

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

ĐỒ ÁN II

**Mạng nơ-ron thông tin vật lý (PINN) trong giải
xấp xỉ các phương trình đạo hàm riêng**

NGUYỄN THÀNH QUANG

MSSV : 20227031

Email : Quang.NT227031@sis.hust.edu.vn

Chuyên ngành : Toán Tin

Giảng viên hướng dẫn: TS. Bùi Xuân Diệu

Chữ ký của GVHD

Tóm tắt đề tài

Đề tài này trình bày các khía cạnh xoay quanh mạng nơ-ron thông tin vật lý (Physics informed neural networks), thường được gọi là PINN. Đây một cách tiếp cận sử dụng các mô hình mạng nơ-ron học sâu để giải xấp xỉ các phương trình đạo hàm riêng (Partial differential equation - PDE). Nội dung được trình bày trong đồ án sẽ đi qua 3 phần chính:

1. Trình bày về động lực, ý tưởng, phương pháp của PINN, kèm các ví dụ cụ thể để áp dụng PINN trong giải xấp xỉ các phương trình đạo hàm riêng (PDE). Tập trung trả lời các câu hỏi : PINN là gì? Làm thế nào một mô hình mạng nơ-ron có thể giải xấp xỉ được PDE?
2. Trình bày phương pháp đo lường sự hiệu quả của mô hình PINN. Tập trung trả lời câu hỏi : Làm sao để biết mô hình PINN có thực sự hiệu quả trong việc giải xấp xỉ PDE?
3. Trình bày về một mô hình PINN được cải tiến, có thể tự động tái cấu trúc mạng nơ-ron dựa trên các ràng buộc từ dữ liệu và PDE. Tập trung trả lời câu hỏi: Làm cách nào để xây dựng một mạng nơ-ron đại diện cho một (bài toán) PDE?

Mục lục

Danh sách hình vẽ	1
Danh sách bảng	2
Mở đầu	4
1 Giới thiệu về PINN	5
1.1 Đặt vấn đề	5
1.2 Khái quát bài toán	5
1.3 Tìm nghiệm dựa trên dữ liệu	6
1.3.1 Mô hình thời gian liên tục	6
1.3.2 Mô hình thời gian rời rạc	9
1.4 Khám phá dựa trên dữ liệu	11
1.4.1 Mô hình thời gian liên tục	11
1.4.2 Mô hình thời gian rời rạc	14
2 Phương pháp đánh giá PINN	18
2.1 Đặt vấn đề	18
2.2 Phương pháp	18
2.3 Áp dụng đánh giá đối với một số trường hợp	19
2.3.1 Cài đặt	19
2.3.2 Ví dụ	19
3 Mô hình PINN tự động khám phá cấu trúc	29
3.1 Đặt vấn đề	29
3.2 Phương pháp	29
3.2.1 Chắt lọc dựa trên thông tin vật lý	30
3.2.2 Phương pháp trích xuất cấu trúc	30
3.2.3 Tái cấu trúc mạng	32
3.2.4 Toàn bộ quá trình triển khai	32
3.3 Kết quả	33
3.3.1 Mạng nơ-ron dựa trên thông tin cấu trúc vật lý.	33
3.3.2 Trích xuất cấu trúc mạng từ PDE.	35
Tổng kết	44
Tài liệu tham khảo	44

Danh sách hình vẽ

1.1	Độ lớn nghiệm $ h(t, x) $ cùng với dữ liệu huấn luyện ban đầu và dữ liệu huấn luyện biên.	8
1.2	So sánh các nghiệm dự đoán và nghiệm chính xác tương ứng tại 3 thời điểm.	8
1.3	Nghiệm $u(t, x)$ tại 2 thời điểm $t = 0.1$ và $t = 0.9$	11
1.4	Dữ liệu đào tạo ban đầu và dự đoán cuối cùng.	11
1.5	Dòng chảy không nén được và xoáy động chảy qua một hình trụ tròn ở $Re = 100$. Dữ liệu đào tạo không-thời gian tương ứng với vùng hình chữ nhật được mô tả trong luồng xoáy hình trụ.	13
1.6	Vị trí của các điểm dữ liệu huấn luyện cho các thành phần vận tốc theo luồng và vận tốc ngang, tương ứng là $u(t, x, y)$ và $v(t, x, y)$. . .	13
1.7	Trường áp suất tức thời $p(t, x, y)$ dự đoán so với trường áp suất tức thời chính xác tại một thời điểm đại diện.	14
1.8	PDE chính xác và PDE được xác định được bằng cách học λ_1, λ_2 và $p(t, x, y)$	14
1.9	Nghiệm $u(t, x)$ tại 2 thời điểm huấn luyện.	16
1.10	Dữ liệu huấn luyện và nghiệm chính xác tương ứng với hai thời điểm.	16
1.11	Phương trình đạo hàm riêng xây dựng từ việc học các tham số λ_1, λ_2	17
2.1	Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Burgers.	22
2.2	Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Fisher.	26
2.3	Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Burgers-Huxley.	27
2.4	Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Newell-Whitehead-Segel.	28
3.1	Thuật toán của mô hình Ψ -NN được đề xuất.	34
3.2	Quy trình của phương pháp Ψ -NN.	34
3.3	a Kết quả phương trình Laplace. b Kết quả phương trình Burgers. c Kết quả phương trình Poisson.	36
3.4	a. Sự tiến triển của các tham số theo phương pháp Ψ -NN. b. Tâm cụm của Ψ -NN trong phương trình Laplace. c. Cấu trúc của mạng học sinh sau khi thay thế tham số trong phương trình Laplace. . . .	37
3.5	So sánh dự đoán theo số lần lặp. a,d: PINN; b,e: PINN-post; c,f: Ψ -NN.	39
3.6	So sánh dự đoán dưới các tham số điều khiển khác nhau của phương trình Burgers. a. $v_1 = 0.01/\pi$. b. $v_2 = 0.04/\pi$. c. $v_3 = 0.08/\pi$	42

Danh sách bảng

2.1	Sai số tuyệt đối của phương trình Burgers	20
2.2	Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Burgers	21
2.3	Sai số tuyệt đối của phương trình Fisher	23
2.4	Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Fisher	23
2.5	Sai số tuyệt đối của phương trình Burgers-Huxley	24
2.6	Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Burgers-Huxley	24
2.7	Sai số tuyệt đối của phương trình Newell-Whitehead-Segel	25
2.8	Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Newell-Whitehead-Segel	25
3.1	So sánh sai số l_2	33
3.2	So sánh điều khiển tham số của phương trình Burgers	41

Mở đầu

Phương trình đạo hàm riêng (PDE) đóng vai trò quan trọng trong việc xây dựng sự hiểu biết của chúng ta về các quá trình cơ bản của thế giới tự nhiên. Ví dụ như sự dao động của chất rắn, dòng chảy của chất lỏng, sự khuếch tán của hóa chất, sự lan truyền nhiệt, cấu trúc của phân tử, sự tương tác giữa photon và electron, hay bức xạ của sóng điện từ. Các phương trình đạo hàm riêng cũng đóng vai trò trung tâm trong toán học hiện đại, đặc biệt là trong hình học và giải tích [1].

Trong những năm gần đây, các phương pháp học máy (ML), đặc biệt là phương pháp học sâu [2], đã mang đến những góc nhìn mới cho mô hình hóa khoa học. ML có thể tự động học các phép ghép nối và ánh xạ phi tuyến giữa các biến từ dữ liệu quan sát, thể hiện tiềm năng to lớn trong việc nhận dạng hệ thống và khám phá phương trình [3], mô hình hóa hệ thống phức tạp [4]. Tuy nhiên, các phương pháp học sâu đều cuối thông thường đòi hỏi một lượng lớn dữ liệu và dễ bị quá khứ, điều này hạn chế khả năng ứng dụng của chúng [5]. Để giải quyết những vấn đề này, mạng nơ-ron thông tin vật lý (PINN) [6] đã xuất hiện, nhưng các ràng buộc vật lý vào mạng nơ-ron (NN) thông qua các hàm mất mát và do đó tích hợp việc học dựa trên dữ liệu với các tiên nghiệm vật lý. PINN cung cấp khả năng mở rộng, tính linh hoạt, khiến chúng trở thành những công cụ mạnh mẽ để giải các phương trình đạo hàm riêng (PDE) và cho thấy triển vọng trong giải các bài toán nghịch. Đồ án này sẽ tập trung trình bày các khía cạnh liên quan đến PINN được quan tâm và nghiên cứu rộng rãi, với 3 phần nội dung chính:

Chương 1: Giới thiệu về PINN

Chương này trình bày về ý tưởng, phương pháp của mô hình PINN, đồng thời kèm theo ví dụ thực tế để làm rõ khả năng xấp xỉ PDE của mô hình PINN. Nội dung chương này dựa trên nghiên cứu của M.Raissi và cộng sự năm 2019 [6], là nghiên cứu đặt nền tảng cho mô hình PINN.

Chương 2: Phương pháp đánh giá PINN

Chương này sẽ trình các phương pháp để đánh giá độ hiệu quả của mô hình PINN thông qua các chỉ số có thể đo lường, sau đó áp dụng với một số phương trình đối lưu - phản ứng - khuếch tán (CRD) tiêu biểu để làm rõ phương pháp. Nội dung chương này dựa trên nghiên cứu của F.Hasan và cộng sự năm 2025 [7].

Chương 3: Mô hình PINN tự động khám phá cấu trúc

Chương này trình bày về mô hình Ψ -NN, một biến thể cải tiến từ mô hình PINN truyền thống, với ý tưởng tự động khám phá và thay đổi cấu trúc của mạng nơ-ron phù hợp với ràng buộc từ dữ liệu và PDE. Nội dung chương này dựa trên nghiên cứu của nhóm nhà khoa học tại Đại học Viện Hàn lâm Khoa học Trung Quốc (UCAS) năm 2025 [8].

Chương 1

Giới thiệu về PINN

1.1 Đặt vấn đề

Sự phát triển của trí tuệ nhân tạo và các mô hình học máy mang lại sự phát triển đột phá trong việc lưu trữ và phân tích dữ liệu, ứng dụng trong nhiều lĩnh vực như quy hoạch giao thông [9], chăm sóc sức khỏe [10], hay các công trình về giải mã gen [11]. Tuy nhiên, trong việc nghiên cứu các quá trình vật lý, sinh học hay nhiều hệ thống phức tạp khác, quá trình thu thập dữ liệu là vô cùng khó khăn, nhiều khi có chi phí rất tốn kém. Do đó, trong nhiều trường hợp, người ta phải giải quyết vấn đề và đưa ra kết luận với thông tin không đầy đủ. Trong tình huống tập dữ liệu nhỏ, phần lớn các mô hình học máy hiện đại (DNN, CNN, RNN) đều không đảm bảo tính hội tụ [6].

Thoạt nhìn thì nhiệm vụ huấn luyện một mô hình học máy để xác định một hàm phi tuyến từ lượng rất nhỏ dữ liệu ban đầu là một việc gần như không thể. Tuy nhiên, trong thực tế có rất nhiều hệ vật lý và sinh học đã được chứng minh là tuân theo các định luật đã biết, hầu hết các mô hình học máy hiện đại đều đang bỏ qua các kiến thức nền tảng này. Việc mã hóa thông tin có cấu trúc từ các định luật này vào một mạng học sâu sẽ khuếch đại nội dung thông tin của dữ liệu mà mô hình nhìn thấy, cho phép mô hình nhanh chóng tự định hướng đến nghiệm đúng và khai quát hóa tốt ngay cả khi chỉ có ít dữ liệu huấn luyện. PINN áp dụng cách tiếp cận sử dụng mạng nơ-ron học sâu và tận dụng khả năng xấp xỉ phổ quát của nó [6]. Đồng thời phương pháp cũng sử dụng vi phân tự động (automatic differentiation) [12] để tính đạo hàm của các mạng nơ-ron theo đầu vào và các tham số mô hình của chúng nhằm thu được các mạng nơ-ron dựa trên thông tin vật lý. Các mạng nơ-ron như vậy bị ràng buộc phải tuân theo bất kỳ quy tắc bắt nguồn từ các định luật vật lý (được mô hình hóa bằng các phương trình đạo hàm riêng phi tuyến) chi phối dữ liệu quan sát (như quy tắc đối xứng, quy tắc bất biến hoặc nguyên lý bảo toàn). Cấu trúc đơn giản nhưng mạnh mẽ này giải quyết nhiều vấn đề trong khoa học tính toán và giới thiệu một công nghệ có khả năng biến đổi, dẫn đến sự phát triển của lớp phương pháp mới để giải PDE, cũng như phương pháp tiếp cận dữ liệu mới giải các bài toán nghịch [6].

1.2 Khái quát bài toán

Xét dạng tổng quát của PDE phi tuyến tham số hóa :

$$u_t + \mathcal{N}(u; \lambda) = 0, \quad x \in \Omega, t \in [0, T], \quad (1.1)$$

trong đó $u(x, t)$ là nghiệm cần tìm, $\mathcal{N}(\cdot; \lambda)$ là toán tử phi tuyến tham số λ , miền xác định $\Omega \subset \mathbb{R}^d$.

Mục tiêu là giải quyết 2 lớp bài toán :

- Tìm nghiệm dựa trên dữ liệu : Tìm nghiệm của PDE dựa trên dữ liệu đã có. Tập trung vào câu hỏi: với các dữ liệu đã có, có thể nói gì về hàm ẩn $u(t, x)$ của hệ thống với tham số λ cố định cho trước?
- Khám phá mô hình dựa trên dữ liệu : Xác định dạng phương trình từ dữ liệu quan sát. Tập trung vào câu hỏi: liệu tham số λ nào thích hợp để mô tả dữ liệu quan sát được?

Với 2 cách tiếp cận khác nhau cho mỗi lớp bài toán:

- Mô hình thời gian liên tục.
- Mô hình thời gian rời rạc.

1.3 Tìm nghiệm dựa trên dữ liệu

1.3.1 Mô hình thời gian liên tục

Đầu tiên ta định nghĩa hàm $f(x, t)$ như sau:

$$f := u_t + \mathcal{N}[u], \quad (1.2)$$

sau đó xấp xỉ $u(x, t)$ bởi mạng nơ-ron. Kết hợp kết quả này với phương trình (1.2) ta có kết quả là mạng nơ-ron thông tin vật lý (PINN) $f(x, t)$. Mạng này có thể tính toán thông qua việc áp dụng chuỗi quy tắc cho các thành phần của hàm bằng vi phân tự động (automatic differentiation) [12], và mạng này cũng có các tham số giống với mạng xấp xỉ $u(x, t)$ nhưng có các hàm kích hoạt khác nhau do ảnh hưởng của toán tử phi tuyến \mathcal{N} . Các tham số chung của 2 mạng $u(x, t)$ và $f(x, t)$ có thể được học bằng cách tối thiểu hóa sai số bình phương trung bình MSE:

$$MSE = MSE_u + MSE_f,$$

với 2 sai số thành phần :

$$MSE_u = \frac{1}{N_u} \sum_{i=1}^{N_u} |u(x_u^i, t_u^i) - u^i|^2,$$

thể hiện cho dữ liệu ban đầu và biên, với $\{x_u^i, t_u^i, u^i\}_{i=1}^{N_u}$ là các điểm huấn luyện dữ liệu ban đầu và điều kiện biên, và :

$$MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |f(x_f^i, t_f^i)|^2,$$

thể hiện cho cấu trúc được mô tả bởi phương trình (1.1) cho tập điểm hữu hạn, với $\{x_f^i, t_f^i\}_{i=1}^{N_f}$ là các điểm trong cụ thể cho mạng $f(x, t)$.

Trong các ví dụ ở phần 1.3, tổng số dữ liệu huấn luyện là tương đối nhỏ (vài trăm đến vài nghìn điểm), do đó việc tối ưu hóa tất cả các hàm mất mát sẽ sử dụng L-BFGS, một thuật toán tối ưu hóa tựa Newton [13]. Đối với các tập dữ liệu lớn hơn, chẳng hạn như các ví dụ ở phần 1.4, ta sẽ sử dụng kết hợp L-BFGS và Adam [14], một biến thể cải tiến của thuật toán hướng giảm gradient.

Ví dụ: Phương trình Schrodinger

Fương trình Schrodinger một chiều phi tuyến là một phương trình trường cổ điển được sử dụng để nghiên cứu các hệ cơ học lượng tử, bao gồm sự lan truyền sóng phi tuyến trong sợi quang và/hoặc ống dẫn sóng, ngưng tụ Bose-Einstein và sóng plasma [6].

Ví dụ này nhằm thể hiện khả năng của PINN trong việc xử lý các điều kiện biên tuần hoàn, các nghiệm phức, cũng như các loại phi tuyến khác nhau trong PDE. Phương trình Schrodinger phi tuyến cùng với các điều kiện biên tuần hoàn được cho như sau:

$$ih_t + 0.5h_{xx} + |h|^2h = 0, \quad x \in [-5, 5], \quad t \in [0, \frac{\pi}{2}], \quad (1.3)$$

với điều kiện biên tuần hoàn :

$$\begin{cases} h(5, t) = h(-5, t), \\ h_x(5, t) = h_x(-5, t), \end{cases}$$

và điều kiện ban đầu :

$$h(0, x) = 2 \operatorname{sech}(x),$$

ở đây $h(t, x)$ là hàm nhận giá trị phức. Ta định nghĩa hàm $f(t, x)$ như sau:

$$f := ih_t + 0.5h_{xx} + |h|^2|h|, \quad (1.4)$$

tiếp theo ta xấp xỉ h bằng mạng nơ-ron đa đầu ra giá trị phức $h(t, x) = [u(t, x) \ v(t, x)]$, với $u(t, x)$ là đầu ra biểu diễn phần thực và $v(t, x)$ là đầu ra biểu diễn phần ảo. Kết quả này giúp ta thu được mạng nơ-ron thông tin vật lý giá trị phức $f(t, x)$. Các mạng $f(t, x)$ và $h(t, x)$ có chung tham số và các tham số này được học thông qua tối thiểu hóa sai số bình phương trung bình:

$$MSE = MSE_0 + MSE_b + MSE_f, \quad (1.5)$$

với các thành phần sau:

$$MSE_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} |h(0, x_0^i) - h_0^i|^2,$$

thể hiện sai số của dữ liệu ban đầu, với $\{x_0^i, h_0^i\}_{i=1}^{N_0}$ là các điểm dữ liệu ban đầu,

$$MSE_b = \frac{1}{N_b} \sum_{i=1}^{N_b} (|h(t_b^i, 5) - h(t_b^i, -5)|^2 + |h_x(t_b^i, 5) - h_x(t_b^i, -5)|^2),$$

thể hiện sai số của dữ liệu biên, với $\{t_b^i\}_{i=1}^{N_b}$ là các điểm dữ liệu trên biên,

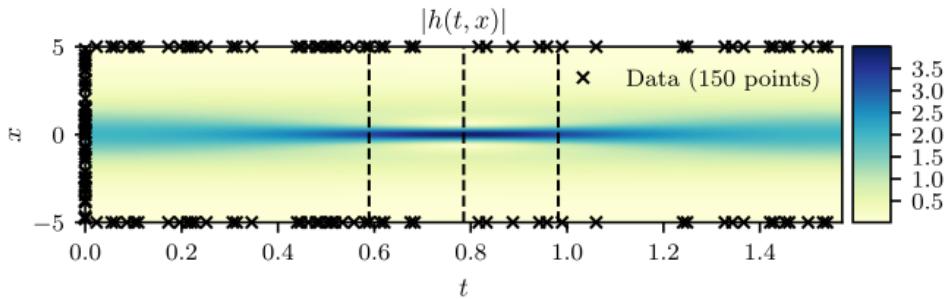
$$MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2,$$

thể hiện sai số của mạng $f(t, x)$, với $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ là các điểm dữ liệu ngẫu nhiên trong miền xác định.

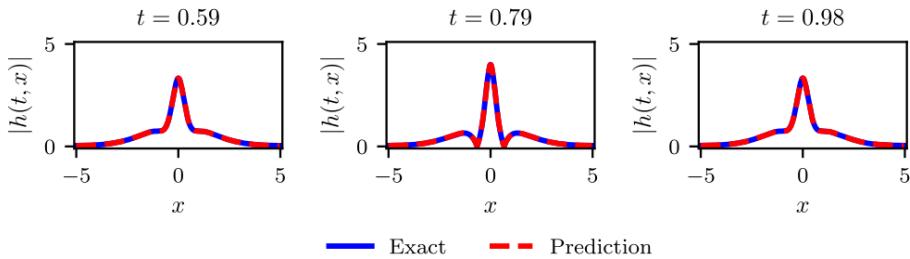
Trong ví dụ, số dữ liệu quan sát tại thời điểm ban đầu, biên và bên trong miền nghiệm như sau: $N_0 = 50, N_b = 50, N_f = 20000$. Các điểm được chọn ngẫu nhiên

theo chiến lược Latin Hypercube Sampling [15], một phương pháp lấy mẫu ngẫu nhiên có cấu trúc từ không gian nhiều chiều, đảm bảo phân bố đều trên mỗi chiều. Ta biểu diễn hàm ẩn $h(t, x) = [u(t, x) \ v(t, x)]$, sử dụng 5 lớp mạng nơ-ron với 100 nơ-ron mỗi lớp và hàm kích hoạt tanh.

