

CHARACTERIZING NONLINEAR DYNAMICS VIA SMOOTH PROTOTYPE EQUIVALENCES

005 **Anonymous authors**

006 Paper under double-blind review

ABSTRACT

011 Characterizing the long term behavior of dynamical systems given limited measure-
 012 ments is a common challenge throughout the physical and biological sciences. This
 013 is a challenging task due to transient variability in systems with equivalent long-
 014 term dynamics. We address this by introducing smooth prototype equivalences
 015 (SPE), a framework for matching sparse observations to prototypical behaviors
 016 using invertible neural networks which model smooth phase space deformations.
 017 SPE can localize the invariant sets describing long-term behavior of the observed
 018 dynamics through the learned mapping from prototype space to data space. Fur-
 019 thermore, SPE enables classification of dynamical regimes by comparing the data
 020 residual of the deformed measurements to prototype dynamics. Our method out-
 021 performs existing techniques in the classification of oscillatory systems and can
 022 efficiently identify invariant structures like limit cycles and fixed points in an
 023 equation-free manner, even when only a small, noisy subset of the phase space
 024 is observed. Finally, we show how our method can be used for the detection of
 025 biological processes like the cell cycle trajectory from high-dimensional single-cell
 026 gene expression data.

1 INTRODUCTION

030 Predicting the long-term behavior of dynamical systems from sparse data is a challenging problem in
 031 the physical and life sciences (Weisshaar, 2012; Dimitriadis, 2017; Chervov & Zinovyev, 2022). This
 032 situation is further exacerbated in realistic cases where equations are unknown and behavior must be
 033 estimated from sparse, noisy or high-dimensional data. To this end, substantial work has been devoted
 034 to accurately estimating the vector fields underlying observed sequential data in the form of ordinary
 035 differential equations (ODEs) (e.g Chen et al. 2018; Brunton et al. 2016; Farrell et al. 2023; Kochkov
 036 et al. 2021; Brunton et al. 2017). However, even when there is access to the full governing equations
 037 of the underlying ODE, understanding the overall behavior of the dynamical system is notoriously
 038 difficult (Smale, 2000). When only sparse, noisy observations from the vector field are available,
 039 even just classifying between a limit cycle or a node attractor in two dimensions becomes a difficult
 040 problem (Moriel et al., 2023). Detecting these long-term behaviors can be crucial across different
 041 fields; for example, in biology, the cell cycle is often modeled as a limit cycle in gene expression
 042 space (Riba et al., 2022; Schwabe et al., 2020; Karin et al., 2023), pinpointing the genes participating
 043 in and timings of this fundamental process is the subject of substantial work in computational biology
 044 (e.g. Mahdessian et al. 2021; Zheng et al. 2022; Kedziora & Stallaert 2024; Zinovyev et al. 2021;
 045 Stallaert et al. 2021). In many such scenarios, it can be useful to leverage domain expertise about
 046 the overall dynamical behavior expected in the observed data, like the existence of a limit cycle in
 047 the cell cycle example above. If such domain expertise is used, then the search space for invariant
 048 structures can be greatly simplified, from all possible structures to only a few which are plausible in
 the system considered.

049 We propose a method for determining the type and location of invariant sets from vector field data,
 050 subject to constraints introduced by domain expertise. Specifically, we assume that there is a true
 051 (hidden) dynamical system governed by an ordinary differential equation (ODE) $\frac{dx}{dt} = f(x)$, and
 052 we only have access to sparse observations from this ODE. These observations come in the form of
 053 snapshots of the dynamics: pairs of positions x_i and noisy estimates of velocities in those positions,
 $\dot{x}_i = f(x_i) + \epsilon_i$, where ϵ_i is a noise vector. To extract information about the long-term behavior of

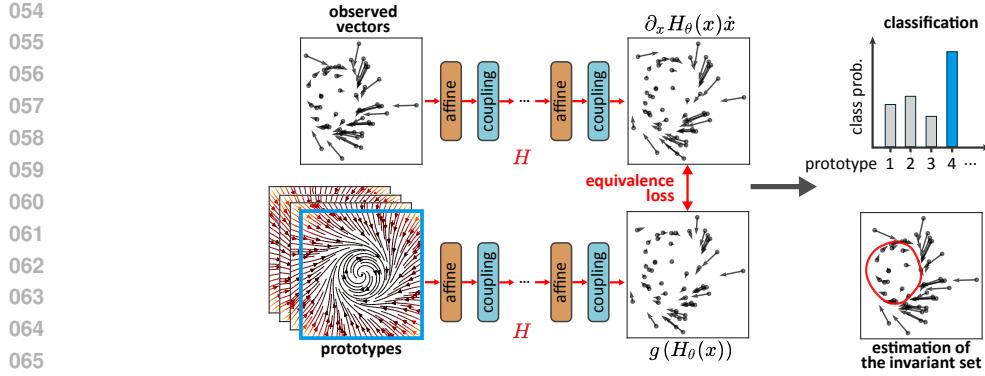


Figure 1: **Schematic for the process of fitting dynamical prototypes using smooth prototype equivalences (SPE).** The observed dynamics are compared to each of the chosen prototypes, which involves training an INN $H_\theta(x)$, for each prototype $g(y)$. After training each of the INNs, the observed dynamics are classified as the prototype with the smallest equivalence loss. Given a trained INN $H_\theta(x)$, we have access to a mapping between the data space and the prototype space, which allows us to estimate the long-term behavior, i.e. invariant set, of the dynamics underlying the observed data.

the underlying dynamics, we leverage the fact that the behavior of a dynamical system is not affected by a smooth, invertible change of coordinates. Two systems related by such an invertible mapping are called *smoothly equivalent*, and share the same invariant sets (Chen et al., 2024). The main intuition behind our approach, which we termed *smooth prototype equivalences* (SPE), is to learn an invertible mapping between the data and a *prototypical* version of the behavior which the data is assumed to follow, turning the problem into a form of prototype learning (Rosch, 1973; Bonilla & Robles-Kelly, 2012; Chen et al., 2023).

Our contributions include the following:

- We first demonstrate how SPE can localize limit cycles in a synthetic dataset of 2D dynamical systems drawn from across the physical and life sciences.
- We show how SPE can be used to classify between node and limit cycle attractors, highly abundant and biologically meaningful prototypes, in these synthetic systems, despite the absence of closed-form governing equations. Moreover, we benchmark SPE against previously proposed methods to classify dynamics on a dense grid. Furthermore, SPE, unlike previous methods, is robust to noisy and low-sample conditions, illustrating the compatibility of SPE to real data-driven scenarios.
- We then demonstrate that SPE can naturally extend to more dimensions by applying it to the synthetic six-dimensional repressilator gene regulatory network (Elowitz & Leibler, 2000).
- Finally, we illustrate how SPE can be used in a real scientific scenario, by deploying our method to localize cell-cycle dynamics from noisy, high-dimensional single-cell gene expression measurements, without relying on a gene expression reference atlas.

2 RELATED WORKS

Data driven approaches in dynamical equivalences. Early notions of equivalence of dynamical systems are due to Poincaré (1892–1899), who introduced foundational ideas like qualitative analysis of trajectories, periodic orbits, and invariant sets. Building on these findings, a large body of work was established to better understand when two systems can be called equivalent (Andronov & Pontryagin, 1937; Hartman, 1960; Sternberg, 1957). Subsequently, in recent years a growing body of work have leveraged relaxed notions of these equivalences (Skufca & Boltt, 2007; 2008) in order to model and compare between nonlinear dynamical systems (Chen et al., 2024; Bramburger et al., 2021; Redman et al., 2022; Glaz, 2024). These characterizations are attractive as they focus on the qualitative behaviors of dynamical systems, such as the stability and type of invariant set. These methodologies

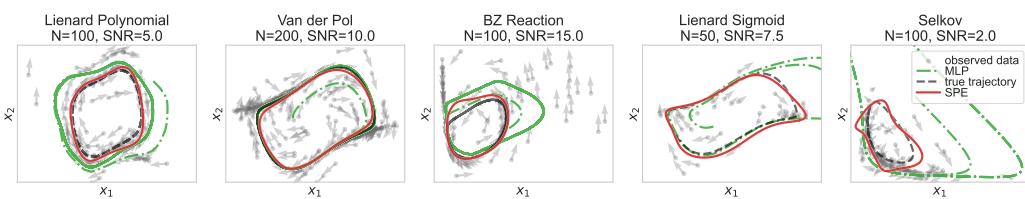
108 are especially relevant in the field of data-driven scientific discovery, which frequently seeks to
 109 compare between experimentally collected dynamical data and analytically derived systems. These
 110 approaches, however, typically require explicit access to the functional form of the dynamical systems,
 111 a limiting factor in their application to experimentally collected data.

112 For experimentally collected data, Ostrow et al. introduced DSA which utilizes a modified form of
 113 Procrustes analysis in order to align between the delay-embeddings of two observed time series. In
 114 this manner, under Taken’s embedding theorem (Takens, 1981), the two sequences will only be well
 115 aligned when they are conjugate and enough delays are used for the embedding. However, DSA
 116 requires sequences of observations long enough for the delay embedding to be beneficial, data which
 117 is not always available. Alternatively, Muriel et al. took an empirical approach to the categorization of
 118 the invariant set in an observed vector field by training a deep network to distinguish random samples
 119 from two conjugacy classes representing oscillatory and non-oscillatory dynamics. This enables
 120 classification of systems when the underlying governing equations are hidden, without relying on
 121 time series data, but is restricted to conjugacy classes defined during the training of the network and
 122 cannot provide more detailed insights beyond the classification of two-dimensional dynamics, such
 123 as those related to the structure of the invariant set, or extensions to higher dimensions.

124 Finally, data-driven approaches to Koopman theory (Brunton et al., 2021) have recently garnered
 125 attention, as they enable high-fidelity modeling of high-dimensional dynamics. Such methodologies
 126 include extensions of Dynamic Mode Decomposition (Schmid, 2010) from linear to more complex
 127 behaviors (Williams et al., 2014; 2015; Li et al., 2017; Bevanda et al., 2022). These methods were
 128 utilized with great success to model and analyze complex dynamical systems in many dimensions
 129 (Redman et al., 2022) with a strong theoretical basis. However, using these approaches, once the
 130 Koopman modes and eigenfunctions are found, more processing must be done in order to further
 131 characterize the underlying vector field. For instance finding and categorizing the specific structure of
 132 the invariant set remains difficult even after finding the Koopman modes. Moreover, to our knowledge,
 133 there is no way to constrain the fitted dynamics to a specific class of dynamical systems.

134 **Modeling vector field data.** Recent advances in experimental biology and related fields have
 135 highlighted the need for improved methods to characterize dynamical systems from noisy data.
 136 Increasingly, new techniques generate datasets with partially or completely unordered temporal
 137 information. A prominent example is single-cell RNA-sequencing along with RNA velocity (Bergen
 138 et al., 2020; La Manno et al., 2018), which defines a vector field in gene expression space and offers
 139 insights into cellular dynamics. Another example of note is the changing proportions of different cell
 140 types over time, which can be represented as noisy vector field data (Mayer et al., 2023). Often, these
 141 systems come with a qualitative understanding of their underlying topology as it often corresponds to
 142 classes of physiological or pathological processes (Nitzan & Brenner, 2021; Karin et al., 2023; Zhou
 143 et al., 2018; Adler et al., 2020). However, aligning this conceptual model with noisy data remains a
 144 challenging task, which is typically pursued separately for each subsystem (Riba et al., 2022; Farrell
 145 et al., 2023). Better methods to characterize such data can have enormous potential, but to the best of
 146 our knowledge there is currently no unified approach to the analysis of such vector field data.

147 3 METHOD



158 **Figure 2: Estimation of limit cycles using SPE for various 2D dynamical systems.** Examples
 159 of invariant sets predicted from observed vectors (in gray), for different dynamical systems which
 160 exhibit a limit cycle attractor. For each system, a ground-truth trajectory was simulated from the
 161 hidden system and plotted in black. The red curves are the limit cycles predicted using SPE. In all
 162 plots, SPE is qualitatively compared to an MLP (in green) trained to reconstruct the vector field.

Throughout this work, we assume that we only observe a sparse subset of the phase space of an ordinary differential equation (ODE), matching observations common to recent advances in computational biology (Bergen et al., 2020; La Manno et al., 2018). Mathematically, let $\mathcal{D} = \{(x_i, \dot{x}_i)\}_{i=1}^N$ be a dataset composed of N pairs of positions $x_i \in \mathbb{R}^d$ and their respective velocities $\dot{x}_i \in \mathbb{R}^d$. We assume that these velocities originate from a *hidden* ground-truth ODE and that we only have access to noisy measurements of the true velocities.

3.1 SMOOTH PROTOTYPE EQUIVALENCES

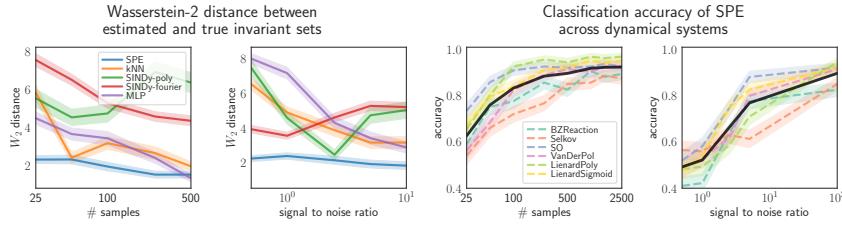


Figure 3: **Wasserstein distance (left) and classification accuracy (right) as a function of the number of observed points and amount of observation noise.** **Left:** W_2 distance between the estimated and true invariant sets, averaged over 1000 randomly sampled datasets (see Appendix C.2), for each method. When changing the number of samples the SNR is held at 2.5, and conversely $N = 100$ points were used when evaluating performance with varying amounts of noise. Shaded areas correspond to standard error of the mean. **Right:** classification performance of SPE, averaged over 1000 randomly sampled datasets (see Appendix C.4). The shaded areas for the different systems represent the standard error of the mean for that system. The black line is the average accuracy over all systems (simple oscillator (SO), BZ reaction, Selkov, Liénard sigmoid, Liénard polynomial and Van der Pol). When changing the number of samples the data was kept noiseless, and $N = 500$ points were used when evaluating performance with varying amounts of noise.

