

Koopman-Based Learning of Infinitesimal Generators without Operator Logarithm

Yiming Meng, Ruikun Zhou, Melkior Ornik, and Jun Liu

Abstract—To retrieve transient transition information of unknown systems from discrete-time observations, the Koopman operator structure has gained significant attention in recent years, particularly for its ability to avoid time derivatives through the Koopman operator logarithm. However, the effectiveness of these logarithm-based methods has only been demonstrated within a restrictive function space. In this paper, we propose a logarithm-free technique for learning the infinitesimal generator without disrupting the Koopman operator learning framework.

Index Terms—Unknown nonlinear systems, Koopman operators, infinitesimal generator, system identification, verification.

I. INTRODUCTION

Verification of dynamical system properties and achieving autonomy are two important directions for the future of industrial intelligence, with applications in numerous fields, including mathematical finance, automated vehicles, power systems, and other physical sciences.

Witnessing the success of problem-solving within the data paradigm, there has been a surge of interest in revealing the governing equations of continuous-time dynamical systems from time-series data to better understand the underlying physical laws [1], [2]. Additional interests in safety-critical industries include data-driven stability and safety analysis, prediction, and control. Techniques such as Lyapunov and barrier certificates have proven effective [3]–[7]. It is worth noting, however, that these practical concerns require information on the vector fields, the value functions that abstract system performance, and the corresponding Lie derivatives, all underpinned by an understanding of the infinitesimal generator [8]–[12]. Considering nonlinear effects, challenges therefore arise in the converse identification of infinitesimal system transitions based on discrete-time observations that represent cumulative trajectory behaviors.

This research was supported in part by NASA under grant numbers 80NSSC21K1030 and 80NSSC22M0070, by the Air Force Office of Scientific Research under grant number FA9550-23-1-0131, and in part by the Natural Sciences and Engineering Research Council of Canada and the Canada Research Chairs program.

Yiming Meng is with the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA. yymeng@illinois.edu.

Ruikun Zhou is with the Department of Applied Mathematics, University of Waterloo, Waterloo ON N2L 3G1, Canada ruikun.zhou@uwaterloo.ca.

Melkior Ornik is with the Department of Aerospace Engineering and the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA. mornik@illinois.edu.

Jun Liu is with the Department of Applied Mathematics, University of Waterloo, Waterloo ON N2L 3G1, Canada j.liu@uwaterloo.ca.

Direct methods, such as the sparse identification of nonlinear dynamics (SINDy) algorithm [13], have been developed to identify state dynamics by relying on nonlinear parameter estimation [14], [15] and static linear regression techniques. However, the accurate approximation of time derivatives of the state may not be achieved due to challenges such as low sampling rates, noisy measurements, and short observation periods. Furthermore, the data cannot be reused in the proposed structure for constructing other value functions (e.g., Lyapunov or barrier functions) for stability, reachability, and safety analysis. This limitation extends to verifying their Lie derivatives along the trajectories, which is crucial for demonstrating the evolving trends of the phase portraits.

Comparatively, the operator logarithm-based Koopman generator learning structure [16]–[19] does not require the estimation of time derivatives, enabling a data-driven estimation of Lie derivatives. This approach can potentially circumvent the need for high sampling rates and extended observation periods. Heuristically, researchers tend to represent the Koopman operator \mathcal{K}_t by an exponential form of its infinitesimal generator \mathcal{L} as $\mathcal{K}_t = e^{t\mathcal{L}}$, leading to the converse representation $\mathcal{L} = \frac{1}{t} \log(\mathcal{K}_t)$ for any $t > 0$. However, representing Koopman operators in exponential form requires the boundedness of the generator. Additionally, the operator logarithm is a single-valued mapping only within a specific sector of the spectrum. Recent studies [20], [21] have investigated the sufficient and necessary conditions under which the Koopman-logarithm-based generator learning method can be uniquely identified. However, these conditions are less likely to be verifiable for unknown systems.

In this paper, we introduce a logarithm-free generator learning scheme that does not require knowledge of the spectrum properties and present preliminary results. This scheme will be compatible with the current advances in [7] for Koopman-based construction of maximal Lyapunov functions. It is important to note that the method in [7] assumes full knowledge of the equilibrium point and acknowledges that verification of the constructed Lyapunov function depends on information about the actual system transitions, which may diminish its predictive value in stability analysis.

The rest of the paper is organized as follows. Preliminaries are covered in Section II. The formal finite-horizon approximation of the generator using Koopman operators is presented in Section III. The data-driven algorithm is discussed in Section IV. Finally, case studies are presented in Section V.

Due to page limitations and to enhance readability, this paper presents only the primary results of the proposed

method. Detailed proofs are omitted and will be published elsewhere.

Notation: We denote by \mathbb{R}^n the Euclidean space of dimension $n > 1$, and by \mathbb{R} the set of real numbers. For $x \in \mathbb{R}^n$ and $r \geq 0$, we denote the ball of radius r centered at x by $\mathcal{B}(x, r) = \{y \in \mathbb{R}^n : |y - x| \leq r\}$, where $|\cdot|$ is the Euclidean norm. For a closed set $A \subset \mathbb{R}^n$ and $x \in \mathbb{R}^n$, we denote the distance from x to A by $|x|_A = \inf_{y \in A} |x - y|$ and r -neighborhood of A by $\mathcal{B}(A, r) = \cup_{x \in A} \mathcal{B}(x, r) = \{x \in \mathbb{R}^n : |x|_A \leq r\}$. For a set $A \subseteq \mathbb{R}^n$, \bar{A} denotes its closure, $\text{int}(A)$ denotes its interior and ∂A denotes its boundary. For finite-dimensional matrices, we use the Frobenius norm $\|\cdot\|_F$ as the metric. Let $C(\Omega)$ be the set of continuous functions with domain Ω . We denote the set of i times continuously differentiable functions by $C^i(\Omega)$.

