Learn Singularly Perturbed Solutions via Homotopy Dynamics

Chuqi Chen[1], Yahong Yang[2], Yang Xiang[1,3], Wenrui Hao[2]

[1]Hong Kong University of Science and Technology, [2] Pennsylvania State University, [3] HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute

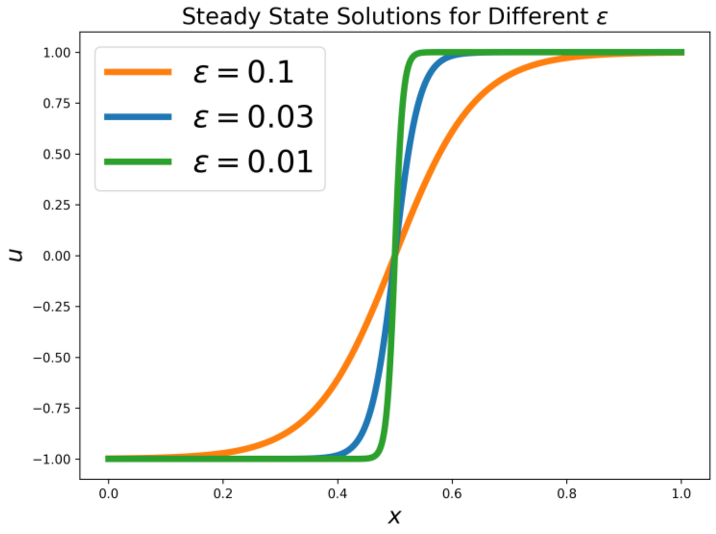
Problem Setup： Singular Perturbed Problems

$$
\begin{cases}
-\varepsilon^2 \Delta u(x) = f(u), & x \in \Omega \\
u(x) = g(x), & x \in \partial\Omega
\end{cases}
$$

Parameter ε ： influence **the structure of the solution** and **the difficulty of training neural network solvers.**

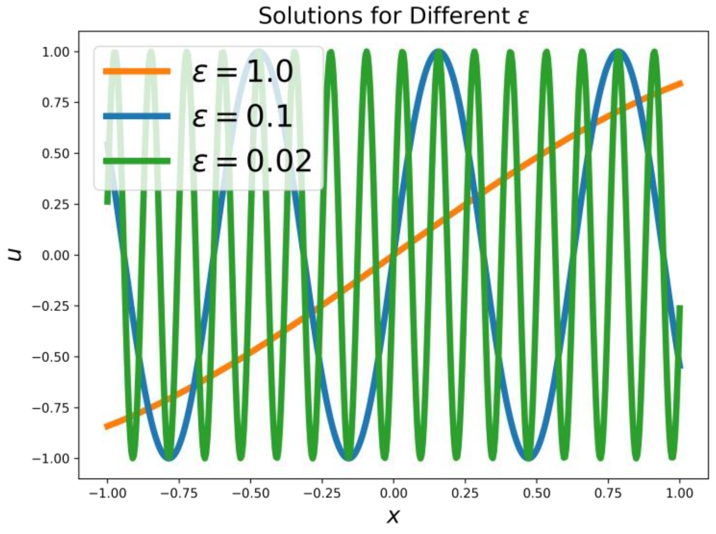
Example: Allen-Cahn equation

$$
\begin{cases}
\varepsilon^2 u''(x) + u^3 - u = 0, & x \in [0,1], \\
u(0) = -1, \quad u(1) = 1.
\end{cases}
$$



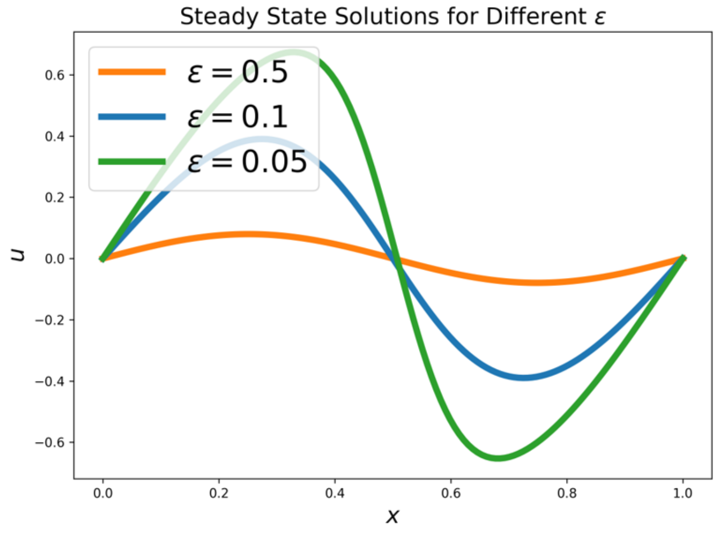
Example: Helmholtz equation

$$
\begin{cases}
\varepsilon^2 u''(x) + u = 0, & x \in [-1,1], \\
u(-1) = \sin(-\frac{1}{\varepsilon}), \quad u(1) = \sin(\frac{1}{\varepsilon}).
\end{cases}
$$