Trong ví dụ, PINN còn thể hiện khả năng chống lại vấn đề quá khớp (overfitting) nổi tiếng trong huấn luyện mô hình học sâu. Cụ thể, MSE_f trong phương trình (1.5) hoạt động như một cơ chế chính quy hóa, phạt các nghiệm không thỏa mãn phương trình (1.4). Do đó, một đặc tính quan trọng của PINN là chúng có thể được huấn luyện hiệu quả bằng cách sử dụng các tập dữ liệu nhỏ; một bối cảnh thường gặp trong nghiên cứu các hệ thống vật lý mà chi phí thu thập dữ liệu có thể rất cao [6].



Hình 1.1: Độ lớn nghiệm $|h(t, x)|$ cùng với dữ liệu huấn luyện ban đầu và dữ liệu huấn luyện biên.



Hình 1.2: So sánh các nghiệm dự đoán và nghiệm chính xác tương ứng tại 3 thời điểm.

Hình 1.1 cho thấy độ lớn của nghiệm dự đoán $|h(t, x)| = \sqrt{u(t, x)^2 + v(t, x)^2}$, cùng với vị trí của dữ liệu huấn luyện ban đầu và dữ liệu huấn luyện biên. Sai số dự đoán kết quả được xác thực dựa trên dữ liệu kiểm tra cho bài toán này và được đo ở mức $1,97 \cdot 10^{-3}$ trong chuẩn l_2 tương đối. Dánh giá chi tiết hơn về nghiệm dự đoán được trình bày trong hình 1.2, trình bày so sánh giữa nghiệm chính xác và nghiệm dự đoán tại các thời điểm khác nhau $t = 0.59, 0.79, 0.98$. Chỉ sử dụng một số ít dữ liệu ban đầu, PINN có thể nắm bắt chính xác toán tử phi tuyến phức tạp của phương trình Schrodinger.

Một hạn chế tiềm ẩn của các mô hình thời gian liên tục xuất phát từ việc sử dụng một số lượng lớn các điểm ngẫu nhiên N_f để áp đặt các ràng buộc trong toàn bộ miền không-thời gian. Mặc dù điều này không gây ra vấn đề đáng kể nào cho các bài toán một hoặc hai chiều không gian, nhưng lại có thể gây ra vấn đề nghiêm trọng trong các bài toán nhiều chiều hơn, vì tổng số điểm cần thiết sẽ tăng theo cấp số nhân. Trong phần tiếp theo, một cách tiếp cận khác giúp loại bỏ nhu

cầu về các điểm ngẫu nhiên bằng cách giới thiệu một mô hình rời rạc, tận dụng các lược đồ Runge-Kutta [16].

1.3.2 Mô hình thời gian rời rạc

Áp dụng dạng tổng quát của lược đồ Runge-Kutta bậc q [16] cho phương trình (1.1), ta có :

$$\begin{aligned} u^{n+c_i} &= u^n - \Delta t \sum_{j=1}^q a_{ij} \mathcal{N}[u^{n+c_j}], \quad i = 1, \dots, q, \\ u^{n+1} &= u^n - \Delta t \sum_{j=1}^q b_j \mathcal{N}[u^{n+c_j}], \end{aligned} \quad (1.6)$$

ở đây :

$$u^{n+c_j}(x) = u(t^n + c_j \Delta t, x), j = 1, \dots, q.$$

Dạng tổng quát này bao hàm cả lược đồ bước thời gian tưởng minh và ngầm định, hệ phương trình (1.6) có thể biểu diễn lại như sau :

$$\begin{aligned} u^n &= u_i^n, i = 1, \dots, q, \\ u^n &= u_{q+1}^n, \end{aligned} \quad (1.7)$$

trong đó :

$$\begin{aligned} u_i^n &= u^{n+c_i} + \Delta t \sum_{j=1}^q a_{ij} \mathcal{N}[u^{n+c_j}], i = 1, \dots, q, \\ u_{q+1}^n &= u^{n+1} + \Delta t \sum_{j=1}^q b_j \mathcal{N}[u^{n+c_j}]. \end{aligned} \quad (1.8)$$

Tiếp tục đặt 1 mạng nơ-ron đa đầu ra:

$$[u^{n+c_1}(x), u^{n+c_2}(x), \dots, u^{n+c_q}(x), u^{n+1}(x)], \quad (1.9)$$

các đầu ra này kết hợp với phương trình (1.8) cho kết quả là mạng PINN nhận x là đầu vào và có đầu ra :

$$[u_1^n(x), u_2^n(x), \dots, u_q^n(x), u_{q+1}^n(x)]. \quad (1.10)$$

Ví dụ : Phương trình Allen-Cahn

Phương trình Allen-Cahn là một phương trình nổi tiếng trong lĩnh vực hệ thống phản ứng khuếch tán. Nó mô tả quá trình phân tách pha trong các hệ hợp kim đa thành phần, bao gồm cả các chuyển đổi trật tự-mất trật tự [6].

Ví dụ này nhằm mục đích làm nổi bật khả năng của các mô hình thời gian rời rạc trong việc xử lý các loại toán tử phi tuyến khác nhau của PDE. Ta xét phương trình Allen-Cahn cùng với các điều kiện biên tuần hoàn:

$$u_t - 0.0001u_{xx} + 5u^3 - 5u = 0, \quad x \in [-1, 1], t \in [0, 1], \quad (1.11)$$

với điều kiện biên tuần hoàn :

$$u(1, t) = u(-1, t) \quad ; \quad u_x(1, t) = u_x(-1, t),$$

và điều kiện ban đầu :

$$u(0, x) = x^2 \cos(\pi x).$$

Lúc này toán tử phi tuyến trong phương trình (1.8) sẽ là :

$$\mathcal{N}[u^{n+c_j}] = -0.0001u_{xx}^{n+c_j} + 5(u^{n+c_j})^3 - 5u^{n+c_j},$$

các tham số chung của 2 mạng (1.9) và (1.10) sẽ được học bằng cách tối thiểu hóa sai số bình phương tổng SSE:

$$SSE = SSE_n + SSE_b \quad (1.12)$$

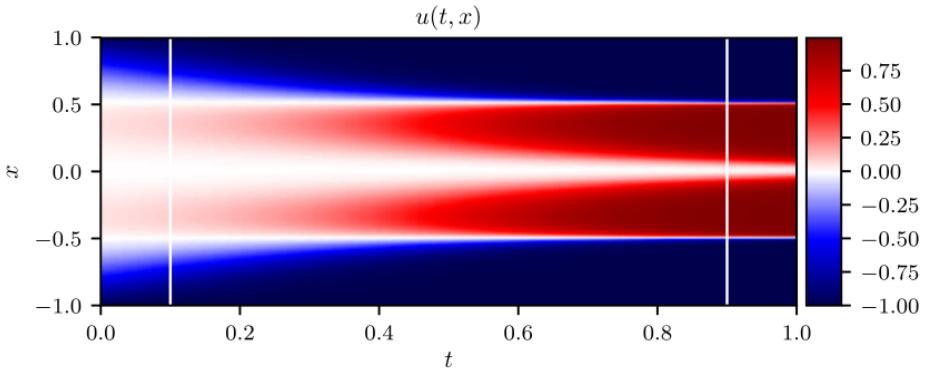
trong đó :

$$\begin{aligned} SSE_n &= \sum_{j=1}^{q+1} \sum_{i=1}^{N_n} |u_j^n(x^{n,i}) - u^{n,i}|^2, \\ SSE_b &= \sum_{i=1}^q |u^{n+c_i}(-1) - u^{n+c_i}(1)|^2 + |u^{n+1}(-1) - u^{n+1}(1)|^2 \\ &\quad + \sum_{i=1}^q |u_x^{n+c_i}(-1) - u_x^{n+c_i}(1)|^2 + |u_x^{n+1}(-1) - u_x^{n+1}(1)|^2, \end{aligned}$$

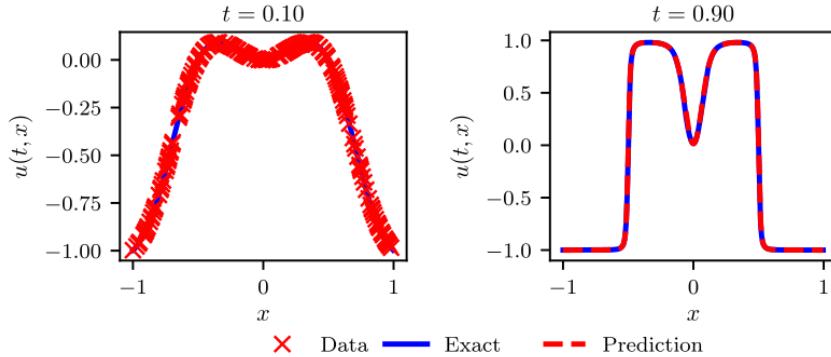
ở đây $\{x^{n,i}, u^{n,i}\}_{i=1}^{N_n}$ là điểm dữ liệu tại bước thời gian t^n . Trong phương pháp số cổ điển, các bước thời gian này thường bị giới hạn ở mức nhỏ do các ràng buộc về tính ổn định đối với các lược đồ tường minh hoặc các ràng buộc về độ phức tạp tính toán đối với các lược đồ ngầm định [16]. Những ràng buộc này trở nên quan trọng hơn khi số bậc Runge-Kutta q tăng lên, và đối với hầu hết các bài toán thực tế, người ta cần thực hiện hàng nghìn đến hàng triệu bước như vậy cho đến khi giải được nghiệm đến thời điểm cuối cùng mong muốn. Trái ngược hoàn toàn với các phương pháp cổ điển, PINN có thể sử dụng các lược đồ Runge-Kutta ngầm định với số lượng giai đoạn lớn tùy ý với chi phí bổ sung rất nhỏ. Điều này cho phép chúng ta thực hiện các bước thời gian rất lớn trong khi vẫn duy trì tính ổn định và độ chính xác dự đoán cao, do đó cho phép tìm nghiệm trong một bước duy nhất.

Trong ví dụ này, ta tạo ra một tập dữ liệu huấn luyện và kiểm tra bằng cách mô phỏng phương trình Allen-Cahn (1.11) bằng cách sử dụng các phương pháp phổ thông thường. Cụ thể, bắt đầu từ điều kiện ban đầu $u(0, x) = x^2 \cos(\pi x)$ và giả sử các điều kiện biên tuần hoàn $u(t, -1) = u(t, 1)$ và $u_x(t, -1) = u_x(t, 1)$, ta tích phân phương trình (1.11) đến thời điểm cuối cùng $t = 1.0$ sử dụng gói Chebfun [17] với phép rời rạc Fourier phổ với 512 chế độ và bộ tích phân thời gian Runge-Kutta bậc bốn tường minh với bước thời gian $\Delta t = 10^{-5}$.

Dữ liệu cho ví dụ bao gồm $N_n = 200$ điểm được lấy ngẫu nhiên từ nghiệm chính xác tại $t = 0.1$. Mục tiêu là dự đoán nghiệm tại $t = 0.9$ với bước thời gian $\Delta t = 0.8$. Ví dụ sử dụng mạng PINN 4 lớp ẩn, 200 nơ-ron mỗi lớp, và lớp đầu ra gồm 101 giá trị, bao gồm 100 bậc Runge-Kutta $u^{n+c_j}(x)$, $j = 1, \dots, 100$, và nghiệm tại thời điểm cuối cùng $u^{n+1}(x)$. Bắt đầu từ dữ liệu ban đầu tại $t = 0.1$, PINN có thể dự đoán nghiệm gần như không liên tục tại $t = 0.9$ (Hình 1.3) trong một bước thời gian duy nhất với sai số l_2 tương đối là $6, 99 \cdot 10^{-3}$, như minh họa trong Hình 1.4. Sai số này hoàn toàn là do khả năng của mạng nơ-ron trong việc ước lượng $u(t, x)$, cũng như mức độ mà tổng bình phương sai số cho phép nội suy dữ liệu huấn luyện.



Hình 1.3: Nghiệm $u(t, x)$ tại 2 thời điểm $t = 0.1$ và $t = 0.9$.



Hình 1.4: Dữ liệu đào tạo ban đầu và dự đoán cuối cùng.

1.4 Khám phá dựa trên dữ liệu

1.4.1 Mô hình thời gian liên tục

Ta định nghĩa hàm f như sau:

$$f(x, t) = u_t + \mathcal{N}[\lambda; t], \quad (1.13)$$

sau đó tiếp tục xấp xỉ u bởi mạng học sâu, rồi sử dụng kết quả cùng phương trình (1.13) để xấp xỉ mạng PINN $f(t, x)$. Lưu ý lúc này tham số λ trở thành tham số của mạng $f(t, x)$.

Ví dụ : Phương trình Navier-Stokes

Các phương trình Navier-Stokes mô tả vật lý của nhiều hiện tượng khoa học và kỹ thuật. Chúng có thể được sử dụng để mô hình hóa thời tiết, dòng hải lưu, dòng chảy của nước trong ống và chuyển động của không khí xung quanh cánh máy bay. Các phương trình Navier-Stokes ở dạng đầy đủ và đơn giản hóa giúp thiết kế máy bay và ô tô, nghiên cứu dòng chảy của máu, thiết kế nhà máy điện, phân tích sự phân tán của chất ô nhiễm và nhiều ứng dụng khác [6]. Ví dụ tiếp theo xem xét các phương trình Navier-Stokes trong hai chiều (2D) được đưa ra một cách tường minh như sau:

$$\begin{aligned} u_t + \lambda_1(uu_x + vu_y) &= -p_x + \lambda_2(u_{xx} + u_{yy}) \\ v_t + \lambda_1(uv_x + vv_y) &= -p_y + \lambda_2(v_{xx} + v_{yy}), \end{aligned} \quad (1.14)$$

trong đó $u(t, x, y)$ mô tả vận tốc theo trục x, $v(t, x, y)$ mô tả vận tốc theo trục y, $p(t, x, y)$ mô tả áp suất, $\lambda = [\lambda_1, \lambda_2]$ là các tham số chưa biết. Nghiệm của phương

trình thuộc tập các hàm không phân kì.

$$u_x + v_y = 0, \quad (1.15)$$

phương trình bổ sung này là phương trình liên tục cho chất lỏng không nén được, mô tả sự bảo toàn khối lượng của chất lỏng.

Ngoài ra, ta tạo hàm ẩn $\psi(t, x, y)$ và giả sử:

$$u = \psi_y, \quad v = -\psi_x, \quad (1.16)$$

từ đó phương trình (1.15) luôn được thỏa mãn.

Cho các phép đo nhiều $\{t^i, x^i, y^i, u^i, v^i\}_{i=1}^N$ của trường vận tốc. Tập trung vào việc học các tham số λ cũng như áp suất $p(t, x, y)$.

Định nghĩa 2 hàm $f(t, x, y)$ và $g(t, x, y)$ như sau:

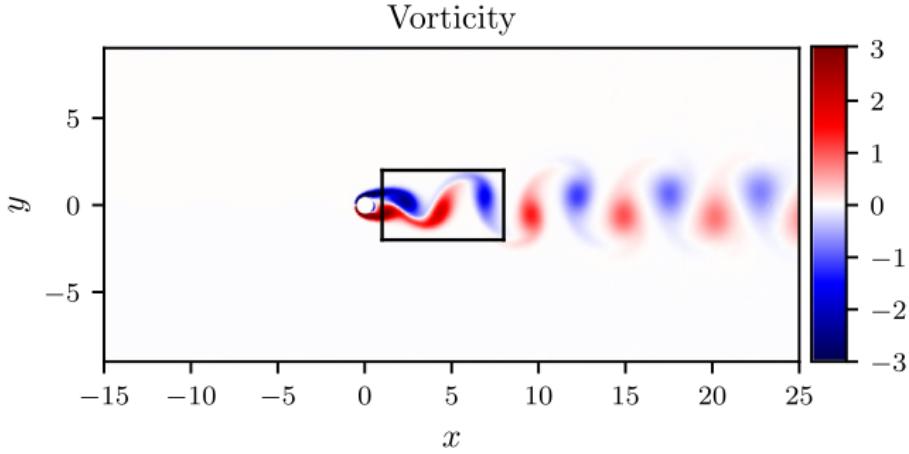
$$\begin{aligned} f &:= u_t + p_x + \lambda_1(uu_x + vu_y) - \lambda_2(u_{xx} + u_{yy}), \\ g &:= v_t + p_y + \lambda_1(uv_x + vv_y) - \lambda_2(v_{xx} + v_{yy}), \end{aligned} \quad (1.17)$$

và tiếp tục xấp xỉ $[\psi(t, x, y) \ p(t, x, y)]$ bằng mạng nơ-ron đơn hai đầu ra. Kết quả này kết hợp với các phương trình (1.16) và (1.17) thu được mạng PINN $[f(t, x, y) \ g(t, x, y)]$. Tham số λ của toán tử phi tuyến cũng như các tham số của các mạng nơ-ron được học bằng cách tối thiểu sai số bình phương trung bình MSE:

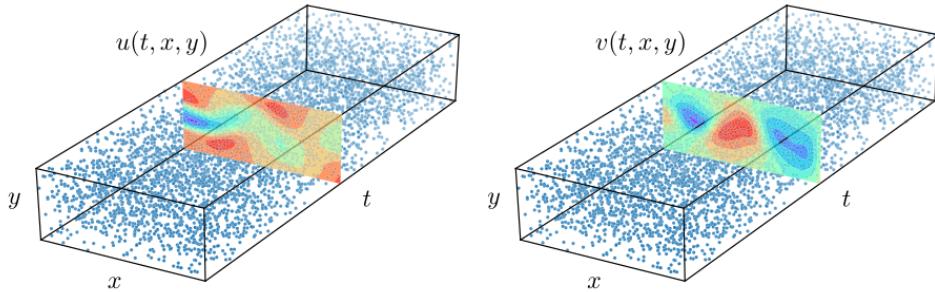
$$\begin{aligned} MSE &:= \frac{1}{N} \sum_{i=1}^N (|u(t^i, x^i, y^i) - u^i|^2 + |v(t^i, x^i, y^i) - v^i|^2) \\ &\quad + \frac{1}{N} \sum_{i=1}^N (|f(t^i, x^i, y^i)|^2 + |g(t^i, x^i, y^i)|^2) \end{aligned} \quad (1.18)$$

Ví dụ giả định một biên dạng vận tốc dòng chảy tự do đồng nhất được áp đặt tại ranh giới bên trái, một điều kiện dòng chảy ra áp suất bằng 0 được áp đặt tại ranh giới bên phải nằm cách hình trụ 25 lần đường kính về phía hạ lưu, và tính tuần hoàn cho các ranh giới trên cùng và dưới cùng của miền $[-15, 25] \times [-8, 8]$. Sau đó tích phân phương trình (1.14) bằng cách sử dụng lược đồ ổn định bậc ba cho đến khi hệ thống đạt đến trạng thái ổn định tuần hoàn, như được mô tả trong Hình 1.5. Tiếp theo, một phần nhỏ của tập dữ liệu kết quả tương ứng với nghiệm trạng thái ổn định này sẽ được sử dụng để huấn luyện mô hình, trong khi phần dữ liệu còn lại sẽ được sử dụng để xác thực các dự đoán của ví dụ. Để đơn giản, ta sẽ chọn giới hạn việc lấy mẫu trong một vùng hình chữ nhật thuận dòng của hình trụ như thể hiện trong Hình 1.5.

Với dữ liệu phân tán và có khả năng bị nhiễu trên các thành phần vận tốc $u(t, x, y)$ theo luồng và $v(t, x, y)$ theo chiều ngang, mục tiêu của ví dụ là xác định các tham số chưa biết λ_1 và λ_2 , cũng như thu được bản tái tạo chính xác về mặt định tính toàn bộ trường áp suất $p(t, x, y)$ trong luồng xoáy hình trụ, mà theo định nghĩa chỉ có thể được xác định đến một hằng số. Để đạt được mục đích này, ví dụ đã tạo ra một tập dữ liệu huấn luyện bằng cách lấy mẫu ngẫu nhiên toàn bộ tập dữ liệu có độ phân giải cao. Để làm nổi bật khả năng học từ dữ liệu huấn luyện phân tán và ít ỏi của phương pháp, ví dụ đã chọn $N = 5000$, tương ứng với chỉ 1% tổng dữ liệu khả dụng như minh họa trong Hình 1.6. Ngoài ra, biểu đồ cũng thể hiện các ảnh chụp nhanh đại diện của các thành phần vận tốc dự đoán $u(t, x, y)$ và $v(t, x, y)$ sau khi mô hình được huấn luyện. Kiến trúc mạng nơ-ron được sử dụng ở đây bao gồm 9 lớp với 20 nơ-ron trong mỗi lớp.



Hình 1.5: Dòng chảy không nén được và xoáy động chảy qua một hình trụ tròn ở $Re = 100$. Dữ liệu đào tạo không-thời gian tương ứng với vùng hình chữ nhật được mô tả trong luồng xoáy hình trụ.