II. PRELIMINARIES

A. Dynamical Systems

Given a pre-compact state space $\mathcal{X} \subseteq \mathbb{R}^n$, we consider a continuous-time nonlinear dynamical system of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \quad \mathbf{x}(0) = x \in \mathcal{X}, \quad t \in [0, \infty), \quad (1)$$

where x denotes the initial condition, and the vector field $f : \mathcal{X} \rightarrow \mathcal{X}$ is assumed to be locally Lipschitz continuous. We denote by $\phi : \mathcal{I} \times \mathcal{X} \rightarrow \mathcal{X}$ the forward flow, also known as the solution map.

The evolution of observable functions of system (1) restricted to $C(\mathcal{X})$ is governed by the family of Koopman operators, defined by

$$\mathcal{K}_t h = h \circ \phi(t, \cdot), \quad h \in C(\mathcal{X}) \quad (2)$$

for each $t \geq 0$, where \circ is the composition operator. By the properties of the flow map, it can be easily verified that $\{\mathcal{K}_t\}_{t \geq 0}$ forms a semigroup, with its (infinitesimal) generator defined by

$$\mathcal{L}h(x) := \lim_{t \rightarrow 0} \frac{\mathcal{K}_t h(x) - h(x)}{t}. \quad (3)$$

Within the space of continuous observable functions, the limit in Eq. (3) exists on the subspace of all continuously differentiable functions, indicating that $\text{dom}(\mathcal{L}) = C^1(\mathcal{X})$. In this case, we can verify that the generator of Koopman operators is such that

$$\mathcal{L}h(x) = \nabla h(x) \cdot f(x).$$

Note that the finite-difference method [22]–[24] indirectly approximates the generator by setting a small terminal time without taking the limit on the r.h.s. of (3). Through this approximation scheme of the time derivative, it can be anticipated that the precision heavily depends on the size of that terminal time.

B. Representation of Koopman Operators

If \mathcal{L} is a bounded linear operator that generates $\{\mathcal{K}_t\}$, then $\mathcal{K}_t = e^{t\mathcal{L}}$ for each t in the uniform (operator) topology. However, many of the differential operators that occur in mathematical physics are unbounded. We revisit some concepts to show how to interpret \mathcal{K}_t in terms of the possibly unbounded generator on an exponential scale.

Definition 2.1 (Resolvents): The resolvent set of \mathcal{L} is defined as $\rho(\mathcal{L}) := \{\lambda \in \mathbb{C} : \lambda I - \mathcal{L} \text{ is invertible}\}$. For any $\lambda \in \rho(\mathcal{L})$, the resolvent operator is defined as

$$R(\lambda; \mathcal{L}) := (\lambda I - \mathcal{L})^{-1}, \quad (4)$$

which is a bounded linear operator [25, Chap. I, Theorem 4.3]. \diamond

We further define the *Yosida approximation* of \mathcal{L} as

$$\mathcal{L}_\lambda := \lambda \mathcal{L} R(\lambda; \mathcal{L}) = \lambda^2 R(\lambda; \mathcal{L}) - \lambda I. \quad (5)$$

Note that $\{\mathcal{L}_\lambda\}_{\lambda \in \rho(\mathcal{L})}$ is a family of bounded linear operators, and $e^{t\mathcal{L}_\lambda}$ is well-defined for each $\lambda \in \rho(\mathcal{L})$.

Definition 2.2 (Strong convergence): Let $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ be a Banach space. Let $B : \mathcal{F} \rightarrow \mathcal{F}$ and $B_n : \mathcal{F} \rightarrow \mathcal{F}$, for each $n \in \mathbb{N}$, be linear operators. Then, the $\{B_n\}_{n \in \mathbb{N}}$ is said to converge to B *strongly*, denoted by $B_n \rightarrow B$, if $\lim_{n \rightarrow \infty} \|B_n h - B h\|_{\mathcal{F}} = 0$ for each $h \in \mathcal{F}$. We also write $B = s\text{-}\lim_{n \rightarrow \infty} B_n$. \diamond

Theorem 2.3: [25, Chap. I, Theorem 5.5] $\mathcal{K}_t = s\text{-}\lim_{\lambda \rightarrow \infty} e^{t\mathcal{L}_\lambda}$ for all $t \geq 0$ on the uniform topology.

III. THE FORMAL APPROXIMATION OF THE INFINITESIMAL GENERATORS

Section II presents preliminary results on representing Koopman operators using the asymptotic approximation of $e^{t\mathcal{L}_\lambda}$ as λ approaches infinity. In this section, we aim to find the correct converse representation of \mathcal{L} based on $\{\mathcal{K}_t\}_{t \geq 0}$. The ultimate goal is to leverage the Koopman learning structure to learn the generator.

Note that it is intuitive to take the operator logarithm such that $\mathcal{L} = \frac{1}{t} \log \mathcal{K}_t$ when \mathcal{L} is bounded. The spectrum's sector should be confined to make the logarithm a single-valued mapping [20], [21]. However, for an unbounded \mathcal{L} , there is no direct connection. In this section, we establish the connection for how the unbounded \mathcal{L} can be properly approximated based on $\{\mathcal{K}_t\}_{t \geq 0}$.

