DATA-DRIVEN RANDOM-FEATURE APPROXIMATION OF KOOPMAN GENERATORS OF NONLINEAR CONTINUOUS-TIME SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning continuous-time dynamical systems from data is an active area of research. The Koopman formalism is often used here, because it is a flexible approach that is informative about the system's fundamental properties. The formalism represents dynamical systems through linear operators acting on a suitable function space. A practical difficulty arising in the numerical approximation of Koopman operators is the choice of a "dictionary", meaning an effective finite-dimensional representation of the function space. In this paper, we focus on learning dynamical systems in continuous-time, and so we consider numerical approximations of the Koopman generator instead of the Koopman operator. Our main contribution is a data-driven random-feature method to obtain a parsimonious and effective function representation for Galerkin approximation methods such as gEDMD. The representation we construct is data-informed, which allows us to improve upon data-agnostic methods. We empirically demonstrate the suitability of our method for forecasting tasks and system analysis. The numerical experiments show that our approach is accurate, and faster than iterative optimization. Further, we prove convergence of our approximation of the vector field and the resulting trajectories. In addition to the numerical results, we also make theoretical contributions, including a general convergence criterion for the semigroup generated by the gEDMD approximation and a counterexample showing that gEDMD does not generally preserve exponential boundedness. With our contribution, we advance the frontier of efficiently learning continuous-time dynamics from data, towards interpretable and mathematically robust modelling based on the Koopman framework.

1 INTRODUCTION

Identifying and inferring fundamental properties of complex nonlinear dynamical systems with an unknown analytical expression using only sparse or irregularly sampled observations is an important but difficult problem in scientific machine learning. One viable framework for such analyses is Koopman theory (Koopman, 1931), which has received much attention in recent years due to benefiting greatly from advancements in data-driven methods (Mezić, 2005; Budišić et al., 2012). The Koopman framework shifts the perspective from the evolution of states in physical coordinates to the evolution of functions of the state, called observables (Budišić et al., 2012). This enables the study of the underlying system through linear operators acting on the space of observables rather than directly handling the nonlinear dynamics in state space. The central operators of study are the semigroup of Koopman operators and its associated generator, which capture most properties of the semigroup of flow maps and vector field. The operators are linear on the space of observables of the system, which makes them amenable to powerful tools from spectral theory and which can help to reveal global system behaviour, such as stability, attractors and constants of motion (Mezić & Banaszuk, 2004). For general dynamical systems, this linearisation is only possible when the space of observables is chosen to be sufficiently large, typically infinite-dimensional, thus turning the problem of approximating an unknown finite-dimensional nonlinear map into approximating an infinite-dimensional but linear map. Given a dataset of trajectories, common methods for approximating these operators are the EDMD (Williams et al., 2015a) and gEDMD (Klus et al., 2020) algorithms, which learn the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

evolution of the observables projected onto a finite-dimensional subspace. This subspace is spanned by a chosen set of functions, often called a "dictionary". Rather than targeting a single member of the operator semigroup, it is useful to learn its generator, which represents the instantaneous change rather than a fixed time increment and makes the whole semigroup accessible via its spectrum. The performance of EDMD and gEDMD depends heavily on selecting a dictionary capable of capturing the dynamics of the underlying system, a task that is often difficult without sufficient prior knowledge of the system. A solution to this problem is to construct the dictionary from the available data using data-driven methods, a task which is commonly referred to as dictionary learning. For instance, Li et al. (2017) proposed training a neural network as a dictionary for EDMD by minimising a suitable loss function. This has later been extended to an analogous algorithm for learning the Koopman generator using gEDMD (Buzhardt et al., 2025; Xu et al., 2025). As alternatives to neural networks, researchers have also investigated using random features as dictionary functions (Tabish et al., 2025; Sprink et al., 2026), which are computationally efficient by avoiding the iterative optimisation of training a neural network. In this work, **we employ data-driven random-features to learn parsimonious and effective dictionaries for gEDMD**. The random feature distribution is constructed using the SWIM algorithm (Bolager et al., 2023), an approach in which training data prescribes the choice of random features. This approach allows us to combine the performance of a learned dictionary with the efficiency of random-feature methods to approximate the Koopman generator from continuous-time data.

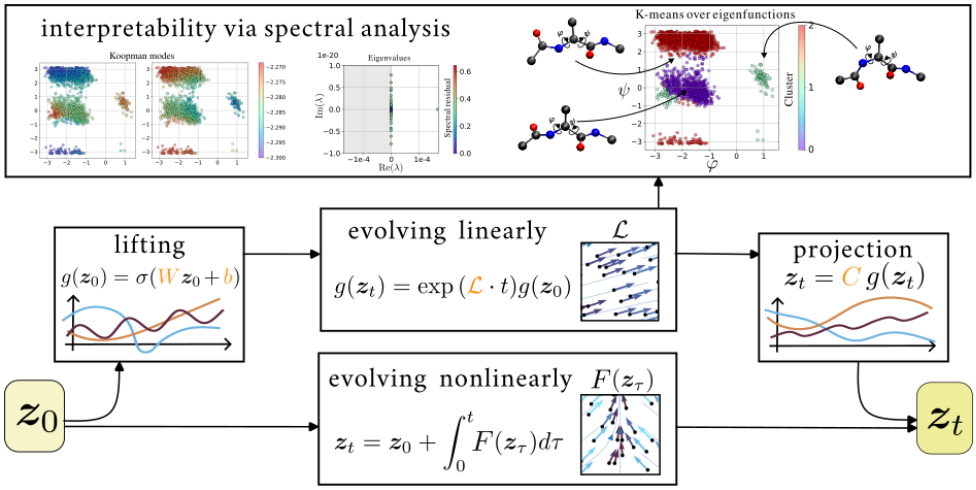


Figure 1: A schematic representation of our method. We construct all terms in orange from the training data.

Chaotic dynamical systems present a particular challenge to data-driven methods, because prediction is only possible for a limited time range. In the paper, we discuss this challenge with respect to dictionary learning for gEDMD, and contrast it to adjoint-based methods such as Neural ODEs.

Our key contributions in this paper are the following.

- We propose a method for learning linear representations of continuous-time dynamical systems (see Figure 1, Section 2).
- We provide theoretical guarantees for the convergence of our method given sufficient data and dictionary functions (Theorem 1, Theorem 2).
- We extend the convergence theory of gEDMD (Theorem 2, Theorem 3).
- We investigate challenges of learning chaotic continuous-time systems from data and contrast it to Neural ODEs (Theorem 4).
- We demonstrate the effectiveness of our approach in forecasting and characterizing the long-term behavior of dynamical systems (Section 4).

We include a comprehensive review of related works in Appendix A.

2 METHODOLOGY

We briefly explain the mathematical background for Koopman operators in Section 2.1 and for random feature methods in Section 2.2. We provide a detailed explanation of our method in Section 2.3.

2.1 KOOPMAN OPERATOR & GENERATOR

Consider a dynamical system generated by the vector field $d\boldsymbol{z}_t/dt = F(\boldsymbol{z}_t)$, with flow $\varphi_t : \mathcal{X} \mapsto \mathcal{X}$ satisfying $\boldsymbol{z}_t = \varphi_t(\boldsymbol{z}_0)$, $\varphi_0 = \mathrm{Id}_{\mathcal{X}}$, and $\varphi_{t_1} \circ \varphi_{t_2} = \varphi_{t_1+t_2}$. The system is defined by the triplet $(\mathcal{X}, \mathcal{T}, \varphi_\cdot)$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is a compact state space equipped with a measure μ, and $\mathcal{T} \subseteq \mathbb{R}$ is the time domain. Koopman theory allows us to define a linear operator $\mathcal{K}^t : \mathcal{F} \mapsto \mathcal{F}, \forall\, g \in \mathcal{F}$, $[\mathcal{K}^t g](\boldsymbol{z}_0) = g \circ \varphi_t(\boldsymbol{z}_0)$, where $\mathcal{F}$ is a function(al) space, typically a Hilbert space. In the literature, g is often called an *observable*. Due to $\{\varphi_t\}_{t\geq 0}$ forming a semigroup, $\{\mathcal{K}^t\}_{t\geq 0}$ also forms a semigroup called the *Koopman semigroup*. For a strongly continuous semigroup $\{\mathcal{K}^t\}_{t\geq 0}$ one can define an operator $\mathcal{L} : \mathcal{D} \mapsto \mathcal{F}$ as $\mathcal{L}g := \lim_{t\to 0} (\mathcal{K}^t g - g)/t, \forall g \in \mathcal{D}$. $\mathcal{L}$ is called a *generator* of the Koopman semigroup and is defined on the domain $\mathcal{D}$ consisting of all functions g for which the limit is defined. One can use $\mathcal{L}$ to express an ODE on the space $\mathcal{F}$ as $\dot{g} = \mathcal{L}g$, where a solution is given with $g(\boldsymbol{z}_t) = \exp(\mathcal{L} \cdot t) g(\boldsymbol{z}_0) = \mathcal{K}^t g(\boldsymbol{z}_0)$. The transformation to a functional representation of the system is illustrated in Figure 2, where the finite-dimensional nonlinear vector field is the usual representation of a process, and using the Koopman framework, this can be transformed to a linear vector field, determined by the Koopman generator $\mathcal{L}$. A more detailed introduction to Koopman theory can be found in Appendix B.

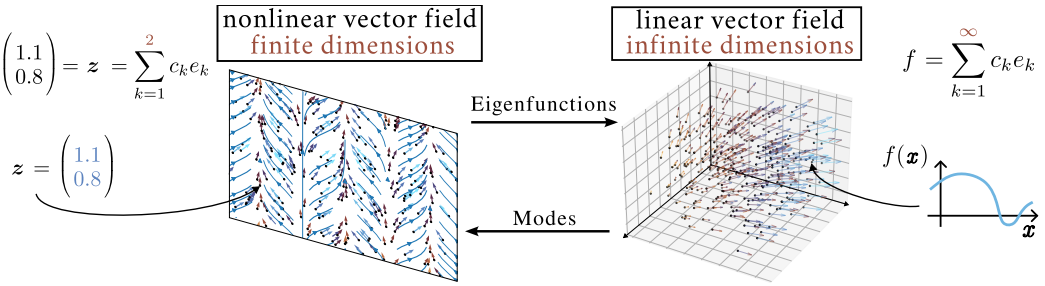


Figure 2: Illustration of the Koopman framework. Focusing on the Koopman generator trades the finite-dimensional nonlinear vector field for an infinite-dimensional linear one.

Nüske et al. (2023) derive finite data error bounds for approximating Koopman generators. Recently, Llamazares-Elias et al. (2026) extended and generalised existing results for a large class of operators and algorithms, including EDMD and gEDMD as special cases.

2.2 RANDOM FEATURES METHOD: SAMPLING WEIGHTS

Our main algorithm relieas on the construction of a dictionary via the SWIM method introduced by Bolager et al. (2023), which we discuss in detail in Appendix A. To summarize, let $\mathbb{P}_{\mathcal{X}}$ be a probability distribution over the data domain $\mathcal{X} \times \mathcal{X}$ with the density

$$p_{\mathcal{X}}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \propto \|f(\boldsymbol{x}^{(2)}) - f(\boldsymbol{x}^{(1)})\| / \|\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}\|, \tag{1}$$

which is high at steep gradients. For each neuron in the hidden layer of $F_{\mathcal{X}}$, sample $(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \sim \mathbb{P}_{\mathcal{X}}$ and set the weight $\boldsymbol{w}$ and the bias b of said neuron to

$$\boldsymbol{w} = s_1 \frac{\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}}{\|\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}\|^2}, \quad b = -\langle \boldsymbol{w}, \boldsymbol{x}^{(1)} \rangle + s_2, \tag{2}$$

where $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ denote the Euclidean norm and inner product, and $s_1, s_2 \in \mathbb{R}$ are scalars that are fixed by the choice of activation function (Bolager et al., 2023).

2.3 DRAGONS

We now introduce the main approach of this paper: a **D**ata-driven **R**andom-feature **A**pproximation of Koopman **G**enerators **o**f **N**onlinear continuous-time **S**ystems. We thus refer to our approach

as DRAGONS. The general setting for our approach is as follows. Let $\boldsymbol{X} = \{\boldsymbol{z}_m, \varphi_{t_m}(\boldsymbol{z}_m)\}_{m=1}^M$ denote a dataset sampled from a dynamical system, consisting of $M \in \mathbb{N}$ snapshots, with $\boldsymbol{z}_m \in \mathcal{X} \subset \mathbb{R}^d$. We assume the flow map $\{\varphi_t\}_{t\geq 0}$ is not known, and we also will not attempt to approximate it directly; instead, we will approximate the Koopman semigroup $\{\mathcal{K}^t\}_{t\geq 0}$ of the system.

First, a set of observables needs to be chosen. Motivated by the success of data-driven random features, we set the observables to be neurons of a hidden layer with parameters sampled according to SWIM using Equation (2), with pairs of trajectory snapshots $(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}) \sim \mathbb{P}_{\mathcal{X}}$. Since there are multiple plausible choices for the sampling target f in Equation (1), we investigate three different options for $\mathbb{P}_{\mathcal{X}}$, which we summarise in Table 1, while providing a detailed discussion in Appendix C. We evaluate constructing the dictionary as a basis for regressing against the vector field or flow map, or constructing the neurons based on how much information they reveal about the generator. In our experiments we found that the generator-driven sampling consistently performs the best. Denote a SWIM hidden layer with $N \in \mathbb{N}$ neurons as $\Phi_N : \mathcal{X} \to \mathbb{R}^N$, with each neuron taking the role of one observable, denoted as $\{\phi_i\}_{i=1}^N$. In the Koopman literature, embedding the finite-dimensional states $\boldsymbol{z}_m$ into the observable space by evaluating $\Phi_N(z_m)$ is called *lifting*.

Table 1: DRAGONS Zoo. Different densities $p_{\mathcal{X}}(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)})$ for sampling distribution $\mathbb{P}_{\mathcal{X}}$.

Flow-driven:	**Vector field-driven:**	**Generator-driven:**
$\frac{\left\|\varphi_{\tau_2}(\boldsymbol{z}^{(2)}) - \varphi_{\tau_1}(\boldsymbol{z}^{(1)})\right\|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|}$	$\frac{\left\|F(\boldsymbol{z}^{(2)}) - F(\boldsymbol{z}^{(1)})\right\|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|}$	$\frac{\left\|(\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)})^\top (F(\boldsymbol{z}^{(2)}) - F(\boldsymbol{z}^{(1)}))\right\|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|^3}$

Under simple assumptions (see Lemma 3), a SWIM-sampled dictionary is suitable for approximating a Koopman generator $\hat{\mathcal{L}}_{NM}$ as the matrix $\hat{L}_{NM}$ with gEDMD such that the operator it represents converges pointwise, $\hat{\mathcal{L}}_{NM} \overset{\text{a.s.}}{\to} \mathcal{L}$. Thus, we use gEDMD for the approximation of the generator on the span of SWIM-neuron observables using a standard Galerkin approximation, i.e.,

$$[\hat{\mathbf{C}}_{NM}]_{i,j} = \frac{1}{M}\sum_{m=1}^{M} \overline{\phi_j(\boldsymbol{z}_m)}\mathcal{L}\phi_i(\boldsymbol{z}_m) \qquad [\hat{\mathbf{G}}_{NM}]_{i,j} = \frac{1}{M}\sum_{m=1}^{M} \overline{\phi_j(\boldsymbol{z}_m)}\phi_i(\boldsymbol{z}_m) \tag{3}$$
$$\hat{L}_{NM}^T = \hat{\mathbf{C}}_{NM}\hat{\mathbf{G}}_{NM}^+.$$

This enables us to evolve observables ϕ_i in time: we have $\dot{\phi}_i(\boldsymbol{z}_m) = \mathcal{L}\phi_i(\boldsymbol{z}_m)$, which we can integrate to obtain a linear map $\mathcal{K}^t = \exp(\mathcal{L}t)$ to compute $\phi_i(\varphi_t(\boldsymbol{z}_m)) = \mathcal{K}^t\phi_i(\boldsymbol{z}_0)$.

For many applications we are interested in forecasting the state of the dynamical system. Thus, we need to invert the lifting function which maps finite-dimensional states into observable space. In practice, and for a wide range of Koopman dictionaries, a common approach is to approximate the inverse of Φ_N with a linear map $C = \boldsymbol{X}\Phi_N(\boldsymbol{X})^+$; we refer to this mapping as the *projection* step. The matrix C contains the so-called Koopman modes (Budišić et al., 2012).

Algorithm 1 Training DRAGONS.

procedure TRAIN($\boldsymbol{X}, \dot{\boldsymbol{X}}$)
  $\boldsymbol{W}, \boldsymbol{b} \leftarrow$ SAMPLE-Layer($\boldsymbol{X}, \mathbb{P}_X$)
  $\Phi_N(\cdot) \leftarrow \sigma(\boldsymbol{W}\cdot + \boldsymbol{b})$
  $\boldsymbol{L}^T = (\partial_t\Phi_N(\boldsymbol{X}))(\Phi_N(\boldsymbol{X}))^+$
  $\boldsymbol{C} = \boldsymbol{X}\,\Phi_N(\boldsymbol{X})^+$
  Return $\boldsymbol{C}, \boldsymbol{L}, \Phi_N$
end procedure

Algorithm 2 Forecasting with DRAGONS.

procedure PREDICT($\boldsymbol{X}, \dot{\boldsymbol{X}}, (t_i)_{i=1}^K, \boldsymbol{x}_{t_0}$)
  $\boldsymbol{C}, \boldsymbol{L}, \Phi_N \leftarrow$ train($\boldsymbol{X}, \dot{\boldsymbol{X}}$)
  for $i = 1, 2, \ldots, K$ **do**
    $\boldsymbol{x}_{t_i} \leftarrow \boldsymbol{C}\exp(\boldsymbol{L}^T(t_i - t_{i-1}))\Phi_N(\boldsymbol{x}_{t_{i-1}})$
  end for
  Return $(\boldsymbol{x}_{t_i})_{i=1}^K$
end procedure

Our approach is summarized in Algorithm 1. It provides a discretized approximation of the Koopman generator, which can be used to predict system states through Algorithm 2. If they are not given, we approximate the time derivatives of X using a two-point finite difference scheme. In the worst case, the computational time for Algorithm 1 scales cubically with the number of datapoints. This can be reduced drastically, even to linear scaling, by using efficient, iterative linear solvers and by appropriate subsampling of the full dataset.

3 THEORETICAL RESULTS

In this section, we present our theoretical results. Here we only state the main theorems and provide proof sketches, while the full proofs can be found in the Appendices D and E.

3.1 CONVERGENCE RESULTS

A major benefit of connecting our method to Koopman theory is that it facilitates theoretical analysis by making existing results on the convergence of gEDMD and from the general theory of C_0-Semigroups available for use. We start by stating the required assumptions, for which we provide practically oriented sufficient conditions through Lemma 1.

Assumption 1. For the results presented in Section 3.1, we assume the following.

- The measure μ has an everywhere positive and bounded density ρ w.r.t. the Lebesgue measure λ on $\mathcal{X}$, where $0 < \alpha \leq \rho(\boldsymbol{x}) \leq \beta < \infty, \forall \boldsymbol{x} \in \mathcal{X}$.
- The data is sampled i.i.d. from μ.
- The Koopman operators $\mathcal{K}^t$ form a C_0-semigroup.

The first two points are not very restrictive as they include a large class of measures. Furthermore, as Lemma 1 shows, the third point is readily attained for Lipschitz vector fields, which is the natural setting to work in due to the Picard-Lindelöf Theorem. For SWIM-sampled networks, we additionally require the activation function to be ReLU to enable approximation of Sobolev functions as required by Llamazares-Elias et al. (2026). Our results naturally extend to other dictionaries that fulfil a Sobolev approximation property analogous to Lemma 2, including classically trained neural networks with a broad class of activation functions (Hornik et al., 1990; Hornik, 1991). Additionally, we also require $\mathcal{X}$ to follow Definition 1 in Appendix D.1, which allows the use of existing universal approximation theory from Bolager et al. (2023). We can now state our first convergence result.

Theorem 1 (Convergence of the approximate vector field)**.** *Let the assumptions be as in Assumption 1. Then $\forall \epsilon > 0$, there are constants $N, M_N \in \mathbb{N}$, a SWIM sampled hidden layer Φ_N and outer weights $\boldsymbol{c}^i$ such that for any $M'_N \geq M_N$ almost surely*

$$\left\| \hat{F}_i - F_i \right\|_{\mathcal{F}} = \left\| \sum_{j=1}^{N} c_j^i \hat{\mathcal{L}}_{NM'_N} \phi_j - F_i \right\|_{\mathcal{F}} < \epsilon, \quad \forall i \in \{1, 2, \ldots, d\}. \tag{4}$$

Proof sketch. Llamazares-Elias et al. (2026) provide a thorough analysis and sufficient conditions on the dictionary for pointwise convergence of the gEDMD approximations to $\mathcal{L}$. We show that our dictionary fulfills their assumptions in Lemma 3. The claim then follows from their results. □

As an auxiliary result, we also extend the existing convergence theory for sampled networks to weighted Sobolev spaces through Lemma 2. Furthermore, we extend the existing convergence theory for gEDMD by providing, to the best of the authors' knowledge, the first proof for pointwise convergence of the semigroup generated by the gEDMD approximation under suitable conditions.

Theorem 2 (Convergence of the approximate trajectories)**.** *Let the assumptions be as in Assumption 1 and assume the true Koopman operators to form a quasi-contractive semigroup, i.e. $\|\mathcal{K}^t\| \leq e^{\omega t}$ for some $\omega_* \in \mathbb{R}$ and all $t \in [0, T]$. Then $\forall \epsilon > 0$ and $t \in [0, T]$, there are constants $N, M_N \in \mathbb{N}$, a SWIM sampled hidden layer Φ_N and outer weights $\boldsymbol{c}^i$ such that for any $M'_N \geq M_N$ almost surely*

$$\|(\varphi_t)_i - (\hat{\varphi}_t)_i\|_{\mathcal{F}} = \left\| \exp(\mathcal{L}t)\, e_i - \sum_{j=1}^{N} c_j^i \exp\left(\hat{\mathcal{L}}_{NM'_N} t\right) \phi_j \right\|_{\mathcal{F}} < \epsilon, \quad \forall i \in \{1, 2, \ldots, d\}. \tag{5}$$

Proof sketch. We show the statement for arbitrary observables $g \in \mathcal{D}$. The main idea is to apply the First Trotter-Kato Approximation Theorem (Engel & Nagel, 2000, Theorem 4.8), which guarantees the convergence of the approximate semigroups given exponential boundedness and the convergence of the generators. The exponential boundedness is obtained in our setting for quasi-dissipativite systems as we show via Lemma 4 that gEDMD preserves this property. □

This result is not specific to our dictionary, as it does not use any specific properties beyond universal approximation and thus transfers to any dictionaries in the general setting of Llamazares-Elias et al. (2026). We empirically found the standard practice of lifting and projecting in between evaluations, as written in Algorithm 2, to perform better than a single projection, which is why this is done in the numerical experiments. Regarding the assumption of quasi-dissipativity, we have two results worth highlighting. Firstly, Lemma 1 shows that this is fulfilled for Lipschitz vector fields with μ being the normalised Lebesgue measure. Secondly, we provide an explicit counterexample (Theorem 3) showing that without imposing further restrictions, the semigroups generated by gEDMD do not generally inherit the exponential boundedness from the true semigroup, which we formalise in the following theorem.

Theorem 3 (gEDMD does not preserve exponential boundedness)**.** *There are systems for which the true semigroup $\mathcal{K}^t$ is exponentially bounded and the sequence of gEDMD approximants $\hat{\mathcal{L}}_{NM'_N}$ converges pointwise on $\mathcal{D}$ to $\mathcal{L}$, while $\nexists W, \omega \in \mathbb{R}$ such that*

$$\left\| \exp\left(\hat{\mathcal{L}}_{NM'_N} t\right) \right\| \leq W e^{\omega t}, \quad \forall N \in \mathbb{N}, \tag{6}$$

i.e. the sequence of approximate semigroups can not be bounded exponentially by uniform constants.

Proof sketch. We show that the scalar Lipschitz system $\dot{x} = -x$ with a dictionary in the setting of Lemma 1 and Llamazares-Elias et al. (2026) meeting the stated convergence and boundedness requirements can yield a sequence $\hat{\mathcal{L}}_{NM'_N}$ with eigenvalues lower bounded by N for each $N \in \mathbb{N}$. The main mechanism is choosing our dictionaries so that one function concentrates on a cusp pathology of the density. The details can be found in Theorem 3 in Appendix D.3. □

Since the enabling mechanism is the pathology of the density, this motivates the restriction to the Lebesgue measure as a sufficient condition for quasi-dissipativity. A more thorough investigation into when gEDMD preserves exponential boundedness is left for future work.

3.2 Limitation: Spectral Convergence

In this section, we briefly highlight what we can and cannot theoretically guarantee about how the spectrum of our approximant relates to that of the true generator. In general, spectral convergence is a challenging problem within the Koopman community (Colbrook & Townsend, 2024), that has as of yet not been fully resolved. Llamazares-Elias et al. (2026, Theorem 6.3) show the almost sure convergence of eigenpairs of $\hat{\mathcal{L}}_{NM'_N}$ to some (nontrivial) eigenpairs of $\mathcal{L}$ when taking an appropriate subsequence of (N, M_N) pairs, where the convergence of the eigenfunctions is weak. By Lemma 3, this result also applies to our method. However, it is worth highlighting the limitations of this result. Firstly, the result holding only for a subsequence means that increasing the dictionary size and training data can potentially increase the error on the spectral approximation, even when reducing the pointwise error for the generator over $\mathcal{F}$. Secondly, the result explicitly guarantees convergence only to a subset of the true spectrum. This is an inherent limitation of gEDMD, as it cannot resolve continuous spectra, which commonly arise in sufficiently complex dynamical systems (Mezić, 2022; Giannakis & Valva, 2024). To account for this limitation, in addition to the computed spectra, we also approximate the associated residual according to Colbrook & Townsend (2024) and report it in all of our experiments. Chaotic systems are particularly susceptible to this limitation. Next to posing difficulties for spectral approximation, these systems also pose more general challenges for learning them from data using neural ODEs, which we highlight in the next section.