Example: Burgers' equation

$$
\begin{cases}
u_t + \left(\frac{u^2}{2}\right)_x - \varepsilon u_{xx} = \pi \sin(\pi x)\cos(\pi x), & x \in [0,1], \\
u(x,0) = u_0(x), u(0,t) = u(1,t) = 0.
\end{cases}
$$



Problem Setup： Neural Networks for Solving PDEs

$$\begin{cases} \mathcal{L}_\varepsilon u = f(u), & \text{in } \Omega, \\ \mathcal{B}u = g(x), & \text{on } \partial\Omega. \end{cases}$$

Solution approximation: Use Neural Network to approximate the solution:

$$u(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{j=1}^{M} a_j \phi_j(\boldsymbol{x}) = \sum_{j=1}^{M} a_j (\boldsymbol{k}_j \cdot \boldsymbol{x} + b_j) = \sum_{j=1}^{M} a_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}).$$

Transform the problem into an optimization problem:

PINN[1] Loss $\underset{\boldsymbol{\theta}\in\mathbb{R}^p}{\text{minimize}}$

$$L(\boldsymbol{\theta}) := \underbrace{\frac{1}{2n_{\text{res}}} \sum_{i=1}^{n_{\text{res}}} \left(\mathcal{L}_\varepsilon u(\mathbf{x}_r^i;\boldsymbol{\theta}) - f(u(\mathbf{x}_r^i;\boldsymbol{\theta}))\right)^2}_{L_{\text{res}}} + \lambda \underbrace{\frac{1}{2n_{\text{bc}}} \sum_{i=1}^{n_{\text{bc}}} \left(\mathcal{B}u(\mathbf{x}_b^i;\boldsymbol{\theta}) - g(\mathbf{x}_b^i)\right)^2}_{L_{\text{bc}}}$$

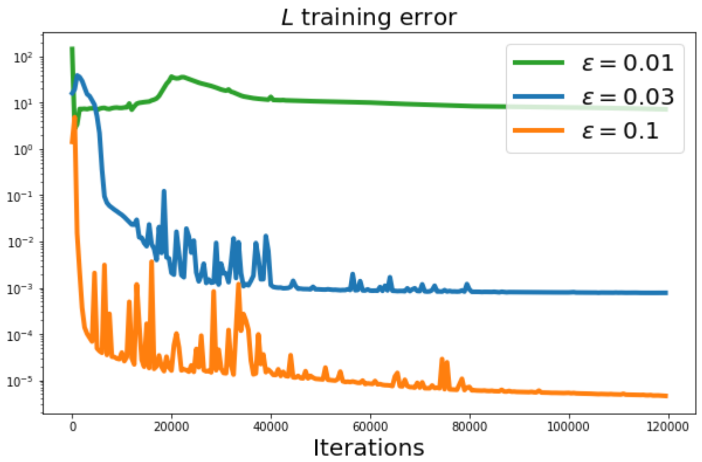
L_{res}: PDE residual loss

L_{bc}: Boundary loss

x_r: collocation points
x_b: boundary & initial points
n_{res}: # collocation points
n_{bc}: # boundary & initial points

[1] Raissi 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]. Journal of Computational physics, 2019, 378: 686-707.

Motivations



$\varepsilon \downarrow \rightarrow$ Hard to train for small ε

- **Small ε leads to training challenges — why?**
- **Can we overcome this with a more robust training strategy?**

Contributions

- We provide a theoretical explanation for **why small ε leads to slow training convergence**.
- We propose **a training algorithm based on Homotopy Dynamics** to efficiently solve singularly perturbed problems.
- We demonstrate the **effectiveness** of our method through both theoretical analysis and empirical results.

Why hard to train for small ε?

Neural Tangent Kernel farmwork:

GD dynamic:

$$\frac{\mathrm{d}L(\boldsymbol{\theta})}{\mathrm{d}t} = \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\frac{1}{n^2} \boldsymbol{l} \cdot \boldsymbol{K}_{\varepsilon} \cdot \boldsymbol{l}^{\top}.$$

Hence, the kernel of the gradient descent update is given by

$$\boldsymbol{K}_{\varepsilon} := \boldsymbol{D}_{\varepsilon} \boldsymbol{S} \boldsymbol{S}^{\top} \boldsymbol{D}_{\varepsilon}^{\top},$$

where $\boldsymbol{D}_{\varepsilon}$ depends on PDEs, and $\boldsymbol{S}$ depends on neural networks.