A. Asymptotic Approximations of Generators

Let $C(\mathcal{X})$ be endowed with the uniform norm $\|\cdot\|_\infty := \sup_{x \in \mathcal{X}} |\cdot|$. Suppose that $\{\mathcal{K}_t\}_{t \geq 0}$ is a contraction semigroup where

$$\|\mathcal{K}_t\| := \sup_{\|h\|_{\mathcal{F}}=1} |\mathcal{K}_t h|_\infty \leq 1$$

for all $t \geq 0$. Then, $\rho(\mathcal{L}) \supseteq (0, \infty)$, and by [25, Lemma 1.3.3], we have that $\{\mathcal{L}_\lambda\}_{\lambda > 0}$ converges strongly to \mathcal{L} . Specifically, when working on the $C^2(\mathcal{X})$ subspace, the convergence rate is given by $\mathcal{O}(\lambda^{-1})$.

To address more general cases, we first present the following facts.

Proposition 3.1: For system (1), there exist constants $\omega \geq 0$ and $M \geq 1$ such that $\|\mathcal{K}_t\| \leq Me^{\omega t}$ for all $t \geq 0$. In addition, for any $\lambda \in \mathbb{C}$, the family $\{e^{\lambda t}\mathcal{K}_t\}_{t \geq 0}$ is a C_0 semigroup with generator $\mathcal{L} + \lambda I : \text{dom}(\mathcal{L}) \rightarrow C(\mathcal{X})$.

Intuitively, M represents the uniform scaling of the magnitude of the Koopman operator, while ω indicates the dominant exponential growth or decay rate of the flow on $C(\mathcal{X})$. The above properties provide us with a tool to convert the dominant exponential growth or decay rate of the flow on $C(\mathcal{X})$ by introducing an extra term λI to the original generator. Specifically, the transformation $e^{-\omega t}\mathcal{K}_t$ allows us to work in a topology (by shifting the direction of flows and uniformly compressing by M) that is equivalent to the uniform topology of continuous functions, where the semigroup $\{e^{-\omega t}\mathcal{K}_t\}$ is a contraction. The convergence of $\text{s-lim}_{\lambda \rightarrow \infty} \mathcal{L}_\lambda = \mathcal{L}$ for $\{\mathcal{L}_\lambda\}_{\lambda > \omega}$ and its reciprocal convergence rate can similarly be demonstrated in this new topology.

B. Representation of Resolvent Operators

Motivated by representing \mathcal{L} by $\{\mathcal{K}_t\}_{t \geq 0}$ and the Yosida approximation for \mathcal{L} on $\{\lambda > \omega\}$, we establish a connection between $R(\lambda; \mathcal{L})$ and $\{\mathcal{K}_t\}_{t \geq 0}$.

Proposition 3.2: Let $R(\lambda)$ on $C(\mathcal{X})$ be defined by

$$R(\lambda)h := \int_0^\infty e^{-\lambda t}(\mathcal{K}_t h) dt. \quad (6)$$

Then, for all $\lambda > \omega$,

- 1) $R(\lambda)(\lambda I - \mathcal{L})h = h$ for all $h \in C^1(\mathcal{X})$;
- 2) $(\lambda I - \mathcal{L})R(\lambda)h = h$ for all $h \in C(\mathcal{X})$.

Restricted to the $C^1(\mathcal{X})$ subspace, to use the approximation in Section III-A, we can replace $R(\lambda; \mathcal{L})$ with $R(\lambda)$.

Corollary 3.3: For each $\lambda > \omega$,

$$\mathcal{L}_\lambda = \lambda^2 \int_0^\infty e^{-\lambda t} \mathcal{K}_t dt - \lambda I \quad (7)$$

and $\mathcal{L}_\lambda \rightarrow \mathcal{L}$ on $C^1(\mathcal{X})$.

C. Finite Time-Horizon Approximation

The current form of the improper integral in (7) cannot be directly used for a data-driven approximation. We need to further derive an approximation approach based on finite-horizon observable data. Below, we present a direct truncation modification based on (7).

Definition 3.4: For any $h \in C(\mathcal{X})$ and $\tau \geq 0$, we define $\mathcal{T}_\tau : C(\mathcal{X}) \rightarrow C(\mathcal{X})$ as

$$\mathcal{T}_\tau h(x) := \int_0^\tau e^{-\lambda s} \mathcal{K}_s h(x) ds. \quad (8)$$

It can be shown that for any λ , the truncation in Definition 3.4 results in an error term $e^{-\lambda \tau} R(\lambda)h(\phi(\tau, x))$ that is uniformly bounded for each observable function h , with the dominant term being $e^{-\lambda \tau}$. This indicates that for any arbitrarily large λ , this truncation does not significantly affect the accuracy compared to the procedure from \mathcal{L}_λ to \mathcal{L} .

For any fixed $\tau > 0$, we can therefore use

$$\tilde{\mathcal{L}}_\lambda := \lambda^2 \mathcal{T}_\tau - \lambda I \quad (9)$$

to approximate \mathcal{L} within a small time-horizon. We illustrate this approximation with the following simple example.

Example 3.5: Consider the simple dynamical system $\dot{\mathbf{x}}(t) = \mathbf{x}(t)$ and the observable function $V(x) = x^n$ for any $n \geq 1$. Then, analytically, $\phi(\tau, x) = xe^\tau$ and $\mathcal{L}V(x) = nx^n$. We test the validity of using Eq. (9). Note that, for sufficiently large λ , we have

$$\begin{aligned} & \lambda^2 \int_0^\tau e^{-\lambda s} (\mathcal{K}_s V(x)) ds \\ &= \lambda^2 \int_0^\tau e^{-\lambda s} e^{sn} x^n ds = \lambda^2 x^n \int_0^\tau e^{-(\lambda-n)s} ds \\ &= \frac{\lambda^2 x^n}{\lambda-n} (1 - e^{-(\lambda-n)\tau}) \approx \frac{\lambda^2 x^n}{\lambda-n} \end{aligned}$$

and

$$\lambda^2 \mathcal{T}_\tau V(x) - \lambda V(x) \approx \frac{\lambda^2 x^n}{\lambda-n} - \lambda x^n \approx nx^n = \mathcal{L}V(x).$$

With high-accuracy evaluation of the integral, we can achieve a reasonably good approximation. \diamond

IV. DATA-DRIVEN ALGORITHM

We continue by considering a data-driven algorithm to learn $\tilde{\mathcal{L}}_\lambda$, which indirectly approximate \mathcal{L} as demonstrated in Section III.

