Characterizing Nonlinear Dynamics via Smooth Prototype Equivalences

Roy Friedman [1,*] Noa Moriel [1] Matthew Ricci [2,3] Guy Pelc [1] Yair Weiss [1] and Mor Nitzan [1,4,5,†]

[1] School of Computer Science and Engineering, Hebrew University of Jerusalem, Israel
[2] Université Paris Cité, Paris, France
[3] Institut Imagine, INSERM UMR 1163, Paris, France
[4] Racah Institute of Physics, Hebrew University of Jerusalem, Israel
[5] Faculty of Medicine, Hebrew University of Jerusalem, Israel

(Received 8 March 2026; accepted 10 September 2026; published 26 October 2026)

Characterizing the long-term behavior of dynamical systems given limited measurements is a common challenge throughout the physical and biological sciences. This is a challenging task due to the sparsity and noise inherent to empirical observations, as well as the variability of possible long-term dynamics. We address this by introducing smooth prototype equivalences (SPE), a framework for matching sparse observations to prototypical behaviors using invertible neural networks that model smooth phase-space deformations. SPE can localize the invariant sets describing long-term behavior of the observed dynamics through the learned mapping from prototype space to data space. Furthermore, SPE can classify dynamical regimes by comparing the data residual of the deformed measurements to prototype dynamics. Our method outperforms existing techniques in the classification of oscillatory systems and can efficiently identify invariant structures like limit cycles and fixed points in an equation-free manner, even when only a small, noisy subset of the phase space is observed. SPE further reveals driving genes in synthetic oscillators such as the repressilator regulatory circuit, and traces cyclic biological processes like the cell cycle trajectory directly from experimental high-dimensional single-cell gene expression data. Finally, the flexibility of the prototype-matching approach of SPE makes it possible to interrogate a diverse set of biologically relevant behaviors, enabling the localization of a bifurcation between epiblast and primitive endoderm fates in the early mouse embryo, and identification of differentiation endpoints, along with candidate marker genes, from hematopoietic lineage-tracing data.

DOI: 10.1103/yj4q-8n54

I. INTRODUCTION

In stable nonlinear systems, such as those exhibited in some contexts of cellular biology, orbits converge to sets of the phase space which are invariant to the dynamics. The characterization of these invariant sets is of high interest in the context of single-cell biology, where invariant sets in gene expression space correspond to reproducible and stable biological processes, or programs carried out by individual cells. For instance, multiple cellular processes approximately correspond to limit cycles in gene expression space, including the cell cycle, where proliferating cells traverse different phases of this cycle at recurring intervals [1–4], and the circadian rhythm, where the expression levels of a set of genes oscillate with a diurnal period [3,5]. Conversely, the transition of cells from progenitors to terminally differentiated cells of distinct fates can be modeled as the convergence of cells to multiple node attractors in gene expression space [6–9]. In this way, the exact form of invariant sets encoded by cells in gene expression space can reveal the set of dynamical processes cells go through in different biological contexts, including homeostasis and development, and uncover the genetic programs and interactions that drive these dynamics. Moreover, the forms of deviations from these stable patterns can potentially inform mechanistic interpretations in the context of aging or disease progression. Thus, distinguishing between different possible invariant sets, pinpointing the underlying driving genes, and determining the timings of these processes is the subject of intense work in computational biology (i.e., [10–16]).

A natural contender for the analysis of such dynamical information is single-cell RNA-sequencing (scRNA-seq), which measures high-dimensional gene expression in thousands of cells simultaneously at a single time-point [17,18]. Recent extensions to this technique measure gene expression information through time using new experimental designs or inference of kinematic variables directly from the static snapshots. These methods include multitime-point sequencing within a single biological process [9,19], lineage tracing methods to experimentally access rates of change of cell types [20,21], metabolically labeling transcripts before sequencing to directly observe rates of change in RNA counts [22,23], estimating instantaneous changes of counts through RNA velocity [24,25], and even live tracking of RNA levels in

*Contact author: roy.friedman@mail.huji.ac.il
†Contact author: mor.nitzan@mail.huji.ac.il

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

live cells [26]. This wide range of methodologies converts static molecular data into dynamical information and can be used to support fine-grained physical modeling and long-term forecasting. Nevertheless, constructing these robust models from sparse, noisy, and high-dimensional dynamical expression data remains a challenging problem.

In the broader context of data-driven analysis of nonlinear dynamics, substantial work has been devoted to the estimation of vector fields underlying observed time series data in the form of ordinary differential equations (ODEs) [8,27–30]. However, even when the governing equations are known and the dynamics are stable, the exact structure of long-term equilibrium or oscillatory behaviors is challenging to quantify [31,32], and this challenge is exacerbated in realistic scenarios given sparse and noisy data, when the governing equations of the dynamics are often unknown *a priori*. Moreover, invariant sets can exhibit complex geometry in phase space and can change drastically when introduced to small perturbations [33]. These attributes, particularly in high-dimensional systems, frequently hinder the detection and localization of such invariant sets [34]. A different approach, taken by a recent line of work, has sought to determine whether two systems are governed by qualitatively similar dynamics [34–40] through the use of relaxed notions of dynamical equivalences [41,42]. These characterizations are attractive because they focus on the qualitative behaviors of dynamical systems, such as the stability and type of invariant sets. To do so, these approaches compare two systems, using observations from long time series in order to form a delay embedding where Takens' embedding theorem can be employed [43] or to compare spectra of linear approximations to the dynamics in the context of Koopman theory [44]. It is therefore difficult to apply the above methodologies in practice to many types of experimental data, such as single-cell gene expression data, where cells are destroyed as part of the measurement and cannot be followed in time. In this work, we offer an alternative view, wherein noisy empirical data are matched to an analytical model guided by domain expertise, regardless of the sparsity, temporal resolution, or dimensionality of the observations.

Our work focuses on the identification and tracing of invariant sets in sparse, high-dimensional observations measured when the underlying dynamical system is unknown. To extract information about the long-term behavior of the underlying dynamics, we rely on the fact that the topology and stability of the invariant sets of a dynamical system are not affected by a smooth, invertible change of coordinates. Two systems related by such a smooth invertible mapping are called smoothly orbitally equivalent, and their invariant sets share the same stability and topology [35]. Our approach, called smooth prototype equivalences (SPE), learns an invertible mapping, parametrized by an invertible neural network, between the data and a simple, prototypical dynamical system. Such a prototypical system could be, for example, a set of governing equations that exhibit a bifurcation between a limit cycle and a node attractor. This framework naturally lends itself to modeling and localizing invariant sets exhibited in the data. However, the fitting procedure also captures transient trajectories toward these invariant sets by matching orbits in data space to their respective location in prototype space.

We first demonstrate how SPE can classify and localize node and limit cycle attractors in a synthetic dataset of two-dimensional (2D) dynamical systems drawn from models across the physical and life sciences. Furthermore, we show that SPE, unlike previous methods, is robust against noisy and low-sample conditions, illustrating the compatibility of SPE with real data-driven scenarios. We then demonstrate that SPE can naturally extend to higher dimensions by applying it to the synthetic six-dimensional repressilator gene regulatory network [45].

Beyond synthetic data, we illustrate how SPE can be used in real scientific scenarios by deploying our method to localize cell cycle dynamics from noisy, high-dimensional single-cell gene expression measurements, without relying on a known reference atlas of genes expressed in the cell cycle. We continue by modeling an early bifurcation in embryo development directly from scRNA-seq data. Finally, we demonstrate how SPE can be used to localize differentiation fates by applying it to lineage tracing data.

II. METHOD

A. Smooth prototype equivalences

In single-cell RNA sequencing (scRNA-seq) data, individual cells are sequenced and represented by sparse high-dimensional gene expression data. Recent advances in computational biology enable the estimation of local change rates of gene expression in scRNA-seq data, either through experimental design [9,19,20] or by estimating the rates of change of RNA counts, as in RNA velocity [23–25]. Such temporal information can be modeled as arising from a hidden set of ordinary differential equations (ODEs) governing the dynamics of the whole biological system. In this work, we assume access to such data, corresponding to sparse observations from a subset of the phase space of an ODE. Mathematically, let $\mathcal{D} = \{(x_i, \dot{x}_i)\}_{i=1}^{N}$ be a dataset of N pairs of positions $x_i \in \mathbb{R}^d$ and their respective velocities $\dot{x}_i \in \mathbb{R}^d$, which were generated by a hidden ground-truth structurally stable ODE, $\dot{x} = f(x)$. Realistically, observed data represent noisy estimates of the velocities, $\dot{x}_i = f(x_i) + \epsilon_i$, where ϵ_i is a per-sample noise term.

To effectively restrict the search space when modeling the dynamics of the observed data, we use strong prior information for the possible dynamical patterns within the data. In other words, we can constrain our model to only match dynamical systems similar to a candidate set of governing equations, $\dot{y} = g(y)$. In this scenario, the dynamics of $g(y)$ form a prototype for the behavior expected to exist in the observed data.

To achieve this, we utilize the notion of smooth orbital equivalences between dynamical systems [35]. Specifically, two systems $\dot{x} = f(x)$ and $\dot{y} = g(y)$ are said to be smoothly orbitally equivalent if

$$\exists\, H \text{ s.t. } \mu(x)\partial_x H(x)\, \dot{x} = g(H(x)), \tag{1}$$

where $\mu(x) \in \mathbb{R}_+$ is a positive scalar function, $H : \mathcal{X} \to \mathcal{Y}$ is a diffeomorphism, and $\partial_x H(x)$ is the abbreviated notation for the Jacobian of H with respect to x. If two systems obey this equivalence relation, each orbit of the system $f(x)$ has a smooth one-to-one mapping to an orbit in $g(y)$, and thus the

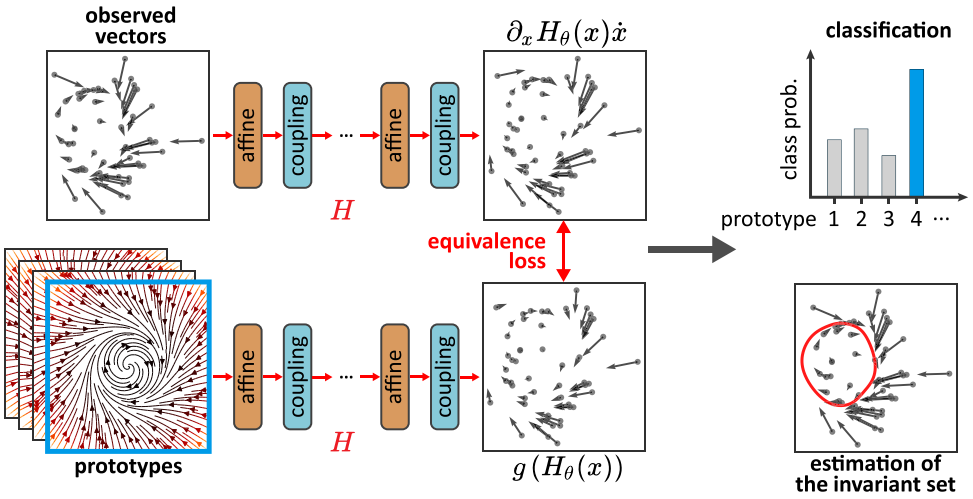


FIG. 1. Fitting dynamical prototypes using smooth prototype equivalences (SPE). The observed dynamics (positions x_i and velocities $\dot{x}_i$) are compared with each of the chosen prototypes $g(y)$, which involves training an invertible neural network (INN) $H_\theta(x)$ for each prototype $g(y)$. After training, the observed dynamics are classified as the prototype that achieves the smallest equivalence loss [Eq. (2)]. A trained INN $H_\theta(x)$ provides a mapping between data space and prototype space, which in turn can be used to estimate the long-term behavior, i.e., invariant set, of the dynamics underlying the observed data.

two systems share the same qualitative behavior, such as the topologies and stabilities of their invariant sets, and the shapes of trajectories which approach these invariant sets.

In this work, we aim to learn a diffeomorphism between the data, $\dot{x} = f(x)$, and the prototype, $\dot{y} = g(y)$, and we parametrize such a diffeomorphism using an invertible neural network (INN) $H_\theta(x)$, which is optimized in order to approximately satisfy Eq. (1). This is achieved by minimizing the following equivalence loss [35]:

$$L_{\mathrm{E}}(H_\theta, g) = \frac{1}{N}\sum_{i=1}^{N}\left\| \frac{\partial_{x_i} H_\theta(x_i)\dot{x}_i}{\|\partial_{x_i} H_\theta(x_i)\dot{x}_i\|} - \frac{g(H_\theta(x_i))}{\|g(H_\theta(x_i))\|}\right\|^2. \quad (2)$$

When the equivalence loss is zero, the dynamics that are generated by $\dot{x}$ and $g(y)$ are smoothly orbitally equivalent for any point $x \in \mathcal{D}$ (see Fig. 1).

In practical scenarios given sparse and noisy observations, the loss in Eq. (2) will not necessarily be equal to zero, even when the two underlying systems are equivalent. Nonetheless, the equivalence loss provides a good indication of similarity between dynamical systems when there is access to a large amount of samples [35]. To improve convergence, we optimize the diffeomorphism using a regularized version of Eq. (2), penalizing transformations with a large determinant and those that push the prototype's invariant set far from the data (see Appendix A 6 for full details).

B. Invariant set localization and prototype classification with SPE

Our explicit construction using a mapping between the prototype and the observed data allows us to effectively localize structures in the observed data, which are in general very difficult to find [32]. For example, suppose that our prototype $g(y)$ has a limit cycle, and we can generate points on this limit cycle $\gamma = \{y_1, \ldots, y_M\}$. Given the learned explicit mapping between the two spaces H_θ, $H_\theta^{-1}(\gamma) = \{H_\theta^{-1}(y_1), \ldots, H_\theta^{-1}(y_M)\}$ is guaranteed to be a closed loop in data space. Moreover, if the equivalence loss between the observed data and the prototype is low, then $H_\theta^{-1}(\gamma)$ will be a set of points close to the limit cycle of the true hidden system, $\dot{x} = f(x)$ (see Fig. 1).

Moreover, SPE allows for classification between a set of prototypes, as the equivalence loss directly corresponds to a measure of dissimilarity. So, given a dictionary of possible prototypes $\{g_k(y)\}_{k=1}^K$, we optimize a diffeomorphism $H_k(x)$ for each, using the equivalence loss from Eq. (2). We then calculate the equivalence loss, $\mathrm{L_E}(H_k, g_k)$, between the observed data and each of the prototypes. The data are then classified according to the prototype that has the smallest equivalence loss.

C. Modeling diffeomorphisms with invertible neural networks

The equivalence loss from Eq. (2) acts as a training objective for fitting the diffeomorphism between data and prototype space, H_θ. To properly capture the mapping between data and prototype space, H_θ has to be expressive enough to capture the correct mapping while also being computationally lightweight. Standard neural networks (NNs) are apt choices as tunable parametric functions for such tasks, but are not guaranteed to be invertible, which is one of our core requirements from H_θ. Instead, we opt to use invertible neural networks (INNs), a specific type of NNs which are adapted from normalizing flows (NFs [46]), to model our parametric diffeomorphisms. In INNs, each layer is invertible by design, which makes inverting the whole mapping a simple algorithmic procedure. Furthermore, INNs enable efficient and explicit computations of both the log-determinants of the Jacobian as well as the Jacobian-vector products (JVPs), $\partial_x H_\theta(x)\dot{x}$, needed to calculate the equivalence loss. As such, INNs are particularly well suited as the basis of our parametric diffeomorphisms.

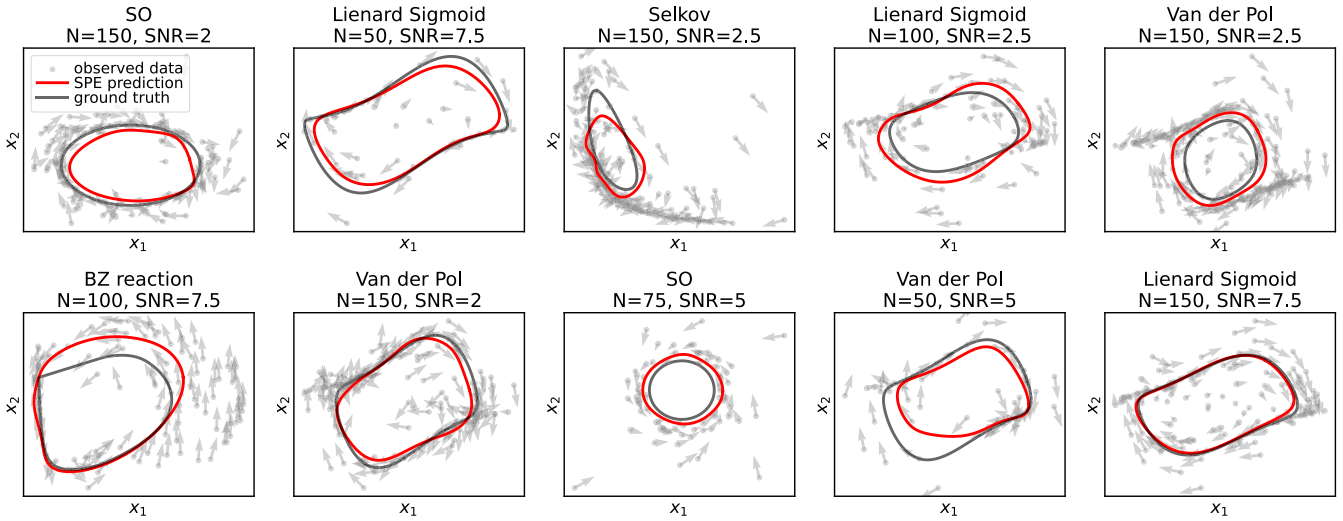


FIG. 2. Predicting limit cycles for various 2D dynamical systems using SPE. Examples of invariant sets predicted by SPE (red) from observed vectors (gray), for different dynamical systems which exhibit a limit cycle attractor. Ground-truth trajectories were simulated for each hidden system (black) for visual comparison. Dynamical systems: simple oscillator (SO), BZ reaction, Sel'kov, Liénard sigmoid, Liénard polynomial, and Van der Pol; corresponding order parameters were uniformly chosen from the set of ranges that exhibit a limit cycle (Appendix B 1).

