

# Supervised learning of Lyapunov functions using Laplace averages of approximate Koopman eigenfunctions

Shankar A. Deka, *Member, IEEE* and Dimos V. Dimarogonas, *Fellow, IEEE*

**Abstract**—Modern data-driven techniques have rapidly progressed beyond modelling and systems identification, with a growing interest in learning high-level dynamical properties of a system, such as safe-set invariance, reachability, input-to-state stability etc. In this paper, we propose a novel supervised Deep Learning technique for constructing Lyapunov certificates, by leveraging Koopman Operator theory-based numerical tools (Extended Dynamic Mode Decomposition and Generalized Laplace Analysis) to robustly and efficiently generate *explicit ground truth data* for training. This is in stark contrast to existing Deep Learning methods where the loss functions plainly penalize Lyapunov condition violation in the absence of labelled data for direct regression. Furthermore, our approach leads to a linear parameterization of Lyapunov candidate functions in terms of stable eigenfunctions of the Koopman operator, making them more interpretable compared to standard DNN-based architecture. We demonstrate and validate our approach numerically using 2-dimensional and 10-dimensional examples.

**Index Terms**—Data-driven modeling, Koopman operator, Lyapunov function, Machine learning, Neural networks

## I. INTRODUCTION

DEVELOPING formal guarantees for safety and performance of dynamical systems is an essential step in the design and control of cyber-physical systems. Lyapunov functions primarily provide stability and robustness guarantees for a large class of dynamical systems [1][2]. More recently, data-driven certification of such critical behaviours and properties within cyber-physical systems has become an area of growing interest to the controls and machine learning communities alike. In the absence of a well-known dynamical model, leveraging trajectory data for control and analysis is particularly well-suited. Even for the cases where we have a model, it often takes expert intuition and knowledge to handcraft Lyapunov functions. Therefore, learning-based approaches show promise in constructing certificates from data. For example, Deep Learning techniques were utilized for synthesizing Lyapunov functions in [3][4][5].

In this paper, we are interested in leveraging tools from Koopman Operator theory towards learning Lyapunov functions by efficient utilization of data, in an interpretable manner. These infinite dimensional linear operators provide powerful and practical techniques that allow well-developed control and analysis principles from linear systems to be applied

All the authors are with the Department of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, deka@kth.se, dimos@kth.se. This work is supported by EU CANOPIES, VR, and KAW Foundation.

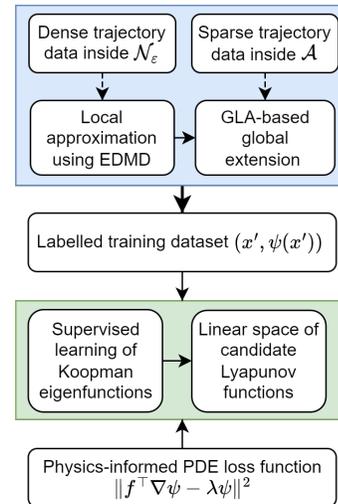


Fig. 1. Flowchart summarizing the use of local and global trajectory data in conjunction with EDMD and GLA approaches to learn eigenfunction parameterized Lyapunov functions. The bottom-most block is optional, in case system dynamics  $f$  is unknown.

to nonlinear dynamical systems. Koopman theory has been successfully utilized for a wide range of applications, including modelling and identification in robotics, nonlinear optimal control and MPC via global bi-/linearization, as well as dynamical systems analysis (please see [6] and the references therein). The eigenfunctions of Koopman operators encode useful information about the underlying dynamics, such as invariant sets and stable/unstable manifolds, and are central to our work. In this paper, we are primarily interested in finding Lyapunov functions, for which we utilize Koopman eigenfunctions. We propose a data-driven technique to learn Lyapunov functions parameterized via Deep Neural Networks (DNN), that are interpretable through the lens of Koopman operator theory: We construct Lyapunov basis functions using Koopman eigenfunctions, which then provide a linear space of Lyapunov candidates. The linearity of this space is instrumental in further refining it and formally verifying the learnt Lyapunov functions [7].

Towards that end, we present a novel learning-based framework that efficiently integrates together and complements the strengths of different techniques, namely Extended Dynamic Mode Decomposition (EDMD) [8][9], Generalized Laplace Analysis (GLA) [10][11], and Deep Learning [12]. Thus, the main contributions of our paper are as follows: (i) We leverage dense “local trajectories” (that are initialized inside a local neighborhood  $\mathcal{N}_\epsilon$  of the equilibrium) and sparse “global

trajectories” (that are initialized anywhere inside the domain of attraction  $\mathcal{A} \supset \mathcal{N}_e$ ) by combining EDMD with GLA, to create explicit ground truth data for Koopman eigenfunctions in absence of a dynamic model. Ground truth data means that corresponding to any point  $x' \in \mathcal{A}$ , we can obtain the actual value of an unknown eigenfunction  $\psi$  evaluated at  $x'$ . Figure 1 outlines our proposed approach. (ii) Exploiting the ground truth labels  $(x', \psi(x'))$  renders our learning problem into a simplified regression problem, allowing the use of vanilla neural network architectures. This distinguishes our work from related literature [7][13][14][15], that design their training loss criterion primarily on the eigenfunction PDE equation and/or prediction error, and commonly require additional regularization, like autoencoders within their architecture. Similarly, [3][4][5] train indirectly using so-called “Lyapunov risk” which penalizes violation of Lyapunov condition. (iii) Finally, we provide technical results to complement our proposed computation technique, leading to numerically well-behaved implementation<sup>1</sup>, as often times, Laplace averages suffer from convergence issues due to exponential terms within the integral [11].