The idea is to modify the conventional Koopman operator learning scheme [7], [16], [26], which involves learning linear operators using a finite-dimensional dictionary of basis functions and representing the image function as a linear combination of the dictionary functions. To be more specific, obtaining a fully discretized version \mathbf{L} of the bounded linear operator $\lambda^2 \mathcal{T}_\tau^N - \lambda I$ based on the training data typically relies on the selection of a discrete dictionary of continuously differentiable observable test functions, denoted by

$$\mathcal{Z}_N(x) := [\mathfrak{z}_0(x), \mathfrak{z}_1(x), \dots, \mathfrak{z}_{N-1}(x)], \quad N \in \mathbb{N}. \quad (10)$$

Then, the followings should hold:

- 1) Let $(\mu_i, \xi_i)_{i=0}^{N-1}$ be the eigenvalues and eigenvectors of \mathbf{L} . Let $(\rho_i, \varphi_i)_{i=0}^{N-1}$ be the eigenvalues and eigenfunctions of \mathcal{L} . Then, for each i ,

$$\mu_i \approx \rho_i, \quad \varphi_i(x) \approx \mathcal{Z}_N(x) \xi_i. \quad (11)$$

- 2) For any $h \in \text{span}\{\mathfrak{z}_0, \mathfrak{z}_1, \dots, \mathfrak{z}_{N-1}\}$ such that $h(x) = \mathcal{Z}_N(x) \mathbf{w}$ for some column vector \mathbf{w} , we have that

$$\mathcal{L}h(\cdot) \approx \tilde{\mathcal{L}}_\lambda h(\cdot) \approx \lambda^2 \mathcal{T}_\tau^N h(\cdot) - \lambda h(\cdot) \approx \mathcal{Z}_N(\cdot) (\mathbf{L} \mathbf{w}). \quad (12)$$

In this section, we modify the existing Koopman learning technique to obtain \mathbf{L} .

A. Generating Training Data

For any fixed $\lambda > 0$, given a dictionary \mathcal{Z}_N of the form (10), for each $\mathfrak{z}_i \in \mathcal{Z}_N$ and each $x \in \mathcal{X}$, we consider $\mathfrak{z}_i(x)$ as the features and $\lambda^2 \mathcal{T}_\tau \mathfrak{z}_i(x) - \lambda \mathfrak{z}_i(x) = \lambda^2 \int_0^\tau e^{-\lambda s} \mathfrak{z}_i(\phi(s, x)) ds - \lambda \mathfrak{z}_i(x)$ as the labels. To compute the integral, we employ numerical quadrature techniques for

approximation. This approach inevitably requires discrete-time observations (snapshots), the number of which is denoted by \mathbf{N} , within the interval $[0, \tau]$ of the flow map $\phi(\tau, x)$.

To streamline the evaluation process for numerical examples, drawing inspiration from [27], for any τ and i , we can assess both the trajectory and the integral, i.e. the pair $(\phi(\tau, x), \lambda^2 \int_0^\tau e^{-\lambda s} \mathfrak{z}_i(\phi(s, x)) ds)$, by numerically solving the following augmented ODE system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t)), \quad \mathbf{x}(0) = x \in \mathbb{R}^n, \\ \dot{I}_i(t) &= \lambda^2 e^{-\lambda t} \mathfrak{z}_i(\mathbf{x}), \quad I_i(0) = 0. \end{aligned} \quad (13)$$

We summarize the algorithm for generating training data for one time period as in Algorithm 1.

Algorithm 1 Generating Training Data

Require: f , n , N , \mathcal{X} , \mathcal{Z}_N , τ , λ , and uniformly sampled $\{x^{(m)}\}_{m=0}^{M-1} \subseteq \mathcal{X}$.

for m **from** 0 **to** $M - 1$ **do**

for i **from** 0 **to** $N - 1$ **do**

 Compute $\mathfrak{z}_i(x^{(m)})$

 Compute $\lambda^2 \mathcal{T}_\tau \mathfrak{z}_i(x^{(m)})$ using (13)

end for

 Stack

$$\begin{aligned} \tilde{\mathcal{L}}_\lambda \mathcal{Z}_N(x^{(m)}) &= [\lambda^2 \mathcal{T}_\tau \mathfrak{z}_0(x^{(m)}) - \lambda \mathfrak{z}_0(x^{(m)}), \dots, \\ &\quad \lambda^2 \mathcal{T}_\tau \mathfrak{z}_{N-1}(x^{(m)}) - \lambda \mathfrak{z}_{N-1}(x^{(m)})] \end{aligned}$$

end for

Stack $\mathfrak{X}, \mathfrak{Y} \in \mathbb{C}^{M \times N}$ such that $\mathfrak{X} = [\mathcal{Z}_N(x^{(0)}), \mathcal{Z}_N(x^{(1)}), \dots, \mathcal{Z}_N(x^{(M-1)})]^T$ and $\mathfrak{Y} = [\tilde{\mathcal{L}}_\lambda \mathcal{Z}_N(x^{(0)}), \tilde{\mathcal{L}}_\lambda \mathcal{Z}_N(x^{(1)}), \dots, \tilde{\mathcal{L}}_\lambda \mathcal{Z}_N(x^{(M-1)})]^T$