In many practical modeling scenarios, there is strong prior information as to the possible behaviors of the data. For instance, cyclic processes such as the cell cycle or the circadian rhythm in single-cell RNA sequencing (scRNA-seq) typically correspond to a limit cycle behavior (Riba et al., 2022; Schwabe et al., 2020; Karin et al., 2023), or hierarchical processes such as the differentiation of cells correspond to convergence to fixed points in gene expression space (Lange et al., 2022; Farrell et al., 2023). We can use this information to effectively shrink the search space when modeling the dynamics of the observed data, constraining the model to approximately follow our prior knowledge. 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.

We utilize *smooth equivalences* between dynamical systems in order to constrain the model to the prototype's dynamics (Chen et al., 2024). Specifically, two systems $\dot{x} = f(x)$ and $\dot{y} = g(y)$ are said to be smoothly equivalent if:

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

where $H : \mathcal{X} \rightarrow \mathcal{Y}$ 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 two systems share the same qualitative behavior, such as the types and stabilities of their invariant sets. In this work, we parametrize the diffeomorphism using an invertible neural network (INN), $H_\theta(x)$, which is optimized in order to approximately satisfy Equation (1). This is achieved by minimizing the following *equivalence loss* (Chen et al., 2024):

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

When the equivalence loss is 0, the dynamics that generated \dot{x} and $g(y)$ are smoothly equivalent for any point $x \in \mathcal{D}$. Beyond this, it can be shown that when the two systems are equivalent, lower

216 values of the equivalence loss correspond to more faithful reconstructions of the underlying vector
 217 field (Appendix E.1). This further corresponds to a faithful reconstruction of the invariant set of the
 218 underlying vector field if it is structurally stable (Guckenheimer & Holmes, 2013) and under mild
 219 conditions on the partial derivatives of the INN with respect to the inputs (Appendix E.2).

220 In practical scenarios, when the data is both sparse and noisy, even when the two systems are
 221 not equivalent the loss will not necessarily be equal to zero. Nonetheless, it was shown that the
 222 equivalence loss provides a good indication of similarity between dynamical systems when there is
 223 access to a large amount of samples (Chen et al., 2024). To improve convergence, we optimize the
 224 diffeomorphism using a regularized version of Equation (2), penalizing transformations with a large
 225 determinant and those that push the prototype’s invariant set far from the data (see A.5 for details).

226

227 3.2 INVARIANT SET LOCALIZATION AND PROTOTYPE CLASSIFICATION WITH SPE

228

229 Our explicit construction using a mapping between the prototype and the observed data allows us to
 230 effectively *localize* structures in the observed data, which are in general very difficult to find (Smale,
 231 2000). As in our running example, suppose that our prototype $g(y)$ has a limit cycle, and we can
 232 generate points on this limit cycle $\gamma = \{y_1, \dots, y_M\}$. Because we learned an explicit mapping
 233 between the two spaces, then $H_\theta^{-1}(\gamma) = \{H_\theta^{-1}(y_1), \dots, H_\theta^{-1}(y_M)\}$ is guaranteed to be a closed
 234 loop in data space. Moreover, if the equivalence loss between the observed data and the prototype is
 235 low, then $H_\theta^{-1}(\gamma)$ will be a set of points close to the limit cycle of the true hidden system.

236

237 Moreover, SPE allows for classification between a set of prototypes, as the equivalence loss directly
 238 corresponds to a measure of dissimilarity. So, given a dictionary of possible prototypes $\{g_k(y)\}_{k=1}^K$,
 239 we optimize an INN $H_k(x)$ for each, using the equivalence loss from Section 3.1. We then calculate
 240 the equivalence loss, $L_E(H_k, g_k)$, between the observed data and each of the prototypes. If a
 241 prototype and the observed data are smoothly equivalent, and the NN is adequately optimized, then
 242 the equivalence loss for that prototype will be small. This allows us to classify the observed data
 243 according to the prototype that has the smallest equivalence loss.

244

245 3.3 MODELING DIFFEOMORPHISMS WITH INNS

246

247 An important aspect of the equivalence loss from Equation (2) is the differentiable and smooth
 248 mapping between data and prototype space. In our work, we use INNs adapted from normalizing
 249 flows (NFs, Papamakarios et al. 2021) to model our parametric diffeomorphisms. This choice is
 250 guided by the need for efficient and explicit computations of both the log-determinants of the Jacobian
 251 as well as the Jacobian-vector products (JVPs), $\partial_x H_\theta(x) \dot{x}$, needed to calculate the equivalence loss.

252

253 Our INNs utilize a relatively small number of blocks, composed of alternating invertible linear
 254 transformations and affine coupling layers (Dinh et al., 2014). Both of these layer types are adapted so
 255 that their JVPs can be calculated in closed form. Specifically, instead of using multi-layer perceptrons
 256 (MLPs) typically found in affine coupling layers (Papamakarios et al., 2021) we use cosine features,
 257 which we call *Fourier feature coupling*. These Fourier feature coupling layers are highly expressive
 258 even when the network is narrow and shallow, and their JVPs can be calculated efficiently and in
 259 closed form. For details, see Appendix A.

260

261 Other options for invertible mappings exist, such as invertible ResNets (Behrmann et al., 2019; Chen
 262 et al., 2024) or Neural ODEs (Chen et al., 2018). However, while these options are typically more
 263 expressive than INNs adapted from NFs, they come with steep overheads in the calculations of the
 264 inverse function, JVPs and their log-determinants. Because of this, we opted to use sufficiently
 265 expressive networks comprised of affine coupling layers.

266

267 4 RESULTS

268

269 4.1 RECONSTRUCTING ATTRACTORS FROM SPARSE DATA

270

271 First, we demonstrate how SPE can be used to locate attractors in sparse data. This is a challenging
 272 problem since sparsity and noise make the underlying, unknown dynamical equations strongly
 273 undetermined. We demonstrate how SPE circumvents this difficulty on data simulated from a set

270
 271 **Table 1: Comparison of the classification accuracy of SPE to different models across various**
 272 **families of dynamical systems, when classifying between periodic and node dynamics.** The
 273 classification accuracy is averaged over 100 instances for each family of system (see Appendix B.1
 274 for details). The observed vectors were positioned on a dense 64×64 grid, with the addition of
 275 Gaussian noise with a standard deviation of $\sigma = 0.1$ on the velocities. Where relevant, the errors in
 276 accuracy are one standard deviation, calculated over different initializations. The highest scores for
 277 each column are shown in bold, with the second highest underlined.

	SO	Aug. SO	Liénard Poly	Liénard Sig- moid	Van der Pol	BZ Re- action	Sel'kov	Attractor estima- tion
SPE	91±1	81±1	96±1	92±1	97±3	88±2	68±1	✓
TWA	98±1	93±1	<u>86±13</u>	92±7	83±15	82±11	<u>65±3</u>	✗
Critical Points	54	56	70	49	56	84	51	✗
Phase2vec	73±8	71±4	49±6	48±0.1	49±4	50±0.1	49±0.1	✗
Autoencoder	<u>95±3</u>	<u>87±1</u>	63±16	88±9	81±16	82±13	49±2	✗

285
 286
 287 of standard families of two-dimensional dynamical systems exhibiting limit cycles. The governing
 288 equations for these families of systems can be found in Appendix B.1.
 289

290 To that end, we match between observed systems and a single limit cycle prototype, chosen to flexibly
 291 model point and oscillatory dynamics, which is defined in polar coordinates by:

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

292
 293 Here, a is an order parameter that controls whether the system has a node attractor (negative values)
 294 or a limit cycle (positive values), and ω determines the orientation of the cycle (clockwise when
 295 negative and counter-clockwise when positive). Data were acquired by sampling initial conditions
 296 and points along simulated trajectories (details in Appendix B).

297 After fitting an INN $H_\theta(x)$ between the observed data \mathcal{D} and our limit cycle prototype, we estimated
 298 the location of the limit cycle in data space by uniformly sampling a sequence of points, $T_Y =$
 $\{y_j\}_{j=1}^M$, which lie along the limit cycle in prototype space, \mathcal{Y} . Each point was then mapped back
 299 into data space to define a trajectory along the estimated invariant set of the observed dynamics:
 $\hat{T}_X = H_\theta^{-1}(T_Y) = \{H_\theta^{-1}(y_j)\}_{j=1}^M$. Examples of predicted limit cycles can be seen in Figure 2,
 300 where the black orbits are the true limit cycles underlying the observed vectors while the red orbits
 301 are the estimated locations of the same limit cycles by SPE, based on the observed vectors. We
 302 observed a tight fit between predicted and true limit cycles in all example systems (Figures 2 and 8).
 303

304 To more quantitatively evaluate SPE, we calculate the Wasserstein (W_2) distance between samples
 305 from the invariant set of the ground-truth system and those from the invariant set of the fitted model.
 306 To do so, we sample a large number of initial conditions from the phase space of the true system.
 307 We then simulate long trajectories starting from these initial conditions, taking the final position
 308 of these trajectories as an estimate for samples from the true invariant sets. In this manner, we are
 309 also able to compare SPE to baseline methods for estimating vector fields from noisy data. The
 310 baselines we compare to SPE are (Appendix C.1): a ***k*-nearest neighbor (kNN) interpolator** which
 311 estimates the velocity at a position to be the average velocities of the k nearest neighbors to the
 312 position; **SINDY** (Brunton et al., 2016), which approximates the vector as a sparse linear regression
 313 with a prespecified dictionary of functions; **MLP**, which approximates the underlying ODE using a
 314 multilayer perceptron. Note that while these methods serve as useful baselines for the reconstruction
 315 quality of the underlying vector field, unlike SPE they do not explicitly model the invariant sets and
 316 do not allow for its direct reconstruction.

317 When the positions x_i are sparse, or the vectors are noisy, SPE more accurately reconstructs the
 318 underlying invariant sets than other baselines (Figure 3). In particular, when the phase space is very
 319 sparsely populated or noisy, the baselines tend to estimate a vector field with large or none existent
 320 invariant sets, an example of which is shown in Figure 2 (right). In contrast, as SPE is constrained to
 321 model only stable dynamics, it is able to more closely reconstruct the underlying vector field.

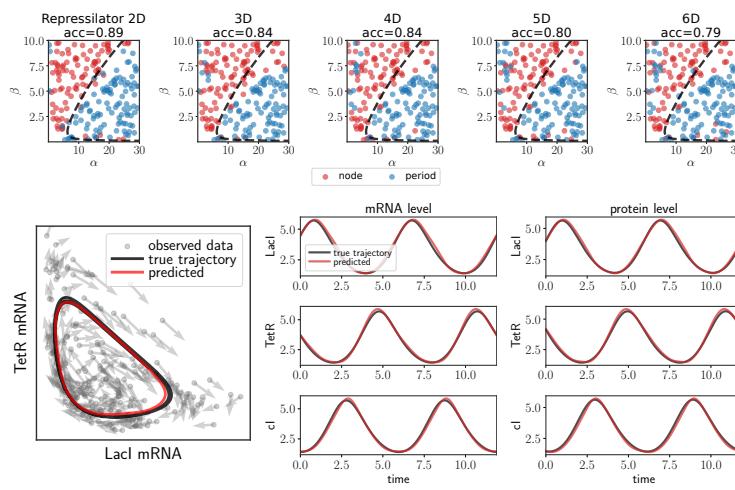


Figure 4: **Classifying dynamics in higher dimensions and recovering the limit cycle from the repressilator system.** **Top:** each point in the scatter represents a system with specific parameters that was classified by SPE as either a 2D embedded limit cycle (blue) or node attractor (red). The dashed black line depicts the ground truth bifurcation boundary between periodic and node behavior. Each column depicts dynamics projected onto a different dimensionality. **Bottom:** qualitative results of fitting a 6-dimensional repressilator system, projected to the LacI-TetR mRNA plane (left). The gray arrows depict observed position-velocity pairs, (x_i, \dot{x}_i) , the black line is the ground-truth invariant set of the dynamics and the red line is the predicted invariant set using SPE. Time series of the gene expression (center) and protein levels (right) can be extracted from the predicted limit cycle (shown in red). These are overlayed on top of a trajectory simulated from the underlying (hidden) system, which was simulated for a long time to ensure convergence to the limit cycle, shown in black.

4.2 CLASSIFYING DYNAMICAL SYSTEMS WITH PROTOTYPE EQUIVALENCE

The results of Section 4.1 relied on knowing that the observed system exhibited a particular invariant set. Here, we show that SPE can discover this information directly from the data by fitting multiple prototypes and using the loss to classify the behavior of the dynamics. As our prototype set, we choose all four effective behaviors implied by Equation (3), with $a = \pm \frac{1}{4}$ and $\omega = \pm \frac{1}{2}$. Four diffeomorphisms, $H_k(x)$, were fit to each observed system in parallel and the one with the minimal value of the equivalence loss defined in Equation (2) was used to determine class membership.