## II. BACKGROUND

**Notations:**  $Re(\cdot)$  denotes the real part of its argument, which can be a scalar or a complex-valued function.  $\mathcal{L}_f g$  represents the Lie-derivative of a function  $g(x)$  with respect to vector field  $f(x)$ . The space of continuously differentiable functions on  $X$  is denoted by  $C^1(X)$ . The notation  $\text{spec}(M)$  denotes the spectrum of a matrix  $M \in \mathbb{C}^{n \times n}$ . Positive definiteness of a matrix  $P$  is denoted using  $P > 0$ .  $\lambda_{\min}(\cdot)$  and  $\lambda_{\max}(\cdot)$  are eigenvalues with smallest and largest magnitudes, respectively.  $\mathbb{R}$  denotes set of reals and  $\mathbb{C}$  is the complex set. The notation  $\|\cdot\|$  is used for 2-norm of a complex vector.

We now consider the nonlinear dynamical system

$$\frac{dx}{dt} = f(x) \quad (1)$$

where function  $f : X \rightarrow \mathbb{R}^n$  is continuously differentiable and states  $x(t)$  evolve in the compact set  $X \subset \mathbb{R}^n$  for all  $t \geq 0$ . Let  $\phi_t : X \rightarrow X$  denote the flow-map of this system, that is,  $\phi_t(x) = x + \int_0^t f(x(s))ds$ , for all  $t \geq 0$  and  $x \in X$ .

**Definition 1.** The Koopman semigroup<sup>2</sup> of operators is defined as the linear operators  $\mathcal{K}_t$  acting on a (Banach) space  $\mathcal{F}$  of functions  $g : X \rightarrow \mathbb{C}$  such that  $[\mathcal{K}_t g](x) = g(\phi_t(x))$ , for every  $g \in \mathcal{F}$ . Elements of  $\mathcal{F}$  are referred to as ‘observables’.

In the rest of this paper, we shall take our observable space  $\mathcal{F}$  to be the Banach space  $C^1(X)$  due to mathematical convenience. Corresponding to this infinite dimensional linear operator, one can define Koopman eigenfunctions as functions  $\psi(x) \in \mathcal{F}$  that satisfy  $[\mathcal{K}_t \psi](x) = e^{\lambda t} \psi(x)$  for some constant  $\lambda \in \mathbb{C}$  (in other words,  $\frac{d}{dt} \psi(x(t)) = \lambda \psi(x(t))$ ). Scalar  $\lambda$  is the eigenvalue associated with eigenfunction  $\psi$ .

Assume system (1) has an asymptotically stable equilibrium at  $x_e$ . We call a Koopman eigenfunction  $\psi$  as a *principle eigenfunction* if  $\psi(x_e) = 0$  and  $\nabla_x \psi(x_e) \neq 0$ , such that its corresponding eigenvalue belongs to  $\text{spec}(\nabla_x f(x_e))$ . Given the

semigroup of eigenpairs  $E$  (which is the set comprising of  $(\lambda, \psi)$  pairs), the set of principle eigenpairs are the minimal generator  $G$  of the set  $E$ . That is,

$$E = \left\{ \left( \sum_{i=1}^m n_i \lambda_i, \prod_{i=1}^m \psi_i^{n_i} \right) \mid (\lambda_i, \psi_i) \in G, m \in \mathbb{N}, n_i \in \mathbb{N} \right\}.$$

For systems with a hyperbolic equilibrium  $x_e$ , the uniqueness and existence of principle eigenpairs can be characterized by  $\text{spec}(\nabla_x f(x_e))$  under conditions of nonresonance [16].

**Definition 2.** Let  $\lambda \in \mathbb{C}$  be a scalar and  $g : X \rightarrow \mathbb{C}$  be an observable. Then the Laplace average of  $g$  is defined as

$$g_\lambda^*(x) \doteq \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{-\lambda t} g(\phi_t(x)) dt. \quad (2)$$

If the above limit converges for the observable  $g$ , this Laplace average  $g^*$  can be verified to be a Koopman eigenfunction corresponding to eigenvalue  $\lambda$ , since

$$\begin{aligned} [\mathcal{K}_\tau g_\lambda^*](x) &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{-\lambda t} [\mathcal{K}_{t+\tau} g](x) dt \\ &= e^{\lambda \tau} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{-\lambda(t+\tau)} [\mathcal{K}_{t+\tau} g](x) dt \\ &\stackrel{s=t+\tau}{=} e^{\lambda \tau} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{-\lambda s} [\mathcal{K}_s g](x) ds = e^{\lambda \tau} g_\lambda^*(x). \end{aligned}$$

Note that stable Koopman eigenfunctions can be used to describe invariant sets [17]. Given Koopman eigenfunctions  $\psi_i(x) : \mathcal{A} \subseteq X \rightarrow \mathbb{C}$  for  $i = 1, \dots, N$  (with corresponding eigenvalues  $\lambda_i \in \mathbb{C}$  with negative real parts), we can define functions  $V_i(x) \doteq \frac{1}{2} \|\psi_i(x)\|^2 \geq 0$ , which means  $\dot{V}_i(x) = Re(\lambda_i) \|\psi_i(x)\|^2 \leq 0$ . This implies every  $\gamma$ -sublevel set of  $V_i$  denoted by  $M_i^\gamma \doteq \{x \mid V_i(x) \leq \gamma\}$  is forward-invariant. In particular, it follows that all trajectories starting inside  $\mathcal{A}$  converge to the set  $M_i^0$ , due to LaSalle’s Invariance Principle, i.e.,  $\phi_t(x) \rightarrow \bigcap_i M_i^0, \forall x \in \mathcal{A}$ . We use these  $V_i$ ’s in our final step for constructing Lyapunov functions from data.