3.3 Related Limitations of Neural ODEs: chaotic systems

Neural ODEs (Chen et al., 2018) are a popular approach for training a neural network to approximate a vector field from data. The main idea is to use the adjoint equations to propagate the error through time. We now describe the main issue of this approach when applied to data from a chaotic system, and provide a more detailed discussion in Appendix E. Consider training a neural network to parameterise a vector field $\dot{\boldsymbol{z}}_{\boldsymbol{\theta}} = \boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\theta})$ by minimising a suitable scalar loss $\mathcal{J}(\boldsymbol{\theta}) = \mathcal{J}(\texttt{ODESolve}(\boldsymbol{z}_0, \boldsymbol{f}, T, \boldsymbol{\theta}))$, i.e., between the predicted and ground-truth future states for a given set of initial conditions and evaluation time points. Here, $\boldsymbol{\theta}$ denotes the trainable parameters and T the prediction horizon. The idea of Neural ODEs is to compute the loss gradient $\nabla_{\boldsymbol{\theta}} \mathcal{J} = \mathrm{d}\mathcal{J}/\mathrm{d}\boldsymbol{\theta}$ by employing the adjoint sensitivity method (Pontryagin, 1961; Céa, 1986). In this

method, the sensitivity $\boldsymbol{a}(t) = \mathrm{d}\mathcal{J}/\mathrm{d}\boldsymbol{z}(t)$ is used to define $\nabla_{\boldsymbol{\theta}}\mathcal{J} = -\int_0^T a(t)^\top \partial \boldsymbol{f}/\partial \boldsymbol{\theta}(\boldsymbol{z}_{\boldsymbol{\theta}}(t); \boldsymbol{\theta})\mathrm{d}t$. This error is integrated backwards from a predicted state at time T using a numerical solver to obtain the parameter update. We now state a simplified version of our main result. It establishes that Neural ODEs trained using the adjoint method are prone to exploding and vanishing gradients on certain systems, regardless of the numerical integrator used for training. The full statement (Theorem 7) and proof can be found in Appendix E.

Theorem 4 (Simplified: Exponential growth of the parameter gradient of a neural ODE)**.** *Consider the Neural ODE, $\dot{\boldsymbol{z}_{\boldsymbol{\theta}}}(t) = \boldsymbol{f}(\boldsymbol{z}_{\boldsymbol{\theta}}(t), \boldsymbol{\theta})$ trained with the adjoint method using the loss function $\mathcal{J}$ defined above, on time series data from a chaotic system over a time horizon T. Since the data are chaotic, the largest Lyapunov Exponent (LE) is positive, $\lambda_{\max} > 0$, and for sufficiently long $[t, T]$, the largest Finite-Time Lyapunov Exponent (FTLE) is also positive, $\lambda_{\max}^{FT} > 0$, with $\lambda_{\max}^{FT} \to \lambda_{\max}$ as $T \to \infty$. Then in the finite-time regime, $\|\nabla_\theta \mathcal{J}\| = \Omega\Big(e^{\lambda_{\max}^{FT} T}\Big)$, and asymptotically as $T \to \infty$, $\|\nabla_\theta \mathcal{J}\| = \Omega\big(e^{(\lambda_{\max} - \epsilon)T}\big)$ for every $\epsilon > 0$. Here $\Omega(\cdot)$ denotes a lower bound up to a constant factor.*

Proof sketch. We first establish analogous bounds on the adjoint state in Theorems 5 and 6. The main idea is to express the solution of the adjoint equation in terms of the tangent linear flow via its fundamental matrix. For the FTLE bounds, the upper bound follows from the definition of $\lambda_{\max}^{FT}$, while the lower bound follows from the singular value decomposition of the fundamental matrix and the projection of the terminal adjoint onto its maximal left singular vector (Theorem 5). For the LE bounds, the upper bound follows from the definition of $\lambda_{\max}$, and the lower bound follows from Oseledec's multiplicative ergodic theorem applied to the dual Oseledec decomposition, where the top dual Oseledec subspace determines the backward growth rate (Theorem 6). These estimates are then transferred to the loss gradient via norm estimates on the integral expression and the stated boundedness assumption. □

The asymptotic (LE) results show that training on chaotic systems is prone to exploding gradients when sufficiently long trajectories are used. The finite-time (FTLE) results give exact, computable bounds at the actual training horizon, without requiring ergodicity or asymptotic limits, and thus provide a practical diagnostic for anticipating gradient explosion or vanishing on finite intervals. We validate this numerically on the example of the Lorenz 63 system in Appendix F.7.

4 Computational experiments

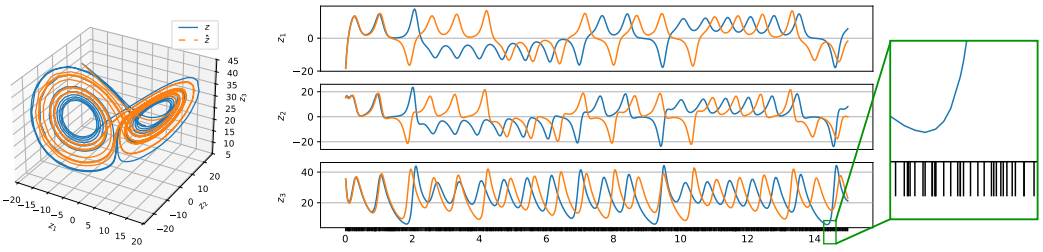
4.1 Forecasting

To assess the predictive performance of DRAGONS, we choose three test problems: a Van der Pol oscillator, the chaotic Lorenz 63 system, and a mean-field model of cylinder flow. Details on the datasets, experimental setup and hyperparameter search are given in Appendix F. We compare our approach to state-of-the-art dictionary learning methods using an Adam optimizer (gEDMD-DL), and against data-agnostic random features (Gaussian) and common dictionary choices such as radial basis functions (RBF-Dict) and polynomials (Poly-Dict). For the chosen systems, a polynomial dictionary is sufficiently expressive and invariant, thus the Poly-Dict sets the standard for a good dictionary for all of these problems.

The results are given in Table 2, where we report the test error, dictionary size and training time. DRAGONS shows excellent performance, either on par with or better than existing methods in terms of error. The fit time is reasonably low when considering the dictionary size, indicating that the computation is dominated by the linear least squares solver. For chaotic systems in particular we achieve a lower error with DRAGONS compared to the Gaussian and the Adam-trained neural network dictionary. This is an noteworthy observation because chaotic systems have continuous spectra, which pose a challenge for gEDMD (see Section 3.2). DRAGONS is thus a robust choice for forecasting tasks on irregular time intervals. Further details and insights from these experiments can also be found in Appendix F.

Table 2: Results are evaluated on a test set and averaged over 5 random seeds.

Example	Model	Error: avg (min, max)	# dict	Time [s]: avg (min, max)
Van der Pol	DRAGONS	6.23e-7 (2.84e-7, 8.08e-7)	130	0.79 (0.76, 0.89)
	Gaussian	3.18e-6 (1.31e-6, 6.23e-6)	130	0.69 (0.67, 0.70)
	gEDMD-DL	1.36e-5 (3.73e-6, 2.73e-5)	130	90.61 (55.92, 223.04)
	RBF-Dict	1.58e-2 (1.26e-2, 1.81e-2)	202	0.60 (0.56, 0.68)
	Poly-Dict	2.29e-5	9	0.37
Lorenz 63	DRAGONS	3.75e-4 (1.87e-4, 5.69e-4)	131	2.72 (2.68, 2.76)
	Gaussian	3.68e-1 (1.18e-1, 8.36e-1)	67	3.19 (2.01, 5.02)
	gEDMD-DL	1.48e-1 (2.39e-2, 4.14e-1)	19	11.83 (11.41, 12.14)
	RBF-Dict	1.58e-1 (9.76e-2, 2.47e-1)	303	2.53 (2.48, 2.58)
	Poly-Dict	3.85e-4	34	1.87
Cylinder	DRAGONS	3.34e-7 (3.28e-7, 3.42e-7)	515	5.50 (4.69, 8.06)
	Gaussian	3.24e-7 (3.13e-7, 3.41e-7)	131	2.35 (1.84, 3.35)
	gEDMD-DL	3.24e-7 (3.23e-7, 3.28e-7)	35	14.67 (14.14, 15.50)
	RBF-Dict	1.64e-3 (1.61e-3, 1.65e-3)	103	1.37 (1.15, 1.74)
	Poly-Dict	3.13e-7	34	0.93

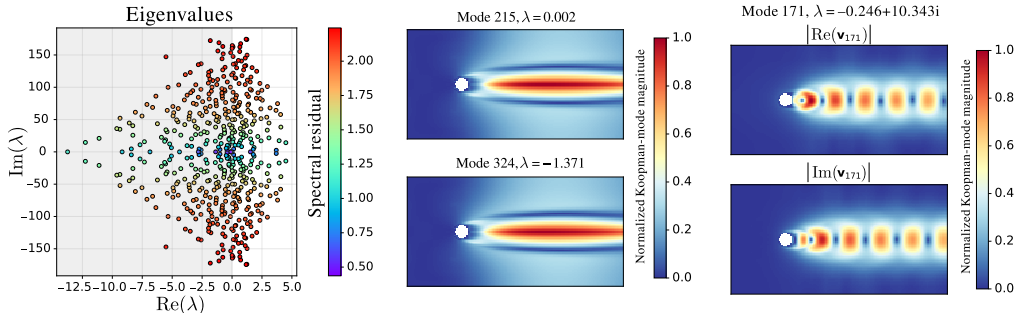


(a) Phase space (b) Trajectory

Figure 3: Comparison of numerically calculated versus predicted trajectories for Lorenz experiment. Ticks on the x-axis show the evaluation points in time, which are irregularly spaced.

4.2 Spectral Analysis

Koopman Mode Analysis: Flow around a cylinder In this experiment, we demonstrate that spectral properties of a system's Koopman generator can be extracted accurately with DRAGONS. Specifically, it allows us to interpret the underlying system dynamics on a global scale for the widely used model of the flow around a cylinder. Figure 4a shows the eigenvalues of the generator and their corresponding spectral residuals. Nine eigenvalues with large negative real parts are omitted for visualization. The real part of an eigenvalue determines the growth or decay rate of the corresponding mode, while the imaginary part determines its oscillation frequency. The two modes of a complex-conjugate pair typically represent one oscillatory pattern. Representative spatial Koopman modes corresponding to three selected eigenvalues are shown in Figure 4b and Figure 4c.



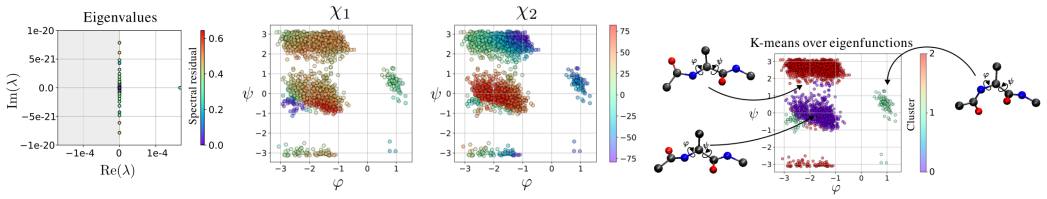
(a) Eigenvalues of $\hat{L}$. (b) Non-oscillatory modes. (c) Oscillatory mode.

Figure 4: Generator spectrum and representative spatial modes for the flow around a cylinder.

Several modes with positive real parts are present in the learned spectrum, which may reflect the transient dynamics away from the unstable steady solution. Although such modes are associated with growth, we do not observe unstable predictions near the limit cycle, probably because their net contribution to the reconstructed dynamics remains small in this region. In addition, the relatively large residuals of some growing modes may indicate that they are only locally accurate near the unstable state rather than globally valid along the entire trajectory.

Molecular Conformation Dynamics: Alanine Dipeptide In this final experiment, we demonstrate that our approach is applicable to high-dimensional data sets. Specifically, we show that the generator approximation can be used to extract distinct molecular conformations and metastable states of alanine dipeptide purely from long and high-dimensional trajectory data.

For molecular dynamics, an evolution in finite time does not suffice for an understanding of the system's long term behaviour; typically high-dimensional systems exhibit a few metastable states determined by a potential energy landscape. The global conformation pattern of alanine dipeptide can be represented using two dihedral angles at the central backbone denoted as φ and ψ (Ramachandran et al., 1963). We use trajectory data from Nüske et al. (2017) and perform a molecular conformation analysis similar to Klus et al. (2018) and Herwig et al. (2025). We find that our model identifies the metastable states present in the dataset, due to the eigenfunctions assigning different values to datapoints in different metastable regions, as can be seen in Figure 5b; these can be clustered using k-means to make the distinctions between regimes clearer as shown in Figure 5c. In addition, we take a look at the eigenvalues of the generator and their spectral residuals calculated according to Appendix F.6. Eigenvalues with an almost zero magnitude have the smallest spectral residual, which is in line with our expectations, and most eigenvalues are along the imaginary axis.



(a) Eigenvalues of $\hat{L}$. (b) Two eigenfunctions of $\hat{L}$. (c) K-means on five eigenfunctions.

Figure 5: Spectral insights into the alanine dipeptide dataset using the approximated Koopman generator $\hat{L}$. The datapoints are projected from their 30-dim. state to its corresponding dihedral angles for visualization purposes. With DRAGONS, we are able to directly extract the important metastable conformations of the molecule (Figure 5c) from the high-dimensional trajectory data.

5 Conclusion

We have shown that data-driven random features provide a robust dictionary for the approximation of the Koopman generator. Our method is supported by empirical results on forecasting problems, demonstrating high accuracy without the high computational costs typically associated with iterative dictionary learning. We also demonstrate how our method can be used to gain insight into the system's spectral properties. Our theoretical results affirm the validity of our method and extend the existing convergence theory for gEDMD. We also extend known limitations of iteratively trained neural networks from discrete-time to a continuous-time setting.

Several limitations remain for future work. Firstly, our convergence results are existence statements without an expression relating the error to the required number of data points or dictionary functions. This could be alleviated by combining results from Llamazares-Elias et al. (2026) and Rudi & Rosasco (2017). Secondly, our theoretical guarantees for spectral convergence are bound to the limitations of gEDMD, which is why future work should look into combining our dictionary with other generator estimation algorithms with improved guarantees. Finally, since our algorithm involves solving a potentially large linear system, in the case of sizeable datasets the user needs to carefully choose an efficient linear solver or subsampling the dataset.

AI USE STATEMENT

Generative AI tools have been used to assist in the writing of proofs, implement methods, assist with translation and interpret results.

However, we did not use generative AI tools to help develop theoretical models or conceptual frameworks, generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat dataset, support qualitative and thematic data analysis. Activities such as formulating questions for surveys or interviews, creation of artifacts and transcribing recordings of research material are not applicable to this work.

Additionally, we used generative AI tools to summarize or analyse existing literature, source information, edit a research paper to improve readability and identify relevant literature.

The majority of work done to produce this manuscript was done without the assistance of AI. Key methodological and theoretical ideas came directly from the authors or were inspired by other research which is clearly cited in the manuscript. Every task performed with the assistance of AI was cross reviewed by authors. AI tools aided the writing of the proof however not in the sense of bridging important gaps, but rather to aid clarity. The core code base is written without any usage of AI tools; AI-generated code was used for auxiliary coding tasks and was verified by at least two authors. We take full responsibility for the completed work, both for the tasks performed with the assistance of AI as well as the ones without.

REPRODUCIBILITY STATEMENT

We support our methodology with theoretical results as well as empirical evidence. For the theoretical results we clearly state the necessary assumptions and provide the proofs with sufficient details in the appendix. For the numerical experiments we provide details of the dataset as well as model hyperparameters in the appendix. Our experiments are performed with different random seeds in order to insure robustness to random effects. We provide an anonymized source code and upon acceptance we will make our code repository public so that others can reproduce this work.

REFERENCES

Alexandre Allauzen, Thiago Petrilli Maffei Dardis, and Hannah Plath. Experimental study of neural ode training with adaptive solver for dynamical systems modeling. In *The Symbiosis of Deep Learning and Differential Equations II, NeurIPS 2022 Workshop*, 2022.

April Herwig. Alanine dipeptide model simulations. https://github.com/April-Hannah-Lena/schoolwork/tree/21db29b63016fe822ce455bac86ae035cd91d88c/Master%20Thesis/scripts, 2024.

Hanru Bai and Weiyang Ding. KoNODE: Koopman-Driven Neural Ordinary Differential Equations with Evolving Parameters for Time Series Analysis. In *Forty-Second International Conference on Machine Learning*, June 2025.

Stanley Bak, Sergiy Bogomolov, Brandon Hencey, Niklas Kochdumper, Ethan Lew, and Kostiantyn Potomkin. Reachability of koopman linearized systems using random fourier feature observables and polynomial zonotope refinement. In *International Conference on Computer Aided Verification*, pp. 490–510. Springer, 2022.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Erik L Bolager, Iryna Burak, Chinmay Datar, Qing Sun, and Felix Dietrich. Sampling weights of deep neural networks. *Advances in neural information processing systems*, 36:63075–63116, 2023.

Erik Lien Bolager, Ana Čukarska, Iryna Burak, Zahra Monfared, and Felix Dietrich. Koopman-informed recurrent neural networks. *Transactions on Machine Learning Research*, 2026a. ISSN 2835-8856. URL `https://openreview.net/forum?id=KHsnxKYG6k`.

Erik Lien Bolager, Boumediene Hamzi, Houman Owhadi, Ioannis G Kevrekidis, and Felix Dietrich. Dictionary learning for kernel edmd. *arXiv preprint arXiv:2604.25572*, 2026b.

Erik Bollt. On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrast to var and dmd. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(1), 2021.

Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

Jake Buzhardt, C Ricardo Constante-Amores, and Michael D Graham. On the relationship between koopman operator approximations and neural ordinary differential equations for data-driven time-evolution predictions. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(4), 2025.

Jean Céa. Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. *ESAIM: Modélisation mathématique et analyse numérique*, 20(3):371–402, 1986.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Matthew J. Colbrook and Alex Townsend. Rigorous data-driven computation of spectral properties of Koopman operators for dynamical systems. *Communications on Pure and Applied Mathematics*, 77(1):221–283, 2024. ISSN 1097-0312. doi: 10.1002/cpa.22125.

C. Ricardo Constante-Amores, Alec J. Linot, and Michael D. Graham. Enhancing predictive capabilities in data-driven dynamical modeling with automatic differentiation: Koopman and neural ode approaches. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(4):043119, 2024. doi: 10.1063/5.0180415.

George Coote and Matthew J Colbrook. Residual-guided dictionary learning for spectrally accurate koopman approximation. *arXiv preprint arXiv:2606.29083*, 2026.

Wojciech Marian Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4281–4290, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Anthony M DeGennaro and Nathan M Urban. Scalable extended dynamic mode decomposition using random kernel approximation. *SIAM Journal on Scientific Computing*, 41(3):A1482–A1499, 2019.

Felix Dietrich, Thomas N Thiem, and Ioannis G Kevrekidis. On the koopman operator of algorithms. *SIAM Journal on Applied Dynamical Systems*, 19(2):860–885, 2020.

Klaus-Jochen Engel and Rainer Nagel. *One-Parameter Semigroups for Linear Evolution Equations*, volume 63. 01 2000. ISBN 0-387-98463-1. doi: 10.1007/b97696.

Carl Folkestad, Daniel Pastor, Igor Mezic, Ryan Mohr, Maria Fonoberova, and Joel Burdick. Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control. In *2020 american control conference (acc)*, pp. 3906–3913. IEEE, 2020.

Daniel J Gauthier, Erik Bollt, Aaron Griffith, and Wendson AS Barbosa. Next generation reservoir computing. *Nature communications*, 12(1):5564, 2021.

Dimitrios Giannakis and Claire Valva. Consistent spectral approximation of koopman operators using resolvent compactification. *Nonlinearity*, 37(7):075021, jun 2024. doi: 10.1088/1361-6544/ad4ade. URL https://doi.org/10.1088/1361-6544/ad4ade.

Marvyn Gulina and Alexandre Mauroy. Two methods to approximate the koopman operator with a reservoir computer. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(2), 2021.

Tobias Günther, Markus Gross, and Holger Theisel. Generic objective vortices for flow visualization. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4):141:1–141:11, 2017.

Yue Guo, Milan Korda, Ioannis G Kevrekidis, and Qianxiao Li. Learning parametric koopman decompositions for prediction and control. *SIAM Journal on Applied Dynamical Systems*, 24(1): 744–781, 2025.

April Herwig, Matthew J. Colbrook, Oliver Junge, Péter Koltai, and Julia Slipantschuk. Avoiding spectral pollution for transfer operators using residuals. (arXiv:2507.16915), July 2025. doi: 10.48550/arXiv.2507.16915.

Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized Teacher Forcing for Learning Chaotic Dynamics. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023. ISSN: 2640-3498.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL https://www.sciencedirect.com/science/article/pii/089360809190009T.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(90)90005-6. URL https://www.sciencedirect.com/science/article/pii/0893608090900056.

Yuhong Jin, Lei Hou, and Shun Zhong. Extended dynamic mode decomposition with invertible dictionary learning. *Neural networks*, 173:106177, 2024.

Annika Junker, Julia Timmermann, and Ansgar Trächtler. Data-driven models for control engineering applications using the koopman operator. In *2022 3rd International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, pp. 1–9. IEEE, 2022.

Stefan Klus, Andreas Bittracher, Ingmar Schuster, and Christof Schütte. A kernel-based approach to molecular conformation analysis. *The Journal of Chemical Physics*, 149(24):244109, December 2018. ISSN 0021-9606. doi: 10.1063/1.5063533.

Stefan Klus, Feliks Nüske, Sebastian Peitz, Jan-Hendrik Niemann, Cecilia Clementi, and Christof Schütte. Data-driven approximation of the koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, 2020.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.

Daniel Lehmberg, Felix Dietrich, Gerta Köster, and Hans-Joachim Bungartz. datafold: data-driven models for point clouds and time series on manifolds. *Journal of Open Source Software*, 5 (51):2283, 2020. doi: 10.21105/joss.02283. URL https://doi.org/10.21105/joss.02283.

Qianxiao Li, Felix Dietrich, Erik M Bollt, and Ioannis G Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10), 2017.

Chunyang Liao. On the approximation of koopman eigenfunctions using random features. *Foundations of Data Science*, pp. 0–0, 2026.

Liam Llamazares-Elias, Samir Llamazares-Elias, Jonas Latz, and Stefan Klus. Data-driven approximation of koopman operators and generators: Convergence rates and error bounds. *Journal of Computational Dynamics*, 15(0):163–206, 2026. ISSN 2158-2491. doi: 10.3934/jcd.2026012. URL https://www.aimsciences.org/article/id/6a43ac2a4b4042019b12f11b.

Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141, 1963. ISSN 1520-0469.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Norman G. Meyers and James Serrin. H = w. *Proceedings of the National Academy of Sciences*, 51 (6):1055–1056, 1964. doi: 10.1073/pnas.51.6.1055. URL https://www.pnas.org/doi/abs/10.1073/pnas.51.6.1055.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

Igor Mezić and Andrzej Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1-2):101–133, 2004.

Igor Mezić. On numerical approximations of the koopman operator. *Mathematics*, 10(7), 2022. ISSN 2227-7390. doi: 10.3390/math10071180. URL https://www.mdpi.com/2227-7390/10/7/1180.

Jonas Mikhaeil, Zahra Monfared, and Daniel Durstewitz. On the difficulty of learning chaotic dynamics with rnns. *Advances in neural information processing systems*, 35:11297–11312, 2022.

Bernd Noack, Konstantin Afanasiev, Marek Morzynski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 12 2003. doi: 10.1017/S0022112003006694.

Feliks Nüske and Stefan Klus. Efficient approximation of molecular kinetics using random fourier features. *The Journal of Chemical Physics*, 159(7), 2023.

Feliks Nüske, Hao Wu, Jan-Hendrik Prinz, Christoph Wehmeyer, Cecilia Clementi, and Frank Noé. Markov state models from short non-equilibrium simulations—Analysis and correction of estimation bias. *The Journal of Chemical Physics*, 146(9):094104, March 2017. ISSN 0021-9606. doi: 10.1063/1.4976518.

Feliks Nüske, Sebastian Peitz, Friedrich Philipp, Manuel Schaller, and Karl Worthmann. Finite-data error bounds for koopman-based prediction and control. *Journal of Nonlinear Science*, 33(1):14, 2023.

Samuel E. Otto and Clarence W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Nauka, Moscow, 1961.

S. Popinet. Free computational fluid dynamics. *ClusterWorld*, 2(6), 2004. URL `https://gerris.dalembert.upmc.fr/clusterworld/clusterworld.pdf`.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th annual allerton conference on communication, control, and computing*, pp. 555–561. IEEE, 2008a.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in neural information processing systems*, 21, 2008b.

G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7:95–99, July 1963. ISSN 0022-2836. doi: 10.1016/s0022-2836(63)80023-6.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science advances*, 3(4):e1602614, 2017.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

Roland Schurig, Pieter van Goor, Karl Worthmann, and Rolf Findeisen. Shaping the koopman dictionary by learning on the grassmannian. *arXiv preprint arXiv:2511.07234*, 2025.

Lawrence Sirovich. Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of Applied Mathematics*, 45:561–571, 10 1987. doi: 10.1090/qam/910462.

Hauke Sprink, Yanchen Zhu, Antonia SJS Mey, and Feliks Nüske. Accurate and robust analysis of molecular kinetics with random features. *The Journal of Chemical Physics*, 164(11), 2026.

