |
|                   | DnCNN [73] | 0.0007        | 32.212            |
|                   | FNN [74]   | 0.000009      | 43.615            |

Table 7  The comparison of the training loss values and times for high-order surface diffusion flow of Cahn–Hilliard under different initial conditions

| Initial condition | Model     | Training loss | Training time (s) |
|-------------------|-----------|---------------|-------------------|
| Initial condition-1 | CBDNet [70] | 1.0285        | 395.204           |
|                   | RDNet [71] | 31.4177       | 791.001           |
|                   | VDNNet [72] | 0.9729       | 68.605            |
|                   | DnCNN [73] | 976.5492      | 52.6293           |
|                   | FNN [74]   | 373.9023      | 72.109            |
| Initial condition-2 | CBDNet [70] | 6409.9716     | 390.993           |
|                   | RDNet [71] | 75.7234       | 795.453           |
|                   | VDNNet [72] | 13556.854554624.00 | 68.438 |
|                   | DnCNN [73] | 703.5822      | 52.212            |
|                   | FNN [74]   | 97.3872       | 72.737            |
| Initial condition-3 | CBDNet [70] | 0.7561        | 392.306           |
|                   | RDNet [71] | 0.00049       | 798.342           |
|                   | VDNNet [72] | 0.000003      | 70.868            |
|                   | DnCNN [73] | 605.8007      | 53.794            |
|                   | FNN [74]   | 0.0029        | 72.109            |

PSNR \( (u^*, \hat{u}) = 10\log_{10} \left( \frac{255^2mn}{\|u^*-\hat{u}\|^2_2} \right) \), here, \( u^* \in R^{m \times n} \) is the clean solution image, \( \hat{u} \in R^{m \times n} \) is the learned solution image, and \( \tilde{u} \in R^{m \times n} \) is the noisy data.

2. The structural similarity index measure (SSIM) [75] is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. SSIM is used for measuring the similarity between two images. SSIM measures the two patches, \( x \) and \( y \), as follows:

\[
SSIM(x, y) = \left( \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \left( \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right)
\]

where \( \mu_x \) and \( \mu_y \) are the means of images \( x \) and \( y \), \( \sigma_x^2 \) and \( \sigma_y^2 \) are the variances of images \( x \) and \( y \), and \( \sigma_{xy} \) is the covariance of images \( x \) and \( y \). The constants \( C_1 \) and \( C_2 \) are used to avoid division by zero.
Table 8 Summary of some initial conditions of $u$
(observed data)

| Initial condition | $u_{init}$ |
|-------------------|------------|
| Condition-1       | $u_0 = \begin{pmatrix} 10 \text{max}(0.04 - (x - 0.2)^2 - (y - 0.65)^2, 0) \\
+ 12 \text{max}(0.03 - (x - 0.5)^2 - (y - 0.2)^2, 0) \\
+ 12 \text{max}(0.03 - (x - 0.8)^2 - (y - 0.55)^2, 0) \\
\tanh(0.2 - \sqrt{(x - 0.3)^2 + (y - 0.3)^2}) \\
\times \tanh(0.2 - \sqrt{(x - 0.3)^2 + (y - 0.5)^2}) \\
\end{pmatrix}$ |
| Condition-2       | $u_0 = \begin{pmatrix} 0.03 \sqrt{2} \\
\tanh(0.2 - \sqrt{(x - 0.3)^2 + (y - 0.5)^2}) \\
\end{pmatrix}$ |
| Condition-3       | \[
\begin{cases} 
  u_0 = 0.5 \tanh \frac{-r + \sqrt{(x-a_x)^2 + (y-a_y)^2}}{2\sqrt{2}}, \\
  + 0.5 \tanh \frac{-r + \sqrt{(x-b_x)^2 + (y-b_y)^2}}{2\sqrt{2}}, \\
  \varepsilon = 0.01 \\
  a_x = -\frac{r}{\sqrt{2}}, a_y = \frac{r}{\sqrt{2}}, b_x = \frac{r}{\sqrt{2}}, b_y = -\frac{r}{\sqrt{2}}, r = 0.2\sqrt{2}.
\end{cases}
\]