Our INNs utilize a small number of blocks, composed of alternating invertible linear transformations and affine coupling layers [47] (see Appendix A for more information). Both of these layer types are adapted so that their JVPs can be calculated in closed form. Specifically, instead of using multilayer perceptrons (MLPs) typically found in affine coupling layers [46], we use cosine features, which we term Fourier feature coupling. These Fourier feature coupling layers are highly expressive even when the network is narrow and shallow, and their JVPs can be calculated efficiently and in closed form. For more details regarding network design choices, see Appendix A.

III. RESULTS

A. Reconstructing attractors from sparse data

Locating attractors given realistic observations is generally a challenging problem since sparsity and noise make the underlying, unknown dynamical equations strongly undetermined. We demonstrate how SPE circumvents this difficulty on data simulated from a set of standard families of two-dimensional dynamical systems exhibiting limit cycles. The governing equations for these families of systems can be found in Appendix B 1.

To that end, we optimize a transformation between observed systems and a prototype chosen to flexibly model point and oscillatory dynamics. We use the simple oscillator system, described by a set of a coupled pair of ODEs, defined in polar coordinates by

$$\dot{r} = r(a - r^2), \quad \dot{\theta} = \omega. \tag{3}$$

Here, a is an order parameter that controls whether the system has a node attractor (negative values) or a limit cycle (positive values), and ω determines the orientation of the cycle (clockwise when negative and counterclockwise when positive). Data were acquired by sampling initial conditions and points along simulated trajectories (details in Appendix B 2).

After fitting an INN $H_\theta(x)$ between the observed data $\mathcal{D}$ and our limit cycle prototype, we estimated the location of the limit cycle in data space by uniformly sampling a sequence of points, $T_\mathcal{Y} = \{y_j\}_{j=1}^M$, which lie along the limit cycle in prototype space, $\mathcal{Y}$. Each point was then mapped back into data space to define a trajectory along the estimated invariant set of the observed dynamics: $\hat{T}_\mathcal{X} = H_\theta^{-1}(T_\mathcal{Y}) = \{H_\theta^{-1}(y_j)\}_{j=1}^M$. SPE can accurately trace the structure of limit cycles across diverse dynamical systems given sparse and noisy observations (Fig. 2 and in Appendix C).

To quantitatively evaluate the performance of SPE, we calculate the Wasserstein (W_2) distance between samples from the invariant set of the ground-truth system and those from the invariant set of the fitted model (details in Appendix C 2). When the positions x_i are sparse, or the vectors are noisy, SPE more accurately reconstructs the underlying invariant sets than existing baselines, including a k-nearest neighbor (kNN) interpolator, SINDy [28], SINDy-AutoEncoders (SINDy-AE, [48]) and a multilayer perceptron (MLP) (Fig. 3 left; baseline details are in Appendix C 1). In particular, when the sparsity level or noise level is high, SPE outperforms existing baselines more substantially (Fig. 3, left) because it is constrained to model only stable dynamics and can therefore more closely reconstruct the underlying vector field.

SPE can avoid overfitting by detecting mismatches between input prototypes and the data through sensitivity analysis, where small changes in the data result in substantially different fitted diffeomorphisms for mismatched prototypes. Indeed, we observe across 300 system-prototype pairs that fitted invariant sets by SPE for matched prototypes (reflecting the underlying dynamics in the data) are robust against random splits of the data and exhibit low variance of invariant set localization relative to mismatched prototype-

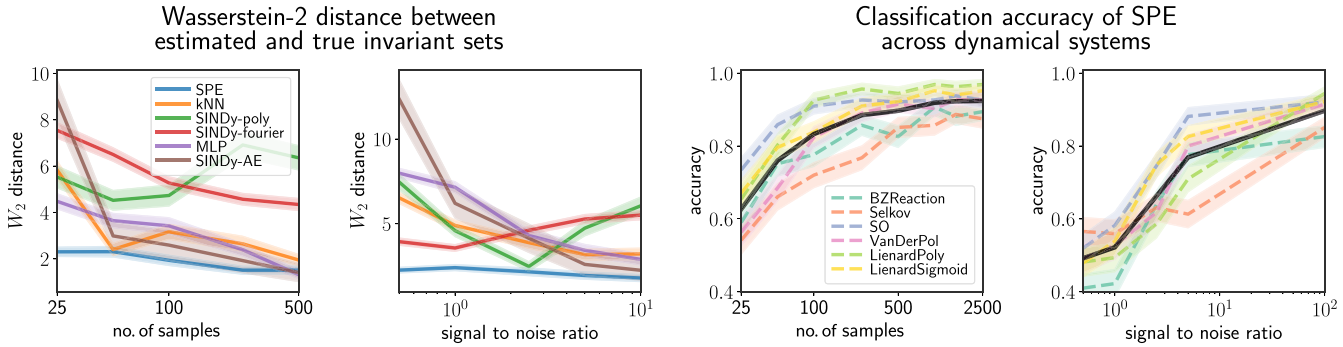


FIG. 3. Evaluation of invariant sets prediction across synthetic dynamical systems. (Left) Wasserstein (W2) distance between the estimated and true invariant sets, averaged over 1000 randomly sampled datasets (see Appendix C 2), produced by SPE along with additional baselines [k-nearest-neighbor interpolation (kNN), SINDy [28] using polynomial basis functions (SINDy-poly), SINDy using Fourier features (SINDy-Fourier), SINDy-AutoEncoders [48] (SINDy-AE), and a multilayer perceptron (MLP)], as a function of the number of observed points and signal-to-noise ratio (SNR). Default parameters: SNR $= 2.5$, $N = 100$ observed points. (Right) Classification accuracy of SPE, averaged over 1000 randomly sampled datasets (see Appendix C 5), across different dynamical systems [simple oscillator (SO), BZ reaction, Sel'kov, Liénard sigmoid, Liénard polynomial, and Van der Pol]. Average accuracy over all systems is shown in black. Default parameters: SNR $=$ 100, $N = 500$ observed points. In all plots, shaded areas correspond to the standard error of the mean.

system pairs (Fig. 4; more details in Appendix C 3). Thus, this measure of fitting variance, paired with the equivalence loss, can be used to detect out-of-dictionary scenarios, where the prototypes and data do not exhibit the same behavior.

B. Classifying dynamical systems with prototype equivalences

SPE can extend beyond the characterization of a given invariant set based on a single prototype to distinguishing between different potential prototypes using the relative loss in fitting multiple prototypes. As our prototype set, we choose all four effective behaviors implied by Eq. (3), with $a = \pm\frac{1}{4}$ and $\omega = \pm\frac{1}{2}$. Four diffeomorphisms $H_k(x)$ were fit to each observed system, and the one with the minimal value of the equivalence loss defined in Eq. (2) was used to determine class membership.

To test the capabilities of SPE in sparse and noisy scenarios, in Fig. 3 (right) we track the classification accuracy when decreasing the number N of observed vectors and under increasingly noisy scenarios, with low signal-to-noise ratios (SNRs) (see Appendix C 5 for details regarding data). The decline in performance is gradual along both of these axes; even when classifying using only 50 random vectors, average accuracy was still relatively high (approximately 70%). Beyond these results, in Appendix C 4 we also compared the performance of SPE on the task of classification in the low-noise regime to baselines that take as input a dense grid of the dynamics. We found that SPE outperforms all baselines on

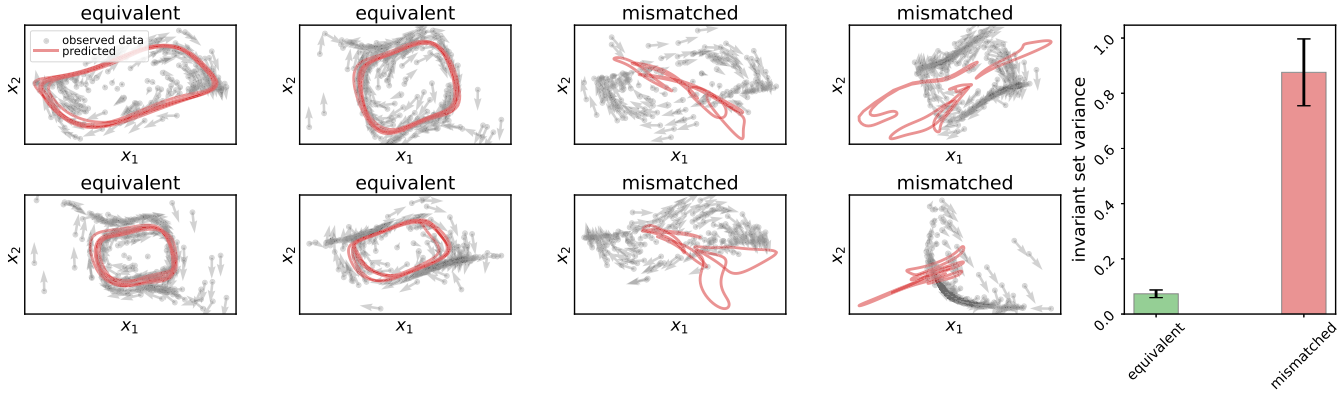


FIG. 4. Detecting prototype mismatch through variance of the estimated invariant set. (Left) Shown are representative examples of observed vectors from systems exhibiting limit cycles (gray, from the simple oscillator, BZ reaction, Sel'kov, Liénard sigmoid, Liénard polynomial, and Van der Pol families of systems; Appendix B 1), and predicted invariant sets by SPE (red) based on oscillatory prototypes with either clockwise or counterclockwise angular motion (Appendix C 3). Data and prototype are considered equivalent when they share the same direction of angular velocity, or mismatched when they do not. When the two systems are equivalent, the location of the estimated invariant set is robust to subsampling of the data (where randomly chosen 50% of the data points were omitted) (left two columns). Conversely, when there is a mismatch between the prototype and the system underlying the observed data, the learned diffeomorphism tends to overfit, and different random subsampling of the data yields substantially different estimated limit cycles (red) (right two columns). (Right) The variance in the estimated positions of the invariant set, averaged over the whole invariant set, is significantly higher for mismatched relative to equivalent prototype-system pairs ($p < 0.05$). The error bars represent the standard error of the mean of the variance.

average accuracy in the classification of the considered family of systems.

SPE's classification and localization performance is not constrained to limit cycles and node attractors and can be flexibly generalized to additional prototypes. To demonstrate such capabilities, we evaluated the performance of SPE on a challenging classification of multistable systems with a varying number of basins of attraction (Appendix G 2). Our results show that SPE can successfully distinguish between dynamical systems with varying numbers of fixed points and can additionally localize them successfully (Fig. 15). Furthermore, as the loss used to train SPE corresponds to a graded notion of equivalence, we found that SPE is also able to capture dominant behaviors of the system even when there is a mismatch between the true behavior and the chosen prototype (Appendix G 3).

C. Localizing and classifying invariant sets in the synthetic repressilator system

To test our approach in higher-dimensional systems, we first classify and localize invariant structures in a synthetic repressilator model of gene regulation [45] (see Appendix B 3 for governing equations). The repressilator models a genetic regulatory network in bacteria, based on the mRNA and protein counts of three genes—TetR, LacI, and λ phage cI—which inhibit each other in a cyclic fashion. The repressilator system acts as a simplified model for clock circuits in biological systems and was one of the first oscillating regulatory circuits that was synthetically engineered [45]. The repressilator system spans six dimensions, but under specific parameter values it exhibits either a point attractor or contains an embedded 2D limit cycle [49,50]. To model simulated data from this system, we use higher-dimensional prototypes that combine two behaviors: (1) a 2D limit cycle or equilibrium behavior as in Eq. (3); (2) exponential decay to a fixed point along variables uncoupled from the embedded 2D system. Mathematically, we define these prototypes as

$$g(y) = [g_{2D}(y_1),\ -\tau y_2]^T, \tag{4}$$

with $y_1 \in \mathbb{R}^2$ representing the first two dimensions of y and $y_2 \in \mathbb{R}^{d-2}$ the remaining dimensions. $g_{2D}(\cdot)$ are velocities in two dimensions, and τ is a decay factor. We used $\tau = 1/2$ in all our experiments.

We then classify repressilator systems into node or cycle attractors using SPE, when the systems vary in terms of transcription α, and protein and mRNA degradation rates β. For each system, we simulate $N = 1000$ observed pairs of positions and vectors using the same sampling scheme for the data as described in Sec. III A, Fig. 5. To compare the performance along different dimensionalities, we project the observed positions x_i and velocities $\dot{x}_i$ onto a predefined subset of the six dimensions (see Appendix B 3 for details). We observed (Fig. 5) that our model can classify two-dimensional repressilator systems better than earlier work [34], whose accuracy was 87% (next to our 89%), and that SPE can accurately distinguish point from cyclic behavior across all dimensionalities considered. Classification of this kind, carried out explicitly in higher dimensionalities, is infeasible using techniques from previous work. Moreover, our optimized diffeomorphisms in the six-dimensional case can be used to accurately locate the cyclic dynamics of the repressilator system, recapitulating the behavior of the system and its invariant set (Fig. 5, Appendix G 1).

D. Uncovering cell cycle expression patterns in single-cell data

To demonstrate the efficacy of SPE on empirical data, we show how it can be used to trace the periodic gene expression dynamics of proliferating cells. This demonstrates our method's ability to recover biologically meaningful periodic processes directly from high-dimensional, noisy, and sparse experimental data. Here, we analyze single-cell RNA-sequencing (scRNA-seq) data from two human cell lines, profiling over thousands of genes across thousands of cells [2,10]. Previous approaches have inferred cell cycle progression using marker genes [51], a learned projection [2,11], or using structural priors [1,3,4]. However, these existing approaches either do not allow for simulation of future states as they do not directly model the dynamics, or rely on explicit knowledge of cell cycle marker genes, information that is not necessarily available for other processes.

For each scRNA-seq dataset, we infer local velocities of cells in gene expression space using scVelo [25], which computes high-dimensional RNA velocity vectors for each of the sampled cells. We reduce the dimensionality of the gene expression profiles and the respective velocities using principal component analysis (PCA) to 10–100 dimensions (Fig. 6), enough to capture the variability in the data. This projection reduces the computational cost of the fitting procedure, while providing an approximate map back from the lower-dimensional prototype space to the full gene expression space. We then fit a high-dimensional limit-cycle attractor [as described in Eq. (4)] to the RNA velocity data, capturing the periodic nature of the cell cycle (Fig. 6, middle, see Appendix D for more details on data and prototype).

The expression of cell cycle genes follows a well-defined trajectory through different phases: initial growth (G1), DNA replication (S phase), further growth (G2), and mitosis (M). Utilizing the PCA projection, we can map the learned attractor back to gene expression space, revealing periodic, anticorrelated expression patterns of key cell cycle marker genes for the S and G2/M phases (Fig. 6, right).

E. Modeling bifurcations in early mouse embryo development

SPE can also be used to match biological data and prototypes defined by gradient systems. To demonstrate this, we considered a well-studied bifurcation in early development of the mouse embryo [52,53]—the differentiation of the inner cell mass (ICM) into either the epiblast (EPI) or the primitive endoderm (PrE), which happens between embryonic development days E3 and E4 [52,53] (Fig. 7, left). This differentiation dictates which of the cells become part of the embryo itself (originating from the epiblast) and those that act as supporting tissue (from the PrE). Previous works [6,52–54] have modeled this transition as a flow in a Waddington landscape [55] determined by a heteroclinic flip potential (shown in Fig. 7, middle left), which captures dynamics as a binary choice.