Liangyu Su, Jun Shu, Rui Liu, Deyu Meng, and Zongben Xu. Koopgen: Koopman generator networks for representing and predicting dynamical systems with continuous spectra. *arXiv preprint arXiv:2602.14011*, 2026.

Mohammad Tabish, Benedict Leimkuhler, and Stefan Klus. How deep is your network? Deep vs. shallow learning of transfer operators. (arXiv:2509.19930), September 2025. doi: 10.48550/arXiv.2509.19930.

Hiroaki Terao, Sho Shirasaka, and Hideyuki Suzuki. Extended dynamic mode decomposition with dictionary learning using neural ordinary differential equations. *Nonlinear Theory and Its Applications, IEICE*, 12(4):626–638, 2021.

Patrik Valábek, Marek Wadinger, Michal Kvasnica, and Martin Klauco. Deep dictionary-free method for identifying linear model of nonlinear system with input delay. In *2025 25th International Conference on Process Control (PC)*, pp. 1–6. IEEE, 2025.

Pieter Van Goor, Robert Mahony, Manuel Schaller, and Karl Worthmann. Reprojection methods for koopman-based modelling and prediction. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 315–321. IEEE, 2023.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015a.

Matthew O. Williams, Clarence W. Rowley, and Ioannis G. Kevrekidis. A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015b. doi: 10.3934/jcd.2015005.

Yuanchao Xu, Kaidi Shao, Isao Ishikawa, Yuka Hashimoto, Nikos Logothetis, and Zhongwei Shen. A data-driven framework for koopman semigroup estimation in stochastic dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(10):103123, 10 2025.

Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pp. 4832–4839. IEEE, 2019.

A Related Works

Related works in this direction have primarily focused on learning effective dictionaries for Koopman operator approximation. Ideally, the observables span a compact approximately invariant subspace, but constructing such representations often requires strong prior knowledge or restrictive assumptions (Folkestad et al., 2020). Data-driven approaches therefore learn the mapping from the state space to observables directly from data (Dietrich et al., 2020). Neural-network dictionaries are a common choice (Li et al., 2017), including methods that learn Koopman eigenfunctions (Lusch et al., 2018), jointly optimize the dictionary and operator (Yeung et al., 2019), or construct more structured representations using time delays, invertible mappings, or residual information (Valábek et al., 2025; Jin et al., 2024; Coote & Colbrook, 2026). Although expressive, these methods generally require iterative gradient-based optimization and can therefore be substantially more expensive than the final linear Koopman fit.

A computationally attractive alternative is to construct nonlinear features without iteratively training the hidden representation. Random-feature methods fix or directly sample the nonlinear mapping and reduce learning to a linear estimation problem (Rahimi & Recht, 2007; Rudi & Rosasco, 2017). In Koopman approximation, this idea has been explored using random Fourier features (Bak et al., 2022; Nüske & Klus, 2023; Sprink et al., 2026), randomized approximations of Koopman eigenfunctions (Liao, 2026), and reservoir computing (Gulina & Mauroy, 2021).

We discuss these directions in detail below.

Explicitly finding a governing equation. Learning governing dynamics directly from data has long been an important goal in data-driven modeling, particularly for systems whose analytical equations are unknown or only partially available. One approach is symbolic regression, which searches over candidate analytical expressions to identify compact representations of the underlying dynamics (Bongard & Lipson, 2007; Schmidt & Lipson, 2009). Sparse Identification of Nonlinear Dynamics (SINDy) instead assumes that the governing equations admit a sparse representation in a predefined library of candidate functions and identifies the active terms through sparse regression (Brunton et al., 2016). This sparse-regression framework has also been extended to the discovery of partial differential equations through PDE-FIND (Rudy et al., 2017).

Koopman operator approximation. The Koopman operator provides a linear, although generally infinite-dimensional, representation of nonlinear dynamical systems by describing the evolution of observable functions rather than the state itself. Data-driven methods approximate this operator on a finite-dimensional subspace spanned by a dictionary of observables. In particular, Extended Dynamic Mode Decomposition (EDMD) constructs a finite-dimensional approximation from snapshot pairs once such a dictionary has been specified (Williams et al., 2015a). The quality of the resulting approximation, however, depends strongly on this choice. Simple predefined dictionaries, such as monomials, radial basis functions, or other problem-specific features, can be effective for low-dimensional systems but may grow prohibitively large as the state dimension or required approximation order increases.

The dictionary-selection problem. Selecting an effective dictionary is therefore a central challenge in Koopman operator approximation. Ideally, the observables would span a compact Koopman-invariant, or approximately invariant, subspace and thus provide coordinates in which the nonlinear dynamics evolve linearly. Koopman eigenfunctions constitute a natural choice for such coordinates, but their construction generally requires significant prior knowledge or restrictive assumptions on the underlying dynamical system (Folkestad et al., 2020). Conversely, choosing an excessively rich generic dictionary increases both the computational cost and the amount of data required to identify the projected operator. Data-informed dictionaries seek to mitigate this trade-off by constructing representations adapted to the observed dynamics and thereby reducing the number of features required for an accurate approximation (Dietrich et al., 2020).

Kernel-based observable spaces. Kernel methods provide one route toward richer observable spaces without explicitly enumerating a large dictionary. Kernel DMD represents observables implicitly through the feature map associated with a prescribed kernel, making it possible to operate in very high- or infinite-dimensional feature spaces while carrying out the main computations in terms of kernel evaluations (Williams et al., 2015b). This alleviates the explicit feature-growth problem of conventional EDMD, particularly for high-dimensional systems. Nevertheless, the kernel type and

its hyperparameters remain design choices, and the computational cost typically scales unfavorably with the number of snapshots. More recently, kernel selection itself has therefore also been formulated as a dictionary-learning problem, in which combinations of kernels and their parameters are optimized from data (Bolager et al., 2026b). Thus, although kernel methods reduce the need to manually specify individual observables, the broader problem of identifying a representation appropriate for the dynamics remains.

Neural dictionary learning. A widely studied alternative is to parametrize the dictionary directly using neural networks. Early EDMD dictionary-learning approaches jointly adapt a neural representation and the associated Koopman approximation to the observed dynamics (Li et al., 2017). DeepKoopman similarly learns approximate Koopman eigenfunctions from data, demonstrating that nonlinear coordinate transformations can be constructed without prescribing the observable functions beforehand (Lusch et al., 2018). Related architectures combine an encoder, approximately linear latent dynamics, and a decoder to learn low-dimensional representations in which the evolution is close to linear (Otto & Rowley, 2019). Other formulations directly optimize both the neural-network dictionary and the projected Koopman operator (Yeung et al., 2019), use time-delay embeddings to incorporate temporal information (Valábek et al., 2025), or employ invertible neural networks to obtain an explicit inverse transformation between the state and observable spaces (Jin et al., 2024). These approaches substantially increase the flexibility of the dictionary and can yield compact representations for systems for which suitable analytical observables are not known.

Neural ODE-based dictionaries and joint Koopman optimization. Several recent works extend this basic neural dictionary-learning paradigm. Neural ODEs have been used to construct parameter-efficient EDMD dictionaries by interpreting the feature transformation as the flow of an auxiliary continuous-depth network (Terao et al., 2021). Connections between Koopman dictionary learning and neural state-space models have subsequently been studied in more detail. In particular, repeatedly projecting the linearly evolved lifted state back to the physical state space, as commonly done in practical EDMD implementations (Van Goor et al., 2023; Junker et al., 2022), reintroduces a nonlinear state-space evolution and creates a close relationship between projected EDMD models and neural ODE or neural flow-map representations (Buzhardt et al., 2025). Automatic differentiation through the EDMD solution has likewise been used to optimize the dictionary and Koopman approximation simultaneously rather than alternating between the two problems (Constante-Amores et al., 2024).

Structure- and residual-aware dictionaries. The representation can also be adapted to properties beyond short-term prediction accuracy. Trainable dictionaries have, for example, been combined with Residual DMD by incorporating spectral residuals of approximate Koopman eigenvalues and eigenfunctions into the learning objective (Coote & Colbrook, 2026). This explicitly targets the spectral consistency of the learned representation rather than relying solely on reconstruction or trajectory errors. Related work constructs approximately invariant observable subspaces through optimization on the Grassmann manifold, allowing part of the observable space to be constrained by prior knowledge while the remaining subspace is adapted to the data (Schurig et al., 2025). For parametrized dynamical systems, both the dictionary and a parameter-dependent family of projected Koopman operators can be represented by neural networks and learned jointly, permitting the operator itself to vary with static or time-dependent parameters (Guo et al., 2025). Bai & Ding (2025) use the Koopman framework to evolve the parameters of a Neural ODE, although they do not perform dictionary learning, making them an exception to this type of learning.

Neural Network approximation of generators for continuous spectra. Neural approaches have also been developed directly at the generator level. KoopGen, for instance, represents structured state-dependent Koopman generators and is designed to accommodate systems with broadband or continuous spectral components without assuming that the relevant dynamics admit a simple global finite-dimensional eigendecomposition (Su et al., 2026), however since it does not learn dictionaries, the approach is not directly comparable to our method.

Computational cost of trainable dictionaries. These developments highlight the expressive power of neural dictionary learning, but this flexibility comes at a computational cost. The parameters defining the observables must typically be determined through iterative gradient-based optimization, often jointly with operator, decoder, or reconstruction parameters. Consequently, dictionary construction can require many passes over the training data and can become substantially more expensive than the final linear least-squares problem underlying classical EDMD. The result may

additionally depend on network architecture, initialization, learning rates, regularization, and other optimization hyperparameters. This distinction is important: once a fixed dictionary has been constructed, the corresponding Koopman approximation is essentially a linear regression problem, whereas learning the dictionary can turn the overall procedure into a large non-convex optimization problem.

Convergence and spectral reliability. Care is also required when discussing convergence of such approximations. For a fixed sequence of sufficiently rich approximation spaces and appropriate sampling assumptions, convergence results are available for EDMD; in particular, finite-dimensional projected operators converge to the Koopman operator in suitable operator senses as the number of samples and dictionary functions increases (Korda & Mezić, 2018). This does not, however, imply that every eigenvalue of a finite EDMD matrix provides a reliable approximation of the spectrum of the infinite-dimensional Koopman operator. Finite-dimensional truncations can exhibit spectral pollution, particularly when the underlying operator has continuous or otherwise complicated spectral components. Residual DMD addresses this distinction by computing operator-level residuals that permit a posteriori verification of candidate spectral quantities and, under suitable assumptions, provide rigorous convergence guarantees for spectra, pseudospectra, and spectral measures (Colbrook & Townsend, 2024). Recent residual-guided dictionary-learning methods build directly on this observation by incorporating such residuals into the representation-learning objective (Coote & Colbrook, 2026).

Random-feature alternatives. A conceptually different approach is to retain expressive nonlinear representations while avoiding iterative optimization of the hidden mapping altogether. Random-feature methods follow precisely this strategy: rather than learning all parameters of a nonlinear feature map through gradient descent, the hidden parameters are sampled or otherwise constructed once, after which only a linear output model is fitted (Rahimi & Recht, 2007; 2008b). Such models possess strong approximation properties for broad classes of functions, including Barron-type functions (Barron, 1993) and functions in reproducing kernel Hilbert spaces (Rahimi & Recht, 2008a). Conventional random-feature methods typically draw their parameters from a fixed, data-independent distribution; random Fourier features, for example, sample frequencies according to the spectral distribution associated with a shift-invariant kernel (Rahimi & Recht, 2007). Despite their simple training procedure, random features provide favorable trade-offs between approximation accuracy, feature dimension, and computational complexity (Rudi & Rosasco, 2017). From the perspective of Koopman approximation, this separation is particularly natural: the nonlinear features define a candidate observable space, while the action of the Koopman operator on this space can subsequently be estimated through linear regression.

Data-dependent sampling with SWIM. Rather than sampling features from a prescribed, data-independent distribution, Bolager et al. (2023) introduced *Sampling Where It Matters* (SWIM), a data-driven procedure for constructing the weights and biases of shallow and deep neural networks without gradient-based optimization. SWIM samples pairs of training points and constructs individual neurons from their relative positions, assigning higher sampling probability to pairs exhibiting large changes in the target function relative to their distance in feature space. In this way, features are preferentially placed in regions where the target function varies strongly. After the hidden layers have been constructed, only the final linear layer is fitted. The resulting sampled networks retain universal approximation properties and, in the experiments of Bolager et al. (2023), achieve accuracy comparable to conventionally trained networks while requiring substantially less training time.

A closely related combination of SWIM and Koopman operator approximation was recently proposed by Bolager et al. (2026a) for constructing *Koopman-informed recurrent neural networks*. There, SWIM is used to generate a data-adapted nonlinear lifting without backpropagation, while EDMD determines a finite-dimensional Koopman operator governing the evolution in the lifted space. A recurrent update is then obtained by repeatedly lifting the current state, evolving it linearly through the learned Koopman operator, and projecting it back to the state space. Consequently, both the nonlinear feature representation and the recurrent dynamics can be constructed without gradient-based training. This work demonstrates that SWIM-generated observables can be successfully coupled with Koopman approximations for dynamical-system learning and provides a particularly close precursor to the dictionary-construction strategy considered here.

Random Fourier features for Koopman learning. Randomized feature construction has consequently appeared in Koopman methods in several forms. Random Fourier features approximate

translation-invariant kernels with an explicit finite-dimensional feature map, avoiding the construction and factorization of the full kernel matrix. Importantly, their use in EDMD predates recent neural dictionary-learning approaches: random Fourier features and Nyström approximations were proposed as scalable approximations of kernel EDMD specifically to reduce the computational cost associated with large datasets (DeGennaro & Urban, 2019). Random Fourier dictionaries have since been applied to Koopman approximation in different settings (Bak et al., 2022; Nüske & Klus, 2023; Tabish et al., 2025; Sprink et al., 2026), and randomized shallow feature models have also been used directly to approximate Koopman eigenfunctions (Liao, 2026). These approaches replace the nonlinear optimization of a neural dictionary by the generation of a fixed collection of nonlinear observables followed by the usual linear operator fit.

Reservoir-based dictionaries. Reservoir computing follows a closely related philosophy. A randomly initialized recurrent network maps the input into a high-dimensional nonlinear dynamical representation, while its internal parameters remain fixed and only a linear readout is trained (Bollt, 2021; Gauthier et al., 2021). The resulting optimization is therefore typically reduced to a least-squares problem. In a Koopman context, reservoir states can themselves serve as observables, allowing the reservoir to provide a nonlinear dictionary without backpropagating through its internal weights. Reservoir-based methods have explicitly exploited this idea for Koopman operator approximation and dictionary construction using linear convex optimization (Gulina & Mauroy, 2021). The recurrent structure additionally provides an implicit representation of temporal history, distinguishing reservoir features from purely state-dependent random mappings.

B KOOPMAN THEORY

B.1 KOOPMAN OPERATOR

The Koopman operator provides a linear representation of potentially nonlinear dynamical systems by describing the evolution of observable functions rather than the state itself (Budišić et al., 2012).

Consider a dynamical system on a state space $\mathcal{X} \subseteq \mathbb{R}^d$ equipped with a measure μ, where $\boldsymbol{z}_t \in \mathcal{X}$ denotes the state at time t. Its evolution is governed by

$$\frac{d}{dt}\boldsymbol{z}_t = F(\boldsymbol{z}_t), \tag{7}$$

where $F : \mathcal{X} \mapsto \mathbb{R}^d$ is the vector field describing the instantaneous velocity at each state. Assuming that each initial state in $\mathcal{X}$ admits a unique solution that remains in $\mathcal{X}$ for all $t \geq 0$, the flow map $\varphi_t : \mathcal{X} \mapsto \mathcal{X}$ maps an initial state to its state at time t:

$$\boldsymbol{z}_t = \varphi_t(\boldsymbol{z}_0). \tag{8}$$

The system can therefore be described by the triplet $(\mathcal{X}, \mathcal{T}, \varphi_\cdot)$, where $\mathcal{T} \subseteq \mathbb{R}$ is the time domain. For forward-time evolution, the flow maps $\{\varphi_t\}_{t\geq 0}$ form a semigroup, satisfying

$$\varphi_0 = \mathrm{Id}_{\mathcal{X}}, \qquad \varphi_{t_1} \circ \varphi_{t_2} = \varphi_{t_1+t_2}. \tag{9}$$

Let $\mathcal{F}$, typically a Hilbert space, be a space of scalar-valued functions $g : \mathcal{X} \mapsto \mathbb{C}$ called *observables*, chosen such that $g \circ \varphi_t \in \mathcal{F}$ for every $g \in \mathcal{F}$ and $t \geq 0$. The Koopman operator $\mathcal{K}^t : \mathcal{F} \mapsto \mathcal{F}$ is defined by

$$\mathcal{K}^t g = g \circ \varphi_t, \qquad (\mathcal{K}^t g)(\boldsymbol{z}_0) = g(\varphi_t(\boldsymbol{z}_0)) = g(\boldsymbol{z}_t). \tag{10}$$

Thus, evolving the state for a time t and then evaluating g gives the same result as applying $\mathcal{K}^t$ to g and evaluating the resulting observable at the initial state. Although the underlying flow map may be nonlinear, the Koopman operator is linear in the observable:

$$\mathcal{K}^t(\alpha g_1 + \beta g_2) = \alpha \mathcal{K}^t g_1 + \beta \mathcal{K}^t g_2, \qquad g_1, g_2 \in \mathcal{F}, \quad \alpha, \beta \in \mathbb{C}. \tag{11}$$

The flow's semigroup property is inherited by $\{\mathcal{K}^t\}_{t\geq 0}$, called the *Koopman semigroup*.

This framework trades nonlinear evolution in a finite-dimensional state space for linear evolution in a typically infinite-dimensional function space, enabling the use of linear spectral methods to analyze nonlinear dynamics. A nonzero observable $\chi \in \mathcal{F}$ is an eigenfunction of $\mathcal{K}^t$ with eigenvalue λ if

$$(\mathcal{K}^t \chi)(\boldsymbol{z}_0) = \chi(\varphi_t(\boldsymbol{z}_0)) = \lambda \chi(\boldsymbol{z}_0), \qquad \forall \boldsymbol{z}_0 \in \mathcal{X}. \tag{12}$$

When a vector-valued observable admits an eigenfunction expansion, it can be expressed as a linear combination of eigenfunctions with vector coefficients called Koopman modes. These modes specify fixed directions in the observable's output space, while the eigenfunctions determine their state-dependent amplitudes.

For numerical approximation, a finite dictionary of observables is selected to span a finite-dimensional subspace of $\mathcal{F}$. Such a subspace is generally not invariant under the Koopman operator: evolving an observable in the subspace may produce a function outside it. Representing the evolution within the chosen subspace therefore introduces an approximation.

B.2 GEDMD ALGORITHM

Generator extended dynamic mode decomposition (gEDMD) approximates the Koopman generator on a finite-dimensional space of observables using derivative information (Klus et al., 2020).

For a strongly continuous Koopman semigroup $\{\mathcal{K}^t\}_{t\geq 0}$, the generator $\mathcal{L} : \mathcal{D} \to \mathcal{F}$ is defined by

$$\mathcal{L}g := \lim_{t \downarrow 0} \frac{\mathcal{K}^t g - g}{t}, \qquad g \in \mathcal{D}. \tag{13}$$

It describes the instantaneous rate of change of observables. The domain $\mathcal{D} \subseteq \mathcal{F}$ consists of those observables for which this limit exists in the norm of $\mathcal{F}$.

For the system $\dot{\boldsymbol{z}}_t = F(\boldsymbol{z}_t)$ and sufficiently differentiable $g \in \mathcal{D}$, the chain rule yields

$$(\mathcal{L}g)(\boldsymbol{z}) = \nabla g(\boldsymbol{z})^\top F(\boldsymbol{z}). \tag{14}$$

Along a trajectory, this gives

$$\frac{d}{dt} g(\boldsymbol{z}_t) = (\mathcal{L}g)(\boldsymbol{z}_t). \tag{15}$$

A nonzero observable $\chi \in \mathcal{D}$ is a generator eigenfunction with eigenvalue λ if

$$\mathcal{L}\chi = \lambda\chi. \tag{16}$$

It therefore evolves exponentially along trajectories:

$$\chi(\boldsymbol{z}_t) = (\mathcal{K}^t\chi)(\boldsymbol{z}_0) = e^{\lambda t}\chi(\boldsymbol{z}_0). \tag{17}$$

Thus, χ is also an eigenfunction of $\mathcal{K}^t$, with eigenvalue $e^{\lambda t}$. The real and imaginary parts of the generator eigenvalue λ describe its growth or decay rate and angular oscillation frequency, respectively.

To approximate the generator we gEDMD, choose a dictionary of sufficiently differentiable functions $\{\phi_i\}_{i=1}^N \subseteq \mathcal{D}$ spanning a finite-dimensional subspace of $\mathcal{F}$ and collect its values in

$$\Phi_N(\boldsymbol{z}) = \begin{bmatrix}\phi_1(\boldsymbol{z}) & \cdots & \phi_N(\boldsymbol{z})\end{bmatrix}^T. \tag{18}$$

Given M sampled states $\boldsymbol{z}_m \in \mathcal{X}$ and their corresponding velocities $\dot{\boldsymbol{z}}_m = F(\boldsymbol{z}_m)$, Eq. 14 yields

$$\dot{\phi}_i(\boldsymbol{z}_m) = (\mathcal{L}\phi_i)(\boldsymbol{z}_m) = \nabla\phi_i(\boldsymbol{z}_m)^\top \dot{\boldsymbol{z}}_m. \tag{19}$$

Here, $\dot{\phi}_i(\boldsymbol{z}) := (\mathcal{L}\phi_i)(\boldsymbol{z})$ denotes the derivative of ϕ_i along the dynamics. The velocities must be available from measurements or the vector field, or estimated from trajectory data. Define

$$\dot{\Phi}_N(\boldsymbol{z}) = \begin{bmatrix}\dot{\phi}_1(\boldsymbol{z}) & \cdots & \dot{\phi}_N(\boldsymbol{z})\end{bmatrix}^T, \tag{20}$$

and assemble the data matrices

$$\begin{aligned}\Phi_X &= \begin{bmatrix}\Phi_N(\boldsymbol{z}_1) & \cdots & \Phi_N(\boldsymbol{z}_M)\end{bmatrix}, \\ \dot{\Phi}_X &= \begin{bmatrix}\dot{\Phi}_N(\boldsymbol{z}_1) & \cdots & \dot{\Phi}_N(\boldsymbol{z}_M)\end{bmatrix}.\end{aligned} \tag{21}$$

Both matrices have dimensions $N \times M$.

The generator matrix approximation is obtained by solving a least-squares problem:

$$\hat{L}_{NM}^T \in \underset{B\in\mathbb{C}^{N\times N}}{\arg\min} \left\|\dot{\Phi}_X - B\Phi_X\right\|_F^2, \tag{22}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Its minimum Frobenius-norm solution is

$$\hat{L}_{NM}^T = \dot{\Phi}_X\Phi_X^+, \tag{23}$$

where $+$ denotes the Moore–Penrose pseudoinverse.

The transpose arises because dictionary values are arranged as column vectors.

Equivalently, define the empirical Galerkin matrices

$$\hat{\mathbf{C}}_{NM} = \frac{1}{M}\dot{\Phi}_X\Phi_X^*, \qquad \hat{\mathbf{G}}_{NM} = \frac{1}{M}\Phi_X\Phi_X^*, \tag{24}$$

where $*$ denotes the conjugate transpose. Their entries are

$$\begin{aligned}[\hat{\mathbf{C}}_{NM}]_{i,j} &= \frac{1}{M}\sum_{m=1}^M \overline{\phi_j(\boldsymbol{z}_m)}\mathcal{L}\phi_i(\boldsymbol{z}_m), \\ [\hat{\mathbf{G}}_{NM}]_{i,j} &= \frac{1}{M}\sum_{m=1}^M \overline{\phi_j(\boldsymbol{z}_m)}\phi_i(\boldsymbol{z}_m).\end{aligned} \tag{25}$$

The generator approximation can then be written as

$$\hat{L}_{NM}^T = \hat{\mathbf{C}}_{NM}\hat{\mathbf{G}}_{NM}^+, \tag{26}$$

which is equivalent to Eq. 23.

The resulting matrix enables spectral analysis and prediction of dictionary values through

$$\Phi_N(\boldsymbol{z}_t) \approx \exp\left(t\hat{L}_{NM}^T\right) \Phi_N(\boldsymbol{z}_0). \tag{27}$$

This finite-dimensional evolution is generally approximate because the dictionary span need not be invariant under the generator. Its accuracy also depends on the sampled states and the quality of the derivative information. Predicting states additionally requires a reconstruction map from dictionary values to state coordinates.

C DRAGONS SAMPLING DISTRIBUTION

By choosing the target function f from which we construct $\mathbb{P}_{\mathcal{X}}$, we can determine how the data informs the dictionary. Here, we provide further details on and the intuition behind the different sampling strategies we propose and evaluate for DRAGONS.

Flow-driven This is a natural choice since it is a direct application of the SWIM algorithm to our given dataset. Further, this has been thoroughly evaluated and empirically shown to perform well across various tasks in the Koopman-Informed Recurrent Neural Networks architecture proposed in Bolager et al. (2026a). Since our dataset consists of pairs $(\boldsymbol{z}_i, \varphi_{\tau_i}(\boldsymbol{z}_i))$, their strategy can immediately be used to place the neurons in regions of the state space where the flow map exhibits complex behaviour, indicated by large changes over small regions. These considerations yield the density

$$p_{\mathcal{X}}(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}) \propto \frac{\left\|\varphi_{\tau_2}(\boldsymbol{z}^{(2)}) - \varphi_{\tau_1}(\boldsymbol{z}^{(1)})\right\|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|}. \tag{28}$$

Notably, however, the observed future states are in general not advanced by the same time step in our setting, i.e. $\tau_1 \neq \tau_2$. This means that in the limit of $z^{(2)} \to z^{(1)}$, the above score does not converge to a directional derivative since the temporal offset introduces additional variation that is not explained by the spatial offset between the input points. This merits investigating alternative constructions.

