

Active learning attraction basins of dynamical system

Xiao-Wei Cao(曹小尾), Xiao-Lei Ru(茹小磊), and Gang Yan(严钢)

Citation: Chin. Phys. B, 2025, 34 (**5**): 058901. DOI: 10.1088/1674-1056/adbede
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Dynamic modeling and analysis of brucellosis on metapopulation network: Heilongjiang as cases

Xin Pei(裴鑫), Xuan-Li Wu(武绚丽), Pei Pei(裴沛), Ming-Tao Li(李明涛), Juan Zhang(张娟), and Xiu-Xiu Zhan(詹秀秀)
Chin. Phys. B, 2025, 34 (**1**): 018904. DOI: 10.1088/1674-1056/ad92ff

Important edge identification in complex networks based on local and global features

Jia-Hui Song(宋家辉)
Chin. Phys. B, 2023, 32 (**9**): 098901. DOI: 10.1088/1674-1056/aca6d8

Voter model on adaptive networks

Jinming Du(杜金铭)
Chin. Phys. B, 2022, 31 (**5**): 058902. DOI: 10.1088/1674-1056/ac43b4

Contagion dynamics on adaptive multiplex networks with awareness-dependent rewiring

Xiao-Long Peng(彭小龙) and Yi-Dan Zhang(张译丹)
Chin. Phys. B, 2021, 30 (**5**): 058901. DOI: 10.1088/1674-1056/abe1ab

Traffic dynamics considering packet loss in finite buffer networks

Jie Chen(陈杰), Jin-Yong Chen(陈金邕), Ming Li(李明), Mao-Bin Hu(胡茂彬)
Chin. Phys. B, 2019, 28 (**4**): 048901. DOI: 10.1088/1674-1056/28/4/048901

SPECIAL TOPIC — Computational programs in complex systems

Active learning attraction basins of dynamical system

Xiao-Wei Cao(曹小尾)[1], Xiao-Lei Ru(茹小磊)[1,2,†], and Gang Yan(严钢)[1,2,‡]

[1] *MOE Key Laboratory of Advanced Micro-Structured Materials, and School of Physical Science and Engineering, Tongji University, Shanghai* 200092, *China*
[2] *National Key Laboratory of Autonomous Intelligent Unmanned Systems, MOE Frontiers Science Center for Intelligent Autonomous Systems, and Shanghai Research Institute of Intelligence Science and Technology, Tongji University, Shanghai* 200092, *China*

(Received 2 December 2024; revised manuscript received 2 March 2025; accepted manuscript online 11 March 2025)

Dynamical systems often exhibit multiple attractors representing significantly different functioning conditions. A global map of attraction basins can offer valuable guidance for stabilizing or transitioning system states. Such a map can be constructed without prior system knowledge by identifying attractors across a sufficient number of points in the state space. However, determining the attractor for each initial state can be a laborious task. Here, we tackle the challenge of reconstructing attraction basins using as few initial points as possible. In each iteration of our approach, informative points are selected through random seeding and are driven along the current classification boundary, promoting the eventual selection of points that are both diverse and enlightening. The results across various experimental dynamical systems demonstrate that our approach requires fewer points than baseline methods while achieving comparable mapping accuracy. Additionally, the reconstructed map allows us to accurately estimate the minimum escape distance required to transition the system state to a target basin.

Keywords: complex system, attraction basin, active learning

PACS: 89.75.–k, 05.45.–a, 07.05.Mh

DOI: 10.1088/1674-1056/adbede **CSTR:** 32038.14.CPB.adbede

1. Introduction

The attraction basin is the set of initial states $\{S_0\}$ from which the system converges toward the common asymptotic state, called the attractor (see Fig. 1(a)). Many complex systems have multiple attractors, some of which are desirable while others are not. Examples include power grids (functionality/outages), biological brains (health/diseases) and wildlife populations (survival or extinction).[1]

A map of attraction basins is highly valuable, for example for (I) quantifying the system's resilience against disturbance. When disturbed, the system would return to the current attractor as long as its state remains within the basin.[2] (II) Aiding researchers in designing efficient control trajectories that drive the system from an undesired attractor across basin boundaries to a desired attractor.[3]

Our task is to accurately map attraction basins by labeling as few points as possible in the given state space, as illustrated in Fig. 1(b). We treat this task as a classification problem, where the state space is partitioned into distinct basins of attraction, each corresponding to a class. The decision boundary of the classifier is then utilized to approximate the real basin boundary. In this context, a label query means identifying the corresponding attractor that the initial state eventually reaches. Because the exact dynamics equations governing the evolution of real systems are usually unknown, in practical situations labeling usually necessitates demanding procedures.

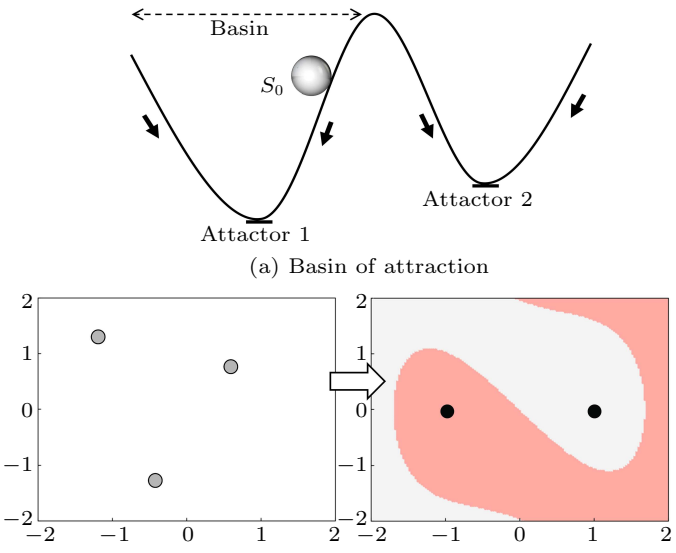


Fig. 1. (a) Graphic representation of attraction basins. The pellet (symbolizing the initial state of the system S_0) rolls towards the bottom when located within the basin (indicated by the dotted line). (b) Given an initial state space, the prior information about interior attraction basins is lacking. We map basins by selecting a few initial states (depicted as grey circles) to query attractors they converge towards. Pink and grey areas represent distinct attraction basins, respectively. Black circles represent attractors (which are fixed points in this case).