## III. MAIN RESULTS

In this section and the rest of the paper, we shall focus our attention on nonlinear systems that have an asymptotically stable equilibrium point (taken to be origin without loss of generality). We first present convergence results for Laplace averages based on easy to verify conditions. Next, we show how one may use trajectory data to exactly compute Koopman eigenfunctions using the combination of EDMD and GLA. In the final part of this section, we present how the learned eigenfunctions are used for linearly parameterizing a space of candidate Lyapunov functions.

### A. Convergence results for Laplace averages

GLA has been discussed widely in [10][11] as an analysis tool for obtaining Koopman mode decomposition of observables onto the space spanned by eigenfunctions. It is important to note that the computation of Laplace averages in general is not numerically amenable, due to convergence issues posed by the exponentially growing term within the integral. Thus, we first present some convergence results that ultimately aid us in their numerical computations.

<sup>1</sup>Code for this paper can be accessed online by clicking [\[this link\]](#).

<sup>2</sup>Informally, a semigroup is a set that is closed under an associative binary operation. Particularly, note that  $\mathcal{K}_{t_1} \cdot \mathcal{K}_{t_2} = \mathcal{K}_{t_2} \cdot \mathcal{K}_{t_1} = \mathcal{K}_{t_1+t_2}$ .

**Lemma 1.** Consider the system (1) with asymptotically stable equilibrium  $x_e = 0$ . If an observable  $g$  with an isolated root at  $x_e$  locally satisfies  $\mathcal{L}_f g = \lambda g$  (for a stable  $\lambda$  with  $\text{Re}(\lambda) < 0$ ) in some  $\varepsilon$ -neighborhood  $\mathcal{N}_\varepsilon$  of the equilibrium  $x_e$ , then  $g_\lambda^*(x)$  is well-defined for all  $x$  in the domain of attraction  $\mathcal{A} \supseteq \mathcal{N}_\varepsilon$ .

*Proof.* Without loss of generality, we can consider  $\mathcal{N}_\varepsilon$  to be forward-invariant due to the following. Since  $x_e$  is an isolated root of  $g$ , by definition, there is a positive  $r$  such that  $\|g(x)\| > 0$  if  $0 < \|x - x_e\| \leq r$ . Thus, for any  $\alpha < \min_{\|x - x_e\| = r} \|g(x)\|$ , the  $\alpha$ -sublevel set of  $\|g\|$  has a connected component  $C_\alpha$  strictly contained inside the ball  $\|x - x_e\| \leq r$ . This ball can itself be contained inside  $\mathcal{N}_\varepsilon$  by choosing a small enough  $r$ , so that  $C_\alpha \subseteq \mathcal{N}_\varepsilon$ . Since  $\frac{d}{dt} \|g\|^2 = 2\text{Re}(\lambda) \|g\|^2 < 0$  inside  $\mathcal{N}_\varepsilon$ , the set  $C_\alpha$  is forward-invariant.

Next, for any point  $x \in \mathcal{A}$ , let us define  $T^*(x)$  as

$$T^*(x) = \inf_{t \geq 0} \{t \mid \phi_t(x) \in \mathcal{N}_\varepsilon\} \quad (3)$$

which is the time it takes for a trajectory starting at point  $x$  to reach the set  $\mathcal{N}_\varepsilon$ , and is always finite for all  $x$  in the region of attraction  $\mathcal{A}$ . Thus, the Laplace average is  $g_\lambda^*(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \left[ \int_0^{T^*} e^{-\lambda t} g(\phi_t(x)) dt + \int_{T^*}^T e^{-\lambda t} g(\phi_t(x)) dt \right]$ . The first term in the right-hand side of the above equation is equal to zero since the integral is finite. In the second term,  $t \geq T^*(x)$ , which by definition (3) along with the forward-invariance of  $\mathcal{N}_\varepsilon$  means that  $\phi_t(x) \in \mathcal{N}_\varepsilon$ . Furthermore, inside this set,  $\mathcal{L}_f g = \lambda g$ . Thus,  $g(\phi_t(x)) = e^{\lambda(t-T^*)} g(\phi_{T^*}(x))$  for all  $t > T^*$ , giving us

$$g_\lambda^*(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{T^*}^T e^{-\lambda t} e^{\lambda(t-T^*)} g(\phi_{T^*}(x)) dt = e^{-\lambda T^*} g(\phi_{T^*}(x)) \quad (4)$$

which is finite for all  $x \in \mathcal{A}$  since  $g$  is continuous.  $\square$

Hartman-Grobman theorem applies to the asymptotically stable system of Lemma 1 (and any system with a hyperbolic equilibrium in general), guaranteeing the existence of a local neighborhood of the equilibrium where the flow of the nonlinear system is homeomorphic to that of its linearization. One may note that (3)-(4) provide a more numerically robust way of computing the Laplace average compared to equation (2), since the approximation errors in the function  $g(x)$  may accumulate and grow exponentially under the integral. We show in the following theorem how the Laplace average computations of any function may be robust to approximation errors up to a certain order.

**Theorem 1.** Let us again consider the function  $g(x)$  as described in Lemma 1 and its approximation  $\hat{g}(x)$  such that  $\|g(x) - \hat{g}(x)\| \doteq \xi(x) = o(\|x\|^2)$ . For any matrices  $P, Q > 0$  and  $A = \nabla_x f(0)$  satisfying the Lyapunov equation  $PA + A^\top P = -Q$ , if the matrix

$$-A - \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} I \quad (5)$$

is Hurwitz, then  $\hat{g}_\lambda^*(x) = g_\lambda^*(x) \forall \lambda \in \text{spec}(A)$  and  $\forall x \in \mathcal{A}$ .