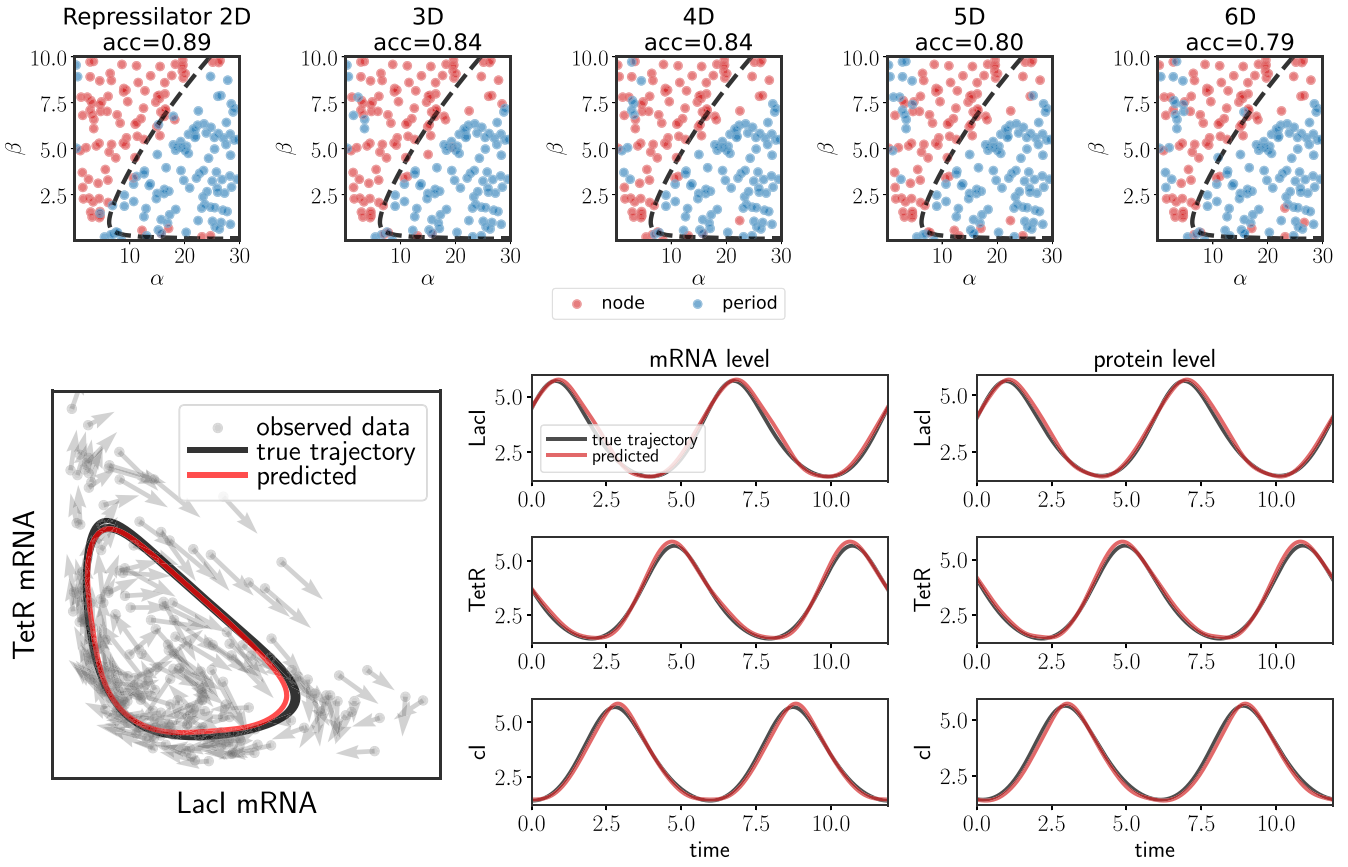


FIG. 5. Characterizing dynamics in higher dimensions and recovering the invariant structure of the repressilator genetic circuit. (Top) In each scatter plot, each point represents a repressilator system with different order parameters chosen to emulate a bifurcation between periodic and node behavior (full details in Appendix B 3). Each system was classified by SPE as either a 2D embedded limit cycle (blue) or node attractor (red). The dashed black line depicts the ground truth bifurcation boundary between periodic and node behavior. In each column, only a subset of dimensions were used for the classification, from 2D (left, LacI-TetR mRNA plane) to the full six dimensions (6D) (right). (Bottom) An example of the invariant set predicted by SPE (red) from the observed vectors (gray), projected from the 6D space to the 2D LacI-Tetr mRNA plane for visualization. A ground-truth trajectory was simulated (black) for visual comparison. Time series of the mRNA (center) and protein (right) expression levels can be extracted from the predicted limit cycle (red). These are overlaid on top of trajectories simulated from the underlying (hidden) system (black).

We use scRNA-seq data from Ref. [19], where mouse embryos were sequenced at different developmental time points. We were interested specifically in cells from embryonic days E3.5 and E4.5, which include the ICM, EPI, and PrE cell types (see Appendix E for preprocessing information). To infer these dynamics, we find pairings between the population of cells at E3.5 and cells at E4.5 using optimal transport [9,56,57]. Velocities $\dot{x}$ are then inferred from these pairings (full details in Appendix E), providing a noisy estimate of cellular trajectories on timescales compatible with the true dynamics.

Figure 7 (middle right) shows the results of using SPE to match these data and a prototype based on the heteroclinic flip potential (see Appendix E for the governing equations). Through this prototype, it is possible to localize both a source in the potential and the minima of the flip potential (Fig. 7). As this prototype is matched to the scRNA-seq, it is then possible to predict the expression of genes in landmarks or key trajectories of the model. Under the heteroclinic flip, points starting at positive and negative values of the horizontal axis flow directly downward until they reach a bifurcation point (Fig. 7, middle right). Figure 7 (right) shows the predicted gene expressions of key genes along these paths according to SPE, which are known to play a part in the decision between EPI and PrE: *Nanog*, *Gata6*, and *Fgf4*. *Nanog* plays a critical role in the differentiation process, but it was shown that it decreases in both fates [19], as predicted by SPE. *Gata6* is known to be exclusively expressed in PrE cells [58], matching the fall and rise in expression predicted by SPE in EPI and PrE fates, respectively. *Fgf4* has been shown to be expressed in EPI cells and not in PrE cells [19], again matching the trends predicted by SPE.

F. Localizing differentiation fates in hematopoiesis from lineage tracing data

Finally, we demonstrate how SPE can be used to localize cell fates in complex differentiation data with multiple cell fates. We focus on hematopoiesis, where progenitor cells residing in the bone marrow maintain a steady production of blood cells [20]. These progenitor cells progress through numerous fate decisions, before finally differentiating to their

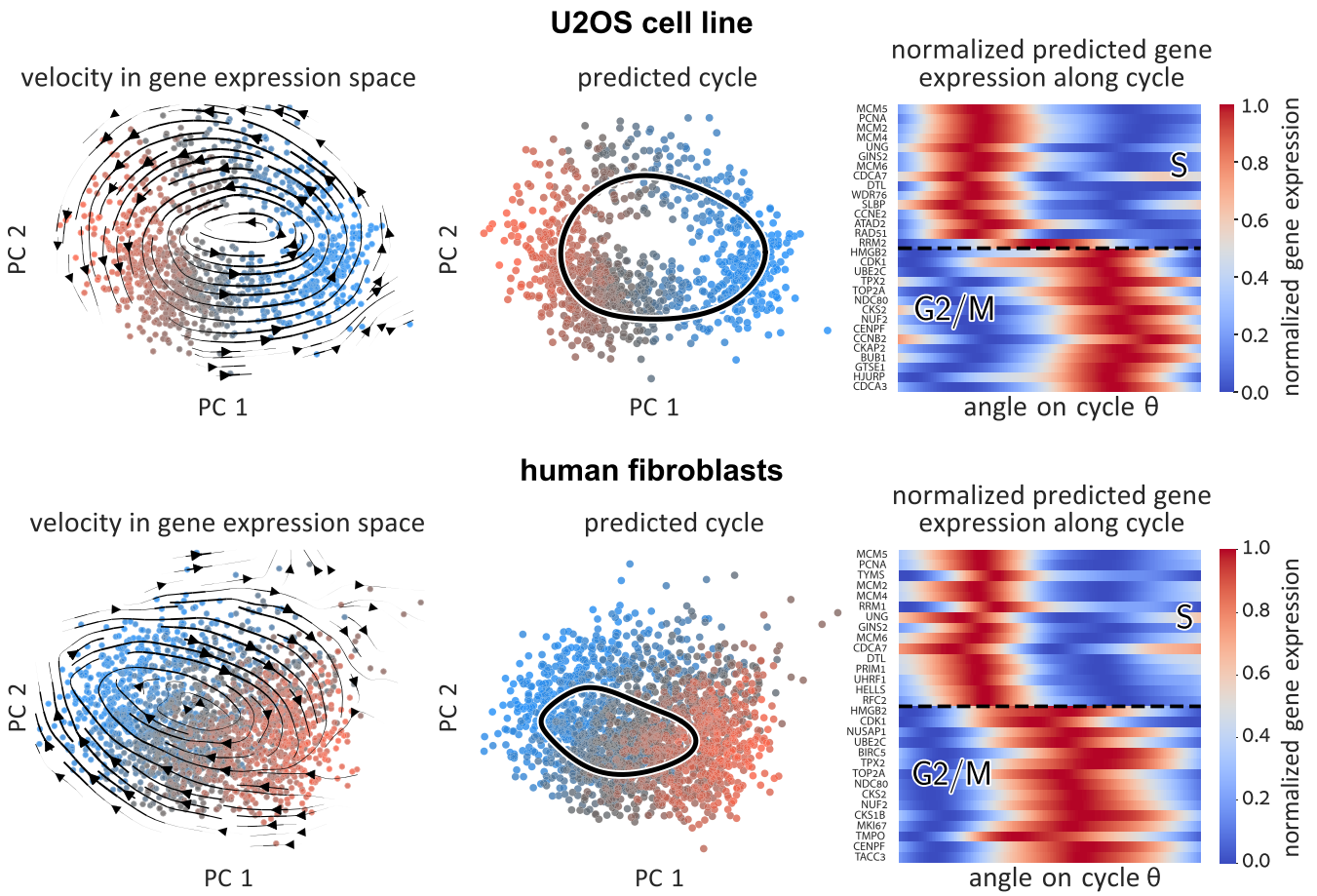


FIG. 6. Recovering the oscillatory gene expression patterns of the cell cycle from single-cell RNA-sequencing data. Results are shown for scRNA-seq of U2OS cell line (top) [10] and human fibroblast cells (bottom) [2]. (Left) Gene expression RNA velocity of cells undergoing proliferation in 2D PCA space, estimated using scVelo [25]. Colors denote the cycle phase score based on marker gene expression; red corresponds to high expression of S-phase markers, blue to high expression of G2/M-phase genes, and gray to low levels of both (details in Appendix D). (Middle) SPE was used to fit a periodic prototype from Sec. III C to PCA projections of the RNA velocity data (100 dimensions for U2OS data, 10 dimensions for human fibroblast data); the resulting attractor (black), along with the sampled cells (locations and colors correspond to left column), were projected onto 2D PCA space. (Right) Normalized gene expressions along the cycle attractor predicted by SPE, as a function of the angle along the limit cycle, θ, for a set of marker genes of S phase (top) and G2/M phase (bottom). See details in Appendix D.

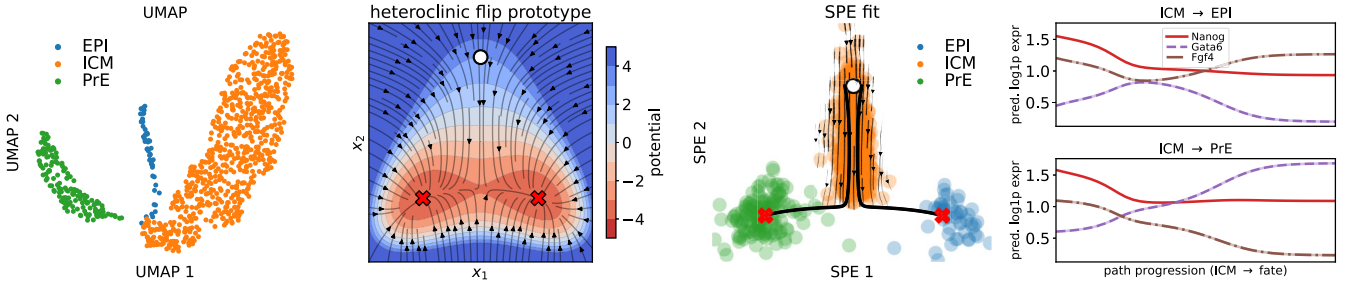


FIG. 7. Modeling a bifurcation in the mouse gut endoderm. (left) UMAP visualization of the data from Ref. [19] of early mouse embryo development. The inner cell mass (ICM) represents the major cell type in the early mouse endoderm (embryo day E3.5), which can then differentiate into epiblast (EPI) cells and primitive endoderm (PrE) cells (embryo day E4.5). Dynamics were inferred by matching between cells from E3.5 and those in E4.5 using optimal transport. (Middle left) The gradient of the heteroclinic flip potential function [52,53], depicted by the different levels in the contour plot, acts as the basis for the prototype used to fit the differentiation data, depicted by the stream plot. This prototype exhibits a low-energy region which acts as a "source" (white circle), flowing to two stable attractors (red crosses), in distinct energy wells. (Middle right) A streamplot of the bifurcating dynamics in SPE's fitted prototype space. The two black curves represent bifurcating trajectories to either the PrE fate (left) or EPI fate (right). (Right) Predicted log-transformed gene expressions of known genes in the decision between PrE and EPI: *Nanog* (red), *Gata6* (purple), and *Fgf4* (brown). The SPE-fitted prototype accurately recapitulates previously observed behavior of these genes during this decision process.

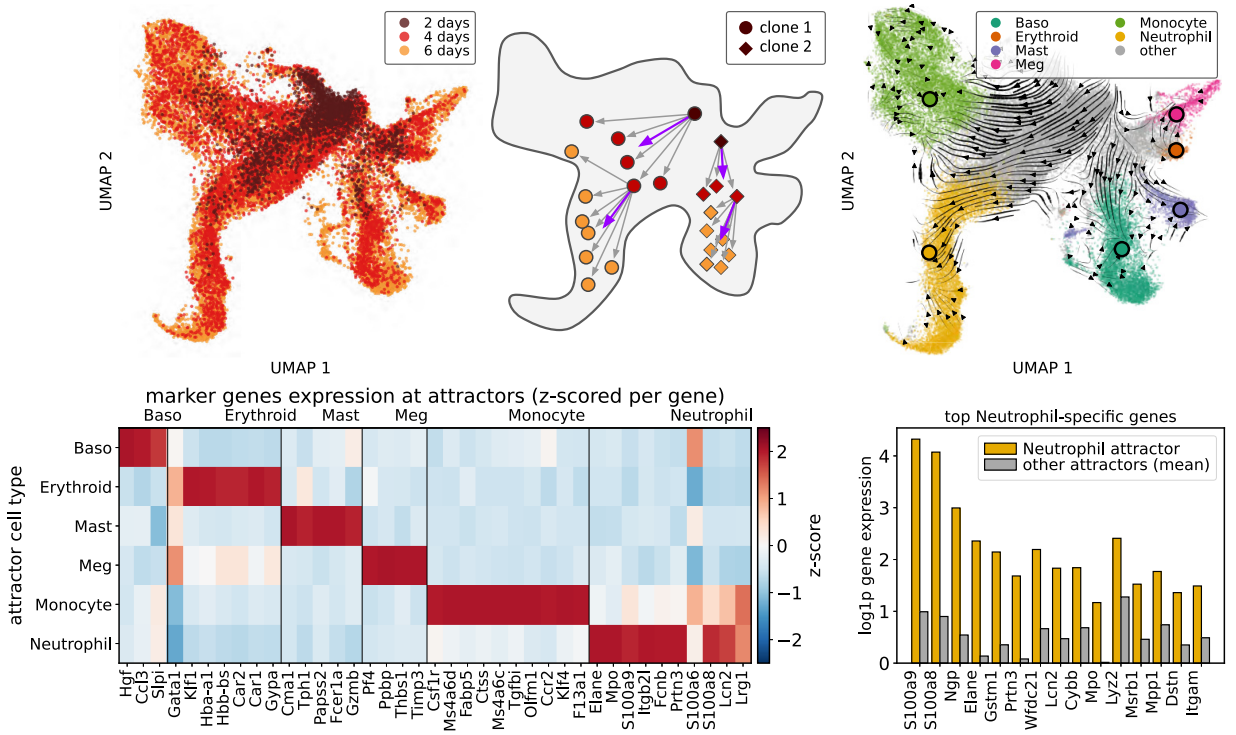


FIG. 8. Localizing differentiation fates in hematopoiesis from lineage tracing data. (Top left) UMAP visualization of the data colored by harvesting time after cultivation. (Top middle) A schematic of the procedure we used to infer dynamics. The hematopoiesis data from Ref. [20] follow clonal populations across time, collecting and sequencing clones at different days. Using this information, we infer dynamics by matching between clones at time t and those from time $t+1$, defining rates of change as the average direction from the clone at time t to that at time $t+1$. (Top right) A streamplot of the resulting dynamics, shown in UMAP space. The differentiation fate of six key cell types was then estimated from these data: basophils (Baso), erythrocytes (Erythroid), mast cells (Mast), megakaryocytes (Meg), monocytes, and neutrophils. A node attractor prototype was matched to each cluster using SPE, which enabled the localization of differentiation fates (represented by the colored circles). (Bottom left) A heatmap of z-scored expressions of key marker genes [20] in the inferred differentiation fate location, grouped by cell type. (Bottom right) Inferred log1p-transformed expressions of genes which are more highly expressed in the neutrophil attractor (orange bars) versus the other attractors (gray bars).

functional end states. Recent experimental advances have enabled researchers to track a heterogeneous population of progenitor cells at various stages of the differentiation process using lineage tracing methodologies [20,21], which provide precise observations of cellular dynamics in gene expression space. Applying SPE, we aim to localize the terminally differentiated fates in gene-expression space.

The lineage tracing data of Ref. [20] consist of *in vitro* clonal hematopoietic progenitor cells, sequenced 2, 4, and 6 days after plating (Fig. 8, top left). Each progenitor cell was tagged with a random inheritable DNA barcode, which can be matched across the different time points. Using this temporal information, we inferred the velocities $\dot{x}$ as the average direction of the pairing of each cell at time t to all of its clones at time $t+1$, as shown in the schematic in Fig. 8 (top middle) for two different clones (see Appendix F for details). These estimated velocities show a clear trend from undifferentiated cells (Fig. 8, top right, in gray) to the different fates (Fig. 8, top right). Having inferred velocities in this manner, we focused on clusters of differentiated cell types that account for more than 1% of the cells in the data. For each of these clusters, we used SPE to match between a node attractor prototype and the specific differentiated cell types (see Appendix F for fitting details). The positions of the localized attractors, relative to all other cells in the data, are represented by the colored circles in Fig. 8 (top right). To validate these results, we considered the differential gene expression of key marker genes [20] at the inferred position of the attractor for each of the cell types. In all of the cell types considered, the estimated nodes align well with the expected marker genes, as shown in Fig. 8 (bottom left). With access to these predicted nodes, it is possible to estimate the gene expression profile at the differentiation targets in the data. An example of this can be seen in Fig. 8 (bottom right), which shows 15 genes with the highest gene expression for the neutrophil cluster versus the differentiation fates of the other clusters. The inferred genes contain the set of marker genes considered in previous works [20,21], but also include additional genes that can be used as potential targets for more fully differentiated cell states (shown in Fig. 8, bottom left). These include genes known to be expressed in mouse neutrophils, such as *Ngp, Lyz2* and *Mpp1*. Used in this way, SPE can expose new, previously overlooked marker genes from data.