Theorem 1 (Effectiveness of Training via the Eigenvalue of the Kernel)

Suppose $\lambda_{min}(\boldsymbol{S}\boldsymbol{S}^{\top}) > 0$ and $\boldsymbol{D}_{\varepsilon}$ is non-singular, and let $\varepsilon \geq 0$ be a constant. Then, we have $\lambda_{min}(\boldsymbol{K}_{\varepsilon}) > 0$, and there exists $T > 0$ such that

$$L(\boldsymbol{\theta}(t)) \leq L(\boldsymbol{\theta}(0)) \exp\left(-\frac{\lambda_{min}(\boldsymbol{K}_{\varepsilon})}{n} t\right)$$

for all $t \in [0, T]$, where n is the number of sample points. Furthermore,

$$\lambda_{min}(\boldsymbol{K}_{\varepsilon}) \leq \lambda_{min}(\boldsymbol{S}\boldsymbol{S}^{\top}) \lambda_{max}(\boldsymbol{D}_{\varepsilon}\boldsymbol{D}_{\varepsilon}^{\top}).$$

Training Speed is controlled by $\lambda_{min}(K_{\varepsilon})$

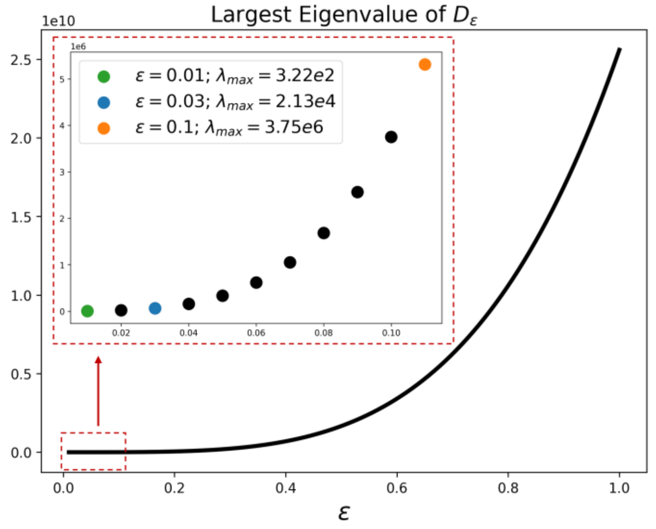
Why hard to train for small ε?

$$\begin{cases} -\varepsilon^2 \Delta u + f(u) = 0, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega. \end{cases} \longrightarrow \boldsymbol{D}_\varepsilon = -\varepsilon^2 \Delta_{\text{dis}} + \text{diag}\big(f'(u(\boldsymbol{x}_1)), \dots, f'(u(\boldsymbol{x}_n))\big).$$

$\varepsilon \downarrow \longrightarrow \lambda_{\max}(\boldsymbol{D}_\varepsilon \boldsymbol{D}_\varepsilon^T) \downarrow$, Training speed $\downarrow$, training complexity $\uparrow$

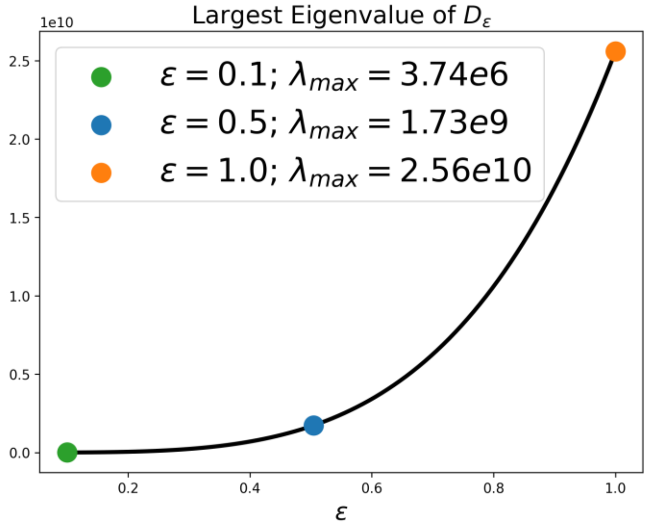
Example: Allen-Cahn equation

$$\begin{cases} \varepsilon^2 u''(x) + u^3 - u = 0, & x \in [0,1], \\ u(0) = -1, \quad u(1) = 1. \end{cases}$$