B. Extended Dynamic Mode Decomposition Algorithm

After obtaining the training data $(\mathfrak{X}, \mathfrak{Y})$ using Algorithm 1, we can find \mathbf{L} by $\mathbf{L} = \operatorname{argmin}_{A \in \mathbb{C}^{N \times N}} \|\mathfrak{Y} - \mathfrak{X}A\|_F$. The \mathbf{L} is given in closed-form as $\mathbf{L} = (\mathfrak{X}^T \mathfrak{X})^\dagger \mathfrak{X}^T \mathfrak{Y}$, where \dagger is the pseudo inverse. Similar to Extended Dynamic Mode Decomposition (EDMD) [26] for learning Koopman operators, the approximations (11) and (12) can be guaranteed. In addition, by the universal approximation theorem, all of the function approximations from above should have uniform convergence to the actual quantities.

It is worth noting that operator learning frameworks, such as autoencoders, which utilize neural networks (NN) as dictionary functions, can reduce human bias in the selection of these functions [5]. Incorporating the proposed learning scheme with NN falls outside the scope of this paper but will be pursued in future work.

Remark 4.1 (The Logarithm Method): We compare the aforementioned learning approach of \mathcal{L} to those obtained using the benchmark approach as described in [16]. Briefly speaking, at a given $s \in [0, \tau]$, that method first obtains a matrix $\mathbf{K} \in \mathbb{C}^{N \times N}$ such that $\mathbf{K} = (\mathfrak{X}^T \mathfrak{X})^\dagger \mathfrak{X}^T \tilde{\mathfrak{Y}}$, where $\tilde{\mathfrak{Y}} = [\mathcal{Z}_N(\phi(s, x^{(0)})), \mathcal{Z}_N(\phi(s, x^{(1)})), \dots, \mathcal{Z}_N(\phi(s, x^{(M-1)}))]^T$. Let $(\mu_i^K, \mathbf{c}_i^K)_{i=0}^{N-1}$ be the eigenvalues and eigenvectors of

\mathbf{K} . Let $(\rho_i^K, \varphi_i^K)_{i=0}^{N-1}$ be the eigenvalues and eigenfunctions of \mathcal{K}_s . Similar to (11) and (12), for each i , we have $\varphi_i^K(x) \approx \mathcal{Z}_N(x) \xi_i^K$ and $\mathcal{K}_s h(\cdot) \approx \mathcal{Z}_N(\cdot) (\mathbf{K} \mathbf{w})$ for $h(x) = \mathcal{Z}_N(x) \mathbf{w}$.

However, even when \mathcal{L} can be represented by $(\log \mathcal{K}_s)/s$, we cannot guarantee that $\frac{\log \mathcal{K}_s}{s} h(\cdot) \approx \mathcal{Z}_N(\cdot) (\frac{\log(\mathbf{K})}{s} \mathbf{w})$, not to mention the case where the above logarithm representation does not hold. Denoting $\Phi(\cdot) = [\varphi_0^K(\cdot), \varphi_1^K(\cdot), \dots, \varphi_{N-1}^K(\cdot)]$ and $\Xi = [\xi_0^K, \xi_1^K, \dots, \xi_{N-1}^K]$, then it is clear that $\Phi(\cdot) = \mathcal{Z}_N(\cdot) \Xi$. The (possibly complex-valued) rotation matrix Ξ establishes the connection between finite-dimensional eigenfunctions and dictionary functions through data-fitting, ensuring that any linear combination within \mathcal{Z}_N can be equivalently represented using Φ with a cancellation of the imaginary parts.

This imaginary-part cancellation effect does not generally hold when applying the matrix logarithm. Suppose the imaginary parts account for a significantly large value, the mutual representation of Φ and \mathcal{Z}_N does not match in the logarithmic scale. An exception holds unless the chosen dictionary is inherently rotation-free with respect to the true eigenfunctions [20], or there is direct access to the data for $\log(\mathcal{K}_s)$ allowing for direct training of the matrix. However, such conditions contravene our objective of leveraging Koopman data to conversely find the generator. In comparison, the approach in this paper presents an elegant method for approximating \mathcal{L} regardless of its boundedness. This enables the direct learning of \mathbf{L} without computing the logarithm, thereby avoiding the potential appearance of imaginary parts caused by basis rotation. \diamond

V. CASE STUDY

We provide a numerical example to demonstrate the effectiveness of the proposed approach. The research code can be found at <https://github.com/Yiming-Meng/Log-Free-Learning-of-Koopman-Generators>.

Consider the Van der Pol oscillator

$$\dot{\mathbf{x}}_1(t) = \mathbf{x}_2(t), \quad \dot{\mathbf{x}}_2(t) = -\mathbf{x}_1(t) + (1 - \mathbf{x}_1^2(t)) \mathbf{x}_2(t),$$

with $\mathbf{x}(0) = x := [x_1, x_2]$. We assume the system dynamics are unknown to us, and our information is limited to the system dimension, $n = 2$, and observations of sampled trajectories. To generate training data using Algorithm 1, for simplicity of illustration, we select $\mathcal{X} = (-1, 1)^2$ and obtain a total of $M = 100^2$ uniformly spaced samples $\{x^{(m)}\}_{m=0}^{M-1}$ in \mathcal{X} . We choose the dictionary as

$$\begin{aligned} \mathcal{Z}_N &= [\mathfrak{z}_{0,0}, \mathfrak{z}_{1,0}, \dots, \mathfrak{z}_{3,0}, \\ &\quad \mathfrak{z}_{0,1}, \mathfrak{z}_{1,1}, \mathfrak{z}_{2,1}, \dots, \mathfrak{z}_{i,j}, \dots, \mathfrak{z}_{3,2}], \quad N = 12, \end{aligned} \quad (14)$$

where $\mathfrak{z}_{i,j}(x_1, x_2) = x_1^i x_2^j$ for each $i \in \{0, 1, 2, 3\}$ and $j \in \{0, 1, 2\}$. We also set $\tau = 1$ and $\lambda = 10^6$. The discrete form \mathbf{L} can be obtained according to Section IV-B. We apply the learned \mathbf{L} to identify the system vector fields, and construct a local Lyapunov function for the unknown system.