Fig. 5 Training loss under different situation
and $y$ correspond to the same spatial window of the images $X$ and $Y$, respectively. The SSIM value for the patches $x$ and $y$ is given as $SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\mu_x^2 + \mu_y^2 + C_1(\sigma_x^2 + \sigma_y^2 + C_2)}$. $\mu_x$ refers to the mean of $x$, $\sigma_x$ refers to the standard deviation of $x$, $\sigma_{xy}$ is the cross-correlation of the mean shifted images $x - \mu_x$ and $y - \mu_y$, and the $C_i$, $(i = 1, 2)$ are small positive constants. Therefore, SSIM is used to measure the similarity between two images of learned solutions.

3. **Normalized Root Mean Square Error (NRMSE)**, given a ground-truth image $y(x)$ and a generated $\hat{y}(x)$, NRMSE can be defined as follow,

$$\text{NRMSE} = \sqrt{\frac{\|y(x) - \hat{y}(x)\|_2^2}{\|y(x)\|_2^2}}.$$ 

In this section, we describe more details about the experimental setup and results while test the performance of our method. We train the deep neural network models on our equipment with a GeForce RTX 1080 super GPU. The software is developed based on the PyTorch framework [76]. We present several numerical examples in two dimensions, including various phenomena to test the convergence of the proposed framework on the synthetic data. Our computational domain is the square $\Omega = [0, 1] \times [0, 1]$, and $t \in [0, 5]$. We use periodic boundary conditions in all directions. And the computational parameters with a uniform space-grid $N_x = N_y = 100$ and a uniform time-grid $N_t = 100$ in all examples. Also, we test five NNs, namely, the feed-forward NN with 14 hidden layers (each layer contains 50 neurons) [74], DnCNN [73], CBDNet [70], RDNet [71], and VDNet [72] to test the robustness of our framework for various networks. We choose 50 boundary sampling points, 50,000 inner sampling points, and 5000 initial sampling points for training. Examples of the visualization process of evolutionary for high-order flows with initial conditions are shown in Figs. 2, 3 and 4. And the training loss under different situations is shown in Fig. 5. Also, the comparison of the training loss values and times for high-order flows under different initial conditions are shown in Tables 6 and 7. The initial conditions of $u$ are given in Table 8. Furthermore, comparison results of SSIM, PSNR, and NRMSE for high-order flows under different initial conditions (FNN) is shown in Table 4-5. To study the influence of various optimizers on the algorithm system, we used the following optimizers to test the system, including Adadelta [77], Adagrad [78], Adam [79], and RMSprop [80]. And the training loss under different situations is shown in (f) of Fig. 5. We found that, except for Adadelta, other optimizers converge very quickly.

About parameter settings, it should be noted here that during the training process, the initial condition weighting parameter is 100. Moreover, the boundary value condition and PDE-loss weighting parameters are 1. In all experiments, $\varepsilon$ in the Heaviside function is 0.5, and $\varepsilon$ in the initial condition 3 is 0.01. Furthermore, the learning rate is 0.001, $\gamma$ in Eq. (11) is 0.1, and the number of iterations is 1000.

5 Conclusion

In this work, we explore the problem of high-order Quasi Xuguoc flow and high-order surface diffusion CH flow with different initial conditions based on deep LSM. Also, we use different initial conditions that aim to study the sensitivity of the algorithm on different input data. Moreover, to verify the robustness of the algorithm, we used two networks for testing our algorithm. Theoretically, almost all networks fit our framework. Besides, through the combination of the traditional LSM and DNN, such that our framework becomes a powerful tool for solving high-order GPDEs. Furthermore, we study the influence of different optimizers on the learning results and convergence of the algorithm. And these optimization meth-
ods mainly include Adadelta [77], Adagrad [78], Adam [79], and RMSProp [80]. The test results show that Adam is more suitable for our framework. In future work, we will use real-world data to test our framework and solve practical problems in computer vision such as denoising, inpainting, reconstruction, and segmentation problem.