*Proof.* Let us pick any  $\lambda$  from the set  $\text{spec}(A)$ . Since  $g$  satisfies conditions of Lemma 1, its Laplace average  $g_\lambda^*$  is well-defined on the entire set  $\mathcal{A}$ . Now, using the triangular inequality for integrals in equation (2), we get

$$\hat{g}_\lambda^*(x) - g_\lambda^*(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{-\lambda t} [\hat{g} - g](\phi_t(x)) dt$$

$$\Rightarrow \|\hat{g}_\lambda^* - g_\lambda^*\| \leq \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T e^{-\lambda t} \xi(\phi_t(x)) dt.$$

We then need to show that the integrand in the above inequality converges to zero under the stated conditions.

Let us rewrite the right-hand side of dynamics (1) as  $f(x) = Ax + h(x)$ , where  $A = \nabla_x f(0)$  (assuming  $f$  is analytic and the equilibrium is hyperbolic allows to do this). One can show that since  $h(0) = 0$  and  $\nabla_x h(0) = 0$ ,  $\|h(x)\| \leq c_\varepsilon \|x\|^2$  for a positive constant  $c_\varepsilon$  in some compact set  $\|x\| \leq \varepsilon$  using the Mean Value Theorem for function in several variables [18]. We can then estimate the convergence rate of  $\|x\|^2$  in this compact set. Let the matrix  $P > 0$  be the solution to the Lyapunov equation  $PA + A^\top P = -Q$  for a given matrix  $Q > 0$ . This means  $\frac{d}{dt} x^\top P x = -x^\top Q x + 2x^\top P h(x)$ .

Now, in the region  $\|x\| \leq \varepsilon$ ,

$$\begin{aligned} \frac{d}{dt} x^\top P x &\leq -\lambda_{\min}(Q) \|x\|^2 + 2c_\varepsilon \|P\| \cdot \|x\|^3 \\ &\leq -\lambda_{\min}(Q) \|x\|^2 \left(1 - \frac{2\varepsilon c_\varepsilon \|P\|}{\lambda_{\min}(Q)}\right) \leq -\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \kappa_\varepsilon x^\top P x, \end{aligned}$$

where  $\kappa_\varepsilon \doteq 1 - \frac{2\varepsilon c_\varepsilon \|P\|}{\lambda_{\min}(Q)}$  increases monotonically as  $\varepsilon$  gets smaller (and  $\varepsilon$  is henceforth taken to be small enough so that  $\kappa_\varepsilon > 0$ ). Therefore, it follows that  $\|x\|^2$  exponentially decays with rate  $\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \kappa_\varepsilon$ . Since the matrix expression in equation (5) is Hurwitz, every eigenvalue  $\lambda$  of  $A$  satisfies  $\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} + \text{Re}(\lambda) > 0$ . We can pick  $\varepsilon$  small enough such that  $\kappa_\varepsilon$  is arbitrarily close to 1, thus leading to

$$\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} \kappa_\varepsilon + \text{Re}(\lambda) \doteq \Delta_\lambda > 0$$

by continuity. Finally, whenever  $\|x\| \leq \varepsilon$ , we get

$$\frac{\xi(\phi_t(x))}{\|\phi_t(x)\|^2} \geq \frac{\xi(\phi_t(x))}{e^{\text{Re}(\lambda)t} \|x\|^2} \geq \frac{\|e^{-\lambda t} \xi(\phi_t(x))\|}{\varepsilon^2}.$$

Since  $\xi(x) = o(\|x\|^2)$ , taking the limit  $t \rightarrow \infty$  on both sides, we obtain  $\lim_{t \rightarrow \infty} \|e^{-\lambda t} \xi(\phi_t(x))\| = 0$ .

All trajectories starting anywhere in  $\mathcal{A}$  will eventually enter the region  $\|x\| \leq \varepsilon$ , and so this limit holds true for all  $x$  in  $\mathcal{A}$ . This completes the proof.  $\square$

**Remark.** Equation (5) imposes a condition on the *distribution* of the eigenfunctions of  $A$ , such that they are not spread too far apart. Intuitively, if  $\text{Re}(\lambda_a) \ll \text{Re}(\lambda_b) < 0$ , then the Laplace average of an observable  $g$  corresponding to  $\lambda_a$  (i.e.  $g_{\lambda_a}^*$ ) may not converge, since the exponential term  $e^{-\lambda_a t}$  inside integral (2) will grow very fast, whereas  $g(\phi_t(x))$  will decay relatively slower due to the ‘‘slow’’ eigenvalues in  $\text{spec}(A)$ .

**Numerical Example 1.** Consider a system with asymptotically stable equilibrium at the origin as follows:

$$\begin{aligned} \dot{x}_1 &= (-7.2x_1^3 + 7.2x_1 - 2.4x_2^3 + 2.4x_2) / (7(3x_1^2 - 1)) \\ \dot{x}_2 &= (-0.8x_1^3 + 0.8x_1 - 6.8x_2^3 + 6.8x_2) / (7(3x_2^2 - 1)). \end{aligned}$$

For this system, we obtain eigenvalues  $\lambda_1 = -1.2$ ,  $\lambda_2 = -0.8$  corresponding to  $A = \begin{bmatrix} -1.03 & -0.11 \\ -0.34 & -0.97 \end{bmatrix}$ . Taking  $Q$  to be the identity matrix and solving  $PA + A^\top P = -Q$ , we get  $P = \begin{bmatrix} 0.52 & -0.12 \\ -0.12 & 0.53 \end{bmatrix}$ . Thus, the matrix in equation (5) is

$\begin{bmatrix} -0.52 & 0.11 \\ 0.34 & -0.58 \end{bmatrix}$  which satisfies the Hurwitz condition of Theorem 1. Additionally, corresponding to the eigenvalue  $-0.8$ , this system has a principle eigenfunction  $\psi(x) = x_1^3 - x_1 - 2x_2^3 + 2x_2$ , which means that the function  $g(x) = \psi(x)$  satisfies the condition of Lemma (1). Thus, any approximation of  $g(x)$  within an error of  $o(\|x\|^2)$  must result in the same Laplace average  $g_{-0.8}^*(x)$  according to Theorem 1. We revisit this later in Example 2.