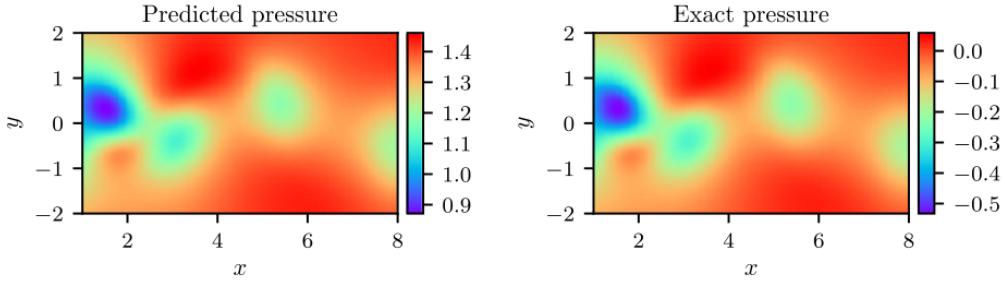
Hình 1.6: Vị trí của các điểm dữ liệu huấn luyện cho các thành phần vận tốc theo luồng và vận tốc ngang, tương ứng là $u(t, x, y)$ và $v(t, x, y)$.

Tóm tắt kết quả của thí nghiệm được trình bày trong Hình 1.8. Có thể quan sát thấy PINN có thể xác định chính xác các tham số chưa biết λ_1 và λ_2 với độ chính xác rất cao ngay cả khi dữ liệu huấn luyện bị nhiễu. Cụ thể, đối với trường hợp dữ liệu huấn luyện không nhiễu, sai số khi ước tính λ_1 và λ_2 lần lượt là 0,078% và 4,67%. Các dự đoán vẫn mạnh mẽ ngay cả khi dữ liệu huấn luyện bị nhiễu 1%, trả về sai số lần lượt là 0,17% và 5,70% cho λ_1 và λ_2 .

Một kết quả thú vị hơn xuất phát từ khả năng của mạng lưới trong việc cung cấp dự đoán chính xác về mặt định tính của toàn bộ trường áp suất $p(t, x, y)$ khi không có bất kỳ dữ liệu huấn luyện nào về chính áp suất đó. Hình 1.7 trình bày một so sánh trực quan với nghiệm áp suất chính xác để có một ảnh chụp nhanh của áp suất đại diện. Lưu ý rằng sự khác biệt về độ lớn giữa áp suất chính xác và áp suất dự đoán được chứng minh bởi bản chất của hệ thống Navier-Stokes không nén được, vì trường áp suất chỉ có thể nhận dạng được đến một hằng số. Kết quả này suy ra từ một lượng liên tục các phép đo phụ trợ bằng cách tận dụng các định luật vật lý cơ bản là một ví dụ về khả năng PINN mang lại, đồng thời làm nổi bật tiềm năng của chúng trong việc giải quyết các bài toán nghịch đa chiều.

Cách tiếp cận này giả định tính khả dụng của dữ liệu phân tán trên toàn bộ miền không-thời gian. Tuy nhiên, trong nhiều trường hợp thực tế, người ta có thể chỉ quan sát được hệ thống tại các thời điểm riêng biệt. Trong phần tiếp theo, ta đi tới một cách tiếp cận khác để giải quyết vấn đề khám phá dựa trên dữ liệu chỉ

bằng cách sử dụng hai điểm dữ liệu. Ở đó ta sẽ xem xét cách tận dụng các lược đồ bước thời gian Runge-Kutta cổ điển, từ đó có thể xây dựng các mạng nơ-ron rời rạc dựa trên vật lý, giúp duy trì độ chính xác dự đoán cao ngay cả khi khoảng cách thời gian giữa các thời điểm của dữ liệu là rất lớn.



Hình 1.7: Trường áp suất tức thời $p(t, x, y)$ dự đoán so với trường áp suất tức thời chính xác tại một thời điểm đại diện.

Correct PDE	$u_t + (uu_x + vu_y) = -p_x + 0.01(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.01(v_{xx} + v_{yy})$
Identified PDE (clean data)	$u_t + 0.999(uu_x + vu_y) = -p_x + 0.01047(u_{xx} + u_{yy})$ $v_t + 0.999(uv_x + vv_y) = -p_y + 0.01047(v_{xx} + v_{yy})$
Identified PDE (1% noise)	$u_t + 0.998(uu_x + vu_y) = -p_x + 0.01057(u_{xx} + u_{yy})$ $v_t + 0.998(uv_x + vv_y) = -p_y + 0.01057(v_{xx} + v_{yy})$

Hình 1.8: PDE chính xác và PDE được xác định được bằng cách học λ_1, λ_2 và $p(t, x, y)$

1.4.2 Mô hình thời gian rời rạc

Áp dụng dạng tổng quát của phương pháp Runge-Kutta bậc q cho phương trình (1.1), ta có:

$$\begin{aligned} u^{n+c_i} &= u^n - \Delta t \sum_{j=1}^q a_{ij} \mathcal{N}[u^{n+c_j}; \lambda], \quad i = 1, \dots, q, \\ u^{n+1} &= u^n - \Delta t \sum_{j=1}^q b_j \mathcal{N}[u^{n+c_j}; \lambda]. \end{aligned} \tag{1.19}$$

Ở đây, $u^{n+c_j}(x) = u(t_n + c_j \Delta t, x)$, $j = 1, \dots, q$, là trạng thái ẩn của hệ đang xét tại thời điểm $t = t_n + c_j \Delta t$. Dạng tổng quát này bao hàm cả lược đồ bước thời gian ngầm định lẩn tưởng minh, phụ thuộc vào cách chọn tham số a_{ij}, b_j, c_j . Phương trình (1.19) tương đương với:

$$\begin{aligned} u^n &= u_i^n, \quad i = 1, \dots, q, \\ u^{n+1} &= u_i^{n+1}, \quad i = 1, \dots, q, \end{aligned} \tag{1.20}$$

trong đó :

$$\begin{aligned} u_i^n &= u^{n+c_i} + \Delta t \sum_{j=1}^q a_{ij} \mathcal{N}[u^{n+c_j}; \lambda], \quad i = 1, \dots, q, \\ u_i^{n+1} &= u^{n+c_i} + \Delta t \sum_{j=1}^q (a_{ij} - b_j) \mathcal{N}[u^{n+c_j}; \lambda]. \end{aligned} \quad (1.21)$$

Tiếp tục đặt mạng nơ-ron đa đầu ra:

$$[u^{n+c_1}, u_{n+c_2}, \dots, u_{n+c_q}], \quad (1.22)$$

kết hợp đầu ra (1.22) với hệ phương trình (1.21), ta có kết quả là 2 mạng PINN:

$$[u_1^n, u_2^n, \dots, u_q^n], \quad (1.23)$$

và :

$$[u_1^{n+1}, u_2^{n+1}, \dots, u_q^{n+1}]. \quad (1.24)$$

Với kết quả các phép đo nhiễu $\{x^n, u^n\}$ và $\{x^{n+1}, u^{n+1}\}$ tại 2 thời điểm khác nhau t^n và t^{n+1} , ta có thể học các tham số chung của 3 mạng (1.22) (1.23) (1.24) cùng với tham số λ thông qua tối thiểu sai số bình phương tổng SSE:

$$SSE = SSE_n + SSE_{n+1} \quad (1.25)$$

trong đó :

$$SSE_n = \sum_{j=1}^q \sum_{i=1}^{N_n} |u_j^n(x^{n,i}) - u^{n,i}|^2,$$

và :

$$SSE_{n+1} = \sum_{j=1}^q \sum_{i=1}^{N_{n+1}} |u_j^{n+1}(x^{n+1,i}) - u^{n+1,i}|^2.$$

Ở đây : $x_n = \{x^{n,i}\}_{i=1}^{N_n}$, $u_n = \{u^{n,i}\}_{i=1}^{N_n}$, $x_{n+1} = \{x^{n+1,i}\}_{i=1}^{N_{n+1}}$, $u_{n+1} = \{u^{n+1,i}\}_{i=1}^{N_{n+1}}$.

Ví dụ : Phương trình Korteweg-de Vries (KdV)

Ở đây, ta xem xét một mô hình toán học về sóng trên bề mặt nước nông: phương trình Korteweg-de Vries (KdV) [6]. Ví dụ này thể hiện khả năng của PINN trong việc xử lý PDE bậc cao. Phương trình KdV có dạng như sau:

$$u_t + \lambda_1 uu_x + \lambda_2 u_{xxx} = 0 \quad (1.26)$$

với $\lambda = (\lambda_1, \lambda_2)$ là các tham số chưa biết. Lúc này toán tử phi tuyến của phương trình (1.21) là :

$$\mathcal{N}[u^{n+c_j}] = \lambda_1 u^{n+c_j} u_x^{n+c_j} + \lambda_2 u_{xxx}^{n+c_j}$$

Các tham số của các mạng (1.22) (1.23) (1.24) cùng với $\lambda = (\lambda_1, \lambda_2)$ được học bằng cách tối thiểu hóa sai số bình phương tổng (1.25).

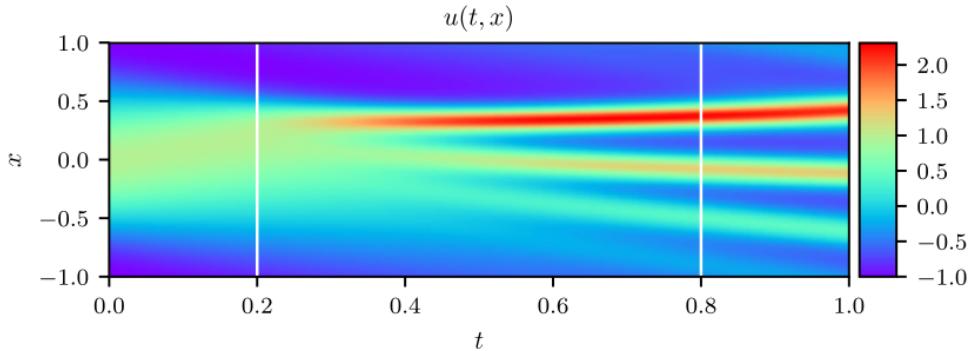
Để có được một tập dữ liệu huấn luyện và kiểm tra, ta sẽ mô phỏng (1.26) bằng các phương pháp phổ thông thường. Cụ thể, bắt đầu từ điều kiện ban đầu $u(0, x) = \cos(\pi x)$ và giả sử các điều kiện biên tuần hoàn, ta tích phân phương trình (1.26) đến thời điểm cuối cùng $t = 1.0$ bằng cách sử dụng gói Chebfun [17] với phép rời rạc Fourier phổ với 512 mode và một bộ tích phân thời gian Runge-Kutta bậc bốn tường minh với bước thời gian $\Delta t = 10^{-6}$. Sử dụng tập dữ liệu này, sau đó

ta trích xuất hai ảnh chụp nhanh nghiệm tại thời điểm $t^n = 0.2$ và $t^{n+1} = 0.8$, và lấy mẫu con ngẫu nhiên bằng cách sử dụng $N_n = 199$ và $N_{n+1} = 201$ để tạo ra một tập dữ liệu huấn luyện. Sau đó, ta sử dụng dữ liệu này để huấn luyện một mạng PINN theo mô hình thời gian rời rạc bằng cách tối thiểu hóa sai số bình phương tổng (1.25). Kiến trúc mạng được sử dụng ở đây bao gồm 4 lớp ẩn, 50 neuron mỗi lớp và một lớp đầu ra dự đoán nghiệm ở bậc q của lược đồ Runge-Kutta, tức là $u^{n+c_j}(x)$, $j = 1, \dots, q$, trong đó q được chọn theo kinh nghiệm để tạo ra sự tích lũy sai số theo thời gian theo độ chính xác của dự đoán ϵ bằng cách thiết lập:

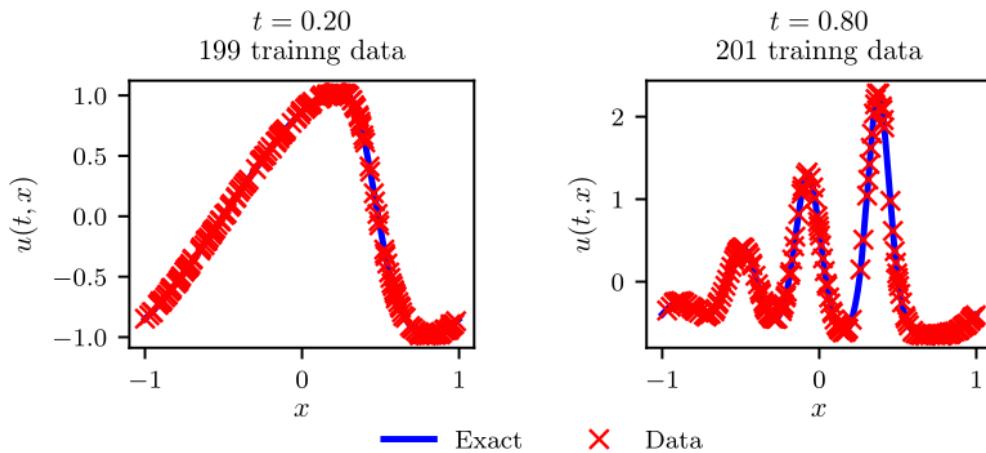
$$q = 0.5 \log \epsilon / \log(\Delta t) \quad (1.27)$$

với $\Delta t = 0.6$ trong ví dụ này.

Ở hình 1.9, ví dụ thể hiện nghiệm chính xác $u(t, x)$, cùng với vị trí của hai ảnh chụp nhanh dữ liệu được sử dụng để huấn luyện. Tổng quan chi tiết hơn về nghiệm chính xác và dữ liệu huấn luyện được trình bày ở hình 1.10. Điều đáng chú ý là toán tử phi tuyến phức tạp của phương trình (1.26) gây ra sự khác biệt đáng kể về dạng nghiệm giữa hai ảnh chụp nhanh được ghi lại. Bất chấp những khác biệt này, và khoảng cách thời gian lớn giữa hai ảnh chụp nhanh của dữ liệu huấn luyện, PINN có thể xác định chính xác các tham số chưa biết, bất kể dữ liệu huấn luyện có bị nhiễu hay không. Cụ thể, đối với trường hợp dữ liệu huấn luyện không nhiễu, sai số khi ước tính λ_1 và λ_2 lần lượt là 0,023% và 0,006%, trong khi trường hợp có 1% nhiễu trong dữ liệu huấn luyện, sai số lần lượt là 0,057% và 0,017%.



Hình 1.9: Nghiệm $u(t, x)$ tại 2 thời điểm huấn luyện.



Hình 1.10: Dữ liệu huấn luyện và nghiệm chính xác tương ứng với hai thời điểm.

Correct PDE	$u_t + uu_x + 0.0025u_{xxx} = 0$
Identified PDE (clean data)	$u_t + 1.000uu_x + 0.0025002u_{xxx} = 0$
Identified PDE (1% noise)	$u_t + 0.999uu_x + 0.0024996u_{xxx} = 0$

Hình 1.11: Phương trình đạo hàm riêng xây dựng từ việc học các tham số λ_1, λ_2 .

Chương 2

Phương pháp đánh giá PINN

2.1 Đặt vấn đề

Các PDE phi tuyến vô cùng đa dạng và có nhiều ứng dụng quan trọng trong nhiều lĩnh vực thực tế. Tuy nhiên sự phức tạp trong các bài toán phi tuyến đối lưu - phản ứng - khuyếch tán (CRD) gây khó khăn trong việc tìm nghiệm chính xác và hiệu quả. Việc giải chính xác phương trình này vô cùng quan trọng vì chúng được sử dụng cho mô hình thực tế, do vậy chỉ một sai số nhỏ cũng có thể dẫn tới các sai lầm lớn trong thực tế [7].

Chương 1 đã trình bày về cách tiếp cận sử dụng mô hình PINN, tận dụng các định luật vật lý của bài toán cho quá trình huấn luyện, giúp mô hình có thể học hỏi hiệu quả từ dữ liệu đầu vào đồng thời thỏa mãn phương trình cơ bản, giúp xấp xỉ các PDE hiệu quả và có độ chính xác cao.

Chương này sẽ xây dựng phương pháp đánh giá một cách có hệ thống hiệu suất, độ chính xác và tính nhất quán của PINN thuộc lớp bài toán thuận sử dụng mô hình dữ liệu liên tục (phần 1.3), ứng dụng trong giải xấp xỉ các PDE đối lưu - phản ứng - khuyếch tán (CRD).

2.2 Phương pháp

Hiệu suất của mô hình PINN sẽ được đánh giá bằng ba chỉ số : *chỉ số sai số tuyệt đối* ($\|\Delta\|_1$), *sai số tuyệt đối cực đại* (l_∞) và chuẩn l_2 . Các chỉ số này cung cấp sự đánh giá toàn diện về độ chính xác của mô hình trong việc giải các PDE phi tuyến.

Chỉ số sai số tuyệt đối $\|\Delta\|_1$ đo dựa trên sai số của giá trị dự đoán so với giá trị thực tế trên toàn miền tính toán:

$$\|\Delta\|_1 = \frac{1}{N} \sum_{i=1}^N |u(t_i, x_i) - u_\phi(t_i, x_i)|, \quad (2.1)$$

trong đó N là tổng số điểm đo đặc trong miền đang xét, u là nghiệm chính xác, u_ϕ là nghiệm dự đoán bởi PINN. Chỉ số này cung cấp thước đo trực quan về hiệu suất của mô hình bằng cách tính trung bình các sai số tuyệt đối và khả năng chống lại các giá trị ngoại lai.

Sai số tuyệt đối cực đại l_∞ "bắt" sai số tệ nhất tại thời điểm cụ thể t :

$$l_\infty = \max_{x \in D} |u(t, x) - u_\phi(t, x)|, \quad (2.2)$$

trong đó D là miền tính toán, u là nghiệm chính xác, u_ϕ là nghiệm dự đoán bởi PINN. Chỉ số này có giá trị trong việc xác định các miền mà mô hình có thể hoạt động kém, phản ánh độ lệch cao nhất trong nghiệm.

Chuẩn l_2 định lượng biên độ sai số tổng thể :

$$l_2 = \sqrt{\frac{1}{N} \sum_{i=1}^N |u(t_i, x_i) - u_\phi(t_i, x_i)|^2}, \quad (2.3)$$

trong đó N là tổng số điểm đo đặc trong miền đang xét, u là nghiệm chính xác, u_ϕ là nghiệm dự đoán bởi PINN. Chỉ số này phạt các độ lệch lớn hơn nhiều so với các độ lệch nhỏ hơn, khiến nó nhạy cảm hơn với các giá trị ngoại lai và cung cấp thước đo có ý nghĩa về độ chính xác tổng thể của mô hình.

Sử dụng ba thước đo này, ta có thể đánh giá toàn diện hiệu suất và độ chính xác của mô hình PINN trong việc xây dựng mô hình động lực học dựa trên PDE phi tuyến. Việc so sánh trực quan giữa các nghiệm dự đoán và thực tế cũng được thực hiện để đánh giá độ chính xác của mô hình.

2.3 Áp dụng đánh giá đối với một số trường hợp

2.3.1 Cài đặt

Dữ liệu huấn luyện bao gồm 50 điểm huấn luyện điều kiện ban đầu và 50 điểm huấn luyện điều kiện biên. Ta sử dụng hai tập hợp điểm trong có số lượng khác nhau: 3000 và 5000 điểm, được tạo ra bằng phương pháp lấy mẫu siêu lập phương Latinh (LHS) [15]. Sau đó tiến hành phân tích các chỉ số ở phần 2.2 để đánh giá hiệu suất của các mô hình trên bài toán khác nhau. Các tham số của mạng nơ-ron bao gồm tốc độ học là 1×10^{-3} , với tổng cộng 4 lớp và 20 nơ-ron mỗi lớp, hàm kích hoạt được sử dụng trong mạng là tanh. Đầu tiên quá trình huấn luyện chạy trong 5000 epoch bằng thuật toán tối ưu Adam [14], tiếp theo là 3000 lần lặp với thuật toán L-BFGS [13] để tinh chỉnh. Hàm mất mát kết hợp (1.5) được tối thiểu hóa, phương pháp cắt gradient [18] được sử dụng để ổn định quá trình huấn luyện. Một phần quan trọng của thiết lập là việc sử dụng tốc độ học và chính quy hóa L2 (Ridge) [19], giúp mô hình hội tụ hiệu quả và duy trì tính ổn định. Chính quy hóa cũng đóng vai trò then chốt trong việc ngăn ngừa hiện tượng quá khớp, đảm bảo rằng các mô hình có khả năng khai quật hóa tốt trong các kịch bản khác nhau.

2.3.2 Ví dụ

Phương trình Burgers

Phương trình Burgers phi tuyến là một PDE được nghiên cứu kỹ lưỡng và có ứng dụng rộng rãi trong động lực học chất lưu, từ thủy động lực học, và các lĩnh vực khoa học ứng dụng và kỹ thuật khác [7]. Ví dụ này xem xét phương trình Burgers với điều kiện biên Dirichlet đồng nhất, có thể được biểu thị như sau:

$$u_t = \frac{1}{Re} u_{xx} - uu_x, \quad x \in D \equiv [0, 1], \quad t > 0 \quad (2.4)$$

với điều kiện biên:

$$u(0, t) = u(1, t) = 0, \quad t > 0 \quad (2.5)$$

và điều kiện ban đầu:

$$u(x, 0) = \frac{2\pi \sin(x)}{Re(\sigma + \cos(\pi x))}, \quad x \in D. \quad (2.6)$$

Nghiệm của phương trình này với điều kiện biên và điều kiện ban đầu trên là:

$$u(x, t) = \frac{2\pi e^{-\pi^2 t/Re} \sin(\pi x)}{Re(\sigma + e^{-\pi^2 t/Re} \cos(\pi x))} \quad (2.7)$$

với các tham số $Re = 1$ và $\sigma = 2$.