One distinction between active learning for mapping at-

†Corresponding author. E-mail: ruxiaolei@qq.com
‡Corresponding author. E-mail: gyan@tongji.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

traction basins and its other common applications, such as image classification, is the absence of an explicit sample pool (i.e., unlabeled dataset) in the former. In this context, any point within the continuous state space can be considered a potential sample, implying that the number of samples is theoretically infinite. The key challenge in active learning lies in designing a query strategy to maximize classification accuracy at a low labeling cost.[4] The query strategy is iteratively employed to augment the training data and retrain the classifier.

Due to the infinite nature of the sample pool, selecting the top-K samples after sorting the entire pool based on a specific measure becomes unfeasible.[5] Instead, we obtain samples by driving K random points towards the decision boundary, where the classifier exhibits maximum uncertainty. Additionally, to ensure diversity[6] and mitigate redundancy resulting from similar samples, we introduce a repulsive force into the crawling process. This force encourages dispersion of the obtained samples along the classification boundary.

The effectiveness of our approach is evaluated on theoretical dynamics systems, with the results indicating that, compared with baseline methods using an equal number of labeled samples, our approach achieves superior accuracy in mapping two-dimensional state spaces with intricate basin boundaries. Notably, our approach exhibits remarkable robustness when the training data is corrupted, i.e., a fraction of samples are labeled incorrectly. We also observe the map of attraction basins reconstructed using only a few samples, which allows the identification of a boundary point with an approximately minimum escape distance for transitioning the system state between different attraction basins.

2. Related works

2.1. Basins of attraction

In addition to addressing this task as a classification problem,[7] an alternative way involves reconstructing dynamics from time series. These two ways are based on different types of data. The former requires pairs of the initial state of the system and the corresponding attractor to converge, while the latter needs the state evolution sequences of the dynamical system. Reservoir computer is trained to reproduce the dynamics using the time series collected under various parameter configurations, and attraction basins of the system at the target parameter configuration are identified through dynamics simulation for a range of initial states with the reservoir.[8,9] In another study,[10] reservoir computer is also employed, but the collected time series start from randomly chosen initial states and attraction basins are inferred at the identical system. The integration of active learning is worth exploring to determine initial states for time series collection.

2.2. Active learning

Margin-based active learning relies on the distance to the classification boundary to measure the uncertainty of a sample. The efficiency of labeling the samples that are the closest to the decision boundary is demonstrated on support vector machine (SVM).[11–13] For deep learning models, GAAL[14] utilizes a generative network to synthesize samples, which is trained by minimizing the distance between the decision boundary and hidden features of synthesized samples in the penultimate layer of the classifier. ASAL[15] also employs a generative network to synthesize samples with high entropy and subsequently retrieves real samples that match with synthesized samples in the feature space obtained through PCA. Since the geometry of the decision boundary of a deep neural network is unknown, computing the distance from samples to the decision boundary is intractable. Thus, DFAL[16] computes adversarial attacks (perturbations added to samples in order to induce misclassification) with L_p norms and selects K samples with the smallest L_p norm perturbation.

3. Approach

We now introduce the proposed approach step by step.

(i) Crawl Using uncertainty measures is a fundamental strategy in active learning.[17] Information entropy is commonly used to quantify the uncertainty of the classifier's prediction on the sample, expressed as $E = -\sum_{i=1}^{C} p_i \cdot \log(p_i)$, where p_i is the probability that the sample is predicted to be of class i and C is the number of classes. The top-K samples with the maximum entropy would be selected from the unlabeled dataset to query labels. However, in our task, the state space is continuous and theoretically encompasses an infinite number of samples. One alternative scheme is to discretize the continuous space, by uniformly drawing a large number of points from the state space to form the sample pool. The denser the sampling, the better this pool represents the state space. This strategy is taken as a baseline in Section 4.

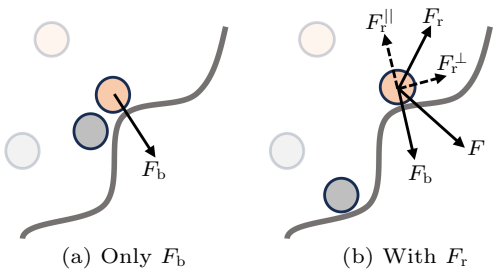


Fig. 2. Driving force for crawl. The gray curve represents the decision boundary of the classifier. The translucent balls represent the initial points of random seeding in the state space. The opaque balls in (a) and (b) represent the points obtained after a crawl with different driving forces.

In this work, we obtain high uncertainty samples through an optimization process, thereby avoiding the need to con-

struct a sample pool. Specially, we randomly seed K points in the state space and then iteratively update their positions along the directions provided by optimizing a predefined objective function.[18] An intuitive idea is to maximize the information entropy expressed as

$$\boldsymbol{X}^* = \underset{\boldsymbol{X}}{\operatorname{argmax}} \sum_{n=1}^{K} E_n.$$

However, unless they happen to fall near the classification boundary, the gradients on point set $\boldsymbol{X}$ are too small to effectively push these points. Another common measure of uncertainty is the distance L between the sample and the classification boundary.[14] The objective function then becomes

$$\boldsymbol{X}^* = \underset{\boldsymbol{X}}{\operatorname{argmin}} \sum_{n=1}^{K} L_n. \tag{1}$$

We adopt an algorithm from DeepFool proposed originally for adversarial attacks,[16] to apply a force F_{b} on each point to drive them towards the classification boundary iteratively, as illustrated in Fig. 2(a). At each iteration, the point $\boldsymbol{x}$ is orthogonally projected onto the decision boundary of the locally linearized classifier around that point.[19] Compared to constructing a sample pool, samples obtained by optimization processes are located precisely on the classification boundary. The results on an example dynamical system are shown in Fig. 3(b).