IV. DISCUSSION

Our proposed approach, smooth prototype equivalences (SPE), addresses a challenging problem in data-driven

dynamical systems: characterizing long-term behavior from sparse, noisy data. The key idea is to learn the correspondence between idealized prototype dynamics and their noisy, real-world counterparts in a way that reveals long-term structure. To do so, an invertible neural network is trained to capture the mapping from data space to prototype space. We demonstrated the efficacy and efficiency of this approach in identifying and tracing invariant set structures and classifying between multiple potential long-term behaviors on both challenging synthetic and real-world datasets.

While SPE relies on a set of predefined prototypes, it is a useful tool for characterizing dynamical systems from realistic data due to its relaxed notion of equivalence, where an optimized transformation between prototype and data can be found, so that even a partially matching prototype can be useful for the localization and categorization of dynamical behaviors (Appendix G 3).

To parametrize our diffeomorphism, we used invertible neural networks (INNs) based on affine coupling layers. To ensure that our INNs are expressive while remaining lightweight and easily differentiable, we implemented a specific form of coupling which we termed Fourier feature coupling. Other options for parametric invertible mappings exist, such as invertible ResNets [35,59] or Neural ODEs [27]. However, while these options are typically more expressive than INNs adapted from normalizing flows (NFs), they come with steep overheads in the calculations of the inverse function, Jacobian-vector products (JVPs), and their log-determinants. Therefore, we opted for sufficiently expressive networks where needed components (inverse, JVPs, log-determinants) could be easily calculated.

While in this work our primary focus is locally simple behaviors, such as node or cycle attractors or a small collection of node attractors (Appendix G 2), an important future step would entail the combination of multiple local dynamics into a larger, more complex system. This could be through the use of mixtures of experts [60], the incorporation of more complicated prototypes, or the combination of SPE together with other methodologies from the field of data-driven nonlinear dynamics. In all cases, SPE could be potentially used as an interpretable baseline component, capturing intricate dynamics in many dimensions. We also envisage that SPE can be paired with more advanced parametrizations of INNs in order to tackle more complex behaviors, in higher dimensional settings, which may take inspiration from computer vision [46,47,61].

We expect prototype-based dynamical matching as explored here to be applicable to many phenomena of interest beyond the scope presented in this work. Finding the locations of node attractors is of interest not only in differentiation (considered in Sec. III F) but also could help establish the end states in reprogramming [9] and determine the steady-state effects of chemical and environmental perturbations on gene expression. We also envisage that SPE can be used to match between interpretable biophysical models and empirical observations, where prior knowledge over the behavior of the biophysical model can be used to interrogate the data [52,53,62], as demonstrated in Sec. III E. Taking advantage of SPE's classification capabilities, an interesting future direction could involve the determination of which biophysical models and their invariant sets are associated with specific perturbations [6,54].

While versatile, SPE is currently limited by a number of factors. First, our focus in this work was the detection and classification of long-term behaviors that are robust against noise, such as limit cycles and node attractors. Indeed, our main focus were phenomena from empirical biological systems, where dynamics reflecting and driving collective cellular functions are robust against perturbations, including differentiation and oscillatory processes such as the cell cycle. At the other side of this scale are systems which are structurally unstable and include chaotic systems with strange attractors. Extending prototype equivalence approaches to such systems, whose underlying vector fields are sensitive to small perturbations, and where identifying smoothly equivalent prototypes could be a challenging task by itself, is an important future challenge. Another limitation of SPE originates in the characteristic sparsity of empirical scRNA-seq data. This sparsity means that multiple plausible mappings from prototype to data may all correctly capture the observed dynamics, meaning the problem is inherently ill-posed. A key assumption made in this work is that observed data are more likely to have been generated by dynamics that are related to the prototype by a smooth transformation close to the identity transformation, rather than a transformation which strongly perturbs small regions of input space. Such solutions are given preference by the added weight decay and determinant regularization, and are apt for the biological scenarios considered. However, these assumptions are inappropriate when the data exhibit an invariant set characterized by strong changes in both the low and high frequencies of input space. There may also be situations where the underlying system is not identifiable. For instance, if the portion of state space containing the invariant set is hidden, discerning between a node attractor and a limit cycle (or any other invariant behavior) or estimating the invariant behavior itself will be impossible, regardless of the classification methodology. SPE handles such scenarios, which plague all methods for modeling dynamical systems, by injecting strong prior information in the form of the chosen prototype to match to the data. Finally, SPE relies on velocity information in scRNA-seq, which is frequently noisy [63,64]. As shown in Fig. 3, SPE is robust against some noise, but if the method used to infer velocities is globally biased, SPE will inherit these flaws.

Our method directly addresses challenges faced in scientific domains such as biology, neuroscience, and physics. By offering an equation-free approach for the detection and classification of invariant sets, our method has the potential to advance the identification and interpretation of complex, nonlinear processes given sparse, noisy, high-dimensional experimental measurements.

ACKNOWLEDGMENTS

We thank D. Skinner and members of the Nitzan and Weiss labs for their suggestions and thoughtful feedback. This work was supported by the Alon Fellowship; the Israel Science Foundation (Grants No. 1079/21, No. 1901/26); and the European Union (ERC, DecodeSC, 101040660) (M.N.).

R.F., N.M., M.R. and M.N. conceptualized the work; R.F. implemented the software under the supervision of Y.W. and M.N.; R.F., N.M. and G.P. conducted experiments using the proposed framework; all authors interpreted the results, shaped the research, and wrote the paper.

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council.

DATA AVAILABILITY

All cell cycle data from Sec. III D are publicly available from Riba *et al.* [2] and Mahdessian *et al.* [10]. The mouse embryo development data from Sec. III E are publicly available from Nowotschin *et al.* [19] and the hematopoiesis data are publicly available from Wang *et al.* [21]. The code for SPE is publicly available [65].

APPENDIX A: SPE METHOD

The diffeomorphisms $H : \mathcal{X} \to \mathcal{Y}$ used for SPE are invertible neural networks (INNs) with alternating Affine and Fourier Feature Coupling (FFCoupling) transforms, both of which are explained in further detail below. Furthermore, the first layer of the normalizing flow was set to standardize the data, effectively an ActNorm layer [46], defined as

$$[\text{ActNorm(x)}]_i = \frac{x_i - \text{mean}(x_i)}{\text{std}(x_i)}. \tag{A1}$$

In the above, $\text{mean}(x_i)$ and $\text{std}(x_i)$ are the mean and standard deviations of the ith coordinate (i.e., the ith gene) of the observed data, respectively. This ActNorm layer was kept frozen after initialization.

1. Fourier feature coupling

The FFCoupling layer we used is a modification of the standard AffineCoupling [46,47] layer used in normalizing flows. AffineCoupling splits an input x into two subvectors x_1 and x_2, and the transformation itself is defined as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \text{AffineCoupling}(x) = \begin{bmatrix} x_1 \\ \exp[f_s(x_1)] \circ x_2 + f_t(x_1) \end{bmatrix}, \tag{A2}$$

where $f_s(\cdot)$ is a scaling function, $\exp[f_s(\cdot)]$ is the element-wise exponent of $f_s(\cdot)$, $f_t(\cdot)$ is a translation (or displacement) function, and $\circ$ stands for element-wise multiplication. In all of our use-cases, x_1 and x_2 were split into the first $d/2$ (rounded to the closest integer) dimensions of x for x_1 and the rest of the dimensions for x_2. AffineCoupling is defined in this manner to ensure invertibility, in which case $f_s(\cdot)$ and $f_t(\cdot)$ can be arbitrarily complex functions. In previous applications [47], these two functions are typically parametrized by a multilayer perceptron (MLP). Baked into the AffineCoupling transformation is the simple form of its log-determinant:

$$\ln|\text{AffineCoupling}(x)| = \sum_i [f_s(x_1)]_i. \tag{A3}$$

Since AffineCoupling only transforms x_2 as a function of x_1, it is standard to chain two transformations, one where x_1 acts on x_2 and another with the roles reversed. For simplicity, we regard such a chaining of two AffineCoupling transformations as a single layer.

When both the scale and the translation functions are defined as MLPs, calculating the Jacobian or Jacobian-vector products (JVPs) requires autograd operations and cannot be calculated in closed form. These JVPs need to be computed for SPE's loss [Eq. (2)] and can potentially be found using autograd methods, albeit at a high computational cost. Instead, we opt to use transformations whose JVPs can be calculated in closed form, while remaining expressive. Therefore, for our scale and translation functions, we use the following FFCoupling:

$$\mathcal{F}(x) = \sum_{k=1}^{K} \Theta_k \cos\left(\frac{2\pi}{R} k \cdot x + \phi_k\right), \tag{A4}$$

where $\cos(x)$ is an element-wise cosine. As defined, the frequency coefficients $\Theta_k \in \mathbb{R}^{d\times d}$ and phases $\phi_k \in \mathbb{R}^d$ are learnable parameters, while the number of frequencies K and the range R are hyperparameters. This transformation is a parametric form of a discrete cosine transform, such that when K is large, it can approximate any smooth function bounded in the range $[-R/2, R/2]$. It also admits a very simple form for the Jacobian, given by

$$\frac{\partial}{\partial x}\mathcal{F}(x) = -\frac{2\pi}{R}\sum_k k \cdot \Theta_k \,\text{diag}\left[\sin\left(\frac{2\pi}{R} k \cdot x + \phi_k\right)\right]. \tag{A5}$$

2. Affine transformation layer

In addition to the FFCoupling layers, we also use affine transformations. To ensure invertibility, we parametrize these transformations as

$$\text{Affine}(x) = WW^T x + e^{\varphi} \cdot x + \mu, \tag{A6}$$

where $W \in \mathbb{R}^{d\times q}$ is a low-rank weight matrix with rank q, $\varphi \in \mathbb{R}$ is a scalar added to the diagonal of the low-rank matrix $W^T W$ to ensure that the whole transformation is invertible, and $\mu \in \mathbb{R}^d$ represents a translation. W, φ, and μ are all learnable parameters. This factorization ensures that the matrix $WW^T + Ie^{\varphi}$ is positive-definite and thus invertible. It also enables efficient inversion using the matrix-inversion lemma:

$$(WW^T + Ie^{\varphi})^{-1} = Ie^{-\varphi} - W(W^T W + Ie^{\varphi})^{-1} W^T, \tag{A7}$$

which involves inverting a $q \times q$ matrix ($W^T W$), instead of a $d \times d$ matrix (WW^T).

3. Householder transformations

In addition to the Affine and FFCoupling layers, for higher dimension data we also used Householder transformations [46,66], which act as building blocks for orthogonal transformations. The Householder transformation is defined as

$$\text{HH}(x) = \left(I - \frac{2}{\|v\|^2} vv^T\right)x, \tag{A8}$$

where $v \in \mathbb{R}^d$ are the parameters of the transformation. This transformation is orthogonal, and so has a determinant of 1.

TABLE I. Hyperparameter settings.

Experiment	Number of blocks	K	iterations	learning rate	weight decay	λ_{det}	λ_{proj}	prototype a, ω
Fig. [2–4, 11, 16]	3	4	500	10^{-3}	10^{-3}	10^{-3}	–	± 0.25, ± 0.5
Fig. 5	3	5	600	10^{-3}	10^{-3}	10^{-3}	e^{-1}	± 0.25 , ± 2
Fig. 6	3	3	1,000	5×10^{-4}	10^{-3}	0	e^{-2}	0.1, 5
Table IV	3	5	600	10^{-3}	0	0		± 0.25, ± 0.5

Furthermore, it can be shown [66] that any rotation in a d-subspace can be described by $d-1$ of these transformations. Thus, these Householder layers can directly rotate space, unlike the Affine transformations defined above, which are more constrained.

4. Network specifications

As mentioned above, the INNs we use have alternating Affine and FFCoupling transforms. A single block of our network is defined as Affine → FFCoupling → ReverseFFCoupling, where the ReverseFFCoupling switches the roles of x_1 and x_2, as explained in Appendix A 1.

Before these blocks, our networks include a frozen ActNorm layer that standardizes the data and ensures that all learned diffeomorphisms have similar inputs, and then a full-rank Affine transformation with learnable weights. For the experiments performed in Sec. III D and Appendix G 2, we added two Householder transforms after the frozen ActNorm layer and before the Affine and FFCoupling layers.

5. Projection regularization

For high-dimensional prototypes [Eq. (4)], the diffeomorphisms are regularized to map the data as close as possible to the invariant sets of the embedded 2D dynamics, in accordance with the prototypes construction. Specifically, the observed points are projected onto the 2D invariant set of the prototype using the diffeomorphism, and then back into data space as follows:

$$P(x;H_\theta) = H_\theta^{-1}([H_\theta(x)]_1, [H_\theta(x)]_2, 0, \ldots, 0), \tag{A9}$$

where $[H_\theta(x)]_i$ is the ith coordinate of $H_\theta(x)$ and $P(x;H_\theta)$ denotes the projection according to $H_\theta(x)$.

The projection into data space is modeled by a Gaussian observation model, with the following loss added when training:

$$L_{\text{proj}}(H_\theta, g, \lambda_{\text{proj}}) = e^{\lambda_{\text{proj}}} \frac{1}{N} \sum_i^N \|x_i - P(x_i;H_\theta)\|^2 - \frac{1}{N}\lambda_{\text{proj}}, \tag{A10}$$

where λ_{proj} is a regularization coefficient, which takes the form of the log of the precision of the Gaussian observation model. Adding the regularization coefficient in this manner means that it can be optimized in parallel to the diffeomorphism, and allows for an adaptive fitting process. When modeled in this way, the projection is similar to generative topographic mapping (GTM) [67] when the matching between the latent codes and the data is known.

6. Optimization criteria

The main term in the optimization objective of our diffeomorphisms is the equivalence loss [Eq. (2)], given by

$$L_{\text{E}}(H_\theta, g) = \frac{1}{N} \sum_{i=1}^N \left\| \frac{\partial_{x_i} H_\theta(x_i)\dot{x}_i}{\|\partial_{x_i} H_\theta(x_i)\dot{x}_i\|} - \frac{g(H_\theta(x_i))}{\|g(H_\theta(x_i))\|} \right\|^2. \tag{A11}$$

In addition to the above loss, we use several regularizations to encourage convergence, such as the projection regularization described in Appendix A 5 and weight decay. We also penalize networks excessively expanding or contracting space by regularizing the absolute value of the log-determinant to be close to zero:

$$\text{L}_{\text{det}}(H_\theta) = \frac{1}{N} \sum_{i=1}^N |\ln \det[\partial_{x_i} H_\theta(x_i)]|. \tag{A12}$$

The full loss used to optimize the diffeomorphism H in SPE is thus given by

$$L_{\text{tot}}(H_\theta, g) = L_{\text{E}}(H_\theta, g) + L_{\text{proj}}(H_\theta, g, \lambda_{\text{proj}}) + \lambda_{\text{det}}\text{L}_{\text{det}}(H_\theta), \tag{A13}$$

where $\lambda_{\text{det}} > 0$ is a regularization coefficient.

All training is carried out using the Adam optimizer [68] with weight decay.

7. Hyperparameter settings

For all simulated systems, hyperparameters were chosen to maximize performance on a separate, held-out validation set of 100 datasets which were not used for evaluation. In all experiments, unless specified otherwise, we used the simple oscillator (SO) prototype defined in Eq. (3). The hyperparameters of the prototype are a, the radius of the limit cycle, and ω, the angular velocity.

As our INNs are relatively small, and most of the data are of a low dimension (relative to typical machine-learning tasks), a single CPU with 16 GB RAM was sufficient for training our networks. Fitting a single INN takes less than 10 s for all of the 2D simulated data and approximately 90 to 120 s for the cell-cycle data. Hyperparameter settings for all experiments are summarized in Table I.

8. Effects of hyperparameters on performance

The classification results by SPE were robust against the choice of number of frequencies and layers.

In our experiments, we remained within the robust range demonstrated in Table II, avoiding too many frequencies that can lead to overfit to overly complex estimations of the invariant set (Fig. 9), while also avoiding too few frequencies

TABLE II. Classification performance for different numbers of layers and frequencies of the trained invertible neural network. Numbers correspond to the average accuracy over 1000 different systems uniformly sampled from the set of systems in Table III, with $N = 250$ observations per system.

Frequencies (K)	2	3	4	5	6	7
2 layers	88.7%	84.9%	89.1%	85.8%	84.7%	81.3%
3 layers	84.9%	86.1%	84.9%	83.6%	83.2%	80.1%
4 layers	84%	86.8%	83.5%	81.5%	79.7%	80.9%
5 layers	82.2%	84.9%	82.5%	80.3%	77.8%	77.3%

and layers to keep the network expressive enough to capture nonelliptical invariant sets.

If many frequencies are needed, for instance when the invariant set has a complex structure, we found it beneficial to begin training with frozen coupling layers, as shown in Fig. 10. This essentially initializes the network at the solution found when the diffeomorphism is constrained to be affine, after which the behavior is fine-tuned using the available frequencies. More examples of the use of SPE can be found in Fig. 11.

APPENDIX B: SIMULATION DETAILS

1. 2D systems

For the 2D simulated dynamics (Sec. III A), we use the same systems as in Ref. [34], whose governing equations appear in Table III. We additionally study the Augmented Simple Oscillator (Augmented SO) system for comparison in Table IV, which is an instance of a simple oscillator (SO) whose 2D phase space was transformed by a random initialization of a neural spline flow [61]. All of these dynamical systems (Table III) undergo a Hopf bifurcation and change their dynamics from exhibiting node attractors to limit cycles, at known parametric settings, making them particularly relevant for our application.