Kết quả đánh giá sai số tuyệt đối thể hiện trong bảng 2.1. Với $t = 0.02$ sai số dao động từ 1.52×10^{-5} tới 2.05×10^{-4} , và từ 1.7×10^{-5} tới 2.93×10^{-4} khi $t = 0.04$. Cho thấy sai số nhỏ và ổn định theo thời gian. Kết quả đánh giá sai số l_∞ và l_2 được thể hiện trong bảng 2.2. Trong trường hợp 3000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 1.81×10^{-4} đến 1.9×10^{-3} , còn sai số l_2 nằm trong khoảng 8.28×10^{-5} tới 7.87×10^{-4} . Trong trường hợp 5000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 3.38×10^{-5} đến 3.46×10^{-3} , còn sai số l_2 nằm trong khoảng 1.35×10^{-5} tới 9.36×10^{-4} . Cho thấy không có sự khác biệt rõ rệt khi thay đổi số điểm dữ liệu. Hình 2.1 minh họa trực quan cho thấy sự tương khớp giữa các dự đoán của PINN và nghiệm chính xác của phương trình Burgers. Điều này kết hợp với sai số đo lường cho thấy hiệu quả của phương pháp PINN trong việc giải phương trình Burgers với độ chính xác cao.

t	x	$ \Delta_1 $
0.02	0.1	9.2×10^{-5}
	0.2	1.6×10^{-4}
	0.4	2.04×10^{-4}
	0.6	4.69×10^{-5}
	0.8	1.52×10^{-5}
	1.0	2.05×10^{-4}
0.04	0.1	7.4×10^{-5}
	0.2	1.7×10^{-5}
	0.4	6.8×10^{-5}
	0.6	7.0×10^{-5}
	0.8	7.7×10^{-5}
	1.0	2.93×10^{-4}

Bảng 2.1: Sai số tuyệt đối của phương trình Burgers

Phương trình Fisher

Phương trình Fisher với các điều kiện biên Neumann không đồng nhất là một mô hình cơ bản trong nhiều lĩnh vực như truyền nhiệt và khói lượng, sinh học và sinh thái học [7]. Phương trình này có công thức như sau:

$$u_t = u_{xx} + 6u(1 - u), \quad x \in D \equiv [0, 1], \quad t > 0, \quad (2.8)$$

với điều kiện biên :

$$\begin{aligned} u(0, t) &= (1 + e^{-5t})^{-2}, \quad t > 0 \\ u(1, t) &= (1 + e^{1-5t})^{-2}, \quad t > 0 \end{aligned} \quad (2.9)$$

t	3000 điểm dữ liệu		5000 điểm dữ liệu	
	l_∞	l_2	l_∞	l_2
0.0	1.9×10^{-3}	6.03×10^{-4}	3.46×10^{-3}	8.31×10^{-4}
0.2	4.52×10^{-4}	1.4×10^{-4}	1.64×10^{-4}	1.07×10^{-4}
0.4	2.53×10^{-4}	1.02×10^{-4}	2.55×10^{-4}	1.66×10^{-4}
0.6	9.11×10^{-4}	7.87×10^{-4}	3.38×10^{-5}	1.35×10^{-5}
0.8	1.81×10^{-4}	1.03×10^{-4}	1.3×10^{-4}	9.36×10^{-4}
1.0	2.29×10^{-4}	8.28×10^{-5}	5.01×10^{-4}	2.11×10^{-4}

Bảng 2.2: Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Burgers

và điều kiện ban đầu:

$$u(x, 0) = (1 + e^x)^{-2}, \quad x \in D. \quad (2.10)$$

Ở đây $u(x, t)$ mô tả mật dân số hoặc một số liệu nào đó dạng như thế, với $D = [0, 1]$ mô tả miền không gian. Nghiệm chính xác của phương trình là:

$$u(x, t) = (1 + e^{x-5t})^{-2}. \quad (2.11)$$

Phương trình này mô tả sự tiến hóa của u theo thời gian t , nắm bắt các hiện tượng trong đó u thể hiện sự tăng trưởng logistic hoặc các quá trình khuếch tán chịu ảnh hưởng bởi sự tương tác giữa các số hạng khuếch tán và phản ứng.

Kết quả đánh giá sai số tuyệt đối thể hiện trong bảng 2.3. Với $t = 0.05$ sai số dao động từ 1×10^{-6} tới 8.4×10^{-5} , và từ 1.9×10^{-5} tới 8.5×10^{-5} khi $t = 0.1$. Cho thấy sai số ở mức thấp và tăng nhẹ theo thời gian. Kết quả đánh giá sai số l_∞ và l_2 được thể hiện trong bảng 2.4. Trong trường hợp 3000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 4.72×10^{-5} đến 1.83×10^{-4} , còn sai số l_2 nằm trong khoảng 2.42×10^{-5} tới 9.09×10^{-5} . Trong trường hợp 5000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 1.13×10^{-5} đến 8.1×10^{-5} , còn sai số l_2 nằm trong khoảng 1.92×10^{-5} tới 7.61×10^{-5} . Cho thấy sai số thấp và không bị ảnh hưởng bởi số điểm dữ liệu. Hình 2.2 minh họa trực quan cho thấy sự tương khớp giữa các dự đoán của PINN và nghiệm chính xác của phương trình Fisher. Điều này kết hợp với sai số đo lường cho thấy hiệu quả của phương pháp PINN trong việc giải phương trình Fisher với độ chính xác cao.

Phương trình Burgers-Huxley

Phương trình Burgers-Huxley là một phương trình đạo hàm riêng parabol phi tuyến nổi tiếng đã được ứng dụng trong nhiều lĩnh vực, bao gồm cơ học chất lưu, sinh học và lý thuyết đốt cháy [7]. Ví dụ xem xét phương trình Burgers-Huxley với điều kiện biên không có thông lượng, được biểu thị như sau:

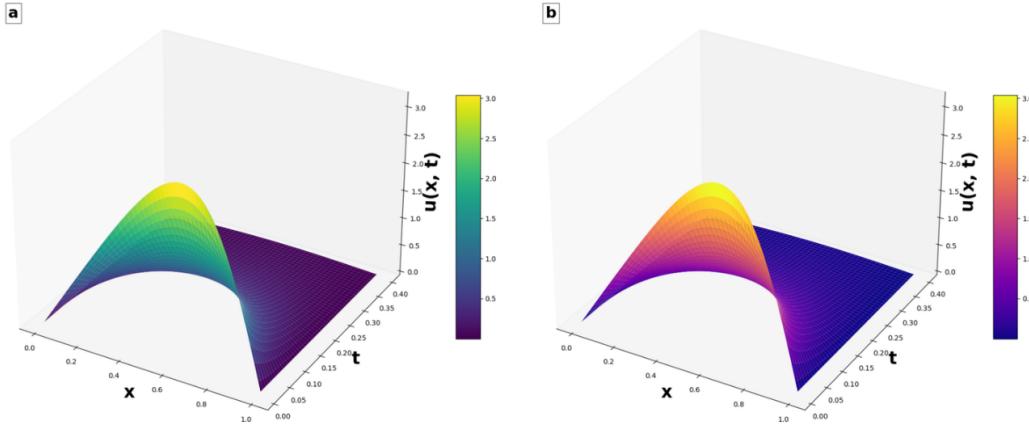
$$u_t = u_{xx} - \phi uu_x + \epsilon u(1-u)(\tau - u), \quad x \in [-25, 17], t > 0, \quad (2.12)$$

với điều kiện biên:

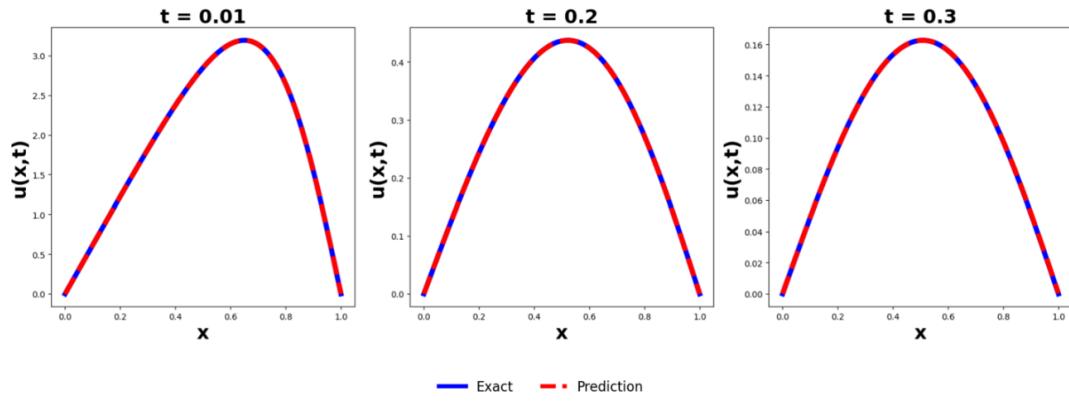
$$u(-25, t) = u(17, t) = 0, \quad t > 0, \quad (2.13)$$

với điều kiện ban đầu:

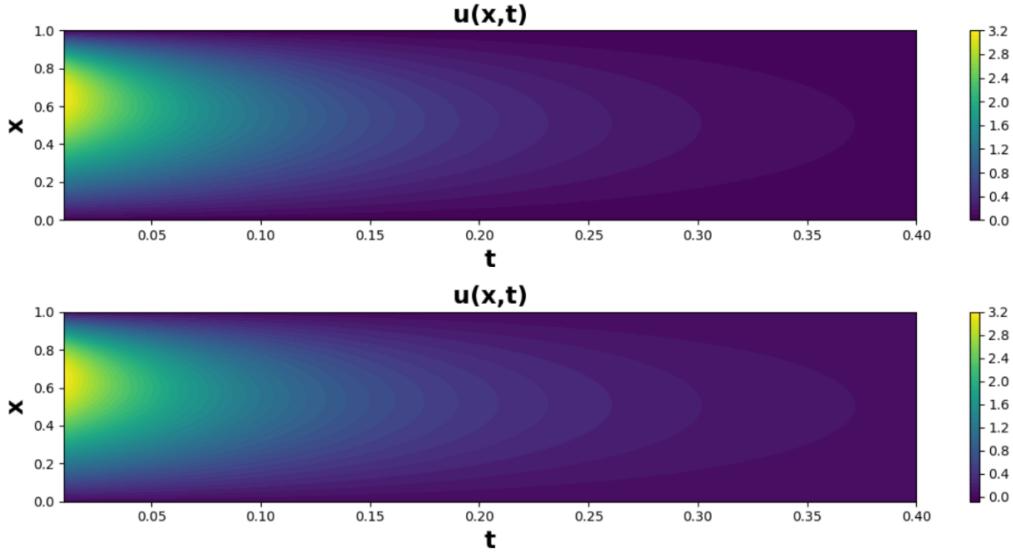
$$u(x, 0) = \frac{3}{2} + \frac{1}{2} \tanh\left(\frac{x}{2}\right), \quad x \in D, \quad (2.14)$$



(a) Biểu diễn đồ thị bề mặt 3D cho thấy các nghiệm chính xác **a** và các nghiệm của mô hình PINN **b** cho phương trình Burgers tại các thời điểm khác nhau.



(b) So sánh các nghiệm PINN với các nghiệm chính xác cho phương trình Burgers tại các thời điểm: $t = 0.01$, $t = 0.2$ và $t = 0.3$. Các đường liền màu xanh biểu thị các nghiệm chính xác, trong khi các đường đứt nét màu đỏ biểu thị các nghiệm dự đoán PINN.



(c) Biểu đồ phân phối 2D minh họa các nghiệm chính xác ở phía trên và các nghiệm dự đoán tương ứng thu được sử dụng PINN ở phía dưới cho phương trình Burgers tại các thời điểm khác nhau.

Hình 2.1: Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Burgers.

t	x	$\ \Delta_1\ $
0.05	0.1	2.00×10^{-5}
	0.2	1.00×10^{-6}
	0.4	1.80×10^{-5}
	0.6	2.20×10^{-5}
	0.8	4.20×10^{-5}
	1.0	8.40×10^{-5}
0.10	0.1	4.80×10^{-5}
	0.2	2.00×10^{-5}
	0.4	1.90×10^{-5}
	0.6	3.30×10^{-5}
	0.8	5.80×10^{-5}
	1.0	8.50×10^{-5}

Bảng 2.3: Sai số tuyệt đối của phương trình Fisher

t	3000 điểm dữ liệu		5000 điểm dữ liệu	
	l_∞	l_2	l_∞	l_2
0.0	1.01×10^{-4}	6.13×10^{-5}	1.63×10^{-4}	7.61×10^{-5}
0.2	1.83×10^{-4}	9.09×10^{-5}	4.99×10^{-5}	3.26×10^{-5}
0.4	1.70×10^{-4}	6.33×10^{-5}	1.13×10^{-5}	6.01×10^{-5}
0.6	6.11×10^{-5}	4.59×10^{-5}	4.07×10^{-5}	1.92×10^{-5}
0.8	4.72×10^{-5}	2.42×10^{-5}	5.43×10^{-5}	2.23×10^{-5}
1.0	1.35×10^{-4}	5.07×10^{-5}	8.10×10^{-5}	3.26×10^{-5}

Bảng 2.4: Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Fisher

ở đây ϕ, ϵ, τ là các tham số thực.

Nghiệm chính xác như sau:

$$u(x, t) = \frac{3}{2} + \frac{1}{2} \left[\tanh \frac{1}{2} (x + 3t) \right], \quad (2.15)$$

với các tham số $\phi = -1, \epsilon = 1, \tau = 2$.

Kết quả đánh giá sai số tuyệt đối thể hiện trong bảng 2.5. Với $t = 0.1$ sai số dao động từ 6.89×10^{-4} tới 6.3×10^{-3} , và từ 1×10^{-3} tới 6.09×10^{-3} khi $t = 0.2$. Cho thấy sai số thấp và biến đổi tăng giảm tùy vào vị trí không gian mà không phụ thuộc thời gian. Kết quả đánh giá sai số l_∞ và l_2 được thể hiện trong bảng 2.6. Trong trường hợp 3000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 1.37×10^{-2} đến 2.07×10^{-2} , còn sai số l_2 nằm trong khoảng 4.07×10^{-3} tới 5.96×10^{-3} . Trong trường hợp 5000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 1.05×10^{-2} đến 2.22×10^{-2} , còn sai số l_2 nằm trong khoảng 5.87×10^{-3} tới 1.25×10^{-2} . Cho thấy khi số điểm dữ liệu tăng, sai số l_∞ giảm chứng tỏ độ lệch lớn nhất giảm, nhưng chuẩn l_2 tăng lên cho thấy mức độ sai số tổng thể tăng lên. Hình 2.3 minh họa trực quan cho thấy sự tương khớp giữa các dự đoán của PINN và nghiệm chính xác của phương trình Burgers-Huxley. Điều này kết hợp với sai số đo lường cho thấy hiệu quả của phương pháp PINN trong việc giải phương trình Burgers-Huxley với độ chính xác cao.

Phương trình Newell-Whitehead-Segel

t	x	$\ \Delta_1\ $
0.10	-25.00	4.27×10^{-3}
	-20.80	7.27×10^{-4}
	-12.40	2.53×10^{-3}
	-4.00	5.37×10^{-3}
	0.20	6.89×10^{-4}
	4.40	5.41×10^{-3}
	12.80	3.04×10^{-3}
	17.0	6.30×10^{-3}
0.20	-25.00	5.38×10^{-3}
	-20.80	1.00×10^{-3}
	-12.40	3.47×10^{-3}
	-4.00	6.09×10^{-3}
	0.20	1.81×10^{-3}
	4.40	4.19×10^{-3}
	12.80	2.08×10^{-3}
	17.0	5.22×10^{-3}

Bảng 2.5: Sai số tuyệt đối của phương trình Burgers-Huxley

t	3000 điểm dữ liệu		5000 điểm dữ liệu	
	l_∞	l_2	l_∞	l_2
0.0	1.37×10^{-2}	4.07×10^{-3}	1.05×10^{-2}	5.87×10^{-3}
0.2	1.55×10^{-2}	4.57×10^{-3}	1.09×10^{-2}	5.94×10^{-3}
0.4	1.81×10^{-2}	4.95×10^{-3}	1.19×10^{-2}	6.62×10^{-3}
0.6	1.99×10^{-2}	5.30×10^{-3}	1.39×10^{-2}	7.78×10^{-3}
0.8	2.06×10^{-2}	5.66×10^{-3}	1.73×10^{-2}	9.72×10^{-3}
1.0	2.07×10^{-2}	5.96×10^{-3}	2.22×10^{-2}	1.25×10^{-2}

Bảng 2.6: Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Burgers-Huxley

Phương trình Newell-Whitehead-Segel mô hình hóa sự tương tác giữa hạng tử khuếch tán và hiệu ứng phi tuyến tính của hạng tử phản ứng [7]. Phương trình này có công thức như sau:

$$u_t = \eta u_{xx} + \mu u - \nu u^\rho, \quad x \in D \equiv [0, 1], t > 0, \quad (2.16)$$

với các điều kiện biên:

$$\begin{aligned} u(0, t) &= -\frac{2}{\sqrt{6}} \frac{e^{-5t/6}}{(1 + e^{-5t/6})^3}, \quad t > 0 \\ u(1, t) &= -\frac{2}{\sqrt{6}} \frac{e^{(\sqrt{6}-5t)/6}}{(1 + e^{(\sqrt{6}-5t)/6})^3}, \quad t > 0 \end{aligned} \quad (2.17)$$

và điều kiện ban đầu:

$$u(x, 0) = (1 + e^{x/\sqrt{6}})^{-2}, \quad x \in D. \quad (2.18)$$

Ở đây η, μ, ν là các tham số thực với $\eta > 0$ và ρ là tham số nguyên dương. Nghiên cứu chọn $\eta = \mu = \nu = 1$ và $\rho = 2$. Nghiệm chính xác là :

$$u(x, t) = \left(1 + e^{\frac{x}{\sqrt{6}} - \frac{5t}{6}}\right)^{-2}. \quad (2.19)$$

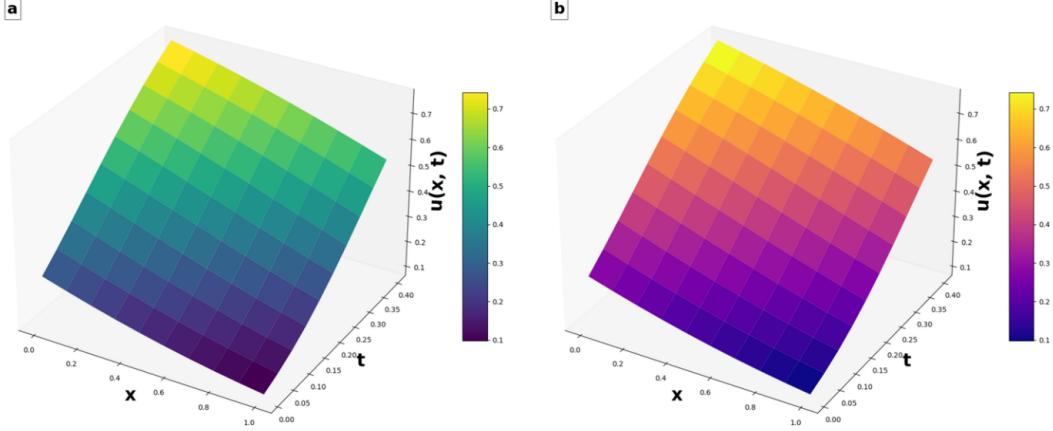
Kết quả đánh giá sai số tuyệt đối thể hiện trong bảng 2.7. Với $t = 0.03$ sai số dao động từ 9.1×10^{-6} tới 3.24×10^{-5} , và từ 1.38×10^{-5} tới 3.78×10^{-5} khi $t = 0.05$. Kết quả đánh giá sai số l_∞ và l_2 được thể hiện trong bảng 2.8. Cho thấy sai số ổn định ở mức thấp và biến đổi không phụ thuộc không-thời gian. Trong trường hợp 3000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 9.96×10^{-5} đến 4.88×10^{-3} , còn sai số l_2 nằm trong khoảng 5.64×10^{-5} tới 1.25×10^{-3} . Trong trường hợp 5000 điểm dữ liệu, sai số l_∞ nằm trong khoảng 1.54×10^{-5} đến 8.49×10^{-5} , còn sai số l_2 nằm trong khoảng 1.05×10^{-5} tới 5.72×10^{-5} . Cho thấy khi số điểm dữ liệu tăng, cả sai số l_∞ và chuẩn l_2 đều giảm cho thấy độ lệch lớn nhất và mức độ sai số tổng thể giảm xuống. Hình 2.4 minh họa trực quan cho thấy sự tương khớp giữa các dự đoán của PINN và nghiệm chính xác của phương trình Newell-Whitehead-Segel. Điều này kết hợp với sai số đo lường cho thấy hiệu quả của phương pháp PINN trong việc giải phương trình Newell-Whitehead-Segel với độ chính xác cao.