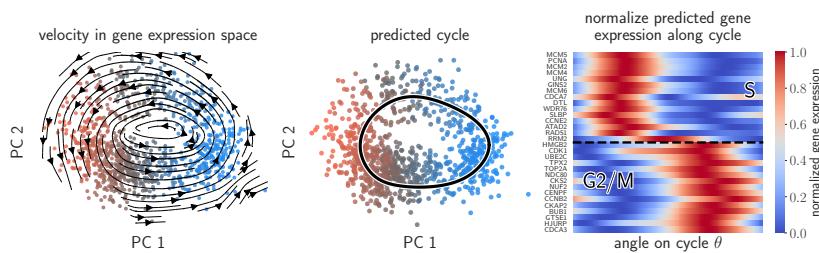
We begin by benchmarking SPE next to existing methods for classifying between node or cyclic behaviors on the dataset introduced by Moriel et al. 2023. This dataset is based on simulated families of systems whose order parameters correspond to either a node or a periodic attractor (details for governing equations can be found 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 1 shows the average accuracy of our method for each family of systems, compared to the following baselines (more details in Appendix C.3): **TWA** (Moriel et al., 2023), a convolutional network trained to classify between periodic and point dynamical systems; **Critical Points** (Helman & Hesselink, 1989; 1991), a heuristic approach that identifies critical points in the vector field; **Phase2vec** (Ricci et al., 2023), a deep learning approach for dynamical systems embedding, which learns vector field representations; and **Autoencoder**, an autoencoder trained to reconstruct the vectors, whose latent code is used to classify the dynamics. 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, is more performant (Table 1).

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 Section 4.1). To test the capabilities of SPE in sparse and noisy scenarios, in Figure 3 (right) we track the classification accuracy when decreasing

378 the number of observed vectors N and under increasingly noisy scenarios, with low signal-to-noise
 379 ratios (SNRs) (see Appendix C.4 for details regarding data). The decline in performance is gradual
 380 along both of these axes - even when classifying using only 50 random vectors, average accuracy was
 381 still relatively high, at $\sim 70\%$.

382 SPE’s classification and localization performance is not limited to limit cycles and node attractors, and
 383 can be flexibly generalized to additional prototypes. To demonstrate such capabilities, we evaluated
 384 the performance of SPE on a challenging classification of multistable systems with a varying number
 385 of basins of attraction (Appendix F.1). Our results show that SPE can successfully distinguish between
 386 dynamical systems with varying numbers of fixed points, and can additionally localize them in space
 387 (Figure 11). Furthermore, as the loss used to train SPE corresponds to a graded notion of equivalence,
 388 we found that SPE is also able to capture dominant behaviors of the system even when there is a
 389 mismatch between the true behavior and the chosen prototype. We tested this on the classification of
 390 the orientation of the flow in Appendix F.2.

391
 392 **4.3 BEYOND TWO DIMENSIONS**
 393
 394



405 **Figure 5: Recovering the cell cycle gene expression pattern from single-cell data.** **Left:** gene
 406 expression velocity of cells undergoing proliferation in PCA space. Colors denote the cycle phase
 407 score based on marker gene expression; red corresponds to high expression of S-phase markers, blue
 408 to high expression of G2/M-phase genes, and gray to low levels in both. **Middle:** SPE was used to fit
 409 a periodic prototype to the RNA velocity data; the resulting attractor is plotted in black. The cells’
 410 locations and colors are as in the plot on the left. **Right:** predicted normalized gene expressions along
 411 the cycle attractor predicted by SPE. Specifically, the normalized expression is plotted vs. the angle
 412 on the limit cycle (θ), for a set of common marker genes of the S (top of the heatmap) and G2/M
 413 (bottom of the heatmap) phases. See details in Appendix D.

414 Empirical measurements of dynamical systems, whose intrinsic behavior often lies on a low-
 415 dimensional manifold, are often embedded in high-dimensional spaces (Champion et al., 2019).
 416 Characterizing higher-dimensional systems, whether by localization of invariant structures or by
 417 classification of qualitative behaviors from a set of possibilities, is a challenging problem as the
 418 location of the lower-dimensional invariant set needs to be identified, and then matched, within the
 419 higher dimensional ambient space.

420 To test our approach in higher-dimensional systems, we first classify and localize invariant structures
 421 in data from the repressilator model of gene regulation (Elowitz & Leibler, 2000) (see Appendix B.3
 422 for governing equations). The repressilator models a genetic regulatory network in bacteria, based on
 423 the mRNA and protein counts of three genes - TetR, LacI and λ phage cI - which inhibit each other
 424 in a cyclic fashion. Even though this family of systems spans six dimensions, under specific parameter
 425 values it exhibits either a 2D point attractor or contains an embedded limit cycle (Verdugo, 2018a;
 426 Potvin-Trottier et al., 2016). As such, to model data from this system we use higher-dimensional
 427 prototypes that are the combination of two behaviors: (1) a 2D limit cycle/equilibrium behavior as in
 428 Equation (3); (2) exponential decay to a fixed point along variables uncoupled from the embedded
 429 2D system. Mathematically, we define these prototypes as:

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

430 with $y_1 \in \mathbb{R}^2$, $y_2 \in \mathbb{R}^{d-2}$, where $g_{2D}(\cdot)$ are velocities in two dimensions, and τ is a decay factor. We
 431 used $\tau = 1/2$ in all of 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/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 Section 4.1, Figure 4. 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 (Figure 4) that our model can classify two-dimensional repressilator systems better than earlier work (Moriel et al., 2023), 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 (Figure 4).

4.4 UNCOVERING THE CELL CYCLE EXPRESSION PATTERN IN SINGLE-CELL DATA

Finally, we show how SPE can be used to trace the periodic gene expression dynamics of proliferating cells. 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). Here, we analyze single-cell RNA-sequencing data from a human cell line, profiling over 50,000 genes across more than 1,000 cells (Mahdessian et al., 2021). Previous approaches have inferred cell-cycle progression using marker genes (Tirosh et al., 2016), a learned projection (Zheng et al., 2022; Riba et al., 2022), or using structural priors (Schwabe et al., 2020; Karin et al., 2023), but these do not directly leverage the fact that the cycle has a characteristic dynamical structure in the form of a limit cycle. Here, we show that cell cycle structure can be estimated from RNA velocity (Bergen et al., 2021; Gorin et al., 2022), effective local velocities of cells in gene expression space that are based on differences in spliced and unspliced RNA counts.

Using scVelo (Bergen et al., 2020), we infer high-dimensional RNA velocities for all sampled cells and project them together with their associated gene expression profiles, into a 100-dimensional space using principal component analysis (PCA). The resulting streamlines (Figure 5, left) reveal a noisy trajectory traversing across distinct S-phase and G2/M-phase gene expression states. We then fit a high-dimensional limit-cycle attractor (as described in Equation (4)) to the RNA velocity data, capturing the periodic nature of the cell cycle (Figure 5, middle, see Appendix D for more details on data and prototype). Utilizing the PCA projection, we can map the learned attractor back to gene expression space, revealing periodic, anti-correlated expression patterns of key cell-cycle marker genes for the S and G2/M-phase (Figure 5, right). This demonstrates our method’s ability to recover biologically meaningful periodic processes directly from high-dimensional, noisy and sparse experimental data. These results generalize across datasets; in Figure 10 in the appendix, we analyze an additional dataset of proliferating human fibroblasts (Riba et al., 2022), and show that there too SPE is able to localize the cell-cycle from single-cell gene expression and RNA velocity alone, using the same preprocessing and fitting procedure as described for Figure 5 (Appendix D).

5 DISCUSSION AND FUTURE DIRECTIONS

Our proposed approach, *smooth prototype equivalences*, 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. We demonstrated the efficacy and efficiency of this approach on both challenging synthetic and real-world data sets. Furthermore, SPE can be used as a way to classify the long-term dynamics in situations where there are multiple possible ground-truth behaviors.

In this work, our focus was the detection and classification of long-term behaviors that are robust to noise, such as limit cycles and node attractors. Indeed, our main focus were phenomena from empirical biological systems, where dynamics need to be repeatable for proper function. Examples for such dynamics are differentiation of cells into different types (Bergen et al., 2020), the oscillatory process of the circadian clock (Karin et al., 2023), attractors of artificial and biological recurrent neural networks, or the cell cycle (Riba et al., 2022; Mahdessian et al., 2021; Schwabe et al., 2020)

486 which we explored in Section 4.4, among others. For such systems, which are structurally stable,
 487 SPE converges at a faithful reconstruction of the true invariant structures (as shown in Appendix E).
 488

489 At the other side of this scale are systems which are structurally unstable and, as a special case of
 490 particular interest, chaotic systems with strange attractors. SPE was not designed for such scenarios,
 491 where even very small perturbations of the underlying vector field can break the structure of the
 492 invariant sets they contain. Applying prototype equivalence approaches to chaotic systems, in
 493 particular, is an important future challenge, since strange attractors of different fractal dimensions
 494 are not smoothly equivalent to each other leaving open the question of how to learn appropriate
 495 invariances, especially when subject to experimental noise.

496 While we have shown that SPE is a useful tool for characterizing of dynamical systems, it explicitly
 497 depends on the definition of prototypes which are to be matched to empirical data. SPE makes up for
 498 its reliance on pre-defined prototypes, however, with its relaxed notion of equivalence, where a best
 499 possible transformation between prototype and data can be found, so that even a partially matching
 500 prototype can help localize and categorize dynamical behaviors of scientific interest (demonstrated
 501 in Appendix F.2). Empirically, we observe that the equivalence loss can give an ordering on the
 502 amount of mismatch between the true underlying dynamics and the chosen prototypes (as shown
 503 Appendix F.4).

504 We expect that in the future SPE can be combined with other methodologies from the field of nonlinear
 505 dynamics. For instance, we view our method as complementary to Koopman theory (Brunton et al.,
 506 2021), which might allow more diverse dynamics in the operator space. Another extension of interest
 507 would be the combination of SPE together with time-delay embeddings, techniques which can be
 508 used together in order to elucidate invariant structures in time series data using SPE. Finally, while in
 509 this work our primary focus were locally simple behaviors, such as node or cycle attractors or a small
 510 collection of node attractors (Appendix F.1), an important future step would entail the combination of
 511 multiple local dynamics into a larger, more complex system. One way to do so is the incorporation of
 512 SPE in models which explicitly break down space into separate regions, such as mixtures of experts
 513 (Jacobs et al., 1991). In such models, SPE can be used as an interpretable component of a more
 514 complicated model, capturing intricate dynamics in many dimensions. We also envisage that SPE can
 515 be paired with more advanced parameterizations of INNs in order to tackle more complex behaviors,
 516 in higher dimensional settings, which may take inspiration from computer vision (Papamakarios
 517 et al., 2021; Dinh et al., 2014; Durkan et al., 2019b).

518 Our method directly addresses challenges faced in scientific domains such as biology, neuroscience,
 519 and physics. By offering an equation-free approach for the detection and classification of invariant
 520 sets, our method has the potential to advance the understanding of complex, nonlinear processes in
 521 systems where analytical modeling is impractical.

522 REPRODUCIBILITY STATEMENT

524 All details necessary to reproduce the results of this work can be found in the appendix:

- 526 • Full specifications of our invertible NNs are in Appendix A.
- 527 • Hyperparameter choices for all experiments can be found in Appendix A.7.
- 528 • Governing equations of all simulated systems are described in Appendix B.1 for 2D systems
 529 and Appendix B.3 for the 6D repressilator system.
- 530 • Detailed instructions to generate observations from the simulated systems are given in
 531 Appendix B.2.

533 Furthermore, the code will be made publicly available.

536 REFERENCES

- 538 Miri Adler, Avi Mayo, Xu Zhou, Ruth A Franklin, Matthew L Meizlish, Ruslan Medzhitov, Stefan M
 539 Kallenberger, and Uri Alon. Principles of cell circuits for tissue repair and fibrosis. *IScience*, 23
 (2), 2020.

- 540 Aleksandr A. Andronov and Lev S. Pontryagin. Systèmes grossiers. *Doklady Akademii Nauk SSSR*,
 541 14:247–251, 1937. Translated as "Rough Systems" in later works.
 542
- 543 Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen.
 544 Invertible residual networks. In *International conference on machine learning*, pp. 573–582.
 545 PMLR, 2019.
- 546 Volker Bergen, Marius Lange, Stefan Peidli, F Alexander Wolf, and Fabian J Theis. Generalizing
 547 rna velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38(12):
 548 1408–1414, 2020.
- 549 Volker Bergen, Ruslan A Soldatov, Peter V Kharchenko, and Fabian J Theis. Rna velocity—current
 550 challenges and future perspectives. *Molecular systems biology*, 17(8):e10282, 2021.
- 552 Petar Bevanda, Johannes Kirmayr, Stefan Sosnowski, and Sandra Hirche. Learning the koopman
 553 eigendecomposition: A diffeomorphic approach. In *2022 American Control Conference (ACC)*,
 554 pp. 2736–2741. IEEE, 2022.
- 555 Christopher M Bishop, Markus Svensén, and Christopher KI Williams. Gtm: The generative
 556 topographic mapping. *Neural computation*, 10(1):215–234, 1998.
- 558 Edwin Bonilla and Antonio Robles-Kelly. Discriminative probabilistic prototype learning. *arXiv*
 559 *preprint arXiv:1206.4686*, 2012.
- 560 Jason J Bramburger, Steven L Brunton, and J Nathan Kutz. Deep learning of conjugate mappings.
 561 *Physica D: Nonlinear Phenomena*, 427:133008, 2021.
- 563 Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data
 564 by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U. S. A.*, 113(15):
 565 3932–3937, 2016.
- 567 Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as
 568 an intermittently forced linear system. *Nature communications*, 8(1):19, 2017.
- 569 Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for
 570 dynamical systems. 2021.
- 572 Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of
 573 coordinates and governing equations. *Proc. Natl. Acad. Sci. U. S. A.*, 116(45):22445–22451, 2019.
- 575 Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary
 576 differential equations. *Advances in neural information processing systems*, 31, 2018.
- 577 Ruiqi Chen, Giacomo Vedovati, Todd Braver, and ShiNung Ching. Dform: Diffeomorphic vector
 578 field alignment for assessing dynamics across learned models. *arXiv preprint arXiv:2402.09735*,
 579 2024.
- 581 Shiming Chen, Wenjin Hou, Ziming Hong, Xiaohan Ding, Yibing Song, Xinge You, Tongliang
 582 Liu, and Kun Zhang. Evolving semantic prototype improves generative zero-shot learning. In
 583 *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org,
 584 2023.
- 585 Alexander Chervov and Andrei Zinovyev. Computational challenges of cell cycle analysis using
 586 single cell transcriptomics. *arXiv [q-bio.QM]*, August 2022.
- 587 Brian M de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J Nathan Kutz,
 588 and Steven L Brunton. PySINDy: A python package for the sparse identification of nonlinear
 589 dynamics from data. 2020.
- 591 Grigorios Dimitriadis. *Introduction to nonlinear aeroelasticity*. John Wiley & Sons, 2017.
- 592 Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components
 593 estimation. *arXiv preprint arXiv:1410.8516*, 2014.