## B. Numerical computation from trajectory data

We have shown in the previous subsection how any function  $g$  defined on  $X$  that locally satisfies  $\mathcal{L}_f g = \lambda g$  in some neighborhood of the asymptotically stable equilibrium results in a well-defined Laplace average  $g_\lambda^*(x)$  everywhere in the region of attraction  $\mathcal{A}$ . This function  $g_\lambda^*$  can then be used further as a Koopman eigenfunction. However, such a function  $g$  itself needs to be obtained first. In this section, we see how EDMD can be effectively combined with GLA to compute Koopman eigenfunctions through dense local trajectory data near  $x_e$  and sparse global trajectory data throughout the set  $\mathcal{A}$ . We propose to first utilize just local trajectory data with EDMD towards obtaining accurate local estimates of suitable functions  $g$ . With these local estimates, we compute their Laplace averages  $g_\lambda^*$  defined over the larger domain  $\mathcal{A}$ . The local estimates via EDMD are expected to have approximation errors, but thankfully, through Theorem 1, the GLA computations are shown to be robust to small ( $o(\|x\|^2)$ ) approximation errors in  $g$ . One may use the global trajectory data to directly perform EDMD and estimate Koopman eigenfunctions defined over the entire set  $\mathcal{A}$ . However, this may lead to poor estimates if global trajectory data is insufficient; thus utilizing dense, local data for local estimates is a more reasonable alternative.

We now outline the EDMD process, which is a well studied data-driven algorithm to approximate the Koopman operator in finite dimensions by lifting the state space into a higher dimensional embedding wherein the dynamics are (approximately) linear [8][9]. This approximation is obtained as a solution to a least squares optimization problem. Consider the function  $\theta(x) = [\theta_1(x), \theta_2(x), \dots, \theta_N(x)]^\top$  comprised of basis functions  $\theta_i(x) \in \mathcal{F}$ ,  $i = 1, 2, \dots, N$ . Given that we have trajectory snapshots at some uniform sampling time  $\tau$ , in form of  $M$  pairs  $(x_i, y_i) \in \mathcal{N}_\varepsilon \times \mathcal{N}_\varepsilon$  where  $y_i = \phi_\tau(x_i)$  for  $i = 1, 2, \dots, M$ , the EDMD procedure is used to estimate a finite dimensional approximation of the Koopman Operator  $\mathcal{K}_\tau$  restricted to the space spanned by the basis functions. This is expressed as the Koopman matrix  $K$ , by solving the following least-squares problem:

$$K = \arg \min_{A \in \mathbb{C}^{N \times N}} \|\theta(Y) - A\theta(X)\|_F^2 \quad (6)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm, and matrices  $\theta(X) = [\theta(x_1), \theta(x_2), \dots, \theta(x_M)]$  and  $\theta(Y) = [\theta(y_1), \theta(y_2), \dots, \theta(y_M)]$ . The  $K$  that minimizes (6) is obtained in closed-form as  $K = \theta_{XY} \theta_{XX}^\dagger$ , where  $\dagger$  denotes the pseudo-inverse,  $\theta_{XY} = \theta(Y)\theta(X)^\top$  and  $\theta_{XX} = \theta(X)\theta(X)^\top$ . Suppose that  $(v_i, \mu_i)$  are the eigenvectors and eigenvalues of  $K^\top$  for  $i = 1, 2, \dots, N$ . Then, in the set  $\mathcal{N}_\varepsilon$ , letting  $g_i(x) \doteq v_i^\top \theta(x)$  and  $\lambda_i \doteq \frac{1}{\tau} \log(\mu_i)$ , we get

$$\begin{aligned} \theta(\phi_\tau(x)) \approx K\theta(x) &\Rightarrow g_i(\phi_\tau(x)) \approx \mu_i g_i(x) = e^{\lambda_i \tau} g_i(x) \\ &\Leftrightarrow \mathcal{L}_f g_i(x) \approx \lambda_i g_i(x). \end{aligned} \quad (7)$$

Next, with these functions  $g_i$  obtained by EDMD, we compute the Laplace averages  $g_{\lambda_i}^*(x)$  for any point  $x \in \mathcal{A}$ , by ‘unrolling’ the trajectory  $\phi_t(x)$  starting at  $x$  forward for a sufficiently large time<sup>3</sup>, followed by numerically integrating (2) using the unrolled trajectory. Note that one can combine the trajectory unrolling and GLA integration steps into one, by augmenting the system dynamics (1) using a new scalar state variable  $z$  as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} f(x(t)) \\ \frac{1}{t+\delta} (-z(t) + e^{-\lambda t} g(x(t))) \end{bmatrix}, \quad t \geq 0 \quad (8)$$

for some small constant  $\delta > 0$ . If  $z(0) = 0$ , this state  $z$  can be shown to evolve as  $z(t) = \frac{1}{t+\delta} \int_0^t e^{-\lambda s} g(\phi_s(x)) ds$  for all  $t \geq 0$ . Thus, when  $z(0) = 0$  and  $x(0) = x$ , we get  $z(t) \rightarrow g_\lambda^*(x)$  as  $t \rightarrow \infty$ . In other words, unrolling this augmented dynamics (8) from an initial point  $(x, 0)$  gives us the Laplace average through the augmented state variable  $z$ .