$\beta_{i,j}$	$j = 0$	$j = 1$	$j = 2$
$i = 0$	-1.6×10^{-10}	$\mathbf{1} + \mathbf{10}^{-6}$	-4.8×10^{-10}
$i = 1$	-1.3×10^{-5}	-1.9×10^{-9}	-1.4×10^{-7}
$i = 2$	-3.6×10^{-10}	-1.0×10^{-6}	-6.1×10^{-10}
$i = 3$	-1.5×10^{-8}	-2.3×10^{-9}	-2.9×10^{-7}

TABLE I: The weights of \mathcal{Z}_N obtained by \mathbf{Le}_2 .

$\beta_{i,j}$	$j = 0$	$j = 1$	$j = 2$
$i = 0$	-3.8×10^{-10}	$\mathbf{1} - \mathbf{10}^{-5}$	1.3×10^{-9}
$i = 1$	$-\mathbf{1} - \mathbf{10}^{-6}$	9.5×10^{-10}	-2.0×10^{-6}
$i = 2$	8.1×10^{-10}	$-\mathbf{1} - \mathbf{10}^{-6}$	-2.0×10^{-9}
$i = 3$	1×10^{-8}	-6.9×10^{-10}	-3.0×10^{-9}

TABLE II: The weights of \mathcal{Z}_N obtained by \mathbf{Le}_5 .

A. System Identification

The actual vector field is

$$f(x) := [f_1(x), f_2(x)] = [x_2, -x_1 + (1 - x_1^2)x_2].$$

As we can analytically establish that $\mathcal{L}_{\beta_{1,0}}(x) = f_1(x)$ and $\mathcal{L}_{\beta_{0,1}}(x) = f_2(x)$, we use the approximation $[\tilde{\mathcal{L}}_{\lambda\beta_{1,0}}, \tilde{\mathcal{L}}_{\lambda\beta_{0,1}}]$ to conversely obtain f .

Note that $\beta_{1,0}(x) = \mathcal{Z}_N(x)\mathbf{e}_2$ and $\beta_{0,1}(x) = \mathcal{Z}_N(x)\mathbf{e}_5$, where each \mathbf{e}_i for $i \in \{0, 1, \dots, N\}$ is a column unit vector with all components being 0 except for the i -th component, which is 1. To apply \mathbf{L} , we have that $\tilde{\mathcal{L}}_{\lambda\beta_{1,0}}(x) \approx \mathcal{Z}_N(x)(\mathbf{Le}_2)$ and $\tilde{\mathcal{L}}_{\lambda\beta_{0,1}}(x) \approx \mathcal{Z}_N(x)(\mathbf{Le}_5)$.

In other words, we approximate $\tilde{\mathcal{L}}_{\lambda\beta_{1,0}}$ and $\tilde{\mathcal{L}}_{\lambda\beta_{0,1}}$ (and hence f_1 and f_2) using a linear combination of functions within \mathcal{Z}_N . The weights for these approximations are given by \mathbf{Le}_2 and \mathbf{Le}_5 , respectively. We report the corresponding results in Table I and II.

We compare the aforementioned results with those obtained using the benchmark approach as described in [16] with the same M and \mathcal{Z}_N . As described in Remark 4.1, $\mathbf{K} \in \mathbb{C}^{N \times N}$ can be obtained such that $\mathcal{K}_s\beta_{1,0}(x) \approx \mathcal{Z}_N(x)(\mathbf{Ke}_2)$ and $\mathcal{K}_s\beta_{0,1}(x) \approx \mathcal{Z}_N(x)(\mathbf{Ke}_5)$. Then, we have the approximation, as claimed by [16], $f_1(x) \approx \mathcal{Z}_N(x)(\log(\mathbf{K})\mathbf{e}_2/s)$ and $f_2(x) \approx \mathcal{Z}_N(x)(\log(\mathbf{K})\mathbf{e}_5/s)$ w.r.t. $|\cdot|_\infty$. According to [16, Section VI.A], we set $s = 0.5$ to avoid the multi-valued matrix logarithm. We report the weights obtained by taking the real parts of $\log(\mathbf{K})\mathbf{e}_2/s$ and $\log(\mathbf{K})\mathbf{e}_5/s$ in Table III and IV, respectively. Multiple orders of magnitude in accuracy have been established using the proposed method.

It is worth noting that, unlike the experiment in [16] where $i, j \in \{0, 1, 2, 3\}$, we deliberately choose different numbers for i and j , leading to non-negligible imaginary parts after taking the matrix logarithm of \mathbf{K} . The imaginary parts of the learned weights using the Koopman-logarithm approach are reported in Table V and VI. The presence of non-negligible imaginary parts indicates that the underlying system is sensitive to the selection of dictionary functions when employing the Koopman-logarithm approach, an effect that is significant and cannot be overlooked when aiming to minimize human intervention in identifying unknown systems in practice. Furthermore, the original experiment in [16] used $M = 300^2$ samples for a data fitting, and when

$\beta_{i,j}$	$j = 0$	$j = 1$	$j = 2$
$i = 0$	-9.5×10^{-16}	$\mathbf{1} - \mathbf{3.0} \times \mathbf{10}^{-4}$	-4.8×10^{-10}
$i = 1$	$-\mathbf{4.1} \times \mathbf{10}^{-3}$	-3×10^{-15}	$-\mathbf{1.8} \times \mathbf{10}^{-2}$
$i = 2$	-1.7×10^{-16}	$-\mathbf{6.4} \times \mathbf{10}^{-3}$	-7.8×10^{-17}
$i = 3$	$-\mathbf{6.0} \times \mathbf{10}^{-3}$	-6.7×10^{-15}	$-\mathbf{3.7} \times \mathbf{10}^{-2}$

TABLE III: The weights of \mathcal{Z}_N obtained by $\log(\mathbf{K})\mathbf{e}_2/s$.