- 594 Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances*
 595 *in neural information processing systems*, 32, 2019a.
 596
- 597 Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances*
 598 *in neural information processing systems*, 32, 2019b.
 599
- 600 Michael B Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators.
 601 *Nature*, 403(6767):335–338, 2000.
 602
- 603 Spencer Farrell, Madhav Mani, and Sidhartha Goyal. Inferring single-cell transcriptomic dynamics
 604 with structured latent gene expression dynamics. *Cell Reports Methods*, 3(9), 2023.
 605
- 606 Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas
 607 Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron,
 608 Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet,
 609 Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and
 610 Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8,
 611 2021. URL <http://jmlr.org/papers/v22/20-451.html>.
 612
- 613 Bryan Glaz. Efficient pseudometrics for data-driven comparisons of nonlinear dynamical systems,
 614 2024. URL <https://arxiv.org/abs/2409.18681>.
 615
- 616 Gennady Gorin, Meichen Fang, Tara Chari, and Lior Pachter. Rna velocity unraveled. *PLOS*
 617 *Computational Biology*, 18(9):e1010492, 2022.
 618
- 619 John Guckenheimer and Philip Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations*
 620 *of vector fields*, volume 42. Springer Science & Business Media, 2013.
 621
- 622 Philip Hartman. A lemma in the theory of structural stability of differential equations. *Proceedings*
 623 *of the American Mathematical Society*, 11(4):610–620, 1960. doi: 10.2307/2034363.
 624
- 625 James Helman and Lambertus Hesselink. Representation and display of vector field topology in fluid
 626 flow data sets. *Computer*, 22(08):27–36, 1989.
 627
- 628 James L Helman and Lambertus Hesselink. Visualizing vector field topology in fluid flows. *IEEE*
 629 *Computer Graphics and Applications*, 11(3):36–46, 1991.
 630
- 631 Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint*
 632 *arXiv:1606.08415*, 2016.
 633
- 634 Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of
 635 local experts. *Neural computation*, 3(1):79–87, 1991.
 636
- 637 Jonathan Karin, Yonathan Bornfeld, and Mor Nitzan. scPrisma infers, filters and enhances topological
 638 signals in single-cell data using spectral template matching. *Nat. Biotechnol.*, February 2023.
 639
- 640 Katarzyna M Kedziora and Wayne Stallaert. Cell cycle mapping using multiplexed immunofluores-
 641 cence. *Methods Mol. Biol.*, 2740:243–262, 2024.
 642
- 643 Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*,
 644 2014.
 645
- 646 Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer.
 647 Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy*
 648 *of Sciences*, 118(21):e2101784118, 2021.
 649
- 650 Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov,
 651 Katja Lidschreiber, Maria E Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of
 652 single cells. *Nature*, 560(7719):494–498, 2018.
 653
- 654 Marius Lange, Volker Bergen, Michal Klein, Manu Setty, Bernhard Reuter, Mostafa Bakhti, Heiko
 655 Lickert, Meshal Ansari, Janine Schniering, Herbert B Schiller, et al. Cellrank for directed single-cell
 656 fate mapping. *Nature methods*, 19(2):159–170, 2022.
 657

- 648 Qianxiao Li, Felix Dietrich, Erik M Boltt, and Ioannis G Kevrekidis. Extended dynamic mode
 649 decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the
 650 koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10), 2017.
- 651
- 652 Diana Mahdessian, Anthony J Cesnik, Christian Gnann, Frida Danielsson, Lovisa Stenström, Muham-
 653 mad Arif, Cheng Zhang, Trang Le, Fredric Johansson, Rutger Schutten, et al. Spatiotemporal
 654 dissection of the cell cycle with single-cell proteogenomics. *Nature*, 590(7847):649–654, 2021.
- 655 Shimrit Mayer, Tomer Milo, Achinoam Isaacson, Coral Halperin, Shoval Miyara, Yaniv Stein, Chen
 656 Lior, Meirav Pevsner-Fischer, Eldad Tzahor, Avi Mayo, et al. The tumor microenvironment shows
 657 a hierarchy of cell-cell interactions dominated by fibroblasts. *Nature communications*, 14(1):5810,
 658 2023.
- 659 Noa Moriel, Matthew Ricci, and Mor Nitzan. Let’s do the time-warp-attend: Learning topological
 660 invariants of dynamical systems. *arXiv preprint arXiv:2312.09234*, 2023.
- 661
- 662 Mor Nitzan and Michael P Brenner. Revealing lineage-related signals in single-cell gene expres-
 663 sion using random matrix theory. *Proceedings of the National Academy of Sciences*, 118(11):
 664 e1913931118, 2021.
- 665 Mitchell Ostrow, Adam Eisen, Leo Kozachkov, and Ila Fiete. Beyond geometry: Comparing the
 666 temporal structure of computation in neural circuits with dynamical similarity analysis. *Advances
 667 in Neural Information Processing Systems*, 36:33824–33837, 2023.
- 668 George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji
 669 Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of
 670 Machine Learning Research*, 22(57):1–64, 2021.
- 671
- 672 Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University
 673 of Denmark*, 7(15):510, 2008.
- 674 Henri Poincaré. *Les Méthodes Nouvelles de la Mécanique Céleste*, volume 1–3. Gauthier-Villars,
 675 Paris, France, 1892–1899. Available as a reprint from Dover Publications, 1957.
- 676
- 677 Laurent Potvin-Trottier, Nathan D Lord, Glenn Vinnicombe, and Johan Paulsson. Synchronous
 678 long-term oscillations in a synthetic gene circuit. *Nature*, 538(7626):514–517, 2016.
- 679 William T. Redman, Maria Fonoberova, Ryan Mohr, Ioannis G. Kevrekidis, and Igor Mezić. Algo-
 680 rithmic (semi-)conjugacy via koopman operator theory, 2022. URL <https://arxiv.org/abs/2209.06374>.
- 681
- 682 Andrea Riba, Attila Oravecz, Matej Durik, Sara Jiménez, Violaine Alunni, Marie Cerciat, Matthieu
 683 Jung, Céline Keime, William M Keyes, and Nacho Molina. Cell cycle gene regulation dynamics
 684 revealed by rna velocity and deep-learning. *Nature communications*, 13(1):2865, 2022.
- 685
- 686 Matthew Ricci, Noa Moriel, Zoe Piran, and Mor Nitzan. Phase2vec: Dynamical systems embedding
 687 with a physics-informed convolutional network. *Proceedings of the 11th International Conference
 688 on Learning Representations, ICLR, Kigali, Rwanda, May 1-5*, 2023.
- 689
- 690 Eleanor H Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.
- 691
- 692 Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*,
 656:5–28, August 2010.
- 693
- 694 Daniel Schwabe, Sara Formichetti, Jan Philipp Junker, Martin Falcke, and Nikolaus Rajewsky. The
 695 transcriptome dynamics of single cells during the cell cycle. *Mol. Syst. Biol.*, 16(11):e9946,
 November 2020.
- 696
- 697 Joseph D. Skufca and Erik M. Boltt. Relaxing conjugacy to fit modeling in dynamical systems. *Phys.
 698 Rev. E*, 76:026220, Aug 2007. doi: 10.1103/PhysRevE.76.026220. URL <https://link.aps.org/doi/10.1103/PhysRevE.76.026220>.
- 699
- 700 Joseph D Skufca and Erik M Boltt. A concept of homeomorphic defect for defining mostly conjugate
 701 dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18(1):013118,
 2008.

- 702 Steve Smale. Mathematical problems for the next century. *Mathematics: frontiers and perspectives*,
 703 pp. 271–294, 2000.
 704
- 705 Wayne Stallaert, Katarzyna M Kedziora, Colin D Taylor, Tarek M Zikry, Holly K Sobon, Sovanny R
 706 Taylor, Catherine L Young, Juanita C Limas, Jeanette G Cook, and Jeremy E Purvis. The structure
 707 of the human cell cycle. *Cold Spring Harbor Laboratory*, pp. 2021.02.11.430845, February 2021.
- 708 Shlomo Sternberg. Local contractions and a theorem of poincaré. *American Journal of Mathematics*,
 709 79(4):809–824, 1957. doi: 10.2307/2372447.
 710
- 711 Floris Takens. Detecting strange attractors in turbulence. In David A. Rand and Lai-Sang Young (eds.),
 712 *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*,
 713 pp. 366–381. Springer-Verlag, Berlin, Heidelberg, 1981. doi: 10.1007/BFb0091924.
 714
- 715 Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth, Daniel Treacy, John J Trombetta,
 716 Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, et al. Dissecting the multi-
 717 cellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, 352(6282):189–196,
 718 2016.
- 719 Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow.
 720 *arXiv preprint arXiv:1611.09630*, 2016.
 721
- 722 Anael Verdugo. Hopf bifurcation analysis of the repressilator model. *American Journal of Computational Mathematics*, 8(02):137–152, 2018a.
 723
- 724 Anael Verdugo. Hopf bifurcation analysis of the repressilator model. *American Journal of Computational Mathematics*, 8(2):137–152, 2018b.
 725
- 727 Terry A Weisshaar. Aeroelasticity, an introduction to fundamental problems-with an historical
 728 perspective, examples and homework problems, 2012.
 729
- 730 Matthew O Williams, Clarence W Rowley, and Ioannis G Kevrekidis. A kernel-based approach to
 731 data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.
 732
- 733 Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation
 734 of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*,
 735 25(6):1307–1346, 2015.
- 736 Shijie C Zheng, Genevieve Stein-O’Brien, Jonathan J Augustin, Jared Slosberg, Giovanni A Carosso,
 737 Briana Winer, Gloria Shin, Hans T Bjornsson, Loyal A Goff, and Kasper D Hansen. Universal
 738 prediction of cell-cycle position using transfer learning. *Genome Biol.*, 23(1):41, January 2022.
 739
- 740 Shijie C Zheng, Genevieve Stein-O’Brien, Leandros Boukas, Loyal A Goff, and Kasper D Hansen.
 741 Pumping the brakes on rna velocity by understanding and interpreting rna velocity estimates.
 742 *Genome biology*, 24(1):246, 2023.
- 743 Xu Zhou, Ruth A Franklin, Miri Adler, Jeremy B Jacox, Will Bailis, Justin A Shyer, Richard A
 744 Flavell, Avi Mayo, Uri Alon, and Ruslan Medzhitov. Circuit design features of a stable two-cell
 745 system. *Cell*, 172(4):744–757, 2018.
 746
- 747 Andrei Zinovyev, Michail Sadovsky, Laurence Calzone, Aziz Fouché, Clarice S Groeneveld, Alexan-
 748 der Chervov, Emmanuel Barillot, and Alexander N Gorban. Modeling progression of single cell
 749 populations through the cell cycle as a sequence of switches. *bioRxiv*, pp. 2021.06.14.448414,
 750 August 2021.
- 751

752 A NORMALIZING FLOW DETAILS 753

754 The diffeomorphisms $H : \mathcal{X} \rightarrow \mathcal{Y}$ used in this work are invertible neural networks (INNs) with
 755 alternating *Affine* and *Fourier Feature Coupling* (FFCoupling) transforms, both of which are explained

in further details below. Furthermore, the first layer of the normalizing flow was set to standardize the data, effectively an *ActNorm* layer Papamakarios et al. (2021), defined as:

$$[\text{ActNorm}(x)]_i = \frac{x_i - \text{mean}(x_i)}{\text{std}(x_i)} \quad (5)$$

In the above, $\text{mean}(x_i)$ and $\text{std}(x_i)$ are the mean and standard deviations of i -th coordinate of the observed data, respectively. This *ActNorm* layer was kept frozen after initialization, and means that the normalization of the data is explicitly part of the learned diffeomorphism.

A.1 FOURIER FEATURE COUPLING

The FFCoupling layer we used is a modification of the standard *AffineCoupling* Papamakarios et al. (2021); Dinh et al. (2014) layer used in normalizing flows. *AffineCoupling* splits an input x to two sub-vectors x_1 and x_2 , and the transformation itself is defined as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \text{AffineCoupling}(x) = \begin{bmatrix} x_1 \\ \exp[f_s(x_1)] \circ x_2 + f_t(x_1) \end{bmatrix} \quad (6)$$

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 function, and \circ stands for element-wise multiplication. *AffineCoupling* is defined in this manner to ensure invertibility, in which case $f_s(\cdot)$ and $f_t(\cdot)$ can be arbitrarily complex functions. These two functions are typically parameterized by a multi-layer perceptron (MLP). Baked into the *AffineCoupling* transformation is the simple form of its log-determinant:

$$\log |\text{AffineCoupling}(x)| = \sum_i [f_s(x_1)]_i \quad (7)$$

Finally, note that *AffineCoupling* only transforms x_2 as a function of x_1 . To account for this, 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.