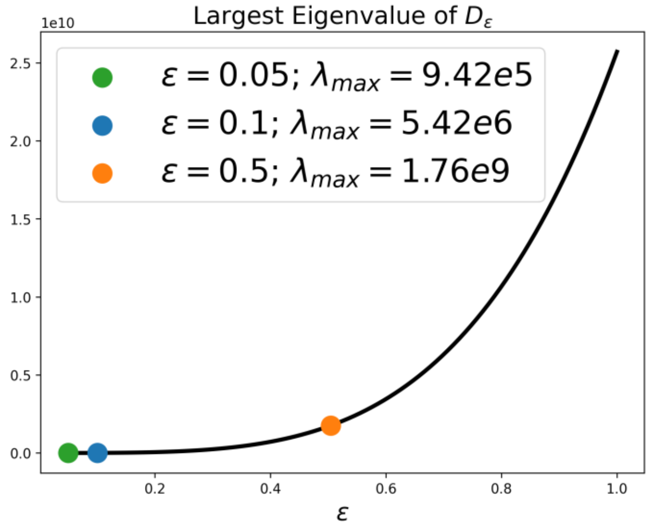
Example: Helmholtz equation

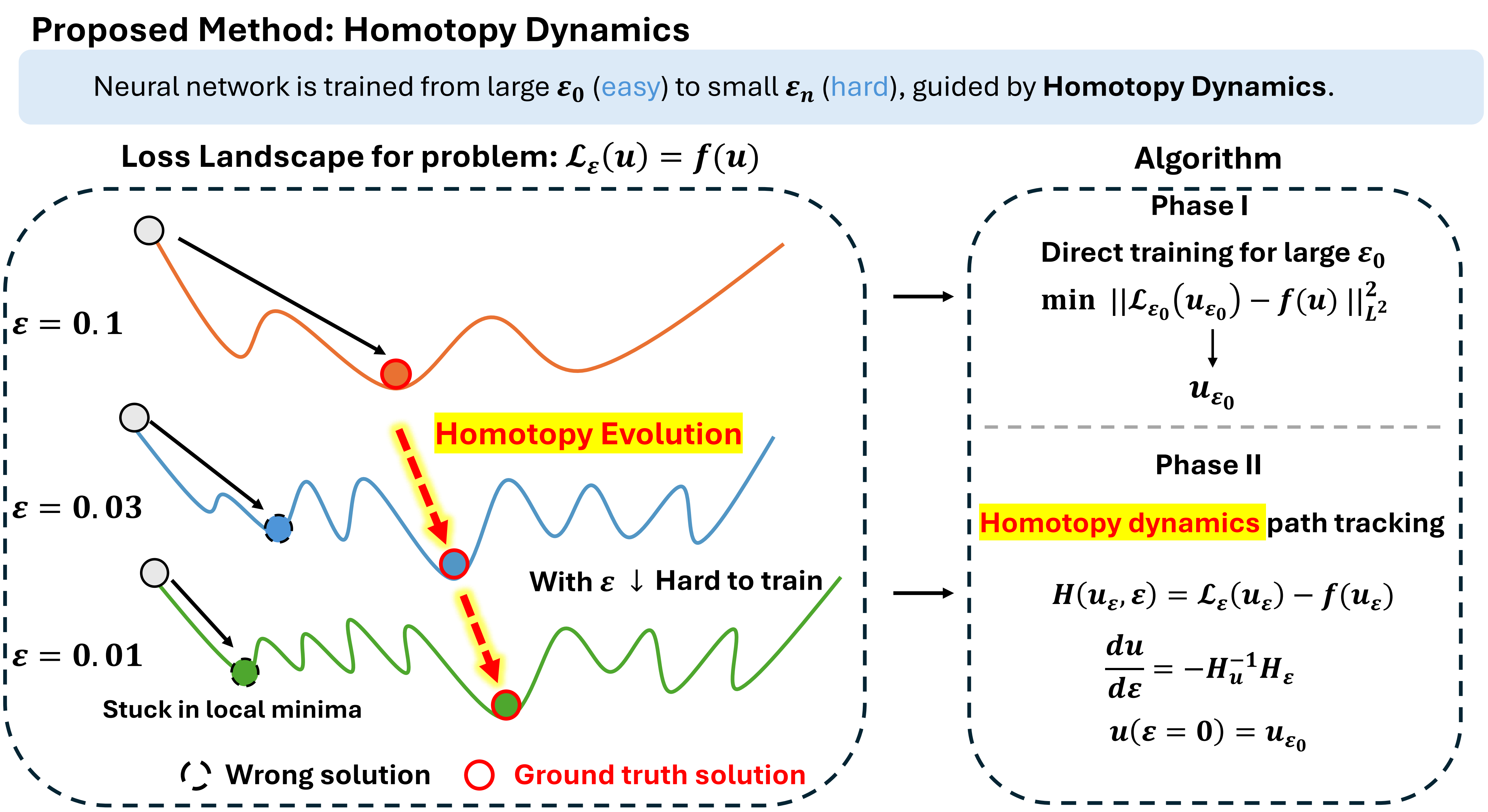
$$\begin{cases} \varepsilon^2 u''(x) + u = 0, & x \in [-1,1], \\ u(-1) = \sin(-\frac{1}{\varepsilon}), \quad u(1) = \sin(\frac{1}{\varepsilon}). \end{cases}$$