Acknowledgements Thank you for your valuable comments from any anonymous reviewers.

Funding This Project is funded by China Postdoctoral Science Foundation (No. 2021M690837) and Shenzhen Higher Education Institutions Stable Support Plan (Nos. GXWD20201230155 427003-20200729105427008).

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Data Availability All materials (data and code) are available from GitHub (https://github.com/lichun0503/learning-Geometric-Flow).

References

1. Bachmann, G., Bécigneul, G., Ganea, O.-E.: Constant curvature graph convolutional networks. In: International Conference on Machine Learning (2020)
2. Long, Z., Lu, Y., Ma, X., Dong, B.: PDE-net: learning PDEs from data. In: International Conference on Machine Learning, pp. 3208–3216, PMLR (2018)
3. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P.: Learning to simulate complex physics with graph networks. In: International Conference on Machine Learning, pp. 8459–8468, PMLR (2020)
4. Xue, T., Beatson, A., Adriaenssens, S., Adams, R.: Amortized finite element analysis for fast PDE-constrained optimization. In: International Conference on Machine Learning, pp. 10638–10647. PMLR (2020)
5. Li, C., Yang, Y., Liang, H., Wu, B.: Robust PCL discovery of data-driven mean-field game systems and control problems. In: IEEE Transactions on Circuits and Systems I-Regular Papers, pp. 1–14 (2021)
6. Li, C., Yang, Y., Liang, H., Wu, B.: Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets. Knowl. Based Syst. 218, 106849–106849 (2021)
7. Bretin, E., Masmou, S., Sengers, A., Terri, G.: Approximation of surface diffusion flow: a second order variational Cahn–Hilliard model with degenerate mobilities. arXiv:2007.03793 (2020)
8. Dang, W., ke Gao, Z., Hou, L., Lv, D., Qiu, S., Chen, G.: A novel deep learning framework for industrial multiphase flow characterization. In: IEEE Transactions on Industrial Informatics, vol. 15, pp. 5954–5962 (2019)
9. Gao, Z., Dang, W., Mu, C., Yang, Y., Li, S., Grebogi, C.: A novel multiplex network-based sensor information fusion model and its application to industrial multiphase flow system. IEEE Trans. Industr. Inf. 14, 3982–3988 (2018)
10. Liang, L., Jin, L., Xu, Y.: PDE learning of filtering and propagation for task-aware facial intrinsic image analysis. IEEE Trans. Cybern. 55, 1–14 (2020)
11. Chen, D., Zhu, J., Zhang, X., Shu, M., Cohen, L.D.: Geodesic paths for image segmentation with implicit region-based homogeneity enhancement. IEEE Trans. Image Process. 30, 5138–5153 (2021)
12. Chen, J., Amini, A.A.: Quantifying 3-D vascular structures in MRA Images Using Hybrid PDE and geometric deformable models. IEEE Trans. Med. Imaging 23(10), 1251–1262 (2004)
13. Salinas, H.M., Fernández, D.C.: Comparison of PDE-based nonlinear diffusion approaches for image enhancement and denoising in optical coherence tomography. IEEE Trans. Med. Imaging 26(6), 761–771 (2007)
14. Karasev, P., Kolesov, I., Fritscher, K., Vela, P., Mitchell, P., Tannenbaum, A.: Interactive medical image segmentation using PDE control of active contours. IEEE Trans. Med. Imaging 32(11), 2127–2139 (2013)
15. Biron, R.A., Biron, Z.A., Pisu, P.: False data injection attack in a platoon of CACC: real-time detection and isolation with a PDE approach. IEEE Trans. Intell. Transp. Syst. 22, 1–12 (2021)
16. Deutschcr, J.: Robust cooperative output regulation for a network of parabolic PDE systems. IEEE Trans. Autom. Control 66, 1 (2021)