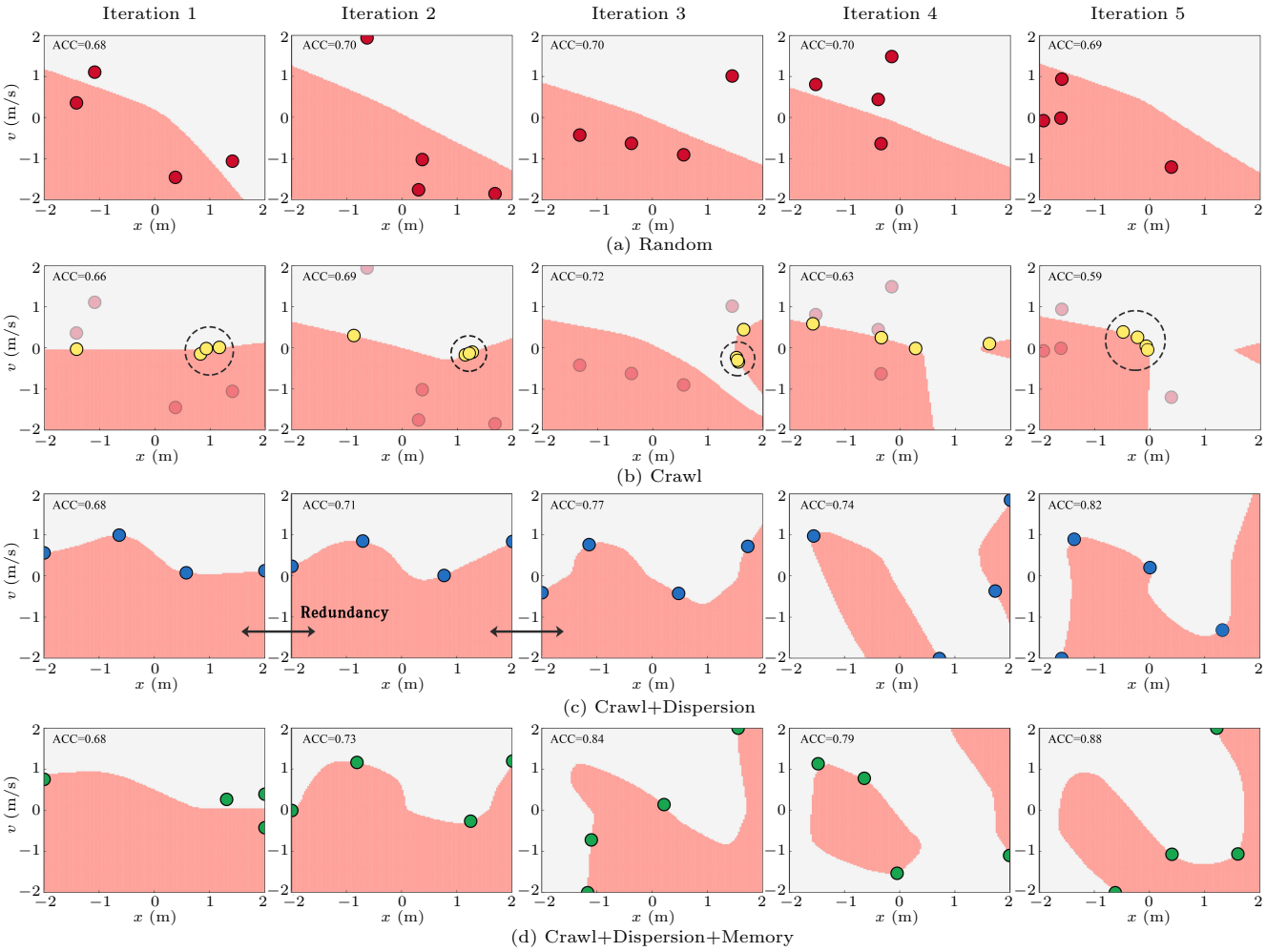


Fig. 3. Visualisation of different query strategies for the first five iterations. The more effective the query strategy, the sooner the decision boundary of the classifier approximates the actual boundary of the attraction basins shown in the right panel of Fig. 1(b). (a) Randomly seeded initial points. (b) The force F_{b} drives points to crawl toward the classification boundary, but has led to the identification of many similar points (indicated by dotted circles). The translucent points represent the initial positions. (c) The force F drives points to crawl toward the classification boundary and disperse. The schemes of sample selection in the first three iterations are similar. (d) Memory of previous samples is included when calculating the repulsive force F_{r}.