Once we obtain the ground truth values of the eigenfunctions at randomly sampled points in  $\mathcal{A}$ , we create a training dataset to learn DNN-based approximations of the eigenfunctions. Due to space limitations, we kindly refer readers to our code link for details.

**Numerical Example 2.** Consider the system described in Example 1 with an asymptotically stable equilibrium at the origin. We consider the neighbourhood  $\mathcal{N}_\varepsilon = [-0.05, 0.05] \times [-0.05, 0.05]$ , and sample 100 trajectories of length 100 seconds each, with a sampling period  $\tau = 0.01$ s. Basis functions  $\theta_i$  are taken to be monomials of degree up to 3. From EDMD, we obtain

$$g(x) = 1.014x_1^3 - x_1 - 2.027x_2^3 + 2.00x_2$$

corresponding to the principle eigenvalue  $\lambda = -0.8$ , which is a very close estimate of the actual principle eigenfunction  $x_1^3 - x_1 - 2x_2^3 + 2x_2$ , with an approximation error  $\xi(x)$  in the order of  $o(\|x\|^2)$ . We compute the Laplace average  $g_{-0.8}^*(x)$  at randomly sampled points in  $[-0.4, 0.4] \times [-0.4, 0.4]$  using this  $g(x)$  obtained from EDMD using trajectory data within a much smaller domain  $\mathcal{N}_\varepsilon$ . Figure 2 compares this Laplace average with the actual eigenfunction  $x_1^3 - x_1 - 2x_2^3 + 2x_2$ .

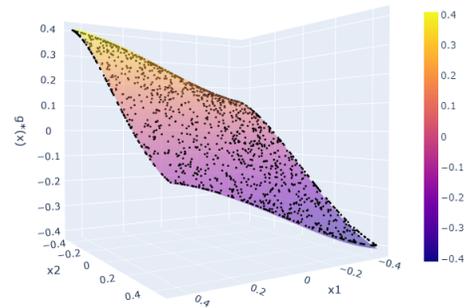


Fig. 2. Actual eigenfunction  $x_1^3 - x_1 - 2x_2^3 + 2x_2$  (translucent surface) vs computed Laplace average  $g_{-0.8}^*(x)$  evaluated at random points (black dots).

<sup>3</sup>In practice, this simply means until  $\frac{1}{T} \int_0^T e^{-\lambda t} g(\phi_t(x)) dt$  numerically converges to a steady state.

### C. Lyapunov certificates from Koopman eigenfunctions

Our main goal for this paper is to develop a data-driven technique that efficiently leverages trajectory samples of a dynamical system to construct Lyapunov certificates. We particularly focus on developing an approach suitable for high-dimensional systems wherein several learning techniques may need to be combined together to make the best use of the available trajectory data.

One can use functions  $V_i = \frac{1}{2}\|\psi_i(x)\|^2$  in a number of different ways to construct Lyapunov functions, but we are primarily interested in the linearly parameterized space of candidate Lyapunov functions given by

$$\mathcal{V}_{Lyap} = \left\{ \sum_i \alpha_i V_i(x) \mid \alpha_i \in [0, 1], \sum_i \alpha_i \neq 0 \right\}. \quad (9)$$

Each  $V \in \mathcal{V}_{Lyap}$  is a Lyapunov function since  $V(x) \geq 0$  and  $\dot{V}(x) = \sum_i 2\alpha_i \text{Re}(\lambda_i) V_i(x) \leq 0$  by construction.  $\mathcal{V}_{Lyap}$  is particularly interesting for the case when Koopman eigenfunction  $\psi_i$ 's are represented as polynomials. In that case, each element in  $\mathcal{V}_{Lyap}$  is a sum-of-squares (SOS) polynomial, and one can leverage SOS semi-definite programming to verify Lyapunov conditions (when the system dynamics are also polynomial)[7].

In our prior work [7], we discuss how inaccuracies in the computation of Koopman eigenfunction via data-driven techniques may lead to violations in the Lyapunov conditions for functions  $V_i$  and consequently, not every element of the space  $\mathcal{V}_{Lyap}$  would be a valid Lyapunov candidate for establishing asymptotic stability guarantees for the equilibrium point/manifold. Nevertheless, thanks to linearity of the space  $\mathcal{V}_{Lyap}$ , it was shown that under certain approximation error bounds on the basis  $V_i$ , one can still pick functions  $V(x) \in \mathcal{V}_{Lyap}$  that satisfy  $\dot{V}(x) \leq \beta(\gamma - V(x))$  for some constants  $\beta, \gamma > 0$ , thus providing certificates for forward-invariance. In this paper however, we develop efficient computation techniques for Koopman eigenfunctions which are more accurate in comparison to [7], enabling us to directly obtain Lyapunov functions via the estimated eigenfunctions through (9), as evidenced from our numerical experiments.

## IV. NUMERICAL RESULTS

In this section, we demonstrate our method towards construction of Lyapunov certificates for nonlinear consensus in a 10-dimensional multi-agent system. We demonstrate how trajectory data from the agents are used in the absence of agent model and communication graph to represent the dynamics via Koopman eigenfunctions, which then serve as a building block for obtaining Lyapunov functions.