Example: Burgers' equation

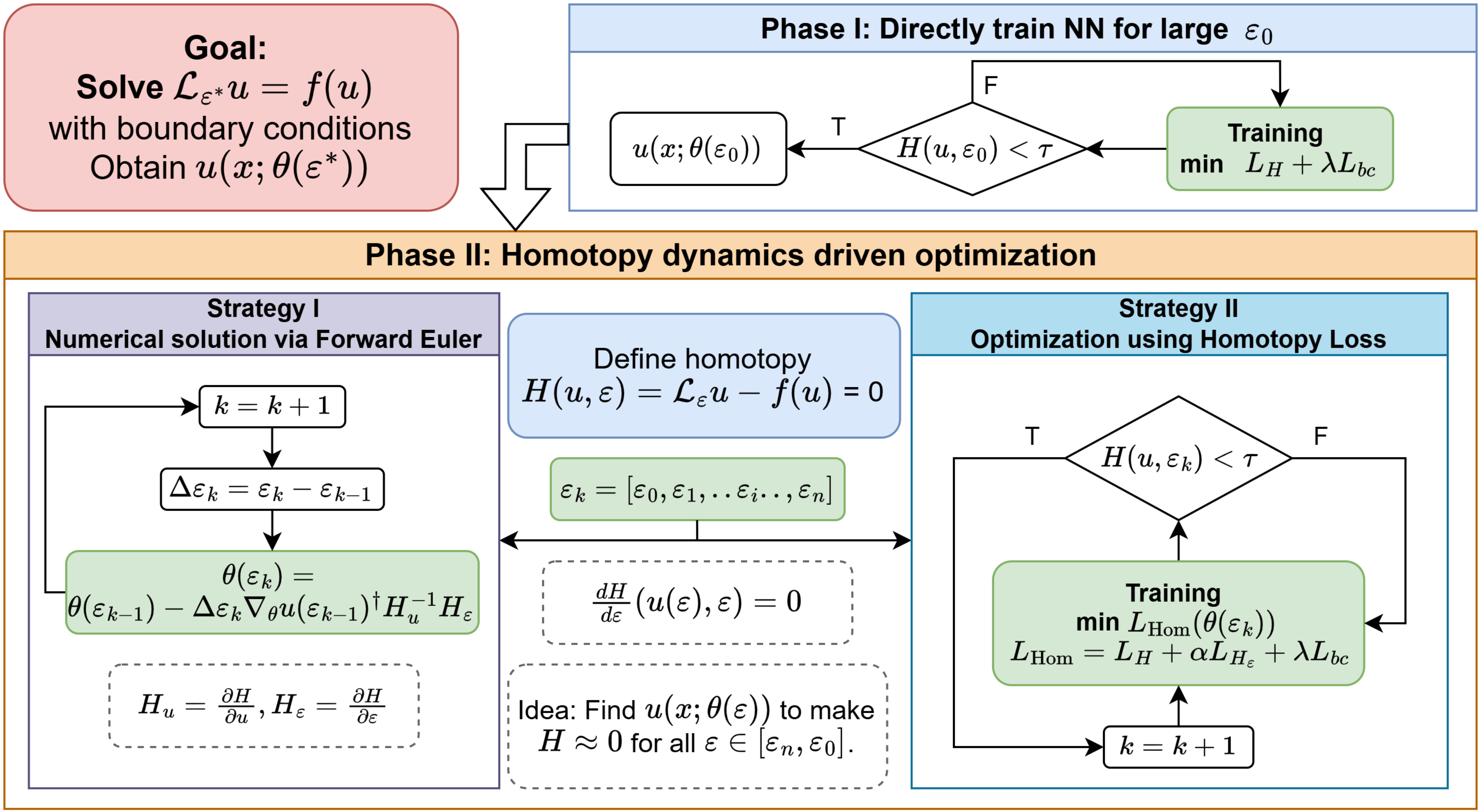
$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x - \varepsilon u_{xx} = \pi \sin(\pi x)\cos(\pi x), & x \in [0,1], \\ u(x,0) = u_0(x), u(0,t) = u(1,t) = 0. \end{cases}$$