(ii) Crawl + dispersion The algorithm from DeepFool has theoretical guarantees in linear classifiers that points are orthogonally projected onto the decision boundary, but this guarantee does not hold for nonlinear classifiers. From Fig. 3(b), we observe that many points scattered in the state space eventually crawl to similar positions (as locally optimal solutions) on the classification boundary. Labeling for these samples provides redundant information again and again to eliminate the uncertainty near the same position. The lack of diversity in the batch samples reduces the efficiency of the query strategy. We exert an additional force F_{r} on each point to promote their dispersion along the whole classification bound-

ary, as illustrated in Fig. 2(b). The objective function then becomes

$$\boldsymbol{X}^* = \underset{\boldsymbol{X}}{\operatorname{argmin}} \sum_{n=1}^{K} L_n + \sum_{n=1}^{K}\sum_{m=1}^{K} \frac{1}{d_{n,m}}, \tag{2}$$

where $d_{n,m}$ is the Euclidean distance between the pair of points. The closer the points are to each other, the stronger the repulsive force F_r . However, the original force F_r may interfere with points crawling towards the boundary (especially when the angle between forces F_r and F_b is greater than $\pi/2$). We decompose the force F_r into two components

$$F_\mathrm{r} = F_\mathrm{r}^{\perp} + F_\mathrm{r}^{\parallel}, \tag{3}$$

where $F_\mathrm{r}^{\perp}$ and $F_\mathrm{r}^{\parallel}$ act perpendicularly and parallelly to the force F_b, respectively, resulting from[20,21]

$$F_\mathrm{r}^{\perp} = F_\mathrm{r} - (F_\mathrm{r}\cdot\hat{\tau})\cdot\hat{\tau}, \tag{4}$$

$$F_\mathrm{r}^{\parallel} = (F_\mathrm{r}\cdot\hat{\tau})\cdot\hat{\tau}, \tag{5}$$

where $\hat{\tau}$ is the unit direction vector of the force F_b. By discarding the parallel component, the final driving force F acting on the point is determined by

$$F = F_\mathrm{b} + \lambda F_\mathrm{r}^{\perp}, \tag{6}$$

where the hyper-parameter λ controls the step size of crawling driven by $F_\mathrm{r}^{\perp}$. Compared to the strategy of applying only force F_b, the improved results are shown in Fig. 3(c), where we see the estimated dynamical map rapidly approaching the ground truth.

(iii) Crawl + dispersion + memory (CDM) From Fig. 3(c), we observe that samples are scattered at different positions along the classification boundary, particularly at certain inflection points. However, although the initial points in two subplots have different random seeds, they converge towards similar local optimal solutions. That is, sample schemes in the first, second and third subplots are similar, meaning that labeling for selected samples in the second and third subplots does not provide substantial new information to the classifier beyond that already provided by the first subplot. The diversity in sample selection needs to be considered not only within each iteration, but also between different iterations.

We set up a dedicated memory to store historical samples selected before the current iteration. During the crawling process of a point, it receives the repulsion force not only from other points in the current batch, but also from historical points in the memory. The objective function finally becomes

$$\boldsymbol{X}^* = \underset{\boldsymbol{X}}{\operatorname{argmin}} \sum_{n=1}^{K} L_n + \sum_{n=1}^{K+M}\sum_{m=1}^{K+M} \frac{1}{d_{n,m}}, \tag{7}$$

where M is the number of samples in the memory. The further improved results are shown in Fig. 3(d).

A compact pseudo-code of the proposed approach is provided in Algorithm 1 and we offer some implementation details as follows. (1) Points may crawl out the given state space in certain dimensions (mainly due to the repulsive force F_r). In such cases, we clamp them to the edge of the state space in those dimensions. (2) If two or more points appear in an identical position, we retain one of them and place others randomly within the state space (equivalent to random initialization).

Algorithm 1. CDM

Input: Iteration number of query N_q, size of batch samples K, number of crawl N_b and N_c, classifier f_θ, memory $\mathcal{M}$
Output: Classifier f_θ with optimal parameters
Initialization: $j \leftarrow 0, g \leftarrow 0, h \leftarrow 0$

while $j < N_\mathrm{q}$
  Seed K points randomly within the state space
  while $g < N_\mathrm{b}$
    Drive points to crawl by optimizing Eq. (1)
    $g \leftarrow g+1$ ▷ *crawl to the boundary firstly*
  end while
  while $h < N_\mathrm{c}$
    Drive points to crawl by optimizing Eq. (7)
    $h \leftarrow h+1$ ▷ *disperse along the boundary*
  end while
  Put batch samples into $\mathcal{M}$ and query their labels
  Retrain f_θ using all labeled samples
  $j \leftarrow j+1$
end while

4. Experiments

4.1. Dynamical system models for tests

In theoretical models, the governing equations of system dynamics are well-defined. By substituting initial states into these equations, we can rapidly label samples by numerically simulating the dynamical processes until they converge toward the attractor. This approach allows for extensive experimentation to test and evaluate the proposed method. The equations for the dynamical systems used in the experiments are provided in Table 1. The parameters of these models are as follows: Duffing $k = 0.5$; VSC $I = 0.4, D = 0.39, \alpha = 0.7$; multistable $\alpha = -0.02$, $\varepsilon = 0.4$, $\beta = 0.15$, $k = 1$. For each of the dynamical systems, we visualize the real attractor basins in the two-dimensional state space, as shown on the top of Fig. 4. The boundaries of these basins are complex, irregular, and non-convex, which presents challenges for their mapping.

Table 1. Equations of the three dynamical systems considered.[8,10,22]

System	Equations
Duffing	$\dot{x} = v$
	$\dot{v} = -kv + x - x^3$
VSC	$\dot{\theta} = \omega$
	$\dot{\omega} = I - \sin\theta - (\alpha\cos\theta - D)\,\omega$
Multistable	$\dot{x} = y$
	$\dot{y} = z$
	$\dot{z} = -y + 3y^2 - x^2 - xz + \alpha + \varepsilon u$
	$\dot{u} = -ku - \varepsilon\,(z - \beta)$