Vector field-driven Since the network ultimately parameterises the vector field F rather than the flow map φ, we also consider using this as a sampling target. This is further motivated by the close relation of the Koopman generator to the vector field, as evidenced by the relationship $\mathcal{L}g = \langle \nabla g, F \rangle$. This suggests sampling according to

$$p_{\mathcal{X}}(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}) \propto \frac{\left\|F(\boldsymbol{z}^{(2)}) - F(\boldsymbol{z}^{(1)})\right\|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|}. \tag{29}$$

If the vector field is unavailable, it can be approximated from the flow map using a suitable finite-difference method.

Generator-driven Another route, motivated by dictionary learning, is to choose observables that faithfully represent the generator's action on the dictionary, analogous to how Li et al. (2017) construct the training loss by targeting invariant dictionaries. Using the explicit expression of the SWIM-neurons in terms of the sampled points, we can already compute the sampling distribution from how a candidate neuron would behave if sampled without actually having to construct and evaluate it. We now describe how to turn this into a sampling strategy targeting the generator, since this is the main object the neurons need to represent.

Define ϕ_{kl} as the SWIM-neuron constructed from the data points, $\boldsymbol{z}^{(k)}$ and $\boldsymbol{z}^{(l)}$, concretely $\phi_{kl}(\boldsymbol{z}) = \sigma(\boldsymbol{w}_{kl}^{\top}\boldsymbol{z} + b_{kl})$, where

$$\boldsymbol{w}_{kl} = s_1 \frac{\boldsymbol{z}^{(l)} - \boldsymbol{z}^{(k)}}{\left\|\boldsymbol{z}^{(l)} - \boldsymbol{z}^{(k)}\right\|^2}, \tag{30}$$

$$b_{kl} = -\left\langle \boldsymbol{w}_{kl}, \boldsymbol{z}^{(k)} \right\rangle + s_2. \tag{31}$$

An important consequence of the above is that $\phi_{kl}(\boldsymbol{z}^{(k)}) = \sigma(s_2)$ and $\phi_{kl}(\boldsymbol{z}^{(l)}) = \sigma(s_1 + s_2)$, which means that we can precompute $\nabla\phi_{kl}$ at the sampled points. Even further, for ReLU and Tanh as activation functions with the default values for s_1, s_2, $\sigma'(\boldsymbol{w}_{kl}^{\top}\boldsymbol{z}^{(k)} + b_{kl}) = \sigma'(\boldsymbol{w}_{kl}^{\top}\boldsymbol{z}^{(l)} + b_{kl})$, using the convention that $\mathrm{ReLU}'(x) = \mathbf{1}_{\{x \geq 0\}}(x)$.

With these considerations in mind, we can define a sampling distribution with $f = \mathcal{L}\phi_{12}$ through the density

$$p_{\mathcal{X}}(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}) \propto \frac{\left\|\mathcal{L}\phi_{12}(\boldsymbol{z}^{(2)}) - \mathcal{L}\phi_{12}(\boldsymbol{z}^{(1)})\right\|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|} \tag{32}$$

$$\propto \frac{\left|\left\langle \nabla\phi_{12}(\boldsymbol{z}^{(2)}), F(\boldsymbol{z}^{(2)})\right\rangle - \left\langle \nabla\phi_{12}(\boldsymbol{z}^{(1)}), F(\boldsymbol{z}^{(1)})\right\rangle\right|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|} \tag{33}$$

$$\propto \frac{\left|\sigma'(\boldsymbol{w}_{12}^{\top}\boldsymbol{z}^{(2)} + b_{12})\left\langle \boldsymbol{w}_{12}, F(\boldsymbol{z}^{(2)})\right\rangle - \sigma'(\boldsymbol{w}_{12}^{\top}\boldsymbol{z}^{(1)} + b_{12})\left\langle \boldsymbol{w}_{12}, F(\boldsymbol{z}^{(1)})\right\rangle\right|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|} \tag{34}$$

$$\propto \frac{\left|\left\langle \boldsymbol{w}_{12}, F(\boldsymbol{z}^{(2)})\right\rangle - \left\langle \boldsymbol{w}_{12}, F(\boldsymbol{z}^{(1)})\right\rangle\right|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|} \tag{35}$$

$$\propto \frac{\left|\left\langle s_1 \frac{\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|^2}, F(\boldsymbol{z}^{(2)}) - F(\boldsymbol{z}^{(1)})\right\rangle\right|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|} \tag{36}$$

$$\propto \frac{\left|(\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)})^{\top}(F(\boldsymbol{z}^{(2)}) - F(\boldsymbol{z}^{(1)}))\right|}{\left\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^{(1)}\right\|^3}, \tag{37}$$

where we absorbed the σ' term and s_1 into the normalisation constant. This can be intuitively understood as using the neuron to test how $\mathcal{L}$ varies over the space of observables.

D CONVERGENCE OF VECTOR FIELD APPROXIMATION AND RESULTING TRAJECTORIES

In this section, we give the necessary assumptions and proofs for the theoretical results about DRAGONS and gEDMD in the main paper.

D.1 ASSUMPTIONS & NOTATION

Following Bolager et al. (2026a), we define the state space $\mathcal{X}$ as in Bolager et al. (2023). We set

$$d(\boldsymbol{z}, A) = \inf\{d(\boldsymbol{z}, \boldsymbol{a}) \colon \boldsymbol{a} \in A\},$$

where d is the canonical Euclidean distance in the space $\mathbb{R}^{d_z}$. The medial axis is defined as

$$\text{Med}(A) = \{\boldsymbol{x} \in \mathbb{R}^d \colon \exists \boldsymbol{p} \neq \boldsymbol{q} \in A, \|\boldsymbol{p} - \boldsymbol{x}\| = \|\boldsymbol{q} - \boldsymbol{x}\| = d(\boldsymbol{x}, A)\}$$

and the reach is the scalar

$$\tau_A = \inf_{\boldsymbol{a} \in A} d(\boldsymbol{a}, \text{Med}(A)),$$

i.e., the point in A that is closest to the projection of points in A^c.

Definition 1. *Let $\widetilde{\mathcal{X}}$ be a nonempty compact subset of $\mathbb{R}^d$ with reach $\tau_{\widetilde{\mathcal{X}}} > 0$. The input space $\mathcal{X}$ is defined as*

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d \colon d(\boldsymbol{x}, \widetilde{\mathcal{X}}) \leq \epsilon_{\mathcal{X}}\},$$

where $0 < \epsilon_{\mathcal{X}} < \min\{\tau_{\widetilde{\mathcal{X}}}, 1\}$.

Remark 1. This restriction to the type of state and input spaces we consider is sufficient to construct all neural networks of interest by choosing pairs of points from the space in question, and construct the weight and bias as in Equation (2). It is also argued in Bolager et al. (2023) that most interesting real-world applications will contain some noise and make sure that the state and input spaces are approximately in the form given in the definition.

Furthermore, the convergence results from Llamazares-Elias et al. (2026), on which our convergence proofs are based, require the following assumptions, which we restate here in full for convenience before showing that sampled networks meet the requirements on the dictionary and giving a simple sufficient condition on the dynamical system for the assumptions.

Assumption 2. We assume the following:

2.1. The dictionary functions $\Phi_N = \{\phi_1, \ldots, \phi_N\} \subset \mathcal{D}$ are linearly independent.

2.2. The functions $\{\phi_n, \mathcal{L}\phi_n\}_{n=1}^N$ are continuous μ almost everywhere.

2.3. The points $\{\boldsymbol{x}_m\}_{m=1}^M \subset \mathcal{X}$ are i.i.d. samples from μ.

Assumption 3. The dictionary functions $\Phi_N = \{\phi_n\}_{n=1}^N$ satisfy $\phi_j \mathcal{L}\phi_i \in \mathcal{F}$ as well as $\phi_i^2 \in \mathcal{F}$ for all $i, j \in \{1, \ldots, N\}$.

Assumption 4.

$$\lim_{N \to \infty} \|\mathcal{P}_{\mathcal{F}_N} f - f\|_{\mathcal{F}} = 0, \quad \forall f \in \mathcal{F},$$

where $\mathcal{P}_{\mathcal{F}_N}$ is the projection of $\mathcal{F}$ onto $\mathcal{F}_N$ using the inner product on $\mathcal{F}$.

Assumption 5. $\mathcal{L}$ is a closed operator, and

$$\lim_{N \to \infty} \|\mathcal{P}_{\mathcal{D}_N} f - f\|_{\mathcal{D}} = 0, \quad \forall f \in \mathcal{D}.$$

Here, $\mathcal{P}_{\mathcal{D}_N}$ is the projection of $\mathcal{D}$ onto $\mathcal{F}_N$ using the inner product on $\mathcal{D}$.

Finally, we add the following standard assumption on the probability measure μ, which leads to simple sufficient conditions for the boundedness and strong continuity of the Koopman Operator semigroup.

Assumption 6. The measure μ has an everywhere positive and bounded density ρ w.r.t. the Lebesgue measure λ on $\mathcal{X}$, where $0 < \alpha \leq \rho(\boldsymbol{x}) \leq \beta < \infty, \forall \boldsymbol{x} \in \mathcal{X}$.

Notations. We further define the following notations to be used throughout this section.

- Space of observables $\mathcal{F} = L^2(\mathcal{X}, \mu)$, equipped with its standard inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ and norm $\|\cdot\|_{\mathcal{F}}$.
- Operator norm induced by the $\mathcal{F}$ norm $\|\mathcal{K}^t\|_{\mathcal{B}(\mathcal{F})}$.
- Span of the dictionary $\mathcal{F}_N = \text{span}(\Phi_N)$.
- Indicator functions $\mathbf{1}_{\mathbb{A}}(x) = \begin{cases} 1, x \in \mathbb{A}, \\ 0, x \notin \mathbb{A}. \end{cases}$
- The Euclidean inner product and norm, $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$, are denoted without subscript.

D.2 Validity of SWIM-sampled Dictionary

Next, we will show the validity of our dictionary and state simple, practically oriented, sufficient conditions for abstract properties that we require on the dynamical system.

Firstly, we require exponential boundedness of the Koopman operator $\mathcal{K}^t$ and closedness of the generator $\mathcal{L}$. Since both of these properties are fulfilled when the Koopman operators form a C_0-Semigroup, we provide the following lemma showing that this is, for example, attained in the case of Lipschitz Vector Fields. This is also the natural setting in which to work, since only then are we guaranteed the existence of a unique solution to the ODE by the Picard-Lindelöf Theorem.

Lemma 1 (Lipschitz Vector Fields have strongly continuous Koopman operators)**.** *Let $F(z) = (F_1(z), F_2(z), \dots, F_d(z))^\top$ be a Lipschitz continuous vector field with Lipschitz constant L, and $\mathcal{X}$ be forward invariant under the flow map φ_t, i.e. $\varphi_t(\mathcal{X}) \subset \mathcal{X}$. Then the Koopman operators form a strongly continuous semigroup. In specific, the Koopman generator $\mathcal{L}$ is a closed operator and $\mathcal{K}^t$ fulfills the exponential bound $\|\mathcal{K}^t\|_{\mathcal{B}(\mathcal{F})} \leq \sqrt{\frac{\beta}{\alpha}} e^{\omega t}$ with $\omega = \frac{d \cdot L}{2}$.*

Proof. We will first show exponential boundedness and then use it to prove strong continuity, from which the closedness of the generator immediately follows (Engel & Nagel, 2000, Theorem 1.4). Let $f \in \mathcal{F} = L^2(\mathcal{X}, \mu)$. Then

$$\left\|\mathcal{K}^t f\right\|_{\mathcal{F}}^2 = \int_{\mathcal{X}} f\left(\varphi_t(\boldsymbol{z}_0)\right)^2 \mathrm{d}\mu(\boldsymbol{z}_0) \tag{38}$$

$$= \int_{\mathcal{X}} f\left(\varphi_t(\boldsymbol{z}_0)\right)^2 \rho(\boldsymbol{z}_0) \mathrm{d}\lambda(\boldsymbol{z}_0) \tag{39}$$

$$= \int_{\varphi_t(\mathcal{X})} f\left(\boldsymbol{z}_t\right)^2 \left|\det \nabla \varphi_{-t}(\boldsymbol{z}_t)\right| \frac{\rho\left(\varphi_{-t}(\boldsymbol{z}_t)\right)}{\rho(\boldsymbol{z}_t)} \rho(\boldsymbol{z}_t) \mathrm{d}\lambda(\boldsymbol{z}_t) \qquad (\boldsymbol{z}_t = \varphi_t(\boldsymbol{z}_0)) \tag{40}$$

$$\leq \frac{\beta}{\alpha} \int_{\varphi_t(\mathcal{X})} f\left(\boldsymbol{z}_t\right)^2 \left|\det \nabla \varphi_{-t}(\boldsymbol{z}_t)\right| \rho(\boldsymbol{z}_t) \mathrm{d}\lambda(\boldsymbol{z}_t). \tag{41}$$

Next, for $z_t \in \varphi_t(\mathcal{X})$, we estimate the Jacobian determinant by

$$\left|\det \nabla \varphi_{-t}(\boldsymbol{z}_t)\right| = \prod_{i=1}^{d} \sigma_i(\nabla \varphi_{-t}(\boldsymbol{z}_t)) \leq \sigma_{\max}(\nabla \varphi_{-t}(\boldsymbol{z}_t))^d = \left\|\nabla \varphi_{-t}(\boldsymbol{z}_t)\right\|_{\mathcal{B}(\ell^2)}^d, \tag{42}$$

where the norm is the matrix operator norm induced by the ℓ^2 norm, i.e. the spectral norm. Since the operator norm of the Jacobian is bounded by the Lipschitz constant, we next show that φ_{-t} is Lipschitz continuous and estimate its constant. For $z_1, z_2 \in \varphi_t(\mathcal{X})$ and $\tau \leq t$ (so that the backwards

flow map is defined), we have

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\left\|\varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\|^2 = \frac{\mathrm{d}}{\mathrm{d}\tau}\left\langle \varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2),\ \varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\rangle \tag{43}$$

$$= 2\left\langle \frac{\mathrm{d}}{\mathrm{d}\tau}\varphi_{-\tau}(z_1)-\frac{\mathrm{d}}{\mathrm{d}\tau}\varphi_{-\tau}(z_2),\ \varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\rangle \tag{44}$$

$$= -2\left\langle F(\varphi_{-\tau}(z_1))-F(\varphi_{-\tau}(z_2)),\ \varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\rangle \tag{45}$$

$$\leq 2\left\|F(\varphi_{-\tau}(z_1))-F(\varphi_{-\tau}(z_2))\right\|\left\|\varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\| \tag{46}$$

$$\leq 2L\left\|\varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\|^2, \tag{47}$$

where we used Cauchy-Schwartz and the Lipschitzness of the vector field in the last two steps. Applying Grönwall's inequality to the above now yields

$$\left\|\varphi_{-\tau}(z_1)-\varphi_{-\tau}(z_2)\right\|^2 \leq \exp(2L\tau)\left\|\varphi_0(z_1)-\varphi_0(z_2)\right\|^2 = \exp(2L\tau)\left\|z_1-z_2\right\|^2, \tag{48}$$

which, after taking the square root, shows that φ_{-t} is $\exp(Lt)$-Lipschitz. The same argument shows that φ_t is also $\exp(Lt)$-Lipschitz. Bounding the (matrix) operator norm and combining with equation (42), we get

$$\left|\det \nabla\varphi_{-t}(\boldsymbol{z}_t)\right| \leq \exp(Ldt), \tag{49}$$

and together with the previous calculation

$$\left\|\mathcal{K}^t f\right\|_{\mathcal{F}}^2 \leq \frac{\beta}{\alpha}\exp(Ldt)\int_{\varphi_t(\mathcal{X})} f\left(\boldsymbol{z}_t\right)^2 \rho(\boldsymbol{z}_t)\mathrm{d}\lambda(\boldsymbol{z}_t) \tag{50}$$

$$\leq \frac{\beta}{\alpha}\exp(Ldt)\int_{\mathcal{X}} f\left(\boldsymbol{z}_t\right)^2 \mathrm{d}\mu(\boldsymbol{z}_t) = \frac{\beta}{\alpha}\exp(Ldt)\left\|f\right\|_{\mathcal{F}}^2. \tag{51}$$

Since f was arbitrary, taking the square root gives the exponential bound $\left\|\mathcal{K}^t\right\|_{\mathcal{B}(\mathcal{F})} \leq \sqrt{\frac{\beta}{\alpha}}e^{\omega t}$, where $\omega = \frac{d\cdot L}{2}$ as claimed.

Next, we show that the stated assumptions on μ and F lead to a C_0-Semigroup of Koopman operators. The identity and composition properties follow immediately from the definition of $\mathcal{K}^t$. Thus, we only need to show that it is continuous in t in the strong operator topology induced by the $\mathcal{F}$-norm, that is

$$\forall f \in \mathcal{F} : \lim_{t\downarrow 0}\left\|\mathcal{K}^t f - f\right\|_{\mathcal{F}} = 0, \ \text{ i.e. } \ \forall f\in\mathcal{F}, \forall\epsilon>0, \exists t>0 : \left\|\mathcal{K}^t f - f\right\|_{\mathcal{F}} < \epsilon. \tag{52}$$

Let $f \in \mathcal{F}$ and $\epsilon > 0$ be fixed. From the above we have that $\left\|\mathcal{K}^t\right\|_{\mathcal{B}(\mathcal{F})} < Me^{\omega t}$ where $\omega \in \mathbb{R}$ and w.l.o.g. $M \geq 1$. Further, because $C^0(\mathcal{X})$ is dense in $\mathcal{F}$ w.r.t. the L^2-norm and by the assumptions on the measure also w.r.t. the $\mathcal{F}$-norm, we can choose $g \in C^0$ s.t. $\left\|f-g\right\|_{\mathcal{F}} < \frac{\epsilon}{3M(1+\epsilon)} < \frac{\epsilon}{3}$. We then estimate

$$\left\|\mathcal{K}^t f - f\right\|_{\mathcal{F}} \leq \left\|\mathcal{K}^t(f-g)\right\|_{\mathcal{F}} + \left\|\mathcal{K}^t g - g\right\|_{\mathcal{F}} + \left\|g-f\right\|_{\mathcal{F}} \tag{53}$$

$$\leq \left\|\mathcal{K}^t\right\|_{\mathcal{B}(\mathcal{F})}\frac{\epsilon}{3M(1+\epsilon)} + \left\|g\circ\varphi_t - g\right\|_{\mathcal{F}} + \frac{\epsilon}{3}. \tag{54}$$

Next, because of the boundedness of $\mathcal{K}^t$, we can choose a t_0 s.t. for $t_0' \leq t_0$ $\left\|\mathcal{K}^{t_0'}\right\|_{\mathcal{B}(\mathcal{F})} < M(1+\epsilon)$. Further, we estimate

$$\left\|g\circ\varphi_t - g\right\|_{\mathcal{F}} \leq \sqrt{\mu(\mathcal{X})}\left\|g\circ\varphi_t - g\right\|_{\infty} = \max_{z\in\mathcal{X}}\left|g(\varphi_t(z))-g(z)\right|, \tag{55}$$

where the supremum is taken as a maximum because $|g(\varphi_t(z))-g(z)|$ is continuous in z as the composition of continuous functions (as shown above) and $\mathcal{X}$ is compact. Letting now $z^* = \arg\max_{z\in\mathcal{X}}|g(\varphi_t(z))-g(z)|$, by continuity $\exists\delta>0$ such that

$$\left\|\varphi_t(z^*)-z^*\right\| < \delta \implies \left|g(\varphi_t(z^*))-g(z^*)\right| < \frac{\epsilon}{3}. \tag{56}$$

Further, because $\varphi_t(z^*)$ is continuous in t, we can choose such a t_1 for which for all $t_1' \leq t_1$ $\left\|\varphi_{t_1'}(z^*)-z^*\right\| < \delta$. Finally, setting $t = \min(t_0, t_1)$ yields

$$\left\|\mathcal{K}^t f - f\right\|_{\mathcal{F}} < M(1+\epsilon)\frac{\epsilon}{3M(1+\epsilon)} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \tag{57}$$

$$= \frac{\epsilon}{3}+\frac{\epsilon}{3}+\frac{\epsilon}{3} = \epsilon, \tag{58}$$

completing the proof. □

Remark 2. Two special cases of the above are worth noting. Firstly, for finite time intervals of interest, we get a constant bound on the operator norm. Secondly, if $\mu = \frac{1}{\lambda(\mathcal{X})}\lambda$ is the normalised Lebesgue measure, then the operator fulfils the stronger bound $\|\mathcal{K}^t\|_{\mathcal{B}(\mathcal{F})} \leq e^{\omega t}$.

Next, we also need the following extension of existing results on the approximation capabilities of SWIM-sampled neural networks from Bolager et al. (2023).

Lemma 2 (Universal approximation of Sobolev functions)**.** *Let $g \in W^{1,2}(\mathcal{X}^\circ, \mu)$. Then, for any $\epsilon > 0$, there exists a SWIM-sampled network Φ with ReLU activation function ϕ, such that*

$$\|g - \Phi\|_{W^{1,2}} < \epsilon. \tag{59}$$

Proof. By the Meyers-Serrin Theorem (Meyers & Serrin, 1964), we can pick a $f \in C^1(\mathcal{X}^\circ)$ such that $\|f - g\|_{W^{1,2}} < \frac{\epsilon}{3}$. Further, by Czarnecki et al. (2017, Theorem 1), there exists a shallow network $\tilde{\Phi}$ with N neurons and ReLU activation function with $\left\|f - \tilde{\Phi}\right\|_{W^{1,2}} < \frac{\epsilon}{3}$. Thus, the only non-trivial step is extending the universal approximation of ReLU networks from Bolager et al. (2023, Lemma 3) to the $W^{1,2}$ norm. We again show this for unit-sampled networks before extending the result to SWIM by invoking Bolager et al. (2023, Lemma 4). In the setting of Bolager et al. (2023, Lemma 3), we can construct a network $\hat{\Phi}$ with $N_1 \leq 6N$ neurons and biases of the form $\hat{b}_{1,i} = \langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x}_i \rangle$ with $\boldsymbol{x}_i \in \mathcal{X}$ such that $\hat{\Phi} = \tilde{\Phi}$ on $\mathcal{X}$. Next, we construct the N_1 weights and biases of a unit sampled network Φ^U exactly as in the proof of the Lemma and investigate the per-neuron error on the weak gradients. Let

$$\hat{n}_i(\boldsymbol{x}) = \hat{w}_{2,i}\phi(\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i \rangle), \tag{60}$$

$$n_i(\boldsymbol{x}) = \hat{w}_{2,i}\phi(\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)} \rangle) \tag{61}$$

respectively denote i-th neuron in $\hat{\Phi}$ and Φ^U, where $\boldsymbol{x}_{0,i}^{(1)} \in \mathcal{X}^\circ$ was chosen such that $\left\|\boldsymbol{x}_i - \boldsymbol{x}_{0,i}^{(1)}\right\| < \delta$ with a δ s.t. by continuity $\left|\phi(\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)} \rangle) - \phi(\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i \rangle)\right| < \frac{\epsilon}{6N_1 w_2 \mu(\mathcal{X})}$, $\forall \boldsymbol{x} \in \mathcal{X}$ and $w_2 = \max_{1 \leq i \leq N_1} |\hat{w}_{2,i}|$. Their weak gradients are thus

$$\nabla \hat{n}_i(\boldsymbol{x}) = \hat{w}_{2,i}\hat{\boldsymbol{w}}_{1,i}\mathbf{1}_{\{\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i \rangle \geq 0\}}(\boldsymbol{x}), \tag{62}$$

$$\nabla n_i(\boldsymbol{x}) = \hat{w}_{2,i}\hat{\boldsymbol{w}}_{1,i}\mathbf{1}_{\left\{\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)} \rangle \geq 0\right\}}(\boldsymbol{x}), \tag{63}$$

with error (the dependence on δ will be shown in what follows)

$$\varepsilon_i^2(\delta) = \|\nabla \hat{n}_i(\boldsymbol{x}) - \nabla n_i(\boldsymbol{x})\|_{L^2}^2 \tag{64}$$

$$= \|\hat{\boldsymbol{w}}_{1,i}\|^2 |\hat{w}_{2,i}|^2 \int_{\mathcal{X}} \left|\mathbf{1}_{\{\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i \rangle \geq 0\}}(\boldsymbol{x}) - \mathbf{1}_{\left\{\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)} \rangle \geq 0\right\}}(\boldsymbol{x})\right|^2 \mathrm{d}\mu(\boldsymbol{x}) \tag{65}$$

$$\leq \|\hat{\boldsymbol{w}}_{1,i}\|^2 |\hat{w}_{2,i}|^2 \beta \int_{\mathcal{X}} \left|\mathbf{1}_{\{\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i \rangle \geq 0\}}(\boldsymbol{x}) - \mathbf{1}_{\left\{\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)} \rangle \geq 0\right\}}(\boldsymbol{x})\right|^2 \mathrm{d}\lambda(\boldsymbol{x}). \tag{66}$$

Inspection of the integrand reveals that it evaluates to 1 exactly on the region enclosed between two parallel hyperplanes and 0 otherwise, and can thus be written as the indicator function for this region. Concretely, let

$$\hat{\mathbb{H}}_i = \{\boldsymbol{x} \in \mathcal{X} \mid \langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i \rangle = 0\}, \tag{67}$$

$$\mathbb{H}_i = \left\{\boldsymbol{x} \in \mathcal{X} \mid \langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)} \rangle = 0\right\}, \tag{68}$$

$$\mathbb{B}_i(\delta) = \left\{\boldsymbol{x} \in \mathcal{X} \mid \min\left(\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x}_i \rangle, \langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x}_{0,i}^{(1)} \rangle\right) \leq \langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} \rangle \leq \max\left(\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x}_i \rangle, \langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x}_{0,i}^{(1)} \rangle\right)\right\}. \tag{69}$$