17. Song, X., Zhang, Q., Zhang, Y., Song, S.: Fuzzy event-triggered control for pde systems with pointwise measurements based on relaxed Lyapunov-Krasovskii functionals. IEEE Trans. Fuzzy Syst. 29, 1 (2021)
18. Zhao, D., Jiang, B., Yang, H.: Backstepping-based decentralized fault-tolerant control of hypersonic vehicles in PDE-ODE form. IEEE Trans. Autom. Control 66, 1–1 (2021)
19. Oliveira, T.R., Feiling, J., Koga, S., Krsti´ c, M.: Multivariable extremum seeking for PDE dynamic systems. IEEE Trans. Autom. Control 65(11), 4949–4956 (2020)
20. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J. Comput. Phys. 79(1), 12–49 (1988)
21. Raisi, M., Karniadakis, G.E.: Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. 357, 125–141 (2018)
22. Raisi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707 (2019)
23. Guoliang, X., Qin, Z.: Construction of geometric partial differential equations in computational geometry. Math. Numer. Sin. Chin. Ed. 28(4), 337 (2006)
24. Lustig, E., Yair, O., Talmon, R., Segev, M.: Identifying topological phase transitions in experiments using manifold learning. Phys. Rev. Lett. 125(12), 127401 (2020)
25. Eo, T., Shin, H., Jun, Y., Kim, T., Hwang, D.: Accelerating Cartesian MRI by domain-transform manifold learning in phase-encoding direction. Med. Image Anal. 63, 101689 (2020)
26. Ma, Z., Zhan, Z., Feng, Z., Guo, J.: Manifold learning based on straight-like geodesics and local coordinates. IEEE Trans. Neural Netw. Learn. Syst. 31, 4965–4970 (2020)
27. Pourmgnat, A., Adibi, P., Chanussot, J.: Semisupervised charting for spectral multimodal manifold learning and alignment. Pattern Recogn. 111, 107645 (2021)
28. Mehrdad, S., Kahaei, M.H.: Deep learning approach for matrix completion using manifold learning. Signal Process. 188, 108231 (2021)
29. Rodrigues, C.M., Soriano-Vargas, A., Lavi, B., Rocha, A., Dias, Z.: Manifold learning for real-world event understanding. IEEE Trans. Inf. Forensics Secur. 16, 2957–2972 (2021)
30. Chen, X., Chen, R., Wu, Q., Nie, F., Yang, M., Mao, R.: Semisupervised feature selection via structured manifold learning. IEEE Trans. Cybern. 51, 1–11 (2021)
31. Yan, S., Tai, X., Liu, J., Huang, H.: Convexity shape prior for level set-based image segmentation method. IEEE Trans. Image Process. 29, 7141–7152 (2020)
32. Falcone, M., Paolucci, G., Tozza, S.: A high-order scheme for image segmentation via a modified level-set method. SIAM J. Imag. Sci. 13, 497–534 (2020)
33. Liu, D., Gu, D., Smyl, D., Deng, J., Du, J.: B-spline level set method for shape reconstruction in electrical impedance tomography. IEEE Trans. Med. Imaging 39, 1917–1929 (2020)
34. Howard, A.A., Tartakovsky, A.: A conservative level set method for N-phase flows with a free-energy-based surface tension model. J. Comput. Phys. 426, 109955 (2021)
35. Larios-Cárdenas, L.A., Gibou, F.: A deep learning approach for the computation of curvature in the level-set method. J. Comput. Phys. 43, A1754–A1779 (2021)
36. He, W., Song, H., Yao, Y., Jia, X., Long, Y.: A novel level set method for inhomogeneous SAR image segmentation. IEEE Geosci. Remote Sens. Lett. 18, 1044–1048 (2021)
37. Huang, J., Wang, H., Yang, H.: Int-deep: a deep learning initialized iterative method for nonlinear problems. J. Comput. Phys. 419, 109675 (2020)
38. Belbute-Peres, F. d. A., Economon, T., Kolter, Z.: Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In: International Conference on Machine Learning, pp. 2402–2411. PMLR (2020)