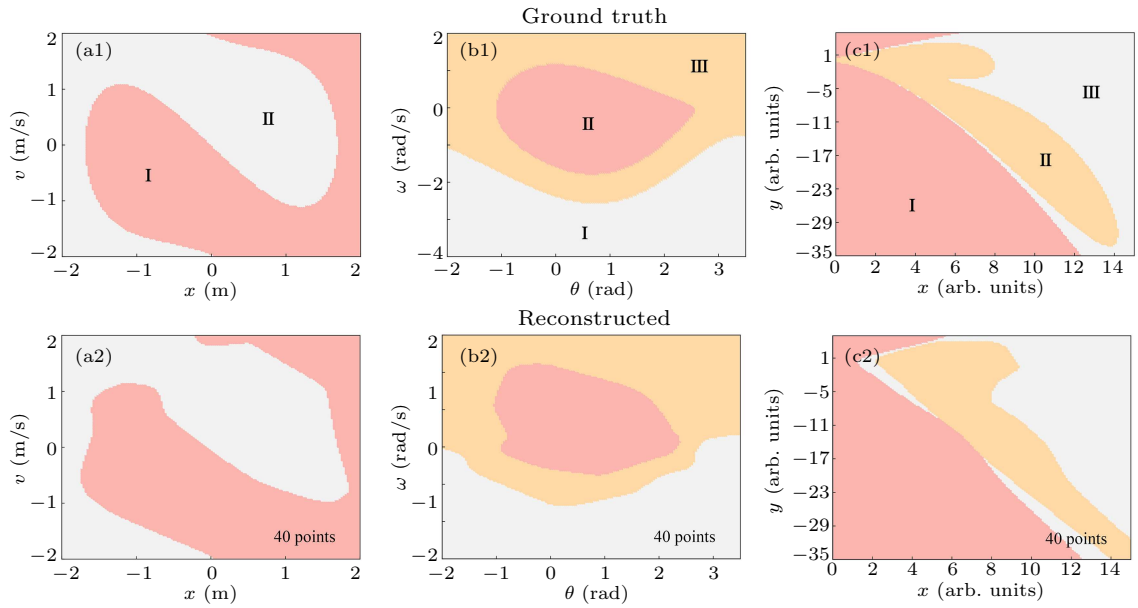


Fig. 4. Visualisation of real and reconstructed maps of attraction basins in given state spaces. The labels on horizontal and vertical coordinates of subplots indicate which two dimensions of dynamical systems constitute the initial state space. In (a1), regions I and II correspond to distinct fixed points. In (b1), regions I, II and III correspond to a positive infinite solution, a fixed point and a negative infinite solution. In (c1), regions I, II and III correspond to a positive infinite solution, an oscillation and a fixed point. In (a2), (b2), and (c2), the reconstructed maps of attraction basins are obtained by our proposed approach after ten iterations of querying.

4.2. Baselines

We choose several representative active learning methods as baselines to gauge the extent to which existing methods can solve the problem of mapping attraction basins. (1) **Random**: The naive baseline of randomly seeding K points in the state space to query. (2) **Coreset**:[23] Each point is embedded in a latent space by the activation of the network's penultimate layer, and K points are selected using a greedy furthest-first traversal conditioned on all labeled points. (3) **Conf**:[24] The top probability $\max_{i=1}^{C} p_i$ is found from the predicted class probability distribution $\{p_i\}$ for each point, and K points are selected with the smallest top probabilities. (4) **Entropy**: K points are selected with the maximum entropy of predicted class probability distribution. (5) **BADGE**:[25] Cross entropy loss is computed by the predicted class probability distribution with the hypothetical label $\hat{y} = \text{argmax}_{i=1}^{C} p_i$. Each point is embedded in a latent space using the gradient of the loss with respect to parameters of the neural network's final layer, and K points are selected by K-MEANS++. Except for Random, we firstly construct a sample pool for other baselines. The size of sample pools is 1000 in these experiments.

4.3. Evaluation measure

We use the measure of accuracy (ACC) to evaluate the performance of methods. Specially, we sample 100000 points uniformly from the given state space to make a test set, and calculate the accuracy of the trained classifier on the test set. The higher the accuracy, the closer the classification boundary approximates the boundary of attraction basins.

4.4. Training details

The classifier employed is a neural network comprising five fully-connected layers with the activation function of ReLU. The classifier is trained for 500 epochs per iteration, utilizing the Adam optimizer with a learning rate of 5.0×10^{-4}. The number of iterations N_{q} is 20 and the number of labeling samples K per iteration is 4. The step numbers of crawl N_{b} and N_{c} are 50 and 150. The step size of crawl λ is 0.1 and follows a cosine annealing schedule. For all methods, the classifier is initially trained once using K points evenly distributed along a diagonal line spanning the state space.

5. Results

5.1. Performance

The results of the performance comparison are presented in Table 2. Our approach achieves an accuracy exceeding 0.9 on all three dynamical systems with only 40 labeled samples. Notably, our algorithm demonstrates superior efficiency compared to baselines by achieving comparable mapping accuracy while reducing the number of samples, thereby resulting in cost savings for labeling. Compared with ground truths (top of Fig. 4), the overall pictures of attraction basins are well reconstructed (bottom of Fig. 4), although the classification boundaries usually have more twists and turns than the actual attraction basin boundaries do.

5.2. Robustness against mislabeling

We examine the robustness of the proposed approach in a scenario where samples have a certain probability of being mislabeled (i.e., randomly assigned labels from other classes).

Results are presented in Table 3. In this scenario, the mislabeled samples would also be stored in the memory $\mathcal{M}$. According to Eq. (7), these mislabeled points repel newly selected points close to them, potentially making the corresponding regions hard to be rechecked and corrected.

With this concern, we design a variant called CDM_once, where only the samples of the previous query iteration are stored in $\mathcal{M}$. From Table 3, we observe that the performance of CDM is superior to that of CDM_once in the Duffing system, and is similar in the other two systems. Notably, in comparison to the baselines whose accuracy rapidly decreases with mislabeling, our approach is quite robust against corrupted data.

Table 2. Performance comparison of CDM against baselines in different dynamical systems. The second column of table indicates the total number of labeled samples currently. Standard deviations of ACC are computed in ten experiments with random seeds..