$\beta_{i,j}$	$j = 0$	$j = 1$	$j = 2$
$i = 0$	-8.6×10^{-16}	$\mathbf{1} + \mathbf{6.5} \times \mathbf{10}^{-3}$	4.1×10^{-16}
$i = 1$	$-\mathbf{1} + \mathbf{3.0} \times \mathbf{10}^{-2}$	-1.6×10^{-15}	$-\mathbf{1.5} \times \mathbf{10}^{-1}$
$i = 2$	1.0×10^{-15}	$-\mathbf{1} - \mathbf{3.8} \times \mathbf{10}^{-2}$	-2.0×10^{-15}
$i = 3$	$-\mathbf{5.8} \times \mathbf{10}^{-2}$	1.8×10^{-15}	$\mathbf{3.3} \times \mathbf{10}^{-1}$

TABLE IV: The weights of \mathcal{Z}_N obtained by $\log(\mathbf{K})\mathbf{e}_5/s$.

M is reduced to 100^2 , the quality of the operator learning deteriorates.

B. Stability Prediction for the Reversed Dynamics

Observing the identified system dynamics, we anticipate that the time-reversed system is (locally) asymptotically stable w.r.t. the origin. We use the learned \mathbf{L} to construct polynomial Lyapunov functions based on the Lyapunov equation $\mathcal{L}V(x) = |x|^2$, where the sign on the r.h.s. is reversed due to the dynamics being reversed. To use \mathbf{L} and the library functions from \mathcal{Z}_N , we define the weights as $\theta := [\theta_0, \theta_1, \dots, \theta_{N-1}]^T$ and seek a θ such that $V(x; \theta) := \mathcal{Z}_N(x)\theta$, with the objective of minimizing $|\mathcal{Z}_N(x)\mathbf{L}\theta - x_1^2 - x_2^2|$. Ignoring the small terms of the magnitude 10^{-9} , the constructed Lyapunov function is given by

$$V(x; \theta) = 1.39x_1^2 - 1.56x_1x_2 + 1.16x_2^2 + 0.74x_1^2x_2^2.$$

Additionally, $\mathcal{L}V(x; \theta)$ is approximated by $\mathcal{Z}_N(x)\mathbf{L}\theta$, with the maximal value observed at $\mathcal{Z}_N(0)\mathbf{L}\theta = -0.068$, which indicates that stability prediction using the data-driven Lyapunov function is verified to be valid. This indicates that the stability prediction, as inferred using the data-driven Lyapunov function, is verified to be valid. The visualization can be found in Fig 1.

Remark 5.1: The idea illustrated above can be expanded to cover a larger region of interest. Specifically, a Zubov equation, instead of a Lyapunov equation, can be solved within the Koopman learning framework using the same dataset [7] as proposed in Algorithm 1. The solution obtained

$\beta_{i,j}$	$j = 0$	$j = 1$	$j = 2$
$i = 0$	-1.8×10^{-17}	$\mathbf{2.0} \times \mathbf{10}^{-3}$	-2.2×10^{-18}
$i = 1$	$\mathbf{5.3} \times \mathbf{10}^{-3}$	6.9×10^{-17}	$-\mathbf{2.3} \times \mathbf{10}^{-2}$
$i = 2$	1.6×10^{-17}	$-\mathbf{8.4} \times \mathbf{10}^{-3}$	-3.0×10^{-17}
$i = 3$	$-\mathbf{8.4} \times \mathbf{10}^{-3}$	-1.1×10^{-16}	$\mathbf{5.0} \times \mathbf{10}^{-2}$

TABLE V: The imaginary parts obtained by $\log(\mathbf{K})\mathbf{e}_2/s$.

$\beta_{i,j}$	$j = 0$	$j = 1$	$j = 2$
$i = 0$	-8.0×10^{-17}	$\mathbf{8.8} \times \mathbf{10}^{-3}$	-9.5×10^{-18}
$i = 1$	$\mathbf{2.3} \times \mathbf{10}^{-2}$	3.0×10^{-16}	$-\mathbf{1.0} \times \mathbf{10}^{-1}$
$i = 2$	6.9×10^{-17}	$-\mathbf{3.7} \times \mathbf{10}^{-2}$	-1.3×10^{-16}
$i = 3$	$\mathbf{3.7} \times \mathbf{10}^{-2}$	-4.9×10^{-16}	$\mathbf{2.2} \times \mathbf{10}^{-1}$

TABLE VI: The imaginary parts obtained by obtained by $\log(\mathbf{K})\mathbf{e}_5/s$.