t	x	$\ \Delta_1\ $
0.03	0.1	9.10×10^{-6}
	0.2	1.77×10^{-5}
	0.4	3.15×10^{-5}
	0.6	3.24×10^{-5}
	0.8	2.47×10^{-5}
	1.0	1.03×10^{-5}
0.05	0.1	2.55×10^{-5}
	0.2	3.18×10^{-5}
	0.4	3.78×10^{-5}
	0.6	3.50×10^{-5}
	0.8	2.31×10^{-5}
	1.0	1.38×10^{-5}

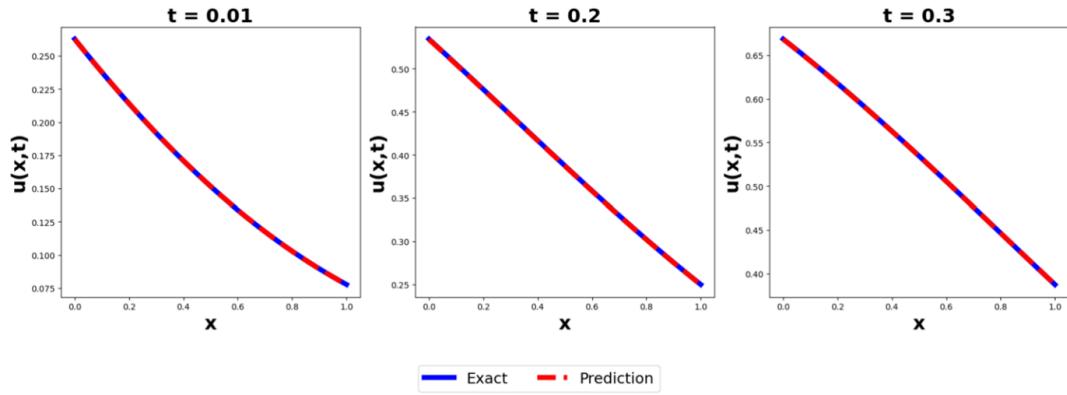
Bảng 2.7: Sai số tuyệt đối của phương trình Newell-Whitehead-Segel

t	3000 điểm dữ liệu		5000 điểm dữ liệu	
	l_∞	l_2	l_∞	l_2
0.0	4.88×10^{-3}	1.25×10^{-3}	8.49×10^{-5}	3.93×10^{-5}
0.2	1.04×10^{-4}	6.75×10^{-5}	2.23×10^{-5}	1.28×10^{-5}
0.4	1.77×10^{-4}	8.31×10^{-5}	4.20×10^{-5}	2.63×10^{-5}
0.6	9.96×10^{-5}	5.64×10^{-5}	1.54×10^{-5}	1.05×10^{-5}
0.8	3.26×10^{-4}	8.48×10^{-5}	6.79×10^{-5}	5.09×10^{-5}
1.0	3.75×10^{-4}	1.37×10^{-4}	7.49×10^{-5}	5.72×10^{-5}

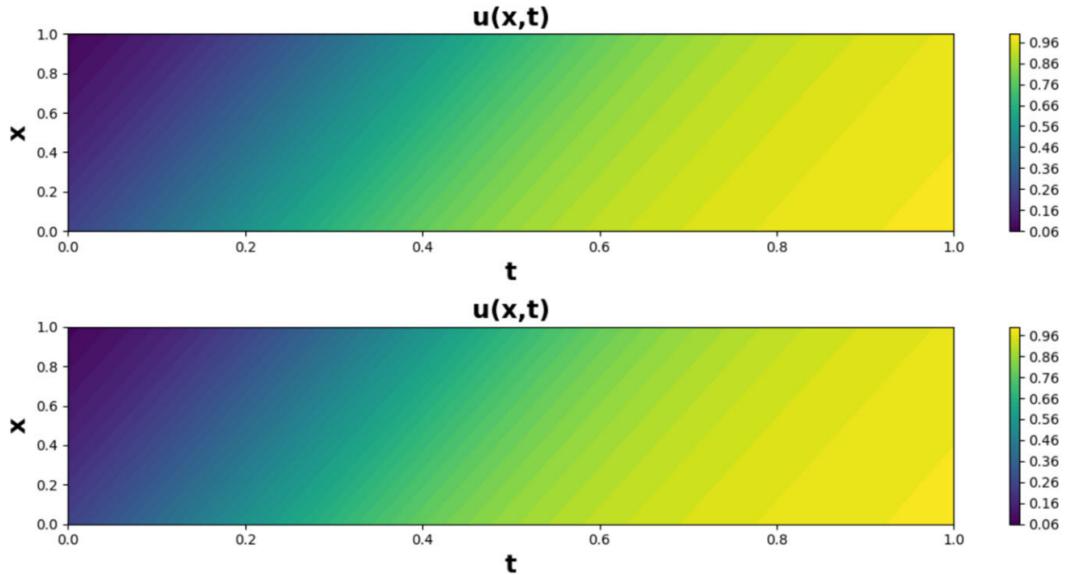
Bảng 2.8: Sai số l_∞ và l_2 với số điểm dữ liệu khác nhau của phương trình Newell-Whitehead-Segel



(a) Biểu diễn đồ thị bề mặt 3D cho thấy các nghiệm chính xác **a** và các nghiệm của mô hình PINN **b** cho phương trình Fisher tại các thời điểm khác nhau.

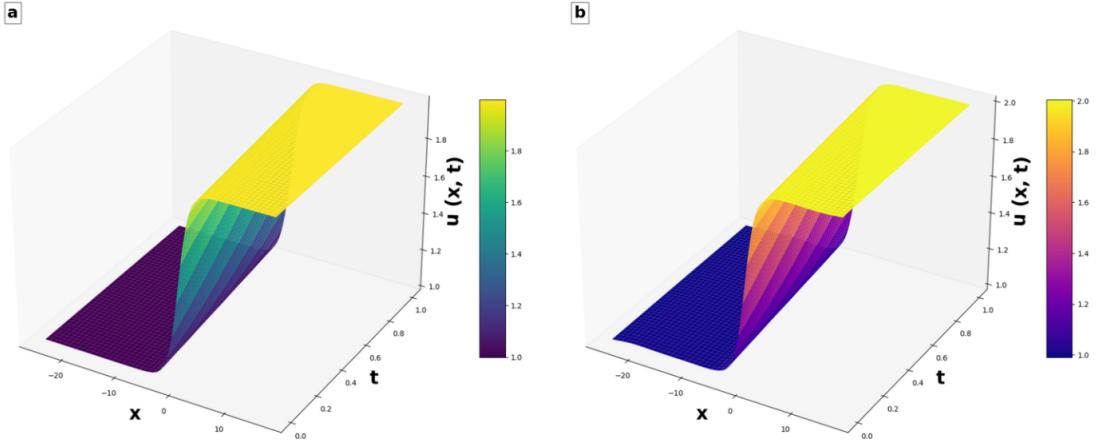


(b) So sánh các nghiệm PINN với các nghiệm chính xác cho phương trình Fisher tại các thời điểm: $t = 0.01$, $t = 0.2$ và $t = 0.3$. Các đường liền màu xanh biểu thị các nghiệm chính xác, trong khi các đường đứt nét màu đỏ biểu thị các nghiệm dự đoán PINN.

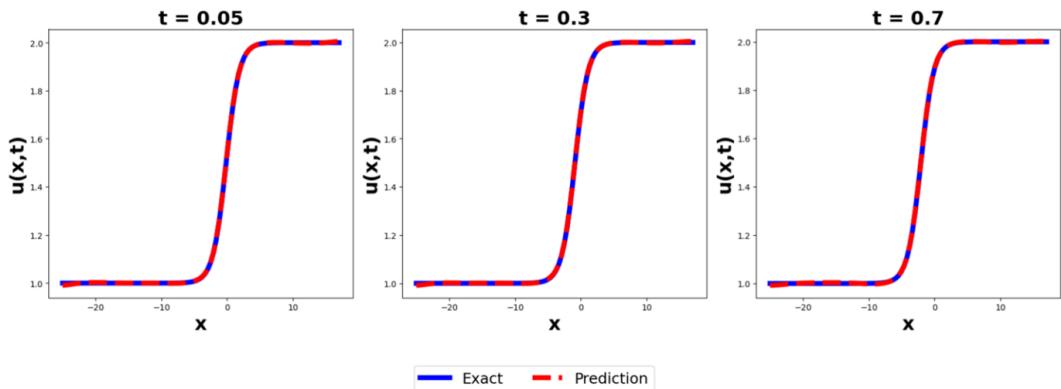


(c) Biểu đồ phân phối 2D minh họa các nghiệm chính xác ở phía trên và các nghiệm dự đoán tương ứng thu được sử dụng PINN ở phía dưới cho phương trình Fisher tại các thời điểm khác nhau.

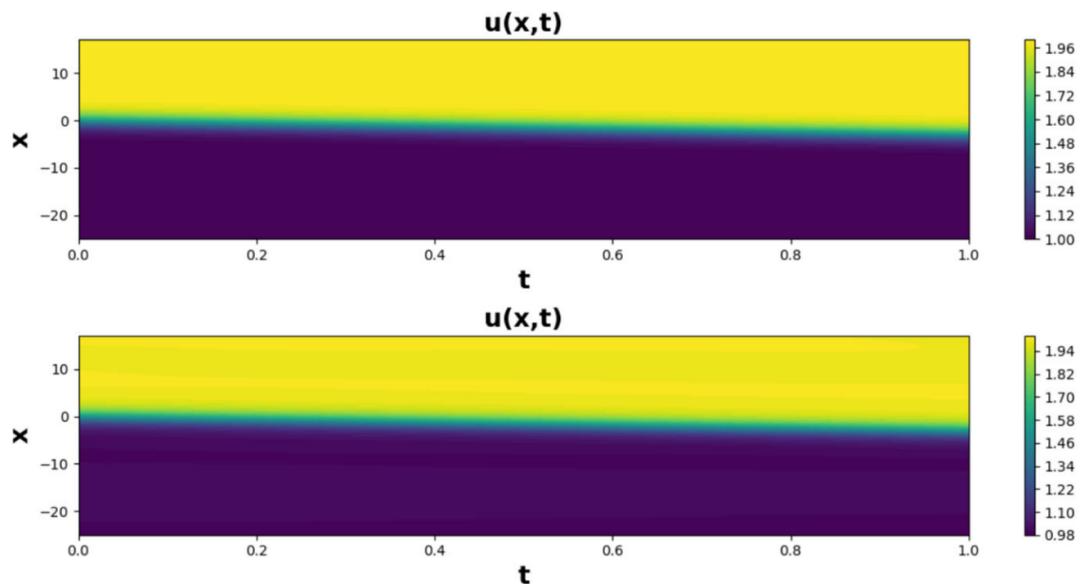
Hình 2.2: Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Fisher.



(a) Biểu diễn đồ thị bề mặt 3D cho thấy các nghiệm chính xác **a** và các nghiệm của mô hình PINN **b** cho phương trình Burgers-Huxley tại các thời điểm khác nhau.

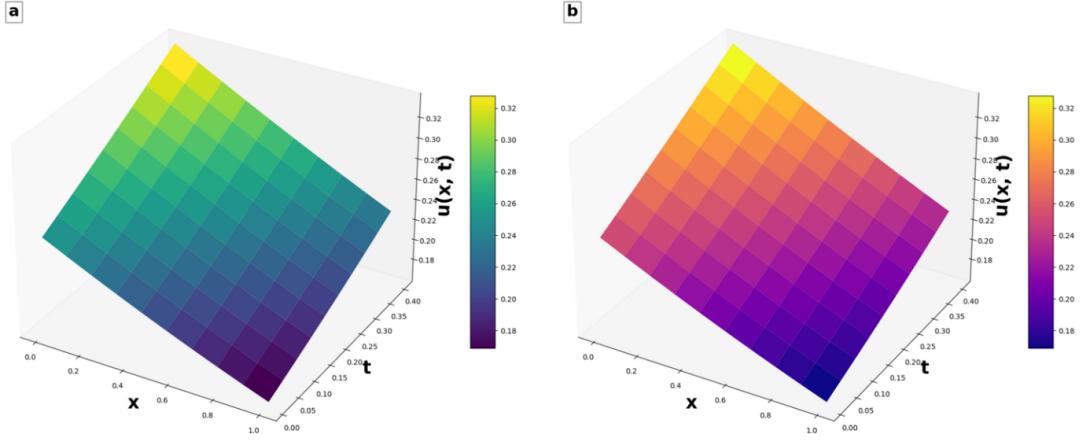


(b) So sánh các nghiệm PINN với các nghiệm chính xác cho phương trình Burgers-Huxley tại các thời điểm: $t = 0.01$, $t = 0.2$ và $t = 0.3$. Các đường liền màu xanh biểu thị các nghiệm chính xác, trong khi các đường đứt nét màu đỏ biểu thị các nghiệm dự đoán PINN.

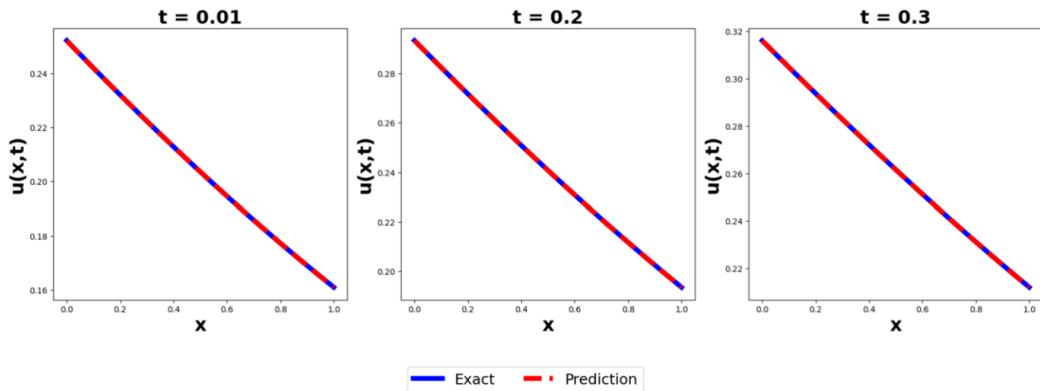


(c) Biểu đồ phân phối 2D minh họa các nghiệm chính xác ở phía trên và các nghiệm dự đoán tương ứng thu được sử dụng PINN ở phía dưới cho phương trình Burgers-Huxley tại các thời điểm khác nhau.

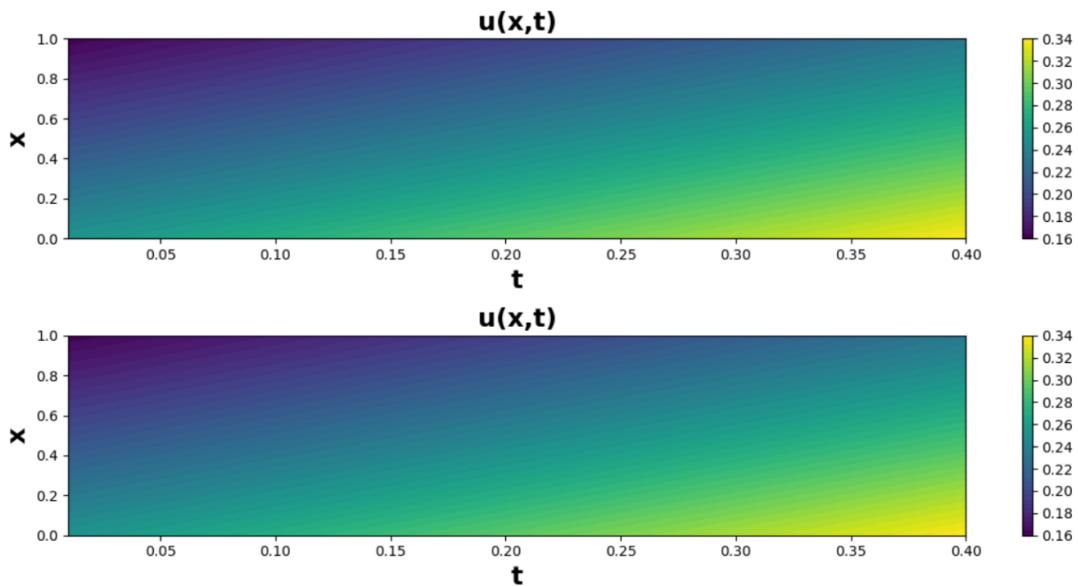
Hình 2.3: Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Burgers-Huxley.



(a) Biểu diễn đồ thị bề mặt 3D cho thấy các nghiệm chính xác **a** và các nghiệm của mô hình PINN **b** cho phương trình Newell-Whitehead-Segel tại các thời điểm khác nhau.



(b) So sánh các nghiệm PINN với các nghiệm chính xác cho phương trình Newell-Whitehead-Segel tại các thời điểm: $t = 0.01$, $t = 0.2$ và $t = 0.3$. Các đường liền màu xanh biểu thị các nghiệm chính xác, trong khi các đường đứt nét màu đỏ biểu thị các nghiệm dự đoán PINN.



(c) Biểu đồ phân phối 2D minh họa các nghiệm chính xác ở phía trên và các nghiệm dự đoán tương ứng thu được sử dụng PINN ở phía dưới cho phương trình Newell-Whitehead-Segel tại các thời điểm khác nhau.

Hình 2.4: Kết quả thể hiện sự so sánh giữa mô hình PINN và nghiệm chính xác với phương trình Newell-Whitehead-Segel.

Chương 3

Mô hình PINN tự động khám phá cấu trúc

3.1 Đặt vấn đề

Phương pháp PINN hiện có tập trung vào việc áp đặt các ràng buộc vật lý thông qua các hàm mất mát, bỏ qua các đặc điểm cấu trúc rõ ràng của các hệ thống vật lý cơ bản. Hơn nữa, các hàm mất mát bên ngoài chỉ giảm thiểu sự chênh lệch trung bình giữa các dự đoán của mô hình và cơ chế vật lý thực tế [20]. Do đó, đối với các bài toán đòi hỏi tính nhất quán vật lý cao, một hạn chế của PINN là các dự đoán của chúng không tuân thủ nghiêm ngặt các định luật bảo toàn vật lý cơ bản [21].

Lấy cảm hứng từ phương pháp chắt lọc kiến thức, một nhóm nghiên cứu từ Đại học Viện Hàn lâm Khoa học Trung Quốc (UCAS) đề xuất một khuôn khổ khám phá PINN mới là Ψ -NN (phát âm là Psi-NN) [8]. Khuôn khổ này cho phép tự động nhận dạng, trích xuất và tái tạo kiến trúc mạng. Trong Ψ -NN, chính quy hóa vật lý (từ các chương trình điều khiển) và chính quy hóa tham số được tách rời và áp dụng riêng biệt cho hai mạng khác nhau, gọi là mạng giáo viên (Teacher NN) và mạng học sinh (Student NN). Thông tin vật lý được truyền tải hiệu quả từ mạng giáo viên sang mạng học sinh thông qua phương pháp chắt lọc, bảo toàn các ràng buộc vật lý thiết yếu đồng thời mở rộng khả năng biểu diễn của mạng học sinh. Sau đó, một thuật toán trích xuất cấu trúc được tối ưu hóa sẽ tự động nhận dạng các ma trận tham số có ý nghĩa vật lý, đồng thời giữ lại tối đa không gian đặc trưng của mạng học sinh. Cuối cùng, một cơ chế khởi tạo lại được sử dụng để tái cấu trúc mạng, đảm bảo tính nhất quán vật lý trong cấu trúc mạng đồng thời mang lại khả năng ứng dụng cho mô hình. Bằng cách tích hợp quá trình chắt lọc, trích xuất cấu trúc và tái cấu trúc mạng, Ψ -NN kỳ vọng đạt được tính nhất quán vật lý, khả năng diễn giải và dự đoán có độ chính xác cao trong các mạng nơ-ron có cấu trúc.

3.2 Phương pháp

Ý tưởng cốt lõi của Ψ -NN là nhúng thông tin vật lý - chẳng hạn như tính đối xứng không gian-thời gian và các định luật bảo toàn - trực tiếp vào kiến trúc mạng. Phương pháp Ψ -NN bao gồm ba thành phần chính: chắt lọc, trích xuất cấu trúc và tái cấu trúc mạng. Quá trình chắt lọc cho phép truyền tải thông tin vật lý mà không cần can thiệp thêm, tách rời việc tối ưu hóa hướng vật lý và tham số bằng

cách tách các sai số đạo hàm bậc cao khỏi PDEs. Phương pháp trích xuất cấu trúc sau đó tự động xác định các ma trận tham số hạng thấp có tính nhất quán vật lý trong khi vẫn bảo toàn thông tin vật lý. Cuối cùng, các ma trận tham số hạng thấp được tái cấu trúc để tạo thành các cấu trúc mạng có liên quan đến vật lý.

3.2.1 Chất lọc dựa trên thông tin vật lý

Xét một PDE với tọa độ thời gian t và tọa độ không gian $x \in \mathbb{R}^n$, có nghiệm được ký hiệu là $u(t, x) \in \mathbb{R}^k$:

$$\mathcal{L}(u, t, x) = 0, \quad x \in \Omega, \quad t \in [0, T]. \quad (3.1)$$

Các mạng giáo viên và học sinh được định nghĩa như sau:

$$\mathcal{N}_T(x, t; \theta_T) = \tilde{u}_T, \quad (3.2)$$

$$\mathcal{N}_S(x, t; \theta_T) = \tilde{u}_S, \quad (3.3)$$

với θ là vector các tham số có thể huấn luyện của mạng, còn \tilde{u} là nghiệm được dự đoán của PDE. Kí hiệu T đại diện cho mạng giáo viên, trong khi S đại diện cho mạng học sinh.