Unfortunately, when both the scale and the translation functions are defined as MLPs, calculating the Jacobian or Jacobian-vector products (JVPs) is not straightforward. These JVPs are needed for our loss (Equation (2)) and can potentially be found using autograd methods, albeit at a high computational cost. Furthermore, we expect these diffeomorphisms to typically be quite smooth and act on a range of frequencies, properties which are not easily enforced in MLPs. Instead, we opt to use transformations whose JVPs can be calculated in closed-form and whose properties are well understood.

For our scale and translation functions we use the following *FFCoupling*:

$$\mathcal{F}(x) = \sum_{k=1}^K \Theta_k \cos\left(\frac{2\pi}{R}k \cdot x + \phi_k\right) \quad (8)$$

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. When K is large, this transformation is expressive enough to fit any function defined in the range $[-R/2, R/2]$. It also admits a very simple form for the Jacobian, given by:

$$\frac{\partial}{\partial x} \mathcal{F}(x) = -\frac{2\pi}{R} \sum_k k \cdot \Theta_k \text{diag}\left[\sin\left(\frac{2\pi}{R}k \cdot x + \phi_k\right)\right] \quad (9)$$

A.2 AFFINE TRANSFORMATION LAYER

Besides the FFCoupling layer, we also use affine transformations. To ensure invertibility, we parameterize this transformation as:

$$\text{Affine}(x) = WW^T x + e^\varphi \cdot x + \mu \quad (10)$$

where $W \in \mathbb{R}^{d \times q}$ with some chosen q , $\varphi \in \mathbb{R}$ and $\mu \in \mathbb{R}^d$ are learnable parameters. This factorization ensures that the matrix $WW^T + Ie^\varphi$ is positive-definite (PD), and thus invertible. It also enables closed-form inversion using the matrix-inversion lemma.

810 A.3 HOUSEHOLDER TRANSFORMATIONS
811

812 In general, we found that the Affine and FFCoupling layers were expressive enough, when the
813 observed data was mostly aligned with the axes of the prototype. However, in higher dimensions
814 there is no reason to a-priori believe that this should be true. To overcome this difficulty, we used
815 *Householder transformations* (Papamakarios et al., 2021; Tomczak & Welling, 2016), which act as
816 building blocks for orthogonal transformations. The Householder transformation is defined as:

$$817 \quad 818 \quad \text{HH}(x) = \left(I - \frac{2}{\|v\|^2} vv^T \right) x \quad 819 \quad (11)$$

820 where $v \in \mathbb{R}^d$ are the parameters of the transformation. This transformation is orthogonal, and so has
821 a determinant of 1. Furthermore, it can be shown (Tomczak & Welling, 2016) that any rotation in a
822 d -subspace can be carried out with d of these transformations. Thus, these transforms are useful for
823 more directly rotating space, which is difficult to achieve directly with the Affine transformations
824 defined above.

825 A.4 NETWORK AND TRAINING SPECIFICATIONS
826

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

830 Before these blocks, our networks include a frozen ActNorm layer that standardizes the data and
831 ensures that all learned diffeomorphisms have similar inputs, and then a full-rank Affine transforma-
832 tion, with learnable weights. In the cell-cycle (Section 4.4) and multi-stable dynamics (Appendix F.1)
833 experiments we also added d Householder transforms, where d is the dimension of the data.

834 A.5 OPTIMIZATION CRITERIA
835

836 In addition to the equivalence loss, L_E , we introduced a projection regularizer, L_{proj} , that aligns the
837 invariant structures of the prototype with the observed data points. This regularization significantly
838 improves convergence in complex scenarios and promotes identifiability of the solution by encour-
839 aging the observed data to lie close to the invariant set implied by the prototype dynamics and the
840 learned diffeomorphism. This regularization is described in full detail in Appendix A.6. Moreover,
841 we found it helpful to regularize the log-determinant to be close to 0, which penalizes networks
842 excessively expanding or contracting space. The full loss used to optimize the diffeomorphism H in
843 SPE is thus given by:

$$844 \quad L_{\text{tot}}(H_\theta, g) = L_E(H_\theta, g) + L_{\text{proj}}(H_\theta, g, \lambda_{\text{proj}}) + \lambda_{\text{det}} L_{\text{det}}(H_\theta). \quad 845 \quad (12)$$

846 All training is carried out using the Adam optimizer Kingma (2014) with weight decay.

847 A.6 PROJECTION REGULARIZATION
848

849 When using our high-dimensional prototypes from Equation (4), we found that performance was
850 improved by regularizing the diffeomorphism to map the data as close as possible to the invariant
851 sets of the embedded 2D dynamics. Our definition of the high-dimensional prototypes essentially
852 assumes that most of the observed dynamics are at, or close to, a 2D subset of the full space which
853 contains the invariant set of the dynamics. Moreover, our explicit construction of the prototypes,
854 with simple governing equations, allows us to project points in 2D onto their invariant sets. For
855 dynamics of this sort, we expect the positions x_i to be close to the invariant set embedded in a higher
856 dimension. Utilizing this knowledge during optimization can aid convergence, and can be done using
857 our definition of the prototype. Specifically, using the diffeomorphism we can project the points onto
858 the 2D invariant set of the prototype, and then back into data space as follows:

$$859 \quad P(x; H_\theta) = H_\theta^{-1} ([H_\theta(x)]_1, [H_\theta(x)]_2, 0, \dots, 0) \quad 860 \quad (13)$$

861 where $[H_\theta(x)]_i$ is the i -th coordinate of $H_\theta(x)$ and $P(x; H_\theta)$ denotes the projection according to
862 $H_\theta(x)$.

864 We model the projection into data space through a Gaussian observation model, with the following
 865 loss added when training:
 866

$$867 \quad L_{\text{proj}}(H_\theta, g, \lambda_{\text{proj}}) = e^{\lambda_{\text{proj}}} \frac{1}{N} \sum_i^N \|x_i - P(x_i; H_\theta)\|^2 - \frac{1}{N} \lambda_{\text{proj}} \quad (14)$$

$$868$$

$$869$$

870 where λ_{proj} is the regularization coefficient, which takes the form of the log of the precision of the
 871 Gaussian observation model. Adding the regularization coefficient in this manner means that it can
 872 be optimized in parallel to the diffeomorphism, and allows for an adaptive fitting process. When
 873 modeled in this way, the projection is similar to generative topographic mapping (GTM, Bishop et al.
 874 (1998)) when the matching between the latent codes and the data is known.

875 A.7 HYPERPARAMETERS SETTINGS

876 For all simulated systems, hyperparameters were chosen to maximize performance on a separate,
 877 held out validation set of 100 datasets. During evaluation time, the performance of the methods was
 878 not tested on these held out datasets.

879 **Comparison of classification on dense grids:** for the comparisons in Table 1, we found that weight
 880 decay and the other regularizations were not helpful, so all regularization coefficients were set to
 881 0. Additionally, because this data is specified on a dense grid, with the invariant set typically in the
 882 center of this grid, we found that the initial ActNorm hurt performance. The initial ActNorm layer in
 883 the INN was designed to center data that is assumed to be near the invariant set, but the grid data
 884 from Moriel et al. does not follow this assumption. Thus, for this data we did not use the initial
 885 ActNorm layer. All networks for those comparisons had 3 blocks with a width of 5 (i.e. using $K = 5$
 886 frequencies) and were trained for 600 iterations with a learning rate of $1e^{-3}$. In all of these, the
 887 prototypes were the oscillator prototype defined in Equation (3), with $a = \pm 0.25$ and $\omega = \pm 0.5$.
 888

889 **Other results on 2D simulated datasets:** for figures Figure 2 and Figure 3, all networks had 4
 890 blocks with a width of 5, and were trained for 500 iterations. For these, we used a learning rate
 891 and weight decay of $1e^{-3}$, and the determinant regularization coefficient was $1e^{-3}$. As before, the
 892 prototypes used for these were the oscillator prototype with $a = \pm 0.25$ and $\omega = \pm 0.5$.

893 **Repressilator system:** for Figure 4, we used NFs with 3 blocks and a width of 5. These were
 894 trained for 500 iterations, with the same regularization settings as above. Additionally, to fit the
 895 systems in more than 2D, we added the projection regularization from Appendix A.6, with the
 896 initial regularization $\lambda_{\text{proj}} = e^{-1}$. These were trained for 2000 iterations, and were matched to
 897 the high-dimensional versions of the oscillator prototypes from Equation (4), with $a = \pm 0.25$ and
 898 $\omega = \pm 2$.

899 For the hyperparameters used in the cell-cycle experiments, see Appendix D.

900 As our INNs are quite small, and most of the data is of a low dimension (relative to typical machine
 901 learning tasks), a single CPU with 16GB RAM was sufficient for training our networks. Fitting a
 902 single INN takes less than 10 seconds for all of the 2D simulated data and around 90-120 seconds for
 903 the cell-cycle data.

904 A.8 EFFECTS OF HYPERPARAMETERS ON PERFORMANCE

905 We found that the classification results by SPE are robust to the choice of number of frequencies and
 906 layers, as demonstrated in the table below.

907 In our experiments, we observed that beyond the robust range of SPE demonstrated in Table 2,
 908 when many frequencies are used, the networks tend to overfit to overly complex estimations of the
 909 invariant set, which are typically not faithful to the true invariant set. These effects can be seen in
 910 Figure 6. However, when using too few frequencies and layers, the network is not expressive enough
 911 to capture non-elliptical invariant sets. For our experiments in the main text, we chose 4 layers and 5
 912 frequencies as a good balance between classification performance, quality of invariant set localization,
 913 and computational efficiency.

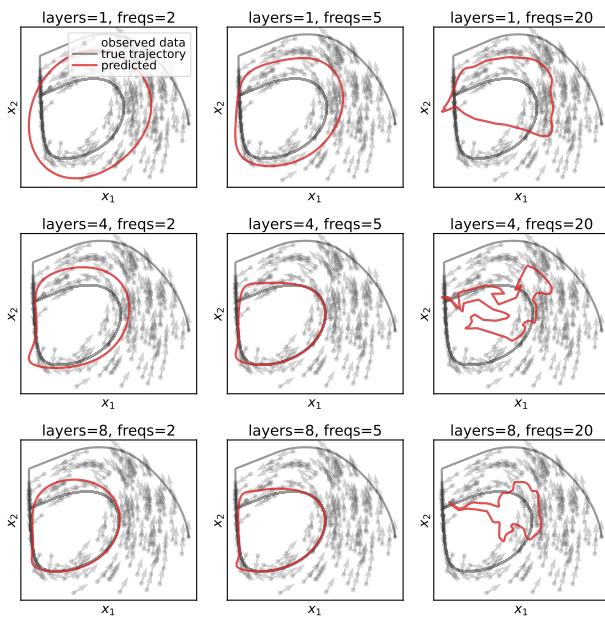


Figure 6: **Estimation of limit cycles using different numbers of layers and frequencies.** Examples of invariant sets predicted (in red) from observed vectors (in gray), for different dynamical systems which exhibit a limit cycle attractor. For each system, a ground-truth trajectory was simulated from the hidden system and plotted in black. The red curves are the limit cycles predicted using SPE.

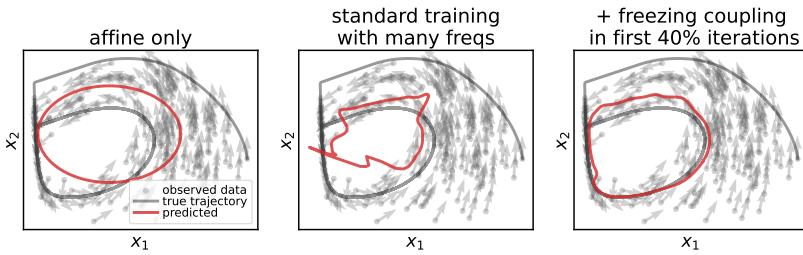


Figure 7: **Initializing network from the solution of an affine-only diffeomorphism improves fitting.** **Left:** when the invariant set is not an ellipse, using a diffeomorphism defined only by an affine transformation does not faithfully capture the underlying invariant set. **Middle:** if too many frequencies are used with SPE, the model overfits. **Right:** when initializing the model with the result from an affine transformation, SPE converges more faithfully to the true invariant set, even when using many frequencies in the transformation.

Table 2: **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 from Table 3, with $N = 250$ observations per system.

freqencies	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%

If many frequencies are needed, we found it beneficial to begin training with frozen coupling layers, as shown in Figure 7. 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.

B SIMULATION DETAILS

B.1 2D SYSTEMS

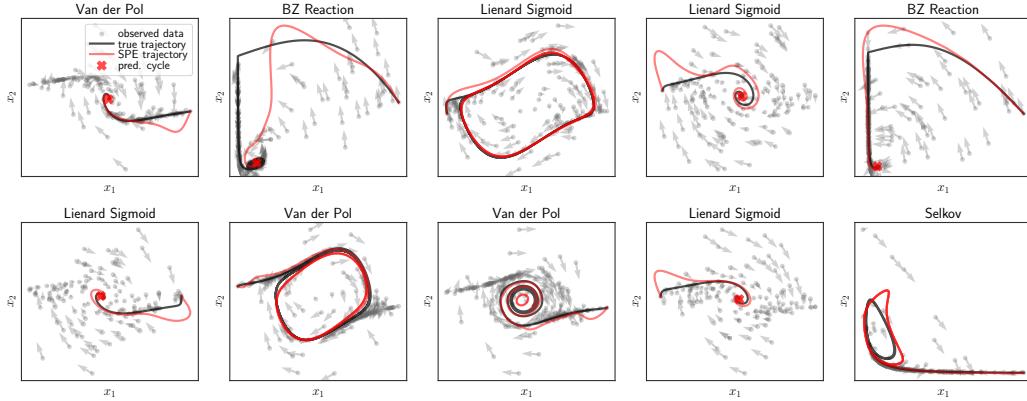


Figure 8: **Randomly picked examples of SPE on 2D synthetic data.** In all of these, notice how SPE is able to localize the invariant set from the data, even when there might be uncertainty regarding the intermediate trajectories.