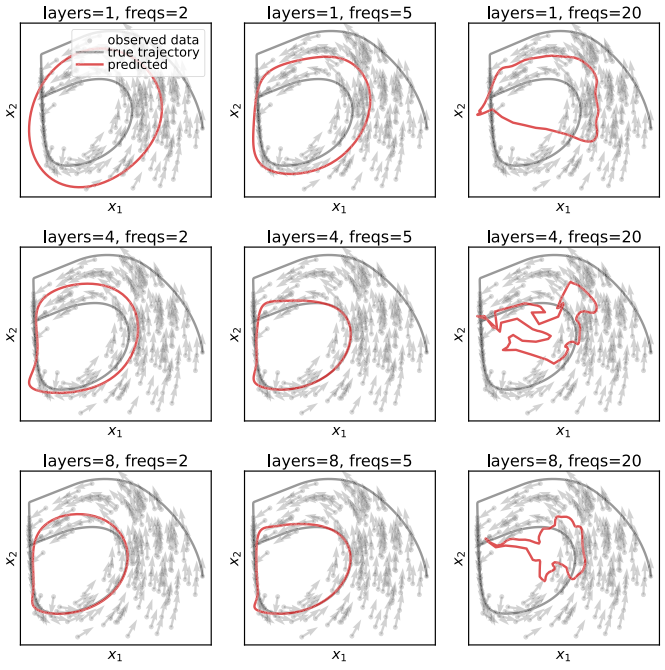


FIG. 9. Estimation of limit cycles using different numbers of layers and frequencies in the diffeomorphism H_θ. Examples of invariant sets predicted by SPE (red) from observed vectors (gray), for a BZ reaction system (Appendix B 1) exhibiting a limit cycle attractor. Ground-truth trajectories were simulated for each hidden system (black) for visual comparison.

2. Simulating sparse data

A straightforward method to draw sparse data from each dynamical system is to draw observed points uniformly at random from the simulated phase space. However, this process is generally inconsistent with sparse measurements collected in experiments, and for high-dimensional systems this runs the risk of under-sampling the invariant set region, which is a prerequisite for meaningful predictions. Instead, we resort to the following data sampling process, where each observed pair $(x_i, \dot{x}_i)$ is generated by simulating a trajectory starting from a prespecified range of the phase space for a randomly drawn time:

(1) For each $i \in \{1, \ldots, N\}$, draw the initial conditions $x_i^{(0)}$ uniformly from a prespecified portion of phase space.

(2) Sample t_i uniformly in the range $[T_{\min}, T_{\max}]$.

(3) Simulate a trajectory beginning at $x_i^{(0)}$ at time $t = 0$ up to time t_i, resulting in an observed position and velocity: $x_i^{(t_i)}$ (denoted as x_i for short), $\dot{x}_i = f(x_i)$.

(4) Optionally, Gaussian noise with standard deviation σ is then added to $\dot{x}_i$.

We set the parameter $T_{\min} = 0$ for all simulated sparse data in this study (Fig. 2–4), except for the repressilator, where it is set to $T_{\min} = 3$ to ensure that when the samples are projected to fewer than six dimensions, they still define a vector field with nonintersecting trajectories. $T_{\max}$ is a hyperparameter that effectively controls how much samples tend to concentrate around the invariant set. An example of the effect of different values for $T_{\max}$ can be seen in Fig. 12. When $T_{\max}$ is close to 0, the pairs $(x_i, \dot{x}_i)$ revert back to being uniformly sampled from phase space. When $T_{\max}$ is small, the observations mostly include transients of the dynamics. As $T_{\max}$ increases, more points aggregate near the invariant set, which is in turn easier to estimate. For our 2D simulations, we use $T_{\max} = 3$, whereas for the repressilator system we used $T_{\max} = 10$. A higher $T_{\max}$ value was used for the repressilator as trajectories in this higher-dimensional system take longer to converge to the invariant set.

All trajectories are simulated with the Runge-Kutta method (RK4) and a step size of $\Delta t = 0.01$.

3. Repressilator model

The repressilator system is governed by six coupled differential equations, structured as three instances of the following gene-regulation model [45]:

$$\dot{m}_i = -m_i + \frac{\alpha}{1 + p_j^n} + \alpha_0,$$

$$\dot{p}_i = -\beta(p_i - m_i),$$

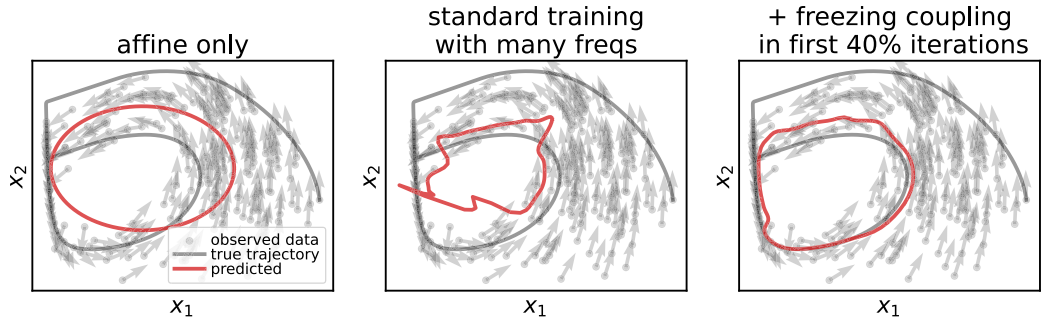


FIG. 10. Initializing the network from the solution of an affine-only diffeomorphism improves fitting. (Left) when the invariant set is not an ellipse, using a diffeomorphism defined only by an affine transformation does not faithfully capture the underlying invariant set. (Middle) if too many frequencies are used with SPE in such a case, the model overfits. (Right) when freezing all coupling layers for the initial iterations of the fitting procedure, SPE converges more faithfully to the true invariant set.

where m_i and p_i denote the mRNA and protein concentrations for gene i, corresponding to LacI, TetR, and λ cI, with cyclic inhibition: $i =$ LacI, TetR, cI and $j =$ cI, LacI, TetR. The parameters α_0 (basal transcription), α (maximal transcription), β (protein and mRNA degradation rates), and n (Hill coefficient) define the system's regulatory dynamics.

We analyze trajectories across $\alpha \in (0, 30)$, $\beta \in (0, 10)$, with $\alpha_0 = 0.2$, $n = 2$. A supercritical Hopf bifurcation emerges at critical values of β [49]:

$$\beta_1 = \frac{3A^2 - 4A - 8}{4A + 8} + \frac{A\sqrt{9A^2 - 24A - 48}}{4A + 8},$$
$$\beta_2 = \frac{3A^2 - 4A - 8}{4A + 8} - \frac{A\sqrt{9A^2 - 24A - 48}}{4A + 8},$$

where

$$A = \frac{-\alpha n \hat{p}^{(n-1)}}{(1 + \hat{p}^n)^2}, \quad \hat{p} = \frac{\alpha}{1 + \hat{p}^n} + \alpha_0. \tag{B1}$$

These are the boundaries plotted in Fig. 5 (top).

APPENDIX C: BASELINE DETAILS

1. Baselines in invariant set localization

In Sec. III A we showcase the ability of SPE to estimate the position of limit cycles from observations. The few existing approaches that can do so rely on data that is dense on a regular grid [69,70]. Instead, we compare the quality of the fit of SPE to other methods which take as input sparse data and estimate an ODE, and in particular how close limiting structures of the ODEs determined by these different methods approximate the true underlying invariant set. Accordingly, we compared SPE to a number of baselines:

(1) *kNN.* a simple interpolator which uses the k-nearest observed positions in order to estimate the velocity. In our experiments, we found that $k = 5$ neighbors approximates clean systems fairly well and used that setting for all of our comparisons.

(2) *SINDy* [28,71] is an established method for estimating vector fields from empirical observations. SINDy uses sparse linear regression with a fixed library of functions to estimate the velocities of the system. For our evaluations, we use the pysindy package [71] with a function library composed of either third order polynomials (SINDy-poly) or 10 Fourier features (SINDy-Fourier). All other hyperparameters were set according to the default values in the pysindy package.

(3) *SINDy-AE.* A method that fits a SINDy-based estimator of dynamics in the latent space of an AutoEncoder [48]. For our evaluations, we used a fixed library of third-order polynomial functions, where both the encoder and decoder were MLPs with two hidden layers, a width of 32, SiLU activations [72], and trained with the Adam optimizer [68] for 5000 iterations. We used a learning rate of 10^{-3}, weight decay, ℓ_1 and ℓ_2 regularizations of 10^{-3} on the network parameters and an ℓ_1 regularization of 10^{-5} on the SINDy weight matrix.

(4) *MLP.* A simple multilayer perceptron with two hidden layers, with a width of 128, and SiLU activations [72]. These MLPs were trained to estimate the velocities from the positions using an mean-square error loss over the predicted velocities, with the Adam optimizer [68], for 5000 (full-batch)

TABLE III. Simulated dynamical systems. Columns correspond to dynamical system type, corresponding governing equations, range of phase space used for initial conditions, and range of parameter values used in our simulations, and parameter settings that correspond to a limit cycle.

System Name	Equation	Initial cond.	Parameter range	Cycle condition
Simple Oscillator (SO)	$\dot{r} = r(a - r^2),\ \dot{\theta} = \omega$	$x_1, x_2 \in [-1, 1]$	$a \in [-0.5, 0.5], \omega \in [-1, 1]$	$a > 0$
Liénard Polynomial	$\dot{x}_1 = x_2,\ \dot{x}_2 = -(ax_1 + x_1^3) - (c + x_1^2)x_2$	$x_1, x_2 \in [-4.2, 4.2]$	$a \in [0, 1], c \in [-1, 1]$	$c < 0$
Liénard Sigmoid	$\dot{x}_1 = x_2,\ \dot{x}_2 = -(1/(1 + e^{-ax_1}) - 0.5) - (b + x_1^2)x_2$	$x_1, x_2 \in [-1.5, 1.5]$	$a \in [0, 1], b \in [-1, 1]$	$b < 0$
Van der Pol	$\dot{x}_1 = x_2,\ \dot{x}_2 = \mu x_2 - x_1 - x_1^2 x_2$	$x_1, x_2 \in [-3, 3]$	$\mu \in [-1, 1]$	$\mu > 0$
BZ Reaction	$\dot{x}_1 = a - x_1 - \frac{4x_1 x_2}{1+x_1^2},\ \dot{x}_2 = bx_1(1 - \frac{x_2}{1+x_1^2})$	$x_1, x_2 \in [0, 10]$	$a \in [2, 19], b \in [2, 6]$	$b < \frac{3a}{5} - \frac{25}{a}$
Sel'kov	$\dot{x}_1 = x_1 + ax_2 + x_1^2 x_2,\ \dot{x}_2 = b - ax_2 - x_1^2 x_2$	$x_1, x_2 \in [0, 3]$	$a \in [0.01, 0.11], b \in [0.02, 1.2]$	$b^{-2} > 2 - 4a \pm \sqrt{1 - 8a}$

TABLE IV. Comparison of the classification accuracy of SPE to different models across various families of dynamical systems, when classifying between periodic and node dynamics. The classification accuracy is averaged over 100 instanced for each family of system (see Appendix B 1 for details). The observed vectors were positioned on a dense 64×64 grid, with the addition of Gaussian noise with a standard deviation of $\sigma = 0.1$ on the velocities. Where relevant, the errors in accuracy are one standard deviation, calculated over different initializations. The highest scores for each column are shown in bold, with the second-highest scores underlined.

	SO	Aug. SO	Liénard Poly	Liénard Sigmoid	Van der Pol	BZ Reaction	Sel'kov	Attractor estimation
SPE	91 ± 1	81 ± 1	$\mathbf{96 \pm 1}$	$\mathbf{92 \pm 1}$	$\mathbf{97 \pm 3}$	$\mathbf{88 \pm 2}$	$\mathbf{68 \pm 1}$	$\surd$
TWA	$\mathbf{98 \pm 1}$	$\mathbf{93 \pm 1}$	$\underline{86 \pm 13}$	$\mathbf{92 \pm 7}$	83 ± 15	82 ± 11	$\underline{65 \pm 3}$	$\times$
Critical Points	54	56	70	49	56	$\underline{84}$	51	$\times$
Phase2vec	73 ± 8	71 ± 4	49 ± 6	48 ± 0.1	49 ± 4	50 ± 0.1	49 ± 0.1	$\times$
Autoencoder	$\underline{95 \pm 3}$	$\underline{87 \pm 1}$	63 ± 16	88 ± 9	81 ± 16	82 ± 13	49 ± 2	$\times$

iterations. We used a learning rate and weight decay of 10^{-3}. Empirically, we observed that these hyperparameters result in MLPs that accurately approximate systems from $N = 1000$ noiseless observations.

All of these methods (including SPE) take as input the sparse $(x, \dot{x})$ data and can estimate velocities at new positions, $\dot{x} = f_\theta(x)$, effectively defining an ODE. Using these velocity estimates, it is possible to simulate long trajectories which converge at the fitted system's invariant set.

A qualitative example of the predictions made by the different models on noisy limit cycle data (with an SNR of 2.5) can be seen in Fig. 13. All models are competent at predicting dynamics in previously unseen regions of phase space. However, except for SPE, none of these methods constrain the solution to contain only a limit cycle. Moreover, even when the ODEs by the baselines exhibit limit cycles, it is difficult to ascertain the position of the limit cycle itself, which is one of the main contributions of SPE.

2. Evaluation of invariant set localization

To quantitatively evaluate the localization of the invariant set, we simulate trajectories from initial conditions in both the true (hidden) system and in the fitted systems. These long trajectories give an effective stationary distribution of points along the invariant set of the considered systems. Accordingly, our goodness-of-fit measure is the Wasserstein-2 (W_2) distance between effective stationary distributions of the hidden governing equations and those generated by the estimated velocities for each model.

For the evaluations in Fig. 3 (left), we randomly sampled 1000 systems that contain limit cycle behaviors, uniformly from the set containing SO, Liénard Polynomial, Liénard Sigmoid, Van der Pol, BZ Reaction, and Sel'kov families of systems (Appendix B 1), with order parameters uniformly sampled from the ranges corresponding to limit cycle behaviors. For each system, different numbers N of positions x_i and velocities $\dot{x}_i$ were sampled according to the procedure described in Appendix B 2, with $T_{\max} = 3$. Additive Gaussian noise was then added to these velocities with a standard deviation of $\sigma = \mu/\mathrm{SNR}$, where $\mu = \frac{1}{N}\sum_i \|\dot{x}_i\|$ and we use a prespecified signal-to-noise ratio (SNR). Adding noise in this manner allows for direct control of the SNR of the observed velocities between different families of systems, whose domains are defined at different scales. Each of the methods, SPE and the baselines described in Appendix C 1, was then fitted to all N simulated positions x_i and noisy velocities $\dot{x}_i$.

Each method was then evaluated by comparing its effective stationary distribution to that of the ground-truth system. To do so, $M = 1000$ initial conditions were sampled from the range of initial conditions of the corresponding family of systems (see Appendix B 1). The same initial conditions were used to generate long trajectories under the true system and the one determined by the fitted velocities, and the W_2 distance was then calculated between the last time points for the true system and the fitted ODE. The trajectories were integrated using the Runge-Kutta method, with a step size of $\Delta t = 0.01$, for $T_{\mathrm{eval}} = 100$, which we found to be long enough for the true system to be close to its invariant set. To calculate the W_2 distance, we use the PYTHON OPTIMAL Transport package from Ref. [56].

3. Evaluation given prototype mismatch

To evaluate the effects of prototype mismatch, when the input prototype is inconsistent with the dynamical systems underlying the observed data, we considered a scenario where both prototypes and data exhibit limit cycles with either clockwise (CW) or counterclockwise (CCW) angular motion. Because our parametric diffeomorphisms (Sec. II C) cannot change the orientation of the motion, this results in a potential mismatch: if the data contain CW motion but the prototype has CCW motion (or vice versa), then the two are mismatched. Otherwise, the data and prototype are considered equivalent. For CW-based systems, we used the simple oscillator (SO) with $\omega < 0$, Sel'kov, Liénard sigmoid, Liénard polynomial, and Van der Pol family of systems from Appendix B 1. For systems with CCW motion, we used SO with $\omega > 0$ and the BZ-reaction family of systems. The prototypes in these experiments have the same settings as in Sec. III A: an SO system with $a = 0.25$ and $\omega = 0.5$ for CCW motion and $\omega = -0.5$ for CW angular velocity. All trained networks used four blocks with a width of 5 and were trained for 500 iterations with a learning rate and weight decay of 10^{-3}.

We observed that when there is a mismatch between data and prototype, the diffeomorphism tends to overfit and can change drastically under small perturbations of the data. To quantify these effects, we fit five different diffeomorphisms to five random splits of the data, where 50% of the data points is randomly omitted. We then calculated the variance in the position of the inferred localized invariant set under the different splits.

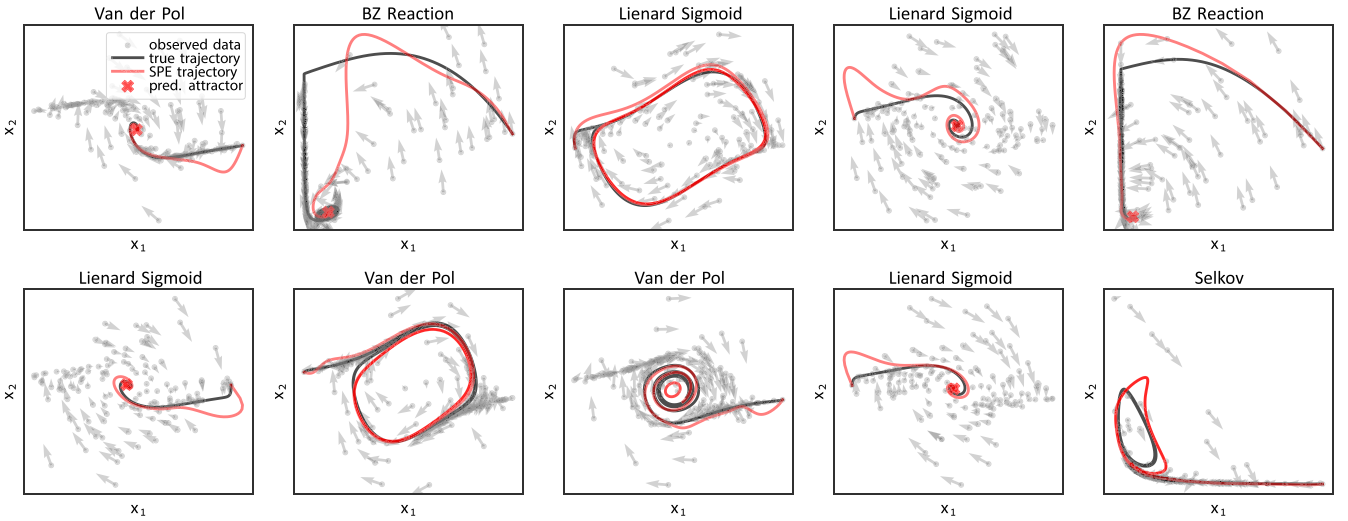


FIG. 11. Randomly picked examples of localizing invariant sets—limit cycles and node attractors—using SPE on synthetic data from Appendix B 1. In all of the above, solid black lines depict ground-truth trajectories simulated according to the true governing equations which were used to generate the data, represented by the gray vectors. The solid red lines are trajectories predicted by SPE. SPE localizes the invariant set from the data, even when transient states are ambiguous in the observed data.