Về cơ bản, mạng học sinh được thiết kế để sao chép đầu ra của mạng giáo viên [8]. Hàm mất mát chất lọc được cho bởi:

$$MSE_T = \frac{1}{M_T} \sum_{i=1}^{M_T} |\tilde{u}_T^i - \tilde{u}_S^i| \quad (3.4)$$

trong đó M_T là số lượng điểm thiết lập trong miền tính toán.

Do đó, như một chiến lược huấn luyện theo từng giai đoạn (Đầu tiên, mạng lưới giáo viên được sử dụng để dự đoán trong miền tính toán, sau đó mạng học sinh được huấn luyện để học các kết quả của mạng giáo viên.), mạng giáo viên mang theo sự thiên lệch trong việc học thông tin vật lý chứa các gradient bậc cao (như đạo hàm bậc hai hoặc bậc cao hơn trong PDE), trong khi mạng học sinh được phép dịch chuyển theo hướng gradient trong chính quy hóa tham số. Để huấn luyện và trích xuất các cấu trúc mạng có ý nghĩa từ mạng học sinh, cần có thêm các kỹ thuật phân tích tham số [8].

3.2.2 Phương pháp trích xuất cấu trúc

Fương pháp trích xuất cấu trúc ở đây dựa trên chính quy hóa L2 [19] (làm mịn tham số), bản chất toán học của nó có thể được suy ra bằng phương pháp nhân tử Lagrange [22], thúc đẩy sự hội tụ tham số trong cùng một lớp. Chính quy hóa L2 chỉ được áp dụng cho mạng học sinh, trong khi mạng giáo viên không sử dụng. Trước tiên, hãy xem xét chính quy hóa L2 trên các tham số của mạng học sinh:

$$\Omega(\theta_S) = \sum_{i=1}^L \omega_i \|\theta_i\|_2^2, \quad (3.5)$$

trong đó L là số lớp của mạng, ω_i là trọng số chính quy, và θ_i là các tham số biến đổi afin phi tuyến của lớp thứ i.

Dưới chính quy hóa L2, vector tham số bị kéo giãn theo hướng vector riêng chính (principal eigenvector) của nó, từ đó tăng cường các đặc trưng chính (major

features) trong khi ức chế các đặc trưng phụ (minor ones) [8]. Đối với xu hướng tiến hóa của các tham số dưới sự điều chỉnh của L2, chúng ta có các định lý sau:

Định lý 1: Đối với n tham số có thể huấn luyện $\theta_1, \theta_2, \dots, \theta_n$ trong cùng một lớp ẩn, nếu chúng đóng vai trò tương đương trong mạng, chúng sẽ hội tụ dưới sự điều chỉnh L2.

Định lý 2: Với n giá trị tham số có thể huấn luyện $|\theta_1|, |\theta_2|, \dots, |\theta_n|$ trong cùng một lớp ẩn, nếu tồn tại tính đối xứng tham số giữa chúng, chúng sẽ hội tụ dưới sự chính quy hóa.

Dựa trên ảnh hưởng của chính quy hóa L2 lên giá trị tham số, để đảm bảo rằng các tham số biểu thị các mối tương quan tương tự trong mỗi nơ-ron có cùng giá trị và để nén thêm mạng nơ-ron, thuật toán phân cụm kết hợp phân cấp (HAC) [23] được thực hiện trên các giá trị tuyệt đối của trọng số trong mỗi lớp. HAC không yêu cầu số lượng cụm được thiết lập trước và có thể giữ lại một cách thích ứng các tâm cụm cần thiết; nó cũng tạo ra một biểu đồ cây phân cấp, cho phép lựa chọn mức độ phân cụm thích hợp thông qua đánh giá. Ở đây sử dụng khoảng cách Euclide để đo lường sự phân bố của dữ liệu.

Với mỗi trong số θ_i của lớp i , ta coi là 1 vector đặc trưng và áp dụng thuật toán HAC để phân cụm như sau:

$$\{C_1, C_2, \dots, C_K\} = HAC(|\theta_i|, K), \quad (3.6)$$

trong đó K là số cụm còn C_k thể hiện cho cụm thứ k .

Sau khi phân cụm, giá trị tuyệt đối của trọng số trong mỗi cụm sẽ được thay thế bởi tâm cụm μ_k :

$$\mu_k = \frac{1}{|C_k|} \sum_{\theta_i \in C_k} \theta_i. \quad (3.7)$$

Cuối cùng, cập nhật trọng số mới:

$$\theta_i^{new} = sgn(\theta_i) \cdot \mu_K, \quad \forall \theta_i \in C_k.$$

Theo cách này, vectơ tham số n-chiều có thể huấn luyện θ_i ở lớp thứ i được rút gọn thành một vectơ tâm cụm K chiều và một vectơ dấu hai chiều, đạt được sự tinh chỉnh cấu trúc tối đa.

Sau khi phân cụm và thay thế, mạng thể hiện một cấu trúc ma trận được sắp xếp mới, có thể vẫn chứa một số tham số dư thừa và tái sử dụng, đòi hỏi phải phân tích thêm. Do bản chất của việc phân cụm, việc chỉ sử dụng độ thừa có cấu trúc [24] có thể bỏ qua các mối quan hệ giữa các tham số của các lớp ẩn liền kề, do đó không thể xác định được các cấu trúc mạng có liên quan về mặt vật lý được mã hóa trong các mối quan hệ tham số này. Vì vậy, thay vì phân rã hạng thấp thuần túy, chúng ta áp dụng chiến lược nén lai kết hợp các ràng buộc hạng thấp với chia sẻ tham số có cấu trúc. Về cơ bản, phương pháp này nén và đơn giản hóa ma trận trọng số thông qua chia sẻ tham số và thiết kế có cấu trúc, ánh xạ ma trận trọng số chiều cao sang không gian con có cấu trúc chiều thấp. Cụ thể, Ψ -NN không chỉ giảm hạng của ma trận tham số mà còn tránh được các cấu trúc mạng quá phức tạp bằng cách xác định sự tái sử dụng dư thừa dưới dạng các vectơ cơ sở ma trận con lặp lại.

Khác với các nút cắt tỉa có cấu trúc (tức là các cấu trúc con mạng hoàn chỉnh của trọng số-kích hoạt-trọng số), mục tiêu chính của Ψ -NN trong quá trình trích xuất ma trận tham số là xác định các mối quan hệ giữa các tham số. Để tối đa hóa việc tinh chỉnh các mối quan hệ tham số, Ψ -NN giữ lại các đặc điểm dấu của tham số và tinh chỉnh các đối tượng phân cụm thành các giá trị tham số có thể

huấn luyện giữa mỗi cặp nút. Trên cơ sở này, Ψ -NN có thể vượt qua các chiến lược thua thót truyền thống chỉ đơn thuần hợp nhất các nút lặp lại, dẫn đến các ma trận tham số có ý nghĩa vật lý.

3.2.3 Tái cấu trúc mạng

Mục tiêu cốt lõi của giai đoạn tái cấu trúc là nâng cao khả năng ứng dụng của mạng trong khi vẫn bảo toàn tính phù hợp vật lý. Bằng cách tái cấu trúc ma trận quan hệ tham số đã trích xuất, Ψ -NN tạo ra một kiến trúc mạng nơ-ron không chỉ tích hợp các ràng buộc vật lý mà còn thích ứng với các bài toán mới. Không giống như các phương pháp chỉ dựa vào việc cắt tia hoặc đặt tham số về 0, việc tái cấu trúc Ψ -NN không chỉ nhấn mạnh việc nén tham số mà quan trọng hơn là bảo tồn và biểu diễn rõ ràng các mối quan hệ vật lý giữa các tham số. Kết quả là, mạng được tái cấu trúc vừa phản ánh các định luật vật lý cơ bản vừa cho phép khả năng chuyển giao cấu trúc và khả năng ứng dụng trên các vấn đề khác nhau.

Cụ thể, trước tiên ta đánh giá ma trận tham số được nhóm lại và làm thưa các tham số dư thừa hoặc không đáng kể bằng cách đặt chúng về 0, nhờ đó cải thiện hiệu quả tham số. Đối với các tập con tham số nổi bật (chẳng hạn như tâm cụm), ta thực hiện khởi tạo lại để khôi phục khả năng huấn luyện của chúng. Đồng thời, ma trận quan hệ cấu trúc được sử dụng để đảm bảo tính nhất quán của các mối quan hệ số học và các ràng buộc vật lý giữa các nút trong mạng được tái cấu trúc. Cách tiếp cận này đảm bảo rằng mạng vẫn giữ đủ mức độ tự do để học tập trong khi vẫn bảo tồn cấu trúc vật lý được trích xuất.

Ma trận quan hệ \mathbf{R} mã hóa các mối quan hệ cấu trúc giữa các tham số có thể huấn luyện được, với mỗi hàng về cơ bản được tạo ra từ một vectơ nhị phân đã được biến đổi. Cụ thể: (1) Nếu hai tập hợp tham số giống hệt nhau, các hàng tương ứng trong \mathbf{R} sẽ giống nhau, cho thấy sự chia sẻ tham số; (2) Nếu các tham số có độ lớn bằng nhau nhưng dấu ngược nhau, các phần tử hàng tương ứng trong \mathbf{R} là -1, biểu thị mối quan hệ đảo dấu; (3) Nếu có mối quan hệ hoán vị giữa các tham số, các hàng liên quan trong \mathbf{R} sẽ được hoán đổi tương ứng, phản ánh sự hoán vị tham số. Do đó, các phần tử của \mathbf{R} thường là -1, 0 hoặc 1, tương ứng với các mối quan hệ nghịch đảo, không liên quan và trực tiếp. Theo cách này, \mathbf{R} thể hiện một cách có hệ thống thông tin cấu trúc như chia sẻ tham số, dấu hiệu và sự sắp xếp, và được sử dụng trong giai đoạn tái cấu trúc cấu trúc để hạn chế biểu diễn tham số của mạng mới, từ đó đạt được sự nhúng có cấu trúc của thông tin vật lý.

Phương pháp này bảo toàn khả năng khớp lặp của mạng mới và nhúng cấu trúc vật lý vào mạng nơ-ron thông qua ma trận quan hệ \mathbf{R} , do đó cân bằng tính nhất quán vật lý với khả năng biểu đạt của mô hình. Bằng cách này, việc tái cấu trúc mạng của Ψ -NN không chỉ đạt được sự nén tham số và bảo toàn các mối quan hệ vật lý mà còn tăng cường đáng kể khả năng ứng dụng và khả năng giải thích của mạng thông qua cơ chế hợp nhất thích ứng có cấu trúc.

3.2.4 Toàn bộ quá trình triển khai

Thông qua toàn bộ quy trình chắt lọc, trích xuất cấu trúc và tái cấu trúc mạng, phương pháp Ψ -NN cuối cùng đạt được thiết kế nội tại của cấu trúc mạng với các ràng buộc vật lý. Các bước chính được mô tả như trong hình 3.1, có thể tóm tắt như sau:

1. Quá trình chắt lọc: Trong giai đoạn chắt lọc, việc lựa chọn mạng giáo viên không phải là duy nhất và phụ thuộc vào đặc điểm của vấn đề. Khi đã hiểu rõ vấn đề vật lý và có sẵn mô hình hiện đại (state-of-the-art), việc lựa chọn mô hình là điều nên làm, vì đầu ra của nó phản ánh chính xác hơn kịch bản vật lý thực sự. Việc tái tạo chính xác này kết hợp thông tin vật lý phong phú hơn vào dữ liệu được tạo ra, tạo điều kiện thuận lợi cho việc trích xuất các cấu trúc có liên quan đến vật lý hơn cho mô hình học sinh. Ở đây, PINN được sử dụng làm mạng giáo viên.

2. Trích xuất cấu trúc: Phương pháp trích xuất không chỉ bảo toàn cấu trúc vật lý đã học được mà còn giản hóa đáng kể kiến trúc mạng. Trong quá trình trích xuất, do sự hội tụ của các tham số cấu trúc, thuật toán phân cụm HAC nén các vectơ tham số thành các vectơ tâm cụm nhỏ hơn, nhờ đó chuyển đổi các đặc điểm vật lý thành ma trận tham số mạng và đảm bảo tính liên quan vật lý của khung mạng.

3. Tái cấu trúc mạng: Trong quá trình tái cấu trúc, Ψ -NN giữ lại tối đa các mối quan hệ tham số ban đầu trong khi chỉ khởi tạo lại các tham số có thể huấn luyện, dẫn đến cấu trúc mạng cuối cùng kết hợp tính liên quan vật lý. Cách tiếp cận này cho phép cấu trúc Ψ -NN tận dụng tối đa các tham số có thể huấn luyện, nâng cao hiệu quả tham số và cho phép áp dụng trên một phạm vi bài toán rộng hơn.

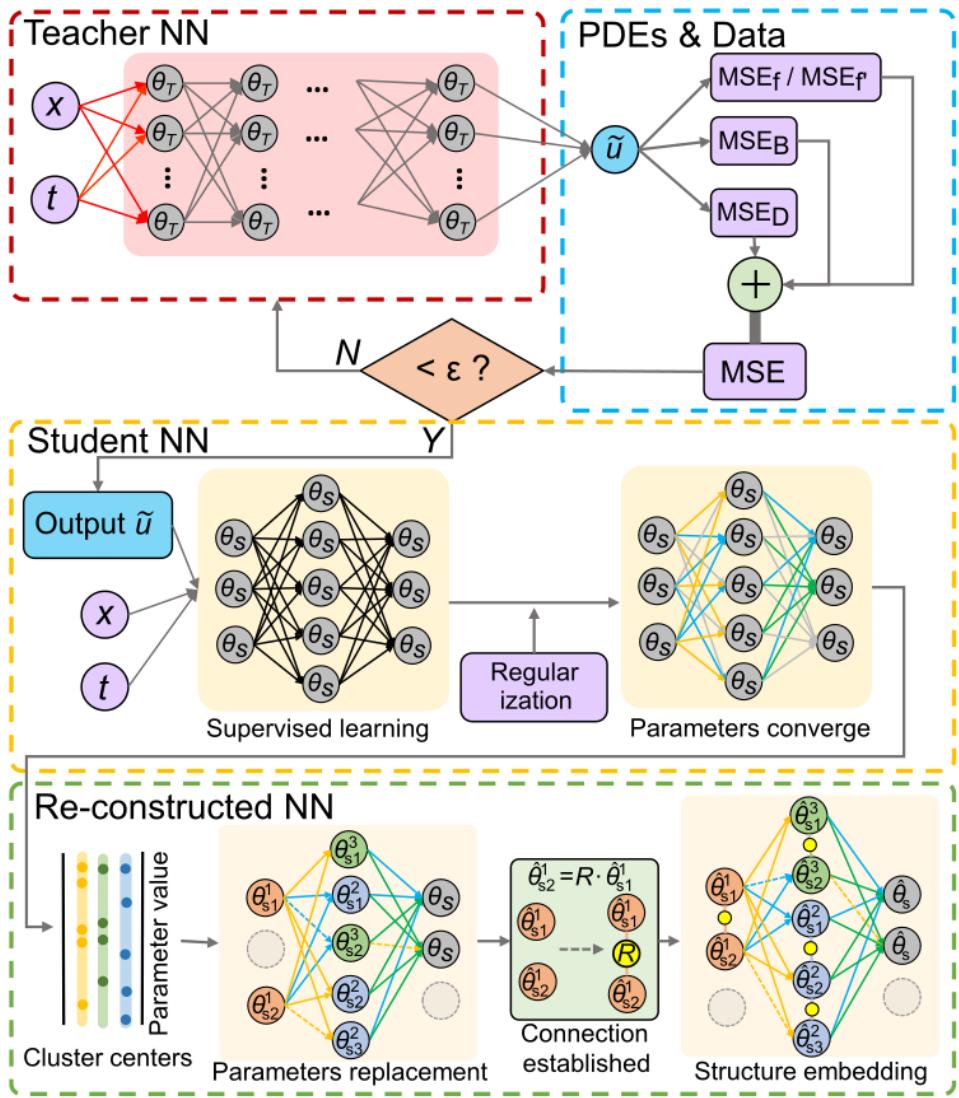
3.3 Kết quả

3.3.1 Mạng nơ-ron dựa trên thông tin cấu trúc vật lý.

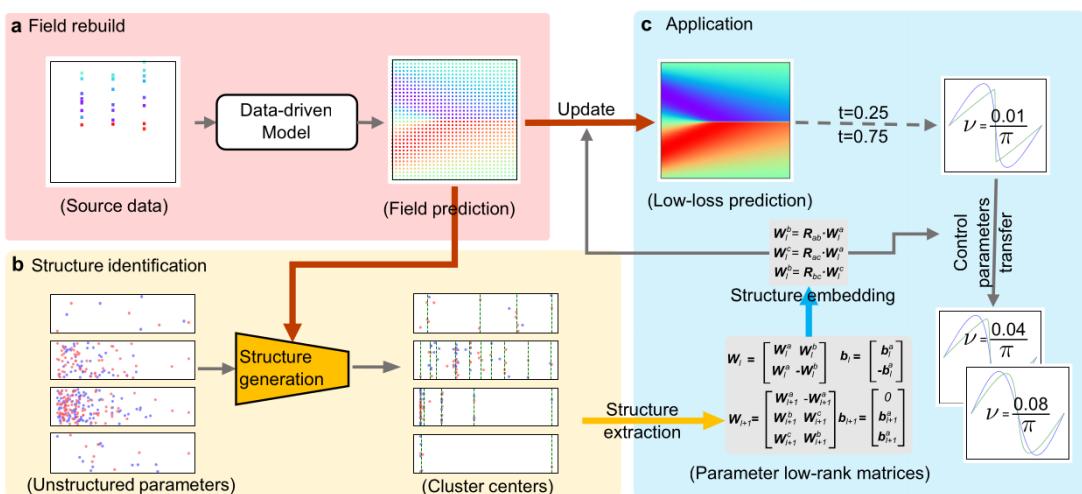
Trong các ví dụ, Ψ -NN đạt được mục tiêu trích xuất cấu trúc mạng từ dữ liệu theo các định luật vật lý chưa được biết rõ. Quy trình tổng thể sử dụng phương trình Burgers làm ví dụ được minh họa trong Hình 3.2, ba mũi tên đậm biểu thị quá trình chắt lọc - chiết xuất - tái tạo. Đầu tiên, toàn bộ dữ liệu thực tế được dự đoán bằng mô hình hướng dữ liệu, sau đó các mạng học sinh được huấn luyện bằng phương pháp tạo cấu trúc, sử dụng đầu ra của mô hình giáo viên. Cuối cùng, phương pháp trích xuất và tái tạo cấu trúc được áp dụng để tìm cấu trúc mạng tối ưu cho phương trình Burgers. Kết quả so sánh sai số l_2 được tóm tắt trong bảng 3.1, ở đây ta so sánh Ψ -PINN với PINN truyền thống và PINN-post [20, 25] (mô hình hậu xử lý các kết quả của PINN thông qua một bước ánh xạ cứng (hard mapping) ở lớp đầu ra để áp đặt thủ công các ràng buộc vật lý (như đối xứng, bảo toàn)). Chuyển đổi tham số điều khiển dễ dàng đến trường hợp dạng của PDE được cố định trong bài toán ngược, nhưng một số tham số nhất định (chẳng hạn như hệ số độ nhớt trong phương trình Burgers được hiển thị trong hình 3.2) thì thay đổi.

Problem	PINN	PINN-post	Ψ -NN
Laplace (1e-4)	11.59	4.017	0.7422
Burgers (1e-2)	14.47	3.014	1.287
Poisson (1e-2)	2.663	2.563	2.464

Bảng 3.1: So sánh sai số l_2 .



Hình 3.1: Thuật toán của mô hình Ψ -NN được đề xuất.



Hình 3.2: Quy trình của phương pháp Ψ -NN.

3.3.2 Trích xuất cấu trúc mạng từ PDE.

Các ví dụ chọn một số PDE tiêu biểu phương trình Laplace, phương trình Burgers, và phương trình Poisson - để sử dụng các mô hình cơ sở với các ràng buộc chặt tiên nghiệm, từ đó chứng minh rõ hơn khả năng khái quát hóa của Ψ -NN. Những bài toán này được sử dụng rộng rãi trong vật lý. Trong nhóm đối chứng, nghiên cứu sử dụng PINN với hậu xử lý ánh xạ cứng (PINN-post) cũng như PINN truyền thống. PINN-post đưa ra các ràng buộc bắt buộc bổ sung bằng cách hậu xử lý các đầu ra mạng, trong khi PINN truyền thống đóng vai trò là bộ giải NN tổng quát cho các PDE. Trong mọi trường hợp, thuật toán tối ưu Adam là được sử dụng để huấn luyện. Để đảm bảo khả năng tái tạo, random seed được cố định ở 1234.

Kết quả tính toán được thể hiện trong Hình 3.3. Cột đầu tiên bên trái hiển thị sai số bình trung bình (MSE), được sử dụng để so sánh tốc độ tối ưu hóa (mức giảm sai số trong cùng số bước) và độ chính xác tối ưu hóa (giá trị MSE cuối cùng) của các mô hình đại diện; cột thứ hai bên trái hiển thị giá trị thực tế của các trường hợp. Ba cột bên phải lần lượt là kết quả của PINN, PINN-post và Ψ -NN. Hàng đầu tiên hiển thị các dự đoán của mô hình, và hàng thứ hai hiển thị sai số tuyệt đối giữa mô hình và giá trị thực tế.