Proposed Method: Homotopy Dynamics
Neural network is trained from large ε_0 (easy) to small ε_n (hard), guided by Homotopy Dynamics.
Loss Landscape for problem: $\mathcal{L}_\varepsilon(u) = f(u)$
$\varepsilon = 0.1$
$\varepsilon = 0.03$
$\varepsilon = 0.01$
Homotopy Evolution
With ε ↓ Hard to train
Stuck in local minima
Wrong solution
Ground truth solution
Algorithm
Phase I
Direct training for large ε_0
$\min \ ||\mathcal{L}_{\varepsilon_0}(u_{\varepsilon_0}) - f(u)\,||_{L^2}^2$
u_{ε_0}
Phase II
Homotopy dynamics path tracking
$H(u_\varepsilon, \varepsilon) = \mathcal{L}_\varepsilon(u_\varepsilon) - f(u_\varepsilon)$
$\frac{du}{d\varepsilon} = -H_u^{-1} H_\varepsilon$
$u(\varepsilon = 0) = u_{\varepsilon_0}$

Proposed Method: Homotopy Dynamics



Experimental Results: 2D Allen-Cahn equations

- **Homotopy:** For $s : 1 \to 0$, $\varepsilon(s)$ is from 1 to 0.05, u_0 is the solution for $\varepsilon = 1$,

$$H(u, s, \varepsilon) = (1 - s)\left(\varepsilon(s)^2 \Delta u - u(u^2 - 1)\right) + s(u - u_0),$$

$$\begin{cases} u_t = 1 \cdot \Delta u - u^3 + u, \boldsymbol{x} \in [-1,1]^2 \\ u(\boldsymbol{x}, t) = 0, \boldsymbol{x} \in \partial[-1,1]^2 \\ u(\boldsymbol{x}, 0) = -\sin \pi x_1 \sin \pi x_2, \end{cases} \Longrightarrow \begin{cases} u_t = 0.0025 \cdot \Delta u - u^3 + u, \boldsymbol{x} \in [-1,1]^2 \\ u(\boldsymbol{x}, t) = 0, \boldsymbol{x} \in \partial[-1,1]^2 \\ u(\boldsymbol{x}, 0) = -\sin \pi x_1 \sin \pi x_2, \boldsymbol{x} \in [-1,1]^2 \end{cases}.$$

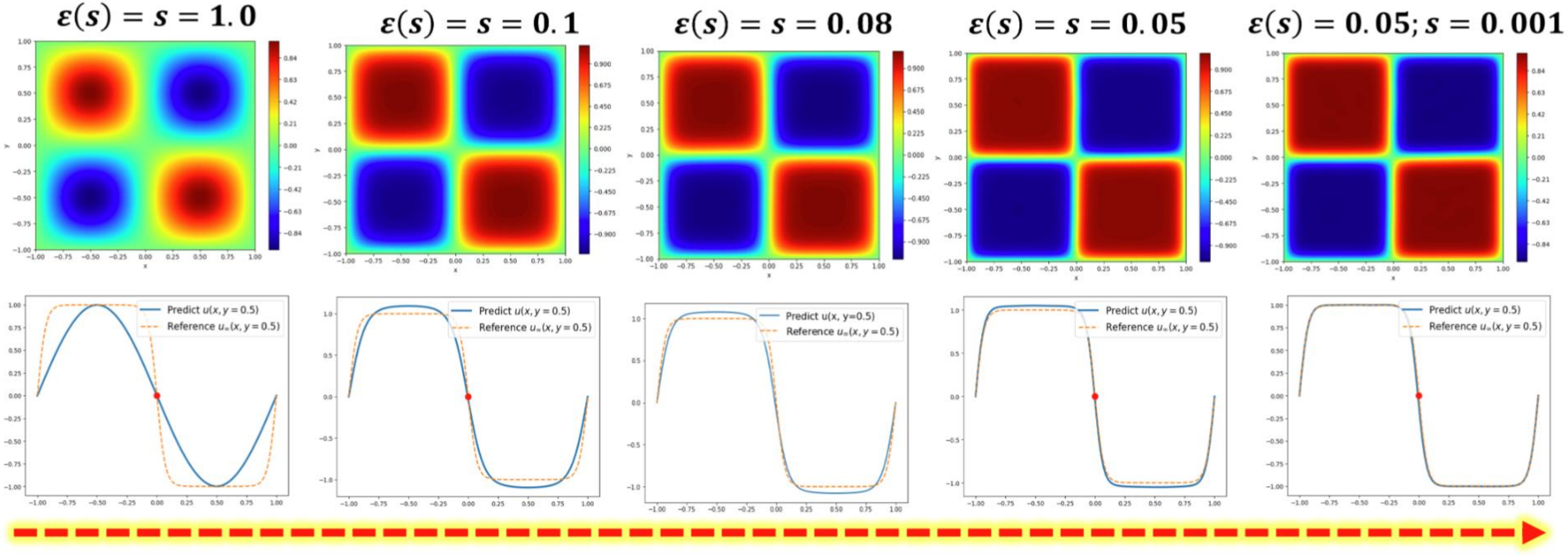


Figure: Steady state solution of Allen-Cahn equation from $\varepsilon = 1 \to 0.05$.