For the 2D simulated dynamics, we use the same systems as in Moriel et al.. The governing equations for these systems are shown in Table 3 (adapted from Moriel et al. 2023). We additionally study the

Table 3: Governing equations, range of phase space used for initial conditions, range of parameters for simulated systems and parameter settings that are known to correspond with a limit cycle. Adapted from Moriel et al. 2023.

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

Augmented SO system, which is an instance of a simple oscillator (SO) that was transformed by a random initialization of a neural spline flows Durkan et al. (2019a). All of these systems change behavior from node attractors to limit cycles, i.e. undergo a Hopf bifurcation, at known parametric settings, making them particularly relevant for our application.

See Figure 8 for randomly sampled systems, their behavior and the fitting quality of SPE.

B.2 SIMULATING SPARSE DATA

The most straightforward method to draw sparse data from each system is to draw points uniformly from the phase-space. However, for high-dimensional systems this runs the risk of under-sampling the region of the invariant set, observations which we require in order to make any meaningful prediction. Moreover, this does not accurately portray how we expect data to be collected in real scenarios. Instead, we assume that each point of the data is a point on a random trajectory, with initial conditions coming from a uniform distribution on pre-specified ranges of the phase-space. Additionally, we assume that each observed pair (x_i, \dot{x}_i) has independently traversed a different amount of time on its respective trajectory. The data sampling process can be summarized through the following steps:

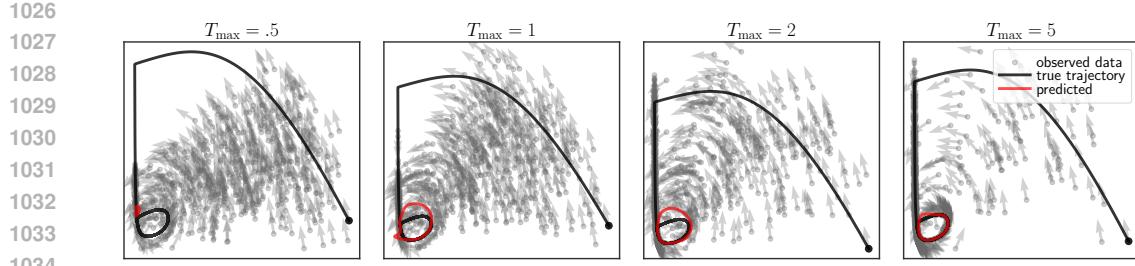


Figure 9: The effects of changing the maximum simulation sampling time T_{\max} . When T_{\max} is small (e.g. $T_{\max} = 0.5$; left), most of the observations are transients. On the other hand, when T_{\max} is large (e.g. $T_{\max} = 5$; right), most of the observations are on the invariant set.

1. For each $i \in \{1, \dots, N\}$, draw the initial conditions $x_i^{(0)}$ uniformly from a pre-specified portion of space
2. Sample the amount of time t_i uniformly in the range $[T_{\min}, T_{\max}]$
3. Simulate a trajectory beginning at $x_i^{(0)}$ to time t_i . The point x_i reached on the trajectory in this manner is defined as the position, and $\dot{x}_i = f(x_i)$ as the velocity
4. Optionally, Gaussian noise with standard deviation σ is then added to \dot{x}_i

Usually we set the parameter $T_{\min} = 0$, except when sampling from the repressilator, where it is set to $T_{\min} = 3$. T_{\max} is a hyperparameter that effectively controls how much samples tend to concentrate around the invariant set. When T_{\max} is close to 0, then the pairs (x_i, \dot{x}_i) revert back to being uniformly sampled from phase-space.

An example of the effect of different values for T_{\max} can be seen in Figure 9. When T_{\max} is small, the observations mostly include transients of the dynamics, in which case not much can be estimated with regards to the invariant set of the system. As T_{\max} increases, more points aggregate near the invariant set. For our 2D simulations we use $T_{\max} = 3$, whereas for the repressilator system we used $T_{\max} = 10$.

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

B.3 REPRESSILATOR

The repressilator system is governed by six coupled differential equations, structured as three instances of the following gene-regulation model (Elowitz & Leibler, 2000):

$$\begin{aligned} \dot{m}_i &= -m_i + \frac{\alpha}{1 + p_j^n} + \alpha_0, \\ \dot{p}_i &= -\beta(p_i - m_i), \end{aligned}$$

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: $i = \text{LacI}, \text{TetR}, \text{cI}$ and $j = \text{cI}, \text{LacI}, \text{TetR}$. The parameters α_0 (basal transcription), α (maximal transcription), β (protein/mRNA degradation ratio), 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 (Verdugo, 2018b):

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

1080 where:

1081
$$A = \frac{-\alpha n \hat{p}^{(n-1)}}{(1 + \hat{p}^n)^2}, \quad \hat{p} = \frac{\alpha}{1 + \hat{p}^n} + \alpha_0 \quad (15)$$
 1082

1083 These are the boundaries plotted in Figure 4 (top).

1085 To simulate instances of the repressilator, we use the same sampling procedure as described in
1086 Appendix B.2, with $T_{\min} = 3$ and $T_{\max} = 10$. In this context we use $T_{\min} = 3$ to ensure that
1087 when the samples are projected to lower than six dimensions they still define a vector field with
1088 non-intersecting trajectories. Furthermore, even though we used $T_{\max} = 10$, a larger value than in
1089 the other simulated systems, we found that the observed points do not all lie on the invariant set of
1090 the repressilator (as can be seen in Figure 4).1091

C BASELINE DETAILS

1092

C.1 BASELINES IN INVARIANT SET LOCALIZATION

1093 In Section 4.1 we showcase the ability of SPE at estimating the position of limit cycles from
1094 observations. Few existing approaches to do so exist, and those that do rely on data which is dense
1095 on a regular grid. However, the quality of the fit can be evaluated compared to other methods for
1096 estimating the underlying ODE. Accordingly, we compared SPE to a number of simple baselines:

- 1097
- 1098 • **kNN**, a simple interpolator which uses the k -nearest observed positions in order to estimate
1099 the velocity. In our experiments, we found that $k = 5$ neighbors approximates clean systems
1100 fairly well, and used that setting for all of our comparisons.
 - 1101 • **SINDy (Brunton et al., 2016; de Silva et al., 2020)** is an established method for estimating
1102 vector fields from empirical observations. SINDy uses sparse linear regression with a fixed
1103 library of functions to estimate the velocities of the system. For our evaluations we use the
1104 `pysindy` package (de Silva et al., 2020) and consider a function library with 3rd order
1105 polynomials (SINDy-poly) or comprised of 10 Fourier features (SINDy-fourier). All other
1106 hyperparameters used were those that are the default in the `pysindy` package.
 - 1107 • **MLP** is a simple multi-layer perceptron with 2 hidden layers, with a width of 128 and
1108 SiLU activations (Hendrycks & Gimpel, 2016). These MLPs were trained to estimate the
1109 velocities from the positions using an MSE loss over the predicted velocities, with the Adam
1110 optimizer (Kingma, 2014), for 5000 (full-batch) iterations. We used a learning rate of 0.001
1111 and a weight decay of 0.001. These hyperparameters were chosen as MLPs trained in such
1112 a manner accurately approximate clean systems with $N > 250$ observations.

1113 All of these methods form as alternatives for approximating the underlying ODE from sparse
1114 observations. However, unlike SPE, these methods cannot directly localize the invariant set of the
1115 observed (or even, fitted) system. Even when the fit is very accurate, determining where and what
1116 kind of invariant set was learned is a very difficult problem. Instead, we opt to compare the long-term
1117 performance of SPE against all of these baselines, fully described in the next section.1118

C.2 EVALUATION OF INVARIANT SET LOCALIZATION

1119 To quantitatively evaluate the localization of the invariant set, for SPE and the other baselines
1120 considered, we simulate long trajectories from fixed initial conditions in both the true (hidden) system
1121 and in the fitted systems. When the initial conditions are sampled from some distribution, these long
1122 trajectories give an effective stationary distribution of points along the invariant set of the considered
1123 systems. Accordingly, our goodness-of-fit measure is the Wasserstein-2 (W_2) distance between
1124 effective stationary distributions of the hidden governing equations and the fitted ODEs.1125 For the evaluations in Figure 3 (left), we randomly sampled 1000 systems which contain limit cycle
1126 behaviors. Systems were sampled uniformly from the SO, Liénard Polyomial, Liénard Sigmoid,
1127 Van der Pol, BZ Reaction and Selkov families of systems (Appendix B.1), with order parameters
1128 uniformly sampled from the ranges corresponding to limit cycle behaviors. For each system, different
1129 numbers N of positions x_i and velocities \dot{x}_i were sampled according to the procedure described in
1130 Appendix B.2, with $T_{\max} = 3$. Additive Gaussian noise was then added to these velocities with a

standard deviation of $\sigma = \mu/\text{SNR}$ where $\mu = \frac{1}{N} \sum_i \|\dot{x}_i\|$ and a prespecified signal-to-noise ration (SNR). Adding noise in this manner allows for direct control of the SNR of the observed velocities between different families of systems, whose phase spaces and velocities are defined on a range of sizes. Each of the methods, SPE and the baselines described in Appendix C.1, were then fitted to all N simulated positions x_i and noisy velocities \dot{x}_i in order to approximate the underlying ODE.

Each method was then evaluated by comparing their 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 estimated ODE, 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 RK4, with a step size of $\Delta t = 0.01$, for $T_{\text{eval}} = 100$ which we found is long enough for the true system to always be close to its invariant set. To calculate the W_2 distance we utilize the Python Optimal Package from Flamary et al. 2021.

C.3 BASELINES IN CLASSIFICATION

To benchmark SPE in the classification of dynamical systems, we turn to datasets previously considered by Moriel et al. 2023 and compare to the same methods. For all of these baselines, we use exactly the same settings as described in Section 7.4 of Moriel et al. 2023. Below we provide a brief summary of each of these methods:

- **TWA** (Moriel et al., 2023), 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 in each position. TWA is trained on entirely simulated data, which is generated by sampling the order parameters from the SO family of systems. These are then augmented using neural spline flows (Durkan et al., 2019b), INNs. 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.
- **Critical Points**, is based off of an algorithm devised by Helman & Hesslink 1989 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 repeler, and is otherwise classified as an attractor.
- **Phase2vec** is 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 output of the decoder are the predicted coefficients of the parameters, which are multiplied by the polynomial function library in order 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.
- **Autoencoder** is 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 in order 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.

Beyond accuracy, we also compared the F1, precision and recall scores of SPE versus each of the above methods. These comparisons are summarized in Table 4 and Table 5. In all families of systems except SO and Aug. SO, SPE most consistently achieves the highest scores.

C.4 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 Selkov families of systems. For each system, 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

1188

1189
1190
1191
Table 4: Comparison of the F1/Precision/Recall scores for all methods on the classification task
from Moriel et al. 2023 (continued in Table 5). The highest numbers for each column and score are
shown in bold.

	SO	Aug. SO	Lienard Poly	Lienard Sigmoid
SPE	90±1/ 100±0 /82±2	79±1/88±1/71±2	96±1/100±1/92±2	93±1/88±2/98±1
TWA	97±1 /97±2/97±3	93±1/92±2/94±2	85±22/ 100±0 /78±23	90±13/ 100±1 /84±17
Phase2Vec	62±17/ 100±1 /47±18	60±10/ 92±2 /46±11	7±22/10±29/8±25	0±0/0±0/0±0
Autoencoder	95±2/93±5/ 99±3	87±1/84±3/92±4	55±37/54±37/64±43	84±18/100±1/75±21
Critical Points	17/87/9	34/67/23	59/96/42	7/100/4

1192

1193

1194

1195

1196

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
Table 5: Continuation of Table 4.

	Van der Pol	BZ reaction	Selkov
SPE	97±3/95±5/100±0	87±2/100±0/77±3	66±2/98±1/50±2
TWA	80±26/96±20/71±25	85±7/78±11/ 96±9	49±10/ 98±5 /34±11
Phase2Vec	4±12/20±40/3±8	0±0/0±0/0±0	0±0/0±0/0±0
Autoencoder	74±31/89±26/71±34	81±17/87±13/84±24	3±8/12±32/2±5
Critical Points	33/ 98 /20	87/79/ 96	15/62/9

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 $\sigma = \mu/\text{SNR}$ where $\mu = \frac{1}{N} \sum_i \|\dot{x}_i\|$ and a prespecified signal-to-noise ration (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.

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

We analyzed single-cell RNA-sequencing data from U2OS cell line that was integrated with the fluorescent ubiquitination-based cell cycle indicator (FUCCI) in order to investigate the spatiotemporal dynamics of human cell cycle expression (data collected in (Mahdessian et al., 2021)). The original dataset included 1,152 cells characterized by the expression of 58,884 genes. After preprocessing, following the protocol in Zheng et al. (2023) — which involved gene and cell filtering, normalization, log transformation, and the selection of highly variable genes — 1,021 cells remained, described by 2000 genes.

To infer local measures of cellular dynamics in the high-dimensional gene expression space, we used scVelo (Bergen et al., 2020) which computes RNA velocity vectors 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 and to infer the latent transcriptional state of each cell.