Informally, $\mathbb{B}_i(\delta)$ denotes exactly the region inside $\mathcal{X}$ between $\hat{\mathbb{H}}_i$ and $\mathbb{H}_i$. Furthermore, the perpendicular distance $d(\hat{\mathbb{H}}_i, \mathbb{H}_i)$ along the shared normal $\hat{\boldsymbol{w}}_{1,i}$ is given by

$$d(\hat{\mathbb{H}}_i, \mathbb{H}_i) = \frac{\left|\langle \hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x}_i - \boldsymbol{x}_{0,i}^{(1)} \rangle\right|}{\hat{\boldsymbol{w}}_{1,i}} \leq \left\|\boldsymbol{x}_i - \boldsymbol{x}_{0,i}^{(1)}\right\| \leq \delta. \tag{70}$$

By the properties of the Lebesgue measure, we thus have $\lambda(\mathbb{B}_i(\delta)) < \delta\lambda^{d_z-1}\left(\mathbb{B}_i(\delta) \cap \{\hat{\boldsymbol{w}}_{1,i}\}^{\perp}\right)$, i.e. $\lambda(\mathbb{B}_i(\delta)) < \delta C_i$ with C_i finite by compactness of $\mathcal{X}$ and independent of δ. After these considerations, we can now bound

$$\varepsilon_i^2(\delta) \leq \|\hat{\boldsymbol{w}}_{1,i}\|^2 |\hat{w}_{2,i}|^2 \beta \int_{\mathcal{X}} \mathbf{1}_{\mathbb{B}_i(\delta)}(\boldsymbol{x}) \,\mathrm{d}\lambda(\boldsymbol{x}) \tag{71}$$

$$= \|\hat{\boldsymbol{w}}_{1,i}\|^2 |\hat{w}_{2,i}|^2 \beta\lambda(\mathbb{B}_i(\delta)) \tag{72}$$

$$< \|\hat{\boldsymbol{w}}_{1,i}\|^2 |\hat{w}_{2,i}|^2 \beta\delta C_i, \tag{73}$$

and respectively

$$\varepsilon_i(\delta) \leq w_1 w_2 \sqrt{\beta C \delta}. \tag{74}$$

Defining $C = \max_{1\leq i\leq N_1} C_i$ and $w_1 = \max_{1\leq i\leq N_1} \|\hat{\boldsymbol{w}}_{1,i}\|$, we now choose

$$\delta' = \min\left(\delta, \frac{\epsilon^2}{36 N_1^2 w_1^2 w_2^2 C\beta}\right), \tag{75}$$

giving

$$\sqrt{\delta'} < \frac{\epsilon}{6N_1 w_1 w_2 \sqrt{\beta C}}, \tag{76}$$

and further

$$\left\|\tilde{\Phi} - \Phi^U\right\|_{W^{1,2}} = \left\|\hat{\Phi} - \Phi^U\right\|_{W^{1,2}} = \left\|\sum_{i=1}^{N_1} \hat{n}_i - \sum_{i=1}^{N_1} n_i\right\|_{W^{1,2}} \tag{77}$$

$$\leq \sum_{i=1}^{N_1} \|\hat{n}_i - n_i\|_{W^{1,2}} = \sum_{i=1}^{N_1} \|\hat{n}_i - n_i\|_{L^2} + \|\nabla\hat{n}_i - \nabla n_i\|_{L^2} \tag{78}$$

$$= \sum_{i=1}^{N_1} |\hat{w}_{2,i}| \sqrt{\int_{\mathcal{X}} \left|\phi(\langle\hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_{0,i}^{(1)}\rangle) - \phi(\langle\hat{\boldsymbol{w}}_{1,i}, \boldsymbol{x} - \boldsymbol{x}_i\rangle)\right|^2 \mathrm{d}\mu(\boldsymbol{x})} + \varepsilon_i(\delta') \tag{79}$$

$$< \sum_{i=1}^{N_1} w_2 \frac{\epsilon}{6N_1 w_2 \mu(\mathcal{X})} \mu(\mathcal{X}) + w_1 w_2 \sqrt{C\beta\delta'} < \sum_{i=1}^{N_1} \frac{\epsilon}{6N_1} + \frac{\epsilon}{6N_1} = \frac{\epsilon}{3}. \tag{80}$$

Finally, by Bolager et al. (2023, Lemma 4), we choose a SWIM sampled Φ s.t. $\Phi^U = \Phi$ on $\mathcal{X}$ and thus $\left\|\tilde{\Phi} - \Phi\right\|_{W^{1,2}} < \frac{\epsilon}{3}$. With this, we get

$$\|g - \Phi\|_{W^{1,2}} \leq \|g - f\|_{W^{1,2}} + \left\|f - \tilde{\Phi}\right\|_{W^{1,2}} + \left\|\tilde{\Phi} - \Phi\right\|_{W^{1,2}} < \epsilon, \tag{81}$$

completing the proof. □

Now we are ready to show that our SWIM-sampled dictionary fulfils the above assumptions, meaning that we may invoke the results from Llamazares-Elias et al. (2026), which we summarise by the following lemma.

Lemma 3 (Validity of SWIM-sampled Dictionary)**.** *Let $\mathcal{K}^t$ be a C_0 semigroup, μ be as in Assumption 6 with the data being i.i.d. samples from μ. Then, there exists a sequence of (N, M_N) pairs such that for $M'_N \geq M$ almost surely $\hat{\mathcal{L}}_{NM'_N}\mathcal{P}_{\mathcal{D}_N} \to \mathcal{L}$ pointwise in $\mathcal{D}$ when using a dictionary of SWIM-sampled ReLU neurons.*

Proof. Since obviously $\mu \ll \lambda$, we have (see Llamazares-Elias et al. (2026) for details) that $\mathcal{D} = W^{1,2}(\mathcal{X}^\circ, \mu)$. For

$$\phi_i(\boldsymbol{x}) = \max(0, \boldsymbol{w}_i^\top \boldsymbol{x} - b_i), \tag{82}$$

we have the representation in the sense of weak derivatives

$$\mathcal{L}\phi_i(\boldsymbol{x}) = \langle\nabla\phi_i(\boldsymbol{x}), F(\boldsymbol{x})\rangle = \mathbf{1}_{\{\langle\boldsymbol{w}_i, \boldsymbol{x}\rangle \geq b_i\}}(\boldsymbol{x}) \sum_{k=1}^{d_z} \boldsymbol{w}_{ik} F_k(\boldsymbol{x}). \tag{83}$$

Regarding Assumption 2.1, given enough data, SWIM can produce neurons with weight vectors of arbitrary orientation and arbitrary biases inside the domain. Therefore, for any N, SWIM can produce a dictionary of N linearly independent functions using $M \leq 2N$ data points. Furthermore, all ϕ_i are in $\mathcal{D}$ as they are continuous and thus bounded and therefore square integrable on the compact set $\mathcal{X}$, while their weak derivatives are bounded and therefore again square integrable. Regarding Assumption 2.2, ϕ_i are continuous everywhere on $\mathcal{X}$, while $\mathcal{L}\phi_i$ are only discontinous on the λ-null set $\{\boldsymbol{x} \in \mathcal{X} \mid \langle \boldsymbol{w}_i, \boldsymbol{x} \rangle = b_i\}$ which by absolute continuity is also a μ-null set.

Regarding Assumption 3, $\phi_i^2 \in \mathcal{F}$ again follows from continuity. Further, again by continuity of ϕ and boundedness of $\mathbf{1}(\boldsymbol{x})$, we have that

$$\phi_j(\boldsymbol{x})\mathcal{L}\phi_i(\boldsymbol{x}) = \phi_j(\boldsymbol{x})\mathbf{1}_{\{\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle \geq b_i\}}(\boldsymbol{x}) \sum_{k=1}^{d_z} \boldsymbol{w}_{ik} F_k(\boldsymbol{x}) \tag{84}$$

is square integrable by being the product of a bounded and a square integrable function.

Regarding Assumption 4 and Assumption 5, the convergence requirements follow immediately from the density of SWIM neural networks in $\mathcal{F}$ (Bolager et al., 2023, Theorem 1) and $W^{1,2}$ (Lemma 2). The requirement that $\mathcal{L}$ is a closed operator is fulfilled by assumption.

The claim now follows from Theorem 5.1 in Llamazares-Elias et al. (2026). □

Remark 3. This result and thus all the following naturally extend to other dictionaries, training methods or activation functions as long as they fulfil an analogue of Lemma 2.

D.3 Convergence Results

We are now ready to show our main results, namely – informally – that, the predicted vector field $\hat{F}$ converges to the true vector field F and the resulting trajectories $\hat{z}$ to the true trajectories z.

Theorem 1 (Convergence of the approximate vector field)**.** *Let the assumptions be as in Lemma 3. Then $\forall \epsilon > 0$, there are constants $N, M_N \in \mathbb{N}$, a SWIM sampled hidden layer Φ_N and outer weights $\boldsymbol{c}^i$ such that for any $M'_N \geq M_N$ almost surely*

$$\left\| \hat{F}_i - F_i \right\|_{\mathcal{F}} = \left\| \sum_{j=1}^{N} c_j^i \hat{\mathcal{L}}_{NM'_N} \phi_j - F_i \right\|_{\mathcal{F}} < \epsilon, \quad \forall i \in \{1, 2, \ldots, d\}. \tag{4}$$

Proof.

$$\left\| \hat{F}_i - F_i \right\|_{\mathcal{F}} = \left\| \sum_{j=1}^{N} c_j^i \hat{\mathcal{L}}_{NM'_N} \phi_j - F_i \right\|_{\mathcal{F}} \tag{85}$$

$$= \left\| \hat{\mathcal{L}}_{NM'_N} \sum_{j=1}^{N} c_j^i \phi_j - \mathcal{L} e_i \right\|_{\mathcal{F}} \tag{86}$$

$$\leq \left\| \hat{\mathcal{L}}_{NM'_N} \sum_{j=1}^{N} c_j^i \phi_j - \hat{\mathcal{L}}_{NM'_N} \mathcal{P}_{\mathcal{D}_N} e_i \right\|_{\mathcal{F}} + \left\| \hat{\mathcal{L}}_{NM'_N} \mathcal{P}_{\mathcal{D}_N} e_i - \mathcal{L} e_i \right\|_{\mathcal{F}} \tag{87}$$

$$< \epsilon, \tag{88}$$

where in the last step the first term is identically zero and the second term can be controlled by Lemma 3 because the coordinate functions e and dictionary functions ϕ are in $\mathcal{D}$ if the components of F are in L^2 as per assumption. □

For the convergence of the trajectories, we need the approximate semigroups to be exponentially bounded so that they can not arbitrarily amplify errors in finite time. The following example shows that this property is, in general, not preserved by Galerkin methods such as g-EDMD even when the true semigroup is, and the generators converge.

Theorem 3 (gEDMD does not preserve exponential boundedness)**.** *There are systems for which the true semigroup $\mathcal{K}^t$ is exponentially bounded and the sequence of g-EDMD approximants $\hat{\mathcal{L}}_{NM'_N}$ converges pointwise on $\mathcal{D}$ to $\mathcal{L}$, while $\nexists W, \omega \in \mathbb{R}$ such that*

$$\left\| \exp\left(\hat{\mathcal{L}}_{NM'_N} t\right) \right\|_{\mathcal{B}(\mathcal{F}_N)} \leq W e^{\omega t}, \quad \forall N \in \mathbb{N}, \tag{6}$$

i.e. the sequence of approximate semigroups can not be bounded exponentially by uniform constants.

Proof. We construct a system with a Lipschitz vector field and a sequence of dictionaries in the setting of Lemma 1 fulfilling Assumptions 2-6, such that we know from Llamazares-Elias et al. (2026) and Lemma 1 that the stated convergence and boundedness are fulfilled while showing that the matrix representing $\hat{\mathcal{L}}_{NM'_N}$ has an eigenvalue lower bounded by N for each $N \in \mathbb{N}$, preventing the generated semigroups from sharing a uniform growth-bound ω. Consider the system

$$\mathcal{X} = [-1, 1], \quad \dot{x} = -x \tag{89}$$

with the observables $\mathcal{F} = L^2(\mathcal{X}, \mu)$, where μ has the density $\rho(x) = 1 + \sqrt{\left|x - \frac{1}{2}\right|}$ with respect to the Lebesgue measure λ. Clearly, we are in the setting of Lemma 1, so the Koopman operators form an exponentially bounded C_0-Semigroup. The associated Koopman generator is of the form

$$\mathcal{L}f = -x \frac{\mathrm{d}f}{\mathrm{d}x}. \tag{90}$$

We will now show that a valid but poorly chosen dictionary can cause the stiffness term $\langle \mathcal{L}u, u \rangle$ and thus an eigenvalue of the approximations $\mathcal{K}_N^t = \exp(\mathcal{L}_N t)$ to grow unboundedly. Away from the singularity at $x_0 = \frac{1}{2}$, the derivative of ρ is continuously differentiable and given by

$$\rho'(x) = \frac{\operatorname{sign}(x - x_0)}{2\sqrt{|x - x_0|}}, \tag{91}$$

however, $\rho'(x) \to \infty$ as $x \downarrow x_0$. We set up the sequence of dictionaries as $\Psi_N = \{e_1, \ldots, e_{N-1}, e_N\}$, where $\{e_1, \ldots, e_{N-1}\}$ are a system of orthonormal polynomials fulfilling the density requirements in $\mathcal{F}$ and $\mathcal{D}$, while we choose e_N to be the orthonormalisation of a standard bump function moving towards the singularity of ρ' as N increases. Concretely, with $\varphi \in \mathcal{C}_C^\infty(-1, 1)$ being a standard bump, let

$$\varphi_N(x) = \varphi\left(\frac{x - x_N}{\delta_N}\right), \quad \psi_N = \varphi_N / \left\|\varphi_N\right\|_{\mathcal{F}}, \tag{92}$$

and set

$$e_N = \frac{1}{z_N}\left(\psi_N - \sum_{k=1}^{N-1} \langle e_i, \psi_N \rangle_{\mathcal{F}} e_i\right) := \frac{1}{z_N}(\psi_N - r_N), \tag{93}$$

where the normalisation constant z_N is chosen so that $\|e_N\|_{\mathcal{F}} = 1$, and $x_N > x_0, \delta_N \in (0, \frac{x_n - x_0}{2}]$ will be specified below. It is easy to verify that Ψ_N fulfils all the assumptions for Theorem 5.1 from Llamazares-Elias et al. (2026), meaning that the approximate generators will indeed converge pointwise to the true generator. Next, we observe that $S_N := \operatorname{supp}(\psi_N) = \operatorname{supp}(\psi'_N) = [x_N - \delta_N, x_N + \delta_N]$ and because the functions in Ψ_N are mutually orthonormal, the population generator matrix simplifies to $L_N = C_N G_N^{-1} = C_N$, so we only need to estimate the eigenvalues of C_N in what follows. Now follows a sequence of estimates that will finally serve to bound the eigenvalues

by invoking the Gershgorin Theorem. Firstly,

$$\langle \mathcal{L}\psi_N, \psi_N \rangle_{\mathcal{F}} = \int_{S_N} -x\psi_N'(x)\psi_N(x)\rho(x)\mathrm{d}x \tag{94}$$

$$= -\frac{1}{2}\int_{S_N} x\frac{\mathrm{d}}{\mathrm{d}x}\left[\psi_N(x)^2\right]\rho(x)\mathrm{d}x \tag{95}$$

$$= \frac{1}{2}\int_{S_N} \psi_N(x)^2(\rho(x) + x\rho'(x))\mathrm{d}x \tag{96}$$

$$= \frac{1}{2}\int_{S_N} \psi_N(x)^2 \underbrace{\left(1 + \frac{x\rho'(x)}{\rho(x)}\right)}_{:=g(x)} \mathrm{d}\mu(x) \tag{97}$$

$$\geq \frac{1}{2}\min_{x\in S_N} g(x) \int_{S_N} \psi_N(x)^2 \mathrm{d}\mu(x) \geq \frac{1}{4}g(x_N) := g_N, \tag{98}$$

for δ_N small enough, where it was used that ψ_N has unit norm and g is continuous on S_N. Next, for $k < N$

$$\langle \mathcal{L}\psi_N, e_k \rangle_{\mathcal{F}} = -\int_{S_N} x\psi_N'(x)e_k(x)\rho(x)\,\mathrm{d}x \tag{99}$$

$$= \int_{S_N} \psi_N(x)\left[xe_k(x)\rho(x)\right]'\mathrm{d}x \tag{100}$$

$$= \int_{S_N} \psi_N(x)\left(xe_k'(x) + e_k(x) + e_k(x)\frac{x\rho'(x)}{\rho(x)}\right)\mathrm{d}\mu(x) \tag{101}$$

$$= \int_{S_N} \psi_N(x)\left(xe_k'(x) + e_k(x)g(x)\right)\mathrm{d}\mu(x), \tag{102}$$

$$\left|\langle \mathcal{L}\psi_N, e_k \rangle_{\mathcal{F}}\right| \leq \int_{S_N} |\psi_N(x)|\left(|xe_k'(x)| + |e_k(x)g(x)|\right)\mathrm{d}\mu(x) \tag{103}$$

$$\leq E_k(1 + \kappa_N g_N)\int_{S_N} |\psi_N(x)|\cdot 1\,\mathrm{d}\mu(x) \tag{104}$$

$$\leq E_k(1 + \kappa_N g_N)\left\|\psi_N\right\|_{\mathcal{F}}\sqrt{\mu(S_N)} = E_k(1 + \kappa_N g_N)\sqrt{\mu(S_N)}, \tag{105}$$

where we defined $E_k := \|e_k\|_{\mathcal{C}^1} = \|e_k\|_\infty + \|e_k'\|_\infty$ and $\kappa_N = \sup_{x\in S_N}\frac{g(x)}{g(x_N)}$, which exists by continuity and can be bounded independent of δ_N, because δ_N is bounded above. Similarly, we have

$$\left|\langle \mathcal{L}e_k, \psi_N \rangle_{\mathcal{F}}\right| = \left|\int_{S_N} xe_k'(x)\psi_N(x)\,\mathrm{d}\mu(x)\right| \leq E_k\int_{S_N} |\psi_N(x)|\,\mathrm{d}\mu(x) \leq E_k\sqrt{\mu(S_N)}, \tag{106}$$

and

$$\left|\langle \psi_N, e_k \rangle_{\mathcal{F}}\right| \leq \int_{S_N} |\psi_N|\,|e_k|\,\mathrm{d}\mu \leq E_k\sqrt{\mu(S_N)}. \tag{107}$$

The previous estimates pertain to the behaviour of ψ_N, which we now need to adapt for $e_N = \frac{1}{z_N}(\psi_N - r_N)$. From (107) and using orthonormality of the dictionary, we get

$$\|r_N\|_{\mathcal{F}}^2 = \langle r_N, r_N \rangle_{\mathcal{F}} = \sum_{k=1}^{N-1}\langle e_k, \psi_N \rangle_{\mathcal{F}}^2 \leq NE^2\mu(S_N), \tag{108}$$

with $E := \max_{k<N} E_k$, which implies that

$$-\sqrt{N}E\sqrt{\mu(S_N)} \leq -\|r_N\|_{\mathcal{F}} \leq \|r_N\|_{\mathcal{F}} \leq \sqrt{N}E\sqrt{\mu(S_N)}. \tag{109}$$

Using Bessel's and the reverse triangle inequality, we can bound z_N by

$$1 \geq z_N = \|\psi_N - r_N\|_{\mathcal{F}} \geq \left|\|\psi_N\|_{\mathcal{F}} - \|r_N\|_{\mathcal{F}}\right| \geq 1 - \sqrt{N}E\sqrt{\mu(S_N)}. \tag{110}$$

Further observing that $\langle e_k, r_N\rangle_{\mathcal{F}} = \langle e_k, \psi_N\rangle_{\mathcal{F}}$, we get $\left|\langle e_k, r_N\rangle_{\mathcal{F}}\right| \leq E_k\sqrt{\mu(S_N}$. Based on the previous estimations and defining $E' = \max_{k,j<N}\left|\langle \mathcal{L}e_k, e_j\rangle_{\mathcal{F}}\right|$, straightforward calculations yield

$$\left|\langle \mathcal{L}r_N, e_k\rangle_{\mathcal{F}}\right| = \left|\sum_{j=1}^{N-1}\langle e_j, \psi_N\rangle_{\mathcal{F}}\langle \mathcal{L}e_j, e_k\rangle_{\mathcal{F}}\right| \leq NEE'\sqrt{\mu(S_N)}, \tag{111}$$

$$\left|\langle \mathcal{L}e_k, r_N\rangle_{\mathcal{F}}\right| = \left|\sum_{j=1}^{N-1}\langle e_j, \psi_N\rangle_{\mathcal{F}}\langle \mathcal{L}e_k, e_j\rangle_{\mathcal{F}}\right| \leq NEE'\sqrt{\mu(S_N)}, \tag{112}$$

$$\left|\langle \mathcal{L}\psi_N, r_N\rangle_{\mathcal{F}}\right| = \left|\sum_{j=1}^{N-1}\langle e_j, \psi_N\rangle_{\mathcal{F}}\langle \mathcal{L}\psi_N, e_j\rangle_{\mathcal{F}}\right| \leq E\sqrt{\mu(S_N)}\left|\sum_{j=1}^{N-1}\langle \mathcal{L}\psi_N, e_j\rangle_{\mathcal{F}}\right| \tag{113}$$

$$\leq NE^2\sqrt{\mu(S_N)}(1+\kappa_N g_N), \tag{114}$$

$$\left|\langle \mathcal{L}r_N, \psi_N\rangle_{\mathcal{F}}\right| = \left|\sum_{j=1}^{N-1}\langle e_j, \psi_N\rangle_{\mathcal{F}}\langle \mathcal{L}e_j, \psi_N\rangle_{\mathcal{F}}\right| \leq NE^2\sqrt{\mu(S_N)}, \tag{115}$$

$$\left|\langle \mathcal{L}r_N, r_N\rangle_{\mathcal{F}}\right| = \left|\sum_{j=1}^{N-1}\langle e_j, \psi_N\rangle_{\mathcal{F}}\langle \mathcal{L}e_j, r_N\rangle_{\mathcal{F}}\right| \leq N^2E^2E'\sqrt{\mu(S_N)}, \tag{116}$$

which again implies

$$-N^2E^2E'\sqrt{\mu(S_N)} \leq \langle Lr_N, r_N\rangle_{\mathcal{F}} \leq N^2E^2E'\sqrt{\mu(S_N)}. \tag{117}$$

We can now use these estimates to bound the entries of the matrix C_N for $k, j < N$ by

$$|(C_N)_{kj}| = \left|\langle \mathcal{L}e_k, e_j\rangle_{\mathcal{F}}\right| \leq E', \tag{118}$$

$$|(C_N)_{Nk}| = \left|\langle \mathcal{L}e_N, e_k\rangle_{\mathcal{F}}\right| \leq \frac{1}{z_N}\left(\left|\langle \mathcal{L}\psi_N, e_k\rangle_{\mathcal{F}}\right| + \left|\langle \mathcal{L}r_N, e_k\rangle_{\mathcal{F}}\right|\right) \tag{119}$$

$$\leq \frac{1}{z_N}\left(E(1+\kappa_N g_N)\sqrt{\mu(S_N)} + NE'E\sqrt{\mu(S_N)}\right) \tag{120}$$

$$= \frac{1}{z_N}E\sqrt{\mu(S_N)}\left(1+\kappa_N g_N + NE'\right), \tag{121}$$

$$|(C_N)_{kN}| = \left|\langle \mathcal{L}e_k, e_N\rangle_{\mathcal{F}}\right| \leq \frac{1}{z_N}\left(\left|\langle \mathcal{L}e_k, \psi_N\rangle_{\mathcal{F}}\right| + \left|\langle \mathcal{L}e_k, r_N\rangle_{\mathcal{F}}\right|\right) \tag{122}$$

$$\leq \frac{1}{z_N}E\sqrt{\mu(S_N)}\left(1+NE'\right), \tag{123}$$

$$(C_N)_{NN} = \frac{1}{z_N^2}\langle \mathcal{L}\psi_N - \mathcal{L}r_N, \psi_N - r_N\rangle_{\mathcal{F}} \tag{124}$$

$$= \frac{1}{z_N^2}\left(\langle \mathcal{L}\psi_N, \psi_N\rangle_{\mathcal{F}} - \langle \mathcal{L}\psi_N, r_N\rangle_{\mathcal{F}} - \langle \mathcal{L}r_N, \psi_N\rangle_{\mathcal{F}} + \langle \mathcal{L}r_N, r_N\rangle_{\mathcal{F}}\right) \tag{125}$$

$$\geq \frac{1}{z_N^2}\left(g_N - NE^2\mu(S_N)(1+\kappa_N g_N) - NE^2\mu(S_N) - N^2E^2E'\sqrt{\mu(S_N)}\right). \tag{126}$$

The concluding step will be to invoke the Gershgorin Circle Theorem to show the existence of an eigenvalue with growing real part by choosing a sequence of dictionaries parameterised by x_N, δ_N such that one Gershgorin Circle is disjoint from the others and "moves towards infinity". Concretely, consider the Gershgorin Disks resulting from row estimates

$$\mathbb{D}_i = \mathbb{K}((C_N)_{ii}, R_i) := \{z \in \mathbb{C} \mid |z - (C_N)_{ii}| \leq R_i\}, \tag{127}$$

$$R_i = \sum_{j\neq i}|(C_N)_{ij}|. \tag{128}$$

For $k < N$, we get the radii

$$R_k \leq (N-2)E' + \frac{1}{z_N} E\sqrt{\mu(S_N)}(1 + NE'), \tag{129}$$

$$\leq R^* := R_k + 2E' = NE' + \frac{1}{z_N} E\sqrt{\mu(S_N)}(1 + NE'), \tag{130}$$

which, together with the bound $(C_N)_{kk} \leq E'$ gives

$$\mathbb{D}_k \subseteq \mathbb{D}^* := \mathbb{K}(E', R^*), \quad k = 1, \ldots, N-1, \tag{131}$$

since the centres can be at most $2E'$ apart. Analogously,

$$R_N = \sum_{k=1}^{N-1} |(C_N)_{Nk}| \leq (N-1)\frac{1}{z_N} E\sqrt{\mu(S_N)}\,(1 + \kappa_N g_N + NE'), \tag{132}$$

which together with the estimate for $(C_N)_{NN}$ defines the remaining disk. Finally, to enforce the sought behaviour for the remaining disk and thus the last eigenvalue, we can, because of the singularity of $\rho'(x)$ and respectively $g(x)$, choose x_N such that

$$g_N \geq N + E' + NE' + 1, \tag{133}$$

and afterwards δ_N such that

$$S_N \subsetneq [-1, 1], \tag{134}$$

$$\wedge\, \frac{1}{2} \min_{x \in S_N} g(x) \geq \frac{1}{4} g(x_N), \tag{135}$$

$$\wedge\, -\sqrt{N} E\sqrt{\mu(S_N)} \geq -\frac{1}{2}, \tag{136}$$

$$\wedge\, 2E\sqrt{\mu(S_N)}(1 + NE') \leq \frac{1}{2}, \tag{137}$$

$$\wedge\, -4\left(NE^2\mu(S_N)(1 + \kappa_N g_N) - NE^2\mu(S_N) - N^2E^2E'\sqrt{\mu(S_N)}\right) \geq -\frac{1}{2}, \tag{138}$$

$$\wedge\, 2(N-1)E\sqrt{\mu(S_N)}(1 + \kappa_N g_N + NE') \leq \frac{1}{2}. \tag{139}$$

We do not spell out the details here, as it is immediately apparent that all constraints on δ_N can be fulfilled by shrinking δ_N and thus $\mu(S_N)$, which is always possible as we never required a positive lower bound on δ_N. Observing that now

$$1 \geq z_N \geq \frac{1}{2} \implies 1 \leq \frac{1}{z_N} \leq 2, \tag{140}$$

and $R^* \leq NE' + 1$, while

$$(C_N)_{NN} \geq g_N - \frac{1}{2} \geq N + E' + R^* + \frac{1}{2} \geq N + E' + R^* + R_N, \tag{141}$$

we have $\mathbb{D}_N \cap \mathbb{D}^* = \emptyset$ for $N \geq 1$ and further by the Gershgorin Circle Theorem,

$$Re(\lambda_N) \geq \min_{x \in \mathbb{D}_N} Re(x) \geq N. \tag{142}$$

The claim now immediately follows from

$$\|\exp(\mathcal{L}_N t)\|_{\mathcal{B}(\mathcal{F}_N)} \geq e^{\lambda_N t}. \tag{143}$$

□

Remark 4. The example used a single bump function that moved as the dictionary size increased. This was merely used to simplify the presentation; however, the underlying effect of a dictionary function concentrating on a density pathology naturally occurs in the same way when refining a bump-like basis discretisation, such as Radial Basis Functions or Nodal Bases.