A. Phương trình Laplace

Fương trình Laplace có ứng dụng trong nhiều lĩnh vực, bao gồm điện trường, dẫn nhiệt và tĩnh học chất lỏng. Với các điều kiện biên và ban đầu thích hợp, phương trình này có thể thể hiện các tính chất đối xứng rõ ràng, mang lại các đặc điểm cấu trúc riêng biệt hơn cho mạng [8]. Phương trình Laplace được sử dụng để minh họa đầy đủ quá trình triển khai Ψ -NN và chứng minh tính khả thi của cấu trúc Ψ -NN.

Xét phương trình Laplace trạng thái ổn định trong tọa độ hai chiều $x \in \mathbb{R}^2$ với phương trình sau:

$$\mathcal{L} := \nabla^2 u = 0, \quad x \in [-1; 1]^2, \quad (3.8)$$

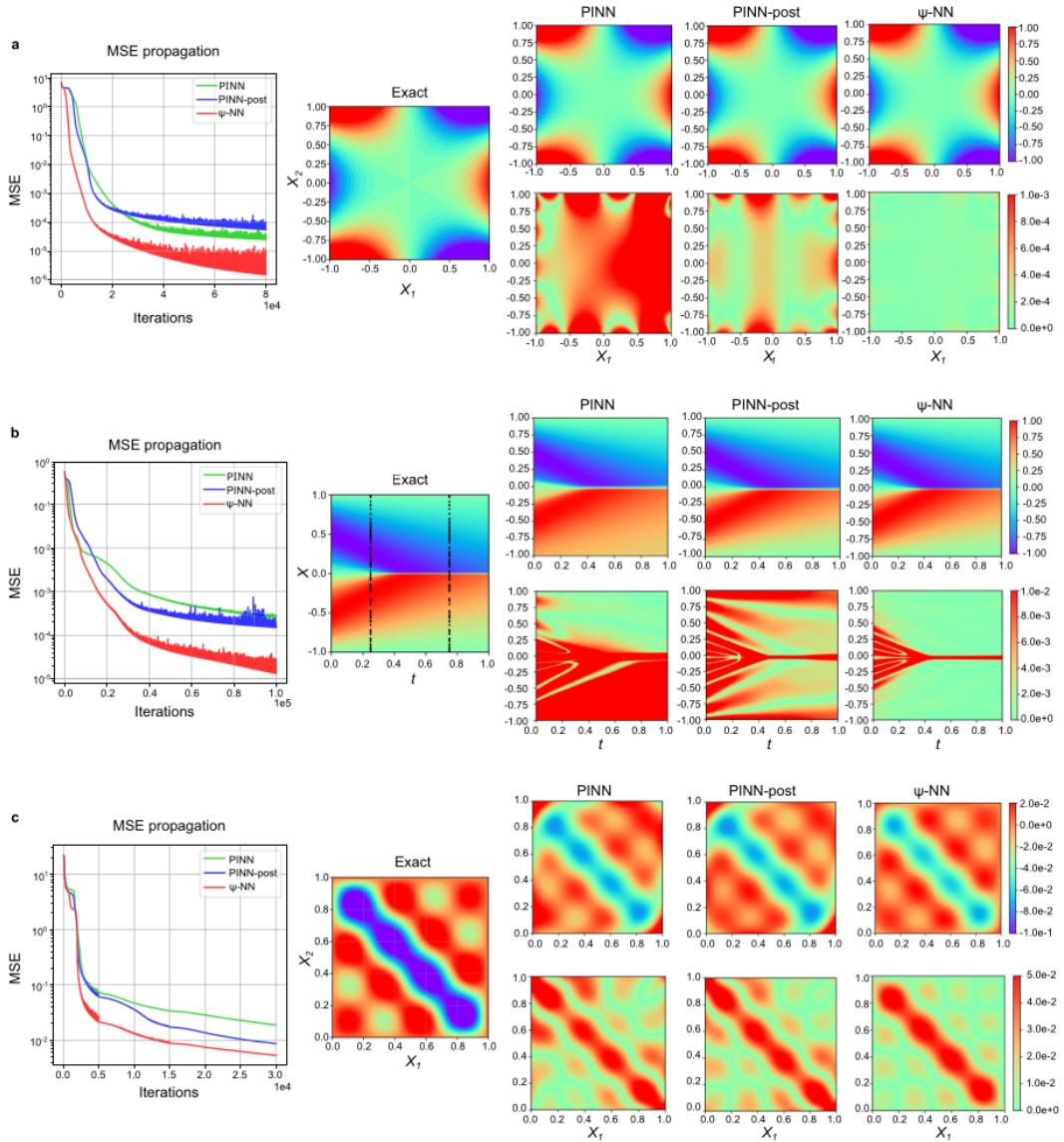
ở đây $x = (x_1, x_2)$, điều kiện biên như sau:

$$B := u = \begin{cases} -1 + 3x_2^2, & x_1 = -1 \\ 1 - 3x_2^2, & x_1 = 1 \\ x_1^3 - 3x_1, & x_2 = -1 \\ x_1^3 - 3x_1, & x_2 = 1 \end{cases} \quad (3.9)$$

Cấu trúc được trích xuất.

Fương pháp Ψ -NN cho phép trích xuất rõ ràng các cấu trúc mạng dưới sự hướng dẫn của các định luật vật lý, trong khi các phương pháp hiện có khác có thể ảnh hưởng tiêu cực đến độ chính xác của mạng. Hình 3.4a cho thấy sự tiến triển của các tham số lớp ẩn đầu tiên trong quá trình huấn luyện. Khi độ mờ mát của mạng học sinh ổn định, sự hội tụ tham số trở nên rõ rệt hơn, dẫn đến các cấu trúc mạng có thể trích xuất được. Hiện tượng hội tụ này được quan sát thấy trên các lớp khác nhau, và kết quả phân cụm tham số cuối cùng dưới Ψ -NN được thể hiện trong Hình 3.4b. Sự phân cụm của các độ lệch cũng hội tụ và tiến tới 0, làm giảm các đặc trưng độ lệch giữa các lớp và làm cho tính đối xứng trở nên rõ ràng hơn. Hình 3.4c cho thấy cấu trúc mạng sau khi thay thế các tham số ban đầu bằng các tâm cụm.

Vì cấu trúc lớp ẩn thứ hai bao gồm việc tái sử dụng, đảo dấu và hoán đổi, nghiên cứu lấy nó làm ví dụ để mô tả chi tiết sự hình thành ma trận quan hệ \mathbf{R}_2



Hình 3.3: a Kết quả phương trình Laplace. b Kết quả phương trình Burgers. c Kết quả phương trình Poisson.

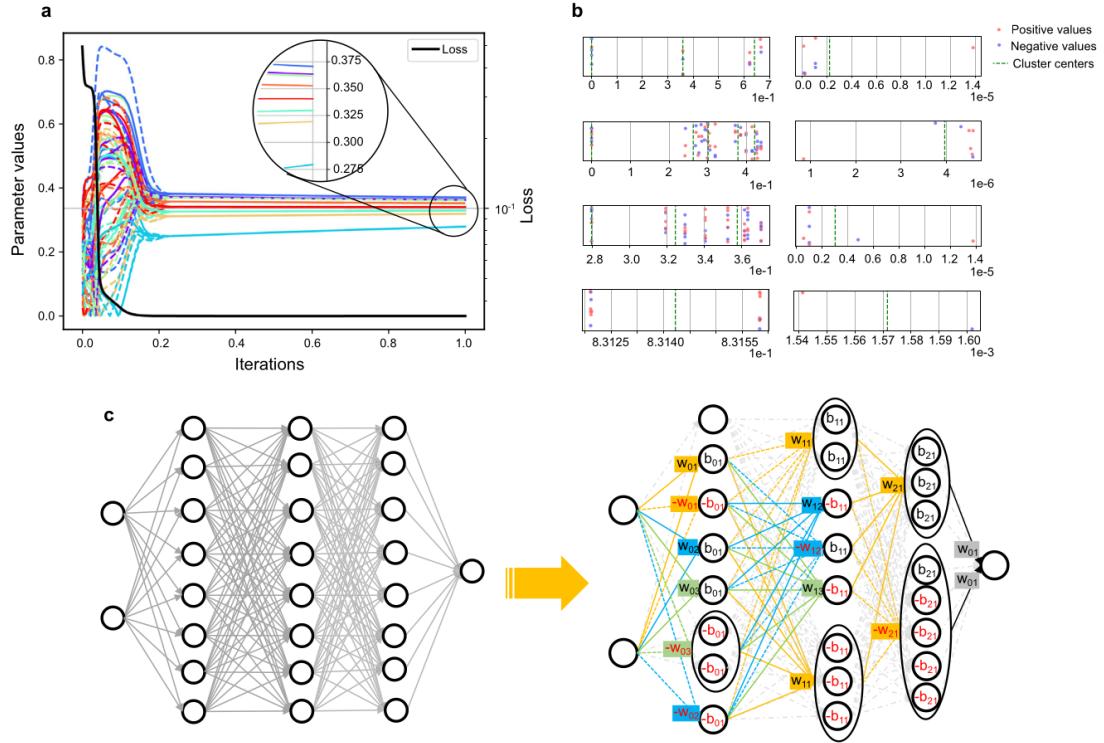
trong quá trình "trích xuất cấu trúc" và vai trò của nó trong quá trình "tái tạo mạng" của Ψ-NN. Chỉ số dưới cho biết lớp thứ hai. Đầu tiên, sau khi thay thế các tham số có thể huấn luyện của mạng học viên bằng các tâm cụm, ma trận tham số c_2 là:

$$c_2 = \begin{bmatrix} -c_2^a & -c_2^b \\ c_2^b & c_2^a \end{bmatrix} \quad (3.10)$$

ma trận tham số c_2 được xây dựng dưới dạng ma trận đường chéo với các tham số tâm cụm khác nhau được sắp xếp trên đường chéo và được ký hiệu bằng chỉ số mũ như sau:

$$c_2 = \begin{bmatrix} c_2^a & 0 \\ 0 & c_2^b \end{bmatrix} \quad (3.11)$$

Sau khi làm phẳng c_{S2} , chọn các tâm cụm bằng các vector one-hot và kết hợp các



Hình 3.4: a. Sự tiến triển của các tham số theo phương pháp Ψ -NN. b. Tâm cụm của Ψ -NN trong phương trình Laplace. c. Cấu trúc của mạng học sinh sau khi thay thế tham số trong phương trình Laplace.

mỗi quan hệ dấu, ma trận quan hệ \mathbf{R}_2 được biểu diễn như sau:

$$\mathbf{R}_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.12)$$

Trong ma trận này, các hàng có cùng tham số được nhân đôi, các hàng có dấu trái dấu bị âm, và việc hoán đổi các hàng 1, 2 và 4, 3 (bao gồm cả hoán đổi hàng và đảo dấu trong trường hợp này) biểu diễn mối quan hệ hoán đổi của các tham số. Ma trận quan hệ \mathbf{R}_2 lưu trữ các mối quan hệ giữa các tham số mạng, với mỗi hàng biểu diễn một tâm cụm được chọn. Do đó, trong quá trình tái cấu trúc mạng mới, ma trận tham số có thể huấn luyện của lớp ẩn đầu tiên θ_2 được ràng buộc bởi \mathbf{R}_2 như sau:

$$\theta_2 = \mathbf{R}_2 \cdot \begin{bmatrix} \theta_2^a & 0 \\ 0 & \theta_2^b \end{bmatrix} = \begin{bmatrix} -\theta_2^a & 0 \\ 0 & -\theta_2^b \\ 0 & \theta_2^b \\ \theta_2^a & 0 \end{bmatrix} \quad (3.13)$$

Sau khi sắp xếp các tham số có thể đào tạo khác không đã chọn theo thứ tự nút, kết quả thu được như sau:

$$\theta_2 = \begin{bmatrix} -\theta_2^a & -\theta_2^b \\ \theta_2^b & \theta_2^a \end{bmatrix} \quad (3.14)$$

chuyển sang dạng ma trận chuẩn, được W_2 như sau:

$$\mathbf{W}_2 = \begin{bmatrix} -W_2^a & -W_2^b \\ W_2^b & W_2^a \end{bmatrix} \quad (3.15)$$

Các lớp khác cũng tương tự. Cấu trúc này được biểu diễn thêm bằng các ma trận tham số bậc thấp, với các ma trận trọng số được ký hiệu là W_i^j , trong đó i chỉ lớp, và j là nhãn của ma trận con tương ứng. Độ lệch được ký hiệu là b . Kiến trúc được thể hiện như sau:

$$\mathbf{W}_1 = \begin{bmatrix} W_1^a & W_1^b \\ -W_1^a & -W_1^b \end{bmatrix}, b_1 = \begin{bmatrix} b_1^a \\ -b_1^a \end{bmatrix} \quad (3.16)$$

$$\mathbf{W}_2 = \begin{bmatrix} -W_2^a & -W_2^b \\ W_2^b & W_2^a \end{bmatrix}, b = \begin{bmatrix} b_2^a \\ b_2^a \end{bmatrix} \quad (3.17)$$

$$\mathbf{W}_3 = [W_3^a \quad W_3^b] \quad (3.18)$$

Ma trận bậc thấp này được tái tạo thông qua nhúng cấu trúc (nhúng ma trận quan hệ \mathbf{R} không đổi vào một mạng mới), trong đó các tham số của các nút d (chiều ma trận con tham số) được biểu diễn bằng $W_1^a = [w_{11}^a \ w_{12}^a \dots w_{1d}^a]^T$.

Bằng cách mở rộng biểu thức của hai tập hợp các nút của lớp ẩn cuối cùng, mỗi quan hệ giữa đầu vào hai chiều $x = (x_1, x_2)$ và đầu ra của nút u_a và u_b có thể thu được:

$$u_a(x_1, x_2) = \tanh[W_2^a \cdot \tanh(W_1^a \cdot x_1 + W_1^b \cdot x_2 + b_1^a) + W_2^b \cdot (\tanh(W_1^a \cdot x_1 - W_1^b \cdot x_2 + b_1^a)) + b_2^a], \quad (3.19)$$

$$u_b(x_1, x_2) = \tanh[W_2^b \cdot \tanh(W_1^a \cdot x_1 + W_1^b \cdot x_2 + b_1^a) + W_2^a \cdot (\tanh(W_1^a \cdot x_1 - W_1^b \cdot x_2 + b_1^a)) + b_2^a]. \quad (3.20)$$

Từ 2 biểu thức này, ta có:

$$u_a(x_1, x_2) = u_b(x_1, -x_2), \quad (3.21)$$

cuối cùng, ta có biểu thức đầu ra:

$$u(x_1, x_2) = W_3^a[u_a(x_1, x_2) + u_b(x_1, -x_2)], \quad (3.22)$$

với W_3^a là ma trận tham số. Biểu thức trên còn có thể viết như sau:

$$u(x_1, x_2) = W_3^a[u_a(x_1, x_2) + u_a(x_1, -x_2)]. \quad (3.23)$$

Do đó, phương pháp có cấu trúc này ngầm nhúng tính đối xứng có trong PINN-post. Các phương pháp PINN có cấu trúc trước đây yêu cầu các tiên nghiệm đối xứng và dựa vào việc áp đặt thủ công phương sai nhóm, nhằm mục tiêu vào các bài toán cụ thể, trong khi cấu trúc mạng trong Ψ -NN được tự động trích xuất từ dữ liệu và các ràng buộc vật lý, giảm sự phụ thuộc vào thiết kế thủ công và các kiến thức có từ trước.

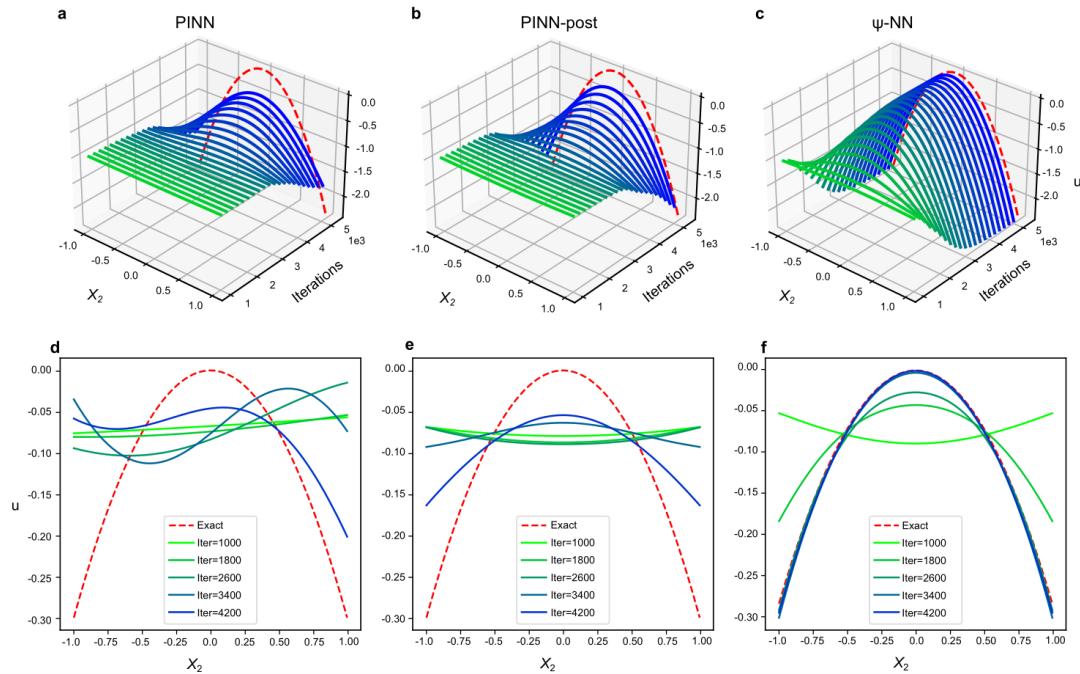
So sánh Ψ -NN và PINN.

Kết quả được thể hiện trong Hình 3.3a, và sai số l_2 được tóm tắt trong Bảng 3.1. So với PINN, Ψ -NN giảm khoảng 50% số lần lặp cần thiết để đạt được cùng sai số (10^{-3}) và giảm sai số l_2 khoảng 95%. Như minh họa trong Hình 3.5a và d, PINN không thể hiện tính đối xứng nhất quán trong quá trình huấn luyện, trong khi các ràng buộc về cấu trúc trong Ψ -NN, phù hợp với các đặc điểm của PDE, cho phép thu hẹp không gian tìm kiếm và cho phép tìm ra lời giải nhanh hơn và chính xác hơn trong giai đoạn đầu của quá trình huấn luyện.

So sánh Ψ -NN và PINN-post.

PINN-post kết hợp tính đối xứng vào lớp đầu ra của mạng thông qua các ràng buộc rõ ràng, tạo ra các đầu ra đáp ứng tốt hơn các tính chất vật lý đối xứng - giảm lỗi l_2 khoảng 65% so với PINN thông thường, đặc biệt là trong miền tính toán. Tuy nhiên, MSE hội tụ của PINN-post cao hơn MSE của Ψ -NN, cho thấy giá trị mất mát tối thiểu trong khuôn khổ PINN không phản ánh đầy đủ độ chính xác thực sự của mạng, mà chỉ phản ánh giá trị trung bình phù hợp với dữ liệu và PDE có sẵn. Tốc độ giảm sai số phản ánh tốc độ hội tụ: để đạt được sai số 10^{-3} , PINN-post cần ít hơn 5000 lần lặp so với PINN.

Cả Ψ -NN và PINN-post đều nhúng tính đối xứng không gian vào mạng, nhưng điểm khác biệt chính là kiến trúc Ψ -NN được tái tạo vốn đã chứa đặc tính vật lý này, trong khi PINN-post áp dụng ánh xạ cứng như một bước hậu xử lý tại lớp đầu ra. Về thời gian tính toán: thời gian tính toán cho PINN-post là 32.68 phút, dài hơn 29.87 phút của Ψ -NN. Hơn nữa, Ψ -NN đạt sai số 10^{-4} với số lần lặp ít hơn 15000, và tốc độ hội tụ trung bình của nó gấp khoảng hai lần so với PINN-post. Sai số l_2 được giảm từ 1.159×10^{-3} xuống 7.422×10^{-5} . Như thể hiện trong Hình 3.3a, ràng buộc ánh xạ cứng trong PINN-post không làm giảm các sai số tính toán lớn gần biên. Điều này cho thấy, do các ràng buộc ngầm định phong phú trong các trường vật lý, việc nhúng thủ công các đặc trưng thông qua hậu xử lý chỉ cung cấp một ràng buộc cần thiết nhưng không đủ, và khả năng ứng dụng của nó có thể bị giới hạn trong một phạm vi nhất định. Ngược lại, Ψ -NN khám phá các cấu trúc mạng hoàn toàn dựa trên dữ liệu quan sát và PDE, nhằm mục đích tự động nhúng tất cả thông tin đã biết về bài toán vật lý vào cấu trúc mạng, do đó giảm thiểu sai số trên một phạm vi tính toán rộng hơn.



Hình 3.5: So sánh dự đoán theo số lần lặp. a,d: PINN; b,e: PINN-post; c,f: Ψ -NN.