Consider a graph  $\mathcal{G}$ , with vertices denoted by the set  $\mathcal{V} = \{0, \dots, n-1\}$  and edge set denoted by  $\mathcal{E} = \{(i, j), \dots, m\}$ . The set of neighboring nodes of agent  $i$  is denoted by  $N_i \subset \mathcal{V}$ . We are interested in consensus problem for multi-agent systems corresponding to this network  $\mathcal{G}$ , wherein scalar state  $x_i \in \mathbb{R}$  for each agent  $i \in \mathcal{V}$  is required to converge to a common value. Let us next consider the following nonlinear distributed consensus protocol for single-integrator agent dynamics:

$$\dot{x}_i = u_i = -\sigma_i(x_i) \sum_{j \in N_i} a_{ij}(x_i - x_j), \quad i \in \mathcal{V} \quad (10)$$

where  $\sigma_i(\cdot)$  is continuous and positive for all  $i \in \mathcal{V}$ , and  $a_{ij}(\cdot)$  is Lipschitz continuous for all  $(i, j) \in \mathcal{E}$  with  $x a_{ij}(x) > 0$  when  $x \neq 0$  and  $a_{ij}(0) = 0$ .

It was shown in [19] that if  $\mathcal{G}$  is a connected, undirected graph and  $a_{ij}(\cdot)$  are odd functions for all  $(i, j) \in \mathcal{E}$ , then the states  $x$  converge to a point  $x^*$  uniquely determined by  $\sum_{i \in \mathcal{V}} \int_0^{x^*} \frac{1}{\sigma_i(y)} dy = \sum_{i \in \mathcal{V}} \int_0^{x_i(0)} \frac{1}{\sigma_i(y)} dy$ . The converse holds as well. The proof utilizes a Lyapunov function of the form  $V(x) = \sum_{i \in \mathcal{V}} \sum_{j \in N_i} \int_0^{x_i - x_j} a_{ij}(y) dy$ .

Note that this Lyapunov function needs knowledge of both the underlying dynamics of the system as well as the graph topology. Our goal is to directly learn a Lyapunov function just from the trajectory data of the agents. As a specific example, we consider a  $n = 10$  agent system, with  $a_{ij}(y) = r_{ij} [1.2 \tanh(y) + \sin(y)]$ , for some random constants  $r_{ij} \in (0, 1)$ ,  $\sigma_i(y) = \frac{1}{1 + \exp(-y)}$ , and graph  $\mathcal{G}$  as follows:

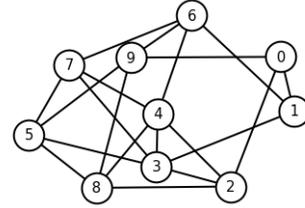


Fig. 3. A 10 agent system with fully connected network

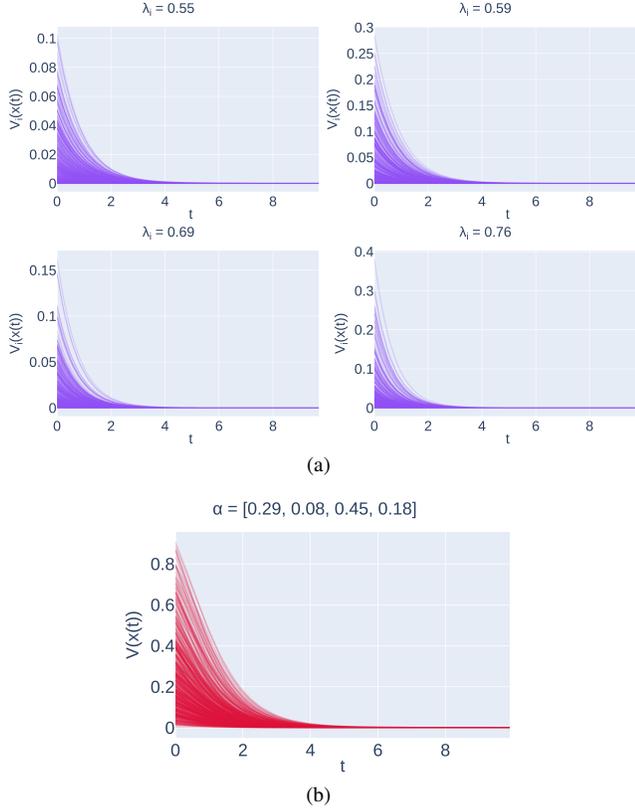
We modify this system slightly by adding a linear term  $-x_i$  to the right-hand side of equation (10) to ensure that the origin is an asymptotically stable isolated equilibrium and that condition (5) holds. We begin our approach with the EDMD step, by sampling  $N = 100$  trajectories of length 10 seconds from within a local neighborhood  $\mathcal{N}_\epsilon = [-0.05, 0.05]^n$ , with sampling interval of 0.01s. We pick monomials of degree up to 3 as our choice of basis functions  $\theta(x)$  for EDMD. By allowing the sampling neighbourhood  $\mathcal{N}_\epsilon$  to be small in our approach, we are able to precisely extract principle eigenvalues of this system, of which we pick four with the smallest magnitude, given by  $\lambda_1 = -0.55$ ,  $\lambda_2 = -0.59$ ,  $\lambda_3 = -0.69$ , and  $\lambda_4 = -0.76$ . The local eigenfunction estimates  $g_i$  (that are valid within  $\mathcal{N}_\epsilon$ ) corresponding to these four eigenvalues are used to compute Laplace averages over a much larger domain  $[-0.45, 0.45]^n$ . The Laplace average computations are numerically stable and the integral (2) converges, in agreement with our technical results.