39. So, C.C., Li, T.O., Wu, C., Yang, S.P.: Differential spectral normalization (DSN) for PDE discovery. Proc. AAAI Conf. Artif. Intell. 35, 9675–9684 (2021)
40. Bar, L., Sochen, N.: Strong solutions for PDE-based tomography by unsupervised learning. SIAM J. Imag. Sci. 14, 128–155 (2021)
41. Lu, L., Jin, P., Pang, G., Zhang, Z., Karnaíadakis, G.E.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell 3(3), 218–229 (2021)
42. Thanasutives, P., Fukui, K.-i., Numao, M.: Adversarial multi-task learning enhanced physics-informed neural networks for solving partial differential equations. arXiv:2104.14320 (2021)
43. Chen, D., Jacob, L., Mairal, J.: Convolutional kernel networks for graph-structured data. In: International Conference on Machine Learning, pp. 1576–1586. PMLR (2020)
44. Chowdhury, A.R., Rekatsinas, T., Jha, S.: Data-dependent differentially private parameter learning for directed graph-
62. Sirignano, J.A., Spiliopoulos, K.: DGM: a deep learning algorithm for solving partial differential equations. J. Comput. Phys. 375, 1339–1364 (2018)
63. Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. In: The Conference on Neural Information Processing Systems (2018)
64. Han, J., Jentzen, A.E.W.: Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. 115, 8505–8510 (2018)
65. Lu, L., Meng, X., Mao, Z., Karniadakis, G.E.: DeepXDE: a deep learning library for solving differential equations. SIAM Rev. 63(1), 208–228 (2021)
66. Cai, S., Wang, Z., Lu, L., Zaki, T.A., Karniadakis, G.E.: DeepM&Mnet: inferring the electroconvection multiphysics fields based on operator approximation by neural networks. J. Comput. Phys. 23, 110296 (2021)
67. Zhao, L., Li, Z., Wang, Z., Caswell, B., Ouyang, J., Karniadakis, G.E.: Active-and transfer-learning applied to microscale-macroscale coupling to simulate viscoelastic flows. J. Comput. Phys. 427, 110069 (2021)
68. Yang, L., Meng, X., Karniadakis, G.E.: B-PINNs: Bayesian physics-informed deep learning networks for forward and inverse PDE problems with noisy data. J. Comput. Phys. 425, 109913 (2021)
69. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, vol. 147. Springer, Berlin (2006)
70. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1712–1722 (2019)
71. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.R.: Residual dense network for image super-resolution. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2472–2481 (2018)
72. Yue, Z., Yong, H., Zhao, Q., Zhang, L.M., Meng, D.: Variational denoising network: toward blind noise modeling and removal. In: NeurIPS (2019)
73. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. 26(7), 3142–3155 (2017)
74. Bourlard, H., Wellekens, C.J.: Links between Markov models and multilayer perceptrons. IEEE Trans. Pattern Anal. Mach. Intell. 12(12), 1167–1178 (1990)
75. Hore, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition, pp. 2366–2369. IEEE (2010)
76. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: an imperative style, high-performance deep learning library. Adv. Neural. Inf. Process. Syst. 32, 8026–8037 (2019)
77. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv:1212.5701 (2012)
78. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 7 (2011)
79. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 (2014)
80. Graves, A.: Generating sequences with recurrent neural networks. arXiv:1308.0850 (2013)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.