The gene expression data and the inferred velocities were projected into a 100-dimensional space using principal component analysis (PCA). We chose to project the data into a 100-dimensional space in order to reduce the computational load of our method. Streamlines of the resulting velocity field are visualized in Figure 5, 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 Tirosh et al. (2016), using the function “score_genes_cell_cycle” from scVelo package (Bergen et al., 2020). The color in Figure 5, left and middle, represents the fractional component of the S-phase score relative to the combined S and G2/M scores.

We then applied our method to fit the RNA velocities, embedded in the 100-dimensional PCA space, to a 100-dimensional cyclic oscillator prototype, as defined in Equation (4) of Section 4.3. 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 $\text{wd} = 1e^{-3}$, projection regularization set to $\text{proj_reg} = e^{-2}$, determinant regularization of 0, a learning rate of $\text{lr} = 5e^{-4}$, and 1,000 iterations.

Given the fitted coordinates of the cycle attractor in the 100-dimensional PCA space, we inverted the PCA transformation to reconstruct the corresponding predicted gene expression. In Figure 5 (right), we plot representative marker genes for both the S-phase (top half of the heatmap) and G2/M-phase (bottom half) taken from Tirosh et al. 2016, showing distinct oscillatory patterns consistent with cell cycle progression.

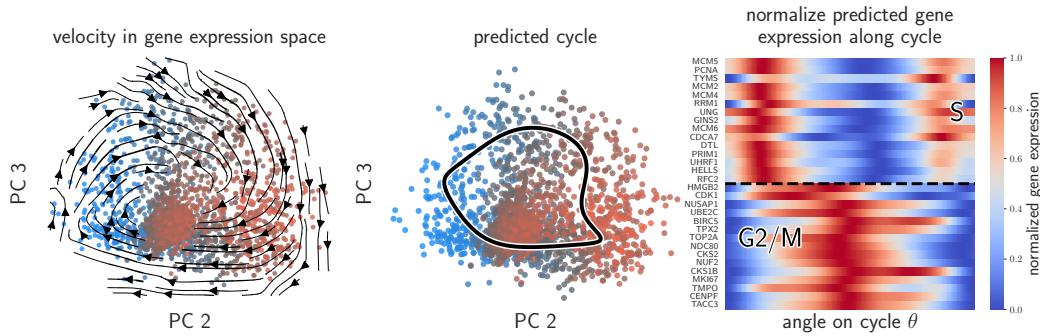


Figure 10: **Fitting of the cell cycle in human fibroblasts (Riba et al., 2022).** **Left:** gene expression velocity of cells undergoing proliferation in PCA space, as in Figure 5. PCs 2 and 3 are shown as the cycle is more apparent in this projection. **Middle:** the SPE localization of the cycle attractor. **Right:** normalized gene expressions along the cycle attractor as predicted by SPE. The heatmap depicts a set of common marker genes for the S (top of the heatmap) and G2/M (bottom of the heatmap) phases.

These qualitative results are not unique to the FUCCI dataset. Using the same preprocessing, we fit an alternative dataset of human fibroblast cells during their cell cycle (Riba et al., 2022) using 10 PCs, which results in qualitatively similar performance, as can be seen in Figure 10. 10 PCs were used in this dataset to reduce computational load.

E THEORETICAL TREATMENT

As a proof of correctness for the use of the equivalence loss, we will consider small perturbations to the true underlying field and the implications of these perturbations for the equivalence loss. We will then show how our diffeomorphisms imply that these perturbations are C^1 -close to the original field (Guckenheimer & Holmes, 2013). This, together with an assumption regarding the stability of the underlying field, will demonstrate a correspondence between small equivalence loss and the recovery of the true invariant set.

E.1 CORRECTNESS OF EQUIVALENCE LOSS

We will analyze the effects of small perturbations of the diffeomorphism from its optimal form:

$$H_\delta(x) = H^*(x) + \delta H(x) \quad (16)$$

where $H^*(x)$ is the optimal diffeomorphism under which the prototype $g(y)$ and the vector field $f(x)$ are equivalent, $H(x)$ is a diffeomorphism that perturbs $H_\delta(x)$ from the optimum, and $\delta > 0$ is a small positive constant that represents the magnitude of the perturbation. As described, we will look at the case when $f(x)$ and $g(y)$ are smoothly equivalent, meaning:

$$\partial_x H^*(x) f(x) = g(H^*(x)) \Leftrightarrow f \stackrel{H^*}{\sim} g \quad (17)$$

Furthermore, we will analyze an unnormalized form of the equivalence loss from Equation (2):

$$\mathbb{E}_x \left[\|\partial_x H_\delta(x) f(x) - g(H_\delta(x))\|^2 \right] \quad (18)$$

Under our weak perturbations, we have:

$$\partial_x H_\delta(x) f(x) = \partial_x H^*(x) f(x) + \delta \partial_x H(x) f(x) \quad (19)$$

$$= g(H^*(x)) + \delta \partial_x H(x) f(x) \quad (20)$$

We will further assume that $g \in C^2$ on the data domain, i.e. that it has bounded second derivatives. Then, approximating $g(H_\delta(x))$ with its Taylor expansion around $H^*(x)$:

$$g(H_\delta(x)) = g(H^*(x) + \delta H(x)) \quad (21)$$

$$= g(H^*(x)) + \delta \cdot \partial_x g(H^*(x)) H(x) + \mathcal{O}(\delta^2) \quad (22)$$

Under these conditions, the unnormalized equivalence loss is equal to:

$$\mathbb{E}_x [\| [\partial_x H_\delta(x)] f(x) - g(H_\delta(x)) \|^2] = \mathbb{E}_x [\| \delta ([\partial_x H(x)] f(x) - [\partial_x g(H^*(x))] H(x)) + \mathcal{O}(\delta^2) \|^2] \quad (23)$$

$$= \delta^2 \cdot \mathbb{E}_x [\| [\partial_x H(x)] f(x) - [\partial_x g(H^*(x))] H(x) + \mathcal{O}(\delta) \|^2] \quad (24)$$

So, our equivalence scales together with the size of the perturbations. In other words, optimizing diffeomorphisms to reduce the equivalence loss corresponds with optimizing the diffeomorphism to remove the effects of the added perturbation, $\delta H(x)$.

E.2 IMPLICATIONS REGARDING TRUE INVARIANT SET

In the above, we showed that the equivalence loss shrinks as the size of the perturbation from the optimal diffeomorphism becomes smaller. This was true for any perturbation from the optimum, but does not necessarily mean on its own that the fitted invariant set is recapitulated. We will now move our attention to this matter.

To be able to faithfully reconstruct the true invariant set, we will only consider vector fields which are structurally stable (Guckenheimer & Holmes, 2013), in the sense that their invariant sets are retained under small C^1 -perturbations. In our scenario, the vector field modeled by SPE is a function of both the prototype and the fitted diffeomorphism, and is given by:

$$f_H(x) = [\partial_x H_\delta(x)]^{-1} g(H_\delta(x)) \quad (25)$$

$$= [\partial_x H^*(x) + \delta \partial_x H(x)]^{-1} g(H_\delta(x)) \quad (26)$$

$$= [\partial_x H^*(x) + \delta \partial_x H(x)]^{-1} [g(H^*(x)) + \delta \partial_x g(H^*(x)) H(x) + \mathcal{O}(\delta^2)] \quad (27)$$

$$= \left[I + \delta (\partial_x H^*(x))^{-1} \partial_x H(x) \right]^{-1} \left[f(x) + \delta (\partial_x H^*(x))^{-1} \partial_x g(H^*(x)) H(x) + \mathcal{O}(\delta^2) \right] \quad (28)$$

where the final step is due to the equality (through smooth equivalence):

$$f \stackrel{H^*}{\sim} g \Leftrightarrow f(x) = [\partial_x H^*(x)]^{-1} g(H^*(x)) \quad (29)$$

The above can be further expanded through the Neuman series when the eigenvalues of $[\partial_x H^*(x)]^{-1} \partial_x H(x)$ are bounded and δ is small enough such that $\delta |\lambda_i([\partial_x H^*(x)]^{-1} \partial_x H(x))| < 1$ with $\lambda_i(A)$ the eigenvalues of A (Petersen et al., 2008):

$$\left(I + \delta (\partial_x H^*(x))^{-1} \partial_x H(x) \right)^{-1} = \sum_{\ell=0}^{\infty} (-\delta)^\ell \left[(\partial_x H^*(x))^{-1} \partial_x H(x) \right]^\ell \quad (30)$$

$$= I + \sum_{\ell=1}^{\infty} (-\delta)^\ell \left[(\partial_x H^*(x))^{-1} \partial_x H(x) \right]^\ell \quad (31)$$

$$= I - \delta (\partial_x H^*(x))^{-1} \partial_x H(x) + \mathcal{O}(\delta^2) \quad (32)$$

Plugging this in, the perturbed vector field is given by:

$$f_H(x) = \left(I - \delta (\partial_x H^*(x))^{-1} \partial_x H(x) + \mathcal{O}(\delta^2) \right) \left[f(x) + \delta (\partial_x H^*(x))^{-1} \partial_x g(H^*(x)) H(x) + \mathcal{O}(\delta^2) \right] \quad (33)$$

$$= f(x) + \delta \cdot (\partial_x H^*(x))^{-1} [\partial_x g(H^*(x)) H(x) - \partial_x H(x) f(x)] + \mathcal{O}(\delta^2) \quad (34)$$

1350 The perturbation $\epsilon(\delta) = f(x) - f_H(x)$ from $f(x)$ is thus equal to:
 1351

$$1352 \quad \epsilon(\delta) = \delta \cdot (\partial_x H^*(x))^{-1} [\partial_x g(H^*(x))H(x) - \partial_x H(x)f(x)] + \mathcal{O}(\delta^2) \quad (35)$$

1353
 1354 All elements of this perturbation are scaled by δ . Thus, when all elements of the Jacobian of both
 1355 $H(x)$, $H^*(x)$ and $f(x)$ are bounded in the region of the data, then $f_H(x)$ is a C^0 perturbation of
 1356 $f(x)$ in the limit $\delta \rightarrow 0$, the range where our equivalence loss is low. In addition to the above, C^1
 1357 perturbations, defined by the magnitude of the partial derivatives $\partial\epsilon(\delta)/\partial x_i$, can also be guaranteed
 1358 under specific conditions. In particular, the partial derivatives of the perturbation $\epsilon(\delta)$ with respect
 1359 to the inputs x_i will depend on all partial derivatives up to order 2 and their products of both the
 1360 diffeomorphism, $\partial^2 H(x)/\partial x_i \partial x_j$, the optimal diffeomorphism, the prototype, and the underlying
 1361 vector field. Therefore, if all of these derivatives, up to the second order, are bounded from below and
 1362 above by a constant, then C^1 closeness will also be guaranteed in the limit of $\delta \rightarrow 0$.

1363 Under these conditions then, as shown above, $f(x)$ and $f_H(x)$ are C^1 -close at $\delta \rightarrow 0$. Since $f(x)$
 1364 was assumed to be structurally stable, this means that $f(x)$ and $f_H(x)$ share the same invariant sets
 1365 (Guckenheimer & Holmes, 2013).

1366 **To summarize,** consider the following constraints:
 1367

- 1369 • The prototype, $g(y)$, is smoothly equivalent to the underlying vector field, $f(x)$. Denote by
 1370 $H^*(x)$ a diffeomorphism such that $f \xrightarrow{H^*} g$.
- 1371 • In the compact space containing all observed data points, the partial derivatives of up to
 1372 order 2 for $f(x)$, $g(y)$, $H^*(x)$ and $H(x)$ are bounded from above and below by a constant.
- 1373 • In the same compact space, the vector field $f(x)$ is structurally stable (and thus, $g(y)$ is also
 1374 structurally stable).

1375
 1376 Under the above conditions, we have:
 1377

- 1378 1. Minimizing the equivalence loss defined in Equation (2) pushes to decrease the amount of
 1379 perturbation from the optimal diffeomorphism, and subsequently the perturbation from the
 1380 true vector field.
- 1381 2. Under small perturbations, the true vector field and the modeled fields are C^1 close.
- 1382 3. The above, together with the assumption that $f(x)$ is structurally stable, means that when
 1383 the equivalence loss is low, the true invariant sets of the dynamics are faithfully estimated.

1386 F FURTHER EXPERIMENTS

1388 F.1 MULTIPLE BASINS OF ATTRACTION

1389 While this work mostly focused on prototypes with only a single basin of attraction, such as a node
 1390 attractor or limit cycle, SPE can be used when there are multiple basins of attraction.

1392 As a simple example, we looked at synthetic dynamics with multiple fixed points in a high-dimensional
 1393 space. The governing equations for these dynamics were based on a mixture of experts decomposition
 1394 of space, where each expert is a basin of attraction. Specifically, the governing equations of the
 1395 dynamics were defined as:
 1396

$$1397 \quad f(x) = \sum_{k=1}^K r_k(x) \varphi_k(x - c_k) \quad (36)$$

$$1400 \quad r_k(x) = \frac{\exp\left[-\frac{1}{2s^2}\|x - c_k\|^2\right]}{\sum_{k'} \exp\left[-\frac{1}{2s^2}\|x - c_{k'}\|^2\right]} \quad (37)$$

1403 where $\varphi_k(\cdot)$ are the local dynamics around the center $c_k \in \mathbb{R}^d$, s determines the strength with which
 one basin of attraction transitions to another, and K is the number of basins of attraction.