As Theorem 3 shows, the root cause for losing exponential boundedness is a pathology of the density ρ. We conjecture that a large class of well-behaved densities could be shown to preserve exponential boundedness, but leave a thorough characterisation of such "safe" densities for future work. In what follows, we will thus restrict ourselves to a special form of exponential boundedness, namely quasi-contractivity, which corresponds to a quasi-dissipativite generator. The following Lemma shows that g-EDMD in fact preserves this property. As per Lemma 1, a sufficient condition for quasi-contractivity of the semigroup is restricting μ to the Lebesgue measure, i.e. $\alpha = \beta = \frac{1}{\lambda(\mathcal{X})}$.

Lemma 4 (g-EDMD preserves quasi-dissipativity)**.** *Let assumptions 2 and 3 hold and assume the true Koopman operators to form a quasi-contractive semigroup, i.e.* $\|\mathcal{K}^t\|_{\mathcal{B}(\mathcal{F})} \leq e^{\omega t}$ *for some* $\omega \in \mathbb{R}$ *and all* $t \in [0, T]$. *Then* $\forall \epsilon > 0$*, there are constants* $N, M_N \in \mathbb{N}$ *such that for any* $M'_N \geq M_N$ *the g-EDMD approximation* $\hat{\mathcal{L}}_{NM'_N}$ *is almost surely quasi-dissipativite with* $\omega' = \omega + \epsilon$. *In particular, there exists a sequence* $\{(N, M_N)\}_{N=1}^{\infty}$ *such that the sequence of approximations is almost surely quasi-dissipativite with a uniform quasi-dissipativity-constant* ω'.

Proof. By the Lumer-Phillips Theorem (Engel & Nagel, 2000, Theorem 3.15), $\mathcal{L}$ is quasi-dissipativite, i.e.

$$\operatorname{Re} \langle \mathcal{L} f, f \rangle_{\mathcal{F}} \leq \omega \langle f, f \rangle_{\mathcal{F}}, \quad \forall f \in \mathcal{D}. \tag{144}$$

Because $\mathcal{F}_N \subset \mathcal{D}$ and projectors onto finite subspaces are self-adjoint

$$\langle \mathcal{L}_N f, f \rangle_{\mathcal{F}} = \langle \mathcal{P}_{\mathcal{F}_N} \mathcal{L} f, f \rangle_{\mathcal{F}} = \langle \mathcal{L} f, \mathcal{P}_{\mathcal{F}_N} f \rangle_{\mathcal{F}} = \langle \mathcal{L} f, f \rangle_{\mathcal{F}}, \quad \forall f \in \mathcal{F}_N, \tag{145}$$

which yields that $\mathcal{L}_N$ is also quasi-dissipativite with the same ω on $\mathcal{F}_N$. Further, by Llamazares-Elias et al. (2026)[Corollary 3.7], we can for every N find an M_N s.t. for any $M'_N \geq M_N$, $\left\| \hat{\mathcal{L}}_{NM'_N} - \mathcal{L}_N \right\| < \epsilon$ almost surely. Finally, we have

$$\begin{aligned}
\left\langle \hat{\mathcal{L}}_{NM'_N} f, f \right\rangle_{\mathcal{F}} &= \left\langle \left(\hat{\mathcal{L}}_{NM'_N} + \mathcal{L}_N - \mathcal{L}_N \right) f, f \right\rangle_{\mathcal{F}} && (146)\\
&= \langle \mathcal{L}_N f, f \rangle_{\mathcal{F}} + \left\langle \left(\hat{\mathcal{L}}_{NM'_N} - \mathcal{L}_N \right) f, f \right\rangle_{\mathcal{F}} && (147)\\
&\leq \omega \|f\|_{\mathcal{F}}^2 + \left\| \left(\hat{\mathcal{L}}_{NM'_N} - \mathcal{L}_N \right) f \right\|_{\mathcal{F}} \|f\|_{\mathcal{F}} && (148)\\
&\leq (\omega + \epsilon) \|f\|_{\mathcal{F}}^2 = \omega' \langle f, f \rangle_{\mathcal{F}}, && (149)
\end{aligned}$$

as claimed. □

We are now ready to show our final convergence result about the generated semigroups and, by extension, the predicted trajectories.

Theorem 2 (Convergence of the approximate trajectories)**.** *Let the assumptions be as in Lemma 3 and assume the true Koopman operators to form a quasi-contractive semigroup, i.e.* $\|\mathcal{K}^t\|_{\mathcal{B}(\mathcal{F})} \leq e^{\omega t}$ *for some* $\omega_* \in \mathbb{R}$ *and all* $t \in [0, T]$. *Then* $\forall \epsilon > 0$ *and* $t \in [0, T]$*, there are constants* $N, M_N \in \mathbb{N}$*, a SWIM sampled hidden layer* Φ_N *and outer weights* $\boldsymbol{c}^i$ *such that for any* $M'_N \geq M_N$ *almost surely*

$$\left\| \exp(\mathcal{L} t)\, e_i - \sum_{j=1}^{N} c_j^i \exp\left(\hat{\mathcal{L}}_{NM'_N} t \right) \phi_j \right\|_{\mathcal{F}} < \epsilon, \quad \forall i \in \{1, 2, \ldots, d\}. \tag{5}$$

Proof. The main idea is to apply the First Trotter-Kato Approximation Theorem (Engel & Nagel, 2000, Theorem 4.8), which guarantees the convergence of the Koopman Operator Semigroup given Exponential Boundedness and the convergence of the generators. Since we already showed the convergence of the generators in our setting, the main work of the proof is showing boundedness, which will follow from quasi-dissipativity via the Lumer-Phillips Theorem (Engel & Nagel, 2000, Theorem 3.15).
Let $g \in \mathcal{D}$ be an observable.

1. We start by separating the convergence of the projected observable and the convergence of the projected generator

$$\left\| e^{\mathcal{L}t} g - \sum_{j=1}^{N} c_j^i e^{\hat{\mathcal{L}}_{NM'_N} t} \phi_j \right\|_{\mathcal{F}} \tag{150}$$

$$= \left\| e^{\mathcal{L}t} g - e^{\hat{\mathcal{L}}_{NM'_N} t} \mathcal{P}_{\mathcal{D}_N} g \right\|_{\mathcal{F}} \tag{151}$$

$$\leq \left\| e^{\mathcal{L}t} g - e^{\mathcal{L}t} \mathcal{P}_{\mathcal{D}_N} g \right\|_{\mathcal{F}} + \left\| \left(e^{\mathcal{L}t} - e^{\hat{\mathcal{L}}_{NM'_N} t} \right) \mathcal{P}_{\mathcal{D}_N} g \right\|_{\mathcal{F}} \tag{152}$$

$$\leq \left\| \mathcal{K}^t \right\|_{\mathcal{B}(\mathcal{F})} \left\| g - \mathcal{P}_{\mathcal{D}_N} g \right\|_{\mathcal{F}} + \left\| \left(e^{\mathcal{L}t} - e^{\hat{\mathcal{L}}_{NM'_N} t} \right) \mathcal{P}_{\mathcal{D}_N} g \right\|_{\mathcal{F}} \tag{153}$$

Since the true operator is bounded by assumption, the first term converges to zero by the universality of the dictionary (Bolager et al., 2023, Theorem 1). The second term is exactly the pointwise convergence of the operator, which we show in the following steps.

2. Next, we will show that $\hat{\mathcal{L}}_{NM'_N}$ generates a quasi-contraction (and thus exponentially bounded) semigroup. By Lemma 4, we can choose M_N so that $\hat{\mathcal{L}}_{NM'_N}$ is almost surely quasi-dissipativite with $\omega' = \omega + \epsilon$. By being finite-dimensional, it is densely defined on $\mathcal{F}_N$ and bounded. Furthermore, with $\lambda > \omega'$

$$(\hat{\mathcal{L}}_{NM'_N} - \lambda \mathrm{Id}) f = 0 \tag{154}$$

$$\iff (\hat{\mathcal{L}}_{NM'_N} - \omega' \mathrm{Id}) f = (\lambda - \omega') f \tag{155}$$

$$\implies \underbrace{\langle (\hat{\mathcal{L}}_{NM'_N} - \omega' \mathrm{Id}) f, f \rangle_{\mathcal{F}}}_{\leq 0} = \underbrace{(\lambda - \omega')}_{>0} \underbrace{\langle f, f \rangle_{\mathcal{F}}}_{\geq 0} \tag{156}$$

$$\implies f = 0. \tag{157}$$

Thus $\hat{\mathcal{L}}_{NM'_N}$ is injective and, by being finite dimensional, also surjective. With this, we can invoke the Lumer-Phillips Theorem (Engel & Nagel, 2000, Theorem 3.15) for quasicontraction semigroups, yielding that $\left\| e^{\hat{\mathcal{L}}_{NM'_N} t} \right\|_{\mathcal{B}(\mathcal{F})} \leq e^{\omega' t}$ for all N.

3. Finally, since we have exponential stability for the semigroup generated by $\hat{\mathcal{L}}_{NM'_N}$, and pointwise convergence of $\hat{\mathcal{L}}_{NM'_N}$ on $\mathcal{D}$ by Lemma 3, we are in the setting of the First Trotter-Kato Approximation Theorem (Engel & Nagel, 2000, Theorem 4.8). Specifically, we apply the implication (b) $\implies$ (d) with $D = \mathcal{D}$ and $x_n = \mathcal{P}_{\mathcal{D}_n} f$ in Engel & Nagel (2000), yielding the sought-after convergence of the Koopman Operators, completing the proof. The number of data points M_N is simply chosen as the larger of the requirements imposed by Lemma 3 and Lemma 4.

□

Remark 5. It should be emphasised that the proof made no reference to specific properties of sampled networks, making this a general result about g-EDMD rather than our specific dictionary.

E TRAINING NODES WITH ADJOINT SENSITIVITY METHOD, EXPLODING AND VANISHING GRADIENTS PROBLEM

Consider a Neural ODE defined as

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}(\boldsymbol{z}(t), t, \boldsymbol{\theta}), \tag{158}$$

where $\boldsymbol{z}(t) \in \mathbb{R}^d$ is the state of the network at time t and $\boldsymbol{\theta}$ denote the set of system's parameters.

The main difficulty in training continuous depth/time networks is reverse-mode differentiation through the ODE solver. Treating the ODE solver as a black box, gradients are computed via the *adjoint sensitivity method*. This approach computes gradients by solving a second (augmented) ODE backwards in time. It works with any ODE solver, scales linearly, has low memory cost, and explicitly controls numerical error (Chen et al., 2018).

Consider optimizing a scalar loss $\mathcal{J}(\cdot)$ applied to the ODE solver output

$$\mathcal{J}(\boldsymbol{z}(t_N)) = \mathcal{J}\left(\boldsymbol{z}(t_0) + \int_{t_0}^{t_N} f(\boldsymbol{z}(t), t, \theta)\, dt\right) = \mathcal{J}(\texttt{ODESolve}(\boldsymbol{z}(t_0), f, t_0, t_N, \theta)). \tag{159}$$

For $\theta \in \boldsymbol{\theta}$, we define an augmented adjoint state $\boldsymbol{a}_{aug}(t) = [\boldsymbol{a}(t), \boldsymbol{a}_\theta(t)]$ with the adjoint state

$$\boldsymbol{a}(t) = \frac{d\mathcal{J}}{d\boldsymbol{z}(t)}, \tag{160}$$

and the parameter sensitivity or parameter adjoint

$$\boldsymbol{a}_\theta(t) := \frac{d\mathcal{J}}{d\theta}, \tag{161}$$

which respectively represent the sensitivity of the loss with respect to the state trajectory and the model parameters. Their dynamics are governed by the following ODEs

$$\begin{aligned} \frac{d\boldsymbol{a}(t)}{dt} &= -\boldsymbol{a}(t)^\mathsf{T} \frac{d\boldsymbol{f}(\boldsymbol{z}(t), t, \theta)}{d\boldsymbol{z}(t)}, \\ \frac{d\boldsymbol{a}_\theta(t)}{dt} &= -\boldsymbol{a}(t)^\mathsf{T} \frac{d\boldsymbol{f}(\boldsymbol{z}(t), t, \theta)}{d\theta}. \end{aligned} \tag{162}$$

These equations are integrated backward in time from the terminal time t_N to t_0. Specifically

$$\begin{aligned} \boldsymbol{a}(t_0) &= \boldsymbol{a}(t_N) + \int_{t_N}^{t_0} \frac{d\boldsymbol{a}(t)}{dt} dt = \boldsymbol{a}(t_N) - \int_{t_N}^{t_0} \boldsymbol{a}(t)^\mathsf{T} \frac{d\boldsymbol{f}(\boldsymbol{z}(t), t, \theta)}{d\boldsymbol{z}(t)} dt, \\ \boldsymbol{a}_\theta(t_0) &= \boldsymbol{a}_\theta(t_N) + \int_{t_N}^{t_0} \frac{d\boldsymbol{a}(t)}{dt} dt = -\int_{t_N}^{t_0} \boldsymbol{a}(t)^\mathsf{T} \frac{d\boldsymbol{f}(\boldsymbol{z}(t), t, \theta)}{d\theta} dt. \qquad (\boldsymbol{a}_\theta(t_N) = 0) \end{aligned} \tag{163}$$

For simplicity, we present the results for autonomous Neural ODEs. Similar results can be extended to non-autonomous systems by considering the augmented adjoint state as $\boldsymbol{a}_{aug}(t) = [\boldsymbol{a}(t), \boldsymbol{a}_\theta(t), \boldsymbol{a}_t(t)]$, replacing the flow map with the corresponding evolution operator (fundamental matrix) and invoking Oseledec's Multiplicative Ergodic Theorem for non-autonomous dynamical systems.

Definition 2. *Consider an ODE system*

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}(\boldsymbol{z}(t)), \quad \boldsymbol{z}(t_0) = \boldsymbol{z}_0, \quad \boldsymbol{f} \in C^1(E), \quad E \subseteq \mathbb{R}^d, \tag{164}$$

with a trajectory $\boldsymbol{z}(t)$ as its solution. The time evolution of a tangent vector $\delta\boldsymbol{z}(t)$ at $\boldsymbol{z}(t)$ is given by the linearized variational (tangent) equation

$$\delta\dot{\boldsymbol{z}}(t) = \boldsymbol{A}(t)\, \delta\boldsymbol{z}(t), \tag{165}$$

where

$$\boldsymbol{A}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}(\boldsymbol{z}(t)), \tag{166}$$

is the Jacobian matrix of $\boldsymbol{f}$ along the trajectory $\boldsymbol{z}(t)$. A principal fundamental matrix $\boldsymbol{\Phi}(t, t_0) \in \mathbb{R}^{d\times d}$ is a matrix-valued solution of

$$\frac{d}{dt}\boldsymbol{\Phi}(t, t_0) = \boldsymbol{A}(t)\ \boldsymbol{\Phi}(t, t_0), \quad \boldsymbol{\Phi}(t_0, t_0) = \boldsymbol{I}_d. \tag{167}$$

Then the general solution satisfies

$$\delta\boldsymbol{z}(t) = \boldsymbol{\Phi}(t, t_0)\,\delta\boldsymbol{z}(t_0), \tag{168}$$

and $\boldsymbol{\Phi}$ has the semigroup property

$$\boldsymbol{\Phi}(t, t_1)\,\boldsymbol{\Phi}(t_1, t_0) = \boldsymbol{\Phi}(t, t_0) \quad \textit{for all } t, t_1, t_0. \tag{169}$$

Also $\boldsymbol{\Phi}(t, t_0)$ is invertible and

$$\boldsymbol{\Phi}(t, t_0)^{-1} = \boldsymbol{\Phi}(t_0, t). \tag{170}$$

Definition 3. *Let $\boldsymbol{\Phi}(t, s)$ denote the fundamental matrix of the tangent linear system* (165)*, propagating tangent vectors along the trajectory from time s to time t, where $s < t$. Let $\tau := t - s > 0$ denote the propagation time. The largest Finite-Time Lyapunov Exponent (FTLE) over an interval of length τ is defined by*

$$\lambda_{\max}^{FT}(\tau) := \frac{1}{\tau}\log\left\|\boldsymbol{\Phi}(s+\tau, s)\right\|, \tag{171}$$

where $\|\cdot\|$ denotes the spectral norm or operator 2-norm. The largest Lyapunov Exponent (LE) along the trajectory is

$$\lambda_{\max} := \limsup_{\tau\to\infty}\frac{1}{\tau}\log\left\|\boldsymbol{\Phi}(s+\tau, s)\right\|. \tag{172}$$

If the limit exists, then

$$\lambda_{\max} = \lim_{\tau\to\infty}\frac{1}{\tau}\log\left\|\boldsymbol{\Phi}(s+\tau, s)\right\|. \tag{173}$$

Here $\|\cdot\|$ is any matrix norm; the finite-time value depends on the norm, while the asymptotic Lyapunov exponent is norm-independent in finite dimensions. Note that $\lambda_{\max}$ is the supremum over all nonzero initial tangent vectors $\delta\boldsymbol{z}_0$ of the vector-specific exponent

$$\lambda(\delta\boldsymbol{z}_0) = \limsup_{\tau\to\infty}\frac{1}{\tau}\log\frac{\|\boldsymbol{\Phi}(s+\tau, s)\delta\boldsymbol{z}_0\|}{\|\delta\boldsymbol{z}_0\|}, \tag{174}$$

i.e. the largest possible asymptotic growth rate of any tangent perturbation, and it coincides with that vector-specific exponent when $\delta\boldsymbol{z}_0$ lies in the maximal asymptotic growth direction.

We now establish theoretical results that characterize the Exploding and Vanishing Gradient Problem (EVGP) for Neural ODEs trained via the adjoint sensitivity method. The first theorem gives FTLE bounds for the backward-in-time growth of the state adjoint $\boldsymbol{a}(t)$ over an interval $[t, T]$. The second theorem gives the corresponding LE bounds for the state adjoint in the asymptotic regime $T \to \infty$, under Oseledec's multiplicative ergodic theorem. The third theorem extends both results to the parameter gradient $\nabla_{\boldsymbol{\theta}}\mathcal{J}$, yielding unified FTLE and LE bounds together with the corresponding exploding, vanishing, and bounded regimes.

Theorem 5. *Consider the autonomous Neural ODE*

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}(\boldsymbol{z}(t), \boldsymbol{\theta}), \tag{175}$$

trained on a time series data $\{\boldsymbol{z}(t) : 0 \le t \le T\}$, using the loss function $\mathcal{J}$ defined as (159)*. Let $\boldsymbol{f} \in C^1$, $\boldsymbol{z}(t)$ be a solution on $[0, T]$. Fix $t \in [0, T)$ and let $\lambda_{\max}^{FT}(t, T)$ denote the largest FTLE over the interval $[t, T]$, i.e.*

$$\lambda_{\max}^{FT}(t, T) := \frac{1}{T-t}\log\left\|\boldsymbol{\Phi}(T, t)\right\|. \tag{176}$$

Then the adjoint state $\boldsymbol{a}(t) = \dfrac{d\mathcal{J}}{d\boldsymbol{z}(t)}$ satisfies

$$\|\boldsymbol{a}(t)\| \le \|\boldsymbol{a}_T\|\,\exp\Big(\lambda_{\max}^{FT}(t, T)\,(T-t)\Big), \qquad \boldsymbol{a}(T) = \boldsymbol{a}_T, \tag{177}$$

for all $t \in [0,T]$. Moreover, assume that $\boldsymbol{u}_{\max}(t,T)$ is a unit left singular vector of $\boldsymbol{\Phi}(T,t)$ corresponding to the largest singular value $\sigma_{\max}(\boldsymbol{\Phi}(T,t)) = \|\boldsymbol{\Phi}(T,t)\|$. If the terminal adjoint vector is not orthogonal to this maximally expanding direction, i.e.

$$\beta(t,T) := \boldsymbol{a}_T\, \boldsymbol{u}_{\max}(t,T) \neq 0, \tag{178}$$

then the following finite-time lower bound holds:

$$\|\boldsymbol{a}(t)\| \geq |\beta(t,T)|\, \exp\Big(\lambda^{FT}_{\max}(t,T)\,(T-t)\Big). \tag{179}$$

In particular, if there exists a constant $c_0 > 0$ such that $|\boldsymbol{a}_T\, \boldsymbol{u}_{\max}(t,T)| \geq c_0 \|\boldsymbol{a}_T\|$, then

$$\|\boldsymbol{a}(t)\| \geq c_0 \|\boldsymbol{a}_T\|\, \exp\Big(\lambda^{FT}_{\max}(t,T)\,(T-t)\Big). \tag{180}$$

Consequently, for the finite interval $[t,T]$:

1. *If $\lambda^{FT}_{\max}(t,T) > 0$ and $\beta(t,T) \neq 0$, then the adjoint norm grows exponentially backward in time, as quantified by the upper and lower bounds above.*
2. *If $\lambda^{FT}_{\max}(t,T) < 0$, then the upper bound shows that the adjoint norm decays exponentially backward in time for all $\boldsymbol{a}_T$.*
3. *If $\lambda^{FT}_{\max}(t,T) = 0$, then the upper bound gives $\|\boldsymbol{a}(t)\| \leq \|\boldsymbol{a}_T\|$.*

Proof. Let $\boldsymbol{\Phi}(t,s)$ be the fundamental matrix for the tangent variational equation

$$\frac{d}{dt}\boldsymbol{\Phi}(t,s) = \boldsymbol{A}(t)\boldsymbol{\Phi}(t,s), \qquad \boldsymbol{\Phi}(s,s) = \boldsymbol{I}_d, \tag{181}$$

where $\boldsymbol{A}(t) = \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}(\boldsymbol{z}(t))$. For ease of notation, we consider the row-vector adjoint, whose evolution is given by

$$\dot{\boldsymbol{a}}(t) = -\boldsymbol{a}(t)\, \boldsymbol{A}(t), \qquad \boldsymbol{a}(T) = \boldsymbol{a}_T. \tag{182}$$

Consider the function

$$\boldsymbol{a}_T\, \boldsymbol{\Phi}(T,t), \tag{183}$$

for $t \in [0,T]$, with T fixed. By the semigroup property we have

$$\boldsymbol{\Phi}(T,t) = \boldsymbol{\Phi}(T,s)\, \boldsymbol{\Phi}(t,s)^{-1} \tag{184}$$

Differentiating with respect to t yields

$$\frac{\partial}{\partial t}\boldsymbol{\Phi}(T,t) = \boldsymbol{\Phi}(T,s)\, \frac{\partial}{\partial t}\left[\boldsymbol{\Phi}(t,s)^{-1}\right] \tag{185}$$