Dynamical system	Points	Method					
		Random	Coreset	Conf	Entropy	BADGE	CDM
Duffing	20	0.79±0.07	0.78±0.03	0.80±0.06	0.79±0.06	0.79±0.03	**0.82**±0.05
	40	0.87±0.04	0.85±0.04	0.87±0.04	0.88±0.05	0.87±0.04	**0.92**±0.02
	60	0.90±0.03	0.89±0.02	0.93±0.03	0.94±0.02	0.94±0.01	**0.95**±0.01
	80	0.93±0.02	0.92±0.03	**0.96**±0.01	**0.96**±0.02	0.95±0.01	**0.96**±0.01
VSC	20	0.82±0.05	0.65±0.08	0.68±0.11	0.85±0.07	0.85±0.04	**0.88**±0.03
	40	0.88±0.02	0.76±0.05	0.80±0.10	0.93±0.02	0.92±0.02	**0.94**±0.01
	60	0.92±0.01	0.84±0.05	0.87±0.06	0.94±0.02	0.94±0.01	**0.96**±0.01
	80	0.93±0.01	0.86±0.05	0.89±0.04	**0.97**±0.01	0.96±0.00	**0.97**±0.01
Multistable	20	0.81±0.03	0.80±0.05	0.76±0.05	0.82±0.04	0.81±0.06	**0.84**±0.02
	40	0.86±0.02	0.84±0.05	0.80±0.03	0.85±0.04	0.88±0.02	**0.90**±0.02
	60	0.87±0.05	0.86±0.02	0.83±0.03	0.88±0.03	0.91±0.02	**0.92**±0.01
	80	0.90±0.02	0.87±0.02	0.87±0.04	0.89±0.03	**0.92**±0.02	**0.92**±0.01

Table 3. Robustness against mislabeling in different dynamical systems. The probability that the sample is mislabeled increases gradually from 10% to 20%. Standard deviations of ACC are computed in ten experiments with random seeds. The number of labeled samples is 80, obtained by 20 iterations of querying.

Dynamical system	Mislabeled ratio	Method						
		Random	Coreset	Conf	Entropy	BADGE	CDM	CDM_Once
Duffing	10%	0.81±0.05	0.81±0.03	0.87±0.05	0.88±0.03	0.86±0.04	**0.90**±0.03	0.86±0.04
	15%	0.79±0.04	0.81±0.03	0.84±0.06	0.85±0.05	0.82±0.05	**0.88**±0.03	0.83±0.04
	20%	0.75±0.06	0.72±0.05	0.78±0.08	0.80±0.07	0.79±0.03	**0.82**±0.04	0.79±0.06
VSC	10%	0.85±0.03	0.80±0.06	0.79±0.08	0.84±0.06	0.81±0.04	**0.92**±0.03	**0.92**±0.04
	15%	0.84±0.04	0.71±0.13	0.77±0.07	0.75±0.12	0.76±0.07	0.89±0.04	**0.90**±0.03
	20%	0.76±0.07	0.65±0.08	0.74±0.07	0.72±0.10	0.73±0.06	0.82±0.10	**0.86**±0.08
Multistable	10%	0.83±0.03	0.82±0.05	0.80±0.04	0.81±0.06	0.86±0.03	0.87±0.02	**0.89**±0.03
	15%	0.80±0.05	0.77±0.06	0.77±0.08	0.76±0.06	0.81±0.05	**0.87**±0.02	0.86±0.03
	20%	0.76±0.07	0.72±0.06	0.76±0.08	0.78±0.06	0.77±0.08	**0.83**±0.04	**0.83**±0.05

5.3. Minimum escape distance

In a dynamical system with multiple attractors, such as Duffing, certain attractors correspond to functional states S_{f} while others may lead to dysfunctional system states S_{d}. When the system currently resides in S_{d}, the restorer aims to revive the system to S_{f} (conversely, the attacker disrupts the system from S_{f} to S_{d}).[26,27] Consequently, a crucial question arises regarding the effort required for such transitions.[28,29] The reconstructed map enables us to identify the minimum distances for transitions between different basins, providing essential insights into this question.

Given the characteristics of the attraction basin, once the system state enters within a basin, it spontaneously converges towards the corresponding attractor. Therefore, driving the system state towards the boundary of the target attractor's basin is sufficient to transition from the current attractor to the target one. The point on the boundary that is closest to the current attractor is referred to as the minimum escape point (MEP). From the actual map of attraction basins, the MEP can be readily identified, as illustrated in Fig. 5(b). In the absence of any map information, driving the system state towards the target attractor will bring it to the boundary but not the MEP. The trajectory connecting the current and target attractors inevitably intersects with the boundary of the attraction basin of the latter, as depicted in Fig. 5(a). Therefore, with the information from the real map, the distance $d_{\triangle} - d_{\star}$ for transitioning between system states is shortened.

Subsequently, we investigate how much distance can be shortened based on reconstructed maps, as illustrated in Fig. 5(c). This reduction can be quantified by the expression

$$\frac{d_{\triangle} - d_{\triangledown}}{d_{\triangle} - d_{\star}}. \tag{8}$$

As shown in Fig. 7(a), for the Duffing system, the map reconstructed using only 16 labeled samples achieves a distance shortening of about 63% compared to the real map. Furthermore, the difference between $d_{\triangledown}$ and $d_{\star}$ diminishes as the number of labeled samples increases.