In the final step, we train the DNN parameterizing principle eigenfunctions  $\psi_i(x)$  corresponding to those four principle eigenvalues, over the larger domain  $[-0.45, 0.45]^n$  using supervised learning, as outlined in Figure 1. We use a multilayer perceptron (MLP) architecture with sine activation function. For every  $i = 1, \dots, 4$ , our MLP  $\psi_i$  consists of 3 hidden layers with 128 neurons each. The loss is comprised of a supervised learning term  $L_1(x)$  and a physics-informed term  $L_2(x)$ :

$$\underbrace{\|\psi_i(x) - g_{\lambda_i}^*(x)\|^2}_{\doteq L_1(x)} + 0.01 \cdot \underbrace{\|\mathcal{L}_f \psi_i(x) - \lambda_i \psi_i(x)\|^2}_{\doteq L_2(x)},$$

where  $g_{\lambda_i}^*$  is the Laplace average of the function  $g_i$  obtained by EDMD that locally satisfies (7). For this example, loss  $L_1$  is computed at 3000 randomly sampled points in  $[-0.45, 0.45]^n$  and  $L_2$  is computed at 2000 randomly sampled points in

$[-0.45, 0.45]^n$  during training. For further details, please see the associated Pytorch code implementation. In Figure 4, we show the Lyapunov basis functions  $V_i(x) = \frac{1}{2}\|\psi_i(x)\|^2$  corresponding to  $i = 1, \dots, 4$  along random trajectories, as well as an example of a Lyapunov function from linear combination of the basis  $V_i$ 's.



**Fig. 4.** Learned Lyapunov basis functions for (unknown) dynamics (10) with damping. (a) Each panel shows  $V_i(x) = \frac{1}{2}\|\psi_i(x)\|^2$  for  $i = 1, \dots, 4$  decreasing strictly monotonically along **500** randomly sampled trajectories in  $[-0.45, 0.45]^n$ , as expected from our data-driven construction. (b) Using the Lyapunov basis  $V_i$ 's, we construct our Lyapunov candidate using randomly chosen  $\alpha_i$ 's. The red lines show  $V$  decreases monotonically along **1000** trajectories randomly initialized on a larger domain  $[-1.5, 1.5]^n$  compared to the domain on which the eigenfunctions  $\psi_i$  were trained on.

## V. CONCLUSION AND DISCUSSION

This work presents a Deep Learning approach for construction of Lyapunov certificates, parameterized linearly using Koopman eigenfunctions of the underlying dynamical system. One of the novelty of our approach lies in the creation of *labelled training dataset* for supervised learning of Lyapunov basis functions via Koopman eigenfunctions, by combining EDMD and GLA. Trajectory data near the equilibrium are used to locally estimate eigenfunctions. Laplace averages of these local estimates computed along sparse but global trajectories are then used to evaluate eigenfunctions at sample points throughout the domain of attraction, generating ground truth function values for the final deep supervised learning step. The paper additionally contributes towards addressing numerical challenges involved with GLA computations, both in terms of analysis and efficient implementation.

The assumption on the second order estimation error bounds of eigenfunctions obtained in the EDMD step requires further investigation, and is a matter of our current focus. To the

best of our knowledge, only constant error bounds for EDMD are available in present literature. Additionally, once the ground truth values for the eigenfunctions are obtained, a key question to be answered is how well can a particular choice of function approximator (DNNs in our specific case) fit the synthesized training data. Studying the sample complexity of our approach as well as formal verification of the learnt Lyapunov function are therefore important future directions. Fortunately, despite both these open questions, our numerical examples successfully illustrate the practicality of our approach. For the low-dimensional example where the eigenfunctions are known in closed analytical form, the second order EDMD error bound is found to hold true and consequently our computed eigenfunction estimates exactly match the actual eigenfunctions. The high-dimensional example illustrates scalability of our proposed method, thereby alleviating concerns surrounding sample complexity of the deep learning step, since ground truth training labels can be generated efficiently for our 10-d system.

## REFERENCES

- [1] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, NJ, 2002.
- [2] M. Vidyasagar, *Nonlinear systems analysis*. SIAM, 2002.
- [3] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," *Advances in neural information processing systems*, vol. 32, 2019.
- [4] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, "Formal synthesis of Lyapunov Neural Networks," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 773–778, 2021.
- [5] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov Neural Network: Adaptive stability certification for safe learning of dynamical systems," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87, pp. 466–476, PMLR, 2018.
- [6] P. Bevanda, S. Sosnowski, and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," *Annual Reviews in Control*, vol. 52, pp. 197–212, 2021.
- [7] S. A. Deka, A. M. Valle, and C. J. Tomlin, "Koopman-based Neural Lyapunov functions for general attractors," in *IEEE 61st Conference on Decision and Control (CDC)*, pp. 5123–5128, IEEE, 2022.
- [8] 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, no. 6, pp. 1307–1346, 2015.
- [9] B. Huang and U. Vaidya, "Data-driven approximation of transfer operators: Naturally structured dynamic mode decomposition," in *American Control Conference (ACC)*, pp. 5659–5664, IEEE, 2018.
- [10] R. Mohr and I. Mezić, "Construction of eigenfunctions for scalar-type operators via Laplace averages with connections to the Koopman operator," *arXiv preprint arXiv:1403.6559*, 2014.
- [11] I. Mezić, "On numerical approximations of the Koopman operator," *Mathematics*, vol. 10, no. 7, p. 1180, 2022.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [13] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, pp. 1–10, 2018.
- [14] S. E. Otto and C. W. Rowley, "Linearly recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 558–593, 2019.
- [15] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems," in *American Control Conference (ACC)*, pp. 4832–4839, IEEE, 2019.
- [16] M. D. Kvalheim and S. Revzen, "Existence and uniqueness of global Koopman eigenfunctions for stable fixed points and periodic orbits," *Physica D: Nonlinear Phenomena*, vol. 425, p. 132959, 2021.
- [17] 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.
- [18] W. Fleming, *Functions of several variables*. Springer Science & Business Media, Berlin, 2012.
- [19] M. Andreasson, D. V. Dimarogonas, H. Sandberg, and K. H. Johansson, "Distributed control of networked dynamical systems: Static feedback, integral action and consensus," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1750–1764, 2014.