4. Benchmarking classification results

To benchmark SPE in the classification of dynamical systems, we turn to datasets and existing baselines previously considered by Ref. [34]. Below we provide a brief summary of each of these methods:

(1) *TWA* [34]. A convolutional NN trained to classify between point and periodic dynamics on a dense grid. Instead of training on the input vectors directly, TWA predicts the class of the system using an angular representation of each vector at each position. TWA is trained on simulated data, which is generated by sampling the order parameters from the SO family of systems. These are then augmented using neural spline flows [61]. Augmenting in this way ensures that the same invariant structure is kept between the originally sampled SO system and the augmented version. TWA was then trained on 10 000 of these randomly augmented systems.

(2) *Critical Points* is based on an algorithm devised by Ref. [69] which detects and characterizes critical points from a representation of an autoencoder. A system is considered periodic under CriticalPoints if at least one detected point is a repeller and is otherwise classified as an attractor.

(3) *Phase2vec* [73]. A convolutional autoencoder trained to reconstruct 10 000 training systems defined by sparse polynomial equations. The encoder maps to a 100-dimensional latent space using a single layer. The outputs of the decoder are the predicted coefficients for the sparse polynomial equations. These coefficients are then multiplied by the polynomial function library to reconstruct the original vector field. The latent representations of the autoencoder then form features for an additional linear classifier, which was trained for classification using the same dataset as TWA.

(4) *Autoencoder.* A convolutional autoencoder with a bottleneck of 10 latent dimensions, trained to directly reconstruct the vectors generated as in Phase2vec. Both encoder and decoder have three convolutional blocks with a stride of 2×2 to reduce the resolution of the input during training. After training, similarly to Phase2vec, the latent representations are turned into features for a linear classifier fitted to the same data as TWA.

To benchmark SPE against existing methods for classification between node and cyclic behaviors, we use the dataset introduced by Ref. [34]. This dataset is based on the simulated

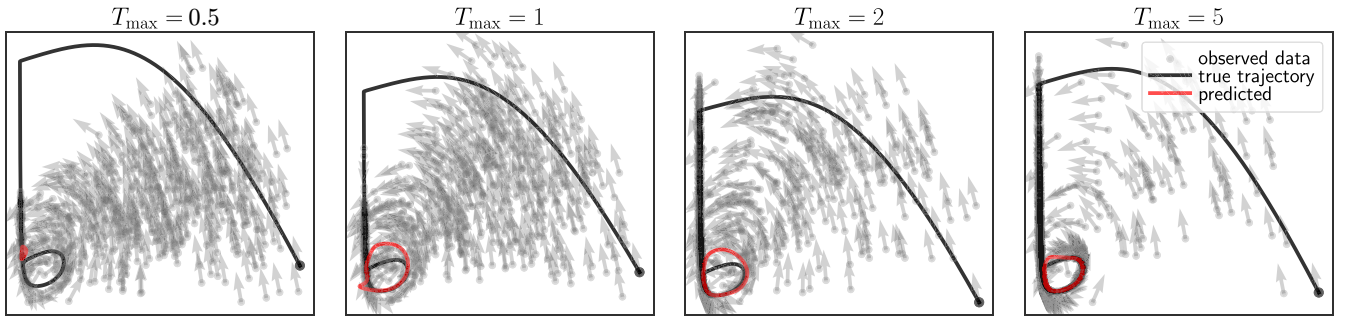


FIG. 12. The effects of changing the maximum simulation sampling time $T_{\max}$. When $T_{\max}$ is small (i.e., $T_{\max} = 0.5$; left), most of the observations are transients. On the other hand, when $T_{\max}$ is large (i.e., $T_{\max} = 5$; right), most of the observations are on the invariant set.

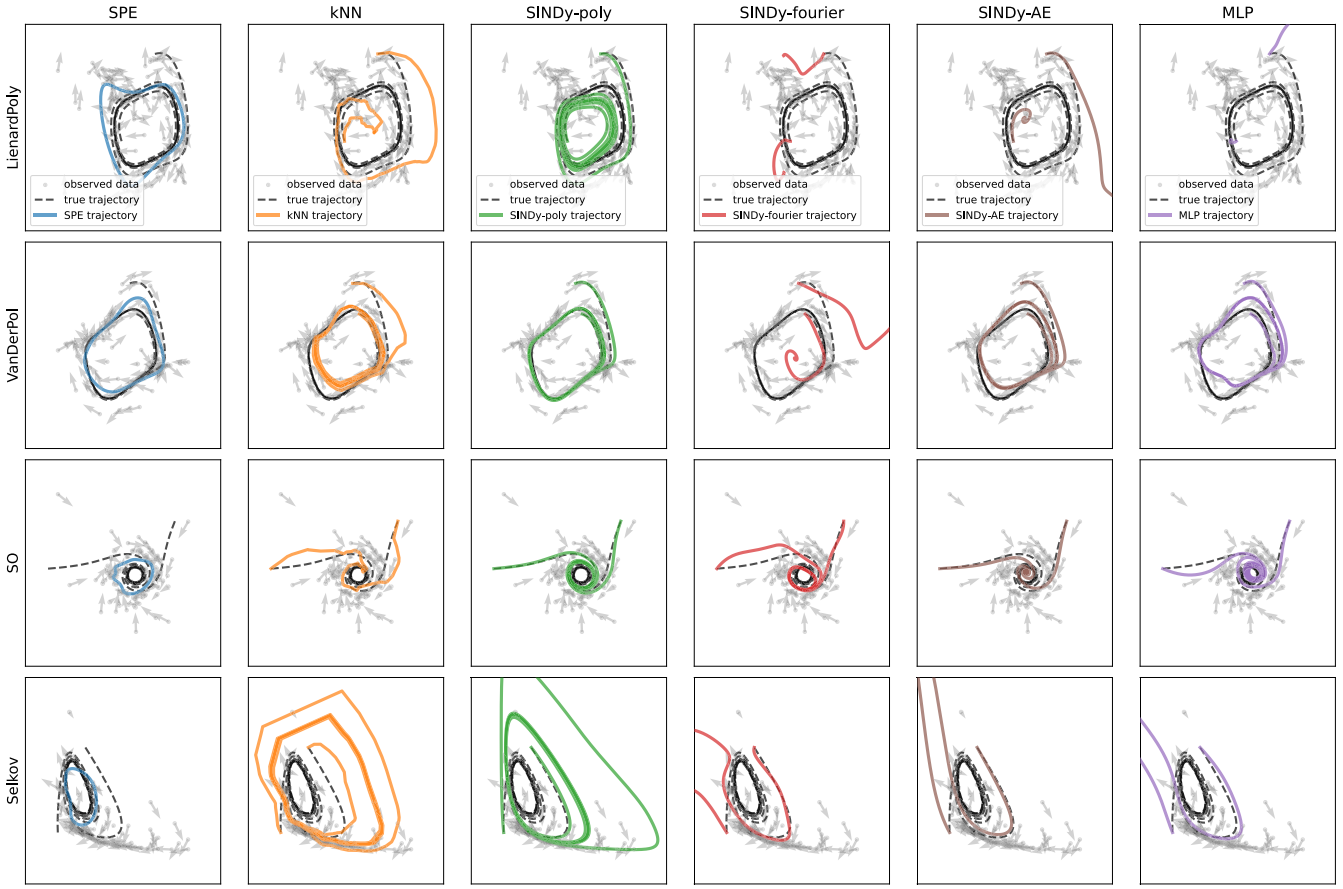


FIG. 13. Examples of baseline performance on synthetic systems with signal-to-noise ratio (SNR) of 2.5 and $N = 100$ data points. Dashed black lines depict ground-truth trajectories simulated according to the true governing equations which were used to generate the data, represented by the gray vectors. The solid colored lines are trajectories predicted by the different models, arranged into columns (left to right): SPE, k-nearest neighbor (kNN), SINDy with a library of polynomial functions up to degree 3 (SINDy-poly), SINDy with a library of 10 Fourier features (SINDy-Fourier), SINDy-AutoEncoders with a library of polynomial functions up to degree 3 (SINDy-AE), and a multilayer perceptron (MLP). SPE's prediction for the limit cycle is attained by transforming the position of the limit cycle in prototype space back to data space. For all other models, where this information cannot be directly accessed, long trajectories ($T = 60$) were simulated according to the model in order to visualize the captured invariant sets.

families of systems with order parameters corresponding to either a node or a periodic attractor, as described in Appendix B 1. Accordingly, the task is to classify each system in the dataset as either a point or periodic attractor. Systems in the dataset are comprised of a set of positions $x_i \in \mathbb{R}^2$ and the velocities corresponding to those positions, $\dot{x}_i \in \mathbb{R}^2$, organized on a dense, regular 64×64 grid for a total of $N = 4096$ observations. Table IV shows the average accuracy of SPE for each family of systems, compared with existing baselines. SPE outperforms existing baselines across all families of systems except for the family of simple oscillators (SO) and Augmented SO, where TWA, which was trained on thousands of these systems using a dense grid of points in phase space, exhibits higher accuracy (Table IV).

Note that, unlike most of the available baselines, SPE is not restricted to systems defined on a regular grid and can directly localize the invariant set (as shown in Sec. III A).

5. Evaluation in sparse classification

To evaluate the classification performance of SPE on sparse and noisy data, we simulated systems in the same manner as in Appendix C 2. We randomly sampled 1000 systems from the SO, Liénard Polynomial, Liénard Sigmoid, Van der Pol, BZ Reaction, and Sel'kov families of systems. For each system, the values of the order parameters were uniformly sampled from their corresponding ranges (see Appendix B 1), which correspond to either node attractors or cyclic attractors. For each system, N positions x_i and velocities $\dot{x}_i$ were sampled according to the procedure described in Appendix B 2, with $T_{\max} = 3$. Additive Gaussian noise was then added to these velocities with a standard deviation of $\sigma = \mu/\mathrm{SNR}$ where $\mu = \frac{1}{N}\sum_i \|\dot{x}_i\|$ and a prespecified signal-to-noise ratio (SNR). The task was then to classify the set of pairs $\{(x_i, \dot{x}_i)\}_{i=1}^N$ either as a node attractor or a limit cycle.

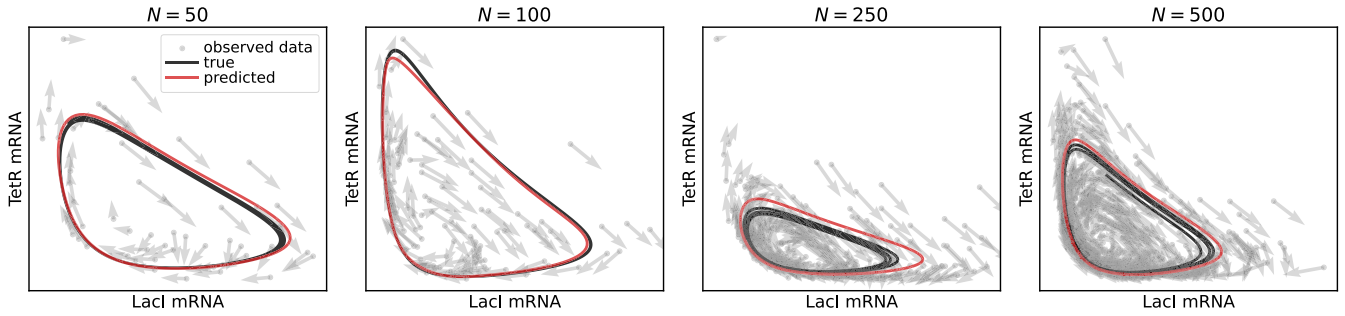


FIG. 14. The effects of the number of samples when fitting the repressilator system to a limit cycle with SPE.

APPENDIX D: IDENTIFICATION OF THE CELL CYCLE TRAJECTORY FROM SINGLE-CELL RNA-SEQUENCING AND RNA VELOCITY DATA OF PROLIFERATING CELLS

For the scRNA-seq data (Sec. III D), we use two datasets of proliferating cell lines: U2OS cell line data collected in Ref. [10] and human fibroblast cells from Ref. [2]. In both cases, the data are predominantly comprised of cells that are proliferating. The U2OS cell line data includes 1152 cells characterized by the expression of 58 884 genes, whereas the human fibroblast data contains 3086 cells and 10 789 genes. For preprocessing, we followed the protocol in Ref. [63], which involved gene and cell filtering, normalization, log transformation, and the selection of highly variable genes. We also used labels provided in both studies to filter out cells that are not proliferating.

To infer local measures of cellular dynamics in the high-dimensional gene expression space, we used scVelo [25], which computes RNA velocity vectors [24] from spliced and unspliced gene expression counts of a cell population. scVelo models the transcriptional dynamics of each gene using a dynamical model of transcription, splicing, and degradation. It applies an Expectation-Maximization algorithm to estimate gene-specific kinetic parameters for each cell on smoothed versions of the spliced and unspliced counts, which are estimated from a k-nearest-neighbor graph over gene expression [25].

The gene expression data and the inferred velocities were projected into a lower-dimensional space using principal component analysis (PCA), of dimensionality 100 for the U2OS cell line and 10 for the human fibroblasts. In both cases, using PCA greatly reduces the computational load of our method, while retaining almost all variability in the data and allowing us to project back into gene expression space after fitting. The option to project back into gene expression space enables the projection of invariant sets found using SPE into the full gene expression space and to probe the specific role of each gene in trajectories on the invariant set. Streamlines of the resulting velocity field are visualized in Fig. 6, left. Independently of the RNA velocities, we computed S-phase and G2/M-phase scores per cell by aggregating the expression of marker genes for these phases, as defined by Ref. [51], using the function "score_genes_cell_cycle" from scVelo package [25]. The color in Fig. 6, left and middle, represents the ratio of the S-phase score (red) and the G2/M-phase score (blue).

We then applied our method to fit the RNA velocities, embedded in the PCA space, to a high-dimensional cyclic oscillator prototype, as defined in Eq. (4) of Sec. III C. This prototype oscillates in the first two dimensions and decays exponentially in the others. The fitting parameters were as follows: angular speed $\omega = 5$, radius $a = 1$, three blocks as defined in Appendix A, $K = 3$ frequencies per layer, weight decay of $\texttt{wd} = 10^{-3}$, projection regularization set to $\texttt{proj\_reg} = e^{-2}$, determinant regularization of 0, a learning rate of $\texttt{lr} = 5 \times 10^{-4}$, and 1000 iterations.

Given the fitted coordinates of the cycle attractor in PCA space, we inverted the PCA transformation to find the expression of the different genes along the fitted cycle. These predicted gene expressions were normalized for visual clarity to the range $[0, 1]$ by subtracting the minimal value and then dividing by the maximal value. In Fig. 6 (right), we plot the normalized gene expression of representative marker genes for both S-phase and G2/M-phase [51], showing distinct oscillatory patterns consistent with cell cycle progression.

APPENDIX E: LOCALIZING AN EARLY BIFURCATION IN THE DEVELOPING MOUSE EMBRYO

The early development of the mouse embryo is well studied and is characterized by two subsequent binary decisions [52,53]. First, the blastomere differentiates into the inner cell mass (ICM) and trophectoderm. Afterward, cells in the ICM differentiate either into epiblast (EPI) cells or into primitive endoderm cells (PrE). The epiblast cells later form the embryo, whereas the PrE become the supporting tissue. In this work, we focused on the binary choice from ICM to either EPI or PrE, which happens between embryonic days E3 and E4. To this end, we used the E3.5 and E4.5 samples of the scRNA-seq data from Ref. [19]. E3.5 consists of 698 cells and 204 E4.5 cells, with 20 897 genes captured for each, after gene and cell filtering. After normalization and log transformation, we kept only the top 7500 highly variable genes.

The tissues from E3.5 and E4.5 were sampled destructively, meaning there is no access to direct dynamics. Moreover, the transition in question is on the scale of days, longer than the lifetime of RNAs, rendering RNA velocity impractical for modeling the relevant dynamics. Instead, in order to infer the dynamics, we find pairings between points in E3.5 and E4.5 using optimal transport (OT [56,57,74,75]) based on a Euclidean cost between the top 10 principal components (PCs) of the cells. This method, originally employed by Ref. [9], results in noisy estimates of the true underlying dynamics of the data.