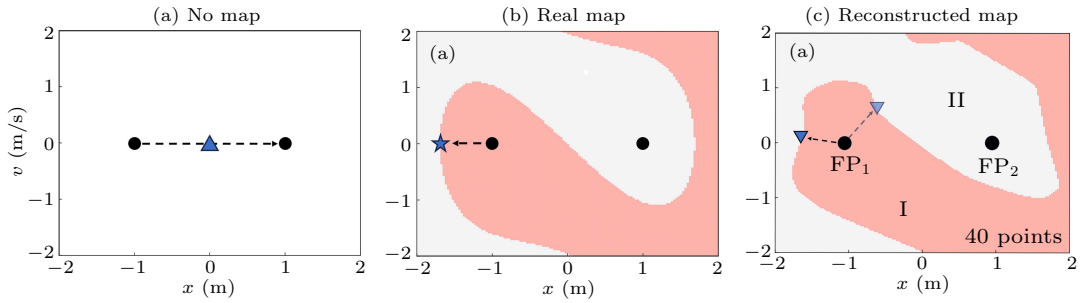


Fig. 5. Escape points for transitioning from current and target attractors (region I → II) based on different maps in the system of Duffing. (a) Symbol △ denotes the intersection of the boundary and connecting lines between attractors. (b) Symbol ⋆ denotes the minimum escape point. (c) Symbol ▽ denotes the approximate minimum escape point on the map reconstructed using 40 labeled samples. Transparent ▽ denotes a potential point, of which required escape energy is slightly smaller than ▽. All boundary points that do not really converge to the target attractor have been excluded.

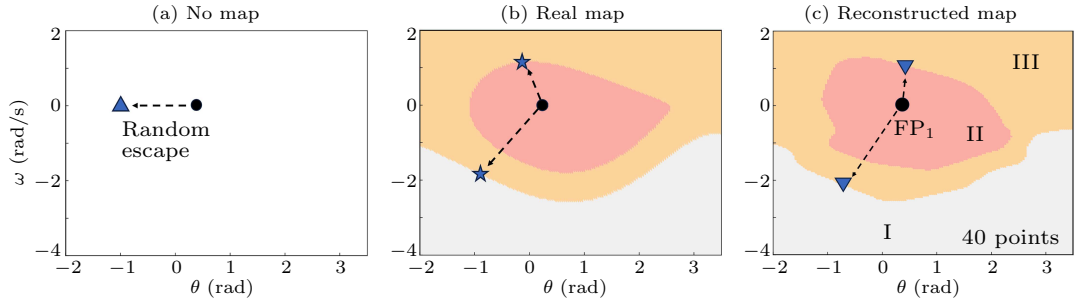


Fig. 6. As Fig. 5. but in the VSC system. (a) Symbol △ denotes a point on the boundary along the random direction. (b) Top ⋆ denotes the MEP for region II → III, and bottom ⋆ denotes the MEP for region II → I.

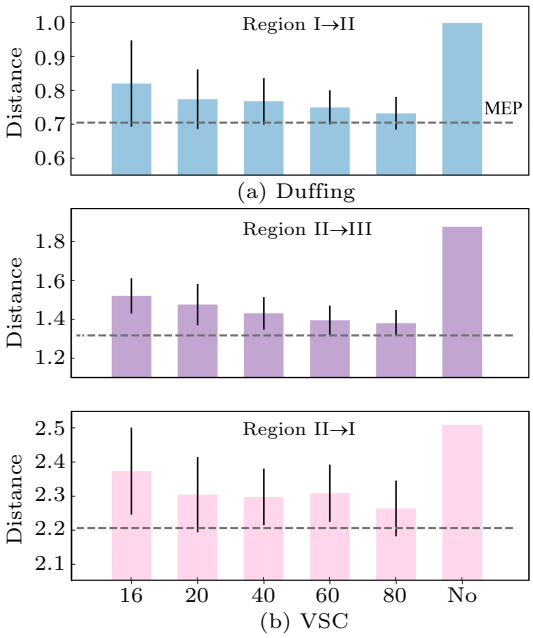


Fig. 7. Escape distance based on different maps. The horizontal dashed line indicates the minimum escape distance. Three subplots share the same abscissa, which indicates the number of labeled samples used for mapping attraction basins. The term 'No' implies the absence of any information about maps. Black short lines at top of bars represent standard deviations computed in twenty experiments with random seeds.

The VSC system evolves towards positive infinity in region I and negative infinity in region III. These two regions do not have attractors coinciding with fixed points, which makes our previous strategy ambiguous for determining control energy in the absence of map information. In this case, rather than driving the system towards a target attractor, we can only drive the system in a random direction to escape the current attraction basin, as illustrated in Fig. 6(a). The random escape distance is calculated as the average over the whole boundary. The results are provided in Fig. 7(b). The map reconstructed by only 16 labeled samples achieves the distance shortening of about 64% and 45% of those for the real map for region II → III and region II → I, respectively. Overall, these observations underscore the practical significance of reconstructed maps.

5.4. Impact of batch samples size

We investigate the impact of varying the number K of selected samples per query on accuracy. A smaller K necessitates more query iterations to acquire an equivalent number of labeled samples. As shown in Table 4, the accuracy is influenced by the variation of K when the number of labeled samples is small, while it tends to stabilize as the number of labeled samples increases.

Table 4. The impact of batch sample size in the Duffing system. The first column represents the number of currently labeled samples, and the first row represents the number of selected samples in each query. Standard deviations of ACC are computed in ten experiments with random seeds.

	1	2	4	8
8	0.71±0.05	**0.76**±0.04	0.71±0.04	0.67±0.03
16	0.79±0.03	**0.80**±0.04	**0.80**±0.06	0.77±0.05
32	0.89±0.01	**0.90**±0.03	0.89±0.03	0.86±0.03
48	0.92±0.02	0.93±0.01	**0.94**±0.01	0.92±0.01
64	**0.95**±0.01	**0.95**±0.01	**0.95**±0.01	**0.95**±0.01
80	**0.96**±0.00	**0.96**±0.01	**0.96**±0.01	**0.96**±0.01

6. Discussion

Our work aims to detect attraction basins with an efficient sampling strategy. The distinction between traditional classification and ours is that the decision boundary in the former is not unique as long as it can efficiently classify the dataset, while our goal is to approximate the real boundary of attraction basins. We systematically investigate the active learning methodology for solving this problem, and introduce an interesting task for the active learning community.