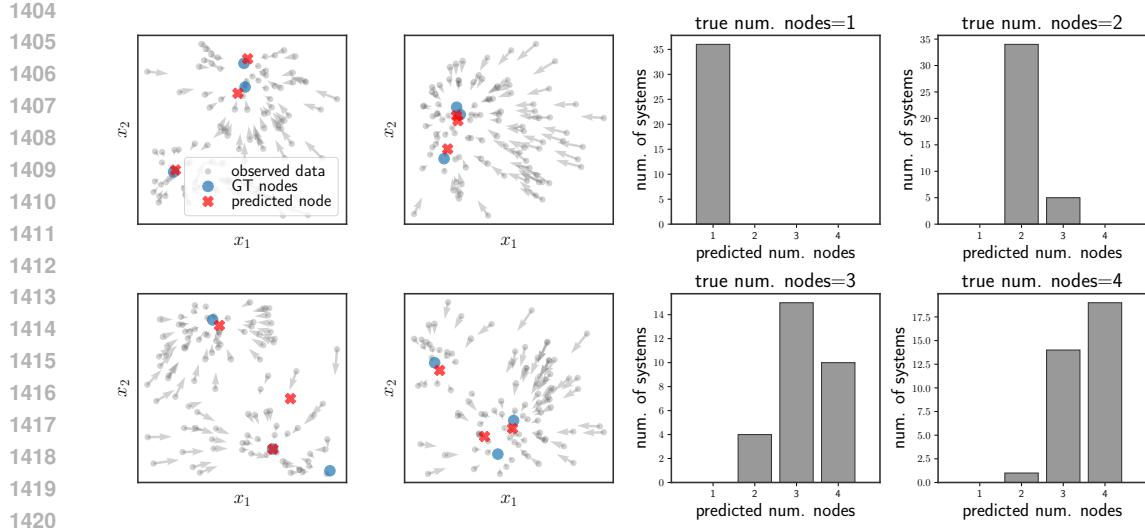


Figure 11: Classifying and localizing invariant sets in systems with multiple basins of attraction. **Left:** 4 examples of fitting SPE to a system with multiple basins of attraction in 3 dimensions. In each example, the positions of the node attractors was randomly sampled (these positions are depicted 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 between this data and a fixed prototype with 3 basins of attraction. The estimated positions of the attractors according to SPE is 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.

Under this setting, we considered systems whose centers were sampled uniformly from the range $[-0.9, 0.9]^d$, all of which were node attractors:

$$\varphi_k(y) = -y \quad (38)$$

Finally, the number of attractors was randomly chosen to be between 1 and 4. We then fit SPE with 4 different prototypes, for each number of possible basins of attraction. The positions of the nodes in the prototypes were fixed and distinct from the true underlying system, so that the task was both the prediction of the location of the nodes as well as the estimation of their number. Results can be seen in Figure 11 for systems with 100 points.

Throughout the experiments shown in Figure 11, INNs with 2 blocks of width 3 were used, together with 5 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 0. The classification experiment was carried out in 5D.

F.2 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, we expect SPE to capture the *dominant behavior* of the system. To test this, we looked at the families of systems from Appendix B.1, and simulated data from oscillatory systems. As the limit cycle of these systems is bounded to a finite region, the dominant behavior of the system as a whole is determined by the orientation of the flow - either clockwise or counter-clockwise. To emulate a mismatch between the chosen prototype and the underlying behavior, we fit SPE using only node attractor prototypes, as opposed to the true underlying oscillatory behavior, using the same prototypes from Equation (3), with $a = -0.25$ and $\omega = \pm 0.5$. Figure 12 demonstrates that even when there is a mismatch between the underlying dynamics and the prototype, SPE is able to capture the dominant behavior of the

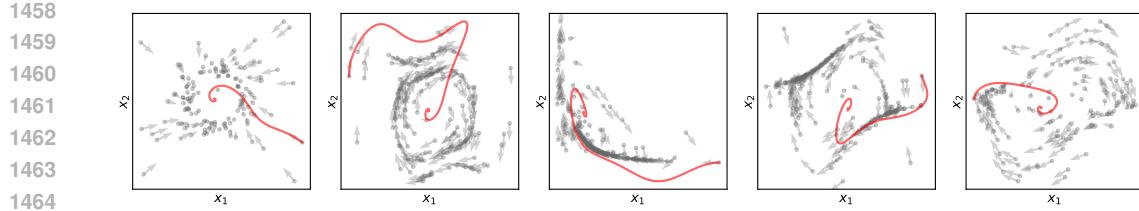


Figure 12: **Direction of angular velocity is captured, even when there is a mismatch between the prototype and the true underlying system.** Randomly sampled systems and examples of fitting SPE to an inherently oscillatory process using a node attractor prototype. Despite this mismatch between underlying 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.

system - that is, the orientation of the flow is matched. Quantitatively, we find that the average accuracy when trying to predict the orientation of the SO family of systems was 98%, averaged over 100 trials using the same hyperparameters described in Appendix A.7.

This demonstration illustrates how SPE might be used, even when the prototype does not fully match the ground-truth behavior. This matches scientific scenarios, where there is usually some domain expertise as to the dominant behavior of an observed system, but this knowledge may not capture all subtleties of the empirically collected data. In such scenarios, it can still be beneficial to match between an analytical model (i.e. the prototype) and the data for scientific discovery.

F.3 LIMIT CYCLE PROTOTYPES FROM ALTERNATIVE OSCILLATORS

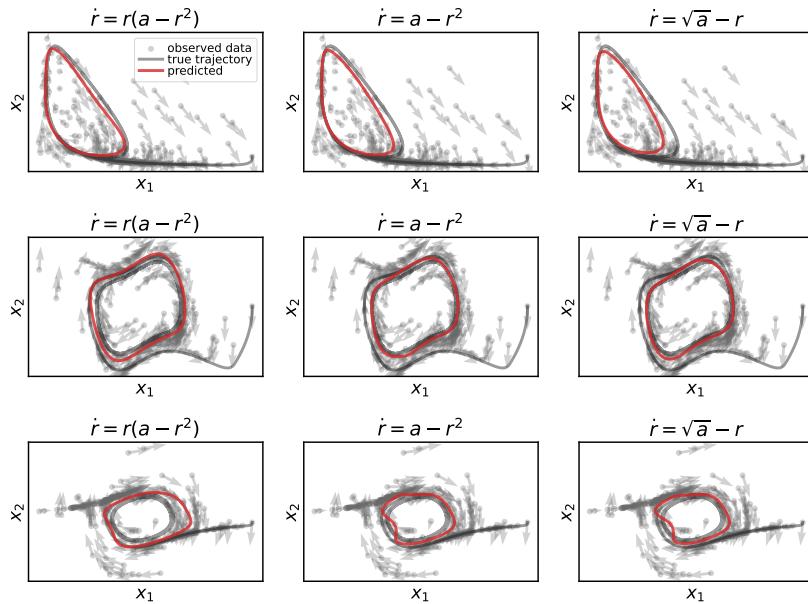
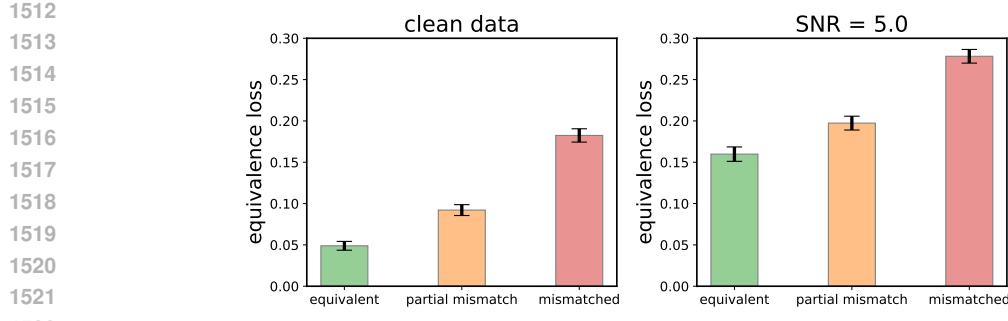


Figure 13: **Estimation of limit cycles using different oscillator prototypes.** The three rows correspond to different possible cycle prototypes. The left-most column corresponds with the prototypes used in the rest of this work, and the middle and right-most columns have decreasing speeds towards the limit cycle in the radial direction. The three prototypes result in similar estimated invariant sets.

Throughout the main text, the form of the prototypes we used were:

$$\dot{r} = r(a - r^2), \quad \dot{\theta} = \omega \quad (39)$$



1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565

Figure 14: **The equivalence loss can be used as a measure of prototype mismatch**, for both clean (left) or noisy (right) data. The equivalence is averaged over multiple instances of the systems from Table 3, fitted with the node or cycle prototypes, which can either be partially mismatched with the data (when the angular velocity matches, but not the invariant set), or fully mismatched. The error bars represent the standard error of the mean of the average equivalence loss.

where r and the θ are polar coordinates. We chose this prototype as it is a well-studied form of a supercritical Hopf system. However, SPE is robust to the use of other governing equations which result in limit cycles, as these systems are smoothly equivalent to each other.

Two alternative examples for oscillators which can be used as prototypes are:

$$\dot{r} = a - r^2, \quad \dot{\theta} = \omega \quad (40)$$

$$\dot{r} = \sqrt{a} - r, \quad \dot{\theta} = \omega \quad (41)$$

These two systems are qualitatively equivalent to each other and to Equation (39). These systems exhibit different speeds in the radial direction towards the radius of the limit cycle, but otherwise remain qualitatively similar. Using SPE with these prototypes results in very similar estimated invariant sets, as shown in Figure 13.

F.4 DETECTING PROTOTYPE MISMATCH

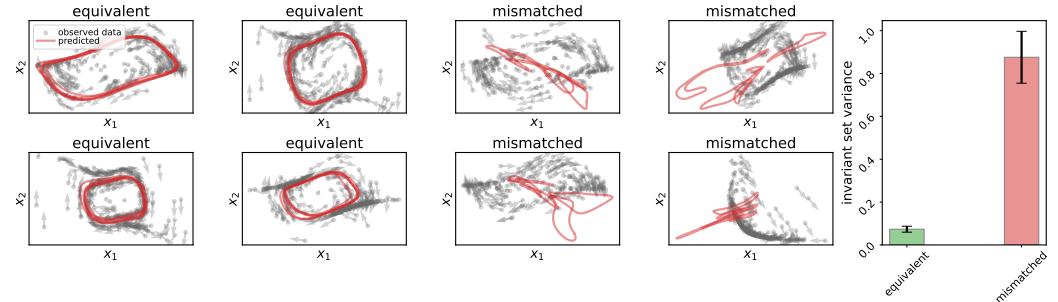


Figure 15: **Effects of mismatch between the true system and the prototype on the estimated invariant set.** **Left:** when the two systems are equivalent, the location of the estimated invariant set is robust to randomly subsampling the positions. **Middle:** on the other hand, when there is a mismatch between the prototype and system, the diffeomorphism tends to overfit. In such scenarios, for every random subsampling of the data, a dramatically different limit cycle is estimated. **Right:** the variance, average over the whole invariant set, in the estimated positions is enough to differentiate between equivalent and mismatched prototype-true system pairs. The error bars represent the standard error of the mean of the variance.

We expect the equivalence loss from Equation (2) to correspond with the level of mismatch between the true underlying vector field and the prototype. As shown in Table 1, this allows for accurate classification between possible prototypical systems. Beyond this, the loss after optimization can be

1566 used as a measure of the mismatch between prototype and data; To demonstrate this, we examined
 1567 300 instances of the simulated systems from Table 3, corresponding to both cycles and node attractors.
 1568 We then randomly assigned the sign of the angular velocity of the prototype, ω , and radius of the
 1569 limit cycle a (where negative values correspond to node attractors). In this setting, systems can be
 1570 either: (1) equivalent (angular velocity and radius match), (2) partially mismatched (angular velocity
 1571 matches, but radius does not), (3) mismatched (both angular velocity and radius are mismatched
 1572 relative to the true system). For systems that are equivalent, setting (1), we expect a small loss. For a
 1573 partial mismatch, which was qualitatively analyzed in Appendix F.2, we expect the loss to be larger
 1574 than the equivalent systems. Finally, when the two systems are mismatched, we expect the loss to be
 1575 larger still, accounting for the error in the direction of the vectors for the whole space. This expected
 1576 trend is what is demonstrated in practice, as shown in Figure 14 (left). The average loss is lowest for
 1577 systems that are equivalent to each other, and highest for systems that are mismatched. This trend
 1578 holds for noisy data (SNR = 5), as shown in Figure 14 (right).

1579 Another indication for a possible mismatch between the prototype and the underlying system can be
 1580 identified by the variance of the estimated invariant set under multiple initializations. When the true
 1581 system and the prototype are mismatched, we find that the parameterized diffeomorphism overfits
 1582 to the data, whereas equivalent systems are robust to small changes in the data. In particular, when
 1583 there is a mismatch, the estimated invariant set changes dramatically under a random subsampling
 1584 of the data. Examples of this behavior can be seen in Figure 15, where 5 different diffeomorphisms
 1585 were fit to 5 random splits of the data, in which only 50% of the data was kept in each split. When
 1586 the prototype and underlying system are equivalent, the fitted invariant set is robust to the particular
 1587 split (Figure 15, left). On the other hand, when there is a mismatch, the position and shape of the
 1588 invariant sets change wildly (Figure 15, middle). To quantify this effect, we sampled 300 systems
 1589 and randomly chose prototypes that are either equivalent or mismatched (as in Figure 14). We then
 1590 calculated the variance of the positions on the estimated invariant set, averaged over the full limit
 1591 cycle. Figure 15 (right) shows that this variance is much higher for mismatched prototype-system
 1592 pairs when compared to those that are equivalent. Thus, we expect that this measure of fitting variance,
 1593 paired with the equivalence loss, can be used in order to detect out-of-dictionary scenarios, where the
 1594 prototypes and data do not exhibit the same behavior.

1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619