Experimental Results: High dimension Helmholtz equation

- **Homotopy:**

For $\varepsilon : 1 \to 0.02$, u_0 is the solution for $\varepsilon = 1$,

$$H(u, \varepsilon) = \varepsilon^2 \Delta u + \frac{1}{d} u.$$

For high-dimensional Helmholtz equation:

$$\begin{cases} \varepsilon^2 \Delta u + \frac{1}{d} u = 0, & \mathbf{x} \in \Omega, \\ u = g, & \mathbf{x} \in \partial\Omega, \end{cases}$$

where $\Omega = [-1, 1]^d$, which admit the exact solution

$$u(\mathbf{x}) = \sin\left(\frac{1}{d} \sum_{i=1}^{d} \frac{1}{\varepsilon} x_i\right).$$

Dimension $d = 20$	$\varepsilon = 1/2$	$\varepsilon = 1/20$	$\varepsilon = 1/50$
Classical Training	1.23e-3	7.21e-2	9.98e-1
Homotopy dynamics	5.86e-4	5.00e-4	5.89e-4

Table: Comparison of the relative L^2 error achieved by the classical training and homotopy dynamics for different values of ε in Helmholtz equation

Experimental Results: Operator learning Burgers' equation

- **Homotopy:** For $s : 1 \to 0$, $\varepsilon(s)$ is from 1 to 0.05, u_0 is the solution for $\varepsilon = 1$,

$$H(u, s, \varepsilon) = (1 - s)\left(\left(\frac{u^2}{2}\right)_x - \varepsilon(s)u_{xx} - \pi \sin(\pi x)\cos(\pi x)\right) + s(u - u_0),$$

$\varepsilon(s)$ can be set to

$$\varepsilon(s) = \begin{cases} s, & s \in [0.05, 1], \\ 0.05 & s \in [0, 0.05]. \end{cases}$$