Our method has some limitations in the field of dynamical systems research. We have only verified it in basic dynamical systems with stable attractors, which means it may not be applicable to more complex systems, such as those involving cycles or chaotic behavior. Our approach could be significantly improved by extending its application to real-world complex systems, especially those involving multiple variables, nonlinear dynamics, and multiscale interactions. Additionally, our artificial neural network is relatively conventional, and there is room for enhancement. In the future, we aim to explore the use of more advanced artificial intelligence models, such as network models based on transformers and graph neural networks, to better understand and precisely explore the boundaries of attractors in dynamical systems.

A more challenging future direction is to map attraction basins in high-dimensional state space, where the boundary is a $(N_{\mathrm{d}}-1)$-dimensional manifold and N_{d} is the number of dimensions. The continuous two-dimensional state space can be represented well by a large number of uniformly sampled discrete points. However, as dimensionality increases, there is an exponential growth in the requisite number, following a relationship of $q^{N_{\mathrm{d}}}$, where q is the number of sampled points required along one dimension. Consequently, pool-based active learning that necessitates constructing a sample pool possesses inherent limitations, while obtaining informative samples through an optimization process holds promise.

Program availability

The source code is openly available in Science Data Bank at https://www.doi.org/10.57760/sciencedb.j00113.00201.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. T2225022, 12350710786, 62088101, and 12161141016), Shuguang Program of Shanghai Education Development Foundation (Grant No. 22SG21), Shanghai Municipal Education Commission, and the Fundamental Research Funds for the Central Universities. We thank Dr. Xin-Ya Zhang and Jia-Jie Qin for helpful discussion.

References

[1] Scheffer M, Bascompte J, Brock W A, Brovkin V, Carpenter S R, Dakos V, Held H, Nes E V, Rietkerk M and Sugihara G 2009 *Nature* **461** 53
[2] Menck P J, Heitzig J, Marwan N and Kurths J 2013 *Nat. Phys.* **9** 89
[3] Cornelius S P, Kath W L and Motter A E 2013 *Nat. Commun.* **4** 1942
[4] Zhang B C, Li L, Yang S J, Wang S H, Zha Z J and Huang Q M 2020 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 13-19, 2020, Seattle, USA, p. 8756
[5] Ren P Z, Xiao Y, Chang X J, Huang P Y, Li Z H, Gupta B B, Chen X J and Wang X 2021 *ACM Comput. Surv.* **9** 180
[6] Kirsch A, Amersfoort J V, Gal Y 2019 33*rd International Conference on Neural Information Processing Systems*, December 8–14, 2019, Vancouver, Canada, p. 7026
[7] Che Y M and Cheng C Q 2021 *Chaos* **31** 053129
[8] Roy M, Mandal S, Hens C, Prasad A, Kuznetsov N V and Shrimali M D 2022 *Chaos* **32** 101104
[9] Gauthier D J, Fischer I and Röhm A 2022 *Chaos* **32** 113107
[10] Du Y, Li Q, Fan H, Zhan M, Xiao J H and Wang X G 2024 *Phys. Rev. Res.* **6** 013181
[11] Campbell C, Cristianini N and Smola A 2000 *Proceedings of the* 17*th International Conference on Machine Learning*,June 29–July 2, 2000, Stanford, USA, p. 111
[12] Tong S and Koller D 2001 *J. Mach. Learn. Res.* **2** 45
[13] Brinker K 2003 *Proceedings of the* 20*th International Conference on Machine Learning*, August 21-24, 2003, Washington, USA, p. 59
[14] Zhu J J and Bento J 2017 arXiv:1702.07956 [cs-LG]
[15] Mayer C and Timofte R 2020 *IEEE Winter Conference on Applications of Computer Vision*, March 1–5, 2020, Snowmass, USA, p. 3071
[16] Ducoffe M and Precioso F 2018 arXiv:1802.09841 [cs-LG]
[17] Munjal P, Hayat N, Hayat M, Sourati J and Khan S 2022 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 21–24, 2022, New Orleans, USA, p. 223
[18] Zhu L, Liu Z and Han S 2019 33*rd International Conference on Neural Information Processing Systems*, December 8-14, 2019, Vancouver, Canada, p. 14774
[19] Moosavi-Dezfooli S M, Fawzi A and Frossard P 2016 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 27-30, 2016, Las Vegas, USA, p. 2574
[20] Barkema G T and Mousseau N 1996 *Phys. Rev. Lett.* **77** 4358
[21] Draxler F, Veschgini K, Salmhofer M and Hamprecht F 2018 *Proceedings of the* 35*th International Conference on Machine Learning*, July 10–15, 2018, Stockholmsmässan, Sweden, p. 1309
[22] Daza A, Wagemakers A, Georgeot B, Guéry-Odelin D and Sanjuán M A F 2016 *Sci. Rep.* **6** 31416
[23] Sener O and Savarese S 2018 *6th International Conference on Learning Representations*, April 30–May 3, 2018, Vancouver, Canda, p. 1
[24] Wang D and Shang Y 2014 *International Joint Conference on Neural Networks*, July 6–11, 2014, Beijing, China, p. 112
[25] Ash J T, Zhang C C, Krishnamurthy A, Langford J and Agarwal A 2020 8*th International Conference on Learning Representations*, April 26–May 1, 2020, Virtual Only Conference, p. 1
[26] Sanhedrai H, Gao J X, Bashan A, Schwartz M, Havlin S and Barzel B 2022 *Nat. Phys.* **18** 338
[27] Zhang X, Zhang X, Yan G and Moore J M 2023 *Phys. Rev. Res.* **5** 033138
[28] Yan G, Ren J, Lai Y C, Lai C H and Li B W 2012 *Phys. Rev. Lett.* **108** 218703
[29] Klinshov V V, Nekorkin V I and Kurths J 2015 *New J. Phys.* **18** 013004