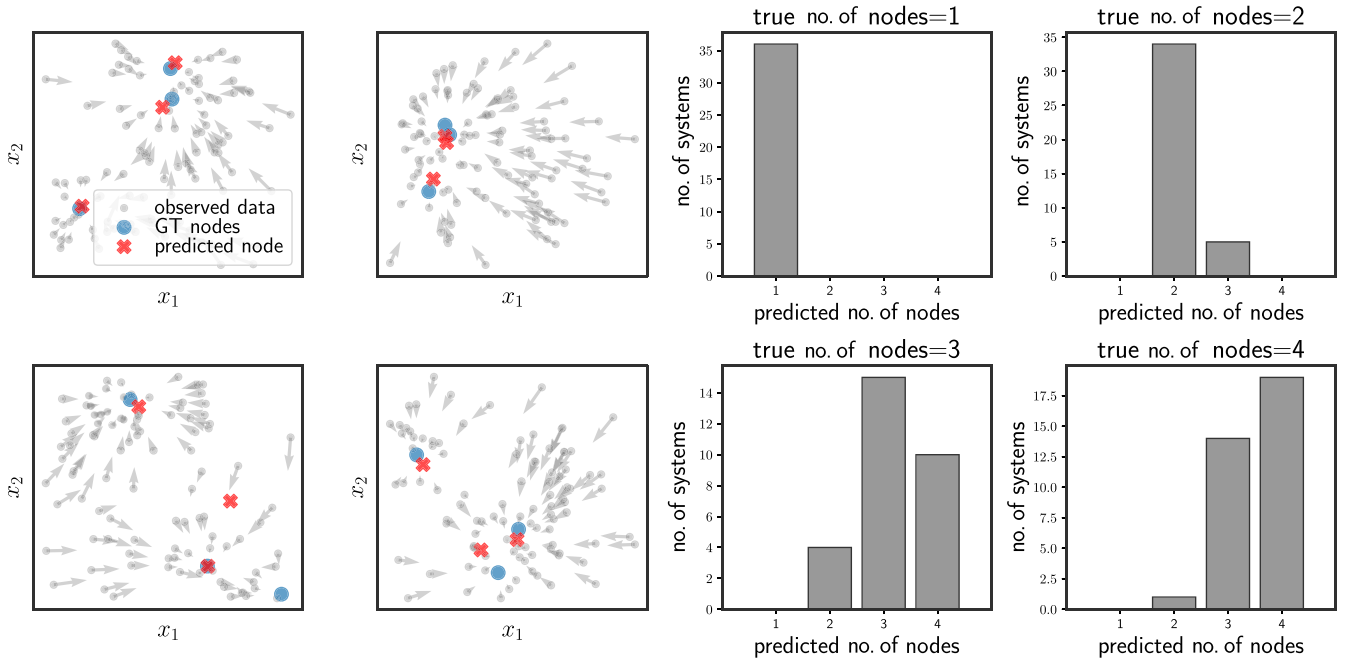


FIG. 15. Classifying and localizing invariant sets in systems with multiple basins of attraction. (Left) Four examples of fitting SPE to a system with multiple basins of attraction in three dimensions. In each example, the positions of the node attractors were randomly sampled (these positions are depicted by the blue circles in the plot), and 100 position-velocity pairs were then sampled using the scheme described in Appendix B 2. SPE was then used to match these data to a fixed prototype with three basins of attraction. The estimated positions of the attractors according to SPE are depicted by the red crosses. We find that SPE frequently matches the positions of the nodes. (Right) results in the classification of the number of nodes. Each bar plot shows how SPE categorizes systems with a set number of nodes, for instance, the top left plot shows the performance only on systems that had one node attractor. In each of these cases, the main mode predicted by SPE aligns with the true number of nodes in the system.

The prototype we used is based on the heteroclinic flip (HF) potential function from [52,53], defined as

$$\psi_{\mathrm{HF}}(x_1, x_2) = x_1^4 + x_2^4 - x_2^3 + 2x_1^2 x_2 - x_2^2 + \kappa x_2, \tag{E1}$$

where $\kappa = 1.4$. We fit this prototype to the data using SPE with the following parameters: two blocks as defined in Appendix A, $K = 4$ frequencies per layer, weight decay of $\texttt{wd} = 10^{-3}$, determinant regularization of $\lambda_{\mathrm{det}} = 10^{-3}$, a learning rate of $\texttt{lr} = 5 \times 10^{-4}$, and 1000 iterations. We additionally added a potential-based regularization on all points, defined as

$$L_{\mathrm{poten}} = \frac{1}{N} \sum_{i=1}^{N} \psi_{\mathrm{HF}}(x_i), \tag{E2}$$

with a regularization coefficient $\lambda_{\mathrm{poten}} = 10^{-3}$. This extra potential regularization helps ensure that, on average, the positions x_i are mapped to regions of prototype space that have low energy. This regularization is based on the assumption that in potential-based dynamics, the invariant set is localized to regions of space with low potential energy.

APPENDIX F: LOCALIZING DIFFERENTIATION FATES IN HEMATOPOIESIS LINEAGE TRACING DATA

To predict differentiation fates in hematopoiesis, we used the data from Ref. [20], where clonal hematopoietic progenitor cells are barcoded and collected at different time points after plating. At day 2, 4585 cells were collected, 14 962 at day 4, and 29 569 at day 6, after the preprocessing steps of Ref. [21]. In our analysis, we kept only the 7500 top highly variable genes of the 25 289 in the data.

Because clones share the same barcode, it is possible to "follow" the differentiation process through time. Velocities in these data were inferred as the average direction from cells at time t to all of the matching clones at time $t + 1$, as schematically described in Fig. 8 (top middle). This results in estimated velocities for the 19 547 cells collected at 2 and 4 days, which flow from undifferentiated states to various differentiated cell types (Fig. 8, top right). In this work, we used the same labeling of cell fates as Ref. [20], which label cells according to a panel of marker genes and clustering. The marker genes in Fig. 8 (bottom left) are the genes used by Ref. [20] to cluster the different blood cell types.

With the inferred dynamics, we chose each of the cell types that make up more than 1% of the total data. Both gene expression and velocities were projected using PCA to the first 50 PCs. We then fit each cluster to a node attractor prototype with SPE, whose dynamics are given by

$$\dot{y}_i = -e^{\delta} y_i, \tag{F1}$$

with $\delta = \frac{1}{2}$. The fitting parameters for SPE were 2 blocks, $K = 4$ frequencies, a learning rate of 10^{-3}, a weight decay of 10^{-3}, determinant regularization of 10^{-3}, and 1000 iterations.

APPENDIX G: FURTHER EXPERIMENTS

1. More results on the repressilator system

SPE is effective when data are sparse and higher dimensional. To demonstrate this, we consider the repressilator system as in Sec. III C. To test this, we sampled data from the repressilator system with different numbers of observations, $N = [50, 100, 250, 500]$, sampled as described in Appendix B. Applying SPE with a limit cycle prototype to these data, it accurately localizes the limit cycle across the different sparsities of the data, as qualitatively shown in Fig. 14. Quantitatively, the Wasserstein-2 (W_2) distance between long trajectories of the data under the true system and SPE's prediction (as described in Appendix C 2) remains low across all sparsities: $N = 50, 100, 250, 500$ for $W_2 = 1.5 \pm 0.5,\ 1.4 \pm 0.3,\ 1.7 \pm 0.4,\ 1.8 \pm 0.3$, respectively.

These W_2 distances were calculated over 25 randomly sampled systems. Reported numbers are average distances and standard error of the mean.

2. Multiple basins of attraction

While this work mostly focused on prototypes with a single basin of attraction, such as a node attractor or a limit cycle, SPE can also be used given multiple basins of attraction.

As a simple example, we studied synthetic dynamics with multiple fixed points in a high-dimensional space. The governing equations for these dynamics were based on a mixture-of-experts decomposition of space, where each expert represents a single basin of attraction. Specifically, the governing equations of the dynamics were defined as

$$f(x) = \sum_{k=1}^{K} r_k(x)\varphi_k(x - c_k), \tag{G1}$$

$$r_k(x) = \frac{\exp\left[-\frac{1}{2s^2}\|x - c_k\|^2\right]}{\sum_{k'} \exp\left[-\frac{1}{2s^2}\|x - c_{k'}\|^2\right]}, \tag{G2}$$

where $\varphi_k(\cdot)$ are the dynamics of expert k, whose coordinate system is centered around $c_k \in \mathbb{R}^d$, s determines the size of the region that the $k-$th expert models, and K is the number of basins of attraction. Through these, $f(x)$ models dynamics across the whole space, where for each expert $0 < r_k(x) < 1$ determines how strongly the dynamics at position x are controlled by expert k.

Synthetic data were generated by sampling systems with centers sampled uniformly from the range $[-0.9, 0.9]^d$, all of which were node attractors:

$$\varphi_k(y) = -y. \tag{G3}$$

For each generated system, the number of attractors K was chosen between 1 and 4. From these generated systems, we followed the sampling procedure from Appendix B 2 to create 100 pairs of positions and velocities. In our experiments, we considered five-dimensional data, meaning that each node c_k is determined by a five-dimensional vector.

We then fit SPE to each system generated this way, with prototypes of the same form—4 prototypes, one for each possible number of centers. This allowed us, in turn, to classify the number of basins of attraction in the observed data. The positions of the nodes c_k in the prototypes were fixed and distinct from the positions of the nodes in the true underlying system, so that the fitting process decomposes into the prediction of both the number of nodes as well as their location. Results can be seen in Fig. 15.

For the experiments shown in Fig. 15, we used INNs with two blocks of width 3, and five Householder transformations. The networks were trained for 1500 iterations, with a learning rate of $1e^{-3}$, weight decay of 0.1, with all other regularization coefficients equal to zero.

3. Mismatch between prototype and underlying system

SPE uses a soft notion of equivalence to match between the prototypes and the data. Thus, even when there is a mismatch between the prototype and the underlying system, SPE converges to a solution that transforms the dynamics of the chosen prototype to resemble the observed dynamics. Because of this, we expect SPE to find a mapping that retains certain characteristics of the dynamics that are shared between the data and the prototype, even in cases where the two systems are not fully smoothly equivalent. We tested this through the analysis of simulated oscillatory systems from Appendix B 1. Systems representing node attractors and limit cycles, while not equivalent, can still have shared attributes. One example is the orientation of the flow–either clockwise or counterclockwise—around the basin of attraction. To emulate a mismatch between the input prototype and the underlying dynamics, we fit SPE using only node attractor prototypes, as opposed to the true underlying oscillatory behavior, using the same prototypes from Eq. (3), with $a = -0.25$ and $\omega = \pm 0.5$. Figure 16 demonstrates that even when there is a mismatch between the underlying dynamics and the prototype, SPE is able to match the orientation of the flow. Specifically, if the system exhibits a clockwise limit cycle, SPE fits a node attractor spiraling clockwise, corresponding to a negative value for the angular velocity parameter ω of the prototype, and vice versa. Quantitatively, SPE achieves 98% accuracy in predicting the flow orientation, ω, of the SO family of systems, averaged over 100 trials using the same hyperparameters described in Appendix A 7.

4. Alternative limit cycle prototypes

Throughout the main text, the form of the prototypes we used were those of the simple oscillator (SO):

$$\dot{r} = r(a - r^2), \quad \dot{\theta} = \omega, \tag{G4}$$

where r and θ are the polar coordinates. We focused on the SO as it can be fully characterized analytically and exhibits a well-studied form of a supercritical Hopf bifurcation, transitioning from a fixed point to a limit cycle attractor. However, SPE shows robust performance of invariant set localization over observations from other forms of dynamical systems exhibiting limit cycle attractors because these systems are smoothly equivalent to each other.

This is demonstrated over two alternative prototypes, representing dynamical systems of other oscillator forms:

$$\dot{r} = a - r^2, \quad \dot{\theta} = \omega, \tag{G5}$$

$$\dot{r} = \sqrt{a} - r, \quad \dot{\theta} = \omega, \tag{G6}$$

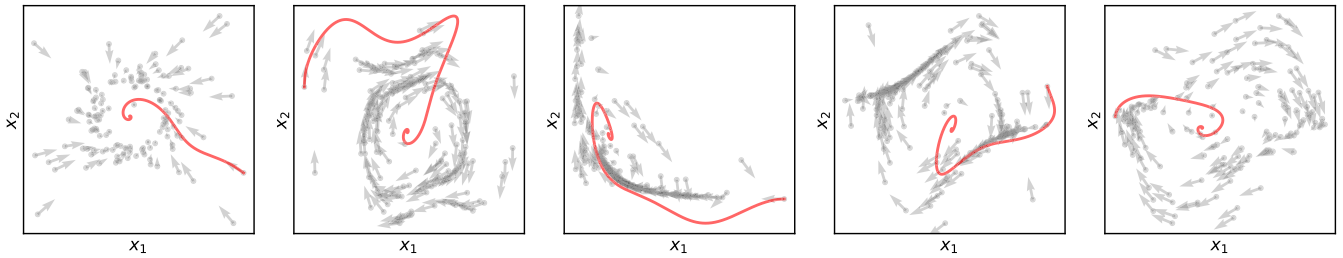


FIG. 16. Direction of angular velocity is captured, even when there is a mismatch between the prototype and the true underlying system. Randomly sampled systems and examples of fitting SPE to an inherently oscillatory process using a node attractor prototype. Despite this mismatch between underlying behavior and prototype, SPE is still able to characterize the dominant behavior of the system, mainly the direction of the angular velocity and the focal point of the limit cycle.

where r and θ are the polar coordinates. Here, as in the SO prototype, the parameter a represents the radius of the limit cycle and ω represents the angular velocity. These systems exhibit different speeds in the radial direction toward the limit cycle, but otherwise remain qualitatively similar. Using SPE for invariant set localization with these alternative prototypes Eq. (G 4), instead of the SO prototype [Eq. (G4)], for the localization of limit cycles in simulated 2D data (systems from Table III) results in similar estimated invariant sets (Fig. 17).

5. Effects of normalization on the loss

The equivalence loss used by SPE [Eq. (2)], L_{E}, is given by

$$L_{\mathrm{E}}(H_\theta, g) = \frac{1}{N}\sum_{i=1}^{N}\left\| \frac{\partial_{x_i} H_\theta(x_i)\dot{x}_i}{\left\|\partial_{x_i} H_\theta(x_i)\dot{x}_i\right\|} - \frac{g(H_\theta(x_i))}{\|g(H_\theta(x_i))\|}\right\|^2, \quad \text{(G7)}$$

which normalizes both the observed and prototype velocities to enforce smooth orbital equivalence. While effective, this loss does not penalize discrepancies in the magnitude of the predicted velocities. Other normalization strategies, which penalize differences in velocity magnitude, include ordinary least squares, L_{OLS}, or a shared normalization between the two terms, L_{shared},

$$L_{\mathrm{OLS}}(H_\theta, g) = \frac{1}{N}\sum_{i=1}^{N}\left\|\partial_{x_i} H_\theta(x_i)\dot{x}_i - g(H_\theta(x_i))\right\|^2, \quad \text{(G8)}$$

$$L_{\mathrm{shared}}(H_\theta, g) = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\partial_{x_i} H_\theta(x_i)\dot{x}_i - g(H_\theta(x_i))\right\|^2}{\left\|\partial_{x_i} H_\theta(x_i)\dot{x}_i\right\| \cdot \|g(H_\theta(x_i))\|}. \quad \text{(G9)}$$

We qualitatively compare the three options in Fig. 18, keeping all other hyperparameters fixed. We found that the

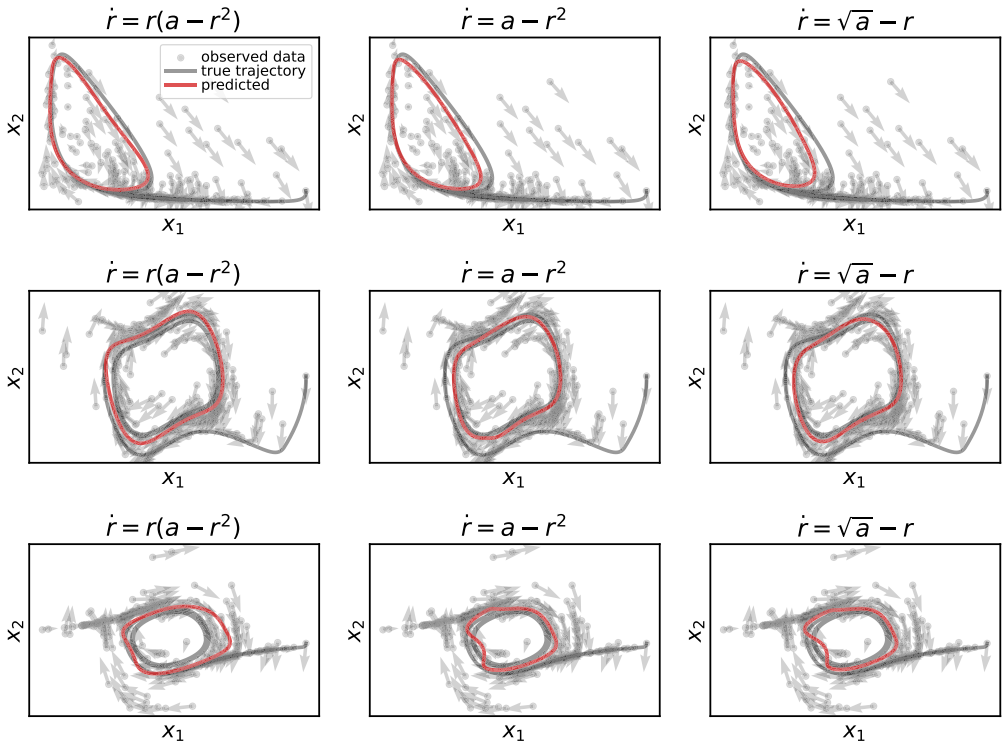


FIG. 17. Estimation of limit cycles using different oscillator prototypes. The three rows correspond to different possible cycle prototypes. The leftmost column corresponds to the prototypes used in the rest of this work, and the middle and rightmost columns have decreasing speeds toward the limit cycle in the radial direction. The three prototypes result in similar estimated invariant sets.