Now differentiate the inverse using $\partial_t(M^{-1}) = -M^{-1}(\partial_t M)M^{-1}$ gives

$$\begin{aligned}\frac{\partial}{\partial t}\left[\boldsymbol{\Phi}(t,s)^{-1}\right] &= -\boldsymbol{\Phi}(t,s)^{-1}\left(\frac{\partial}{\partial t}\boldsymbol{\Phi}(t,s)\right)\boldsymbol{\Phi}(t,s)^{-1} \\ &= -\boldsymbol{\Phi}(s,t)\,(\boldsymbol{A}(t)\,\boldsymbol{\Phi}(t,s))\,\boldsymbol{\Phi}(t,s)^{-1} = -\boldsymbol{\Phi}(s,t)\,\boldsymbol{A}(t).\end{aligned} \tag{186}$$

Hence, from (185) and (186)

$$\frac{\partial}{\partial t}\boldsymbol{\Phi}(T,t) = \boldsymbol{\Phi}(T,s)\,(-\boldsymbol{\Phi}(s,t)\,\boldsymbol{A}(t)) = -\boldsymbol{\Phi}(T,t)\,\boldsymbol{A}(t). \tag{187}$$

Therefore

$$\frac{d}{dt}\big(\boldsymbol{a}_T\, \boldsymbol{\Phi}(T,t)\big) \;=\; \boldsymbol{a}_T\, \frac{\partial}{\partial t}\boldsymbol{\Phi}(T,t) \;=\; \boldsymbol{a}_T\, \big(-\boldsymbol{\Phi}(T,t)\,\boldsymbol{A}(t)\big) \;=\; -\big(\boldsymbol{a}_T\, \boldsymbol{\Phi}(T,t)\big)\,\boldsymbol{A}(t). \tag{188}$$

Moreover, evaluating at $t = T$

$$\boldsymbol{a}_T\, \boldsymbol{\Phi}(T,T) = \boldsymbol{a}_T\, \boldsymbol{I}_d = \boldsymbol{a}_T. \tag{189}$$

Thus, $\boldsymbol{a}_T\,\boldsymbol{\Phi}(T,t)$ satisfies the adjoint differential equation (182). By the uniqueness of solutions to linear ODEs, we conclude that

$$\boldsymbol{a}(t) = \boldsymbol{a}_T\,\boldsymbol{\Phi}(T,t) \quad \text{for all } t \in [0,T]. \tag{190}$$

Hence the adjoint state is obtained by propagating the terminal covector backward along the tangent linear flow. Furthermore

$$\|\boldsymbol{a}(t)\| = \|\boldsymbol{a}_T\,\boldsymbol{\Phi}(T,t)\| \le \|\boldsymbol{a}_T\|\,\|\boldsymbol{\Phi}(T,t)\|. \tag{191}$$

By definition of $\lambda_{\max}^{FT}(t,T)$

$$\|\boldsymbol{\Phi}(T,t)\| = \exp\Big(\lambda_{\max}^{FT}(t,T)\,(T-t)\Big), \tag{192}$$

which gives the claimed FTLE-based upper bound:

$$\|\boldsymbol{a}(t)\| \le \|\boldsymbol{a}_T\|\,\exp\Big(\lambda_{\max}^{FT}(t,T)\,(T-t)\Big). \tag{193}$$

For the lower bound, let

$$\boldsymbol{\Phi}(T,t) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$$

be a singular value decomposition, and let $\boldsymbol{u}_{\max}(t,T)$ be the first column of $\boldsymbol{U}$, corresponding to the largest singular value

$$\sigma_{\max}(\boldsymbol{\Phi}(T,t)) = \|\boldsymbol{\Phi}(T,t)\|.$$

Write the row vector $\boldsymbol{a}_T$ as

$$\boldsymbol{a}_T = \beta(t,T)\,\boldsymbol{u}_{\max}(t,T)^\top + \boldsymbol{w}^\top,$$

where $\beta(t,T) = \boldsymbol{a}_T \cdot \boldsymbol{u}_{\max}(t,T)$ and $\boldsymbol{w} \perp \boldsymbol{u}_{\max}(t,T)$. Then

$$\boldsymbol{a}_T\boldsymbol{\Phi}(T,t) = \beta(t,T)\,\boldsymbol{u}_{\max}(t,T)^\top\boldsymbol{\Phi}(T,t) + \boldsymbol{w}^\top\boldsymbol{\Phi}(T,t). \tag{194}$$

Since

$$\boldsymbol{u}_{\max}(t,T)^\top\boldsymbol{\Phi}(T,t) = \sigma_{\max}(\boldsymbol{\Phi}(T,t))\,\boldsymbol{v}_{\max}(t,T)^\top,$$

for the corresponding right singular vector $\boldsymbol{v}_{\max}(t,T)$, and since $\boldsymbol{w} \perp \boldsymbol{u}_{\max}(t,T)$ implies

$$\boldsymbol{w}^\top\boldsymbol{\Phi}(T,t) \perp \boldsymbol{u}_{\max}(t,T)^\top\boldsymbol{\Phi}(T,t),$$

we obtain, by Pythagorean theorem

$$\begin{aligned}\|\boldsymbol{a}_T\boldsymbol{\Phi}(T,t)\|^2 &= |\beta(t,T)|^2\|\boldsymbol{u}_{\max}(t,T)^\top\boldsymbol{\Phi}(T,t)\|^2 + \|\boldsymbol{w}^\top\boldsymbol{\Phi}(T,t)\|^2 \\ &\ge |\beta(t,T)|^2\|\boldsymbol{u}_{\max}(t,T)^\top\boldsymbol{\Phi}(T,t)\|^2.\end{aligned} \tag{195}$$

Therefore

$$\begin{aligned}\|\boldsymbol{a}_T\boldsymbol{\Phi}(T,t)\| &\ge |\beta(t,T)|\,\|\boldsymbol{u}_{\max}(t,T)^\top\boldsymbol{\Phi}(T,t)\| \\ &= |\beta(t,T)|\,\sigma_{\max}(\boldsymbol{\Phi}(T,t)) = |\beta(t,T)|\,\exp\Big(\lambda_{\max}^{FT}(t,T)\,(T-t)\Big).\end{aligned}$$

This proves the lower bound.

If additionally

$$|\boldsymbol{a}_T\,\boldsymbol{u}_{\max}(t,T)| \ge c_0\|\boldsymbol{a}_T\|, \tag{196}$$

then substituting $\beta(t,T) = \boldsymbol{a}_T\boldsymbol{u}_{\max}(t,T)$ gives

$$\|\boldsymbol{a}(t)\| \ge c_0\|\boldsymbol{a}_T\|\exp\Big(\lambda_{\max}^{FT}(t,T)\,(T-t)\Big). \tag{197}$$

The three consequences stated in the theorem follow immediately from the upper and lower bounds by substituting the sign of $\lambda_{\max}^{FT}(t,T)$. □

Theorem 6. *Consider the autonomous Neural ODE*

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}(\boldsymbol{z}(t), \boldsymbol{\theta}), \tag{198}$$

trained on a time series data $\{\boldsymbol{z}(t) : 0 \le t \le T\}$*, using the loss function* $\mathcal{J}$ *defined as* (159)*. Let* $\boldsymbol{f} \in C^1$ *and let* $\boldsymbol{z}(t)$ *be a solution that exists for arbitrarily long times. Fix an initial time* $t \in [0, T]$ *and let* $\lambda_{\max}(t)$ *denote the largest LE along the trajectory starting at* $\boldsymbol{z}(t)$*. Then, as* $T \to \infty$ *with* t *fixed, the adjoint state* $\boldsymbol{a}(t) = \dfrac{d\mathcal{J}}{d\boldsymbol{z}(t)}$ *satisfies*

$$\|\boldsymbol{a}(t)\| \le \|\boldsymbol{a}_T\|\, M(\epsilon)\, e^{(\lambda_{\max}(t)+\epsilon)(T-t)} \qquad \forall \epsilon > 0.$$

Moreover, under the assumptions of Oseledec's multiplicative ergodic theorem, for terminal adjoint vectors $\boldsymbol{a}_T$ *that have a nonzero component in the top dual Oseledec subspace associated with* $\lambda_{\max}(t)$*, there exists a matching lower bound*

$$\|\boldsymbol{a}(t)\| \ge m(\epsilon, \boldsymbol{a}_T)\, e^{(\lambda_{\max}(t)-\epsilon)(T-t)} \qquad \forall \epsilon > 0,$$

for sufficiently large $T - t$*. Consequently:*

1. *If* $\lambda_{\max}(t) > 0$*, the upper bound allows at most exponential growth. For such adjoint vectors* $\boldsymbol{a}_T$*, the lower bound ensures that* $\|\boldsymbol{a}(t)\|$ *actually grows exponentially backward in time. For exceptional* $\boldsymbol{a}_T$*, e.g.,* $\boldsymbol{a}_T = 0$ *or lying entirely in a lower-growth Oseledec subspace, the growth may be slower.*
2. *If* $\lambda_{\max}(t) < 0$*, the upper bound decays exponentially backward. Hence, for all* $\boldsymbol{a}_T$*,* $\|\boldsymbol{a}(t)\|$ *decays exponentially backward.*
3. *If* $\lambda_{\max}(t) = 0$*, the upper bound gives at most subexponential growth for all* $\boldsymbol{a}_T$*.*

Proof. From the proof of Theorem 5, the adjoint state satisfies

$$\boldsymbol{a}(t) = \boldsymbol{a}_T\, \boldsymbol{\Phi}(T, t) \quad \text{for all } t \in [0, T]. \tag{199}$$

Hence

$$\|\boldsymbol{a}(t)\| = \|\boldsymbol{a}_T\, \boldsymbol{\Phi}(T, t)\| \le \|\boldsymbol{a}_T\|\, \|\boldsymbol{\Phi}(T, t)\|. \tag{200}$$

Now fix $t \in [0, T]$ and let $\tau = T - t$. By Definition 3, the largest Lyapunov exponent along the trajectory starting at $\boldsymbol{z}(t)$ is

$$\lambda_{\max}(t) = \limsup_{T\to\infty} \frac{1}{T-t} \log \|\boldsymbol{\Phi}(T, t)\|. \tag{201}$$

Thus, for any $\epsilon > 0$, there exists a constant $M(\epsilon) > 0$ such that for all sufficiently large $T - t$

$$\|\boldsymbol{\Phi}(T, t)\| \le M(\epsilon)\, e^{(\lambda_{\max}(t)+\epsilon)(T-t)}. \tag{202}$$

Substituting this into the adjoint norm bound gives

$$\|\boldsymbol{a}(t)\| \le \|\boldsymbol{a}_T\|\, M(\epsilon)\, e^{(\lambda_{\max}(t)+\epsilon)(T-t)}. \tag{203}$$

Thus, for every $\epsilon > 0$,

$$\|\boldsymbol{a}(t)\| = \mathcal{O}\left(e^{(\lambda_{\max}(t)+\epsilon)(T-t)}\right) \tag{204}$$

as $T - t \to \infty$ with t fixed. This proves the claimed exponential-order upper bound.

To obtain the lower bound, we invoke Oseledec's multiplicative ergodic theorem. Let μ be an invariant measure for the flow of the state ODE (198). For μ-almost every initial condition, the tangent space admits an Oseledec decomposition. The adjoint state $\boldsymbol{a}(t)$ evolves backward in time according to

$$\boldsymbol{a}(t) = \boldsymbol{a}_T\, \boldsymbol{\Phi}(T, t). \tag{205}$$

Hence its asymptotic growth rate is determined by the dual Oseledec decomposition. The top dual Oseledec subspace is the set of adjoint vectors whose backward growth rate equals $\lambda_{\max}(t)$. If the terminal adjoint $\boldsymbol{a}_T$ has a nonzero component in this subspace, then, by Oseledec's theorem, the following limit exists:

$$\lim_{\tau\to\infty} \frac{1}{\tau} \log \|\boldsymbol{a}_T\, \boldsymbol{\Phi}(T, T-\tau)\| = \lambda_{\max}(t). \tag{206}$$

This means that the backward growth rate of $\|\boldsymbol{a}(t)\|$ equals $\lambda_{\max}(t)$ for such $\boldsymbol{a}_T$. Therefore, for any $\epsilon > 0$, there exists a constant $m(\epsilon, \boldsymbol{a}_T) > 0$ such that for all sufficiently large $\tau = T - t$

$$\|\boldsymbol{a}(t)\| \geq m(\epsilon, \boldsymbol{a}_T)\, e^{(\lambda_{\max}(t)-\epsilon)(T-t)}. \tag{207}$$

Finally, the three consequences follow immediately from the upper and lower bounds by substituting the sign of $\lambda_{\max}(t)$: positive yields exponential growth backward in time for terminal adjoint vectors $\boldsymbol{a}_T$ having a nonzero component in the top dual Oseledec subspace associated with $\lambda_{\max}(t)$ (and possibly slower growth for exceptional $\boldsymbol{a}_T$), negative yields exponential decay backward in time for all $\boldsymbol{a}_T$, and zero yields at most subexponential growth backward in time for all $\boldsymbol{a}_T$. □

Theorem 7. *Consider the autonomous Neural ODE*

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}(\boldsymbol{z}(t), \boldsymbol{\theta}), \tag{208}$$

trained on a time series data $\{\boldsymbol{z}(t) : 0 \leq t \leq T\}$, using the loss function $\mathcal{J}$ defined as (159)*. Let $\boldsymbol{f} \in C^1$, let $\boldsymbol{z}(t)$ be a solution that exists for arbitrarily long times, and assume*

$$\left\| \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} \right\| \leq C_\theta, \tag{209}$$

for some constant $C_\theta > 0$. Fix $t \in [0, T)$ and let $\lambda^{FT}_{\max}(t, T)$ denote the largest FTLE over the interval $[t, T]$

$$\lambda^{FT}_{\max}(t, T) := \frac{1}{T-t} \log \|\boldsymbol{\Phi}(T, t)\|\,. \tag{210}$$

Let $\lambda_{\max}(t)$ denote the largest LE along the trajectory starting at $\boldsymbol{z}(t)$.

(i) Finite-time (FTLE) bounds. *For every $T > 0$, the parameter gradient $\nabla_{\boldsymbol{\theta}}\mathcal{J} = \dfrac{d\mathcal{J}}{d\boldsymbol{\theta}}$ satisfies*

$$\|\nabla_{\boldsymbol{\theta}}\mathcal{J}\| \leq C_\theta \|\boldsymbol{a}_T\| \int_0^T \exp\Big(\lambda^{FT}_{\max}(t, T)\,(T-t)\Big)\, dt. \tag{211}$$

Moreover, if the terminal adjoint vector $\boldsymbol{a}_T$ is not orthogonal to the maximally expanding direction over $[t, T]$, i.e.

$$\beta(t, T) := \boldsymbol{a}_T\, \boldsymbol{u}_{\max}(t, T) \neq 0, \tag{212}$$

where $\boldsymbol{u}_{\max}(t, T)$ is a unit left singular vector of $\boldsymbol{\Phi}(T, t)$ corresponding to the largest singular value, then

$$\|\nabla_{\boldsymbol{\theta}}\mathcal{J}\| \geq C_\theta \int_0^T |\beta(t, T)| \exp\Big(\lambda^{FT}_{\max}(t, T)\,(T-t)\Big)\, dt. \tag{213}$$

(ii) Asymptotic (LE) bounds. *As $T \to \infty$ with t fixed, for every $\epsilon > 0$*

$$\|\nabla_{\boldsymbol{\theta}}\mathcal{J}\| \leq C_\theta \|\boldsymbol{a}_T\|\, M(\epsilon) \int_0^T e^{(\lambda_{\max}(t)+\epsilon)(T-t)}\, dt. \tag{214}$$

Moreover, under the assumptions of Oseledec's multiplicative ergodic theorem, for terminal adjoint vectors $\boldsymbol{a}_T$ that have a nonzero component in the top dual Oseledec subspace associated with $\lambda_{\max}(t)$, there exists a constant $m(\epsilon, \boldsymbol{a}_T) > 0$ such that for sufficiently large T

$$\|\nabla_{\boldsymbol{\theta}}\mathcal{J}\| \geq C_\theta\, m(\epsilon, \boldsymbol{a}_T) \int_0^T e^{(\lambda_{\max}(t)-\epsilon)(T-t)}\, dt. \tag{215}$$

(iii) Finite-time regime consequences (FTLE). *Let $\lambda^{FT}_{\max} := \lambda^{FT}_{\max}(t, T)$.*

1. *If* $\lambda_{\max}^{FT} > 0$*, then*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq C_\theta \, \|\boldsymbol{a}_T\| \frac{e^{\lambda_{\max}^{FT} T} - 1}{\lambda_{\max}^{FT}} = \mathcal{O}\Big(e^{\lambda_{\max}^{FT} T}\Big).$$

If in addition $|\beta(t,T)| \geq \beta_{\min} > 0$ *for all* $t \in [0,T]$*, then*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \geq C_\theta \, \beta_{\min} \frac{e^{\lambda_{\max}^{FT} T} - 1}{\lambda_{\max}^{FT}}.$$

2. *If* $\lambda_{\max}^{FT} = 0$*, then*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq C_\theta \, \|\boldsymbol{a}_T\| \, T = \mathcal{O}(T).$$

3. *If* $\lambda_{\max}^{FT} < 0$*, then*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq C_\theta \, \|\boldsymbol{a}_T\| \frac{1 - e^{\lambda_{\max}^{FT} T}}{|\lambda_{\max}^{FT}|} \leq \frac{C_\theta \, \|\boldsymbol{a}_T\|}{|\lambda_{\max}^{FT}|},$$

so the parameter gradient is uniformly bounded independently of T*.*

(iv) Asymptotic regime consequences (LE).

1. *If* $\lambda_{\max}(t) > 0$*, then*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq C_\theta \, \|\boldsymbol{a}_T\| \, M(\epsilon) \frac{e^{(\lambda_{\max}(t)+\epsilon)T} - 1}{\lambda_{\max}(t) + \epsilon} = \mathcal{O}\Big(e^{(\lambda_{\max}(t)+\epsilon)T}\Big),$$

and for $\boldsymbol{a}_T$ *satisfying the top dual Oseledec condition*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \geq C_\theta \, m(\epsilon, \boldsymbol{a}_T) \frac{e^{(\lambda_{\max}(t)-\epsilon)T} - 1}{\lambda_{\max}(t) - \epsilon} \geq \frac{C_\theta \, m(\epsilon, \boldsymbol{a}_T)}{\lambda_{\max}(t) - \epsilon} \, e^{(\lambda_{\max}(t)-\epsilon)T}. \tag{216}$$

For exceptional $\boldsymbol{a}_T$ *(e.g.,* $\boldsymbol{a}_T = 0$ *or lying entirely in a lower-growth Oseledec subspace), the growth may be slower.*

2. *If* $\lambda_{\max}(t) = 0$*, then*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq C_\theta \, \|\boldsymbol{a}_T\| \, M(\epsilon) \frac{e^{\epsilon T} - 1}{\epsilon} = \mathcal{O}(T) \tag{217}$$

for fixed $\epsilon > 0$*, so the parameter gradient grows at most linearly.*

3. *If* $\lambda_{\max}(t) < 0$*, then choosing* $\epsilon < |\lambda_{\max}(t)|$ *gives*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq C_\theta \, \|\boldsymbol{a}_T\| \, M(\epsilon) \frac{1 - e^{-(|\lambda_{\max}(t)|-\epsilon)T}}{|\lambda_{\max}(t)| - \epsilon} \leq \frac{C_\theta \, \|\boldsymbol{a}_T\| \, M(\epsilon)}{|\lambda_{\max}(t)| - \epsilon},$$

so the parameter gradient remains uniformly bounded independently of T*.*

Proof. From the adjoint sensitivity equations, the parameter gradient admits the integral representation

$$\nabla_{\boldsymbol{\theta}} \mathcal{J} = -\int_0^T \boldsymbol{a}(t) \, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}(\boldsymbol{z}(t)) \, dt, \tag{218}$$

where $\boldsymbol{a}_\theta(T) = 0$. Taking norms and applying the triangle inequality

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{J}\| \leq \int_0^T \|\boldsymbol{a}(t)\| \left\| \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}(\boldsymbol{z}(t)) \right\| dt \leq C_\theta \int_0^T \|\boldsymbol{a}(t)\| \, dt. \tag{219}$$

The finite-time bounds in (i) follow directly from Theorem 5. Indeed, that theorem gives, for every $t \in [0,T]$

$$\|\boldsymbol{a}(t)\| \leq \|\boldsymbol{a}_T\| \exp\Big(\lambda_{\max}^{FT}(t,T) \, (T-t)\Big), \tag{220}$$

and, if $\beta(t,T) \neq 0$

$$\|\boldsymbol{a}(t)\| \geq |\beta(t,T)| \exp\Big(\lambda_{\max}^{FT}(t,T)\,(T-t)\Big). \tag{221}$$

Substituting these pointwise bounds into the integral representation and using $\|\partial \boldsymbol{f}/\partial \boldsymbol{\theta}\| \leq C_\theta$ yields the FTLE upper and lower bounds in (i).

The asymptotic bounds in (ii) follow directly from Theorem 6. Indeed, that theorem gives, for every $\epsilon > 0$ and for sufficiently large $T - t$

$$\|\boldsymbol{a}(t)\| \leq \|\boldsymbol{a}_T\|\, M(\epsilon)\, e^{(\lambda_{\max}(t)+\epsilon)(T-t)}, \tag{222}$$

and, under the assumptions of Oseledec's multiplicative ergodic theorem, for terminal adjoint vectors $\boldsymbol{a}_T$ having a nonzero component in the top dual Oseledec subspace associated with $\lambda_{\max}(t)$

$$\|\boldsymbol{a}(t)\| \geq m(\epsilon, \boldsymbol{a}_T)\, e^{(\lambda_{\max}(t)-\epsilon)(T-t)}. \tag{223}$$

Substituting these pointwise bounds into the integral representation yields the LE upper and lower bounds in (ii).

The regime consequences in (iii) and (iv) follow by evaluating the elementary exponential integrals in (i) and (ii), respectively, according to the sign of $\lambda_{\max}^{FT}(t,T)$ and $\lambda_{\max}(t)$. This completes the proof. □

F EXPERIMENT DETAILS

We provide details on the datasets and hyperparameters for all experiments in this chapter. Each problem has a dedicated section. We also include information on our calculation of spectral residuals in Appendix F.6 and an experiment demonstrating the exploding gradient problem for the adjoint sensitivity method in Appendix F.7.

Our implementation is based on the existing python library for learning operators from data called `datafold` (Lehmberg et al., 2020). Derivatives were computed using a simple finite difference scheme whenever necessary. Further, our implementation appends the raw states to the dictionary in order to facilitate a mapping from the observables space back into the state space, for this reason the size of dictionaries is typically the sum of the number of neurons and the dimension of the state.

The experiment code and datasets are provided as supplementary material. All computations were performed on a AMD EPYC 7702 64-Core Processor.

F.1 VAN DER POL

The Van der Pol oscillator is a good example of a nonlinear system with simple dynamics and a well-understood behavior. The dynamics are given by

$$\begin{aligned} \dot{z_1} &= z_2 \\ \dot{z_2} &= \mu(1 - z_1^2)z_2 - z_1, \end{aligned} \tag{224}$$

and we set $\mu = 1$. We generate an time-array of irregularly spaced timepoints by drawing each $\Delta t \sim U(0, 0.05)$. Each trajectory consists of 200 snapshots in time. For training, validation and testing we simulate 50 trajectories for each dataset, with each trajectory starting from a randomly sampled initial condition $(z_1, z_2) \sim U(-3, +3) \times U(-3, +3)$.

We perform a grid search over possible hyperparameter combinations for the different dictionary choices, these are given in Table 3, Table 4, Table 5, Table 6, Table 7 for the DRAGONS approach, Gaussian parameters, learned dictionary with Adam, an RBF dictionary and a polynomial dictionary, respectively. The range of hyperparameters was determined via manual trials in advance of the grid search. We evaluate each model on the validation set using a mean squared error loss and averaging this over the 50 trajectories. All models are initialized 5 times with different random seed, and we report the average, minimum and maximum result across the random seeds in Table 2.

Table 3: Hyperparameter search scope for DRAGONS on the Van der Pol problem. Best combination shown with bold font.

sampling	flow-, vector field-, **generator-driven**
layer width	16, 32, 64, 128, 256, **512**
activation	relu, **tanh**
rcond	1e-6, 1e-10, 1e-14, **1e-18**

Table 4: Hyperparameter search scope for Gaussian random features on the Van der Pol problem. Best combination shown with bold font.

layer width	16, 32, 64, **128**, 256, 512
activation	relu, **tanh**
rcond	1e-6, 1e-10, **1e-14**, 1e-18

Table 5: Hyperparameter search scope for gEDMD-DL on the Van der Pol problem. Best combination shown with bold font.

layer width	16, 32, 64, **128**, 256, 512
activation	relu, **tanh**
rcond	1e-6, 1e-10, 1e-14, **1e-18**

We visualize predictions using the DRAGONS approach on the test set in Figure 6. The predicted trajectory matches the trajectory obtained with a numerical ODE solver.

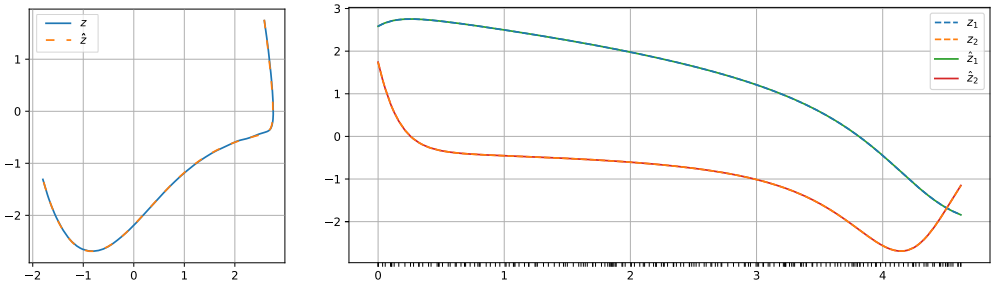
Further, we investigate potential ill-conditioning of the system to be solved using least squares. For this purpose, we compute the condition number of $\Phi_N(X)^T \Phi_N(X)$ for each dictionary choice and provide these in Table 8. A higher number of dictionary elements is more likely to lead to a high condition number due to an increased likelihood for similar features. We observe that random features approaches have better conditioning than Adam-trained dictionaries of the same size.