B.Phương trình Burgers

Phương trình Burgers là một công cụ quan trọng để mô tả các hiện tượng sóng phi tuyến, thường được sử dụng để nghiên cứu các hệ thống phức tạp như động lực

học chất lưu và hành vi sóng. Ví dụ chọn phương trình Burgers do sự hình thành xung kích rõ rệt của nó và các đặc tính phản đối xứng do đó, giúp xác nhận lợi thế về hiệu suất của mạng có cấu trúc và khả năng ứng dụng của chúng trên một loạt các biến thể tham số [8].

Trong bài toán ngược, hệ số độ nhớt trong phương trình Burgers được thay thế bằng một tham số λ_1 chưa biết, và phương trình đạo hàm riêng trở thành:

$$\mathcal{L} := u_t - uu_x - \lambda_1 u_{xx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1]. \quad (3.24)$$

Kết quả trích xuất và so sánh hiệu năng.

Trong quá trình trích xuất Ψ -NN, lấy lớp ẩn thứ ba làm ví dụ, các biến thể tham số được hiển thị trong hình 3.5. Các ma trận tham số hạng thấp được trích xuất là:

$$\mathbf{W}_1 = \begin{bmatrix} W_1^a & W_1^b \\ W_1^a & -W_1^b \end{bmatrix}, b_1 = \begin{bmatrix} b_1^a \\ -b_1^a \end{bmatrix} \quad (3.25)$$

$$\mathbf{W}_2 = \begin{bmatrix} W_2^a & -W_2^a \\ W_2^b & W_2^c \\ W_2^c & W_2^b \end{bmatrix}, b_2 = \begin{bmatrix} 0 \\ b_2^a \\ b_2^a \end{bmatrix} \quad (3.26)$$

$$\mathbf{W}_3 = [W_3^a \quad W_3^b \quad -W_3^b], b_3 = 0 \quad (3.27)$$

Tương tự, mỗi quan hệ giữa đầu vào hai chiều $x = (x_1, x_2)$ và đầu ra nút u_a, u_b, u_c của lớp ẩn cuối cùng có thể được thu được:

$$u_a(x_1, x_2) = [\tanh(W_2^a \cdot (\tanh W_1^a \cdot x_1 + W_1^b \cdot x_2 + b_1^a)) - W_2^a \cdot (\tanh W_1^a \cdot x_1 - W_1^b \cdot x_2 - b_1^a)], \quad (3.28)$$

$$u_b(x_1, x_2) = [\tanh(W_2^b \cdot (\tanh W_1^a \cdot x_1 + W_1^b \cdot x_2 + b_1^a)) + W_2^c \cdot (\tanh W_1^a \cdot x_1 - W_1^b \cdot x_2 - b_1^a) + b_2^a], \quad (3.29)$$

$$u_c(x_1, x_2) = [\tanh(W_2^c \cdot (\tanh W_1^a \cdot x_1 + W_1^b \cdot x_2 + b_1^a)) + W_2^b \cdot (\tanh W_1^a \cdot x_1 - W_1^b \cdot x_2 - b_1^a) + b_2^a]. \quad (3.30)$$

Từ 3 biểu thức trên ta có:

$$\lim_{b_1^a \rightarrow 0} u_a(x_1, -x_2) = -u_a(x_1, x_2) \quad (3.31)$$

$$u_b(x_1, x_2) = u_c(x_1, -x_2).$$

Biểu thức của đầu ra cuối cùng là :

$$u(x_1, x_2) = W_3^a \cdot u_a + W_3^b \cdot (u_b - u_c). \quad (3.32)$$

với W_3^a, W_3^b là các ma trận tham số.

Do đó, nửa trái của u ($W_3^a \cdot u_a$), chứa tính đối xứng của PINN-post dưới điều kiện $\lim_{b_1^a \rightarrow 0}$, trong khi nửa còn lại ($W_3^b \cdot (u_b - u_c)$), trực tiếp chứa tính đối xứng của PINN-post. Ngoài ra, các tính chất của u_a trong lớp ẩn thứ hai cũng chỉ ra rằng sự xuất hiện của tính đối xứng không hoàn toàn phụ thuộc vào số lượng lớp mạng.

Xu hướng của hàm mất mát được thể hiện trong Hình 3.3b. Mạng có cấu trúc được tái tạo của Ψ -NN thể hiện tốc độ lặp lại nhanh hơn đáng kể trong quá trình huấn luyện so với PINN và PINN-post, đạt được độ chính xác hàm mất mát tối thiểu là 10^{-5} , thấp hơn so với hai mô hình còn lại.

Sai số l_2 được tóm tắt trong bảng 3.1. Cả PINN-post và Ψ -NN đều đạt được lỗi thấp hơn PINN, chứng minh tính hiệu quả của việc nhúng các đặc trưng phương trình vào cấu trúc mạng để cải thiện độ chính xác. Như thể hiện trong Hình 3.3b, sau khi hình thành xung đột tại $t > 0.4$, cả PINN và PINN-post đều thể hiện phân phôi lỗi lớn, trong khi Ψ -NN giảm đồng đều các lỗi ở cả hai phía của xung đột. Hơn nữa, PINN-post, do tính đối xứng hậu xử lý của nó, áp đặt phân phôi lỗi đối xứng trên toàn bộ xung đột nhưng không làm giảm lỗi thực tế. Khả năng khám phá cấu trúc của Ψ -NN cung cấp một không gian đặc trưng chính xác và phù hợp hơn, dẫn đến sai số thấp nhất.

Dối với bài toán nghịch, ngoài việc tái cấu trúc toàn bộ trường, một nhiệm vụ quan trọng khác là ước tính tham số chưa biết λ trong Phương trình (3.24), với giá trị thực $\lambda_1 = 0.01/\pi$. Kết quả cuối cùng được đưa ra trong Bảng 3.2, trong đó Ψ -NN đạt được giá trị gần nhất với giá trị thực tế. Sự tiến triển của tham số này trong quá trình huấn luyện được thể hiện trong Hình 3.6. Vì λ_1 được bao gồm trong vectơ tham số có thể huấn luyện của NN, các đường cong này có thể được hiểu là quỹ đạo tối ưu; đường đi ngắn hơn biểu thị hướng tìm kiếm rõ ràng hơn và hội tụ nhanh hơn. Do đó, các đặc điểm cấu trúc được Ψ -NN phát hiện làm giảm không gian đầu ra và giúp việc tìm ra lời giải chính xác dễ dàng hơn.

Hiệu suất của cấu trúc Ψ -NN với các trường hợp tham số khác nhau.

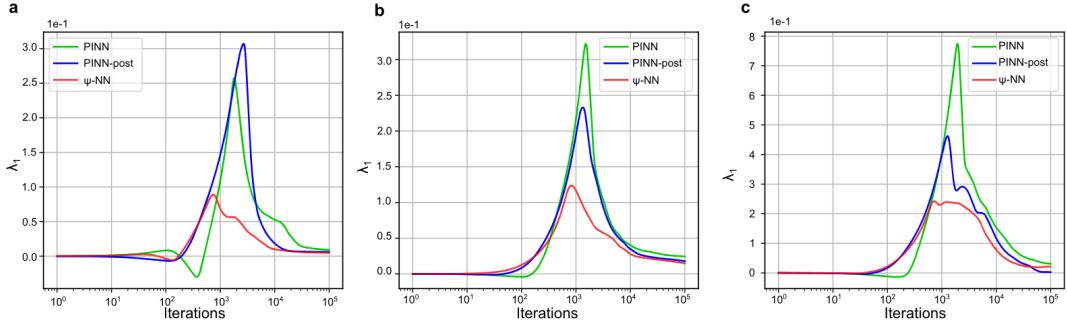
Số hạng độ nhót v cần giải trong bài toán ngược được biểu diễn bằng tham số λ_1 chưa biết. Điều này cho phép nghiên cứu xác thực khả năng ứng dụng của cấu trúc được tái tạo cho một bài toán cụ thể dưới các tham số khác nhau. Nghiên cứu đã tiến hành các thí nghiệm sử dụng cùng một cấu trúc Ψ -NN được tái tạo mà không sửa đổi các cấu hình khác, cụ thể là ở $v_2 = 0.04/\pi$ và $v_3 = 0.08/\pi$. Kết quả tính toán được thể hiện trong Hình 3.6, trong đó phương pháp Ψ -NN duy trì các đường đi ngắn hơn qua các tham số khác nhau. Đường đi ngắn hơn qua các tham số khác nhau cho thấy Ψ -NN có quy trình tối ưu hóa hiệu quả hơn trong không gian tham số. Kết quả dự đoán cuối cùng được tóm tắt trong Bảng 3.2, với các giá trị gần nhất với giá trị thực tế.

Mô hình	Thực tế	PINN	PINN-post	Ψ -NN
$v_1.\pi(1e-2)$	1	2.820	1.898	1.465
$v_2.\pi(1e-2)$	4	6.625	4.960	4.084
$v_3.\pi(1e-2)$	8	9.705	0.885	6.673

Bảng 3.2: So sánh điều khiển tham số của phương trình Burgers

C.Phương trình Poisson

Phương trình Poisson đóng một vai trò quan trọng trong nhiều phép tính, bao gồm dẫn nhiệt, điện từ và trường hấp dẫn [8]. Ở đây, ví dụ chọn một bài toán Poisson trong miền hình vuông đơn vị với một nguồn trơn $f(x_1, x_2)$ chứa bốn tần số tăng. Lựa chọn này cho phép chứng minh khả năng ứng dụng và hiệu suất của phương pháp Ψ -NN trên các tham số khác nhau. Các hệ thống vật lý tần số cao thường thể hiện các cấu trúc đối xứng vốn có. Để giải quyết vấn đề này, ta sử dụng Ψ -NN để khám phá và tận dụng các mẫu đối xứng có trong bài toán, giảm thiểu hiệu quả các thách thức liên quan đến các đặc tính tần số cao và nâng cao hiệu quả



Hình 3.6: So sánh dự đoán dưới các tham số điều khiển khác nhau của phương trình Burgers. a. $v_1 = 0.01/\pi$. b. $v_2 = 0.04/\pi$. c. $v_3 = 0.08/\pi$.

cũng như độ chính xác của mô hình. Cụ thể, phương trình Poisson thỏa mãn ràng buộc:

$$\mathcal{L} := \nabla^2 u = f(x_1, x_2), \quad x_1, x_2 \in [0, 1], \quad (3.33)$$

với nguồn được cho bởi:

$$f(x_1, x_2) = \frac{1}{4} \sum_{k=1}^4 (-1)(k+1)2k^2 \sin(k\pi x_1) \sin(k\pi x_2). \quad (3.34)$$

Hàm f có tính đối xứng hoán vị.

Nghiệm của phương trình trên là $u = x_1^2 - x_2^2$. Ma trận tham số hạng thấp được trích xuất cho hàm tần số thấp này là:

$$\mathbf{W}_1 = \begin{bmatrix} W_1^a & 0 \\ 0 & W_1^a \end{bmatrix}, b_1 = \begin{bmatrix} b_1^a \\ -b_1^a \end{bmatrix} \quad (3.35)$$

$$\mathbf{W}_2 = \begin{bmatrix} W_2^a & W_2^b \\ W_2^b & W_2^a \end{bmatrix}, b_2 = \begin{bmatrix} b_2^a \\ -b_2^a \end{bmatrix} \quad (3.36)$$

$$\mathbf{W}_3 = \begin{bmatrix} W_3^a & W_3^a \end{bmatrix}, b_3 = 0 \quad (3.37)$$

Tương tự, mối quan hệ giữa đầu vào hai chiều $x = (x_1, x_2)$ và đầu ra của nút u_a, u_b của lớp ẩn cuối cùng có thể được thu được:

$$u_a(x_1, x_2) = [\tanh(W_2^a \cdot (\tanh W_1^a \cdot x_1 + b_1^a) + W_2^b \cdot (\tanh W_1^a \cdot x_2 - b_1^a) + b_2^a)], \quad (3.38)$$

$$u_b(x_1, x_2) = [\tanh(W_2^b \cdot (\tanh W_1^a \cdot x_1 + b_1^a) + W_2^a \cdot (\tanh W_1^a \cdot x_2 - b_1^a) - b_2^a)]. \quad (3.39)$$

Từ 2 biểu thức trên ta có:

$$u_a(x_1, x_2) = -u_b(-x_2, -x_1). \quad (3.40)$$

Từ đó thu được biểu thức đầu ra cuối cùng:

$$u(x_1, x_2) = W_3^a(u_a + u_b). \quad (3.41)$$

Để kết quả cấu trúc mạng phù hợp với đặc điểm của nghiệm tần số cao, bằng cách đặt dấu của các giá trị trong b_1 giống nhau, chúng ta có thể thu được:

$$\bar{u}_a(x_1, x_2) = [\tanh(W_2^a \cdot (\tanh W_1^a \cdot x_1 + b_1^a) + W_2^b \cdot (\tanh W_1^a \cdot x_2 + b_1^a) + b_2^a)] \quad (3.42)$$

$$\begin{aligned}\bar{u}_b(x_1, x_2) = & [\tanh(W_2^b \cdot (\tanh W_1^a \cdot x_1 + b_1^a) \\ & + W_2^a \cdot (\tanh W_1^a \cdot x_2 + b_1^a) - b_2^a)],\end{aligned}\tag{3.43}$$

từ đây ta có:

$$\bar{u}_a(x_1, x_2) = \bar{u}_b(x_2, x_1),\tag{3.44}$$

và biểu thức cuối cùng:

$$\bar{u}(x_1, x_2) = W_3^a(\bar{u}_a + \bar{u}_b).\tag{3.45}$$

Hơn nữa, cấu trúc có thể được điều chỉnh bằng dấu để thỏa mãn các dạng đối xứng khác. Ví dụ, sau (3.40), việc điều chỉnh dấu của các trọng số trong lớp ẩn cuối cùng có thể mang lại:

$$u_r = W_3^a \cdot (u_a - u_b) = u_a(x_1, x_2) + u_a(-x_1, -x_2).\tag{3.46}$$

Kết quả số được thể hiện trong Hình 3.3c. Kích thước bước lặp được sử dụng trong bài toán này là 10^{-3} , giảm xuống còn 0,2 lần giá trị ban đầu ở các bước 5000 và 1500. Mạng NN tái tạo Ψ -NN vượt trội hơn cả mô hình PINN và PINN-post về tốc độ và độ chính xác. Sai số l_2 của chúng được tóm tắt trong Bảng 3.1. Cả ba mô hình đều có sai số đỉnh dọc theo đường $x_2 = -x_1 + 1$, nhưng Ψ -NN duy trì sai số tổng thể thấp hơn, đặc biệt là tại các biên. Ngược lại, các mô hình khác trong nhóm đối chứng cho thấy sai số gradient lớn tại các biên cục bộ, làm nổi bật hiệu suất vượt trội của Ψ -NN trong việc khớp tần số cao.

Tổng kết

Đồ án này đã trình bày một cách hệ thống về Mạng nơ-ron thông tin vật lý (PINN) và mô hình cải tiến trong việc giải xấp xỉ các phương trình đạo hàm riêng (PDE). Qua ba chương nội dung chính, đồ án đã làm rõ các khía cạnh lý thuyết, phương pháp đánh giá hiệu quả, và hướng phát triển tự động hóa cấu trúc mạng dựa trên thông tin vật lý.

Giới thiệu về PINN

PINN là một phương pháp kết hợp giữa học sâu và kiến thức vật lý, cho phép giải xấp xỉ các PDE ngay cả khi dữ liệu huấn luyện ít và phân tán. Bằng cách nhúng các ràng buộc vật lý vào hàm mất mát thông qua vi phân tự động, PINN không chỉ học từ dữ liệu mà còn tuân thủ các định luật vật lý cơ bản. Các ví dụ minh họa với phương trình Schrödinger, Allen-Cahn, Navier-Stokes và KdV đã chứng minh khả năng ứng dụng rộng rãi của PINN trong cả bài toán thuận và bài toán ngược.

Đánh giá hiệu quả của PINN

Chương 2 đã đề xuất một phương pháp đánh giá định lượng dựa trên ba chỉ số: sai số tuyệt đối trung bình $||\Delta||_1$, sai số cực đại l_∞ và chuẩn l_2 . Các ví dụ với các phương trình Burgers, Fisher, Burgers-Huxley và Newell-Whitehead-Segel cho thấy sự hiệu quả của PINN trong việc giải xấp xỉ các PDE phản ứng - đối lưu - khuyếch tán.

Mô hình PINN tự động khám phá cấu trúc: Ψ -NN

Chương 3 giới thiệu mô hình Ψ -NN, một biến thể cải tiến của PINN, có khả năng tự động trích xuất và tái cấu trúc mạng nơ-ron dựa trên thông tin vật lý từ PDE. Bằng cách sử dụng cơ chế chắt lọc kiến thức (knowledge distillation) kết hợp với phân cụm tham số và tái cấu trúc mạng, Ψ -NN không chỉ cải thiện độ chính xác mà còn tăng tốc độ hội tụ và khả năng diễn giải của mô hình. Các kết quả thử nghiệm trên phương trình Laplace, Burgers và Poisson cho thấy Ψ -NN vượt trội hơn so với PINN truyền thống và PINN-post về cả sai số và tốc độ huấn luyện.

Kết luận chung

PINN đại diện cho một bước tiến quan trọng trong việc tích hợp học máy và mô hình hóa vật lý, mang lại giải pháp hiệu quả cho việc giải các PDE trong điều kiện dữ liệu hạn chế. Với sự phát triển của các biến thể như Ψ -NN, khả năng tự động hóa và tối ưu hóa cấu trúc mạng sẽ tiếp tục mở rộng phạm vi ứng dụng của phương pháp này trong khoa học tính toán và kỹ thuật.

Tài liệu tham khảo

- [1] W. A. Strauss, *Partial differential equations: An introduction*. John Wiley & Sons, 2007.
- [2] E. Alpaydin, *Machine learning*. MIT press, 2021.
- [3] M. Schmidt and H. Lipson, “Distilling free-form natural laws from experimental data,” *science*, vol. 324, no. 5923, pp. 81–85, 2009.
- [4] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [5] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
- [6] 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,” *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.
- [7] F. Hasan, H. Ali, and H. A. Arief, “From mesh to neural nets: A multi-method evaluation of physics informed neural network and galerkin finite element method for solving nonlinear convection–reaction–diffusion equations,” *International Journal of Applied and Computational Mathematics*, vol. 11, no. 3, pp. 1–24, 2025.
- [8] Z. Liu, L. Yang, X. Yan, L. Wen, H. Nie, S.-Q. Guo, and C.-a. Zhang, “Automatic network structure discovery of physics informed neural networks via knowledge distillation,” *Nature Communications*, vol. 16, 10 2025.
- [9] A. Tizghadam, H. Khazaei, M. H. Moghaddam, and Y. Hassan, “Machine learning in transportation.,” *Journal of Advanced Transportation*, 2019.
- [10] H. Habehh and S. Gohel, “Machine learning in healthcare,” *Current genomics*, vol. 22, no. 4, pp. 291–300, 2021.
- [11] M. W. Libbrecht and W. S. Noble, “Machine learning applications in genetics and genomics,” *Nature Reviews Genetics*, vol. 16, no. 6, pp. 321–332, 2015.
- [12] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

- [13] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.
- [14] I. K. M. Jais and A. R. Ismail, “Adam optimization algorithm for wide and deep neural network,” *Knowledge Engineering and Data Science*, vol. 2, no. 1, p. 10, 2019.
- [15] W.-L. Loh, “On latin hypercube sampling,” *The annals of statistics*, vol. 24, no. 5, pp. 2058–2080, 1996.
- [16] A. Iserles, *A first course in the numerical analysis of differential equations*. No. 44, Cambridge university press, 2009.
- [17] T. A. Driscoll, N. Hale, and L. N. Trefethen, “Chebfun guide,” 2014.
- [18] J. Zhang, T. He, S. Sra, and A. Jadbabaie, “Why gradient clipping accelerates training: A theoretical justification for adaptivity,” *arXiv preprint arXiv:1905.11881*, 2019.
- [19] T. Hastie, “Ridge regularization: An essential concept in data science,” *Tech-nometrics*, vol. 62, no. 4, pp. 426–433, 2020.
- [20] Y. Chen, D. Huang, D. Zhang, J. Zeng, N. Wang, H. Zhang, and J. Yan, “Theory-guided hard constraint projection (hcp): A knowledge-based data-driven scientific machine learning method,” *Journal of Computational Physics*, vol. 445, p. 110624, 2021.
- [21] Y. Shin, Z. Zhang, and G. E. Karniadakis, “Error estimates of residual minimization using neural networks for linear pdes,” 2023.
- [22] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *Journal of structural biology*, vol. 181 2, pp. 116–27, 2010.
- [23] J. H. Ward, “Hierarchical grouping to optimize an objective function,” *Journal of the American Statistical Association*, vol. 58, pp. 236–244, 1963.
- [24] D. Zhang, H. Wang, M. A. T. Figueiredo, and L. Balzano, “Learning to share: simultaneous parameter tying and sparsification in deep learning,” in *International Conference on Learning Representations*, 2018.
- [25] J. N. Hendriks, C. Jidling, A. G. Wills, and T. B. Schön, “Linearly constrained neural networks,” *ArXiv*, vol. abs/2002.01600, 2020.