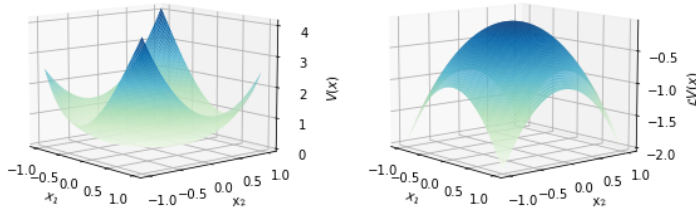


Fig. 1: Data-driven Lyapunov function V and its Lie derivative $\mathcal{L}V$ w.r.t. the predicted system dynamics.

can potentially serve as a Lyapunov function. Since it cannot guarantee the properties of the learned function's derivatives, to confirm it as a true Lyapunov function, the data can be reused as in Algorithm 1 to verify its Lie derivative. \diamond

VI. CONCLUSION

In this paper, we propose a logarithm-free Koopman operator-based learning framework for the infinitesimal generator, demonstrating both theoretical and numerical improvements over the method proposed in [16]. In particular, for more general cases where the generator is unbounded and, consequently, the logarithm of the Koopman operator cannot be used for representation, we draw upon the rich literature to propose an approximation (Eq. (9)) based on Yosida's approximation. A convergence result, along with the convergence rate, is proved in this paper to guide users in tuning the parameters. A numerical example with application in system identification is provided in comparison with the experiment in [16] demonstrating the learning accuracy. Unlike the experiment in [16], where learning accuracy is sensitive to the choice of dictionary functions, the method presented in this paper shows significant improvement in this regard. In applications where automatic computational approaches surpass human computability, such as in constructing Lyapunov-like functions using the Lie derivative, the proposed logarithm-free method holds more promise. We will pursue future efforts to provide more analysis on the sampling rate, numerical simulations, and real-world applications using real data.

REFERENCES

- [1] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.
- [2] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems—a survey on input/output approaches," *Automatica*, vol. 26, no. 4, pp. 651–677, 1990.
- [3] A. Mauroy and I. Mezić, "A spectral operator-theoretic framework for global stability," in *52nd IEEE Conference on Decision and Control*, pp. 5234–5239, IEEE, 2013.
- [4] A. Mauroy and I. Mezić, "Global stability analysis using the eigenfunctions of the Koopman operator," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3356–3369, 2016.
- [5] S. A. Deka, A. M. Valle, and C. J. Tomlin, "Koopman-based neural lyapunov functions for general attractors," in *61st Conference on Decision and Control (CDC)*, pp. 5123–5128, IEEE, 2022.

- [6] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.
- [7] Y. Meng, R. Zhou, and J. Liu, "Learning regions of attraction in unknown dynamical systems via Zubov-Koopman lifting: Regularities and convergence," *arXiv preprint arXiv:2311.15119*, 2023.
- [8] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [9] Y. Lin, E. D. Sontag, and Y. Wang, "A smooth converse lyapunov theorem for robust stability," *SIAM Journal on Control and Optimization*, vol. 34, no. 1, pp. 124–160, 1996.
- [10] J. Liu, Y. Meng, M. Fitzsimmons, and R. Zhou, "Physics-informed neural network lyapunov functions: Pde characterization, learning, and verification," *arXiv preprint arXiv:2312.09131*, 2023.
- [11] Y. Meng, Y. Li, M. Fitzsimmons, and J. Liu, "Smooth converse Lyapunov-barrier theorems for asymptotic stability with safety constraints and reach-avoid-stay specifications," *Automatica*, vol. 144, p. 110478, 2022.
- [12] Y. Meng and J. Liu, "Lyapunov-barrier characterization of robust reach-avoid-stay specifications for hybrid systems," *Nonlinear Analysis: Hybrid Systems*, vol. 49, p. 101340, 2023.
- [13] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [14] J. C. Nash and M. Walker-Smith, "Nonlinear parameter estimation," *An integrated system on BASIC. NY, Basel*, vol. 493, 1987.
- [15] J. M. Varah, "A spline least squares method for numerical parameter estimation in differential equations," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 1, pp. 28–46, 1982.
- [16] A. Mauroy and J. Goncalves, "Koopman-based lifting techniques for nonlinear systems identification," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2550–2565, 2019.
- [17] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, "Data-driven approximation of the Koopman generator: Model reduction, system identification, and control," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132416, 2020.
- [18] Z. Drmač, I. Mezić, and R. Mohr, "Identification of nonlinear systems using the infinitesimal generator of the Koopman semigroup—a numerical implementation of the Mauroy–Goncalves method," *Mathematics*, vol. 9, no. 17, p. 2075, 2021.
- [19] M. Black and D. Panagou, "Safe control design for unknown nonlinear systems with koopman-based fixed-time identification," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11369–11376, 2023.
- [20] Z. Zeng, Z. Yue, A. Mauroy, J. Goncalves, and Y. Yuan, "A sampling theorem for exact identification of continuous-time nonlinear dynamical systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 6686–6692, IEEE, 2022.
- [21] Z. Zeng and Y. Yuan, "A generalized nyquist-shannon sampling theorem using the koopman operator," *arXiv preprint*, 2023.
- [22] J. J. Bramburger and G. Fantuzzi, "Auxiliary functions as Koopman observables: Data-driven analysis of dynamical systems via polynomial optimization," *Journal of Nonlinear Science*, vol. 34, no. 1, p. 8, 2024.
- [23] A. Nejati, A. Lavaei, S. Soudjani, and M. Zamani, "Data-driven estimation of infinitesimal generators of stochastic systems," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 277–282, 2021.
- [24] C. Wang, Y. Meng, S. L. Smith, and J. Liu, "Data-driven learning of safety-critical control with stochastic control barrier functions," in *61st Conference on Decision and Control (CDC)*, pp. 5309–5315, IEEE, 2022.
- [25] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, vol. 44. Springer Science & Business Media, 2012.
- [26] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.
- [27] W. Kang, K. Sun, and L. Xu, "Data-driven computational methods for the domain of attraction and Zubov's equation," *IEEE Transactions on Automatic Control*, 2023.