Table 6: Hyperparameter search scope for RBF-Dict on the Van der Pol problem.
Best combination shown with bold font.

num of samples	100, **200**
rcond	1e-8, **1e-12**, 1e-16

Table 7: Hyperparameter search scope for Poly-Dict on the Van der Pol problem.
Best combination shown with bold font.

degree	1, 2, **3**
rcond	1e-8, 1e-10, 1e-12, 1e-14, **1e-16**



(a) Phase space (b) Trajectory

Figure 6: Comparison of numerically calculated states z versus DRAGONS predicted trajectories $\hat{z}$ for Van der Pol experiment. In subplot b) the ticks on the x-axis show the evaluation points in time, which are irregularly spaced.

F.2 Lorenz 63

Our second experiment is concerned with the Lorenz system (Lorenz, 1963) in the chaotic regime with $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$ given as

$$\begin{aligned} \dot{z_1} &= \sigma(z_2 - z_1) \\ \dot{z_2} &= z_1(\rho - z_3) - z_2 \\ \dot{z_3} &= z_1\, z_2 - \beta z_3. \end{aligned} \tag{225}$$

We generate datasets by numerically solving the system using an 8th order explicit Runge Kutta scheme at time values where each consecutive $\Delta t \sim U(0, 0.03)$ using 1000 evaluation points for one orbit, and creating 50 orbits for training, validation and for test datasets. The initial conditions for each new orbit are sampled according to $(z_1, z_2, z_3) \sim U(-20, +20) \times U(-20, +20)$.

We perform a hyperparameter search similarly to the Van der Pol problem, and report the combinations of parameters in Table 9, Table 10, Table 11, Table 12 and Table 13 for the DRAGONS, Gaussian parameters, gEDMD-DL RBF dictionary and polynomial dictionary, respectively. Since this is a chaotic system, using the mean squared error metric is not suitable, thus we use an empirical estimate of Kullback-Leibler (KL) divergence between the true and predicted orbits, which should be very similar for very long trajectories due to ergodic properties of the system.

Table 9: Hyperparameter search scope for DRAGONS on the Lorenz 63 problem.
Best combination shown with bold font.

sampling	flow-, vector field-, **generator-driven**
layer width	16, 32, 64, 128, **256**, 512
activation	**relu**, tanh
rcond	1e-2, 1e-6, 1e-10, **1e-14**, 1e-18

To this end, we treat the trajectories as distributions, denoting the true distribution as $p(\boldsymbol{z})$ and the predicted distribution as $q(\boldsymbol{z})$. KL divergence is defined as

$$D_{KL}(p(\boldsymbol{x}) \| q(\boldsymbol{x})) = \int_{\boldsymbol{x} \in \mathbb{R}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x}. \tag{226}$$

.

dictionary	DRAGONS	Gaussian	gEDMD-DL	RBF-Dict	Poly-Dict
#dict	130	130	130	202	9
cond num	1.18e+14	2.32e+13	9.86e+18	3.77e+18	3.97e+02

Table 8: Condition numbers of different dictionaries for Van der Pol problem.

To empirically estimate this quantity we use an approach similar to Hess et al. (2023) to use Gaussian Mixture Models (GMM) placed on the true trajectory $\boldsymbol{z}$ and predicted $\hat{\boldsymbol{z}}$ trajectories

$$\hat{p}(\boldsymbol{z}) = \frac{1}{T}\sum_{t=1}^{T}\mathcal{N}(\boldsymbol{z}, \boldsymbol{z}_t, \Sigma) \qquad \hat{q}(\boldsymbol{z}) = \frac{1}{T}\sum_{t=1}^{T}\mathcal{N}(\boldsymbol{z}, \hat{\boldsymbol{z}}_t, \Sigma) \tag{227}$$

for T points in time; we assume unit variance. These GMMs are then sampled to approximate the integral in Equation (226) as

$$D_{KL}(\hat{p}(\boldsymbol{z})\|\hat{q}(\boldsymbol{z})) \approx \frac{1}{1000}\sum_{i=1}^{1000}\log\frac{\hat{p}(\boldsymbol{z})}{\hat{q}(\boldsymbol{z})}. \tag{228}$$

In Figure 3 we show predictions on a test trajectory with our DRAGONS approach.

Table 10: Hyperparameter search scope for Gaussian random features on the Lorenz 63 problem.
Best combination shown with bold font.

layer width	16, 32, **64**, 128, 256, 512
activation	relu, **tanh**
rcond	1e-6, **1e-10**, 1e-14, 1e-18

Table 11: Hyperparameter search scope for gEDMD-DL on the Lorenz 63 problem.
Best combination shown with bold font.

layer width	**16**, 32, 64, 128, 256, 512
activation	relu, **tanh**
rcond	**1e-6**, 1e-10, 1e-14, 1e-18

Table 12: Hyperparameter search scope for RBF-Dict on the Lorenz 63 problem.
Best combination shown with bold font.

num of samples	100, 200, **300**
rcond	1e-4, **1e-8**, 1e-12, 1e-16

Table 13: Hyperparameter search scope for Poly-Dict on the Lorenz 63 problem.
Best combination shown with bold font.

degree	1, 2, **3**
rcond	1e-6, 1e-8, 1e-10, 1e-12, **1e-14**, 1e-16

To investigate the conditioning of the linear system we again report the condition number of $\Phi_N(X)^T\Phi_N(X)$, for the Lorenz 63 problem the results are given in Table 14. Here we see that even when random feature dictionaries are much larger than the Adam-trained dictionary, they lead to a better conditioned linear system.

F.3 Flow around a cylinder

The simulation data for the flow past a cylinder at Reynolds number $Re = 160$ are taken from the Computer Graphics Laboratory at ETH Zurich (Popinet, 2004; Günther et al., 2017). The corresponding unstable steady-state solution is computed by enforcing symmetric boundary conditions at $y = 0$. Using these data, we construct a three-dimensional spatial basis consisting of two oscillatory modes and one shift mode, see Figure 7a. The oscillatory modes ϕ_1 and ϕ_2 are obtained as the first two POD modes of the velocity fluctuations around the mean limit-cycle flow (Sirovich, 1987). The shift mode ϕ_{shift} represents the change from the unstable steady flow to the mean limit-cycle flow. The corresponding temporal coordinates are modeled using a three-dimensional mean-field model following Noack et al. (2003):

$$\begin{aligned}\dot{z}_1 &= \mu z_1 - \omega z_2 + A z_1 z_3,\\ \dot{z}_2 &= \omega z_1 + \mu z_2 + A z_2 z_3,\\ \dot{z}_3 &= -\lambda(z_3 - z_1^2 - z_2^2).\end{aligned} \tag{229}$$

dictionary	DRAGONS	Gaussian	gEDMD-DL	RBF-Dict	Poly-Dict
#dict	131	67	19	303	34
cond num	6.85e+09	1.07e+08	3.24e+18	1.60e+12	1.72e+10

Table 14: Condition numbers of different dictionaries for the Lorenz 63 problem.

Here, z_1 and z_2 are the normalized oscillatory coordinates and z_3 is the shift coordinate. The angular frequency ω is identified from the developed limit-cycle data by fitting the unwrapped phase of the oscillatory coordinates. The transient growth rate is set to $\mu = 0.1|\omega|$ and the shift relaxation rate to $\lambda = 5|\omega|$. The nonlinear coupling coefficient is chosen as $A = -\mu$, such that the radial growth rate vanishes at $z_3 = 1$. With these choices, the system has an unstable equilibrium at the origin and a stable limit cycle defined by

$$z_1^2 + z_2^2 = 1, \qquad z_3 = 1.$$

The dataset is generated by numerically solving the reduced dynamical system. Each trajectory consists of 600 snapshots separated by irregular time increments, independently sampled according to $\Delta t \sim U(0, 0.01)$. We generate 50 trajectories each for training, validation, and testing. The initial condition for each trajectory is sampled close to the parabolic slow manifold $z_3 = z_1^2 + z_2^2$ by drawing

$$(r, \theta) \sim U(0.05, 1.1) \times U(0, 2\pi),$$

and setting

$$z_1 = r\cos\theta, \qquad z_2 = r\sin\theta, \qquad z_3 = r^2 + \delta,$$

where δ is sampled uniformly from the admissible interval within $[-0.3, 0.3]$ such that $0 \leq z_3 \leq 1.2$.

We again perform a hyperparameter search following the same procedure as for the Van der Pol and Lorenz 63 problems. The considered hyperparameter combinations are given in Table 15, Table 16, Table 17, Table 18 and Table 19 for DRAGONS, Gaussian random parameters, gEDMD-DL using an Adam optimizer, RBF-Dict and Poly-Dict, respectively. The search range was determined manually by testing out sensible combinations. We evaluate on the validation set using a mean squared error loss to select the best hyperparameter combination, which then is evaluated on a test set and reported in Table 2. Similarly to the other forecasting experiments, a generator-driven sampling yields the best results.

Table 15: Hyperparameter search scope for DRAGONS on the cylinder flow problem.
Best combination shown with bold font.

sampling	flow-, vector field-, **generator-driven**
layer width	**128**, 256, 512
activation	**relu**, tanh
rcond	1e-8, 1e-10, **1e-12**, 1e-14

Table 16: Hyperparameter search scope for Gaussian random features on the cylinder flow problem.
Best combination shown with bold font.

layer width	**128**, 256, 512
activation	relu, **tanh**
rcond	1e-8, 1e-10, 1e-12, 1e-14, **1e-18**

Table 17: Hyperparameter search scope for gEDMD-DL on the cylinder flow problem.
Best combination shown with bold font.

layer width	16, **32**, 64, 128, 256, 512
activation	relu, **tanh**
rcond	1e-6, 1e-8, 1e-10, 1e-12, **1e-14**

Table 18: Hyperparameter search scope for RBF-Dict on the cylinder flow problem.
Best combination shown with bold font.

degree	**100**, 300, 500
rcond	1e-6, **1e-10**, 1e-14

Table 19: Hyperparameter search scope for Poly-Dict on the cylinder flow problem.
Best combination shown with bold font.

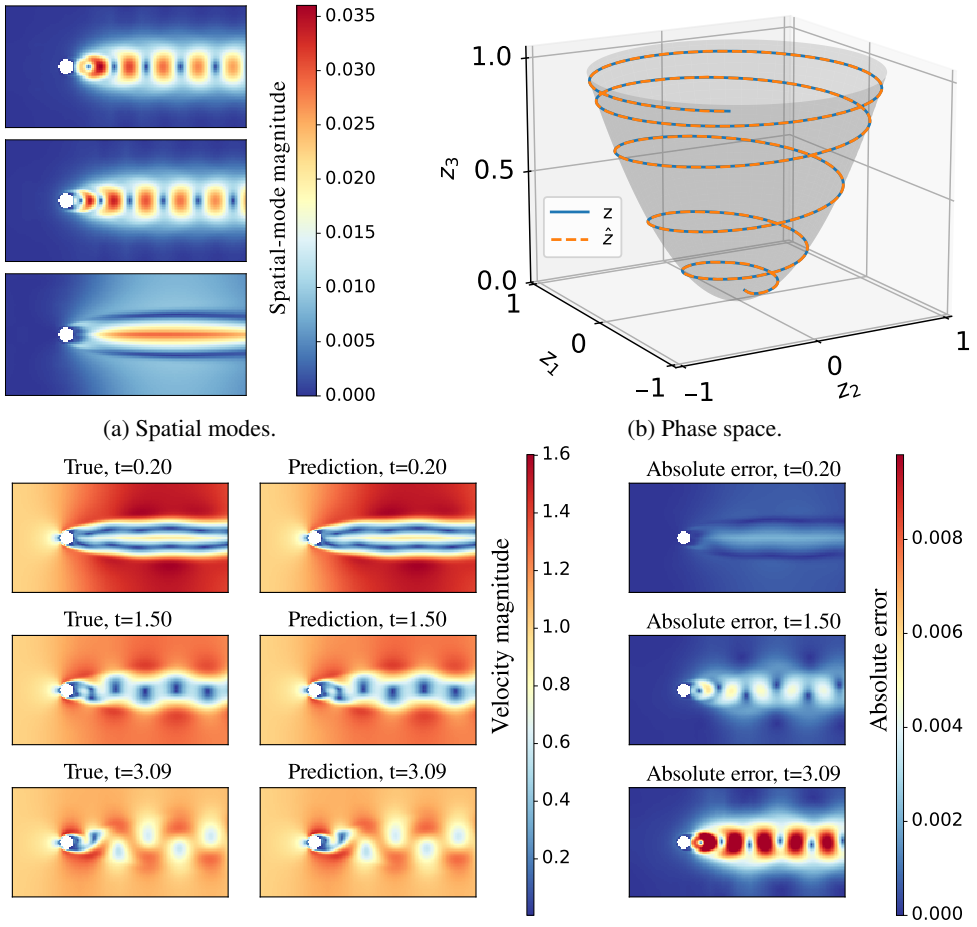
degree	2, **3**, 4
rcond	**1e-8**, 1e-12, 1e-16

In this case again we check the condition number of feature matrices and report them in Table 20. Once again the worst conditioning is attained by the iteratively trained neural network dictionary.

dictionary	DRAGONS	Gaussian	gEDMD-DL	RBF-Dict	Poly-Dict
#dict	515	131	35	103	34
cond num	5.76e+13	1.30e+15	3.55e+17	1.17e+05	7.47e+05

Table 20: Condition numbers of different dictionaries for the flow around a cylinder problem.

Figure 7b shows the true and predicted trajectory of one test sample. The full velocity field can be reconstructed by mapping the reduced coordinates to the physical modal coefficients associated with the three spatial modes and adding them to the unstable steady base flow. Figure 7c shows the true and predicted velocity fields at different time steps and the corresponding absolute error clipped at the 99th percentile. The prediction closely follows the reference trajectory in the reduced phase space and captures the dominant spatial flow structures of the reconstructed velocity field.



(a) Spatial modes.

(b) Phase space.

(c) Velocity field reconstruction.

Figure 7: Flow around a cylinder experiment. (a) Two oscillatory modes and one shift mode. (b) Reference and DRAGONS-predicted reduced trajectories in phase space. (c) Reference and predicted velocity fields with the corresponding absolute error, clipped at the 99th percentile.

To complement the forecasting results, we examine the spectrum of the learned Koopman generator and the associated Koopman modes. In Figure 4b, mode 28 corresponds to an eigenvalue close to zero and represents an almost stationary contribution, while mode 64 represents a transient decaying mode. Together with its complex-conjugate counterpart, mode 31 in Figure 4c represents an oscillatory mode whose eigenvalue has an imaginary part with magnitude close to the estimated vortex-shedding angular frequency $|\omega|$.

F.4 Alanine Dipeptide

We consider a molecular dynamics simulation of the alanine dipeptide molecule in explicit water. The alanine dipeptide molecule consists of 22 atoms, however we consider only the 10 heavy atoms, as depicted in Figure 8 and their position in physical space, resulting in a state vector $\boldsymbol{z} \in \mathbb{R}^{30}$. This system has been commonly studied in computational modeling (Nüske et al., 2017; Klus et al., 2018; Herwig et al., 2025), thus it represents a robust benchmark on which we test our approach. The datasets are downloaded from April Herwig (2024), and subsampled to every 100th entry.

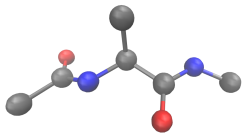


Figure 8: Molecular representation of alanine dipeptide (only heavy atoms are shown).

In this experiment our aim is not to demonstrate predictive accuracy but rather to perform a conformation analysis. Since our results are not quantifiable we do not directly compare with the existing results from existing approaches in the manuscript; nonetheless the obtained insights align with prior results, the interested reader is invited to make comparison with the cited publications.

We use a DRAGONS model with 58 neurons with a ReLU activation in the hidden layer, where the parameters are sampled using the flow-driven sampling (Equation (28)). In the linear system solver we use a truncated SVD with a cut-off ratio of 0.6. The hyperparameters were found using a manual search. To perform a clustering we use the k-means algorithm and create three clusters using the five eigenfunctions whose eigenvalues are closest to zero. The clustering is performed in the 30 dimensional state space of the system. However for the purpose of visualization, we map the positions of the molecule to the conformation described by its two dihedral angles as proposed by Ramachandran et al. (1963) to obtain plots in 2D. The model does not have access to the dihedral angles. We visualize two of these eigenfunctions in Figure 5b and the resulting clustering in Figure 5c. The results show clear distinctions in the appropriate regions where the metastable states reside, this aligns with previous observations obtained in prior work.

F.5 Comparison of sampling strategies

We provide the results across different sampling strategies for the forecasting problems in Table 21. The flow-driven sampling already provides a good baseline, while the vector-field-driven sampling is occasionally inaccurate. The generator driven sampling is robust and yields the lowest error on all forecasting problems.

Table 21: Comparison of different sampling strategies on the forecasting tasks.

Example	sampling distribution	Error: avg (min, max)
Van der Pol	flow-driven	4.22e-6 (2.71e-6, 7.74e-6)
	vector-field-driven	2.22 (1.35e-7, 1.11e+1)
	generator-driven	6.23e-7 (2.84e-7, 8.08e-7)
Lorenz 63	flow-driven	5.27e-4 (1.10e-4, 9.53e-4)
	vector-field-driven	4.20e-4 (1.97e-4, 6.54e-4)
	generator-driven	3.75e-4 (1.87e-4, 5.69e-4)
Cylinder	flow-driven	8.68e-4 (4.51e-7, 4.33e-3)
	vector-field-driven	6.09e-7 (3.67e-7, 1.21e-6)
	generator-driven	3.35e-7 (3.28e-7, 3.42e-7)

F.6 gEDMD SPECTRAL RESIDUALS

The spectra of the Koopman operator and generator are of interest because the dynamical system's behavior is determined by them. Computing the spectrum of an approximated operator is also subject to inaccuracies arising from a finite-dimensional approximation. Discretizations can lead to spurious eigenvalues that are not related to the underlying operator (Colbrook & Townsend, 2024). Thus we use an approach to identify the presence of spectral pollution based on an existing approach by Colbrook & Townsend (2024) by adjusting their approach for the Koopman generator.

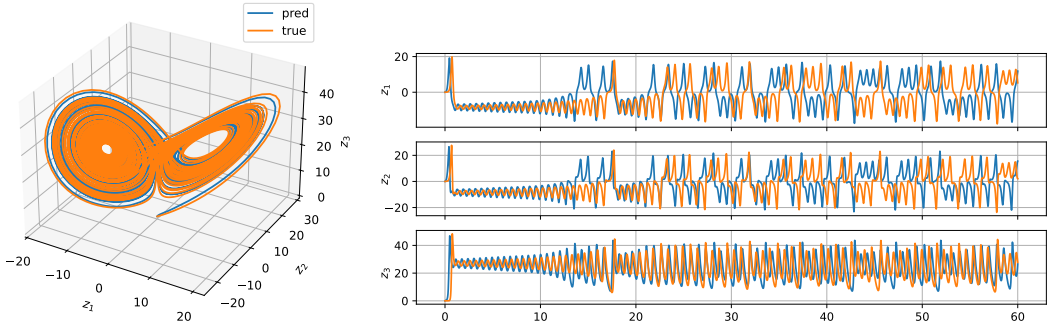
The suitability of an eigenvalue-eigenvector pair λ, χ of the generator $\mathcal{L}$ can be evaluated by the following relative residual

$$\begin{aligned} \text{res} &= \frac{\int_{\mathcal{X}} |\mathcal{L}\chi(\boldsymbol{x}) - \lambda\chi(\boldsymbol{x})|^2 d\mu(\boldsymbol{x})}{\int_{\mathcal{X}} |\chi(\boldsymbol{x})|^2 d\mu(\boldsymbol{x})} \\ &= \frac{\langle (\mathcal{L}-\lambda)\chi, (\mathcal{L}-\lambda)\chi \rangle}{\langle \chi, \chi \rangle} \\ &= \frac{\sum_{i,j=1}^{N} \bar{\chi}_i \chi_j \left[\langle \mathcal{L}\phi_i, \mathcal{L}\phi_j \rangle - \lambda \langle \phi_i, \mathcal{L}\phi_j \rangle - \bar{\lambda} \langle \mathcal{L}\phi_i, \phi_j \rangle + |\lambda|^2 \langle \phi_i, \phi_j \rangle \right]}{\sum_{i,j=1}^{N} \bar{\chi}_i \chi_j \langle \phi_i, \phi_j \rangle}. \end{aligned} \tag{230}$$

In the context of finite data and a finite sized dictionary, we approximate this residual by

$$\hat{\text{res}} = \frac{\sum_{i,j=1}^{N} \bar{\hat{\chi}}_i \hat{\chi}_j \left[(\dot{\Phi}_X \dot{\Phi}_X^*)_{i,j} - \lambda (\Phi_X \dot{\Phi}_X^*)_{i,j} - \bar{\lambda} (\dot{\Phi}_X \Phi_X^*)_{i,j} + |\lambda|^2 (\Phi_X \Phi_X^*)_{i,j} \right]}{\sum_{i,j=1}^{N} \bar{\hat{\chi}}_i \hat{\chi}_j (\Phi_X \Phi_X^*)_{i,j}}. \tag{231}$$

The Galerkin matrices $\hat{\mathbf{C}}_{NM} = \frac{1}{M} \dot{\Phi}_X \Phi_X^*$ and $\hat{\mathbf{G}}_{NM} = \frac{1}{M} \Phi_X \Phi_X^*$ from Equation (24) already provide much of the necessary calculations, and in addition we compute $\dot{\Phi}_X \dot{\Phi}_X^*$.

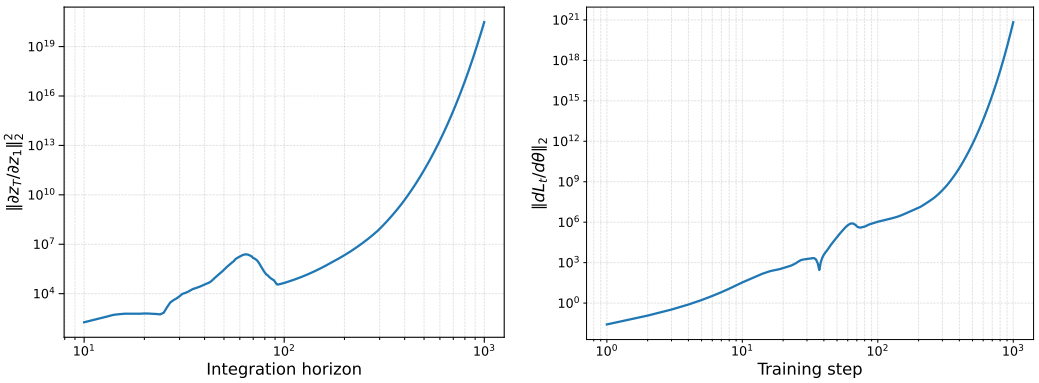


(a) Phase space (b) Trajectory

Figure 9: Comparison of numerically calculated versus predicted trajectories for Lorenz experiment. Ticks on the x-axis show the evaluation points in time, which are irregularly spaced.

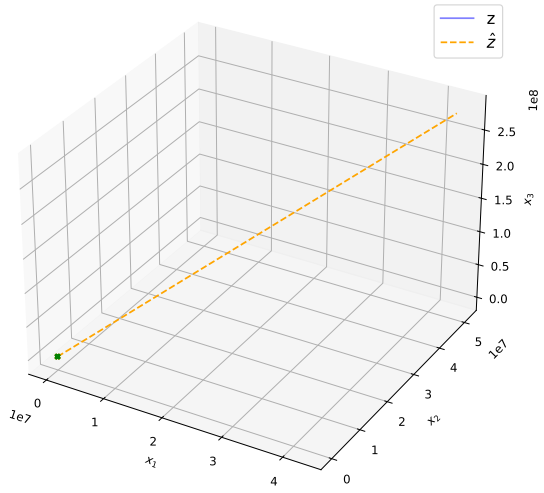
F.7 EXPLODING GRADIENT PROBLEM IN ADJOINT SENSITIVITY METHOD

In Figure 10a, we empirically demonstrate the explosion of the flow-map Jacobian and adjoint state norms, discussed in Section 3.3 and proved in Appendix E for a neural ODE trained on the Lorenz-63 system. This experiment follows Mikhaeil et al. (2022), who theoretically established and empirically demonstrated gradient explosion in RNNs trained on chaotic Lorenz dynamics. The gradient of the adjoint state with respect to the model parameters exhibits a similar exponential growth, as shown in Figure 10b, making long-horizon training particularly challenging for chaotic systems. This behavior is demonstrated using a Neural ODE trained on the Lorenz–63 system, with the implementation adapted from Allauzen et al. (2022). Their work investigates the use of adaptive numerical solvers within Neural ODE training and shows that a naive application of adaptive time



(a) Flow-map Jacobian of the receding state w.r.t. the initial one.

(b) Gradient of the adjoint state w.r.t. the neural network parameters.



(c) Phase space of neural ODE prediction for Lorenz-63.

Figure 10: Explosion of the neural ODE's adjoint sensitivity when trained on the chaotic Lorenz-63 system as the integration horizon increases.

stepping is insufficient, motivating a tighter coupling between the solver and the training procedure. Our results nevertheless show that the observed gradient growth is intrinsic to the dynamics and does not depend on the particular ODE solver. A forecasting with neural ODE of the Lorenz system becomes impossible due to explosion, as seen on Figure 10c. We perform forecasting on the same trajectory with our already trained DRAGONS model on Lorenz-63 from our forecasting experiment. The predictions are on the attractor, and are visualized in Figure 9, and the prediction error is 7.45e-2.