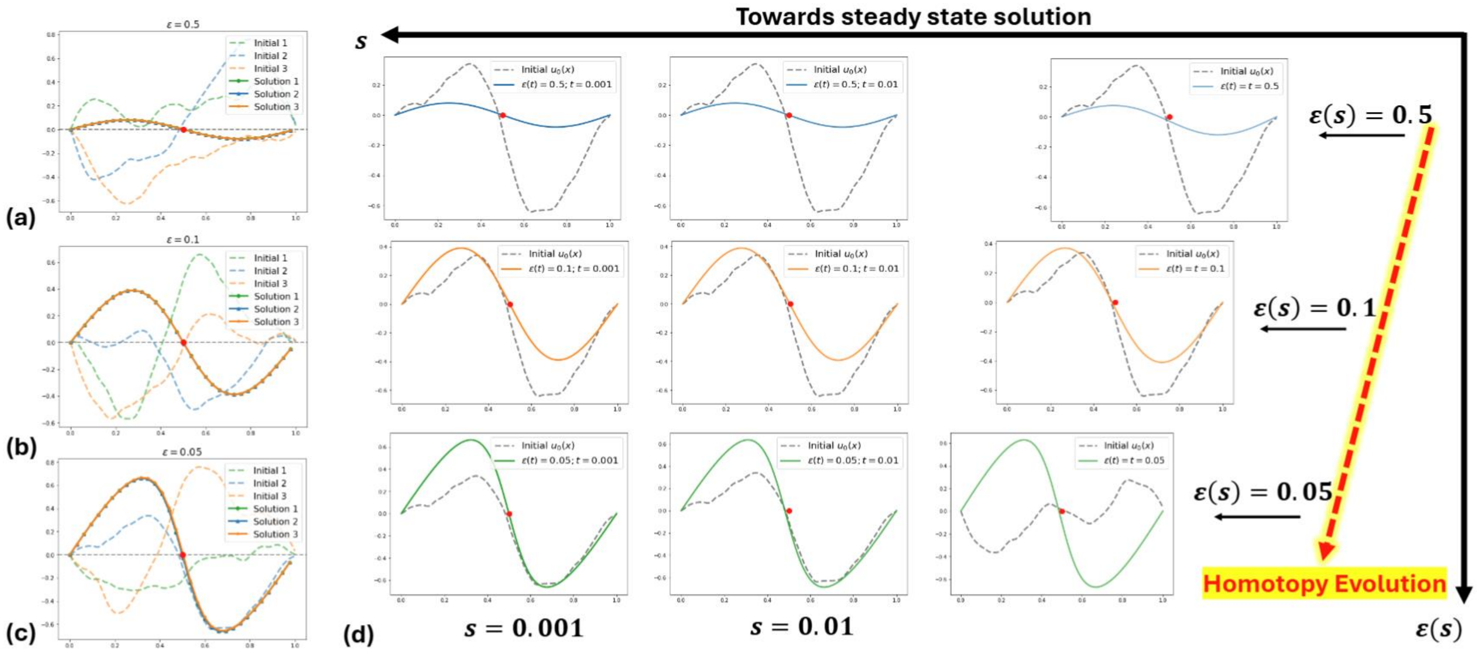


Figure: Steady state solution of Allen-Cahn equation from $\varepsilon = 1 \to 0.05$.

Why Homotopy works?

Homotopy Dynamics based training can provide a **better initialization** for the problem.

Better initialization for problem

$$\begin{cases} 1^2 \cdot u''(x) + u^3 - u = 0, \\ u(0) = -1, \quad u(1) = 1. \end{cases} \Longrightarrow \begin{cases} 0.01^2 \cdot u''(x) + u^3 - u = 0, \quad x \in [0,1], \\ u(0) = -1, \quad u(1) = 1. \end{cases}$$

$$L(\boldsymbol{\theta}(t)) \leq L(\boldsymbol{\theta}(0)) \exp\left(-\frac{\lambda_{\min}(\boldsymbol{K}_\varepsilon)}{n} t\right), \boldsymbol{K}_\varepsilon = \boldsymbol{D}_\varepsilon \boldsymbol{S}\boldsymbol{S}^\top \boldsymbol{D}_\varepsilon^\top.$$

Initialization	Xavier	Hom $\varepsilon = 0.1$	Hom $\varepsilon = 0.05$	Hom $\varepsilon = 0.03$	Hom $\varepsilon = 0.02$
$\lambda_{\min}(\boldsymbol{K}_\varepsilon)$	7.38×10^{-8}	2.11×10^{-6}	7.77×10^{-5}	1.57×10^{-4}	1.48×10^{-2}

Table: Minimum eigenvalue $\lambda_{min}(K_\varepsilon)$ under different initializations for $\varepsilon = 0.01$

Theoretical Results

Small step size $\triangle \boldsymbol{\varepsilon_k}$, and initial value for the initial error for solving large $\boldsymbol{\varepsilon_0}$

Theorem 2 (Convergence of Homotopy Dynamics)

Suppose $h(\varepsilon, u)$, the homotopy operator, is a continuous operator for $0 < \varepsilon_n \leq \varepsilon_0$ and $u \in H^4(\Omega)$, and

$$\|h(u_1, \varepsilon) - h(u_2, \varepsilon)\|_{H^2(\Omega)} \leq K_\varepsilon \|u_1 - u_2\|_{H^2(\Omega)}.$$

Assume there exists a constant K such that $(\varepsilon_k - \varepsilon_{k+1})K_{\varepsilon_k} \leq K \cdot \frac{\varepsilon_0 - \varepsilon_n}{n}$ and

$$\tau := \frac{n}{\varepsilon_0 - \varepsilon_n} \sup_{0 \leq k \leq n} (\varepsilon_k - \varepsilon_{k+1})^2 \|u(\varepsilon_k)\|_{H^4(\Omega)} \ll 1,$$

$$e_0 := \|u(\varepsilon_0) - U(\varepsilon_0)\|_{H^2(\Omega)} \ll 1$$

then we have

Step size $\triangle \boldsymbol{\varepsilon_k}$

$$\|u(\varepsilon_n) - U(\varepsilon_n)\|_{H^2(\Omega)} \leq e_0 e^{K(\varepsilon_0 - \varepsilon_n)} + \frac{\tau(e^{K(\varepsilon_0 - \varepsilon_n)} - 1)}{2K} \ll 1.$$

Initial error solved by original method for large $\boldsymbol{\varepsilon}$

Conclusion

- We theoretically analyze the root cause of training difficulties under small ε.
- To overcome this challenge, we develop a Homotopy Dynamics-based training algorithm.
- Extensive theoretical and empirical results demonstrate the effectiveness of our method in solving singularly perturbed problems.

Paper