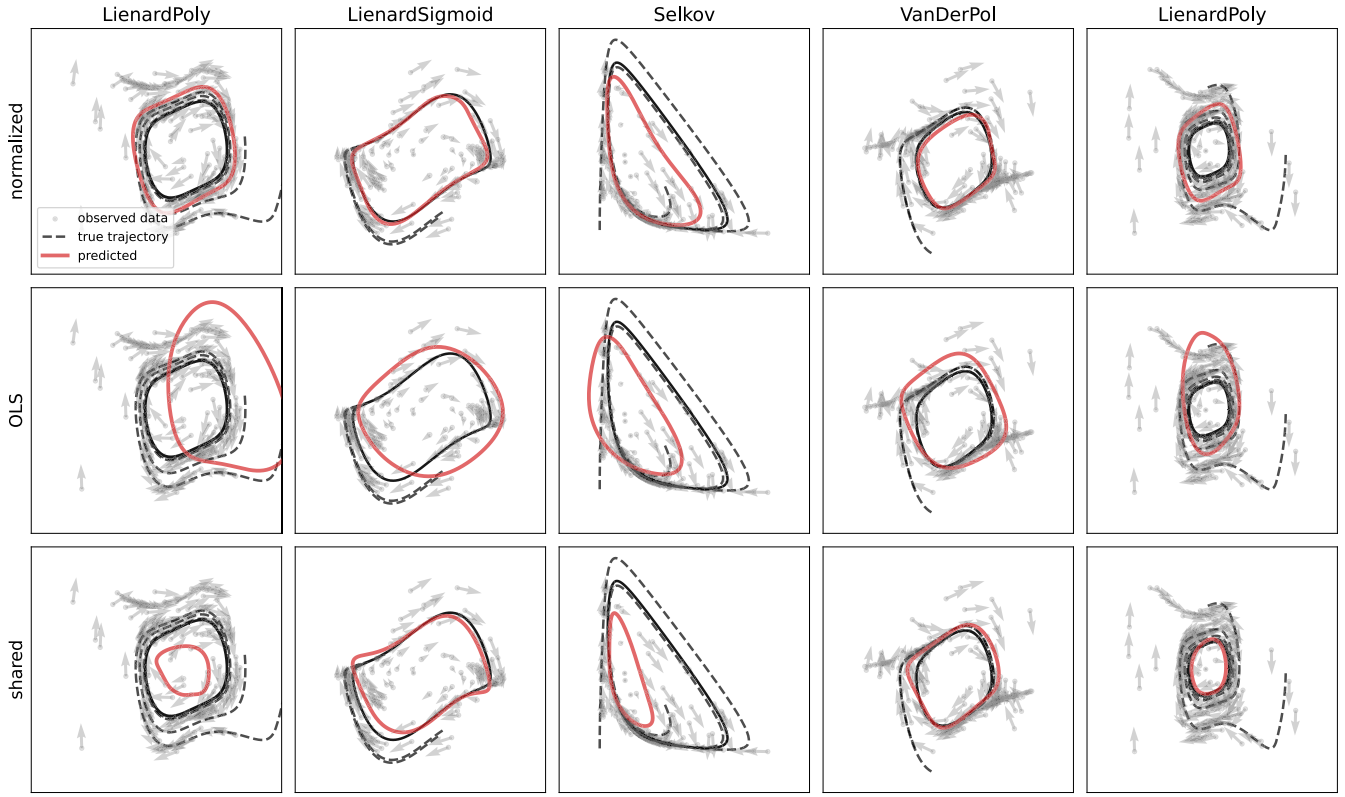


FIG. 18. Comparison of normalization strategies in the SPE loss on the predicted invariant set. The three rows correspond to different possible normalization strategies: the top row corresponds to SPE base loss from Eq. (2) $L_{\mathrm{E}}(H_\theta, g)$, the middle row corresponds to an ordinary least-squares (OLS) loss that involves no normalization $L_{\mathrm{OLS}}(H_\theta, g)$, and the bottom row corresponds to a loss that shares the normalization between the predicted and observed velocities $L_{\mathrm{shared}}(H_\theta, g)$.

normalized equivalence loss, L_{E}, tends to lead to faster and more faithful convergence to the true invariant set, relative to the results achieved via the other two normalization strategies.

[1] D. Schwabe, S. Formichetti, J. P. Junker, M. Falcke, and N. Rajewsky, The transcriptome dynamics of single cells during the cell cycle, Mol. Syst. Biol. **16**, e9946 (2020).

[2] A. Riba, A. Oravecz, M. Durik, S. Jiménez, V. Alunni, M. Cerciat, M. Jung, C. Keime, W. M. Keyes, and N. Molina, Cell cycle gene regulation dynamics revealed by RNA velocity and deep-learning, Nat. Commun. **13**, 2865 (2022).

[3] J. Karin, Y. Bornfeld, and M. Nitzan, scPrisma infers, filters and enhances topological signals in single-cell data using spectral template matching, Nat. Biotechnol. **41**, 1645 (2023).

[4] A. R. Lederer, M. Leonardi, L. Talamanca, D. M. Bobrovskiy, A. Herrera, C. Droin, I. Khven, H. J. F. Carvalho, A. Valente, A. D. Mantes, *et al.*, Statistical inference with a manifold-constrained RNA velocity model uncovers cell cycle speed modulations, Nat. Methods **21**, 2271 (2024).

[5] C. Dibner, U. Schibler, and U. Albrecht, The mammalian circadian timing system: Organization and coordination of central and peripheral clocks, Annu. Rev. Physiol. **72**, 517 (2010).

[6] M. Sáez, J. Briscoe, and D. A. Rand, Dynamical landscapes of cell fate decisions, Interface focus **12**, 20220002 (2022).

[7] M. Lange, V. Bergen, M. Klein, M. Setty, B. Reuter, M. Bakhti, H. Lickert, M. Ansari, J. Schniering, H. B. Schiller, *et al.*, CellRank for directed single-cell fate mapping, Nat. Methods **19**, 159 (2022).

[8] S. Farrell, M. Mani, and S. Goyal, Inferring single-cell transcriptomic dynamics with structured latent gene expression dynamics, Cell Rep. Methods **3**, 100581 (2023).

[9] G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, *et al.*, Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming, Cell **176**, 928 (2019).

[10] D. Mahdessian, A. J. Cesnik, C. Gnann, F. Danielsson, L. Stenström, M. Arif, C. Zhang, T. Le, F. Johansson, R. Schutten, A. Bäckström, U. Axelsson, P. Thul, N. H. Cho, O. Carja, M. Uhlén, A. Mardinoglu, C. Stadler, C. Lindskog, B. Ayoglu, *et al.*, Spatiotemporal dissection of the cell cycle with single-cell proteogenomics, Nature (London) **590**, 649 (2021).

[11] S. C. Zheng, G. Stein-O'Brien, J. J. Augustin, J. Slosberg, G. A. Carosso, B. Winer, G. Shin, H. T. Bjornsson, L. A. Goff, and

K. D. Hansen, Universal prediction of cell-cycle position using transfer learning, Genome Biol. **23**, 41 (2022).

[12] K. M. Kedziora and W. Stallaert, Cell cycle mapping using multiplexed immunofluorescence, Methods Mol. Biol. **2740**, 243 (2024).

[13] A. Zinovyev, M. Sadovsky, L. Calzone, A. Fouché, C. S Groeneveld, A. Chervov, E. Barillot, and A. N Gorban, Modeling progression of single cell populations through the cell cycle as a sequence of switches, Front. Mol. Biosci. **8** 793912 (2022).

[14] W. Stallaert, K. M. Kedziora, C. D. Taylor, T. M. Zikry, J. S. Ranek, H. K. Sobon, S. R. Taylor, C. L. Young, J. G. Cook, and J. E. Purvis, The structure of the human cell cycle, Cell Sys. **13**, 230 (2022).

[15] R. Qu, X. Cheng, E. Sefik, J. S. Stanley Iii, B. Landa, F. Strino, S. Platt, J. Garritano, I. D Odell, R. Coifman, *et al.*, Gene trajectory inference for single-cell data by optimal transport metrics, Nat. Biotechnol. **43**, 258 (2025).

[16] K. Maehara and Y. Ohkawa, Geometry-preserving vector field reconstruction of high-dimensional cell-state dynamics using ddHodge, Nat. Commun. **16**, 11342 (2025).

[17] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, *et al.*, Eleven grand challenges in single-cell data science, Genome Biol. **21**, 31 (2020).

[18] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, The technology and biology of single-cell RNA sequencing, Mol. Cell **58**, 610 (2015).

[19] S. Nowotschin, M. Setty, Y.-Y. Kuo, V. Liu, V. Garg, R. Sharma, C. S. Simon, N. Saiz, R. Gardner, S. C. Boutet, *et al.*, The emergent landscape of the mouse gut endoderm at single-cell resolution, Nature (London) **569**, 361 (2019).

[20] C. Weinreb, A. Rodriguez-Fraticelli, F. D. Camargo, and A. M. Klein, Lineage tracing on transcriptional landscapes links state to fate during differentiation, Science **367**, eaaw3381 (2020).

[21] S.-W. Wang, M. J. Herriges, K. Hurley, D. N. Kotton, and A. M. Klein, CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information, Nat. Biotechnol. **40**, 1066 (2022).

[22] N. Battich, J. Beumer, B. De Barbanson, L. Krenning, C. S. Baron, M. E. Tanenbaum, H. Clevers, and A. Van Oudenaarden, Sequencing metabolically labeled transcripts in single cells reveals mRNA turnover strategies, Science **367**, 1151 (2020).

[23] X. Qiu, Y. Zhang, J. D. Martin-Rufino, C. Weng, S. Hosseinzadeh, D. Yang, A. N. Pogson, M. Y. Hein, K. Hoi, J. Min, L. Wang, *et al.*, Mapping transcriptomic vector fields of single cells, Cell **185**, 690 (2022).

[24] G. L. Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, *et al.*, RNA velocity of single cells, Nature (London) **560**, 494 (2018).

[25] V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis, Generalizing RNA velocity to transient cell states through dynamical modeling, Nat. Biotechnol. **38**, 1408 (2020).

[26] W. Chen, O. Guillaume-Gentil, P. Y. de Rainer, C. G. Gäbelein, W. Saelens, V. Gardeux, A. Klaeger, R. Dainese, M. Zachara, T. Zambelli, *et al.*, Live-seq enables temporal transcriptomic recording of single cells, Nature (London) **608**, 733 (2022).

[27] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Neural ordinary differential equations, Adv. Neural Inf. Proc. Syst. **31**, 1 (2018).

[28] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA **113**, 3932 (2016).

[29] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, Machine learning–accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. **118**, e2101784118 (2021).

[30] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, Chaos as an intermittently forced linear system, Nat. Commun. **8**, 19 (2017).

[31] D. Hilbert, M. W. Newsom, F. E. Browder, D. A. Martin, G. Kreisel, and M. Davis, Mathematical problems. Lecture delivered before the International Congress of Mathematicians at Paris in 1900, J. Symbolic Logic **44**, 1 (1979).

[32] S. Smale, Mathematical problems for the next century, The Mathematical Intelligencer **20**, 7 (1998).

[33] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer Science & Business Media, New York, 2013), Vol. 42.

[34] N. Moriel, M. Ricci, and M. Nitzan, Let's do the time-warp-attend: Learning topological invariants of dynamical systems, in *Proceedings of the International Conference on Learning Representations* (ICLR, 2024), Vol. 2024.

[35] R. Chen, G. Vedovati, T. Braver, and S. Ching, Dform: Diffeomorphic vector field alignment for assessing dynamics across learned models, arXiv:2402.09735.

[36] J. J. Bramburger, S. L. Brunton, and N. J. Kutz, Deep learning of conjugate mappings, Physica D **427**, 133008 (2021).

[37] W. T. Redman, M. Fonoberova, R. Mohr, I. G. Kevrekidis, and I. Mezić, Algorithmic (semi-)conjugacy via Koopman operator theory, in *Proceedings of the 2022 IEEE 61st Conference on Decision and Control (CDC)* (IEEE, Piscataway, NJ, 2022), pp. 6006–6011.

[38] B. Glaz, Efficient pseudometrics for data-driven comparisons of nonlinear dynamical systems, Nonlinear Dynamics **113**, 12465 (2025).

[39] M. Ostrow, A. Eisen, L. Kozachkov, and I. Fiete, Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis, Adv. Neural Inf. Process. Syst. **36**, 33824 (2023).

[40] N. Romeo, C. Chi, A. R. Dinner, and E. R. Jerison, Characterizing nonlinear dynamics by contrastive cartography, PRX Life **3**, 033022 (2025).

[41] J. D. Skufca and E. M. Bollt, Relaxing conjugacy to fit modeling in dynamical systems, Phys. Rev. E **76**, 026220 (2007).

[42] J. D. Skufca and E. M. Bollt, A concept of homeomorphic defect for defining mostly conjugate dynamical systems, Chaos **18**, 013118 (2008).

[43] F. Takens, Detecting strange attractors in turbulence, in *Dynamical Systems and Turbulence, Warwick (1980)*, edited by D. A. Rand and L.-S. Young, Lecture Notes in Mathematics (Springer-Verlag, Berlin, Heidelberg, 1981), Vol. 898, pp. 366–381.

[44] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, Modern Koopman theory for dynamical systems, SIAM Review **64**, 229 (2022).

[45] M. B. Elowitz and S. Leibler, A synthetic oscillatory network of transcriptional regulators, Nature (London) **403**, 335 (2000).
[46] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, J. Mach. Learn. Res. **22**, 1 (2021).
[47] L. Dinh, D. Krueger, and Y. Bengio, NICE: Non-linear independent components estimation, arXiv:1410.8516.
[48] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proc. Natl. Acad. Sci. USA **116**, 22445 (2019).
[49] A. Verdugo, Hopf bifurcation analysis of the repressilator model, Am. J. Comput. Math. **8**, 137 (2018).
[50] L. Potvin-Trottier, N. D. Lord, G. Vinnicombe, and J. Paulsson, Synchronous long-term oscillations in a synthetic gene circuit, Nature (London) **538**, 514 (2016).
[51] I. Tirosh, B. Izar, S. M. Prakadan, M. H. Wadsworth, D. Treacy, J. J. Trombetta, A. Rotem, C. Rodman, C. Lian, G. Murphy, *et al.*, Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq, Science **352**, 189 (2016).
[52] N. Schrode, P. Xenopoulos, A. Piliszek, S. Frankenberg, B. Plusa, and A.-K. Hadjantonakis, Anatomy of a blastocyst: Cell behaviors driving cell fate choice and morphogenesis in the early mouse embryo, genesis **51**, 219 (2013).
[53] A. Raju and E. D. Siggia, A geometrical model of cell fate specification in the mouse blastocyst, Development **151**, dev202467 (2024).
[54] A. Howe and M. Mani, Learning geometric models for developmental dynamics, Phys. Rev. X **15**, 031070 (2025).
[55] C. H. Waddington, *The Strategy of the Genes: A Discussion of Some Aspects of Theoretical Biology* (George Allen & Unwin, London, 1957).
[56] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. H. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, *et al.*, POT: Python optimal transport, J. Mach. Learn. Res. **22**, 1 (2021).
[57] R. Flamary, C. Vincent-Cuaz, N. Courty, A. Gramfort, O. Kachaiev, H. Q. Tran, L. David, C. Bonet, N. Cassereau, T. Gnassounou, E. Tanguy, J. Delon, A. Collas, S. Mazelet, L. Chapel, T. Kerdoncuff, X. Yu, M. Feickert, P. Krzakala, T. Liu, *et al.*, POT Python optimal transport (version 0.9.5), 2024, https://github.com/PythonOT/POT.
[58] N. Schrode, N. Saiz, S. Di Talia, and A.-K. Hadjantonakis, GATA6 levels modulate primitive endoderm cell fate choice and timing in the mouse blastocyst, Dev. Cell **29**, 454 (2014).
[59] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen, Invertible residual networks, in *International Conference on Machine Learning* (ICML, 2019), pp. 573–582.
[60] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, Adaptive mixtures of local experts, Neural Comput. **3**, 79 (1991).
[61] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows, Adv. Neural Inf. Proc. Syst. **32**, 1 (2019).
[62] M. Adler, A. Mayo, X. Zhou, R. A. Franklin, M. L. Meizlish, R. Medzhitov, S. M. Kallenberger, and U. Alon, Principles of cell circuits for tissue repair and fibrosis, IScience **23**, 100841 (2020).
[63] S. C. Zheng, G. Stein-O'Brien, L. Boukas, L. A. Goff, and K. D. Hansen, Pumping the brakes on RNA velocity by understanding and interpreting RNA velocity estimates, Genome Biol. **24**, 246 (2023).
[64] G. Gorin, M. Fang, T. Chari, and L. Pachter, RNA velocity unraveled, PLoS Comput. Biol. **18**, e1010492 (2022).
[65] https://github.com/nitzanlab/prototype-equivalences.
[66] J. M. Tomczak and M. Welling, Improving variational auto-encoders using Householder flow, arXiv:1611.09630.
[67] C. M. Bishop, M. Svensén, and C. K. I. Williams, GTM: The generative topographic mapping, Neural Comput. **10**, 215 (1998).
[68] D. P. Kingma, Adam: A method for stochastic optimization, arXiv:1412.6980.
[69] J. Helman and L. Hesselink, Representation and display of vector field topology in fluid flow data sets, Computer **22**, 27 (1989).
[70] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang, Vector field editing and periodic orbit extraction using Morse decomposition, IEEE Trans. Visualization Comput. Graph. **13**, 769 (2007).
[71] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, PySINDy: A Python package for the sparse identification of nonlinear dynamics from data, arXiv:2004.08424.
[72] D. Hendrycks and K. Gimpel, Gaussian error linear units (GELUs), arXiv:1606.08415.
[73] M. Ricci, N. Moriel, Z. Piran, and M. Nitzan, Phase2vec: Dynamical systems embedding with a physics-informed convolutional network, in *Proceedings of the 11th International Conference on Learning Representations* (ICLR, Kigali, 2023).
[74] C. Villani, *Topics in Optimal Transportation* (American Mathematical Soc., Providence, RI, 2021), Vol. 58.
[75] P. Gabriel and C. Marco, Computational optimal transport with applications to data sciences, Found. Trends Mach. Learn. **11**, 355 (2019).